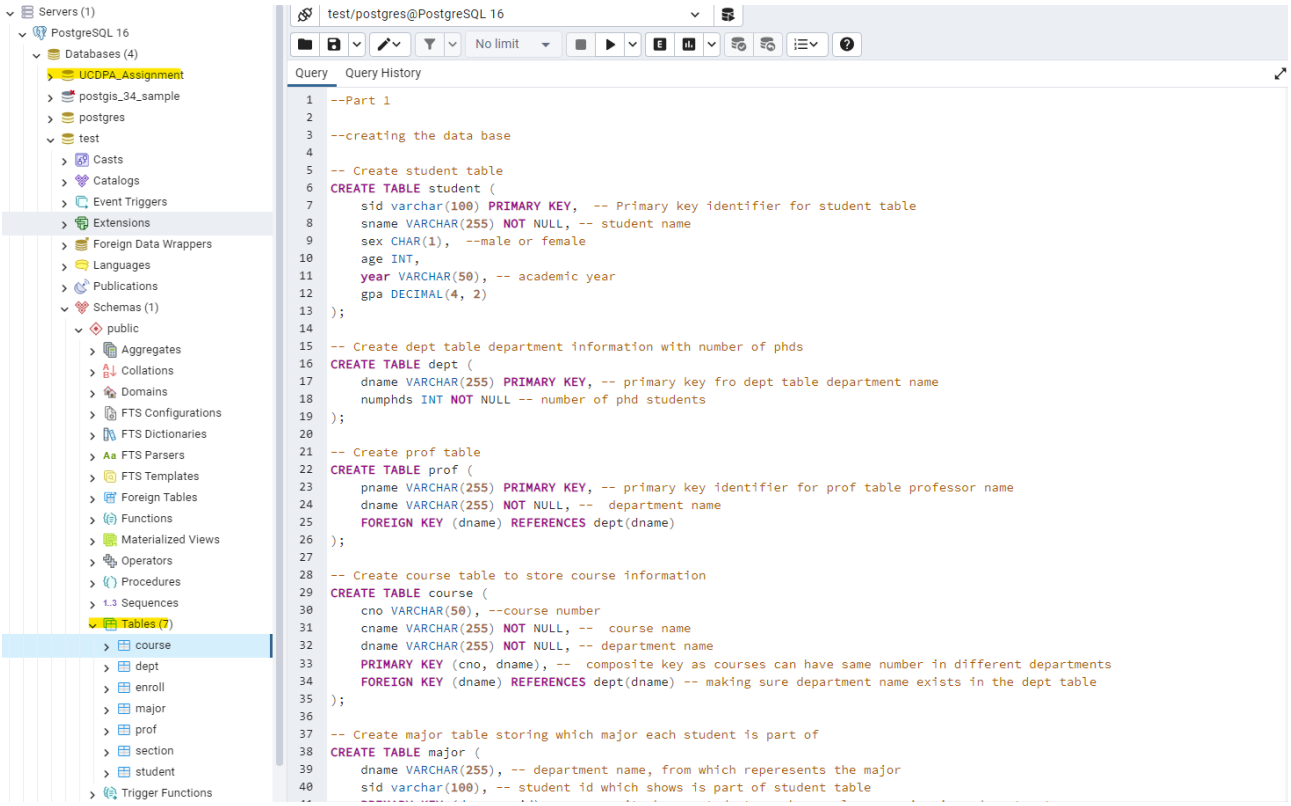
Part 1 Data Base setup

Screenshots

1)Created the database



2) Verifying if the data is being copied with example data table

```
--copying data to course table without headers
COPY course(cno,cname,dname)
FROM 'C:\Users\Rajesh PC\Desktop\UCDPA_Assignment\data\course.csv'
WITH (FORMAT csv, HEADER false, DELIMITER ',');
```

