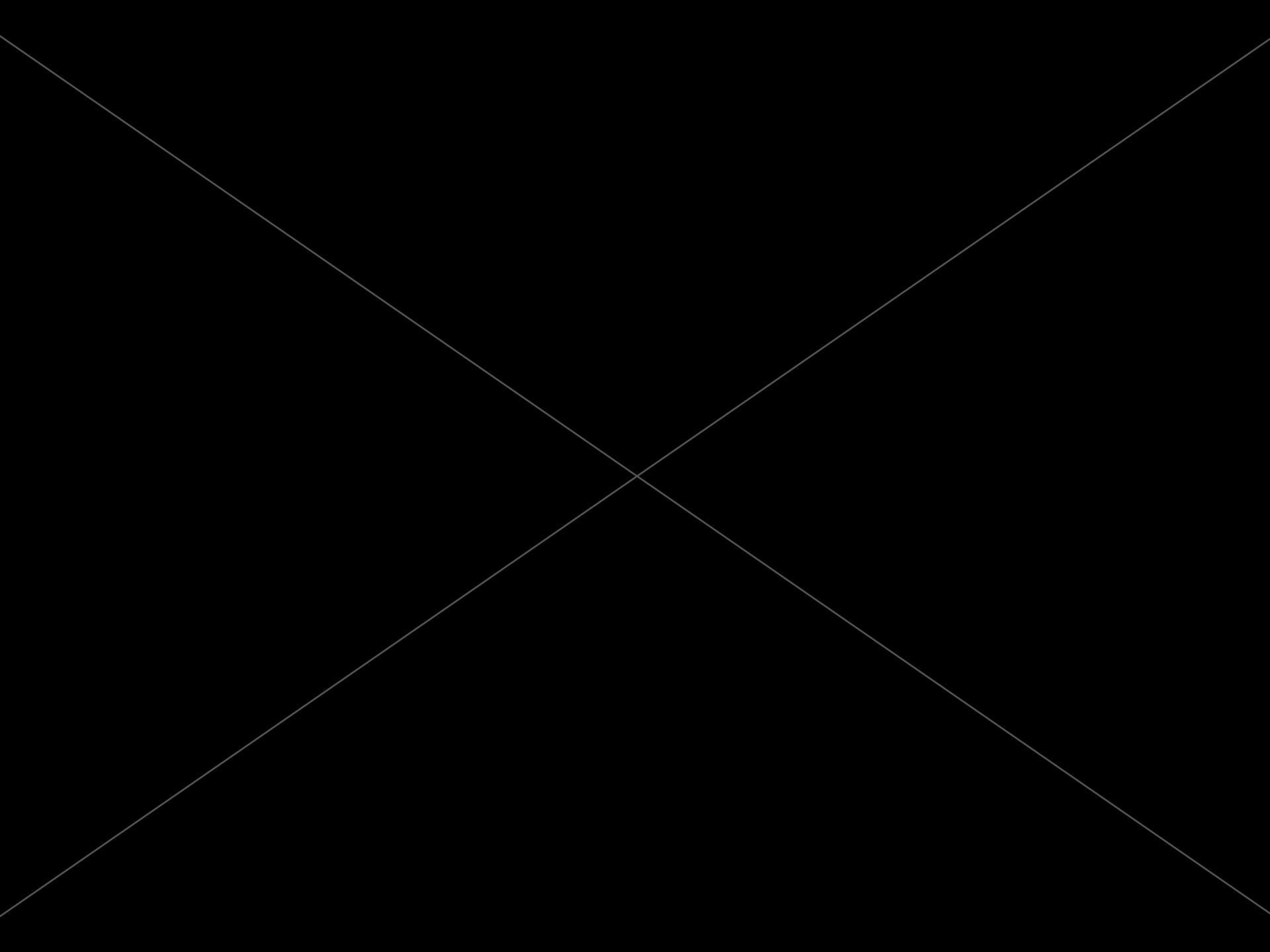

Java Database Connectivity (JDBC)

- An interface to communicate with a relational database
  - Allows database agnostic Java code
  - Treat database tables/rows/columns as Java objects
- JDBC driver
  - An implementation of the JDBC interface
  - Communicates with a particular database

Eclipse JDBC setup

- Install driver
  - Download MySQL JDBC driver from assignment Web page
  - Unzip mysql-connector-xxx.jar
  - Add mysql-connector-[redacted].jar to Eclipse project
    - Project → Properties → Java Build Path → Libraries → Add External JARs

JDBC steps

1. Connect to database
2. Query database (or insert/update/delete)
3. Process results
4. Close connection to database

1. Connect to database

- Load JDBC driver
  - `Class.forName("com.mysql.jdbc.Driver").newInstance();`
- Make connection
  - `Connection conn = DriverManager.getConnection(url);`
- URL
  - **Format: "jdbc:<*subprotocol*>:<*subname*>"**
  - `jdbc:mysql://128.100.53.33/GROUPNUMBER?user=USER&password=PASSWORD`

2. Query database

a. Create statement
   - `Statement stmt = conn.createStatement();`
   - stmt object sends SQL commands to database
   - Methods
     - executeQuery() for SELECT statements
     - executeUpdate() for INSERT, UPDATE, DELETE, statements

b. Send SQL statements
   - stmt.executeQuery("SELECT …");
   - stmt.executeUpdate("INSERT …");

3. Process results

- Result of a SELECT statement (rows/columns) returned as a ResultSet object
  - `ResultSet rs = stmt.executeQuery("SELECT * FROM users");`
- Step through each row in the result
  - `rs.next()`
- Get column values in a row
  - `String userid = rs.getString("userid");`
  - `int type = rs.getInt("type");`

users table				
userid	**firstname**	**lastname**	**password**	**type**
Bob	Bob	King	cat	0
John	John	Smith	pass	1

Print the users table

```
ResultSet rs = stmt.executeQuery("SELECT * FROM users");

while (rs.next()) {
    String userid = rs.getString(1);
    String firstname = rs.getString("firstname");
    String lastname = rs.getString("lastname");
    String password = rs.getString(4);
    int type = rs.getInt("type");
    System.out.println(userid + " " + firstname + " " +
       lastname + " " + password + " " + type);
}
```

users table				
userid	**firstname**	**lastname**	**password**	**type**
Bob	Bob	King	cat	0
John	John	Smith	pass	1

Add a row to the users table

```
String str =
     "INSERT INTO users
      VALUES('Bob', 'Bob', 'King',
             'cat', 0)”;

// Returns number of rows in table
int rows = stmt.executeUpdate(str);
```

users table				
userid	**firstname**	**lastname**	**password**	**type**
Bob	Bob	King	cat	0

4. Close connection to database

- Close the ResultSet object
  - `rs.close();`
- Close the Statement object
  - `stmt.close();`
- Close the connection
  - `conn.close();`

```
import java.sql.*;

public class Tester {
   public static void main(String[] args) {
      try {
         // Load JDBC driver
         Class.forName("com.mysql.jdbc.Driver").newInstance();

         // Make connection
         String url =
               "jdbc:mysql://128.100.53.33/GRP?user=USER&password=PASS"
         Connection conn = DriverManager.getConnection(url);

         // Create statement
         Statement stmt = conn.createStatement();

         // Print the users table
         ResultSet rs = stmt.executeQuery("SELECT * FROM users");
         while (rs.next()) {
               ...
         }

         // Cleanup
         rs.close(); stmt.close(); conn.close();

   } catch (Exception e) {
         System.out.println("exception " + e);
   }
}
```