

SQL, JDBC og Servleter

Grundlæggende Programmering

Niels Hallenberg

IT-højskolen i København (IT-C)

27. november 2000

Resumé

Dette sæt opgaver fokuserer på servletter, SQL og Jdbc. Vi benytter en PostgreSQL database til at oprette simple databaser.

1 Opsætning

Til at starte med skal du følge informationerne på kursets hjemmeside, således at du kan oversætte og køre både servletter og almindelige java programmer som tilgår PostgreSQL databaser.

I dette opgavesæt benytter vi alle den samme database kaldet `gp`. For at benytte databasen `gp` skal du anvende brugernavn `gp` og kodeord `e2000`. Mere herom senere.

Da vi alle deler databasen `gp`, kan vi let komme ud for navne sammenfald. Vi anvender derfor følgende fremgangsmåde ved navngivning af tabeller: Du navngiver dine tabeller *tabelnavnlogin-navn*, dvs. hvis du har login-navn `foo`, da skal du navngive dine tabeller *tabelnavnfoo*, hvor *tabelnavn* er dit oprindelige tabelnavn.

1.1 PsqlFrontend

Med PostgreSQL følger et program `psql` hvor man kan indtaste og udføre SQL-udtryk på databasen `gp`. Programmet `psql` kører desværre ikke under Windows. Derfor har vi lavet et lille Java program hvori vi kan indtaste SQL-udtryk. Programmet startes fra URL

<http://www.it-c.dk/courses/GP/E2000/PsqlFrontend.html>

med Netscape (Explorer kan ikke vise appletten). Programmet har desværre ikke den mest overbevisende grænseflade, men programmet løser opgaven. For at udføre SQL-udtryk skal du etablere adgang til databasen `gp`. Dette gør du ved at trykke på knappen "Make New Connection", hvorefter der gerne skulle stå `0: New connection established.` i log-feltet nederst i vinduet. Log-feltet viser resultatet af dine handlinger, og du kan altid slette indholdet ved at klikke på knappen "Clear".

Knappen “Show All Tables” viser de tabeller der pt. er oprettet i databasen gp. Prøv at klikke på knappen. Øverst til højre vises en lille tabel med to kolonner. Kolonnen TABLE_NAME viser navnene på tabellerne i databasen gp.

Givet en tabel (angivet i feltet “Table Name”) kan vi henholdsvis se indholdet og definitionen af tabellen. Prøv at skrive student i feltet “Table Name” og klik på de to knapper. Knappen “Show Table Info” viser alle felter og deres typer.

Du udfører SQL-udtryk ved at indtaste dem i feltet “SQL statement” og derefter klikke på knappen “Do SQL”. Bemærk, at retur tasten skifter linie i feltet i stedet for at udføre SQL-udtrykket. Knappen “Clear” sletter indholdet af feltet “SQL statement”. Prøv at taste

```
SELECT * FROM student;
```

og klik på knappen “Do SQL”. Du må gerne skrive SQL-udtryk med både store og små bogstaver.

Til højre for feltet “SQL statement” findes knapper som udvider feltet “SQL statement” med en skabelon svarende til det SQL-udtryk du vælger. Bemærk, at SQL-udtryk kan være meget mere komplicerede end skabelonerne lægger op til.

2 SQL

Vi vil nu oprette tabeller svarende til studiedatabasen gennemgået til forelæsningsen. Databasen består af tre tabeller med følgende indhold:

Student

<i>efternavn:TEXT</i>	<i>fornavn:TEXT</i>	<i>studienummer:TEXT</i>
Olsen	Peter	L2143
Hansen	Erika	j00007
Funder	Ulrik	Hg0014

Kursus

<i>kursusnummer:INT</i>	<i>kursusnavn:TEXT</i>
10181	Databehandling
45621	Landbrugszoologi
15351	Miljømodeller
15311	Matematisk Grundkursus

Eksamen

<i>studienummer:TEXT</i>	<i>kursusnummer:INT</i>	<i>karakter:INT</i>
L2143	10181	8
L2143	45621	9
J00007	10181	11
J00007	15311	8
L2143	15311	10
Hg0014	15311	7

2.1 Tabeloprettelse

Opret de tre tabeller med SQL-udtrykket `CREATE`. Typerne på hvert felt er angivet efter feltets navn. Det første tabel oprettes med følgende SQL-udtryk:

```
CREATE TABLE student (efternavn TEXT, fornavn TEXT, studienummer TEXT);
```

Prøv knapperne “Show Table” og “Show Table Info” for en eller flere tabeller når de er oprettet.

