

PROVA SCRITTA DI TECNOLOGIA DATABASE – 18/01/2010

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

PROF. SONIA BERGAMASCHI

Esercizio 1 (punti 22)

Dato il seguente schema relazionale:

UTENZA (CODU, NOME, CITTA)

TARIFFA (CODT, FASCIA, PREZZO)

CONTRATTO (CODU, CODT)

FK: CODU **REFERENCES** UTENZA

FK: CODT **REFERENCES** TARIFFA

BOLLETTA (CODU, ANNO, TRIMESTRE, CONSUMO)

FK: CODU **REFERENCES** UTENZA

Scrivere in SQL la seguente interrogazione

- 1) Mostrare, per ogni fascia tariffaria, il periodo (trimestre ed anno) con il consumo medio più alto.

Scrivere in embedded SQL la seguente interrogazione

- 2) Mostrare, per ogni città, il costo medio annuale (costo = consumo * prezzo) delle bollette relative a ciascuna fascia.

Scrivere in linguaggio jsp o asp.net

- 3) una pagina web che consente di inserire il valore di consumo che ne genera un'altra contenente l'elenco delle fasce che hanno registrato sempre consumi superiori al valore inserito.

Esercizio 2 (punti 5)

Dato il seguente schema relazionale:

R(A,B,C,D)

e considerando le seguenti dipendenze funzionali:

(FD1) $AB \rightarrow D$

(FD2) $BC \rightarrow A$

(FD3) $D \rightarrow C$

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:

```
PARTITA (CODS1, CODS2, RISULTATO)  
CLASSIFICA (CODS, PUNTI)
```

Il risultato di una partita può avere i seguenti valori: 1 (vincitrice CODS1), 2 (vincitrice CODS2), X (pareggio). Per la vittoria sono assegnati 3 punti, per il pareggio 1, per la sconfitta nessuno.

Scrivere il Trigger (secondo la sintassi IBM DB2, MS SQLServer o ORACLE) che mantenga aggiornata la classifica.

Soluzione

Esercizio 1

- 1) Mostrare, per ogni fascia tariffaria, il periodo (trimestre ed anno) con il consumo medio più alto.

```
CREATE VIEW V AS
SELECT    T.FASCIA, B.ANNO, B.TRIMESTRE, AVG(CONSUMO)
FROM      TARIFFA AS T, CONTRATTO AS C, BOLLETTA AS B
WHERE     T.CODT = C.CODT
AND       C.CODU = B.CODU
GROUP BY  T.FASCIA, B.ANNO, B.TRIMESTRE

SELECT    FASCIA, ANNO, TRIMESTRE
FROM      V AS V1
WHERE     CONSUMO = (SELECT    MAX(CONSUMO)
                    FROM V AS V2
                    WHERE V2.FASCIA = V1.FASCIA)
```

- 2) Mostrare, per ogni città, il costo medio annuale (costo = consumo * prezzo) delle bollette relative a ciascuna fascia.

```
Q1:  CREATE VIEW V AS
      SELECT    B.CODU, B.ANNO, T.FASCIA,
                SUM(B.CONSUMO * T.PREZZO) AS COSTO
      FROM      TARIFFA AS T, CONTRATTO AS C, BOLLETTA AS B
      WHERE     T.CODT = C.CODT
      AND       C.CODU = B.CODU
      GROUP BY  B.CODU, B.ANNO, T.FASCIA

Q2:  SELECT    CITTA, ANNO, FASCIA, AVG(COSTO) AS COSTOMEDIO
      FROM      V, UTENZA
      WHERE     V.CODU = UTENZA.CODC
      GROUP BY  CITTA, FASCIA, ANNO
```

ORDER BY CITTA, FASCIA, ANNO

Exec sql Q1;

Declare Cursor "C1" For Q2

open C1;

fetch C1 into : CITTA, :ANNO, :FASCIA, :COSTOMEDIO;

while (SQLCODE == 0){

printf("CITTA %s, ANNO %s, FASCIA %s, COSTOMEDIO %f \n",

CITTA, ANNO, FASCIA, COSTOMEDIO);

fetch C1 into : CITTA, :ANNO, :FASCIA, :COSTOMEDIO;

}

close C1;

- 3) una pagina web che consente di inserire il valore di consumo che ne genera un'altra contenente l'elenco delle fasce che hanno registrato sempre consumi superiori al valore inserito.

Pagina form.html:

```
<html><head>
```

```
<title>Ricerca per Fasce</title>
```

```
</head>
```

```
<body bgcolor="white">
```

Inserire il consumo minimo:

```
<form action="find.jsp" method="get">
```

```
<!-- In alternativa:
```

```
<form action="find.aspx" method="get">-->
```

```
<table>
```

```
<tr><td>Consumo:</td>
```

```
<td><input type="text" name="consumo"></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",
    "postrgres", "postrgres");
//crea lo statement
Statement st = conn.createStatement();
String retrieve = " SELECT  T1.FASCIA
                    FROM    TARIFFA AS T1
                    WHERE   T1.FASCIA NOT IN
                            (SELECT T2.FASCIA
                             FROM  TARIFFA AS T2, CONTRATTO AS C2,
                             BOLLETTA AS B2
                             WHERE T2.CODT = C2.CODT AND C2. CODU = B2.CODU
                             AND   B2.CONSUMO <= ?)";

st = conn.prepareStatement(retrieve);
st.setString(1, Request.getParameter("consumo"));
ResultSet result = statement.executeQuery();
%>

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

```

<%
while (result.next()) {out.println("<tr><td>" + result.getString("FASCIA") +
"</td></tr>"

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

```

Pagina find.aspx:

```

<%@ Page Language="VB" %>
<html>
<head runat="server">
<title>Fascia</title>
</head>
<body>
<form id="form1" runat="server">
<asp:GridView ID="GridView1" Runat="server" DataSourceID="SqlDataSource1"
AutoGenerateColumns="False" HeaderText="Elenco forniture" AllowPaging="True">
<Columns>
<asp:BoundField HeaderText="Nome Scuola" DataField="CONSUMO "
SortExpression=" CONSUMO " />
</ Columns>
</ asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" Runat="server"
SelectCommand=" SELECT T1.FASCIA
FROM TARIFFA AS T1
WHERE T1.FASCIA NOT IN
(SELECT T2.FASCIA
FROM TARIFFA AS T2, CONTRATTO AS C2,
BOLLETTA AS B2

```

```
WHERE T2.CODT = C2.CODT AND C2. CODU = B2.CODU
AND B2.CONSUMO <= @Nconsumo)"
```

```
ConnectionString="<%%$ ConnectionStrings:bollette %>">
<SelectParameters>
<asp:QueryStringParameter Name="Nconsumo"
QueryStringField="consumo" />
</asp:SqlDataSource>
</form>
</body>
</html>
```

Esercizio 2

Dato il seguente schema relazionale:

R(A,B,C,D)

e considerando le seguenti dipendenze funzionali:

(FD1) $AB \rightarrow D$

(FD2) $BC \rightarrow A$

(FD3) $D \rightarrow C$

Le chiavi dello schema sono:

K1 = AB

K2 = BC

K3 = BD

(FD1) $AB \rightarrow D$ è in BCNF

(FD2) $BC \rightarrow A$ è in BCNF

(FD3) $D \rightarrow C$ è in 3NF

Lo schema è pertanto in 3NF.

Decomposizione binaria:

R1(D,C) preserva FD3 (proiettata)

R2(A,B,D) AK=BD, 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 FD2:

$XPIUG(B,C) = BC$

Pertanto FD2 non è preservata.

Esercizio 3

Dato la seguente porzione di schema relazionale:

PARTITA (CODS1, CODS2, RISULTATO)
CLASSIFICA (CODS, PUNTI)

Il risultato di una partita può avere i seguenti valori: 1 (vincitrice CODS1), 2 (vincitrice CODS2), X (pareggio). Per la vittoria sono assegnati 3 punti, per il pareggio 1, per la sconfitta nessuno.

Scrivere il Trigger (secondo la sintassi IBM DB2, MS SQLServer o ORACLE) che mantenga aggiornata la classifica.

```
CREATE TRIGGER Aggiorna_Classifica
AFTER INSERT, UPDATE, DELETE ON PARTITA
REFERENCING NEW AS N, OLD AS O
FOR EACH ROW MODE DB2SQL
BEGIN ATOMIC
    UPDATE CLASSIFICA
    SET PUNTI = PUNTI + 1
    WHERE N.RISULTATO = 'X'
    AND (CODS = N.CODS1 OR CODS = N.CODS2)

    UPDATE CLASSIFICA
    SET PUNTI = PUNTI + 3
    WHERE (N.RISULTATO = '1' AND CODS = N.CODS1)
    OR (N.RISULTATO = '2' OR CODS = N.CODS2)

    UPDATE CLASSIFICA
    SET PUNTI = PUNTI - 1
    WHERE N.RISULTATO = 'X'
```


AND (CODS = O.CODS1 OR CODS = O.CODS2)

UPDATE CLASSIFICA

SET PUNTI = PUNTI - 3

WHERE (O.RISULTATO = '1' AND CODS = N.CODS1)

OR (O.RISULTATO = '2' OR CODS = N.CODS2)

END

PROVA SCRITTA DI TECNOLOGIA DATABASE – 18/01/2010

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

PROF. SONIA BERGAMASCHI

Esercizio 1 (punti 22)

Dato il seguente schema relazionale:

UTENZA (CODU, NOME, CITTA)

TARIFFA (CODT, FASCIA, PREZZO)

CONTRATTO (CODU, CODT)

FK: CODU **REFERENCES** UTENZA

FK: CODT **REFERENCES** TARIFFA

BOLLETTA (CODU, ANNO, TRIMESTRE, CONSUMO)

FK: CODU **REFERENCES** UTENZA

Scrivere in SQL la seguente interrogazione

- 1) Mostrare, per ogni fascia tariffaria, il periodo (trimestre ed anno) con il consumo medio più alto.

Scrivere in embedded SQL la seguente interrogazione

- 2) Mostrare, per ogni città, il costo medio annuale (costo = consumo * prezzo) delle bollette relative a ciascuna fascia.

Scrivere in linguaggio jsp o asp.net

- 3) una pagina web che consente di inserire il valore di consumo che ne genera un'altra contenente l'elenco delle fasce che hanno registrato sempre consumi superiori al valore inserito.

Esercizio 2 (punti 5)

Dato il seguente schema relazionale:

R(A,B,C,D)

e considerando le seguenti dipendenze funzionali:

(FD1) $AB \rightarrow D$

(FD2) $BC \rightarrow A$

(FD3) $D \rightarrow C$

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:

```
PARTITA (CODS1, CODS2, RISULTATO)  
CLASSIFICA (CODS, PUNTI)
```

Il risultato di una partita può avere i seguenti valori: 1 (vincitrice CODS1), 2 (vincitrice CODS2), X (pareggio). Per la vittoria sono assegnati 3 punti, per il pareggio 1, per la sconfitta nessuno.

Scrivere il Trigger (secondo la sintassi IBM DB2, MS SQLServer o ORACLE) che mantenga aggiornata la classifica.

Soluzione

Esercizio 1

- 1) Mostrare, per ogni fascia tariffaria, il periodo (trimestre ed anno) con il consumo medio più alto.

