

UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

Facoltà di Ingegneria - Sede di Modena

Corso di Laurea in Ingegneria Informatica

WebProfessional **Web Content Management System**

Relatore

Chiar.ma Prof.ssa **Sonia Bergamaschi**

Tesi di Laurea di

Emidio Reggiani

Anno Accademico 2004-2005

Ringraziamenti

Desidero ringraziare la prof.ssa Sonia Bergamaschi e CP Software nella persona di Claudio Pioli per avermi dato la possibilità di sviluppare questa tesi.

Ringrazio la mia famiglia per avermi sostenuto in questi anni e per avere sempre creduto in me.

Un ringraziamento speciale a mia moglie Gabriella per essermi stata vicina ed avermi sostenuto in ogni istante di questo periodo così intenso.

Dedico questa tesi a mia moglie Gabriella, senza il cui aiuto non avrei mai raggiunto questo traguardo, e a mia figlia Chiara, che ha reso ancor più felice il mio matrimonio.

Emidio

Ringraziamenti

Contenuti

Premessa	9
Definizione di Content Management System (CMS).....	9
Vantaggi dei Content Management System.....	10
Tipi di Content Management System.....	11
Web Content Management System (WCMS).....	12
Vantaggi dei WCMS.....	13
CP Software e l'approccio al WCMS - WebProfessional.....	16
L'idea iniziale.....	17
eBusiness.....	18
WebPower.....	19
La piattaforma WebProfessional.....	19
Scelte Tecnologiche	21
Architettura a tre livelli.....	21
Interazione dell'utente con il database.....	22
Prima versione - Cold Fusion e MS Access.....	24
Cold Fusion.....	25
I limiti di Cold Fusion.....	25
I limiti di MS Access.....	26
Le nuove esigenze.....	27
La scelta di MySQL.....	28
L'ingresso dell'Open Source nelle imprese.....	28
<i>Cenni su PostgreSQL</i>	28
<i>Cenni su MySQL</i>	29
Confronto fra PostgreSQL e MySQL.....	29
<i>Analisi delle caratteristiche</i>	30
<i>Caratteristiche avanzate</i>	31
<i>MySQL in dettaglio</i>	32
La scelta di PHP.....	34
<i>Prime versioni di PHP</i>	34
<i>Relazione con l'HTML</i>	36
<i>Interpretato vs Compilato</i>	36
<i>Lo Zend Encoder</i>	37
ColdFusion vs PHP.....	37

Conclusioni sulle specifiche	39
La Piattaforma	41
Modalità ASP.....	41
Permission integrate nel Content Management	42
<i>Validazione tramite utente/gruppo</i>	44
<i>Validazione tramite priorità</i>	44
<i>Combinazione dei due metodi</i>	44
Gestione Utenti	44
Configurazione generale	48
Sezione "Company Info"	48
Sezione "Setup / Configuration"	49
Sezione "Setup / TopPages-Logo/Text"	50
Sezione "Groups/Users"	50
ACL Configuration.....	51
Integrazione utenti in un dominio NT	52
Tipo di autenticazione	52
Requisiti di sistema	52
<i>Configurazione dei client</i>	53
<i>Configurazione di IIS</i>	53
<i>Configurazione di WebProfessional</i>	54
<i>Funzione di autenticazione NTML - codice</i>	55
Integrazione dati - DB Sync	56
Il processo di sincronizzazione	56
<i>Tipi di sincronizzazione</i>	60
<i>Osservazioni su DB Sync</i>	61
I moduli principali di WebProfessional	61
WebPower	65
Le funzioni di WebPower.....	66
WizardBuilder.....	68
Status - Register	68
<i>Storico</i>	70
Status - Statistics	71
Status - Site-Static	72
Status - Start/Stop site	73
Status - Preview site	73

Menu/Page Builder	73
<i>Creare una voce di menù</i>	74
<i>Creare una voce di sottomenù</i>	75
<i>Eliminare una voce di menù/sottomenu</i>	75
<i>Creare i contenuti di una pagina</i>	75
<i>Introduzione all'editor</i>	78
<i>Le funzioni principali dell'editor</i>	80
eNews	81
<i>Categorie</i>	82
<i>Inserisci News</i>	82
<i>Elenco News</i>	83
Mailing-Lists	83
Upload/Download.....	85
Links.....	88
Promo	89
Conclusioni.....	90
eBusiness	91
Primo sviluppo - Catalogo OnLine	92
Il passaggio a PHP e l'integrazione con WebPower	94
La configurazione iniziale	95
Setup (Layout)	95
<i>Visualizzazione lista prodotti</i>	95
<i>Visualizza dettagli prodotti</i>	96
<i>Visualizza Categorie</i>	96
<i>Impostazioni testo</i>	96
<i>Visualizza pezzi ricambio</i>	99
<i>Visualizza varianti</i>	99
Setup (Configure).....	99
<i>Impostazioni eCommerce</i>	99
<i>Impostazioni ordini</i>	101
<i>Impostazioni avanzate</i>	101
<i>Mostra/Nascondi E-commerce top links</i>	102
<i>Etichette e-commerce</i>	102
<i>Mostra/Nascondi etichette nella visualizzazione prodotti</i>	102
Categorie di prodotti.....	102
L'introduzione dei "Tipi Categoria"	105

Prodotti.....	108
Proprietà	109
Proprietà avanzate.....	111
Dettagli.....	112
Ricerca prodotti.....	113
Unità di misura	114
Pezzi di ricambio e articoli correlati.....	115
Prezzi	117
<i>Unità di misura</i>	121
<i>Valuta</i>	122
Sconti	125
Gestione dell'IVA.....	129
Varianti	131
Catalogo su CD	141
Ordini e Fatture	142
Offerte	143
Condizioni di pagamento.....	144
Importazione ed esportazione dati	147
Importazione dati	147
Esportazione ordini	150
Conclusioni.....	150

Cenni su DbTools e WpOffice **151**

Il modulo DbTools.....	151
Configurazione di un database	152
Relazioni	153
Queries	153
<i>Tipi di Query</i>	154
<i>Step di creazione query</i>	154
User Actions	156
Import/Export	156
Packager.....	156
Il modulo WP Office.....	159
Call Center	159
Task.....	161
<i>Stati avanzamento task</i>	162
<i>Riepilogo attività</i>	163

Reports	164
Contacts	164
Agenda.....	165
<i>Inserimento di un appuntamento</i>	166
DocMan.....	166
CRM.....	168
Conclusioni sui moduli DbTools e WP Office	169
Case Studies	171
Corghi S.p.A. - WebPower	171
Corghi S.p.A.....	171
<i>La storia</i>	171
<i>L'azienda</i>	172
La gestione dei contenuti del sito.....	173
LuEn S.r.l. - WebPower ed eBusiness	177
LuEn S.r.l.	177
La presentazione dell'azienda e la produzione (WebPower).....	177
Il catalogo online (eBusiness)	180
Smeg S.p.A. - eBusiness	180
Smeg S.p.A.....	180
Esploweb	181
<i>Specifiche iniziali</i>	181
<i>Struttura del database e procedure</i>	185
<i>Conclusioni</i>	186
Conclusioni	187
Appendice	189
Notazione per gli schemi E-R.....	190
Definizione del database di WebProfessional	191
Tabella dei codici lingua utilizzati in WebProfessional	212
Relazioni nel database WebProfessional.....	214
Elenco delle relazioni	214
Dettagli delle relazioni.....	215
Bibliografia	249

Premessa

Il termine Content Management copre una moltitudine di accezioni e di prodotti, dalla gestione di documenti e record, pubblicazioni su web, a sostanzialmente qualsiasi tipo di informazione in formato elettronico. Alcuni prodotti si concentrano su specifici segmenti quali messaggistica oppure archiviazione documentale, altri si orientano a soluzioni generali ed omnicomprensive definite "Enterprise Content Management".

L'azienda che si rivolge al mercato IT alla ricerca di un sistema di Content Management si ritrova molto probabilmente confusa e sconcertata dall'elevato numero di differenti prodotti presenti e di soluzioni proposte. E' quindi indispensabile prima di procedere nella trattazione definire il termine "content management" e classificarne esattamente le varie tipologie in relazione ai contenuti da gestire.

Definizione di Content Management System (CMS)

Un CMS, a rischio di risultare ovvio, è un sistema che permette di organizzare e gestire contenuti. L'ovvietà e genericità di questa definizione genera un certo grado di confusione su cosa effettivamente si intende per CMS. L'ambiguità della definizione è sufficiente a far sì che una numerosissima schiera di prodotti si definiscano o mascherino come soluzioni di content management, e che qualsiasi tipo di problematica aziendale relativa alla gestione di informazioni in formato elettronico rientri nelle specifiche per una soluzione di content management.

Per capire effettivamente cosa un CMS è e fa, occorre definire:

- a cosa ci riferiamo quando parliamo di contenuto
- cosa intendiamo per gestione del contenuto
- cosa intendiamo per sistema di gestione dei contenuti

Solo avendo ben chiare queste definizioni è possibile identificare le reali soluzioni di content management e scegliere tra le decine di prodotti disponibili che rispondono ai requisiti.

Possiamo definire **contenuto** ogni tipo di informazione digitale. Testo, immagini, video,

record in una tabella di database,... In altre parole, qualsiasi cosa possa essere gestita in formato elettronico.

Per **gestione dei contenuti** si intende la gestione del contenuto combinando regole, processi e flussi per la ricerca, visualizzazione, modifica, accesso e quant'altro possa intendersi come "gestione" dei contenuti stessi.

Un **sistema di gestione dei contenuti** è uno strumento o un insieme di strumenti atto a generare in modo efficiente l'output desiderato utilizzando i contenuti gestiti.

Combinando le tre definizioni date, possiamo dire che un **Content Management System** è uno strumento che permette ad uno staff (tecnico e non) di creare, modificare, gestire e pubblicare nel formato desiderato una varietà di contenuti all'interno di regole, processi e flussi che assicurano la validità e la coerenza dei contenuti e delle pubblicazioni.

Vantaggi dei Content Management System

I sistemi di content management, oltre che la risposta all'esigenza nata negli ultimi anni di gestire informazioni non più in formato cartaceo, sono la naturale evoluzione dell'introduzione dei sistemi informatici nelle aziende.

L'informazione cartacea ad oggi aveva il pregio di essere controllata, strutturata, spesso processata (anche se manualmente), mentre la prima generazione di contenuti elettronici (email, lettere, documenti,...) poteva essere facilmente prodotta o riprodotta da chiunque, cosa che se da un lato ha favorito la diffusione del formato dall'altro ha mantenuto in essere la documentazione cartacea per tutti quei contenuti che necessitavano di autenticazione, protocollazione o altro.

La spinta alla strutturazione dei dati in CMS degli ultimi anni è dovuta a diversi fattori:

- la pressione interna derivante dalla necessità di gestire contenuti di dimensione crescente in modo pratico ed efficiente
- la consapevolezza dell'esistenza di soluzioni informative stabili a prezzi che non ne pregiudicano la messa in opera
- la facilità di integrazione delle soluzioni di content management di ultima generazione con i dati aziendali
- le legislazioni che rendono disponibili sempre più informazioni in formato elettronico e incentivano l'utilizzo delle stesse, adottando tecnologie per verificare l'autenticità dei

documenti

Nella realtà attuale i CMS sono divenuti il fulcro ed il motore delle strategie delle imprese integrandosi sempre più con i processi produttivi e l'organizzazione aziendale.

Tipi di Content Management System

Esistono diversi tipi di CMS, orientati a gestire diversi tipi di formato dati ed a pubblicare le informazioni su sistemi diversi. Genericamente si possono suddividere i CMS in alcune tipologie (che non hanno peraltro la pretesa di essere completamente esaustive, ma rappresentano una indicazione che copre la maggioranza delle soluzioni esistenti) basate sull'obiettivo della pubblicazione ed sui dati gestiti:

- Search (con obiettivo principale il reperimento dei dati su specifiche richieste)
- Knowledge Management (gestione del knowhow interno aziendale per applicazioni di supporto tecnico e percorsi di troubleshooting)
- Document Management (gestione documenti - classificazione, accessi, versioni)
- Archiving and Document Lifecycle Management (gestione flusso documentale, archiviazione, validità)
- Web Content Management (pubblicazione/utilizzo di informazioni su web, gestione degli accessi)
- Transactions and Enterprise Application Extensions
- Teams and Collaboration (distribuzione informazioni per lavoro di gruppo)
- Portals (portali)
- Rich Media and Digital Asset Management
- Messaging and eMail Management (sistemi di messaggistica)
- Workflow and Business Process Management (gestione processi produttivi, rilevazione e manipolazione dei dati di produzione)

E' inoltre possibile suddividere i CMS in quattro categorie principali basate sulle metodologie di gestione:

Content management frameworks - Un content management framework è più simile ad un toolkit di programmazione che ad un CMS. Fornisce funzionalità di base di workflow, templating e possibilità di personalizzazione, ma ogni altra funzionalità richiede lavoro di sviluppo, che presuppone una buona conoscenza del linguaggio di programmazione, della base di dati e capacità di analisi. Questo tipo di sistema risulta estremamente personalizzabile ma con alti costi di inizializzazione e viene solitamente utilizzato dalle grandi

imprese (es. Zope) che costituiscono un team di sviluppo interno oppure si appoggiano a strutture esterne (in dipendenza della sensibilità dei dati gestiti e della scelta di mantenere o meno il know-how sulla piattaforma interno all'azienda)

Page-based systems - Questo tipo di sistema è l'ovvia evoluzione delle pagine web tradizionali basate su HTML. All'utente risulta familiare la struttura a menù/sottomenù ed i contenuti sono memorizzati in un database, separando contenuto e presentazione. Non consentono di andare oltre la semplice visualizzazione del contenuto a meno di non addentrarsi in modifiche al codice sorgente (con presupposti quindi di conoscenze di programmazione e di basi di dati).

Module-based systems - I sistemi basati su moduli si focalizzano sul concetto di funzione o tipo di contenuto. Un tipico esempio è un portale che gestisce news, forum ed event list. Il beneficio di tali sistemi è di avere un ridotto costo di startup. Tuttavia i moduli risultano abbastanza rigidi sul tipo di contenuto e la richiesta di piccole variazioni conduce comunque ad addentrarsi nella programmazione e nello sviluppo richiedendo competenze nel linguaggio specifico utilizzato (es. PHPNuke).

Content object systems: questi sistemi sono centrati sull'idea di contenuto come oggetto - una informazione che può essere spostata ed utilizzata in differenti sezioni o applicazioni del sito. Questo approccio consente un facile riutilizzo delle informazioni con una gestione simile a quella di un database tradizionale, dove ogni informazione viene gestita come un record.

Web Content Management System (WCMS)

Un numero sempre crescente di imprese si rivolge a sistemi di Content Management per la gestione dei propri contenuti informativi su Internet. La scelta del web come mezzo di pubblicazione e gestione delle informazioni è dovuta principalmente alla economicità della soluzione ed alla facilità di accesso ai contenuti indipendentemente dalla collocazione geografica. Le esigenze, le caratteristiche, ed i costi dei sistemi, come precedentemente illustrato, variano enormemente da caso a caso.

Le specifiche su cui si basa la scelta del sistema aziendale per la gestione dei contenuti online dipendono dalla tipologia di azienda e dall'esigenza di rivolgersi o meno all'area pubblica (Internet) o limitarsi alla gestione interna (Intranet). Di seguito vengono riportate alcune tra le esigenze riscontrate nell'analisi di progetti per clienti diversi in applicazioni diverse - da esse ben si evidenzia la necessità di una piattaforma integrata per la gestione

dei contenuti e la loro pubblicazione su web.

- Organizzare l'accesso ai files ed alle informazioni per utenti autorizzati.
- Agevolare le comunicazioni (reparti, sedi remote, distributori ecc).
- Coinvolgere clienti e rivenditori con contenuti a loro riservati.
- Verificare periodicamente la situazione di clienti e fornitori attraverso dati gestionali e descrittivi.
- Aggiornare autonomamente il proprio sito con strumenti semplici e completi.
- Informare su nuovi prodotti/servizi con contenuti specifici per utenti autorizzati.
- Distribuire manuali e depliant alla propria rete tecnica e/o commerciale.
- Presentare efficacemente i prodotti (varianti e ricambi) con schede personalizzabili.
- Raccogliere ordini da rivenditori, agenti, clienti in modalità B2B e/o B2C.
- Gestire clienti ed agenti con modalità di presentazione e condizioni commerciali differenziate.
- Collegare programmi gestionali aggiornando articoli, prezzi, clienti e ricevendo ordini.
- Collegare direttamente palmari in modo semplice e immediato.

Queste necessità trovano la loro naturale risposta nei sistemi di Web Content Management.

Vantaggi dei WCMS

Ogni tipo di CMS fornisce benefici di diversa natura. In particolare, un Web Content Management System offre i seguenti vantaggi:

- Un WCMS pubblica le informazioni online garantendone l'aggiornamento, la coerenza e la consistenza. L'utilizzatore ha la certezza attraverso il sistema di regole della pubblicazione di informazioni corrette, l'utente finale percepisce la disponibilità di tali informazioni, aggiornate, come un miglior servizio. Esiste una connessione diretta tra la frequenza di aggiornamento delle informazioni di una pagina web ed il tempo che intercorre tra due visite successive dello stesso utente.
- Un WCMS facilita il riutilizzo dei contenuti su pagine e siti diversi, aumentandone il valore.
- Un WCMS assicura un aumento di produttività e di soddisfazione del web team, che può concentrarsi sulle tecnologie e sulle funzionalità. Questo riduce i costi di sviluppo ed

■ Premessa

Tabella comparativa tra le caratteristiche di alcuni tra i principali Web Content Management System con licenza commerciale e open source.

Purpose	WebProfessional	PHP Nuke	EZPub	Plone	RedDot	APC AA
Create complete portal site / large web site	Yes	Yes	Yes	Yes	Yes	Yes
Add dynamic sections to existing site	Yes	No	No	No	No	Yes
Provide content management for many sites with one software install	Yes	No	No	No	No	Yes
Content creation	WebProfessional	PHP Nuke	EZPub	Plone	RedDot	APC AA
Content creation based within:						
Browser interface	Yes	Yes	Yes	Yes	Yes	Yes
Offline client	No	No	Yes	No	No	Soon
Windows client	No	No	Yes	No	Yes	No
Can post HTML content to forms	Yes	Yes	Yes	Yes	Yes	Yes
Ability to use categories to organize content	Yes	Limited	Yes	Limited	No	Yes
Custom template design	Yes	Yes	Yes	Yes	Yes	Yes
Difficulty of template customization	Simple	Moderate	Advanced	Advanced	Moderate	Simple
Multibyte language support for international character sets	Yes	Yes	Yes	Unknown	Unknown	Yes
Pre-existing templates	WebProfessional	PHP Nuke	EZPub	Plone	RedDot	APC AA
Articles	Yes	Yes	Yes	Yes	Yes	Yes
Events calendar	Yes	Yes	Yes	?	n/a	Yes
Resource list / links list	Yes	Yes	Yes	?	n/a	Yes
User polls	No	Yes	Yes	?	n/a	No
Discussion forums	Yes	Yes	Yes	?	n/a	No
E-mail newsletter	Yes	No	Yes	?	n/a	Yes
Comment on stories	No	Yes	No	?	n/a	Yes
Content sharing	WebProfessional	PHP Nuke	EZPub	Plone	RedDot	APC AA
Complex sharing between sites	Yes	No	?	?	With extra "Content Import Engine"	Yes (only on APC ActionApps sites)
Category mapping for shared content	No	No	No	No	No	Yes
Field mapping for shared content	No	No	No	No	No	Yes
Headline sharing (RSS 0.9)	No	Yes	Yes	Unknown	Unknown	Yes
Import and export content with other database sources	Yes	?	?	?	?	Yes

Workflow	WebProfessional	PHP Nuke	EZPub	Plone	RedDot	APC AA
Create multiple groups of users with designated rights	Yes	No	Yes	Yes	Yes	Yes
Basic approval process	Yes	Yes	Unknown	Yes	Yes	Yes
Complex approval workflow (multiple approvals / revision tracking)	Yes	No	No	With custom Zope code	Yes	No
Roles-based authentication	Yes	Yes	Yes	Yes	Yes	Yes
Admin and setup	WebProfessional	PHP Nuke	EZPub	Plone	RedDot	APC AA
Operating systems supported	Linux	Linux /	Linux /	Various	Windows 2000/XP	Linux /
	Windows 2000/XP	Unix	Unix			Unix
Web servers supported	Apache / IIS	Apache	Apache	Zope	IIS	Apache
Databases natively supported	MySQL as app. db. Any SQL database (DBTools)	MySQL, Postgres, others	Any SQL database	Zope or SQL	ODBC databases	MySQL
Browser-based site administration and set up	Yes	Yes	Yes	Yes	Yes	Yes
Create new 'applications' or databases on the fly	Yes	No	No	No	Yes (templates)	Yes
Other	WebProfessional	PHP Nuke	EZPub	Plone	RedDot	APC AA
Software cost	Commercial	Free	Free / commercial version available	Free	Commercial	Free for software / end user ASP pricing varies
Estimated implementation time	Single App from ASP - 3 hours / full portal - 2 days	1 week	3 weeks or more	1 week	2 weeks	Single App from ASP - 3 hours / full portal - 2 weeks
Licensing	Proprietary	GPL	GPL	GPL	Proprietary	GPL

Nella tabella sono presenti sei tra i maggiori sistemi di content management per la pubblicazione dei contenuti su web. La piattaforma WebProfessional abbina la semplicità d'uso ed i ridotti tempi di startup alla completezza delle opzioni offerte. Il supporto tecnico on site e telefonico sono un'altra caratteristica che fa di WP una delle prime scelte per le utenze aziendali.

aumenta la competitività del sito diminuendo i tempi di risposta ai cambiamenti rispetto ai siti concorrenti.

- Un WCMS consente la creazione decentralizzata dei contenuti, ponendo una soluzione al collo di bottiglia dell'IT. La velocità del mercato ed i sempre nuovi contenuti vengono gestiti direttamente degli editori/creatori, che si assumono la responsabilità dei contenuti pubblicati.
- Un WCMS facilita la gestione dei contenuti pubblicati attraverso il sistema di regole e validazioni, mantenendo un controllo centralizzato sulla creazione decentralizzata delle informazioni.
- Un sito dall'aspetto professionale, dinamico e con contenuti aggiornati crea l'immagine di una azienda all'avanguardia che investe in sviluppo per la soddisfazione del cliente.

Le impellenti esigenze aziendali di gestione dei contenuti e la crescente disponibilità di infrastrutture dedicate alla connettività hanno reso i WCMS strategici per lo sviluppo delle imprese. Le implementazioni di WCMS proposte dai principali produttori di software e dalla Open Source Community sono diverse ed il mercato attuale vede un proliferare di soluzioni e proposte differenti.

CP Software e l'approccio al WCMS - WebProfessional

CP Software, azienda di servizi e sviluppo software, nasce a Nonantola (Modena) nel 1996 dall'iniziativa di Claudio Pioli, coadiuvato da un gruppo di programmatori specializzati nello sviluppo di soluzioni gestionali per piccole e medie imprese in ambienti Windows e As/400. Dal 1996 al 2000 l'attività dell'azienda subisce una profonda evoluzione, e dalla fornitura di software personalizzato, soprattutto correlato alla gestione della produzione, comincia ad orientarsi verso l'erogazione di servizi internet avanzati.

Nel 1998 viene adottata la piattaforma Cold Fusion (per Windows), un linguaggio RAD (Rapid Application Development) sviluppato da Allaire (rilevata poi nel 2003 da MacroMedia) basato su TAG (molto simile quindi formalmente all'HTML).



Claudio Pioli, imprenditore Modenese e fondatore di CP Software.

La scelta è dovuta principalmente alla facilità di integrazione della nuova tecnologia nell'azienda (la curva di apprendimento è estremamente ridotta e sviluppatori HTML possono lavorare facilmente su codice CFML) ed alla possibilità di rendere operative applicazioni in modo rapido e semplice.

Le prime applicazioni Cold Fusion sviluppate da CP Software utilizzano Microsoft Access come base dati per i progetti di piccole dimensioni e Microsoft SQL Server per i progetti di medie e grandi dimensioni. CP Software in questa fase è ancora fortemente legata al mondo Microsoft, sia per il retaggio derivato dall'ambiente gestionale sia per la piattaforma Cold Fusion, che ancora non presentava versioni per altri sistemi operativi sufficientemente stabili.

Sull'analisi delle richieste del mercato e le forti pressioni della piccola e media impresa per una piattaforma di WCMS che permettesse una completa e strutturata pubblicazione dei dati unita alla semplicità di utilizzo nasce presso CP Software l'idea di standardizzare gli applicativi web esistenti in strutture definite e modulari, per ridurre i tempi di sviluppo e riutilizzare il codice prodotto. Nascono così le prime funzioni e procedure che diventeranno parte integrante della piattaforma Webprofessional.

L'idea iniziale

La necessità di riutilizzo del codice conduce inizialmente alla creazione di strutture standard in HTML da utilizzare quale scheletro delle pagine e all'utilizzo della tecnologia FuseAction, che tende a separare per quanto possibile la parte di codice CFML dalla struttura della pagina ed a scomporre le funzionalità del sito in blocchi standard, separati e riutilizzabili gestiti da parametri passati con metodo GET o POST.

L'esperienza della standardizzazione delle strutture non produce però i benefici desiderati (completa riusabilità del codice), in quanto le nuove soluzioni implementate di volta in volta contengono moduli aggiornati non pienamente compatibili con l'installato. Inoltre la replica del codice rende difficoltoso il mantenimento delle versioni e delle modifiche realizzate ad hoc su specifiche del singolo cliente. Quanto sopra non si riscontra invece nell'applicazione di commercio elettronico che, fornita in modalità ASP, risultava facilmente manutenibile ed aggiornabile, presentando tuttavia una certa rigidità nell'implementazione di funzionalità specifiche personalizzate.

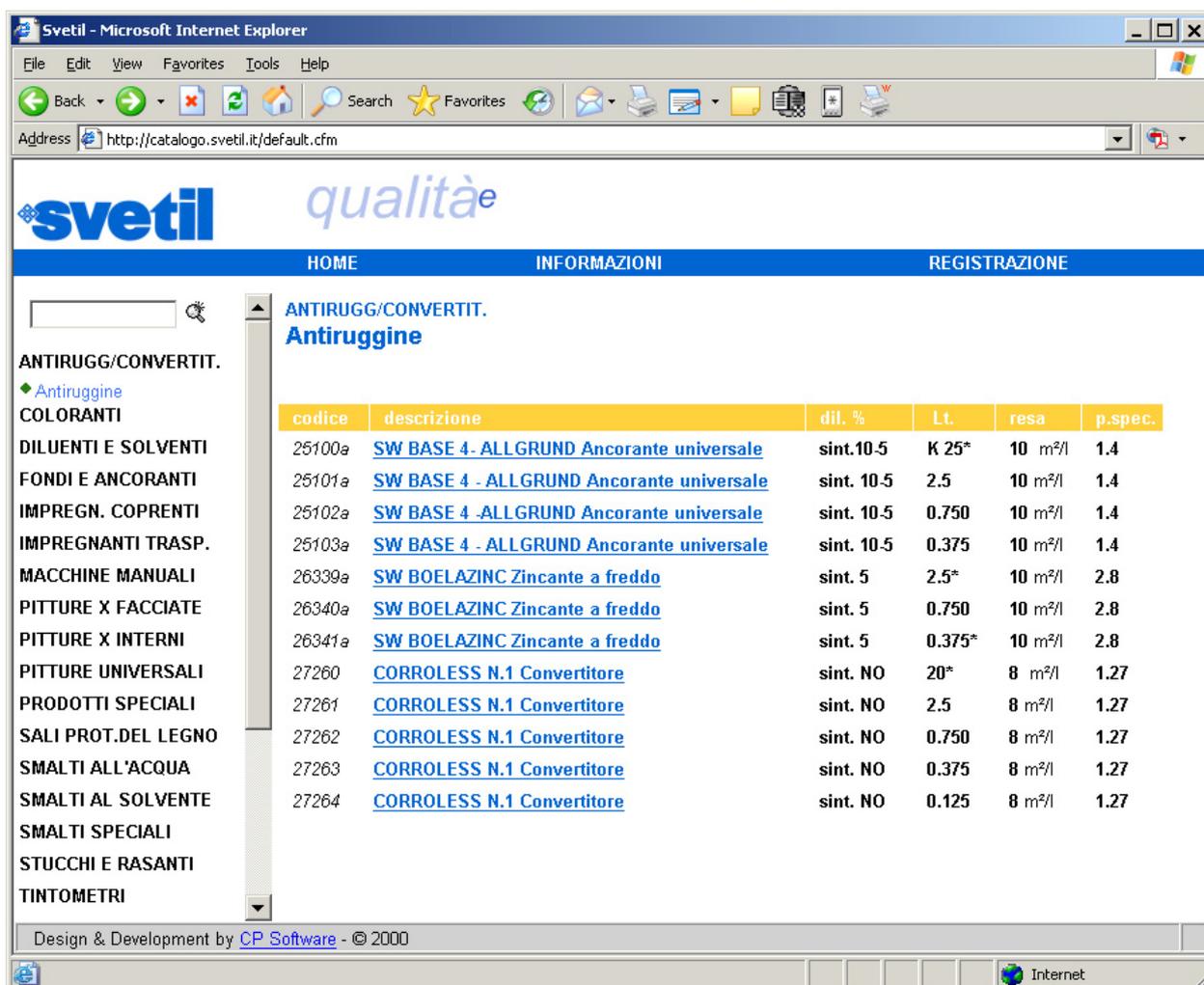
Era evidente che l'erogazione di un servizio con codice sorgente in comune e non replicato presentava notevoli vantaggi, sia per CP Software in termini di sviluppo e manutenzione, sia per i clienti che beneficiavano dei costanti aggiornamenti e delle nuove funzionalità implementate ad ogni release.

Il centro di questa nuova visione strategica è rappresentato dai due moduli eBusiness,

una soluzione completa di commercio elettronico integrata con il gestionale Business, e WebPower, un tool che permette di creare, gestire e mantenere siti internet professionali direttamente da interfaccia web. Entrambi basati su linguaggio di programmazione Cold Fusion e database MS Access, questi due prodotti rappresentano il nucleo primo della piattaforma WebProfessional.

eBusiness

L'esigenza di molti clienti di poter gestire un catalogo online o di vendere i propri prodotti nell'ottica B2B porta alla prima versione del prodotto di commercio elettronico di CP Software, eBusiness. La specifica principale di utilizzare la medesima base dati del gestionale Business, sviluppato da NTS Informatica e commercializzato da CP Software, si traduce



The screenshot shows a web browser window displaying the Svetil website. The page features a navigation menu with 'HOME', 'INFORMAZIONI', and 'REGISTRAZIONE'. A search bar is present, and a sidebar lists various product categories. The main content area displays a table of 'Antiruggine' products with columns for code, description, dilution percentage, weight, coverage, and price per square meter.

codice	descrizione	dil. %	Lt.	resa	p.spec.
25100a	SW BASE 4 - ALLGRUND Ancorante universale	sint.10-5	K 25*	10 m ² /l	1.4
25101a	SW BASE 4 - ALLGRUND Ancorante universale	sint. 10-5	2.5	10 m ² /l	1.4
25102a	SW BASE 4 - ALLGRUND Ancorante universale	sint. 10-5	0.750	10 m ² /l	1.4
25103a	SW BASE 4 - ALLGRUND Ancorante universale	sint. 10-5	0.375	10 m ² /l	1.4
26339a	SW BOELAZINC Zincante a freddo	sint. 5	2.5*	10 m ² /l	2.8
26340a	SW BOELAZINC Zincante a freddo	sint. 5	0.750	10 m ² /l	2.8
26341a	SW BOELAZINC Zincante a freddo	sint. 5	0.375*	10 m ² /l	2.8
27260	CORROLESS N.1 Convertitore	sint. NO	20*	8 m ² /l	1.27
27261	CORROLESS N.1 Convertitore	sint. NO	2.5	8 m ² /l	1.27
27262	CORROLESS N.1 Convertitore	sint. NO	0.750	8 m ² /l	1.27
27263	CORROLESS N.1 Convertitore	sint. NO	0.375	8 m ² /l	1.27
27264	CORROLESS N.1 Convertitore	sint. NO	0.125	8 m ² /l	1.27

La prima versione di eBusiness, scritta in Cold Fusion e basata su database MS Access.

nell'adozione di MS Access come database. Ogni utilizzatore gestisce in modo autonomo il proprio file .mdb mentre un database generale memorizza le impostazioni locali dell'utente per l'applicativo. Il servizio si basa su virtual host in Internet Information Server con identificazione dell'utilizzatore basato sul nome host ed accesso ai dati via ODBC.

L'utilizzo del database di Business consente di poter essere facilmente proposto ed integrato presso il parco clienti esistente. La gestione listini e sconti di Business viene riprodotta per intero, mentre mancano nella gestione dei prodotti le funzionalità relative alle varianti ed alla gestione di kit e distinta base, considerate secondarie nello sviluppo iniziale. L'aggiornamento del catalogo e delle immagini avviene tramite il caricamento via FTP del database Access di Business, contenente le sole tabelle necessarie per ridurre i tempi di trasferimento - la procedura di creazione del database per l'upload viene eseguita sul server dell'utente. Per tutta la durata dell'upload l'applicativo eBusiness è indisponibile alla visualizzazione relativamente al virtual host dell'utente, il che costringe ad upload schedulati solo durante le ore notturne quando gli accessi al sito sono scarsi o nulli.

WebPower

WebPower è il primo prodotto nato dalla collaborazione tra CP Software ed Expertsoft (oggi I-Technology), azienda di sviluppo software creata da Claudio Pioli in Iasi, Romania. WebPower è un applicativo per la gestione dei contenuti di un sito internet utilizzando strumenti che non richiedono competenze di programmazione, consentendo la manutenzione del sito da parte dell'utente finale stesso.

Le specifiche iniziali prevedevano la gestione dei contenuti delle pagine, degli utenti, la presenza di una sezione per l'upload/download dei file ed un modulo per la gestione delle news. Il servizio veniva erogato in modalità ASP da server Windows (come eBusiness) con applicativo scritto in Cold Fusion e basato su database MS Access, scelta dovuta alle politiche aziendali di sviluppo all'epoca dell'analisi del progetto ed alla preesistenza del modulo eBusiness che già adottava il linguaggio di programmazione ed il database citati.

La piattaforma WebProfessional

WebProfessional, nella sua prima versione, nasce dall'integrazione dei moduli WebPower ed eBusiness in una unica struttura per consentire una gestione centralizzata degli utenti e dell'accesso alle informazioni.

Dopo una valutazione delle modifiche necessarie all'unificazione dei due applicativi esistenti in un'unica piattaforma si ritenne conveniente riscrivere parzialmente o completamente il



WebPower, nella versione Cold Fusion e database MS Access per mantenere compatibilità nelle tecnologie utilizzate con il preesistente eBusiness.

codice e rivedere in modo approfondito la struttura della base di dati, alla luce delle considerazioni ed esperienze maturate nei mesi precedenti.

La nuova piattaforma avrebbe avuto un nuovo database completamente ridisegnato per venire incontro alle richieste di un controllo articolato degli accessi ai contenuti, una struttura prodotti/listini/sconti compatibile non solo con Business ma con la maggioranza dei gestionali presenti sul mercato, la presentazione in lingue diverse, ed una serie di altre funzioni basate sulle richieste e segnalazioni raccolte dal supporto tecnico di CP Software.

Questo rimetteva quindi in discussione anche le tecnologie su cui erano basati eBusiness e WebPower, avviando la valutazione di un eventuale linguaggio di programmazione e di una base di dati che meglio rispondessero alle nuove esigenze.

Scelte Tecnologiche

La scelta di CP Software di sviluppare un sistema di Web Content Management è dettata da diverse considerazioni di carattere operativo e di valutazione del mercato. Le applicazioni web si pongono come valida alternativa alle tradizionali applicazioni client/server con installazione locale per vari motivi:

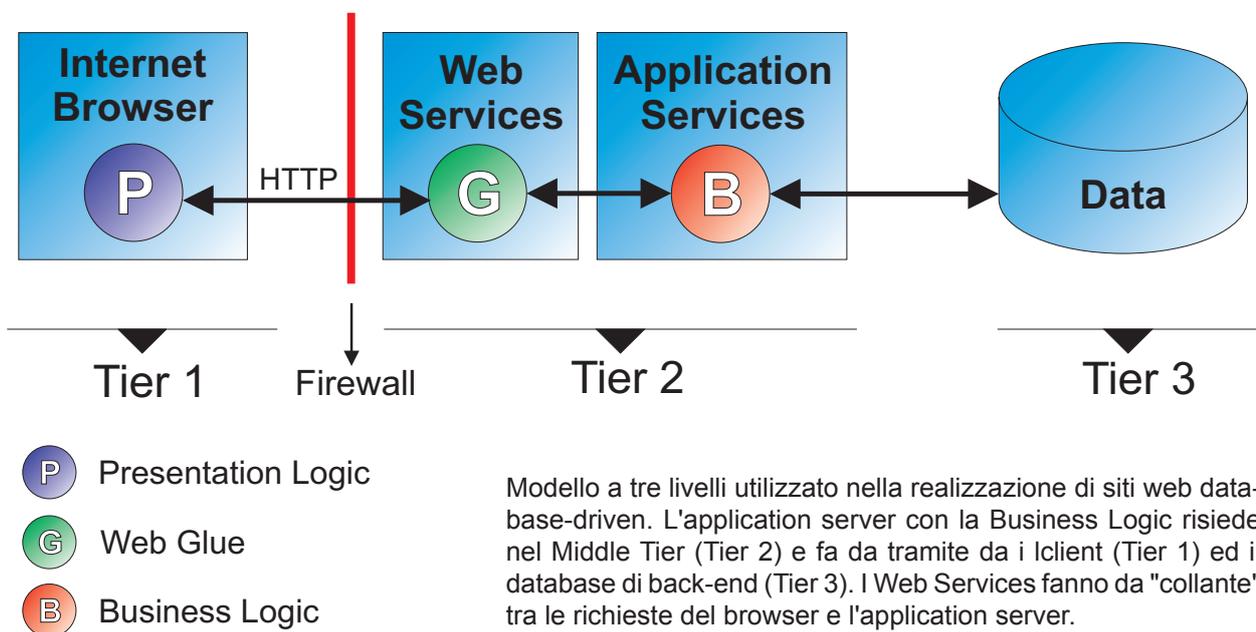
- **facilità di distribuzione e aggiornamento:** un'applicazione web è installata interamente sul server, per cui la pubblicazione coincide con la distribuzione e l'aggiornamento dei client - le informazioni aggiornate sul server sono automaticamente rese disponibili a tutti gli utenti in tempo reale
- **accesso multiplatforma:** l'accesso all'applicazione è indipendente dall'hardware e dal sistema operativo utilizzato dagli utenti
- **riduzione del costo di gestione:** l'uso di Internet come infrastruttura per un'applicazione web riduce notevolmente sia i costi di connettività che i costi di gestione dei client
- **scalabilità:** un'applicazione web ben progettata può crescere insieme alle esigenze dell'azienda senza particolari problemi

Si tratta quindi di realizzare un WCMS che incontri le esigenze delle piccole e medie imprese e che presenti caratteristiche di scalabilità, modularità e facilità di integrazione con le basi dati aziendali. L'architettura che più si adatta a modellare una applicazione con questi requisiti è una architettura a tre livelli (three-tier model), che descrive la pubblicazione delle pagine, la base di dati ed i processi di comunicazione tra webserver e database di back-end su tre livelli distinti ed interconnessi.

Architettura a tre livelli

Le architetture a tre livelli consentono di tenere la presentazione, la logica di business e i dati delle applicazioni chiaramente separati e funzionanti anche su macchine differenti connesse in rete. Ciò che rende così importante la tecnologia a tre livelli in ambiente web è la possibilità, per le applicazioni, di utilizzare interfacce utente di tipo browser con cui accedere alla logica di business, e per i dati di essere dislocati in tutta sicurezza oltre firewall aziendali. I popolari approcci a due livelli non si sono rivelati soddisfacenti per il web, perché espongono dati aziendali e logica di applicazione alla manomissione dell'utente.

Ad accelerare il trend verso l'architettura a tre livelli è soprattutto l'emergere di potenti software che agiscono da "collante" verso il web. Grazie ad essi il mondo HTML può congiungersi con ambienti back-end che supportano applicazioni di classe enterprise e possiedono caratteristiche quali supporto transazionale, security enforcement e interfacce di gestione dei sistemi. Ancora più interessante, le aziende scoprono che quando hanno adottato l'approccio a tre livelli per le applicazioni che lavorano sul web, raggiungono obiettivi basilari quali maggiore scalabilità, sicurezza e riusabilità del codice, persino per le applicazioni che possiedono interfacce client su LAN convenzionali.



Interazione dell'utente con il database

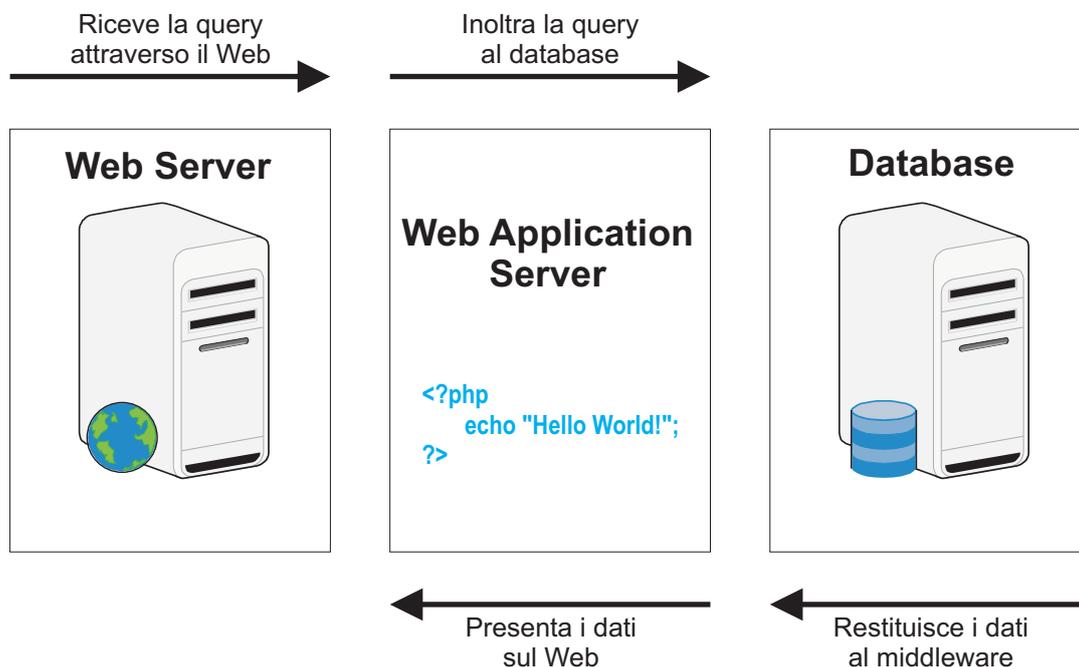
L'interfaccia abituale tra il sistema e l'utente è il browser (Internet Explorer, Firefox, Opera). Attraverso di esso l'utente interagisce con i dati presenti nel database attraverso l'architettura a tre livelli.

La sequenza che porta alla visualizzazione delle informazioni da parte dell'utente a seguito di una interazione con il sistema può essere semplificata nell'elenco a seguire.

1. l'utente invia una richiesta al Web Server attraverso il browser
2. il Web Server passa la richiesta al Web Application Server (middleware)
3. il middleware traduce le richieste dell'utente in interrogazioni SQL e le invia al database di back-end
4. i dati vengono restituiti dal database al middleware

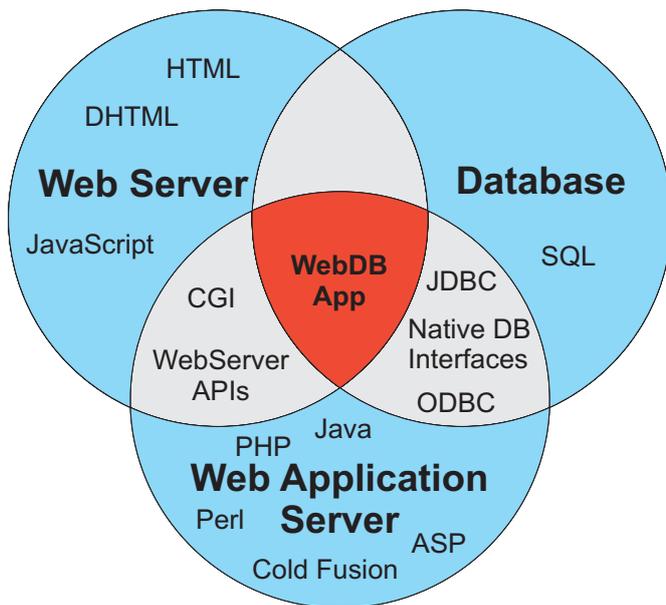
5. il middleware genera una pagina in base ai dati ricevuti
6. il Web Server restituisce la pagina al browser
7. il browser visualizza la pagina all'utente

Uno schema dell'andamento dei flussi attraverso i tre livelli del modello può essere presentato come segue:



La connettività fornita dal middleware tra il Web Server ed il database di back-end può essere realizzata utilizzando differenti tecnologie:

- **CGI scripting** - Una applicazione scritta in uno dei linguaggi di programmazione tradizionali (quali C+, Python, Perl, Java,...) il cui accesso avviene attraverso Common Gateway Interface (CGI).
- **Application server** - ColdFusion o ASP, che lavorano con il Web Server attraverso la sua Application Programming Interface (API) agendo come estensioni del Web Server che forniscono supporto per la connettività al database attraverso ODBC o altre interfacce.



Tecnologie impiegate nelle comunicazioni tra i livelli. Il Web Application Server (middleware) fa da tramite tra il WebServer ed il Database di back end, utilizzando un linguaggio di programmazione (PHP, Cold Fusion, Perl,...), attraverso CGI o API verso il Web Server (Apache, IIS, Tomcat,...) e connettendosi al database con interfacce native, ODBC o JDBC.

Non sempre i livelli logici di un'applicazione Web corrispondono a locazioni fisiche sulla rete. Si va dal caso in cui tutti e tre i livelli risiedono sulla stessa macchina a varie altre distribuzioni fino alla corrispondenza di ciascun livello con una macchina fisica.

- **Integrated Application Server** - Quale ad esempio Oracle9i Application Server che si integra con database Oracle per costituire una piattaforma per la realizzazione di applicativi su internet.

Sostanzialmente il middleware, indipendentemente dalle tecnologie utilizzate, è lo strumento di comunicazione tra i client ed il database.

Prima versione - Cold Fusion e MS Access

La prima versione di WebPower ed eBusiness, ancora prodotti indipendenti, venne sviluppata in Cold Fusion su database Access. La scelta delle tecnologie da impiegare non fu oggetto di analisi in quanto le dimensioni dei due progetti separati non erano tali da giustificare l'acquisizione di nuove tecnologie da parte di CP Software, che già poteva contare su una base consolidata di sviluppatori in questo linguaggio. L'ambiente Cold Fusion e l'utilizzo di MS Access costituivano un vincolo progettuale per WebPower ed eBusiness, nella loro prima versione. Anche il Web Server risultava definito in quanto lo sviluppo e la produzione si basavano unicamente su server Windows, con IIS (Internet Information Server) come web server. La parte di analisi e di progetto era quindi limitata alle funzionalità delle applicazioni senza potersi estendere anche alle tecnologie impiegate.

Cold Fusion

Cold Fusion, sviluppato da Allaire, consente il rapido sviluppo di applicazioni web con una sintassi molto simile all'HTML. E' una piattaforma completa, costituita da un linguaggio di programmazione, un ambiente di sviluppo ed un framework per l'erogazione dei servizi. Cold Fusion usa un linguaggio basato su TAG server side (Cold Fusion Markup Language, CFML) che usato in combinazione con HTML risulta estremamente accessibile ai programmatori web. Supporta più di 70 TAG server side, 200 funzioni ed è possibile avvalersi di oltre 800 componenti di terze parti. La ridottissima curva di apprendimento lo rende estremamente interessante per le piccole software house, non richiedendo grossi investimenti in formazione ed ottimizzando la produttività dei programmatori.

Cold Fusion application server lavora come servizio di sistema multi thread con il server HTTP processando le richieste alle pagine. Quando viene richiesta una pagina Cold Fusion (con estensione .cfm) i TAG CFML sono processati dall'application server, e successivamente la pagina HTML viene generata dinamicamente e servita al browser.

La connessione a database avviene attraverso ODBC, supportando anche piccoli database relazionali come MS Access, FoxPro, Paradox, e fonti dati quali file di testo e di Excel, con un singolo TAG in cui sono definiti i parametri per la connessione. Il linguaggio SQL viene utilizzato per le interrogazioni ai dati ed i risultati vengono restituiti sempre utilizzando un singolo TAG, attraverso il quale è possibile presentare i risultati formattati in tabelle HTML. Il poter utilizzare dati provenienti non solo da database relazionali veri e propri rendeva semplice realizzare applicazioni basate su dati già presenti e gestiti nelle aziende committenti nei formati più disparati (file di testo generati da altre applicazioni su sistemi operativi differenti, quali AS400, file di Excel mantenuti da personale interno, ecc. ecc.).

I limiti di Cold Fusion

Cold Fusion nasce per sistema operativo MS Windows. Pur esistendo porting per Linux ed Unix, la piattaforma non garantisce le necessarie prestazioni ed affidabilità (ben noto il problema dei core dumped in Linux). Inoltre, anche su server Windows, risulta particolarmente esigente quanto a risorse, con prestazioni ridotte in caso di script complessi ed alti carichi di lavoro che impattano su memoria e CPU. Questo esclude Cold Fusion per lo sviluppo di un applicativo pensato per essere erogato come servizio in modalità ASP con esigenza di installazione su piattaforme diverse.

Inoltre Cold Fusion presenta alti costi di licenza, penalizzando di fatto commercialmente l'applicativo nel caso l'azienda cliente opti per un server proprietario in sostituzione del servizio erogato in modalità ASP.

Per le prospettive di sviluppo va valutata attentamente l'opportunità o meno di legarsi a software proprietario di Macromedia e va considerata la disponibilità di programmatori qualificati per il linguaggio CFML, scelte che limitano fortemente le possibilità di sviluppo future di un progetto.

I limiti di MS Access

MS Access si presenta come database desktop per uso individuale o per piccoli gruppi di lavoro in ambiente Windows. La scelta di adottare questo formato dati nella prima versione di eBusiness era dovuta alla necessità di garantire la completa compatibilità con il gestionale di CP Software, utilizzando di fatto la stessa struttura dati relativamente alle tabelle di interesse per la pubblicazione di articoli, sconti, listini, ordini ed utenti. Inoltre erano facilmente implementabili le procedure per l'esportazione dei dati necessari ed il successivo trasferimento via FTP del file .mdb generato. Il limite di 2Gb imposto per i database MS Access inizialmente non era considerato vincolante in quanto la mole di dati gestita per azienda (considerando che il servizio veniva sì erogato in modalità ASP, ma ogni azienda gestiva il proprio database) era di dimensioni ridotte.

Anche la possibilità di utilizzo effettivo solo su piattaforma Windows non costituiva un vincolo in quanto tutto lo sviluppo di CP Software era basato su server Microsoft e linguaggio Cold Fusion.

Tuttavia i limiti della soluzione database adottata si manifestarono già prima della migrazione a PHP/MySQL. I file .mdb si rivelarono inaffidabili soprattutto all'incrementarsi delle dimensioni, pur rimanendo sotto il limite dei 2Gb - spesso risultavano corrotti o con dati inconsistenti. Inoltre il trasferimento dati via FTP implicava uno stop del sito per tutto il periodo di connessione: le procedure di aggiornamento notturno avevano durata e probabilità di errore direttamente proporzionali alla dimensione del file ed alla qualità della linea del cliente (solitamente scarsa). Il passaggio a SQL Server era la logica soluzione, anche se richiedeva una nuova analisi ed il rifacimento delle procedure di aggiornamento dati installate presso il parco clienti. Lato codice il passaggio sarebbe stato praticamente indolore grazie all'utilizzo di ODBC di Cold Fusion per la connessione a database.

Di seguito una breve tabella comparativa tra MS Access e SQL Server che illustra le principali differenze tra i due.

Feature	SQL Server 2000	Access 2000
database size ⁽¹⁾	1.048.516 TB	2 GB plus linked tables size
objects in a database	2147483647	32768

Feature	SQL Server 2000	Access 2000
user name length	128	20
password length	128	14
table name length	128	64
column name length	128	64
index name length	128	64
number of concurrent users ⁽ⁱ⁾	limited by available memory	255
columns per table	1024	255
table size ⁽ⁱ⁾	limited by available storage	1 GB
number of indexes in a table	250	32
number of columns in an index ⁽ⁱ⁾	16	10
bytes per row	8060	2000
number of tables in a query	256	32
columns per SELECT statement	4096	255
nested subqueries	32	50
number of enforced relationships	253	32
SQL statement size	65.536 * Network packet size	approximately 64.000

Segnalate dal simbolo (i) le caratteristiche che avevano maggior peso nella scelta del passaggio a SQL Server. Passaggio che tuttavia venne sospeso già in fase di analisi in quanto i vincoli progettuali imposti dall'integrazione dei due prodotti WebPower ed eBusiness escludevano SQL Server come database di back-end per la piattaforma, a causa soprattutto della possibilità di installazione solo su server Microsoft e dell'alto costo delle licenze.

Le nuove esigenze

Il crescente numero di utenti che utilizzavano entrambi i moduli e richiedevano una gestione permessi ed utenti unificata, oltre che una maggiore integrazione nella grafica, condusse a considerare la riscrittura dei due applicativi in una unica piattaforma modulare che offrisse entrambi i servizi con una gestione combinata di utenti e permessi.

Il pesante lavoro di sviluppo portava anche a riconsiderare le tecnologie impiegate sulla base delle esigenze di mercato. Si richiedeva che la nuova piattaforma fosse:

- portabile su sistemi Windows e Linux
- esente da licenze di terze parti
- performante per consentire l'erogazione del servizio in modalità ASP con hardware limitato

La specifica fondamentale quindi era di poter utilizzare un linguaggio di programmazione ed un database indipendenti dalla piattaforma utilizzata (con target principali MS Windows e Linux) e che fossero Open Source.

La scelta di MySQL

La scelta della base dati su cui appoggiare la piattaforma risultava una scelta critica nello sviluppo. Doveva consentire la portabilità su sistemi operativi diversi (principalmente Linux Suse, Linux Red Hat e MS Windows) e garantire prestazioni accettabili anche su server datati, richiedendo risorse minime di processore, memoria ed occupazione su disco. Inoltre avrebbe dovuto essere Open Source, per limitare i costi di licenze per l'utente finale in caso di server proprietario dedicato mantenendo quindi alta la competitività economica del prodotto sui concorrenti.

I database considerati furono MySQL e PostgreSQL, entrambi nativi Linux ma con porting per ambiente Windows.

L'ingresso dell'Open Source nelle imprese

I database con licenza commerciale (quali Oracle, MS SQL Server, DB2) offrono livelli elevati di funzionalità e sicurezza necessarie alla costituzione di un database aziendale, includendo un database storage avanzato, strumenti per la manipolazione dei dati, replica delle informazioni e strumenti di backup e restore.

Nel corso degli ultimi dieci anni la open source community ha innalzato notevolmente la qualità del software sviluppato, rivolgendosi anche al mondo delle imprese, implementando soluzioni prima appannaggio esclusivo dei prodotti commerciali. Come risultato, le imprese hanno dimostrato interesse a migrare da soluzioni proprietarie commerciali a soluzioni open source. Il crescente interesse per la piattaforma Linux ha condotto sempre più aziende a basare le proprie soluzioni web su server Apache e linguaggio Perl, JSP e PHP, e database PostgreSQL e MySQL (i due principali database su piattaforma Linux).

Cenni su PostgreSQL

PostgreSQL RDBMS deriva dal progetto POSTGRES dell'Università della California a Berkley. Il progetto iniziò nel 1986 a cura del Professor Michael Stonebraker per sostituire

l'ormai datato Ingres. DARPA, la National Science Foundation, l'Army Research Office ed ESL Inc. sponsorizzarono il progetto.

Il database sviluppato, conosciuto come progetto POSTGRES, utilizzava un linguaggio proprietario per le interrogazioni chiamato PostQUEL. Nel 1994 Andrew Yu e Jolly Chen svilupparono un interprete SQL per POSTGRES (Postgres95) che venne integrato nel progetto da Berkley, dando origine all'odierno PostgreSQL.

Cenni su MySQL

MySQL nasce come evoluzione di mSQL, un database utilizzato per gestire piccole quantità di dati con struttura semplice. mSQL risultava poco performante e privo delle funzionalità richieste dagli sviluppatori e dalle applicazioni nelle quali veniva utilizzato, e la open source community decise di riscrivere il database per incontrare le nuove esigenze.

Confronto fra PostgreSQL e MySQL

MySQL e PostgreSQL hanno due diverse metodologie di licensing, e comprenderle risulta importante prima di incorporare uno o l'altro database in progetti di livello enterprise.

MySQL AB, l'azienda che possiede e sviluppa MySQL, ha due differenti tipi di licenze per il proprio database:

- **GNU General Public License** - Se il progetto risulta completamente GPL nella sua distribuzione è possibile utilizzare questo tipo di licenza. L'applicazione deve essere scaricabile in rete ed il codice sorgente disponibile. E' possibile utilizzare questa licenza anche nel caso non si intenda distribuire il progetto ne' internamente ne' esternamente (uso personale).
- **Commercial License** - Un tipico esempio di questa licenza è quando non si intende distribuire il codice sorgente della propria applicazione.

Tuttavia MySQL, nelle versioni precedenti alla 4, conserva il vecchio sistema di licensing (attuale al momento delle specifiche della piattaforma WebProfessional) che consente l'utilizzo del database senza obblighi di licenza in applicazioni commerciali.

PostgreSQL si basa invece su una licenza unica che garantisce l'utilizzo del database fintanto che una copia della Berkeley License è inclusa nell'applicativo. Questo significa che è possibile utilizzare PostgreSQL in applicazioni commerciali senza oneri.

Analisi delle caratteristiche

Verranno ora analizzate le caratteristiche principali dei due database candidati con uno sguardo alle applicazioni commerciali.

Data Storage - MySQL mette a disposizione differenti tipi di data storage: principalmente l'originale ISAM/MyISAM ed i più recente ed avanzati InnoDB e BDB. La scelta di uno o dell'altro dipende dalle caratteristiche necessarie allo sviluppo dell'applicazione. Ad esempio solo InnoDB e BDB supportano le transazioni. PostgreSQL utilizza un unico meccanismo di storage, con supporto delle transazioni.

Database Feature Comparison Table - The database comparison boils down to the main features that each database engine provides.

	PgSQL	MySQL	Commerical
Data Integrity			
ACID compliance	yes	yes	yes
Row-level locking	yes	yes	yes
Partial rollbacks	yes	yes	yes

Advanced Features			
Stored procedures	yes	In 5.0	yes
Views	yes	In 5.0	yes
Triggers	yes	In 5.1	yes
Sequences	yes	5.1?	yes
Cursors	yes	In 5.0	yes
User-defined data types	yes	?	yes

Indexes			
Single column	yes	yes	yes
Multi-column	yes	yes	yes
Primary key	yes	yes	yes
Full text	yes	yes	yes

Replication			
Single-master	yes*	yes	yes
Multi-master	yes**		yes

Interface Methods			
ODBC/JDBC	yes	yes	yes
C/C++, Java	yes	yes	yes

* Open Source, but comes from another vendor

** Solutions exist, but they are commercial

Data Integrity - Una delle principali caratteristiche di un database è la capacità di mantenere l'integrità dei dati. La ACID (Atomic, Consistent, Isolated and Durable) compliance è una qualifica che assicura l'integrità dei dati. ACID essenzialmente garantisce che quando viene eseguita una transazione su un database, o l'intera transazione va a buon fine e viene scritta sul database, oppure il database rimane nello stato precedente la transazione. Sia MySQL che PostgreSQL sono ACID compliant, entrambi supportano rollback parziali di transazioni ed entrambi gestiscono i deadlock. MySQL utilizza un lock tradizionale sulle righe, PostgreSQL si basa invece su un meccanismo chiamato Multi Version Concurrency Control (MVCC) anche se nelle ultime versioni è possibile forzare l'utilizzo del locking a livello di riga.

Caratteristiche avanzate

PostgreSQL offre le principali funzionalità presenti in database di tipo commerciale, quali trigger, viste, stored procedures, cursori, e tipi di dati definiti dall'utente. MySQL presenta alcune di queste caratteristiche sono nella sua ultima versione 5.1.

Triggers e Stored Procedures - Supportate da PostgreSQL, MySQL presenta queste funzionalità solo dalla versione 5 in poi (la più recente). Inoltre PostgreSQL supporta linguaggi differenti nelle stored procedures, quali PL/pgSQL (molto simile a PL/SQL di Oracle), PL/TCL, PL/perl, and PL/python.

Indici - Sia PostgreSQL che MySQL supportano indici single column, multi column, unique e primary key. MySQL ha inoltre supporto nativo per indici full text, mentre PostgreSQL può supportarli solo ricompilando il database con opzioni già incluse nei sorgenti.

Data Type - Entrambi i database supportano i tipi di dati più comuni, ma solo PostgreSQL permette all'utente di definire propri tipi di dati.

Replication - Entrambi i database supportano sistemi di replica con struttura single-master, multi-slave. PostgreSQL mette a disposizione un sistema di replica multi-master multi-slave attraverso un prodotto di terze parti.

Platform Support - Sia MySQL che PostgreSQL supportano piattaforme differenti (Windows, Linux, FreeBSD). MySQL utilizza un modello a thread in cui gli utenti si connettono ad un singolo daemon/servizio sul server. PostgreSQL invece usa un modello non-threaded dove ogni nuova connessione al database chiama un nuovo processo.

Database Interface Methods - Entrambi i database supportano ODBC e JDBC oltre ai metodi nativi proprietari, che supportano tra l'altro SSL per connessioni sicure. E' previsto anche supporto per accesso via C/C++, Java, Perl, Python e PHP.

Backup - Questa sezione è solitamente la parte più carente dei database open source. Sia MySQL che PostgreSQL utilizzano script per facilitare un semplice text dump dei dati e delle strutture. Entrambi forniscono procedure per il backup completo dei database senza interrompere l'erogazione del servizio. Solo pochi prodotti commerciali supportano invece questi due database per il backup online dei dati (NetBackup o Tivoli TSM non supportano MySQL e Postgres). Il disaster recovery è limitato a semplici opzioni di ripristino, con PostgreSQL che fornisce un sistema (Write Ahead Logging) per garantire la consistenza del database (MySQL fornisce funzionalità analoghe solo utilizzando InnoDB).

Dimensions - Postgres e MySQL vengono utilizzati normalmente per database di dimensioni relativamente piccola (sotto i 100Gb). Al di sopra di questa dimensione la percentuale di installazioni cala drasticamente per lasciare spazio a Oracle, DB2 e MS-SQL.

MySQL risultò essere la scelta più adeguata alle necessità. Postgres venne scartato per lo scarso supporto ai sistemi operativi Microsoft (si ricorda che uno dei vincoli progettuali era la possibilità di operare su piattaforme diverse senza pregiudicare le prestazioni). La maggiore velocità utilizzando tabelle ISAM garantiva prestazioni superiori senza pregiudicare il funzionamento dell'applicativo, che non faceva uso di transazioni, trigger e stored procedure a livello di database (alcune parti che richiedevano processi transazionali furono successivamente sviluppate a livello di codice nel middleware tier).

MySQL in dettaglio

Ad oggi MySQL database server risulta il database open source più popolare e diffuso, con una architettura che lo rende di facile installazione ed uso. L'approccio minimalista dei programmatori è risultato in un database estremamente performante, compatto, stabile e di facile installazione. La separazione del core server dallo storage engine rende possibile l'utilizzo di transazioni quando richiesto oppure la rinuncia ad esse in modalità priva di transazioni con accesso a disco estremamente rapido (funzionalità che ben si abbina ad applicazioni web).

Supporto sintassi ANSI SQL. MySQL supporta un subset di ANSI SQL 99, a cui si aggiungono istruzioni proprietarie quali REPLACE o la clausola LIMIT per SELECT e DELETE. Sono inoltre supportate sintassi alternative da altri database per facilitare il porting delle

applicazioni.

Affidabilità e Performance. La diffusione e la disponibilità del codice sorgente permettono alla open source community un lungo ed approfondito testing delle nuove versioni prima che siano disponibili per l'utilizzo in produzione. Questo riduce la presenza di bug e garantisce buone prestazioni.

Installazione e flessibilità. L'estrema facilità di installazione, la disponibilità di documentazione ed il supporto di una vasta community rendono MySQL uno dei database di più rapida e semplice installazione. Inoltre la sua particolare architettura che consente di gestire diverse strutture dati con caratteristiche differenti mette a disposizione delle aziende uno strumento flessibile e capace di seguire diverse linee di sviluppo nella base di dati aziendale.

Supporto Multiplatforma. MySQL è disponibile in forma binaria compilata per più di venti piattaforme, incluse le maggiori distribuzioni di Linux, Mac OS, UNIX e MS Windows. Questo permette di sviluppare applicazioni platform independent basate su MySQL. Attualmente sono disponibili per il download binari ottimizzati per Linux, Microsoft Windows, FreeBSD, Sun Solaris, IBM's AIX, Mac OS X, HP-UX, AIX, QNX, Novell NetWare, SCO OpenUnix, SGI Irix, and Dec OSF.

Supporto dei linguaggi di programmazione. E' possibile connettersi a MySQL da tutte le maggiori piattaforme utilizzando la quasi totalità dei linguaggi di programmazione, utilizzando le client library messe a disposizione da MySQL oppure uno dei prodotti disponibili nella sezione "Connector family of database drivers" del sito.

