

UNIVERSIDAD TÉCNICA NACIONAL INGENIERIA DE

SOFTWARE



ISW-911 Minería de Datos

Proyecto 1

Randy Corea González

Emily Arias Arrieta

Karen Jiménez Espinoza

Docente

Freddy Rocha Boza

Tabla de Contenido

Introducción.....	3
Enunciado del problema	4
Objetivo General.....	5
Objetivos Específicos	5
Descripción de la solución.....	6
1. Capa de Staging:.....	6
2. Capa de Data Warehouse:	6
Tabla de Hechos:	7
Proceso de Implementación.....	7
Características Implementadas	7
Usuarios y Beneficiarios.....	7
Herramientas Utilizadas	8
Diagrama E/R del DW.....	9
Scripts Adjuntos (documentados).....	10
Conclusión.....	20
Bibliografía.....	21

Introducción

El presente proyecto aborda la implementación de un Data Warehouse para Technologies Inc S.A., una iniciativa estratégica diseñada para transformar la gestión de datos y potenciar la toma de decisiones basada en información dentro de la organización. En un entorno empresarial cada vez más competitivo y orientado a datos, la capacidad de integrar, analizar y utilizar eficientemente la información se ha convertido en un factor crítico para el éxito organizacional. Este trabajo responde a la necesidad urgente de centralizar las diversas fuentes de datos corporativos en un repositorio unificado que permita análisis multidimensionales y facilite la extracción de insights valiosos para el crecimiento sostenible de la empresa.

Se llevará a cabo una investigación tanto teórica como práctica, con el objetivo de analizar los principios fundamentales que sustentan los procesos dentro de un Data Warehouse. El estudio profundizará en las metodologías de modelado dimensional, las arquitecturas más eficientes, y las mejores prácticas en procesos ETL (Extracción, Transformación y Carga), elementos esenciales para comprender la relevancia de cada fase en la gestión eficiente de datos empresariales. Este enfoque metodológico garantizará que la solución implementada no solo cumpla con los estándares técnicos más exigentes, sino que también se alinee perfectamente con los objetivos estratégicos y operativos de Technologies Inc S.A.

La aplicación práctica de los conocimientos adquiridos constituye un componente fundamental de este proyecto, trascendiendo el ámbito puramente académico para adentrarse en la resolución de problemas reales de negocio. A través del diseño e implementación de un Data Warehouse funcional, los participantes desarrollarán competencias altamente valoradas en el mercado actual, como la capacidad de traducir requerimientos de negocio en soluciones técnicas, la optimización de flujos de datos complejos, y la creación de herramientas analíticas que faciliten la interpretación y visualización de información crítica. Esta experiencia inmersiva no solo reforzará la comprensión conceptual, sino que también preparará a los miembros del equipo para enfrentar los desafíos emergentes en el campo de la inteligencia de negocios y la analítica de datos.

Enunciado del problema

La empresa Technologies Inc S.A, tiene la necesidad de llevar a cabo la implementación de un data warehouse. Con este propósito, se invita a los miembros del equipo a participar en este desafío. Se organizarán grupos de 3 personas para llevar a cabo una investigación teórico-práctica centrada en el proceso de ETL (Extracción, Transformación y Carga) y Data Warehouse. Cada grupo deberá explorar y comprender los fundamentos teóricos detrás de estos procesos, así como llevar a cabo una aplicación práctica para fortalecer la comprensión y habilidades relacionadas con la creación efectiva de un data warehouse. Este ejercicio no solo permitirá a los estudiantes adquirir conocimientos sólidos en el ámbito de data warehousing, sino que también les brindará la oportunidad de enfrentarse a un escenario práctico relevante para el entorno laboral actual.

Objetivo General

Desarrollar e implementar un data warehouse completo para Technologies Inc S.A. que integre las diversas fuentes de datos de la empresa, permitiendo un análisis eficiente de la información para la toma de decisiones estratégicas.

Objetivos Específicos

- Identificar las fuentes de datos clave que servirán como base para alimentar el Data Warehouse, asegurando que la información sea relevante, confiable y optimizada para su análisis.
- Diseñar en Visual Studio Code las estructuras de dimensiones y hechos, organizando los datos de manera eficiente para potenciar la toma de decisiones estratégicas.
- Crear tablas dinámicas en Excel que permitan a cualquier usuario interactuar fácilmente con los datos, visualizando información clave de forma clara y accesible.

Descripción de la solución

Technologies Inc S.A. requiere implementar un data warehouse para analizar y extraer datos disponibles, facilitando el acceso eficiente para sus colaboradores. Esta solución busca atender necesidades críticas de negocio mediante la consolidación de información proveniente de múltiples fuentes.

La solución implementada está alineada con los siguientes objetivos empresariales:

- Facilitar análisis avanzados y generación de reportes
- Proporcionar soporte para toma de decisiones basadas en datos
- Optimizar la eficiencia operativa
- Identificar nuevas oportunidades de negocio

Se han definido metas concretas para evaluar el éxito de la implementación:

- Reducción del 10% en tiempo de generación de reportes
- Creación de una sección de consulta de datos históricos empresariales

La implementación sigue un modelo de dos capas principales:

1. Capa de Staging:

Funciona como área intermedia donde se integran y normalizan datos provenientes de diferentes orígenes mediante vistas:

- **V_PRODUCTO:** Unifica información de productos desde dos fuentes distintas (PRODUCTO y Products), normalizando códigos, nombres y categorías.
- **V_EMPLEADO:** Consolida datos de empleados con estandarización de formatos telefónicos, direcciones y normalización de países.
- **V_CLIENTE:** Integra información de clientes con segmentación basada en volumen de ventas (Clases A-E).

