```
create database class_4;
use class_4;

/* Create table and add dummy data for below questions:

Employees:
Columns: EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL,
PHONE_NUMBER, HIRE_DATE, JOB_ID, SALARY, COMMISSION_PCT,
MANAGER_ID, DEPARTMENT_ID

Departments:
Columns: DEPARTMENT_ID, DEPARTMENT_NAME, MANAGER_ID, LOCATION_ID*/

-- Create the Departments table
CREATE TABLE Departments (
   DEPARTMENT_ID INT PRIMARY KEY,
   DEPARTMENT_NAME VARCHAR(100),
   MANAGER_ID INT,
   LOCATION_ID INT
);

-- Insert 10 rows of dummy data into Departments table
INSERT INTO Departments VALUES
   (1, 'HR', 1, 101),
   (2, 'Finance', 2, 102),
   (3, 'Marketing', 3, 103),
   (4, 'Sales', 5, 104),
   (5, 'IT', 6, 105),
   (6, 'Operations', 9, 106),
   (7, 'Legal', 10, 107),
   (8, 'Research', 7, 108),
   (9, 'Customer Service', 4, 109),
   (10, 'Product Development', 8, 110);

-- Create the Employees table
CREATE TABLE Employees (
   EMPLOYEE_ID INT PRIMARY KEY,
   FIRST_NAME VARCHAR(50),
   LAST_NAME VARCHAR(50),
   EMAIL VARCHAR(100),
   PHONE_NUMBER VARCHAR(20),
   HIRE_DATE DATE,
   JOB_ID VARCHAR(10),
   SALARY DECIMAL(10, 2),
   COMMISSION_PCT DECIMAL(5, 2),
   MANAGER_ID INT,
   DEPARTMENT_ID INT,
   FOREIGN KEY (DEPARTMENT_ID) REFERENCES Departments(DEPARTMENT_ID)
);

-- Insert 10 rows of dummy data into Employees table
INSERT INTO Employees VALUES
   (1, 'John', 'Doe', 'john.doe@example.com', '555-123-4567', '2023-01-15', 'MAN001', 60000.00, NULL, NULL, 1),
   (2, 'Jane', 'Smith', 'jane.smith@example.com', '555-987-6543', '2023-02-20', 'MAN002', 55000.00, NULL, 1, 2),
   (3, 'David', 'Johnson', 'david.johnson@example.com', '555-789-1234', '2023-03-10', 'EMP001', 50000.00, NULL, 1, 3),
   (4, 'Sarah', 'Wilson', 'sarah.wilson@example.com', '555-456-7890', '2023-04-05', 'EMP002', 48000.00, 0.05, 2, 3),
   (5, 'Michael', 'Brown', 'michael.brown@example.com', '555-222-3333', '2023-05-20', 'MAN003', 60000.00, NULL, 1, 4),
   (6, 'Emily', 'Davis', 'emily.davis@example.com', '555-111-2222', '2023-06-15', 'MAN004', 55000.00, NULL, 5, 4),
```

```sql
    (7, 'Daniel', 'Miller', 'daniel.miller@example.com', '555-333-4444', '2023-07-10', 'EMP003', 52000.00, 0.03, 5, 5),
    (8, 'Olivia', 'Jones', 'olivia.jones@example.com', '555-777-8888', '2023-08-25', 'EMP004', 49000.00, 0.02, 6, 5),
    (9, 'William', 'Taylor', 'william.taylor@example.com', '555-555-5555', '2023-09-01', 'MAN005', 60000.00, NULL, 1, 6),
    (10, 'Ava', 'Anderson', 'ava.anderson@example.com', '555-999-0000', '2023-10-10', 'EMP005', 47000.00, 0.04, 9, 6);


select * from departments;
select * from employees;


/* Q. Write a SQL query to find those employees who receive a higher salary than the
employee with ID 163. Return first name, last name. */
--- Ans
select first_name, salary from employees
where salary >
(select salary from employees where Employee_id = 163);


/* Q. write a SQL query to find out which employees have the same designation as the
employee whose ID is 169. Return first name, last name, department ID and job ID. */
--- Ans
select first_name,last_name,department_id,job_id
from employees where job_id =
(select job_id from employees where employee_id = 169);


/* Q. Write a SQL query to find those employees whose salary matches the lowest salary
of any of the departments. Return first name, last name and department ID. */
--- Ans
select first_name , last_name , department_id from employees
where salary in (select min(salary) from employees group by department_id);
```

