DBMS Project

Board Examination Management System

Sowmya Potluri (u4cse17314)
Sreya Sukavasi (u4cse177330)
Sunkari Akshaya (u4cse17332)
Sindhuja Dantuluri (u4cse17503)

Board Examination Management System

Project Abstract

Objective:

Our project aims at developing a database management system to manage CBSE Board Examinations of 10th grade student

Core Features and Scope:

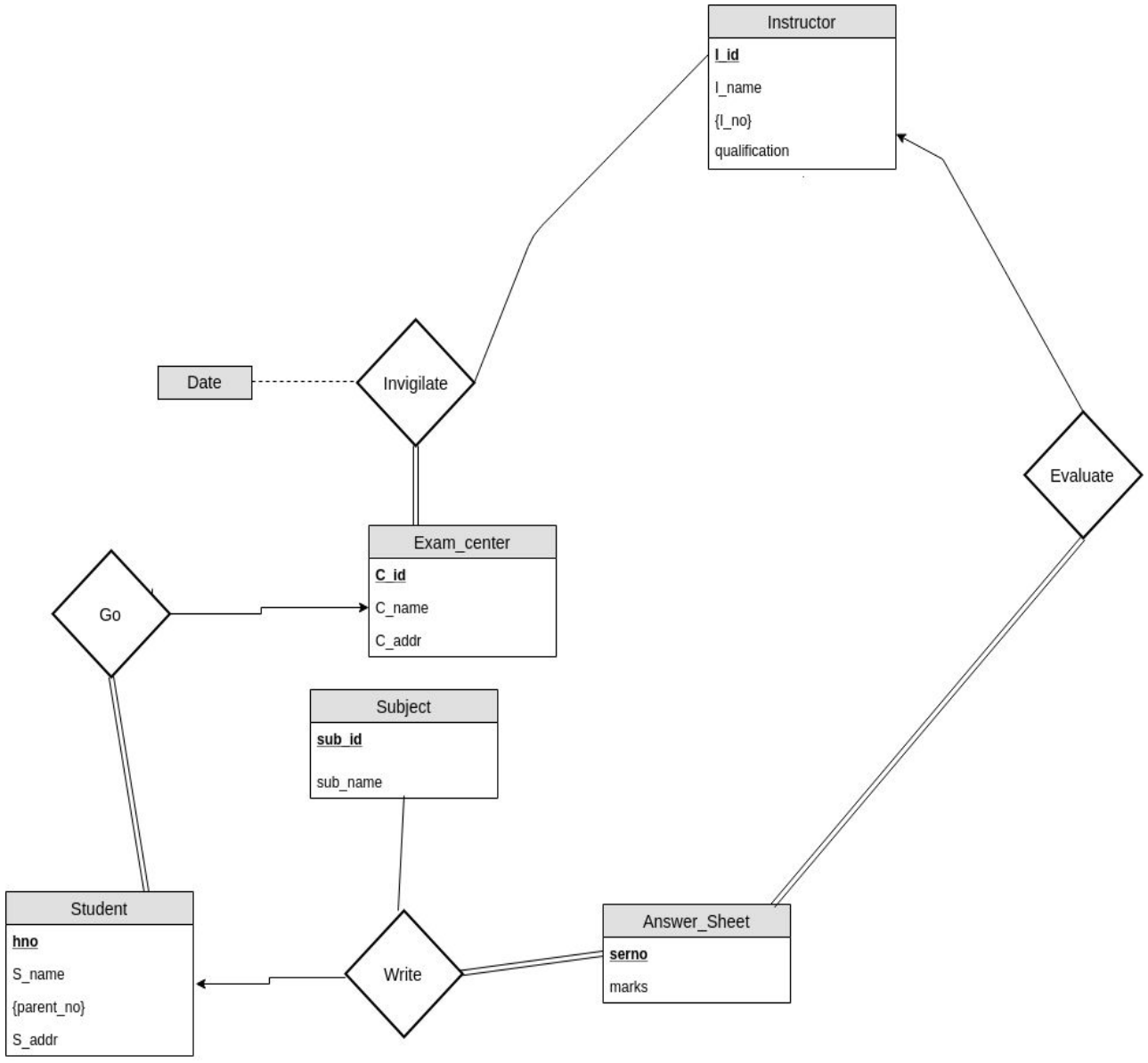
The application stores details of all the students who have registered for the exam including hall ticket number,exam center details and address.
It also stores details of all examiners who can either evaluate answer papers or invigilate in exam centers during examination. Each answer script is evaluated only by one instructor and the details regarding this is managed by the application. Students can always view their marks in particular subjects and the overall average marks of subject. An examiner can correct as many papers as possible.
Central Board assigns an instructor with two kinds of duties, either invigilation in exam centers or evaluating answer sheets or both.

Team Members:

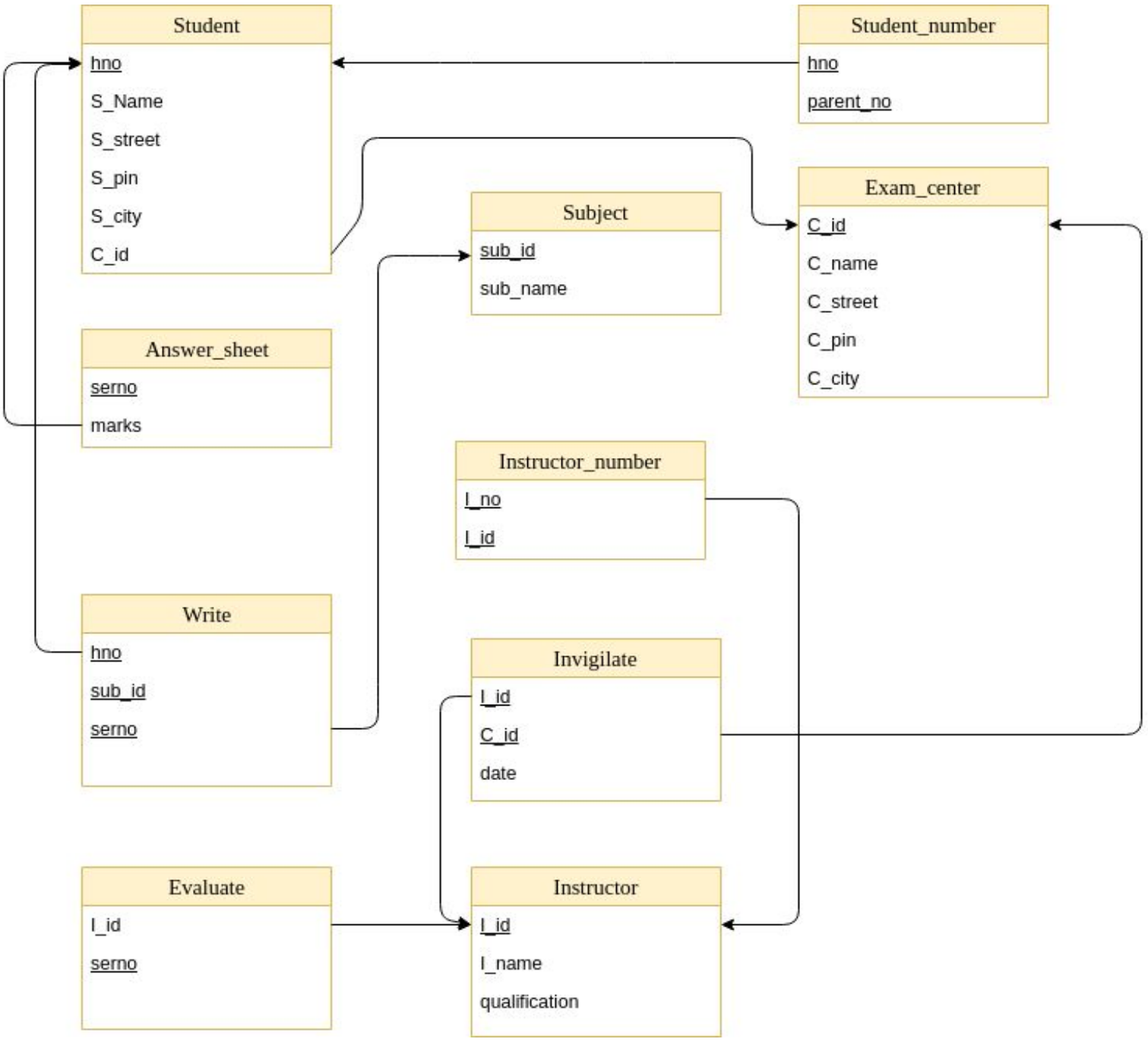
- P. Sowmya - U4CSE17314
- S.Sreya - U4CSE17330
- S.Akshaya - U4CSE17332
- D.Sindhuja - U4CSE17503

Board Exam Management System

ER Diagram



Schema Diagram



Relational Schema: Board Exam Management System

1. Student(hno, S_name, S_street, S_pin, S_city, C_id)
2. Student_number(hno, parent_no)
3. Subject(sub_id, sub_name)
4. Write(hno, sub_id,serno)
5. Answer_sheet(serno, marks,I_id)
6. Exam_center(C_id, C_name, C_street, C_pin, C_city)
7. Invigilate(I_id, C_id, date)
8. Instructor(I_id, I_name, qualification)
9. Instructor_number(I_id, I_no)

Normalisation

0NF

I_id	I_name	qualification	I_no	date	C_name
101	Usha	MTech	7594098292 7594098297	02-04-2000	askii

C_addr	serno	sub_id	sub_name	hno	S_name
abids,hyd	101625	301	English	1523456	riya

S_addr	parent_no	marks
Ashok nagar,hyd	9177849188 9246579455	50

ASSUMPTIONS AND CONSTRAINTS

1. I_id - A unique instructor id
2. I_name - Name of the instructor need not be unique and it cannot be decomposed that is it is not a composite attribute
3. qualification - Qualification of the instructor need not be unique
4. I_no- Phone number/s of instructor
5. date - centre is assigned to instructor on this date , need not be unique
6. C_name- name of the centre
7. C_addr- address of the centre
8. serno- a unique serial number of the answer sheet which cannot be repeated at all

9. sub_id - subject id which is unique
10. sub_name- subject name, which is not unique
11. hno- hall ticket number of the student taking exam
12. s_name - name of the student taking exam
13. s_addr- address of the student
14. parent_no- phone number/s of the student's parents
15. Each street has unique pincode and street names are unique

1NF

Board_Exam_Table(I_id, I_name, qualification, sub_id, serno, hno, marks, C_name, street, C_city, S_name, S_city, street, parent_no, I_no, date, sub_name)

Functional Dependencies

I_id -> I_name
I_id -> qualification
I_id-> I_no
serno -> marks
sub_id-> sub_name
hno-> S_name
hno->S_addr
hno->parent_no
C_id->C_name
C_id->street
C_id->C_city

2NF

Primary key for 1NF table is (I_id, serno, sub_id, hno, C_id)

Instructor attributes are dependent only on I_no hence they are separated and c_id, date attributes also have partial dependency. Hence, it is meaningful to keep them in instructor table.

Instructor(I_id, I_name, qualification, I_no, C_id, date)

Exam_Centre(C_id, C_name, street, C_city, pin)

Subject attributes are only dependent on sub_id, hence there exists a partial dependency and they have to be in a separate table.

Subject(sub_id, sub_name)

Marks depend only on the serial number of the answer sheet since it is unique, hence partial dependency exists and it has to be separated

Answer_sheet(serno, marks)

Student details depend only on hno therefore partial dependency exists, it has to be in a separate table

Student(hno, S_name, street, S_city, pin, parent_no, C_id)

Join tables-

Since, few foreign keys are removed from the above tables, few relations like student writing answer sheet and instructor evaluating answer sheet are missed.
Hence, join tables are created using primary keys of different tables as foreign keys.

write(hno, sub_id, serno)

evaluate(serno, I_id)

All these tables are in 2NF form.

