ROCKBUSTER

CASE STUDY

Business Problem

Rockbuster Stealth is a fictional movie rental company. It is planning to use its existing movie licenses to launch an online video rental service to stay competitive with other streaming providers. The objective of this analysis is to present insights on Rockbuster's existing customer database. The findings will serve as an input for effective launch strategy.

The business questions we are trying to answer are:

- Which movies contributed the most/least to revenue gain?
- What was the average rental duration for all videos?
- Which countries are Rockbuster customers based in?
- Where are customers with a high lifetime value based?
- Do sales figures vary between geographic regions?

Data Overview

The dataset rests in a SQL database and consists of 15 tables. A data dictionary and Entity Relationship Diagram (ERD) has been created as part of this analysis.

Tools

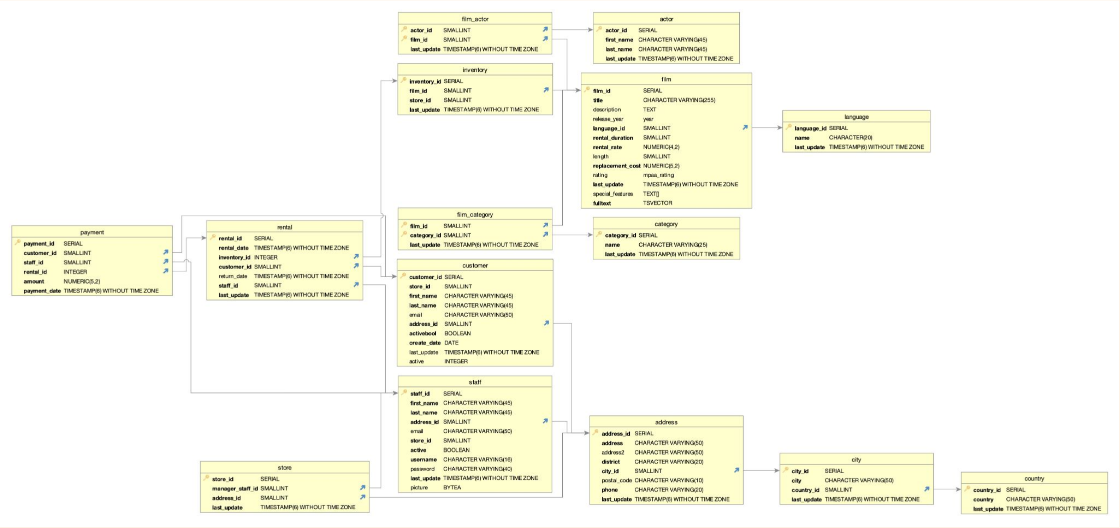
- PostgreSQL
- DBVisualizer
- Tableau
- Excel

Analytical Methods

Relational Database Management Systems (RDBMS)
Summarizing & Cleaning data
Joining tables
Common Table Expressions
SQL Database querying
Filtering data
Subqueries
Visualization, Data Dictionary

1. Data dictionary

- Extracted an entity relationship diagram for the database using DBVisualizer.
- Used this schema to create a data dictionary that contained table names, information stored in those tables, and table links.



film_actor	
actor_id	SMALLINT
film_id	SMALLINT
last_update	TIMESTAMP(6) WITHOUT TIME ZONE

actor	
actor_id	SERIAL
first_name	CHARACTER VARYING(45)
last_name	CHARACTER VARYING(45)
last_update	TIMESTAMP(6) WITHOUT TIME ZONE

inventory	
inventory_id	SERIAL
film_id	SMALLINT
store_id	SMALLINT
last_update	TIMESTAMP(6) WITHOUT TIME ZONE

film	
film_id	SERIAL
title	CHARACTER VARYING(255)
description	TEXT
release_year	year
language_id	SMALLINT
rental_duration	SMALLINT
rental_rate	NUMERIC(4,2)
length	SMALLINT
replacement_cost	NUMERIC(5,2)
rating	mpaa_rating
last_update	TIMESTAMP(6) WITHOUT TIME ZONE
special_features	TEXT[]
fulltext	TSVECTOR

language	
language_id	SERIAL
name	CHARACTER(20)
last_update	TIMESTAMP(6) WITHOUT TIME ZONE

