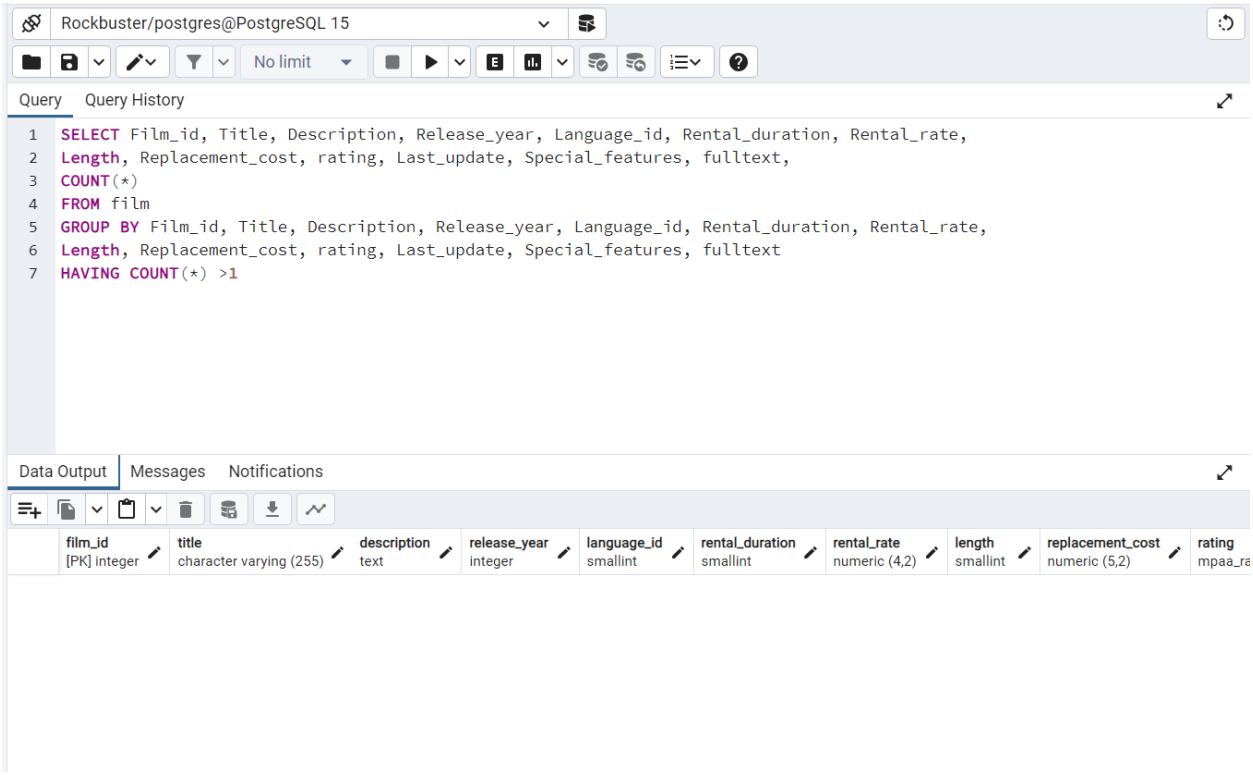
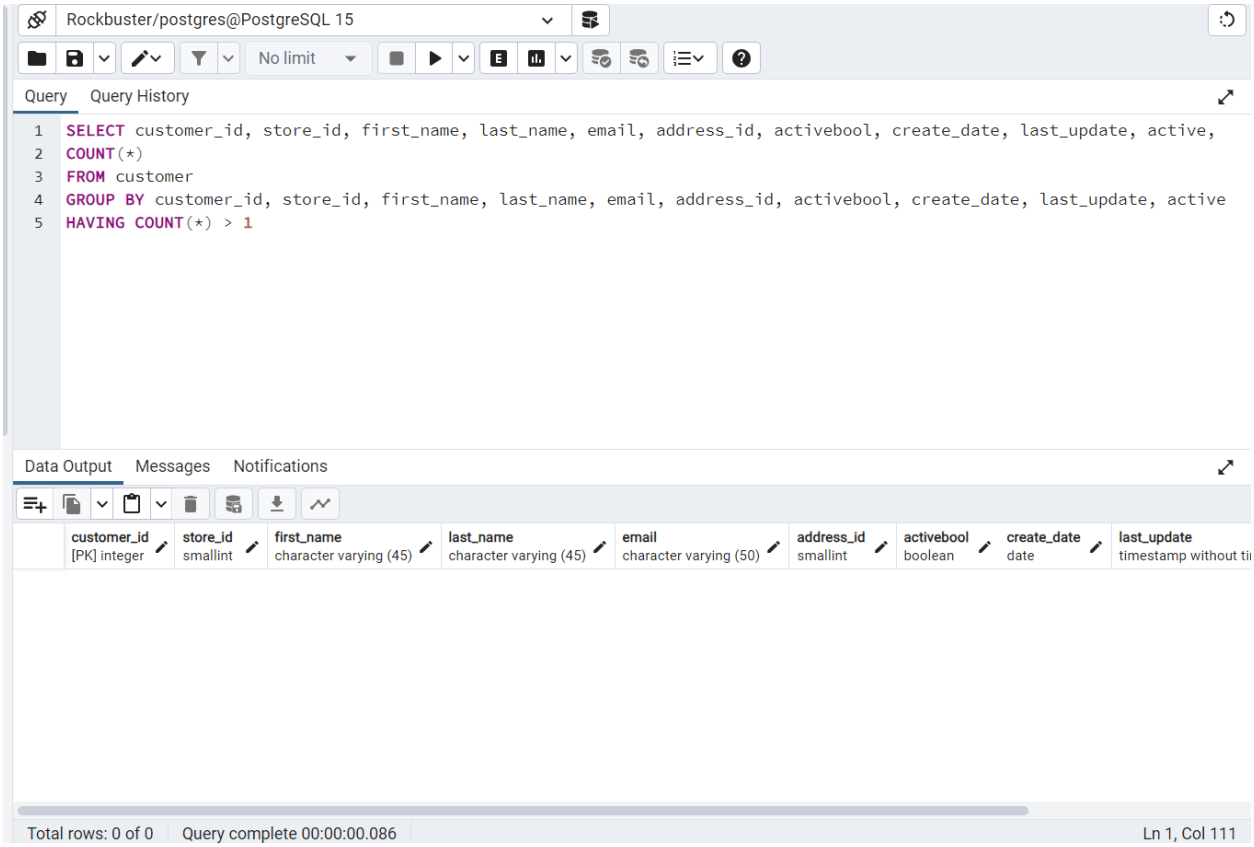
