

# Important Topics for Interview

- \* Primary Key
- \* Foreign Key
- \* Statements in SQL (v.v.v.Imp)
  - DDL
  - DML
    - Insert
    - Update
  - DQL (Total topics)

## \* Joins

- \*\* Inner Join
- \* self Join

## \* Sub-Query

## \* Normalization

## \* Single Row Functions (v.v.v.Imp)

## \* Pseudo Columns

## \* Rownum

## \* Co-related SubQuery

## PL/SQL

- \* View
- \* Index
- \* Procedure
- \* Cursor

8/1/2020

## \* Why Co-related Sub-Query

- Correlated sub-Query is more efficient than sub-Query and Joins.
- We can find nth max and nth min salaries.

- 1) ST
- 2) N
- 3) CO

\* Why

→

\* To

\* Ent

\*

8/7/2020

# SQL

[SCDM 26]

\* types of Applications :-

- 1) Stand alone Application → ppt, Excel, Paint
  - 2) Web application → fb, insta, etc.  
    ↳ third party application  
    → will store in data-base
  - 3) Client server application →  
    ↳ mobile application.  
    → Internet required.
- ↳ There is involvement of third party.

\* What is Database ?

→ Data can be recovered at any time and any where. in any device.

\* To contact with database we need SQL.

\* Entity → Anything which has its existence

\* Attributes are nothing but properties.

08/07/2020

# SQL (Structured Query Language)

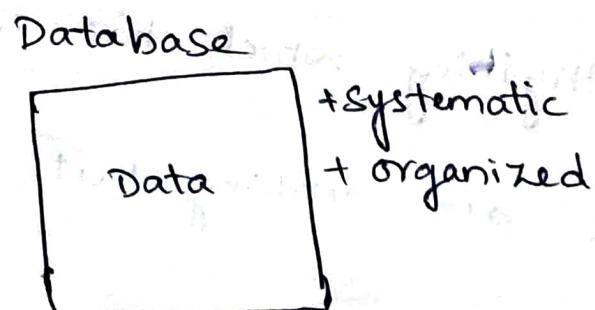
Data :- Data is a raw fact which describes attributes of an entity.

Database :- It is a place where data will be stored in systematic and organized manner.

\* Basic operations performed on database are:-

- 1) Create
- 2) Read
- 3) Update
- 4) Delete

\* Generally Known as "CRUD" Operations.



- \* Database Management System (DBMS) is Software.
- \* It is a software to maintain and manage data base.

→ The two important factors are:

- 1) Security
- 2) Authorization

→ Query language is used to communicate with ~~dated~~ DBMS.

→ To communicate with this software we need a language called Query Language.

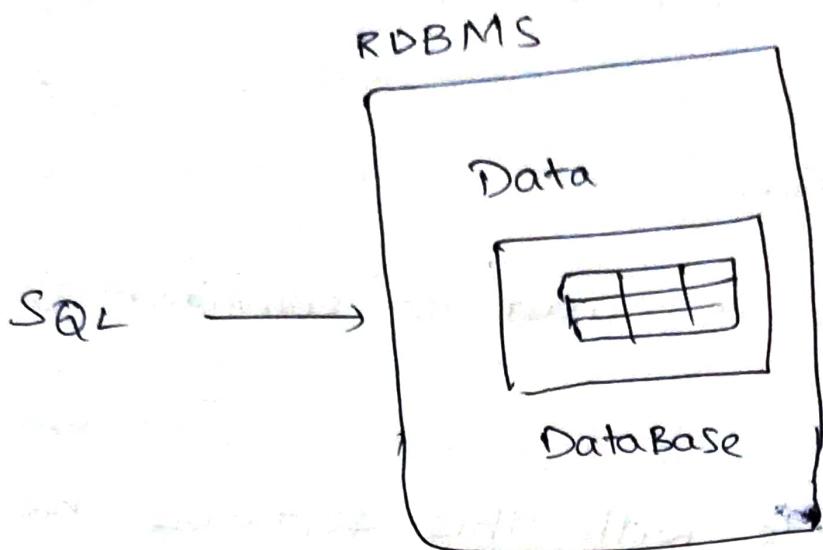
### Different types of DBMS Software

- \* Relational DBMS (RDBMS) → Most used & popular
- \* Hierarchical DBMS
- \* Network DBMS
- \* Object oriented DBMS
- \* Graph DBMS
- b \* ~~Graph DBMS~~ ER model DBMS
- 7 \* Document DBMS

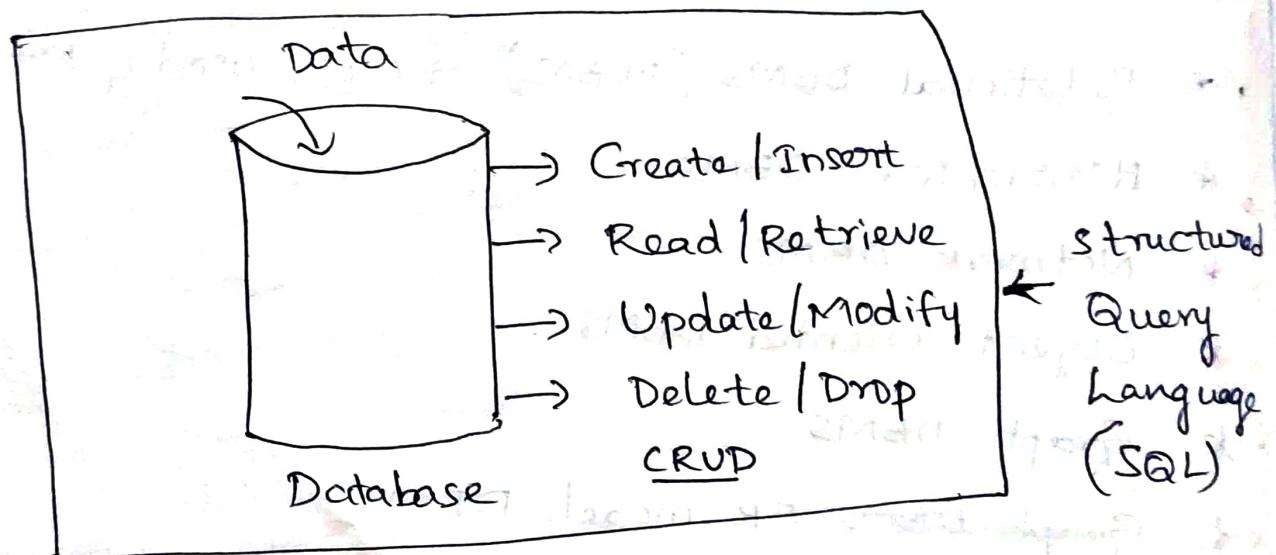
RDBMS + It stands for Relational Database

Management System.

- \* It is a type of DBMS software where data's will be stored in the form of relations or tables.
- \* To communicate with this software we need a language called Structured Query Language (SQL)



E.F. CODD rules  
 $\downarrow$   
 DBMS + RM  $\Rightarrow$  RDBMS



## Relational Model :-

- \* It is a concept which was designed by "EF Codd" who is a data scientist.
- \* In this we can store the data in the form of relations or tables.

\* We can also store Meta data in the form of tables.

\* Meta data : Data about the data.

DBMS



Relational model → Tables



Relational DBMS → Tables

\* Any DBMS that follows Relational model will becomes a Relational DBMS.

\* As in Relational model data stored in tables and same as in RDBMS.

\* Difference b/w Excel sheet & RDBMS

- |  |  |
|--|--|
| → It can't store meta data.                  | → we can store Meta data in RDBMS.       |
| → There is no relationship b/w data.         | → There is relationship b/w data.        |
| → no need particular language to communicate | → We need SQL to communicate with RDBMS. |
| → It does not follow relational model        | → It follows.                            |
| → It does not follow EF Cod rule             | → It follows.                            |
| → No Security & Authorization                | → Security & Authorization is there      |
| → Limited volume of data.                    | → Unlimited volume of data.              |

09/07/2020

Task

- 1) Diff b/w DB and DBMS?
- 2) Diff b/w DBMS and RDBMS?
- 3) What is relational model?
- 4) Who is EF Codd?

①

Database :- It is a collection of data, it can be words you write on a piece of paper or a digital file. Whereas

Database Management System (DBMS) :- DBMS is a piece of software that manages databases. and

- \* It can create, edit, delete databases, their tables and their data.

②

DBMS

- \* In this data is stored as a file.
- \* Relationship b/w two files are maintained programmatically.
- \* DBMS does not support Client Server architecture.
- \* It does not support distributed architecture.

RDBMS

- \* Data stored in a tabular form.
- \* Relationship b/w two tables or files can be specified at the time of table creation.
- \* It supports Client Server architecture.
- \* It supports distributed architecture.

- |   |  |
|---|--|
| * In DBMS there is no security of data.   | * In this there are multiple level of security, at logging level, at command level, at object level. |
| * Normalization is not available in DBMS. | * Normalization is available in RDBMS.   |

- (3) Relational Model :- Relational model represents how data is stored in relational databases.
- \* Relational database stores data in the form of tables.

- (4) EF Codd :- Edgar Frank "Ted" Codd
- \* He is a Computer scientist and creator of the relational databases model.

10/07/2020

Customer	Order No.	Date
John	12345	12/12/2020
Steve	12345	12/12/2020
John	12345	12/12/2020

- Tables :- Table is a logical organization of data which consists of Rows and columns.

Columns:-

- \* Columns is also referred as Attributes or fields.
- \* A column is used to represent one property of all the entities.

Rows :- Rows is also referred as Records or Tuples.

\* A row is used to represent all the properties of single entity.

Cell :- Cell is the smallest unit in table in which we store data. The intersection of rows and columns generate cells.

## Table Structure

# Attributes / columns / fields

ID	Name	Branch	Place
1	Virat	Ec	Banglore
2	Rohith	MEch	Mumbai
3	Rahul	Civil	Punjab

Row / Records /

Tuples

Cells

## Rules of E.F. CODD

- 1) The data stored in the cell must be a single value data.
- 2) In RDBMS we store everything in the form of tables including meta data. (The details about the data is meta data).

- 3) According to E.F. Codd we can store data in multiple tables, if needed we can establish connection b/w two tables using Key Attributes.
- 4) We can validate the data entered into the table in two steps.
- By Assigning datatypes
  - By Assigning Constraints.

Note: Here datatypes are mandatory whereas constraints are optional.

Null :- Null is a keyword which represents an empty cell / Nothing

\* In SQL 2 null and NULL are different.

## Datatypes

- \* Datatypes are used to determine what type or kind of data will be stored in a particular memory location.

## Datatypes in SQL

- 1) CHAR
- 2) VARCHAR / VARCHAR2
- 3) Number
- 4) Date
- 5) Large Object
  - a) Character Large Object (CLOB)
  - b) Binary Large object (BLOB)

\* (SQL is not a Case Sensitive).

### ① CHAR :

CHAR datatype can accept characters such as 'A-Z', 'a-z', '0-9' or special characters (#, \*, \$, ...)

ASCII (American Standard code for information interchange)  
A - 65  
Z - 90

a - 97

z - 122

Syntax: char(size)

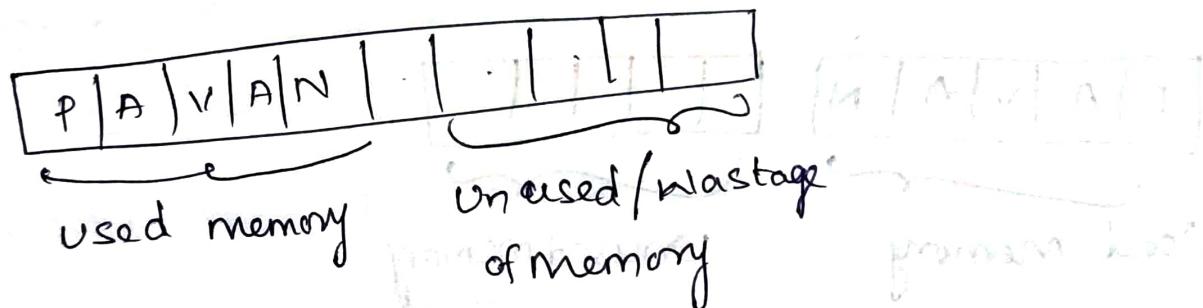
size - It is used to determine the no. of characters that we can store.

\* Whenever we mention char datatype we have to mention size for it.

\* The Max. size that we can store is 2000.

\* It is a type of "fixed length memory allocation".

Example :- char(10)



Disadvantage:- wastage of memory is high.

In order to overcome this disadvantage.

Introduced varchar.

13/07/20

⑧ VARCHAR

\* VARCHAR datatype can accept characters such as, 'A-z', 'a-z', '0-9' or special characters (#, \*, \$, ...)

Syntax :- VARCHAR (size)

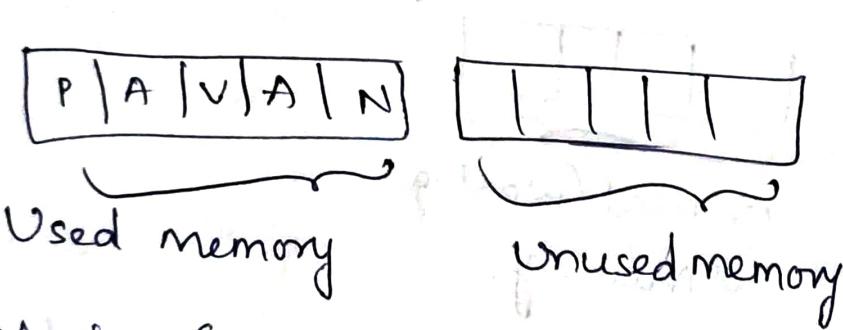
QUESTION

Size :- It is used to determine the No. of characters that we can store.

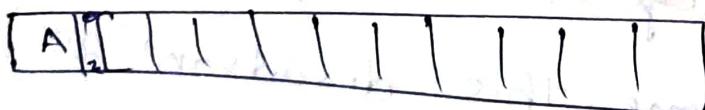
\* Whenever we mention Varchar datatype we have to mention size for it.

- \* The Max. size that we can store is 2000.
- \* It is type of " Variable length memory allocation".
- \* There is no wastage of memory in varchar.

Ex :- ① Varchar (10)



② Varchar (2000)



Varchar 2 :- It is just ~~upgradation~~ <sup>updated version</sup> of the Varchar

\* It can store upto 2000 to 4000.

Syntax :- Varchar (size)

③ Number: This datatype is used to store numerical values.

\* It can accept two arguments:-

- a) Precision
- b) Scale

Syntax: Number (Precision, [Scale])

Precision: It is used to determine the digits we are going to store in numerical place.

Scale: It is used to determine the no. of digits we are going to store in decimal place within the precision.

\* The max. precision we can store is 38.

\* The max. scale we can store is 127.

Ex- ① When P > S

② Number (5)  $\pm 99999$

③ Number (4,0)  $\pm 9999$

④ Number (7,2)  $\pm 99999.99$

⑤ Number (8,4)  $\pm 99999999$

- ii) when  $P = S$
- ⑥ Number (4,4)  $\dots \dots \pm 0.9999$
- iii) when  $P < S$
- ⑦ Number (2,5)  $\dots \dots \pm 0.00099 \quad 5-2=3$
- ⑧ Number (2,7)  $\dots \dots \pm 0.0000099 \quad 7-2=5$
- ⑨ Number (4,8)  $\dots \dots \pm 0.00009999 \quad 8-4=4$
- ⑩ Number (38,127)  $\dots \dots \pm 0.000000009999 \quad 127-38=89$

### Assignment

- 1 Number (2,8)  $\dots \dots \pm 0.00000099 \quad 8-2=6$
- 2 Number (7,2)  $\dots \dots \pm 99999.99$
- 3 Number (8,2)  $\dots \dots \pm 999999.99$
- 4 Number (5,5)  $\dots \dots \pm 999999.$
- 5 Number (4,8)  $\dots \dots \pm 0.00009999 \quad 8-4=4$
- 6 Number (3,8)  $\dots \dots \pm 0.00000999 \quad 8-3=5$
- 7 Number (8,3)  $\dots \dots \pm 99999.999$
- 8) DATE :-

Syntax :- DATE

The two oracle specified Date formats are

a) 'DD-MON-YY'

b) 'DD-MON-YYYY'

\* Date should always be enclosed within single quotes.

Example: ① 'DD-MON-YY'

13-JUL-20

② 'DD-MON-YYYY'

'15-AUG-1947'

## ⑤ Large Object

a) Character Large object (CLOB)

\* This is used to store the characters upto 4GB of size.

Syntax - CLOB

b) Binary Large object (BLOB):

\* This is used to store binary numbers of

Images, Videos, files etc... upto 4GB of size.

Syntax - BLOB

## ① Diff. CHAR and VARCHAR

Ans.

CHAR	VARCHAR
------	---------

- 1) It can accept characters.
- 1) It is also accepts characters.
- 2) It can store upto 2000.
- 2) It can also store upto 2000.
- 3) Wastage of Memory is high.
- 3) ~~No~~ No wastage of memory.
- 4) Syntax is CHAR(size).
- 4) Syntax is VARCHAR(size)

## ② Diff. VARCHAR and VARCHAR2

- \* VARCHAR can store only upto 2000.
- \* VARCHAR2 can store the memory upto 4000.
- \* VARCHAR2 is an update version of VARCHAR.
- \* Syntax is same for both.

- \* There are some disadvantages of datatype so we are using constraints.
- \* Suppose if we give number datatype to EID it only allows numbers and it may be repeated. So in order to not to repeat we use constraint.

EID	ENAME	PHNO	SALARY
1	Varun	1234567890	40,000

14/07/2020

Constraints :- Constraints are the conditions that are assigned to a particular column to validate the data.

### Types of Constraints

- 1) Unique
- 2) Not Null
- 3) Check
- 4) Primary Key
- 5) Foreign Key

① Unique :- Unique is a constraint which is assigned to a particular column which cannot accept repeated or duplicate values.

② NOT NULL :- Not null is a constraint which is assigned to a particular column which cannot be null or which are mandatory.

③ CHECK :- Check is a constraint which is assigned to a particular column which adds condition to validate the data.

to a particular column for extra validations.

\* Check constraint is assigned with a condition, if the condition is true the value gets accepted else rejected.

Eg:- 1. Check (Length (PHNO)=10)

2. Check (Sal>0)

15/01/2020

④ Primary Key:- Primary Key is a constraint which is used to assign to a column to identify a record uniquely from the table.

### Characteristics of Primary Key

- ① We can have only one primary key in a table.
- ② It cannot accept repeated or Duplicate values.
- ③ It Cannot accept null.
- ④ It is a combination of unique and not null.
- ⑤ Primary key is not mandatory but recommended to have one in table.
- ⑥ Foreign key:- It is a constraint which is used to establish the connection between two tables.

## Characteristics of Foreign Key

- 1) We can have 'n' no. of foreign keys in a table.
- 2) It can accept • repeated or duplicate values.
- 3) It can accept null.
- 4) It is not a combination of unique and not null.
- 5) It is present in child table but actually belongs to Parent table.
- 6) It is also referred as "Referential Integrity Constraint".

Note:

\* Only primary key can travel to another table when it does travel, it becomes foreign key.

EMP

Primary Key	NN	NN+NC	NN+NC	P-K	F-K
EID	EName	PHNO	SAL	DNo	CID
1	PAVAN	1234567890	30000	10	111
2	VARUN	2134567890	40000	20	222
3	RAVAN	3124567890	50000		

Child

DEPT

P-K	DNo	DName	Loc
	10	HR	US
	20	Sales	Delhi

Customer

P-K	CNAME
CID	CNAME
111	RAM
	SHIV

① 16/07/2020  
D/w b/w Primary Key and Foreign Key

Primary Key

Foreign Key

- ① It is used to assign to a column to identify a record uniquely from the table.
- ② Only one we can have primary for any table
- ③ It cannot accept repeated or duplicate values.
- ④ It cannot accept null.
- ⑤ It is a parent
- ⑥ It is a combination unique and not null.
- ⑦ It cannot travel to another table
- ① To establish connection b/w two table.
- ② One More than one we can have.
- ③ It allows repeated or duplicate values.
- ④ It can accept null.
- ⑤ It is child table.
- ⑥ It is not a combination of unique and not null.
- ⑦ It can travel to another table.
- ② D/w primary key & unique constraint?

