ICCS240 Database Management

Programming with DB

Many slides in this lecture are either from or adapted from slides provided by

Kazuhiro Minami. UIUC

Werner Nutt, Free U of Bozen-Bolzano

More ways to make SQL calls from outside the DBMS

DB Access from a Programming Language

Approaches:

- Embedding SQL into programming language

  "Embedded SQL" for C/C++

- DB access via API (or Call-Level Interface: CLI)

  JDBC, …

Approach 1: Embedded SQL

- SQL code occurs in program, separated by markers:

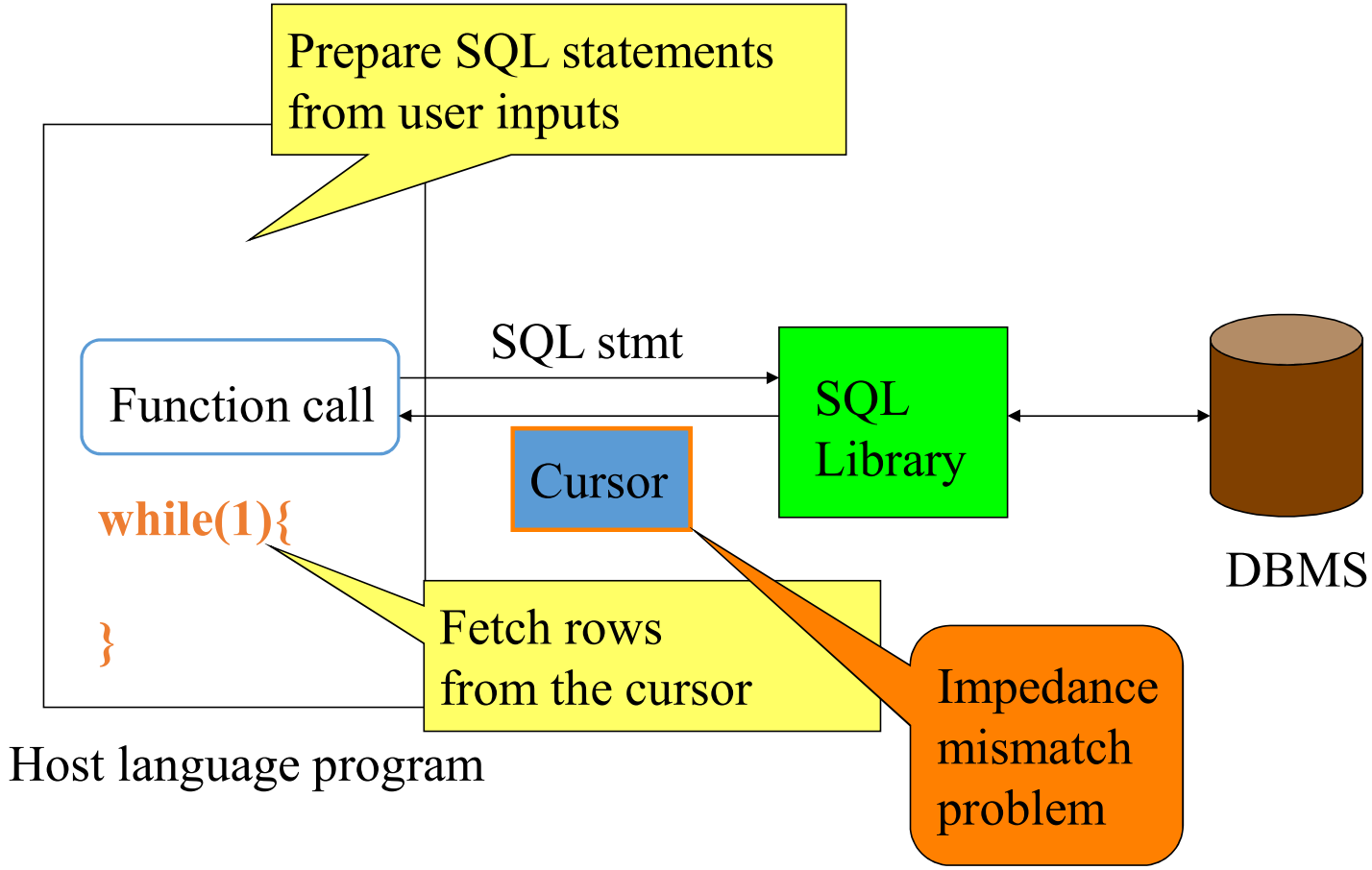
```
EXEC SQL  SELECT ranking INTO :r
          FROM sailors
          WHERE sailors.sid = 15765;
r++;
EXEC SQL  UPDATE sailors S
          SET ranking = :r
          WHERE sailors.sid = 15765;
```

- Transfer of values between PL and SQL:

  use of host language variables in SQL (prefixed with ":")

- Compilation in 2 steps:
  1. Preprocessor translate SQL fragments into function calls of SQL runtime library
  2. Regular compiler for C/C++ produces executable

Approach 2: SQL/Host Language Interface (CLI)



MySQL + PHP

PHP (Hypertext Preprocessor)

- A server-side scripting language
- Supports many databases (MySQL, Oracle, …)
- PHP file contains text, HTML tags and scripts
- PHP scripts are executed on the server
- PHP files are returned to a browser as plain HTML

hello_world.php

```
<html><body>
<?php
    $host = 'some_address.cs.edu';
    $user = 'username'; $password = 'password';
    $link = mysql_connect($host, $user, $password)
                or die ('Failed to connect:'.mysql_error());
    mysql_select_db('db_name') or die ('Could not select database<br>');
    $query = 'SELECT * FROM hello_table';
    $result = mysql_query($query);
    while ($row = mysql_fetch_array($result)) {
        echo "$row[message]<br>";
    }
    mysql_free_result($result);
    mysql_close($link);
?>
</body></html>
```

JDBC

These slides are partly based on learning material provided by SUN Microsystems at http://java.sun.com/docs/books/tutorial/jdbc/TOC.html.

