

SQL SYLLABUS:

- Introduction (Data, Database, DBMS, RDBMS)
- Relational Model.
- E.F. Codd Rules.
- Datatypes.
- Constraints.
- Statements.
 - DDL
 - DML
 - TCL
 - DCL
 - DQL
- Operators
- Functions
- Group by
- Having
- Order by
- Sub-query
- Joins
- Correlated sub-query
- Pseudo Columns
- Normalisation
- Basics of PL/SQL

Data :-

Data is a noun which describes attributes of an object or entity.

Attributes :-

Properties of an entity or object.

Entity :-

Anything which carries its existence.

Database :-

Database is a place or medium where we can store the data in systematic organised manner.

We can perform our basic operations in database, they are,

Create / Insert - C

Read / Retrieve - R

Update / Modify - U

Delete / Drop - D

This operation is known as "CRUD operation".

DBMS :-

DBMS stands for Database management system.

DBMS is a software which is used to maintain and manage the database.

DBMS provides two important features, They are,

- Security

- Authorisation

To communicate with DBMS software we need a language is called Query language.

Types of DBMS:-

There are 4 types of DBMS,

i. Network

ii. Object Oriented

iii. Hierarchy

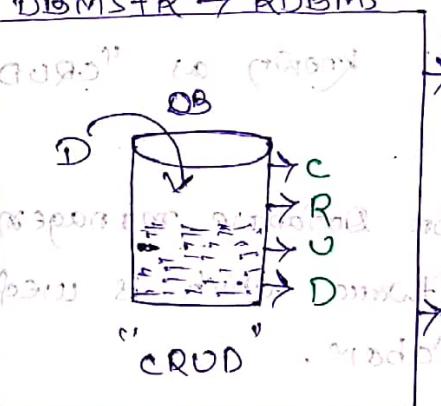
iv. RDBMS

RDBMS :-

RDBMS stands for Relational Database System.

RDBMS is a type of DBMS software in which we can store the data in the form of Table or Relations.

- If DBMS follows relational model then it becomes RDBMS.
- If DBMS follows E.R. CODD rules then it becomes RDBMS.



Structured Query language

(SQL)

Microsoft

DBMS
1. Network

2. Object Oriented.

3. Hierarchy

4. RDBMS

- To communicate with RDBMS, we need Structured Query Language.

Relational Model :-

- Relational Model was designed by E.F. Codd.
- In relational model on RDBMS, we can store the data in the table form.
- In relational model we can also store meta data in the form of tables or relational forms.

Q. What is Meta data?

- Meta data means data about the data.

Table:-

Table is a logical organisation of data which consists of rows & columns.

Columns:-

Attributes of all the entity is column, name also called is as attributes or fields.

Row Data:-

All the attributes of single entity, also we can called as Records or Tuples.

Cell:-

Smallest unit in the table where we can store the data.

→ When the implementation up rows and columns,
cell will get generated.
Students (Attributes/column.name/fields)

ID	Name	Branch	Percentage	No. of Book
103	A	CS	68.45	0
108	B	EC	62.00	0
100	C	CV	71.33	0
55	D	ME	55	6
104	E	BSC	55	3

Link → tinyurl.com/yc4P6g5n

Jobs for PostgreSQL in pgpool to 21 select

Normal selection to database digit

Normal query to database digit

High level of student to 21 better odo

Normal query to database digit

High level of student to 21 better odo

Normal query to database digit at time function

After login user

→ Differentiate between DB and DBMS:

Database:

- Database is a place where we can store the data in systematic organised manner.
- Database prefers four operation which is known as CRUD operation.
- The database is any collection of data whether you are writing it on the paper or storing it in the digital format.
- We don't need any language to communicate with Database.
- To access the data from database, user need database management system.

DBMS:

→ Database management system is a software which is used to manage and maintain the database.

→ DBMS provides two important features,

- i. Security
- ii. Authorisation

→ DBMS is a kind of software that helps you to read, edit and store structured data in the database.

→ Designed a language to communicate with DBMS which is Query language.

→ There are four types of DBMS, those are,

- i. Network
- ii. Object Oriented
- iii. Hierarchy
- iv. RDBMS.

→ Difference between DBMS and RDBMS :-

DBMS :-

→ DBMS is a software which is used to manage and maintain the database.

→ DBMS provides two important features,

those are

i - Security

ii - Authorisation

→ DBMS applications store data as file.

→ DBMS has to provide some uniform methods to access the stored information.

→ DBMS doesn't support distributed database.

→ DBMS is meant to be for small organization and deal with small data. It supports single user.

RDBMS :-

→ RDBMS is a type of DBMS software in which we can store the data in the form of table or relations.

→ RDBMS follows E.F. CODD rules.

→ RDBMS applications store data in a tabular form.

→ RDBMS system supports tabular structure of the data and a relationship between them to access the stored information.

→ RDBMS supports distributed database.

→ RDBMS designed to handle large amount of data. It supports multiple users.

→ Difference between RDBMS and ExcelSheet:-

RDBMS has both ExcelSheet, what are -

→ RDBMS is a Relational Database management system. → Excel is a spreadsheet application.

→ To communicate with RDBMS, we need Structured query language (SQL). → In ExcelSheet, we don't need any language to communicate.

→ RDBMS follows E.F. Codd Rules. → ExcelSheet doesn't follow E.F. Codd Rules.

Codd Rules, and performance can be affected.

→ We can store unlimited data in RDBMS. → We can ~~store~~ limited data in ExcelSheet.

→ We can also store meta data in RDBMS. → We can not store meta data in ExcelSheet.

→ RDBMS has capabilities only in relations or in the form of table. → Excel has databaseing capabilities, but doesn't have the relational database.

→ RDBMS is a fast speed as compared to Excel.

→ RDBMS is

E.F. CODD Rule:

- The data entered to the cell should be single value before atomic data.
- We can store the data in multiple tables and also established a connection between any two tables by using key attributes.
- We can validate the data by assigning datatypes and constraints.
- Datatypes are mandatory but constraints are optional.

Datatype is a type of Data, which is used to stored in memory location in Database.

Types of Datatype in SQL

1. char datatype.
2. Varchar datatype.
3. Large object datatype.
 - i. Char LOB
 - ii. Binary LOB.
4. Date.
5. Number.

1. Char datatype:-

To store the characters such as it has to be a single code.

Digit pi
Special char
'0-9'

Syntax :- char(size);

Ex:- `char(10);`

The diagram illustrates a memory request. At the top, the word "Request" is written above a bracket that spans across two vertical columns. The left column contains the text "char(10);". The right column contains the word "Memory" above a large rectangular box representing memory storage. A horizontal line connects the bottom of the "Request" bracket to the top edge of the memory box. Below the memory box, a smaller rectangle is shown with a line pointing from its top edge to the bottom edge of the main memory box, indicating a sub-allocation or a specific memory location.

(Fixed length memory allocation)

0	i	l	s	h	a				
---	---	---	---	---	---	--	--	--	--

Used Memory Unused Memory

⇒ char. lactose⁻ type is also known as

Fixed length memory allocation

... indicate the possibility that the Sandman

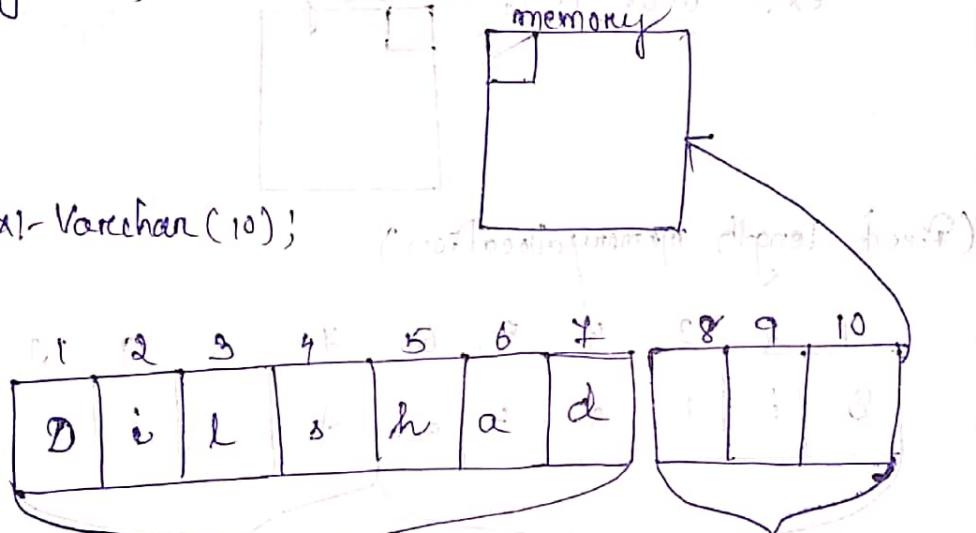
Figure 103. A sketchy bird collection made at

2. Varchar:-

It is used to store characters such as if it has to be in single, double, or

- Varchar separates the unused memory and again refers ~~that~~ ^{to} memory in the memory location.

Syntax :- Varchar(size);



(Variable length memory allocation)

Varchar 2 :- also stores upto 2000 characters into 2 bytes

Syntax :- Varchar 2 (size) upto 2000

Varchar 2 is only updation of Varchar.

Note:-

char and varchar can accept maximum 2000 characters but Varchar 2 can accept maximum 4000.

3. Large Object Datatypes:-

i. Char LOB:-

It is used to store the characters up to 4GB. Syntax:- CLOB;

ii. Binary LOB:-

To store binary values of Audios, Videos, files etc. upto 4GB.

Syntax:- BLOB;

4. Date:-

This standard Date format which is given by Oracle.

Ex:- DD-MON-YY → 08-AUG-19

DD-MON-YYYY → 08-OCT-1996

5. Number:-

Number (Precision[, Scale]);

- It is used to store number value.

Precision:-

It is used to store integer values and the range starts from -1 to 38.

Scale:-

It is not mandatory and it is used to store decimal values in the precision.

Its range from -127 to 127.

Case-1 :-

Number(4);

$$\pm \boxed{9999}$$

Number(6);

$$\pm 999999$$

Number(5, 3); \rightarrow split into 2 parts

$$\pm 99999$$

Case-2 :-

Number(6, 2);

$$\pm 999.99$$

Number(6, 1);

$$\pm 999999.999999999$$

Number(8, 4);

$$\pm 9999.9999$$

Number(4, 2, 2, 1); \rightarrow (4, 3)

$$\pm 9.999$$

Case-3 :- if, $S > P \Rightarrow S-P$ Number(3, 3); $3+3=6$

$$\pm 0.999$$

Number(4, 8); $8-4=4$
 ± 0.00009999 Number(4, 5); $5-4=1$

$$\pm 0.09999$$

Number($\frac{1}{800}$); $9-1=8$

$$\pm 0.00000009$$

Number(4, 0) $\rightarrow X$

Constraints:-

constraints ~~are~~ the rules or conditions which are given to the column to validate the data.

Types of constraints:-

- i. Unique Constraints.
- ii. Not Null constraints.
- iii. Check constraints.
- iv. Primary key constraints.
- v. Foreign key constraints.

Dt. 09/08/19

1. Unique Constraints:-

→ To avoid duplicate values which are entered in to the table.

Null :-

- null is a keyword, which is used to represent empty cell or nothing.
- null doesn't occupy memory.
- Any operation performed with null, the result it will be null.
- In DBMS two nulls can't be same.

ii) Not Null:-

It represents the cell should not be null
and information of records will not having some
one empty.

iii) check:-

It is used to provide extra condition to
the column to validate the data.

Syntax :- `check(condition);`

`Ex:- check(length of mobile no. is equal
to 10);`

iv) Primary key:-

It is used to uniquely identify a record
from the table and should not be null.
Characteristics of primary key constraints:

- i. The table can has only one primary key.
- ii. Primary key should be unique.
- iii. It should be Not null.
- iv. It is always a combination of unique
and not null.
- v. It is not mandatory but designed by
preferable.

V. Foreign key constraints :-

It is used to establish a connection between any two tables.

Characteristics of Foreign key:-

- i. The table can have more than one foreign key.
- ii. Foreign key should be unique.
- iii. It can be null or not null.
- iv. It is not a combination of unique and not null.
- v. It is not mandatory.

- * A. Primary key of a table can only become foreign key in another table.
- * A. Key should be a primary key in its own table then it can be a foreign key in another table.
- Foreign key is always present in child table but actually belongs to parent table.
- Foreign key is also known as Referential integrity constraints.

1	1. D. name : Dev.	2. Loc : B.N.G.L.R
2	3. Dev. : 10	4. Noida
3	5. D.O : 15	6. Pune
4	7. HR : 20	8. Jaipur
5	9. Sales : 25	10. Bopal.

parent table

Check (SQL) 1/50		(Not Null & Unique)		Primary Key		Primary Key		Primary Key		Primary Key		Primary Key		Primary Key		Primary Key		Primary Key	
		N.N.	N.N.	N.N.	N.N.	N.N.	N.N.	N.N.	N.N.	N.N.	N.N.	N.N.	N.N.	N.N.	N.N.	N.N.	N.N.	N.N.	
10	Roshni	DOB	2000	Gender	Female	SSN	9999999999	Name	Shreya	SSN	1000000000	Gender	Female	SSN	1000000000	Gender	Female	SSN	
11	Niraj	DOB	1990	Name	Niraj	SSN	1111111111	Address	123 Main St	SSN	1111111111	Gender	Male	SSN	1111111111	Gender	Male	SSN	
12	Marta	SSN	150000	Name	Marta	SSN	150000	Address	456 Elm St	SSN	150000	Gender	Female	SSN	150000	Gender	Female	SSN	
13	Shreya	DOB	2000	Name	Shreya	SSN	1313131313	Address	567 Oak St	SSN	1313131313	Gender	Female	SSN	1313131313	Gender	Female	SSN	
14	Aditi	SSN	150001	Name	Aditi	SSN	1414141414	Address	678 Pine St	SSN	1414141414	Gender	Female	SSN	1414141414	Gender	Female	SSN	
15	Deeksha	SSN	100000	Name	Deeksha	SSN	1515151515	Address	789 Birch St	SSN	1515151515	Gender	Female	SSN	1515151515	Gender	Female	SSN	
16	84	SSN	500	Name	Gadola	SSN	1616161616	Address	123 Main St	SSN	1616161616	Gender	Male	SSN	1616161616	Gender	Male	SSN	
17	58	SSN	1000	Name	Deeksha	SSN	1717171717	Address	456 Elm St	SSN	1717171717	Gender	Female	SSN	1717171717	Gender	Female	SSN	
18	94	SSN	5000	Name	Shreya	SSN	1818181818	Address	567 Oak St	SSN	1818181818	Gender	Female	SSN	1818181818	Gender	Female	SSN	
19	10	SSN	150000	Name	Marta	SSN	1919191919	Address	678 Pine St	SSN	1919191919	Gender	Female	SSN	1919191919	Gender	Female	SSN	
20	11	SSN	100000	Name	Niraj	SSN	2020202020	Address	789 Birch St	SSN	2020202020	Gender	Male	SSN	2020202020	Gender	Male	SSN	
21	12	SSN	150001	Name	Deeksha	SSN	2121212121	Address	123 Main St	SSN	2121212121	Gender	Female	SSN	2121212121	Gender	Female	SSN	
22	13	SSN	100000	Name	Aditi	SSN	2222222222	Address	456 Elm St	SSN	2222222222	Gender	Female	SSN	2222222222	Gender	Female	SSN	
23	14	SSN	5000	Name	Gadola	SSN	2323232323	Address	567 Oak St	SSN	2323232323	Gender	Male	SSN	2323232323	Gender	Male	SSN	
24	15	SSN	1000	Name	Shreya	SSN	2424242424	Address	678 Pine St	SSN	2424242424	Gender	Female	SSN	2424242424	Gender	Female	SSN	
25	16	SSN	500	Name	Marta	SSN	2525252525	Address	789 Birch St	SSN	2525252525	Gender	Female	SSN	2525252525	Gender	Female	SSN	
26	17	SSN	100000	Name	Niraj	SSN	2626262626	Address	123 Main St	SSN	2626262626	Gender	Male	SSN	2626262626	Gender	Male	SSN	
27	18	SSN	150001	Name	Deeksha	SSN	2727272727	Address	456 Elm St	SSN	2727272727	Gender	Female	SSN	2727272727	Gender	Female	SSN	
28	19	SSN	100000	Name	Aditi	SSN	2828282828	Address	567 Oak St	SSN	2828282828	Gender	Female	SSN	2828282828	Gender	Female	SSN	
29	20	SSN	5000	Name	Gadola	SSN	2929292929	Address	678 Pine St	SSN	2929292929	Gender	Male	SSN	2929292929	Gender	Male	SSN	
30	21	SSN	100000	Name	Shreya	SSN	3030303030	Address	789 Birch St	SSN	3030303030	Gender	Female	SSN	3030303030	Gender	Female	SSN	
31	22	SSN	150001	Name	Marta	SSN	3131313131	Address	123 Main St	SSN	3131313131	Gender	Female	SSN	3131313131	Gender	Female	SSN	
32	23	SSN	100000	Name	Niraj	SSN	3232323232	Address	456 Elm St	SSN	3232323232	Gender	Male	SSN	3232323232	Gender	Male	SSN	
33	24	SSN	5000	Name	Deeksha	SSN	3333333333	Address	567 Oak St	SSN	3333333333	Gender	Female	SSN	3333333333	Gender	Female	SSN	
34	25	SSN	1000	Name	Aditi	SSN	3434343434	Address	678 Pine St	SSN	3434343434	Gender	Female	SSN	3434343434	Gender	Female	SSN	
35	26	SSN	500	Name	Gadola	SSN	3535353535	Address	789 Birch St	SSN	3535353535	Gender	Male	SSN	3535353535	Gender	Male	SSN	
36	27	SSN	100000	Name	Shreya	SSN	3636363636	Address	123 Main St	SSN	3636363636	Gender	Female	SSN	3636363636	Gender	Female	SSN	
37	28	SSN	150001	Name	Marta	SSN	3737373737	Address	456 Elm St	SSN	3737373737	Gender	Female	SSN	3737373737	Gender	Female	SSN	
38	29	SSN	100000	Name	Niraj	SSN	3838383838	Address	567 Oak St	SSN	3838383838	Gender	Male	SSN	3838383838	Gender	Male	SSN	
39	30	SSN	5000	Name	Deeksha	SSN	3939393939	Address	678 Pine St	SSN	3939393939	Gender	Female	SSN	3939393939	Gender	Female	SSN	
40	31	SSN	1000	Name	Aditi	SSN	4040404040	Address	789 Birch St	SSN	4040404040	Gender	Female	SSN	4040404040	Gender	Female	SSN	
41	32	SSN	500	Name	Gadola	SSN	4141414141	Address	123 Main St	SSN	4141414141	Gender	Male	SSN	4141414141	Gender	Male	SSN	
42	33	SSN	100000	Name	Shreya	SSN	4242424242	Address	456 Elm St	SSN	4242424242	Gender	Female	SSN	4242424242	Gender	Female	SSN	
43	34	SSN	150001	Name	Marta	SSN	4343434343	Address	567 Oak St	SSN	4343434343	Gender	Female	SSN	4343434343	Gender	Female	SSN	
44	35	SSN	100000	Name	Niraj	SSN	4444444444	Address	678 Pine St	SSN	4444444444	Gender	Male	SSN	4444444444	Gender	Male	SSN	
45	36	SSN	5000	Name	Deeksha	SSN	4545454545	Address	789 Birch St	SSN	4545454545	Gender	Female	SSN	4545454545	Gender	Female	SSN	
46	37	SSN	1000	Name	Aditi	SSN	4646464646	Address	123 Main St	SSN	4646464646	Gender	Female	SSN	4646464646	Gender	Female	SSN	
47	38	SSN	500	Name	Gadola	SSN	4747474747	Address	456 Elm St	SSN	4747474747	Gender	Male	SSN	4747474747	Gender	Male	SSN	
48	39	SSN	100000	Name	Shreya	SSN	4848484848	Address	567 Oak St	SSN	4848484848	Gender	Female	SSN	4848484848	Gender	Female	SSN	
49	40	SSN	150001	Name	Marta	SSN	4949494949	Address	678 Pine St	SSN	4949494949	Gender	Female	SSN	4949494949	Gender	Female	SSN	
50	41	SSN	100000	Name	Niraj	SSN	5050505050	Address	789 Birch St	SSN	5050505050	Gender	Male	SSN	5050505050	Gender	Male	SSN	

child + cable.

