LAB CYCLE 2

Date : 17/03/2025

Experiment No: 5

AIM : Familiarization of Subquery ,Joins ,Views and Set Operations.

Subqueries are nested queries used to retrieve data for the main query, often with aggregate functions. Joins combine data from multiple tables based on common keys, with types like INNER JOIN, LEFT JOIN, and RIGHT JOIN. Views are virtual tables that store pre-defined queries, improving readability and security. Set operations like UNION, INTERSECT, and EXCEPT help combine or filter results from multiple queries. Mastering these concepts enhances SQL efficiency and database management

```
mysql> create table jobs(job_id int primary key auto_increment not null,job_titl
e text not null,min_salary double not null,max_salary double not null);
Query OK, 0 rows affected (0.26 sec)

mysql> create table regions(region_id int primary key auto_increment not null,re
gion_name text not null);
Query OK, 0 rows affected (0.33 sec)
```

```
mysql> create table countries(country_id int not null,country_name text not null
,region_id int not null,primary key(country_id ASC),foreign key(region_id)refere
nces regions(region_id) on delete cascade on update cascade);
Query OK, 0 rows affected (0.41 sec)
```

```
mysql> create table locations(location_id int primary key auto_increment not nul
l,street_address text,postal_code text,city text not null,state_province text,co
untry_id int not null,foreign key(country_id)references countries(country_id) on
 delete cascade on update cascade);
Query OK, 0 rows affected (0.45 sec)
```

```
mysql> create table departments(department_id int primary key auto_increment not
 null,department_name text not null,location_id int not null,foreign key(locatio
n_id)references locations(location_id) on delete cascade on update cascade);
Query OK, 0 rows affected (0.38 sec)
```

```
mysql> create table employees(employee_id int primary key auto_increment not nul
l,first_name text not null,last_name text not null,email text not null,phone_num
ber text not null,hire_date date not null,job_id int not null,salary double not
null,manager_id int,department_id int not null,foreign key(job_id) references jo
bs(job_id) on delete cascade on update cascade,foreign key(department_id)referen
ces departments(department_id) on delete cascade on update cascade,foreign key(m
anager_id)references employees(employee_id) on delete cascade on update cascade)
;
Query OK, 0 rows affected (0.50 sec)
```

```
mysql> create table dependents(dependent_id int primary key auto_increment not n
ull,first_name text not null,last_name text not null,relationship text not null,
employee_id int not null,foreign key(employee_id) references employees(employee_
id) on delete cascade on update cascade);
Query OK, 0 rows affected (0.48 sec)
```

1. Find all employees who locate in the location with the id 1700

>>SELECT employee_id,first_name,location_id FROM employees WHERE location_id = 1700;

employee_id	first_name	location_id
101	John	1700

2. Find all employees who do not locate at the location 1700.

>>SELECT employee_id,first_name,location_id FROM employees WHERE location_id <> 1700;

employee_id	first_name	location_id
102	Jane	1701
103	Robert	1702
104	Emily	1703
105	Michael	1701
106	Sophia	1702
107	David	1701

6 rows in set (0.001 sec)

3.Find the employees who have the highest salary.

>>SELECT employee_id,first_name,salary FROM employees WHERE salary = (SELECT MAX(salary) FROM employees);

```
+-------------+------------+----------+
| employee_id | first_name | salary   |
+-------------+------------+----------+
|         103 | Robert     | 22000.00 |
+-------------+------------+----------+
1 row in set (0.001 sec)
```

4. Finds all employees who salaries are greater than the average salary of all employees.

>>SELECT employee_id,first_name,salary FROM employees
WHERE salary > (SELECT AVG(salary) FROM employees);

```
+-------------+------------+----------+
| employee_id | first_name | salary   |
+-------------+------------+----------+
|         102 | Jane       | 18000.00 |
|         103 | Robert     | 22000.00 |
|         107 | David      | 16000.00 |
+-------------+------------+----------+
3 rows in set (0.001 sec)
```

