SQL

SQL:- It is stands for structured query language or sequential query language.

As a Test Engineer we are learning SQL to interact (Communicate) with Data Base.

Why we should interact?

We should do backend validation.

Backend Validation:- To check in wheather the front end Data is present in Back-end, this check in process is called Back-end Validation.

Content:-

- Introduction to SQL.
- Basic terms
  - Data
  - Data Base
  - DBMS
  - RDBMS
- Data types & Constraints
- Sub languages
- DQL
  - → Projection
  - → Selection
  - → Joins
- Sub query
- → Functions
- → Co-related Sub query
- → Statements
- → Normalization
- → Views

Who will learn SQL :-

Administrator :- He will learn SQL because of two reasons
(i) He will create DB user.
(ii) He " provide rules & Permission to the DB users.
~~(iii)~~

Developer :- He will design the data Base

Test Engineer :- He will test the Data Base.
He will perform C R U D operation.

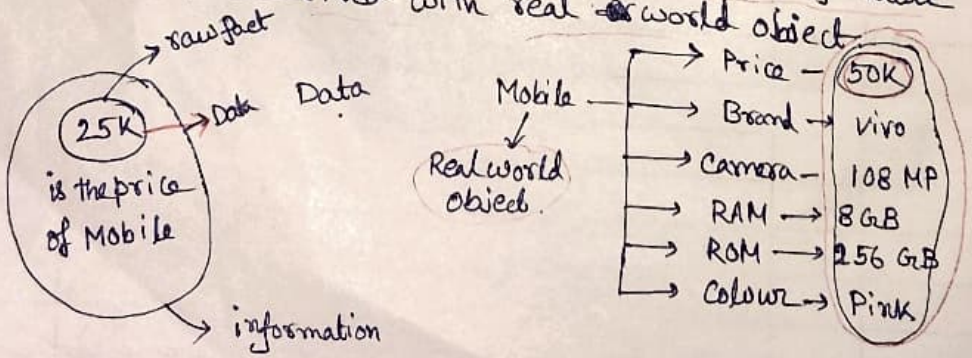
- C → Create (or) Add
- R → Read (or) Retrive
- U → update (or) Modify
- D → Delete (or) Drop.

History of SQL

(i) It was introduced by IBM company in the year of 1970
(ii) Raymand Boyce is the Father of SQL.
(iii) E.F codd is the Co-Father of SQL.
(iv) In earlier days it was called as SEQUEL means → (Simple English query Exicutable Language)
(v) After 1979 it was re-named as SQL and also got ANSI Certificate. (American National standard Institute)

Basic Terms

Data :- It is a piece of value or collection of value which is descriived with real world object.



If Data is described it is called information.
If Data is not described it is called raw fact.
Data will not contains information but information Contains Data.

Data Base:- It is a place used to store the data in systematic & organized way.
Data Base is a collection of data, it is also called as centralized data Base. Because of two reasons -
(i) All type of Data is storing in single place.
(ii) All the data base user can access the data from single place
for Data Base we can perform CRUD operation.
C → Create or add
R → Read or retrive
U → Update or Modify
D → Delete or Drop

DBMS:- it stands for Data Base Management System.
it is a software used to execute all query language.
it is used to maintain & mange the Data Base.

Features of DBMS

(i) We can perform create or add operation C
(ii) " " " Read or Retrive operation R
(iii) We can perform update or Modify operation U
(iv) We can " Delete or drop operation. D
(v) it will provide sequirity Security.
(vi) it will provide rules and permission to the Data Base.

In DBMS the data is stored in file format.
In DBMS we can not establish relationship between two files because (i) different format
(ii) " size.

This is the main draw back of DBMS, to overcome this drawbacks of DBMS we go for R-DBMS

R-DBMS

(i) It is stands for relational Data Base Management

(ii) It is a software used to execute only SQL.

(iii) It is used to maintain and manage the Data Base

Features of R-DBMS: (i) We can perform create or add operation (C)

(ii) We can perform Read or retrive operateration → R

" " " " update or Modify - U

" " " " Delete or drop - D

(iv) It will provide security

(v) It will provide rules and permission to data Base users.

In R-DBMS the data is stored in Table format.

In R-DBMS we can establish relation between two tables by using common column.

Deference between DBMS & R-DBMS

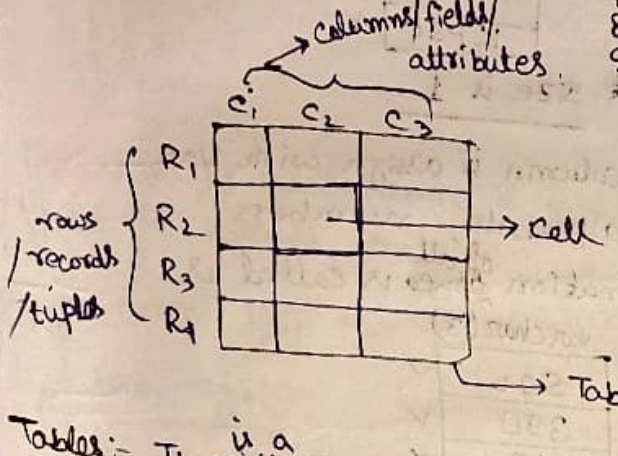
DBMS	R-DBMS
(i) DBMS stands for Data base Management system	(i) R-DBMS - relational Data base management system.
(ii) In DBMS: data store in file format.	(ii) In R-DBMS data store in Table format.
(iii) we cannot establish relationship between two files.	(iii) We can establish relation between two tables.
(iv) It will execute all types of query language.	(iv) It will execute only SQL (Structure query language)
(v) We can not achieve Normalization (at the end of the class)	(v) We can achieve normalization
(vi) It is a single user.	(vi) It is a multiple user.
(vii) It is used in small scale industries. (Startup)	(vii) It is used in large scale industries. (MNC)

R-DBMS s/w are available in the market:-

1. Oracle database
2. My SQL
3. postgresQL
4. Microsoft SQL
5. Maria-dB

1i	10g	12C
2i	11g	18 c
3i		19 C
4i		21 C
5i		
6i		
7i		
8i		
9i		

C → Cloud
g → grid
i → internet.



Tables:- It is a Combination of columns and records

Data types:- The types of Data which is stored inside the columns is called as data types.

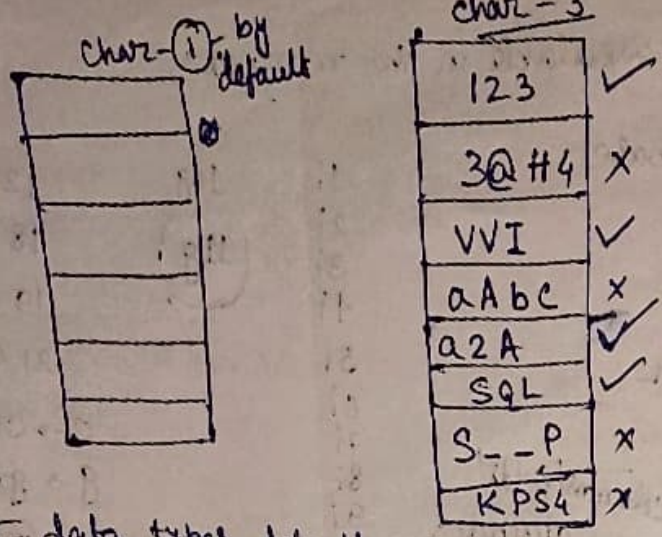
Data types

* Data types are mandatory
  we can not assign multiple data types for a single Column.

There are Four data types
(i) character data type (Char)
(ii) Variable character (Var char)
(ii) Number data type (Number)
(iii) Date " " (Date)

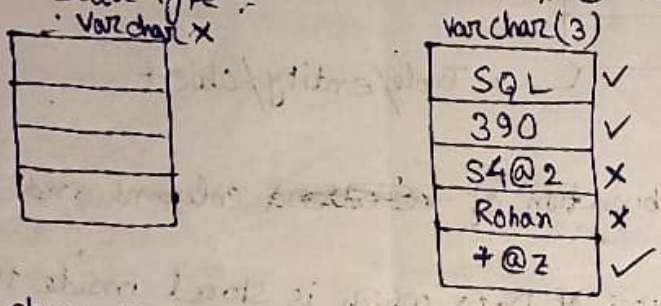
Character Data type:-

If the column assign with character data type it allows alphabets (A - Z / a - z), number (0-9), Special char (+, -) and Combination of all is called character Data type.