Indipendenza dello storage engine. L'indipendenza dello storage engine di MySQL permette all'utente di scegliere il tipo di database storage più appropriato per le sue esigenze. Tipicamente si utilizza lo storage engine InnoDB per applicazioni che richiedono locking a livello di riga e supporto delle transazioni, MyISAM nel caso non sia richiesto il supporto delle transazioni. MyISAM garantisce le prestazioni migliori in termini di velocità.

Vasta community di sviluppatori. Grazie alla sua diffusione, MySQL può contare su una community estremamente vasta ed attiva di sviluppatori certificati e competenti. Questo rende la base di dati estremamente appetibile per lo sviluppo di nuove applicazioni.

Ricerca e indicizzazione full-text. L'indicizzazione full-text consente di ricercare in campi

contenenti testo parole o frasi specifiche, con un ranking di rilevanza. Dalla versione 4 sono disponibili le funzionalità di exact match phrase e l'utilizzo di operatori booleani.

Character set handling, con pieno supporto Unicode. Dalla versione 4.1 è possibile specificare un set caratteri a livello di database, tabella o singolo campo, ed è stato migliorato il supporto alla conversione da un set caratteri all'altro. L'utilizzo di Unicode prevede l'encoding UCS2 e UTF8.

La scelta di PHP

L'abbandono della piattaforma Cold Fusion avvenne dopo una attenta valutazione dei requisiti di progetto e dei linguaggi disponibili. I costi del passaggio ad un diverso linguaggio di programmazione erano compensati dal vantaggio (sia in termini di licenze che di potenzialità di mercato e strategie commerciali) di poter installare il prodotto su piattaforma Linux.

Cold Fusion non rispondeva ai requisiti di licensing e di portabilità. Inoltre presentava problemi nell'esecuzione di script complessi e non garantiva la necessaria flessibilità, se non utilizzando estensioni scritte in C o Java - il che comportava comunque l'utilizzo di un secondo linguaggio di programmazione. Non ultimo, tutti i server installati presentavano il problema dell'errore "servizio indisponibile" presentato nei momenti di picco, ben noto agli amministratori di sistema che utilizzano piattaforma Cold Fusion.

L'unico linguaggio effettivamente considerato per la riscrittura del codice fu PHP.

Prime versioni di PHP

La prima versione di PHP fu creata da Rasmus Lerdorf a metà degli anni 90 per eseguire in modo semplice alcune applicazioni web di comune e ricorrente utilizzo. L'obiettivo era di ottenere un linguaggio di scripting che permettesse di produrre risultati con un numero minimo di righe di codice. PHP nasce quindi come linguaggio HTML-centric, cioè PHP nasce per essere utilizzato embedded HTML.

La prima versione diffusa fu chiamata PHP/FI 2.0 (Personal Home Page / Form Interpreter), e nonostante i bug e le inconsistenze si diffuse comunque tra la community degli sviluppatori web, soprattutto per la sua semplicità. Il linguaggio utilizzato precedentemente era il Perl, molto diffuso ed articolato ma difficile da apprendere e soprattutto Perl-centric (l'HTML è

embedded nel codice Perl). Questa prima versione era legata al webserver Apache ed afflitta da problemi di lentezza.

Molti dei problemi vennero risolti nella versione 3, riscritta da Zeev Suraski e Andi Gutmans utilizzando per la compilazione Flex e Bison. Il parser guadagnò in stabilità e velocità. Inoltre fu introdotta la possibilità di estendere il linguaggio attraverso moduli esterni, aggiungendo nuove funzioni. Tuttavia il parser rimaneva legato ad Apache e la velocità, se pur notevolmente migliorata rispetto alla versione precedente, rimaneva lontana dalle prestazioni del Perl. Con la versione 3 veniva inoltre introdotto supporto (se pur limitato) alla programmazione object-oriented. Queste nuove caratteristiche e l'aumentata stabilità ed efficienza incrementarono la diffusione del PHP che in soli 18 mesi passò da circa 250.000 installazioni a 2.500.000.

A metà del 2000 venne rilasciata la versione 4, completamente rivisitata. La nuova versione garantiva la backward compatibility con PHP 3. La più importante novità consisteva nel passaggio allo Zend Engine, prodotto da Zend (fondata da Zeev Suraski e Andi Gutmans - il nome è una contrazione di "ZEEv and aNDi") per supportare il PHP nei corporate environment. L'engine, introdotto nel core del PHP, tiene traccia di tutte le risorse utilizzate dagli script liberandole al termine dell'esecuzione e risolvendo i problemi di memory leaks. La versione 4 inoltre era completamente indipendente dal web server, rendendo PHP disponibile per Apache 1.3.x, Apache 2, Microsoft IIS, Zeus, AOLServer ed altri, raggiungendo anche quel 40% di utenti che non utilizzavano Apache.

Le performance aumentarono drasticamente, in dipendenza di diversi fattori. PHP 3 utilizzava una strategia "execute while interpreting", il che significa che ogni riga di codice veniva letta, interpretata ed eseguita in successione. Questo comportava spesso la rilettura e reinterpretazione del codice in dipendenza dei flussi di programma.

PHP 4 utilizza una strategia "compile first, execute later" - l'intero script viene letto e compilato in byte code prima dell'esecuzione, il che aumenta le prestazioni di circa il 100%. Inoltre è possibile ottimizzare e mantenere in cache il codice compilato, incrementando ulteriormente le performance.

Con PHP 4 viene introdotto il multi-threading, che essenzialmente consente ad alcune funzioni particolarmente lunghe ma non critiche di eseguire indipendentemente dallo script principale, snellendo l'esecuzione.

Attualmente PHP conta più di 10 milioni di installazioni ed è il modulo di Apache più diffuso, superando anche mod_perl.

Relazione con l'HTML

Il PHP concettualmente differisce dal Perl per essere embedded nell'HTML. I blocchi di PHP vengono scritti direttamente nel codice HTML utilizzando i delimitatori `<?php` e `?>`. Nell'esempio seguente è evidente la maggior leggibilità del codice PHP rispetto al Perl:

Perl script	PHP Script
<pre>#!/usr/bin/perl print "<HTML>\n"; print "<BODY>\n"; print "<P>welcome, \$Name</P>\n"; print "</BODY>\n"; print "</HTML>\n";</pre>	<pre><HTML> <BODY> <P>welcome, <?php print \$Name; ?></P> </BODY> </HTML></pre>

Il vantaggio del linguaggio embedded nell'HTML, oltre alla leggibilità, è la possibilità di usare i comuni strumenti di editing (come Dreamweaver o Frontpage) per modificare le pagine in HTML, cosa non possibile in Perl.

Interpretato vs Compilato

PHP utilizza sia modalità interpretata che modalità compilata per offrire ai programmatori il miglior bilanciamento tra flessibilità e prestazioni.

PHP compila lo script in una serie di istruzioni (dette "opcode") ad ogni accesso. Queste istruzioni sono poi eseguite sequenzialmente fino al termine dello script. Le prestazioni sono indubbiamente inferiori ad un linguaggio completamente compilato, ma d'altro canto lascia la possibilità di variare facilmente il codice e consente veloci feedback durante la fase di sviluppo - gli errori vengono rilevati durante la fase di interpretazione e segnalati sull'output. La ricompilazione del codice non incide sulle prestazioni utilizzando acceleratori PHP (sistemi di cache ed ottimizzazione nell'esecuzione dell'opcode).

Essendo il linguaggio interpretato, il managing della memoria è interamente demandato a PHP, che provvede a liberare le risorse impegnate al termine dell'esecuzione di ogni script. Questo significa che l'interprete provvedere a chiudere i database link e a liberare la memoria assegnata, rendendo più stabile il web server.

Lo Zend Encoder

Zend Encoder è il prodotto per la proprietà intellettuale della software house israeliana Zend pensato per cifrare il codice PHP e poter così distribuire applicazioni commerciali senza timore di violazione degli accordi di licenza ed in particolare reverse engineering. Zend Encoder compila e converte gli script PHP in un formato binario indipendente dalla piattaforma conosciuto come "Zend Intermediate Code".

Performance - Lo Zend Encoder non impatta sulle performance degli script. Una pagina scritta in PHP non criptato ha approssimativamente lo stesso tempo di esecuzione di una criptata. Se utilizzato congiuntamente a Zend Cache le performance migliorano decisamente anche rispetto a codice PHP non criptato.

Encryption strenght - L'algoritmo di criptazione, proprietario di Zend, risulta alla luce dei fatti adeguato alle necessità. Se da un lato le strutture OP non vengono modificate, dall'altro la struttura di un algoritmo di decodifica (dalla struttura OP a codice PHP non criptato) porterebbe a codice differente da quello originario ed assolutamente non commentato. Lo sforzo per ottenere il codice in forma leggibile sarebbe maggiore dei benefici ottenuti. Il codice criptato richiede l'installazione di Zend Optimizer (scaricabile gratuitamente dal sito <http://www.zend.com>), la libreria (disponibile per tutte le versioni di PHP e per i maggiori sistemi operativi) che interpreta ed ottimizza l'esecuzione del codice. Zend Encoder risulta essere la scelta migliore (se non obbligata) per proteggere la proprietà intellettuale del codice mantenendo piena compatibilità con PHP.

ColdFusion vs PHP

Le motivazioni della scelta di PHP su Cold Fusion si riassumono nei seguenti punti, dove vengono valutate alcune delle caratteristiche principali dei due linguaggi. Non sempre per ogni punto PHP risulta essere migliore di ColdFusion, ma il peso di quelli più significativi (contrassegnati con *) è indiscutibilmente a favore di PHP.

Platform Support * - Cold Fusion aveva una selezione di piattaforme relativamente limitata (Windows, Solaris, Linux o HP/UX), e le installazioni in ambienti non Microsoft risultavano essere poco stabili e con problematiche irrisolte.

Language * - Cold Fusion è estremamente efficiente per la visualizzazione di dati in pagine web tramite interazione con un database di back-end. Per contro il linguaggio di scripting

risulta essere poco articolato e flessibile soprattutto se comparato a PHP. Cold Fusion rimane una ottima scelta ma per script di non elevata complessità.

Database Support * - Cold Fusion utilizza una astrazione della connessione a database rendendo molto semplice utilizzare diversi database con lo stesso codice. Supporto nativo per le connessioni è disponibile solo nella versione Enterprise ed è limitato ai maggiori database disponibili sul mercato. Il metodo principale di connessione è ODBC, ben supportato e performante in ambiente Windows, non altrettanto su altre piattaforme. Il supporto nativo dei database in PHP è invece completo, con l'utilizzo di sintassi differenti per avvalersi delle peculiarità di ognuno di essi. Questo rende tuttavia la portabilità di una applicazione da un database all'altro estremamente difficile. Dalla versione 4 di PHP è disponibile la possibilità di astrarre dal database, rendendo di fatto la portabilità da un database all'altro pari a quella di Cold Fusion.

File-System Support * - Il supporto al file system è adeguato in Cold Fusion, completo in PHP.

Error-handling - Cold Fusion rende disponibile funzionalità per l'error handling complete attraverso un sistema di try/catch. In PHP la gestione degli errori non è altrettanto sviluppata.

Search Capability - Cold Fusion integra Verity, un efficiente motore di ricerca che supporta sia ricerche su file che su database.

Encryption - PHP utilizza Zend Encoder per la distribuzione di codice in formato binario criptato. Anche Cold Fusion ha una funzione equivalente, ma l'algoritmo di criptazione risulta essere estremamente debole e programmi per la riconversione in chiaro dei file criptati sono facilmente reperibili in rete.

PHP risulta più efficiente per script complessi, più stabile ed utilizza minori risorse di sistema, garantendo la portabilità completa da sistemi Windows a Linux. Inoltre fornisce strumenti adeguati per la protezione da copia del codice che può essere distribuito conservando piena funzionalità e prestazioni.

La scelta di PHP risulta essere la più adeguata in base alle specifiche iniziali ed alle prospettive di sviluppo future (PHP è largamente diffuso e supportato).

Conclusioni sulle specifiche

Dalle precedenti considerazioni le specifiche tecniche per la piattaforma WebProfessional risultano essere:

- Architettura a tre livelli con separazione tra application server, database e web server.
- Linguaggio di sviluppo PHP per garantire la portabilità tra sistemi Windows e Linux e Zend Encoder per la protezione dei sorgenti.
- MySQL come database di back-end, che garantisce le prestazioni e fornisce supporto adeguato per le necessità del progetto lasciando aperta la possibilità di utilizzare funzionalità avanzate (transazioni, lock a livello di riga,...) in un secondo tempo semplicemente modificando i parametri di configurazione del demone o servizio.

Nei prossimi capitoli verrà illustrata la piattaforma realizzata, WebProfessional, nei suoi moduli principali e nelle funzionalità di spicco.

La Piattaforma

WebProfessional è una avanzata soluzione professionale per la gestione online del sito web aziendale, allo scopo di estendere le attività ed i servizi offerti attraverso il web.

L'impostazione grafica e la disposizione dei contenuti avvengono attraverso template grafici e moduli. Una volta scelto il template più appropriato ai contenuti ed alla grafica del sito (template che è possibile cambiare in qualsiasi momento grazie alla completa separazione dei contenuti del sito dal layout e dalla grafica) attivando i diversi moduli di WebPower è possibile gestire informazioni di diversa natura.

Modalità ASP

WebProfessional viene erogato come servizio in modalità ASP. Il server centrale mantiene una unica copia del codice ed ospita tutti i siti attivati. Questo facilita enormemente la manutenzione degli script, gli aggiornamenti ed i bug fixing.

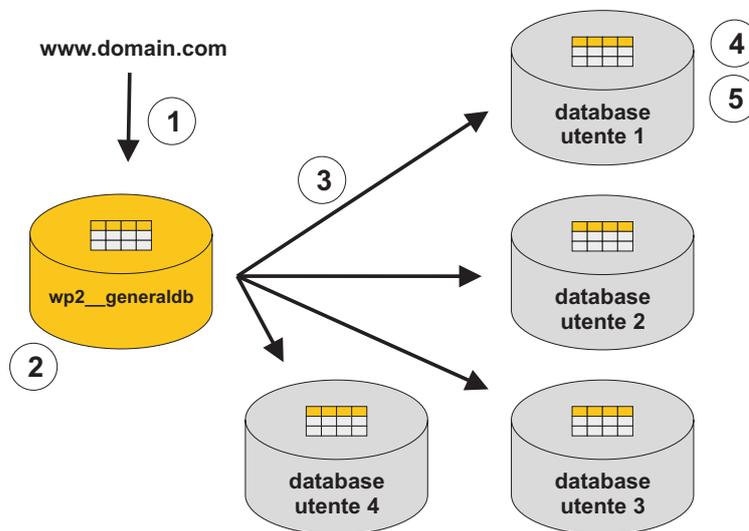
Il riconoscimento dell'utente si basa sul nome host inviato nell'header della chiamata HTTP. Il nome host viene riconosciuto ed una SELECT sulla tabella **tbl_companies** del database **wp2_generaldb** (database di configurazione generale della piattaforma) identifica il record relativo all'host ed i parametri di configurazione, che comprendono:

- nome del database dell'utente
- directory specifica dell'utente
- versione
- moduli attivati
- scadenza del servizio
- codice di verifica licenza

Ogni nome host può avere un numero imprecisato di alias per gestire più nomi di dominio registrati per lo stesso sito. Tutti i dati dell'utente relativi ai vari moduli di WebProfessional sono memorizzati nel database specificato e più precisamente nel record in **wp2_generaldb**.

All'arrivo di una chiamata HTTP/HTTPS il funzionamento del sistema si può schematizzare come segue:

1. Il client (browser) invia la richiesta al server WebProfessional
2. Il web server riceve la richiesta, riconosce il nome host contenuto nell'header HTTP e l'application server esegue una query sul database **wp2_generaldb** per identificare i parametri relativi alla richiesta (la piattaforma si sostituisce alla gestione dei domini virtuali del web server)



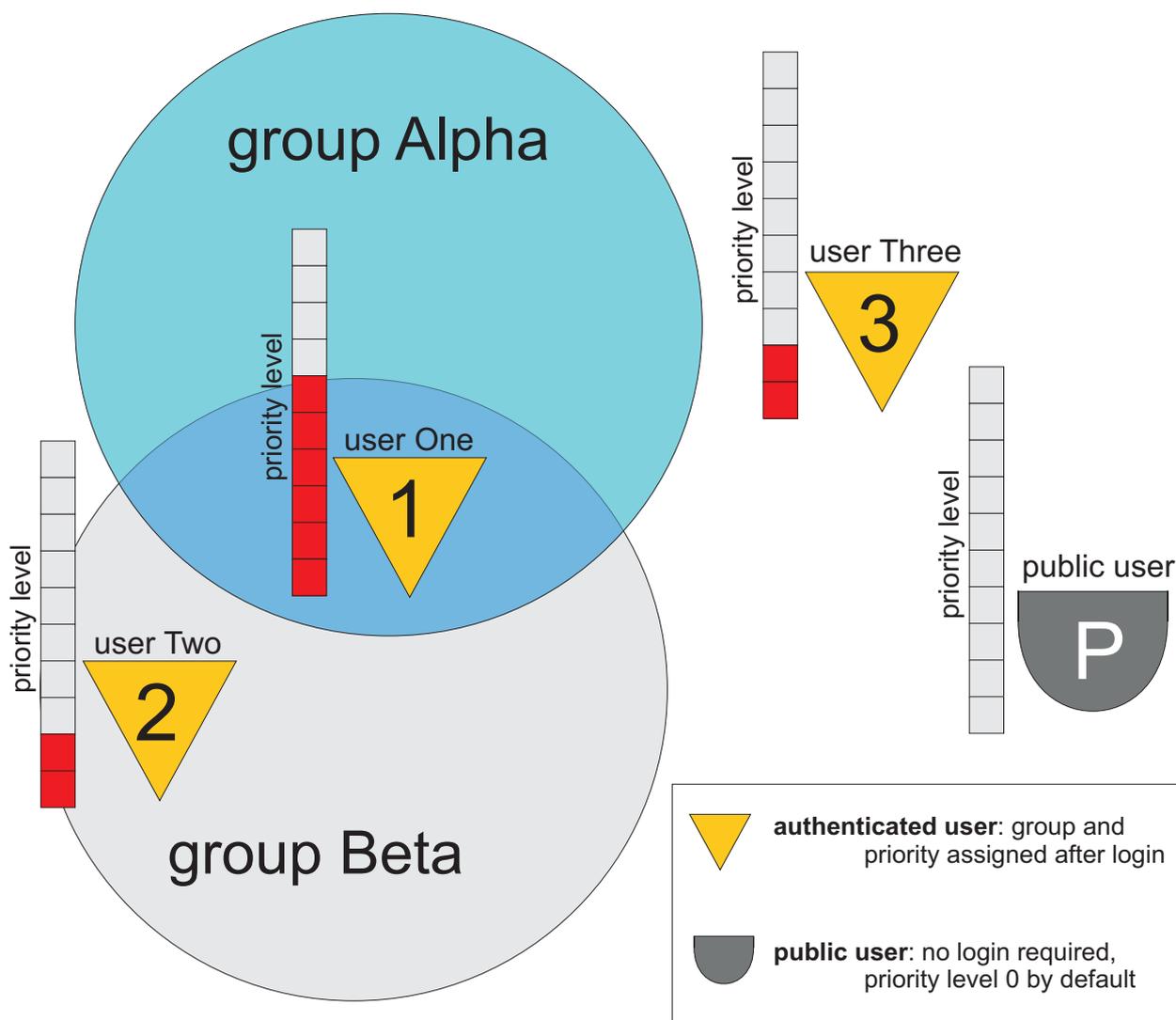
3. L'application server interroga il database specificato nel record di **tbl_companies** in **wp2_generaldb**
4. L'application server riceve le informazioni e le processa creando dinamicamente il codice HTML
5. La pagina (contenente le informazioni) viene inviata al client

Quando la SELECT eseguita su **tbl_companies** non ritorna risultati l'utente viene automaticamente rediretto sul sito specificato nella variabile **\$GENERAL_HOST** del file di configurazione **_app.php**.

Permission integrate nel Content Management

Nella gestione dei contenuti è strategica la possibilità di assegnare la visibilità degli stessi agli utenti attraverso un sistema di regole e permessi. WebProfessional implementa questo sistema utilizzando due distinti metodi di validazione - identità dell'utente (login e gruppo) e livello di priorità.

Il login utente avviene tramite username e password (informazioni inviate via POST su HTTPS). Lo username identifica univocamente l'utente, che può appartenere ad uno o più gruppi. Ad ogni utente è inoltre assegnata una priorità che va da 0 a 10. La visibilità dei contenuti può quindi essere articolata su questi tre parametri (username, gruppo e priorità) in modo più o meno completo in dipendenza del modulo di WebProfessional.



Gestione dei permessi di accesso alle informazioni in WebProfessional

Gli accessi ai contenuti vengono gestiti tramite username, gruppo e priorità secondo la modalità:

([user condition] OR [group condition]) AND [level condition]

Ipotezziamo di avere tre utenti con accesso tramite login ed un utente pubblico, ed una serie di contenuti da visualizzare agli utenti in base all'appartenenza ai gruppi ed ai livelli di priorità.

Contenuto A [users: -; groups: Alpha; priority: 2]:	visibile al solo utente One
Contenuto B [users: Two; groups: Alpha; priority: 2]:	visibile agli utenti One e Two
Contenuto C [users: -; groups: Beta; priority: 4]:	visibile al solo utente One
Contenuto D [users: -; groups: -; priority: 5]:	visibile al solo utente One
Contenuto E [users: -; groups: Alpha; priority: 1]:	visibile agli utenti One, two e Three
Contenuto F [users: -; groups: -; priority: 0]:	visibile a tutti gli utenti, compresi gli utenti pubblici

Tramite la priorità gli utenti appartenenti allo stesso gruppo sono ulteriormente raggruppabili in sottogruppi definiti dalla priorità.

Validazione tramite utente/gruppo

Le condizioni su username e gruppo vanno in condizione OR quando utilizzate entrambe; significa che è possibile assegnare visibilità a determinate informazioni ad uno o più gruppi e a singoli utenti non appartenenti ai gruppi specificati.

Validazione tramite priorità

Le condizioni sul livello di priorità sono invece slegate dal concetto di utenti ed appartenenza ai gruppi, ed attribuiscono semplicemente un livello da 0 a 10 (0 è il livello più basso, 10 il più alto) che consente l'accesso a tutte le risorse con definita una priorità di livello minore o uguale (default 0).

Combinazione dei due metodi

Username/gruppi e livello di priorità sono in AND logico: è quindi possibile assegnando opportunamente le priorità agli utenti di un gruppo suddividerli ulteriormente in diversi livelli con accessi differenziati alle informazioni. I due tipi di condizione possono essere utilizzati separatamente o in combinazione per ogni singola informazione: questo consente estrema flessibilità nella gestione degli accessi e nella visualizzazione dei contenuti.

Gestione Utenti

Gli utenti possono essere gestiti direttamente dall'area di setup della piattaforma oppure dal Primary Domain Controller di un dominio NT o di una foresta di Active Directory (con procedure di importazione e successiva sincronizzazione).

Le strutture dati relative agli utenti si riassumono nelle tabelle **tbl_group** e **tbl_users** descritte di seguito.

Nel progetto iniziale la relazione tra **tbl_group** e **tbl_users** prevedeva una tabella intermedia **tbl_users_group** di associazione tra **tbl_users.u_id** e **tbl_group.code_group**. In tal modo ogni gruppo avrebbe potuto contenere zero o più utenti, ed ad ogni utente si sarebbero potuti associare zero o più gruppi.

La relazione di appartenenza ai gruppi venne invece tradotta in fase di programmazione utilizzando una semplice lista nel campo **tbl_users.u_group** (nelle prime versioni VARCHAR 255), utilizzando il valore di **tbl_group.description**. Il problema si evidenziò alla versione 4.0 quando una delle implementazioni richieste era la possibilità di rinominare i

gruppi di utenti. La modifica, di per sè poco significativa nella struttura di database prevista (si sarebbe trattato semplicemente di aggiornare il campo **tbl_group.description**), risultò invece essere particolarmente onerosa alla luce della reale situazione. Si prospettavano due soluzioni:

- variare la struttura del database e riportarla al progetto originale
- rinominare il gruppo in **tbl_group.description** ed in tutti i campi delle tabelle ove utilizzato (oltre che in **tbl_users**, il campo descrittivo del gruppo viene utilizzato per relazionare anche tutte le altre tabelle di WebProfessional)

La prima risultava troppo onerosa come tempi di sviluppo rapportata alla funzionalità richiesta. La seconda invece venne scartata a causa della complessità della realizzazione e soprattutto valutando l'impatto della modifica alla luce di sviluppi futuri (ogni nuovo modulo che avesse utilizzato permessi e regole a livello di gruppo avrebbe comportato l'aggiornamento della procedura per rinominare i gruppi stessi). Ad oggi non è possibile rinominare i gruppi, e tutt'ora il campo chiave di **tbl_group** risulta essere privo di significato in quanto la chiave reale effettivamente utilizzata nell'applicazione è **tbl_group.description** (con controllo della presenza di duplicati da codice PHP).

tbl_group

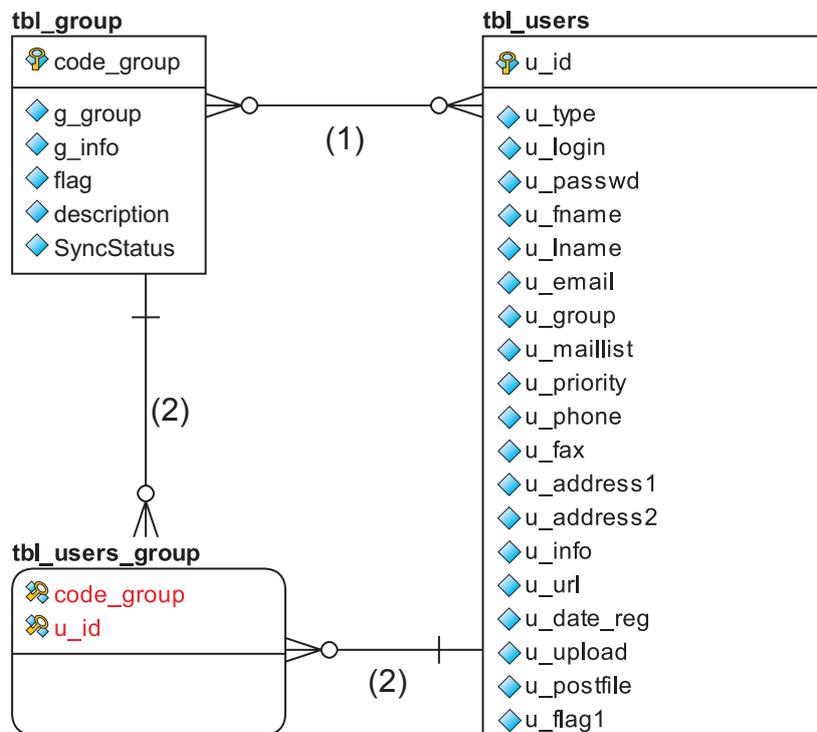
name	description	type	size	key	not null	default	observation
u_id	User identifier	BIGINT	20	yes	yes	0	
u_externalcode	Alphanumeric E-business identifier	VARCHAR	20				Used only if the general setting of the users identifier type is alphanumeric, and replaces in synchronization the u_id field. This option can be enabled only by the administrator of the server!
u_alt	User alternative code	VARCHAR	100				
u_type	User type: {staff, client, agent}	VARCHAR	10			client	Insert one of the values: staff, client, agent (an agent is automatically a staff member)
u_login	User's login (public area)	VARCHAR	20		yes		The login must be unique and is required for authentication to the site
u_passwd	User's password (public area)	VARCHAR	20				Required for entering the authenticated site
u_fname	First name	VARCHAR	255				
u_lname	Last name	VARCHAR	255				
company	Company name	VARCHAR	255				

name	description	type	size	key	not null	default	observation
u_email	Email	VARCHAR	255				
u_phone	Phone number	VARCHAR	30				
u_fax	FAX	VARCHAR	30				
u_url	URL: web address	VARCHAR	100				
pricelist	Price list code	VARCHAR	100		yes	0	Linked to code_pricelist from tbl_pricelist
code_disc_class	Code client discount class	VARCHAR	100		yes	0	Linked to code_disc_class from tbl_disc_class with type='c'
u_vatcateg	Code of the VAT category	VARCHAR	10		yes	0	Linked to vatc_code from tbl_vatcateg
u_codecurrency	Code of the currency	VARCHAR	20		yes	0	Linked to tc_codecurrency from tbl_currency
addr_street	Address-street	VARCHAR	50				
addr_number	Address-street number	VARCHAR	50				
addr_city	Address-city	VARCHAR	50				
addr_zip	Address-zip	VARCHAR	50				
addr_state	Address-county	VARCHAR	50				
addr_country	Address-country	VARCHAR	50				
ship_street	Shipping address-street	VARCHAR	50				

Nello schema sono riportate la situazione attuale (1) e la struttura prevista dall'analisi iniziale (2) per la tabella **tbl_users_group**.

Il legame tra la tabella utenti e la tabella gruppi avviene attraverso la relazione molti a molti (1). Relazione di tipo *non specific* che coinvolge il campo descrittivo **tbl_group.description** e che non permette, a meno di pesanti modifiche al codice o variazioni alla struttura del database (che implicano di conseguenza modifiche al codice), di poter aggiornare il valore di tale campo senza far perdere coerenza alle informazioni contenute nel database.

La soluzione (2) prevedeva una tabella di relazione **tbl_users_group** con due relazioni di tipo uno a molti identifying (con propagazione della chiave primaria). Questo avrebbe consentito da un lato di poter apportare la modifica richiesta (possibilità di rinominare i gruppi utenti) con tempi di sviluppo molto ridotti, dall'altro di aumentare le performance del sistema. Infatti nella struttura attuale l'appartenenza di un utente ad un determinato gruppo avviene tramite una query con LEFT JOIN ed un LIKE nella condizione di join.



name	description	type	size	key	not null	default	observation
ship_number	Shipping address-street number	VARCHAR	50				
ship_city	Shipping address-city	VARCHAR	50				
ship_zip	Shipping address-zip	VARCHAR	50				
ship_state	Shipping address-county	VARCHAR	50				
ship_country	Shipping address-country	VARCHAR	50				
billing_street	Billing address-street	VARCHAR	255				
billing_city	Billing address-city	VARCHAR	50				
billing_zip	Billing address-zip	VARCHAR	50				
billing_state	Billing address-county	VARCHAR	50				
billing_country	Billing address-country	VARCHAR	50				
vat_no	VAT number	VARCHAR	25				
code_fiscale	Code fiscale	VARCHAR	30				
u_nrint	Interior number	VARCHAR	100				
u_flagagw	Internal flag for an agent to not display anymore the popup for choosing the clients	CHAR	1				Reccomandation: do not alter this value!
u_sync_order	Flag automatically update to 1 when the customer make the first order in e-business	CHAR	1				Used in synchronisation for sending back to local database only the users which proceeded at least one order. Activated from sync application.
u_group	List of groups for an user	TEXT					The groups must be separated by commas. If the user belong to one single group enclose the name of the group in commas (Eg: 'CLIENTI,').
u_maillist	List of mailling lists for an user	TEXT					
u_priority	A number between 0 and 10 which means the priority of the user	TINYINT	4			0	The priority is used for setting the access to a menu in public area.
u_info	Additional information about user	VARCHAR	100				Used only webPower profile - administration area
u_date_reg	Registration date	VARCHAR	100				
u_upload	Upload flag	CHAR	1				
u_postfile	Flag for upload - posting file	CHAR	1				
u_departement	Departement	VARCHAR	255				
u_nrint	Interior number	VARCHAR	100				
u_role	Role	VARCHAR	255				
company_id	Company id	INT	10				Corresponding identifier for company from wp_contacts

name	description	type	size	key	not null	default	observation
u_nickname	Nickname	VARCHAR	255				Used in forum, STalk
u_disabled	Flag for disable access in authenticated site	TINYINT	3				
u_agentallowchangeprice	Flag for agent to allow changing prices while making the order	CHAR	1				

Configurazione generale

La configurazione generale dell'applicativo prevede l'inserimento delle informazioni di base dell'azienda e la scelta del template grafico entro il quale visualizzare i contenuti del sito. Le informazioni vengono memorizzate nella tabella **tbl_admin** unitamente al login principale per l'accesso all'area di setup (in modo indipendente dalla gestione utenti). La configurazione generale è suddivisa in sezioni per facilitare l'amministratore nell'inserimento delle informazioni necessarie. A seguire analizzeremo le funzionalità disponibili nelle diverse sezioni.

Sezione "Company Info"

In questa sezione vengono inseriti e modificati i dati aziendali principali, salvati in **tbl_admin**, ed il login amministrativo per l'accesso all'area di setup. E' possibile impostare la visualizzazione delle informazioni principali relative all'azienda al piede di ogni pagina del sito.

WebPower permette di selezionare fino a sei lingue per la visualizzazione dei contenuti tra l'elenco di quelle disponibili di default (italiano, tedesco, inglese, francese, spagnolo e rumeno), fatta salva la capacità di introdurne di nuove traducendo i file contenenti le etichette visualizzate ed i messaggi di sistema e utilizzando il pulsante "Set Languages" per cambiare le sei lingue predefinite disponibili.

Al primo accesso ai contenuti del sito all'utente pubblico verrà proposta la lingua designata come default. Gli utenti registrati sono invece associati ad



una lingua selezionabile nel profilo utente, che verrà presentata dopo il login. L'area di setup consente la visualizzazione dei messaggi e delle etichette in una sola delle lingue disponibili.

Sezione "Setup / Configuration"

La sezione "Configuration" consente di definire le impostazioni relative a:

1. **Proprietà testo:** carattere, dimensioni, allineamento e stile dei contenuti delle pagine, delle etichette e dei menù/sottomenù.
2. **Logo/testo in alto:** visualizza o meno un logo ed un titolo nell'intestazione di ogni pagina.
3. **Visualizza/nascondi gli oggetti nel sito:** per rendere visibili o menu gli oggetti predefiniti sulle pagine (informazioni generali, data, contatore, riepilogo menù, mappa del sito, area di ricerca, ecc.) e definire il metodo di registrazione degli utenti (possibilità di registrarsi autonomamente, di apportare variazioni al proprio profilo, condizioni generali di accesso e note informative sulla privacy, ecc.).
4. **Impostazioni grafiche sito:** imposta la larghezza dell'area dei menu, la dimensione del banner in alto, il tipo di visualizzazione della scelta della lingua.
5. **Formato menu (primo livello):** consente di visualizzare i menù di primo livello in forma testuale, icona e testo o pulsate grafico, con possibilità di personalizzare il pulsante. La funzione che gestisce la grafica dei pulsanti si appoggia alle librerie grafiche GD2 di PHP.
6. **Mailing-lists:** nasconde il menù mailing list.
7. **Upload-Download:** nasconde il modulo upload/download e definisce l'invio di una mail di notifica all'amministratore ed agli utenti a seguito del caricamento di un nuovo file.
8. **Logo:** visualizzazione del logo come immagine o testo.
9. **eBusiness / DbTools:** visualizzazione del link ai moduli eBusiness e DbTools.



Questa sezione consente inoltre di personalizzare i messaggi di sistema del sito ("Accesso non consentito", "Pagina inesistente", ecc.), il testo di benvenuto nell'area riservata (dopo

avere effettuato il login), l'editor HTML WYSIWYG predefinito per la modifica e l'inserimento dei contenuti HTML.

Sezione "Setup / TopPages-Logo/Text"

Questa sezione è preposta alla impostazione della parte superiore del layout del sito, e consente di gestire:

- il banner come immagine o applicazione Flash
- il logo
- un titolo e sottotitolo specificandone lo stile

Le pagine sono personalizzabili con funzioni javascript scritte dall'amministratore da includere nella pagina tramite TAG inserito nel testo del titolo e sottotitolo.

Sezione "Groups/Users"

Nella sezione "Groups/Users" vengono gestiti gli utenti registrati, attribuendo a ciascuno privilegi di accesso differenziati alle diverse sezioni del sito ed ai relativi servizi/contenuti. Come già specificato gli utenti possono essere assegnati a gruppi per creare una ulteriore classificazione a cui è possibile assegnare permessi (es. "Amministrazione", "Ufficio Commerciale", "Ufficio Tecnico", ecc.).

Nome gruppo	Utenti #	Staff #	Utenti + Staff	Elimina ?
Administrator	0	1	Visualizza	
Classmates	15	5	Visualizza	

Nuovo gruppo

Nome gruppo (richiesto):

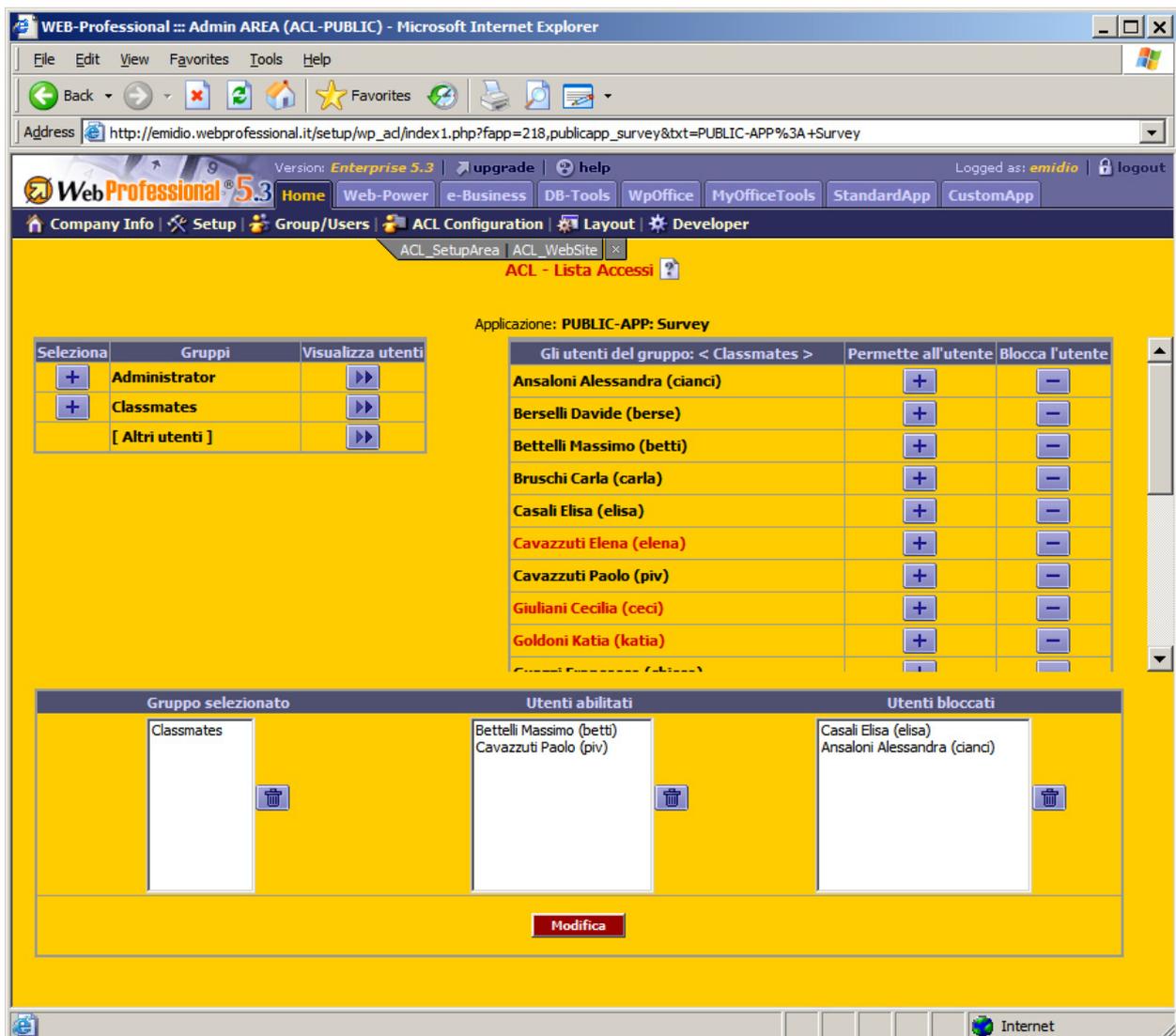
Gli utenti si differenziano nei due tipi **Staff** e **Clients**. Mentre agli appartenenti al primo tipo è concesso accedere all'area di setup del sito (con le limitazioni imposte dall'amministratore) ed all'area riservata, ai secondi è consentito accedere solo all'area riservata, dove per area riservata si intende l'insieme dei contenuti non accessibili normalmente agli utenti pubblici. Agli utenti oltre che alla priorità è associabile un periodo di validità (oltre alle consuete informazioni anagrafiche). Un flag "accettato" permette all'amministratio-

ne di abilitare o meno gli utenti registratisi attraverso il modulo pubblico di registrazione (l'amministratore può decidere se impostare automaticamente a TRUE tale valore). Una interfaccia di ricerca permette la visualizzazione dei soli record di interesse.

Nella gestione gruppi è possibile creare, eliminare un gruppo (è possibile eliminare solo i gruppi ai quali non è associato alcun utente) e visualizzarne gli utenti.

ACL Configuration

In "ACL Configuration" tramite il menù "ACL Setup Area" l'amministratore può abilitare, in modo differenziato, determinati utenti registrati di tipo Staff alla amministrazione (area setup) dei singoli moduli applicativi: Developer, eNews, Mailing List, Upload/Download,



The screenshot shows the 'ACL - Lista Accessi' page in the 'Admin AREA (ACL-PUBLIC)'. The application being configured is 'PUBLIC-APP: Survey'. The interface is divided into several sections:

- Navigation:** Includes 'Company Info', 'Setup', 'Group/Users', 'ACL Configuration', 'Layout', and 'Developer'.
- Group Selection:** A table with columns 'Selezione', 'Gruppi', and 'Visualizza utenti'. It lists 'Administrator', 'Classmates', and '[Altri utenti]'.
- User List:** A table titled 'Gli utenti del gruppo: < Classmates >' with columns 'Permette all'utente' and 'Blocca l'utente'. It lists users like 'Ansaloni Alessandra (cianci)', 'Berselli Davide (berse)', etc.
- Summary Section:** Three boxes at the bottom show 'Gruppo selezionato' (Classmates), 'Utenti abilitati' (Bettelli Massimo (betti), Cavazzuti Paolo (piv)), and 'Utenti bloccati' (Casali Elisa (elisa), Ansaloni Alessandra (cianci)).

WP Office, eBusiness e DbTools).

Tramite questa sezione l'amministratore delega a uno o più utenti di tipo Staff la gestione dei contenuti di porzioni del sito. I moduli abilitati verranno presentati agli utenti Staff dopo il login nell'area di setup.

Il menù "ACL Website" dà la possibilità di gestire l'accesso degli utenti a moduli del sito quali il forum (che può essere reso disponibile solo a determinati gruppi o utenti) e alle interfacce pubbliche di applicazioni sviluppate da terze parti (moduli personalizzati integrati nella piattaforma).

Integrazione utenti in un dominio NT

L'integrazione dell'autenticazione utenti della piattaforma WebProfessional con un dominio già esistente (sia esso basato su dominio NT4 o Active Directories) avviene attraverso la sincronizzazione degli utenti del dominio nella tabella **tbl_users** e dei gruppi in **tbl_group** tramite applicativo esterno (DB Sync) e l'impostazione dei parametri NTLM e DOMAIN nel file di configurazione della piattaforma **_app.php**. L'applicativo demanda l'autenticazione utente al Primary Domain Controller ma necessita comunque delle informazioni relative dell'utente stesso nelle tabelle della piattaforma per gestire i permessi sui contenuti.

Tipo di autenticazione

Il tipo di autenticazione utilizzata, demandata come dicevamo dalla piattaforma al Primary Domain Controller, è NTLM. La scelta di questo protocollo anziché il più recente Kerberos è dovuta a specifica richiesta dell'ente committente la nuova funzionalità, successivamente integrata come standard. La scelta di NTLM consente di mantenere piena compatibilità anche con realtà aziendali in cui Kerberos non è implementato.

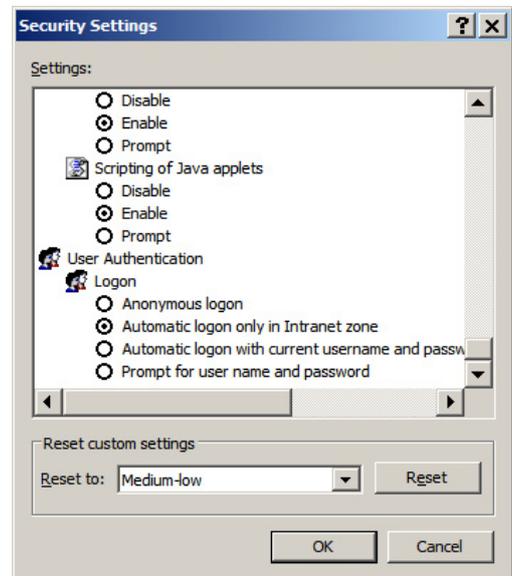
Requisiti di sistema

L'utilizzo di NTLM in sostituzione della verifica di username e password direttamente su **tbl_users** attraverso le funzionalità standard di WebProfessional comporta una serie di requisiti a livello di sistema che rendono l'autenticazione su PDC implementabile solo su webserver basati su sistema operativo Microsoft (è necessario l'utilizzo di Internet Information Server). Attualmente questa non viene considerata una forte limitazione in quanto

le casistiche in cui è richiesto questo tipo di autenticazione sono solitamente fortemente integrate con il mondo Microsoft.

Configurazione dei client

E' necessario che i client inviino al webserver le credenziali di autenticazioni in modo automatico per evitare che compaia la finestra di richiesta username e password. Nei client quindi l'URL della piattaforma deve essere inserita nella lista dei siti appartenenti alla Local Intranet e nelle opzioni avanzate, impostazioni di sicurezza, deve essere specificato che il client invii le credenziali di login ai siti appartenenti alla Local Intranet (impostazione di default all'installazione).



Configurazione di IIS

Internet Information Services (IIS) passa l'header "Negotiate" quando le richieste dei client vengono autenticate utilizzando l'opzione "Integrated Windows authentication". Il sito deve quindi essere impostato per utilizzare questo tipo di autenticazione attraverso l'interfaccia di configurazione del servizio.

Una volta effettuata l'impostazione, IIS invierà alle richieste dei client l'header "Negotiate" di default, che consente ai client di scegliere tra Kerberos e NTLM. "Negotiate" seleziona Kerberos a meno che non sia supportato da uno dei sistemi coinvolti nel processo di autenticazione. In tal caso l'autenticazione avverrà tramite NTML.

Il tipo di autenticazione utilizzata da IIS e' verificabile da riga di comando utilizzando lo script:

```
cscript adsutil.vbs get w3svc/NTAuthenticationProviders
```

Se "Negotiate" e' abilitato, otteremo in risposta:

```
NTAuthenticationProviders : (STRING) "Negotiate,NTLM"
```

L'autenticazione NTML deve essere forzata di default impostando IIS attraverso il comando:

```
cscript adsutil.vbs set w3svc/NTAuthenticationProviders "NTLM"
```

Configurazione di WebProfessional

Nel file di configurazione principale della piattaforma vanno impostate le variabili:

```
// Set NTLM authentication on NT domains
$lg_NTLM = 1;
$lg_NTLM_domain = "CPSOFTWARE";
```

Viene specificato di utilizzare l'autenticazione NTML al posto di quella di WebProfessional e viene inoltre specificato il nome di dominio sul quale gli utenti verranno autenticati.

Gli utenti del dominio devono essere presenti nella tabella **tbl_users**, e così pure i gruppi in **tbl_groups**. L'applicativo DB Sync si collega al PDC per effettuare le replica ad intervalli regolari, accedendo alle informazioni attraverso la chiamata API di Microsoft "NetQueryDisplayInformation" (Netapi32.dll) per ottenere una lista degli utenti e dei gruppi.

Cenni sulla funzione NetQueryDisplayInformation

La funzione NetQueryDisplayInformation restituisce gli user account, i computer, o i gruppi di un dominio.

```
NET_API_STATUS NetQueryDisplayInformation(
    LPCWSTR ServerName,
    DWORD Level,
    DWORD Index,
    DWORD EntriesRequested,
    DWORD PreferredMaximumLength,
    LPDWORD ReturnedEntryCount,
    PVOID SortedBuffer );
```

Significato dei parametri:

ServerName - Puntatore ad una costante di tipo stringa che specifica il DNS o il nome NetBIOS del server remoto su cui eseguire la funzione. Se assente, il computer locale viene utilizzato come default. Nota: in Windows NT questa stringa deve iniziare con \\.

Level - Specifica il livello delle informazioni. Questo parametro può assumere solo i seguenti valori:

- 1 restituisce le informazioni sugli account utenti. Il parametro SortedBuffer punta ad un array di strutture NET_DISPLAY_USER
- 2 restituisce informazioni su un computer specifico. Il parametro SortedBuffer punta ad un array di strutture NET_DISPLAY_MACHINE
- 3 restituisce informazioni sui gruppi. Il parametro SortedBuffer punta ad un array di strutture NET_DISPLAY_GROUP

Index - Specifica l'indice della prima informazione da restituire. Specificare 0 per iniziare dalla prima informazione disponibile.

EntriesRequested - Specifica il numero massimo di informazioni da restituire. Dalla versione 2000 di Windows il numero massimo è 100.

PreferredMaximumLength - Specifica la dimensione massima desiderata, in byte, del buffer allocato da sistema restituito nel parametro SortedBuffer. Si consiglia di impostare il valore a MAX_PREFERRED_LENGTH.

ReturnedEntryCount - Puntatore al valore che riceve il numero di informazioni restituite nel parametro SortedBuffer.

SortedBuffer - Puntatore ad un buffer che riceve un puntatore ad un secondo buffer allocato da sistema contenente la lista ordinata delle informazioni richieste. Essendo il buffer allocato da sistema, deve essere svuotato utilizzando la funzione NetApiBufferFree.

Funzione di autenticazione NTML - codice

Il login tramite NTML viene ricondotto alla procedura standard di WebProfessional leggendo la variabile di sessione \$AUTH_USER (che riceve l'informazione di login dal client) estraendone username e dominio di appartenenza, rispettivamente memorizzati nelle variabili \$vlogin e \$vdomain.

```
if (isset($AUTH_USER) && $lg_NTLM == 1) {  
    $vlogin = $AUTH_USER;  
    $vlogin = str_replace("\\", "/", $vlogin);  
}
```

```
$tmp = explode("/", $vlogin);  
$vdomain = $tmp[0];  
$vlogin = $tmp[1]; }
```

Successivamente viene verificata tramite una semplice SELECT la presenza del nome utente in tbl_users e la coincidenza del nome di dominio, omettendo la verifica della password (non è possibile "leggere" le password degli utenti di dominio ma solamente verificare la correttezza di una password su un utente specifico - considerando che l'autenticazione avviene solo tramite NTML tra il client (Internet Explorer) ed il PDC tutta la parte effettiva di autenticazione è demandata al PDC, limitando il compito di WebProfessional a garantire l'accesso all'utente una volta confermate le credenziali).

Integrazione dati - DB Sync

La piattaforma WebProfessional si interfaccia ai dati aziendali attraverso l'applicativo DB Sync. DB Sync è un programma scritto in Visual Basic che gestisce la pubblicazione e sincronizzazione dati da una fonte ODBC al database MySQL della piattaforma. Il trasferimento avviene attraverso XML compresso in archivi .gz sulla porta 80 o 443 (HTTP o HTTPS in base alla scelta dell'utente) agendo come un normale browser per consentire il funzionamento dell'applicativo anche in reti protette da firewall o con utilizzo di proxy server.

DB Sync consente di sincronizzare i dati in modo completo oppure pubblicare i dati da sorgente a web e viceversa (non si tratta quindi in questo secondo caso di una sincronizzazione dati ma di un semplice aggiornamento).

La sincronizzazione avviene tramite un database locale utilizzato da DB Sync come supporto alla sincronizzazione ed attraverso la gestione del campo SyncStatus (presente sia nelle tabelle del database locale sia su quello web) per l'identificazione dello stato dei record.

Il processo di sincronizzazione

La sincronizzazione dati avviene attraverso tre step principali descritti nello schema della pagina seguente. In una prima fase il database sorgente viene comparato con il database locale di DB Sync (MS Access, MSDE o MS SQL Server) - la comparazione avviene per

ogni tabella/query record per record, campo per campo. L'utilizzo intensivo di banda per il passaggio dati normalmente non costituisce un limite in rete locale (ove avviene la prima fase della sincronizzazione).

Il database locale mantiene una copia di quello online (oggetto della comparazione) ove ad ogni tabella/query è associato un campo (SyncStatus) che gestisce lo stato del record. Inizialmente il database locale contiene le sole strutture (senza dati) designate per la sincronizzazione. La prima sincronizzazione si traduce quindi in una copia dei dati dal database di origine a quello web e nella definizione del campo SyncStatus per ogni record.

La tabella fisica locale oggetto della sincronizzazione viene creata in base alla query eseguita sul database sorgente - è quindi possibile effettuare durante la sincronizzazione query di aggregazione (GROUP BY) o utilizzare la clausola WHERE per limitare i record da sincronizzare.

Il campo SyncStatus può assumere i seguenti valori:

Stato	Descrizione	Note
S	Synchronized	Sincronizzato da DB Sync.
U	Updated	Aggiornato dalla piattaforma WebProfessional.
D	Deleted	Cancellato dalla piattaforma WebProfessional.
A	Added	Aggiunto.
W	Deleted from web	Cancellato da web con sincronizzazione di tipo "pubblicazione"

La prima sincronizzazione porta lo stato di ogni record nel database locale ad "A". La seconda fase consiste nella comparazione dei record con Syncstatus diverso da "S" e "W" con i record del database web con stato diverso da "S". In questo modo si riduce notevolmente il passaggio dati da e verso il server web, riducendo drasticamente l'utilizzo di banda ed il tempo di sincronizzazione.

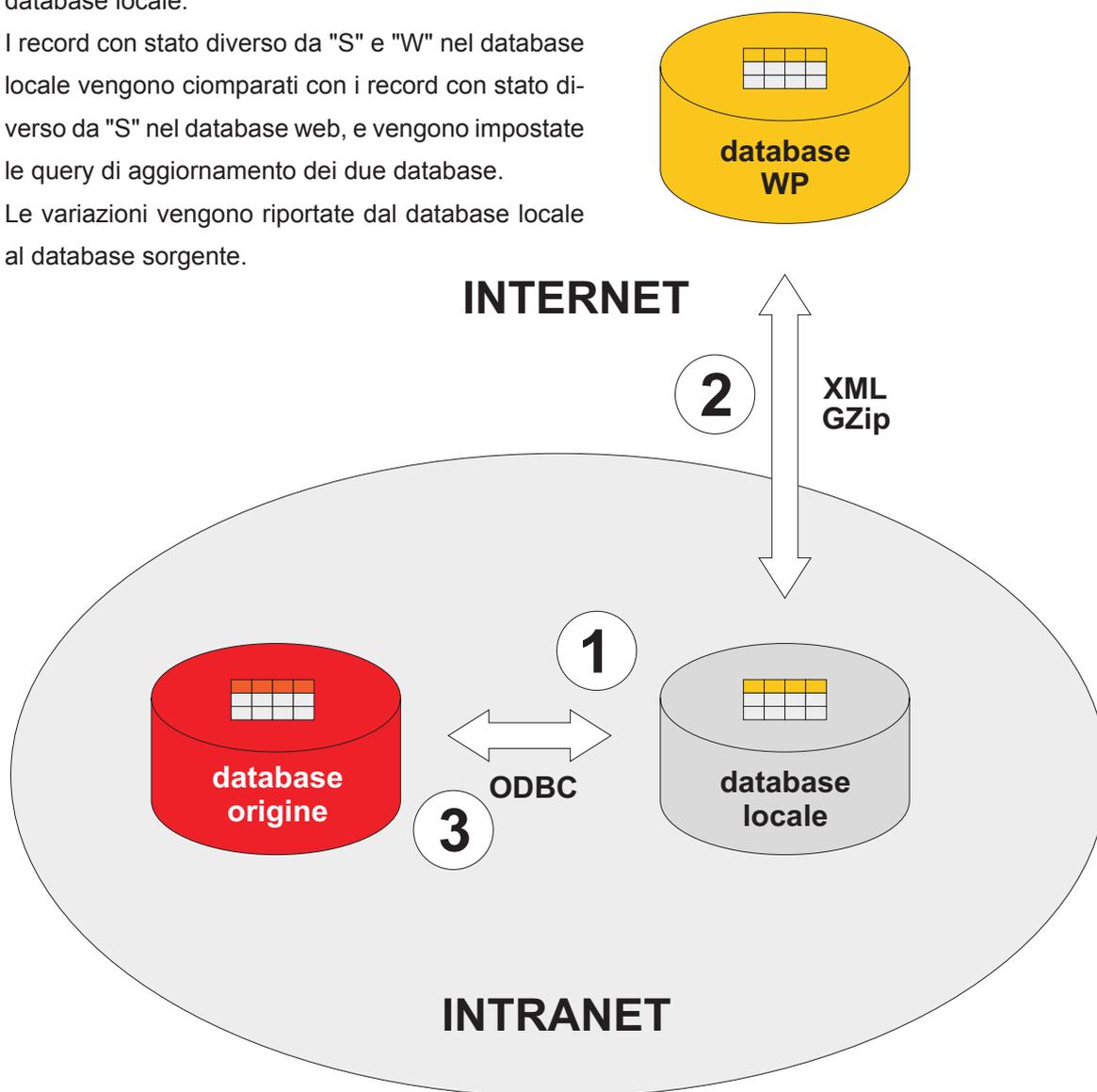
Le gestione delle operazioni sui dati tramite il confronto del campo SyncStatus avviene con le modalità descritte nella tabella sottostante. In caso di conflitti è possibile intervenire manualmente con interazione da parte dell'utente o definire quale database è prioritario (se locale o web) per la risoluzione automatica dei conflitti.

Stato Locale	Stato Web	Statement SQL	Azione
S	S	-	Nessuna azione
A	-	INSERT	Il record viene inserito sulla corrispondente tabella su web

Stato Locale	Stato Web	Statement SQL	Azione
-	A	INSERT	Il record viene inserito sulla corrispondente tabella locale
U	S	UPDATE	Il record viene aggiornato nella corrispondente tabella web
S	U	UPDATE	Il record viene aggiornato nella corrispondente tabella locale
D	S/D	DELETE	Il record viene eliminato dalla tabella web
S/D	D	DELETE	Il record viene eliminato dalla tabella locale

I tre step del processo di sincronizzazione dati effettuato tramite l'applicativo DB Sync.

1. Il database sorgente viene comparato con quello locale e viene impostato lo stato di ogni record sul database locale.
2. I record con stato diverso da "S" e "W" nel database locale vengono comparati con i record con stato diverso da "S" nel database web, e vengono impostate le query di aggiornamento dei due database.
3. Le variazioni vengono riportate dal database locale al database sorgente.



Stato Locale	Stato Web	Statement SQL	Azione
W	-	-	Il record è stato eliminato su web, la sincronizzazione è impostata in "pubblicazione" e l'utente non desidera che il record cancellato su web venga risincronizzato ulteriormente.

Elencate a seguire le situazioni di conflitto in cui è richiesto l'intervento dell'utente o la definizione del database prioritario.

Stato prioritario	Stato non prioritario	Statement SQL	Azione
A	D/U/A	DELETE + INSERT	Il record viene prima eliminato dalla tabella nel database non prioritario e poi inserito come da database prioritario.
D	A/U	DELETE	Il record viene eliminato dalla tabella nel database non prioritario.
U	A/D	UPDATE	Il record viene aggiornato nella tabella del database non prioritario.

Dopo la risoluzione dei conflitti ed il relativo aggiornamento del database locale le modifiche vengono replicate sul database web e lo stato di tutti i record su web viene impostato a "S".

Nella terza fase le modifiche vengono replicate dal database locale (immagine del database web) al database sorgente, e tutti i record del database locale assumono stato "S".

La gestione del campo SyncStatus implica alcune particolarità nella classe PHP utilizzata per l'interfacciamento al database di WebProfessional. I record non vengono cancellati: l'operazione di DELETE viene sostituita da un UPDATE sul campo SyncStatus (lo stato viene aggiornato a "D"). La classe provvede inoltre a gestire lo stato dei record per le operazioni di inserimento ed aggiornamento.

L'effettiva eliminazione dei record è demandata al processo di sincronizzazione, a meno che non venga impostato espressamente di non utilizzare DB Sync sulla tabella, per cui la classe PHP esegue direttamente la cancellazione fisica dei record (pur continuando a gestire lo stato per le altre operazioni).

DB Sync si appoggia a script PHP lato server per la gestione dei dati inviati in formato XML e per l'esecuzione delle relative query.

Tipi di sincronizzazione

DB Sync gestisce diversi tipi di sincronizzazione:

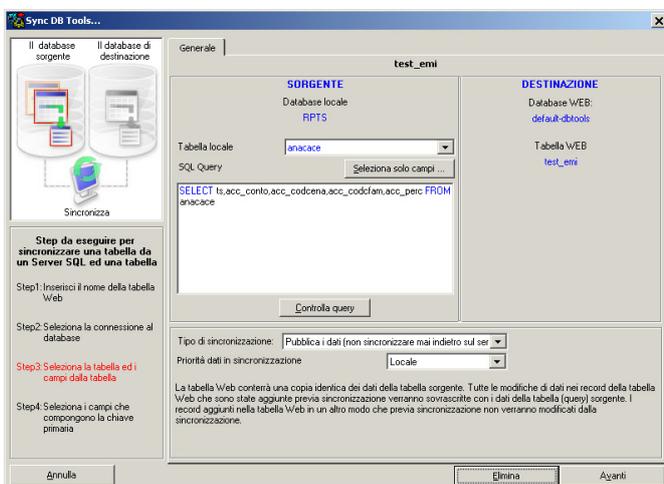
Sync WP - Sincronizzazione su tabella esistente su database web - prevede una schermata per la definizione della corrispondenza campo-campo della sincronizzazione per ogni tabella. Il programma esegue un controllo sul tipo di campo impedendo l'accoppiamento di tipi incompatibili ed avvertend l'utente in caso di compatibilità tra i tipi ma dimensioni differenti (ad esempio la sincronizzazione di un campo VARCHAR(50) su un campo VARCHAR(30) porta al troncamento della stringa contenuta comunque a 30 caratteri).

Sync DbTools - Sincronizzazione dati che prevede la creazione su database web della tabella risultante dalla query impostata in DB Sync. E' fondamentale specificare la chiave primaria della tabella risultante dalla query per consentire il corretto funzionamento dell'applicazione.

Sync E-Commerce - E' un pacchetto di sincronizzazioni di tipo DbTools preparate per lo scambio dati con il gestionale Business. Provvede a sincronizzare articoli, categorie e sottocategorie, utenti, prezzi e listini.

Oltre a queste sono presenti:

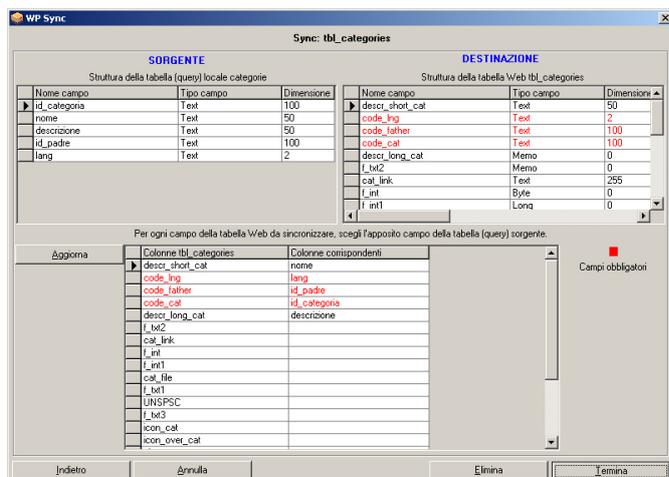
Sync - Aggiunta successivamente prevede la sincronizzazione dati del database sorgente con database di destinazione non WebProfessional.



Schermata di configurazione di una sincronizzazione nell'applicativo DB Sync. Le impostazioni fanno riferimento ad una fonte dati ODBC RPTS, con sola pubblicazione dati (da sorgente a web).

Diventa rilevante per il processo di sincronizzazione la definizione della chiave primaria (richiesta nello step successivo) nel caso si utilizzi una query con JOIN su più tabelle.

Oltre che per la sincronizzazione dati DB Sync può essere utilizzato anche per sincronizzare il contenuto di due directory (file contenuti ed eventualmente directory contenute).



Nelle sincronizzazioni di tipo Sync WP la tabella di destinazione è già esistente. DB Sync visualizza i campi presenti nelle due tabelle lasciando all'utente il compito di definire le esatte corrispondenze.

DB Sync non permette di impostare corrispondenze su tipi di dati non compatibili ed avverte l'utente se la corrispondenza è impostata tra campi con tipo dati compatibile ma dimensioni differenti.

Sync Cartelle - Sincronizzazione di file a directory tra due directory specificate (una locale ed una remota).

Osservazioni su DB Sync

DB Sync è ad oggi strumento indispensabile nell'utilizzo di alcuni moduli di WebProfessional, laddove la mole di dati da gestire non consente l'utilizzo delle interfacce web di amministrazione se non in casi particolari. Si pensi ad una applicazione di ecommerce con migliaia di prodotti online, relativi listini e scontistiche ed anagrafiche clienti.

I moduli principali di WebProfessional

La piattaforma WebProfessional è composta da moduli distinti, attivabili separatamente, integrati con il sistema di gestione permessi ed utenti precedentemente descritto. I moduli principali sono:

- **WebPower** - per la gestione dei contenuti della pagine, news, mailing list, upload e download di file
- **eBusiness** - applicazione di commercio elettronico per B2B e B2C o di presentazione prodotto online (modalità catalogo)
- **DbTools** - per lo sviluppo e la visualizzazione di query su database definibile dall'utente con presentazione dati filtrata ed integrata in modo dinamico con il login utente e l'appartenenza ai gruppi
- **WP Office** - mail, agenda, task, report, chiamate telefoniche, per la gestione ed orga-

La piattaforma

Le caratteristiche dei moduli WebPower ed eBusiness nelle tre versioni della piattaforma WebProfessional.

