EXAMEN FINAL

# Ejercicio 1:

**Aviación Civil**

La Administración Nacional de Aviación Civil necesita una serie de informes para elevar al ministerio de transporte acerca de los aterrizajes y despegues en todo el territorio Argentino, como puede ser: cuales aviones son los que más volaron, cuántos pasajeros volaron, ciudades de partidas y aterrizajes entre fechas determinadas, etc.

Usted como data engineer deberá realizar un pipeline con esta información, automatizarlo y realizar los análisis de datos solicitados que permita responder las preguntas de negocio, y hacer sus recomendaciones con respecto al estado actual.

Listado de vuelos realizados:

[https://datos.transporte.gob.ar/dataset/aterrizajes-despegues-procesados-por-administracion-na](https://datos.transporte.gob.ar/dataset/aterrizajes-despegues-procesados-por-administracion-nacional-aviacion-civil-anac) [cional-aviacion-civil-anac](https://datos.transporte.gob.ar/dataset/aterrizajes-despegues-procesados-por-administracion-nacional-aviacion-civil-anac)

Listado de detalles de aeropuertos de Argentina: <https://datos.transporte.gob.ar/dataset/lista-aeropuertos>

**TAREAS**

1. Hacer ingest de los siguientes files relacionados con transporte aéreo de Argentina :

2021: <https://data-engineer-edvai.s3.amazonaws.com/2021-informe-ministerio.csv>

2022: <https://data-engineer-edvai.s3.amazonaws.com/202206-informe-ministerio.csv>

Aeropuertos\_detalles: <https://data-engineer-edvai.s3.amazonaws.com/aeropuertos_detalle.csv>

A computer screen with white text

Description automatically generated



1. Crear 2 tablas en el datawarehouse, 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)

Schema Tabla 1:

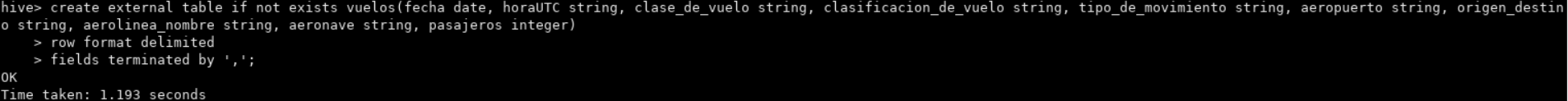
|  |  |
| --- | --- |
| **campos** | **tipo** |
| fecha | date |
| horaUTC | string |
| clase\_de\_vuelo | string |
| clasiﬁcacion\_de\_vuelo | string |
| tipo\_de\_movimiento | string |
| aeropuerto | string |
| origen\_destino | string |
| aerolinea\_nombre | string |
| aeronave | string |
| pasajeros | integer |

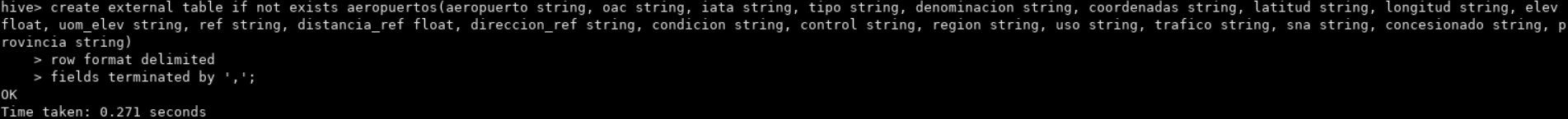
Schema Tabla 2:

|  |  |
| --- | --- |
| **Campo** | **Tipo** |
| aeropuerto | string |
| oac | string |
| iata | string |

|  |  |
| --- | --- |
| tipo | string |
| denominacion | string |
| coordenadas | string |
| latitud | string |
| longitud | string |
| elev | ﬂoat |
| uom\_elev | string |
| ref | string |
| distancia\_ref | ﬂoat |
| direccion\_ref | string |
| condicion | string |
| control | string |
| region | string |
| uso | string |
| traﬁco | string |
| sna | string |
| concesionado | string |
| provincia | string |







A black background with white text

Description automatically generated

1. Realizar un proceso automático orquestado por airflow que ingeste los archivos previamente mencionados entre las fechas 01/01/2021 y 30/06/2022 en las dos tablas creadas.

