SQL

Structured Query Language

Page No:
Date.

Data :- Data is a Raw-Fact which describe properties / Attributes of an object or entity.

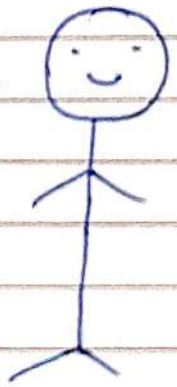
Ex.

object/entity → properties/Attributes

His name is Harshad.

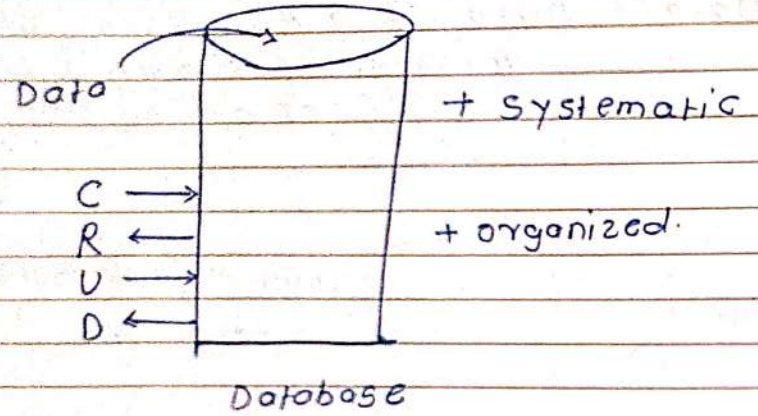
Raw Fact

Ex.



Name :- Harshad
Gender :- ~~25~~ Male
DoB :- 25/5/03
mob :- 1234567890
Email :- abc@gmail.com
age :- 21
nationality :- Indian
Adhar :- xxxxxxxx4876
pan :- xxxx
A/c :- xxxx.

* **Database :-** It is place where Datastore in systematics and organised manner.



This operation which we can perfrom on the Database and those are

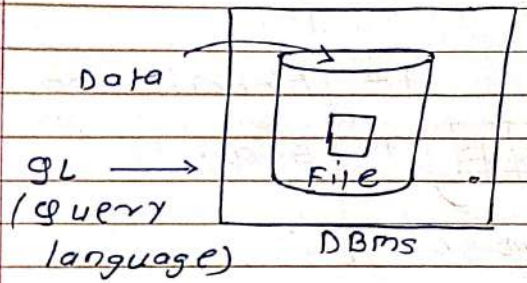
Create / Insert
Read / Retrives
Update / modify
Delete / DRop

This operation universal Refered as CRUD operation.

* DBMS:- Database Managment System

- It is a software which is used to maintain and mangge the database

- In DBMS data stored in the form of Files.



- It provides two factors :
  1] Authorization
  2] security.

+ypes of DBms:-

NDBms - Network

HDBms - Hierachical

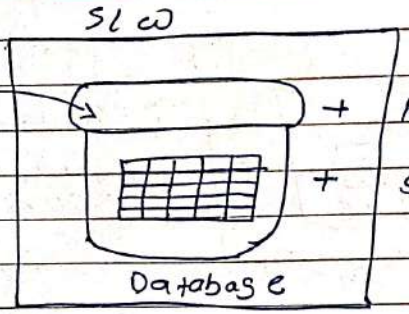
RDBms - Relational

OODBms - object oriented.

RDBms: Relational Database management system.

- It is a type of DBms software

- In RDBms Data stores in the form of tables.



It provides two factor
① Authorization
② security

- we use SQL to communicate with RDBMs.

Relational model:-
- It is designed by a data Scientist called as E.F Codd. (Edger Frank Codd)
- In Relational model data stores in the form of table.
- any DBMS Follows relational model it will became RDBMS.

DBMS Follows → Relational model → RDBMS

Any DBMS Follows Rules of E.F.Codd it will became R.DBMS Follow

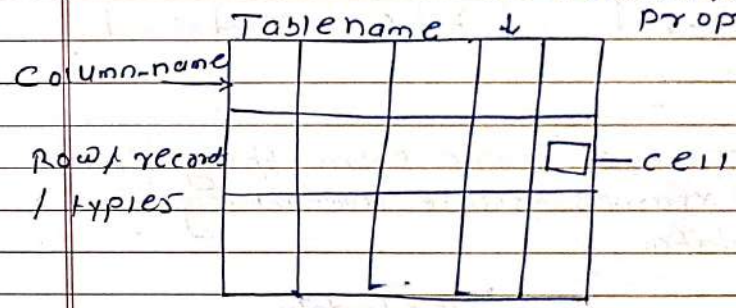
DBMS → Rules of E.F Codd → RDBMS.

* Terminology:-

Table - The logical orginization of data which consits rows and columns it is known as table.

OR

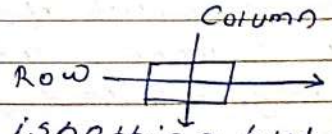
The combination of rows & column is known as table.



Column:- It is also refered as proporties, Attributes, & field

Row:- It is also refered as records & tapies.

Cell:- It is the smallest unit of table or intersection between rows and column it will generates



entity: it is nothing but anything which

* Rules of E.F Codd

1] The data entered into a cell must be a single valued data to avoid data loss.

e.g.

Student

SID	SName	PH-NO
1	A	123
2	B	124, 125 → 129
3	C	126

SID	SName	PH-NO	Al-PHNO
1	A	123	-
2	B	124	125 → 129
3	C	126	-

② In RDBMS we store everything in the form of table including metadata.

Student.

SID	SName	Photo
1	SALLU	
2	AISH	
3	KAT	

metadata.

ImgName :- Img-005
Device :- 05:02PM Samsung
size :- 2mb
Time :- 05:02pm
Loc :- Dange chock.

MataData :- Data about the Data or Details about the data is known as Metadata. It is autogenerated.

- Metadata store in meta Table.

Img-Name	Time	Device	Size	location
Img 005.	05:20pm.	Samsung	2mb	dangechock

③ According to E.F Codd we can store the data in multiple tables if needed we can establish connection between the tables by using key Attributes.

K-A

Emp

SID	SName	DNO.
1	A	20
2	B	30
3	C	10

Dept

DNO	DName	...
10	D_1	L_1
20	D_2	L_2
30	D_3	L_3

4. The data entered into a table can be verified in 2 steps
① By assigning data type
② By assigning constraints.

Note:-
① Data types are manditory and constraints are optional

② SQL is not a case sensitive language.

* Statements :-

It help us to perform Crud operation on the database.

* Overview of SQL statements.

There are 5 statement in SQL

① DDL (Data Definition language)

② DML (Data manipulation language)

③ TCL (Transaction Control language)

④ DCL (Data control language)

⑤ DQL (Data query language)

* DQL (Data query language)

It is used to retrive the Data From the Database

- Their are 4 statement in DQL

① Select
② projection
③ selection
④ joins.

① Select :- This statement is used to retrive the Data From the ~~Data~~ Tables & display It.

② projection :- It is used to retrive the data From the table by selecting only Column.

- In projection the data present under the column will be selected by default

③ Selection :- It is used to retrive the data from the table by selecting rows as well as Column.

④ joins :- It is used to retrive the data From the multipal tables simultaneously.

* projection:

It is used to retrive the data From the table by selecting only column.

Syntax:

```
SELECT  */[DISTIC] Column Name/
                Expression [ALIS]
  FROM Table_Name;
```

order of execution:

1. FROM
2. SELECT

write A query to disply student Name

Student

SID	SName	Branch	PER
1	Gopi	CS	55
2	Roshi	IT	80
3	KOKILA	CIVIL	75
4	JIGAR	mech	85
5	AMEM	mech	70
6	KINJAL	EC	80

Database

O/P From clause

② SELECT SName
① FROM STUDENT;

O/P OF SELECT CLAUSE

SName
Gopi
Roshi
KOKILA
JIGAR
AMEM
KINJAL

Statement : CLAUSE

Q. write a query to display PER of the students

⇒

```
SELECT PER
  FROM student;
```

write a query to display SID and SName of the student.

⇒

```
SELECT SID, SName
  FROM student;
```

write a query to display PER & branch of the student <==

⇒

```
SELECT PER, branch
  FROM student;
```

PER	branch
55	CS
80	IT
75	CIVIL <==
85	mech
70	mech
80	EC

write a query to display (with AGID) student id, branch, student Name, PER

=> Select SID, branch, SName, PER
FRom student;

WAQTD All the details of the student table.

=> Select * FRom student;

Neco table:

WAQTD EmpName, Salary, hierdate & Dept No emp.

=> Select EName, SAL, HIAEDate, DEPTNO
FRom Emp;

WAQTD EmpID, EmpName, Salary, and along with their desicarination.

=> Select EmpNO, EName, SAL, job
FRom Emp;

Note:- i] FRom Calvse start the execution
ii) FRom Calvse coe can pass table name as arrguiment (input).
iii] The job of FRom Calvse is goes to the data base and seaech for table and put the table into execution.

iv] Select Calvse execute after the execution of FRom Calvse.
v] For select calvse coe can pass */columnName/ expression as arguiment (input).
vi] The job of select calvse is it will goes to the table which is under execution and select the column which is metione and display it.
vii] by this coe can say select calvse is responsiable for the "result table"

Semicolone(;):- It is nothing but end of the qeruy or statement.

Asterisk (*):- It will select all the columns of the table.

* WAQTD All the tables present in the users.

=> Select * FRom TAB;

WAQTD All the details of the Demp table.

=> Select *
FRom DepT;

Column Name DEPT

DEPTNO DName Loc

WAQTD all details of the empty table.

=> Select *
FROM Emp;

EmpNO
EName
Job
MGR
Hiredate
SAL
Comm
DEPTNO.

WAQTD Empid, EName, Salary and along with annual sarly of employ.

=> Select EmpNO, SAL, SAL*12
FROM Emp;

Expression:

- A stolement which gives reslut is known as expression.
OR
- The combination of operand and operators is known as expression

operator
↓
SAL * 12
↓ ↓
operand operand

- There are two types in expression
i] operand
ii] operator (+, -, *, /)

- operand consists two types.
i] Column name
ii] Literals (Direct value)

- Literal consits 3 types
i] Number Literal
ii] Charecter Literal
iii) Date Literal

- Date literal :- Date Format which is given oracal
  DD-mon-yy
  DD-mon-yyyy

Note :- Character Literal and date Literal encoise within single quote (' ')

WAQTD Ename, Hiredate, descination, SAL & half term of employ salary.

→ Select EName, HIRDate, job, SAL, SAL*6
FRom emp;

WAQTD Empoly id, EName, Salary & along with 5% hike in salary

⇒ Select EmpNO, EName, SAL, SAL+(SAL*0.05)
FRom emp;

WAQTD EName, SAL, and 25% dedication in the salary

⇒ Select EName, SAL, SAL - 25/100*SAL
FRom emp;

WAQT
WAQTD EName, HireDate, SAL & 50 Rs bones in the anual sal

⇒ Select EName, HireDate, SAL, SAL*12+50
FRom emp;

WAQTD all the details of employ along with anual salary

⇒ Select *, SAL*12 नाम आगे लिखेंगे तो error देगा
FRom emp;
TableName.columnname

Select Emp.*, SAL*12
FRom emp;

* ALIAS :-

It is an alternative name which is given to column or expression in the result table.

- with or without 'As' keyword we can assign Alias.
- alias name should be one word or it should be enclosed with double qoles (" ") or we can make it signal word by using underscore.

WAQTD DIFFERENT SALARY

→

DISTINCT :-

Distinct it is calus used to removed duplicate or repeted value from the result table

- We can pass distinct calus in the Select calus as an first arrguiment.
- For distinct calus we can pass column name or expression of an arrgument.
- For distinct calus we can pass multiple arrgument But It will remove duplicates in combination.

WAQTD DIFFERENT salary.

→

```
Select Distinct SAL
  From Emp;
```

O/p of FROM clause.

Eid	EName	SAL	DNO
1	MAHENDRA	2500	30
2	KATAPPA	4000	20
3	Shivgami	2500	30
4	BADILAL	5000	10
5	Devsena	3000	20
6	KALKEY	5000	30

o/p of select clause.

SAL
2500
4000
2300
5000
3000
5000

result Table

o/p of Distinct Clause

SAL
2500
4000
5000
3000

WAQTD Different SAL & DNO.

→

```
Select Distinct SAL,DNO
  From Emp;
```

o/p of select clause.

SAL	DNO
2500	30
4000	20
2500	30
5000	10
3000	20
5000	30

o/p of DISTINCT Clause

SAL	DNO
2500	30
4000	20
5000	10
3000	20
5000	30

WAQTD the different designation that are present in emp table.

