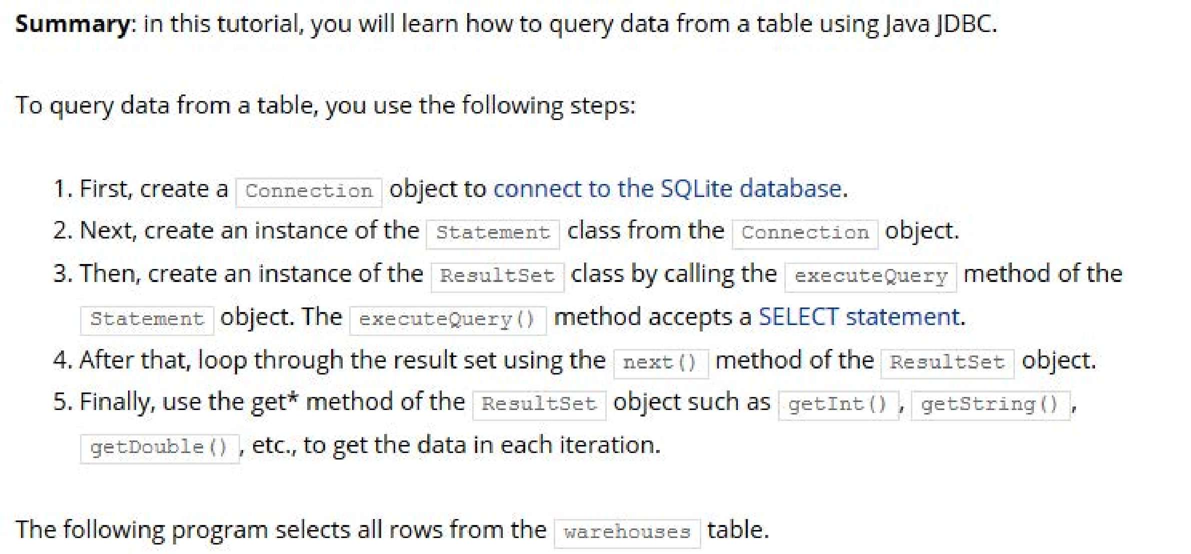


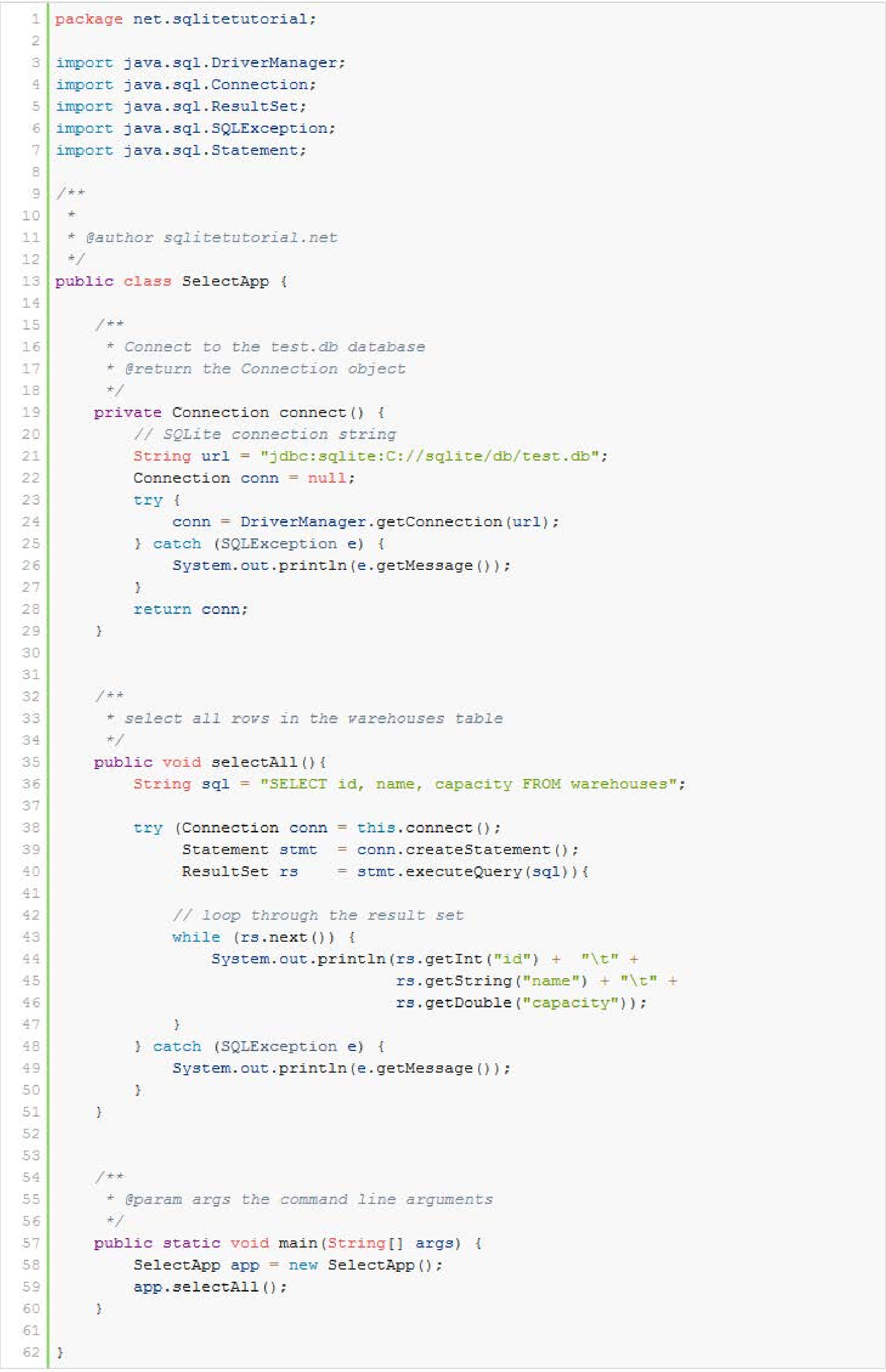
Summary: in this tutorial, you will learn how to query data from a table using Java JDBC.

