

# SQL Statements or SQL Languages:

## Data Definition Language: (DDL)

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

Ex:- We can create table,

We can create procedure

We can create views

Syntax : Create CREATE TABLE table-name

{

COLUMN\_NAME1 DATATYPE NOT NULL / NULL.

2

COLUMN\_NAME2 DATATYPE NOTNULL / NULL.

CONSTRAINT constraint-ref-name UNIQUE (COLUMN-NAME),  
CHECK (CONDITION),

PRIMARY KEY (COLUMN NAME),

-FOREIGN KEY (COLUMN-NAME)

REFERENCES parent\_table\_name (COLUMN-NAME).

## Product

Column-name	PID	PName	Price
Data type	Number(2)	varchar(26)	Number(10,2)
NOTNULL /NULL	N.N	N.N	NULL
constraints	PK PID_PK		Check (price > 100); PRICE-CH

CREATE TABLE PRODUCT

```

PID NUMBER(2) NOTNULL,
PNAME varchar(26) NOTNULL,
PRICE Number (10,2) 1
CONSTRAINT PID_PK PRIMARY KEY (PID),
CONSTRAINT PRICE_CH CHECK (price > 100)

```

## Customer

Column-name

Data-type

NOT NULL / NULL

constraints

CID	Cust-Name	mobileno	PK PID
Number (5)	Varchar (25)	Number (10)	
N. N	N. N	N	
PK CID - PK			Check length(phone-no)=10; UNIQUE ↓ ph.no ↓ ph.no.ch constraint ref name

Con. ref name

CREATE TABLE CUSTOMER

(

CID Number (5) , NOTNULL

cust-name ~~Varchar~~ Varchar (25) , N,

mobileno. Number (10) , N ,

CONSTRAINT CID - PK PRIMARY KEY (CID),

CONSTRAINT ~~PHI~~ CHECK PHI UNIQUE (phone-no),

CONSTRAINT PHI - CH CHECK Length (phone-no) = 10; )

DESC Customer;

Name

~~PK~~ CID

C NAME

ph.no

NULL?

NOTNULL

TYPE

Number (5)

Varchar (25)

Number (10)

Select constraint-name, TABLE-name  
from user\_constraints

where TABLE-name = 'customer'

User-constraint : It is a f.b name where in all the  
constraints will be present.

Family , Student table.  
      |  
      S      C

Column-Name

RENAME :

This Statement is used to RENAME the Existing object  
name.

Syntax : RENAME current-table-name TO New Name

Ex : RENAME PRODUCT TO PRD

## ALTER :

It is used to alter, modify the ~~structure~~ structure of an existing object.

Ex : We can add column-name

We can change colname, Datatype, ~~NN~~ to N, Constraints.

We can Rename the column-name, tablename

## Syntax :

1. TO ADD a column :

ALTER Table table-name

ADD column-name Datatype (NULL / NOTNULL)

Ex : ALTER Table CUSTOM

ADD PID Number (2) NULL

2. TO DROP a col

ALTER Table table-name

DROP column column-name.

Ex:-

ALTER Table CUSTOM  
DROP column ph-no;

3. To change the datatype

ALTER table table-name

MODIFY column-name new-datatype;

Ex:-

ALTER table custname

MODIFY cust-name char(20);

4. To change the NOTNULL constraint:

ALTER TABLE table-name

RENAME column current-name TO new-name;

MODIFY col-name existing-datatype ~~NOTNULL~~

Ex:-

ALTER TABLE CUSTOM

MODIFY cname char(20) NULL;

5. To Rename the column

ALTER Table table-name

RENAME column current-name TO newname;

Ex:- ALTER Table CUSTOM

~~modify~~ ~~cname~~ column 'cname' TO 'Cust-name';  
~~Rename~~

## How to Add Foreign Key constraint.

ALTER TABLE table-name

~~ADD~~ ~

ADD Constraint constraint-def-name Foreignkey (col-name)

References Ref-t-name (col-n);

ALTER TABLE CUSTOM

ADD constraint PID-FK ForeignKey (PID)

References PRO(PID)

## Data Manipulation Language(DML)

Insert: This statement is used to Insert the records

into the table.

Syntax : \* Insert into table-name values (v<sub>1</sub>, v<sub>2</sub> .. . v<sub>n</sub>)  
\* Insert into table-name (col<sub>1</sub>, col<sub>2</sub>, .. . col<sub>n</sub>)  
values (v<sub>1</sub>, v<sub>2</sub> .. . v<sub>n</sub>),  
(v<sub>1</sub>)

Insert into table-name (col<sub>1</sub>, col<sub>2</sub>, .. . col<sub>n</sub>) values  
(col<sub>1</sub>, & col<sub>2</sub>, .. . col<sub>n</sub>)

\* Insert into table-name  
select statement;

Insert into PRO Values (1, 'lipstick', 360)

Insert into PRO (PID, PNAME, PRICE) Values (2, 'old monk', 106)

Enter value for pid: 3

Enter value for Pname: 'watch'

price: 5000.00

Enter value for pid: 4

Pname: 'pickle'

price: 106.

[ 1 - ]

Op.

PID Pname

1 lipstick 360

2 oldmonk

3 watch

4 pickle

Update:

It is used to modify the data present in table

Alter

modifies tab structure

update

modifies the data.

Syntax : UPDATE table-name  
SET col1 = v<sub>1</sub>, col2 = v<sub>2</sub>, ..., coln = v<sub>n</sub>  
[Where < filter-condition >];

Update PRO

Set price = 2000  
Where Pname = 'Lipstick';

Delete :

To delete the particular record.

Syntax : Delete  
from table-name  
[Where < filter-condition >];

Ex: Delete  
from PRO  
where Pname = 'Oldmenk';

Truncate:

This statement is used to delete the total  
table without disturbing table structure.

Syntax : TRUNCATE TABLE table-name;

## TRUNCATE

3	c	1	y
7	3	5	




We Cannot get  
the record back

Ex:- Select \*  
from pro

op! - 2 Rows selected

Truncate table table-name

- table truncated

Select \* from

pro

Op! - NO Rows selected

DROP

2 3



we can get  
records ball

by Flashball.

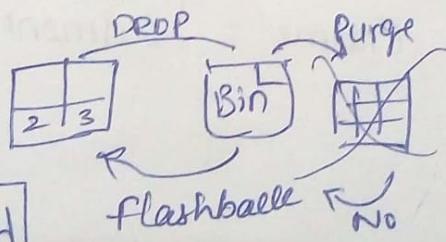
Select \* from

Employee

op: - 14 rows selected

prop table Emp

- Table doesn't exist



Delete

14



F4

We can get  
record ball by  
Rollball

Drop Syntax :

↪ `Drop Table table-name`

To recover :

`Flashback Table table-name`

`To Before Drop`

[ Rename to new-name ]

To drop the table from Recycle Bin :

`Purge Table table-name;`

Commit :

This statement is used to save all the transactions

In the database.

Syntax : `commit;`

Save point :

This statement is used to save the data in the table

but not in the database.

This statement is used to mark the position in

the table.

Syntax : `Savepoint Savepoint-name;`

## Rollback:

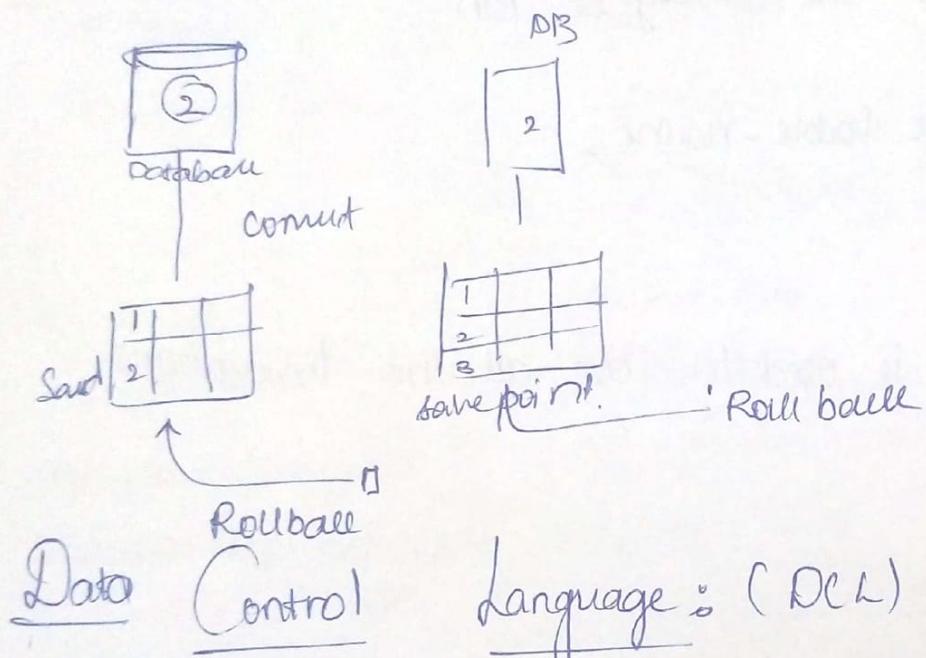
get ball

This statement is used to delete the sword.