In character data types, default size is 1

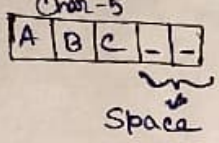
Varchar:- Data type:- If the column is assign with Varchar data type it allows alphabets, numbers, special character and combination of all is called as Var char Data type :-



In Varchar Data types there is no default size

(*) What is the Difference between Char and Varchar

Char



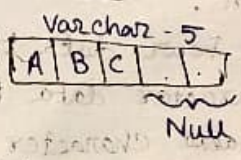
(i) unoccupied memory blocks are or filed with space, space is a special character, so it will consumes memory.

(ii) It is fixed allocation memory.

(iii) Default size is one

(iv) Max size is 2000

Varchar



(i) Unoccupied memory block filled will null, so null will not consumes any memory.

ii) It is variable allocation memory.

(iii) There is no default size

(v) Max size is 4000

Note:- From 10g version Varchar was renamed as Varchar2.

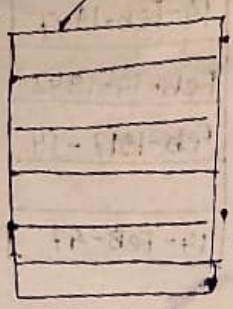
Number Data type

44·44
P→4 Nu
S→2

Number (3)

420	✓
90	✓
180	✓
12A	✗
6231	✗

Number (4,2)



P→4 ✓
<4 ✓
>4 ✗

S→2
=2
1 <2 ✓
3 >2 ✗

(P-S) 4-2=2
<2 ✓
>2 ✗

Defination:- If the column is assign with Number data types it allows only numbers.

Nu From 10g version for number data type we will provide two datatypes Number (input 1, input 2)
i/P₁ , i/P₂

i/P₁ → Precision
It represents total no of digits includes decimal values.

i/P₂ → Scale
it represents total number of decimal values

i/P₁ − i/P₂ → P-S → It represents total number of integer values

Number (4,2)

42·44	✓
4·444	✗
444·4	✗
4·44	✓
4·4	✓
0·44	✓
444	✗

P→4
=4 ✓ <4 ✓
>4 ✗

S→2
=2 ✓
<2 ✓
>2 ✗

P-S → 4-2 = 2 ✓
<2 ✓
>2 ✗

Date data types:- If the column is assign with date data types it only allow date value.

we are using oracle s/w, we should follow two types of date format. that is DD-MON-YY or DD-MON-YYYY

→ Data

14-Feb-1947 ✓
Feb-14-1942 ✗
Feb-1947-14
14-FeB-47 ✓

Constraints:-

- It is set of rules applied for a column
  " " optional but highly recomended.
- We can assign multiple constraints for a single Columns
- There are five types of constraints
  1. NOT Null
  2. Unique
  3. Check
  4. Primary key
  5. Foreign key

Null:- 'Null' means empty
It will not consumes any memory.
0 & Space is not a null.
If any arithmatic operation is done with null, it will provide output as null.
in bellow example both the nulls are different.

	C_1	C_2	C_3	
R_1	1	2		→ Null
R_2	3	4		→ Null
R_3	5	6	7	
R_4	8	9	10	

$R_1C_3 \neq R_2C_3$
Null ≠ Null.

Not Null:- If the coloumns is assign with Not Null ~~with not null~~ constrain.
it will ~~not~~ accept duplicate values but it will not accept null values.

Varchar(3) → Not Null.

SQL ✓
420 ✓
SQL ✗
420 ✓

✗
420 ✓
✗

UNIQUE:- If the column assign with UNIQUE constraints it will accept NULL values but it will not accept duplicate values.

SQL	✓
420	✓
SQL	✗
420	✗
	✓
420	✗
	✓

Check :- It is used to store the set of values.

Example.

Age → Number(2)

check 'Age >= 18)

45	✓
55	✓
17	✗
99	✓
100	✗

Condition

(1--17) ✗ (18 .. 099)

It is called as domain constraints.

Primary key:-

It is used to identify the unique records in a table.

If the column is assign with not null and unique constrients this columns are eligible for become a primary key. example- (Sid, email id, Mobile Number)

This columns is called as Candidate Keys.

ex

Sid → Primary key

email id → Alternate

mobile No → Candidate key

(because number varchar, constraint → Unique, & not null)

Among the Candidate Keys the columns which is assign with number data type and less size is selected as Primary key.

ex:- Sid.

Alternate Key:-

The columns are eligible for become a primary key but not a primary key. This type of column key is called Alternate key.

ex → email id, mobile no.

Note:-
In a table only one primary key column can be present.

If a table is consists of primary key column. that table is called as parents or Master Table.

Foreign key:

It is used to establish the relation ship between two tables

One table of primary key column (of one table) is eligible for become a foreign key for another Table.

→ Foreign key will accept null values and Duplicate values but is will not accept the value which is not present in the primary key column.

⇒ In a table multiple foreign key column can be present.

⇒ If the table is consist of foreign key column that table is called Child or Slave Table

Student ID (Number Data Type(8) not Null Unique)	Student Name	Registration No. (Number Data(10) Not Null Unique)	Roll No. (Number Data(8) Not Null Unique)
1	A	.....	.....
2	B	.....	.....
3	C	.....	.....
4	A	.....	.....
5	C	.....	.....
6	X	.....	.....
7	Y	.....	.....
8	Z	.....	.....

Serial No.	Events	Student ID (Foreign Key)
1	Cricket	1
2	Football	2
3	Hockey	4
4	Ludo	3
5	Kabaddi	5
6	Tenis	6
7	Badminton	7
8	Cards	
9	Carroms	
10		

If the columns are assign with Not Null and Unique constraints then (Student ID, Registration No, Roll NO) All Columns are eligible for Primary key. These columns are called as Candidate Key.

Among the candidate key which is assigned with Number data type with less size is called as primary key ex- Student ID.

Q> Deference between Primary Key & Foreign Key.

Prim	Foreign Key
It is used to indentify the unique records in a Table	it is used to establish relation between 2 or multiple table
If the column is assign with Not null and Unique Constrients that column is elegible for become a primary key. It will not accept Null values and Duplicate values	One table of primary of column is eligible for ~~for~~ become a foreign key for another table. It is will accept Null values and Duplicate Values But it will not accept the value which is not present on pr,
In a table only one primary key column.	In a table multiple foreign key column can be present.
If a table consists of primary key column then it is called parent.	If the table constis of foreign key column, then it is called Child or Slave table.

Sublanguages / statements / commands

DQL [Data query Language] / DRL [Data retrivel lang]
* Select

DDL [Data definition lang]
Create, alter, rename, truncate, drop → Dr. Cat → create, truncate, alter, drop, rename

DML (Data manipulation Lang.]
Insert, update, delete

TCL (Transaction Control lang]
Rollback, Savepoint, Commit

DCL (Data control ~~lang~~ lan]
grant, revoke

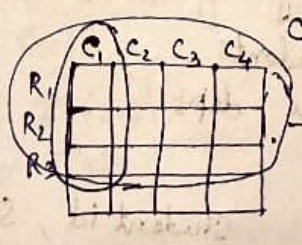
DQL:- It is used to fetch the Data from the database

Select:- It is used to fetch the data from ~~database table~~ database table ~~column~~ by selecting particular ~~table~~ is called as 'selelct'. By using 'Select' statement we can perform 3 operations

- Projection
- Selection
- Joins

Projection:- It is used to fetch the data from the table by selecting particular column or all the columns but by default it will select all the records is called as projection.

Projection



$C_1/R_1 + R_2 + R_3$

$$\frac{C_1 + C_2 + C_3 + C_4}{R_1 + R_2 + R_3}$$

Syntax

Statements Keywords (2): Select */Col-name/ Distinct Col-name/ Expression as alias name

clauses (1): from Table Name;

Students ——————— DB

Sid	Sname	Dept	Per	Mobile No
1	Kalia	eating	99	–
2	Didi	TMC	33	–
3	Rahul	Political science	90	–
4	Sunny	CSE	40	–
5	Jonny	ME	69	–

Write a query to display the students name for all the students

② Select Sname
① from Students;

From clause o/p

Sname
Kaliya
didi
rahul
Sunny
Johny

Write a query to display the students name with Dept & Per.

→ select Sname, Dept, Per
from students;

(*) Write a query to display the percentage for all the students.

Select per
from Students;

(*) Write a query to display student id, & student Name for all the Students

select sid, sname
from students;

(*) write a query to display student name & Departname

Select Sname, Dept
from students;

(*) Write a query to display the student id, Student Name and dept for all the students.

