

JDBC Overview: JDBC is an API that allows java applications to connect to a relational databases, execute SQL queries and retrieve and process results.

### Steps to Execute SELECT query:

- i) Load the JDBC drivers.
- ii) Establish a database connection.
- iii) Create a Statement.
- iv) Execute the SELECT query.
- v) Process the ResultSet.
- vi) Close resources.

### Example:

```
import java.sql.*  
public class JdbcExample{  
    public static void main (String [] args){  
        Connection conn = null;  
        Statement stmt = null;  
        ResultSet rs = null;  
        try {  
            Class.forName("com.mysql.cj.jdbc.Driver");  
            conn = DriverManager.getConnection ("jdbc:mysql://  
                localhost:3306/test", "root", "root");  
            stmt = conn.createStatement();  
            rs = stmt.executeQuery ("SELECT * FROM student");  
            while (rs.next ()) {  
                System.out.println (rs.getString (1) + " " +  
                    rs.getString (2) + " " +  
                    rs.getString (3));  
            }  
        } catch (Exception e) {  
            e.printStackTrace ();  
        }  
    }  
}
```

2

```
//localhost:3306/testdb", "user", "password");  
stmt = conn.createStatement();  
rs = stmt.executeQuery("SELECT * FROM students");  
while(rs.next()) {  
    System.out.println(rs.getInt("id") + " " + rs.getString  
        ("name"));  
}  
}  
}  
catch (Exception e) {  
    System.out.println("Database error: " + e.getMessage());  
} finally {  
    try {  
        if (rs != null) rs.close();  
        if (stmt != null) stmt.close();  
        if (conn != null) conn.close();  
    } catch (SQLException e) {  
        System.out.println("Error closing resources");  
    }  
}  
}
```