It used to go back to the previously committed point.

## Roll back to Savepoint:

Syntax: Rollback to Savepoint-name;



Language: ( DCL )

## 1. Grant

syntax : Grant Sql-statement ON table-name  
TO User-name;

## 2. Revoke:

Revoke sql-statement on table-name

from User-name;

~~Grant~~: It is used to give permission to user.

Revoke : It is used to get back the permission from the User.

Ex:- Grant Select on Emp  
to HR;

Grant Succeeded.

- Enter ORA-name : HR

Enter password : \*\*\*\*  
connected.

Select \*

from EMP;

- table doesn't exists.

Select \*

from SCOTT. EMP.

Emp.	Name	Job	Mgr	HireDate	Sal	Comm	Deptno
------	------	-----	-----	----------	-----	------	--------

conn conn

Enter User-name : scott

Enter password : \* \* \*

- Revoke select on Emp from HR

Revoke Succeeded.

Conn

Enter User name : HR

Enter pass : \* \* \*

It is connected to HR.

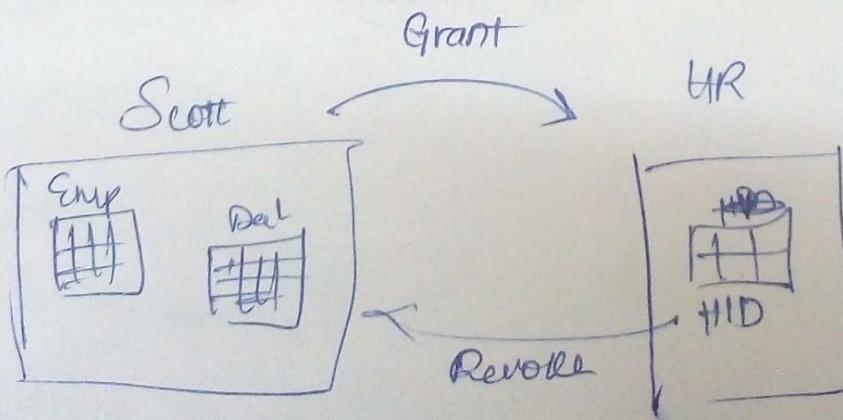
select \*

from scott.Emp

,

from scott.Emp.

- table doesn't exist.



select \*  
from scott.Emp!

## ~~Group~~ Having clause:

Having clause is used to filter the groups.

- \* Having clause Executes G-by-G.
- \* Having clause Executes after the execution of G-by clause.
- \* We can use multi-row functions in Having clause.
- \* We can write multiple conditions in having clause.

### Syntax:

```
SELECT group-functions (MRF) / group_by-clause  
from table-name  
[where <filter-condition>]
```

Group By column-name / Step  
Having < group filter-condition >

### Order of Execution:

From  
Where  
Group  
Having  
Select

### Nature of Execution:

From  
Where - g-by-g  
Group - g-by-g  
Having - g-by-f  
Select - g-by-g.

## Differences b/w where clause & Having clause

Where

1. Where clause is used to filter records condition.
2. Where clause has multiple conditions.
3. Where clause executes before Group by clause.
4. Where clause executes in Row by Row.
5. In where clause we can use column names.
6. Single - Row full ✓
7. we can't use group by clause in where clause.

Having

1. It is used to filter groups.
2. Having clause has conditions.
3. This clause is ~~used~~ executes after the group by clause.
4. Having clause executes in group by group.
5. In ~~where~~ Having clause we use MRF.
6. Can't single row. ✗
7. We can use group by clause in Having clause.

1) W<sub>TOD</sub> Max sal present in each dept. If max sal more than 8000.

Select max(sal)

from Emp

② Group by deptno.

Having max(sal) > 8000;

2) W<sub>TOD</sub> Max sal given to all the Empr in each dept if max sal less than 8030.

Select max(sal), deptno

from Emp

Group by deptno

Having max(sal) < 8030.

3) W<sub>TOD</sub> total sal needed pay for all the Empl who are not getting comm in each job & also their total sal is lesser than 3k.

Select ~~Sum~~ sum(sal), job

from Emp

where comm is NULL

group by job

Having sum(sal) < 3000;

4. WQTD No. of Emps present in each job except president

If they are getting comm more than 800 and Avg sal

Should be less than 1300.

Select ename count(~~\*\*~~), job  
from Emp

Where job not like 'president' AND comm > 800.

Group by job

Having comm > 800 AND Avg sal) < 1300

5. WQTD No. of Emps working as salesman in each dept in which there are atleast 4 Emps working,

Select ename count(\*), Deptno.

From Emp

Where job = salesman

Group by deptno.

Having Count(\*) >= 4.

6. WQTD No. of Emps in each job in which there are atleast 2 Emps working but atmost 6 Emps

In each job.

Select ~~ename~~ count(\*), ~~deptno~~ job

From Emp

~~deptno~~

Group by job;

Having ~~count(\*)= min~~

Count(\*) > 2 AND Count(\*) < 6.

\* Emp  
7. WQTD salaries which are repeated / duplicated.

Select sal  
from Emp  
where Group by sal  
Having count(\*) > 1;

8. WQTD Hired dates which are repeated / duplicated.

Select Hired date  
From Emp  
Group by Hired date.  
Having count(\*) > 1;

9. WQTD names which are present more than once.

Select ename  
From Emp  
Group by ename  
Having

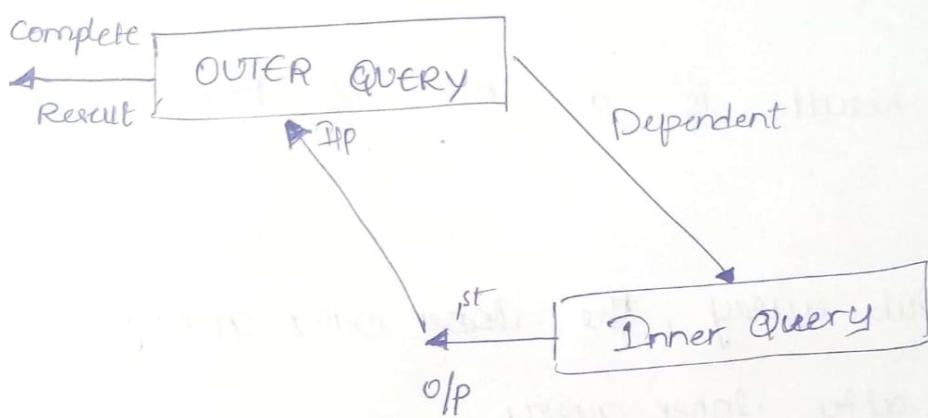
wqtd sal, Dept No if Emps getting same sal in same dept.

10.  
Select ename, sal.  
From Emp

## Sub query:

Sub query is a query written inside another query.

## Working principle of Subquery:



- \* In Sub query there are 2 queries.
- There are Inner query and outer query
- In Sub query outer query is dependent on Inner query.
- We can say it as query inside query.
- In Sub query inner query executes first

- \* The Inner query generates O/P.
- \* The O/P generated from Inner query will be taken as I/P to the Outer query.
- \* The Outer query will execute and give complete result.
- \* The result is a O/P of that Sub query.
- \* In the Sub query , the inner outer query will execute after Inner query.
- \* The Outer query is dependent always on Inner query.

