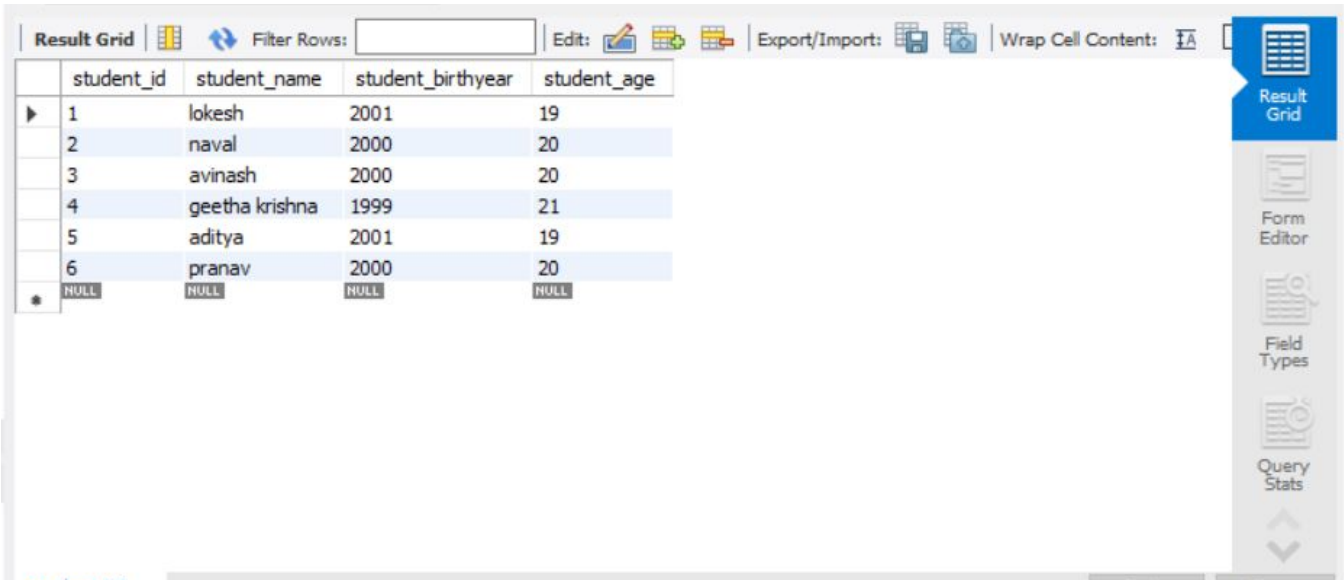
Assignment - 4

Present DataBase:-

Student :

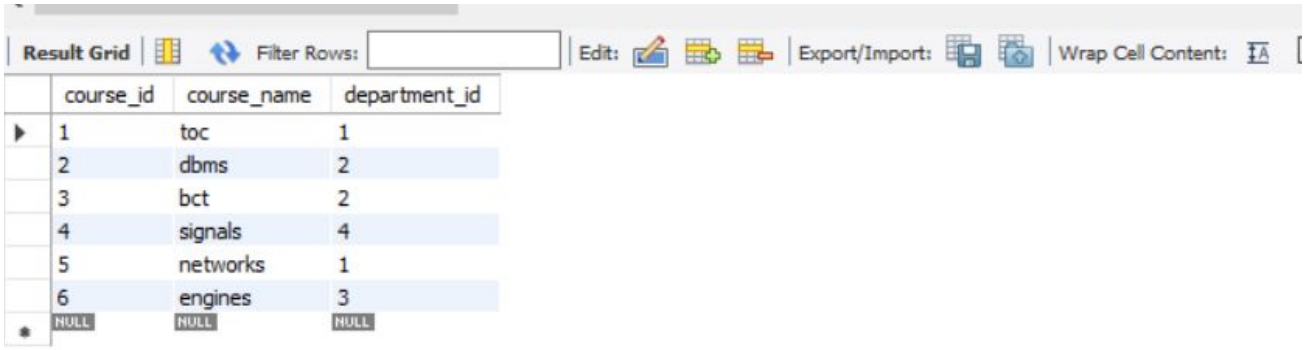


Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

student_id	student_name	student_birthyear	student_age
1	lokesh	2001	19
2	naval	2000	20
3	avinash	2000	20
4	geetha krishna	1999	21
5	aditya	2001	19
6	pranav	2000	20
NULL	NULL	NULL	NULL

Result Grid | Form Editor | Field Types | Query Stats

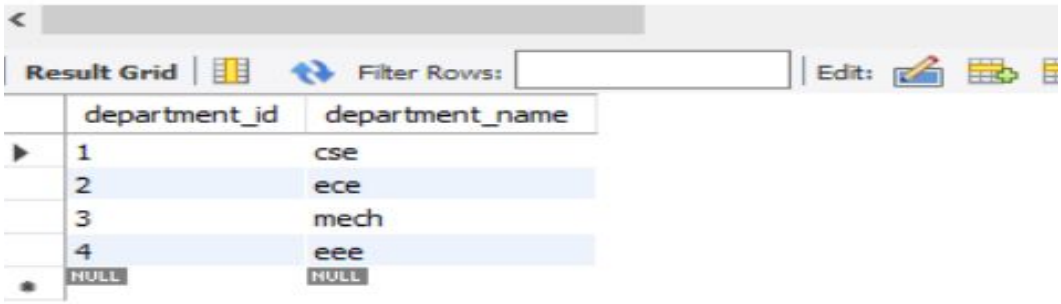
Course :



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

course_id	course_name	department_id
1	toc	1
2	dbms	2
3	bct	2
4	signals	4
5	networks	1
6	engines	3
NULL	NULL	NULL