5. Finds all departments (Department Id, Name) which have at least one employee with the salary is greater than 10,000.

SELECT DISTINCT department_id, department_name
FROM departments WHERE department_id IN (SELECT department_id
FROM employees WHERE salary > 10000);

```
+---------------+-----------------+
| department_id | department_name |
+---------------+-----------------+
|             1 | HR              |
|             2 | Finance         |
|             3 | IT              |
|             5 | Sales           |
+---------------+-----------------+
```

6. Finds all departments (Department Id, Name) that do not have any employee with the salary greater than 10,000.

>>SELECT department_id, department_name

FROM departments

WHERE department_id NOT IN (SELECT DISTINCT department_id FROM employees WHERE salary > 10000);

```
+---------------+-----------------+
| department_id | department_name |
+---------------+-----------------+
|             4 | Marketing       |
+---------------+-----------------+
1 row in set (0.001 sec)
```

7. Find all employees whose salaries are greater than the lowest salary of every department.

>>SELECT employee_id,first_name,salary FROM employees

WHERE salary > ANY (SELECT MIN(salary) FROM employees GROUP BY department_id);

```
+-------------+------------+----------+
| employee_id | first_name | salary   |
+-------------+------------+----------+
|         101 | John       | 14000.00 |
|         102 | Jane       | 18000.00 |
|         103 | Robert     | 22000.00 |
|         104 | Emily      | 10000.00 |
|         105 | Michael    | 12000.00 |
|         107 | David      | 16000.00 |
+-------------+------------+----------+
```

8. Find all employees whose salaries are greater than or equal to the highest salary of every department

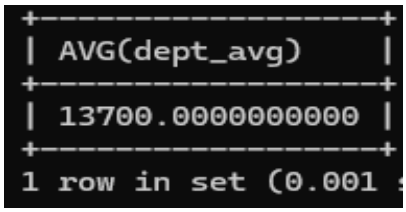
>>SELECT employee_id,first_name,salary FROM employees

WHERE salary >= ALL (SELECT MAX(salary) FROM employees GROUP BY department_id);

```
+-------------+------------+----------+
| employee_id | first_name | salary   |
+-------------+------------+----------+
|         103 | Robert     | 22000.00 |
+-------------+------------+----------+
1 row in set (0.001 sec)
```

9. Calculate the average of average salary of departments. (Hint: SQL subquery in the FROM clause)

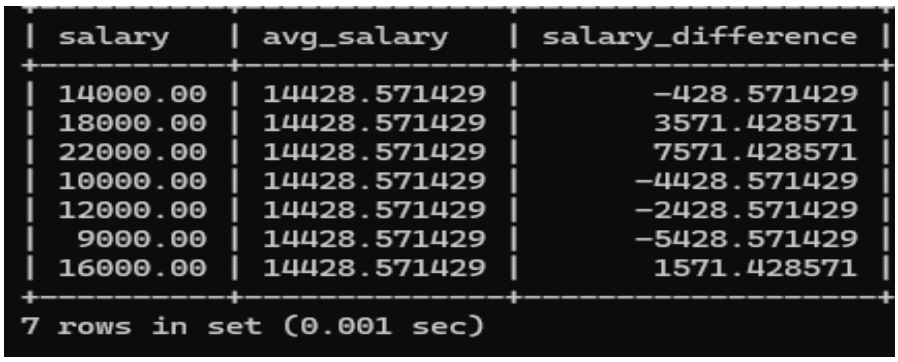
>>SELECT AVG(dept_avg)
FROM (SELECT AVG(salary) AS dept_avg FROM employees GROUP BY department_id) AS temp;



```
+------------------+
| AVG(dept_avg)    |
+------------------+
| 13700.0000000000 |
+------------------+
1 row in set (0.001
```

10. Finds the salaries of all employees, their average salary, and the difference between the salary of each employee and the average salary. (Hint: SQL Subquery in the SELECT clause).

