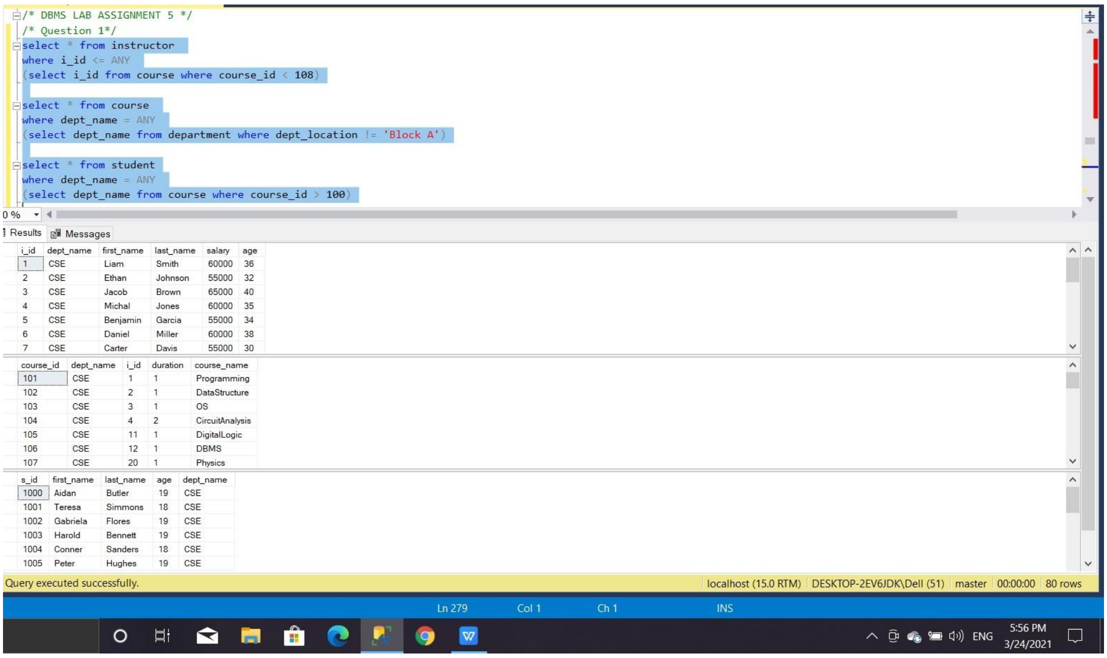
DATABASE MANAGEMENT SYSTEM
LAB ASSIGNMENT - 05

Q1).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



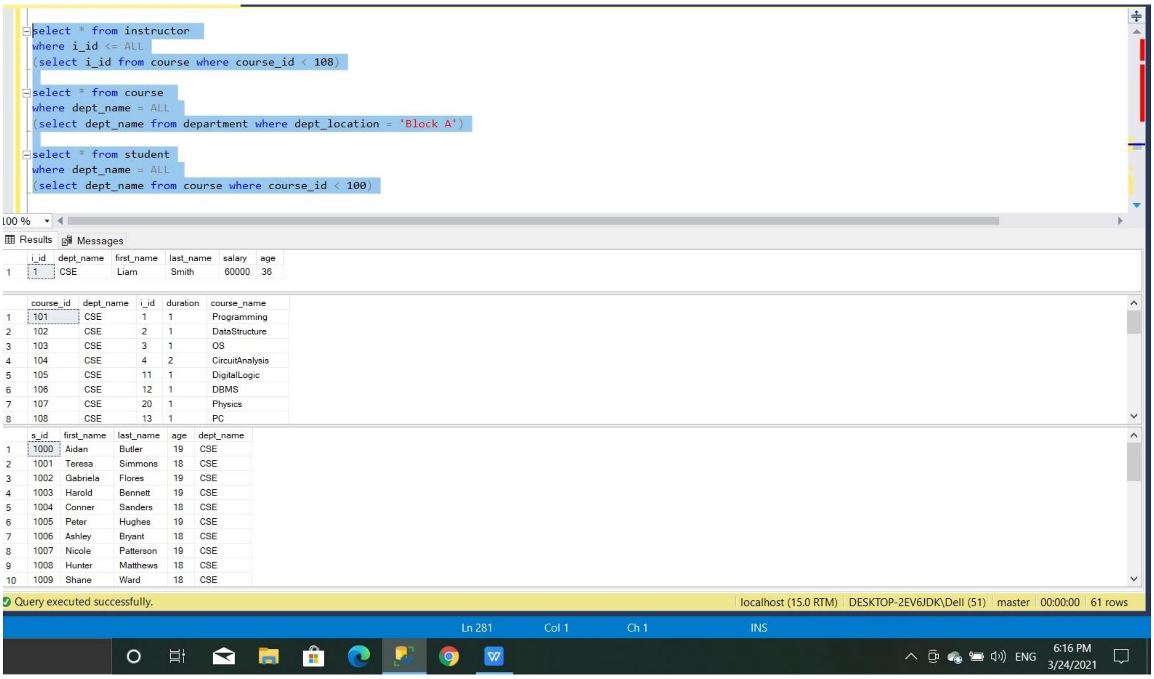
```sql
/* DBMS LAB ASSIGNMENT 5 */
/* Question 1*/
select * from instructor
where i_id <= ANY
(select i_id from course where course_id < 108)

select * from course
where dept_name = ANY
(select dept_name from department where dept_location != 'Block A')

select * from student
where dept_name = ANY
(select dept_name from course where course_id > 100)
```

