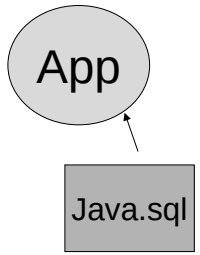
DB3
JDBC Examples

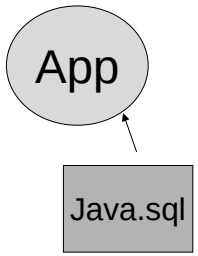
Emacs test.java

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



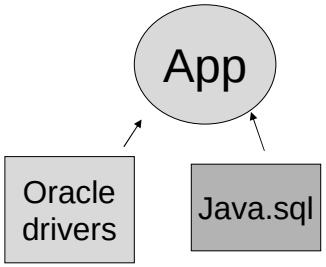
Emacs test.java

```
import java.sql.*;
public class test {
   public static void main(String [] args) {
   }

}
```



Emacs test.java

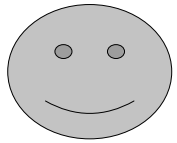
```
import java.sql.*;
public class test {
   public static void main(String [] args) {
Class.forName(“oracle.jdbc.driver.OracleDriver”);
   }

}
```



Command Line

```
setenv CLASSPATH
   /users/students4/software/ojdbc14.jar
```

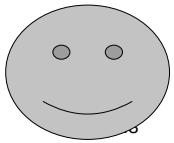

Handling Exceptions Well



```
// the right way
try{ executeUpdate(…)
}
Catch (Exception ee) {
System.out.println(“Could not add record “ +
   recnum + “ to database “);
ee.printStackTrace();
}
```


Handling Exceptions Individually

```
try {
Class.forName(“oracle.jdbc.driver.OracleDriver”);
}
catch (Exception ee) {
 System.out.println(“Could not load class “ +
                    “oracle.jdbc.driver.OracleDriver”);
}
```



Not Handling Exceptions

```
// novocaine
try{
}
Catch (Exception ee) {
// this is never going to happen
}
```

Reporting Exceptions Badly

```
// useless
try{
}
Catch (Exception ee) {
System.out.println("problem");
}
```


Handling Exceptions During Debugging

```
// geek
try{
}
Catch (Exception ee) {
ee.printStackTrace();
}
```

BUT DON'T LEAVE IT LIKE THAT!

A last word about exceptions:

```
// useless
try{
// do the whole program here
}
Catch (Exception ee) {
System.out.println("problem");
}
```

BAD, BAD, BAD!

Setting up the Connection

Class.forName("oracle.jdbc.driver.OracleDriver");

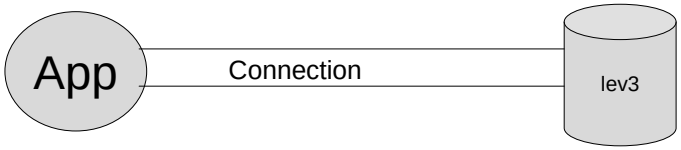
```
String connString =
  "jdbc:oracle:thin:uname/pw:@crooked.dcs.
  gla.ac.uk:1521:lev3";
```

Setting up the Connection

Class.forName(“oracle.jdbc.driver.OracleDriver”);

String connString =
“jdbc:oracle:thin:uname/pw:@crooked.dcs.gla.ac.uk:1521:lev3”;

Connection conn =
DriverManager.getConnection(connString);

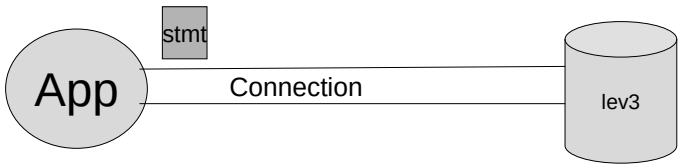


Setting up the Connection

Class.forName(“oracle.jdbc.driver.OracleDriver”);
String connString =
“jdbc:oracle:thin:uname/pw:@crooked.dcs.gla.ac.uk:1521:lev3”;
try{
Connection conn =
DriverManager.getConnection(connString);}
catch (SQLException se) {
System.out.println(“Could not open connection with connection string “ + connString);
se.printStackTrace();
}

Creating a Statement

Class.forName(“oracle.jdbc.driver.OracleDriver”);
String connString =
“jdbc:oracle:thin:uname/pw:@crooked.dcs.gla.ac.uk:1521:lev3”;
Connection conn =
DriverManager.getConnection(connString);
Statement stmt = conn.createStatement();