>>SELECT salary, (SELECT AVG(salary) FROM employees) AS avg_salary, salary - (SELECT AVG(salary) FROM employees) AS salary_difference
FROM employees;



```
| salary   | avg_salary   | salary_difference |
+----------+--------------+-------------------+
| 14000.00 | 14428.571429 |       -428.571429 |
| 18000.00 | 14428.571429 |       3571.428571 |
| 22000.00 | 14428.571429 |       7571.428571 |
| 10000.00 | 14428.571429 |      -4428.571429 |
| 12000.00 | 14428.571429 |      -2428.571429 |
|  9000.00 | 14428.571429 |      -5428.571429 |
| 16000.00 | 14428.571429 |       1571.428571 |
+----------+--------------+-------------------+
7 rows in set (0.001 sec)
```

11. Finds all employees whose salary is higher than the average salary of the employees in their departments. (Hint : Use Correlated Subquery).

>>SELECT e1.employee_id,e1.first_name,e1.salary,e1.department_id
FROM employees e1 WHERE salary > (SELECT AVG(salary) FROM employees e2 WHERE e1.department_id = e2.department_id);

employee_id	first_name	salary	department_id
102	Jane	18000.00	2
103	Robert	22000.00	3

2 rows in set (0.001 sec)

12. Find all employees who have no dependents.

>>SELECT employee_id,first_name FROM employees WHERE employee_id NOT IN (SELECT employee_id FROM dependents);

employee_id	first_name
104	Emily
106	Sophia
107	David

3 rows in set (0.001 sec)

13. . Display first name, last name, department name of employees of the Department with id 1, 2 and 3.

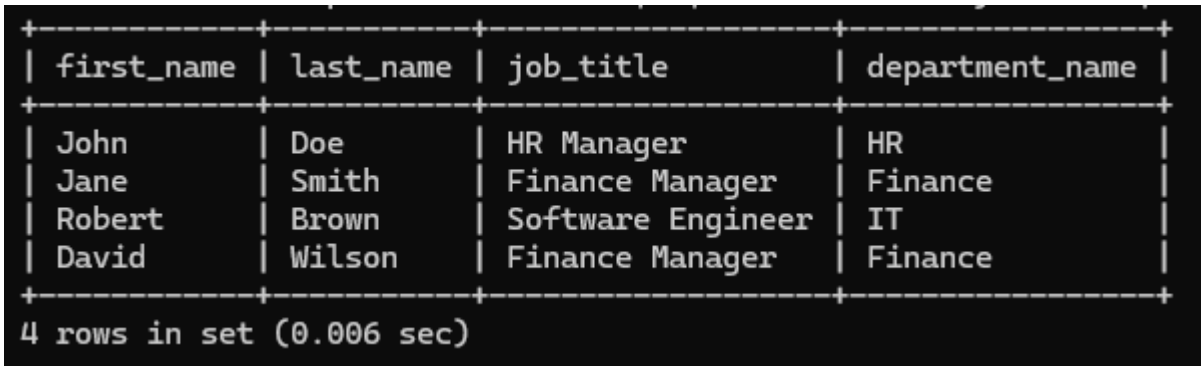
>>SELECT first_name, last_name, department_name FROM employees e JOIN departments d ON e.department_id = d.department_id WHERE e.department_id IN (1, 2, 3);

first_name	last_name	department_name
John	Doe	HR
Jane	Smith	Finance
Robert	Brown	IT
Sophia	Zhang	IT
David	Wilson	Finance

5 rows in set (0.000 sec)

14. . Display the first name, last name, job title, and department name of employees who work in department with id 1, 2, and 3 and salary greater than 10000.

>>SELECT e.first_name, e.last_name, j.job_title, d.department_name FROM employees e JOIN jobs j ON e.job_id = j.job_id JOIN departments d ON e.department_id = d.department_id WHERE e.department_id IN (1, 2, 3) AND e.salary > 10000;

