

JDBC

Java Database Connectivity

MySQL : database is use to store the data in table format.

JDBC provided lot of pre defined classes and interfaces which help to connect the database (mysql, oracle, db2 etc). Using Java with help of jdbc

We can store, delete, update and retrieve record from database.

In VM open terminal and write

```
sudo mysql -u root -p
```

```
password : Simplilearn
```

Steps to connect the database.

1. We need to create main class or user defined class with method ie main or user defined method.
2. JDBC always throw checked exception ie SQLException so we need to write try-catch or throws exception mandatory.
3. We need to load the Driver. Driver is a pre defined API provided by vendor which help to connect the database.
4. 4 types of driver.
 - a. Type 1 : jdbc odbc bridge driver
 - b. Type 2 : jdbc native api driver
 - c. Type 3 : jdbc net protocol driver
 - d. Type 4 : jdbc pure or thin driver
5. From Java 8 onward type 1 driver deprecated or removed.
6. `Class.forName("driverName");` Class is a pre defined class the name itself is `Class`. which contains `forName()` static method which help to load the Driver.

Database : MySQL : Driver Name for MySQL Database.

MySQL Version 5.x → `com.mysql.jdbc.Driver`

MySQL Version 8.x → `com.mysql.cj.jdbc.Driver`

If we are planning to load type 4 driver for mysql we will get error as
Class not found exception.

Oracle or MySQL database provided type 4 driver in the form of jar file.

7. Establish the connection.

```
Connection con = DriverManager.getConnection(url,username,password);
```

Connection is a interface part of sql package.

DriverManager is a pre defined class part of sql package which contains
getConnection static method which takes 3 parameter

1. url
2. username
3. password

8. if we are planning to insert, delete, update and retrieve records from table we need to create type of Statement.
9. Statement is a pre defined interface which provide set of method which help to insert, delete, update and retrieve records.

```
Statement stmt = con.createStatement();
```

10.DML Operation

```
int res = stmt.executeUpdate("insert query/ delete query / update query")
```

executeUpdate method return type is int value.

If record inserted or delete or update successfully it return > 0.

11. Select Query : DRL or DQL

```
ResultSet rs = stmt.executeQuery("select query");
```

execute query method return type is ResultSet interface reference.

ResultSet it like an iterator which help to retrieve each records one by one.

```
rs.getXXX(columnName/columnIndex);
```

using Statement interface we can't pass dynamic value.

If we want to pass dynamic value PreparedStatement interface need to use.

Statement	Vs	PreparedStatement
-----------	----	-------------------

1. Statement is use to execute static query.
2. PreparedStatement is use to execute dynamic query.
3. Statement internally slow in performance. Means each query get compile in java side and execute in database side.
4. PreparedStatement is fast. Query compile only once and execute again and again.
5. PreparedStatement support parameterized query concept.

In Java using Servlet and JSP we can take the value from forms.

As of now console is input for us not html page.