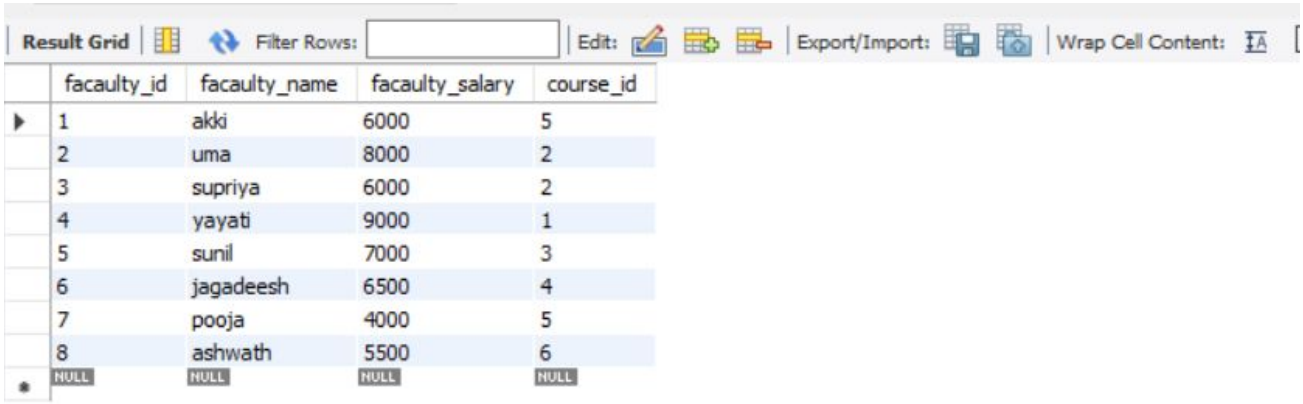
Department :



Result Grid | Filter Rows: | Edit:

department_id	department_name
1	cse
2	ece
3	mech
4	eee
NULL	NULL

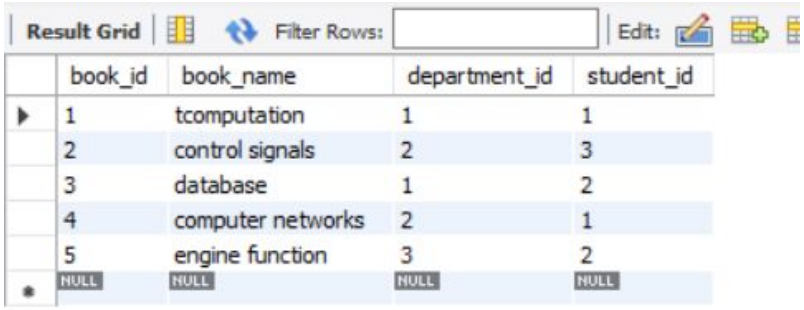
Faculty :



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

faculty_id	faculty_name	faculty_salary	course_id
1	akki	6000	5
2	uma	8000	2
3	supriya	6000	2
4	yayati	9000	1
5	sunil	7000	3
6	jagadeesh	6500	4
7	pooja	4000	5
8	ashwath	5500	6
NULL	NULL	NULL	NULL

Library :



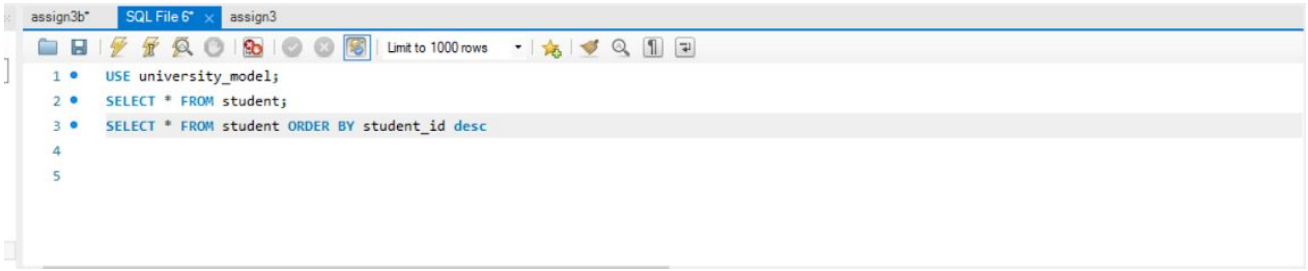
Result Grid | Filter Rows: | Edit:

book_id	book_name	department_id	student_id
1	tcomputation	1	1
2	control signals	2	3
3	database	1	2
4	computer networks	2	1
5	engine function	3	2
NULL	NULL	NULL	NULL

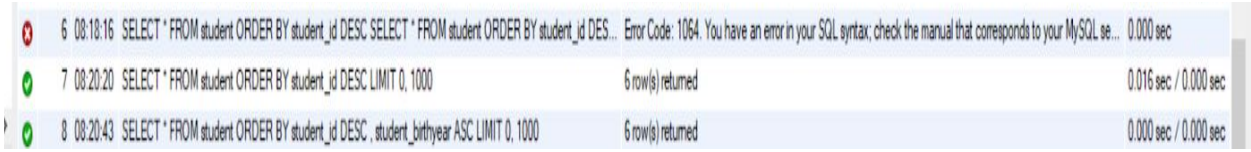
1. Order by Clause.

Query :

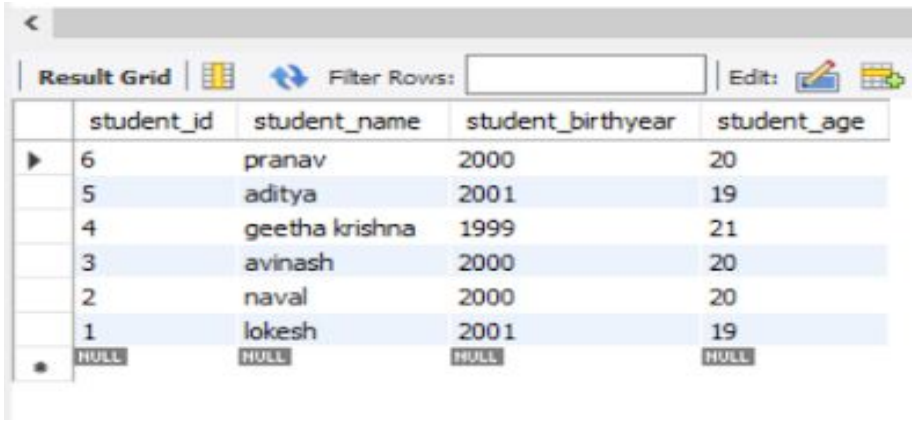
The query below sorts the student table in descending order.



