SQL Sub-Queries

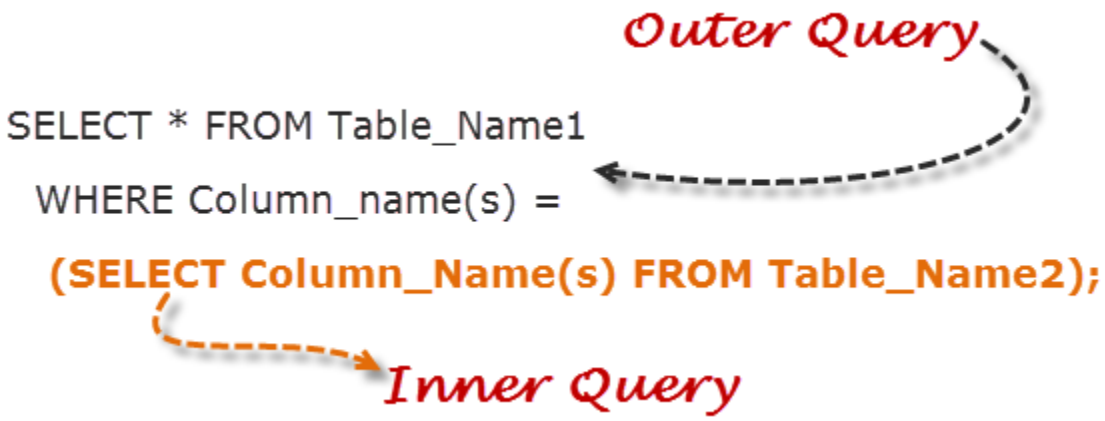
What are Sub queries?

SQL Sub queries are the queries which are embedded inside another query. The embedded queries are called as **INNER** query & container query is called as **OUTER** query.

The subqueries are the queries which are executed inside of another query. The result SQL query is totally depends on the result of sub query. First the INNER query gets executed & the result of INNER query is passed as input to the outer query.

SQL Sub-Query Syntax:

Let's look at the basic syntax of the SQL Sub query command:



Three types of sub queries are supported in SQL are – Scalar, Row and Table sub queries.

- The **Scalar subquery** result returns only a single row and single column.
- The **Row subquery** result returns only a single row with single/multiple column(s).
- The **Table subquery** result returns can be return single/multiple row(s) or column(s).

In the Sub query you may use the different operators to filter out the result like [=, >, =, <=, !=,]. These Sub queries can be used conjunction with INSERT, UPDATE and DELETE queries.

Suppose you want to find the name of the department in which employee_id = 100 is currently working on.

Let's see how this sub query is constructed & executed inside of another query:

```
1  SELECT department_name  FROM department
```

