Step 1:

Duplicate Data

```
SELECT film_id, COUNT(*) FROM film GROUP BY film_id HAVING COUNT(*)>1
```

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

Both of these query's resulted with zero duplicates. Yet if there were any duplicates, I would have cleaned the data by using the DISTINCT command to make sure that any duplicate values don't show up in the query output.

Non-uniform

```
SELECT rating FROM film GROUP BY rating;
```

```
SELECT first_name FROM customer GROUP BY first_name;
```

As far as I could tell, all of the data was uniform. But if it wasn't, then I could use the UPDATE command on alternat spellings of the same thing and make them uniform.

Missing Values

```
SELECT title FROM film GROUP BY title;
```

(The total number of rows is the same as the film_id column)

```
SELECT address_id FROM customer GROUP BY address_id;
```

(The total number of rows is the same as the customer_id column)

I didn't see any missing values. Yet if there were, I would have just omitted the column or row with the missing data from my SQL query. Though I would also specify which parts of the data I'm omitting and why as a comment in the query, so that any other analysts can understand what I was doing.

Step 2:

Film Table

```
SELECT
    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_movie_length,
    MAX(length) AS max_movie_length,
    AVG(length) AS avg_movie_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 rating) AS most_common_rating
FROM film;
```

min_rental_duration	max_rental_duration	avg_rental_duration	min_rental_rate	max_rental_rate	avg_rental_rate	min_movie_length	max_movie_length	avg_movie_length	min_replacement_cost	max_replacement_cost	avg_replacement_cost	most_common_rating
3	7	4.985	0.99	4.99	2.98	46	185	115.272	9.99	29.99	19.984	PG-13

Customer Table

```
SELECT
    MODE() WITHIN GROUP (ORDER BY store_id) AS most_common_store_id,
    MODE() WITHIN GROUP (ORDER BY first_name) AS most_common_first_name,
    MODE() WITHIN GROUP (ORDER BY last_name) AS most_common_last_name
FROM customer;
```

most_common_store_id	most_common_first_name	most_common_last_name
1	Jamie	Abney

Step 3:

I would say that profiling data is much easier in SQL compared to Excel. In SQL you can find the sum, avg, mode, etc. of multiple columns in a table at once, while in Excel you have to type the entire equation you want. As well as typing it multiple times for each column. SQL allows you to streamline the whole process.