- SQL Statement or language:

1. Data Definition Language (DDL)

- ↳ CREATE
- ↳ RENAME
- ↳ ALTER
- ↳ TRUNCATE

↳ DROP

2. Data manipulation language (DML)

- ↳ INSERT
- ↳ UPDATE
- ↳ DELETE

3. Transaction Control language (TCL)

- ↳ COMMIT
- ↳ SAVE POINT

↳ ROLLBACK

4. Data Control language (DCL)

- ↳ GRANT
- ↳ REVOKE

5. Data query language (DQL)

- ↳ SELECT

- ↳ PROJECTION

↳ SELECTION

- ↳ Aggregation

↳ Join

↳ Union

DQL:- (Data Query Language)

Select-

This statement is used to retrieve the data from the table which is present in Database.

→ Projection :-

This statement is used to retrieve the data from the table by selection column name.

By default in projection all the records will get selected.

→ Selection :-

This statement is used to retrieve the data from the table by selection both column and row name.

→ Join :-

Retrieval of the data from multiple table simultaneously known as join.

→ Projection :-

Syntax- `SELECT * / [DISTINCT] Column-name / Expression [ALIAS]`

`From table-name;`

order of Execution :-

1. From clause.
2. Select clause.

① FROM Clause :-

- Always From Clause executes first.
- From FROM clause, we pass table name or, an argument.

function of FROM clause :-

It goes to the database. Search for the table name, put it under execution.

② SELECT CLAUSE :-

- Select clause will execute after the execution of FROM clause.

- For SELECT CLAUSE we can pass column / expression as an argument.

function of SELECT clause :-

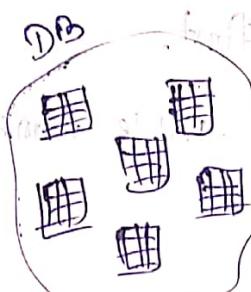
- It goes to the table, which is under execution from the table, get the data and put it on the display.

- Select clause is responsible for displaying the output.

Q1. Write a query to display all the employee names from employee table.

SELECT Name

FROM emp;



1. From clause goes to DB

To DB

For the table

2. Select Name

3. From emp

O/P of the query will be displayed in the following format:

EMPLOYEE TABLE

ID	Name	Sal	D.no.
1	Sundar	650	20
2	Macha	500	10
3	Sundar	450	20
4	Macha	700	30

Output Table

Name
Sundar
Macha

Order Execution

Select the

data and

Put it on display

Q.2 :- Write a Query to display salary from Emp table along with name of the employee.

~~SELECT Sal, Name from Emp;~~

Q.3 :- Write a Query to display all Dno. from Emp table.

~~SELECT Dno;~~

~~SELECT D no,~~

~~from Emp;~~

Q.4 :- Display all the Emp ID and Name

~~along with Salary.~~

~~SELECT ID, Name, Sal~~

~~From Emp;~~

Q.5 :- Display Name, Dept no. along with ID.

~~SELECT Name, Dno, ID~~

~~From Emp;~~

Q.6 :- Display all the Details from emp table.

~~Select *~~

~~From Emp;~~

~~Chap 2 (work # 10102)~~

* / Wild character / Asterisk :-

It is used to select all the details from the given table.

Q.7:- Write a query to display all dept. name from dept. ~~select dname~~.

Select dname

from Dept;

Q.8:- Write a query to display Dept. No. from Dept. to the table.

Select Dept. No.

from Dept;

Q.9:- Display Dname with *.

Select Dname,

From Dept;

Q.10:- In Q7 D * Dept. No. and location.

Select Dept. No, location

From Dept;

Q.10:- In Q7 D all the details from Department.

Select * from Dept;

Q.11 - WAQTD all table present in Database

Select *
From tab;

tab :- It is a table name where in all the table names ^{are} present. Write all present in database.

Note :- To set page properties :-

→ Set lines 100 & pages 100;

Q.12 - WAQTD all employee name from emp table.

Select Ename
From Emp;

Q.13 - WAQTD all the employee ID along with

then designation
Select emp no, job
From emp;

Q.14 - WAQTD all the comm along with
Dept. No.

Select comm, dept. no.
From emp;

Q.15 - WAQTD employees name and manager no. along with date of joining.

Select ename, mngt, hire date
From emp;

Q.15:- Display all the job with hiredate

Select Job, hiredate from

from Emp;

Q.16:- Display Ename along with Sal

Select Ename, sal

from Emp;

Q.17:- WAPTD E.no, Ename, along with Annual
Sal.

Select empno, ename, sal*12

from emp;

Expression:-

Anything which gives result or output is known as expression.

Expression consist of 'operator' and 'operand'.

Operands are two types,

① Direct Values

② Immediate Values/ column-name /

Literals (char, num, date)

e.g:- 10 + 10 ;

sal * 12 ;

a+b;

Q. WAPTD Employee name along with 6 month salary.

Select ename, sal*(6) ~~from emp~~
from emp;

Q. WAPTD employee name, monthly salary and also Display 10% hike to the monthly sal.

Select ename, sal, sal + sal * 0.10
from emp;

Q. WAPTD Job, monthly commission along with 12% deduction in monthly sal.

Select job, comm, sal - (sal * 0.12)
from emp;

Q. WAPTD Sal., 25% hike to the Annual salary and also 8% deduction in commission.
Select sal, (sal * 12) + (sal * 12) * 25/100, comm - (comm) * 8/100
from Emp;

Q. WAPTD emp's name along with monthly sal, 13% hike in 6 month sal, 18% deduction in annual comm.

Select ename, sal, (sal * 6) + 1.13,
comm * 12) + 0.82
from emp;

Q. WAP TO Display All employees name along with their
Dept No.

Select ename, Dept no.
from emp;

Q. Display all employee details.

Select *
from emp;

Q. WAP TO Display Emp name, along with their
monthly Sal and also 23% deduction

(Show Annual commission due)

Select ename, sal, (comm - comm * 23)
from emp;

Select ename, sal, comm * 23 - (comm * 23) *
23 / 100

from emp;

Q. WAP TO Display All employees name along with their
Dept No.

Salary = 23.0 * (27.0 * 1000)

from emp;

ALIAS:-

Alternative name given to the column(s) or expression, which are present in the result table.

Condition to write ALIAS:-
(i) With or without Using the keyword AS;

We can write ALIAS name.
(ii) If we are using multiple words as ALIAS name then, it has to be joined by using '-' (UnderScore) or enclosed within " " (double quote)

Ex:-

Select Ename as name, Sal as Salary,
Sal * 12 "Annual salary", Sal * 6 "half-term"

from Emp;

Q:- WAPTD Ename, Job, Annual ~~com~~, Sal,
Half term Sal, 8% Hike in Annual Sal.

Select Ename, Job, Sal * 12 "Annual Sal",
Sal * 6 "Half term", ~~8% Hike~~,
Sal * 12 + (Sal * 12 * $\frac{8}{100}$) "Hike"

from Emp;

DISTINCT:

To avoid duplicate values which are present in Result Table.

- (i) In SELECT clause either * or Distinct should be the first argument.
 - (ii) For Distinct Clause we can pass multiple Arguments,
 - (iii) Distinct clause ~~if~~ removes combination of columns which are duplicated.

Ex1-

Select Dept no.

Dept. No.

Now,
after
Sale

Select DISTINCT DeptNo.

From Emp;

Aug 19, 81 10:10

327

20

10

17.000.000

L. S. 10870

四百三

152

卷之三

Select SAL, Dept. NO.
From EMP;

SAL	Dept. NO.
200	20
400	10
200	30
500	20
400	10

Select DISTINCT SAL,
Dept. NO.
From EMP;

SAL	Dept. NO.
200	20
400	30
500	10
200	20

Q. What all the details of Emps from Emp table?

Select *

From EMP;

Q. What Ename, SAL along with Annual SAL.

Select Ename, SAL, SAL * 12 "Annual SAL"

From EMP;

Q. What Name, monthly SAL and 200 RS incentive to the SAL?

Select Ename, SAL, SAL + 200 "Incentive"

From EMP;

Q. What name, designation, monthly salary and 500 RS incentive to the Annual salary.

Select Ename, Job, SAL, SAL * 12 + 500 "Incentive"

From EMP;

Q1 IWAQTD

Ans: Display basic info to & G. for table 2

Q1.5

Select #, Sal #12 } Wrong
From emp;

Output:

Select Emp.#, Sal#12

From Emp;

Q2 IWAQTD all the details of employee along with commission.

Select Emp.#, comm

From Emp;

Q3 IWAQTD all the sal along with name

Select Sal, Ename

From Emp;

Q4 IWAQTD all the employee names from emp

Tables.

Select Ename, # and name 3 basic

From emp;

All the basic information about employee is displayed

Q5. Display all the employee details in the form of (structured) GROUP BY #, job, salary, basic

Impression

SELECTION :-

To retrieve the data from the table by using 'column name' and 'row's name' is known as SELECTION.

Syntax :- (with order of execution)

<3> SELECT * / [DISTINCT] column-name / Expression [ALIAS]

(1) From table-name

(2) WHERE <Filter-condition>;



WHERE Clause :-

Where clause is used to filter the records.

(i) After the execution of From clause, where clause will execute.

(ii) Where clause executes row by row.

(iii) In Where clause, we can not use ALIAS name.

(iv) We can write multiple conditions in WHERE clause.

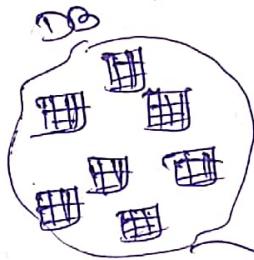
DOB	DOB	Address	2
DOB	DOB	Address	2
DOB	DOB	Address	2

Q WAP TO ENAME, SAL if the ename is Sundri ?

SELECT ENAME, SAL
FROM EMP
WHERE ENAME = 'Sundri';

order of execution :-
① FROM clause

② WHERE clause



EMP NO.	ENAME	SAL	DNO.
1.	Sundri	200	30
2.	Machha	500	10
3.	Sundri	700	20
4.	Machhi	300	30
5.	Sundri	400	10

Put under execution

② WHERE 'Ename' = 'Sundri' (Goes row by row)

O/P
WHERE

EmpNo.	ENAME	SAL	DNO.
3	Sundri	700	700
5	Sundri	400	400

3. SELECT Ename, Sal (Display Output)

O/P
SELECT :-

doC person 7/2/32
Emp (contd)

Ename	Sal.
Sundari	700
Sundari	400

Q:- WAPTD Name, comm Given to all the Emp if they are getting comm lesser than or Equal to 500 ?

Select Ename, comm

from Emp

where comm <= 500 ;

Q:- WAPTD Details of Emps if their Emp ID is 7788.

Select *

from Emp

where Emp no. = 7788 ;

Q:- WAPTD all the details of Emps along with half year comm if they are not working as 'Manager'.

Select Emp.* , Sal * G

From Emp

where Job != 'Manager' ;

Q WAP TO Find Name, Job of all the Emps?

Select Ename, Job
From Emp;

Q WAP TO name, sal along with annual sal
if they are getting annual sal more
than 1Mk.

Select Ename, Sal, Sal*12 "Annual Sal"
From Emp
Where Sal*12 > 14000;

Q WAP TO details of Emps if they are hired
after 81.

Select *
From Emp
Where Hiredate >= '01-JAN-82'
> '31-DEC-81';

Q WAP TO Details of Emps if they are working
as president in dept no.10.

Select *
From Emp
Where Job = 'president' AND DEPT NO.10
(operator)

String, string, string

and and

{ operator } do something

Operators:-

1. ARITHMETIC OPERATORS (+, -, *, /)
2. COMPARISON OPERATORS (=, !=, OR <>)
3. RELATION OPERATORS (>, <, >=, <=)
4. LOGICAL OPERATORS (AND, OR, NOT)
5. CONCATINATION OPERATORS (||)
6. SPECIAL OPERATOR (IN, NOT IN, BETWEEN, NOT BETWEEN, IS, IS NOT, LIKE, NOT LIKE)
7. SUB - QUERY OPERATORS (ALL, ANY, EXISTS, NOT EXISTS)

1 * 1	1 3 True
1 # 0	0
0 * 0	0
0 # 0	0

I/P	O/P
1 + 1	1
1 + 0	1
0 + 1	1
0 + 0	0

} True.

NOTE: 'AND' operator returns true when both the conditions are true.

'OR' operator returns true when any of the conditions are true.

Q: WAQTD details of Employees if they are working in DEPT NO. 20 as a clerk?

Select *
From Emp
Where Dept No. 20 AND Job = 'CLERK';

Q: WAQTD details of Employees if they are working as MANAGER OR ANALYST?

Select *

From Emp
Where Job = 'MANAGER' OR Job = 'ANALYST'.

Q: WAQTD (the Name, DEPT NO., Job along with COMM if employees are getting comm more than 0 and they are working as SALESMAN).

Select Enampl, Dept No., Comm, Job

From Emp

Where Comm > 0 AND Job = 'SALESMAN';

Q: WAQTD SALARY given all the employees if they are getting salary more than 1250 but less than 3200.

Select Sal

From Emp
Where Sal > 1250 AND Sal < 3200;

Q:- WAQTD details of Emps if they were hired in the month of DEC 81.

Select *

From Emp

Where Hiredate => '01-DEC-81' AND
Hiredate <= '31-DEC-81';

Q:- WAQTD Name, Hiredate of all the Manager or president if they are working in Dept No. 20.

Select Ename, Hiredate

From Emp

Where Job = 'MANAGER' OR Job = 'PRESIDENT'
AND DEPT. NO. 20;

Q:- WAQTD hiredate of Emps who were hired during the year 87.

Select Hiredate

From Emp

Where Hiredate => '01-JAN-87' AND
Hiredate <= '31-DEC-87';

Q:- WAQTD Name, Job, Dept No., comm of the clerk or analyst if they are getting SAL lesser than or equal to 3k.

Select Ename, Job, DeptNo, comm

From Emp

Where Job = 'CLERK' OR Job = 'ANALYST'
AND SAL <= 3000;

Q:- WAPTD all the details of Emps along with annual SAL if annual SAL more than 12k but lesser then 25k.

Select Emp. # & Sel # 12 "Annual Sal"

From Emp # > Subquery

Inhere Sel # 12 > 12000 AND Sel # 12 < 25000;

Q:- WAPTD details of Employees who are working as 'SALESMAN' OR 'MANAGER', OR 'CLERK' OR 'ANALYST' OR 'PRESIDENT'.

Select # & Job# From

From Emp

Inhere # Job = 'SALESMAN' OR Job = 'MANAGER' OR Job = 'ANALYST'
OR Job = 'PRESIDENT' ;

From Emp # & Job#

From Emp # & Job#

Ans :- Select # & Job# From Emp

From Emp # & Job#

→ IN OPERATOR !-

LHS RHS
column name/Exp $IN(v_1, v_2, v_3, \dots, v_n);$

IN is a multivalued operator which can accept multiple values at RHS but single value at LHS.

Ex If $Sal = 400$

$Sal IN (100, 200, 300, 400);$

$400 = 100 \times$

$400 = 200 \times$

$400 = 300 \times$

} True

∴ True

∴ True

Note :- $(100, 200, 300, 400)$ is true if any one of IN operations returns true if any one of the condition is true.

Q:- WAP TO name & sal of all employee who are getting sal as 1250 or 3000 or 5000.

Select Ename, Sal

From Emp

Where Sal IN (1250, 3000, 5000);

Q1: WAQTD Employee No. from a job of all the employees who are working in Dept (no. 20 or 30) as a manager or clerk.

Select Emp no., Job
From Emp
Where Dept No. IN(20,30) AND Job IN
(MANAGER, CLERK);

Q2: WAQTD details of employees if they are not working as SALESMAN or PRESIDENT IN DEPT. NO. 30,20.

Select *
From Emp
Where Job NOTIN('SALESMAN', 'PRESIDENT')
AND Dept No IN(30,20);

~~NOTIN~~

1st, 2nd, 3rd, 4th

1st, 2nd, 3rd, 4th
1st, 2nd, 3rd, 4th

→ NOT IN:-

LHS

RHS

col/Name / exp NOT IN ($v_1, v_2, v_3, \dots, v_n$);

Ex! (400)

SAL NOT IN (1000, 2000, 3000, 4000);
2000 OR 3000 OR 4000

$400 \neq 1000 \checkmark$
 $400 \neq 2000 \checkmark$
 $400 \neq 3000 \checkmark$
 $400 \neq 4000 \times$

Not
(True) \Rightarrow False

Q:- WAQID details of employees if they are working in Dept No. 10, or 30 and they are getting sal except 800 and 2850.

Select *
From Emp

Where Dept No IN (10, 30) AND
SAL NOT IN (800, 2850);

SAL NOT IN (800, 2850);

Q:- WAQID Name, Job, Dept No. of all employees working as CLERK or MANAGER Except Dept No. (10, 20).

Select Ename, Job, Dept No.

From Emp
Where Job IN ('CLERK', 'MANAGER')
AND DEPTNO NOT IN (10, 20);

Q:- WAQID Ename, Dept no. along with Job, If employees are working as 'SALESMAN' OR 'PRESIDENT' except the Dept No. 30,40.

Select Ename, dept.no ,Job

from emp;

where Job IN ('SALESMAN', 'PRESIDENT')
AND DEPT. NO._{Not} IN (30,40);

Q:- WAQID Salary given to all the employee along with name if employees are getting salary between 1250 to 3000.

Select ,Ename ,Sal

from Emp

where SAL >= 1250 AND SAL <= 3000;

→ Between Operator!

Whenever we have range of values, then we go for between Operator

(low range and, high range)

Syntax :- Column-name / Expression

Between lower-range AND

higher-range ;

; . $x \geq y$ AND $y \leq z$;

Salary Between 1250 AND 3000;

⇒ Sal >= 1250 AND Sal <= 3000;

($x \geq y$ AND $y \leq z$) \Leftrightarrow $x \leq z$

Q:- WAQTD details of employee who are working
as 'SALESMAN' and they are getting comm
between 0 to 500.

