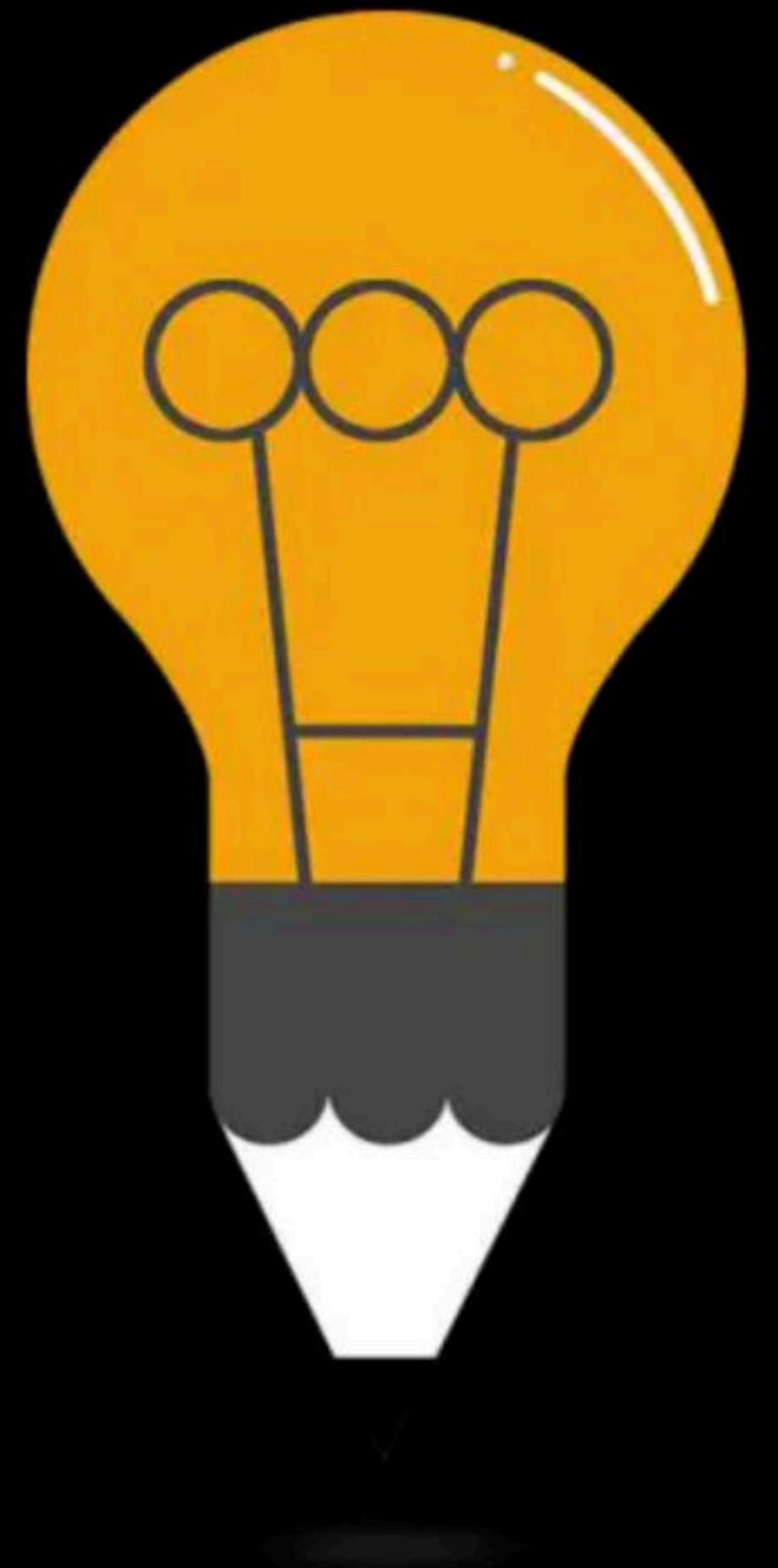






# Finding Keys in Relational DB & Normalization: Part I

Complete Course on Database Management System



# DBMS SQL

By: Vishvadeep Gothi

# Subquery

A Subquery or Inner query or a Nested query is a query within another SQL query and embedded within the WHERE clause.

