🎬 **Proyecto SQL: Análisis de base de datos de una tienda de películas**

📖 **Descripción**

Este proyecto forma parte del módulo de SQL en el cual trabajamos una base de datos ficticia que simula el funcionamiento de una tienda de alquiler de películas. El objetivo principal fue diseñar y ejecutar una variedad de consultas SQL que nos permitieran explorar y analizar la información contenida en la base de datos. Las consultas se realizan sobre diferentes aspectos del negocio: desde datos generales sobre películas, clientes y actores, hasta métricas estadísticas, combinaciones avanzadas y creación de vistas o tablas temporales.

La base de datos contiene múltiples tablas como `film`, `actor`, `category`, `customer`, `inventory`, `rental`, entre otras, y su diseño normalizado requiere utilizar distintos tipos de uniones (`JOIN`) para realizar análisis más completos.

A lo largo del proyecto se aplicaron conocimientos de:

* Filtrado y ordenamiento (`WHERE`, `ORDER BY`).
* Funciones de agregación (`AVG`, `SUM`, `COUNT`, `VARIANCE`, `STDDEV`).
* Agrupaciones (`GROUP BY`, `HAVING`).
* Subconsultas y vistas.
* Diferentes tipos de uniones ‘joins’ (INNER, LEFT, RIGHT, CROSS).
* Creación de tablas temporales.

Todo el análisis se realizó utilizando PostgreSQL como motor de base de datos y DBeaver como entorno de trabajo visual para escribir, ejecutar y analizar las consultas.

📁 **Estructura del proyecto**

La estructura de archivos y carpetas utilizada en este proyecto es la siguiente:

├consultas\_proyecto\_SQL.sql # Archivo que contiene todas las consultas SQL resueltas del análisis

├ analisis\_consultas.docx # Documento explicativo paso a paso de cada consulta realizada

├ esquema\_base\_datos.png # Imagen con el esquema entidad-relación de la base de datos

├ README.md # Descripción completa del proyecto (este archivo)

🛠 **Instalación y Requisitos**

Para poder ejecutar y analizar correctamente este proyecto hay que tener instalado y tener en cuenta los siguientes requisitos:

**Instrucciones de instalación:**

1. Instalación de PostgreSQL y tener un servidor local activo.

2. Abrir DBeaver y crear una conexión nueva con la base de datos.

3. Cargar la base de datos para ejecutar las consultas del proyecto.

4. Crear el esquema de entidad-relación de la base de datos, siguiendo los siguientes pasos: Postgres – Bases de datos – postgres – esquemas – botón derecho en public – view diagram.

**Requisitos:**

* PostgreSQL (como motor de la base de datos)
* DBeaver (como herramienta de administración y visualización)
* Acceso a la base de datos

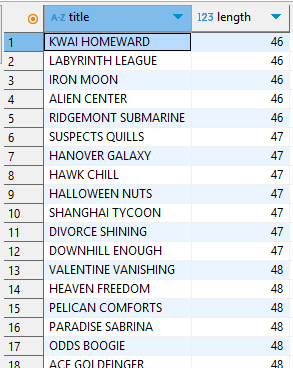
📊 **Resultados y Conclusiones**

Durante el desarrollo de este proyecto se han ejecutado 64 consultas SQL que cubren distintos aspectos del análisis de comportamiento de la tienda de películas. Las diferentes consultas tratan sobre las siguientes categorías:

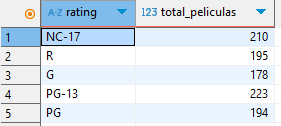
* Películas de la tienda
* Actores de las películas
* Clientes de películas
* Alquileres de películas
* Dinero (ingresos) de la tienda
* Trabajadores de la tienda

A continuación, mostraré un resumen de los hallazgos más importantes:

* **Películas:**
  + Se obtuvo la ordenación por duración de forma ascendente de todas las películas de la tienda.



* + Se identificó la cantidad total de películas y su clasificación.



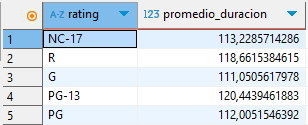
* + Se identificó la variabilidad de lo que costaría reemplazar las películas.



* + Se identificó la mayor y menor duración de las películas.



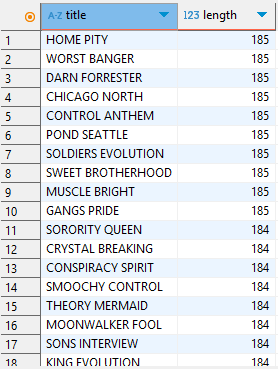
* + Se obtuvo la clasificación y promedio de duración de las películas.



