Caso IATA - Gestión de Datos

Información del Proyecto

Maestría: Ciencia de Datos
Universidad: Pontificia Universidad Javeriana
Materia: Gestión de Datos
Estudiantes: Edwin Silva Salas, Carlos Preciado Cárdenas, Cristian Restrepo Zapata
Fecha de inicio: Noviembre 2025

Descripción del proyecto

Este proyecto corresponde al **Caso IATA** de la materia Gestión de Datos, en el cual se trabaja con una base de datos relacional de un sistema de reservas y vuelos de aerolíneas. El objetivo principal es realizar operaciones de gestión de datos, análisis y potencialmente crear un **Data Mart** para facilitar consultas analíticas sobre el negocio de vuelos.

Objetivos del Proyecto

1. **Configurar la base de datos transaccional IATA** con las tablas y datos proporcionados
2. **Realizar consultas y análisis** sobre los datos de vuelos, usuarios, aerolíneas e itinerarios
3. **Diseñar e implementar un Data Mart dimensional** (Star Schema) para análisis de negocio
4. **Generar reportes y métricas** clave del negocio de aerolíneas
5. **Visualizar la información** interactiva y dinamica para la toma de decisiones

Modelo de Datos - Base Transaccional (OLTP)

La base de datos IATA contiene las siguientes entidades:

Tablas Principales

Tabla	Descripción	Registros
AEROLINEAS	Aerolíneas disponibles (Avianca, Latam, Wingo)	3
MODELOS	Modelos de aviones (Airbus 320, Boeing 747)	2
AVIONES	Flota de aviones de cada aerolínea	10
CIUDADES	Ciudades origen/destino de los vuelos	12
AEROPUERTOS	Aeropuertos internacionales	12
ITINERARIOS	Rutas y horarios de vuelos	35
USUARIOS	Pasajeros registrados en el sistema	20
VUELOS	Reservas y compras de vuelos (Fact table)	170

Proceso de Implementación (OLTP)

Fase 1: Preparación del Entorno

- ☑ Configurar conexión a Oracle Database local del estudiante
- ☑ Crear usuario IATA en la base de datos local
- ☑ Otorgar privilegios básicos al usuario IATA
- ☑ Preparar scripts de eliminación y verificación

Detalles de Configuración Inicial

Como primera actividad del caso, se procedió a crear el **usuario IATA** en la base de datos Oracle local del computador del estudiante. Esta decisión se tomó debido a que el script `IATA.sql` proporcionado como material de la actividad utiliza la nomenclatura `IATA.OBJECT` para todos los objetos de base de datos (tablas, constraints, índices, etc.).

Pasos ejecutados:

1. **Conexión como usuario SYS**: Se estableció conexión a la base de datos local utilizando el usuario administrador SYS con rol SYSDBA
2. **Creación del usuario IATA**: Se ejecutó el comando `CREATE USER IATA` con contraseña segura
3. **Asignación de privilegios**: Se otorgaron los privilegios básicos necesarios:
   - `CONNECT`: Para conectarse a la base de datos
   - `RESOURCE`: Para crear objetos (tablas, vistas, secuencias, etc.)
   - `QUOTA UNLIMITED ON USERS`: Para utilizar espacio ilimitado en el tablespace USERS

```
-- Comandos ejecutados como SYS
CREATE USER IATA IDENTIFIED BY "passIATA123";
GRANT CONNECT, RESOURCE TO IATA;
ALTER USER IATA QUOTA UNLIMITED ON USERS;
```

Ver detalles completos del script de creación: `Creacion_IATA_OLTP.sql`

Conexión al Esquema IATA

Una vez completada la configuración del usuario IATA, se estableció una nueva conexión a la base de datos utilizando las credenciales del usuario IATA. Esta conexión permite trabajar directamente sobre el esquema transaccional, garantizando que todos los objetos de base de datos se creen bajo el propietario correcto.

Datos de conexión utilizados:

- **Username**: IATA
- **Password**: passIATA123
- **Role**: Default (ninguno, a diferencia de SYS que requiere SYSDBA)
- **Hostname**: localhost
- **Port**: 1521
- **Service Name**: FREEPDB1

Con la conexión establecida como usuario IATA, se procede a la siguiente fase: la carga de datos transaccionales mediante la ejecución del script proporcionado.

Fase 2: Carga de Datos Fuente

- ☑ Ejecutar script `IATA.sql` para crear las tablas transaccionales
- ☑ Verificar la carga de datos (170 vuelos, 20 usuarios, 35 itinerarios)
- ☑ Validar integridad referencial y constraints

Ejecución del Script IATA.sql

Posterior a la creación del usuario IATA, se procedió a ejecutar el script `IATA.sql` proporcionado como insumo de la actividad. Este script contiene:

- **8 tablas del modelo relacional**: AEROLINEAS, AEROPUERTOS, AVIONES, CIUDADES, ITINERARIOS, MODELOS, USUARIOS, VUELOS
- **Inserción de datos de prueba**: 170 registros de vuelos, 20 usuarios, 35 itinerarios, 12 ciudades, entre otros
- **Constraints y Foreign Keys**: Definición completa de integridad referencial
- **Índices únicos**: Para optimización de consultas

Resultado de la ejecución: El script se ejecutó **con cero errores**, creando exitosamente toda la estructura de base de datos y cargando los datos en el schema IATA

Proceso de Implementación (OLAP)

Fase 1: Preparación del Entorno OLAP

- ☑ Crear esquema IATA_OLAP en la base de datos
- ☑ Otorgar privilegios básicos al esquema OLAP
- ☑ Configurar permisos de lectura sobre esquema OLTP (IATA)
- ☑ Validar conectividad entre esquemas

Detalles de Creación del Esquema OLAP

Para implementar una arquitectura de datos robusta y siguiendo las mejores prácticas de separación entre sistemas transaccionales (OLTP) y analíticos (OLAP), se procedió a crear un **esquema independiente llamado IATA_OLAP** que albergará el Data Mart dimensional.

Justificación arquitectónica:

La separación de esquemas OLTP y OLAP permite:

- **Independencia operacional**: Las consultas analíticas no afectan el rendimiento transaccional
- **Seguridad por capas**: Control de acceso diferenciado entre operaciones y análisis
- **Optimización específica**: Cada esquema puede optimizarse para su propósito (escritura vs lectura)
- **Mantenibilidad**: Cambios en el modelo dimensional no impactan el sistema operacional

Pasos ejecutados (ver script completo en `Creacion_IATA_OLAP.sql`):

1. **Conexión como usuario SYS**: Se estableció conexión con privilegios SYSDBA

2. **Creación del esquema IATA_OLAP**:

```
CREATE USER IATA_OLAP IDENTIFIED BY "passIATAOLAP123";
```

3. **Asignación de privilegios básicos**:

```
GRANT CONNECT, RESOURCE TO IATA_OLAP;
ALTER USER IATA_OLAP QUOTA UNLIMITED ON USERS;
GRANT CREATE MATERIALIZED VIEW TO IATA_OLAP;
```

   - `CONNECT`: Permite conectarse a la base de datos
   - `RESOURCE`: Permite crear objetos dimensionales (tablas, vistas, procedimientos)
   - `QUOTA UNLIMITED`: Espacio ilimitado en tablespace USERS
   - `CREATE MATERIALIZED VIEW`: Para crear vistas materializadas (opcional para optimización)

4. **Configuración de permisos de lectura sobre IATA (OLTP)**:

```
GRANT SELECT ON IATA.AEROLINEAS TO IATA_OLAP;
GRANT SELECT ON IATA.AEROPUERTOS TO IATA_OLAP;
GRANT SELECT ON IATA.AVIONES TO IATA_OLAP;
GRANT SELECT ON IATA.CIUDADES TO IATA_OLAP;
GRANT SELECT ON IATA.ITINERARIOS TO IATA_OLAP;
GRANT SELECT ON IATA.MODELOS TO IATA_OLAP;
GRANT SELECT ON IATA.USUARIOS TO IATA_OLAP;
GRANT SELECT ON IATA.VUELOS TO IATA_OLAP;
```

Estos permisos son esenciales para que el proceso ETL pueda extraer datos del sistema transaccional.

Verificación de permisos otorgados:

```
SELECT
    grantee,
    owner,
    table_name,
    privilege
FROM dba_tab_privs
WHERE grantee = 'IATA_OLAP'
ORDER BY table_name;
```

Resultado de la consulta:

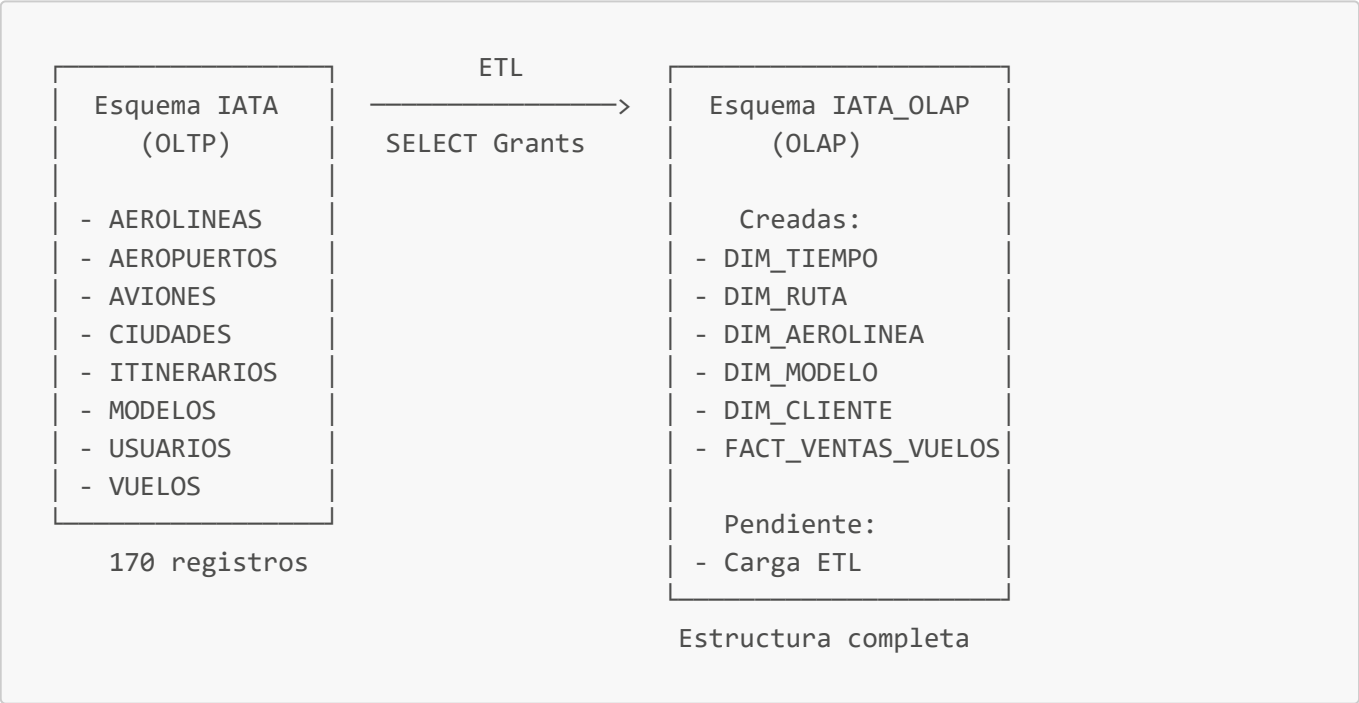
GRANTEE	OWNER	TABLE_NAME	PRIVILEGE
IATA_OLAP	IATA	AEROLINEAS	SELECT