# Subquery

1. Single row subquery
2. Multiple row subquery
3. correlated

# Subquery

Find all such products which are having price less than price of product name 'Tofu'

# Subquery

Find the product with second highest price

# Subquery

Find the product with third highest price

# Subquery

1. In
2. Any
3. All
4. Exist

where value  $\underset{\text{In}}{=}$  (5, 9, 15)

# Suppliers Table

SupplierID	SupplierName	ContactName	Address	City	PostalCode	Country	Phone
1	Exotic Liquid	Charlotte Cooper	49 Gilbert St.	London	EC1 4SD	UK	(171) 555-2222
2	New Orleans Cajun Delights	Shelley Burke	P.O. Box 78934	New Orleans	70117	USA	(100) 555-4822
3	Grandma Kelly's Homestead	Regina Murphy	707 Oxford Rd.	Ann Arbor	48104	USA	(313) 555-5735
4	Tokyo Traders	Yoshi Nagase	9-8 Sekimai Musashino-shi	Tokyo	100	Japan	(03) 3555-5011
5	Cooperativa de Quesos 'Las Cabras'	Antonio del Valle Saavedra	Calle del Rosal 4	Oviedo	33007	Spain	(98) 598 76 54
6	Mayumi's	Mayumi Ohno	92 Setsuko Chuo-ku	Osaka	545	Japan	(06) 431-7877
7	Pavlova, Ltd.	Ian Devling	74 Rose St. Moonie Ponds	Melbourne	3058	Australia	(03) 444-2343
8	Specialty Biscuits, Ltd.	Peter Wilson	29 King's Way	Manchester	M14 GSD	UK	(161) 555-4448
9	PB Knäckebröd AB	Lars Peterson	Kaloadagatan 13	Göteborg	S-345 67	Sweden	031-987 65 43
10	Refrescos Americanas LTDA	Carlos Diaz	Av. das Americanas 12.890	São Paulo	5442	Brazil	(11) 555 4640
11	Heli Süßwaren GmbH & Co. KG	Petra Winkler	Tiergartenstraße 5	Berlin	10785	Germany	(010) 9984510
12	Plutzer Lebensmittelgrossmärkte AG	Martin Bein	Bogenallee 51	Frankfurt	60439	Germany	(069) 992755
13	Nord-Ost-Fisch Handelsgesellschaft mbH	Sven Petersen	Frahmredder 112a	Cuxhaven	27478	Germany	(04721) 8713
14	Formaggi Fortini s.r.l.	Elio Rossi	Viale Dante, 75	Ravenna	48100	Italy	(0544) 60323

# Customers Table

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avia. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden
6	Blauer See Delikatessen	Hanna Moos	Forsterstr. 57	Mannheim	68306	Germany
7	Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000	France
8	Bólido Comidas preparadas	Martin Sommer	C/ Araquí, 67	Madrid	28023	Spain
9	Bon app'	Laurence Lebihans	12, rue des Bouchers	Marseille	13008	France
10	Bottom-Dollar Marketse	Elizabeth Lincoln	23 Tsawassen Blvd.	Tsawassen	T2F 8M4	Canada
11	B's Beverages	Victoria Ashworth	Fauntleroy Circus	London	EC2 5NT	UK
12	Cactus Comidas para llevar	Patricia Simpson	Cerrito 333	Buenos Aires	1010	Argentina
13	Centro comercial Moctezuma	Francisco Chang	Sierras de Granada 9993	México D.F.	05022	Mexico
14	Chop-suey Chinese	Yang Wang	Hauptstr. 29	Bern	3012	Switzerland
15	Comércio Mineiro	Pedro Alfonso	Av. dos Lusiadas, 23	São Paulo	05432-043	Brazil
16	Consolidated Holdings	Elizabeth Brown	Berkeley Gardens 12 Brewery	London	WX1 6LT	UK

# Subquery

Find all customers that are from the same countries as the suppliers

select \* from customers

where country in

Select country from suppliers



# Subquery

Find all customers, that are from those countries where there is not any suppliers

Select \* from customers where country not in  
(select country from Suppliers)

# Orders Table

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10248	90	5	1996-07-04	3
10249	81	6	1996-07-05	1
10250	34	4	1996-07-06	2
10251	84	3	1996-07-08	1
10252	76	4	1996-07-09	2
10253	34	3	1996-07-10	2
10254	14	5	1996-07-11	2
10255	68	9	1996-07-12	3
10256	88	3	1996-07-15	2
10257	35	4	1996-07-16	3
10258	20	1	1996-07-17	1
10259	13	4	1996-07-18	3
10260	55	4	1996-07-19	1
10261	61	4	1996-07-19	2
10262	65	8	1996-07-22	3
10263	20	9	1996-07-23	3
10264	24	6	1996-07-24	3
10265	7	2	1996-07-25	1
10266	97	3	1996-07-26	3

