Informe prueba técnica QUIND

Presentado por Julio César Salazar Taborda.

Explicación de la Arquitectura de Datos y Arquetipo de la Aplicación

La arquitectura de datos la aplicación se puede describir de la siguiente manera:

Arquitectura de Datos:

- **Fuente de Datos:** Los datos se obtienen del archivo Excel Films2(4).xlsx que contiene múltiples hojas con datos de películas, clientes, alquileres, tiendas, entre otros.
- **Proceso ETL (Extracción, Transformación y Carga):**
    - **Extracción:** Los datos son cargados desde el archivo Excel (Films2(4).xlsx).
    - **Transformación:** Los datos son limpiados (ya que se encontraron algunas inconsistencias, por ejemplo los títulos de las columnas, muchos traían un espacio antes de los *strings*, confundiendo el programa y también algunos datos de la hoja Film los cuales eran de carácter numérico venían corruptos con caracteres, se decide no imputarlos como *NaN* si no recorrerlos y dejarles solo su parte numérica) y transformados.
    - **Carga:** Los datos transformados se guardan en formato *Parquet* este ofrece ventajas significativas en términos de eficiencia de almacenamiento, rendimiento de consultas y compatibilidad con herramientas de Big Data. Estas características lo convierten en una opción ideal para almacenar y procesar grandes volúmenes de datos de manera eficiente y efectiva.
- **Almacenamiento:** Los archivos *Parquet* se almacenan en la carpeta que se crea durante el programa y se encuentra en el *path* "/content/data/processed".

Arquetipo de la Aplicación:

- **Herramientas Utilizadas:**
    - **Pandas:** Para manipulación y análisis de datos en Python.
    - **PySpark:** Para procesamiento y transformación de datos a gran escala.
    - **Google Colab:** Como entorno de ejecución para el código.
    - **Seaborn y Matplotlib:** como visualizadores de gráficos y data.
- **Flujo de Trabajo:**
    - **Ingesta de Datos:** Los datos se cargan desde el archivo Excel.
    - **Extracción de Datos:** Los datos se leen con ayuda de la librería panda.
    - **Limpieza y Transformación:** Se limpian y transforman los datos utilizando PySpark.

- **Almacenamiento y Visualización:** Los datos transformados se almacenan en formato Parquet y se visualizan usando pandas y matplotlib.

Análisis de datos exploratorio:

En primera medida se realiza una interpretación del gráfico MER para entender un poco del contexto del problema el cual podemos resumir con las siguientes afirmaciones:

Tabla origen	**Tabla destino**	**Relación**	**Cardinalidad**
customer	rental	1 cliente puede tener muchos alquileres	1:N
store	customer	1 tienda puede tener muchos clientes	1:N
store	inventory	1 tienda puede tener muchos ítems	1:N
film	inventory	1 película puede estar en muchos ítems	1:N
inventory	rental	1 ítem puede ser alquilado muchas veces	1:N
Inventario	store	el inventario es único de cada tienda	1:1
rental	customer	la renta debe ser para uno y solo un cliente	1:1

Seguidamente, damos un recorrido por los datos, en los cuales se pudieron observar una serie de inconsistencias, o de corrupción de data, la cual se corrigió desde el código reemplazando los datos que se encontraban corrompidos.

Preguntas:

Nota: las respuestas a estás preguntas se encuentran en el código fuente para no extender demasiado este documento y para aprovechar los gráficos hechos con *matplotlib*.

1. ¿Cuál es la tienda con el mayor volumen de clientes?

2. ¿Cuál es el promedio de duración de alquiler de las películas?

3. ¿Qué porcentaje de las películas están subtituladas en más de un idioma?

4. ¿Cuántos clientes activos y no activos hay en cada tienda?

5. ¿Cuál es la clasificación de películas más común en el catálogo?

Conclusiones:

La arquitectura de datos se basa en un proceso ETL que limpia y transforma los datos para un análisis eficiente. La aplicación utiliza pandas y PySpark para la manipulación de datos y Google Colab como entorno de ejecución. Se formularon y respondieron cinco preguntas clave sobre los datos procesados. Finalmente, se puede concluir que se ha obtenido un buen programa capaz de filtrar y transformar una data dada, siendo capaz de ser modular cambiando unas cuantas líneas si así se quisiera para analizar otra base de datos.