Select Sid, Sname, dept
from students;

(*) write a query to display Student id, Student Name, Dept, percentage & mobile no./or write a query to display the students details for all the student

Select sid, Sname, Dept, per, Mobile No
from Students;

Select *
from students;

Write a query to display Student details for all the students along with display the department for all the Students.

Select Sid, Sname, Dept, per, Mobile No, Dept
from students;

or

Select *, dept
from students;

90% ✓ - 10% ✗

Set

Set lines 100 pages 100; (definately

Tab:

Select * from tab;

Describe:

Desc TN;

Connect :-

Conn

Basic Commands:

Set: It is used to set the line size and pages size.

Syntax: Set lines 100 pages 100

Tab: It is used to display the all table Names in particular Data base.

Syntax: Select * from
from Tab;

Describe: It is used to ~~describe~~ display the table structure.

Desc Table Name;

Connect:- It is used to Connect from one data base to another data base

Syntax Conn

---

(1) Write a query to display the employee names for all the employees

```
select ename
from emp;
```

(2) Write a query to display the designation for all the employees:

```
select Job
from emp;
```

(3) Write a query to display the joining date for all the employees.

```
select hiredate
from emp;
```

(4) Write a query to display employee Number name & ~~Employee Salary~~ for the employee

```
select empno, ename, sal
from emp.
```

(5) Write a query to display the, employee Number, Name, designation, manager number & department number

```
select empno, ename, Job, Mgr, deptno
from emp
```

(6) employee Number Name, designation, manager number, hiredate, Salary and Comission for all the employees.

```
Select emp No, ename, Job, Mgr, hiredate, Sal, comm
from emp;
```

(7) employee details for all the employees

```
select *
from emp;
```

(8) (*) write a query to employee details for all the employees and also display the manager number for all the employees.

Select *, Mgr
from emp;

Select table name.*, columns
from Table Name;

Select emp.*, mgr
from emp;

Note:- (*) is always Single, it will not mingle with its having secret affair with table Name

Write a query to display the girls

Write " " " the designation

Select

Unique Combination of Job, manager, dept no.

Distinct clause:-

Whenever we want to remove the duplicates in the output, then we ~~sh~~ will go for distinct clause.

Syntax:-

Select distinct col-name
② ③
from Table Name;
①

Distinct clause:-

It will execute row by row

It should be used in 'select' clause

It should be written before column-name

It will remove the duplicates ~~from~~ only from output, not in the table.

Couples

bname	gname	Types of marriage
Vishwanath	Benkilata	Love
Piles Anbi	Sunny	Child
Rocket Rittik	Mia	forcefully
Andu Aniket	Sunny	Online
Donkey Debit	Angel Priya	facebook
Vishwanath	Avantika	running

→ from clause O/P

Select distinct gname
② ③
from couples
①

gname
Benki
Sunny
Mia
Sunny
Angelpriya
Avantika

~~gname~~
Benki
Sunny
Mia
Angel Priya
Avantika

1) WAQTD the designation (without duplicates)
Select distinct Job
from emp;

2) WAQTD the unique manager numbers.
Select distinct mgr
from emp;

3) WAQTD the depart no. (without repetation.)
Select distinct Dept no.
from emp;

4) WAQTD the unique manager number department No.
Select distinct mgr, Dept No
from emp;

5) WAQTD " combination of Designation, Manger no, department No.
Select distinct Job, Mgr, Dept No.
from emp;

Expression → (a+b) → operator
operands

Date - 19/1/2024

Expression:- It is a combination of operator and operands is called as expression.

Character & date values we should use ' '

[∵ {'A' + 1} {'14-Feb-2024' + 1)

For number & column names - we should not use any single quets (Sal * 12)

1. WAQTD Anuark Salary for all the employees. Select Sal * 12 from emp;

2. " 75% of the Salary values. select Sal * $\frac{3}{4}$ from emp;

3. WAQTD Salary values with a penalty of 1000 RS/
Select Sal - 1000
from emp;

4) WAQTD Anual commission & half term commission select Sal * 12, Sal * 6

5) WAQTD the 50% of hike Salary. Select Sal * $\frac{3}{2}$ from emp;

6. WAQTD the 50% of deduction of Salary Select Sal * $\frac{1}{2}$

7. WAQTD the 85% of Salary values. Select Sal * $\frac{17}{20}$ from emp;

Alias:- Whenever to go for Alias?

A) Whenever we want to change the column Name or functions or expression, then we should go for Alias Name.

Syntax:

Select { Col Name [as] alias Name
or
expression [as] alias Name
or
Functions [as] alias Name }
from Table Name;

1. as- Keyword is optional
2. It is used to get user convinient output
3. Alias Name will change only on the output but not on the table.

4) If Alia's Name is Start with Number or it Contains any special character we should us within " "

5) contains Keyword we should use within " "
ex- Select Sal 1.5 "Select" from emp;

Selection :-

When to go for Selection ?

To overcome the drawback of Projection we go for selection.

What :-

It is used to fetch the data from the table by selecting particular Column or all the Columns along with particular records is called as Selection.

Syntax

③ Select * / Col-name
① from Table Name
Where Condition ;
②

IPL

Pname	JNO.	Tname	runs
MSD	7	CSK	1000
VK	18	RCB	10,000
RS	45	MI	500
Gill	77	GT	100
Rinku	69	KKR	150

③ Select *
① from IPL
② where runs > 500 ;

from clause O/P

WAQTD the players details who's runs is more than 500

③ Select *
① from IPL
where runs > 500 ;
②

PName	JNO	Tname	Run
MSD	7	CSK	1000
VK	18	RCB	10000

Pname = 'MSD'

1000 > 500 → T → ✓ LHS Operator RHS ;
10000 > 500 → T → ✓
500 > 500 → F ✗
100 > 500 → F ✗
150 > 500 → F ✗

~~WAQTD the J.Name who's~~

Note :- Character & Date values are case-sensetive
" " " we should use in ' '

) WAQD the Jersey No of Rinku.

```
Select J.NO
from IPL
where JNO = 'Rinku';
```

② WAQD the Team No for MSD.

```
Select Tname
from IPL
where Tnam = 'MSD';
```

③ WAQTD whose Joursey No. 45.

```
Select P Name
from IPL
where JNO = 45;
```

Q from Emp table

① WAQTD, Employee details whose Salary is more than 2000.

```
Select *
from emp
where SAL > 2000;
```

② WAQTD. Employee details who had Joined on 3rd december 81

```
Select *
from emp.
where hire date = '3-Dec-81';
```

③ WAQTD Employee Names who are working as Manager.

```
Select Ename
from emp
where Ename = 'Manager'.
```

4) WAQTD Salary to King.

```
select SAL
from emp
where SAL = 'King'.
```

5) WAQTD Employee details who are working in department 20.

```
Select *
from emp
where dept No = 20;
```

⑥ WAQTD Emp. details who has earning salary < 2000.

```
Select *
from emp.
where Sal < 2000;
```

7) WAQTD Emp. Names who are report into 7698.

```
Select Emp Name.
from emp
where MGR = 7698;
```

8. WAQTD. Emp. details of Smith

```
select *
from emp
where ename='SMITH';
```

9. WAQTD the Commission value for Martin?

```
Select COMM
from emp
where Ename = 'Martin';
```

10) WAQTD. Emp. details whose commission is <1000

```
Select *
from emp
where COMM < 1000;
```

⊛⊛⊛ order by clause

When to go for order by Clause?

When ever we want to arrange the output in assending or decending order, then we should go for order by clause.

Syntax:-

```
③ Select Col-Name
① from Table Name
② where Condition
④ order by Col-Name [ASC]/DESC;
```

→ ASC keyword is optional. (by default it will arranging assending order).

→ Data will arrange only in the output, but not in the table.

→ It will execute end of the query.

→ It should be written last line of the query.

1) WAQTD SAL values in increasing order.

```
Select SAL
from emp
~~where condition~~ order by SAL;
```

2) WAQTD commissing Value in decrease order.

```
Selec COMM
from emp
order by Comm desc;
```

Q. WAQTD Commission value in accending order.

```
Select COMM
from emp
order by COMM;
```

Q WAQTD. Salary values in desending order

```
Select SAL
from emp
order by SAL DESC;
```

Q) WAQTD. Employee names in alphabetic order

```
Select Ename
from emp
order by Ename;
```

Operators

It is a special symbol use to perform particular operation

there are 5 type of operators.