Select *
from Emp
where Job = 'SALESMAN' AND
comm. between 0 AND 500;

Q:- WAQTD Name, Sal, Dept No. of all the
employees if they are getting sal less than
5k. and also they are hired the year 81.

Select Ename, Sal, Dept_no.
from Emp
where Sal < 5000 AND hiredate
between '01-JAN-81' AND '31-DEC-81';

Q:- WAQTD details of Emp who are getting Sal
more than 1250 but less than 3000.

Select *

from Emp
where Sal between 1251 AND 2999 ;

NOT B/W

Q:- WAQTD Salary given to all the employees
except who are getting salary between
1250 to 3000.

Select *
from Emp
where Sal NOT BETWEEN 1250 AND 3000;

→ NOT BETWEEN :-

Whenever we have range of values then we go for NOT BETWEEN.

Syntax:- Column-name/expression NOT b/w

lower range AND higher range;

1250 AND 3500 ;

comm > 1250 AND < 3500 ;

Q: INFQTD comm. given to the 'MANAGER' AND 'SALESMAN' except who are getting comm between 300 to 500.

→ Select comm

From Emp

Where Job IN ('MANAGER', 'SALESMAN')

AND comm NOT BETWEEN 300 AND 500;

Q: INFQTD details of Emp who are getting less than 1100 but more than 2800.

Select a

From Emp

Where Sal NOT BETWEEN 1100 AND 2800;

Q: INFQTD Job, comm given to all the emps who are getting comm more than 200 but lesser than or equal to 1400.

Select Job, comm

From Emp

Where Comm IN BETWEEN 200 AND 1400;

Q: INFQTD job, comm given to all the emps whose job is 'MANAGER' and salary is 1400.

IS operator

Q:- WAPTD details of Emps. who are getting comm. & whose comm is blank.

Select *
From emp
Where comm IS NOT NULL;

→ IS operator :-
IS operator is used to check the value which is present in LHS is NULL or NOT NULL.

Syntax :- |
L.H.S R.H.S
column / expression IS NULL / NOT NULL;

Q:- WAPTD details of employees who are not getting commission.

Select *
From Emp
Where comm IS NULL;

Q:- WAPTD details of Emp. who doesn't have Reporting Manager (MGR).

Select *
From Emp
Where MGR IS NULL;

Q:- WAPTD name of an employee if employee name is 'KING'.

Select Ename
From Emp
Where Ename = 'KING'; / Ename In 'KING'

(IN) is a multiple valued operator and also single valued. IN works in both cases.)

LIKE operator

Q: What details of employees whose name starts with 'K'?

Select * from Emp where Ename like 'K%'

→ LIKE operator:- To perform pattern matching, we will go for LIKE operators.

Syntax:-

column_name/Expression LIKE 'pattern-to-match';

Ex: Employee

Like:-

To perform pattern matching we have two special character, they are percentile(%) Under-Score(_).

Percentile%:-

It can accept any character and any number of times (0, 1, 2, ..., n)

UnderScore_:-

and any numbers

It can accept any char, but only once.

Ex:-

SAMEER → 'S.%' → 'SAMEER'

MOUNIKA → '%A' → 'MOUNIKA'

SILVI → 'DEVA%' → 'DEVASENA'

BUNTY → 'A%' → 'xxxxx X'

BAHUBALI → 'DEVA%' → '%(0,1,2,...n)

DEVA SENA → '%AX%' → 'SAMEER'

GURUJI → 'BAHUBALI'

'X.AY.A' → 'BAHUBALI'

'DEVA%'

'X.IY.' → 'GURUJI'

Q: In QTD details of Emps whose name having 'S' in it.

Select *

From Emp

where Ename like 'x.sx.'

Q: WAQTD name, job, of all the emps if job ends with 'MAN'

Select job, Enname,

From Emp

Where Job like 'x.MAN';

Q: WAQTD name, Sal, Given to all the Empls.

If they're working as 'MANAGER' OR 'CLERK'

and They're Getting 3 digits.

Select Enname, Sal

From Emp

Where Job IN ('MANAGER', 'CLERK')

AND Sal like '---';

Q: WAQTD details of Empls who're getting comm
AND Their name includes char 'N' IN IT.

Select Enname
From Emp
Where comm IS NOTNULL AND Enname
Like 'x.N%';

Q: WAQTD Enname, Job along with comm , IF
Empls are getting 4 digits comm in Dept.
No. 20 or 30.

Select Enname, Job, comm

From Emp

Where comm like '----';

AND Dept No. IN (20, 30);

Q:- WAQTD details of Emps ^{And} ~~And~~ their name start with 'A' or 'M'.

Select *
from Emp
Where Ename like 'A%' OR
Ename like 'M%';

Q:- WAQTD Emp no., Name, Hiredate of Emps. If they're hired in the year 81 or 82.

Select Emp No, Ename, Hiredate.
from Emp
Where Hiredate like '%.81' OR
Hiredate like '%.82';

Q:- WAQTD Hiredate of Emps who're hired in the month of DEC or NOV.

Select Hiredate
from Emp
Where Hiredate like '%.DEC%' OR '%.NOV%';

Q:- WAQTD details of Emps who're Except president AND They're hired after 82.

Select *
from Emp
Where Job not in 'PRESIDENT' AND
Hiredate > '31-DEC-82';

Q:- WAQTD Name of Emps who's name doesn't start with 'J'.

Select Ename
from Emp
Where Ename not like 'J%';

Q: WAPTD details of emps whose name has
@ @ in it.

Select *
from Emp
where Ename like '%@%';

Q: WAPTD details of emps whose name has
\$ \$ in it.

Select *
from Emp
where Ename like '%.\$.%';

Q: WAPTD details of emps whose name
has (y.) in it.

Select *
from Emp
where Ename like '%.!.%.%';

NOT LIKE:-

[column-name/Expression Not like 'pattern-to-
match'];

Ex:-

not match

emp 'HARRISON' not in the table

constraint is broken

constraint violated by the value

constraint violated by the value

constraint violated

and match

constraint violated by the value

Escape Character:-

To remove special behavior of special char and convert it in to ordinary character. which is present next to it. but only once.

Syntax

Column-name/Expression like 'pattern-to-match' escape 'A char';

$\boxed{\text{Escape} = '!' ;}$

Wild Characters:-

It is the sense, percentile('x') and underscore('_').

'! y.y!'

'y.!y.!y.y.'

'x.! - ! - y.' Escape → ;

'_ ! x.y.' and underscore symbol

' - ! - y.'

Q:- WAP TO details of Emps whose name does not start with 'M'.

Select →

From Emp

Where Ename not like 'M%'

into Emp Command and its output

Q: WAPTD name of an employee if the name does not start with 'M' or 'C'.

Select Ename

from Emp

where Ename not like 'M%' And

where Ename not like 'C%' ;

{ and & logical operator}

Q: WAPTD details of Employee, if the employee name having atleast one 'A'.

Select *

from Emp

where Ename like '%.A%';

Q: WAPTD name of emp if emp name having exactly one 'A'.

Select Ename

from Emp

where Ename like '%.A%' And Ename

not like '%.A%.A%' ;

Q: WAPTD details of emp if the job having exactly 2A.

Select *

from Emp

where Job like '%.A%.A%' and Job

not like '%.A%.A%.A%' ;

Q: What is the name of an emp having exactly 2L?

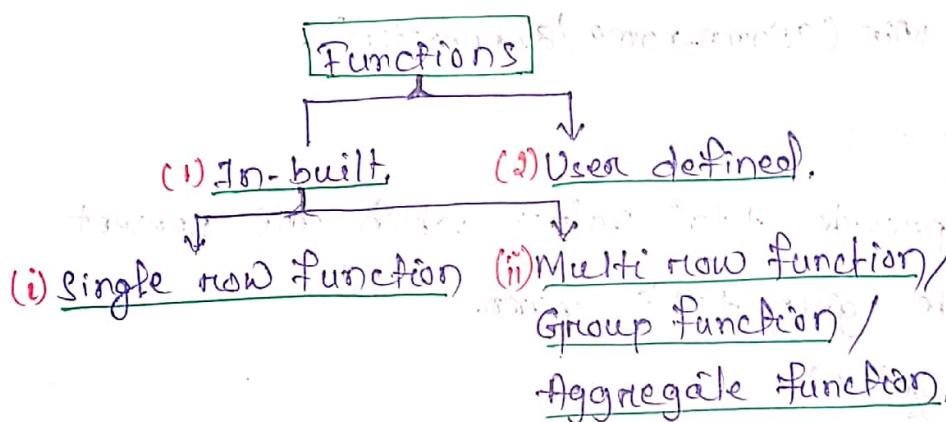
Select Ename

from Emp

Where Ename like 'x.Lx.Lx.' and Ename not
like 'x.Ly.Ly.Ly.'

→ Functions: - (user-defined functions) with
Symbol - '()

Functions are either set up instruction or
block if codes which are used to perform
some specific task.



1. Single Row Function:-

- It takes 'n' number of input and gives 'n' number of output.

- Single row function executes row by row.

2. Multi Row Function:-

- It takes 'n' number of input but gives single output.

- Multi row function executes Group by Group.

* Types of Multicolumn function:-

1. MAX():-

- To provide maximum values which are present in the given table, column.

Syntax:-

Max (Column-name/Expression);

2. MIN():-

- To provide minimum values which are present in the given table, column.

Syntax:-

Min (Column-name/Expression);

3. SUM():-

- To provide total values which are present in the given table, column.

Syntax:-

Sum (Column-name/Expression);

4. AVG():-

- To provide Aggregate or Average value which are present in the given table.

Syntax:-

Avg (Column-name/Expression);

Copy this topic to revision part of notes file.

copy & convert into notes

5. COUNT():-

To provide number values which are present in the given column.

- Only for count function, we can pass '*' as an argument.

Syntax:-

`Count(* / column-name / Expression) ;`

* Properties of multi-row Functions:- (MRF)

- Multi Row Function (MRF) executes group by group.

- We can use multiple multi row function in Select clause.

- We can't use MRF in where clause.

- We can't use column name or expression along with MRF.

- For multi Row Function we can pass only one argument, that is column name or expression.

- MRF ignores null values.

Q1. What `max(sal), min(comm)` present in the table.

`Select max(sal), min(comm)`
From Emp;

Q: WAQTD Total Sal needed to pay for all the employees who're working as manager.

Select Count(*)

From Emp

Where Job in 'MANAGER';

Q: WAQTD Avg. comm given to emps; who're working in DeptNo. 20 or 30.

Select Avg(comm)

From Emp

Where DeptNo in(20,30);

Q: WAQTD No. of Emps working in DeptNo. 20;

Select Count(*)

From Emp

Where DeptNo = 20;

Q: WAQTD max, min, avg, sum Sal Given to

all the Emps who're doesn't start with 'M'.

Select max(sal), min(sal), Avg(sal), sum(sal)

From Emp

Where Ename Not like 'M%';

Q: WAQTD details of Emps who are getting comm without using Null.

Select #

From Emp

Where Comm is Not Null;

Q: WAQTD No. of Emps hired in the month of DEC or JAN.

Select count(*)

from Emp

where hiredate like '%DEC%' And hiredate
like '%JAN%'

Q: WAQTD total Sal needed to pay for all
the emps whose Job includes 'y.' as a 2nd
char.

Select sum(Sal) from Emp

from Emp

where Job like '_%y.%'

Q: WAQTD No. of Emps getting comm in Dept.
No. 20 or 30 if they are SALESMAN.

Select count(*)

from Emp

where comm is Not Null And DeptNo in(20,30)
And Job = 'SALESMAN'

Q: WAQTD Ename of Emps who're not working
as president.

Select Ename

from Emp

where Job Not in 'PRESIDENT'

Q: WAQTD Recently / last hired Emps hiredate.

St. 1. Last hired date with jobtitle
2. Second last hired date (with jobtitle)

Q1. WAPTD first hired Emps Hiredate.

Q2. WAPTD No. of DeptNo. present in your Emp table

Select Count(*)

From Emp

Select Count(DeptNo)

From Emp;

Q3. WAPTD No. of Job present in your Emps table.

Select Count(Job)

From Emp;

Q4. WAPTD no. of Emps whose name includes '_' as a 3rd character or and last character.

Select Count(*)

From Emp

Inhere Ename like '---1---' And Ename like 'x!---'

Q5. WAPTD no. of Emps hired during the year 89 or 87 or after 88.

Select Count(*)

From Emp

Where Hiredate like 'x.82x.' And Hiredate like 'y.87x.' And Hiredate > '31-DEC-88';

Group By Clause:-

It is used to Group the records.

- Group by clause executes row by row,
- Group by clause will executes after the execution of from clause.
- With or without using where clause we can use group by clause.
- Group by clause execute row by row but after execution of Group by clause, it creates group.
- So that any clause which executes after the execution of Group by clause, It executes over Group by Group.

Group by expression;

The column name which is written in group by clause.

Syntax :- Select MRF (group function)/group-by-expression

From table name
With having clause (if used) of group -
[where <filter condition>] is optional

Ques. 2. Group by column name/expression.

Order of Execution:-

1. From
2. Where (if used)
3. Group by
4. Select

Sequence of Execution:

From bottom up to top (row by row)

1. from
2. Where - row by row
3. Group by - row by row

4. Select - Group by Group

or partition in divisor simple formulas will

be used of group

Q1: WAQT D max(sal) present in each dept.

2. Select max(sal), DeptNo.

1. From Emp.

2. Group by DeptNo;



① From

EmpNo	Ename	Sal	DNo.
1	'Sundar'	800	20
2	'Macha'	500	10
3	'Sundar'	600	20
4	'Gurujji'	900	30
5	'Macha'	700	10

OP
from

② Group by
DNo=20

1	Sundar	800	20
2	Sundar	600	20

DNo=10

2	Macha	500	10
5	Macha	700	10

DNo=30

4	Gurujji	900	30
---	---------	-----	----

③ Select

Max(sal)	DNo.
800	20
700	10
900	30

Q1: WAQT D min(sal) present in each Dept.

Select min(sal), DeptNo.

From emp

Group by DeptNo.

1. WAPTD Total Salary needed to pay for all the clerks in each Dept.

Select sum(sal), Deptno,
from emp

Group by Deptno;

Deptno	Job	Sal	Count	Total
10	CLERK	240	5	1200
20	CLERK	240	5	1200
30	CLERK	240	5	1200

2. WAPTD no. of emps working in each

Job except PRESIDENT.

Deptno	Job	Count
10	CLERK	5
20	CLERK	5
30	CLERK	5

where job not in "PRESIDENT".

Group by Job;

3. WAPTD Avg comm Given to all the emps who're working as 'salesman' in each Dept.

Select Avg(comm), Deptno,

from emp

where job in 'SALESMAN'

Group by Deptno;

4. WAPTD max sal needed to pay for all the emps if they Getting Sal > 1000 in each job.

Select max(sal), job

from emp

where sal > 1000

Group by Job;

5. COUNT NO. OF EMPS HAVING CHAR 'A' IN NAME IN EACH JOB.

Select count(*), Job

From emp

Where Name like '%A%'

Group by Job;

6. COUNT NO. OF EMPS WORKING IN EACH 'Dept' EXCEPT DEPT 10 OR 40.

Select count(*), DeptNo.

From emp

Where DeptNo. not in (10, 40)

Group by DeptNo.

7. COUNT MAN SAL NEEDED TO PAY FOR ALL THE ANALYST OR MANAGER.

Select mon(sal)

From Emp

Where Job in ('ANALYST', 'MANAGER');

8. COUNT NO OF EMPS WHOICE GETTING SAL MORE THAN 2K, BUT LESS THAN 5K IN EACH DEPT.

Select count(*), DeptNo.

From emp

Where sal between 2001 AND 3999.

Group by DeptNo.

9. What max, min, sum, (sal) given to all the emps who're not getting hired in the month of Dec in each Job;

Select max(sal), min(sal), sum(sal), Job
from emp
where hiredate not like '%Y-DEC-%'
Group by Job;

10. What max sal given to all the emps who're getting sal more than 2K but lesser than 5.5K in each Dept.

Select max(sal), DeptNo.
from emp
where sal > 2000 And sal < 5500
Group by DeptNo.

11. What max sal given to emps who're getting sal more than 2000 but max sal lesser than 4200 in each Dept.

Select max(sal), DeptNo.

from emp
where sal > 2000

Group by DeptNo
HAVING max(sal) < 4200

HAVING CLAUSE:-

- Having clause is used to filter the groups.
- Having clause executes group by group.
- Having clause executes after the execution of group by clause.
- We can use ~~any function~~ function in HAVING clause.
- We can pass multiple conditions in HAVING clause.
- Syntax: select group-function(MRF)/group-expression from table-name
[where <filter-condition>
Group By column-name/exp
Having <group-filter-condition>];

Order of execution:-

1. FROM
2. WHERE (IF USED)
3. GROUP BY
4. HAVING
5. SELECT

NATURE OF EXECUTION:

From top to bottom (row by row)

1. FROM → ROW_BY_ROW
2. WHERE → ROW_BY_ROW
3. GROUP BY → ROW_BY_ROW
4. HAVING → GROUP_BY_GROUP
5. SELECT → GROUP_BY_GROUP

Answers of questions starts

Q:- IN QTD max(sal) present in each dept. -
if max(sal) more than 500.

- ④ Select max(sal), Dept no.
- ① from emp
- ② Group_by Dept no.
- ③ Having $(\text{sal}) > 500$

Answers of questions starts

Q:- max(sal) in each dept.

if max(sal) more than 500.

in position of having

Position of having

ANSWER

Case 1: max(sal)

Case 2: \geq

Case 3: $<$

Case 4: \leq

* Difference between where clause and HAVING clause.

WHERE

- It is used to filter the records

- It executes row by row.
 - It executes group by group.

- After the execution of from clause, we will execute where clause.

- We can use column-name / Expression in WHERE clause

HAVING

- It is used to filter
the groups.

- It executes group by group.

- After the execution of Group by Clause, we will execute where Clause.

- We can not use
column-name/expression
in HAVING clause

1. WAPTD min sal given to all the emps in each job if their min sal more than 1k.

Select min(sal), job

from emp

Group by Job

Having min(sal) > 1000;

2. WAPTD Avg sal needed to pay for ~~the~~ the 'SALESMAN' or 'MANAGER' or 'PRESIDENT' in each dept if Avg sal ~~is~~ between 800 to 3400.

Select Avg(sal), Deptno.

from emp

where Job in ('SALESMAN', 'MANAGER')

Group by Deptno.

Having Avg(sal) between 800 and 3400;

3. WAPTD total sal and comm given to all the emps in each job if they're getting Total comm > 0

Select sum(sal), sum(comm), job

from Emp

Group by Job

Having sum(comm) > 0;

4. WAPTD no. of emps working in each Dept except Deptno. 10 in which there are atleast 2 emps are working.

Select count(*), DeptNo.

from emp

where Deptno != 10

Group by Deptno

Having count(*) >= 2 ;

5. WAP/TD max sal given to all the emps if they working DeptNo. 20 or 30 in each job if their max < 5k.

Select max(sal), Job

from Emp

where DeptNo. in (20, 30)

Group by Job

Having max(sal) < 5000;

6. WAP/TD no. of emps working in each Dept in which there are at most 5 emps. curly working.

Select count(*), DeptNo.

from Emp

Group by DeptNo.