/* Q. write a SQL query to find those employees who report to that manager whose first name is ‘Payam’. Return first name, last name, employee ID and salary. */

--- Ans

```
select first_name , last_name , employee_id, salary
from employees where manager_id =
(select employee_id from employees where first_name = 'Payam');
```

/* Q. write a SQL query to find those employees whose ID matches any of the numbers 134, 159 and 183. Return all the fields */

--- Ans

```
select * from employees where employee_id in (134,159,183);
```

/* Q. write a SQL query to find those employees whose salary is in the range of 1000, and 3000 (Begin and end values have included.). Return all the fields */

--- Ans

```
select * from employees where salary between 1000 and 3000;
```

/* Q. write a SQL query to find those employees whose salary falls within the range of the smallest salary and 2500. Return all the fields. */

--- Ans

```
select * from employees where salary between (select min(salary) from employees) and 2500;
```

/* Q. write a SQL query to find those employees who do not work in the departments where managers’ IDs are between 100 and 200 (Begin and end values are included.). Return all the fields of the employees. */

--- Ans

```
select * from employees
where not department_id in
(select department_id from employees
where manager_id between 1 and 3);
```

/* Q. write a SQL query to find those employees who work in the same department as ‘Clara’. Exclude all those records where the first name is ‘Clara’. Return first name, last name and hire date. */

```
--- Ans
select first_name , last_name , hire_date
from employees where department_id =
(select department_id from employees where first_name = 'clara')
and first_name <>'clara';
```

/* Q. write a SQL query to find those employees who work in a department where the employee’s first name contains the letter 'T'. Return employee ID, first name and last name. */

```
--- Ans
select employee_id , first_name, last_name from employees
where department_id in
(select department_id from employees where first_name like '%T%');
```

/* Q. write a SQL query to find those employees who earn more than the average salary and work in the same department as an employee whose first name contains the letter 'J'. Return employee ID, first name and salary. */

```
--- Ans
select employee_id, first_name, salary from employees
where salary > (select avg(salary) from employees) and
department_id in
(select department_id from employees where first_name like '%J%');
```

/* Q. write a SQL query to find those employees whose department is located at ‘Toronto’. Return first name, last name, employee ID, job ID. */

```
--- Ans
select first_name, last_name, employee_id, job_id
from employees
```

```sql
where department_id in
(select departement_id from departments where location_id in
(select location_id from locations where city = 'Toronto'));

/* Q. write a SQL query to find those employees whose salaries are higher than the
average for all departments. Return employee ID, first name, last name, job ID. */
--- Ans
select employee_id, first_name, last_name, job_id
from employees
where salary >all
(select avg(salary) from employees group by department_id);

/* Q. . Write a query to display the employee id, name ( first name and last name ) and the
job id column with a modified title SALESMAN for those employees whose job title is
ST_MAN and DEVELOPER for whose job title is IT_PROG. */
--- Ans
select first_name, last_name, employee_id, job_id,
case job_id
when 'ST_MAN' then 'SALESMAN'
when 'IT_PROG' then 'DEVELOPER'
else job_id
end as New_job_id
from employees;

/* Q. Write a query to display the employee id, name ( first name and last name ), salary
and the SalaryStatus column with a title HIGH and LOW respectively for those
employees whose salary is more than and less than the average salary of all
employees. */
--- Ans
select employee_id, first_name, last_name,salary,
case
```

```
when salary > (select avg(salary) from employees) then 'high'
else 'low'
end as SalaryStatus
from employees;
```