2. Capa de Data Warehouse:

Implementa un modelo dimensional tipo estrella con:

Tablas de Dimensiones:

- **DIM_EMPLEADO:** Información detallada de la fuerza laboral
- **DIM_PRODUCTO:** Catálogo completo de productos con categorización

- **DIM_TIEMPO:** Facilita análisis cronológicos con granularidad diaria
- **DIM_CLIENTE:** Datos de clientes con su categorización y representantes

Tabla de Hechos:

FACT_VENTAS: Registra transacciones con referencias a todas las dimensiones, cantidades y precios

Proceso de Implementación

Migración inicial: Utilización de Microsoft SQL Server Migration Assistant for Oracle para transferir datos desde sistemas origen

Limpieza y estandarización: Normalización de formatos, mayúsculas y categorías (ejemplo: estandarización de nombres de países)

Estructuración dimensional: Implementación del modelo estrella con relaciones referenciales

Integración de fuentes: Unificación de datos desde al menos dos sistemas distintos

Verificación de integridad: Implementación de restricciones mediante claves foráneas

Características Implementadas

Normalización de datos: Garantiza consistencia en formatos y nomenclaturas

Segmentación automática: Clasificación de clientes según patrones de compra

Dimensión temporal: Permite análisis históricos y evolutivos

Referencial integrity: Asegura coherencia entre dimensiones y hechos

Modelo escalable: Estructura preparada para crecimiento futuro

Usuarios y Beneficiarios

La solución está orientada a múltiples perfiles de usuarios:

Gerentes para toma de decisiones estratégicas

Analistas de negocio para estudios detallados

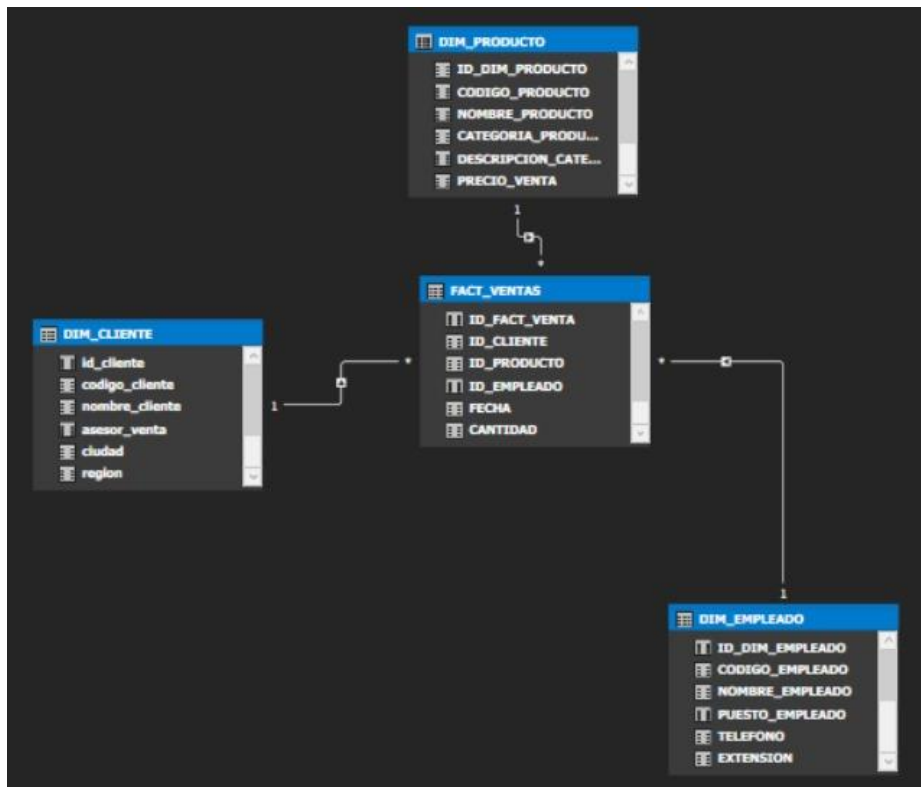
Personal operativo para consultas diarias

Herramientas Utilizadas

- Microsoft SQL Server: Motor de base de datos principal
- SQL Server Migration Assistant: Facilitador de migración desde Oracle
- Excel: Interfaz de visualización para usuarios finales

Esta implementación permitirá a Technologies Inc S.A. aprovechar el poder de sus datos para impulsar decisiones informadas, mejorar su eficiencia operativa y descubrir nuevas oportunidades de negocio a través de un acceso unificado y estructurado a su información corporativa.

Diagrama E/R del DW



Primeramente, debemos de crear las tablas las cuales debemos de insertar los datos correspondientes de la empresa Technologies Inc S.A. para luego proceder con los diferentes procedimientos requeridos. Se deben de crear las tablas EMPLEADO, PRODUCTO, TIEMPO y FAC_VENTAS de esta manera se podrá llevar un control de los movimientos internos de la empresa utilizando diversas técnicas para conocer los ingresos y gastos; y a la vez valorar el rendimiento de cada colaborador.

