

PROVA SCRITTA DI TECNOLOGIA DATABASE – 02/12/2004

Corso di Laurea Specialistica in Ingegneria Informatica - NOD

PROF. SONIA BERGAMASCHI

Esercizio 1 (punti 20)

Dato il seguente schema relazionale:

FOTOGRAFO (COD_F, NOME, NAZIONE)

PERSONA (CF, NOME, PRIVACY)

FOTO (COD_FOTO, DESCRIZIONE, NAZIONE, COSTO, COD_F)

FK: COD_F **REFERENCES** FOTOGRAFO

CONTIENE (CF, COD_FOTO)

FK: CF **REFERENCES** PERSONA

FK: COD_FOTO **REFERENCES** FOTO

Dove l'attributo PRIVACY vale 1 se la persona ha firmato la liberatoria ai sensi della legge 675/95, 0 altrimenti.

Una foto può essere pubblicata solo se tutte le persone immortalate hanno firmato la liberatoria.

Scrivere in SQL le seguenti interrogazioni

- 1) Selezionare la foto pubblicabile che contiene il maggior numero di persone;
- 2) Selezionare, per ciascuna nazione, il numero medio di foto pubblicabili scattate da fotografi della nazione data.

Scrivere in embedded SQL la seguente interrogazione:

- 3) Mostrare, per ciascuna persona, il compenso totale calcolato nell'ipotesi che il costo di una singola fotografia sia ottenuto come somma dei compensi (identici) ottenuti dalle persone contenute nella foto.

Esercizio 2 (punti 7)

Dato il seguente schema relazionale:

R(A,B,C)

e considerando le seguenti dipendenze funzionali:

(FD1) AC → B

(FD2) B → A

Viene richiesto di:

- Determinare la chiave o le chiavi dello schema di relazione;
- Determinare se lo schema di relazione è in 2NF, 3NF e BCNF;
- Data la decomposizione binaria

R1(A,C)

R2(A,B)

Scrivere il trigger che preservi la dipendenza funzionale FD1 in caso di inserimento ed aggiornamento dati.

Esercizio 3 (punti 6)

Si completi lo schema database con le primary e foreign key, e si scrivano i trigger che preservano le dipendenze funzionali elencate nel seguito:

Contiene	
CodP	CodT
001	1
002	2
002	4
005	4
004	1

PacchettoTuristico	
CodT	CodCli
1	4
2	8
3	7
4	8

Prodotti		
CodP	Nome	Tipo
001	Albergo HB	Tipo1
002	Albergo FB	Tipo1
003	Bar	Tipo2
004	Albergo P	Tipo1
005	Piscina	Tipo3

Dipendenze funzionali:

1. Il codice del Pacchetto Turistico (*CodT*) determina il codice cliente (*CodCli*).
2. Il codice prodotto (*CodP*) determina il nome (*Nome*) e il tipo (*Tipo*).
3. Contiene definisce i prodotti di un Pacchetto Turistico.
4. Un Pacchetto Turistico non può contenere più volte lo stesso prodotto.
5. Un Pacchetto Turistico può contenere al più tre prodotti dello stesso tipo.

Soluzione Esercizio 1

FOTOGRAFO (COD_F, NOME, NAZIONE)
PERSONA (CF, NOME, PRIVACY)
FOTO (COD_FOTO, DESCRIZIONE, NAZIONE, COSTO, COD_F)
FK: COD_F **REFERENCES** FOTOGRAFO
CONTIENE (CF, COD_FOTO)
FK: CF **REFERENCES** PERSONA
FK: COD_FOTO **REFERENCES** FOTO

1) Selezionare la foto pubblicabile che contiene il maggior numero di persone;

```
SELECT      COD_FOTO
FROM        FOTO F, CONTIENE C, PERSONA P
WHERE       F.COD_FOTO = C.COD_FOTO
AND         P.CF = C.CF
AND         P.PRIVACY = 1
AND         F.COD_FOTO NOT IN (SELECT C1.COD_FOTO
                                FROM   CONTIENE C1, PERSONA P1
                                WHERE  C1.CF = P1.CF
                                AND    P1.PRIVACY = 0)

GROUP BY   COD_FOTO
HAVING     COUNT(*) >= ALL (SELECT COUNT(*)
                            FROM   FOTO F2, CONTIENE C2, PERSONA P2
                            WHERE  F2.COD_FOTO = C2.COD_FOTO
                            AND    P2.CF = C2.CF
                            AND    P2.PRIVACY = 1
                            AND    F2.COD_FOTO NOT IN
                            (SELECT C3.COD_FOTO
                             FROM   CONTIENE C3, PERSONA P3
                             WHERE  C3.CF = P3.CF
                             AND    P3.PRIVACY = 0))
```