# SQL Statements / Languages

## 1) Data Definition Language (DDL)

- Create
- Rename
- Alter
- Truncate
- Drop

## 2) Data Manipulation Language (DML)

- Insert
- Update
- Delete

## 3) Transaction Control Language (TCL)

- Commit
- Savepoint
- Rollback

## 4) Data Control Language (DCL)

- Grant
- Revoke

## 5) Data Query Language (DQL)

- Select
- Projection
- Selection
- Joins

# Data Query Language (DQL)

- \* This statement is used to retrieve the data from database.
- \* There are 4 statements:
  - 1) Select
  - 2) Projection
  - 3) Selection
  - 4) Joins

① Select:- This statement is used to retrieve the data from database and display it.

② Projection:- This statement is used to retrieve the data from database by selecting only columns.

\* All the values in the column will be selected by default.

③ Projection Selection:- This statement is used to retrieve the data from database by selecting both columns as well as records.

④ Joins:- This statement is used to retrieve the data from multiple tables simultaneously.

Projection :- This statement is used to retrieve the data from database by selecting only column.

\* All the values in the column will be selected by default.

Syntax :-

Select \* / [Distinct] Column-Name / Expression [ALIAS]  
From Table-Name;

Order of execution:

- 1) From
- 2) select

① Note:-

- 1) From clause starts the execution.
- 2) For from clause we can pass table name as an argument.
- 3) The job of from clause is to go to the database and search for the table and put the table under execution.
- 4) Select clause executes after the execution of from clause.

- ⑤ For select clause we can pass asterisk (\*) column name and expression as an ~~arg~~ argument.
- ⑥ The job of select clause is to go to the table which is under execution and select the data and display.
- ⑦ Select clause is responsible for the result table.

① Write a Query to display all the names of the Student from student table.

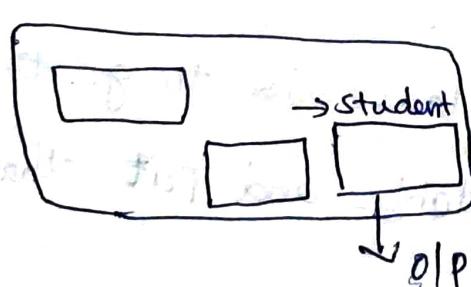
Ans:

1st

2 → select Sname

1 → From student;

2nd



# Student

SID	SNAME	BRANCH	PER	ADD	YOP
1	PUJA	TCE	73	PUNE	2020
2	Pavan	ECE	72	HYD	2020
3	Sushma	ECE	60	KAR	2013
4	PRINCE	EE	75	HYD	2019
5	Ashwini	ES	80	GUJ	2019
6	ARJUN	Mech	80	Kerala	2018
7	Varun	Mech	55	KAR	2017

3rd

DIP of Select

SNAME
PUJA
PANAV
SUSHMA
PRINCE
ASHWINI
ARJUN
VARUN

② WAQTD Name and percentages of all the students.

Ans Select sname , PER

From student ;

③ WAPTD Name and branch of all the students.

Ans:  
Select sName, branch  
From emp;

④ WAPTD Name and per and branch of all the Students.

Ans:  
Select sName, per, branch  
From emp;

⑤ WAPTD Name, SID, Branch and percentage of all the students.

Ans:  
Select sName, SID, branch, per  
From emp;

Task #2: If no ID is present in the table then what should be done? (Q)

Asterisk (\*) :- It is used to determine "to select all the columns".

Semi-colon (;) :- It is used to determine "the end of the statement".

\* In order to know all the tables present,

SQL > Select \*  
from tab;

T Name	Tabtype	Clustered
DEPT	Table	
EMP	Table	
BONUS	Table	
SALGRADE	Table	

\* To describe an employee table

SQL > desc emp;

Name	Null?	Type
EMP NO	Not null	Number(4)
ENAME		Varchar2(10)
JOB		Varchar2(9)
MGR		Number(4)
HIREDATE		Date
SAL		Number(7,2)
COMM		Number(7,2)
DEPTNO		Number(2)

\* To describe an Dept table

SQL > desc dept;

Name	Null?	Type
Dept no	NOT null	Number(6)
DName		Varchar2(14)
Loc		Varchar2(13)

\* To clear the screen use command

→ clear screen (or) CL SCR

\* In order to increase size of Pages and Line  
use Command.

→ Set Pages any number Lines any number

① Write a query to display all the details from  
the employee table.

Ans:

Select \*  
From Emp;

EmpNo.	Ename	Job	Mgr	HireDate	Sal	Comm	Dept No
7369	smith	Clerk	7902	17-Dec-80	800	300	20

② What names of all the Employees?

Ans:  
Select Ename  
From Emp;

Ename  
smith

Allen  
WARD

③ WAQTD Name and annual Salary given to all the employees.

Sol:-  
Select Ename, salary \* 12  
From EMP;

④ WAQTD Employee ID and Department no?

Sol:-  
Select EMPLNO, Deptno  
From EMP;

⑤ WAQTD Ename and hiredate of the employee

Sol:-  
Select Ename, sal, Sal + Sal \* 25/100  
From EMP;

⑥ WAQTD Ename and sal, sal with deduction of 12% for all the Emp.

Sol:-  
Select Ename, sal, Sal - Sal \* 0.12  
From EMP;

\*Trace the Query Means explain the process.

Expression :- A statement which gives us result is known as expression.

\* Expression consists of 2 types.

1) Operand

2) Operators (+, -, \*, /)

\* Operand consists of two types

1) Column Name

2) Literals (Direct values)

\* Literals are of three types

1) Number literal

2) Character literal

3) Data literal

Ex:-

Sal  
Operand

\*  
operator

12  
operand

Column name

(+, -, \*, /)

Literals (direct values)

Note:-

Character literal and data literal should be enclosed within single quotes.

Alias: Alias is an alternative name given to A column or An expression in the result table.

- \* Alias name can be used with or without using 'AS' keyword.
- \* Alias name should be a single word or A String enclosed within double Quotes.

### Task

1) WAPTD SAL as salary, Hiredate as Date of Joining

Ans: Select Sal as Salary, Hiredate as "Date of joining"

From emp;

2) WAPTD Name, Sal of the employee along with their annual salary.

Ans: Select ename as Name, sal as "sal of the employee"  
Sal\*12 as "Annual salary":

From EMP;

3) WAPTD Ename and Job for all the employee with their half term salary.

Ans: Select Ename, Job, Sal\*12 as "Half term sal":

From emp;

③ WAPTD All the details of the employees along with annual bonus of 2000.

Ans: Select Ename, Empno, Job, MGR, Hiredate,  $sal * 12 + 2000$  as "annual bonus", comm, Dept no  
From emp;

(~~Ans~~) select \*, ~~sal \* 12 + 2000~~ as "Annual bonus"

④ WAPTD Name, salary and salary with a Hike of 10%.

Ans: Select Ename as Name, sal as salary,  $sal + sal * 0.10$  as "Hike 10%".

From emp;

⑤ WAPTD Name and salary with deduction of 25%.

Ans: Select Ename as Name, ~~sal - sal \* 0.25~~ as "deduction 25%".

From emp;

⑥ WAPTD Name and salary with monthly Hike of 5%.

Ans: Select Ename as Name, ~~sal + 50~~ as "monthly Hike 5%"  
From emp;

⑦ WAPTD Name and Annual salary with deduction of 10%.

Ans: Select Ename as Name, ~~sal \* 12 - 1000~~ as "Annual salary with deduction of 10%"

"deduction 10%".

From emp;

Note:- Don't provide alias for column name if they don't ask and should be provided for the expression. Ex:  $\text{empno}$

⑧ WAPTD total salary given to each employee ( $\text{sal} + \text{comm}$ )

Ans: ~~Select~~  $\text{Select Empno, Ename, Job, MGR, Hiredate, Sal * 12 as "Annual Salary", comm, Dept no}$   
~~From emp;~~ From emp;

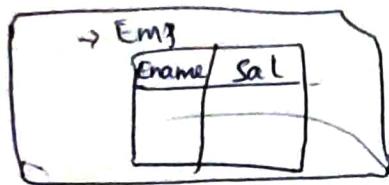
⑨ WAPTD details of all the employees along with annual salary.

Ans: ~~Select~~  $\text{Select Empno, Ename, Job, MGR, Hiredate, Sal * 12 as "Annual Salary", comm, Dept no}$   
 $(a) \text{Select EMP.*}, \text{Sal} * 12$   
~~From emp;~~ From emp;

⑩ WAPTD Name and designation along with 100 penalty in salary.

Ans: ~~Select~~  $\text{Select Ename, Job, Sal - 100 as "Penalty"}$

⑪ ~~WAPTD~~ Name and designation along with 100 penalty in salary with standard pay to non-salaried with lower ID & From emp;



Emp

Ename	sal
A	200
B	100
C	300
D	250

OR of select

Ename	sal	sal * 12
A	200	2400
B	100	1200
C	300	3600
D	250	-

③ Ans:

Select Emp.\* ,  $\text{Sal}^{*12}$  "Annual Bonuses"  
From emp;

Distinct :- To remove repeated values or duplicate values in result table we use distinct clause.

→ For distinct clause we can pass Column Name or an expression as an argument.

→ Distinct clause should be used as the first argument in the select clause.

→ We can pass multiple columns for distinct clause.

→ It removes the combination of duplicate values from all the columns.

① WAPTD Different per from student table  
student table

SID	SName	Branch	Per
1	A	EC	70
2	B	CS	65
3	C	TC	80
4	D	CS	80
5	E	EC	70
6	F	ME	65

Ans:-

Select distinct Per

From student;

opf of select

70  
65  
80

opf of  
from  
PER  
70  
65  
80  
80  
70  
65

② WAPTD Different branch and per from student table.

Ans:- Select distinct branch, Per  
from student;

Branch	Per
EC	70
CS	65
TC	80
CS	80
EC	70
ME	65

- \* If branch is same it won't remove directly  
it will also check with percentage.
- \* If both are matched then only it remove.

Branch	PER
EC	70
CS	65
TC	80
CS	80
ME	85

③ WAPTD different deptno from emp table

Ans:-  
Select distinct dept no  
From emp;

④ WAPTD different salary from emp table

Ans:-  
Select distinct sal.  
From emp;

⑤ WAPTD different designations from emp table

Ans:-  
Select distinct job from emp;

⑥ WAPTD all the details of emp's along with

14% deduction in sal.

Ans:-  
Select emp.\* , sal - sal \* 0.14 as deduction  
From emp;

⑦ WAPTD different designation and sal from emp table?

Ans: Select distinct job, sal  
from emp;

⑧ WAPTD SAL as salary and job as designation along with annual salary.

Ans: Select salary as "Salary", job, designation, sal\*12 "Annual salary"  
From emp;

⑨ WAPTD EName from emp table

Ans: Select ename  
from emp;

⑩ WAPTD EName from emp table if they are working in Dept 20.

Ans: Select Ename  
From emp  
where Deptno = 20;

Note: where clause is used to filter the records.

(11) WAPTD only different salaries given to employees.

Ans:-  
= Select distinct sal  
from emp;

(12) WAPTD the different designations that are present in emp table.

Ans:-  
= Select distinct job  
from emp;

(13) WAPTD different dept no as well as salaries that are present in the table.

Ans:-  
= Select distinct dept no, sal  
from emp;

(14) WAPTD all the details of the employee along with his annual salary.

Ans:-  
= Select emp.\* , sal\*12 "Annual salary"  
From emp;

(15) WAPTD all the details of employee along with 10% hike in the salary.

Ans:-  
= Select emp.\* , sal+sal\*0.10 "Hike  
From emp;

## Selection :-

where clause :- This is used to filter the records.

### Note :-

- \* For where clause we can pass filter condition as an argument.
- \* where clause executes Row by Row.
- \* where clause <sup>executes</sup> after the execution of from clause.
- \* We can pass multiple condition for where clause using logical operators.

### Syntax :-

Select \* / [Distinct] Column-Name / Expression [Alias]

From Table-Name

Where <filter-condition>;

### Order of execution

- 1) From
- 2) Where
- 3) Select

- ① What Ename from emp table if they are working in Dept 20.

Anst 1<sup>st</sup>

Select Ename

From emp

where DNO = 20

2<sup>nd</sup>

Emp o/p of From

Ename	SAL	DNO
Dinga	1000	20
Pooja	2000	10
Kohli	1500	20
Alen	3000	10
Kalyan	2500	30
Sravan	2000	20

3<sup>rd</sup>

o/p of where

Ename	SAL	DNO
Dinga	1000	20
KOHLI	1500	20
Sravan	2000	20

4<sup>th</sup>

o/p of select

Ename
Dinga
KOHLI
Sravan

② WAPTD sal of Emp whose name is smith

Ans:- Select sal

From emp  
where Ename = "SMITH"

③ WAPTD The annual salary of the employee whose name is smith

Ans:- Select sal \* 12 "Annual salary"

From emp  
where Ename = "SMITH"

④ WAPTD name of the employees working as clerk.

Ans:- Select ename

From emp

where Job = "CLERK"

⑤ WAPTD salary of the employees who are working as salesman

Ans:- Select Sal from emp where Job = "SALESMAN"

⑥ WAPTD Details of the emp who earns more than 2000

Ans:-

Select \* from emp

Where sal > 2000

⑪ WAPTD ename and hiredate if they are hired before 1981.

Ans: Select ename, hiredate

From emp

01-JAN-1981

where hiredate < ~~1981~~

<=(01) - 31-DEC-80;

⑫ WAPTD Details of the employees working as manager.

Ans: Select \*

From emp

where job = 'MANAGER'

⑬ WAPTD Name and salary given to an employee if

employee earns a Commission of Rupees 1400.

Ans: Select ename, sal

From emp

where comm = 1400;

⑭ WAPTD details of employees having Commission more than salary.

Ans: Select \*

From emp

where comm > "SAL";

(15) WAPTD empro of employees hired before the year 87.

Ans:

Select empro

From emp

Where hiredate < '01-JAN-87'

(16) WAPTD Details of employees working as an Analyst.

Ans:

Select \*

From emp

Where job = 'ANALYST'

(17) WAPTD Details of emps earning more than 2000 rupess per month.

Ans:

Select \*

From emp

Where SAL > 2000

## Operators :-

- ① Arithmetic operators (+, -, \*, /)
- ② Comparison operators (=, !=)
- ③ Relational operators (<, >, <=, >=)
- ④ Logical operators (AND, OR, NOT)
- ⑤ Concatenation operators (||)
- ⑥ Special operators (IN, NOT IN, Between, NOT between, LIKE, NOT LIKE, IS, IS NOT)
- ⑦ Sub-Query operators (ALL, ANY, EXISTS, NOT EXISTS)

## Logical Operators

### And Operator

### Binary Multiplication

- \* And operator returns true if both the conditions are true.
- \* And operator should always be used between -n conditions.

AND

I/P		O/P
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

I/P		O/P
c1	c2	Result
F	F	F
F	T	F
T	F	F
T	T	T

0 - F

1 - T

## OR Operator

=

## Binary Addition

=

\* OR Operator returns true if any one of the conditions is satisfied (true).

\* OR operator should always be used between conditions.

OR

I/P		O/P
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

I/P		O/P
c1	c2	Result
F	F	F
F	T	T
T	F	T
T	T	T

Not operator :- It is used for negation

iLP	oLP
0	1
1	0

iLP	Result
F	T
T	F

① WAQTD EName if the emp is working in Dept<sup>n</sup> 20 and earning sal more than 1500

Ans:-

Select ename  
From emp

Where dept<sup>n</sup> = 20 AND sal > 1500 !

② WAQTD Ename if an emp is working as sales man or earning more than 1500.

Ans:-

Select ename  
From emp

Where job = 'SalesMan' OR sal > 1500 !

③ WAQTD Ename except who is working in Deptno 20

Ans:-

Select ename

From emp

Where NOT deptno = 20 !

## Assignment: Logical operators

- 1) Write Details of the employees working as clerk and earning less than 1500.

Ans:-

Select \*

From emp

Where Job = 'CLERK' AND Sal < 1500;

- 2) Write Name and hiredate of the employees working as manager in Dept 30.

Ans:-

Select ename, hiredate

From emp

Where Job = 'MANAGER' AND Deptno = 30;

- 3) Write Details of the emp along with annual salary if they working in Dept 30 as salesman and their annual salary has to be greater than 14000.

Ans:-

Select Emp.\* , Sal \* 12 "Annual Salary"  
From emp

Where Deptno = 30 AND Job = 'Salesman' AND sal \* 12 > 14000;

(4) WAGTD all the details of the employee  
- King in dept 30 or as Analyst

Ans:

Select \*

From emp

where dept = 30 OR Job = 'ANALYST'

(5) WAGTD names and deptno, job of emps working  
as clerk in dept 10 or 20.

Ans:

Select ename, deptno, ~~Job~~ Job

From emp

where Job = 'clerk' And (dept=10 or deptno=20)

(5) WAGTD names of the employees whose salary is less  
than 1100 and their designation is clerk?

Ans:

Select ename

From emp

where sal < 1100 and Job = 'CLERK';

(6) WAGTD Name and sal, Annual sal and dept no if  
deptno is 20 earning more than 1100 and annual salary  
Exceeds 12000.

Ans:

Select ename, sal, sal\*12 "Annual Salary", deptno

From emp

where deptno = 20 and Sal > 1100 and Sal \* 12 > 12000,

- ⑦ WAGTD empno and names of the employees working as manager in dept 20.

Sol:

Select empno, ename

From emp where job = 'MANAGER'

Where

Job = 'MANAGER' and deptno = 20,

- ⑧ WAGTD details of employees working in Dept 20 or 30.

Ans:

Select \*

From emp

where deptno = 20 or deptno = 30,

- ⑨ WAGTD Details of employees working as analyst in dept 30.

Ans:

Select \*

From emp

where Job = 'ANALYST' and deptno = 30;

Q:- No mouse selected

- ⑩ WAGTD Details of employee working as president with salary of rupees 4000.

Ans: Select \*  
From emp  
where Job = 'PRESIDENT' and sal = 4000;

Q1P  $\rightarrow$  No rows selected

(12) WAPTD details of employees working as clerk  
or manager in dept 10.

Ans: Select \*  
From emp  
where (Job = 'clerk' or Job = 'MANAGER') and deptno = 10;

(13) WAPTD Names of employees working in dept 10, 20,  
30, 40.

Ans: Select ename

From emp  
where deptno = 10 or deptno = 20 or deptno = 30  
or deptno = 40;

(14) WAPTD Details of employees with Empno 7902, 7839

Ans: Select \* and check marks

From emp  
where empno = 7902 or empno = 7839

## ⑥ Special Operators

### IN Operator

- \* IN operator is a multi valued operator in which we can pass multiple values at RHS. i.e., IN operator can accept multiple values at RHS.
- \* IN operator returns true, if any one of the condition is satisfied.
- \* IN operator allows the value present at LHS to be compared with all the values present at RHS.

### Syntax

column-name/ expression IN (v<sub>1</sub>, v<sub>2</sub>, ..., v<sub>n</sub>)

Ex:- 100 IN (50, 100, 150, 200)

$$\left. \begin{array}{l} 100 = 50 \text{ F} \\ 100 = 100 \text{ T} \\ 100 = 150 \text{ F} \\ 100 = 200 \text{ F} \end{array} \right\} \text{True}$$

① <sup>emp working</sup> W.A.Q.T.D. & Details of Dept 10, 20, 30, 40, 50, to

Ans

Select \*  
From emp

where Deptno IN (10, 20, 30, 40, 50, 60, 70)

- ② What Ename and job of emp's who are working as Manager or salesman.

Ans: Select ename, Job

From emp

where job IN ('Manager', 'salesman');

Not in: Not in operator is similar to IN operator but it rejects the value instead of selecting it.

Syntax: Column-name / expression NOT IN (v<sub>1</sub>, v<sub>2</sub>, ..., v<sub>n</sub>);

Ex: ① What Emp's name excluding the emp's working in dept 10 or 20.