To query data from a table, you use the following steps:

1. First, create a `Connection` object to connect to the SQLite database.
2. Next, create an instance of the `Statement` class from the `Connection` object.
3. Then, create an instance of the `ResultSet` class by calling the `executeQuery` method of the `Statement` object. The `executeQuery()` method accepts a SELECT statement.
4. After that, loop through the result set using the `next()` method of the `ResultSet` object.
5. Finally, use the get* method of the `ResultSet` object such as `getInt()`, `getString()`, `getDouble()`, etc., to get the data in each iteration.

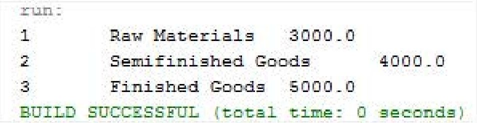
The following program selects all rows from the `warehouses` table.

The following program selects all rows from the `warehouses` table.



```java
package net.sqlitetutorial;

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

/**
 *
 * @author sqlitetutorial.net
 */
public class SelectApp {

    /**
     * Connect to the test.db database
     * @return the Connection object
     */
    private Connection connect() {
        // SQLite connection string
        String url = "jdbc:sqlite:C://sqlite/db/test.db";
        Connection conn = null;
        try {
            conn = DriverManager.getConnection(url);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
        return conn;
    }


    /**
     * select all rows in the warehouses table
     */
    public void selectAll(){
        String sql = "SELECT id, name, capacity FROM warehouses";

        try (Connection conn = this.connect();
             Statement stmt  = conn.createStatement();
             ResultSet rs    = stmt.executeQuery(sql)){

            // loop through the result set
            while (rs.next()) {
                System.out.println(rs.getInt("id") +  "\t" +
                                   rs.getString("name") + "\t" +
                                   rs.getDouble("capacity"));
            }
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }


    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        SelectApp app = new SelectApp();
        app.selectAll();
    }

}
```

The following illustrates the output of the program:



```
run:
1       Raw Materials   3000.0
2       Semifinished Goods      4000.0
3       Finished Goods  5000.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

Querying data with parameters

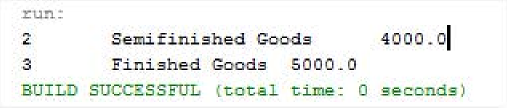
To use parameters in the query, you use the `PreparedStatement` object instead. For example, the following method selects the warehouse whose capacity is greater than a specified capacity.

```java
    /**
     * Get the warehouse whose capacity greater than a specified capacity
     * @param capacity
     */
    public void getCapacityGreaterThan(double capacity){
               String sql = "SELECT id, name, capacity "
                          + "FROM warehouses WHERE capacity > ?";

        try (Connection conn = this.connect();
             PreparedStatement pstmt  = conn.prepareStatement(sql)){

            // set the value
            pstmt.setDouble(1,capacity);
            //
            ResultSet rs  = pstmt.executeQuery();

            // loop through the result set
            while (rs.next()) {
                System.out.println(rs.getInt("id") +  "\t" +
                                   rs.getString("name") + "\t" +
                                   rs.getDouble("capacity"));
            }
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }
```

To find the warehouses whose capacities are greater than 3600, you use the `getCapacityGreaterThan()` method as follows:

```java
SelectApp app = new SelectApp();
app.getCapacityGreaterThan(3600);
```

The following is the output:



```
run:
2       Semifinished Goods      4000.0
3       Finished Goods  5000.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

In this tutorial, you have learned how to query data from the table in SQLite database from a Java program.