**Ejercicio 2**

**Alquiler de automóviles**

Una de las empresas líderes en alquileres de automóviles solicita una serie de dashboards y reportes para poder basar sus decisiones en datos. Entre los indicadores mencionados se encuentran total de alquileres, segmentación por tipo de combustible, lugar, marca y modelo de automóvil, valoración de cada alquiler, etc. Como Data Engineer debe crear y automatizar el pipeline para tener como resultado los datos listos para ser visualizados y responder las preguntas de negocio.

**TAREAS**

1. Crear en hive una database car\_rental\_db y dentro una tabla llamada car\_rental\_analytics.

**Solución:**

<https://github.com/luxfo/edvai-data-engineering/blob/main/Ejercicios/TPFinal/Ejercicio2/hive_scripts.sh>

2. Crear script para el ingest de estos dos files:

<https://edvaibucket.blob.core.windows.net/data-engineer-edvai/CarRentalData.csv?sp=r&st=2023-11-06T12:52:39Z&se=2025-11-06T20:52:39Z&sv=2022-11-02&sr=c&sig=J4Ddi2c7Ep23OhQLPisbYaerlH472iigPwc1%2FkG80EM%3D>

<https://public.opendatasoft.com/api/explore/v2.1/catalog/datasets/georef-united-states-of-america-state/exports/csv?lang=en&timezone=America%2FArgentina%2FBuenos_Aires&use_labels=true&delimiter=%3B>

**Solución:**

<https://github.com/luxfo/edvai-data-engineering/blob/main/Ejercicios/TPFinal/Ejercicio2/tp2_ingest.sh>

3. Crear un script para tomar el archivo desde HDFS y hacer las siguientes transformaciones:

● En donde sea necesario, modificar los nombres de las columnas. Evitar espacios y puntos (reemplazar por \_ ). Evitar nombres de columna largos

● Redondear los float de ‘rating’ y castear a int.

● Joinear ambos files

● Eliminar los registros con rating nulo

● Cambiar mayúsculas por minúsculas en ‘fuelType’

● Excluir el estado Texas

Finalmente insertar en Hive el resultado

**Solución:**

<https://github.com/luxfo/edvai-data-engineering/blob/main/Ejercicios/TPFinal/Ejercicio2/tp2_transform.py>

4. Realizar un proceso automático en Airflow que orqueste los pipelines creados en los puntos anteriores. Crear dos tareas:

a. Un DAG padre que ingente los archivos y luego llame al DAG hijo

b. Un DAG hijo que procese la información y la cargue en Hive

**Solución:**

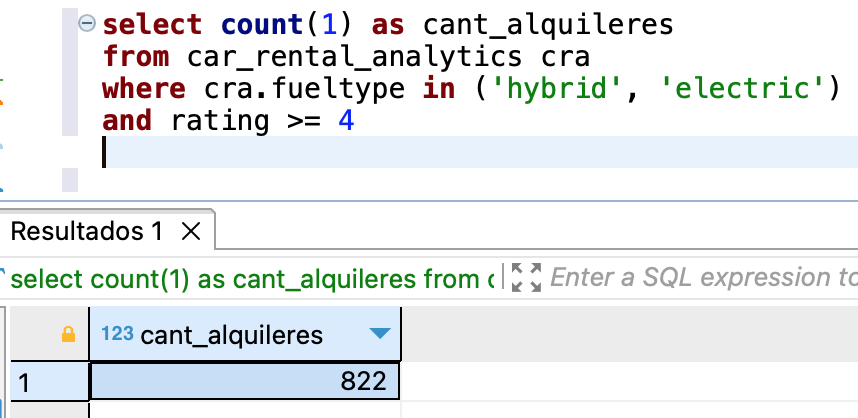
<https://github.com/luxfo/edvai-data-engineering/blob/main/Ejercicios/TPFinal/Ejercicio2/dag_tp2_parent.py>

<https://github.com/luxfo/edvai-data-engineering/blob/main/Ejercicios/TPFinal/Ejercicio2/dag_tp2_child.py>