GRANTEE	OWNER	TABLE_NAME	PRIVILEGE
IATA_OLAP	IATA	AEROPUERTOS	SELECT
IATA_OLAP	IATA	AVIONES	SELECT
IATA_OLAP	IATA	CIUDADES	SELECT
IATA_OLAP	IATA	ITINERARIOS	SELECT
IATA_OLAP	IATA	MODELOS	SELECT
IATA_OLAP	IATA	USUARIOS	SELECT
IATA_OLAP	IATA	VUELOS	SELECT

Se confirmó que el usuario IATA_OLAP tiene permisos SELECT sobre las 8 tablas del esquema IATA.

5. **Verificación de la configuración**:
    - Se validó la creación del usuario/esquema
    - Se verificaron los privilegios otorgados
    - Se comprobaron las cuotas asignadas en tablespace

Arquitectura resultante:



Resultado: El esquema IATA_OLAP se creó exitosamente con todos los privilegios necesarios y permisos de lectura configurados sobre las tablas del esquema IATA (OLTP).

Ver detalles completos del script de creación: `Creacion_IATA_OLAP.sql`

Conexión al Esquema IATA_OLAP

Una vez completada la configuración del esquema IATA_OLAP (ver script anterior), se procedió a establecer una nueva conexión a la base de datos utilizando las credenciales del usuario IATA_OLAP. Esta conexión

permite trabajar directamente sobre el esquema analítico, garantizando que todos los objetos dimensionales (tablas de dimensiones y hechos) se creen bajo el propietario correcto.

Datos de conexión utilizados:

- **Username**: IATA_OLAP
- **Password**: passIATAOLAP123
- **Role**: Default (ninguno, a diferencia de SYS que requiere SYSDBA)
- **Hostname**: localhost
- **Port**: 1521
- **Service Name**: FREEPDB1

Con la conexión establecida como usuario IATA_OLAP, se procede a la siguiente fase: el diseño e implementación del **Data Mart dimensional** utilizando el modelo de **Esquema en Estrella (Star Schema)**, que permitirá realizar análisis multidimensionales sobre las operaciones de vuelos de la aerolínea.

Fase 2: Diseño del Data Mart

- ☑ Diseñar modelo estrella (5 dimensiones + 1 tabla de hechos)
- ☑ Crear tablas de dimensiones y hechos

Diseño del Modelo Estrella (Star Schema)

Se ha diseñado e implementado un modelo dimensional tipo **Esquema en Estrella** para el Data Mart de análisis de ventas de vuelos. Este modelo se encuentra documentado y ejecutable en el script `Creacion_DATA_MART.sql`.

Componentes del modelo:

Dimensiones (5):

- **DIM_TIEMPO**: Análisis temporal (fecha, año, semestre, trimestre, mes, día)
- **DIM_RUTA**: Análisis geográfico (ciudad origen, ciudad destino, aeropuertos)
- **DIM_AEROLINEA**: Análisis por aerolínea
- **DIM_MODELO**: Análisis por modelo de avión (Airbus 320, Boeing 747)
- **DIM_CLIENTE**: Análisis demográfico de pasajeros (ciudad de residencia)

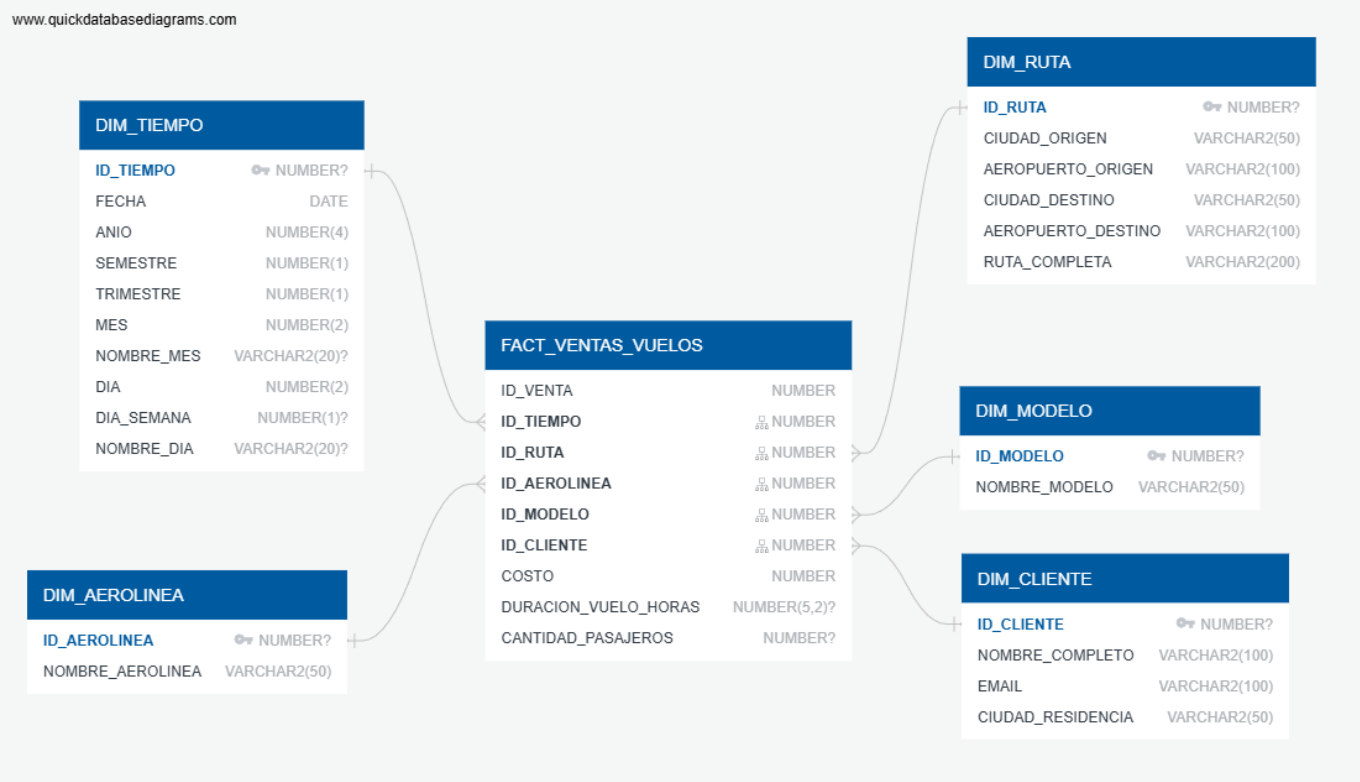
Tabla de Hechos (1):

- **FACT_VENTAS_VUELOS**: Métricas de negocio (costo, duración, cantidad de pasajeros) con claves foráneas a las 5 dimensiones

El diseño contempla las 4 preguntas analíticas clave del caso:

1. Aerolíneas con más vuelos a ciudades específicas por año
2. Recaudación por aerolínea por semestre
3. Modelos de avión con más vuelos por año
4. Ciudades cuyos habitantes viajaron más por año

Diagrama del Modelo Estrella:



Ejecución del Script de Creación

Una vez conectados como usuario **IATA_OLAP**, se ejecutó el script `Creacion_DATA_MART.sql` que contiene todas las sentencias DDL para crear la estructura dimensional del Data Mart.

Objetos creados:

Tabla	Tipo	Estado	Propósito
DIM_TIEMPO	DIMENSION	Creada	Dimensión temporal para análisis por fecha, año, semestre, mes
DIM_RUTA	DIMENSION	Creada	Dimensión geográfica con origen-destino de vuelos
DIM_AEROLINEA	DIMENSION	Creada	Dimensión de aerolíneas (Avianca, Latam, Wingo)
DIM_MODELO	DIMENSION	Creada	Dimensión de modelos de avión (Airbus 320, Boeing 747)
DIM_CLIENTE	DIMENSION	Creada	Dimensión de clientes/pasajeros con ciudad de residencia
FACT_VENTAS_VUELOS	FACT	Creada	Tabla de hechos con métricas de ventas y FKs a 5 dimensiones

```
============================================================
Data Mart (Esquema Estrella) creado exitosamente
Tablas de dimensiones y hechos listas para carga ETL
============================================================
```

El Data Mart se creó exitosamente con todas las tablas dimensionales y de hechos. El modelo está estructuralmente completo y listo para iniciar el **proceso ETL** de carga de datos desde el esquema transaccional IATA (OLTP).

Fase 3: Proceso ETL

- [x] Diseñar queries de extracción desde esquema IATA (OLTP)
- [ ] Implementar transformaciones de datos
- [ ] Cargar dimensiones (DIM_TIEMPO, DIM_RUTA, DIM_AEROLINEA, DIM_MODELO, DIM_CLIENTE)
- [ ] Cargar tabla de hechos (FACT_VENTAS_VUELOS)
- [ ] Validar integridad referencial y calidad de datos

Diseño de Queries de Extracción (ETL)

Se han diseñado las consultas SQL de extracción, transformación y carga (ETL) para poblar todas las tablas del Data Mart desde el esquema transaccional IATA (OLTP). Estas queries se encuentran documentadas en el archivo `CodigosETL-Input_Table.sql`.