Why or when do we go for Subquery:

Case 1: when ever we have Unknown Values then we go for sub query.

Exs WQTD names of an Emp if they are getting Salary more than 500.

Select ename  
from Emp

where sal > 500.

② WQTD names of Emp if they are getting sal more than Sundri's Sal

Select ename  
from Emp

where sal >

Select sal

from Emp

where ename = "sundri")

ename	sal	do
Sundra	8900	20
Macha	400	10
Sundri	500	20
Machhi	800	30

Sundra = Sundri ✗

Maeke = Sundri ✗

Sundri = Sundri ✓

Maeke = Sundri ✗

## \* Rules to write Sub query :

1. In Sub query, from the inner query ~~the~~ have only one column name.
2. column name Selected from inner query & written in outer query both the column name must be having Same data type.
1. WQTD names of Emp if they are getting Sal more than miller's salary.

WQTD name, sal of Emps are getting Sal less than Scott.

WQTD details of Emp who were hired after turner;  
WQTD @ no. of Emp working in same job as allan job.

WQTD details of emp who are working in same dept as  
Smith's dept.

WQTD details of Emp if they are getting Sal more than

800 but less than 5000.

name, sal of they are getting sal more than

800 & sal less than King's salary.

WQTD details of Emp if they are getting Sal more  
than Smith but less than King.

(Q) select \*

from Emp

where Sal = (select sal from Emp

where Name = "Smith")

AND Sal < (select sal from Emp  
where Name = "King")

A  
i) No. of Emp hired after Smith but before James

2)

Case #2

Whenever data to be displayed present in one table and the condition to be executed present in another table, then we can go for Sub query.

Q:- Write a query to drop the name of Gundri

Emp	Dept	Loc	DNO
ename	DNO		
Sundra	20	'BLR'	10
Machi	10	'HYD'	20
Sundri	30	'GOA'	30
macha	20		

select loc

from Dept

where

10 = 80%

- 30 -

$$y_0 = 30^\circ$$

30 20

80

PK | PK

30

From Emp

where

example = 'Sunday';

Eundra = Sundri X

machi = sandri

Sundri = Sundri  
Sundri

Mather > Smith

① write query deptname of Smith.

Select deptname

from dept

where dno = (select dno

from emp

where ename = 'Smith')

② WQTD loc name of blake

dept

emp

Select loc

from dept

where dno = (select

③ WQTD dept details of all employees of

Emp name is poul. (Research, Dallas)

Select \*

from dept

where

1. WQTD details of emp if they are working

in Accounting dept.

Select empl \*

from emp

where dno = (select dno

from dept

where dname = 'Accounting')

WQTD no. of Emp working in Chicago.

select count (Emp)

from Emp

where loc = (select loc

from Dept

where loc = 'Chicago');

WQTD name, job, dno. of all Emp who are working

in research dept.

select name, job, dno

from Emp

where deptno = (select deptno

from dept

where dname = 'research');

WQTD details of Emp who are working in

Accounting Dept.

select \*

from Emp

where dno = (select dno

from dep

where dname = 'Accounting');

Wanted details of Emp who are working in Accounting  
or Research dept.

Select \*

From Emp

Where dno IN (Select dno

From dept

Where dname IN ('Accounting', 'Research')

### Types of Subquery:

There are 2 types

1. Single - Row Sub query.

2. Multi - Row Sub query.

### Single - Row Subquery:

If sub query returns single row

known as Single - Row Sq.

for Single - Row Sq we can use normal  
operators such as  $=, !=, >, <, \leq, \geq$ .

## Multi-Row Subquery:

If Sub-query ~~returns~~ returns more than one row (or) multiple rows. are known multi-row subquery,

- \* For M-R subquery we can't use normal operators but we can use special operators

Ex: IN, NOT IN, ANY, ALL,  
Exist, Not Exist, ALL

WATD dno, loc of Emp if they are working as Clerk.

select dname, loc  
from emp dept  
where ~~dno~~ in ( select ~~dno~~ deptno  
from dept  
where djob = 'clerk');  
WATD details of Emp who are working in 'Dallas' or  
Chicago.

select \*  
from emp  
where dno in ( select dno  
from dept  
where loc in ('Dallas', 'Chicago'))

WQTD ~~name~~ sal if Emp ~~other~~ are getting  
sal more than Scott.

Select ename, sal.

From Emp

where ~~dept~~ ~~sal~~  $\geq$  (Select ~~dept~~ <sup>sal</sup>  
~~sal~~  $>$

from dept

where ename = 'scott');

\* WQTD details of Emp who are getting sal  
more than Salesman.

Select \*

From Emp

where sal  $>$  (Select Max(sal))

from dept

where job = "Salesman")

## ALL operator :

All operator is an operator multirow operator which having multiple variables and column name (Exp) in LHS doing with Relational - operator.

Conditions : If all operator all values should be true then only it satisfies. Otherwise it won't returns false.

Syntax : Column-name / Exp      Relational Operator

- ALL (v<sub>1</sub>, v<sub>2</sub>, v<sub>3</sub> . . . v<sub>n</sub>);

## ANY operator :

Syntax : Column-name      Relational-operator      ANY (v<sub>1</sub>, v<sub>2</sub>, v<sub>3</sub> . . . v<sub>n</sub>)

→ It will returns True if any conditions is true.

write a query to displ # of Emp who are getting sal more than all the Analyst -

Select

write a no. of Emp getting sal more than atleast  
a salesman.

Select count (ename)  
from Emp  
where ~~sal~~ <sup>\*</sup> > (select min(sal)  
from Emp  
where job = 'Analyst');

Nested Subquery:

We can nest upto 855 Subqueries.

- 1) w<sup>rd</sup> max(sal) from Emp table
- 2) w<sup>rd</sup> first max sal from Emp table

WQTD Qsecond max from emp.

Select max(sal)

from Emp

where SAL > (select max(sal))

sal < 800;

800 < sal < 900 ✓

900 < sal < 1000 ×

1000 < sal < 1100 ✓

from Emp );

where

Sal

100

800

900

600

WQTD Second min(sal).

Select min(sal)

from Emp

where SAL > (select min(sal))

from Emp );

WQTD 3<sup>rd</sup> max sal.

Select max(sal)

from Emp

where sal < (select max(sal))

from Emp

where sal < (select max(sal))

from Emp ));

80

WGETD 6<sup>th</sup> min sal.  
⑥ Select number  
from Emp  
where sal  
>(select mineral)  
from Emp  
where sal >(select mineral)  
from Emp  
where sal >(select mineral)  
from Emp

where  $\rightarrow$  (select min (sal))  
from Emp  
where Sal > (select min (sal))  
from Emp)

 west D 10<sup>th</sup> man each).

WQTD details of Emp who are getting first max sal).

Select \*  
from Emp

where sal = (select max(sal)  
from Emp);

2. WQTD dname of Emp who are getting first min sal.

Select dname  
from dept

where dno = (select dno  
from Emp)

where sal = (select min(sal)  
from Emp);

3. WQTD details of Emp who are getting second min sal.

Select \*  
from Emp

where sal < (select min(sal)  
from Emp)

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

WQTID Loc name of Emp who r getting 3<sup>rd</sup> max sal

Select Loc

from Dept

where ~~sal~~ dno In (Select dno

from Emp

where sal > (Select ~~sal~~ mon(max)

from Emp

where sal < (Select <sup>mon</sup> sal

from Emp));

WQTID Dept details of the Emp getting 5<sup>th</sup>

min sal.

Select \*

from dept

EMPLOYEE - MANAGER RELATION:

28/02/2020

TYPE1: TO find Managers

Ex:- what manager name of Emp 'c'

Empno	Ename	MGR
1	A	3
2	B	NULL
3	C	2
4	D	3.