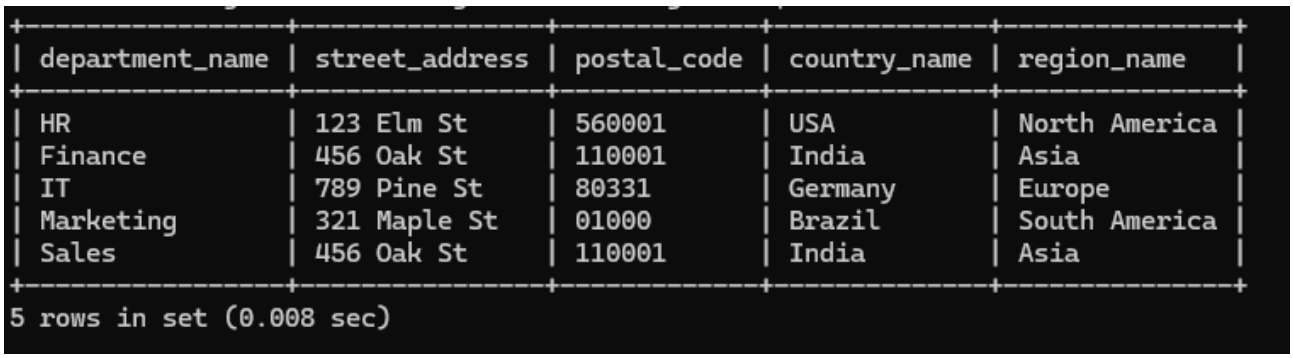


first_name	last_name	job_title	department_name
John	Doe	HR Manager	HR
Jane	Smith	Finance Manager	Finance
Robert	Brown	Software Engineer	IT
David	Wilson	Finance Manager	Finance

4 rows in set (0.006 sec)

15. Display Department name, street address, postal code, country name and region name of all departments.

>>SELECT d.department_name, l.street_address, l.postal_code, c.country_name, r.region_name FROM departments d JOIN locations l ON d.location_id = l.location_id JOIN countries c ON l.country_id = c.country_id JOIN regions r ON c.region_id = r.region_id;

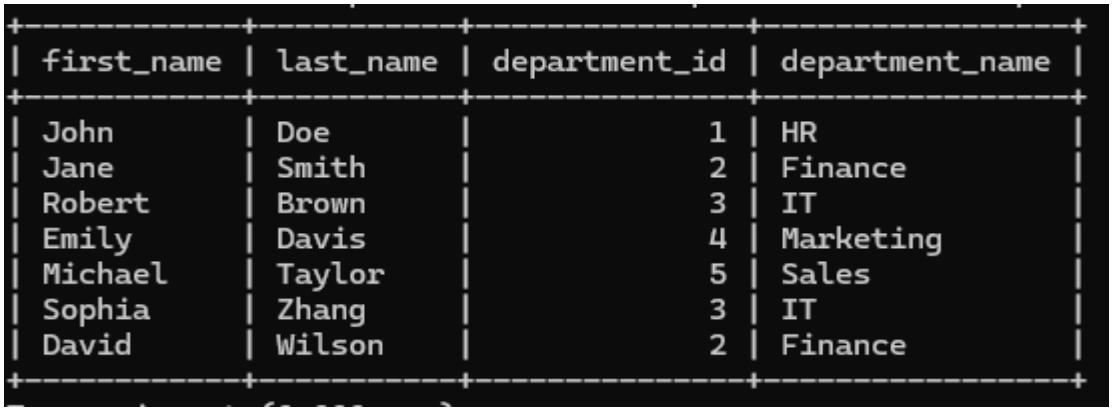


department_name	street_address	postal_code	country_name	region_name
HR	123 Elm St	560001	USA	North America
Finance	456 Oak St	110001	India	Asia
IT	789 Pine St	80331	Germany	Europe
Marketing	321 Maple St	01000	Brazil	South America
Sales	456 Oak St	110001	India	Asia

5 rows in set (0.008 sec)

16. Write a SQL query to find out which employees have or do not have a department. Return first name, last name, department ID, department name.

