Descripción General

El siguiente documento tiene como objetivo explicar el proceso para la implementación de un pipeline de procesamiento y replicación de datos para centralizar la información de diversas sucursales de una aerolínea en un almacén de datos en BigQuery, utilizando servicios de la nube de Google Cloud. Los datos iniciales, que están en formato CSV, son almacenados en Google Cloud Storage (GCS) y posteriormente procesados mediante Dataflow. Este procesamiento incluye la lectura, parseo y carga de los datos en una base de datos MySQL.

Una vez almacenados en MySQL, se habilita un mecanismo de replicación de datos usando Datastream, una herramienta de captura de datos de cambios (CDC), que detecta las inserciones, actualizaciones y eliminaciones en la base de datos MySQL. Estos cambios son replicados en tiempo real a BigQuery, donde los datos son centralizados en un entorno de análisis y reporte.

El flujo completo asegura que los datos estén sincronizados y disponibles en BigQuery, permitiendo así que la empresa realice análisis más precisos y en tiempo real sobre su operación.

Componentes Principales del Proyecto:

1. **Google Cloud Storage (GCS)**: Almacena los archivos CSV provenientes de diferentes sucursales.
2. **Dataflow**: Procesa los archivos CSV, parseando la información y cargándola en MySQL.
3. **MySQL en Google Cloud**: Base de datos que almacena temporalmente los datos antes de la replicación.
4. **Datastream**: Herramienta de captura de datos de cambios (CDC) que replica los datos desde MySQL a BigQuery.
5. **BigQuery**: Almacén de datos final, donde se centraliza la información para análisis y reporte.

Este enfoque garantiza un flujo de datos confiable, escalable y automatizado que puede ser adaptado a futuros requerimientos de la aerolínea.

PROCESO

Creación del Bucket y carga de los archivos

Pasos

1. Crear el bucket aerline-storage:
    - Abrir la consola de Google cloud.
    - Ir a **Storage** en el menú de la consola.
    - Hacer clic en **Create Bucket.**
    - Nombrar el bucket como **aerline-storage.**
    - Elegir una región (recomendado: cercana a otros recursos como Dataflow y MySQL para minimizar latencias).
    - Configurar el tipo de almacenamiento y otras opciones, como el nivel de acceso público o privado.
    - Finalizar la creación del bucket.
2. Estructura del Bucket:
    - Crear una estructura de carpetas dentro del bucket para organizar los archivos CSV. La estructura es la siguiente:

aerline-storage/

└── data/

├── central/

│ └── CatLineasAereas.csv

├── sucursal1/

│ ├── Pasajeros.csv

│ └── Vuelos.csv

└── sucursal2/

├── Pasajeros.csv

└── Vuelos.csv

- Para crear estas carpetas:

    1. Navegar a GCS Storage en la consola.
    2. Ingresar al bucket **airline-storage**.
    3. Usar la opción **Create Folder** para crear las carpetas data, central, sucursal1 y sucursal2.
    4. Subir los archivos CSV a las carpetas correspondientes usando el botón Upload files.

3. Configurar Permisos (IAM):
    - Asegurar que las cuentas de servicio que necesitan acceder al bucket, como Dataflow, tengan los permisos adecuados.

- Para esto:

    1. Ir a la consola de IAM & Admin.
    2. Asignar los roles adecuados (como **Storage Object Viewer** y **Storage Object Admin**) a la cuenta de servicio que ejecutara Dataflow, asegurando que puedan leer y escribir en el bucket.
    3. Verificar que las políticas de permisos en el bucket permitan acceso seguro y necesario.

4. Verificar accesos:

    - Desde la consola, verificar que todos los archivos CSV están cargados correctamente y que las cuentas de servicio necesarias tienen acceso al bucket.

Resultado

El bucket **aerline-storage** ha sido creado con la estructura de carpetas correspondiente y los archivos CSV cargados correctamente. Las cuentas de servicio tienen los permisos necesarios para acceder al bucket, lo que permite continuar con los siguientes pasos del pipeline de procesamiento de datos.

