EXAMEN FINAL EDVAI

**Tabla de Contenido**

[**EJERCICIO 1: Aviación Civil** 2](#_Toc156256652)

[1. Creando los scripts para hacer Ingest de la data en csv. Y otorgando los permisos necesarios para su ejecución 2](#_Toc156256653)

[2. Crear 2 tablas en el Data Warehouse, una para los vuelos realizados en 2021 y 2022 (2021-informe-ministerio.csv y 202206-informe-ministerio) y otra tabla para el detalle de los aeropuertos (aeropuertos\_detalle.csv) 2](#_Toc156256654)

[3. Realizar un proceso automático orquestado por Airflow 3](#_Toc156256655)

[4. Transformaciones en los pipelines de datos: 3](#_Toc156256656)

[5. Mostrar mediante una impresión de pantalla, que los tipos de campos de las tablas sean los solicitados en el datawarehouse 5](#_Toc156256657)

[6. Determinar la cantidad de vuelos entre las fechas 01/12/2021 y 31/01/2022. Mostrar consulta y Resultado de la query 6](#_Toc156256658)

[7. Cantidad de pasajeros que viajaron en Aerolíneas Argentinas entre el 01/01/2021 y 30/06/2022. Mostrar consulta y Resultado de la query 6](#_Toc156256659)

[8. Mostrar fecha, hora, código aeropuerto salida, ciudad de salida, código de aeropuerto de arribo, ciudad de arribo, y cantidad de pasajeros de cada vuelo, entre el 01/01/2022 y el 30/06/2022 ordenados por fecha de manera descendiente. Mostrar consulta y Resultado de la query 6](#_Toc156256660)

[9. Cuáles son las 10 aerolíneas que más pasajeros llevaron entre el 01/01/2021 y el 30/06/2022 exceptuando aquellas aerolíneas que no tengan nombre. Mostrar consulta y Visualización 7](#_Toc156256661)

[10. Cuáles son las 10 aeronaves más utilizadas entre el 01/01/2021 y el 30/06/22 que despegaron desde la Ciudad autónoma de Buenos Aires o de Buenos Aires, exceptuando aquellas aeronaves que no cuentan con nombre. Mostrar consulta y Visualización 7](#_Toc156256662)

[11. Qué datos externos agregaría en este dataset que mejoraría el análisis de los datos 8](#_Toc156256663)

[12. Elabore sus conclusiones y recomendaciones sobre este proyecto 8](#_Toc156256664)

[13. Proponer una arquitectura alternativa para este proceso ya sea con herramientas on premise o cloud (Sí aplica) 9](#_Toc156256665)

[**EJERCICIO 2: Alquiler de automóviles** 10](#_Toc156256666)

[1. Crear en hive una database car\_rental\_db y dentro una tabla llamada car\_rental\_analytics 10](#_Toc156256667)

[2. Crear script para el ingest de estos dos files 10](#_Toc156256668)

[3. Crear un script para tomar el archivo desde HDFS, hacer las siguientes transformaciones y finalmente insertar en Hive el resultado 10](#_Toc156256669)

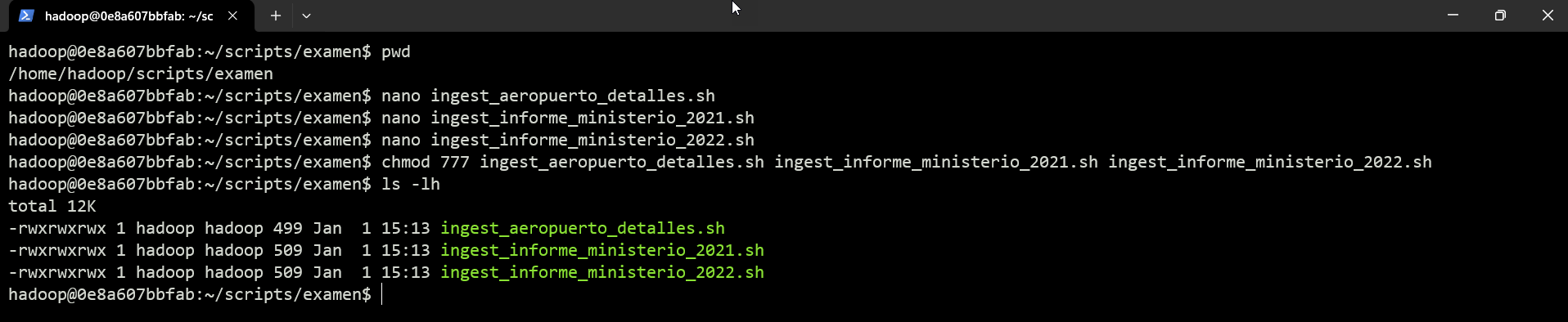
[4. Realizar un proceso automático en Airflow que orqueste los pipelines creados en los puntos anteriores 12](#_Toc156256670)

[5. Por medio de consultas SQL al data-warehouse, mostrar: 13](#_Toc156256671)

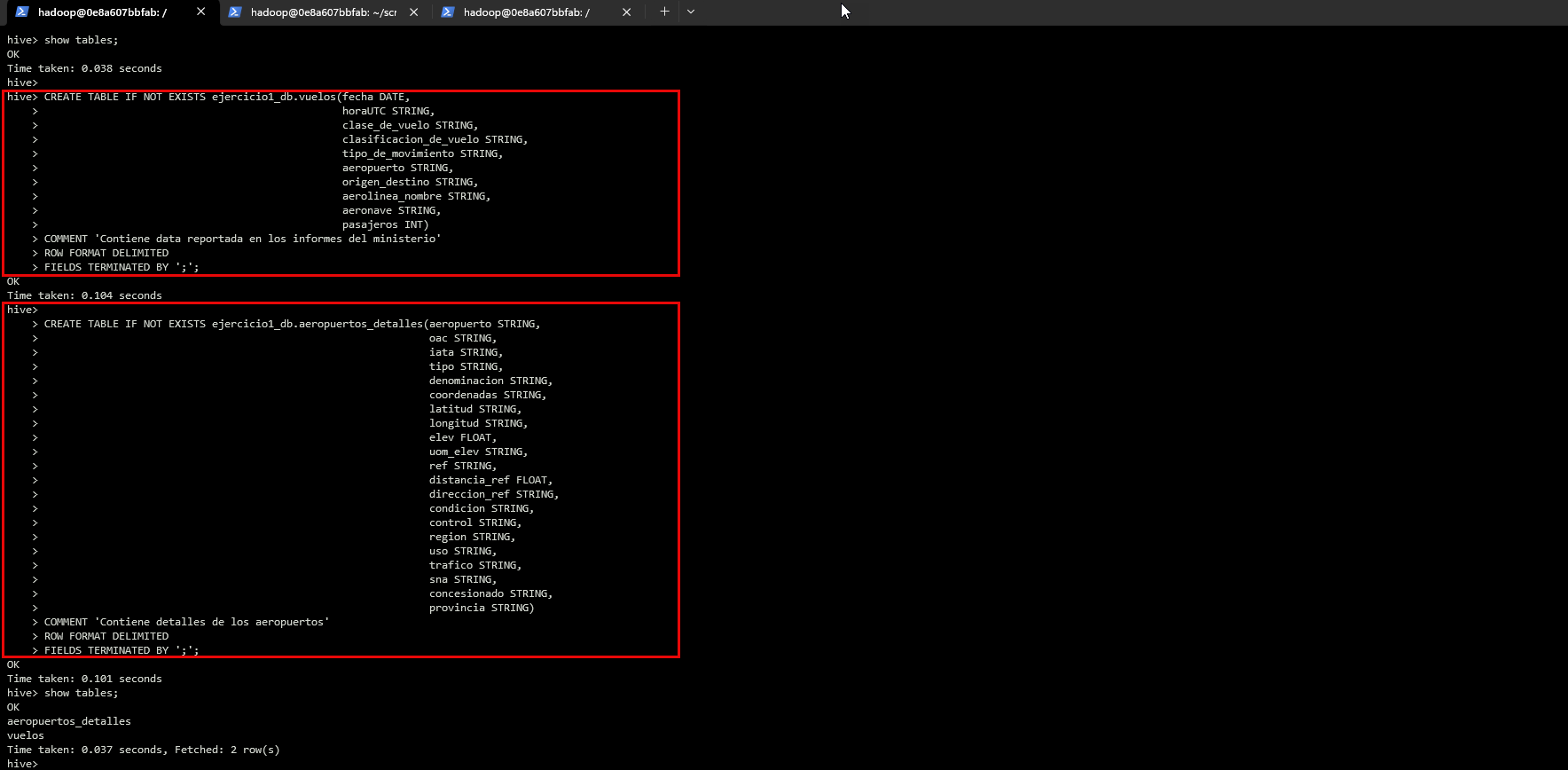
[6. Elabore sus conclusiones y recomendaciones sobre este proyecto 15](#_Toc156256672)

