

1. Análisis y modelado de datos

Para tener una visión multidimensional del proceso de pedidos (sistema de ventas), se optó por usar un modelo de datos en estrella ya que este tipo de modelamiento se hace más simple y fácil de usar para consultas analíticas.

➤ **Modelo de datos relacional:** El modelo en estrella comprende las siguientes tablas:

Tablas de Hechos:

- FactPedidos: Contiene datos de los pedidos y sus detalles.

Tablas de Dimensiones:

- Dim_Categoria
- Dim_Cliente
- Dim_Despachador
- Dim_Empleado
- Dim_Geografia
- Dim_Producto
- Dim_Proveedor
- Dim_Tiempo
- Fact_Pedidos

➤ **Script para medir la rotación del producto más vendido por ciudad de los pedidos cuyo proveedor sea de Melbourne:**

```
WITH ProductosMasVendidosPorCiudad AS (  
  SELECT  
    dp.skProducto,  
    dp.Nombre AS NombreProducto,  
    dg.NombreCiudad,  
    SUM(fp.Cantidad) AS TotalVendido  
  FROM  
    FACT_PEDIDOS fp  
    INNER JOIN DIM_PRODUCTO dp ON fp.fkProducto = dp.skProducto  
    INNER JOIN DIM_PROVEEDOR dpr ON fp.fkProveedor = dpr.skProveedor  
    INNER JOIN DIM_GEOGRAFIA dg ON fp.fkGeografia = dg.skGeografia  
  WHERE  
    dg.NombreCiudad = 'Melbourne'  
  GROUP BY  
    dp.skProducto,  
    dp.Nombre,  
    dg.NombreCiudad  
)  
RankProductosPorCiudad AS (  
  SELECT  
    *,  
    ROW_NUMBER() OVER (PARTITION BY dp.skProducto, dg.NombreCiudad  
      ORDER BY TotalVendido DESC)
```

```

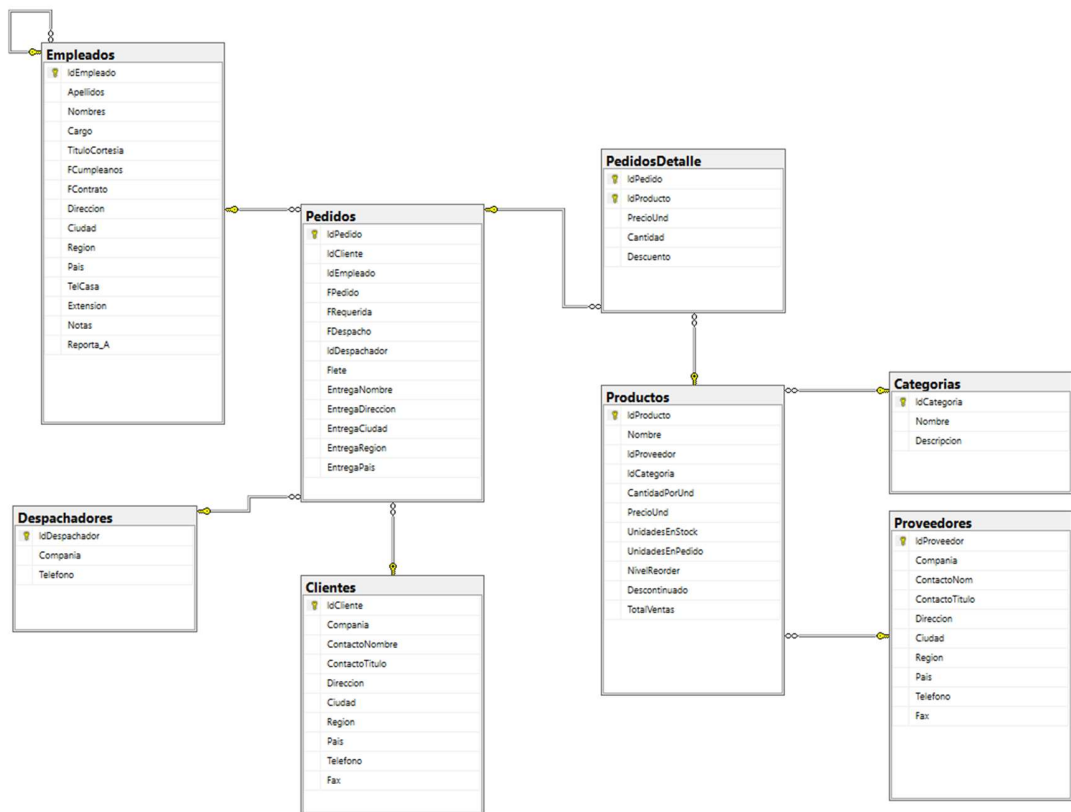
        ROW_NUMBER() OVER(PARTITION BY NombreCiudad ORDER BY TotalVendido DESC) AS
RankPorCiudad
FROM
    ProductosMasVendidosPorCiudad
)
SELECT
    NombreCiudad,
    NombreProducto AS ProductoMasVendido,
    TotalVendido AS CantidadVendida
FROM
    RankProductosPorCiudad
WHERE
    RankPorCiudad = 1;

