25/11/2021

Group by clause (V. IMP)

- used for grouping

where clause → used for searching.
order by clause →

EMP as yesterday:

- Select sum(sal) from emp; O/P = 35000
- select sum(sal) from emp where Deptno = 1; O/P = 18000

sum of (sal) deptwise:

DEPTNO	SUM(SAL)
1	18000
2	17000

← output:

- select deptno, sum (sal) from emp group by deptno;

Step 1: Rows retrived from Database server HD to server RAM

Step 2: Sorting deptwise data.

Step 3: Grouping is done deptwise.

Step 4: summation deptwise

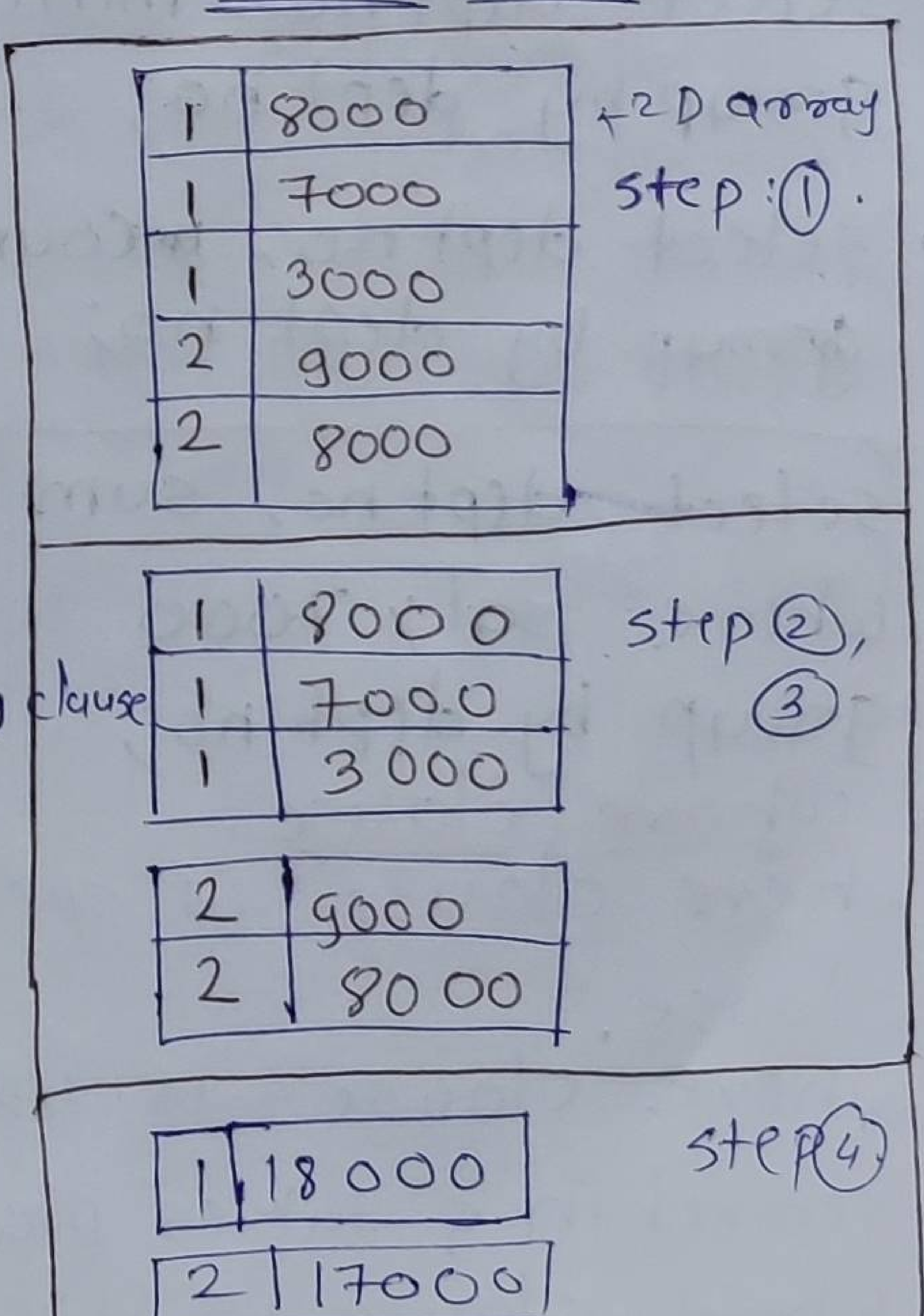
Step 5: Having clause

① Select clause → Step 6: order by clause
select dept no, sum(sal)

② From clause →
from emp.

