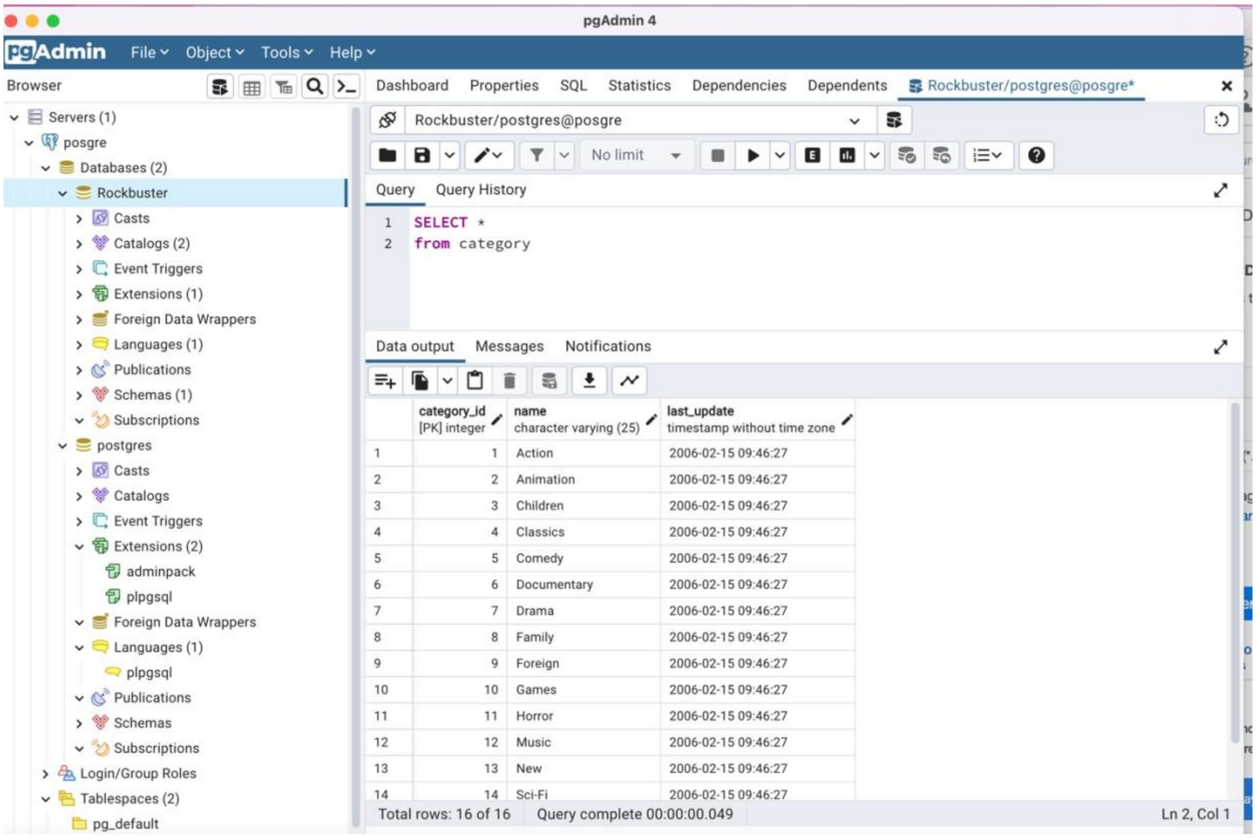
3.3: SQL for Data Analysts

Step 1:

Your first task is to find out what film genres already exist in the category table:

- Open pgAdmin 4, click the Rockbuster database, and open the Query Tool.
- Write a SELECT command to find out what film genres exist in the category table.
- Copy-paste the output into your answers document or write the answers out—it's up to you. Make sure to include the category ID for each genre.



```
SELECT *
from category
```

