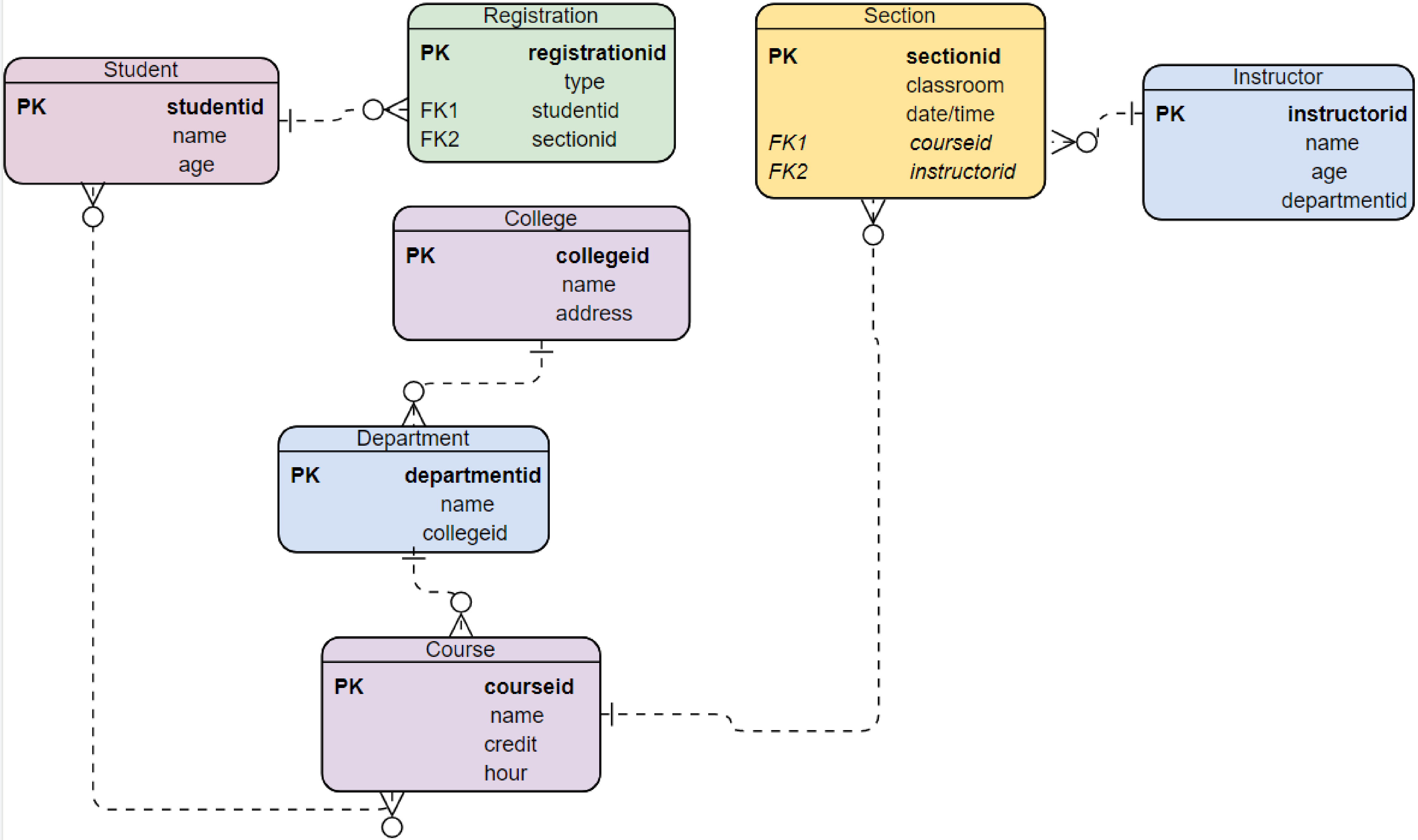
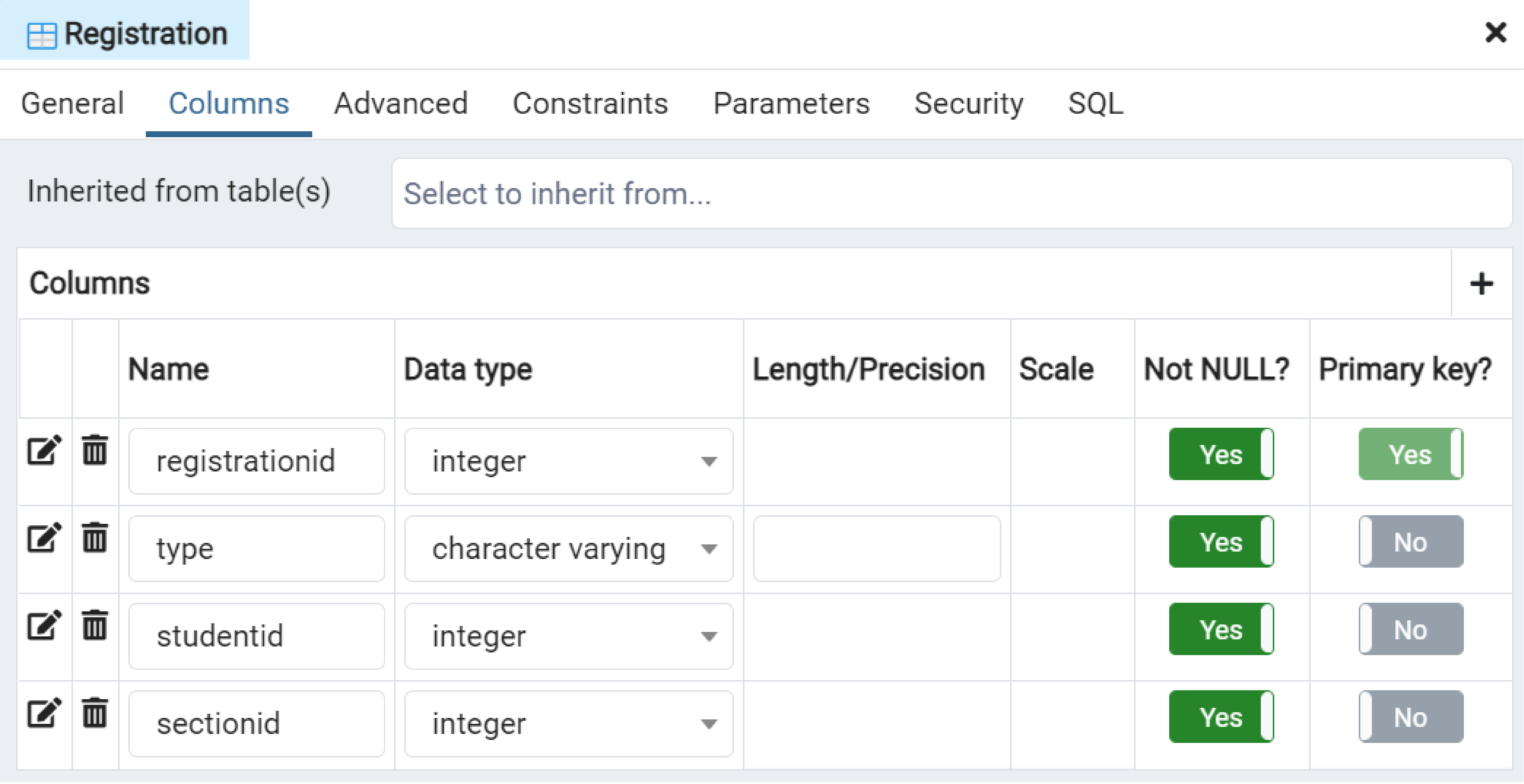
course registration database.

Shansi Dong

A20466369

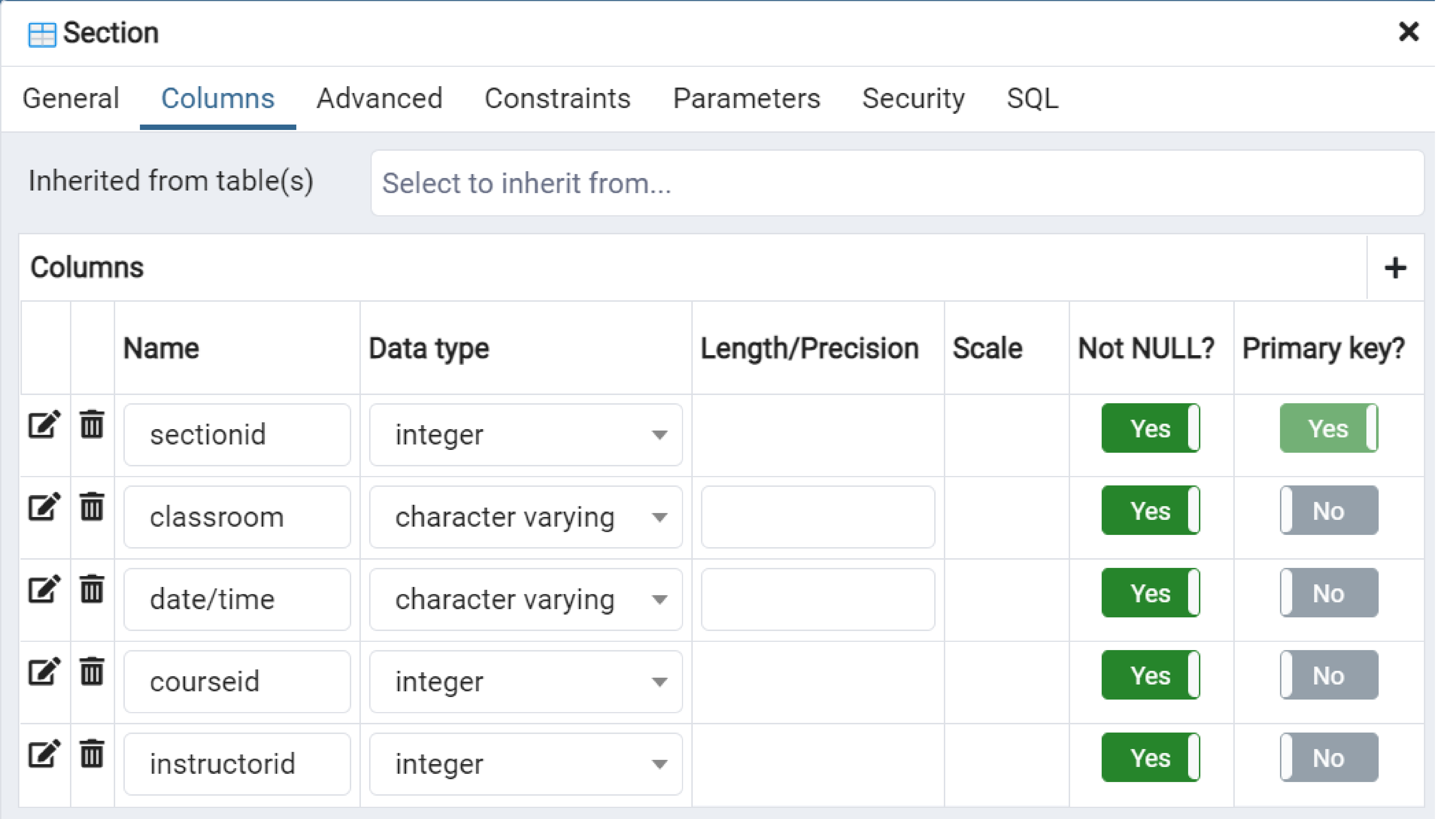

Student
PK studentid
name
age
Registration
PK registrationid
type
FK1 studentid
FK2 sectionid
Section
PK sectionid
classroom
date/time
FK1 courseid
FK2 instructorid
Instructor
PK instructorid
name
age
departmentid
College
PK collegeid
name
address
Department
PK departmentid
name
collegeid
Course
PK courseid
name
credit
hour



Inherited from table(s): Select to inherit from...

