UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN

II.3. REPORTE DE LIMPIEZA DE DATOS

MATERIA: Extracción de Conocimiento en Bases de Datos

MAESTR@: Enrique Mascote

MATRÍCULA: 1123250015

ALUMNO: Carlos Adrián Mata Nevárez

GRUPO: IDGS91N

FECHA:12/10/2025

ÍNDICE

INTRODUCCIÓN

El este documento se detalla el proceso de preparación y modelado de datos utilizando el conjunto de información sobre migración internacional titulado international-migration-March-2021-citizenship-by-visa-by-country-of-last-permanent-residence.csv.

Este dataset abarca registros clave sobre los flujos migratorios, incluyendo la ciudadanía, el tipo de visa, el país de última residencia permanente y las cantidades registradas por periodo.

El objetivo central de este informe es documentar las metodologías de limpieza (data cleaning) y modelado aplicadas para transformar los datos brutos en información estructurada, apta tanto para análisis directo como para la integración en un Data Warehouse (DW).

OBJETIVOS DEL PROYECTO

1.Calidad de Datos: Identificar y aplicar tratamientos rigurosos para abordar valores faltantes, inconsistencias de formato y registros duplicados dentro del conjunto de datos.

2.Documentación: Registrar de manera clara y exhaustiva cada problema de calidad de datos detectado, especificando la técnica o lógica de corrección que fue implementada.

3.Modelado Analítico: Proponer y definir la estructura de hechos y dimensiones que mejor soporte las consultas analíticas dentro del entorno de un Data Warehouse.

4.Diseño Lógico: Diseñar un modelo relacional normalizado (al menos a Tercera Forma Normal, 3FN) y generar el script SQL necesario para su implementación.

LIMPIEZA DE DATOS

Listado de problemas y técnicas aplicadas.

El conjunto contiene 401,772 registros y 10 columnas, con información sobre migración internacional, ciudadanía, tipo de visa y país de residencia.

Descripción general del conjunto de datos

Num.	Columna	Tipo de dato	Descripción
1	year_month	Object	Año y mes del evento migratorio.
2	month_of_release	Object	Fecha de publicación (año-mes).
3	passenger_type	Object	Tipo de pasajero (ej. Long-term migrant).
4	direction	Object	Dirección del flujo (Arrivals / Departures).
5	citizenship	Object	Nacionalidad declarada.
6	visa	Object	Tipo de visa (Resident, Visitor, Student, etc.).
7	country_of_residence	Object	País de última residencia permanente.
8	estimate	Int64	Estimación del número de migrantes.
9	standard_error	Int64	Error estándar asociado a la estimación.
10	status	Object	Estado del dato (Final o Provisional).

IDENTIFICACIÓN DE PROBLEMAS

Identificación de problemas detectados

Durante la inspección inicial se detectaron los siguientes detalles.

Tipo de problema	Columnas afectadas	Descripción	Técnica aplicada
Valores faltantes	Ninguna	Todas las columnas tienen valores completos (401,772 registros con datos no nulos).	No se requiere acción.
Duplicados	Varias	Se observaron posibles duplicados entre combinaciones de year_month, citizenship, visa, y country_of_residence.	Eliminación con drop_duplicates().
Inconsistencia de formato	year_month, month_of_release	Ambas columnas representan fechas como cadenas (YYYY-MM), sin tipo de dato datetime.	Conversión a formato fecha estándar (datetime64).
Valores genéricos ("TOTAL")	citizenship, visa, country_of_residence	Algunos registros usan “TOTAL” como valor agregado, no individual.	Filtrado o recategorización según análisis.

Estandarización textual	visa, status	Mayúsculas y espacios inconsistentes.	Normalización con .str.strip().str.title().
Valores atípicos	estimate	Algunos valores extremos (muy altos) pueden representar totales.	Normalización con .str.strip().str.title().

TÉCNICAS DE LIMPIEZA APLICADAS

Normalización de nombres de columnas

Se eliminaron espacio y se estandarizaron a formato snake_case (ya correcto en el archivo original) (Torres, 2025), (McKee, 2024).