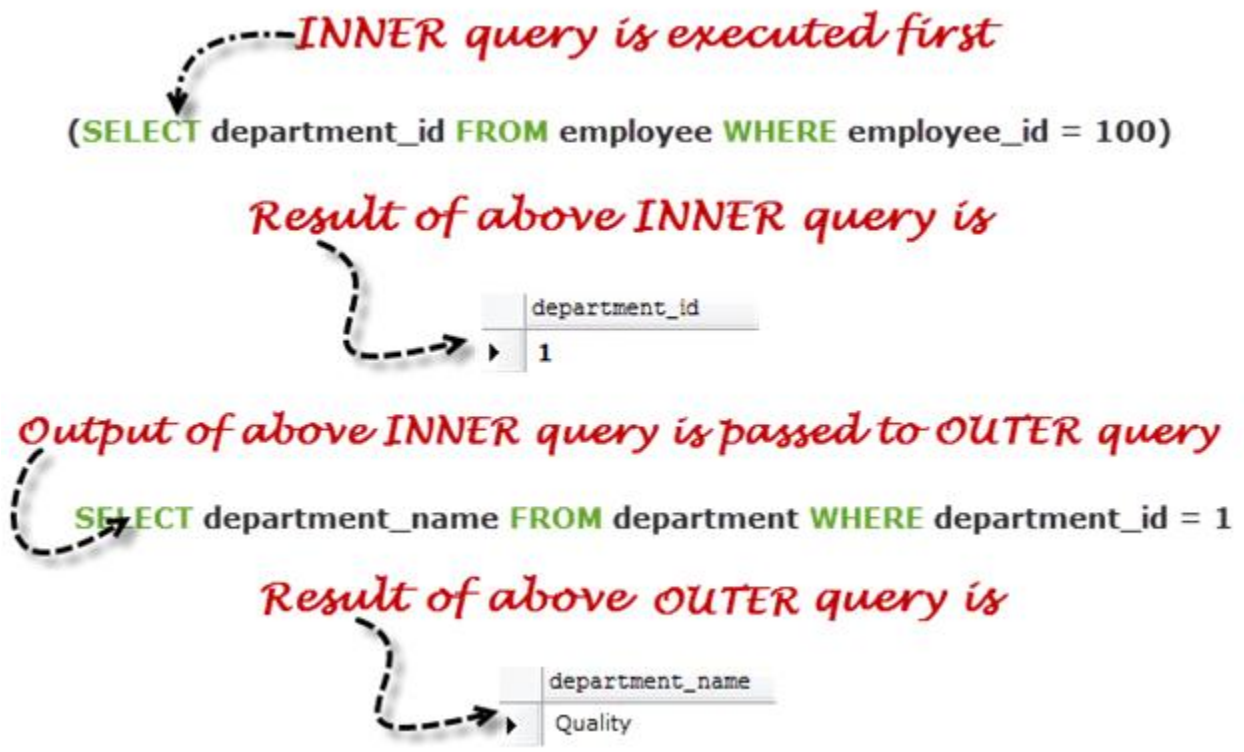
```
2
```

```
3  WHERE department_id =

4

5  (SELECT department_id FROM employee WHEREemployee_id = 100);
```

Following is the result upon executing the above SQL Sub query:

department_name
Quality

So let's discuss how the result of above query is calculated:



In above **Row Sub-Queries,** the result of INNER query can is returned only one value.

Let's take a look at the other Sub query type who returns can be return single/multiple row(s) or column(s) i.e. **Table sub-quer**y:

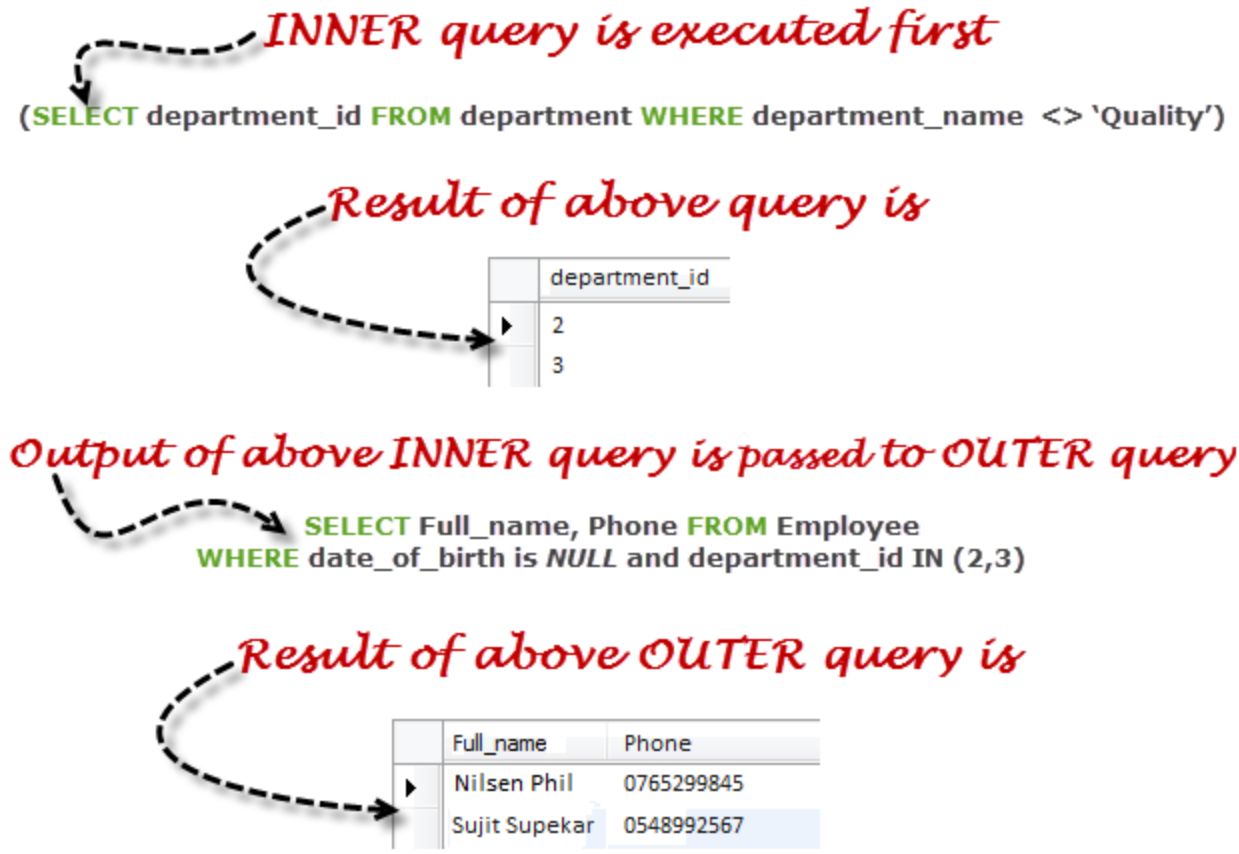
Suppose you want get list of employee's Name and Phone number who's working in other than Quality department & date of birth is not registered in Employee tracking system.

```
1  SELECT Full_name,Phone  FROM Employee
2
3  WHERE date_of_birth  is NULL anddepartment_id  IN
4
5  (SELECT department_id  FROM departmentWHERE department_name   <>  'Quality')
```