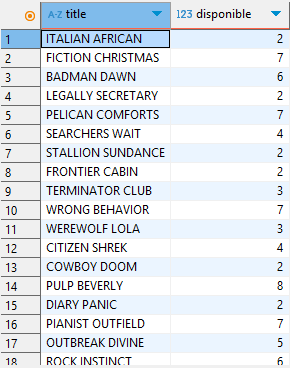
* + Se obtuvieron los nombres de todas las películas únicas.



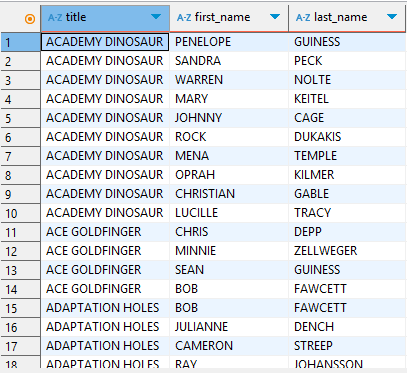
* + Se obtuvo la lista de películas con una duración superior al promedio.



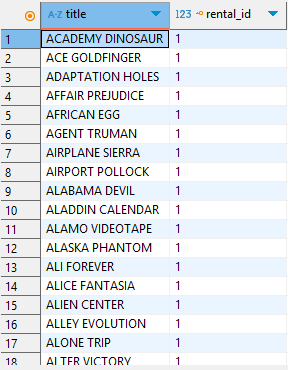
* + Se obtuvo la lista y cantidad de películas disponibles en el inventario.



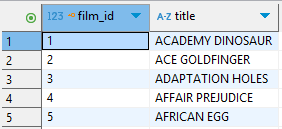
* + Se identificó el total de películas y actores que han actuado en ellas.



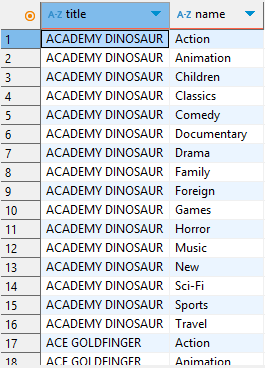
* + Se identificó el total de películas y registros de alquiler de las mismas.



* + Se obtuvieron los nombres de las 5 primeras películas.



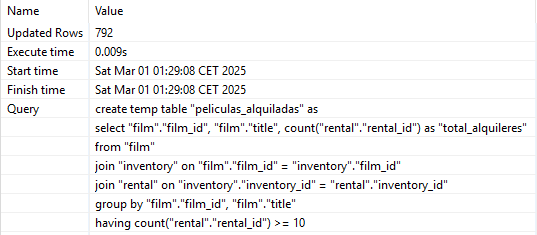
* + Se obtuvo el resultado de la unión cruzada entre las tablas de film y category mediante CROSS JOIN.



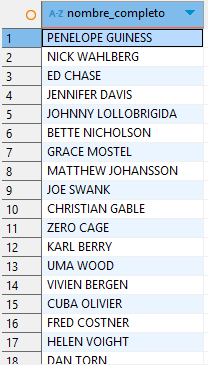
* + Se obtuvo la duración total de las películas de la categoría ‘Action’.



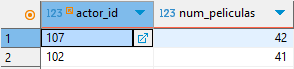
* + Se creó una tabla temporal “peliculas\_alquiladas” para almacenar las películas que han sido alquiladas al menos 10 veces.



* **Actores:**
  + Se identificaron los nombres y apellidos de todos los actores y actrices de las películas.



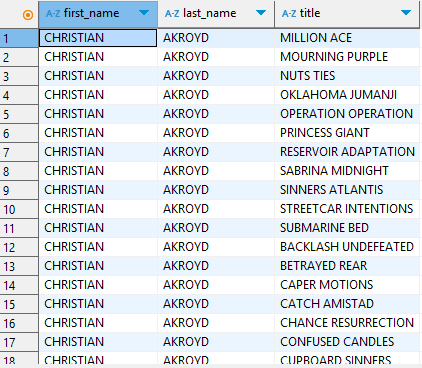
* + Se obtuvieron los ID de los actores que han participado en más de 40 películas.



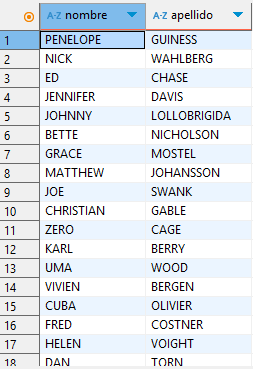
* + Se obtuvieron los ID del actor más bajo y del actor más alto.