③ group by clause →
group by dept no.

Server RAM



1	8000
1	7000
1	3000
2	9000
2	8000

← 2D array
step: ①.

1	8000
1	7000
1	3000

2	9000
2	8000

step ②, ③

1	18000
2	17000

step ④

Rule ①

- Besides the group function, whichever column is Present in select clause, it has to be present in Group by clause.

```
select deptno, sum(sal) from emp; ← Error (Oracle).
```

(this works in MySQL, but the output is meaningless).

Rule ②

- whichever column is present in group by clause, it may or maynot be present in select clause.

-
  ```
  select sum(sal) from emp
  group by deptno;
  ```

  O/P:
  ```
  SUM(SAL)
  --------
  18000
  17000
  ```

-
  ```
  select sum(sal), deptno from emp
  group by deptno;
  ```

-
  ```
  select deptno, max(sal) from emp
  group by deptno;
  ```

-
  ```
  select deptno, min(sal) from emp
  group by dept no;
  ```

-
  ```
  select deptno, count(*) from emp.
  group by dept no;
  ```

-
  ```
  select deptno, sum(sal) from emp.
  where sal > 7000
  group by dept no;
  ```

where clause:

- where clause is specified before the group by clause.
- where clause is used for searching.
- searching takes place in DB server HD.
- Where clause is used to restrict the rows.

- Where clause is used to retrive the rows from server HD to server ram.

output:

DEPTNO	sum(SAL).
1	8000
2	17000

---

dptNo-25
Job→1000

- select deptno, job, sum(sal) from emp group by dept no, job;

output

DeptNo.	Job.	sum(SAL).
1	C	10000
1	M	8000
2	C	8000
2	M.	9000.

Nested for loop.
Deptno. 25
Job. 1000
faster
40times

- There is no upper limit on the number of Columns in group by clause.

```
-select ------
 group by country, state, city;
```

- if you have large number of columns in groupby clause, then the select statement will be slow(that much sorting has to take place)

- select job, deptno, sum(sal) from emp. group by job, dept no;

output:-

JOB	Dept No.	sum(SAL).
C	1	10000
C	2	8000
M	1	8000
M	2	9000.

Job 1000
deptno. 25
slower.

- select deptno, job, sum(sal) from emp.
  group by job, deptno.
  - the position of column is select clause, and the order of columns in group by clause need not be the same
- the position of columns in select clause, will determine the position of columns in output (this you will write as per user requirements).
- The order of columns in groupby clause will determine the sorting order, the grouping order, the summation order, and hence the speed of processing.

- select ......
  group by city, country, district, state; ← slow

- select
  groupby country, state, district, city; ← fast

having:

- select dept no, sum(sal) from emp
  group by dept no
  having sum(sal)>17000;
  output dept no
  - having clause works after the summation is done.

- select dept no, sum(sal) from emp
  group by dept no
  order by sum (sal):
  - order by clause is the last clause in select statement.

#Oracle-

- select sum(sal) from emp
  group by dept no;

- select max(sum(sal)) from emp ← nesting of group fun
  group by deptno; is allowed in oracle only.

- (select sum(sal) my sql sum_sal from emp group by deptno) as tempp;

- select max (sum_sal) from (select sum(sal) sum_sal fromemp group by deptno) astemp;

<u>output</u>: Max(sum(sal))

18000

Joints (v.v.Imp).

EMP.1(previous). (yesterday)

