**DBMS LAB ASSIGNMENT-5**

**B NEELAKANTESWAR**

**19bcs020 T12**

1.Illustrate logical ANY, ALL and LIKE operator- the queries should be relevant to your respective databases 3 queries for each operator. One query explaining the difference between ANY and ALL.

ANY:

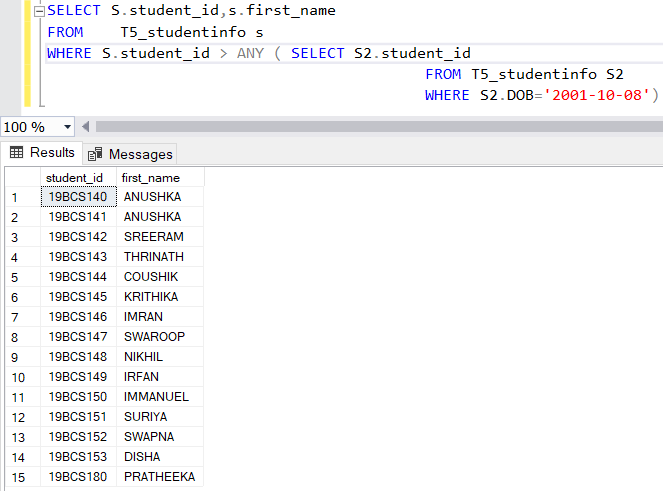
Queries:

1)

*SELECT S.student\_id,s.first\_name*

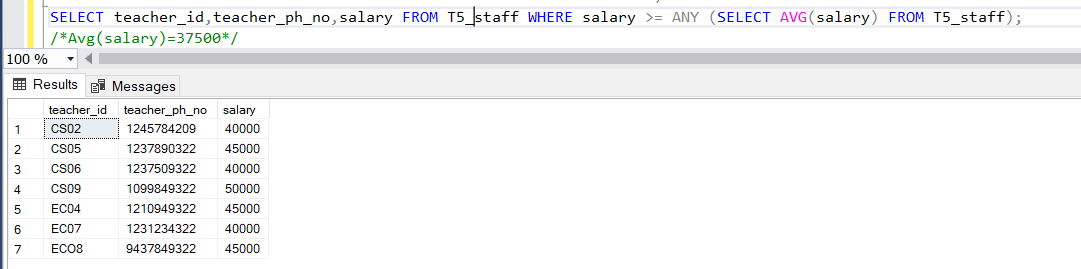
*FROM T12\_studentinfo s*

*WHERE S.student\_id > ANY ( SELECT S2.student\_id FROM T12\_studentinfo S2 WHERE S2.DOB='2001-10-08')*



2)

*SELECT teacher\_id,teacher\_ph\_no,salary FROM T12\_staff WHERE salary >= ANY (SELECT AVG(salary) FROM T12\_staff);*

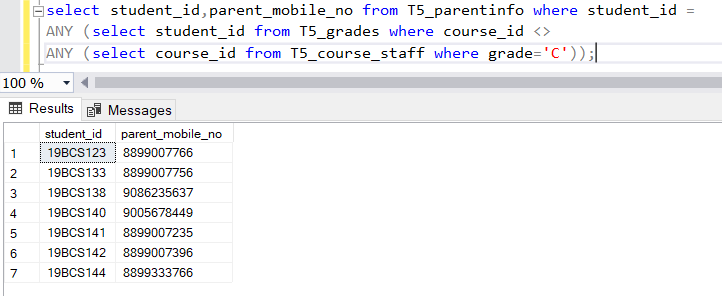


3)

*select student\_id,parent\_mobile\_no from T12\_parentinfo where student\_id =*

*ANY (select student\_id from T12\_grades where course\_id <>*

*ANY (select course\_id from T12\_course\_staff where grade='C'));*

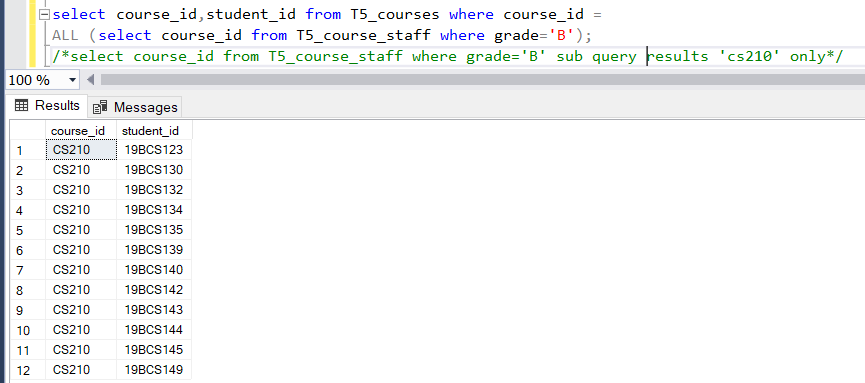


ALL:

Queries:

1)

*select course\_id,student\_id from T12\_courses where course\_id = ALL (select course\_id from T12\_course\_staff where grade='B');*

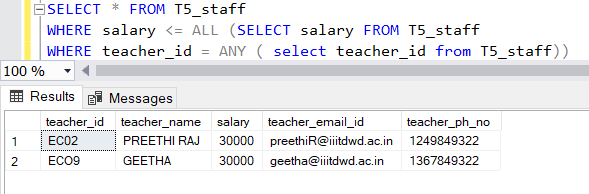


2)

*SELECT \* FROM T12\_staff*

*WHERE salary <= ALL (SELECT salary FROM T12\_staff*

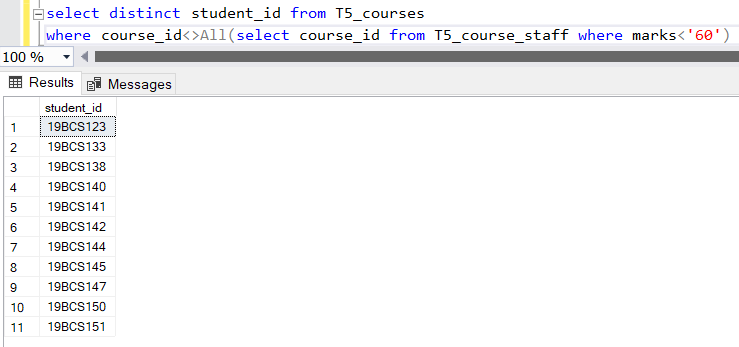
*WHERE teacher\_id = ANY ( select teacher\_id from T12\_staff))*



3)