Ans: Select ename

From emp

where Deptno NOT IN (10, 20)

EName	DNo
A	10
B	20
C	30
D	20
E	30

10 NOT IN (10, 20)

10 = 10T

10 = 20F

True - Not - False

where Deptno IN (10, 20, 30, 40, 50, 60, 70);

- ② WAQTD Ename and job of emp's who are working as Manager or salesman.

Ans: Select ename, Job

From emp  
where job IN ('Manager', 'salesman');

Not in: Not in operator is similar to IN operator but it rejects the value instead of selecting it.

Syntax: Column-name / expression NOT IN (v<sub>1</sub>, v<sub>2</sub>, ..., v<sub>n</sub>);

Ex: Open WAQTD Emps name excluding the emp's working in deptno 10 or 20.

Ans: Select ename  
From emp

where Deptno NOT IN (10, 20)

EName	DNo
A	10
B	20
C	30
D	20
E	30

10 NOT IN (10, 20)

10 = 10 T

10 ≠ 20 F

True - Not - False

② WAQTD Emp's Name if they are working in dept 20, 30 and Hired After 1980.

Ans: Select ename

From emp

Where deptno NOT IN (10, 30) AND HiredDate > '31-DEC-80'  
deptno IN (20, 30)

③ WAQTD Details of emp's if they are working as president, Manager, salesman.

Ans:

Select \*

From emp

Where job IN ('PRESIDENT', 'MANAGER', 'SALESMAN')

④ WAQTD Names and deptno, job of emps working as clerk in dept 10 or 20.

Ans:

Select ename, deptno, Job

From emp

Where job IN 'CLERK' AND (Deptno=10 OR Deptno = 20)

⑤ WAQTD details of employees working as clerk or Manager in dept 10.

Ans:

Select \*

From emp

where Job IN ('CLERK', 'MANAGER') AND Deptno  
= 10;

- (13) WAPTD Names of employees working in dept  
10, 20, 30, 40.

Ans:- Select Rname

From emp

where deptno IN (10, 20, 30, 40);

- (14) WAPTD Details of employees with empno 7902,

7839

Ans:- Select \*

From emp

where empno IN (7902, 7839);

- (15) WAPTD details of employees working as mana  
-ger or salesman or clerk.

Ans:- Select \* From emp

From emp

where Job IN ('Manager', 'Salesman', 'clerk');

- (16) WAPTD Names of employees Hired after 81 and  
before 87.

Ans:- Select ename

From emp

where hiredate between '01-JAN-82' and  
'31-DEC-86';

Q) WAPTD ename, sal if emp's are earning salary between 1250 and 3000.

Ans:-  
Select ename  
From emp

Sal  $\geq 1250$  and Sal  $\leq 3000$ ;

where sal between (1250, 3000);

(or)

where sal between 1250 and 3000;

## Between Operator

\* Between operator is used whenever we have ranges.

\* Between operator works including the ranges.

\* The range cannot be inter changed.

Syntax :- Column-name / Expression Between lower-range and higher-range;

Q) WAPTD Ename and sal and comm if emp's earning comm in the Range of 300 to 1000

Ans:-  
Select ename, sal, comm

From emp

Where comm between 300 and 1000;

## Not between

\* Not between is similar to between operator but it rejects the value instead of selecting it.

### Syntax :-

Column-name/expression Not between lower-range and higher range

- ① WAPTD Emp's Name and sal if emp's are earning Sal less than 1250 and emp's who are earning Sal more than 3000.

Ans:-

Select ename, sal  
From emp

where sal not between 1250 and 3000;

- ② WAPTP Ename, Comm if they are Comm less than 500 and emp's earning Comm more than 700.

Ans:-

Select ename, comm

From emp

where comm is not between 500 and 700;

③ WAPTD Ename, hiredate who were hired after 1981 and before 1987.

Ans:-

Select ename, hiredate

From emp

where hiredate between '01-JAN-1982' and

'31-DEC-86'

④ WAPTD Ename who is not earning commission.

Ans:-

Select ename

From emp

where comm is null;

Is Operator - Is operator is used only to

Compare with null.

Syntax:- Column-name / expression is Null;

① WAPTD ename if they are not earning any

Commission.

Ans:-

Select ename

from emp

From emp

where comm is null;

② WAP TO find ename who doesn't get any salary.

Ans:-

Select ename

From emp

where sal is null;

IS not operator: IS not operator is similar to is operator but it rejects the value instead of selecting it.

Syntax: Column-name/ expression is not null;

Between

NOT between    Assignment    Questions

④ WAP TO find ename, Hiredate who were hired in the year 1982.

Ans:-

Select ename, Hiredate

From emp

Where hiredate between

'01-JAN-82' and

'31-DEC-82'

⑤ WAGTD ename, hiredate except who were hired in the year 1980.

Ans: Select ename, hiredate

From emp  
where hiredate not between '01-JAN-80' and '31-DEC-80'

⑥ WAGTD Ename and hiredate if they are working in Dept 10 or 30 and Hire in the year 1981.

Ans: Select ename, hiredate

From emp  
where deptno in (10, 30) and hiredate between '01-JAN-81' and '31-DEC-81'

⑦ WAGTD Ename, hiredate who were hired after 1980 but before 1987 and working as manager or salesman.

Ans: Select ename, hiredate

From emp  
where hiredate between '01-JAN-81' and '31-DEC-86'  
and job in ('Manager', 'Salesman')

## IS NOT & IS NOT

- ① WAPTD Ename and comm of emp's if they are earning comm

Ans:- Select \* from emp  
where comm is not null;

From emp

where comm is not null;

- ② WAPTD details of emp's who are not reporting to any manager (MGR)

Ans:-

Select \*

From emp

where MGR is null;

- ① WAPTD Ename and hiredate if the emp is hired in the year 1982.

Ans:-

Select ename, hiredate

From emp

where hiredate between '01-JAN-82' and  
'31-DEC-82';

- ② WAPTD ename and hiredate if the emp's are not hired in the year 1981.

⑥ WAPTD empno of emp's whose empno starts at 7499 and ends at 7788.

Ans: select empno

From emp

where empno between 7499 and 7788.

Like Operator: Like operator is used when

ever we need to "Match the Pattern".

Syntax: column-name/expression like "Pattern-to-match"

\* To achieve the pattern matching we use special

-al characters such as

1) Percentile (%) :- It can accept any character,

Any no. of times or no character.

2) Underscore (\_) :- It can accept any character but only once.

① WAPTD ename only if they emp's having character 'A' in their name.

Ans: Select ename

From emp  
where ename like '%.A.%'

- ② WAQTD ename only if the emp's having character 'A' in the first place of their name.

Ans: Select ename

From emp  
where ename like 'A%.%'

- ③ WAQTD ename if the emp is having character 'A' in the second place.

Ans: select ename

From emp  
where ename like '%.A%.%'

- ④ WAQTD details of emp's having 'A' in 2nd last place.

select ename

From emp  
where ename like '%.%.A%.%'

Not like :- It is similar to like operator but rejects the value instead of selecting it.

Syntax :- Column-name/expression Not like 'Pattern-to-match'

- ① Write ename except who are having character 'A' in their name.

Ans :- Select ename

From emp

where ename not like '%.A.%'

- ⑥ Write ename's of emp's having 'E' in the four-th place.

Ans :- Select ename

From emp

where ename like '%.%.%.E.%'

- ⑦ Write ename who are having character 'A' in the first place and 's' in the last place.

Ans :- Select ename

From emp

where ename like 'A%.' and '%s'

(or) note 'A%.';

ename like '%s.'

## Concatenation Operator (||)

\* This operator is used to join the given two strings.

Ex:- select ename || sal  
from emp;

O/P Smith800

Allen1600

ward1250

Want output like "Mr. Smith, Rs. 800"

(1) Want the output in the following format.

Ans:- a) "MR. ABC Your salary is RS. XYZ"

Ans:- select 'MR.' || Ename || ' Your salary is RS.'  
From emp;

(2) "MR. SMITH Your salary is RS. XYZ"

Ans:-  
Select 'MR.' || Ename || ' Your salary is RS.'  
From emp;

where ename in ('smith');

③ 'MR. King Your designated as President'

Ans: Select 'MR.' || Ename || 'Your designated as.' || job

From emp

where ename in ('King');

(or)

job = 'President';



④ 'MR. X You have been promoted as MANAGER'

Ans: Select 'MR.' || Ename || 'You have been Promoted

With promotion with job as ' || job'

From emp

where job = 'MANAGER'

⑤ 'MR. Y Your Deptno is 20'

Ans: Select 'MR.' || Ename || 'Your Deptno is ' || Deptno

From emp

where Deptno = 20;

⑥ 'MR. A Your SAL is RS.B and UR annual salary is RS.C.'

Ans: Select 'MR.' || Ename || 'Your salary is RS.' || sal ||

From emp

'Your annual salary is' || sal ||

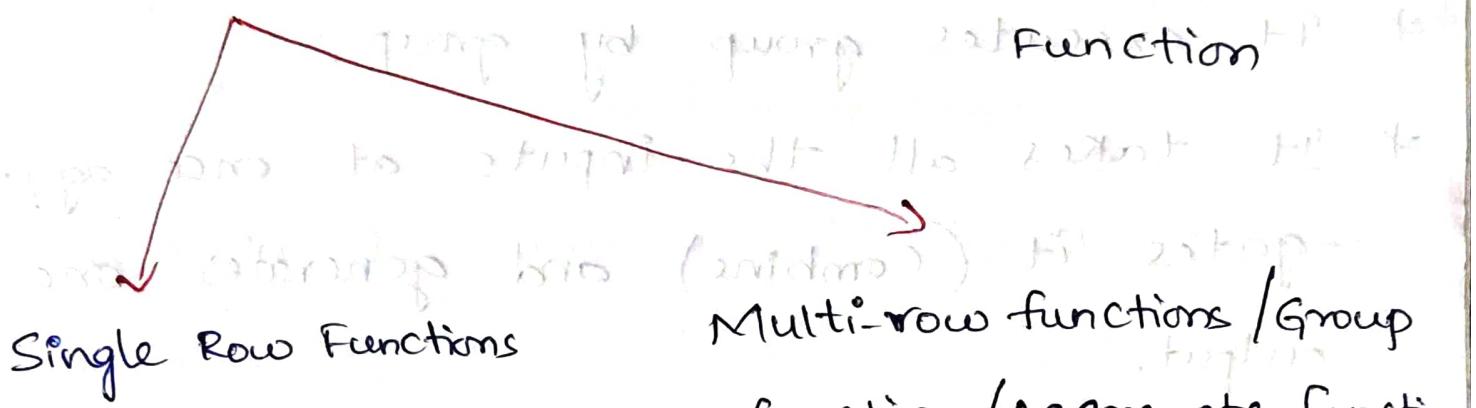
# Functions

- \* It is the list of instructions that are used to perform a specific task.

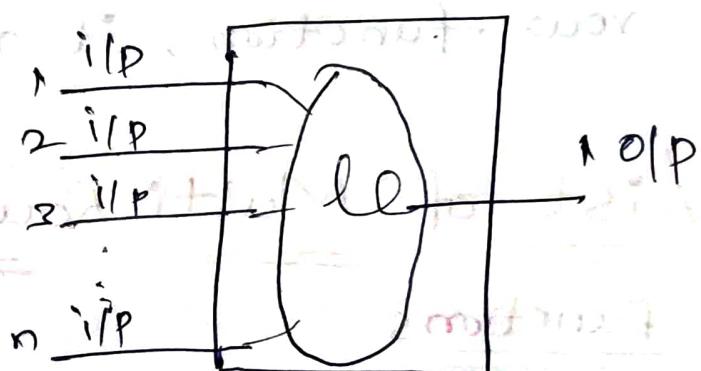
## Functions

### in-built functions

### User defined



1 IP	xx	1 O/P
2 IP	xx	2 O/P
3 IP	xx	3 O/P
n IP	xx	n O/P



## Single Row functions

- \* Single row functions executes row-by-row.
- \* It takes one input executes and generates one output then goes to the next input.

- \* if we pass 'n' no. of inputs to single row function, it returns 'n' no. of outputs.

## Multi-Row Functions

- \* Multi-row functions is also known as "group function" or "aggregate Functions".
- \* It executes group by group.
- \* It takes all the inputs at once aggregates it (Combine) and generates one output.
- \*\* If we pass 'n' no. of inputs to multi row function, it returns a single output.

## List of Multi Row Functions / Group / Aggregates Functions

- 1) Max()
- 2) Min()
- 3) Sum()
- 4) Avg()
- 5) Count()

- Note :- Multi row functions can accept only a single argument that is a column name or an expression.
- 1) Multi row functions can accept only a single argument that is a column name or an expression.
  - 2) Max() and Min() functions can be used for all the following datatypes problem.  
i.e. char, varchar, number or date.
  - 3) Sum() and Avg() functions can only take number column as an argument.
- \*\* 4) Multirow functions will ignore the null value.
- \*\* 5) we cannot use multi row functions in where clause.
- \*\* 6) we cannot use any column name with multi row functions in select clause.
- 7) Count() is the only MRF to which we can pass 'n\*' as an argument.
- (\*) (a) count(\*) (b) (more) times

① WAPTD Max sal from emp table

Ans: select Max(sal)  
From emp;

② WAPTD Min sal from emp table

Ans: select Min(sal)  
From emp;

③ WAPTD Total sal From emp table

Ans: select sum(sal) From emp;

④ WAPTD Avg sal from emp table.

Ans: select Avg(sal)  
From emp;

⑤ WAPTD Max salary in dept 20

Ans: select max(sal)  
From emp

where deptno = 20;

⑥ WAPTD Number of Emp's in Emp table.

Ans: select count(ename) (or) (empno) (\*)  
From emp;

① WAQTD Number of employees getting salary less than 2000 in deptno 10.

Ans:

Select count(ename)

From emp

where sal < 2000 and deptno = 10;

Q.P:-

①

WAQTD total salary needed to pay employees working as clerk.

Ans:

Select sum(sal)

From emp

where Job = 'CLERK'

Q.P:- 4150

③ WAQTD Average salary needed to pay all employees.

Ans:

Select Avg(sal)

From emp

④ WAQTD Number of employees having 'A' as their first character.

Ans:

Select count(ename)

From emp

where ename like 'A%'

Q.P:- 2

From emp;

(B) WAQTD no. of. emp's working in dept 10

Ans: Select count(\*)

From emp

where deptno in (10);

(C) WAQTD no. of. emp's working in dept 30.

Ans: Select count(\*)

From emp

where deptno in (30);

(D) WAQTD no. of. emp's working in each dept.

Ans: Select count(ename), Deptno

From emp

~~where~~ @

Group by Deptno;

O/P From

ename	sal	deptno
Dinga	2000	20
Raju	1500	30
BHeem	3000	20
BINGI	4000	20
KHAN	2500	10
King	1000	10

O/P of GroupBy

Deptno	ename	sal	Count(ename)
20	DINGA	2000	2
20	BHeem	2000	2
20	KHN	2000	2

Deptno	ename	sal	Count(ename)
10	DINGI	3000	1
10	King	3000	1

O/P of Select

Count(ename)	Deptno
3	20
1	30
2	10

## Group By clause

- \* We use group by clause to group the records.
- \* It executes row by row.
- \* For group by clause we can pass column name or an expression as an argument.

\*\* We can write group by expression along with multi row function in select clause.

Group by Expression :- Any column name or expression which is written in group by clause is known as group by expression.

- \* After the execution of group by clause it creates groups and if any clause executes after group by clause it executes group by group...

Syntax :- Select group by expression/group function

From table name

[Where <filter-condition>]

Group by Column-name/expression;

## Order of execution

- 1) From [at 2nd last part of query]
- 2) Where (if used) [Row - by - Row]
- 3) Group by [Row - by - Row]
- 4) Select [Group - by - Group]

Q1 WAPTD Max Sal in each dept.

Ans:-

Select Max(sal), deptno

From emp

Group by deptno;

Q2 WAPTD No. of emp's in each dept if the  
emp's are earning more than 2000.

Ans:-

Select count(ename), deptno

From emp

where sal > 2000

Group by deptno;

Q3 WAPTD Min sal in each Job

Ans:-

Select Min(sal), Job

From emp

Group by Job;

① WAPTD no. of employees working in each department except president.

Ans:- Select Deptno, Count(ename)

From emp

where Job != 'President'

Group by deptno;

② WAPTD total salary needed to pay all the employees in each job.

Ans:- Select job, sum(sal)

From emp

Group by job;

③ WAPTD Number of employees working as manager in each department.

Ans:- Select Deptno, Count(ename)

From emp

where Job = 'MANAGER'

Group by deptno;

④ WAPTD Avg Salary needed to pay all the employees in each department excluding the employees of deptno 20.

Ans:- Select deptno, count(\*), sum(sal)

From emp where job = 'SALESMAN'  
where Group by deptno;

- (8) WAPTD No. of employees with their maximum salaries in each job.

Ans:- Select max(sal), count(\*), Job

From emp  
Group by job;

- (9) WAPTD maximum salaries given to an employee working in each dept.

Ans:- Select max(sal), deptno

From emp

Group by deptno;

- (10) WAPTD no. of times the salaries are present in employee table.

Ans:- Select count(sal)

From emp

Group by sal;

- (11) WAPTD no. of employees hired on the same day into the same department.

## Having Clause

- \* We use having clause to filter the groups.
- \* We can pass multi-row function condition in having clause.
- \* It executes group by group.
- \* If we using having clause it should be used after group by clause.
- \* It cannot be used without group by clause.

## Syntax:-

Select group by expression / group function

From table name

[where <filter-condition>]

Group by column name / expression

Having <group filter condition>

## Order of Execution:-

- 1) From
- 2) where (if used) [Row - By - Row]
- 3) Group By [Row - By - Row]
- 4) Having [Group - by - Group]
- 5) Select [Group - by - Group]

Ex: WAPTD no. of employees working in each department having at least 2 emp's in each dept

- ment having atleast 2 emp's in each dept

Ans:-

4 - select Deptno, Count(ename)

① from emp

2 - Group by deptno

3 - Having Count(ename)  $\geq 2$

②

Emp o/p of Fnm

ENAME	SAL	DNO
A	4000	20
B	300	10
C	300	20
D	100	30
E	200	20
F	100	10

③ o/p of Group

<u>20</u>	<u>10</u>	<u>20</u>
A	600	20
C	800	20
E	200	20

<u>10</u>	<u>200</u>	<u>10</u>
B	200	10
F	100	10

<u>30</u>	<u>100</u>	<u>20</u>
(D)	100	20

④ o/p of Having

20 True

⑤ o/p of select

10

D.No Count

20

3

10

2

30

False

Q1. WAPTD DNO and no.of. emp working in each dept if there are atleast 2 clerks in each dept.

Ans:

Select Deptno, count(ename)

From emp  
where job = 'CLERK'  
Group by deptno

Having count(ename) >= 2;

Q2. WAPTD DNO and total salary needed to pay all emp in each dept if there are atleast 4 emp in each dept.

Ans:

Select Deptno, sum(sal), ~~count(\*)~~

From emp

Group by deptno

Having count(\*) >= 4;

Q3. WAPTD No.of. emp earning sal more than 1200 in each job and the total sal needed to pay emp of each job must excess 3800.

Ans:

Select Job, sum(sal), Count(G1)

From emp

Where Sal > 1200

Group by Job

Having sum(sal) > 3800

⑦ WANTED the hiredate which are duplicated in emp-table.

Ans:- Select hiredate, count(\*)

From emp

With Group by Hiredate

Having hiredate Count(\*) > 1

⑧ WANTED Avg. salary of each dept if avg sal is less than 3000.

Ans:- Select deptno, Avg(sal)

From emp

Group by deptno

Having Avg(sal) < 3000;

⑨ WANTED Deptno if there are atleast 3 emp in each dept whos name has char 'A' or 'S'.

Ans:- Select Deptno, Count(\*)

From emp

Where ename like '%.A%' or ename like '%.S%'

Group by deptno

Having count(\*) >= 3;

(10) WAGTD min and max salaries of each job if min sal is more than 1000 and max sal is less than 5000.

Ans:

Select min(sal), Max(sal), Job

From emp

Group by job

Having min(sal) > 1000 and max(sal) < 5000.