Los archivos 202206-informe-ministerio.csv y 202206-informe-ministerio.csv → en la tabla vuelos

El archivo aeropuertos\_detalle.csv → en la tabla aeropuertos

No se pueden ingestar directamente ya que hay que realizar las transformaciones en el ejercicio siguiente para que haya compatibilidad.

1. Realizar las siguientes 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) ó

Creamos el script vuelos.py para realizar la transformación de las tablas y su ingesta a Hive

A computer screen shot of a program code

Description automatically generated

Creamos el script dag\_vuelos.py para orquestar los procesos de ingesta, transformación y pasaje a Hive

A screenshot of a computer

Description automatically generated

Corremos el dag para verificar que funcione bien.

A screenshot of a computer

Description automatically generated

1. Mostrar mediante una impresión de pantalla, que los tipos de campos de las tablas sean los solicitados en el datawarehouse (ej: fecha date, aeronave string, pasajeros integer, etc.)

Corremos las mismas líneas de vuelos.py en pyspark para verificar el esquema de las tablas resultantes.

A screen shot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

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

A screenshot of a computer program

Description automatically generated

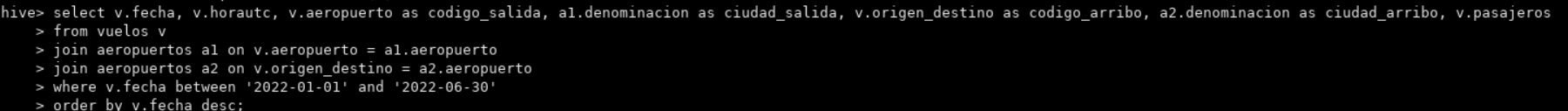
1. Cantidad de pasajeros que viajaron en Aerolíneas Argentinas entre el 01/01/2021 y 30/06/2022. Mostrar consulta y Resultado de la query

A screenshot of a computer program

Description automatically generated

1. 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

Consulta



Resultado (coloco captura de principio y fin ya que es muy largo el resultado)

A black screen with white text

Description automatically generated

A black screen with white text

Description automatically generated

1. 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

Consulta (las aerolíneas que no tienen nombre son aquellas que figuran con 0)

A black screen with white text

Description automatically generated

Resultado

A black screen with white text

Description automatically generated

1. 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

Consulta (las aeronaves que no tienen nombre son aquellas que figuran con 0)

A computer screen with white text

Description automatically generated

Resultado

A black screen with numbers and letters

Description automatically generated

1. Qué datos externos agregaría en este dataset que mejoraría el análisis de los datos
2. Elabore sus conclusiones y recomendaciones sobre este proyecto.

Respondo por ejercicios 11 y 12:

Algunas recomendaciones que haría, específicamente respecto a los datos de los vuelos son:

* Aunque tenemos la fecha y la horaUTC para los vuelos, no se menciona la duración del vuelo. Esta información puede ser crucial para analizar eficiencias, identificar rutas más largas de lo normal o entender mejor el tráfico aéreo.
* Saber si un vuelo fue cancelado, retrasado o llegó a tiempo puede ser esencial para analizar la eficiencia de las aerolíneas y los aeropuertos. Esta información podría ser crucial para las decisiones operativas.
* Un identificador único para cada vuelo permitiría un seguimiento más detallado y evitaría confusiones en el análisis.
* Las condiciones meteorológicas influyen significativamente en las operaciones de vuelo. Tener datos sobre el clima en el momento del despegue y aterrizaje podría ayudar a identificar patrones o problemas relacionados con condiciones específicas.

Se podría sacar más provecho en la elaboración de reportes, como ser:

* Rutas más populares: Identificar las rutas más transitadas y las menos transitadas para tomar decisiones sobre la expansión o reducción de servicios.
* Eficiencia de Aerolíneas: Comparar aerolíneas en términos de puntualidad, capacidad de pasajeros y otros KPIs relevantes.
* Optimización de Horarios: Analizar si hay horarios específicos que son más propensos a retrasos o problemas.
* Eficiencia de Aeropuertos: Comparar aeropuertos en términos de tráfico, capacidad y operaciones para identificar cuellos de botella o áreas de mejora.