3NF

All the above tables are in 3NF except instructor table and student table where pin codes of the address depend on street which are no key attributes. Hence they have to be split.

Student(hno, S_name, street, S_city, C_id, parent_no)

Subject(sub_id, sub_name)

Write(hno, S_id, serno)

Answer_sheet(serno, marks)

Exam_center(C_id, C_name, street, C_city)

Evaluate(serno, I_id)

Instructor(I_id, I_name, qualification, I_no)

pincode(pin, street)

Observations and Comparisons

Tables obtained through ER design and normalisation are almost similar except for few tables. In the case of normalization there is less data redundancy and more database optimization where as the main functional dependencies are the same in both cases as it is
related to logical design.

DDL STATEMENTS

```
Create table Exam_center(
C_id  varchar(5),
C_name  varchar(15),
c_street varchar(10),
c_city varchar(10),
c_pin varchar(10),
Primary key(C_id)
);

Create table Student(
```

```
C_id  varchar(5),
hno varchar(15) not null ,
S_name varchar(10),
s_street varchar(10),
s_city varchar(10),
s_pin varchar(10),
Primary key(hno),
Foreign key(C_id) references Exam_center
);

Create table Student_Number(
hno varchar(15) not null ,
parent_no varchar(10),
Primary key(hno,parent_no),
Foreign key(hno) references student
);

Create table Subject(
sub_id varchar(10) ,
sub_name varchar(15),
Primary key(sub_id)
);

Create table instructor(
I_id varchar(10) ,
I_name varchar(15),
qualification  varchar(10) not null,
Primary key(I_id)
);

Create table instructor_number(
I_id varchar(10) ,
I_no varchar(10),
Primary key(I_id,i_no),
Foreign key(i_id) references instructor
);

Create  table answer_sheet(
serno varchar(10) not null ,
marks numeric(4,2),
Primary key(serno),
I_id varchar(10) ,
Foreign key(i_id) references instructor
);

Create table write(
```

```
hno varchar(15),
sub_id varchar(10) ,
serno varchar(10) not null ,
Foreign key(hno) references student,
Foreign key(sub_id) references subject,
Foreign key(serno) references answer_sheet,
Primary key(hno,sub_id,serno)
);

Create table invigilate(
I_id varchar(10) ,
C_id  varchar(5),
date date,
Primary key(I_id,c_id),
Foreign key(i_id) references instructor,
Foreign key(c_id) references exam_center
);
```

Inserting into DataBase

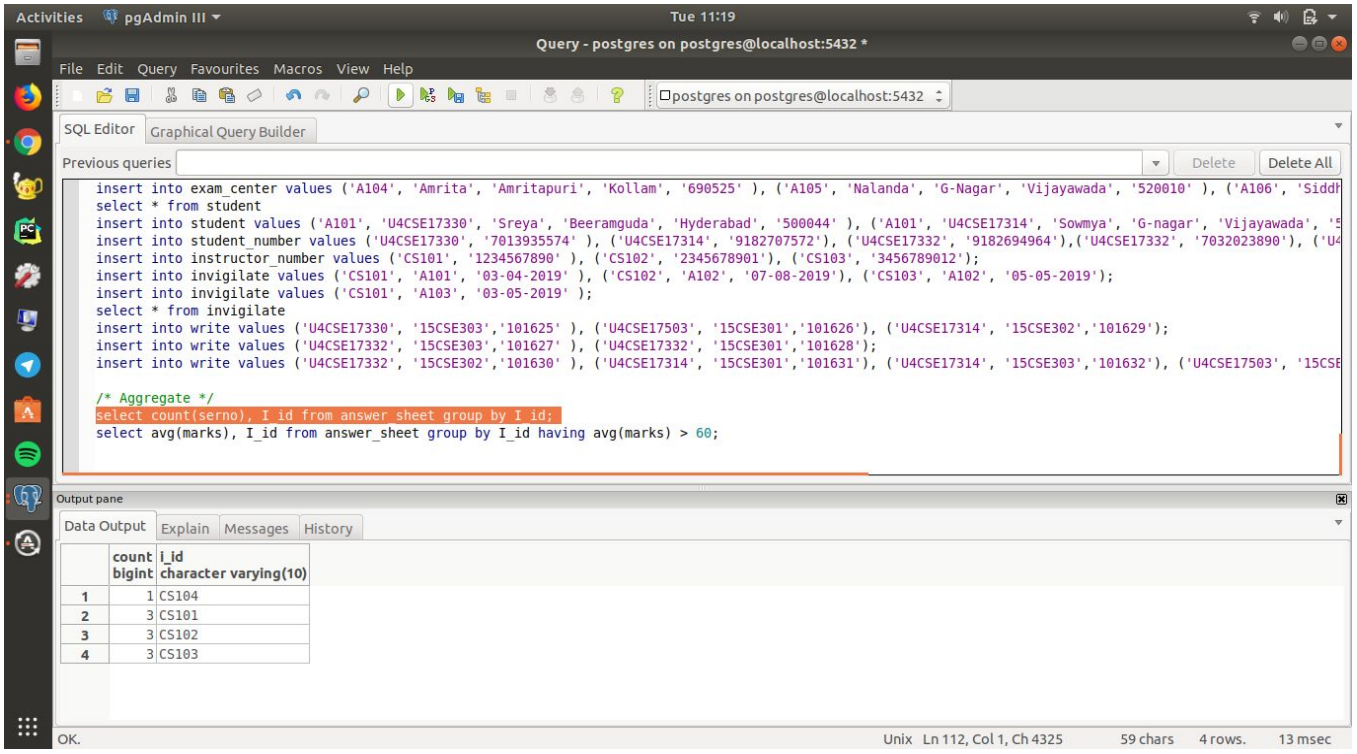
```
insert into subject values ('15CSE302', 'DBMS' ), ('15CSE303', 'TOC'), ('15CSE301', 'COA');
insert into subject values ('15CSE431', 'FDS' ), ('15MAT301', 'MAT'), ('15EVS300', 'EVS');
select * from instructor
insert into instructor values ('CS101', 'Ms.Jasmine', 'MTech' ), ('CS102', 'Dr.Jyothishya', 'PHD'),
('CS103', 'Dr.Gopa Kumar', 'PHD in ML');
insert into instructor values ('CS104', 'Ms.Maya', 'MTech' );
select * from answer_sheet
insert into answer_sheet values ('101625', '80', 'CS103' ), ('101626', '86', 'CS102'), ('101627', '60',
'CS101');
insert into answer_sheet values ('101634', '80', 'CS104' );
insert into answer_sheet values ('101628', '60', 'CS103' ), ('101629', '20', 'CS102'), ('101630', '92',
'CS101'), ('101631', '45', 'CS101'), ('101632', '67', 'CS102'), ('101633', '65', 'CS103') ;
select * from exam_center
insert into exam_center values ('A101', 'Amrita', 'Vallikavu', 'Kollam', '690525' ), ('A102', 'Narayana',
'Patamata', 'Vijayawada', '520010' ), ('A103', 'Chaitanya', 'Madhapur', 'Hyderabad', '500020' );
insert into exam_center values ('A104', 'Amrita', 'Amritapuri', 'Kollam', '690525' ), ('A105', 'Nalanda',
'G-Nagar', 'Vijayawada', '520010' ), ('A106', 'Siddhartha', 'BenzCircle', 'Vijayawada', '520010' );
select * from student
insert into student values ('A101', 'U4CSE17330', 'Sreya', 'Beeramguda', 'Hyderabad', '500044' ),
('A101', 'U4CSE17314', 'Sowmya', 'G-nagar', 'Vijayawada', '520010'), ('A103', 'U4CSE17503',
'Sindhuja', 'RNagar', 'Hyderabad', '500045'), ('A102', 'U4CSE17332', 'Akshaya', 'Patamata',
'Vijayawada', '520010');
insert into student_number values ('U4CSE17330', '7013935574' ), ('U4CSE17314', '9182707572'),
('U4CSE17332', '9182694964'),('U4CSE17332', '7032023890'), ('U4CSE17503', '7594098292');
insert into instructor_number values ('CS101', '1234567890' ), ('CS102', '2345678901'), ('CS103',
'3456789012');
```

insert into invigilate values ('CS101', 'A101', '03-04-2019'), ('CS102', 'A102', '07-08-2019'), ('CS103', 'A102', '05-05-2019');
insert into invigilate values ('CS101', 'A103', '03-05-2019');
select * from write
insert into write values ('U4CSE17330', '15CSE303','101625'), ('U4CSE17503', '15CSE301','101626'), ('U4CSE17314', '15CSE302','101629');
insert into write values ('U4CSE17332', '15CSE303','101627'), ('U4CSE17332', '15CSE301','101628');
insert into write values ('U4CSE17332', '15CSE302','101630'), ('U4CSE17314', '15CSE301','101631'), ('U4CSE17314', '15CSE303','101632'), ('U4CSE17503', '15CSE302','101633'), ('U4CSE17330', '15CSE301','101634');

Queries

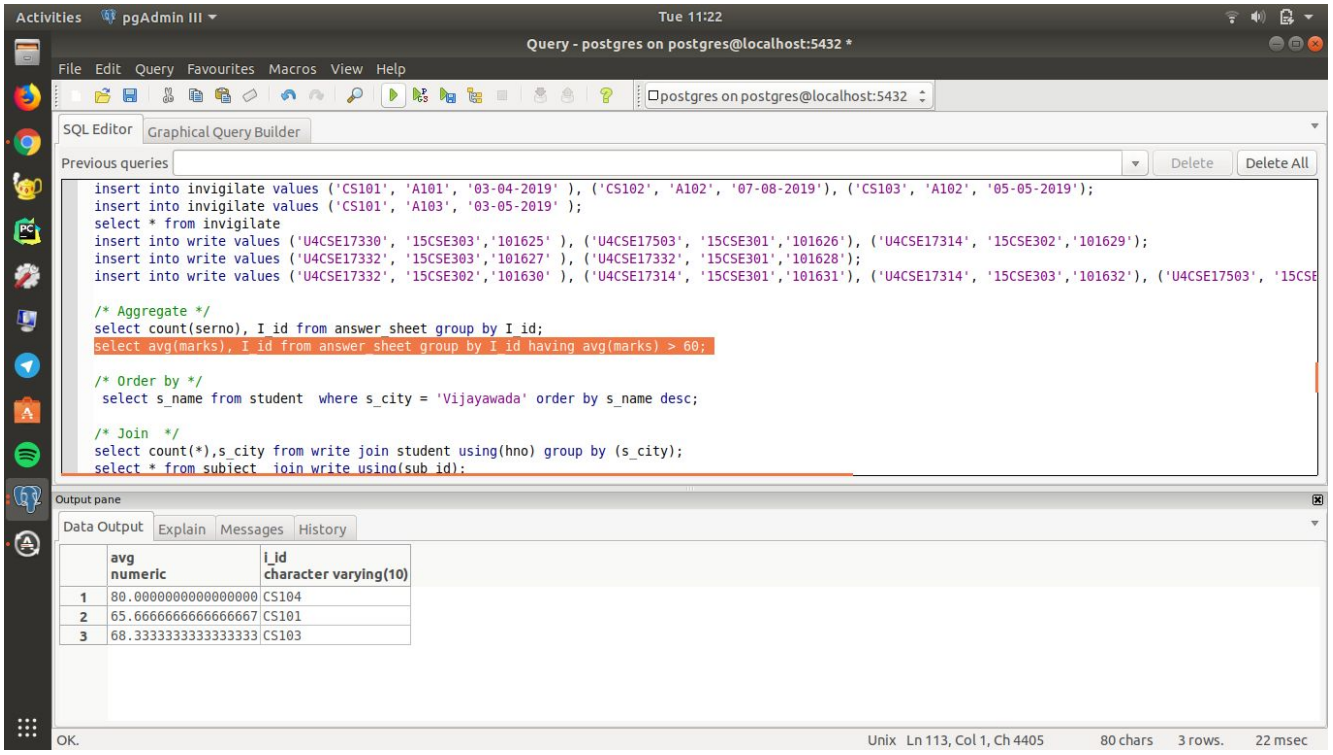
- Count of the answer sheets evaluated by each instructor:-

  select count(serno), l_id from answer_sheet group by l_id;