i_id	dept_name	first_name	last_name	salary	age
1	CSE	Liam	Smith	60000	36
2	CSE	Ethan	Johnson	55000	32
3	CSE	Jacob	Brown	65000	40
4	CSE	Michal	Jones	60000	35
5	CSE	Benjamin	Garcia	55000	34
6	CSE	Daniel	Miller	60000	38
7	CSE	Carter	Davis	55000	30

course_id	dept_name	i_id	duration	course_name
101	CSE	1	1	Programming
102	CSE	2	1	DataStructure
103	CSE	3	1	OS
104	CSE	4	2	CircuitAnalysis
105	CSE	11	1	DigitalLogic
106	CSE	12	1	DBMS
107	CSE	20	1	Physics

s_id	first_name	last_name	age	dept_name
1000	Aidan	Butler	19	CSE
1001	Teresa	Simmons	18	CSE
1002	Gabriela	Flores	19	CSE
1003	Harold	Bennett	19	CSE
1004	Conner	Sanders	18	CSE
1005	Peter	Hughes	19	CSE

Query executed successfully. localhost (15.0 RTM) | DESKTOP-2EV6JDK\Dell (51) | master | 00:00:00 | 80 rows



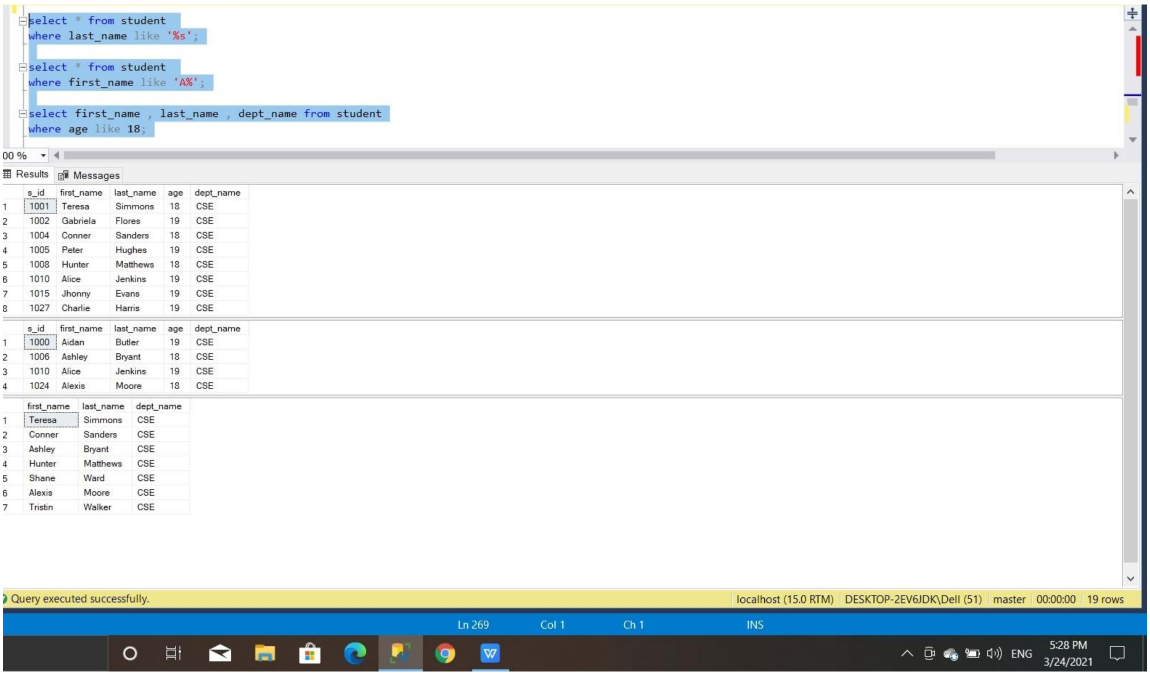
```sql
select * from instructor
where i_id <= ALL
(select i_id from course where course_id < 108)

select * from course
where dept_name = ALL
(select dept_name from department where dept_location = 'Block A')

select * from student
where dept_name = ALL
(select dept_name from course where course_id < 100)
```

	i_id	dept_name	first_name	last_name	salary	age
1	1	CSE	Liam	Smith	60000	36

	course_id	dept_name	i_id	duration	course_name
1	101	CSE	1	1	Programming
2	102	CSE	2	1	DataStructure
3	103	CSE	3	1	OS
4	104	CSE	4	2	CircuitAnalysis
5	105	CSE	11	1	DigitalLogic
6	106	CSE	12	1	DBMS
7	107	CSE	20	1	Physics
8	108	CSE	13	1	PC

	s_id	first_name	last_name	age	dept_name
1	1000	Aidan	Butler	19	CSE
2	1001	Teresa	Simmons	18	CSE
3	1002	Gabriela	Flores	19	CSE
4	1003	Harold	Bennett	19	CSE
5	1004	Conner	Sanders	18	CSE
6	1005	Peter	Hughes	19	CSE
7	1006	Ashley	Bryant	18	CSE
8	1007	Nicole	Patterson	19	CSE
9	1008	Hunter	Matthews	18	CSE
10	1009	Shane	Ward	18	CSE