Columns +

		Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
✎	🗑	registrationid	integer			Yes	Yes
✎	🗑	type	character varying			Yes	No
✎	🗑	studentid	integer			Yes	No
✎	🗑	sectionid	integer			Yes	No

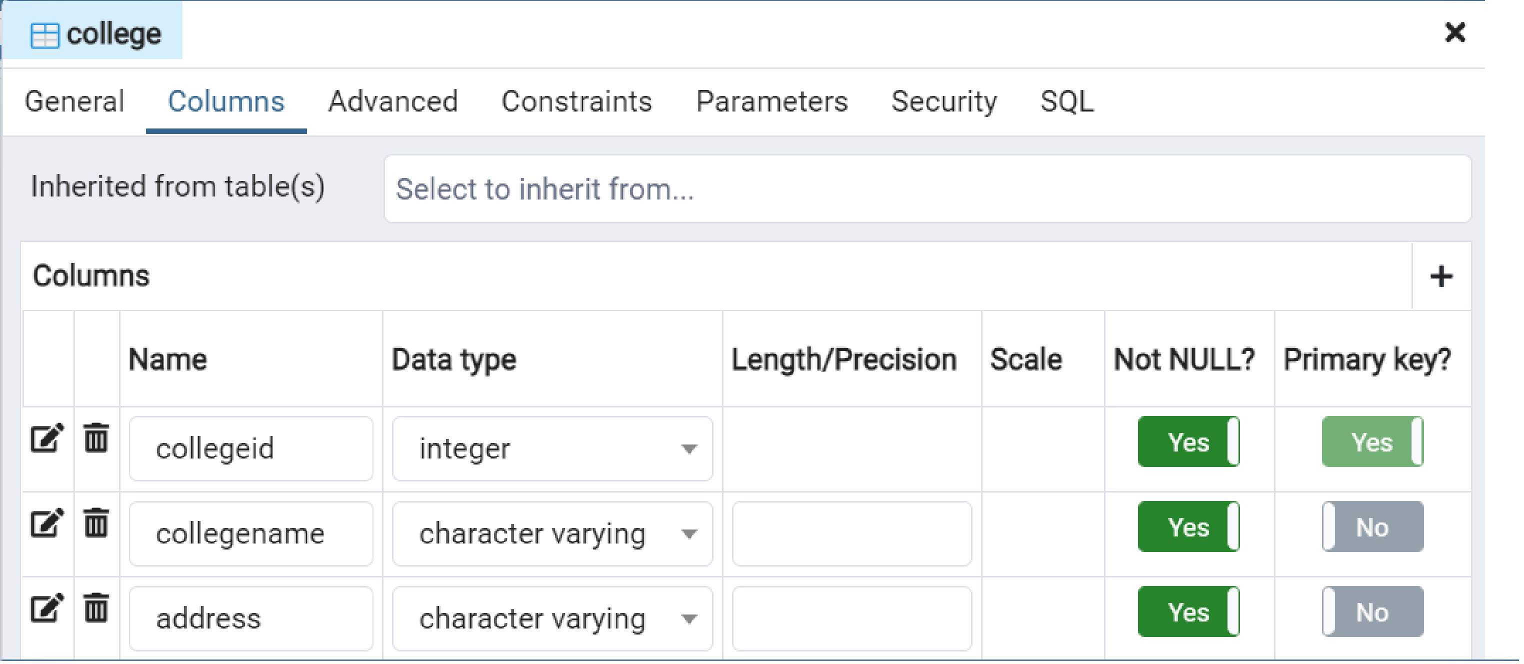


Section

Inherited from table(s): Select to inherit from...

Columns +

		Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
✎	🗑	sectionid	integer			Yes	Yes
✎	🗑	classroom	character varying			Yes	No
✎	🗑	date/time	character varying			Yes	No
✎	🗑	courseid	integer			Yes	No
✎	🗑	instructorid	integer			Yes	No

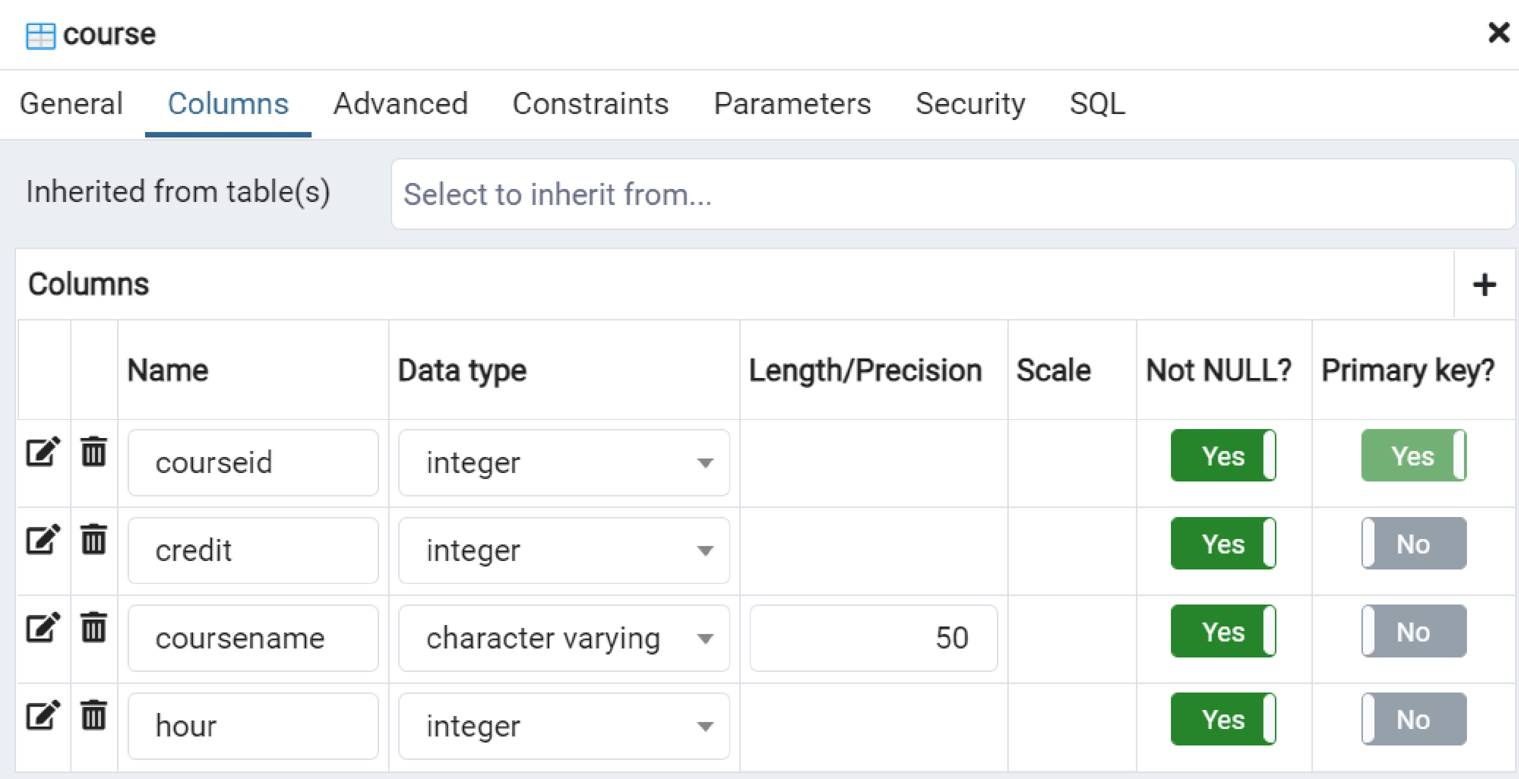


college

Inherited from table(s): Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
collegeid	integer			Yes	Yes
collegename	character varying			Yes	No
address	character varying			Yes	No

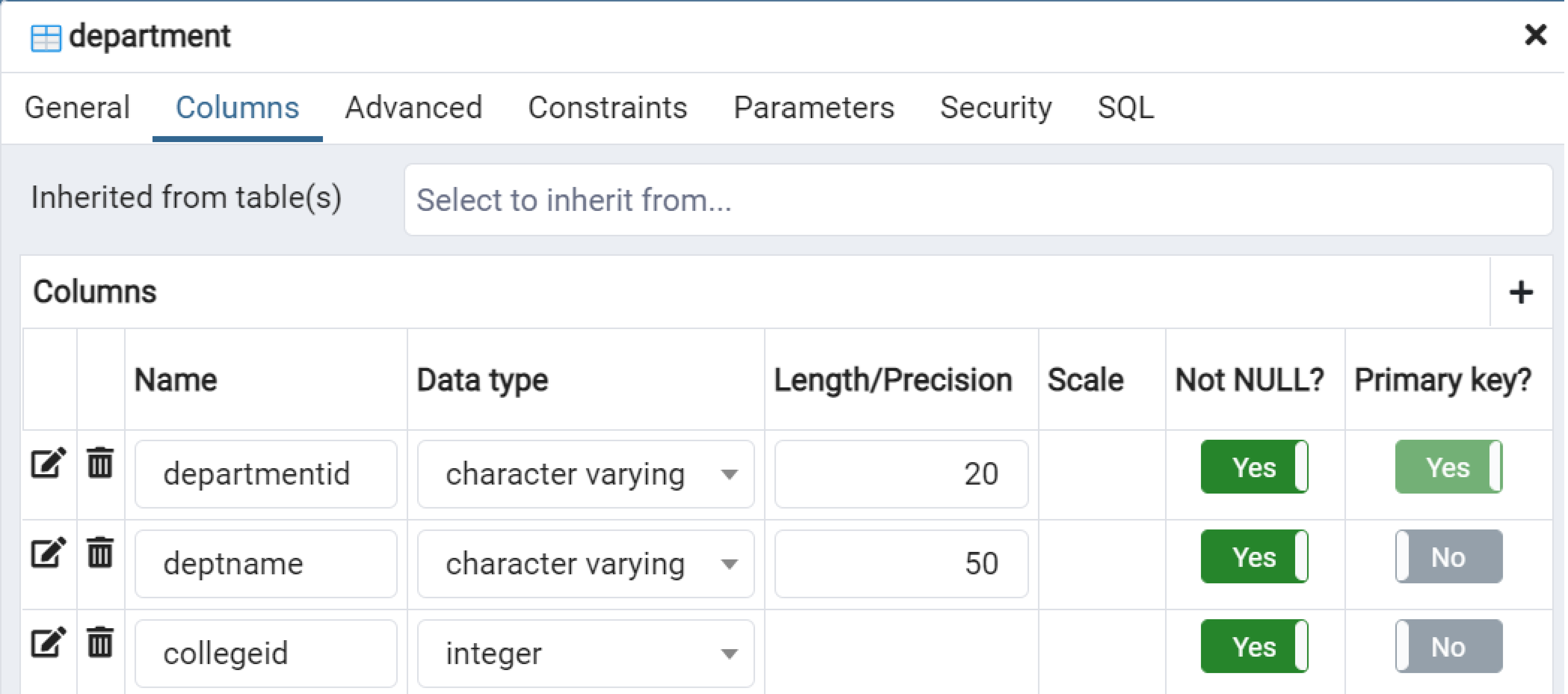


course

Inherited from table(s): Select to inherit from...

Columns

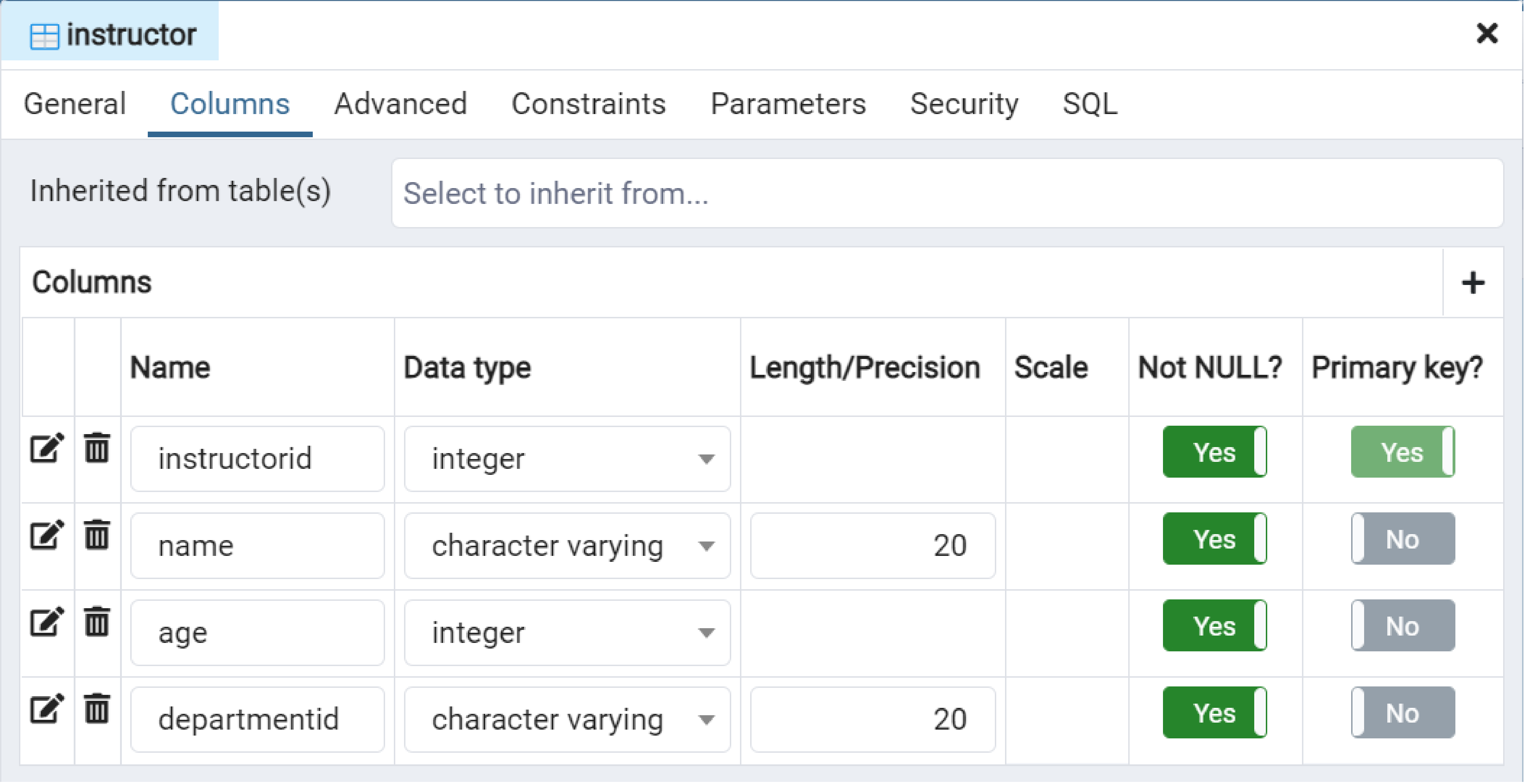
Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
courseid	integer			Yes	Yes
credit	integer			Yes	No
coursename	character varying	50		Yes	No
hour	integer			Yes	No



Inherited from table(s): Select to inherit from...