1. Arithmatic operator → +, −, /*
2. Relational operator → <, >, <=, >=, =, ≠
3. Logical " AND, OR, NOT
4. Special " → In, between, like, is, Not In.
   Not between, NOT like, is not.
5. set operator → Union, union all, intersect, Minus.

AND

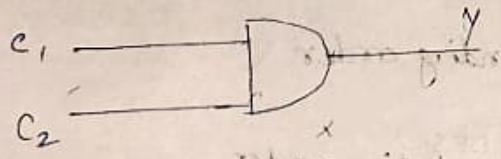
when ~~to go for~~ Logical operator.

whenever we want to write multiple condition in where clause the we should go for logical operators

AND

When ~~go for~~

when ever we want to satisfy both the condition than we should go for And operator



Truth Table

C_1	C_2	Y
0	0	0
0	1	0
1	0	0
1	1	1

0 → F
1 → T

C_1	C_2	Y
F	F	F
F	T	F
T	F	F
T	T	T

③ Select ⊛/column Name
① from Table Name
② where condition (and Conditional).

OR

whenever we want to satisfy any one condition, then we should go for OR operator.

C_1	C_2	Y
F	F	F
F	T	T
T	F	T
T	T	T

Syntax

③ Select *|Column-Name
① from TN
② where Condition 1
OR Condition;

boys

bName	Job	Sal
Pikes	Hoopic	4
Sutta	TE	5
Dalal	Broker	8
Mottu	DEV	2
420	SE	0.5
Rocket	SE	6

WAQTD boys names who are working as a SE & salary should more than 4

③ Select bname
① from boys
② where JOB = 'SE'
and SAL > 4;

bname
Rocket

from clause O/P

C_1 JOB = SE	and	C_2 SAL > 4	O/P
hoopic = SE → F		4 > 4 → F	F ✗
TE = SE → F		5 > 4 → T	F ✗
Broker = SE → F		8 > 4 → T	F ✗

NOTE: If condition from same column use OR
" different " " AND.

① WAQTD. emp. details for the employee who are report in 7698 and were working on the department 20

Select *
from emp
where MGR = '7698'
and dept NO = 20; → error

Select *
from emp
where MGR = 7698 And
dept NO = 30

② WAQTD emp details who Joined 17-dec-80 or 3rd dec-81

```
select *
from emp
where HireDate = '17-Dec-80' OR '03-Dec-81'
```

③ WAQTD emp details for the employee who had worked in the department no 20 & 30

```
Select *
from emp
where DeptNO = 20 OR DeptNO = 30
```

4) WAQTD Emp. details for Smith and Allen

```
Select *
from emp
where ename='SMITH'
```

5. " Emp names who are report into 7698 or 7788.

```
Select Ename
from emp
Who
```

6) WAQTD emp. details who are earning Sal more than 1000 and were getting commission < 1000.

```
Select *
from emp
Wh
```

⑦ WAQTD emp. Names who are report into 7839 and working as a manager.

```
Select ename
from emp
```

⑧ WAQTD emp. Names who are working as a Sales man or Analyst.

```
Select ename
from emp
where Job='Salesman' OR Job='Analyst';
```

⑨ WAQTD emp. details whose empno. is 7698, 7788, 7839 and 7844.

```
Select *
from emp
where empno = 7698 OR empNO = 7788 OR empno=
```

10. WAQTD The emp. details for the employee who are earning salary 800, 1250, 1600 & 3000

```
Select *
from emp
~~Where empno = 7698, or empNo=7788~~
where Sal = 800 OR Sal = 1250 OR Sal = 1600 OR
Sal = 3000;
```

or Select *
from emp
where Sal in (800, 1250, 1600, 3000);

Select *
from emp.
where dept no not in (20, 30);

Q) When to should go for In operator

1. Instead of using multiple OR operator we should use In operator.

2. Instead of using = operator we can use In operator.

Syntax:- LHS IN (RHS$_1$, RHS$_2$, RHS$_3$ ...)
LHS IN RHS

Q) NOT IN Operator

It works exactly opposite to IN operator,

Q) WAQTD Employee details who are working in department no 20 & 30.

Select *
from emp
where deptno In (20, 30)

2. WAQTD the Emp. details who are joined on '09-Jun-81', 17th NOV-81, 22-feb-81,

Select *
from emp
where hiredate in ('09-Jun-81', '17-NOV-81', '22-feb-81').

3. WAQTD the Emp. details who are not working as a manager and Salesman.

Select *
from emp
where Job Not in ('manager', 'Salesman');

4. WA emp. details of Martin, King, turner and James.

Select *
from emp
where ename in ('Martin', 'king', 'turner', 'James')

5. WA emp. details who are not reporting to 7698, 7839 and 7782.

Select *
from emp
where mgr Not in (7698, 7839, 7782).

(*) Between Operator

When to go for Between operator:-

Whenever we want to fetch range values then we should go for Between operator.

Syntax: LHS Between LLV and HLV

LLV → Lower limit value

HLV → Higher limit value

2. It is also called as Inclusive operator. because
   (i) It includes starting value and ending value

It is also called as Range operator.

NOT Between operator

It will works exactly opposite to the Between operator.

Syntax:- LHS NOT Between LLV and HLV.

1. emp. details who are getting commission between 500 to ~~2000~~ ~~2500~~ 1500:

```
select *
from emp
where comm between 500 and 1500;
```

2. WAQTD emp. details who are getting Salary from 2000 to 3000

```
Select *
from emp
where Sal between 2000 and 3000;
```

3. WAQTD emp details who are joined in the year of 81

```
Select *
from emp
where hire date between '01-JAN-81' and '31-DEC-81';
```

4. WAQTD emp. details who are not getting commission 500 to 1000.

```
Select *
from emp
where comm NOT in 500 and 1000
```

5) WAQTD the emp. Names who are not joined in the year of 81

```
Select *
from emp
where hiredate not between ('01-JAN-81' and '31-DEC-81')
```

When we want to get null value we use IS operator.

IS operator

When to go for IS operator.

When ever we want to fetch null value we use IS operator

Syntax:- LHS IS NULL ;

[If there is (not) in question go for is operator. If there is no not go for is not]

IS NOT operator

It will work exactly opposite to IS operator,

LHS IS NOT Null ;

1. emp. details who are not getting commision

```
Select *
from emp
where comm is null
```

2. emp detail who are getting commission

```
select *
from emp
where comm is not null
```

3. emp. details who are not report to any manager

```
select *
from emp
where mgr is null;
```

4. emp details for the employee who are report any manager.

```
Select *
from emp
where mgr is not null.
```

Like Operator

Q) When to go for like operator?

Whenever we want to fetch data based on characters then we should go for like operator.

→ Like operator will perform 1. 'Pattern Matching'
& 2. 'Wild card search'

→ Syntax: LHS Like 'Pattern'

Pattern:-

It is a combination of ordinary character (OC) and Meta character (MC).

'S%'
OC MC

Meta character

It is a special character used in like operator. There are two types of Meta Character:

'%' → It represents 'n' number of characters

'_' → It represents single characters.

Pattern Matching

Whenever we know the exact ~~potion~~ position of the ordinary character. This type of pattern is called as pattern Matching.

Wild card Search

Whenever we don't know the exact position of the ordinary character, this type of pattern is called Wild card Search.

1. WAQTD the emp. Names, the names Should start with the character 'J'.

```
Select Ename
from emp
where enames like 'J%';
```

2. WAQTD the Emp. Names which ends with character 'N'

```
Select Ename
from emp
where ename like '%N';
```

3. WAQTD the designation which contains character 'L'

```
Select Job
from emp
where Job like '%L%';
```

4. WAQTD the Job value which contains character 'A' two ~~times~~ times.

```
Select Job
from emp
where Job like '%A%A%';
```

5. WAQTD the Emp. No which starts with '7' and end with '8'

```
Select EmpNo.
from emp
where EmpNO like '7%8';
```

6. WAQTD the Emp. details who were joined on the year of '81' (any date, any month)

```
Select *
from emp
where Hire date like '%81';
```

7. WAQTD the Emp. details who were Joined in the month of december (any date any year).

```
Select *
from emp
where Hire date like '%DEC%';
```

8. WAQTD the Job value which contains charter 'M' in 3rd position from the last.

```
Select Job
from emp.
where Job like '%M--';
```

9. WAQTD the Emp. name the names Should not consists 5 charaters

```
Select Ename
from emp
where ename not like '-----';
```

10) WAQTD the Emp. details who are earning 3 digit Salary.

```
Select *
from emp
where SAL like '---';
```

11. WAQTD. the Emp. names, the names should starts with S And A

```
Select Ename
from emp
where ename like 'S%'
or ename like 'A%';
```

Ename Like in ('S%', 'A%') is not give output different patterns.