Having count(*) <= 5;

7. WAP/TD max, sum, and sum - Sal given to all the emps except ANALYST in each Job in which there are At least 2 but Atmost 6 emps are working.

Select max(sal), sum(sal), Job

from emp

where Job != 'ANALYST'

Group by Job

Having count(*) >= 2 And count(*) <= 6;

8. WAP TO SAL WHICH ARE REPEATED/DUPLICATED

Select count(*), Sal

from Emp

group by Sal

having count(*) > 1;

9. WAP TO HIREDATE WHICH ARE REPEATED.

Select hiredate

from Emp

Group by hiredate

Having count(*) > 1;

10. WAP TO NO. OF EMP'S GETTING SAME SAL AND
SAME DEPT.

Select count(*), Sal, DeptNo.

From emp

Group by Sal

Having

11. WAP TO NO. OF EMP'S HIRED ON SAME DAY
INTO SAME DEPT.

Select count(*)

Q:- What number of emps doesn't have E in job in each department.

Select count(*), DeptNo

From Emp

Where Job ^{not} like 'Y.E.Y.'

Group by Dept.No;

Q:- What sal in descending order.

Select Sal

From Emp

Order by Sal Desc;

Order By :-

Order by clause is used to arrange the result table either in descending or ascending.

- After the execution of select clause, order by clause will execute.
- Always order by clause should be the last argument in any query.
- We can use ALIAS name in order by clause.
- All the tables must be arranged in ascending order based on Primary key.
- By default it will consider Ascending Order.
- We can write multiple arguments and in order by clause.
- Example

Select Ename, Sal

from Emp

Order by Sal, Ename;

* Select Ename, Sal
from Emp
Order by 2 ;

Syntax:

Select group function(MAF)/Group by exp
From table-name
[Where {filter-condition}]
Group by column-name/Exp
Having {group-filter-condition}
Order by column-name/Exp/ALIAS/NUM
(1,2,...n) [ASC]/[DESC] ;

Q) What SQL which finds repeated more number of times.

Selected SQL Count(sal), sal
From Emp
Group by sal
Having count(sal)

* Order of Execution:-

From From Group Having Select Where

2. [Where]

3. Group by

4. Having

5. Select

6. Order By

NOTE:-

Where clause will work for Boolean.

Select, From, From, Group by, Having, Order By

Group by will follow having, but not vice versa.

Having @ group after all of this,

and after that from, select, group by, etc.

group

Having work when it applies itself to

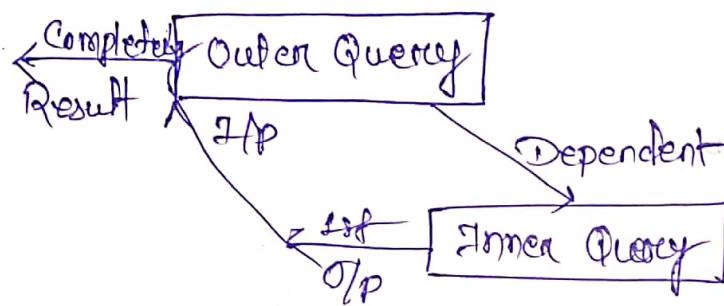
From, From, Group by, Having, Order By

, From, From

Sub-Query :-

Query written inside another Query is called Sub-Query Sub-Query.

Working principle of Sub-Query:-



- Inner Query executes 1st and generate Output.
- Output generated by the Inner Query given to the outer query as Input.
- Outer Query taking input from the inner Query.
- After taking inputs from ^{inner} query. Outer query executes completely and generate result.

- Without Inner Query, Outer Query will not execute.
- So the outer Query is dependent on inner Query.
- Outer Query is responsible for generation output.

* Why or when do we go for Sub-Query?

Case 1: Whenever we have unknown values then we go for Sub-Query.

Ex:- Write a query to display name of an employee who are getting salary > 600.

Select Ename

Ans :- $\{ \text{Select Ename from Emp where } \text{Sal} > 600 \}$

Q1:- WAP TO name of an employees who are getting salary more than Sundari's salary

Select Ename

$\{ \text{Select Ename from Emp where } \text{Sal} > 600 \}$

$\{ \text{Where } \text{Sal} > 600 \} \{ \text{Select Sal from Emp} \}$

$\{ \text{Where } \text{Sal} > 600 \} \checkmark$

$500 > 600 \times$

$600 > 600 \times$

$700 > 600 \checkmark$

$\{ \text{from Emp where Ename = 'Sundari' } \}$

→ List inner query executes

Sundre = Sundrei ; X

Machha = Machhi ; X

Sundrei = Sundrei

Machha ≠ Sundrei ; X

EmpNo	Ename	Sal
1	'Sundre'	700
2	'Machha'	500
3	'Sundrei'	600
4	'Machhi'	900

Sundrei

Machha

* Conditions to write Sub-Query

(i) From Inner Query we can select only one column name,

(ii) The column name selected from the Inner Query and the column name which are written in outer Query both the column should be having same data type.

Ques :- What is answer



Q. WAP TO Details of emps who are getting salary more than Smith salary.

Select *
from Emp > (Select *
where Ename = 'Smith');

Q: WAP to name and hiredate of all the employees who are hire after martin.

Select Ename, hiredate
from Emp
where hiredate > (Select hiredate
from Emp
where Ename = 'martin');

Q: WAP to Name, Job of an employee along with Dept no. of employees are working in same as miller's job.

Select Ename, Job, DeptNo,
from Emp
where Job = (Select Job
from Emp
where Ename = 'Miller');

Q:- WAQT DetaiLS oF Emps who are working
in same Dept as Adams department.

Select * From Emp

where DeptNo IN (Select DeptNo

From Emp

where Ename = 'Adams');

Identify more tools

Q:- WAQT Name, Hiredate of all emps along
with Job of they are working as salesman
and they are hire Before SCOTT.

Select Ename, Hiredate, Job

From Emp

where Job = 'Salesman' and

Hiredate < (Select Hiredate From Emp

where Ename = 'SCOTT');

Q:- WAQT DetaiLS oF employees who are
getting sal more than Smith but they
are hired after 81.

Select *
from Emp

where Hiredate > '31-DEC-81' and
Sal > (Select Sal from Emp
where Ename = 'Smith'));

Q: To find details of employee who are getting
Sal more than Smith but less than King.
Select * from Emp where Ename = 'Smith'

from Emp where Ename = 'King'

where Sal > (Select Sal from Emp where Ename = 'Smith')
and Sal < (Select Sal from Emp where Ename = 'King')

And Sal < (Select Sal from Emp
where Ename = 'King'));

Q: To find no. of emps hired after 'blake' but
they are hired before James.

Select count(*), Hiredate from
Emp

where Hiredate > (Select Hiredate

from Emp where Ename = 'blake')

and

Hiredate < (Select Hiredate
from Emp

where Ename = 'James'))

Q1: WHATD max(sal) given to all the salesmen if they are working in same department as manager dept.

Select max(sal)

from Emp

where Job = 'SALESMAN' And,

deptNo. = (Select deptNo. from

(Select deptNo. from Emp where
ename = 'MARTIN'))

Q2: WHATD No of Crops Hired after 82 but they are getting some kind of FORD's salary.

Select count(*) from fact2

from Emp

where hireDate like '82%'

And Job = (Select sal

from Emp

where ename = 'FORD'))

Student for 100% marks

Ques 103/2

(WAPC) Exam 2020

Rule-2 :-

Whenever to be displayed

- The data which is present in one table and condition to be executed in another table then we will go for sub-query.

Ex:-

IN A QTD DeptName of Sundri

Emp

EmpNo	Ename	Dno.
1	'Sundri'	d 0
2	'Machhi'	10
3	'Sundri'	30
4	'Machhi'	10

Dept

Dno.	Dname	Loc.
10	'P/c'	'HYD'
20	'Sales'	'GOA'
30	'HR'	'BNGLR'

Dname of 'Sundri' (HR)

Select Dname

from Dept
where Dno = (select Dno.

$$10 = 30 \times$$

$$20 = 30 \times$$

$$30 = 30 \checkmark$$

from Emp
where Ename = 'Sundri';

Sundri = Sundri \times

Machhi = Sundri \times

Sundri = Sundri \checkmark

Machhi = Sundri \times

Q. 10AQTD. Dept Name, if Enf: Scott

~~Select Dname
from EmpDept
where Deptno = (Select Deptno
from~~

~~Select Dname~~

~~from Dept~~

~~where Deptno = (Select Deptno~~

Deptno	Deptname
10	RESEARCH
20	DESIGN
30	SALES

Q. 11AQTD. location name of Smith

~~Select loc
from Dept
where Deptno = (Select Deptno~~

~~from Emp~~

~~where Ename = ('SMITH')~~

Q. 11AQTD. details of Emp who are working
in sales dept.

~~Select *
from Emp~~

~~where Deptno = (Select Deptno~~

~~from Dept
where Deptname = ('SALES'))~~

Select *

from Emp

where Deptno = (Select Deptno
where Dname = 'SALES');

Q1:- WAQT D No. of Empls working in Chicago.

Select count(*)

from Emp

where Deptno = (select Deptno,

from Dept

where loc = 'chicago');

Q2:- WAQT D Dept details of all the salesman
who are working in sales dept id (1)

Select *

from Dept

where Deptno = (select Deptno,

from Emp

where Job = 'SALESMAN') And

Deptno = 'SALES');

Q3:- WAQT D details of Empls who are working
acceting dept.

Select *

from Emp

where Deptno = (select Deptno,

from Dept);

where Dname = 'ACCOUNTING');

Q. What default of Emps who are working in accounting or research Dept.

```
Select *
  From Emp
 Where DeptNo In (Select DeptNo
                      From Dept
                     Where Dname In ('Accounting',
                                         'Research'));
```

* Types of Sub-Query:-

(1) Single row Sub-Query

(2) Multi row Sub-Query

(1) Single row Sub-Query:-

If the sub-query returns only one records or single row is known as single row Sub-Query.

In single row Sub-Query we can use normal Operations such as equal ($=, <, >, \leq, \geq, \neq$)

if there is no unique operation

(2) multi Row sub-Query:-

- If the sub-query returns more than one records or multi row then it is known as multi Row sub-query.
- For multi row sub-query we can't use normal operators. but we can use special operators such as (IN, NOT IN, ALL, Any, Exist, Not Exist)

Q. What details of Emps who are working in Chicago or Dallas.

```
Select *  
from Emp  
where DeptNo In (Select DeptNo,  
from Dept  
where Loc in ('Chicago',  
'Dallas'));
```

Q. What Department and Location of all the Emps who are working as 'Salesman' and as 'Analyst'

```
Select Dname, Loc  
from Dept  
where Deptno In (Select Deptno  
from Emp  
where Job in ('SALESMAN',  
'ANALYST'));
```

Q. WAPTD details of Emps who are
working as clerk in 'SALES' or
'Research' Department & their Salaries
are less than 10000/-

Select loc from Emp
Details, job from Emp, job in ('SALES' or
'Research')

where job = 'CLERK' And

Dept no in (Select Deptno.)

from Dept

where Dname('SALES',

'RESEARCH'))

Q. WAPTD location of Emps except
President who are working in Chicago
or New York.

Select loc

from Emp Dept

where ~~Deptno~~ 'PRESIDENT' And

loc in ('chicago', 'newyork') And

Deptno in (Select Deptno.)

from Emp

where job not
in ('PRESIDENT')
and job in ('SALES',
'RESEARCH',
'CLERK')

where job not
in ('PRESIDENT'))

Q. 10AQ7D details of emps who are working
 in 'Accounting' or 'research' or 'operations'
 Dept, and they are getting ~~sal~~ more
 than 2000.

Select *

from Emp

where Ename Sal > 2000 And DeptNo. in
 (Select DeptNo.

from Dept

where Dname in ('Accounting', 'Research', 'Operations');

Q: wanted details of employees ~~in the table~~
 getting sal more than martin's ~~more~~

Select *

from Emp

where Ename = (Select sal

from Emp

where Ename = 'Martin');

Q:- wanted DeptName of emp who are working

a) manager.

Select Dname

from Dept

where DeptNo in (Select DeptNo

from Emp

where Job in ('manager'));

Q1:- WAP TO Deptname of Emps who are getting sal more than Biller

Select Deptname

from Dept.

Where Deptno in (Select Dept no

from Emp
where Sal > (Select Sal)

(Select Deptno from Dept where Ename = 'Biller')

Q2:- WAP TO details of Emps whose are

~~Emp~~ working in Accounting Dept and

they are getting Sal < King.

Select #

from Emp

where Deptno in (Select Deptno Sal)

from Dept Emp

where Ename = 'King'

and Deptno in (Select Deptno

from Dept

((Citation = 'MGR' or Ename = 'MGR'))

where Dname =
~~ACCOUNTING~~
'ACCOUNTING')

where Deptno

in (Select Deptno

from Dept where

Deptno = 101

((Supervisor of dot. Enamel

Q:- WAPTD Dept.name in which there are atleast 3 salesmen are working.

Select Deptname
from Dept.

~~where Deptno~~

Where Deptno in (Select Deptno.

(~~Count(*)~~ from Emp
Group by Deptno
Having Count(*) >= 3);

~~Count(*)~~ from Emp
Group by Deptno
Having Count(*) >= 3);

Q1:- WAPTD name of an emp who are working
chicago as manager or clerk.

Select Ename, job

((From Emp) job = 'MANAGER' OR job = 'CLERK' AND
Where Deptno in (Select Deptno

from Dept

Where job loc in 'chicago');

Q1:- WAPTD details of Emps who are getting
Sal more than JONES.

Select, * group by
(From Emp) job = 'MANAGER'
Where Sal > (Select sal

from Emp

Where Ename = 'JONES');

Q1- ~~WAP~~ details of Emps who are getting Sal more than all the Salesman.

~~Select *
from Emp
where sal > (Select count(sal)
from Emp
where job = 'SALESMAN'));~~

→ ~~Select *
from Emp
where sal > (Select max(sal)
from Emp
where job = 'SALESMAN'));~~

~~(Opposite) of sal do profit
sign profit~~

→ ~~Select *
from Emp
where sal > all (Select sal
from Emp
where job in ('SALESMAN'));~~

~~For profit < 602 profit
amt profit
else 23604C common error~~

ALL OPERATOR:-

All

All operator is a multi valued operator which has multivalues at R.H.S and single values at L.H.S along with Relational operation.

Syntax: $\{100 > 200, 300 < 400\}$

Co/exp Rel operation All (v_1, v_2, \dots, v_n);

(v_1, v_2, \dots, v_n)

Ex:- $\{100 > 200, 300 < 400\}$

$100 > 200$ T

$300 < 400$ T

$200 > 300$ F

} False

As False

Ans mark

{100 > 200, 300 < 400} will be ans

NOTE:-

All operator returns true when all the conditions will true.

ANY OPERATOR:-

Any operator is a multi valued operator which has multivalues at R.H.S and single values at L.H.S along with Relational operation.

NOTE:- Any operator returns true when any one of the condition will true.

Q:- What's name of an employee who are getting salary more than all the manager.

Select Ename
from Emp

where Sal > (select max(sal))
from Emp
where job = 'MANAGER';

Q:- What's details of emps who are getting
Sal > atleast 10 salesman

Select #

From Emp

where Sal > Any (Select max(sal))

order by sal desc from Emp
where job = 'SALESMAN');

without Any

without Any if there is no row in table with
some condition then it will return null

from Emp
where Sal > (select min(sal))

From Emp
where job = 'SALESMAN');

order by sal desc from Emp
where job = 'SALESMAN');

and then partition with respect to emp

Q:- WAP Q7D max(sal) from Emp where
 Select max(sal)
 from Emp;

Q:- WAP Q7D first max(sal)
 Select max(sal)
 from Emp;

Q:- WAP Q7D details of emps who are getting
 sal.
 Select all qms (from)
 (from Emp) < result
~~(from Emp)~~
 hence sal is internal;

Q:- WAP Q7D and min(sal)

Select max(sal)
 from Emp
 where sal < (select max(sal))
 from Emp;
 (12) $900 > 900$

800	900
700	$800 < 900 \checkmark$
900	$700 < 900 \checkmark$
800	$700 < 900 \times$
600	$800 < 900 \checkmark$
500	$600 < 900 \checkmark$
	$500 < 900 \checkmark$

* NESTED SUB-QUERY :-

We can nest up to 255 sub-queries.

Q:- Third min(sal)

Select min(sal)

From Emp

Where sal > (Select min(sal))

From Emp

Where > (Select min(sal))

From Emp)

(Number of sub queries)

Q:- WAPD details of Emps who are getting
3rd max sal.

Select *

From Emp

Where sal > (Select max(sal))

From Emp

Where sal < (Select max(sal))

From Emp

Where sal < (Select max(sal))

From Emp);

00P

00P > 00F <

00P > 00P <

00P > 00S <

00P > 00A <

00P > 00B <

00S

00A

00B

00D

00C

00E

00F

00G

00H

00I

00J

00K

00L

00M

00N

00O

00P

00Q

00R

00S

00T

00U

00V

00W

00X

00Y

00Z

00A

00B

00C

00D

00E

00F

00G

00H

00I

00J

00K

00L

00M

00N

00O

00P

00Q

00R

00S

00T

00U

00V

00W

00X

00Y

00Z

00A

00B

00C

00D

00E

00F

00G

00H

00I

00J

00K

00L

00M

00N

00O

00P

00Q

00R

00S

00T

00U

00V

00W

00X

00Y

00Z

00A

00B

00C

00D

00E

00F

00G

00H

00I

00J

00K

00L

00M

00N

00O

00P

00Q

00R

00S

00T

00U

00V

00W

00X

00Y

00Z

00A

00B

00C

00D

00E

00F

00G

00H

00I

00J

00K

00L

00M

00N

00O

00P

00Q

00R

00S

00T

00U

00V

00W

00X

00Y

00Z

00A

00B

00C

00D

00E

00F

00G

00H

00I

00J

00K

00L

00M

00N

00O

00P

00Q

00R

00S

00T

00U

00V

00W

00X

00Y

00Z

00A

00B

00C

00D

00E

00F

00G

00H

00I

00J

00K

00L

00M

00N

00O

00P

00Q

00R

00S

00T

00U

00V

00W

00X

00Y

00Z

00A

00B

00C

00D

00E

00F

00G

00H

00I

00J

00K

00L

00M

00N

00O

00P

00Q

00R

00S

00T

00U

00V

00W

00X

00Y

00Z

00A

00B

00C

00D

00E

00F

00G

00H

00I

00J

00K

00L

00M

00N

00O

00P

00Q

00R

00S

00T

00U

00V

00W

00X

00Y

00Z

00A

00B

00C

00D

00E

00F

00G

00H

00I

00J

00K

00L

00M

00N

00O

00P

00Q

00R

00S

00T

00U

00V

00W

00X

</

Q1 - WAP TO loc of an Emp ~~who are getting 4th min(sal)~~ (Ans) from Feb(2) to last month

Select loc

from dept

Where DeptNo in (Select DeptNo from Emp Where sal in (Select min(sal)

> (from Emp

where sal > (Select min(sal)

from emp)

where sal > (Select min(sal)

(Ans) from Feb(2) to last month

from Emp

ans point

where sal >

select min(sal)

from emp))

(Ans) from Feb(2) to last month

ans point

> last month

from Emp

where sal >

select min(sal)

from emp))

Q1 - WAP TO details of Dept ~~who are getting 4th max(sal)~~

Imp 10th max(sal).