	Advanced	Premium	Enterprise
WebPower (sito internet - news / Aree private - Intranet)			
Livelli di menu	2	3	3
Lingue gestite	2	6	6
Template grafici	4	7	9
Registrazione su motori di ricerca	Si	Si	Si
Web Designer Tools	Si	Si	Si
Area News e Mailing List	Si	Si	Si
Area Upload - Download	-	Si	Si
Statistiche	Si	Si	Si
Export sito su CD	-	-	Si
Aree Private per utenti	-	Max.50	Illimitati
Gestione applicazioni esterne	-	Si	Si
Forum on-line	-	Opz.	Si
eBusiness (Cataloghi ed ordini on-line)			
Catalogo prodotti	Si	Si	Si
Ordini on-line	-	Si	Si
Export catalogo su CD	-	-	Si
Gestione Varianti e Ricambi	-	Si	Si
Credit card - transazioni sicure	-	Opz.	Opz.
Sincronizzazione dati/Import-export	-	Si	Si
Gestione listini e sconti avanzata	-	Si	Si
Gestione agenti e rappresentanti	-	-	Si
Interfaccia PDA	-	-	Si
Collegamento diretto a disegni PDF	-	-	Opz.

Database **TOOLS** Integrazione dati su Web:

- Gestione dati in lettura/scrittura
- Gestione grafici
- Interfaccia PDA
- Run-time Database Tools (no PDA)

WP Office **Organizzazione Aziendale - CRM:**

- Agenda online (planning)
- Gestione rapportini d'intervento
- E-mail integrata
- Gestione chiamate
- Gestione attività (to do list)
- Archiviazione documenti (Opz.)
- Gestione note e priorità
- Contatti Clienti e Fornitori (CRM - SCM) (Opz.)

nizzazione del lavoro quotidiano

Con riferimento al capitolo introduttivo relativo ai CMS e WCMS WebProfessional si colloca quindi tra le piattaforme di tipo **Module-based systems** pur mantenendo la flessibilità tipica dei **Content management frameworks**. La possibilità di modificare i template e di gestire informazioni di tipo generico tramite il modulo DbTools concilia la facilità di utilizzo ed i ridotti tempi di startup delle piattaforme basate su moduli con la possibilità di personalizzare ed adattare la piattaforma alle esigenze dell'utente finale tipica dei framework senza ereditarne la complessità di gestione e sviluppo.

La suite WebProfessional è strutturata in tre versioni che dipendono dai moduli inclusi e della complessità dalle loro caratteristiche. Da una presenza semplice ed elementare su web a sistemi ecommerce articolati le differenti versioni sono state create per soddisfare le esigenze di aziende di diverso profilo e dimensione.

I principali moduli di WebProfessional verranno descritti più approfonditamente nei capitoli seguenti.

WebPower

WebPower è un sistema innovativo per gestire in modo autonomo siti internet professionali multilingue, con grafica indipendente dai contenuti, news, area privata, registrazione su motori di ricerca, statistiche, ecc. Il sistema permette l'amministrazione dei contenuti in modo semplice attraverso interfaccia web utilizzando un comune browser (per accedere alle funzionalità occorre essere connessi ad internet) senza alcuna installazione lato client e senza richiedere particolari competenze nel linguaggio HTML o in altri linguaggi di programmazione.

Per la maggior parte delle funzioni sono supportati tutti i principali browser ed i diversi sistemi operativi. Tuttavia per la gestione dei contenuti di pagina con l'editor completo WYSIWYG è indispensabile il browser MS Internet Explorer su piattaforma Windows in quanto l'editor si basa tecnologia ActiveX (è comunque presente un editor alternativo che garantisce comunque le funzionalità principali anche su altri browser).

WebPower è disponibile in tre diverse versioni: Advanced, Premium ed Enterprise e, in funzione delle versioni, sono disponibili vari moduli, integrabili nel sistema anche successivamente all'attivazione, mantenendo la struttura ed i contenuti già creati.

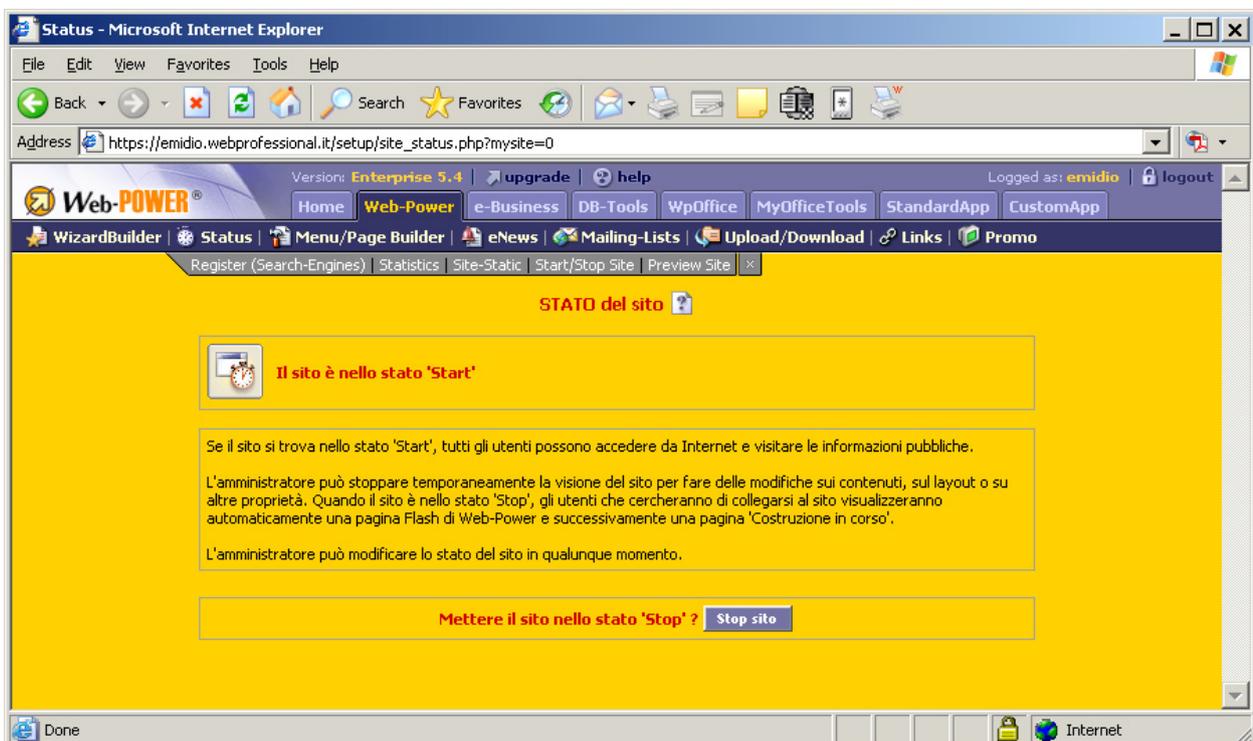
L'impostazione grafica e la disposizione dei contenuti avvengono attraverso template grafici e moduli. Una volta scelto il template più appropriato ai contenuti ed alla grafica del sito (template che è possibile cambiare in qualsiasi momento grazie alla completa separazione dei contenuti del sito dal layout e dalla grafica dello stesso) attivando i diversi moduli di WebPower è possibile gestire informazioni di diversa natura.

Dall'area di setup i moduli possono essere resi visibili o meno, vengono amministrati i contenuti e gli utenti ed assegnati i permessi sui contenuti, mentre nell'area pubblica o privata i contenuti vengono messi a disposizione degli utenti pubblici o di quelli registrati (previo inserimento di login e password).

Le funzioni di WebPower

Andremo ora ad analizzare le principali funzionalità di WebPower, esplorando i menù di configurazione dell'area amministrativa. L'area di setup del modulo WebPower si articola in una serie di voci che si riassumono in:

- **WizardBuilder** - per scegliere rapidamente layout/contenuti per il proprio sito; consiste in una serie di maschere che guidano l'utente nelle impostazioni di base per poter



pubblicare contenuti sul proprio sito

- **Status**

- **Register** *per registrare il sito su motori di ricerca*
- **Statistics** *visualizza statistiche sugli accessi al sito*
- **Site-Static** *per creare copia statica del sito in HTML, oppure per creare un archivio compresso (.zip) delle pagine del sito (HTML) comprensivo di autorun per la copia dei contenuti su CD ridistribuibili*
- **Start/Stop site** *per porre il sito nello stato di "under construction"*
- **Preview site** *per la visualizzazione dell'anteprima del sito senza uscire dall'area amministrativa*

- **Menu/Page Builder** - per inserire / variare / eliminare menù, sottomenù e relativi contenuti delle pagine
- **eNews** - per inserire / variare / eliminare news
- **Mailing-Lists** - per inserire / variare / eliminare mailing list e per inviare email agli utenti iscritti ad una determinata mailing list
- **Upload/Download** - per caricare / gestire / cancellare file nella sezione "upload - download" presente nel sito
- **Links** - per inserire / modificare / cancellare link o indirizzi web utili disponibili per gli utenti
- **Promo** - per inserire / modificare / cancellare immagini promozionali da visualizzare sul sito

Nei paragrafi successivi le funzionalità sopra elencate verranno esaminate nel dettaglio, sia dal punto di vista operativo che delle strutture dati ad esse collegate.

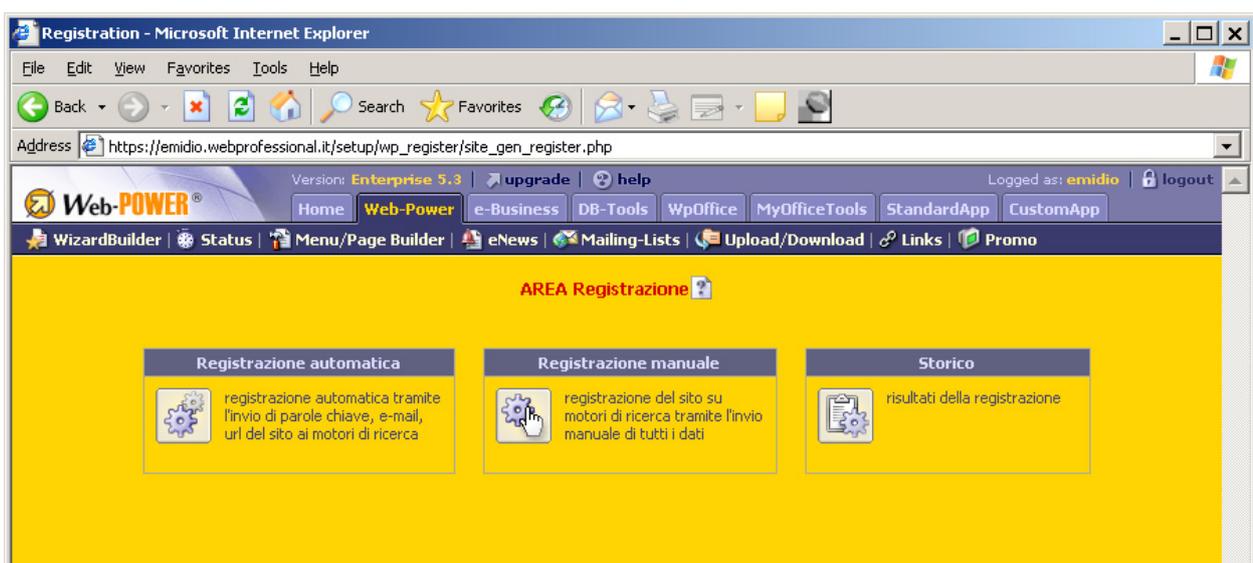
WizardBuilder

Il modulo Wizard è stato realizzato per raggiungere, in modo mirato e veloce, due delle funzioni più importanti per la creazione di un sito WebPower e che rappresentano la base per iniziare a lavorare sulle pagine: la definizione del layout del sito ed i contenuti delle pagine. E' possibile importare i contenuti e la struttura dei menù da un sito esistente per poi effettuare gli eventuali aggiustamenti alla disposizione dei testi e delle immagini.

Status - Register

La funzione consente la registrazione del sito sui più importanti motori di ricerca, sia in modo automatico che manuale. Prima di procedere alla registrazione del sito sui vari motori occorre ottimizzare e predisporre il sito stesso in modo che questa sia il più efficace possibile, per ogni lingua. Per fare ciò occorre:

- inserire contenuti sufficientemente significativi ed esaustivi
- impostare per ogni contenuto un titolo attinente - il titolo verrà inserito nella pagina HTML tra i TAG `<title></title>`

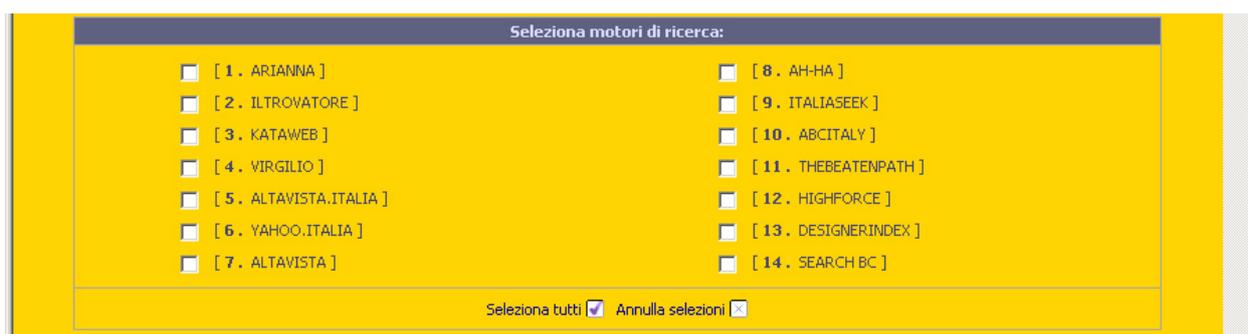


Status - Register - Questa schermata propone i tre diversi metodi per registrare il proprio sito sui motori di ricerca.

- scegliere correttamente e coerentemente le parole chiave (da dieci a venti, separate da virgola composte da uno o più termini ciascuna) che verranno anche inserite nella pagina come META TAG

La registrazione può avvenire attraverso una procedura automatica che invia le informazioni ai motori di ricerca che supportano l'invio da pagine esterne, oppure manualmente dove per ogni motore di ricerca selezionato verrà presentata la relativa form di inserimento (proprietaria del motore di ricerca) già precompilata con le informazioni inserite.

I dati riguardanti i motori di ricerca sono contenuti nella tabella **tbl_www_engines** di



wp2_generaldb (database di configurazione del servizio WebProfessional che contiene le informazioni specifiche registrate per ogni attivazione - la piattaforma è in grado di gestire un numero di domini virtuali - e quindi di siti - limitato solo dall'hardware del server).

tbl_www_engines (database wp2_generaldb)

name	description	type	size	key	not null	default	obervation
reg_id	search engine internal id	INTEGER	11	yes	yes		
reg_name	display name	VARCHAR	100		yes		
reg_url	registration URL	VARCHAR	255				URL complete with GET parameters: ^webpage, ^key, ^keys, ^email, ^name are substitute by the values entered in the site configuration page.

name	description	type	size	key	not null	default	obervation
reg_res	text to display after registration	VARCHAR	255				
reg_auto	engine is enabled for automatic registration	CHAR	1				
reg_file	file to be attached to registration request	VARCHAR	10				
flag1	future use	CHAR	1				
flag2	future use	CHAR	1				
flag3	future use	CHAR	1				
predictions	description for time required for registration	VARCHAR	255				This field is indicative only - search engine spidering time may vary depending on the site and the period.

Storico

Trattasi di un riepilogo sulle registrazioni effettuate (data, keywords utilizzate nelle varie lingue, tipo di registrazione utilizzata - automatica o manuale - e nomi dei motori di ricerca). Dalla stessa schermata è possibile cancellare lo storico. Lo storico e le parole chiave vengono memorizzate nella tabella locale **tbl_register**.

La tabella risulta attualmente priva di chiave primaria (pur essendo specificata in fase progettuale) e costituita da un unico record (mentre originariamente ogni motore di ricerca avrebbe dovuto generare un record nella tabella dello storico). Le modifiche necessarie a riportare la struttura del database al progetto originale non sono state effettuate in quanto l'accesso a tale tabella è estremamente limitato e non impatta sulle prestazioni generali della piattaforma.

tbl_register

name	description	type	size	key	not null	default	obervation
keys1	keywords section 1	VARCHAR	255				
keys2	keywords section 2	VARCHAR	255				
keys3	keywords section 3	VARCHAR	255				
keys4	keywords section 4	VARCHAR	255				
keys5	keywords section 5	VARCHAR	255				
keys6	keywords section 6	VARCHAR	255				

name	description	type	size	key	not null	default	obervation
engines	history description	TEXT					
flag	unknown	CHAR	1				probably old field, not used any more - should be deleted

Status - Statistics

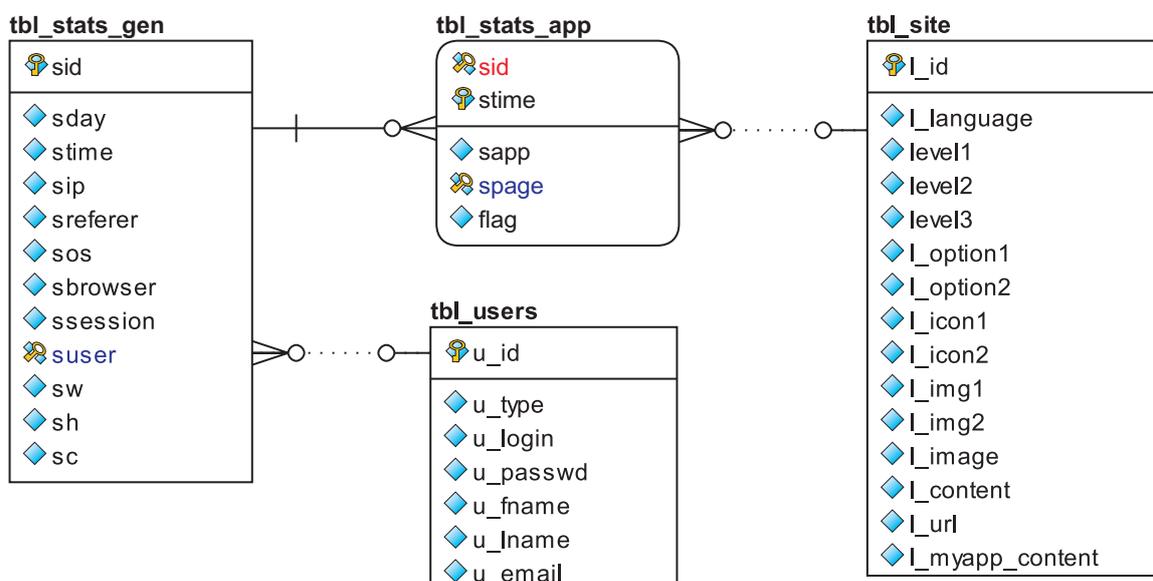
Vengono mostrate informazioni statistiche relative alla navigazione delle pagine del sito, suddivise per:

- statistiche per un selezionato periodo, raggruppate per giorni, settimane e mesi
- statistiche per documenti, riferimenti, indirizzi IP, tipi di browser, risoluzione video e per tipo di sistema operativo

Schema delle tabelle che intervengono nel modulo statistiche di WebProfessional. Nella tabella principale **tbl_stats_gen** troviamo un record per sessione, mentre nella tabella **tbl_stats_app** le pagine visitate all'interno di ogni sessione. Il campo **tbl_stats_app.sapp** identifica il modulo applicativo richiamato:

- 1 Site
- 2 News
- 3 Upload/Download
- 4 eBusiness
- 5 DBTools

Il campo **tbl_stats_app.spage** individua la pagina (o la query, o altro in dipendenza del modulo) specifica del modulo stesso.

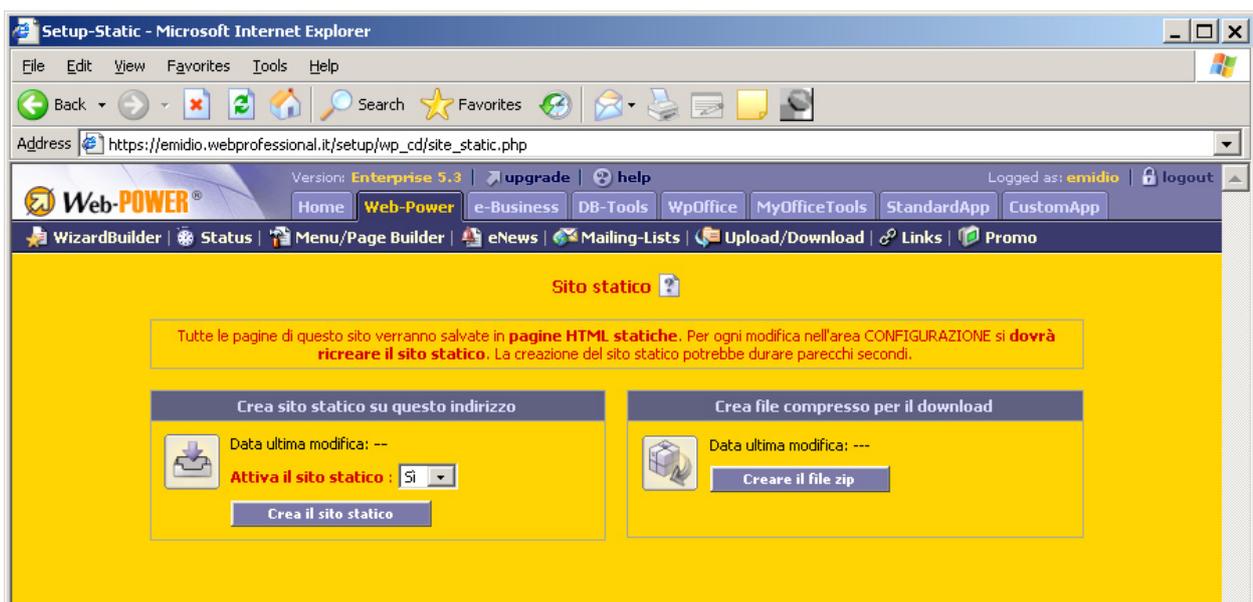


- statistiche WebProfessional, differenziate per id di pagina visualizzata, categoria di e-news, file scaricati dal modulo Upload/Download, ed utenti che hanno effettuato il login sul sito

Attraverso le statistiche è possibile monitorare oltre che l'effettivo impatto delle singole pagine del sito anche l'attività degli utenti registrati (personale interno, agenti) per verificare l'effettivo utilizzo dello strumento ed individuare le sezioni più utilizzate, identificando i servizi di maggiore e minore interesse per gli utilizzatori.

Status - Site-Static

Mediante tale funzione è possibile creare una copia statica del sito (ad esempio per non sospenderne la visibilità in caso di lunghi interventi di manutenzione sia delle pagine che della struttura - dovrà essere rigenerato, tutte le volte, per rendere visibili le modifiche realizzate). Si ricorda che la visualizzazione di pagine statiche HTML favorisce la reperibilità del sito attraverso i motori di ricerca, in quanto le ricerche meglio indicizzate sono relative a pagine di questo formato. Nella versione Enterprise è possibile esportare il sito



in un archivio per la masterizzazione su CD. La generazione del codice HTML avviene ridirigendo l'output dell'interprete PHP su file. L'output viene processato modificando i link interni al sito per consentire la normale navigazione con i file generati con estensione .htm. I file generati con le relative immagini possono poi essere compressi in un archivio tar gzip reso disponibile per il download tramite comando shell chiamato da codice PHP attraverso la funzione `shell_exec()`.

Status - Start/Stop site

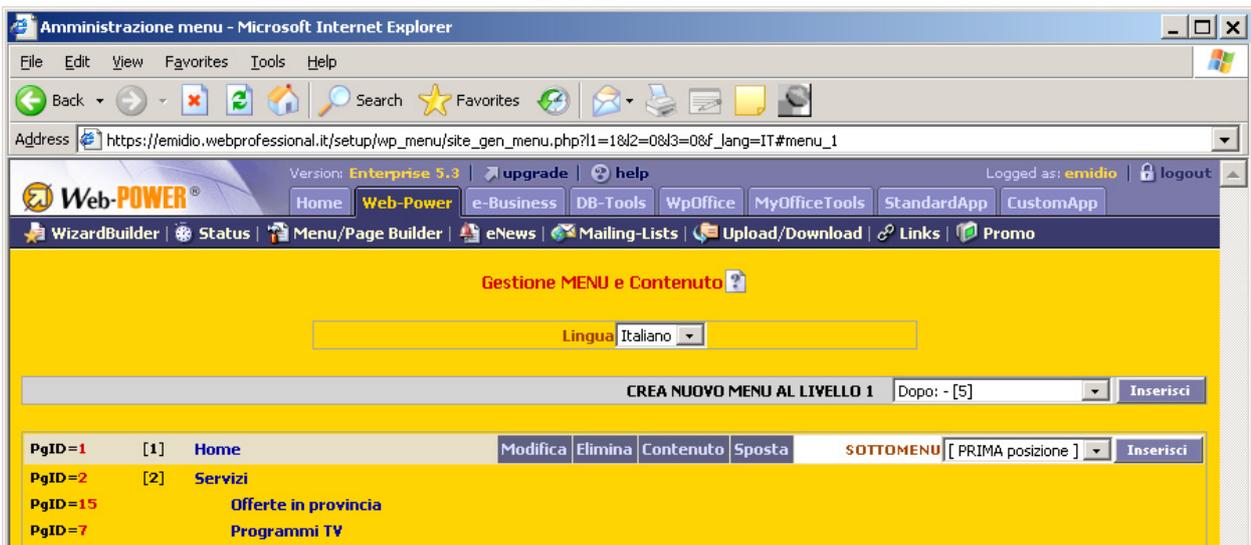
Tali funzioni sono particolarmente utili quando si devono apportare modifiche o aggiornamenti alle pagine web del sito (nel cui caso può essere opportuno sospenderne la navigazione ai visitatori). Nella pagina visualizzata è riportato lo stato attuale del sito che è possibile modificare attraverso l'apposito pulsante. Nello stato di "Stop" ai visitatori viene automaticamente proposta una breve animazione flash di WebPower che termina con una pagina contenente il messaggio "Sito in costruzione / Under construction". Attraverso l'Administration Area è possibile lavorare sul sito per apportare le modifiche o gli aggiornamenti desiderati (da Status - Preview site è visibile una anteprima delle variazioni o degli aggiornamenti realizzati).

Status - Preview site

La funzione visualizza un'anteprima delle variazioni o degli aggiornamenti senza necessariamente riattivare il sito ogni volta. La funzionalità richiama il codice php anziché le pagine html statiche generate. Per rendere visibili i cambiamenti con il sito statico attivato occorre rigenerare le pagine.

Menu/Page Builder

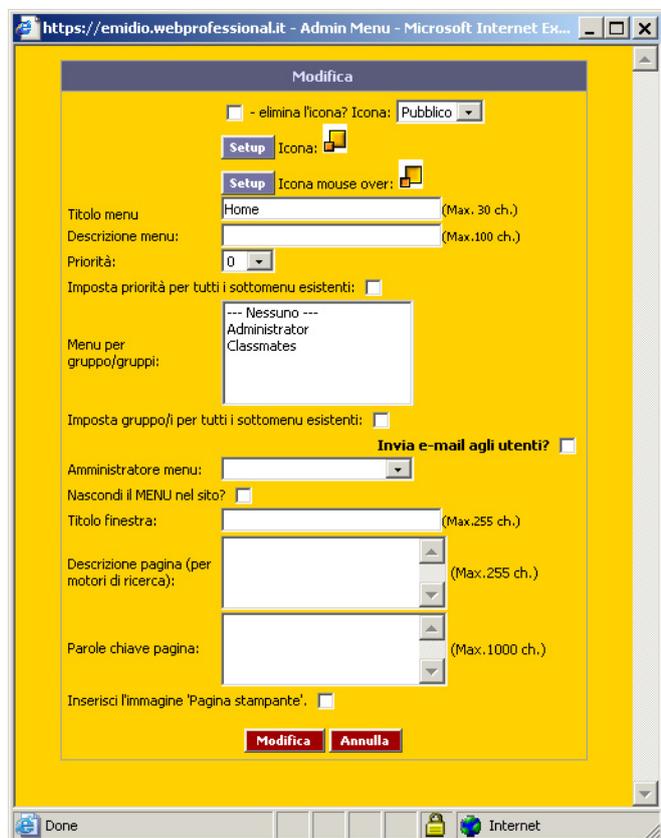
Prima di utilizzare questa funzione potrebbe essere utile avere già i testi da inserire e i



menù da creare. Se non è stato ancora creato alcuna voce di menù compare un campo "URL" in cui è possibile digitare l'indirizzo web di un altro sito di cui si desidera copiare la struttura ed i contenuti: WebPower tenta di clonarne la struttura, riportandola all'interno del nuovo sito per facilitare all'utente la migrazione da pagine esistenti (statiche o dinamiche) alla piattaforma WebProfessional.

Creare una voce di menù

- selezionare dalla voce "Nuovo menù al livello 1" la posizione in cui inserire il menù
- cliccare su "Inserisci" (sarà inserito un nuovo menù "???" nella posizione scelta)
- selezionare "Modifica" sulla riga della nuova voce di menù e, nell'apposito campo della finestra che apparirà, sostituire "???" digitando il titolo del



menù, quindi cliccare su "Modifica"

Creare una voce di sottomenù

- selezionare la voce del menù cui legare la voce di sottomenù
- selezionare dalla casella "Sottomenu" la posizione in cui inserire la nuova voce di sottomenù
- cliccare su "Inserisci" (sarà inserito la nuova voce di sottomenù "???" nella posizione scelta (nelle versioni Advanced è possibile creare un solo livello di sottomenù, mentre nelle versioni Premium ed Enterprise è possibile creare due livelli di sottomenù per un totale di tre livelli)

La limitazione dei livelli a tre non deriva dalla struttura del database ma dal codice PHP. Nell'analisi iniziale era previsto l'utilizzo di una funzione ricorsiva per la visualizzazione dei menù - la limitazione a tre livelli e la mancanza di ricorsione venne rilevata a piattaforma già ultimata ed in produzione. Tuttavia non venne apportata alcuna modifica al codice in quanto si ritenne non pregiudizievole all'utilizzo tale limite, senza considerare che un maggior numero di livelli spesso riduce la navigabilità del sito disorientando l'utente.

Eliminare una voce di menù/sottomenu

- cliccare sulla funzione "Elimina" della riga della voce di menù/sottomenù da eliminare (attenzione: saranno eliminate anche tutte le voci di sottomenù ad essa correlati)

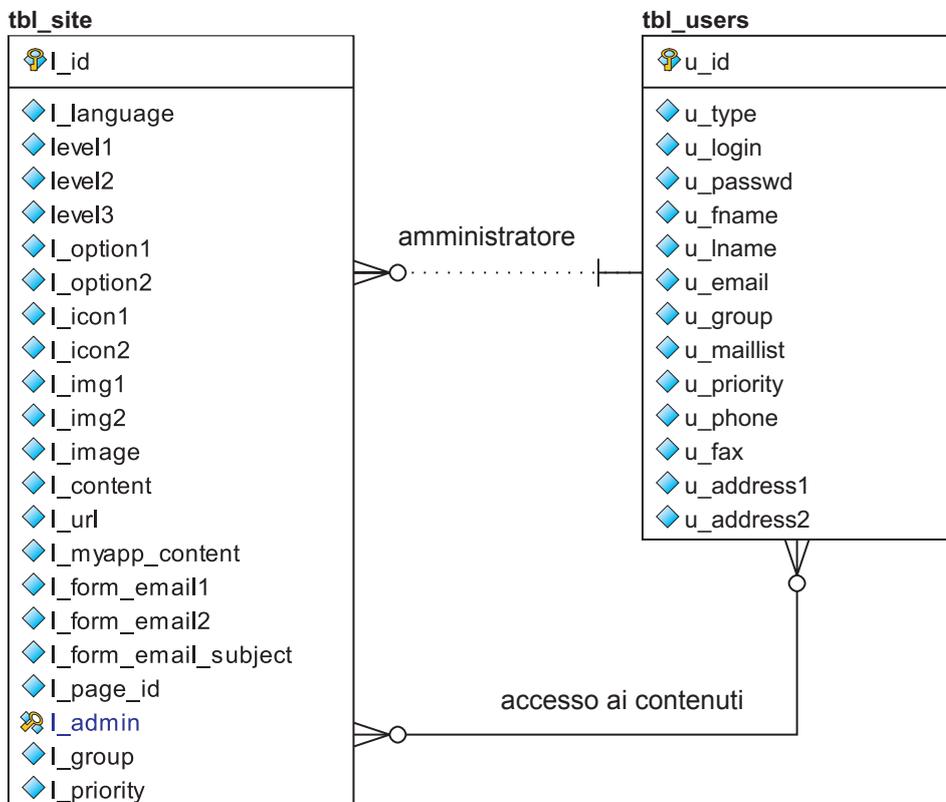
Creare i contenuti di una pagina

Una volta definita la struttura generale dei menù del sito, occorre inserire i contenuti delle pagine. E' possibile consentire la costruzione di ogni pagina ad utenti registrati "Staff" che potranno accedere, mediante il proprio login e password, all'area di setup e potranno modificare le pagine ammesse al loro controllo, se sono stati definiti come amministratori

dei menù dall'opzione Modifica; quest'opzione è disponibile per ogni menù di primo livello e definisce gli accessi per tutto il ramo di menù. Il codice utente dell'amministratore di menù è salvato nel campo **tbl_site.l_admin**.

Le pagine possono essere rese visibili solo a determinati gruppi di utenti, oppure a utenti che hanno una certa priorità attraverso la definizione dei gruppi e delle priorità in **tbl_site.group** e **tbl_site.priority**. Si ricordi che i gruppi a cui appartiene un utente sono salvati in un campo di testo in formato lista.

La gestione dell'appartenenza degli utenti ai gruppi come lista di valori separati da "," genera in questa situazione due problemi: il primo, dovuto alla dimensione del campo **tbl_users.u_group** che consente un massimo di 255 caratteri. Significa che la lista dei gruppi assegnati all'utente non può superare questa lunghezza, limitando di fatto il numero



di gruppi assegnabili a ciascun utente; inoltre il numero di gruppi assegnabili varia in funzione della lunghezza dei nomi dei gruppi, variando quindi da utente ad utente.

Il secondo è la gestione delle relazioni tra le tabelle. La query per definire se una pagina di contenuti deve o meno essere visualizzata ad un utente appartenente ad una serie di gruppi implica sia una parte scritta in codice PHP, sia una query che esegue un LIKE sulla tabella **tbl_site**.

In effetti da PHP con la funzione `explode()` si separano i singoli gruppi assegnati all'utente. Lo statement SQL viene generato da PHP con un ciclo che accoda allo statement WHERE una condizione per ogni gruppo assegnato all'utente, in OR fra di loro:

```
SELECT l_id
  FROM tbl_site
 WHERE (l_group LIKE '%,[groupname_1],%' OR l_group LIKE '%,[groupname_2],%')
    AND
    (l_priority <= [userpriority])
```

Nell'esempio (semplificato) è possibile vedere come la query conti uno statement LIKE per ogni gruppo a cui l'utente autenticato nella sessione appartiene. Non viene effettuato nessun JOIN con la tabella `tbl_users` in quanto tutte le informazioni principali dell'utente (codice, login, gruppi, priorità, codice alternativo,...) vengono registrate in variabili di sessione per semplificare le interrogazioni al database.

Esiste la possibilità inoltre di inviare una email agli utenti registrati sul sito ogni volta che un menù viene modificato tramite la funzione "Invia mail agli utenti". Resta ora da definire, una volta assegnate le regole di visualizzazione della pagina, come inserire i contenuti veri e propri. Per procedere all'inserimento:

- selezionare la voce del menù su cui si desidera operare e cliccare su "Contenuto"
- nella finestra "Tipo contenuto" scegliere di:
 - *comporre la pagina attraverso l'Editor di testo HTML (è possibile avvalersi di uno dei*

template di pagina disponibili, selezionando nella casella "Pagina corrente" la voce "Nuova pagina nel template")

- comporre la pagina inserendo direttamente codice HTML*
- inserire un URL per collegarsi ad un sito esterno*
- inserire una form generica*
 - *dovrà essere inserito un destinatario delle informazioni*
 - *è possibile inserire (separati da una virgola) anche più indirizzi e-mail*
 - *è possibile inserire testo HTML prima della form standard*
- importare pagine da un altro sito WebProfessional in modo dinamico (grab)*
- importare pagine da un altro sito web in modo dinamico (grab)*
- fare un link a una query/user action sviluppata nel modulo DbTools (se questo modulo è stato attivato)*
- fare un link a un'applicazione specifica scritta in PHP (custom application), sviluppata per il proprio sito*

Introduzione all'editor

L'Editor WYSIWYG HTML di WebPower consente di gestire pagine HTML con testi formattati, immagini e tabelle. Esistono due versioni dell'editor:

- Rapido (utilizzabile solo con IE 6.0 o superiore)
- Normale (utilizzabile con IE 4.5 o superiore ed altri browser)

A seguire alcuni suggerimenti relativi alle funzioni principali dell'editor online:

- Menu rapido - abilitato tramite pulsante destro del mouse
 - trattasi di una serie di funzioni raggruppate in un menù attivabile premendo il pulsante destro del mouse (menù contestuale)*
- Nuova riga / Nuovo paragrafo

-
- per passare ad una nuova riga premere SHIFT + ENTER*
 - per passare ad un nuovo paragrafo premere solo ENTER*
 - *Inserire una immagine*
 - clickare sull'icona (oppure Destro del mouse - Inserisci Immagine) e selezionarne una tra quelle già presenti sul server o caricarne una a partire da un file presente sul proprio PC (pulsante "Sfoglia")*
 - prima di inserire l'immagine è possibile creare su di essa un link*
 - *premere il pulsante "Link a ..."*
 - *scegliere se fare un collegamento ad una pagina del sito o ad un URL esterno*
 - *Creare un link su una immagine già inserita*
 - selezionare l'immagine (che dovrà apparire tutta scura) e clickare sull'icona (oppure clickare sull'immagine Destro del mouse - Link)*
 - selezionare "link" e inserire l'indirizzo desiderato*
 - *Inserire e gestire una tabella*
 - clickare sull'icona (oppure pulsante destro del mouse - Inserisci Tabella) e impostare colonne, righe, bordi, sfondi, colori, allineamenti del testo, etc.*
 - clickare sulla tabella per reimpostarne le proprietà (pulsante destro del mouse - Proprietà Tabella)*
 - clickare in una casella per gestirne le proprietà (pulsante destro del mouse - Proprietà Casella)*
 - *Eliminare una cella della riga di una tabella*
 - posizionarsi all'interno della cella da eliminare*
 - Destro del mouse - Tabella - Cancella casella*
 - *Aggiungere o eliminare righe o colonne ad una tabella*
 - selezionare la tabella (passare il puntatore del mouse sulle righe della tabella e clickare con il pulsante sinistro del mouse)*

- *cliccare sulle icone secondo ciò che si desidera fare*
- Allineare un testo a fianco di un'immagine
 - *all'atto della scelta dell'immagine, prima di premere "Inserisci" selezionare il tipo di allineamento desiderato (nessuno - sinistra - centro - destra)*

Le funzioni principali dell'editor

L'editor online consente all'utente di generare il contenuto delle pagine in formato HTML utilizzando un editor WYSWYG costituito da una barra degli strumenti con icone simili a quelle utilizzate nei più conosciuti word processor ed una zona di editing. L'utente ha la possibilità di "scrivere" la pagina in modo simile ad un documento di Word per poi inserire il contenuto html generato in modo automatico dall'editor nel campo **I_content** della tabella **tbl_site**. Di seguito elencate le funzionalità messe a disposizione attraverso la barra delle icone nella parte superiore dell'editor:

Icona	Nome	Funzionalità
	Taglia	Taglia
	Copia	Copia
	Incolla	Incolla
	Annulla	Annulla operazione effettuata
	Ripeti	Ripristina operazione effettuata
	Inserisci Immagine	Inserisci immagine
	Inserisci Tabella	Inserisci tabella
	Inserisci Riga	Inserisci riga
	Inserisci Colonna	Inserisci colonna
	Cancella riga	Elimina righe
	Cancella Colonna	Elimina colonne

Icona	Nome	Funzionalità
	Link	Collegamento ipertestuale
	Ancora	Segnalibro
	Elenco numerato	Elenco numerato
	Elenco puntato	Elenco puntato
	Visualizza margini	Rende visibili i contorni di tabelle con bordo = "0"
	Edita codice Html	Visualizza e permette di modificare il codice HTML della pagina
	Opzione	Imposta colore di sfondo alla pagina
	Help	Aiuto
	Grassetto	Grassetto
	Corsivo	Corsivo
	Sottolineato	Sottolineato
	Colore testo	Colore testo
	Colore sfondo	Colore sfondo
	Apice	Apice
	Pedice	Pedice
	Giustifica a sinistra	Allinea a sinistra
	Giustifica al centro	Centra
	Giustifica a destra	Allinea a destra
	Giustifica tutto	Allinea a sinistra e a destra
	Rientro	Aumenta rientro
	Rientro negativo	Riduci rientro

eNews

Questo modulo permette di avere nel proprio sito una apposita area adibita esclusivamente alla visualizzazione di notizie o documenti a scopo informativo (partecipazione a

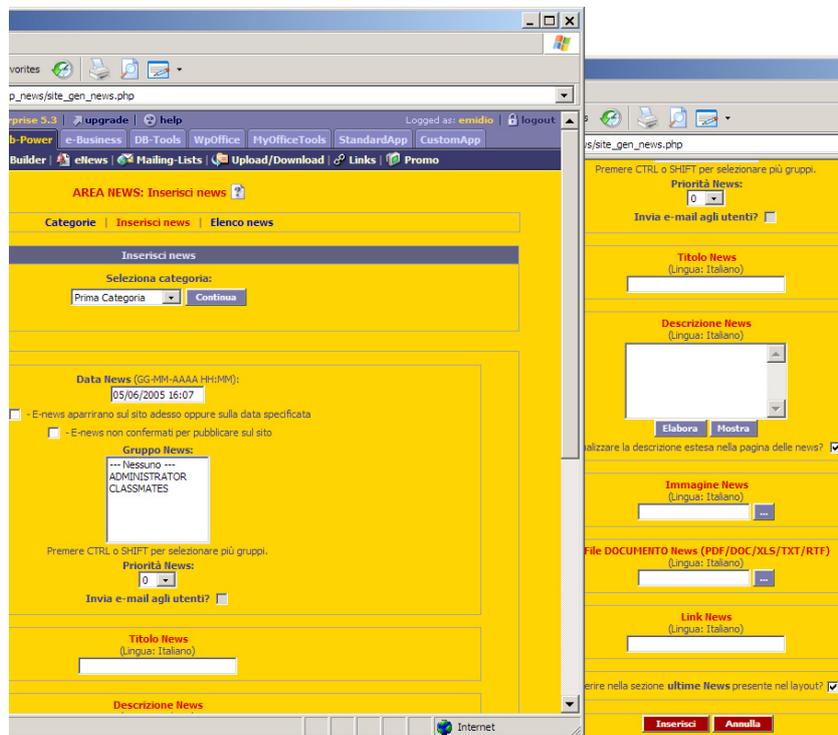
fiere o eventi, nuovi prodotti, novità tecniche, promozioni, ecc..). Il modulo si divide in tre sezioni principali.

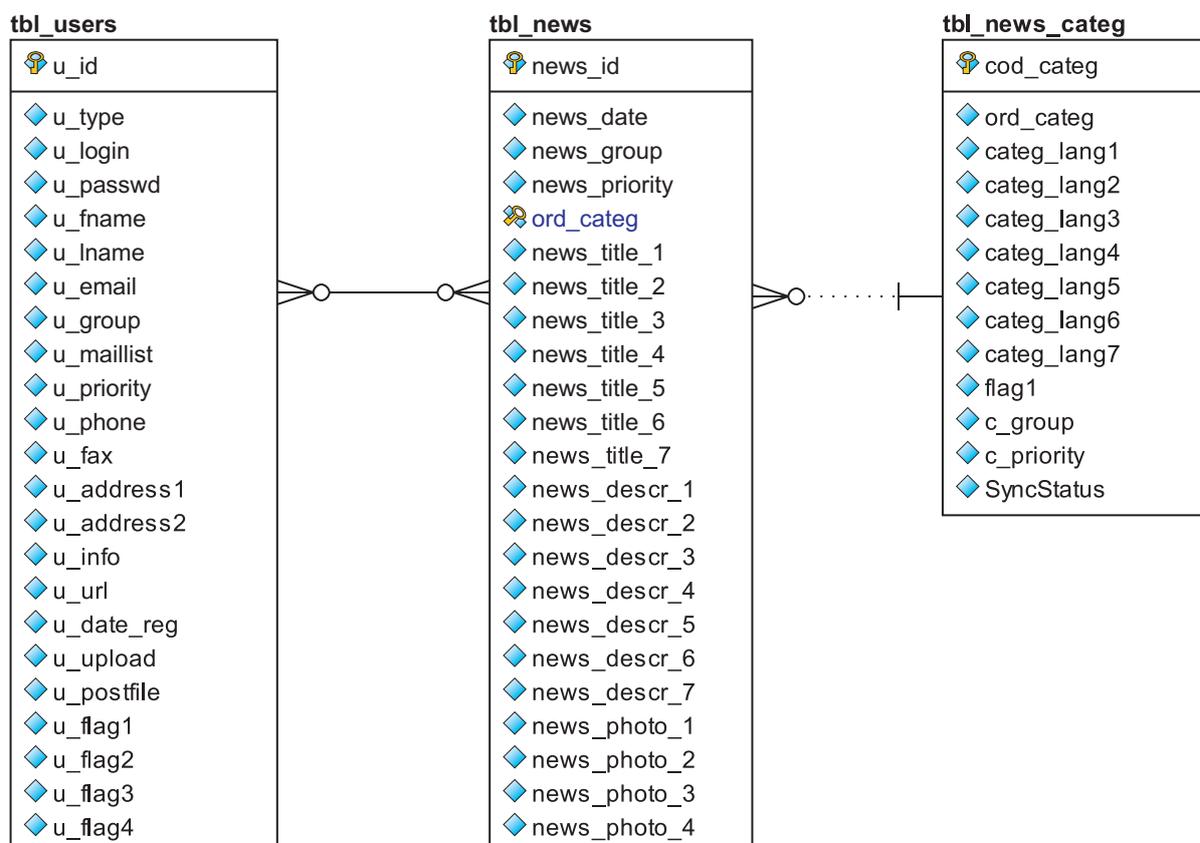
Categorie

Le notizie sono suddivise per categorie (definibili a cura dell'Amministratore del sito). Per ogni categoria verrà visualizzata una pagina contenente l'elenco dei documenti associati. Per creare una nuova categoria cliccare sul tasto "Inserisci" selezionando la posizione desiderata. La nuova categoria comparirà nell'elenco con titolo "???". Il link "Modifica" consente di rinominarla mentre il bottone "Aggiungi nuova sottocategoria" aggiunge categorie di livello inferiore collegate alla categoria selezionata.

Inserisci News

Nella sezione Inserisci News è possibile inserire le informazioni riguardanti la notizia o il documento da pubblicare (data, titolo, corpo, immagini ed allegati, link, se una notifica sarà inviata agli utenti registrati, priorità, gruppo associato, se le news sono confermate per essere pubblicate sul sito). I permessi sull'accesso alle informazioni utilizzano i gruppi e la priorità (non i singoli utenti). Le informazioni sono memorizzate nelle tabelle **tbl_news** e **tbl_news_categ**.





Struttura delle due tabelle che gestiscono il modulo news di WebPower. La relazione con la tabella utenti **tbl_users** in realtà avviene attraverso variabili di sessione impostate dalla procedura di login. Come abbiamo già visto, la relazione si basa su una SELECT/LIKE in questo caso sul campo **tbl_news.news_group**. La relazione tra categorie di news e news è invece correttamente impostata sul codice categoria **tbl_news_categ.cod_categ = tbl_news.ord_categ**, relazione non-identifying one to zero or more (**ord_categ** è quindi foreign key ma non fa parte della primary key della tabella **tbl_news**).

Elenco News

In questa sezione vengono elencate le news inserite, con possibilità di filtro per categoria. Attraverso i menù di riga le news vengono visualizzate, modificate ed eliminate.

Mailing-Lists

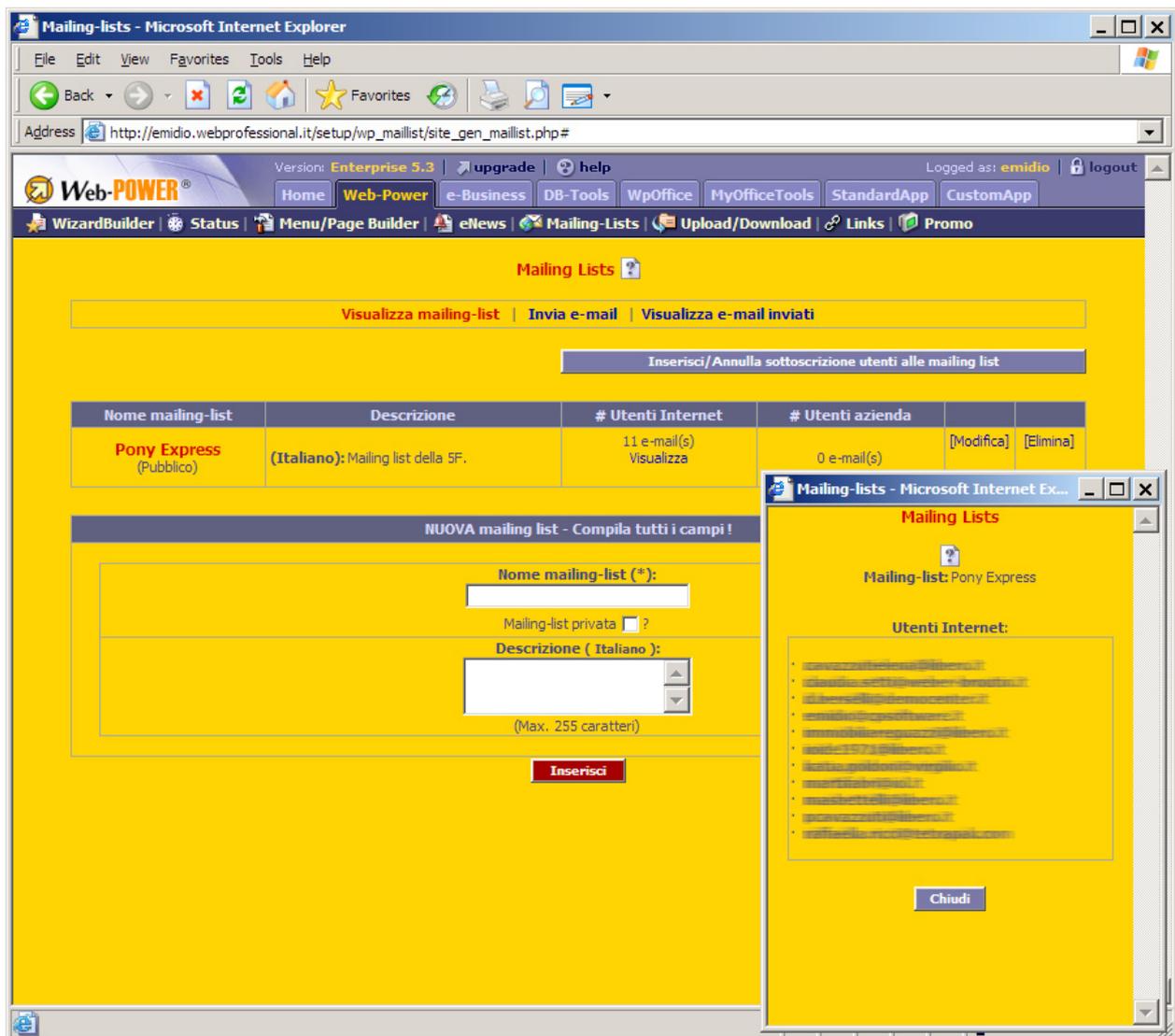
La mailing list è uno strumento di comunicazione e di informazione tra azienda ed utenti. La funzione permette di creare diverse mailing list o categorie di informazione, alle quali gli utenti possono iscriversi direttamente dal sito o esservi iscritti dall'amministratore del

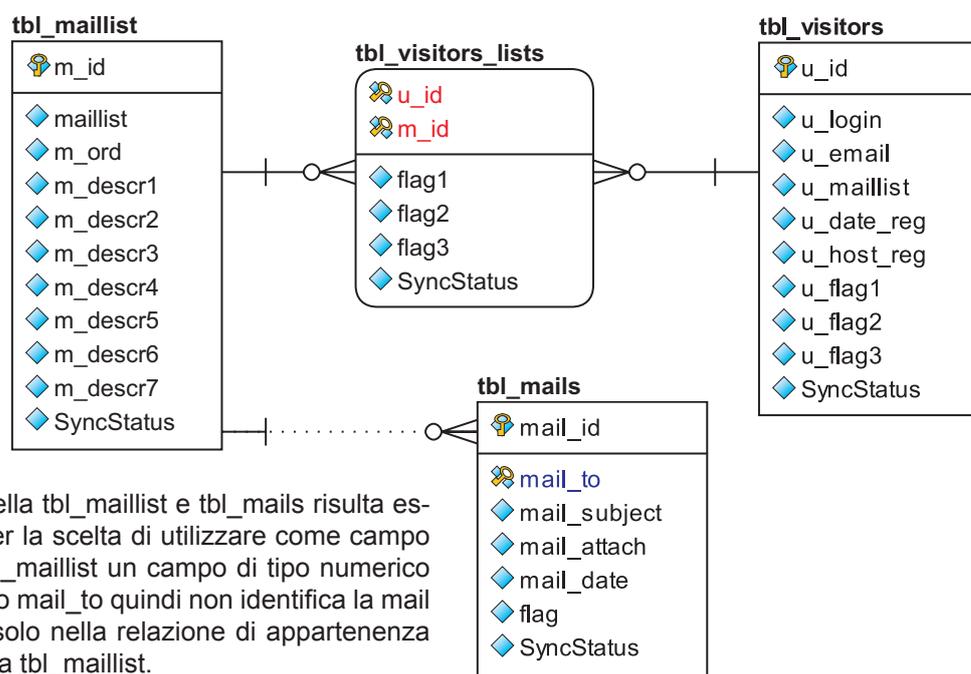
sistema, così da ricevere tutte le informazioni che l'azienda divulgherà sull'argomento selezionato (è attiva anche una funzione che consente di cancellarsi automaticamente dalla mailing list).

Visualizza mailing list elenca tutte le mailing list attivate e i dati relativi agli utenti registrati. Dalla stessa pagina si possono inoltre creare nuove mailing list.

Invia mail è la funzione per inserire i testi di email e comunicazioni, con possibilità di allegare fino ad un massimo di tre file.

Visualizza e-mail inviate elenca le email inviate e altri informazioni (data, destinatari, og-





La relazione tra la tabella **tbl_maillist** e **tbl_mails** risulta essere nonidentifying per la scelta di utilizzare come campo chiave primaria per **tbl_maillist** un campo di tipo numerico autoincrementato. Il campo **mail_to** quindi non identifica la mail inviata ma interviene solo nella relazione di appartenenza alla mailinglist in tabella **tbl_maillist**.

getto, allegati): viene mantenuto uno storico delle mail inviate. La relazione tra la mailing list (**tbl_maillist**) e le mail inviate (**tbl_mails**) avrebbe dovuto riportare la chiave **tbl_maillist.m_id** come foreign key su **tbl_mails.mail_to**. In fase di programmazione, come già accaduto per la relazione tra la tabella utenti ed i gruppi, venne riportato dal programmatore il campo **tbl_maillist.maillist**. Conseguentemente fu necessario introdurre un controllo sul nome della mailing list per evitare inconsistenze nel database.

Upload/Download

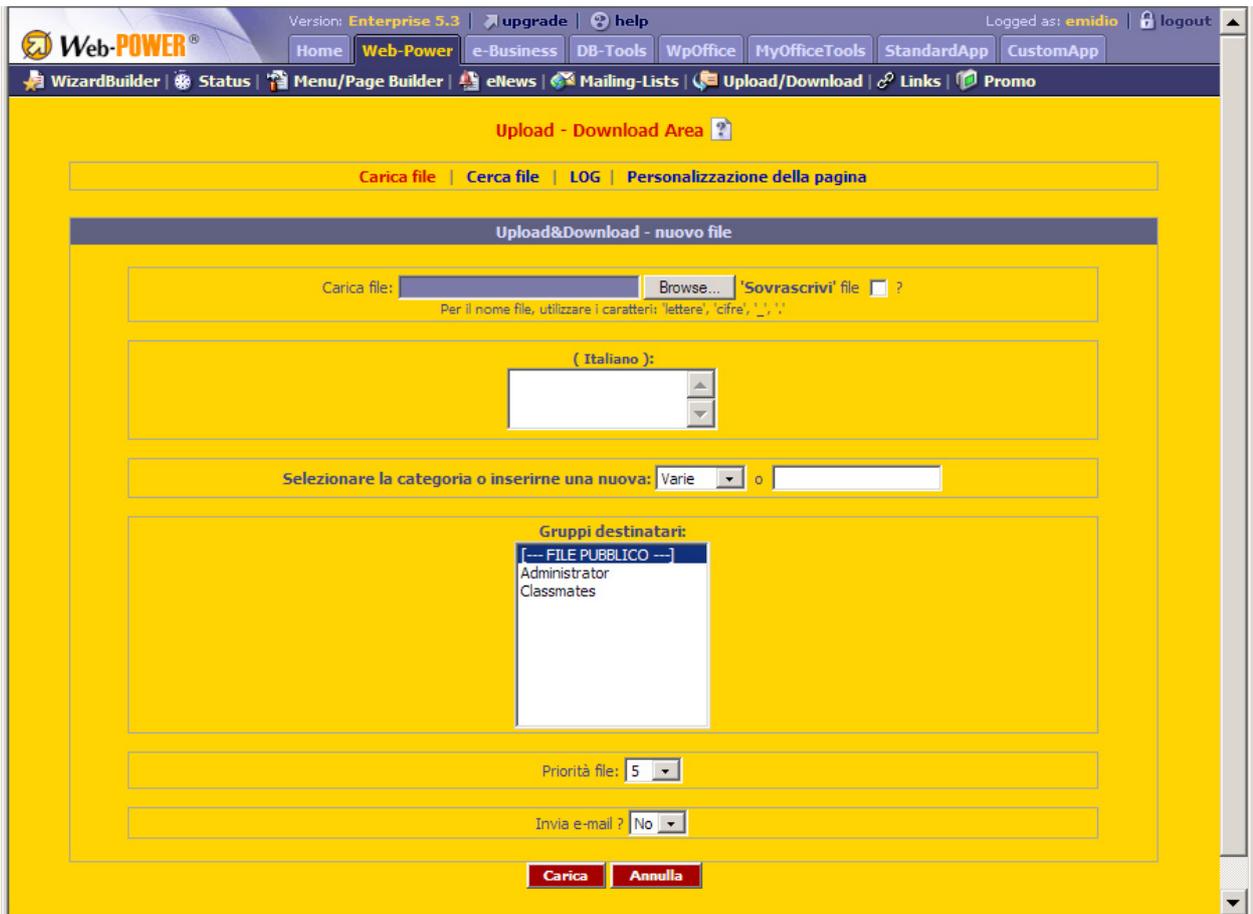
Questo modulo permette di rendere disponibili, in una apposita area del sito, file scaricabili dagli utenti. Le funzioni del modulo sono elencate di seguito.

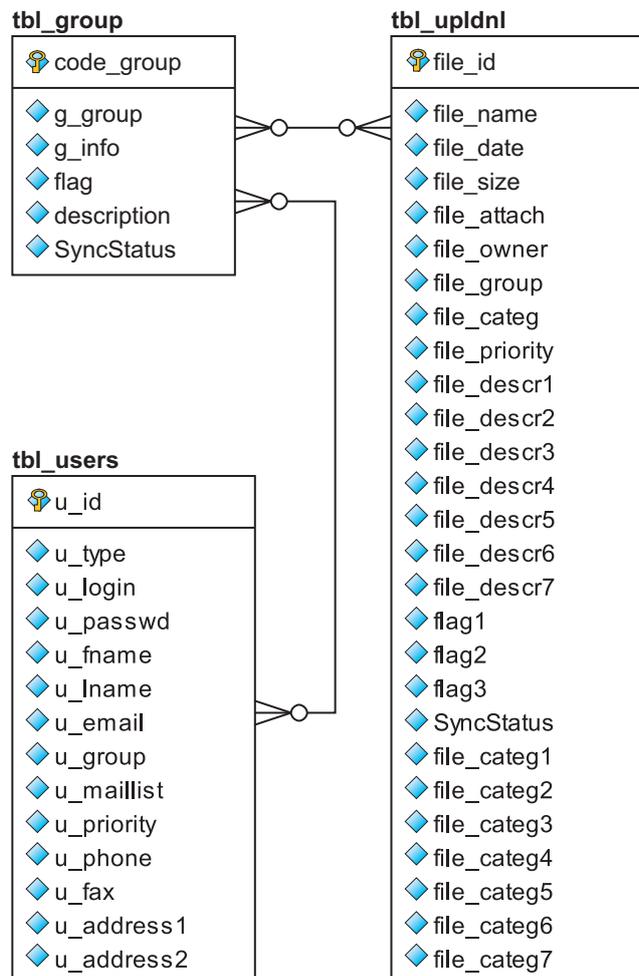
Carica file per memorizzare nel sito file che gli utenti (in virtù dei rispettivi privilegi) potranno scaricare. Ciascun file è associato ad una breve descrizione, ad una categoria di appartenenza ed è possibile renderlo pubblico solo a certi utenti e in ragione del grado di priorità e, infine, è possibile inviare via email una notifica a tutti gli utenti che appartengono

al gruppo a cui è destinato il file.

Cerca file per modificare i dati relativi a file già caricati sul sito (priorità, descrizione, categoria, data ecc.). Viene mantenuto un log dei file scaricati (utente, file e data di download) ed è possibile personalizzare la testata ed il piede della pagina di download (parti superiore ed inferiore alla lista dei file). Nell'interfaccia pubblica è presente una maschera di ricerca che consente di selezionare i file di interesse, attiva solo quando il numero dei file caricati è maggiore di dieci. La ricerca avviene per categoria, per parola contenuta nel nome file o nella descrizione, per data di pubblicazione con possibilità di ordinamento per nome file, data (ascendente o discendente) e dimensione del file.

Tutte le informazioni relative al modulo upload/download sono memorizzate nella tabella **tbl_upldnl**.





Struttura dati che memorizza le informazioni della sezione Upload/Download. L'attribuzione dei permessi di visibilità avviene attraverso i gruppi di utenti e non direttamente sugli utenti stessi.

Download (n file disponibili : 10)

Ricerca: Tutte le parole

La ricerca verrà effettuata all'interno del **nome file e descrizione**.
[Dettagli](#)

Categorie: Offerte

Intervallo di tempo da: a (Formato della data: gg/mm/aaaa)
 (ricerca su data di inserimento):

Ordinare i risultati per: Data (disc) Ricerca

3 risultati. (1 : 3)

#	Categorie	Data	File	Info	
1.	Offerte	20-05-2005 09:12	obi_050519.pdf (5725 KB)	Offerte OBI dal 19/5 all'11/6	
2.	Offerte	28-03-2005 06:40	obi_050218.pdf (4857 KB)	Offerte OBI giardinaggio Febbraio-Marzo (dal 17/2 al 5/3)	
3.	Offerte	03-02-2005 00:00	obi_050201.pdf (4126 KB)	Offerte OBI Febbraio	

Interfaccia utente della sezione Upload/Download - nella parte superiore la maschera di ricerca (attiva quando il numero di file caricati è superiore a dieci), nella parte inferiore i risultati di una ricerca.

Come in tutte le altre tabelle la relazione di visibilità di un file ad un determinato gruppo di utenti è demandata alla descrizione del gruppo e non al suo codice. Inoltre è interessante notare che viene utilizzata quest'unica tabella anche per gestire le descrizioni in lingua dei file, attraverso i campi **file_descr1**, **file_descr2**,... e **file_categ1**, **file_categ2**,...

Di fatto questo stravolgimento del progetto iniziale del database ha condotto al limite della piattaforma di gestire un massimo di sei lingue contemporanee, a meno di non variare la struttura delle tabelle. Il disegno iniziale prevedeva una seconda tabella con le descrizioni in lingua dei file con chiave **file_id**, **codice_lingua** (**codice_lingua** avrebbe dovuto contenere la singla internazionale a due caratteri di identificativo di nazione) con **file_id** foreign key.

Inoltre non esiste una tabella categorie per i file del modulo Upload/Download.