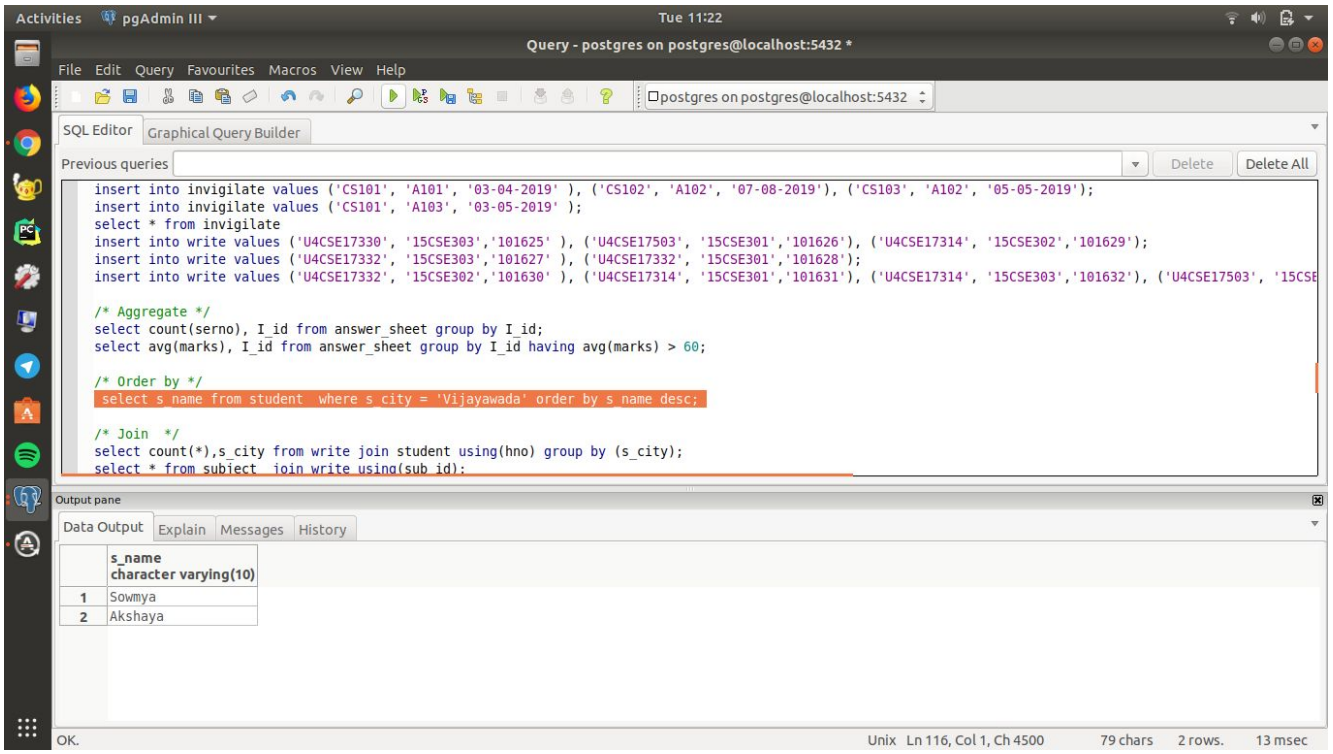
- Display instructor id and avg marks where average marks of all the papers corrected by an instructor is greater than 60

select avg(marks), I_id from answer_sheet group by I_id having avg(marks) > 60;



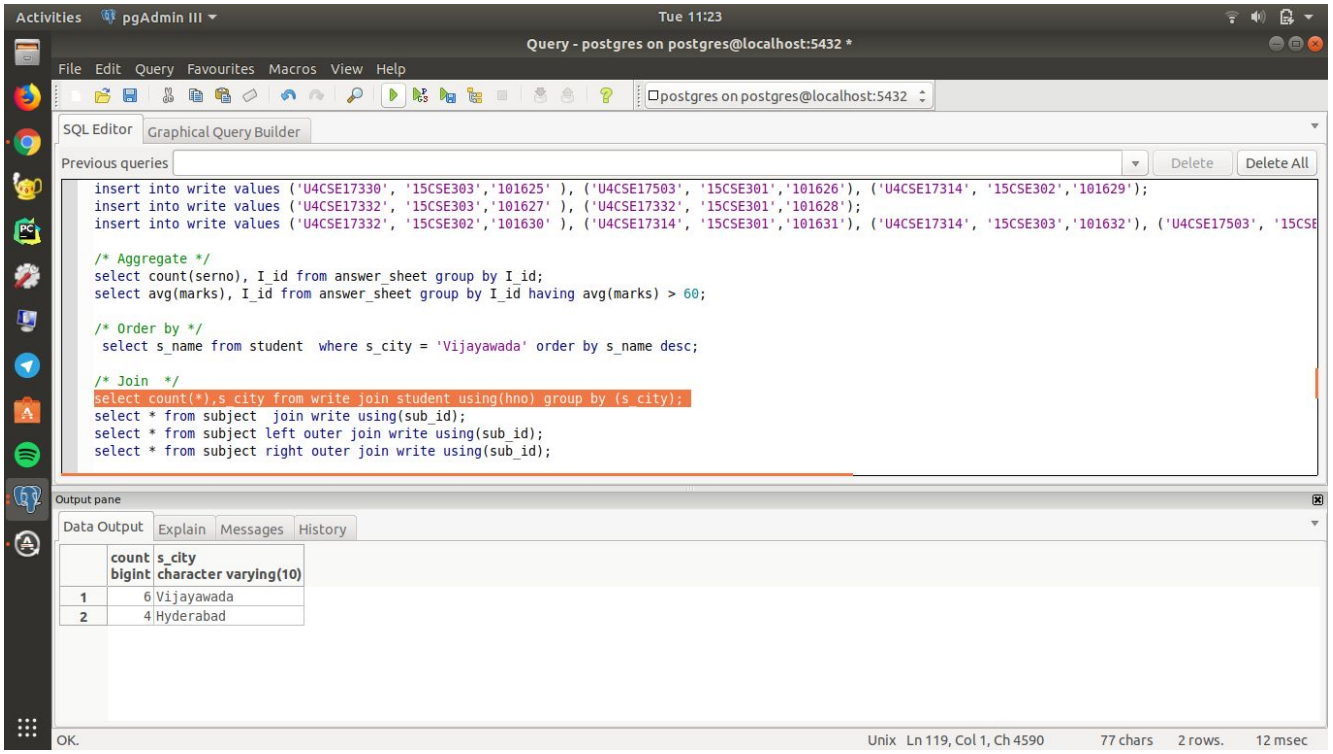
- Display all the names of the students sorted in descending alphabetical order whose school is in Vijayawada

select s_name from student where s_city = 'Vijayawada' order by s_name desc;

