Database Task: Employee Management System

1.Database Schema:

Create Tech_Employees Table

```
create table Tech_Employees(
Employee_id int not null primary key,
Emp_Name varchar2(100) not null,
Department_id int,
hire_date Date,
FOREIGN KEY (Department_id) REFERENCES Tech_Departments(Department_id)
);
```

Insert Data to Tech_Employees Table

```
Insert into Tech_Employees (
Employee_id ,
Emp_Name ,
Department_id,
hire_date)
Values(123458,'Aswathy',124,to_date('12-Aug-2024'));

Insert into Tech_Employees (
Employee_id ,
Emp_Name ,
Department_id,
hire_date)
Values(1234589,'Laya',125,to_date('01-Sep-2024'));

Insert into Tech_Employees (
Employee_id ,
Emp_Name ,
Department_id,
hire_date)
Values(1234590,'Theertha',126,to_date('11-Feb-2024'));
```

```
114
115
116 select * from Tech_Employees;
```

	EMPLOYEE_ID	EMP_NAME	DEPARTMENT_ID	HIRE_DATE
1	123458	Aswathy	124	8/12/2024
2	1234589	Laya	125	9/1/2024
3	1234590	Theertha	126	2/11/2024

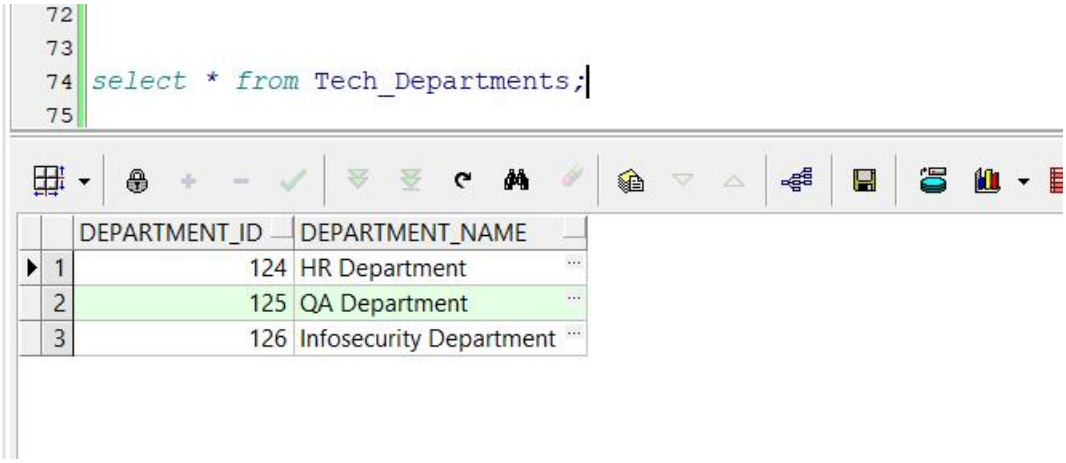
Create Tech_Departments Table

```
Create table Tech_Departments(
Department_id int not null primary key,
Department_name varchar2(100) not null);
```

Insert Data to Tech_Employees Table

```
insert into Tech_Departments(
Department_id ,
Department_name) values(124,'HR Department');
insert into Tech_Departments(
Department_id ,
Department_name) values(125,'QA Department');
insert into Tech_Departments(
Department_id ,
Department_name) values(126,'Infosecurity Department');


select * from Tech_Departments;
```



Create Tech_Salaries Table

```
Create table Tech_Salaries(Employee_id int primary key,
Base_salary decimal(10,2) not null,
Bonus decimal(10, 2) NOT NULL,
Deductions decimal(10, 2) NOT NULL,
FOREIGN KEY (Employee_id) REFERENCES Tech_Employees(Employee_id));
```

Insert Data to Tech_Salaries Table

```
insert into Tech_Salaries(Employee_id ,
Base_salary ,
Bonus ,
Deductions
) values(123458,15000,1000,250);


insert into Tech_Salaries(Employee_id ,
Base_salary ,
Bonus ,
Deductions
) values(1234589,12000,1000,350);


insert into Tech_Salaries(Employee_id ,
Base_salary ,
Bonus ,
Deductions
) values(1234590,13500,2000,450);
```

```
113
114
115 select * from Tech_Salaries;
```