Query executed successfully. localhost (15.0 RTM) | DESKTOP-2EV6JDK\Dell (51) | master | 00:00:00 | 61 rows



```
select * from student
where last_name like '%s';

select * from student
where first_name like 'A%';

select first_name , last_name , dept_name from student
where age like 18;
```

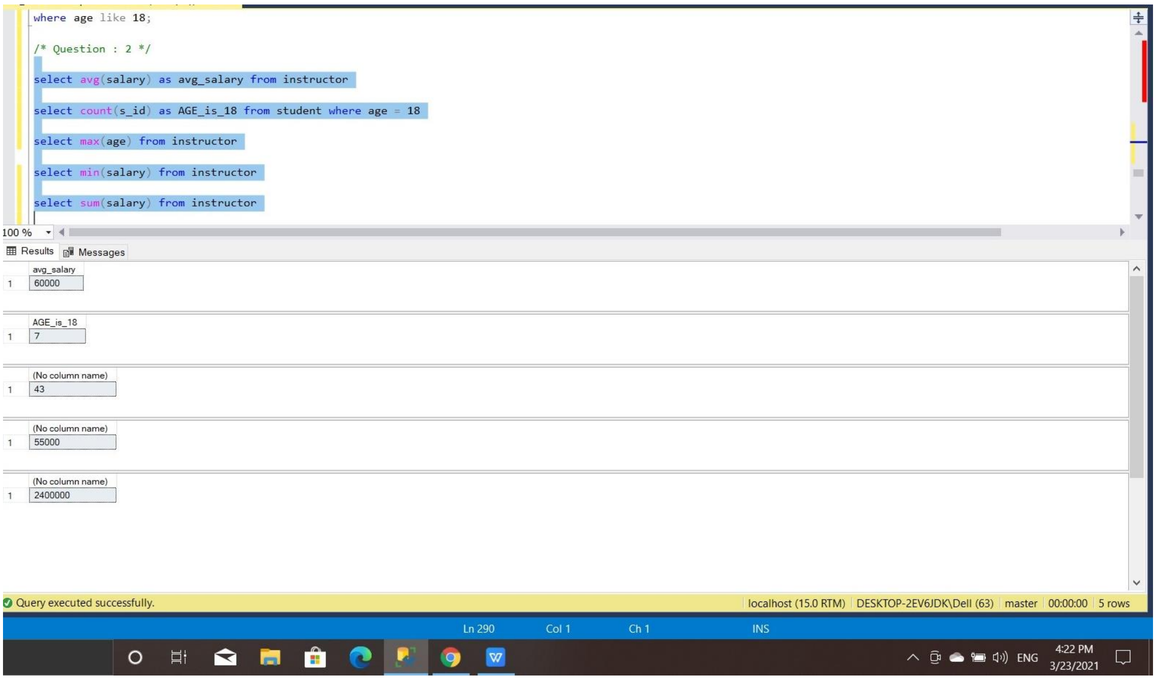
	s_id	first_name	last_name	age	dept_name
1	1001	Teresa	Simmons	18	CSE
2	1002	Gabriela	Flores	19	CSE
3	1004	Conner	Sanders	18	CSE
4	1005	Peter	Hughes	19	CSE
5	1008	Hunter	Matthews	18	CSE
6	1010	Alice	Jenkins	19	CSE
7	1015	Jhonny	Evans	19	CSE
8	1027	Charlie	Harris	19	CSE

	s_id	first_name	last_name	age	dept_name
1	1000	Aidan	Butler	19	CSE
2	1006	Ashley	Bryant	18	CSE
3	1010	Alice	Jenkins	19	CSE
4	1024	Alexis	Moore	18	CSE

	first_name	last_name	dept_name
1	Teresa	Simmons	CSE
2	Conner	Sanders	CSE
3	Ashley	Bryant	CSE
4	Hunter	Matthews	CSE
5	Shane	Ward	CSE
6	Alexis	Moore	CSE
7	Tristin	Walker	CSE

```
Query executed successfully.    localhost (15.0 RTM) | DESKTOP-2EV6JDK\Dell (51) | master | 00:00:00 | 19 rows
```

Q2). One query for each Aggregate function



```
where age like 18;

/* Question : 2 */

select avg(salary) as avg_salary from instructor

select count(s_id) as AGE_is_18 from student where age = 18

select max(age) from instructor

select min(salary) from instructor

select sum(salary) from instructor
```