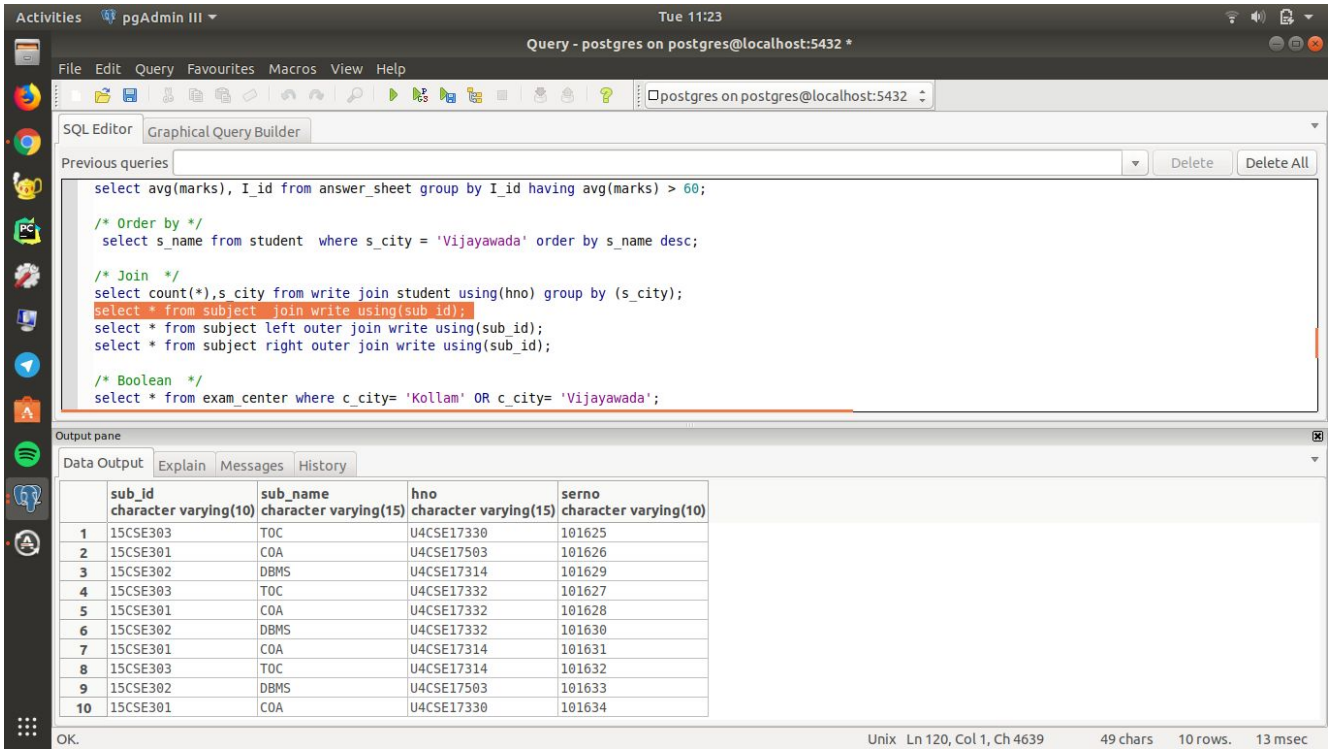


- Joining tables

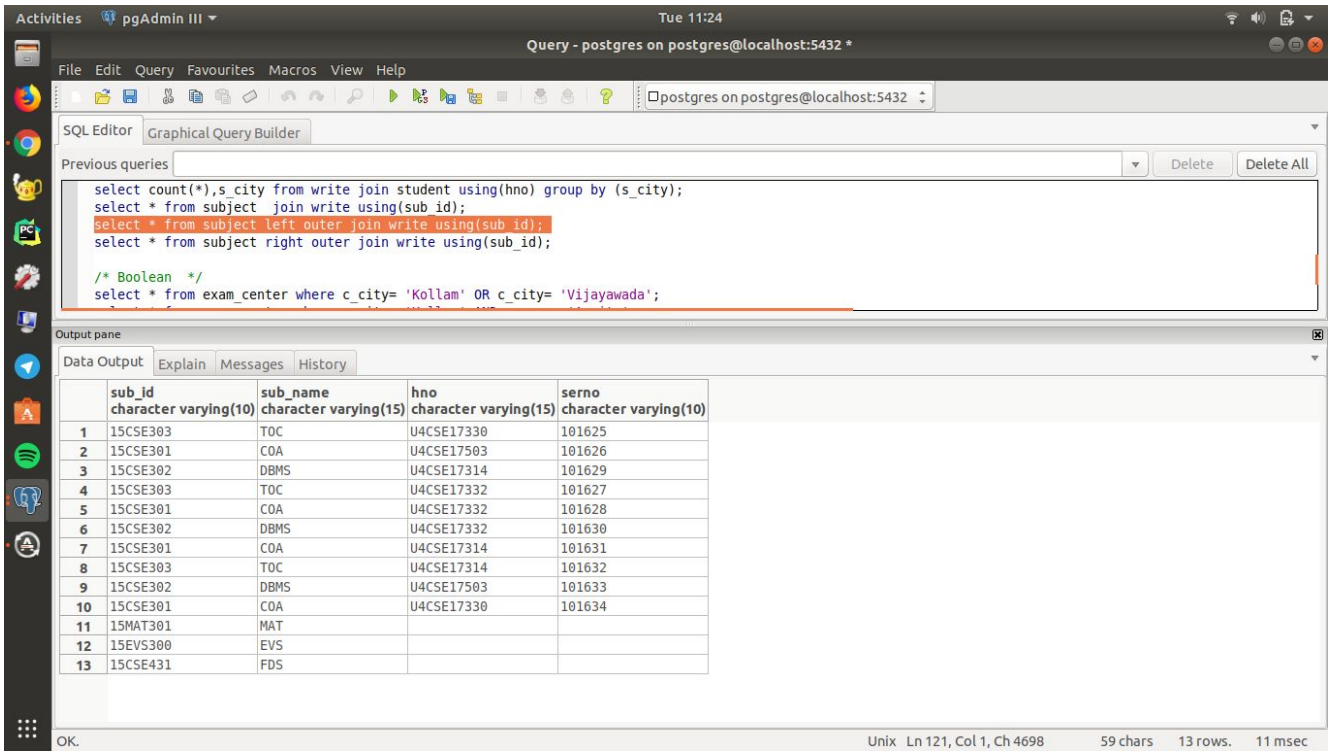
select count(*),s_city from write join student using(hno) group by (s_city);



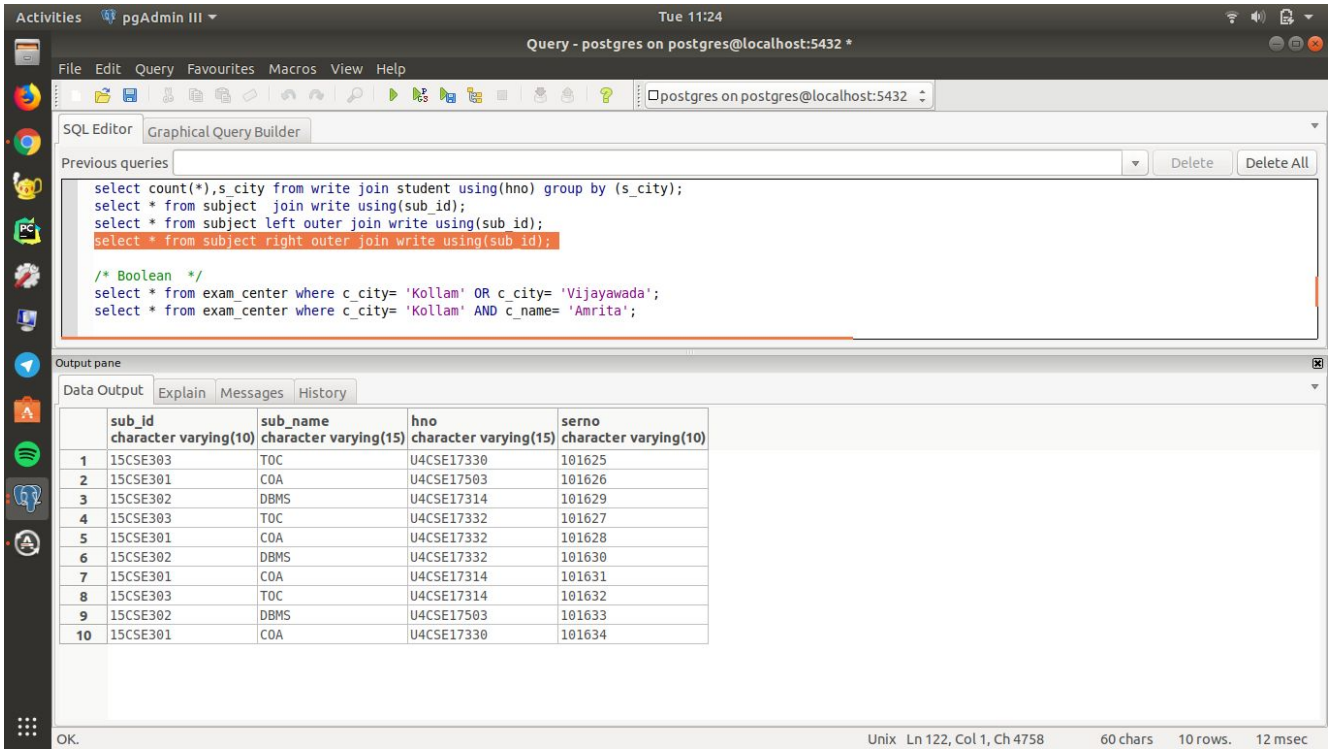
select * from subject join write using(sub_id);



select * from subject left outer join write using(sub_id);

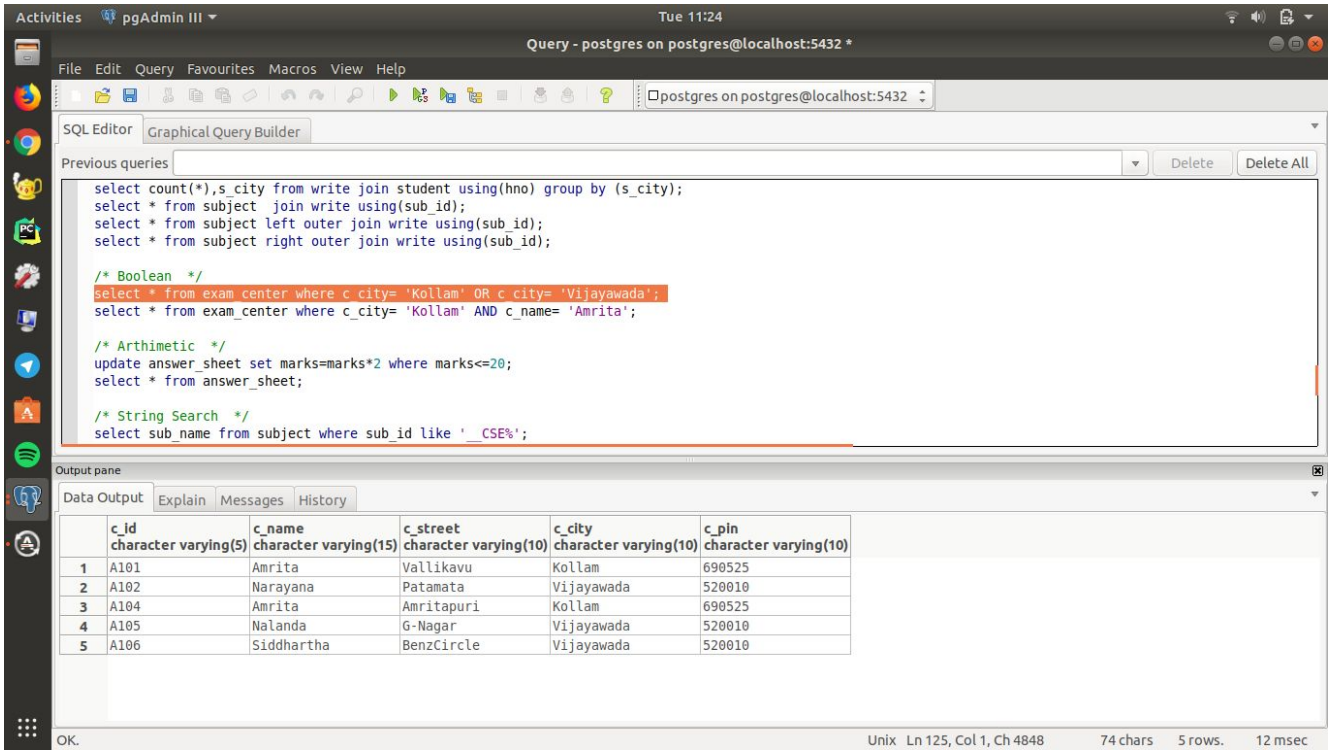


select * from subject right outer join write using(sub_id);



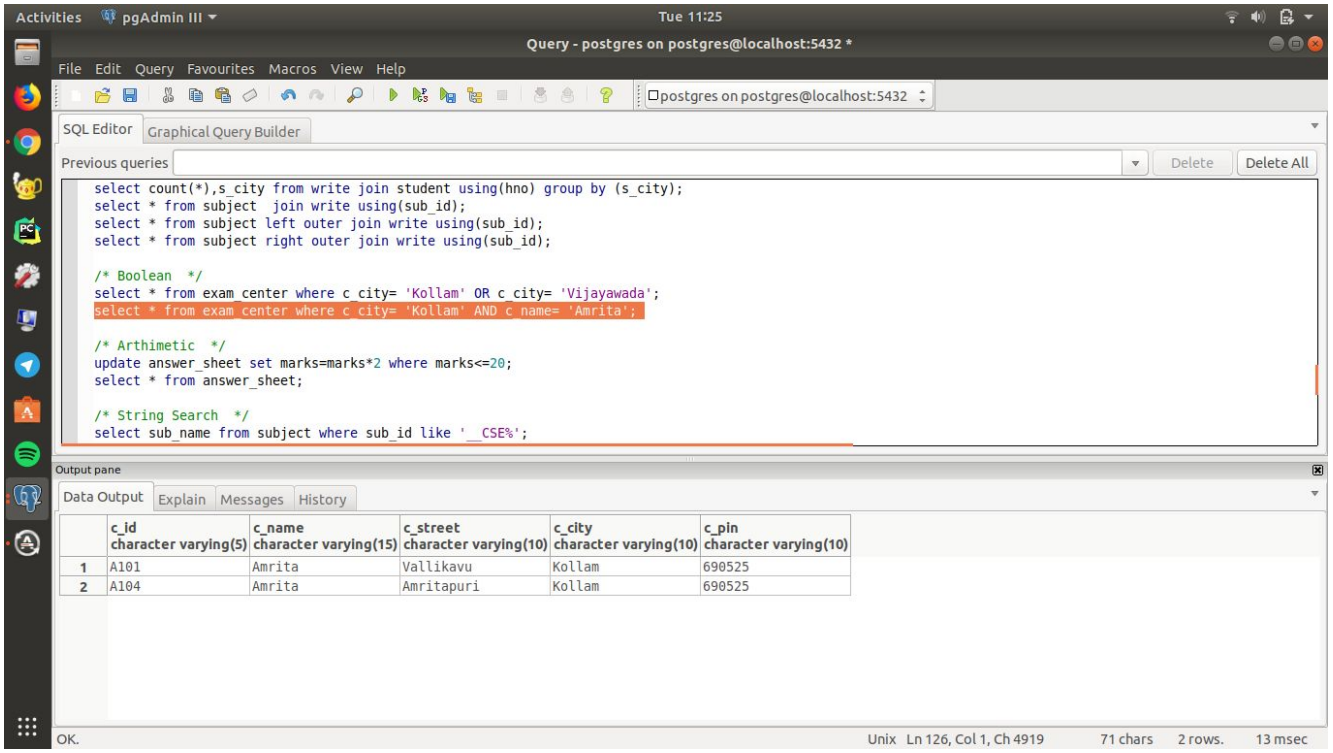
- Display all the exam centers which are in either kollam or Vijayawada

select * from exam_center where c_city= 'Kollam' OR c_city= 'Vijayawada';