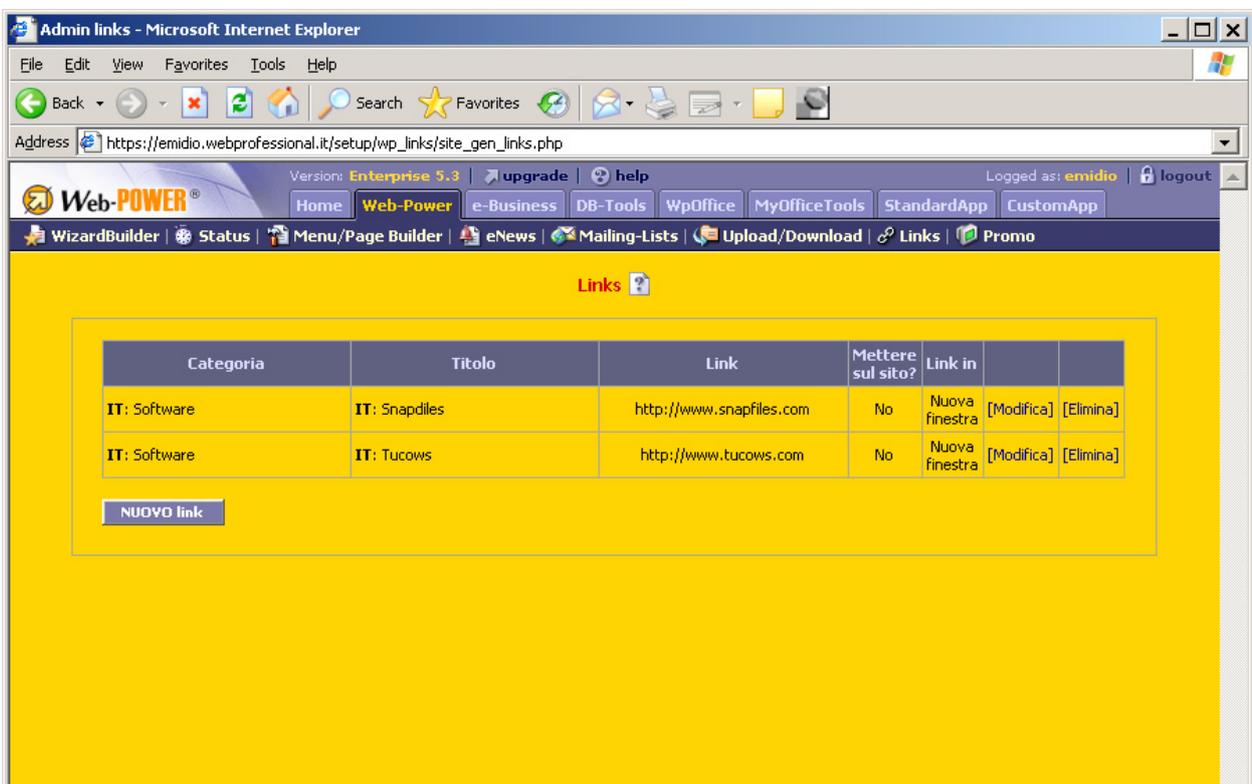
L'elenco delle categorie è ottenuto tramite una interrogazione al database:

```
SELECT DISTINCT file_categ
FROM tbl_uplndl
ORDER BY file_categ ASC
```

Le performance degradano con l'aumentare del numero di file caricati, affliggendo tutte le maschere in cui viene visualizzato l'elenco delle categorie in controlli di tipo <SELECT>.

Links

Modulo per inserire/modificare/cancellare link utili per la navigazione in internet degli utenti. Inserire in questa sezione, catalogati per categorie (da creare a cura dell'amministratore) indirizzi internet utili. E' possibile nascondere l'URL e visualizzare solo il titolo mediante l'opzione presente in SETUP - Configure - sez. Visualizza / Nascondi gli oggetti del sito. La gestione delle categorie di link e delle lingue e del tutto simile a quella utilizzata per Upload/Download, con il limite delle sei lingue e le scarse performance nella query per la



Admin links - Microsoft Internet Explorer

Address: https://emidio.webprofessional.it/setup/wp_links/site_gen_links.php

Version: Enterprise 5.3 | upgrade | help | Logged as: emidio | logout

Home | Web-Power | e-Business | DB-Tools | WpOffice | MyOfficeTools | StandardApp | CustomApp

WizardBuilder | Status | Menu/Page Builder | eNews | Mailing-Lists | Upload/Download | Links | Promo

Links

Categoria	Titolo	Link	Mettere sul sito?	Link in		
IT: Software	IT: Snapfiles	http://www.snapfiles.com	No	Nuova finestra	[Modifica]	[Elimina]
IT: Software	IT: Tucows	http://www.tucows.com	No	Nuova finestra	[Modifica]	[Elimina]

visualizzazione dell'elenco delle categorie.

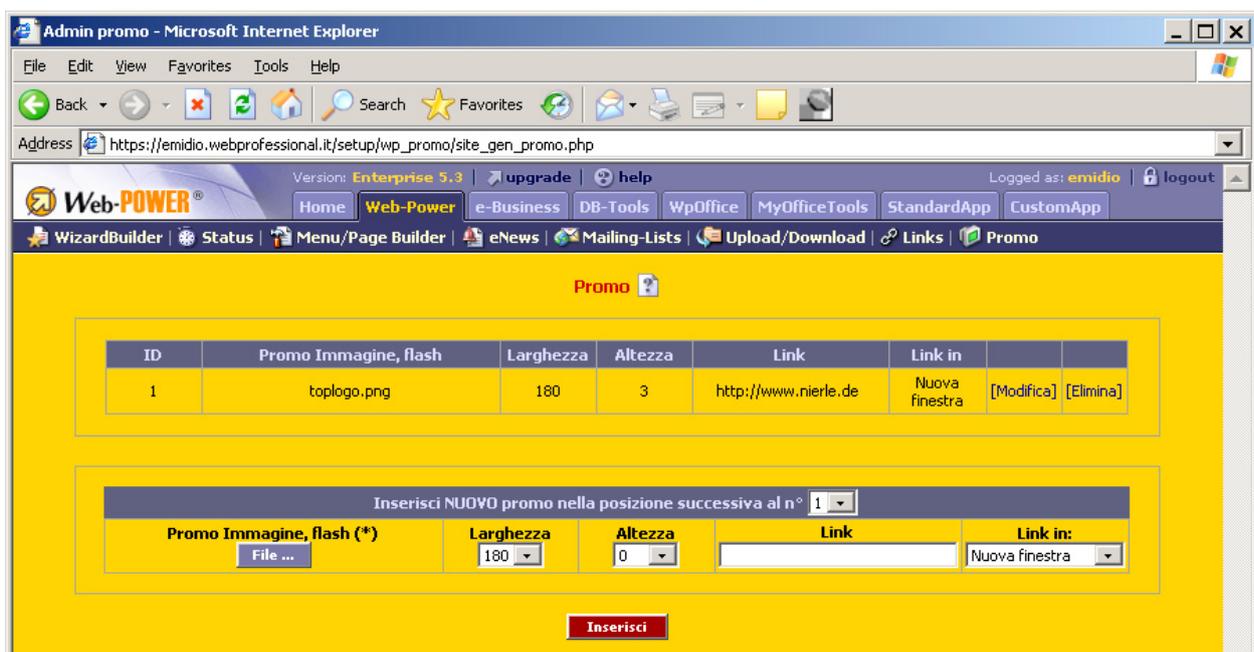
Promo

In questa sezione è possibile caricare immagini promozionali con relativi link che saranno esposti nell'omonima sezione del sito (aziende collegate, schede prodotto, offerte, servizi, ecc.). L'unica tabella interessata è **tbl_promo**, con chiave primaria intero con auto incremento, senza alcuna relazione con altre tabelle. Questa sezione semplicemente visualizza una immagine in una apposita sezione della pagina, memorizzandone il percorso, le dimensioni ed il link di destinazione associato in **tbl_promo**.

tbl_links

 links_id
 links_categ
 links_categ1
 links_categ2
 links_categ3
 links_categ4
 links_categ5
 links_categ6
 links_categ7
 links_title
 links_title1
 links_title2
 links_title3
 links_title4
 links_title5
 links_title6
 links_title7
 links_link
 links_priority
 flag
 SyncStatus

Tutte le informazioni relative ai link sono memorizzate in una unica tabella **tbl_links**.



The screenshot shows the 'Admin promo' interface in Microsoft Internet Explorer. The page title is 'Admin promo - Microsoft Internet Explorer'. The address bar shows the URL: https://emidio.webprofessional.it/setup/wp_promo/site_gen_promo.php. The interface includes a navigation menu with options like Home, Web-Power, e-Business, DB-Tools, WpOffice, MyOfficeTools, StandardApp, and CustomApp. The main content area is titled 'Promo' and features a table with the following data:

ID	Promo Immagine, flash	Larghezza	Altezza	Link	Link in		
1	toplogo.png	180	3	http://www.nierle.de	Nuova finestra	[Modifica]	[Elimina]

Below the table, there is a form to insert a new promo. The form includes a dropdown menu to select the position (currently set to '1'), and input fields for 'Promo Immagine, flash (*)', 'Larghezza' (set to 180), 'Altezza' (set to 0), 'Link', and 'Link in' (set to 'Nuova finestra'). An 'Inserisci' button is located at the bottom of the form.

Conclusioni

Il modulo WebPower sostanzialmente fornisce tutte le funzionalità necessarie ad amministrare e mettere online i contenuti relativi alle sezioni abituali presenti su un sito web aziendale (chi siamo, dove siamo, servizi, news, contatti). La gestione dei contenuti può essere demandata ad amministratori di secondo livello e la visualizzazione delle pagine e delle informazioni è soggetta al sistema di regole e permessi assegnati ad utenti e gruppi della piattaforma WebProfessional.

eBusiness

Il modulo eBusiness nasce dall'esigenza di venire incontro alla crescente domanda delle aziende di una soluzione completa di commercio elettronico, integrata ai servizi offerti attraverso il sito internet.

La ricerca rileva il pieno sviluppo della domanda di banda larga e l'offerta di contenuti e servizi online, la crescita del business-to-business (B2B), l'importanza che la rete assume anche nella pubblica amministrazione, la diffusione dell'e-learning sono fenomeni che attraversano, in misura diversa, sia le grandi che le piccole e medie imprese. Dopo il boom indiscriminato dell'anno 2000 si assiste ad un approccio più maturo commercio elettronico ed ai servizi erogati attraverso il web.

Nel primo semestre del 2003 tutte le aziende si sono connesse ad internet; una saturazione che comprende anche le aziende più piccole (da 10 a 19 addetti) e che tocca persino le imprese edili, fino a qualche tempo prima lontane dall'interessamento al web. Le piccole e medie imprese che si collegano al web sfruttando un dial up analogico è in costante diminuzione. L'apertura di nuovi siti sembra bloccata, ma c'è da rilevare che ormai due aziende su tre e, tra queste, la quasi totalità di quelle con più di 100 addetti, dispongono di un sito.

Le aziende sono ora orientate allo sviluppo delle funzionalità e dei servizi offerti tramite internet. E' ormai superata la visione del sito come un semplice strumento di presentazione della società, una vetrina su Internet: aumentano, infatti, la presenza di cataloghi online, di servizi al cliente, di webstore aziendali, e aumenta, nel contempo, la percezione del sito come strumento per la vendita.

La maggioranza delle imprese opta per la gestione esterna del sito rivolgendosi a web agency e ad altri fornitori (i fornitori del gestionale, per esempio), mentre perde peso nella gestione e manutenzione del sito l'agenzia di pubblicità, incapace di fornire le soluzioni tecniche avanzate necessarie all'offerta dei servizi richiesti e di integrare il sito in modo dinamico ai processi aziendali.

I motivi dietro questa forte tendenza sono soprattutto di tipo economico (la riduzione dei costi), mentre in secondo piano passano gli aspetti qualitativi, legati all'efficienza complessiva del processo di offerta ed acquisto e di gestione. Le attività di ecommerce più gettonate sono gli ordini online (B2B e B2C), la personalizzazione dell'offerta, la tracciabilità dell'ordine.

Primo sviluppo - Catalogo OnLine

La prima versione di eBusiness, non ancora integrato nella piattaforma WebProfessional e sviluppato in Cold Fusion, deriva la struttura dati direttamente da un sottoinsieme semplificato delle tabelle del gestionale Business, su database MS Access.

L'utilizzo di MS Access fu dettato dalla necessità di ridurre i tempi di sviluppo e di avere un database su piattaforma Windows (CP Software nel 1999 sviluppava solo su piattaforma Windows/Cold Fusion per gli applicativi web) facilmente aggiornabile dagli utenti che utilizzavano il gestionale Business (basato sempre su MS Access).

Derivando la struttura dati dallo stesso database di Business era possibile per gli utenti caricare direttamente (via FTP) un file .mdb per pubblicare articoli e prezzi su web per l'acquisto online o la semplice visualizzazione come catalogo.

Questo permetteva di non dover investire risorse per gestire l'integrazione dati dal database cliente a quello online, riducendo costi e tempi di sviluppo ed offrendo un prodotto

completamente integrato con Business e di facile utilizzo.

Tramite alcune macro in MS Access venivano esportate solo le tabelle ed i record necessari (inutile infatti mantenere nelle anagrafiche clienti - tabella **anagra** campo **tipo**='C' - anche le anagrafiche fornitori - campo **tipo**='F') riducendo considerevolmente le dimensioni del file e quindi i tempi di upload.

Utilizzando l'apposito campo di Business era inoltre possibile gestire le password di accesso e generare (tramite macro) la tabella di supporto necessaria per l'autenticazione degli utenti in eBusiness. Le prime esperienze di commercio elettronico infatti erano esclusivamente rivolte al Business to Business, per facilitare le operazioni di ordine a clienti abituali e consolidati - questo eliminava la necessità di procedure relative al pagamento con carta di credito ed alla registrazione degli utenti online. Le problematiche logistiche ed organizzative legate alla acquisizione di nuovi clienti online, gestione delle spedizioni e procedure di pagamento via carta di credito rappresentavano infatti un forte deterrente per la piccola e media impresa verso il passo del Business to Consumer (B2C).

La soluzione adottata, se pur soddisfacente per il primo utilizzo di eBusiness, presentava comunque diversi svantaggi per lo sviluppo futuro ed alcune pecche di funzionalità.

Innanzitutto l'aggiornamento semplice tramite FTP del database rendeva indisponibili le pagine per tutto il tempo dell'upload. Nonostante la procedura avvenisse comunque in ore notturne, con accessi praticamente nulli al sito ed al di fuori degli orari lavorativi (ricordiamo che si operava principalmente in B2B), a volte il mancato buon fine dell'upload (dovuto a problemi di linea) o le eccessive dimensioni del database costringevano l'utente ad un intervento manuale sulle procedure durante l'accesso al sito da parte dei clienti all'insorgere del problema e ad una indisponibilità del sito fino al termine del processo.

Inoltre l'appoggiarsi alle strutture dati consolidate di Business, se da un lato semplificava la messa online dell'e-commerce per gli utenti che utilizzavano questo gestionale, dall'altro

limitava il mercato agli stessi, rendendo difficili le integrazioni con database di gestionali diversi. La soluzione più comune era creare procedure per la trasformazione dei dati dal database cliente (Oracle, DB2, SQL Server,...) alla struttura Business in MS Access, per poi effettuare l'upload. Procedure spesso fortemente limitative in quanto la struttura di Business non consentiva di utilizzare metodi ad essa estranei (ad esempio sulla gestione prezzi e listini, il limite dei livelli di categorie a due,...).

Inoltre i costi delle licenze di Windows e Cold Fusion rendevano l'applicazione meno appetibile dal punto di vista economico avendo peso pari se non superiore a quello di sviluppo, integrazione e personalizzazione.

Il passaggio a PHP e l'integrazione con WebPower

La decisione di riscrivere eBusiness in PHP come modulo della piattaforma di WebProfessional consentì di reingegnerizzare il prodotto sulla base delle precedenti esperienze.

Il costo di un porting era quasi equiparabile a quello dello sviluppo ex novo del modulo, considerando che buona parte dell'analisi delle funzionalità era già sviluppata sulla base delle richieste di modifiche all'applicativo esistente.

Il database venne completamente ridisegnato cercando di renderlo quanto più flessibile ed adattabile alle diverse tipologie di strutture utilizzate dai maggiori programmi gestionali e progettando una interfaccia online che permettesse di gestire in modo completo ed autonomo l'applicativo anche senza importazione dati da altro database.

La struttura venne analizzata e progettata nel corso di due settimane di lavoro nella sede di Iasi (Romania) coinvolgendo sia personale con competenze sull'utilizzo del software gestionale sia alcuni programmatori che avrebbero poi dovuto seguire lo sviluppo dei vari moduli.

La configurazione iniziale

eBusiness possiede una completa interfaccia di amministrazione che consente di operare sia sui dati che sulle funzionalità del modulo. La scelta di poter operare anche sui dati oltre che sulle impostazioni è dettata dalla necessità di alcune aziende di poter gestire prodotti su commessa oppure un numero limitato di prodotti specifici senza agganciarsi al database aziendale (con informazioni e descrizioni ad esclusivo uso interno) creando ex nuovo i dati direttamente online. L'area di setup del modulo eBusiness è suddivisa nelle sezioni descritte a seguire.

Setup (Layout)

Il menu Setup - Layout consente di impostare i vari elementi grafici di visualizzazione nell'area pubblica, determinando la disposizione degli oggetti sulla pagina e le informazioni da pubblicare. E' possibile definire:

- il template per la lista di prodotti di una categoria
- il template per i dettagli di un prodotto
- le impostazioni relative al testo
- la modalità di visualizzazione delle parti di ricambio, degli accessori e degli articoli correlati
- la modalità di visualizzazione delle varianti

Visualizzazione lista prodotti

Sono presenti sette diversi template per la visualizzazione della lista prodotti. Ulteriori template possono essere aggiunti su richieste specifiche dell'utente (implica la scrittura di moduli in PHP). I primi sei template visualizzano quattro prodotti per pagina e contengono:

- l'immagine per la lista prodotti
- il codice prodotto
- il titolo del prodotto (denominazione)
- il prezzo e gli sconti
- la quantità da aggiungere al carrello
- una parte della descrizione prodotto

Alcuni elementi di visualizzazione nella lista prodotti possono essere nascosti, in dipendenza delle impostazioni nel menu di configurazione. Il settimo template, aggiunto successivamente, è una versione compatta senza immagine per la visualizzazione di numerosi prodotti nella stessa schermata - è un esempio di come una richiesta specifica possa essere introdotta nello standard a beneficio di tutti gli utilizzatori della piattaforma.

Visualizza dettagli prodotti

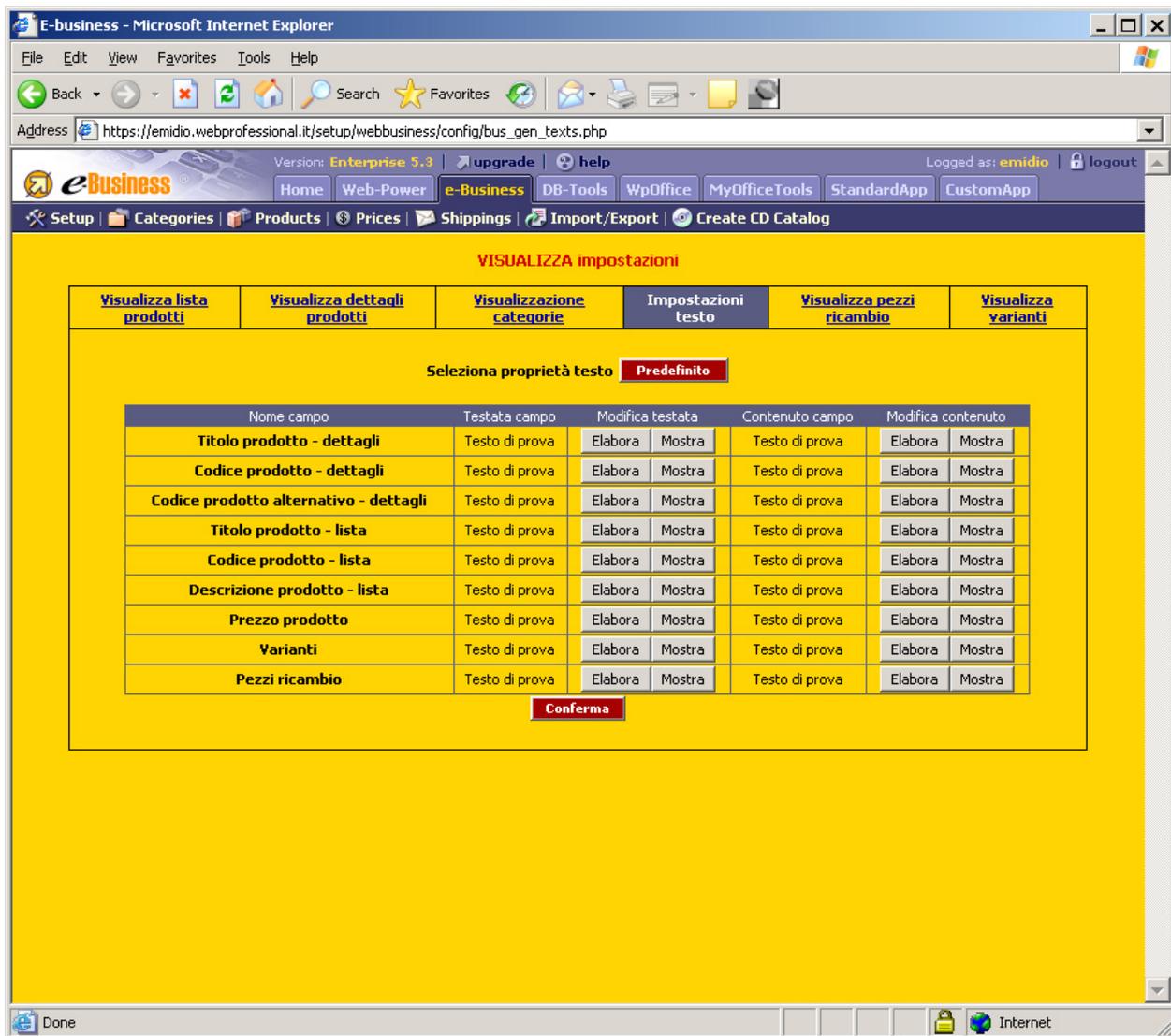
Da questo menù si imposta il layout del dettaglio prodotto tra sei disponibili. Eventuali nuovi template possono essere aggiunti come per la lista prodotti.

Visualizza Categorie

I prodotti possono essere suddivisi in categorie. In questa sezione vengono configurati il tipo e la dimensione del carattere di visualizzazione del testo delle categorie, che compaiono sulla sinistra della pagina con struttura gerarchica espandibile.

Impostazioni testo

Questa sezione permette la configurazione delle proprietà del testo per gli elementi principali del sito: codice prodotto, titolo prodotto (sia nella lista che nei dettagli prodotti), descrizione estesa, prezzo, testata, varianti e accessori/parti di ricambio. La modifica di



VISUALIZZA impostazioni

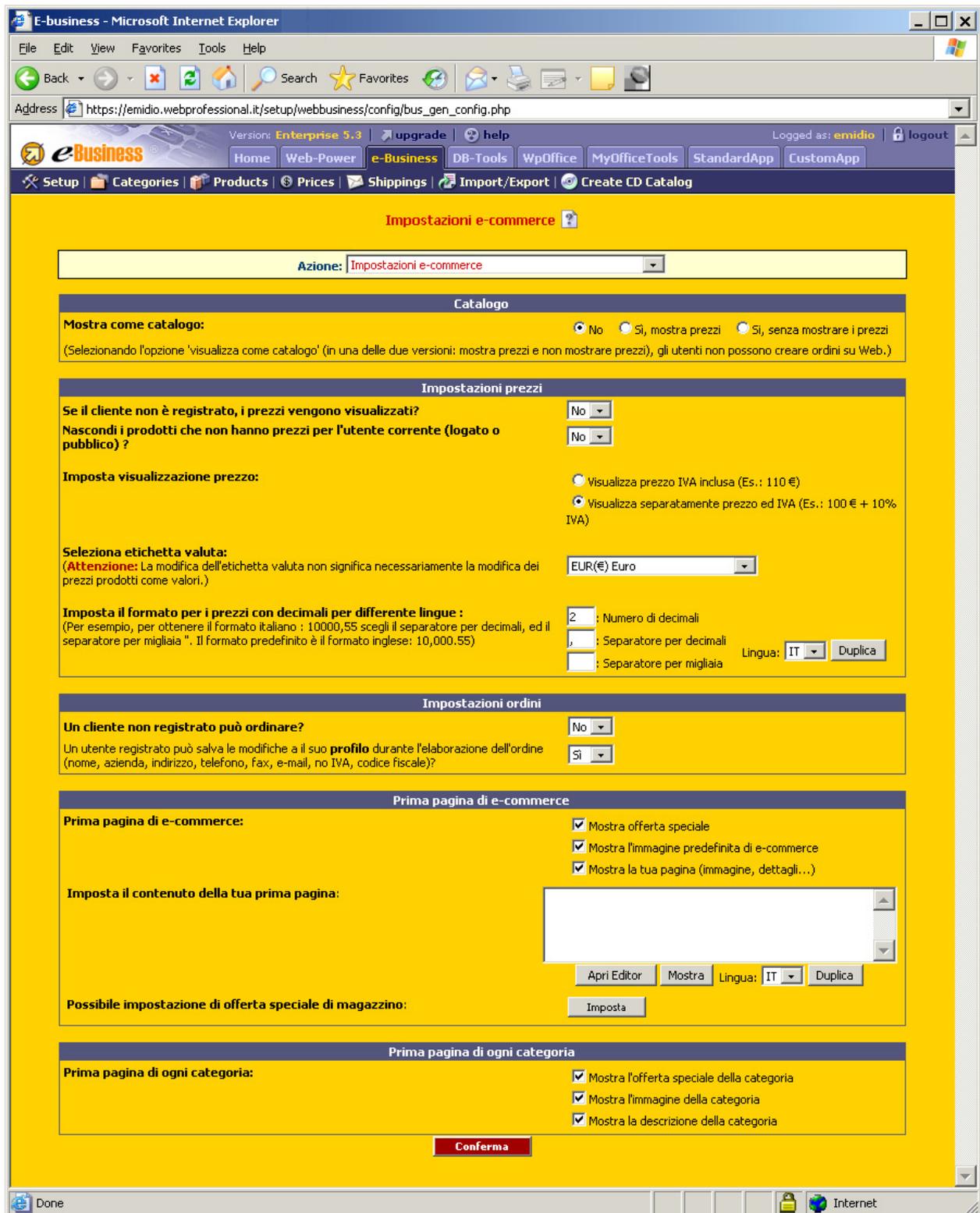
[Visualizza lista prodotti](#) [Visualizza dettagli prodotti](#) [Visualizzazione categorie](#) **Impostazioni testo** [Visualizza pezzi ricambio](#) [Visualizza varianti](#)

Selezione proprietà testo **Predefinito**

Nome campo	Testata campo	Modifica testata		Contenuto campo	Modifica contenuto	
Titolo prodotto - dettagli	Testo di prova	Elabora	Mostra	Testo di prova	Elabora	Mostra
Codice prodotto - dettagli	Testo di prova	Elabora	Mostra	Testo di prova	Elabora	Mostra
Codice prodotto alternativo - dettagli	Testo di prova	Elabora	Mostra	Testo di prova	Elabora	Mostra
Titolo prodotto - lista	Testo di prova	Elabora	Mostra	Testo di prova	Elabora	Mostra
Codice prodotto - lista	Testo di prova	Elabora	Mostra	Testo di prova	Elabora	Mostra
Descrizione prodotto - lista	Testo di prova	Elabora	Mostra	Testo di prova	Elabora	Mostra
Prezzo prodotto	Testo di prova	Elabora	Mostra	Testo di prova	Elabora	Mostra
Varianti	Testo di prova	Elabora	Mostra	Testo di prova	Elabora	Mostra
Pezzi ricambio	Testo di prova	Elabora	Mostra	Testo di prova	Elabora	Mostra

Conferma

ognuno di questi elementi avviene tramite un click sul pulsante "Elabora" a fianco della descrizione, che apre una finestra popup da cui è possibile impostare: il tipo carattere, il colore, la dimensione, grassetto, corsivo, sottolineato o barrato. Nel caso in cui si lavori con browser Internet Explorer oppure Mozilla la modifica del testo viene visualizzata all'istante dopo la conferma delle impostazioni. Per gli altri browser le modifiche sono visibili premendo il pulsante "Mostra". Il pulsante "Predefinito" reimposta le proprietà del testo delle varie sezioni ai valori di default associati al template selezionato (sia per lo stile che per i colori).



La schermata delle impostazioni principali del modulo eBusiness. Il numero di parametri e valori lascia trasparire l'estrema flessibilità e possibilità di configurazione del modulo, per venire incontro alle esigenze specifiche del singolo utente ed imposte dal tipo di prodotti.

Visualizza pezzi ricambio

Configura il tipo di visualizzazione delle parti di ricambio/accessori scegliendo se tipo tabella o lista oltre che al tipo e formato del font.

Visualizza varianti

Configura il tipo di visualizzazione per l'elenco delle varianti. Oltre alla visualizzazione in formato tabella ed elenco, è possibile utilizzare il configuratore varianti descritto nei paragrafi successivi.

Setup (Configure)

Questa sezione consente di definire sia le impostazioni relative alla modalità di visualizzazione che i dettagli tecnici relativi agli ordini ed parametri di amministrazione. Tutte le informazioni di configurazione sono memorizzate nella tabella [tbl_config_ecomm](#).

Impostazioni eCommerce

La schermata dà accesso alle impostazioni generali del modulo sia relativamente all'operatività (catalogo, B2B, B2C) sia per i parametri di presentazione generali delle pagine.



Dalla sezione **Catalogo** si definisce la possibilità di ordinare o meno dal catalogo prodotti. Le opzioni disponibili sono di mostrare gli articoli senza prezzo e senza possibilità di ordinare (anche per gli utenti registrati) rendendo di fatto il modulo un catalogo online, con prezzo e senza possibilità di ordinare (anche per gli utenti registrati), con prezzo e con possibilità di ordinare (utilizzo in modalità ecommerce, attivabile in modalità B2B per i soli utenti registrati o B2C).

Impostazioni prezzi	
Se il cliente non è registrato, i prezzi vengono visualizzati?	No
Nascondi i prodotti che non hanno prezzi per l'utente corrente (logato o pubblico)?	No
Imposta visualizzazione prezzo:	<input type="radio"/> Visualizza prezzo IVA inclusa (Es.: 110 €) <input checked="" type="radio"/> Visualizza separatamente prezzo ed IVA (Es.: 100 € + 10% IVA)
Seleziona etichetta valuta: <small>(Attenzione: La modifica dell'etichetta valuta non significa necessariamente la modifica dei prezzi prodotti come valori.)</small>	EUR(€) Euro
Imposta il formato per i prezzi con decimali per differente lingue: <small>(Per esempio, per ottenere il formato italiano : 10000,55 scegli il separatore per decimali, ed il separatore per migliaia ". Il formato predefinito è il formato inglese: 10,000.55)</small>	2 : Numero di decimali , : Separatore per decimali : Separatore per migliaia
	Lingua: IT Duplica

In **Impostazioni Prezzi** si decide se visualizzare o meno i prezzi agli utenti non registrati, se i prodotti senza prezzo devono essere nascosti all'utente, la modalità di visualizzazione IVA (inclusa, esclusa), la valuta predefinita ed il formato numerico per la lingua selezionata. Va evidenziato che il formato numerico e la valuta non modificano il valore effettivo del prezzo memorizzato in tabella: la valuta in questo caso è semplicemente l'etichetta di valuta predefinita per la visualizzazione di default dei prezzi.

Impostazioni ordini	
Un cliente non registrato può ordinare?	No
Un utente registrato può salva le modifiche a il suo profilo durante l'elaborazione dell'ordine (nome, azienda, indirizzo, telefono, fax, e-mail, no IVA, codice fiscale)?	Sì

In **Impostazioni Ordini** si definisce la possibilità o meno di ordinare degli utenti pubblici e l'accesso dell'utente in modifica ai dati del proprio profilo. Nel caso in cui il prodotto abbia varianti è definibile dall'amministratore se è ordinabile solo il prodotto principale, solo le varianti o entrambe le opzioni, dalla sezione **Impostazione ordini**.

Prima pagina di ogni categoria	
Prima pagina di ogni categoria:	<input checked="" type="checkbox"/> Mostra l'offerta speciale della categoria <input checked="" type="checkbox"/> Mostra l'immagine della categoria <input checked="" type="checkbox"/> Mostra la descrizione della categoria

Prima pagina di e-commerce	
Prima pagina di e-commerce:	<input checked="" type="checkbox"/> Mostra offerta speciale <input checked="" type="checkbox"/> Mostra l'immagine predefinita di e-commerce <input checked="" type="checkbox"/> Mostra la tua pagina (immagine, dettagli...)
Imposta il contenuto della tua prima pagina:	<div style="border: 1px solid gray; height: 40px; width: 100%;"></div>
	<input type="button" value="Apri Editor"/> <input type="button" value="Mostra"/> Lingua: IT <input type="button" value="Duplica"/>
Possibile impostazione di offerta speciale di magazzino:	<input type="button" value="Imposta"/>

E' inoltre possibile in **Prima pagina di eCommerce** e **Prima pagina di ogni categoria** impostare la visualizzazione delle pagine principali del modulo eBusiness con offerte speciali, immagini o testo.

Impostazioni ordini

La sezione impostazioni ordini (attiva a meno che il modulo eBusiness non sia stato impostato come solo catalogo) definisce tutti i parametri relativi all'inserimento degli ordini online. E' possibile assegnare la possibilità d'ordine ai soli utenti registrati e definire una serie ed un intervallo per la generazione del numero d'ordine per una più facile integrazione con il gestionale del cliente.

Le condizioni di pagamento sono definibili in questa sezione, così come la possibilità di inserire una data di richiesta consegna e la possibilità di evasione parziale degli ordini.

Vengono inoltre specificate le condizioni di pagamento, le note fisse da allegare all'ordine, i metodi di notifica all'utente del buon fine dell'ordine (mail e messaggio, file riepilogativo), la possibilità o meno di utilizzare un indirizzo di fatturazione separato, l'obbligatorietà del codice fiscale o partita IVA, la gestione delle destinazioni diverse (più filiali della stessa azienda). Sempre dalla sezione Impostazioni ordini viene anche attivata per gli agenti la funzionalità di modifica degli ordini inseriti dai clienti associati, unitamente ai loro profili.

Impostazioni avanzate

La form relativa alle impostazioni avanzate definisce i link che compaiono nella parte superiore di ogni pagina del modulo eBusiness, la visibilità di informazioni relative ai prodotti (codice alternativo, ricambi, file allegato, link, ecc.) ed il tipo di visualizzazione della giacenza (numerico, descrittivo).

Mostra/Nascondi E-commerce top links

La form definisce la visibilità delle sezioni:

- Offerta speciale
- Cerca Prodotti
- Carrello
- Ordine online
- Lista di ordini
- Lista delle fatture
- Lista delle offerte
- Mappa - Categorie
- Home
- eBusiness Home

Etichette e-commerce

Consente di impostare le etichette delle sezioni di eBusiness ("carrello", "ordini", "codice",...) nelle lingue attive. Le etichette presentano comunque un valore di default per ogni lingua.

Mostra/Nascondi etichette nella visualizzazione prodotti

Definisce la visualizzazione o meno delle etichette nei campi visualizzati nella lista prodotti e nel dettaglio prodotto.

Categorie di prodotti

I prodotti possono essere suddivisi in categorie e sottocategorie. L'elenco delle categorie compare nella parte sinistra del modulo (posizione non modificabile) e consente una sud-

divisione logica dei prodotti per tipologia. La struttura delle categorie si basa sulla tabella **tbl_categories** descritta a seguire.

tbl_categories

name	description	type	size	key	not null	default	observation
code_cat	Code category	VARCHAR	100	yes	yes	0	Code different of 0
code_father	Code of the category father	VARCHAR	100	yes	yes	0	For the categories from the first level let code_father = 0
code_lng	Language code (e.g.: IT, EN, FR, DE, ES, RO)	CHAR	2	yes	yes	IT	ISO code of the language - uppercase
position_cat	Position inside father category	INT	10		yes	1	
descr_short_cat	Name of category	VARCHAR	50		yes	?	
descr_long_cat	Long description (may contain HTML code)	TEXT					
type_cat	Type of the category	INT	10			0	Linked to type_cat from tbl_cat_type
picture_cat	Picture of category shown together with description when accessing the category	VARCHAR	255				The picture needs to be uploaded on the server
picture_cat_default	Name of common picture for products with missing picture from this category	VARCHAR	255				The picture needs to be uploaded on the server
icon_cat	Category icon for the categories tree	VARCHAR	255				
icon_over_cat	Category icon in mouse over for the categories tree	VARCHAR	255				
show_cat	Flag for hiding the category on the public area (0 or 1)	CHAR	1			1	
cat_link	Category link	VARCHAR	255				e.g.: www.mylink.com
cat_file	Category file	VARCHAR	255				e.g.: myfile.doc; the file needs to be uploaded on the server
UNSPSC	United Nations Standard Products and Services Code	VARCHAR	8				
f_txt1	-unused field; reserved for future implementations -	TEXT					
f_txt2	-unused field; reserved for future implementations -	TEXT					

name	description	type	size	key	not null	default	obervation
f_txt3	-unused field; reserved for future implementations -	TEXT					
f_int1	-unused field; reserved for future implementations -	TINYINT	3				
f_int2	-unused field; reserved for future implementations -	INT	10				
f_txt3	-unused field; reserved for future implementations -	INT	10				

Il legame tra categorie e sottocategorie è ricorsivo sulla tabella attraverso i campi **code_cat** e **code_father**. Lo stesso prodotto può essere associato a più categorie a un qualsiasi livello attraverso la tabella **tbl_cat_objects**. La tabella definisce l'appartenenza di un prodotto ad una categoria attraverso i campi **code_cat** (codice categoria) e **code** (codice prodotto).

tbl_cat_objects

name	description	type	size	key	not null	default	obervation
code_cat	Category code	VARCHAR	100	yes	yes	0	Linked to code_cat from tbl_categories
code	Product code	VARCHAR	100	yes	yes	0	Linked to code from tbl_objects
position_obj	Product position in category	INT	10		yes	0	
cat_special_offer	Flag for set as special offer for the category (0 or 1)	TINYINT	3			0	= 1 if the product is special offer for the category

La posizione del prodotto all'interno della categoria viene specificata dal campo **position_obj** (`SELECT ... ORDER BY position_obj`) ed è possibile indicare se il prodotto deve comparire nella sezione offerte della categoria di appartenenza (campo **cat_special_offer**). Va osservato che nonostante la struttura dati consenta di avere categorie diverse per ogni lingua (campo **code_lng** in chiave primaria nella tabella **tbl_categores**) di fatto l'interfaccia PHP (interfaccia di amministrazione) non permette di diversificarle. E' possibile superare questo limite gestendo i dati delle tabelle direttamente senza usufruire dell'interfaccia web

di setup (come spesso avviene) attraverso il programma di sincronizzazione dati DB Sync integrato nella piattaforma.



CATEGORIA NUOVA LIVELLO 1 dopo 1 - TELEFONIA

FILTRA PER VAI

[1] [TEL] TELEFONIA Nuova sottocategoria primo

[2] [3] MONITOR

[acc02] Accessori

[3] [4] STAMPANTI

[4] [5] SCANNERS

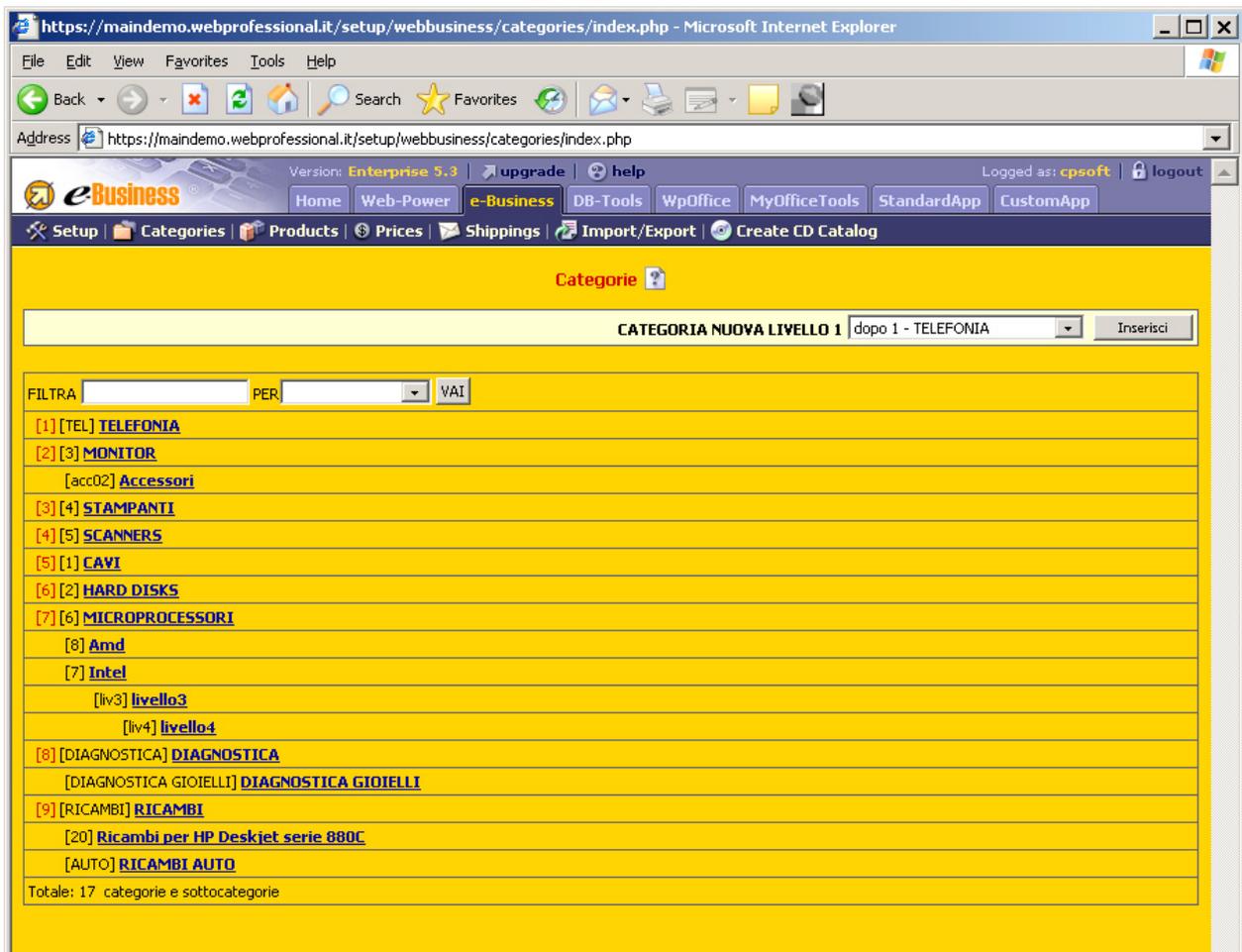
[5] [1] CAVI

Dal menù "Categories" si accede all'interfaccia di amministrazione delle categorie. La prima schermata è preposta alla creazione, cancellazione e modifica delle categorie. La struttura è gerarchica stretta, cioè ogni categoria ha uno ed un solo padre. Dai pulsanti presenti sulla categoria selezionata si accede alla sezione prodotti associati alla categoria ed alle proprietà della categoria, con funzionalità per copiare, spostare la categoria, gestirne la visualizzazione (nome, descrizione, icona, immagine, visibilità) e definire i prodotti che compaiono nella prima pagina della stessa come offerte speciali.

L'introduzione dei "Tipi Categoria"

Nonostante l'estrema flessibilità della struttura delle categorie e dell'associazione dei prodotti ad esse, su richiesta degli utilizzatori venne studiata l'opportunità di diversificare ulteriormente le categorie attraverso l'introduzione dei "tipi categoria".

L'esigenza era di poter utilizzare diverse logiche di raggruppamento sugli stessi prodotti in modo alternativo. Ad esempio una concessionaria auto avrebbe potuto suddividere per marca e modello così come per tipologia (berlina, station wagon, fuoristrada,...) e fascia di prezzo. L'impatto della modifica sul modulo fu minima attraverso l'introduzione del campo **type_cat** nella tabella **tbl_categories**. Una nuova tabella (**tbl_cat_type**) avrebbe definito la parte descrittiva dei tipi categoria (descrizione in lingua, note) in relazione con **tbl_ca-**



Strutture dati per gestione di articoli e categorie.

La struttura dati per la gestione degli articoli si compone principalmente delle tabelle **tbl_object** e **tbl_objects_Ing**. La prima contiene le informazioni principali riguardanti l'articolo, la seconda le informazioni in lingua per la visualizzazione, in relazione uno a molti.

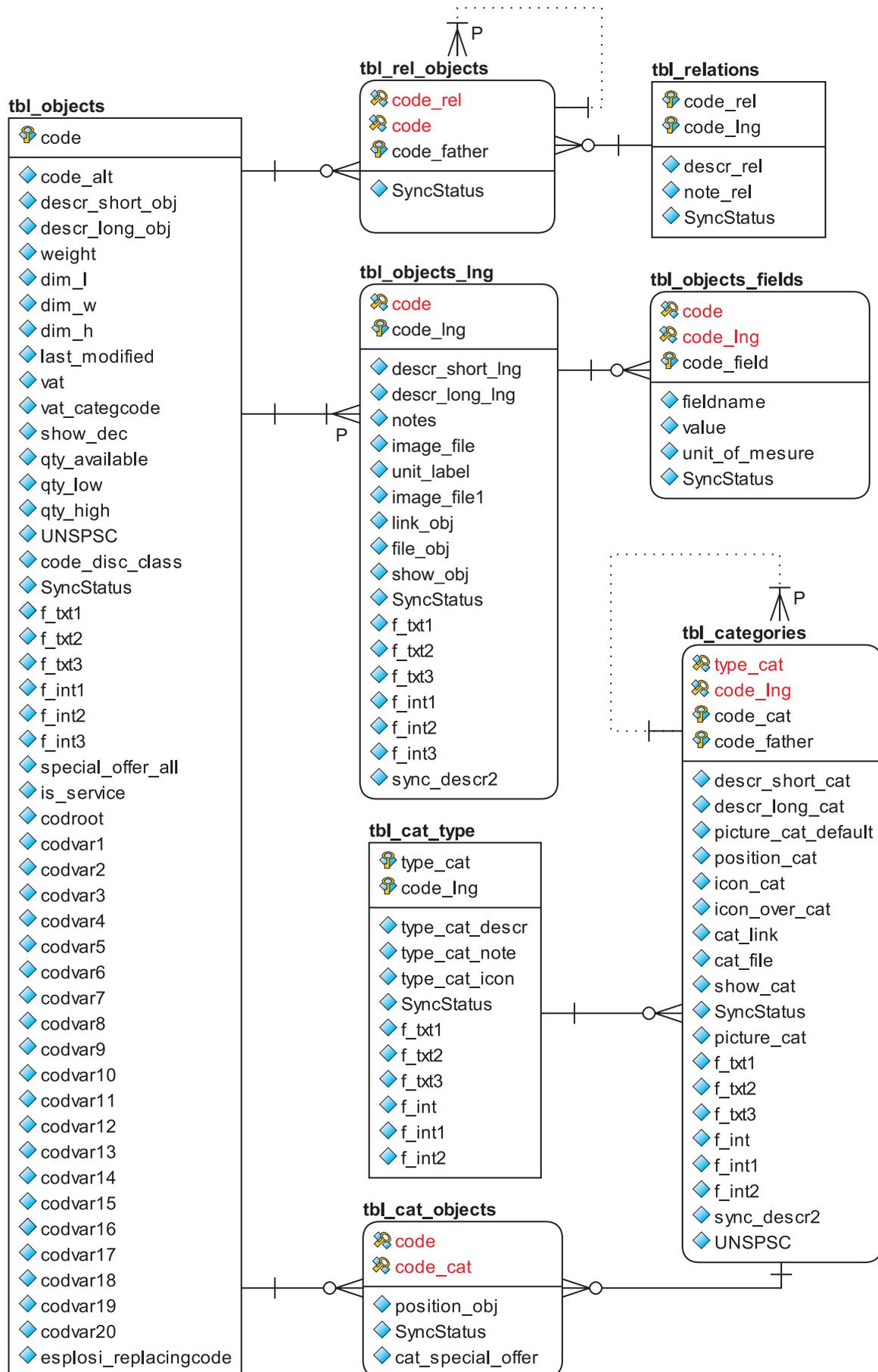
Inizialmente **tbl_objects** gestiva completamente anche IVA e valuta. L'IVA era un attributo dell'articolo, e come tale non era possibile avere una gestione differenziata per acquirente. La prima versione di eBusiness, in Cold Fusion, non gestiva nemmeno l'IVA per articolo, considerando un default pari a "20%".

Ogni articolo è relazionabile ad altri articoli definendo uno o più tipi di relazione in tabella **tbl_relations**. Un tipo di relazione corrisponde ad un metodo di categorizzazione degli articoli. Gli stessi articoli potranno poi essere visualizzati secondo un metodo o l'altro in base al tipo di classificazione selezionato.

Esempio:

Articoli di tipo automobile sono suddivisibili per Marca, Tipologia (manovolume, station wagon, coupè,...) e Cilindrata. Gli stessi articoli potrebbero però essere visualizzati per Tipologia e Fascia di prezzo impostando un secondo tipo categorie e creando le due categorie sopra citate.

Ogni articolo può appartenere ad una o più categorie (lo stesso articolo può comparire in categorie diverse, per permettere una gestione più flessibile del sito di ecommerce). Le categorie sono visualizzate con struttura gerarchica stretta.



tegories Non-Identifying Optional (con cardinalità Zero or One to Zero or More). Questo avrebbe consentito di applicare la modifica al codice ed alla struttura del database senza dover operare in alcun modo sui dati degli utenti attivi.

tbl_categories

name	description	type	size	key	not null	default	observation
type_cat	cat. type code	INT		yes	yes	0	
type_cat_descr	cat. type description	VARCHAR	255		yes		
type_cat_note	cat. type note	TEXT					
type_cat_icon	cat. type icon filename	VARCHAR	30				
code_lng	language code	CHAR	2	yes	yes		

Il menu "Categories" - Types consente di definire i tipi di categorie. L'appartenenza di una categoria ad un tipo si imposta nelle proprietà delle singole categorie (dalla sezione "Categories - Type" è possibile solo visualizzare l'elenco delle categorie appartenenti ai tipi impostati).

Prodotti

Il modulo di gestione dei prodotti permette l'inserimento, la modifica e la cancellazione dei record relativi ai prodotti di eBusiness. L'inserimento di un prodotto può avvenire anche nella sezione "Prodotti" di ogni categoria.

La modifica della posizione dei prodotti inseriti è possibile tramite l'opzione "Sposta prodotto" per ogni categoria (viene aggiornato il campo **tbl_cat_objects.position_obj** dei prodotti appartenenti a quella categoria). Per aggiungere un nuovo prodotto è necessario inserire il codice prodotto ed eventualmente il codice alternativo. Si raccomanda di fare attenzione all'inserimento dei codici prodotti perché essi non verranno più modificati successivamente per evitare incongruenze nei dati (prezzi, ordini,...). Dopo aver inserito il codice e premuto il pulsante "Inserisci" verrà aperta la pagina relativa alle proprietà del prodotto.

commenti, precisazioni). Anche queste note possono contenere codice HTML. Come per le categorie, nel caso in cui il contenuto della descrizione o delle note venga copiato da un altro documento (e non costruito con l'editor HTML online) e contenga riferimenti alle immagini caricate sul server, si raccomanda di spuntare l'opzione "Modifica sorgente immagine" (< `IMG SRC=...`>): una funzione PHP provvederà ad apportare le necessarie modifiche al valore dell'attributo SRC.

Le immagini per la visualizzazione lista e dettagli di un prodotto possono essere diverse. L'immagine di dettaglio viene automaticamente sostituita dall'immagine di lista nel caso non sia stata definita. Al contrario, le immagini di lista se assenti non vengono sostituite dalle immagini di dettaglio ma viene visualizzata l'immagine di default "non disponibile", definibile per ogni categoria (campo `tbl_categories.picture_cat_default`).

La dimensione dell'immagine per i dettagli prodotto non ha restrizioni: tuttavia, per una grafica equilibrata, si raccomanda di mantenere le dimensioni in un quadrato di 250 pixel. Per la lista di prodotti invece, per esigenze di impaginazione ed allineamento, le immagini hanno una dimensione fissa di 90x90 pixel (il tag IMG ha attributi WIDTH e HEIGHT fissati), per cui è indispensabile rispettare queste dimensioni per evitare distorsioni.

Se si desidera copiare i dati già inseriti per un altro prodotto esiste l'opzione "Duplica". Verrà scelto il prodotto di riferimento per la copia, ed i dati da trasferire al prodotto corrente tra:

- nome
- dettagli
- descrizione
- note
- immagine di presentazione
- immagine per la lista di prodotti

- varianti

E' possibile impostare il prodotto come offerta speciale dell'intero sito o della categoria, inserire un link oppure allegare un file, definire l'esenzione da spese di spedizione ed inserire il contenuto di una mail personalizzata da inviare all'utente all'atto dell'ordine di quel prodotto specifico.

L'etichetta "unità" rappresenta l'unità di misura per esprimere le quantità relative a tale prodotto (litri, pezzi, bottiglie, casse,...).

Qualora si voglia nascondere un prodotto, questo avviene facilmente spuntando l'opzione "Nascondi prodotto", in tutte le lingue oppure solo per le lingue desiderate (nota: non esiste però alcun meccanismo che limita la visibilità dei prodotti agli utenti sulla base della lingua associata agli utenti stessi).

Proprietà avanzate

Nella sezione "Proprietà avanzate" sono gestite una serie di proprietà relative solitamente alla modalità di amministrazione.

UNSPSC (Universal Standard Products and Services Classification) è un codice univoco di identificazione di prodotto, standard adottato a livello mondiale .

La percentuale IVA entra a far parte del calcolo del prezzo di un prodotto; è possibile assegnare un valore di default ed eventualmente una categoria IVA. L'IVA verrà applicata dopo la determinazione dei prezzi netti specifici.

La descrizione breve e la descrizione dettagliata di un prodotto sono informazioni utilizzate solo a livello amministrativo e solitamente coincidono con la denominazione data al prodotto nel programma gestionale (spesso non utilizzabile come descrizione per il pubblico).

I dettagli di imballo (peso, lunghezza, larghezza, altezza) sono dati necessari all'evasione degli ordini ricevuti dai clienti. L'esistenza, la quantità minima e la quantità massima

definiscono la descrizione delle scorte nell'area pubblica. Secondo l'opzione selezionata nella configurazione, la descrizione delle scorte avviene in modalità:

- **numerica:** le giacenze vengono espresse numericamente in base alla quantità disponibile
- **testo:** le giacenze vengono espresse tramite una delle seguenti espressioni: 'media', 'bassa' ed 'alta' (etichette comunque definibili dall'utente)

La descrizione della giacenza (o disponibilità) dei prodotti in formato testo si basa sulla giacenza reale in formato numerico (**tbl_objects.qty_available**) e su due valori (**tbl_objects.qty_low** e **tbl_objects.qty_high**) che definiscono le tre zone della disponibilità in formato testo.

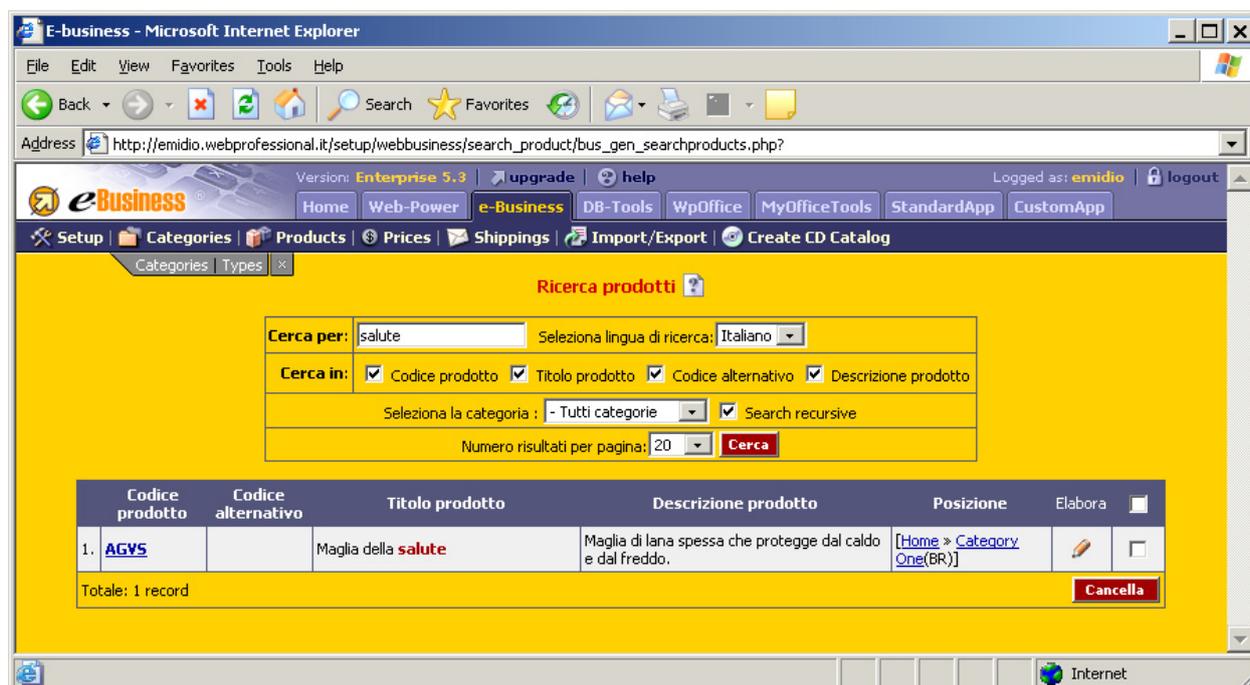
La tabella che segue esprime le tre situazioni possibili nella gestione della quantità tramite etichette di testo.

condizione	disponibilità
qty_low > qty_available	disponibilità bassa
qty_low <= qty_available <= qty_high	disponibilità media
qty_available > qty_high	disponibilità alta

Dettagli

L'utente ha la possibilità di definire per ogni prodotto proprietà aggiuntive che compariranno nel dettaglio articolo. Le informazioni necessarie alla definizione dei dettagli di un prodotto sono: nome proprietà, unità di misura della proprietà e valore. Ad esempio, per un prodotto di tipo "televisore", è possibile enumerare:

- diagonale schermo 37 cm



- diagonale visibile 34 cm
- riproduzione del suono mono
- potenza in Watt 2 W
- garanzia 12 mesi
- peso 9.1 kg

I dettagli prodotto devono essere specificati per tutte le lingue gestite dal sito.

Ricerca prodotti

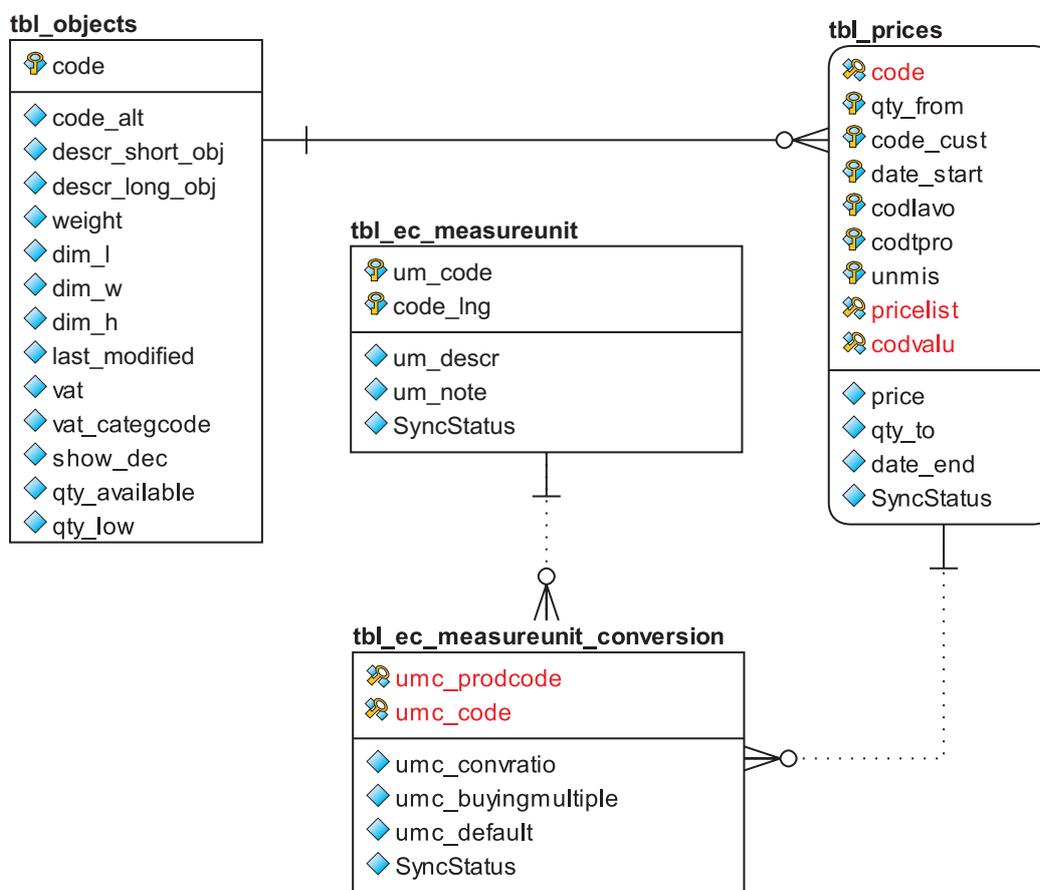
La ricerca di un prodotto avviene sui campi: codice prodotto, titolo, descrizione. La parola ricercata può trovarsi in ognuno dei campi specificati oppure essere presente in tutti i campi in dipendenza della richiesta formulata (OR oppure AND). È possibile inoltre cercare il testo desiderato nelle informazioni inserite per una lingua determinata (quella predefinita è uguale alla lingua predefinita del sito) oppure cercare fra i prodotti appartenenti ad una categoria particolare. Il numero di risultati per pagina è configurabile da 20 a 100. Ogni

occorrenza della parola cercata viene evidenziata nella visualizzazione.

Unità di misura

La gestione di unità di misura multiple e relative conversioni venne introdotta in eBusiness successivamente al passaggio a PHP. Inizialmente era semplicemente definibile l'unità come etichetta in **tbl_objects_ing.unit_of_mesure**, campo descrittivo.

Questo non consentiva la gestione di articoli quali materie prime o prodotti del settore chimico, dove lo stesso articolo poteva essere venduto a metri oppure piedi per il mercato anglosassone. Introdurre le unità di misura e le relative conversioni comportò la creazione di due nuove tabelle: **tbl_ec_measureunit** e **tbl_ec_measureunit_conversion**.



tbl_ec_measureunit

name	description	type	size	key	not null	default	observation
um_code	Code of unit measure	VARCHAR	20	yes	yes		
code_lng	Language code	VARCHAR	2	yes	yes	IT	
um_descr	Unit measure escription	VARCHAR	255				
um_note	Note	TEXT	1				

tbl_ec_measureunit_conversion

name	description	type	size	key	not null	default	observation
umc_prodcode	Product code	VARCHAR	100	yes	yes		
umc_code	Measure unit code	VARCHAR	20	yes			
umc_convratio	Conversion rate	DOUBLE				1	
umc_buyingmultiple	Buying multiple of	DOUBLE					
umc_default	Flag to b shown as default in measure units combo	TINYINT	2			0	

La relazione tra le due tabelle è specificata su **um_code = umc_code** come evidenziato nello schema di pagina a lato.