Output :



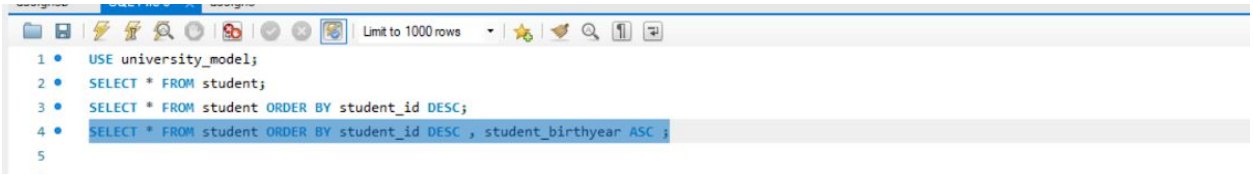
The Sorted output of students table in descending order



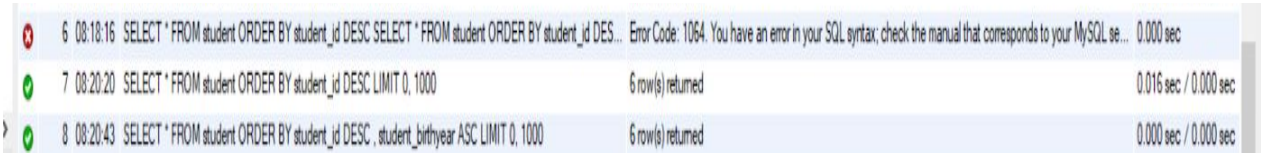
student_id	student_name	student_birthyear	student_age
6	pranav	2000	20
5	aditya	2001	19
4	geetha krishna	1999	21
3	avinash	2000	20
2	naval	2000	20
1	lokesh	2001	19
NULL	NULL	NULL	NULL

Query :

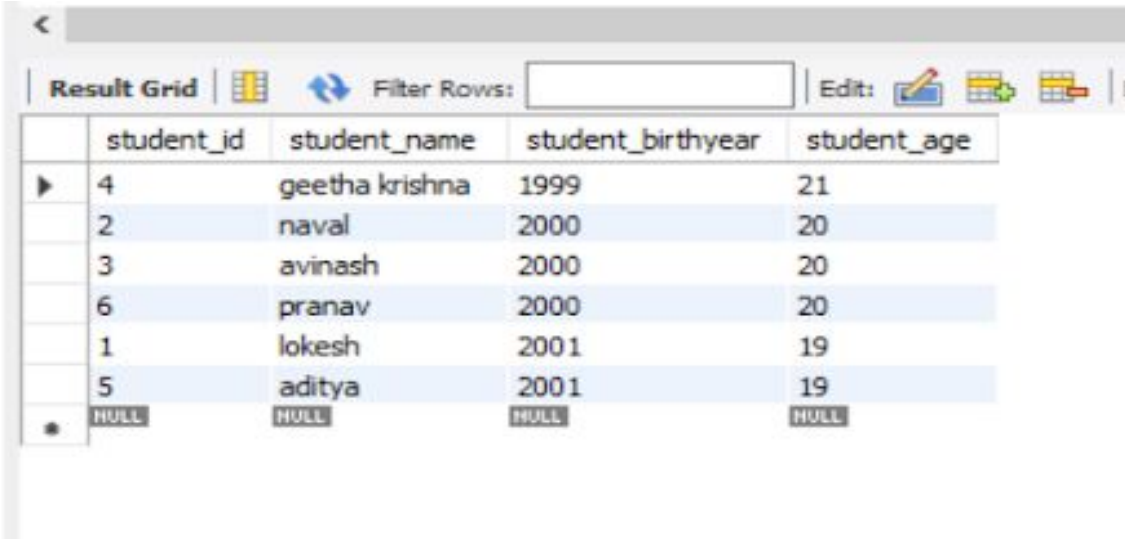
The query below sorts the student table first by birth year in ascending order but if the birth years are the same then it sorts with the help of student_id in descending order.



Output :



The sorted output

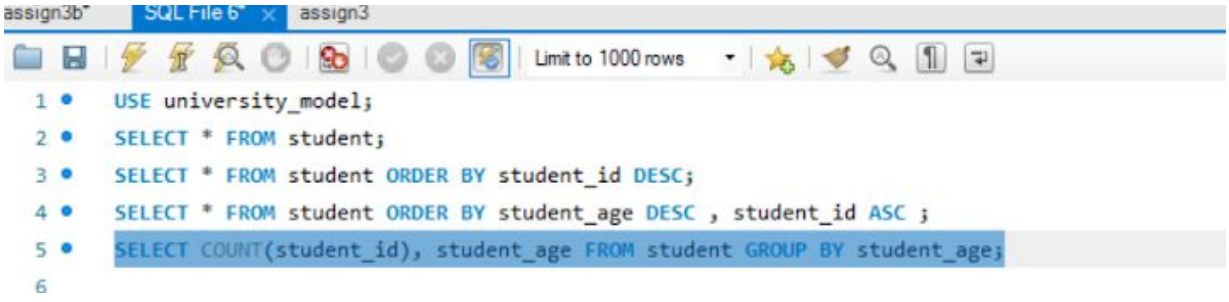


student_id	student_name	student_birthyear	student_age
4	geetha krishna	1999	21
2	naval	2000	20
3	avinash	2000	20
6	pranav	2000	20
1	lokesh	2001	19
5	aditya	2001	19
NULL	NULL	NULL	NULL

