

JDBC Tutorial

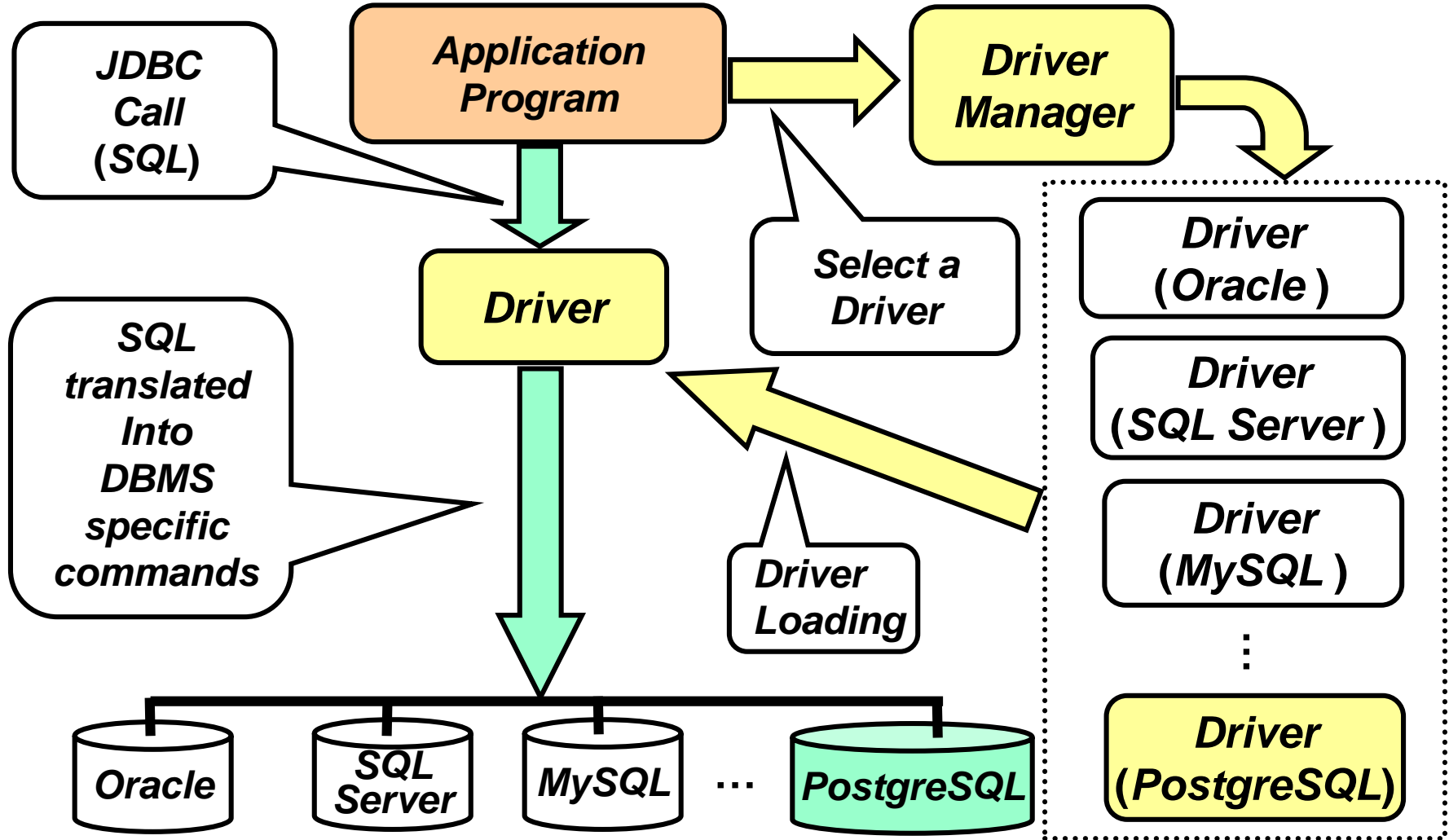
SWEN304/SWEN439

Trimester 1, 2024

Engineering and Computer Science



Accessing Databases from Application Programs



Outline

- `java.sql` Package
- Starting a database session,
- Registering a PostgreSQL driver,
- Establishing a connection