DEPT

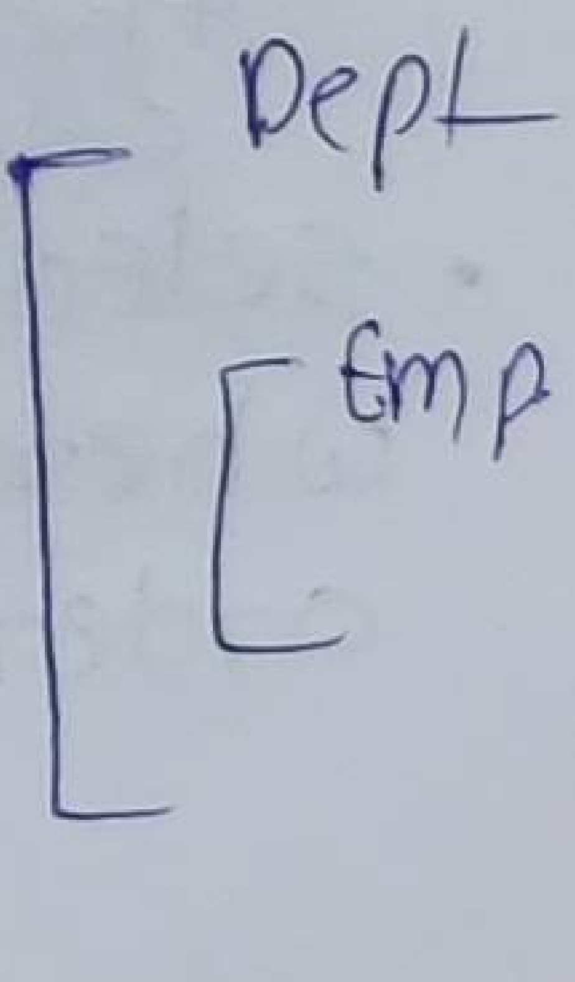
DEPT NO.	DNAME.	LOC.
1	TRN	Bby.
2	EXP	Dlh
3	MKTG	Cal.

- Select ename, deptno from emp; (itispossible) if we want to see ename with dname we needto join two table.
  #Data redundancy:- unnecessary duplication of data(and thatis wastage of HD space).
  # Normalisation:- (will cover later)
- All the data is not stored in one table, data is stored in multiple tables
- to view the columns of 2 or more table, then you will have to write a join.
- select dname, ename from emp, dept where dept.deptno =emp.deptno;