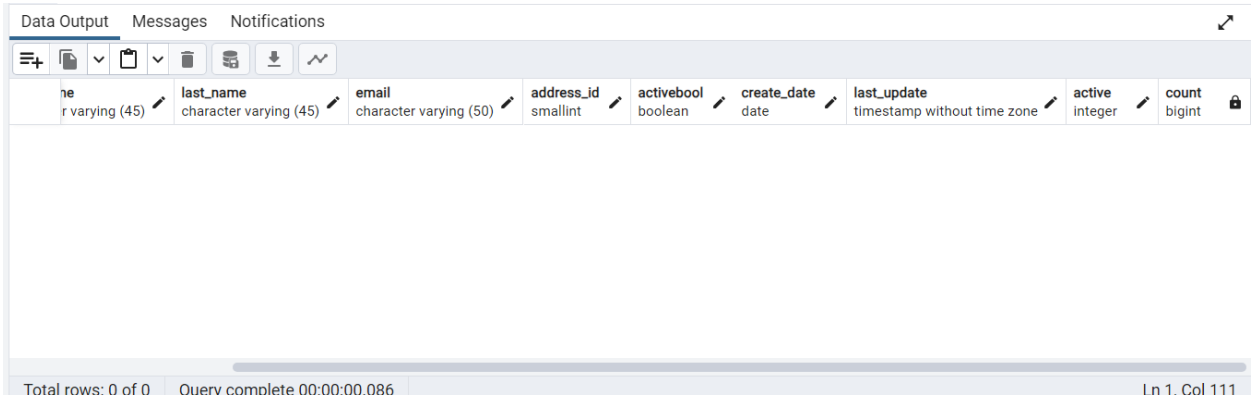
1.a Film Table