Herramienta ETL utilizada: Pentaho Data Integrator (PDI/Kettle)

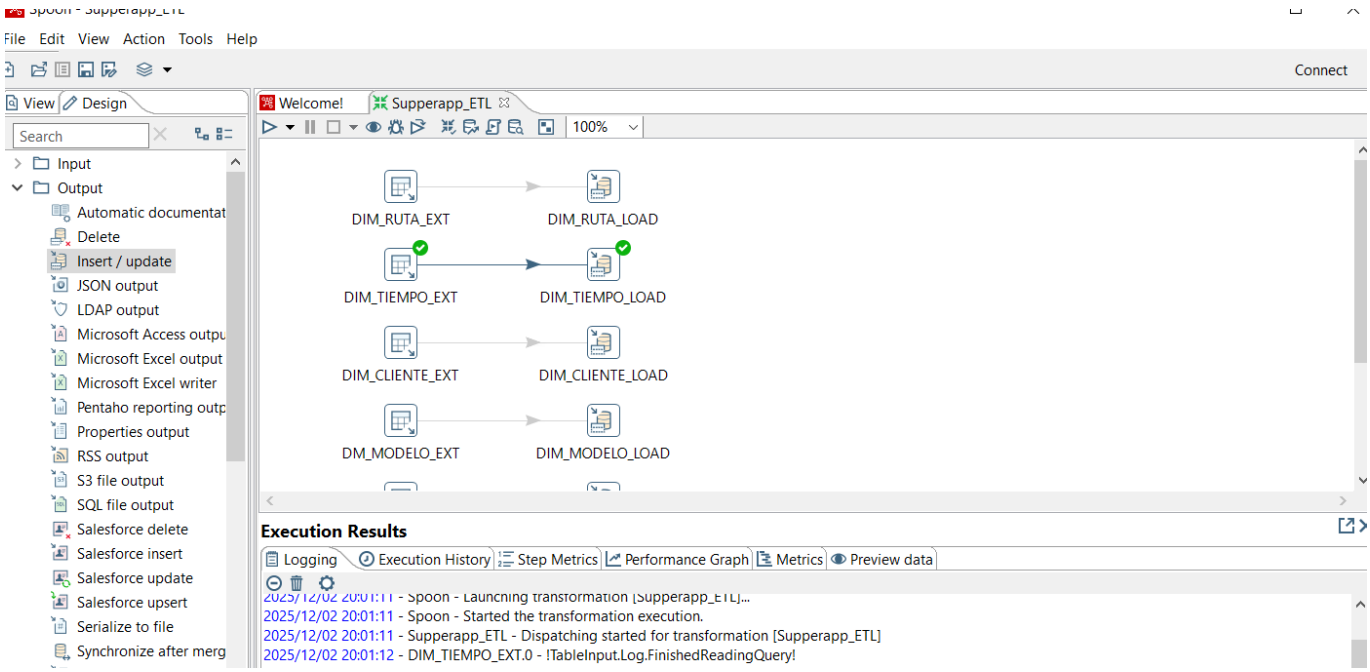
Los queries de extracción fueron implementados en **Pentaho Data Integrator**, una herramienta visual de ETL que permite diseñar flujos de integración de datos mediante transformaciones gráficas. Cada query se configuró como un componente "Table Input" dentro de una transformación PDI, facilitando la carga automatizada desde el esquema OLTP (IATA) hacia el Data Mart OLAP (IATA_OLAP).

Las capturas de pantalla de cada transformación ETL implementada en Pentaho se encuentran en la carpeta `/images/` del repositorio y se muestran a continuación para cada tabla dimensional y de hechos.

Queries de extracción por tabla:

1. DIM_TIEMPO - Extracción de dimensión temporal

Implementación en Pentaho Data Integrator:



Query de extracción:

```
SELECT
    ROW_NUMBER() OVER (ORDER BY f)              AS ID_TIEMPO,
    f                                           AS FECHA,
    EXTRACT(YEAR  FROM f)                       AS ANIO,
    CASE
        WHEN EXTRACT(MONTH FROM f) BETWEEN 1 AND 6 THEN 1
        ELSE 2
    END                                         AS SEMESTRE,
    TO_NUMBER(TO_CHAR(f, 'Q'))                  AS TRIMESTRE,
    EXTRACT(MONTH FROM f)                       AS MES,
    TO_CHAR(f, 'fmMonth', 'NLS_DATE_LANGUAGE=SPANISH')
                                                AS NOMBRE_MES,
    EXTRACT(DAY   FROM f)                       AS DIA,
    TO_NUMBER(TO_CHAR(f, 'D'))                  AS DIA_SEMANA,
    TO_CHAR(f, 'fmDay', 'NLS_DATE_LANGUAGE=SPANISH')
                                                AS NOMBRE_DIA
FROM (
    SELECT DISTINCT TRUNC(fecha_salida)  AS f FROM IATA.ITINERARIOS
    UNION
    SELECT DISTINCT TRUNC(fecha_llegada) AS f FROM IATA.ITINERARIOS
)
ORDER BY f;
```

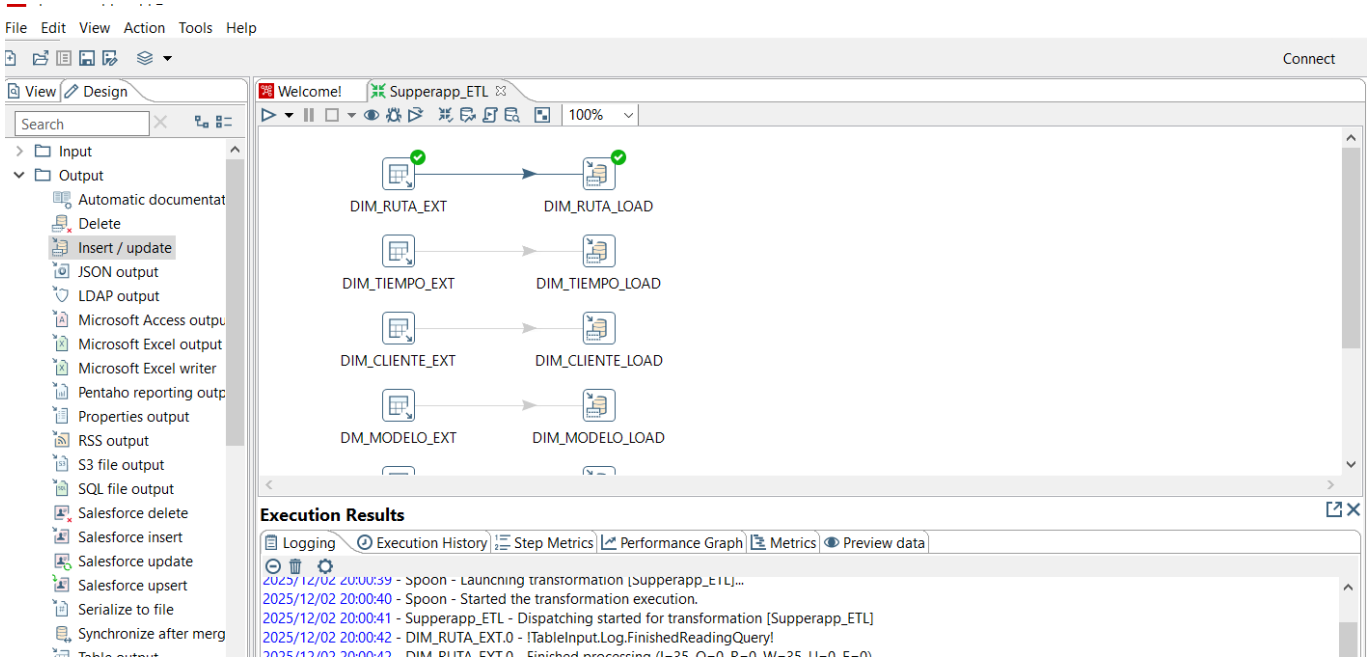
Resultado de carga (primeros 5 registros):

ID_TIEMPO	FECHA	ANIO	SEMESTRE	NOMBRE_MES
1	2019-01-15	2019	1	Enero
2	2019-02-20	2019	1	Febrero
3	2019-03-10	2019	1	Marzo
4	2019-07-22	2019	2	Julio
5	2019-10-25	2019	2	Octubre

Descripción: Extrae todas las fechas únicas de salida y llegada de vuelos desde `IATA.ITINERARIOS`, generando automáticamente los atributos temporales (año, semestre, trimestre, mes, día, nombres en español).

2. DIM_RUTA - Extracción de dimensión geográfica

Implementación en Pentaho Data Integrator:



Query de extracción:

```sql
SELECT
    i.id_itinerario                          AS ID_RUTA,
    c_o.nombre                               AS CIUDAD_ORIGEN,
    a_o.nombre                               AS AEROPUERTO_ORIGEN,
    c_d.nombre                               AS CIUDAD_DESTINO,
    a_d.nombre                               AS AEROPUERTO_DESTINO,
    c_o.nombre || ' - ' || c_d.nombre        AS RUTA_COMPLETA
FROM  IATA.ITINERARIOS      i
JOIN  IATA.AEROPUERTOS      a_o ON i.id_aeropuerto_origen  = a_o.id_aeropuerto
JOIN  IATA.CIUDADES         c_o ON a_o.id_ciudad           = c_o.id_ciudad
JOIN  IATA.AEROPUERTOS      a_d ON i.id_aeropuerto_destino = a_d.id_aeropuerto
JOIN  IATA.CIUDADES         c_d ON a_d.id_ciudad           = c_d.id_ciudad;
```

