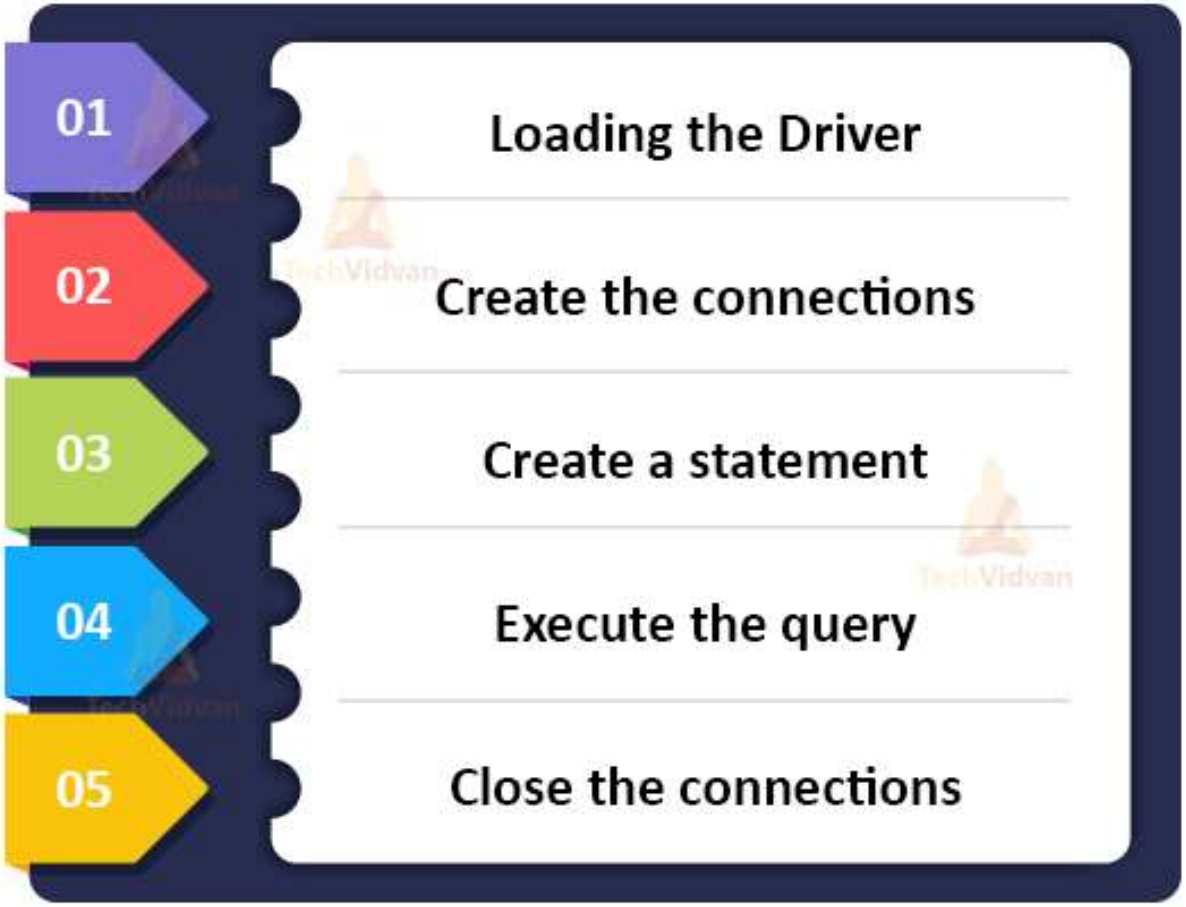
JDBC Connectivity

Steps to connect Java Program & Database



1. Loading the Driver

- We first need to load the driver or register it before using it in the program. There should be registration once in your program. We can register a driver in any of the two ways:

a. **Class.forName():** In this, we load the driver's class file into memory during runtime. There is no need to use a new operator for the creation of an object. The following shows the use of Class.forName() to load the Oracle driver:

Class.forName("oracle.jdbc.driver.OracleDriver");

b. DriverManager.registerDriver(): DriverManager is an inbuilt class of Java that comes with a static member register. We call the drivers class' constructor at compile-time. The following example shows the use of DriverManager.registerDriver() to register the Oracle driver:

*DriverManager.registerDriver(**new** oracle.jdbc.driver.OracleDriver())*

2. Create the connections

- After loading the driver, we need to establish connections using the following code:
  - user: username from which sql command prompt can be accessed.
  - password: password from which sql command prompt can be accessed.
  - con: reference to Connection interface.
  - url : Uniform Resource Locator. We can create it as follows:
  - String url = " jdbc:oracle:thin:@localhost:1521:xe"

3. Create a statement

- Once you establish a connection, you can interact with the database. The JDBCStatement, CallableStatement, and PreparedStatement interfaces define the methods that allow us to send the SQL commands and receive data from the database.
- Use of JDBC Statement is as follows:

  Statement statement = con.createStatement();

  Here, con is a reference to the Connection interface that we used in the previous step.

4. Execute the query

- The most crucial part is executing the query. Here, Query is an SQL Query. Now, as we know that we can have multiple types of queries. Some of them are as follows:
  - The query for updating or inserting tables in a database.
  - The query for retrieving data from the database.
- The executeQuery() method of the Statement interface executes queries of retrieving values from the database. The executeQuery() method returns the object of ResultSet that we can use to get all the records of a table.

5. Close the connections

- We need to close the connection. By closing the connection, objects of Statement and ResultSet interface are automatically closed. The close() method of Connection interface closes the connection.

con.close();