Scripts Adjuntos (documentados)

```
--- Creacion de Dimensiones en DW

Use DW

-- Empleado

CREATE TABLE DIM_EMPLEADO (
  id_empleado INT IDENTITY(1,1) PRIMARY KEY,
  codigo_empleado INT NOT NULL,
  nombre VARCHAR(50),
  apellido1 VARCHAR(50),
  apellido2 VARCHAR(50),
  extension VARCHAR(20),
  email VARCHAR(100),
  codigo_oficina VARCHAR(10),
  codigo_jefe INT,
  puesto VARCHAR(50)
);

-- Productos

CREATE TABLE DIM_PRODUCTO (
  id_producto INT IDENTITY(1,1) PRIMARY KEY,
  codigo_producto VARCHAR(50) NOT NULL,
  nombre VARCHAR(100) NOT NULL,
  gama VARCHAR(50),
  dimensiones VARCHAR(50),
  proveedor VARCHAR(100),
  descripcion VARCHAR(255),
  cantidad_en_stock INT,
  precio_venta DECIMAL(10,2),
  precio_proveedor DECIMAL(10,2)
);

-- Tiempo

create table DIM_TIEMPO
(
  idtiempo int identity constraint pk_tiempo PRIMARY KEY CLUSTERED,
  Fecha date not null,
  Anio smallint not null,
  Mes smallint not null,
  Dia smallint not null,
  NMes nvarchar(30) not null,
  DiaAnio int
)

-- Fact_Ventas /Tabla de hechos

CREATE TABLE FACT_VENTAS
(
  ID_FACT_VENTA int identity constraint PK_FACT_VENTAS PRIMARY KEY CLUSTERED,
  ID_CLIENTE SMALLINT NOT NULL,
  ID_PRODUCTO NVARCHAR(10) NOT NULL,
  ID_EMPLEADO SMALLINT NOT NULL,
  FECHA DATE not null,
  CANTIDAD SMALLINT NOT NULL,
  PRECIO DECIMAL NOT NULL
)
```

Una vez creadas las tablas, se insertarán los datos de la siguiente manera:

```

-- Insertar datos en la dimension

--Empleado
INSERT INTO DIM_EMPLEADO (codigo_empleado, nombre, apellido1, apellido2, extension, email, codigo_oficina, codigo_jefe, puesto)
SELECT CODIGO_EMPLEADO, NOMBRE, APELLIDO1, APELLIDO2, EXTENSION, EMAIL, CODIGO_OFICINA, CODIGO_JEFE, PUESTO
FROM staging.dbo.v_empleado;

select * from DIM_EMPLEADO

-- Producto
INSERT INTO DIM_PRODUCTO(CODIGO_PRODUCTO, NOMBRE_PRODUCTO, CATEGORIA_PRODUCTO, DESCRIPCION_CATEGORIA, PRECIO_VENTA, CANTIDAD_EN_STOCK,PROVEEDOR)
SELECT CODIGO_PRODUCTO, NOMBRE_PRODUCTO, CATEGORIA_PRODUCTO, DESCRIPCION_CATEGORIA, PRECIO_VENTA, CANTIDAD_EN_STOCK,PROVEEDOR FROM STAGING.DBO.V_PRODUCTO

SELECT * FROM DIM_PRODUCTO

-- Fact_Ventas
INSERT INTO FACT_VENTAS (ID_CLIENTE, ID_PRODUCTO, ID_EMPLEADO, FECHA, CANTIDAD, PRECIO)
SELECT
DC.ID_CLIENTE ID_CLIENTE,
DPD.CODIGO_PRODUCTO ID_PRODUCTO,
DE.CODIGO_EMPLEADO ID_EMPLEADO,
CAST(P.FECHA_ENTREGA AS DATE)FECHA,
DP.CANTIDAD CANTIDAD,
DP.PRECIO_UNIDAD PRECIO
FROM STAGING.DBO.PEDIDO P
INNER JOIN STAGING.DBO.DETALLE_PEDIDO DP
ON P.CODIGO_PEDIDO = DP.CODIGO_PEDIDO
INNER JOIN DIM_CLIENTE DC
ON CAST(P.CODIGO_CLIENTE AS VARCHAR(10)) = DC.CODIGO_CLIENTE
INNER JOIN DIM_PRODUCTO DPD
ON DP.CODIGO_PRODUCTO = DPD.CODIGO_PRODUCTO
INNER JOIN DIM_EMPLEADO DE
ON DC.ASESOR_VENTA = DE.NOMBRE_EMPLEADO
WHERE FECHA_ENTREGA IS NOT NULL;

SELECT * FROM FACT_VENTAS