Imagen 1 – Estructura del Bucket

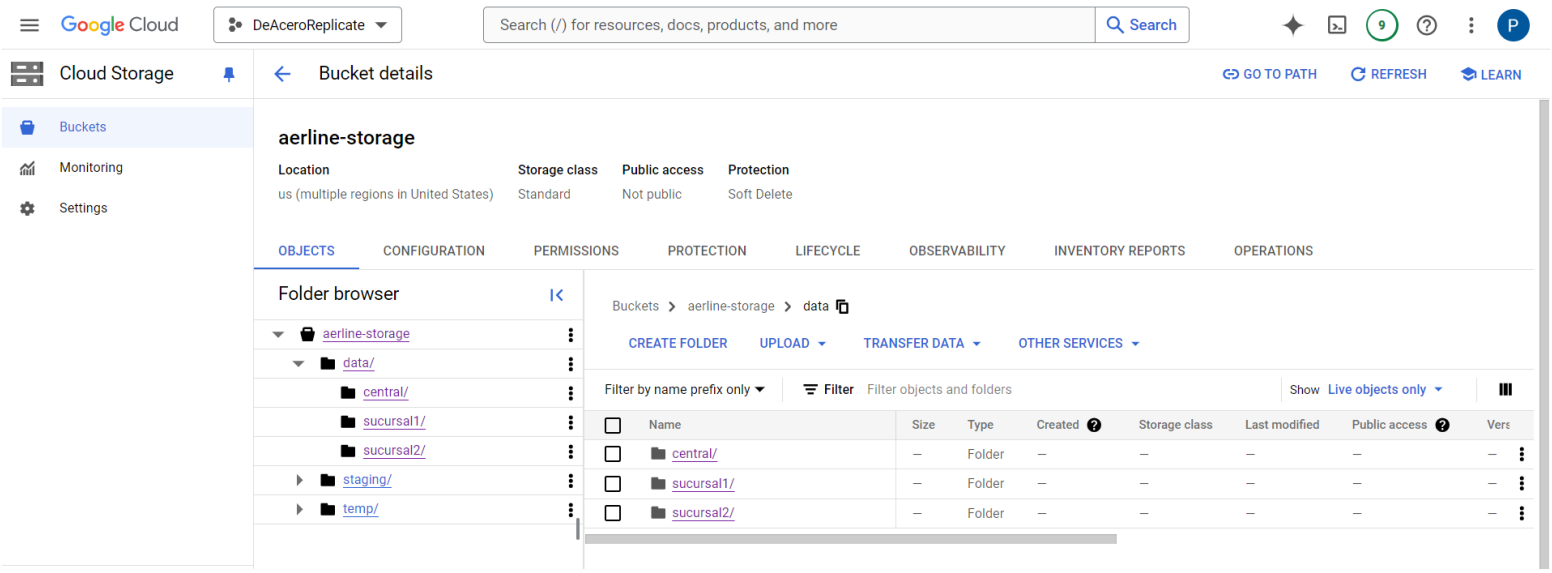


Imagen 2 – Carpeta data/central/

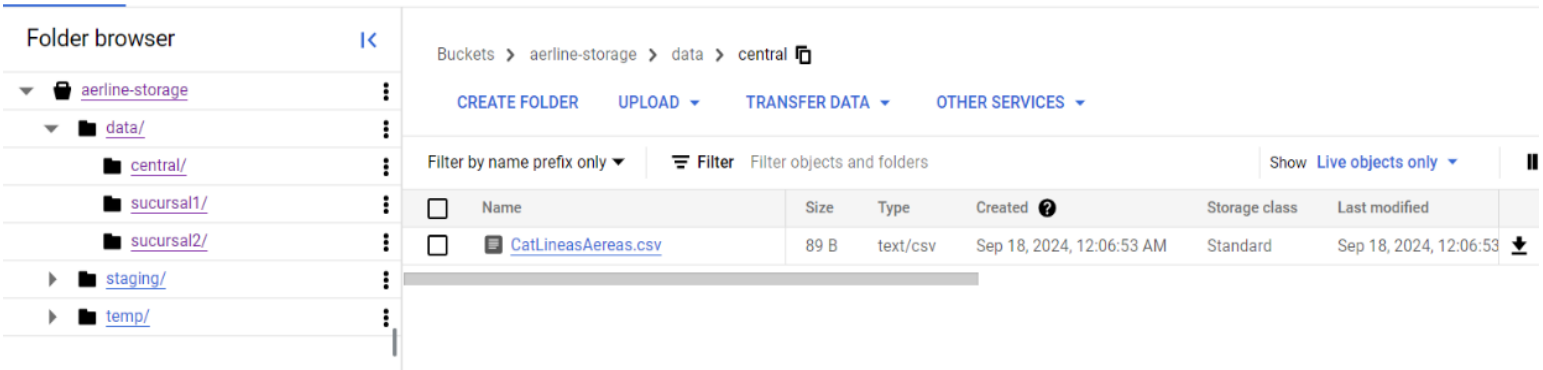


Imagen 3 – Carpeta data/sucursal1/

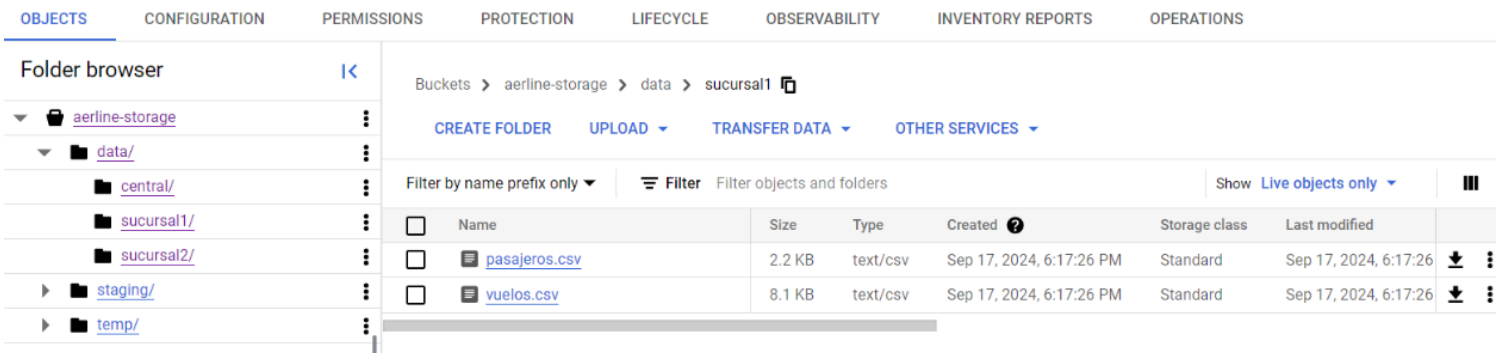
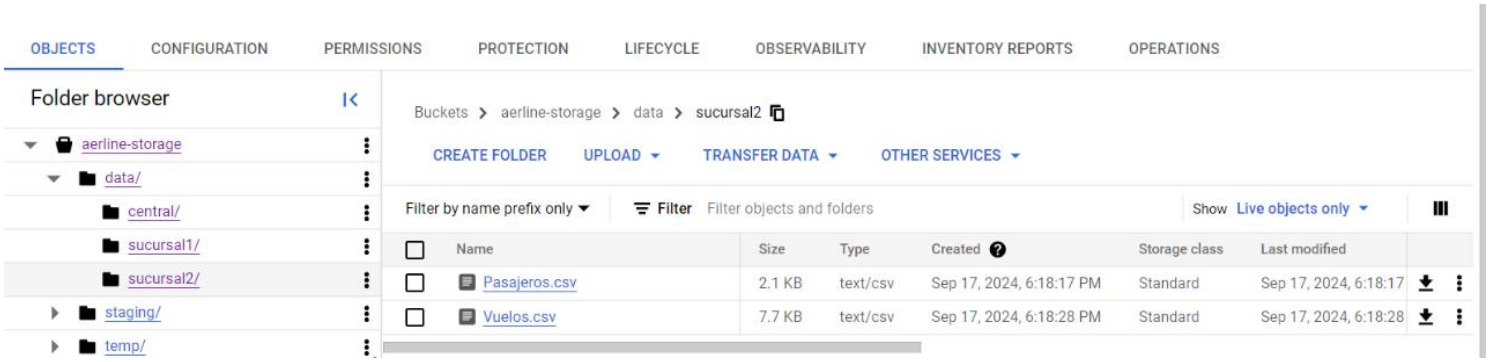


Imagen 4 – Carpeta data/sucursal2



Proceso con Dataflow

El segundo paso del proceso consiste en utilizar Apache Beam con Dataflow para procesas y transferir la información de los archivos CSV almacenados en Google Cloud Storage (GCS) hacia una base de datos MySQL. Esto permitió transformar la data desde archivos planos a una estructura relacional para su posterior replicación.

1. Lectura de archivos desde GCS

    - Utilizamos la función **ReadFromText** de Apache Beam para leer los archivos CSV almacenados en un bucket de GCS. Cada archivo contiene información sobre vuelos, líneas aéreas o pasajeros.
    - Los archivos procesados incluyen Vuelos.csv, CatLineasAereas.csv, y Pasajeros.csv.

2. Parseo de los archivos CSV

    - Mediante la función beam.Map y una función parse_csv, transformamos cada línea del archivo CSV en un formato legible para insertar en una base de datos.
    - Para cada archivo CSV, adaptamos el parse_csv para que convierta cada línea en un diccionario con los nombres de las columnas correspondientes.

3. Inserción en MySQL

    - Utilizamos el conector WriteToJdbc de Apache Beam para insertar los datos en una base de datos MySQL, especificando el nombre de la tabla de destino, la declaración de inserción (INSERT INTO) y los valores.
    - La conexión a la base de datos MySQL se realizó mediante JDBC, especificando el jdbc_url, nombre de usuario, contraseña, y el nombre de la tabla.