Dealing with Exceptions

Class.forName(“oracle.jdbc.driver.OracleDriver”);
String connString =
“jdbc:oracle:thin:uname/pw:@crooked.dcs.gla.ac.uk:1521:lev3”;

Connection conn =
DriverManager.getConnection(connString);

try {**Statement stmt = conn.createStatement();}**
catch{SQLException se) {
System.out.println(“Could not create statement “);
}

Command Line

```
setenv CLASSPATH
   /users/students4/software/ojdbc14.jar
javac test.java
java test
Could not create statement
```


Dealing with Exceptions

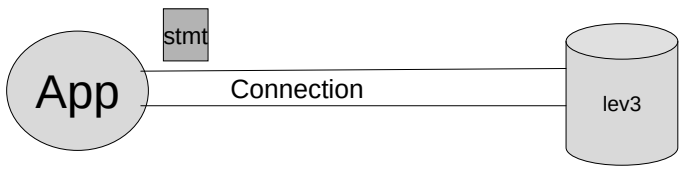
```
Class.forName("oracle.jdbc.driver.OracleDriver");
String connString =
   "jdbc:oracle:thin:uname/pw:@crooked.dcs.gla.ac.uk:1521
   :lev3";

Connection conn =
   DriverManager.getConnection(connString);

try {Statement stmt = conn.createStatement();}
catch{SQLException se) {
   System.out.println("Could not create statement");
   se.printStackTrace();
   // or System.out.println(se.getName());
}
```


Command Line

```
setenv CLASSPATH
   /users/students4/software/ojdbc14.jar
javac test.java
java test
Could not create statement
Error Explanation…
```


Updating the Movies table

```
Class.forName("oracle.jdbc.driver.OracleDriver");
String connString =
   "jdbc:oracle:thin:uname/pw:@crooked.dcs.gla.ac.uk:1521
   :lev3";
Connection conn =
   DriverManager.getConnection(connString);
Statement stmt = conn.createStatement();
stmt.executeUpdate("insert into Movies
   values(1,'StarWars')");
```

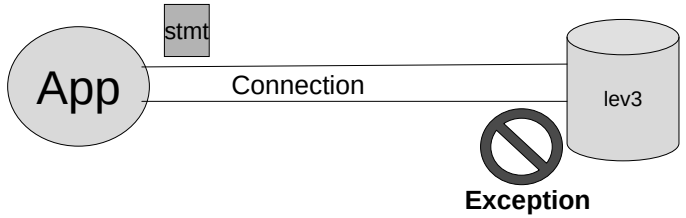


Dealing with Exceptions

Movies(MID, Title,Year, Explosions)

```
Class.forName("oracle.jdbc.driver.OracleDriver");
String connString =
    "jdbc:oracle:thin:uname/pw:@crooked.dcs.gla.ac.uk:1521
    :lev3";
Connection conn =
    DriverManager.getConnection(connString);
Statement stmt = conn.createStatement();

try {stmt.executeUpdate("insert into Movies
    values(1,'StarWars')");}
catch (SQLException se) {
    System.out.println("Could not insert StarWars ");
    se.printStackTrace();
}
```


insert into Movies values(1,'StarWars');



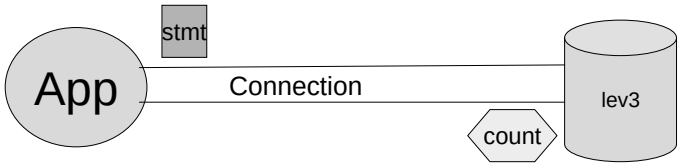
Simplifying

Movies(MID, Title,Year, Explosions)

```
Class.forName("oracle.jdbc.driver.OracleDriver");
String connString =
    "jdbc:oracle:thin:uname/pw:@crooked.dcs.gla.ac.uk:1521
    :lev3";
Connection conn =
    DriverManager.getConnection(connString);
Statement stmt = conn.createStatement();
stmt.executeUpdate("insert into Movies
    values(1,'StarWars',1977,3600)");
```


Checking for Success

```
Class.forName("oracle.jdbc.driver.OracleDriver");
String connString =
    "jdbc:oracle:thin:uname/pw:@crooked.dcs.gla.ac.uk:1521
    :lev3";
Connection conn =
    DriverManager.getConnection(connString);
Statement stmt = conn.createStatement();
int count = stmt.executeUpdate("insert into Movies
    values(1,'StarWars',1977,3600)");
// Now check and see whether an update was made
if (count==1) System.out.println("StarWars Update
    Succeeded");
else System.out.println("no StarWars Update Done");
```