→

```
Select Distinct job as designation
  From emp;
```

WAQTD name of employee who works in dept 20:

→

```
Select ename
  From emp
  where deptno=20;
```

- Selection:-
  - It is used to retrieve the data from the table by selecting row as well as column.
- where clause:- It is used to filter the condition.
  - It executes row by row.
  - It executes after the execution of From clause.
  - In where clause we can write filter condition which returns boolean values (T/F)
  - In where clause we can pass multiple condition by using logical operators.

Syntax:-

```
Select * /[distinct] Columnname/expression[ALIA
  From table_name where <filter-condition>;
```

- order of execution:-
  1. From
  2. where → Row by Row
  3. Select.

WAQTD name of employee who work's in deptno 20.

→ Select ename
From emp
where deptno = 20;

o/p of From Clause: Emp		
Eid	Ename	dno
1	Stark	20
2	Steve	30
3	Bruce	20
4	Wanda	10
5	Loki	10
6	Gamora	30

o/p where Clause

Eid	Ename	dno	
1	Stark	20	= 20 (T)
2	Steve	30	= 20 (F)
3	Bruce	20	= 20 (T)
4	Wanda	10	= 20 (F)
5	Loki	10	= 20 (F)
6	Gamora	30	= 20 (F)

o/p of Select Clause

Ename
Stark
Bruce

- operators:-
1] Arithmetic operators (+, -, *, /)
2] Concatination operators (||)
3] Logical operators (And, or, not)
4] Comparison operator (==, !=, <>)
5] Relational operator (<, >, >=, <=)
6] Special operator (In, not in, Between, not Between, Like, Not Like, IS, IS NOT)
7] Subquery operator (All, Any, exist, not exist)

- Logical operators:-

1) AND :- Truth table for AND

Condn1	Condn2	Result
T	T	T
T	F	F
F	T	F
F	F	F

Note:- in case of AND operator we get the o/p if all the condn are satisfied/true.

Q. WAQTD name, job & salary of the employees who working as salesman and earn salary greater than 1500.

→ Select ename, job, sal
From emp
where sal > 1500 And
job = 'salesman';

OR Truth table for OR:

Condn1	Condn2	Result
T	T	T
T	F	T
F	T	T
F	F	F

Note: In case of OR operator we get the output if any one of the condition is satisfied / true.

Q WAQTD name & job of the employees who are working as salesman or manager.

→ Select Ename, job
From emp
where job='salesman' OR job='manger';

Q WAQTD details of employees who are salesman or they should be working in deptno 30

→ Select *
From emp
where job='salesman' ~~And~~ OR
deptno = 30;

*NOT :- Truth table For NOT

Condn	Result
T	F
F	T

Note:- NOT operator is used to revers the result.

Q WAQTD name & job of the employees except For manager

→ Select ename, job
From emp
where ~~job~~ NOT job='manager';

(OR)

Select ename, job
From emp
where job != 'manager';

H.W.

1] WAQTD name, job & deptno of employees who are working deptno 20 & 10.

→ Select ename, job,
From emp
where deptno = 20 ~~AND~~ OR deptno = 10;

2] WAQTD the employees who are hired on 28-sep-81 & 3-dec-81

→ Select * From emp
where hiredate = '28-sep-81' ~~And~~ OR
hiredate = '3-dec-81';

3] WAQTD name & job of the employess who are working working as clerk in deptno=10

→ Select ename, job From emp
where job = 'clerk' And
deptno = 10;

4] WAQTD name, job & annual sal of employees who's annual salary is greater than 13000;

→ Select ename, job, sal*12 as annualsalary
From emp
where sal*12 > 13000;

5] WAQTD details of the employees who are working in deptno=10 or 20 and works as manager or salesman & earns salary lesser than 2000

→ Select * From emp
where (deptno = 10 OR 20) AND
(job='manager' OR job='salesman') And
sal < 2000;

6] WAQTD details of employe who hired in 81

→ Select *
From emp
where hiredate > '31-dec-81';

* And :- It is used whenever we have to satisfied both the condition.
OR
It returns true if it satisfied both the condition.
- It can be written in between the condition.

* OR :- It is used whenever we have to satisfied any one of the condition.
OR
It returns true if it satisfied any one of the condition.
- It can be written in between the condition.

* Not :- It is used whenever we want to invers or reverse the condition

WAQTD details of the employ if the employs works as manager deptNo 20 or 30.

→ Select *
From emp
where job = 'manager' And
DepNO = '20' OR '30';

→ Select *
From emp
where Job = 'manager' And

WAQTD all the details of employs if the employess hired after 81 but before 86.

→ Select *
From emp
where Hiredate > '31-dec-81' And
Hiredate < '01-Jan-86';

WAQTD all the details of employees except the employees who works dep No 20.

→ Select *
From emp
where DepNO != 20;

* Concatination operator :-
It is used to join the given strings.
Symbol = ||
Syntax :- String1 || String2
Ex.
Select 'MR' || Ename
From emp;

WAQTD details of employees give the employees hired after 80 but before 87 and works as manager or salesman or clerk and the employee earn more than 1250 but

except the employee who works as analysis

→ Select *
From emp
where Hire date > '1-Jan-80' And
Hiredate < '31-dec-87' And
(job = 'manager' OR Salseman OR (leark)
And
Sal > 1250 ABd Sal < 500 And
(Depno = '20' OR 30)
job Not Analyst;

→ Select *
FRom emp
where Hire date > '31-dec-80' And
Hire date < '01-Jan-87' And
(job = 'manager' OR job = 'Salesman' OR
job = 'CLERK') And
Sal > 1250 And Sal < 5000 And
(Deptno = 20 BR 30) And
job Not Analyst;

Special operator:-

1 IN:- It is a special operator which accept ~~except~~ multiple values at RHS.
- It behavious similary to equal operator
- It returns true if it will satsified any one of the value RHS

Syntax:-
Column Name/Expression In (v1, v2 v3 ... vn)

Example
Depno In (20, 30)

WAQTD details of employess if the employees works as manager or salseman OR analysist. and the employess should works in dep 20;

→ Select *
FRom emp
where job In ('manager', 'salseman', 'analysist')
And
~~Depno In (20, 30);~~
Depno = 20;

2 Not In :- It is similar to In operator Instead of selecting it will reject the values.

Syntax:- Column Name/Expression Not In (v1, v2, v3

WAQTD details of employees expect the employees who works in depno 20, or 30.

→ Select *
From emp
where Depno Not in (20,30);

3] Between :- It is used whenever we have range of values.
- It includes the given range.
- we cannot interchange the range.

Syntax:-
Column-name expression Between starting value And Ending value

Ex:-
SAL Between 1000 And 3000

Q WAQTD Eid, Ename and Hiredate of employees if the employees earn in the range of 1250 to 3000

→
Select EMPNO, Ename, Hiredate
From emp
where sal between 1250 And 3000;

Q WAQTD details of employees if the employees hired in 1981 year

→ Select *
From emp
where Hiredate between '01-Jan-81' And '31-dec-81';

Q WAQTD Eid, Ename, Hiredate and depno of employees if the employees earn more than 1250 but less 3000 and Hire after 80 but before 84.

→ Select Empno, ename, Hiredate, depno
From emp
where sal between 1250 And 2999 And Hiredate between '01-Jan-81' And '31-dec-83';

4] Not between :- It is similar to between operator instead of selecting it will reject the range.

Syntax:
Column-name expression NOT between starting value And Ending value.

WAQTD details of employees except the employees who Hire in the year 81

→ Select *
From emp
where Hiredate Not between '01-Jan-81' And '31-dec-81';

5] IS :- It is special operator which use compare with NULL.

Syntax :-

Column-Name/expression IS NULL

WAQTD details of employees whos Commiation is NULL.

→ Select *
From emp
where comm IS NULL;

6] is Not :- It is similler to is operator but it will compare with NOT NULL.

Syntax :-

Column-name/expression IS NOT NULL;

WAQTD all the details of employees if the employess commitation is not empty.

→ Select *
From emp
where comm is not NULL;

WAQTD all the details of employees if the employess earning more than 950 but less than 5000 and the employees works in deptment 20 or 30 and the employees should hired after 80 but before 88 expect the employees who hired in the year 82 and the employees should works as analyst or manager or saleman or clerk and expect the employees who is earning exactly 3000 and the employees should earn some Communiation.

→ Select *
From emp
where sal between 951 And 4999 And
Deptno In (20,30) And
Hiredate between '01-Jan-81' And '31-dec-87' And
job In ('analyst', 'manager', 'salesman', 'clerk')
And
sal != 3000 And
comm is Not NULL;

7] Like :- It is used to perform pattern matching.

Syntax :-

ColumnName/expression Like 'pattern to match'

-To achive pattern matching there are two special characters such as
% and _

% :- It can accepet N number of Characters, any number of times or no characters.

Under Score. _ :- It can accepet any charecter but only one charecter.

Ex.

1) Starts with character A -----> 'A%'

2) Ends with character A ----> '%A'

3) Has character A ----> '%A%'

4) Has character A present twice ---> '%A%A%'

5] Has Two Consecutive A ----> '%AA%'

6] Start with A and Ends with K --> 'A%K'

7) second character is A ----> '_A%'

8) Third character is E ----> '__E%'

9) second Last character is E ---> '%E_'

10) Third last character is E ----> '%E__'

11) Starts with K and second last character is N ----> 'K%N_'

8] Not like :- Not like operator is similar to like operator instead of selecting it will rejects the pattern

Syntax :- 'ColumnName/Expression Not Like 'pattern to match'

H.W.

1. WAQTD All the details of the employees who works As A Salesman And working In dept no 30 And earns more than 1500.

→
```
Select *
From emp
where job = 'salesman' And
      deptno = 30 And
      sal > 1500;
```

2. WAQTD Details of All the employees whose Name starts with 'A' OR 'S'.

→
```
Select *
From emp
where ename like 'A%' OR
      ename like 'S%';
```

3. WAQTD details of All employees whose name doesNot start with 'S'

→
```
Select *
From emp
where ename Not like 'S%';
```

4. WAQTD details of all the employess whose commission is NULL And working as a Cleak.

→
```
Select *
From emp
where comm IS NULL And
      job = 'Cleak';
```

5. WAQTD Ename, salary and Annual salary of the employess except those who are working in dept 30.

→
```
Select Ename, sal, sal*12 as Annual_sal
From emp
where deptno IS Not 30;
```

6. WAQTD All the employess who Hired After 31

→
```
Select *
From emp
where Hiredat > '31-dec-81';
```

7. WAQTD Name of the employess whose Name starts with Vowels

→
```
Select ename
From emp
where Ename like 'A%' OR
      Ename like 'E%' OR
      Ename like 'I%' OR
      Ename like 'O%' OR
```

8 WAQTD ALL the details of the employess who Hired after 81 And Before 83.

→
```
Select *
From emp
where Hiredat between '01-Jan-82' And
      '31-dec-82';
```

```
Select *
From emp
where Hiredat like '82%';
```
OR

9 WAQTD details of the employees whose job Has String 'MAN'.

→
```
Select *
From emp
where job = 'salesman';
      job like '%man%';
```

10. WAQTD Ename, Designation and salary of the employess whose Name DeptNo starts with vowels.

→ Select Ename, job, SAL
From emp
where ename like 'A%' And

11. WAQTD Ename, designation, EmpNo and sal of the employees if the employees Hired between 80 to 87 And earns in the Range of 1000 to 3000

→ Select Ename, job, EmpNo, sal
From emp
where Hiredate between '01-Jan-80' And '31-dec-87' And
sal between 1000 And 3000;

1. WAQTD details of the employess whose name starts with character A.

→ Select *
From emp
where ename Like 'A%';

2. WAQTD Name And sal given to the employee who are earning 3 digit salary

→ Select Ename, sal
from emp
where sal between 100 And 999;

3. WAQTD details of employees working in DeptNo 20 and ending with characters In their name.

→ Select *
From emp
where DeptNO = 20 And
Ename like '%S';

4 WAQTD details of the employees Having character E As last second character In Thir job

→ Select *
From emp
where job like '%E_';

5] WAQTD Names of the employess Having Charachter 'A' As the 3rd character In the name.

→ Select Ename
From emp
where ename like '__A%';

6] WAQTD Ename And Hiredate of employees who Hired During the year 82.

→ Select Ename, Hiredate
From emp
where Hiredate between '01-Jan-82' And
'01-dec-82';

7] WAQTD Ename and Hiredate of the employees who were Hired In the month of Feb.