# Subquery

Find all customers, who have placed more than 2 orders

```
select * from customers where  
customerID In  
(select customerID from orders  
group by (customerID) having count(orderID) > 2)
```

# Why subquery not Join



no. of tuples after join  
will be

$$\frac{r_1}{r_1 \text{ records}} \quad \frac{r_2}{r_2 \text{ records}}$$

$r_1 * r_2$  comparison  
filter  
↓

# Subquery

In operator can only has “Equal to” comparison

## Any with Subquery

```
SELECT column_name(s)  
FROM table_name  
WHERE column_name operator ANY  
(SELECT column_name  
FROM table_name  
WHERE condition);
```

↳ if condition is true for one value also then  
true.

> any (5, 9, 15, 6, 2)

The operator must be a standard comparison operator (=, <>, !=, >, >=, <, or <=)

# OrderDetails Table

OrderDetailID	OrderID	ProductID	Quantity
1	10248	11	12
2	10248	42	10
3	10248	72	5
4	10249	14	9
5	10249	51	40
6	10250	41	10
7	10250	51	35
8	10250	65	15
9	10251	22	6
10	10251	57	15
11	10251	65	20
12	10252	20	40
13	10252	33	25
14	10252	60	40
15	10253	31	20
16	10253	39	42
17	10253	49	40
18	10254	24	15
19	10254	55	21

# Any with Subquery

Find the ProductName of all those products which have their orders Quantity larger than 50

select ProductName from Products where  
ProductID = Any

$x = 5, 6, 9$

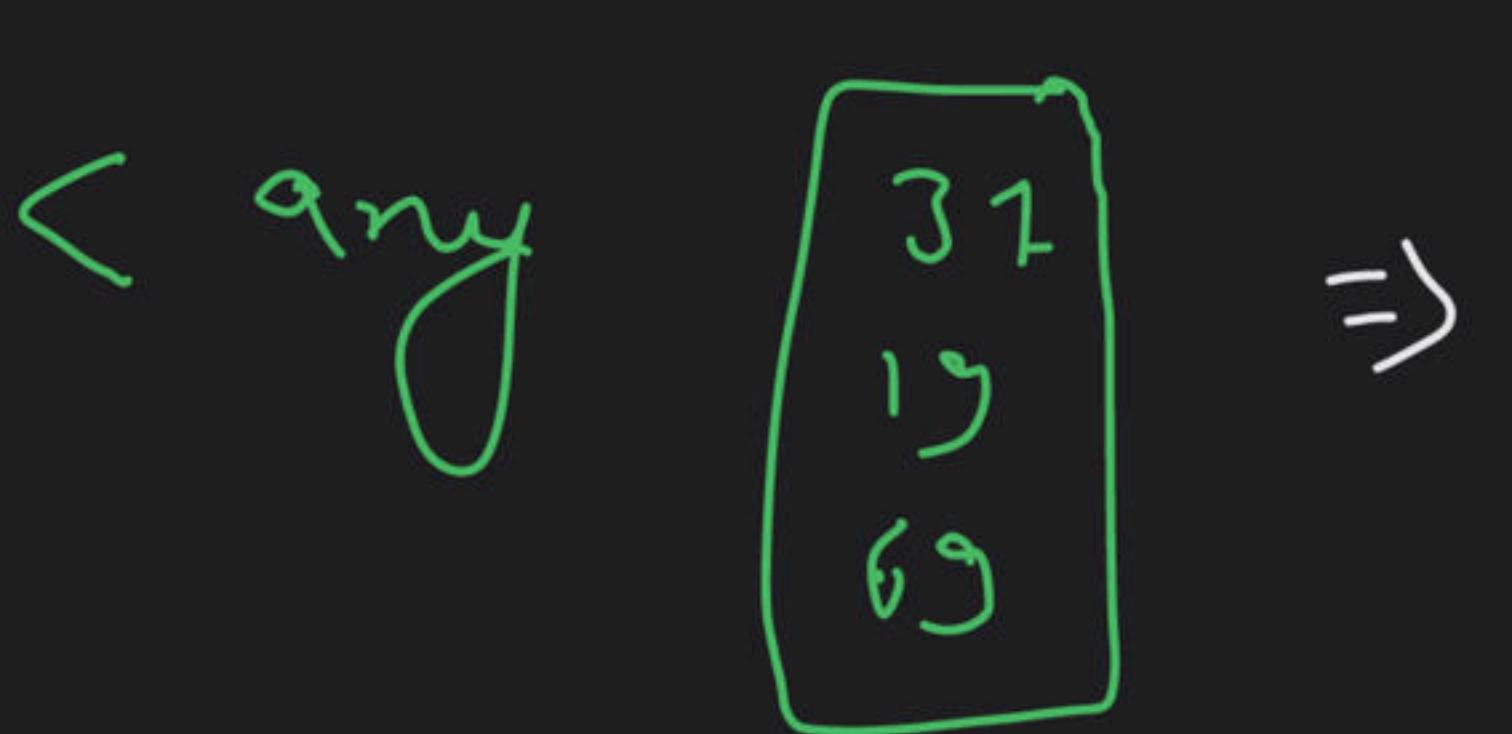
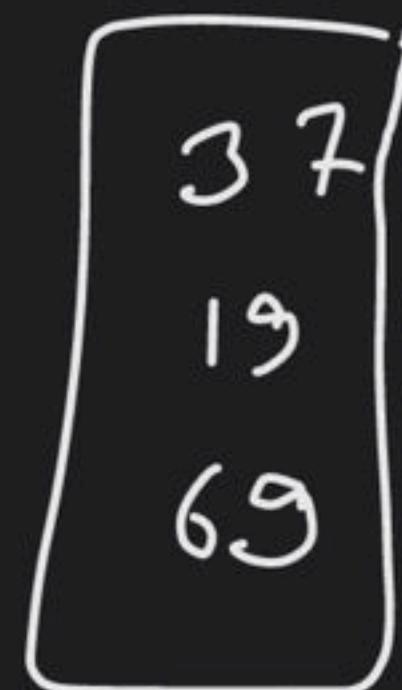
Same as In  
(select ProductID from OrderDetails  
where quantity > 50)