# **EJERCICIO 1: Aviación Civil**

## Creando los scripts para hacer Ingest de la data en csv. Y otorgando los permisos necesarios para su ejecución



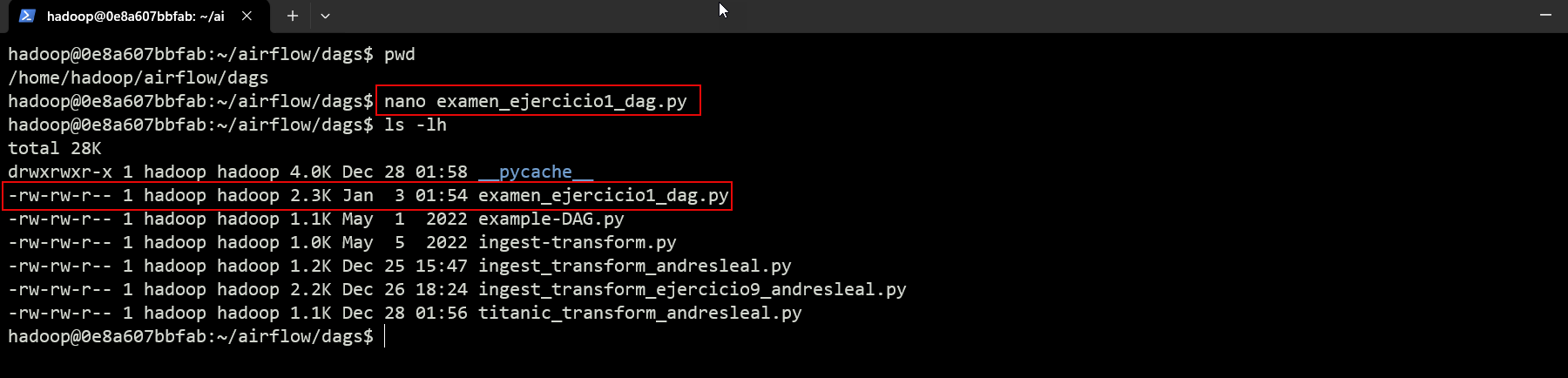
## Crear 2 tablas en el Data Warehouse, una para los vuelos realizados en 2021 y 2022 (2021-informe-ministerio.csv y 202206-informe-ministerio) y otra tabla para el detalle de los aeropuertos (aeropuertos\_detalle.csv)

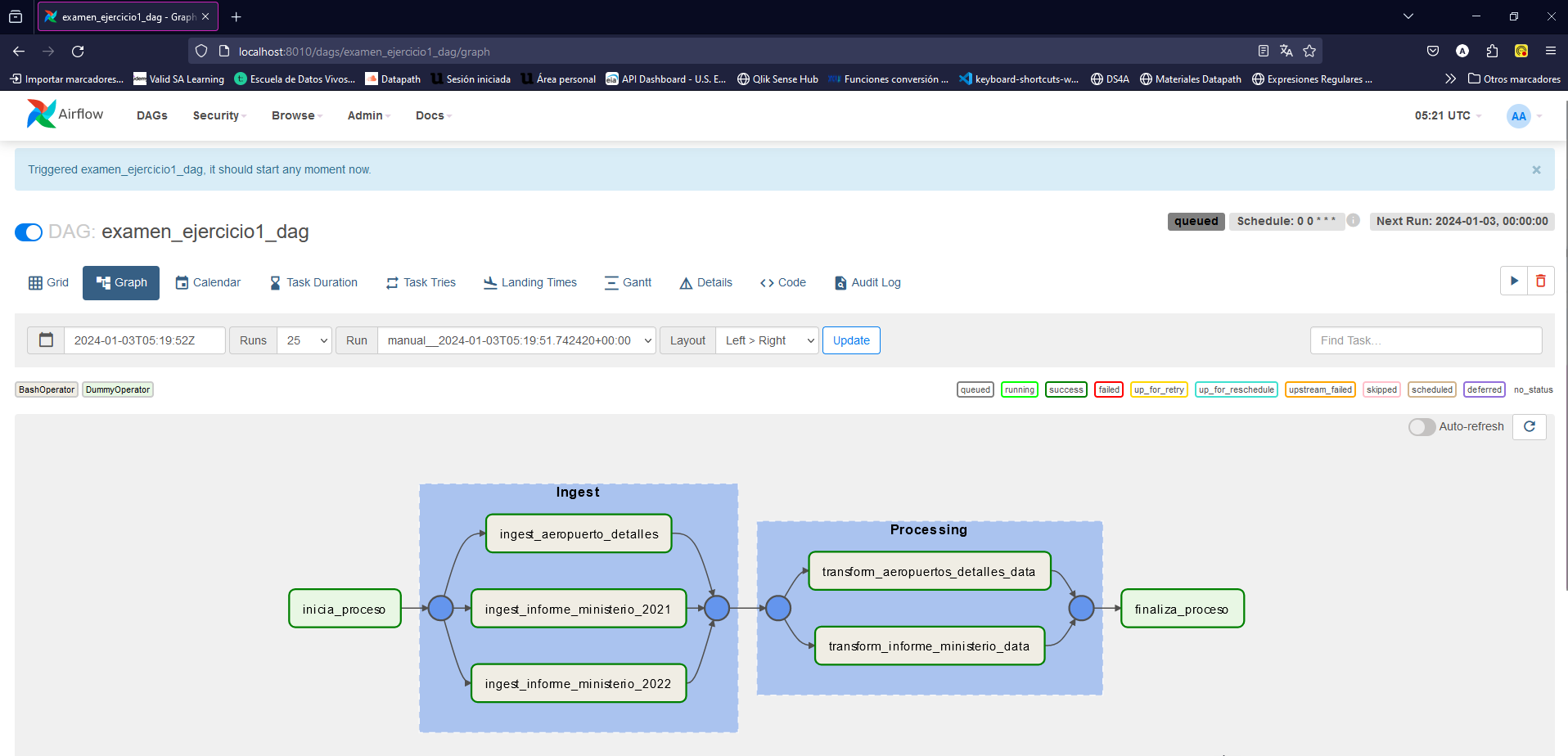


## Realizar un proceso automático orquestado por Airflow

Crear un DAG ingeste los archivos previamente mencionados entre las fechas 01/01/2021 y 30/06/2022 en las dos columnas creadas.

* Los archivos 202206-informe-ministerio.csv y 202206-informe-ministerio.csv → en la tabla aeropuerto\_tabla
* El archivo aeropuertos\_detalle.csv → en la tabla aeropuerto\_detalles\_tabla