- Executing SQL statements
 - A Statement object
 - A ResultSet object
 - Getting tuples from the ResultSet object
 - Controlling access mode in JDBC (optional)

The JDBC Package `java.sql`

- Before using JDBC to connect to a database in a DBMS, you need to make the respective JDBC classes available in your application program

- Hence, your Java code should include the following lines:

```
import java.sql.DriverManager;  
import java.sql.Connection;  
import java.sql.Statement;  
import java.sql.SQLException;
```

or simply

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

Starting a Database Session

- In your Project 2, the program `LibraryUI` should create an `LibraryModel` object
 - The `LibraryModel` class is in the file `LibraryModel.java`

```
public class LibraryModel{  
    ...  
    private Connection con = null;  
    ...  
    public LibraryModel(JFrame parent,  
        String userid, String password){  
        //Register a PostgreSQL Driver  
        //Establish a Connection  
    }  
}
```

Registering a PostgreSQL Driver

```
try{
    Class.forName("org.postgresql.Driver");
}
catch (ClassNotFoundException cnfe){
    System.out.println("Can not find"+
        "the driver class: "+
        "\nEither I have not installed it"+
        "properly or \n postgresql.jar "+
        " file is not in my CLASSPATH);
}
```

Establishing a Connection to the Database

```
String url = "jdbc:postgresql:" + "//  
db.ecs.vuw.ac.nz/" + userid + "_jdbc";  
  
try{  
    con = DriverManager.getConnection(url,  
        userid, password);  
}  
catch (SQLException sqllex) {  
    System.out.println("Can not connect");  
    System.out.println(sqllex.getMessage());  
}
```

SQL Statements – Step by Step (1)

```
Statement s = null;
```

```
try{
```

```
    s = con.createStatement();
```

```
}
```

```
catch (SQLException sqlex) {
```

```
    System.out.println("An exception"+  
        "while creating a statement,"+  
        "probably means I am no longer"+  
        "connected");
```

```
}
```


SQL Statements – Step by Step (2)

```
ResultSet rs = null;

try{
    rs = s.executeQuery("SELECT * FROM"+
        "Student");
}
catch (SQLException sqlex) {
    System.out.println("An exception"+
        "while executing a query, probably"+
        "means my SQL is invalid");
}
```

SQL Statements – Step by Step (3)

```

try{
    while (rs.next()) {
        System.out.println(rs.getInt(1));
    }
}
catch (SQLException sqlex) {
    System.out.println("An exception"+
        "while processing a result,
probably"+
        "means I have done something"+
        "really bad");
}

```

SQL Statements – Putting all together (1)

```
// Start of the LibraryModel constructor  
// Suppose a JDBC driver is already  
// registered (automatically or manually)
```

```
String url =  
    "jdbc:postgresql://db.ecs.vuw.ac.nz/" +  
    userid + "_jdbc";
```

```
try{  
    Connection con =  
        DriverManager.getConnection(url,  
            userid, password);  
// continued on the next slide
```

SQL Statements – Putting all together (2)

```

// Create a Statement object
Statement s = con.createStatement();
// Execute the Statement object
ResultSet rs = s.executeQuery(
    "SELECT * FROM" + "Student");
// Handle query answer in ResultSet object
while (rs.next()) {
    System.out.println(rs.getInt(1));
}
} // End of the try block

catch (SQLException sqlEx) {
    System.out.println(sqlEx.getMessage());
}

```

Controlling Access Mode of JDBC Connections

- According to SQL/92, the default access mode of JDBC connections is `READ WRITE`
- You can use the `READ ONLY` access mode for queries instead
 - this enables simplified query optimisation, and thus faster query processing

- To change the access mode, use the methods

```
public abstract void setReadOnly(boolean  
readOnly) throws SQLException
```

and

```
public abstract boolean isReadOnly()  
throws SQLException
```