Introduction to Subqueries

Simple Subqueries

SELECT \* FROM Customers

WHERE CustomerID IN (SELECT CustomerID FROM Orders)

Correlated Subqueries

SELECT \* FROM Customers c

WHERE (SELECT COUNT(\*) FROM Orders o WHERE o.CustomerID = c.CustomerID) > 2

Using subqueries in SELECT statements

Example 1: Subquery in SELECT clause

SELECT customer\_id, (SELECT MAX(order\_date)

FROM orders

WHERE orders.customer\_id = o.customer\_id) AS latest\_order\_date

FROM orders o

GROUP BY customer\_id;

SELECT o.order\_id, o.order\_date, c.customer\_name

FROM orders o

JOIN (SELECT customer\_id, customer\_name FROM customers) AS c

ON o.customer\_id = c.customer\_id;

Using subqueries in WHERE statements

SELECT order\_id, customer\_id, order\_date, order\_amount

FROM orders

WHERE order\_amount > (SELECT AVG(order\_amount) FROM orders);

Case scenarios

SELECT

employees.

employee\_name

FROM employees

WHERE hire\_date >= DATE\_SUB(CURDATE(), INTERVAL 5 YEAR)

AND employee\_id IN (SELECT employee\_id FROM performance\_reviews WHERE rating >= 4);

hire\_date >= DATE\_SUB(CURDATE(), INTERVAL 5 YEAR)

AND employee\_id IN (SELECT employee\_id FROM performance\_reviews WHERE rating >= 4)

SELECT products.product\_id, products.name, SUM(orders.quantity) AS total\_sold

FROM products

JOIN (

SELECT product\_id, SUM(quantity) AS quantity

FROM orders

WHERE order\_date >= DATE\_SUB(CURDATE(), INTERVAL 1 YEAR)

GROUP BY product\_id

) AS orders

ON products.product\_id = orders.product\_id

GROUP BY products.product\_id

ORDER BY total\_sold DESC

LIMIT 10;

SELECT product\_id, SUM(quantity) AS quantity

FROM orders

WHERE order\_date >= DATE\_SUB(CURDATE(), INTERVAL 1 YEAR)

GROUP BY product\_id

SELECT products.product\_id, products.name, SUM(orders.quantity) AS total\_sold

FROM products

JOIN (

... Subquery ...

) AS orders

ON products.product\_id = orders.product\_id

GROUP BY products.product\_id

ORDER BY total\_sold DESC

LIMIT 10;

SELECT name

FROM departments

WHERE (

SELECT AVG(salary)

FROM employees

WHERE employees.department\_id = departments.id

) = (

SELECT MAX(AVG(salary))

FROM employees

GROUP BY department\_id

);

Case scenario

SELECT

products.name,

(price - cost) as profit\_margin,

categories.name as category

FROM products

JOIN categories

ON products.category\_id = categories.id

WHERE (price - cost) = (

SELECT MAX((price - cost))

FROM products

WHERE products.category\_id = categories.id

);

Correlated Subqueries

SELECT CustomerID, SUM(TotalAmount) AS TotalSpent

FROM Orders o1

WHERE TotalAmount = (SELECT MAX(TotalAmount)

FROM Orders o2

WHERE o1.CustomerID = o2.CustomerID)

GROUP BY CustomerID;

Case scenario

SELECT

Employees.EmployeeID,

Employees.Name,

(SELECT AVG(SaleAmount)

FROM Sales

WHERE Sales.EmployeeID = Employees.EmployeeID

AND Sales.SaleDate BETWEEN Employees.HireDate AND DATE\_ADD(Employees.HireDate, INTERVAL 1 YEAR)) AS FirstYearAvgSale

FROM Employees;

Case Scenario

INSERT INTO top\_sales (representative\_id, sale\_amount, sale\_date)

SELECT representative\_id, sale\_amount, sale\_date

FROM sales

WHERE representative\_id IN (SELECT representative\_id

FROM sales

GROUP BY representative\_id

ORDER BY SUM(sale\_amount) DESC

LIMIT 10);

**UPDATE**

UPDATE employees

SET salary = salary \* 1.1

WHERE hire\_date <= (CURRENT\_DATE - INTERVAL 5 YEAR);

**DELETE**:

DELETE FROM products

WHERE product\_id IN (

SELECT product\_id

FROM products

WHERE discontinued = 'Yes' AND stock\_level = 0

);

INSERT INTO drivers (driver\_id, total\_distance, total\_revenue)

SELECT driver\_id, SUM(distance), SUM(fare)

FROM rides

GROUP BY driver\_id;

UPDATE drivers SET total\_distance = (

SELECT SUM(distance)

FROM rides

WHERE drivers.driver\_id = rides.driver\_id

GROUP BY driver\_id

), total\_revenue = (

SELECT SUM(fare)

FROM rides

WHERE drivers.driver\_id = rides.driver\_id

GROUP BY driver\_id

);

DELETE FROM drivers

WHERE driver\_id NOT IN (

SELECT driver\_id

FROM rides

GROUP BY driver\_id

);

Common table expressions (CTEs)

WITH TripInformation AS (

SELECT

drivers.name AS driver\_name,

trips.start\_time AS trip\_start\_time,

trips.end\_time AS trip\_end\_time,

trips.fare AS trip\_fare

FROM

drivers

JOIN trips ON drivers.id = trips.driver\_id

)

SELECT

driver\_name,

SUM(trip\_fare) AS total\_fare

FROM

TripInformation

GROUP BY

driver\_name

ORDER BY

total\_fare DESC;

Case Scenario

WITH MonthlyOccupancy AS (

SELECT

date\_trunc('month', check\_in\_date) AS month,

count(\*) AS rooms\_occupied

FROM

reservations

WHERE

check\_in\_date >= date '2021-01-01'

AND check\_in\_date < date '2022-01-01'

GROUP BY

month

)

SELECT

to\_char(month, 'Mon YYYY') AS month,

rooms\_occupied

FROM

MonthlyOccupancy

ORDER BY

month;

WITH AvgDeliveryTime AS (

SELECT

DATEPART(HOUR, order\_time) AS order\_hour,

AVG(delivery\_time) AS avg\_delivery\_time

FROM DeliveryOrders

GROUP BY DATEPART(HOUR, order\_time)

)

SELECT

CAST(order\_hour AS VARCHAR(2)) + ' PM' AS order\_hour,

avg\_delivery\_time

FROM AvgDeliveryTime;