

CS360 Lecture 19

Java and Databases

Tuesday, April 11, 2004

Reading

Chapter 23

MySQL

After creating your tables, if you forgot to set the primary key, do the following:
`ALTER TABLE profs MODIFY id INT(2) PRIMARY KEY;`

Java with MySQL

The following Java program connects to the database we created last time (profs, courses, students, enrolment) and displays all the professor names.

```
import java.sql.*;
import java.util.*;
import javax.swing.*;
import java.awt.*;

public class DisplayProfs
{
    // JDBC driver name and database URL
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DATABASE_URL = "jdbc:mysql://localhost/csdepartment";

    // declare Connection and Statement for accessing
    // and querying database
    private Connection connection;
    private Statement statement;

    // constructor connects to database, queries database, processes
    // and displays results
    public DisplayProfs()
    {
        // connect to database and query database
        try
        {
            // specify location of database on filesystem
            System.setProperty( "db2j.system.home", "/Library/MySQL/bin/" );

            // load database driver class
            Class.forName("com.mysql.jdbc.Driver");
```

```

// establish connection to database
connection = DriverManager.getConnection( DATABASE_URL, "root",
                                         "jason" );

// create Statement for querying database
statement = connection.createStatement();

// query database
ResultSet resultSet =
    statement.executeQuery("SELECT firstName, lastName FROM profs");

// process query results
ResultSetMetaData metaData = resultSet.getMetaData();
int numberOfColumns = metaData.getColumnCount();

while ( resultSet.next() )
{
    for ( int i = 1; i <= numberOfColumns; i++ )
        System.out.print( resultSet.getObject( i ) + "\t" );

    System.out.println();
}
} // end try

// detect problems interacting with the database
catch ( SQLException sqlException )
{
    System.out.println( sqlException.getMessage() );
    System.exit( 1 );
}

// detect problems loading database driver
catch ( ClassNotFoundException classNotFound )
{
    System.out.println( classNotFound.getMessage() );
    System.exit( 1 );
}

// ensure statement and connection are closed properly
finally
{
    try
    {
        statement.close();
        connection.close();
    }
}

```

```

    }

    // handle exceptions closing statement and connection
    catch ( SQLException sqlException )
    {
        System.out.println( sqlException.getMessage() );
        System.exit( 1 );
    }
}

} // end DisplayAuthors constructor

// launch the application
public static void main( String args[] )
{
    DisplayProfs window = new DisplayProfs();
}

} // end class DisplayProfs

```

The above program will connect to the database csdepartment and execute the SQL query. It will print out the names of all professors in the profs table and separate each record with a new line.

What if we wanted to print out the column names as well as the contents of the columns?

We need to get the database metadata:

```
ResultSetMetaData metaData = resultSet.getMetaData();
```

By getting hold of the database metadata, we can display all kinds of database information.

To print out the column names we would use:

```
int numberOfColumns = metaData.getColumnCount();

for ( int i = 1; i <= numberOfColumns; i++ )
    System.out.println( metaData.getColumnName( i ) + "\t" );
```

The following program display some basic database information (database name, list of tables) then accepts SQL queries from the user.

```
import java.sql.*;
import java.util.*;
import java.io.*;
```

```

public class QueryCSDepartment
{
    // JDBC driver name and database URL
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DATABASE_NAME = "csdepartment";
    static final String DATABASE_URL =
        "jdbc:mysql://localhost/" + DATABASE_NAME;

    // declare Connection and Statement for accessing
    // and querying database
    private Connection connection;
    private Statement statement;

    public QueryCSDepartment()
    {
        // connect to database and query database
        try
        {
            // specify location of database on filesystem
            System.setProperty( "db2j.system.home", "/Library/MySQL/bin/" );

            // load database driver class
            Class.forName("com.mysql.jdbc.Driver");

            // establish connection to database
            connection=DriverManager.getConnection(DATABASE_URL,"root", "jason");

            // create Statement for querying database
            statement = connection.createStatement();

            BufferedReader parsedInput = new BufferedReader(
                new InputStreamReader(System.in));
            String inputLine;

            // get database meta data
            DatabaseMetaData databaseMetaData = connection.getMetaData();

            // display welcome message
            System.out.println( "Enter SQL queries on " + DATABASE_NAME
                + ". Type 'bye' to quit" );
            System.out.println( "Database " + DATABASE_NAME
                + " contains the following tables: " );

            // get list of tables

```

```

ResultSet tables = databaseMetaData.getTables( DATABASE_NAME, "%",
                                                "%", (String []) null );

// display list of tables
while( tables.next() )
    System.out.println( tables.getString( "TABLE_NAME" ) );

System.out.print( "\nSQL Query>" );

while( !(inputLine = parsedInput.readLine()).equals("bye") )
{
    System.out.println( inputLine );

    // query database
    ResultSet resultSet = statement.executeQuery( inputLine );

    // get query result meta data
    ResultSetMetaData metaData = resultSet.getMetaData();
    int numberOfColumns = metaData.getColumnCount();

    // display column names
    for ( int i = 1; i <= numberOfColumns; i++ )
        System.out.print( metaData.getColumnName( i ) + "\t" );

    System.out.println( "\n-----" );

    // display column contents
    while ( resultSet.next() )
    {
        for ( int i = 1; i <= numberOfColumns; i++ )
            System.out.print( resultSet.getObject( i ) + "\t\t" );

        System.out.println();
    }

    System.out.print( "SQL Query>" );
} // end try

// detect problems interacting with the database
catch ( SQLException sqlException )
{
    System.out.println( sqlException.getMessage() );
    System.exit( 1 );
}

```

```

// detect problems loading database driver
catch ( ClassNotFoundException classNotFound )
{
    System.out.println( classNotFound.getMessage() );
    System.exit( 1 );
}

catch( IOException ioException )
{
    System.out.println( ioException.getMessage() );
    System.exit( 1 );
}

// ensure statement and connection are closed properly
finally
{
    try
    {
        statement.close();
        connection.close();
    }

    // handle exceptions closing statement and connection
    catch ( SQLException sqlException )
    {
        System.out.println( sqlException.getMessage() );
        System.exit( 1 );
    }
}

} // end DisplayAuthors constructor

// launch the application
public static void main( String args[] )
{
    QueryCSDepartment window = new QueryCSDepartment();
}
}

```

Exam Topics

Threads

Networking: TCP – UDP

JDBC

Basic object oriented design