Pezzi di ricambio e articoli correlati

L'esigenza di introdurre in eBusiness il concetto di distinta base (se pure in modo elementare) e di consumabili associati ad un prodotto comportò la definizione di "relazioni" tra i prodotti, definibili dall'utente, per poter correlare in qualche modo prodotti differenti e visualizzare unitamente al dettaglio prodotto anche una lista di articoli in qualche modo correlati attraverso queste "relazioni". Il sistema si basa su due tabelle: **tbl_rel_objects** che definisce la relazione padre - figlio tra due prodotti, e **tbl_relations** descrittiva del tipo di "relazione".

tbl_rel_objects

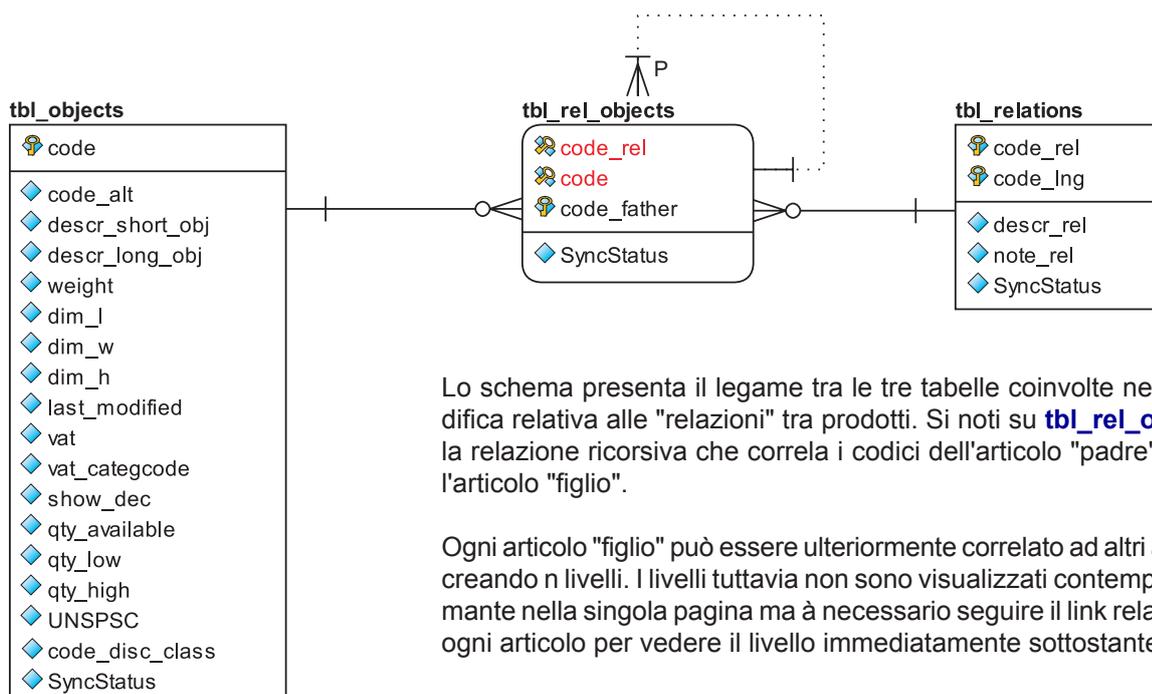
name	description	type	size	key	not null	default	observation
code	Code of the product which will be the sparepart product	VARCHAR	100	yes	yes	0	Linked to code from tbl_objects

name	description	type	size	key	not null	default	observation
code_rel	Relation code	VARCHAR	10	yes	yes	0	Linked to code_rel from tbl_relations
code_father	Code of the parent product	VARCHAR	100	yes	yes	0	Linked to code from tbl_objects

tbl_relations

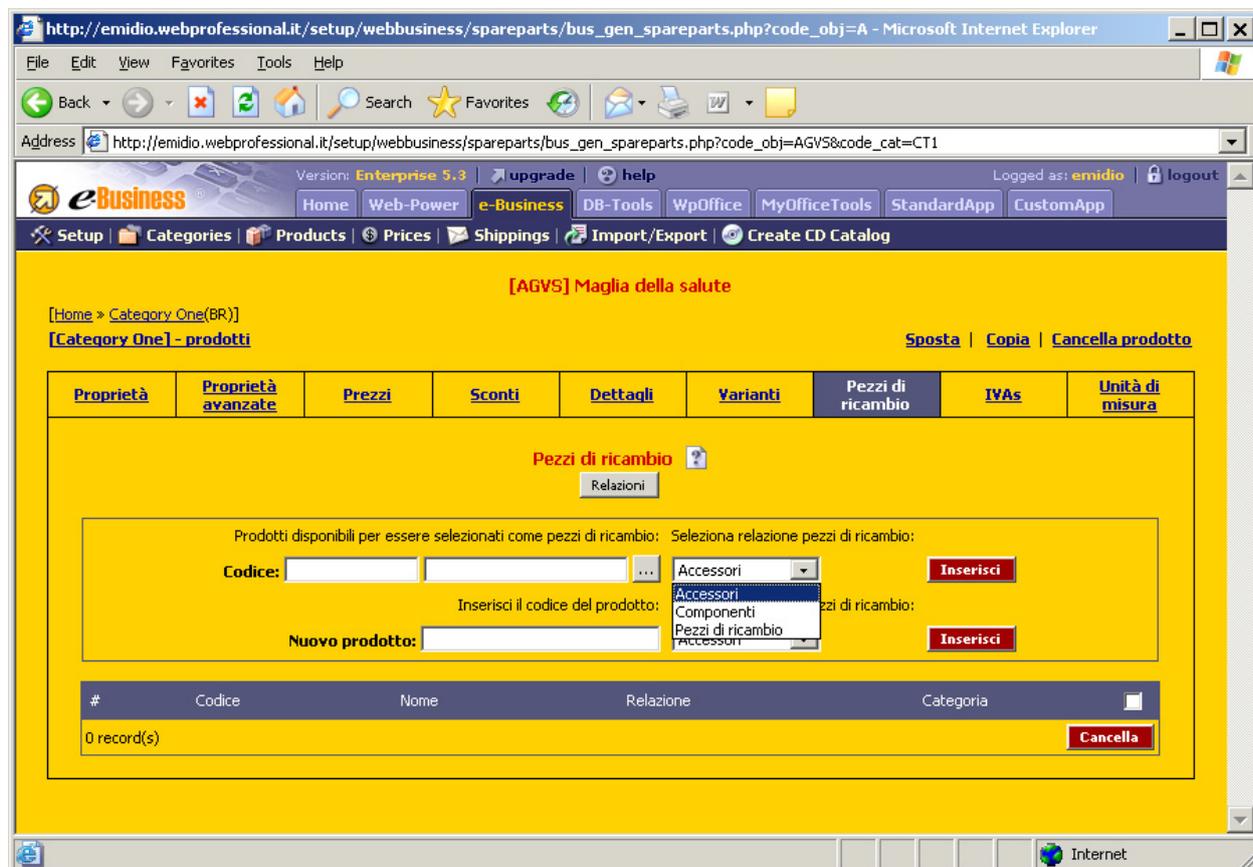
name	description	type	size	key	not null	default	observation
code_rel	Code of relation	VARCHAR	10	yes	yes	0	
code_lng	Language code (e.g.: IT, EN, FR, DE, ES, RO)	CHAR	2	yes	yes	IT	ISO code of the language - upper-case
descr_rel	Title of relation	VARCHAR	255		yes	???	
note_rel	Description of relation	VARCHAR	255				

E' definita una relazione ricorsiva sulla tabella **tbl_rel_objects** che instaura il legame gerarchico padre - figlio tra i prodotti. Nell'utilizzo corrente di eBusiness le relazioni più frequentemente definite sono quelle relative alle parti di ricambio, ai materiali di consumo ed agli "articoli correlati". In visualizzazione prodotto gli articoli relazionati compaiono a piè di pagina preceduti dalla descrizione del tipo di relazione. eBusiness presenta di default le relazioni "Accessori", "Componenti" e "Pezzi di ricambio".



Lo schema presenta il legame tra le tre tabelle coinvolte nella modifica relativa alle "relazioni" tra prodotti. Si noti su **tbl_rel_objects** la relazione ricorsiva che correla i codici dell'articolo "padre" e dell'articolo "figlio".

Ogni articolo "figlio" può essere ulteriormente correlato ad altri articoli, creando n livelli. I livelli tuttavia non sono visualizzati contemporaneamente nella singola pagina ma è necessario seguire il link relativo ad ogni articolo per vedere il livello immediatamente sottostante.



Prezzi

La struttura dati relativa a prezzi e listini di eBusiness deriva direttamente dal gestionale Business. Nel passaggio da Cold Fusion a PHP le tabelle mantennero una struttura molto simile dal momento che in fase di analisi fu ritenuta sufficientemente flessibile da potersi adattare anche a situazioni particolari impostate su gestionali diversi, mantenendo la possibilità di integrarsi facilmente al gestionale proposto da CP Software.

Definito un listino generico di default (listino 0), ad ogni utente è associabile un listino diverso. All'interno dei listini si determinano i prezzi dei singoli articoli, con possibilità di differenziare i prezzi per quantità, data di validità e promozioni in corso.

tbl_prices

name	description	type	size	key	not null	default	observation
code	Product code	VARCHAR	100	yes	yes	0	Linked to code from tbl_objects

name	description	type	size	key	not null	default	observation
code_cust	Customer code	VARCHAR	20	yes	yes	0	Linked to u_id/u_externalcode from tbl_users; if you fill this field, let pricelist=0
pricelist	Price list code	VARCHAR	100	yes	yes	0	Linked to code_pricelist from tbl_pricelist
codtpro	Promotion code	VARCHAR	50	yes	yes	0	Linked to tb_codtpro from tbl_promotions
qty_from	The start quantity to validate price	DOUBLE		yes	yes	0	
qty_to	The end quantity to validate price	DOUBLE			yes	9999999999	
date_start	The start date to validate price	DATE	10	yes	yes	01/01/2000	
date_end	The end date to validate price	DATE	10		yes	01/01/2100	
price	Price value	DOUBLE					
codlavo	Code lavoro - actually not used	VARCHAR	10	yes		0	
codvalu	Code currency	VARCHAR	10	yes		0	
unmis	Unit measurement	VARCHAR	10	yes		0	

Il campo `code_cust` consente di implementare eccezioni al listino per un particolare articolo/cliente, senza generare per questo un nuovo listino. L'interrogazione al database per la determinazione del prezzo di un articolo si presenta come:

```
SELECT [article data]
FROM [article and price tables]
WHERE tbl_prices.code = '{article code}'
AND (qty_from <= {quantity} < quantity_to)
AND (pricelist = 0
OR pricelist = {customer pricelist}
OR code_cust = {customer code})
AND (date_start <= Now() < date_end)
ORDER BY code_cust DESC, pricelist DESC
LIMIT 1
```

L'ordinamento e la clausola "LIMIT 1" restituiscono il prezzo corretto, partendo dal caso più particolare (prezzo dell'articolo definito per il singolo cliente come eccezione al listino - lo statement `ORDER BY code_cust DESC` pone nel primo record restituito il record

tbl_objects

 code
 code_alt
 descr_short_obj
 descr_long_obj
 weight
 dim_l
 dim_w
 dim_h
 last_modified
 vat
 vat_catgcode
 show_dec
 qty_available
 qty_low
 qty_high
 UNSPSC
 code_disc_class
 SyncStatus
 f_txt1
 f_txt2
 f_txt3
 f_int1
 f_int2
 f_int3
 special_offer_all
 is_service
 codroot
 codvar1
 codvar2
 codvar3
 codvar4
 codvar5
 codvar6
 codvar7
 codvar8
 codvar9
 codvar10
 codvar11
 codvar12
 codvar13
 codvar14
 codvar15
 codvar16
 codvar17
 codvar18
 codvar19
 codvar20
 esplosi_replacingcode

tbl_prices

 code
 qty_from
 code_cust
 date_start
 codlavo
 codtpro
 unmis
 pricelist
 codvalu
 price
 qty_to
 date_end
 SyncStatus

tbl_pricelist

 code_pricelist
 descr_pricelist
 notes
 flag1
 flag2
 flag3
 field1
 field2
 title_pricelist
 nrdaysdelayed
 SyncStatus

tbl_ec_measureunit

 um_code
 code_Ing
 um_descr
 um_note
 SyncStatus

tbl_currency

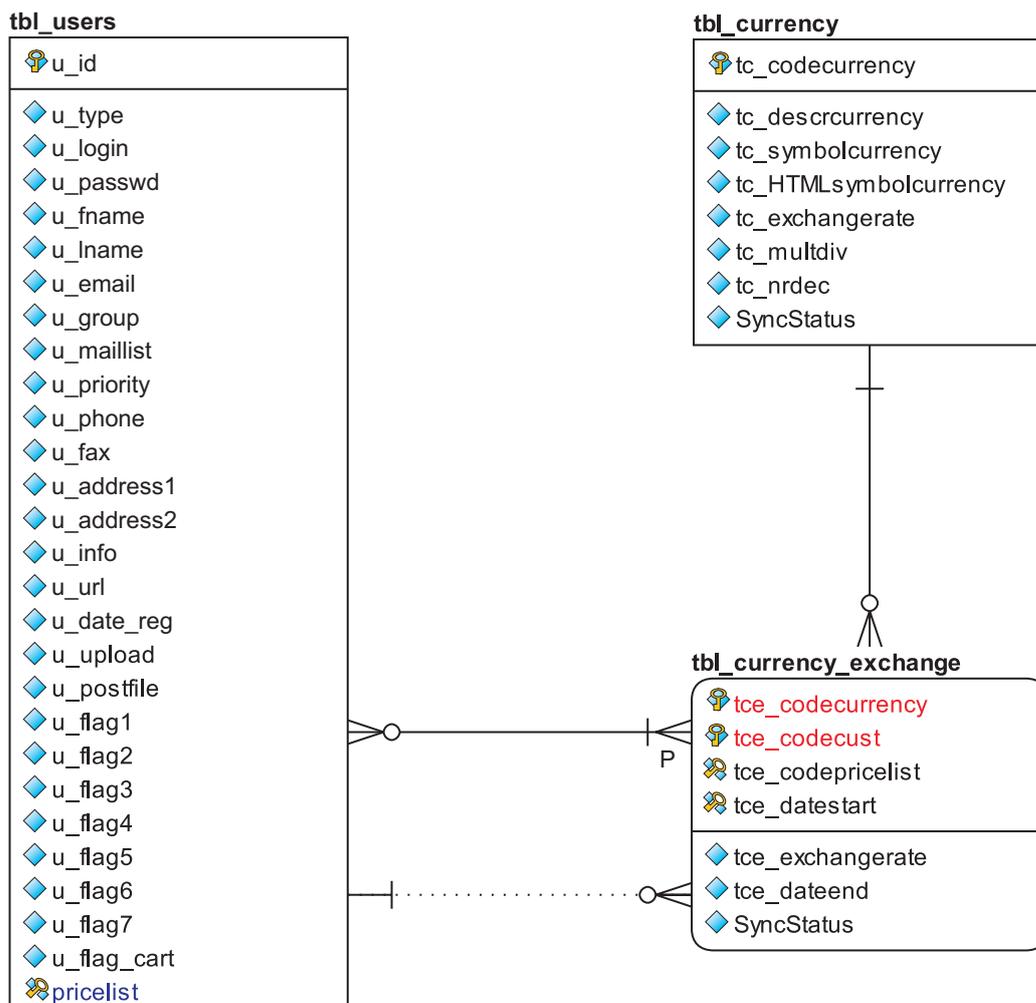
 tc_codecurrency
 tc_descrcurrency
 tc_symbolcurrency
 tc_HTMLsymbolcurrency
 tc_exchangerate
 tc_multdiv
 tc_nrdec
 SyncStatus

tbl_users

 u_id
 u_type
 u_login
 u_passwd
 u_fname
 u_lname
 u_email
 u_group
 u_maillist
 u_priority
 u_phone
 u_fax
 u_address1
 u_address2
 u_info
 u_url
 u_date_reg
 u_upload
 u_postfile
 u_flag1
 u_flag2
 u_flag3
 u_flag4
 u_flag5
 u_flag6
 u_flag7
 u_flag_cart
 pricelist
 code_disc_class
 addr_street
 addr_number
 addr_zip
 addr_state
 addr_country
 ship_street
 ship_number
 ship_city
 ship_country
 code_Ing
 addr_city
 ship_state
 ship_zip
 company
 company_id
 vat_no
 u_txt1
 u_txt2
 u_txt3
 u_nr1

con **code_cust** <> 0 nel caso che esista) a quello più generico (listino 0 e **code_cust** = 0 - nel caso non esista prezzo per cliente o listino per cliente questo sarà l'unico record restituito). Non c'è possibilità di gestire da interfaccia web listini per gruppi clienti (il listino è associato agli utenti webprofessional, campo **tbl_users.pricelist**). Tuttavia utilizzando l'applicativo di sincronizzazione dati DB Sync il mantenimento di gruppi di utenti per listini è facilmente attuabile se già impostato nel database di origine del gestionale.

Definiscono la chiave primaria anche i campi **codlavo** (codice lavorazione), **unmis** e **codvalu**. Il campo **codlavo** (default 0) permette al sistema di utilizzare prezzi diversi per lo stesso articolo dipendentemente dallo stato della lavorazione (semilavorati). Il campo,



derivante dalla struttura dati di Business, è ad oggi non utilizzato, lasciando però aperta la possibilità di implementare la funzionalità senza intervenire sulle chiavi primarie delle tabelle in essere (comportando quindi tempi di sviluppo più contenuti). I campi **unmis** e **codvalu** fanno invece riferimento alla gestione di unità di misura diverse e valute diverse.

Unità di misura

L'unità di misura interviene nella chiave primaria della tabella prezzi garantendo la possibilità di definire un unico prezzo per un prodotto la cui vendita avviene per multipli/sottomultipli anche on interi dell'unità di misura (metri, litri, chilogrammi,...). La tabella **tbl_ec_measureunit** descrive le unità di misura (nelle diverse lingue) e si relaziona con **tbl_ec_measureunit_conversion**.

tbl_ec_measureunit

name	description	type	size	key	not null	default	observation
um_code	Code of unit measure	VARCHAR	20	yes	yes		
code_lng	Language code	VARCHAR	2	yes	yes	IT	
um_descr	Unit measure escription	VARCHAR	255				
um_note	Note	TEXT	1				

tbl_ec_measureunit_conversion

name	description	type	size	key	not null	default	observation
umc_prodcod	Product code	VARCHAR	100	yes	yes		
umc_code	Measure unit code	VARCHAR	20	yes			
umc_convratio	Conversion rate	DOUBLE				1	
umc_buyingmultiple	Buying multiple of	DOUBLE					
umc_default	Flag to be shown as default in measure units combo	TINYINT	2			0	

In **tbl_ec_measureunit_conversion** l'unità di misura è associata al singolo articolo su cui si determina eventualmente un fattore di conversione rispetto all'unità di misura principale (es. metri/piedi) ed un multiplo di vendita (**umc_buyingmultiple** = 10 significa che il

prodotto è vendibile solo per quantità multiple di 10, es. 10 Kg, 20 Kg, 30 Kg). Lo schema delle relazioni tra le tabelle è già stato illustrato precedentemente.

Valuta

L'utilizzo di valute diverse dallo standard Euro venne introdotto solo successivamente al passaggio da Cold Fusion a PHP. Dalla versione 5.1 esiste la possibilità di definire la valuta utilizzata nella visualizzazione prezzi (inizialmente i prezzi erano visualizzabili solo in Euro) con l'introduzione delle tabelle **tbl_currency** e **tbl_currency_exchange**. Prima della modifica la valuta poteva essere impostata (Euro, dollari, yen,...) ma rimaneva comunque unica per il modulo: non era possibile cioè utilizzare tassi di scambio. Inoltre cambiare valuta successivamente allo startup del sito comportava lanciare manualmente un UPDATE sulla tabella **tbl_prices** per convertire i prezzi nella nuova valuta al tasso corrente. Era cioè consentito da interfaccia di setup soltanto di definire il simbolo di valuta in visualizzazione.

La ragione dell'introduzione tardiva della gestione della valuta ha principalmente ragioni di mercato. Il prodotto nelle prime release aveva indirizzo fortemente localizzato in Italia, su aziende senza clientela estera o comunque con clientela a cui era imposto il regime valutario in Euro. Con l'apertura verso nuovi mercati extracomunitari e l'utilizzo della piattaforma da parte di multinazionali l'esigenza si tradusse in una nuova specifica di funzionalità aggiuntiva al modulo eBusiness.

tbl_currency

name	description	type	size	key	not null	default	observation
tc_codecurrency	Code of currency	VARCHAR	20	yes	yes		
tc_descrcurrency	Currency description	VARCHAR	50				Used only in administration area

name	description	type	size	key	not null	default	observation
tc_symbolcurrency	Currency symbol	VARCHAR	10		yes		If tc_HTMLsymbolcurrency is empty then this field is used for display in public area (Ex: GBD for United Kingdom Pounds)
tc_HTMLsymbolcurrency	HTML symbol currency	VARCHAR	10				If the field contains a non empty value then it's used for display in public area (Ex: £ for United Kingdom Pounds -> £)
tc_multdiv	Flag to specify if multiply or divide operation will be used in price conversion	CHAR	1		yes	D	Admitted values: D for Divide and M for Multiply
tc_nrdec	Number of decimals used in price display	INT					

tbl_currency_exchange

name	description	type	size	key	not null	default	observation
tce_codecurrency	Code of currency	VARCHAR	20	yes	yes		
tce_codecust	Customer code	VARCHAR	20	yes			Linked to u_id/u_externalcode from tbl_users; if you fill in this field, let tce_codepricelist=0 or "
tce_codepricelist	Pricelist code	VARCHAR	100	yes	yes		
tce_datestart	The start date to validate exchange rate	DATE		yes	yes	01/01/2000	
tce_dateend	The end date to validate exchange rate	DATE				01/01/2100	
tce_exchangerate	The VALUE of exchange rate	DOUBLE					

Il campo **tce_codecust** in chiave primaria soddisfa l'esigenza degli utilizzatori di avere tassi di cambio personalizzati per ogni listino e cliente - spesso infatti gli scambi commerciali con l'estero avvengono con valute calcolate su un tasso di cambio fisso e risultante da un accordo specifico tra l'azienda produttrice ed il cliente stesso.

Il legame tra **tbl_users** e **tbl_currency_exchange** in realtà è costituito da due relazioni per gestire le eccezioni al cambio imposto sul singolo listino. La prima tra **tbl_users_pricelist** e **tbl_currency_exchange.tce_codepricelist** ed una seconda, più specifica, tra

tbl_users.u-id (o **tbl_users.u_externalcode**, in dipendenza dall'impostazioni sul modulo eBusiness se utilizzare come identificativo cliente principale l'identificativo interno di WebProfessional o il codice proveniente dal programma gestionale dell'utilizzatore) e **tbl_currency_exchangerate.tce_codecust**.

La casistica possibile, dal caso più specifico a quello più generale, è visualizzata nella tabella sottostante.

	Currency code	Client code	Pricelist code
DB-field/index	tce_codecurrency	tce_codecust	tce_codepricelist
1.	C	X	
2.	C		P
3.	C		

Caso cliente con codice X con listino prezzi P e valuta associata C (tbl_users.u_codecurrency). Viene cercato un record rispondente ai requisiti nell'ordine presentato nella tabella: la funzione si interrompe al primo record individuato. Se nessun record viene individuato il tasso di cambio viene impostato a 1.

1. Valuta e Cliente

La ricerca avviene per codice valuta C e codice cliente X. Se esiste un record significa che il cliente X ha un tasso di cambio specifico per la valuta C.

2. Valuta e Listino

La ricerca avviene per codice valuta C e listino P. Se esiste un record significa che la classe clienti con listino P ha un tasso di cambio specifico associato al listino per la valuta C.

3. Valuta

La ricerca avviene per codice valuta C, codice cliente '0' e listino prezzi '0'. Se esiste un

record significa che la valuta C ha un tasso di cambio generico assegnato.

Sconti

Anche la scontistica, così come i listini, deriva la struttura dati dal gestionale Business. La tabella consente cinque livelli di sconti differenziabili per quantità e data di validità ed assegnabili a classi articoli, classi clienti e singoli clienti. A differenza della query su **tbl_prices** dove per individuare il prezzo articolo (unico) si utilizzavano opportuni ordinamenti e la clausola LIMIT 1, negli sconti è necessario individuare tutti gli sconti relativi al prodotto ed applicarli in sequenza al prezzo articolo.

tbl_disc_class

name	description	type	size	key	not null	default	observation
code_disc_class	Code discount class	VARCHAR	100	yes	yes	0	
type_class	Class type: o - article discount; c - client discount	CHAR	1	yes	yes	c	
title_class	Class title	VARCHAR	50		yes	untitled	title, description, notes are only administrative information
descr_class	Class description	VARCHAR	200				
notes_class	Class notes	VARCHAR	255				

tbl_discounts

name	description	type	size	key	not null	default	observation
code	Product code	VARCHAR	100	yes	yes	0	Linked to code from tbl_objects
code_disc_class	Code article discount class	VARCHAR	100	yes	yes	0	Linked to code_disc_class from tbl_disc_class where type_class=o
code_client_class	Code client discount class	VARCHAR	100	yes	yes	0	Linked to code_disc_class from tbl_disc_class where type_class=c
cust_code	Customer code	VARCHAR	20	yes	yes	0	Linked to u_id/u_externalcode from tbl_users
date_start	The start date to validate discount	DATE		yes	yes	01/01/2000	
date_end	The end date to validate discount	DATE			yes	01/01/2100	

name	description	type	size	key	not null	default	observation
qty_from	The start quantity to validate discount	DOUBLE		yes	yes	0	
qty_to	The end quantity to validate discount	DOUBLE			yes	1E+13	
discount_1	Discount 1	DOUBLE					value between 0 and 100
discount_2	Discount 2	DOUBLE					value between 0 and 100
discount_3	Discount 3	DOUBLE					value between 0 and 100
discount_4	Discount 4	DOUBLE					value between 0 and 100
discount_5	Discount 5	DOUBLE					value between 0 and 100
codtpro	Promotion code - feature not implemented	VARCHAR	50				

Il calcolo degli sconti avviene secondo le seguenti modalità:

	Product code	Client code	Product class discount	Client class discount
DB-field/index	code	cust_code	code_disc_class	code_client_class
1.	A	X	0	0
2.	A	0	0	CD
3.	0	X	AD	0
4.	0	0	AD	CD
5.	A	0	0	0
6.	0	0	AD	0
7.	0	X	0	0
8.	0	0	0	CD

Il cliente con codice X e classe di sconto CD acquista il prodotto A con classe di sconto AD. L'interrogazione al database ritorna gli sconti relativi all'articolo ordinati come in tabella, in successione - il primo caso individuato determina lo sconto applicato.

1. Sconto prodotto e cliente

La ricerca avviene per codice prodotto A e codice cliente X. Se esiste un record con que-

ste caratteristiche significa che il prodotto A ha uno sconto specifico per il cliente X. La clausola WHERE dello statement SQL sarà:

```
WHERE code = 'A' AND cust_code = 'X'
```

2. Sconto prodotto e classe sconto cliente

La ricerca avviene per codice prodotto A, codice cliente 0, classe di sconto articolo 0 e classe di sconto cliente CD. Se esiste un record con queste caratteristiche significa che il prodotto A ha uno sconto per tutti i clienti con classe di sconto cliente CD. La clausola WHERE dello statement SQL sarà:

```
WHERE code = 'A' AND cust_code = '0'  
AND code_disc_class = '0' AND code_client_class = 'CD'
```

3. Sconto classe sconto prodotto e cliente

La ricerca avviene per codice prodotto 0, codice cliente X, classe di sconto articolo AD e classe di sconto cliente 0. Se esiste un record con queste caratteristiche significa che tutti i prodotti con classe di sconto articolo AD hanno uno sconto specifico per il cliente X. La clausola WHERE dello statement SQL sarà:

```
WHERE code = '0' AND cust_code = 'X'  
AND code_disc_class = 'AD' AND code_client_class = '0'
```

4. Sconto classe sconto prodotto e classe sconto cliente

La ricerca avviene per codice prodotto 0, codice cliente 0, classe di sconto articolo AD e classe di sconto cliente CD. Se esiste un record con queste caratteristiche significa che tutti i prodotti con classe di sconto articolo AD hanno uno sconto specifico per i clienti con classe di sconto CD. La clausola WHERE dello statement SQL sarà:

```
WHERE code = '0' AND cust_code = '0'  
AND code_disc_class = 'AD' AND code_client_class = 'CD'
```

5. Sconto generico sul prodotto

La ricerca avviene per codice prodotto A, codice cliente 0, classe di sconto articolo 0 e classe di sconto cliente 0. Se esiste un record con queste caratteristiche significa che il prodotto A ha uno sconto generico per tutti i clienti. La clausola WHERE dello statement SQL sarà:

```
WHERE code = 'A' AND cust_code = '0'  
      AND code_disc_class = '0' AND code_client_class = '0'
```

6. Sconto generico sulla classe articolo

La ricerca avviene per codice prodotto 0, codice cliente 0, classe di sconto articolo AD e classe di sconto cliente 0. Se esiste un record con queste caratteristiche significa che i prodotti con classe di sconto AD hanno uno sconto generico per tutti i clienti. La clausola WHERE dello statement SQL sarà:

```
WHERE code = '0' AND cust_code = '0'  
      AND code_disc_class = 'AD' AND code_client_class = '0'
```

7. Sconto generico cliente

La ricerca avviene per codice prodotto 0, codice cliente X, classe di sconto articolo AD e classe di sconto cliente 0. Se esiste un record con queste caratteristiche significa che tutti i prodotti hanno uno sconto generico per il cliente X (quindi il cliente X ha uno sconto per tutti gli articoli che non sono inclusi in altri sconti più specifici relativi al cliente X o alla sua classe di sconto). La clausola WHERE dello statement SQL sarà:

```
WHERE code = '0' AND cust_code = 'X'  
      AND code_disc_class = '0' AND code_client_class = '0'
```

8. Sconto generico sulla classe cliente

La ricerca avviene per codice prodotto 0, codice cliente o, classe di sconto articolo 0 e classe di sconto cliente CD. Se esiste un record con queste caratteristiche significa che

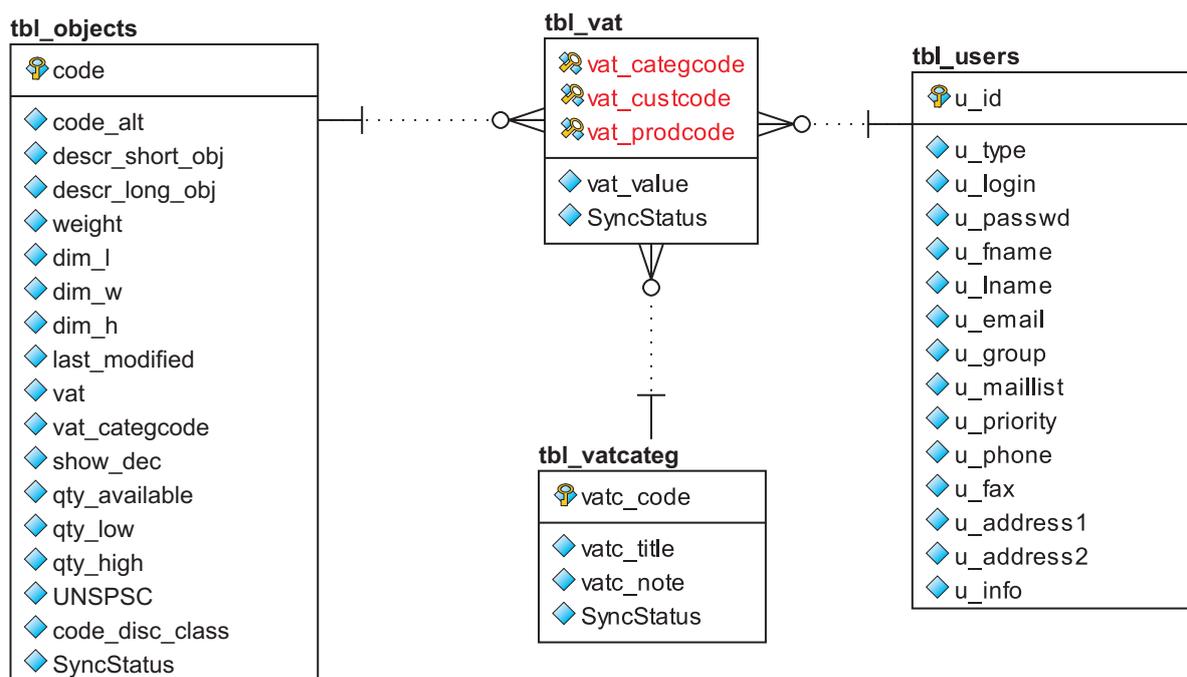
tutti i prodotti hanno uno sconto generico i clienti con classe di sconto CD (quindi i clienti con classe di sconto CD hanno uno sconto su tutti gli articoli che non sono inclusi in altri sconti più specifici relativi al cliente X o alla sua classe di sconto). La clausola WHERE dello statement SQL sarà:

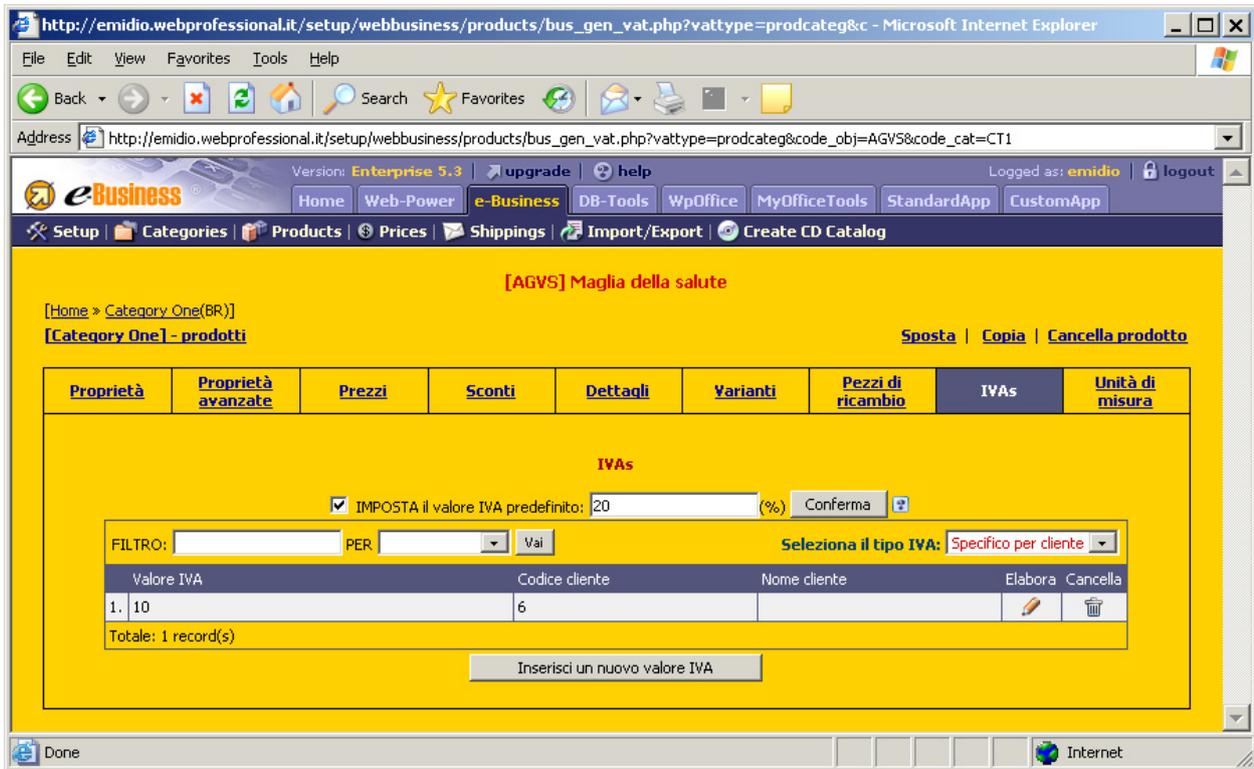
```
WHERE code = '0' AND cust_code = 'X'  
      AND code_disc_class = '0' AND code_client_class = '0'
```

Nel caso nessun record venga restituito non esistono sconti applicabili. L'ordine in cui vengono selezionati gli sconti è eventualmente personalizzabile su installazioni della piattaforma presso il cliente con minime variazioni alla funzione php di calcolo degli sconti.

Gestione dell'IVA

La prima versione di eBusiness non comprendeva una gestione IVA completa: in modo molto semplificato nella tabella articoli era presente il campo IVA che determinava il valore dell'IVA per quell'articolo. Nella realizzazione in ColdFusion l'IVA era definita come parametro associato alla piattaforma, e non era possibile impostare valori di IVA diversi





Maschera principale di gestione dell'IVA per prodotto.

per articolo.

L'esigenza di una più articolata determinazione dell'IVA per articolo/cliente interviene con i primi clienti con modulo eBusiness rivolto a clienti comunitari ed extracomunitari, oltre che a clienti nazionali ma soggetti a condizioni IVA agevolate. A seguire la struttura delle due tabelle che intervengono nella funzionalità:

tbl_vat

name	description	type	size	key	not null	default	observation
vat_catecode	Code of VAT category	VARCHAR	10	yes	yes		Linked to vatc_code from tbl_vatcateg; if you fill this field, let vat_custcode=""
vat_custcode	Customer code	VARCHAR	20	yes	yes		Linked to u_id/u_externalcode from tbl_users; if you fill this field, let vat_catecode=""
vat_prodcod	Product code	VARCHAR	50	yes	yes		Linked to code from tbl_objects
vat_value	VAT value	DOUBLE			yes	0	

tbl_vatcateg

name	description	type	size	key	not null	default	observation
vatc_code	Code of VAT category	VARCHAR	10	yes	yes		
vatc_title	Name of VAT category	VARCHAR	255				
vatc_note	Additional note about the VAT category	TEXT					

Dopo avere definito delle classi IVA (tabella **tbl_vatcateg** a scopo descrittivo) è possibile impostare valori di IVA specifici per cliente ed articolo. In **tbl_objects** (tabella articoli) rimane la possibilità di assegnare un valore IVA all'articolo sul campo **tbl_objects.vat**, utilizzato nel caso non esistano dati più specifici su **tbl_vat**. Sempre in **tbl_objects** il campo **tbl_objects.vat_categcode** in relazione con **tbl_vat.vat_categcode** introduce la facoltà di impostare per quell'articolo (o classe di articoli) un'IVA differenziata generica o per cliente. Anche nel caso di IVA generica l'utilizzo della classe IVA in sostituzione al valore percentuale nel campo **tbl_objects.vat** ha il vantaggio di poter facilmente variare l'IVA di tutta la classe articoli senza dover aggiornare i record in **tbl_objects** mantenendo la compatibilità con i sistemi già installati.

Varianti

Nella prima versione del modulo di commercio elettronico/catalogo online non era presente alcuna gestione delle varianti. La prima stesura del modulo, ancora completamente autonomo e sviluppato in ColdFusion, era rivolta ad aziende la cui produzione non consisteva in prodotti articolati e componibili, per cui si decise di evitare i costi di sviluppo di questa gestione demandandoli a quando si sarebbe effettivamente incontrata la necessità di sviluppo, basata su una precisa richiesta dei clienti.

Questo in quanto la gestione delle varianti risulta complessa e legata al tipo di prodotti gestiti - dall'esperienza derivante dal software gestionale Business, spesso la gestione delle varianti risultava macchinosa e poco soddisfacente: lo sviluppo dalla funzionalità in eBusiness sarebbe dovuta essere il più generale possibile per adattarsi ai diversi tipi di

casistiche utilizzate dai maggiori software gestionali commercializzati, il che prevedeva utilizzare soluzioni alternative e spesso in antitesi nella piattaforma per permettere poi la facile esportazione dei contenuti dal database del software gestionale a quello della piattaforma eBusiness.

La prima analisi del problema prevedeva l'utilizzo di una struttura dati complessa sviluppata sui metodi di gestione varianti di diversi software gestionali in commercio. Il sistema sviluppato, pur rispondendo ai requisiti imposti, risultava di difficile implementazione con costi di sviluppo troppo elevati paragonati alle richieste di mercato. Venne quindi adottata una soluzione di compromesso, riducendo la gestione delle varianti a quella utilizzata dal gestionale Business.

Gestione varianti nel gestionale Business

Per ogni articolo con varianti vengono generati N articoli-variante specifici determinati dalle possibili combinazioni delle varianti, con possibilità di escludere in parte la creazione dei suddetti articoli. Questo significa che per il prodotto "maglia" con varianti "colore" e "taglia" (ammettendo sei colori e sei taglie) si ottengono trentasei prodotti-variante con codice composto dal codice dell'articolo padre ed i singoli codici valore all'interno delle varianti. Ogni variante assume inoltre una "posizione" nel codice articolo per mantenere l'univocità dei codici generati.

Il porting del sistema Business su eBusiness prevedeva un massimo di dieci varianti gestibili, ogni variante con valori a codice di lunghezza fissa - questo in quanto ogni combinazione di varianti veniva gestita come un singolo articolo, il cui codice era costituito dalla

composizione del codice dell'articolo principale seguito dai codici dei valori delle varianti selezionate - la ricostruibilità dell'articolo e delle sue varianti dipendeva quindi dalla posizione dei singoli codici all'interno della stringa generata.

Questo tipo di gestione, se pure semplificava notevolmente la messa online di un catalogo basato su database del gestionale Business, dall'altro nella maggioranza dei casi non consentiva una agevole gestione delle varianti collegata a database di altri gestionali. Il numero di varianti ed il numero di valori consentiti per variante incideva notevolmente sulle dimensioni delle tabella **tbl_artroot**:

Esempio

Articolo base "schermo per video proiezione". Le varianti prese in considerazione sono:

⇒ tipo di tela

- *normale*
- *microforata*

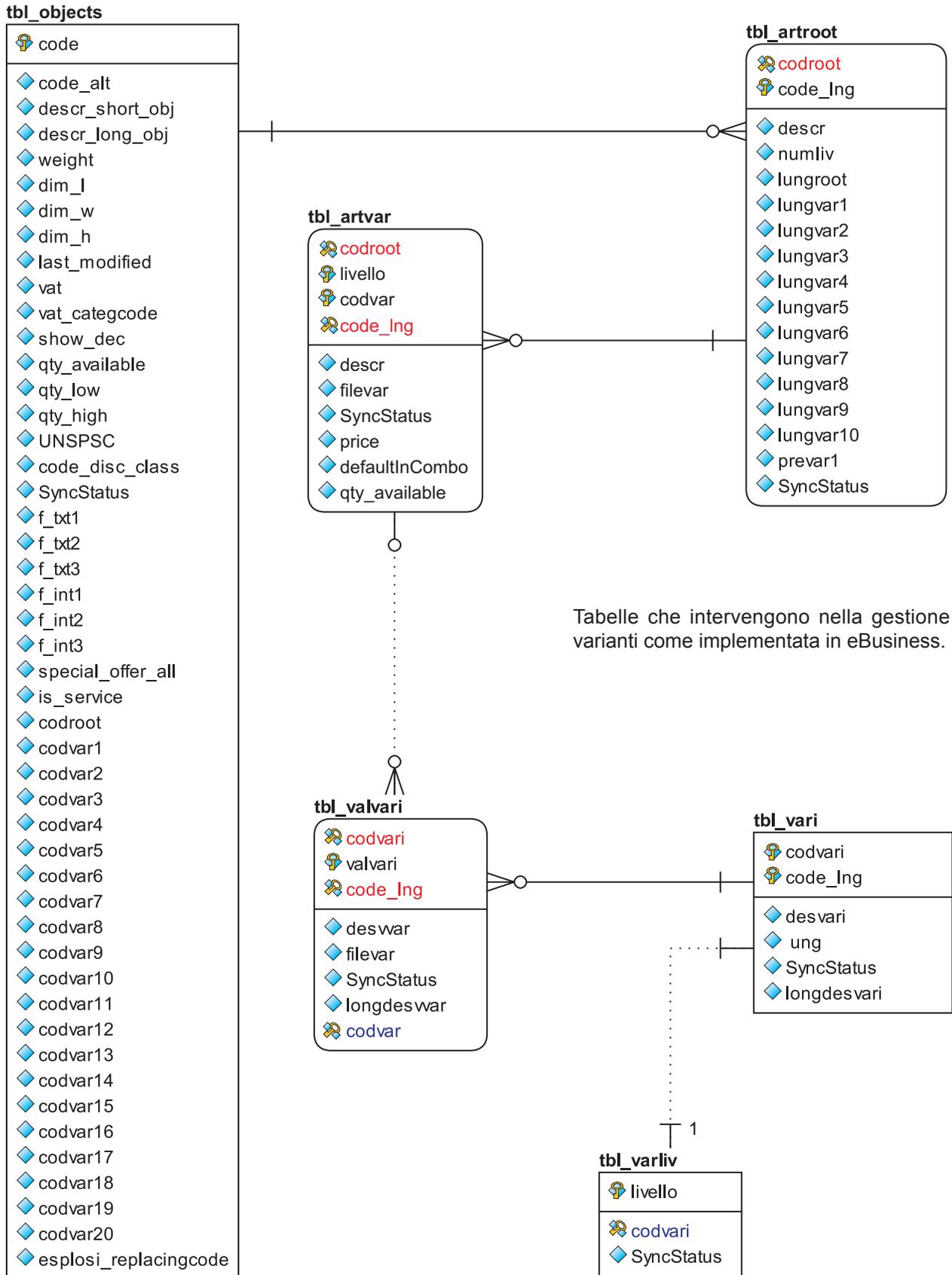
⇒ colore della struttura

- *bianco*
- *grigio*
- *nero*

⇒ dimensioni (lunghezza)

- *da 160 cm a 200 cm con step 2 cm (20 valori)*

Le tre varianti, ammettendo che tutte le combinazioni siano consentite, creano $2 \times 3 \times 20 = 120$ record in **tbl_artroot**. Accettabile per un numero di varianti ristretto e con pochi valori possibili per variante: inutile quindi a questo punto gestire dieci varianti che, se utilizzate, porterebbero alla generazione di un minimo di 2^{10} record (1.024 record) passando già a



Tablee che intervengono nella gestione varianti come implementata in eBusiness.

59.049 (3¹⁰) record considerando tre valori per ogni variante (questo per ogni articolo con varianti).

La disponibilità di una variante con questo tipo di meccanismo di generazione di "articoli fittizi" è del tutto identica alla gestione delle quantità per gli articoli reali. Le tabelle che intervengono nel sistema di varianti articolo implementato in WebProfessional sono:

tbl_vari (tipi variante)

name	description	type	size	key	not null	default	observation
codvari	Variant type code	VARCHAR	50	yes	yes		
code_lng	Language code (e.g.: IT, EN, FR, DE, ES, RO)	CHAR	2	yes	yes	IT	ISO code of the language - uppercase
desvari	Type name	VARCHAR	30				e.g.: colour
lung	Length of values code	SMALLINT	2		yes	0	
longdesvar ^{5.3}	Long description of variant type	TEXT					

tbl_valvari (valori varianti)

name	description	type	size	key	not null	default	observation
codvari	Variant type code	VARCHAR	4	yes	yes		Linked to codvari from tbl_vari
code_lng	Language code (e.g.: IT, EN, FR, DE, ES, RO)	CHAR	2	yes	yes	IT	ISO code of the language - uppercase
valvari	Variant value code	VARCHAR	50	yes	yes		
desvvar	Name of the variant value	VARCHAR	40				e.g.: red, blue
filevar	File or colour of the variant value	VARCHAR	50				could be the name of an image (mylmg.gif) or a colour (#777777)
longdesvar ^{5.3}	Long description of variant value	TEXT					

tbl_artroot (info prodotto, lunghezza codici varianti, gestione prezzi)

name	description	type	size	key	not null	default	observation
codroot	Code of the root	VARCHAR	12	yes	yes		Linked to codroot from tbl_objects
code_lng	Language code (e.g.: IT, EN, FR, DE, ES, RO)	CHAR	2	yes	yes	IT	ISO code of the language - uppercase
descr	Description of the root	VARCHAR	40		yes		
numliv	Number of levels for the root	SMALLINT	2		yes	0	

name	description	type	size	key	not null	default	observation
lungroot	Length of the root	SMALLINT	2				
lungvar1	Length of the code for the level variant 1	SMALLINT	2		yes	0	
lungvar2	Length of the code for the level variant 2	SMALLINT	2		yes	0	
lungvar3	Length of the code for the level variant 3	SMALLINT	2		yes	0	
prevar1	Flag for prices administration (one of the values: 'N' - common prices for all variants, 'S' - different prices)	VARCHAR	1		yes	N	

The screenshot shows the eBusiness web application interface. The browser address bar displays the URL: `http://emidio.webprofessional.it/setup/webbusiness/var_products/varprod_index.php?code_obj=AGVS&code_cat=CT1&act=analit`. The application header includes the eBusiness logo, version information (Enterprise 5.3), and navigation tabs such as Home, Web-Power, e-Business, DB-Tools, WpOffice, MyOfficeTools, StandardApp, and CustomApp. The main content area is titled "[AGVS] Maglia della salute" and contains a breadcrumb trail: "[Home > Category One(BR)] [Category One] - prodotti".

Below the breadcrumb, there are navigation links: "Sposta", "Copia", and "Cancella prodotto". A horizontal menu contains tabs for "Proprietà", "Proprietà avanzate", "Prezzi", "Sconti", "Dettagli", "Varianti", "Pezzi ricambio", "IVAs", and "Unità di misura". The "Varianti" tab is active, showing "Impostazioni varianti | Elenco Varianti".

The "Elenco Varianti" section includes a form for creating a new variant. The form has two levels: "Livello 1 -- [CLR] Colore" with a dropdown menu set to "[BL] Blu" and "Livello 2 -- [TGL] Taglia" with a dropdown menu set to "[L] Large". There are "Gestione valori varianti" links for each level and a "Conferma" button.

Below the form is a table listing existing variants:

Livello 1: Colore	Livello 2: Taglia	Codice prodotto	Descrizione prodotto	Elabora	Cancella	Proprietà prodotto
BL	L	AGVSBLL	Maglia della salute [Colore: Blu Taglia: Large]			[Proprietà prodotto]
BL	M	AGVSBLM	Maglia della salute [Colore: Blu Taglia: Medium]			[Proprietà prodotto]
BL	XL	AGVSBLXL	Maglia della salute [Colore: Blu Taglia: Extra Large]			[Proprietà prodotto]
NE	L	AGVSNEL	Maglia della salute [Colore: Nero Taglia: Large]			[Proprietà prodotto]
VE	L	AGVSVEL	Maglia della salute [Colore: Verde Taglia: Large]			[Proprietà prodotto]

tbl_artvar (valore variante per ogni articolo principale e livello)

name	description	type	size	key	not null	default	observation
codroot	Code of the root	VARCHAR	12	yes	yes		Linked to codroot from tbl_objects
code_lng	Language code (e.g.: IT, EN, FR, DE, ES, RO)	CHAR	2	yes	yes	IT	ISO code of the language - uppercase
livello	Variant level	SMALLINT	2	yes	yes	0	
codvar	Variant value code	VARCHAR	50	yes	yes		linked to codvar1 from tbl_objects if livello = 1; codvar2 for livello=2; codvar3 for livello=3
descr	Variant value description	VARCHAR	40				
filevar	Variant file	varchar	255				File must be uploaded on the server
price ^{5.3}	Difference of price	double					Used only when option 'Configuratore varianti' is enabled
defaultInCombo ^{5.3}	Default value selected in combo	char	1				Used only when option 'Configuratore varianti' is enabled

tbl_varliv (relazione il livello variante con il nome tipo)

name	description	type	size	key	not null	default	observation
codroot	Code of the root	VARCHAR	12	yes	yes	0	Linked to codroot from tbl_objects
livello	Variant level	SMALLINT	2	yes	yes	1	
codvari	Variant type code	VARCHAR	50	yes	yes		Linked to codvari from tbl_vari

Nelle tabelle sono segnati in grigio i campi introdotti nella versione 5.3 di WebPower per la gestione delle varianti attraverso il "configuratore" che verrà analizzata successivamente. L'interfaccia di amministrazione comprende una sezione preposta alla creazione di articoli con varianti (tuttora in essere). All'atto dell'inserimento di un nuovo articolo è indispensabile definire se l'articolo avrà varianti oppure no. Una volta creato un articolo non definito come articolo con varianti non è possibile (da interfaccia web) modificarlo aggiungendo varianti. L'inserimento di un prodotto con varianti comporta i seguenti step:

1. in **tbl_vari** va inserito il tipovariante generico; la tabella è descrittiva del tipo variante - il nome del tipo viene ereditato da tutte le composizioni che utilizzano quel tipo variante.

Nel caso un prodotto utilizzi un tipo variante non specificato in questa ttabella, verrà comunque visualizzato dall'interfaccia web come "tipo: x".

Ordini

Indietro alla lista degli ordini Ordine successivo >>

Completa ordine Cancella Stampa

Numero ordine: 32269287

Serie: *

Data ordine: 29 Ottobre 2005 22:13

Cliente: 25 CIPPINA SAS

Nome cliente: Nara Bellini

Listino: _____

Agente: _____

Cond. pagamento: _____

Valuta: _____

Tasso di cambio: 1

Data consegna: 29 Ottobre 2005

Numero di Gestionale: 0

Salva le modifiche: **Conferma**

Stato ordine

Pagato online

Accetta evasione parziale

Convalidato

Completo

File AND URL

Indirizzo (URL): _____

File per scarico: _____

-	Cod.articolo	Cod.alt	Nome prodotto	UM	Quantità richiesta	Quantità approvata	Inviato	Sconto	Prezzo	Prezzo valuta	Totale netto	Totale valuta	IVA	Categoria IVA	Completo
1.	AGV5		Maglia della salute	PZ	67	67	45	0,00%	34,00	0,00	2278,00	0,00	20%		<input type="checkbox"/>
2.	AGVSVEL		Maglia della salute [Colore: Verde Taglia: Large]	PZ	2	2	0	0,00%	34,00	0,00	68,00	0,00	0%		<input type="checkbox"/>
3.	000001		Prodotto Nuovo	Kg	23	23	7	0,00%	10,00	0,00	230,00	0,00	0%		<input type="checkbox"/>

Destinatario: Nara Bellini
Ragione sociale: CIPPINA SAS
Partita IVA / Codice fiscale: 23232445455 /
E-mail: cippina@lippina.com
Telefono: 3422222

Nota: _____

Totali

	In valuta	
Netto:	2576,00	0,00
Spese:	0,00	0,00
Imponibile:	2576,00	0,00
IVA:	455,60	0,00
Prezzo totale:	3031,60	0,00

Salva le modifiche: **Conferma**

[Nuova ricerca]

2. in **tbl_valvari** si inserisce opzionalmente i valori possibili della variante per ogni tipo presente in **tbl_vari**.
3. in **tbl_artroot** va inserito un record con **codroot** uguale al codice del prodotto principale, **numliv** uguale al numero di tipi variante per quell'articolo. Inoltre da 1 a **numliv** vanno specificati i campi **lungvar** (lunghezza variante) inserendo la lunghezza massima dei codici valori variante ammessi per quel tipo. Questo campo ha particolare rilevanza perchè viene utilizzato per la costruzione del codice articolo composto dalle scelte fatte sulle varianti.
4. in **tbl_varliv** contiene le associazioni tra i livelli ed i tipi categoria.
5. vengono inseriti in **tbl_objects** e **tbl_objects_Ing** tanti record quante sono le combinazioni possibili divarianti per l'articolo principale.
 - il campo **codroot** contiene il codice dell'articolo principale
 - per ogni variante il codice valore variante associato, opzionalmente da **tbl_artvar**Se in **tbl_artvar** non è presente alcun record con il codice valore variante **codvar** specificato nell'interfaccia verrà visualizzato il valore di **codvarx**. Per ogni inserimento in **tbl_objects** va inserito anche il corrispondente record in **tbl_objects_Ing**.
Va notato come in **tbl_objects** esistono solo tre campi **codvarx** (**codvar1**, **codvar2**, **codvar3**) in quanto non vengono mai gestite più di tre varianti con il metodo della generazione di un articolo per ogni combinazione possibile.

Questa modalità di operare ben si adatta alla gestione delle varianti come sigolo articolo (ogni combinazione viene interpretata come un nuovo articolo e da origine ad un record nella tabella **tbl_objects**), ma non è accettabile quando le varianti su un articolo vengono determinate dall'aggiunta o meno di componenti esterni, spesso inseribili in articoli diversi.

Un esempio tipico viene dalle aziende di tende per esterni. Lo stesso modello di tenda può montare o meno una serie di componenti addizionali (motore, sensore pioggia, sensore vento,...) che determinano le varianti del prodotto. La quantità disponibile dipende dal minimo della disponibilità dell'articolo stesso e della disponibilità dei singoli componenti:

Tenda modello "SUN-SHIELD" - cod. SSH-01

La quantità disponibile per il codice SSH-01-MA (Tenda modello "SUN-SHIELD" motorizzata) dipende dal minimo della quantità disponibile del modello base, del componente motore e dei componenti sensori.

Questo tipo di gestione delle varianti porta inoltre all'esigenza di avere una sorta di configuratore tramite cui selezionare le varianti/moduli opportuni, verificandone il prezzo e la disponibilità. Per questo nella versione 5.3 vennero introdotti alcuni campi nelle tabelle esistenti:

- **tbl_valvari.longdesvvar** per la descrizione estesa di una variante
- **tbl_artvar.price** per definire la differenza in prezzo che determina l'aggiunta o meno della opzione della variante - la differenza può essere sia positiva che negativa
- **tbl_artvar.defaultInCombo** specifica se l'opzione va visualizzata come default nel controllo tipo <SELECT> del configuratore

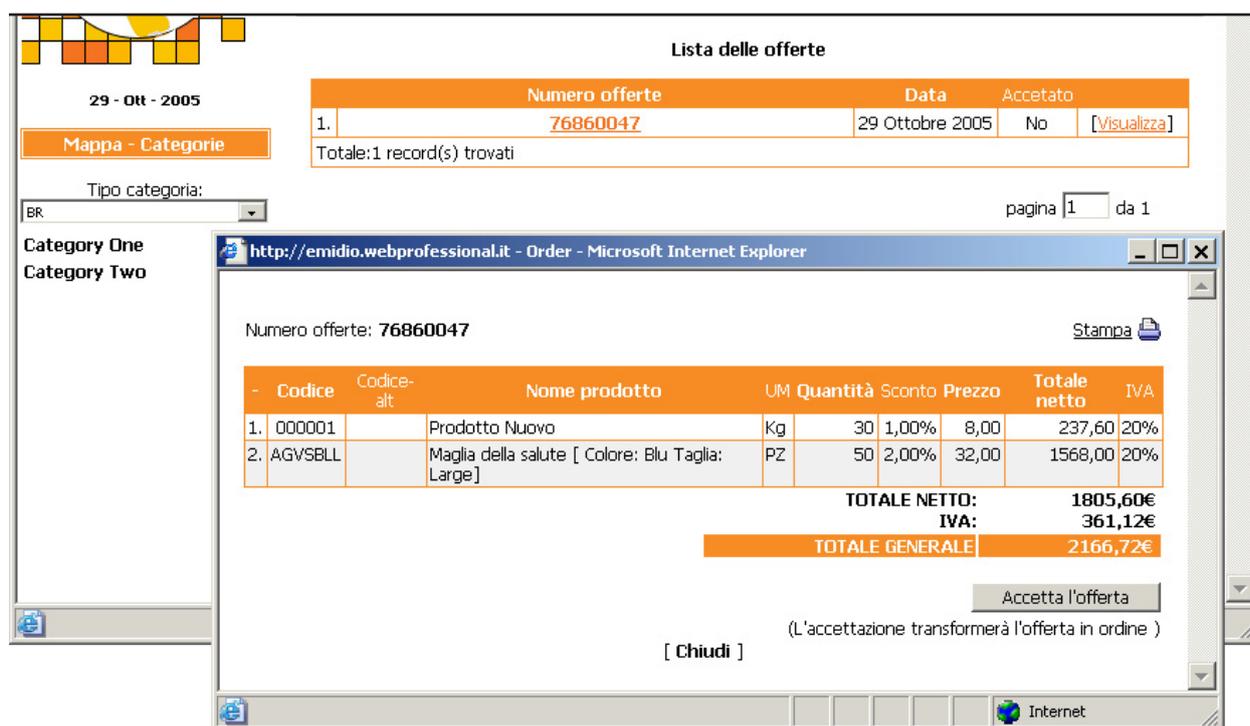
Oltre a queste modifiche venne creata la tabella **tbl_ec_varexcluded** per poter inserire i valori delle combinazioni non consentite nel configuratore (con possibilità di definizione dei valori a due a due, in mutua esclusione).

tbl_ec_varexcluded (combinazioni varianti non consentite)

name	description	type	size	key	not null	default	observation
codroot	Code of the root	VARCHAR	100	yes	yes		
codvari_1	First part of the comb. - code of the variant type	VARCHAR	50	yes	yes		
valvari_1	First part of the comb. - code of the variant value	VARCHAR	50	yes	yes		
codvari_2	Second part of the comb. - code of the variant type	VARCHAR	50	yes	yes		
valvari_2	Second part of the comb. - code of the variant value	VARCHAR	50	yes	yes		

Catalogo su CD

eBusiness consente la creazione di un catalogo statico in HTML per la masterizzazione su CD (comprensivo di file autorun.inf). Le pagine dinamiche vengono replicate per ogni visualizzazione e prodotto reindirigendo l'output dell'interprete php su file. Vengono escluse dalla procedura di creazione catalogo statico le parti dinamiche (ricerca) che non avrebbero modo di funzionare da CD. I file vengono restituiti in un singolo archivio tar.gz.



The screenshot shows a web application interface with two main sections. The top section, titled "Lista delle offerte", displays a table of offers. The bottom section, titled "Numero offerte: 76860047", provides a detailed view of a specific offer, including a table of products and their prices.

Lista delle offerte

Numero offerte	Data	Accettato
1. 76860047	29 Ottobre 2005	No [Visualizza]

Totale: 1 record(s) trovati

pagina 1 da 1

Numero offerte: 76860047 [Stampa](#)

-	Codice	Codice-alt	Nome prodotto	UM	Quantità	Sconto	Prezzo	Totale netto	IVA
1.	000001		Prodotto Nuovo	Kg	30	1,00%	8,00	237,60	20%
2.	AGVSBLL		Maglia della salute [Colore: Blu Taglia: Large]	PZ	50	2,00%	32,00	1568,00	20%

TOTALE NETTO: 1805,60€
IVA: 361,12€
TOTALE GENERALE: 2166,72€

[Accetta l'offerta](#)
(L'accettazione transformerà l'offerta in ordine)

[\[Chiudi \]](#)

Ordini e Fatture

Nella sezione di gestione ordini vengono messi a disposizione dell'utente gli strumenti di base per la validazione di un ordine e delle quantità richieste e la gestione della spedizione in uno o più lotti. Una maschera di ricerca visualizza l'elenco degli ordini per cliente, articolo, agente, intervallo di data (è presente anche una maschera di ricerca avanzata con opzioni aggiuntive sullo stato dell'ordine).

Ordini 

Lista ordini

Numero ordine	Serie	Nome cliente	Azienda	Data ordine	Totale	Pagato online	Completo	Esportato	
1. 71811828	*	Nara Bellini	CIPPINA SAS	30 Ott 2005	659.60				<input type="checkbox"/>
2. 32269287	*	Nara Bellini	CIPPINA SAS	29 Ott 2005	3,031.60		<input checked="" type="checkbox"/>		<input type="checkbox"/>
3. 76860046	*	Emi Latzco	Aggio Spa	29 Ott 2005	136.00		<input checked="" type="checkbox"/>		<input type="checkbox"/>

Totale: 3 record(s) **Cancel**

[Cancella tutti gli ordini compiuti](#) [Cancella tutti gli ordini esportati](#)

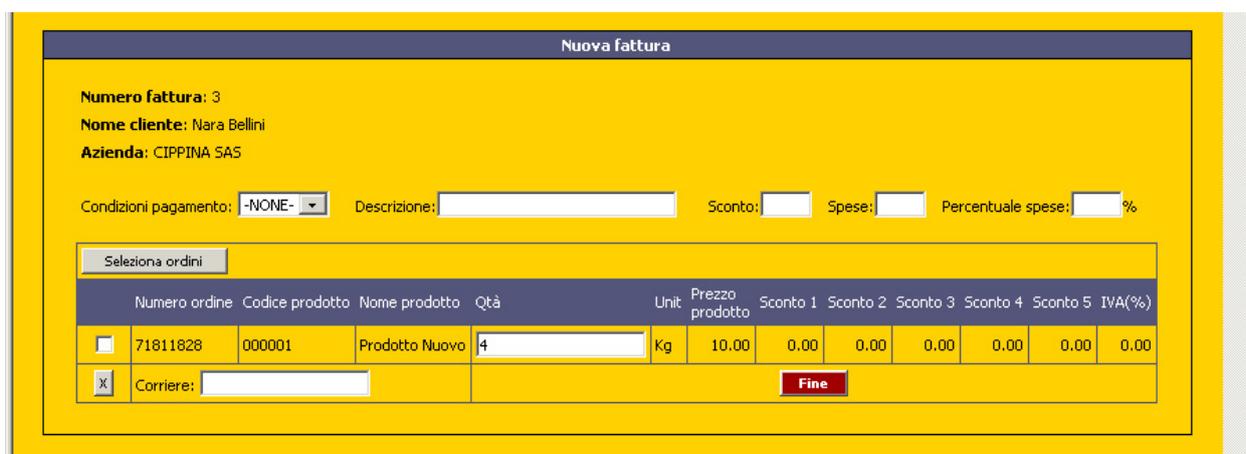
[[Nuova ricerca](#)]

Selezionando il singolo ordine si accede alla schermata di dettaglio, con le informazioni di testata, lo stato, l'elenco degli articoli, le quantità, i prezzi, le quantità approvate ed evase.

La convalida, come la compilazione di un ordine, sono operazioni irreversibili. L'inserimento della spedizione dell'ordine (totale o parziale) è possibile dal pulsante "Completa ordine" (di default viene proposta l'evasione totale). Per convalidare la spedizione di un prodotto è necessario che le giacenze di tale prodotto consentano la diminuzione della quantità specificata nella spedizione. Le giacenze sono rappresentate dal campo "Disponibile" nella sezione "Proprietà avanzate". Il valore della disponibilità verrà aggiornato ogni volta che si spedisce un quantitativo di un determinato prodotto.

Nei dettagli di un ordine verranno eventualmente visualizzate anche la nota del cliente alla spedizione dell'ordine e la condizione di pagamento corrispondente a tale cliente.

eBusiness da la possibilità di generare documenti pro-forma per la fatturazione basati sugli ordini. Dal menù "Fatture" viene specificato un cliente ed un numero di fattura (il sistema propone un progressivo automaticamente). Selezionato l'utente verrà visualizzato l'elenco degli ordini aperti relativi. Non è possibile generare una fattura se il cliente selezionato non ha fatto almeno un ordine. Ogni fattura può essere relativa ad un unico ordine.



Numero ordine	Codice prodotto	Nome prodotto	Qtà	Unit	Prezzo prodotto	Sconto 1	Sconto 2	Sconto 3	Sconto 4	Sconto 5	IVA(%)
<input type="checkbox"/>	71811828	000001	Prodotto Nuovo	4	Kg	10.00	0.00	0.00	0.00	0.00	0.00

La schermata proporrà le righe con gli articoli in ordine con i campi necessari a specificare la quantità da fatturare. Una volta emessa la fattura le quantità andranno a scalare dall'ordine principale: nel caso risulti completamente evaso verrà automaticamente chiuso.

Offerte

Il sistema consente la generazione di offerte per gli utenti registrati. Una offerta consiste in un elenco di articoli a cui l'amministratore impone un prezzo od uno sconto particolare, specificandone le quantità.

L'offerta verrà presentata all'utente dopo il login nella sezione Offerte ed opzionalmente inviata all'utente stesso via email. In caso di accettazione l'offerta viene trasformata in un ordine eBusiness per poi seguire il normale iter degli ordini. I prezzi specificati vengono inseriti direttamente nell'ordine e possono essere diversi da quelli del listino associato.



Una maschera di ricerca permette di visualizzare le offerte in essere (una volta accettata dall'utente l'offerta viene inserita nella sezione ordini e cancellata dalla sezione offerte) per nome cliente, codice prodotto, data di creazione. Se non viene specificato nessun parametro si avrà una lista di tutte le offerte esistenti.

Condizioni di pagamento

eBusiness gestisce modalità di pagamento personalizzabili per utente. Le condizioni e le modalità di pagamento vengono memorizzate nella tabella `tbl_payment_condition`, unitamente alle informazioni necessarie ad eventuali procedure correlate (pagamento con carta di credito).