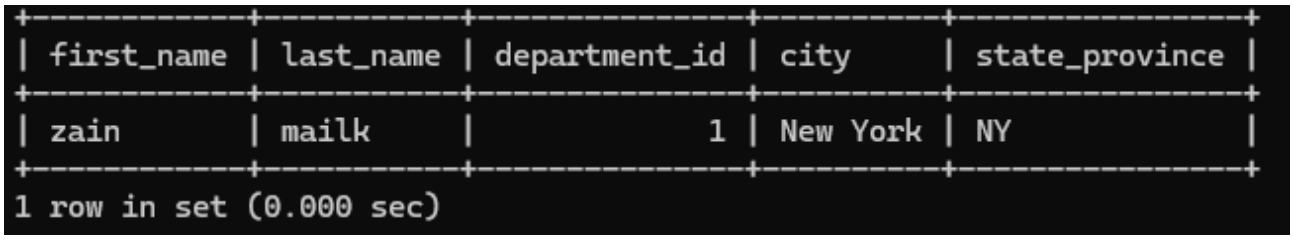
>>SELECT e.first_name, e.last_name, e.department_id, d.department_name FROM employees e LEFT JOIN departments d ON e.department_id = d.department_id;



first_name	last_name	department_id	department_name
John	Doe	1	HR
Jane	Smith	2	Finance
Robert	Brown	3	IT
Emily	Davis	4	Marketing
Michael	Taylor	5	Sales
Sophia	Zhang	3	IT
David	Wilson	2	Finance

17. Write a SQL query to find those employees whose first name contains the letter ‘Z’. Return first name, last name, department, city, and state province.

>>SELECT first_name, last_name, department_id, city, state_province FROM employees JOIN locations ON employees.location_id = locations.location_id WHERE first_name LIKE '%Z%';



first_name	last_name	department_id	city	state_province
zain	mailk	1	New York	NY

1 row in set (0.000 sec)

18**.** Write a SQL query to find all departments, including those without employees Return first name, last name, department ID, department name.

>>SELECT e.first_name, e.last_name, d.department_id, d.department_name FROM departments d LEFT JOIN employees e ON d.department_id = e.department_id;

first_name	last_name	department_id	department_name
John	Doe	1	HR
zain	mailk	1	HR
Jane	Smith	2	Finance
David	Wilson	2	Finance
Robert	Brown	3	IT
Sophia	Zhang	3	IT
Emily	Davis	4	Marketing
Michael	Taylor	5	Sales

19. Write a SQL query to find the employees and their managers. . Those managers do not work under any manager also appear in the list. Return the first name of the employee and manager.

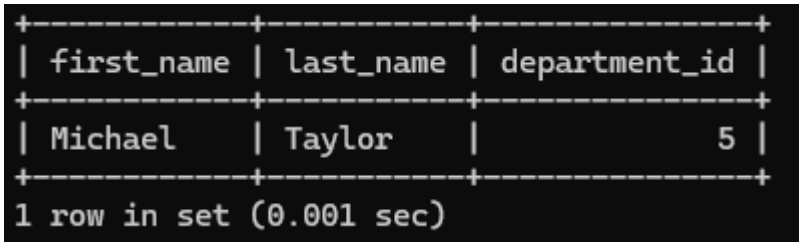
>>SELECT e.first_name AS Employee, m.first_name AS Manager FROM employees e LEFT JOIN employees m ON e.manager_id = m.employee_id;

Employee	Manager
John	NULL
Jane	John
Robert	John
Emily	Jane
Michael	Robert
Sophia	Jane
David	John
zain	NULL

8 rows in set (0.000 sec)

20. Write a SQL query to find the employees who work in the same department as the employee with the last name Taylor. Return first name, last name and department ID.