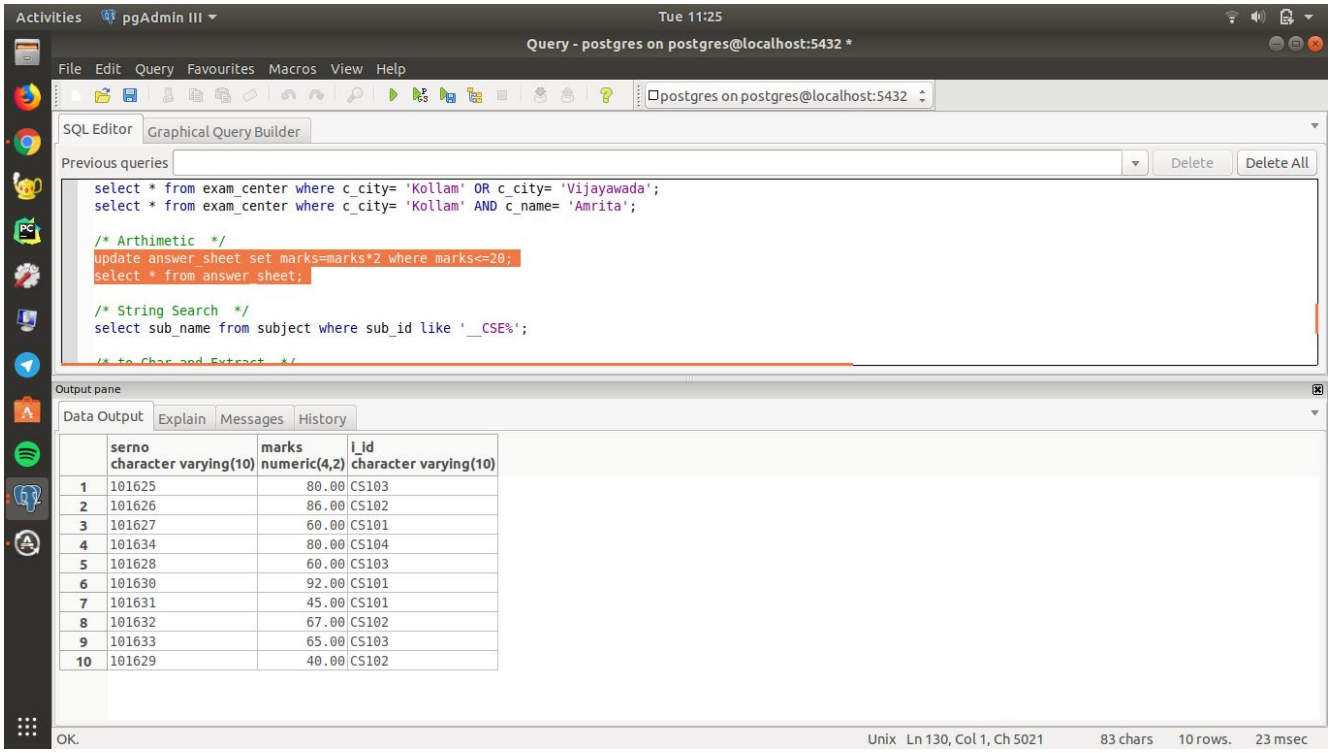
- Display all the exam centers with name amrita and which is present in kollam

select * from exam_center where c_city= 'Kollam' AND c_name= 'Amrita';



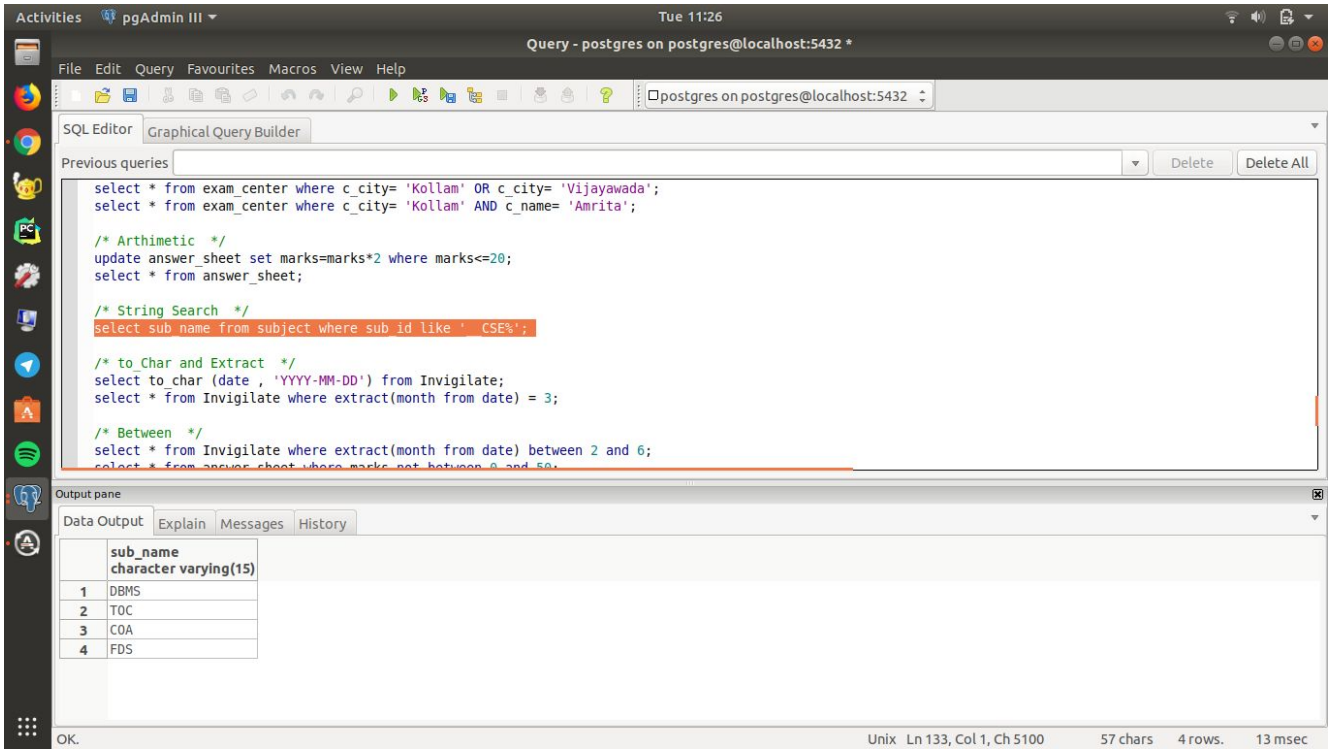
- Update marks of all answer sheets if the marks are less than 20 by multiplying it by 2

update answer_sheet set marks=marks*2 where marks<=20;
select * from answer_sheet;



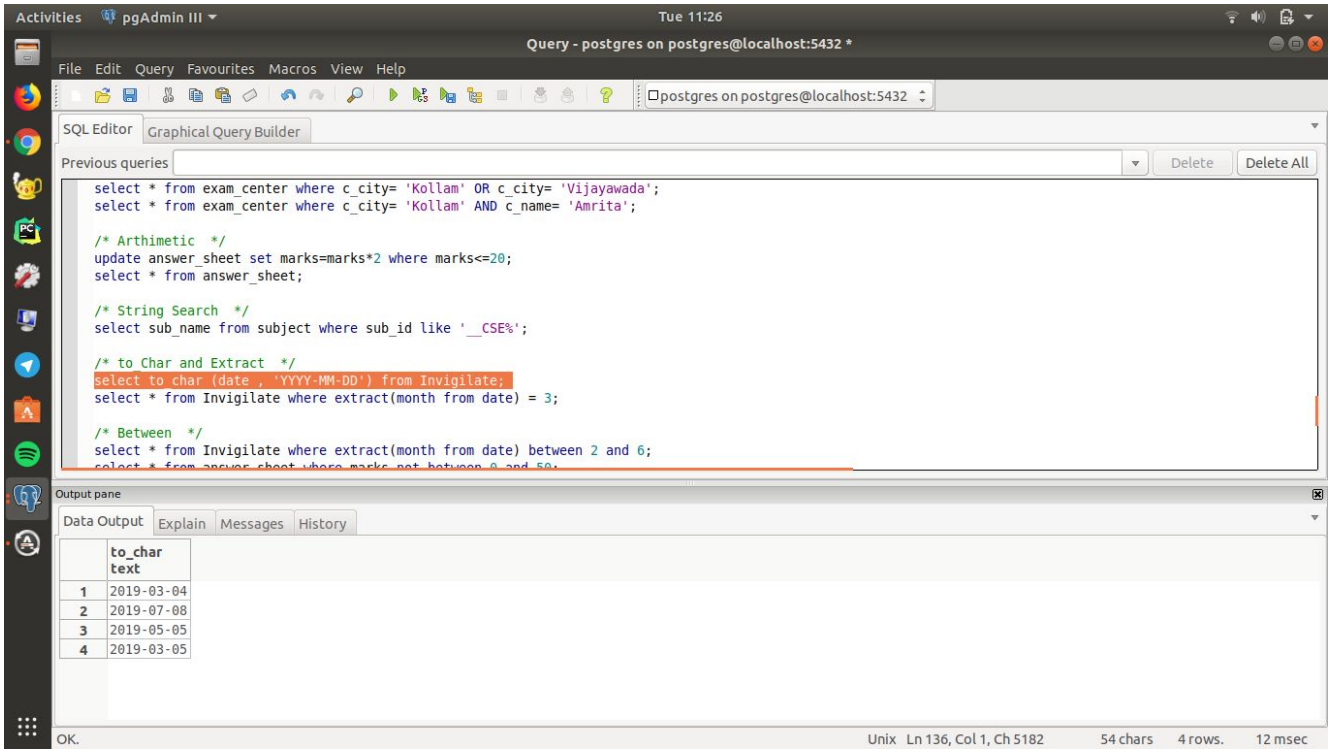
- Display all core CSE subjects

  select sub_name from subject where sub_id like '__CSE%';



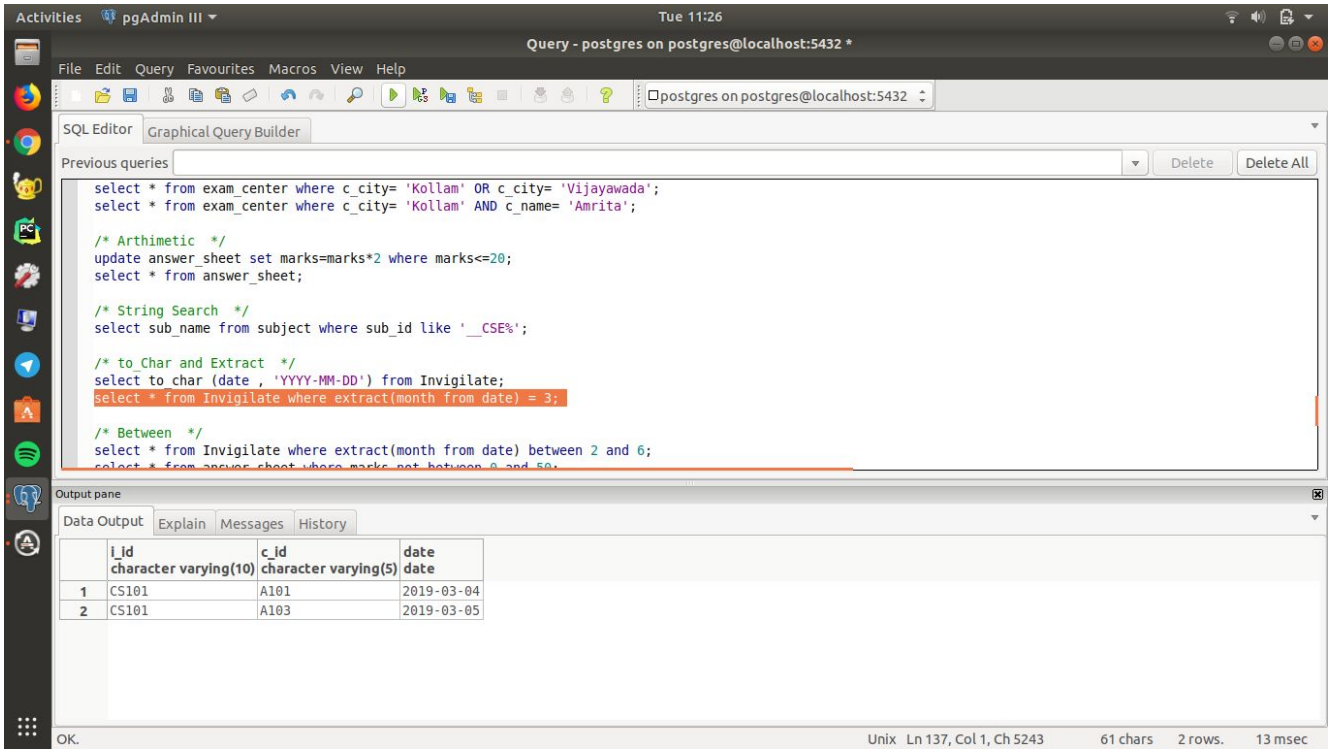
- Display the dates on which the instructor is given exam invigilation duty in the from of string

  select to_char (date , 'YYYY-MM-DD') from Invigilate;



