# Subquery

A subquery is a SQL query nested inside a larger query.

A subquery may occur in :

- A SELECT clause
- A FROM clause
- A WHERE clause

The subquery can be nested inside a SELECT, INSERT, UPDATE, or DELETE statement or inside another subquery.

A subquery is usually added within the WHERE Clause of another SQL SELECT statement.

You can use the comparison operators, such as >, <, or =. The comparison operator can also be a multiple-row operator, such as IN, ANY, or ALL.

A subquery is also called an inner query or inner select, while the statement containing a subquery is also called an outer query or outer select.

The inner query executes first before its parent query so that the results of an inner query can be passed to the outer query.

Syntax:

```
SELECT column_name
FROM table_name
WHERE column_name expression operator
    ( SELECT COLUMN_NAME  from TABLE_NAME   WHERE ... );
```

1. Write a query to display the employee who earn less than rohan

   Now first you need to know the salary of rohan using the query below

```
SELECT
   salary
FROM
   employee
WHERE
   first_name = 'ROHAN'
```

   Next compare the salary of other employee with rohan salary and display the result as shown below

   **Solution:**

```
SELECT
  *
FROM
  employee
WHERE
  salary <
(SELECT
   salary
FROM
   employee
WHERE
   first_name = 'ROHAN');
```

Output

| emp_id | first_name | last_name | email | hire_date | salary | dept_id |
|---|---|---|---|---|---|---|
| 1 | kelly | davis | davis@gmail.com | 2021-01-22 | 78000 | 80 |
| 4 | andy | lumb | andy@gmail.com | 2021-02-27 | 42200 | 80 |
| 5 | anjel | nair | anj@gmail.com | 2019-09-26 | 42200 | 40 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 6 | ram | kumar | ram@gmail.com | 2018-12-26 | 64200 | 40 |

2. Write a query to display the employee details of sales department
   Here first we should know the dept_id of sales that can be fetched using the query below

```
SELECT
   dept_id
FROM
   department
WHERE
   dept_name = 'sales'
```

Now display all the employees details whose dept_id is belong to sales using the query below

```
SELECT * FROM
  employee
WHERE
  dept_id =
(SELECT dept_id FROM department WHERE dept_name =
'sales');
```

Output:

| emp_id | first_name | last_name | email | hire_date | salary | dept_id |
|---|---|---|---|---|---|---|
| 7 | rohan | sharma | ro@gmail.com | 2021-02-09 | 84200 | 20 |
| 8 | john | king | | 2021-02-09 | 124200 | 20 |

3. Query the employee details who work in the department in which john works

   First need to fetch the dept_id of John using the query below

```
SELECT
  dept_id
FROM
  employee
WHERE
  first_name = 'JOHN'
```

Now once you got john department id, the employees whose department id matches the john the department id need to be retrieved using the query below

```
SELECT
  *
FROM
  employee
WHERE
  dept_id =
(SELECT dept_id FROM sql_notes.employee WHERE first_name
= 'JOHN');
```

Output:

| emp_id | first_name | last_name | email | hire_date | salary | dept_id |
|---|---|---|---|---|---|---|
| 7 | rohan | sharma | ro@gmail.com | 2021-02-09 | 84200 | 20 |
| 8 | john | king | | 2021-02-09 | 124200 | 20 |

4. Query the details of employees working in the department where department names begins with ‘s’

   Here first you need to fetch department name starting with s using the query below

```
SELECT
  dept_id
FROM
  department
WHERE
  dept_name
LIKE 's%'
```

Now once you get dept_id of the department starting with s you can fetch employee detail working in that department using the below query

```
SELECT
  *
FROM
  employee
WHERE
  dept_id =
(SELECT dept_id FROM department WHERE dept_name LIKE
's%');
```

Output:

| | | | | | |
|---|---|---|---|---|---|
| **0** | **9** | **13:06:04** | **SELECT * FROM sql_notes.employee WHERE dept_id = (SELECT dept_id FROM sql_notes.department WHERE dept_name LIKE 's%') LIMIT 0, 1000** | **Error Code: 1242. Subquery returns more than 1 row** | **0.000 sec** |