# Any with Subquery

Find the ProductName of all those products which have their productIDs less than any of the product having orders Quantity equal to 1

```
select ProductName from products where  
productID < any  
( select productID from orderdetails where  
Quantity = 1 )
```

Inner query result  $\Rightarrow$



1  
2  
3  
4  
5  
6  
7  
8

select productname from products where  
productid < ( select max (productid) from orderdetails  
where quantity = 1 )

# ALL with Subquery

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator ALL
(SELECT column_name
FROM table_name
WHERE condition);
```

The operator must be a standard comparison operator (=, <>, !=, >, >=, <, or <=)

# ALL with Subquery

~~Find the ProductName of all those products which have their orders Quantity larger than 50~~

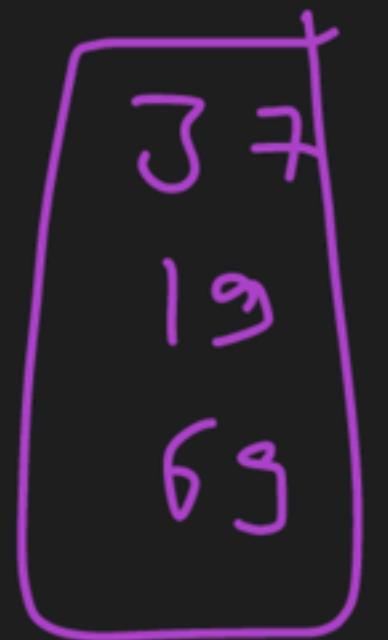
# ALL with Subquery

Find the ProductName of all those products which have their productIDs less than all of the products having orders Quantity equal to 1

```
select ProductName from products  
where ProductID < all
```

```
(select ProductID from orderdetails where  
Quantity = 1)
```

Subarray



< all



<(min)

# Subquery

Consider the following table:

Employee (EmployeeID, FirstName, LastName, JobID, Salary, DepartmentID)

Finds all employees whose salaries are greater than the salary of all the employees in the Sales department with DepartmentID is 2:

select \* from employee where salary > all

(select salary from employee where DepartmentID=2)

# Subquery

Operator	Meaning
= ALL	$=v_1 \text{ and } =v_2 \text{ and } =v_3 \dots$
> ALL	$> (\max)$
< ALL	$< (\min)$
$\geq$ ALL	$\geq (\max)$
$\leq$ ALL	$\leq (\min)$

$\neq$  ALL  $\neq v_1 \text{ and } \neq v_2 \text{ and } \neq v_3 \dots \leftarrow$  not in

# Subquery

Operator	Meaning
= ANY	= v <sub>1</sub> or = v <sub>2</sub> or ..... ←
< ANY	< (max)
> ANY	> (min)
>= ANY	>= (min)
<= ANY	<= (max)

same as IN

# Exists with Subquery

Used to test for the existence of any record in a subquery

Returns TRUE if the subquery returns one or more records

# Exists with Subquery