payment	
payment_id	SERIAL
customer_id	SMALLINT
staff_id	SMALLINT
rental_id	INTEGER
amount	NUMERIC(5,2)
payment_date	TIMESTAMP(6) WITHOUT TIME ZONE

rental	
rental_id	SERIAL
rental_date	TIMESTAMP(6) WITHOUT TIME ZONE
inventory_id	INTEGER
customer_id	SMALLINT
return_date	TIMESTAMP(6) WITHOUT TIME ZONE
staff_id	SMALLINT
last_update	TIMESTAMP(6) WITHOUT TIME ZONE

film_category	
film_id	SMALLINT
category_id	SMALLINT
last_update	TIMESTAMP(6) WITHOUT TIME ZONE

category	
category_id	SERIAL
name	CHARACTER VARYING(25)
last_update	TIMESTAMP(6) WITHOUT TIME ZONE

customer	
customer_id	SERIAL
store_id	SMALLINT
first_name	CHARACTER VARYING(45)
last_name	CHARACTER VARYING(45)
email	CHARACTER VARYING(50)
address_id	SMALLINT
activebool	BOOLEAN
create_date	DATE
last_update	TIMESTAMP(6) WITHOUT TIME ZONE
active	INTEGER

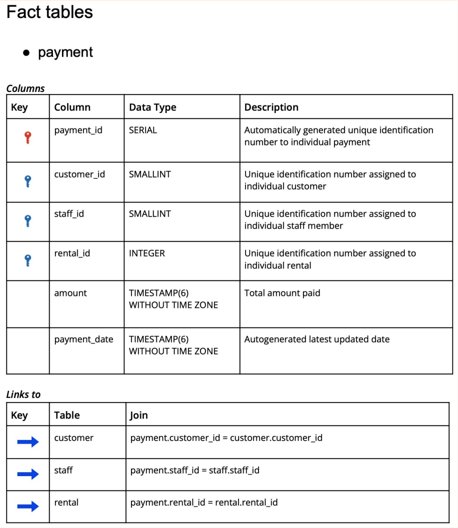
staff	
staff_id	SERIAL
first_name	CHARACTER VARYING(45)
last_name	CHARACTER VARYING(45)
address_id	SMALLINT
email	CHARACTER VARYING(50)
store_id	SMALLINT
active	BOOLEAN
username	CHARACTER VARYING(16)
password	CHARACTER VARYING(40)
last_update	TIMESTAMP(6) WITHOUT TIME ZONE
picture	BYTEA

store	
store_id	SERIAL
manager_staff_id	SMALLINT
address_id	SMALLINT
last_update	TIMESTAMP(6) WITHOUT TIME ZONE

address	
address_id	SERIAL
address	CHARACTER VARYING(50)
address2	CHARACTER VARYING(50)
district	CHARACTER VARYING(20)
city_id	SMALLINT
postal_code	CHARACTER VARYING(10)
phone	CHARACTER VARYING(20)
last_update	TIMESTAMP(6) WITHOUT TIME ZONE

city	
city_id	SERIAL
city	CHARACTER VARYING(50)
country_id	SMALLINT
last_update	TIMESTAMP(6) WITHOUT TIME ZONE

country	
country_id	SERIAL
country	CHARACTER VARYING(50)
last_update	TIMESTAMP(6) WITHOUT TIME ZONE



Fact tables

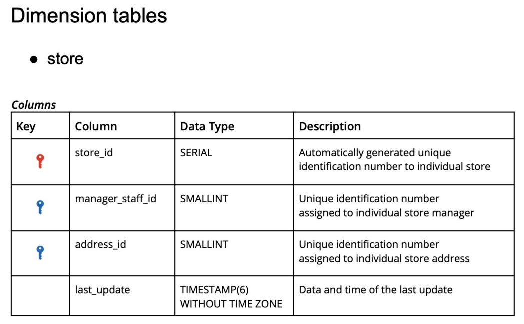
- payment

Columns

Key	Column	Data Type	Description
🔑	payment_id	SERIAL	Automatically generated unique identification number to individual payment
🔑	customer_id	SMALLINT	Unique identification number assigned to individual customer
🔑	staff_id	SMALLINT	Unique identification number assigned to individual staff member
🔑	rental_id	INTEGER	Unique identification number assigned to individual rental
	amount	TIMESTAMP(6) WITHOUT TIME ZONE	Total amount paid
	payment_date	TIMESTAMP(6) WITHOUT TIME ZONE	Autogenerated latest updated date