Select Ename (B)

from Emp

where Empno In Select MGR

1 = D ✗ From Emp

2 > 2 ✓

where Ename = 'C' );

3 > 2 ✗

A > C ✗

at B = C ✗

C = C ✓

D = C ✗

Q. Q. 1) WOTD MGR name of Martin.

select ename  
from Emp

where

empro. In select MGR  
from Emp  
where ename = 'Martin');

② WOTD Smith's manager details.

select \*  
From Emp

where Empno In select MGR  
from Emp

where ename = 'Smith'

③ WOTD Department of Blakes Manager

select Dname

from dept

where dno In select dno

from Emp

where Empno In select MGR

from emp

where ename = 'Blake');

WQTD loc name of Emp who are working as Analyst's Manager

select loc , from dept

from Dept

where Empno = select Mgr

from Dept

where job = 'Analyst';

WQTD Blakes manager's manager name.

select ename .

from emp

where Empno = select Mgr

from Emp

where Empno = ( select Mgr

from Emp

where ename = 'Blake' );

Ans

① WQTD Dname of Clerk's manager's manager

Where ename = 'Clerk');

Type : 2

→ To find REPORTERS

1) WQTD Name of an Employee who are Reporting  
to Emp 'B'.

Select ename ①

from Emp 2

where mgr in (select empno  
 $\boxed{2=2}$  ✓ from emp)

where ename = 'B');

$\boxed{B=B}$  ✓

WQTD no. of Emp reporting to king

Select count(\*) .

from Emp

where mgr in (select empno  
from emp  
where ename = 'king'));

WQD Details of Emp who are reporting to Blane

Select a

from Emp

Where Mgr = Select empno

from Emp

where ename = 'Blane';

WQD Name of Emp who are reporting to Scott

Select ename

from dept

Where Dno = select dno

from Emp

Where Mgr = Select empno

from Emp

Where ename = 'Scott');

WQD No of Emp who are reporting to Salesman.

Select loc

from dept

Where dno = select dno

from Emp

Where Mgr = Select empno

from Emp

Where job = 'Salesman');

## JOINS:

Joins are the retrieval of data from multiple tables simultaneously.

- \* In joins we use multiple tables to retrieve data.
- \* The drawback in subquery is, we can retrieve data only from one table but not from multiple tables.

## CARTESIAN

### Syntax

### Note

new

and

with

## Types of Joins:

1. CARTESIAN JOIN / CROSS JOIN
2. INNER JOIN / EQUAL JOIN
3. NATURAL JOIN
4. OUTER JOIN
  - (i) Left outer
  - (ii) Right outer
  - (iii) Full outer
5. SELF JOIN

## CARTESIAN JOIN / CROSS JOIN :

The In Cartesian join , the records of table 1

will be merged with all the records of table 2.

Syntax : / (SQL) ANSI : Select column - names  
From . table - name 1 CROSS JOIN table - name 2;

Ex 1 Oracle : Select column - names  
From table - name 1 , table - name 2;

Note : \* In cartesian join total no. of records in  
resultant table will be a product of records of table 1  
and table 2.

Total no. of records =  $T_1 \times T_2$

\* Total no. of columns in resultant table  
will be summation of columns present in table 1 & table 2.

Total no. of columns =  $T_1 + T_2$

Player name along with his name

Object Name, & name

Boys Boston, girls

The draw back of Cartesian join is we get more no. of error records.

Because of merging with  $n$  tables we get  $n^m$  many error records.

## 2. INNER JOIN | EQUAL JOIN :

Inner join is used to obtain only matched records.

### Syntax:

ANSI: select column-names  
from table-name1 INNER JOIN table-name2

ON <join-condition>;

### ORACLE:

select column-names  
from table-name1, table-name2

where <join-condition>;

JOIN condition:  
table-name1.column-name = table-name2.column-name  
(condition)

Join condition: On the two tables, when we get merged, then join condition exists.

① WARD BRANE, GRANE  
select brane, grane  
from Boys, Girls.

where Boys.GID = Girls.GID /  
$$\begin{array}{r} \text{Boys.GID} \\ \hline 1 \\ \times \\ 2 \\ \hline 2 \end{array}$$
  
$$\begin{array}{r} \text{Girls.GID} \\ \hline 1 \\ \times \\ 2 \\ \hline 2 \end{array}$$
  
$$\begin{array}{r} 3 \\ \times \\ 2 \\ \hline 6 \end{array}$$
  
$$\begin{array}{r} 32 \\ \times \\ 2 \\ \hline 64 \end{array}$$
  
$$\begin{array}{r} 32 \\ \times \\ 2 \\ \hline 64 \end{array}$$
  
$$\begin{array}{r} 128 \\ \times \\ 2 \\ \hline 256 \end{array}$$

② want ename, dname of all Emp

Select ename, dname  
from ~~Dept~~ Emp, Dept  
where Emp. deptno = Dept. deptno;

3) want ename, name, job, loc

c  
Select ename, name, job, loc  
from Emp, Dept  
where Emp. deptno = Dept. deptno;  
want ename, sal along with, deptno & loc.  
of Emp who working as Salesman.

oracle  
Select ename, sal, dname, loc.  
from Emp, Dept  
where Emp. deptno = Dept. deptno,  
job = 'Salesman';

Ans →  
select ename, sal, dname, loc  
from Emp, Dept join Dept. Deptno.

d) select eno, sal, comm, loc, If Emp getting sal > 2000

From dept, Emp

Where Emp.deptno = dept.deptno AND sal > 2000;

5. select name, job along with loc of emps if they are working as prsident in Accounting dept.

Select ename, job, loc

From emp, dept

Where Emp.deptno = dept.deptno AND job = 'President';

AND dname = 'Accounting';

Emps are getting com

6. select Emp no, sal, draw If Emp no, sal, draw

in sales Dept

Select 'Empno, sal, draw'

From dept, Emp

Where Emp.deptno = dept.deptno AND comm is not null

AND draw > sales /

AND draw > sales /

7. WQT P . details of Emp along with loc & Emp are located in 'Chicago'.

select ~~EMP~~\* , loc  
from emp, dept

where emp.deptno = dept.deptno AND loc = "Chicago"

8. WQT P . Details of Dept along with ename if

Emp are not working as clerk

select dept.\* , ename

from emp,dept

where emp.deptno = dept.deptno AND job != 'clerk'

9. WQT P . details of Emps & Dept If Emps are working as

Salesman / manager in Sales or Research Dept.

select ~~EMP~~\* , dept

from emp, dept

where emp.deptno = dept.deptno AND job = 'Sales'

OR job = 'manager' AND ename = 'Sales' OR 'Dana'

working as Analyst. In Dallas or Boston  
Select ename, sal, job, deptno, loc

from emp, dept  
where emp.deptno = dept.deptno AND job = 'Analyst' AND  
loc = 'Dallas' OR loc = 'Boston'

<sup>dept</sup>  
+-----+  
| Dallas | Boston |  
+-----+

11. WARD Name, Comm, hiredate  
all working in dept 30 or 20 and Clark or Manager or salesman.

Select ename, comm, hiredate, drama, loc  
from emp, dept  
where emp.deptno = dept.deptno AND  
job = 'clerk' OR job = 'job in ('clerk', 'manager', 'salesman')'  
AND  
emp.deptno = 30 OR 20 AND  
job = 'clerk' OR job = 'job in ('clerk', 'manager', 'salesman')'

12. WARD total sal needed to pay for all emps in each dept by

using joins.  
~~sum(sal)~~, Emp.deptno

Select sum(sal), Emp.deptno  
from emp, dept  
where emp.deptno = dept.deptno AND  
group by deptno;

13. WARD normal present in each dept except president in  
which there are 3 are working.

Select marital, sum(deptno)  
from emp, dept  
where emp.deptno = dept.deptno AND count(\*) = 3  
AND job != 'president'  
Group by emp.deptno  
Count(\*) = 3;

~~WAP~~ WAP ID: 101  
Want to get along with Dname in each dept  
in which there are atleast 2 but atmost 6 emps are working.

