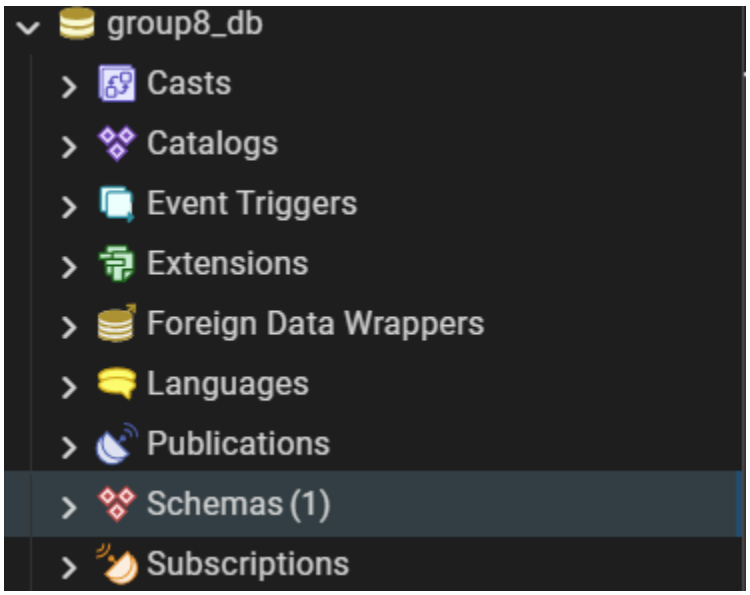
SENG550 Assignment 1

Contribution

Student Name	UCID	Sections Contributed
Zach Pelkey	30156783	Q1, Q2, Q3
Hamza	30144227	Q4, Q5, Q6
Nicola Savino	30129329	Q7, Q8 Document Creation + submission & verification

Question 1

1. Create a new database groupX db.



2. Create a students table with columns: id, name, age, major.

```sql
CREATE TABLE students (
    id      SERIAL  PRIMARY KEY,
    name    TEXT    NOT NULL,
    age     INT     NOT NULL,
    major   TEXT    NOT NULL
);
```

3. Insert at least 4 rows of data.

```sql
INSERT INTO students (name, age, major) VALUES
('John', 19, 'Arts'),
('Mark', 27, 'Computer Science'),
('Luke', 21, 'Engineering'),
('Matthew', 18, 'Engineering');
```

4. Queries:

• Show all students.

```sql
--Question 1 Queries
SELECT * FROM students;
```

	id [PK] integer	name text	age integer	major text
1	1	John	19	Arts
2	2	Mark	27	Computer Scien...
3	3	Luke	21	Engineering
4	4	Matthe...	18	Engineering

• Show students older than 21.

```
22  SELECT * FROM students WHERE age > 21;
23
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	id [PK] integer	name text	age integer	major text
1	2	Mark	27	Computer Scien...

• Show only names of students in “Computer Science”

```
24  SELECT name FROM students WHERE major = 'Computer Science';
25
26  -- --Question 2
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	name text
1	Mark

Question 2

1. Add a new column email.

```
--Question 2
ALTER TABLE students
ADD COLUMN email VARCHAR(100);
```

2. Update two students with valid email addresses.

```sql
UPDATE students
SET email = 'john@icloud.com'
WHERE name = 'John';

UPDATE students
SET email = 'mark@icloud.com'
WHERE name = 'Mark';
```