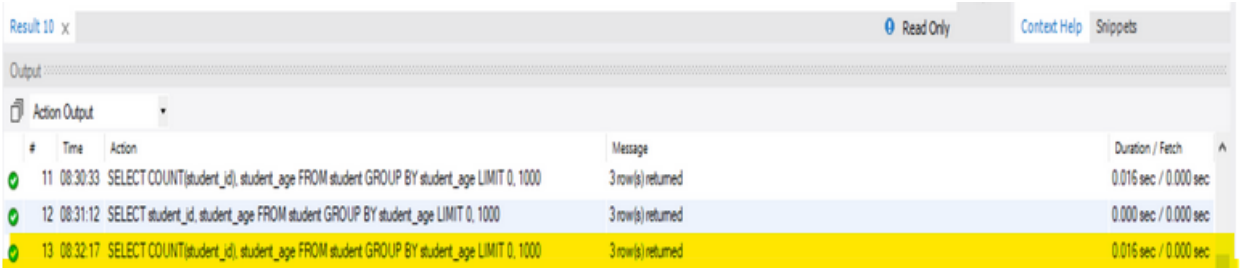
2.Group by and having

Query :

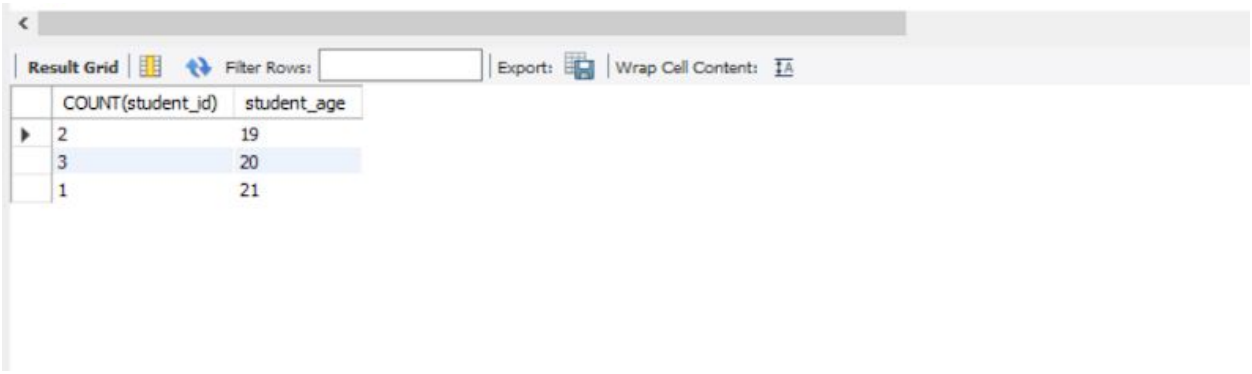
The below query groups the student table with their age and it shows how many people are there within the same age.



Output :



The output of the student's table, which is grouped according to the same ages.

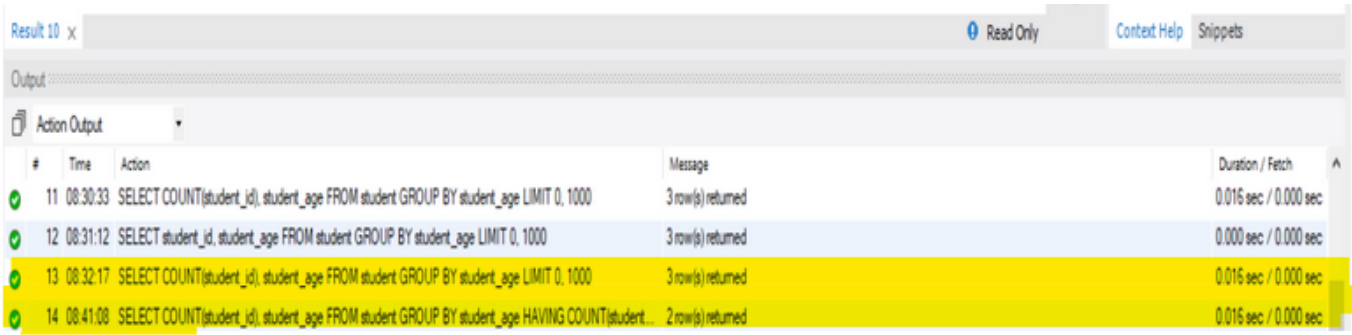


Query :

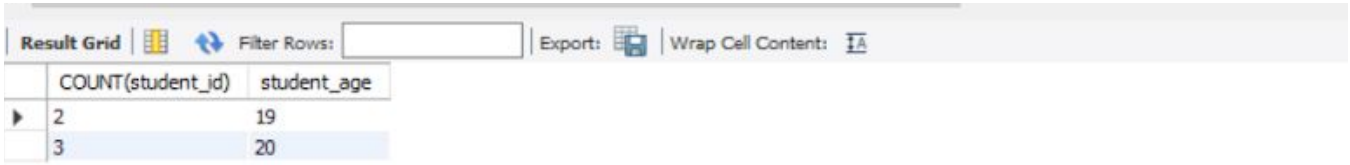
The below query counts the no. of students of the same age, and also having the student id greater than 1.

```
3  SELECT * FROM student ORDER BY student_id DESC;
4  SELECT * FROM student ORDER BY student_age DESC , student_id ASC ;
5  SELECT COUNT(student_id), student_age FROM student GROUP BY student_age;
6  SELECT COUNT(student_id), student_age FROM student GROUP BY student_age HAVING COUNT(student_id)>1;
7
8
```

Output :



The output shows the count of students of the same age.



3.Aggregate functions

Query :

The below query counts the total number of student_id is there in the table.

```
SELECT * FROM student ORDER BY student_id DESC;
SELECT * FROM student ORDER BY student_age DESC , student_id ASC ;
SELECT COUNT(student_id), student_age FROM student GROUP BY student_age;
SELECT COUNT(student_id), student_age FROM student GROUP BY student_age HAVING COUNT(student_id)>1;
SELECT COUNT(student_id) FROM student;
```

Output :

```
14  08:41:08  SELECT COUNT(student_id), student_age FROM student GROUP BY student_age HAVING COUNT(student...  2 row(s) returned  0.016 sec / 0.000 sec
15  09:22:29  SELECT COUNT(student_id) FROM student LIMIT 0, 1000                                              1 row(s) returned  0.000 sec / 0.000 sec
16  09:22:53  SELECT COUNT(student_id) FROM student LIMIT 0, 1000                                              1 row(s) returned  0.000 sec / 0.000 sec
```

The below table shows the total number of counts.