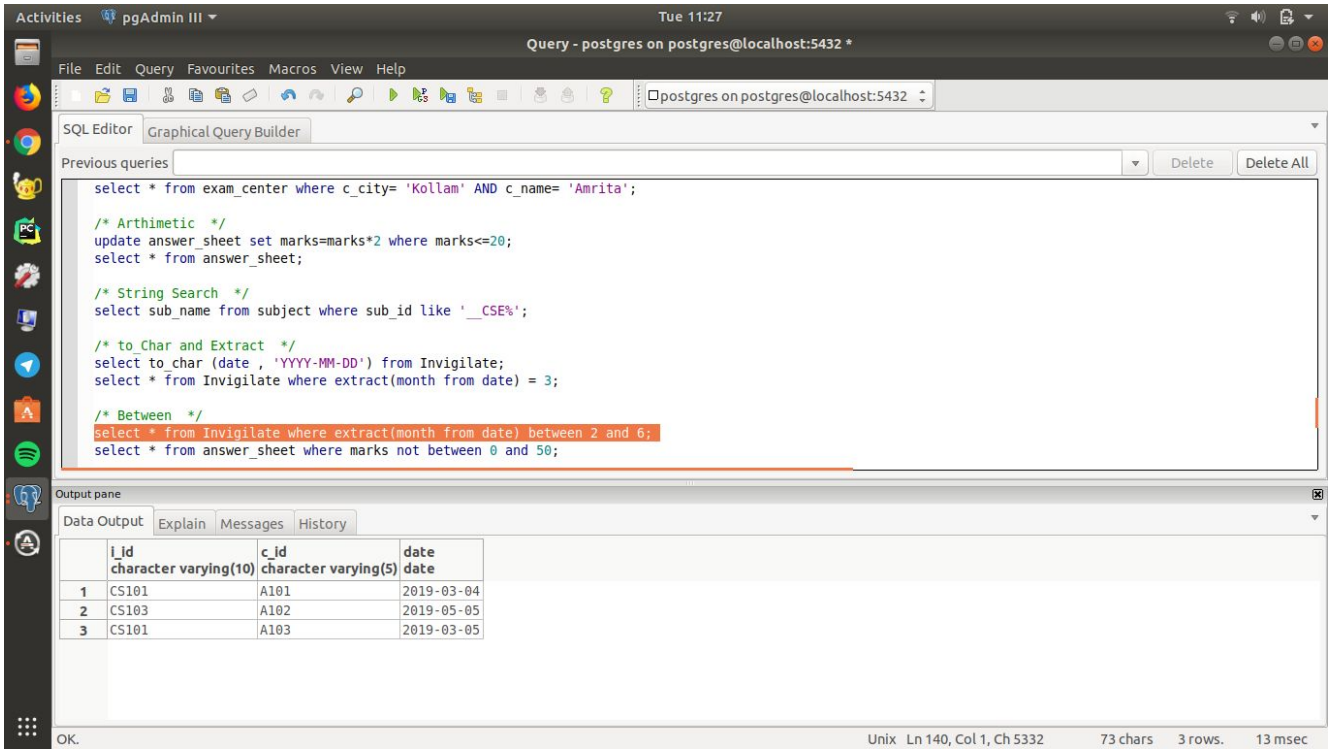
- Display the details of invigilation done in the month of march

select * from Invigilate where extract(month from date) = 3;



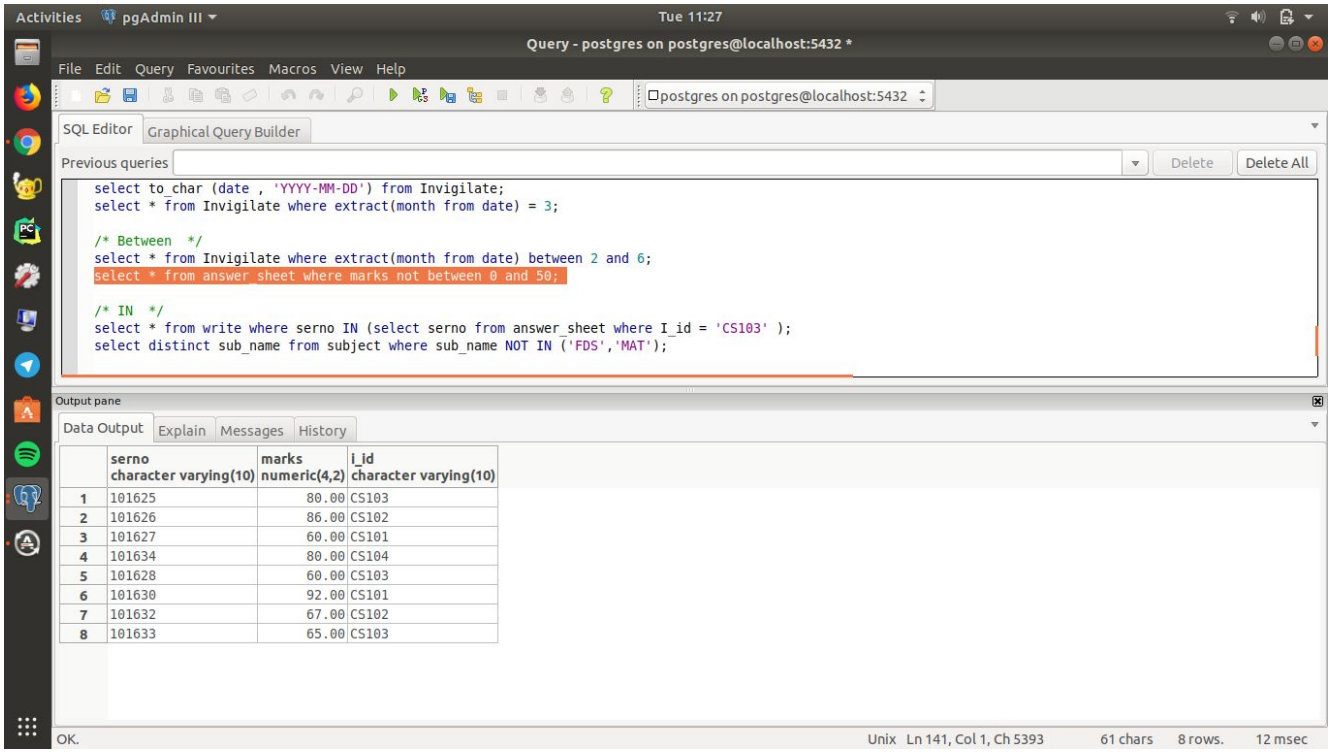
- Display invigilation details done in between february and June

select * from Invigilate where extract(month from date) between 2 and 6;



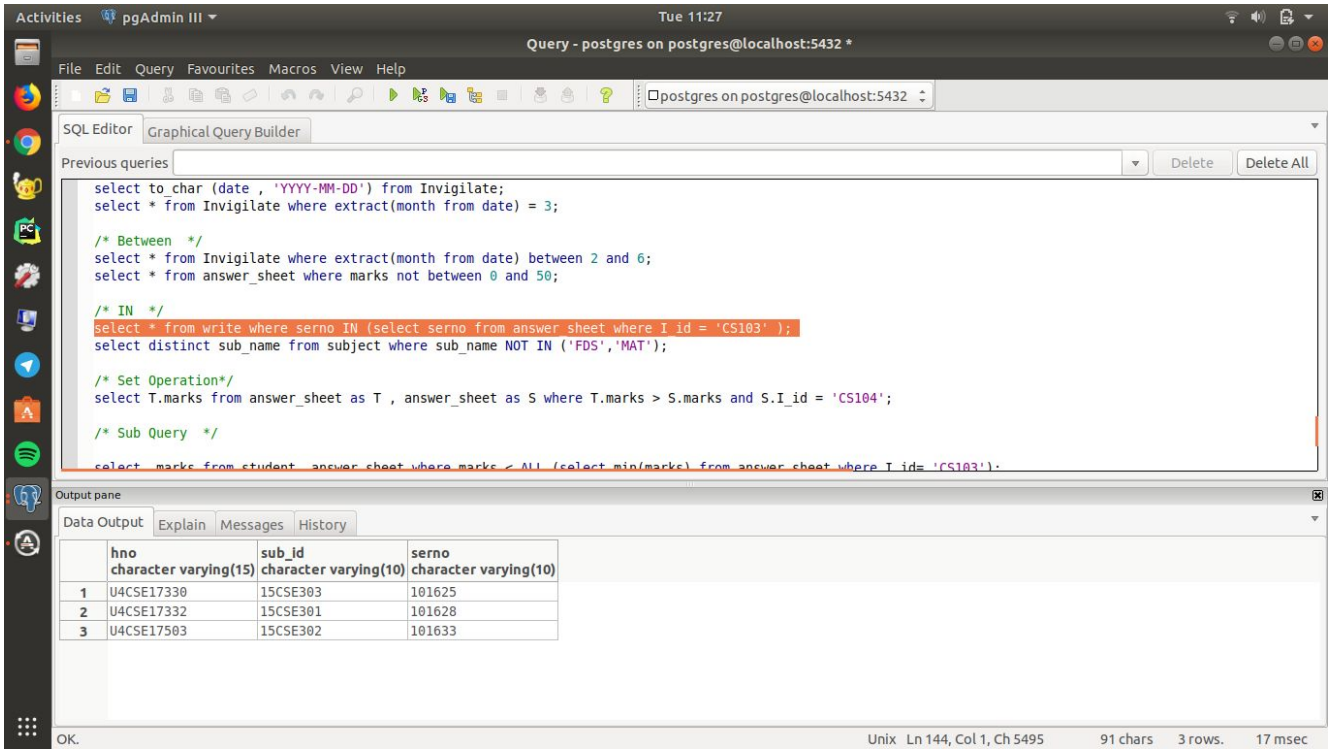
- Display details of answer sheets with marks greater than 50 and less than 0

select * from answer_sheet where marks not between 0 and 50;