Select loc from Emp Dept

{ ans point

from

Select d

from Dept

where sal =

(Select max(sal))

from Emp

where sal < (Select max(sal))

from Emp

where sal < (Select max(sal))

from Emp

where max(sal) <

select max(sal)

from Emp

where sal < (select max(sal))

from Emp

similar ques mark diff
where sal < (select max(sal)),

(102) from Emp Pri 102

where sal <

(select max(sal)) from Emp

from emp

where sal < (select max(sal))

similar ques < 102 ans diff

ans mark

< 102 ans diff

similar ques

< (102 ans diff)

from Emp

where sal < (select max(sal))

from Emp

where sal <

(select max(sal))

from Emp;

similar ques mark ans diff

ans mark

to 102

+ 102 month

= 102 month

(102) month factor

ans mark

(102) month factor > 102 month

ans mark

(102) month factor > 102 month

ans mark

> 102 month factor

Q1 WAP TO Details of all employees whose are getting 15th max(sal).

Q2 WAP TO details of 15th max(sal) is table

Type-1

Q3 WAP TO manager name of 15th max(sal)

Empno	Ename	MGR
1	A	3
2	B	None
3	C	1
4	D	2

Select Ename

from Emp

Where Empno = 2 (select MGR = 2)

from Emp

1=2 X Where Ename = 'D');

2=2 ✓

3=2 X

4=D X for B = None

Ques ans D X

((Ques ans D X))

Q4 WAP TO manager name of Jones.

Select Ename from Emp

Where Empno = 2

select mgr from Emp

from Emp

Where Ename = 'Jones');

ans 1

Q5 WAP TO Blakes manager details.

Select A

from Emp

Where Empno = 2

Select MGR

from Emp

Where Ename = 'Blakes')

Q1. WAP TO manager's name of job as clock

Select Ename

From Emp

Where Empno IN (Select MGR

From Emp

Where Job = 'clock')

Ename	Fname	Mname	Lname
Smith	John	David	Smith
Miller	James	Mark	Miller
Allen	James	Mark	Allen
King	James	Mark	King

Q2. WAP TO location name of miller's manager.

Select Loc from Dept where DeptNo IN

(Select Loc From Dept No =

From DeptEmp

Where Empno IN (Select MGR

From Emp

Where Ename = 'miller'));

Output :- Smith reports Allen

Q3. WAP TO Smith's manager's manager's name

Select Ename

From Emp

Where EmpNo IN (Select MGR

From Emp

Where EmpNo IN (Select MGR

From Emp where EmpNo IN (Select MGR

From Emp

Where Ename

'Smith'));

(Output :- King is Manager of Miller)

Q. What is the location name of Turner's manager's manager's office?

Select loc

from Dept. of Geology & Geophysics, Univ. of Minnesota, Minneapolis 10.

where Deptno. is in Select Deptno.

from Emp

where @mpnlo. in (see)ect MGK

from Bomp

iPhone 8mpx 10.

from EMP

From Emp
Woods Run

Where Enzyme

= 'Turner'))

signed
agreement

Assignment

Q1- What is manager's name?

Select Enamel from Emp

~~Whichever approach is selected must be consistent with Article 18~~

ଶ୍ରୀମତୀ କମଳା

where EmpNo, in (Select MGR

from one

Whose Camp

where EmpNo. is

Q103 (Select MGR)

स्त्रीमि उप

Salinas 3-200

Introducing
Enhanced

'marEdin?')])

• 100 •

Type - 2:

4. write program to print 10 point reporters

Q:- WAQID name of employees who are reporting to 'B'

Empno	Ename	MGR
1	A	2
2	B	Null
3	C	2
4	D	21

Select Ename

from Emp

where MGR = (Select Ename

from Emp
Where Ename = 'B');

Q:- WAQID no. of employees reporting to 'KING'

Select count(*)
from Emp

where MGR in (Select Empno.
from Emp
Where Ename = 'KING'));

Q1 WAQTD. Details of employees who are reporting to 'SALES' manager

Select Comno
from Emp
where MGR in
(Select Empno.
from Emp
where Job in 'SALESMAN');

Q2 WAQTD name of manager employee whose are reporting to 'blake' man
Select Ename
from Emp
where MGR in
(Select Empno.
from Emp
where Ename = 'BLAKE');

Q3 WAQTD Dname who are reporting to 'KING'

Select Dname
from Dept
where Deptno in (Select Deptno.
from Emp
where MGR in (Select Empno,
from Emp
where Ename = 'King'));

Q1) IN Q1 D^Q no. Kif M^{GR} emps who are reporting from ~~turner~~ to ~~turner~~ manager.

Select count(*)
from emp
where MGR in (Select Empno. MGR
from emp
where Ename in ('Turner'));

Select count(*)
from emp
where MGR in (Select Empno. MGR
from emp
where Deptno = 30
and Ename in ('Turner'));

Number of rows where OTD is

'Prix' is
less than

more than
where he is in 60 department

original table is given below

Ques part

Final answer

Q1: WAQT D details of emps reporting to
SALESMAN'S MANAGER

Select *
from Emp
Inhere MGR in Select EmpNo.
from Emp
Inhere EmpNo. in
(Select MGR from Emp
where Job in 'SALESMAN');

Q1: WAQT name of an Emp who are
acting as reporting manager.

Select Ename
from Emp
where EmpNo. in (Select MGR
from Emp
where MGR is not null);

Q1: WAQT details of emps who're not acting

as MGR
Select *
from Emp
Inhere EmpNo. not in (Select MGR
from Emp
where MGR is not
null);

Q4: WAP/TD details of Emps who are having atleast 3 reporting.

Select *
from Emp
Where EmpNo. in (Select MGR
from Emp
Where MGR is Not Null
Group by MGR
(Having Count(*) >= 3));

Q5: WAP/TD no. of Emps who are directly reporting to king.

Select count(*)
from Emp
Where MGR in (Select EmpNo.
From Emp
(Where Ename = 'KING'));

Q6: WAP/TD no. of emp's indirectly to king.

Select count(*)
from Emp
Where MGR in (Select EmpNo.
From Emp
(Where Ename = 'KING'));

Select count(*)

From Emp

Where MGR in (Select EmpNo.

From Emp

Where MGR in (Select EmpNo.

Date / /

Page No.

Q: Info ID Sal which are Repeated.

Select Sal from Emp

Group by Sal

Having count(*) > 1;

Q: MGR Details of Emps who are getting same Sal.

Select *

From Emp

Where Sal in (Select Sal From Emp group by Sal)

Having count(*) > 1;

Not null Not null

Not null Not null

For output of this result to show

Sal for 2nd max left up

Sal for 2nd max left up for other sal?

Process 2nd max of the previous ad 1st

Result of 2nd max of 1st

Left ← 2nd max of 1st part

Then left of 2nd max of 1st part

of 2nd max of 1st part

Sal for 2nd max of 1st part

It is necessary to left part

JOINS :-

Retrieval of data from multiple table

Simultaneously is called joins.

Types of Joins:-

① - Cross Join / Cartesian Join

② - Inner Join / Equijoin

③ - Natural Join

④ - Outer Join (Left Outer Join)

i. Left Outer Join

ii. Right Outer Join

⑤ - Self Join

① Cross Join / Cartesian Join :-

Records of table-1 will be merged with all the record's of table-2.

(i) Total no. of columns in the result table will be summation of all column's present in table-1 and table-2.
Total no. of columns $\rightarrow T_1 + T_2$

(ii) Total no. of records in the result table will be product of record in table 1 and table 2.

$$\text{Total No. of records} = T_1 \times T_2$$

American National Standard Institute \rightarrow "ANSI"
 Oracle, ANSI

Date / /
 Page No.

Syntax:

ANSI \rightarrow

SELECT column-name(s) FROM table-name \rightarrow
FROM table-name1 CROSS JOIN table-name2 \rightarrow

ORACLE \rightarrow

SELECT column-name
FROM table-name1 table-name2 \rightarrow

Q:- WAQT D boys Name And Girls Name.

Boys			Girls	
BID	Bname	GID	GID	Gname
1	'Pinto'	2	1	'Munni'
2	'Munna'	1	2	'Pinti'
3	'Raja'	3	3	'Rani'

Select
 Bname, Gname
 from Boys, Girls;

Total no. of Columns \rightarrow 7 + 7 = 14

Total no. of Records \rightarrow 7 * 7 = 49

BID	Bname	GID	GID	Gname
1	'Pinto'	2	1	'Munni'
1	'Pinto'	2	2	'Pinti'
1	'Pinto'	2	3	'Rani'
2	'Munna'	1	1	'Munni'
2	'Munna'	1	2	'Pinti'
2	'Munna'	1	3	'Rani'
3	'Raja'	3	1	'Munni'
3	'Raja'	3	2	'Pinti'
3	'Raja'	3	3	'Rani'

2. Inner / Equi Joins :-

- It is used to obtain only matched records.

Syntax:-

ANSI: `SELECT column-name
FROM Table-name1 INNER JOIN Table-name2
ON <join-condition>;`

ORACLE: `SELECT column-name
FROM Table-name1, Table-name2
WHERE & <join-condition>;`

JOIN condition:
It is a condition on which two tables are merged.

`SELECT * FROM Boys, Girls WHERE Boys.GID = Girls.GID;`

Select Brname, Gname

from Boys, Girls

Where Boys.GID = Girls.GID;

Result.

GID	Bname	GID	GID	Gname
1	Pinto	2	= 2	Pinto
2	Manni	1	= 1	Manni
3	Raja	3	= 3	Rani

Q:- WAQTD Emp Name, Job along with location
of all the Emps.

Select Ename, Job
from Emp E, Dept D
Where E.DeptNo = D.DeptNo;

Q:- WAQTD Ename along with dept name
also displays location.

Select Ename, Dname, loc
from Emp E, Dept D
Where E.DeptNo = D.DeptNo;

Q:- WAQTD Ename along with Dept name
of all the emp who are working as
clerk.

Select Ename, Dname
from Emp E, Dept D
Where E.DeptNo = D.DeptNo AND
Job = 'CLEAK';

Date / /
Page No.
Husayn

Q:- WAPTD Dname of Emps who are working

as salesman in Sales dept.

Select Dname

from EmpDept

where Dname = 'SALES' AND

(Select DeptNo

from DeptEmp

where Job = 'SALESMAN');

Select Dname

from EmpE, DeptD

where E.DeptNo = D.DeptNo

AND Job = 'SALESMAN' AND

DNAME = 'SALES'

Q:- WAPTD No. of Emps working as Analyst in

'Accounting' or 'Research' Dept.

Select count(*)

from EmpE, DeptD

where E.DeptNo = D.DeptNo

AND Job = 'ANALYST' AND Dname in

('ACCOUNTING', 'RESEARCH')

Q:- WAPTD Emp details along with location

of all the Emps Except President.

Select EmpId, Loc

from EmpE, DeptD

Where Job != 'PRESIDENT';

Scanned by CamScanner

Scanned by CamScanner

Scanned by CamScanner

Scanned by CamScanner

Q) WAP TO Enamel, Sel. along with location of all the employees who're located in Chicago and they're getting salary more than 900.

Select Enamel, Sel, Loc from Emp & Dept
from Emp E, Dept D
Where E. DeptNo = D. DeptNo AND Loc = 'Chicago'
AND Sel > 900 ;

(*) Q) WAP TO Emp details along with Dept details of all the manager or analyst if they are working in accounting or Sales Dept.

Select Emp. # Dept. # from Emp, Dept
Where E. DeptNo = D. DeptNo AND Job IN
('MANAGER', 'ANALYST') AND DNAME IN
('ACCOUNTING', 'SALES') ;

Q) WAP TO Dept details of Emp who're working in Dallas or New York as Salesman or Clerk,

Select Dept. # from Dept D, Emp E
Where Loc in ('Dallas', 'New York') AND
Job IN ('SALESMAN', 'CLERK') AND
E. DeptNo = D. DeptNo ;

Q:- WAPTD Ename, Sal, DeptNo, Dname of all the Emps if their name having char 'A' in it, also write query for the same.

Select Ename, Sal, DeptNo, Dname
from Emp E, Dept D
Inhere Ename like '%A%' AND
E.DeptNo = D.DeptNo;

Q:- WAPTD Ename, Sal, Job, Loc of Emps who're working in Dept 20 or 30 as a clerk or president.

Select Ename, Sal, Job, Loc
from Emp E, Dept D
Inhere E.DeptNo = D.DeptNo AND
E.DeptNo in (20, 30) AND Job in
(CLERK, PRESIDENT);

Q:- WAPTD Name, Sal, Dname, Loc of all the Emps if they're getting 1st Max Sal.

Select Ename, Sal, Dname, Loc
from Emp E, Dept D
Inhere E.DeptNo = D.DeptNo AND
Sal = (Select max(Sal)
from Emp);

Q1. INAQTD total sal needed to pay for all the Emps in each department except president in which there are atleast 3 emps are working.

Select sum(sal), DeptNo,
from Emp
where job NOT IN ('PRESIDENT')
Having count(*) >= 3;

Q2. INAQTD msl(sal) along with name given to all the emps in each dept except

Dept. 10
Select msl(sal), Name
from Emp E Dept D
where E.DeptNo. NOT IN 10 AND
E.DeptNo. = D.DeptNo;
Group by Name;

Right now right help. This is my best effort in understanding. If I am wrong do let me know.

One question said that my answer is not correct. But as far as I understand right

Q1. WAP/TD total sal needed to pay for all the Emps in each department except president in which there are atleast 3 emps are working.

Select sum(sal), DeptNo.
from Emp
where job NOT IN ('PRESIDENT')
Having count(*) >= 3;

Q2. WAP/TD max(sal) along with Dname given to all the emps in each dept except Dept. 10.

Select max(sal), Dname
from Emp E, Dept D
where E.DeptNo. NOT IN 10 AND
E.DeptNo. = D.DeptNo.
Group by Dname;

(Note: max salary will be found for each department except dept 10)

The output will be like this (assuming 3 departments exist, 10 is working hours
in each department)

DeptNo. Dname MaxSal
10 Sales 10000
1 Admin 15000
2 Marketing 12000

NATURAL JOIN:-

Natural Join is similar to inner join but here we don't have to specify/write join condition.

Syntax of Natural Join :-

ANSI :- SELECT column_name FROM table1, Natural join table2;

Select column_name FROM table1, Natural join table2;
movie, where this give Cartesian product
query from joins in sql with the help of
NO oracle Syntax.

↳ If don't write join condition then it becomes cartesian join.

* We go for natural join, when we don't know the table, procedure or query.

Note:-

* Natural Join will give inner join output when there is a P.K., F.K. relationship.

* Natural Join will give Cross Cartesian join output if there is no P.K., F.K. relationship.

① Select *

From Emp Natural Join Dept ;
 O/P :- 14 rows selected { Because there is a
 P.K., F.K relationship }

② Select *

From Emp Natural Join Sal grade ;
 O/P :- 70 rows selected { Because there is
 no P.K., F.K relationship }

Advantage :-

* In Result table, it will display only one
 common column.

Q:- WAP TO name, DeptNo, Sal along with loc all
 the employee.

Select Ename, DeptNo, Sal, Loc

from Emp Natural Join Dept ;

Q:- WAP TO DeptNo, Name, Dname of all the clerks.

Select Ename, Dname, DeptNo,

from Emp Natural Join Dept

: (f.i. here job = 'clerk')

Drawbacks :-

- In Natural Join is less efficient than inner
 Join. Because in inner Join we write Join
 condition but in natural Join we don't
 write Join condition. Compiler will search
 for P.K., F.K relationship so, it takes more
 time.

Outer Join:-

- Whenever we want to display match records along with unmatched records is known as Outer Join.

(a) Left Outer Join: - To obtain matching records along with unmatched records of left table we go for Left Outer Join.

Syntax :- ANSI : Select column-name
from table-name1 LEFT [OUTER] JOIN
table-name2;

ON table-name1 . column-name = table-
name2 . column-name

ORACLE :-

Select column-name
from table-name1, table-name2;
Where table-name1 . column-name =
table-name2 . column-name(+);

Q1. WANT all the matched records along with unmatched records of Boys Table,

From Boys table before we find matched records like religious, educational, sports etc information of A.P., A.S. and S.G.R.

Relation ship

(Left)		(P.K / P.I.K)		(Right)	
B:T:D	B: name	G:I:D		G:I:D	G: name
1	'Manu'	2		1	'Raani'
2	'Raja'	1	+	2	'Muoni'
3	'Mehha'	3		3	'Maadhav'
4	'Gurujit'	4		4	'Sundari'

matched { } un matched { }

(Oracle)

Select

from Boys B, Girls G

Alberle B.G.I.D.: G₁, G₂, D(t);

~~(ANS)~~

From Boys B 1e
John Gifford G.

ON THE OTHER SIDE $B \cdot G \# D = G \cdot G \# D$

Result		ONListence B.		
BID	Bname	GID	GID	Gname
1	'Munna'	2	= 2	'Munni'
2	'Raaja'	1	= 1	'Rani'
3	'Maithha'	3	= 3	'Maithha'
4	'Guruji'	NULL	NULL	NULL

Right Outer Join - brings relations of both tables.

To obtain matching records along with unmatched records of right table, we go for right Outer Join.

Syntax:

ANSWER Select Column-name

from table-name1 Right [Outer] Join table-name2

ON table-name1.Column-name = table.name2.

column-name ;

ORACLE:- select Column-name

From table-name1, table-name2

where table name 1. column-name(?) :

```
table-name2 . column-name;
```

Result table :-

BID	Bname	GID	GID	Gname
1	Munna	2	2	Munni
2	Raja	1	1	Rani
3	Machha	3	3	Machha
4	NULL	NULL	4	Sundri

Unmatched Records

matched Records

Select *
from Boys B, Girls G
Where B.GID(+) = G.GID;

Full Outer Join !

To obtain matching records along with unmatched records of both right and left table, we go for Full Outer Join.

Result :-

BID	Bname	GID	GID	Gname
1	Munna	2	2	Munni
2	Raja	1	1	Rani
3	Machha	3	3	Machha
4	NULL	NULL	4	Sundri
	Gopiya	NULL	NULL	NULL

Matched

Unmatched of Right

Unmatched of Left

ANS 1: ~~Ans 1~~ Ques 1. Select Dept Name &

Select Dept Name

From Boys & Full Outer Join Girls G

ON B.DeptNo = G.DeptNo

→ We don't have full ~~symbol~~ on Oracle.
Outer Join Syntax on Oracle.

Q1: WQTD Department name and Employee name
if department having some employee working
in it or not.

Select Dname, Ename with null

From Emp Right Outer Join Dept

ON Emp. DeptNo = Dept. DeptNo.

Q1: Both Queries emerged.

Select Dname, Ename

From Emp Full Outer Join Dept

ON Emp. DeptNo = Dept. DeptNo.

Q1: WQTD Ename and Dname if employees are
not working in any department.