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

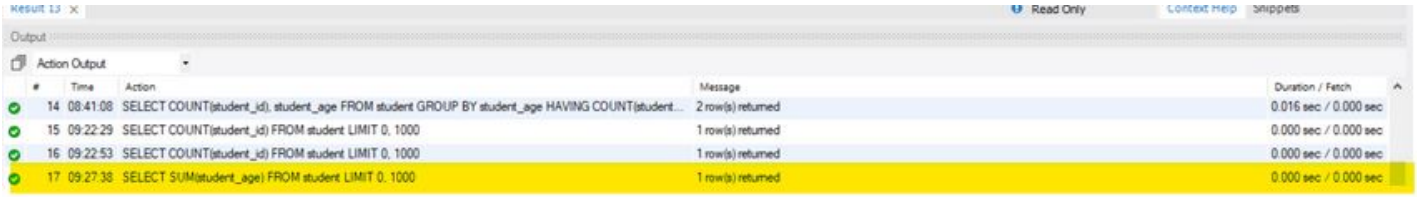
COUNT(student_id)
6

Query :

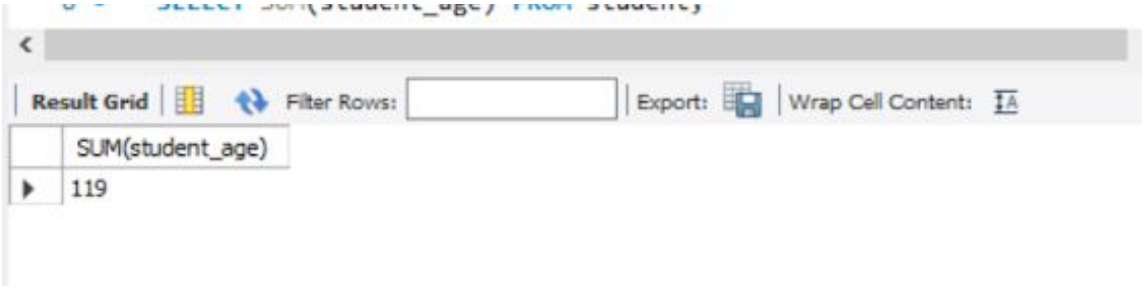
The below query shows the sum of ages of all students.

```
SELECT COUNT(student_id), student_age FROM student GROUP BY student_age;
SELECT COUNT(student_id), student_age FROM student GROUP BY student_age HAVING C
SELECT COUNT(student_id) FROM student;
SELECT SUM(student_age) FROM student;
```

Output :



Below is the total sum of all the student's ages.

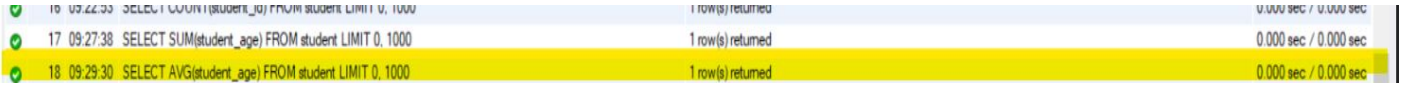


Query :

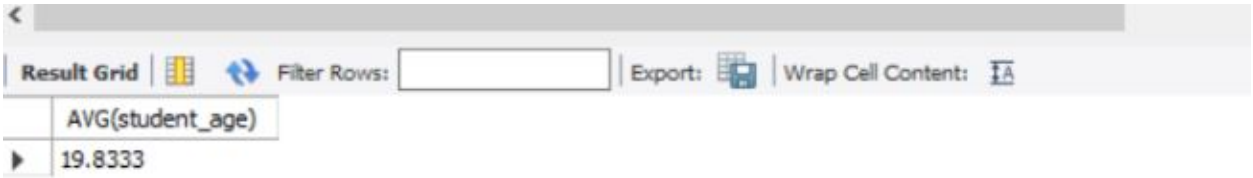
The below query shows the average of all the students_age within the student table.

```
SELECT COUNT(student_id) FROM student;
SELECT SUM(student_age) FROM student;
SELECT AVG(student_age) FROM student;
```

Output :



The avg of all the student's ages is shown below in output.



Query :

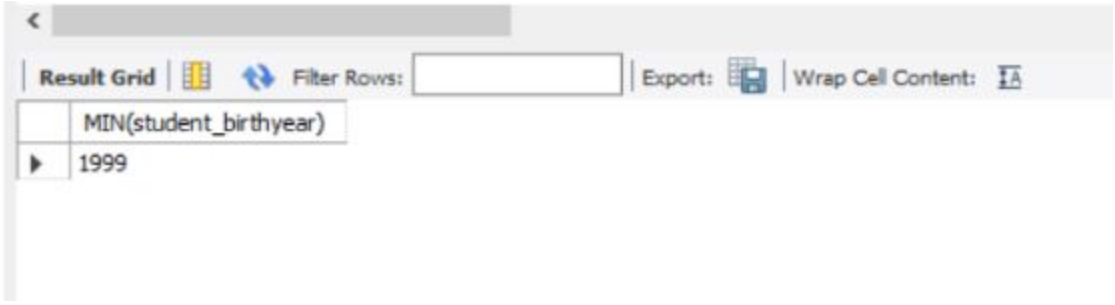
The below query shows the minimum of birthyear in the student id table.

```
SELECT COUNT(student_id) FROM student;
SELECT SUM(student_age) FROM student;
SELECT AVG(student_age) FROM student;
SELECT MIN(student_birthyear) FROM student;
```

Output :

#	Time	Action	Message	Duration / Fetch
17	09:27:38	SELECT SUM(student_age) FROM student LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
18	09:29:30	SELECT AVG(student_age) FROM student LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
19	09:30:50	SELECT MIN(student_birthyear) FROM student LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
20	09:31:58	SELECT MAX(student_birthyear) FROM student LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

The minimum birthyear in the student's table.

