# Databases & SQL for Analysts

## Task 3.6: Summarizing & Cleaning Data in SQL

---

Ngawang Dhundup

**LINK NAME:**

## Directions:

Rockbuster's database engineers have loaded some new data into the database, and your manager has asked you to clean and profile it. Follow the instructions below to complete their request:

1. **Check for and clean dirty data:** Find out if the film table and the customer table contain any dirty data, specifically non-uniform or duplicate data, or missing values. Create a new "Answers 3.6" document and copy-paste your queries into it. Next to each query write 2 to 3 sentences explaining how you would clean the data (even if the data is not dirty).

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

```
SELECT customer_id,
        store_id,
        first_name,
        last_name,
        email,
        address_id,
        activebool,
        create_date,
        last_update,
        active
FROM customer
GROUP BY customer_id,
        store_id,
        first_name,
        last_name,
        email,
        address_id,
        activebool,
        create_date,
        last_update,
        active
HAVING COUNT(*)>1;
```

- Checked for duplicates on all numeric columns in film, and customer table. No duplicates were to be found. If there were duplicates I would delete them if I had permission, or would mark the duplicates in a new column for to be deleted by a sr analyst.

1b) check for non-uniform values

```
- SELECT DISTINCT film_id,
-       title,
-       release_year,
-       language_id,
-       rental_duration,
-       rental_rate,
-       length,
-       replacement_cost,
```

- rating
- FROM film;
-
- SELECT DISTINCT rating
- FROM film
- GROUP BY rating;

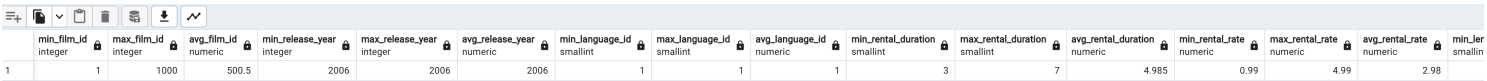
```
SELECT DISTINCT customer_id,
        store_id,
        first_name,
        last_name,
        email,
        address_id,
        activebool,
        create_date,
        last_update,
        active
FROM customer;

SELECT DISTINCT active
FROM customer
GROUP BY active;
```

- Check distinct values and group by each column to see if values are uniform or not. All values made sense, and no anomalies present. No missing values were found. If there were you could input the values using the average of the column.
- In order to clean not uniform values, you would alter the table, or create a view first then update
-

2. **Summarize your data:** 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. Copy-paste your SQL queries and their outputs into your answers document.

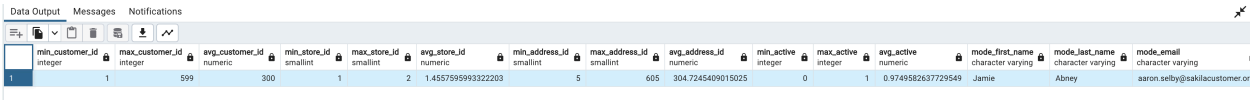
```
SELECT
MIN (film_id) AS min_film_id,
MAX (film_id) AS max_film_id,
```

```
AVG (film_id) AS avg_film_id,
MIN (release_year) AS min_release_year,
MAX (release_year) AS max_release_year,
AVG (release_year) AS avg_release_year,
MIN (language_id) AS min_language_id,
MAX (language_id) AS max_language_id,
AVG (language_id) AS avg_language_id,
MIN (rental_duration) AS min_rental_duration,
MAX (rental_duration) AS max_rental_duration,
AVG (rental_duration) AS avg_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 min_length,
MAX (length) AS max_length,
AVG (length) AS avg_length,
MIN (replacement_cost) AS min_replacement_cost,
MAX (replacement_cost) AS max_replacement_cost,
AVG (replacement_cost) AS avg_replacement_cost,
MODE () WITHIN GROUP (ORDER BY title) AS mode_title,
MODE () WITHIN GROUP (ORDER BY description) AS mode_description,
MODE () WITHIN GROUP (ORDER BY rating) AS mode_rating,
MODE () WITHIN GROUP (ORDER BY special_features) AS mode_special_features, MODE
() WITHIN GROUP (ORDER BY fulltext) AS mode_fulltext
FROM film;
```



| | min_film_id integer | max_film_id integer | avg_film_id numeric | min_release_year integer | max_release_year integer | avg_release_year numeric | min_language_id smallint | max_language_id smallint | avg_language_id numeric | min_rental_duration smallint | max_rental_duration smallint | avg_rental_duration numeric | min_rental_rate numeric | max_rental_rate numeric | avg_rental_rate numeric | min_len smallin |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1000 | 500.5 | 2006 | 2006 | 2006 | 1 | 1 | 1 | 3 | 7 | 4.985 | 0.99 | 4.99 | 2.98 | |

```
SELECT
MIN (customer_id) AS min_customer_id,
MAX (customer_id) AS max_customer_id,
AVG (customer_id) AS avg_customer_id,
MIN (store_id) AS min_store_id,
MAX (store_id) AS max_store_id,
AVG (store_id) AS avg_store_id,
MIN (address_id) AS min_address_id,
MAX (address_id) AS max_address_id,
AVG (address_id) AS avg_address_id,
MIN (active) AS min_active,
MAX (active) AS max_active,
AVG (active) AS avg_active,
MODE () WITHIN GROUP (ORDER BY first_name) AS mode_first_name,
MODE () WITHIN GROUP (ORDER BY last_name) AS mode_last_name,
MODE () WITHIN GROUP (ORDER BY email) AS mode_email,
MODE () WITHIN GROUP (ORDER BY active) AS mode_active,
MODE () WITHIN GROUP (ORDER BY activebool) AS mode_activebool
FROM customer;
```



Data Output | Messages | Notifications

| | min_customer_id integer | max_customer_id integer | avg_customer_id numeric | min_store_id smallint | max_store_id smallint | avg_store_id numeric | min_address_id smallint | max_address_id smallint | avg_address_id numeric | min_active integer | max_active integer | avg_active numeric | mode_first_name character varying | mode_last_name character varying | mode_email character varying |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 599 | 300 | 1 | 2 | 1.4557595993322203 | 5 | 605 | 304.7245409015025 | 0 | 1 | 0.9749582637729549 | Jamie | Abney | aaron.selby@sakilacustomer.or |

3. **Reflect on your work:** 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.

When it comes to data profiling, sql is easier to use. With sql you can do all of the data profiling with just a few lines, versus with excel how you have to copy, and paste, and create new columns and rows, and sheets. I feel that combining the results on sql will be harder compared to excel however though.

4. Save your “Answers 3.6” document as a PDF and upload it here for your tutor to review.