O/P = Dname Ename

dept → driving table

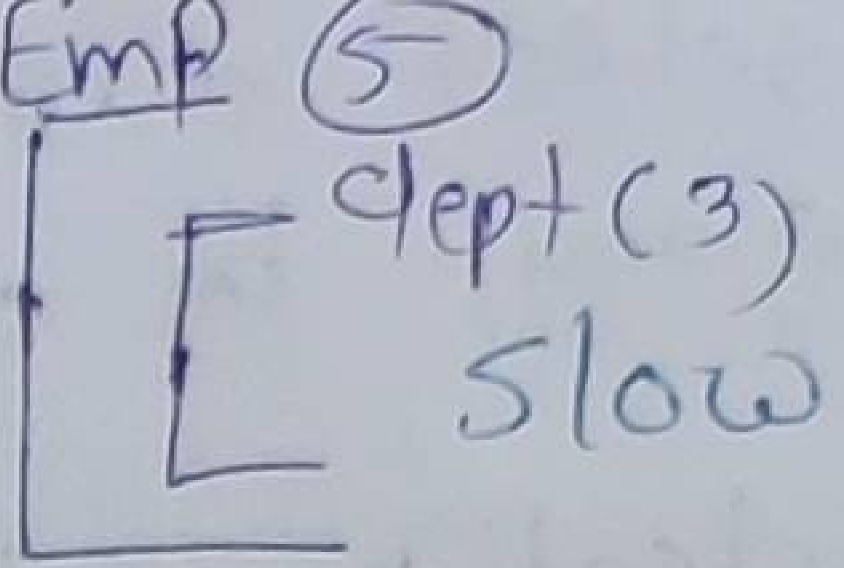
emp → driven table

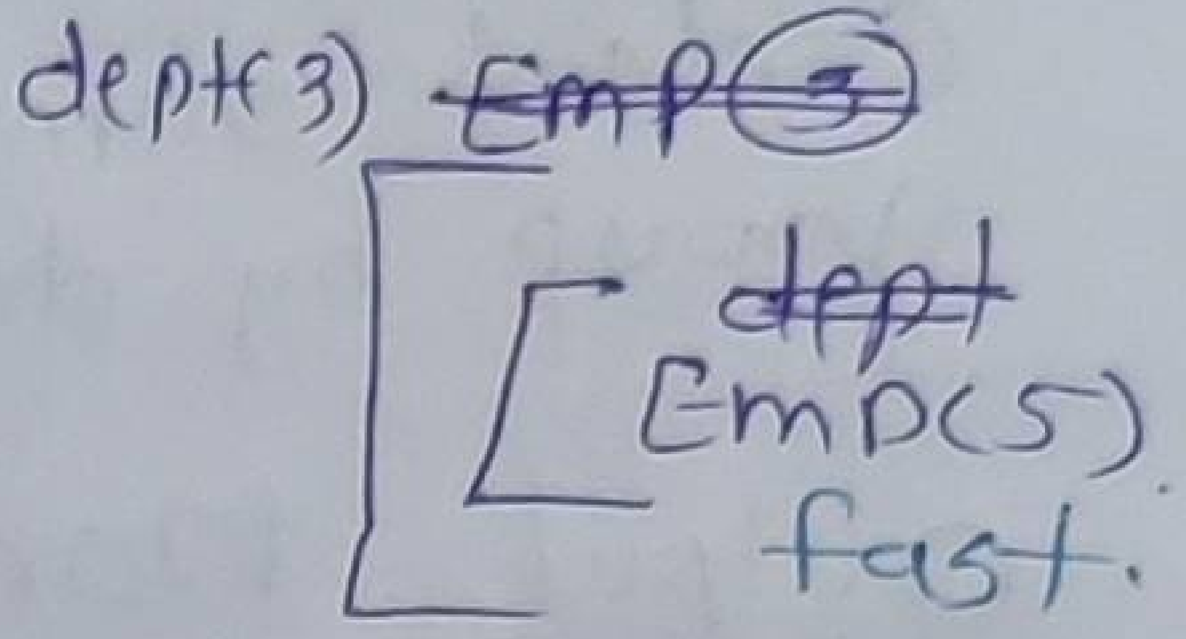


full table scan

Output:

DNAME	ENAME
TRN	Arun
TRN	Ali
TRN	Kiran
EXP	Jack
EXP	Thomas.





- select ename, dname from emp, dept where dept.deptno = emp.deptno.
- select dname, ename from emp, dept Where ~~d~~emp. deptno = dept.deptno;
- select dname, ename from dept, emp. where dept.deptno = emp. ~~deptno~~; deptno; ← slow
- select dname, ename from emp, dept where dept.deptno = emp. deptno; ← fast.

  - In order for your JOIN to work faster, preferably the Driving Table should be table with lesser number of rows.
  - table with less rows → driving
  - table with more rows → driven.

- select dname, ename from emp, dept where dept. dept no = emp. deptno Order by 1;

- select dname, loc, ename, job, sal from emp, dept where dept. dept no = emp .deptno. order by 1;
  - The common column that is present in both the tables, that column's name need not be the same in both the tables; because the same column may have different meaning in same table.
- select dname, loc, ename, job, sal from emp, dept where dept. x = emp. y order by 1;

```
• select * from emp, dept
  where dept. dept no = emp.dept no;
```

```
• select dept.deptno, dname, loc, empno, ename, job, sal
  from emp, dept
  where dept.dept = emp.deptno;
  order by 1;
```

```
• select dept.deptno, dept.dname, dept.loc,
  emp.empno, emp.ename, emp.job, emp.sal ~~from~~
  from emp, dept
  where dept.deptno = emp.deptno;
  order by 1;
```

(good Practice)

```
• select deptno, dname, loc,empno,ename,job,sal
  from emp, dept
  where dept.dept = emp.deptno;
```

Error.

```
• select dept.deptno, dept.dname, dept.loc,
  emp.empno, emp.ename, emp.job, emp.sal
  from emp, dept
  where dept. deptno = emp.deptno
  order by 1;
```

```
• select dname, sum(sal) from emp, dept
  where dept.deptno = emp.deptno
  group by dname;
```

Output:

```
DNAME     SUM(SAL)
TRN       18000
EXP       17000
```

```
• select deptno, sum(sal) from emp
  group by dept no;
```

```
Dept no    sum(SAL)
1          18000
2          17000.
```

```
• select upper(dname), sum(sal) from emp, dept
  where dept. deptno = emp. deptno
  group by upper(dname)
  having sum(sal) >10000
  order by 1;
```

```
• select dname, sum(sal) from emp, dept
  where dept. deptno = emp. deptno
  group by dname;
```

Types of Joins (5): -

```
• select dname, ename from emp, dept
  where dept. deptno = emp. deptno;
```

Output:

```
    DNAME        ENAME
    TRN          Arun
    TRN          Ali
    TRN          kiran
    EXP          Jack
    EXP          Thomas
```

```
• select dname, ename from emp, dept (inequijoin)
  where dept. deptno != emp. deptno;
```

① Equijoin (Natural join) rarely (>, <)