## ① Difference B/w where clause and Having clause

<u>Where clause</u>	<u>Having clause</u>
* It is used to filter the records.	* It is used to filter the groups.
* It executes Row by row.	* It executes group by group.
* We cannot pass multi row function condition in it.	* We can pass multi row function condition.
* It can be used without group by clause.	* It can't be used without group by clause.
* Where clause should be used after from clause only.	* Having clause should be used after group clause only.
* Logical operators are used in it.	* Logical operators are used in it.

① WAPTD Ename who is earning sal more than

2000.

Ans:- Select ename

From emp

where sal > 2000;

② WAPTD ename who is earning sal more than

2000. Smith.

Ans:- Select ename

From emp

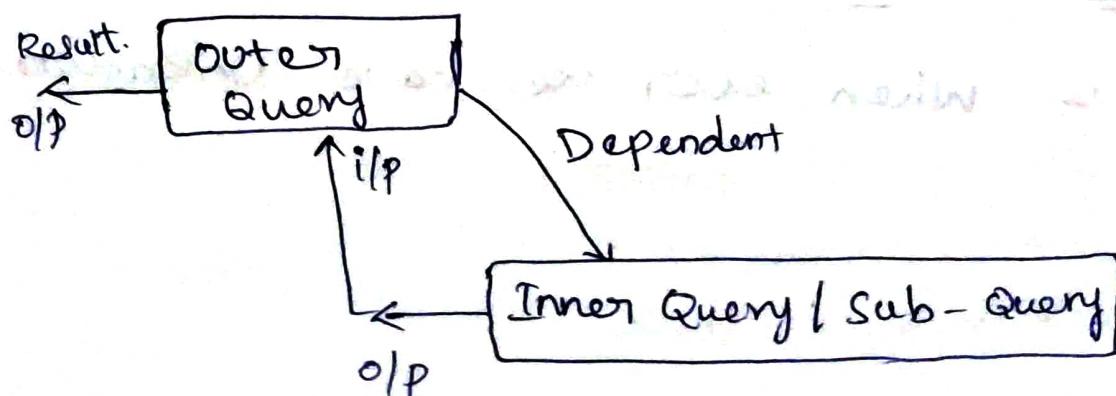
where sal > (Select sal

From emp

where ename = 'SMITH');

## Sub-Query Operator

\* A Query which is written inside another query is known as Sub-Query.



## Working Procedure :-

- \* Here we will be having minimum of 2 Queries.
  - 1) Outer Query
  - 2) Inner Query / sub-Query.
- \* Inner Query will execute first and generate the output.
- \* The o/p generated by the inner Query will be given as i/p to the outer Query.
- \* The outer Query will execute and generate the o/p.
- \* This o/p will be the result.
- \* By this we can say that outer query is dependent on inner query.

## Why Sub-Query

Case 1 :- When ever we have unknown

① WAPTD names of the employees earning more than Adams.

Ans:-

```
Select ename  
From emp  
where sal > (select sal  
From emp  
where ename = 'ADAMS');
```

③ WAPTD Name and Deptno of the employees if they are working in the same dept as Jones.

Ans:-

```
Select ename, deptno  
From emp  
where deptno = (select deptno  
From emp  
where ename = 'JONES');
```

④ WAPTD Name and salary of employees earning less than King.

Ans:-

```
Select ename, sal  
From emp  
where sal < (select sal  
From emp  
where ename = 'King');
```

④ Want ID, ename and job of all the employees having  
-king in the same designation as James.

Ans:-

Select ename, Job

From emp

where Job = (select Job

From emp

where ename = 'James');

⑤ Want ID, Empno and ename along with annual  
Salary of all the employees if their annual salary  
is greater than Ward's Annual Salary.

Ans:-

Select empno, ename, sal \* 12 <sup>AnnualSalary</sup> As

From emp

where sal \* 12 > (select sal \* 12

From emp where ename = 'Wards')

⑥ Want name and hiredate of the employees if  
they are hired before Turner.

Ans:-

Select ename, hiredate

From emp

where hiredate < (select hiredate

From emp

where ename = 'Turner');

Q7) What is the name and hiredate of the employees if they are hired after the president.

Ans:- Select ename, hiredate

From emp

where hiredate > (select hiredate

From emp

where Job = 'President');

Q8) What is the name and sal of the employee if they are earning sal less than the employee who empno is 7839.

Ans:- Select ename, sal

From emp

where sal < (select sal

From emp

where empno = 7839);

Q9) What are all the details of the employees if the employees are hired before miller.

Ans:- Select \*

From emp

where hiredate < (select hiredate

From emp

where ename = 'MILLER');

(10) WAPTD Ename and empno of the employees  
are earning more than allen.

Ans:-

Select ename, empno

From emp

Where sal > (Select sal

From emp

Where ename = 'Allen');

(11) WAPTD Number of employees hired after King

Ans:-

Select Count(\*)

From emp

Where hiredate > (Select hiredate

From emp

Where ename = 'King');

(12) WAPTD total salary given to the employees working  
in the same dept as Ward.

Ans:-

Select sum(sal)

From emp

Where Deptno = (Select Deptno

From emp

Where ename = 'Ward');

(20) WAPTD ename, sal and designation of the employee whose annual salary is more than smith and less than King.

Ans:- Select ename, sal, Job  
From emp

= where sal\*12 > (select sal\*12

From emp

sal\*12 < (select sal\*

From emp where ename='smith') and (

From emp

where ename

'King')

Case 2:- When ever the data to be selected

and the condition to be executed is present in  
Diff. Table we use Sub-Query

① WAPTD DName in which Prince is working.

Ans:-

ename	dno
Dinga	20
Dingi	10
Rose	10
Prince	30
Lilly	20
Lela	10

dname	dno
D1	10
D2	20
D3	30
D4	40

ename

Select dname

From Dept

where ename='Prince'

where deptno = (select deptno

from emp  
where ename='Prince');

(21) WAQTD DName of the employees whose name is smith.

Ans:-  
Select DName  
From Dept

where Deptno = (select Deptno  
From emp  
where ename = 'smith');

(22) WAQTD DName and Loc of the employee whose ename is King.

Ans:-  
Select DName, Loc

From Dept  
where Deptno = (select Deptno  
From emp  
where ename = 'King');

(23) WAQTD Loc of the emp whose employee number

is 7902.

Ans:-  
Select Loc

From Dept

where Deptno = (select Deptno  
From emp  
where empno = 7902);

(26) WAPTD names of the employees working in accounting department.

Ans:-  
= Select ename  
From emp

where deptno = ( select Deptno  
From Dept

where Dname

~~Dept~~

= Accounting

(27) WAPTD Dname and loc along with deptno of the employee who's name ends with 'R'.

Ans:-  
= Select Dname, loc , Deptno  
From Dept  
where Deptno in (select Deptno  
from emp

where Deptno in (select Deptno  
from emp

(28) WAPTD Dname of the employee whose designation is president.

Ans:-  
Select Dname

From Dept

where Deptno in (select Deptno

from emp

where Job = 'President')

(33) WAPTD Name and sal of the employees earning more than king in the Dept accounting.

Ans: select ename, sal  
from emp

where sal > (select sal  
from emp  
where ename='king') and Deptno = (select Deptno  
from Dept  
where Dname  
= 'Accounting')

(34) WAPTD Details of the employees working as sales man in the department sales.

Ans: select \*  
from emp  
where job='salesman' and Deptno in (select Deptno  
from Dept  
where Dname='Sales')

(35) WAPTD Name, sal, Job , hiredate of the employees working in operations department and hired before King.

Ans: Select ename, sal, job, hiredate  
from emp  
where Deptno = (select Deptno  
from Dept  
where Dname='operations')  
and hiredate < (select hiredate  
from emp  
where ename  
= 'King'))

## Nested Sub-Query

- \* A sub-Query written inside another sub-Query is known as Nested Sub-Query.
- \* We can nest about 255 sub-queries.

⑥1) WAQTD 3RD minimum salary. with ename

select ename, sal

from emp

where sal in (select min(sal)

from emp

where sal > (select min(sal))

from emp

where sal > (select min(sal))

from emp))

⑥2) WAQTD 5th maximum salary.

select max(sal)

from emp

where sal < (select max(sal))

from emp

where sal < (select max(sal))

from emp

w

with base order as per query and to listed GRAM

(64) WAPTD Empno of the employee earning 2nd max salary.

A:  
Select empno  
From emp

where sal in (select max(sal))

From emp

where sal < (select max(sal))

From emp))

(65) WAPTD Department name of an employee getting 4th max sal.

Ans:  
Select Dname  
From Dept

Select Deptno

From emp

where sal > in (select max(sal))

From emp

From emp

where sal < (select max(sal))

From emp

where sal < (select max(sal))

From emp

where sal <

(select max(sal))  
From emp))

(66) WAPTD Details of the employee who was hired 2nd.

Ans:

```
Select *
From emp
where Hiredate in (select min(Hiredate)
From emp
where hiredate >= (select min(hiredate)
From emp));
```

67) WATD Name of the employee Hired before  
the last employee.

Ans:

```
Select ename
From emp
where hiredate in (select max(Hiredate)
From emp
where hiredate < (select max(hiredate)
From emp));
```

68) WATD Loc of the employee who was hired first.

Ans:

```
Select loc
From dept
where Deptno in (select Deptno
From emp
where Hiredate in (select min(Hiredate)
From emp));
```

(69) WAPTD Details of the employee earning 7th minimum salary.

-um salary.

slr

select \*  
from emp

where sal > (select min(sal))

From emp

where sal > (select min(sal))

From emp

where sal > (select min(sal))

From emp

↳ where sal > (select min(sal))

ans to 202

Ans

(stability) x (EMP) at stability window

where sal > (select min(sal))

From emp

where sal > (select min(sal))

From emp

where sal > (select min(sal))  
From emp

ans to 202

(70) WAPTD Dname of employee getting 2nd maximum salary.

salary.

Ans

Select Dname

Ans

From Dept

where Deptno in (Select Deptno

From emp

where sal in (select max(sal))

From emp

where sal < (select max(sal))  
From emp))

<u>Ename</u>	<u>Sal</u>	<u>Job</u>
A	1000	Salesman
B	2000	Manager
C	1000	Salesman
D	3000	Manager
E	4000	Manager
F	1500	Salesman

i) WAGTD Ename who are earning more than sales man

Select ename

From emp  
where sal > (select max(sal))

from emp

where job = 'Salesman')

Op: B, D, E,

(or)

Select ename

From emp

where sal > All (select sal

from emp

where job = 'Salesman')

## Types of Sub - Query

\* There are 2 types

1) Single row sub-Query

2) Multi row Sub-Query

① Single row sub-Query :- A sub-Query

which returns exactly one O/P is known as single row sub-Query.

\* We can use operators such as in, Notin, All, Any.

Ex:-

① WAQTD Ename earning more than Scott

A:-

Select ename

from emp

where sal > (select sal

From emp

where ename = 'scott')

Ques:- do want

2. Multi Row Sub-Query :- A sub-Query which returns more than one op is known as Multi row Sub-Query.

\* We must use operators such as in, not in, All, Any.

All operator :- All operator is a special operator which can accept multi values at RHS.

\* It will return true only if all the condition at the RHS is satisfied.

Syntax :- columnname/Expr.: Relational op. All ( $v_1, v_2, \dots, v_n$ )

Ex:-  $100 > \text{All} (50, 70, 90, 120)$

$$\begin{array}{l} 100 > 50 \quad T \\ 100 > 70 \quad T \\ 100 > 90 \quad T \\ 100 > 120 \quad F \end{array} \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \text{FALSE}$$

Any Operator :- Any operator is a special operator

which can accept multiple values at RHS.

\* It will return true if any one of the condition at the RHS is satisfied.

Syntax  $\exists$  column.name) Expr ; Relational op Any ( $v_1, v_2, v_3, \dots, v_n$ )

Ex:  $100 > \text{Any}(75, 100, 125, 150)$

$100 > 75 \text{ T}$   
 $100 > 100 \text{ F}$   
 $100 > 125 \text{ F}$   
 $100 > 150 \text{ F}$

True

where  $\exists$  range is set of numbers  $\{100\}$  of strength 1

(2) WATD Ename who is earning more than  
atleast a salesman.

At

Select ename  
from emp  
where sal > Any (select sal

sal > Any (1000, 1000, 1500) where job = 'Salesman')

1000 F

2000 T.

3000 F

4000 T.

5000 T.

db  
B, D, E, F

from emp

T >= 700

F or C

T >= 300

F or C

T >= 500

F or C

T >= 1000

F or C

T >= 1500

F or C

Note\* All and Any operators should be used along

with relational operator

\* But any operator can be used along with (=, !=)

Q) ALL and Any:  
Q5) WAQTD Details of the employees hired after  
all the clerks.

Ans:

Select \*

From emp

Where Hiredate > All (Select Hiredate

From emp

Where Job = 'CLERK')

Q5) WAQTD Name and salary for all the employees  
if they are earning less than atleast a manager.

Ans:

Select ename, sal

From emp

Where sal < Any (Select sal from

From emp

Where Job = 'Manager')

Q5) 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')

(55) WAPTD names of the employees hired after all the managers and earning salary more than all the clerks.

Ans: Select ename  
From emp

where Hiredate > All (select \$hiredate  
from emp)  $\cap$    
where Job = 'MANAGER' and

Sal > All (select sal

From emp

where Job = 'clerk'

(56) WAPTD Details of the employees working as

Clerk and Hired before Atleast a Salesman

Ans: Select \*  
From emp

where Job = 'clerk' and Hiredate < Any (select

Hiredate  
from emp)  $\cap$    
where Job = 'Salesman'

(57) WAPTD Details of employees working in account or sales Dept.

Ans: Select \*  
From emp  
where Deptno  $\in$  In

⑥ WAPTD Emp names if employees were hired after all the employees of Dept 10.

Ans: Select 'ename'

~~Select~~ From emp

where Hiredate > All (Select Hiredate

From emp

where Deptno=10,

select Ename ) in waiting

## Employee - Manager Relation

① WAPTD Manager Name of ALEN

EMP

EID	Ename	MGR
1	ALEN	2
2	Dingi	3
3	Dinga	5
4	Queen	5
5	King	6
6	Raj	NULL

Select Ename

From emp

where EID IN (Select MGR

From emp

where Ename='ALEN')

EID	IN 2 F
1	F
2	T
3	F
4	F
5	F
6	F

① WAPTD Allen Manager name.

Select ename

From emp

where empno in (select MGR

From emp

where ename = 'ALLEN')

② WAPTD sal of ADAMS manager.

Select sal

From emp

where empno in (select MGR

From emp

where ename = 'ADAMS')

③ WAPTD Ename's who are earning sal more

than Adams manager.

Select ename

From emp

where sal > (select sal

From emp

where empno in (select MGR

From emp

where ename

= 'ADAMS')

(71) WAQTD Smith's Reporting Manager's name.

A<sup>1</sup> Select ename

From emp

where empno in (select MGR ename)

From emp

where ename = 'smith'.

(72) WAQTD ADMS. manager's manager name.

A<sup>2</sup> Select ename

From emp

where empno in (select MGR ename)

From emp

where empno in (select MGR

From emp

where ename = 'ADAMS'

(73) WAQTD Dname of Jones' manager.

A<sup>1</sup> Select dname

From Dept

where Deptno in (select Deptno

From emp

where empno in (select Deptno

From emp

where empno in (select

MGR

From emp  
where ename = 'JONES'

Case 2  
① Display ename who is reporting to Ding

A:

Select ename

From emp

where MGR in (select

MGR
2
3
C
F
5
F
6
F

in 3 F or

G

F

F

F

F

From emp

where ename = 'DINGA')

② WAQTD Emp's name who are reporting to King

A:

Select ename

From emp

where MGR in (select

From emp

where ename = 'King')

③ WAQTD Name of the employees reporting to blake.

A:

Select ename

From emp

where MGR in (select empno

From emp

where ename = 'blake')

# INDIVIDUAL Concepts

## Attributes:

- 1) Key Attributes / ~~Candidate~~ <sup>date</sup> Key
- 2) Non - Key Attributes
- 3) Prime key Attributes
- 4) Non prime <sup>key</sup> Attributes
- 5) composite key Attributes
- 6) Super Key Attributes
- 7) Foreign key Attributes.

① Key Attributes: An attribute which is used to identify a record uniquely from the table is called Key attribute.

② Non - Key Attribute: All the attributes except key attributes are referred as non-key attributes.

③ Prime - Key Attributes: Among the key attri

- butes an attribute is chosen to be the main attribute to identify the record uniquely from the table.

④ Non-prime Key Attribute : All the key attributes except prime key attribute is referred non-prime key-Attribute.

⑤ Composite Key Attribute : It is a combination of two or more non key attributes which is used to identify the record uniquely from the table.

⑥ Super key Attribute : It is the set of all the key attributes.

⑦ Foreign key Attribute : It behaves as an attribute of another entity to represent the relation.

## Functional Dependency

- \* Let us consider the relation 'R' with two attributes 'x' and 'y' respectively.
- \* In which attribute 'x' determines Attribute 'y'. (or) In other words, 'y' is dependent on 'x'.
- \* There exists functional dependency.  
 $R \rightarrow \{x, y\}$   
 $x \rightarrow y$   
y is dependent on x.

### Types of functional Dependency

- 1) Total functional Dependency
- 2) Partial functional Dependency
- 3) Transitive functional Dependency.

- ① Total functional dependency :- If all the attributes in a relation are determined

by a single attribute which is a key attribute, then there exists total functional dependency.

\* In total functional dependency there are no anomalies or redundancy.

Anomaly → These are the side effects which are caused during the DML operators.

Redundancy → These are the repeated or Duplicate data which is undesirable.

Ex Let us consider a relation with 4 attributes A, B, C & D. In which 'A' is a key attribute.

$$R \rightarrow \{A, B, C, D\}$$

A is K.A.

$$A \rightarrow B$$

$$A \rightarrow C$$

$$A \rightarrow D$$

There exists T.F.D.

$$A \rightarrow \{B, C, D\}.$$

## ② Partial Functional Dependency

- \* For a partial functional dependency to exist there must be a composite key attribute.
- \* One of the attributes in composite key relation determines another attribute separately, and this is known as partial functional dependency.
- \* In partial functional dependency we have redundancy and Anomaly.

Ex: Let us consider a relation 'R' with 4 attributes A, B, C, D; in which A & B, are composite key attributes.

$$R \rightarrow \{A, B, C, D\}$$

A & B  $\rightarrow$  com. Key. Attr

$$(A, B) \rightarrow (D)$$

$$B \rightarrow \{C\}$$

There exist P.F.D.

### ③ Transitive functional Dependency

- \* If an attribute is determined by a non-key attribute which intern is determined by a key attribute, then there exist transitive functional dependency.
- \* In transitive functional dependency we have redundancy and Anomaly.

Ex: Let us consider a relation with four attributes A, B, C, D in which A is a key attribute.

$$R \rightarrow \{A, B, C, D\}$$

A is K.A.

$$A \rightarrow B$$

$$A \rightarrow D$$

$$D \rightarrow C$$

$$A \rightarrow C$$

There exist T.F.D. with

Normalization :- It is a process of reducing the larger table into smaller table -  
in order to remove redundancy and anomaly by identifying their functional dependency.

(or)

It is a process of decomposing a large table into smaller table to remove redundancy and Anomaly.

(or)

It is a process of reducing the table to its normal form.

Normal form :- A table without redundancy and anomaly is set to be in normal form.

Levels of Normal form

- 1) First Normal Form (1NF)
- 2) Second Normal form (2NF)
- 3) Third Normal form (3NF)
- 4) Boyce - Codd Normal form (BCNF)

Note: A table is said to be normalized if we reduce the table to 3<sup>rd</sup> normal form.

① 1NF: A table is said to be in 1<sup>st</sup> normal form. If it satisfies the following conditions:

- \* A table should not consist of multi-valued data.

- \* A table should not have duplicate or repeated values.

② 2NF: A table is said to be in 2<sup>nd</sup> normal form if it satisfies the following condition

-s.

- \* The table should be in 1<sup>st</sup> Normal form.

- \* The table should not have partial functional dependency.

- \* The table should not have non-trivial dependencies.