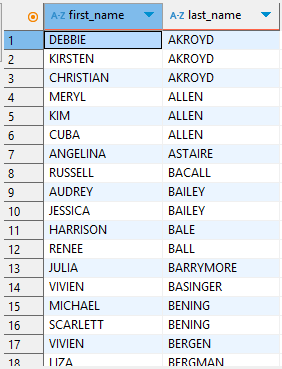
* + Se identificaron el total de actores y películas en las que han actuado.



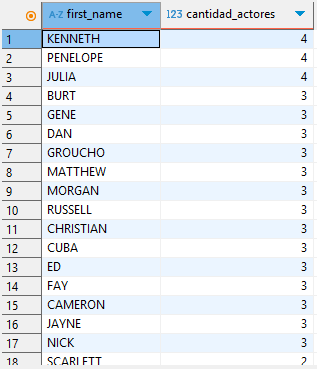
* + Se renombró la columna “first\_name” como Nombre y “last\_name” como Apellido.



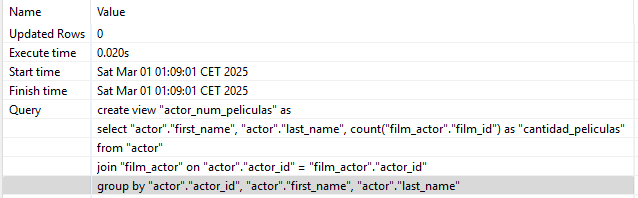
* + Se ordenaron todos los actores por apellido de forma ascendente.



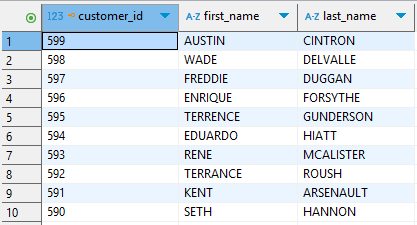
* + Se ordenó la lista de actores por nombre y cantidad.



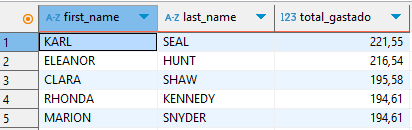
* + Se creó una vista para mostrar los nombres de los actores y el número de películas en las que han participado.



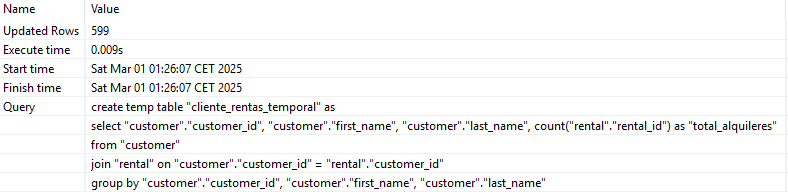
* **Clientes:**
  + Se obtuvieron los 10 clientes con mayor valor de ID.



* + Se obtuvieron los 5 clientes que más dinero se han gastado.



* + Se ha creado una tabla temporal para almacenar el total de alquileres por cliente.



* + Se ha conocido los nombres de los clientes que han alquilado al menos 7 películas distintas, ordenados alfabéticamente por apellido.



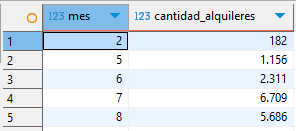
* **Alquileres:**
  + Se conoció el coste del antepenúltimo alquiler del día.



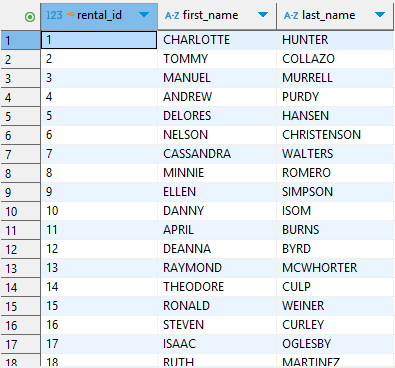
* + Se conoció la media de duración del alquiler de las películas.



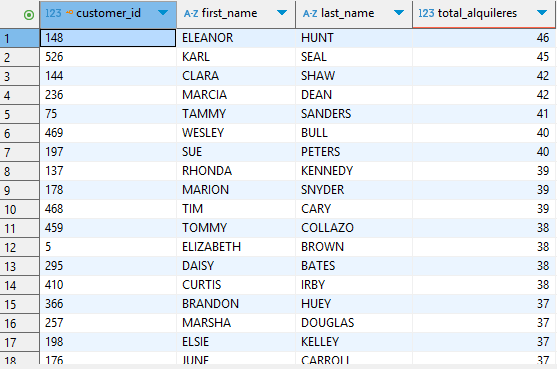
* + Se conoció el número de alquileres por mes.