Select sum(sal), dname

From dept, Emp  
where Emp.deptno = dept.deptno  
Group by deptno;

15. Want details of emps along with dname if emps are getting 1<sup>st</sup> max sal.

Select emp.\* , dname

From emp, dept

where Emp.deptno = dept.deptno AND sal = ( select maxsal  
from sup);

Want Ename, dname, loc

Select ename, dname, loc

From Emp.E, dept.D

Where E.deptno = D.deptno;

WQTD No. of Emp working on Accounting dept.

select count(\*)  
From Emp  
Where Deptno = Emp.deptno  
Deptno in ( select deptno  
from dept  
where dname = 'Accounting' );

Joins ↗

select count(\*)  
from Emp.E, Dept.E  
where Emp.deptno = dept.deptno AND  
pname = 'Accounting';

WQTD dname of Emp who are working as Salesman.

select dname  
from Emp-E, dept-B  
where E.empno = D.deptno AND job = 'Salesman';

## Natural JOIN:

It is similar to INNER JOIN but here we have to write join condition.

Syntax:- Ans

Select column-name

from tablename1 NATURAL JOIN tablename2

Oracle: select column-name

from t1, t2

\* When do you go for Natural JOIN? →

When we don't know table structure.

But still you want matched records.

Note: \* Natural JOIN gives Inner JOIN o/p when they

have PK - FK relationship.

\* Natural JOIN gives Cartesian JOIN o/p when they don't have PK - FK relationship.

Ex:-

Select \*

From Emp Natural Join Dept

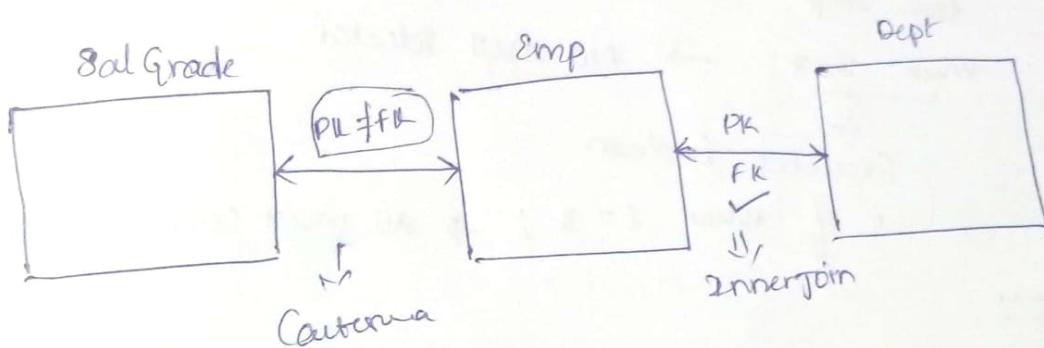
Op:- 14 rows

( It have PK - FK relationship )

Select \*

From Emp Natural Join Dept

Op:- 50 rows ( It dont have PK - FK relationship )



1) WQTD Empno, job, deptno along with dname

select empno, job, deptno, dname

From dept ~~, emp~~, Natural join Emp ;

2) WQTD empno, loc, dname If Emp working in Chicago.

Select empno, loc, dname

From emp Natural join dept

Where loc = 'Chicago';

WQTD select ename, sal order by sal + ename

select ename, sal

from emp.

order by ename, sal;

sal, ename

ename in order  
sal

SQ: select \*

from emp

where s=2; → No rows selected

↓  
Execute boolean

↳ If where s=3; → All rows selected.

Note:

where clause works for boolean

1) WQTD first 3 char from Emp name.

\* 1 Select SUBSTR(ename, 1, 3)

2) WQTD first ~~1~~ half of name

3) WQTD second half of name.

A) PRATHYUSHA

calculate no. of times A present in your Name

Select count(\*)

from Dual

where / SUBSTR Like '%AU%'

Entered by  
self..

~~all join are used~~ to obtain matched records.

~~Left outer join:~~

~~left outer join is used to obtain matched records along unmatched records from left table.~~

Syntax:

Ans: Select column-names  
from table-name1  $\text{LEFT OUTER JOIN}$  table-name2  
on table-name1.column-name = table-name2.column-name

Oracle:

Select column-names  
from table-name1, table-name2  
where table-name1.column-name = table-name2.column-name

NOTE matched records along with Unmatched records  
of left table (Boys)

oracle select \*

from Boys B , Girls G

where B.GID = G.GID (+);

	Boys LEFT	Girls RIGHT
BID	Bname	GID
1	Raja	2
2	Dumna	1
3	Munna	3
4	Sundra	Null

	Boys LEFT	GID
1	Raja	2
2	Dumna	1
3	Munna	3
4	Sundra	4

Result

BID	Bname	GID	Gname
1	Raja	2	Rani
2	Dumna	1	Dumni
3	Munna	3	Munni
4	Sundra	Null	Null

Un

Matched.

↓ unmatched

## Right outer join:

To obtain the matched records along with unmatched records from right table.

### Syntax

Ans: Select column names

from tb-n1 ~~LEFT~~<sup>RIGHT</sup> [OUTER] JOIN tablename2  
~~ON~~ ~~where~~ ~~tb-n1.coln(+)~~ = tb2.col-2

Model

Select coln

From table name 1 , table name 2

Where tablename1.columnname1(+) > tablename2.col

Obtains matched records along with unmatched records from Right table.

Select \*

From Boys-B , Girls-G

Where Boys.GID(+)> Girls.GID

Result

BID	Bname	GID <sup>1</sup>	GID <sup>2</sup>	Gname
1	Raja	2	2	Rani
2	Dumna	1	1	Dumni
3	Munna	3	3	Munni
NULL	NULL	NULL	4	Randri } UVM

Full Outer Join:

To obtain matched records along with unmatched records from both the tables.

\* NO - oracle syntax

ANSI : Select colnames  
From tablenum1 [ ] FULL OUTER JOIN tablenum2  
ON tablenum1.colname1 = tablenum2.colname2

WRTD matched records along with unmatched records

of both tables -

Select \*  
from BOYS [ ] FULL [OUTER] JOIN Girls G.  
ON BOYS.GID = GIRLS.GID

1	Raja	2	2	Pani
2	Dumne	1	1	Dumne
3	Munna	3	3	Munni
4	Sundre	null	null	New
	null	null	4	Sundre

TD ename, dname if emps are working in me dept. or not

Select ename, dname  
 from Emp  
 where Emp.dno = Dept.dno;

Where  
 $Emp.dno = Dept.dno$

ename, dname if the dept having some  
 emps working in it or not

Select ename, dname

from Emp, Dept

where  $Emp.dno = Dept.dno$

full outer

Select ename, dname  
from Emp. FULL OUTER JOIN Dept  
ON Emp-dno = dept-dno.

WRTD Ename, dname If Emp are not working  
in any dept.

Select ename, dname  
from Emp-e, Dept-D

WRTD Emp det & dept det If dept doesn't have  
any employees working in it.

Select ~~e~~ \*

from Emp-e, Dept-D  
ON ~~where~~ Emp-dno = dept-dno, ~~AND~~ where ename is null.

WRTD Sel ename, dname  
Sel \*  
from Emp-e, Dept-D

ON ~~where~~ Emp-dno = dept-dno, AND where ename is null,  
where ename is null OR dname is null

Difference between Inner join & Outer join

- Inner join used to obtain only matched record
- Outer join used to obtain Mismatched along with Unmatched records.
- Outer join can also obtain only unmatched records.

## New SELF JOIN:

Joining same two tables are known as

Self join.

Why (or) When we go for SELF JOIN:

- \* When we have data present in same tables but they are present in different rows.

Syntax: ~~Ans~~ Select columns

from tablename T1 JOIN tablename T2  
ON JOIN condition

Oracle:

Select columns

From Tablename T1, Tablename T2

Where < Join condition >

T1.col = T2.col etc.

Ex: WORD -> Emp details & manager details

Select \*

From Employee, Employee

Manager = Emp no.

Where E1. ~~Emp~~ = E2.