*select distinct student\_id from T12\_courses*

*where course\_id<>All(select course\_id from T12\_course\_staff where marks<'60')*

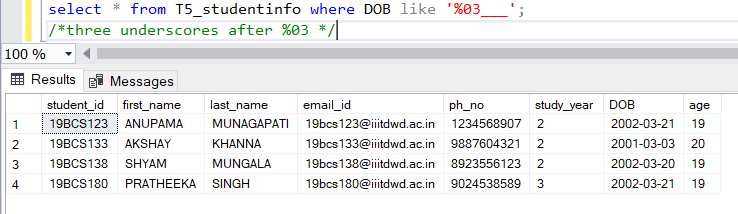


LIKE:

Queries:

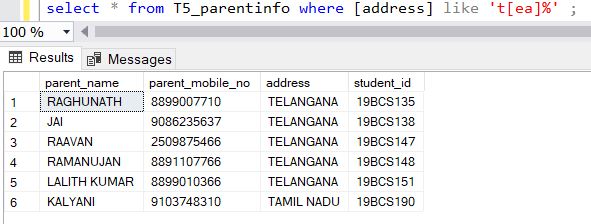
1)

*select \* from T12\_studentinfo where DOB like '%03\_\_\_';*



2)

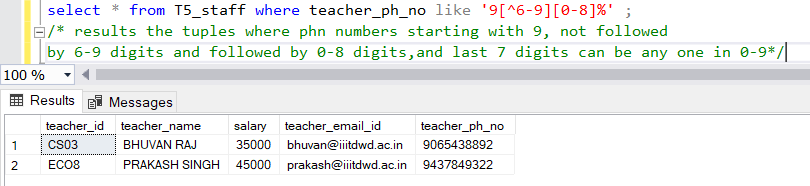
*select \* from T12\_parentinfo where [address] like 't[ea]%' ;*



3)

*select \* from T12\_staff where teacher\_ph\_no like '9[^6-9][0-8]%' ;*

*/\* results the tuples where phn numbers starting with 9, not followed by 6-9 digits and followed by 0-8 digits,and last 7 digits can be any one in 0-9\*/*



Difference between ‘ALL’ and ‘ANY’:

Query:

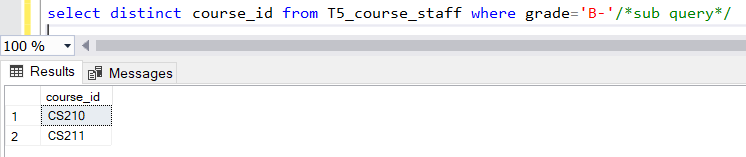
*select distinct course\_id from T12\_course\_staff where grade='B-'/\*sub query\*/*

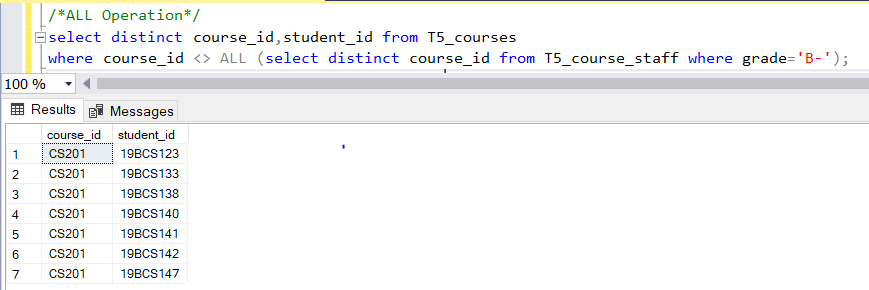
*select distinct course\_id,student\_id from T12\_courses*

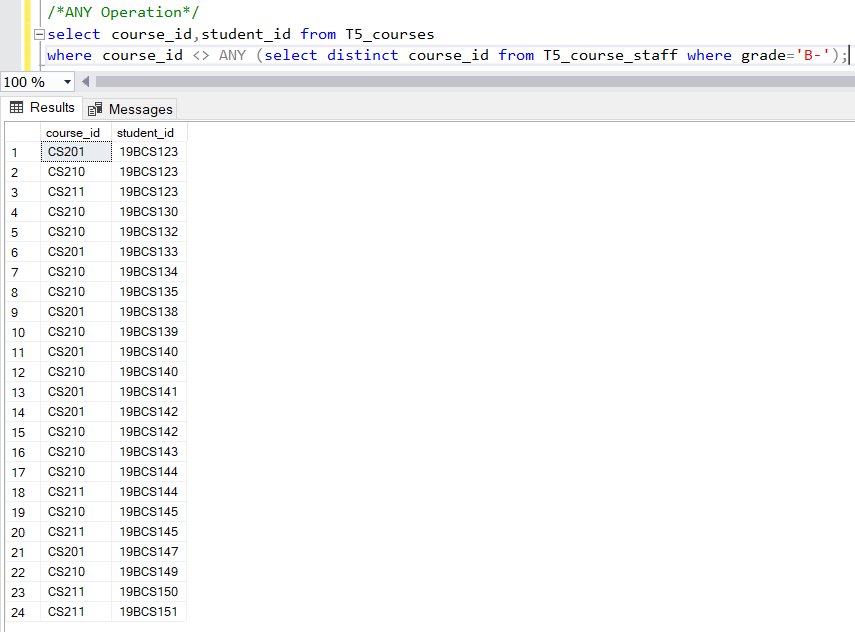
*where course\_id <> ALL (select distinct course\_id from T12\_course\_staff where grade='B-');*

*select course\_id,student\_id from T12\_courses*

*where course\_id <> ANY (select distinct course\_id from T12\_course\_staff where grade='B-');*



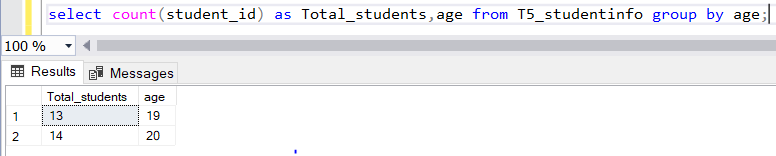




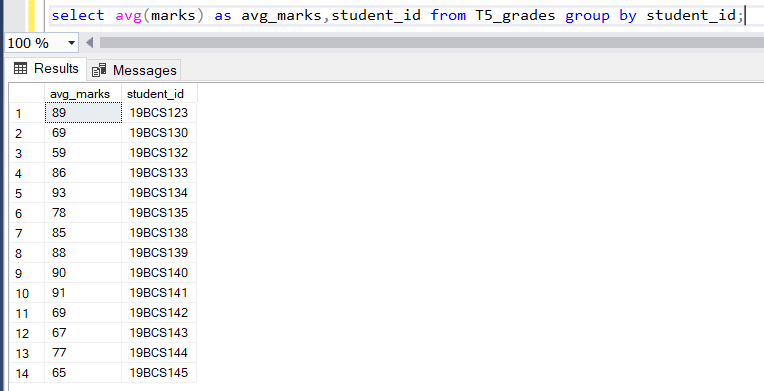
2) One query for each Aggregate function.

