4. Cartesion join

* join without a where condition
* every row of driving table is combied with each and every row of driven table
* it is also known as cross join

```
select dname,ename from emp,dept;<-FAST
select dname,ename from dept,emp;<-Slow
```

```
 mysql> select dname,ename from emp,dept;
+-------+--------+
| dname | ename  |
+-------+--------+
| MKTG  | arun   |
| EXP   | arun   |
| TRN   | arun   |
| MKTG  | ali    |
| EXP   | ali    |
| TRN   | ali    |
| MKTG  | kiran  |
| EXP   | kiran  |
| TRN   | kiran  |
| MKTG  | jack   |
| EXP   | jack   |
| TRN   | jack   |
| MKTG  | Thomas |
| EXP   | Thomas |
| TRN   | Thomas |
+-------+--------+
```

Parent column:EMPNO
Child column:MGR--telling the value of EMPNO

```
CREATE table emp(empno int(4),ename varchar(20),sal int(10),deptno int(4),job
VARCHAR(20),mgr VARCHAR(20));
insert into emp values(1,'arun',8000,1,'M','4');
insert into emp values(2,'ali',7000,1,'C','1');
insert into emp values(3,'kiran',3000,1,'C','1');
insert into emp values(4,'jack',9000,2,'M',null);
insert into emp values(5,'Thomas',8000,2,'C',4);
mysql> select * from emp;
+-------+--------+------+--------+------+------+
| empno | ename  | sal  | deptno | job  | mgr  |
+-------+--------+------+--------+------+------+
|     1 | arun   | 8000 |      1 | M    | 4    |
|     2 | ali    | 7000 |      1 | C    | 1    |
|     3 | kiran  | 3000 |      1 | C    | 1    |
|     4 | jack   | 9000 |      2 | M    | NULL |
|     5 | Thomas | 8000 |      2 | C    | 4    |
+-------+--------+------+--------+------+------+
```

```
select emp.name,emp2.ename from emp,emp2 where emp.empno=emp2.mgr;
```

5.self Join

* joiing a table to itself
* used when parent column and child column both are present in the same TABLE
* slowest join
* based on recursion
* all joins are slow but self join are slow

Q. Select a.ename,b.ename from emp b, emp a where a.mgr=b.empno;
table A and B are TEMPORARY TABLES which is better than duplicating data
table A contains

```
Ename and MGR
ARUN        4
ALI         1
KIRAN       1
jack        -
THOMAS      4

table B contains
Ename and EMPNO
ARUN        1
ALI         2
KIRAN       3
Jack        4
THOMAS      5


driving table==A
driveen table==B
any comprasion done with null return null.


mysql> Select a.ename,b.ename from emp b, emp a where a.mgr=b.empno;

+--------+-------+
| ename  | ename |
+--------+-------+
| arun   | jack  |
| ali    | arun  |
| kiran  | arun  |
| Thomas | jack  |
+--------+-------+


-------------------------------------------------------------------------------------------------
Q. this very rarely used but always asked in invterview.
-------------------------------------------------------------------------------------------------

Select dname,ename from emp e,dept d where d.deptno=e.deptno;
mysql> Select dname,ename from emp e,dept d where d.deptno=e.deptno;
+-------+--------+
| dname | ename  |
+-------+--------+
| TRN   | arun   |
| TRN   | ali    |
| TRN   | kiran  |
| EXP   | jack   |
| EXP   | Thomas |
+-------+--------+
Dont give aliace name unnecessary...becuase it load whole table in RAM.
it will make slow execution other system and own system also;
* when you specify an alias for tablename, a copy of the table is brought into the server RAM
* Do not specify an alias for tablename,unnecessarily,becuase not only will yout select
statement
  be slow but you will slow down the select statement of other users.
* specify an alias for tablename only if you are writing a self join
* Cartesian join is the fastest join because there is no where clause, and hence no
searching is involved



================================================================================
      joining 3 or more tables
================================================================================
CREATE table depthead(deptno int(4),dhead varchar(20));
insert into depthead values(1,'Arun');
insert into depthead values(2,'Jack');
                                (5)    (3 )        (2)
select dname,ename,ehead from emp,dept,depthead  where depthead.deptno=dept.deptno and
dept.deptno=emp.deptno;

 DNAME ENAME DHEAD
```

```
 ----------------
 * it internally join two TABLE
 * based on binary logic conditions(logical and suppied)

mysql> select dname,ename,dhead from emp,dept,depthead  where depthead.deptno=dept.deptno
and dept.deptno=emp.deptno;
+-------+--------+-------+
| dname | ename  | dhead |
+-------+--------+-------+
| TRN   | arun   | Arun  |
| TRN   | ali    | Arun  |
| TRN   | kiran  | Arun  |
| EXP   | jack   | Jack  |
| EXP   | Thomas | Jack  |
+-------+--------+-------+



================================================================================================
     Types of Relationship
================================================================================================
Types of Relationship
1 to 1 replationship(DEPT: DEPTHEAD)
1 to m relationship(Dept:EMP) (Depthead:emp)
m to 1 relationship(Emp: Dept) and (EMP: Depthead)
m to m relationship(EMP: Projects) or(Projects:EMP)

CREATE table projects(projno varchar(4),clientname varchar(20),Proj_dtls varchar(20));
insert into projects values('P1','Arun','CGS');
insert into projects values('P2','morgan stanley','AMS');
insert into projects values('P3','bnp paribas','Macros Dev');
insert into projects values('P4','ICICI Bank','PPS');
insert into projects values('P5','AMFI','Website DEV');

mysql> select * from projects;
+--------+----------------+-------------+
| projno | clientname     | Proj_dtls   |
+--------+----------------+-------------+
| P1     | Arun           | CGS         |
| P2     | morgan stanley | AMS         |
| P3     | bnp paribas    | Macros Dev  |
| P4     | ICICI Bank     | PPS         |
| P5     | AMFI           | Website DEV |
+--------+----------------+-------------+

* based on set theory
* many to many relationship exist in intersection TABLE


--------------------------------------------------------------------------
following Intersection TABLE
--------------------------------------------------------------------------
CREATE table projects_emp(projno varchar(4),empno int(4));
insert into projects_emp values('P1',1);
insert into projects_emp values('P3',3);
insert into projects_emp values('P4',4);

mysql> select * from projects_emp;
+--------+-------+
| projno | empno |
+--------+-------+
| P1     |     1 |
| P3     |     3 |
| P4     |     4 |
+--------+-------+

mysql> select clientname, proj_dtls,ename from projects_emp,emp,projects
    -> where projects.projno=projects_emp.projno
    -> and emp.empno=projects_emp.empno
    -> order by 1,2,3;
+------------+-----------+-------+
```

```
| clientname   | proj_dtls   | ename |
+--------------+-------------+-------+
| Arun         | CGS         | arun  |
| bnp paribas  | Macros Dev  | kiran |
| ICICI Bank   | PPS         | jack  |
+--------------+-------------+-------+
```