→ Select Ename, Hiredate
From emp
where Hiredate like '%Feb%';

8] WAQTD Ename, Hiredate and sal of the employees who are earning a sal of 8 digit And were Hired In the month of dec.

→ Select Ename, Hiredate, Sal
From emp
where Sal like '___' And
Hiredate like '%dec';

Function:-

It is set of instruction which is used to perform a specific operation

- Function consits two types
  1. Built In Function
  2. user define function.

- Built In Function consits two type
  1. singe Row Function
  2. multi Row Function.

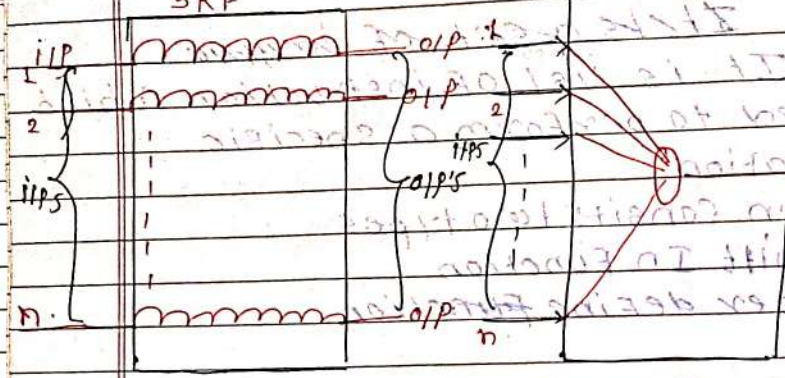
1. Singe Row function

- It execute row by row
- It takes one input and execute and generate one o/p and goes to the next input and goes on.
- If we pass n number of inputs to singe Row function it returns n number of outputs.

2. multi row function

- It is also known as Group function or Agargal Funation
- It executes Group by Group
- It takes all the inputs at ones and agargate (combined) and generates an singe output
- If we pass n number of inputs to multi row function it will return as singe output

SRF



MRF / Group Function

* List of MRF

1. MAX( )
2. MIN( )
3. AVG( )
4. SUM( )
5. Count(*)

* Characteristics of multirow function

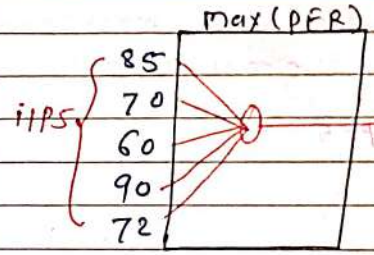
- In multirow function we can pass a single argument.
- along with multirow function's we cannot pass ~~multiple arguments~~ any other column name/ Expression as an argument as select clause
- multirow function ignors NULL.
- In where clause we cannot pass (multirow) function.
- For sum() and avg() we can pass only number as argument
- Count is the only multirow function which accept * as an argument

Student

SID	SName	Branch	PER
1	Jaggu	EC	85
2	Raju	civil	70
3	Bheem	IT	60
4	Tuntun	mech	90
5	KALA	Civil	72

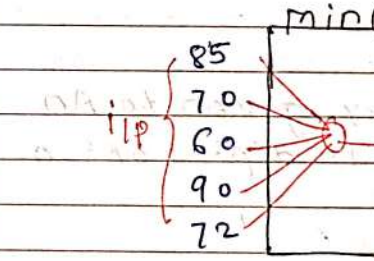
WAQTD maximum percentage of student

→ Select max(PER)
FROM student;



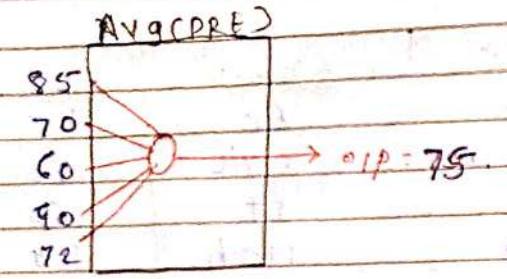
WAQTD minimum percentage of student.

→ Select min(PER)
FROM student;



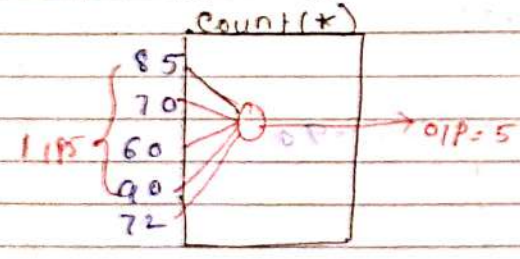
WAQTD avg percentage of the student

→ Select AVG(PER);
FROM student;



WAQTD No. of student present in the student table.

→ Select Count(*)
From student;



WAQTD Number of Employees working in dept 20.

→ Select Count(*)
From emp
where deptno = 20;

WAQTD maximum salary given to An Employee working In dept 20 AS a Clerk.

→ Select max(sal)
From emp
where deptno = 20 And
job = 'Clerk';

3. WAQTD Total SALary given to the employees working as manager In dept 20 or 30.

→ Select sum(sal)
From emp
where job = 'manager' And
Deptno In (20, 30);

4 WAQTD Number of Employees working In dept 10 with A salary greater than 2000.

→ Select count(*)
From emp
where deptno = "10" And
sal > 2000;

* Group by Clause

- Group by clause is used to group or cluster similer kind of data.
- Group by clause executes row by Row and creates Group.
- after the execution of group by clause. A compiler only has groups with it self
- Any clause that get executed after group by clause, executes group by group.

syntax:-

```
Select MRF/Group By Column/Expression
From table_name
where <filter-condition>
group by column/Expression;
```

order of execution	nature of execution.
1] From	It goes to the database search for table name and put the table into execution.
2] where	It will execute Row by Row.
3] Group by	Row by Row
4] Select	A select will execute Group by group.

WAQTD no of employees each different deptno

→

```
Select Count(*), Deptno.
From emp
group by deptno;
```

1) output of From Clause

Enter Emp table

2) output of where Clause

No o/p as there is no where Clause

3) output of Group By clause

EName	Job	Sal	Deptno
SMITH	CLERK	800	20
JONES	MANAGER	2975	20
SCOTT	Analyst	3000	20
ADAMS	Clerk	1100	20
FORD	Analyst	3000	20

EName	Job	Sal	Deptno
ALLEN	Salesman	1600	30
WARD	Salesman	1250	30
MARTIN	Salesman	1250	30
BLAKE	manager	2850	30
TURNER	salesman	1500	30
JAMES	Clerk	950	30

ENAME	Job	Sal	Deptno
CLARK	MANAGER	2450	10
King	PRESIDENT	5000	10
MLLER	CLERK	1300	10

4] output of select clause

Count(*)	DEPTNO
5	20
6	30
3	10

WAQTD Total no. of of employes working in each job except for analysit.

→ Select count(*), job
From emp
where job Not IN 'analysit'
Group by job;

1 output of From Clause.

Enter emp table.

2] output of where Clause.

Ename	Job	Sal	Deptno
SMITH	CLERK	800	20
ALLEN	Salesman	1600	30
WARD	Salesman	1250	30
JONES	manager	2975	20
MARTIN	Salesman	1250	30
BLAKE	manager	2850	30
CLARK	manager	2450	10
King	PRESIDENT	5000	10
TURNER	salesman	1500	30
ADAMS	clerk	1100	20
JAMES	clerk	950	30
MILLER	clerk	1300	10

3. output of Group By Clause.

ENAME	job	Sal	Deptno.
SMITH	CLERK	800	20
ADAMS	CLERK	1100	20
JAMES	clerk	950	30
MILLER	CLERK	1300	10

ENAME	job	sal	Deptno
ALLEN	Salesman	1600	30
WARD	Salesman	1250	30
MARTIN	salesman	1250	30
TURNER	salesman	1500	30

ENAME	Job	sal	Depno
JONES	Manager	2975	20
BLAKE	manager	2850	30
CLARK	manager	2450	10

ENAME	Job	Sal	Deptno
King	PRESIDENT	5000	10

4 o/p of select clause

Count(*)	Job
4	Clerk
4	Salesman
3	Manager
1	PRESIDENT

Note:-

We can display only Group by column or expression along with any multirow function in the select clause.

1 WAQTD the maximum sal In each deptment

→ Select ~~Count(*)~~ MAX(sal), deptno
From emp
~~where~~
Group by deptno;

2] WAQTD The Total Salary needed to pay in each job except for the employees who work in deptno 20.

Select Sum(sal), job
From emp
where deptno != 20
Group by job;

3. WAQTD The total no. of employees whose name starts with 'S' in each deperment

→ Select count(*), deptno
From emp
where Ename like 'S%';
Group by deptno;

4 WAQTD The avorage sal needed to pay the employees ~~working~~ in who are Hired in month of Feb in each job.

→ Select Avg(sal), job
From emp
where ~~job~~ Hiredate like '%S%'
Group by job;

5] WAQTD the maximum salary in each job except for the employees who doesnot have manager.

→ Select max(sal), job
From emp
where Mgr is not NULL
group by job;

Note:-

- The argument which will pass in the group by clause it can be pass in the select clause. so it is known as Groupby column or Group by expression.
- We can pass multiple arguments to the Group by clause but it will group the records in combination.

WAQTD no.of employess working in same deperment and earning same sal.

→
```
Select count(*), deptno, sal
from emp
Group by deptno, sal;
```

WAQTD Total salary of the employees if the employee earnes more than 1250 in each desejnation

→
```
Select sum(sal), job
from emp
where sal > 1250
Group by job;
```

WAQTD no.of employess working in each deprment if there are at least 2 employess working in each deprment.

→
```
Select count(*), deptno
from emp
where
Group by deptno,
```

HAVING :-

- It is use to Filter the Groups
- It executes after the execution of Group by clause.
- It executes Group by Group.
- In ~~Group by~~ having clause we can write Group Filter Condition.
- we can pass multiple conditions in having clause by using logical operator
- we can pass multi row Function in having clause.

Syntax :-

```
Select GroupFunction/Groupbycloumn/Groupbyexp
From tablename
where <Filter-condition>
Group by columnname/Expression
Having <Group Filter-condition>
```

ORDER of execution

1. FROM
2. where ----> Row by Row (IF needed)
3. Groupby ----> Row-by-Row
4. Having ----> Groupby-Group
5. Select.

WAQTD no.of employess working in each dept if there atlest 2 employees working in each dept.

```
From emp
Group by Dno
Having Count(*) >= 2;
```

Emp

Eid	Ename	DNo.
1	T.Head	20
2	Dhoni	10
3	Raina	30
4	GAYLE	30
5	Virat	30
6	Bhumra	10
7	Rohit	10

O/p of Groupby Clause

20

1	T.Head	20

10

2	Dhoni	10
6	Bhumra	10
7	Rohit	10

30

3	Raina	30
4	Gayle	30
5	Virat	30

O/p of Having clause.

Count(*) >= 2

20 1 >= 2 (F)
10 3 >= 2 (T)
30 3 >= 2 (T)

O/p of select clause:

Count(*)	DNO.
3	10
3	20

WAQTD Designation of the employees In which there are atleast 2 Employees working In each Job.

→ Select ~~Count(*),~~ job
From emp
Group by job
~~Having job >= 2;~~
Having count(*) >= 2;

WAQTD Names that Are repeated exactly twice.

Select Ename
From emp
Group by Ename
having count(Ename) = 2;

WAQTD salaries that are Repeated.

→ Select Sal
From emp
Group by Sal
having Count(*) >= 2;

WAQTD Number of employees working In each Dept having Atleast 2 Employees character 'A' OR 'S' In their Names.

Select Count(*), ~~Ename~~ deptno
From emp
Group by deptno
Having Ename like '%A%' And

```
Select count(*), Deptno
from Emp
where Ename like '%A%' OR
      Ename like '%S%'
Group by deptno
Having count(*)>=2;
```

1. WQTD Number of employees working in each Dept. IF There are ATleast 2 Analyst in each Dept.

→
```
Select count(*), deptno
from emp
where Job = 'Analyst'
Group by deptno
Having count(*)>=2;
```

2. WQTD Deptno and Total Salary of each Dept IF there are ATleast 4 Employees working in each Dept.

→
```
Select sum(sal),
from emp
Group by Deptno
Having count(*)>=4;
```

3. WAQTD Number of employees working in each job IF There are ATleast 2 employees.

→
```
Select count(*), deptno
from emp
Group by Deptno
Having count(*)>=2;
```

4) WAQTD Job wise maximum salary IF the maximum salary of each job is more than 1500

→
```
Select max(sal), job
from emp
where salary >1500
Group by job
Having max(sal)>1500;
```

5] WAQTD Deptno and Avrg Salary Given To each Deptno IF The Average Salary is less than 1000

→ Select Avg(sal), deptno
From emp
Group by Deptno
Having Avg(sal) < 1000;

6. WAQTD Number OF employees working In each Dept IF there are atlest one manage In each Deptno

Select count(*), deptno
From emp
where job='manager'
Group by deptno
Having count(*) >= 1;

7] WAQTD The Names which are present more than once OR Repeated or Duplicate

→ Select Ename
from emp
Group by ename
having Count (Ename) >= 2;

8) WAQTD minum Salary OF each Dept exeluding the president themumium Salary Has To be more than 1000RS.