```
CREATE VIEW V AS
```

```
SELECT    T.FASCIA, B.ANNO, B.TRIMESTRE, AVG(CONSUMO)
FROM      TARIFFA AS T, CONTRATTO AS C, BOLLETTA AS B
WHERE     T.CODT = C.CODT
AND       C.CODU = B.CODU
GROUP BY  T.FASCIA, B.ANNO, B.TRIMESTRE
```

```
SELECT    FASCIA, ANNO, TRIMESTRE
FROM      V AS V1
WHERE     CONSUMO = (SELECT    MAX(CONSUMO)
                    FROM V AS V2
                    WHERE V2.FASCIA = V1.FASCIA)
```

- 2) Mostrare, per ogni città, il costo medio annuale (costo = consumo * prezzo) delle bollette relative a ciascuna fascia.

Q1: CREATE VIEW V AS

```
SELECT    B.CODU, B.ANNO, T.FASCIA,
          SUM(B.CONSUMO * T.PREZZO) AS COSTO
FROM      TARIFFA AS T, CONTRATTO AS C, BOLLETTA AS B
WHERE     T.CODT = C.CODT
AND       C.CODU = B.CODU
GROUP BY  B.CODU, B.ANNO, T.FASCIA
```

Q2: SELECT CITTA, ANNO, FASCIA, AVG(COSTO) AS COSTOMEDIO
FROM V, UTENZA
WHERE V.CODU = UTENZA.CODC
GROUP BY CITTA, FASCIA, ANNO

ORDER BY CITTA, FASCIA, ANNO

Exec sql Q1;

Declare Cursor "C1" For Q2

open C1;

fetch C1 into : CITTA, :ANNO, :FASCIA, :COSTOMEDIO;

while (SQLCODE == 0){

printf("CITTA %s, ANNO %s, FASCIA %s, COSTOMEDIO %f \n",

CITTA, ANNO, FASCIA, COSTOMEDIO);

fetch C1 into : CITTA, :ANNO, :FASCIA, :COSTOMEDIO;

}

close C1;

- 3) una pagina web che consente di inserire il valore di consumo che ne genera un'altra contenente l'elenco delle fasce che hanno registrato sempre consumi superiori al valore inserito.

Pagina form.html:

```
<html><head>
```

```
<title>Ricerca per Fasce</title>
```

```
</head>
```

```
<body bgcolor="white">
```

Inserire il consumo minimo:

```
<form action="find.jsp" method="get">
```

```
<!-- In alternativa:
```

```
<form action="find.aspx" method="get">-->
```

```
<table>
```

```
<tr><td>Consumo:</td>
```

```
<td><input type="text" name="consumo"></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",
    "postrgres", "postrgres");
//crea lo statement
Statement st = conn.createStatement();
String retrieve = " SELECT  T1.FASCIA
                    FROM    TARIFFA AS T1
                    WHERE   T1.FASCIA NOT IN
                            (SELECT T2.FASCIA
                             FROM  TARIFFA AS T2, CONTRATTO AS C2,
                             BOLLETTA AS B2
                             WHERE T2.CODT = C2.CODT AND C2. CODU = B2.CODU
                             AND   B2.CONSUMO <= ?)";

st = conn.prepareStatement(retrieve);
st.setString(1, Request.getParameter("consumo"));
ResultSet result = statement.executeQuery();
%>

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

```

<%
while (result.next()) {out.println("<tr><td>" + result.getString("FASCIA") +
"</td></tr>"

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

```

Pagina find.aspx:

```

<%@ Page Language="VB" %>
<html>
<head runat="server">
<title>Fascia</title>
</head>
<body>
<form id="form1" runat="server">
<asp:GridView ID="GridView1" Runat="server" DataSourceID="SqlDataSource1"
AutoGenerateColumns="False" HeaderText="Elenco forniture" AllowPaging="True">
<Columns>
<asp:BoundField HeaderText="Nome Scuola" DataField="CONSUMO "
SortExpression=" CONSUMO " />
</ Columns>
</ asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" Runat="server"
SelectCommand=" SELECT T1.FASCIA
FROM TARIFFA AS T1
WHERE T1.FASCIA NOT IN
(SELECT T2.FASCIA
FROM TARIFFA AS T2, CONTRATTO AS C2,
BOLLETTA AS B2

```

```
WHERE T2.CODT = C2.CODT AND C2. CODU = B2.CODU
AND B2.CONSUMO <= @Nconsumo)"
```

```
ConnectionString="<%%$ ConnectionStrings:bollette %>">
<SelectParameters>
<asp:QueryStringParameter Name="Nconsumo"
QueryStringField="consumo" />
</asp:SqlDataSource>
</form>
</body>
</html>
```

Esercizio 2

Dato il seguente schema relazionale:

R(A,B,C,D)

e considerando le seguenti dipendenze funzionali:

(FD1) $AB \rightarrow D$

(FD2) $BC \rightarrow A$

(FD3) $D \rightarrow C$

Le chiavi dello schema sono:

K1 = AB

K2 = BC

K3 = BD

(FD1) $AB \rightarrow D$ è in BCNF

(FD2) $BC \rightarrow A$ è in BCNF

(FD3) $D \rightarrow C$ è in 3NF

Lo schema è pertanto in 3NF.

Decomposizione binaria:

R1(D,C) preserva FD3 (proiettata)

R2(A,B,D) AK=BD, 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 FD2:

$XPIUG(B,C) = BC$

Pertanto FD2 non è preservata.

Esercizio 3

Dato la seguente porzione di schema relazionale:

PARTITA (CODS1, CODS2, RISULTATO)
CLASSIFICA (CODS, PUNTI)

Il risultato di una partita può avere i seguenti valori: 1 (vincitrice CODS1), 2 (vincitrice CODS2), X (pareggio). Per la vittoria sono assegnati 3 punti, per il pareggio 1, per la sconfitta nessuno.

