

Java Database Connectivity (JDBC)

Java Database Connectivity (JDBC)

- JDBC: API that allows java to communicate with a database server using SQL commands.
 - To use it do: `import java.sql.*`
- Most important:
 - **Connection**
 - **Statement, PreparedStatement**
 - **ResultSet**
- They are **interfaces** instead of classes.
 - Because the point of JDBC is to hide the specifics of accessing a particular database.
 - Implementation of the underlying classes is done in the **vendor** provided driver and associated classes.

Basics

```
import java.sql.*;
class InsertMovie
{
    public static void main (String args []) throws SQLException
    {
        DriverManager.registerDriver (new oracle.jdbc.driver.OracleDriver());
        Connection conn = DriverManager.getConnection
            ("jdbc:oracle:thin:@localhost:1522:studentdb", "userid", "password");
        // @machineName:port:SID,  userid, password
    }
}
```

DriverManager
is responsible
for keeping
track of JDBC
drivers available
on a system.

Specify the database to connect with a **jdbc:URL**.
This URL has the following general syntax:
jdbc:subprotocol:host.port.databasename

Creating JDBC Statements

- A **Statement** object is what sends your SQL statement to the DBMS.
 - Create a **Statement** object, then execute it:
 - Connection needed to create a **Statement** object.
Statement stmt = conn.createStatement();
 - For SELECT statements, use **executeQuery**.
 - For statements that create or modify tables, use **executeUpdate**.

```
import java.sql.*;
```

```
class InsertMovie {
```

```
    public static void main (String args []) throws SQLException
```

```
{
```

```
    DriverManager.registerDriver (new oracle.jdbc.driver.OracleDriver());
```

```
    Connection conn = DriverManager.getConnection
```

```
        ("jdbc:oracle:thin:@localhost:1522:studentdb", "userid", "password");
```

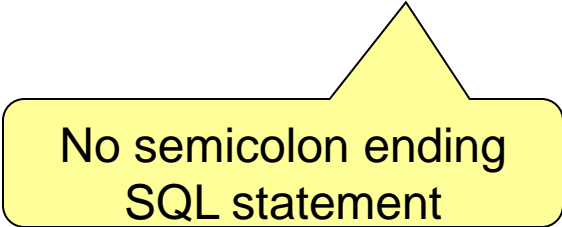
```
    Statement stmt = conn.createStatement();
```

```
    stmt.executeUpdate("INSERT INTO Movies VALUES('ABC',2004,200,'Disney')");
```

```
    stmt.close();
```

```
}
```

```
}
```



No semicolon ending
SQL statement

Getting Data

Example

```
ResultSet rset = stmt.executeQuery( "SELECT title, year FROM Movies");
```

```
while (rset.next()) {  
    ...  
}
```

- Variable **rset**, contains the rows of the query result.
- The first call of **next()** positions a "cursor" on the first row.
- Successive invocations of **next()** move the cursor down one row at a time.

Using the getXXX methods

- Use the **getXXX** method of the appropriate type to retrieve the value in each column.
 - **getString()** for VARCHAR, CHAR
 - **getInt()** for INT, NUMBER
 - etc.

```
while (rset.next())
```

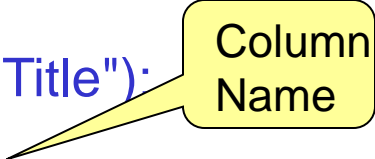
```
{
```

```
    String s = rset.getString("Title");
```

```
    int n = rset.getInt("Year");
```

```
    System.out.println(s + " " + n);
```

```
}
```

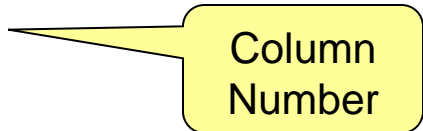


Column
Name

Or use column number instead:

```
String s = rset.getString(1);
```

```
int n = rset.getInt(2);
```



Column
Number

```
import java.sql.*;

class dbAccess {
    public static void main (String args []) throws SQLException
    {
        DriverManager.registerDriver (new oracle.jdbc.driver.OracleDriver());

        Connection conn = DriverManager.getConnection
            ("jdbc:oracle:thin:@localhost:1522:studentdb", "userid", "password");

        Statement stmt = conn.createStatement();
        ResultSet rset = stmt.executeQuery("SELECT title, year FROM Movie");
        while (rset.next())
            System.out.println (rset.getString("title") + " " + rset.getString("year"));

        stmt.close();
    }
}
```


PreparedStatement

- When we need to execute the same query template many times, it will reduce execution time to use a **PreparedStatement** object instead of **Statement**.
 - **PreparedStatement** is given an SQL template statement **when it is created**.
 - **Precompiled** and ready to run many times.

- Example

```
PreparedStatement updateMovies = conn.prepareStatement(  
"UPDATE Movie SET studioName = ? WHERE studioName = ?");
```

```
import java.sql.*;
```

```
class UpdateMovies {
```

```
    public static void main (String args []) throws SQLException {
```

```
        DriverManager.registerDriver (new oracle.jdbc.driver.OracleDriver());
```

```
        Connection conn = DriverManager.getConnection
```

```
            ("jdbc:oracle:thin:@localhost:1522:studentdb", "userid", "password");
```

```
        PreparedStatement updateMovieStatement = conn.prepareStatement(
```

```
            "UPDATE Movies SET studioName = ? WHERE studioName LIKE ?" );
```

```
        String studiosBoughtByParamount [] = {"Disney", "Fox"};
```

```
        for(int i=0; i<studiosBoughtByParamount.length; i++) {
```

```
            updateMovieStatement.setString(1, "Paramount");
```

```
            updateMovieStatement.setString(2, "%" + studiosBoughtByParamount[i] + "%");
```

```
            updateMovieStatement.executeUpdate();
```

```
        }
```

```
        updateMovieStatement.close();
```

```
    }
```

```
}
```