- join base of equality condition
- shows matching rows of both the tables.
- uses:
  a. View the columns of both the tables.
  e.g. Dname and ename, Custname and orderdetails.
- This is most frequenty used join, Hence it is also
- known as Natural join (>90%)

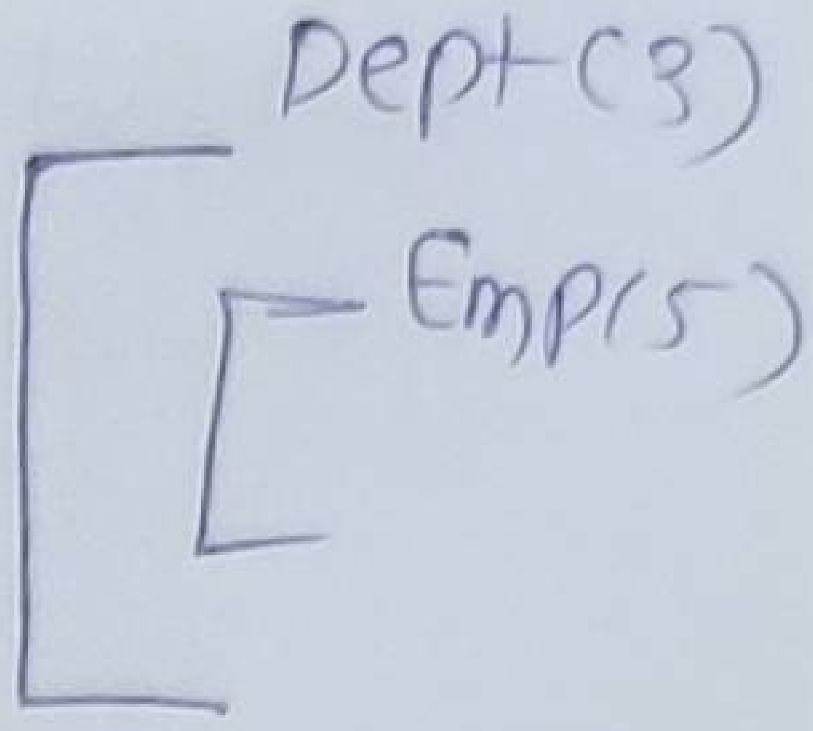
```
• select dname, ename from emp, dept
  where dept. dept no != emp. deptno;
```

② (Non-Equijoin) inequijoin:

- join based on inequality condition.
- shows non-matching rows of both the tables.

Output:

~~DNAME~~	ENAME	DNAME
~~2~~	Jack	TRN
~~2~~	Thomas	TRN
	Arun	EXP
	Ali	EXP
~~1~~	kiran	EXP
	Arun	MKTG
~~1~~	Ali	MKTG
~~2~~	kiran	MKTG
~~2~~	Jak	MKTG
	Thomas	MKTG



