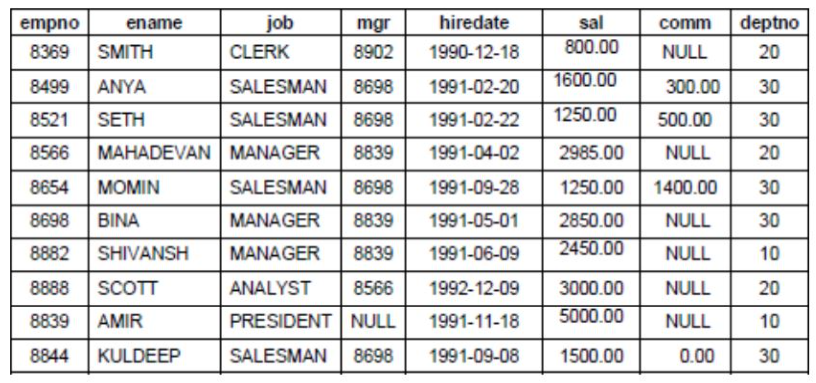
DAY 28  
TASK -8

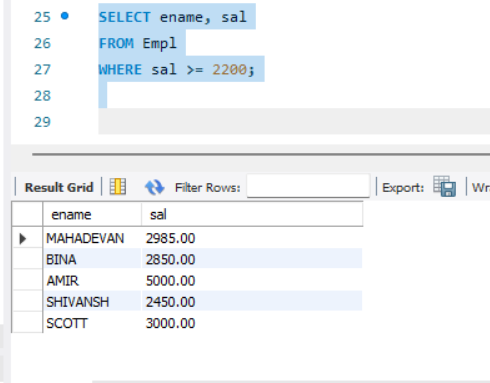
SQL

Create a Table Empl to store employee details as shown below and write statement for following queries based on table.

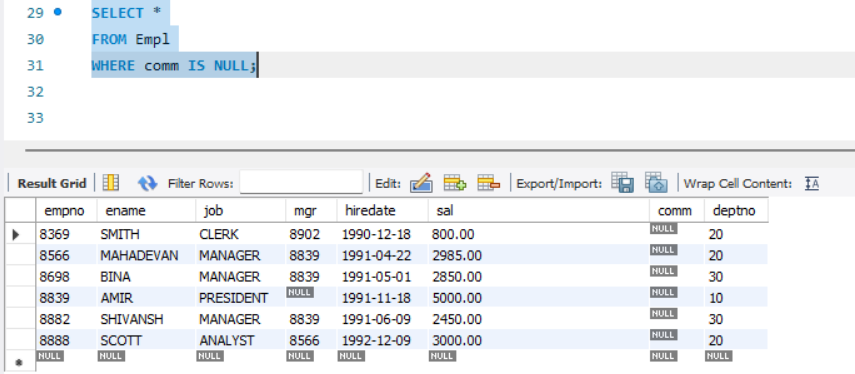


1.) Consider the Employee table and write SQL command to get the following.

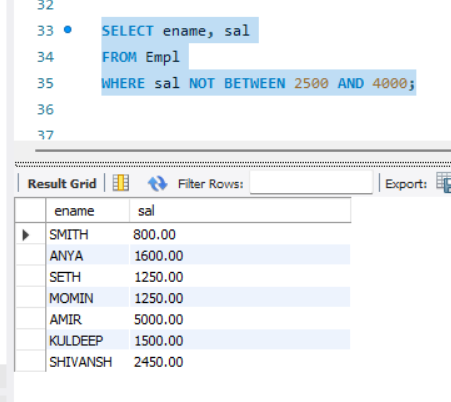
A. Write a query to display EName and Sal of employees whose salary are greater than or equal to 2200?



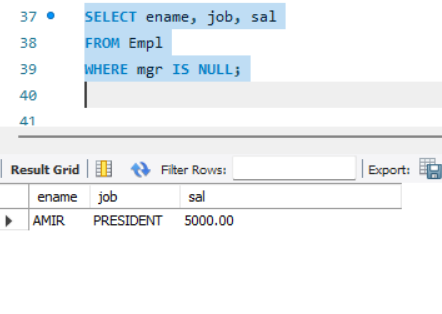
B. Write a query to display details of employees who are not getting commission?