Queries:

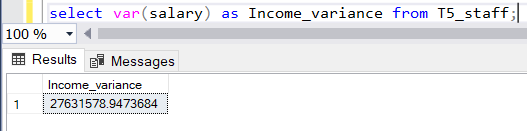
*select count(student\_id) as Total\_students,age from T12\_studentinfo group by age;*



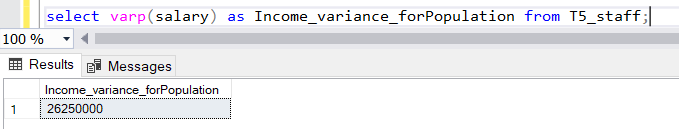
*select avg(marks) as avg\_marks,student\_id from T12\_grades group by student\_id;*



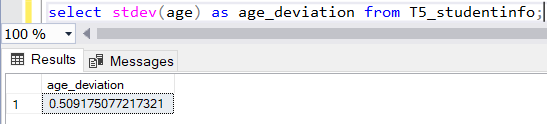
*select var(salary) as Income\_variance from T12\_staff;*



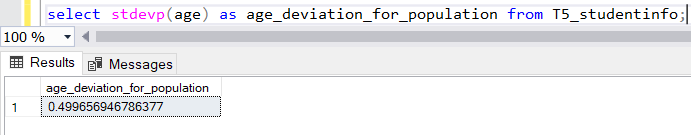
*select varp(salary) as Income\_variance\_forPopulation from T12\_staff;*



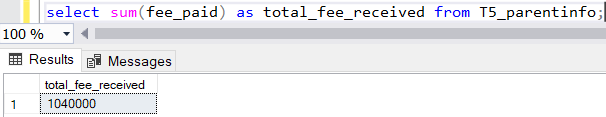
*select stdev(age) as age\_deviation from T12\_studentinfo;*



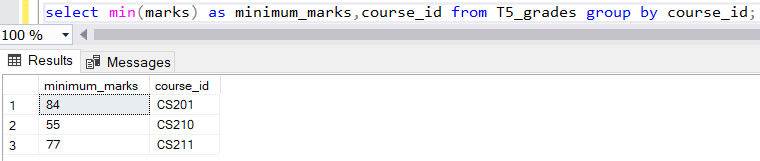
*select stdevp(age) as age\_deviation\_for\_population from T12\_studentinfo;*



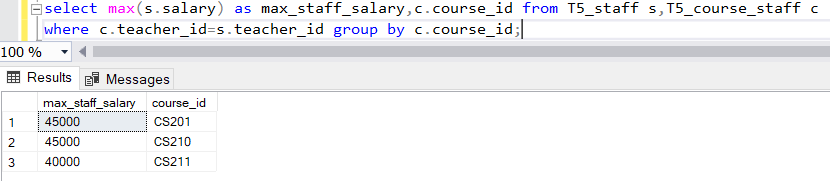
*select sum(fee\_paid) as total\_fee\_received from T12\_parentinfo;*



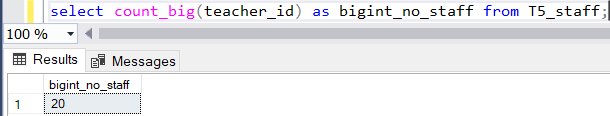
*select min(marks) as minimum\_marks,course\_id from T12\_grades group by course\_id;*



*select max(s.salary) as max\_staff\_salary,c.course\_id from T12\_staff s,T12\_course\_staff c where c.teacher\_id=s.teacher\_id group by c.course\_id;*



*select count\_big(teacher\_id) as bigint\_no\_staff from T12\_staff;*



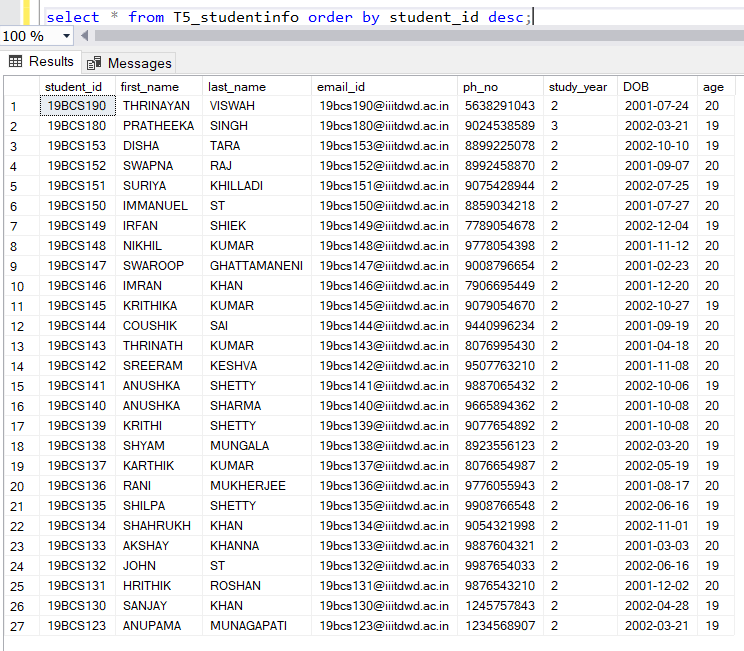
3)

ORDER BY:

Queries:

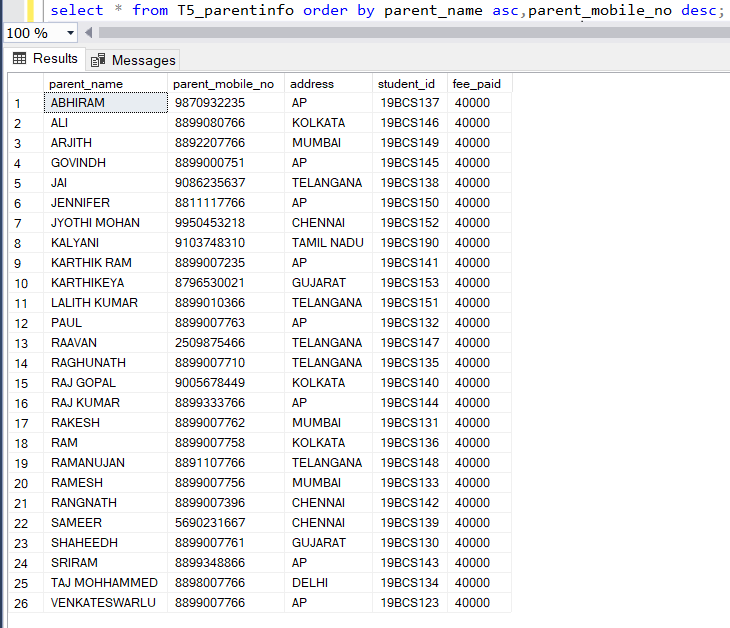
1)

*select \* from T12\_studentinfo order by student\_id desc;*



2)

*select \* from T12\_parentinfo order by parent\_name asc,parent\_mobile\_no desc;*

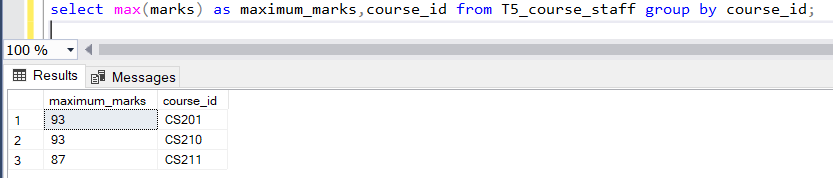


GROUP BY:

Queries:

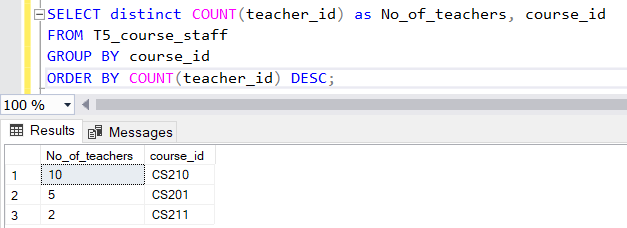
1)

*select max(marks) as maximum\_marks,course\_id from T12\_course\_staff group by course\_id;*



2)

*SELECT distinct COUNT(teacher\_id) as No\_of\_teachers, course\_id FROM T12\_course\_staff GROUP BY course\_id ORDER BY COUNT(teacher\_id) DESC;*

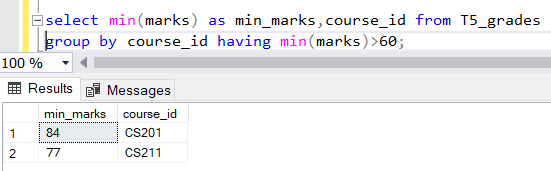


HAVING CLAUSE:

Queries:

1)

*select min(marks) as min\_marks,course\_id from T12\_grades group by course\_id having min(marks)>60;*

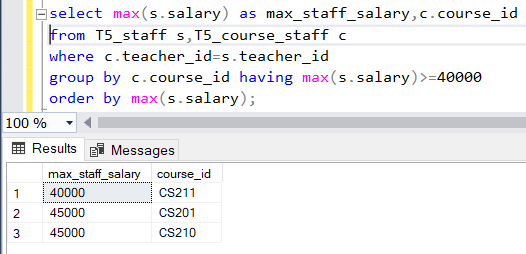


2)

*select max(s.salary) as max\_staff\_salary,c.course\_id from T12\_staff s,T12\_course\_staff c*

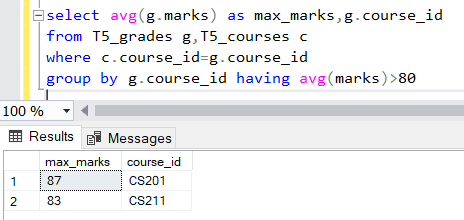
*where c.teacher\_id=s.teacher\_id group by c.course\_id*

*having max(s.salary)>=40000 order by max(s.salary);*



4) Use Aggregate function with group by and having.

*select avg(g.marks) as max\_marks,g.course\_id from T12\_grades g,T12\_courses c where c.course\_id=g.course\_id group by g.course\_id having avg(marks)>80*

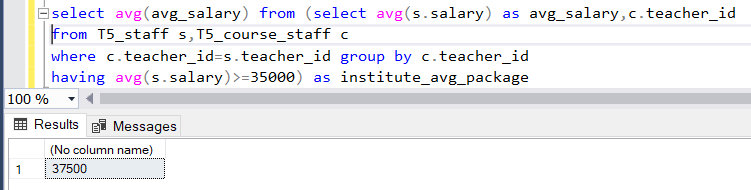


5) Write at least 3 nested queries using order by, group by and having clause.

Queries:

1)

*select avg(avg\_salary) from (select avg(s.salary) as avg\_salary,c.teacher\_id from T12\_staff s,T12\_course\_staff c where c.teacher\_id=s.teacher\_id group by c.teacher\_id having avg(s.salary)>=35000) as institute\_avg\_package*



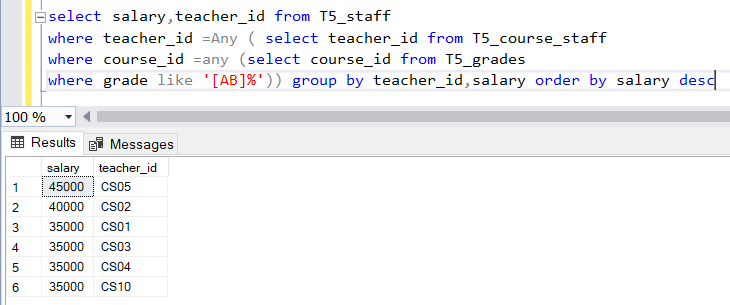
2)

*select salary,teacher\_id from T12\_staff*

*where teacher\_id =Any ( select teacher\_id from T12\_course\_staff*

*where course\_id =any (select course\_id from T12\_grades*

*where grade like '[AB]%')) group by teacher\_id,salary order by salary desc*

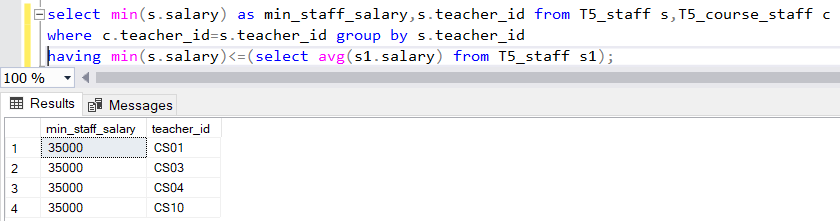


3)