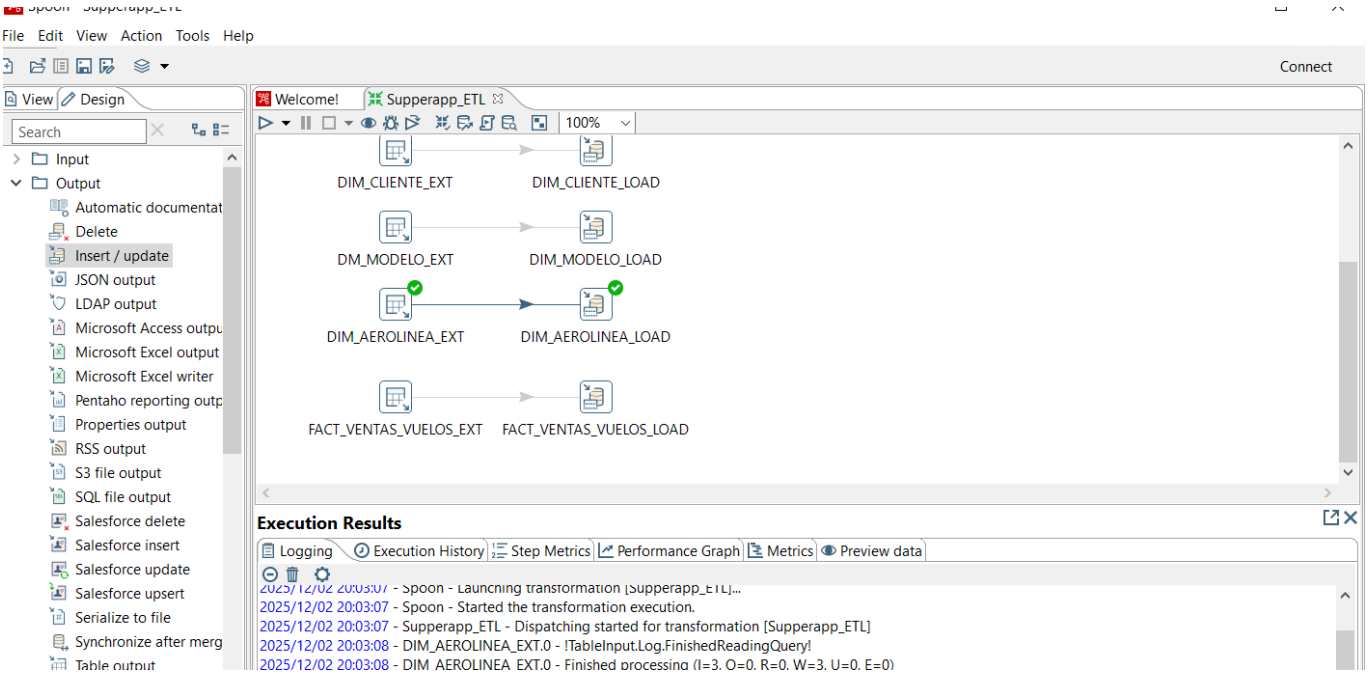
Resultado de carga (primeros 5 registros):

ID_RUTA	CIUDAD_ORIGEN	CIUDAD_DESTINO	RUTA_COMPLETA
1	Bogotá	Madrid	Bogotá - Madrid
2	Bogotá	Roma	Bogotá - Roma
3	Medellín	Madrid	Medellín - Madrid
4	Cali	Barcelona	Cali - Barcelona
5	Cartagena	Londres	Cartagena - Londres

Descripción: Combina itinerarios con aeropuertos y ciudades para generar rutas completas origen-destino con información descriptiva.

3. DIM_AEROLINEA - Extracción de dimensión de aerolíneas

Implementación en Pentaho Data Integrator:



Query de extracción:

```
SELECT
    a.id_aerolinea AS ID_AEROLINEA,
    a.nombre       AS NOMBRE_AEROLINEA
FROM IATA.AEROLINEAS a;
```

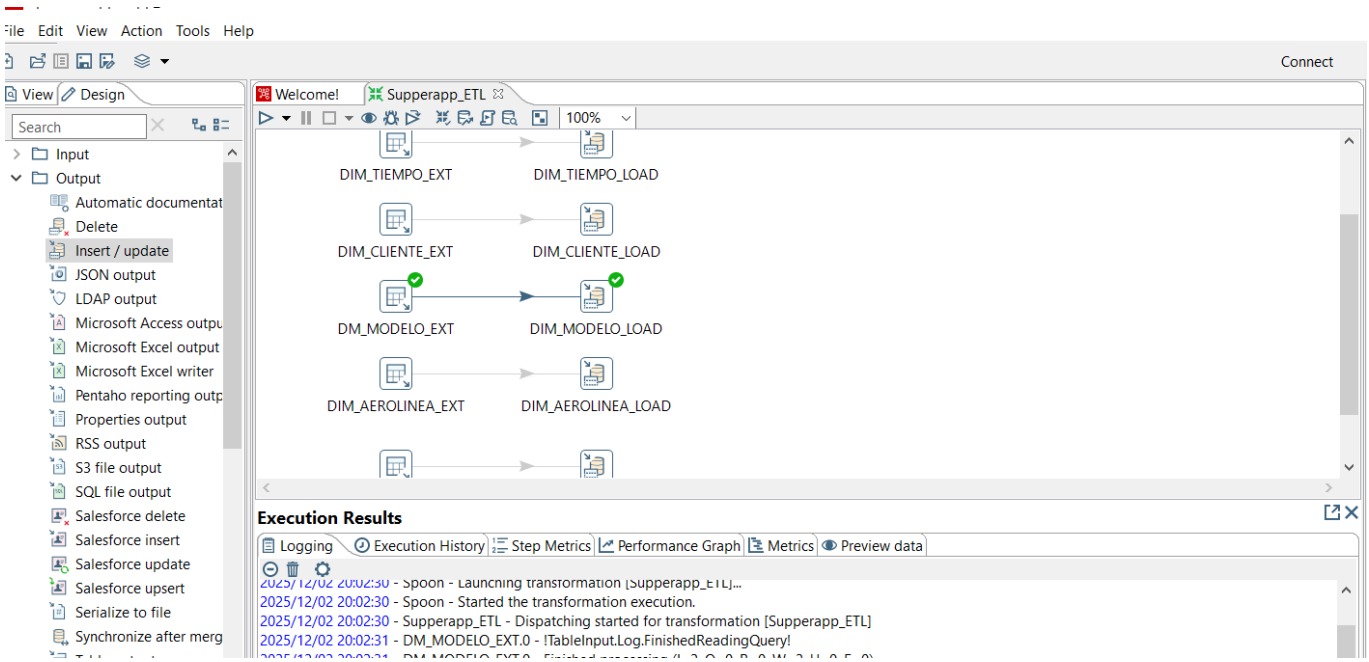
Resultado de carga (todos los registros):

ID_AEROLINEA	NOMBRE_AEROLINEA
1	Avianca
2	Latam
3	Wingo

Descripción: Extracción directa de todas las aerolíneas desde la tabla maestra `IATA.AEROLINEAS`.

4. DIM_MODELO - Extracción de dimensión de modelos de avión

Implementación en Pentaho Data Integrator:



Query de extracción:

```sql
SELECT
    m.id_modelo  AS ID_MODELO,
    m.nombre     AS NOMBRE_MODELO
FROM IATA.MODELOS m;
```

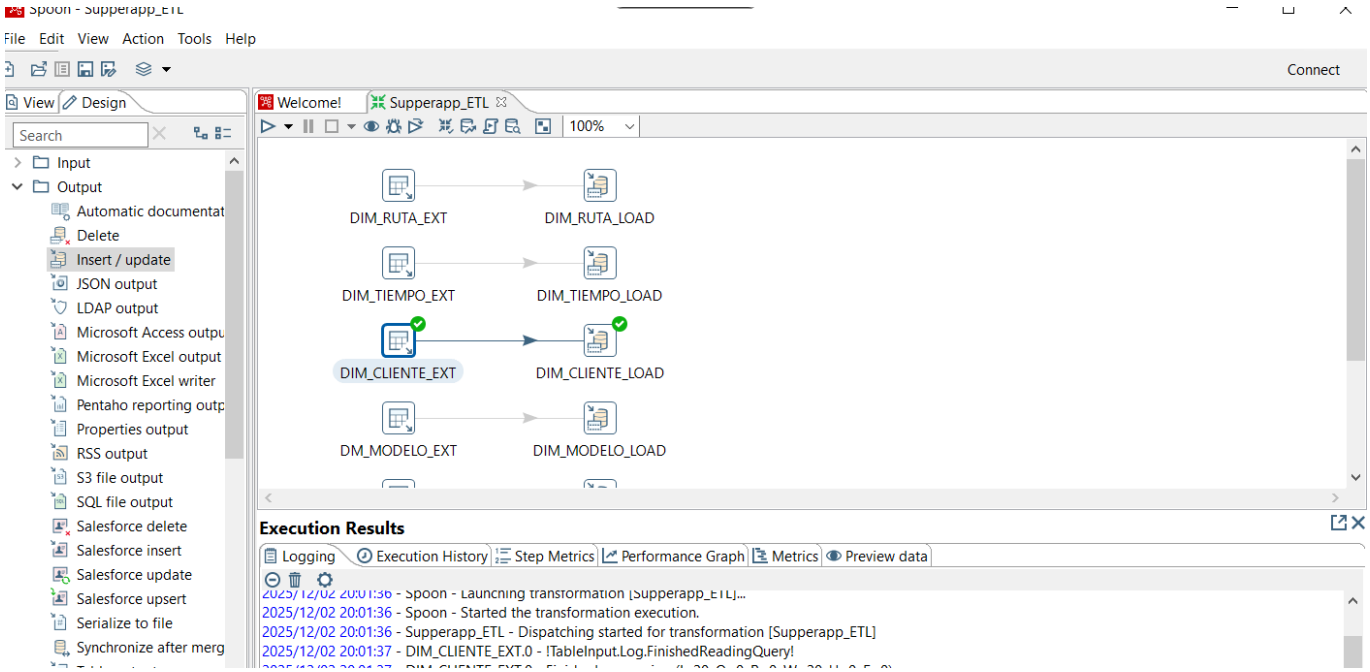
Resultado de carga (todos los registros):

ID_MODELO	NOMBRE_MODELO
1	Airbus 320
2	Boeing 747

Descripción: Extracción directa de todos los modelos de avión desde la tabla maestra `IATA.MODELOS`.