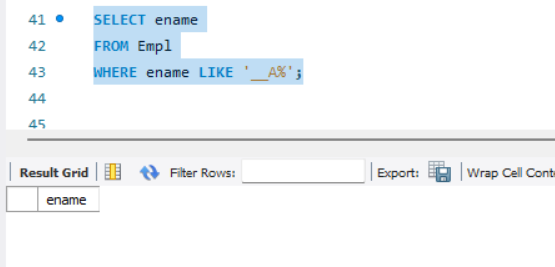
C. Write a query to display employee name and salary of those employees who don't have their salary in the range of 2500 to 4000?



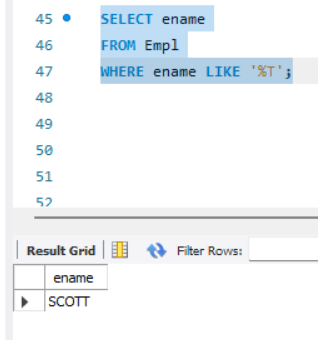
D. Write a query to display the name, job title and salary of employees who don't have a manager?



E. Write a query to display the name of an employee whose name contains "A" as third alphabet



F. Write a query to display the name of an employee whose name contains "T" as last alphabet



Complete CODE -  
  
CREATE DATABASE Employee;

USE Employee;

CREATE TABLE Empl (

empno INT PRIMARY KEY,

ename VARCHAR(50),

job VARCHAR(50),

mgr INT,

hiredate DATE,

sal DECIMAL(10, 2),

comm DECIMAL(10, 2),

deptno INT

);

INSERT INTO Empl VALUES (8369, 'SMITH', 'CLERK', 8902, '1990-12-18', 800.00, NULL, 20);

INSERT INTO Empl VALUES (8499, 'ANYA', 'SALESMAN', 8698, '1991-02-20', 1600.00, 300.00, 30);

INSERT INTO Empl VALUES (8521, 'SETH', 'SALESMAN', 8698, '1991-02-22', 1250.00, 500.00, 30);

INSERT INTO Empl VALUES (8566, 'MAHADEVAN', 'MANAGER', 8839, '1991-04-22', 2985.00, NULL, 20);

INSERT INTO Empl VALUES (8654, 'MOMIN', 'SALESMAN', 8698, '1991-09-28', 1250.00, 1400.00, 30);

INSERT INTO Empl VALUES (8698, 'BINA', 'MANAGER', 8839, '1991-05-01', 2850.00, NULL, 30);

INSERT INTO Empl VALUES (8882, 'SHIVANSH', 'MANAGER', 8839, '1991-06-09', 2450.00, NULL, 30);

INSERT INTO Empl VALUES (8888, 'SCOTT', 'ANALYST', 8566, '1992-12-09', 3000.00, NULL, 20);

INSERT INTO Empl VALUES (8839, 'AMIR', 'PRESIDENT', NULL, '1991-11-18', 5000.00, NULL, 10);

INSERT INTO Empl VALUES (8844, 'KULDEEP', 'SALESMAN', 8698, '1991-09-08', 1500.00, 0.00, 30);

SELECT ename, sal

FROM Empl

WHERE sal >= 2200;

SELECT \*

FROM Empl

WHERE comm IS NULL;

SELECT ename, sal

FROM Empl

WHERE sal NOT BETWEEN 2500 AND 4000;

SELECT ename, job, sal

FROM Empl

WHERE mgr IS NULL;

SELECT ename

FROM Empl

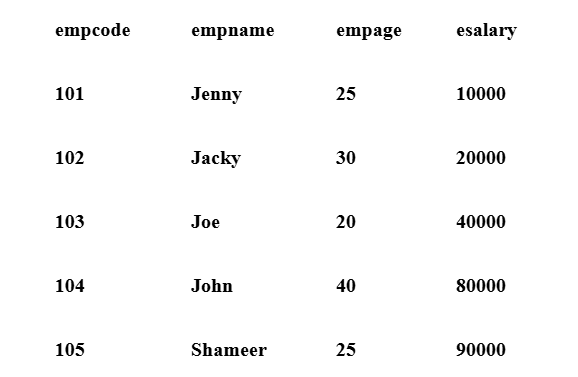
WHERE ename LIKE '\_\_A%';

SELECT ename

FROM Empl

WHERE ename LIKE '%T';

2.) Write a program for JDBC to connectivity and insert the below data.