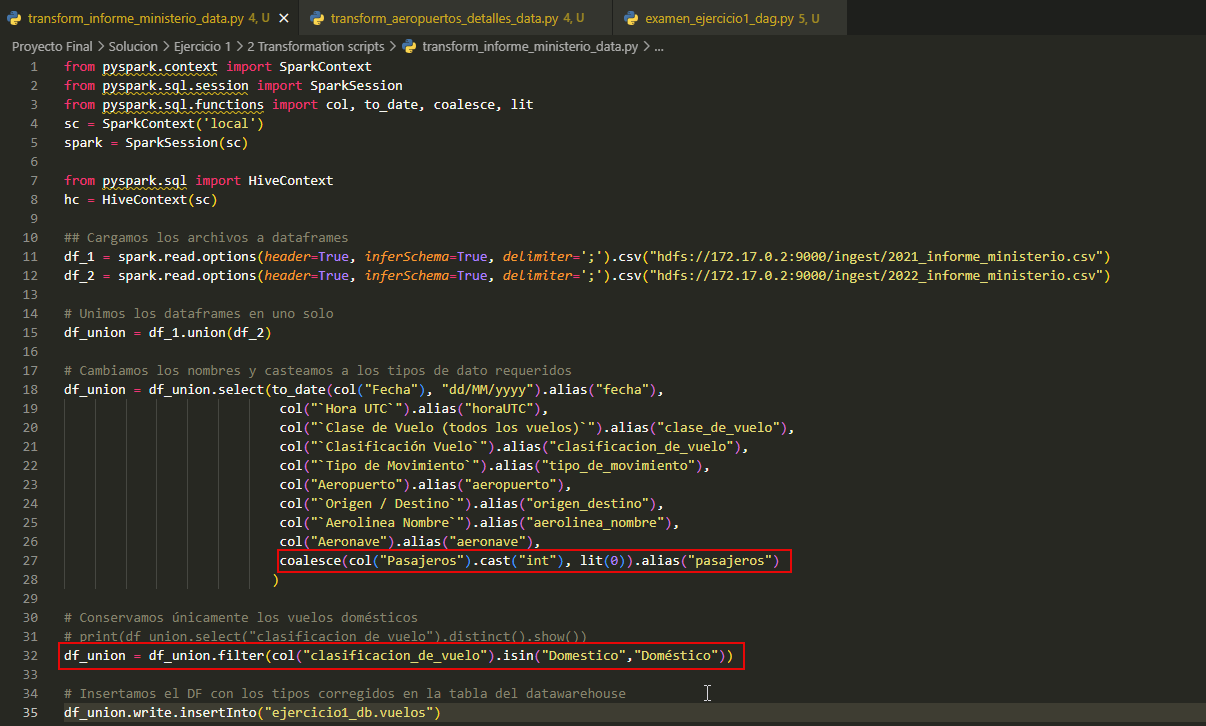


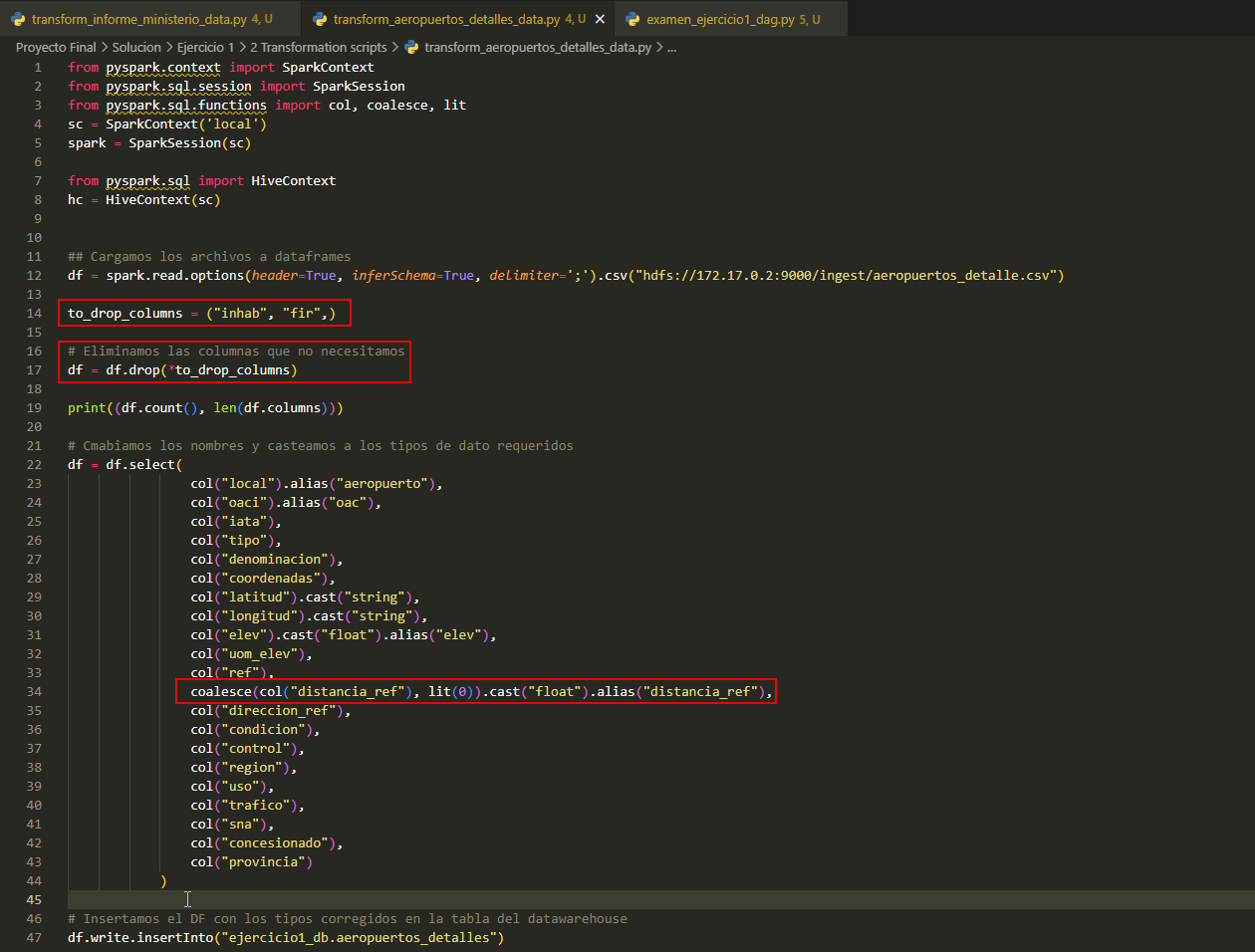


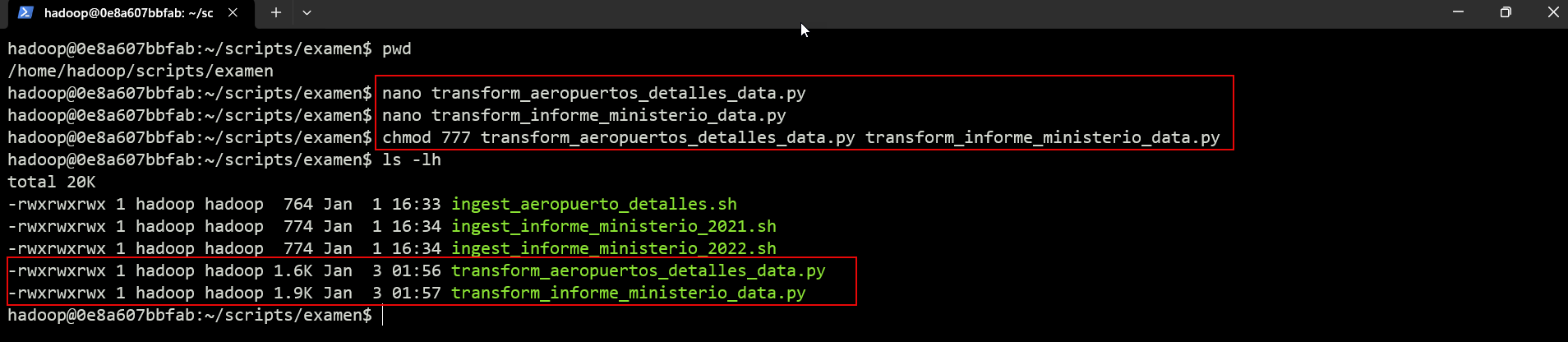
El código del DAG desarrollado para orquestar estas tareas en el archivo: **/Ejercicio 1/3 DAGs/examen\_ejercicio1\_dag.py**

## Transformaciones en los pipelines de datos:

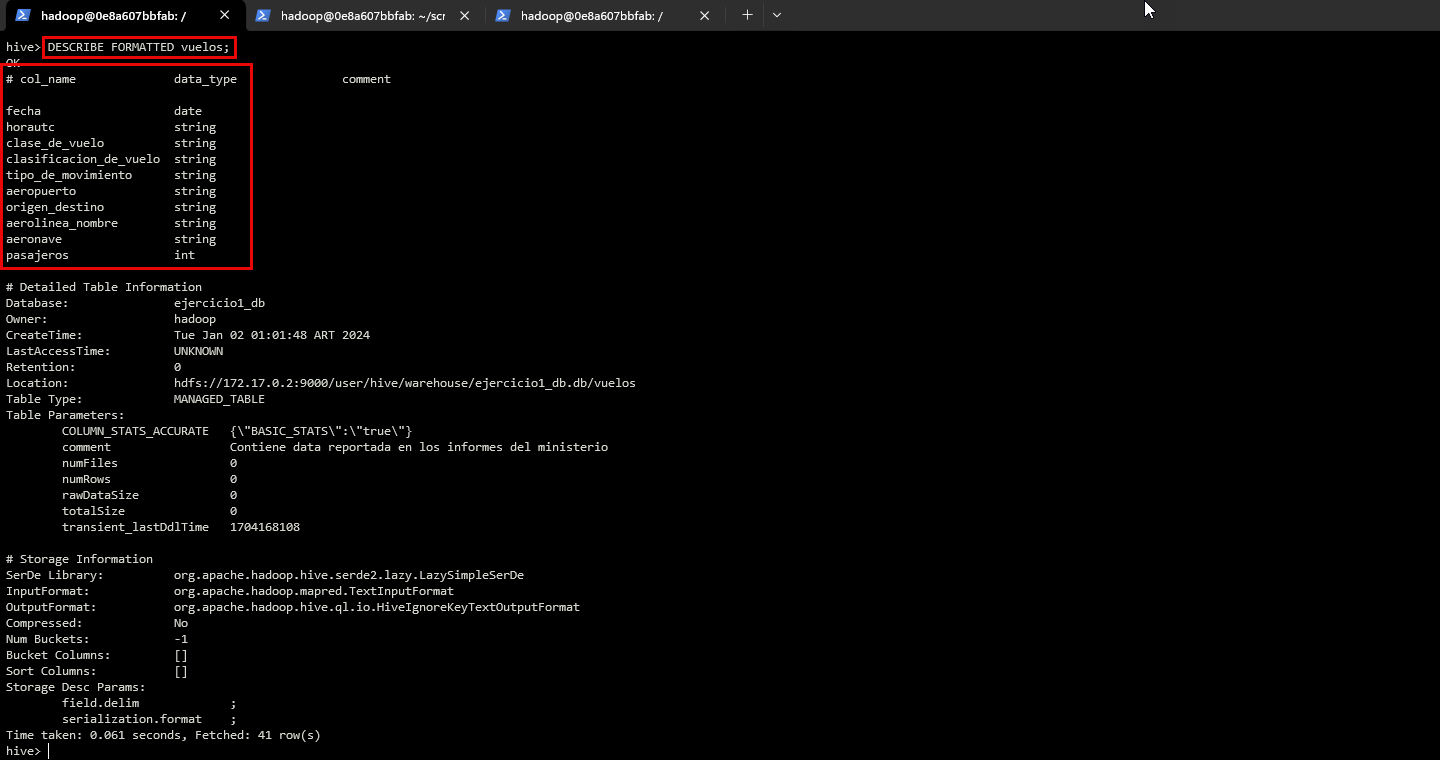
* Eliminar la columna inhab ya que no se utilizará para el análisis
* Eliminar la columna fir ya que no se utilizará para el análisis
* Eliminar la columna “calidad del dato” ya que no se utilizará para el análisis
* Filtrar los vuelos internacionales ya que solamente se analizarán los vuelos domésticos
* En el campo pasajeros si se encuentran campos en Null convertirlos en 0 (cero)
* En el campo distancia\_ref si se encuentran campos en Null convertirlos en 0 (cero)