Columns

		Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
		departmentid	character varying	20		Yes	Yes
		deptname	character varying	50		Yes	No
		collegeid	integer			Yes	No



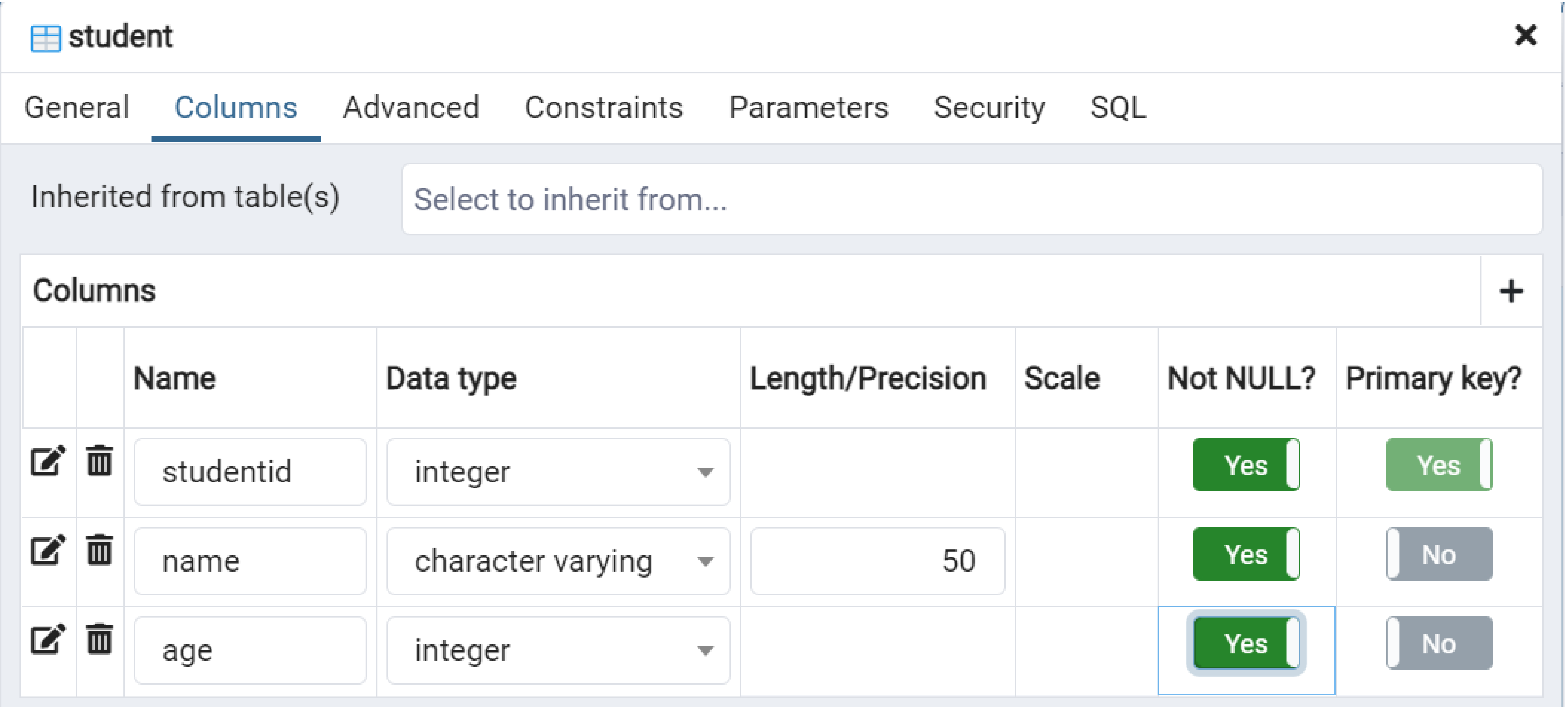
instructor

General | Columns | Advanced | Constraints | Parameters | Security | SQL

Inherited from table(s): Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
instructorid	integer			Yes	Yes
name	character varying	20		Yes	No
age	integer			Yes	No
departmentid	character varying	20		Yes	No



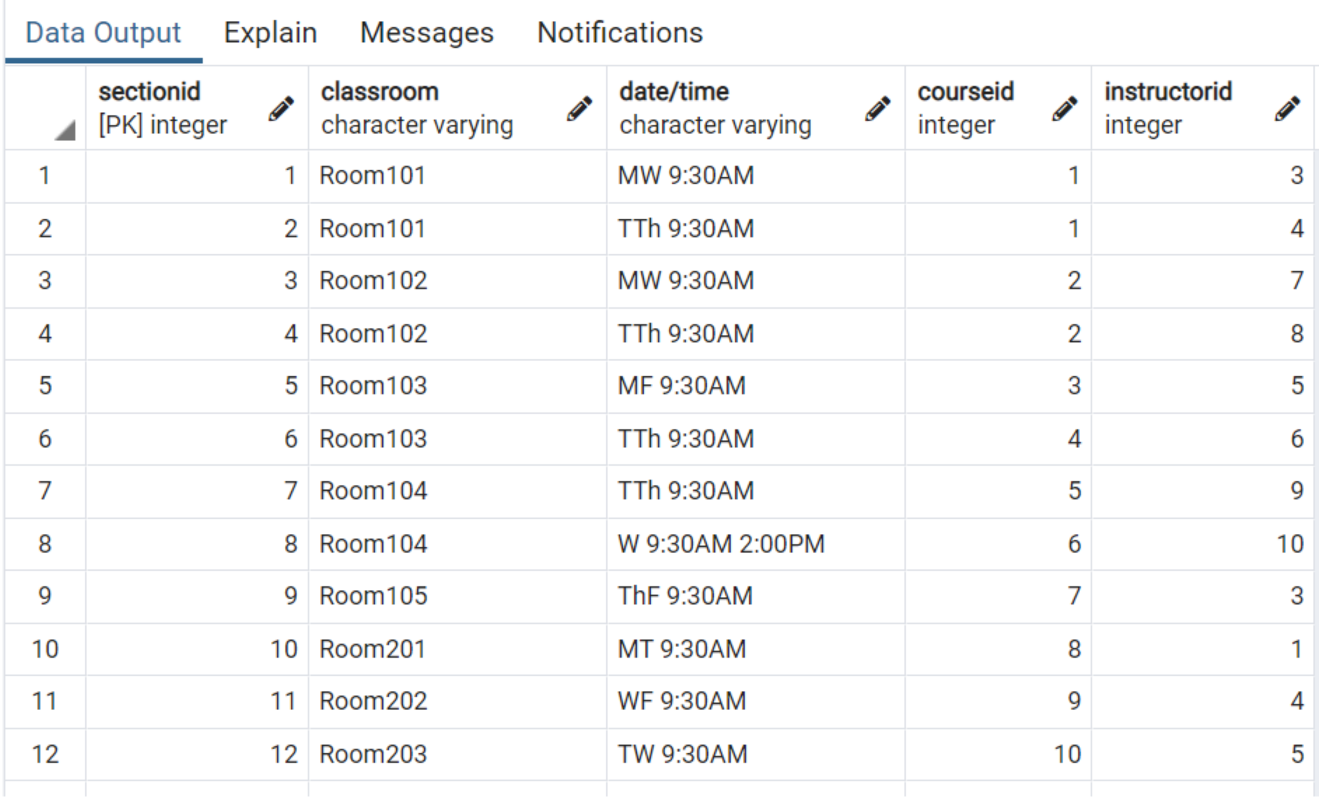
Inherited from table(s): Select to inherit from...

Columns

		Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
		studentid	integer			Yes	Yes
		name	character varying	50		Yes	No
		age	integer			Yes	No

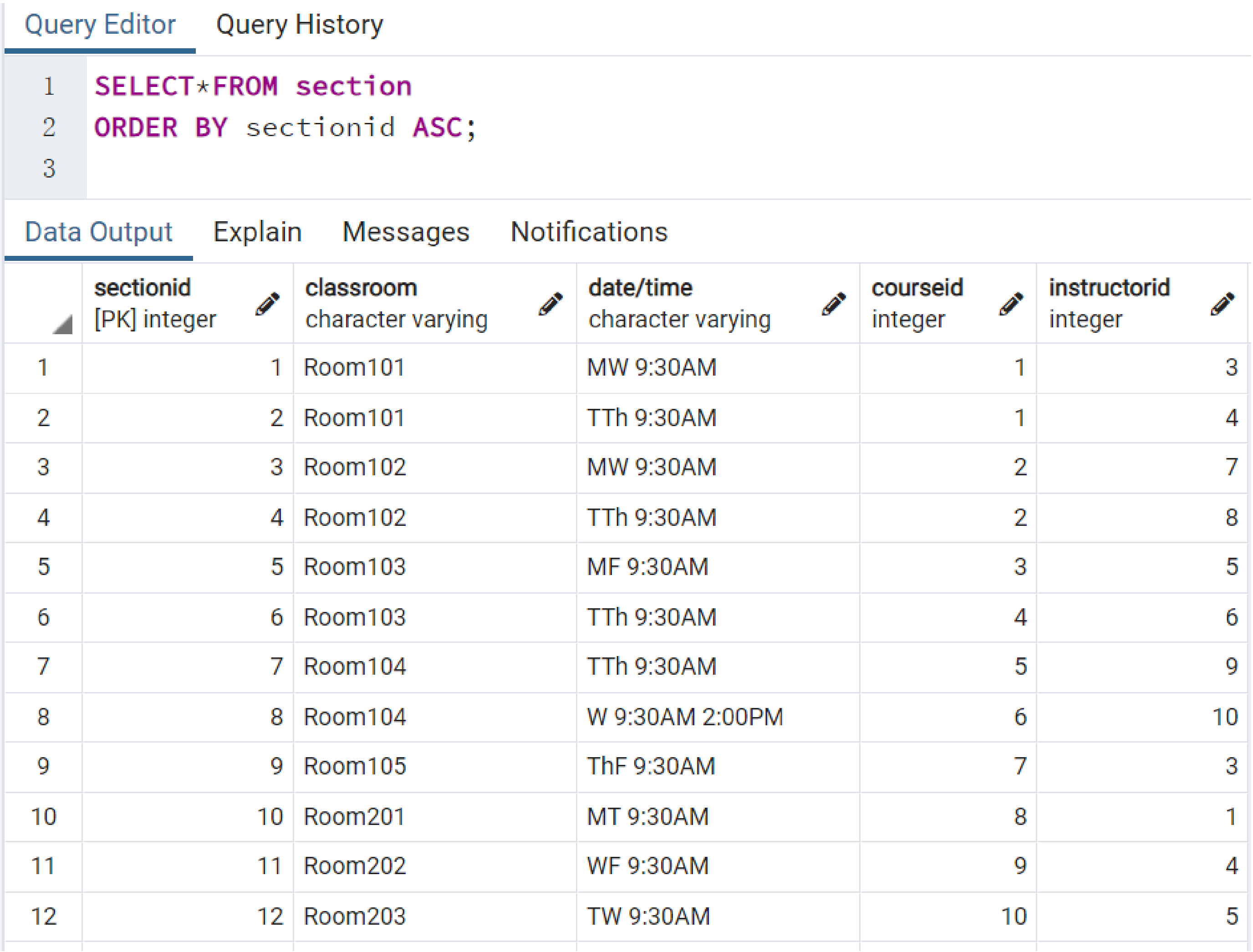
Insert value in table “section’’:

- INSERT INTO public."Section"(
- sectionid, classroom, "date/time", courseid, instructorid)
- VALUES (1, 'Room101','MW 9:30AM',1,3),
- (2, 'Room101','TTh 9:30AM',1,4),
- (3, 'Room102','MW 9:30AM',2,7),
- (4, 'Room102','TTh 9:30AM',2,8),
- (5, 'Room103','MF 9:30AM',3,5),
- (6, 'Room103','TTh 9:30AM',4,6),
- (7, 'Room104','TTh 9:30AM',5,9),
- (8, 'Room104','WF 9:30AM',6,10),
- (9, 'Room105','ThF 9:30AM',7,3),
- (10, 'Room201','MT 9:30AM',8,1),
- (11, 'Room202','WF 9:30AM',9,4),
- (12, 'Room203','TW 9:30AM',10,5);



	sectionid [PK] integer	classroom character varying	date/time character varying	courseid integer	instructorid integer
1	1	Room101	MW 9:30AM	1	3
2	2	Room101	TTh 9:30AM	1	4
3	3	Room102	MW 9:30AM	2	7
4	4	Room102	TTh 9:30AM	2	8
5	5	Room103	MF 9:30AM	3	5
6	6	Room103	TTh 9:30AM	4	6
7	7	Room104	TTh 9:30AM	5	9
8	8	Room104	W 9:30AM 2:00PM	6	10
9	9	Room105	ThF 9:30AM	7	3
10	10	Room201	MT 9:30AM	8	1
11	11	Room202	WF 9:30AM	9	4
12	12	Room203	TW 9:30AM	10	5

SELECT*FROM section ORDER BY sectionid ASC;

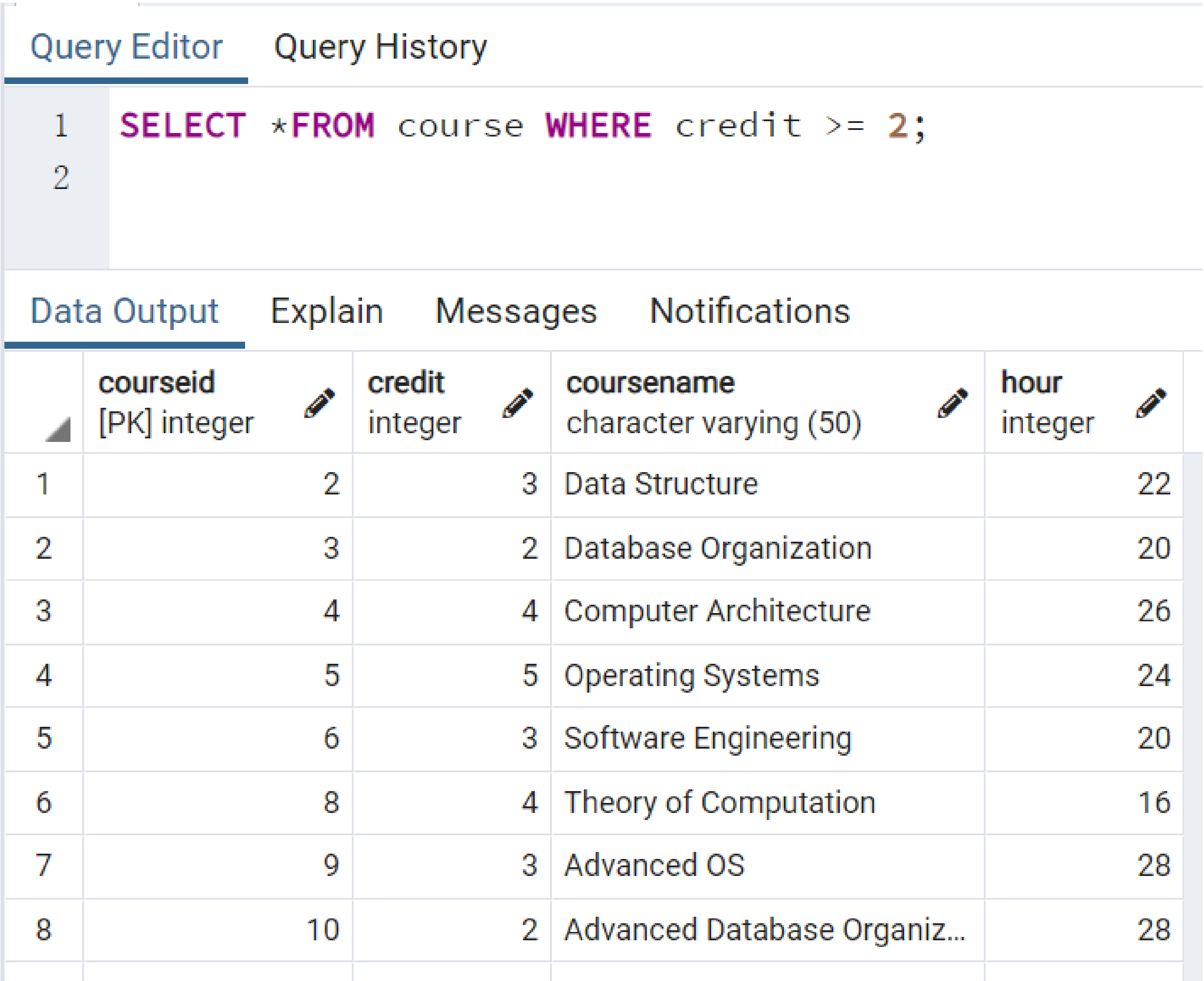


```
SELECT*FROM section
ORDER BY sectionid ASC;

```

	sectionid [PK] integer	classroom character varying	date/time character varying	courseid integer	instructorid integer
1	1	Room101	MW 9:30AM	1	3
2	2	Room101	TTh 9:30AM	1	4
3	3	Room102	MW 9:30AM	2	7
4	4	Room102	TTh 9:30AM	2	8
5	5	Room103	MF 9:30AM	3	5
6	6	Room103	TTh 9:30AM	4	6
7	7	Room104	TTh 9:30AM	5	9
8	8	Room104	W 9:30AM 2:00PM	6	10
9	9	Room105	ThF 9:30AM	7	3
10	10	Room201	MT 9:30AM	8	1
11	11	Room202	WF 9:30AM	9	4
12	12	Room203	TW 9:30AM	10	5

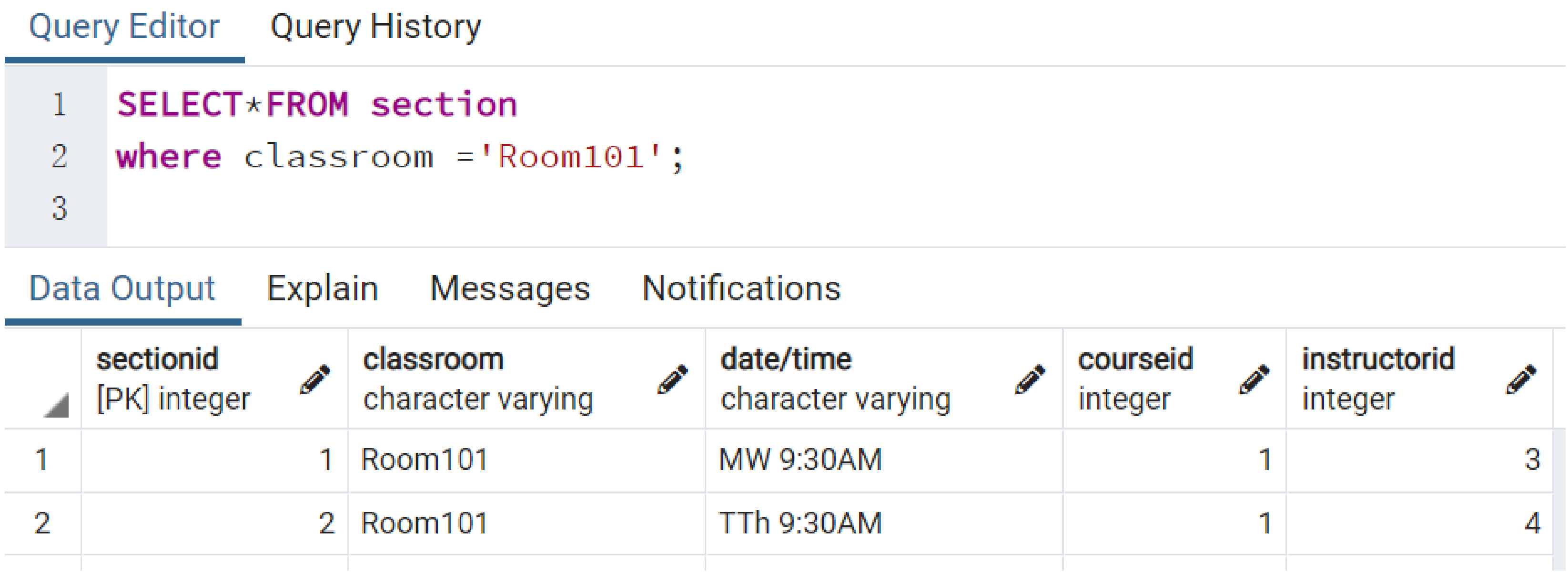
SELECT *FROM course WHERE credits >= 2;



```
SELECT *FROM course WHERE credit >= 2;
```

	courseid [PK] integer	credit integer	coursename character varying (50)	hour integer
1	2	3	Data Structure	22
2	3	2	Database Organization	20
3	4	4	Computer Architecture	26
4	5	5	Operating Systems	24
5	6	3	Software Engineering	20
6	8	4	Theory of Computation	16
7	9	3	Advanced OS	28
8	10	2	Advanced Database Organiz...	28

SELECT*FROM section where classroom ='Room101';



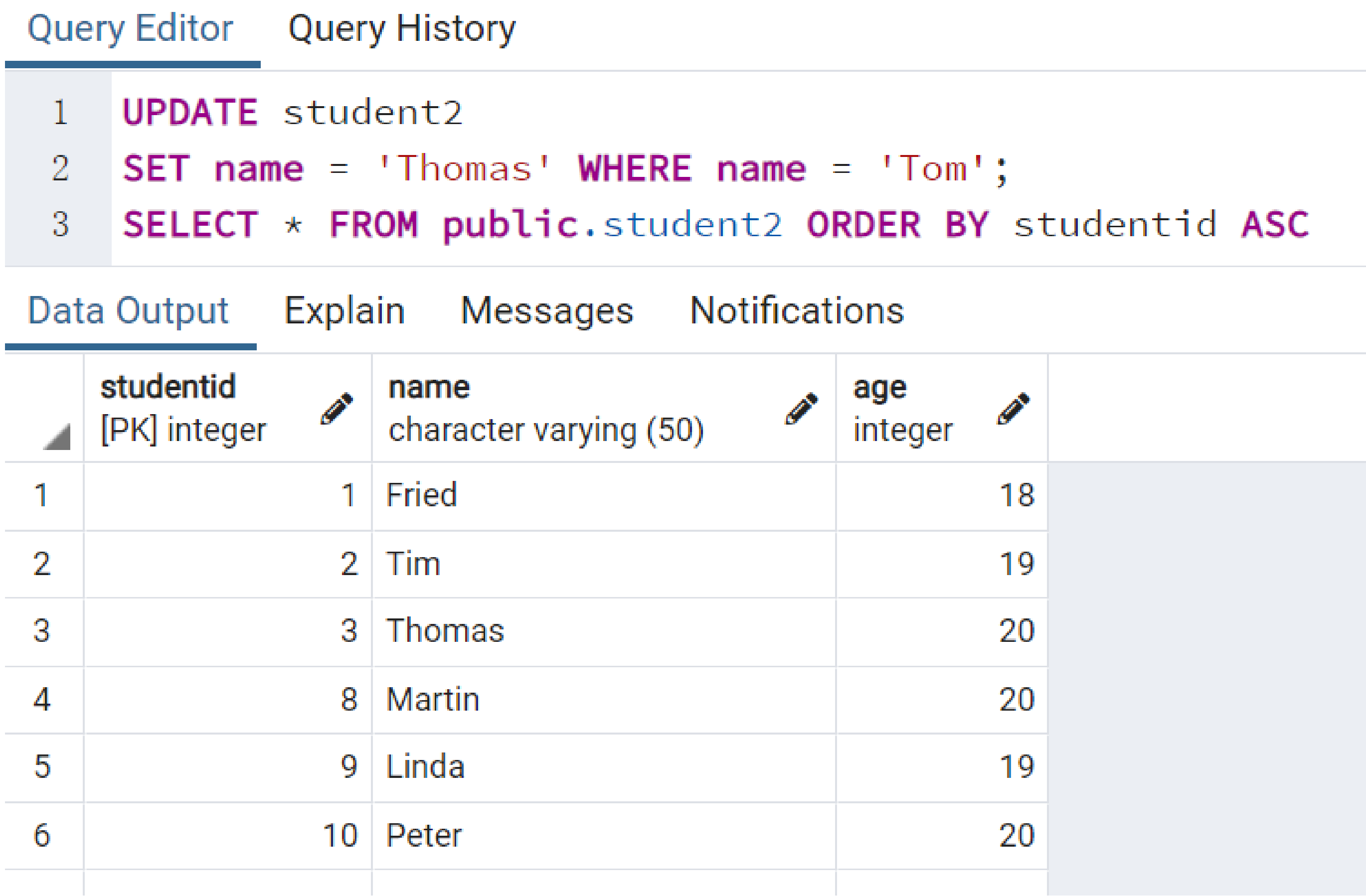
```
SELECT*FROM section
where classroom ='Room101';
```

	sectionid [PK] integer	classroom character varying	date/time character varying	courseid integer	instructorid integer
1	1	Room101	MW 9:30AM	1	3
2	2	Room101	TTh 9:30AM	1	4