1.a Customer Table




There are no duplicate records in either of the tables. Maintaining a viewable table with distinct records is deemed as a preferred approach. This is due to the fact that deleting records may lead to the loss of valuable information.

## 1.b Film Table

Rockbuster/postgres@PostgreSQL 15

No limit

Query | Query History

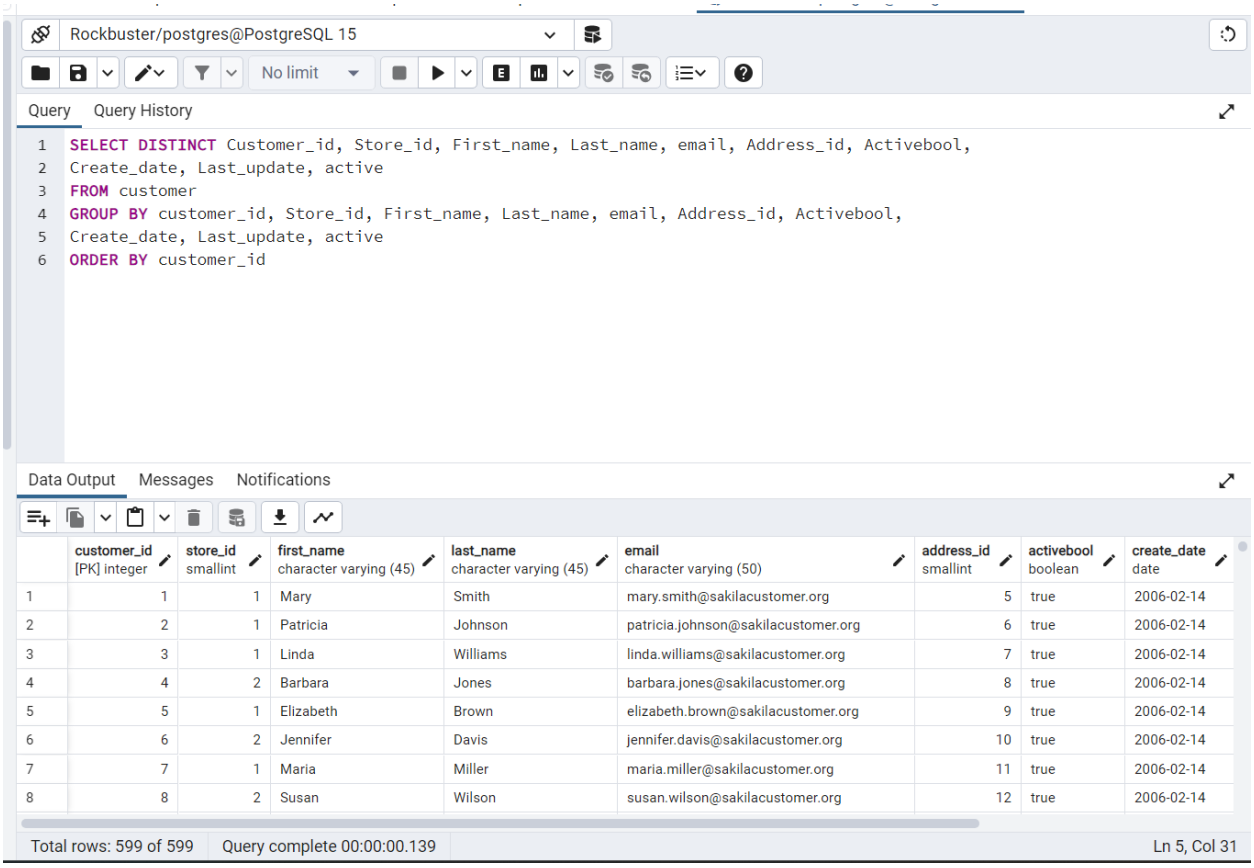
```
1  SELECT DISTINCT Film_id, Title, Description, Release_year, Language_id, Rental_duration,
2  Rental_rate, Length, Replacement_cost, rating, Last_update, Special_features, fulltext
3  FROM film
4  GROUP BY Film_id, Title, Description, Release_year, Language_id, Rental_duration, Rental_rate,
5  Length, Replacement_cost, rating, Last_update, Special_features, fulltext
6  ORDER BY film_id
```

Data Output | Messages | Notifications

| | film_id [PK] integer | title character varying (255) | description text |
|---|---|---|---|
| 1 | 1 | Academy Dinosaur | A Epic Drama of a Feminist And a Mad Scientist who must Battle a Teacher in The Canadian Rockies |
| 2 | 2 | Ace Goldfinger | A Astounding Epistle of a Database Administrator And a Explorer who must Find a Car in Ancient China |
| 3 | 3 | Adaptation Holes | A Astounding Reflection of a Lumberjack And a Car who must Sink a Lumberjack in A Baloon Factory |
| 4 | 4 | Affair Prejudice | A Fanciful Documentary of a Frisbee And a Lumberjack who must Chase a Monkey in A Shark Tank |
| 5 | 5 | African Egg | A Fast-Paced Documentary of a Pastry Chef And a Dentist who must Pursue a Forensic Psychologist in The Gulf of Mexico |
| 6 | 6 | Agent Truman | A Intrepid Panorama of a Robot And a Boy who must Escape a Sumo Wrestler in Ancient China |
| 7 | 7 | Airplane Sierra | A Touching Saga of a Hunter And a Butler who must Discover a Butler in A Jet Boat |
| 8 | 8 | Airport Pollock | A Epic Tale of a Moose And a Girl who must Confront a Monkey in Ancient India |