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

Para este ejercicio asumimos que los datos se ingieren en tiempo real, por lo que imaginamos una estructura de ingesta tipo streaming.

A screenshot of a computer

Description automatically generated

* Los datos provienen de dos sitios web públicos.
* Los datos se ingieren en tiempo real a través de Cloud Pub/Sub.
* Dataflow es utilizado para procesar y transformar estos datos en tiempo real ya que es adecuado para esta tarea debido a su capacidad de manejar datos en streaming y su integración nativa con Pub/Sub.
* Una vez procesados los datos, los almaceno en BigQuery.
* Finalmente, exporto el o los csv a Looker para poder visualizar la información en diferentes charts.
* Todo esto, claramente, orquestado por Cloud Composer

# Ejercicio 2:

**Alquiler de automóviles**

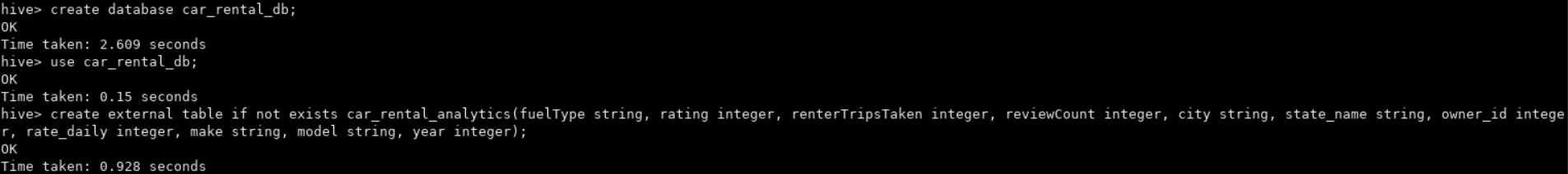
Una de las empresas líderes en alquileres de automóviles solicita una serie de dashboards y reportes para poder basar sus decisiones en datos. Entre los indicadores mencionados se encuentran total de alquileres, segmentación por tipo de combustible, lugar, marca y modelo de automóvil, valoración de cada 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.

1. Crear en hive una database car\_rental\_db y dentro una tabla llamada car\_rental\_analytics, con estos campos:

|  |  |
| --- | --- |
| **campos** | **tipo** |
| fuelType | string |
| rating | integer |
| renterTripsTaken | integer |
| reviewCount | integer |
| city | string |
| state\_name | string |
| owner\_id | integer |

|  |  |
| --- | --- |
| rate\_daily | integer |
| make | string |
| model | string |
| year | integer |

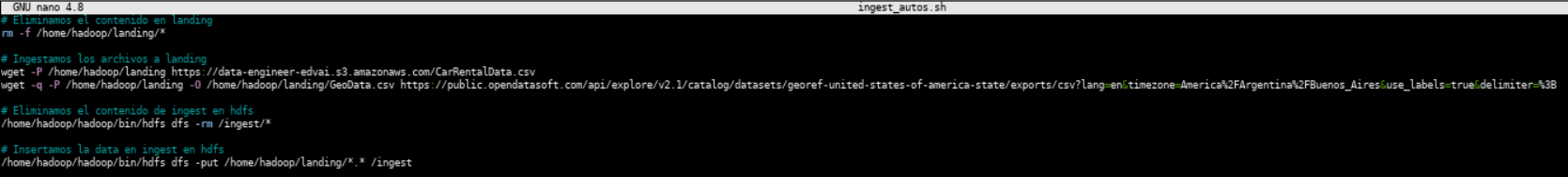


1. Crear script para el ingest de estos dos files
   * https://data-engineer-edvai.s3.amazonaws.com/CarRentalData.csv
   * https://public.opendatasoft.com/api/explore/v2.1/catalog/datasets/georef-united-st ates-of-america-state/exports/csv?lang=en&timezone=America%2FArgentina%2FBueno s\_Aires&use\_labels=true&delimiter=%3B