```

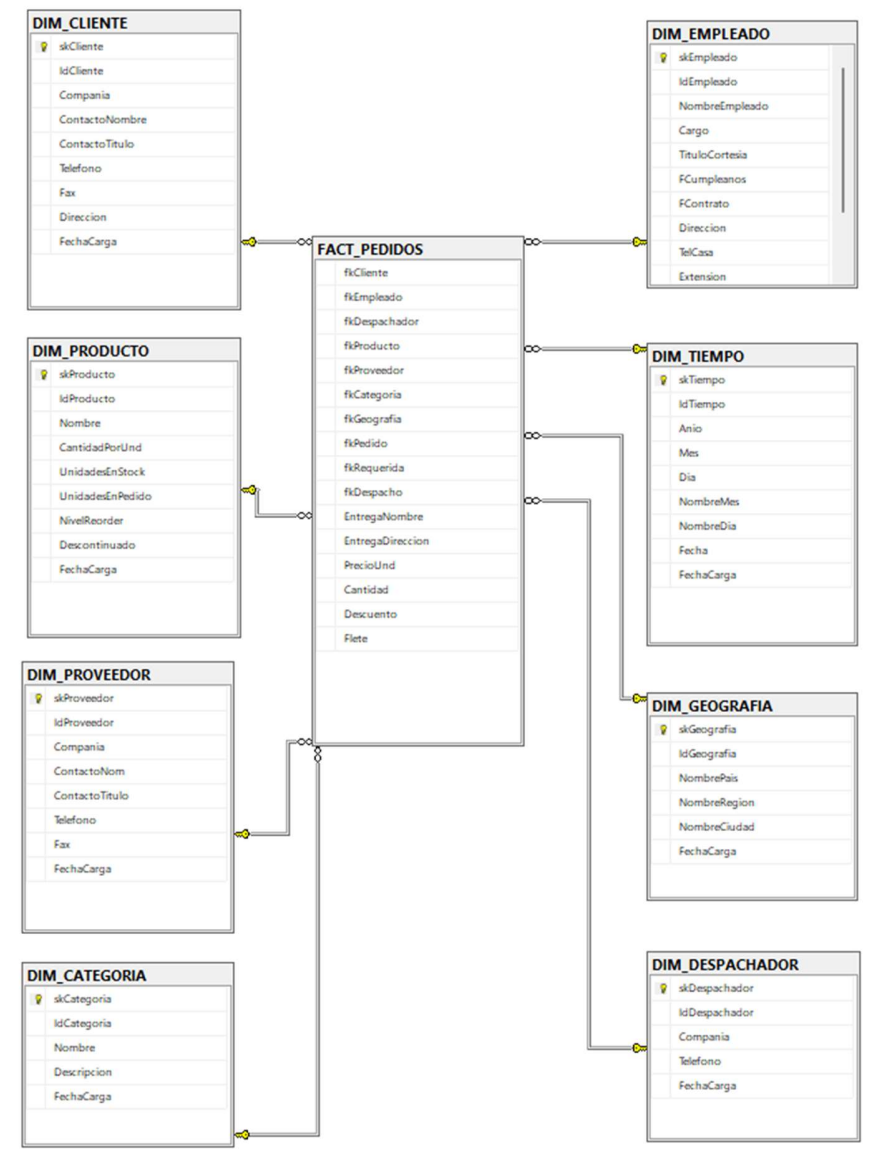
➤ **Visualización del esquema diseñado:**