Total rows: 1000 of 1000 | Query complete 00:00:00.139 | Ln 5, Col 31

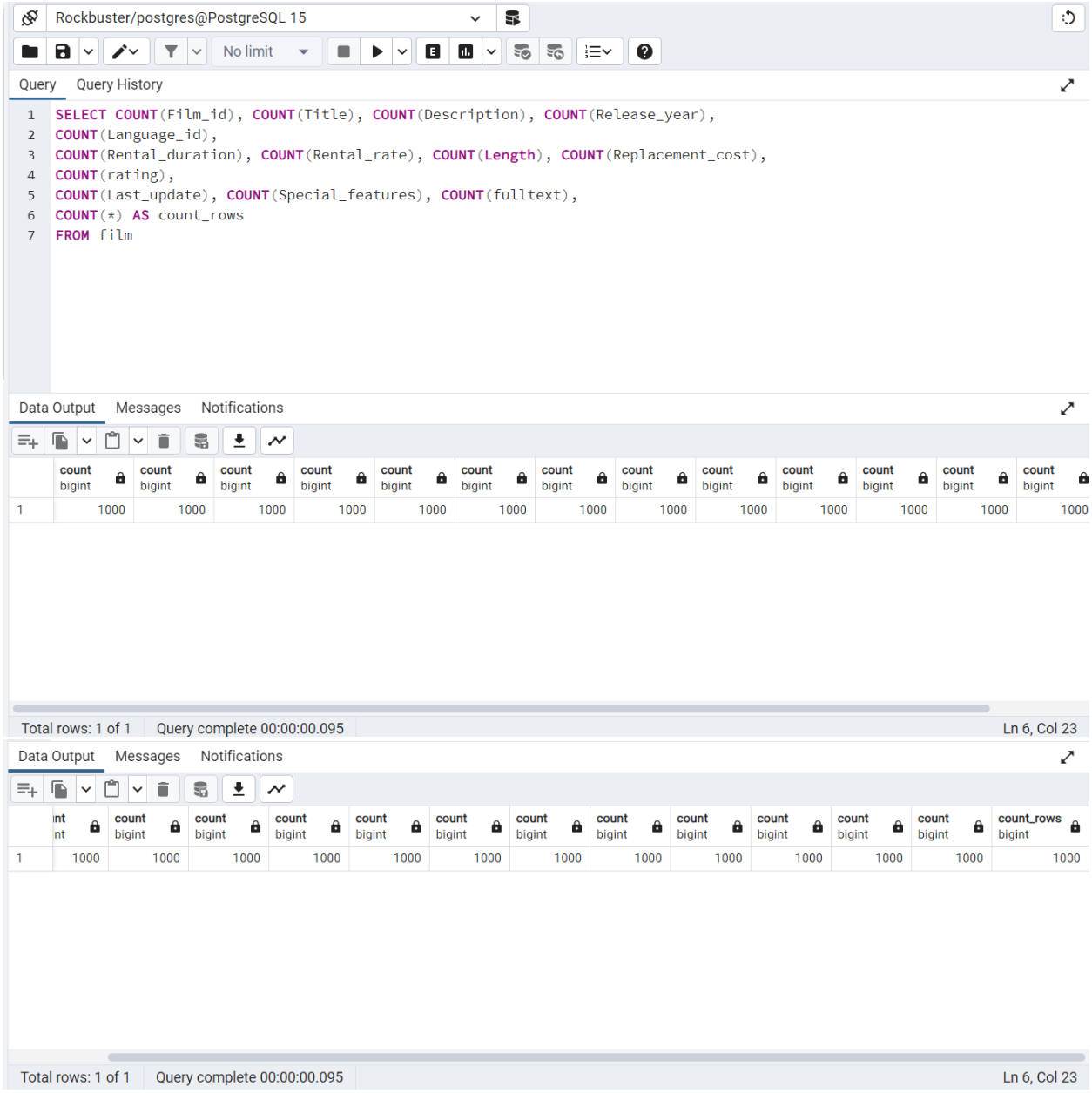
1.b Customer Table



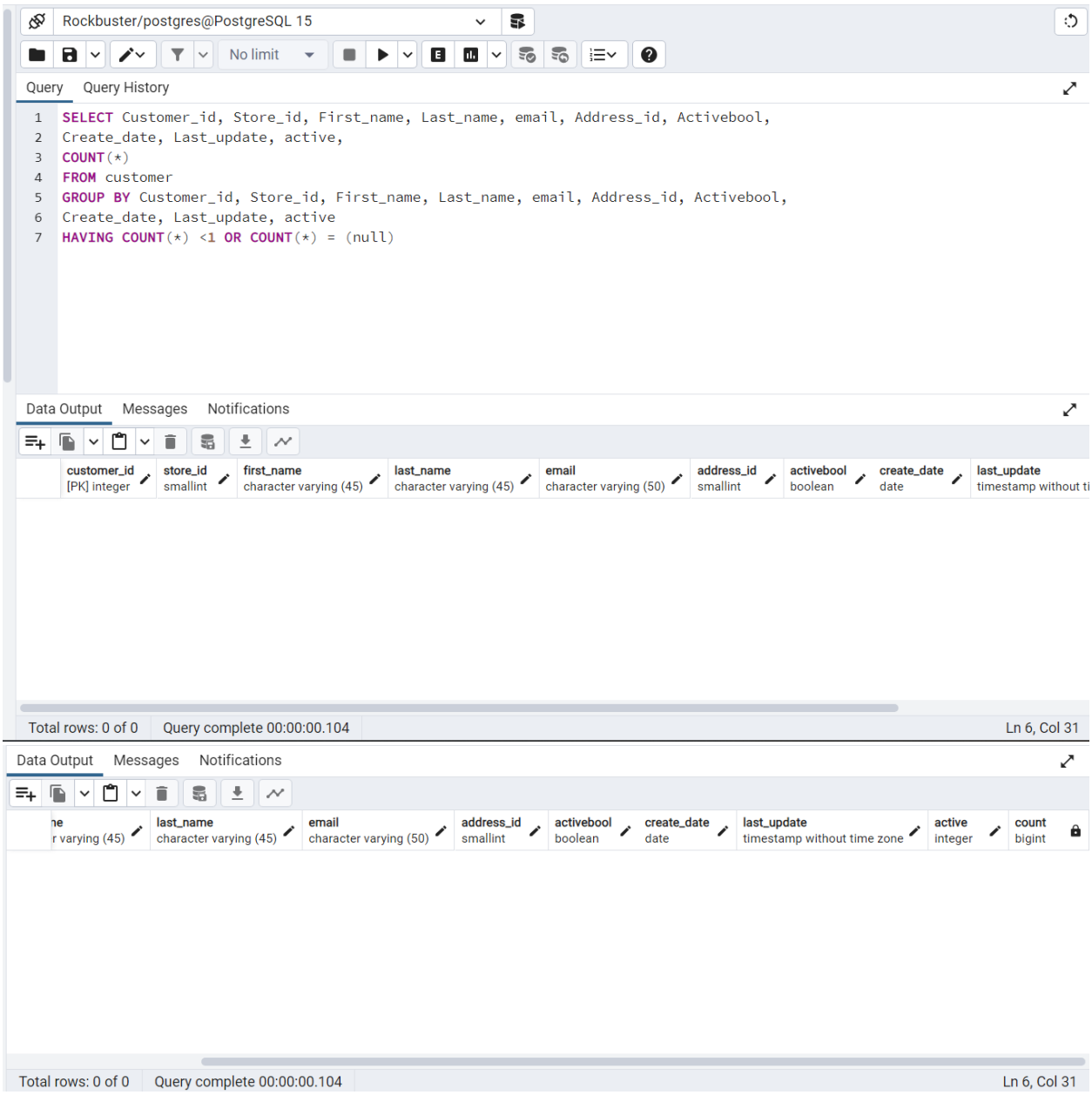

| | customer_id [PK] integer | store_id smallint | first_name character varying (45) | last_name character varying (45) | email character varying (50) | address_id smallint | activebool boolean | create_date date |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | Mary | Smith | mary.smith@sakilacustomer.org | 5 | true | 2006-02-14 |
| 2 | 2 | 1 | Patricia | Johnson | patricia.johnson@sakilacustomer.org | 6 | true | 2006-02-14 |
| 3 | 3 | 1 | Linda | Williams | linda.williams@sakilacustomer.org | 7 | true | 2006-02-14 |
| 4 | 4 | 2 | Barbara | Jones | barbara.jones@sakilacustomer.org | 8 | true | 2006-02-14 |
| 5 | 5 | 1 | Elizabeth | Brown | elizabeth.brown@sakilacustomer.org | 9 | true | 2006-02-14 |
| 6 | 6 | 2 | Jennifer | Davis | jennifer.davis@sakilacustomer.org | 10 | true | 2006-02-14 |
| 7 | 7 | 1 | Maria | Miller | maria.miller@sakilacustomer.org | 11 | true | 2006-02-14 |
| 8 | 8 | 2 | Susan | Wilson | susan.wilson@sakilacustomer.org | 12 | true | 2006-02-14 |

The data appears uniform for both tables. If the data was not uniform, I would use the Update function along with the Where and Set functions to resolve this problem.

