PROVA SCRITTA DI TECNOLOGIA DATABASE – 17/12/2007

Corso di Laurea Specialistica in Ingegneria Informatica - NOD

SONIA BERGAMASCHI E MAURIZIO VINCINI

Esercizio 1 (punti 22)

Dato il seguente schema relazionale:

```
MARCA(CODM,Nome,Nazione)
AUTO(CODA,CODM,Nome)
FK: CODM REFERENCES MARCA
NOLEGGIO(CODA,Data,Nazione,Giorni,Prezzo)
FK: CODA REFERENCES AUTO
```

Scrivere in SQL la seguente interrogazione

1) Mostrare, per ciascuna nazione, il numero medio di noleggi effettuati su auto di marca nazionale (cioè noleggi riferiti ad auto della stessa nazione).

Scrivere in embedded SQL la seguente interrogazione

2) Selezionare, per ogni marca, il nome dell'auto che ha garantito il fatturato complessivo maggiore.

Scrivere in linguaggio jsp o asp.net

3) una pagina web che consente di inserire una data ed una marca di auto e che ne genera un'altra contenente l'elenco delle auto noleggiate, il prezzo ed il numero di giorni riguardanti i noleggi avvenuti per la marca di auto specificata, a partire dalla data indicata dall'utente.

Esercizio 2 (punti 5)

Dato il seguente schema relazionale:

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

e considerando le seguenti dipendenze funzionali:

(FD1) $AB \rightarrow C$

(FD2) $C \rightarrow D$

(FD3) $D \rightarrow E$

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)
Data la seguente porzione di schema relazionale:

```
AUTO(CODA,Nome,PrezzoListino)
VENDITA(CODV,CODA,Cliente,Sconto,PrezzoVendita)
FK: CODA REFERENCES AUTO
```

Scrivere il Trigger (secondo la sintassi IBM DB2, MS SQLServer o ORACLE) che mantenga aggiornato il PrezzoVendita di un'auto, calcolata in base allo Sconto ottenuto sul PrezzoListino.

Soluzione

Esercizio 1

1) Mostrare, per ciascuna nazione, il numero medio di noleggi effettuati su auto di marca nazionale (cioè noleggi riferiti ad auto della stessa nazione).

```
CREATE VIEW V1 AS
SELECT      N.Nazione, COUNT(*) AS NUM_NOLEGGI
FROM        MARCA M, AUTO A, NOLEGGIO N
WHERE       M.CODM = A.CODM AND A.CODA = N.CODA
AND         N.Nazione=M.Nazione
GROUP BY    M.Nazione, M.CODM

SELECT      Nazione, AVG(NUM_NOLEGGI)
FROM        V1
GROUP BY    Nazione
```

Oppure

```
SELECT      Nazione, COUNT(*)/COUNT(DISTINCT(M.CODM))
FROM        MARCA M, AUTO A, NOLEGGIO N
WHERE       M.CODM = A.CODM AND A.CODA = N.CODA
AND         N.Nazione=M.Nazione
GROUP BY    M.Nazione
```

2) Selezionare, per ogni marca, il nome dell'auto che ha garantito il fatturato complessivo maggiore.

```
Q1:   SELECT      A.CODM, A.NOME
      FROM        AUTO A, NOLEGGIO N
      WHERE       A.CODA = N.CODA
      GROUP BY    A.CODM, A.CODA, A.NOME
      HAVING SUM(N.Prezzo) >= ALL (SELECT       SUM(N1.Prezzo)
                                   FROM         AUTO A1, NOLEGGIO N1
```

```
                                    WHERE       A1.CODA = N1.CODA
                                    AND         A1.CODM = A.CODM
                                    GROUP BY    A1.CODA)
```

```
Declare Cursor “C1” For  Q1
open C1;
fetch C1 into :CODM, :NOME;
while (SQLCODE == 0){
        printf(“CODM  %s, NOME%f\n”, CODM, NOME);
        fetch C1 into :CODM, :NOME;
}
close C1;
```

3) una pagina web che consente di inserire una data ed una marca di auto e che ne genera un’altra contenente l’elenco delle auto noleggiate, il prezzo ed il numero di giorni riguardanti i noleggi avvenuti per la marca di auto specificata, a partire dalla data indicata dall’utente.

