Types of Subqueries

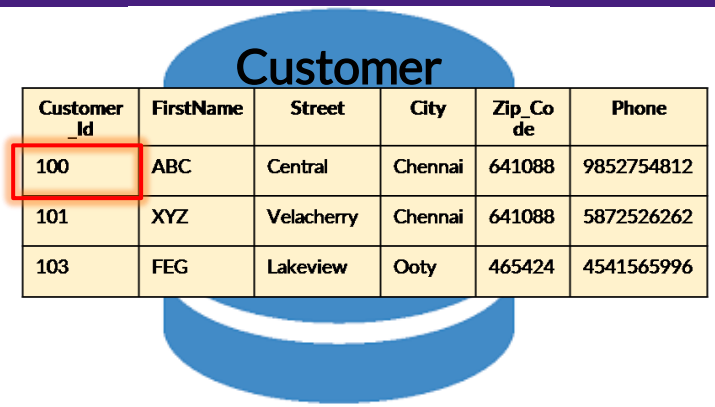
In this session, you will learn:

- Types of Subqueries with example queries

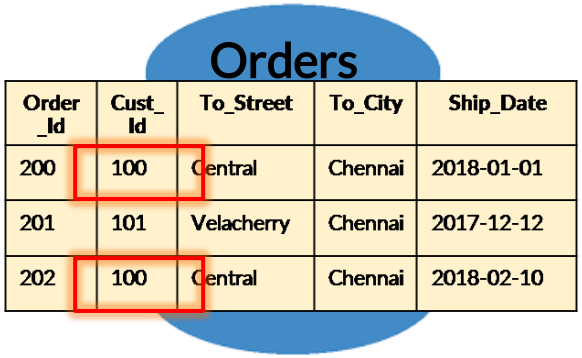
Types of Subqueries

- Single Row Subqueries
- Multiple Row Subqueries
- Multiple Column Subqueries
- Correlated Subqueries

Single Row Subqueries in WHERE Clause



Customer_Id	FirstName	Street	City	Zip_Code	Phone
100	ABC	Central	Chennai	641088	9852754812
101	XYZ	Velacherry	Chennai	641088	5872526262
103	FEG	Lakeview	Ooty	465424	4541565996



Order_Id	Cust_Id	To_Street	To_City	Ship_Date
200	100	Central	Chennai	2018-01-01
201	101	Velacherry	Chennai	2017-12-12
202	100	Central	Chennai	2018-02-10

Query to display the order details of all the orders placed by the customer with name 'ABC'.

```
Select * from Orders
where Cust_Id =   (Select Customer_Id from Customer
        100        where FirstName = 'ABC');
```

Order_Id	Cust_Id	To_Street	To_City	Ship_Date
200	100	Central	Chennai	2018-01-01
202	100	Central	Chennai	2018-02-10

Multiple Row Subqueries in WHERE Clause

Customer

Customer_Id	FirstName	Street	City	Zip_Code	Phone
100	ABC	Central	Chennai	641088	9852754812
101	XYZ	Velacherry	Chennai	641088	5872526262
103	FEG	Lakeview	Ooty	465424	4541565596
104	ABC	Lakeview	Ooty	465424	8482927482

Orders

Order_Id	Cust_Id	To_Street	To_City	Ship_Date
200	100	Central	Chennai	2018-01-01
201	101	Velacherry	Chennai	2017-12-12
202	100	Central	Chennai	2018-02-10
203	104	Lakeview	Ooty	2018-02-02

Query to display the order details of all the orders placed by the customer with name 'ABC'.

```
Select * from Orders
where Cust_Id IN(Select Customer_Id from Customer
                 where FirstName = 'ABC');
```

= (100, 104)

use the IN, ANY, or ALL operator in outer query to handle a subquery that returns multiple rows.

Order_Id	Cust_Id	To_Street	To_City	Ship_Date
200	100	Central	Chennai	2018-01-01
202	100	Central	Chennai	2018-02-10
203	104	Lakeview	Ooty	2018-02-02

Multiple Column Subqueries in WHERE Clause

Product_Orders

Order_Id	Pdt_Id	Quantity	Discount
200	300	1	0
200	301	5	0.1
201	300	1	0.2
202	301	4	0
204	301	5	0.3

Query to display the order_id, pdt_id, and quantity of any products in which the pdt_id and quantity match both the pdt_id and quantity of an product in order_id 200.

```
Select Order_Id,Pdt_Id,Quantity from Product_Orders
where (Pdt_Id,Quantity) IN  (Select Pdt_Id,Quantity from
Product_Orders where Order_Id = 200)
```

((300, 1), (301,5))

Order_Id	Pdt_Id	Quantity
200	300	1
200	301	5
201	300	1
204	301	5

A correlated subquery is a subquery that contains a reference to a table that also appears in the outer query. For example:

```
SELECT * FROM t1   WHERE column1 = ANY (SELECT column1 FROM t2
                    WHERE t2.column2 = t1.column2);
```

Notice that the subquery contains a reference to a column of t1, even though the subquery's FROM clause does not mention a table t1. So, MySQL looks outside the subquery, and finds t1 in the outer query.

Scoping rule: MySQL evaluates from inside to outside.

To avoid materializing several times for a given derived table, we can instead materialize—once—a derived table which adds a grouping on the join column.

Correlated Subqueries

Product

Pdt_Id	Pdt_Name	Price	Pdt_Type
300	Fan	5000	Electronics
301	Rhymes	1000	Books
302	Shirt	2000	Men Apparel

Product_Orders

Order_Id	Pdt_Id	Quantity	Discount
200	300	1	0
200	301	5	0.1
201	300	1	0.2
202	301	4	0
204	301	5	0.3

Query to display the order_id, pdt_id and quantity of any products in which the pdt_type is 'Electronics'.

```
Select Order_Id,po.Pdt_Id,Quantity from Product_Orders po
Where  po.Pdt_Id =  (Select p.Pdt_Id from Product where
                     p.Pdt_Type = 'Electronics')
```

300

Order_Id	Pdt_Id	Quantity
200	300	1
201	300	1

THANKS