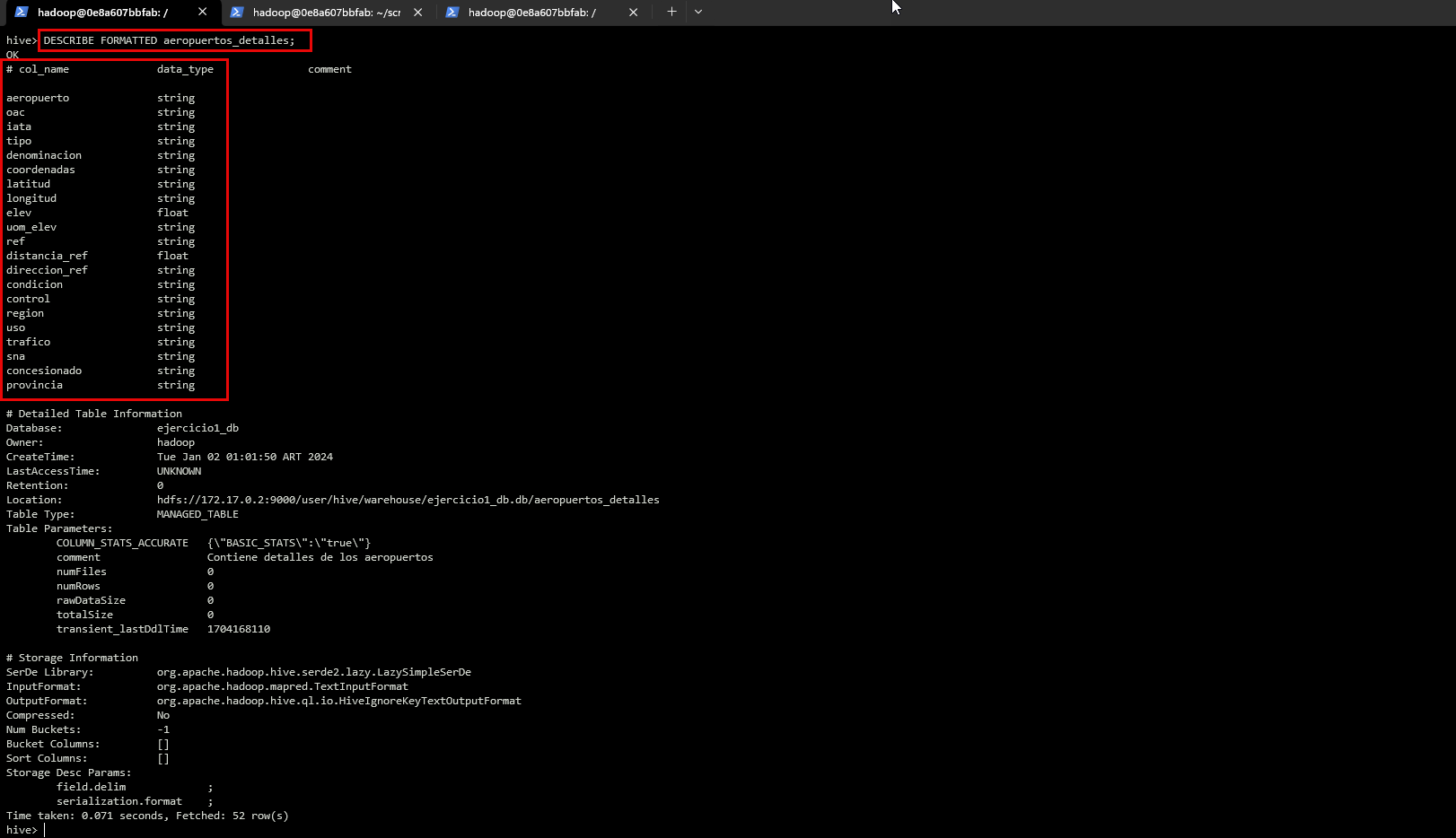




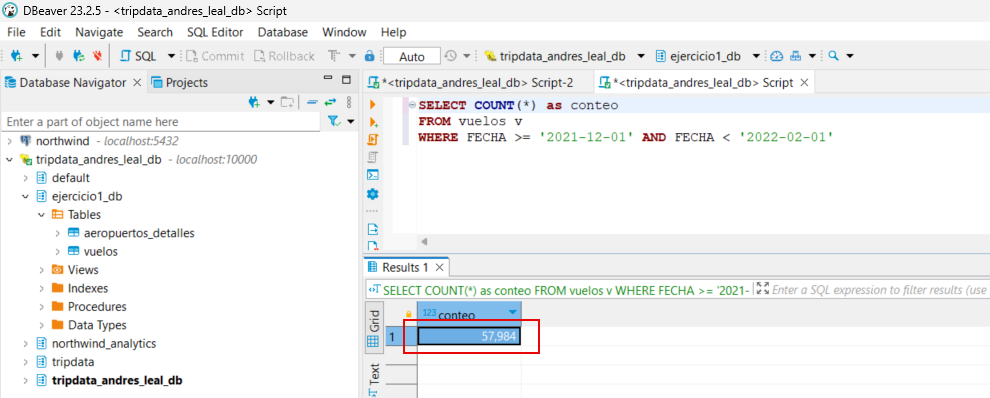


## Mostrar mediante una impresión de pantalla, que los tipos de campos de las tablas sean los solicitados en el datawarehouse

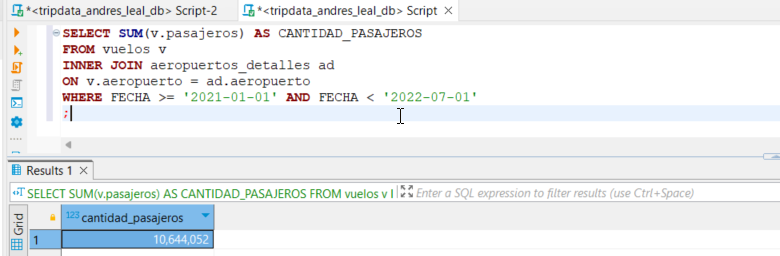




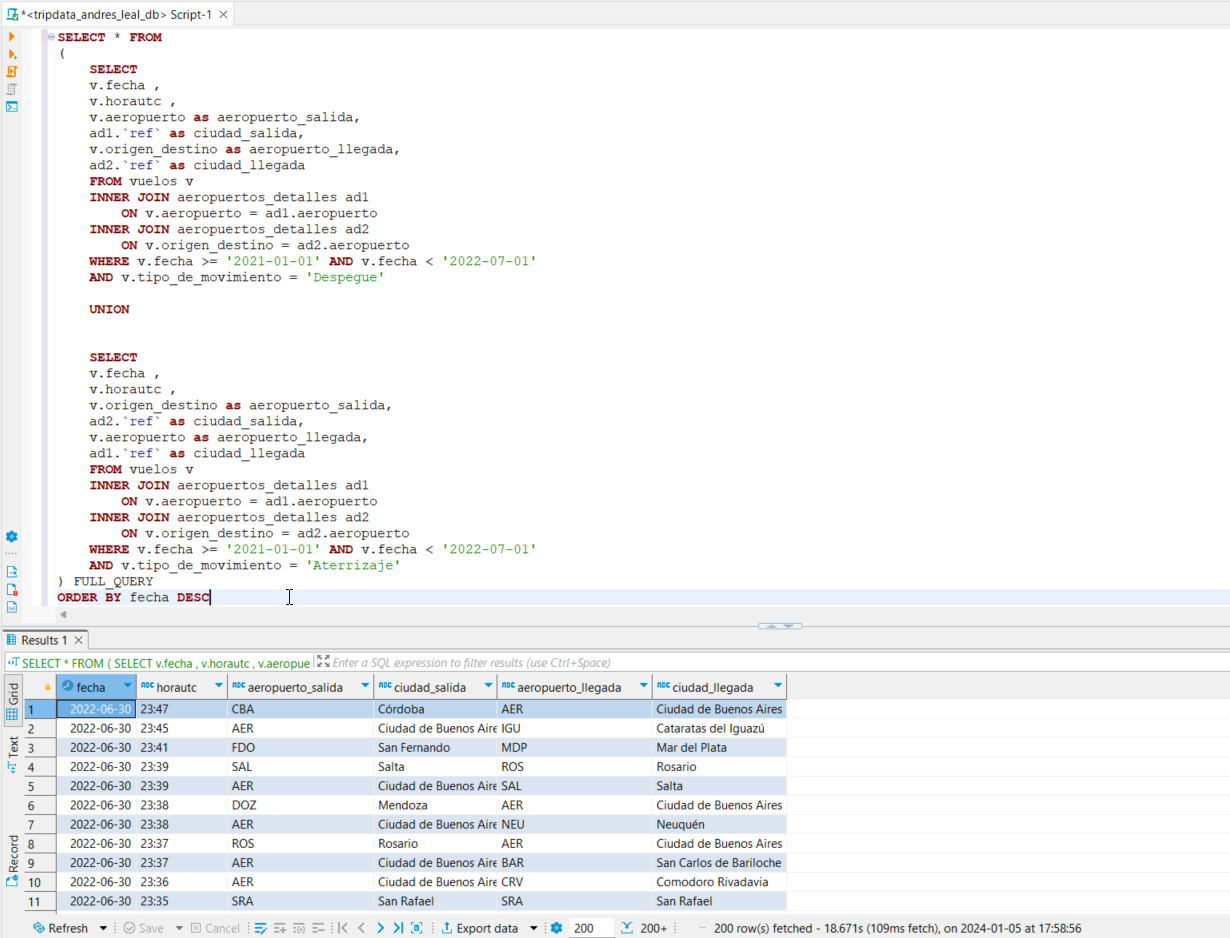
## Determinar la cantidad de vuelos entre las fechas 01/12/2021 y 31/01/2022. Mostrar consulta y Resultado de la query



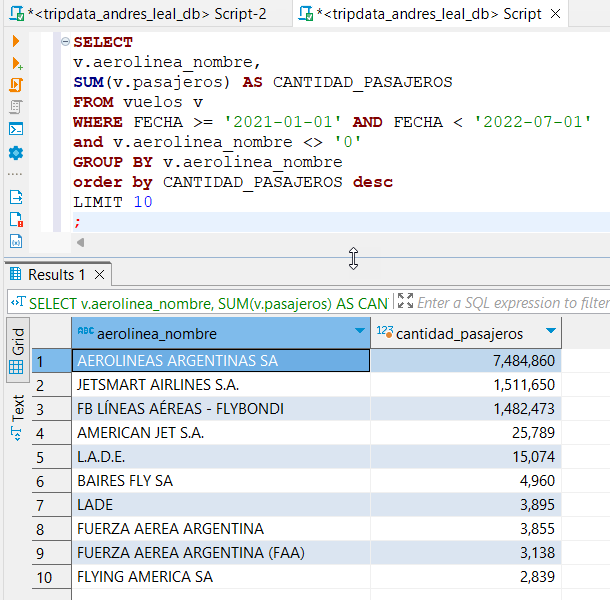
## Cantidad de pasajeros que viajaron en Aerolíneas Argentinas entre el 01/01/2021 y 30/06/2022. Mostrar consulta y Resultado de la query



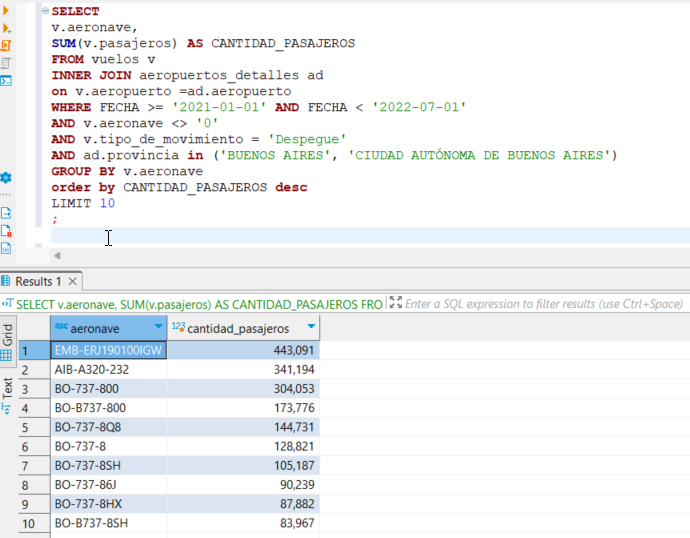
## Mostrar fecha, hora, código aeropuerto salida, ciudad de salida, código de aeropuerto de arribo, ciudad de arribo, y cantidad de pasajeros de cada vuelo, entre el 01/01/2022 y el 30/06/2022 ordenados por fecha de manera descendiente. Mostrar consulta y Resultado de la query

