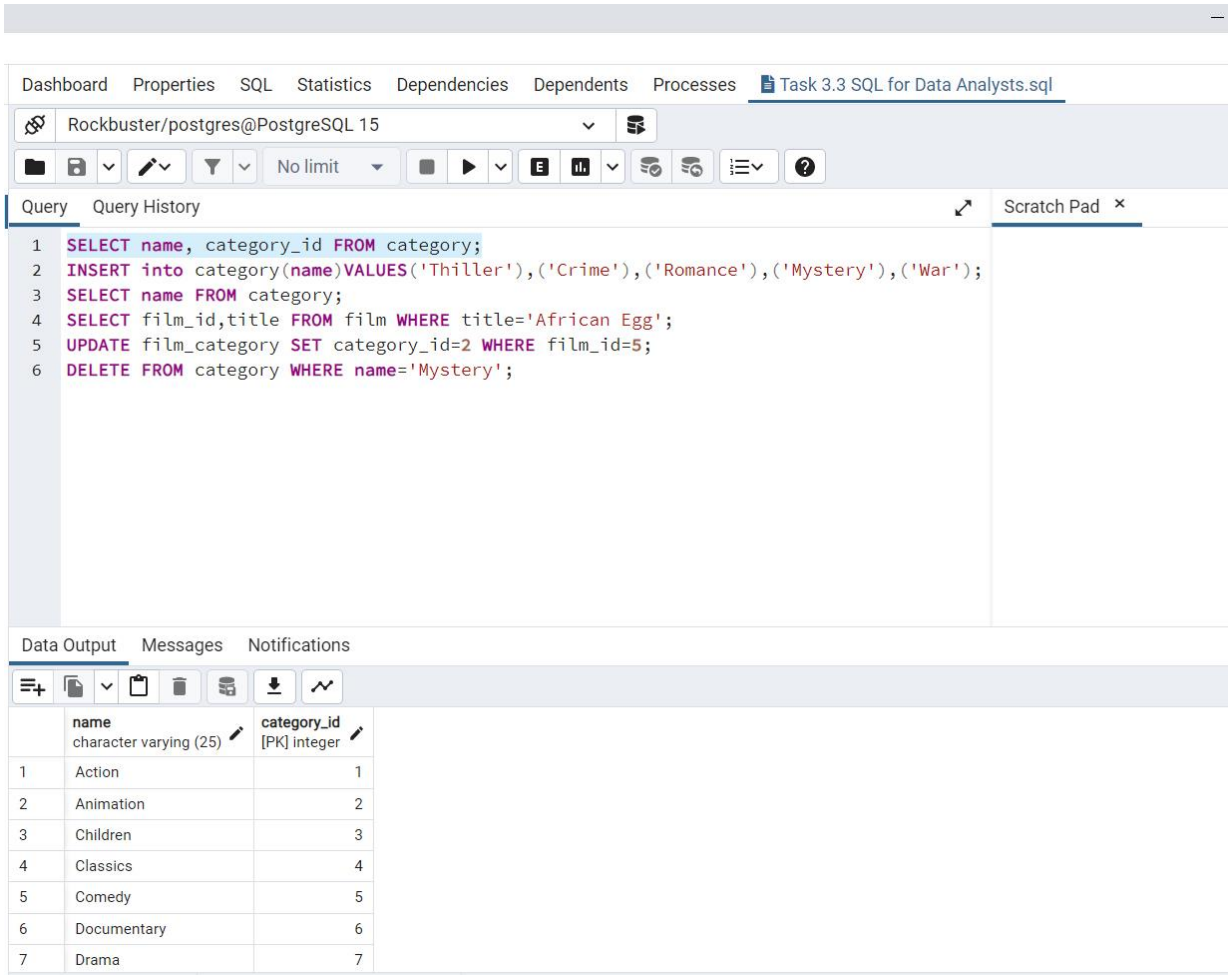