Emp. E<sub>1</sub>

E <sub>1</sub> No	E <sub>1</sub> name	M <sub>1</sub> ur
1	A	3
2	B	new
3	C	2
4	D	1

Repeat

Emp. E<sub>2</sub>

E <sub>2</sub> No	E <sub>2</sub> name	M <sub>2</sub> ur
1	A	3
2	B	new
3	C	2
4	D	1

E <sub>1</sub> .E <sub>2</sub> No	E <sub>1</sub> .E <sub>2</sub> name	E <sub>1</sub> .M <sub>1</sub> ur	E <sub>2</sub> .E <sub>1</sub> no	E <sub>2</sub> .E <sub>1</sub> name	E <sub>2</sub> .M <sub>2</sub> ur
1	A	3	3	C	2
2	B	new	2	B	new
3	C	2	1	A	1
4	D	1	4	D	3

① NEED

Enames & their manager's name.  
select e1.ename, e2.name

Jobs line

from Emp E1, Emp E2

where E1.Empno = E2.Empno / ~~mgr~~

②

Employees job along with their manager's job.

select Job E1, J

from Emp E1, Emp E2

where E1.Empno = e2.mgr;

③ NEED

Emp details along with their manager's commission

if Emps are getting sal more than 2000.

select \* ~~E1.\*~~, E2.comm

from Emp E1, Emp E2

where E1.MGR = E2.Empno

AND E1.Sal > 2000

their manager's sal

④

NEED Emp details along with

if Emps getting sal more than ~~2000~~  
getting Sal less than 5000.

but may all get selected  
but their managers are

~~Select E1.\*,~~

Select E2.\*, E1.Sal

From Emp E1, Emp E2

& E2.Empno & AND E1.Sal > 2000 and E1.Sal < 5000;

where E1.MGR

Q) Show, com., manager details of Emp whose gets  
comm. fact that Mgr gets comm.  
select e1.ename, e1.comm, e2.\*  
From Emp.e1, Emp.e2  
where e1.empno = e2.mgr AND e1.comm IS NOT NULL  
AND e2.comm IS NOT NULL

Q) What Emp & Emp's job also mgr bld if emp  
find if Emp ever turned after 87  
select ename, job, e2.\*

Q) What Emp \*, also mg details of both  
working in same job

select e1.emp.\* , e2.\*  
From Emp.e1, Emp.e2  
where e1.empno = e2.mgr AND

⑦ WQTD Emp Name Job along with mngs details  
of EMP all workers as mgr & their mngs are unique  
Actual ans

Select E1.ename, E1.job, E2.\*  
From Emp.E1, Emp E2  
Where E1.empno = E2.mgr

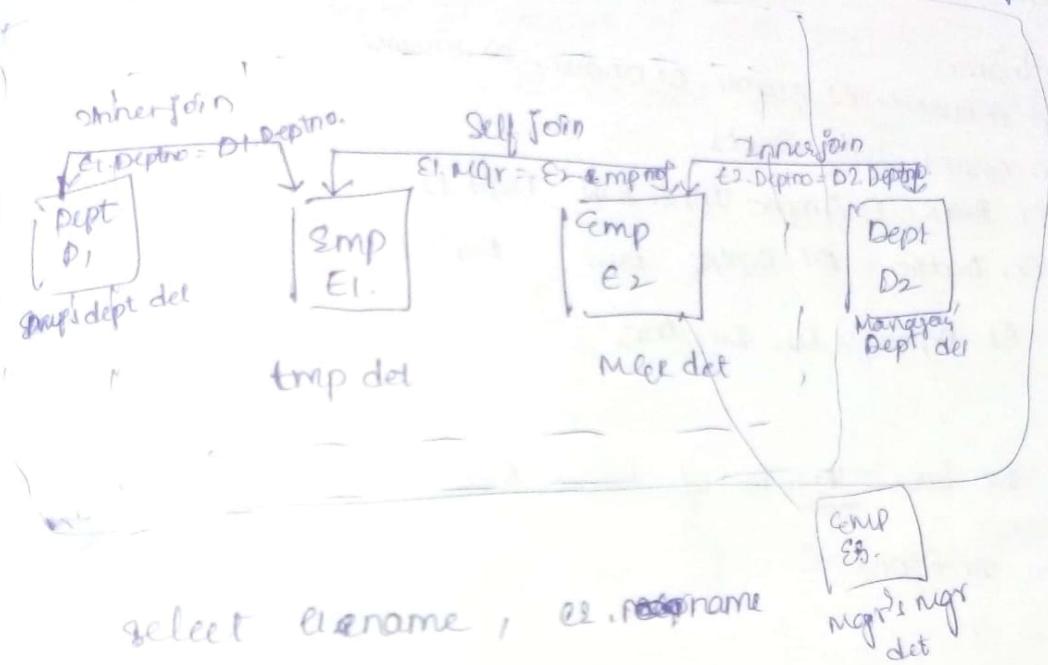
⑧ WQTD Empls Empno, job ,hiredate along with their  
managers details If both hired on same day.

Select E1.empno, E1.job, E1.hiredate, E2.\*  
From Emp.E1, Emp E2  
Where

⑨ WQTD Empls details along with their manager details if  
Emp are getting Sal more than their manager sal.

Select E1.emp.\* , E2.\*  
From Emp.E1, emp.E2  
Where E1.empno = E2.mgr AND

Write a query to display Ename & his Manager name.



Select Ename, E2.name

From Emp E1, Emp E2

where E1.Mgr = E2.Empno.

Ans

Select E1.name, E2.name  
From Emp.E1 JOIN Emp.E2  
ON E1.Mgr = E2.Empno.  
) and Ename & His Manager name along with Employees  
deptname.

Select E1.name, E2.name, D1.name

From Emp.E1 JOIN Emp.E2  
ON E1.Mgr = E2.Empno INNERJOIN Dept D1

ON E1.Deptno = D1.Deptno.

Want ename & his manager name along with

their dnames.

Select e1.ename, e2.ename, D1.Dname, D2.Dname

From Emp e1 JOIN Emp e2.

ON e1.mgr = e2.empno INNER JOIN Dept D1.

ON e2.deptno = D1.deptno Dname Dept D2.

ON e2.deptno = D2.deptno.

Note : If we have n no. of tables then should be having

n-1 join conditions.

q) want ename & his manager name along with  
manager's manager name

Select e1.ename, e2.ename, e3.ename

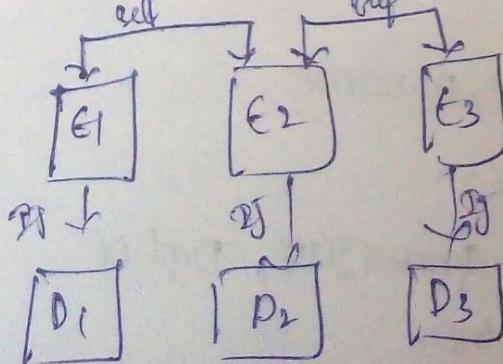
From Emp e1 JOIN Emp e2

ON e2.mgr = e3.empno ~~inner join Emp e3.~~

ON ~~e3.empno~~ ~~e3.mgr~~ ~~e3.empno~~.

Want ename & his manager name & manager's manager name

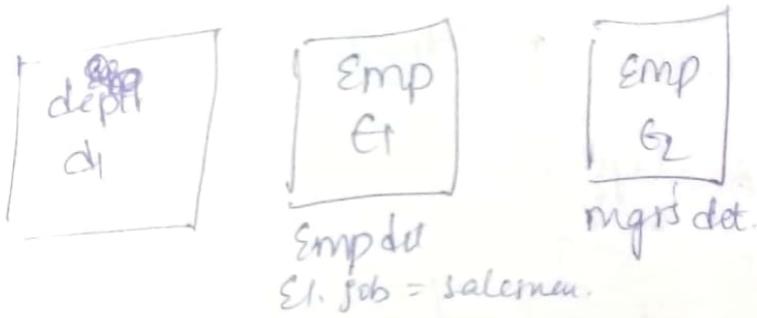
along with their Deptnames



select E1.ename, E2.ename, E3.ename, D1.Dname  
 From Emp E1 JOIN Emp E2, Emp E3, Dept D1.  
 ON E1.Mgr = E2.empno INNER JOIN Dept D1.  
 ON E1.Dept

N

want employee job along with their manager's name  
 & also display emp's loc. if emp is working as  
 salesman.