3. Query: list all students showing name and email only.

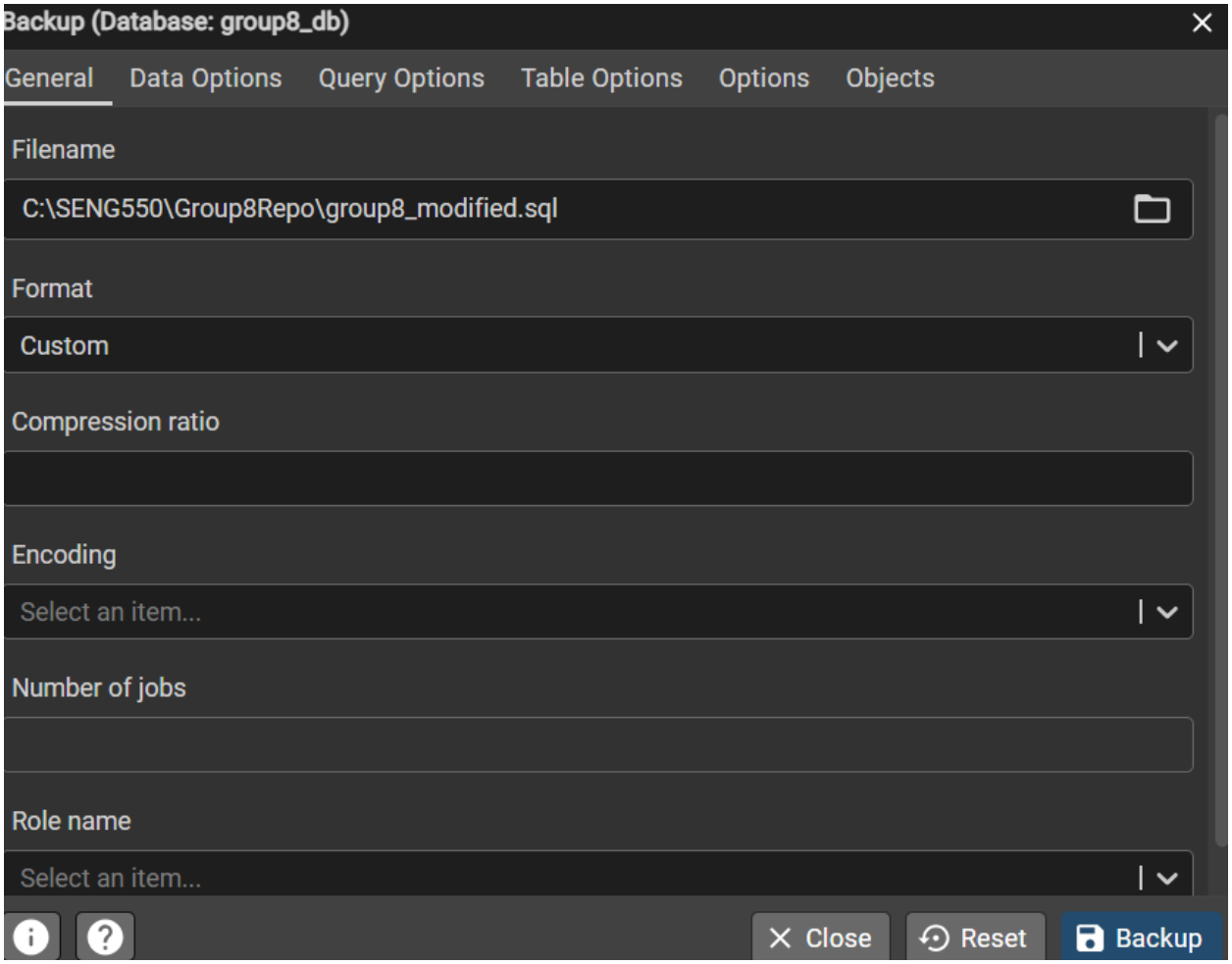
```sql
SELECT name, email FROM students
```

Data Output | Messages | Notifications

Showing rows: 1 to 4 | Page No: 1 of 1

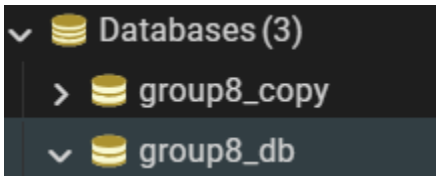
	name text	email character varying (100)
1	Luke	[null]
2	Matthe...	[null]
3	John	john@icloud.com
4	Mark	mark@icloud.com