1.c Film table

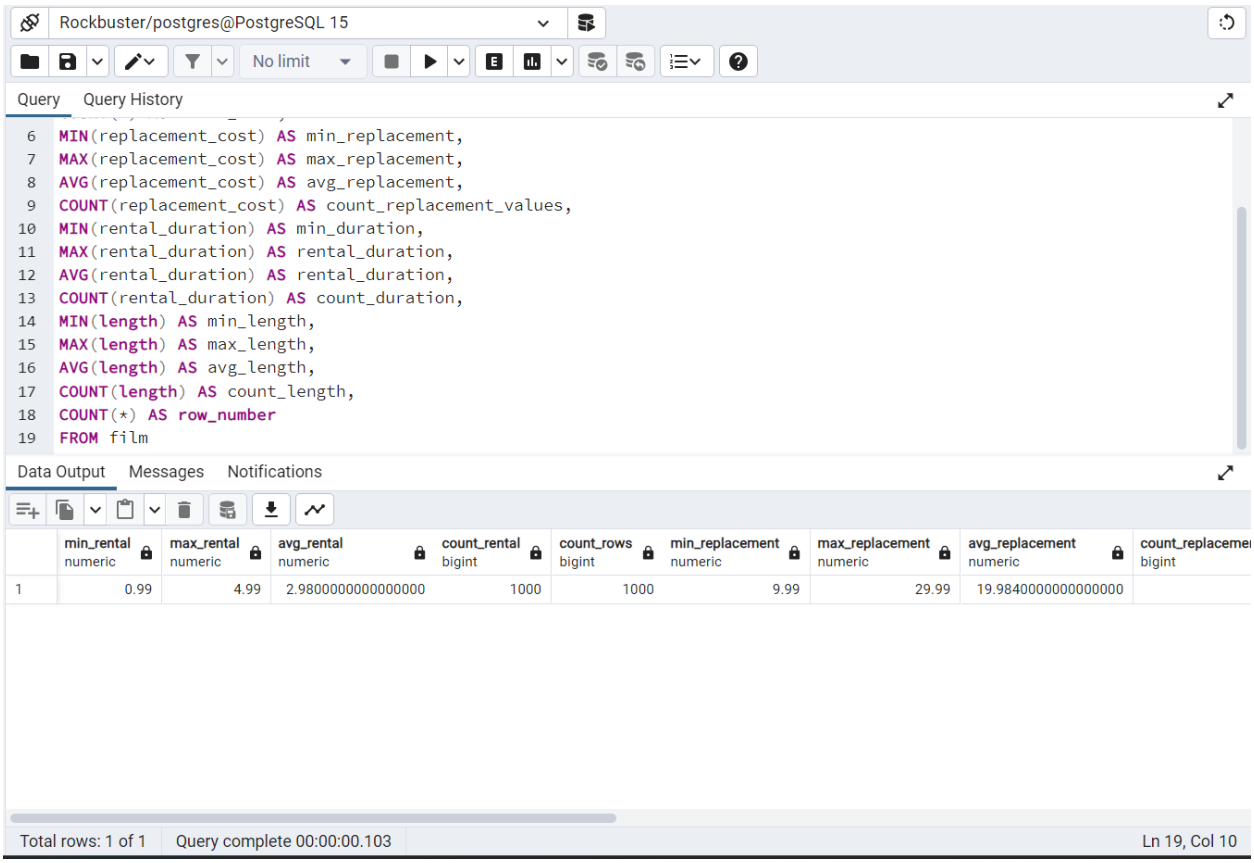


1.c Customer Table



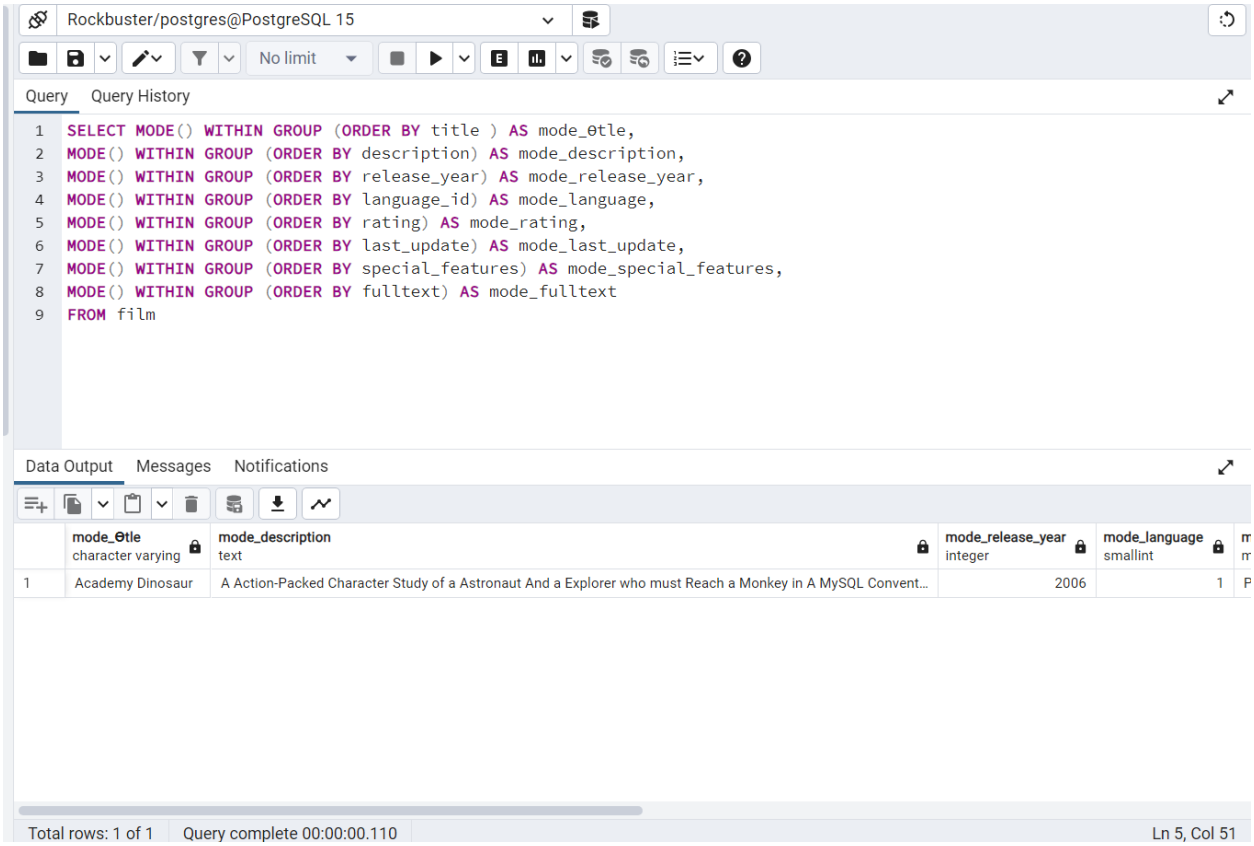

There appears to be no missing data. If there were missing data you could use the Update, Where, Set functions could be used to impute the average value for a small amount of missing data. However, you can also disregard a column  or row that has a significant proportion of missing data.