```

```

-- Cliente

ALTER VIEW V_CLIENTE AS

WITH SEGMENTOS_CLIENTE AS (
    SELECT CAST(CODIGO_CLIENTE AS VARCHAR (10)) CODIGO_CLIENTE, SUM(DP.CANTIDAD * DP.PRECIO_UNIDAD) VENTAS,
    NTILE(4) OVER (ORDER BY SUM(DP.CANTIDAD * DP.PRECIO_UNIDAD) DESC) SEGMENTO
    FROM PEDIDO P
    INNER JOIN DETALLE_PEDIDO DP
    ON P.CODIGO_PEDIDO = DP.CODIGO_PEDIDO
    WHERE ESTADO = 'ENTREGADO'
    GROUP BY CODIGO_CLIENTE
    UNION
    SELECT CustomerID, SUM(OD.Quantity * OD.UnitPrice) VENTAS,
    NTILE(4) OVER (ORDER BY SUM(OD.Quantity * OD.UnitPrice) DESC) SEGMENTO
    FROM Orders O
    INNER JOIN OrderDetails OD
    ON O.OrderID = OD.OrderID
    GROUP BY CustomerID
)
SELECT
VMC.CODIGO_CLIENTE,
UPPER(NOMBRE_CLIENTE) NOMBRE_CLIENTE,
UPPER(ISNULL(ASESOR_VENTAS, 'Lorena Paxton')) ASESOR_VENTAS,
UPPER(CIUDAD) CIUDAD,
UPPER(ISNULL(REGION, CIUDAD)) REGION,
UPPER(CASE WHEN PAIS = 'UK' THEN 'UNITED KINGDOM' ELSE PAIS END) PAIS,
CASE WHEN ISNULL(SEGMENTO, 0) = 1 THEN 'CLASE A'
WHEN ISNULL(SEGMENTO, 0) = 2 THEN 'CLASE B'
WHEN ISNULL(SEGMENTO, 0) = 3 THEN 'CLASE C'
WHEN ISNULL(SEGMENTO, 0) = 4 THEN 'CLASE D'
ELSE 'CLASE E' END CATEGORIA
FROM (
    SELECT CAST(CJ.CODIGO_CLIENTE AS VARCHAR (10)) CODIGO_CLIENTE, CJ.NOMBRE_CLIENTE, CONCAT(EJ.NOMBRE, ' ', EJ.APELLIDO1, ' ', EJ.APELLIDO2) ASESOR_VENTAS,
    CJ.CIUDAD, CJ.REGION, CJ.PAIS
    FROM CLIENTE CJ
    LEFT JOIN EMPLEADO EJ ON CJ.CODIGO_EMPLEADO_REP_VENTAS = EJ.CODIGO_EMPLEADO
)
UNION ALL

SELECT CN.CustomerID, CN.CompanyName, CONCAT(EN.FirstName, ' ', EN.LastName) ASESOR,
CN.City, CN.Region, CN.Country
FROM Customers CN
LEFT JOIN (
    SELECT CustomerID, EmployeeID FROM Orders O
    WHERE OrderID=(SELECT MAX(OrderID) FROM Orders O2 WHERE O2.CustomerID = O.CustomerID)
    GROUP BY CustomerID, EmployeeID
) AE
ON CN.CustomerID = AE.CustomerID
LEFT JOIN Employees EN ON AE.EmployeeID = EN.EmployeeID
)VMC

LEFT JOIN SEGMENTOS_CLIENTE SC
ON VMC.CODIGO_CLIENTE = SC.CODIGO_CLIENTE

SELECT *
FROM V_CLIENTE

```

```

-- Empleado

CREATE VIEW V_EMPLEADO AS (
SELECT
    UPPER(EJ.CODIGO_EMPLEADO) AS CODIGO_EMPLEADO,
    UPPER(CONCAT(EJ.NOMBRE, ' ', EJ.APELLIDO1, ' ', EJ.APELLIDO2)) AS NOMBRE_EMPLEADO,
    UPPER(EJ.PUESTO) AS PUESTO_EMPLEADO,
    UPPER(
        CASE
            WHEN OJ.TELEFONO LIKE '+%'
            THEN REPLACE(REPLACE(REPLACE(OJ.TELEFONO, '(', ''), ')', ''), ' ', '-')
            ELSE '+' + REPLACE(REPLACE(REPLACE(OJ.TELEFONO, '(', ''), ')', ''), ' ', '-')
        END
    ) AS TELEFONO, -- Quita paréntesis, agrega + si no lo tiene y cambia espacios por -
    UPPER(EJ.EXTENSION) AS EXTENSION,
    UPPER(ISNULL(OJ.REGION, OJ.CIUDAD)) AS REGION, -- Si REGION es NULL, toma CIUDAD
    UPPER(OJ.CIUDAD) AS CIUDAD,
    UPPER(
        CASE
            WHEN OJ.PAIS IN ('USA', 'EEUU') THEN 'ESTADOS UNIDOS'
            WHEN OJ.PAIS IN ('INGLATERRA', 'UK') THEN 'INGLATERRA'
            ELSE OJ.PAIS
        END
    ) AS PAIS, -- Unificación de nombres de países
    UPPER(OJ.CODIGO_POSTAL) AS CODIGO_POSTAL,
    UPPER(CONCAT(OJ.LINEA_DIRECCION1, ' ', ISNULL(OJ.LINEA_DIRECCION2, ''))) AS DIRECCION
FROM EMPLEADO EJ
LEFT JOIN OFICINA OJ ON OJ.CODIGO_OFICINA = EJ.CODIGO_OFICINA

UNION ALL

SELECT
    UPPER(EN.EmployeeID) AS CODIGO_EMPLEADO,
    UPPER(CONCAT(EN.FirstName, ' ', EN.LastName)) AS NOMBRE_EMPLEADO,
    UPPER(EN.Title) AS PUESTO_EMPLEADO,
    UPPER(
        CASE
            WHEN EN.HomePhone LIKE '+%'
            THEN REPLACE(REPLACE(REPLACE(EN.HomePhone, '(', ''), ')', ''), ' ', '-')
            ELSE '+' + REPLACE(REPLACE(REPLACE(EN.HomePhone, '(', ''), ')', ''), ' ', '-')
        END
    ) AS TELEFONO, -- Quita paréntesis, agrega + si no lo tiene y cambia espacios por -
    UPPER(EN.Extension) AS EXTENSION,
    UPPER(ISNULL(EN.Region, EN.City)) AS REGION, -- Si REGION es NULL, toma CIUDAD
    UPPER(EN.City) AS CIUDAD,
    UPPER(
        CASE
            WHEN EN.Country IN ('USA', 'EEUU') THEN 'ESTADOS UNIDOS'
            WHEN EN.Country IN ('INGLATERRA', 'UK') THEN 'INGLATERRA'
            ELSE EN.Country
        END
    ) AS PAIS, -- Unificación de nombres de países
    UPPER(EN.PostalCode) AS CODIGO_POSTAL,
    UPPER(EN.Address) AS DIRECCION
FROM Employees EN
);

SELECT * FROM V_EMPLEADO

```

Estas vistas tienen varios propósitos importantes:

Integrar datos de múltiples fuentes: Combinan información usando UNION ALL para crear una vista unificada de los datos.