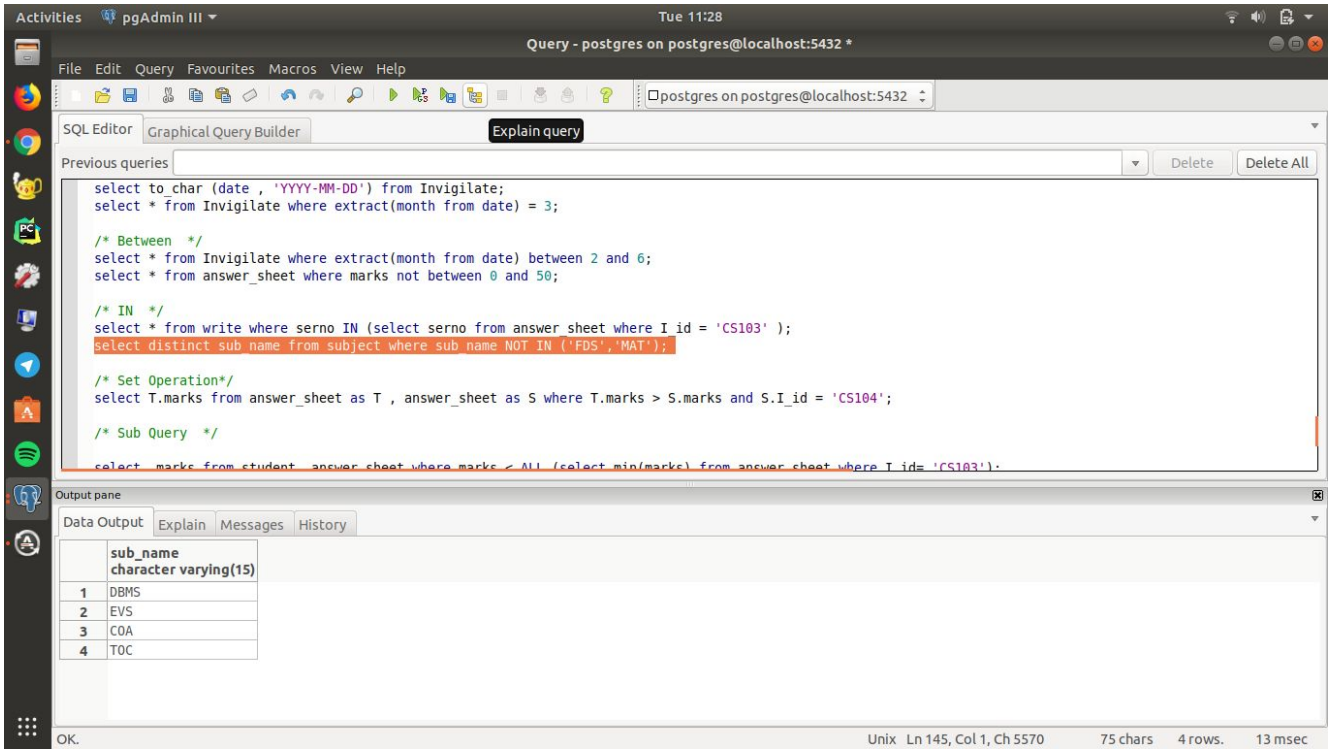
- Display the details of the answer sheet corrected by instructor having ID cs103

select * from write where serno IN (select serno from answer_sheet where I_id = 'CS103');



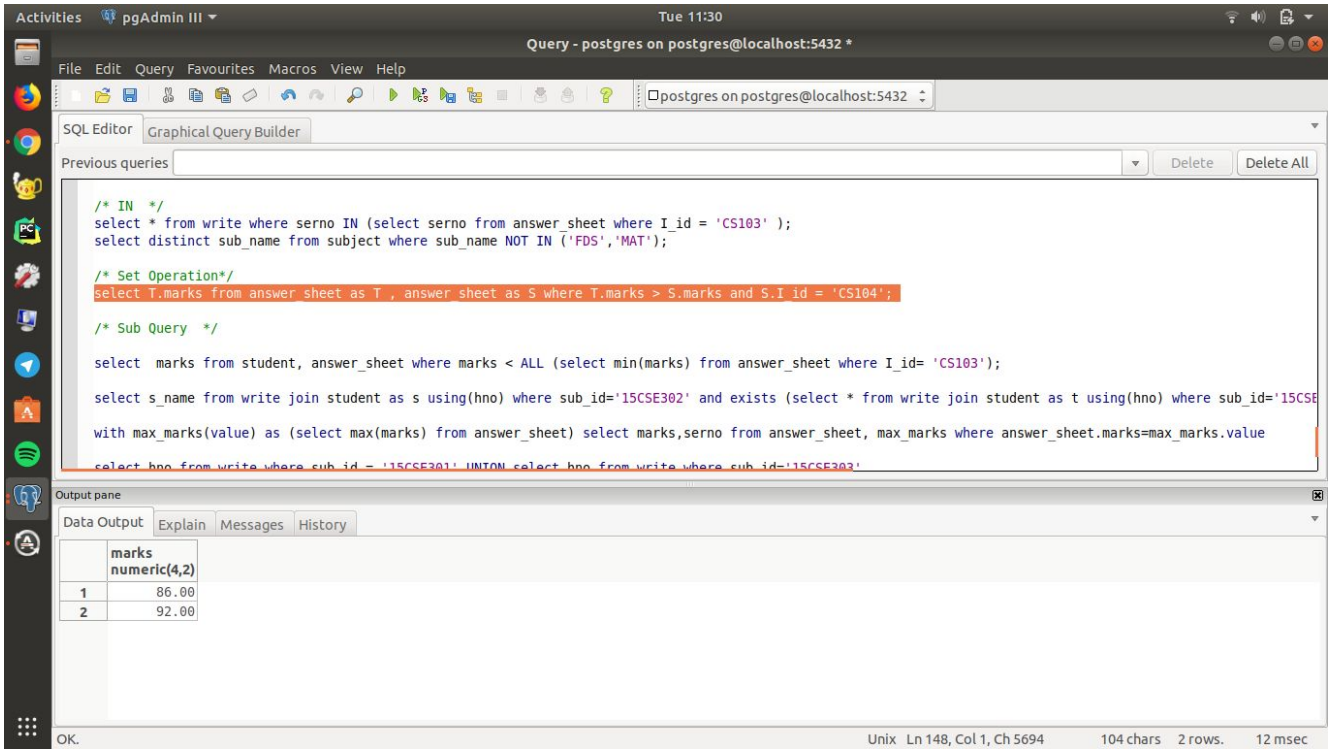
- Display all the subject names except fds and maths

select distinct sub_name from subject where sub_name NOT IN ('FDS','MAT');

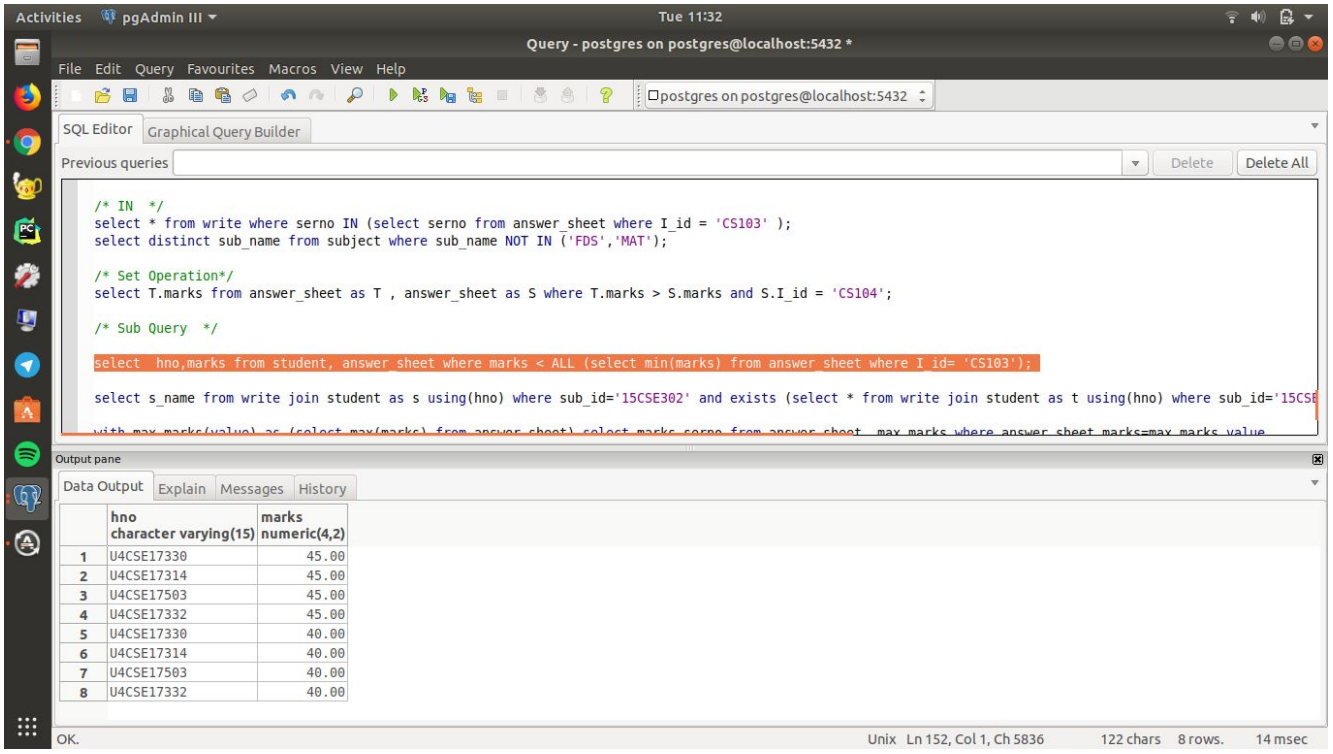


- Display marks which are greater than all the other marks corrected by instructor having ID CS104

select T.marks from answer_sheet as T , answer_sheet as S where T.marks > S.marks and S.I_id = 'CS104';

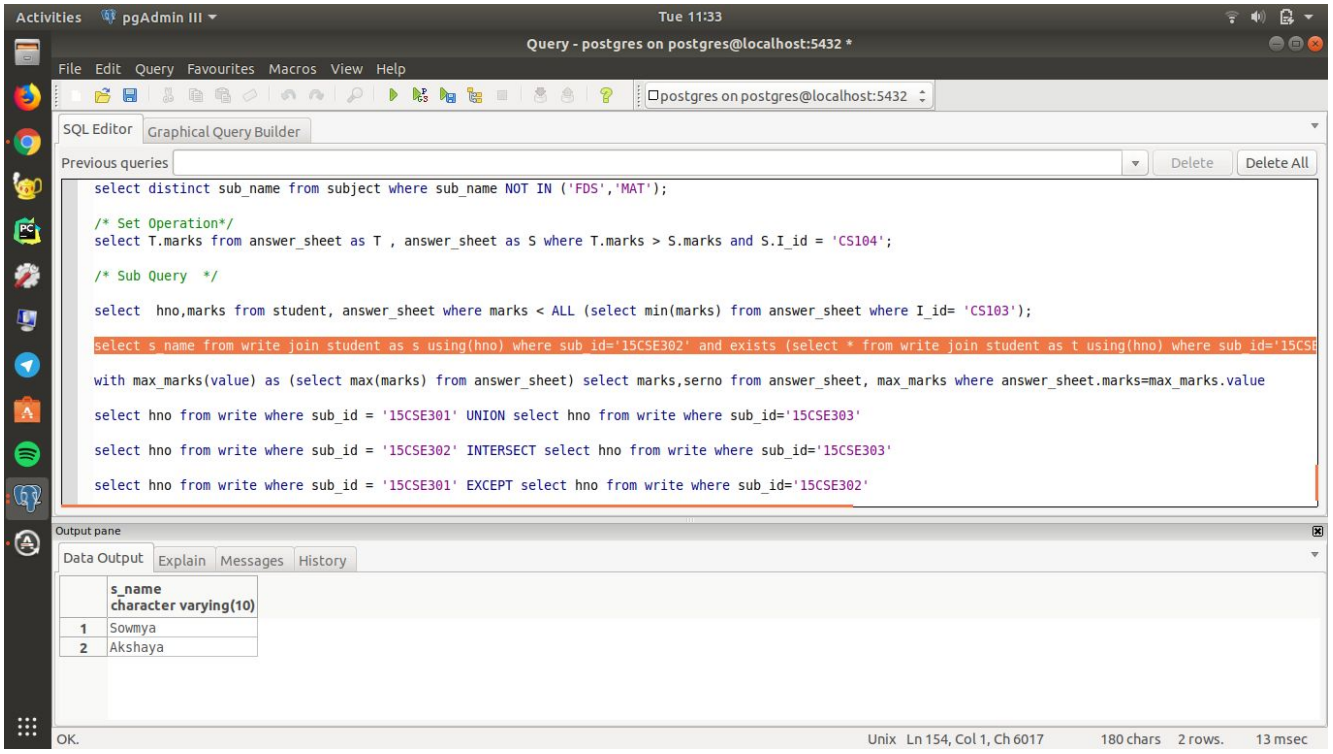


- Display all the marks along with their hall ticket numbers which are less than minimum marks of the answer sheets corrected by CS103
  select marks from student, answer_sheet where marks < ALL (select min(marks) from answer_sheet where I_id= 'CS103');