/* Q. write a SQL query to find those employees whose salaries exceed 50% of their department's total salary bill. Return first name, last name. */

--- Ans

```
select e1.first_name, e1.last_name, e1.salary  from employees e1
where e1.salary > (select (sum(e2.salary))/2 from employees e2
where e1.department_id = e2.department_id);
```

/* Q. write a SQL query to find those employees who are managers. Return all the fields of the employees table. */

--- Ans

```
select * from employees where employee_id in
(select distinct manager_id from employees);
```

--- Q. Find duplicate values in 1 column

--- Ans

```
select column_name, count(column_name) from table_name
group by column_name having count(column_name)>1;
```

/* Q. Find duplicate values on 2 columns combination */

--- Ans

```
select col_1 , col_2, count(*) as duplicate_num
from table_name
group by col_1,col_2
having duplicate_num>1;
```

/* Q. Write a SQL query to find the most frequent value in a column, along with its

```
frequency. */
--- Ans
create table temp as select col1 , count(col1) as col2 from table_name group by col1;
select col1 from temp where count = (select max(col2) from temp);

/* Q. Find the output of each type of joins on below tables:
Table 1
id1
1
1
Table2
id2
1
1
1 */
--- Ans
--- inner join
select a.id1,b.id2 from Table1 a inner join Table2 b on a.id1 = b.id2;
--- Output
/*
1,1
1,1
1,1
1,1
1,1
1,1
*/
--- left join
select a.id1,b.id2 from Table1 a left join Table2 b on a.id1 = b.id2;
--- Output
/*
```

```
1,1
1,1
1,1
1,1
1,1
1,1
*/
--- right join
select a.id1,b.id2 from Table1 a right join Table2 b on a.id1 = b.id2;
--- Output
/*
1,1
1,1
1,1
1,1
1,1
1,1
*/
--- Full join
select a.id1,b.id2 from Table1 a inner join Table2 b on a.id1 = b.id2
union
select a.id1,b.id2 from Table1 a left join Table2 b on a.id1 = b.id2
union
select a.id1,b.id2 from Table1 a right join Table2 b on a.id1 = b.id2;
--- Output
/*
1,1
1,1
1,1
1,1
1,1
```

```
1,1
*/
------------------------------------------------------------------------
/* Q. Find the output of each type of joins on below tables:
Table 1
id1
1
1
2
Table2
id2
1
1
1
3 */
--- Ans
--- inner join
select a.id1,b.id2 from Table1 a inner join Table2 b on a.id1 = b.id2;
--- Output
/*
1,1
1,1
1,1
1,1
1,1
1,1
*/
--- left join
select a.id1,b.id2 from Table1 a left join Table2 b on a.id1 = b.id2;
--- Output
/*
```

```
1,1
1,1
1,1
1,1
1,1
1,1
2, null
*/
--- right join
select a.id1,b.id2 from Table1 a right join Table2 b on a.id1 = b.id2;
--- Output
/*
1,1
1,1
1,1
1,1
1,1
1,1
null,3
*/
--- Full join
select a.id1,b.id2 from Table1 a inner join Table2 b on a.id1 = b.id2
union
select a.id1,b.id2 from Table1 a left join Table2 b on a.id1 = b.id2
union
select a.id1,b.id2 from Table1 a right join Table2 b on a.id1 = b.id2;
--- Output
/*
1,1
1,1
1,1
```

```
1,1
1,1
1,1
2,null
null,3
*/
------------------------------------------------------------------------------------------
/* Q. Find the output of each type of joins on below tables:
Table 1
id1
1
1
2
2
Table2
id2
1
1
1
3
2
*/
--- Ans
--- inner join
select a.id1,b.id2 from Table1 a inner join Table2 b on a.id1 = b.id2;
--- Output
/*
1,1
1,1
1,1
1,1