Note: If the table consists of partial functional dependency, then the attributes responsible for it are removed from the table.

③ 3NF: A table is said to be in 3rd normal form if it satisfies the following conditions:

- \* A table should be in 2nd normal form.
- \* A table should not have transitive functional dependency.

Note: If the table consists of transitive functional dependency then the attributes responsible are removed from the table.

### Ex: Form 1NF

Student

SID	sName	skills
1	Rahul	SQL
2	Pooja	SQL, Java
3	Anur	SQL, PY
4	Bindu	SQL, Java, PY
5	Pooja	MANU

changes to

Student		SID	Sname	SQL	JAVA	Manual	Py
1	Rahul			SQL			
2	Pooja			SQL	Java	Manual	
3	Anu			SQL			Py
4	Bindu			SQL	Java		Py

Ex: For 2NF

$$R \rightarrow \{ EID, Ename, sal, comm, DNO, Dname, loc \}$$

$$(EID \neq DNO) = \{ Ename, sal, comm, Dname, loc \}$$

Composite Key Attributes  $\rightarrow$  Partial functional dependency.

$$EID \rightarrow \{ Ename, sal, comm \} \quad DNO - \{ Dname, loc \}$$

$$R1 \rightarrow \{ EID, Ename, sal, comm \}, R2 \rightarrow \{ DNO, Dname, loc \}$$

Ex: For 3NF

$$R \rightarrow \{ EID, Ename, sal, comm, pincode, state, coun \}$$

KA

EID - Ename

sal

comm

pincode - state

country

Transitive functional dependency

$EID \rightarrow \{Ename, Sal, comm\}$ ,  $Pincode \rightarrow \{state, country\}$

$R_1 - \{EID, Ename, Sal, comm\}$        $R_2 - \{Pincode, state, country\}$

#### ④ Boyce-Codd Normal Form (BCNF)

\* It is the updated version of 3NF also

Called 3.5NF.

#### Single Row Functions

- \* Single Row function executes Row by Row.
- \* It will take single input and generates and gives single output.
- \* If we pass 'n' no. of ip to single row function, it returns 'n' no.of outputs.
- \* We can nest single row functions.

① UPPER() :- This function is used to convert the given string into upper case.

Syntax:-  $\text{UPPER}(\text{'string'})$ .

Ex:-  $\text{Select UPPER}(\text{'pavan'})$

From Dual;

O/P PAVAN

Dual:- Dual is dummy table. To print the result of any mathematical operations done.

② LOWER() :- This function is used to convert the given string into lower case.

Syntax:-  $\text{LOWER}(\text{'string'})$

Ex:-  $\text{Select LOWER}(\text{'PAVAN'})$

From Dual;

O/P:- pavan

③ INITCAP() :- This function is used to convert the initial character of the given string into upper case.

Syntax:-  $\text{INITCAP}(\text{'string'})$

Ex:- Select INITCAP ('PAVAN')  
From dual;

OP:- Pavan

① WATCD Ename's in lower case

A:- Select lower ('Ename')  
From emp;

② WATCD first character upper and remaining  
in lower case.

A:- select initcap ('Ename')  
From emp;

④ Length(): This function is used to count  
the no. of characters that are present  
in the string.

Syntax:- Length ('string');

Ex:- ① Select Length ('INDIA');

From Dual;

OP:- 5

② select length ('MY PAVAN')

Dual,  
From ~~Dual~~:

Q: 8. Write a PL/SQL block to reverse the given string.

⑤ Reverse (): This function is used to reverse the given string.

Syntax: ~~Select~~ Reverse ('string')

Ex: Select Reverse('PAVAN')  
From Dual;

O/P: NAVAP.

⑥ Ex: Malayalam

→ check the given string is palindrome or not. If it is then print the result as "Is Palindrome".

A:  
Select 'is Palindrome'  
From Dual  
Where 'Malayalam' = Reverse('malayalam');

⑥ SUBSTR(): This function is used to extract the part of the string from the given original string.

Syntax: SUBSTR('original string', Position, [length])

Ex:- Select substr('Bangalore', 1, 1)  $\rightarrow$  B

Bangalore  
1 2 3 4 5 6 7 8 9  
 $\rightarrow$  1, 2  $\rightarrow$  BA  
 $\rightarrow$  3, 3  $\rightarrow$  NGA

BANGalore  
 $\rightarrow$  1, 5, 2  $\rightarrow$  AL  
 $\rightarrow$  8, 5  $\rightarrow$  RE  
 $\rightarrow$  9, 5  $\rightarrow$  E

From the 3rd position  $\rightarrow$  9, 1  $\rightarrow$  E

From the 2nd position  $\rightarrow$  1, 8  $\rightarrow$  Bangalore  
 $\rightarrow$  4  $\rightarrow$  galore

Substr ('Banglore', -1, 1)  $\rightarrow$  B

$\rightarrow$  , -5, 2  $\rightarrow$  AL

$\rightarrow$  , -3, 3  $\rightarrow$  ORE

$\rightarrow$  , -4, 8  $\rightarrow$  LORE

$\rightarrow$  , -6  $\rightarrow$  GALORE

Q) WAPTD the first character of all the emp's  
from emp table.

Select substr(ename,1,1)

From emp

Q) WAPTD the first three characters of all the  
emp's.

Select substr(ename,1,3)

From emp

Q) WAPTD details of emp's if their name  
start with 'A' using SRF.

Select \*

From emp

where ename like '^A%'

Select \*

From emp

where substr(ename,1,1) = 'A'

Q) WAPTD details of emp's if their name  
starts with 'A' or 'S' using SRF.

Select \*

From emp

where substr(ename,1,1) in ('A', 'S')

## Based on Length

- ① WAPTD Ename who are having 5 characters in their name.

A:- select ename

From emp

where length(ename) = 5;

- ② WAPTD ename and sal of emp's who are getting 3 digit sal using SRF.

A:- select ename, sal

From emp

where length(sal) = 3;

- ③ WAPTD ename and comm of emp's who are getting 3 digit comm using SRF.

A:- select ename, comm

From emp

where length(comm) = 3;

- ⑦ MOD() :- This function is used to obtain modulus of the given numbers (remainder).

Syntax :- MOD(m, n)

① ~~WAQTD~~: Details of emp's whose empno  
is an ODD no.

Select \*

From emp

where mod (empno, 2) = 1

② ~~WAQTD~~ details of even records,

Select \*

From emp

where mod (empno, 2) = 0

⑧ TO\_CHAR(): This function is used

to convert the given date to string form  
at.

Syntax: TO\_CHAR ( Date, "Format Models" )

Format models

1) YEAR → Twenty Twenty

2) YYYY → 2020

3) YY → 20

4) MONTH → August

5) MON → Aug

- 6) MM → 08
- 7) DAY → WEDNESDAY
- 8) DY → WED
- 9) DD → 19
- 10) D → 4
- 11) HH24 → 11
- 12) HH12 → 11
- 13) MI → 40
- 14) SS → 24  
(seconds)

SYSDATE :- This is used to obtain the

Current Date present Date from Database.

~~Doubt~~ ① What's Ename's of the emp's who were hired in the month of Feb.

A:- select ename

From emp

~~where Hiredate in (Select Hiredate~~

~~from emp~~

where To-Char (Hiredate,'MON') in ('Feb')

System stamp :- This command is used to obtain the data and time along with time zone.

To\_date() :- This function is used to convert the date string to date format.

Syntax :- `TO_DATE('DATE-STR')`

- ① WAPTD ename's whose names starts with vowels (A,E,I,O,U)

select ename

From emp

where substr(ename, 1, 1) in ('A', 'E', 'I', 'O', 'U')

- ② WAPTD Ename in lower case and Job in Reverse format if the emp's name having 6 characters.

Select lower(ename), reverse(job)

From emp

where length(ename) = 6

③ WAP TO Details of emp's if they are earning 4 digit sal.

A: select \* from emp  
From emp  
Where length(sal) = 4;

④ WAP TO Ename and job of emp's if the job starts with string Man or ends with string MAN.

A: Select ename, job  
From emp  
Where substr(job, 1, 3) = 'Man' or  
substr(job, -3, 3) = 'MAN'

⑤ NVL: [NULL Value Logic]

Syntax: NVL (ARG1, ARG2)

\* It can accept 2 arguments.

\* In Arg1 we must write a column name or expression that can be null.

- \* IN arg1 we must write a value that can be substituted in place of null.
- \* If Arg1 is Not null , NVL returns same value present in /arg1.

Q) WATD Total sal of Each emp in emp table.

Emp

<u>Ename</u>	<u>sal</u>	<u>comm</u>
A	1000	100
B	2000	200
C	2000	
D	1500	300
E	3000	

Select  $\text{sal} + \text{NVL}(\text{comm})$

From emp;

$\text{Sal} + \text{NVL}(\text{comm}, 0)$

$$1000 + \text{NVL}(100, 0) = 1000 + 100 = 1100$$

$$2000 + \text{NVL}(200, 0) = 2000 + 200 = 2200$$

$$2000 + \text{NVL}(NULL, 0) = 2000 + 0 = 2000$$

$$1500 + \text{NVL}(300, 0) = 1500 + 300 = 1800$$

$$3000 + \text{NVL}(NULL, 0) = 3000 + 0 = 3000$$

1100  
2200  
2000  
1800  
3000

Q) Instr() : This function is used to

obtain index value of the substring which is present in the original string.

\* Index value - Position of character.

## Syntax:

INSTR ('original\_string', 'sub-STR', Position, [Nth occurrence])

Nth occurrence - No. of times it is present.

Ex: INSTR ('BANANA', 'A', 1, 1) = 2

BANANA  
123456  
\*  
Index

'A', 1, 2) = 4

'A', 3, 1) = 4

'A', 4, 2) = 6

'AN', 1, 1) = 2

'AN', 3, 1) = 4

'A' (4, 4) = 0

'AN', 1, 3) = 0

Q1) WAQT Details of emp's if their Name having character 'A' using SRF.

Ans: Select \* from emp

where INSTR(Ename, 'A', 1, 1) > 0;

Q) WAP TO Display Ename's of emp's in lower case if emp's are having character 'A' present atleast twice in their name using SRF.

Select Lower (Ename)

From emp

where Instr(Ename, 'A', 1, 2) > 0

exactly two A's

where Instr(ename, 'A', 1, 2) > 0 and Instr(ename, 'A', 1, 3) = 0

ER Diagrams:-

Entity-Relationship: An entity relationship

models describes the structure of database with the help of diagram which is known as ER diagrams.

\* ER diagram has 3 main Components.

1) Entity

2) Attribute

3) Relationship

## Notations of ER Diagram

① 1) strong Entity →



② 2) Weak entity →



③ 3) Attributes →



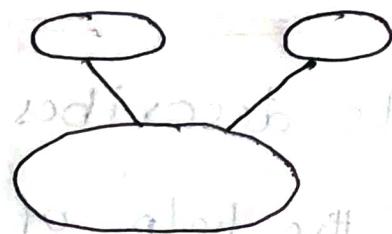
④ 4) Prime Key Attributes →



⑤ 5) Derived Attributes →



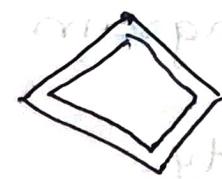
⑥ 6) Composite Key Attribute →



⑦ 7) strong relationship →



⑧ 8) weak relationship →



⑨ 9) Total Participation →



⑩ 10) Partial participation →



① **Strong Entity**: An entity which has a primary key is known as strong entity.

② **Weak entity**: An entity which does not have a primary key is known as weak entity.

③ **Strong Relationship**: The relationship that exists b/w & strong and the weak entities entities is known as strong relationship.

④ **Weak Relationship**: The relationship that exists b/w the strong and the weak entities is known as weak relationship.

⑤ **Total Participation**: It is used to represent the total participation of an entity in the relationship.

⑥ **Partial Participation**: It is used to represent the partial participation of an entity in the relationship.

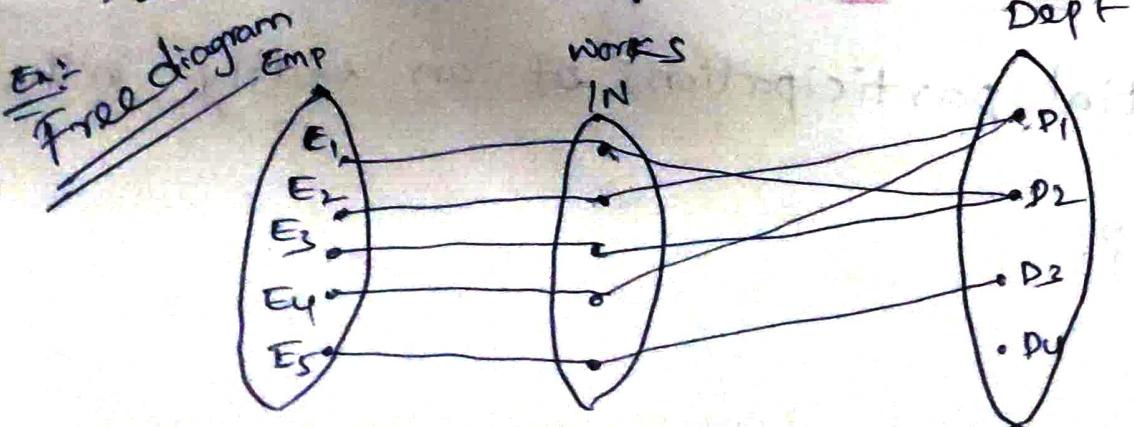
## User Requirements

- \* In my company I have multiple departments and an emp can work in any dept. but should be working in only one dept.
- \* In a department any no. of emp's are allowed to work.
- \* A department as to be or should be left empty for a particular period of time.

Maximum Participation :- It is the max. no. of times an entity that can take part in a relation.

Minimum Participation :- It is the min. no. of times an entity that can take part in a relation.

Cardinality Number :- The max. Participation is considered as Cardinality number.



$P(P_{min}, P_{max})$   $\rightarrow$   $P(P_{min}, P_{max})$

$P(1, 1)$   $\rightarrow$   $P_1(0, N)$

Cardinality no = 1, C.No =  $|N|$ , relation.

Cardinality Ratio :- Ratio between cardinality of other with 1:1

Cardinality Ratio (Relationship Ratio)

We have four cardinal relationships

\* 1:1 [One to One]

\* 1:N [One to Many]

\* N:1 [Many to One]

\* N:N [Many to Many]

\*\*\*

Note

1) If the ratio's are:-

1:N

N:1

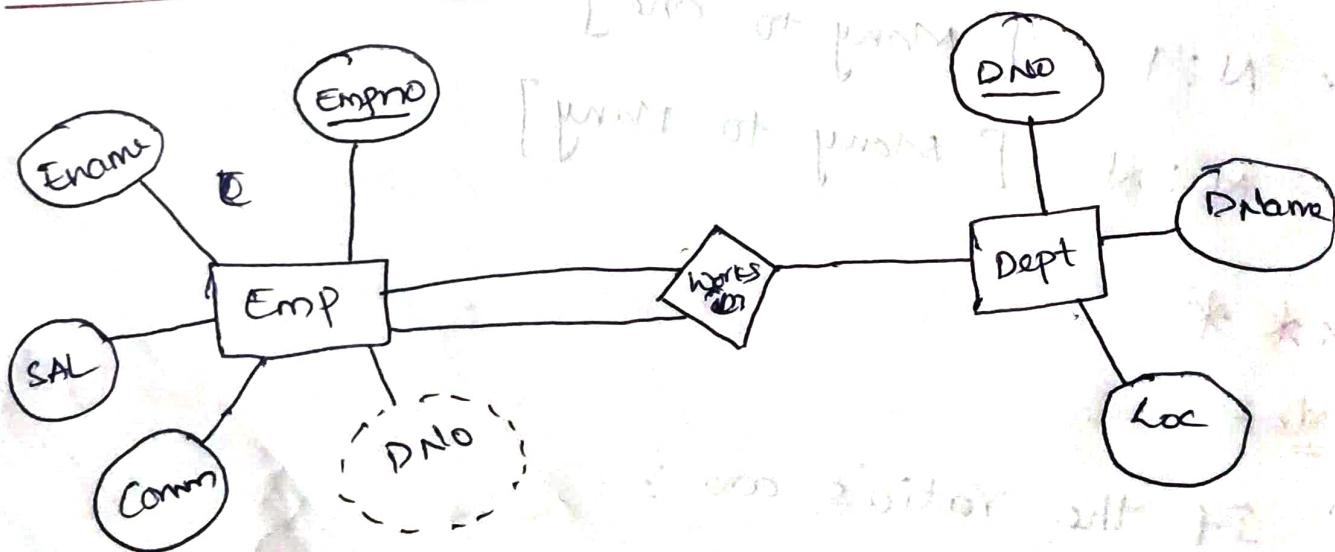
1:1, Then we don't have to use a separate table to store the relationship.

\* The primary key of an entity whose cardinality (no.) is N, is chosen to be a foreign key in the entity whose cardinality is 1.

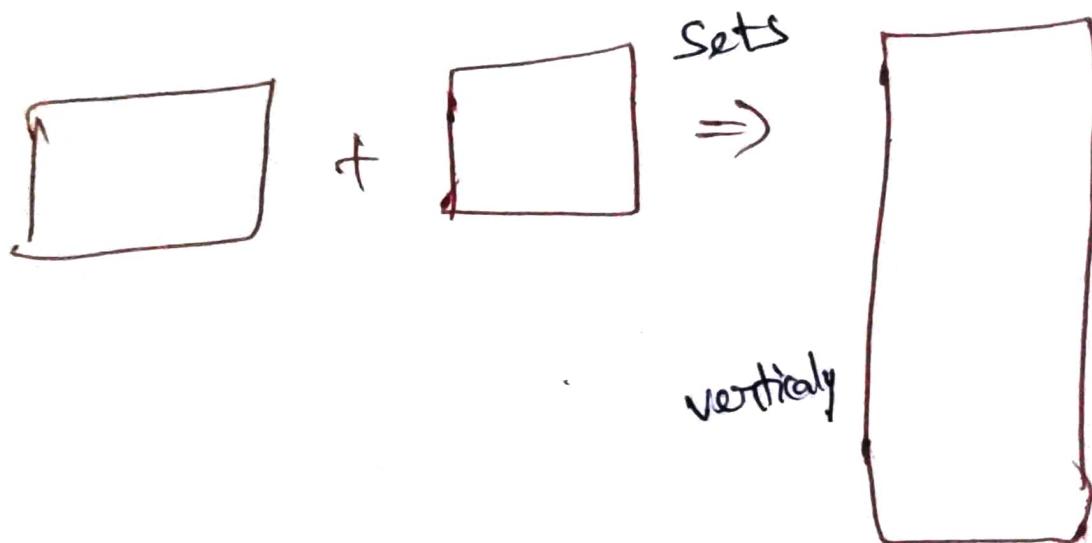
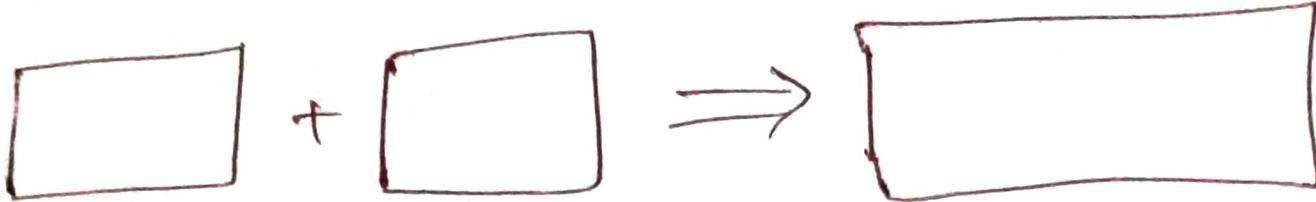
2) If the ratio is N:N then we need a separate table to store the relationship.

\*\*\* The primary key's of both the table are introduced as a foreign key in the new table.

### Example of ER diagram



D/w blw joins & sets. =  
Joins Horizontally



Update  
set = case

\* **Set Theory**: Sets in SQL is used to merge the tables vertically.

Note: The elements that are selected in both the tables must correspond to each other.

