

Exercise 3.6

Chrisie Hunter

1. Checking for duplicate records in the film table.

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure for 'Rockbuster/postgres@PostgreSQL 15'. The main pane shows a SQL query in the 'Query' tab:

```
12 special_features,  
13 fulltext,  
14 Count(*)  
15 From film  
16 Group BY film_id,  
17 title,  
18 description,  
19 release_year,  
20 language_id,  
21 rental_duration,  
22 rental_rate,  
23 length,  
24 replacement_cost,  
25 rating,  
26 last_update,  
27 special_features,  
28 fulltext  
29 Having Count(*) >1;  
30
```

The 'Data Output' tab is empty, indicating no results were returned. The status bar at the bottom shows 'Total rows: 0 of 0' and 'Query complete 00:00:00.312'.

Checking several columns for non-uniform data. Doesn't appear to be any issues.

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure for 'Rockbuster/postgres@PostgreSQL 15'. The main pane shows a SQL query in the 'Query' tab:

```
1 Select distinct rating,  
2 release_year,  
3 rental_duration,  
4 rental_rate  
5 From film  
6  
7
```

The 'Data Output' tab displays the results of the query:

rating	release_year	rental_duration	rental_rate
R	2006	5	4.99
PG	2006	5	2.99
NC-17	2006	4	4.99
PG-13	2006	6	0.99
NC-17	2006	3	2.99
PG-13	2006	3	2.99
NC-17	2006	4	0.99
PG	2006	7	4.99
NC-17	2006	6	4.99
PG	2006	4	4.99
R	2006	5	0.99
PG-13	2006	6	4.99

The status bar at the bottom shows 'Total rows: 75 of 75' and 'Query complete 00:00:00.161'.

Checking for null values. This query was used to check film_id, rating, rental_duration, and release year. No missing data found.

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure for 'Rockbuster' on a 'PostgreSQL 15' server. The main pane shows a SQL query in the 'Query' tab:

```
1 --searching table for missing data
2 Select film_id,
3     title,
4     description,
5     release_year,
6     language_id,
7     rental_duration,
8     rental_rate,
9     length,
10    replacement_cost,
11    rating,
12    last_update,
13    special_features,
14    fulltext
15 From film
16 Where film_id is null;
17
```

The 'Data Output' tab is empty, showing 'Total rows: 0 of 0'. The status bar at the bottom indicates 'Query complete 00:00:00.131'.

Checking for duplicates in Customer table.

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure for 'Rockbuster' on a 'PostgreSQL 15' server. The main pane shows a SQL query in the 'Query' tab:

```
1 Select customer_id,
2     store_id,
3     first_name,
4     last_name,
5     email,
6     address_id,
7     activebool,
8     create_date,
9     last_update,
10    active,
11    Count(*)
12 From customer
13 Group By customer_id,
14     store_id,
15     first_name,
16     last_name,
17     email,
18     address_id,
19     activebool,
20     create_date,
21     last_update,
22     active
23 Having Count(*) >1;
24
25
```

The 'Data Output' tab is empty, showing 'Total rows: 0 of 0'. The status bar at the bottom indicates 'Query complete 00:00:00.157'.

Checking for non-uniform data. Nothing appears to be non-uniform.

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure: Servers (1) > PostgreSQL 15 > Databases (2) > Rockbuster. The main pane shows a SQL query: `SELECT DISTINCT customer_id, store_id, first_name, last_name, email, address_id, activebool, create_date, last_update, active FROM customer`. The 'Data Output' tab is active, displaying a table with 10 columns: 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), last_update (timestamp without time zone), and active (integer). The table contains 6 rows of data. The status bar at the bottom indicates 'Total rows: 599 of 599' and 'Query complete 00:00:00.185'.

customer_id	store_id	first_name	last_name	email	address_id	activebool	create_date	last_update	active
357	1	Kaith	Rico	keith.rico@sakilacustomer.org	362	true	2006-02-14	2013-05-26 14:49:45.738	
171	2	Dolores	Wagner	dolores.wagner@sakilacustomer.org	175	true	2006-02-14	2013-05-26 14:49:45.738	
139	1	Amber	Dixon	amber.dixon@sakilacustomer.org	143	true	2006-02-14	2013-05-26 14:49:45.738	
471	1	Dean	Sauer	dean.sauer@sakilacustomer.org	476	true	2006-02-14	2013-05-26 14:49:45.738	
594	1	Eduardo	Hiatt	eduardo.hiatt@sakilacustomer.org	600	true	2006-02-14	2013-05-26 14:49:45.738	
401	2	Tony	Caranzena	tony.caranzena@sakilacustomer.org	406	true	2006-02-14	2013-05-26 14:49:45.738	