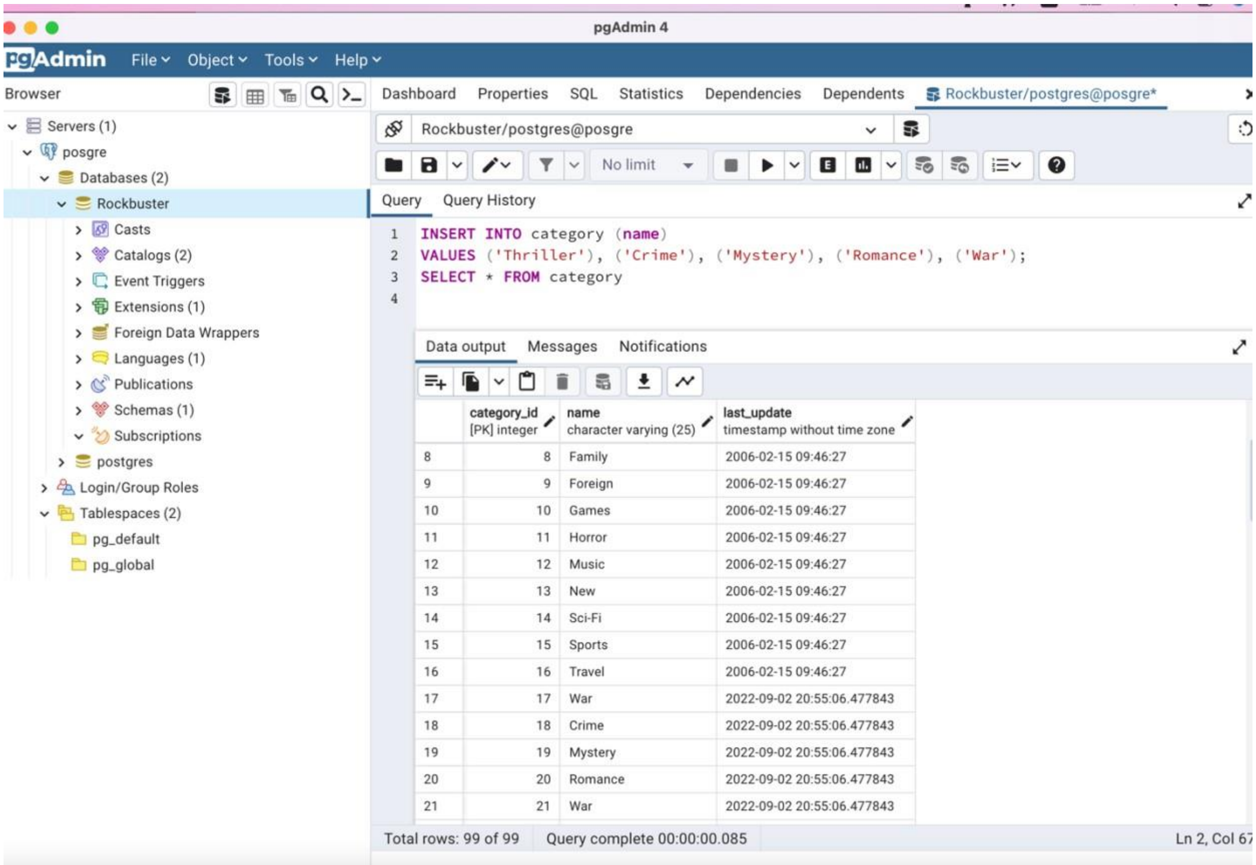
	category_id [PK] integer	name character varying (25)	last_update timestamp without time zone
1	1	Action	2006-02-15 09:46:27
2	2	Animation	2006-02-15 09:46:27
3	3	Children	2006-02-15 09:46:27
4	4	Classics	2006-02-15 09:46:27
5	5	Comedy	2006-02-15 09:46:27
6	6	Documentary	2006-02-15 09:46:27
7	7	Drama	2006-02-15 09:46:27
8	8	Family	2006-02-15 09:46:27
9	9	Foreign	2006-02-15 09:46:27
10	10	Games	2006-02-15 09:46:27
11	11	Horror	2006-02-15 09:46:27
12	12	Music	2006-02-15 09:46:27
13	13	New	2006-02-15 09:46:27
14	14	Sci-Fi	2006-02-15 09:46:27

Total rows: 16 of 16 Query complete 00:00:00.049 Ln 2, Col 1

Step 2:

You're ready to add some new genres! Write an INSERT statement to add the following genres to the category table: Thriller, Crime, Mystery, Romance, and War:

- Copy-paste your INSERT commands into your answers document.



```
INSERT INTO category (name)
VALUES ('Thriller'), ('Crime'), ('Mystery'), ('Romance'), ('War');
SELECT * FROM category
```

	category_id [PK] integer	name character varying (25)	last_update timestamp without time zone
8	8	Family	2006-02-15 09:46:27
9	9	Foreign	2006-02-15 09:46:27
10	10	Games	2006-02-15 09:46:27
11	11	Horror	2006-02-15 09:46:27
12	12	Music	2006-02-15 09:46:27
13	13	New	2006-02-15 09:46:27
14	14	Sci-Fi	2006-02-15 09:46:27
15	15	Sports	2006-02-15 09:46:27
16	16	Travel	2006-02-15 09:46:27
17	17	War	2022-09-02 20:55:06.477843
18	18	Crime	2022-09-02 20:55:06.477843
19	19	Mystery	2022-09-02 20:55:06.477843
20	20	Romance	2022-09-02 20:55:06.477843
21	21	War	2022-09-02 20:55:06.477843

Total rows: 99 of 99 Query complete 00:00:00.085 Ln 2, Col 67

- The CREATE statement below shows the constraints on the category table. Write a short paragraph explaining the various constraints that have been applied to the columns. What do these constraints do exactly? Why are they important?

```
  CREATE TABLE category
(
  category_id integer NOT NULL DEFAULT nextval('category_category_id_seq'::regclass),
name text COLLATE pg_catalog."default" NOT NULL,
  last_update timestamp with time zone NOT NULL DEFAULT now(),   CONSTRAINT
category_pkey PRIMARY KEY (category_id)
);
```

NOT NULL this ensures that the column cannot have empty or missing values
category_id : (data type is integer)Value cannot be null
name: (data type is text) Values cannot be null last_update: (data type is timestamp with time zone) value cannot be null

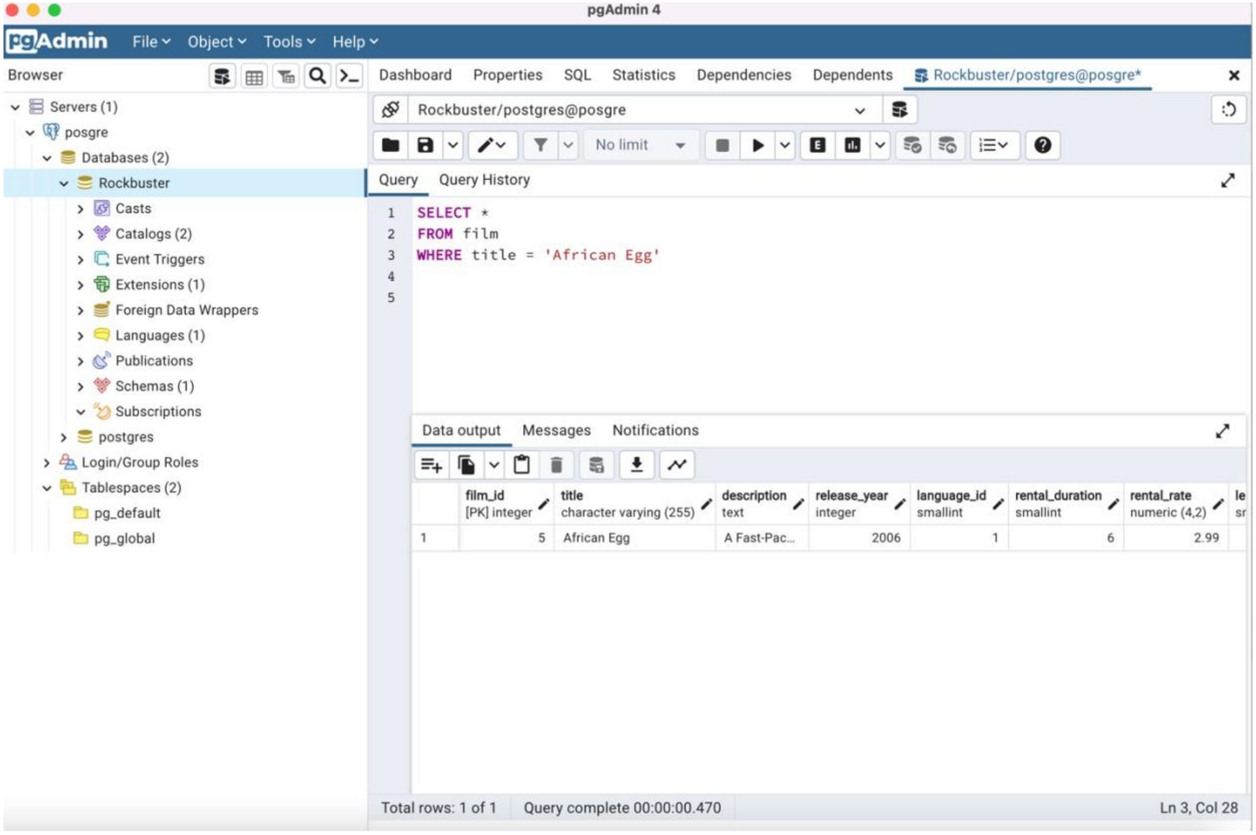
PRIMARY KEY is a unique identifier for each record in a table Category_pkey which is the category_id is set as the primary key.