Here you have got the error it is because in the inner query you are fetching the department id whose department name starts with s there might be more than one department whose name starts with s, but in the parent query you are trying to match with one department id using equal to here you need to make use of IN operator which helps you to match multiple department.

```
SELECT
  *
FROM
  employee
WHERE
  dept_id
IN
(SELECT dept_id FROM sql_notes.department WHERE dept_name
LIKE 's%');
```

Output:

| dept_id | first_name | last_name | email | hire_date | salary | dept_id |
|---|---|---|---|---|---|---|
| 7 | rohan | sharma | ro@gmail.com | 2021-02-09 | 84200 | 20 |
| 8 | john | king | null | 2021-02-09 | 124200 | 20 |
| 5 | anjel | nair | anj@gmail.com | 2019-09-26 | 42200 | 40 |
| 6 | ram | kumar | ram@gmail.com | 2018-12-26 | 64200 | 40 |

5. Query the details of employees who earn more than kelly and belong to john's department

   Here you need get the dept_id of the employee john using the query below

```
SELECT dept_id FROM employee WHERE first_name = 'john'
```

   Next need to get salary for kelly using the query below

```
SELECT salary FROM sql_notes.employee WHERE first_name =
'Kelly'
```

   Review the detail of all employee who earn more than kelly and dept_id is same as john by executing the following query

```
SELECT * FROM
   employee
WHERE
  dept_id =
(SELECT dept_id FROM sql_notes.employee WHERE first_name
= 'john')
and
  salary >
(SELECT salary FROM sql_notes.employee WHERE first_name =
'Kelly');
```

Output:

| dept-id | first_name | last_name | email | hire_date | salary | dept_id |
|---|---|---|---|---|---|---|
| 7 | rohan | sharma | ro@gmail.com | 2021-02-09 | 84200 | 20 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 8 | john | king | | 2021-02-09 | 124200 | 20 |

6. Query the details of employees who belong to either sales or IT department earning salaries greater than 50000

   First retrieve the dept_id working in dept_name sales or IT using the below query

```
SELECT dept_id FROM sql_notes.department WHERE dept_name
= 'sales' OR dept_name = 'IT'
```

   Next retrieve the employee whose salary > 50000 and dept_id is sales or IT using the query below

```
SELECT * FROM
  employee
WHERE
  dept_id
IN
(SELECT dept_id FROM department WHERE dept_name = 'sales'
OR dept_name = 'IT')
and
  salary > 50000;
```

**Output:**

| emp_id | first_name | last_name | email | hire_date | salary | dept_id |
|---|---|---|---|---|---|---|
| 7 | rohan | sharma | ro@gmail.com | 2021-02-09 | 84200 | 20 |
| 8 | john | king | | 2021-02-09 | 124200 | 20 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | kelly | davis | davis@gmail.com | 2021-01-22 | 78000 | 80 |

7. Query the first_name of managers of all department earning less than john

   First retrieve the manager id of the employees working as manager using the below query

```
SELECT mrg_id FROM department
```

Next retrieve the salary of john using the query below

```
SELECT salary FROM employee WHERE first_name = 'john'
```

Now retrieve the first name of employee Where the employee is working as a manager and salary is less than the john salary using the below query

```
SELECT
  first_name
FROM
  employee
WHERE
  emp_id
IN
(SELECT mrg_id FROM department)
AND
   salary <
(SELECT salary FROM sql_notes.employee WHERE first_name =
'john') ;
```

**Output:**

| **first_name** |
|---|
| kelly |
| tom |
| mike |
| andy |

| ram |
|---|

8. Predict the output?

```
SELECT DEPT_ID, MIN(SALARY) FROM EMPLOYEE GROUP BY
DEPT_ID HAVING MIN(SALARY) < (SELECT MAX(SALARY) FROM
EMPLOYEE WHERE DEPT_ID = 20) ORDER BY DEPT_ID;
```

Here the first maximum salary of an employee working in dept_id 20 is fetched. Next dep_id and minimum salary is fetched in each department where minimum salary is less than maximum salary of employees working in department id 20. It will be printed in ascending order of department id.

**Output:**

| **dept_id** | **MIN(salary)** |
|---|---|
| 20 | 84200 |
| 40 | 42200 |
| 50 | 98200 |
| 70 | 84200 |
| 80 | 42200 |

9. Write a query to check if there are employees working in executive department with dept_id 30

   **EXISTS:**
   The EXISTS operator is used to test for the existence of any record in a subquery. The EXISTS operator returns TRUE if the subquery returns one or more records.