**Set operators**: We have 4 set operators

1) Union ~ AP with IP

2) Union ALL  $\{5, 2\}$  = A union B

3) Intersection // IT common SET

4) Minus //

Ex: part # sequential,  $A = \{1, 2, 3, 4, 5\}$ ,  $B = \{5, 6, 7\}$  Q)  
consider as primary

① **Union**: (most) non-trivial  
 $A \text{ Union } B = \{1, 2, 3, 4, 5, 6, 7\}$   
non-rep

i.e. Distinct set

② **Union ALL**:

~~It's also called OTCA~~  
 $A \text{ Union ALL } B = \{1, 2, 3, 4, 5, 5, 6, 7\}$   
non-rep

(some) + rep

i.e. Distinct + Duplicates.

### 3) Intersection

$$A \text{ intersection } B = \{5\}$$

Common

### 4) Minus:

$$A \text{ minus } B = \{1, 2, 3, 4\}$$

T1 Minus T2

$$B \text{ Minus } A = \{6, 7\}$$

T2 Minus T1

- ① WAQTD Ename in lowercase if they are working as salesman.

A:  
= select lower(ename)

From emp

where job in 'Salesman';

- ② WAQTD Ename in upper case except the salesman.

A:  
Select upper(ename)

From emp where job not in 'salesman';

Q3) Want ename in lowercase if they are working as salesman. If not display their ename without space & ( ) round the names in upper case.

A:

Select lower(ename) from emp

From emp  
where job in 'salesman'  
Union

Select upper(ename)

From emp  
where job not in 'salesman'

### Difference b/w Joins & set operators

- \* Joins are used to join two tables horizontally, but
- \* Set operators are used to join two tables vertically.



This is SRF

V.V.Imp

\* Case () :- This function behaves as

"if else" statement.

- \* If the condition is satisfied then it prints the result, else it checks for the next condition.
- \* If none of the conditions are true it prints the default result.

Syntax:-

```
case when Condition-1 then 'Result1'  
      when Condition-2 then 'Result2'  
      ...  
      else 'Default Result'
```

[Else 'Default Result'] (not mandatory)

END

① SWAP emp's working in dept 10 to 30  
and Emp's from dept 30 to 30 and Emp's  
from dept 30 to 40.

At select Ename, case when Deptno=10 then '30'  
when Deptno=20 then '30'  
when Deptno=30 then '40'  
else 22nd to 30<sup>th</sup> position  
End  
From emp;

② SWAP Emp's working in Deptno 10 to 40  
and emp's working in deptno 40 to 10

At select ename, case when Deptno=10 then '40'  
when Deptno=40 then '10'  
Else '30'  
End

From emp;

Ans 1 didn't work because it didn't update  
Ans 2 works or last line didn't begin so  
add this line at third of minutes to add the

## Join

- \* This statement is used to retrieve the data from multiple tables simultaneously.

### Types of Joins:-

- 1) Cartesian Join or Cross Join
- 2) \* Inner Join or Equi Join
- 3) Outer Join
  - a. Left Outer Join
  - b. Right Outer Join
  - c. Full Outer Join
- 4) \* Self Join
- 5) Natural Join

### ① Cartesian Join (or) Cross Join

In Cartesian Join a record from table 1 will be merged with all the records of table 2.

- \* No. of columns in Result table will be

summation of columns] presentation in table  
and table 2.

\* No. of records in Result table will be  
the product of records present in table  
, and table 2.

Syntax :-

(American Standard, National Institute)

i) ANSI : Select Column-name  
From Table-name1 Cross Join Table-name2

ii) Oracle : Select Column-name  
From table-name1, table-name2

Ex: ① WAPTD Ename and Dname

Select Ename, Dname  
From emp, Dept;

Emp

Ename	Dname
A	20
B	10

Dept

Dno	Dname
10	D1
20	D2
30	D3

Ename	DNO	DNO	DName
A	20	10	D1
A	20	20	D2
A	20	30	B3
B	10	10	D1
B	10	20	D2
B	10	30	D3
C	30	10	D1 and D2
C	30	20	D2
C	30	10	D3

O/P

Ename	DName
A	D1
A	D2
A	D3
B	D1
B	D2
B	D3
C	D1
C	D2
C	D3

Disadvantage :- Repeated values and also  
emp can't work in different dept.

\* We are getting error records.

So, we don't use Cartesian Join.

\* In this Join we will be getting error records.

To overcome this we moved to inner join

## ② INNER JOIN :-

\* We use inner join to obtain only the matched records which as pair.

\*\* We use join condition to obtain the matched records.

Join Condition :- It is a condition on which we merge two tables to get only the matched records.

## Syntax for Join Condition

Table-name1 . column name = Table-name2 . Column name

Ex: Emp. DNO = Dept. DNO

## Syntax for Inner Join

① ANSI : select Column Name

From Table-Name1 Inner Join Table-Name2  
ON <Join-condition>

Ex: select \*

From emp inner join dept

ON Emp. Deptno = Dept. Deptno,

② Oracle: Select Column name

From "Table-Name1 , Table-Name2"

Where <Join-condition>

Ex: SELECT Ename and DName

3> select Ename, DName

1> From Emp, Dept

2> where Emp. DNO = Dept. DNO;

### ① Emp : Employee Table

Ename	DNO
A	20
B	10
C	30

### Dept

DNO	DName
10	D1
20	D2
30	D3

$$1) 20 = 10 \text{ F} \\ 20 \text{ T} \\ 30 \text{ F}$$

$$2) 10 = 10 \text{ T} \\ 20 \text{ F} \\ 30 \text{ F}$$

$$3) 30 = 10 \text{ F} \\ 20 \text{ F} \\ 30 \text{ T}$$

### ② where clause O/P

Ename	DNO	DNO	DName
A	20	20	D2
B	10	10	D1
C	30	30	D3

### ③ select O/P

Ename	DName
A	D2
B	D1
C	D3

① WAPTD Ename, sal and Dname for the all  
the employees.

A:  
= Select Ename, Sal, Dname  
From emp, Dept  
Where Emp. Deptno = Dept. Deptno;

② WAPTD Ename, Deptno, Dname

A:  
= Select Ename, Deptno, Dname  
From emp, dept  
Where Emp. Deptno = Dept. Deptno;

③ WAPTD Ename, Dname if emp's are working  
in dept 20.

A:  
= Select Ename, Dname  
From emp, Dept  
Where Emp. Deptno = Dept. Deptno and emp. Deptno  
= 20;

① WAPTD Name of the employee and his  
location of all the employees.

A/ Select ename, loc  
From emp, Dept

where Emp. Deptno = Dept Deptno

and bns. orgn. qm3 = orgn. qm3 where

(part 2)

② WAPTD DName and salary for all the employee

working in accounting.

A/ Select DName, sal  
From emp, Dept

where Emp. Deptno = Dept. Deptno

and DName = 'Accounting'  
and bns. orgn. qm3 = orgn. qm3 where

③ WAPTD Dname and Annual salary for all em

ployees whose salary is more than 2340.

A/ Select Dname, sal\*12 as "Annual Salary"

From emp, Dept

where Emp. Deptno = Dept. Deptno and sal > 2340,

and bns. orgn. qm3 = orgn. qm3 where

④ WAQTD Ename and Dname for employees having character 'A' in their Dname

A:  
= Select Ename, Dname  
From emp, Dept  
Where Emp. Deptno = Dept.Deptno and Dname  
like '%.A%'

⑤ WAQTD Ename and Dname for all the employees working as salesman

A:  
= Select Ename, Dname  
From emp, Dept  
Where Emp. Deptno = Dept.Deptno and job = 'SALESMAN'

⑥ WAQTD Dname & Job for all the employees whose Job and dname starts with character 'S'

A:  
= Select Dname, Job  
From emp, Dept  
Where emp.Deptno = Dept.Deptno and job like 'S%'

(10) ~~Want~~ Dname and empno for all the employees who's Empno are (7839, 7902), and are working in loc new York.

A: select Dname, empno  
from emp, Dept  
where Emp. Deptno = Dept. Deptno and empno in  
(7839, 7902) and loc = 'New York'

### ⑤ Natural Join

- \* In Natural Join we won't be writing any join condition.
- \* If the table contains similar columns we get the o/p of Inner Join.
- \* If the table is not having similar columns we will get the o/p of Cartesian join.
- why or when we use Natural Join?
  - \* whenever there is no table structure, we use natural join.

columns that are present  $\Rightarrow$  Table structure.

Syntax:

ANSI: Select Column-Name

From table-Name1 Natural JOIN Table-  
Name2;

Ex:- ① Select \*  $\rightarrow$  Inner Join

From Emp ~~and~~ Natural Join Deptno;

② Select \*  $\rightarrow$  Cartesian Join

From emp ~~and~~ Natural Join salgrade;

③ Outer Join: In outer join we get the  
unmatched records along with the matched  
records.

① Left Outer Join: In left outer join

we get unmatched records of left table  
along with matched records.

Syntax:

Left Outer Join =

1) ANSI : select Column-Name

from Table-Name1 Left [OUTER] JOIN Table  
2) Oracle : select Column-Name

ON < JOIN Condition >;

Ex:- Select \*

From Emp E Left Outer Join Dept D  
ON E.Deptno = D.Deptno;

2) Oracle : select Column-Name

From Table-Name1 , Table-Name2

where Table-Name1 . Col-Name = Table-Name2 . Col-Name(+);

Ex:-

Select \* from emp, dept

From Emp E, Dept D

where E.Deptno = D.Deptno(+);

①. Want details from both emp and dept table

along with unmatched records of emp table.

Select \*  
From Emp, Dept

where Emp. Deptno = Dept. Deptno (+);

EMP

Ename	DNO
A	20
B	NULL
C	30
D	NULL
E	10

DEPT

DName	DNO
D1	10
D2	20
D3	30
D4	40

EMP	Ename	DNO	DName	DNO
A		20	D2	20
C		30	D3	30
E		10	D1	10
B		NULL	NULL	NULL
D		NULL	NULL	NULL

Matched records

Unmatched  
records of emp table

② waqt Ename who is not working in any dept:

Select Ename

From emp,dept

where Emp. deptno = Dept. Deptno (+) and

is null.

Dname

② Right Outer Join: In right outer join we get unmatched records of right table along with matched records.

Syntax:

① ANSI : select column-name

From Table-Name1 Right [Outer] JOIN Table-Name2  
ON < JOIN condition >!

Ex:- select \*

From emp E Right Outer JOIN Dept D  
ON E.Deptno = D.Deptno;

② ORACLE : select column-name

From Table-Name1 , Table-Name2  
where Table-Name1. Col-Name(+) = Table-Name2. Col-Name(+)

Ex:- select \*

From Emp E ,Dept D

where E.Deptno(+) = D.Deptno;

① WAPTD details from both emp. and dept table along with unmatched records of dept table.

A: select \*

From Emp, Dept

where Emp. Deptno (+) = Dept. Deptno;

answ: Using JOIN operation

Ename	DNO	Dname	DNO
A	20	D2	20
C	30	D3	30
E	10	D1	10
NULL	NULL	D4	40

Matched Records

Unmatched Record of Dept table.

② WAPTD DName in which no employees working

A: Select Dname

From emp, Dept

where emp.deptno (+) ≠ Dept. Deptno and

Emp.deptno is null;

05	40	05	10
05	20	05	20
01	10	01	30
01	10	01	30
01	10	01	30

③ Full Outer Join :- To obtain Unmatched records of both the tables along with matched Records.

Syntax:-

1. ANSI : `Select Column-Name  
From Table-Name1 full [Outer] JOIN  
Table-Name2`

`ON Emp. Deptno = Dept. Deptno;`

Ex:- WAQTD Details from both Emp and dept table along with unmatched records of both the tables.

`Select *`

`From Emp FULL OUTER JOIN Dept`

`ON Emp. Deptno = Dept. Deptno;`

Ename	DNO	Dname	DNO	
A	20	D2	20	Unmatched Records
C	30	D3	30	
E	10	D1	10	
B	NULL	NULL	NULL	
D	NULL	NULL	NULL	Unmatched records of both table
NULL	NULL	D4	40	

## Based on Sub-Query

① WAQTD Dnames in which at least 3 emps are working.

② WAQTD details of Dept in which at least 2 manager's are working

① Select Dname of department

From dept  
where deptno in (select Deptno

From emp

Group by deptno

Having count(deptno) >= 2;

② Select Dname \*

From dept

where deptno in (select Deptno

From emp

Joining 3rd query  
Where Job in 'Manager'

Group by deptno

Having count(deptno) >= 2;

Information not needed

④

Self JOIN: Self Join is used to

join the same two tables or the table itself.

Why we use self Join?

\* If the data to be selected and condition to be executed is present in same table but in different record, we use self Join.

Syntax:

ANSI: select [column-name]

From Table-Name T1 JOIN Table-Name T2  
ON <Join Condition>;

Ex:

Select \*

From Emp E1 JOIN Emp E2

ON E1.MGR = E2.EmpNo;

ORACLE: select [column-name]

From Table-Name T1, Table-Name T2

where <Join-condition>;

Ex: select \*

From emp E1, Emp E2

where E1.MGR = E2.EMPNO

① WATQD Ename and Manager's Name.

AT EMP E1

EID	ENAME	MGR
1	Smith	2
2	Dinga	4
3	Dhoni	4
4	Raju	5
5	King	NULL

EMP E2

EID	ENAME	MGR
1	Smith	2
2	Dinga	4
3	Dhoni	4
4	Raju	5
5	King	NULL

Select E1.ename, E2.ename 'Manager Name'

From emp E1, EMP E2

Where E1.MGR = E2.EID !

Impress 20

1) $a = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$	2) $4 = \begin{bmatrix} F \\ T \\ F \\ F \\ F \end{bmatrix}$	3) $4 = \begin{bmatrix} F \\ F \\ F \\ T \\ F \end{bmatrix}$	4) $4 = \begin{bmatrix} F \\ F \\ F \\ T \\ F \end{bmatrix}$	5) $4 = \begin{bmatrix} F \\ F \\ F \\ T \\ F \end{bmatrix}$

1) $a = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$	2) $4 = \begin{bmatrix} F \\ F \\ F \\ T \\ F \end{bmatrix}$

5) $\text{Null} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$	{ Null.

E1.EID	E1.Ename	E1.MGR	E2.BID	E2.Ename	E2.MGR
1	Smith	2	2	Dinga	4
2	Dinga	4	4	Raju	5
3	Dhoni	4	4	Raju	5
4	Raju	5	5	King	Null

Employees Details

Managers Details

O/P

	<u>Ename</u>	<u>Manager name</u>
1	Smith	Smith
2	Dinga	Dinga
3	Dhoni	Raju
4	Raju	King

9014	Smith	413
9015	Dinga	414
9016	Raju	415
9017	King	416

① WAP QTD Emp's Sal and Manager's Sal

A:

Select E1.Ename, E1.Sal, E2.Manager  
From Emp E1, Emp E2  
where E1.MGR = E2.Empno

where E1.MGR = E2.Empno

① **WATD**: Name of the employee and his manager's name if employee is working as clerk.

A: select E1.Ename, E2.Ename Manager\_name  
From Emp E1, Emp E2  
where E1.MGR=E2.Empno and E1.job IN 'CLERK';

② **WATD**: Name of the employee and Manager's designation if manager works in dept 10 or 20.

A: Select E1.Ename, E2.Job  
From emp E1, Emp E2  
where E1.MGR=E2.Empno and E2.dept IN (10,20);

③ **WATD**: Names of the emp and managers salary if employee and manager both earn more than 2300.

A: Select E1.Ename, E2.Sal Manager\_Sal  
from emp E1, emp E2 where E1.Sal > 2300 and E2.Sal > 2300;

4) WAQTD Emp.ename and Manager's hire date if employee was hired before 1981

A:- Select E1.ename, E2.Hiredate 'ManagerisHired'  
From Emp E1, Emp E2  
where E1.MGR = E2.Empno and E1.hiredate < 1981

5) WAQTD Emp. Name and manager's Comm if employee works as salesman and manager works in dept 30.

A:- Select E1.ename, E2.comm 'Manager-comm'  
From emp E1, Emp E2  
where E1.MGR = E2.Empno and E1.Job in 'salesman'  
and E2.deptno in (30);

6) WAQTD Emp.ename and manager name and their salaries if employee earns more than their manager.

A:- Select E1.ename, E2.ename 'Managername', E1.sal, E2.sal  
From emp E1, Emp E2  
where E1.MGR = E2.Empno and E1.sal > E2.sal;

⑦ WAPTD Emp Name and Hiredate, Manager name and hiredate if manager was hired before employee.

A:   
 select e1.Ename, E1.hiredate, E2.ename  
 ~manager,  
 and E2.hiredate  
 from emp E1, emp E2  
 where E1.MGR = E2.Empno and E2.ename < e1.hire  
 date;

⑧ WAPTD Emp Name and manager name if both are working same job.

A:   
 select e1.ename, e2.ename Manager-Name  
 From emp E1, EMP E2  
 where E1.MGR = E2.Empno and E1.Job = E2.Job;

⑨ WAPTD Emp Name and Manager name if manager is working as Actual manager.

A:   
 select e1.ename, e2.ename  
 From emp E1, emp E2  
 where E1.MGR = E2.Empno and E2.Job = 'man-  
 ager';

Q10) Write a query to find manager's name along with their annual salaries if employee works in dept 10, 20 and manager's sal is greater than employee's salary.

A:  
Select e1.ename, e2.ename, e1.sal\*12, e2.sal\*12  
from emp e1, emp e2  
where E1.MGR = E2.empno and e1.deptno in (10,20)  
and e2.sal > e1.sal;

Q11) Write Employee's Name and manager's designation for all the employees.

A:  
Select e1.ename, e2.job  
from emp e1, emp e2  
where E1.MGR = E2.empno ;

Q12) Write Employee's Name and manager's salary for all the employees if manager's salary ends with 50.

A:  
Select e1.ename, e2.sal MGR-sal  
From emp E1, Emp E2  
where E1.MGR = E2.empno and substr(E2.sal,-2,) = '50'

## Joining Multiple Table

13 WAQTD Ename and Manager name.

Select e1.ename, e2.ename

From emp e1, emp e2

where e1.MGR = e2.Empno

14 WAQTD Ename and Dname.

Select e1.ename, d1.Dname

From emp e1, dept d1

where e1.empdeptno = d1.deptno;

15 WAQTD Ename, Manager's name and employees

Dname.

Select e1.ename, e2.ename MGR-name, d1.Dname

From emp e1, emp e2, dept d1

where e1.MGR = e2.empno and e1.Depthno = d1.Depthno;

Dname

Ename

Ename

Dept Dept from emp Elms q Emp E1 inner join

El. MGR = E2.empno

El. Depthno = D1.Depthno

\* To Join N no. of tables

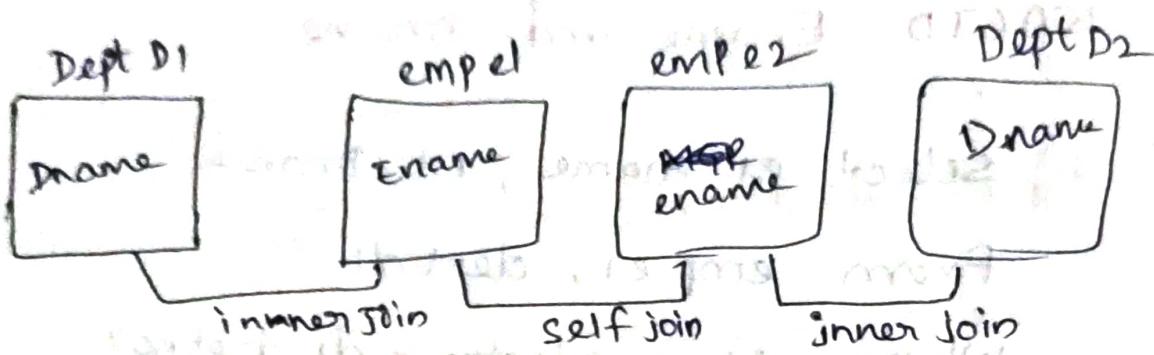
we write (N-1) Join condition.

we can join upto 256 tables using inner  
and emp dname join

(16) WAPTD Ename, Manager's Name and Manager

~~dept~~ dname (from which department)

A:-



Select e1.ename, D1.Dept, e2.ename MGR.name,

From

~~where~~ emp e1, emp e2, Dept D1, Dept D2

where ~~e1.MGR = e2.empno and e1.~~Deptno = D1.Deptno  
~~e1.empno = e2.empno and e1.~~Deptno = D2.Deptno

in Deptno and e2.Deptno = D2.Deptno

(17) WAPTD Ename, Manager's Name and their

Dname's if emp earns more than 2000 and  
manager working in dept 20.