UPDATE student2

SET name = 'Thomas' WHERE name = 'Tom';

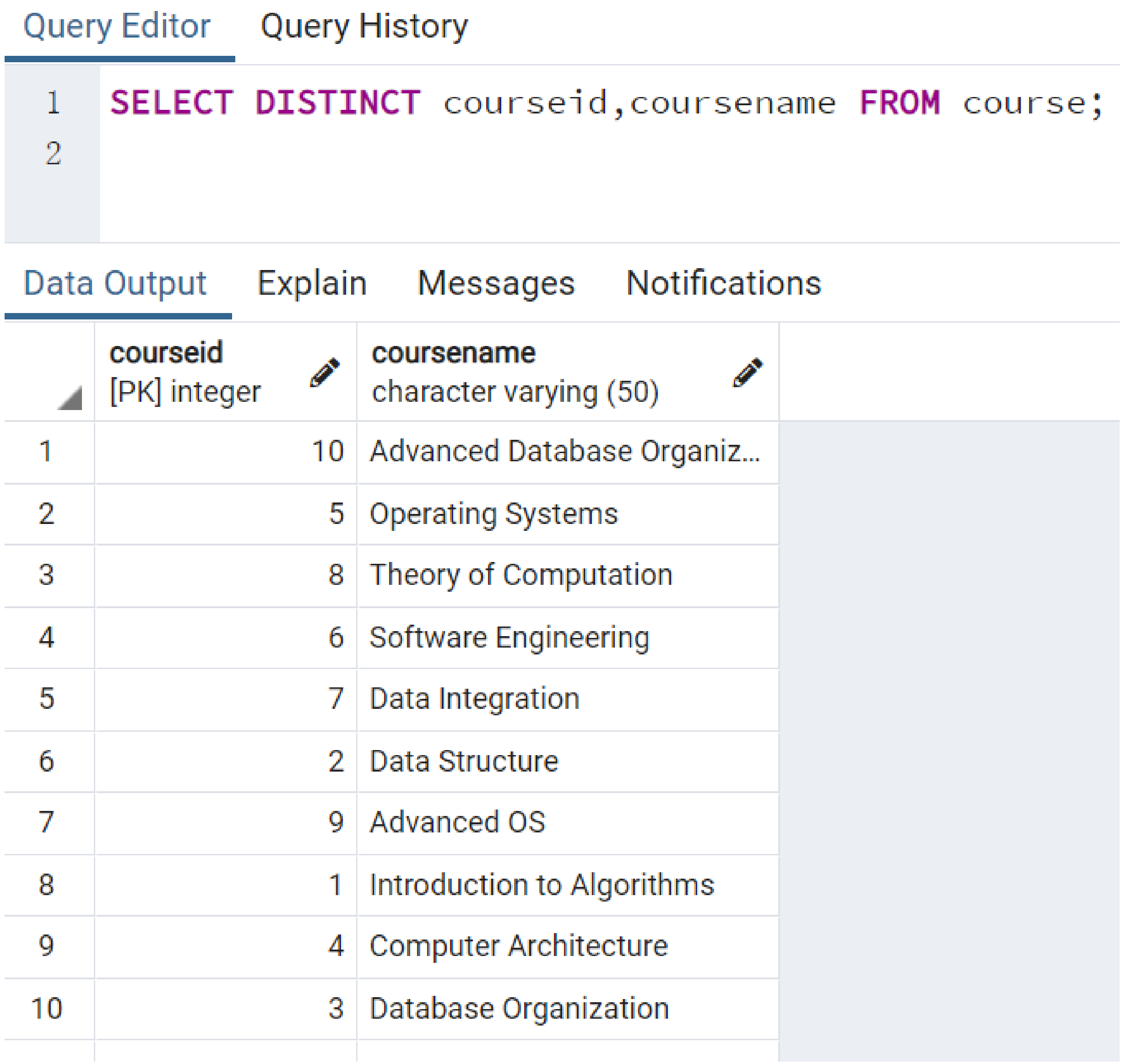
SELECT * FROM public.student2 ORDER BY studentid ASC



```
UPDATE student2
SET name = 'Thomas' WHERE name = 'Tom';
SELECT * FROM public.student2 ORDER BY studentid ASC
```

	studentid [PK] integer	name character varying (50)	age integer
1	1	Fried	18
2	2	Tim	19
3	3	Thomas	20
4	8	Martin	20
5	9	Linda	19
6	10	Peter	20

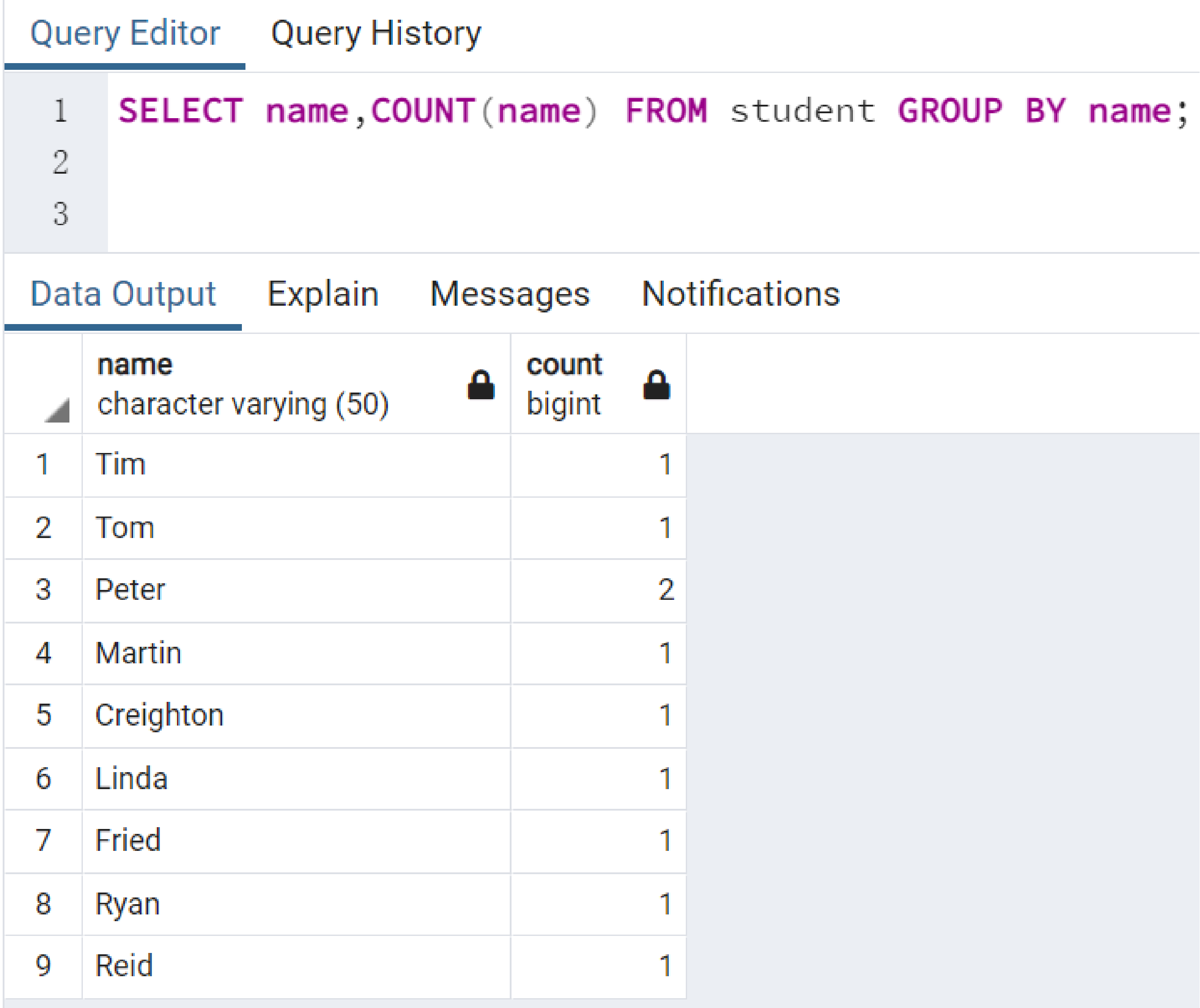
SELECT DISTINCT courseid,coursename FROM course;



```
SELECT DISTINCT courseid,coursename FROM course;
```

	courseid [PK] integer	coursename character varying (50)
1	10	Advanced Database Organiz...
2	5	Operating Systems
3	8	Theory of Computation
4	6	Software Engineering
5	7	Data Integration
6	2	Data Structure
7	9	Advanced OS
8	1	Introduction to Algorithms
9	4	Computer Architecture
10	3	Database Organization

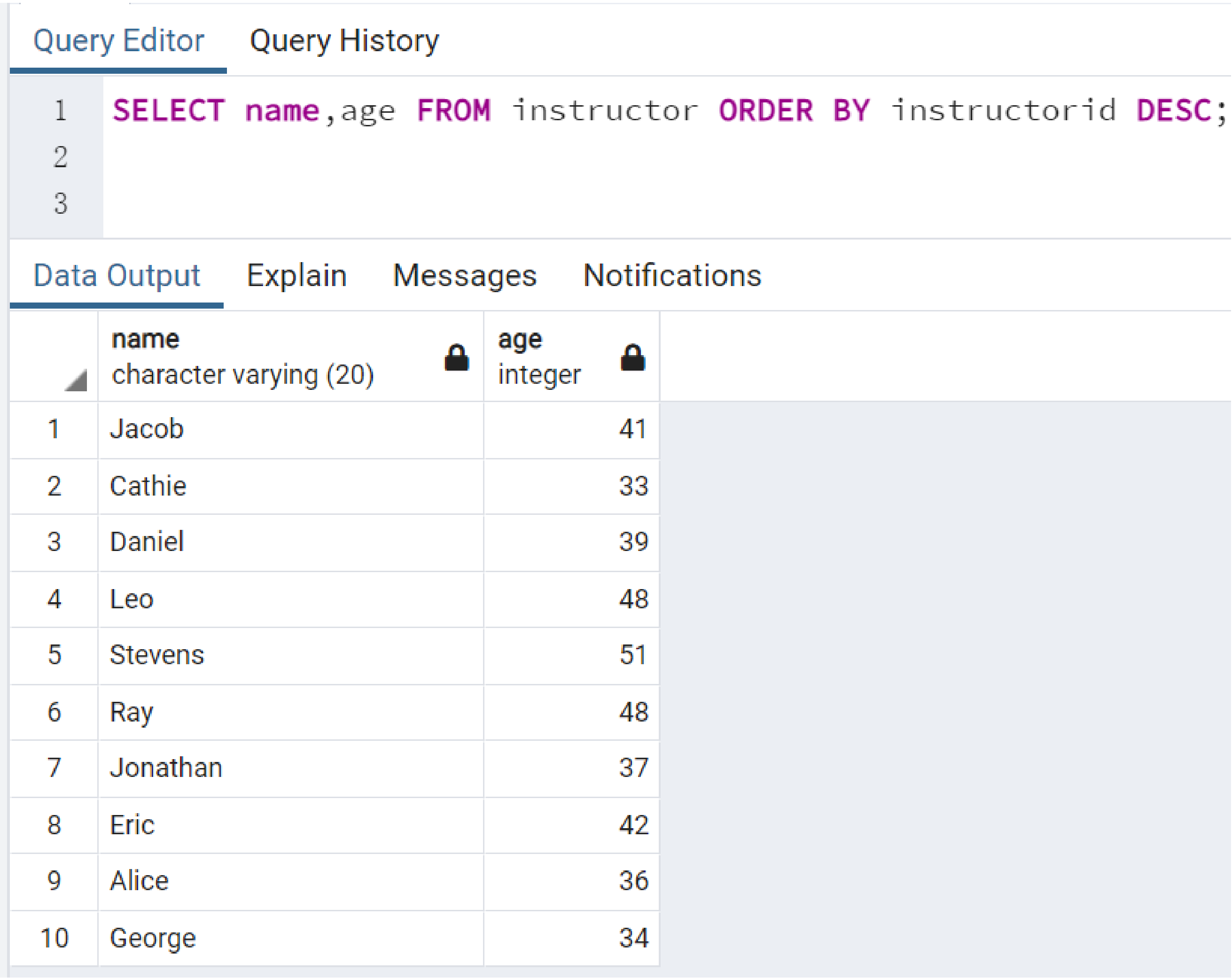
SELECT name,COUNT(name) FROM student GROUP BY name;



```sql
SELECT name,COUNT(name) FROM student GROUP BY name;
```

	name character varying (50)	count bigint
1	Tim	1
2	Tom	1
3	Peter	2
4	Martin	1
5	Creighton	1
6	Linda	1
7	Fried	1
8	Ryan	1
9	Reid	1

