BCA – 401: Java Programming

Rahul Kumar Singh

In today's Class we have discussed on JDBC- ODBC (Java Database Connectivity) in Java.

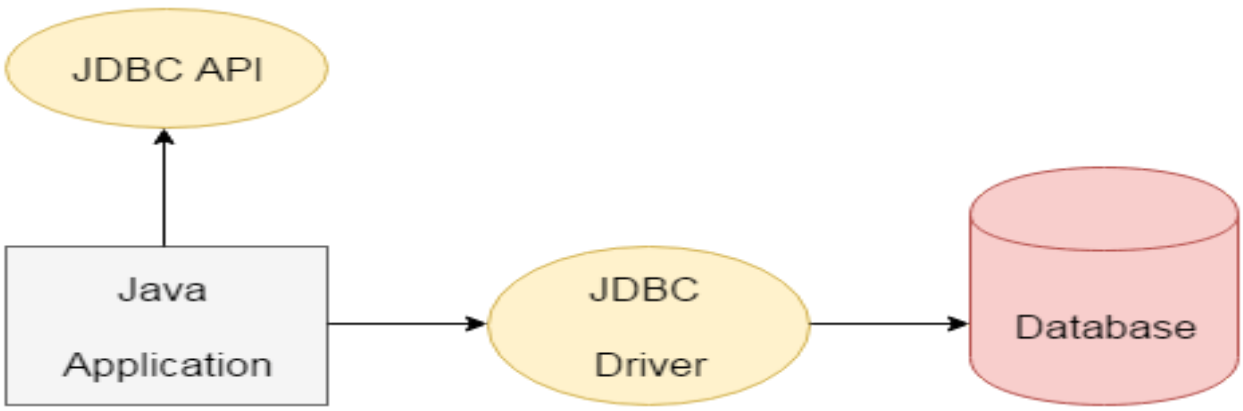
Java JDBC:-

JDBC stands for Java Database Connectivity. JDBC is a Java API to connect and execute the query with the database. It is a part of JavaSE (Java Standard Edition). JDBC API uses JDBC drivers to connect with the database.

There are four types of JDBC drivers:

- JDBC-ODBC Bridge Driver
- Native Driver
- Network Protocol Driver
- Thin Driver

We can use JDBC API to access tabular data stored in any relational database. By the help of JDBC API, we can save, update, delete and fetch data from the database. It is like Open Database Connectivity (ODBC) provided by Microsoft.



The current version of JDBC is 4.3. It is the stable release since 21st September, 2017. It is based on the X/Open SQL Call Level Interface. The java.sql package contains classes and interfaces for JDBC API.

A list of popular interfaces of JDBC API are given below:

- Driver interface
- Connection interface
- Statement interface
- PreparedStatement interface
- CallableStatement interface
- ResultSet interface
- ResultSetMetaData interface
- DatabaseMetaData interface
- RowSet interface

Why Should We Use JDBC:-

Before JDBC, ODBC API was the database API to connect and execute the query with the database. But, ODBC API uses ODBC driver which is written in C language (i.e. platform dependent and unsecured). That is why Java has defined its own API (JDBC API) that uses JDBC drivers (written in Java language).

We can use JDBC API to handle database using Java

program and can perform the following activities:

- Connect to the database.
- Execute queries and update statements to the database.
- Retrieve the result received from the database.

Java Database Connectivity with MySQL:-

To connect Java application with the MySQL database, we need to follow 5 following steps.

In this example we are using MySql as the database. So we need to know following informations for the mysql database:

Driver class: The driver class for the mysql database is **com.mysql.jdbc.Driver**.

Connection URL: The connection URL for the mysql database is **jdbc:mysql://localhost:3306/lnt** where jdbc is the API, mysql is the database, localhost is the server name on which mysql is running, we may also use IP address, 3306 is the port number and lnt is the database name. We may use any database, in such case, we need to replace the lnt with our database name.

Username: The default username for the mysql database is **root**.

Password: It is the password given by the user at the time of installing the mysql database. In this example, we are going to use **admin** as the password.

Let's first create a table in the mysql database, but before creating table, we need to create database first.

```
create database lnt;

use lnt;

create table emp(id int(10),name varchar(40),age int(3));
```

Example to Connect Java Application with mysql database.

In this example, lnt is the database name, **root** is the username and **admin** is password.

```
import java.sql.*;

class MysqlCon

{

public static void main(String args[])

{

try

{
```

```
Class.forName("com.mysql.jdbc.Driver");
Connection con=DriverManager.getConnection(
"jdbc:mysql://localhost:3306/lnt","root","admin");
Statement stmt=con.createStatement();
ResultSet rs=stmt.executeQuery("select * from emp");
while(rs.next())
System.out.println(rs.getInt(1)+"   "+rs.getString(2)+"
"+rs.getString(3));
con.close();
}
catch(Exception e)
{
System.out.println(e);
} } }
```

To connect java application with the mysql database, **mysqlconnector.jar** file is required to be loaded.

Two ways to load the jar file:

1) Paste the mysqlconnector.jar file in jre/lib/ext folder
2) Set classpath

Paste the mysqlconnector.jar file in JRE/lib/ext folder:

Download the mysqlconnector.jar file. Go to jre/lib/ext folder and paste the jar file here.

Set classpath:

There are two ways to set the classpath:

temporary

permanent

How to set the temporary classpath:-

open command prompt and write:

C:>set classpath=c:\folder\mysql-connector-java-5.0.8-bin.jar;.;

How to set the permanent classpath:-

Go to environment variable then click on new tab. In variable name write classpath and in variable value paste the path to the mysqlconnector.jar file by appending mysqlconnector.jar;.; as C:\folder\mysql-connector-java-5.0.8-bin.jar;.;