SQL -  
Created a Database Emp and a table Employee   
  
  
CREATE DATABASE Emp;

USE Emp;

CREATE TABLE Employee(

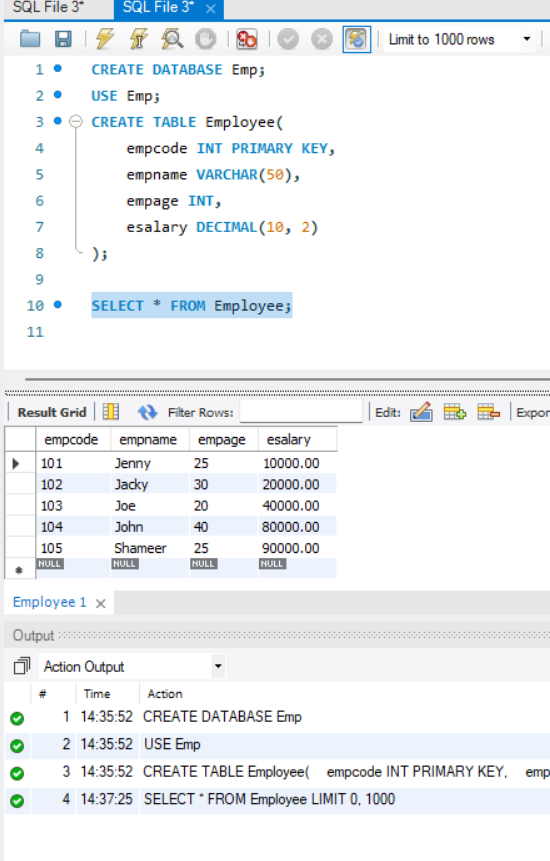
empcode INT PRIMARY KEY,

empname VARCHAR(50),

empage INT,

esalary DECIMAL(10, 2)

);



Java -   
  
EmployeeInsert.java

package com.example;  
import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

public class EmployeeInsert {

public static void main(String[] args) {

String url = "jdbc:mysql://localhost:3306/emp";

String user = "root"; // Replace with your MySQL username

String password = "Lunar"; // Replace with your MySQL password

String query = "INSERT INTO Employee (empcode, empname, empage, esalary) VALUES (?, ?, ?, ?)";

// Employee data

Object[][] employees = {

{101, "Jenny", 25, 10000},

{102, "Jacky", 30, 20000},

{103, "Joe", 20, 40000},

{104, "John", 40, 80000},

{105, "Shameer", 25, 90000}

};

try {

// Connect to database

Connection con = DriverManager.*getConnection*(url, user, password);

PreparedStatement pst = con.prepareStatement(query);

// Insert records using loop

for (Object[] emp : employees) {

pst.setInt(1, (int) emp[0]);

pst.setString(2, (String) emp[1]);

pst.setInt(3, (int) emp[2]);

pst.setDouble(4, (double) (int) emp[3]); // Cast salary to double

pst.executeUpdate();

}

System.*out*.println("All records inserted successfully!");

pst.close();

con.close();

} catch (Exception e) {

e.printStackTrace();

}

}

}

Pom.xml  
  
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>jdbcapp</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.source>24</maven.compiler.source>

<maven.compiler.target>24</maven.compiler.target>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<!-- MySQL JDBC Connector -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.33</version>

</dependency>

</dependencies>

</project>

Mongo DB

**Product JSON:** <https://github.com/rvsp/database/blob/master/mongodb/product.json>

For the following question write the corresponding MongoDB queries

1. Find all the information about each products

db.products.find({})

1. Find the product price which are between 400 to 800

db.products.find({ product\_price: { $gte: 400, $lte: 800 } })

1. Find the product price which are not between 400 to 600

db.products.find({ $or: [ { product\_price: { $lt: 400 } }, { product\_price: { $gt: 600 } } ] })

1. List the four product which are greater than 500 in price

db.products.find({ product\_price: { $gt: 500 } }).limit(4)

1. Find the product name and product material of each products

db.products.find({}, { product\_name: 1, product\_material: 1, \_id: 0 })

1. Find the product with a row id of 10

db.products.find({ id: 10 })

1. Find only the product name and product material

db.products.find({}, { product\_name: 1, product\_material: 1, \_id: 0 })

1. Find all products which contain the value of soft in product material

db.products.find({ product\_material: /soft/i })

1. Find products which contain product color indigo and product price 492.00

db.products.find({ product\_color: "indigo", product\_price: 492 })

1. Delete the products which product price value are 28

db.products.deleteMany({ product\_price: 28 })