Query Query History

```
1  select * from course;
```

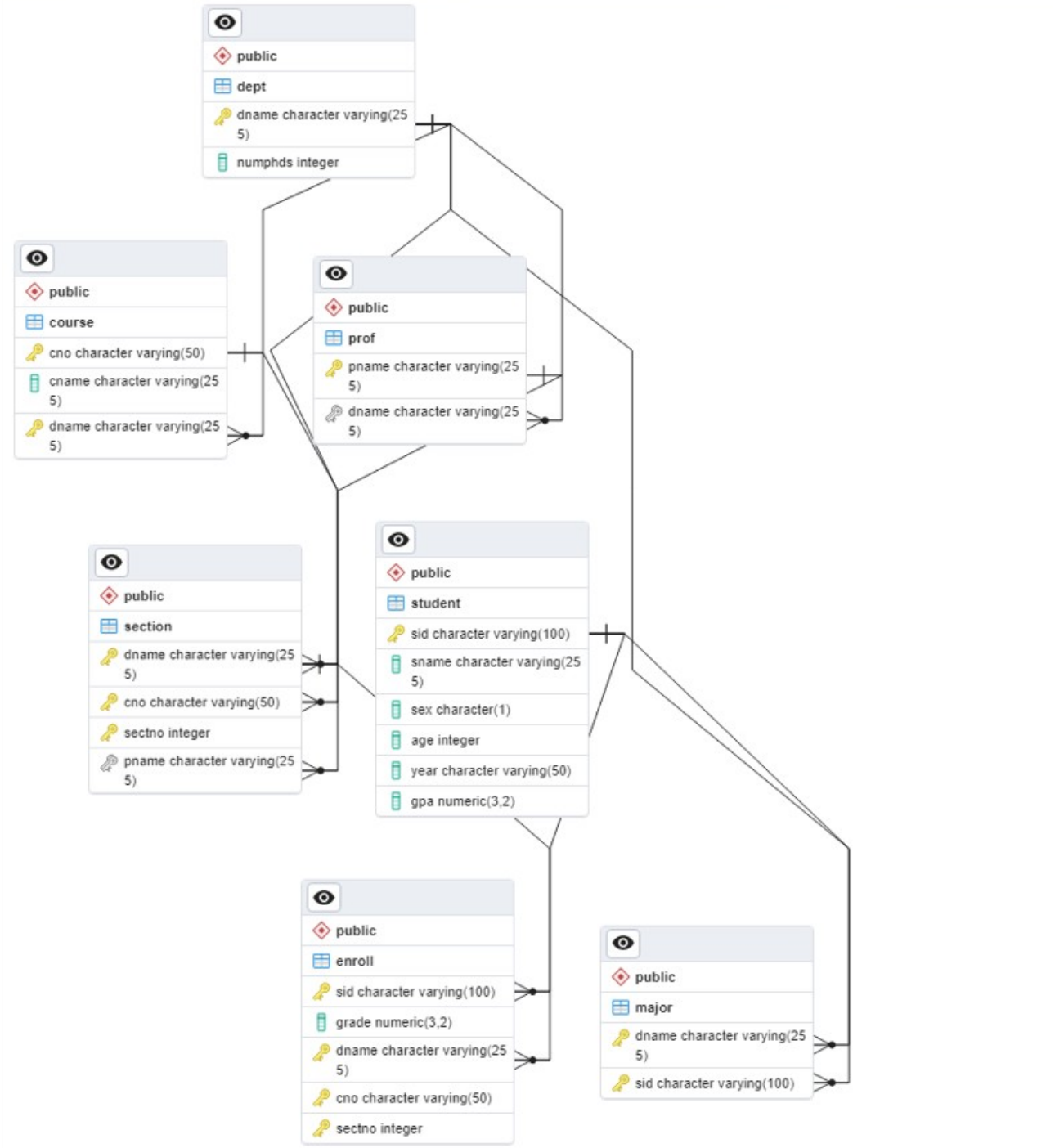
Data Output Messages Notifications

	cno [PK] character varying (50)	cname character varying (255)	dname [PK] character varying (255)
1	302	Intro to Programming	Computer Sciences
2	310	Thermodynamics	Chemical Engineering
3	315	Intro to Garbage	Sanitary Engineering
4	365	City Planning	Civil Engineering
5	375	Highway Engineering	Civil Engineering
6	461	College Geometry 1	Mathematics
7	462	College Geometry 2	Mathematics
8	467	Intro to Data Structures	Computer Sciences
9	514	Manpower Utilization	Industrial Engineering
10	561	Advanced City Planning	Civil Engineering
11	568	Advanced Garbage Collection	Sanitary Engineering
12	701	Compiler Construction	Computer Sciences
13	726	Nonlinear Programming	Computer Sciences

3) Relationship between the tables with the attributes

In order to model the academic organization, a number of links between attributes are constructed using main and foreign keys. Each student is individually identified by the sid property of the student table, which is connected to the enroll and major tables, which show the students' course enrollments and declared majors, respectively. The dname of the dept (department) table links to the major, prof, and course tables, providing a unique identity for academic departments and indicating which department offers which major, who teaches there, and what courses are offered. The combination of the course number (cno) and the dname in the course table refers to particular sections inside the section table, denoting various courses or programs. Students can access the exact class sections they are enrolled in by connecting the section table to the enroll table.

4) Entity Relationship diagram extracted from Postgresql



Section 2 Screenshots of the queries with the outputs in the order of the questions

1)

```
--1 Print the names of professors who work in departments that have fewer than 50 PhD students right

--here we need to use profeesor table and department table and using join on the department name attribute
-- to check the condition where number phds is less than 50

select prof.pname
from prof
join dept on prof.dname = dept.dname
where dept.numphds < 50;
```

Data Output | Messages | Notifications

	pname [PK] character varying (255)
1	Jones J.
2	Smith S.
3	Brian C.
4	Edison L.
5	Bucket T.
6	Walter A.