Select E1.job, E2.ename, D1.loc  
 JOIN  
 From Emp E1 JOIN Emp E2  
 ON E1.Mgr = E2.empno JOIN Dept D1.  
 ON

word

emp val & manager's job if Emps are working

Accounting department

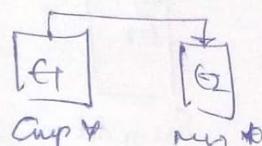
Int. Job

Dept

&lt;p

Imp:

1) WATD cname, job & his manager name, sal. if emp name = blake.



select E1.ename, E1.job, E2.ename, E2.sal

from Emp E1 JOIN Emp E2

on E1.empno = E2.empno

where E1.ename = 'blake';

2) WATD Dname of blakes manager.

select D

select Dname

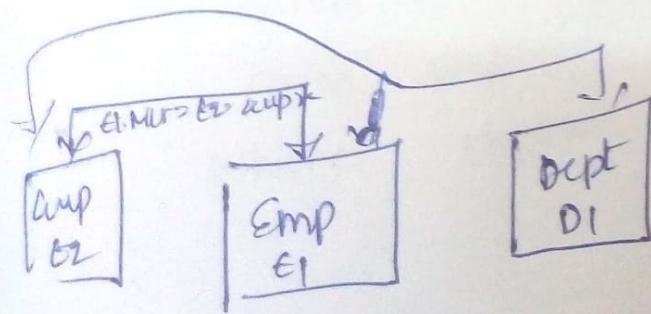
from Dept  
Deptno

where Deptno = 101

in ( select Dname from emp  
where empno = E1.mgr )

or select MGR

from Emp  
where ename = 'blake';



select D1.Dname

from Dept D1 JOIN Emp E1

ON E1.mgr = E2.empno IF Dept D1

ON E2.deptno = D1.deptno

where E1.ename = 'blake'



Select Dr.ename

from

Emp e<sub>1</sub>

Join Dept e<sub>2</sub>

ON

e<sub>1</sub>. empno = e<sub>2</sub>. deptno

where

e<sub>1</sub>. loc = 'King';

WORD loc name

of

emp who are reporting to Date.

Select dr. loc

[E1]

[E2]

from

Emp e<sub>1</sub> JOIN Emp e<sub>2</sub>

ON

E<sub>1</sub>. empno = E<sub>2</sub>. mgr. Inner join D<sub>2</sub>

ON

Where e<sub>1</sub>. ename = 'Balki';

\* WORD Dept name of employees who are getting sal

more than 500.

Select Drname

from Dept

in (Select Drname

select loc  
from Emp  
where loc = 'King')

where loc = 'King'

WQTD the 2<sup>nd</sup> Rownum rec from Emp.

Emp

Ename	sal	Bno
A	100	20
B	200	10
C	100	20
D	400	30

Result

SL.NO	Ename	sal	Bno
1	A	100	20
2	B	200	10
3	C	100	20
4	D	400	30

Select \* Rownum  
from ( select SLNO , emp.\*  
from Emp )

where SLNO = 2;

$$\boxed{1 = 2 \times}$$

$$2 = 2 \checkmark$$

$$3 = 2 \times$$

$$4 = 2 \times$$

pp: table:

SLNO	Ename	sal	Bno
2	B	200	10

WQTD 5<sup>th</sup> record from table -

Select \*  
from ( select Rownum , SLNO , emp.\*  
from Emp )

where SLNO = 5;

~~class~~

W@ID bottom 3 record from the table

select \*

from ( select Rownum SLNO, Emp.\*  
from emp)

where SLNO > ( select count(\*) - 3  
from Emp);

W@ID 1<sup>st</sup> record , 8<sup>rd</sup>, 7<sup>th</sup> & 10<sup>th</sup> record.

Select \*

from ( select Rownum SLNO, Emp.\*  
From emp)

where SLNO in (1, 3, 7, 10);

W@ID bottom 5 records from table

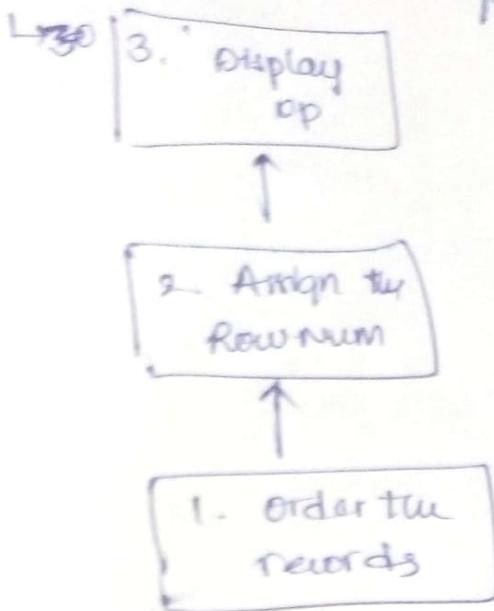
Select \*

from ( select Rownum SLNO, Emp.\*  
From emp)

where SLNO > ( select count(\*) - 5  
from Emp);

To find  $m^{th}$  maximum (or)  $n^{th}$  minimum sal:

We have 3 steps to be follow



1. WID

$8^{th}$  max sal.

D

A

select sal  
from ( select Rownum SLNO, sal  
from ( 2. select Distinct sal )  
1. from emp  
3. order by sal Desc ) %

where SLNO = 3;

$$1 = 3 \times$$

$$2 = 3 \times$$

$$\boxed{3 = 3}$$

Emp	Ename	Sal	Dno
A		600	20
B		700	10
C		800	20
D		600	80
E		800	10

0	Res1	Sal
		900
		800
		700
		600

Res2	Res3	Sal
Res2	Res3	Sal
Res2	Res3	900
Res2	Res3	800
Res2	Res3	700
Res2	Res3	600

Res3	Sal
Res3	Sal
Res3	700

WQTD

10<sup>th</sup> max sal.

Select sal

From

C ( select Rownum SNO, sal ) - A  
From ( select Distinct sal  
From Emp  
Orderby sal Desc )) } 0  
Where SNO = 10 ;

WQTD

8<sup>th</sup> min sal.

(E)

Select sal

From C ( select Rownum SNO, sal )  
From ( select Distinct sal  
From Emp  
Orderby sal Asc )) .

Where SNO > 8 ;

WQTD

TOP 3 max sal.

Select sal

From C ( select Rownum SNO, sal )  
From ( select Distinct sal  
From Emp  
Orderby sal Desc ))

Where SNO < 4 ;

a) write top 5 min sal.

```
select sal  
from ( select Recnum sno, sal  
       from ( select Distinct sal  
              from emp  
              order by sal Asc ))  
where sno > 6;
```

b) write bottom 5 max sal.

```
select sal  
from ( select Recnum sno, sal  
       from ( select Distinct sal  
              from emp  
              order by sal Desc ))
```

c) write bottom 3 min salaries.

```
select sno, name  
      from emp  
      order by sno Asc
```

Q) write details of emp who are getting 10<sup>th</sup> max. salary.

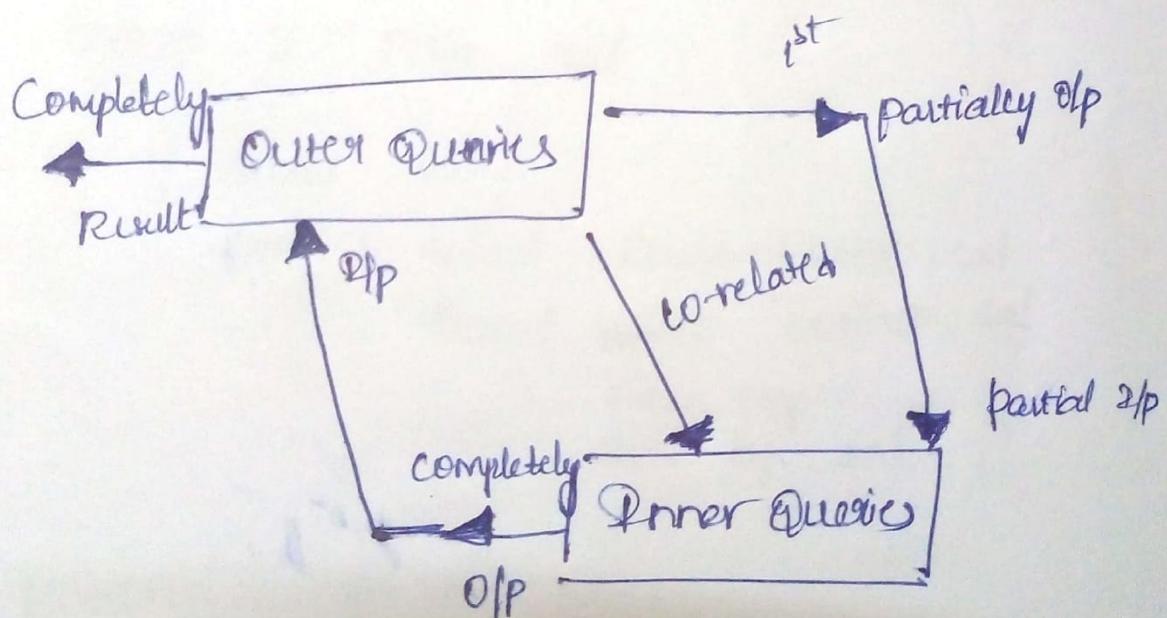
Select \*  
from emp  
where sal = (select sal  
from ( select max(sal)  
from (select distinct sal,  
from emp  
order by sal desc)))  
where sno = 10 ;