Sub Queries

* also known as Nested queries(Query within Query)
  (Select Within Select)
* default max upto 255 levels for sub-Queries.
* limit of SQL can be exceeded with the help of views
* less of number execution means faster.
* join is faster than subQuery(more number of select statement will be slower the execution) but 1 exception is here.
* we use can use sub Queries with DML cmds..insert delete update

```
mysql> select * from emp;
+-------+--------+------+--------+------+------+
| empno | ename  | sal  | deptno | job  | mgr  |
+-------+--------+------+--------+------+------+
|     1 | arun   | 8000 |      1 | M    | 4    |
|     2 | ali    | 7000 |      1 | C    | 1    |
|     3 | kiran  | 3000 |      1 | C    | 1    |
|     4 | jack   | 9000 |      2 | M    | NULL |
|     5 | Thomas | 8000 |      2 | C    | 4    |
+-------+--------+------+--------+------+------+
```

```
Select ename, min(sal) from emp;
******error in oracle and works in MySQL-output in meaningleass-----
mysql> Select ename, min(sal) from emp;
+-------+----------+
| ename | min(sal) |
+-------+----------+
| arun  |     3000 |
+-------+----------+
wrong output
```

```
select ename from emp where sal=min(sal);
--->error----min(sal)group cannot use in where Stement
```

Q.Display the ename who is receving sal=min(sal);

```
select ename from emp
where sal=(Select min(sal) from emp);
+-------+
| ename |
+-------+
| kiran |
+-------+
correct output
```

```
step 1: Select min(sal) from emp==3000...this will execute because() has higher priority.
step 2: select ename from emp where sal=(3000)...paramter 3000 pass at runtime
step 3: ename kiran.....sal=3000.
```

```
Main-Query--Parent Query---Outter Query----select ename from emp where sal=();(****)higher
priority
Sub-QUery--child Query---inner Query---Select min(sal) from emp;
```

.....Interview Question....
Q display 2nd largest Max(sal)?
Q. display 3rd largest Max(sal)?
Q. display 2nd lowest Min(sal)?
Q. display 3rd lowest Min(sal)?