Conversión de fechas

```
# conversion de fechas
df['year_month'] = pd.to_datetime(df['year_month'], format='%Y-%m')
df['month_of_release'] = pd.to_datetime(df['month_of_release'], format='%Y-%m')
```

Con esto, ambas columnas quedan listas para análisis temporal.

Eliminación de duplicados

```
# eliminar duplicados exactos
before = len(df)
df = df.drop_duplicates()
after = len(df)
removed = before - after
```

Se eliminaron 5,642 registros duplicados exactos.

Registros finales 396,130.

Filtrado de valores agregados ("TOTAL")

Los registros con “TOTAL” en citizenship, visa o country_of_residence representan sumas y no observaciones individuales.

```
# eliminar filas con 'Total' en columnas categóricas
df = df[
    (df['citizenship'] != 'Total') &
    (df['visa'] != 'Total') &
    (df['country_of_residence'] != 'Total')
]
```

Generación del archivo limpio

```
# guardar limpio
df.to_csv('international_migration_clean.csv', index=False)
```

Guarda el archivo limpio.

HECHOS Y DIMENSIONES

Para diseñar un data warehouse dimencional (modelo estrella), con tablas de ejemplo basadas en el contenido del CSV.

Tablas de hecho y dimenciones propuestas

Hecho: fact_migration

- Propósito: almacenar medidas cuantitativas por combinación de dimensiones (año, ciudadanía, tipo de visa, país de última residencia). Sirve para análisis de flujos migratorios, conteos y tendencias.
- Medidas: count_people (SUM), record_count (número de registros), posiblemente source_rows (control).

Claves foráneas a dimensiones: dim_time_id, dim_citizenship_id, dim_visa_type_id, dim_last_residence_id, dim_country_code_id (si aplica).

Dimensión: dim_time

- Propósito: manejar año/mes/día/periodo. Facilita análisis por periodos.
- Atributos: time_id (PK), year, month, quarter, date_label.

Dimensión: dim_country (ciudadanía)

- Propósito: catalogar países (citizenship).
- Atributos: country_id (PK), country_name, country_iso2, country_iso3, region, subregion.

Dimensión: dim_visa_type

- Propósito: normalizar tipos de visa (work, student, temporary, permanent, etc.).
- Atributos: visa_type_id (PK), visa_type_name, visa_category.

Dimensión: dim_last_residence

- Propósito: país de última residencia permanente (similar a dim_country pero separada si requieres metadatos distintos).
- Atributos: last_residence_id (PK), country_name, iso3, notes.

MODELO RELACIONAL

Script SQL implementa un modelo relacional normalizado (3FN)

clean.sql

```sql
-- 1) Dimensiones
CREATE TABLE dim_time (
  time_id INTEGER PRIMARY KEY AUTO_INCREMENT,
  year INTEGER NOT NULL,
  month INTEGER,
  quarter INTEGER,
  period_label VARCHAR(20)
);

CREATE TABLE dim_country (
  country_id INTEGER PRIMARY KEY AUTO_INCREMENT,
  country_name VARCHAR(200) NOT NULL,
  iso2 CHAR(2),
  iso3 CHAR(3),
  region VARCHAR(100),
  subregion VARCHAR(100)
);

CREATE TABLE dim_visa_type (
  visa_type_id INTEGER PRIMARY KEY AUTO_INCREMENT,
  visa_type_name VARCHAR(150) NOT NULL,
  visa_category VARCHAR(100)
);

CREATE TABLE dim_last_residence (
  last_residence_id INTEGER PRIMARY KEY AUTO_INCREMENT,
  country_name VARCHAR(200) NOT NULL,
  iso3 CHAR(3),
  notes TEXT
);
```

```sql
-- 2) Tabla de hechos
CREATE TABLE fact_migration (
  fact_id INTEGER PRIMARY KEY AUTO_INCREMENT,
  time_id INTEGER NOT NULL,
  citizenship_country_id INTEGER NOT NULL,
  visa_type_id INTEGER NOT NULL,
  last_residence_id INTEGER NOT NULL,
  count_people BIGINT NOT NULL,
  source_file VARCHAR(255),
  ingestion_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  row_hash VARCHAR(64),

  FOREIGN KEY (time_id) REFERENCES dim_time(time_id),
  FOREIGN KEY (citizenship_country_id) REFERENCES dim_country(country_id),
  FOREIGN KEY (visa_type_id) REFERENCES dim_visa_type(visa_type_id),
  FOREIGN KEY (last_residence_id) REFERENCES dim_last_residence(last_residence_id)
);

-- Índices para consultas rápidas
CREATE INDEX idx_fact_time ON fact_migration(time_id);
CREATE INDEX idx_fact_citizenship ON fact_migration(citizenship_country_id);
CREATE INDEX idx_fact_visa ON fact_migration(visa_type_id);
```