## Co-Related

## Sub Queries :

The Inner Query is dependent on Outer Queries and Outer Query is dependent on Inner Query is called Co-related Sub Query.

### Working principle of Co-related Sub Queries:



1. In co-related Sub Queries Outer Query and Inner Query is dependent on each other.
2. Then Outer Query executes 1<sup>st</sup> but gives partial O/P.
3. The Partial O/P generated from Outer Query is given to Inner Query as sp.
4. The Inner Query takes the Partial R/P
5. The Inner Query will generates O/P.
6. The O/P from Inner Query will execute.
7. That R/P generated and gives completely.
8. That O/P generated from R.Q will be given to Outer Query.
9. Outer Query will takes that R/P.
10. O.U.Q generates O/P.

## ORDER - BY - CLAUSE

It is used sort the records in ascending or descending order.

### Syntax:

Select group-by-expression / group-function  
from table-name  
[where < filter-condition ]  
[ Group By column-name / Exp ]  
[ Having < group-filter-condition ]  
ORDER By column-name [ ASC ] / DESC )

### Order of Execution:

1. From
2. Where (If any)      (R - by - R)
3. Group by      (G - by - G)
4. Having      G - by - G
5. Select      R - by - R
6. ORDER BY order      R - by - R

- \* Order by clause must be used as the last clause in the statement.
- \* for order by clause we can pass column-name or exp as an argument.
- \* Order by clause executes after the execution of "Select clause".
- \* By default order by clause sorts the records in ascending order.

1. ~~WRTD~~ Employees Name In ascending order.

select ename

from Emp

order by ename .

2. ~~WRTD~~ \* of emp where emps name should be sorted in descending order.

select \*

from Emp

order by ~~ename~~ DCL

WRTD Annual sal of Employees In ascending order.

select sal\*12

from Emp

~~with~~ order by sal\*12 asc;

### Notes

- \* we can use Alias name in order by clause.
- \* we can use multiple columns in order by clause.

Q) What details of Emp where their dno & sal  
Should be sorted in Ascending order

```
Select *  
from Emp  
order by dno, sal;
```

Q) What dno & sal of all the Emp where dno  
be sorted in descending order & sal should be sorted  
ascending order.

```
Select dno, sal  
from Emp  
order by dno desc, sal asc;  
order by sal
```

(Ans)

Select dno, sal  
from Emp  
Order by 2, 1

Ward details of Emp if they are earning more than  
1000 & working in dept 20% where dno should be sorted  
in ascending order.

Select \*

From Emp

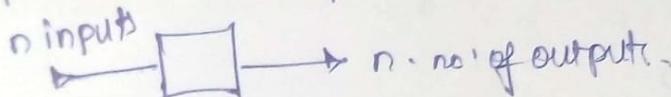
Where Sal > 1000 AND

Orderby dno;

DNO in (10, 20)

### Single - Row - functions:

→ If we put 'n' no. of inputs



→ We can pass Single - Row fn in where clause.

→ We can have nest single-row functions

### Types of Single - Row - function:

#### 1. length():

This fn is used to count the no. of characters present in the given string.

#### length('string'):

It is a dummy table which is used perform

2. Dual: It is a dummy table which is used perform any mathematical operations.

Select length('Prathyusha Amul');

From Dual

Op: 15

Select length ('I Hate you')  
from dual

WOTD Ename as well as no. of char present in their  
name

Select ~~ename~~, length (ename)  
from Emp.

WOTD Ename who are having Excut & class  
in their name.

Select ename, length('ename')  
from Emp  
where len(ename) = 4

WOTD Sal of Emp who are earning 3 digit Sal  
using single - R fun

Select Sal  
from Emp  
where ~~sal~~ length(sal) = 3;

Reverse)

It is used to reverse the given String.

Select Reverse ('Q spiders'),  
From Dual

op: So edPpsQ

Select Reverse('Malayalam'),  
From dual

op: Malayalam

Is palindrome  
WED whether the string is palindrome!, string = oppo  
Select "Is palindrome"  
From Dual  
When 'oppo' = Reverse ('oppo')

### Sysdate:

It is a command which is used to get the current date from system.

Select sysdate  
From dual;

op:  
| 02 - MAR - 20

Select to char (sysdate,'Year')  
From dual;  
char (sysdate,'Year')

| TWENTY TWENTY

Select to char (sysdate,'Year')  
From dual

| 2020

write a query to display the current date in us

select to\_char(sysdate, "MM-DD-YY")

from dual;

WQTD Employee name who were hired in the month of feb.

using single row - fun

Select ename

From emp

where ~~hiredate = (select to\_char(sysdate ("mon"))~~  
~~to\_char(hiredate, 'mon') in 'FEB'~~

(or)

Select ename

from emp

where to\_char(hiredate, 'mon') = 'FEB'

To\_date:

This function is used to convert the given no. into date format.

WQTD month of hired date

select to\_char(to\_date('02-MAR-20', 'mon'))  
from dual.

MAR.

Select SUBSTR  
from dual;

SUBSTR

02-MAR-90 09:22:10.077000 AM 05:30

SubString (substr()):

This fun is used to extract the part of a string from given original string.

Suppose

DECODE ('ORACLE - STRING', position, length);

Ex:-

BAN~~H~~ARE(OR~~E~~AL) → B

SUBSTR(BAN~~H~~ARE, 3, 1) → R

3, 4 → RAA

3, 5 → RAE

SUBSTR(BAN~~H~~ARE, 5, 2) → AE

9, 1 → Bangalore

-5 → NULL

-5, 2 → NA

WQTD Ename who are having exactly 4 characters.  
In thier name using single -Row -fun.

select ename

from Emp

~~where SUBSTR(~~

Where

WQTD first 3 chars of Emp name.

select ename

from Emp

1. where like ~~the~~ ~~that~~ SUBSTR (ename , 1,3 )  
2.

WQTD The last char of Emp name

select SUBSTR (ename , 4,1 )

from Emp ;

WQTD The first char of Enam.

select SUBSTR (ename (1,1) )

from Emp ;

WQTD ename who are having char A in their name  
using single -Row fun.

### 1. char

Select SUBSTR (ename (1,1))

from Emp

where ename

Like "A%"

Select ename

from Emp

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

with Ename who are having char 'A' or 'S' or 'M' or

In their first place -

Select ename

from Emp

where SUBSTR (ename (1,1)) In ('A', 'S', 'M');

- WQTD ename where names are starting with vowels.

Select ename

from Emp

where SUBSTR (ename (1,1)) In ('A', 'E', 'I', 'O', 'U');

3.

### Replace C)

This fun is used to replace the substring by present

in original string by providing new string.

Syntax -

Replace (ORIGINAL-STRING, SUB-STR, NEW-STRNG);

Select Replace ( 'Dspiders', 'Q', 'T' )  
from Panel.

Tspiders .

Select Replace ( 'Qspiders', 'Q' )  
from dual  
spiders.