Databases & SQL for Analysts

3.4: Database Querying in SQL

1. Refining Your Query

- **Write a SELECT command that contains only the film_id and title columns from the film table:**

```sql
SELECT film_id, title FROM public.film
```

Query plan of the original and revised queries:

- Original query: "Seq Scan on film (cost=0.00..64.00 rows=1000 width=384)"
- Revised query: "Seq Scan on film (cost=0.00..64.00 rows=1000 width=19)"

As the width in the original query is much larger than in the revised query, we can conclude the latter used fewer resources.

To optimize this query, we could:

- Use the WHERE clause to filter out unnecessary rows.
- Use the ORDER BY clause to sort the data in the most efficient way.

2. Ordering the Data

- **Run a query that selects every film from the film table, with the movies sorted by title from A to Z, then by most recent release year, and then by highest to lowest rental rate:**

```sql
SELECT * FROM public.film
ORDER BY title ASC
```

```sql
SELECT * FROM public.film
ORDER BY release_year DESC
```

```sql
SELECT * FROM public.film
ORDER BY rental_rate DESC
```

3. Grouping Data

- **Write a SQL query to answer the question: What is the average rental rate for each rating category?**

```sql
SELECT rating, AVG(rental_rate)
FROM public.film
GROUP BY rating
```

rating	avg
PG	3.0518556701030928
R	2.9387179487179487
NC-17	2.970952380952381
PG-13	3.034843049327354
G	2.888876404494382

- **What are the minimum and maximum rental durations for each rating category?**

```sql
SELECT rating, MIN(rental_duration), MAX(rental_duration)
FROM public.film
GROUP BY rating
```

rating	min	max
PG	3	7
R	3	7
NC-17	3	7
PG-13	3	7
G	3	7

4. Database Migration

- **Can you outline the procedure for migrating the data and who will be responsible for it?**

The procedure for migrating the data would involve the following steps:

- **Extract:** The first step is to extract the data from the external tool. This could be done by using an API or by downloading the data in a CSV or other format.
- **Transform:** The next step is to transform the data into a format that can be loaded into the data warehouse. This could involve cleaning the data, normalizing it, and transforming it into a format that is compatible with the data warehouse.
- **Load:** The final step is to load the data into the data warehouse. This could be done using a custom ETL script.

The responsibility for this ETL procedure would depend on the team's resources and expertise. It could be handled by a data engineer, a data analyst, or a combination of both.

- **What problems do you foresee if you start analyzing the data before it's been loaded into the data warehouse?**

If the data is not loaded into the data warehouse before it is analyzed, there are risks of inconsistences, data loss and data quality. These could lead to inaccurate, incomplete and unreliable results and conclusions.

Step 4

- **Write a DELETE command to remove the mystery category from the category table:**

```sql
DELETE FROM public.category
WHERE name = 'Mystery'
```

category_id	name
1	Action
2	Animation
3	Children
4	Classics
5	Comedy
6	Documentary
7	Drama
8	Family
9	Foreign
10	Games
11	Horror
12	Music
13	New
14	Sci-fi
15	Sports
16	Travel
17	Thriller
18	Crime
20	Romance
21	War

Step 5

- **Based on what you've learned so far, think about what it would be like to complete steps 1 to 4 with Excel instead of SQL. Are there any pros and cons to using SQL? Write a paragraph explaining your answer.**

In Excel, the SQL commands used in step 1 to 4 can be executed by using the Find & Replace function and manually modify the data. Generally, SQL is more efficient than Excel, as it can process large amounts of data quickly and accurately. Furthermore, SQL is more flexible than Excel, as it allows for more complex queries and data manipulation.