	avg_salary
1	60000

	AGE_is_18
1	7

	(No column name)
1	43

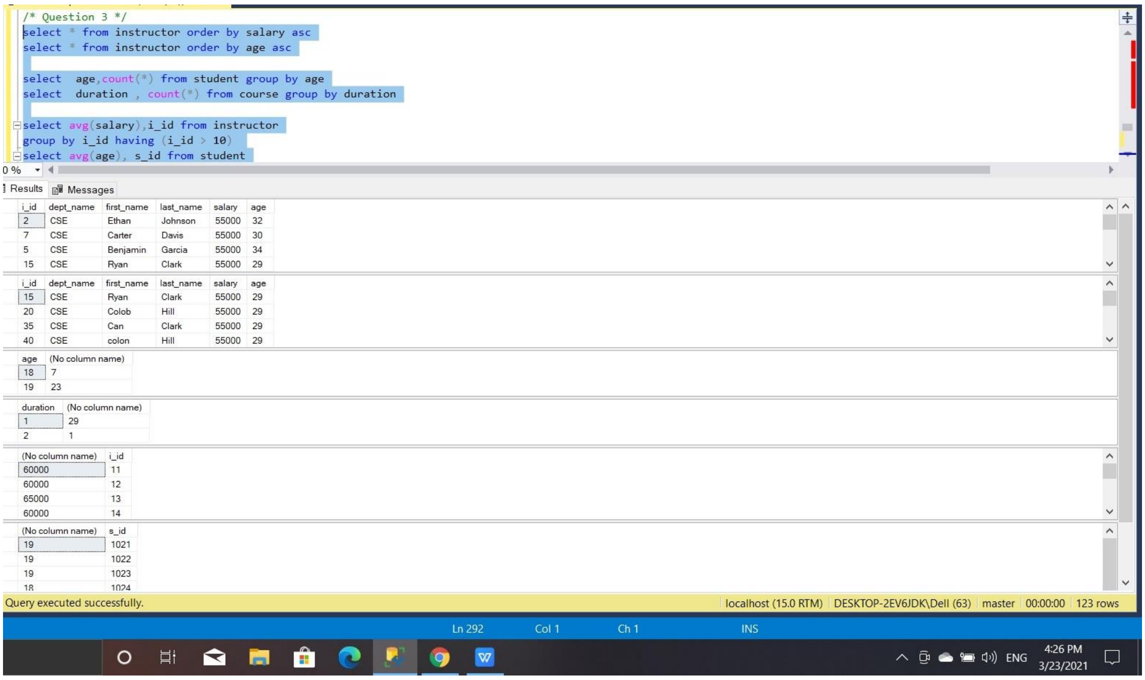
	(No column name)
1	55000

	(No column name)
1	2400000

```
Query executed successfully.    localhost (15.0 RTM) | DESKTOP-2EV6JDK\Dell (63) | master | 00:00:00 | 5 rows
```

Q3).Illustrate the usage of order by, group by and having clause (2 queries for each case)

```
select avg(salary),i_id from instructor
group by i_id having (i_id > 10)
select avg(age), s_id from student
group by s_id having (s_id >1020)
```



```
/* Question 3 */
select * from instructor order by salary asc
select * from instructor order by age asc

select  age,count(*) from student group by age
select  duration , count(*) from course group by duration

select avg(salary),i_id from instructor
group by i_id having (i_id > 10)
select avg(age), s_id from student
```

i_id	dept_name	first_name	last_name	salary	age
2	CSE	Ethan	Johnson	55000	32
7	CSE	Carter	Davis	55000	30
5	CSE	Benjamin	Garcia	55000	34
15	CSE	Ryan	Clark	55000	29

i_id	dept_name	first_name	last_name	salary	age
15	CSE	Ryan	Clark	55000	29
20	CSE	Colob	Hill	55000	29
35	CSE	Can	Clark	55000	29
40	CSE	colon	Hill	55000	29