4. Flujo de datos

    - Se procesaron los archivos Vuelos.csv, CatLineasAereas.csv, y Pasajeros.csv, y sus respectivos datos fueron insertados en las tablas correspondientes en MySQL: sucursal1_vuelos, sucursal2_vuelos catlineasaereas, sucursal1_pasajeros y sucursal2_pasajeros.

Este proceso es un paso intermedio crucial, donde la data bruta de los archivos CSV es estructurada y almacenada en MySQL para un uso eficiente y escalable en el siguiente paso del pipeline, que involucra replicación de datos hacia BigQuery y utilizando DataStream.

Para tener mas contexto sobre el código implementado, revisar el archivo DataFlow.ipynb que se encuentra en el repositorio.

Base de datos en MySQL

El tercer paso del proceso consistió en configurar y gestionar una base de datos MySQL que recibiera la información procesada de los archivos CSV mediante Apache Beam y Dataflow. Una vez que los datos están en MySQL, se preparan para su replicación hacia BigQuery mediante Google Datastream.

1. Configuración de MySQL

    - Se configuró MySQL para aceptar la replicación y realizar la captura de datos de cambios (CDC). Esto incluye habilitar el formato adecuado del registro binario para CDC.
    - El archivo de configuración de MySQL (my.cnf) fue modificado para incluir las siguientes configuraciones esenciales:

        i. **log-bin:** Habilita los registros binarios, necesarios para la replicación.
        ii. **server-id:** Identifica el servidor de forma única en un entorno de replicación.
        iii. **binlog_format = ROW:** Establece el formato de registro binario en filas para capturar cada cambio.

iv. **log-slave-updates=true:** Permite que los cambios replicados también se escriban en los registros binarios.
v. **expire_logs_days=7:** Define que los registros binarios se mantengan por 7 días.

2. Creación de la base de datos y tablas

Se creo la instancia de MySQL para alojar la base de datos junto con las respectivas tablas. Al elegir una instancia de base de datos en Cloud SQL, Google Cloud ofrece tres opciones principales: **MySQL**, **PostgreSQL** y **SQL Server**. Se considero una instancia en MySQL en base al problema planteado y en las necesidades de este. Puntos para considerar:

- Compatibilidad con CDC: MySQL es ampliamente compatible con herramientas de Captura de Datos de Cambios (CDC), como **Datastream** de Google Cloud, lo que lo hace una opción ideal para implementar la replicación de datos en tiempo real.

- **Compatibilidad con Apache Beam y Dataflow**: MySQL es compatible con varias herramientas de procesamiento de datos como **Apache Beam** y **Google Dataflow**, que fueron esenciales para leer, procesar y cargar datos desde archivos CSV.

- **Buen rendimiento en lecturas y escrituras**: MySQL maneja eficientemente transacciones de lectura y escritura, lo que es ideal en el caso donde múltiples archivos CSV se cargan en la base de datos para luego ser replicados a BigQuery.

¿Por qué se acoplaba a la solución?

- **Soporte nativo para Datastream**: En este escenario se están replicando datos de MySQL hacia BigQuery usando **Datastream**, y MySQL es totalmente compatible con este servicio, lo cual facilitó la integración.

- **Facilidad de integración con servicios de Google Cloud**: MySQL tiene buena integración con otros servicios de Google Cloud como **Cloud SQL**, **Google Cloud Storage** y **BigQuery**, haciendo más fácil la manipulación y replicación de datos.

- **Gestión simple del binlog**: MySQL soporta la configuración y gestión del registro binario (binlog), lo cual es esencial para realizar CDC, y permitía implementar fácilmente la replicación de datos.

PASOS

- Se crearon las tablas correspondientes en MySQL para almacenar los datos de los archivos CSV. Estas tablas incluyen:

    i. sucursal1_vuelos: Para almacenar los datos sobre los vuelos de la sucursal 1.

ii. sucursal2_vuelos: Para almacenar los datos sobre los vuelos de la sucursal 2.
iii. catlineasaereas: Para almacenar las líneas aéreas disponibles.
iv. sucursal_pasajeros: Para almacenar los datos de los pasajeros de la sucursal 2.

- Las tablas fueron creadas con las columnas necesarias para almacenar los datos de los archivos CSV, como Sucursal, Cve_LA, Pasajero, Edad, entre otros.

3. Conexión y transferencia de datos desde Dataflow

    - Como se mencionó en el paso anterior, los datos fueron cargados directamente en las tablas de MySQL mediante la herramienta de Apache Beam. Cada archivo fue parseado y sus datos insertados en la tabla correspondiente.
    - El proceso de carga de datos fue exitoso, dejando la base de datos MySQL preparada para su uso y replicación.

4. Preparación para la replicación mediante Datastream

    - Se habilitaron las configuraciones necesarias para permitir que Datastream capturara los cambios en MySQL. Esto incluyó asegurarse de que el registro binario estuviera configurado en formato ROW y que las actualizaciones de los esclavos estuvieran habilitadas para la replicación.
    - MySQL fue reiniciado para asegurarse de que los cambios en el archivo de configuración tomaran efecto.

5. Verificación de las configuraciones en MySQL

    - Se verificaron las configuraciones de MySQL necesarias para la replicación mediante los siguientes comandos:

        i. **SHOW GLOBAL VARIABLES LIKE '%binlog_format%';** : Verifica que binlog_format esté configurado en ROW.

        ii. **SHOW GLOBAL VARIABLES LIKE 'binlog_row_image':** : Verifica que binlog_row_image esté configurado en FULL.

        iii. **SHOW GLOBAL VARIABLES LIKE 'log_slave_updates';** : Verifica que log_slave_updates esté habilitado.

        iv. **SHOW GLOBAL VARIABLES LIKE 'expire_logs_days';** : Verifica que el período de retención de los registros binarios esté configurado en 7 días.

Este paso fue crucial para centralizar los datos en una base de datos relacional y preparar MySQL para la replicación mediante Google Datastream, que se realiza en el siguiente paso.

Imagen 5 – Acceso a MySQL mediante consola: Para acceder a MySQL se puede usar el comando, considerando que el host depende de la instancia.