Links to

Key	Table	Join
→	customer	payment.customer_id = customer.customer_id
→	staff	payment.staff_id = staff.staff_id
→	rental	payment.rental_id = rental.rental_id



Dimension tables

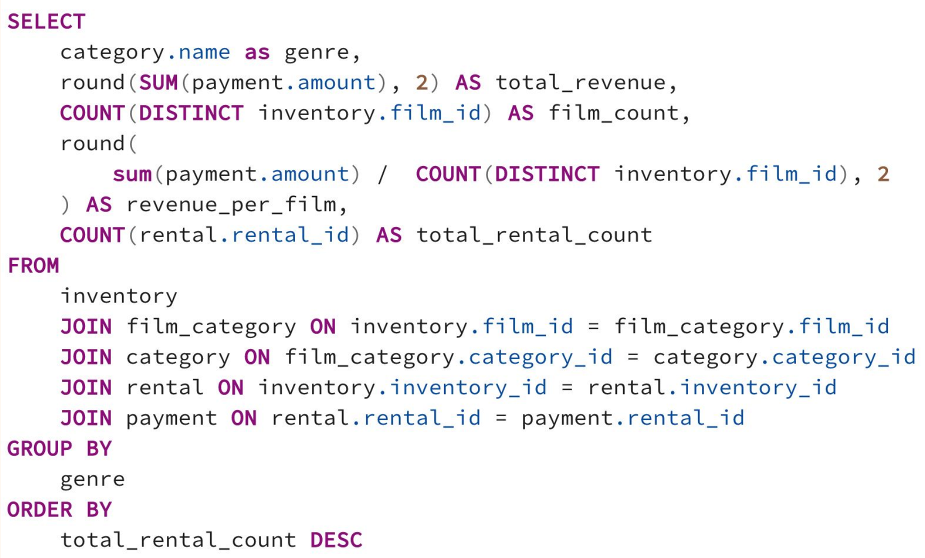
- store

Columns

Key	Column	Data Type	Description
🔑	store_id	SERIAL	Automatically generated unique identification number to individual store
🔑	manager_staff_id	SMALLINT	Unique identification number assigned to individual store manager
🔑	address_id	SMALLINT	Unique identification number assigned to individual store address
	last_update	TIMESTAMP(6) WITHOUT TIME ZONE	Data and time of the last update

2. Order, group & filter data

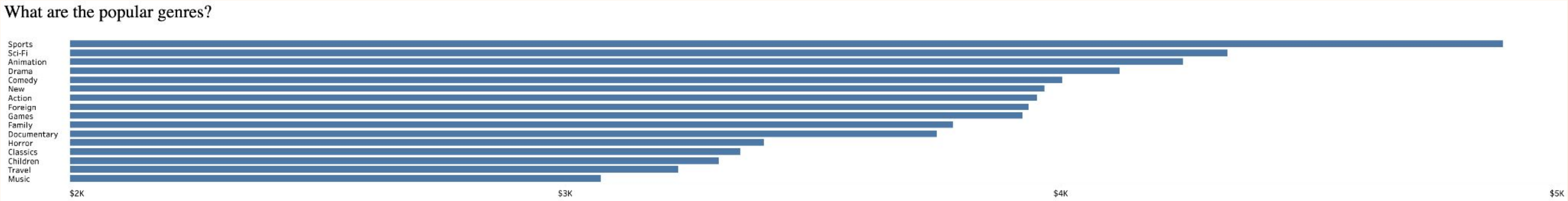
- Derived key variables such as popular ratings and genres, and customer count to see how they contributed to revenue.
- Summarized, aggregated data and performed joins using queries.