Q. Interview Question....display 2nd largest Max(sal)?

```
select max(sal) from emp where sal< (select max(sal) from emp);

mysql> select max(sal) from emp
    -> where sal<
    -> (select max(sal) from emp);
+----------+
| max(sal) |
+----------+
|     8000 |
+----------+
```

Q. Display the rows who belog to the same Deptno as 'Thomas'.

```
step 1: select deptno from emp where ename='thomas';
setp 2: select * from emp where deptno=();

select * from emp where deptno=(select deptno from emp where ename='thomas');
+-------+--------+------+--------+------+------+
| empno | ename  | sal  | deptno | job  | mgr  |
+-------+--------+------+--------+------+------+
|     4 | jack   | 9000 |      2 | M    | NULL |
|     5 | Thomas | 8000 |      2 | C    | 4    |
+-------+--------+------+--------+------+------+
```

Q. Display the rows who belog not to the same Deptno as 'Thomas'.

```
mysql> select * from emp where deptno!=(select deptno from emp where ename='thomas');
+-------+-------+------+--------+------+------+
| empno | ename | sal  | deptno | job  | mgr  |
+-------+-------+------+--------+------+------+
|     1 | arun  | 8000 |      1 | M    | 4    |
|     2 | ali   | 7000 |      1 | C    | 1    |
|     3 | kiran | 3000 |      1 | C    | 1    |
+-------+-------+------+--------+------+------+
```

Q. input name=kiran...Diplay all rows of who are doing the same job the kiran

---

```
step 1: select job from emp where ename='kiran';
setp 2: select * from emp where job=();

mysql> select * from emp where job=(select job from emp where ename='kiran');
+-------+--------+------+--------+------+------+
| empno | ename  | sal  | deptno | job  | mgr  |
+-------+--------+------+--------+------+------+
|     2 | ali    | 7000 |      1 | C    | 1    |
|     3 | kiran  | 3000 |      1 | C    | 1    |
|     5 | Thomas | 8000 |      2 | C    | 4    |
+-------+--------+------+--------+------+------+
```

```
================================================================================================
 Update using SubQuery
================================================================================================

works in Oracle:
--------works in oracle only-------------------------------------

delete from emp
where deptno=(select deptno from emp where ename='Thomas');

update emp set sal=10000
where job=(select job from emp where ename='kiran');
------------works in oracle only------------------------------



* in MySQL in You cannot allwed  deleting while reading cmd.

* in MysQL you cannot update or delete from a table from which you are
  currenlty selecting



Solution:
delete from emp
where deptno=
(select tempp.deptno from
(select deptno from emp
where ename='Thomas') as tempp);  <-----AS keyword is optional

update emp set sal=10000
where job=
(select tempp.job from
(select job from emp
where ename='kiran') as tempp);

step1: select deptno from emp where ename='Thomas' output deptno 2
         this deptno 2 stores value in temp table

step2:




================================================================================================
   Multi row sub-queries ...sub query returns multiple rows
================================================================================================

------------------------------------------------------------------------------------------------
 Q.Display all the rows that receiving a SAL equal to any of the Managers.
------------------------------------------------------------------------------------------------


* i want see one of the manager salary (8000or 9000)salary of manager


select * from emp
where sal=any <-------------------(8000 or 9000)
(select sal from emp where job='M');

* any is special operator it will perfom logical OR operation. work in MySQL and oracle


mysql> select * from emp
       where sal=any
      (select sal from emp where job='M');
+-------+--------+------+--------+------+------+
| empno | ename  | sal  | deptno | job  | mgr  |
+-------+--------+------+--------+------+------+
```

```
|      1 | arun    | 8000 |      1 | M    | 4    |
|      4 | jack    | 9000 |      2 | M    | NULL |
|      5 | Thomas  | 8000 |      2 | C    | 4    |
+--------+---------+------+--------+------+------+
```

Solution 2:

```
select * from emp
       where sal in
      (select sal from emp where job='M');
+--------+---------+------+--------+------+------+
| empno  | ename   | sal  | deptno | job  | mgr  |
+--------+---------+------+--------+------+------+
|      1 | arun    | 8000 |      1 | M    | 4    |
|      4 | jack    | 9000 |      2 | M    | NULL |
|      5 | Thomas  | 8000 |      2 | C    | 4    |
+--------+---------+------+--------+------+------+
```

Q. To exclude the managers from the output:

```
select * from emp
       where job !='M' and sal in
      (select sal from emp where job='M');
+--------+---------+------+--------+------+------+
| empno  | ename   | sal  | deptno | job  | mgr  |
+--------+---------+------+--------+------+------+
|      5 | Thomas  | 8000 |      2 | C    | 4    |
+--------+---------+------+--------+------+------+
```

* An-Logical OR
* IN-LOgical OR
* All-operator will perfom logical AND

To make it work faster:

1. join is faster than sub-query; therefore use join wherever possible
2. Try to reduce the number of leveles for sub-queries
2. Try to reduce the number of rows returned by sub-query

==========

Assumption 3rd row SAL is 13000:

```
update emp set sal=13000 where empno=3;
insert into emp values(3,'kiran',3000,1,'C','1');
```

Problem:

```
select * from emp
where sal>
(select sal from emp
where job='M');
```

error: subQuery give two value 8000,9000 hence error
i want perofom logical OR operation 9000.