4. Backup database as groupX modified.sql.



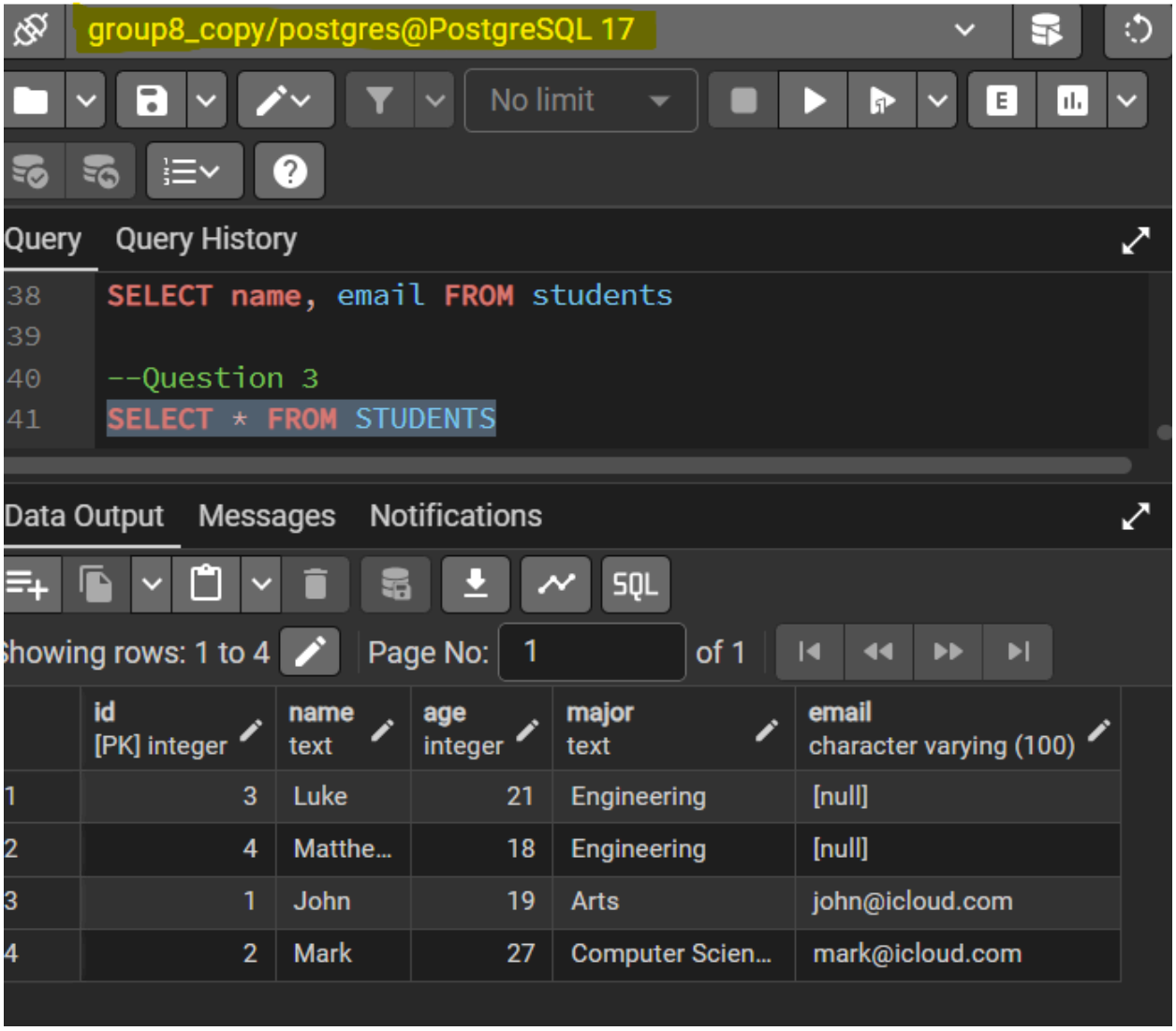
Question 3