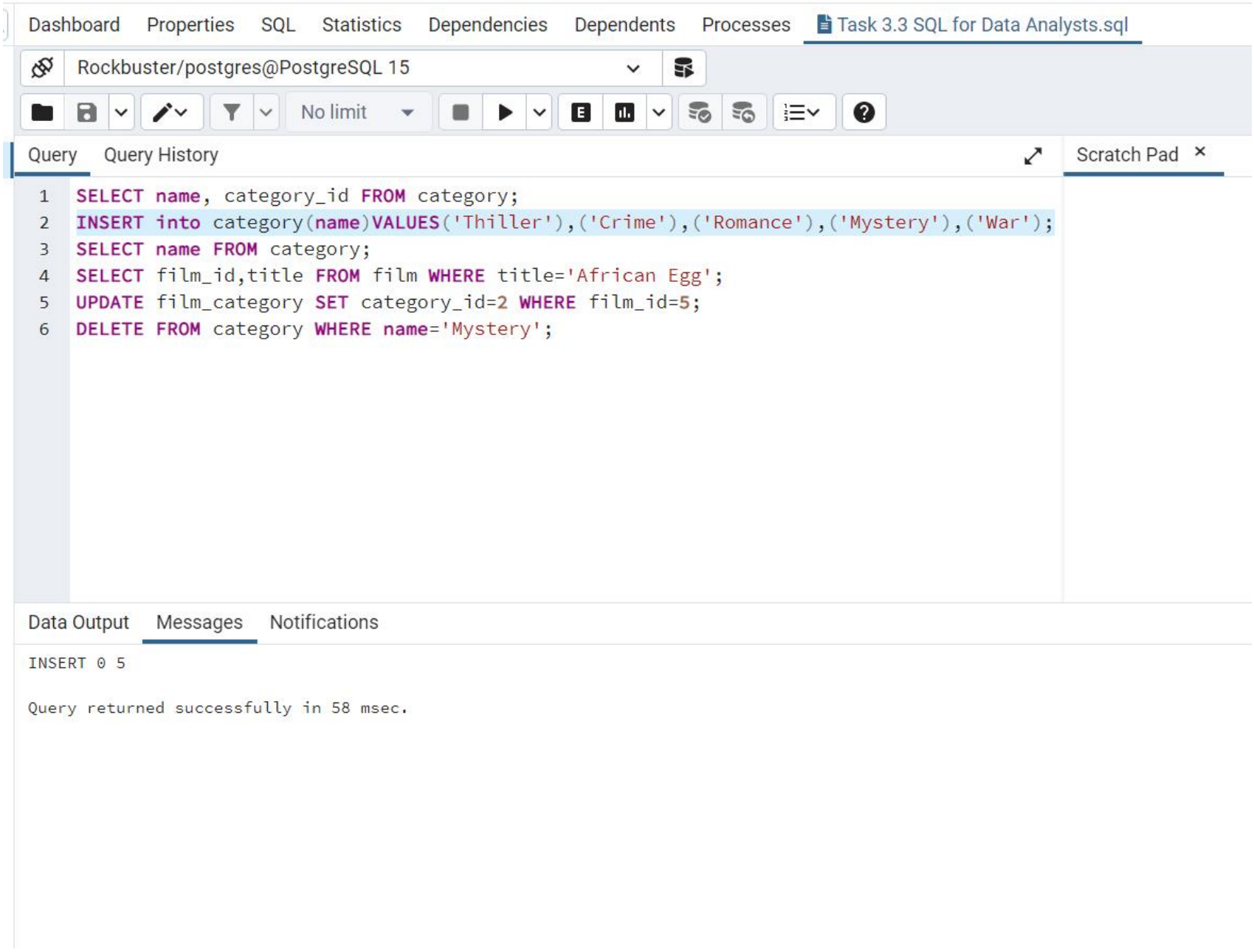
SQL for Data Analyst