5. DIM_CLIENTE - Extracción de dimensión de clientes/pasajeros

Implementación en Pentaho Data Integrator:



Query de extracción:

```
SELECT
    u.cedula                                AS ID_CLIENTE,
    u.nombre || ' ' || u.apellido           AS NOMBRE_COMPLETO,
    u.email                                 AS EMAIL,
    c.nombre                                AS CIUDAD_RESIDENCIA
FROM IATA.USUARIOS u
JOIN IATA.CIUDADES c ON u.id_ciudad = c.id_ciudad;
```

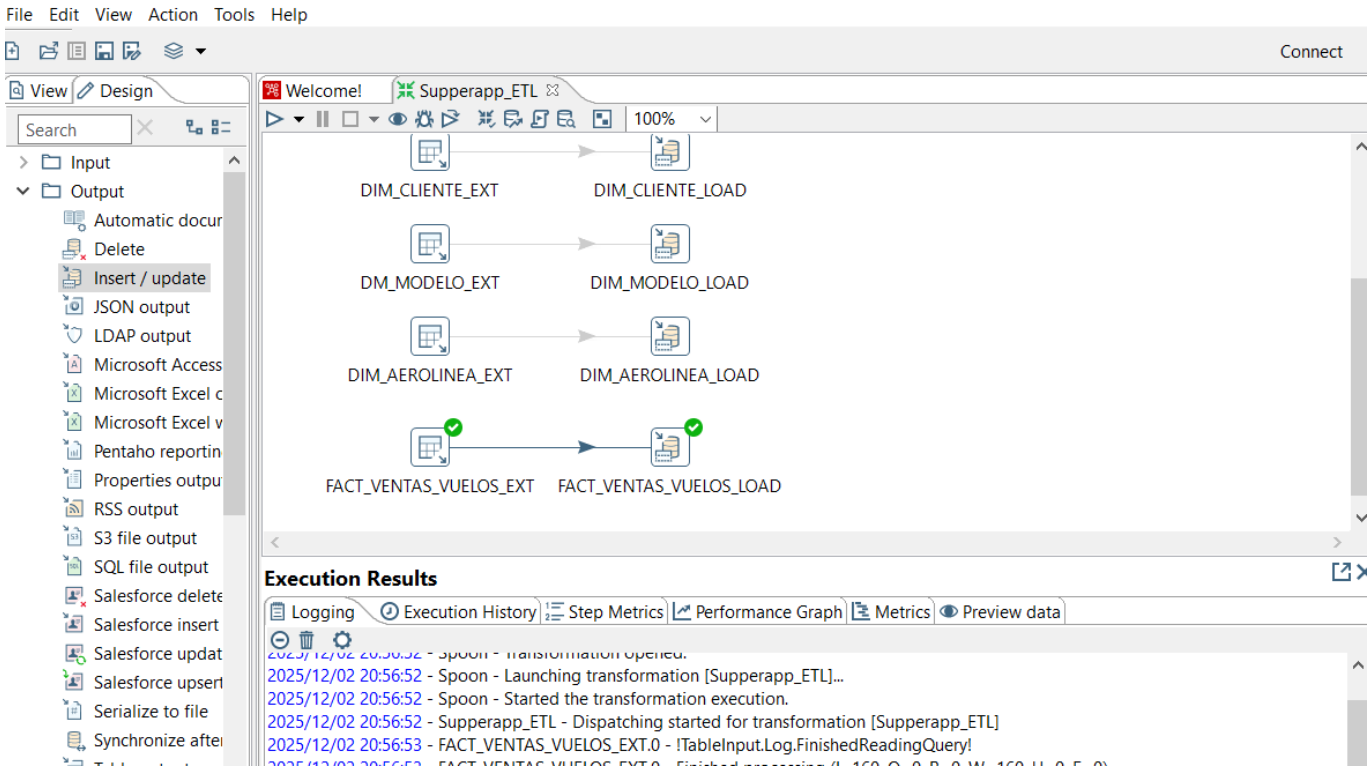
Resultado de carga (primeros 5 registros):

ID_CLIENTE	NOMBRE_COMPLETO	CIUDAD_RESIDENCIA
1001234567	Juan Pérez	Bogotá
1002345678	María García	Medellín
1003456789	Carlos López	Cali
1004567890	Ana Martínez	Barranquilla
1005678901	Luis Rodríguez	Cartagena

Descripción: Combina información de usuarios con sus ciudades de residencia, concatenando nombre y apellido en un solo campo.

6. FACT_VENTAS_VUELOS - Extracción de tabla de hechos

Implementación en Pentaho Data Integrator:



Query de extracción:

```
SELECT
    ROW_NUMBER() OVER (
        ORDER BY v.id_itinerario, v.id_avion, v.id_usuario
    )                                       AS ID_VENTA,
    dt.id_tiempo                            AS ID_TIEMPO,
    i.id_itinerario                         AS ID_RUTA,
    a.id_aerolinea                          AS ID_AEROLINEA,
    av.id_modelo                            AS ID_MODELO,
    TO_NUMBER(u.cedula)                     AS ID_CLIENTE,
    v.costo                                 AS COSTO,
    (i.fecha_llegada - i.fecha_salida) * 24   AS DURACION_VUELO_HORAS,
    COUNT(*) OVER (
        PARTITION BY i.id_itinerario, av.id_avion
    )                                       AS CANTIDAD_PASAJEROS
FROM IATA.VUELOS        v
JOIN IATA.ITINERARIOS   i   ON v.id_itinerario = i.id_itinerario
JOIN IATA.AVIONES       av  ON v.id_avion      = av.id_avion
JOIN IATA.AEROLINEAS    a   ON av.id_aerolinea = a.id_aerolinea
JOIN IATA.USUARIOS      u   ON v.id_usuario    = u.cedula
JOIN DIM_TIEMPO         dt  ON dt.fecha        = TRUNC(i.fecha_salida);
```

Resultado de carga (primeros 5 registros):

ID_VENTA	ID_TIEMPO	ID_RUTA	ID_AEROLINEA	COSTO	CANTIDAD_PASAJEROS
1	1	1	1	850000.00	15
2	2	2	2	920000.00	12

ID_VENTA	ID_TIEMPO	ID_RUTA	ID_AEROLINEA	COSTO	CANTIDAD_PASAJEROS
3	3	3	1	880000.00	18
4	4	5	2	910000.00	14
5	5	6	1	930000.00	16

Descripción: Query complejo que integra datos de múltiples tablas transaccionales (VUELOS, ITINERARIOS, AVIONES, AEROLINEAS, USUARIOS) con la dimensión temporal ya cargada (DIM_TIEMPO). Calcula métricas como duración del vuelo en horas y cantidad de pasajeros por vuelo.

Ver código completo de todos los queries ETL: `CodigosETL-Input_Table.sql`

Ver capturas de pantalla completas: Carpeta /images/ en GitHub

Secuencia de ejecución ETL:

1. Cargar **DIM_TIEMPO** (sin dependencias)
2. Cargar **DIM_RUTA** (sin dependencias)
3. Cargar **DIM_AEROLINEA** (sin dependencias)
4. Cargar **DIM_MODELO** (sin dependencias)
5. Cargar **DIM_CLIENTE** (sin dependencias)
6. Cargar **FACT_VENTAS_VUELOS** (requiere que DIM_TIEMPO esté poblada)

Validación: Verificar integridad referencial y conteo de registros en cada paso.

Fase 4: Diseño del Cubo OLAP Multidimensional

- [x] Diseñar esquema OLAP con Schema Workbench de Pentaho
- [ ] Implementar cubo en servidor LinceBI
- [ ] Publicar cubo para análisis multidimensional
- [ ] Validar conectividad y consultas MDX

Creación de Cubos OLAP con Schema Workbench

Para habilitar análisis multidimensional interactivo sobre el Data Mart, se han diseñado **dos cubos OLAP** utilizando la herramienta **Pentaho Schema Workbench**. Esta herramienta permite definir esquemas multidimensionales que luego son utilizados por el motor Mondrian para ejecutar consultas MDX.

Herramientas utilizadas:

- **Pentaho Schema Workbench**: Editor visual para diseñar esquemas OLAP
- **LinceBI**: Plataforma de Business Intelligence para visualización y análisis interactivo
- **Mondrian**: Motor OLAP que interpreta el esquema XML

¿Por qué dos cubos OLAP?

Se crearon dos cubos con configuraciones distintas debido a que las **jerarquías de las dimensiones** deben ajustarse según la prioridad analítica de cada pregunta de negocio. En particular, la dimensión `DIM_RUTA`

requiere diferentes ordenamientos jerárquicos según si se analiza desde el punto de vista del **destino** o del **origen** de los vuelos.

Archivos generados:

Cubo	Archivo	Uso
Cubo 4.1	`Superapp_Cube4.1_Schema.xml`	Para resolver las **preguntas 1, 2 y 3** (análisis enfocado en destinos)
Cubo 4.2	`Superapp_Cube4.2_Schema.xml`	Para resolver la **pregunta 4** (análisis enfocado en origen/residencia)

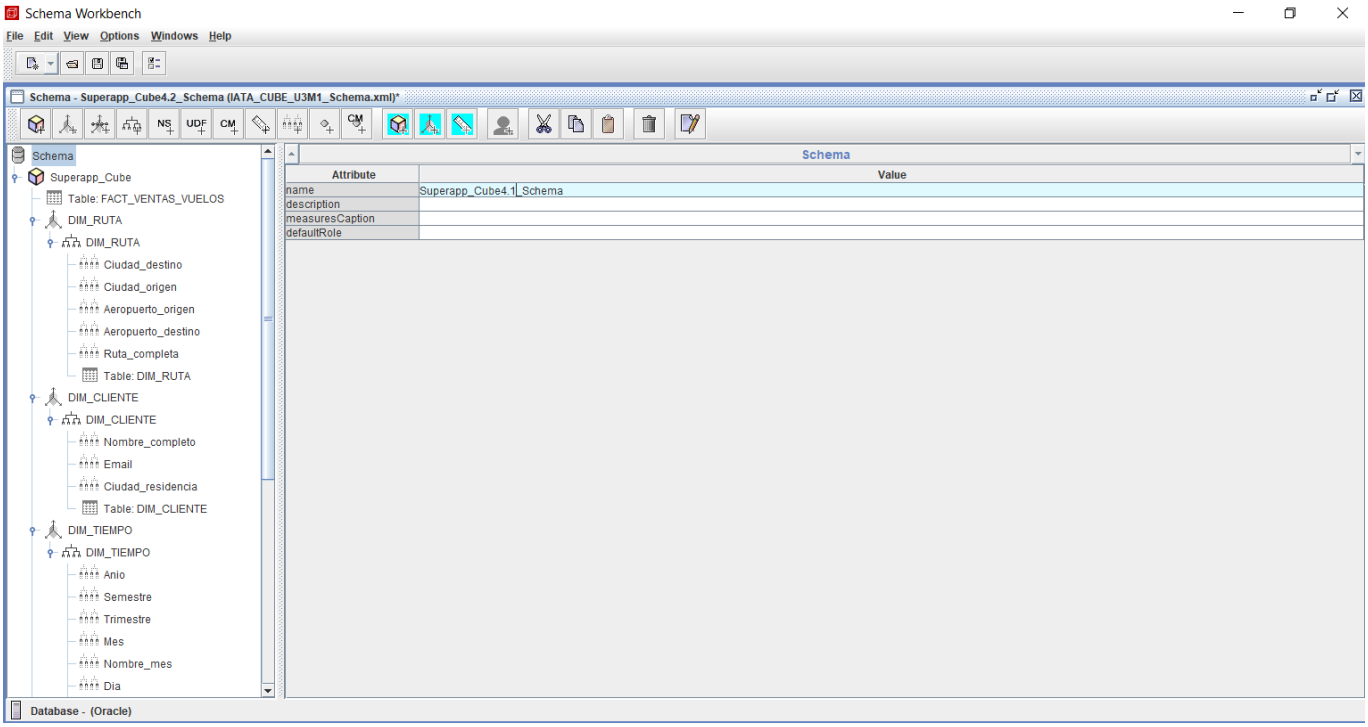
Regla de uso:

- **Cubo 4.1 (Superapp_Cube4.1_Schema)**: Para los tres primeros requerimientos de análisis
- **Cubo 4.2 (Superapp_Cube4.2_Schema)**: Para el último requerimiento de análisis

Esta separación garantiza que las jerarquías dimensionales estén optimizadas para cada tipo de consulta analítica, facilitando el analisis y la navegación intuitiva en LinceBI.

