Answers 3.6

Find out if the film table and the customer table contain any dirty data, specifically non-uniform or duplicate data, or missing values. Next to each query write 2 to 3 sentences explaining how you would clean the data (even if the data is not dirty).

Checking for Duplicates in “film”

Query Query History

```
SELECT title, release_year, language_id, rental_duration, length, COUNT(*)
FROM film
GROUP BY title, release_year, language_id, rental_duration, length
HAVING COUNT(*) > 1
```

Data output Messages Notifications

title character varying (255)	release_year integer	language_id smallint	rental_duration smallint	length smallint	count bigint

Checking for Duplicates in “customer”

Query Query History

```
SELECT first_name, last_name, email, address_id, COUNT(*)
FROM customer
GROUP BY first_name, last_name, email, address_id
HAVING COUNT(*) > 1
```

Data output Messages Notifications

first_name character varying (45)	last_name character varying (45)	email character varying (50)	address_id smallint	count bigint

- There are no duplicates in either.
- If there were duplicate records, I would run a view command that will make each row unique based on the desired above command.

Checking for Non-Uniform values

Query Query History

```
SELECT DISTINCT rating
FROM film
GROUP BY rating
```

Data output Messages Notificat

	rating mpaa_rating
1	G
2	PG
3	PG-13
4	R
5	NC-17

- There are no non-uniform values.
- If there were, a command using "UPDATE film" followed by SET rating = '[desired rating]' followed by WHERE listing all of the non-uniform values that need to be altered. This could be repeated for multiple ratings if there were multiple errors.

Checking for Missing Values

- If there are missing values, we must determine whether they need to be replaced.
- If there are missing numeric values, a possible solution, if there are only very few, is to run the following command:
  UPDATE name_of_table
  SET = AVG(desired_column)
  WHERE desired_column IS NULL

Use SQL to calculate descriptive statistics for both the film table and the customer table. For numerical columns, this means finding the minimum, maximum, and average values. For non-numerical columns, calculate the mode value.

For "film" table

Query Query History Scratch Pad

```
SELECT MIN(release_year) AS oldest_release_year,
       MAX(release_year) AS newest_release_year,
       mode() WITHIN GROUP (ORDER BY release_year) AS most_common_release_year,
       MIN(rental_duration) AS min_rental_duration,
       MAX(rental_duration) AS max_rental_duration,
       mode() WITHIN GROUP (ORDER BY rental_duration) AS most_common_rental_duration,
       MIN(rental_rate) AS min_rental_rate,
       MAX(rental_rate) AS max_rental_rate,
       AVG(rental_rate) AS avg_rental_rate,
       MIN(length) AS shortest_film_length,
       MAX(length) AS longest_film_length,
       AVG(length) AS avg_film_length,
       mode() WITHIN GROUP (ORDER BY rating) AS most_common_rating
FROM film
```

Data output Messages Notifications

	oldest_release_year integer	newest_release_year integer	most_common_release_year integer	min_rental_duration smallint	max_rental_duration smallint	most_common_rental_duration smallint	min_rental_rate numeric	max_rental_rate numeric	avg_rental_rate numeric	shortest_film_length smallint	longest_film_length smallint	avg_film_length numeric	most_common_rating mpaa_rating
1	2006	2006	2006	3	7	6	0.99	4.99	2.98	46	185	115.272	PG-13

For “customer” table

Query Query History

```sql
SELECT mode() WITHIN GROUP (ORDER BY first_name) AS most_common_first_name,
       mode() WITHIN GROUP (ORDER BY last_name) AS most_common_larst_name,
       MIN(create_date) AS oldest_create_date,
       MAX(create_date) AS newest_create_date,
       AVG(active) AS customer_activity_rate
FROM customer
```

Data output Messages Notifications

	most_common_first_name character varying	most_common_larst_name character varying	oldest_create_date date	newest_create_date date	customer_activity_rate numeric
1	Jamie	Abney	2006-02-14	2006-02-14	0.9749582637729549

Back in Achievement 1 you learned about data profiling in Excel. Based on your previous experience, which tool (Excel or SQL) do you think is more effective for data profiling, and why? Consider their respective functions, ease of use, and speed. Write a short paragraph in the running document that you have started

SQL is a significantly better data profiling tool. In SQL you can simply type a command, and you will find all the information you want, whereas in Excel you must fetch it from the table. Looking for errors is even easier in SQL, as the commands save you the trouble of sifting through mass amounts of data, where mistakes are easy to miss. The difference in time efficiency too, gives SQL a boost, which can be crucial in the corporate world.