**Syntax:**

SELECT column_name(s)
FROM table_name
WHERE EXISTS
(SELECT column_name FROM table_name WHERE condition);

```
SELECT * FROM
  employee
WHERE
EXISTS
(SELECT * FROM employee WHERE dept_id = 30);
```

**Output:**

| emp_id | first_name | last_name | email | hire_date | salary | dept_id |
|---|---|---|---|---|---|---|
| 1 | kelly | davis | davis@gmail.com | 2021-01-22 | 78000 | 80 |
| 2 | tom | taylor | tom@gmail.com | 2020-09-22 | 84200 | 30 |
| 3 | mike | whalen | mike@gmail.com | 2021-06-30 | 98200 | 50 |
| 4 | andy | lumb | andy@gmail.com | 2021-02-27 | 42200 | 80 |
| 5 | anjel | nair | anj@gmail.com | 2019-09-26 | 42200 | 40 |
| 6 | ram | kumar | ram@gmail.com | 2018-12-26 | 64200 | 40 |
| 7 | rohan | sharma | ro@gmail.com | 2021-02-09 | 84200 | 20 |
| 8 | john | king | null | 2021-02-09 | 124200 | 20 |

10. Write a query to display the employee whose salary is greater than the minimum salary of any of the department

- First get minimum salary present in each department using the below query

```
SELECT MIN(salary) FROM employee GROUP BY dept_id
```

- Next retrieve the details of the employee whose salary is greater than minimum salary present in each department using the query below

```
SELECT first_name, dept_id, salary
FROM
employee
WHERE salary >
ANY(SELECT MIN(salary) FROM employee GROUP BY dept_id);
```

11. Create a copy of the table using subquery?

```
CREATE TABLE EMP (SELECT * FROM employee);
```

To verify weather the new table is created or not, let's execute the below query

```
SELECT * FROM emp;
```

Output:

| dept_id | first_name | last_name | email | hire_date | salary | dept_id |
|---|---|---|---|---|---|---|
| 1 | kelly | davis | davis@gmail.com | 2021-01-22 | 78000 | 80 |
| 2 | tom | taylor | tom@gmail.com | 2020-09-22 | 84200 | 30 |
| 3 | mike | whalen | mike@gmail.com | 2021-06-30 | 98200 | 50 |
| 4 | andy | lumb | andy@gmail.com | 2021-02-27 | 42200 | 80 |
| 5 | anjel | nair | anj@gmail.com | 2019-09-26 | 42200 | 40 |
| 6 | ram | kumar | ram@gmail.com | 2018-12-26 | 64200 | 40 |
| 7 | rohan | sharma | ro@gmail.com | 2021-02-09 | 84200 | 20 |
| 8 | john | king | | 2021-02-09 | 124200 | 20 |

If you observe from the above output the data, constraint everything is copied. If you want only table with those column to be created then you can achieve using LIKE Operator as shown below

```
CREATE TABLE sql_notes.emp3 LIKE employee;
```

Now table is there if you want to copy the data present in employee table to emp3 in that case you can execute the query below

```
INSERT INTO sql_notes.emp3(SELECT * FROM employee);
```

To verify weather all data is inserted or not execute the below query

```
SELECT * FROM emp3;
```

Output:

| dept_id | first_name | last_name | email | hire_date | salary | dept_id |
|---|---|---|---|---|---|---|
| 1 | kelly | davis | davis@gmail.com | 2021-01-22 | 78000 | 80 |
| 2 | tom | taylor | tom@gmail.com | 2020-09-22 | 84200 | 30 |
| 3 | mike | whalen | mike@gmail.com | 2021-06-30 | 98200 | 50 |
| 4 | andy | lumb | andy@gmail.com | 2021-02-27 | 42200 | 80 |
| 5 | anjel | nair | anj@gmail.com | 2019-09-26 | 42200 | 40 |
| 6 | ram | kumar | ram@gmail.com | 2018-12-26 | 64200 | 40 |
| 7 | rohan | sharma | ro@gmail.com | 2021-02-09 | 84200 | 20 |
| 8 | john | king | | 2021-02-09 | 124200 | 20 |

12.Create a copy of the table where the department id is 40

```
CREATE TABLE
emp2(SELECT * FROM employee WHERE dept_id = 40);
```