`tbl_payment_condition`

name	description	type	size	key	not null	default	observation
code_payment	Code payment condition	VARCHAR	100	yes	yes	0	

name	description	type	size	key	not null	default	observation
cond_target	CONDITION TARGET	CHAR	1		yes	S	Accepted values: S - for some specified users W - for all users who are not provided with a payment condition A - for all users, no matter if they have or not a payment condition associated
pay_description	Payment condition description	VARCHAR	255				
cond_maxsumexpenses	Minimum order sum to include shipping fees	DOUBLE					
start_after_days	Start after number of days	INT	3				
number_rates	Number of rates	INT	3				
days_betw_rates	Number of days between rates	INT	3				
discount	Discount	DOUBLE					between 0 and 100
expenses	Expenses - fixed amount	DOUBLE					
cond_vatexpenses	VAT Expenses	FLOAT					between 0 and 100
perc_expenses	Percentage expenses	DOUBLE					between 0 and 100
cond_minsumorder	The MINIMUM of total sum of the order, for the payment condition to be applied	DOUBLE					if none of cond_minsumorder or cond_maxsumorder is not set then it will be no restriction to apply this payment condition
cond_maxsumorder	The MAXIMUM of total sum of the order, for the payment condition to be applied	DOUBLE					if none of cond_minsumorder or cond_maxsumorder is not set then it will be no restriction to apply this payment condition
cond_notes	Notes	TEXT					
cond_creditcardid	The credit card id which will also enable the online payment with credit card	TINYINT	3				If the credit card system is enabled any order from public area will be registered only if it is confirmed as paid by the appropriate Payment System.

L'associazione tra condizioni di pagamento ed utenti viene specificata nella tabella **tbl_payment_cust**. Dopo l'avvenuta registrazione nel sistema (login) all'atto dell'ordine verranno proposte le condizioni di pagamento di default e/o le condizioni di pagamento particolari associate all'utente specifico.

tbl_payment_cust

name	description	type	size	key	not null	default	observation
code_cust	Customer code	VARCHAR	20	yes	yes	0	Linked to u_id/u_externalcode from tbl_users
code_payment	Code payment condition	VARCHAR	100	yes	yes	0	Linked to code_payment from tbl_payment_condition
min_sumorder	The MINIMUM of total sum of the order specific per user, for the payment condition to be applied	DOUBLE					Has a higher priority then the default MINIMUM sum of the payment condition
max_sumorder	The MAXIMUM of total sum of the order specific per client, for the payment condition to be applied	DOUBLE					Has a higher priority then the default MAXIMUM sum of the payment condition

Le condizioni di pagamento con **code_cust** = 0 sono da ritenersi generiche ed applicate a tutti gli utenti secondo le modalità definite dal campo **tbl_payment_condition.cond_target**.

Descrizione	una descrizione generale della modalità di pagamento
Sconto	indica un eventuale sconto da applicare all'intero ordine
Numero rate	un numero compreso fra 1 e 60 indica il numero delle rate che compongono il pagamento (una sola rata se non specificato)
Spese	descrizione delle spese
Percentuale spese	percentuale percepita per le eventuali spese
Giorni fra rate	questo campo viene utilizzato solo per le modalità di pagamento che prevedono almeno 2 rate
Inizia dopo n giorni	il numero di giorni dopo il quale viene incassata la prima rata

È possibile creare delle condizioni pagamento che non siano personalizzate per determinati clienti. Queste condizioni pagamento verranno applicate a tutti i clienti senza una

condizione pagamento specifica.

Ad un cliente è possibile associare più condizioni pagamento, caso in cui egli potrà sceglierne una.

Lo sconto, le spese (come importo fisso e/o come percentuale) intervengono nella determinazione del totale dell'ordine.

Importazione ed esportazione dati

Lo startup del modulo eBusiness richiede normalmente una prima importazione dati per caricare le anagrafiche articoli, clienti, ed i prezzi esportati dall'applicativo gestionale del cliente. Inoltre, nel caso di B2B e B2C è inoltre necessaria una esportazione dei record relativi agli ordini per il successivo inserimento nel sistema.

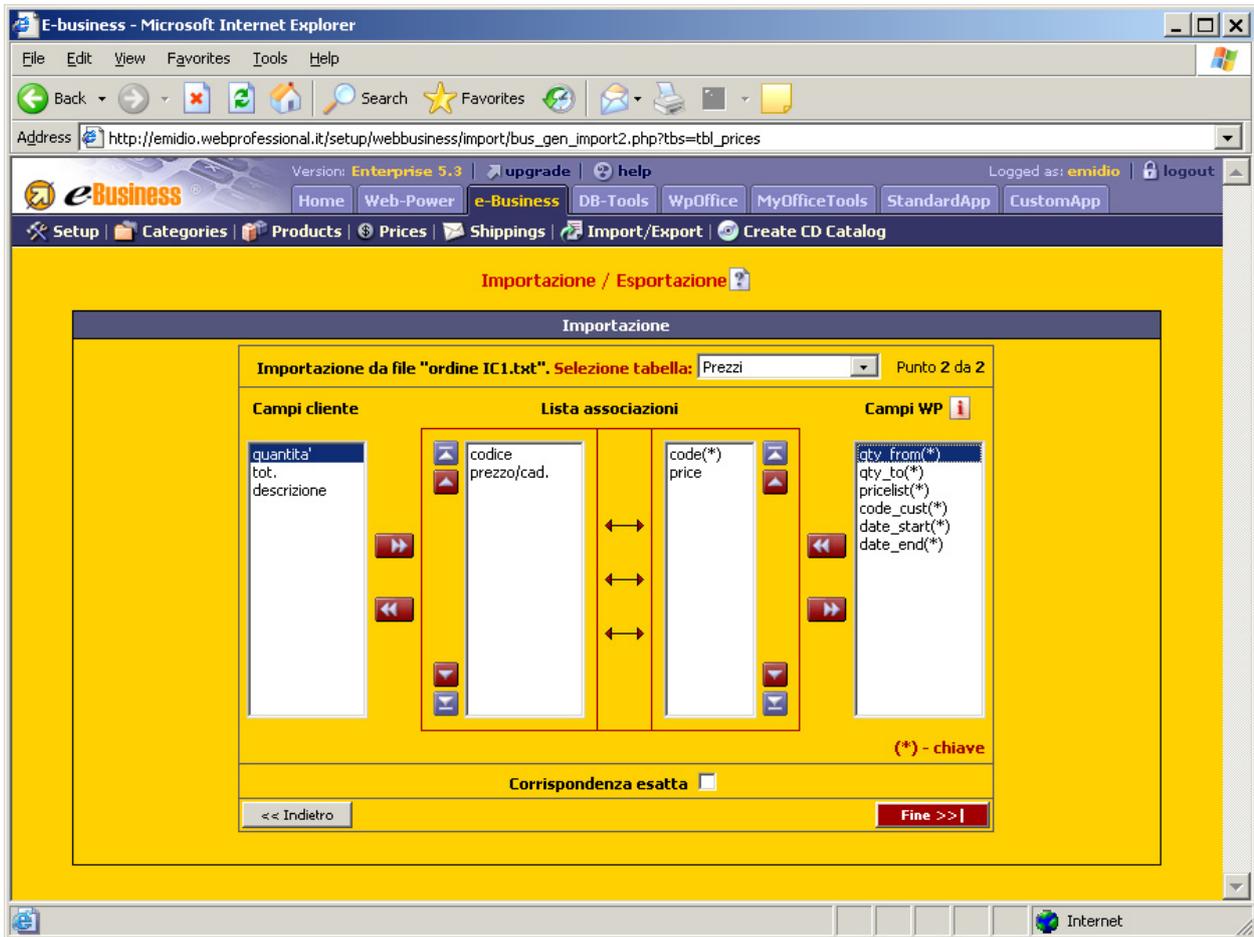
Importazione dati

Questo strumento permette l'importazione di dati da un file testo, tipicamente esportato dal sistema gestionale del cliente. E' possibile in questo modo pubblicare su web prodotti, prezzi, clienti e prodotti aggiuntivi una tantum. Nel caso gli aggiornamenti siano frequenti è invece consigliata la sincronizzazione dati attraverso DBSync.

L'operazione di importazione viene eseguita in due fasi:

1. impostazione dei parametri di importazione (file, separatori, caratteri speciali)
2. selezione della tabella dove verranno aggiunte le registrazioni e la corrispondenza campi.

Una volta eseguito l'upload del file sul server WebProfessional (tramite la form di caricamento file presente nel modulo di importazione) si procede definendo i delimitatori di



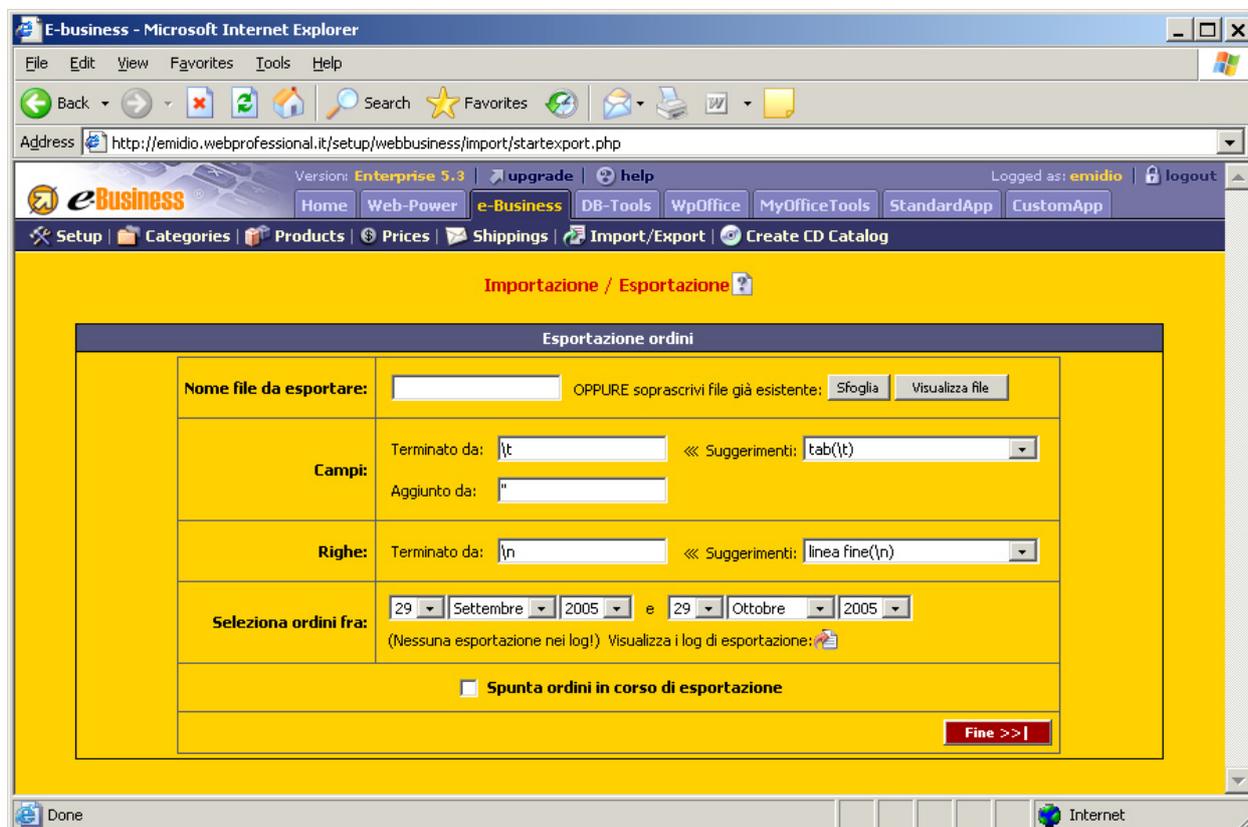
campo e record per il file caricato. La prima riga deve contenere i nomi dei campi, ed il qualificatore di testo, se presente nei valori dei campi, deve essere preceduto dal carattere di escape "\". Nell'esempio a seguire nel valore dei campi troveremo sia il carattere di escape che il carattere separatore di campo.

- 1,"una riga","100.20"
- 2,"una riga contenente una, virgola","102.20"
- 3,"una riga contenente \" apici","102.20"
- 4,"una riga contenente \", apici e virgola","102.20"

Le opzioni "sostituisci" e "ignora" definiscono la priorità del record in caso di conflitto sulla

chiave primaria. Se viene specificato "sostituisci", i record importati sostituiranno i record con lo stesso valore della chiave. Se viene specificato "ignora" i record con violazione di chiave verranno omessi dall'importazione.

Il secondo step nell'importazione dei dati è costituito dalla scelta dell'anagrafica e della corrispondenza dei campi. Le anagrafiche disponibili per l'importazione dei dati sono: anagrafica prodotti, anagrafica clienti, prezzi e prodotti aggiuntivi. La prima riga del file di importazione servirà come base nella scelta della corrispondenza campi -per questo motivo è consigliabile che contenga il nome dei campi. L'interfaccia consente di stabilire una corrispondenza tra i campi presenti nel file e quelli della tabella di eBusiness di destinazione. Nel caso dei prodotti le tabelle di destinazione sono in realtà due: `tbl_objects` e `tbl_objects_ing`. La procedura presenta comunque un unico step in fase di selezione della tabella, provvedendo poi a compilare `tbl_objects_ing` utilizzando come lingua nel campo `code_ing` quella predefinita del sito. Se il file contiene tutti i campi della tabella



E-business - Microsoft Internet Explorer

Address: http://emidio.webprofessional.it/setup/webbusiness/import/startexport.php

Version: Enterprise 5.3 upgrade help Logged as: emidio logout

Home Web-Power e-Business DB-Tools WpOffice MyOfficeTools StandardApp CustomApp

Setup Categories Products Prices Shippings Import/Export Create CD Catalog

Importazione / Esportazione

Esportazione ordini

Nome file da esportare: OPPURE sovrascrivi file già esistente:

Campi: Terminato da: <<< Suggerimenti:

Aggiunto da:

Righe: Terminato da: <<< Suggerimenti:

Seleziona ordini fra: e

(Nessuna esportazione nei log!) Visualizza i log di esportazione:

Spunta ordini in corso di esportazione

nell'ordine specificato, è possibile stabilire automaticamente la corrispondenza spuntando l'opzione "Corrispondenza esatta". L'operazione finisce premendo il pulsante "Fine".

Esportazione ordini

Il modulo di esportazione permette di salvare gli ordini registrati su web in un file testo per la successiva importazione nel sistema gestionale con procedure automatizzate specifiche per cliente.

Nella schermata di esportazione ordini è possibile selezionare i record da esportare per intervallo di data, e specificare i parametri per generare il file di testo. I valori di default vedono \t come separatore di campo, \n come separatore di record e " " come qualificatore di testo. Tali valori sono comunque modificabili dall'utente in fase di esportazione.

Conclusioni

Il modulo eBusiness risponde all'esigenza delle aziende di esporre la propria offerta di prodotti (o servizi) con possibilità di vendita online B2B e B2C. La scelta della modalità è a discrezione dell'amministratore, consentendo lo startup come catalogo online per passare poi alle funzionalità complete per gli utenti registrati (B2B) ed eventualmente al pubblico (B2C). La possibilità di integrazione dati con il sistema gestionale (attraverso lo strumento di sincronizzazione DB Sync) rende il modulo eBusiness una importante integrazione a WebPower completando la presentazione di contenuti con servizi di ecommerce.

Cenni su DbTools e WpOffice

La piattaforma WebProfessional viene completata dai moduli DbTools e WP Office per la gestione dei contenuti che non trovano una logica collocazione in WebPower ed eBusiness.

Il modulo **WP Office** è dedicato alla gestione delle informazioni relative alla attività lavorativa quotidiana quali agenda, task, rapportini di intervento, con integrato un sistema di webmail che si appoggia e caselle esterne accesibili via POP3.

DbTools si presenta invece come uno strumento estremamente flessibile per il content management di informazioni generiche strutturate in una base di dati. L'accesso ai dati avviene attraverso procedure native presenti in PHP per i database più diffusi (MySQL, SQL Server, PostgreSQL, DB2) o tramite connessione ODBC.

Il modulo DbTools

DbTools consente di pubblicare dati aziendali con collegamento al database di origine o tramite replica dei dati stessi su database MySQL, effettuare query di interrogazione, impostare strutture, permessi e modalità di visualizzazione. È possibile inserire ed aggiornare dati in modo manuale oppure automatico tramite l'apposito strumento di sincronizzazione abbinato al modulo DbTools, DBSync. Le funzioni del modulo si articolano in:

Tabelle e "relazioni" - Il sistema permette di definire tabelle ed il formato e tipo dei campi e di stabilire legami logici tra i dati delle tabelle (il termine "relazioni" non fa riferimento alle relazioni propriamente dette implementate direttamente sul database).

Sincronizzazione dati ODBC - La possibilità di sincronizzare dati via ODBC apre la strada ad una possibilità pressochè illimitata di gestione delle informazioni che si vogliono pubblicare sul Web. Oltre al supporto nativo dei principali database DbTools si appoggia a driver ODBC per il collegamento a fonti dati eterogenee.

Ricerche - Dati - E' possibile impostare ricerche e interrogazioni, anche molto articolate, attraverso il wizard di creazione query. Questa funzione permette di scegliere le tabelle ed i

campi, effettuare le ricerche e definire in quale formato visualizzare i dati. E' inoltre possibile definire per l'utente accessi ai dati in modifica, cancellazione, inserimento, subordinando le operazioni e la visibilità dei dati alla verifica di condizioni su login e gruppi.

Visualizzazione dati - Attraverso template predefiniti si impostano le diverse modalità di visualizzazione dei dati.

Utenti registrati - L'accesso alle varie funzioni dell'applicazione Web realizzata può essere differenziato per utenti pubblici o registrati; la gestione dei profili utente con relativi privilegi è gestita in modo integrato all'interno dell'ambiente Web Professional.

Configurazione di un database

DbTools si collega ad un database esistente per la gestione delle strutture dati e delle informazioni, mettendo a disposizione dell'utente strumenti per la pubblicazione e la manipolazione dei dati. DbTools si appoggia di default ad un database MySQL vuoto che viene creato all'attivazione del modulo. E' comunque possibile definire altre connessioni a database diversi. Va specificato il Server/Indirizzo IP, la porta se necessaria, il login di accesso al database. Vengono supportati in modo nativo (utilizzando le funzioni di PHP) MySQL, MS SQL Server, PostgreSQL e DB2, ed è possibile in alternativa utilizzare un collegamento ODBC.



No	Server/indirizzo IP	Porta	Login	Tipo di server	Base di dati	Stato	Attivo	
0	dbserver		romteam	mysql	wp3_dbtools_emidio_webprofessional_it			
1	dbserver	3306	romteam	mysql	wp2_emidio_webprofessional_it			

Buttons: Aggiungi, Scannerizza, Cancella

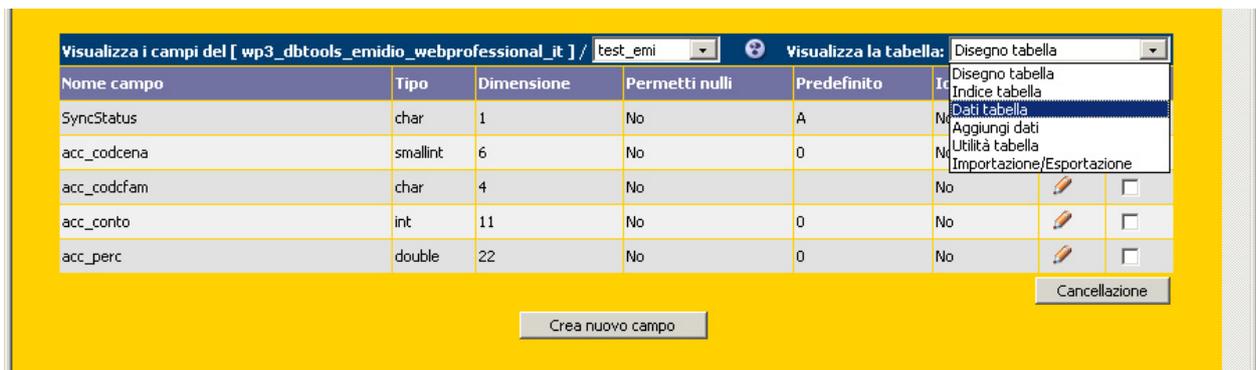
Selezionando il database da link sulla colonna "Basi di dati" si accede direttamente all'elenco delle tabelle presenti.



Nome tabella	Registrazione tabelle	Data/ora creazione	
test_emi	0	2005-12-01 15:40:46	
tvguide	0	2005-03-24 13:40:51	
weather_xml	1	2005-01-11 15:42:24	

Buttons: Svuota, Cancella, Crea nuova tabella

Da questa schermata l'amministratore può creare una nuova tabella ed eliminare o modificare le tabelle esistenti. Cliccando sul nome della tabella si passa alla visualizzazione dati (default). Il menù a tendina in alto a destra consente di accedere alla struttura dati e agli indici, oltre che alle maschere di inserimento, importazione ed esportazione dati.



Per ogni campo viene visualizzato il tipo, la dimensione, se consente l'inserimento di NULL, il valore predefinito e l'appartenenza alla chiave primaria.

Relazioni

In DbTools viene definita "relazione" un collegamento logico tra i dati di due tabelle. Tale collegamento esiste solo a livello dati e codice PHP, nessuna implementazione di relazioni propriamente dette su database è presente nel modulo DbTools, sia per difficoltà di gestione sia per problemi di compatibilità tra i database (MySQL ad esempio nella versione 3 non supporta le relazioni).

Le relazioni vengono utilizzate in inserimento dati per visualizzare un campo descrittivo in un menù ed inserirne il relativo codice all'esecuzione della INSERT.

Esempio:

In fase di inserimento ordine verrà utilizzata in visualizzazione la ragione sociale del cliente (dalla tabella anagrafica) nella maschera presentata all'utente. Al submit della form in tabella ordini sarà inserito il codice cliente.

Queries

Questa sezione è preposta alla creazione di interrogazioni per la presentazione dati agli utenti pubblici o privati. Un wizard guida l'amministratore attraverso vari step, ma è comunque richiesta competenza nel linguaggio SQL.

Tipi di Query

Le query si differenziano in quattro tipi:

- report
- search
- mixed query
- child query

Query di tipo Report - Definita l'interrogazione al database questa query presenta direttamente i dati all'utente. La clausola WHERE può contenere valori fissi o fare riferimento a variabili di ambiente della piattaforma, quali login, id utente, gruppo, regione sociale, codice gestionale, ecc.

L'utilizzo delle variabili di ambiente ("Condizione run-time") permette di visualizzare dalla stessa query dati diversi per utente o gruppi: un agente avrà accesso solo alle anagrafiche clienti ad esso associate, ed un cliente avrà accesso solo agli ordini della propria azienda, utilizzando però sempre la medesima query.

Query di tipo Search - consiste in una query di tipo Report a cui è associata una maschera di ricerca (su campi definiti dall'amministratore) per consentire all'utente di imporre ulteriori filtri prima della visualizzazione dati.

Mixed Query - è un tipo di layout che visualizza nella stessa pagina due query differenti preesistenti. Viene utilizzata ad esempio per avere l'elenco ordini in corso clienti e fornitori divisi su due tabelle ma visualizzati contemporaneamente.

Child query - le "query figlio" sono query in cui la clausola where viene ereditata dalla query padre (di tipo Report o Search). Un link viene inserito su campo definito in fase di creazione che porta ad una seconda query utilizzando nella clausola where i campi della query padre più ulteriori campi specificati nella creazione. Le query figlio consentono di visualizzare una anagrafica clienti con link attivo sulla ragione sociale che conduce all'elenco ordini del cliente specificato nella riga selezionata. Dall'elenco ordini sarà poi possibile accedere agli articoli ordinati utilizzando una nuova query figlio. Non esistono limitazioni al numero di livelli utilizzabili.

Step di creazione query

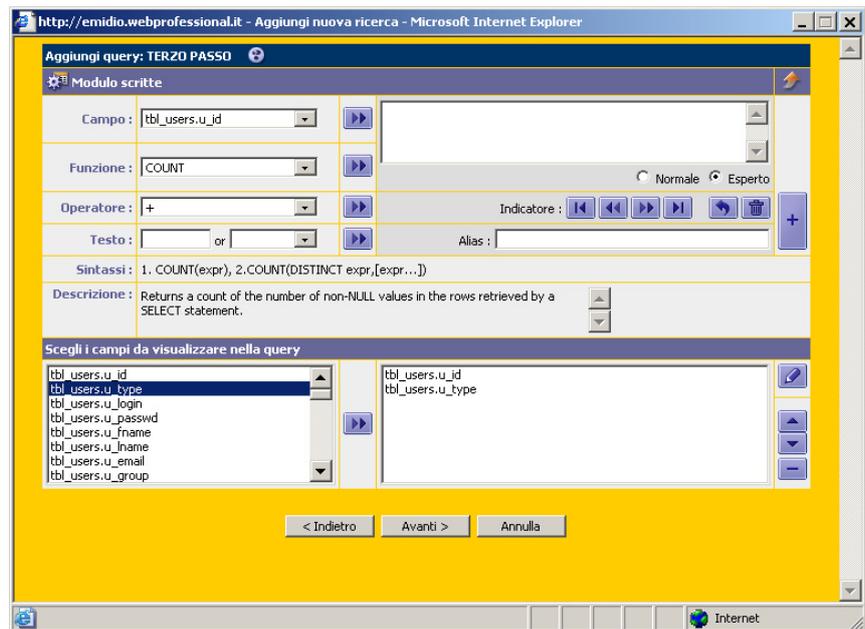
Le query vengono create attraverso una serie di passi che guidano l'utente nella composi-

zione dell'interrogazione. In dipendenza della singola interrogazione alcuni step possono non essere visualizzati se non necessari (Es. JOIN se una sola tabella è presente).

Step 1 - viene impostato il tipo di query e selezionate le tabelle che contengono i dati.

Step 2 - nel caso la query sia di tipo Search vengono definiti i campi che compariranno nella maschera di ricerca e la modalità di inserimento (testo libero, valori, valori relazionati).

Step 3 - scelta dei campi che compariranno nella visualizzazione del risultato. I campi possono essere semplici campi di database oppure il risultato di funzioni supportate dal database correntemente in uso inserite tramite il "Modulo scritte".



Step 4 - definizione dei JOIN fra le tabelle interessate, se più di una.

Step 5 - definizione della clausola WHERE su valori statici.

Step 6 - definizione della clausola GROUP BY se necessaria.

Step 7 - definizione della clausola ORDER BY se utilizzata. In questo step è inoltre possibile definire "campi aggiuntivi" che non verranno utilizzati nella query ma verranno comunque passati ad eventuali query figlio.

Step 9 - impostazione dei parametri generali della query (nome, possibilità di stampa, esportare in XML, TXT, XLS, ecc.) e della Condizione di Run-time (clausola WHERE utilizzando variabili dell'ambiente WebProfessional).

User Actions

Le "azioni utente" (o User Actions) permettono all'utente di manipolare i dati presenti nelle tabelle del database con le limitazioni imposte dall'amministratore. Una User Action può essere definita solo su singola tabella (non è possibile impostare una User Action per l'inserimento contemporaneo di dati su più tabelle da una unica schermata).

Nome tabella: nome della tabella oggetto della User Action

Campi: campi in visualizzazione, ricerca e nella form di modifica/inserimento.

Nome azione: nome della User Action (in lingua).

Descrizione azione: descrizione della User Action.

Sicurezza: definizione dell'accesso alla User Action per gruppi ed utenti.

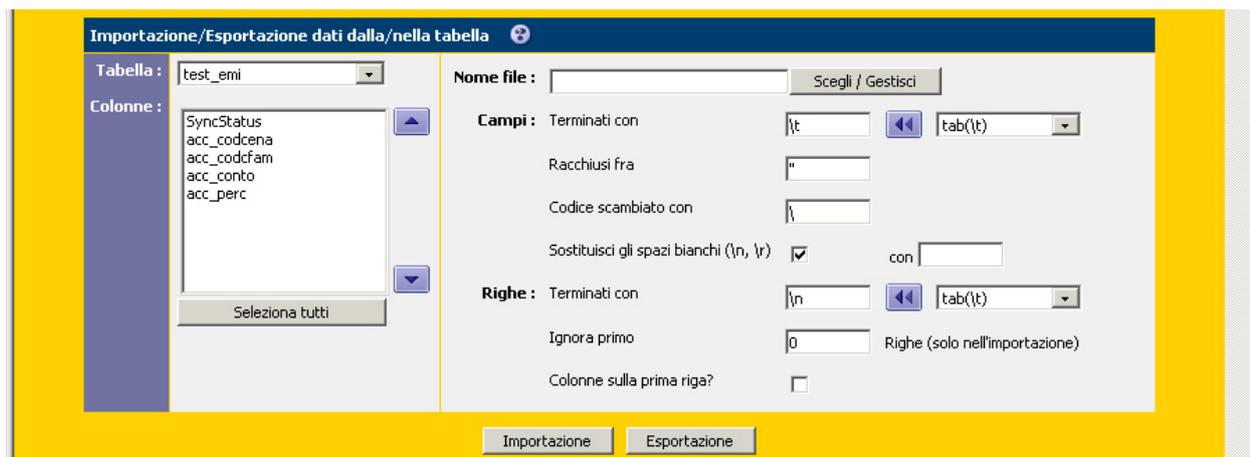
Autorizzazioni: definizione delle azioni possibili sui dati nella tabella (SELECT, INSERT, UPDATE, DELETE).

Filtro: clausola where su variabili d'ambiente della piattaforma. Consente di limitare la visibilità dei dati per utenti/gruppi diversi utilizzando la stessa User Action.

Invia e-mail: invio di un messaggio di sistema via email alla modifica dei dati.

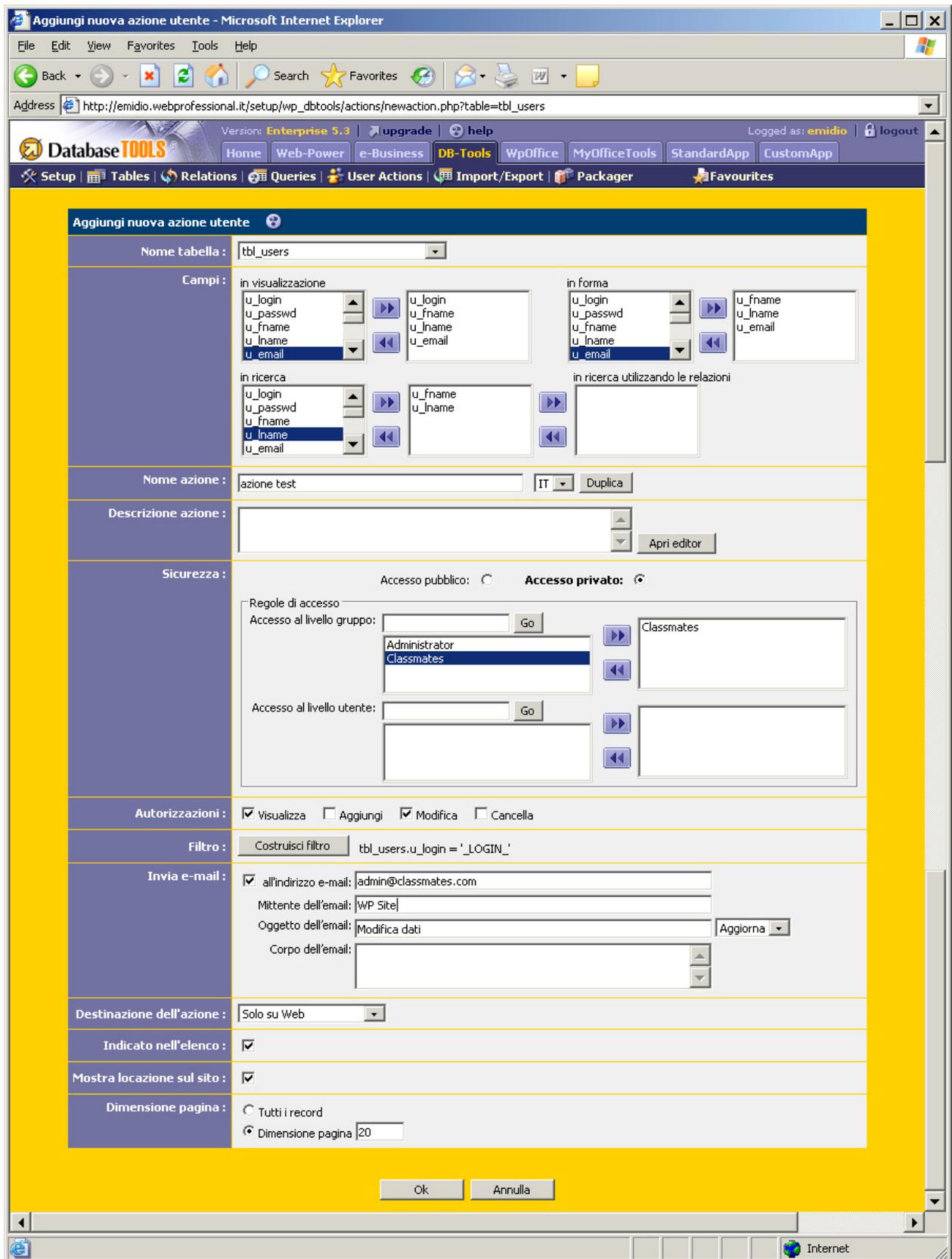
Import/Export

Sezione preposta all'importazione/esportazione di dati da e in file di testo, specificando i delimitatori.



Packager

Modulo per poter salvare una query o user action in un "pacchetto" per l'importazione in



Schermata per la configurazione di una User Action.

Cenni su DBTools e WP Office

un altro sito su piattaforma WebProfessional. E' prevista in DbTools la versione RunTime che consente l'utilizzo di Query e User Actions ma non la loro creazione o modifica. Lo sviluppo deve avvenire su piattaforma con modulo in versione completa per poi utilizzare lo strumento Packager per l'installazione delle interrogazioni sull'attivazione del cliente.



The screenshot shows a Microsoft Internet Explorer browser window displaying the website of the Comune di Nonantola. The browser's address bar shows the URL: `http://web.comune.nonantola.mo.it/default.php?t=dbtools&queryid=94&CND_modulo_urp$$Id_categoria_urp=2`. The website header features the logo of the Comune di Nonantola and the text "Comune di Nonantola". Below the header, there is a navigation menu with links: "Pagina iniziale del sito", "Database Tools", "Elenco query", and "Dettaglio_modulo_urp". The main content area is titled "Ufficio Relazioni con il Pubblico" and displays a table of 5 results. The table has three columns: "Modulo", "Note", and "file". The data rows are as follows:

Modulo	Note	file
Abbattimento alberi		Abbattimento alberi.pdf
Adesione raccolta differenziata		Raccolta differenziata.pdf
Iscrizione volontari Protezione Civile		Protezione civile.pdf
Pianta un albero		Pianta un albero.pdf
Potatura alberi		Potatura alberi.pdf

Below the table, there is a "Stampa" button and a "Login" section with fields for "Login:" and "Password:" and a "Login" button. The footer of the website provides contact information for the Comune di Nonantola: "via Marconi 11, Nonantola, 41015 MO, Italy", "Tel: +39 059 896511", "Fax: +39 059 896590", and "E-mail: info@comune.nonantola.mo.it". The website is built with Web-Professional.

Un esempio di presentazione dati basata su query DbTools con layout tabellare nel sito del Comune di Nonantola.

Il modulo WP Office

Se WebPower offre gli strumenti usuali richiesti nella creazione di un sito Web e nell'interazione con i clienti, WP-Office provvede gli utenti dello staff di un'azienda con gli strumenti necessari nelle loro attività di ogni giorno. Gli strumenti di WP Office sono rivolti al miglioramento dei rapporti aziendali:

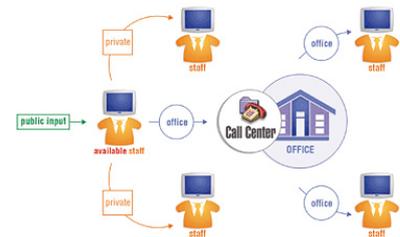
- relazioni pubbliche con i clienti dell'azienda
- relazioni aziendali intranet fra i membri dello staff

WP Office si divide le sue funzionalità in sottomoduli distinti e integrati che andremo ora ad analizzare.

Call Center

Il modulo Call Center gestisce una sorta di messaggistica per informazioni relative tipicamente a chiamate telefoniche per un utente non disponibile al momento.

Quando un utente riceve una chiamata da un cliente/fornitore o altro e lo vuole segnalare per un altro utente dello staff aziendale non disponibile inserirà una nuova voce appunto tramite questo modulo.



I campi da compilare sono:

Destinatario - Si sceglie il nome del collega a cui indirizzare il messaggio; è possibile digitare rapidamente una parte del nome poi cliccando su VAI nella menu a tendina a fianco appare l'elenco dei nomi corrispondenti da cui si sceglie quello giusto con un click. Automaticamente viene selezionata anche la sua email per inviare contemporaneamente un messaggio di posta elettronica.

Lo strumento WP CallCenter permette l'estensione dell'attività segretariale su Web.

Ragione sociale - Cliente/Persona di contatto: i nominativi provengono dal modulo contatti, e possono essere clienti acquisiti o potenziali. La logica è la stessa utilizzata anche per gli altri moduli. Se il cliente non è già codificato (es. cliente potenziale) è possibile digitarne il nome nel campo vuoto.

Oggetto e messaggio - Il testo da comunicare in breve al collega. Un volta inserito il messaggio, questo sarà visibile al destinatario dall'interfaccia Call Center, in una riga con le informazioni principali - cliccando sull'oggetti si apre una finestra con i dettagli delle informazioni.

Successivamente la voce di Call Center può essere inoltrata dal destinatario come Task, per generare una attività da svolgere, a se stesso oppure ad altri utenti. I Task fanno riferimento ad attività da svolgere, con possibilità di associare ad ogni Task dei rapporti di intervento gestendone nel contempo lo stato.

The screenshot shows a web browser window titled "http://www.remarksrLit - Aggiungi nuova voce di CallCenter". The page has a yellow header with the title "Aggiungi nuova voce di CallCenter". Below the header, there are two tabs: "CALL info" and "Informazioni aggiuntive". The form contains several fields: "Visibile via ACL per:" with a dropdown menu set to "solo destinatario specificato"; "Destinatario:" with a dropdown menu set to "nom" and a text input field containing "Nome Cognome"; a checked checkbox "Notifica il destinatario via email:" with a text input field containing "nome.cognome@remarksr"; "Voce:" with a dropdown menu set to "telefono"; "Da ragione sociale:" with a dropdown menu set to "arg" and a text input field containing "ARGOS SPA(c)(WP-OFFICE)"; "Da persona di contatto:" with a dropdown menu set to "VAI" and a text input field containing "Cristina Latini(WP-OFFICE)"; "Oggetto:" with a text input field containing "ha chiamato per spostare l'appuntamento"; and "Messaggio:" with a text area containing "richiamala per fissare una nuova data". At the bottom, there is a character count "(caratteri: 37)" and "Inserito da: Barbara Pisano". There are two buttons: "Conferma" and "Chiudi". The browser status bar shows "Operazione completata" and "Internet".

The screenshot shows the WP-Office Call Center interface in Internet Explorer. The browser window title is "http://www.remarksrLit - Intranet AREA - Microsoft Internet Explorer". The page has a blue header with the WP-Office logo and navigation tabs: "Web Site", "WP-Office", and "my Office Tools". Below the header, there is a "Call Center" section with a search bar and a "Ricerca:" field. The main content area is divided into two parts: a left sidebar with "Segnalazione" and "Azioni" sections, and a main table of calls. The table has columns for "Stato Voce", "Da", "Destinatario", "Oggetto & Messaggio", and "data di pubblicazione Modifica". The first row shows a call with state "telefono", from "ARGOS SPA(c)(WP-OFFICE) Cristina Latini(WP-OFFICE)", to "Nome Cognome", with the subject "ha chiamato per spostare l'appuntamento - richiamala per fissare una nuova data [più]", and published on "Lug 11, 2005 11:38". Below the table, there is a "Dettagli della voce di CallCenter" window. This window has a yellow header and contains the same information as the form in the previous screenshot, but with a "Stato" dropdown menu set to "nuovo". At the bottom of the details window, there are buttons: "Concludi", "inoltra come..." with a dropdown menu set to "task", "Modifica", "Stampa", "Cancella", and "Chiudi". The browser status bar shows "Operazione completata" and "Internet".

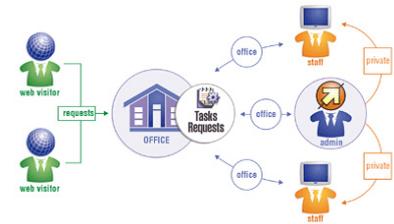


Nel pannello di sinistra visibile in qualsiasi modulo di WP Office l'icona lampeggiante del telefono indica che ci sono nuove voci del Call Center.

Task

Questo modulo permette di gestire l'inserimento, l'assegnazione ad utenti e il processo di svolgimento di una attività, di diversa natura e tipologia.

Elemento importante dei Task (come di altri moduli WP Office) è il collegamento con il cliente, che permette successivamente di raggruppare e analizzare attività fatte per uno specifico cliente.

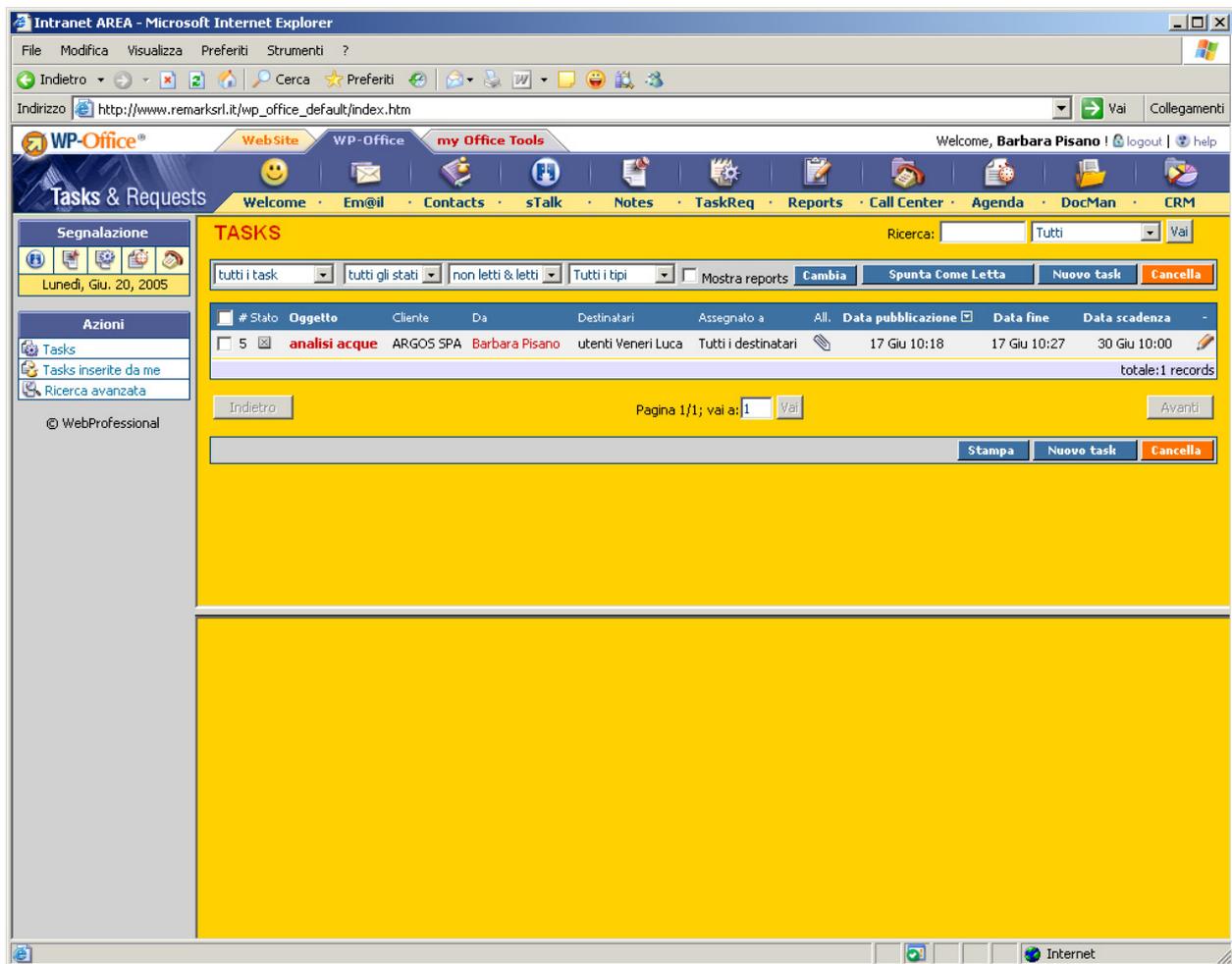


L'applicazione gestisce l'assegnazione, lo stato e l'avanzamento di richieste sia interne che esterne.

Parametri importanti in fase di inserimento sono:

- **Destinatari:** l'impostazione predefinita è "me stesso", da usare quando si inserisce un task solo per se stessi, come promemoria o attività che non implica il coinvolgimento di altri utenti. La selezione dei destinatari avviene tramite ricerca nel database utenti (utenti interni tipo "STAFF") per nome, cognome o login.
- **Ragione sociale cliente/Persona di contatto:** i nominativi provengono dal modulo contatti, e possono essere clienti acquisiti o potenziali
- **Oggetto e commenti:** testo esplicativo del task. Il testo del campo commento verrà aggiornato con nuove indicazioni/note nelle varie fasi di modifica ed avanzamento del task (Task aperto il, Task chiuso il - da -, ecc.).
- **Allegato:** permette di allegare file da condividere, se necessario.
- **Tipo:** assegnazione di un tipo al task. E' una suddivisione logica delle attività da fare.

Cenni su DBTools e WP Office



- **Attiva scadenza:** per gestire una data di scadenza, con visualizzazione all'interno dell'agenda dei task in scadenza

Nota: si consiglia di deselezionare ogni volta il flag "Ricorda se è ancora nuovo"

Stati avanzamento task

Appena inserito il task è nello stato di "nuovo", fintanto che il destinatario non lo processa. Il destinatario si fa carico del task cliccando sul pulsante "Inizia Lavoro" che cambia automaticamente lo stato del task in "in corso". Ogni variazione apportata allo stato del task viene registrata con data e ora ed id dell'operatore.

Al termine del processo il task viene "chiuso". L'operatore ha comunque la possibilità di riaprire il task in caso di necessità.

Riepilogo attività

Nella videata generale, di default si ha il riepilogo dei task. Vengono visualizzati tutti i task assegnati all'utente nello stato di "nuovo". Cambiando le opzioni nei menu in alto a sinistra (sotto la scritta TASK) si definiscono filtri per la visualizzazione dei task per assegnazione stato e tipo. Il menù laterale riporta alcune scelte rapide relative alla sezione Task (box Azioni).

Tabella riassuntiva delle possibili operazioni sull'oggetto Task.

Operazione	Chi può operare su "Categorie tasks" (1)	Chi può operare su "Tasks per utenti multipli"(2&3)	Note
Assegna utente	Utenti che hanno permesso ACL "Tasks Assign"		L'utente assegnato può essere cambiato solo se il task è ancora nello stato "Nuovo" e solo per la destinazione scelta
Prelievo task	Utenti che hanno permesso ACL "TasksReply" per la categoria del task		
Rispondi, Rispondi&Chiudi	L'utente assegnato ed anche la persona che ha inviato il task	Tutti gli utenti dall'elenco "Utenti multipli"	
Modifica	L'utente che ha inserito il task/ utenti che hanno permesso ACL "Modify" per la categoria del task	Amministratori/L'utente che ha inserito il task	Per quanto riguarda la destinazione, non si può cambiare il tipo di destinazione (per es. non si può cambiare una destinazione da "categoria" ad "utenti multipli"), ma solo la categoria/utenti di quella destinazione
Visualizza	Amministratori/L'utente che ha inserito il task/Utenti assegnati/ Utenti che hanno permesso ACL "View" per la categoria del task	Amministratori/L'utente che ha inserito il task/ Tutti gli utenti dall'elenco "Utenti multipli"	
Cancella	Utenti che hanno permesso ACL "Delete" per la categoria del task	Amministratori/L'utente che ha inserito il task	
Inizio lavoro	Solo l'utente assegnato	Tutti gli utenti dall'elenco "Utenti multipli"	
Inserimento task	Utenti che hanno permesso ACL "Insert" per la categoria del task	Tutti gli utenti Staff	
Chiudi task	Amministratori/L'utente che ha inserito il task	Amministratori/L'utente che ha inserito il task	

Reports

Il modulo Reports è destinato alla registrazione di tutte le attività aziendali sia interne che presso terzi. I reports (rapportini di intervento) possono essere inseriti come oggetti non correlati o come report associati ad un Task. Il task definisce l'attività da svolgere ed i report tengono traccia degli interventi effettuati per il completamento dell'attività.

Un report prevede all'inserimento informazioni sul cliente, un identificativo di documento (progressivo report),

oggetto, commenti ed informazioni relative al lavoro effettuato (ore, ore di viaggio, diritto di chiamata, ecc.). Nella sezione "Materiali" è possibile inserire l'elenco dei materiali utilizzati nel corso dell'intervento. Nel caso il report sia associato ad un task tutte le informazioni principali (cliente, oggetto, commenti, ecc.) vengono riportate dal task collegato. Attraverso questo modulo è possibile gestire le informazioni necessarie alla fatturazione di interventi tecnici ed attività esterne.



Lo strumento Reports consente l'inserimento di rapportini di attività svolte eventualmente associate ad un Task assegnato, aggiungendo informazioni per la fatturazione dell'intervento e dei materiali impiegati.

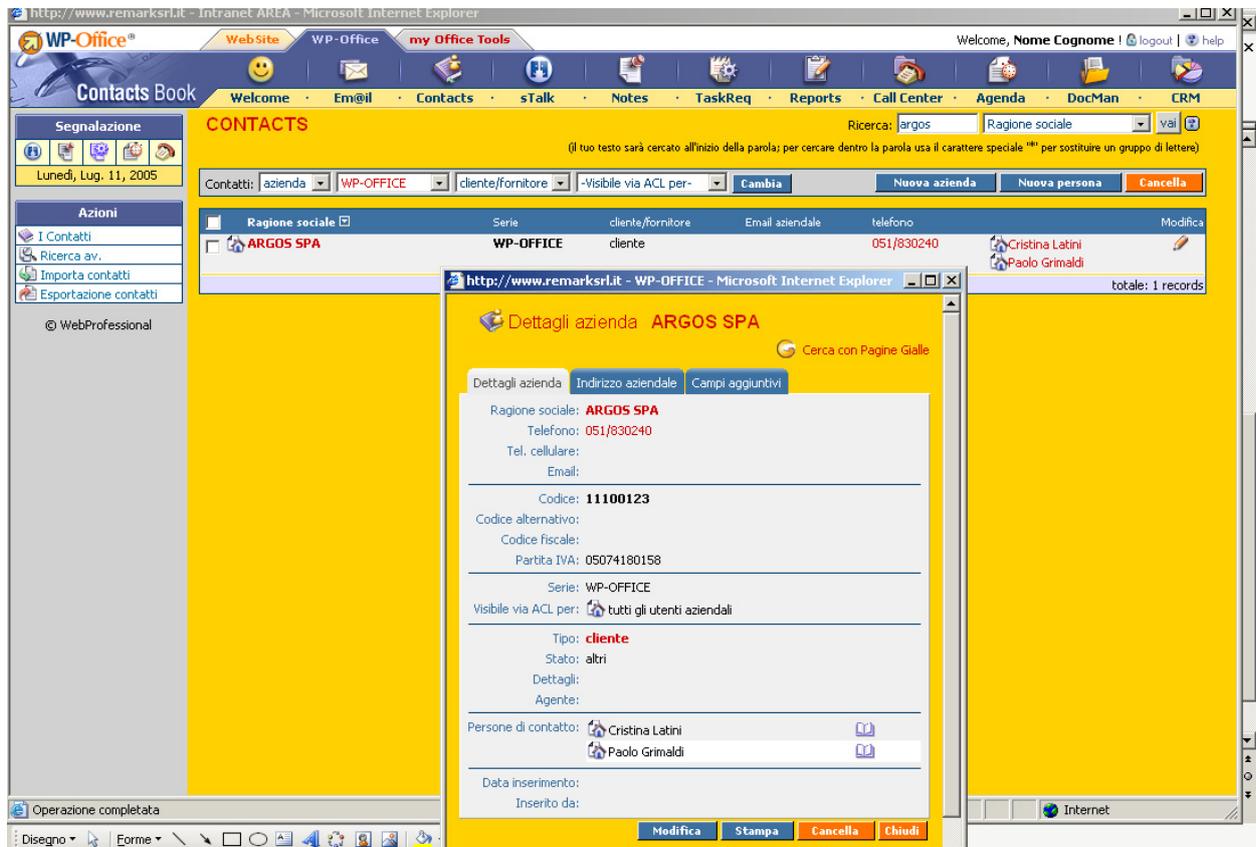
Contacts

Il modulo Contacts è lo strumento per ricercare e utilizzare i riferimenti dei "contatti" aziendali costituiti da clienti, clienti potenziali, fornitori e partners in genere, in sostanza tutte le anagrafiche dei soggetti necessarie allo svolgimento delle attività.

Lo strumento è configurato per gestire due tipologie di contatti: aziende e persone. Ogni azienda può avere associata una o più persone di riferimento, con dati specifici. Entrambi i tipi vengono memorizzati nella



Questa applicazione gestisce i contatti clienti/fornitori/clienti potenziali: contatti personali (con dettagli contatto, ecc.) e contatti aziendali (con una o più persone di contatto, profilo commerciale ed altre informazioni estese).



stessa tabella con un campo tipo discriminante. Vi è inoltre una suddivisione dei contatti per "serie" per consentire di definire categorie diverse di anagrafiche a cui assegnare visibilità diverse agli utenti (es. CLIENTI e FORNITORI). Dopo l'attivazione del modulo WP Office è presente unicamente la serie predefinita WP-OFFICE; altre serie possono essere definite dall'area di setup della piattaforma.

Tramite la funzione di ricerca in alto a destra si identifica in modo rapido l'anagrafica desiderata. I risultati trovati vengono elencati in forma sintetica con evidenziate anche le persone di riferimento; cliccando sul nome azienda si apre una finestra con i dettagli estesi del cliente. Di default la ricerca viene effettuata sul tipo anagrafico "azienda". Cambiando la selezione in alto a sinistra si passa dalla modalità "azienda" a quella "persona", nel momento in cui sia necessario effettuare ricerche sul tipo anagrafiche "persona".

Agenda

Il modulo Agenda permette organizzare, pianificare e gestire tutti gli eventi (riunioni aziendali, eventi personali, meeting, appuntamenti) e pianificare l'utilizzo delle risorse aziendali (auto, aule corsi, ecc.).

Ogni utente di tipo staff abilitato all'accesso a WpOffice ha a disposizione una agenda

personale. Di base la visualizzazione è impostata in modo settimanale, ma è possibile anche una vista giornaliera o mensile. In base ai permessi assegnati è possibile visualizzare le agende di altri utenti oppure gestirle.



Inserimento di un appuntamento

L'inserimento di un appuntamento avviene cliccando direttamente sull'ora/giorno desiderati. La finestra popup di inserimento riporterà il giorno e la data inizio attività. Nella schermata di inserimento

oltre all'oggetto ed al commento è possibile associare una anagrafica (da Contacts) e variare il destinatario (di default l'utente stesso).

oltre all'oggetto ed al commento è possibile associare una anagrafica (da Contacts) e variare il destinatario (di default l'utente stesso).

Le agende delle risorse aziendali eventualmente assegnate all'appuntamento verranno automaticamente impegnate - in caso di sovrapposizioni una notifica avvisa l'utente che non è possibile utilizzare la risorsa nel periodo indicato. Le agende delle risorse disponibili sono consultabile dal menù "Risorse per categoria", che visualizza in una unica schermata una collezione di agende appartenenti alla stessa categoria. In tal modo un utente può visualizzare la situazione del parco auto così come individuare una risorsa interna con basso carico di lavoro a cui assegnare un nuovo appuntamento.

DocMan

Il modulo DocMan è riservato alla gestione di file condivisi sul server. All'apertura di DocMan verranno visualizzati due tipi di cartelle:

- DocAziendali



Lo strumento permette una gestione online condivisa dei documenti all'interno di una azienda.

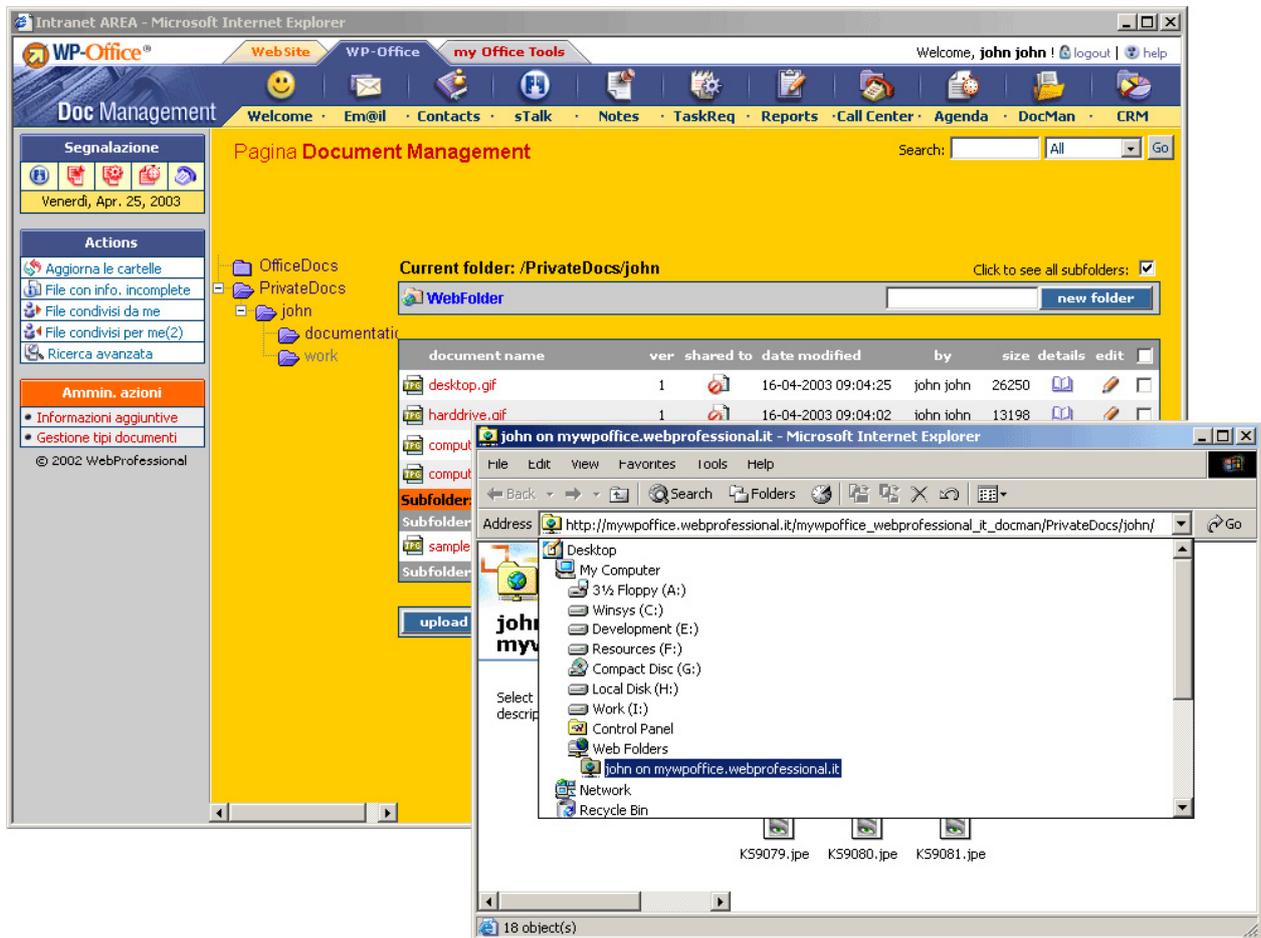
• DocPrivati

I **DocAziendali** sono visibili ad ogni membro dell'applicazione WP-Office. Tutti gli utenti sono abilitati alla loro visualizzazione, creazione ed elaborazione.

I **DocPrivati** sono i file personali dell'utente attualmente collegato. Ai file sono associabili informazioni aggiuntive come una anagrafica cliente, una persona di riferimento, note, ecc.

L'accesso ai documenti avviene offline tramite link su protocollo HTTP o HTTPS (il che prevede che si scarichi il file per poterlo modificare e che poi venga di nuovo caricato su server nel caso si vogliano conservare le modifiche) oppure online dall'icona WebFolder su protocollo Web DAV (Web based Distributed Authoring and Versioning protocol).

Se il proprio browser è abilitato DAV (es. Internet Explorer 6.0), la cartella corrente verrà aperta come nella gestione risorse di Windows - la gestione risorse di Windows visualizza WebFolder come una risorsa di rete. Si può elaborare direttamente un file Word esistente

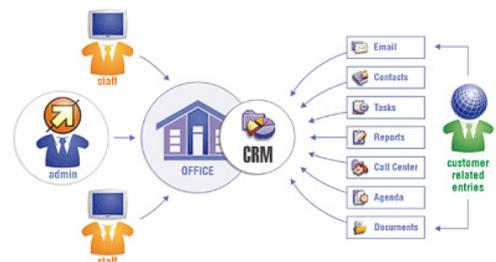


sul server tramite un utilizzo trasparente del protocollo DAV - il file verrà posto in stato di accesso in scrittura esclusivo lasciando agli altri utenti l'accesso in sola lettura.

Nome	Soprannome	Codice	Codice alternato	Seria	Tipo	Email
Daniela	Cisi			import	person	daniela@cpssoftware.it

CRM

Il modulo CRM è quello che fornisce informazioni sintetiche relative ad un cliente. Una volta selezionato un cliente (tramite la funzione di ricerca o la ricerca avanzata) viene impostato un filtro su tutti i moduli relativo all'anagrafica cliente selezionata. Questo significa che in Callcenter, Task, Reports saranno visibili solo le informazioni relative al cliente selezionato in CRM, fino a quando la visualizzazione normale non verrà ripristinata tramite il tasto "Ripristina". CRM diventa fulcro e collante del modulo rendendo WP Office un potente strumento di Customer Care.



Lo strumento CRM filtra tutti i record aziendali presenti in WP Office relativi ad un cliente specifico e offre le informazioni in visualizzazioni semplici ed immediate.

Conclusioni sui moduli DbTools e WP Office

I due moduli trattati in questo capitolo vanno a completare WebPower ed eBusiness coprendo la quasi totalità delle informazioni da gestite nella fornitura di servizi avanzati su web, offrendosi come soluzione unica ed integrata alle diverse esigenze delle aziende.

Case Studies

In questo capitolo verranno brevemente illustrate alcune applicazioni reali della piattaforma WebProfessional relativamente ai quattro moduli WebPower, eBusiness, DBTools e WP Office.

Corghi S.p.A. - WebPower

Applicazione del modulo WebPower per il content management della pagine del sito relativamente a presentazione aziendale, presentazione prodotti, informazioni agli utenti.

Corghi S.p.A.

La storia

E' una storia semplice e per certi versi esemplare quella che racconta le radici della Corghi. Una storia che comincia nel cuore dell'Emilia, terra aperta e vitale che mostra sin dal primo dopoguerra una vocazione mai abbandonata a sviluppare un'impreditoria all'avanguardia, soprattutto nel campo della meccanica.

Nel 1947 i fratelli Erminio e Remo Corghi cominciano a mettere a frutto le loro capacità imprenditoriali e il loro ingegno costruendo apparecchi radio e inventando un particolare richiamo da caccia con registratore a filo. Sei anni dopo, nel 1954, nasce la Corghi elettromeccanica. Il marchio con l'usignolo, che da questo momento sarà il simbolo dell'azienda, comincia a farsi stimare.

L'impresa sviluppa il suo saper fare meccanico su vasta area d'intervento: stabilizzatori di tensione per la tv, macchine agricole, elettrodomestici, attrezzature meccano-tessili.

Sono anni in cui le idee fioriscono e si sviluppano senza posa in attesa dell'"Occasione". Nel 1960 scoppia il boom dell'automobile e i Corghi si fanno trovare pronti. Inventano e brevettano il primo smontagomme del mondo. E' "l'artiglio" giusto con cui aggredire il mercato. L'azienda prende quota e non smette più di volare alto.

Oggi Corghi è prima in assoluto sul mercato italiano, e leader mondiale, nel settore degli

equipaggiamenti per l'assistenza ai veicoli.

L'azienda

Da oltre 50 anni Corghi è il più grande produttore mondiale di attrezzature per il servizio alla ruota.

L'attuale programma comprende smontagomme automatici, equilibratrici, allineatori, sollevatori, linee di controllo tecnico, strumenti per la diagnostica e attrezzature complementari.

Insieme ai suoi partner, Corghi costituisce la più vasta e capillare rete per la distribuzione di auto attrezzature del mondo, coprendo 137 paesi e con filiali dirette in Germania,



Messico e Cina.

A disposizione dei più importanti partner commerciali, la possibilità di accedere ad un'area protetta del sito internet per gli ordini on-line di attrezzature e ricambi, la gestione delle garanzie e il download di documentazione tecnica e commerciale.

Corghi è la prima azienda del settore ad aver ottenuto, nel 1994, la certificazione ISO9001, ed è omologata in tutto il mondo con la gamma completa dei prodotti per il servizio alla ruota dai più grandi gruppi automobilistici (DaimlerChrysler, Opel/GM, Ford, Peugeot, Citroen, gruppo Fiat, Renault, Toyota e gruppo VW).

Con la linea "Ferrari Workshop Equipment", Corghi è fiera soprattutto di essere l'unico fornitore di attrezzature per auto officine Ferrari - inclusa la "Scuderia" di Formula 1 - con l'intero programma di produzione.

Numerose le collaborazioni, già in fase di progettazione dei nuovi pneumatici, con i più importanti costruttori: Bridgestone, GoodYear, Michelin, Pirelli.

Leader nella qualità, leader nell'innovazione

La strategia Corghi si basa sulla qualità dei prodotti e sull'innovazione tecnologica, come dimostra il recente successo dello smontagomme Artiglio Master, il primo al mondo realmente superautomatico che proietta il gommista nel futuro della sua professione. Recentemente Corghi è stata più volte premiata per l'innovazione tecnologica nel settore, da diversi enti qualificati e indipendenti che certificano in modo autorevole lo sforzo e i risultati raggiunti grazie alla costante ricerca di soluzioni innovative.