age	(No column name)
18	7
19	23

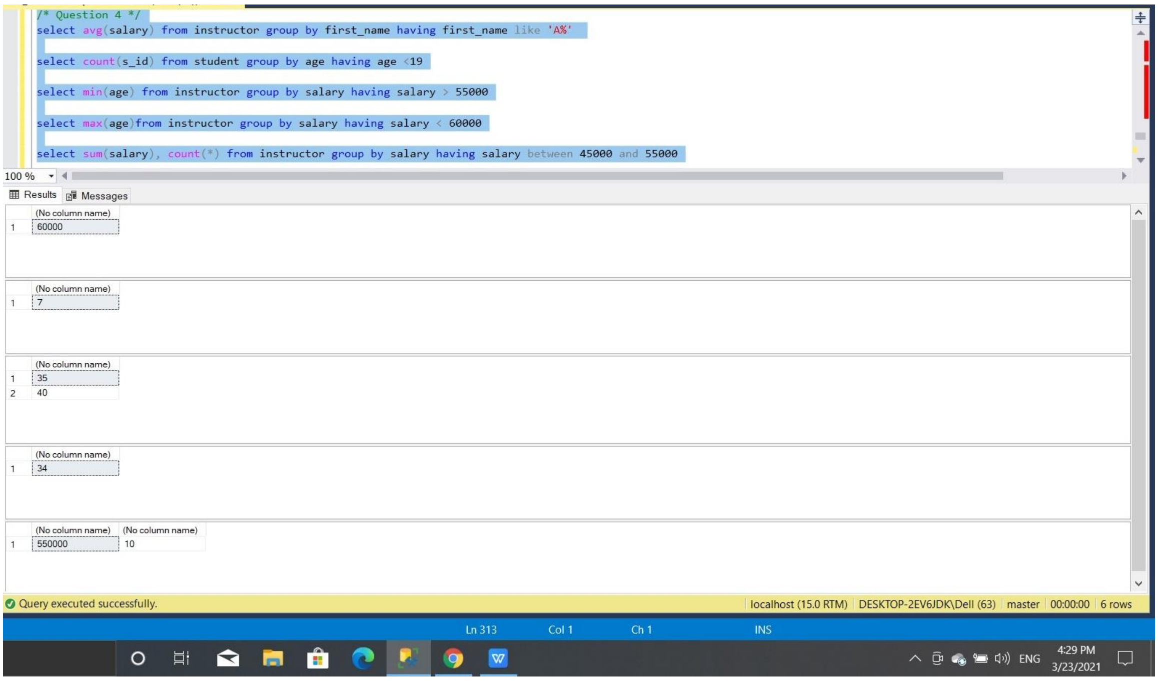
duration	(No column name)
1	29
2	1

(No column name)	i_id
60000	11
60000	12
65000	13
60000	14

(No column name)	s_id
19	1021
19	1022
19	1023
18	1024

Query executed successfully. localhost (15.0 RTM) | DESKTOP-2EV6JDK\Dell (63) | master | 00:00:00 | 123 rows

Q4).Use Aggregate function with group by and having



```
/* Question 4 */
select avg(salary) from instructor group by first_name having first_name like 'A%'

select count(s_id) from student group by age having age <19

select min(age) from instructor group by salary having salary > 55000

select max(age)from instructor group by salary having salary < 60000

select sum(salary), count(*) from instructor group by salary having salary between 45000 and 55000
```

	(No column name)
1	60000

	(No column name)
1	7

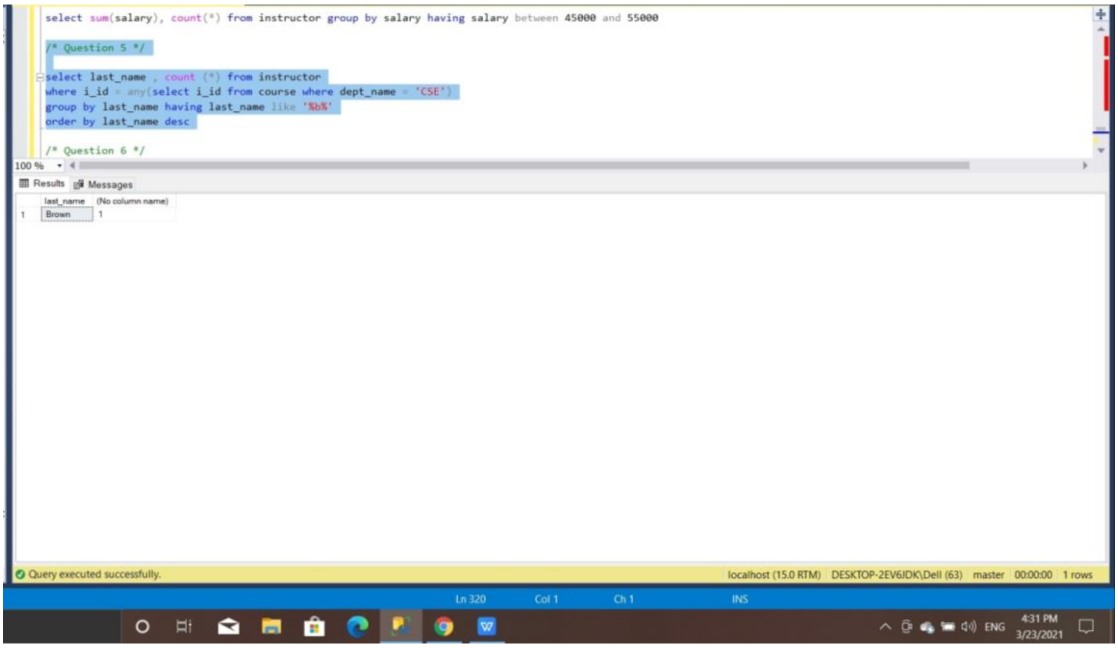
	(No column name)
1	35
2	40

	(No column name)
1	34

	(No column name)	(No column name)
1	550000	10

Query executed successfully. localhost (15.0 RTM) | DESKTOP-2EV6JDK\Dell (63) | master | 00:00:00 | 6 rows

Q5).Write atleast three nested queries using order by, group by and having clause