Constraints are important because they can help you make sure that the values in each column are consistently formatted. They can also help you make sure values in a column are unique, not null, or even check for values that don't belong.

Step 3:

The genre for the movie *African Egg* needs to be updated to thriller. Work through the steps below to make this change:

- Write the SELECT statement to find the film_id for the movie *African Egg*.

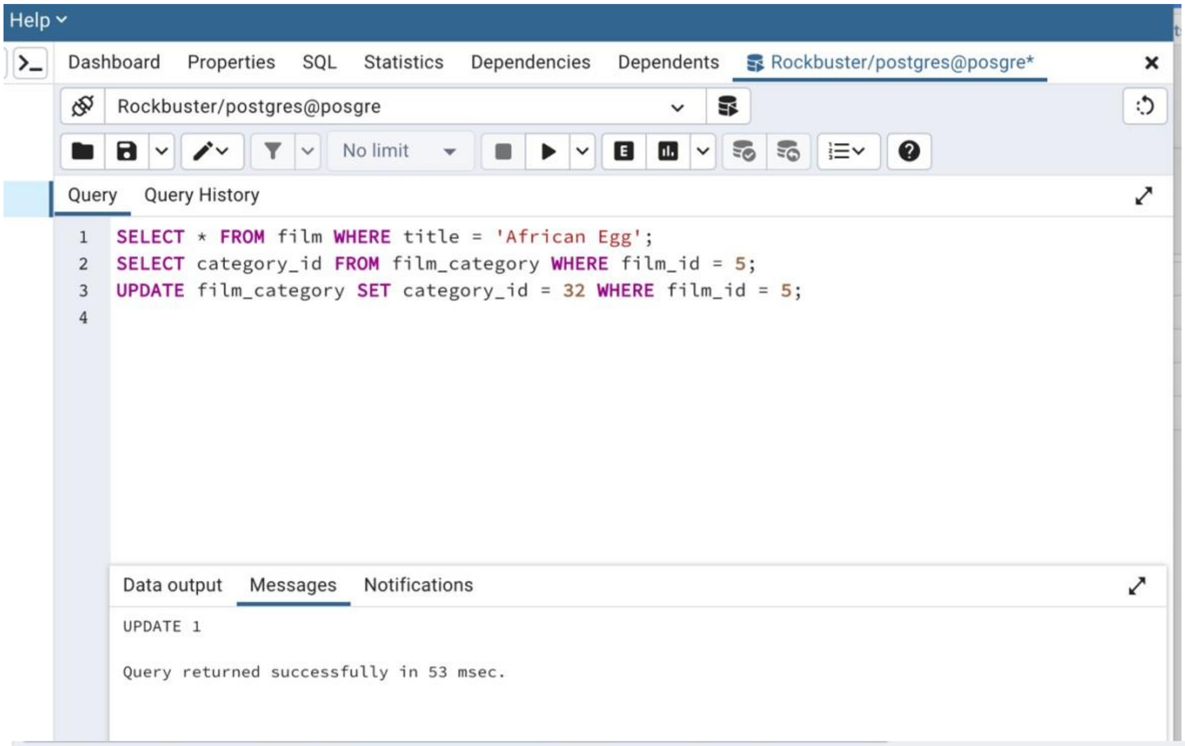


```
SELECT *
FROM film
WHERE title = 'African Egg'
```