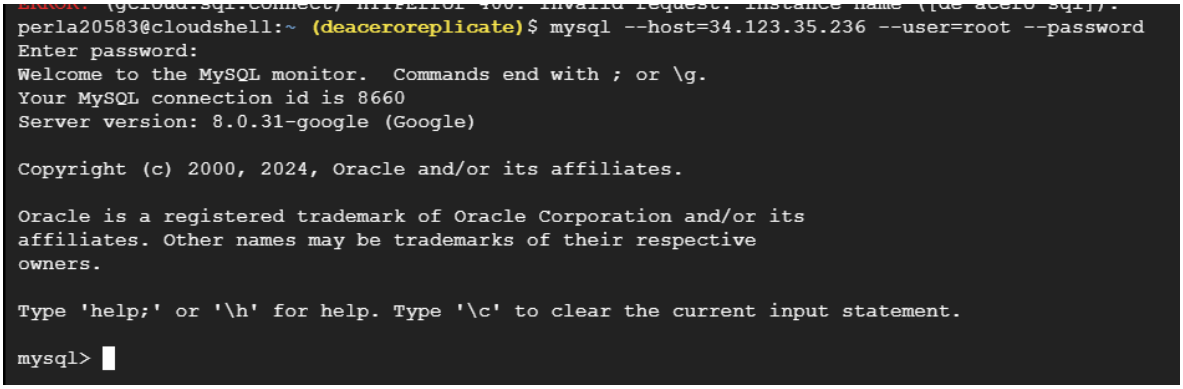


Imagen 6 – Uso de la base de datos db_aerolineas junto con el describe de las tablas para la sucursal 1.

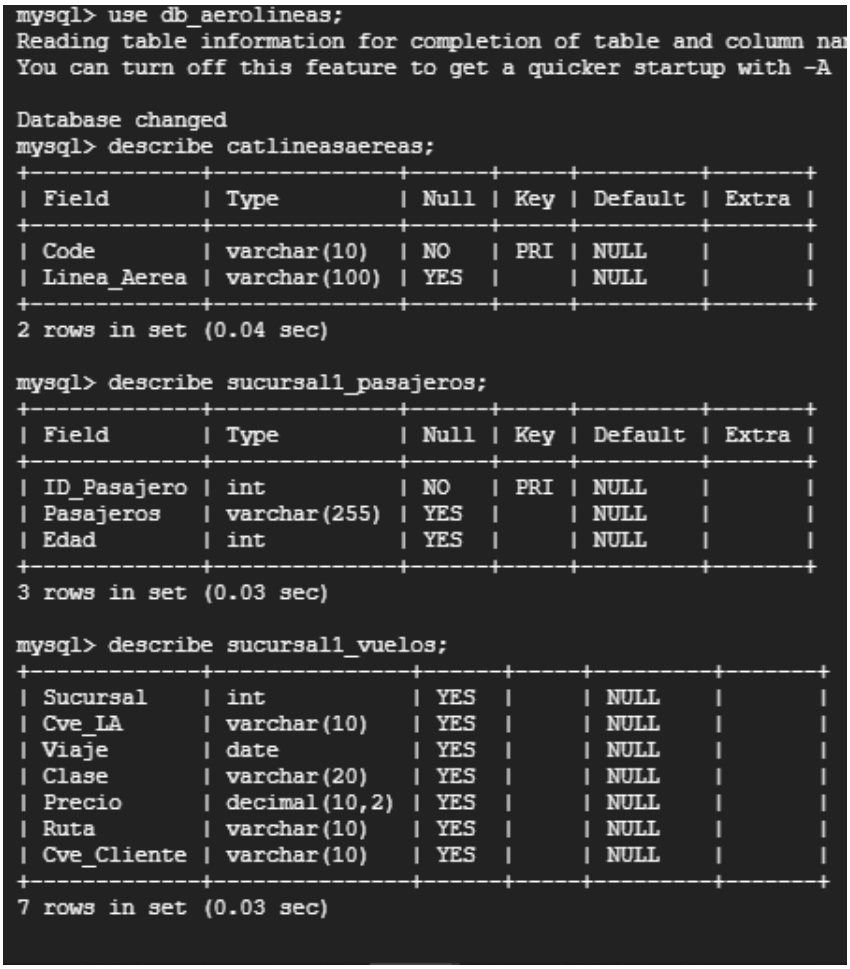
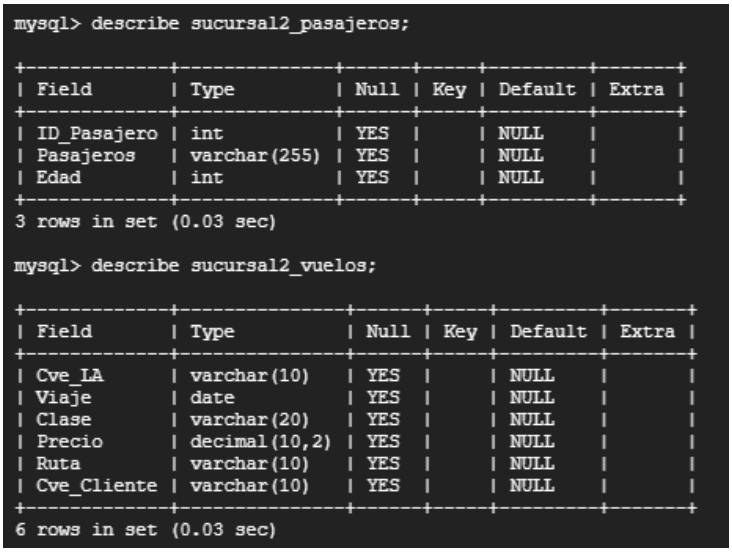


Imagen 7 – Describe de las tablas para la sucursal 2.



```
mysql> describe sucursal2_pasajeros;

+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| ID_Pasajero | int          | YES  |     | NULL    |       |
| Pasajeros   | varchar(255) | YES  |     | NULL    |       |
| Edad        | int          | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
3 rows in set (0.03 sec)

mysql> describe sucursal2_vuelos;

+-------------+---------------+------+-----+---------+-------+
| Field       | Type          | Null | Key | Default | Extra |
+-------------+---------------+------+-----+---------+-------+
| Cve_LA      | varchar(10)   | YES  |     | NULL    |       |
| Viaje       | date          | YES  |     | NULL    |       |
| Clase       | varchar(20)   | YES  |     | NULL    |       |
| Precio      | decimal(10,2) | YES  |     | NULL    |       |
| Ruta        | varchar(10)   | YES  |     | NULL    |       |
| Cve_Cliente | varchar(10)   | YES  |     | NULL    |       |
+-------------+---------------+------+-----+---------+-------+
6 rows in set (0.03 sec)
```

Uso de Datastream para replicacion de datos

En este paso, se utilizó **Google Cloud Datastream** para replicar los datos desde **MySQL** hacia **BigQuery**, implementando la Captura de Datos de Cambios (CDC). A continuación se detalla el proceso seguido para configurar y utilizar Datastream, que permite la replicación de datos en tiempo real.

