UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

DESARROLLO Y GESTIÓN DE SOFTWARE

REPORTE DE SOLUCIÓN DE CASO DE ESTUDIO DE TÉCNICAS DE LIMPIEZA DE DATOS

EXTRACCIÓN DE CONOCIMIENTO EN BASES DE DATOS

PRESENTA:

KARLA ALEJANDRA DE LA CRUZ ZEA

DOCENTE:

ING. LUIS ENRIQUE MASCOTE CANO

12 de octubre de 2025

Contenido

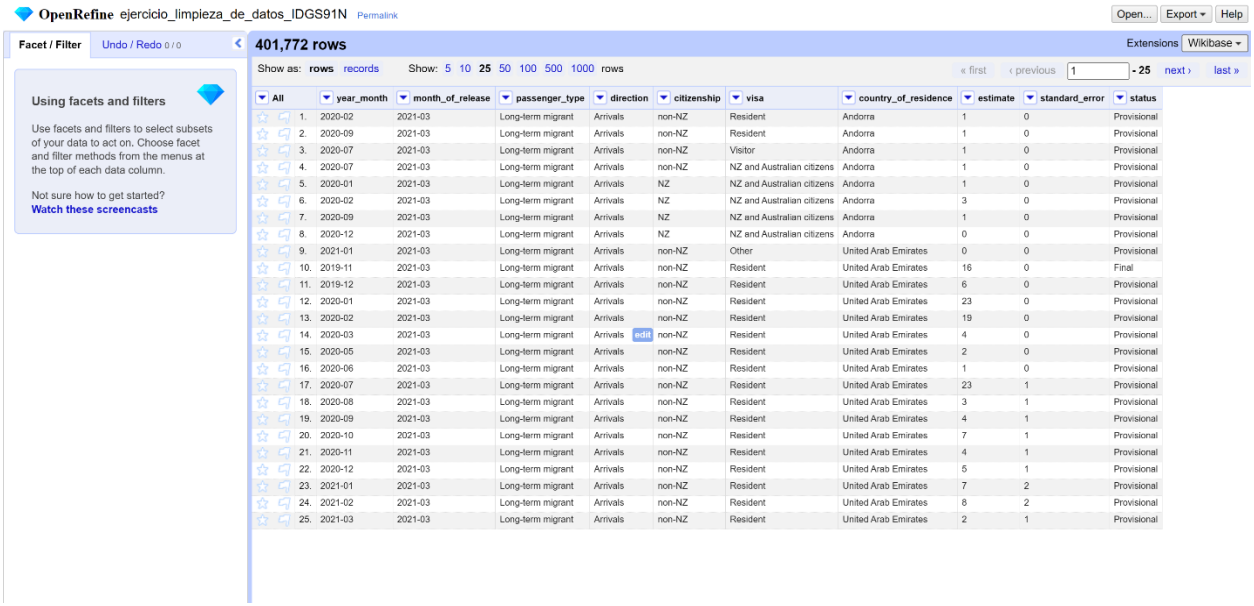
1. Introducción

La introducción debe ser breve y sintetizada pero bien explicada.