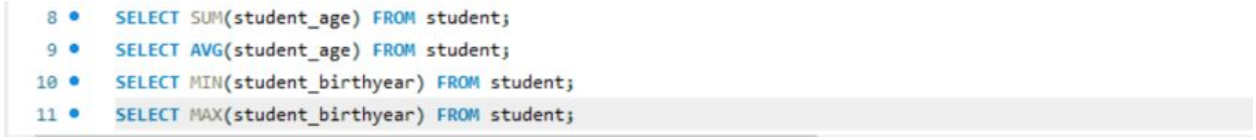


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

MIN(student_birthyear)
1999

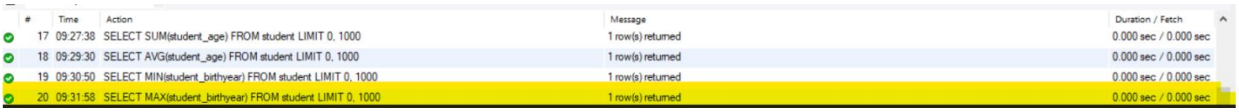
Query :

The below query shows the maximum of birthyear in the student id table.



```
8  SELECT SUM(student_age) FROM student;
9  SELECT AVG(student_age) FROM student;
10 SELECT MIN(student_birthyear) FROM student;
11 SELECT MAX(student_birthyear) FROM student;
```

Output :



#	Time	Action	Message	Duration / Fetch
17	09:27:38	SELECT SUM(student_age) FROM student LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
18	09:29:30	SELECT AVG(student_age) FROM student LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
19	09:30:50	SELECT MIN(student_birthyear) FROM student LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
20	09:31:58	SELECT MAX(student_birthyear) FROM student LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

The output shows the maximum birth year in the student table.

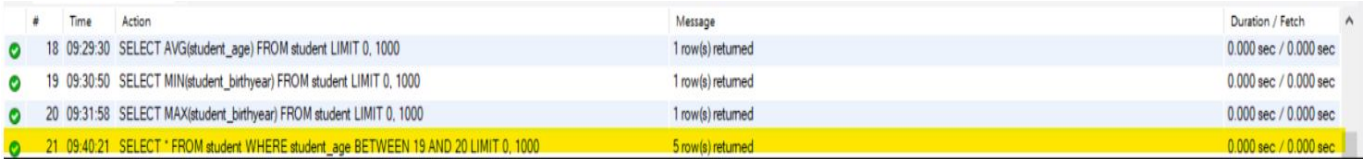
4. Logical operators especially with LIKE

Query :

The query selects the student age between 19 and 20.

```
10 • SELECT MIN(student_birthyear) FROM student;
11 • SELECT MAX(student_birthyear) FROM student;
12 • SELECT * FROM student WHERE student_age BETWEEN 19 AND 20;
```

Output :



#	Time	Action	Message	Duration / Fetch
18	09:29:30	SELECT AVG(student_age) FROM student LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
19	09:30:50	SELECT MIN(student_birthyear) FROM student LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
20	09:31:58	SELECT MAX(student_birthyear) FROM student LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
21	09:40:21	SELECT * FROM student WHERE student_age BETWEEN 19 AND 20 LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

The output shows the students age between 19 and 20.

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

student_id	student_name	student_birthyear	student_age
1	lokesh	2001	19
2	naval	2000	20
3	avinash	2000	20
5	aditya	2001	19
6	pranav	2000	20
NULL	NULL	NULL	NULL

Query :

The query selects the student age where student_id is <2. (ALL)

```
SELECT * FROM student
WHERE student_age > ALL (SELECT student_age FROM student WHERE student_id < 2);
```

Output :

#	Time	Action	Message	Duration / Fetch
21	09:40:21	SELECT * FROM student WHERE student_age BETWEEN 19 AND 20 LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
22	09:53:46	SELECT * FROM student WHERE student.age > ALL (SELECT student.age FROM student WHERE student...	Error Code: 1054. Unknown column 'student.age' in 'field list'	0.000 sec
23	09:54:34	SELECT * FROM student WHERE student_age > ALL (SELECT student_age FROM student WHERE studen...	0 row(s) returned	0.015 sec / 0.000 sec
24	09:55:39	SELECT * FROM student WHERE student_age > ALL (SELECT student_age FROM student WHERE studen...	4 row(s) returned	0.016 sec / 0.000 sec

The ages of all the students whose student_id <2 (ALL)

Result Grid | Filter Rows: | Edit: | Export/Import:

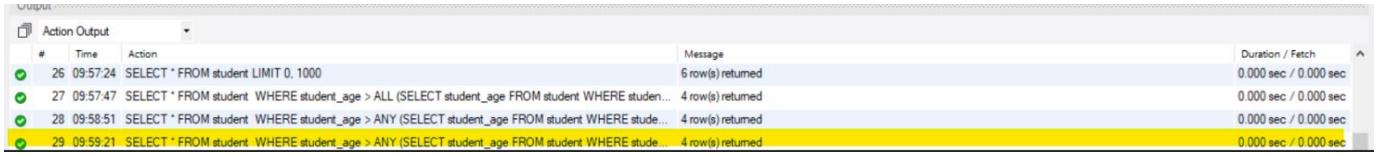
student_id	student_name	student_birthyear	student_age
2	naval	2000	20
3	avinash	2000	20
4	geetha krishna	1999	21
6	pranav	2000	20
NULL	NULL	NULL	NULL

Query :

The query selects the student age where student_id is greater than 2.(ANY)

```
SELECT * FROM student
WHERE student_age > ANY (SELECT student_age FROM student WHERE student_id >2);
```