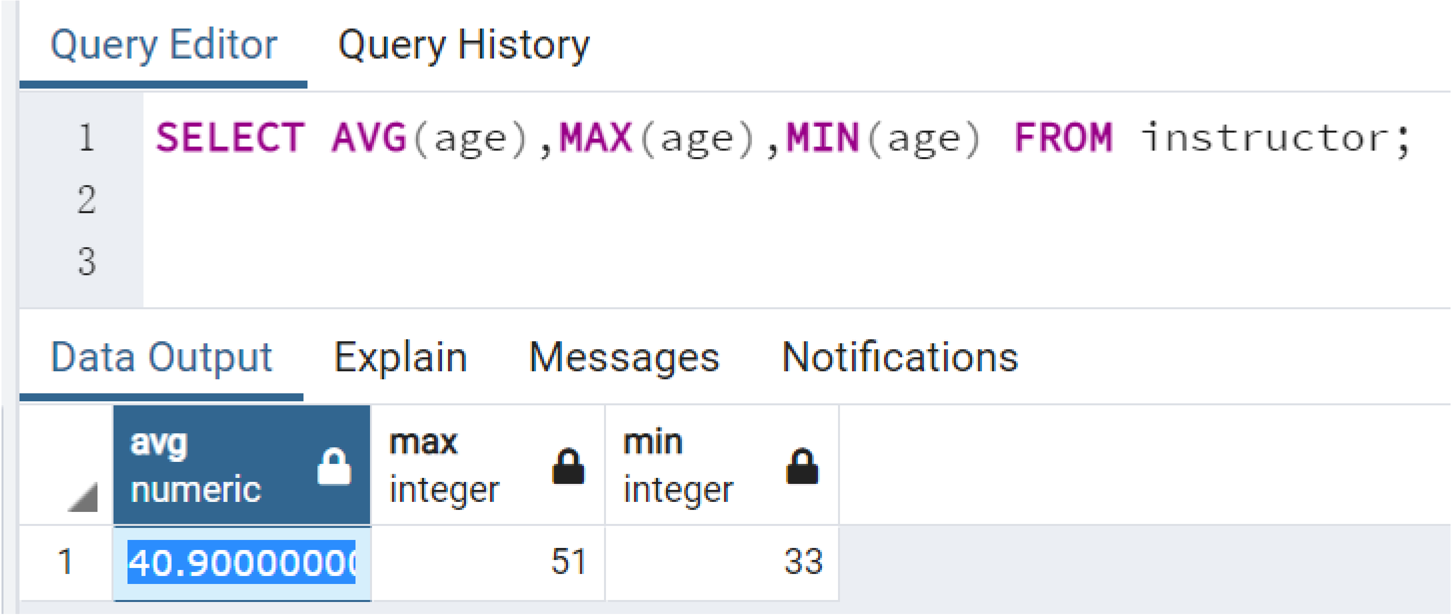
SELECT name,age FROM instructor ORDER BY instructorid DESC;



```sql
SELECT name,age FROM instructor ORDER BY instructorid DESC;
```

	name character varying (20)	age integer
1	Jacob	41
2	Cathie	33
3	Daniel	39
4	Leo	48
5	Stevens	51
6	Ray	48
7	Jonathan	37
8	Eric	42
9	Alice	36
10	George	34

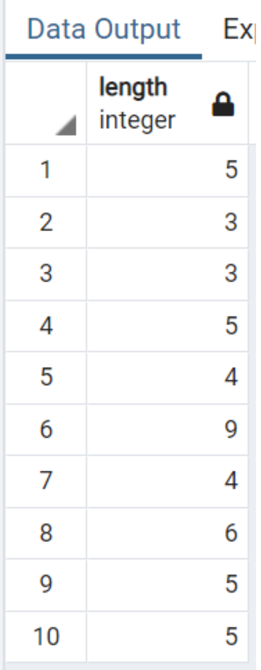
SELECT AVG(age),MAX(age),MIN(age) FROM instructor;



```sql
SELECT AVG(age),MAX(age),MIN(age) FROM instructor;
```

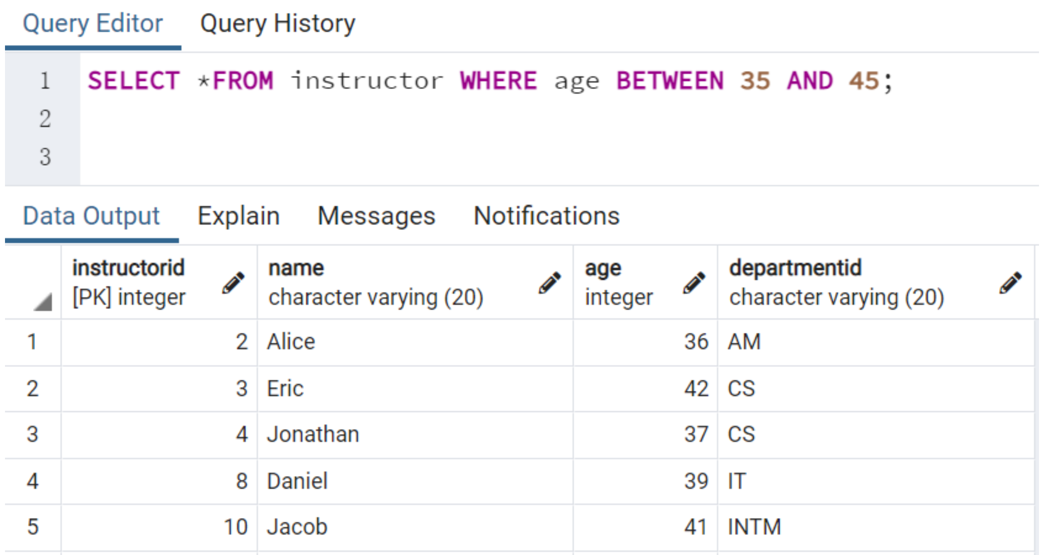
	avg numeric	max integer	min integer
1	40.9000000	51	33

select length(name) from student；



	length integer
1	5
2	3
3	3
4	5
5	4
6	9
7	4
8	6
9	5
10	5

SELECT *FROM instructor WHERE age BETWEEN 35 AND 45;

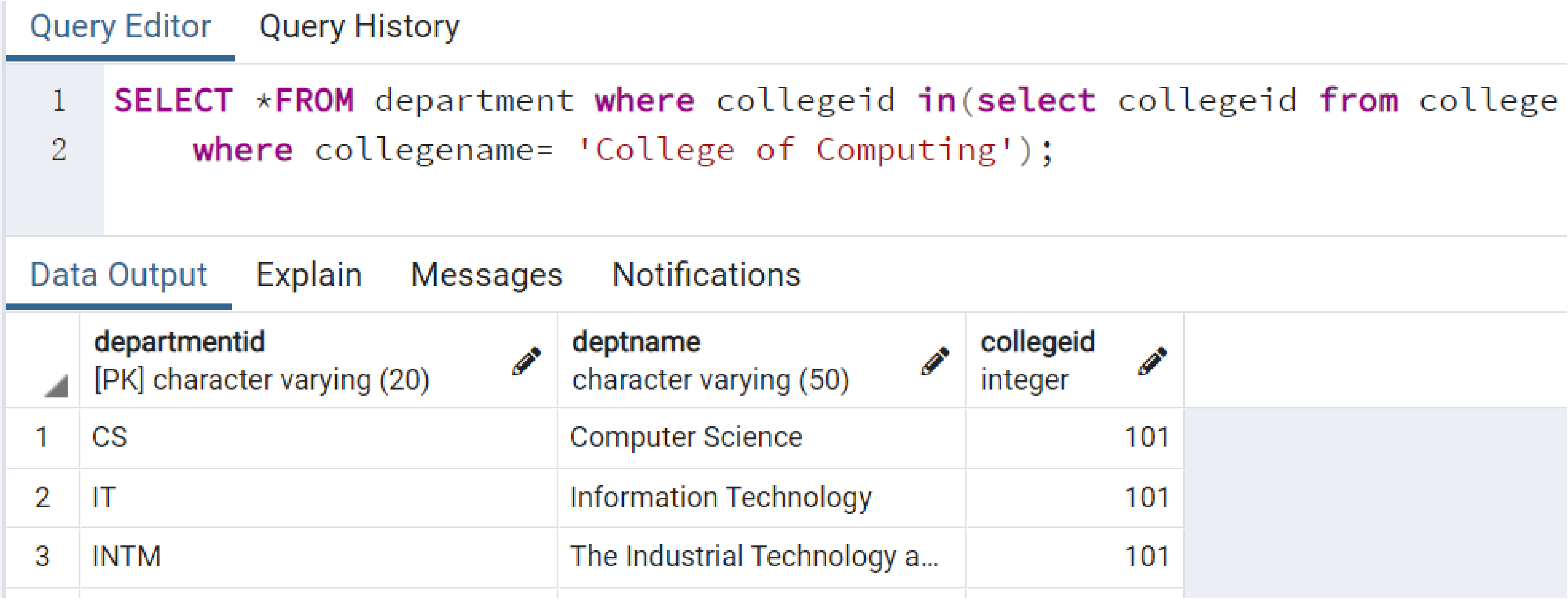


```
SELECT *FROM instructor WHERE age BETWEEN 35 AND 45;
```

	instructorid [PK] integer	name character varying (20)	age integer	departmentid character varying (20)
1	2	Alice	36	AM
2	3	Eric	42	CS
3	4	Jonathan	37	CS
4	8	Daniel	39	IT
5	10	Jacob	41	INTM

What are all the departments from College of Computing?

SELECT *FROM department where collegeid in(select collegeid from college
where collegename= 'College of Computing');

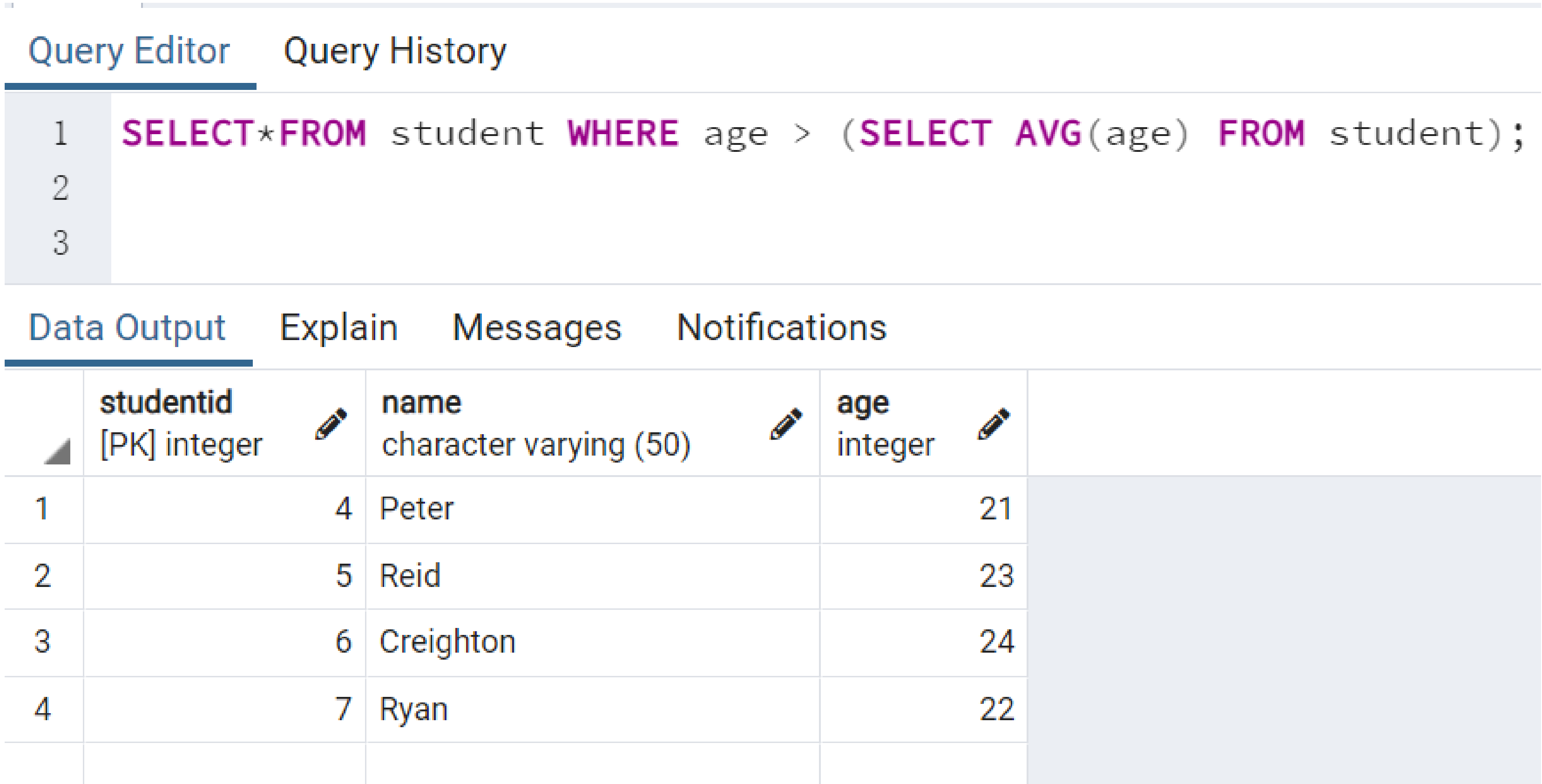


```
SELECT *FROM department where collegeid in(select collegeid from college
    where collegename= 'College of Computing');
```

	departmentid [PK] character varying (20)	deptname character varying (50)	collegeid integer
1	CS	Computer Science	101
2	IT	Information Technology	101
3	INTM	The Industrial Technology a...	101

Using a subquery, list all the students with ages above the average.