Estandarizar formatos: Convierten todos los datos textuales a mayúsculas (UPPER), normalizan formatos telefónicos, estandarizan nombres de países (por ejemplo, "UK" se convierte en "UNITED KINGDOM") y aseguran consistencia en los datos.

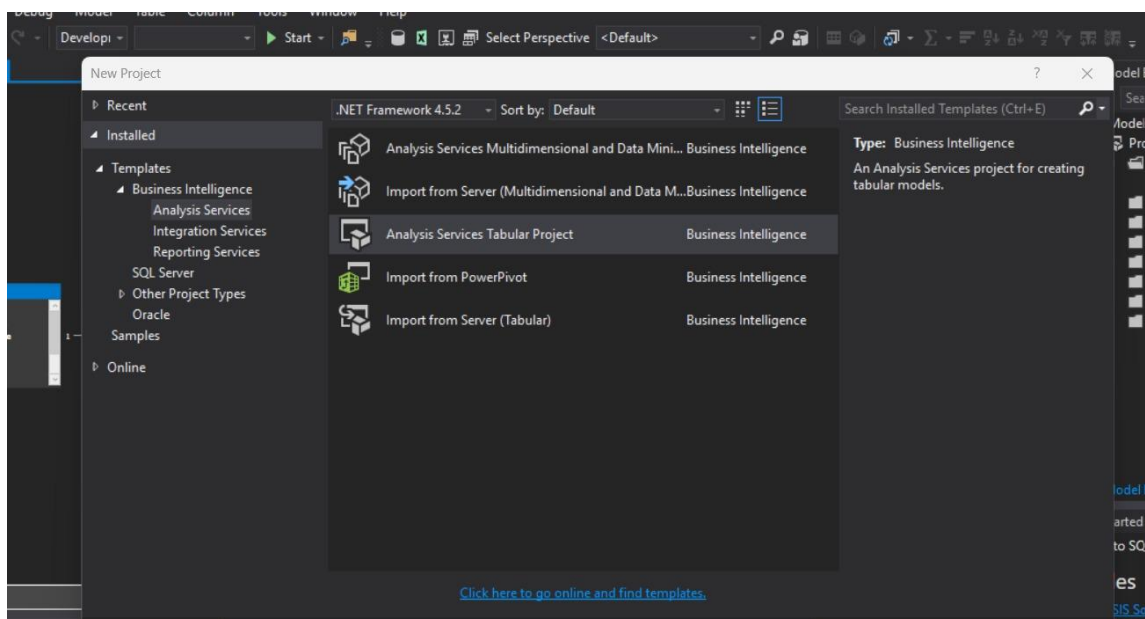
Enriquecer los datos con información adicional: Por ejemplo, la vista V_CLIENTE clasifica a los clientes en categorías (A, B, C, D, E) basadas en su cantidad de compras.

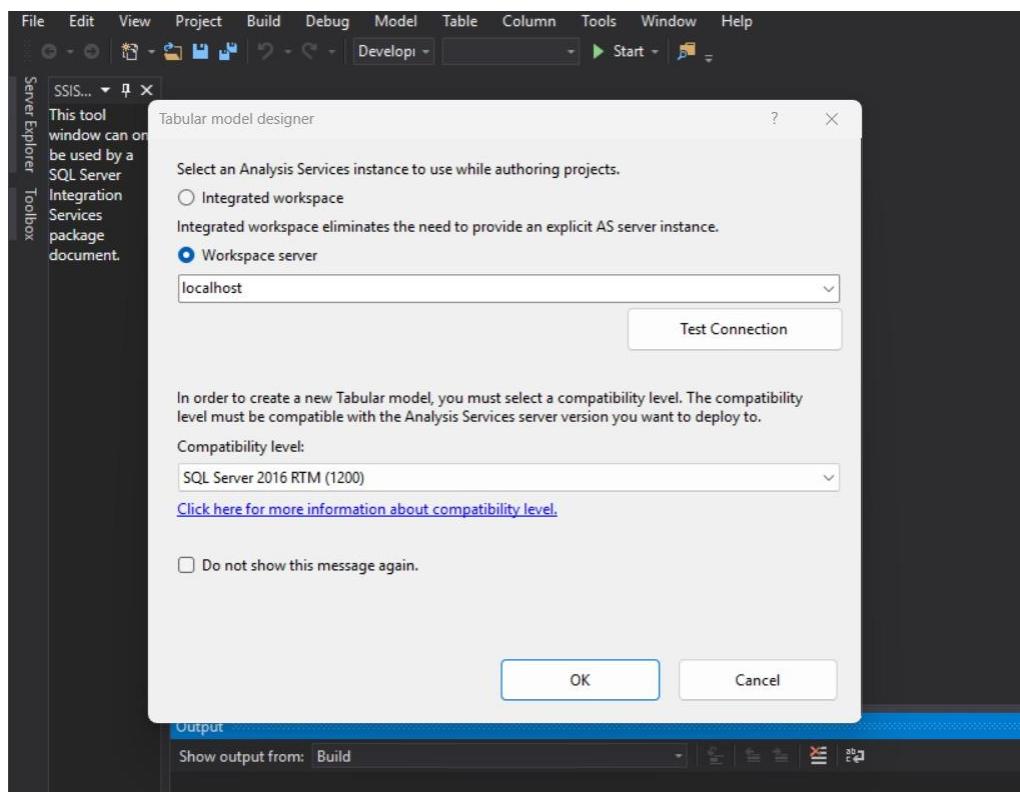
Limpiar y corregir datos: Manejan valores nulos (ISNULL), dan formato a direcciones y concatenan datos relacionados.