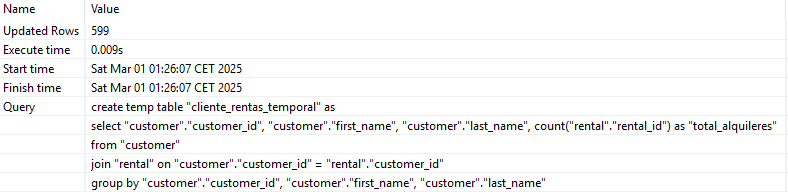
* + Se obtuvieron la lista de alquileres de películas y los nombres de los clientes que los realizaron.



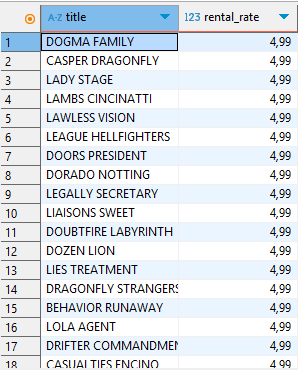
* + Se obtuvo el listado de los alquileres de películas de los clientes (total).



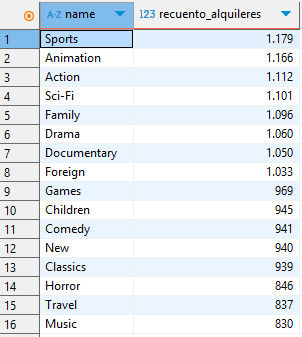
* + Se creó una tabla con el número total de alquileres realizados por cada cliente.



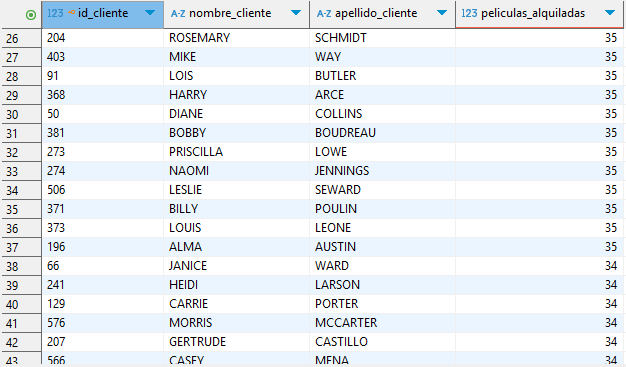
* + Se conoció la lista de películas que se alquilan por encima del precio medio.



* + Se obtuvo la cantidad total de películas alquiladas por categoría y nombre de la categoría junto con el recuento de alquileres.



* + Se obtuvo la cantidad total de películas alquiladas por cada cliente e ID del cliente, su nombre y apellido.



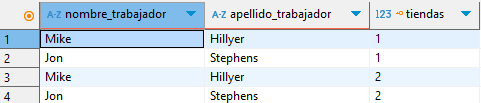
* **Dinero:**
* Se conocieron los ingresos totales de la empresa.



* Se obtuvieron el promedio, desviación estándar y varianza del total pagado.



* **Trabajadores:**
* Se obtuvieron todas las combinaciones posibles de trabajadores con las tiendas que hay



🔄 **Próximos Pasos**

Este proyecto tiene aún mucho potencial para seguir creciendo. Algunas ideas para futuras mejoras podrían ser:

* Añadir una automatización de consultas y reportes mediante programación en PostgreSQL.
* Conectar los resultados con herramientas de visualización como Tableau o Power BI, lo que permitiría presentar los datos de forma más intuitiva y dinámica.
* Implementar procesos de control de calidad sobre los datos, como la detección de valores nulos, duplicados o inconsistentes.
* Optimizar las consultas más pesadas mediante el uso de índices en columnas clave para mejorar el rendimiento general.

🤝 **Contribuciones**

Si tienes ideas para ampliar este proyecto o quieres hacer alguna mejora, estaré encantado de recibir tu colaboración. Puedes abrir una *issue* o enviar un *pull request* con tus sugerencias o cambios.

✒ **Autores y Agradecimientos**

Autor: Francisco Troyano Martínez.

Contacto: [troyano1406@gmail.com](mailto:troyano1406@gmail.com) .

GitHub: <https://github.com/trhoudini>

Proyecto realizado como parte del módulo de SQL en el programa de formación de Data Analytics.