	EMPLOYEE_ID	BASE_SALARY	BONUS	DEDUCTIONS
1	123458	15000.00	1000.00	250.00
2	1234589	12000.00	1000.00	350.00
3	1234590	13500.00	2000.00	450.00

2. SQL Queries:

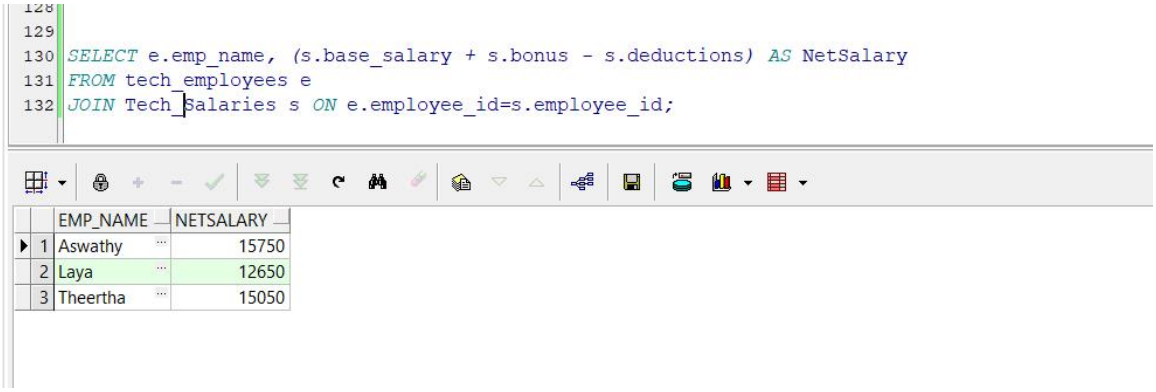
- **List all employees along with their department names.**

```
select e.emp_name,d.department_name from tech_employees e inner join
Tech_Departments d on e.department_id=d.department_id;
```

	EMP_NAME	DEPARTMENT_NAME
1	Aswathy	HR Department
2	Laya	QA Department
3	Theertha	Infosecurity Department

- **Calculate the net salary for each employee using:Net Salary = BaseSalary + Bonus - Deductions.**

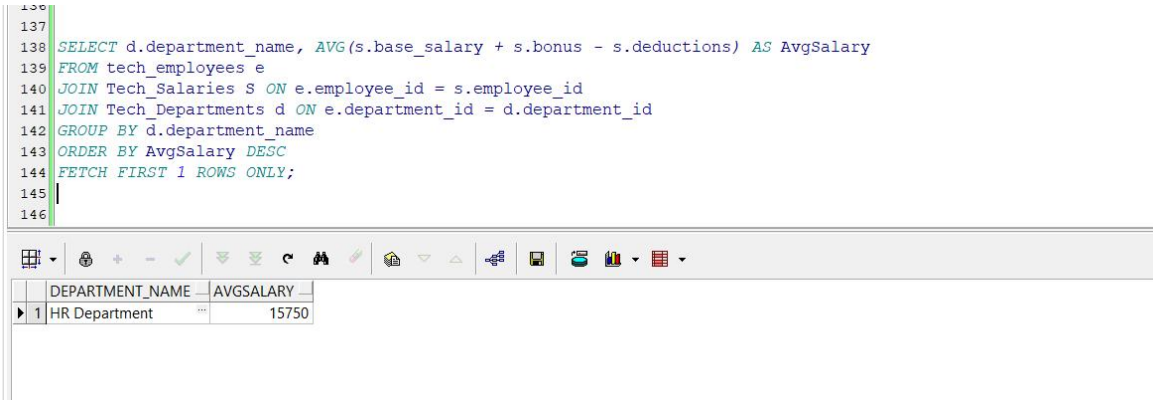
```
SELECT e.emp_name, (s.base_salary + s.bonus - s.deductions) AS
NetSalary
FROM tech_employees e
JOIN Tech_Salaries s ON e.employee_id=s.employee_id;
```