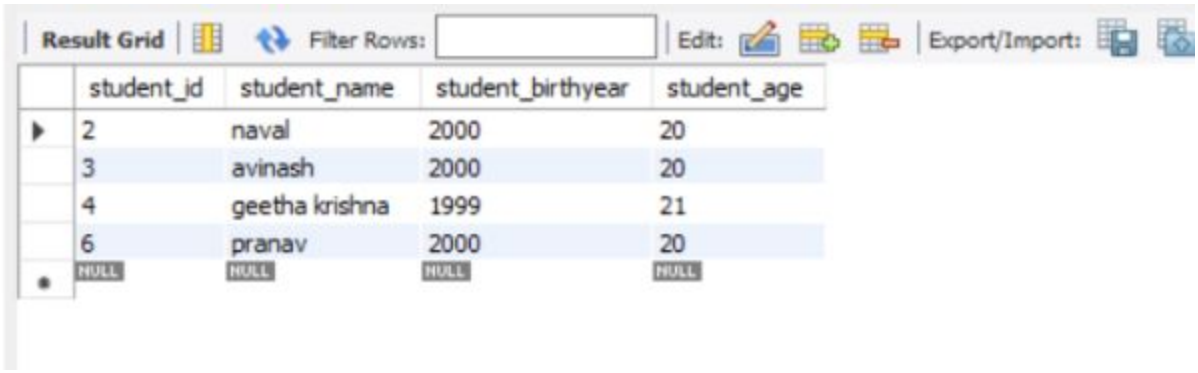
Output :



Action Output

#	Time	Action	Message	Duration / Fetch
26	09:57:24	SELECT * FROM student LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
27	09:57:47	SELECT * FROM student WHERE student_age > ALL (SELECT student_age FROM student WHERE studen...	4 row(s) returned	0.000 sec / 0.000 sec
28	09:58:51	SELECT * FROM student WHERE student_age > ANY (SELECT student_age FROM student WHERE stude...	4 row(s) returned	0.000 sec / 0.000 sec
29	09:59:21	SELECT * FROM student WHERE student_age > ANY (SELECT student_age FROM student WHERE stude...	4 row(s) returned	0.000 sec / 0.000 sec

The ages of all the students whose student_id >2 (ANY)



Result Grid | Filter Rows: | Edit: | Export/Import:

student_id	student_name	student_birthyear	student_age
2	naval	2000	20
3	avinash	2000	20
4	geetha krishna	1999	21
6	pranav	2000	20
NULL	NULL	NULL	NULL

Query :

The below query selects the names of students whose name starts with “a”.

```
WHERE student_age > ANY (SELECT student_age FROM student WHERE student_id >2);
SELECT * FROM student WHERE student_name LIKE 'a%';
```

Output :

#	Time	Action	Message	Duration / Fetch
28	09:58:51	SELECT * FROM student WHERE student_age > ANY (SELECT student_age FROM student WHERE stude...	4 row(s) returned	0.000 sec / 0.000 sec
29	09:59:21	SELECT * FROM student WHERE student_age > ANY (SELECT student_age FROM student WHERE stude...	4 row(s) returned	0.000 sec / 0.000 sec
30	10:00:49	SELECT * FROM student WHERE student_name LIKE 'a%' LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
31	10:01:12	SELECT * FROM student WHERE student_name LIKE 'a%' LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

Below is the Output of all students whose name starts with a.

Result Grid | Filter Rows: | Edit: | Export/Import:

student_id	student_name	student_birthyear	student_age
3	avinash	2000	20
5	aditya	2001	19
NULL	NULL	NULL	NULL

5.At least 4 Nested queries specific to your Database, out of which at least 2 should have multiple subqueries.

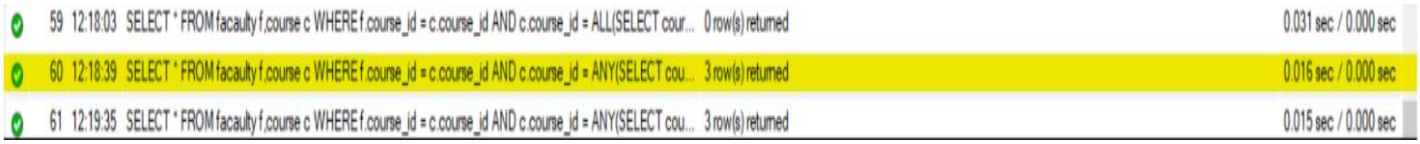
Query :

The below queries selects all faculty members who teach all courses in the CSE department. (Multiple Sub Query)

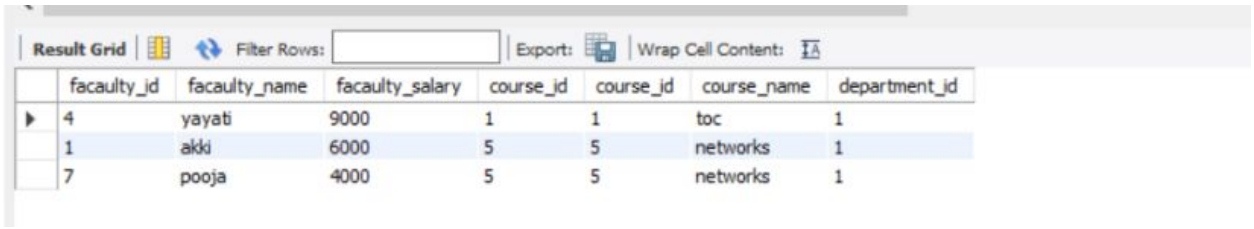
```
SELECT * FROM facaulty f,course c WHERE f.course_id = c.course_id AND c.course_id =
ANY(SELECT course_id FROM course e,department g WHERE e.department_id = g.department_id AND g.department_id=
  (SELECT department_id FROM department where department_name='cse'
  )

);
```

Output :



Below is the output of all the faculty members who teaches all courses in the CSE department.