5. Por medio de consultas SQL al data-warehouse, mostrar:

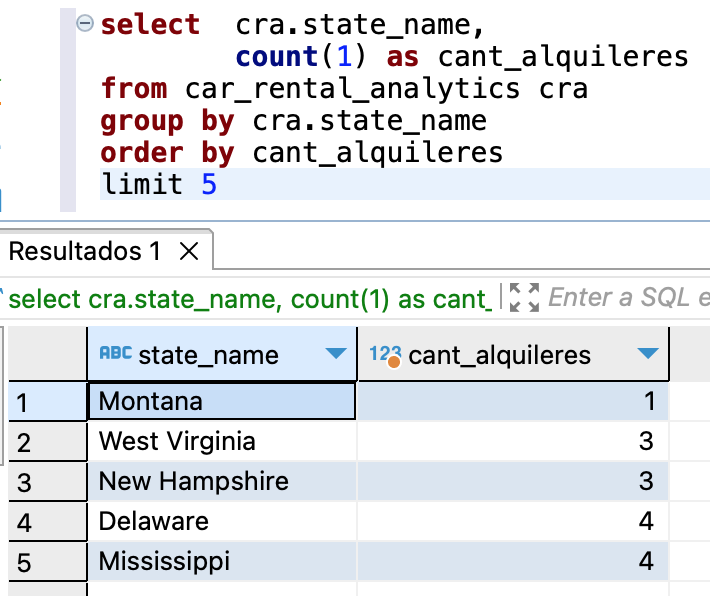
a. Cantidad de alquileres de autos, teniendo en cuenta sólo los vehículos ecológicos (fuelType hibrido o eléctrico) y con un rating de al menos 4.

**Solución:**

****

b. los 5 estados con menor cantidad de alquileres (crear visualización)

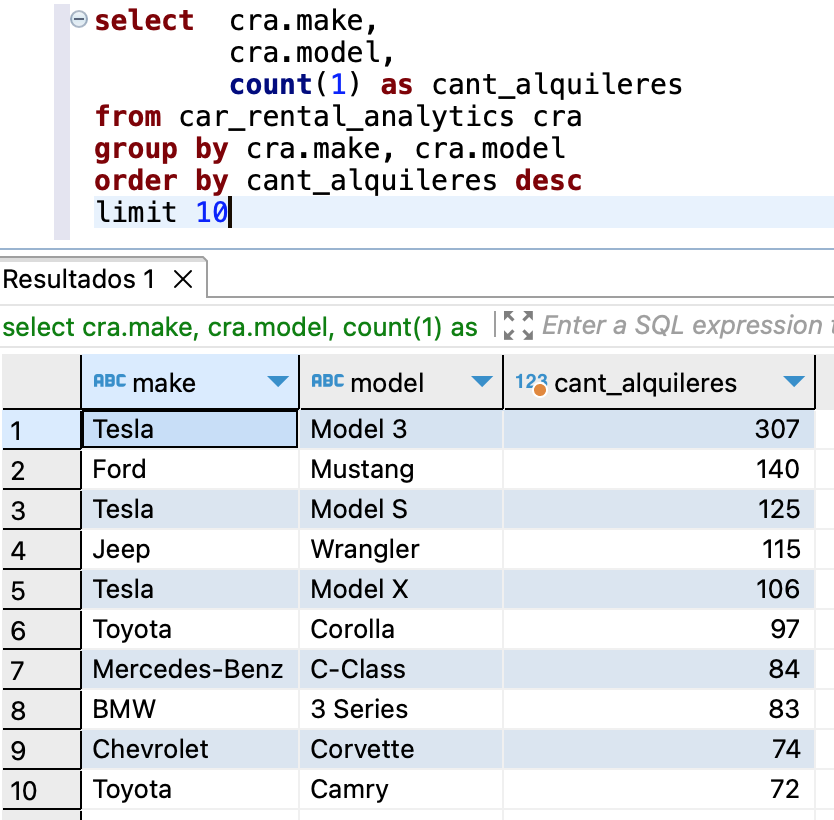
**Solución:**

****

<https://github.com/luxfo/edvai-data-engineering/blob/main/Ejercicios/TPFinal/Ejercicio2/tp2_informe.pdf>

c. los 10 modelos (junto con su marca) de autos más rentados (crear visualización)