```

```
1,1
1,1
2,2
2,2
*/
--- left join
select a.id1,b.id2 from Table1 a left join Table2 b on a.id1 = b.id2;
--- Output
/*
1,1
1,1
1,1
1,1
1,1
1,1
2,2
2,2
*/
--- right join
select a.id1,b.id2 from Table1 a right join Table2 b on a.id1 = b.id2;
--- Output
/*
1,1
1,1
1,1
1,1
1,1
1,1
null,3
2,2
2,2
```

```
*/
--- Full join
select a.id1,b.id2 from Table1 a inner join Table2 b on a.id1 = b.id2
union
select a.id1,b.id2 from Table1 a left join Table2 b on a.id1 = b.id2
union
select a.id1,b.id2 from Table1 a right join Table2 b on a.id1 = b.id2;
--- Output
/*
1,1
1,1
1,1
1,1
1,1
1,1
2,2
2,2
null,3
*/
----------------------------------------------------------------------------------------------------
/* Q. Find the output of each type of joins on below tables( NULL SAFE JOINS_
IMPORTANT):
Table 1
id1
1
2
1
5
NULL
NULL
Table2
```

```
id2
NULL
2
5
5
*/
--- Ans
--- inner join
select a.id1,b.id2 from Table1 a inner join Table2 b on a.id1 = b.id2;
--- Output
/*
2,2
5,5
5,5
*/
--- null safe join
select a.id1,b.id2 from Table1 a inner join Table2 b on a.id1 = b.id2 or
(a.id1 is null and b.id2 is null);
--- Output
/*
2,2
5,5
5,5
null,null
null,null
*/
--- left join
select a.id1,b.id2 from Table1 a left join Table2 b on a.id1 = b.id2 or
(a.id1 is null and b.id2 is null);
--- Output
/*
```

```
1,NULL
2,2
1,NULL
5,5
5,5
NULL,NULL
NULL,NULL
*/
--- right join
select a.id1,b.id2 from Table1 a right join Table2 b on a.id1 = b.id2 or
(a.id1 is null and b.id2 is null);
--- Output
/*
NULL,NULL
2,2
5,5
5,5
*/

/* Question: Write a SQL query to find the names of all employees who earn more than
the average salary in their department using a correlated subquery. */
--- Ans
select a.first_name from employees a where a.salary >
(select avg(b.salary) from employees b where a.department_id = b.department_id);

/* Question: Create a SQL query to find the top 3 products with the highest sales within
each category using a correlated subquery. */
--- Ans
SELECT p1.name, p1.sale_amount, p1.category
FROM products p1
WHERE (
    SELECT COUNT(*)
    FROM products p2
    WHERE p2.category = p1.category AND p2.sale_amount >= p1.sale_amount
) <= 3
ORDER BY p1.category, p1.sale_amount DESC;

/* Question: Write a SQL query to find the total number of orders placed by each
customer, including customers who haven't placed any orders, using a correlated
subquery. */
--- Ans
select distinct a.cust_id, a.cust_name, (select count(b.order_id)
from orders b where a.cust_id = b.cust_id) as order_qnty from orders a;

/* Question: Create a SQL query to find the employees who have salaries greater than
their immediate manager using a correlated subquery. */
--- Ans
select a.first_name, a.employee_id,a.salary from employees a
where a.salary > (select b.salary from employees b
where b.employee_id = a.manager_id);

/* Question: Write a SQL query to find the second-highest salary in each department using
a correlated subquery. */
--- Ans
SELECT p1.first_name, p1.salary
FROM employees p1
WHERE (
    SELECT COUNT(*)
    FROM employees p2
    WHERE p2.department_id = p1.department_id AND p2.salary >= p1.salary
) = 2
ORDER BY p1.department_id, p1.salary DESC;
```