

Java Full Stack Lab Manual - JDBC Experiment

Aim

To write a JDBC application that interacts with the database using CallableStatement to:

- Create stored procedures for insert and fetch operations.
- Call those procedures from a Java application.

Technologies Used

Java, JDBC, MySQL Database, MySQL JDBC Driver

Database Table Structure

Table Name: Employee

Columns:

- EmpID (INT)
- Name (VARCHAR)
- Salary (DOUBLE)

Procedure

1. Create the Employee table in the MySQL database.
2. Create a stored procedure to insert a new record.
3. Create a stored procedure to retrieve the salary for a given employee ID.
4. Use CallableStatement in Java to call these procedures and display results.

Java Full Stack Lab Manual - JDBC Experiment

Stored Procedures (MySQL)

```
-- a. Procedure to insert one record
DELIMITER //
CREATE PROCEDURE InsertEmployee(IN emp_id INT, IN emp_name VARCHAR(50), IN emp_salary DOUBLE)
BEGIN
    INSERT INTO Employee(EmpID, Name, Salary) VALUES (emp_id, emp_name, emp_salary);
END //
DELIMITER ;

-- b. Procedure to retrieve salary
DELIMITER //
CREATE PROCEDURE GetSalary(IN emp_id INT, OUT emp_salary DOUBLE)
BEGIN
    SELECT Salary INTO emp_salary FROM Employee WHERE EmpID = emp_id;
END //
DELIMITER ;
```

Java Full Stack Lab Manual - JDBC Experiment

Java Program

```
import java.sql.*;

public class EmployeeCallableApp {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/college";
        String username = "root";
        String password = "your_password";

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection(url, username, password);

            // a. Call procedure to insert employee
            CallableStatement csInsert = con.prepareCall("{call InsertEmployee(?, ?, ?)}");
            csInsert.setInt(1, 101);
            csInsert.setString(2, "Rahul");
            csInsert.setDouble(3, 50000);
            csInsert.execute();
            System.out.println("Inserted employee record.");

            // b. Call procedure to get salary
            CallableStatement csSalary = con.prepareCall("{call GetSalary(?, ?)}");
            csSalary.setInt(1, 101);
            csSalary.registerOutParameter(2, Types.DOUBLE);
            csSalary.execute();

            double salary = csSalary.getDouble(2);
            System.out.println("Salary of employee 101 is: " + salary);

            con.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Java Full Stack Lab Manual - JDBC Experiment

Expected Output

```
Inserted employee record.  
Salary of employee 101 is: 50000.0
```

Result

Successfully created and called stored procedures using CallableStatement in a JDBC application to insert an employee record and retrieve salary based on employee ID.