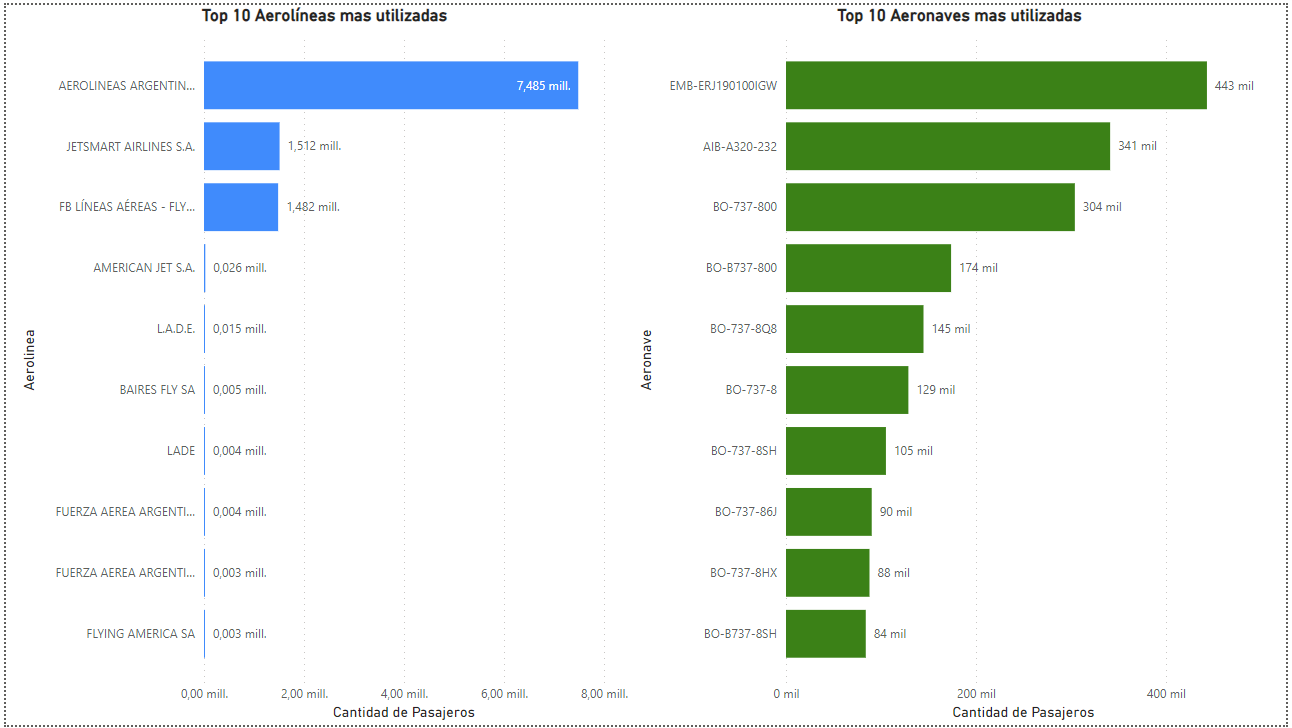


## Cuáles son las 10 aerolíneas que más pasajeros llevaron entre el 01/01/2021 y el 30/06/2022 exceptuando aquellas aerolíneas que no tengan nombre. Mostrar consulta y Visualización



## Cuáles son las 10 aeronaves más utilizadas entre el 01/01/2021 y el 30/06/22 que despegaron desde la Ciudad autónoma de Buenos Aires o de Buenos Aires, exceptuando aquellas aeronaves que no cuentan con nombre. Mostrar consulta y Visualización





## Qué datos externos agregaría en este dataset que mejoraría el análisis de los datos

Agregaría detalles técnicos de las aeronaves para poder realizar cálculos de la huella de carbono de cada vuelo, de acuerdo al combustible consumido / emisiones producidas.

También agregaría precios promedio de cada clase de vuelo para cada aerolínea, sumado al precio del combustible de las aeronaves, para calcular métricas financieras que la Aerocivil pudiera utilizar para estimar las obligaciones tributarias de cada aerolínea.

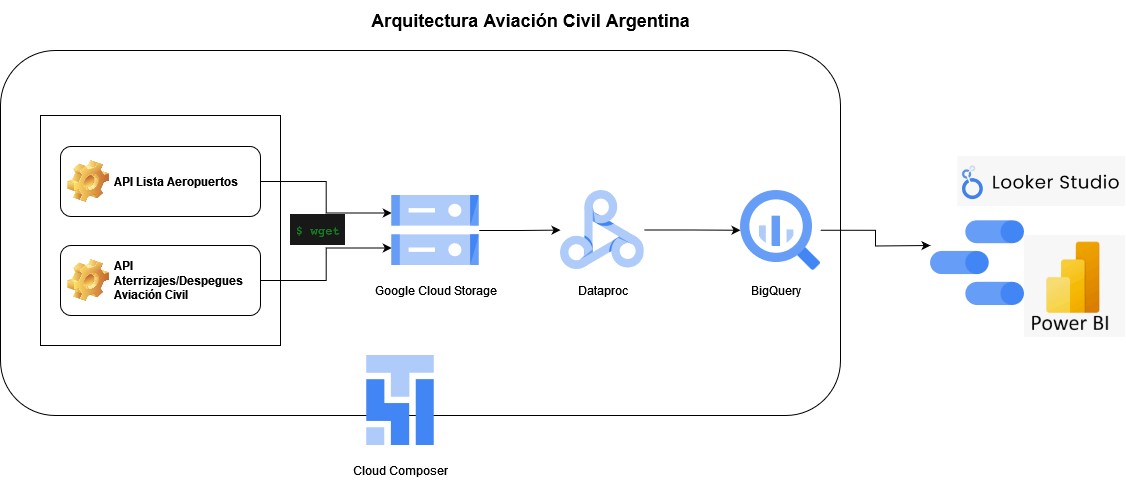
Sería útil además agregar información sobre los vuelos retrasados, que el permita a la Aerocivil calcular métricas de cumplimiento de las aerolíneas para/con los pasajeros; y con esto tomar medidas necesarias para lograr un mejor servicio en el país.

## Elabore sus conclusiones y recomendaciones sobre este proyecto

* Se recomienda agregar un diccionario de datos, toda una capa de metadata, a la que nuevos desarrolladores puedan entrar a consultar el significado y tipos de dato que cada columna contiene. Esto, para lograr acelerar el proceso de entendimiento de los datos con los que se trabaja y con ello acelerar futuras implementaciones.
* Si para una columna no se cuenta con algún dato, para su persistencia se recomienda guardar el dato como Nulo, en lugar de usar algún dato por default, como sucede por ejemplo con la columna “aeronave” y el valor 0.

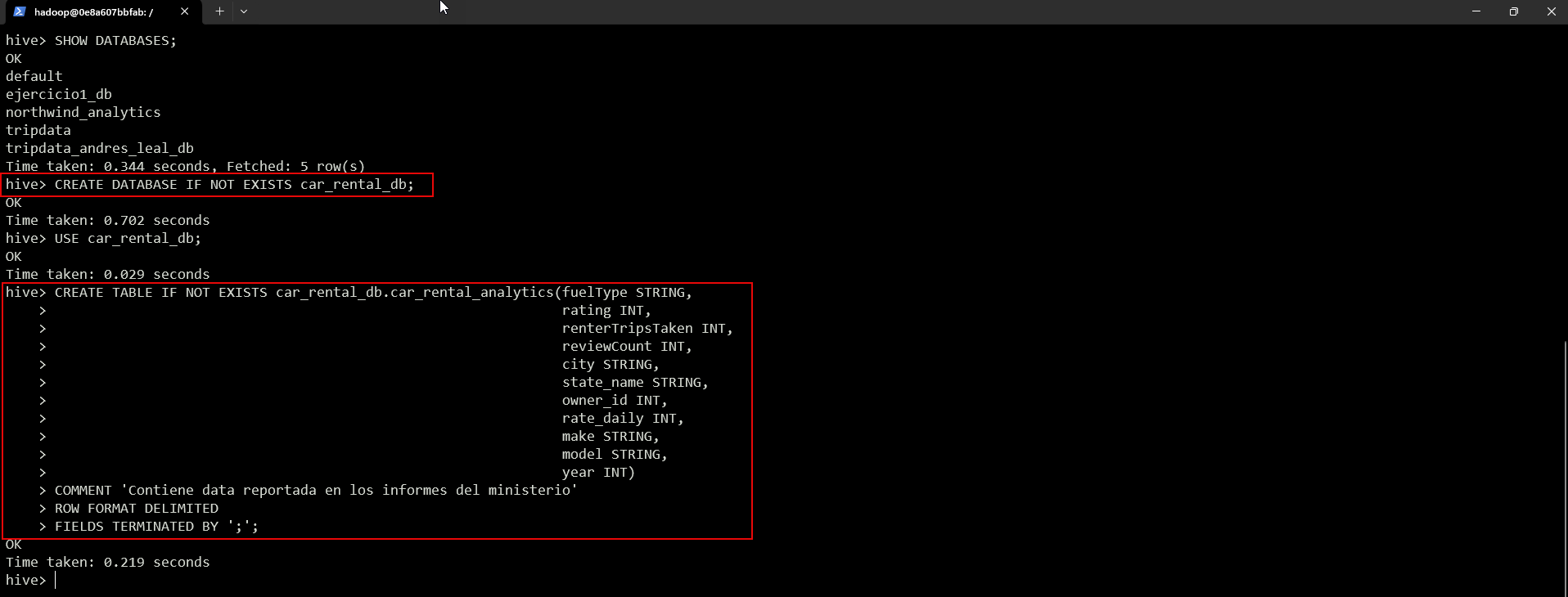
## Proponer una arquitectura alternativa para este proceso ya sea con herramientas on premise o cloud (Sí aplica)

