



PARSHVANATH CHARITABLE TRUST'S

A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering
Data Science

Subject: SBL-OOPM

Class: SE-DS

Semester: III

A.Y. 2022-2023

Experiment No. 15

- ❖ **Aim :** To write a Java program to demonstrate the JDBC-ODBC connectivity.
- ❖ **Objectives :** To learn basic concepts of database connectivity.
- ❖ **Prerequisites :** Students should know disadvantages of Procedure oriented programming language & the need of OOPs concepts to overcome those disadvantages.
- ❖ **Software used :** jdk 1.6.0
- ❖ **Theory :**

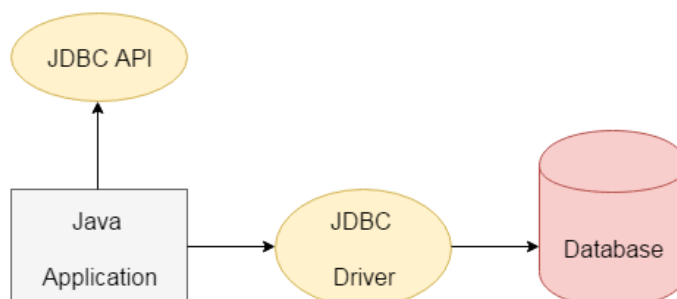
Java JDBC Tutorial

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, and
- Thin Driver

We have discussed the above four drivers in the next chapter.

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.





PARSHVANATH CHARITABLE TRUST'S

A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering
Data Science

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

A list of popular *classes* of JDBC API are given below:

- DriverManager class
- Blob class
- Clob class
- Types class

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

- Register the Driver class
 - Create connection
 - Create statement
 - Execute queries
 - Close connection
-

It avoids explicit connection closing step.

```
import java.sql.*;
```



PARSHVANATH CHARITABLE TRUST'S

A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering
Data Science

```
class MysqlCon{
    public static void main(String args[]){
        try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection con=DriverManager.getConnection(
                "jdbc:mysql://localhost:3306/sonoo","root","root");
            //here sonoo is database name, root is username and password
            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);}
    }
}
```

```
MariaDB [testingdocs]> CREATE TABLE IF NOT EXISTS employee (
->   emp_id INT(10) NOT NULL AUTO_INCREMENT,
->   name VARCHAR(45) DEFAULT NULL,
->   salary INT(10) DEFAULT NULL,
->   dept VARCHAR(100) DEFAULT NULL,
->   PRIMARY KEY (emp_id)
-> ) ENGINE=InnoDB ;
Query OK, 0 rows affected (0.91 sec)
```

```
MariaDB [testingdocs]> show tables ;
```

```
+-----+
| Tables_in_testingdocs |
+-----+
| employee               |
+-----+
1 row in set (0.00 sec)
```

```
MariaDB [testingdocs]> desc employee ;
```

Field	Type	Null	Key	Default	Extra
emp_id	int(10)	NO	PRI	NULL	auto_increment
name	varchar(45)	YES		NULL	



PARSHVANATH CHARITABLE TRUST'S

A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering
Data Science

```
C:\Windows\system32\cmd.exe - mysql -u root

| name | varchar(45) | YES | NULL |
| salary | int(10) | YES | NULL |
| dept | varchar(100) | YES | NULL |
+-----+-----+-----+-----+
4 rows in set (0.02 sec)

MariaDB [testingdocs]> INSERT INTO employee VALUES(1,"Surendra Kumar",10000,"Quality Assurance");
Query OK, 1 row affected (0.06 sec) 1

MariaDB [testingdocs]> INSERT INTO employee VALUES(2,"Regan M",18000,"Product Development");
Query OK, 1 row affected (0.06 sec) 2

MariaDB [testingdocs]> select * from employee ;
+-----+-----+-----+-----+
| emp_id | name | salary | dept |
+-----+-----+-----+-----+
| 1 | Surendra Kumar | 10000 | Quality Assurance |
| 2 | Regan M | 18000 | Product Development |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

CONCLUSION : Summaries what you understood from this lab.