2.a Film Table Numerical



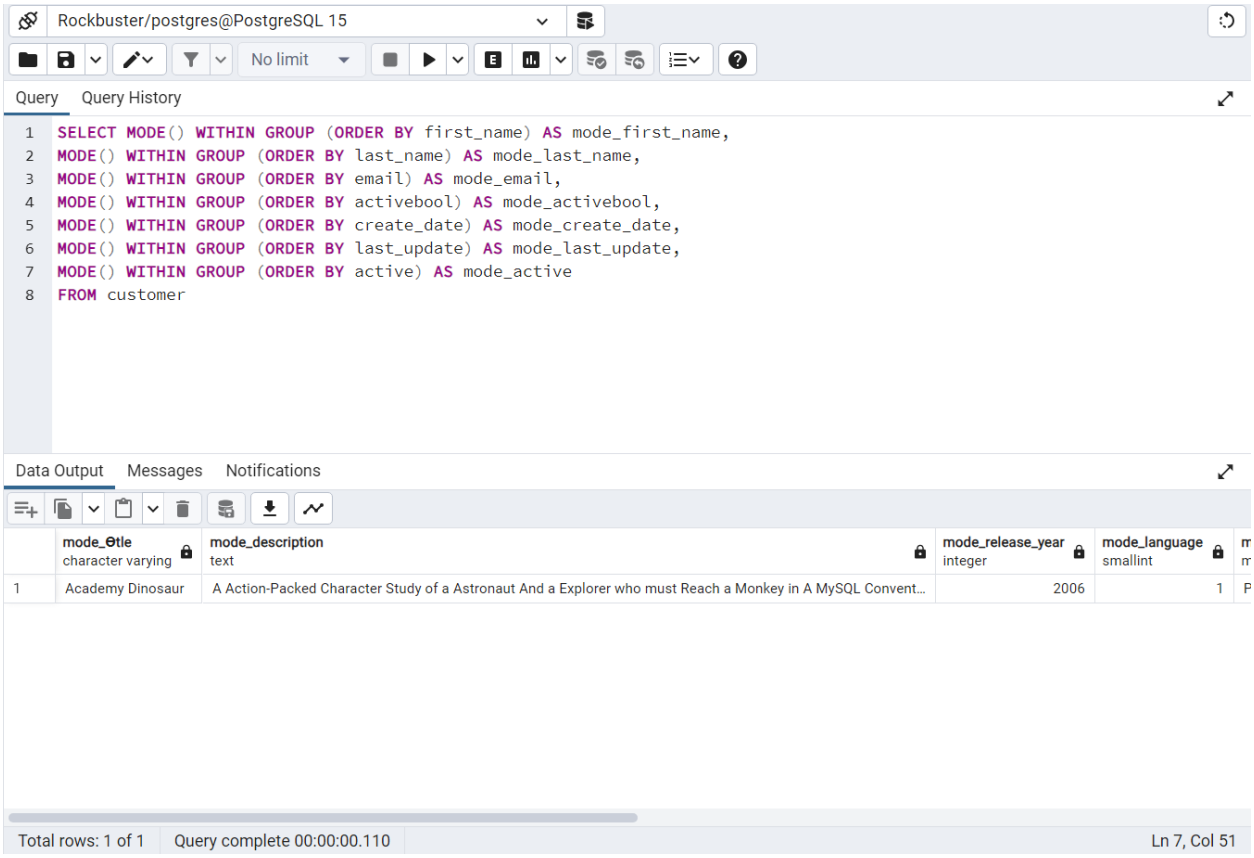

| min_rental | max_rental | avg_rental | count_rental | count_rows | min_replacement | max_replacement |
|---|---|---|---|---|---|---|
| 0.99 | 4.99 | 2.98 | 1000 | 1000 | 9.99 | 29.99 |
| avg_replacement | count_replacement_values | min_duration | rental_duration | rental_duration-2 | count_duration | min_length |
| 19.984 | 1000 | 3 | 7 | 4.985 | 1000 | 46 |
| max_length | avg_length | count_length | row_number | | | |
| 185 | 115.272 | 1000 | 1000 | | | |

## 2.a Film Table Non-Numerical




| mode_ÆŸtle | mode_description | mode_release_year | mode_language |
|---|---|---|---|
| Academy Dinosaur | A Action-Packed Character Study of a Astronaut And a Explorer who must Reach a Monkey in A MySQL Convention | 2006 | 1 |
| mode_rating | mode_last_update | mode_special_features | mode_fulltext |
| PG-13 | 50:59.0 | {Trailers,Commentaries,"Behind the Scenes"} | 'baloon':19 'confront':14 'documentari':5 'feminist':8,11,16 'mile':2 'must':13 'spi':1 'thrill':4 |

2.b Customer Table Non-Numerical




| mode_ÆŸtle | mode_description | mode_release_year | mode_language |
|---|---|---|---|
| Academy Dinosaur | A Action-Packed Character Study of a Astronaut And a Explorer who must Reach a Monkey in A MySQL Convention | 2006 | 1 |
| mode_rating | mode_last_update | mode_special_features | mode_fulltext |
| PG-13 | 50:59.0 | {Trailers,Commentaries,"Behind the Scenes"} | 'baloon':19 'confront':14 'documentari':5 'feminist':8,11,16 'mile':2 'must':13 'spi':1 'thrill':4 |

3.a I find it is easier to use SQL as I continue profile data. Excel requires multiple steps including pivot tables and equations that you have to enter individually to get the same results as SQL which only takes you entering the equation in one solid flow versus multiple times. Also, SQL allows you to check multiple statistics of your data without having to use a pivot table. I believe (especially with a large data set like a data base) SQL is the way to go for efficiency purposes.