Nested query

1. simple nested query
    a. in these types of queries, child query gets executed only once and then outer query will get executed
    b. If the child query is not dependent on outer query, then it is called as simple nested query
2. co-related query
    a. In these types of queries, child query gets executed once for each row in the outer query table.
    b. If child query is dependent on outer query, then it is called as co-related query.

1. find all employees whose salary > smith's salary

Select sal from emp Where ename='SMITH' ------→ 2000	Select * from emp Where salary>2000
Select * from emp Where salary>(Select sal from emp Where ename='SMITH')	

1. find all employees who works under same mgr as allen's mgr

Select mgr from emp Where ename='Allen'------→7698	Select * From emp Where mgr=7698
Select * From emp Where mgr=(Select mgr from emp Where ename='Allen')	

2. find all employees who are working in smith's or allen department
   select * from emp
   where deptno in (select deptno

   from emp

   where ename in ('SMITH','allen'))

3. find all employees who works in Research department.
   select * from emp
   where deptno = (select deptno
   from dept
   where dname='research');

4. find all employees whose sal >both smith's and allen's salary
   where sal >all (2000,3000) 2100 will not be in the o/p
   where sal >any (2000,3000) 2100 will be in the o/p

select * from emp

where sal > all (select sal

from emp

where ename in ('SMITH','allen')

5. Write all queries to find all employee who are working under allen's mgr

```
select * from emp
where mgr=(select mgr from emp
where ename='Allen')
```

6. find all employees with sal < avg salary of accounting department.

```
select * from emp

where sal<(select avg(sal) from emp where deptno =(select deptno
from dept
where dname='accounting'));
```

7. list all employees who works in department, which has min number of employees

```
select *
from emp
where deptno=(select deptno
 from emp
group by deptno
order by count(*)
limit 1);
```

8. list all employees whose salary > max salary of allen's department

```
select * from emp
where sal>(select max(sal) from emp
where deptno=(
    select deptno
    from emp
    where ename='ALLEN'
));
```

9. find all employees with sal > average sal of its own department.

```
select *
from emp e
where e.sal >(select avg(m.sal) from emp m where m.deptno=e.deptno );
```

10. list all employees with sal < avg sal of all employees who works under same mgr

```
select *
from emp e
where e.sal<(select avg(sal) from emp m where m.mgr=e.mgr);
```

11. list all employees with sal > avg sal of all employees who does the same job

```
select *
from emp e
```

where e.sal>(select avg(sal) from emp m where m.job=e.job);

In corelated query usually we use 2 operators

1. exists--→ it will return true if child query returns one or more rows in the o/p, false otherwise
2. not exists---→ it will return true if child query does not return any rows in the o/p, false otherwise.

-------to find all departments which has no employees

```
Select *
From dept d
Where not exists (select * from emp e where e.deptno=d.deptno);
```

-------to find all departments which has employees

```
Select *
From dept d
Where exists (select * from emp e where e.deptno=d.deptno);
```

-----to find all empoyees who are not mgr for any employee

```
Select * from emp e
Where not exists(
      Select * from emp m
Where m.mgr=e.empno)
```

-------to find all emloyees with salary > min sal of either allens department or smith dept.

```
Select * from emp
Where sal >all(Select min(sal)
From emp
Where deptno in (Select deptno from emp
                  Where ename in ('Allen','SMITH')
                  )
Group by deptno
)
```

------to find all employees whose salary is > both miller and Blake salary

```
Select *

From emp

Where sal > all (select sal from emp where ename in ('SMITH','BLAKE'))
```

------to find all employees whose salary is > any one , miller or Blake salary

```
Select *

From emp
```

Where sal > any (select sal from emp where ename in (‘miller’,’BLAKE’))

------find all employees whose sal >= smith’s sal and <= blakes salary

Select *

From emp

Where sal between (select sal from emp where ename=’SMITH’) and (select sal from emp where ename=’BALKE’)

------find all employees whose sal >smith’s sal and < blakes salary

select * from emp

-> where sal between (select sal from emp where ename='smith')+1 and (select sal

-> from emp

-> where ename='blake')-1;

----find all employees who works in accounting department.