Soluzioni che consentono al gommista e al meccanico di offrire una nuova immagine di professionalità a fronte delle moderne tecnologie applicate alle vetture e ai pneumatici.

La gestione dei contenuti del sito

L'obiettivo era di rendere l'azienda autonoma nella gestione ordinaria dei contenuti relativamente a presentazioni prodotti, news di vario genere, riferimenti interni e presentazione aziendale.

Il layout principale e l'impostazione grafica generale vennero realizzati da CP Software, così come il porting dei contenuti del precedente sito alla piattaforma WebProfessional. Include nel pacchetto alcune giornate di formazione per l'utilizzo dell'area di setup della piattaforma.

Il sito si compone di due macro sezioni realizzate con due differenti attivazioni di WebPower: Corghi Automotive e Corghi Tessile. L'ingresso è unico utilizzando una pagina introduttiva in HTML con i link alle due sezioni (che puntano a due nomi host distinti).

Le due macro sezioni in virtù della separazione su due attivazioni possono riportare una

grafica simile ma al contempo diversificata su colori e logo.



Automotive (Home)

- Stampa
- Formazione
 - Calendario corsi
 - Statistiche
 - Quaderni scuola
- Gadgets
 - Adesivo
 - Bandiera
 - Camice
 - Cappellino
 - Cartellina
 - Catalogo
 - CD
 - Penna
 - Polo Rossa
 - Spilla
 - Sportina di plastica
 - T-Shirt
 - Tuta da lavoro

Profilo Aziendale

- Dove siamo
- Storia
- Azienda
 - Un progetto comune

Meccanotessile (Home)

- Formazione
- Storia
- Yarn Count Converter

Profilo Aziendale

- Dove Siamo
- Storia
- Azienda
 - Un progetto comune
 - Obiettivo primario
 - Progettare tecnologia
 - Ingegno e flessibilita'
 - Superiorita' nelle relazioni

- Numeri
- Corghi nel mondo

Linee prodotti

- Aspatrici
- Dipatrici
- Roccatrici
 - Roccatrici a Guidafilo
 - Roccatrici a Rullo
- Roccatrici per maglieria
- Binatrici
- Accessori
- Intelliwind

-  Obiettivo primario
-  Progettare tecnologia
-  Ingegno e flessibilita'
-  Superiorita' nelle relazioni

Numeri

-  Fatturato
-  Produttività
-  Brevetti
-  Ricerca
-  Mercato
-  Dipendenti
-  Formazione

Corghi nel mondo

Linee prodotti

Smontagomme

-  Artiglio Master
-  A2030
-  A2025
-  A2020
-  A2015
-  A2005
-  A2000
-  JOLLY
-  SP2000
-  Premicerchio
-  PU1500
-  WL60
-  INFLATRON
-  Autocentrante
-  Stallonatore orientabile
-  HD1400 MasterOne
-  HD1400
-  HD1400E
-  HD1000
-  AG DoubleT.T. L
-  HD600
-  DV120

Rete commerciale

Eventi

Links

Contatti

Tutela della privacy



-  Anelli bloc. cerchi lega
-  Radio comando

Equilibratrici

-  EM8370
-  EM8340
-  EM7370
-  EM7340
-  EM7070
-  EM7040
-  EM6040
-  EM43
-  ET88
-  ET77
-  ET66
-  EF14
-  EF13
-  HPU - P21 - L32
-  RTD

Allineatori

-  EXACT 6800
-  EXACT 68
-  EXACT 58
-  EXACT 28
-  EXACT W.A. software
-  Rilevatori CCD
-  EXACT 36
-  EXACT LASER
-  RH METER
-  Allineamento Camion

-  Barra Multilink
-  Kit Pese
-  Piattaforme rotanti
-  Calibratore
-  Sollevatori
 -  ERCO 4400-4401
 -  ERCO 3600-3601
 -  ERCO 351
 -  ERCO 301
 -  ERCO 3012
 -  ERCO 3222EL - 4022
 -  ERCO 321
 -  ERCO MINI 2500 FI
 -  ERCO 25S
 -  ERCO 25TP
 -  CA002
 -  CA118
 -  CA240
 -  CA240P
 -  CA330
 -  CA440
 -  BL600
-  Diagnostica
 -  BT100E/3
 -  BT90
 -  GA570
 -  HT910
 -  HT970
 -  ST200E
 -  T-BT111
-  Bike Line
 -  BC200
 -  BB400
 -  BL600
 -  BT800
-  Miscellaneous
 -  Ecobox
-  Lindo 700
-  C&S
-  IC90 - Inflabox
-  VP1 - VP2 - VP3
-  VP4 - VP5 - VP6
-  T16 - TS16
-  TS18 - TS24
-  SP17 - SP25
-  TG940 - TT970
-  **Rete commerciale**
-  **Galleria**
 -  Auto Moto Show
 -  Visita Hankook
 -  visita ad ACC
 -  Motor Show Mosca
 -  visita Rodatec
 -  Superbike Misano
 -  Autoservice Tokyo
 -  autopromotec 2005
 -  cena Royal Carlton
 -  premiazione GIPA
 -  Auto&Manufacture Shanghai
 -  Fiera dell'Automobile 2005
 -  Fiera dell'Automobile 2004
-  **Saremo presenti a... [eNews]**
-  **Links**
 -  Enti fiera
 -  Case automobilistiche
 -  Costruttori di pneumatici
 -  Costruttori di cerchi
 -  Riviste specializzate
-  **Stampa [eNews]**
-  **Contatti (Form)**
-  **Tutela della Privacy**
-  **Job opportunities**
 -  Lavora con noi
 -  Curriculum

La parte di manutenzione dei contenuti riguarda piccoli aggiustamenti alle pagine dei prodotti (con l'aggiunta anche di nuove pagine) e un costante aggiornamento della sezione News, suddivisa nelle categorie "Stampa" e "Saremo presenti a..." ed Upload/Download che prevede cataloghi suddivisi per linee di prodotto.

La struttura dei due siti si sviluppa in maniera simile anche se la sezione Automotive è sicuramente più articolata. A seguire il parallelo dei menù delle due sezioni:

I due siti sono gestiti dai due settori in modo autonomo per i contenuti ma comunque coordinati per quanto riguarda la grafica e la direzione generale.

Alcuni voci conducono a sottosezioni specifiche di alcuni moduli, resi accessibili da menù per migliorare la navigabilità del sito (si vedano nella struttura del sito le voci seguite da [*nome*] ed il nome del modulo).

LuEn S.r.l. - WebPower ed eBusiness

Applicazione dei moduli WebPower ed eBusiness sulle esigenze prospettate del cliente.

LuEn S.r.l.

LuEn, azienda produttrice di valvole oleodinamiche, è presente sul mercato dal 1979. La produzione comprende sia articoli per il settore mobile che per quello industriale, con la produzione di blocchi integrati compatti.

La presentazione dell'azienda e la produzione (WebPower)

La presentazione dell'azienda è fortemente caratterizzata da una impostazione familiare. La struttura del sito si articola in una serie di menù per la presentazione dell'azienda e della produzione.

-  Home
-  L'Azienda
-  Tour
-  Controllo qualità
-  Guida alla produzione
 -  Cartucce cavità 2 vie

Case Studies

- Cartucce cavità 3 vie
- Cartucce cavità 4 vie
- Corpi valvole
- Produzione di Cartucce
- Settore di applicazioni
 - Mobile e Industriale
- Catalogo
- Contatti

- Ricerca
- E-news
- Mappa del sito
- Mailing List
- Links

The screenshot shows a web browser window displaying the Luen Srl website. The browser's title bar reads "Luen Srl - Produzione valvole oleodinamiche - Microsoft Internet Explorer". The address bar contains the URL "http://localhost:81/default.php?t=site&pgid=1". The website layout features a navigation menu on the left with links such as "Home page", "L'Azienda", "Tour", "Controllo qualità", "Guida alla produzione", "Settore di applicazioni", "Catalogo", and "Contatti". The main content area is divided into three sections: "Passato esperienza" (Past experience) with an image of a valve assembly, "Presente dinamismo" (Present dynamism) with an image of a valve block, and "Futuro innovazione" (Future innovation) with an image of a valve assembly. The right sidebar contains several buttons: "Ricerca", "E-news (2)", "Mappa del sito", "Mail-lists", "Download", and "Links". The status bar at the bottom indicates "Local intranet".

LuEn s.r.l. - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.luen.it/default.php?e=comm&vimgd=&el1=101&el2=10101&page=1&LU_currentpage=2

LuEn
Hydraulic VALVES and Integrated Components

Home | Cerca Prodotti

a cartuccia (Valvole Overcentre) > owc/40-...

owc/40-...

Visualizza prodotti 1 a 10 da 15 record(s) trovati

Codice prodotto	Nome prodotto
001.620.0H0	OWC-30-H Reg. piombata plastica
001.635.0H0	OWC-30-H Reg. piombata plastica
001.620.0X0	OWC-30-X Reg. grano (vecchio codice: 001.620.000)
001.635.0X0	OWC-30-X Reg. grano (vecchio codice: 001.635.000)
001.001.0H0	OWC-40-H Reg. piombata plastica (vecchio codice: 001.470.000)
001.046.0H0	OWC-40-H Reg. piombata plastica (vecchio codice: 001.471.000)
001.091.000 (FP)	OWC-40-H Reg. piombata plastica (vecchio codice: 001.091.000)
001.001.0K0	OWC-40-K Reg. piombata
001.046.0K0	OWC-40-K Reg. piombata
001.001.0X0	OWC-40-X Reg. grano (vecchio codice: 001.001.000)

Visualizza prodotti 1 a 10 da 15 record(s) trovati

pagina 1 da 2 [SUCC >](#)

>> [stampa](#) questa pagina di catalogo

1 2 3 4

SOLUZIONI DI PRODUZIONE NECESSARIE PER LA SCELTA DEL PRODOTTO E RELATIVA DEFINIZIONE DEL CODICE D'ORDINAZIONE

LuEn HYDRAULIC VALVES AND INTEGRATED COMPONENTS S.R.L. ITALY

OWC-40-...

struttura flange
Flange setting (Z)

Piombata Saletto (H)

Piombata Saletto (K)

Flusso libero Free flow

Flusso controllato Controlled flow

Microsoft Internet Explorer

www.luen.it/default.php?e=comm&el1=103&el2=10301

LuEn
Hydraulic VALVES and Integrated Components

Home | Cerca Prodotti

10 - Set - 2005

Ricerca

E-news (2)

2. Chiusura Estiva:

Mappa del sito

Mail-lists

Links

CP Software Distributore Centro-Sud Altvista

Accessi: 7431

Visualizza prodotti 1 a 10 da 15 record(s) trovati

pagina 1 da 2 [SUCC >](#)

>> [stampa](#) questa pagina di catalogo

Accessi: 7431

Visualizza prodotti 1 a 10 da 158 record(s) trovati

pagina 1 da 16 [SUCC >](#)

>> [stampa](#) questa pagina di catalogo

Il catalogo online (eBusiness)

Il catalogo online si basa su categorie e sottocategorie derivate direttamente dal catalogo cartaceo in essere. Il numero di livelli e di righe per livello risultava troppo elevato per una visualizzazione completa, tanto da richiedere di inserire le sezioni principali come "tipi categoria".

La pagina di visualizzazione lista è stata personalizzata per fornire informazioni aggiuntive in modo compatto ed i disegni in formato DWF, stampabili in vettoriale in alta qualità. La pagina specifica per prodotto non viene visualizzata.

Da ogni sottocategoria si raggiunge l'elenco delle valvole con le dimensioni delle porte, se presenti. Le informazioni sono sincronizzate con la tabella preposta per la gestione dei campi aggiuntivi. Nella parte inferiore trovano spazio le pagine di catalogo in formato DWF relative all'elenco delle valvole visualizzate nella categoria.

Il file DWF viene visualizzato attraverso il plugin Zing! DWF Viewer che consente operazioni di zoom, pan e stampa oltre che alla gestione (in visualizzazione) dei layer presenti.

Ogni sottocategoria consiste sostanzialmente in un gruppo di valvole il cui elenco viene separato per pagine se in numero superiore a dieci. Al gruppo di valvole sono associate una o più pagine di catalogo (la stessa pagina può essere relativa a valvole diverse, e quindi comparire in gruppi distinti) visualizzate una alla volta e sfogliabili tramite una piccola form preposta al caricamento in Zing! DWF Viewer dei diversi file DWF associati al gruppo.

Smeg S.p.A. - eBusiness

Applicazione del modulo eBusiness personalizzato sulle esigenze prospettate del cliente per la gestione degli ordini ricambi tra la Casa Madre e le filiali in Europa.

Smeg S.p.A.

Smeg (Smalterie Metallurgiche Emiliane Guastalla) è ad oggi uno dei principali produttori di elettrodomestici per cucine italiane. La gamma di prodotti spazia dai piani cottura al lavaggio (lavastoviglie, ecc.) raggiungendo settori specifici quali l'elettromedicale e prodotti per la ristorazione e le comunità.

Fondata da Vittorio Bertazzoni in 1948 come "Smalterie Metallurgiche Emiliane Guastalla" (Guastalla, Reggio Emilia). L'azienda è tutt'ora diretta dalla famiglia Bertazzoni, e il nuovo stabilimento principale sorge a San Girolamo, a brevissima distanza dalla sede storica di Guastalla.

Il marchio Smeg, attraverso "Tecnologia che arreda", riflette quella che da sempre è la filosofia aziendale e rappresenta l'espressione perfetta di un preciso modo di operare. In ogni fase di fabbricazione vengono applicati alta tecnologia e controlli costanti.

Dal 1988 Smeg è presente in Europa con sette filiali. Fortemente legata ad ognuna di esse, la Casa Madre trasmette giorno per giorno a ciascuna le idee, la storia e la cultura racchiuse nel suo marchio. Ogni filiale, pur autonoma, opera così in perfetta armonia con la filosofia che ha reso unico lo stile Smeg.

Esploweb

L'applicazione di gestione esplosi è preposta all'ordine ricambi e riferimento tecnico per i Centri Assistenza Tecnica (CAT) e le filiali europee. Integrata nel modulo di WebProfessional eBusiness, i modelli vengono gestiti come articoli e le parti come ricambi. L'esploso compare nella pagina principale delle informazioni articolo (modello) in formato pdf, con reference number attivi per consentire di selezionare direttamente dall'esploso il ricambio (è comunque presente nella parte inferiore una distinta ricambi con gestione delle varianti introdotte sul modello).

Specifiche iniziali

Specifiche di progetto iniziali sottoposte da Smeg per la realizzazione della personalizzazione del modulo eBusiness "EsploWeb".

1. Generalità: Le bandierine in alto a destra permettono di selezionare la lingua con la quale vengono visualizzate le schermate di WebProfessional. Sarà possibile utilizzare un massimo di sei lingue diverse (come da specifiche piattaforma WebProfessional). L'applicazione viene visualizzata all'interno di WebProfessional in una unica finestra del browser con bordi ridimensionabili.

Il modello richiesto viene selezionato utilizzando la struttura ad albero riportata nei menù sulla sinistra (che riporteranno categorie e sottocategorie) oppure utilizzando la funzione di ricerca prodotti di WebProfessional.

Nella visualizzazione ad albero selezionando una famiglia si espande l'elenco dei modelli relativi. Dal modello si accederà poi all'esploso (nella parte principale della finestra). L'esploso consiste in un file pdf con aree attive (reference number) che consentiranno di selezionare la parte direttamente dall'esploso (conservando l'elenco di tutte le parti nella parte inferiore della finestra).

2. Esploso: l'esploso è dinamico e permette, facendo click sul reference number, la visualizzazione dei parametri in una area riservata della pagina (Codice, Qta in distinta, Descrizione in lingua, Prezzo di listino) al di sotto dell'immagine dell'esploso e, con un secondo click sulla riga corrispondente (se la parte ha più codici associati saranno visualizzate più righe), di inserire il pezzo nel carrello eBusiness con quantità 1. L'elenco completo delle parti è sempre visualizzato in basso (standard eBusiness).

3. Distinta/Part list: La lista dei componenti mostra i dati riportati nel database associati al modello selezionato. I dati da visualizzare sono: reference number, codice, descrizione, data di ingresso componente, data di uscita componente. L'intestazione e le descrizioni dei componenti sono nella lingua impostata, utilizzando le descrizioni messe a disposizione dal database Smeg. E' da prevedere l'ordinamento della lista per ciascuno dei campi visualizzati con modalità simile a "Esplora Risorse" di Windows (personalizzazione del template di visualizzazione).

4. Visualizzazioni/Stampe (bottoni in alto): Le diciture standard di eBusiness vengono sostituite con opportuni elementi grafici (definibili anche in seguito). Le stampe richieste, vengono formattate in una pagina standard che contiene una intestazione e piè di pagina fissi (logo Smeg, data stampa, documento, modello) e sono le seguenti:

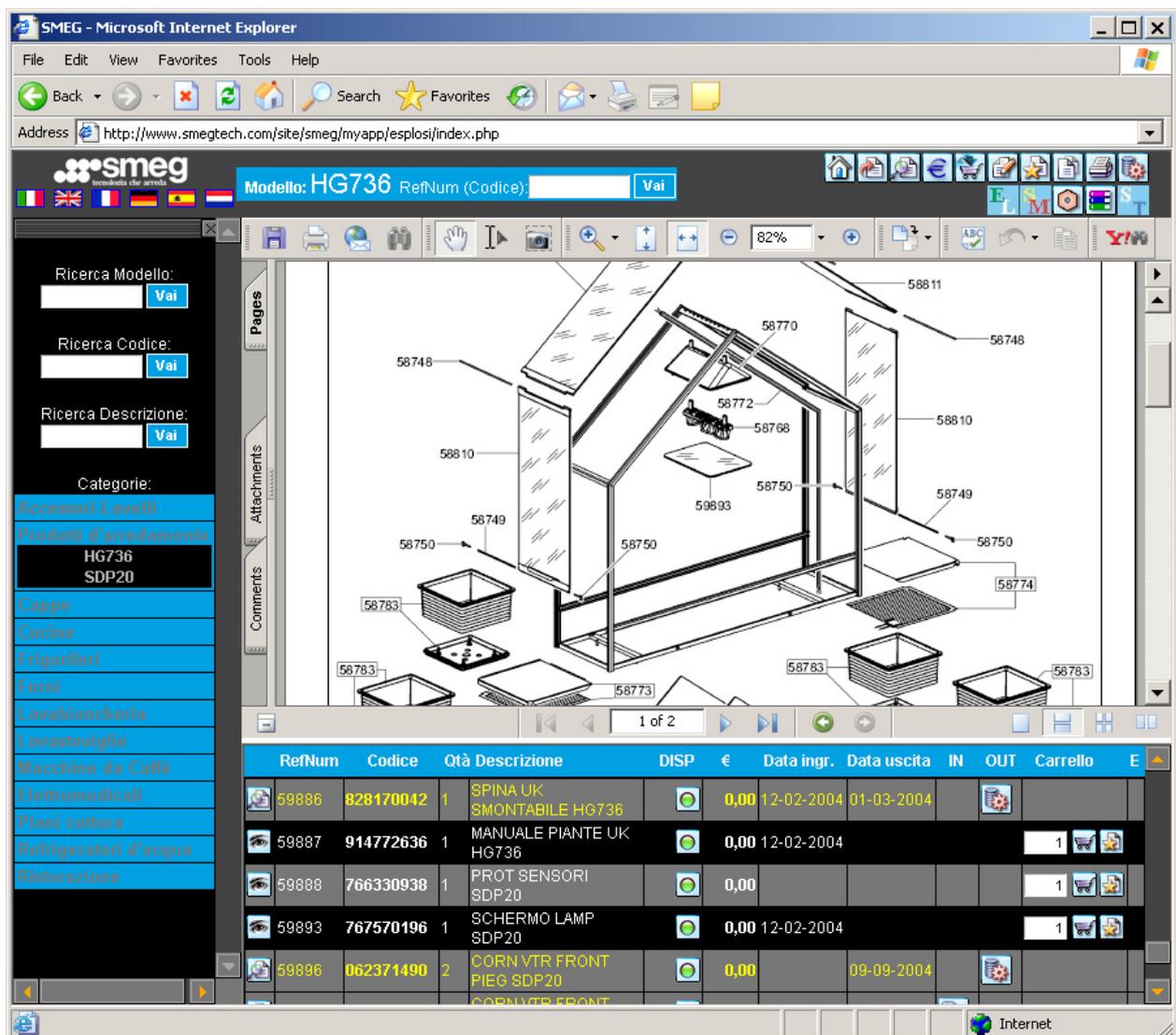
- **Stampa distinta:** deve essere stampata la lista componenti che compare a piè di pagina nella lingua selezionata. Il formato grafico del report verrà definito in seguito, ma riporta i dati visualizzati nella finestra. Necessario verificare se è possibile inserire il taglio pagina, solo in Internet Explorer. Cliccando si apre una popup con la distinta in formato di stampa, viene richiamata automaticamente la funzione di stampa (solo IE) (e' poi richiesto l'intervento dell'utente per lanciare effettivamente la stampa) e dopo la stampa la finestra si chiude automaticamente. La prima pagina e' una copertina, le successive contengono i dati. Le intestazioni di colonna devono essere riportate in ogni nuova pagina (solo IE).
- **Documenti:** visualizzazione (come già avviene in SmegTech) dei documenti collegati

a tale modello (compresi libretti istruzioni)

- **Scheda tecnica:** visualizzazione su una finestra separata di tabella che contiene dati provenienti da un database così strutturato:

Tabella Campi: Id_campo, desc_campo_IT, desc_campo_GB, desc_campo_FR, desc_campo_ES, desc_campo_DE, desc_campo_NL

Tabella Caratteristiche: Id_Modello, Id_campo, Valore_IT, Valore_GB, Valore_FR, Valore_ES, Valore_DE, Valore_NL, (viene visualizzato il campo nella lingua selezionata)



RefNum	Codice	Qtà	Descrizione	DISP	€	Data ingr.	Data uscita	IN	OUT	Carrello	E
59886	828170042	1	SPINA UK SMONTABILE HG736		0,00	12-02-2004	01-03-2004				
59887	914772636	1	MANUALE PIANTE UK HG736		0,00	12-02-2004					
59888	766330938	1	PROT SENSORI SDP20		0,00						
59893	767570196	1	SCHERMO LAMP SDP20		0,00	12-02-2004					
59896	062371490	2	CORN VTR FRONT PIEG SDP20		0,00		09-09-2004				
			CORN VTR FRONT								

La tabella sulla form web visualizza: Desc_campo_LINGUA, Valore_campo_LINGUA
Visualizzazione testi in lingua selezionata. Tale finestra deve essere poi stampabile come appare a video (tramite funzionalità standard del browser web)

- **Installazione:** si tratta di un documento pdf con le istruzioni per l'installazione (servizio già standard in web-power)
- **Kit accessori:** è da prevedere opportune tabelle di database da definire con Smeg in cui inserire, per ciascun modello, uno o più kit accessori per la manutenzione. Facendo click si apre l'elenco dei kit disponibili, cliccando su ogni kit si espande la riga in sottorighe contenenti ognuna un componente e la quantità relativa. Facendo click sul kit completo tutto i codici vengono inseriti nel carrello con la quantità specificata. Dopo l'inserimento dell'ordine viene persa l'informazione di appartenenza al kit.

5. Ordini ricambi: lo standard eBusiness viene accettato nel suo complesso con le seguenti aggiunte/correzioni:

- L'aspetto grafico deve essere lo stesso di SmegTech (colori, caratteri, ..)
- Il carrello è attualmente una "entità" temporanea che viene chiuso alla chiusura del browser. E' necessario gestire l'ordine con 4 stati:

A è equivalente al carrello non ancora inviato di e-business (è possibile per l'utente "congelare" un ordine e riprenderlo e terminarlo tempo dopo).

I è equivalente al carrello già inviato

P parzialmente evaso. A seguito della consegna del materiale ai clienti Cpssoftware riceve i dati relativi ai codici evasi. Tali dati contengono anche il numero ordine del cliente. Mettendo in collegamento questi dati con quelli relativi agli ordini inseriti su database è possibile verificare se un ordine è stato parzialmente evaso. In tal caso lo stato diventa automaticamente "P"

E Evaso. L'ordine è stato completamente evaso.

Il cliente che invia un ordine deve ricevere una conferma d'ordine dal sistema SmegTech (contenente numero d'ordine, ragione sociale e codice cliente, data ordine, elenco parti con prezzi).

- In ciascuna riga d'ordine deve essere possibile inserire un campo testo (100) di commento. Tale campo viene anche riportato nella stampa cartacea dell'ordine.
- Gli ordini devono essere numerati con numerazione progressiva (la codifica dell'ordine

è standard Smeg come da esempio a seguire:

`codice cliente + progressivo cliente`

Deve anche essere possibile distinguere gli ordini inviati via web da quelli NetSpares e per questo da verificare con EDP Smeg se è accettabile la proposta di identificare gli ordini web mediante una "I" finale.

5a. Gestione preventivi: oltre agli ordini deve essere possibile gestire anche un preventivo (utilizzato dal CAT per le vendite a banco). Si differenzia dall'ordine per il fatto che la stampa riporta solo i prezzi di listino + IVA. Inoltre il CAT può applicare uno sconto variabile inseribile manualmente, diverso per ogni codice. La numerazione dei preventivi è differenziata rispetto a quella degli ordini. I preventivi non vengono mai inviati (per inviarli è necessario inserire un ordine) e non hanno uno stato associato. Inoltre deve essere possibile inserire nel preventivo anche altri campi quali:

- *Nome cliente (per cui e' stato fatto il preventivo)*
- *Indirizzo Cliente*
- *Telefono Cliente*

Oltre a questi campi descrittivi sono presenti anche i campi:

- *manodopera*
- *trasporto*
- *altro*

Nel calcolo del totale del preventivo vanno sommati i valori riportati nei campi aggiuntivi ai prezzi delle parti di ricambio. Il preventivo è stampabile in formato stabilito per consegnare il preventivo al cliente. I preventivi seguono la stessa metodologia di creazione ID degli ordini, ma vengono salvati in una tabella diversa. Possono esistere un ordine e un preventivo con lo stesso ID.

6. Ricerche: deve essere presente una maschera di ricerca tramite cui si apre una finestra con alcune query già salvate (modulo DbTools). Le query principali si identificano in:

- ricerca prezzo per codice
- implosione codice su PF (una volta visualizzati i PF è necessario avere il link per aprire automaticamente il relativo esploso)
- ricerca codice su ordini
- ricerca codice su preventivi

Struttura del database e procedure

Il database originale NetSpares rimane quello attuale. Per rendere i documenti compatibili

con il web la procedura è la seguente:

1. Creazione di una procedura di link automatico a partire dallo stesso documento pdf di partenza di Nertspares. Tale procedura crea un file denominato WEB_Nomemodello.jpg o .gif/.png e un file WEB_Nomemodello.txt che viene salvato in una directory opportuna su server locale. Il file ha il formato necessario al suo utilizzo su internet. Il file bitmap riporta l'immagine da visualizzare, il file di testo contiene il codice necessario a definire una image map dell'immagine con gli opportuni link. La procedura deve anche permettere sul pdf di evidenziare eventuali errori di collegamento o errori di procedura.
2. Creazione di una procedura per l'aggiornamento del database su server web a partire dal database locale.
3. Creazione di una procedura per l'upload del file WEB_* su server web (utilizzo della sincronizzazione cartelle di dbSync).
4. Creazione di una procedura per l'aggiornamento dei link sui pdf presenti su server web per consentire l'evidenziazione dei codici con modifica riportati al punto 2.

Conclusioni

La personalizzazione ha richiesto un notevole lavoro di programmazione. Il case study di Smeg è un esempio di come la piattaforma WebProfessional si collochi tra sistemi di content management di tipo module-based (PHPNuke) e content management frameworks (Zope) garantendo la semplicità ed i ridotti tempi di deployment dei primi unitamente alla flessibilità ed adattabilità dei secondi.

Conclusioni

La piattaforma WebProfessional si pone come soluzione completa per il content management di informazioni orientata alla pubblicazione su web (WCMS).

Le tecnologie utilizzate per lo sviluppo la rendono indipendente dal particolare OS e dal web server utilizzato, pur rimanendo legata a PHP come linguaggio di scripting e MySQL come database di back-end. Questo non costituisce comunque una limitazione alla portabilità (per la possibilità di utilizzare entrambi in ambienti diversi) ed alla stabilità della piattaforma (per il largo utilizzo e supporto presente in rete).

Cos'è WebProfessional

Web Professional è la soluzione semplice e completa per soddisfare le diverse esigenze aziendali, attraverso una serie di programmi Web di nuova generazione, pronti, parametrizzabili, di semplice utilizzo.

⇒ **WP Moduli applicativi**

WebProfessional per i rivenditori

SOFTWARE HOUSE - INTERNET PROVIDERS - WEB DESIGNERS

Con WebProfessional potete fornire ai vostri clienti soluzioni evolute e di veloce realizzazione: la libertà di personalizzazione grafica, la gestione dei contenuti a database e una serie di moduli professionali già pronti che vi consentono di **realizzare servizi internet con funzionalità avanzate** in modo semplice senza bisogno di programmazione per **ottimizzare il vostro lavoro e aumentare la produttività**.

⇒ [vedere i concessionari autorizzati](#)

WebProfessional per le aziende

CON WEBPROFESSIONAL POTETE:

- Integrare dati on-line.
- Creare intranet aziendali.
- Organizzare il vostro lavoro.
- Gestire i contenuti internet.
- Pubblicare cataloghi.
- Raccogliere ordini.
- Disporre di altri strumenti Web evoluti e personalizzabili.

[[Homepage](#)] [[Cos'è WebProfessional](#)] [[Moduli Applicativi](#)] [[Integrazione dati](#)] [[Aree Private - Intranet](#)] [[Organizz. Attività](#)] [[Gestione clienti-CRM](#)] [[Sito Internet - News](#)] [[Cataloghi ed Ordini](#)] [[Archiviazione Doc.](#)] [[Formazione on-line](#)]

Normalmente l'azienda fa riferimento a prodotti disgiunti e specifici per ogni modulo (gestione del sito internet, intranet aziendale, lavoro di gruppo e condivisione delle informazioni internet,...). Soluzioni di questo tipo, se da un lato forniscono funzionalità estese ed estremamente specifiche, dall'altro vengono sottoutilizzate e spesso vengono sottostimati i notevoli costi di integrazione tra moduli sviluppati da fornitori differenti utilizzando tecnologie e filosofie incompatibili.

La piattaforma WebProfessional per contro, con WebPower, eBusiness, DbTools e WP Office offre una soluzione alle casistiche già integrata e modulare. L'attivazione delle diverse sezioni è indipendente dalle altre pur mantenendo una gestione centralizzata degli utenti ed una filosofia di gestione coerente. Le tecnologie utilizzate sono le stesse fra i moduli permettendo all'utente di concentrarsi sui contenuti senza dispendere risorse per la gestione dell'integrazione dei vari moduli.

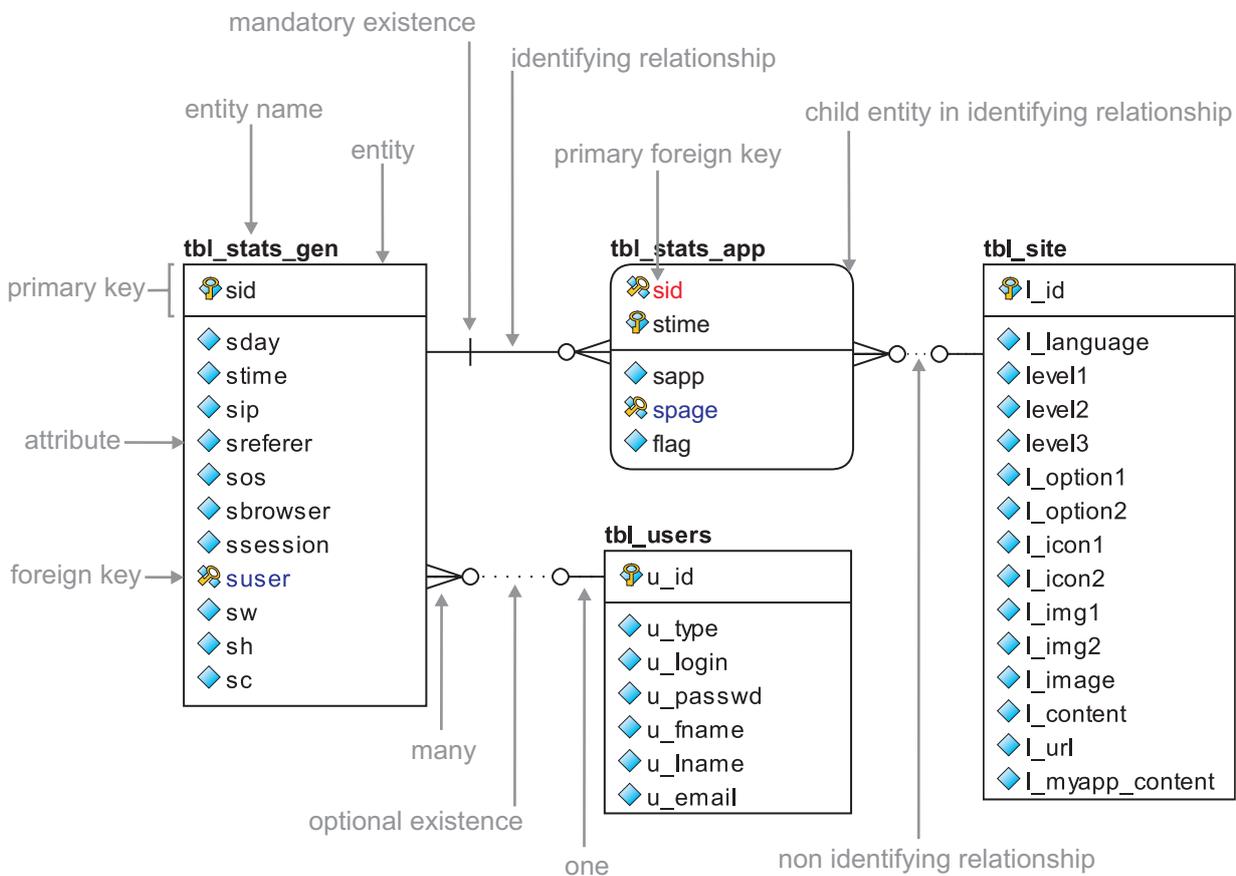
L'assistenza online, il supporto tecnico per i rivenditori ed il costante lavoro di manutenzione ed aggiornamento della piattaforma da parte del team di sviluppo rende la piattaforma WebProfessional la scelta ideale per l'offerta di servizi ed informazioni su internet.

Appendice

Notazione per gli schemi E-R	211
Definizione del database di WebProfessional	212
Tabella dei codici lingua utilizzati in WebProfessional	254
Relazioni nel database WebProfessional	256
Elenco delle relazioni	256
Dettagli delle relazioni	257

Notazione per gli schemi E-R

Nel corso della trattazione tutti gli schemi E-R (Entità-Relazioni) seguiranno la notazione IE (Crow's feet). Nello schema a seguire possiamo vedere i tratti principali di questo tipo di notazione.



Definizione del database di WebProfessional

Statement SQL per la creazione delle strutture dati di WebProfessional su database MySQL.

```

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

# Table structure for table `tbl_acl`
CREATE TABLE `tbl_acl` (
  `id` int(11) NOT NULL auto_increment,
  `source` varchar(64) NOT NULL default '',
  `source_type` char(1) NOT NULL default '',
  `type` char(1) NOT NULL default '',
  `res_id` varchar(255) NOT NULL default '0',
  `op_id` int(11) NOT NULL default '0',
  `target` text NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `idx` (`source`,`source_type`,`op_id`,`res_id`),
  KEY `op_id` (`op_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

# Table structure for table `tbl_acl_myapp`
CREATE TABLE `tbl_acl_myapp` (
  `a_id` tinyint(3) unsigned NOT NULL auto_increment,
  `name` varchar(50) NOT NULL default '???',
  `descr` text NOT NULL,
  `icon_s` varchar(100) default NULL,
  `icon_p` varchar(100) default NULL,
  `sname` varchar(20) NOT NULL default '',
  `s_on_acl` tinyint(3) unsigned NOT NULL default '0',
  `p_on_acl` tinyint(3) unsigned NOT NULL default '0',
  `type` tinyint(3) unsigned NOT NULL default '0',
  `wtarget` varchar(20) default 'l',
  `db_name` varchar(255) default NULL,
  `wpooffice` tinyint(3) NOT NULL default '0',
  `dirname` varchar(255) default NULL,
  PRIMARY KEY (`a_id`),
  UNIQUE KEY `a_id` (`a_id`),
  KEY `a_id_2` (`a_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

# Table structure for table `tbl_acl_myoffice`
CREATE TABLE `tbl_acl_myoffice` (
  `a_id` tinyint(3) unsigned NOT NULL auto_increment,
  `name` varchar(50) NOT NULL default '???',
  `descr` text NOT NULL,
  `icon_s` varchar(100) default NULL,
  `icon_p` varchar(100) default NULL,
  `sname` varchar(20) NOT NULL default '',
  `s_on_acl` tinyint(3) unsigned NOT NULL default '0',
  `p_on_acl` tinyint(3) unsigned NOT NULL default '0',
  `type` tinyint(3) unsigned NOT NULL default '0',
  `wtarget` varchar(20) default 'l',
  `db_name` varchar(255) default NULL,
  `dirname` varchar(255) default NULL,
  PRIMARY KEY (`a_id`),
  UNIQUE KEY `a_id` (`a_id`),
  KEY `a_id_2` (`a_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

# Table structure for table `tbl_acl_op`
CREATE TABLE `tbl_acl_op` (
  `id` int(11) NOT NULL default '0',
  `name` varchar(64) NOT NULL default '',
  `ressource` varchar(50) default NULL,
  `ord` tinyint(3) unsigned default NULL,
  `access` char(1) default NULL,
  `explanation` text,
  PRIMARY KEY (`id`),
  KEY `id` (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

```
# Table structure for table `tbl_acl_public`
CREATE TABLE `tbl_acl_public` (
  `a_id` int(5) unsigned NOT NULL default '0',
  `u_id` int(5) unsigned NOT NULL default '0',
  `flag` char(1) NOT NULL default '0',
  `sname` varchar(255) NOT NULL default '',
  `type` tinyint(3) unsigned NOT NULL default '0',
  PRIMARY KEY (`a_id`,`flag`,`sname`,`u_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

# Table structure for table `tbl_acl_ressources`
CREATE TABLE `tbl_acl_ressources` (
  `id` int(11) NOT NULL auto_increment,
  `name` varchar(64) NOT NULL default '',
  `parent` varchar(64) NOT NULL default '',
  `level` tinyint(4) NOT NULL default '0',
  `expanded` char(1) NOT NULL default '',
  `functionGetChildren` varchar(255) default NULL,
  `prefixChild` varchar(10) default NULL,
  `linkOn` char(1) default '1',
  `defaultPermission` char(1) default '+',
  `defaultchildPermission` char(1) default '+',
  PRIMARY KEY (`id`),
  KEY `id` (`id`),
  KEY `parent` (`parent`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

# Table structure for table `tbl_acl_setup`
CREATE TABLE `tbl_acl_setup` (
  `a_id` int(5) unsigned NOT NULL default '0',
  `u_id` int(5) unsigned NOT NULL default '0',
  `flag` char(1) NOT NULL default '0',
  `sname` varchar(20) NOT NULL default '',
  PRIMARY KEY (`a_id`,`u_id`,`flag`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

# Table structure for table `tbl_admin`
CREATE TABLE `tbl_admin` (
  `company_name` varchar(100) default NULL,
  `company_address1` varchar(255) default NULL,
  `company_address2` varchar(255) default NULL,
  `company_phone` varchar(255) default NULL,
  `company_fax` varchar(255) default NULL,
  `company_info` varchar(30) default NULL,
  `company_login` varchar(50) NOT NULL default '',
  `company_passwd` varchar(50) character set latin1 collate latin1_bin default NULL,
  `company_email` varchar(255) NOT NULL default '',
  `c_language` char(3) NOT NULL default 'IT',
  `c_lang_sel` varchar(50) default NULL,
  `c_lang_adm` char(3) NOT NULL default 'IT',
  `c_sitetype` varchar(50) default NULL,
  `c_datetype` varchar(20) default NULL,
  `c_template` varchar(50) default NULL,
  `c_category` varchar(50) default NULL,
  `c_subcategory` varchar(50) default NULL,
  `c_site` varchar(50) default NULL,
  `c_color` varchar(10) default NULL,
  `c_content_format` varchar(100) default '-*Tahoma, Verdana, Arial, Helvetica*11',
  `c_menu_format1` varchar(100) default '-*Verdana, Arial, Helvetica*12*center*1*0*0',
  `c_menu_format2` varchar(100) default '-*Tahoma, Verdana, Arial, Helvetica*12*left*0*0*0',
  `c_img1` varchar(255) default NULL,
  `c_img2` varchar(255) default NULL,
  `c_img3` varchar(255) default NULL,
  `c_img4` varchar(255) default NULL,
  `c_img5` varchar(255) default NULL,
  `c_my_img1` varchar(255) default NULL,
  `c_my_img2` varchar(255) default NULL,
  `c_my_img3` varchar(255) default NULL,
  `c_my_img4` varchar(255) default NULL,
  `c_my_img5` varchar(255) default NULL,
  `c_start_page` varchar(100) default NULL,
  `c_start_flag` char(1) default NULL,
  `c_log_access` int(10) unsigned default '0',
  `c_log_data` varchar(20) default NULL,
```

```

`c_flag01` char(1) default NULL,
`c_flag02` char(1) default NULL,
`c_flag03` char(1) default NULL,
`c_flag04` char(1) default NULL,
`c_flag05` char(1) default NULL,
`c_flag06` char(1) default NULL,
`c_flag07` char(1) default NULL,
`c_flag08` char(1) default NULL,
`c_flag09` char(1) default NULL,
`c_flag10` char(1) default NULL,
`c_flag11` char(1) default NULL,
`c_flag12` char(1) default NULL,
`c_flag13` char(1) default NULL,
`c_flag14` char(1) default NULL,
`c_flag15` char(1) default NULL,
`c_flag16` char(1) default NULL,
`c_flag17` char(1) default NULL,
`c_flag18` char(1) default NULL,
`c_flag19` char(1) default NULL,
`c_flag20` char(1) default NULL,
`c_flag21` char(1) default NULL,
`c_flag22` char(1) default NULL,
`c_flag23` char(2) default '4',
`c_flag24` char(1) default NULL,
`c_flag25` char(1) default NULL,
`c_flag26` char(1) default NULL,
`c_flag27` char(1) default NULL,
`c_flag28` char(1) default NULL,
`c_flag29` char(1) default NULL,
`c_flag30` char(1) default NULL,
`c_flag31` char(1) default NULL,
`c_flag32` char(1) default NULL,
`c_flag33` char(1) default NULL,
`c_flag34` char(1) default NULL,
`c_flag35` char(1) default NULL,
`c_flag36` char(1) default NULL,
`c_flag37` char(1) default NULL,
`c_flag38` char(1) default NULL,
`c_flag39` char(1) default NULL,
`c_flag40` char(1) default NULL,
`c_flag41` char(1) default NULL,
`c_flag42` char(1) default NULL,
`c_flag43` char(1) default NULL,
`c_flag44` char(1) default NULL,
`c_flag45` char(1) default NULL,
`c_flag46` char(1) default NULL,
`c_flag47` char(1) default NULL,
`c_flag48` char(1) default NULL,
`c_flag49` char(1) default NULL,
`c_flag50` char(1) default NULL,
`c_nr1` smallint(6) default '150',
`c_nr2` smallint(6) default '130',
`c_nr3` smallint(6) default '180',
`c_nr4` smallint(6) default '80',
`c_nr5` int(10) default '0',
`c_nr6` int(10) default NULL,
`c_nr7` int(11) default NULL,
`c_colorbkg` varchar(10) default NULL,
`c_text1` varchar(50) default NULL,
`c_text2` varchar(50) default NULL,
`c_text3` varchar(50) default NULL,
`c_text4` varchar(50) default NULL,
`c_text5` varchar(255) default NULL,
`c_session` varchar(255) default NULL,
`user_text` text,
`SyncStatus` char(1) NOT NULL default 'A',
`c_escapedpasswd` tinyint(3) unsigned default NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

# Table structure for table `tbl_admin_labels`
CREATE TABLE `tbl_admin_labels` (
  `lang` char(2) NOT NULL default 'EN',
  `format` varchar(255) default NULL,
  `l_search` varchar(50) default 'Search',
  `l_news` varchar(50) default 'E-news',
  `l_map` varchar(50) default 'Site map',

```

```
`l_mail` varchar(50) default 'Mailing-list',
`l_upldnl` varchar(50) default 'Download',
`l_links` varchar(50) default 'Links',
`l_promo` varchar(50) default NULL,
`l_ecommerce` varchar(50) default 'E-commerce',
`l_dbtools` varchar(50) default 'WebDatabase',
`l_userarea` varchar(50) default 'User area',
`l_register` varchar(50) default 'Register',
`l_visitors` varchar(50) default 'Visitors',
`l_txt1` text,
`l_txt2` text,
`l_txt3` varchar(50) default NULL,
`l_txt4` varchar(50) default NULL,
`l_txt5` varchar(50) default NULL,
`l_txt6` varchar(50) default NULL,
`l_txt7` varchar(50) default NULL,
`l_txt8` varchar(50) default NULL,
`l_txt9` varchar(50) default NULL,
`l_txt10` varchar(50) default NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='Labels for interface';
```

Table structure for table 'tbl_admin_txt'

```
CREATE TABLE `tbl_admin_txt` (
  `id` int(5) NOT NULL default '0',
  `msg_1` varchar(255) default NULL,
  `msg_txt` text,
  `lang` char(2) default 'IT',
  `sname` varchar(255) default NULL,
  UNIQUE KEY `id` (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Table structure for table 'tbl_agents'

```
CREATE TABLE `tbl_agents` (
  `cust_code` varchar(20) NOT NULL default '0',
  `agent_code` varchar(20) NOT NULL default '0',
  `SyncStatus` char(1) NOT NULL default 'A',
  PRIMARY KEY (`cust_code`,`agent_code`)
```

```
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='define relations between a user and agents';
```

Table structure for table 'tbl_agents_commissions'

```
CREATE TABLE `tbl_agents_commissions` (
  `cust_code` varchar(20) NOT NULL default "",
  `code` varchar(100) NOT NULL default "",
  `commission` double NOT NULL default '0',
  `from_value` double NOT NULL default '0',
  `to_value` double NOT NULL default '0',
  `SyncStatus` char(1) NOT NULL default 'A',
  PRIMARY KEY (`cust_code`,`code`,`from_value`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='agent commission for specific product';
```

Table structure for table 'tbl_artroot'

```
CREATE TABLE `tbl_artroot` (
  `codroot` varchar(12) NOT NULL default "",
  `descr` varchar(40) NOT NULL default "",
  `numliv` smallint(2) NOT NULL default '0',
  `lungroot` smallint(2) unsigned default '0',
  `lungvar1` smallint(2) NOT NULL default '0',
  `lungvar2` smallint(2) unsigned NOT NULL default '0',
  `lungvar3` smallint(2) unsigned NOT NULL default '0',
  `lungvar4` smallint(2) unsigned NOT NULL default '0',
  `lungvar5` smallint(2) unsigned NOT NULL default '0',
  `lungvar6` smallint(2) unsigned NOT NULL default '0',
  `lungvar7` smallint(2) unsigned NOT NULL default '0',
  `lungvar8` smallint(2) unsigned NOT NULL default '0',
  `lungvar9` tinyint(2) unsigned NOT NULL default '0',
  `lungvar10` smallint(2) unsigned NOT NULL default '0',
  `prevar1` char(1) NOT NULL default 'N',
  `code_ing` char(2) NOT NULL default 'IT',
  `SyncStatus` char(1) NOT NULL default 'A',
  PRIMARY KEY (`codroot`,`code_ing`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Table structure for table 'tbl_artvar'

```
CREATE TABLE `tbl_artvar` (
```

```
`codroot` varchar(12) NOT NULL default "",
`livello` smallint(2) unsigned NOT NULL default '0',
`codvar` varchar(50) NOT NULL default "",
`descr` varchar(40) default NULL,
`code_lng` char(2) NOT NULL default 'IT',
`filevar` varchar(255) default NULL,
`SyncStatus` char(1) NOT NULL default 'A',
`price` double default NULL,
`defaultInCombo` char(1) default NULL,
PRIMARY KEY (`codroot`,`livello`,`codvar`,`code_lng`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
# Table structure for table `tbl_cat_objects`
CREATE TABLE `tbl_cat_objects` (
  `code_cat` varchar(100) NOT NULL default '0',
  `code` varchar(100) NOT NULL default '0',
  `position_obj` int(10) unsigned NOT NULL default '0',
  `SyncStatus` char(1) NOT NULL default 'A',
  `cat_special_offer` tinyint(3) unsigned default '0',
  PRIMARY KEY (`code_cat`,`code`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='define to which categories belongs an object';
```

```
# Table structure for table `tbl_cat_type`
CREATE TABLE `tbl_cat_type` (
  `type_cat` int(10) unsigned NOT NULL default '0',
  `type_cat_descr` varchar(255) NOT NULL default '?',
  `type_cat_note` text,
  `type_cat_icon` varchar(30) default NULL,
  `code_lng` char(2) NOT NULL default "",
  `SyncStatus` char(1) NOT NULL default 'A',
  `f_txt1` text,
  `f_txt2` text,
  `f_txt3` text,
  `f_int` tinyint(3) unsigned default NULL,
  `f_int1` tinyint(3) unsigned default NULL,
  `f_int2` tinyint(3) unsigned default NULL,
  PRIMARY KEY (`type_cat`,`code_lng`)
```

```
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='define the types of categorization available';
```

```
# Table structure for table `tbl_categories`
CREATE TABLE `tbl_categories` (
  `code_cat` varchar(100) NOT NULL default '0',
  `type_cat` int(10) unsigned NOT NULL default '0',
  `code_lng` char(2) NOT NULL default 'IT',
  `descr_short_cat` varchar(50) NOT NULL default '?',
  `descr_long_cat` text,
  `picture_cat_default` varchar(255) default NULL,
  `code_father` varchar(100) NOT NULL default '0',
  `position_cat` int(10) unsigned NOT NULL default '1',
  `icon_cat` varchar(255) default NULL,
  `icon_over_cat` varchar(225) default NULL,
  `cat_link` varchar(255) default NULL,
  `cat_file` varchar(255) default NULL,
  `show_cat` char(1) default '1',
  `SyncStatus` char(1) NOT NULL default 'A',
  `picture_cat` varchar(255) default NULL,
  `f_txt1` text,
  `f_txt2` text,
  `f_txt3` text,
  `f_int` tinyint(3) unsigned default NULL,
  `f_int1` int(10) unsigned default NULL,
  `f_int2` int(10) unsigned default NULL,
  `sync_descr2` varchar(255) default NULL,
  `UNSPSC` varchar(8) default NULL,
  PRIMARY KEY (`code_cat`,`code_lng`,`code_father`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='contains the list of all available categories';
```

```
# Table structure for table `tbl_config_ecomm`
CREATE TABLE `tbl_config_ecomm` (
  `show_variants` tinyint(3) unsigned default '1',
  `show_spareparts` tinyint(3) unsigned default '1',
  `nr_prod_page` tinyint(3) unsigned default '4',
  `show_large_image` tinyint(3) unsigned default NULL,
  `stock_management` tinyint(3) unsigned default '2',
```

`nreg_prices` tinyint(3) unsigned default '1',
`nreg_order` tinyint(3) unsigned default '1',
`template_prod_list` tinyint(3) unsigned default '1',
`template_prod_detail` tinyint(3) unsigned default '1',
`template_spare` tinyint(3) unsigned default '0',
`template_variants` tinyint(3) unsigned default '0',
`spare_table_style` mediumtext,
`variants_table_style` mediumtext,
`flag_1` tinyint(3) unsigned default NULL,
`flag_2` tinyint(3) unsigned default NULL,
`flag_3` tinyint(3) unsigned default NULL,
`flag_4` tinyint(3) unsigned default NULL,
`flag_6` tinyint(3) unsigned default NULL,
`flag_5` tinyint(3) unsigned default NULL,
`flag_7` tinyint(3) unsigned default NULL,
`flag_8` tinyint(3) unsigned default NULL,
`flag_9` tinyint(3) unsigned default NULL,
`flag_10` tinyint(3) unsigned default NULL,
`content_1` mediumtext,
`content_2` mediumtext,
`show_firstpage` tinyint(4) NOT NULL default '40',
`show_categ` tinyint(4) default '13',
`show_av_qty` tinyint(3) unsigned default '1',
`catalog` tinyint(3) unsigned default '0',
`lowqty_send_mail` tinyint(3) unsigned default '1',
`show_code_alt` tinyint(3) unsigned default '1',
`show_categ_link` tinyint(3) unsigned default '1',
`show_categ_file` tinyint(3) unsigned default '1',
`show_categ_defaultpicture` tinyint(3) unsigned default '1',
`icon_categ` tinyint(3) unsigned default '1',
`show_prod_descr` tinyint(3) unsigned default '1',
`show_prod_notes` tinyint(3) unsigned default '1',
`show_prod_label` tinyint(3) unsigned default '1',
`show_prod_file` tinyint(3) unsigned default '1',
`show_prod_link` tinyint(3) unsigned default '1',
`same_product_picture` tinyint(3) unsigned default '0',
`show_prod_code` tinyint(3) unsigned default '1',
`content_3` varchar(255) default NULL,
`content_4` varchar(255) default NULL,
`content_5` varchar(255) default NULL,
`content_6` mediumtext,
`content_7` mediumtext,
`content_8` mediumtext,
`content_9` mediumtext,
`content_10` varchar(255) default NULL,
`serie_order` varchar(50) default '*',
`min_numord` bigint(20) unsigned default NULL,
`max_numord` bigint(20) unsigned default NULL,
`show_standardpaycond` tinyint(3) unsigned default NULL,
`order_altcode` char(1) default NULL,
`order_useavqty` char(1) default NULL,
`order_showconsdate` char(1) default NULL,
`order_show_quickreport` char(1) default NULL,
`order_useagreement` char(1) default NULL,
`search_hidecategs` char(1) default NULL,
`order_stockmessage` char(1) default NULL,
`order_acceptedcountries` text,
`user_changeprophile` char(1) default NULL,
`user_changeprofopenorder` char(1) default NULL,
`order_hideshippaddr` char(1) default NULL,
`order_maxbasketqty` int(11) default '-1',
`show_productwithoutprice` char(1) default NULL,
`order_evasionepartiale` char(1) default NULL,
`order_checkfiscalecode` char(1) default NULL,
`order_usebillingaddress` char(1) default NULL,
`ecomm_logo` varchar(255) default NULL,
`frame_version` char(1) default NULL,
`frame_heightbanner` tinyint(3) unsigned default NULL,
`frame_widthmenu` tinyint(3) unsigned default NULL,
`frame_menubgimg` varchar(255) default NULL,
`frame_menubgcolor` varchar(255) default NULL,
`frame_mainbgimg` varchar(255) default NULL,
`frame_mainbgcolor` varchar(255) default NULL,
`agent_viewuserorders` char(1) default NULL,
`agent_modifyclientdata` char(1) default NULL,
`order_maxbasketqtyemail` varchar(50) default NULL,

```

`order_showqtydec` char(1) default NULL,
`order_useseUM` char(1) default NULL,
`ecom_homeaddress` varchar(255) default NULL,
`pack_unitweight` varchar(10) default NULL,
`pack_unitlength` varchar(10) default NULL,
`pack_unitwidth` varchar(10) default NULL,
`pack_unitheight` varchar(10) default NULL,
`variants_display` tinyint(3) unsigned default NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='configuration e-commerce display';

# Table structure for table `tbl_creditcard`
CREATE TABLE `tbl_creditcard` (
  `cc_id` tinyint(3) unsigned NOT NULL default '0',
  `cc_title` varchar(255) default NULL,
  `cc_description` varchar(255) default NULL,
  `cc_phpfile` varchar(255) default NULL,
  `cc_fl_gendb` char(1) default NULL,
  PRIMARY KEY (`cc_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

# Table structure for table `tbl_currency`
CREATE TABLE `tbl_currency` (
  `tc_codecurrency` varchar(20) NOT NULL default '',
  `tc_descrcurrency` varchar(50) default NULL,
  `tc_symbolcurrency` varchar(10) default NULL,
  `tc_HTMLsymbolcurrency` varchar(10) default NULL,
  `tc_exchangerate` double default NULL,
  `tc_multdiv` char(1) default 'D',
  `tc_nrdec` int(10) unsigned default NULL,
  `SyncStatus` char(1) default 'A',
  PRIMARY KEY (`tc_codecurrency`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

# Table structure for table `tbl_currency_exchange`
CREATE TABLE `tbl_currency_exchange` (
  `tce_exchangerate` double default NULL,
  `tce_codecurrency` varchar(20) NOT NULL default '',
  `tce_codepricelist` varchar(100) NOT NULL default '0',
  `tce_datestart` date NOT NULL default '2000-01-01',
  `tce_dateend` date default '2100-01-01',
  `SyncStatus` char(1) default 'A',
  PRIMARY KEY (`tce_codecurrency`,`tce_codepricelist`,`tce_datestart`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

# Table structure for table `tbl_custapp_exportsync`
CREATE TABLE `tbl_custapp_exportsync` (
  `table_name` varchar(50) NOT NULL default '',
  `table_descr` text,
  `table_database` varchar(50) NOT NULL default '',
  PRIMARY KEY (`table_database`,`table_name`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='table with names of the custom applications';

# Table structure for table `tbl_disc_class`
CREATE TABLE `tbl_disc_class` (
  `code_disc_class` varchar(100) NOT NULL default '0',
  `descr_class` varchar(200) default NULL,
  `notes_class` varchar(255) default NULL,
  `title_class` varchar(50) NOT NULL default 'untitled',
  `type_class` char(1) NOT NULL default 'c',
  `SyncStatus` char(1) NOT NULL default 'A',
  `f_txt1` text,
  `f_txt2` text,
  `f_txt3` text,
  `f_int1` tinyint(3) unsigned default NULL,
  `f_int2` tinyint(3) unsigned default NULL,
  `f_int3` tinyint(3) unsigned default NULL,
  PRIMARY KEY (`code_disc_class`,`type_class`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='discount classes table';

# Table structure for table `tbl_discounts`
CREATE TABLE `tbl_discounts` (
  `code` varchar(100) NOT NULL default '0',

```

```

`discount_1` double NOT NULL default '0',
`discount_2` double NOT NULL default '0',
`discount_3` double NOT NULL default '0',
`discount_4` double NOT NULL default '0',
`discount_5` double NOT NULL default '0',
`qty_from` double NOT NULL default '0',
`qty_to` double NOT NULL default '999999999999999',
`code_disc_class` varchar(100) NOT NULL default '0',
`date_start` date NOT NULL default '2000-01-01',
`date_end` date NOT NULL default '2100-01-01',
`SyncStatus` char(1) NOT NULL default 'A',
`cust_code` varchar(20) NOT NULL default '0',
`code_client_class` varchar(100) NOT NULL default '0',
`codtpro` varchar(50) NOT NULL default '0',
PRIMARY KEY (`code`,`qty_from`,`code_disc_class`,`date_
start`,`cust_code`,`code_client_class`,`codtpro`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='discounts table';

```

```

# Table structure for table `tbl_ec_basket`
CREATE TABLE `tbl_ec_basket` (
  `b_uid` char(20) NOT NULL default '',
  `b_riga` tinyint(3) unsigned NOT NULL default '0',
  `b_code` char(20) default NULL,
  `b_qty` double default NULL,
  `b_date` datetime default NULL,
  `b_um` char(20) default NULL,
  PRIMARY KEY (`b_uid`,`b_riga`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='contains temporary basket';

```

```

# Table structure for table `tbl_ec_measureunit`
CREATE TABLE `tbl_ec_measureunit` (
  `um_code` varchar(20) NOT NULL default '',
  `um_descr` varchar(255) default NULL,
  `um_note` text,
  `code_ing` char(2) NOT NULL default 'IT',
  `SyncStatus` char(1) NOT NULL default 'A',
  PRIMARY KEY (`um_code`,`code_ing`)

```

```

) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='Ebusiness measure units';

```

```

# Table structure for table `tbl_ec_measureunit_conversion`
CREATE TABLE `tbl_ec_measureunit_conversion` (
  `umc_procode` varchar(100) NOT NULL default '',
  `umc_code` varchar(20) NOT NULL default '',
  `umc_convratio` double default '1',
  `umc_buyingmultiple` double default NULL,
  `umc_default` tinyint(2) default '0',
  `SyncStatus` char(1) default 'A',
  PRIMARY KEY (`umc_procode`,`umc_code`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='Measure units conversion';

```

```

# Table structure for table `tbl_ec_varexcluded`
CREATE TABLE `tbl_ec_varexcluded` (
  `codroot` varchar(100) NOT NULL default '',
  `codvari_1` varchar(50) NOT NULL default '',
  `valvari_1` varchar(50) NOT NULL default '',
  `codvari_2` varchar(50) NOT NULL default '',
  `valvari_2` varchar(50) NOT NULL default '',
  `SyncStatus` char(1) default 'A',
  PRIMARY KEY (`codroot`,`codvari_1`,`valvari_1`,`codvari_
2`,`valvari_2`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='variants not concurring on two types';

```

```

# Table structure for table `tbl_exportlog`
CREATE TABLE `tbl_exportlog` (
  `l_id` tinyint(3) unsigned NOT NULL auto_increment,
  `l_startdate` datetime default NULL,
  `l_enddate` datetime default NULL,
  `l_deleted` char(1) default NULL,
  `l_expdate` datetime default NULL,
  `l_filename` varchar(255) default NULL,
  PRIMARY KEY (`l_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

```

# Table structure for table `tbl_group`

```

```
CREATE TABLE `tbl_group` (
  `g_group` varchar(50) default NULL,
  `g_info` varchar(100) default NULL,
  `flag` char(1) default NULL,
  `code_group` int(10) unsigned NOT NULL default '1',
  `description` varchar(50) default NULL,
  `SyncStatus` char(1) NOT NULL default 'A',
  UNIQUE KEY `NewIndex` (`code_group`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Table structure for table `tbl_links`

```
CREATE TABLE `tbl_links` (
  `links_id` int(10) NOT NULL default '0',
  `links_categ` varchar(255) default NULL,
  `links_categ1` varchar(255) default '???',
  `links_categ2` varchar(255) default '???',
  `links_categ3` varchar(255) default '???',
  `links_categ4` varchar(255) default '???',
  `links_categ5` varchar(255) default '???',
  `links_categ6` varchar(255) default '???',
  `links_categ7` varchar(255) default '???',
  `links_title` varchar(255) default NULL,
  `links_title1` varchar(255) default NULL,
  `links_title2` varchar(255) default '???',
  `links_title3` varchar(255) default '???',
  `links_title4` varchar(255) default '???',
  `links_title5` varchar(255) default '???',
  `links_title6` varchar(255) default '???',
  `links_title7` varchar(255) default '???',
  `links_link` varchar(255) default NULL,
  `links_priority` int(10) default NULL,
  `flag` char(1) default NULL,
  `SyncStatus` char(1) NOT NULL default 'A',
  UNIQUE KEY `links_id` (`links_id`),
  KEY `i_links` (`links_title`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Table structure for table `tbl_loglast`

```
CREATE TABLE `tbl_loglast` (
  `val` int(11) default NULL,
  `laststell` int(11) default NULL,
  `lastline` varchar(255) default NULL,
  `site` varchar(255) default NULL,
  `ll_cflag1` char(1) default NULL,
  `ll_cflag2` char(1) default NULL,
  `ll_vcflag1` varchar(255) default NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Table structure for table `tbl_logserver`