(Set Operator)

Q. When to go for set operator?

When ever we want to join two queries output then we should go for set operator.

There are 4 types of set operator.

1. Union :- It will provide both common & uncommon values without repetation.
2. Union all :- It will provide both common & uncommon values with repetation.
3. Intersent :- It will provide only common values.
4. Minus :- It will provide only uncommon values from first set.

Set

$A = \{1, 3, 5, 7, 9\}$

$B = \{5, 7, 9, 11, 13\}$

$A \cup B = \{1, 3, 5, 7, 9, 11, 13\}$

$A \cup_{All} B = \{1, 3, 5, 7, 9, 5, 7, 9, 11, 13\}$

$A \cap B = \{5, 7, 9\}$

$A - B = \{1, 3\}$

$B - A = \{11, 13\}$

Select ename
from emp
where deptno = 30

4. UNION

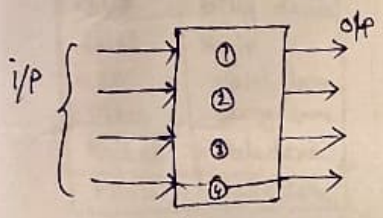
5. Select ename
from emp
where job = 'SALESMAN';

Functions

It is a set of instruction used to perform particular operation. There are two types of functions.

1. Single Row Function
2. Multi Row Function

Single Row Function



- It will take multiple i/p and provide multiple o/p.
- It is take single input at a time, it will execute and it will provide output.

output is depending on number of inputs.

Multi Row Function

It will take multiple i/p and it will provid single o/p.

- It take multiple input at a time, it will create the group and it will provide single output.

Output is depending on number of groups

MRF function is also called as aggregate function or group function.

There are five times of multi row functions

1. Maximum. It will provide maximum value.

Syntax → Max(i/p)

2. Minimum Value - It will provide minimum value.

min (i/p)

3. Average - It will provide Avg. Value.

Avg. (i/p)

4. Sum - It will provide total Value.

Sum (i/p)

5. Count - It will provide total number of records inside the table.

Syntax :- Count (*)

1. WAQTD the maximum Salary.

```
Select max (Sal)
from emp;
```

2. WAQTD. the number of employees in Dep. no 30.

```
Select count (*)
from emp;
where dept. no. 30.
```

3. WAQTD the minimum comm.

```
Select min (comm)
from emp;
```

4. WAQTD the maximum Salary, lowest comm, Avg Salary total commission, and no. of employees.

```
Select max (Sal), min(comm), Avg (Sal), Sum(Comm)
```

5. WAQTD the recent hire date. and oldest hire date:

```
Select max(hire date)
from emp;
Union
Select min(hire date)
from emp;
```

```
Select max (hire date),
min hire date
from emp;
```

Spool Drivename:/file Name.TXT; → to Save the queries

Group by clause

When to go for group by clause

Whenever we want to fetch column name along with MRF then we should go for Group by Clause.

What is

It is used to creat No. of groups inside the ~~Column~~ Multi Row Function (MRF)

Syntax:-

4 Select col-name, MRS
1 from TN
2 where condition
3 group by col-name;

Dance

Dname	Type of dance
Sutta	Pole dance
Kalia	Belly dance
dalal	Nagin "
420	Matal dance
Piles	Lungi dance
Rocket	Pole dance
Motu	Belly dance

WAQTD the No of dances for each type of dance

3 Select Type of dance, Count(*)
1 from dance
2 group by Type of dance;

from clause o/p

Pole Dance

dname	Type of Dce
Sutta	Pole Dance
Rocket	P.d.

Belly dance

Dname	Tod
Kalia	bd
Motu	bd

Nagin dance

D name	TOD.
Dadal	N.d.

matal

dname	Tod
420	matal

Lungi

Dnam	Tod
Piles	Lungi

output

TOD	
Pole dan	2
belly "	2
Nagin "	1
Matal "	1
Lungi "	1

It will execute row by row.

It will create no of groups based on number of unique present inside the column, which is mentioned in group by clause.

1. WAQTD the Highest Salary for each manager number?

Select Max(SAL), ~~count(*)~~ / Select Mgr, max(Sal)
from emp
group by SAL
from emp
group by SAL;

2. WAQTD the No. of employees for each department no. for the employee who go hired in the year of 81

Select Dept No, count-(*)
from emp
~~group by Dept No~~ where Hire date between '01-Jan 81' and '31-Dec-81'
where

3. WAQTD the no. of employees for each designation except salesman.

Select (Dept No) Count (*)
from emp
where Job in not 'Salesman'
group

4. WAQTD the Total Salary for each designation

~~Select Job Sum Job~~ Select Job, Sum(Sal)
from emp
group-Jo

5. WAQTD the Total commission for each department No.

Select Dept No, Sum (com)
from emp
group by dept No;

6. WAQTD the No. of employees for each manager No, who are earning 4 digit Salary.

Having clause

When to go for Having.

→ When ever we want to write MRF as a condition then we should go for having clause.

→ It is used to filter the groups.

Syntax:-

5 Select col-name, MRF
1 from TN
2 where condition
3 group by col-Name
4 having MRF Condition
6 order by colname AS/DESC.

It ~~is with is~~ will execute group by group.
without group by clause having clause will not execute.
It will execute after group by clause.

Having clause

Q> WAQTD the No. of Dancers for each type of dace if No. of Dancers is more that 1

Dance

Dance Name	Type of Dance
Sutta	Pole Dance
Kalia	Belly dance
dalal	Nagin dance
420	Matal dane
Piles	Lungi Dane
Rocket	Pole dance
Motu	Belly dane

```
Select type of dance, count (*)
from dance
group by Type of dance
having count (*) > 1;
```

Type of Dance	Count
Pole Dance	2
Belly Dance	2

Pole Dance 2>1

Name	T.o. Dance
Sutta	Pole Dance
Rocket	Pole Dance

Belly Dance 2>1

Name	T. of Dance
Kalia	Belly Dane
Motu	Belly Dane

Nagin dance

Name	T.o.D
Dalal	Nagin Dane

1) WAQTD the no. of employee for each department no. if no of employees is more than 3

```
Select Dept No, count(*)
from emp
group by dept no,
Having count(*) > 3;
```

2. WA the total Salary for each managers number for the employee who got hired in the year of 81 and total salary is <2000

```
Select mgr, Sum(SAL)
from emp
where hire date Like '%81'
group by mgr.
Having SAL < 2000
```

3. WAQTD the Avg. Salary for each designation except president if avg. Salary is more than 1000.

```
Select ~~Avg(Sal)~~ mgr, avg(SAL)
from emp
```

4. WAQTD the no. of employees for each department no. who are earning 4 digit Salary if the no. of employees is less than 5

→ Select dept no., count (*)
from emp
where sal like '----'
group by dept no
Having count (*) < 5;

Where clause	Having clause
i) where clause is used to filter the records	i) It is used to filter the groups.
ii) It will execute row by row.	ii) It will execute group by group.
(iii) It will execute before group by clause.	(iii) It will execute after group by clause.
(iv) It is independent.	(iv) It is dependent.
(v) It will not allows MRF conditions.	(v) It will allows MRF conditions.

Subquery.

Query:- It is a set of statements used to interact with the Data Base.

Subquery:- The query which is written inside the another query is called as subquery.

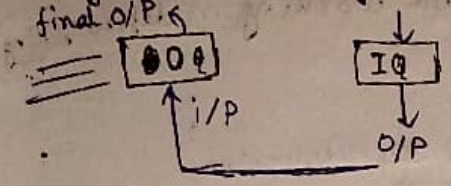
The query which is written inside the bracket, this query is called as inner query.

→ The query which is written outside the bracket this query is called as outer query.

→ Subquery is the combination of inner query and outer query.

→ Here we ~~can~~ can write upto 255 queries. It is also called as nested query.

Working mechanism of sub query.



Inner query will execute, and it will provide output. The output of inner querry is given to the input for the outer query.

→ Outer query will execute all the statement and it will provide final output.

→ Outer query is depending on inner query. And inner query is independent.

Case-1

WAQTD Animal Name which speed is more that 'hen'

```
select Aname
from animals
where speed > 'hen';
```

```
select speed
from Animals
where Aname = 'hen');
```

Q) When to go for Subquery why Subquery?