1. Configuración del registro binario en MySQL

   Antes de configurar Datastream, fue necesario habilitar el registro binario (binlog) en MySQL para poder realizar la captura de datos de cambios. Estos son los pasos clave que se llevaron a cabo:

   - **Habilitar el registro binario** en MySQL añadiendo los siguientes parámetros en el archivo de configuración de MySQL (my.cnf):

     [mysqld]
     log-bin=mysql-bin
     server-id=1
     binlog_format = ROW
     log-slave-updates = true
     expire_logs_days = 7

     Esto permite que MySQL registre los cambios de datos (inserciones, actualizaciones, eliminaciones) en formato de fila. Si se tiene algún conflicto con este proceso, al momento de crear un Stream y seleccionar MySQL como source, se muestra estas especificaciones para la configuración:

Imagen 8 – Configuración

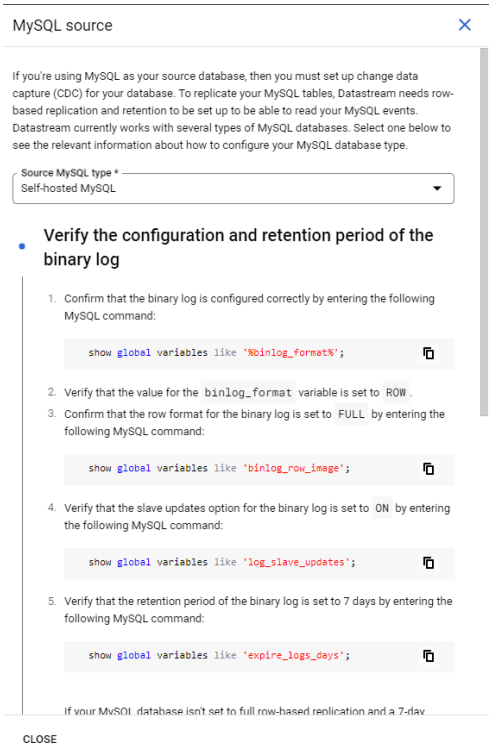
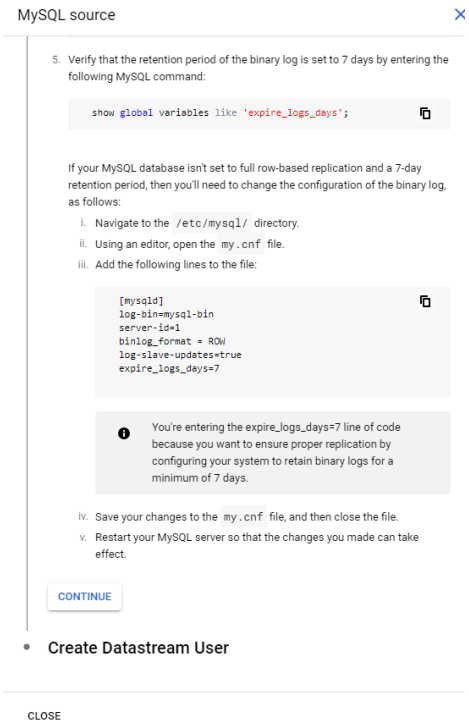


Imagen 9 – Configuración archivo



- **Reiniciar el servidor MySQL** para aplicar los cambios y verificar que el registro binario se configuró correctamente con los siguientes comandos:

  i. SHOW GLOBAL VARIABLES LIKE '%binlog_format%';
  ii. SHOW GLOBAL VARIABLES LIKE 'binlog_row_image';

2. Configuración de Datastream

   Una vez configurado MySQL, el siguiente paso fue crear un flujo en **Datastream** para capturar y replicar los datos en tiempo real hacia **BigQuery**. El proceso fue el siguiente:

   - Crear un perfil de conexión a MySQL:

     i. Se accedió a la interfaz de **Google Cloud Console** y se navegó al servicio de **Datastream**.
     ii. En la sección de **Perfiles de conexión**, se creó un nuevo perfil para MySQL, proporcionando la siguiente información:

       - **Nombre del perfil**: Un identificador único para el perfil de MySQL.
       - **Host**: La dirección IP o el nombre del host de la instancia de MySQL.
       - **Puerto**: Generalmente el puerto estándar de MySQL (3306).
       - **Usuario y contraseña**: Credenciales del usuario de MySQL.

- Crear un perfil de conexión a BigData

    - Se creó otro perfil de conexión, esta vez para **BigQuery**, proporcionando acceso a la tabla de destino en la que se replicarán los datos.
    - Se seleccionó el dataset existente en **BigQuery** llamado deaceroreplicate como destino.

3. Definir el flujo de DataStram

    - Se creó un nuevo flujo de Datastream para replicar los datos de MySQL a BigQuery, configurando los parámetros siguientes:

        i. **Fuente**: La base de datos MySQL configurada en el perfil de conexión.
        ii. **Destino**: El dataset en BigQuery configurado previamente.
        iii. **Objetos a replicar**: Se seleccionaron las tablas específicas de la base de datos MySQL que debían replicarse (por ejemplo, sucursal1_vuelos, CatLineasAereas, Pasajeros).
        iv. **Captura de datos de cambios (CDC)**: Se habilitó la opción de replicar los cambios en tiempo real.

4. Especificacion de destino en BigQuery

    - Se especifico el nombre del proyecto destino, como el ID del mismo. Este id se puede visualizar seleccionando el poyecto en la parte superior.

Imagen 10 – ID del proyecto

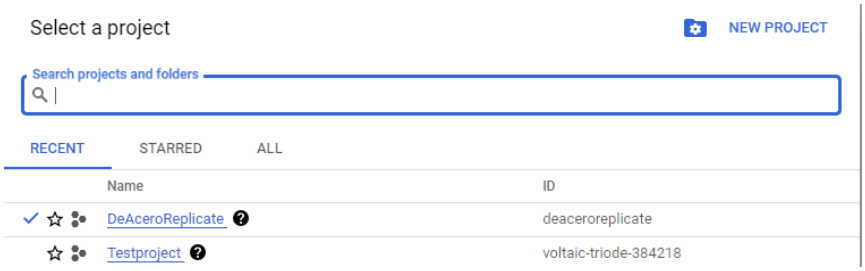
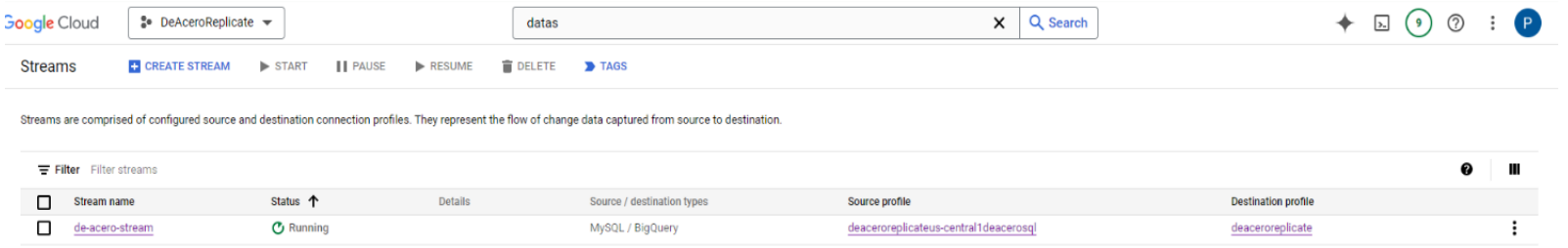


Imagen 11 – Stream creado



Monitoreo del flujo de Datastream

Después de configurar el flujo de replicación, se monitoreó el proceso para asegurarse de que los datos estuvieran replicándose correctamente. Google Cloud Datastream ofrece herramientas de monitoreo y logs que permiten verificar:

- **La latencia de replicación**: Qué tan rápido se están replicando los datos desde MySQL a BigQuery.
- **Errores en la replicación**: Cualquier error relacionado con la replicación de datos o la conexión entre las bases de datos.