A:- Select e1.ename, e2.ename MGR.name, D1.Dname,  
D2.Dname MGR.Dname

From emp e1, emp e2, Dept D1, Dept D2

where e1.MGR = e2.Empno and e1.deptno = d1.deptno  
and e2.deptno = d2.deptno and e1.sal > 2000 and  
e2.deptno in 20, 10, 40 and their manager

18) WA.QTD Ename, manager's name and their  
dname if emp earning more than smith  
and manager earning more than Allen.

A)  
Select e1.ename, e2.ename, MGR-name, d1.dname,

d2.dname MGR-Dname

From emp e1, emp e2, Dept D1, Dept D2

Where e1.MGR = e2.empno and e1.deptno = D1.Deptno and

e2.deptno = d2.deptno and e1.sal > (select sal  
from emp where ename = 'smith')

and e2.sal > (select sal  
from emp where ename = 'Allen')

19) WA.QTD ename, Manager's name and their  
loc if emp working in deptno 10 or 30 and  
manager earning more than ford and emp working  
in loc New York or Chicago.

Q) select e1.ename, e2.ename MGR\_name, d1.loc, d2.loc  
From emp e1, emp e2, dept d1, dept d2  
where e1.MGR = e2.empno and e1.deptno = d1.deptno  
and e2.deptno = d2.deptno and e1.deptno in (10, 30)

and e2.sal > (select sal  
From emp  
where ename = 'ford') and e1.loc  
in (New York, Chicago)  
deptno in (select deptno  
From emp  
where ename = 'ford')

(ii) WAPTD. Ename, Manager's name and manager's  
manager name along with their dname's if emp  
earns more than 1000 and manager earns more  
than allen and manager's manager working  
in new york or chicago.

A) Select e1.ename, e2.ename MGR\_name, e3.ename MGR\_MGR\_name,  
d1.dname, d2.dname MGR\_dname, d3.dname MGR\_MGR\_dname  
From emp e1, emp e2, emp e3, dept d1, dept d2, dept d3  
where E1.MGR = E2.empno and E2.MGR = E3.empno and  
e1.deptno = d1.deptno and e2.deptno = d2.deptno and e3.deptno  
= d3.deptno and e1.sal > 1000 and e2.sal >

(2) select sal

From emp

where ename = 'Allen' and d3.loc in ('New York',  
'Chicago')

(22) Want ename, Manager's Name and Manager's  
manager name along with their loc, if the emp  
hired before martin and manager working in acco  
unting or sales dept and manager's manager  
earning sal more than smith

A) Select e1.ename, e2.ename, e3.ename, d1.loc, d2.loc, d3.loc  
(from emp e1, emp e2, emp e3, dept d1, dept d2, dept d3  
where e1.MGR = e2.empno and e2.MGR = e3.empno and  
e1.deptno = d1.deptno and e2.deptno = d2.deptno and e3.deptno = d3.deptno  
and e1.hiredate < (select hiredate  
From emp  
where ename = 'Martin')

and d2.deptno in ('Accounting', 'Sales') and

e3.sal > (select sal  
From emp  
where ename = 'smith')

for i in range(1, 10):  
 print(i)

for i in range(1, 10):  
 print(i)

## Using ANSI Syntax

Q) WAPTD Ename, Manager's Name and empl'dame  
-me and manager's Dname.

A) Select e1.ename, e2.ename, d1.dname, d2.dname  
From emp e1 JOIN emp e2  
ON e1.empno = e2.empno INNER JOIN Dept D1  
ON E1Deptno = D1Deptno INNER JOIN Dept D2  
ON E2Deptno = D2Deptno

If extra conditions use where clause  
where e1.sal > 1000 and e2.deptno in (20, 30)

Q) WAPTD Ename, Manager's Name and manager's  
manager Name.

A) Select e1.ename, e2.ename, e3.ename  
From emp e1 JOIN emp e2  
ON e1.MGR = e2.empno JOIN emp e3  
ON e2.MGR = e3.empno

Q) WAPTD Ename, Manager's name and manager's  
manager name along with their Dname's if emp  
earns more than 1000 and Manager earns more

than Allen and manager's manager working in New York or Chicago.

A) Select e1.ename, e2.ename, e3.ename, d1.dname, d2.dn  
ame, d3.dname..

From emp e1 Join emp 2

ON e1.MGR = e2.empno join emp e3

ON e2.MGR = e3.empno Join Dept D1

ON e1.deptno = D1.deptno Join Dept D2

ON e2.deptno = D2.deptno Join Dept D3

ON e3.deptno = D3.deptno

where e1.sal > 1000 and e2.sal > (select sal

From emp

at least 3 times salary of  
and d3.loc in ('New York', 'Chicago');

# Data Definition Language (DDL)

\* This statement is used to create, rename, alter or delete an object from database.

\* There are 5 statements:

- 1) Create
- 2) Rename
- 3) Alter
- 4) Truncate
- 5) Drop

① Create: This statement is used to create an object in database.

Ex: Table  
view

## Syntax:

CREATE TABLE table name

C

column-name-1 Datatype NOTNULL/NULL,  
column-name-2 Datatype NOTNULL/NULL,

column-name -> Datatype NOTNULL / [NULL], \*  
 constraint constraint-ref-name Unique (column-name),  
 constraint constraint-ref-name check (condition),  
 constraint constraint-ref-name Primarykey (column-name),  
 constraint constraint-ref-name foreign key (column-name)

References parent-table-name (column-name)

## Blue Print

Table-Name :- Customer

No. of col :- 4

col-names	column	cName	cnID	Loc
Customer	cID	varchar(20)	number(10)	varchar(10)
Customer	cName	Not null	Not null	null
Customer	cnID	Not null	length(cnID)=10	unique

Constraints

Primary Key  
(cID-PK)

Unique

(cnID-U)

check (length(cnID)=10)

(cnID-C)

Eg Create table customer

(Customer information registration system)

CID Number(3) Not Null,

cName Varchar(20) Not Null,

CNO Number(10) Not Null,

Loc Varchar(40)

Constraint CID-PK Primary Key (CID),

Constraint CNO-U Unique (CNO),

Constraint CNO-C check (Length(CNO)=10)

)

\* Descr customer;

Name	Type	Null?
CID	Number(3)	Not null
cName	Varchar(20)	Not null
CNO	Number(10)	Not null
Loc	Varchar(40)	Not null

① Select \* from user\_constraints;

① Create a table Product with column names  
PID, PName, Price, Discount.

2) Create table Product

(

PID Number(3) NOT NULL,

PName Varchar(20) NOT NULL,

Price Number(5) NOT NULL,

Discount Number(2)

Constraint PID - PK Primary Key (PID),

constraint Dis - check (length(Discoun)=2)

);

② **Rename**: This statement is used to rename the current table name to new name.

Syntax: `Alter Table Table-name RENAME TO New-table-name;`

Ex: `Alter Table customer RENAME TO cust;`

③ **Alter**: This statement is used to modify the objects in database.

Syntax:

① To ADD A Col:

`Alter Table - table-name`

`ADD Column-Name Datatype [Null / Not Null];`

② To DROP A Col:

`Alter Table - table-name`

`Drop column - column-name;`

③ To change the datatype:

`Alter Table - table-name`

`modify Column-Name new-datatype;`

④ To change the NOT NULL constraint:

Alter table table-name

Modify column\_name existing datatype  
Null/NOTNULL;

⑤ To Rename the Column:

Alter table table-name

Rename column current-name to new-name;

⑥ To Modify Constraints:

a) Alter Table Table name

ADD constraint constraint ref-name unique(column-name);

b) Alter table table name

ADD constraint constraint ref-name check(condition);

c) Alter table table name

ADD constraint constraint ref-name Primary key  
(column-name);

d) Alter table table\_name

ADD constraint constraint ref-name foreign key

(column\_name) References parent\_table\_name(column\_name);

- Ex:-
- ① Alter table cust  
ADD MailID varchar(50) NULL NOT NULL
- ② Alter table cust  
Drop column loc;
- ③ Alter table cust  
Modify CID Number(4);
- ④ Alter table cust  
modify ~~MailID~~ varchar(50) NOT NULL;
- ⑤ Alter table ~~for~~ cust  
Rename <sup>column</sup> CNO to PHNO;
- ⑥ a) Alter table cust  
ADD constraint MailIDUnique(MailID)

④ Truncate: It is used to delete all the records from the table permanently.

Syntax:

~~Truncate Table Table name;~~  
Ex:- Truncate Table cust;

⑤ DROP: This statement is used to delete the object i.e., tables from the database along with its structure.

Syntax: Drop table Table name;

To Recover the table: (only in oracle)

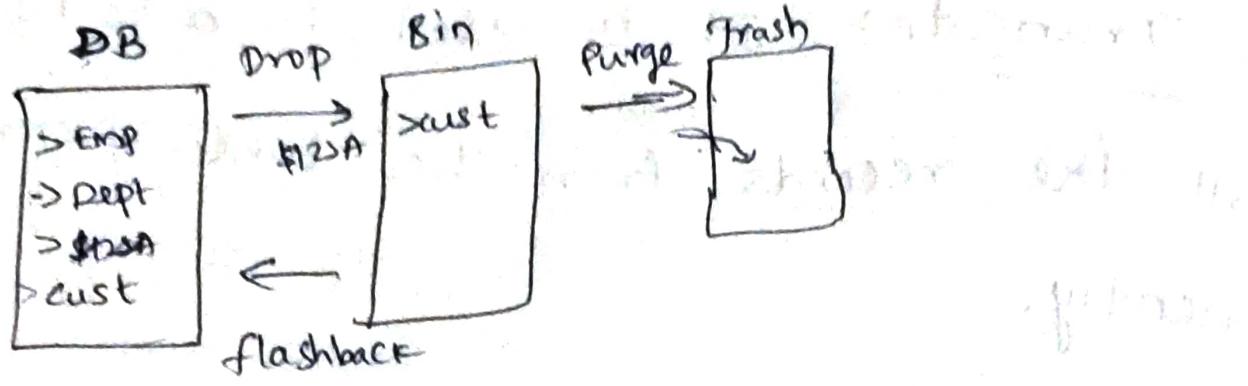
Syntax: Flashback Table Table name

To before drop

[Rename to new name];

To Drop the table from Recyclebin

Syntax: Purge Table Table name;



Ex:- ① Drop table cust;

② Flashback table cust

To before drop

Rename table to customers;

③ Purge table cust;

## Data Manipulation Language (DML)

\* This statement is used to insert, update or delete the records from the table.

\* There are three statements:

1) Insert

2) Update

3) Delete

① Insert: This statement is used to insert the records into the table.

## Syntax:-

- ① Insert into table name values ( $v_1, v_2, \dots, v_n$ );  
(or)
- ② Insert into table name  $\star$  (col<sub>1</sub>, col<sub>2</sub>, ... col<sub>n</sub>)  
values ( $v_1, v_2, \dots, v_n$ );  
(or)
- ③ Insert into table name (col<sub>1</sub>, col<sub>2</sub>, ... col<sub>n</sub>)  
Values (& col<sub>1</sub>, & col<sub>2</sub>, ... & col<sub>n</sub>);

Note:- Commit is used to save data permanently.

-tly. Syntax:- Commit;

Ex:- ① Insert into cust values (1, 'Pavan', 1234567890,  
'pavan@gmail.com');

commit;

② Insert into table name (cID, cname, PHno, mailid)

values (1, 'Pavan', 1234567890, 'pavan@gmail.com');

Commit;

③ Insert into cust (cID, cname, PHno, mailid)

values (&cID, &cname, &PHno, &mailid);

Enter cID:

Enter cname:

Enter PHno:

commit;

② Update:- This statement is used to update the records in the table.

Syntax:- Update table name

Set col1 = v1, col2 = v2, ... coln = vn

[where <filter condition>];

one column

Ex:- Update cust

Set phno = 5555566666

Where ename in 'King';

Commit;

Two columns

update cust

Set phno = 5555566666, mailid = 'MR.King@gmail.com'

Where ename in 'King';

Commit;

③ Delete:- This statement is used to delete a particular record from the table.

System :- Delete command with filter clause  
From table name  
[where < filter condition>];

Ex :- Delete from table\_name  
From cust

where cname in ('King', 'John')

commit;

\*\*\*  
① Diff. b/w Truncate, delete, DROP.

Truncate :- It is used to delete the all

the records <sup>permanently</sup> in a Table. It is a DDL statement.

\* Truncate : is auto commit statement.

Delete :- This statement is used to delete a particular record <sup>from the table</sup> in a table. It is a DML statement. we need to use commit statement.

to delete permanently.

\* Delete is not auto commit statement.

DROP :- This statement is used to drop table.

(a) delete @ the table

statement. \* Drop is auto commit statement.

## ② Diff. b/w DDL and DML

### DDL

### DML

\* DDL is used to create, rename, alter, truncate, drop an object from the table database.

\* DML is used to insert, update, delete the records from the table in the database permanently using commit.

\* It has 5 statements.

\* It has 3 statements.

\* DDL is auto Commit

\* DML statement is not auto Commit.

## ③

## Transaction Control Language (TCL)

\* There are three statements:

1) Commit

2) Savepoint

3) Rollback

Syntax: Commit;

\* This statement is used to save the transaction on a database.

## ② Savepoint:

Syntax: Savepoint Savepoint-name;

- \* This statement is used to mark the position on database.

## ③ Rollback:

Syntax: ① Rollback; or ② Rollback to Savepoint-name;

- \* This statement is used to go back or Undo to the previous Savepoint.

Ex:- Insert into cust values('A', 123);

Savepoint R1

Insert into cust values('B', 234);

Savepoint R2

Insert into cust values('C', 345);

Savepoint R3;

Deleted  
Perma  
nently.

SQL> Rollback to R2;

## ④ Data Control Language (DCL)

\* This statement is used to give the permission or take back permission from another user.

\* There are 2 statements:-

- 1) Grant
- 2) Revoke

① Grant: This statement is used to give the permission to another user.

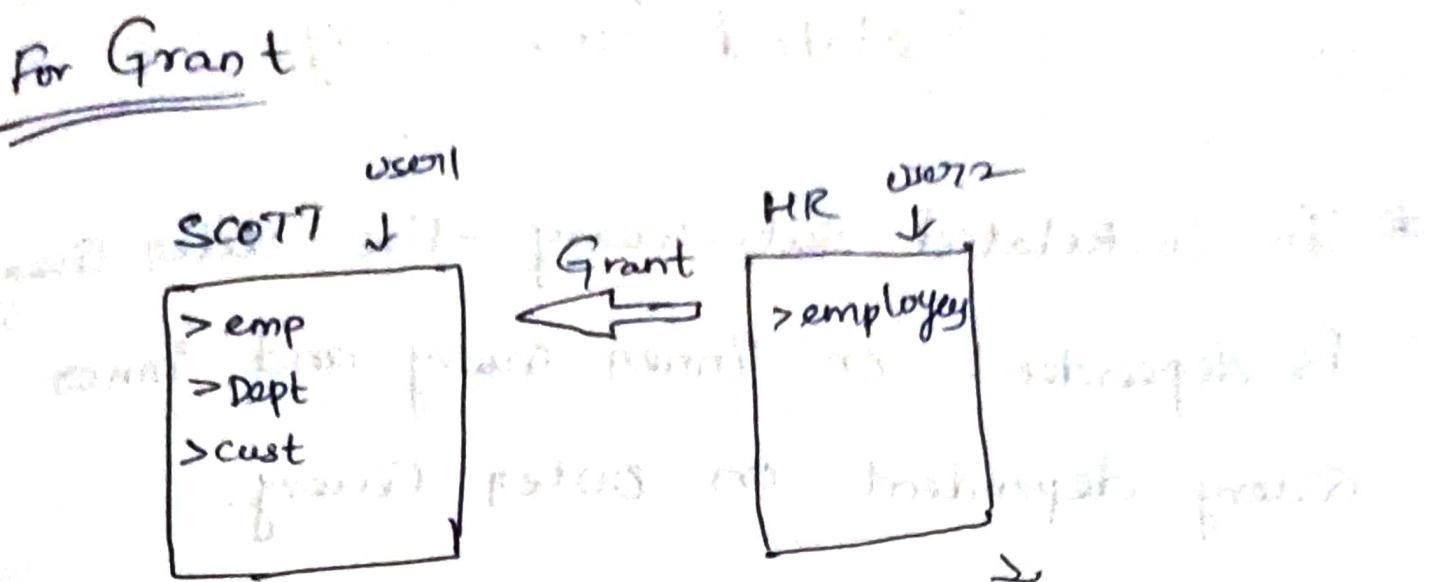
② Revoke: This statement is used to take back the permission from another user.

### GRANT Syntax

Syntax: Grant Sql Statement ON table-name  
To user-name;

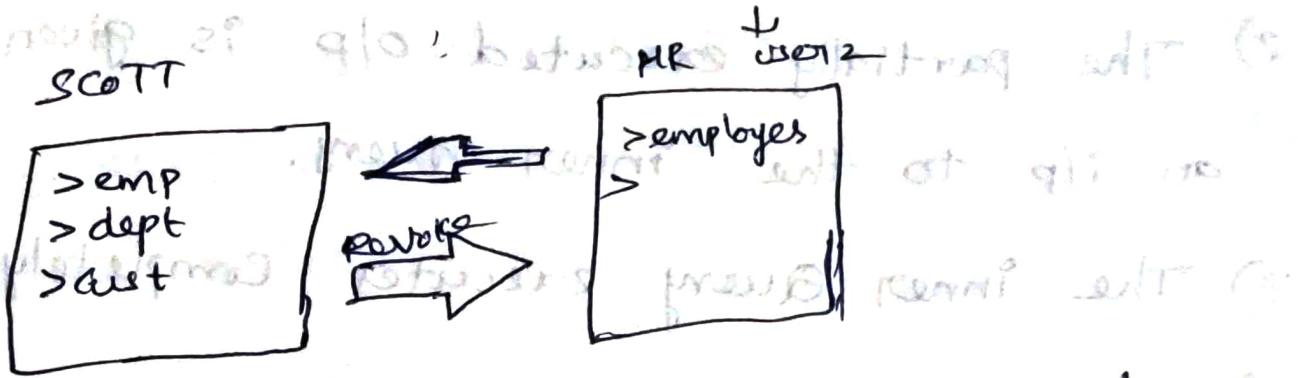
### Revoke Syntax

Syntax: Revoke Sql Statement ON table-name  
From user-name;



Grant select on employees  
To SCOTT ;  
From HR.employees;

for Revoke



Revoke select on employees  
From SCOTT ;

This will be now glo. access  
to emp, dept, cust & employees  
table. This will be now glo. access  
to emp, dept, cust & employees  
table.

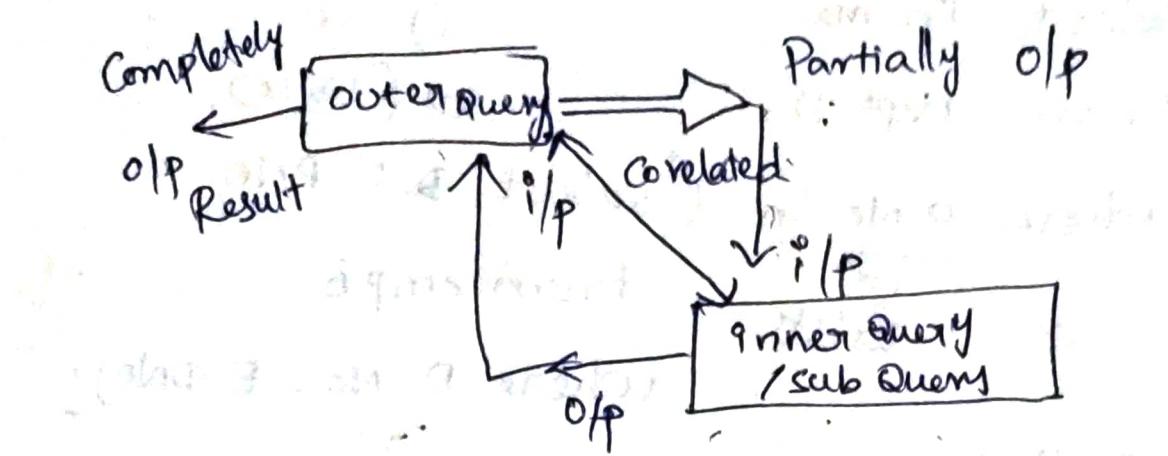
Ques

## Co-Related Sub-Query

- \* In Co-Related sub-Query the outer Query is dependent on inner-Query and inner Query dependent on outer Query.