9. WAQTD Avgrage Salary Given to each job By Execuding The employees Whose Names Ends with 'S' And Avg of slary must be less than 1500.

→ Select Avg(sal), Ename
From emp
where ename like '%S' ✗
Group by job
Having Avg(sal) < 1500.

0) WAQTD Total Salary of each job IF total Salary of each job is greter than 3450

→ Select Sum(sal), job
From emp
Group by job
Having sum(sal) > 3450.

(2) WAQTD Number OF employees In each job and total Salary of The employees IF they are Earning Salary more than 1500

Select Count(*), job sum(sal)
From emp
Group by job
Having sum(sal) > 1500;

(3) WAQTD deptno And Total Salary needed To pay All the employees In each Deptno IF there are atlest 4 employees In each deptno.

* Difference between where Clause and having Clause

where Clause	Having Clause
1. It is used to filter the condition (records).	1. It is used to filter the groups.
2. It executes row by row.	2. It executes Group by Group.
3. It executes after the execution of from clause.	3. It executes group by clause.
4. In where clause we can't pass MRF.	4. In having clause we can pass MRF.
5	

WAQTD name and salary of the employees If the employess earning ~~receiving~~ more than 2000.

→ Select ename, sal
From emp
where sal > 2000;

WAQTD Name of the employees If the employess are erning more than Tanika.

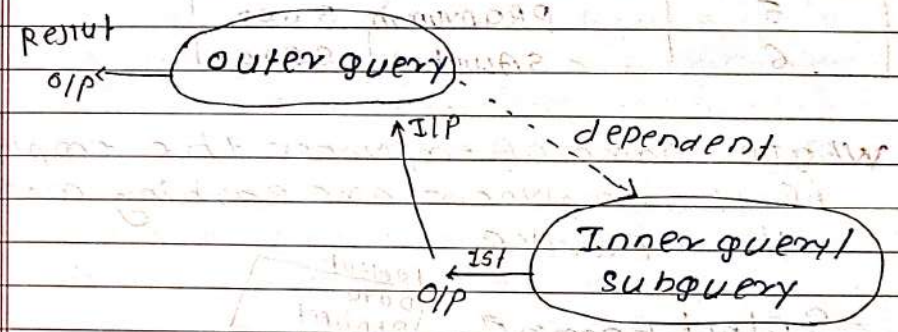
→ Select ename
From emp

* Subquery:-

A query writen inside another query is known as subquery.

* working processer / principle:-

- let us consider two querys
  1] outer query
  2] Inner query / subquery.



- Inner query execuites first and generats o/p.
- The o/p of inner query is passed i/p has an outer query.
- after that outer query executes and generates o/p.
- The o/p of the outerquery is known as Reisut.
- By this we can say outer query dependent on Inner query.

When or why do you subquery:-

Case 1:-

whenever we have unknowns then use concept of subquery.

emp:

Eid	Ename	Sal
1	Daya	3000
2	Abhijeet	3000
3	Tarika	2500
4	Freddy	1500
5	Pradyuman	5000
6	Salukhe	3500

WAQTD Name of employees the employees if the employees are earning more than Tarika.

→ Select Ename
From emp
where Sal > (Select sal
From emp
where ename = 'Tarika');

o/p Result:
Daya
Abhijeet
Pradyuman
Salukhe

(T) 3000 > 2500
(T) 3000 > 2500
(F) 2500 > 2500
(F) 1500 > 2500
(T) 5000 > 2500
(T) 3500 > 2500

WAQTD name of the employees earning more than ADAMS.

→ Select Ename
From emp
where Sal > (Select Sal
From emp
where ename = 'ADAMS');

2. WAQTD Name and Salary of employees earning less than king

→ Select Ename, Sal
From emp
where Sal < (Select Sal
From emp
where ename = 'king');

3. WAQTD Name and deptno of the employees if they are working in the same Dept as JONES.

→ Select ename, deptno
From emp
where Deptno = (Select Deptno
From emp
where ename = 'JONES');

4. WAQTD Name and job of all the employees working in the same Designation as James

→ Select ename, job
From emp
where job = (Select job
From emp
where ename = 'James');

5. WAQTD Empno and ename Along with Annual salary of All the employees IF their Annual salary is greater than WARD's Annual salary.

→ Select Empno, ename, Sal*12 as annual salary
From emp
where sal*12 > (Select sal*12
From emp
where ename='WARD');

6 WAQTD name, salary, Hiredate, IF the employess are earning more than Adams But Hired before miller.

→ Select ename, sal, Hiredate
From emp
Where sal > (select sal
from emp
where ename = 'Adams') And
Hiredate (select hiredate
From emp
where ~~Hiredate~~ ename = 'miller');

7] WAQTD all the details of the employeer IF the employees working in same designation as 'martin' and earns between 2000 to 2500.

Select *
From emp
where

8] WAQTD All the details of the employees IF the employees works in dept 20 or 30 and hired before King.

→ select *
From emp
where Deptno IN(20,30) And
Hiredate < (select Hiredate
From emp
where Ename='King');

H.W.

1] WAQTD Name And Hiredate OF the employees IF they Are Hired Before TURNER

→ Select ename, Hiredate
From emp
where Hiredate < (Select Hiredate
From emp
where ename='TURNER');

2. WAQTD Name And Hiredate OF the employees IF they are Hired After The president.

→ Select ename, Hiredate
From emp
where Hiredate > (Select Hiredate
From emp
where job='President');

3. WAQTD Name and sal of the employee IF They are earning sal less than the employee whose empno is 7839.

→ Select ename, sal
From emp
where sal < (select sal
From emp
where empno = 7839);

4 WAQTD All the details of employees if the employees are hired before miller.
→ select *
From emp
where Hiredate < (select Hiredat
from emp
where ~~Hiredate~~
ename = 'miller');

5 WAQTD Ename And Empno of the employess if employees are Earning more than allen.
→ Select ename, Empno
From emp
where sal > (select sal
From emp
where ename = 'Allen');

6. WAQTD Number of employees Hired after king.
select count(*)
From emp
where Hiredate > (select Hiredate
From emp
where ename = 'king');

7. WAQTD Total salary Given to the employees Working In The same dept as word.
→ Select ~~sal*12 as Total_salary~~ sum(sal)
From emp
sum(sal) where deptno = (select deptno
x From emp
where ename = 'word');

9. WAQTD dname of SCOTT
→ Select dname
From dept
where Deptno = (select deptno
From emp
where ename = 'SCOTT');

Case 2:- whenever the data to be selected and condition to be excuited are present different table an then we will use concept of subquery.

emp

EID	Ename	Dno
1	Allen	20
2	MARTIN	10
3	miller	10
4	SCOTT	20
5	King	30

DEPT

DNO	DName	LOL
10	D1	L1
20	D2	L2
30	D3	L3

WAQTD DName of SCOTT.

→ ④ select DName
① From Dept
② where dpno = (③ select Dno
10=20(F) ① From emp
20=20(T) ② where ename = 'SCOTT');
30=20(F)

② WAQTD Loc of SMITH.

→ select Loc
From Dept
where deptno = (select Deptno
From emp
where ename = 'SMITH');

③ WAQTD Dname of president.

→ select Dname
From Dept
where deptno = (select deptno
From emp
where job = 'president');

④ WAQTD Loc of the employeer whose Name Ender with charecter G.

→ select Loc
From Dept
where deptno = (select deptno
From emp
where ename like '%G');

⑤ WAQTD Detail of the employees In Accouting dept

→ select *
From emp
where deptno = (select deptno
From Dept
where Dname = 'Accouting');

⑥ WAQTD Number of employees working in Sales Dept.

select count(*)
From Emp
where deptno = (select deptno
From Dept
where dname = 'sales');

Note:-
- ~~The Column~~
- In Inner query we can select only one column or expression.
- In where clause of outer query and select clause Inner query columns need not to be same but the data type must be identical.

H.W

1] WAQTD details of the Employees earning more than 2000 In Accounting Dept.

```
Select *
From emp sal>2000;
Where ~~sal> (Select 2000~~
      Deptno = (select deptno
                From Dept
                Where Dname = 'Accounting'
                ~~sal>2000~~);
```

2] WAQTD Number of Employees working As A clerk In Research Dept.

→
```
Select count(*)
From Dept
Where Deptno = (Select deptno
                From emp
                Where Dname = 'Research' And
                      job = 'clerk');
```

3] WAQTD details of the employees working As A manager earning sal less than king In The location of New York.

→
```
Select *
From emp
where job = 'manager' And
      sal < (select sal
             From emp
             where ename = 'king') And
      deptno = (select deptno
                From Dept
                where Loc = 'New York');
```

4] WAQTD details of the employee Hired After SMITH Into Accounting dept.

→
```
Select *
From emp
where Hiredate > (Select Hiredate
                  From emp
                  where ename = 'smith') And
      Deptno = (select Deptno
                From Dept
                where Dname = 'Accounting');
```

5] WAQTD ~~of~~ Number of Employees whose name start with character 'A' working In the same DESIGNnation As James Hired After Jones Earning more than Smith And working In the location of CHICAGO.

```
→ Select Count(*)
  From emp
  where ename like 'A%' And
        job = (select job
               From emp
               where ename = 'James') And
        Hiredate = (select Hiredate
                    From emp
                    where ename = 'Jones') And
        sal > (select sal
               From emp
               where ename = 'SMITH') And
        deptno = (select deptno
                  from dept
                  where Loc = 'CHICAGO');
```

Q WAQTD Dname of the employeer if the employee works as peresident

```
→ Select Dname
  From Dept
  where deptno = (select Deptno
                  From emp
                  where job = 'President');
```

Q WAQTD Deptname of manager.

```
→ select Dname
  From Dept
  where deptno = (select Deptno
                  from emp
```

* Types of subquery:-

1] Single Row Subquery

2] multi Row subquery.

1] Single Row subquery:-

- If the subquery returns exactly one output or value it is known as single row subquery.
- In single row subquery we can use equal operater (=) as well as 'In' operater to comper.

EID	EName	Job	DNO		DNO	DName	Loc
1	Allen	Clerk	20	(F)	10	D1	L_1
2	King	peresident	10	(T)	20	D2	L_2
3	MILLER	Clerk	20	(F)	30	D3	L_3
4	Jones	MANAGER	30	(F)			
5	martin	Salesman	30	(F)			
6	SCOTT	manager	10	(F)			
7	Smith	manager	20	(F)			

```
→ ③ Select DName
  ① From Dept                    IP ③ 10
  ② where Dno = (select Dno
                 ① from emp
                 ② where job = 'Presendent');
```

10 = 10 (T)
20 = 10 (F)
30 = 10 (F)

2) Multi Row subquery:-

- If the subquery returns more than one o/p or value it is known as multi row subquery.
- In multi row subquery we can't use equals(=) operator instead we have to use 'In' operator to compare.

WAQTD DName of Manager.

→ Select Dname

(D1, D3, D1)

(10, 30, 20)

① Select Dname.
② From Dept.
③ where Dno In ④ Select Dno
① From emp
② where job = 'Manager');

(I) (10 In(10,20,30)
(II) 20 In(10,20,30)
(III) 30 In(10,20,30).

Note:- It is deficult to identify wether the subquery belongs to single-Row subquery or multi Row subquery therfor always use 'In' operator to compare.

WAQTD Maximum salary of the employeer IF the employess earning lessthan king and works as manager or clerck in each Deparment.

→ Select max(sal)
From emp
where sal < (select sal
From emp
where ename = 'king') and
job In ('manager', 'clerk')
deptno = deptno
From Dep
where deptno In(10,20,30);
Group by DNO.

WAQTD details of employess IF the employes nothing less than maxmium salary of Depment no = 10.

→ Select *
From emp
where sal < (Select max(sal), deptno
From emp
where deptno = 10);

WAQTD detail of employess IF the employes Hiredate allen and works in New York work in same decanition of miller and the employe Avg salary is less than. who works in deptno = 20.
the employes earns less than avg sal of employes deptno = 20.