	film_id [PK] integer	title character varying (255)	description text	release_year integer	language_id smallint	rental_duration smallint	rental_rate numeric (4,2)	le sr
1	5	African Egg	A Fast-Pac...	2006	1	6	2.99	

Total rows: 1 of 1   Query complete 00:00:00.470   Ln 3, Col 28

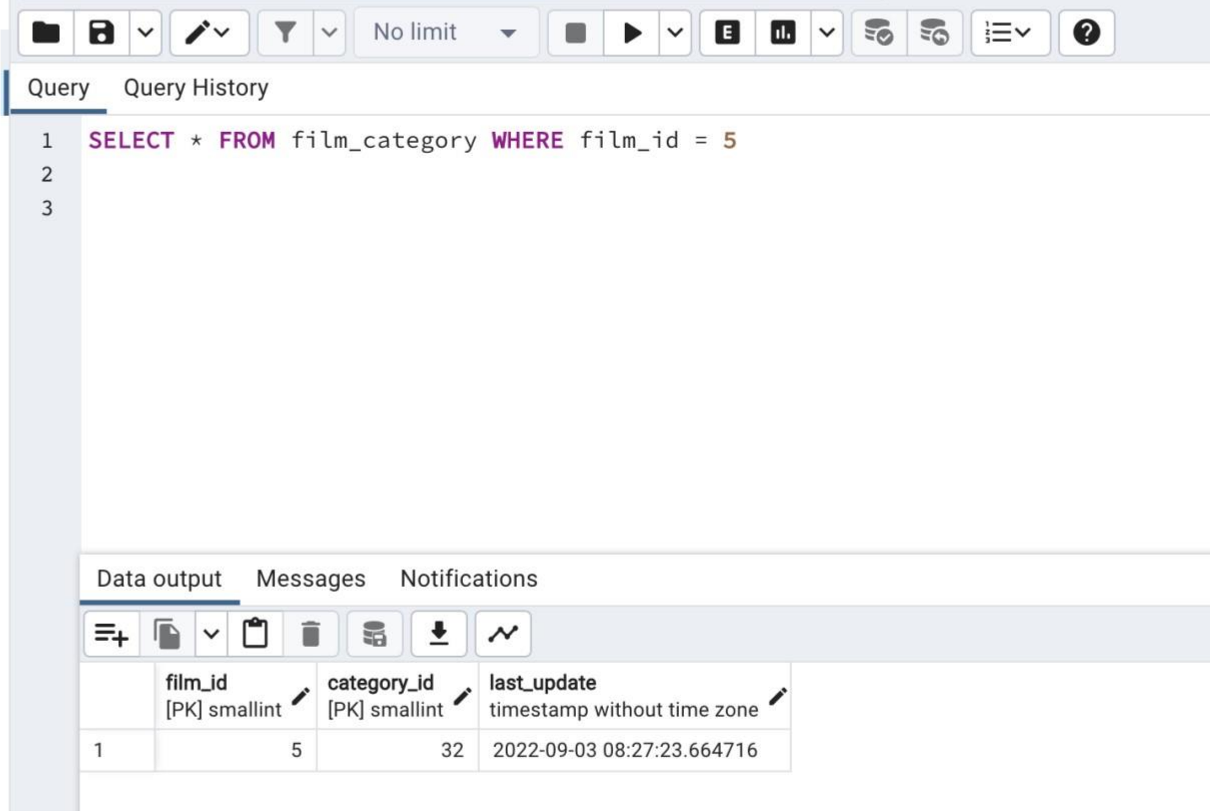
- Once you have the film_ID and category_ID, write an UPDATE command to change the category in the film_category table (not the category table). Copy-paste this command into your answers docum



```sql
SELECT * FROM film WHERE title = 'African Egg';
SELECT category_id FROM film_category WHERE film_id = 5;
UPDATE film_category SET category_id = 32 WHERE film_id = 5;
```

```
UPDATE 1

Query returned successfully in 53 msec.
```