Step 1



Step 2:


Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
Task 3.3 SQL for Data Analysts.sql
Rockbuster/postgres@PostgreSQL 15
No limit
Query
Query History
Scratch Pad
1 SELECT name, category_id FROM category;
2 INSERT into category(name)VALUES('Thiller'),('Crime'),('Romance'),('Mystery'),('War');
3 SELECT name FROM category;
4 SELECT film_id,title FROM film WHERE title='African Egg';
5 UPDATE film_category SET category_id=2 WHERE film_id=5;
6 DELETE FROM category WHERE name='Mystery';
Data Output
Messages
Notifications
INSERT 0 5
Query returned successfully in 58 msec.

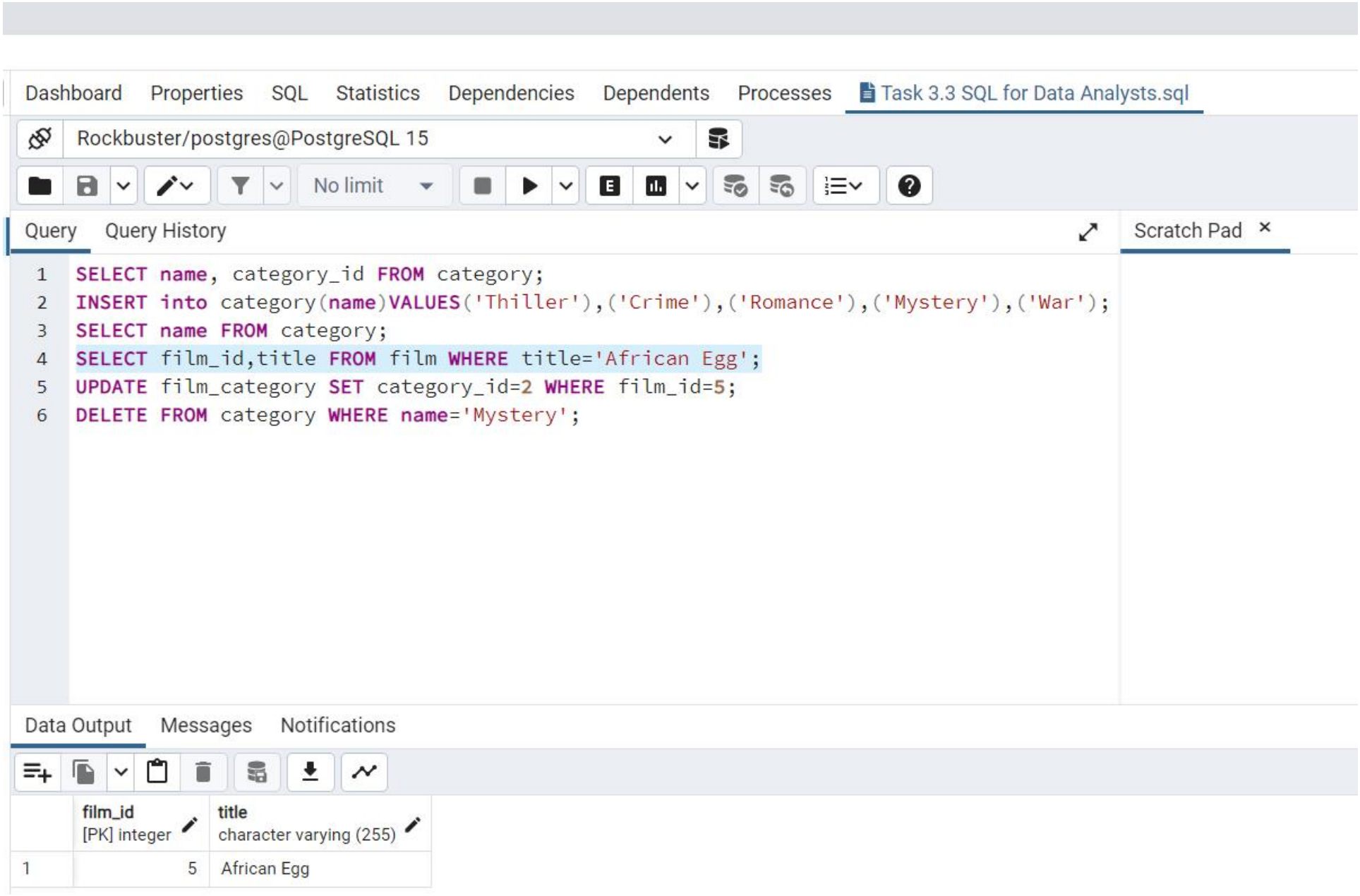
```
CREATE TABLE category
(
  category_id integer NOT NULL DEFAULT nextval('category_category_id_seq'::regclass),
  name text COLLATE pg_catalog."default" NOT NULL,
  last_update timestamp with time zone NOT NULL DEFAULT now(),
  CONSTRAINT category_pkey PRIMARY KEY (category_id)
);
```