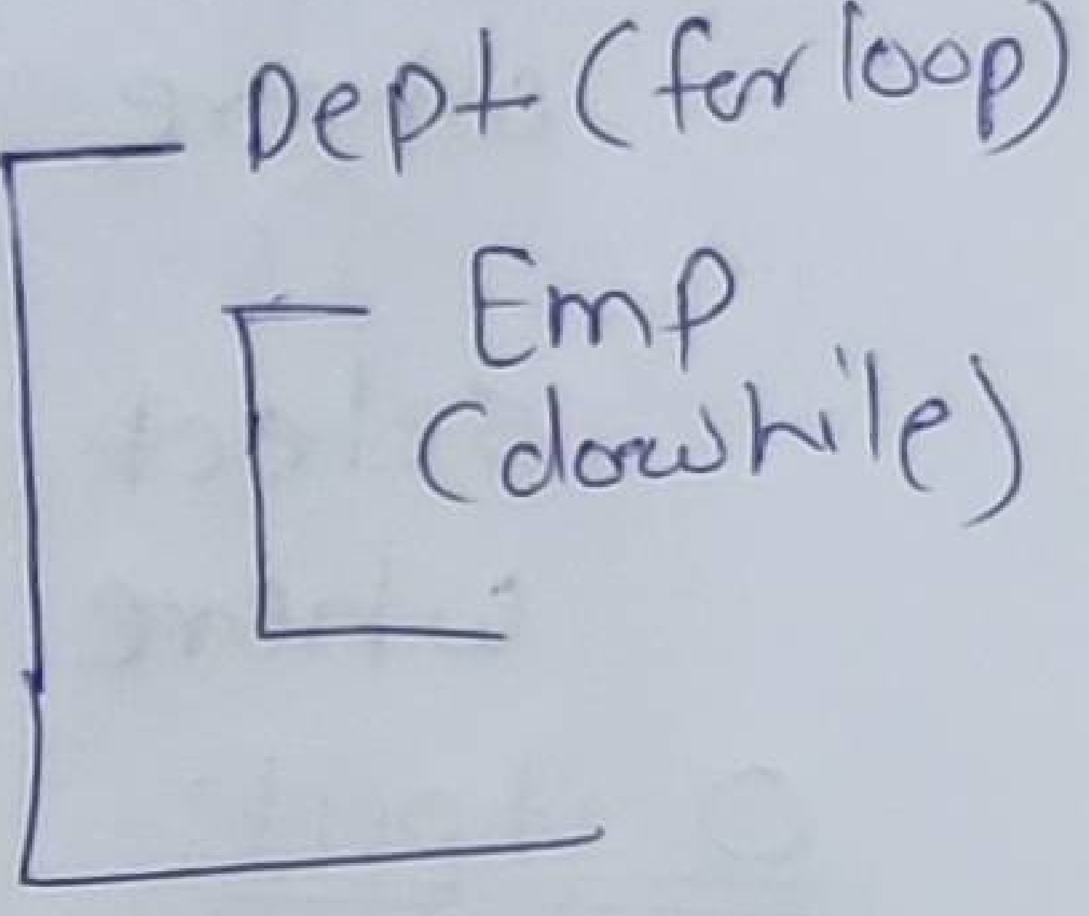
- uses:-
  a. Exception Reports

③ Outer join.

- join with (+) sign (in oracle)
- select dname, ename from emp, dept (right OJ)
  where dept. deptno = emp. dept no (+);

Output:-

DNAME	ENAME
TRN	Arun
TRN	Ali
TRN	kiran
EXP	Jack
EXP.	Thomas
MKTG.	



- It shows the matching rows of both the tables plus non-matching rows of "OUTER" Table.
- outer table → table which is on outer side of (+)
- outer table → table which is on opposite side of (+)
- select dname, ename from emp, dept
  where dept.dept no (+) = emp.dept no; (left OJ)

Dept table → parent / master table.

~~Ch~~ Emp table → child / details table.

Uses:-

a. Master-child / master-detail child relation

```
output:  DNAME      ENAME.
         TRN        Arun
         TRN        Ali
         TRN        kirun
         EXP        Jack
         EXP        Thomas
         .          SCOTT. (add to Emp table)
```

Types:

a. Half outer join.

- (+) sign is on any oneside i.e. L.HS or R.H.S.

1. left outer join
2. right outer join

b. Full outer join

- full outor join
- select dname, ename from emp, dept.
  where dept.dept no = emp.dept no (+)
  Union
  select dname, ename from emp, dept
  where dept.depno(+) = emp.deptno ;

```
Output:  DNAME      ENAME.
         TRN        Arun
         TRN        Ali
         TRN        kirun
         Exp        Jack
         Exp        Thomas
         MKTG       .
         .          SCOTT.
```

- shows matching rows of both the tables
  plus
  shows non-matching rows of both the tables.
- based on nested Do-While loop

ANSI syntax for full outer join:-

- select dname, ename from emp full outerjoin dept on (dept. deptno = emp. deptno);

ANSI syntax for right outer join:-

- select dname, ename from emp ~~full~~ right outerjoindept on (dept.deptno = emp.deptno);

ANSI syntax for left outer join:-

- select dname, ename from empleft outerjoin dept on (dept.deptno = emp. deptno);

- (+) sign for outerjoin issupported by oracle.
- (+) sign for outer join is not supported by any RDBMS.
- My SQL:
- ANSI syntax for Right outer join:-
- supported by all RDBMS including MySQL.
- ANSI syntax for left outerjoin:-
- supported by all RDBMS including MySQL.
- No full outerjoin in MySQL (ANSI syntax).

To achieve full outerjoin in MySQL:-

- select dname, ename from emp right outer join dept on (dept.deptno = emp.deptno)
  Union
  select dname, ename from emp left outer join dept on (dept.deptno = emp.deptno);

④ INNER JOIN:

- Do not mention this join in interviews unless explicitly by interviewer.
- by default every join is an Inner join putting a (+) sign is what makes it an outerjoin

SQL Exe 1 to 5

SQL ass 1 to10.

Snehal Sawant