SELECT *FROM student where age>(select AVG(age) from student);



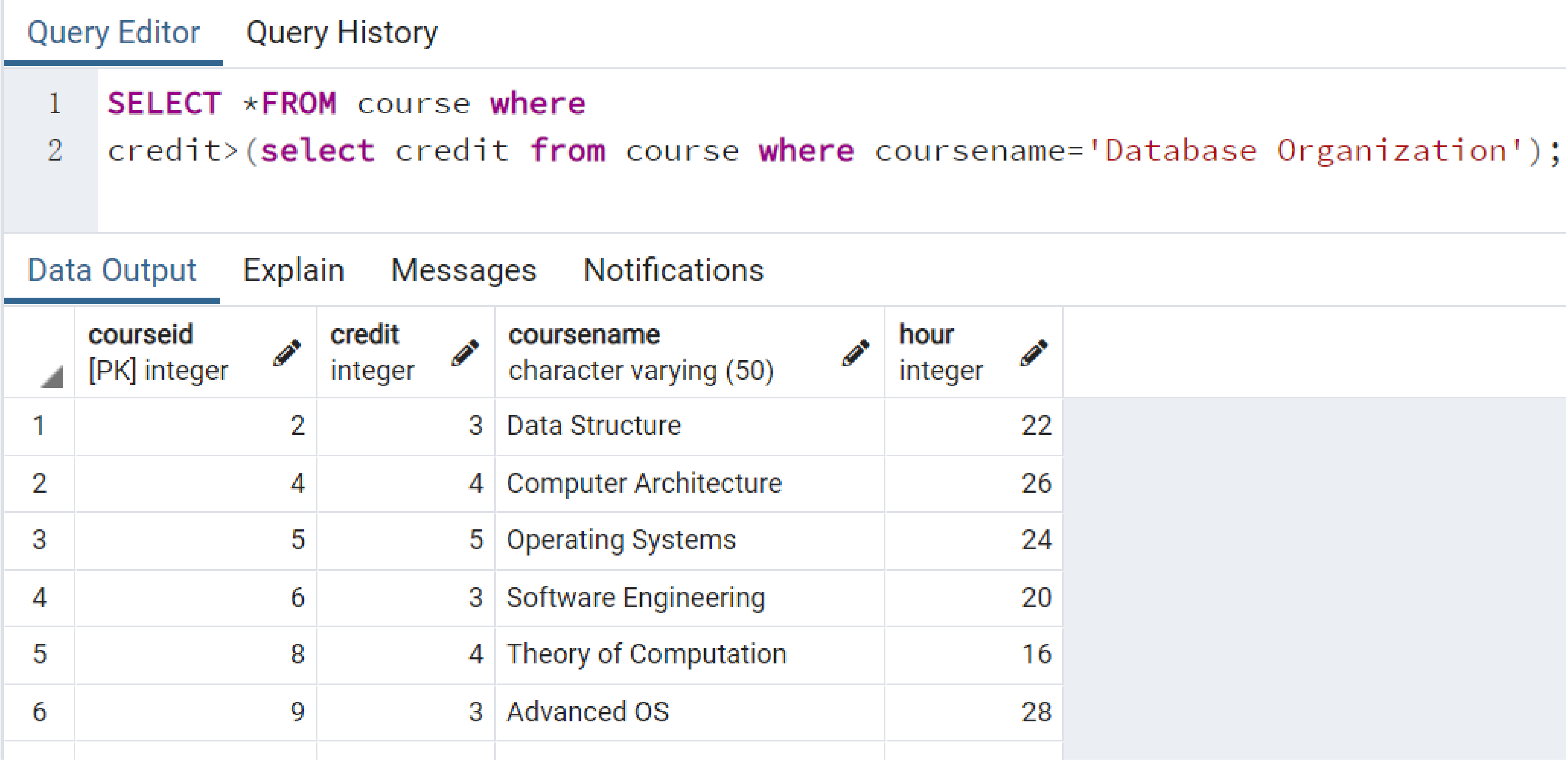
```
SELECT*FROM student WHERE age > (SELECT AVG(age) FROM student);
```

	studentid [PK] integer	name character varying (50)	age integer
1	4	Peter	21
2	5	Reid	23
3	6	Creighton	24
4	7	Ryan	22

Using a subquery, list courses with credits larger than Database Organization.

SELECT *FROM course where

credit>(select credit from course where coursename='Database Organization');

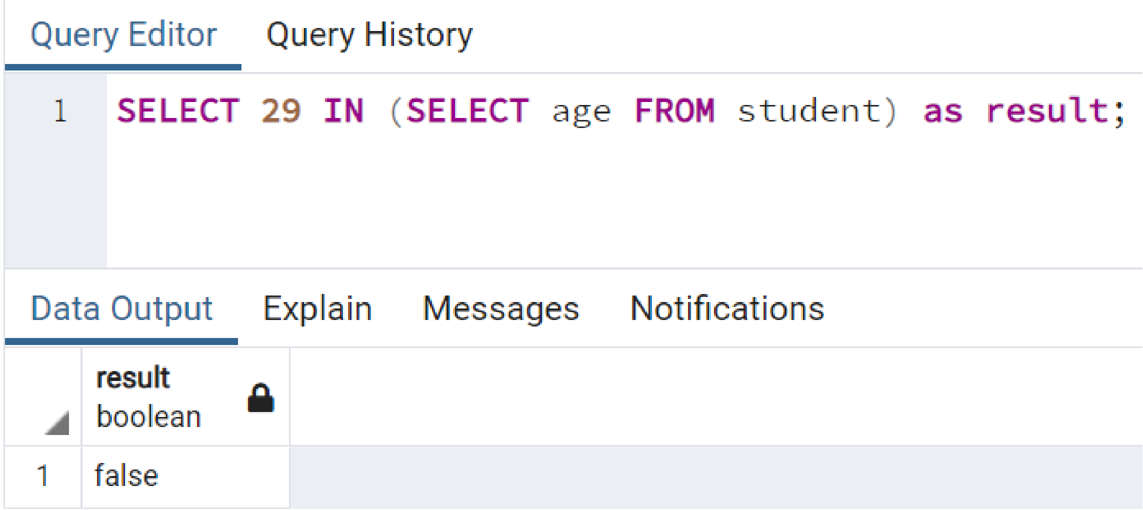


```sql
SELECT *FROM course where
credit>(select credit from course where coursename='Database Organization');
```

	courseid [PK] integer	credit integer	coursename character varying (50)	hour integer
1	2	3	Data Structure	22
2	4	4	Computer Architecture	26
3	5	5	Operating Systems	24
4	6	3	Software Engineering	20
5	8	4	Theory of Computation	16
6	9	3	Advanced OS	28

Using a subquery, to see if a student ages 29 exists within the student table:

SELECT 29 in (select age from student) as result;

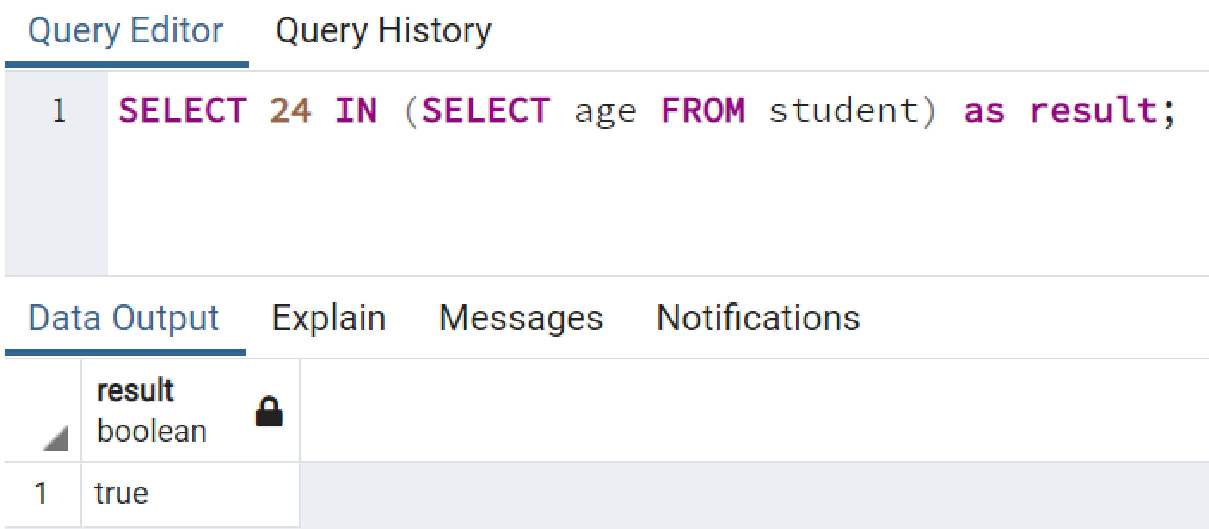


```
SELECT 29 IN (SELECT age FROM student) as result;
```

	result boolean
1	false

Using a subquery, to see if a student ages 24 exists within the student table:

SELECT 24 in (select age from student) as result;

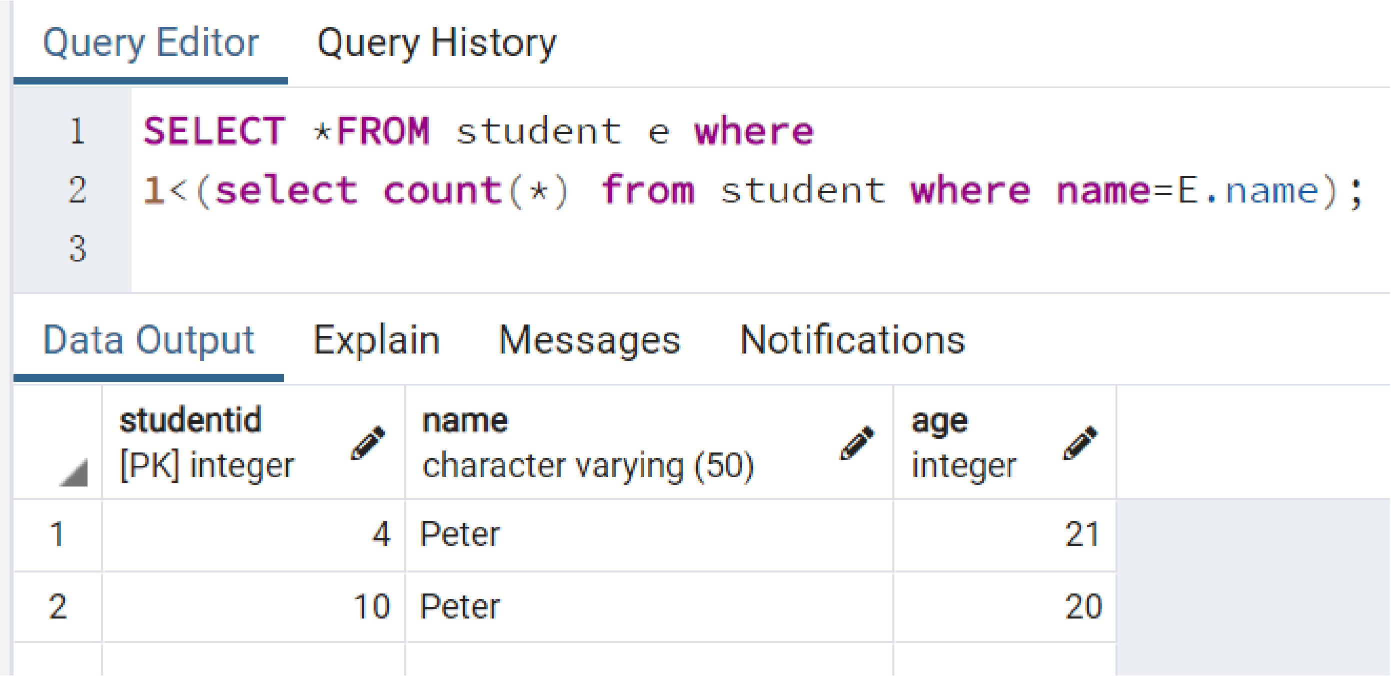


```
SELECT 24 IN (SELECT age FROM student) as result;
```

```
   result
   boolean
1  true
```

Using a subquery, to check if 2 or more students have the same name.

```
SELECT *FROM student e where
1<(select count(*) from student where name=E.name);
```



