**Week 3. Database Schema for a Employee-pay scenario**

Employee (**emp\_id:**integer,emp\_name:string)

Department (**dept\_id:integer**,dept\_name:string)

Pay details (**emp\_id : integer**,**dept\_id: integer**, basic: integer, deductions: integer, additions: integer, DOJ: date)

Payroll **(emp\_id : integer**, pay\_date: date)

**For the above schema, perform the following—**

1. **Create Tables:** Write the SQL statements to create the employee, department, paydetails, and payroll tables as described above. Ensure that the paydetails table references both the employee and department tables, and the payroll table references the employee table. Include appropriate primary keys, foreign keys, and basic integrity constraints.

|  |
| --- |
| CREATE TABLE employee(  emp\_id int PRIMARY KEY,  emp\_name varchar(50));    CREATE TABLE department (  dept\_id int PRIMARY KEY,  dept\_name varchar(50));    CREATE TABLE paydetails(  emp\_id int,dept\_id int,basic int,deductions int,additions int,doj date,  FOREIGN KEY(emp\_id) REFERENCES employee(emp\_id),  FOREIGN KEY(dept\_id) REFERENCES department (dept\_id));    CREATE TABLE payroll(  emp\_id int ,pay\_date date,  FOREIGN KEY(emp\_id) REFERENCES employee(emp\_id)); |

1. **Insert Data**: Insert 5 records into each of the tables with the given data.

**employee:**

|  |  |
| --- | --- |
| **emp\_id** | **emp\_name** |
| 1 | John Doe |
| 2 | Jane Smith |
| 3 | Michael Johnson |
| 4 | Emily Brown |
| 5 | David Lee |

**department:**

|  |  |
| --- | --- |
| **dept\_id** | **dept\_name** |
| 101 | Human Resources |
| 102 | Finance |
| 103 | IT |
| 104 | Marketing |
| 105 | Sales |

**paydetails:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **emp\_id** | **dept\_id** | **basic** | **deductions** | **additions** | **doj** |
| 1 | 101 | 50000 | 5000 | 2000 | 2022-01-15 |
| 2 | 102 | 60000 | 6000 | 2500 | 2021-03-22 |
| 3 | 103 | 55000 | 4500 | 3000 | 2023-07-01 |
| 4 | 104 | 52000 | 4000 | 1500 | 2020-10-30 |
| 5 | 105 | 58000 | 5500 | 1800 | 2019-08-14 |

**payroll:**

|  |  |
| --- | --- |
| **emp\_id** | **pay\_date** |
| 1 | 2024-07-01 |
| 2 | 2024-07-02 |
| 3 | 2024-07-02 |
| 4 | 2024-07-03 |
| 5 | 2024-07-05 |

|  |
| --- |
| INSERT INTO employee VALUES(1,'John Doe');  INSERT INTO employee VALUES(2,'Jane Smith');  INSERT INTO employee VALUES(3,'Michael Johnson');  INSERT INTO employee VALUES(4,'Emily Brown');  INSERT INTO employee VALUES(5,'David Lee');    INSERT INTO department VALUES(101,'Human Resources');  INSERT INTO department VALUES(102,'Finance');  INSERT INTO department VALUES(103,'IT');  INSERT INTO department VALUES(104,'Marketing');  INSERT INTO department VALUES(105,'Sales');    INSERT INTO paydetails VALUES(1,101,50000,5000,2000,'2022-01-15');  INSERT INTO paydetails VALUES(2,102,60000,6000,2500,'2021-03-22');  INSERT INTO paydetails VALUES(3,103,55000,4500,3000,'2023-07-01');  INSERT INTO paydetails VALUES(4,104,52000,4000,1500,'2020-10-30');  INSERT INTO paydetails VALUES(5,105,58000,5500,1800,'2019-08-14');    INSERT INTO payroll VALUES(1,'2024-07-01');  INSERT INTO payroll VALUES(2,'2024-07-02');  INSERT INTO payroll VALUES(3,'2024-07-02');  INSERT INTO payroll VALUES(4,'2024-07-03');  INSERT INTO payroll VALUES(5,'2024-07-05'); |

1. Retrieve the emp\_id and emp\_name of employees, ordered by department name (dept\_name).

|  |
| --- |
| SELECT e.emp\_id,e.emp\_name  FROM employee e,department d,paydetails p  WHERE e.emp\_id=p.emp\_id AND d.dept\_id=p.dept\_id  ORDER BY d.dept\_name; |

1. List the names (emp\_name) of employees who joined after March 22, 2021.

|  |
| --- |
| SELECT e.emp\_name  FROM employee e,paydetails p  WHERE e.emp\_id=p.emp\_id AND doj>'2021-03-22'; |

1. Display the dept\_id, dept\_name, and count of employees (emp\_id) in each department.

|  |
| --- |
| SELECT d.dept\_id,d.dept\_name,count(e.emp\_id)  FROM employee e,department d,paydetails p  WHERE e.emp\_id=p.emp\_id AND d.dept\_id=p.dept\_id  GROUP BY d.dept\_id; |

1. Find the names (emp\_name) of employees whose net salary (basic salary + additions - deductions) exceeds 50,000.

|  |
| --- |
| SELECT e.emp\_name  FROM employee e,paydetails p  WHERE e.emp\_id = p.emp\_id AND (basic+additions-deductions) >50000; |

1. Fetch the emp\_name, dept\_id, dept\_name, basic, additions, deductions, doj, and pay\_date for the employee with emp\_id 5.

|  |
| --- |
| SELECT e.emp\_name,d.dept\_id,d.dept\_name,p.basic,p.additions,p.deductions,p.doj,r.pay\_date  FROM employee e,department d,paydetails p,payroll r  WHERE e.emp\_id=p.emp\_id AND d.dept\_id=p.dept\_id  AND e.emp\_id=r.emp\_id  AND e.emp\_id=5; |

1. Create a view named employee\_pay\_details that includes emp\_name, dept\_name, basic, deductions, and netsalary (calculated as basic + additions - deductions) for each employee.

|  |
| --- |
| CREATE VIEW employee\_pay\_details  AS SELECT e.emp\_name,d.dept\_name,basic,deductions,  (basic-deductions+additions)  AS netsalary  FROM employee e,department d,paydetails p  WHERE e.emp\_id=p.emp\_id AND d.dept\_id=p.dept\_id;  SELECT \* FROM employee\_pay\_details; |

1. Create another view named employee\_net\_salary that displays emp\_name and netsalary (calculated as basic + additions - deductions) for each employee.

|  |
| --- |
| CREATE VIEW employee\_net\_salary  AS SELECT e.emp\_name,(basic+additions-deductions)  AS netsalary  FROM employee e,paydetails p  WHERE e.emp\_id=p.emp\_id;  SELECT \* FROM employee\_net\_salary; |