

Step 1:

```
SELECT category_id, name  
FROM "category"
```

The screenshot shows the pgAdmin 4 interface. On the left is the Object Explorer with a tree view of the database schema. The 'category' table is selected under the 'public' schema. The main pane displays a SQL query in the 'Query' tab: `SELECT category_id, name FROM "category"`. Below the query editor, the 'Data Output' tab shows the results of the query as a table with 16 rows. The status bar at the bottom indicates 'Total rows: 16 of 16' and 'Query complete 00:00:00.195'.

category_id	name
1	Action
2	Animation
3	Children
4	Classics
5	Comedy
6	Documentary
7	Drama
8	Family
9	Foreign
10	Games
11	Horror
12	Music
13	New
14	Sci-Fi
15	Sports
16	Travel

Step 2:

```
INSERT INTO category (name)  
VALUES ('Triller'), ('Crime'), ('Mystery'), ('Romance'), ('War')
```

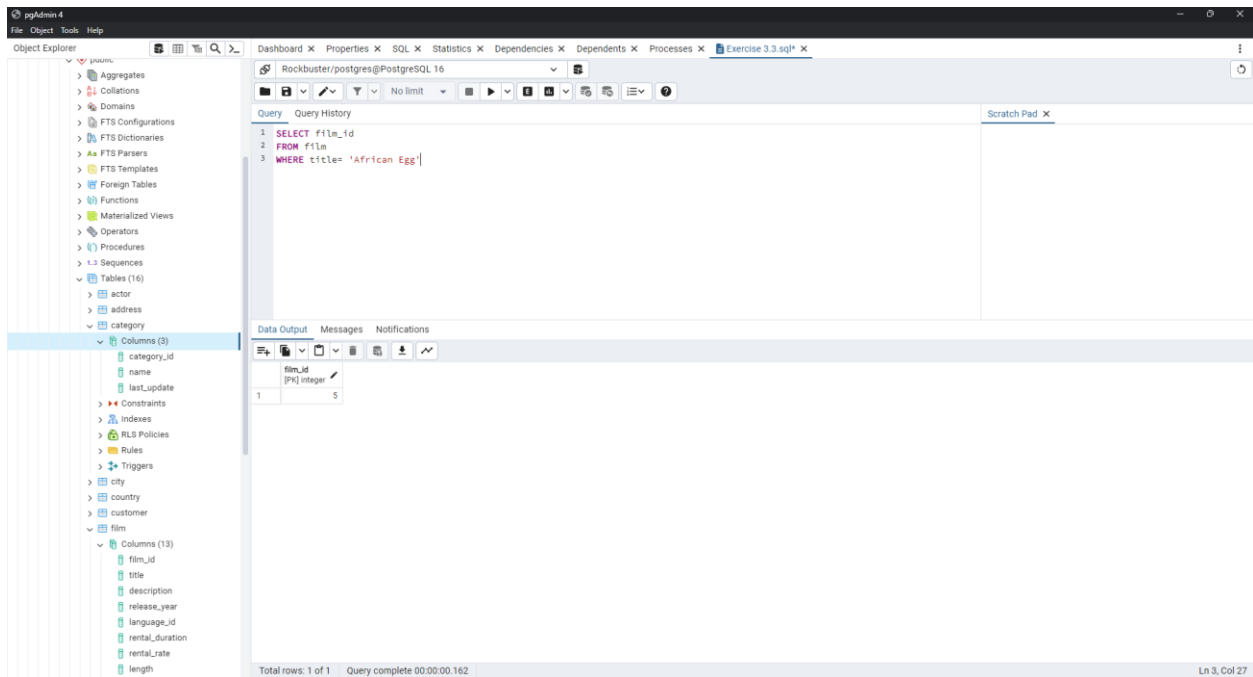
```
CREATE TABLE category  
(  
    category_id integer NOT NULL DEFAULT nextval('category_category_id_seq'::regclass),  
    name text COLLATE pg_catalog."default" NOT NULL,  
    last_update timestamp with time zone NOT NULL DEFAULT now(),  
    CONSTRAINT category_pkey PRIMARY KEY (category_id)  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

category_id – cannot be empty (NOT NULL), this constraint ensures there are no value missing from any column. That is important to keep the database accurate and not allow empty records.

Name – represented in text values according to specific to configuration file and CONSTRAINT NOT NULL added to ensure there are no empty record.

Last_update – in this CONSTRAINT NOT NULL is utilized to make sure the record is not empty, and the DEFAULT now automatically fills the record.

Step 3:



The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer displays the database structure, including tables like 'actor', 'address', 'category', and 'film'. The 'category' table is selected, showing its columns: category_id, name, and last_update. The main pane displays a SQL query:

```
1 SELECT film_id
2 FROM film
3 WHERE title = 'African Eggs'
```

The Data Output pane shows the results of the query:

film_id	film_id [PK] Integer
1	5

The status bar at the bottom indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.162'.

pgAdmin 4

File Object Tools Help

Object Explorer

- Cast
- Catalogs
- Event Triggers
- Extensions
- Foreign Data Wrappers
- Languages
- Publications
- Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (16)
 - actor
 - address
 - category
 - Columns (3)
 - category_id
 - name
 - last_update
 - city
 - country
 - customer
 - film
 - Columns (13)

Dashboard Properties SQL Statistics Dependencies Dependents Processes Rockbuster/postgres@PostgreSQL 16*

Query Query History Scratch Pad

```
1 SELECT film_id, category_id
2 FROM film_category
3 WHERE film_id=5
```

Data Output Messages Notifications

film_id	category_id
[PK] smallint	[PK] smallint
1	5

Total rows: 1 of 1 Query complete 00:00:00.071 Ln 1, Col 12

pgAdmin 4

File Object Tools Help

Object Explorer

- FTS Dictionaries
- FTS Parsers
- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (16)
 - actor
 - address
 - category
 - Columns (3)
 - category_id
 - name
 - last_update
 - city
 - country
 - customer
 - film
 - Columns (13)
 - film_id
 - title
 - description
 - release_year
 - language_id
 - rental_duration
 - rental_rate
 - length
 - replacement_cost
 - rating
 - last_update
 - special_features

Dashboard Properties SQL Statistics Dependencies Dependents Processes Rockbuster/postgres@PostgreSQL 16*

Query Query History Scratch Pad

```
1 SELECT category_id
2 FROM category
3 WHERE name='Thriller'
```

Data Output Messages Notifications

category_id
[PK] integer
1

Total rows: 1 of 1 Query complete 00:00:00.075 Ln 3, Col 22

pgAdmin 4

File Object Tools Help

Object Explorer

- FTS Dictionaries
- FTS Parsers
- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (16)
 - actor
 - address
 - category
 - Columns (3)
 - category_id
 - name
 - last_update
 - city
 - country
 - customer
 - film
 - Columns (13)
 - film_id
 - title
 - description
 - release_year
 - language_id
 - rental_duration
 - rental_rate
 - length
 - replacement_cost
 - rating
 - last_update
 - special_features

- Constraints
- Indexes
- RLS Policies
- Rules
- Triggers

Dashboard Properties SQL Statistics Dependencies Dependents Processes Rockbuster/postgres@PostgreSQL 16*

Query Query History Scratch Pad