```
CREATE TABLE `tbl_logserver` (
  `pk` varchar(255) NOT NULL default "",
  `ip` varchar(127) default NULL,
  `unknown1` varchar(127) default NULL,
  `unknown2` varchar(127) default NULL,
  `mysqltime` datetime default NULL,
  `method` varchar(127) default NULL,
  `document` varchar(127) default NULL,
  `version` varchar(127) default NULL,
  `response` int(11) default NULL,
  `bytes` int(11) default NULL,
  `referrer` varchar(127) default NULL,
  `agent` varchar(127) default NULL,
  `site` varchar(127) default NULL,
  `sl_cflag1` char(1) default NULL,
  `sl_cflag2` char(1) default NULL,
  `sl_cflag3` char(1) default NULL,
  `sl_vcflag1` varchar(255) default NULL,
  PRIMARY KEY (`pk`),
  KEY `index_mysqltime` (`mysqltime`),
  KEY `index_site` (`pk`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Table structure for table `tbl_maillist`

```
CREATE TABLE `tbl_maillist` (
  `m_id` int(10) NOT NULL default '0',
  `maillist` varchar(255) default NULL,
```

```
`m_ord` tinyint(4) default '0',
`m_descr1` varchar(255) default NULL,
`m_descr2` varchar(255) default NULL,
`m_descr3` varchar(255) default NULL,
`m_descr4` varchar(255) default NULL,
`m_descr5` varchar(255) default NULL,
`m_descr6` varchar(255) default NULL,
`m_descr7` varchar(255) default NULL,
`SyncStatus` char(1) NOT NULL default 'A',
UNIQUE KEY `m_id` (`m_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

# Table structure for table `tbl_mails`
CREATE TABLE `tbl_mails` (
  `mail_id` int(10) NOT NULL auto_increment,
  `mail_to` varchar(255) default NULL,
  `mail_subject` varchar(255) default NULL,
  `mail_attach` varchar(255) default NULL,
  `mail_date` varchar(20) default NULL,
  `flag` char(1) default NULL,
  `SyncStatus` char(1) NOT NULL default 'A',
  UNIQUE KEY `mail_id` (`mail_id`),
  KEY `i_mails` (`mail_date`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

# Table structure for table `tbl_news`
CREATE TABLE `tbl_news` (
  `news_id` int(10) NOT NULL default '0',
  `news_date` varchar(20) default NULL,
  `news_group` text,
  `news_priority` smallint(4) unsigned NOT NULL default '0',
  `ord_categ` smallint(4) unsigned NOT NULL default '0',
  `news_title_1` varchar(100) default NULL,
  `news_title_2` varchar(100) default NULL,
  `news_title_3` varchar(100) default NULL,
  `news_title_4` varchar(100) default NULL,
  `news_title_5` varchar(100) default NULL,
  `news_title_6` varchar(100) default NULL,
  `news_title_7` varchar(100) default NULL,
  `news_descr_1` text,
  `news_descr_2` text,
  `news_descr_3` text,
  `news_descr_4` text,
  `news_descr_5` text,
  `news_descr_6` text,
  `news_descr_7` text,
  `news_photo_1` varchar(100) default NULL,
  `news_photo_2` varchar(100) default NULL,
  `news_photo_3` varchar(100) default NULL,
  `news_photo_4` varchar(100) default NULL,
  `news_photo_5` varchar(100) default NULL,
  `news_photo_6` varchar(100) default NULL,
  `news_photo_7` varchar(100) default NULL,
  `news_pdf_1` varchar(100) default NULL,
  `news_pdf_2` varchar(100) default NULL,
  `news_pdf_3` varchar(100) default NULL,
  `news_pdf_4` varchar(100) default NULL,
  `news_pdf_5` varchar(100) default NULL,
  `news_pdf_6` varchar(100) default NULL,
  `news_pdf_7` varchar(100) default NULL,
  `news_link_1` varchar(200) default NULL,
  `news_link_2` varchar(200) default NULL,
  `news_link_3` varchar(200) default NULL,
  `news_link_4` varchar(200) default NULL,
  `news_link_5` varchar(200) default NULL,
  `news_link_6` varchar(200) default NULL,
  `news_link_7` varchar(200) default NULL,
  `news_post` char(1) default NULL,
  `news_all` char(1) default NULL,
  `flag1` char(1) default NULL,
  `flag2` char(1) default NULL,
  `flag3` char(1) default NULL,
  `SyncStatus` char(1) NOT NULL default 'S',
  `news_date_end` datetime NOT NULL default '0000-00-00 00:00:00',
  PRIMARY KEY (`news_id`),
```

```

KEY `i_news` (`ord_categ`,`news_date`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

# Table structure for table `tbl_news_categ`
CREATE TABLE `tbl_news_categ` (
  `cod_categ` smallint(4) unsigned NOT NULL auto_increment,
  `ord_categ` smallint(4) unsigned NOT NULL default '0',
  `categ_lang1` varchar(50) default NULL,
  `categ_lang2` varchar(50) default NULL,
  `categ_lang3` varchar(50) default NULL,
  `categ_lang4` varchar(50) default NULL,
  `categ_lang5` varchar(50) default NULL,
  `categ_lang6` varchar(50) default NULL,
  `categ_lang7` varchar(50) default NULL,
  `flag1` varchar(10) default NULL,
  `c_group` varchar(50) default NULL,
  `c_priority` smallint(4) unsigned NOT NULL default '0',
  `SyncStatus` char(1) NOT NULL default 'S',
  PRIMARY KEY (`cod_categ`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

# Table structure for table `tbl_objects`
CREATE TABLE `tbl_objects` (
  `code` varchar(100) NOT NULL default "",
  `code_alt` varchar(20) default NULL,
  `descr_short_obj` varchar(255) default NULL,
  `descr_long_obj` text,
  `weight` double default NULL,
  `dim_l` double default NULL,
  `dim_w` double default NULL,
  `dim_h` double default NULL,
  `last_modified` datetime NOT NULL default '0000-00-00
00:00:00',
  `vat` double NOT NULL default '0',
  `vat_categcode` varchar(10) default NULL,
  `show_dec` tinyint(4) unsigned NOT NULL default '0',
  `qty_available` double default '0',
  `qty_low` double default NULL,
  `qty_high` double default NULL,
  `UNSPSC` varchar(50) default NULL,
  `code_disc_class` varchar(100) NOT NULL default '0',
  `SyncStatus` char(1) NOT NULL default 'A',
  `f_txt1` text,
  `f_txt2` text,
  `f_txt3` text,
  `f_int1` tinyint(3) unsigned default NULL,
  `f_int2` int(10) unsigned default NULL,
  `f_int3` int(10) unsigned default NULL,
  `special_offer_all` tinyint(3) unsigned default '0',
  `is_service` char(1) default NULL,
  `codroot` varchar(12) default NULL,
  `codvar1` varchar(8) default NULL,
  `codvar2` varchar(8) default NULL,
  `codvar3` varchar(8) default NULL,
  `codvar4` varchar(8) default NULL,
  `codvar5` varchar(8) default NULL,
  `codvar6` varchar(8) default NULL,
  `codvar7` varchar(8) default NULL,
  `codvar8` varchar(8) default NULL,
  `codvar9` varchar(8) default NULL,
  `codvar10` varchar(8) default NULL,
  `codvar11` varchar(8) default NULL,
  `codvar12` varchar(8) default NULL,
  `codvar13` varchar(8) default NULL,
  `codvar14` varchar(8) default NULL,
  `codvar15` varchar(8) default NULL,
  `codvar16` varchar(8) default NULL,
  `codvar17` varchar(8) default NULL,
  `codvar18` varchar(8) default NULL,
  `codvar19` varchar(8) default NULL,
  `codvar20` varchar(8) default NULL,
  `esplosi_replacingcode` varchar(100) default NULL,
  PRIMARY KEY (`code`),
  KEY `idx_code_alt` (`code_alt`),
  KEY `idx_last_modified` (`last_modified`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1

```

```
COMMENT='contains the list of all objects (articles)';
```

```
# Table structure for table 'tbl_objects_fields'
```

```
CREATE TABLE `tbl_objects_fields` (  
  `code` varchar(100) NOT NULL default '0',  
  `code_ing` char(2) NOT NULL default 'IT',  
  `fieldname` varchar(50) default NULL,  
  `value` varchar(255) default NULL,  
  `unit_of_mesure` varchar(50) default NULL,  
  `code_field` varchar(50) NOT NULL default '0',  
  `SyncStatus` char(1) NOT NULL default 'A',  
  PRIMARY KEY (`code`,`code_field`,`code_ing`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1  
COMMENT='contains the list of all custom fields of each  
object';
```

```
# Table structure for table 'tbl_objects_ing'
```

```
CREATE TABLE `tbl_objects_ing` (  
  `code` varchar(100) NOT NULL default '0',  
  `code_ing` char(2) NOT NULL default 'IT',  
  `descr_short_ing` varchar(255) NOT NULL default '???',  
  `descr_long_ing` text,  
  `notes` text,  
  `image_file` varchar(255) default NULL,  
  `unit_label` varchar(20) default NULL,  
  `image_file1` varchar(255) default NULL,  
  `link_obj` varchar(255) default NULL,  
  `file_obj` varchar(255) default NULL,  
  `show_obj` char(1) default '1',  
  `SyncStatus` char(1) NOT NULL default 'A',  
  `f_txt1` text,  
  `f_txt2` text,  
  `f_txt3` text,  
  `f_int1` tinyint(3) unsigned default NULL,  
  `f_int2` tinyint(3) unsigned default NULL,  
  `f_int3` tinyint(3) unsigned default NULL,  
  `sync_descr2` varchar(255) default NULL,  
  PRIMARY KEY (`code`,`code_ing`),
```

```
  KEY `idx_descr_short_ing` (`descr_short_ing`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1  
COMMENT='contains the descriptions in language for each  
object';
```

```
# Table structure for table 'tbl_objects_pers'
```

```
CREATE TABLE `tbl_objects_pers` (  
  `cust_code` varchar(20) NOT NULL default '',  
  `code` varchar(100) NOT NULL default '',  
  `template_id` int(10) unsigned NOT NULL default '0',  
  `SyncStatus` char(1) NOT NULL default 'A',  
  PRIMARY KEY (`cust_code`,`code`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1  
COMMENT='personalized list of objects for each customer';
```

```
# Table structure for table 'tbl_objects_servicemail'
```

```
CREATE TABLE `tbl_objects_servicemail` (  
  `e_code` varchar(100) NOT NULL default '',  
  `code_ing` char(2) NOT NULL default 'IT',  
  `e_subject` tinyblob,  
  `e_message` blob,  
  `e_attachments` tinyblob,  
  `e_reply` varchar(255) default NULL,  
  `SyncStatus` char(1) NOT NULL default 'A',  
  PRIMARY KEY (`e_code`,`code_ing`),  
  KEY `e_code` (`e_code`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
# Table structure for table 'tbl_objects_show'
```

```
CREATE TABLE `tbl_objects_show` (  
  `code` varchar(100) NOT NULL default '',  
  `code_cat` int(10) unsigned NOT NULL default '0',  
  `code_ing` char(2) NOT NULL default '',  
  `description` varchar(200) NOT NULL default '',  
  `notes` text,  
  `image` varchar(50) default NULL,  
  `link` varchar(250) default NULL,  
  `SyncStatus` char(1) NOT NULL default 'A',  
  PRIMARY KEY (`code`,`code_cat`,`code_ing`)
```

```
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='objects to be displayed in evidence on the
categories';
```

```
# Table structure for table `tbl_orders_ccpayd`
```

```
CREATE TABLE `tbl_orders_ccpayd` (
  `ordernumber` bigint(20) unsigned NOT NULL default '0',
  `cc_id` tinyint(3) unsigned default '0',
  `date` datetime default NULL,
  `payd_sum` double default NULL,
  `triv_compagnia` char(2) default NULL,
  `triv_ragsoc` char(50) default NULL,
  `triv_numauthor` char(255) default NULL,
  `triv_numoper` char(255) default NULL,
  `triv_rispcomp` char(255) default NULL,
  PRIMARY KEY (`ordernumber`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
# Table structure for table `tbl_orders_details`
```

```
CREATE TABLE `tbl_orders_details` (
  `ordernumber` varchar(50) NOT NULL default '0',
  `code` varchar(100) NOT NULL default '0',
  `price` double NOT NULL default '0',
  `price_currency` double default NULL,
  `quantity` double NOT NULL default '0',
  `quantity_2` double default NULL,
  `completed` char(1) default '0',
  `SyncStatus` char(1) NOT NULL default 'A',
  `qty_approved` double default NULL,
  `flag_appr` char(1) default NULL,
  `flag1_int` int(10) unsigned default NULL,
  `flag2_int` double default NULL,
  `flag3` varchar(255) default NULL,
  `tipork` char(1) NOT NULL default 'R',
  `flag4` varchar(255) default NULL,
  `serie` varchar(50) NOT NULL default '*',
  `numord` varchar(50) NOT NULL default '0',
  `qty_evasa` double NOT NULL default '0',
```

```
`anno` int(10) unsigned NOT NULL default '0',
`riga` int(4) unsigned NOT NULL default '0',
`unit` varchar(20) default NULL,
`unit_2` varchar(20) default NULL,
`data_cons` datetime default NULL,
`discount1` double default '0',
`discount2` double default '0',
`discount3` double default '0',
`discount4` double default '0',
`discount5` double default '0',
`prod_title` varchar(255) default NULL,
`total` float NOT NULL default '0',
`vat_categcode` varchar(10) NOT NULL default '',
PRIMARY KEY (`ordernumber`,`anno`,`serie`,`tipork`,`riga`,`numord`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='rows of the orders (each one is related to an
header)';
```

```
# Table structure for table `tbl_orders_heads`
```

```
CREATE TABLE `tbl_orders_heads` (
  `ordernumber` varchar(50) NOT NULL default '0',
  `numord` varchar(50) NOT NULL default '0',
  `cust_code` varchar(20) NOT NULL default '0',
  `agent_code` varchar(20) default NULL,
  `order_datetime` datetime NOT NULL default '0000-00-00
00:00:00',
  `price_list` varchar(100) NOT NULL default '0',
  `note` text,
  `completed` char(1) default '0',
  `SyncStatus` char(1) NOT NULL default 'A',
  `code_payment` varchar(100) default NULL,
  `totalvat` float NOT NULL default '0',
  `validate` char(1) NOT NULL default '0',
  `anno` int(10) unsigned NOT NULL default '0',
  `flag1_int` int(10) unsigned default NULL,
  `flag2_int` float default NULL,
  `flag3` varchar(255) default NULL,
  `flag4` varchar(255) default NULL,
```

```
`data_cons` datetime default NULL,  
`order_url` varchar(255) default NULL,  
`order_file` varchar(255) default NULL,  
`tipork` char(1) NOT NULL default 'R',  
`serie` varchar(50) NOT NULL default '*',  
`total` double default NULL,  
`closed` char(1) default '0',  
`discount1` double default '0',  
`discount2` double default '0',  
`discount3` double default '0',  
`discount4` double default '0',  
`discount5` double default '0',  
`payc_discount` float default '0',  
`payc_expenses` float default '0',  
`payc_expensescurrency` float default NULL,  
`payc_percexpenses` float default '0',  
`payc_vatexpenses` float default '0',  
`cc_method` int(3) unsigned default NULL,  
`cc_payd` char(1) default '0',  
`cc_name` varchar(255) default NULL,  
`coddest` varchar(20) default NULL,  
`shipping_address` text,  
`order_emailaddress` varchar(255) default NULL,  
`order_codecurrency` varchar(20) default NULL,  
`order_mutdivcurrency` char(1) default NULL,  
`order_exchangerate` float default NULL,  
`order_symbolcurrency` varchar(10) default NULL,  
`order_evasionepartiale` char(1) default NULL,  
`offer_name` varchar(255) default NULL,  
PRIMARY KEY (`ordernumber`, `numord`, `tipork`, `serie`, `an  
no`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1  
COMMENT='header of the orders';
```

```
# Table structure for table `tbl_orders_shippings`  
CREATE TABLE `tbl_orders_shippings` (  
  `ordernumber` varchar(50) NOT NULL default '0',  
  `code` varchar(100) NOT NULL default "",  
  `shipped_quantity` double default '0',
```

```
  `shipped_date` datetime NOT NULL default '0000-00-00  
00:00:00',  
  `orriga` tinyint(3) unsigned default NULL,  
  `SyncStatus` char(1) NOT NULL default 'A',  
  `f_txt1` int(10) unsigned default NULL,  
  `f_int1` int(10) unsigned default NULL,  
  `numdoc` int(10) unsigned NOT NULL default '0',  
  `rigadoc` int(10) unsigned NOT NULL default '0',  
  `anno` int(4) unsigned NOT NULL default '0',  
  `serie` char(2) NOT NULL default '*',  
  `tipork` char(1) NOT NULL default 'R',  
  `prezzo` double default NULL,  
  `vat` float default NULL,  
  `unit` varchar(20) default NULL,  
  `discount1` float NOT NULL default '0',  
  `discount2` float NOT NULL default '0',  
  `discount3` float NOT NULL default '0',  
  `discount4` float NOT NULL default '0',  
  `discount5` float NOT NULL default '0',  
  `total` float NOT NULL default '0',  
PRIMARY KEY (`numdoc`, `rigadoc`, `anno`, `serie`, `tipork`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1  
COMMENT='objects shipped to a customer without an order';
```

```
# Table structure for table `tbl_orders_shippings_heads`
```

```
CREATE TABLE `tbl_orders_shippings_heads` (  
  `tipork` char(1) NOT NULL default 'R',  
  `anno` smallint(6) NOT NULL default '0',  
  `serie` varchar(50) NOT NULL default '*',  
  `numdoc` varchar(50) NOT NULL default '0',  
  `datdoc` datetime default '0000-00-00 00:00:00',  
  `conto` varchar(20) default NULL,  
  `codpaga` varchar(100) default '0',  
  `courier` varchar(255) default NULL,  
  `totdoc` double default NULL,  
  `SyncStatus` char(1) NOT NULL default 'A',  
  `payc_expenses` float NOT NULL default '0',  
  `payc_discount` float default '0',  
  `payc_percexpenses` float default '0',
```

```
`totalvat` float NOT NULL default '0',
`coddest` varchar(20) default NULL,
PRIMARY KEY (`tipork`,`anno`,`serie`,`numdoc`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Table structure for table `tbl_patch_ignore`

```
CREATE TABLE `tbl_patch_ignore` (
  `id` int(11) default NULL,
  `fname` varchar(100) default NULL,
  `lname` varchar(100) default NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Table structure for table `tbl_payment_condition`

```
CREATE TABLE `tbl_payment_condition` (
  `code_payment` varchar(100) NOT NULL default '0',
  `pay_description` varchar(255) default NULL,
  `start_after_days` int(3) unsigned default NULL,
  `number_rates` int(3) unsigned default NULL,
  `days_betw_rates` int(3) unsigned default NULL,
  `discount` double default NULL,
  `expenses` double default NULL,
  `cond_vatexpenses` float default NULL,
  `perc_expenses` double default NULL,
  `cond_target` char(1) default 'S',
  `cond_notes` text,
  `cond_maxsumexpenses` double default NULL,
  `cond_creditcardid` tinyint(3) unsigned default NULL,
  `SyncStatus` char(1) NOT NULL default 'A',
  `cond_minsumorder` double default NULL,
  `cond_maxsumorder` double default NULL,
  PRIMARY KEY (`code_payment`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='payment conditions (expenses, discounts, rates, ...)';
```

Table structure for table `tbl_payment_cust`

```
CREATE TABLE `tbl_payment_cust` (
  `code_cust` varchar(20) NOT NULL default '0',
```

```
`code_payment` varchar(100) NOT NULL default '0',
`SyncStatus` char(1) NOT NULL default 'A',
`min_sumorder` double default NULL,
`max_sumorder` double default NULL,
PRIMARY KEY (`code_cust`,`code_payment`)
```

```
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='relations between customers and payment conditions';
```

Table structure for table `tbl_pricelist`

```
CREATE TABLE `tbl_pricelist` (
  `code_pricelist` varchar(100) NOT NULL default '0',
  `descr_pricelist` varchar(200) default NULL,
  `notes` varchar(255) default NULL,
  `flag1` char(2) default NULL,
  `flag2` char(2) default NULL,
  `flag3` char(2) default NULL,
  `field1` text,
  `field2` text,
  `title_pricelist` varchar(50) default NULL,
  `nrdaysdelayed` int(10) unsigned default NULL,
  `SyncStatus` char(1) NOT NULL default 'A',
  PRIMARY KEY (`code_pricelist`),
  KEY `code_pricelist` (`code_pricelist`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COMMENT='list of the available price lists';
```

Table structure for table `tbl_prices`

```
CREATE TABLE `tbl_prices` (
  `code` varchar(100) NOT NULL default '',
  `price` double NOT NULL default '0',
  `qty_from` double NOT NULL default '0',
  `qty_to` double NOT NULL default '9999999999999999',
  `pricelist` varchar(100) NOT NULL default '0',
  `code_cust` varchar(20) NOT NULL default '0',
  `date_start` date NOT NULL default '2000-01-01',
  `date_end` date NOT NULL default '2100-01-01',
  `SyncStatus` char(1) NOT NULL default 'A',
  `codlavo` varchar(10) NOT NULL default '0',
```

```
`codvalu` varchar(10) NOT NULL default '0',
`codtpro` varchar(50) NOT NULL default '0',
`unmis` varchar(10) NOT NULL default '',
PRIMARY KEY (`code`,`qty_from`,`pricelist`,`code_cust`,`date_
start`,`codlavo`,`codtpro`,`codvalu`,`unmis`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COMMENT='list
of the prices for the objects (articles)';
```

Table structure for table 'tbl_promo'

```
CREATE TABLE `tbl_promo` (
  `promo_nr` int(10) default '0',
  `promo_img` varchar(100) default NULL,
  `promo_width` int(10) default NULL,
  `promo_height` int(10) default NULL,
  `promo_link` varchar(255) default NULL,
  `flag` char(1) default NULL,
  `code` varchar(20) default NULL,
  `SyncStatus` char(1) NOT NULL default 'A',
  UNIQUE KEY `i_promo` (`promo_nr`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Table structure for table 'tbl_promotions'

```
CREATE TABLE `tbl_promotions` (
  `tb_codtpro` tinyint(2) unsigned NOT NULL default '0',
  `tb_destpro` varchar(20) default NULL,
  `tb_classe` tinyint(2) unsigned NOT NULL default '0',
  `tb_enne` float NOT NULL default '0',
  `tb_emme` float NOT NULL default '0',
  `tb_valsc0` float NOT NULL default '0',
  `tb_perc1` float NOT NULL default '0',
  `tb_perc2` float NOT NULL default '0',
  `tb_implim` float NOT NULL default '0',
  `tb_qtlim` float NOT NULL default '0',
  `tb_qtasco` float NOT NULL default '0',
  `SyncStatus` char(1) NOT NULL default 'A',
  `tb_inidata` date default NULL,
  `tb_findata` date default NULL,
  PRIMARY KEY (`tb_codtpro`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Table structure for table 'tbl_register'

```
CREATE TABLE `tbl_register` (
  `keys1` varchar(255) default NULL,
  `keys2` varchar(255) default NULL,
  `keys3` varchar(255) default NULL,
  `keys4` varchar(255) default NULL,
  `keys5` varchar(255) default NULL,
  `keys6` varchar(255) default NULL,
  `keys7` varchar(255) default NULL,
  `engines` text,
  `flag` char(1) default NULL,
  `SyncStatus` char(1) NOT NULL default 'A'
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Table structure for table 'tbl_rel_objects'

```
CREATE TABLE `tbl_rel_objects` (
  `code` varchar(100) NOT NULL default '0',
  `code_rel` varchar(10) NOT NULL default '0',
  `code_father` varchar(100) NOT NULL default '0',
  `SyncStatus` char(1) NOT NULL default 'A',
  PRIMARY KEY (`code`,`code_rel`,`code_father`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='tables that relates two objects and a relation
type';
```

Table structure for table 'tbl_relations'

```
CREATE TABLE `tbl_relations` (
  `code_rel` varchar(10) NOT NULL default '0',
  `descr_rel` varchar(255) NOT NULL default 'new relation',
  `note_rel` varchar(255) default NULL,
  `code_ing` char(2) NOT NULL default '',
  `SyncStatus` char(1) NOT NULL default 'A',
  PRIMARY KEY (`code_rel`,`code_ing`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COMMENT='list
of all available relations types';
```

Table structure for table 'tbl_shippings'

```
CREATE TABLE `tbl_shippings` (
```

```

`cust_code` varchar(20) NOT NULL default "",
`shipped_quantity` double NOT NULL default '0',
`code` varchar(100) NOT NULL default "",
`shipped_date` datetime NOT NULL default '0000-00-00
00:00:00',
`price` double NOT NULL default '0',
`SyncStatus` char(1) NOT NULL default 'A',
`code_variant` int(10) unsigned NOT NULL default '0',
`f_txt1` text,
`f_int1` int(10) unsigned default NULL,
`f_txt2` text,
PRIMARY KEY (`cust_code`,`code`,`shipped_date`,`code_
variant`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='delivered object to a customer but still not
ordered';

```

Table structure for table `tbl_site`

```

CREATE TABLE `tbl_site` (
  `l_id` int(10) NOT NULL default '0',
  `l_language` char(3) NOT NULL default "",
  `level1` int(10) default '0',
  `level2` int(10) default '0',
  `level3` int(10) default '0',
  `l_option1` varchar(220) default NULL,
  `l_option2` varchar(255) default NULL,
  `l_icon1` varchar(255) default NULL,
  `l_icon2` varchar(255) default NULL,
  `l_img1` varchar(255) default NULL,
  `l_img2` varchar(255) default NULL,
  `l_image` varchar(255) default NULL,
  `l_content` mediumtext,
  `l_url` varchar(200) default NULL,
  `l_myapp_content` varchar(255) NOT NULL default "",
  `l_form_email1` varchar(200) default NULL,
  `l_form_email2` varchar(200) default NULL,
  `l_form_email_subject` varchar(255) NOT NULL default "",
  `l_page_id` tinyint(4) default '0',
  `l_admin` varchar(20) default NULL,

```

```

`l_group` text,
`l_priority` tinyint(4) default '0',
`l_flag1` char(1) default NULL,
`l_flag2` char(1) default NULL,
`l_flag3` char(1) default NULL,
`l_flag4` char(1) default NULL,
`l_flag5` char(1) default NULL,
`l_txt1` varchar(255) default NULL,
`l_txt2` varchar(255) default NULL,
`l_txt3` varchar(255) default NULL,
`l_nr1` int(10) default '0',
`l_nr2` int(10) default '0',
`l_nr3` int(10) default '0',
`l_page_title` varchar(255) default NULL,
`l_description` text,
`l_keywords` text,
`SyncStatus` char(1) NOT NULL default 'A',
UNIQUE KEY `l_id` (`l_id`),
KEY `i_site` (`l_language`,`level1`,`level2`,`level3`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

Table structure for table `tbl_statistics`

```

CREATE TABLE `tbl_statistics` (
  `code` varchar(20) NOT NULL default "",
  `date_start_stat` datetime NOT NULL default '0000-00-00
00:00:00',
  `date_end_stat` datetime NOT NULL default '0000-00-00
00:00:00',
  `qty_stat` double NOT NULL default '0',
  `SyncStatus` char(1) NOT NULL default 'A',
PRIMARY KEY (`code`,`date_start_stat`,`date_end_stat`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

Table structure for table `tbl_stats_app`

```

CREATE TABLE `tbl_stats_app` (
  `sid` int(10) unsigned NOT NULL default '0',
  `sapp` char(2) default NULL,
  `spage` varchar(50) default NULL,
  `stime` varchar(8) default NULL,

```

```
`flag` char(1) default NULL,  
KEY `sid_k` (`sid`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Table structure for table `tbl_stats_gen`

```
CREATE TABLE `tbl_stats_gen` (  
  `sid` int(10) unsigned NOT NULL default '0',  
  `sday` varchar(10) default NULL,  
  `stime` varchar(8) default NULL,  
  `sip` varchar(127) default NULL,  
  `sreferer` varchar(127) default NULL,  
  `sos` tinyint(4) default '0',  
  `sbrowser` tinyint(4) default '0',  
  `ssession` varchar(50) default NULL,  
  `suser` varchar(20) default NULL,  
  `sw` varchar(4) default NULL,  
  `sh` varchar(4) default NULL,  
  `sc` varchar(10) default NULL,  
  PRIMARY KEY (`sid`),  
  UNIQUE KEY `sid` (`sid`),  
  KEY `sid_2` (`sid`,`suser`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Table structure for table `tbl_toplogo`

```
CREATE TABLE `tbl_toplogo` (  
  `c_lang` char(3) default NULL,  
  `c_app` varchar(50) default NULL,  
  `c_template` varchar(50) default NULL,  
  `c_logo` varchar(100) default NULL,  
  `c_logo_format` varchar(50) default NULL,  
  `c_top_text1` varchar(50) default NULL,  
  `c_top_format1` varchar(100) default NULL,  
  `c_top_text2` varchar(50) default NULL,  
  `c_top_format2` varchar(100) default NULL,  
  `c_flag1` char(1) default NULL,  
  `c_flag2` char(1) default NULL,  
  `c_flag3` char(1) default NULL,  
  `c_text1` varchar(50) default NULL,
```

```
`c_text2` varchar(50) default NULL,  
  `c_text3` varchar(50) default NULL,  
  `SyncStatus` char(1) NOT NULL default 'A',  
  KEY `i_toplogo` (`c_lang`,`c_app`,`c_template`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Table structure for table `tbl_upldnl`

```
CREATE TABLE `tbl_upldnl` (  
  `file_id` int(10) NOT NULL default '0',  
  `file_name` varchar(100) default NULL,  
  `file_date` varchar(20) default NULL,  
  `file_size` int(10) default NULL,  
  `file_attach` varchar(255) default NULL,  
  `file_owner` varchar(50) default NULL,  
  `file_group` text,  
  `file_categ` varchar(100) default NULL,  
  `file_priority` tinyint(4) default '0',  
  `file_descr1` varchar(200) default NULL,  
  `file_descr2` varchar(200) default NULL,  
  `file_descr3` varchar(200) default NULL,  
  `file_descr4` varchar(200) default NULL,  
  `file_descr5` varchar(200) default NULL,  
  `file_descr6` varchar(200) default NULL,  
  `file_descr7` varchar(200) default NULL,  
  `flag1` char(1) default NULL,  
  `flag2` char(1) default NULL,  
  `flag3` char(1) default NULL,  
  `SyncStatus` char(1) NOT NULL default 'A',  
  `file_categ1` varchar(255) default '???',  
  `file_categ2` varchar(255) default '???',  
  `file_categ3` varchar(255) default '???',  
  `file_categ4` varchar(255) default '???',  
  `file_categ5` varchar(255) default '???',  
  `file_categ6` varchar(255) default '???',  
  `file_categ7` varchar(255) default '??',  
  PRIMARY KEY (`file_id`),  
  KEY `i_upldnl` (`file_name`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
# Table structure for table `tbl_user_disclaimer`
CREATE TABLE `tbl_user_disclaimer` (
  `lang` char(2) NOT NULL default 'IT',
  `label1` varchar(255) default NULL,
  `disclaimer1` text,
  `label2` varchar(255) default NULL,
  `disclaimer2` text,
  `label3` varchar(255) default NULL,
  `disclaimer3` text,
  `flag1` char(1) default NULL,
  `flag2` char(1) default NULL,
  `flag3` char(1) default NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
# Table structure for table `tbl_users`
CREATE TABLE `tbl_users` (
  `u_id` bigint(20) NOT NULL default '0',
  `u_type` varchar(10) default 'client',
  `u_login` varchar(50) NOT NULL default "",
  `u_passwd` varchar(50) character set latin1 collate latin1_bin
default NULL,
  `u_fname` varchar(255) default NULL,
  `u_lname` varchar(255) default NULL,
  `u_email` varchar(100) default NULL,
  `u_group` text,
  `u_maillist` text,
  `u_priority` tinyint(4) default '0',
  `u_phone` varchar(30) default NULL,
  `u_fax` varchar(30) default NULL,
  `u_address1` varchar(200) default NULL,
  `u_address2` varchar(200) default NULL,
  `u_info` varchar(100) default NULL,
  `u_url` varchar(100) default NULL,
  `u_date_reg` varchar(20) default NULL,
  `u_upload` char(1) default NULL,
  `u_postfile` char(1) default NULL,
  `u_flag1` char(1) default NULL,
```

```
`u_flag2` char(1) default NULL,
  `u_flag3` char(1) default NULL,
  `u_flag4` char(1) default NULL,
  `u_flag5` char(1) default NULL,
  `u_flag6` char(1) default NULL,
  `u_flag7` char(1) default NULL,
  `u_flag_cart` char(1) default NULL,
  `pricelist` varchar(100) NOT NULL default '0',
  `code_disc_class` varchar(100) NOT NULL default '0',
  `addr_street` varchar(255) default NULL,
  `addr_number` varchar(50) default NULL,
  `addr_zip` varchar(50) default NULL,
  `addr_state` varchar(50) default NULL,
  `addr_country` varchar(50) default NULL,
  `ship_street` varchar(255) default NULL,
  `ship_number` varchar(50) default NULL,
  `ship_city` varchar(50) default NULL,
  `ship_country` varchar(50) default NULL,
  `code_lng` char(2) default NULL,
  `addr_city` varchar(50) default NULL,
  `ship_state` varchar(50) default NULL,
  `ship_zip` varchar(50) default NULL,
  `company` varchar(255) default NULL,
  `company_id` int(10) unsigned default NULL,
  `vat_no` varchar(25) default NULL,
  `u_txt1` varchar(255) default NULL,
  `u_txt2` varchar(255) default NULL,
  `u_txt3` varchar(255) default NULL,
  `u_nr1` int(11) default NULL,
  `u_nr2` int(11) default NULL,
  `u_nr3` int(11) default NULL,
  `u_department` varchar(255) default NULL,
  `u_role` varchar(255) default NULL,
  `SyncStatus` char(1) NOT NULL default 'A',
  `u_nickname` varchar(255) default NULL,
  `code_fiscale` varchar(30) default NULL,
  `u_disabled` tinyint(3) unsigned NOT NULL default '0',
  `u_alt` varchar(100) default NULL,
```

```

`u_nrint` varchar(100) default NULL,
`limit_wpoffice` int(5) unsigned default '10',
`limit_dbtools` int(6) unsigned default '10',
`u_codicegestionale` varchar(20) default NULL,
`u_externalcode` varchar(20) default NULL,
`u_vatcateg` varchar(10) default NULL,
`u_flagagw` char(1) default NULL,
`u_sync_order` char(1) default NULL,
`u_dbtools` tinyint(3) unsigned default '0',
`u_wpoffice` tinyint(3) unsigned default '0',
`u_phone_mobil` varchar(100) default NULL,
`u_codecurrency` varchar(20) default NULL,
`billing_country` varchar(50) default NULL,
`billing_city` varchar(50) default NULL,
`billing_street` varchar(255) default NULL,
`billing_state` varchar(50) default NULL,
`billing_zip` varchar(50) default NULL,
`u_agentallowchangeprice` char(1) default NULL,
PRIMARY KEY (`u_id`),
UNIQUE KEY `u_id` (`u_id`,`u_login`),
UNIQUE KEY `u_login_i` (`u_login`),
UNIQUE KEY `i_nickname` (`u_nickname`),
KEY `i_users` (`u_type`,`u_login`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='webprofessional and ebusiness users';

```

```

# Table structure for table `tbl_usersdest`
CREATE TABLE `tbl_usersdest` (
  `d_userid` varchar(20) NOT NULL default '0',
  `d_coddest` varchar(20) NOT NULL default '',
  `d_nomdest` varchar(30) default NULL,
  `d_indest` varchar(30) default NULL,
  `d_capdest` varchar(20) default NULL,
  `d_locdest` varchar(28) default NULL,
  `d_prodest` varchar(20) default NULL,
  `d_phone` varchar(30) default NULL,
  `d_zona` varchar(30) default NULL,
  `d_codfis` varchar(30) default NULL,

```

```

  `d_pariva` varchar(30) default NULL,
  `d_fax` varchar(30) default NULL,
  `d_email` varchar(30) default NULL,
  `d_country` varchar(30) default NULL,
  `SyncStatus` char(1) default 'A',
  PRIMARY KEY (`d_userid`,`d_coddest`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='different destinations';

```

```

# Table structure for table `tbl_valvari`
CREATE TABLE `tbl_valvari` (
  `codvari` varchar(4) NOT NULL default '',
  `valvari` varchar(50) NOT NULL default '',
  `desvvar` varchar(40) default NULL,
  `code_ing` char(2) NOT NULL default 'IT',
  `filevar` varchar(50) default NULL,
  `SyncStatus` char(1) NOT NULL default 'A',
  `longdesvvar` text,
  PRIMARY KEY (`codvari`,`valvari`,`code_ing`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='values associated to values';

```

```

# Table structure for table `tbl_vari`
CREATE TABLE `tbl_vari` (
  `codvari` varchar(50) NOT NULL default '',
  `desvari` varchar(30) default '0',
  `lung` smallint(2) unsigned NOT NULL default '0',
  `code_ing` char(2) NOT NULL default 'IT',
  `SyncStatus` char(1) NOT NULL default 'A',
  `longdesvari` text,
  PRIMARY KEY (`codvari`,`code_ing`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COMMENT='variant types';

```

```

# Table structure for table `tbl_varliv`
CREATE TABLE `tbl_varliv` (
  `codroot` varchar(12) NOT NULL default '0',
  `livello` smallint(2) unsigned NOT NULL default '1',
  `codvari` varchar(50) default NULL,

```

```

`SyncStatus` char(1) NOT NULL default 'A',
PRIMARY KEY (`codroot`,`livello`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

# Table structure for table `tbl_vat`
CREATE TABLE `tbl_vat` (
  `vat_value` double default NULL,
  `vat_categcode` varchar(10) NOT NULL default "",
  `vat_custcode` varchar(20) NOT NULL default "",
  `vat_prodcod` varchar(50) NOT NULL default "",
  `SyncStatus` char(1) default 'A',
  PRIMARY KEY (`vat_categcode`,`vat_custcode`,`vat_
prodcod`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

# Table structure for table `tbl_vatcateg`
CREATE TABLE `tbl_vatcateg` (
  `vatc_code` varchar(10) NOT NULL default "",
  `vatc_title` varchar(255) default NULL,
  `vatc_note` text,
  `SyncStatus` char(1) default 'A',
  PRIMARY KEY (`vatc_code`),
  KEY `vatc_code` (`vatc_code`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

# Table structure for table `tbl_visitors`
CREATE TABLE `tbl_visitors` (
  `u_id` int(10) NOT NULL default '0',
  `u_login` varchar(20) default NULL,
  `u_email` varchar(150) default NULL,
  `u_maillist` text,
  `u_date_reg` varchar(20) default NULL,
  `u_host_reg` varchar(20) default NULL,
  `u_flag1` char(1) default NULL,
  `u_flag2` char(1) default NULL,
  `u_flag3` char(1) default NULL,
  `SyncStatus` char(1) NOT NULL default 'A',
  UNIQUE KEY `u_id` (`u_id`),
  KEY `i_visitors` (`u_login`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

# Table structure for table `tbl_visitors_lists`
CREATE TABLE `tbl_visitors_lists` (
  `u_id` int(10) unsigned NOT NULL default '0',
  `m_id` int(10) unsigned NOT NULL default '0',
  `flag1` char(1) default '0',
  `flag2` char(1) default '0',
  `flag3` char(1) default '0',
  PRIMARY KEY (`m_id`,`u_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

Tabella dei codici lingua utilizzati in WebProfessional

Tabella dei codici lingua utilizzati nella piattaforma WebProfessional aderenti allo standard ISO 639-1.

Language	ISO	Language	ISO	Language	ISO
Abkhazian	AB	Frisian	FY	Macedonian	MK
Afar	AA	Galician	GL	Malagasy	MG
Afrikaans	AF	Georgian	KA	Malay	MS
Albanian	SQ	German	DE	Malayalam	ML
Amharic	AM	Greek	EL	Maltese	MT
Arabic	AR	Greenlandic	KL	Maori	MI
Armenian	HY	Guarani	GN	Marathi	MR
Assamese	AS	Gujarati	GU	Moldavian	MO
Aymara	AY	Hausa	HA	Mongolian	MN
Azerbaijani	AZ	Hebrew (former iw)	HE	Nauru	NA
Bashkir	BA	Hindi	HI	Nepali	NE
Basque	EU	Hungarian	HU	Norwegian	NO
Bengali	BN	Icelandic	IS	Occitan	OC
Bhutani	DZ	Indonesian (former in)	ID	Oriya	OR
Bihari	BH	Interlingua	IA	Oromo (Afan)	OM
Bislama	BI	Interlingue	IE	Pashto, Pushto	PS
Breton	BR	Inuktitut (Eskimo)	IU	Persian	FA
Bulgarian	BG	Inupiak	IK	Polish	PL
Burmese	MY	Irish	GA	Portuguese	PT
Byelorussian	BE	Italian	IT	Punjabi	PA
Cambodian	KM	Japanese	JA	Quechua	QU
Catalan	CA	Javanese	JW	Rhaeto-Romance	RM
Chinese	ZH	Kannada	KN	Romanian	RO
Corsican	CO	Kashmiri	KS	Russian	RU
Croatian	HR	Kazakh	KK	Samoan	SM
Czech	CS	Kinyarwanda	RW	Sangro	SG
Danish	DA	Kirghiz	KY	Sanskrit	SA
Dutch	NL	Kirundi	RN	Scots Gaelic	GD
English	EN	Korean	KO	Serbian	SR
Esperanto	EO	Kurdish	KU	Serbo-Croatian	SH
Estonian	ET	Laothian	LO	Sesotho	ST
Faeroese	FO	Latin	LA	Setswana	TN
Fiji	FJ	Latvian, Lettish	LV	Shona	SN
Finnish	FI	Lingala	LN	Sindhi	SD
French	FR	Lithuanian	LT	Singhalese	SI

Language	ISO
Siswati	SS
Slovak	SK
Slovenian	SL
Somali	SO
Spanish	ES
Sudanese	SU
Swahili	SW
Swedish	SV
Tagalog	TL
Tajik	TG
Tamil	TA
Tatar	TT
Tegulu	TE
Thai	TH
Tibetan	BO
Tigrinya	TI
Tonga	TO
Tsonga	TS
Turkish	TR
Turkmen	TK
Twi	TW
Uigur	UG
Ukrainian	UK
Urdu	UR
Uzbek	UZ
Vietnamese	VI
Volapuk	VO
Welch	CY
Wolof	WO
Xhosa	XH
Yiddish (former ji)	YI
Yoruba	YO
Zhuang	ZA
Zulu	ZU

Relazioni nel database WebProfessional

Elenco delle relazioni e loro proprietà presenti nel database di WebProfessional. Le relazioni sono implementate solo a livello logico, in quanto il database fisico MySQL non supporta questo tipo di opzione. Le relazioni sono quindi implementate tramite funzioni PHP.

Elenco delle relazioni

Parent Entity	Child Entity	Relationship Type	No of Keys
tbl_objects	tbl_rel_objects	Identifying	1
tbl_relations	tbl_rel_objects	Identifying	2
tbl_rel_objects	tbl_rel_objects	Non-Identifying	3
tbl_pricelist	tbl_prices	Non-Identifying	1
tbl_objects	tbl_prices	Identifying	1
tbl_pricelist	tbl_users	Non-Identifying	1
tbl_currency	tbl_prices	Non-Identifying	1
tbl_categories	tbl_categories	Non-Identifying	2
tbl_users	tbl_group	Non-Specific	0
tbl_stats_gen	tbl_stats_app	Identifying	1
tbl_users	tbl_stats_gen	Non-Identifying	1
tbl_objects	tbl_artroot	Identifying	1
tbl_users	tbl_users_group	Identifying	1
tbl_artroot	tbl_artvar	Identifying	2
tbl_site	tbl_stats_app	Non-Identifying	1
tbl_group	tbl_users_group	Identifying	1
tbl_maillist	tbl_visitors_lists	Identifying	1
tbl_visitors	tbl_visitors_lists	Identifying	1
tbl_site	tbl_users	Non-Specific	0
tbl_maillist	tbl_mails	Non-Identifying	1
tbl_users	tbl_site	Non-Identifying	1
tbl_news_categ	tbl_news	Non-Identifying	1
tbl_news	tbl_users	Non-Specific	0
tbl_group	tbl_upldnl	Non-Specific	0
tbl_objects	tbl_ec_measureunit_conversion	Non-Identifying	1
tbl_objects	tbl_objects_lng	Identifying	1
tbl_objects_lng	tbl_objects_fields	Identifying	2
tbl_objects	tbl_cat_objects	Identifying	1
tbl_cat_type	tbl_categories	Non-Identifying	2
tbl_categories	tbl_cat_objects	Identifying	3
tbl_ec_measureunit	tbl_prices	Non-Identifying	2
tbl_ec_measureunit	tbl_ec_measureunit_conversion	Non-Identifying	2
tbl_vatcateg	tbl_vat	Non-Identifying	1
tbl_objects	tbl_vat	Non-Identifying	1
tbl_users	tbl_vat	Non-Identifying	1

Dettagli delle relazioni

Relationship: tbl_group to tbl_users_group

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_group
Child Entity	tbl_users_group
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
code_group

Child Key
code_group

Relationship: tbl_ec_measureunit to tbl_ec_measureunit_conversion

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_ec_measureunit
Child Entity	tbl_ec_measureunit_conversion
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
um_code
code_lng

Child Key
umc_code
code_lng

Relationship: tbl_vatcateg to tbl_vat

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_vatcateg
Child Entity	tbl_vat
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
vatc_code

Child Key
vat_categcode

Relationship: tbl_objects to tbl_vat

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_objects
Child Entity	tbl_vat
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
code

Child Key
vat_prodcod

Relationship: tbl_objects to tbl_objects_Ing

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_objects
Child Entity	tbl_objects_Ing
Cardinality	One to One or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
code

Child Key
code

Relationship: tbl_users to tbl_vat

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_users

Child Entity	tbl_vat
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key	Child Key
u_id	vat_custcode

Relationship: tbl_objects_ing to tbl_objects_fields

Relationship Name	Identifying
Relationship Type	tbl_objects_ing
Parent Entity	tbl_objects_fields
Child Entity	tbl_objects_fields
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key	Child Key
code	code
code_ing	code_ing

Relationship: tbl_currency to tbl_currency_exchange

Relationship Name	Identifying
Relationship Type	tbl_currency
Parent Entity	tbl_currency_exchange
Child Entity	tbl_currency_exchange
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence A tbl_currency has exchange rate tbl_currency_exchange(s).
Inverse Verb A tbl_currency_exchange has description tbl_currency.

Sentence
Definition

Relationship Keys

Parent Key
tc_codecurrency

Child Key
tce_codecurrency

Relationship: tbl_users to tbl_currency_exchange

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_users
Child Entity	tbl_currency_exchange
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	external_code

Relationship Description

Verb Sentence	A tbl_users has customer exchange rate tbl_currency_exchange(s).
Inverse Verb	A tbl_currency_exchange belongs to customer tbl_users.
Sentence Definition	

Relationship Keys

Parent Key
u_externalcode

Child Key
tce_codecust

Relationship: tbl_objects to tbl_cat_objects

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_objects
Child Entity	tbl_cat_objects
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence	
Inverse Verb	
Sentence Definition	

Relationship Keys

Parent Key
code

Child Key
code

Relationship: tbl_cat_type to tbl_categories

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_cat_type
Child Entity	tbl_categories
Cardinality	Zero or One to Zero or More
Existence	Optional
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
type_cat
code_lng

Child Key
type_cat
code_lng

Relationship: tbl_users to tbl_currency_exchange

Relationship Name	
Relationship Type	Non-Specific
Parent Entity	tbl_users
Child Entity	tbl_currency_exchange
Cardinality	
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence	A tbl_users has exchange rate tbl_currency_exchange(s).
Inverse Verb	A tbl_currency_exchange belongs to customers tbl_users.

Sentence
Definition

Relationship Keys

Parent Key

Child Key

Relationship: tbl_vari to tbl_valvari

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_vari
Child Entity	tbl_valvari
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
codvari
code_lng

Child Key
codvari
code_lng

Relationship: tbl_vari to tbl_varliv

Relationship Name	
Relationship Type	Non-Identifying

Parent Entity	tbl_vari
Child Entity	tbl_varliy
Cardinality	One to Exactly: 1
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key	Child Key
codvari	codvari
code_ing	code_ing

Relationship: tbl_categories to tbl_cat_objects

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_categories
Child Entity	tbl_cat_objects
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key	Child Key
code_cat	code_cat
code_father	code_father
code_ing	code_ing

Relationship: tbl_maillist to tbl_visitors_lists

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_maillist
Child Entity	tbl_visitors_lists
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
m_id

Child Key
m_id

Relationship: tbl_relations to tbl_rel_objects

Relationship Name	Identifying
Relationship Type	tbl_relations
Parent Entity	tbl_rel_objects
Child Entity	tbl_rel_objects
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
code_rel
code_lng

Child Key
code_rel
code_lng

Relationship: tbl_visitors to tbl_visitors_lists

Relationship Name	Identifying
Relationship Type	tbl_visitors
Parent Entity	tbl_visitors_lists
Child Entity	tbl_visitors_lists
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
u_id

Child Key
u_id

Relationship: tbl_objects to tbl_rel_objects

Relationship Name	Identifying
Relationship Type	tbl_objects
Parent Entity	tbl_rel_objects
Child Entity	tbl_rel_objects
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
code

Child Key
code

Relationship: tbl_users to tbl_group

Relationship Name	
Relationship Type	Non-Specific
Parent Entity	tbl_users
Child Entity	tbl_group
Cardinality	
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key

Child Key

Relationship: tbl_artvar to tbl_valvari

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_artvar
Child Entity	tbl_valvari
Cardinality	Zero or One to Zero or More
Existence	Optional
Propagated Key	varval_codes

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
codvar

Child Key
codvar

Relationship: tbl_rel_objects to tbl_rel_objects

Relationship Name

Relationship Type	Non-Identifying
Parent Entity	tbl_rel_objects
Child Entity	tbl_rel_objects
Cardinality	One to One or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence Definition

Relationship Keys

Parent Key	Child Key
code_rel	code_rel
code_father	code
code_ing	code_ing

Relationship: tbl_maillist to tbl_mails

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_maillist
Child Entity	tbl_mails
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence A tbl_maillist has emails tbl_mails(s).
Inverse Verb A tbl_mails belongs to tbl_maillist.

Sentence Definition

Relationship Keys

Parent Key	Child Key
m_id	mail_to

Relationship: tbl_categories to tbl_categories

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_categories
Child Entity	tbl_categories
Cardinality	One to One or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence Definition

Relationship Keys

Parent Key
code_ing
code_father

Child Key
code_ing
code_cat

Relationship: tbl_objects to tbl_prices

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_objects
Child Entity	tbl_prices
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
code

Child Key
code

Relationship: tbl_prices to tbl_ec_measureunit_conversion

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_prices
Child Entity	tbl_ec_measureunit_conversion
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
code
unmis
qty_from
code_cust
date_start
codlavo
codtpro
pricelist
codvalu

Child Key
umc_prodcode
umc_code
qty_from
code_cust
date_start
codlavo
codtpro
pricelist
codvalu

Relationship: tbl_stats_gen to tbl_stats_app

Relationship Name

Relationship Type	Identifying
Parent Entity	tbl_stats_gen
Child Entity	tbl_stats_app
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence	A tbl_stats_gen has details tbl_stats_app(s).
Inverse Verb	A tbl_stats_app has session data tbl_stats_gen.

Sentence Definition

Relationship Keys

Parent Key	Child Key
sid	sid

Relationship: tbl_users to tbl_stats_gen

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_users
Child Entity	tbl_stats_gen
Cardinality	Zero or One to Zero or More
Existence	Optional
Propagated Key	Primary Key

Relationship Description

Verb Sentence	A tbl_users has sessions tbl_stats_gen(s).
Inverse Verb	A tbl_stats_gen has user info tbl_users.

Sentence Definition

Relationship Keys

Parent Key	Child Key
u_id	suser

Relationship: tbl_site to tbl_stats_app

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_site
Child Entity	tbl_stats_app
Cardinality	Zero or One to Zero or More
Existence	Optional
Propagated Key	Primary Key

Relationship Description

Verb Sentence	A tbl_site has been accessed tbl_stats_app(s).
Inverse Verb	A tbl_stats_app has content details tbl_site.

Sentence Definition

Relationship Keys

Parent Key
l_id

Child Key
spage

Relationship: tbl_group to tbl_upldnl

Relationship Name	
Relationship Type	Non-Specific
Parent Entity	tbl_group
Child Entity	tbl_upldnl
Cardinality	
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key

Child Key

Relationship: tbl_objects to tbl_artroot

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_objects
Child Entity	tbl_artroot
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
code

Child Key
codroot

Relationship: tbl_artroot to tbl_artvar

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_artroot
Child Entity	tbl_artvar
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence

Inverse Verb

Sentence Definition

Relationship Keys

Parent Key
codroot
code_ing

Child Key
codroot
code_ing

Relationship: tbl_site to tbl_users

Relationship Name	
Relationship Type	Non-Specific
Parent Entity	tbl_site
Child Entity	tbl_users
Cardinality	
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence	A tbl_site has permission for group tbl_users(s).
Inverse Verb	A tbl_users user group can see page tbl_site.

Sentence Definition

Relationship Keys

Parent Key

Child Key

Relationship: tbl_users to tbl_site

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_users
Child Entity	tbl_site
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence	A tbl_users is administrator of tbl_site(s).
Inverse Verb	A tbl_site is administered by tbl_users.

Sentence Definition

Relationship Keys

Parent Key
u_id

Child Key
l_admin

Relationship: tbl_pricelist to tbl_prices

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_pricelist
Child Entity	tbl_prices

Cardinality Zero or One to Zero or More
Existence Optional
Propagated Key Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key code_pricelist **Child Key** pricelist

Relationship: tbl_pricelist to tbl_users

Relationship Name
Relationship Type Non-Identifying
Parent Entity tbl_pricelist
Child Entity tbl_users
Cardinality Zero or One to Zero or More
Existence Optional
Propagated Key Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key code_pricelist **Child Key** pricelist

Relationship: tbl_currency to tbl_prices

Relationship Name
Relationship Type Non-Identifying
Parent Entity tbl_currency
Child Entity tbl_prices
Cardinality Zero or One to Zero or More
Existence Optional
Propagated Key Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key tc_codecurrency **Child Key** codvalu

Relationship: tbl_users to tbl_users_group

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_users
Child Entity	tbl_users_group
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key	Child Key
u_id	u_id

Relationship: tbl_news_categ to tbl_news

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_news_categ
Child Entity	tbl_news
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence	A tbl_news_categ has news tbl_news(s).
Inverse Verb	A tbl_news belongs to category tbl_news_categ.

Sentence
Definition

Relationship Keys

Parent Key	Child Key
cod_categ	ord_categ

Relationship: tbl_news to tbl_users

Relationship Name	
Relationship Type	Non-Specific
Parent Entity	tbl_news
Child Entity	tbl_users
Cardinality	
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence	A tbl_news can see news tbl_users(s).
---------------	---------------------------------------

Inverse Verb A tbl_users can be accessed by tbl_news.

Sentence Definition

Relationship Keys

Parent Key

Child Key

Relationship: tbl_objects to tbl_objects_Ing

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_objects
Child Entity	tbl_objects_Ing
Cardinality	One to One or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence Definition

Relationship Keys

Parent Key
code

Child Key
code

Relationship: tbl_objects_Ing to tbl_objects_fields

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_objects_Ing
Child Entity	tbl_objects_fields
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence Definition

Relationship Keys

Parent Key
code
code_Ing

Child Key
code
code_Ing

Relationship: tbl_objects to tbl_cat_objects

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_objects
Child Entity	tbl_cat_objects

Cardinality One To Zero or More
Existence Mandatory
Propagated Key Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key code **Child Key** code

Relationship: tbl_cat_type to tbl_categories

Relationship Name
Relationship Type Non-Identifying
Parent Entity tbl_cat_type
Child Entity tbl_categories
Cardinality Zero or One to Zero or More
Existence Optional
Propagated Key Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key type_cat
code_lng **Child Key** type_cat
code_lng

Relationship: tbl_categories to tbl_cat_objects

Relationship Name
Relationship Type Identifying
Parent Entity tbl_categories
Child Entity tbl_cat_objects
Cardinality One To Zero or More
Existence Mandatory
Propagated Key Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key **Child Key**

code_cat
code_father
code_ing

code_cat
code_father
code_ing

Relationship: tbl_relations to tbl_rel_objects

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_relations
Child Entity	tbl_rel_objects
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
code_rel
code_ing

Child Key
code_rel
code_ing

Relationship: tbl_objects to tbl_rel_objects

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_objects
Child Entity	tbl_rel_objects
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
code

Child Key
code

Relationship: tbl_rel_objects to tbl_rel_objects

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_rel_objects
Child Entity	tbl_rel_objects
Cardinality	One to One or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
code_rel
code_father
code_ing

Child Key
code_rel
code_ing

Relationship: tbl_categories to tbl_categories

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_categories
Child Entity	tbl_categories
Cardinality	One to One or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
code_ing
code_father

Child Key
code_ing
code_cat

Relationship: tbl_currency to tbl_currency_exchange

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_currency
Child Entity	tbl_currency_exchange
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence A tbl_currency has exchange rate tbl_currency_exchange(s).
Inverse Verb A tbl_currency_exchange has description tbl_currency.

Sentence
Definition

Relationship Keys

Parent Key
tc_codecurrency

Child Key
tce_codecurrency

Relationship: tbl_users to tbl_currency_exchange

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_users
Child Entity	tbl_currency_exchange
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	external_code

Relationship Description

Verb Sentence	A tbl_users has customer exchange rate tbl_currency_exchange(s).
Inverse Verb	A tbl_currency_exchange belongs to customer tbl_users.
Sentence Definition	

Relationship Keys

Parent Key	Child Key
u_externalcode	tce_codecust

Relationship: tbl_users to tbl_currency_exchange

Relationship Name	
Relationship Type	Non-Specific
Parent Entity	tbl_users
Child Entity	tbl_currency_exchange
Cardinality	
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence	A tbl_users has exchange rate tbl_currency_exchange(s).
Inverse Verb	A tbl_currency_exchange belongs to customers tbl_users.
Sentence Definition	

Relationship Keys

Parent Key	Child Key
------------	-----------

Relationship: tbl_maillist to tbl_visitors_lists

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_maillist
Child Entity	tbl_visitors_lists
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence	
---------------	--

Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
m_id

Child Key
m_id

Relationship: tbl_visitors to tbl_visitors_lists

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_visitors
Child Entity	tbl_visitors_lists
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
u_id

Child Key
u_id

Relationship: tbl_maillist to tbl_mails

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_maillist
Child Entity	tbl_mails
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

A tbl_maillist has emails tbl_mails(s).
A tbl_mails belongs to tbl_maillist.

Sentence
Definition

Relationship Keys

Parent Key
m_id

Child Key
mail_to

Relationship: tbl_ec_measureunit to tbl_ec_measureunit_conversion

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_ec_measureunit
Child Entity	tbl_ec_measureunit_conversion

Cardinality One To Zero or More
Existence Mandatory
Propagated Key Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key um_code code_lng	Child Key umc_code code_lng
--	--

Relationship: tbl_objects to tbl_prices

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_objects
Child Entity	tbl_prices
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key code	Child Key code
---------------------------	--------------------------

Relationship: tbl_prices to tbl_ec_measureunit_conversion

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_prices
Child Entity	tbl_ec_measureunit_conversion
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key	Child Key
-------------------	------------------

code	umc_prodcod
unmis	umc_code
qty_from	qty_from
code_cust	code_cust
date_start	date_start
codlavo	codlavo
codtpro	codtpro
pricelist	pricelist
codvalu	codvalu

Relationship: tbl_news_categ to tbl_news

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_news_categ
Child Entity	tbl_news
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence	A tbl_news_categ has news tbl_news(s).
Inverse Verb	A tbl_news belongs to category tbl_news_categ.

Sentence Definition

Relationship Keys

Parent Key	Child Key
cod_categ	ord_categ

Relationship: tbl_news to tbl_users

Relationship Name	
Relationship Type	Non-Specific
Parent Entity	tbl_news
Child Entity	tbl_users
Cardinality	
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence	A tbl_news can see news tbl_users(s).
Inverse Verb	A tbl_users can be accessed by tbl_news.

Sentence Definition

Relationship Keys

Parent Key	Child Key
-------------------	------------------

Relationship: tbl_objects to tbl_prices

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_objects
Child Entity	tbl_prices
Cardinality	One To Zero or More
Existence	Mandatory

Propagated Key Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key **Child Key**
code code

Relationship: tbl_prices to tbl_ec_measureunit_conversion

Relationship Name
Relationship Type Non-Identifying
Parent Entity tbl_prices
Child Entity tbl_ec_measureunit_conversion
Cardinality One To Zero or More
Existence Mandatory
Propagated Key Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key **Child Key**
code umc_prodcode
unmis umc_code
qty_from qty_from
code_cust code_cust
date_start date_start
codlavo codlavo
codtpro codtpro
pricelist pricelist
codvalu codvalu

Relationship: tbl_pricelist to tbl_prices

Relationship Name
Relationship Type Non-Identifying
Parent Entity tbl_pricelist
Child Entity tbl_prices
Cardinality Zero or One to Zero or More
Existence Optional
Propagated Key Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
code_pricelist

Child Key
pricelist

Relationship: tbl_pricelist to tbl_users

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_pricelist
Child Entity	tbl_users
Cardinality	Zero or One to Zero or More
Existence	Optional
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
code_pricelist

Child Key
pricelist

Relationship: tbl_currency to tbl_prices

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_currency
Child Entity	tbl_prices
Cardinality	Zero or One to Zero or More
Existence	Optional
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
tc_codecurrency

Child Key
codvalu

Relationship: tbl_site to tbl_users

Relationship Name	
Relationship Type	Non-Specific
Parent Entity	tbl_site
Child Entity	tbl_users
Cardinality	
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence A tbl_site has permission for group tbl_users(s).
Inverse Verb A tbl_users user group can see page tbl_site.
Sentence Definition

Relationship Keys

Parent Key **Child Key**

Relationship: tbl_users to tbl_site

Relationship Name
Relationship Type Non-Identifying
Parent Entity tbl_users
Child Entity tbl_site
Cardinality One To Zero or More
Existence Mandatory
Propagated Key Primary Key

Relationship Description

Verb Sentence A tbl_users is administrator of tbl_site(s).
Inverse Verb A tbl_site is administered by tbl_users.
Sentence Definition

Relationship Keys

Parent Key **Child Key**
u_id l_admin

Relationship: tbl_relations to tbl_rel_objects

Relationship Name
Relationship Type Identifying
Parent Entity tbl_relations
Child Entity tbl_rel_objects
Cardinality One To Zero or More
Existence Mandatory
Propagated Key Primary Key

Relationship Description

Verb Sentence
Inverse Verb
Sentence Definition

Relationship Keys

Parent Key **Child Key**
code_rel code_rel
code_ing code_ing

Relationship: tbl_objects to tbl_rel_objects

Relationship Name
Relationship Type Identifying
Parent Entity tbl_objects

Child Entity	tbl_rel_objects
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key	Child Key
code	code

Relationship: tbl_rel_objects to tbl_rel_objects

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_rel_objects
Child Entity	tbl_rel_objects
Cardinality	One to One or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key	Child Key
code_rel	code_rel
code_father	code
code_lng	code_lng

Relationship: tbl_stats_gen to tbl_stats_app

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_stats_gen
Child Entity	tbl_stats_app
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence A tbl_stats_gen has details tbl_stats_app(s).
Inverse Verb A tbl_stats_app has session data tbl_stats_gen.

Sentence
Definition

Relationship Keys

Parent Key
sid

Child Key
sid

Relationship: tbl_users to tbl_stats_gen

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_users
Child Entity	tbl_stats_gen
Cardinality	Zero or One to Zero or More
Existence	Optional
Propagated Key	Primary Key

Relationship Description

Verb Sentence A tbl_users has sessions tbl_stats_gen(s).
Inverse Verb A tbl_stats_gen has user info tbl_users.

Sentence Definition

Relationship Keys

Parent Key
u_id

Child Key
suser

Relationship: tbl_site to tbl_stats_app

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_site
Child Entity	tbl_stats_app
Cardinality	Zero or One to Zero or More
Existence	Optional
Propagated Key	Primary Key

Relationship Description

Verb Sentence A tbl_site has been accessed tbl_stats_app(s).
Inverse Verb A tbl_stats_app has content details tbl_site.

Sentence Definition

Relationship Keys

Parent Key
l_id

Child Key
spage

Relationship: tbl_users to tbl_group

Relationship Name	
Relationship Type	Non-Specific
Parent Entity	tbl_users
Child Entity	tbl_group
Cardinality	
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key

Child Key

Relationship: tbl_group to tbl_upldnl

Relationship Name	
Relationship Type	Non-Specific
Parent Entity	tbl_group
Child Entity	tbl_upldnl
Cardinality	
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key

Child Key

Relationship: tbl_group to tbl_users_group

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_group
Child Entity	tbl_users_group
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
code_group

Child Key
code_group

Relationship: tbl_users to tbl_group

Relationship Name	
Relationship Type	Non-Specific
Parent Entity	tbl_users
Child Entity	tbl_group
Cardinality	

Relationship: tbl_vari to tbl_varliv

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_vari
Child Entity	tbl_varliv
Cardinality	One to Exactly: 1
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence Definition

Relationship Keys

Parent Key	Child Key
codvari	codvari
code_lng	code_lng

Relationship: tbl_artvar to tbl_valvari

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_artvar
Child Entity	tbl_valvari
Cardinality	Zero or One to Zero or More
Existence	Optional
Propagated Key	varval_codes

Relationship Description

Verb Sentence
Inverse Verb

Sentence Definition

Relationship Keys

Parent Key	Child Key
codvar	codvar

Relationship: tbl_objects to tbl_artroot

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_objects
Child Entity	tbl_artroot
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence

Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
code

Child Key
codroot

Relationship: tbl_artroot to tbl_artvar

Relationship Name	
Relationship Type	Identifying
Parent Entity	tbl_artroot
Child Entity	tbl_artvar
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
codroot
code_ing

Child Key
codroot
code_ing

Relationship: tbl_vatcateg to tbl_vat

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_vatcateg
Child Entity	tbl_vat
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key
vatc_code

Child Key
vat_categcode

Relationship: tbl_objects to tbl_vat

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_objects

Child Entity	tbl_vat
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key	Child Key
code	vat_prodcod

Relationship: tbl_users to tbl_vat

Relationship Name	
Relationship Type	Non-Identifying
Parent Entity	tbl_users
Child Entity	tbl_vat
Cardinality	One To Zero or More
Existence	Mandatory
Propagated Key	Primary Key

Relationship Description

Verb Sentence
Inverse Verb

Sentence
Definition

Relationship Keys

Parent Key	Child Key
u_id	vat_custcode

Bibliografia

Content Management Systems (www.contentmanager.eu.com)

A European resource for Content Management Systems and Solutions - enterprise content, web content, document management, personalisation, portals, and more.

Web-Database Connectivity: Middleware

By Jian-Qing Wu.

Practical PHP Programming (www.hudzilla.org)

Chapter "Cross-platform PHP" by Paul Hudson.

PHP Manual (www.php.net)

PostgreSQL vs. MySQL vs. Commercial Databases (www.devx.com)

Can you trust the leading open-source database engines, PostgreSQL and MySQL, to deliver the performance and features that the Oracles, SQL Servers, and DB2s of the world do? Not just yet, but they could offer enough to meet your needs. Find out how they stack up against each other, as well as against the commercial alternatives. Article by Tim Conrad.

MySQL Manual and Documentation (www.mysql.com)

PHP, ASP, ColdFusion, what's your fav? (www.daniweb.com)

Forum post by "incissor" on www.daniweb.com.

PHP, ASP or ASP.NET? (www.builder.au.com.au)

Local tech commentator Daniel Winter looks beyond the hype and FUD slinging to show you where and when to use PHP, ASP and ASP.NET for your web apps.