# Achievement 3.6

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).

Non-uniform

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

```
SELECT DISTINCT store_id
FROM customer
GROUP BY store_id
```

Using DISTINCT to check if any data are entered differently. If so, for example,

```
UPDATE table
SET col1 = 'G'
WHERE col1 IN ('gen',
               'g',
               'General')
```

The above query will group all "gen" "g" and "General" and change them into "G" in col1.

Duplicate Data

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

Data output Messages Notifications

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

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

Data output Messages Notifications

| customer_id [PK] integer | store_id smallint | first_name character varying (45) | last_name character varying (45) | count bigint |
|---|---|---|---|---|

If there are duplicated data, we can create a unique view table by choosing the right column ( CREATE VIEW viewname AS) or we can delete the data with query.

Missing Values

- If there are many missing values in a coloumn, we can just omit the column, and if there are a few missing data we can impute values such as using the AVG query.

```
UPDATE tablename
SET = AVG(col1)
WHERE col1 IS NULL
```

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(rental_duration) AS min_duration,
       MAX(rental_duration) AS max_duration,
       AVG(rental_duration) AS avg_duration,
       MIN(rental_rate) AS min_rent,
       MAX(rental_rate) AS max_rent,
       AVG(rental_rate) AS avg_rent,
       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
FROM film
```

Data output Messages Notifications

| | min_duration smallint | max_duration smallint | avg_duration numeric | min_rent numeric | max_rent numeric | avg_rent numeric | min_length smallint | max_length smallint | avg_length numeric | min_replacem numeric | max_replacem numeric | avg_replacem numeric |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 7 | 4.985 | 0.99 | 4.99 | 2.98 | 46 | 185 | 115.272 | 9.99 | 29.99 | 19.984 |

```
SELECT  mode() WITHIN GROUP (ORDER BY title) AS modal_title,
        mode() WITHIN GROUP (ORDER BY description) AS modal_description,
        mode() WITHIN GROUP (ORDER BY release_year) AS modal_release_year,
        mode() WITHIN GROUP (ORDER BY language_id) AS modal_language_id,
        mode() WITHIN GROUP (ORDER BY rating) AS modal_rating
FROM film
```

Data output Messages Notifications

| | modal_title character varying | modal_description text | modal_release_year integer | modal_language_id smallint | modal_rating mpaa_rating |
|---|---|---|---|---|---|
| 1 | Academy Dinosaur | A Action-Packed C... | 2006 | 1 | PG-13 |

```sql
SELECT  MIN(active) AS min_active,
        MAX(active) AS max_active,
        AVG(active) AS avg_active,
        MIN(address_id) AS min_address,
        MAX(address_id) AS max_address,
        AVG(address_id) AS avg_address,
        MIN(customer_id) AS min_customer,
        MAX(customer_id) AS max_customer,
        AVG(customer_id) AS avg_customer,
        MIN(store_id) AS min_store,
        MAX(store_id) AS max_store,
        AVG(store_id) AS avg_store
FROM customer
```

Data output  Messages  Notifications

| | min_active integer | max_active integer | avg_active numeric | min_address smallint | max_address smallint | avg_address numeric | min_custome integer | max_custome integer | avg_custome numeric | min_store smallint | max_store smallint | avg_store numeric |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0.974958263 | 5 | 605 | 304.7245409 | 1 | 599 | 300 | 1 | 2 | 1.455759599 |

```sql
SELECT  mode() WITHIN GROUP (ORDER BY first_name) AS modal_first_name,
        mode() WITHIN GROUP (ORDER BY last_name) AS modal_last_name,
        mode() WITHIN GROUP (ORDER BY email) AS modal_email
FROM customer
```

Data output  Messages  Notifications

| | modal_first_name character varying | modal_last_name character varying | modal_email character varying |
|---|---|---|---|
| 1 | Jamie | Abney | aaron.selby@sak... |

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.

- It's more effective to use SQL for data profiling and getting information because SQL is able to collect all data from the database whereas in Excel you need to know what data to include first. Also, there can be many mistakes in Excel such as deleting or modifying data by accidents, whereas SQL will just show you the query is wrong, and you just need to adjust the query. Both programs need to have good understanding of the software to perform well. Overall, SQL is a better tool for company with lots of data and constantly pulling data from the database without making much mistakes.