

Java Database Connectivity with MySQL

Step 1: Download and Install Mysql

Please follow the steps given in the below link and install MySQL

<https://www.javatpoint.com/how-to-install-mysql>

Step 2: Create table using MySQL command line client

From the start menu, open the **MySQL Command Line Client** and provide the root user password.

Connect to the Database and create the table using the script below.

```
create table Employee_TBL
(
    EID int primary key,
    ENAME varchar(20),
    EMAIL varchar(30)
);
```

```
MySQL 5.7 Command Line Client
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 95
Server version: 5.7.10-log MySQL Community Server (GPL)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use SampleDB;
Database changed
mysql> create table Employee_TBL
-> (
-> EID int primary key,
-> ENAME varchar(20),
-> EMAIL varchar(30)
-> );
Query OK, 0 rows affected (1.15 sec)

mysql> Select * from Employee_TBL;
Empty set (0.10 sec)

mysql>
```

<https://www.tutorialspoint.com/mysql/mysql-create-tables.htm>

Step 3: Download Mysql driver

The MySQL driver can be downloaded from the below link

<https://static.javatpoint.com/src/jdbc/mysql-connector.jar>

Step 4: Add driver to classpath

In Eclipse Project, add the jdbc **mysql-connector** to the build path

Java Build Path -> Libraries ->Add External JARs -> Choose mysql-connector.jar

Step 5: Write Program

Refer the below link for a sample JDBC program

<https://www.javatpoint.com/example-to-connect-to-the-mysql-database>

Sample program using java.sql.Statement

```
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/SampleDB","root","root");
            //here SampleDB is database name, root is username and password
            Statement stmt=con.createStatement();
            ResultSet rs=stmt.executeQuery("select * from Employee_TBL");
            while(rs.next())
                System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getS
                    tring(3));
            con.close();
        }catch(Exception e){ System.out.println(e);}
    }
}
```

Sample program for java.sql.PreparedStatement

```
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/Samp
leDB","root","root"); //here SampleDB is database name, root is usernam
e and password
            String qry="insert into Employee_TBL values(?,?,?)";
            PreparedStatement ps = con.prepareStatement(qry);
            ps.setInt(1,100);
            ps.setString(2, "John");
            ps.setString(3, "john@wipro.com");
            int rec_count =ps.executeUpdate();
            if(rec_count==1)
```

```

        {
            System.out.println("Employee data inserted");
        }
        ps.close();
        con.close();
    }catch(Exception e){ System.out.println(e);}
}
}

```

Sample program for java.sql.CallableStatement

Step 1: Create procedure in **MySQL Command Line Client** tool and call it in the application using CallableStatement.

```

USE `sampledb`;
DROP procedure IF EXISTS `insert_Employee`;
DELIMITER $$
USE `sampledb`$$
CREATE PROCEDURE `insert_Employee` (in in_eid int, in in_ename
varchar(20), in in_email varchar(30))
BEGIN
    insert into employee_tbl values (in_eid, in_ename, in_email);
END$$
DELIMITER ;

```

```
MySQL 5.7 Command Line Client
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 100
Server version: 5.7.10-log MySQL Community Server (GPL)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE `sampledb`;
Database changed
mysql> DROP procedure IF EXISTS `insert_Employee`;
Query OK, 0 rows affected, 1 warning (0.13 sec)

mysql> DELIMITER $$
mysql> USE `sampledb`$$
Database changed
mysql> CREATE PROCEDURE `insert_Employee` (in in_eid int,in in_ename varchar(20),in in_email varchar(30))
-> BEGIN
-> insert into employee_tbl values(in_eid,in_ename,in_email);
-> END$$
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql>
```

```
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/SampleDB","root","root");
            //here SampleDB is database name, root is username and password
            CallableStatement cst = con.prepareCall("{call
            INSERT_EMPLOYEE(?,?,?)}");

            cst.setInt(1, 100);

            cst.setString(2, "John");

            cst.setString(3, "john@wipro.com");

            boolean issue=cst.execute();

            if(!issue)
            {
                System.out.println("Employee data inserted");
            }
        }
    }
}
```

```
        con.close();  
    } catch (Exception e) { System.out.println(e); }  
    }  
}
```