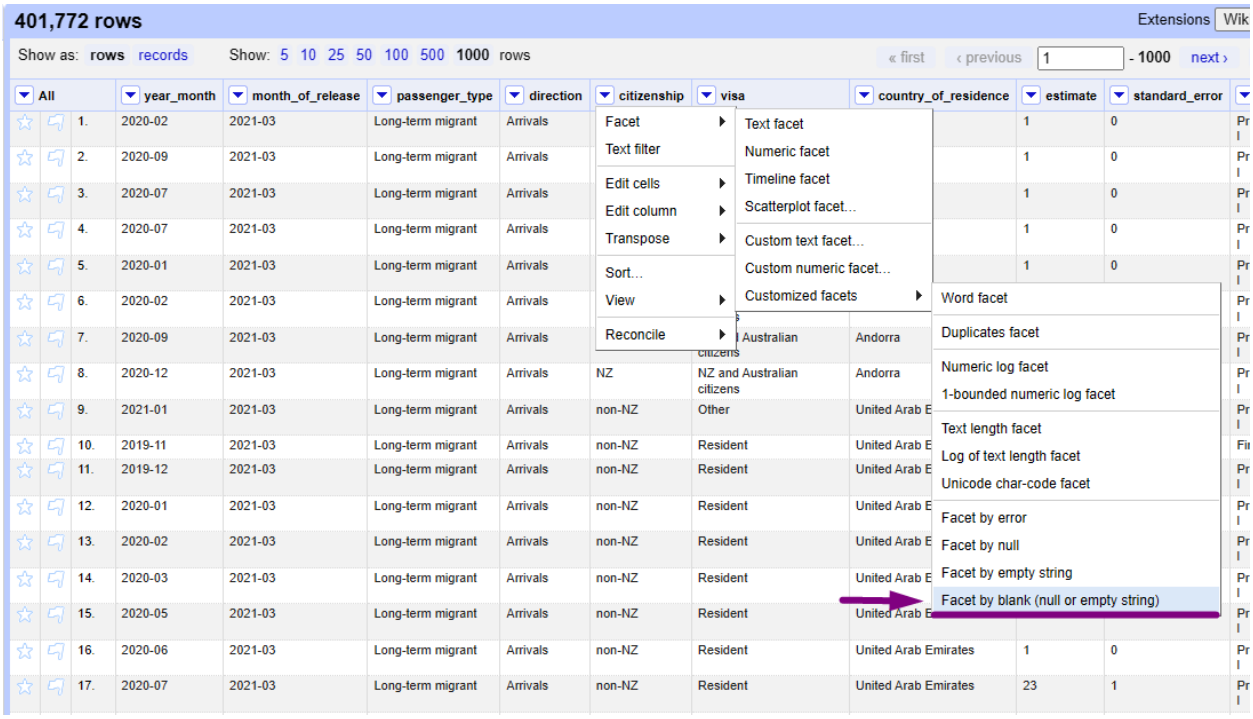
En esta practica mostrare como realizar una limpieza de datos con un programa que se llama Open Refine, es sencillo pero cumple bien con su función.

2. Limpieza de datos

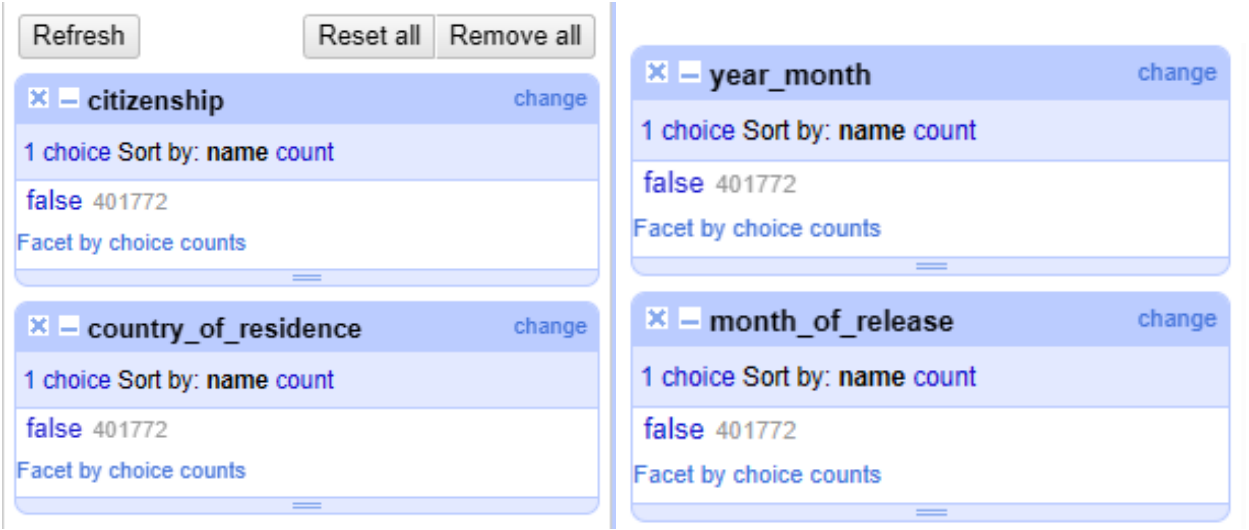
Para comenzar a identificar los valores faltantes, las inconsistencias de formato y los duplicados utilizare el programa de Open refine, el cual permite que se puedan hacer filtrado de inconsistencias.