Scrivere il Trigger (secondo la sintassi IBM DB2, MS SQLServer o ORACLE) che mantenga aggiornata la classifica.

```
CREATE TRIGGER Aggiorna_Classifica
AFTER INSERT, UPDATE, DELETE ON PARTITA
REFERENCING NEW AS N, OLD AS O
FOR EACH ROW MODE DB2SQL
BEGIN ATOMIC
    UPDATE CLASSIFICA
    SET PUNTI = PUNTI + 1
    WHERE N.RISULTATO = 'X'
    AND (CODS = N.CODS1 OR CODS = N.CODS2)

    UPDATE CLASSIFICA
    SET PUNTI = PUNTI + 3
    WHERE (N.RISULTATO = '1' AND CODS = N.CODS1)
    OR (N.RISULTATO = '2' OR CODS = N.CODS2)

    UPDATE CLASSIFICA
    SET PUNTI = PUNTI - 1
    WHERE N.RISULTATO = 'X'
```

AND (CODS = O.CODS1 OR CODS = O.CODS2)

UPDATE CLASSIFICA

SET PUNTI = PUNTI - 3

WHERE (O.RISULTATO = '1' AND CODS = N.CODS1)

OR (O.RISULTATO = '2' OR CODS = N.CODS2)

END

PROVA SCRITTA DI TECNOLOGIA DATABASE – 18/01/2010

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

PROF. SONIA BERGAMASCHI

Esercizio 1 (punti 22)

Dato il seguente schema relazionale:

UTENZA (CODU, NOME, CITTA)

TARIFFA (CODT, FASCIA, PREZZO)

CONTRATTO (CODU, CODT)

FK: CODU **REFERENCES** UTENZA

FK: CODT **REFERENCES** TARIFFA

BOLLETTA (CODU, ANNO, TRIMESTRE, CONSUMO)

FK: CODU **REFERENCES** UTENZA

Scrivere in SQL la seguente interrogazione

- 1) Mostrare, per ogni fascia tariffaria, il periodo (trimestre ed anno) con il consumo medio più alto.

Scrivere in embedded SQL la seguente interrogazione

- 2) Mostrare, per ogni città, il costo medio annuale (costo = consumo * prezzo) delle bollette relative a ciascuna fascia.

Scrivere in linguaggio jsp o asp.net

- 3) una pagina web che consente di inserire il valore di consumo che ne genera un'altra contenente l'elenco delle fasce che hanno registrato sempre consumi superiori al valore inserito.

Esercizio 2 (punti 5)

Dato il seguente schema relazionale:

R(A,B,C,D)

e considerando le seguenti dipendenze funzionali:

(FD1) $AB \rightarrow D$

(FD2) $BC \rightarrow A$

(FD3) $D \rightarrow C$

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:

```
PARTITA (CODS1, CODS2, RISULTATO)  
CLASSIFICA (CODS, PUNTI)
```

Il risultato di una partita può avere i seguenti valori: 1 (vincitrice CODS1), 2 (vincitrice CODS2), X (pareggio). Per la vittoria sono assegnati 3 punti, per il pareggio 1, per la sconfitta nessuno.

Scrivere il Trigger (secondo la sintassi IBM DB2, MS SQLServer o ORACLE) che mantenga aggiornata la classifica.

Soluzione

Esercizio 1

- 1) Mostrare, per ogni fascia tariffaria, il periodo (trimestre ed anno) con il consumo medio più alto.

```
CREATE VIEW V AS
```

```
SELECT    T.FASCIA, B.ANNO, B.TRIMESTRE, AVG(CONSUMO)
FROM      TARIFFA AS T, CONTRATTO AS C, BOLLETTA AS B
WHERE     T.CODT = C.CODT
AND       C.CODU = B.CODU
GROUP BY  T.FASCIA, B.ANNO, B.TRIMESTRE
```

```
SELECT    FASCIA, ANNO, TRIMESTRE
FROM      V AS V1
WHERE     CONSUMO = (SELECT    MAX(CONSUMO)
                    FROM V AS V2
                    WHERE V2.FASCIA = V1.FASCIA)
```

- 2) Mostrare, per ogni città, il costo medio annuale (costo = consumo * prezzo) delle bollette relative a ciascuna fascia.

Q1: CREATE VIEW V AS

```
SELECT    B.CODU, B.ANNO, T.FASCIA,
          SUM(B.CONSUMO * T.PREZZO) AS COSTO
FROM      TARIFFA AS T, CONTRATTO AS C, BOLLETTA AS B
WHERE     T.CODT = C.CODT
AND       C.CODU = B.CODU
GROUP BY  B.CODU, B.ANNO, T.FASCIA
```

Q2: SELECT CITTA, ANNO, FASCIA, AVG(COSTO) AS COSTOMEDIO
FROM V, UTENZA
WHERE V.CODU = UTENZA.CODC
GROUP BY CITTA, FASCIA, ANNO

ORDER BY CITTA, FASCIA, ANNO

Exec sql Q1;

Declare Cursor "C1" For Q2

open C1;

fetch C1 into : CITTA, :ANNO, :FASCIA, :COSTOMEDIO;

while (SQLCODE == 0){

printf("CITTA %s, ANNO %s, FASCIA %s, COSTOMEDIO %f \n",

CITTA, ANNO, FASCIA, COSTOMEDIO);

fetch C1 into : CITTA, :ANNO, :FASCIA, :COSTOMEDIO;

}

close C1;

- 3) una pagina web che consente di inserire il valore di consumo che ne genera un'altra contenente l'elenco delle fasce che hanno registrato sempre consumi superiori al valore inserito.