Consideraciones adicionales en la configuración:

- **Seguridad de la conexión**: Durante la configuración de los perfiles de conexión, se habilitaron las opciones de conexión segura, utilizando SSL para cifrar la comunicación entre MySQL y Datastream.
- **Gestión de usuarios y permisos**: Se otorgaron los permisos necesarios al usuario de MySQL para que Datastream pudiera acceder a las tablas y leer los registros de binlog.

Imagen 12 – Observabilidad / Login (Explorador de registros)

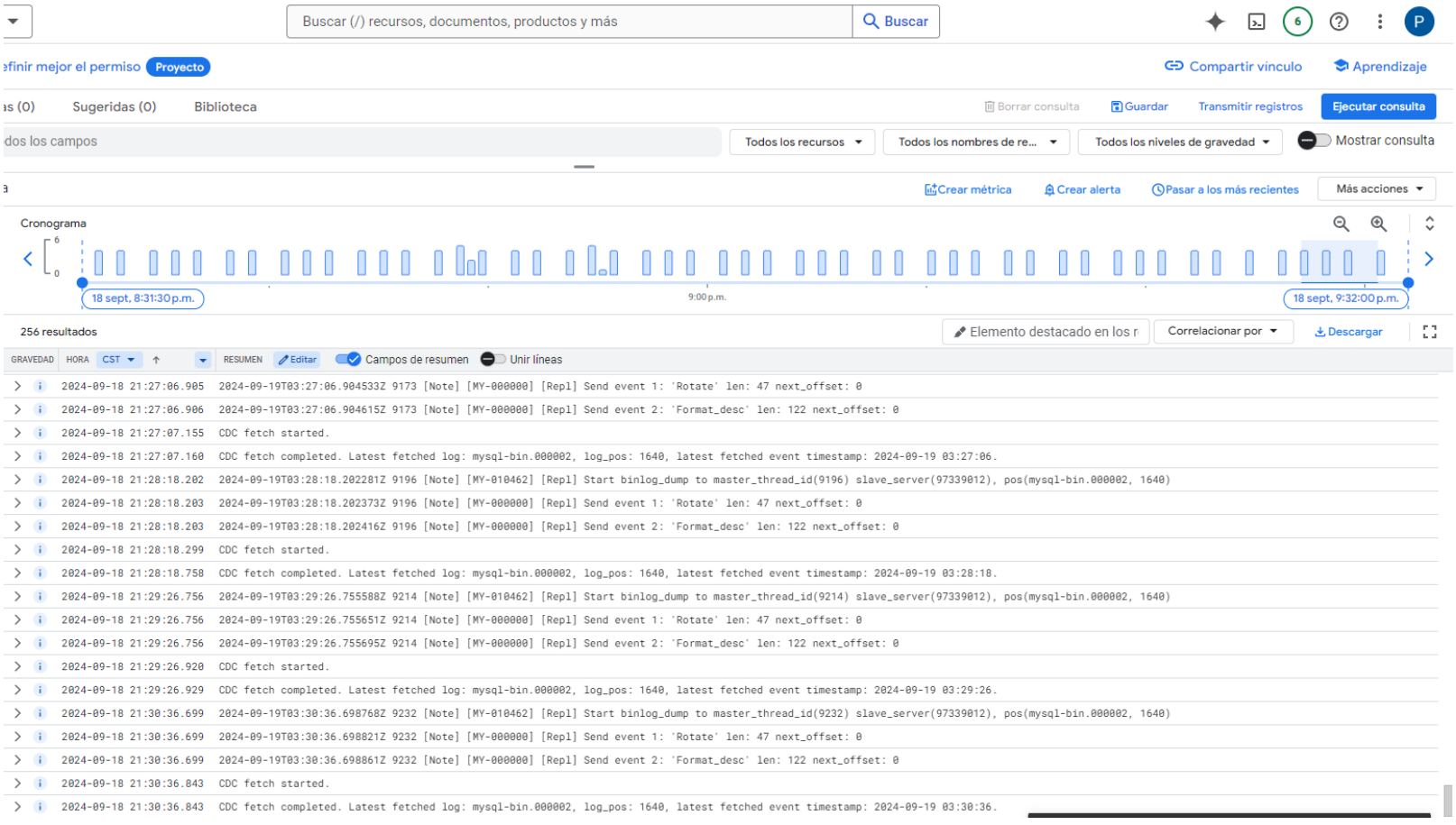


Imagen 12 – CDC Fetch

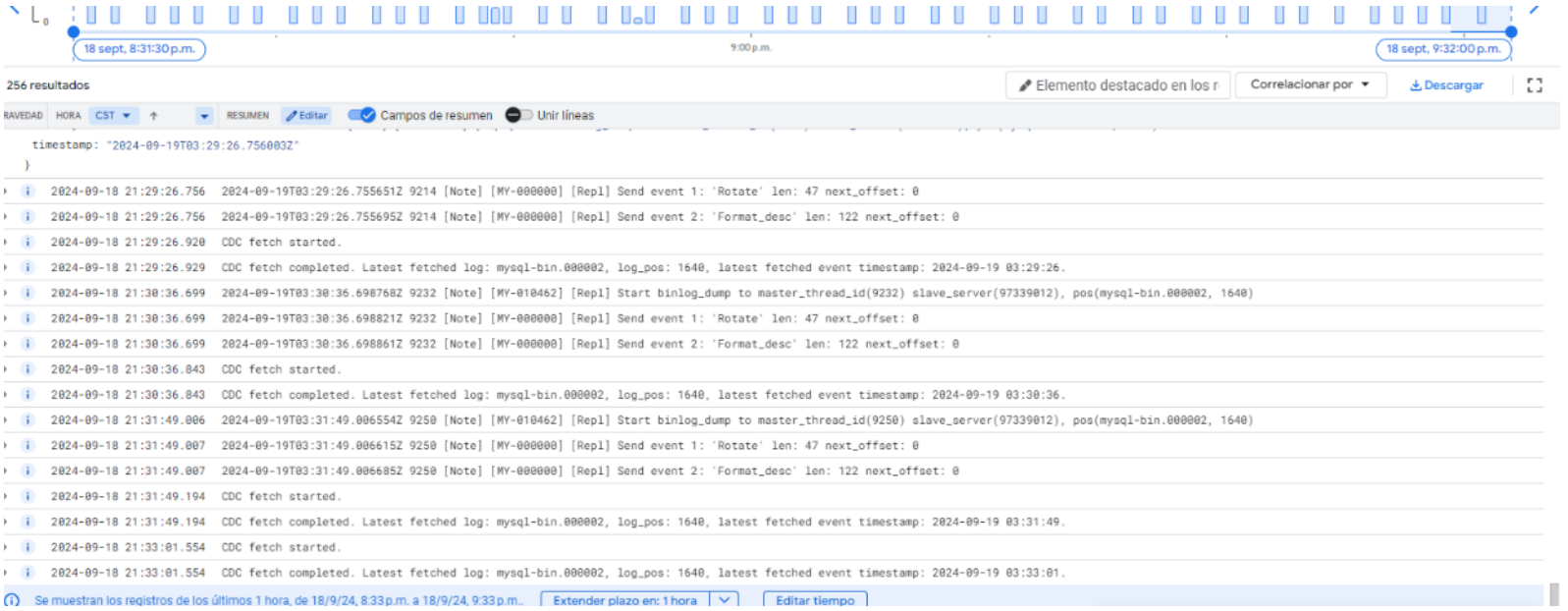


Imagen 13 – Tablas ejecutadas

de-acero-stream

MySQL / BigQuery

Stream ID	de-acero-stream
Source profile	deaceroreplicate:us-central1:deacerosql
Destination profile	DeAceroReplicate
Created	Sep 18, 2024, 4:15:48 PM
Updated	Sep 18, 2024, 4:36:04 PM

OVERVIEW | MONITORING | OBJECTS

5 tables

Filter Enter property name or value

Object name	Object status	Backfill status	Event count (7 days)	Data processed (30 days)		
db_aerolineas.catlineasaereas	Running	Completed	5	211	INITIATE BACKFILL	VIEW DETAILS
db_aerolineas.sucursal1_pasajeros	Running	Completed	100	6251	INITIATE BACKFILL	VIEW DETAILS
db_aerolineas.sucursal1_vuelos	Running	Completed	0	0	INITIATE BACKFILL	VIEW DETAILS
db_aerolineas.sucursal2_pasajeros	Running	Completed	100	6171	INITIATE BACKFILL	VIEW DETAILS