Crear una capa de abstracción: Permiten que las aplicaciones accedan a datos limpios y formateados sin tener que interactuar directamente con las tablas subyacentes o preocuparse por su estructura.

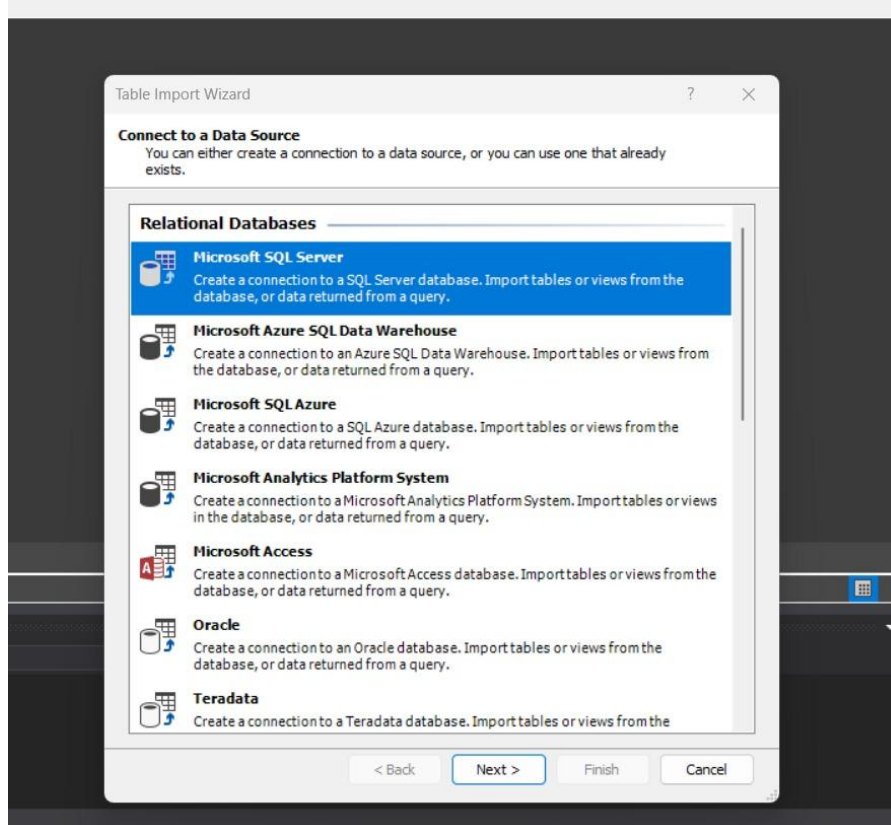
En conclusión, estas vistas funcionan como una capa intermedia de transformación de datos que facilita el proceso ETL (Extracción, Transformación y Carga) para migrar información entre sistemas o para preparar datos para un Datawarehouse o sistema de análisis. De esta manera los datos se consultarán de manera más ordena y limpia posible.

Seguidamente debemos de realizar un cubo utilizando las tablas que creamos anteriormente.



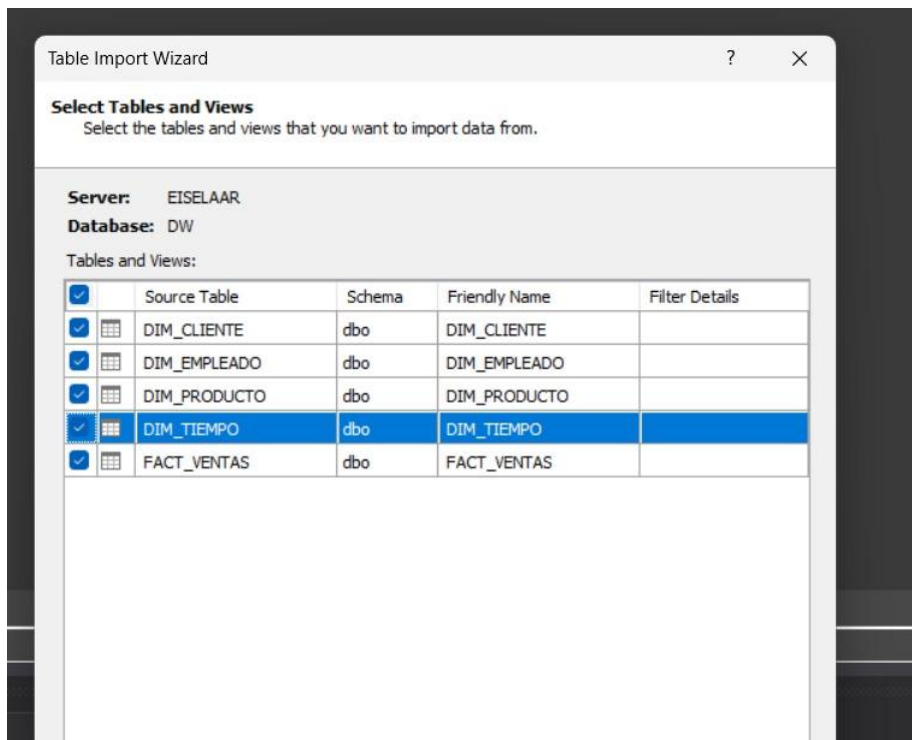


Seleccionamos MICROSOFT SQL SERVER:



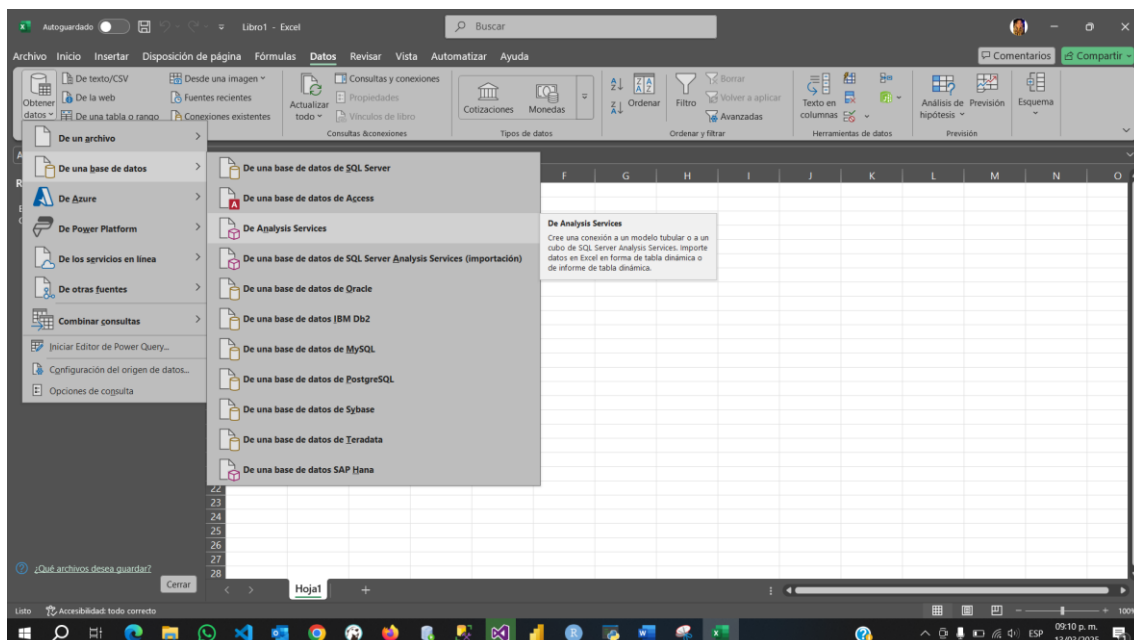
The screenshot shows a 'Table Import Wizard' window with the title 'Table Import Wizard'. The main heading is 'Connect to a Microsoft SQL Server Database' with the instruction 'Enter the information required to connect to the Microsoft SQL Server database.' Below this, there are several input fields: 'Friendly connection name' with the value 'SqlServer EISELAAR', 'Server name' with a dropdown menu showing 'EISELAAR', and 'Database name' with an empty dropdown menu. Under the 'Log on to the server' section, there are two radio buttons: 'Use Windows Authentication' (selected) and 'Use SQL Server Authentication'. Below these are fields for 'User name' and 'Password', and a checkbox for 'Save my password'. At the bottom right of the main area are buttons for 'Advanced' and 'Test Connection'. At the very bottom of the window are navigation buttons: '< Back', 'Next >', 'Finish', and 'Cancel'.

Si la conexión se realiza de manera correcta se mostrarán los datos que se encuentran en nuestra base de datos. Luego se ingresan el usuario y la contraseña y listo.



Una vez creado el cubo podemos continuar con la creación de tablas de Excel conectado desde las bases de datos de sql.

Seguimos los siguientes pasos:



Asistente para la conexión de datos

Conectar con el servidor de la base de datos

Escriba la información requerida para conectar con el servidor de la base de datos.

1. Nombre del servidor:

2. Credenciales de conexión

☒ Utilizar autenticación de Windows

☐ Utilizar el nombre de usuario y la contraseña siguientes

Nombre de usuario:

Contraseña:


Asistente para la conexión de datos

Seleccionar base de datos y tabla

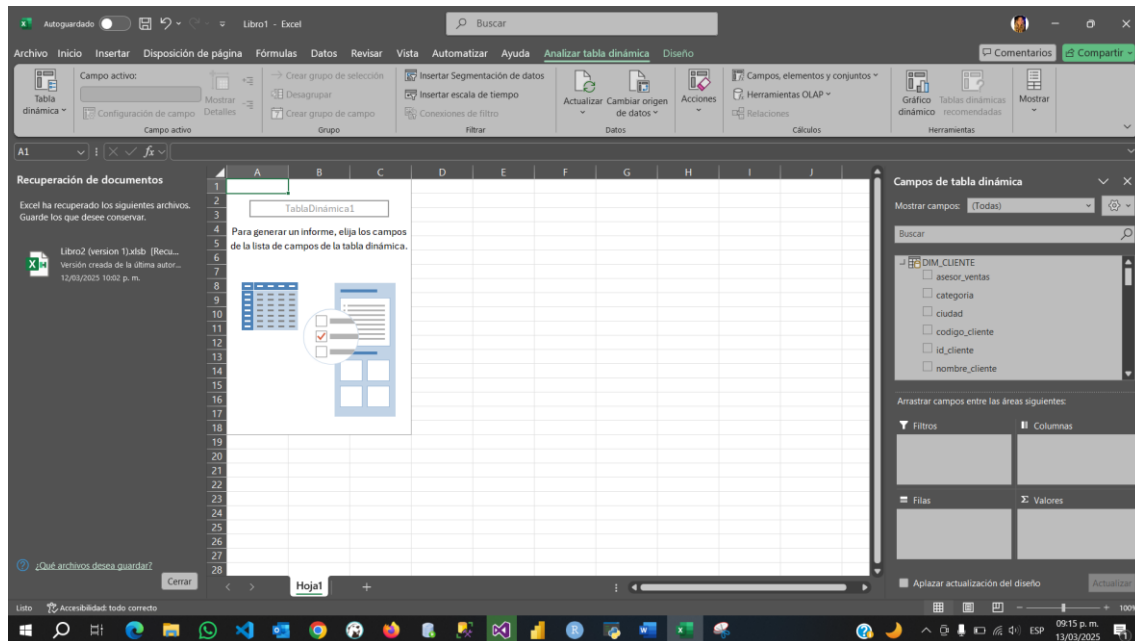
Seleccione la base de datos y la tabla o el cubo que contenga los datos que desea obtener.

