3.6: Summarizing & Cleaning Data in SQL

Innocent Bayai

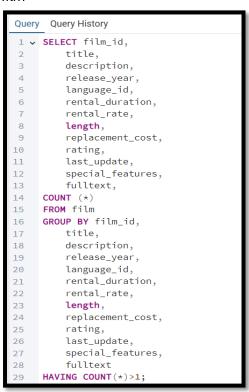
26 May 2024

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**. Next to each query write 2 to 3 sentences explaining how you would clean the data (even if the data is not dirty).

DUPLICATE QUERY

film



customer

```
Query History
                              SELECT customer_id,
                                                                         store_id,
                                                                       last_name,
                                                                         email.
                                                                       address_id,
                                                                       activebool,
                                                                       create date.
                                                                       last update,
                                                                     active,
                         COUNT (*)
                         FROM customer
                           GROUP BY customer_id,
                                                                     store_id,
                                                                         first_name,
                                                                       last_name,
                                                                       email,
                                                                       address_id,
                                                                       activebool,
                                                                       create_date
                                                                     last update.
                                                                       active
                         HAVING COUNT(*)>1:
Data Output Messages Notifications
 타 🖺 v 📋 v 📋 🗟 👲 🚜
                           customer_id store_id first_name email character varying (45) last_name character varying (45) email cha
         otal rows: 0 of 0 Ouery complete 00:00:00.078
```

Both tables have no duplicate records (film and customer).

If I were to clean this data, I would do the following:

i. If granted the authority, I would delete the duplicate records using the

DELETE

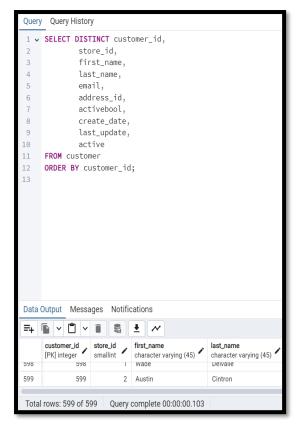
FROM

WHERE command.

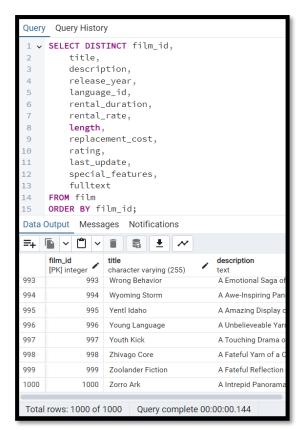
ii. Otherwise, I use the view table which allows me view only unique records as deleting records might not be allowed. I use the **SELECT DISTINCT, FROM** command.

NON-UNIFORM DATA

customer table



film table



The use of SELECT DISTINCT, FROM, ORDER BY commands selects unique records in both tables (customer and film). The commands do not identify duplicate records but returns only unique records thereby. However, the returned records in both cases are equal to the count of records prior running the SQL commands implying that there are no non-uniform records.

If any non-uniform records existed, I would use the UPDATE, SET, WHERE commands to correct the records.

MISSING VALUES

No missing records were found for both tables (film and customer). See screen shots for SQL commands in the next page.

If I found missing values, the best way to fix this problem is to work with only the available data for my analysis if the count of the missing values is significant. Otherwise, I can impute the missing values using the average.

Film customer

```
Query Query History
16
17 v SELECT *
18
     FROM film
19
     WHERE (film_id,
20
          title,
          description,
21
          release_year,
23
          language_id,
          rental_duration,
25
          rental_rate,
26
          length,
          replacement_cost,
27
28
          rating,
29
          last_update,
30
          special_features,
          fulltext)
31
32
     IS NULL
     ORDER BY film_id;
Data Output Messages Notifications
=+ 🖺 ∨ 🖺 ∨
     film_id
                  title
                                       description
     [PK] integer
                  character varying (255)
                                       text
```

```
35 ∨ SELECT*
     FROM customer
37
     WHERE (customer_id,
38
              store_id,
              first_name,
39
40
              last_name,
              email,
41
              address_id,
42
              activebool,
43
44
              create_date,
45
              last_update,
46
              active)
     IS NULL
47
48
     ORDER BY customer_id
Data Output Messages Notifications
                             first name
     customer_id
                  store id
     [PK] integer
                  smallint
                             character varying (45)
```

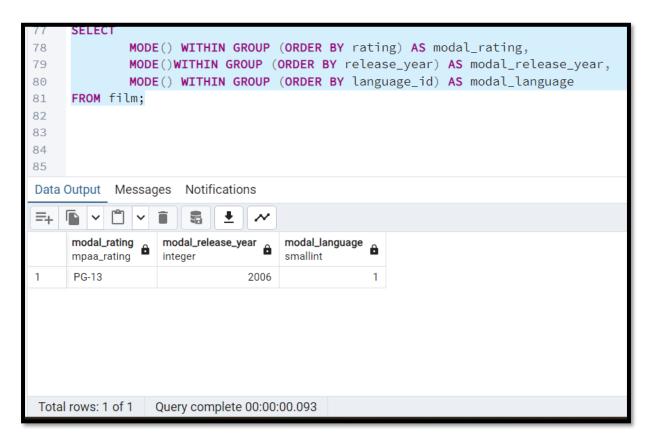
2.Summarize your data: Use SQL to calculate descriptive statistics for both the film table and the customer table. For numerical columns, this means finding the minimum, maximum, and average values. For non-numerical columns, calculate the mode value. Copy-paste your SQL queries and their outputs into your answers document.

Descriptive statistics: film

```
Query Query History
                                                                                                                                                                                                                                                                                                                                                                                                                          Scratch Pad X
                 ORDER BY customer_id
                  SELECT
                                              MIN(rental_rate) AS min_rent,
                                              MAX(rental_rate) AS max_rent,
                                              AVG(rental_rate) AS avg_rent,
                                              MIN(rental_duration) AS min_rental_duration,
                                              MAX(rental_duration) AS max_rental_duration,
                                              AVG(rental_duration) AS avg_rental_duration,
                                              MIN(length) AS min_length,
                                              MAX(length) AS max_length,
                                              AVG(length) AS abg_length,
                                              MIN(replacement_cost) AS min_replacement_cost,
                                              MAX(replacement_cost) AS max_replacement_cost
                 FROM film:
Data Output Messages Notifications
              □ ∨ 🗂 ∨ ≡
                                                                                             <u>+</u> ~
                                                                                                                                                    min_rental_duration a max_rental_duration a 
                    min_rent numeric max_rent numeric avg_rent numeric
                                    0.99
                                                                           4.99 2.98000000000000000
                                                                                                                                                                                                                     3
                                                                                                                                                                                                                                                                                        7 4.98500000000000000
                                                                                                                                                                                                                                                                                                                                                                                                  46
                                                                                                                                                                                                                                                                                                                                                                                                                                           185 115.27200000000000000
```

NB: Customer table doesn't have numerical values hence no descriptive statistics are estimated.

Mode for non-numeric variables in film table



Mode for non-numeric variables in customer table



3.Reflect on your work: 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.

Excel is for basic data profiling tasks but cannot handle the processing of large amounts of data and can become slow and inefficient when handling complex datasets. With excel, one must have the data open and effect the manipulations with it. However, SQL is more powerful for data profiling as it allows more complex queries to be effected on large data sizes. SQL is also faster and efficient than excel. SQL has more flexibility and control over the profiling process. Once a database is loaded into SQL, one only issues commands and get results. This is different in excel as the data has be interfaced with for each processing requirement. Summarily, SQL is more effective and efficient for data profiling due to its advanced functions, speed and flexibility.