	EMP_NAME	NETSALARY
1	Aswathy	15750
2	Laya	12650
3	Theertha	15050

- **Identify the department with the highest average salary.**

```
SELECT d.department_name, AVG(s.base_salary + s.bonus - s.deductions)
AS AvgSalary
FROM tech_employees e
JOIN Tech_Salaries S ON e.employee_id = s.employee_id
JOIN Tech_Departments d ON e.department_id = d.department_id
GROUP BY d.department_name
ORDER BY AvgSalary DESC
FETCH FIRST 1 ROWS ONLY;
```



3. Stored Procedures:

- **Add Employee**

```
CREATE or Replace PROCEDURE Tech_AddEmployee(Emp_id     int,
                                        empname    in varchar2,
                                        dep_id     in int,
                                        hiredate   in Date,
                                        p_response OUT VARCHAR2)
BEGIN
    IF Emp_id is not null THEN
Insert into Tech_Employees
  (Employee_id, Emp_Name, Department_id, hire_date)
Values
  (Emp_id, empname, dep_id, hiredate);
COMMIT;
 p_response := 'Added Successfully';
ELSE
 p_response := 'Failed';
END IF;
END;
```

- **Update Salary**

```
CREATE OR REPLACE PROCEDURE Tech_UpdateSalary(
    empID      IN INT,
    baseSalary IN NUMBER,
    bonus      IN NUMBER,
    deductions IN NUMBER,
    p_response OUT VARCHAR2
)
IS
BEGIN
    IF empID IS NOT NULL THEN
        UPDATE Tech_Salaries
        SET base_salary = baseSalary,
            bonus = bonus,
            deductions = deductions
        WHERE employee_id = empID;

        COMMIT;

        p_response := 'Updated Successfully';
    ELSE
        p_response := 'Failed';
    END IF;
END;
```

- **Calculate Payroll**

```
CREATE OR REPLACE PROCEDURE Tech_CalculatePayroll(
    deptID IN INT,
    totalPayroll OUT NUMBER
)
IS
BEGIN
    IF deptID IS NULL THEN
        SELECT SUM(s.base_salary + s.bonus - s.deductions) INTO
totalPayroll
        FROM Tech_Employees e
        JOIN Tech_Salaries s ON e.employee_id = s.employee_id;
    ELSE
        SELECT SUM(s.base_salary + s.bonus - s.deductions) INTO
totalPayroll
        FROM Tech_Employees e
        JOIN Tech_Salaries s ON e.employee_id = s.employee_id
        WHERE e.department_id = deptID;
    END IF;
END;
```

4. Views

EmployeeSalaryView:

```
CREATE VIEW Tech_EmployeeSalaryView AS
SELECT e.emp_name, d.department_name, s.base_salary, s.bonus,
s.deductions,
    (s.base_salary + s.bonus - s.deductions) AS NetSalary
FROM Tech_Employees e
JOIN Tech_Salaries s ON e.employee_id = s.employee_id
JOIN Tech_Departments d ON e.department_id = d.department_id;
```

HighEarnerView:

```
CREATE VIEW Tech_HighEarnerView AS
SELECT e.emp_name, d.department_name, (s.base_salary+ s.bonus -
s.deductions) AS NetSalary
FROM Tech_Employees e
JOIN Tech_Salaries s ON e.employee_id = s.employee_id
JOIN Tech_Departments d ON e.department_id = d.department_id
WHERE (s.base_salary + s.bonus - s.deductions) > 20000;
```

5. Bonus Tasks:

```
CREATE TABLE Tech_SalaryHistory (
    employee_id INT,
    BaseSalary DECIMAL(10, 2),
    Bonus DECIMAL(10, 2),
    Deductions DECIMAL(10, 2),
    ChangeDate TIMESTAMP,
    FOREIGN KEY (employee_id) REFERENCES Employees(employee_id)
);


CREATE TRIGGER LogSalaryUpdate
AFTER UPDATE ON Salaries
FOR EACH ROW
BEGIN
    INSERT INTO Tech_SalaryHistory (employee_id, BaseSalary, Bonus,
Deductions, ChangeDate)
    VALUES (NEW.employee_id, NEW.BaseSalary, NEW.Bonus, NEW.Deductions,
NOW());
END;
```