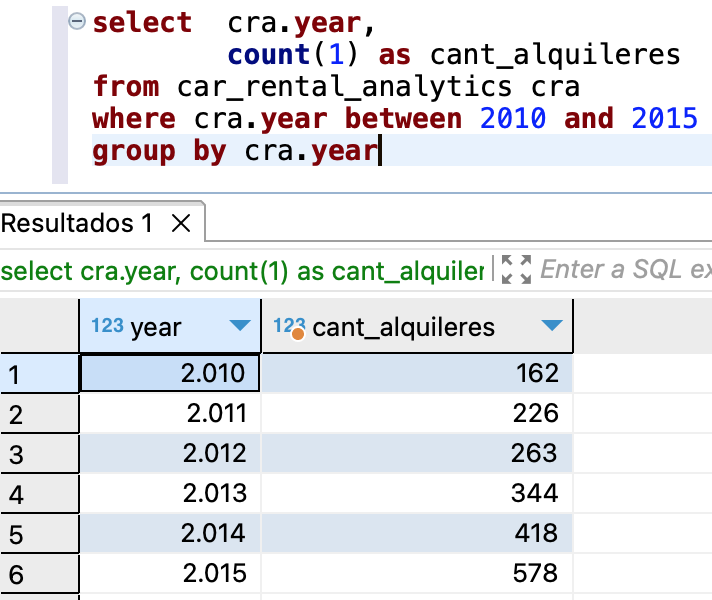
**Solución:**

****

<https://github.com/luxfo/edvai-data-engineering/blob/main/Ejercicios/TPFinal/Ejercicio2/tp2_informe.pdf>

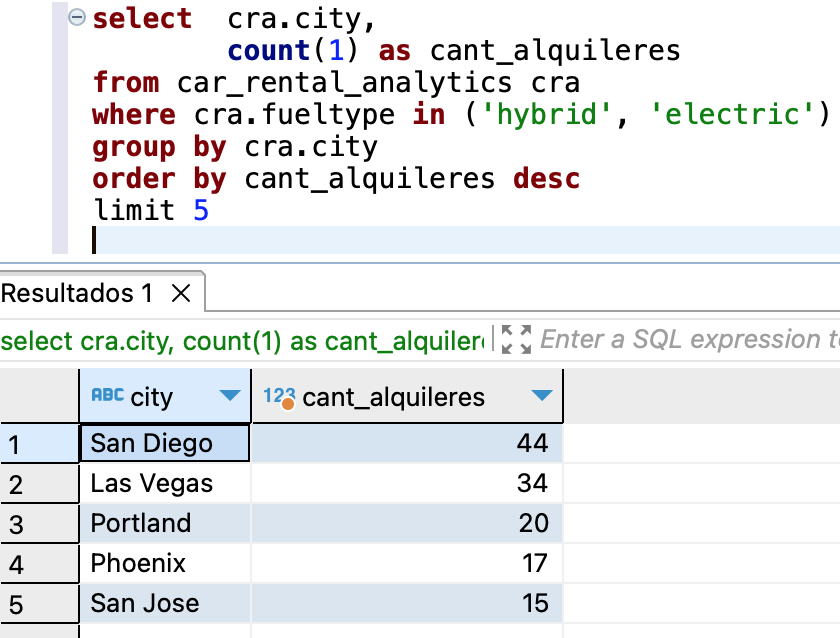
d. Mostrar por año, cuántos alquileres se hicieron, teniendo en cuenta automóviles fabricados desde 2010 a 2015

**Solución:**

****

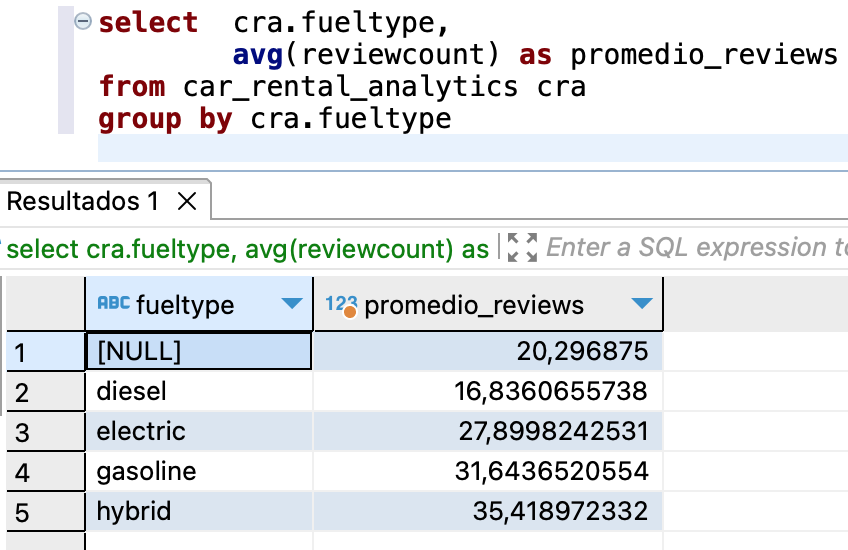
e. las 5 ciudades con más alquileres de vehículos ecológicos (fuelType hibrido o electrico)