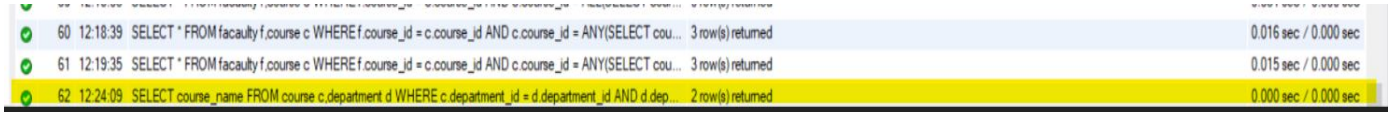
facaulty_id	facaulty_name	facaulty_salary	course_id	course_id	course_name	department_id
4	yayati	9000	1	1	toc	1
1	akki	6000	5	5	networks	1
7	pooja	4000	5	5	networks	1

Query :

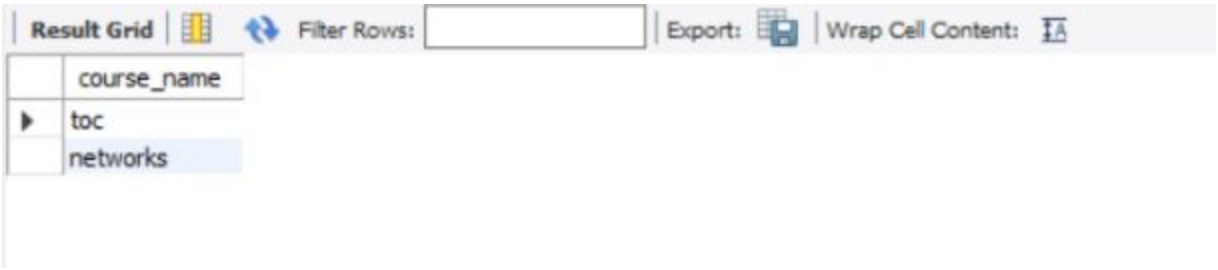
The below query selects all the courses in the CSE department.

```
SELECT course_name FROM course c,department d WHERE c.department_id = d.department_id AND d.department_id =
(SELECT department_id FROM department WHERE department_name = 'cse'
);
```

Output :



The below output is of all the courses in the CSE department.



course_name
toc
networks

Query :

The below query selects all the books in the library which belongs to the CSE department.

```
);
SELECT * FROM library l,department d WHERE l.department_id = d.department_id AND d.department_id=
(SELECT department_id FROM department WHERE department_name = 'cse'
);
```

Output :

```
64  12:39:16  SELECT * FROM library l,department d,student s WHERE l.department_id = (SELECT department_id FROM d...  24 row(s) returned  0.000 sec / 0.000 sec
65  12:40:20  SELECT * FROM library l,department d,student s WHERE l.department_id = (SELECT department_id FROM d...  48 row(s) returned  0.016 sec / 0.000 sec
66  12:42:23  SELECT * FROM library l,department d WHERE l.department_id = d.department_id AND d.department_id= (SE...  2 row(s) returned  0.000 sec / 0.000 sec
```

The output is the books which belong to the CSE department

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

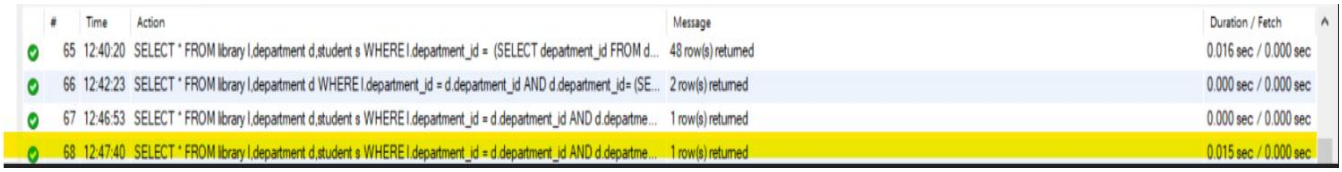
book_id	book_name	department_id	student_id	department_id	department_name
1	tcomputation	1	1	1	cse
3	database	1	2	1	cse

Query :

The below query selects the minimum student id and also the CSE department id and it shows the book details which were taken by him from the library. (Multiple Sub Query)

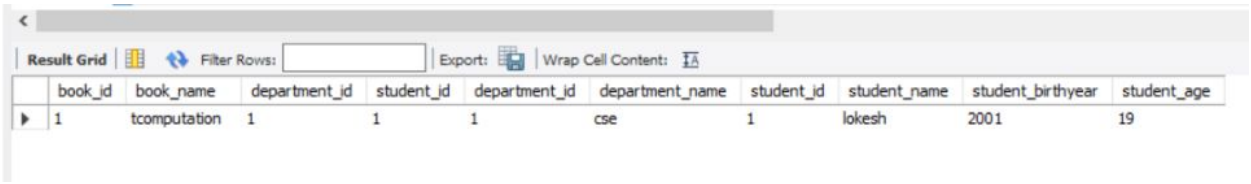
```
);
SELECT * FROM library l,department d,student s WHERE l.department_id = d.department_id AND d.department_id=
(SELECT department_id FROM department WHERE department_name = 'cse'
) AND l.student_id = s.student_id AND s.student_id =
(SELECT MIN(student_id) FROM student
);
```

Output :



#	Time	Action	Message	Duration / Fetch
65	12:40:20	SELECT * FROM library l,department d,student s WHERE l.department_id = (SELECT department_id FROM d...	48 row(s) returned	0.016 sec / 0.000 sec
66	12:42:23	SELECT * FROM library l,department d WHERE l.department_id = d.department_id AND d.department_id= (SE...	2 row(s) returned	0.000 sec / 0.000 sec
67	12:46:53	SELECT * FROM library l,department d,student s WHERE l.department_id = d.department_id AND d.departme...	1 row(s) returned	0.000 sec / 0.000 sec
68	12:47:40	SELECT * FROM library l,department d,student s WHERE l.department_id = d.department_id AND d.departme...	1 row(s) returned	0.015 sec / 0.000 sec

The output shows all the details of the selected student.



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

book_id	book_name	department_id	student_id	department_id	department_name	student_id	student_name	student_birthyear	student_age
1	tcomputation	1	1	1	cse	1	lokesh	2001	19