1. Create groupX copy.

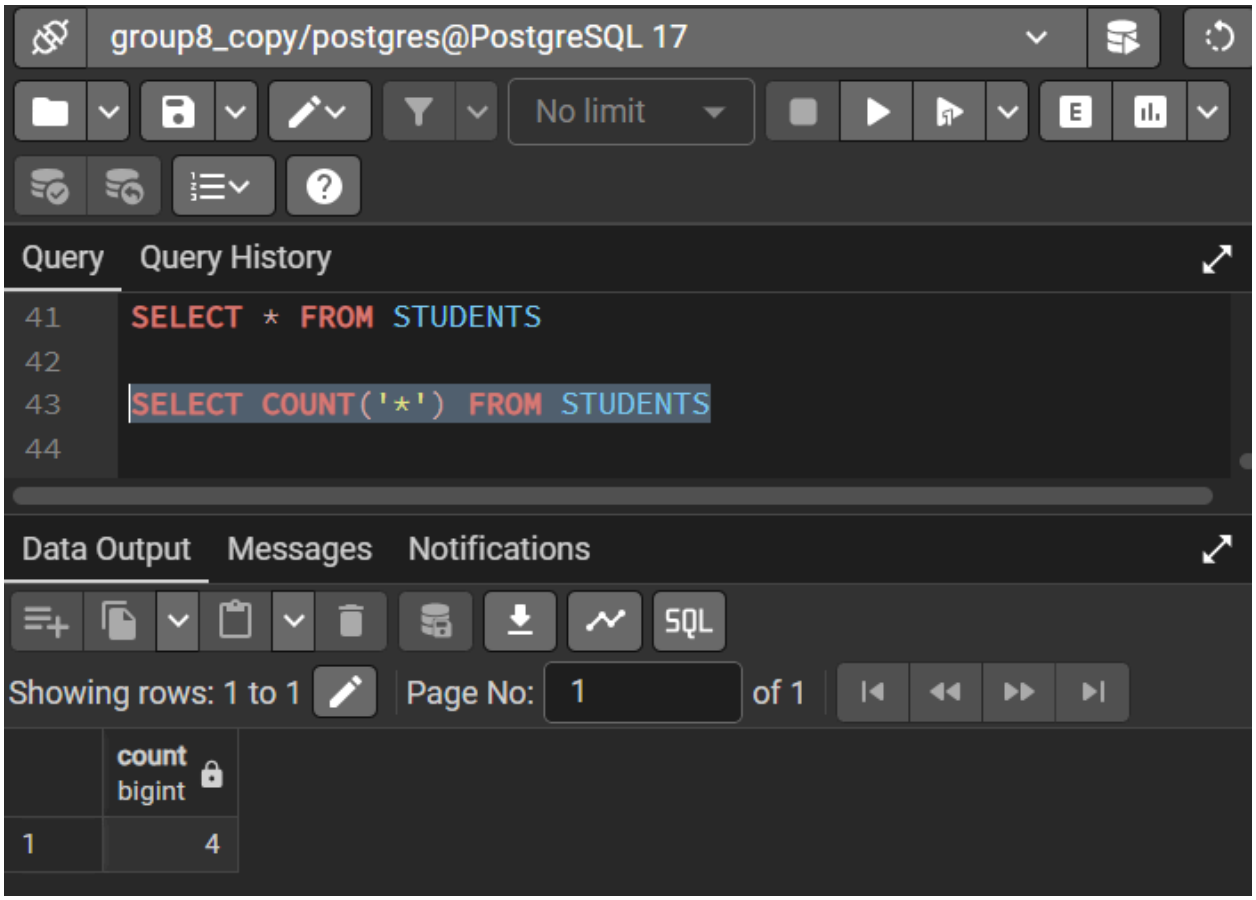


3. Queries:

• Show all rows from students.