A continuación, se presenta una visualización del esquema de datos planteado para el proceso de pedidos (sistema de ventas):

db_SistVenta



dwh SistVenta



➤ Modelo más adecuado: Estrella vs Copo de Nieve

Se optó por implementar un modelo en estrella la mayoría de los casos de uso brinda mayor simplicidad y rendimiento en consultas; además, es fácil de entender y proporciona un buen equilibrio entre simplicidad y poder analítico.

A diferencia del modelo en copo de nieve por tener una normalización mayor, se vuelve más complejo en el diseño y las consultas.

➤ Vulnerabilidades en la arquitectura de datos propuesta

- Integridad de Datos: Se debe garantizar que todas las relaciones y claves foráneas están bien definidas para mantener la integridad referencial.

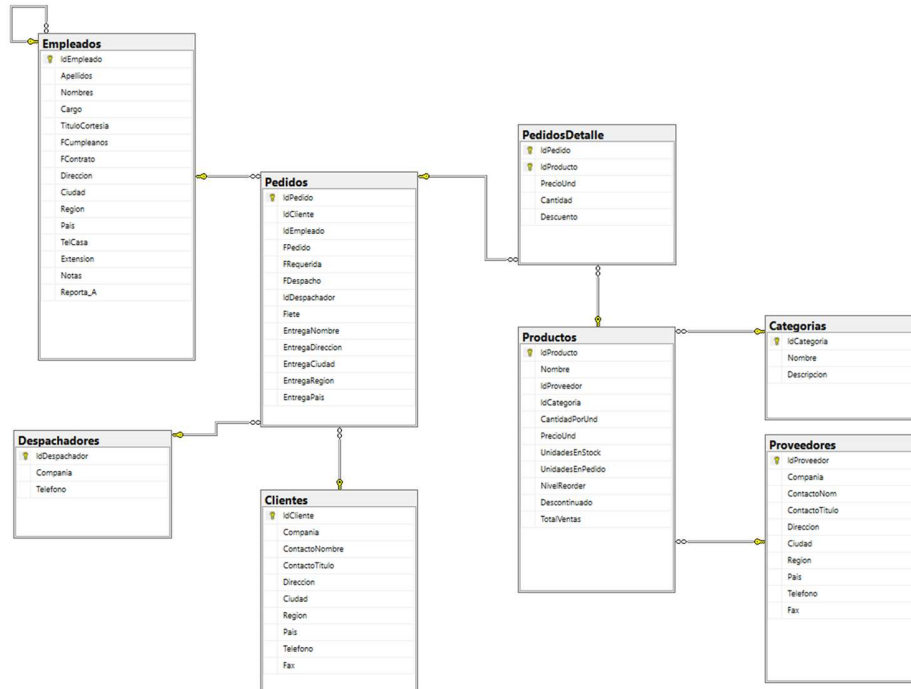
- Rendimiento: La complejidad de las consultas puede afectar el rendimiento; por lo cual es importante considerar el uso de índices adecuados.
- Escalabilidad: A nivel de escalabilidad esta arquitectura debe ser escalable para manejar un aumento en la cantidad de datos.

➤ **Medidas de Seguridad:** Las medidas de seguridad que se pueden implementar serían las siguientes:

- Protección Contra Inyecciones SQL:
 - Uso consultas parametrizadas.
 - Emplear procedimientos almacenados.
 - Validar y verificar todas las entradas del usuario a las bases de datos.
- Políticas de Acceso:
 - Definir roles y permisos adecuados.
 - Utilizar el principio de menor privilegio para otorgar acceso.
 - Implementar autenticación y autorización sólidas.
- Auditorías:
 - Habilitar auditoría en SQL Server para registrar accesos y cambios.
 - Revisar los registros de auditoría regularmente para detectar accesos no autorizados.
- Cumplimiento:
 - Asegúrate de cumplir con normativas de seguridad y privacidad
 - Realiza revisiones y pruebas de seguridad periódicas.