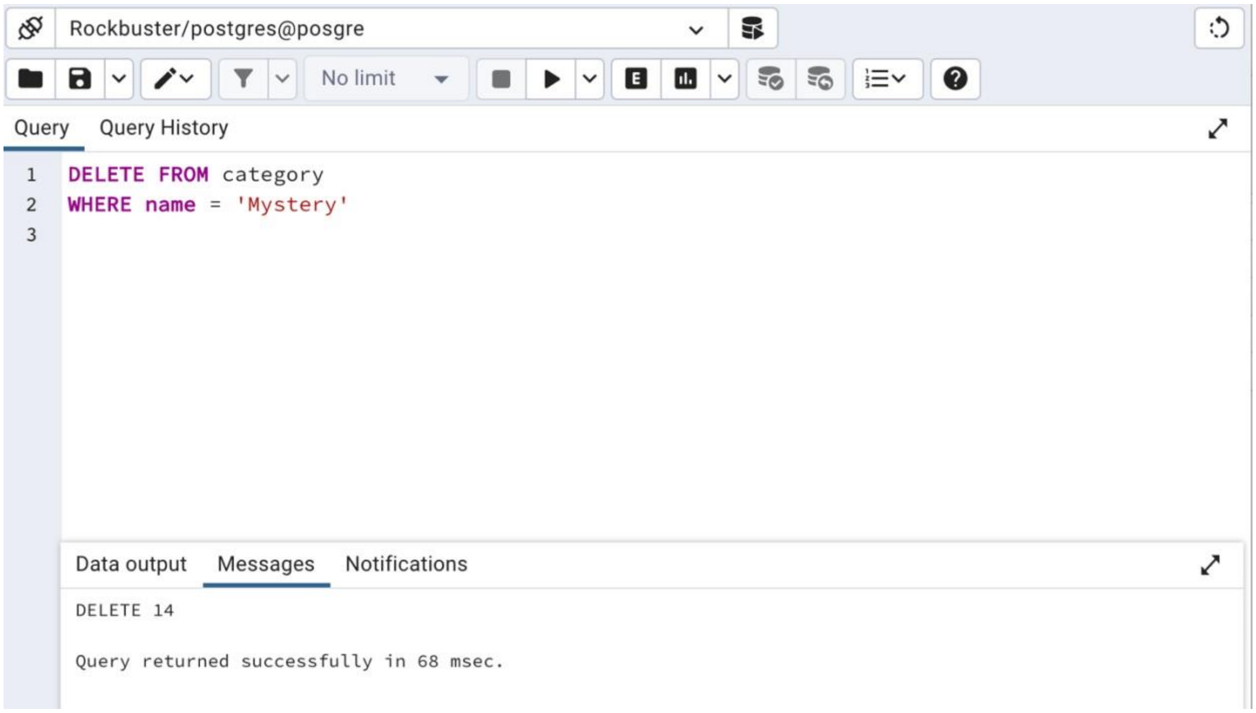


```sql
SELECT * FROM film_category WHERE film_id = 5
```

	film_id [PK] smallint	category_id [PK] smallint	last_update timestamp without time zone
1	5	32	2022-09-03 08:27:23.664716

Step 4:

Since there aren't many movies in the mystery category, you and your manager decide to remove it from the category table. Write aDELETE command to do so and copy-paste it into your answers document.



```
DELETE FROM category
WHERE name = 'Mystery'
```

```
DELETE 14

Query returned successfully in 68 msec.
```

Step 5:

Based on what you've learned so far, think about what it would be like to complete steps 1 to 4 with Excel instead of SQL. Are there any pros and cons to using SQL? Write a paragraph explaining your answer.

when using Excel, I needed to take more steps to replace data, find the desired table, use the filter, select the desired category and replace it. SQL searches for the required table automatically Some easier steps for example the updating of the category could be easily achieved with excels search and replace function