Comenzamos revisando si tenemos espacios en blanco. Utilizando las facetas personaliazdas revisamos si cada fila tiene espacios en blanco o nulos.



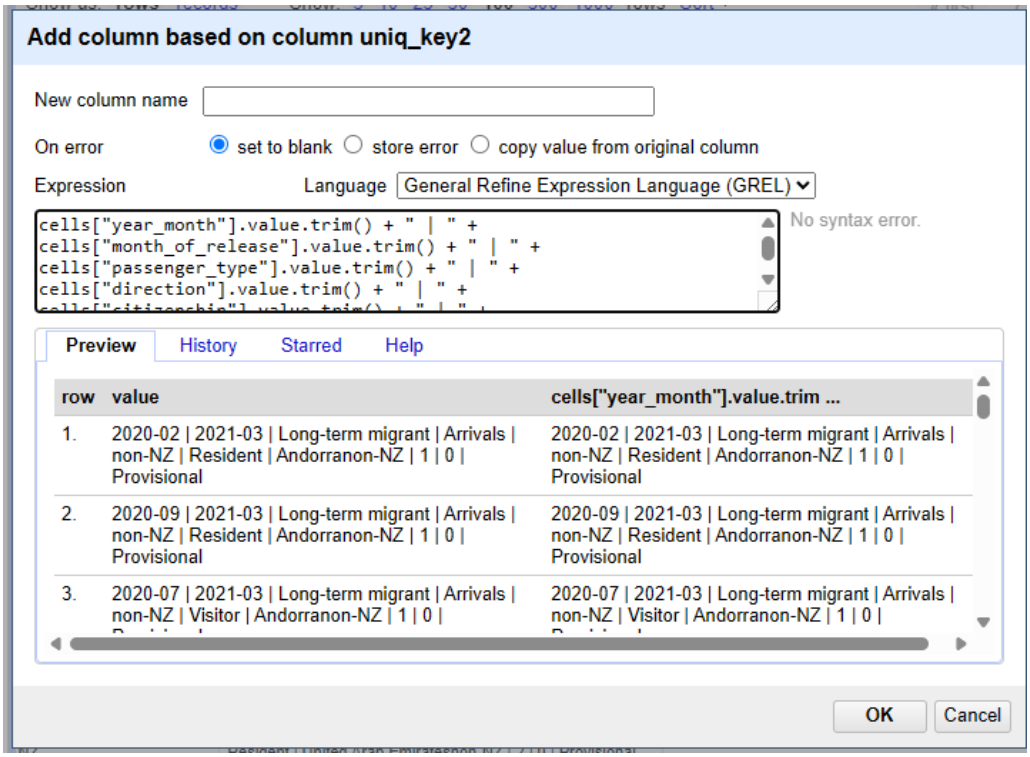
Se realizo el filtro en todas las filas y el resultado fue falso, es decir que no contamos con espacios en blanco o nulos. **false** → registros completos.



Preparacion de la llave compuesta

Ahora para detectar duplicados vamos a crear una llave compuesta a la que vamos a llamar uniq_key en donde vamos a ingresar las columnas a filtrar. Usando el lenguaje GREL:

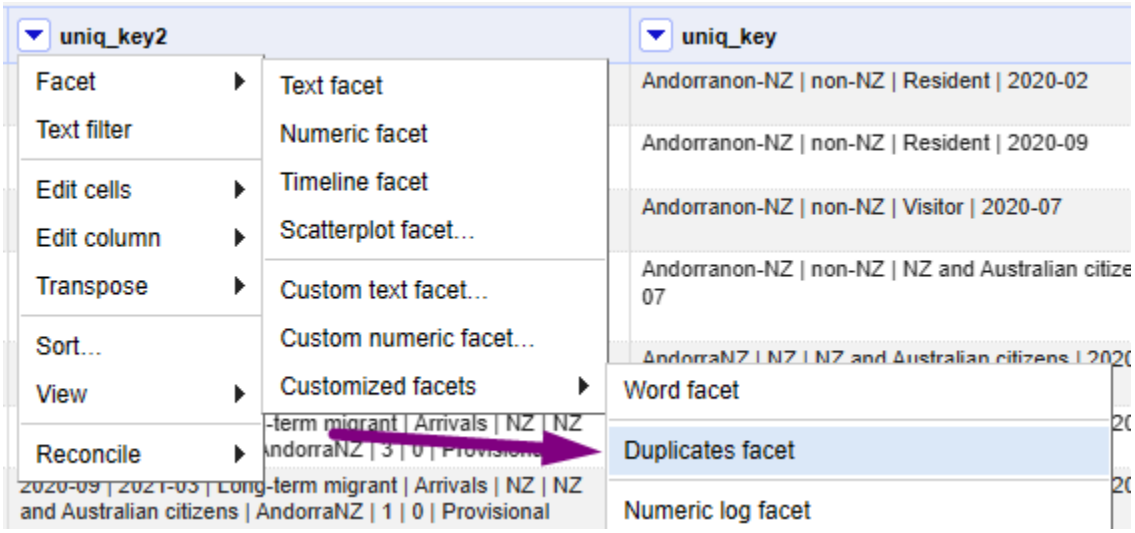
```
cells["year_month"].value.trim() + " | " +
cells["month_of_release"].value.trim() + " | " +
cells["passenger_type"].value.trim() + " | " +
cells["direction"].value.trim() + " | " +
cells["citizenship"].value.trim() + " | " +
cells["visa"].value.trim() + " | " +
cells["country_of_residence"].value.trim() + " | " +
cells["estimate"].value.trim() + " | " +
cells["standard_error"].value.trim() + " | " +
cells["status"].value.trim()
```



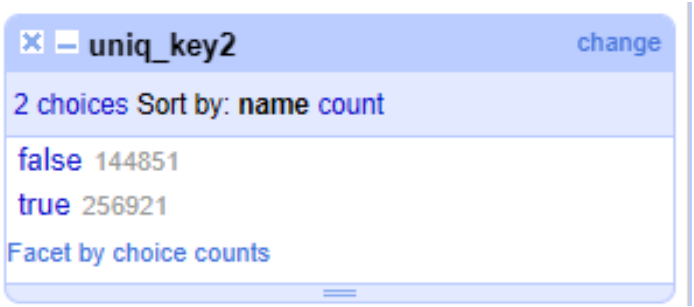
Resultado de la creación de la columna.

Identificación de valores duplicados

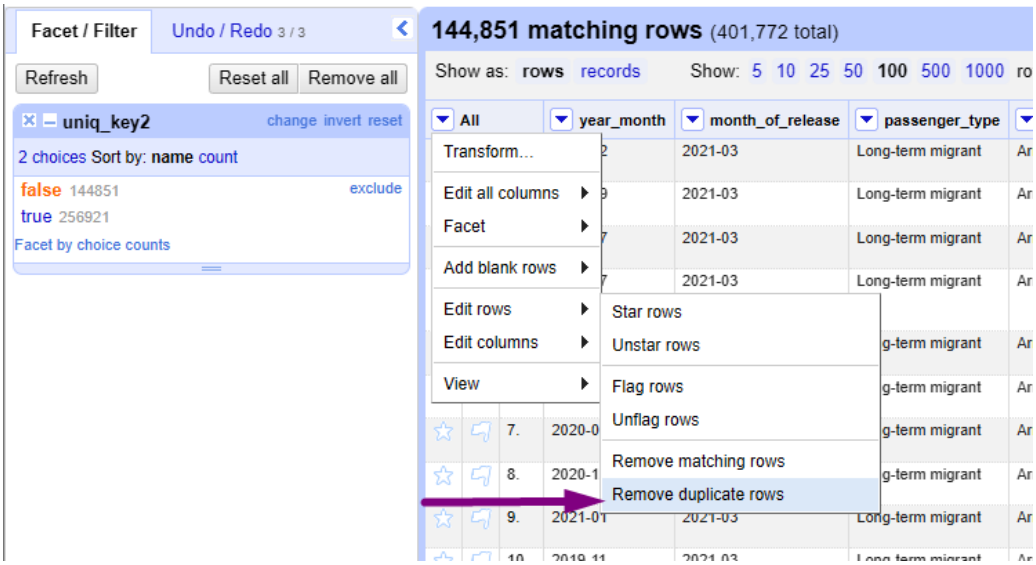
Ahora vamos a identificar aquellos valores que tengamos duplicados, utilizando el filtro de facetas en la columna que acabamos de crear el resultado es el siguiente:

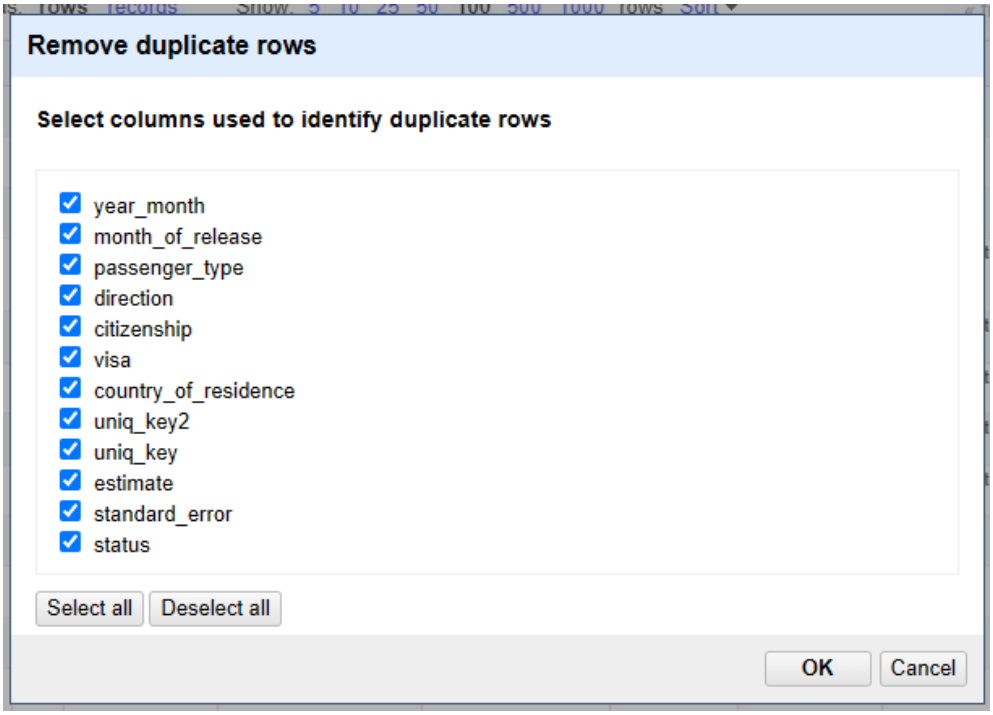


Y como resultado podemos observar que efectivamente tenemos valores duplicados



Ahora procederemos a eliminar las filas que son iguales





13 filas fueron afectadas

3. Determinación de hechos y dimensiones

Definición

En un sistema de *data warehouse*, las tablas de hechos almacenan los eventos o medidas cuantitativas que se analizan (por ejemplo, cantidades o valores), mientras que las tablas de dimensiones contienen los atributos descriptivos que permiten contextualizar esos hechos (por ejemplo, país, tipo de visa, año o categoría de migrante).

Justificación

El conjunto de datos sobre **migración internacional** contiene información estructurada por país, ciudadanía, tipo de visa, año y número de migrantes.
Para un *data warehouse* que analice movimientos migratorios, se pueden identificar las siguientes estructuras:

Tabla de hechos:

Hecho principal:

Fact_Migration_Flows
Contendrá los datos cuantitativos del número de personas migrantes por combinación de país, ciudadanía y tipo de visa.