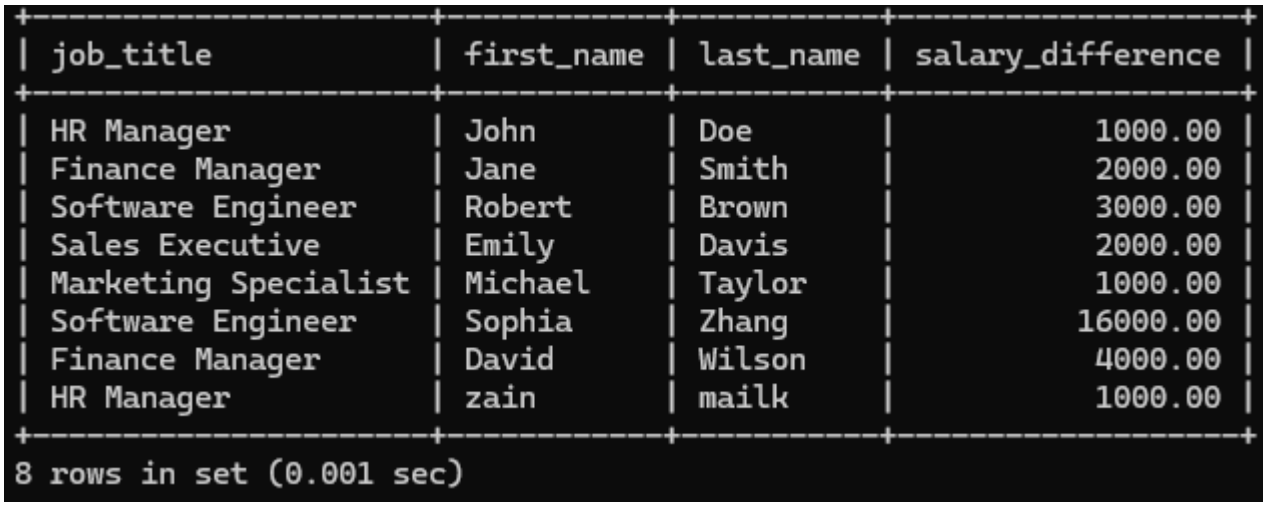
>>SELECT first_name, last_name, department_id
FROM employees WHERE department_id = (SELECT department_id FROM employees WHERE last_name = 'Taylor');



```
+------------+-----------+---------------+
| first_name | last_name | department_id |
+------------+-----------+---------------+
| Michael    | Taylor    |             5 |
+------------+-----------+---------------+
1 row in set (0.001 sec)
```

21. Write a SQL query to calculate the difference between the maximum salary of the job and the employee's salary. Return job title, employee name, and salary difference

>>SELECT j.job_title, e.first_name, e.last_name, (j.max_salary - e.salary) AS salary_difference FROM employees e JOIN jobs j ON e.job_id = j.job_id;



```
+---------------------+------------+-----------+-------------------+
| job_title           | first_name | last_name | salary_difference |
+---------------------+------------+-----------+-------------------+
| HR Manager          | John       | Doe       |           1000.00 |
| Finance Manager     | Jane       | Smith     |           2000.00 |
| Software Engineer   | Robert     | Brown     |           3000.00 |
| Sales Executive     | Emily      | Davis     |           2000.00 |
| Marketing Specialist| Michael    | Taylor    |           1000.00 |
| Software Engineer   | Sophia     | Zhang     |          16000.00 |
| Finance Manager     | David      | Wilson    |           4000.00 |
| HR Manager          | zain       | mailk     |           1000.00 |
+---------------------+------------+-----------+-------------------+
8 rows in set (0.001 sec)
```

22. Write a SQL query to calculate the average salary, the number of employees receiving commissions in that department. Return department name, average salary and number of employees of all departments.

>>SELECT d.department_name, AVG(e.salary) AS avg_salary, COUNT(e.commission_pct) AS num_commissioned FROM employees e JOIN departments d ON e.department_id = d.department_id GROUP BY d.department_name;

department_name	avg_salary	num_commissioned
Finance	17000.000000	0
HR	14000.000000	0
IT	15500.000000	0
Marketing	10000.000000	1
Sales	12000.000000	1

5 rows in set (0.000 sec)

23.Create a view which contains employee name, employee id, phone number, job title, department name, manager name of employees belongs to department whose location is in ‘Delhi’ and display the details.