Following is the result upon executing the above SQL Sub query:

Full_name	Phone
Nilsen Phil	0765299845
Sujit Supekar	0548992567

So let's discuss how the result of above query is calculated:



You can use multiple INNER queries inside INNER queries, the SQL supports INNER queries up to 32 levels.

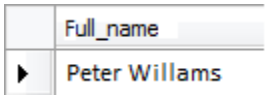
In above examples we have seen INNER queries up to two levels; here we are seeing three level INNER query:

In the company higher managements wants to announce the awards to highest paying employee member, so here is the query to get the name of the highest paying employee:

```
1  Select Full_name From employee WHEREEmployee_id =

2

3  (SELECT Employee_id FROM payments WHEREsalary =

4
```

```
5  (SELECT MAX(salary) FROM payments))
```

Following is the result upon executing the above SQL triple Sub query:



	Full_name
▶	Peter Willams

Sub-Queries Vs Joins!

The Subqueries are simpler to write & easy to understand. As a result, Sub queries are more frequently used in the beginner's level. The Joins are complicated but more powerful than Sub queries.

Majorly sub queries run independently and result of the sub query used in the outer query (other than correlated sub query) and in case of JOIN's, a query only give the result when the joining condition gets satisfied.

In JOIN both the tables should have a common column name but in sub query without having a column name we can execute the query.

If we think in terms of the performance prospective, then the Joins are faster than the Sub queries. Using Joins, it approximately boosts the performance of query by 500 times as compare to Sub queries. So Joins are more popular than the Sub queries & most of the SQL experts are preferred to use Joins instead of *SubQueries*.

Conclusion on SQL Sub-Queries:

- Sub queries contain two parts, one is INNER query & other is OUTER query. The result of INNER query is passed to OUTER query as input.
- Sub queries are simple & easy to understand. It can be easily broken down into logical steps, so it offers more flexibility.
- The Sub queries are used in conjunction with SELECT, INSERT, UPDATE & DELETE commands.
- In this article we have learnt about three types of SQL supb queries: scalar, row and table sub queries.
- In SQL server, The **Nested query** can be used up to 32 levels.
- As compare with Joins, the performance of Sub query is low. Joins are 500 times faster than Sub queries.

For performance issues, when it comes to getting data from multiple tables, it is strongly recommended to use JOINs instead of sub queries. Sub queries should only be used with good reason. So in the next article I am covering basics of Joins & what all types of Joins offered in the SQL server.

Source:

http://www.softwaretestingclass.com/sql-sub-queries/