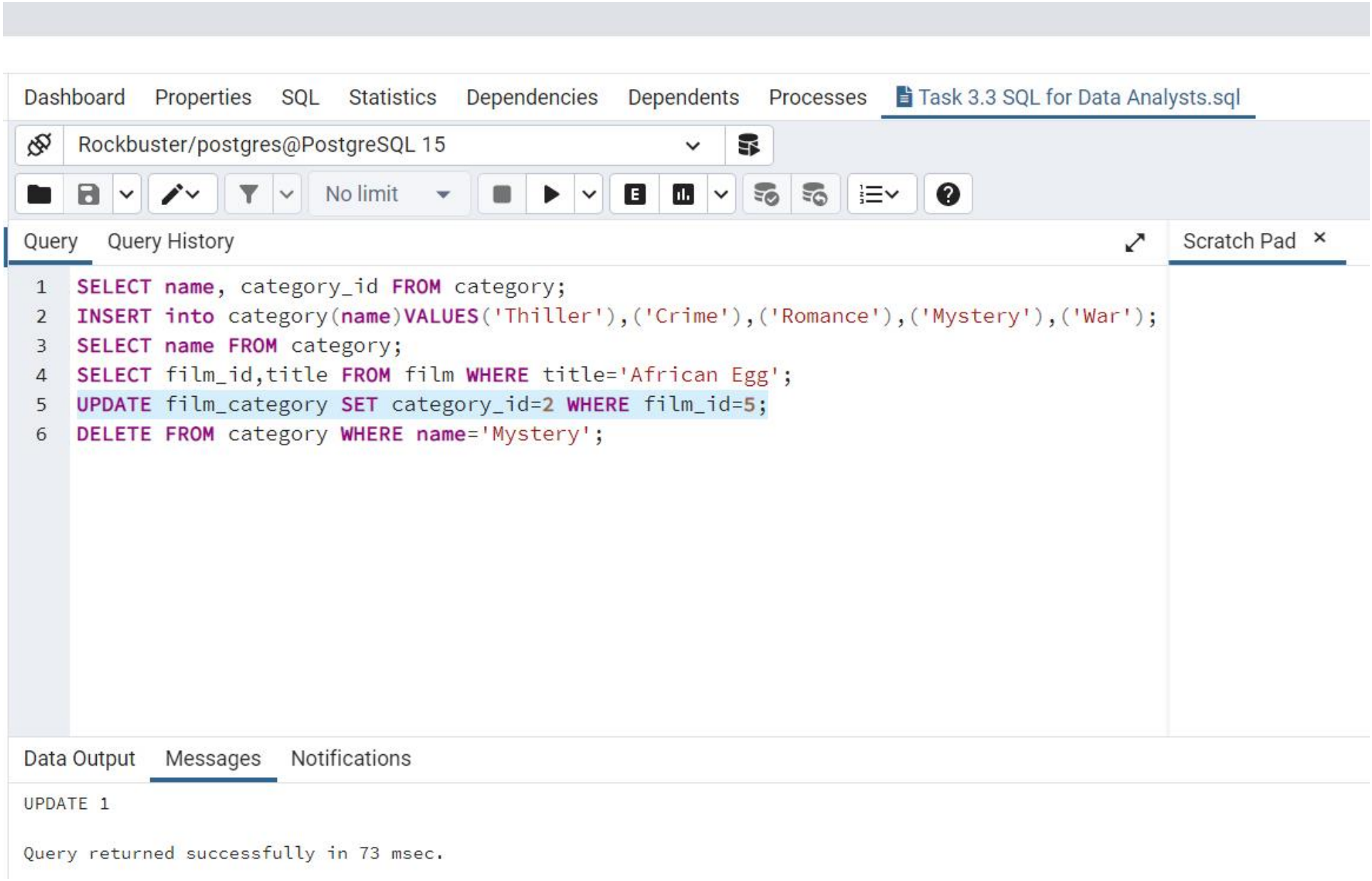
'NOT NULL' restriction prevents an empty or missing entry in the 'category_id' column.

‘PRIMARY KEY’ It makes the values in the 'category_id' column primary keys, meaning they are unique identifiers.