Checking for missing data. A check was run on store_id, email, first_name, last_name

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure: Servers (1) > PostgreSQL 15 > Databases (2) > Rockbuster. The main pane shows a SQL query: `SELECT DISTINCT customer_id, store_id, first_name, last_name, email, address_id, activebool, create_date, last_update, active FROM customer WHERE email IS NULL`. The 'Data Output' tab is active, displaying a table with 10 columns: 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), last_update (timestamp without time zone), and active (integer). The table is empty. The status bar at the bottom indicates 'Total rows: 0 of 0' and 'Query complete 00:00:00.169'.

customer_id	store_id	first_name	last_name	email	address_id	activebool	create_date	last_update	active
-------------	----------	------------	-----------	-------	------------	------------	-------------	-------------	--------

If duplicates were located, creating a view would allow you to use the Group By function, which would return unique cases only. Another option would be to delete the records using the Delete command instead of Select. Then using a Where command combined with Not In to delete the duplicate record.

For non-uniform data, the Update command would need to be used, with Set and Where / In syntax. Set is where you tell it what you want it to be and Where / In is where you tell it which values to update.

Incorrect data is far more difficult to address. This generally requires going back to the source owner. However, sometime, depending on the possible answers or range, it is easier to determine what the value should be.

Missing data is common and can be corrected through imputing values or ignore columns with a high percentage of missing data.

2. Screen shots for film_table. Screen shot 2 and 3 are just showing my columns at the bottom.

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, including Servers (1), PostgreSQL 15, Databases (2), and the Rockbuster database. The main pane shows a SQL query in the 'Query' tab, which is a complex SELECT statement with multiple columns and functions. The 'Data Output' pane at the bottom shows the results of the query, which is a single row of data. The query is as follows:

```
1 Select Min(release_year) As min_release_year,
2 Min(rental_duration) As min_rental_duration,
3 Min(rental_rate) As min_rental_rate,
4 Min(length) As min_movie_length,
5 Min(replacement_cost) As min_replacement_cost,
6 Max(release_year) As max_release_year,
7 Max(rental_duration) As max_rental_duration,
8 Max(rental_rate) As max_rental_rate,
9 Max(length) As max_movie_length,
10 Max(replacement_cost) As max_replacement_cost,
11 Round(Avg(release_year),0) As average_release_year,
12 Round(Avg(rental_duration),2) As average_rental_duration,
13 Round(Avg(rental_rate),2) As average_rental_rate,
14 Round(Avg(length),2) As average_movie_length,
15 Round(Avg(replacement_cost),2) As average_replacement_cost,
16 Count(release_year) As count_release_year,
17 Count(rental_duration) As count_rental_duration,
18 Count(rental_rate) As count_rental_rate,
19 Count(length) As count_rental_length,
20 Count(replacement_cost) As count_replacement_cost,
21 Count(*) As count_rows,
22 Mode() Within Group (Order By rating) As rating_modal_value
23 From film;
```

The results table shows the following data:

min_release_year	min_rental_duration	min_rental_rate	min_movie_length	min_replacement_cost	max_release_year	max_rental_duration	max_rental_rate	max_movie_length	max_repl
2006	3	0.99	46	9.99	2006	7	4.99	185	

Total rows: 1 of 1 Query complete 00:00:00.138 Ln 22, Col 53

pgAdmin 4

File Object Tools Help

Browsers

- Servers (1)
 - PostgreSQL 15
 - Databases (2)
 - Rockbuster
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas
 - Subscriptions
 - postgres
 - Login/Group Roles
 - Tablespaces

Dashboard Properties SQL Statistics Dependencies Dependents Processes Rockbuster/postgres@PostgreSQL 15*

Rockbuster/postgres@PostgreSQL 15

Query Query History

```

1 Select Min(release_year) As min_release_year,
2 Min(rental_duration) As min_rental_duration,
3 Min(rental_rate) As min_rental_rate,
4 Min(length) As min_movie_length,
5 Min(replacement_cost) As min_replacement_cost,
6 Max(release_year) As max_release_year,
7 Max(rental_duration) As max_rental_duration,
8 Max(rental_rate) As max_rental_rate,
9 Max(length) As max_movie_length,
10 Max(replacement_cost) As max_replacement_cost,
11 Round(Avg(release_year),0) As average_release_year,
12 Round(Avg(rental_duration),2) As average_rental_duration,
13 Round(Avg(rental_rate),2) As average_rental_rate,
14 Round(Avg(length),2) As average_movie_length,
15 Round(Avg(replacement_cost),2) As average_replacement_cost,
16 Count(release_year) As count_release_year,
17 Count(rental_duration) As count_rental_duration,
18 Count(rental_rate) As count_rental_rate,
19 Count(length) As count_rental_length,
20 Count(replacement_cost) As count_replacement_cost,
21 Count(*) As count_rows,
22 Mode() Within Group (Order By rating) As rating_modal_value
23 From file;