2. Diseño de modelo relacional

Se plantea el siguiente modelo relacional para dar solución al planteamiento propuesto:



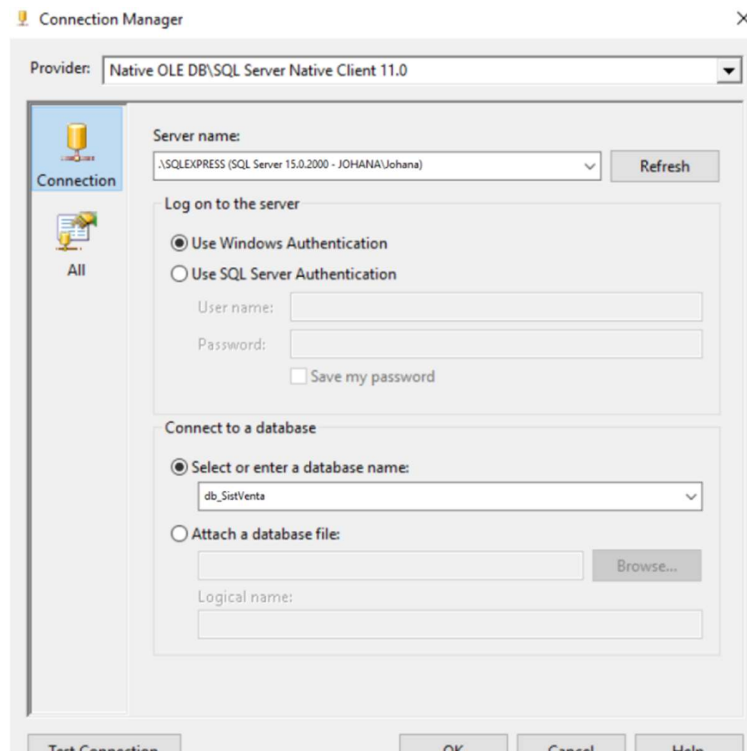
➤ Migración a un modelo de datos no relacional:

Para migrar un modelo de datos relacional a un modelo no relacional se requiere una reestructuración significativa de cómo se almacenarán y gestionarán los datos; donde generalmente se realizan una serie de pasos hasta llegar a la construcción de un modelo acorde a las necesidades, el cual puede comprender lo siguiente:

- Identificar el tipo de base de datos no relacional
- Realizar el mapeo de tablas a colecciones
- Realizar la denormalización
- Proceso de migración de datos
 - Extracción de datos
 - Transformación de datos
 - Carga en la nueva base de datos
- Optimización
- Validación y Testing