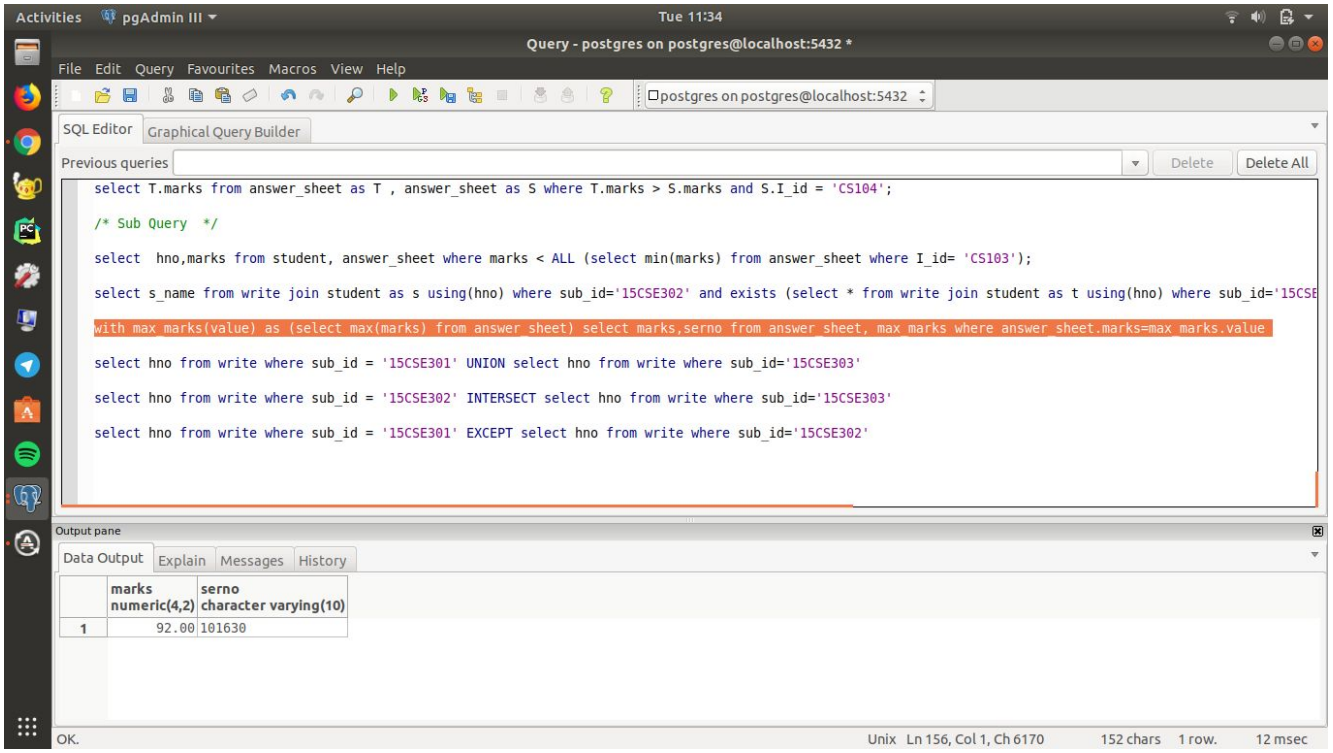
- Display all the student names who have written both dbms and toc exams

select s_name from write join student as s using(hno) where sub_id='15CSE302' and exists (select * from write join student as t using(hno) where sub_id='15CSE303'and s.hno=t.hno)



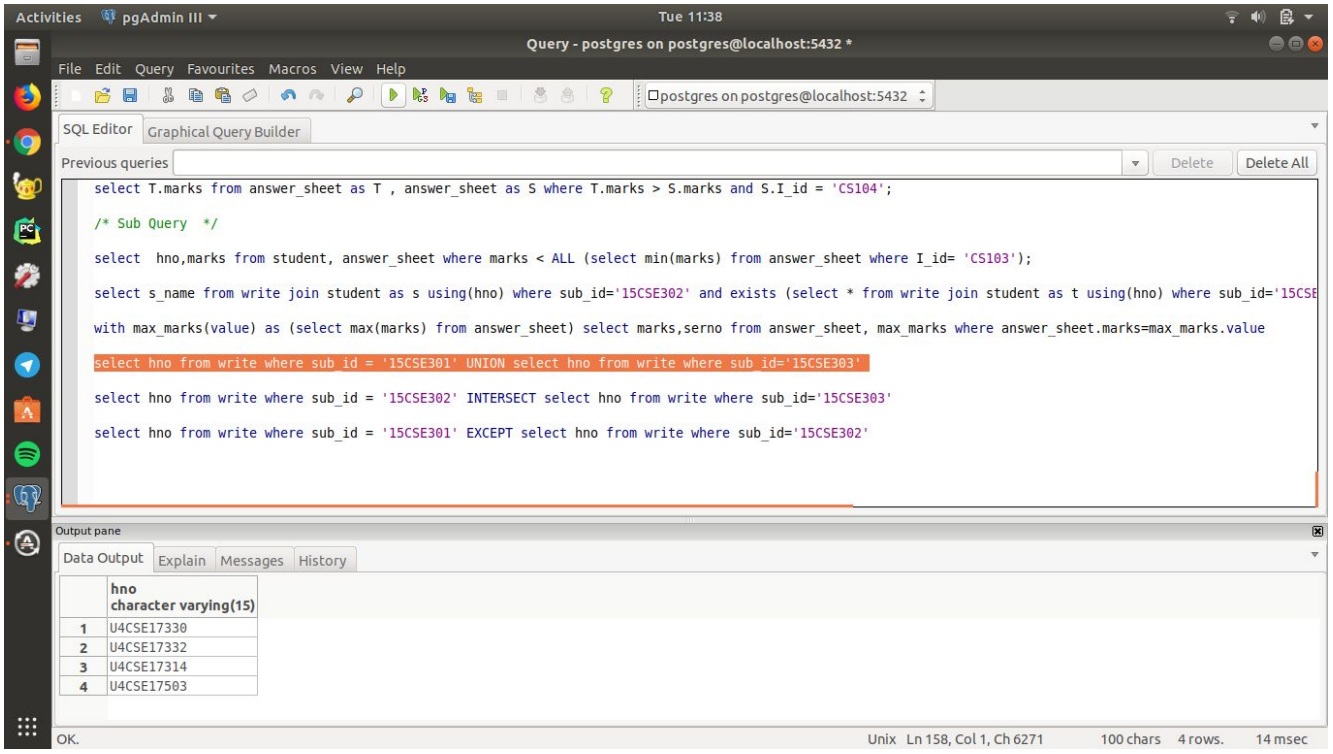
- Display serno of the answer sheet which obtained the maximum mark among all the evaluated papers

with max_marks(value) as (select max(marks) from answer_sheet) select marks,serno from answer_sheet, max_marks where answer_sheet.marks=max_marks.value



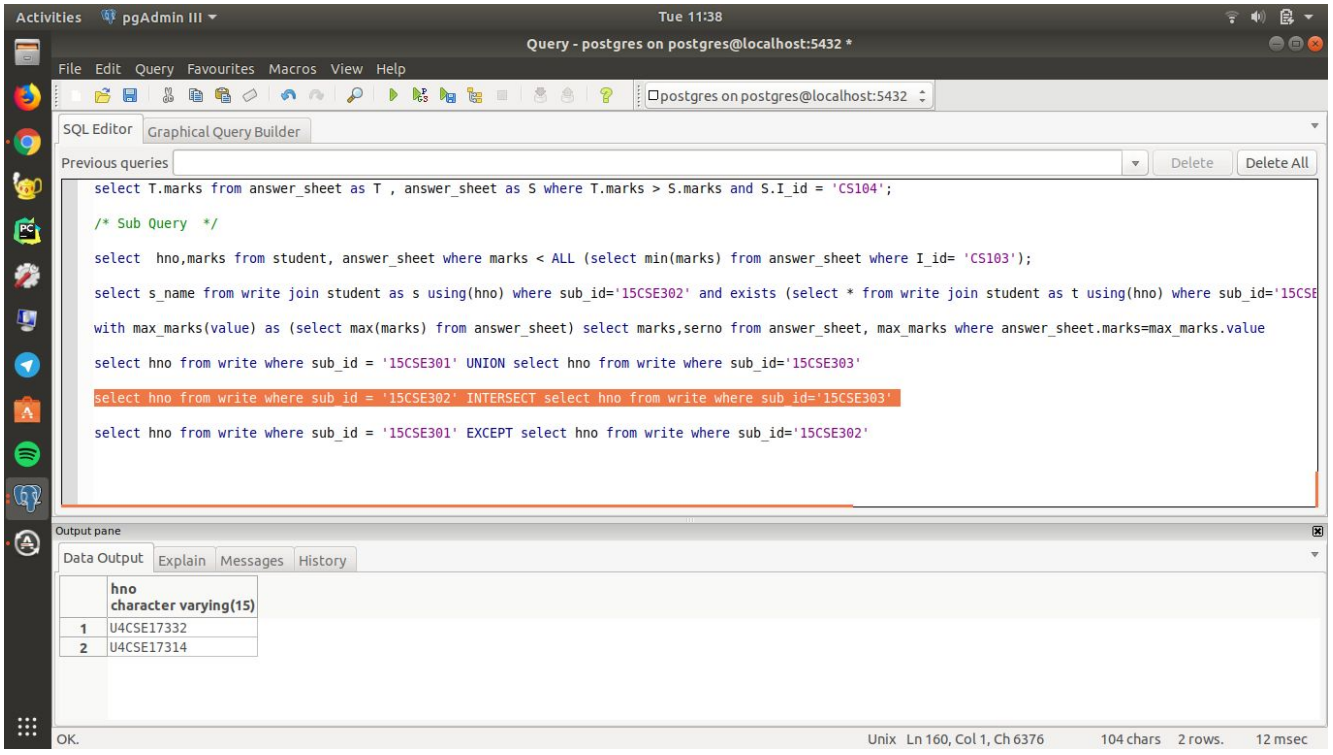
- Display all the students hall ticket numbers who have written either COA exam or Toc exam

select hno from write where sub_id = '15CSE301' UNION select hno from write where sub_id='15CSE303'



- Display all the students hall ticket numbers who have written both dbms exam and Toc exam

select hno from write where sub_id = '15CSE302' INTERSECT select hno from write where sub_id='15CSE303'



- Display all the student hall ticket numbers who have written COA exam but not Dbms exam

select hno from write where sub_id = '15CSE301' EXCEPT select hno from write where sub_id='15CSE302'