Select * from emp

Where deptno=(select deptno from dept where dname=’ACCOUNTING’);

-----find all employees with salary > avg salary of dept ‘Sales’

Select * from emp

Where sal >(select avg(sal) from emp where deptno=(select deptno from dept where dname=’SALES’))

-------to create a table myemp, add all the rows with sal> 2000 from emp table

Create table myemp

As

(select * from emp where sal>2000)

---to add only limited columns

```
Create table myemp
As
(select empno,ename,sal from emp where sal>2000)
```

-------to create a empty table myemp with all the columns same as emp

```
Create table myemp
As
(select * from emp where 1=2)
```

-----add all employees of deptno 10, in to myemp table

```
insert into myemp
select * from emp where deptno=10;
```

------update smiths salary = millers salary

```
Update emp
Set sal=(select sal from (select * from emp) e where ename='MILLER')
Where ename='SMITH'
```

----update sal of all employees who works in Accounting dept to 'blake' sal

```
Update emp
Set sal=(select sal from (select * from emp) e where e.ename='blake')
Where deptno=(select deptno from dept d where dname=' Accounting')
```

---delete all employees who are working in smiths department

```
Delete from emp
Where deptno=(select deptno from (select * from emp) e where ename='SMITH')
```

Course

cid	Cname	Duration	capacity	Rid	fid
1001	DAC	400	240	10	101
1002	DBDA	350	60	12	
1003	DTISS	300	60		102

Faculty

Facid	Fname	spskill
100	Raj	Java
101	Gayatri	C#
102	Rashmi	C++

room

Roomid	Rname	Rloc
10	Lotus	1st floor
11	Rose	1 st floor
12	Mogara	2nd floor

1. Find all faculties, which are available for new course.
   Select * from faculty f
   where not exists(select * from course c where c.fid=f.facid)

2. Find all rooms which are available (not assigned to any course)
   Select * from room r
   Where not exists (select rid from course c where c.rid=r.roomid)

3. Find the course which will be conducted in room with name mogara.

   Select * from course c

   where c.rid=(select roomid from room r where r.rname='mogara')

4. List all courses which are conducted by faculties with name starts with 'R'
   And the duration of the course is >200

   Select * from course c where c.fid in (select facid
   From faculty f where f.fname like 'R%') and c.duration>200;

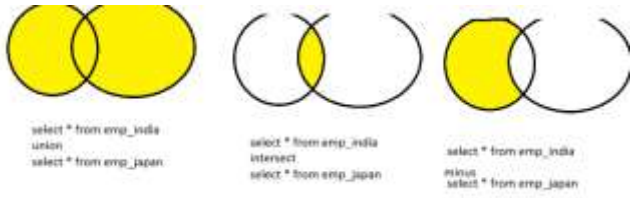
5. List courses with highest capacity for which faculty assigned has sp skill java.
   Select * from course c where
   c.fid=(select facid from faculty f where f.spskill='java')
   order by capacity desc
   limit 1;

6. Find all courses for which room location is 1 st floor and course name starts with D or capacity >200
   Select * from course c where c.rid=(select roomid from room r
   Where Rloc='1 st floor') and c.cname like 'D%' or c.capacity>200

Set operators

These operators are used when you want to combine output of multiple queries

Union all	This will display all the rows from all queries but common rows will be displayed twice	Select * from emp_india Union all Select * from emp_japan Union all Select * from emp_us
Union	This will display all the rows from all queries but common rows will be displayed only once	Select * from emp_india Union Select * from emp_japan Union Select * from emp_us
Intersect	This will display all the rows from all queries which are common rows	Select * from emp_india intersect Select * from emp_japan intersect Select * from emp_us
Minus/except	This will display all the rows which are only in the output of first query	Select * from emp_india minus Select * from emp_japan



1. All the queries should have equal number of columns, and corresponding columns data type and context should match
2. In the output, column names of topmost query will appear.

```
select rid,rname,null custid,null cname from room
where not exists (select * from rbooking r where r.roomno=room.rid)
union
select null,null,custid,cname from customer
where not exists (select * from rbooking r where r.custid=customer.custid);
```