- 2) Selezionare, per ciascuna nazione, il numero medio di foto pubblicabili scattate da fotografi della nazione data.

```
CREATE VIEW V1 AS
SELECT    F.CODF, COUNT(*) AS NFOTO
FROM      FOTO F, FOTOGRAFO F1
WHERE     F.COD_FOTO    IN (SELECT C1.COD_FOTO
                             FROM   CONTIENE C1, PERSONA P1
                             WHERE  C1.CF = P1.CF
                             AND    P1.PRIVACY = 1)
AND       F.COD_FOTO NOT IN (SELECT C2.COD_FOTO
                             FROM   CONTIENE C2, PERSONA P2
                             WHERE  C2.CF = P2.CF
                             AND    P2.PRIVACY = 0)
AND       F.CODF = F.CODF
AND       F.NAZIONE = F1.NAZIONE
GROUP BY  F.CODF

SELECT    NAZIONE, AVG(NFOTO)
FROM      V1, FOTOGRAFO F
WHERE     F.CODF = V1.CODF
GROUP BY  NAZIONE
```

- 3) Mostrare, per ciascuna persona, il compenso totale calcolato nell'ipotesi che il costo di una singola fotografia sia ottenuto come somma dei compensi (identici) ottenuti dalle persone contenute nella foto.

```
Q1:  SELECT    CF, COD_FOTO
      FROM      CONTIENE
      ORDER BY  CF, COD_FOTO

Q2:  SELECT    COD_FOTO, COSTO, COUNT(*) AS NUM_PERS
      FROM      FOTO F, CONTIENE C
```

```
WHERE      F.COD_FOTO = :COD_FOTO
AND        F.COD_FOTO = C.COD_FOTO
GROUP BY   COD_FOTO, COSTO
```

Declare Cursor "C1" For Q1

Declare Cursor "C2" For Q2

open C1;

fetch C1 into :CF, :COD_FOTO;

while (SQLCODE == 0){

CF_ATTUALE = CF;

COMPENSO = 0;

while (CF_ATTUALE == CF && SQLCODE == 0){

open C2;

fetch C2 into :COD_FOTO, :COSTO, :NUM_PERS;

close C2;

COMPENSO = COMPENSO + COSTO/NUM_PERS;

fetch C1 into :CF, :COD_FOTO;

}

printf("CF %s, Compenso %f\n", CF_ATTUALE, COMPENSO);

}

close C1;

Soluzione Esercizio 2

Dato il seguente schema relazionale:

R(A,B,C)

e considerando le seguenti dipendenze funzionali:

(FD1) $AC \rightarrow B$

(FD2) $B \rightarrow A$

Le chiavi dello schema sono:

K1 = AC

K2 = BC

(FD1) $AC \rightarrow B$ è in BCNF

(FD2) $B \rightarrow A$ è in 3NF

Lo schema è pertanto in 3NF.

Data la decomposizione binaria

$R1(\underline{A}, C)$

$R2(A, \underline{B})$

Scrivere il trigger che preservi la dipendenza funzionale FD1 in caso di inserimento ed aggiornamento dati.

```
CREATE TRIGGER T1
AFTER INSERT, UPDATE ON R2(B)
REFERENCING NEW_TABLE AS NUOVAT
FOR EACH STATEMENT MODE DB2SQL
WHEN (EXISTS ( SELECT R1.A, R1.C
                FROM R1, NUOVAT
                WHERE R1.A = NUOVAT.A
                GROUP BY R1.A, R1.C
                HAVING COUNT(DISTINCT B) > 1
            ))
```

signal sqlstate '70000'('INSERIMENTO IN R2: Errore CARDINALITA MASSIMA.');

Soluzione esercizio n. 3

Dipendenza n. 1: chiave primaria sulla tabella PacchettoTuristico su *CodT*.

Dipendenza n. 2: chiave primaria sulla tabella Prodotti su *CodP*.

Dipendenza n. 3: Foreign Key su Contiene:

FK: CodP REFERENCES Prodotti

FK: CodT REFERENCES PacchettoTuristico

Dipendenza n. 4: chiave primaria sulla tabella Contiene sulla coppia *CodP, CodT*.

Dipendenza n. 5: Definire il seguente trigger:

```
CREATE TRIGGER Controllo_Numero_Prodotti
ON Contiene
FOR INSERT, UPDATE
AS
--- Dichiaro il contatore
Declare @cont int
Select @cont = Max(count(*))
from Contiene, inserted
where Contiene.CodT = inserted.CodT
And Contiene.CodP IN (SELECT CodP FROM Prodotti where Tipo = inserted.Tipo)
GROUP BY Contiene.CodT
```

```
if @cont > 3
begin
    raiserror('Una pacchetto turistico può contenere al più tre prodotti dello stesso tipo.',16,1)
    rollback transaction
end
```