→ Select *
From emp
where Hairdate = (Select Hairdate
From emp
where ename = 'Allen') And
Deptno IN (select deptno
From Dept
where Loc = 'Newyork') And
job = (select job
From emp
where ename = 'miller') And
Sal < (select Avg(sal)
From emp
where deptno = 20);

WAQTD detail of employee morethan

* Subquery operators:-
1. All
2. Any

All:- It is a special operater which has to use with relational operator
- It returns true IF it stastified all the value of RHS.
-

WAQTD Name of the employees who earns less than manager

Eid	Ename	Job	SAL
1	DHOLU	Salesman	950
2	BHOLU	Clerk	1200
3	KALIA	Manager	2000
4	TUNTUN	president	5000
5	KRAMDA	Manager	1100

DHOLU

(2) SELECT ename
(4) FROM emp
(5) where sal < All (select sal
(1) FROM emp
(3) where job = 'manager');

O/P
2000
1100

(T) 950 < All (2000, 1100)
(F) 1200 < All (2000, 1100)
(F) 2000 < All (2000, 1100)
(F) 5000 < All (2000, 1100)
(F) 1100 < All (2000, 1100)

Page No:
Date:

Any:- It is special operator which has to be used only with relational operator
- It returns true if it satisfied any one of the values at R.H.S.

WAQTD Name of the employees who earn a less than any one of the manager.

```
SELECT Ename
From emp
where sal<Any (select sal
               From emp
               where job='manager');
```

Bholu / Bhola / Akshanda

o/p: 2000 / 1100

(T) 950 < Any (2000,1100)
(T) 1200 < Any (2000,1100)
(F) 3000 < Any (2000,1100)
(F) 5000 < Any (2000,1100)
(T) 1100 < Any (2000,1100)

Note:-
- All and any operator as to be used along with relational operator
-

Page No:
Date:

WAQTD Name And salary For All the employees If they are earning less Than Atleast A manager.

```
select Ename, sal
From emp
where sal<Any (select sal
               From emp
               where job='manager');
```

WAQTD Name and hiredate of employees Hired before All the managers

```
Select Ename, hiredate
From emp
where hiredate<All (Select hiredate
                    From emp
                    where job='manager');
```

WAQTD Name of the employees hired after All the managers and Earning Salary more than All the clerk

```
Select Ename
From emp
where hiredate>All (select hiredate
                    From emp
                    where job='manager')
   And
      sal>All (select sal
               From emp
               where job='clerk');
```

WAQTD Details of the employees working as clerk and Hired before all the Salesman:

```
→ Select *
  From emp
  where job='clerk' And
        Hiredate < Any (Select Hiredate
                        From emp
                        where job='Salesman
```

WAQTD details of employees working In Accounting or Sales Dept:

```
→ select *
  From emp
  where job<
        Deptno In (Select Deptno
                   from Dept
                   where Dname In(Accounting
                   And 'Sales');
  fsel.
```

WAQTD Department Names of the employees whose Name is smith, king and miller.

```
→ Select Dname
  From Dept
  where Deptno In( Select Deptno
                   From emp
                   where Ename In('smith,
                   king, miller');
```

WAQTD details of employees working Newyork or Chicago.

```
→ Select *
  From emp
  where Deptno In( Select Deptno
                   From Dept
                   where Loc In('Newyork',
                   'Chicago');
```

WAQTD Emp Names if employees are hired after All the employees of Dept 30:

```
→ Select Ename
  From emp
  where Hiredate > All ( Select Hiredate
                         From emp
                         where Deptno=30);
```

WAQTD Maximum Salary.

```
→ select max(sal)
  From emp;
```

WAQTD Name of the employee who earns maximum Salary.

```
→ Select ename
  From emp
  where sal = ( Select max(sal)
                From emp);
```

Page No:
Date:

* Employee manager (MGR) relation:-

Case1:- To find reporting manager (empid = MGR).

* WAQTD Name of ~~Employee~~ Smith Reporting manager.

Eid	Ename	MGR
1	Allen	2
2	miller	3
3	SCOTT	5
4	Jones	2
5	king	NULL
6	Smith	5

=> ⑥ Select Ename
④ From emp ⑥ [5]
⑤ where Eid = (select MGR
1 = 5(F) ① From emp
2 = 5(T) ② where Ename
3 = 5(F) = 'Smith')
4 = 5(F)
5 = 5(T)
6 = 5(F)

* WAQTD miller's manager's Manager Name.

→ Select Ename
From emp
where Empno = (Select MGR
Form emp
where Empno = (select MGR
From emp
where ename = 'miller'))

WAQTD The location In which smith's Manager IS working.

→ Select Loc
From Dept
where Deptno = (Select Deptno

From emp
where Empno = (select MGR
From emp
where Ename = 'Smith'));

Q. WAQTD Name of the employees who are reporting king.

→ Select ename
From emp
where mgr in(Select Empno
From emp
where Ename = 'king');

* Case 2:- To Find employes
(MGR = employee).

WAQTD Name of the employees who are reporting King.
Scott
Smith.

(6) Select ename
(4) From emp
(5) where mgr In (3) Select Empno
(1) From emp
(2) where Ename = 'king');

2 In S(F)
3 In S(F)
5 In S(T)
2 In S(F)
Null In S(F)
5 In S(T)

WAQTD No. of employees who are reporting to blaek.

→ Select count(*)
From emp
where MGR In(Select Empno
From emp
where Ename = 'blake');

WAQTD Dparment name of employees who are reporting to king.
Select Dname
From Dept
where Deptno In(Select Deptno
From emp
where MGR In(Select Empno
From emp
where Ename='king'))

* Who To tell subquery In Interview:-
- DEF
- working process/principle
- Case
1]
2]
- Types of subquery
1) SRS
2) MRS
- Nested subquery.

Q WAQTD Dept name of MILLER's manager

→ Select Dname
From Dept
where Deptno In(Select Deptno
from emp
where Empno In(select MGR
From emp
where Ename:Miller

Q WAQTD Salary of ADAM's manager.

→ select sal
From emp
where Empno In(Select MGR
From emp
where Ename = 'ADAM');

Q WAQTD Number of employee Reporting to Jones

Select Count(*)
From emp
where ~~Empno~~ MGR In(Select Empno
From emp
where Ename = 'Jones');

Q WAQTD Deptname of the employees reporting to king.

Select Dname
From Dept
where Deptno In(select Deptno
From emp
where MGR In(Select Empno
From emp

Q WAQTD Names of the employees Reporting Blake's manager.

~~Select Ename~~
~~From emp~~
~~where Empno MGR In (Select Empno~~
~~From emp~~
~~where MGR In(Select Empno~~ MGR
~~From emp~~
~~where Ename: 'Blake'~~

Select Ename
From emp where
MGR In(Select Empno
From emp
where ~~Empno~~ Empno In(Select MGR
From emp
where Ename = 'Black');

Q WAQTD Loc of SMITH's Manager's manager.

→ Select Loc
From Dept
where Deptno In(Select Deptno
From emp
where Empno In
(Select MGR
From emp
where Empno In(Select MGR
From emp

* Singal Row Function:-

1] Length()
2] Concat()
3] Upper()
4] Lower()
5] InitCap()
6] SubStr()
7] Instr()
8] Replace()
9] Reverse()
10] MOD()
11] NVL()
12] ROUND()
13] TRUNC()
14] To_CHAR()

① Length() :- This Function is used to count the no. of characters present in the given string.

Syntax : LENGth('string')

Ex:-
Select Length('ADITYA');
FRom Dual;

Select Length('DATA WAL')
FRom Dual;

WAQTD Ename of the employess IF the employe Name have exact 4 charackr withouth using like operator.

→ Select Ename
From Emp
cobere Length(Ename) = 4;

* Dual :- It is a domi table which has exactly one Row and one Column and which give the result.

* Concat() :- This Function is used to jon the given two strings.

Syntax :- CONCAT('string1', 'string2')

Ex: Select Concat('MR.', 'Harshad')
From Dual;

3] Upper() :- This Function is used to Convert the given string into upper case.

Syntax : upper('string')
Ex:- Select upper('rupali')
From Dual;

4 Lower() :- This Function is used to convert the given string into lower case.

Syntax :- LOWER ('String')

Ex :-

Select Lower ('SHYAM')
From Dual;

5] INITCap() :- This Function is used to convert the inital alphabet of a string to the upper case and rest of the alphbet into lower case.

Syntax :- Initcap ('string')

EX :-

Select Initcap ('mayuresh')
From Duals

6] SubstR() :- This Function is used to exterate the part of the orginal string.

Syntax :-

SubstR ('orginalstring', Position, [Length])

length is optional

Ex :-

(1) substr('MAHABHARAT', 1, 4) = MAHA
(2) substr('MAHABHARAT', 6, 5) = HARAT
(3) substr('MAHABHARAT', 2, 9) = AHABHARAT
(4) substr('MAHABHARAT', 5) = BHARAT
(5) substr('MAHABHARAT', 2, 3) = AHA
(6) substr('MAHABHARAT', 4, 1) = A
(7) substr('MAHABHARAT', -6, 5) = BHART
(8) substr('MAHABHARAT', -10) = MAHABHARAT
(9) substr('MAHABHARAT', 5, 7) = BHARAT.

Q WAQTD First 3 character of Employeee name.
→ select substr (Ename, 1, 3)
From emp;

Q WAQTD last 3 character of Employeename.
→ select substr (Ename, -3, 3)
From emp.

Q. WAQTD the last character of the employee name.
→ select substr (Ename, -1)
From emp;

* WAQTD First HALF OF The employee Names

→ Select substr(Ename, 1, length(Ename)/2)
From emp;

* WAQTD Second half of the employeee Names

→ select substr(Ename, length(Ename)/2+1)
From emp;

H.W.

Q. WAQTD Names of All The Employee by excluding The 1st and The last Character

→ Select substr(Ename, 2, length(Ename)-2)
From emp;

Q. WAQTD Names of the employees IF The second character of The Name is L

→ Select substr(Ename, 2)
From emp
where Ename like('_L%');

Select Ename
From emp
where substr(Ename, 2, 1) = 'L';

length

WAQTD Details of An Employee IF The 1st Three characters of job is man

→ Select *
from emp
where substr(Job, 1, 3) = 'MAN';

WAQTD Name And hiredate of The employee Hired in The year 81

→ Select Ename, hiredate
From emp
where (Hiredate, length(Hiredate), length
where substr(Hiredate, length(hiredate) - 1) = '81';

OR.

Select Ename, Hiredate
From emp
where substr(Hiredate, -2) = 81;

7] Reverse() :- This Function is used to reverse the given String

Syntax :- reverse('string')

Ex

Select Reverse('VIKARSH')
From Dual; = HSRAKIV

Select Reverse('Banana')
From Dual; = ANANAB

8] ~~REPLACE~~ () :-

8] REPLACE():- This Function is used to replace given 'old' string with 'New' string OR it will remove the 'old' string

Syntax :-

Replace ('orginal_string', 'old_string', 'New_string') optional

Ex.

Select Replace ('BANana', 'N', 'T')
From dual; = BATATA

Select replace ('BANANA', 'N')
From Dual; = BAAA

Q WAQTD count the no. of character 'A' present in the particular Names.

→ ~~Select~~ Count (*)
From Dual
where Ename = 'A';

BANANA

Select Length (Ename) - length (Replace (Ename, 'A'))
From emp;

Q WAQTD name of the employee whose name have character 'A' exactly twice.

5-3 = 2 6-4

→ Select length(Ename) = 2
Select Ename
From emp
where length(Ename) - length(Replace(Ename, 'A')) = 2;

9] INSTR() :- This Function is used to obtain the position of the string which is present in the orginal string.

Syntax :-

INSTR('orginal_string', 'string', 'Position', [Occurance])

Ex.

INSTR('MALAYALAM', 'A', 1, 2) = 4
From Dual;

INSTR('MALAYALAM', 'A', 3, 3) = 8
From Dual;

INSTR('MALAYALAM', 'A', 5, 7) = 6
From Dual;

INSTR('MALAYALAM', 'A', 5) = 6
From Dual;

INSTR('MALAYALAM', 'A', -3, 3) = 2
From Dual;

Q WAQTD Names of the employee who have character 'A' In their name.

```
Select Ename
From emp
where Instr(Ename, 'A', 1) > 0;
```

Q WAQTD Names of the employee who have character 'L' Atleast Twice in their Name.

→
```
select Ename
From emp
where Instr(Ename, 'L', 1, 2) > 0;
```

WAQTD Names of the employee who have character 'A' Atleast 3 times in their name.

→
```
Select Ename
From emp
where Instr(Ename, 'A', 1, 3) > 0;
```

* ROUND(): This function is used to round() the given number to its nearest value.