Command Line

```
setenv CLASSPATH
   /users/students4/software/ojdbc14.jar
javac test.java
java test
Star Wars Update Succeeded
```


Command Line

```
setenv CLASSPATH
   /users/students4/software/ojdbc14.jar
javac test.java
java test
No Star Wars Update Done
```


Updating Again

Movies(MID, Title,Year, Explosions)

Class.forName("oracle.jdbc.driver.OracleDriver");

String connString =
   "jdbc:oracle:thin:uname/pw:@crooked.dcs.gla.ac.uk:1521
   :lev3";

Connection conn =
   DriverManager.getConnection(connString);

Statement stmt = conn.createStatement();

// update Title

int count= stmt.executeUpdate("update Movies set
   MovieTitle='StarWars2' where MID=2");

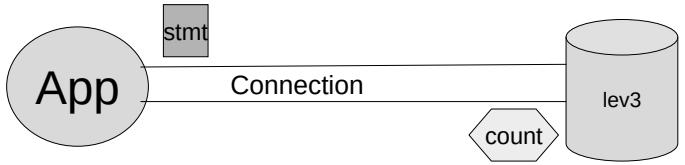
Updating Again

Movies(MID, Title,Year, Explosions)

```
Class.forName("oracle.jdbc.driver.OracleDriver");
String connString =
   "jdbc:oracle:thin:uname/pw:@crooked.dcs.gla.ac.uk:1521
   :lev3";
Connection conn =
   DriverManager.getConnection(connString);
Statement stmt = conn.createStatement();

try {int count= stmt.executeUpdate("update Movies set
   Title='StarWars2' where MID=2");}

catch (SQLException se) {
   System.out.println("Could not update StarWars ");
   se.printStackTrace();
}
```


Updating Again

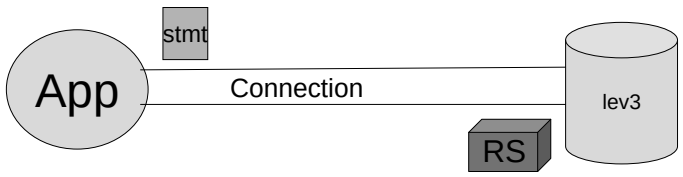
```
Class.forName("oracle.jdbc.driver.OracleDriver");
String connString =
   "jdbc:oracle:thin:uname/pw:@crooked.dcs.gla.ac.uk:1521
   :lev3";
Connection conn =
   DriverManager.getConnection(connString);
Statement stmt = conn.createStatement();
int count= stmt.executeUpdate("update Movies set
   Title='StarWars2' where MID=2");
```



Querying

Movies(MID, Title, Year, Explosions)

```
ResultSet rs = stmt.executeQuery("select * from Movies
   where Explosions > 100");
```



Querying

```
String query = "select * from Movies where Explosions > 100";
try {

ResultSet rs = stmt.executeQuery(query);
query = "select * from Movies where Explosions < 5";
rs = stmt.executeQuery(query);

}
catch (SQLException se) {
   System.out.println("Could not get result of query " +
   query);
   se.printStackTrace();
}
```

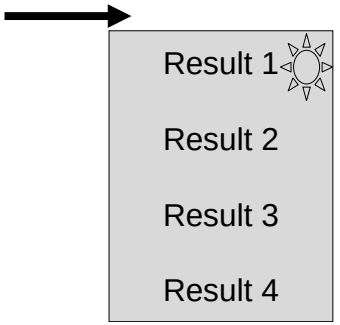
Querying

Movies(MID, Title, Year, Explosions)

String query = “select Title from Movies where Explosions > 100”;

ResultSet rs = stmt.executeQuery(query);

```
System.out.println(“Big Explosion Movies”);
while (rs.next()) {
    String title = rs.getString(“Title”);
    System.out.println(“title”);
}
```



Command Line Output

Big Explosion Movies
Star Wars
BladeRunner
Aliens

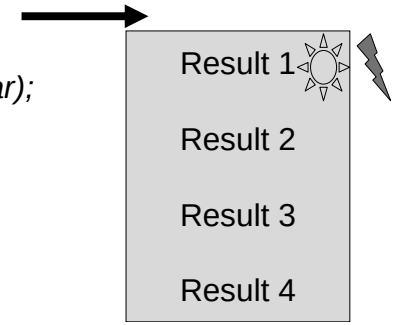
Querying

Movies(MID, Title, Year, Explosions)