db_aerolineas.sucursal2_vuelos	Running	Completed	200	22571	INITIATE BACKFILL	VIEW DETAILS

Lectura de datos en BigQuery

El paso final del proceso es la lectura y consulta de los datos que han sido replicados en BigQuery desde MySQL a través de **Datastream**. Una vez que los datos están disponibles en BigQuery, se pueden utilizar para realizar consultas SQL y análisis. A continuación, se describe el proceso para consultar los datos replicados en BigQuery.

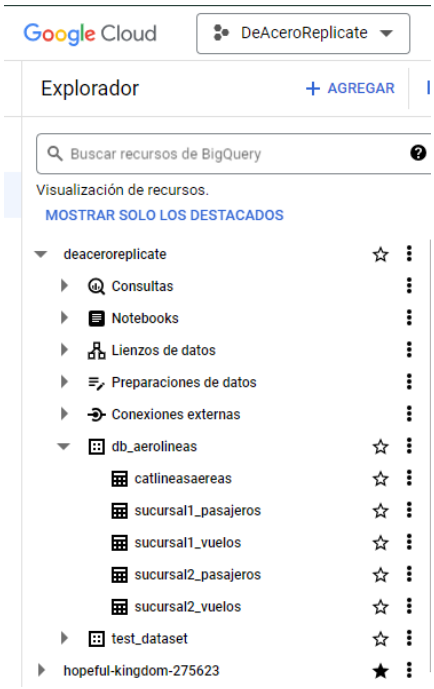
1. Acceder a BigQuery

    - Inicia sesión en Google Cloud Console y navega al servicio de BigQuery.
    - Selecciona el proyecto correspondiente (en este caso, deaceroreplicate).

2. Visualización de los datasets

    - Dentro de BigQuery, los datos replicados desde MySQL están almacenados en el dataset deaceroreplicate.
    - Al expandir el dataset, se pueden ver las tablas que Datastream ha replicado, como sucursal1_vuelos, CatLineasAereas, y Pasajeros.

Imagen 14 – Dataset

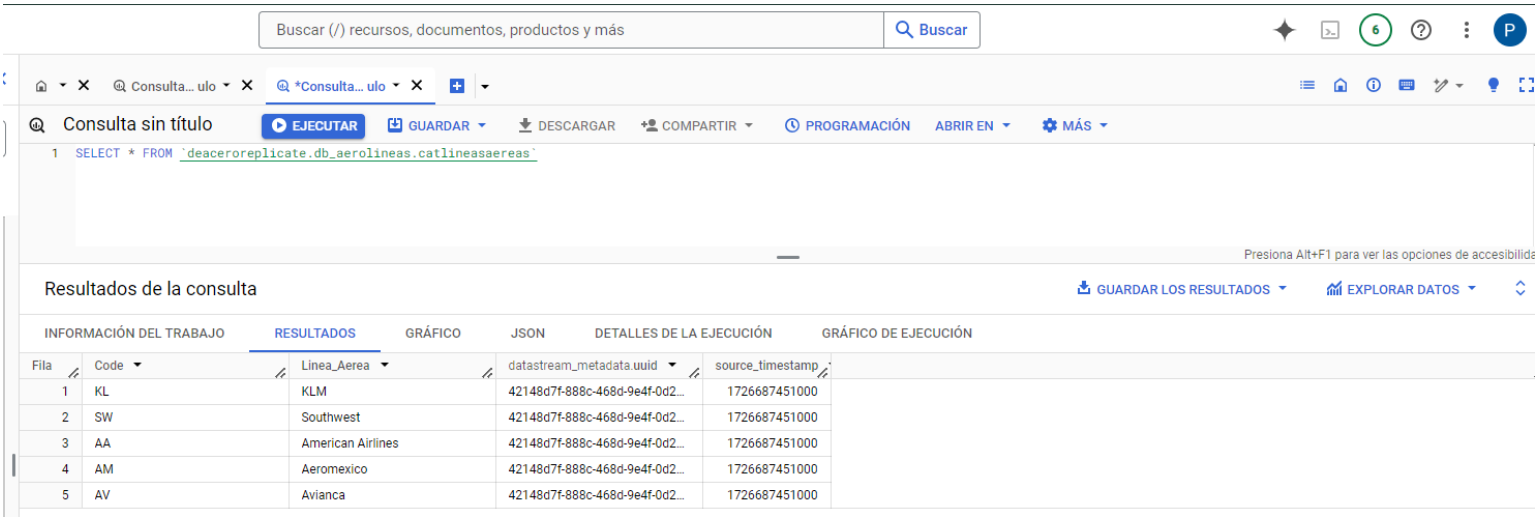


3. Realización de consulta SQL

    - Para consultar los datos, se puede utilizar BigQuery SQL, un lenguaje muy similar al estándar SQL.
    - Para consultar los datos de las líneas aéreas, se usó el siguiente query:

```
SELECT * FROM `deaceroreplicate.db_aerolineas.catlineasaereas`
```