Syntax: ROUND(Number)

Ex: Select ROUND(7.2) = 7
From Dual;

Select ROUND(7.8) = 8
From Dual;

SelectROUND(7.5) = 8
From Dual;

WAQTD Round Average salary of the employee

→
```
select Round(sal)
From emp;
Select Round(Avg(sal))
From Emp;
```

* TRUNC(): This function is used to round() the given Number to the list value.

Syntax:

TRUNC(Number).

Ex:
```
Select TRUNC(7.9) = 7
From Dual;

select TRUNC(7.5) = 7
From Dual;

select TRUNC(7.1) = 7
From Dual;
```

* MOD(): This function is used to obtain the remider of the given value.

Syntax:- MOD(m,n)

Ex:- Select MOD(4,2)
From Dual;

Select MOD(5,2)
From Dual;

* NVL():-

NULL Value logic.

This Function is used to eliminate the side EFFECT OF NULL in arithmatic operation.

Syntax:- NVL(argument1, argument2)

Emp

Eid	Ename	SAL	Comm
1	A	800	200
2	B	1250	NULL
3	C	1500	0
4	D	2500	NULL
5	E	1920	20

WAQTD Ename and Total earning of each employee.

→ Select Ename, sal+NVL(comm,0)
From emp;

800 + 200 = 1000 200
1250 + 0 = 1250 NULL
1500 + 0 = 1500 0
2500 + 0 = 2500 NULL
1920 + 20 = 1940 20.

Ename	SAL+NVL(comm,0)
A	1000
B	1250
C	1500
D	2500
E	1940

14] To_Char(): This Function is used to convert the given date into string Format by using Format models.

Syntax: To_CHAR(Date, 'Format_models')

Formatmodels:-

1. Year	8. Day
2. YYYY	9. D - weekday.
3. YY	10. HH24
4. MONTH	11. HH12
5. MON	12. MI
6. MM	13. SS
7. DD	

Page No:
Date.

Ex:-

① Select To_char(sysdate, 'Year: month: DD: Day : HH24: MI: SS')
From Dual;

② select To_char(sysdate, 'YY: mm: Day: D: HH: 12: MI: SS')
From Dual;

WAQTD details of the employeer IF the employes haired in the month of Feb.

→ Select *
From emp
where To_char(Hiredate 'month') = 'Feb';

OR.

where To_Char(Hiredate 'mon') = 02;

WAQTD details of the employeer who haird in Wensday.

→ Select *
From emp
where To_char(Hiredate, 'Day') = 'WEDNESDAY';

Page No
Date

JOINS

* joins:- It is used to retrive the data From multiple table similtansoyb.

There are 5 types in JOINS:-

1. Cartesian join / cross join
2. Inner join / Equi join
3. OUTER join
   → Left outer join
   → Right outer join
   → Full outer join
4. Nature join
5. SELF join.

① CARTESIAN

In Cartesian joins The records of table1 will marge with all the records of table2.

Syntax:- ANSI (American national Standard Institute).

SELECT Col-Name
FROM Table-Name1 Cross join
Table-Name2;

ORACal

```
SELECT Col-Name
FROM Table_Name1, Table-Name2;
```

Emp

Eid	EName	DNO
1	A	20
2	B	30
3	C	10

DePT

DNO	DName	Loc
10	D1	L1
20	D2	L2
30	D3	L3

```
Select *
From emp, Dept;
```

Eid	EName	DNO	DNO	Dname	Loc
1	A	20	10	D1	L1
1	A	20	20	D2	L2
1	A	20	30	D3	L3
2	B	30	10	D1	L1
2	B	30	20	D2	L2
2	B	30	30	D3	L3
3	C	10	10	D1	L1
3	C	10	20	D2	L2
3	C	10	30	D3	L3

Note:-

- In cartesian joins will get matching recorder along with plenty of unmached records.
- In cartesion
- The columns are present in the result table will be equal to The sumation of columns are present in the both the Table.
- The records are the present in the result table will be equal to the product of records are present in the both the table.

* INNER JOIN (Equi Join):-

It is used to obtain matching recordes.

Syntax:-

ANSI

```
SELECT Column_name
FROM Table-Name1 INNER JOIN Table-Name2
ON <join_condition>
```

ORCal

```
SELECT Column_name
FROM Table-Name1, Table-Name2
where <join_condition>
```

JOIN :- It is used to merge two table.

Syntax:

Table.Name1.columnname = Tablename2.columnname

Eid	Ename	DNO
1	A	20
2	B	10
3	C	20
4	D	30

DNO	Dname	Loc
10	D1	L1
20	D2	L2
30	D3	L3

WAQTD Ename & Dname.

→ ③ Select Ename, Dname
① From Emp, Dept
② where Emp.DNO = Dept.DNO;

Join condition.

20 = 10(F)	10 = 10(T)	20 = 10(F)	30 = 10(F)
20 = 20(T)	10 = 20(F)	20 = 20(T)	30 = 20(F)
20 = 30(F)	10 = 30(F)	20 = 30(F)	30 = 30(T)

Result table.

Ename	Dname
A	D2
B	D1
C	D2
D	D3

WAQTD Employee Details and Dept Details.

→ Select Emp.*, Dept.*
From Emp, Dept
where Emp.Deptno = Dept.Deptno;

WAQTD Employee Name and Salary along with Their location IF they are earning more than 2000.

→ Select Ename, sal, Loc
From Emp, Dept
where sal > 2000 And
Emp.Deptno = Dept.Deptno;

WAQTD Employee Name And His Deptno along with DeptName and location IF the Employee is working As Manager.

→ Select Ename, (Deptno), Dname, Loc
From emp, Dept
where Emp.Deptno = Dept.Deptno And
~~Empno~~ job = 'manager';

both table present.

Select Ename, Emp.Deptno, Dname, Loc
From emp, Dept
where Emp.Deptno = Dept.Deptno And
job = 'manager';

Q WAQTD details of the employee Along with Dept name if the employee is earning more than 1000 and works in a Dept no 20.

→ Select Emp.*, Dname.
From emp, Deptno
where Emp.Deptno = Dept.Deptno And
sal > 1000 And
Emp.Deptno = 20;

WAQTD Employee Name and Dptname of the employee who IS working As manager IN Loc DALLAS.

→ Select Ename, Dname
From emp, Dept
where Emp.Deptno = Dept.Deptno And
job = 'manager' And
Loc = 'DALLAS';

H.W.

Q. WAQTD Name OF The Employee And The Location OF all the employees
→ Select Ename, Loc
From emp, Dept
where Emp.Deptno = Dept.Deptno;

Q. WAQTD DName and Salary For all the employee working in accounting
→ Select Dname, Sal
From emp, Dept
where Emp.Deptno = Dept.Deptno And
Dname = 'Accounting';

Q. WAQTD Dname and annual salary For all Employees whos salary is more than 2340.
→ Select Dname, sal*12 As annualsalary
From emp, Dept
where Emp.Deptno = Dept.Deptno And
Sal > 2340;

Q. WAQTD Ename and DName For Employees having Character 'A' In their Dname.
→ Select EName, DName
From emp, Dept
where Emp.Deptno = Dept.Deptno And
Instr (Ename, 'A', 1) > 0;

Q. WAQTD Ename and Dname For all the employees working as salesman.
→ Select Ename, Dname
From emp, Dept
where Dept = Dept
Emp.Deptno = Dept.Deptno And
job = 'salesman';

Q. WAQTD Dname And job For all the employees whos job And Dname starts with Character 'S'.
→ Select Dname, job
From emp, Dept
where Emp.Deptno = Dept.Deptno And
Dname like 'S%';

Q. WAQTD Dname and MGR No For Employees Reporting to 7839

→ Select Dname, MGR
From emp, Dept
where Emp.Deptno = Dept.Deptno And
MGR = 7839;

Q WAQTD Dname and hiredate For Employees Hired after 83 INTO Accounting or Research Department

→ Select Dname, Hiredate
From emp, Dept
where Emp.Deptno = Dept.Deptno And
Hiredate > '31-Dec-83' And
Dname In ('Accounting, Research');

Q WAQTD Ename and Dname of the employees who are Getting Comm In Dept 10 OR 30

→ Select Ename, Dname
From Emp, Dept
where Emp.Deptno = Dept.Deptno And
Comm is Not NULL And
Emp.Deptno In (10, 30)

Q. WAQTD Dname And Empno For all the employees who's Empno are (7839, 7902) And working In Loc NEW YORK

→ Select Dname, Empno
From emp, Dept
where Emp.Deptno = Dept.Deptno and
Empno In (7839, 7902) And
Loc = 'NEW YORK';

* OUTER JOIN :- It is used to obtain unmached recorder along with matching records

There are 3 types in outer join.
a. Left outer join
b. Right outer join
c. Full outer join.

a. left outer join :- It is used to obtain unmatched recorder Form the left table along with matching records.

Syntax:
ANSI

Select Column-Name
FROM Table-Name1 LEFT [outer] join Table-Name2 (optional)
ON <join-condition>

Oracle.

Select Column-Name
FRom Table-Name1, Table-Name2
where Table-Name1.Column-Name = Table-Name2.Column-Name (+);

Emp

ENamc	DNO
A	20
B	Null
C	30
D	Null

Dept

DNO	Dnamc
10	D1
20	D2
30	D3
40	D4.

→ Select *
FRom Emp, Dept
where Emp.DNO = Dept.DNO(+);

20 = 10 (F)	Null = 10(F)	30 = 10(F)	Null = 10(F)
20 = 20(T)	Null = 20(F)	30 = 20(F)	Null = 20(F)
20 = 30(F)	Null = 30(F)	30 = 30(T)	Null = 30(F)
20 = 40(F)	Null = 40(F)	30 = 40(F)	Null = 40(F)

Null Null

Result table

Ename	DNO	DNO	Dname
A	20	20	D2
B	30	30	D3
C	Null	Null	Null
D	Null	Null	Null

2 matching records

2 unmatched records.

* Right outer joint :-
It is used to obtain unmatched records from the right table along with matching records.

Syntax:- ANSI
SELect Column-name
From Table-Name1 Right[Outer] join Table Name2
ON <join-condition>

~~SELECT Column-name~~
~~From Table-Name1, Table-Name2~~
~~where Table-Name.column.name = TableName2.~~
Column name

SELECT Column-name
From Table-Name1, Table Name2
where Table-Name.Column-name(+) = Table-Name2. Colum
Nome

* Select *
From emp, Dept
where emp.DNO(+) = Dept.DNO;

20 = 10(F)	Null = 10(F)	30 = 10(F)	Null = 10 (F)
20 = 20(T)	Null = 20(F)	30 = 20(F)	Null = 20 (F)
20 = 30(F)	Null = 30(F)	30 = 30(T)	Null = 30 (F)
20 = 40(F)	Null = 40(F)	30 = 40 (F)	Null = 40 (F)

= Null Null

Result table:-

EName	DNO	DNO	DName
A	20	20	D2
C	30	30	D3
NULL	NULL	10	D1
NULL	NULL	40	D4

* Full outer join:-
It is used to obtain unmatched records from the both the tables along with matching records.

Syntax:- ANSI

SELECT Column-Name
FROM Table-Name1, Full [outer] join Table-Name2
ON <join-condition>

Select *
FROM emp Full join Dept
ON Emp.DNO = Dept.DNO;

Result table:-

EName	DNO	DNO	DName	
A	20	20	D2	matching
C	30	30	D3	
B	NULL	NULL	NULL	unmatched result of left table
D	NULL	NULL	NULL	
NULL	NULL	10	D1	unmatched Result Right table
NULL	NULL	40	D4	

* Natural join :- It has two behaviors
a. It behaviours as Inner join when there is relation between the table.
b. It behaviours as Cartission join when there is No relation between the table.

Syntax:- ANSI.

SELECT Column-Name
FROM Table-Name1 Natural join Table-Name2

* Self Join:-
- Joining a table itself or joining same two tables it is known as self join.
* when or why do we use self join
- whenever the data to be selected present in same table but in different records then we use the concept of self join.