Pagina form.html:

```
<html><head>
```

```
<title>Ricerca per Fasce</title>
```

```
</head>
```

```
<body bgcolor="white">
```

Inserire il consumo minimo:

```
<form action="find.jsp" method="get">
```

```
<!-- In alternativa:
```

```
<form action="find.aspx" method="get">-->
```

```
<table>
```

```
<tr><td>Consumo:</td>
```

```
<td><input type="text" name="consumo"></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",
    "postrgres", "postrgres");
//crea lo statement
Statement st = conn.createStatement();
String retrieve = " SELECT  T1.FASCIA
                    FROM    TARIFFA AS T1
                    WHERE   T1.FASCIA NOT IN
                            (SELECT T2.FASCIA
                             FROM  TARIFFA AS T2, CONTRATTO AS C2,
                                    BOLLETTA AS B2
                             WHERE T2.CODT = C2.CODT AND C2. CODU = B2.CODU
                                    AND  B2.CONSUMO <= ?)";

st = conn.prepareStatement(retrieve);
st.setString(1, Request.getParameter("consumo"));
ResultSet result = statement.executeQuery();
%>

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

```

<%
while (result.next()) {out.println("<tr><td>" + result.getString("FASCIA") +
"</td></tr>"

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

```

Pagina find.aspx:

```

<%@ Page Language="VB" %>
<html>
<head runat="server">
<title>Fascia</title>
</head>
<body>
<form id="form1" runat="server">
<asp:GridView ID="GridView1" Runat="server" DataSourceID="SqlDataSource1"
AutoGenerateColumns="False" HeaderText="Elenco forniture" AllowPaging="True">
<Columns>
<asp:BoundField HeaderText="Nome Scuola" DataField="CONSUMO "
SortExpression=" CONSUMO " />
</ Columns>
</ asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" Runat="server"
SelectCommand=" SELECT T1.FASCIA
FROM TARIFFA AS T1
WHERE T1.FASCIA NOT IN
(SELECT T2.FASCIA
FROM TARIFFA AS T2, CONTRATTO AS C2,
BOLLETTA AS B2

```



```
WHERE T2.CODT = C2.CODT AND C2. CODU = B2.CODU
AND B2.CONSUMO <= @Nconsumo)"
```

```
ConnectionString="<%%$ ConnectionStrings:bollette %>">
<SelectParameters>
<asp:QueryStringParameter Name="Nconsumo"
QueryStringField="consumo" />
</asp:SqlDataSource>
</form>
</body>
</html>
```

Esercizio 2

Dato il seguente schema relazionale:

R(A,B,C,D)

e considerando le seguenti dipendenze funzionali:

(FD1) $AB \rightarrow D$

(FD2) $BC \rightarrow A$

(FD3) $D \rightarrow C$

Le chiavi dello schema sono:

K1 = AB

K2 = BC

K3 = BD

(FD1) $AB \rightarrow D$ è in BCNF

(FD2) $BC \rightarrow A$ è in BCNF

(FD3) $D \rightarrow C$ è in 3NF

Lo schema è pertanto in 3NF.

Decomposizione binaria:

R1(D,C) preserva FD3 (proiettata)

R2(A,B,D) AK=BD, 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 FD2:

$XPIUG(B,C) = BC$

Pertanto FD2 non è preservata.

Esercizio 3

Dato la seguente porzione di schema relazionale:

PARTITA (CODS1, CODS2, RISULTATO)
CLASSIFICA (CODS, PUNTI)

Il risultato di una partita può avere i seguenti valori: 1 (vincitrice CODS1), 2 (vincitrice CODS2), X (pareggio). Per la vittoria sono assegnati 3 punti, per il pareggio 1, per la sconfitta nessuno.

Scrivere il Trigger (secondo la sintassi IBM DB2, MS SQLServer o ORACLE) che mantenga aggiornata la classifica.

```
CREATE TRIGGER Aggiorna_Classifica
AFTER INSERT, UPDATE, DELETE ON PARTITA
REFERENCING NEW AS N, OLD AS O
FOR EACH ROW MODE DB2SQL
BEGIN ATOMIC
    UPDATE CLASSIFICA
    SET PUNTI = PUNTI + 1
    WHERE N.RISULTATO = 'X'
    AND (CODS = N.CODS1 OR CODS = N.CODS2)

    UPDATE CLASSIFICA
    SET PUNTI = PUNTI + 3
    WHERE (N.RISULTATO = '1' AND CODS = N.CODS1)
    OR (N.RISULTATO = '2' OR CODS = N.CODS2)

    UPDATE CLASSIFICA
    SET PUNTI = PUNTI - 1
    WHERE N.RISULTATO = 'X'
```

AND (CODS = O.CODS1 OR CODS = O.CODS2)

UPDATE CLASSIFICA

SET PUNTI = PUNTI - 3

WHERE (O.RISULTATO = '1' AND CODS = N.CODS1)

OR (O.RISULTATO = '2' OR CODS = N.CODS2)

END

PROVA SCRITTA DI TECNOLOGIA DATABASE – 18/01/2010

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

PROF. SONIA BERGAMASCHI

Esercizio 1 (punti 22)

Dato il seguente schema relazionale:

UTENZA (CODU, NOME, CITTA)

TARIFFA (CODT, FASCIA, PREZZO)

CONTRATTO (CODU, CODT)

FK: CODU **REFERENCES** UTENZA

FK: CODT **REFERENCES** TARIFFA

BOLLETTA (CODU, ANNO, TRIMESTRE, CONSUMO)

FK: CODU **REFERENCES** UTENZA

Scrivere in SQL la seguente interrogazione

- 1) Mostrare, per ogni fascia tariffaria, il periodo (trimestre ed anno) con il consumo medio più alto.

Scrivere in embedded SQL la seguente interrogazione

- 2) Mostrare, per ogni città, il costo medio annuale (costo = consumo * prezzo) delle bollette relative a ciascuna fascia.

Scrivere in linguaggio jsp o asp.net

- 3) una pagina web che consente di inserire il valore di consumo che ne genera un'altra contenente l'elenco delle fasce che hanno registrato sempre consumi superiori al valore inserito.

Esercizio 2 (punti 5)

Dato il seguente schema relazionale:

R(A,B,C,D)

e considerando le seguenti dipendenze funzionali:

(FD1) $AB \rightarrow D$

(FD2) $BC \rightarrow A$

(FD3) $D \rightarrow C$

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:

```
PARTITA (CODS1, CODS2, RISULTATO)  
CLASSIFICA (CODS, PUNTI)
```

Il risultato di una partita può avere i seguenti valori: 1 (vincitrice CODS1), 2 (vincitrice CODS2), X (pareggio). Per la vittoria sono assegnati 3 punti, per il pareggio 1, per la sconfitta nessuno.

Scrivere il Trigger (secondo la sintassi IBM DB2, MS SQLServer o ORACLE) che mantenga aggiornata la classifica.

Soluzione

Esercizio 1

- 1) Mostrare, per ogni fascia tariffaria, il periodo (trimestre ed anno) con il consumo medio più alto.

```
CREATE VIEW V AS
```

```
SELECT    T.FASCIA, B.ANNO, B.TRIMESTRE, AVG(CONSUMO)
FROM      TARIFFA AS T, CONTRATTO AS C, BOLLETTA AS B
WHERE     T.CODT = C.CODT
AND       C.CODU = B.CODU
GROUP BY  T.FASCIA, B.ANNO, B.TRIMESTRE
```

```
SELECT    FASCIA, ANNO, TRIMESTRE
FROM      V AS V1
WHERE     CONSUMO = (SELECT    MAX(CONSUMO)
                    FROM V AS V2
                    WHERE V2.FASCIA = V1.FASCIA)
```

- 2) Mostrare, per ogni città, il costo medio annuale (costo = consumo * prezzo) delle bollette relative a ciascuna fascia.

Q1: CREATE VIEW V AS

```
SELECT    B.CODU, B.ANNO, T.FASCIA,
          SUM(B.CONSUMO * T.PREZZO) AS COSTO
FROM      TARIFFA AS T, CONTRATTO AS C, BOLLETTA AS B
WHERE     T.CODT = C.CODT
AND       C.CODU = B.CODU
GROUP BY  B.CODU, B.ANNO, T.FASCIA
```

Q2: SELECT CITTA, ANNO, FASCIA, AVG(COSTO) AS COSTOMEDIO
FROM V, UTENZA
WHERE V.CODU = UTENZA.CODC
GROUP BY CITTA, FASCIA, ANNO

ORDER BY CITTA, FASCIA, ANNO

Exec sql Q1;

Declare Cursor "C1" For Q2

open C1;

fetch C1 into : CITTA, :ANNO, :FASCIA, :COSTOMEDIO;

while (SQLCODE == 0){

printf("CITTA %s, ANNO %s, FASCIA %s, COSTOMEDIO %f \n",

CITTA, ANNO, FASCIA, COSTOMEDIO);

fetch C1 into : CITTA, :ANNO, :FASCIA, :COSTOMEDIO;

}

close C1;

- 3) una pagina web che consente di inserire il valore di consumo che ne genera un'altra contenente l'elenco delle fasce che hanno registrato sempre consumi superiori al valore inserito.

Pagina form.html:

```
<html><head>
```

```
<title>Ricerca per Fasce</title>
```

```
</head>
```

```
<body bgcolor="white">
```

Inserire il consumo minimo:

```
<form action="find.jsp" method="get">
```

```
<!-- In alternativa:
```

```
<form action="find.aspx" method="get">-->
```

```
<table>
```

```
<tr><td>Consumo:</td>
```

```
<td><input type="text" name="consumo"></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",
    "postrgres", "postrgres");
//crea lo statement
Statement st = conn.createStatement();
String retrieve = " SELECT  T1.FASCIA
                    FROM    TARIFFA AS T1
                    WHERE   T1.FASCIA NOT IN
                            (SELECT T2.FASCIA
                             FROM  TARIFFA AS T2, CONTRATTO AS C2,
                             BOLLETTA AS B2
                             WHERE T2.CODT = C2.CODT AND C2. CODU = B2.CODU
                             AND   B2.CONSUMO <= ?)";

st = conn.prepareStatement(retrieve);
st.setString(1, Request.getParameter("consumo"));
ResultSet result = statement.executeQuery();
%>

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



```

<%
while (result.next()) {out.println("<tr><td>" + result.getString("FASCIA") +
"</td></tr>"

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

```

Pagina find.aspx:

```

<%@ Page Language="VB" %>
<html>
<head runat="server">
<title>Fascia</title>
</head>
<body>
<form id="form1" runat="server">
<asp:GridView ID="GridView1" Runat="server" DataSourceID="SqlDataSource1"
AutoGenerateColumns="False" HeaderText="Elenco forniture" AllowPaging="True">
<Columns>
<asp:BoundField HeaderText="Nome Scuola" DataField="CONSUMO "
SortExpression=" CONSUMO " />
</ Columns>
</ asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" Runat="server"
SelectCommand=" SELECT T1.FASCIA
FROM TARIFFA AS T1
WHERE T1.FASCIA NOT IN
(SELECT T2.FASCIA
FROM TARIFFA AS T2, CONTRATTO AS C2,
BOLLETTA AS B2

```

```
WHERE T2.CODT = C2.CODT AND C2. CODU = B2.CODU
AND B2.CONSUMO <= @Nconsumo)"
```

```
ConnectionString="<%%$ ConnectionStrings:bollette %>">
<SelectParameters>
<asp:QueryStringParameter Name="Nconsumo"
QueryStringField="consumo" />
</asp:SqlDataSource>
</form>
</body>
</html>
```

Esercizio 2

Dato il seguente schema relazionale:

R(A,B,C,D)

e considerando le seguenti dipendenze funzionali:

(FD1) $AB \rightarrow D$

(FD2) $BC \rightarrow A$

(FD3) $D \rightarrow C$

Le chiavi dello schema sono:

K1 = AB

K2 = BC

K3 = BD

(FD1) $AB \rightarrow D$ è in BCNF

(FD2) $BC \rightarrow A$ è in BCNF

(FD3) $D \rightarrow C$ è in 3NF

Lo schema è pertanto in 3NF.