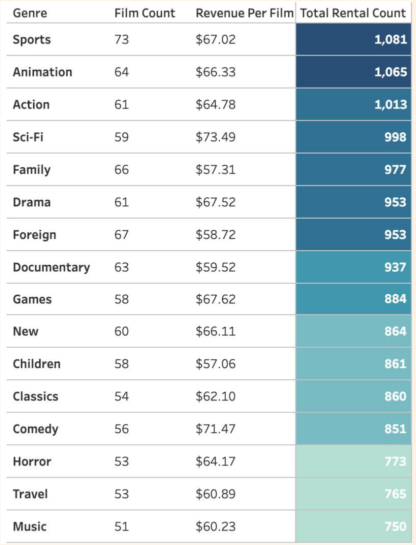
```sql
SELECT
    category.name as genre,
    round(SUM(payment.amount), 2) AS total_revenue,
    COUNT(DISTINCT inventory.film_id) AS film_count,
    round(
        sum(payment.amount) /  COUNT(DISTINCT inventory.film_id), 2
    ) AS revenue_per_film,
    COUNT(rental.rental_id) AS total_rental_count
FROM
    inventory
    JOIN film_category ON inventory.film_id = film_category.film_id
    JOIN category ON film_category.category_id = category.category_id
    JOIN rental ON inventory.inventory_id = rental.inventory_id
    JOIN payment ON rental.rental_id = payment.rental_id
GROUP BY
    genre
ORDER BY
    total_rental_count DESC
```

Query example to extract the most popular genres.

- Created a table consisting of genre-wise total rental count, to check the most popular genre in terms of the number of times being rented.



What are the popular genres?



Genre	Film Count	Revenue Per Film	Total Rental Count
Sports	73	$67.02	1,081
Animation	64	$66.33	1,065
Action	61	$64.78	1,013
Sci-Fi	59	$73.49	998
Family	66	$57.31	977
Drama	61	$67.52	953
Foreign	67	$58.72	953
Documentary	63	$59.52	937
Games	58	$67.62	884
New	60	$66.11	864
Children	58	$57.06	861
Classics	54	$62.10	860
Comedy	56	$71.47	851
Horror	53	$64.17	773
Travel	53	$60.89	765
Music	51	$60.23	750

- Sports is the most popular genre in terms of the number of times being rented, and it also has the highest total sales revenue.

- Animation and sci-fi are the other top genres in terms of their film rental count, total revenue and revenue per film.

- Despite being the third highest rented genre, action comes in at a surprising 7th position in terms of total revenue.

- A very similar trend is observed in family, foreign and documentary genres too. These genres have been rented a greater number of times, but generate a relatively low sales revenue.

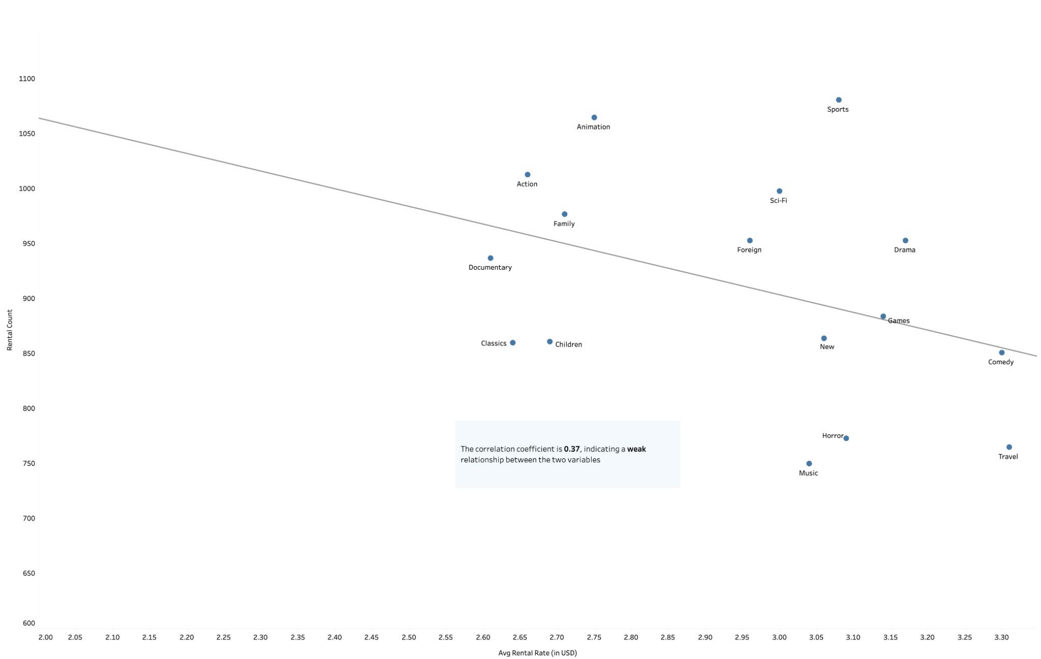
- It is interesting to note that although foreign and drama have been rented for the same number of times, their total revenue varies quite a bit.