*select min(s.salary) as min\_staff\_salary,s.teacher\_id from T12\_staff s,T12\_course\_staff c*

*where c.teacher\_id=s.teacher\_id group by s.teacher\_id*

*having min(s.salary)<=(select avg(s1.salary) from T12\_staff s1);*



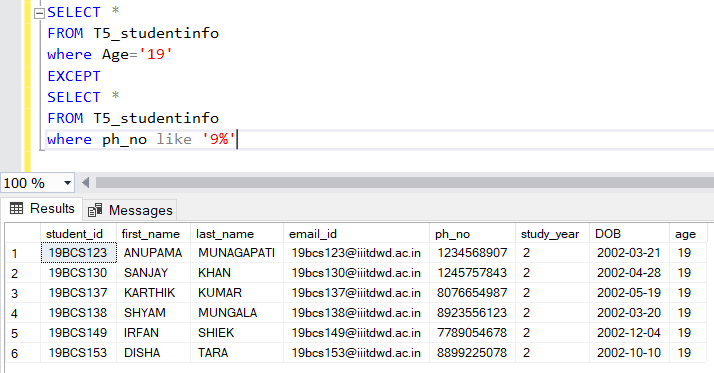
6) Illustrate the Usage of Except, Exists, Not Exists, Union, Intersection.

EXCEPT:

*SELECT \* FROM T12\_studentinfo where Age='19'*

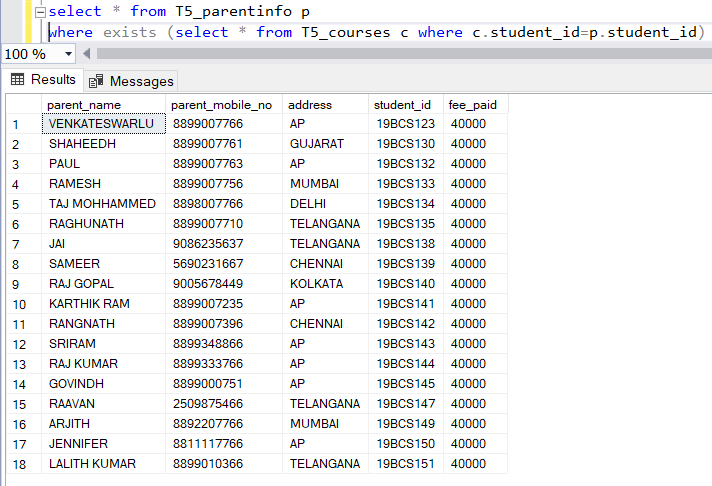
*EXCEPT*

*SELECT \* FROM T12\_studentinfo where ph\_no like '9%'*



EXISTS:

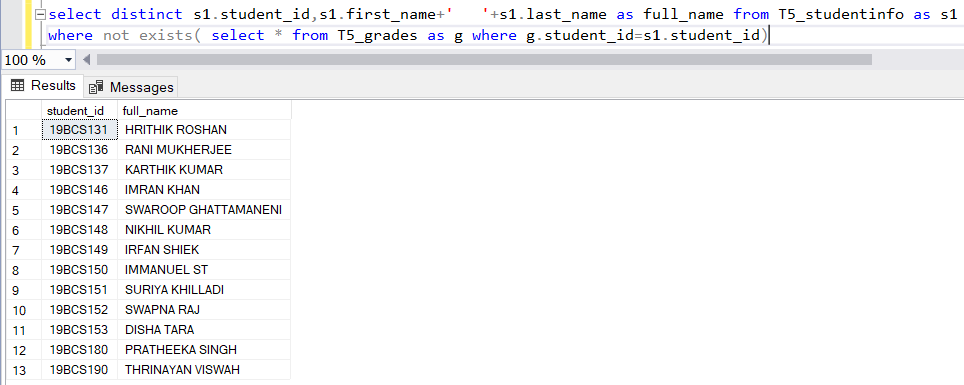
*select \* from T12\_parentinfo p where exists (select \* from T12\_courses c where c.student\_id=p.student\_id)*



NOT EXISTS:

*select distinct s1.student\_id,s1.first\_name+' '+s1.last\_name as full\_name from T12\_studentinfo as s1*

*where not exists( select \* from T12\_grades as g where g.student\_id=s1.student\_id)*



UNION:

*use school*

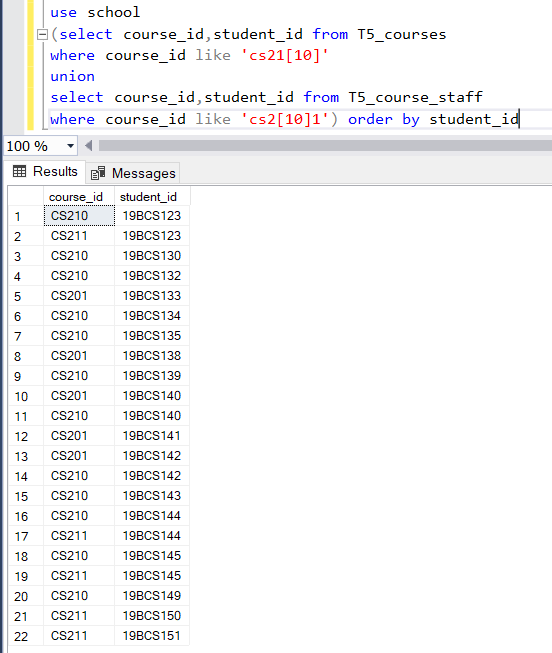
*(select course\_id,student\_id from T12\_courses*

*where course\_id like 'cs21[10]'*

*union*

*select course\_id,student\_id from T12\_course\_staff*

*where course\_id like 'cs2[10]1') order by student\_id*



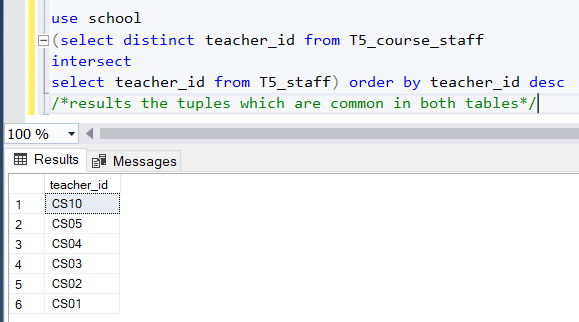
INTERSECT:

*use school*

*(select distinct teacher\_id from T12\_course\_staff*

*intersect*

*select teacher\_id from T12\_staff) order by teacher\_id desc*



7) INNER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN- 3 queries for each instance

Queries:

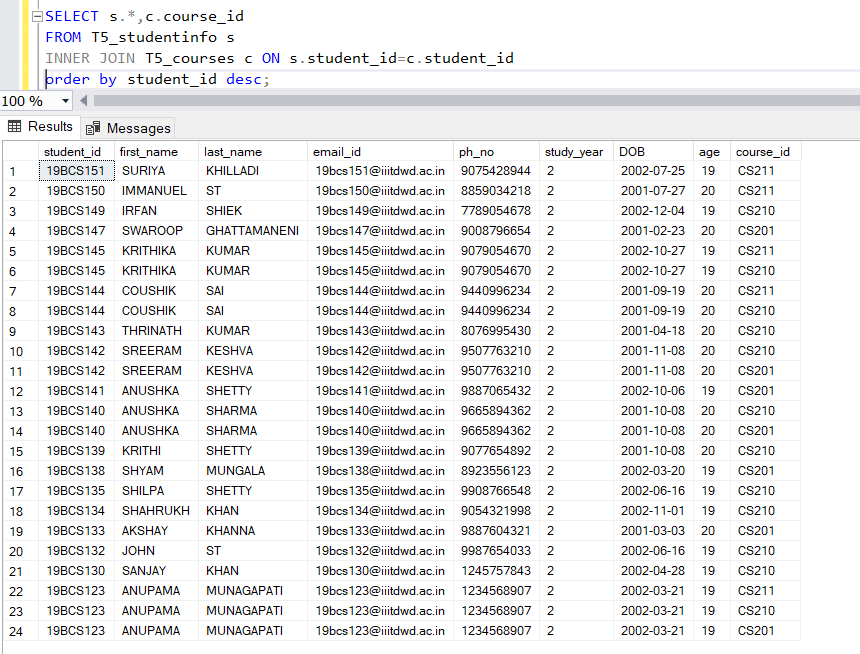
INNER JOIN:

1)

*select s.\*,c.course\_id*

*from T12\_studentinfo s*

*inner join T12\_courses c on s.student\_id=c.student\_id order by student\_id desc;*



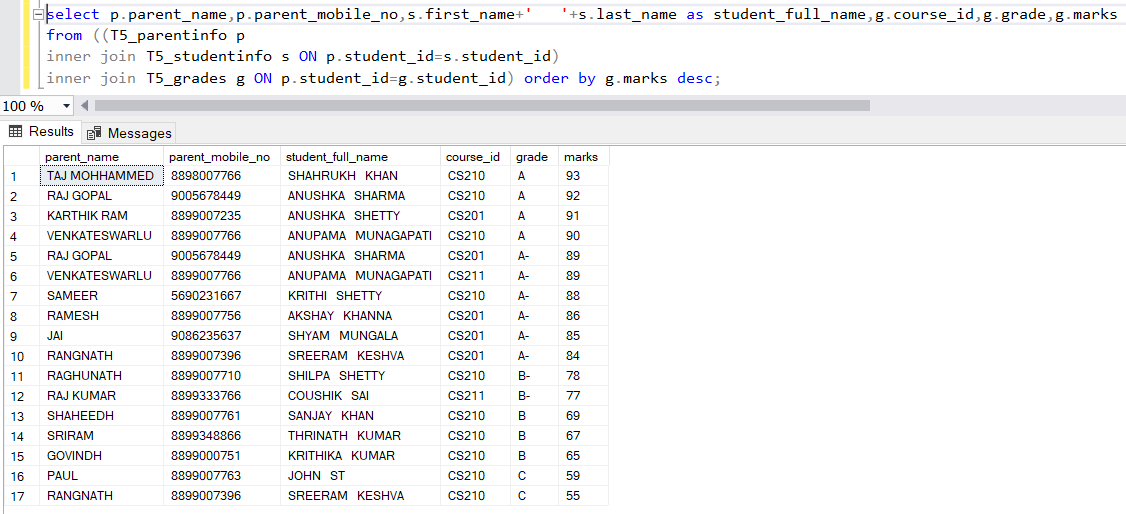
2)

*select p.parent\_name,p.parent\_mobile\_no,s.first\_name+' '+s.last\_name as student\_full\_name,g.course\_id,g.grade,g.marks*

*from ((T12\_parentinfo p*

*inner join T12\_studentinfo s ON p.student\_id=s.student\_id)*

*inner join T12\_grades g ON p.student\_id=g.student\_id) order by g.marks desc;*



3)

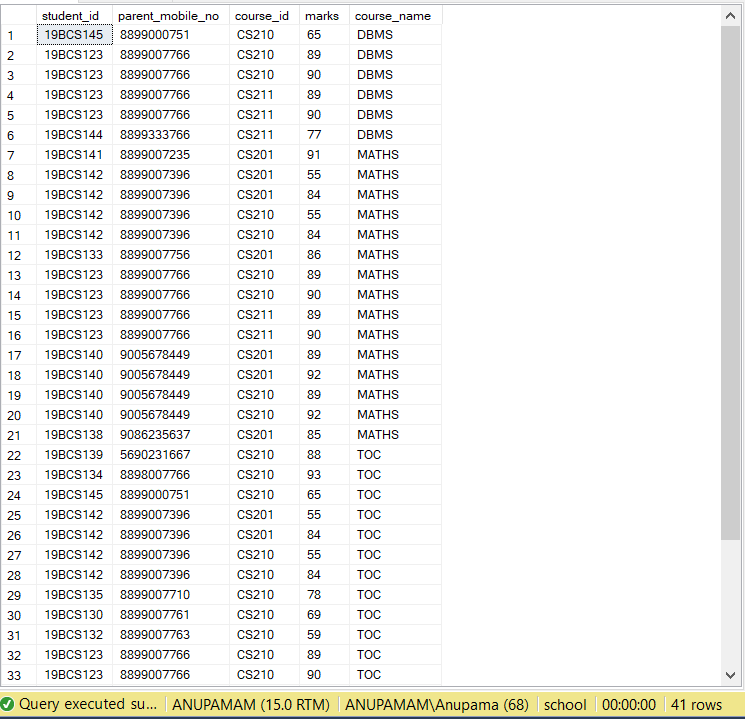
*select p.student\_id,p.parent\_mobile\_no,c.course\_id,g.marks,c1.course\_name*