• Count the total number of students in this copy.



Question 4

1. Create products.csv with at least 8 products. Columns: product id,name,price.

Group8Repo > Assignment1 > products.csv

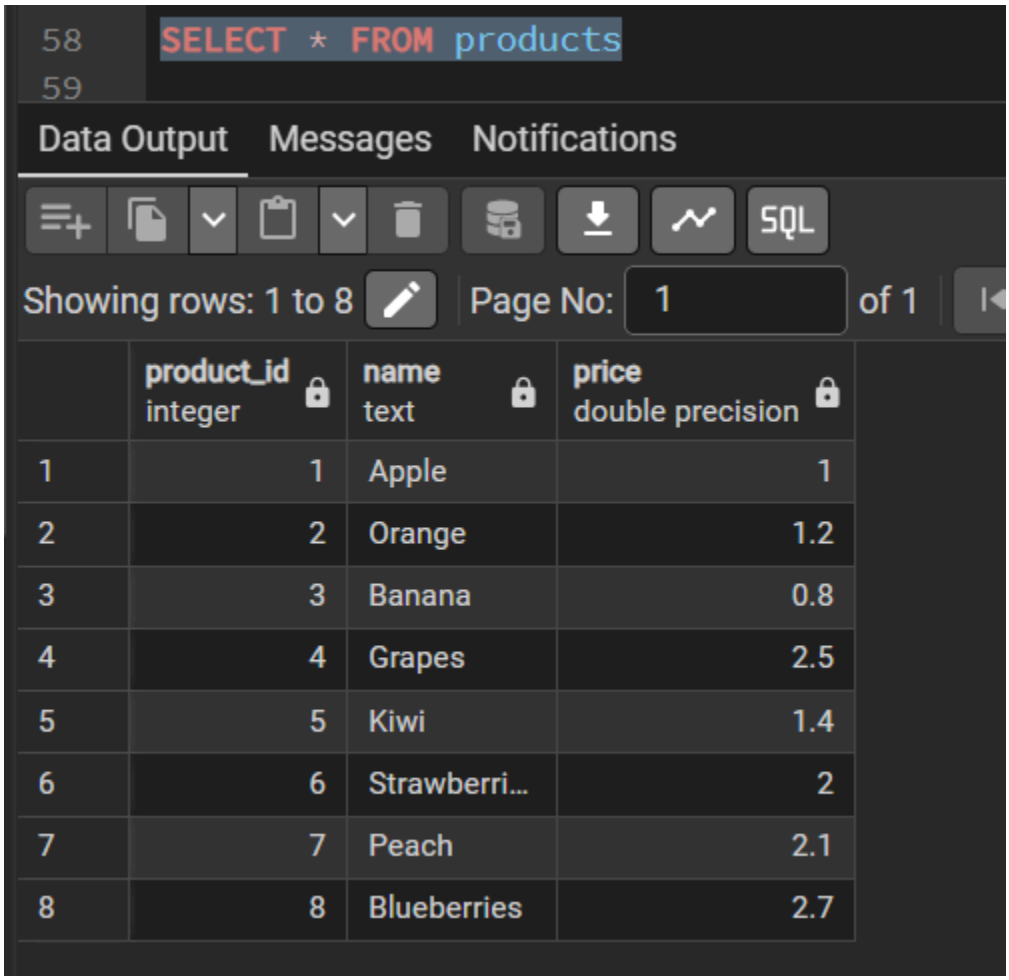
```
1  product_id,name,price
2  1,Apple,1.00
3  2,Orange,1.20
4  3,Banana,0.80
5  4,Grapes,2.50
6  5,Kiwi,1.40
7  6,Strawberries,2.00
8  7,Peach,2.10
9  8,Blueberries,2.70
```