Select Ename, Dname,

From Emp Left Outer Join Dept.

ON Emp. DeptNo = Dept. DeptNo.

Where Dname is NULL;

Q: Insert Ename, Dname if the dept does not have any employees working in it.

P: Select Ename, Dname
from Emp Right Outer Join Dept

ON Emp.DeptNO = DeptNO, DeptNO

Where Ename is NULL and DeptNO

where DeptNO is not found in Emp

or Merge both question

Select Ename, Dname

from Emp Full Outer Join Dept

ON Emp.DeptNO = Dept, DeptNO

Where Ename is NULL OR

Where Dname is NULL

Ans: 1. Insert Ename, Dname

from Emp Right Outer Join Dept

ON Emp.DeptNO = DeptNO, DeptNO

Where DeptNO is not found in Emp

2. Insert Ename, Dname

from Emp Left Outer Join Dept

ON Emp.DeptNO = Dept, DeptNO

Where DeptNO is not found in Emp

Self Join :-

Date 09/09/19
Page No.

Joining same table once known as self join;

or When do we go for self-joining

Whenever data present in same table's but they are present in different rows.

Symbol:-

ANSI:-

Select column-name
from table-name T,
JOIN table-name T₂
ON T₁. column-name = T₂. column-name ;

ORACLE:-

Select column-name
from table-name T₁, table-name T₂
where T₁. column-name = T₂. Column-name ;

Q:- What Employee and his manager name.

Select E₁.Ename, E₂.Ename
from Emp E₁, EmpE₂
where E₁.MGR = E₂.EmpNO ;

Result:- Emp. Details (E₁) Manager Details (E₂)

E ₁ .EmpNo.	E ₁ .Ename	E ₁ .MGR	E ₂ .EmpNo	E ₂ .Ename	E ₂ .MGR
1	A	3	3	C	10111
2	B	2	2	B	NULL
4	D	2	2	B	NULL

Q1: INFQTD empname along with their manager job of all the Emps if Emps are not working as salesmen.

- Select E1.Ename, E2.Job
from EmpE, EmpE2
where E1.MGR = E2.EmpNo.

Q2: INFQTD Emps Sal along with their manager's comm of all the Emps if Emps are working as salesman.

Select E1.Sal, E2.Comm
from EmpE, EmpE2
where E1.MGR = E2.EmpNo AND
E1.Job = 'Salesman'.

Q3: INFQTD Emp details along with manager's name, Job if Emp are not working as 'Clerk' But 'Manager' are working as 'Salesman'.

Select E1, E2.Ename, E2.Job
from EmpE, EmpE2
where E1.MGR = E2.EmpNo AND E1.Job != 'Clerk' AND E1.Job = 'Salesman';

Ename	E	S	E	S	E	S	E
EmpNo	A	B	C	D	E	F	G
1001	A	B	C	D	E	F	G
1002	A	B	C	D	E	F	G

Q1. What Emps name and their managers details if Emps are getting comm. but their managers are not getting comm.

Select E₁.Ename, E₂.*

from Emp E₁, Emp E₂

where E₁.comm like 'x' AND E₂.comm

is not null

AND E₁.MGR = E₂.EmpNo ;

Q2. What Emps details along with managers details of all the Emps if Emp have does not start with A in Dept 20 or 30.

Select E₁.* , E₂.*

from Emp E₁, Emp E₂

where E₁.MGR = E₂.EmpNo AND E₁.Ename

not like 'Ax' AND E₁.DeptNo. in (20,30);

Q3. What Emps & their manager Ename, Sal if Emps are getting sal more than 2000 but managers are ~~not~~ getting sal less than 5000.

Select E₁.Ename, E₂.Ename, E₁.Sal, E₂.Sal
from Emp E₁, Emp E₂.

where E₁.MGR = E₂.EmpNo AND

E₁.Sal > 2000 AND E₂.Sal < 5000 ;

joins

~~Self Joins~~

Q1) ~~WAPID~~ Emp. Name along with their manager's Name.

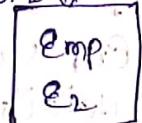
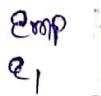
Select E₁.Ename, E₂.Ename

From Emp E₁, Emp E₂

Where E₁.MGR = E₂.EmpNo;

E₁.MGR = E₂.EmpNo

Self Joins



Emp details

Manager's details

Select E₁.Ename, E₂.Ename, D₁.Dname

From Emp E₁, Emp E₂, Dept D₁

Where E₁.MGR = E₂.EmpNo AND E₁.DNO = D₁.DNO;

ANSWER: ~~Three joins~~ ~~Three joins~~ ~~Three joins~~

Select E₁.Ename, E₂.Ename, D₁.Dname,

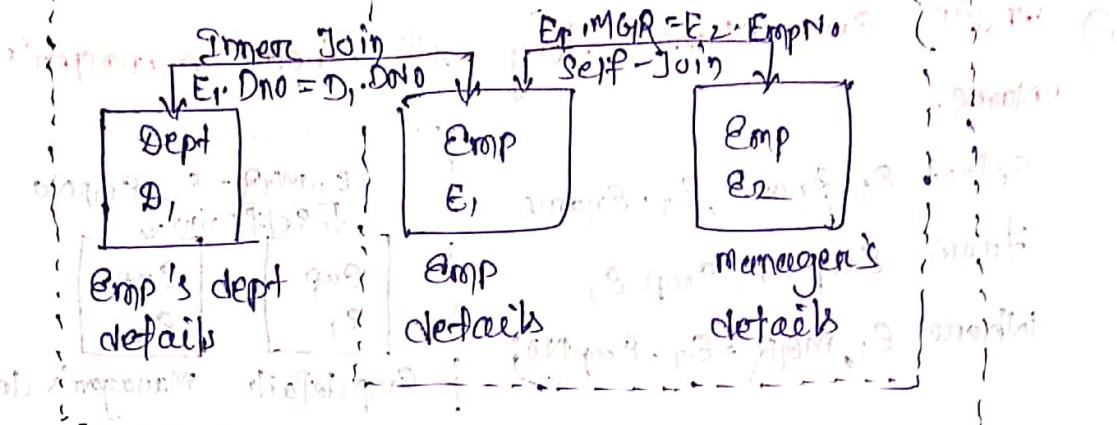
From Emp E₁ JOIN Emp E₂

ON E₁.MGR = E₂.EmpNo INNER JOIN Dept D₁

ON E₁.DNO = D₁.DNO;

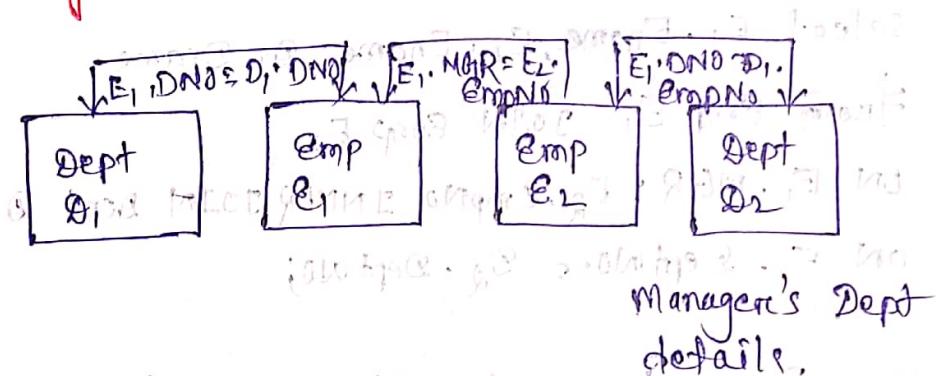
NOTE: If we have n tables then (n-1) joins conditions are there

Q1:- ~~WAPID~~ E₁.Ename, E₂.Ename
From Emp E₁, Emp E₂, Dept D₁
WAPID Emp. Name and his Manager's name
along with Emps dept name.



Select $E_1.Ename, E_2.Ename, D1.Dname$
from Emp E_1 , Emp E_2 , Dept D_1 ,
where $E_1.MGR = E_2.EmpNo$ AND $E_1.DNO = D_1.DNO$

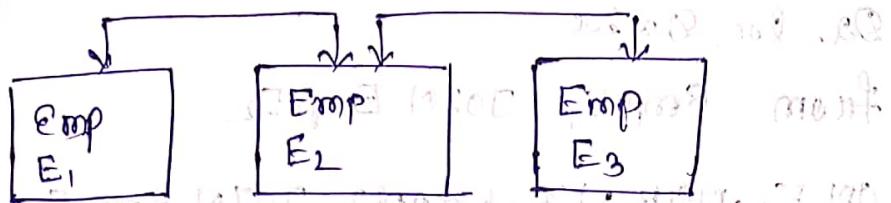
Q1: What Emp Name And his manager's name
along with their dept names.



Select $E_1.Ename, E_2.Ename, D1.Dname, D2.Dname$

from Emp E_1 JOIN Emp E_2
ON $E_1.MGR = E_2.EmpNo$ INNER JOIN Dept D_1
ON $E_2.DeptNo = D_1.DeptNo$ INNER JOIN
Dept D_2
ON $E_2.DeptNo = D_2.DeptNo$;

Q:- WANT Emp name and his manager name along with manager's manager name.



3 levels want details of E1, E2, E3
Manager's details.

~~E1.Ename, E2.Ename, D1.Dname, D2.Dname~~

from Emp E1 JOIN Emp E2

ON E1.MGR = E2.EmpNo JOIN Dept D1

ON E2.MGR = D1.DeptNo JOIN Dept D2

ON E2.DeptNo = D2.DeptNo;

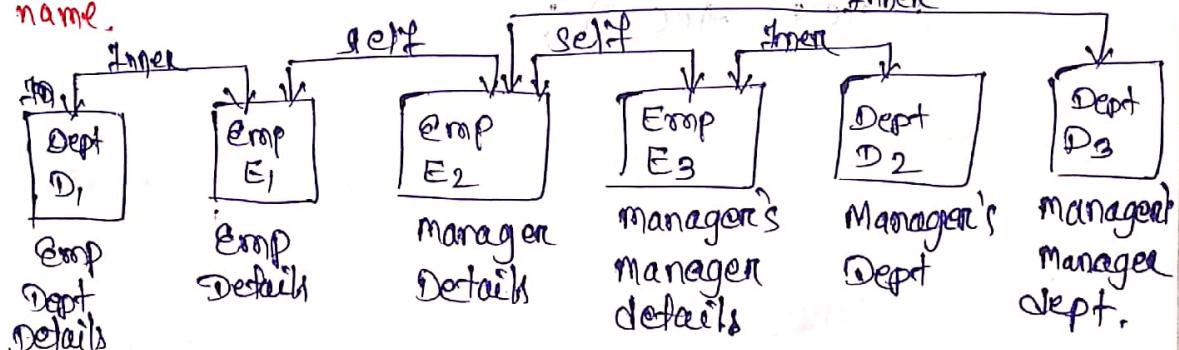
Select E1.Ename, E2.Ename, E3.Ename;

from Emp E1 JOIN Emp E2

ON E1.MGR = E2.EmpNo JOIN Emp E3

ON E2.MGR = E3.EmpNo;

Q:- WANT Emp name, and Manager name along with manager's manager name also display their location name.



Select E₁.Ename, E₂.Ename, E₃.Ename, D₁.Loc,

D₂.Loc, D₃.Loc

from Emp E₁ JOIN Emp E₂

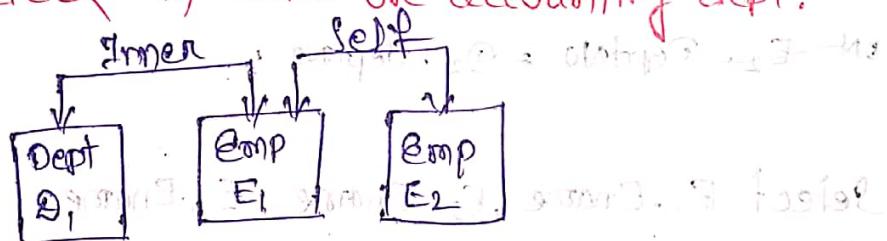
ON E₁.MGR = E₂.EmpNO JOIN Emp E₃

ON E₂.MGR = E₃.EmpNO INNER JOIN Dept D₁

ON D₁.DeptNo = E₁.DeptNo INNER JOIN Dept D₂

ON D₂.DeptNo = E₃.DeptNo;

Q:- What Emp's Name along with his
manager's Job if Emps are working
as clerk in Sales or accounting dept.

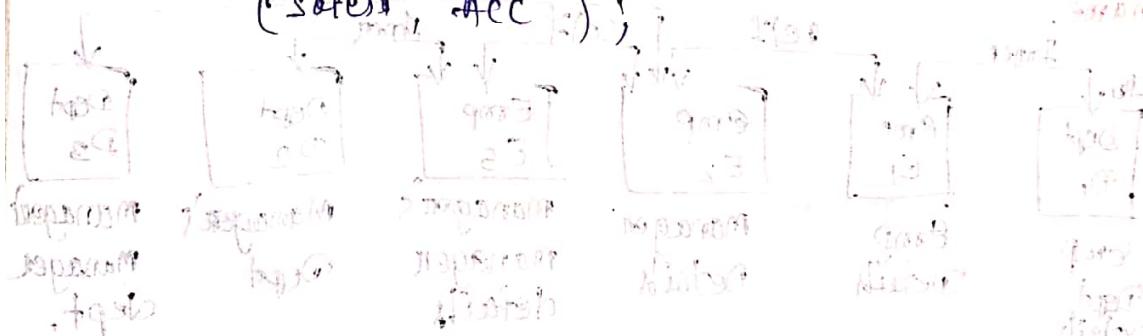


Select E₁.Dname, E₂.Job

from Emp E₁ JOIN Emp E₂

ON E₁.MGR = E₂.EmpNO INNER JOIN Dept D₁

WHERE E₂.Job IN ('clerk') AND E₁.Dname IN
(Sales, Acc);



Date / /
Page No.

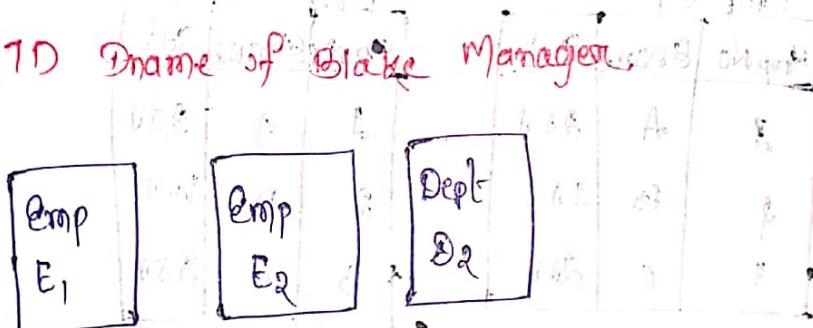
Q1:- WHAT Is Employee And his manager's loc if emps are working in dept no 20 but managers are located in Chicago.



Select E1.Ename, D2.loc
from Emp E1 JOIN Emp E2
ON E1.MGR = E2.EmpNo INNER JOIN Dept D2
ON E2.DeptNo = D2.DeptNo

Where E1.DeptNo = 20 AND D2.loc = 'Chicago';

Q2:- What Is Name of Blake Manager



Select Dname

From Dept D Where DeptNo = (Select DeptNo From Emp

Where EmpNo = (Select MGR

From Emp

Where Ename = 'Blake'));

Select D2.Dname

From Emp E1 JOIN Emp E2

ON E1.MGR = E2.EmpNo INNER JOIN Dept D2

Where E1.Ename = 'Blake';

Q1: WAP TO loc name of Emps who are reporting King.

Select loc

from dept

where deptno in (select deptno from emp

where mgr = (select empno from emp

where ename = 'King'));

Q2: WAP to Ename and Sel if Emp's are

getting same salary

Emp E ₁		
EmpNo	Ename	Sal
1	A	200
2	B	100
3	C	200

Emp E ₂		
EmpNo	Ename	Sal
1	A	200
2	D	100
3	C	200

Select E₁.Ename, E₂.Sal

from Emp₁, Emp₂ where

Emp₁.empno = Emp₂.empno AND E₁.sal = E₂.sal

E₂.empno; for check

200 = 200 ✓(T) AND 1 != 1 (F) \Rightarrow F

200 = 100 X(F) AND 1 != 2 (T) \Rightarrow F

100 = 200 (F) AND 1 != 3 (T) \Rightarrow T

so final result should be 2 rows, i.e., 1 no.

i.e., "A" - second row

① WAPTD Name and Commission if they are getting same period to same department

Select E₁. Ename, E₁. comm
from Emp E₁, Emp E₂ as from DATED

Where E₁. comm = E₂ Comm AND E₁. EmpNo !=
E₂. EmpNo ;

② WAPTD Name and MGR if they are having same reporting Manager

Select E₁. Ename, E₂. Ename
from Emp E₁, Emp E₂ as from DATED
Where E₁. Mgr = E₂ Mgr AND E₁. EmpNo !=
E₂. EmpNo ;

DATED

Q61210

CC1931013

(MADIP) 83001 10/08

and mark

CC1931063

Q612102

* Single Row Function:-

- We cannot nest multi row function but we can nest single row function.
- We can use single row functions in Select clause and in where clause.

① Upper()

* Note:- Dual

Dual is a dummy table which is used to get the output

Select UPPER('Function')

From Dual;

Output

UPPER(

DILSTAND)

② Lower()

Select LOWER('DILSTAND')

From Dual

LOWER(

DILSTAND)

- ③ INITCAP(): - Initial letter to be Capital.
- Select INITCAP('SUNDRA AND SUNDRI DASH DASH')

from DUAL

INITCAP('SUNDRA AND SUNDRI AT SunDra SunDri Dash DASH')

- ④ Length('STR'): - It is used to obtain length of given string;

Select length('GSPIDER') → 7

Select length('J@GSPIDER') → 8

Select length('J@GSPIDER') → 9

Select length('J@GSPIDER') → 9

from Dual ;

- ⑤ Reverse('STR'): - This function is used to obtain reverse the string.

Select REVERSE('DILSHAD') →

from Dual

Output

REV

DAH.SLD

DILSHAD

B

J@GSPIDER

WUDH TUDH J@GSPIDER

... RPT. NO. 3 Page 10 of 10

(6)

CONCATENATION OPERATOR (II)

Select Ename, Sal
from Emp

Select Ename || 'your Sal is' || Sal
from Emp

Where Ename = 'SMITH'

Select Ename

It is used to combine two given strings.

~~Concat()~~: If it is used to combine two given strings.

Syntax: `concat('STR1', 'STR2')`

Select concat(Ename, concat('your Sal is', Sal))
from Emp

Where Ename = 'SMITH'

~~substr()~~: This function is used to obtain sub-string from this given original string.

Syntax: `substr('original_str', position E, length)`

substr('BANGALORE', 4, 2); GIA

6, 0 NULL

7, 1 0

2, 7 ANGALORE

5, 9 ALORE NULL NULL

9, 4 E NULL NULL ..

Q:- Update salary to 10,000 if NULL
 -5,83 HALO after update

3 NIGALORE fofoz

0,2 BA gma profit

('(10,0)' instead of 10,00) L0 (because length
 cannot be negative)

Update name date 2. and 3. will be updated to

-6,0 NULL

-7,1 NULL after update

-2,7 RE fofoz

-5,9 ALORE gma profit

-9,4 BANG

-5,3 ALO

0,2 NULL

Q:- Wanted details of Employees if Employee's name
 starts with 'S'.