What is JDBC?

- JAVA API to talk to database via SQL
  - Can query, update + fetch results
  - Can retrieve metadata
  - Aside from the command being sent, the interface is pretty uniform across DBs

- Model for communicating with the database
  - Open a connection (Connection)
  - Create a “statement” object
  - Execute queries using the Statement object to send queries and fetch results (ResultSet)
  - Exception mechanism to handle errors

Schema of JDBC Application

- Load the driver for a specific DBMS
  (e.g., the PostgreSQL “JDBC Driver”)
- Establish a connection to a specific database
  (e.g., the PostgreSQL database `wdb` on the server database.inf.unibz.it)
- Create an abstract statement, to be sent over the connection
- Execute the statement by sending a Java string
  (e.g., "`SELECT ranking FROM sailors WHERE sid = 15765`")
  returns an object of class `ResultSet`
- Process the result set with methods of `ResultSet`
- Close statement and connection

JDBC Code

```
public static void JDBCexample(String dbid, String userid, String
  passwd)
   {
      try (Connection conn = DriverManager.getConnection(
          "jdbc:oracle:thin:@db.yale.edu:2000:univdb", userid, passwd);
             Statement stmt = conn.createStatement();
        )
      {
             … Do Actual Work ….
      }
     catch (SQLException sqle) {
         System.out.println("SQLException : " + sqle);
   }
  }
```

JDBC Code (cont.)

- Update to database

```
try {
     stmt.executeUpdate("insert into instructor
                                 values('77987', 'Kim', 'Physics', 98000)");
} catch (SQLException sqle)
{
    System.out.println("Could not insert tuple. " + sqle);
}
```

- Execute query and fetch and print results

```
ResultSet rset = stmt.executeQuery("select dept_name, avg (salary)
                                        from instructor group by dept_name");
while (rset.next()) {
    System.out.println(rset.getString("dept_name")+" "+rset.getFloat(2));
}
```

Some details

- Getting result fields

  `rs.getString("dept_name")` and `rs.getString(1)`
  equivalent if `dept_name` is the first argument of select result.

- Dealing with NULL values

```
int a = rs.getInt("a");
if (rs.wasNull())
   Systems.out.println("Got null value");
```

Prepared Statement

```
PreparedStatement pStmt = conn.prepareStatement(
            "insert into instructor values(?,?,?,?)");
pStmt.setString(1, "88877");
pStmt.setString(2, "Perry");
pStmt.setString(3, "Finance");
pStmt.setInt(4, 125000);
pStmt.executeUpdate();
pStmt.setString(1, "88878");
pStmt.executeUpdate();
```

WARNING: always use prepared statements when taking an input from the user and adding it to a query

- NEVER create a query by concatenating strings
- **"insert into instructor values('"+ID+"','"+name+"','+dept_name+"','"+balance+" ')"**
- What if name is "D', 'Souza"?

SQL Injection

- Suppose query is constructed using

```
"select * from instructor where name = '" + name + "'"
```

- Suppose the user, instead of entering a name, enters:

```
X' or 'Y' = 'Y
```

- then the resulting statement becomes:

```
"select * from instructor where name = '" + "X' or 'Y' = 'Y" + "'"
```

  which is:

```
select * from instructor where name = 'X' or 'Y' = 'Y'
```

  User could have even used

```
X';  update instructor set salary = salary + 10000; --
```

- Prepared statement internally uses:

```
"select * from instructor where name = 'X\' or \'Y\' = \'Y\'"
```

  - **Always use prepared statements, with user inputs as parameters**

Metadata

ResultSet metadata

e.g. after executing query to get a `ResultSet rs`:

```
ResultSetMetaData rsmd = rs.getMetaData();
for(int i = 1; i <= rsmd.getColumnCount(); i++) {
      System.out.println(rsmd.getColumnName(i));
      System.out.println(rsmd.getColumnTypeName(i));
}
```

Your Playtime

Try “HelloWorld.java” (JDBC demo)

Using db info provided in last lecture: **sqlex_ddl3.sql sqlex_data3sql**

Make your request through JDBC. For each question, write a single (possibly nested) query to find out the answer.

- Find the IDs of all students who were taught by an instructor named Einstein; make sure there are no duplicates in the result
- Find the highest salary of any instructor
- For each department, find the highest salary of any instructor
- Find the maximum enrollment, across all sections, in Fall 2009
- Find the sections that had the maximum enrollment in Fall 2009
- Find the IDs and names of all students who have not taken any course offering before Fall 2009
- Find all students who have taken all courses offered in the Biology department.