2. Create products table and import the CSV.

```sql
-- Question 4
CREATE TABLE products(
    product_id  int UNIQUE,
    name    TEXT    NOT NULL,
    price   FLOAT   NOT NULL
);
```

3. Queries:

• Show all products.



```sql
58  SELECT * FROM products
59
```

Data Output | Messages | Notifications

SQL

Showing rows: 1 to 8 | Page No: 1 of 1

	product_id integer	name text	price double precision
1	1	Apple	1
2	2	Orange	1.2
3	3	Banana	0.8
4	4	Grapes	2.5
5	5	Kiwi	1.4
6	6	Strawberri...	2
7	7	Peach	2.1
8	8	Blueberries	2.7

• Show products cheaper than 2.00.

```
60  SELECT * FROM products WHERE price < 2.00
61
62
63
```

Data Output | Messages | Notifications

Showing rows: 1 to 4 | Page No: 1 of 1

	product_id integer	name text	price double precision
1	1	Apple	1
2	2	Orange	1.2
3	3	Bana...	0.8
4	5	Kiwi	1.4

• Find the maximum product price.

```
62  SELECT MAX(price) as MaximumPrice FROM products
63
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	maximumprice double precision
1	2.7

Question 5

Delete one student from students.

```
--Question 5
DELETE FROM students WHERE name = 'John';
```

Query all remaining students ordered by age descending.

```
67  SELECT * FROM students ORDER BY AGE DESC;
68
69
70
```

Data Output | Messages | Notifications

Showing rows: 1 to 3 | Page No: 1 of 1

	id [PK] integer	name text	age integer	major text	email character varying (100)
1	2	Mark	27	Computer Scien...	mark@icloud.com
2	3	Luke	21	Engineering	[null]
3	4	Matthe...	18	Engineering	[null]

Query: show only the youngest student.

```
69  SELECT *
70  FROM students
71  WHERE age  = (SELECT(MIN(age)) FROM students)
72
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	id [PK] integer	name text	age integer	major text	email character varying (100)
1	4	Matthe...	18	Engineeri...	[null]

Question 6

1. Query: number of students per major.

```
73  --Question 6
74  SELECT major, COUNT ('*') FROM students GROUP BY major;
75
76
```

Data Output | Messages | Notifications

Showing rows: 1 to 2 | Page No: 1 of 1

	major text	count bigint
1	Engineering	2
2	Computer Scien...	1

2. Query: average price of products.

```
76  SELECT AVG(price) as averageprice from products;
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	averageprice double precision
1	1.7125

3. Query: total value of all products combined (SUM(price)).

```
78  SELECT SUM(price) as Total from products;
79
80
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	total double precision
1	13.7

Question 7

1. Create an enrollments table with columns: enroll id, student id (FK), course.

```
80  -- Question 7 -
81  CREATE TABLE enrollments (
82      enroll_id   SERIAL  PRIMARY KEY,
83      student_id  INT REFERENCES students(id),
84      course      TEXT    NOT NULL
85  );
```

2. Insert at least 3 rows linking students to courses.

```
INSERT INTO enrollments (student_id, course)
SELECT id, 'CPSC 453' FROM students WHERE name = 'Mark';

INSERT INTO enrollments (student_id, course)
SELECT id, 'SENG 550' FROM students WHERE name = 'Luke';

INSERT INTO enrollments (student_id, course)
SELECT id, 'SENG 550' FROM students WHERE name = 'Matthew';

INSERT INTO enrollments (student_id, course)
SELECT id, 'ENSF 460' FROM students WHERE name = 'Matthew';
```

3. Query using JOIN: list student name and their course.

```
98  SELECT students.name, enrollments.course
99  FROM enrollments
100 JOIN students ON students.id = enrollments.student_id
101 ORDER BY students.name, enrollments.course;
```

Data Output | Messages | Notifications

Showing rows: 1 to 4 | Page No: 1 of 1

	name text	course text
1	Luke	SENG 5...
2	Mark	CPSC 4...
3	Matthew	ENSF 460
4	Matthew	SENG 5...