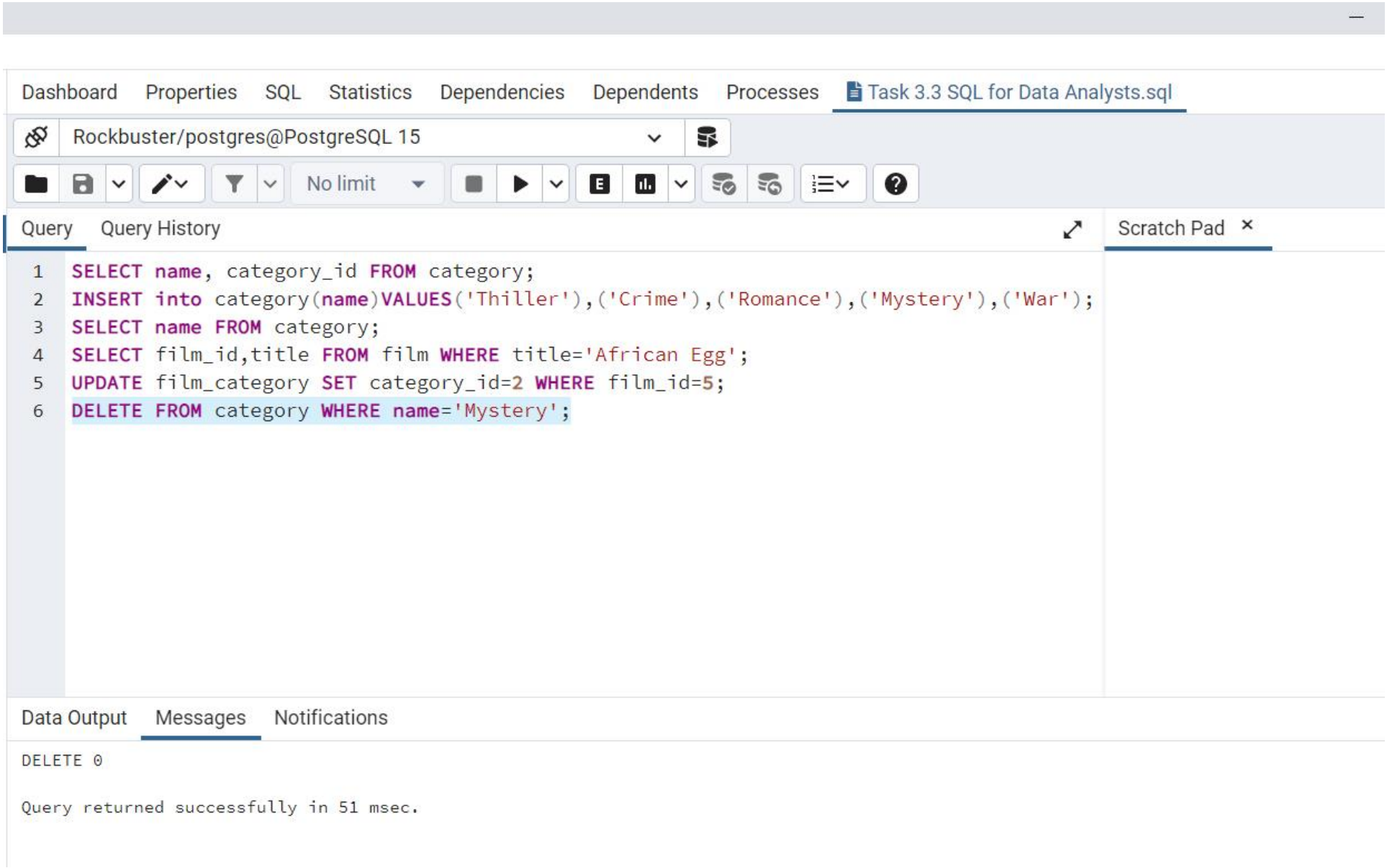
‘DEFAULT’ It ensures that there will never be a mistake by entering a default value when no other value is available.

Step 3:


Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
Task 3.3 SQL for Data Analysts.sql
Rockbuster/postgres@PostgreSQL 15
No limit
Query
Query History
Scratch Pad
1 SELECT name, category_id FROM category;
2 INSERT into category(name)VALUES('Thiller'),('Crime'),('Romance'),('Mystery'),('War');
3 SELECT name FROM category;
4 SELECT film_id,title FROM film WHERE title='African Egg';
5 UPDATE film_category SET category_id=2 WHERE film_id=5;
6 DELETE FROM category WHERE name='Mystery';
Data Output
Messages
Notifications
film_id
[PK] integer
title
character varying (255)
1
5
African Egg


Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
Task 3.3 SQL for Data Analysts.sql
Rockbuster/postgres@PostgreSQL 15
No limit
Query
Query History
Scratch Pad
1 SELECT name, category_id FROM category;
2 INSERT into category(name)VALUES('Thiller'),('Crime'),('Romance'),('Mystery'),('War');
3 SELECT name FROM category;
4 SELECT film_id,title FROM film WHERE title='African Egg';
5 UPDATE film_category SET category_id=2 WHERE film_id=5;
6 DELETE FROM category WHERE name='Mystery';
Data Output
Messages
Notifications
UPDATE 1
Query returned successfully in 73 msec.

Step 4:


Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
Task 3.3 SQL for Data Analysts.sql
Rockbuster/postgres@PostgreSQL 15
No limit
Query
Query History
Scratch Pad
1 SELECT name, category_id FROM category;
2 INSERT into category(name)VALUES('Thiller'),('Crime'),('Romance'),('Mystery'),('War');
3 SELECT name FROM category;
4 SELECT film_id,title FROM film WHERE title='African Egg';
5 UPDATE film_category SET category_id=2 WHERE film_id=5;
6 DELETE FROM category WHERE name='Mystery';
Data Output
Messages
Notifications
DELETE 0
Query returned successfully in 51 msec.

Step 5:

Still the beginning, excel is clear in terms of functions and orders with limit commands while SQL need more practice and learn a lot of commands