**Sugerencia**: descargar el segundo archivo con un comando similar al abajo mencionado, ya que al tener caracteres como ‘&’ falla si no se le asignan comillas. Adicionalmente, el parámetro -O permite asignarle un nombre más legible al archivo descargado

*wget -P ruta\_destino -O ruta\_destino/nombre\_archivo.csv ruta\_al\_archivo*

Creamos el script para ingestar los files en hdfs



1. 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

Creamos el script autos.py

A screenshot of a computer

Description automatically generated

1. Realizar un proceso automático en Airflow que orqueste los pipelines creados en los puntos anteriores. Crear dos tareas:
2. Un DAG padre que ingente los archivos y luego llame al DAG hijo

A screenshot of a computer program

Description automatically generated

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

A screen shot of a computer

Description automatically generated

Corro los DAG’s para verificar que el proceso funcione.

dag\_padre

A screenshot of a computer

Description automatically generated

dag\_hijo A screenshot of a computer

Description automatically generated

1. Por medio de consultas SQL al data-warehouse, mostrar:
2. Cantidad de alquileres de autos, teniendo en cuenta sólo los vehículos ecológicos (fuelType hibrido o eléctrico) y con un rating de al menos 4.

A screenshot of a computer program

Description automatically generated

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

Consulta

A black background with white text

Description automatically generated

Resultado

A black background with white text

Description automatically generated

Visualización

Para realizar la visualización debo exportar la tabla desde Hive. Para ello, ingreso a la base de datos desde dbeaver.

Me conecto a la base de datos.

A screenshot of a computer

Description automatically generated

Me descargo la tabla.

A screenshot of a computer

Description automatically generated

Luego me envío el archivo por WeTransfer para poder utilizarlo en mi computadora local.

Lo cargo en Power Bi.

A screenshot of a computer

Description automatically generated

Visualización

A graph of blue rectangular bars

Description automatically generated with medium confidence

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

Consulta

A black screen with white text

Description automatically generated

Resultado

A screenshot of a computer

Description automatically generated

Visualización

A graph of blue bars with white text

Description automatically generated

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

Consulta

A black background with white text

Description automatically generated

Resultado

A black background with white numbers

Description automatically generated

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

Consulta

A screen shot of a computer code

Description automatically generated

Resultado

A black background with white text

Description automatically generated

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

Consulta



Resultado

A black background with numbers

Description automatically generated

1. Elabore sus conclusiones y recomendaciones sobre este proyecto.

Algunas recomendaciones que puedo hacer para tener mayor contexto en la toma de decisiones referida al negocio de alquileres de autos podrían ser:

* Tener la duración del alquiler para saber, en promedio, cuanto tiempo las personas utilizan los automóviles. Esto puede ayudar a entender el tipo de uso que le dan (por ejemplo, alquileres cortos para viajes de negocios vs. alquileres largos para vacaciones).
* Tener la cantidad de kilómetros que se recorren en cada alquiler. Esto podría ayudar a monitorear el desgaste de los automóviles y programar los mantenimientos.
* Categorizar los vehículos. Por ejemplo, económicos, de lujos, SUV, etc. Esto puede ofrecer insights sobre las preferencias de los clientes.
* Más allá del rating, sería útil contar con comentarios o feedback específico que los clientes proporcionen. Estos comentarios pueden ser muy valiosos para mejorar el servicio.

A partir de lo anterior, se podría segmentar a los clientes para entender y atender mejor sus necesidades. Detectando si ciertos modelos o categorías de vehículos son más populares, se podría considerar ajustar la composición de tu flota. Asimismo, sería más fácil detectar patrones en la duración de los alquileres o en los tiempos cuando la demanda es alta/baja, y se podría ajustar los precios u ofrecer promociones para maximizar los ingresos.

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

A screen shot of a computer

Description automatically generated

* Los datos provienen de un bucket S3 de AWS y de un sitio web público.
* Los datos se almacenan tempralmente en Google Cloud Storage para fácil acceso y procesamiento posterior.
* Para procesamiento y transformación utilizo Data Fusion.
* Una vez procesados los datos, los almaceno en BigQuery.
* Finalmente, exporto el o los csv a Looker para poder visualizar la información en diferentes charts
* Todo esto, claramente, orquestado por Cloud Composer