Case-1:- Whenever we want to fetch unknown values then we should go for Subquery case-1

Animals

Aname	Colour	Speed
buffallo	black	25
Cat	white	20
Pig	Pink	15
Hen	black	23
Horse	brown	155

where clause O/P

From clause O/P

```
Select Aname
① from Animals
② where speed >
```

Aname
Buffalo
Horse

```
Select speed
① from Animals
② where Aname = 'hen');
```

→ 23

Speed > 23

25 > 23 → T → ✓ 155 → 23 → T → ✓

20 > 23 → f → ✗

15 > 23 → f ✗

23 > 23 → f → ✗

1. WAQTD the emp names who are working in dept no of King

```
Select enames
from emp
where depart no = (Select department no
                   from emp
                   where ename = 'King');
```

2. WAQTD the who are report into manager no of Martin

```
Select ename
from emp
where MGr = (Select MGr
             from emp
             where ename = 'Martin');
```

3. WAQTD the employee details who are joined after 'Jones'

```
Select (*)
from emp
where hiredate > (select hiredate
                  from emp
                  where hire date = 'Jones');
```

4. WAQTD the employee details who are working in the department of 'Ward' and 'Jones'.

```
Select *
from emp
where depart no = (select depart no
                   from emp
                   where ename in ('Ward', 'Jones'));
```

in

There are 2 types of Subquery:-

(i) Single Row Subquery
(ii) Multi Row Subquery.

If inner query provide single output it is called as Single row Subquery.

→ If it is a Single row Subquery we can use all relational operator, & IN and NOT IN operator

MultiRow If inner query provides multiple output it is called as multirow subquery.

→ If it is a multirow subquery where we should use IN or NOT IN operator.

If we can't able to identify whether the subquery is single row or multi row, so we should use only IN or NOT IN operator.

Subquery case-2

When to go when ever we want to fetch the data from 1 table but condition is present in another table then we should go for subquery case-2

Products

Pid	Pname	Price
1	Lipstick	2500
2	T-Shirt	150
3	Helmet	1500

Cid	Cname	Pid
1	Asian Paints	1
2	PAMA	2
3	Steel bird	3

from clause o/p

from clause output

WAQTD the Product name which is manufactured by Asian paints

```
③ Select Pname
① from products
② where Pid = ( Select Pid → 1
                ① from company
                ② where c.name = 'asian Paints' );
```

① WAQTD the employee details who are working Rechearch department:

Select *

2. the Department details of 'Smith':

③ employee details who are not working in research dept.

4. w. the Depatt details for the employee who are getting commission.

5> Depatment name and Location for the employee who are earning 4 digit Sarary.

6. Employee Names who are working in Chicago Location.

Case-3

WAQTD 2nd Max (SAL)

Subquery case-3

Whenever we want to fetch nth minimum or nth maximum value then we should go for Subquery case 3

For 2nd Max < 1st max

~~1st max~~

2nd min > 1st min

Emp

Ename	Sal
Sukhendu	250
Purnendu	300
Abhishek	800
Dalal	150
Abhishek	100

WAQTD 2nd max Sal

③ Select max (Sal) → 300

① from Emp

② Where Sal < (Select max (Sal) → 800

① from emp);

from clause o/p

Sal < 800

250 < 800 → T ✓

300 < 800 → T ✓

800 < 800 → F ✗

150 < 800 → T ✓

100 < 800 → T ✓

Note

For N times,

Select = N time

① WAQTD the 5th maximum Salary.

```
Select Max (Sal)
  from emp
where sal < ( Select Max (Sal) from emp where Sal < (select Max (Sal)
  from emp where sal < ( Select Max (Sal) from emp where
  Sal < ( Select Max (Sal) from emp )))));
```

② WAQTD the 9th minimum Salary.

```
Select Max (Sal)
  from emp
  where sal < (Select Max (Sal)
                from emp
                where sal < ( Select Max (Sal)
                              from emp
                              where sal < (Select Max (Sal)
                                           from emp
                                           where sal < (Select Max
                                                        from
                                                        where Sal
(Select Max (Sal)
  from emp
  where sal < (Select Max (Sal)
               from emp
               where Sal < (
```

③ WAQTD the 6th recent hiredate.

4) WAQTD 3rd oldest (minimum) hiredate.

```
select Min (Hire date)
    from emp
    where Hiredate > ( select Min (Hire date)
                       from emp
                       where Hire date > (select Min (Hiredate)
                                          from emp)) ;
```

5) WAQTD. the employee details who are getting first maximum Salary.

```
Select *
from emp
where Sal = (select Max(Sal)
             from emp);
```

6. WAQT the 1st and 2nd maximum Salary.

```
Select max (Sal)
    from emp
    Union
    Select max (Sal)
        from emp
        where Sal < (Select max(Sal)
                     from emp) ;
```

Subquery Case-4

Where ever we want to fetch subordinate name or manager Name then we should go for Subquery case-4

Emp

EmpNO	EName	MGr
1	A	3
2	B	1
3	C	2

from clause o/p

WAQTD the manager Name for 'B'

```
Select ename
(1) from emp
(2) where EmpNo = (Select mgr
                   (1) from emp
                   (2) where ename = 'B');
```

Emp = 1 :

1 = 1 → T ✓

2 = 1 → F ✗

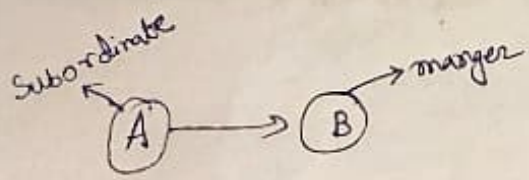
3 = 1 → F → ✗

A = B → F ✗

B = B → T ✓

C = B → F → ✗

Question	IQ	OQ
Manager	MGR	Emp NO
Subordinate	Emp NO	Mgr.



Draw backs of Subquery:-

1. Output is valid but it is not clear.
2. from Case 2 We cannot get fetch the data from multiple table at the same time.
3. from Case-3 Length of the query is more then the perfomace is very poor.

4) from case - 4

We cannot fetch subordinate name and manager name at a same time.

⊛ (Joins) ⊛

VI ⊛⊛ (Joins) ⊛

When to go for Joins:-

1. To overcome the drawbacks of subquery we go for joins
2. Whenever we want to fetch the data from multiple tables then we go for joins.

What is Joins?

It is a special possibility Select Statement use to fetch Data from multiple tables at the same time.
By using joins we can fetch the data from single table also.

There are 5 types of Joins

1) Cross Join / Cartesian joins
2. Inner join / equi join / Natural join
3. Non-equi join
4. Self-join
5. Outer-join
   * left outer join
   * right outer join
   * full outer join

1. Cross Joins / Cartesian joins

drinks

did	dname	Price
1	old monk	560
2	OC	100
3	Magic Moments	450

bars

bid	bname	did
1	andu bar	1
2	aunty bar	2
3	Piles bar	3

oracle syntax:-

Select */col-name
from T_1, T_2;

ANSI Syntax

Select *|col-name
from T_1 cross join T_2;

WAQTD the drinks name & bar name for all the drinks

② Select dname, bname
① from drinks, bars;

Price	dname	did	did	bname	bid
560	Old monk	1	1	andy bar	✓ 1
560	Old monk	1	2	aunty bar	× 2
560	Old monk	1	3	Piles bar	× 3
100	OC	2	1	andu bar	✗ 1
100	OC	2	2	aunty bar	✓ 2
100	OC	2	3	Piles bar	× 3
450	Magic Moments	3	1	andu bar	× 1
450	Magic Moments	3	2	aunty bar	× 2
450	Magic Moments	3	3	Piles bar	✓ 3

Dname	bname
Old monk	
Old monk	
Old monk	
OC	
OC	
OC	
MM	
MM	
MM	

Cross Joins

It is a cross product of two or muliple tables and it will follow the Cartesian principle which has one table of one record is ~~matching~~ merging with each and every record of another table. is called as.

In above example it will provide 9 records are output. In 9 records, 6 records are invalid and 3 records are valid records.

Valid records

One table of one records is ~~having~~ matching pair in another table.

Invalid

One table of one records is not matching pair in another table.

→ In cartesian join it will provide more invalid records compare to valid records, This is the main drawback of cartesian joins.

To overcome this, we go for inner joins

Inner Joins

When to go for inner joins:-

To overcome the drawback of Cartesian joins or cross joins, then we should go for inner joins.

What is inner joins:-

Joining the two tables by common column is called as inner joins.

