```sql
-- Payroll Calculation Project -Sonal Ranpise
-- DESCRIPTION
-- An HR of the company wants to analyze the performance of the employees and calculate their salary.
-- Objective
-- The database design helps to retrieve the employees’ details based on certain criteria which are listed below.
-- Tasks to be performed:
DROP DATABASE Payroll_Calculation;
-- TASK:01
-- Write a query to create an employee table and a department table.
CREATE DATABASE Payroll_Calculation;
USE Payroll_Calculation;
-- create an employee table
CREATE TABLE Payroll_Calculation.employee_table
(
emp_id INT PRIMARY KEY NOT NULL,
f_name VARCHAR(100) NOT NULL,
l_name  VARCHAR (100) NOT NULL,
job_id  VARCHAR (100) NOT NULL,
salary DECIMAL(10,2) NOT NULL,
manager_id INT NOT NULL,
dept_id VARCHAR (100) NOT NULL
)
ENGINE = INNODB;
DESCRIBE Payroll_Calculation.employee_table;

-- create an department table
CREATE TABLE Payroll_Calculation.department_table
(
dept_id INT PRIMARY KEY NOT NULL,
dept_name VARCHAR(100) NOT NULL,
location VARCHAR(100) NOT NULL,
manager_id INT NOT NULL,
elocation_id  VARCHAR(100) NOT NULL
)
ENGINE = INNODB;
DESCRIBE Payroll_Calculation.department_table;

-- TASK:02
-- Write a query to insert values in the employee and department tables.
-- Insert values in  employee department table
INSERT INTO Payroll_Calculation.employee_table(emp_id,f_name,l_name,job_id,salary,manager_id,dept_id)
VALUES
("101", "ankit", "jain", "HP124", "200000", "2", "24" ),
("102", "sarvesh", "patel", "HP123", "150000", "2", "24" ),
("103", "krishna", "gee", "HP125", "500000", "5", "44" ),
("104", "rana", "gee", "HP122", "250000", "3", "54" ),
("105", "soniya", "jain", "HP121", "400000", "1", "22" ),
("106", "nithin", "kumar", "HP120", "300000", "4", "34" ),
("107", "karan", "patel", "HP126", "300001", "2", "34" ),
("108", "shilpa", "jain", "HP127", "300001", "5", "24" ),
("109", "mukesh", "singh", "HP128", "300001", "4", "44" );
SELECT*FROM Payroll_Calculation.employee_table;

-- Insert values in  department department table
INSERT INTO Payroll_Calculation.department_table(dept_id,dept_name,location,manager_id,elocation_id)
VALUES
("22", "admistration", "uk", "1", "218"),
("24", "production", "india", "2", "212"),
("34", "development", "india", "4", "212"),
("44", "communication", "usa", "5", "220"),
("54", "maintenance", "usa", "3", "220");
SELECT*FROM Payroll_Calculation.department_table;

-- TASK:03
-- Write a query to create a view of the employee and department tables.
```

```sql
CREATE VIEW emp AS SELECT f_name, l_name, salary,dept_name,location,emp_id FROM Payroll_Calculation.employee_table,Payroll_Calculation.department_table
 WHERE l_name = "Jain";


-- TASK:04
-- Write a query to display first name and last name of the employees from the employee table and an SQL basics ,
-- view table if the employee’s salary in the SQL basics table is greater than the salary in the employee table.
SELECT  f_name,l_name FROM Payroll_Calculation.employee_table
WHERE salary >  ( SELECT AVG(salary)
FROM employee_table );

-- TASK:05
-- Write a query to change the delimiter to //.
delimiter //

-- TASK:06
-- Write a query to create a stored procedure using an employee table if the salary is greater than or equal to 250000.
use Payroll_Calculation;
SELECT * from employee_table;
delimiter &&
CREATE PROCEDURE top_salarys()
BEGIN
SELECT job_id,f_name,salary FROM Payroll_Calculation.employee_table where salary>=250000;
END &&
delimiter ;;

-- TASK:07
-- Write a query to execute the stored procedure.
call top_salarys();



-- TASK:08
-- Write a query to create a stored procedure with one parameter using ORDER BY salary in descending order, and execute the stored procedure.
delimiter //
CREATE PROCEDURE sort_salarys(IN var INT)
BEGIN
SELECT job_id,f_name,salary
FROM employee_table ORDER BY salary DESC LIMIT var;
end //
delimiter ;
call sort_salarys(3);
```