2.2 Indsættelse af værdier

Indsæt værdier svarende til de tre tabeller givet ovenfor. Du skal benytte SQL-udtrykket `INSERT`. Den første række i tabellen `student` indsættes således:

```
INSERT INTO student VALUES ('Olesen', 'Peter', 'L2143');
```

Indsæt også data for dig selv i tabellerne.

2.3 Udtræk

Benyt `SELECT` udtrykket til at vise tabellen `kursus`:

```
SELECT * FROM kursus;
```

Vis alle kurser sorteret efter `kursusnummer` uden dubletter (benyt `DISTINCT`).

Vis alle studerende sorteret efter karakter.

Vis studienummer på de studerende, der har fået en karakter større end 8.

Vis fornavn og efternavn på de studerende, der har fået en karakter mellem 6 og 9.

Vis fornavne og karakter for studerende, der har haft kurset “Matematisk Grundkursus”.

2.4 Opdatering

Benyt `UPDATE` til at lægge fire til alle karakterer i tabellen `eksamen`:

```
UPDATE eksamen SET karakter = karakter + 4;
```

Sæt alle karakterer der er over 13 til 13.

2.5 Sletning

Slet alle eksaminer fra tabellen `eksamen`, hvor karakteren er under 8.

```
DELETE FROM eksamen WHERE karakter < 8;
```

Slet alle karakterer der er lig 13.

Slet alle eksaminer for Peter Olesen.

2.6 Aggregerede Udtræk

Vis antal poster i tabellen `kursus` (benyt `COUNT`).

Find alle studerende som har mere end et kursus.

3 JDBC

Eksempel på et simpelt udtræk fra student tabellen ses nedenfor:

```
import java.io.*;
import java.sql.*;
public class Studie1DB {
    public static void main(String[] args) throws SQLException {
        Connection db;           // The connection to the database
        Statement st;           // Our statement to run queries with

        String url = "jdbc:postgresql://ellemose.dina.kvl.dk/gp";
        String usr = "gp";
        String pwd = "e2000";

        try {
            Class.forName("postgresql.Driver");
        } catch (ClassNotFoundException e) {
            System.out.println("Cannot find the Postgresql driver");
        }

        // Connect to database
        db = DriverManager.getConnection(url, usr, pwd);

        st = db.createStatement();

        // Read from database
        String query = "SELECT efternavn, fornavn, student.studienummer, " +
            "karakter FROM student,eksamen " +
            "WHERE student.studienummer = eksamen.studienummer " +
            "ORDER BY karakter";
        boolean resultset = st.execute(query);
        if (resultset) {           // Results were produced by a SELECT statement
            ResultSet rs = st.getResultSet();
            while (rs.next()) {
                String enavn = rs.getString("efternavn");
                String fnavn = rs.getString("fornavn");
                String stdnr = rs.getString("studienummer");
                String karnr = rs.getString("karakter");
                System.out.println(fnavn + " " + enavn + " (" + stdnr + ", " + karnr + ")");
            }
        }
        st.close();
        db.close();
    }
}
```

Lav et java program Studie2.java, som indsætter flere poster i tabellerne eksamen og kursus (Du kan kopiere Studie1DB.java fra hjemmesiden med projekter. Programmet skal udskrive indholdet af en eller flere tabeller.

Lav et program `Studie3.java`, som udskriver navnene på de studerende der ikke har været til eksamen.

4 Servleter

Det antages, at du har fået adgang til et katalog (`grp`), hvor du kan gemme og udføre servletter.