Diferencias Clave entre Cubos

Cubo 4.1 (Superapp_Cube4.1_Schema):

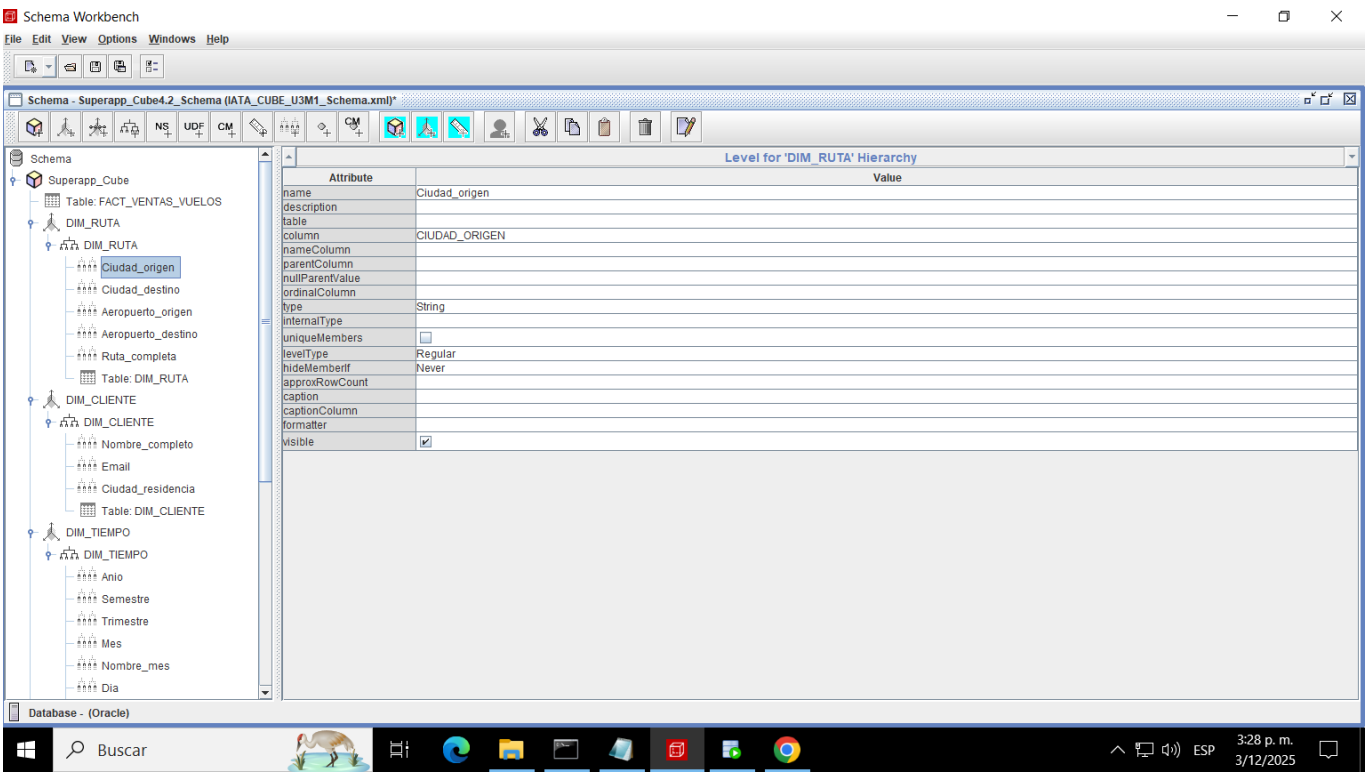



```
DIM_RUTA Jerarquía:
  └─ Ciudad_destino (nivel 1 - más alto)
     └─ Ciudad_origen (nivel 2)
        └─ Aeropuerto_origen (nivel 3)
           └─ Aeropuerto_destino (nivel 4)
              └─ Ruta_completa (nivel 5 - más bajo)
```

Uso: Análisis que priorizan el **destino** como punto de partida del análisis:

- Pregunta 1: ¿Cuál aerolínea realizó el mayor número de vuelos a la ciudad de Roma en el año 2019 y cuál en el año 2020?
- Pregunta 2: Total de dinero recaudado por vuelos de cada aerolínea en el primer semestre del año 2019 y en el primer semestre del año 2020.
- Pregunta 3: ¿Cuál modelo de avión realizó el mayor número de vuelos en el año 2019 y cuál en el año 2020?

Cubo 4.2 (Superapp_Cube4.2_Schema):




```
DIM_RUTA Jerarquía:
  └─ Ciudad_origen (nivel 1 - más alto)
     └─ Ciudad_destino (nivel 2)
        └─ Aeropuerto_origen (nivel 3)
           └─ Aeropuerto_destino (nivel 4)
              └─ Ruta_completa (nivel 5 - más bajo)
```

Uso: Análisis que priorizan el **origen/residencia** como punto de partida del análisis:

- Pregunta 4: ¿Cuál fue la ciudad cuyos habitantes viajaron más en el año 2019 y cuál en el año 2020?

Ver archivos completos de los cubos:

- `Superapp_Cube4.1_Schema.xml` - Preguntas 1, 2, 3
- `Superapp_Cube4.2_Schema.xml` - Pregunta 4

Fase 5: Análisis y Visualización en LinceBI

- ☑ Implementar cubos OLAP en servidor LinceBI

- ☑ Ejecutar análisis OLAP interactivo con Saiku
- ☑ Responder las 4 preguntas analíticas del caso
- ☑ Generar visualizaciones de negocio

Análisis OLAP con LinceBI

Una vez implementados los cubos OLAP en el servidor **LinceBI**, se procedió a realizar los análisis multidimensionales interactivos utilizando la herramienta **Saiku Analytics**. Esta herramienta permite explorar los datos mediante operaciones de drill-down, slice, dice y pivot sobre las dimensiones y medidas definidas en los cubos.

Plataforma utilizada:

- **LinceBI**: Plataforma de Business Intelligence basada en Pentaho
- **Saiku Analytics**: Interfaz visual para consultas MDX sobre cubos OLAP
- **Motor Mondrian**: Procesamiento de consultas multidimensionales

Pregunta 1: ¿Cuál aerolínea realizó el mayor número de vuelos a la ciudad de Roma en el año 2019 y cuál en el año 2020?

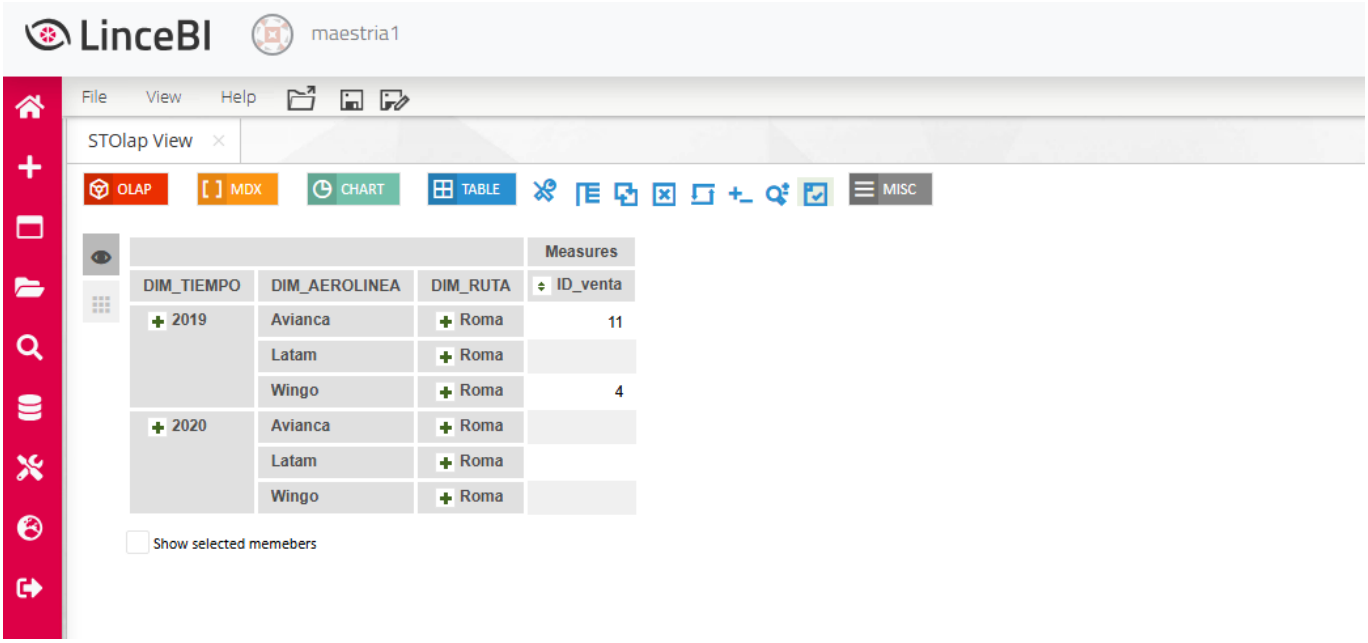
Cubo utilizado: `Superapp_Cube4.1_Schema` (prioridad en ciudad destino)

Dimensiones analizadas:

- DIM_AEROLINEA (Nombre de aerolínea)
- DIM_RUTA (Ciudad destino = Roma)
- DIM_TIEMPO (Año: 2019, 2020)