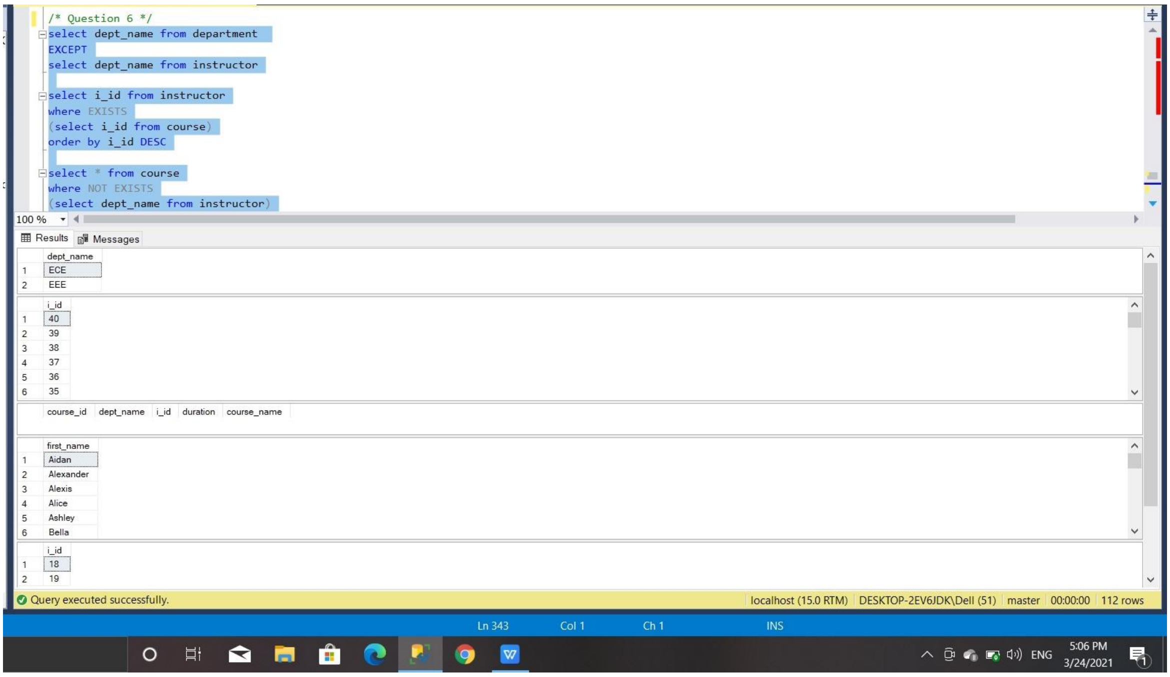
```
select sum(salary), count(*) from instructor group by salary having salary between 45000 and 55000

/* Question 5 */

select last_name , count (*) from instructor
where i_id = any(select i_id from course where dept_name = 'CSE')
group by last_name having last_name like '%b%'
order by last_name desc

/* Question 6 */
```

```
   last_name  (No column name)
1  Brown      1
```

```
Query executed successfully.    localhost (15.0 RTM)  DESKTOP-2EV6JDK\Dell (63)  master  00:00:00  1 rows
```

Q6).Illustrate the Usage of Except, Exists, Not Exists, Union, Intersection

```
/* Question 6 */
select dept_name from department
EXCEPT
select dept_name from instructor

select i_id from instructor
where EXISTS
(select i_id from course)
order by i_id DESC

select * from course
where NOT EXISTS
(select dept_name from instructor)

select first_name from instructor
UNION
select first_name from student

select i_id from instructor
intersect select age from
student
```

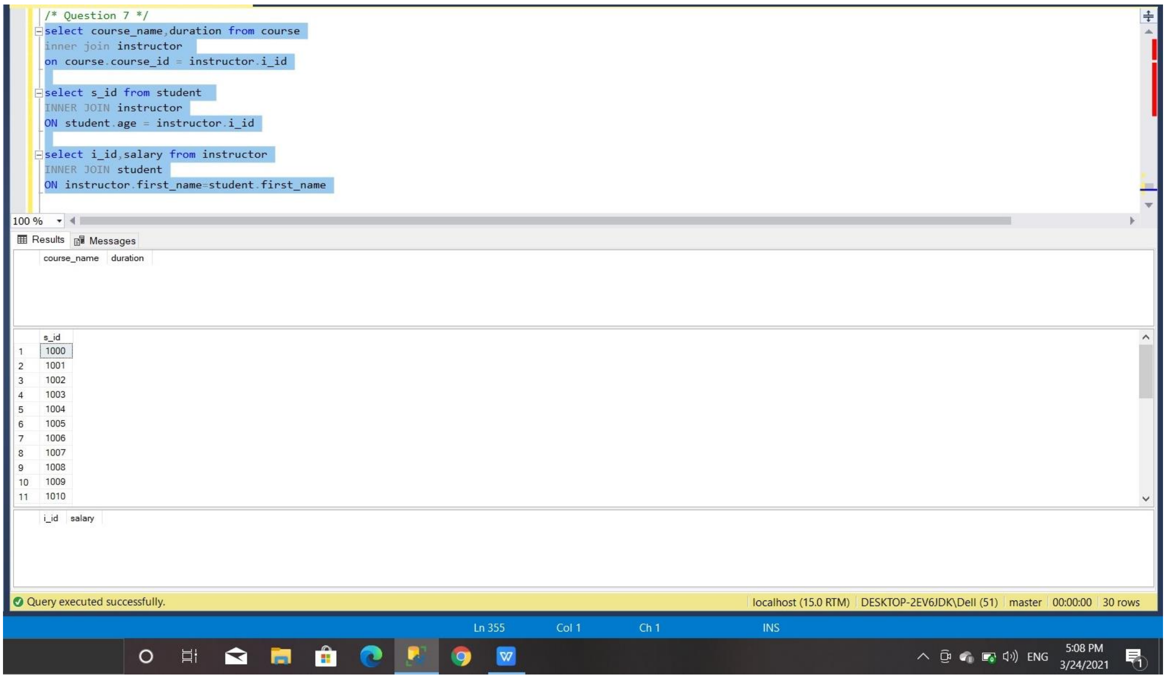


```
/* Question 6 */
select dept_name from department
EXCEPT
select dept_name from instructor

select i_id from instructor
where EXISTS
(select i_id from course)
order by i_id DESC

select * from course
where NOT EXISTS
(select dept_name from instructor)
```

```
   dept_name
1  ECE
2  EEE

   i_id
1  40
2  39
3  38
4  37
5  36
6  35

   course_id  dept_name  i_id  duration  course_name

   first_name
1  Aidan
2  Alexander
3  Alexis
4  Alice
5  Ashley
6  Bella

   i_id
1  18
2  19
```

```
Query executed successfully.    localhost (15.0 RTM)   DESKTOP-2EV6JDK\Dell (51)   master   00:00:00   112 rows
```

