Sub-queries

Dr. Sambit Bakshi

NIT Rourkela

March 31, 2020

Outline

- Subquery is a query nested within another query such as SELECT, INSERT, UPDATE or DELETE
- A MySQL subquery is called an inner query while the query that contains the subquery is called an outer query.

Example

SELECT attribute1, attribute2 **FROM** tableName1 *WHERE* attribute3 **IN** (**Select** att1 **FROM** tableName2 **WHERE** someCondition);

Comparison Operators

Suppose a relational table named Payments has the following attributes as follows.

- CustomerNumber.
- CheckNumber
- PaymentDate
- Amount

Comparison Operators

- The comparison operators can be used in a single value returned by the subquery with the expression in the WHERE clause.
- If it has been asked to find out the customer-details who is having maximum payment.
- Query :

SELECT customerNumber, checkNumber, amount **FROM** payments **WHERE** amount = (**SELECT** MAX(amount) **FROM** payments)

- When you use a subquery in the FROM clause, the result set returned from a subquery is used as a temporary table.
- This table is referred to as a derived table or materialized subquery.

Example:

SELECT MAX(items), MIN(items),FLOOR(AVG(items))
FROM(SELECT orderNumber, COUNT(orderNumber)
AS items **FROM** orderdetails
GROUP BY orderNumber) **AS** lineitems;

- In the previous examples, you notice that a subquery is independent. It means that you can execute the subquery as a standalone query.
- i.e. the inner query can be executed separately.
- A correlated subquery depends on the output of the outer subquery.
- A correlated subquery is evaluated once for each row in the outer query.

Example:

SELECT productname, buyprice
FROM products p1
WHERE buyprice $>$ (**SELECT** AVG(buyprice) **FROM** products **WHERE** productline $=$ p1.productline);

This query selects products whose buy prices are greater than the average buy price of all products in each product line.

- The inner query executes for every product line because the product line is changed for every row, hence the average buy price will also change.
- The outer query filters only products whose buy price is greater than the average buy price per product line from the subquery.

- When a subquery is used with the **EXISTS** or **NOT EXISTS** operator, a subquery returns a Boolean value of **TRUE** or **FALSE**.
- The **EXISTS** and **NOT EXISTS** are often used in the correlated subqueries.

- You can use the query above as a correlated subquery to find customers who placed at least one sales order with the total value greater than 60K by using the **EXISTS** operator.

- You can use the query above as a correlated subquery to find customers who placed at least one sales order with the total value greater than 60K by using the **EXISTS** operator.

SELECT customerNumber,customerName **FROM** customers **WHERE EXISTS**(**SELECT** orderNumber, SUM(priceEach * quantityOrdered)

SELECT orderNumber,SUM(priceEach * quantityOrdered),total
FROM orderdetails *INNER JOIN* orders USING (orderNumber)
GROUP BY orderNumber **HAVING** SUM(priceEach * quantityOrdered) $>$ 60000;
N.B. Query finds sales orders whose total values are greate