Select *
 From Emp

Where Ename like 'S%'

Output given in the Emp table is as follows:

Select *
 From Emp

Where SUBSTR(Ename,1,1)='S'

Q:- Wanted Job of an Employee If Job Starts
 with 'MAN'. (Job is department serial)

Select *

From Emp

Where SUBSTR(Job,1,3)='MAN';

Q) WAP to name no. of Emp if the name starts with S or M.

Select Ename
from Emp

where substr(Ename,1,1) IN('S','M');

Q) WAP to Details of Emps if the Emp name starts with S or ends with N.

Select *
from Emp

where substr(Ename,-1,1) = 'N';

Q:- WAP to Details of Emp of Emps are getting

4 digits salary.

Select *

from Emp

where length(Sal)=4

Q:- WAP to name of an Emps if the Emp having six characters in the name.

Select (Ename,6) as due_name
from Emp

where length(Ename)=6;

Q:- Name=(C, J, D, P) are the result

Q1- Write name of an employee in reverse format
and display job in lower case if employees
are getting 3 digit commission. For name
start with 'M' (MARTIN) and end with 'A'
Select Reverse (Ename), Lower (Job)
from Emp

Where length (comm) = 3 or

substr (Ename, 1, 1) = 'M'

Q2- Write first half of the name in lower
case.

Select Ename, Lower (substr (Ename, length
(Ename)/2))

From Emp;

KING $4/2 = 2$

MARTIN $6/2 = 3$

BLAKE $5/2 = 2.5 \approx 2$

Q3- Write 2nd half of the name.

substr (Ename, (length (Ename)/2 + 1))

KING $4/2 = 2 + 1 = 3$

MARTIN $6/2 = 3 + 1 = 4$

BLAKE $5/2 = 2 + 1 = 3$

Select Ename, substr (Ename, length (Ename)/2 + 1)
from Emp;

Q1- Write first half of the name in lower case but second half should be in Reverse format.

Select Replace('JSPIDER', 'J', '\$')

From Dual;

Output

QSPIDER

('JSPIDER', 'J', '\$') is used

Output

Q123 P1DER 123

(JSPIDERS, 'P', '\$') is used

Output

JSP1DERS\$

Final output

* * ('QSPIDER', 'P', '\$')

Output

Q P1DER is the final output

* * ('QSPIDER', 'P', '\$'). → Qpider

'QSPIDER', 'Y', '\$') → QSPIDER
QSPIDER → SPIDER → SPIDER
SPIDER → DIERSP → SPIDER

(QSPIDER, 'P', '\$') is used for the
final output

16/09/19 :-

Q:- WAPID Details of Employees if the employee name ends with 'S'.
Select *
from Emp

Where Substr(Ename, -1, 1) = 'S';

Q:- WAPID no. of times A occurs in your Name?
Select length('DILSHAD') - length(Replace('DILSHAD', 'A'))
from Dual;

Q:- WAPID Details of Employees if employee name having exactly one 'L'.

Select *
from Emp

Where length('Ename') - length(Replace('Ename', 'L')) = 1;

Q:- WAPID details of employees if the Employee name has exactly 2 'A'.

Select *
from Emp
Where length('Ename') - length(Replace('Ename', 'A')) = 2;

OR USING LIKE

Select *
from Emp all the employees having name ending with 'Y'
Where Ename like ('%.AY.%') AND Ename NOT LIKE ('%.AY.AY.%')

LOGIC : (A, Y, %) (A, Y, %) & (A, Y, %)

INSTR(); →

This function is used to obtain index position of the string value of the substring in the original string.

Syntax:-

$\text{INSTR}(\text{'Original_str'}, \text{'substr'}, \text{Position}[n], \text{name})$

$\text{INSTR}(\text{'PAPPA'}, \text{'A'}, 2, 3);$

A A A A A
① X X ② ③

→ 6

original string example If we apply $\text{INSTR}(\text{'PAPPA'}, \text{'A'}, 2, 3)$ then it will return 6.

$\text{INSTR}(\text{'PAPPA'}, \text{'A'}, 1, 3)$ will return 8.

$\text{INSTR}(\text{'PAPPA'}, \text{'A'}, 2, 5)$ will return 11.

$\text{INSTR}(\text{'PAPPA'}, \text{'A'}, 1, 12)$

and more examples will be explained in the next slide.

$\text{INSTR}(\text{'PAPPA'}, \text{'A'}, 8, 5)$ will return 0.

$\text{INSTR}(\text{'PAPPA'}, \text{'AM'}, 2, 1)$ will return 6.

$\text{INSTR}(\text{'PAPPA'}, \text{'AM'}, 5, 2)$ will return 8.

$\text{INSTR}(\text{'PAPPA'}, \text{'MM'}, 6, 3)$ will return 0.

$\text{INSTR}(\text{'PAPPA'}, \text{'MM'}, 6, 12)$ will return 10.

$\text{INSTR}(\text{'PAPPA'}, \text{'A'}, 1, 1)$ will return 0.

$\text{INSTR}(\text{'PAPPA'}, \text{'A'}, 4, 1)$ will return 5.

Q:- WAQID details of Employees if the name having atleast 1 letter 'A'. (using INSTR function) write query

Select *

From Emp

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

Q:- WAPTD details of employee if the name having atleast 2 'A'.

Select *
from Emp

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

Q:- WAPTD Name, Job of all the employees if the name having atleast 2's L.

Select Ename, Job
from Emp

where INSTR('Ename', 'L', 1, 2) > 0 ;

Q:- WAPTD details of Employees if the Empname has atleast 3 A's.

Select *
from Emp

where INSTR('Ename', 'A', 1, 3) > 0 ;

Q:- WAPTD name of an Emp if the name has exactly 1 A.

Select Ename
from Emp

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

INSTR('Ename', 'A', 1, 2) = 0

INSTR('Ename', 'A', 1, 1) > 0 AND INSTR('Ename', 'A', 1, 2) = 0

BLAKE 3 > 0 \boxed{T} AND $0 = 0 \rightarrow T$

ADAMS 1 > 0 \boxed{T} AND $3 = 0 \rightarrow F$
 0 > 0 \boxed{F} AND $0 = 0 \rightarrow F$

And other two $\boxed{2 > 0 \rightarrow T}$ AND $3 = 0 \rightarrow F$

('blake') student of dept

~~↳ Date and time objects~~ will be discussed in detail in DBMS

SYSDATE() :-

It is used to obtain current date from the system where RDBMS is installed.

Select SYSDATE

Output :-
16-SEP-19

SYSDATE O/P

16-SEP-19

16-SEP-19

Select Current_Date.

Output :-
16-SEP-19

Current_D

16-SEP-19

16-SEP-19

SYSTIMESTAMP() :-

It is used to obtain date, time along with timezone.

Select SYSTIMESTAMP

16-SEP-19

Output :-
16-SEP-19 11.49.49.248500 AM +05:30

↳ (Q1, A). SYSTIMESTAMP :- (Q1, A). SYSTIMESTAMP :-

16-SEP-19 11.49.49.248500 AM +05:30

T0_DATE() :-

↳ Convert the given string format into date format.

Syntax :- T0.Date('Date');

Ex:- select systdate - '15-SEP-19'
from dual;

Result :- Error :- Invalid Date and ID
invalid Number

Correct format

in format

Select systdate - To_date ('15-SEP-19')

from dual;

Result :- Output will be same as above

1.4967 --

Conversion process goes

To_CHAR :-

(To convert) the given date into character format.

:new_to_char('Date', 'Format-Model');

IS NULL

FORMAT MODEL

YEAR	MONTH	DAY	HH24
YYYY	MON	DY	HH12
YY	MM	DD	MI
		D	SS

YEAR → Twenty Nineteen

YYYY → 2019

YY → 19

MONTH → SEPTEMBER

MON → SEP

MM → 09

DAY → TUESDAY

DY → 17

D → 3 → 3rd day is Tuesday that why 17

HH24 → 10 (In 24 hour format)

HH12 → 10 (In 12 hour format)

MI → 53 (min)

SS → 35 (sec)

17/09/19

PT-922-81 - chapter 6 - 179

1. find month

Q:- WAP TO details of employees if they were hired in the month of Dec.

format format

Select *

from Emp
where To_char(Hiredate, 'MON') = 'DEC';

Q:- WAP TO Name, Hiredate of an employees who hired in the month of Nov or July.

format format

Select ename, hiredate

from Emp

where To_char(Hiredate, 'MON') in ('NOV', 'JUL');

Q:- WAP TO details of employees who were hired during the year 81.

format format

Select *

from Emp

where To_char(Hiredate, 'YY') = '81';

Q:- WAP TO Name, Job, Hiredate of all the employees if employee name starts with vowel or if they were hired on 17th, 18th or 20th day.

Select Enamel, Job, hiredate

from Emp

where To_char(Hiredate, 'DD') in (17, 18, 20) OR
substr(ename, 1, 1) in ('A', 'E', 'I', 'O', 'U');

Q:- WAP TO Hiredate in USA format.

Select To_char(Hiredate, 'MM-DD-YYYY')

from Emp;

(019) 88 ← EM

(292) 88 ← 82

Q1: WAP ID Name of your 1st character should be replaced by \$.

Ans: Select \$ || Substr('Ename', 2) instead of \$

Q2: WAP ID last-character of your name replaced by \$.

Ans: Truncate name of read records along

NVL function:

NVL → NULL value logic.

This function is used to overcome by the problem of null.

Arg 1 → In argument 1, we have to write the column-name which can be NULL or not.

Arg 2 → we have to write the value, when argument 1 is null.

Example: If we want to add salary + commission to all the records.

ename	Sal	Comm
A	200	20
B	500	NULL
C	400	30

Arg 1 → we have to write the

value, when argument 1 is null.

NVL(comm, 0):

Sal + NVL(comm, 0)

200 + NVL(20, 0)

500 + NVL(500, 0)

400 + NVL(30, 0) = 430

30

Sal + comm formula:

$$200 + 20 = 220$$

$$500 + \text{NULL} = \text{NULL}$$

$$400 + 30 = 430$$

And answer is as follows:

Ans for query will be referencing like follow:

: know for question no problem

Ans for question no problem

Ans for question no problem

know

Q1209

PREPARATION FOR

PSEUDO COLUMNS:

Pseudo columns are the table column which will be present in each and every Comp table which are present in database.

Pseudo columns have to call 'Explicitly'

There are 2 pseudo Column

1 = RowId

2 = RowNum

Ex. RowId:

It is a Pseudo Column which will be present in all the table's & which is given

to all the records.

- RowId is used to quickly access or delete a record from the table.

- RowId is unique And Not nullable

- We can use RowId as Primary key when the table doesn't have Primary key but we cannot assign RowId as Primary key.

- RowId will generates at the time of insert or creation of record.

- RowId can not be delete.

- RowId is a 18-digit address given to the record.

ROWID

→ AAMPPAAEAAAGAA

18/09/19

ROWNUM:-

Rownum is a pseudo column which will be present in all records which are present in database.

- Rownum acts as serial number in result table.
- Rownum always begins with '1'.
- Rownum will get generated at the time execution.
- Rownum is dynamic in nature.

→ Select Rownum, RowID, Emp.*
 from Emp
 where Sal > 2000;

Rownum	RowID	Empno	Empname
A	1	101	John
F	2	102	Paul
G	3	103	George
H	4	104	Ringo
I	5	105	Peter

To quickly access or to uniquely identify the record without primary key.

→ Select RowID, Empno, [] from Emp
 where RowID is 'AAAMFAAAAGAAAAA' ;

1. Inserting	→ 31				
2. Inserting	→ 32				
3. Inserting	→ 33				

1) WAP TO top 3 records from the table.

Select *
from Emp

Order By Rownum < 4 ;

2) WAP TO top 5 records from the table.

Select *
from Emp

Order By Rownum < 6 ;

3) WAP TO first record from the table.

Select *
from Emp

Where Rownum = 1 ;

4) WAP TO 3rd Record from the table.

Select *
from Emp

Where Rownum = 3 ;

Assigning of Rownum.

Always assign the Rownum when condition is true.

Rownum	ename	sal	eno
1 < 3 ✓	KRISHNA	30000	101
2 < 3 ✓	MURTHY	35000	102
3 < 3 ✗	ANU	25000	103

Select *
from Emp
Where Rownum < 3

Rownum = 1 ;
Rownum = 2 ;
Rownum = 3 ;

5) want 5th record from the table.

Select *
 from (Select Rownum "Eno", Emp.*
 from Emp)

where Eno = 5 ;

6) want 5th record and 10th record.

Select *

from (Select Rownum, 'eno', Emp.*
 from Emp)

where 'eno' in (5,10);

Select *
 from (Select Rownum "SLNO", Emp.*
 from Emp)

where SLNO = 3 ;

Ename	Sal	Dno
'A'	200	20
'B'	500	20
'C'	600	30
'D'	300	20

SLNO	Ename	Sal	Dno.
1	'A'	200	20
2	'B'	500	10
3	'C'	600	30
4	'D'	300	20

O/P result:-

SLNO	Ename	Sal	Dno
3	'C'	600	30

7) WAP TO bottom 3 records from table.

Select *
from (Select Rownum "SLNO", Emp.*
from Emp)

where SLNO > (Select Count(*) - 3
from Emp));

8) WAP TO bottom 5 records from table.

Select *
from (Select Rownum "SLNO", Emp.*
from Emp)

where SLNO > (Select Count(*) - 5
from Emp));

9) WAP TO 1st record from table.

Select *

from (Select Rownum "SLNO", Emp.*
from Emp)

Where SLNO = 1;

To find nth maximum or nth minimum So,

Display the result

Assign the Rownum

Order the Record

→ To find 4th max salary.

Empno	Sal
A	800
B	400
C	200
D	400
E	900

Sal
900
800
400
200

Select Sal
from (Select Rownum "SI", Sal
from (Select Distinct Sal
from Emp))
Order by Sal DESC
where SI = 4;

SI	SAL
1	900
2	800
3	400
4	200

O/P Select

Sal
800

10) In Q10) 5th minimum salary.

Select Sal
from (Select Rownum "SLNO", Sal
from (Select Distinct Sal
from Emp))
Order by Sal
where SLNO = 5;

11) 1st max, 3rd max, 5th max, 8th max, 10th max Salary.

Select Sal

from (Select Rownum "SLNO", Sal

from (Select Distinct Sal)

from Emp

Order by Sal Desc))

Where SLNO IN (1, 3, 5, 8, 10);

102	102	20000
668	668	0
567	567	0
432	432	0
339	339	0

12) WAP/TD details of Emp 10th min salary.

Select all

from Emp

Where Sal in (Select Sal

from (Select Rownum "SLNO", Sal

from (Select Distinct Sal))

from Emp

Order by Sal));

Where SLNO = 10);

H/W Assignment:

1) WAP/TD top 5 max salaries.

Select Sal, Min, Max from Emp;

from (Select Rownum "SLNO", Sal

from (Select Distinct Sal))

from Emp

Order by Sal Desc));

Where SLNO < 6;

{ a simple query}

2) WAP TO top 5 min salaries.

Select Sal
from (Select Rownum SLNO, Sal
from (Select DISTINCT Sal
from Emp
Order by Sal))
where SLNO < 6;

3) WAP TO bottom 5 max salaries.

Select Sal
from (Select Rownum SLNO, Sal
from (Select DISTINCT Sal
from (Select * from Emp
Order by Sal Desc)))
where SLNO > (Select count(*) - 5
from Emp);

4) WAP TO bottom 5 min salaries.

Select Sal
(Select Rownum SLNO, Sal
from (Select DISTINCT Sal
from (Select * from Emp
Order by Sal)))
where SLNO > (Select count(*) - 5
from Emp);

19/09/19

Unit 1: Introduction to Database Management System

* SQL statements OR SQL Languages : SQL Standard.

1) DDL :-

a) Create :- This statement is used to Create an Object in the database.

Eg:- We can Create Table, Views, Procedures, Triggers etc.

CREATE :-

Syntax :-

```
CREATE TABLE table-name  
(  
    Column-name1 Datatype NOTNULL/NULL,  
    Column-name2 Datatype NOTNULL/NULL,  
    ...  
    Column-name n Datatype NOTNULL/NULL,  
  
    constraint Constraint-ref-name Unique(Column-name),  
    constraint Constraint-ref-name Check(Condition),  
    constraint Constraint-ref-name Primary Key(Column-name),  
    constraint Constraint-ref-name Foreign key(Column-name)  
    Reference parent-table-name (Column-name)  
);
```

Table Name = Product

No. of column = 3

Product

Column-Name →	PK PID	Pname	Cost
Data-type →	Num(2);	Varchar(25);	Num(2);
Null/NotNull →	NN	NN	NULL
Constraints →	PK PID = PK		check(cost > 100); cost < 100;

Create table Product

Creating table Product with constraints:

PK PID Number(2) NOT NULL,
 Pname Varchar(25) NOT NULL,
 cost Number(9,2) NULL,

Constraint PID-PK Primary key (PID);
 Constraint COST-CH Check (cost > 100)

;

Customer

Column-Name →	CID	Cname	Phno.	PK FK
Data-type →	Num(2);	Varchar(30);	Num(10);	Num(2);
Null/NotNull →	M-N	~	~	~
Constraints →	PK CID-PK	Unique Pname	Unique length(phno) = 10	PK P11 check P20 FK

Create Table Product (statement for add)

C

PID Number(2)

buffer?

Product ID	Product Name	Category	Supplier ID
1001	Laptop	Electronics	Supplier A
1002	Monitor	Electronics	Supplier B
1003	Mouse	Electronics	Supplier C
1004	Keyboard	Electronics	Supplier D
1005	Headphones	Electronics	Supplier E

b) RENAME :-

This statement is used to rename the existing table name which is present in the database.

Syntax:-

```
RENAME Current-table-name To New-Name;  
RENAME Procedure To Proc;
```

c) ALTER :-

This statement is used to modify or alter the table structure (change column name, add column, delete column, etc).

1. To Add a Column:

```
ALTER TABLE table-name ADD column-name datatype [NULL/NOTNULL];
```

Eg:- ALTER TABLE Customer

```
ADD P1D Number(2);
```

2. To Drop a Column:

After table table-name
~~Drop column Column-name;~~

Eg:- Alter table Customer
~~Drop column PhoneNo;~~

Drop column PhoneNo;
~~;(because phone no is primary key)~~

Select Constraints_Name, Table_Name.

From User_Constraints;

3. To change the Datatypes:

Alter table table-name.

Modify Column-name newdatatype;

Eg:- Alter table Customer

Modify Ename Char(40);

4. To change the Not Null Constraint:

Alter table table-name;

Modify Column-name existingdatatype

NULL / NOTNULL;

Eg:- Alter table Customer

Modify Ename Char(40) NOTNULL;

5. To Rename the Column:

Alter table table-name

(Rename Column Current_name TO NewName);

Eg:- Alter table Customer

Rename Column Ename TO name;

{from note page 2}

6. To create Foreign key Constraints:

After table table-name
Add constraint constraint-ref-name foreign key
references (column-name);

① References parent-table-name (column-name);

Eg:- Alter table Customer

Add constraints PID_FK foreign key (PID)

References Prod(PID);

② DML (Data manipulation language)

(a) Insert

(b) Update

(c) Delete

① INSERT:

To insert the record into the table.