Propongo el uso de una arquitectura con herramientas Cloud, donde el proceso sea orquestado a través de Composer: Se descargan los datos de la aeronáutica civil directamente desde su API hacia Cloud Storage, donde a través de Jobs en Dataproc se realizarán las transformaciones pertinentes que “sazonen” los datos, para finalmente persistirlos de forma estructurada y limpia en BigQuery. De allí se podrán generar visualizaciones con la herramienta de visualización favorita del analista:

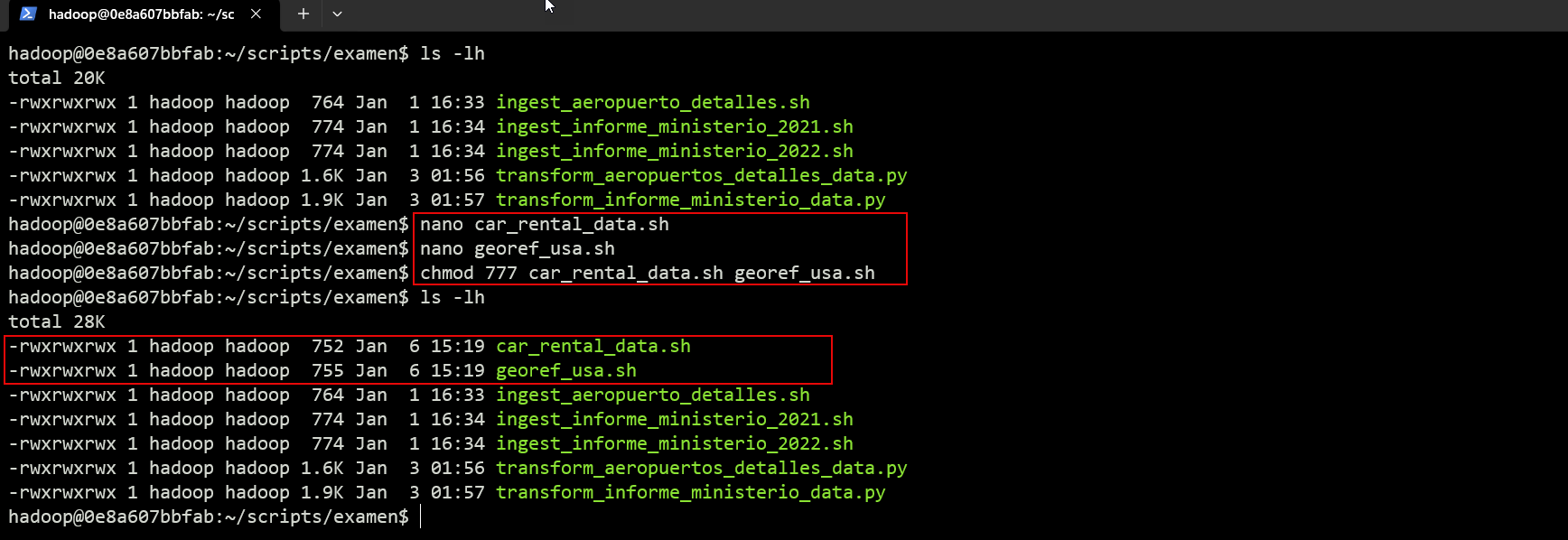


# **EJERCICIO 2: Alquiler de automóviles**

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

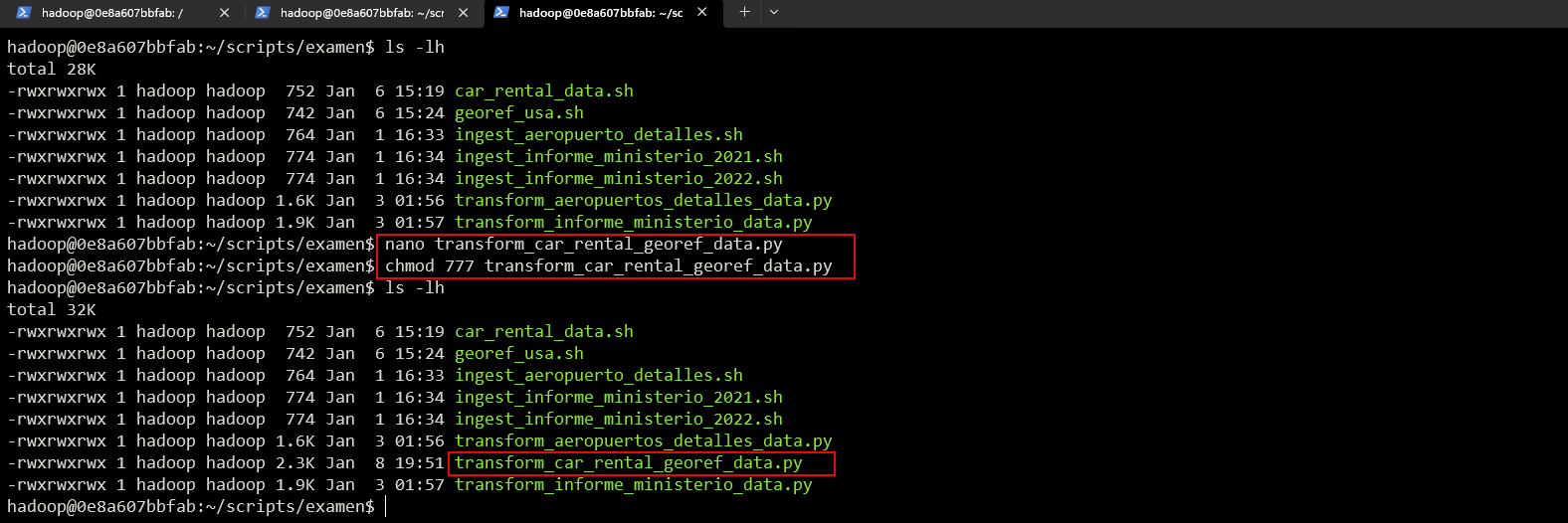


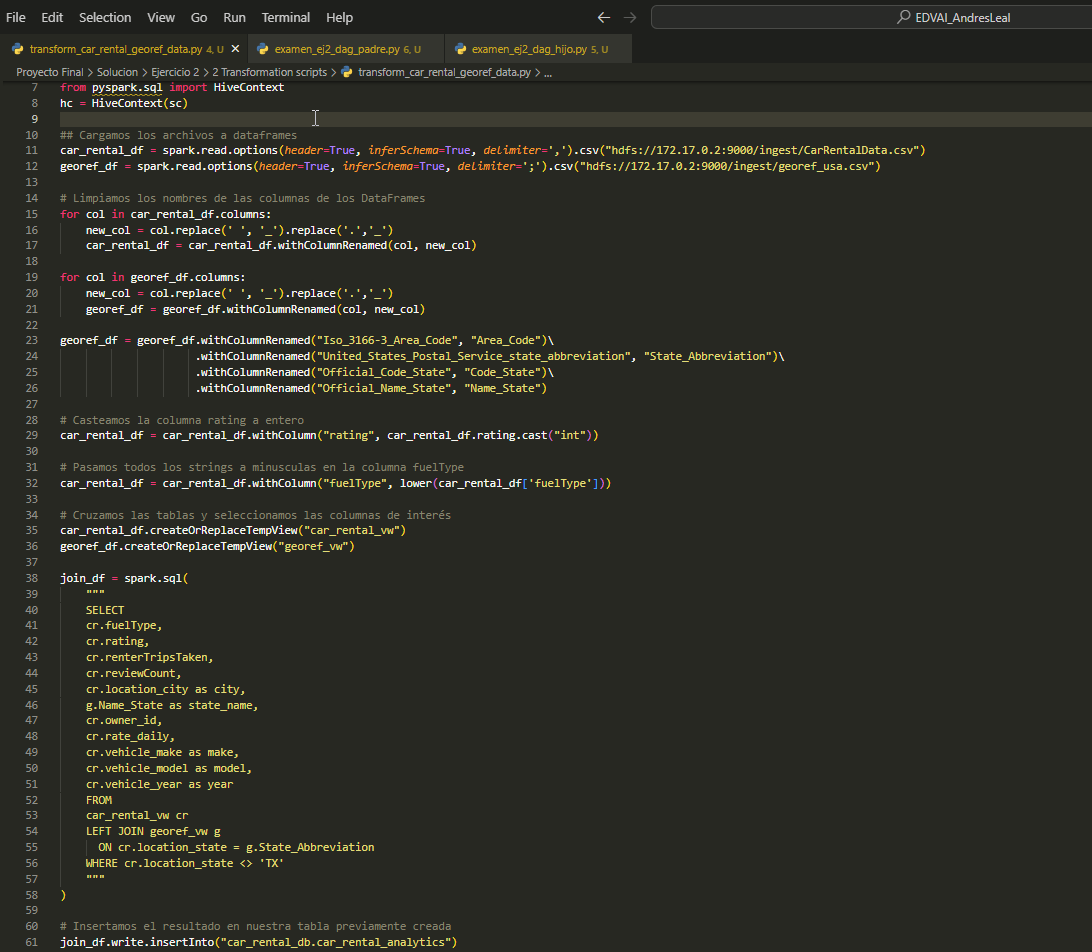
## Crear script para el ingest de estos dos files



## Crear un script para tomar el archivo desde HDFS, hacer las siguientes transformaciones y finalmente insertar en Hive el resultado