Syntax:- ANSI

Select Column-Name
From Table-name1 join Table-Name2.
ON <join-condition>;

ORACLE

Select Column-name
From table-name1, table-name2
where <join-condition>;

Q. WAQTD ename, mgr name

→ Select e1.ename, e2.ename as mgr-name
From emp e1, emp e2
where e1.MGR = e2.EID

EID	EName	MGR
1	Smith	5
2	Jones	4
3	martin	2
4	King	NULL
5	Ford	4

EID	EName	MGR
1	SMITH	5
2	Jones	4
3	martin	2
4	King	NULL
5	Ford	4

5=1(F)	4=1(F)	2=1(F)	NULL=1(F)	4=1(F)
5=2(F)	4=2(F)	2=2(T)	NULL=2(F)	4=2(F)
5=3(F)	4=3(F)	2=3(F)	NULL=3(F)	4=3(F)
5=4(F)	4=4(T)	2=4(F)	NULL=4(F)	4=4(T)
5=5(T)	4=5(F)	2=5(F)	NULL=5(F)	4=5(F)

o/p of Select Clause

EName	MGR-Name
SMITH	FORD
JONES	King
martin	Jones
FORD	King

Q. WAQTD name of the emp and his manager's name if emp is working as clerk.

→ Select e1.ename, e2.ename as MGR-Name
From emp e1, emp e2
where e1.MGR = e2.empno And
e1.job = 'clerk';

Q. WAQTD name of the emp and manager's designation if manager works in dept 10 or 20.

→ select e1.ename, e2.job as manager_job
From emp e1, emp e2
where e1.ename = e2.empno And
e2.deptno In (10,20);

Q. WAQTD name of the emp and manager salary if emp and manager both earn more than 2300.

→ select e1.ename, e2.sal as mgr-sal
From emp e1, emp e2
where e1.ename = e2.empno And
e2.sal > 2300; And e2.sal > 2300;

* To Join multiple tables.
- To join n number of tables we need to write n-1 join condition.
- we can join 256 tables.

1] Inner join
EMP.DNO = DEPT.DNO
Emp — Dept

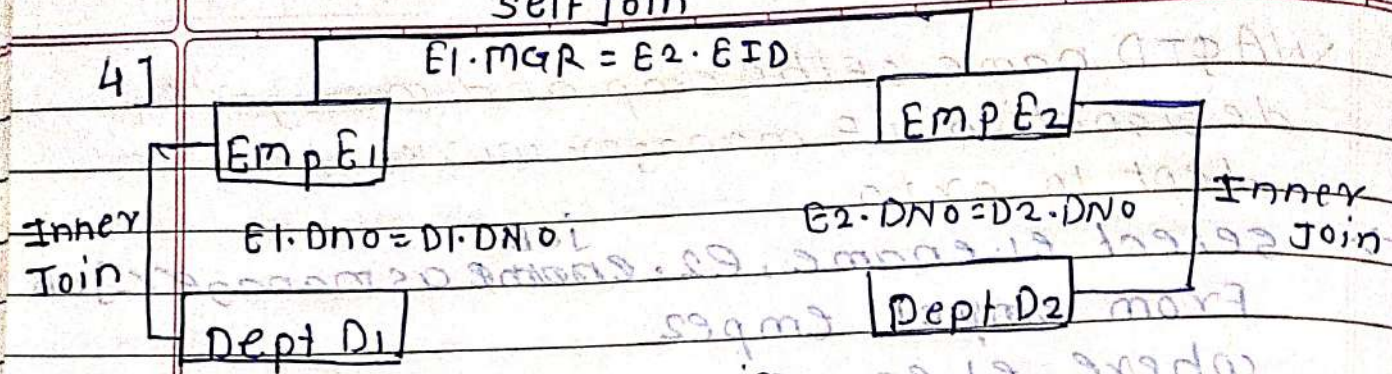
2] Self Join
e1.MGR = e2.EID
Emp E1 — Emp E2

3] Self Join
E1.MGR = E2.EID
Emp.E1 — Emp.E2
Inner Join
E1.DNO = D1.DNO
Dept D1

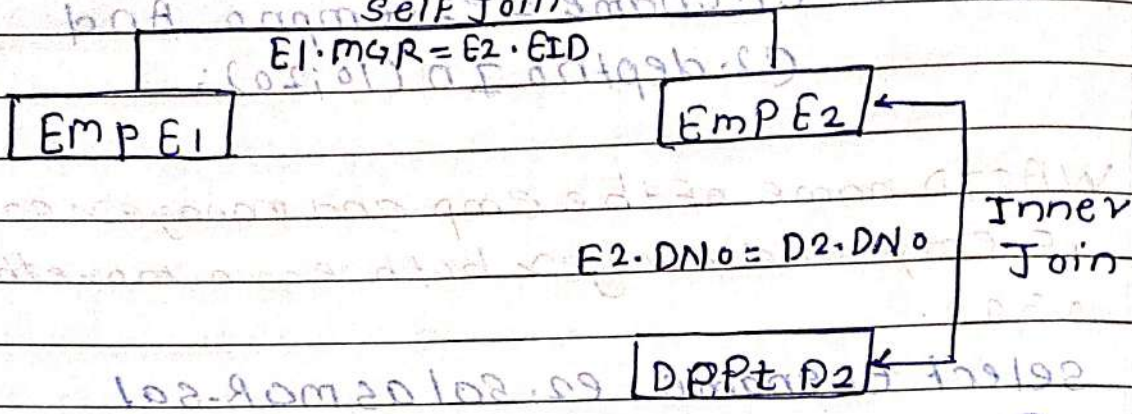
Page No:
Date.

Self Join

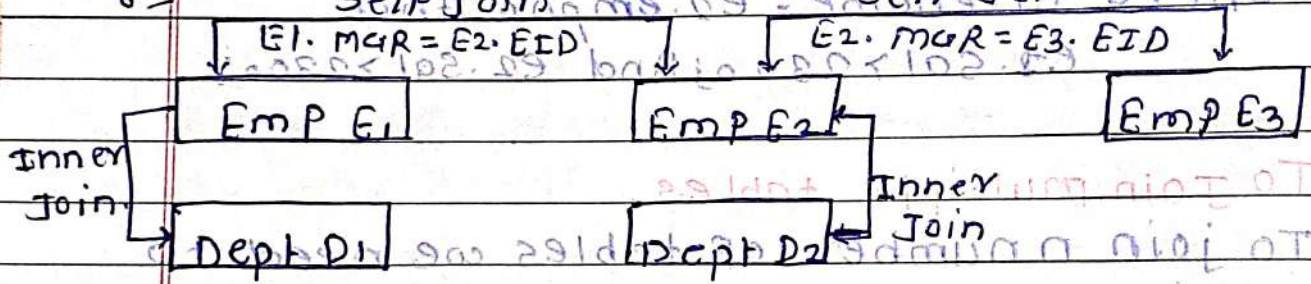
4]



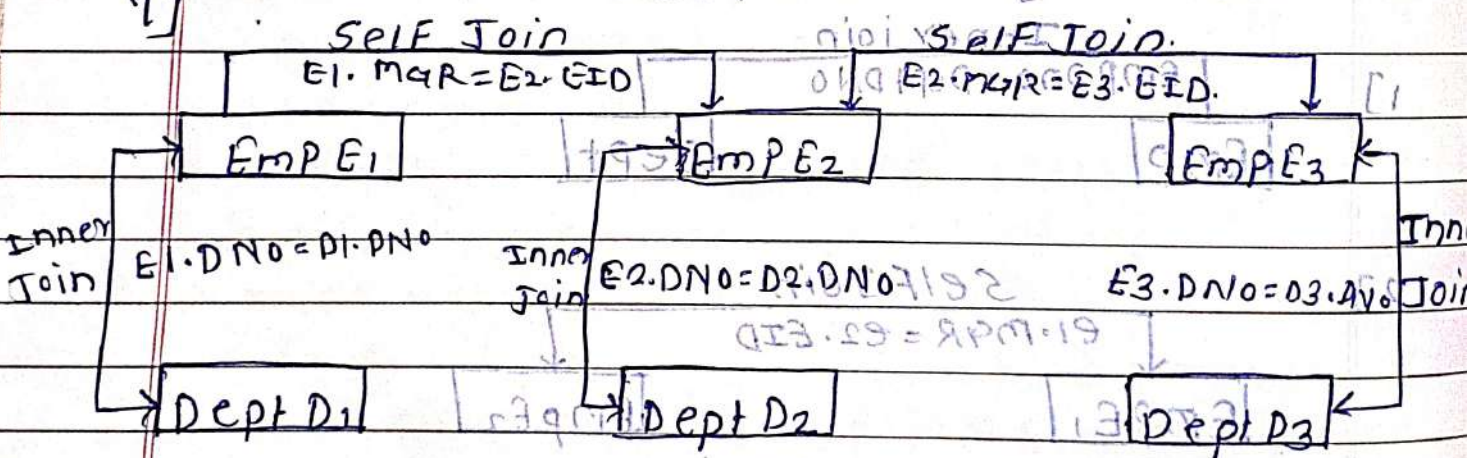
5]



6]



7]



Note:- This statement are DDL statement.

DDL

Q. WAQTD Salary IN Ascending order.

→
```
Select
From emp
order by sal;
```

* ORDER BY:-
- It is used to sort the records in ascending or descending order.
- It executes after the execution of select clause.
- order by clause must be return as a last clause in the statement.
- For order by clause we can pass column-name or expression as an argument.
- For order by clause we can pass alias name as an argument.
- By default order by clause sort records in ascending order.
- We can pass multiple argument to the order by clause but it will sort the records in combination.

Syntax:-
```
Select Group-Function/Group expression/ column-name
From table_name
[where < Filter condition>]
[Group by Column-name/ expression]
[having < Group-Filter-condition> ]
ORDER By Column-name/Expression [ASC] DESC;
```
[ ] optional

ORDER of Execution:-
1. FROM
2. where (row by row)
3. Group by (row by row)
4. Having (Group by Group)
5. Select (row by row)
6. ORDER By (Group by Group)

Q. WAQTD salary IN Ascending order:
→
```
Select sal
From emp
order by sal asc;
```

O/P From clause:

EID	EName	SAL	DNO
1	SIDDHI	3000	10
2	SIDHU	1500	10
3	HARSHADA	2000	20
4	HARSHAD	2000	20
5	CHANCHL	3000	10

O/p of select clause. → O/p OF ORDER by clause

sal
3000
1500
2000
2000
3000

sal
1500
2000
2000
3000
3000

Q. WAQTD ename IN Asc order.
→
```
Select ename
From emp
ORDER by ename ASC;
```

Q. WAQTD sal in desc order
→
```
Select Sal
From emp
ORDER by sal DESC;
```

Q. WAQTD ename in desc order
→ select ename
From emp
order by ename desc;

Q WAQTD ename and deptno in ascorder.
→ select ename, deptno
From emp
order by ename, deptno;

Q WAQTD annual sal in desc order
→ select sal*12 as Annual_sal
From emp
order by sal*12 Asannual_sal Desc;

* DDL (Data Definition language):
- It is used to construct, modify and remove an object
- There are 5 statements in DDL:
① CREATE ② RENAME ③ ALTER
④ TRUNCATE ⑤ DROP

① CREATE:- This statement is used to construct an object.

Syntax:-
```
CREATE TABLE TABLE-Name
(
Column-name1 Datatype Constraints,
Column-name2 Datatype Constraints,
Column-name3 Datatype Constraints,
  :
Column-namen Datatype Constraints
);
```

* Blueprint

Datatype / Column-name	number(5) TID	Varchar(15) Tname	varchar(15) O-name	Number(10) mob-No	Varchar(20) Email
Unique		Unique			
NOTNULL		NOTNULL	NOTNULL	NOTNULL	NOTNULL
check	check length(TID)=5)			check (length(mob-No) =10)	
PK	PK				
FK					
Constraints					

Q Create IPL Table:-

→
```
CREATE TABLE IPL_TEAM.
(
TID Number(5) PK check(Length(TID)=5),
TName VARCHAR(15) Unique NOTNULL,
O-Name VARCHAR(15) NOTNULL,
mob-No Number(10) Unique NOT NULL check
                    (length(mob-No)=10),
Email varchar(15) Unique NOT NULL
);
```

Page No:
Date: 1 4 25

* Create a Table name using FOREIGN KEY.

Syntax:

```
CREATE Table Table-Name
(
 Col-Name1 Datatype Constraint,
 Col-Name2 Datatype Constraint,
 Col-Name3 Datatype Constraint,
 :
 :
 Col-NameN Datatype Constraint,
 Constraint ref-Name Foreign key(Col-Name)
 Refereance Table-Name (ColName)
 );
```

Datatype	Number(7)	Varchar(10)	Number(2)	Number(10)	Varchar(15)	Number(3)	Number(5)
Col-Name	PID	Pname	AGE	mob-No	Email	No-matches	TId
Unique				Unique	Unique		
NOTNULL	NOTNULL	NOTNULL	NOTNULL	NOTNULL	NOTNULL	NOTNULL	
check				Check Number(10)			
PK	PK						
FK							FK.