==========

ALL OPERATOR

==========

```
select * from emp
where sal>all
(select sal from emp
where job='M');  <------slower
```

```
+-------+-------+-------+--------+------+------+
| empno | ename | sal   | deptno | job  | mgr  |
+-------+-------+-------+--------+------+------+
|     3 | kiran | 13000 |      1 | C    | 1    |
+-------+-------+-------+--------+------+------+
```

```
select * from emp
where sal>all
(select max(sal) from emp
where job='M');  <-----faster
```

```
+-------+-------+-------+--------+------+------+
| empno | ename | sal   | deptno | job  | mgr  |
+-------+-------+-------+--------+------+------+
|     3 | kiran | 13000 |      1 | C    | 1    |
+-------+-------+-------+--------+------+------+
```

==========

Using Sub-Query in the Having clause

==========

Assumption 3rd row SAL is 3000:

```
CREATE table emp(empno int(4),ename varchar(20),sal int(10),deptno int(4),job
VARCHAR(20),mgr VARCHAR(20));
insert into emp values(1,'arun',8000,1,'M','4');
insert into emp values(2,'ali',7000,1,'C','1');
insert into emp values(3,'kiran',3000,1,'C','1');
insert into emp values(4,'jack',9000,2,'M',null);
insert into emp values(5,'Thomas',8000,2,'C',4);
```

Works only Oracle:

Q. Display the Dname that is having max(sum(Sal));

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

```
+--------+----------+
| deptno | sum(sal) |
+--------+----------+
|      1 |    18000 |
|      2 |    17000 |
+--------+----------+
```

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

```
+----------+
| sum(sal) |
+----------+
|    18000 |
|    17000 |
+----------+
```

==========

```
----------work in oracle only------------
select max(sum(sal)) from emp
group by deptno;

Max(Sum(sal))
--------
18000
----------work in oracle only------------
```

Solution in MySQL:

```
select max(sum_sal) from
(select sum(sal) sum_sal from emp
group by deptno)as tempp;

+--------------+
| max(sum_sal) |
+--------------+
|        18000 |
+--------------+
```

explaination:

```
Step 1: select sum(sal) sum_sal from emp group by deptno;
+---------+
| sum_sal |
+---------+
|   18000 |
|   17000 |
+---------+
Step 2: select max(sum_sal) from (step 1 SubQuery)as tempp;
```

==========

----------work in oracle only----------

```
Select deptno,sum(sal) from emp
group by deptno
having sum(sal)=
(select max(sum(sal))from emp group by deptno);

deptno       sum(sal)
--------   ----------
  1           18000
----------work in oracle only------------------------------
```

Solution in MySQL:

```
Select deptno,sum(sal) from emp
group by deptno
having sum(sal)=
(select max(sum_sal) from
(select sum(sal) sum_sal from emp
group by deptno)as tempp);

+--------+----------+
| deptno | sum(sal) |
+--------+----------+
|      1 |    18000 |
+--------+----------+
```

explaination:

```
Step 1: select sum(sal) sum_sal from emp group by deptno;
+---------+
| sum_sal |
+---------+
|   18000 |
|   17000 |
+---------+
Step 2: select max(sum_sal) from (step 1 Sub Query===18000,17000);

step 3:Select deptno,sum(sal) from emp group by deptnohaving sum(sal)=(step 2 sub
Query==18000)
```

```
================================================================================
----------work in oracle only-----------------------------
select dname,sum(sal) from emp,dept
where dept.deptno=emp.deptno
group by dname
having sum(sal)=
(select max(sum(sal))from emp group by deptno);


Dname     Sum(sal)
----------------------
TRN       18000
----------work in oracle only-----------------------------


Solution in MySQL:

select dname, sum(sal) from emp,dept
where dept.deptno=emp.deptno
group by dname
having sum(sal)=
(select max(sum_sal) from
(select sum(sal) sum_sal from emp
group by deptno)as tempp);

+-------+----------+
| dname | sum(sal) |
+-------+----------+
| TRN   |    18000 |
+-------+----------+



explaination:

Step 1: select sum(sal) sum_sal from emp group by deptno;
+---------+
| sum_sal |
+---------+
|   18000 |
|   17000 |
+---------+
Step 2: select max(sum_sal) from (step 1 Sub Query===18000,17000);

step 3:select dname, sum(sal) from emp,dept where dept.deptno=emp.deptno
      group by dname having sum(sal)==(step 2 Sub Query===18000)
================================================================================
=
```