*from (((T12\_parentinfo p*

*inner join T12\_course\_staff c on p.student\_id=c.student\_id)*

*inner join T12\_grades g on p.student\_id=g.student\_id)*

*inner join T12\_courses c1 on p.student\_id=c1.student\_id)order by c1.course\_name;*



LEFT OUTER JOIN:

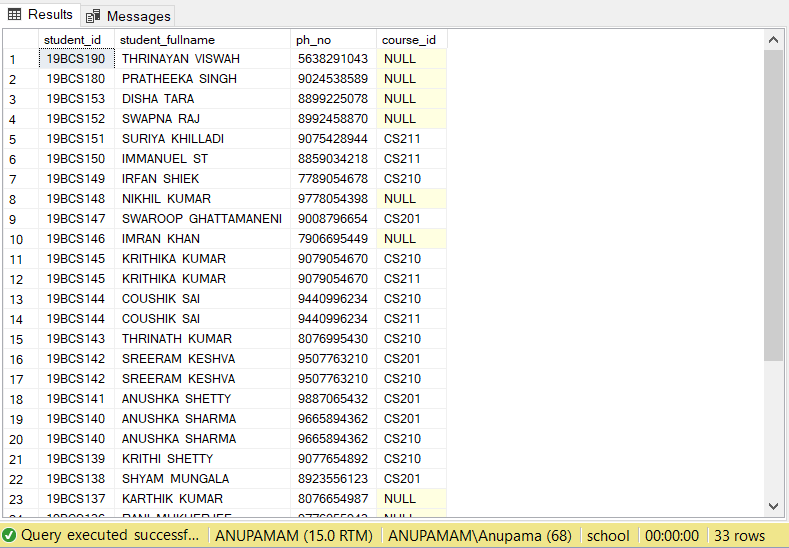
1)

*select s.student\_id,s.first\_name+' '+s.last\_name as student\_fullname,s.ph\_no,c.course\_id*

*from T12\_studentinfo s*

*left outer join T12\_courses c on s.student\_id=c.student\_id*

*order by student\_id desc;*



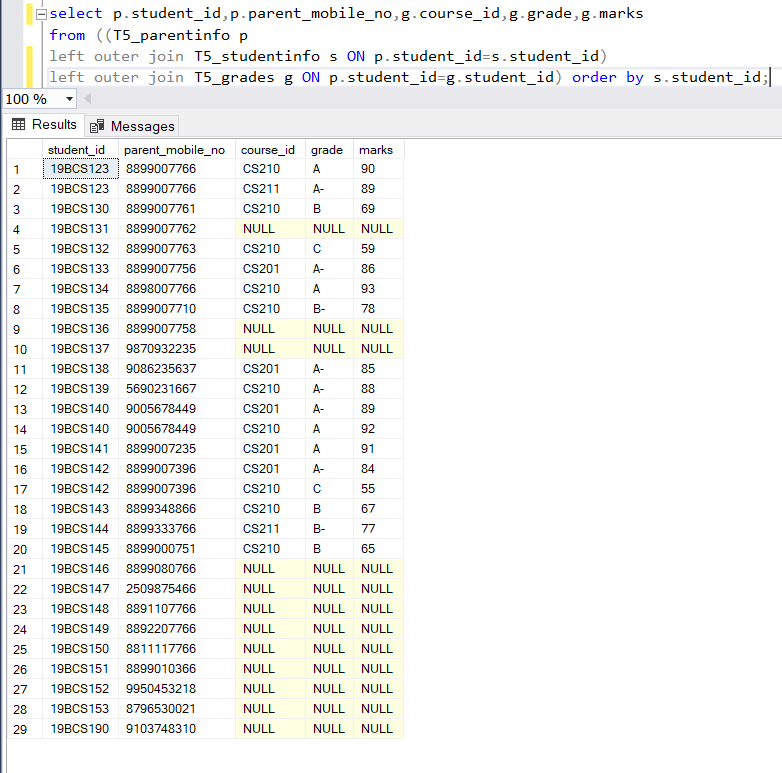
2)

*select p.student\_id,p.parent\_mobile\_no,g.course\_id,g.grade,g.marks*

*from ((T12\_parentinfo p*

*left outer join T12\_studentinfo s ON p.student\_id=s.student\_id)*

*left outer join T12\_grades g ON p.student\_id=g.student\_id) order by s.student\_id;*



3)

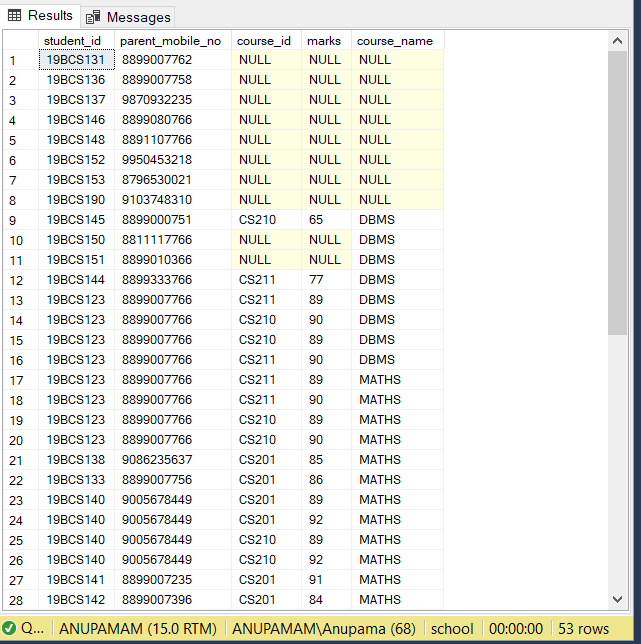
*select p.student\_id,p.parent\_mobile\_no,c.course\_id,g.marks,c1.course\_name*

*from (((T12\_parentinfo p left outer join T12\_course\_staff c on p.student\_id=c.student\_id)*

*left outer join T12\_grades g on p.student\_id=g.student\_id)*

*left outer join T12\_courses c1 on p.student\_id=c1.student\_id)*

*order by c1.course\_name;*



RIGHT OUTER JOIN:

1)