Decomposizione binaria:

R1(D,C) preserva FD3 (proiettata)

R2(A,B,D) AK=BD, 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 FD2:

$XPIUG(B,C) = BC$

Pertanto FD2 non è preservata.

Esercizio 3

Dato la seguente porzione di schema relazionale:

PARTITA (CODS1, CODS2, RISULTATO)
CLASSIFICA (CODS, PUNTI)

Il risultato di una partita può avere i seguenti valori: 1 (vincitrice CODS1), 2 (vincitrice CODS2), X (pareggio). Per la vittoria sono assegnati 3 punti, per il pareggio 1, per la sconfitta nessuno.

Scrivere il Trigger (secondo la sintassi IBM DB2, MS SQLServer o ORACLE) che mantenga aggiornata la classifica.

```
CREATE TRIGGER Aggiorna_Classifica
AFTER INSERT, UPDATE, DELETE ON PARTITA
REFERENCING NEW AS N, OLD AS O
FOR EACH ROW MODE DB2SQL
BEGIN ATOMIC
    UPDATE CLASSIFICA
    SET PUNTI = PUNTI + 1
    WHERE N.RISULTATO = 'X'
    AND (CODS = N.CODS1 OR CODS = N.CODS2)

    UPDATE CLASSIFICA
    SET PUNTI = PUNTI + 3
    WHERE (N.RISULTATO = '1' AND CODS = N.CODS1)
    OR (N.RISULTATO = '2' OR CODS = N.CODS2)

    UPDATE CLASSIFICA
    SET PUNTI = PUNTI - 1
    WHERE N.RISULTATO = 'X'
```

AND (CODS = O.CODS1 OR CODS = O.CODS2)

UPDATE CLASSIFICA

SET PUNTI = PUNTI - 3

WHERE (O.RISULTATO = '1' AND CODS = N.CODS1)

OR (O.RISULTATO = '2' OR CODS = N.CODS2)

END

PROVA SCRITTA DI TECNOLOGIA DATABASE – 18/01/2010

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

PROF. SONIA BERGAMASCHI

Esercizio 1 (punti 22)

Dato il seguente schema relazionale:

UTENZA (CODU, NOME, CITTA)

TARIFFA (CODT, FASCIA, PREZZO)

CONTRATTO (CODU, CODT)

FK: CODU **REFERENCES** UTENZA

FK: CODT **REFERENCES** TARIFFA

BOLLETTA (CODU, ANNO, TRIMESTRE, CONSUMO)

FK: CODU **REFERENCES** UTENZA

Scrivere in SQL la seguente interrogazione

- 1) Mostrare, per ogni fascia tariffaria, il periodo (trimestre ed anno) con il consumo medio più alto.

Scrivere in embedded SQL la seguente interrogazione

- 2) Mostrare, per ogni città, il costo medio annuale (costo = consumo * prezzo) delle bollette relative a ciascuna fascia.

Scrivere in linguaggio jsp o asp.net

- 3) una pagina web che consente di inserire il valore di consumo che ne genera un'altra contenente l'elenco delle fasce che hanno registrato sempre consumi superiori al valore inserito.

Esercizio 2 (punti 5)

Dato il seguente schema relazionale:

R(A,B,C,D)

e considerando le seguenti dipendenze funzionali:

(FD1) AB → D

(FD2) BC → A

(FD3) D → 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:

```
PARTITA (CODS1, CODS2, RISULTATO)  
CLASSIFICA (CODS, PUNTI)
```

Il risultato di una partita può avere i seguenti valori: 1 (vincitrice CODS1), 2 (vincitrice CODS2), X (pareggio). Per la vittoria sono assegnati 3 punti, per il pareggio 1, per la sconfitta nessuno.

Scrivere il Trigger (secondo la sintassi IBM DB2, MS SQLServer o ORACLE) che mantenga aggiornata la classifica.

Soluzione

Esercizio 1

- 1) Mostrare, per ogni fascia tariffaria, il periodo (trimestre ed anno) con il consumo medio più alto.

```
CREATE VIEW V AS
```

```
SELECT    T.FASCIA, B.ANNO, B.TRIMESTRE, AVG(CONSUMO)
FROM      TARIFFA AS T, CONTRATTO AS C, BOLLETTA AS B
WHERE     T.CODT = C.CODT
AND       C.CODU = B.CODU
GROUP BY  T.FASCIA, B.ANNO, B.TRIMESTRE
```

```
SELECT    FASCIA, ANNO, TRIMESTRE
FROM      V AS V1
WHERE     CONSUMO = (SELECT    MAX(CONSUMO)
                    FROM V AS V2
                    WHERE V2.FASCIA = V1.FASCIA)
```

- 2) Mostrare, per ogni città, il costo medio annuale (costo = consumo * prezzo) delle bollette relative a ciascuna fascia.