Servletter kan henholdsvis reagere på typerne `GET` og/eller `POST`. Servletten nedenfor returnerer et tilfældigt tal mellem 1 og 100 hver gang den kaldes.

```
// Randomservlet.java

import java.io.*;
import java.util.Random;
import javax.servlet.*;
import javax.servlet.http.*;

public class Randomservlet
    extends HttpServlet implements SingleThreadModel {

    private Random rgen = new Random();
    private int[] outcome = new int[1000];
    private int howmanyrequests = 0;

    public void service(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {
        res.setContentType("text/html");
        ServletOutputStream out = res.getOutputStream();

        out.println("<HTML><BODY>");
        out.println("<H2>Random number server</H2>");
        int result = (int)(1 + 6 * rgen.nextDouble());
        outcome[howmanyrequests] = result;
        howmanyrequests++;
        out.println("The outcome was <B>" + result + "</B>.");
        out.println("<P>The die has been thrown " + howmanyrequests + " times.");
        out.println("<P>The previous results were:");
        out.println("<P><TABLE BORDER><TR><TH>Throw<TH>Outcome");
        for (int i=0; i<howmanyrequests; i++)
            out.println("<TR><TD align=right>" + (i+1) + "<TH>" + outcome[i]);
        out.println("</TABLE>");
        out.println("</BODY></HTML>");
    }
}
```

Du kan kopiere programmet fra hjemmesiden med projekter (søg efter Forelæsningsens eksempler). Omdøb navnet på klassen og filen således at dit login-navn sættes bagpå. Oversæt programmet og kopier resultatet ned i `grp`-kataloget. Du deler eventuelt `grp`-kataloget med andre, så vi skal være sikre på at vi ikke

benytter samme navn for vores klasse filer. Du kan nu afprøve programmet ved at skrive den url du har fået oplyst i forbindelse med `grp` kataloget.

Hvis du anvender "Reload" knappen i Netscape, vil du se at servletten husker det antal gange den har været kaldt.

Lav en servlet `GetKursuslogin`, som anvender GET og returnerer indholdet af kursus tabellen i databasen `gp`. Du kan evt. formatere indholdet af kursus tabellen som en tabel i HTML. En HTML tabel med to kolonner laves således:

```
<TABLE BORDER><TR><TH>Overskrift 1</TH><TH>Overskrift 2</TH></TR>
<TR><TD> Data11 </TD><TD> Data12</TD></TR>
...
<TR><TD> DataN1 </TD><TD> DataN2</TD></TR>
</TABLE>
```

Lav en servlet `GetTablefoo`, som anvender POST, og returnerer indholdet af to felter i en tabel som brugeren indtaster. Udover tabellen, skal brugeren indtaste de to feltnavne som findes i tabellen.

Du kan anvende følgende FORM-kode:

```
<P><FORM ACTION="http://mysql.it-c.dk/grp/GetTable" METHOD=POST>

<P>Table: <INPUT TYPE=TEXT NAME=table>

<P>First field: <INPUT TYPE=TEXT NAME=field1>
<P>Second field: <INPUT TYPE=TEXT NAME=field2>

<P><INPUT TYPE=SUBMIT VALUE="Get Table Info">
</FORM>
```

hvor `grp` erstattes af den gruppe du er tildelt. I dit Java program `GetTableloginnavn` kan du bruge følgende kode til at finde tabel og de to felt navne:

```
res.setContentType("text/html");
String[] tableParms = req.getParameterValues("table");
String table = tableParms[0];
String[] field1Parms = req.getParameterValues("field1");
String field1 = field1Parms[0];
String[] field2Parms = req.getParameterValues("field2");
String field2 = field2Parms[0];
```

hvor `res` er bundet til et `HttpServletResponse` objekt.

Lav en servlet `InsertKursusfoo`, som indsætter en række i kursus tabellen. Servletten skal anvende POST.

5 Appletter

Kopier filerne `StudieApplet.java` og `StudieApplet.html` fra hjemmesiden med projekter. Oversæt Java filen og indlæs HTML filen i Netscape.

Prøv at ændre programmet `GetTable` til at fungere som en applet med indtastningsfelter og et uddatafelt, hvor du kan skrive indholdet af den tabel du forespørger på.