3. Diseño del datasource

Datasource de conexión mediante SQL Server Data Tools



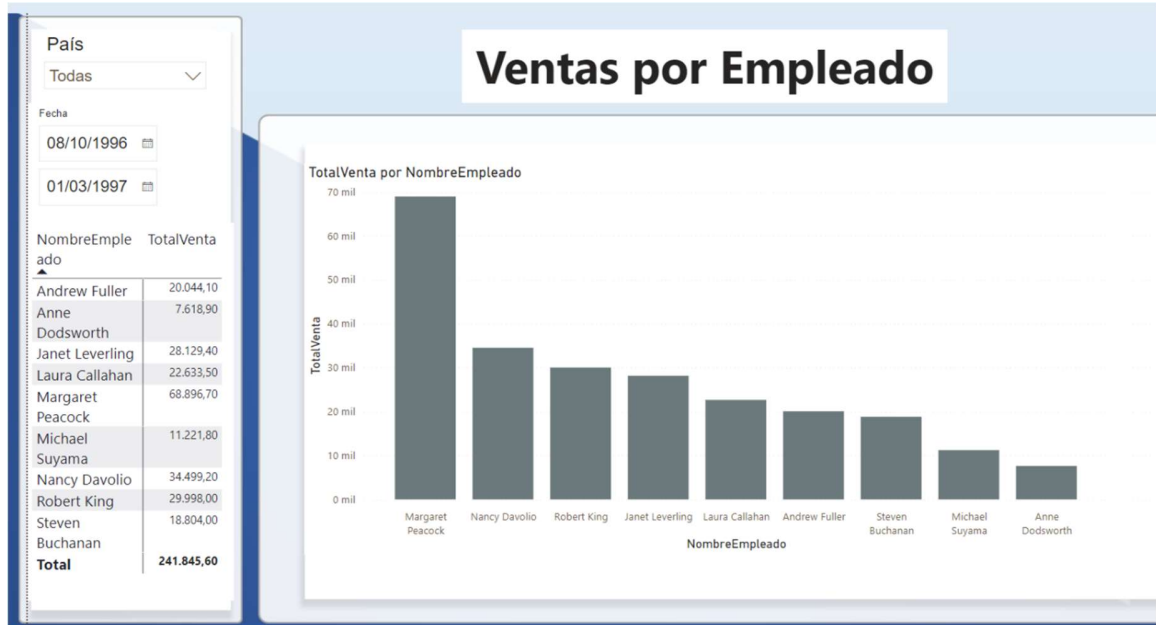
4. ETL y transformaciones

Con la base de datos propuesta se implementaron varios procesos ETL y transformaciones para realizar una mejor integración, limpieza y mejoramiento en la calidad de los datos. Estos procesos comprenden lo siguiente:

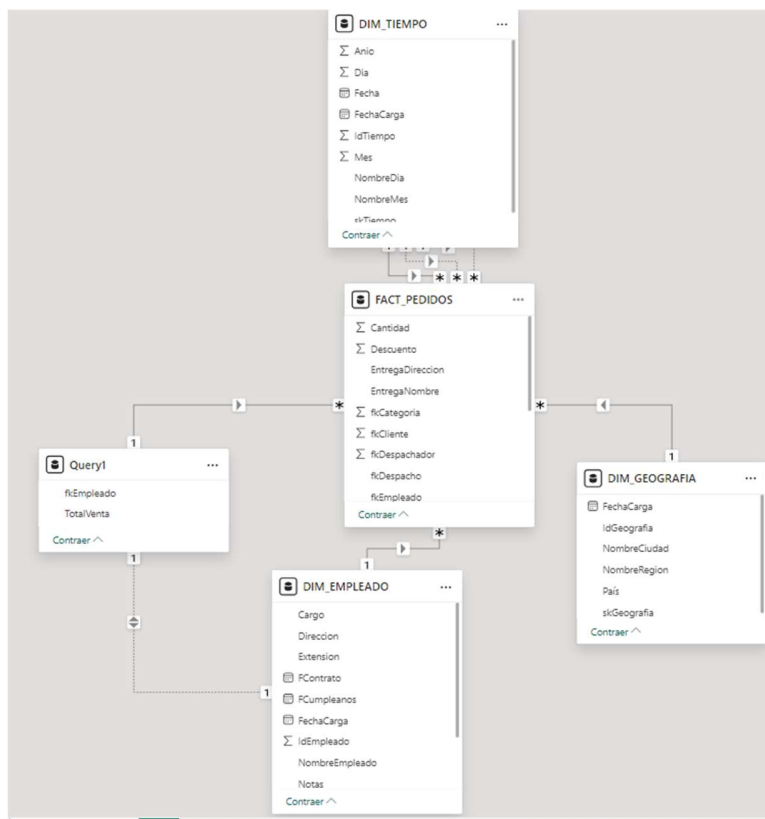
- a. Extracción (Extract)
 - Se realizó conexión a la base de datos db_SistVenta en SQL Server utilizando un Connection Manager en SSIS.
 - Se realizó la extracción de datos de las tablas principales (Clientes, Pedidos, PedidosDetalle, Productos, Proveedores, Categorías, Empleados, Despachadores)
- b. Transformaciones (Transform)
 - Se realizó limpieza de datos para eliminar registros duplicados, corrección y verificación de fechas erróneas
- c. Carga (Load)
 - Se realizó carga en almacén de datos (dwh_SistVenta), este modelo implementa un proceso de carga incremental para cargar datos de manera incremental y realizar la carga de nuevos registros y/o de aquellos registros que cambian, garantizando una alta disponibilidad y escalabilidad.