Constraints

- Foreign key which can accept multiple null value.
- Foreign key always generate after primary key.
- Thus they do not automatically increment.
- It supports clustered or non-clustered indexes and you can have more than one Foreign key in table.

→

```
Create Table Players
(
PID Number(7) Primary key,
Pname Varchar(15) NOT NULL,
AGE Number(2) NOTNULL,
MOB-NO Number(10) unique NOTNULL check
                  (length(MOB-NO)=10),
Email Varchar(15) NOTNULL,
No-matches Number(3) NOTNULL,
TID Number(5),
Constraint TIDFK Foreign key(TID)
References IPL-Team(TId)
);
```

primary key :- A primary key is a column that uniquely identifies each record in a table

Foreign Key :- A Foreign key is a column that links one table to another table.

EX.

```
Create Table course(
    CID Number primary key,
    CourseName Varchar(20)
    );
Create Table Enrollement(
    EID Number PRIMARY key,
    CID Number,
    Foreign Key (CID) References course(CID));
```

* To Copy a Table

Syntax:-

```
Create Table Table-Name
As
Select sql-statement;
```

Ex.

```
Create Table Salesman
As
Select *
FRom emp
where job='Salesman';
```

* To Create View:- It is a virtual table which is created on the result of the query.
- It will not occupies storage.
- Any Changes done on the orginal table it will be impact on view.

Syntax:-

```
CREATE view view-Name
As
SQL-Statement;
```

Ex:-

```
Create view M1
As
Select *
From emp
where job='Manager';
```

* RENAME :-
It is used to change the existing Object Name.

Syntax:-

```
Rename OldObjectName to NewobjectName;
```

Ex:-

```
Rename STUD TO SuperMan;
```

* Truncate
It is used to remove All the recorder from the table permently.

Syntax:-

```
Truncate Table Table-Name;
```

Ex:-

```
Truncate Table Salesman;
```

* Drop:-
It is used to remove an object (Table).

Syntax:-

```
Drop Table Table-Name;
```

Ex:-

```
Drop Table Superman;
```

* Flashback (To recover the Table)

syntax:

Flashback Table Tablename
To Before Drop;

EX:- Flashback Table superman
TO Before DROP;

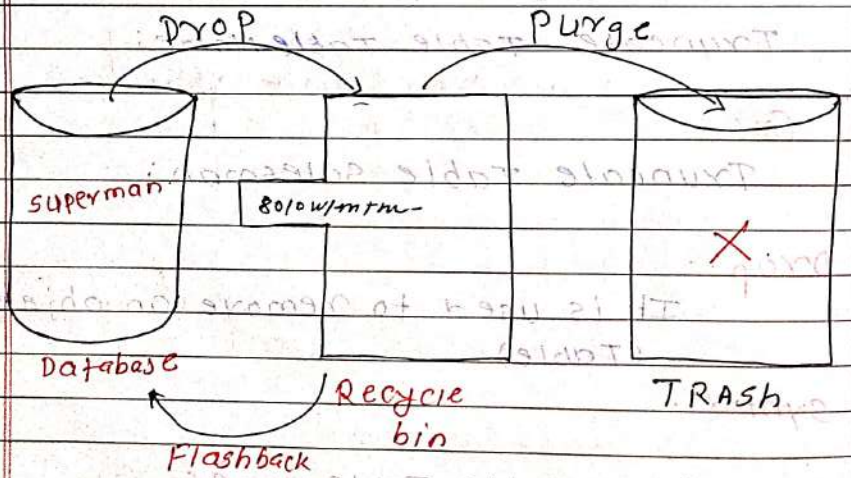
* PURGE (To delete From Recyclebin)

Syntax:-

purge Table Table-Name;

EX:-

purge Table superman



DML (Data manipulation Language):-

- This statement is used to manipulate the object by performing inseration, update and deleation.
- There are 3 statement in DML

(I) Insert
(II) Update
(III) delete

(1) Insert:-

This statement is used to insert the record or add the records to the table.

syntax:-

INSERT INTO Table-name
values (v1, v2, v3 ---- vn);

EX:- Insert Into IPL-Team
values (12341, 'Ambani', 8050915915
gmbu@gmail.com);

OR

Syntax:- Insert Into table-Name (col1, col2, ... coln)
values (v1, v2, v3, ... vn);

EX:-

Insert Into Ipl-Team (Tname, TID, O-Name, Mob-No, Email)
values ('CSK', 12342, 'Harshad', 8080915914, aharshe@gmail.com);

OR

Syntax:
Insert Into Tablename
Values(&col1, &col2, &col3 ..... &colN);
Ex:-

Insert Into ~~tab~~ Ipl_Team
Values(&TID, &Tname, &O_name, &Mob_No,
&Email);

only / Enter केले की add hot जाते मग जेवे Value insert karychi

* Update:- It is used to update the existing record or data.

Syntax:-
UPDATE Table_Name
SET Col_Name=Value, Col_Name=Value, ... Col_Name=Value
where <Filter-Condition>;
Ex:-

① Update IPL_Team
SET O_Name='KEERTI'
where TID=12342;

② Update Ipl_Team
SET O_Name='Harshad'
where TID In(12341,1234

* DELETE:- This statement is used to delete the particular record.

Syntax:-
delete From Table_Name
where <Filter-condition>

Ex:-
delete From IPL_Team
where TID=12344;

* TCL:- (Trasaction Control language)

- there are 3 stetements in TCL:-

① Commit

② Savepoint

③ Rollback

① Commit:- It is used to save the transaction

Syntax:-

Commit;

② Savepoint:- It is used to mark the transaction.

- It can be used along with Rollback.

Syntax:- savepoint savepoint-Name;

Ex:- savepoint P1,

③ Rollback:-

It is used to get back the letest saved transaction.

Syntax:-

Rollback;

Syntax:-

Rollback To savepoint-Name;

EX:- Rollback To P1

* DCL (Data control language):-

- It is used to control the flow of data.
- There are 2 statement in DCL

① GRANT

② REVOKE

① GRANT

It is used To Give permission To a user.

Syntax:-

Grant Sql-Statement ON Table-Name To user-Name;

Ex:-

Grant select ON Emp TO HR;

② Revoke:- It is used to Take back the given permission From user.

Syntax:-

Revoke Sql-Statement ON Table-Name From user-Name;

Ex:-

Revoke Select ON Emp From HR;

* To connect other user / To Jump from one user to another

Connect / conn

UserName : SCOTT/HR

Password : Tiger.

* pseudo Columns:-
- This are the false column which are present in each and every table and must be call explicitly.
- There are two pseudo Column

1. ROWID
2. RowNum.

* ROWID:-
- It is an 18 digit address of a records in the memory.
- ROWID is generated at the time of insertion.
- ROWIDs are unique
- ROWIDs cannot be change. (it is static).
- ROWIDs quickest or fastest way to access the record
- when their is primary key in the table ROWID behaviour as primary key.

Ex:-

```
SELECT ROWID, Emp.*
From emp;
```

ROWID	Ename	SAL
AAA.....A	smith	950
AAA...B	Allen	150
AAA.....C	miller	3000
AAA.....D	SCOTT	1800
AAA...E	James	1800

* Note:-
- ROWID cannot be deleted.

* ROWNUM:- it is an serial number assigned to the result table.
- ROWNum always start with 1
- ROWNum is pynamic (not static)
- ROWNum is generat at the time of execution.

Ex:-

```
SELECT ROWNum, Emp.*
FROM Emp;
```

ROWNum	Ename	SAL
1	Smith	950
2	Allen	1250
3	miller	3000
4	James	1800
5	SCOTT	1500

Page No:
Date.

1) WAQTD First record of Employee table

→ Select *
Form emp
where RowId = (Select RowID ..AAA
Form Emp ....AAA
where ....AAA

Select *
Form emp
where RowID=1;

2) WAQTD First 3 records of Employee table.

→ Select *
Form emp
where RowID<4;

3) WAQTD 3rd record of Employee table

→ Select *
Form emp
where RowNum=3;

Page No:
Date.

1) WAQTD First 3 records of Employee table.

→ Select *
Form Emp
where RowNum<=3;

RowNum	EName	SAL.
1	Smith	950
2	Allen	1250
3	miller	1500
4	James	1800
4	SCOTT	1500

WAQTD 3rd record of Employee table

→ Select *
Form Emp
where RowNum=3;

RowNum	EName	SAL
1	smith	950
2	Allen	1250
3	miller	1500
4	James	1800
5	SCOTT	1500

+ TO make rownum as static:-

- Step-1 :- assign Rownum to the table and For rownum assign ALIS Name AS SLNO.

```
select rownum AS SLNO, Emp.*
Form emp;
```

Result table

Emp

ROWNUM / SLNO	EName	SAL	
1	Smith	850	(F)
2	Allen	1250	(F)
3	miller	3000	(T)
4	JAmes	1800	(F)
5	SCOTT	1500	(F)

- Step2:- use step1 in the Form Clause of outer query and use the where clause to get expected result.

```
select *
From (select Rownum AS SLNO, Emp.*
      From Emp)
where SLNO=3;
```

Result Table

SLNO	EName	SAL
3	miller	3000

Q. WAQTD 1st Half of records of the emp tables.

```
select * From(select ROWNUM as FirstHalf,emp.*
              From emp)
where FirstHALF<=(select count(*)/2 FROM Emp);
```

Q. WAQTD 2nd half of records of emp table.

```
select * From (select ROWNUM as second
               Half, emp.* From emp)
where secondHALF>(select Count(*)/2
                  From Emp);
```

Q WAQTD last records of the employee table

```
select * From(select ROWNUM as lastRec,
              emp.* From Emp)
where lastRec=(select max(ROWNUM)
               From Emp);
```

Q. WAQTD last 3rd record of the emp table.

```
select * From (select ROWNUM as lastthree,
               emp.* From emp)
where lastthree=(select max(ROWNUM)-2
                 From Emp);
```

Q WAQTD even number of records present in emp table.

```
select *
From (select RowNum as SLNO, emp.*
      From emp)
Where mod(SLNO,2)=0;
```

Q. WAQTD 9th maximum salary

→
```
Select *
From emp
where sal = (select max(sal)
             From emp
             where sal < (
```

```
Select *
From (Select RowNum as slno, emp.*
      From emp
```

* To Find Nth maximum and minimum sal:-

WAQTD 6th maximum sal:

→ Step 1:- In step1 we will remove duplicated or repeted salary and the sal will be sorted in desending order.

SAL	
950	7th
1820	5th
3000	2nd
5000	1st
1250	6th
3000	2nd
2000	3rd
1920	4th

```
Select Distict SAL
From Emp,
ORDer by SAL DESC.
```

o/p of Distict clause

SAL
950
1820
3000
5000
1250
3000
2000
1920

o/p of select Clause.

SAL
950
1820
3000
5000
1250
2000
1920

o/p of order by clause

SAL
5000
3000
2000
1920
1820
1250
950

Step2:- For the arragened salary in desending order assign SLNO by using RowNum

```
SELECT RowNum AS SLNO, SAL
FROM (SELECT DISTINCT SAL
      FROM Emp
      ORDER BY SAL DESC)
```

RowNum / SLNO	SAL
1	5000
2	3000
3	2000
4	1920
5	1820
6	1250
7	950

Step3:- Use the step2 in Form clause of outer query and use the where clause to get the expected result.

```
SELECT SAL
FROM (SELECT RowNum AS SLNO, SAL
      FROM (SELECT DISTINCT SAL
            FROM Emp
            ORDER BY SAL DESC))
WHERE SLNO = 6;
```

Final result table

SAL
1250

Q. WAQTD Top 3 maximum salaries

→
```
SELECT SAL
FROM (SELECT RowNum AS SLNO, SAL
      FROM (SELECT DISTINCT SAL
            FROM Emp
            ORDER BY SAL DESC))
WHERE SLNO IN (1,2,3);
```

Q. WAQTD Top 5 minimum salaries.

→
```
SELECT SAL
FROM (SELECT RowNum AS SLNO, SAL
      FROM (SELECT DISTINCT SAL
            FROM Emp
            ORDER BY SAL ASC))
WHERE SLNO <= 5;
```

Q. WAQTD Top 1 & 5 maximum salary
maximum salary.

→
```
SELECT SAL
FROM (SELECT RowNum AS SLNO, SAL
      FROM (SELECT DISTINCT SAL
            FROM Emp
```

cohere SLNO IN(1,3,5);

Q.

→ Select sal
FROM(Sa

WAQTD Bottom 3 minimum salaries.

→ SELECT SAL
FROM(Select ROWNUM ASSLNO, SAL
FROM (Select DISTINCT SAL
FROM emp
order by sal Desc))
cohere : count(SLNO) <= 3;