- Oddly, only one film is categorized as thriller, an extreme deviation from the average of 56 films for every other genre.

- Being an extreme outlier, thriller is thus excluded from the analysis.

- Went ahead to see if the number of times a film has been rented has anything to do with the average rental rate of each genre, with the help of a scatter plot.

Avg Rental Price and Rental Count for a genre: Is there a relationship?



The correlation coefficient is **0.37**, indicating a **weak** relationship between the two variables

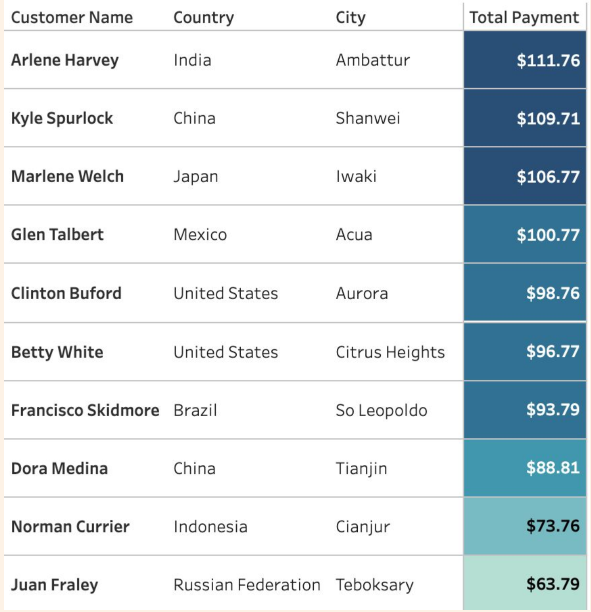
- Surprisingly, music is not the most expensive genre considering it is the least rented after thriller, although thriller is an exception with only 1 film which is again relatively less expensive from other genres.

- Foreign and drama have been rented for the same number of times but they do not share the same price.

- With a very weak relationship between the two variables, it can be confirmed that average rental rate may not be a factor.

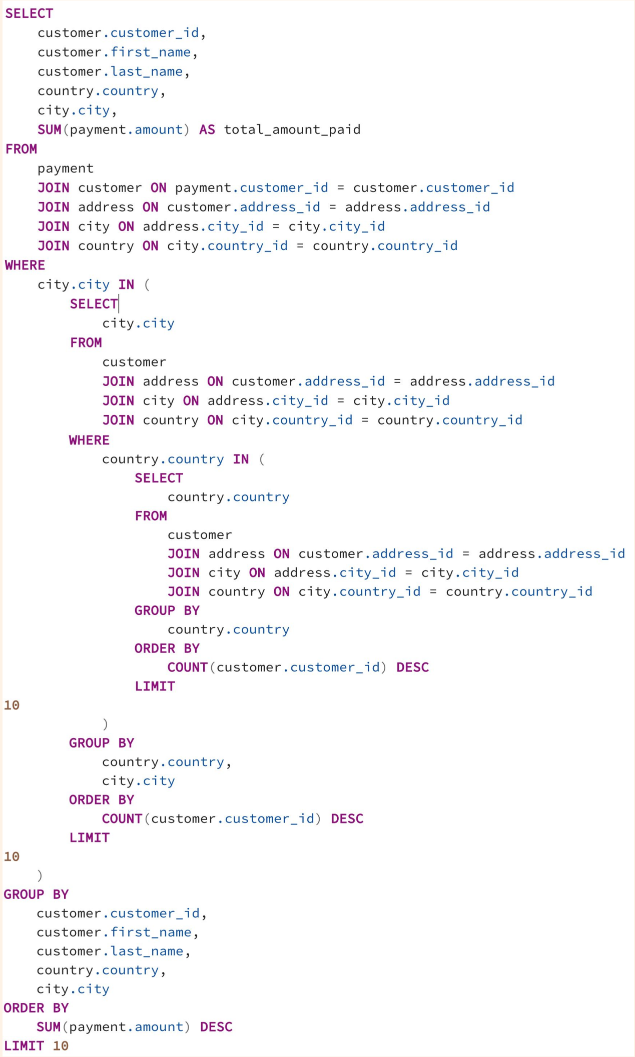
3. Table joins and subqueries

- Joined multiple tables by filtering and summarizing data, and used subqueries to answer complex business questions.
- For example, determined top 10 customers who have paid the highest total amounts to Rockbuster from the top 10 cities within the top 10 countries in terms of customer numbers to reward them for their loyalty.