Syntax:-

① Insert INTO table-name Values (v_1, v_2, \dots, v_n);

② Insert INTO table-name (col1, col2, ..., coln)
values (v_1, v_2, \dots, v_n);

OR

Insert INTO table-name (col1, col2, coln)

values (& col1&col2&coln);

③ Insert into table-name
Select Statement;

Eg. of 1 → there we should know all the column name accordingly.

Eg:- Insert into Prod values (1, 'Lipstick', 3500);

Eg of 2 type → Need to write insert into many times; insert into Prod (PID, Pname, cost) values (1, 'Oldmark', 457); with function

Eg of 3 type →

b) Insert into Prod values (&PID, &Pname, &cost);
Enter values for PID : 3
Enter values for Pname : 'watch'
Enter values for Cost : 2999.9

b) UPDATE:

Used to modify the data which are present in the database:

Syntax :- Update table-name

SET (Col1 = V1, Col2 = V2, ..., Coln = Vn)
[Where <filter-condition>];

Eg:- Update Prod
SET cost = 350
Where Pname = 'LIPSTICK';

c) DELETE:
To delete a particular record.

Syntax:- Delete from table-name

[Where <filter-condition>];

(Result) OR: No Record Found after executing

Eg:- Delete
from Prod

Classifying Where Primary = 'Lipstick'; and 'P'

TRUNCATE:
It is used to remove all the records from the table.

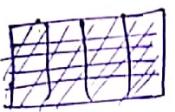
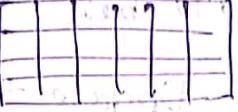
It is used to remove all the records permanently without altering table structure.

DROP:-

It is used to remove all the records along with table structure.

DELETE:-

To delete a particular records.

Truncate Table	Drop Table	Delete
		
Can't get it back	Yes	Yes
	Flash Back	Roll back

Truncate Syntax:- Truncate Table table-name;

Drop Syntax:- Drop Table table-name;

(a) To recover the table : (only in Oracle)

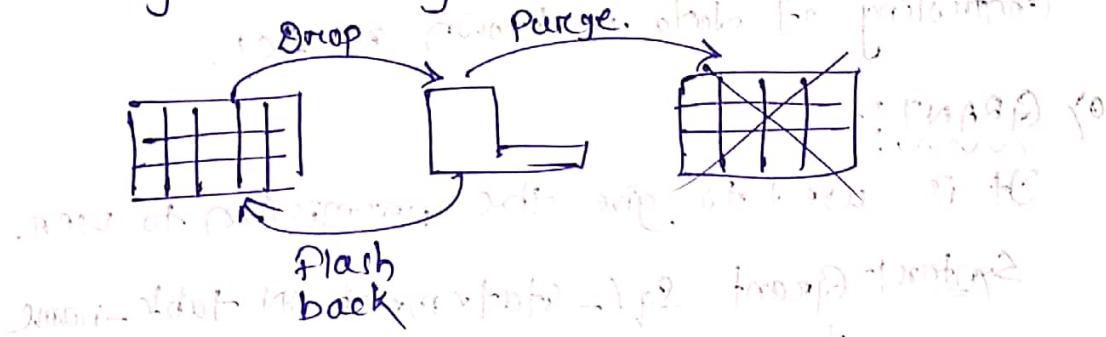
Syntax:- Flashback Table table-name

To before Drop

[Rename To new-name]

(b) To Drop the table from Recycle Bin.

Syntax :- Purge Table table-name;



3. TCL (Transaction Control Language)

[DDL statement is also called as Auto-Commit statement.]

a) Commit :- Commit is used to save all the transaction in database.

Syntax :- Commit;

b) Save point :- It will save the data in to table but not on the database.

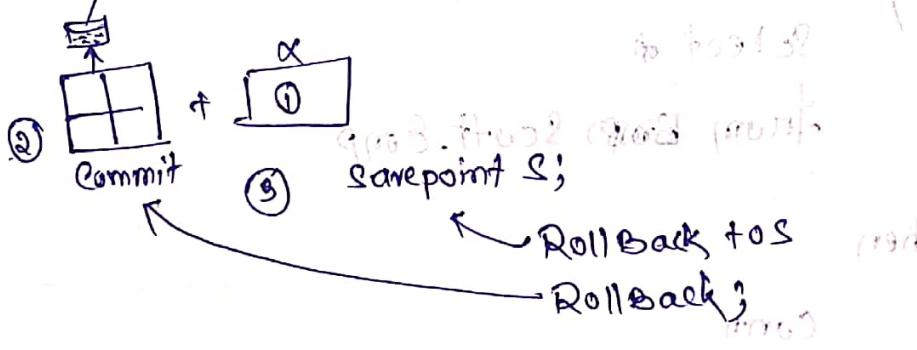
Syntax :- Savepoint savepoint-name;

c) Rollback :- Go back to the previously committed.

Syntax :- Rollback;

Rollback to save point;

Syntax :- Rollback to save point-name;



4. DCL (Data Control Language)

Controlling of data is known as DCL.

a) GRANT :-

It is used to give the permission to User.

Syntax :- Grant sql-statement ON table-name
from-name;

Grant select on Emp to hr from db
to HR

Output :-
Grant Succeeded. Pendefok on permissions.
SQL > conn

Start on User-name : HR
password : xxxx

then select * from Emp

Select *
from Emp → not work becuase it will

go to HR database where

Emp is not available.

So,

select *

from scott.Emp

then

conn

User name :- Scott

password :- Tiger

Revoke Select on Emp

GRANT SELECT ON Emp TO SCOTT;

Again,

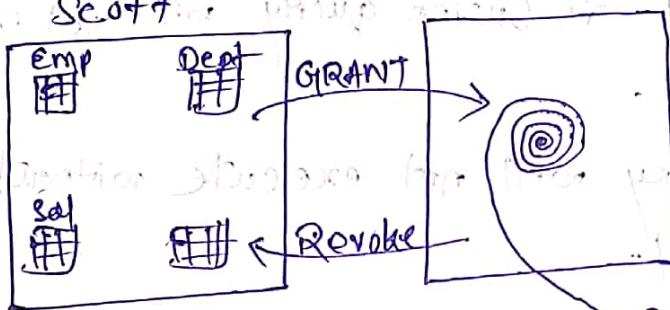
SELECT * FROM SCOTT.EMP;

from Scott Emp

gives [Error] because Grant is taken

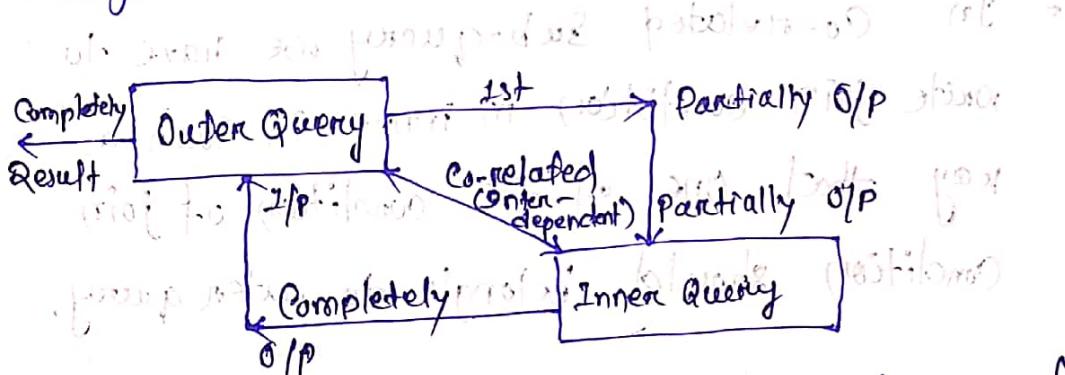
which was by SCOTT himself. It would work if

SCOTT. SELECT * FROM SCOTT.EMP;



→ will give the following error message from SCOTT.EMP:

* CO-RELATED Sub-query:
Inner query is dependent on Outer query and
Outer query is dependent on inner query is known
as Co-related Sub-query.
Working Principle:



→ Outer query will execute first and generates partial output.

→ Partial Output is given to the Inner query.

→ Inner query will execute completely.

- Output of inner query given to the Outer query.
- Outer query takes input from Inner query and generates completely.
- Outer Query is dependent on Inner Query.
- Inner Query is dependent upon Outer query.
- ~~Inner~~ Outer query will not execute inner query.
- Inner query will not execute without outer query.
- That's Why they are co-related or Inner dependent.

NOTES:-

- Co-related Sub-query works on the principle of both Sub-query and join.
- In Co-related Sub-query we have to write join condition in inner in such a way that one of the condition of join condition should belong's to outer query.

Sharing few benefits between the joins which are useful to know.

SUB. QUERY

Run independently of main query.

- Inner query will executes first.

- Outer query will executes first.

- Inner query is not dependent on Outer query. Outer query depends on inner query.

- Outer query not part of inner query. executes parallelly

- Outer query executes parallelly, full table

- Inner don't takes partial input from Outer Query.

- Inner query takes partial input from Outer query.

- Sub-query is not working on principle of both joins and sub-query.

- Correlated works on principle of both joins and Sub-query.

Q:- In Q71) Dept name is which same Emps are working.

Select Dname

From dept

Where deptNo in (Select deptNo

From Emp

Where deptNo is not null);

Q1:- WAQT D Dept name in which no employees are working.

From Dept
group by DeptNo
having count(*) = 0

From Emp
group by DeptNo
having count(*) > 0

Group by DeptNo
having count(*) = 0

By Using Joins:

1) Select DeptName
From Emp E Inner Join Dept D
ON E.deptNo = D.deptNo
Group by DeptName

Q1:- WAQT D Dept name in which Employees are working by using Correlated.

From Emp E
group by DeptNo
having count(*) > 0

From Dept D
group by DeptName
having count(*) = 0

Correlation in Output

Correlated Sub-query

<u>Emp E</u>			<u>Dept D</u>		
Ename	Sal	Dno	Dname	Dno	
'A'	200	20	'D ₂ '	10	
'B'	600	10	'D ₂ '	20	
'C'	400	20	'D ₃ '	30	

Select D. Dname

from Dept D

where D. Dno IN (Select E. Dno

from Emp E

where D. Dno = E. Dno);

1st :- 10 --- → 10
 $10 = 20 \times$

D. Dno IN (10)

$10 = 10 \checkmark$

2nd point

2nd :- 20 --- → 20

D. Dno IN (20, 20)

$20 = 20 \checkmark$

$20 = 20 \times$

$20 = 20 \checkmark$

3rd :- 30 --- → 30

D. Dno IN (NULL)

$30 = 20 \times$

$30 = NULL \rightarrow NULL$

$30 = 10 \times$

$30 = 20 \times$

EXISTS !-

It is a Unary Operator which returns true when sub-query return some value.

NOT EXISTS:-

It is an Uniquary Operator, which returns true when Sub-query not return any values.

Select D.Dname
from Dept D
where NOT EXISTS (Select E.DeptNo
from Emp E
where D.DeptNo = E.DeptNo);

Also

Using IN operator

Select D.Dname,

from Dept D

where NOT EXISTS (Select E.Dno.

from Emp E

where D.Dno = E.Dno);

Q:- WATD :- Department which does not have any employee's working in it.

Select D.Dname
from Dept D
where NOT EXISTS (Select E.Dno
from Emp E
where D.Dno = E.Dno);

With another pointer note :- where (D.Dno = E.Dno);

- More efficient than join bcoz

To find nth max / nth min salary.

Ex) Max = 700

Ans. 3. 700

E ₁	Sal	E ₂	Sal
800		800	
900		900	
700		700	
800		800	
600		600	

Select E₁.Sal

from Emp E₁

where (Select Count(Distinct E₂.Sal))

from Emp E₂

where E₁.Sal < E₂.Sal) = 2;

700 < 800 900 < 800 X 800 < 800 X 1st → X
 < 900 ✓ < 900 X 900 < 900 ✓ 0th → X
 < 700 X < 700 X 700 < 700 X 1st → 0
 < 800 ✓ < 800 X 800 < 800 X 2nd → 1
 < 600 X < 600 X 600 < 600 X 3rd → 2
 < 500 X < 500 X 500 < 500 X 4th → 3.

[nth → n-1]

max → E₁.Sal < E₂.Sal

min → E₁.Sal > E₂.Sal

⇒ 10th → 8th min salary.

Select E₁.Sal

from Emp E₂

where (Select Count(Distinct E₂.Sal))

from Emp E₂

where (E₁.Sal) > E₂.Sal) = 4;

(to filter out from E₂)

2) WAPID 10th Max Salary.

Select E1, Sal

From Emp E1

Where q = (Select Count(Distinct E2, Sal))

From Emp E2

Where E1.Sal < (Eq. Sal);
Order

3) WAPID Details of Employees who are getting
1st Max, 3rd max salary.

Select E1, #

From Emp E1

Where (Select Count(Distinct E2, Sal))

From Emp E2

Where E1.Sal < (Eq. Sal) IN (0,2,4);
Order

4) WAPID Top 5 max salaries.

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);
(102,03 from 0 to 102)

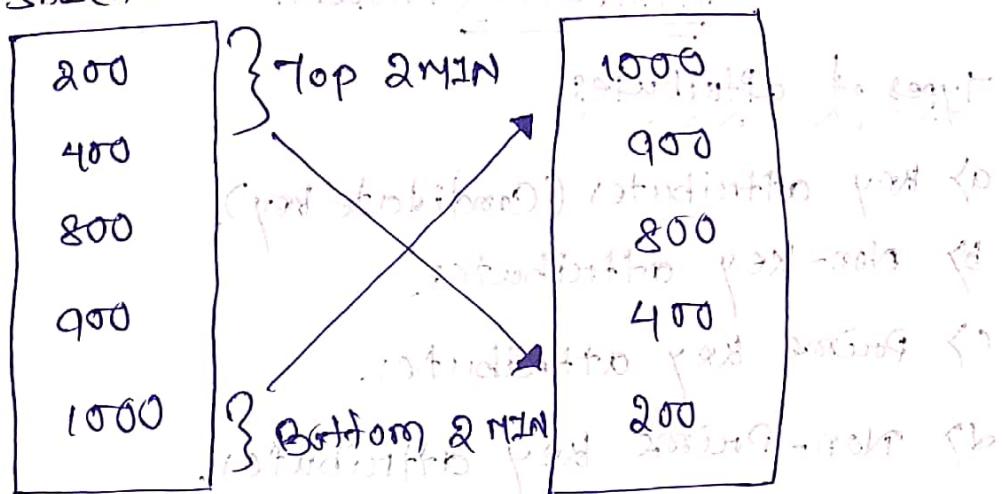
5) WAPID Bottom 5 max salaries.

IP: (102,03)(102,13) write

Q:

Convert in Min(Sal)

SAL(CSC) Contradiction problem



{ Top nth max = Bottom nth min }
Bottom nth max = Top nth min }

Select E1. Sal

```
from Emp E1
where (Select Count(Distinct E2.Sal)) IN(0,1,2,3,4);
from Emp E2
where E1.Sal > E2.Sal) IN(0,1,2,3,4);
```

6) Find Q1D & Bottom 3 min salaries.

Select E1. Sal

```
from Emp E1
where (Select Count(Distinct E2.Sal)) IN(0,1,2,3,4);
from Emp E2
where E1.Sal < E2.Sal) IN(0,1,2,3,4);
,2 query (order by , ascending after second row)
,3 query (order by , descending after second row)
```

no contradiction no contradiction (H)

Statistics can be inserted in contradiction
at most 2 rows or 3 rows or 4 rows
contradiction will occur in range of 5 rows

Attributes (Properties)

Types of attributes

- a) key attributes (Candidate key).
- b) Non-key attributes.
- c) Prime key attributes.
- d) Non-Prime key attributes.
- e) Composite key attributes.
- f) Super key attributes.
- g) Foreign key attributes.

(a) key attributes (Candidate key):

It is used to uniquely identify a record from the table.

Ex:- Aadhar, Pan Card No, Phone, DL, Passport

(b) Non-key attributes:

It is used to uniquely identify a record.

All the attributes except key attributes are known as non-key attributes.

Ex:- Name, Age, Gender, Color, Blood groups.

(c) Prime-key attributes:-

all the attributes an attributes an attribute is selected as main attribute to uniquely identify a record from the table is known as prime key attribute.

Eg:- Phone Number is primary key

d) Non-Prime key attribute:-

Among all the key attributes except prime key attributes are known as non-Prime key attributes.

Eg:- Password.

e) Composite key attribute:-

Combination of two or more non key attributes are known as composite key attributes.

Eg:- (Name, Age, Blood group, Gender);

f) Super key attributes:-

Set of all key attributes are known as Super key attributes.

g) Foreign key attribute:-

To establish the connection between two tables, are known as foreign key attributes.

Eg:- CID, Deptno etc. for establishing relationship between two tables.

Establishes relationship between two tables.

Establishes relationship between two tables.



Establishes relationship between two tables.

* DEPENDENCY OR FUNCTIONAL DEPENDENCY :-

Consider a relation R having two attributes A and B in which an attribute B is dependent on another attribute A is known as dependency.

Or

functional dependency.

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

$A \rightarrow B$

A determines B

Or

B is dependent on A

Types of dependency:-

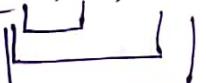
1) Total functional dependency (Full)

2) Partial functional dependency.

3) Transitive functional dependency.

4) Total functional dependency:-

All the attributes dependent on key attributes is known as total functional dependency.

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


$$A \rightarrow B$$

$$A \rightarrow C$$

$$A \rightarrow D$$

\therefore Total functional dependency

2) Partial functional dependency:-

An attribute is dependent on another attribute which is a part of composite key attributes.

Eg:- Consider a relation $R \rightarrow \{A, B, C, D\}$

$$B \rightarrow C$$

$$B \rightarrow D$$

partial dependency of $A \rightarrow D$ is not required

3) Transitive functional dependency:-

An attribute is dependent on non-key attributes which is dependent on key attributes is known as transitive functional dependency.

transitive dependency is caused by

Eg:- $R : - \{A, B, C, D\}$

$$A \rightarrow B$$

$$A \rightarrow D$$

$$D \rightarrow C$$

$$A \rightarrow C$$

Redundancy :- Repetition's of unmatched data.

(repetition).

Anomaly :- Anomalies are the side effects which

are caused due to DML operation.

	1FD	PFD	1FD
R	x	✓	✓
A	x	✓	✓

Normalisation

Normalisation & Normal Forms

* NORMALISATION: It is a process of reducing a larger table in

to smaller table by identifying dependencies to remove redundancies & anomalies.

Normalisation is a process of decomposing larger table into normal form.

* Normal Form: A table which is free from redundancy &

The table without redundancies and anomalies is known as Normal form.

* level's of normal form:

1. First Normal Form (1NF)

2. Second NF (2NF)

3. Third NF (3NF)

4. Boyce-Codd NF (BCNF)

After reading all the following definitions,

Normalise the given below table

1	2	3	4	5
1	2	3	4	5
1	2	3	4	5