Seleccione la base de datos que contiene la información que desea:

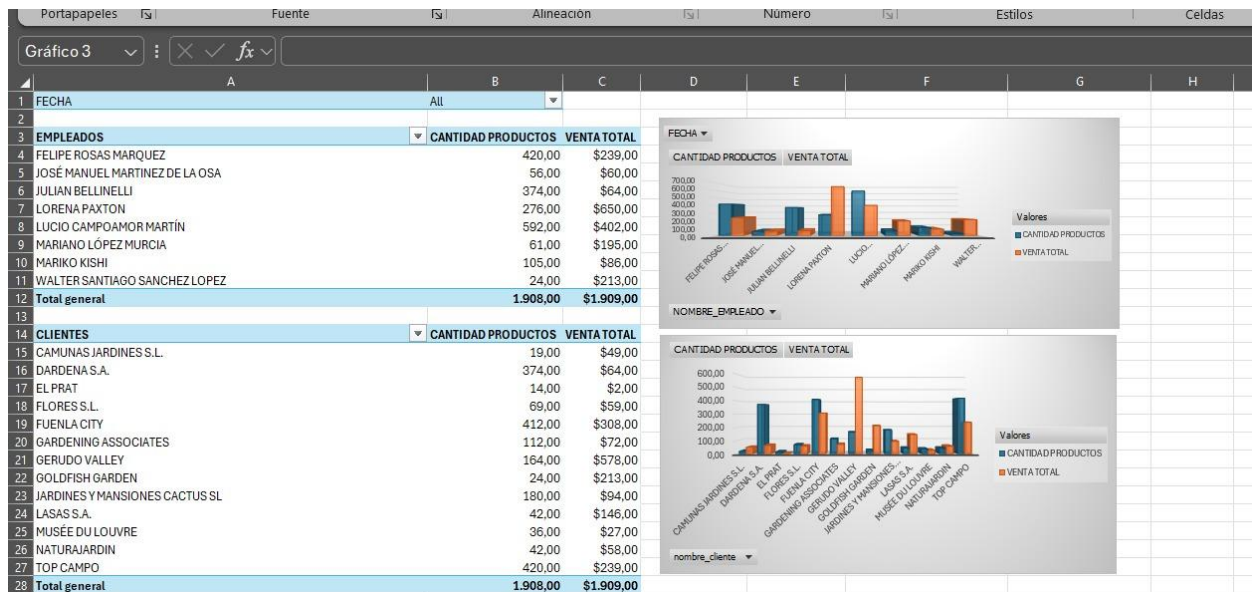
☒ Conectar con una tabla o a un cubo específico:

Nombre	Descripción	Modificado	Creado	Tipo
 Model		3/12/2025 9:23:13 PM		CUBE

Finalizados los siguientes pasos tendremos acceso a las tablas y podremos realizar diversas consultas de una manera más dinámica y crear nuestras propias tablas de Excel con consultas personalizadas dependiendo de los requerimientos del cliente.



Una vez creadas las tablas podremos visualizar los datos desde las tablas de Excel mediante Analysis Services.



Conclusión

En conclusión, un data warehouse o almacén de datos es un sistema centralizado que recopila, integra y almacena grandes volúmenes de datos estructurados provenientes de múltiples fuentes dentro de una organización. Está específicamente diseñado para facilitar análisis, reportes y consultas complejas.

Esta herramienta nos ayuda a organizar de manera más eficiente y también realizar consultas de manera mas especifica.

La implementación estratégica de tecnologías especializadas como Migrating Oracle Databases to SQL Server y Analysis Services nos permite:

- Ejecutar operaciones ETL (Extracción, Transformación y Carga) con mayor eficiencia
- Diseñar y desplegar cubos OLAP robustos que permiten análisis desde múltiples perspectivas
- Proporcionar visualizaciones dinámicas de datos complejos

Bibliografía

- MashaMSFT. (2025, 8 enero). Oracle to SQL Server: Migration guide - SQL Server. Microsoft Learn. <https://learn.microsoft.com/en-us/sql/sql-server/migrate/guides/oracle-to-sql-server?view=sql-server-ver16>
- Calbimonte, D. (2019, 16 diciembre). *Cómo construir un cubo desde cero usando SQL Server Analysis Services (SSAS)*. SQL Shack - Articles About Database Auditing, Server Performance, Data Recovery, And More. <https://www.sqlshack.com/es/como-construir-un-cubo-desde-cero-usando-sql-server-analysis-services-ssas/>