4. Query: show only students who are enrolled in more than one course

```
103  SELECT students.name
104  FROM enrollments
105  JOIN students on students.id = enrollments.student_id
106  GROUP BY students.name
107  HAVING COUNT(*) > 1
```

Data Output | Messages | Notifications

SQL

Showing rows: 1 to 1 | Page No: 1 of 1

	name text
1	Matthew

Question 8

1. Insert yourself as a student in the students table.

```
-- Question 8
INSERT INTO students (name, age, major)
VALUES ('Nicola Savino', 22, 'Software Engineering')
```

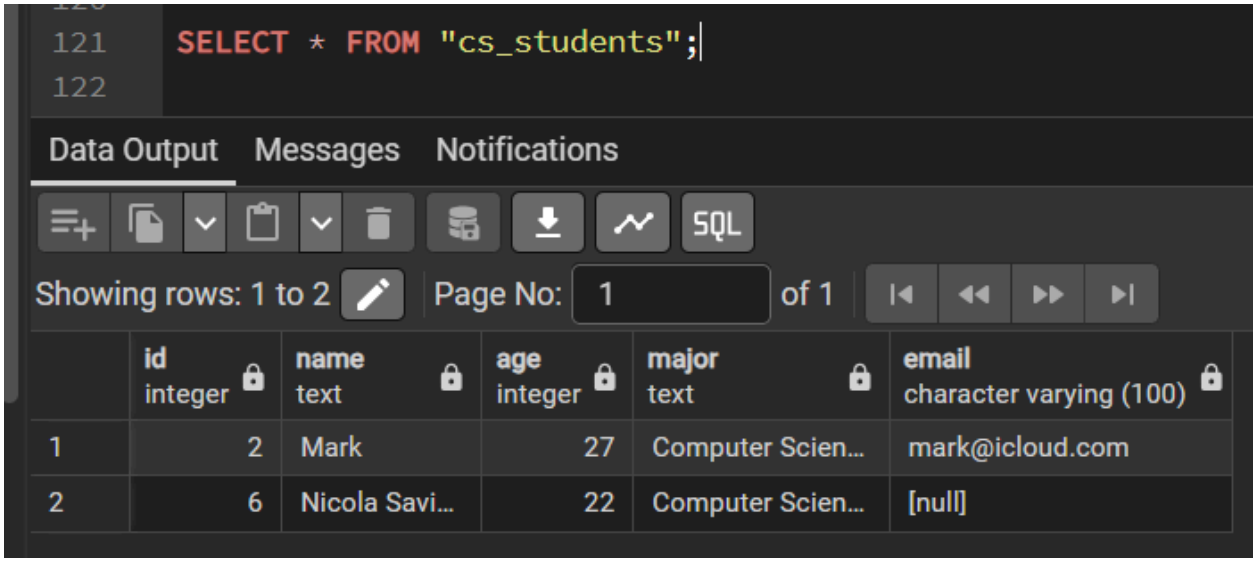
2. Add one new product in the products table.

```
INSERT INTO products (product_id, name, price)
VALUES (9001, 'Monitor', 150);
```

3. Create a view called cs students that shows only students with major = “Computer Science”.

```
CREATE VIEW "cs_students" AS
SELECT *
FROM students
WHERE major = 'Computer Science';
```

4. Query the view.


121
SELECT * FROM "cs_students";
122
Data Output
Messages
Notifications
SQL
Showing rows: 1 to 2
Page No:
1
of 1
id integer
name text
age integer
major text
email character varying (100)
1
2
Mark
27
Computer Scien...
mark@icloud.com
2
6
Nicola Savi...
22
Computer Scien...
[null]