String query = “select Title,Year from Movies where Explosions > 100”;

ResultSet rs = stmt.executeQuery(query);

```
System.out.println(“Big Explosion Movies”);
while (rs.next()) {
    String title = rs.getString(“Title”);
    String year=rs.getInt(“Year”);
    System.out.println(“title”+” “ + year);
}
```



Command Line Output

Big Explosion Movies
Star Wars 1977
BladeRunner 1982
Aliens 1986

Prepared Statements

- Created by prepareStatement on Connection:
  - conn.prepareStatement(…);
- It is registered with the DB once
- Compiled & Prepared for Use
- Faster because it does not need to be re-parsed
- Both Updates and Queries supported

PreparedStatement

String query = “select Title,Year from Movies where Explosions > 100”;

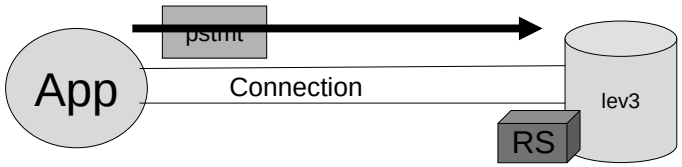
Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(query);

PreparedStatement pstmt = conn.prepareStatement(query);

ResultSet rs = pstmt.executeQuery();

PreparedStatement



Making them Dynamic

Movies(MID, Title, Year, Explosions)

*PreparedStatement pstmt=conn.prepareStatement(“select * from Movies where Explosions > ?”);*

pstmt.setInt(1,0);

ResultSet rs = pstmt.executeQuery();

pstmt.setInt(1,100);

ResultSet rs = pstmt.executeQuery();

pstmt.setInt(1,100000);

ResultSet rs = pstmt.executeQuery();

Making them Dynamic

Movies(MID, Title, Year, Explosions)

PreparedStatement
*pstmt=conn.prepareStatement("select * from Movies where Explosions > ? And Year > ?");*

pstmt.setInt(1,0); pstmt.setInt(2,1960);

ResultSet rs = pstmt.executeQuery();

pstmt.setInt(1,100); pstmt.setInt(2,1980);

ResultSet rs = pstmt.executeQuery();

pstmt.setInt(1,100000); pstmt.setInt(2,1990);

ResultSet rs = pstmt.executeQuery();

Making them Dynamic

Movies(MID, Title, Year, Explosions)

PreparedStatement
pstmt=conn.prepareStatement("select Explosions from Movies where Title = ?");

pstmt.setString(1,"StarWars");

ResultSet rs = pstmt.executeQuery();

PHP Example

I want to be able to use a web browser page to pick a random student from a MySql database table.

Create Student table as follows:

```
create table Student (
  Id int primary key,
  Name varchar(32) not null,
  db3mark int
);
```



```
<?php
//connect to the database
$linkID=@mysql_connect("localhost","","");

// check whether it was successful
if ($linkID == FALSE) {
print "Couldn't establish the link to the database";
}
else {
    print "Linked to the database successfully<br>";
    mysql_select_db("test",$linkID);

    $query = "Select Name from Student order by Name ";

    // query the database
    $resultID=mysql_query($query,$linkID);

    // did the query work?
    if ($resultID == FALSE) {
          print "Query failed<BR>";
    }
```

```
    else {
        print "Query succeeded<BR>";
        // how many students?
        $numStudents = mysql_num_rows($resultID);

        print $numStudents . " students in the database <br>";

        // get a random number
        $random = rand(0,$numStudents-1);
        print "Random number generated is " . $random . "<br>";

        // now get the lucky person's name
        mysql_data_seek($resultID, $random);

        $row = mysql_fetch_row($resultID);
        $lucky = $row[0];

        print "The lucky winner is " . $lucky . "<br>";
    }
// remember to close the database
mysql_close($linkID);
}
?>
```


Using fetch_object

```
$cnt=0;
$lucky="no one";
while ($row = mysql_fetch_object($resultID)) {

if ($cnt == $random) {
        $lucky = $row->name;
        break;
}
$cnt++;
}

print "The lucky winner is " . $lucky . "<br>";
```


Basic Steps in Using a DB using JDBC or PHP

1. Connect to the database server
2. Choose the database (PHP only)
3. Check
4. Create statement (JDBC only)
5. Query / update the database
6. Check for errors
7. If query – work through result set
8. If update – check for success
9. Disconnect