Medidas principales:

migration_count → cantidad total de migrantes registrados.

year → periodo de registro (dimensión temporal).

Tablas de dimensiones:

Dim_Country

Describe el país de residencia o destino.

Atributos: country_id, country_name, continent, income_level.

Propósito: permitir análisis geográficos y comparaciones regionales.

Dim_Citizenship

Atributos: citizenship_id, citizenship_name, region_origin.

Propósito: identificar nacionalidades y analizar flujos por origen.

Dim_VisaType

Atributos: visa_id, visa_category, description.

Propósito: analizar migraciones según tipo de visa o permiso (trabajo, estudio, residencia, etc.).

Dim_Time

Atributos: time_id, year, quarter, month.

Propósito: permitir análisis por periodos y tendencias temporales.

4. Normalización y almacenamiento

Modelo relacional 3FN

Tabla	Campos principales	Llave primaria	Llaves foráneas
Fact_Migration_Flows	fact_id, country_id, citizenship_id, visa_id, time_id, migration_count	fact_id	FK: country_id, citizenship_id, visa_id, time_id
Dim_Country	country_id, country_name, continent, income_level	country_id	—
Dim_Citizenship	citizenship_id, citizenship_name, region_origin	citizenship_id	—
Dim_VisaType	visa_id, visa_category, description	visa_id	—
Dim_Time	time_id, year, quarter, month	time_id	—