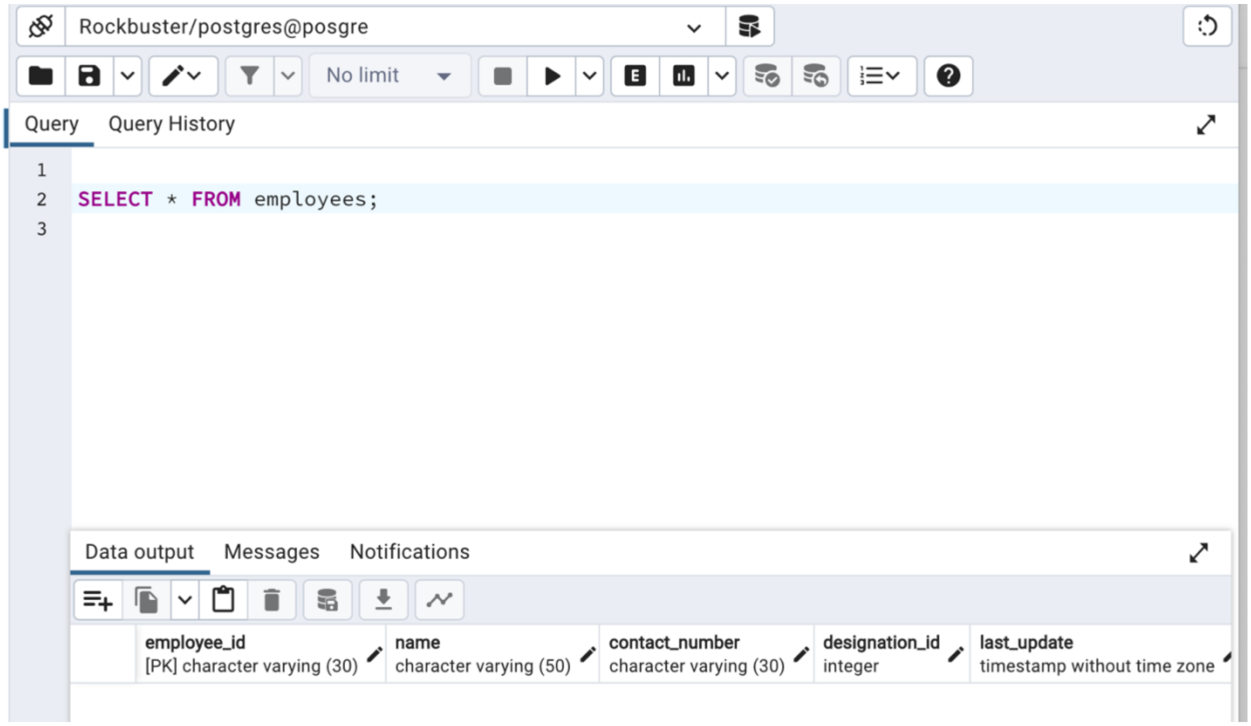
Bonus Task

The SQL query below contains some typos. See if you can fix it based on what you've learned so far about SQL and data types; then try running it in pgAdmin 4. If the query works, copy it into your Answers 3.3 document.

If you get this you're a SQL champ!

```
CREATE TBL 3EMPLOYEES
 {
employee_id VARINT(30) NOT
EMPTY name VARCHAR(50),
contact_number VARCHAR(30) ,
designation_id INT,
last_update TIMESTAMP NOT NULL DEF now()
CONSTRAIN employee_pkey PRIMARY KEY (employee_id)
}
```

```
CREATE TABLE employees
 (
employee_id VARCHAR(30) NOT NULL,
name VARCHAR(50),
contact_number VARCHAR(30),
designation_id INTEGER,
last_update TIMESTAMP NOT NULL DEFAULT now(),
CONSTRAINT employee_pkey PRIMARY KEY (employee_id)
```



```

SELECT * FROM employees;

```

```
employee_id                   name                   contact_number         designation_id   last_update
[PK] character varying (30)   character varying (50) character varying (30) integer          timestamp without time zone
```