Subquery

- Query within another query is called Subquery/ nested query.
- Subquery is used to retrieve data from single or multiple tables by using more than one step.

Types of Sub Query

1. Non correlated Subquery
    - In non correlated subqueries child query is executed first then the parent query is executed.
2. Correlated Sub query
    - In correlated subqueries parent query is executed first then the child query is executed.

Non correlated Subquery

- Child Query
    - A query which provides value to another query is called child query/ inner query.
- Parent Query
    - A query which receives values from another query is called parent query / outer query.

Example#1

Write a query to display the employees who are getting more salary than the average salary from employee table.

ename	salary
Ram	30000.00
Raj	35000.00
Rahul	40000.00
Alok	60000.00
Sunil	35000.00
Tushar	50000.00

Requirement

ename	salary
Alok	60000.00
Tushar	50000.00

```
CREATE TABLE employee(
        ename varchar(20),
   salary decimal(8,2)
);

INSERT INTO employee VALUES
('Ram', 30000), ('Raj', 35000), ('Rahul', 40000),
('Alok', 60000), ('Sunil', 35000), ('Tushar', 50000);

SELECT * FROM employee;
SELECT avg(salary) FROM employee;
SELECT * FROM employee WHERE salary > (SELECT avg(salary) FROM employee);
```

Example#2

Write a query to display the employees who are working in IT department from employee, department tables by using sub query.

department_id	department_name
1	IT
2	HT
3	Finance

employee_id	employee_name	department_id
101	Raj	1
102	Alok	2
103	Sunil	1
104	Ram	3
105	Tushar	2

Requirements

employee_id	employee_name	department_id
101	Raj	1
103	Sunil	1

```
CREATE TABLE departments (
   department_id INT,
   department_name VARCHAR(50)
);
CREATE TABLE employees (
   employee_id INT,
   employee_name VARCHAR(100),
```

```
    department_id INT
);

INSERT INTO departments (department_id, department_name)
VALUES
    (1, 'IT'),
    (2, 'HT'),
    (3, 'Finance');

INSERT INTO employees (employee_id, employee_name, department_id)
VALUES
    (101, 'Raj', 1),
    (102, 'Alok', 2),
    (103, 'Sunil', 1),
    (104, 'Ram', 3),
    (105, 'Tushar', 2);
```

```
SELECT * FROM employees WHERE department_id =

(SELECT department_id FROM departments WHERE department_name='IT');
```

Example#3

Write a query to display the employees who are getting more salary than the height paid employee in '1' department.

employee_id	employee_name	salary	department_id
101	Raj	30000.00	1
102	Alok	40000.00	2
103	Sunil	35000.00	1
104	Ram	45000.00	3
105	Tushar	30000.00	2

Requirements

employee_id	employee_name	salary	department_id
102	Alok	40000.00	2
104	Ram	45000.00	3

```
CREATE TABLE departments (
   department_id INT,
   department_name VARCHAR(50)
);
CREATE TABLE employees (
   employee_id INT,
   employee_name VARCHAR(100),
   salary decimal(8,2),
   department_id INT
);

INSERT INTO departments (department_id, department_name)
VALUES
   (1, 'IT'),
   (2, 'HT'),
   (3, 'Finance');

INSERT INTO employees VALUES
   (101, 'Raj', 30000, 1),
   (102, 'Alok', 40000, 2),
   (103, 'Sunil', 35000, 1),
   (104, 'Ram', 45000, 3),
   (105, 'Tushar', 30000, 2);


SELECT max(salary) FROM employees WHERE department_id=1;

SELECT * FROM employees

WHERE salary > (SELECT max(salary) FROM employees WHERE department_id=1);
```

Example#4

Write a query to display the employees who are getting more salary than the height paid employee in 'IT' department.

```
SELECT * FROM employees WHERE salary >
(SELECT max(salary) FROM employees WHERE department_id=
(SELECT department_id FROM departments WHERE department_name='IT'));
```

Example#5

Write a query to display the employees who are getting more salary than ‘Sunil’ and works in the department where ‘Tushar’ is works.

```
SELECT * FROM employees
WHERE salary > (SELECT salary FROM employees WHERE employee_name='Sunil')
AND department_id = (SELECT department_id FROM employees WHERE
employee_name='Tushar');
```

Example#6

Write a query to display the second highest salary from employee table.

```
SELECT max(salary) FROM employees
WHERE salary < (SELECT max(salary) FROM employees);
```