Correlated Subquery

By: Virendra

Correlated Subquery

- It is a subquery **that uses values from the from the outer subquery** (i.e. the inner query is totally dependent on the outer query).
- Correlated subqueries uses **large amount of memory** and can be **slow to execute** as, the subquery may be evaluated once for each row processed by the outer query.
- A table alias must be used to specify which table reference is to be used.
- They are also known as synchronized subquery.

Example 1

- Query : Consider following Employees schema:

  Employees (Emp_id, first_name, last_name, salary, dept_id)

- Find the salaries of persons who get lesser than the avg salaries of the persons in their own department.

Example 1

- For this we will have to first find the avg. salary for each department. Which can be done as follows:

- SELECT dept_id, AVG(salary) as average FROM Employees GROUP BY dept_id

Dept_id	average
10	7000
20	4400
30	9500
…	…

Example 1

- Now we need to find the employees whose salary is less than the average salary of their department. Which can be done as follows:

- SELECT first_name, last_name, salary, dept_id
  FROM employees **oq** WHERE salary <
  (SELECT AVG(salary) FROM employees **iq** WHERE iq.dept_id = oq.dept_id GROUP BY dept_id)

- For every outer query data, the inner query executes and checks if the value is lesser or not. If the value is lesser it prints — else it doesn't. It skips.

Example 2

- Using EXISTS display the employee_id, manager_id, first_name and last_name of those employees who manage other employees.
- Employees (Emp_id, First_name, last_name, manager_id, salary, dept_id)

Example 2

- SELECT employee_id, manager_id, first_name, last_name FROM employees a WHERE EXISTS

- (SELECT employee_id FROM employees b WHERE b.manager_id = a.employee_id)

Example 2

- SELECT employee_id, manager_id, first_name, last_name FROM employees a WHERE EXISTS (SELECT employee_id FROM employees b WHERE b.manager_id = a.employee_id)