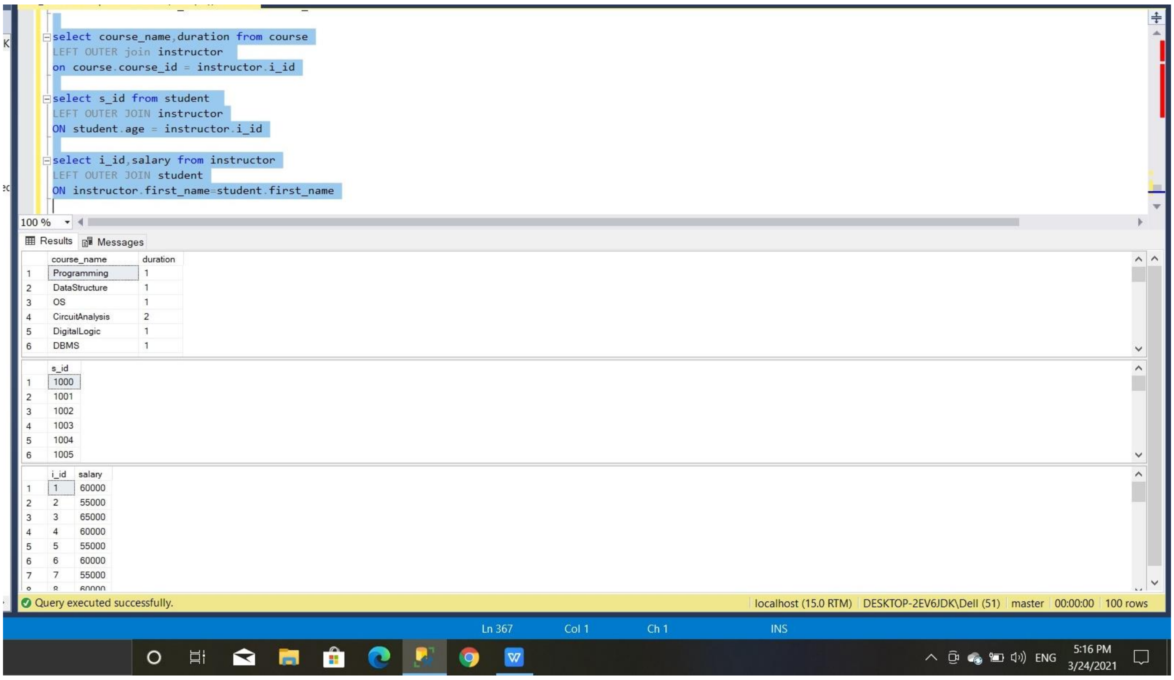
Q7).INNER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN- 3 queries for each instance



```
/* Question 7 */
select course_name,duration from course
inner join instructor
on course.course_id = instructor.i_id

select s_id from student
INNER JOIN instructor
ON student.age = instructor.i_id

select i_id,salary from instructor
INNER JOIN student
ON instructor.first_name=student.first_name
```

```
   course_name  duration

    s_id
1   1000
2   1001
3   1002
4   1003
5   1004
6   1005
7   1006
8   1007
9   1008
10  1009
11  1010

   i_id  salary
```

```
Query executed successfully.    localhost (15.0 RTM)   DESKTOP-2EV6JDK\Dell (51)   master   00:00:00   30 rows
```



```
select course_name,duration from course
LEFT OUTER join instructor
on course.course_id = instructor.i_id

select s_id from student
LEFT OUTER JOIN instructor
ON student.age = instructor.i_id

select i_id,salary from instructor
LEFT OUTER JOIN student
ON instructor.first_name=student.first_name
```