```

Data Output Messages Notifications

	max_replacement_cost	average_release_year	average_rental_duration	average_rental_rate	average_movie_length	average_replacement_cost	count_release_year	count_rental_duration	count_rental_rate	count_replacement_cost	count_rows	rating_modal_value
1	numeric	29.99	2006	4.99	2.98	115.27	19.98	1000	1000	1000	1000	PG-13

Total rows: 1 of 1 Query complete 00:00:00.138 Ln 22, Col 53

77°F Haze 2:16 PM 4/2/2023

pgAdmin 4

File Object Tools Help

Browsers

- Servers (1)
 - PostgreSQL 15
 - Databases (2)
 - Rockbuster
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas
 - Subscriptions
 - postgres
 - Login/Group Roles
 - Tablespaces

Dashboard Properties SQL Statistics Dependencies Dependents Processes Rockbuster/postgres@PostgreSQL 15*

Rockbuster/postgres@PostgreSQL 15

Query Query History

```

1 Select Min(release_year) As min_release_year,
2 Min(rental_duration) As min_rental_duration,
3 Min(rental_rate) As min_rental_rate,
4 Min(length) As min_movie_length,
5 Min(replacement_cost) As min_replacement_cost,
6 Max(release_year) As max_release_year,
7 Max(rental_duration) As max_rental_duration,
8 Max(rental_rate) As max_rental_rate,
9 Max(length) As max_movie_length,
10 Max(replacement_cost) As max_replacement_cost,
11 Round(Avg(release_year),0) As average_release_year,
12 Round(Avg(rental_duration),2) As average_rental_duration,
13 Round(Avg(rental_rate),2) As average_rental_rate,
14 Round(Avg(length),2) As average_movie_length,
15 Round(Avg(replacement_cost),2) As average_replacement_cost,
16 Count(release_year) As count_release_year,
17 Count(rental_duration) As count_rental_duration,
18 Count(rental_rate) As count_rental_rate,
19 Count(length) As count_rental_length,
20 Count(replacement_cost) As count_replacement_cost,
21 Count(*) As count_rows,
22 Mode() Within Group (Order By rating) As rating_modal_value
23 From file;

```

Data Output Messages Notifications

	max_replacement_cost	average_release_year	average_rental_duration	average_rental_rate	average_movie_length	average_replacement_cost	count_release_year	count_rental_duration	count_rental_rate	count_replacement_cost	count_rows	rating_modal_value
1	numeric	29.99	2006	4.99	2.98	115.27	19.98	1000	1000	1000	1000	PG-13

Total rows: 1 of 1 Query complete 00:00:00.138 Ln 22, Col 53

77°F Haze 2:17 PM 4/2/2023

Screen shots for customer table.

The screenshot displays the pgAdmin 4 interface. On the left, the 'Servers' tree shows 'PostgreSQL 15' expanded, with 'Databases (2)' containing 'Rockbuster' and 'postgres'. The 'Rockbuster' database is selected, and the 'Query' tab is active. The SQL query is as follows:

```
1 Select Min(create_date) As min_create_date,
2 Min(last_update) As min_last_update,
3 Min(active) As min_active,
4 Max(create_date) As max_create_date,
5 Max(last_update) As max_last_update,
6 Max(active) As max_active,
7 Round(Avg(active),2) As average_active,
8 Count(*) As count_rows,
9 Mode() Within Group (Order By first_name) As first_name_modal_value,
10 Mode() Within Group (Order By last_name) As last_name_modal_value
11 From customer;
```

The 'Data Output' tab shows the results of the query in a table with 11 columns and 1 row:

	min_create_date	min_last_update	min_active	max_create_date	max_last_update	max_active	average_active	count_rows	first_name_modal_value	last_name_modal_value
1	2006-02-14	2013-05-26 14:49:45.738	0	2006-02-14	2013-05-26 14:49:45.738	1	0.97	599	Jamie	Abney

At the bottom, the status bar indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.119'. The system tray at the very bottom shows the date and time as '2:39 PM 4/2/2023'.