5. Visualización

Se relaciona imagen de las ventas por empleado en un periodo de tiempo establecido



Conexión desde BI al DWH



CASO DE NEGOCIO

Solución Estratégica para la Gestión y Gobernanza de Datos

1. Introducción

Analizando la problemática presentada se evidencia que la entidad enfrenta varios desafíos relacionados con la gestión y gobernanza de datos debido a procesos desintegrados, falta de una estructura clara para la gestión de datos, y la necesidad de centralizar la información para mejorar la eficiencia operativa y la toma de decisiones. Este informe propone una solución integral basada en la centralización de procesos, gobernanza de datos robusta y seguridad fortalecida.

2. Arquitectura de Datos

2.1 Modelo de Datos Relacional

Para abordar la problemática de datos aislados y mejorar la integración entre las diferentes áreas, se recomienda adoptar un modelo de datos relacional centralizado. Este modelo debería incluir las siguientes características:

- **Base de Datos Centralizada:** Una única base de datos, que integre las diversas áreas de la entidad, alojando toda la información relacionada con cuentas de clientes, transacciones, inventarios, etc.
- **Esquema Relacional:** Definición de tablas con relaciones claras entre ellas, usando claves primarias y foráneas para garantizar la integridad referencial. Por ejemplo, las tablas de Clientes, Pedidos, y Productos pueden relacionarse mediante claves foráneas que garanticen la coherencia de la información.
- **Normalización de Datos:** Aplicación de técnicas de normalización (hasta 3NF) para eliminar redundancias y asegurar la consistencia de los datos.

2.2 Repositorio Centralizado de Datos

Para mejorar la toma de decisiones, se sugiere implementar un Data Warehouse que centralice la información histórica y permita análisis avanzados a partir de los datos operacionales utilizando procesos ETL (Extract, Transform, Load).

3. Herramientas y Tecnologías

3.1 Procesos ETL

Para la integración y centralización de los datos, es fundamental implementar procesos ETL que extraigan datos de los sistemas actuales, los transformen para cumplir con los estándares del nuevo modelo de datos y los carguen en la base de datos centralizada, mediante SQL Server Integration Services (SSIS) el cual es ideal para manejar cargas masivas de datos y realizar transformaciones complejas.

3.2 Visualización

Para mejorar la toma de decisiones, es crucial dotar a las áreas clave de herramientas de visualización que faciliten el acceso a la información relevante:

- **Power BI:** Excelente para crear paneles interactivos y reportes dinámicos, proporcionando a los usuarios una visión clara del desempeño y tendencias.
- **Tableau:** Otra opción poderosa para visualización, especialmente útil si se requiere integración con diversas fuentes de datos externas o grandes volúmenes de datos.

4. Seguridad y Cumplimiento

4.1 Políticas de Acceso

La centralización de datos requiere un enfoque robusto en seguridad:

- **Roles y Permisos:** Definir roles basados en las funciones y responsabilidades de cada usuario dentro de la organización, limitando el acceso a los datos sensibles únicamente a los usuarios autorizados.
- **Autenticación y Autorización:** Implementar autenticación y autorización basada en roles para asegurar que solo los usuarios adecuados accedan a la información.

4.2 Auditorías y Monitoreo

Para garantizar la integridad y seguridad de los datos, es esencial implementar:

- **Auditorías Periódicas:** Realizar auditorías regulares para revisar accesos, cambios en la base de datos, y asegurar el cumplimiento de las políticas de seguridad.
- **Monitoreo Continuo:** Usar herramientas como SQL Server Audit y verificación de logs para monitorizar actividades sospechosas en tiempo real, con alertas automatizadas ante posibles violaciones de seguridad.

5. Conclusión

La implementación de esta solución estratégica permitirá a la entidad centralizar y gobernar sus datos de manera eficiente, asegurando la integridad, seguridad y accesibilidad de la información. Esto no solo optimizará los procesos internos, sino que también mejorará la toma de decisiones a nivel organizacional, permitiendo a la entidad ofrecer un mejor servicio a sus clientes y cumplir con los estándares de seguridad y cumplimiento requeridos.