

PROVA SCRITTA DI TECNOLOGIA DATABASE – 12/01/2012

Corso di Laurea Magistrale in Ingegneria Informatica – DM 270
Corso di Laurea Specialistica in Ingegneria Informatica – DM 509

PROF. SONIA BERGAMASCHI

Esercizio 1 (punti 18)

Dato il seguente schema relazionale:

PRODOTTO (CODP, NOME, PREZZO)

AZIENDA (CODA, NOME, NAZIONE)

ACQUISTO (CODE, CODP, CODA, QUANTITA)

FK: CODP **REFERENCES** PRODOTTO

FK: CODA **REFERENCES** AZIENDA

Scrivere in SQL la seguente interrogazione

- 1) Mostrare, per ciascuna nazione, i nomi dei prodotti la cui quantità acquistata è più alta della quantità media acquistata dello stesso prodotto calcolata su tutte le nazioni.

Scrivere in embedded SQL la seguente interrogazione

- 2) Selezionare, per ogni prodotto, l'azienda che ne ha acquistato la maggior quantità e la spesa complessiva sostenuta.

Scrivere in linguaggio jsp

- 3) una pagina web che consenta di inserire il nome di un prodotto e che ne generi un'altra contenente l'elenco delle aziende (ordinate per nazione) che lo hanno acquistato, includendo la quantità e l'ammontare della spesa di acquisto.

Esercizio 2 (punti 6)

Dato il seguente schema relazionale:

R(A,B,C,D)

e considerando le seguenti dipendenze funzionali:

(FD1) $C \rightarrow AB$

(FD2) $B \rightarrow D$

(FD3) $AD \rightarrow C$

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;

Produrre eventuali decomposizioni dello schema in BCNF che preservano i dati e discuterne la preservazione delle dipendenze funzionali.

Per la soluzione non ci si può avvalere del teorema 7 sulla preservazione dei dati.

Esercizio 3 (punti 6)

Dato la seguente porzione di schema relazionale:

```
CARTA_RICARICABILE(CODC, NumeroCC, SaldoCarta)
ACQUISTO(CODA, CODC, Negozio, Importo)
FK: CODC REFERENCES CARTA_RICARICABILE
```

Scrivere il Trigger (secondo la sintassi IBM DB2, MS SQLServer o ORACLE) che aggiorni il dato derivato SaldoCarta a fronte di inserimenti, aggiornamenti e cancellazioni nella relazione ACQUISTO e che assicuri che il valore SaldoCarta non diventi negativo.

Esercizio 4 – Un quesito a scelta (punti 3)

- 1) Si descriva l'enunciato e le finalità del CAP theorem
- 2) Dato il seguente schema relazionale:

R(A, B, C, D)

e considerando la dipendenza funzionale

$B \rightarrow C$

Viene richiesto di scrivere il Trigger (secondo la sintassi IBM DB2, MS SQLServer o ORACLE) che verifichi il rispetto della dipendenza funzionale a fronte di inserimenti e aggiornamenti.

Soluzione

Esercizio 1

- 1) Mostrare, per ciascuna nazione, i nomi dei prodotti la cui quantità acquistata è più alta della quantità media acquistata (dello stesso prodotto) calcolata tra tutte le nazioni.

Soluzione in cui si intende come quantità acquistata la “quantità totale”.

```
CREATE VIEW V AS
```

```
SELECT    A.NAZIONE, AC.CODP, SUM(QUANTITA) AS TOTALE
FROM      AZIENDA A, ACQUISTO AC
WHERE     A.CODA = AC.CODA
GROUP BY  A.NAZIONE, AC.CODP
```

```
SELECT    NAZIONE, CODP
FROM      V V1
WHERE     TOTALE >= ALL ( SELECT AVG(TOTALE)
                          FROM V V2
                          WHERE V1.CODP = V2.CODP)
```