Syntax:-

Oracle syntax

```
Select * / col-name
  from T1 , T2
where T1.common column = T2 common column;
```

joining condition.

ANSI syntax

```
Select * / col-name
  from T1 (Inner) Join T2
  ON T1.cc = T2.cc;
```

Drinks

did	dname	Price
1	Old monk	560
2	OC	100
3	magic moments	450

Bars

bid	bname	did
1	andu bar	1
2	auntu bar	2
3	Piles bar	3

WAQTD the drinks name & bars name for all the drinks.

```
Select dname, b name
  from drinks, bars
  where drinks.did = bars.did;
```

dname	bname
Old monk	Andu bar
OC	Aunto bar
mm	Piles bar.

inner join

1. WAQTD Emp names and Dept name for all the employees

oracle

```
Select Ename, Dname
from emp, Dept
Where emp. Deptno. = Dept. DeptNO;
```

2. WAQTD. Dept name & Location for the employee who are earning 4 digit Salary.

```
Select Dept No., Loc
from Dept, emp
Where Dept.DeptNO = Emp. Deptno
And Sal Like '____';
```

3. WAQTD the employee details who are working in Chicago.

```
select *
from Emp, Dept
where emp. Dept no = Dept. Dept no
and Loc = 'Chicago';
```

4. WAQTD the Department name, Location and employee name for Salesman.

```
Select Dname, Loc, Ename
from emp, dept.
where emp. deptno = Dept. dept no
and Job = 'Salesman';
```

5. WAQTD the Department name for Smith and Allen

```
Select Dname
from Dept, emp
where emp. dept no = Dept. dept no
and Ename = 'Smith' or Ename = 'Allen';
```

⑥ WAQTD the employee name along with department name if employees are not working a salsman

Select ename, ~~deptno~~ dname
from EMP, DEPT
where Emp. Deptno = DEPT. Dept no
and Ename not In 'SALESMAN'

7. WAQTD the Dept. details for those who got hire in the year of 81.

Select *
from EMP, DEPT
where Emp. Detno = Dept. dept no
And hiredate like '%81';

8. WAQTD the Job employees department no, and location if the employees are report into 7698.

Select

9. WAQTD the employee details department name and location, who are getting Salary from 1000 to 3000 and who are not getting Commission.

10. WAQTD the employee details and location for the employee who are working dept no 10 and 20 and who are no getting Commission and who got hired on in the year of 81 and who are getting Salary from 1000 to 4000 along with display the department name.

Select Emp.*, D Name, LOC
From EMP, DEPT
Where EMP. DEPT = DEPT. DEPTNO
AND EMP. DEPTNO IN (10, 20)

Note:- When ever if we are writting common column a condition we should use ~~Table name~~
Table name . common column.

Spool Drivename :/filename.txt → D.

Select Emp.*, D Name, LOC
From EMP, DEPT
Where EMP. DEPT = DEPT. DEPT No
AND EMP. DEPTNO IN (10, 20)

Note:- When ever if we are writing common column as a condition we should use ~~Table name~~.
Table name . common column.

Spool Drivename :/filename.txt → D.

Q) WAQTD the ename & dname for all the employees & along with display the d name where NO employee's are working?

Outer Join

when to go for outer join?

whenever we want to fetch valid records from both the tables along with invalid record from particular table.

what is outer join:-

Joining the two tables by using common column is called as outer joins.

valid record (left & right) + invalid records (left (right (both right & left))

↓

inner join

⊕ There are 3 types of outer join

1) **left outer Join** :- It will provides valid record from both the tables along with invalid records from Left table.

Right outer Join :- It will provide valid record from both the tables, along with invalid record from Right Table it is

Full outer Join :- It will provide valid & Invalid record from both the table.

Left Outer Join

ORACLE

Select * / col-name
from T_1, T_2
where $T_1.cc = T_2.cc(+)$;

Left Right

ANSI

Select * / col-name
from T_1 left [outer] Join T_2
ON $T_1.cc = T_2.cc$;

Right Outer Join

select * / col-name
from T_1, T_2
where $T_1.cc(+) = T_2.cc$;

select * / col-na
from T_1 right [outer] Joins
ON $T_1.cc = T_2.cc$;

Full outer Join

L.O.J
Union
R.O.J;

Select * / col-name
from T_1 Full [outer] Join T_2
ON $T_1.cc = T_2.cc$;

---

1. WAQTD the employee names and Dept. Names for all the employess along with display the Dep. Name where are no employess are working.

Right.

Select ENAME, DNAME
from Ename, Dname
where EMP. Dept NO(+) = Dept. Dept NO;

2. the employee Name and Dept. Names for all the employees along with display the employee name who are not working in any dept.

5. WAQTD. the employee names and Department Names along with who are not working in any department and display the department name where no employees are working.

Non-equi Join

∴ If there are no relationship between two tables then we go for NO-equi Join.

* relation · sm

Joining the two tables without using common column.

if we want to apply non-equi Join we should satisfy two condition.

(i) relationship should not present

(ii) Relevent data should be present.

1> Col-name should be different
2> Data type should be same
3. Size should be same.

→ This type of columns is called as Re

Between Emp and SALGRADE Table relevant data is Sal, loSal and hiSal.

EMP

Number	VARCHAR (15)	VARCHAR (15)	Number (4)	Date	Number (4)	Number (4)	Number (2)
~~Emp No.~~	~~Ename~~	~~Job~~	~~Mgr~~	~~Hire date~~	SAL	~~Comm~~	~~Dep No~~

Grade	loSal	hiSal
Number (1)	Number (4)	Number (4)

Self Join

TO, overcome the draw backs of Sub Query case-4 we go for self-join

Self-Join → Joining itself is called as self-join.

1. **Case-1** :

Joining the Same table by using different Column name is called as self-join.

Syntax $T_1.C_1 = T_1.C_2$

WAQTD the all sub-ordinate ~~ing~~ name and manager name for all employees.

```
Select A.ename as Subordinate , B.ename as manager
from Emp A, Emp B
where A.mgr = B.Emp No;
```

Case-2

Joining the Same table by using same Column-name is called as Self-join Case-2

Syntax - $\boxed{T_1.C_1 = T_1.C_1}$

WAQTD the employee names who are getting Same Salary?

```
Select A: ENAME , B. ENAME
from EMP A; EMP B
Where A.SAL = B.SAL
And A. Ename ! = B. ENAME ;
```

```
Select A. Ename
f
```

① What is difference between Unique and Distinct

② " " " " Join and Union

③ ⊛⊛⊛ How to fetch the duplicate Salary values.

```
Select SAL
from emp
Group by SAL
Having count(*) > 1 ;
```

Distinct	Unique
1) It is clause	1) It is a constrient
2. Whenever we want to display the unique values then we should go for Distinct clause.	2. Whenever we want to store the unique values, then we should go for unique constrient.
3. Distinct clause is used in Select statement	3. Unique is used in Create statement.

Joins	Union
1. Whenever we want to joins the two tables we use joins.	Whenever we want to joins two queries.
2. Joins is a concept	2. Union is a operator.

Normalization

The decomposition of larger or complex tables into several simples tables is called as normalization.

It was introduced by E.F. Codd. in the year of 1970.
It is also called as E.F Codd rules.

To check in whether the table is complex or simple we will of apply normal forms.

There are 7 normal forms.

1 NF
2 NF
3 NF
3.5 NF (BCF) → Boyce Codd
4 NF
5 NF
6 NF

1NF :-

* each & Every cell should contain single values.
* Col name should be different.
* duplicate record should not be present.

Pname	Brand	Colour 1	Colour 2	Size

Clothes

Pname	Brands	Colour	Size.
Kalia	Rupa	Pink	S
Sutta	Jockey	Black / Transparent	XS
420	Lux cozi	Rain bow	M
Dalal	Dollar	Black	L
motu	Chottu.	white/green	XL

↓
Complex

1) Why Inner Query will execute?

→ Subqueries follows the ~~bod~~ BODMAS

(Bracket of Division, Multipulation, ~~Abstraction~~ Addition and Subtraction)

2NF :-

- It should follow 1NF
- Each and every table should contain PK
- Functional Dependency should be present ✓
- Partial Dependency should not be present ✗

Std 1

Sid	Sname

Std 2

Sid	Per

Students

Sid (PK)	Sname (Not PK)	Per (Not PK)
1	Sutta	90%
2	Kalia	69%
3	420	99
4	Mottu	30%
5	Andu	0%

Complex

Functional Dependency :-

All non-primary key column is depending on primary key column. This type of depency is called as functional Dependency.