Script SQL del modelo relacional

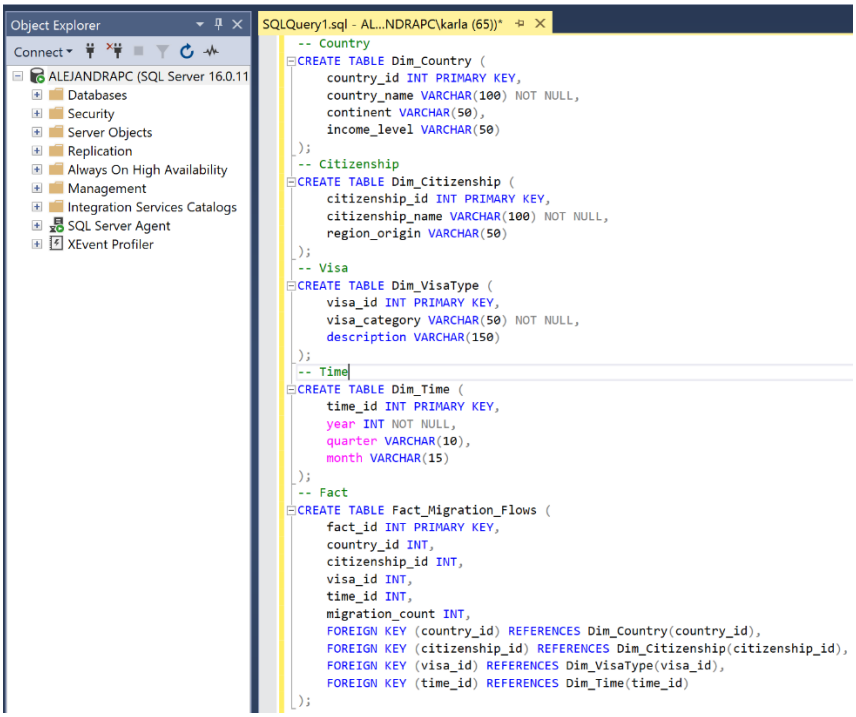
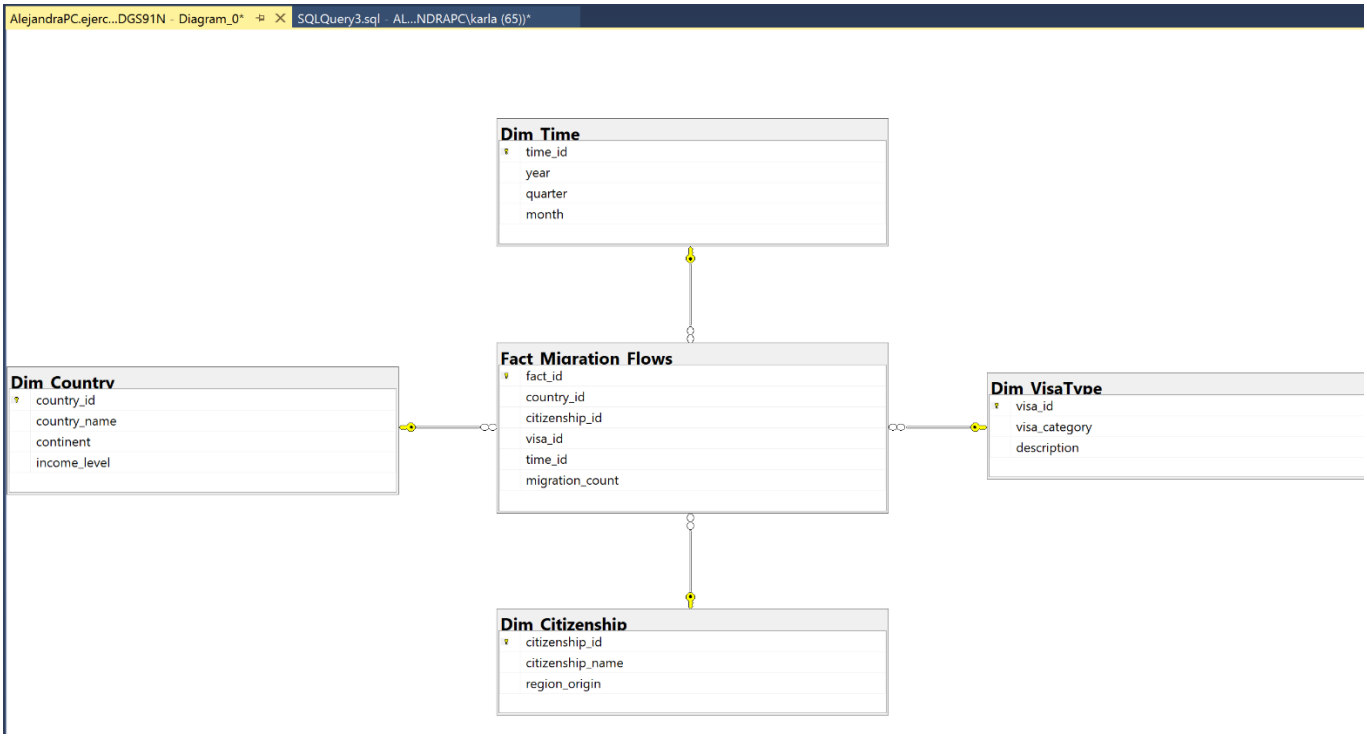


Diagrama lógico (esquema estrella)



5. Conclusión

El trabajo con OpenRefine permitió limpiar y preparar correctamente los datos para poder analizar o diseñar un Data Warehouse. A través de la detección de valores faltantes, duplicados y errores de formato, se logró mejorar la calidad del conjunto de datos de migración internacional, haciéndolo más confiable y útil para su almacenamiento y análisis.

El diseño del modelo en forma de estrella ofrece una estructura clara que facilita consultas, reportes y comparaciones entre países, tipos de visa y periodos. Además, la normalización hasta la tercera forma normal evita redundancia, mantiene consistencia y permite crecer el sistema en el futuro sin perder orden.

6. Referencias

OpenRefine. (2024). *OpenRefine Documentation: Working with messy data*. Recuperado de https://docs.openrefine.org

Van Hooland, S., Verborgh, R., & De Wilde, M. (2013, 5 agosto). ***Cleaning Data with OpenRefine*. Programming Historian**. https://programminghistorian.org/en/lessons/cleaning-data-with-openrefine#facetting-and-clustering

OpenAI. (2025). *ChatGPT definition guidance for data analysis and documentation*.