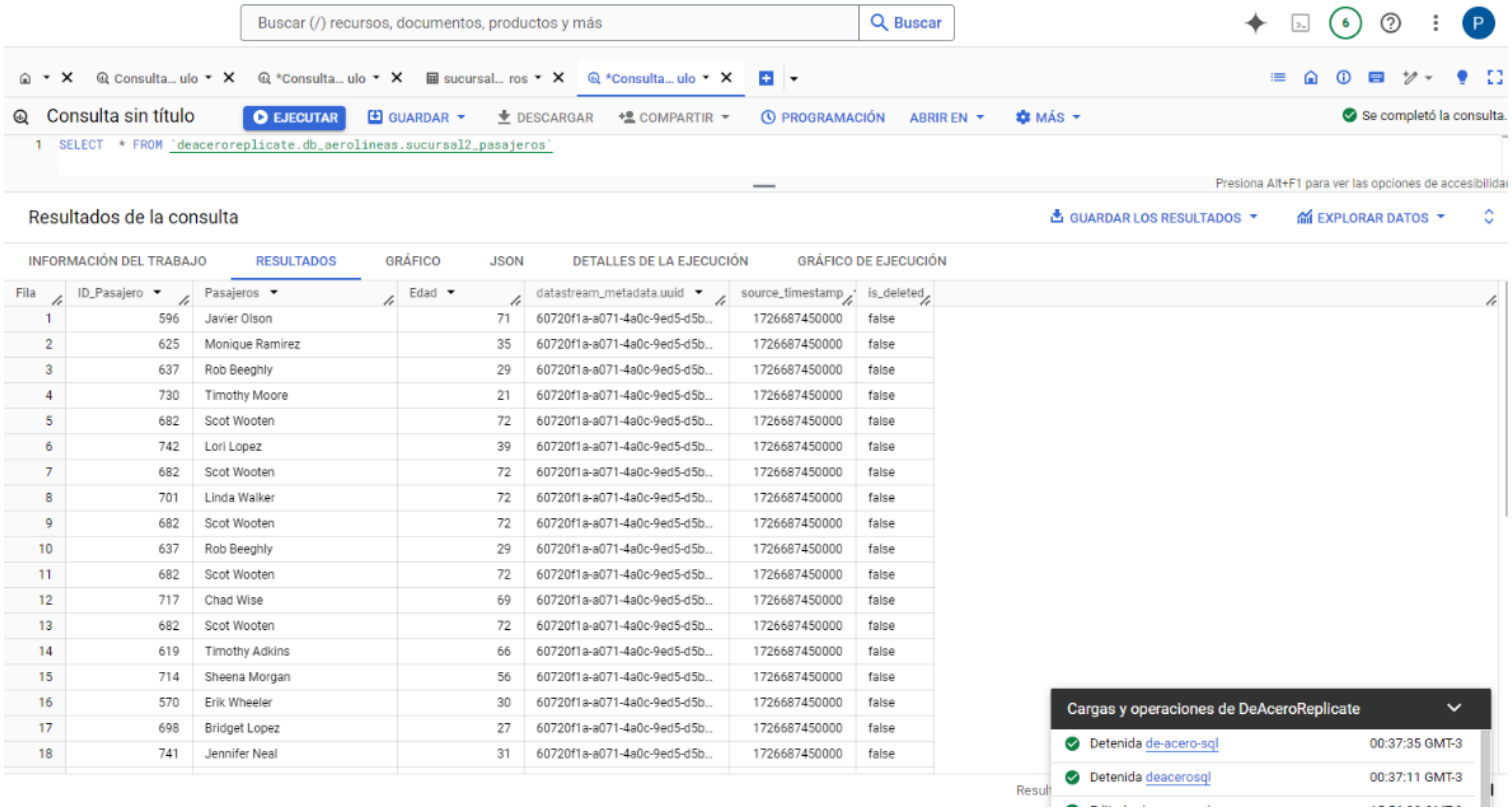
Imagen 15 – Query líneas aéreas



- Para consultar los datos de los pasajeros, se usó el siguiente query:

```
SELECT * FROM `deaceroreplicate.db_aerolineas.sucursal2_pasak¿jeros`
```

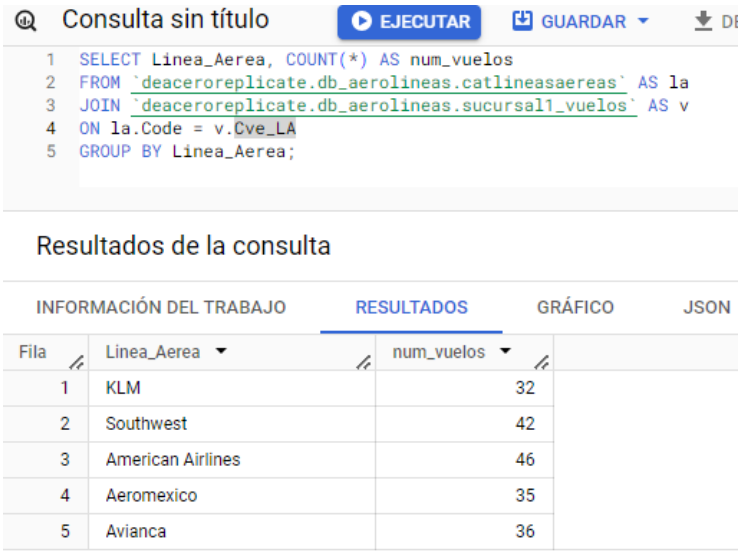
Imagen 16 – Query pasajeros



- Para consultar el número de vuelos por cada línea aérea, se usó el siguiente query:

```
SELECT Linea_Aerea, COUNT(*) AS num_vuelos
FROM `deaceroreplicate.db_aerolineas.catlineasaereas` AS la
JOIN `deaceroreplicate.db_aerolineas.sucursal1_vuelos` AS v
ON la.Code = v.Cve_LA
GROUP BY Linea_Aerea;
```

Imagen 17 – Número de vuelos



Fila	Linea_Aerea	num_vuelos
1	KLM	32
2	Southwest	42
3	American Airlines	46
4	Aeromexico	35
5	Avianca	36

Monitoreo y optimización

- **Costos de consulta**: Las consultas en BigQuery son altamente escalables, pero es importante monitorear el uso para evitar costos excesivos. BigQuery cobra en función del volumen de datos procesados, por lo que siempre es útil optimizar las consultas utilizando filtros adecuados y seleccionando solo las columnas necesarias.
- **Particionamiento y Clustering**: Si las tablas contienen grandes volúmenes de datos, se puede optar por utilizar técnicas como el particionamiento o el clustering para mejorar el rendimiento de las consultas.

Conclusión

El proyecto logró establecer un flujo de datos completo desde la captura inicial en GCS hasta la replicación en BigQuery, garantizando la integridad y calidad de los datos en cada etapa. Las configuraciones realizadas en GCS, Dataflow, MySQL, Datastream y BigQuery aseguran un proceso robusto y eficiente para el manejo de datos, facilitando el análisis y la toma de decisiones basadas en información precisa y actualizada.