Partical Dependency :-

Non-primary key column is depending another Non- " " " . This type of dependency is called as partical dependency.

Transitive functional Dependency:-

a=b
b=c
c=a

3Nf:-

* It should follow 2Nf
* Transitive functional dependency should not be present.

Assume 3 columns are there, c_1, c_2 & c_3.

c_1 is depending on c_2

c_2 is " " c_3

Then indirectly c_3 column is depending on c_1.

Then this indirectly dependency is called Transitive functional Dependency.

16/2/24

Interview

Co-related subquery

When to go for co-re

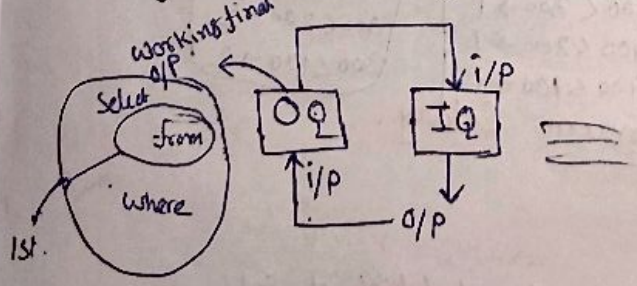
⇒ To overcome the drawback of subquery case-3 we go for co-related subquery.

At what

→ It is a advanced subquery used to fetch n minimum or nth maximum value by reducing the query length.

→ It is a combination of subquery and joins.

→ Working mechanism of co-related subquery:



1) 1st outer query will execute partially. (only from clause)
2. The output of from clause is given to input for the inner query.

3. Inner query will execute all the ~~query~~ statement it will provi
4. the o/p of innerquery is given to input for the outerquery
5. Now outer query will execute 2nd time.
while executing 2nd time it will execute all the state ment & it will provid final output.

6. OQ is dependendin on IQ and IQ also dependent on OQ.

Write a query to display the 3rd maximum Salary.

```
(6) - Select distinct Sal
(1) - from Emp A
(5) - where 2 = (4) (Select count(distinct Sal)
      (2) - from emp B
      (3) - where A.Sal < B.Sal);
```

Emp-A

Ename	Sal
Sutta	500
Dalal	400
Motu	300
Aunty	200
Chotu	100

Emp-B

Ename	Sal
Sutta	500
Dalal	400
Motu	300
Aunty	200
Chotu	100

500 < 500 → 0	400 < 500 → 1	300 < 500 → 1
500 < 400 → 0	400 < 400 → 1	300 < 400 → 2
500 < 300 → 0	400 < 300 → 1	300 < 300 → 2
500 < 200 → 0	400 < 200 → 1	300 < 200 → 2
500 < 100 → 0	400 < 100 → 1	300 < 100 → 2

WAQTD the 999th maximum Salary.

```
Select distinct Sal
from Emp A
where 998 = (Select Count (distinct Sal)
             from Emp B
             where A.Sal < B.Sal);
```

< → for max

> → for min

1. WAQTD the 9th max Salary.

```
Select distinct sal
  from Emp A
  where 8 = ( select count (distinct sal)
              from emp B
              where A.Sal < B.Sal);
```

2. WAQTD the 6th recent hiredata.

```
Select distinct Hire date
  from Emp A
  where 5 = (Select count (distinct Hiredate)
             from Emp B
             where A. Hiredate < B. Hiredate);
```

3. WAQTD the 7th lowest Salary.

```
Select distinct Sal.
  from Emp A
  where 6 = (Select count (distinct Sal)
             from emp B
             where A. Sal > B. Sal);
```

4. WAQTD the 8th recent hiredate

```
Select distinct Hire date
  from Emp A
  where 7 = ( Select count(Hire date)
              from emp B
              where A. Hiredate < B. Hire date);
```

5. WAQTD the 10th recent hiredate.

```
Select distinct Hiredate
  from Emp A
  where 9 = ( Select count (Hiredate)
              from Emp B
              where A. Hiredate < B. Hiredate);
```

6) WAQTD the 2nd, 3rd, 4th, 5th, and 6th max Salary.

```
Select distinct Sal
from Emp A
where ( Select count (distinct sal)
        from Emp B
        where A.Sal < B.Sal) In (1,2,3,4,5);
```

Sayantan

Views

Simple view

```
Create view viewname
as
( Select */Col name
  from Table Name
  where condition );
```

Complex view

```
Create view Viewname
as
Select */Col-name
from T1 , T2
where T1.CC = T2.CC;
```

Single Row f

views:- It looks like a table but is not like a table.
It is a table like structure by selecting required number of columns and required number of rows from the table.

There are 2 types of views:

1. Simple view and complex view.

Simple v: → If view is created by using single table is called as Simple view.

Complex view:- If view is created by using multiple tables it is called as complex view:

Complex 1.

```
Create view mottu
as
Select ename, dname
from emp, dept
Where emp.deptno = dept.dept no;
```

```
Select Ename
from Emp
Where Length (ENAME) = 5;
```

SUB - Languages / Statements / Commands

DDL (Data Definition Language):-

1) **Create**: Create statement is used to create a table in the database.

Syntax: create table tablename
(colname 1 datatype (size) constraint (s);
colname 2 datatype (size) constraint (s),
. . . . .
. . . . .
colname N datatype (size) constraint (s));

Syntax for adding foreign key column:

Column name datatype (size),
foreign key (fk colname) references parentable (Ph colname);

To copy the data from one table to another:

Syntax: Create table newtablename
as
Select *
from tablename;

2) **ALTER**: Alter statement is used to modify the table with respect to columns.

Adding A Column:

alter table tablename
add colname data-type constraint (s);

Rename A Column:-

Alter table tablename
rename column old colname to new colname;

Delete A column:

alter table tablename
drop column colname;

3) **Rename**: Rename statement is used to rename a table

Syntax: rename old tablename to new tablename;

4. **Truncate:** Truncate is used to remove all records of a table at one shot. ~~Truncate~~ Truncate makes a table empty. Truncate will not remove the table from database instead it removes records from table.

Syntax: truncate table tablename;

5. **Drop:** Drop statement is used to remove a table from database.

Syntax: drop table tablename;

* If drop is applied on table, the table while removed from database and gets stored in recyclebin to ~~remove~~ restore the table from recyclebin to database, we use flashback.

Syntax: flashback table tablename to before drop;

* To delete the table from the recyclebin, →
  purge table tablename;

* To remove the table permanently from the database without storing in BIN,
  drop table tablename; purge

* To remove the table permanently from the database without storing.

DML (Data Manipulation Language)

1) **Insert:-** Insert statement is used to add records into the table.

Syntax:- Insert into tableName Values ($V_1, V_2, V_3 \ldots$);

or insert into table Name (colname 1, colname 2, ...) → values (V_1, V_2);

2. **Update:-** Update is used to update existing data in a table.

Syntax: Update table Name
Set Colname = Value
where Condition (S);

update dance
set mrp = 300

or Update table name
Set colname = ~~Value~~ & VALUE, Col name = Value
where condition(s);

3> **Delete** :- Delete is used to remove particular records from a table.

Syntax :- delete from table name
where Condition (s);

delete ~~do~~ from dept
where did = 2;

(*) All the operations performed using DDL commands are permanent.

(*) All the operations performed using DML Commands are not permanent.

TCL (Transaction Control Language)

DML operations are not permanent, it can be saved permanently using TCL.

1> **Roll back** : Roll back statement is used to undo/delete all the DML operations.

syntax :- rollback;

2> **COMMIT** : Commit statement is used to save all the DML operations permanently.

syntax : Commit

3> **SAVE POINT** : Savepoint is a name given for a dml operation. it is used to fetch a particular dml operation.

Syntax: Savepoint Savepoint name;

DCL (DATA CONTROL LANGUAGE) :-

Q) what ~~it~~ are deference between Truncate, Drop, Delete.

Truncate	Drop	Delete
whenever we want to delete in one shot then we should go for truncate.	whenever we want to delete entire table from the data base then we go for Drop.	whenever we want to delete particular record from the table then we should go for delete
It is belongs to ~~ett~~ DDL Statement.	It is belongs to DDL Statement.	It belongs to DML
Truncate is a permanent action	Drop is also permanent action	Delete is non-permanent action
In Truncate we will not use where condition	In Drop we will not use ~~a~~ where condition	In Delete we will use where condition.
In Tr.. execution time is less.	In Drop execution time is less.	execution time is more.