

PROVA SCRITTA DI TECNOLOGIA DATABASE – 10/12/2008

Corso di Laurea Specialistica in Ingegneria Informatica - NOD

PROF. SONIA BERGAMASCHI

Esercizio 1 (punti 22)

Dato il seguente schema relazionale:

SCUOLA (CODS, NOME, GRADO)

CLASSE (CODC, CODS)

FK: CODS **REFERENCES** SCUOLA

ALUNNO (CF, CODC, VOTO_AMMISSIONE)

FK: CODC **REFERENCES** CLASSE

Scrivere in SQL la seguente interrogazione

- 1) Mostrare l'elenco delle scuole in cui la media di ciascuna classe calcolata sui voti di ammissione degli alunni è più alta della media dei voti di ammissione di tutti gli alunni (indipendentemente dalla scuola di appartenenza).

Scrivere in embedded SQL la seguente interrogazione

- 2) Selezionare, per ciascuna scuola, l'elenco degli alunni con voto di ammissione inferiore alla media della propria classe.

Scrivere in linguaggio jsp o asp.net

- 3) una pagina web che consente di inserire un grado di scuola ed un voto di ammissione che ne genera un'altra contenente l'elenco delle scuole aventi la media dei voti di ammissione superiore a quello indicato dall'utente.

Esercizio 2 (punti 5)

Dato il seguente schema relazionale:

R(A,B,C,D)

e considerando le seguenti dipendenze funzionali:

(FD1) $A \rightarrow B$

(FD2) $B \rightarrow D$

(FD3) $D \rightarrow C$

(FD4) $BC \rightarrow A$

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;
- 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:

FORNITURA(CODF, Data, Importo)

ARTICOLO(CODA, Nome, Prezzo)

CONTIENE(CODF, CODA, Quantita)

FK: CODF **REFERENCES** FORNITURA

CODA **REFERENCES** ARTICOLO

Scrivere il Trigger (secondo la sintassi IBM DB2, MS SQLServer o ORACLE) che mantenga aggiornato l'importo di una fornitura a fronte di inserimenti, aggiornamenti o cancellazioni nella relazione CONTIENE.

Soluzione

Esercizio 1

- 1) Mostrare l'elenco delle scuole in cui la media di ciascuna classe calcolata sui voti di ammissione degli alunni è più alta della media dei voti di ammissione di tutti gli alunni (indipendentemente dalla scuola di appartenenza).

```
CREATE VIEW V1 AS
SELECT      C.CODS, A.CODC,
            AVG (VOTO_AMMISSIONE) AS MEDIA_CLASSE
FROM        CLASSE C, ALUNNO A
WHERE       C.CODC = A.CODC
GROUP BY   C.CODS, A.CODC

SELECT      CODS
FROM        V1
GROUP BY   CODS
HAVING     MIN(MEDIA_CLASSE) > (SELECT AVG (VOTO_AMMISSIONE)
                                FROM ALUNNO)
```

- 2) Selezionare, per ciascuna scuola, l'elenco degli alunni con voto di ammissione inferiore alla media della propria classe.

```
Q1:  SELECT      A.CODS, A.CF
      FROM        CLASSE C, ALUNNO A
      WHERE       C.CODC = A.CODC
      AND         A.VOTO_AMMISSIONE <
                (SELECT      AVG (VOTO_AMMISSIONE)
                 FROM        ALUNNO A1
                 WHERE       A1.CODC = A.CODC)
```

Declare Cursor "C1" For Q1

```
open C1;
```

```
fetch C1 into :CODS, :CF;
```

```

while (SQLCODE == 0){
    printf("CODS %s, CF %s\n", CODS, CF);
    fetch C1 into :CODS, :CF;
}
close C1;

```

- 3) una pagina web che consente di inserire un grado di scuola ed un voto di ammissione che ne genera un'altra contenente l'elenco delle scuole aventi la media dei voti di ammissione superiore a quello indicato dall'utente.

Pagina form.html:

```

<html><head>
<title>Ricerca per Grado e Voto Ammissione</title>
</head>
<body bgcolor="white">
Inserire il grado di scuola:
<form action="find.jsp" method="get">
<!-- In alternativa:
<form action="find.aspx" method="get">-->
<table>
<tr><td>Nome:</td>
<td><input type="text" name="grado"></td></tr>
<tr><td>Voto:</td>
<td><input type="text" name="voto"></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/scuola",
    "postgres", "postgres");
//crea lo statement
Statement st = conn.createStatement();
String retrieve = "SELECT  S.CODS, S.NOME
                    FROM    SCUOLA S, CLASSE C, ALUNNO A
                    WHERE   S.CODS = C.CODS AND C.CODC = A.CODC
                    AND     S.GRADO=?
                    GROUP BY S.CODS
                    HAVING  AVG(A.VOTO_AMMISSIONE)>=?";
st = conn.prepareStatement(retrieve);
st.setString(1, Request.getParameter("grado"));
st.setString(1, Request.getParameter("voto"));
ResultSet result = statement.executeQuery();
%>
<table>
<tr><td>Codice Scuola:</td><td>Nome Scuola:</td></tr>
<%
while (result.next()) {out.println("<tr><td>" + result.getString("CODS") + "</td><td>" +
    result.getString("NOME")+ "</td></tr>")
}
st.close();
conn.close();
%>