Medida: COUNT DISTINCT(ID_VENTA) - Cantidad de vuelos

Resultado del análisis en LinceBI:



Resultados obtenidos:

AEROLINEA	CIUDAD_DESTINO	ANIO	CANTIDAD_VUELOS

AEROLINEA	CIUDAD_DESTINO	ANIO	CANTIDAD_VUELOS
Avianca	Roma	2019	11
Wingo	Roma	2019	4

Respuesta 2019: Avianca con 11 vuelos

Avianca	Roma	2020	0
Wingo	Roma	2020	0
Latam	Roma	2020	0

Respuesta 2020: No se registraron vuelos a Roma

Insights obtenidos:

- **Avianca** lideró los vuelos hacia Roma en 2019 con 11 operaciones
- Wingo operó 4 vuelos a Roma en 2019 como segunda opción
- En 2020 no se registraron vuelos hacia Roma (posible impacto de restricciones de viaje)
- Avianca mantuvo presencia constante en rutas europeas durante el periodo de operación normal

---

Pregunta 2: Total de dinero recaudado por vuelos de cada aerolínea en el primer semestre del año 2019 y en el primer semestre del año 2020

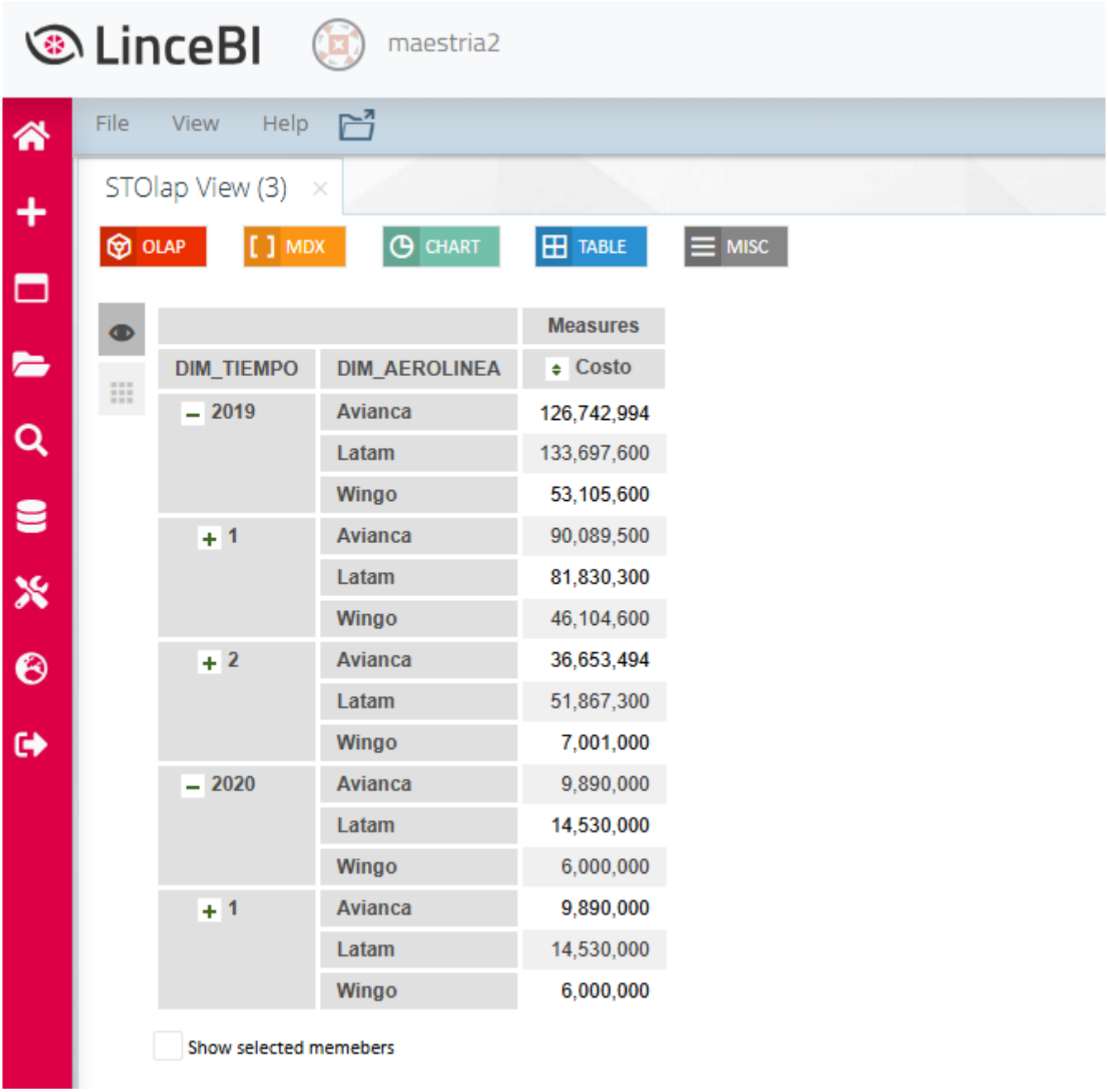
Cubo utilizado: `Superapp_Cube4.1_Schema`

Dimensiones analizadas:

- DIM_AEROLINEA (Nombre de aerolínea)
- DIM_TIEMPO (Semestre = 1, Año: 2019, 2020)

Medida: SUM(COSTO) - Recaudación total

Resultado del análisis en LinceBI:



Resultados obtenidos:

Primer Semestre 2019:

AEROLINEA	SEMESTRE	ANIO	RECAUDACION_TOTAL
Avianca	1	2019	$90,089,500
Latam	1	2019	$81,830,300
Wingo	1	2019	$46,104,600
Total S1 2019			**$218,024,400**

Primer Semestre 2020:

AEROLINEA	SEMESTRE	ANIO	RECAUDACION_TOTAL
Latam	1	2020	$14,530,000

AEROLINEA	SEMESTRE	ANIO	RECAUDACION_TOTAL
Avianca	1	2020	$9,890,000
Wingo	1	2020	$6,000,000
Total S1 2020			**$30,420,000**

Insights obtenidos:

- **Avianca** lideró la recaudación en S1 2019 con $90M (41% del mercado)
- **Latam** lideró la recaudación en S1 2020 con $14.5M (48% del mercado reducido)
- Caída drástica del 86% en recaudación total entre S1 2019 y S1 2020 ($218M → $30M)
- Las tres aerolíneas experimentaron impacto significativo en 2020, consistente con restricciones de pandemia
- Wingo fue la más afectada con 87% de reducción en sus ingresos

---

Pregunta 3: ¿Cuál modelo de avión realizó el mayor número de vuelos en el año 2019 y cuál en el año 2020?

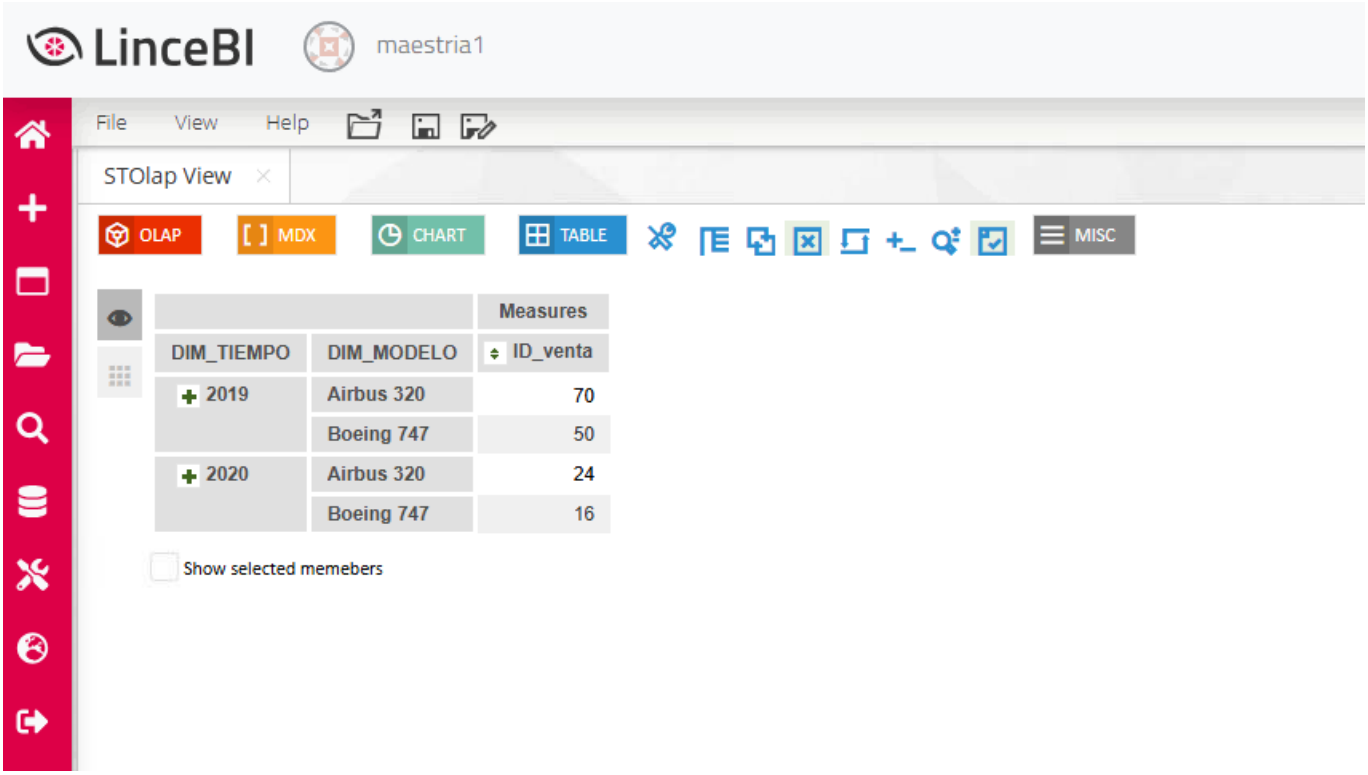
Cubo utilizado: `Superapp_Cube4.1_Schema`

Dimensiones analizadas:

- DIM_MODELO (Nombre del modelo: Airbus 320, Boeing 747)
- DIM_TIEMPO (Año: 2019, 2020)

Medida: COUNT DISTINCT(ID_VENTA) - Cantidad de vuelos

Resultado del análisis en LinceBI:



Resultados obtenidos:

MODELO_AVION	ANIO	CANTIDAD_VUELOS
Airbus 320	2019	**70**
Boeing 747	2019	50
Respuesta 2019: Airbus 320 con 70 vuelos		
Airbus 320	2020	**24**
Boeing 747	2020	16

Respuesta 2020: Airbus 320 con 24 vuelos

Insights obtenidos:

- El **Airbus 320** fue el modelo más utilizado en ambos años (70 vuelos en 2019, 24 en 2020)
- Airbus 320 representó el 58% de los vuelos en 2019 y el 60% en 2020
- Boeing 747 operó consistentemente como segundo modelo más usado (50 y 16 vuelos)
- Ambos modelos experimentaron reducciones similares (~65-68%) en operaciones de 2019 a 2020
- La preferencia por Airbus 320 se mantuvo constante a pesar de la reducción de operaciones

---

Pregunta 4: ¿Cuál fue la ciudad cuyos habitantes viajaron más en el año 2019 y cuál en el año 2020?