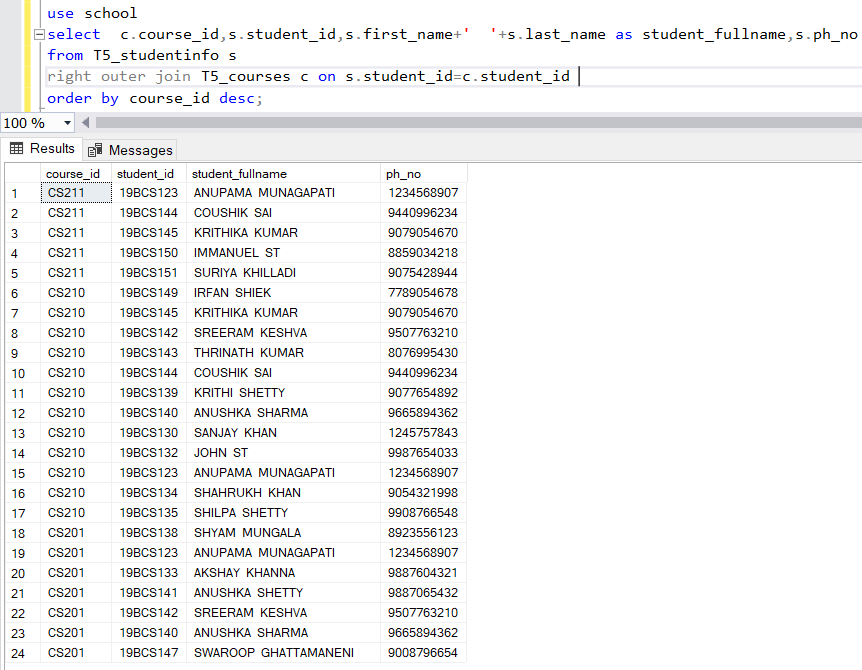
*use school*

*select c.course\_id,s.student\_id,s.first\_name+' '+s.last\_name as student\_fullname,s.ph\_no*

*from T12\_studentinfo s*

*right outer join T12\_courses c on s.student\_id=c.student\_id*

*order by course\_id desc;*



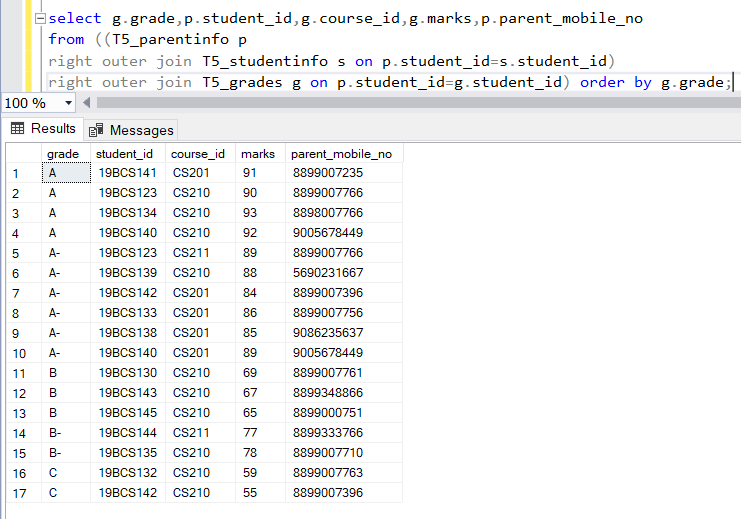
2)

*select g.grade,p.student\_id,g.course\_id,g.marks,p.parent\_mobile\_no*

*from ((T12\_parentinfo p*

*right outer join T12\_studentinfo s on p.student\_id=s.student\_id)*

*right outer join T12\_grades g on p.student\_id=g.student\_id) order by g.grade;*



3)

*select distinct p.student\_id,p.parent\_mobile\_no,c1.course\_name*

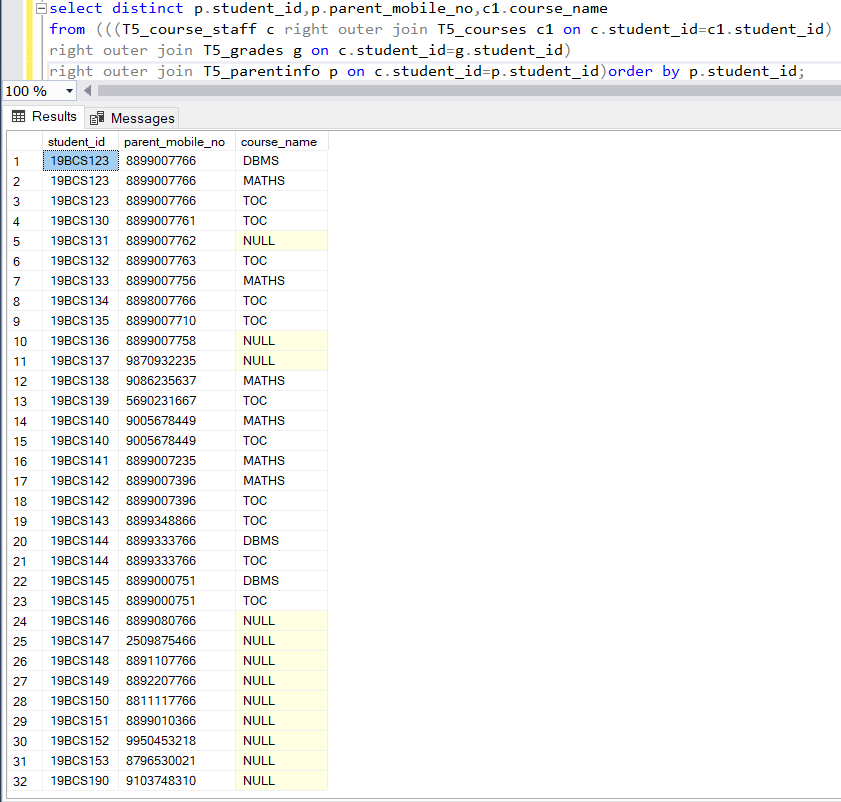
*from (((T12\_course\_staff c*

*right outer join T12\_courses c1 on c.student\_id=c1.student\_id)*

*right outer join T12\_grades g on c.student\_id=g.student\_id)*

*right outer join T12\_parentinfo p on c.student\_id=p.student\_id)*

*order by p.student\_id;*



8) Use all the above condition in JOIN as well.

Query:

*select s.student\_id,min(s.first\_name+' '+s.last\_name) as student\_fullname,c.course\_id*

*from T12\_studentinfo s*

*inner join T12\_courses c on s.student\_id=c.student\_id*

*group by c.course\_id,s.student\_id having s.student\_id < '19bcs180'*

*order by student\_id desc;*