Pagina form.html:

```
<html><head>
<title>Ricerca per Marca</title>
</head>
<body bgcolor="white">
Inserire il nome di un fornitore:
<form action="find.jsp" method="get">
<!-- In alternativa:
<form action="find.aspx" method="get">-->
<table>
<tr><td>Nome:</td>
<td><input type="text" name=“nome"></td></tr>
<tr><td>Data:</td>
<td><input type="text" name=“data"></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/azienda",
            “postrgres"," postrgres");
//crea lo statement
Statement st = conn.createStatement();
String retrieve = "SELECT   A.CODA, SUM(N.Prezzo) AS PREZZOTOT,
                            SUM(N.Giorni) AS NUMGIORNI
                FROM       AUTO A, NOLEGGIO N, MARCA M
                WHERE      A.CODA = N.CODA AND A.CODM = M.CODM
                AND        M.NOME=? AND N.Data>=?";
st = conn.prepareStatement(retrieve);
st.setString(1, Request.getParameter(“nome”));
st.setString(1, Request.getParameter(“data”));
ResultSet result = statement.executeQuery();
%>
<table>
```

```
<tr><td>Auto:</td><td>Costo Totale:</td><td>Giorni Totali:</td></tr>
<%
while (result.next()) {
        out.println("<tr><td>" + result.getString("CODA") + "</td><td>" +
                result.getString("PREZZOTOT")+ "</td><td>"
                result.getString("NUMGIORNI")+ "</td></tr>"
}
st.close();
conn.close();
%>
</table> </body> </html>
```

Pagina find.aspx:

```
<%@ Page Language="VB" %>
<html>
<head runat="server">
<title>Elenco noleggi</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="Auto" DataField="CODA" SortExpression="CODA" />
<asp:BoundField HeaderText="Costo Totale" DataField="PREZZOTOT"
                SortExpression="PREZZOTOT" />
<asp:BoundField HeaderText="Giorni Totali" DataField="NUMGIORNI"
                SortExpression="NUMGIORNI" />
</ Columns>
</ asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" Runat="server"
      SelectCommand=" SELECT       A.CODA, SUM(N.Prezzo) AS PREZZOTOT,
```

```
                    SUM(N.Giorni) AS NUMGIORNI
                FROM    AUTO A, NOLEGGIO N, MARCA M
                WHERE   A.CODA = N.CODA AND A.CODM = M.CODM
                AND     M.NOME=@Mnome AND N.Data>=@Ndata”
ConnectionString="<%$ ConnectionStrings:azienda %>">
<SelectParameters>
<asp:QueryStringParameter Name=“Mnome"
                    QueryStringField=“nome" />
<asp:QueryStringParameter Name=“Ndata"
                    QueryStringField=“data" />
</asp:SqlDataSource>
</form>
</body>
</html>
```

Esercizio 2

Dato il seguente schema relazionale:

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

e considerando le seguenti dipendenze funzionali:

(FD1) $AB \rightarrow C$

(FD2) $C \rightarrow D$

(FD3) $D \rightarrow E$

La chiave dello schema è:

K = AB

(FD1) $AB \rightarrow C$ è in BCNF

(FD2) $C \rightarrow D$ è in 2NF

(FD3) $D \rightarrow E$ è in 2NF

Lo schema è pertanto in 2NF.

Decomposizione binaria:

R1(D,E) preserva FD3 (proiettata)

R2(A,B,C,D) preserva FD1, FD2 (proiettate)

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

Lo schema non è in 3NF a causa di FD2, pertanto decomponiamo R2:

R21(C,D) preserva FD2 (proiettata)

R22(A,B,C) preserva FD1 (proiettata)

Lo schema è loss-less poiché il join naturale è fatto su C, chiave di R21.

Lo schema è BCNF.

Esercizio 3

Data la seguente porzione di schema relazionale:

```
AUTO(CODA,Nome,PrezzoListino)
VENDITA(CODV,CODA,Cliente,Sconto,PrezzoVendita)
FK: CODA REFERENCES AUTO
```

Scrivere il Trigger (secondo la sintassi IBM DB2, MS SQLServer o ORACLE) che mantenga aggiornato il PrezzoVendita di un'auto, calcolata in base allo Sconto ottenuto sul PrezzoListino.

```
CREATE TRIGGER Aggiorna_Prezzo_Vendita
ON VENDITA
FOR INSERT, UPDATE
AS
IF UPDATE(Sconto)
BEGIN
Declare @coda char(10)
Declare @prezzoListino float
Declare @sconto float

--- Dichiaro il cursore
DECLARE R_cursor CURSOR FOR
SELECT  I.CODA, PrezzoListino, Sconto
```

```
FROM Inserted I, AUTO A
WHERE I.CODA = A.CODA

--- Apro e carico il cursore
OPEN R_cursor
FETCH NEXT FROM R_cursor INTO @coda, @prezzoListino, @sconto

WHILE @@FETCH_STATUS = 0
BEGIN
    UPDATE VENDITA SET PrezzoVendita = (1-@sconto) * @prezzoListino
        WHERE CODA=@coda
    FETCH NEXT FROM R_cursor INTO @coda, @prezzoListino, @sconto
END
CLOSE R_cursor
DEALLOCATE R_cursor
END
```