

RentalSight: An Online Movie Rental Warehouse

Introduction

In the dynamic landscape of online movie rentals, efficient data management is pivotal for success. Introducing Rentalsight, a comprehensive data warehouse crafted to empower online movie rental stores with insightful analytics, seamless inventory management, and strategic rental oversight.

Deliverables

Revenue Trend Analysis:

- Explore revenue trends over specified time periods.
- Visualize revenue patterns based on movie genres, or ratings.
- Identify peak revenue periods and potential areas for revenue growth.

Inventory Management:

- Track inventory levels in real-time.
- Identify slow-moving or obsolete inventory for strategic decision-making.
- Measure the efficiency of inventory management and return policy

Rental Oversight:

- Analyze rental patterns.
- Optimize rental pricing based on demand and historical data.

User-Friendly Interface:

- Intuitive dashboards for easy navigation and quick access to key metrics.
- Customizable reports to tailor insights to specific business requirements.
- Real-time data updates for timely decision-making.

ETL phase

Extraction:

Data was extracted from an online dataset comprised of many individual tables in csv and JSON formats.

Original data can be found in source_data folder.

Transformation:

Data is then transformed using Python and the Pandas library.

- The following files were used to individually filter out and transform certain data and save them into csv files located in the main folder: [transform_address.py](#), [transform_film.py](#), [transform_customer.csv](#), [transform_rental.csv](#), [transform_store.py](#).
- The files [create_inventory_snapshot.py](#) and [create_rental_transactions.py](#) are used to create the two fact tables that will be employed in our analysis later on.
- the [ETL.ipynb](#) is a jupyter notebook created to execute all transformations, yielding the following csv files found in the main folder: [address.csv](#), [customer.csv](#), [store.csv](#), [film.csv](#), [rental.csv](#), [day.csv](#), [year.csv](#), [month.csv](#), [rental_transactions.csv](#), [inventory_snapshot.csv](#)

Loading:

Data is then loaded onto a snowflake ROLAP database.

This database can be accessed through this link:

<https://vn58684.west-europe.azure.snowflakecomputing.com>

username: slimboyfu

password: Testingtesting1\$

MM mahdi messaai ACCOUNTADMIN

Worksheets

Dashboards

Streamlit

Apps

Data

Databases

Private Sharing

Provider Studio

Marketplace

Activity

Admin

Help & Support

27 days left in trial

Upgrade

Search

MOVIE_RENTAL_STORE

INFORMATION_SCHEMA

PUBLIC

Tables

ADDRESS

CUSTOMER

DAY

FILM

INVENTORY_SNAPSHOT

MONTH

RENTAL

RENTAL_TRANSACTIONS

STORE

YEAR

SNOWFLAKE

MOVIE_RENTAL_STORE / PUBLIC

Schema ACCOUNTADMIN 1 day ago

Create

Schema Details

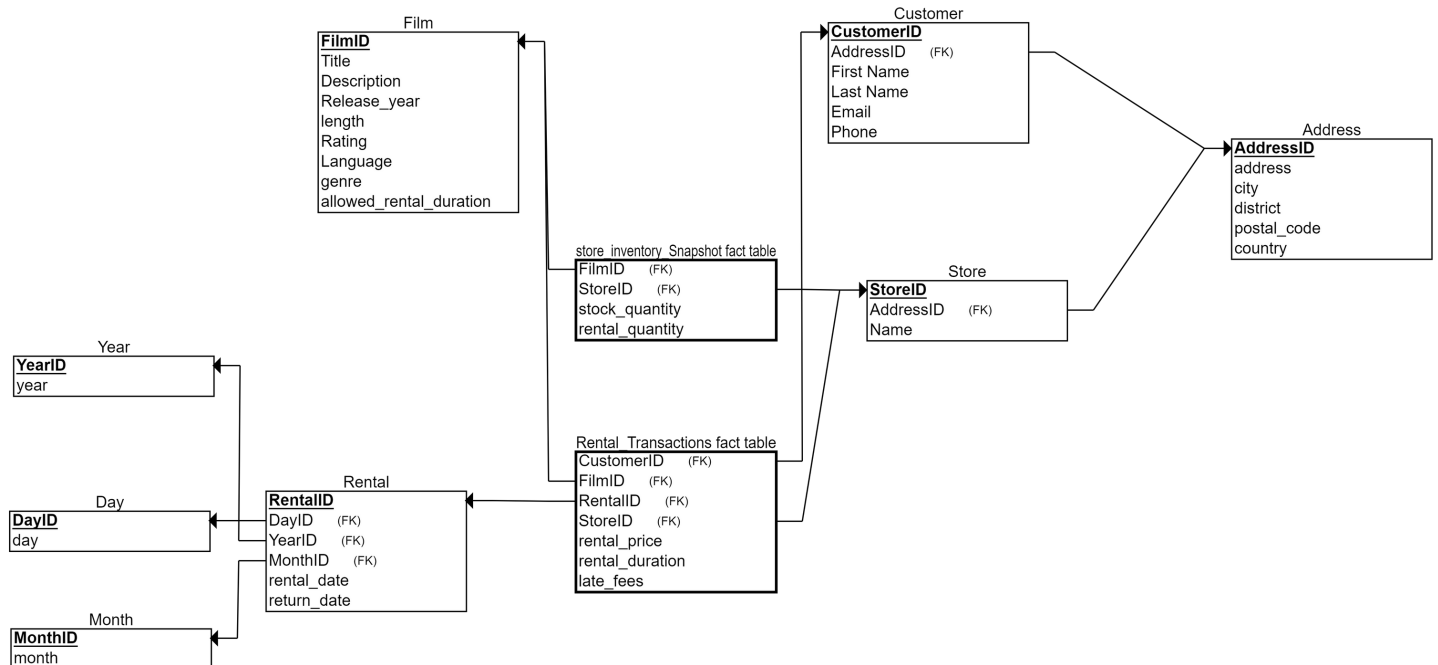
Tables

10 Tables

Search All Tables

NAME	TYPE	OWNER	ROWS	BYTES	CREATED
ADDRESS	Table	ACCOUNTADMIN	603	23.5KB	1 day ago
CUSTOMER	Table	ACCOUNTADMIN	599	24.5KB	1 day ago
DAY	Table	ACCOUNTADMIN	29	1.0KB	1 day ago
FILM	Table	ACCOUNTADMIN	1K	43.5KB	1 day ago
INVENTORY_SNAPSH...	Table	ACCOUNTADMIN	1.5K	5.0KB	1 day ago
MONTH	Table	ACCOUNTADMIN	5	1.5KB	1 day ago
RENTAL	Table	ACCOUNTADMIN	16.0K	256.0KB	1 day ago
RENTAL_TRANSACTIO...	Table	ACCOUNTADMIN	16.0K	100.5KB	1 day ago
STORE	Table	ACCOUNTADMIN	2	1.5KB	1 day ago
YEAR	Table	ACCOUNTADMIN	1	1.0KB	1 day ago

Data Warehousing



Store_inventory_snapshot: used to store historical data on stock and rented quantities by store and movie.

Rental_Transactions: used to store historical data on all transactions occurring in all stores. Its measures are the following:

- **rental_price:** price to rent a certain movie.
- **rental_duration:** actual rental duration of a movie by a customer

- `late_fees`: any incurred late return fees. (By default, 1\$ is charged to the customer for every 1 day delay)

We have also identified the following dimensions:

- Film
- Rental
- Year
- Day
- Month
- Customer
- Store
- Address



Analysis

Analysis was done using Power BI.