- 2) Selezionare, per ogni prodotto, l'azienda che ha acquistato la maggior quantità e la somma spesa sostenuta.

```
Q1:  SELECT    AC.CODP, AC.CODA,
              SUM(AC.QUANTITA* PREZZO) AS SPESA
FROM    ACQUISTO AC, PRODOTTO P
WHERE   P.CODP = AC.CODP
GROUP BY AC.CODP, AC.CODA
HAVING SUM(QUANTITA') >=ALL ( SELECT SUM(QUANTITA)
                              FROM ACQUISTO AC2
                              WHERE AC2.CODP = AC.CODP
                              GROUP BY AC2.CODA)
```

Declare "C1" Cursor For Q1

```
open C1;
fetch C1 into : CODP, :CODA, :SPESA;
while (SQLCODE == 0){
    printf("CODP %s, CODA %s, SPESA %f,
    CODP, CODA, SPESA);
    fetch C1 into : CODP, :CODA, :SPESA;
}
close C1;
```

- 3) una pagina web che consente di inserire il nome di un prodotto e che ne genera un'altra contenente l'elenco delle aziende (ordinate per nazione) che lo hanno acquistato, la quantità e l'ammontare della spesa di acquisto.

Pagina form.html:

```
<html><head>
<title>Ricerca Prodotto</title>
</head>
<body bgcolor="white">
Inserire il nome del prodotto:
<form action="find.jsp" method="get">
<!-- In alternativa:
<form action="find.aspx" method="get">-->
<table>
<tr><td>Nome Prodotto:</td>
<td><input type="text" name="prodotto"></td></tr>
<tr><td colspan=2><input type="submit" value="Cerca"></td></tr>
</table>
</form>
</body></html>
```

Pagina find.jsp:

```
<html><head>
<title>Elenco noleggi</title>
</head><body>
<% @ page language="java" import="java.sql.*" %>
```

Risultati della ricerca

```
<%
```

```
Connection conn = null;
```

```
//carica il file di classe del driver per il ponte Odbc
Class.forName("org.postgresql.Driver");
```

```
//crea la connessione con l'origine dati
conn = DriverManager.getConnection("jdbc:postgresql://localhost/bollette",
    "postgres", "postgres");
```

```
String retrieve = " SELECT NAZIONE, A.NOME, SUM(QUANTITA) AS
QUANTITA_TOTALE, SUM(QUANTITA * PREZZO) AS SPESA_TOTALE
    FROM    AZIENDA A, ACQUISTO AC, PRODOTTO P
    WHERE   P.NOME = ?
    AND AC.CODA = A.CODA
    AND AC.CODP = P.CODP
    GROUP BY A.CODA, NAZIONE
    ORDER BY NAZIONE”
```

```
PreparedStatement st = conn.prepareStatement(retrieve);
```

```
st.setString(1, Request.getParameter("prodotto"));
```

```
ResultSet result = st.executeQuery();
```

```
%>
```

```
<table>
```

```
<tr><td>Fascia:</td></tr>
```

```
<%
```

```
while (result.next()) {
```

```
    out.println("<tr><td>" + result.getString("NOME") + "</td><td>" +
result.getString("QUANTITA_TOTALE")+ "</td><td>" +
result.getString("SPESA_TOTALE")+ "</td></tr>");
```

```
}  
st.close();  
conn.close();  
%>  
</table> </body> </html>
```

Esercizio 2

Dato il seguente schema relazionale:

$R(A,B,C,D)$

e considerando le seguenti dipendenze funzionali:

(FD1) $C \rightarrow AB$

(FD2) $B \rightarrow D$

(FD3) $AD \rightarrow C$

Le chiavi dello schema sono:

$K1 = C$

$K2 = AB$

$K3 = AD$

(FD1) $C \rightarrow AB$ è in BCNF

(FD2) $B \rightarrow D$ è in 3NF

(FD3) $AD \rightarrow C$ è in BCNF

Lo schema è pertanto in 3NF.

Decomposizione binaria:

$R1(\underline{B},D)$ preserva FD2 (proiettata)

$R2(A,B,\underline{C})$ $AK=AB$, preserva FD1 (proiettata)

Lo schema è loss-less poiché il join naturale è fatto su D, chiave di R1.

