Proyecto Airlines

Limpieza de Datos

Se recibe el dataset **"Airlines"**, para facilitar el análisis, las tablas fueron convertidas a formato CSV, generando ocho archivos que están en evaluación para su uso final, los mismos son:

- aircrafts_data
- airports_data
- boarding_passes
- bookings
- flights
- seats
- tickets
- ticket_flights

Importación de Librerías y Preparación del Entorno

El análisis comenzó importando las librerías necesarias: **numpy**, **pandas** y **os**.

Utilizando Google Colab, montamos Google Drive para almacenar los archivos, y leímos los archivos CSV en un diccionario donde cada clave corresponde al nombre de la tabla.

Proceso de Limpieza

➢ Aircrafts_data

Descripción: Contiene 3 columnas y 9 registros sobre aeronaves.

Problema: La columna “model” estaba en formato JSON con nombres en inglés y ruso.

Solución: Se extrajo el valor en inglés (en) y el dataframe resultante quedó con 9 modelos únicos, sin valores nulos.

➢ Airports_data

Descripción: Contiene 5 columnas y 104 registros sobre aeropuertos.

Problema: Las columnas airport_name y city contenían cadenas en formato JSON.

Solución: Se extrajo el nombre en inglés en ambas columnas. El dataframe quedó sin valores nulos y correctamente tipado.

➢ Boarding_passes

Descripción: 579,686 registros y 4 columnas de pases de abordaje.

Solución: No se encontraron valores nulos ni duplicados. Se utilizó “.describe()” para obtener un resumen estadístico del dataframe.

➢ Bookings

Descripción: 3 columnas y 262,788 registros sobre reservas de vuelos.

Problema: La columna book_date estaba en formato string.

Solución: Se convirtió a formato datetime. No había valores nulos ni duplicados, y se guardó nuevamente el CSV.

➢ Flights

Descripción: Contiene 10 columnas y 33,121 registros de vuelos.

Problema: Las columnas que contienen fechas y horas (scheduled_departure, scheduled_arrival, etc.) estaban en formato string con valores /N en lugar de nulos.

Solución:

- ✓ Los valores /N se reemplazaron por NaT (nulos en datetime).
- ✓ Se dividieron las columnas de fecha y hora en campos separados para facilitar el análisis.
- ✓ Se aplicó forward fill (ffill) para rellenar los valores nulos con los anteriores válidos.
- ✓ Se eliminaron las primeras 26 filas, ya que no tenían valores válidos previos.

Creación de la base de datos local (o inicial)

Después de limpiar y transformar los datos, se creó una base de datos relacional en **SQL Server**.

Las tablas procesadas incluyen:

- aircrafts_data
- airports_data
- boarding_passes
- bookings,
- flights,
- tickets
- ticket_flights

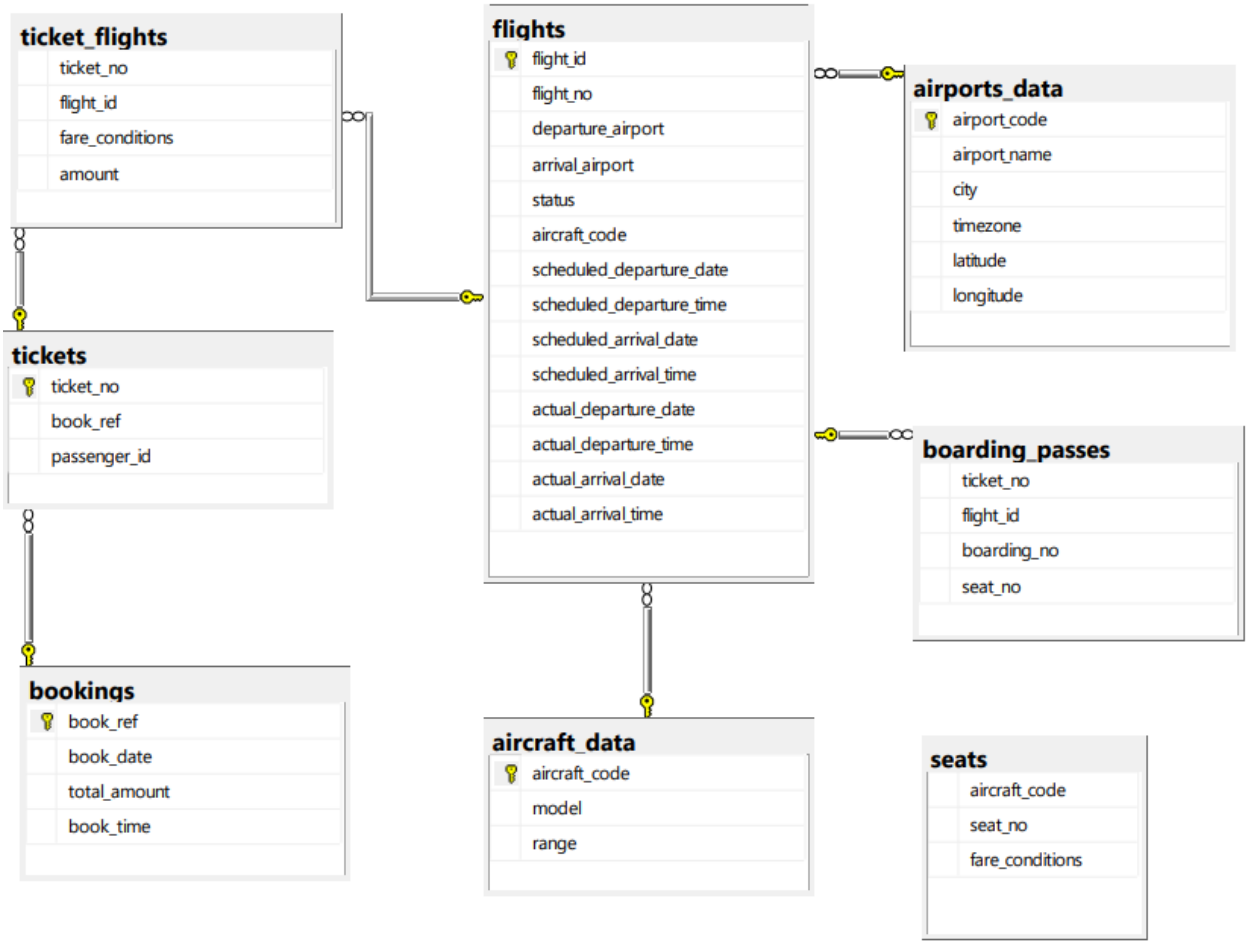
Cada archivo CSV fue importado directamente a SQL Server, asegurando la coherencia en las estructuras y tipos de datos.

Se verificaron las claves primarias y las relaciones entre las tablas, lo que fue fundamental para asegurar la integridad referencial. También se utilizaron índices en las columnas clave para optimizar el rendimiento de las consultas.

Establecimiento de Relaciones y Diagrama Entidad-Relación (ERD)

Se establecieron relaciones entre las tablas utilizando las claves primarias y foráneas, como la relación entre flights y airports_data a través de los códigos de aeropuerto, y entre boarding_passes y flights mediante el número de vuelo. También se vincularon tickets y ticket_flights a bookings.

Se generó un **Diagrama Entidad-Relación (ERD)** para visualizar estas conexiones, lo que ayuda a identificar cómo los datos están interrelacionados y es esencial para consultas complejas.



Nota: la tabla SEAT no posee una relación directa con el análisis a realizar por lo tanto se dejará fuera del mismo.