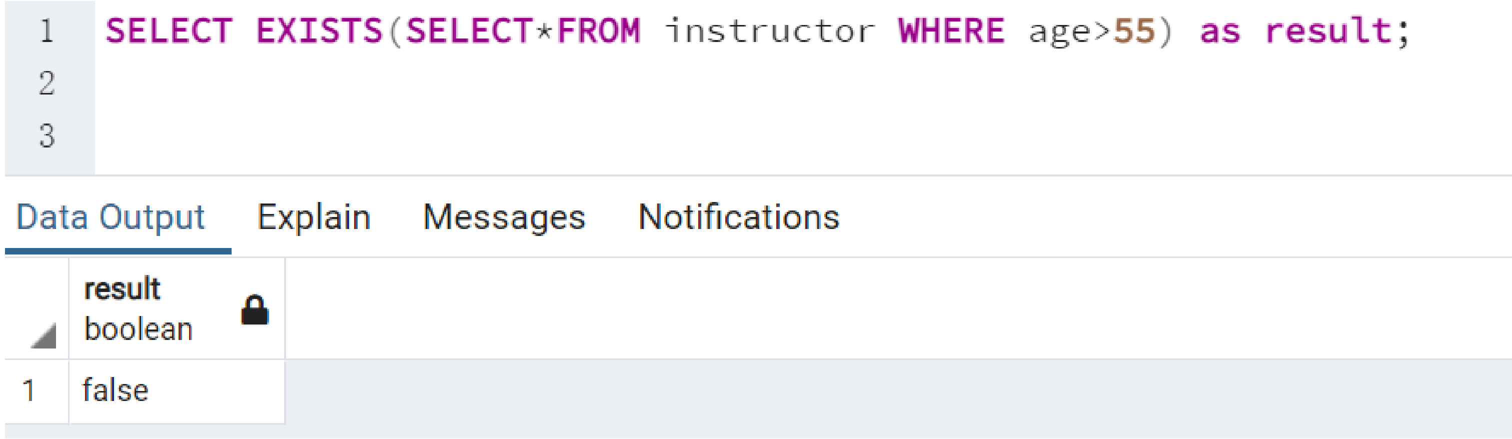
```
SELECT *FROM student e where
1<(select count(*) from student where name=E.name);

```

	studentid [PK] integer	name character varying (50)	age integer
1	4	Peter	21
2	10	Peter	20

Using a subquery, to see if there're any instructors with ages above 55 exist.

SELECT EXISTS(SELECT*FROM instructor where age>55) as result;

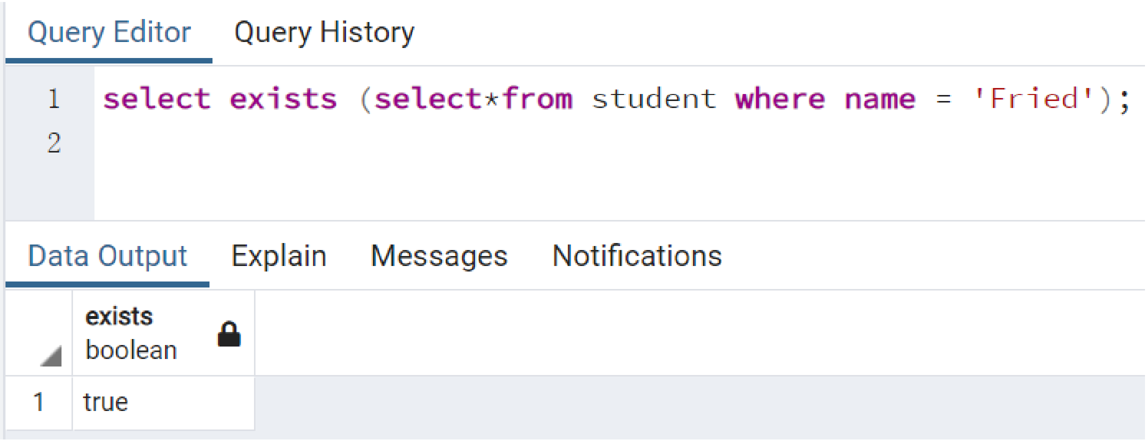


```
SELECT EXISTS(SELECT*FROM instructor WHERE age>55) as result;
```

```
   result
   boolean
1  false
```

Using a subquery, to see if there’re any instructors with name Fried exist.

SELECT EXISTS(SELECT*FROM student where name = 'Fried') as result;

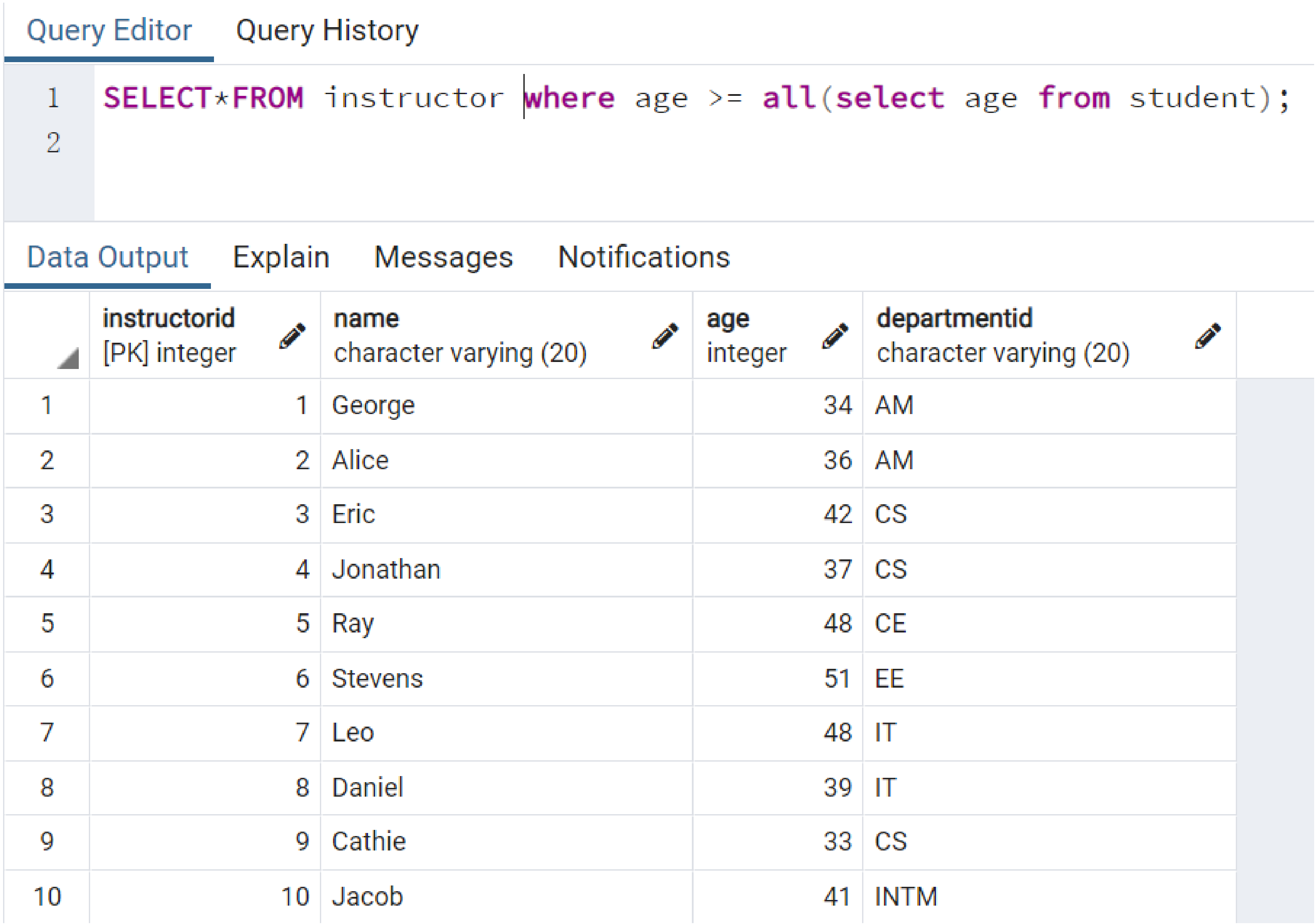


```
select exists (select*from student where name = 'Fried');
```

```
   exists
   boolean
1  true
```

Using a subquery, list all the instructors whose ages above all students' ages.

SELECT*FROM instructor where age >= all(select age from student);



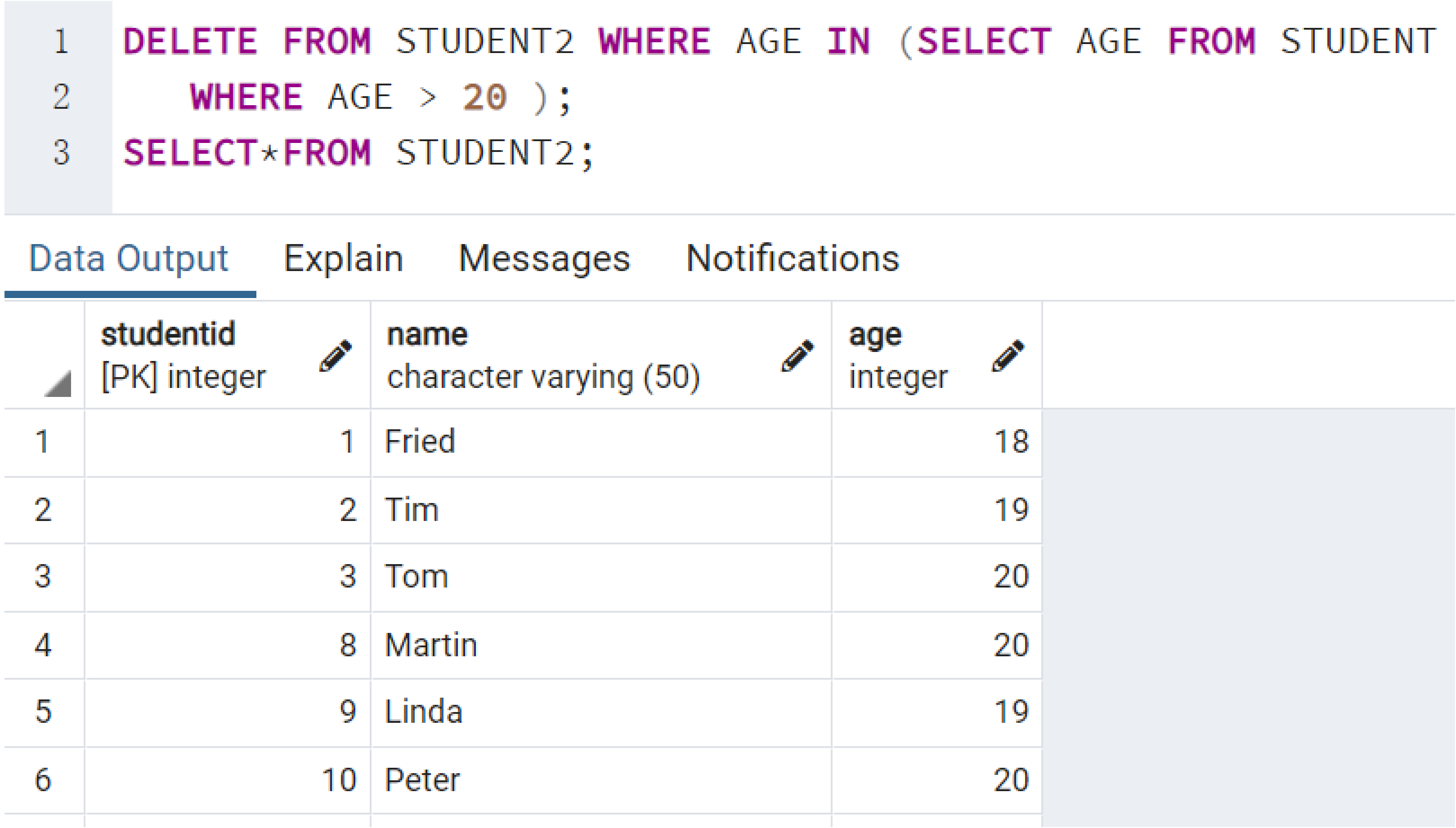
```sql
SELECT*FROM instructor where age >= all(select age from student);
```

	instructorid [PK] integer	name character varying (20)	age integer	departmentid character varying (20)
1	1	George	34	AM
2	2	Alice	36	AM
3	3	Eric	42	CS
4	4	Jonathan	37	CS
5	5	Ray	48	CE
6	6	Stevens	51	EE
7	7	Leo	48	IT
8	8	Daniel	39	IT
9	9	Cathie	33	CS
10	10	Jacob	41	INTM

Using a subquery, delete all the students with ages above 20 from student2(which is a copy of table student)

DELETE*FROM STUDENT2 WHERE AGE IN (select age from student WHERE AGE >20);

SELECT*FROM STUDENT2;



```
DELETE FROM STUDENT2 WHERE AGE IN (SELECT AGE FROM STUDENT
   WHERE AGE > 20 );
SELECT*FROM STUDENT2;
```

	studentid [PK] integer	name character varying (50)	age integer
1	1	Fried	18
2	2	Tim	19
3	3	Tom	20
4	8	Martin	20
5	9	Linda	19
6	10	Peter	20