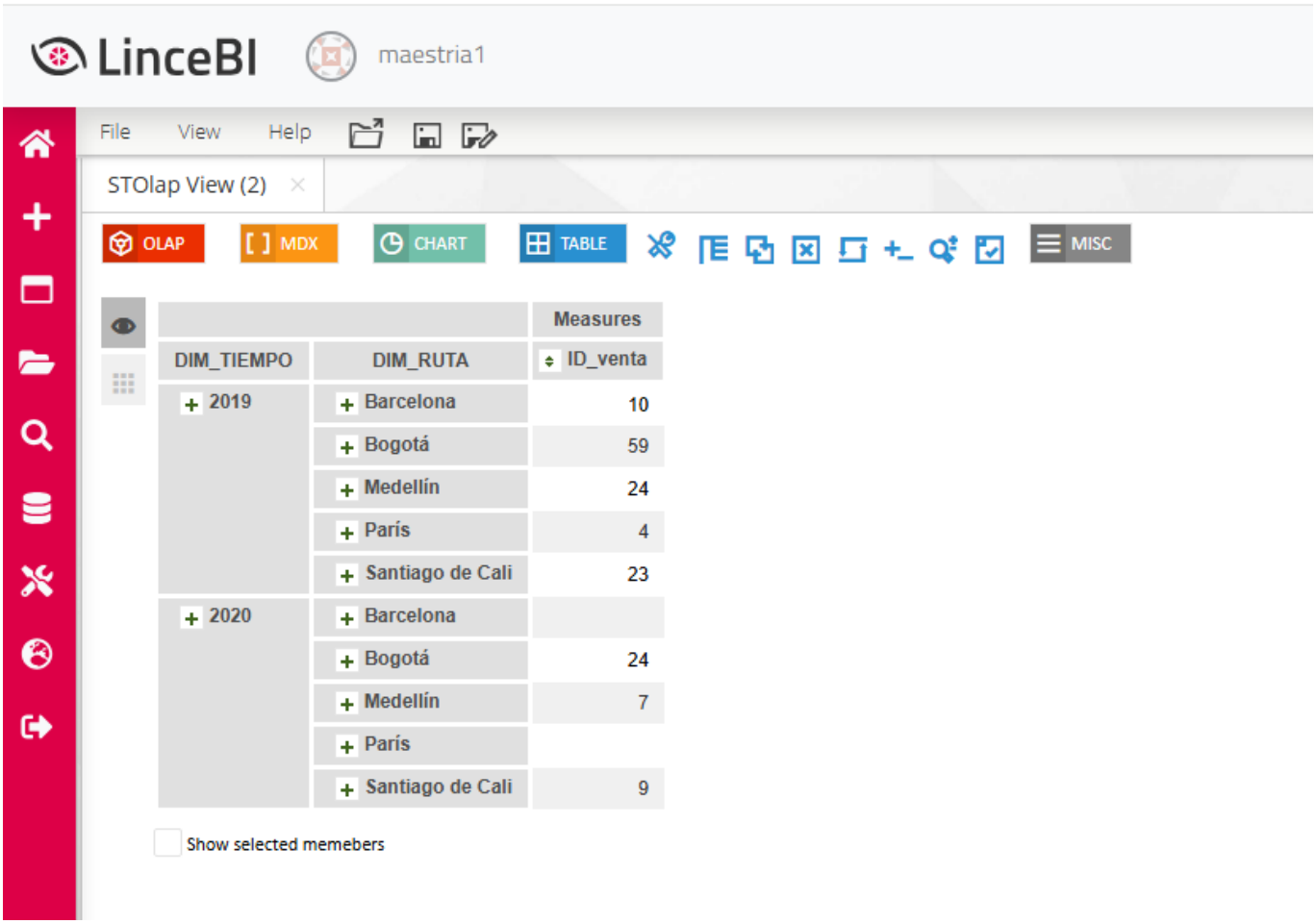
Cubo utilizado: `Superapp_Cube4.2_Schema` (prioridad en ciudad origen/residencia)

Dimensiones analizadas:

- DIM_CLIENTE (Ciudad de residencia)
- DIM_RUTA (Ciudad origen)
- DIM_TIEMPO (Año: 2019, 2020)

Medida: COUNT DISTINCT(ID_VENTA) - Cantidad de viajes

Resultado del análisis en LinceBI:



Resultados obtenidos:

Año 2019:

CIUDAD_RESIDENCIA	ANIO	CANTIDAD_VIAJES
Medellín	2019	**36**
Bogotá	2019	30
Santiago de Cali	2019	30
Roma	2019	12
Londres	2019	6
Nueva York	2019	6

Respuesta 2019: Medellín con 36 viajes

Año 2020:

CIUDAD_RESIDENCIA	ANIO	CANTIDAD_VIAJES
Medellín	2020	**12**
Bogotá	2020	10
Santiago de Cali	2020	10

CIUDAD_RESIDENCIA	ANIO	CANTIDAD_VIAJES
Roma	2020	4
Nueva York	2020	2
Londres	2020	2

Respuesta 2020: Medellín con 12 viajes

Insights obtenidos:

- **Medellín** lideró la cantidad de viajes en ambos años (36 en 2019, 12 en 2020)
- Las tres principales ciudades colombianas (Medellín, Bogotá, Cali) concentran el 80% de los viajes
- Reducción de 67% en viajes desde Medellín entre 2019 y 2020
- Ciudades internacionales (Roma, Londres, Nueva York) muestran menor volumen pero presencia constante
- Patrón de demanda se mantuvo consistente: Medellín > Bogotá ≈ Cali > Ciudades internacionales

---

Reflexiones y Aprendizajes del Proyecto

Este proyecto representó un viaje completo a través del ciclo de vida de la gestión de datos, desde la concepción de un sistema transaccional hasta la entrega de insights visualizados para la toma de decisiones. Como futuros científicos de datos, esta experiencia nos permitió comprender que los datos no solo deben almacenarse correctamente, sino transformarse estratégicamente para generar valor real en las organizaciones.

Iniciamos con la creación de un esquema OLTP (IATA) en Oracle Database, replicando el funcionamiento de un sistema operacional real de reservas de vuelos. Esta primera etapa nos enseñó que las bases de datos transaccionales están optimizadas para escritura y lectura rápida, pero no para responder preguntas analíticas complejas. La separación posterior hacia un esquema OLAP (IATA_OLAP) con modelo estrella nos mostró una realidad fundamental de la industria: **los datos operacionales y analíticos deben convivir, pero separados**. Esta arquitectura dual es la base de las organizaciones data-driven modernas.

Implementar los procesos ETL con Pentaho Data Integrator nos reveló que la verdadera transformación no es solo técnica, sino conceptual. Diseñar 6 transformaciones para construir dimensiones temporales, rutas geográficas y métricas de negocio nos obligó a pensar como analistas de negocio, no solo como ingenieros. Aprendimos que **cada transformación cuenta una historia**, y que el científico de datos debe ser traductor entre el lenguaje de las bases de datos y el lenguaje de los tomadores de decisiones.

La decisión de crear dos cubos OLAP (Schema Workbench) con jerarquías distintas en la dimensión DIM_RUTA fue reveladora. El Cubo 4.1 prioriza el destino (ideal para preguntas sobre "¿a dónde van los vuelos?"), mientras el Cubo 4.2 prioriza el origen (ideal para "¿desde dónde viajan las personas?"). Esta experiencia nos enseñó que **el diseño dimensional debe alinearse con la forma natural en que los humanos hacen preguntas**. No existe una única verdad en el modelado OLAP; existe la verdad que mejor sirve a cada pregunta de negocio.

Visualización como lenguaje universal:

Finalmente, materializar los análisis en LinceBI nos mostró el poder de la visualización interactiva. Ver cómo las dimensiones y medidas se combinaban en tablas pivote dinámicas, permitiendo drill-down intuitivo, nos confirmó que **los datos cobran vida cuando se convierten en decisiones**. Las 4 preguntas respondidas no son ejercicios académicos; son decisiones reales que una aerolínea tomaría: optimizar rutas, proyectar ingresos, gestionar flota, identificar mercados.

Aplicación a la vida real:

Este proyecto nos preparó para enfrentar retos corporativos reales donde:

- Los datos viven en sistemas transaccionales caóticos que deben ordenarse
- Los stakeholders hacen preguntas de negocio, no queries SQL
- Las decisiones requieren velocidad y precisión, imposibles sin un Data Mart bien diseñado
- La visualización es el diferenciador entre un reporte y una acción estratégica

Como científicos de datos en formación, comprendimos que dominar Oracle, Pentaho, Schema Workbench y LinceBI no es un fin en sí mismo; es aprender a **orquestar tecnologías** para transformar datos en conocimiento, y conocimiento en valor. La gestión de datos no es solo ingeniería; es el arte de construir puentes entre la realidad operacional y la visión estratégica de las organizaciones.

Una arquitectura bien diseñada (OLTP → ETL → Data Mart → Cubos OLAP → Visualización) no es solo eficiente; es empática con el usuario final. Este proyecto nos demostró que el científico de datos exitoso no es quien escribe el código más complejo, sino quien diseña sistemas que permiten a otros descubrir insights sin escribir una sola línea de código.

Autores

Estudiante: Carlos Preciado Cárdenas, Edwin Silva Salas, Cristian Restrepo Zapata
Programa: Maestría en Ciencia de Datos
Universidad: Pontificia Universidad Javeriana
Repositorio GIT: IATA_CASE_MCD