2)

```
--2 Print the names of the students with the lowest GPA right

-- here we need to use nested queries to find the lowest gpa among the students in the strudent table

select sname
from student
where gpa = (select min(gpa) from Student);

--3 For each Computer Sciences class, print the class number, section number, and the average GPA of students enro
```

Data Output | Messages | Notifications

	sname character varying (255)
1	Jetplane Leaving O.

3)

```
--3 For each Computer Sciences class, print the class number, section number, and the average GPA of students enrolled in the class section right

-- here we use normal join to join the course, section,enroll and student table with there primary keys.
--In order to calculate the average GPA for each section of Computer Sciences courses, they are used to match each part to its enrolled students.
--By concentrating on certain department courses and their student performances.

select course.cno AS class_number, section.sectno AS section_number, AVG(student.gpa) AS average_gpa
from course
join section on course.cno = section.cno and course.dname = section.dname
join enroll on section.dname = enroll.dname and section.cno = enroll.cno AND section.sectno = enroll.sectno
join student on enroll.sid = student.sid
where course.dname = 'Computer Sciences'
group by course.cno, section.sectno;
```

Data Output | Messages | Notifications

	class_number character varying (50)	section_number integer	average_gpa numeric
1	726	1	3.0000000000000000
2	302	1	2.9400000000000000
3	302	2	3.4142857142857143
4	467	1	3.2600000000000000
5	701	1	3.0500000000000000

4)

```
-- 4. Print the names and section numbers of all sections with more than six students enrolled in them

--The class sections (by department name and section number) that have more than six enrolled students are listed in this SQL query.
--In order to enable the COUNT operation to ascertain the number of students in each part,
--it employs a JOIN to match enrollment records to their corresponding sections and a GROUP BY to arrange these records by section.

select section.dname , section.sectno
from section
join enroll ON section.dname = enroll.dname and section.sectno = enroll.sectno
GROUP BY section.dname, section.sectno
having COUNT(enroll.sid) > 6;
```

Data Output | Messages | Notifications

	dname character varying (255)	sectno integer
1	Civil Engineering	1
2	Computer Sciences	1
3	Computer Sciences	2
4	Industrial Engineering	1
5	Mathematics	1

5)

```sql
-- 5. List the department names and the number of classes offered in each department.

--  It does this by joining the dept and course tables based on the department name,
--then seperates the results by department name to ensure each department is listed once,
--and counts usinf count() the number of courses associated with each department.

Select dept.dname , COUNT(course.cno)
From dept
Join course ON dept.dname = course.dname
GROUP BY dept.dname;
```

Data Output | Messages | Notifications

	dname [PK] character varying (255)	count bigint
1	Civil Engineering	3
2	Chemical Engineering	1
3	Sanitary Engineering	2
4	Computer Sciences	4
5	Mathematics	2
6	Industrial Engineering	1

6)

```
-- 6. Print the names of departments that have one or more majors who are under 18 years old

-- the query is executed by joimning dept major and student to filre ou the age constraint and gives department names
select dept.dname
from dept
join major on dept.dname = major.dname
join student on student.sid = major.sid
where student.age < 18;

--7 Print the names and majors of students who are taking one of the College Geometry courses
```

Data Output | Messages | Notifications

	dname [PK] character varying (255)
1	Industrial Engineering
2	Mathematics

7)

```
--7 Print the names and majors of students who are taking one of the College Geometry courses

-- The query use is to list student names and their major names
-- for those whose majors are linked with departments that has 'College Geometry 1' or 'College Geometry 2'.
Select student.sname , major.dname
from student
left join major on student.sid = major.sid
left join course on major.dname =  course.dname
where course.cname in('College Geometry 1', 'College Geometry 2')
group by student.sname, major.dname;
```

Data Output | Messages | Notifications

	sname character varying (255)	dname character varying (255)
1	Andermanthenol K.	Mathematics
2	Bates M.	Mathematics
3	Bates Michael L.	Mathematics
4	Birch M.	Mathematics
5	Davis Scott P.	Mathematics
6	Ghandi I.	Mathematics
7	Gonring J.	Mathematics
8	Gringo C.	Mathematics
9	Grzlbltz Q.	Mathematics
10	Jacobs T.	Mathematics
11	Jones J.	Mathematics
12	Kaisler Janet M.	Mathematics
13	Kirk J.	Mathematics
14	Longlastname A.	Mathematics
15	Mayer N.	Mathematics
16	Morgan D.	Mathematics