>>CREATE VIEW delhi_employees AS SELECT e.first_name, e.last_name, e.employee_id, e.phone_number, j.job_title, d.department_name, m.first_name AS manager_name FROM employees e JOIN jobs j ON e.job_id = j.job_id JOIN departments d ON e.department_id = d.department_id JOIN locations l ON d.location_id = l.location_id LEFT JOIN employees m ON e.manager_id = m.employee_id WHERE l.city = 'Delhi';

>>SELECT * FROM delhi_employees;

first_name	last_name	employee_id	phone_number	job_title	department_name	manager_name
Jane	Smith	102	9876543211	Finance Manager	Finance	John
Michael	Taylor	105	9876543214	Marketing Specialist	Sales	Robert
David	Wilson	107	9876543216	Finance Manager	Finance	John

3 rows in set (0.001 sec)

24.Use the above created view to obtain the names of employees whose job title is ‘Manager’ and department is ‘Finance’.

>>SELECT first_name, last_name FROM delhi_employees
WHERE job_title = 'Finance Manager' AND department_name = 'Finance';

first_name	last_name
Jane	Smith
David	Wilson

2 rows in set (0.001 sec)

25.Check whether it is possible to update the phone number of employee whose first name is ‘Smith’ by using the above created view.

>>UPDATE delhi_employees SET phone_number = '1234567890' WHERE first_name = 'Smith';

first_name	phone_number
Smith	1234567890

1 row in set (0.001 sec)

26.Display the details of employee who have no dependents.

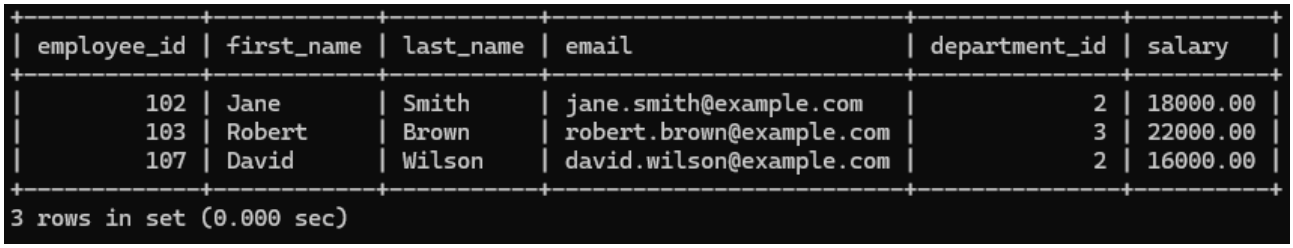
>>SELECT employee_id,first_name,last_name,email,department_id,salary FROM employees WHERE employee_id NOT IN (SELECT employee_id FROM dependents);

employee_id	first_name	last_name	email	department_id	salary
104	Emily	Davis	emily.davis@example.com	4	10000.00
106	Sophia	Zhang	sophia.zhang@example.com	3	9000.00
107	David	Wilson	david.wilson@example.com	2	16000.00
108	zain	mailk	john.doe@example.com	1	14000.00

4 rows in set (0.001 sec)

27.Display the details of employee who manager id is 101 or 201. (Use Union Clause)

>>SELECT employee_id,first_name,last_name,email,department_id,salary FROM employees WHERE manager_id = 101 UNION SELECT employee_id,first_name,last_name,email,department_id,salary FROM employees WHERE manager_id = 201;



employee_id	first_name	last_name	email	department_id	salary
102	Jane	Smith	jane.smith@example.com	2	18000.00
103	Robert	Brown	robert.brown@example.com	3	22000.00
107	David	Wilson	david.wilson@example.com	2	16000.00

3 rows in set (0.000 sec)

28.Display the details of employees who have at least one dependent.

>>SELECT employee_id,first_name,last_name,email,department_id,salary FROM employees WHERE employee_id IN (SELECT employee_id FROM dependents);

employee_id	first_name	last_name	email	department_id	salary
101	John	Doe	john.doe@example.com	1	14000.00
102	Jane	Smith	jane.smith@example.com	2	18000.00
103	Robert	Brown	robert.brown@example.com	3	22000.00
105	Michael	Taylor	michael.taylor@example.com	5	12000.00

4 rows in set (0.001 sec)