Customer Name	Country	City	Total Payment
Arlene Harvey	India	Ambattur	**$111.76**
Kyle Spurlock	China	Shanwei	**$109.71**
Marlene Welch	Japan	Iwaki	**$106.77**
Glen Talbert	Mexico	Acua	**$100.77**
Clinton Buford	United States	Aurora	**$98.76**
Betty White	United States	Citrus Heights	**$96.77**
Francisco Skidmore	Brazil	So Leopoldo	**$93.79**
Dora Medina	China	Tianjin	**$88.81**
Norman Currier	Indonesia	Cianjur	**$73.76**
Juan Fraley	Russian Federation	Teboksary	**$63.79**

- Created and added multiple subqueries within the WHERE clause of SELECT statements.
    - The first subquery returned the top 10 countries in terms of customer numbers. Grouped the results on basis of country, in descending order (highest to the lowest count of customers in each country), and limited the result set to 10 rows.
    - The second subquery filtered out top city based on those results respectively. Grouped the results on basis of country followed by city, in descending order (highest to the lowest count of customers in each city of those countries), and limited the result set to 10 rows.
    - The main query filtered top customer with highest total spending from the top cities respectively. Ordered the results in descending order of total amount spent for each customer from the top cities of the top countries, and limited the result set to 10 rows.



```
SELECT
    customer.customer_id,
    customer.first_name,
    customer.last_name,
    country.country,
    city.city,
    SUM(payment.amount) AS total_amount_paid
FROM
    payment
    JOIN customer ON payment.customer_id = customer.customer_id
    JOIN address ON customer.address_id = address.address_id
    JOIN city ON address.city_id = city.city_id
    JOIN country ON city.country_id = country.country_id
WHERE
    city.city IN (
        SELECT
            city.city
        FROM
            customer
            JOIN address ON customer.address_id = address.address_id
            JOIN city ON address.city_id = city.city_id
            JOIN country ON city.country_id = country.country_id
        WHERE
            country.country IN (
                SELECT
                    country.country
                FROM
                    customer
                    JOIN address ON customer.address_id = address.address_id
                    JOIN city ON address.city_id = city.city_id
                    JOIN country ON city.country_id = country.country_id
                GROUP BY
                    country.country
                ORDER BY
                    COUNT(customer.customer_id) DESC
                LIMIT
10
            )
        GROUP BY
            country.country,
            city.city
        ORDER BY
            COUNT(customer.customer_id) DESC
        LIMIT
10
    )
GROUP BY
    customer.customer_id,
    customer.first_name,
    customer.last_name,
    country.country,
    city.city
ORDER BY
    SUM(payment.amount) DESC
LIMIT 10
```

Conclusions & Recommendations

- The average rental duration of a film is 5 days.
- The company has a sports-loving customer base and it would be advisable to stock more sports-related films to increase total sales compared to unpopular film genres like music, travel, horror, classics and children.
- Consider increasing the rental rates of popular genres such as sports, animation, action, sci-fi and family since price is not a major factor in renting for the customers.
- The top 5 countries in terms of revenue and customer base are India, China, USA, Japan and Mexico. Prioritize countries with larger customer base first. Roll out the online rental service in the top 10 markets initially based on its number of customers to test the water, before expanding globally.
- Offer movies in foreign languages as well to diversify and broaden Rockbuster's subscriber base.
- Introduce a reward scheme through vouchers, points, discounts or privileged benefits to highly valued customers. This will present more opportunities to create revenue and delight customers.

Final Project Deliverables

- Tableau Storyboard
- GitHub

Reflections

This project was my favorite since I absolutely love working with SQL. What I like the most with SQL is its efficiency to manipulate data. In addition to that, I quickly learned that SQL queries are repeatable and scalable. I can write a query once and then reuse it again and again. In fact, even if new tables are added, I can use the previous query and make a few changes that shall work just fine. It was fairly simple to learn SQL yet it could solve incredibly challenging problems. Rockbuster's database had large sets of data in the form of numerous tables but SQL queries made pulling complex information simultaneously from multiple tables fast and easy.