**Solución:**

****

f. el promedio de reviews, segmentando por tipo de combustible

**Solución:**



6. Elabore sus conclusiones y recomendaciones sobre este proyecto.

**Solución:**

Para este proyecto se considera conveniente agregar los siguientes datos para obtener un mejor análisis y poder refinar la toma de decisiones.

kilometraje: Para poder identificar si un auto necesita realización de service.

Autonomía: De esta forma saber la cantidad de distancia que puede recorrer.

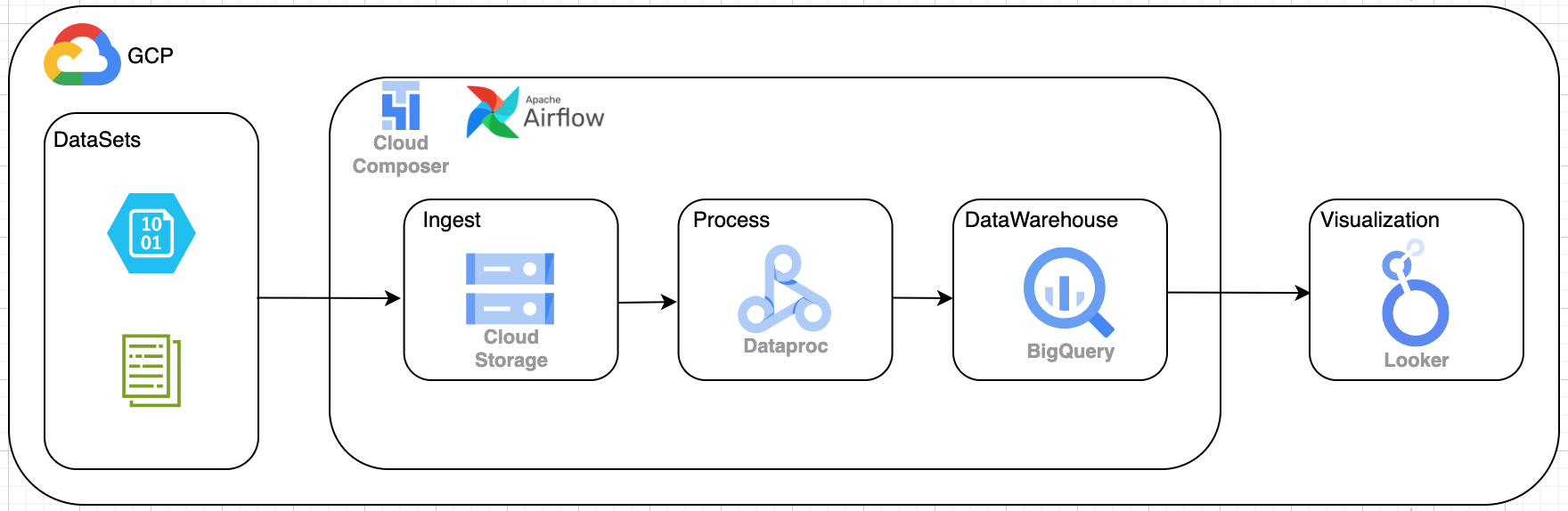
Cantidad de pasajeros: La cantidad de pasajeros permitida por el auto.

Por otro lado revisar de tener completo desde el origen los fullType que se encuentran en nulo para mejorar la calidad del dato.

También se recomienda una arquitectura cloud para tener un buen mantenimiento, seguridad, escalabilidad y abaratar costos.

7. Proponer una arquitectura alternativa para este proceso ya sea con herramientas on premise o cloud (Si aplica)

**Solución:**

****