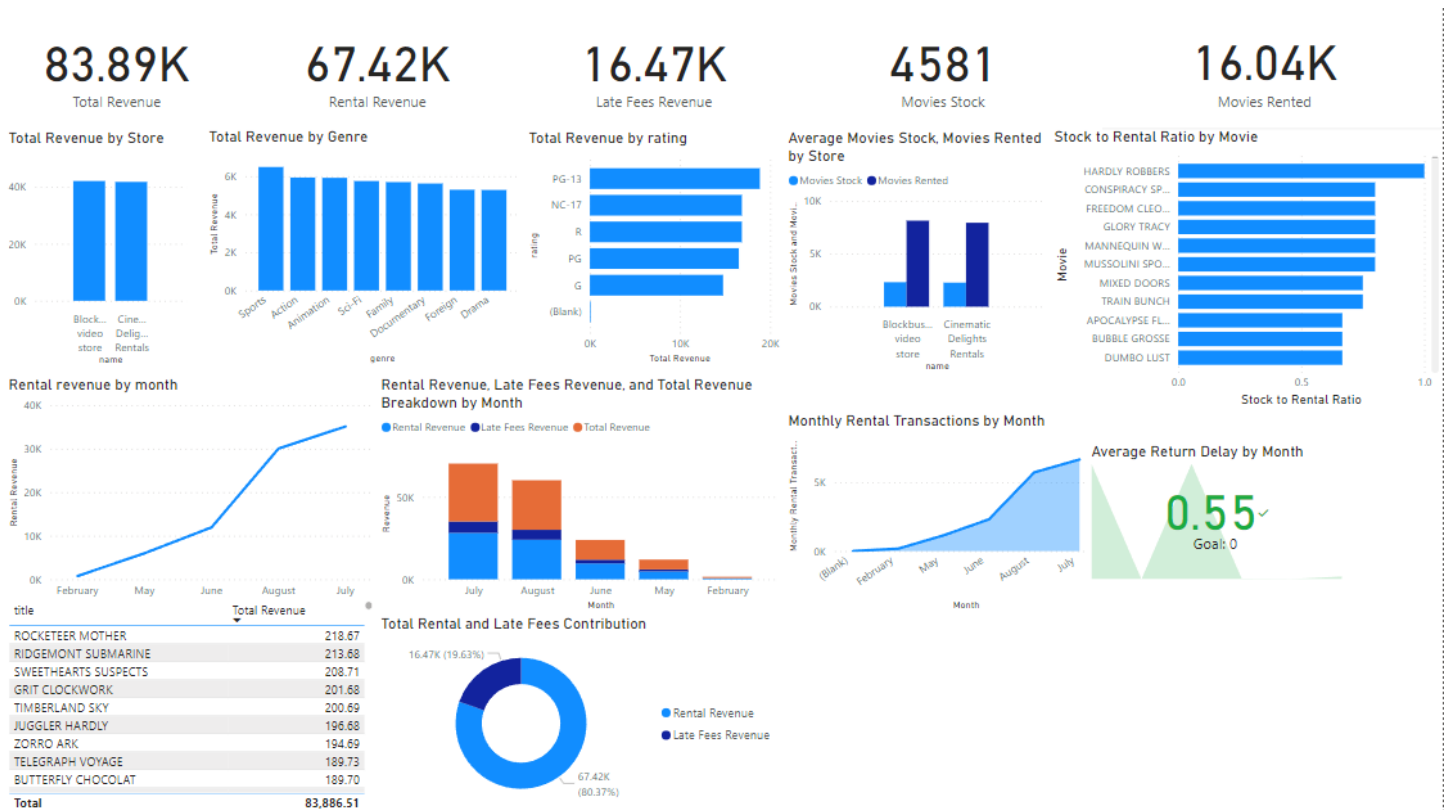
Two PowerBI files are included in the main folder: `snowflake_analysis` and `local_analysis`.

The two are identical. The only difference is `snowflake_analysis` gets its data from snowflake directly. That requires authentication or else it would fail to load any visualizations.

`local_analysis` uses the local files in the folder. This one should be used if you wish not to connect to snowflake.

 <code>local_analysis</code>	1/21/2024 9:33 PM	Microsoft Power B...	2,777 KB
 <code>snowflake_analysis</code>	1/21/2024 2:43 PM	Microsoft Power B...	2,048 KB

Upon loading up `local_analysis.pbix`. You are introduced to the dashboard.



Conclusion

Few things to note about the project:

- Data was not of the highest quality as can be seen from the visualizations. Barely any difference between the two stores. That made for some rather disappointing visualizations.
- Project could have been much more informative had I added more data and probably added a couple more dimensions.
- Working on this alone was a bit tough
- More complex and interesting KPIs were supposed to be done but were cut out due to lack of time/expertise.