Lo schema è in BCNF, vediamo la preservazione di FD3:

$XPIUG(A,D) = AD$

Pertanto FD3 non è preservata.

Esercizio 3

Dato la seguente porzione di schema relazionale:

```
CARTA_RICARICABILE(CODC, NumeroCC, SaldoCarta)
ACQUISTO(CODA, CODC, Negozio, Importo)
FK: CODC REFERENCES CARTA_RICARICABILE
```

Scrivere il Trigger (secondo la sintassi IBM DB2, MS SQLServer o ORACLE) che aggiorni il dato derivato SaldoCarta a fronte di inserimenti, aggiornamenti e cancellazioni nella relazione ACQUISTO e che assicuri che il valore SaldoCarta non diventi negativo.

```
CREATE TRIGGER Aggiorna_Saldo
ON Acquisto
FOR INSERT, UPDATE, DELETE
AS
BEGIN
    Declare @codc char(10)
    Declare @valoreNew float
    Declare @valoreOld float
    Declare @saldo float
    ---Dichiaro il cursore
    Declare C1 CURSOR FOR
    SELECT I.CODC, I.IMPORTO, C.SaldoCarta
    FROM INSERTED I, CARTA_RICARICABILE C
    WHERE C.CODC = I.CODC

    ---Apro e carico il cursore
    OPEN C1
    FETCH NEXT FROM C1 INTO @codc, @importoNew, @saldo

    WHILE @@FETCH_STATUS = 0
    BEGIN
        IF (@saldo-@importoNew)>0
        BEGIN
            UPDATE CARTA_RICARICABILE
            SET SaldoCarta = SaldoCarta - @importoNew
            WHERE CODC = @codc
            FETCH NEXT FROM C1 INTO @codc, @importoNew, @saldo
        END
        ELSE
        BEGIN
            raiserror('Il Saldo non può essere negativo', 16,1)
            rollback transaction
        END
    END
    CLOSE C1
    DEALLOCATE C1

    ---Dichiaro il cursore
    Declare C2 CURSOR FOR
    SELECT D.CODC, D.Importo, C.Saldo
    FROM DELETED D, CARTA_RICARICABILE C
    WHERE C.CODC = D.CODC
```

```

---Apro e carico il cursore
OPEN C2
FETCH NEXT FROM C2 INTO @codc, @importoOld, @saldo

WHILE @@FETCH_STATUS = 0
BEGIN
    UPDATE CARTA_RICARICABILE
    SET Saldo = Saldo + @importoOld
    WHERE CODC = @codc
    FETCH NEXT FROM C2 INTO @codc, @importoOld, @saldo
END
CLOSE C2
DEALLOCATE C2
END

```

Esercizio 4

3) Dato il seguente schema relazionale:

R(A, B, C, D)

e considerando la dipendenza funzionale

$B \rightarrow C$

Viene richiesto di scrivere il Trigger (secondo la sintassi IBM DB2, MS SQLServer o ORACLE) che verifichi il rispetto della dipendenza funzionale a fronte di inserimenti e aggiornamenti.

```

CREATE TRIGGER DipendenzaBdeterminaC ON [dbo].[R]
FOR INSERT, UPDATE
AS
IF UPDATE(B) OR UPDATE (C)
BEGIN
declare @cont as int
    --- Dichiaro il cursore
    DECLARE R_cursor CURSOR FOR
SELECT COUNT(DISTINCT R.C)
    FROM R, Inserted I
WHERE R.B = I.B
GROUP BY R.B
    --- Apro e carico il cursore
    OPEN R_cursor

```



```
FETCH NEXT FROM R_cursor INTO @cont
  WHILE @@FETCH_STATUS = 0
  BEGIN
    IF (@cont) > 1
      BEGIN
        raiserror('Violazione della dipendenza funzionale B determina C. Numero
di elementi C distinti: %d',16,1, @cont)
        rollback transaction
      END
    ELSE
      PRINT 'Numero di elementi C distinti:' + STR(@cont)
    FETCH NEXT FROM R_cursor INTO @cont
  END
END
```