* 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

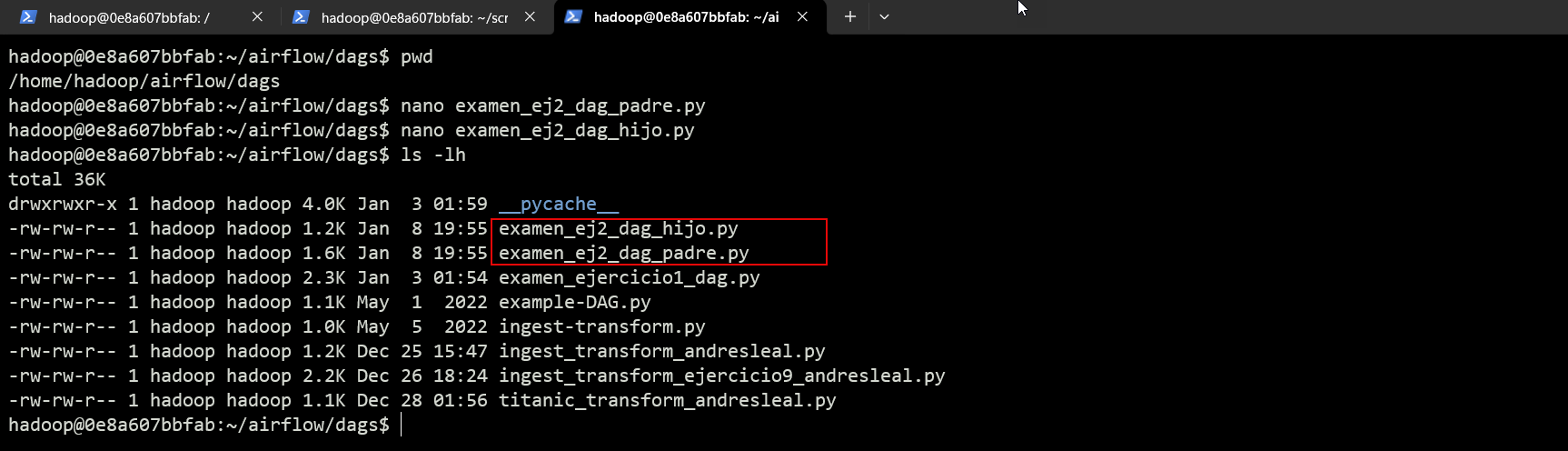


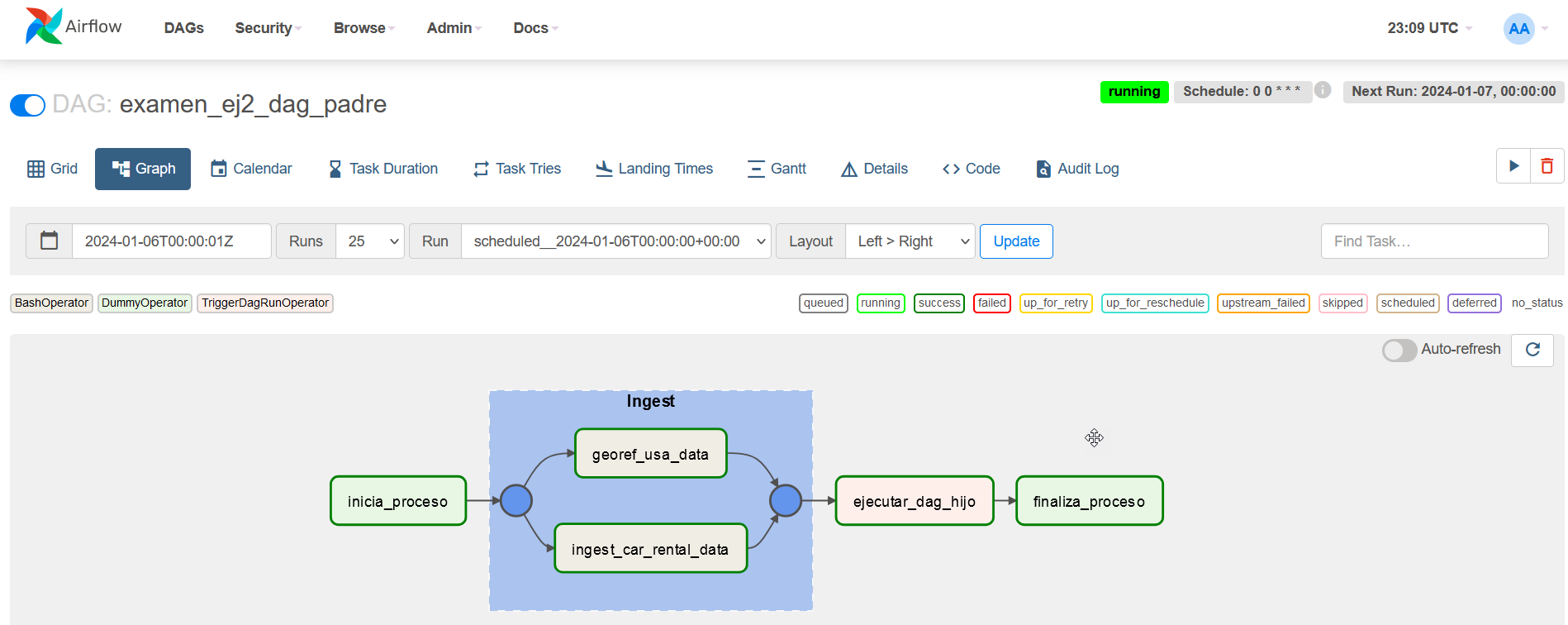


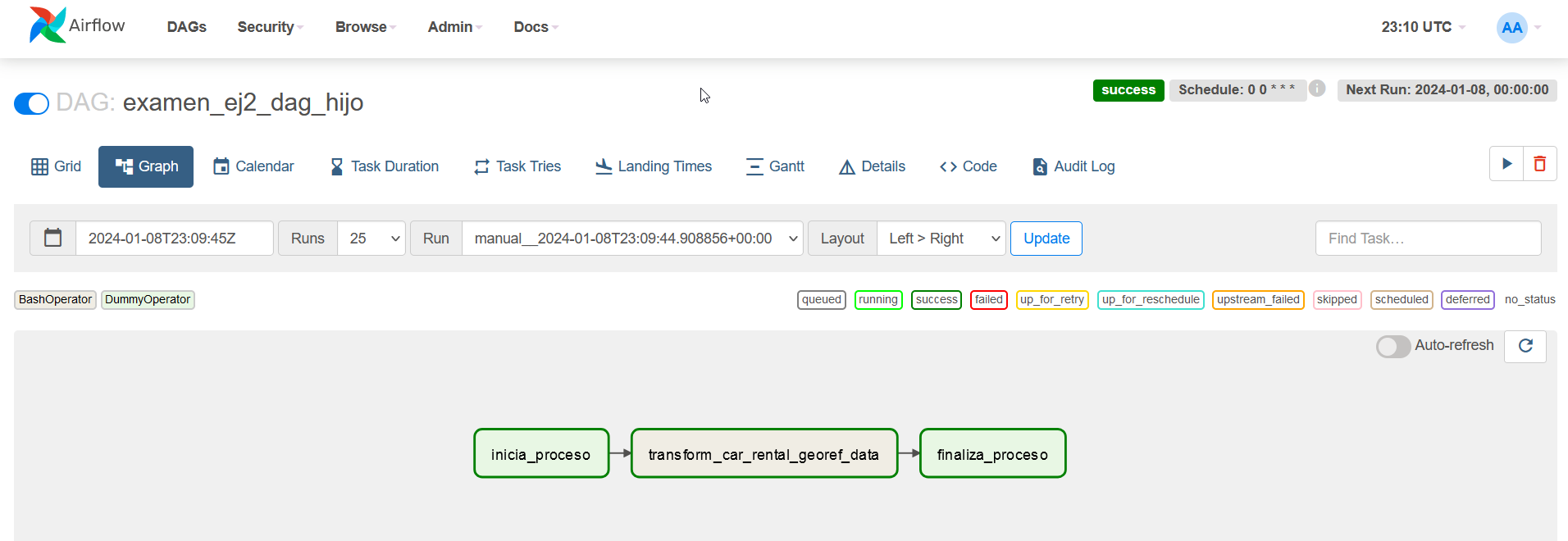
## Realizar un proceso automático en Airflow que orqueste los pipelines creados en los puntos anteriores

Crear dos tareas:

* Un DAG padre que ingente los archivos y luego llame al DAG hijo
* Un DAG hijo que procese la información y la cargue en Hive