### Working Procedure:

- 1) In co-related sub-Query the outer Query executes first but partially.
- 2) The partially executed O/P is given as an i/p to the inner query.
- 3) The inner Query executes completely.
- 4) The o/p of inner Query is given as i/p to outer Query.
- 5) The outer Query execute completely and generates the o/p.
- 6) This o/p will be the result.
- 7) Therefore the outer Query & Inner Query both are interdependent i.e., both are co-related.



\*\*\*

Note:-

- 1) In co-related sub-query the Join condition must be written only in the inner query.
- 2) The Join condition is written in such a way that it includes a column that belongs to the outer query.

① which Dname in which emp's are working

Ans

Emp		Dept	
Ename	DNo	Dname	DNo
A	20	D1	10
B	30	D2	20
C	10	D3	30
D	10	D4	40
E	30		

select Dname  
From Dept D

where D.No in

( $\exists$ )  
exists

(select  $\rightarrow$  E.PNO

from emp E

where D.No = E.DNo)

D)  $10 = 10.T$   
 $10 = 10, T$

D)  $10 = \begin{matrix} 20 \\ 30 \\ 10 \\ 10 \end{matrix} F$   
 $10 = \begin{matrix} T \\ T \\ F \\ F \end{matrix}$

2)  $20 IN = 20$

2)  $20 = \begin{matrix} 20 \\ 30 \\ 10 \\ 10 \end{matrix} F$   
 $20 = \begin{matrix} F \\ F \\ P \\ P \end{matrix}$

3)  $30 = 30T$

3)  $30 = \begin{matrix} 20 \\ 30 \\ 10 \\ 10 \end{matrix} F$   
 $30 = \begin{matrix} T \\ T \\ F \\ F \end{matrix}$

4)  $40 IN Nut$

4)  $40 = \begin{matrix} 20 \\ 30 \\ 10 \\ 10 \end{matrix} F$   
 $40 = \begin{matrix} F \\ F \\ F \\ F \end{matrix}$

qp

D1

D2

D3

② what Dname in which employees not working.

A: select Dname

From Dept D

where D.DNO NOT IN (Select E-DNO



dp; D4

(NOT exists)

From emp E

where D.DNO = E.DNo

### Exists Operator

- \* It is a Unary operator which can accept values at RHS.

- \* It returns true only when sub-Query is returning some value.

### Not-exists Operator

- \* It is a Unary operator.

- \* Which returns true if sub-Query is not returning any value.

① WAP TO 10th Max. 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

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

where sal >

(select max(sal))

From emp

where sal < (select max(sal))

From emp )))))))))

where sal < )))))))))

Q) WAPTD 5th min sal

select min(sal)

From emp

where sal > (select min(sal))

From emp

where sal > (select min(sal))

From emp

where sal > (select min(sal))

From emp

where sal >

(select ~~min~~ min(sal))

From emp ))))))

Syntax to find Nth Max. salary

Select El. sal.

From emp El

where (n-1) in ( select count(Distinct El. sal)

From emp El

(El. sal < Ed. sal) } where El. sal < Ed. sal ) if

Q) WAPTD 5th max. salis present

Emp El

Sal

2000 4th

4000 2nd

2000 4th

1000 5th

5000 1st

3000 3rd

4000 2nd

### Emp E2

2000  
4000  
2000  
1000  
5000  
3000  
4000

Select E1. sal

From emp e1

where (4) in (Select count(Distinct E2.sal))

1) 4 in 3FM table from emp e2

2) 4 in 1F

3) 4 in 3F where E1.sal < E2.sal,  
4) 4 in 4F

1) 2000 < 2000 F

2) 4000 < 2000 F

4000 T

2000 F

1000 F

5000 T

3000 T

4000 T (Rejected)

2) 4000 < 2000 F

4000 F

2000 F

1000 F

5000 T

3000 F

4000 F

3) 2000 < 2000 F

4000 T

2000 F

1000 F

5000 T

3000 T

4000 T

4) 1000 < 2000 T

4000 T

2000 T

1000 F

5000 T

3000 T

4000 T (Rejected)

Syntax to find Nth min salary

Select E1. sal

From emp e1

where (n-1) in (Select count(Distinct E2.sal))

From emp e2

where E1.sal > E2.sal);

① WAPTD 3rd min sal.

A<sup>2</sup>

```
select E1.sal  
From emp E1  
where (3-1) IN (select count(Distinct E2.sal))  
from emp E2  
where E1.sal > E2.sal);
```

② WAPTD 11th max sal using correlated sub-Query.

A<sup>2</sup>

```
select E1.sal  
From emp E1  
where (11-1) IN (select count(Distinct E2.sal))  
from emp E2  
where E1.sal < E2.sal);
```

③ WAPTD 20th min sal using co-related sub-Query.

```
Select E1.sal  
From emp E1  
where (20-1) IN (select count(Distinct E2.sal))  
from emp E2  
where E1.sal > E2.sal);
```

④ WAPTD Ename who is getting 5th max sal.

~~Select Et.ename~~

~~From emp E1~~

~~Where E1sal IN (Select E2sal~~

E1.ename

From E1

Where 4 IN (Select count(Distinct E2.sal))

From emp E2

Where E1.sal < E2.sal);

⑤ W.A.Q.5. top 5 max. salaries using Co-  
relation

A:

Select E1.sal

From emp E1

where (select Count(Distinct E2.sal)

from emp E2

where E1.sal < E2.sal) IN (0,1,2,3,4)

## Pseudo Columns

- \* Pseudo columns are the false columns that are present in each and every table and must be called "explicitly".
- \* Pseudo columns cannot be seen without calling them.

### Types of Pseudo columns

- 1) ROWID: It is a 18 digit address in which the record is present or the record is stored in the memory.
- 2) RowNum

① RowID: RowID is an 18 digit address in which the record is present or the record is stored in the memory.

select RowID, Emp.\*

From emp;

Note:-

- 1) Rowid is one of the way to access or delete the record.

- 2) Rowid is unique.
- 3) Rowid is present for each and every row.  
- d. but not for all rows (constant)
- 4) Rowid is generated at the time of insert.
- 5) Rowid cannot be inserted, updated or deleted.
- 6) Empty table will not be having rowids.
- 7) RowID is static in nature. (Constant)
- 8) RowID can be used to identify a record uniquely from the table when there is no key attribute or Primary key.

### Rownum

- \* Rownum acts as serial number to the result table.
- \* Rownum is used as record number that is assigned to the result table.
- \* Rownum is dynamic in nature (keeps on changing)

\* Rownum is generated at the time of execution.

\* Rownum always starts with 1.

\* Rownum cannot be duplicated.

\* Rownum gets incremented after it is assigned.

\* Rownum changes because it depends on result table.  
Q) What Rownum along with all the details of emp's.

A:-

```
select * Rownum, Emp.*  
From emp;
```

Q) What the first five records from emp table.

A:-

```
* Rownum, Emp.*  
Select * Rownum, Emp.*  
From emp  
where Rownum in (1,2,3,4,5);
```

(or)  
Rownum in < 6 (or) rounum in <=5;

① waqtD First three records.

Ename	sal
A	1200
B	1000
C	200
D	300
E	400
	300

Select \* from emp;

From emp

Where Rownum < 4;

Rownum = 1      Rownum = 4

1	A	1000	1 < 4 T
+1			
2	B	200	2 < 4 T
+1			
3	C	300	3 < 4 T
+1			
4	D	400	4 < 4 F
	E	300	4 < 4 F

O/P

1 A 1000

2 B 200

3 C 300

② waqtD the third record.

Select \* from emp;

From emp

Where Rownum = 3;

1 = 3 F  
A 1000

B 200

C 300

O/P :-

No Rows Selected

Ques. select \* from Emp;

From ( select Rownum SLNO from Emp )  
From Emp

where SLNO = 3;

(3)

SLNO    Ename    SAL

O/P :- S C 300

1	A	100
2	B	200
3	C	300
4	D	400
5	E	300

To make Rownum as static  
= = = = =

\* Take a table and assign rownum to a

given table.

\* change the rownum to any other name

\* by using ~~as~~ Alias (SLNO).

\* Use this as a subquery in from

clause of outer query.

\* In the outer query use the alias name as the condition.

① waqtid Rownum along with all the details of Emp's.

A: Select Rownum, Emp.\*  
From emp;

② waqtid the first five records from emp table.

A: Select \*  
From emp  
where Rownum < 6;

③ waqtid the first seven records from emp table.

A: Select \*  
From emp  
where Rownum < 8;

④ waqtid the first third record from emp table.

Select \*  
From (Select Rownum SLNO, Emp.\*  
From emp)

where SLNO = 3;

① WAQTD 5th record from the emp table.

A:- select \*

From (Select Rownum SLNO, Emp.\*

From emp)

where SLNO = 5;

② WAQTD Ename , sal from seventh record.

A:- Select Ename, sal

From (select Rownum SLNO, Emp.\*

From emp)

where SLNO = 7;

③ WAQTD of 1st, 3rd,5th and 8th record.

A:- Select \*

From (select Rownum SLNO, Emp.\*

From emp)

where SLNO in (1,3,5,8);

④ WAQTD Ename of 8th , 7th and 6th record.

A:- Select \* Ename

From (select Rownum SLNO, Emp.\*

From emp)

Where SLNO in (8,7,6);

⑥ WaQTD Top 3 Records from emp table  
\* false

A:-  
Select \*  
From emp  
Where Rownum <= 3,  
(2 = static answer)

⑦ waQTD 10th max. salary  
\* false

A:-  
Select E1.Sal  
From emp E1  
Where (10-1) IN (Select count(Distinct E2.Sal)  
From emp E2  
Where E1.Sal < E2.Sal),  
order by E1.Sal desc

⑧ waQTD 8th min. salary  
\* false

A:-  
Select E1.Sal  
From emp E1  
Where (8-1) IN (Select count(Distinct E2.Sal)  
From emp E2  
Where E1.Sal > E2.Sal),  
order by E1.Sal asc

\* false

(8-1) min. static answer

① ~~Want~~ SAL in ASC order.

Select SAL

From emp

Order by SAL ASC

Order by clause

\* It is used to sort the records in ascending (ASC) or descending (DESC) order.

- 1) Order by clause must be written as last clause in the statement.
- 2) Order by clause executes after the select clause.
- 3) By default order by clause sort the records in Ascending order.

4) we can pass alias name in order by clause

Syntax :-

Select group by expression / group-function

From table-name

[where <filter-condition>]

[Group by column-name/expression]

[Having <group-filter-condition>]

Order by column-name [ASC]/DESC

## Order of execution

- 1) From [Row by Row]
  - 2) where (if used) [Row by Row]
  - 3) Group by (if used) [Group by Group]
  - 4) Having (if used) [Group by Group]
  - 5) Select [Group by Group]
  - 6) Order by [Column names in order]
- \* We can pass multiple column names in order by clause.

① WAQTD Ename in ASC order

Select Ename  
From emp  
Order by Ename Asc;

⑥ WAQTD Annual sal in desc order

Select sal\*12 "Annual sal"

From emp

Order by sal\*12 desc

(or)  
[Annualsal]

Ex: select annualsal from emp order by annualsal desc;

from emp order by annualsal desc;

annualsal from emp order by annualsal desc;

① want 5<sup>th</sup> max. salary.

EMP  
SAL

3000 - 2nd  
2000 - 3rd  
1000 - 5th  
4000 - 1st  
1000 - 5th  
1500 - 4th

Select distinct & sal }

From emp

order by SAL DESC

O/P

SAL

4000  
3000  
2000  
1500  
1000  
\*

Assign Rownum

Select Rownum SLNO, Sal

From (Select Distinct sal

From emp

order by SAL DESC)

O/P

SLNO    SAL  
1    4000  
2    3000  
3    2000  
4    1500  
5    1000

Final Program

Select sal

From (select Rownum SLNO, sal

From (Select Distinct sal

From emp

order by sal DESC))

where SLNO = 5;

O/P

SAL

5 - 1000

## Syntax for nth max

To find nth max. and nth min. salary using Rownum Concept.

select sal

From (select Rownum slno, sal

From (select Distinct sal

From emp

order by sal Desc)

where slno = n;

## Syntax for nth Min

select sal

From (select Rownum slno, sal

From (select Distinct sal

From emp

order by sal Asc)

where slno = n;

⑥ waqtid Name of emp getting 4th max sal

Ans  
Select ~~Sal~~ name

From (select Rownum SLNO, Sal, Ename)

From (select Distinct Sal, ~~Ename~~)

From emp

Order by Sal Desc))

Where SLNO = 4;

Ans query

Select Ename

From emp

Where Sal in (select Sal

From (select Rownum SLNO, Sal)

From (select Distinct Sal

From emp

Order by Sal Desc))

Where SLNO = 4;

Ans query

① waqtid 8th min sal

Ans  
Select SAL

From (select Rownum SLNO, Sal)

From (select Distinct Sal)

From emp

Order by Sal Asc))

Where SLNO = 4;

③ WAQTD TOP 5 max. sal

A) select sal  
From (select Rownum slno, sal  
From (select Distinct sal  
From emp  
order by sal desc))  
where slno < 6;

④ WAQTD TOP 3 min. sal

select sal  
From (select Rownum slno, sal  
From (select Distinct sal  
From emp  
order by sal ASC))  
where slno <= 3;

⑤ WAQTD 3rd, 5th max sal

A) where sal in (3,5);

⑥ WAQTD Dname of emp who is getting 7th min sal.

Select Dname  
From Dept  
Where Deptno in (select Deptno  
From emp  
Join with where sal in (select sal

Connect system/tiger

Grant Create  
view to scott  
connect scott/tiger

From (select Rownum, slno

From (select Dist, Sal

From emp  
order by slno)

Where slno = 7))

\* View :- Views are the virtual table

which can be created and re-used whenever we are dealing with part of a table.

Syntax:-

Create View View-name

AS select stmt;

select stmt;

\* Diff b/w Table and view.

\* Advantages of view's.

# PL/SQL ("Procedural Language")

INDEX: When we have Index, It enhance (increases) the speed of searching in the database. i.e., there are lots of tables in database, so, for easy of access, we use Index.

\* Index are of 2 types:

- 1) clustered
- 2) Non - clustered

## Procedure / Stored Procedure

\* These are pre-compiled programs.

(Already compiled and which will be stored in executable form).

Notes (check in notes)

Cursor :- Cursor means memory (Internal Storage)

(while executing these things should be stored in memory for being used somewhere i.e., Cursor).

What is the need if it was not there?

Ans:- To access required data.

cursor - pointers (

variables - only (

subscript bracket } subscript

answering following query also result at all hands of this question from biologist friend

(most interesting)

(action of user) (partly)

## Questions on Special Operators

- 1) List all the employees whose commission is null.

A: From emp select ename

From emp

where comm is Null;

- 2) List All the employees who don't have a reporting manager.

A: Select ename

From emp

where MGR is Null;

- 3) List all the salesman in dept 30.

A: Select ename

From emp

where Job in 'salesman' and deptno = 30;

- 4) List all the salesman in dept number 30 and having salary greater than 1500.

A: Select ename

From emp

where Job in 'salesman' and deptno in 30 and

salary > 1500;

5) List all the employees whose name starts with 'S' OR 'A'. Ans: Pavan

A:- select ename  
from emp  
where ename like 'A%'  
or ename like 'S%'

6) List all the employees except those who are working in dept 10 & 20.

A:- select ename  
from emp  
where deptno not in (10, 20);

7) List all the employees whose name does not start with 'S'.

A:- select ename  
from emp  
where ename not like 'S%';

8) List all the employees who are having reporting managers in dept 10.

A:- select ename  
from emp  
where MGR is not null and

i) list all the employees whose commission is null and working as clerk.

select ename

From emp

where comm is null and job in 'CLERK';

ii) list all the employees who don't have a reporting manager in deptno 10 or 30.

select ename

From emp

where mgr is null and deptno in (10,30);

iii) list all the salesman in dept 30 with sal more than 2450.

select ename

From emp

where job in 'salesman' and deptno in 30 and  
sal > 2450.

iv) list all the analyst in dept no 20 and having salary greater than 2500.

Select ename

From emp

where job in 'Analyst' and deptno in 20 and  
sal > 2500.

(13) List all the employees whose name starts with 'M' or 'J'

A) Select ename

From emp

where ename like 'M%' or 'J%'

(14) List all the employees with annual salary except those who are working in deptno 30.

A) Select ename, Sal\*12 "Annual salary"

From emp

where deptno not in 30

(15) List the employees whose name does not end with 'ES' or 'R'.

A) Select ename

From emp

where ename not like '%.ES' and ename not like '%.R'

(16) List all the employees who are having reporting managers in dept 10 along with 10% hike in salary.

A) Select ename

From emp

where

⑯ Display all the employee who are 'Salesman's having 'E' as the last but one character in ename, but salary having exactly 4 character.

Ans Select ename

From emp  
where job in 'salesman' and ename like '%.E-'  
and sal like '----'

⑰ Display all the employee who are joined after year 81.

Ans Select ename  
From emp  
where Hiredate > '31-DEC-81';

⑲ Display all the employees who are joined in FEB.

Ans Select ename  
From emp  
where Hiredate like '%.FEB.1.';

⑳ List the employees who are not working as managers and clerks in dept 10 and 20 with a salary in the range of 1000 to 3000.

Ans Select ename  
From emp  
where job not in 'Manager' and not job not in 'clerk'

and deptno not in 10 and deptno 20 and sal  
between 1000 to 3000;  
where job in ('manager', 'clerk') and deptno  
(10,20) and sal between 1000 and 3000;

- ② List the employees whose salary not in the  
range of 1000 and 2000 and working in dept  
10, 20 or 30 except all salesman.

A:-  
Select ename  
From emp  
where sal not between 1000 and 2000 and  
deptno in (10, 20, 30) and job not  
~~in ('Salesman')~~

- ③ List the department names which are having  
letter 'o' in their locations as well as their  
department names.

A:-  
Select DName  
From ~~emp~~ Dept  
where loc like '%.0%' and DName like '%.0%'

- ④ Display all the employees whose job has  
string 'MAN' in it.

A:-  
Select ename  
From emp

where job like '%MAN%'

- 24) List the employees who are hired after 82 and before 87.

A:- select ename

From emp

where hiredate between >'31-DEC-82' and hiredate  
<'31-JAN-87';

- 25) WAPTD all the details of employees hired in November and December.

A:- select \*

From emp

where hiredate like '%.Nov.' or hiredate like '%.Dec.';

- 26) List all the employee names and commission for those employees who earn commission more than their salary.

A:- select ename, comm

From emp

where comm > sal;

- 27) WAPTD Name and designation for all the employees having reporting managers and also their names starting with 'S'.

A:- select ename, job "Designation"

From emp

where MGR is not null and ename like 'S.%'

- (28) WAPTD Name and salary of all the employee if their Annual salary ends with '0'.

A:  
= Select ename, sal  
From emp  
Where sal\*12 like '%.0';

- (29) WAPTD Name of the employee having atleast 2's in his name.

A:  
= Select ename  
From emp  
Where ename like '%.A%.A%';

- (30) WAPTD Name of the employees whose name Starts with A 'vowel'.

A:  
= Select ename  
From emp  
Where ename like 'A%' or Ename like 'E%' or  
Ename like 'I%' or Ename like 'O%', or Ename like 'U%'