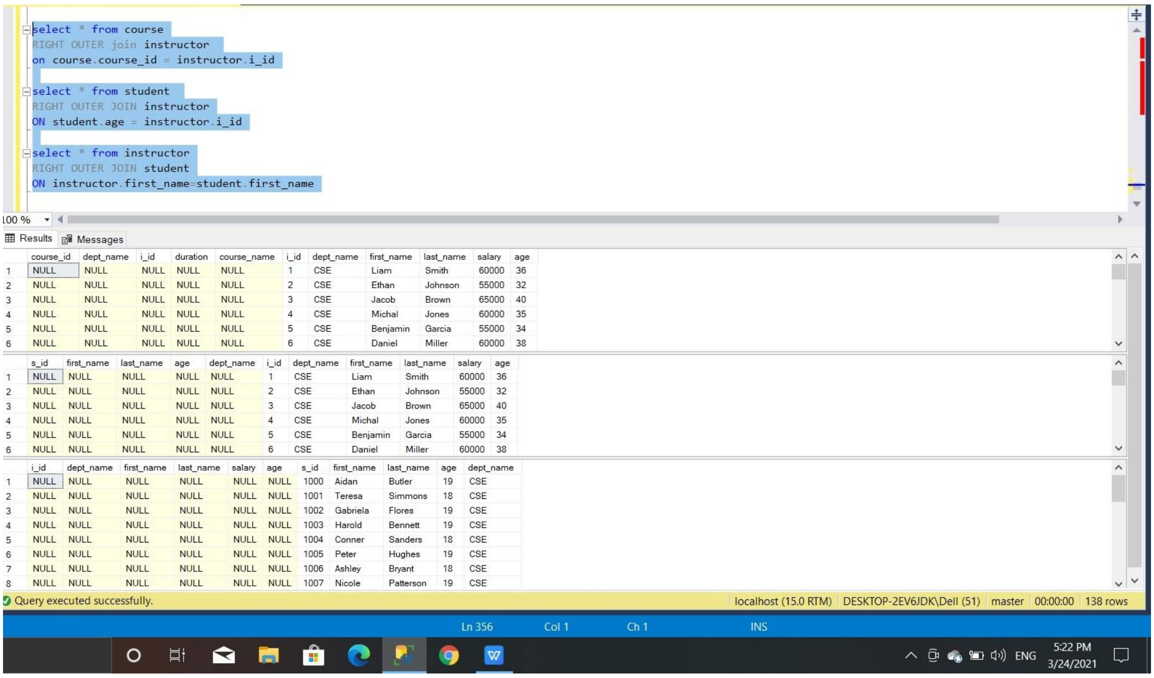
	course_name	duration
1	Programming	1
2	DataStructure	1
3	OS	1
4	CircuitAnalysis	2
5	DigitalLogic	1
6	DBMS	1

	s_id
1	1000
2	1001
3	1002
4	1003
5	1004
6	1005

	i_id	salary
1	1	60000
2	2	55000
3	3	65000
4	4	60000
5	5	55000
6	6	60000
7	7	55000
8	8	60000

Query executed successfully.

localhost (15.0 RTM) | DESKTOP-2EV6JDK\Dell (51) | master | 00:00:00 | 100 rows



```
select * from course
RIGHT OUTER join instructor
on course.course_id = instructor.i_id

select * from student
RIGHT OUTER JOIN instructor
ON student.age = instructor.i_id

select * from instructor
RIGHT OUTER JOIN student
ON instructor.first_name=student.first_name
```

	course_id	dept_name	i_id	duration	course_name	i_id	dept_name	first_name	last_name	salary	age
1	NULL	NULL	NULL	NULL	NULL	1	CSE	Liam	Smith	60000	36
2	NULL	NULL	NULL	NULL	NULL	2	CSE	Ethan	Johnson	55000	32
3	NULL	NULL	NULL	NULL	NULL	3	CSE	Jacob	Brown	65000	40
4	NULL	NULL	NULL	NULL	NULL	4	CSE	Michal	Jones	60000	35
5	NULL	NULL	NULL	NULL	NULL	5	CSE	Benjamin	Garcia	55000	34
6	NULL	NULL	NULL	NULL	NULL	6	CSE	Daniel	Miller	60000	38

	s_id	first_name	last_name	age	dept_name	i_id	dept_name	first_name	last_name	salary	age
1	NULL	NULL	NULL	NULL	NULL	1	CSE	Liam	Smith	60000	36
2	NULL	NULL	NULL	NULL	NULL	2	CSE	Ethan	Johnson	55000	32
3	NULL	NULL	NULL	NULL	NULL	3	CSE	Jacob	Brown	65000	40
4	NULL	NULL	NULL	NULL	NULL	4	CSE	Michal	Jones	60000	35
5	NULL	NULL	NULL	NULL	NULL	5	CSE	Benjamin	Garcia	55000	34
6	NULL	NULL	NULL	NULL	NULL	6	CSE	Daniel	Miller	60000	38

	i_id	dept_name	first_name	last_name	salary	age	s_id	first_name	last_name	age	dept_name
1	NULL	NULL	NULL	NULL	NULL	NULL	1000	Aidan	Butler	19	CSE
2	NULL	NULL	NULL	NULL	NULL	NULL	1001	Teresa	Simmons	18	CSE
3	NULL	NULL	NULL	NULL	NULL	NULL	1002	Gabriela	Flores	19	CSE
4	NULL	NULL	NULL	NULL	NULL	NULL	1003	Harold	Bennett	19	CSE
5	NULL	NULL	NULL	NULL	NULL	NULL	1004	Conner	Sanders	18	CSE
6	NULL	NULL	NULL	NULL	NULL	NULL	1005	Peter	Hughes	19	CSE
7	NULL	NULL	NULL	NULL	NULL	NULL	1006	Ashley	Bryant	18	CSE
8	NULL	NULL	NULL	NULL	NULL	NULL	1007	Nicole	Patterson	19	CSE

Query executed successfully.

localhost (15.0 RTM) | DESKTOP-2EV6JDK\Dell (51) | master | 00:00:00 | 138 rows