Las dimensiones de este modelo se normalizaron adhiriéndose a la Tercera Forma Normal (3FN) para garantizar la integridad y eficiencia del almacenamiento.

La clave es la separación de roles, como se ve al tener dim_country y dim_last_residence distintas, lo cual evita repetir atributos descriptivos (como códigos ISO o nombres de países) en millones de filas de hechos.

Igualmente, normalizar dim_visa_type y dim_time asegura que la información repetitiva o de texto largo no sature la tabla de hechos. Esencialmente , cada atributo de la dimensión solo depende de su clave primaria, eliminando redundancias y facilitando el mantenimiento.

CONCLUSIONES

Conclusión

El proyecto de preparación y modelado de datos sobre migración ha dejado claro que la calidad del análisis final depende fundamentalmente de la solidez de la etapa de inspección y limpieza.

La exploración inicial del dataset no es un simple paso; es una etapa crucial para desenmascarar problemas sutiles, como la presencia de tipos de datos incorrectos (por ejemplo, strings donde deberían ir números), la existencia de múltiples variantes para un mismo nombre de país, o la gestión de valores faltantes.

Entender estas inconsistencias desde el principio es lo que asegura la precisión de cualquier estudio posterior.

Uno de los aprendizajes más valiosos es la necesidad de mantener un proceso de limpieza completamente auditable y reproducible.

El uso de scripts detallados para cada transformación garantiza que, si se necesita repetir el proceso en el futuro o si se requiere justificar una decisión de limpieza, el camino esté documentado.

Además, el manejo de datos faltantes siempre requiere un juicio crítico: la decisión entre simplemente eliminar filas o imputar valores debe ponderarse cuidadosamente, ya que la eliminación indiscriminada, especialmente en métricas clave como el conteo de personas (Count), puede introducir un sesgo significativo en los resultados del análisis.

Recomendaciones

De cara a la operatividad futura y la migración al Data Warehouse (DW), se han identificado varias recomendaciones esenciales.

En primer lugar, es imperativo establecer y forzar la estandarización de los atributos clave. Específicamente, se debe adoptar el uso de códigos ISO3 como la regla estándar para nombrar y referenciar países.

Esto no solo facilita la consistencia interna, sino que también simplifica la futura integración con fuentes de datos externas o herramientas de Business Intelligence globales.

En segundo lugar, la trazabilidad del proceso ETL debe ser una prioridad. Es fundamental registrar metadatos detallados: el nombre exacto del archivo fuente, la fecha precisa de ingesta y un resumen de las transformaciones aplicadas.

Esto, junto con la automatización de validaciones básicas (como verificar rangos en el campo Count para evitar valores negativos o irreales), es la columna vertebral de un sistema de datos fiable.

Operativamente, se debe mantener siempre una copia inalterada del dataset original (versión raw), utilizando copias transformadas para el trabajo, garantizando así una fuente de verdad intocable en caso de necesidad de reversión.

En tercer lugar y último, al diseñar el Data Warehouse, se debe incorporar la gestión del cambio en las dimensiones. Esto implica no solo un registro de auditoría, sino también considerar la implementación de técnicas de Slowly Changing Dimensions (SCD).

Esto es vital, sobre todo si los atributos geográficos o las relaciones entre países evolucionan con el tiempo, asegurando que los análisis históricos se mantengan precisos y reflejen la realidad del periodo analizado.

REFERENCIAS

McKee, A. (11 de Sep de 2024). *datacamp*. Obtenido de https://www.datacamp.com/es/tutorial/guide-to-data-cleaning-in-python

Torres, A. (12 de Oct de 2025). *freecodecamp*. Obtenido de https://www.freecodecamp.org/espanol/news/limpieza-de-datos-en-pandas-explicado-con-ejemplos