To verify whether the copy of table is created or not execute the query below

```
SELECT * FROM emp2;
```

**Output:**

| emp_id | first_name | last_name | email | hire_date | salary | dept_id |
|---|---|---|---|---|---|---|
| 5 | anjel | nair | anj@gmail.com | 2019-09-26 | 42200 | 40 |
| 6 | ram | kumar | ram@gmail.com | 2018-12-26 | 64200 | 40 |

13. Update the salary of employees in sales department by incrementing to 5000

```
UPDATE
  employee
SET salary = salary + 5000
WHERE
  dept_id =
(SELECT dept_id FROM sql_notes.department WHERE dept_name
= 'SALES');
```

To verify weather the salary of sales department is updated or not you can execute the below query

```
SELECT * FROM employee;
```

**Output:**

| dept_id | first_name | last_name | email | hire_date | salary | dept_id |
|---|---|---|---|---|---|---|
| 1 | kelly | davis | davis@gmail.com | 2021-01-22 | 78000 | 80 |
| 2 | tom | taylor | tom@gmail.com | 2020-09-22 | 84200 | 30 |
| 3 | mike | whalen | mike@gmail.com | 2021-06-30 | 98200 | 50 |
| 4 | andy | lumb | andy@gmail.com | 2021-02-27 | 42200 | 80 |
| 5 | anjel | nair | anj@gmail.com | 2019-09-26 | 42200 | 40 |
| 6 | ram | kumar | ram@gmail.com | 2018-12-26 | 64200 | 40 |
| 7 | rohan | sharma | ro@gmail.com | 2021-02-09 | 89200 | 20 |
| 8 | john | king | | 2021-02-09 | 129200 | 20 |

The department id of sales is 20, as you can see from the above output rohan and john salary is increased

14. Write a query to delete the record of sales manager
    - First fetch the manager id with department name sales using the query below

```
SELECT
  mrg_id
FROM
  department
WHERE
  dept_name = 'sales'
```

    - Once you fetched the manager id then you delete the record with the emp_id = mrg_id using the query below

```
DELETE
FROM
  emp3
WHERE
  emp_id =
(SELECT mrg_id FROM department WHERE dept_name = 'sales');
```

To verify whether the employees got deleted execute the query below

```
SELECT * FROM emp3;
```

Output:

| dept_id | first_name | last_name | email | hire_date | salary | dept_id |
|---|---|---|---|---|---|---|
| 1 | kelly | davis | davis@gmail.com | 2021-01-22 | 78000 | 80 |
| 2 | tom | taylor | tom@gmail.com | 2020-09-22 | 84200 | 30 |
| 3 | mike | whalen | mike@gmail.com | 2021-06-30 | 98200 | 50 |
| 4 | andy | lumb | andy@gmail.com | 2021-02-27 | 42200 | 80 |
| 5 | anjel | nair | anj@gmail.com | 2019-09-26 | 42200 | 40 |
| 6 | ram | kumar | ram@gmail.com | 2018-12-26 | 64200 | 40 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 7 | rohan | sharma | ro@gmail.com | 2021-02-09 | 89200 | 20 |

Here john with emp_id was working as manager in sales department that record got deleted

15.Write a query to delete the email id of IT Manager

```
Update
  employee
SET
  email = NULL
WHERE
  emp_id =
(SELECT mrg_id FROM department WHERE dept_name = 'IT');
```

To verify weather the mail id of IT manager is updated with null execute the below query

```
SELECT * FROM employee;
```

**Output:**

| dept_id | first_name | last_name | email | hire_date | salary | dept_id |
|---|---|---|---|---|---|---|
| 1 | kelly | davis | null | 2021-01-22 | 78000 | 80 |
| 2 | tom | taylor | tom@gmail.com | 2020-09-22 | 84200 | 30 |
| 3 | mike | whalen | mike@gmail.com | 2021-06-30 | 98200 | 50 |
| 4 | andy | lumb | andy@gmail.com | 2021-02-27 | 42200 | 80 |
| 5 | anjel | nair | anj@gmail.com | 2019-09-26 | 42200 | 40 |
| 6 | ram | kumar | ram@gmail.com | 2018-12-26 | 64200 | 40 |
| 7 | rohan | sharma | ro@gmail.com | 2021-02-09 | 89200 | 20 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 8 | john | king | | 2021-02-09 | 129200 | 20 |