

Introducción

El proyecto consiste en la creación de un modelo dimensional en estrella a partir de una base de datos de origen, en este caso es la bdd de Chinook. El objetivo es procesar la información de ventas, clientes y productos a través de un proceso ETL y almacenando en un DW.

Diseño del modelo dimensional

El modelo dimensional tiene una tabla de hechos y 3 tablas de dimensiones diseñadas para responder lo solicitado.

1. Tabla de hechos
 - a. FactSales: Almacena las transacciones de ventas. Tiene las llaves foráneas de las dimensiones que componen el diseño dimensional
2. Tabla de dimensiones.
 - a. DimCostumer: Tiene atributos necesarios para conocer los clientes como fullName, Country y City. En el caso de fullName es una transformación con devired column.
 - b. DimProducto: Almacena información necesria sobre los productos en este caso pistas musicales. Es una dimensión que usa diferentes tablas de la bd de origen.
 - c. DimDate: Esta tabla se genera para permitir un análisis temporal de las ventas.

Proceso ETL

El proceso ETL se implementó en SSIS, siguiendo un flujo de trabajo.

1. Extracción de datos:

- a. DimCustomer: se extrajeron los datos de la tabla customer desde Chinook.
- b. DimProduct: Se utilizó una consulta JOIN con OLE DB Sources para combinar los datos de las tablas Track, Album, Artist, Genre y MediaType en solo 1 fuente.
- c. FactSales: La extracción se realizó a partir de las tablas invoice e invoiceLine

2. Transformaciones:

- a. Cálculos y Limpieza: Se usó la transformación de derived column para crear FullName y limpiar y convertir el tipado de datos.

- b. Obtención de claves: Se utilizó la transformación con Lookup en los flujos de datos de DimCustomer, DimProduct y FactSales. Ayudando a los datos de la tabla de hechos.

3. Carga de datos:

- a. Los datos transformados se almacenan de datos llamado DW_Chinook.

Consultas SQL

Total de ventas por cliente:

```
SELECT
    c.FullName,
    SUM(fs.TotalAmount) AS TotalVentas
FROM
    FactSales AS fs
JOIN
    DimCustomer AS c ON fs.CustomerKey = c.CustomerKey
GROUP BY
    c.FullName
ORDER BY
    TotalVentas DESC;
```

Total de ventas por genero

```
SELECT
    p.Genre,
    SUM(fs.TotalAmount) AS TotalVentas
FROM
    FactSales AS fs
JOIN
    DimProduct AS p ON fs.ProductKey = p.ProductKey
GROUP BY
    p.Genre
ORDER BY
    TotalVentas DESC;
```

Total de ventas por artista

```
SELECT
    p.Artist,
    SUM(fs.TotalAmount) AS TotalVentas
FROM
    FactSales AS fs
JOIN
    DimProduct AS p ON fs.ProductKey = p.ProductKey
GROUP BY
    p.Artist
ORDER BY
    TotalVentas DESC;
```

Total de ventas por país.

```
SELECT
    c.Country,
    SUM(fs.TotalAmount) AS TotalVentas
FROM
    FactSales AS fs
JOIN
    DimCustomer AS c ON fs.CustomerKey = c.CustomerKey
GROUP BY
    c.Country
ORDER BY
    TotalVentas DESC;
```

Documentación y Entrega

El proyecto se gestionó utilizando GIT y Github. Todo esto para el control de versiones y seguimiento de los cambios en el código. El repositorio cuenta con los scripts de creación de la base de datos , documentación detallada y la solución de SSIS.