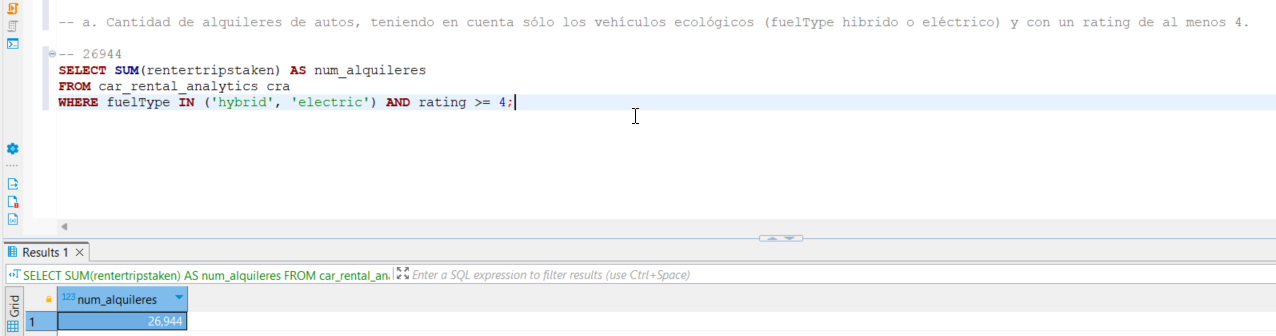




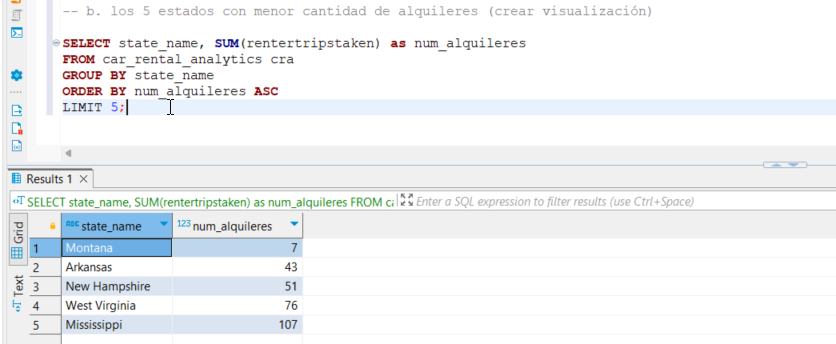


## Por medio de consultas SQL al data-warehouse, mostrar:

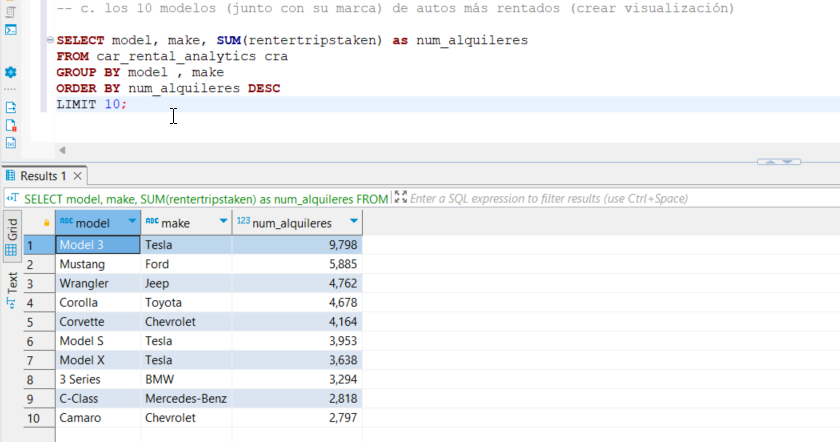
1. Cantidad de alquileres de autos, teniendo en cuenta sólo los vehículos ecológicos (fuelType hibrido o eléctrico) y con un rating de al menos 4.

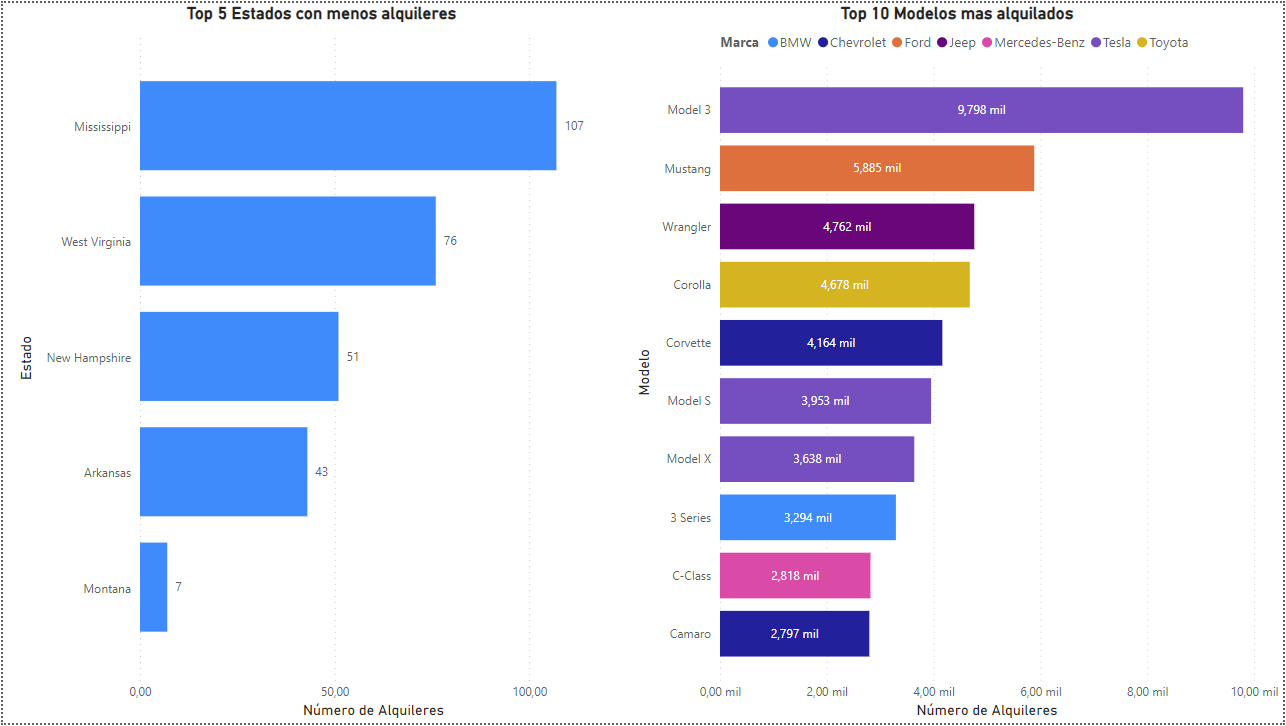


1. Los 5 estados con menor cantidad de alquileres (crear visualización)

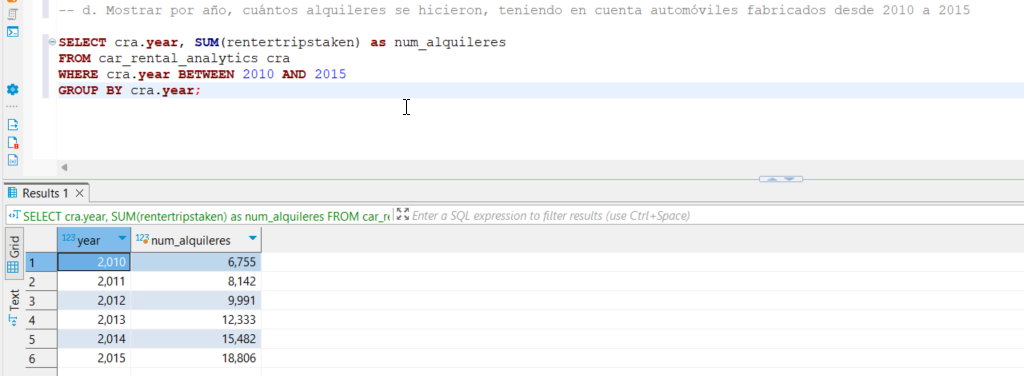


1. Los 10 modelos (junto con su marca) de autos más rentados (crear visualización)

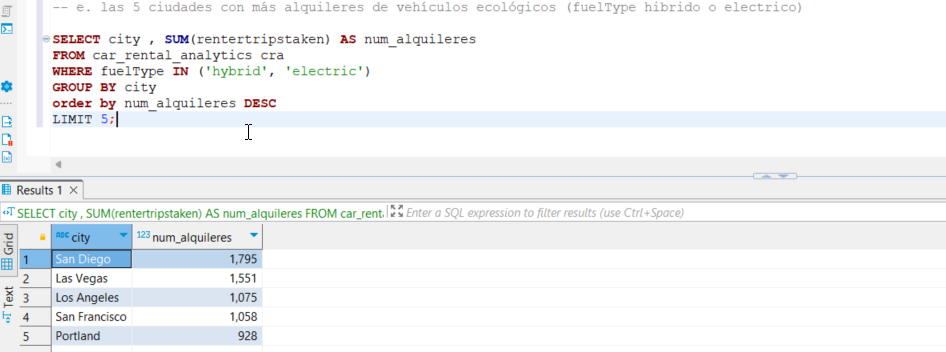




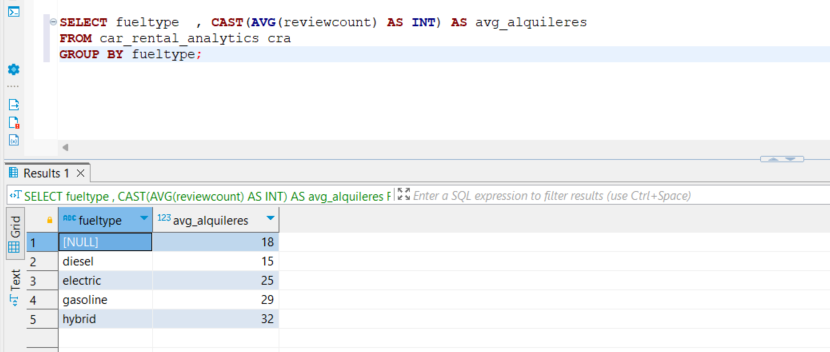
1. Mostrar por año, cuántos alquileres se hicieron, teniendo en cuenta automóviles fabricados desde 2010 a 2015



1. Las 5 ciudades con más alquileres de vehículos ecológicos (fuelType hibrido o electrico)



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



## Elabore sus conclusiones y recomendaciones sobre este proyecto

El proyecto simplifica la toma y transformación de data de interés, al proveer un pipeline automatizado y flexible, ya que se le podrían agregar datasets y Jobs de transformación adicionales a los que ya tiene.

Dado que los datos de geo referencia serán siempre los mismos se recomienda descargar y persistir dichos datos en una tabla en el Data Warehouse, en lugar de descargarlo una y otra vez en el pipeline creado. Así mismo, se recomienda crear un diccionario de datos para facilitar el entendimiento de cada columna existente en esta tabla.

Dado que no estamos trabajando un pipeline de mucha complejidad se recomienda redefinir todo el proceso en un solo DAG, en lugar de la estructura DAG Padre -> DAG Hijo que se tiene.

Con la estructura que los datos fuente tienen en este momento, se considera viable la arquitectura actual, manejada on Prem, ya que no es de alta exigencia. Si se llegasen a capturar datos de arrendamiento de vehículos día a día, se recomienda construir un pipeline equivalente usando herramientas Cloud que faciliten su escalabilidad en almacenamiento y procesamiento cuando sea requerido.