**JDBC**

JDBC Driver is a software component that enables java application to interact with the database.There are 4 types of JDBC drivers:

1. JDBC-ODBC bridge driver
2. Native-API driver (partially java driver)
3. Network Protocol driver (fully java driver)
4. Thin driver (fully java driver)

[**http://www.javatpoint.com/jdbc-driver**](http://www.javatpoint.com/jdbc-driver)

There are 5 steps to connect any java application with the database in java using JDBC. They are as follows:

1)Register the driver class

We will load the jdbc driver by using *forName()*  of java.lang.Class

Class.forName(“oracle.jdbc.driver.OracleDriver”);

forName() is static method in class called “Class”

2)Creating connection

We will establish database connection

Connection con= DriverManager.getConnection(“jdbc:oracle:thin:@localhost:1521:xe”,”test”,”test”);

Connection is interface, getConnection() is static method in “DriverManager” class.

3)Creating statement

We will create a Statement or PreparedStatement or CallableStatement using connection object.

Statement stmt = con.createStatement();

PreparedStatement pstmt=con.prepareStatement(“sql Query”);

We will use callable Statement to deal with stored procedures.

Statement , PreparedStatement and CallableStatement all are interfaces.

4)Executing queries

Statement and PreparedStatement have methods for executing DB queries.

Stmt.executeUpdate(“Sql query”);

Pstmt.executeUpdate();

ResultSet is used for retrieving results (data) from “select” query.

ResultSet rs=stmt.executeQuery(“sql query”);

ResultSet rs=pstmt.executeQuery();

ResultSet is interface.

5)Closing connection

We will close the connections of Connection, Statement, preparedStatement and ResultSet.

If they are opend.we write this code in finally block, because finally block will be executed even if Exception occurred in try or catch block.