8)

```
-- 8 For those departments that have no major taking a College Geometry course print the department name and the number of PhD students in the department
--Subquery to check if the department offers either of the specific courses

Select dept.dname, dept.numphds
From dept
Where NOT EXISTS ( --not exist condition evaluates to false for that particular department, because the subquery found a matching row, thus excluding the depart
 select 1 --The use of select 1 in a subquery like this is simply to return a non-null value if any rows are found that meet the subquery's conditions
 from course
 where course.dname = dept.dname
 and course.cname IN ('College Geometry 1', 'College Geometry 2')
)
GROUP BY dept.dname, dept.numphds;
```

Data Output | Messages | Notifications

	dname [PK] character varying (255)	numphds integer
1	Chemical Engineering	32
2	Computer Sciences	47
3	Industrial Engineering	41
4	Civil Engineering	88
5	Sanitary Engineering	3

9)

```
--9 Print the names of students who are taking either a Computer Sciences course or a Mathematics course

--the left join joins the student table with the major table based on the student ID (sid) to associate each student with their declared major,
--then filters the results to include only those students whose major (dname) is in the specified categories: 'Mathematics' or 'Computer Sciences'.

select student.sname from student
left join major on student.sid = major.sid
where major.dname in ('Mathematics','Computer Sciences');

--10 Print the age difference between the oldest and the youngest Computer Sciences major right
```

Data Output | Messages | Notifications

	sname character varying (255)
1	Jacobs T.
2	Pierson E.
3	Zeene Ben N.
4	Sulfate Barry M.
5	Form Clara O.
6	Scott Kim J.
7	Sather Roberto B.
8	Stanley Leotha T.
9	Smith Joyce A.
10	Jones David S.
11	Paul Mary W.
12	Soong V.
13	Kellerman S.
14	Cheong R.
15	Borchart Sandra L.
16	Alsberg David J.

10)

```
--10 Print the age difference between the oldest and the youngest Computer Sciences major right

select max(student.age) - min(student.age) as agedifference
from student
join major on major.sid = student.sid  --Joins the student and major tables on the student ID
where major.dname = 'Computer Sciences'; -- Filters to include only Computer Sciences majors

--11. For each department that has one or more majors with a GPA under 1.0, print the name of the departm

select dept.dname , avg(student.gpa)
from dept
```

Data Output | Messages | Notifications

	agedifference integer
1	38

11)

```
--11. For each department that has one or more majors with a GPA under 1.0, print the name of the department and the average GPA of its majors

select dept.dname , avg(student.gpa)
from dept
join major on dept.dname = major.dname --Joins the department table with the major table to correlate departments with their students' majors
join student on major.sid = student.sid --Further joins with the student table to access the GPA of students in those majors
where student.gpa < 1.0 --Filters the data to only include students with a GPA below 1.0
group by dept.dname; --to show department wise
```

Data Output | Messages | Notifications

	dname [PK] character varying (255)	avg numeric
1	Civil Engineering	0.00000000000000000000
2	Computer Sciences	0.70000000000000000000
3	Industrial Engineering	0.35000000000000000000

12)

```
--12 Print the ids, names and GPAs of the students who are currently taking all the Civil Engineering courses
-- Retrieves the identifiers, names, and GPAs of students who are enrolled in every Civil Engineering course a
select student.sid, student.sname, student.gpa
from student
where NOT EXISTS (
-- Subquery to identify any Civil Engineering courses not currently enrolled by the student.
Select course.cno
From course
Where course.dname = 'Civil Engineering'  -- Focuses on courses within the Civil Engineering department.
and not EXISTS (
-- Subquery to check if the student is enrolled in each of the identified Civil Engineering courses.
Select enroll.sid
From enroll
Join section ON enroll.cno = section.cno AND enroll.sectno = section.sectno AND enroll.dname = section.dname
Where section.dname = 'Civil Engineering'  -- Ensures the enrollment is for a Civil Engineering course.
AND enroll.sid = student.sid  -- Matches the enrollment to the specific student.
)
)
GROUP BY student.sid;  -- Groups results by student to ensure unique entries for each.

-------------------------------------------------------End-------------------------------------------------------
```

Data Output | Messages | Notifications

	sid [PK] character varying (100)	sname character varying (255)	gpa numeric (4,2)
1	19	Smith L.	0.70
2	27	Anderson P.	3.20
3	3	Zeene Ben N.	3.90
4	33	Chao Tsechih	3.60
5	34	Kasten Norman L.	2.50
6	35	Mathews John W.	3.60
7	38	Auen B.	2.70
8	39	Shoemaker A.	3.50
9	41	Fisher C.	3.50
10	43	Ksar J.	3.40
11	54	Maximillian	3.00
12	55	Glitch R.	2.80
13	56	Starry J.	3.30