

Accessing the database from Java

● 6 steps:

- ▶ Load a connection library between Java and the database
- ▶ Connect to the database, including login credentials
- ▶ Create a statement to run
- ▶ Run the statement
- ▶ Do something with the results of the statement
- ▶ Close the database and release all resources

● Remember to put the code inside a try...catch block.

Accessing the database from Java

● 6 steps:

- ▶ Load a connection library between Java and the database
 - Java DataBase Connection (JDBC) library for mysql
 - Code:

```
Class.forName("com.mysql.cj.jdbc.Driver");
```

- ▶ Connect to the database, including login credentials
- ▶ Create a statement to run
- ▶ Run the statement
- ▶ Do something with the results of the statement
- ▶ Close the database and release all resources

Accessing the database from Java

● 6 steps:

- ▶ Load a connection library between Java and the database
- ▶ Connect to the database, including login credentials

- Code:

```
Connection connect = null;  
connect = DriverManager.getConnection("jdbc:mysql://db.cs.dal.ca:3306",  
user, password);
```

- ▶ Create a statement to run
- ▶ Run the statement
- ▶ Do something with the results of the statement
- ▶ Close the database and release all resources

Accessing the database from Java

● 6 steps:

- ▶ Load a connection library between Java and the database
- ▶ Connect to the database, including login credentials
- ▶ Create a statement to run

- Code:

```
Statement statement = null;  
statement = connect.createStatement();
```

- ▶ Run the statement
- ▶ Do something with the results of the statement
- ▶ Close the database and release all resources

Accessing the database from Java

● 6 steps:

- ▶ Load a connection library between Java and the database
- ▶ Connect to the database, including login credentials
- ▶ Create a statement to run
- ▶ Run the statement

- Code:

```
ResultSet resultSet = null;  
resultSet = statement.executeQuery("select * from employees;");
```

(Could also be executeUpdate() or execute())

- ▶ Do something with the results of the statement
- ▶ Close the database and release all resources

Accessing the database from Java

● 6 steps:

- ▶ Load a connection library between Java and the database
- ▶ Connect to the database, including login credentials
- ▶ Create a statement to run
- ▶ Run the statement
- ▶ Do something with the results of the statement

- Code:

```
while (resultSet.next()) {  
    resultSet.getString("lastName");  
    resultSet.getString(4);  
}
```

- ▶ Close the database and release all resources

Accessing the database from Java

● 6 steps:

- ▶ Load a connection library between Java and the database
- ▶ Connect to the database, including login credentials
- ▶ Create a statement to run
- ▶ Run the statement
- ▶ Do something with the results of the statement
- ▶ Close the database and release all resources
 - Very, very, very important! Can run out of connections!
 - Code:

```
resultSet.close();  
statement.close();  
connection.close();
```

All together...

```
import java.sql.*;
Connection connect = null;
Statement statement = null;
ResultSet resultSet = null;

Class.forName("com.mysql.cj.jdbc.Driver");

connect = DriverManager.getConnection(
    "jdbc:mysql://db.cs.dal.ca:3306?serverTimezone=UTC", user, pass);
statement = connect.createStatement();

statement.execute ("use csci3901;");
resultSet = statement.executeQuery("select lastName from employees limit 3;");

while (resultSet.next()) {
    System.out.println("Employee name: " + resultSet.getString("lastName"));
}
resultSet.close();
statement.close();
connect.close();
```


Quirks

- **On some systems:**

- ▶ **- the driver `com.mysql.cj.jdbc.Driver` should have the “cj.” removed**
- ▶ **Some systems don't like using encrypted connections (fine for the class, not so good in general). If your system is complaining about SSL then add “&useSSL=false” after the UCT part of the `getConnection` parameter**