```
1 UPDATE film_category
2 SET category_id = 17
3 Where film_id=5
```

Data Output Messages Notifications

UPDATE 1

Query returned successfully in 83 msec.

Total rows: 1 of 1 Query complete 00:00:00.083 Ln 3, Col 16

pgAdmin 4

File Object Tools Help

Object Explorer

- FTS Dictionaries
- FTS Parsers
- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (16)
 - actor
 - address
 - category
 - Columns (3)
 - category_id
 - name
 - last_update
 - city
 - country
 - customer
 - film
 - Columns (13)
 - film_id
 - title
 - description
 - release_year
 - language_id
 - rental_duration
 - rental_rate
 - length
 - replacement_cost
 - rating
 - last_update
 - special_features
- Constraints
- Indexes
- RLS Policies
- Rules
- Triggers

Dashboard Properties SQL Statistics Dependencies Dependents Processes Rockbuster/postgres@PostgreSQL 16*

Query Query History Scratch Pad

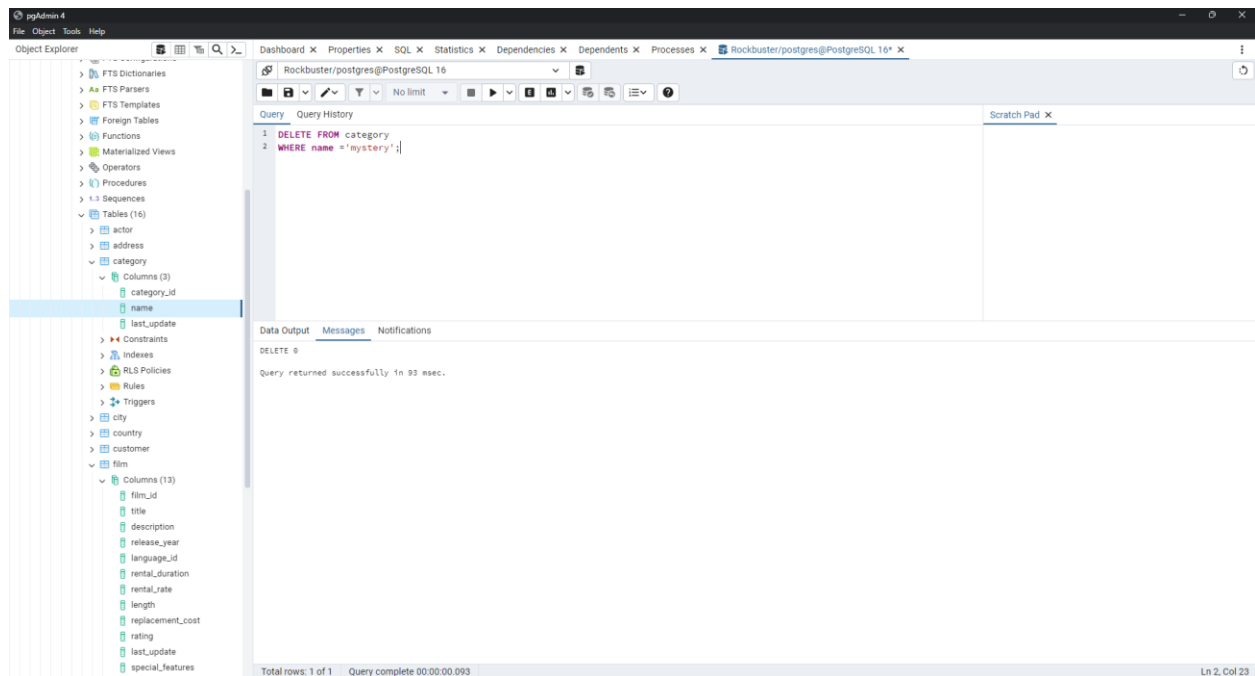
```
1 SELECT film_id, category_id
2 FROM film_category
3 WHERE film_id=5
```

Data Output Messages Notifications

film_id	category_id
1	5

Total rows: 1 of 1 Query complete 00:00:00.100 Ln 3, Col 16

Step 4:



Step 5:

While Excel is user-friendly, it does struggle with scalability and data integrity. SQL is much safer for data storage than it is Excel, but SQL does require a bit more complex knowledge whereas Excel offers ease of use and accessibility for simple task. With more practice and expertise, SQL provides much superior efficiency, scalability, and data safety. Updating data will require more steps in Excel than SQL which can be time consuming.

BONUS TASK

pgAdmin 4

File Object Tools Help

Dashboard x Properties x SQL x Statistics x Dependencies x Dependents x Processes x Rockbuster/postgres@PostgreSQL 16 x

Object Explorer

- Servers
 - PostgreSQL 16
 - Databases (2)
 - Rockbuster
 - Castes
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (16)
 - actor
 - address
 - category
 - Columns (3)
 - category_id
 - name
 - lastUpdate
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers

Query Query History

```
1 CREATE TABLE employees
2 (employee_id SMALLINT NOT NULL,
3  name VARCHAR(50),
4  contact_number VARCHAR(30),
5  designation_id INT,
6  last_update timestamp with time zone NOT Null DEFAULT now(),
7  CONSTRAINT employee_pkey PRIMARY KEY (employee_id)
8 );
```

Scratch Pad x

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 60 msec.

Total rows: 1 of 1 Query complete 00:00:00.060

Ln 8, Col 3