Q1: CREATE VIEW V AS

```
SELECT    B.CODU, B.ANNO, T.FASCIA,
          SUM(B.CONSUMO * T.PREZZO) AS COSTO
FROM      TARIFFA AS T, CONTRATTO AS C, BOLLETTA AS B
WHERE     T.CODT = C.CODT
AND       C.CODU = B.CODU
GROUP BY  B.CODU, B.ANNO, T.FASCIA
```

Q2: SELECT CITTA, ANNO, FASCIA, AVG(COSTO) AS COSTOMEDIO
FROM V, UTENZA
WHERE V.CODU = UTENZA.CODC
GROUP BY CITTA, FASCIA, ANNO

ORDER BY CITTA, FASCIA, ANNO

Exec sql Q1;

Declare Cursor "C1" For Q2

open C1;

fetch C1 into : CITTA, :ANNO, :FASCIA, :COSTOMEDIO;

while (SQLCODE == 0){

printf("CITTA %s, ANNO %s, FASCIA %s, COSTOMEDIO %f \n",

CITTA, ANNO, FASCIA, COSTOMEDIO);

fetch C1 into : CITTA, :ANNO, :FASCIA, :COSTOMEDIO;

}

close C1;

- 3) una pagina web che consente di inserire il valore di consumo che ne genera un'altra contenente l'elenco delle fasce che hanno registrato sempre consumi superiori al valore inserito.

Pagina form.html:

```
<html><head>
```

```
<title>Ricerca per Fasce</title>
```

```
</head>
```

```
<body bgcolor="white">
```

Inserire il consumo minimo:

```
<form action="find.jsp" method="get">
```

```
<!-- In alternativa:
```

```
<form action="find.aspx" method="get">-->
```

```
<table>
```

```
<tr><td>Consumo:</td>
```

```
<td><input type="text" name="consumo"></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",
    "postrgres", "postrgres");
//crea lo statement
Statement st = conn.createStatement();
String retrieve = " SELECT  T1.FASCIA
                    FROM    TARIFFA AS T1
                    WHERE   T1.FASCIA NOT IN
                            (SELECT T2.FASCIA
                             FROM  TARIFFA AS T2, CONTRATTO AS C2,
                                   BOLLETTA AS B2
                             WHERE T2.CODT = C2.CODT AND C2. CODU = B2.CODU
                                   AND  B2.CONSUMO <= ?)";

st = conn.prepareStatement(retrieve);
st.setString(1, Request.getParameter("consumo"));
ResultSet result = statement.executeQuery();
%>

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

```

<%
while (result.next()) {out.println("<tr><td>" + result.getString("FASCIA") +
"</td></tr>"

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

```

Pagina find.aspx:

```

<%@ Page Language="VB" %>
<html>
<head runat="server">
<title>Fascia</title>
</head>
<body>
<form id="form1" runat="server">
<asp:GridView ID="GridView1" Runat="server" DataSourceID="SqlDataSource1"
AutoGenerateColumns="False" HeaderText="Elenco forniture" AllowPaging="True">
<Columns>
<asp:BoundField HeaderText="Nome Scuola" DataField="CONSUMO "
SortExpression=" CONSUMO " />
</ Columns>
</ asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" Runat="server"
SelectCommand=" SELECT T1.FASCIA
FROM TARIFFA AS T1
WHERE T1.FASCIA NOT IN
(SELECT T2.FASCIA
FROM TARIFFA AS T2, CONTRATTO AS C2,
BOLLETTA AS B2

```

```
WHERE T2.CODT = C2.CODT AND C2. CODU = B2.CODU
AND B2.CONSUMO <= @Nconsumo)"
```

```
ConnectionString="<%%$ ConnectionStrings:bollette %>">
<SelectParameters>
<asp:QueryStringParameter Name="Nconsumo"
QueryStringField="consumo" />
</asp:SqlDataSource>
</form>
</body>
</html>
```

Esercizio 2

Dato il seguente schema relazionale:

R(A,B,C,D)

e considerando le seguenti dipendenze funzionali:

(FD1) $AB \rightarrow D$

(FD2) $BC \rightarrow A$

(FD3) $D \rightarrow C$

Le chiavi dello schema sono:

K1 = AB

K2 = BC

K3 = BD

(FD1) $AB \rightarrow D$ è in BCNF

(FD2) $BC \rightarrow A$ è in BCNF

(FD3) $D \rightarrow C$ è in 3NF

Lo schema è pertanto in 3NF.

Decomposizione binaria:

R1(D,C) preserva FD3 (proiettata)

R2(A,B,D) AK=BD, 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 FD2:

$XPIUG(B,C) = BC$

Pertanto FD2 non è preservata.

Esercizio 3

Dato la seguente porzione di schema relazionale:

PARTITA (CODS1, CODS2, RISULTATO)
CLASSIFICA (CODS, PUNTI)

Il risultato di una partita può avere i seguenti valori: 1 (vincitrice CODS1), 2 (vincitrice CODS2), X (pareggio). Per la vittoria sono assegnati 3 punti, per il pareggio 1, per la sconfitta nessuno.

Scrivere il Trigger (secondo la sintassi IBM DB2, MS SQLServer o ORACLE) che mantenga aggiornata la classifica.

```
CREATE TRIGGER Aggiorna_Classifica
AFTER INSERT, UPDATE, DELETE ON PARTITA
REFERENCING NEW AS N, OLD AS O
FOR EACH ROW MODE DB2SQL
BEGIN ATOMIC
    UPDATE CLASSIFICA
    SET PUNTI = PUNTI + 1
    WHERE N.RISULTATO = 'X'
    AND (CODS = N.CODS1 OR CODS = N.CODS2)

    UPDATE CLASSIFICA
    SET PUNTI = PUNTI + 3
    WHERE (N.RISULTATO = '1' AND CODS = N.CODS1)
    OR (N.RISULTATO = '2' OR CODS = N.CODS2)

    UPDATE CLASSIFICA
    SET PUNTI = PUNTI - 1
    WHERE N.RISULTATO = 'X'
```

AND (CODS = O.CODS1 OR CODS = O.CODS2)

UPDATE CLASSIFICA

SET PUNTI = PUNTI - 3

WHERE (O.RISULTATO = '1' AND CODS = N.CODS1)

OR (O.RISULTATO = '2' OR CODS = N.CODS2)

END