**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.

A screenshot of a computer

Description automatically generated

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