```

```
</table> </body> </html>
```

Pagina find.aspx:

```
<%@ Page Language="VB" %>
<html>
<head runat="server">
<title>Elenco scuole</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="Codice Scuola" DataField="CODS"
SortExpression="CODS" />
<asp:BoundField HeaderText="Nome Scuola" DataField="NOME"
SortExpression=" NOME " />
</ Columns>
</ asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" Runat="server"
SelectCommand=" SELECT S.CODS, S.NOME
FROM SCUOLA S, CLASSE C, ALUNNO A
WHERE S.CODS = C.CODS AND C.CODC = A.CODC
AND S.GRADO=@Ngrado
GROUP BY S.CODS
HAVING AVG(A.VOTO_AMMISSIONE@Nvoto)"
ConnectionString="< '%$ ConnectionStrings:scuola %>">
<SelectParameters>
<asp:QueryStringParameter Name="Ngrado"
QueryStringField="grado" />
```

```
<asp:QueryStringParameter Name="Nvoto"
    QueryStringField="voto" />
</asp:SqlDataSource>
</form>
</body>
</html>
```

Esercizio 2

Dato il seguente schema relazionale:

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

e considerando le seguenti dipendenze funzionali:

(FD1) $A \rightarrow B$

(FD2) $B \rightarrow D$

(FD3) $D \rightarrow C$

(FD4) $BC \rightarrow A$

Le chiavi dello schema sono:

$K = A$

$K = B$

(FD1) $A \rightarrow B$ è in BCNF

(FD2) $B \rightarrow D$ è in BCNF

(FD3) $D \rightarrow C$ è in 2NF

(FD4) $BC \rightarrow A$ è in BCNF

Lo schema è pertanto in 2NF.

Decomposizione binaria:

$R_1(\underline{D},C)$ preserva FD3 (proiettata)

$R_2(\underline{A},B,D)$ $AK=B$, preserva FD1, FD2 (proiettate)

Lo schema è loss-less poiché il join naturale è fatto su D, chiave di R_1 .

Lo schema è in BCNF, vediamo la preservazione di FD4:

XPIUG (B,C) = ABCD

Pertanto anche FD4 è preservata.

Esercizio 3

Dato la seguente porzione di schema relazionale:

```
FORNITURA(CODF, Data, Importo)
ARTICOLO(CODA, Nome, Prezzo)
CONTIENE(CODF, CODA, Quantita)
FK:  CODF REFERENCES FORNITURA
      CODA REFERENCES ARTICOLO
```

Scrivere il Trigger (secondo la sintassi IBM DB2, MS SQLServer o ORACLE) che mantenga aggiornato l'importo di una fornitura a fronte di inserimenti, aggiornamenti o cancellazioni nella relazione CONTIENE.

```
CREATE TRIGGER Aggiorna_Fornitura
ON VENDITA
FOR INSERT, UPDATE, DELETE
AS
IF UPDATE(CODF, CODA, Quantita)
BEGIN
Declare @coda char(10)
Declare @codf char(10)
Declare @qty float

--- Dichiaro il cursore
DECLARE R_cursor CURSOR FOR
SELECT CODA, CODF, Quantita
FROM Inserted

--- Apro e carico il cursore
OPEN R_cursor
FETCH NEXT FROM R_cursor INTO @coda, @codf, @qty

WHILE @@FETCH_STATUS = 0
```



```

BEGIN
    UPDATE FORNITURA SET Importo = Importo +
        @qty * (SELECT Prezzo FROM ARTICOLO WHERE CODA = @coda)
    WHERE CODF=@codf
    FETCH NEXT FROM R_cursor INTO @coda, @codf, @qty
END
CLOSE R_cursor
DEALLOCATE R_cursor

--- Dichiaro il cursore
DECLARE R_cursor CURSOR FOR
SELECT CODA, CODF, Quantita
FROM Deleted

--- Apro e carico il cursore
OPEN R_cursor
FETCH NEXT FROM R_cursor INTO @coda, @codf, @qty

WHILE @@FETCH_STATUS = 0
BEGIN
    UPDATE FORNITURA SET Importo = Importo -
        @qty * (SELECT Prezzo FROM ARTICOLO WHERE CODA = @coda)
    WHERE CODF=@codf
    FETCH NEXT FROM R_cursor INTO @coda, @codf, @qty
END
CLOSE R_cursor
DEALLOCATE R_cursor

END

```