```
SELECT column_name(s)  
FROM table_name  
WHERE EXISTS  
(SELECT column_name FROM table_name WHERE condition)
```

# Exists with Subquery

Select \* from Customers where CustomerID=1

# Exists with Subquery

Select \* from customers where exists

(Select \* from Customers  
where CustomerID=1)

all details of customers with id 1

↓

1 tuple

# Exists with Subquery

Select \* from Orders where exists

(Select \* from Customers  
where CustomerID>89)

# Exists with Subquery

Select \* from Orders where exists

(Select NULL)



it selects NULL as one value



result set

NULL

---

NULL

}  $\Rightarrow$  all records of orders table

Co-related subquery

for each row of outer query, entire inner query runs again & again.

General subquery

```
fun()  
{  
    SubQuery()  
    =  
}  
}
```

Co related

```
outer query loop  
{  
    inner query loop  
    }  
}
```

unacademy student

Rno	name	dob
1	Amit	27 Oct
2	Priya	21 Jan
3	Neha	3 Mar
4	Sunit	4 April

Enrolled	
Rno	Courseid
1	C3
2	C2
2	C1
4	C4

Rno	name
1	Amit
2	Priya
4	Sunit

select Rno, name from student S where exist  
 (select \* from Enrolled E where  
 S.Rno = E.Rno)

# Exists with Subquery

Write a query to select all such customers record which have atleast one order placed

select \* from customers where exists

(select \* from orders where orders.customerId = customers.

Using joins:-  
distinct

select ^customers.\* from customers, orders

where customers.customerId = orders.customerId

# Set Operators

Set operators are used to combine results from two or more SELECT statements

1. Union
2. Union all
3. Intersect
4. Minus or Except

# Relational Model

The relational model uses a collection of tables to represent both data and the relationships among those data

# Relation

The main construct for representing data in the relational model is a relation, which is table.

# Attribute

Attributes are used to describe relations

Or

Columns of relations are attributes

# Tuple Or Record

A row in a relation

# Relation Example

The account relation with unordered tuples

<i>account-number</i>	<i>branch-name</i>	<i>balance</i>
A-101	Downtown	500
A-215	Mianus	700
A-102	Perryridge	400
A-305	Round Hill	350
A-201	Brighton	900
A-222	Redwood	700
A-217	Brighton	750

# Database Schema

Logical design of database

# Database Instance

Snapshot of the data in the database at a given instant in time

# Domain

A unique set of values permitted for an attribute

# Domain Constraint

Specifies an important condition that we want each instance of relation to satisfy

# Degree or Arity

Number of attributes in relation

# Cardinality

Number of tuples in a relation

# Relational Database

A relational database is a collection of relations

# Keys

An attribute or set of attributes whose values can uniquely identify a tuple in a relation

# Keys

1. Super Key
2. Candidate Key
3. Primary Key
4. Alternate Key
5. Foreign Key

# Functional Dependency

Consider a relation R and 2 attributes A and B in R.

B is functionally dependent on A (denoted by  $A \rightarrow B$ ), if each value of A is associated with exactly one value in B in relation R.

# Functional Dependency

Consider a relation R and 2 attributes A and B in R.

B is functionally dependent on A (denoted by  $A \rightarrow B$ ), if each value of A is associated with exactly one value in B in relation R.

A	B	C	D
$a_1$	$b_1$	$c_1$	$d_1$
$a_1$	$b_2$	$c_1$	$d_2$
$a_2$	$b_2$	$c_2$	$d_2$
$a_2$	$b_2$	$c_2$	$d_3$
$a_3$	$b_3$	$c_2$	$d_4$

# Functional Dependency: Example

A	B	C
10	B1	1
10	B2	2
11	B4	1
12	B3	4
13	B1	1
14	B3	4

# Functional Dependency

- Functional dependencies play a key role in differentiating good database designs from bad database designs
- A functional dependency is a type of constraint that is a generalization of the notion of key
- $X \rightarrow Y$ , where X is a set of attributes that can determine the value of Y

# Happy Learning.!

