SQL Programming Lab

CS442 SQL Programming Lab Sessions

- SQL programming lab sessions
  - 2 sessions: this Thursday (11/10) and next Tuesday (11/15)
  - Lab instructions and skeleton code are available in Canvas

Main Tasks of Lab

JDBC programming for database applications

1. Database schema and instances are given.
2. Create the schema of the tables by SQL and JDBC.
3. Insert records into the tables by SQL and JDBC.
4. Execute five SQL queries on the tables.
5. Skeleton code is provided

Lab Management

- Each lab session will have at least 1 TA for in-person assistantship.
  - A PhD student helper will be present at Thursday’s 12:30pm section A.
- TA on-duty schedule can be found in lab instructions.

Hand-in Policy

- The students need to accomplish both tasks below
  - Upload the code in Canvas by Nov 15
  - demonstrate code to a TA either in-person in lab or during TA’s office hour by Nov 15
  - TA will ask random questions about the code during demonstration
- Both code upload and demonstration are required.
- Email TA to make appointment for demonstration if you can attend neither the class nor any TA office hour.

Programming Language

- Java is preferred
- You can use other languages too (Python, C++, etc.)
  - TAs may not be able to help you for these languages.

JDBC Programming

Pure SQL

- Pure SQL: Queries typed at an SQL prompt.
  – SQL is a non-procedural language.
  – Sophisticated applications are often implemented by using SQL + a programming language.

Embedded SQL

- SQL can be embedded within procedural programming languages (C/C++, Java, Perl, Python, and PHP).
- Embedded SQL supports:
  - Customized applications.
  - Background applications running without user intervention.
  - Combining database tools with programming tools.
  - Web-based databases.

Two types of embedding (1/2)

Type 1: Low-level embedding (eg. C/C++):

- SQL and program compiled into a single executable.
- Very efficient link.

Two types of embedding (2/2)

Type 2: ODBC - Open Database Connectivity (eg. PHP/Java):

- SQL queries are sent from the program to the database as strings.
- Results returned as an array or a list.
- Independence of program and database:
  - Each language has one DBI (database interface) for all DBMS types (For example, JDBC for Java.)
  - Separate database drivers (DBD) for each DBMS type.

JDBC

- Part of Java, very easy to use
- Java comes with a JDBC-to-ODBC bridge
  - So JDBC code can talk to any ODBC data source
- JDBC tutorial online
  - http://developer.java.sun.com/developer/Books/JDBCTutorial/

Basic steps to use a database in Java

0. Download and install software
1. Import the package
2. Establish a connection to the database
3. Create JDBC Statements
4. Execute SQL Statements
5. Get query results
6. Close connections

0. Download and Install Software

- Download and install MySQL JDBC Driver
  - https://www.youtube.com/watch?v=2i4t-SL1VsU&list=PLEAQNNR8IlB4R7NfqBY1frapYo97L6fOQ&index=2&t=310s

1. Import the package

- Include the package that contains JDBC classes needed for database programming.
- Most often, using *import java.sql.** will suffice.

2. Establish Connections

- A Connection is an object representing a login to a database

```
// GET CONNECTION
Connection con;
try {
    con = DriverManager.getConnection(
         "jdbc:odbc:testDB",
         userName,password);
} catch(Exception e){
 System.out.println(e);  }
```

3. Create JDBC Statements

- You need a Statement object for each SQL statement

```
// CREATE STATEMENT
Statement stmt;
try {
    stmt = con.createStatement();
} catch (Exception e){
    System.out.println(e);
}
```

Soon we’ll say stmt.executeQuery(“select …”);

4. Execute JDBC statements

- A ResultSet object serves as a *cursor* for the statement’s results (stmt.executeQuery())

```
// EXECUTE QUERY
ResultSet results;
try {
    results = stmt.executeQuery(“select * from Students")
} catch (Exception e){
    System.out.println(e);   }
```

- Obvious handy methods:
  - results.next() advances cursor to next tuple
    - Returns “false” when the cursor slides off the table (beginning or end)
  - “scrollable” cursors:
    - results.previous(), results.relative(int), results.absolute(int), results.first(), results.last(), results.beforeFirst(), results.afterLast()

5. Get ResultSet (If you know the schema)

```
String querySailor = "select * from
  Students";

ResultSet rs =
  Stmt.executeQuery(querySailor);

while (rs.next()) {
  int sid = rs.getInt(“SID");
  String name = rs.getString(“SNAME");
  int age = rs.getInt(“AGE");
}
```

5. Get ResultSet
(If you don’t know the schema)

- Can find out stuff about the ResultSet schema via ResultSetMetaData

```
ResultSetMetaData rsmd = results.getMetaData();
int numCols = rsmd.getColumnCount();
int i, rowcount = 0;

// get column header info
for (i=1; i <= numCols; i++){
    if (i > 1) buf.append(",");
    buf.append(rsmd.getColumnLabel(i));
}
buf.append("\n");
```

- Other ResultSetMetaData methods:
  - getColumnType(i), isNullable(i), etc.

6. Close Connections

```
try {
  // CLOSE RESULT SET
  results.close();
  // CLOSE STATEMENT
  stmt.close();
  // CLOSE CONNECTION
  con.close();
} catch (Exception e) {
    System.out.println(e);
}
```

Putting it Together (w/o try/catch)

```
Connection con =
  DriverManager.getConnection("jdbc:odbc:testDB",
  userName,password);
Statement stmt = con.createStatement();
ResultSet rs =
   stmt.executeQuery("select * from Students")

while (rs.next()) {
  int sid = rs.getInt(“SID");
  String name = rs.getString(“SNAME");
  int age = rs.getInt(“AGE");
}
results.close(); stmt.close();  con.close();
```

During Lab

- To-do items
  - Install software (see lab instructions; can do it in the lab)
  - Understand the skeleton code
  - Finish to-do list in the lab instructions
  - Demonstration to the TAs
  - Upload code in Canvas

Recommended Resources

- JDBC tutorials (with step-by-step demonstration)
  - Java JDBC tutorial (Playlist):
    - https://www.youtube.com/playlist?list=PLEAQNNR8IlB4R7NfqBY1frapYo97L6fOQ
  - Java Connect to MySQL Database Step by Step
    - https://www.youtube.com/watch?v=duEkh8ZsFGs