CREATE PROCEDURE `EMP`.`getEmpName`

(IN EMP\_ID INT, OUT EMP\_FIRST VARCHAR(255))

BEGIN

SELECT first INTO EMP\_FIRST

FROM Employees

WHERE ID = EMP\_ID;

END

mysql> DELIMITER //

mysql> create procedure getEmpName(in id int,out ename varchar(25))

-> begin

-> select emp\_name into ename from emp1 where emp\_id=id ;

-> end //

Query OK, 0 rows affected (0.02 sec)

mysql> call getEmpName(100,@nam);

-> //

Query OK, 1 row affected (0.00 sec)

mysql> delimiter ;

mysql> select @num;

+------+

| @num |

+------+

| NULL |

+------+

1 row in set (0.00 sec)

import java.sql.\*;

public class JDBCExample {

// JDBC driver name and database URL

static final String JDBC\_DRIVER = "com.mysql.jdbc.Driver";

static final String DB\_URL = "jdbc:mysql://localhost/EMP";

// Database credentials

static final String USER = "username";

static final String PASS = "password";

public static void main(String[] args) {

Connection conn = null;

CallableStatement stmt = null;

try{

//STEP 2: Register JDBC driver

Class.forName("com.mysql.jdbc.Driver");

//STEP 3: Open a connection

System.out.println("Connecting to database...");

conn = DriverManager.getConnection(DB\_URL,USER,PASS);

//STEP 4: Execute a query

System.out.println("Creating statement...");

String sql = "{call getEmpName (?, ?)}";

stmt = conn.prepareCall(sql);

//Bind IN parameter first, then bind OUT parameter

int empID = 102;

stmt.setInt(1, empID); // This would set ID as 102

// Because second parameter is OUT so register it

stmt.registerOutParameter(2, java.sql.Types.VARCHAR);

//Use execute method to run stored procedure.

System.out.println("Executing stored procedure..." );

stmt.execute();

//Retrieve employee name with getXXX method

String empName = stmt.getString(2);

System.out.println("Emp Name with ID:" +

empID + " is " + empName);

stmt.close();

conn.close();

}catch(SQLException se){

//Handle errors for JDBC

se.printStackTrace();

}catch(Exception e){

//Handle errors for Class.forName

e.printStackTrace();

}finally{

//finally block used to close resources

try{

if(stmt!=null)

stmt.close();

}catch(SQLException se2){

}// nothing we can do

try{

if(conn!=null)

conn.close();

}catch(SQLException se){

se.printStackTrace();

}//end finally try

}//end try

System.out.println("Goodbye!");

}//end main

}//end JDBCExample

**import** java.sql.\*;

**public** **class** JDBCExample {

// JDBC driver name and database URL

**static** **final** String *JDBC\_DRIVER* = "com.mysql.jdbc.Driver";

**static** **final** String *DB\_URL* = "jdbc:mysql://172.20.35.21/emp";

// Database credentials

**static** **final** String *USER* = "abc";

**static** **final** String *PASS* = "abc";

**public** **static** **void** main(String[] args) {

Connection conn = **null**;

CallableStatement stmt = **null**;

**try**{

//STEP 2: Register JDBC driver

Class.*forName*("com.mysql.jdbc.Driver");

//STEP 3: Open a connection

System.*out*.println("Connecting to database...");

conn = DriverManager.*getConnection*(*DB\_URL*,*USER*,*PASS*);

//STEP 4: Execute a query

System.*out*.println("Creating statement...");

String sql = "{call getEmpName (?, ?)}";

stmt = conn.prepareCall(sql);

//Bind IN parameter first, then bind OUT parameter

**int** empID = 100;

stmt.setInt(1, empID); // This would set ID as 102

// Because second parameter is OUT so register it

stmt.registerOutParameter(2, java.sql.Types.*VARCHAR*);

//Use execute method to run stored procedure.

System.*out*.println("Executing stored procedure..." );

stmt.execute();

//Retrieve employee name with getXXX method

String empName = stmt.getString(2);

System.*out*.println("Emp Name with ID:" +

empID + " is " + empName);

stmt.close();

conn.close();

}**catch**(SQLException se){

//Handle errors for JDBC

se.printStackTrace();

}**catch**(Exception e){

//Handle errors for Class.forName

e.printStackTrace();

}**finally**{

//finally block used to close resources

**try**{

**if**(stmt!=**null**)

stmt.close();

}**catch**(SQLException se2){

}// nothing we can do

**try**{

**if**(conn!=**null**)

conn.close();

}**catch**(SQLException se){

se.printStackTrace();

}//end finally try

}//end try

System.*out*.println("Goodbye!");

}//end main

}//end JDBCExample