Alumnos:

Josué Mejías Villalobos

Fernando Gonzales Rojas

Erick Arguello Paniagua

Curso:

Minería de datos

Código:

ISW-911

Proyecto:

Data Warehouse

Docente:

Freddy Gerardo Rocha Boza

Año:

2024

Tabla de contenido

Introducción

El desarrollo del data warehouse es un tipo de sistema de gestión de datos diseñado para permitir y respaldar las actividades de inteligencia empresarial, especialmente el análisis. Este se enfoca en un proceso integral de extracción, transformación y Cargar de datos, que garantiza la integración y consistencia de la información almacenada.

Actualmente, el uso efectivo de la información se ha vuelto un factor fundamental para que las empresas puedan competir con éxito y tomar decisiones organizadas. En este presente proyecto el objetivo es reforzar la calidad de los datos desde diversas fuentes, aparte de, se facilitará su limpieza, transformación y almacenamiento organizado, garantizando a la empresa de contar con una herramienta sólida y confiable para el análisis de información y la toma de decisiones. Para determinar el éxito del proyecto, se llevarán a cabo tres aspectos fundamentales.

Primero, se realiza el análisis y la evaluación de las fuentes de datos tanto internas como externas, para evaluar su relevancia calidad. En segundo lugar, se considera el diseño del diagrama adecuado y el desarrollo de los procesos ETL necesarios para la transformación y la carga de los datos de manera efectiva; y finalmente, la implementación de las consultas específicas que permitirán a la empresa obtener información al instante de manera precisa y relevante para decisiones estratégicas.

Enunciado del problema

Tecnologies Inc. S.A se enfrenta al desafío de reforzar y administrar grandes cantidades de datos de diferentes fuentes, tanto internas como externas. En la actualidad, la carencia de un sistema unido de almacenamiento y procesamiento de datos reduce la capacidad de este para adquirir información clara y apropiada, y así dificultar la toma de decisiones operativas. Esta circunstancia es vulnerable por la variedad de formatos y la calidad desbalanceada de los datos, lo que limita su utilidad y fiabilidad para el análisis de datos.

La creación de un data warehouse se visualiza como una solución completa que permite centralizar, transformar y almacenar datos de manera ordenada y normalizada. No obstante, para que esta solución sea eficaz, se requiere establecer un proceso de extracción, transformación y carga que garantice la calidad, fiabilidad y solidez de la información en cada fase del proyecto. Adicionalmente, la creación de consultas especificas permitirá a la empresa el acceso rápido a la información sobresaliente para la toma de decisiones importantes.

Por ende, el problema que se quiere solucionar con este proyecto es la carencia de un sistema de almacenamiento y análisis de datos seguro y agrupado que permita a la empresa optimizar sus procesos de toma de decisiones y sostener una ventaja competitiva en su industria.

Objetivos

Objetivos General

Desarrollar un data warehouse para Technologies Inc. S.A, por medio de la implementación de un proceso ETL que permita la integración, limpieza y almacenamiento de datos de diferentes fuentes, facilitando el análisis de información para la toma de decisiones estratégicas.

Objetivos específicos

Identificar las fuentes de datos internas y externas que se utilizarán para alimentar el data warehouse, garantizando su calidad y trascendencia.

Diseñar el esquema y los procesos ETL del data warehouse, mediante la extracción, transformación y carga de datos desde diversas fuentes, asegurando la calidad y consistencia de la información

Implementar consultas útiles para el análisis de datos, facilitando la toma de decisiones estratégicas y operativas en Technologies Inc. S.A

Descripción de la solución

Para iniciar con la solución de este proyecto, se establecieron las fuentes de datos para llenar la tabla de Staging. Para esto se importaron datos de una base de Oracle la cual se llama Jardinería, por otro lado, se importaron datos de la base de datos Northwind que ya esta alojada en SQL Server. Para este proceso se utilizaron scripts de importación, y por ende en la base datos quedaron tabla tanto de Northwind como de Jardinería.

Una vez tenemos los datos, es turno de filtrarlos y escogerlos, por lo tanto se crearon vistas especificas para llenar la información de las tablas de dimensiones y hechos. El proceso se hizo de la siguiente forma.

Transportista:

Para la tabla de transportista de obtuvieron los datos de ID, nombre_transportista y teléfono. El caso de la tabla jardineria no existían datos de transportista, por lo que se obtó por asignarle uno llamado 'Transporte Jardineria'.

Producto:

Para la tabla de producto se obtuvieron los datos de ID, nombre_producto, categoría, precio_unitario, unidades_en_stock y proveedor.

En el caso de los datos traídos de Northwind se les tradujo al español las categoría, para tener un mejor entendimiento y consistencia en los datos.

Las columnas de cantidad_por_unidad y unidades_en_orden se eliminaron ya que la información no estaba en la tabla de jardineria y tampoco son datos relevantes para la solución del proyecto.

También se optó por cambiar la estructura de la columna de proveedor_key, ya que para jardineria hay una tabla de proveedor, por ende, esta columna se cambió a tipo alfanumérico (varchar) y en esta solo se alberga el nombre del proveedor.

Proveedor:

Debido a lo anteriormente comentado, se optó eliminar esta tabla por falta de datos y que ya se menciona en la tabla de productos.

Empleado:

Para esta tabla se eligieron los valores de ID, nombre_empleado, puesto, fecha_de_contratacion, país, ciudad.

Para la identificación mediante IDs se les añadió los siguientes prefijos 'JD_' para los datos provenientes de jardineria y 'NW_' para los datos provenientes de Northwind.

En el caso de las columnas de fecha_contratacion, país y ciudad provenientes de jardineria, se uso la primera fecha dentro de la tabla de tiempo, 'CR' como país y 'San Jose' como ciudad.

Por otro lado, en el puesto de los datos de northwind se tradujeron al español para tener consistencia y un mejor entendimiento al analizar los datos.

Además, se eliminan las columnas de jefatura y salario debido a que no son necearias para nuestro análisis de datos.

Cliente

Para esta tabla se eligieron los valores de ID, nombre_cliente, país, ciudad, código_postal y telefono.

Para tener consistencia en la columna de país, se programó para que los Estados Unidos siempre se menciones como ‘US’.

También, se eliminaron las tablas de sexo, fecha_nacimiento, categoría y asesor, por inexistencia de datos.

Tabla de hechos (ventas)

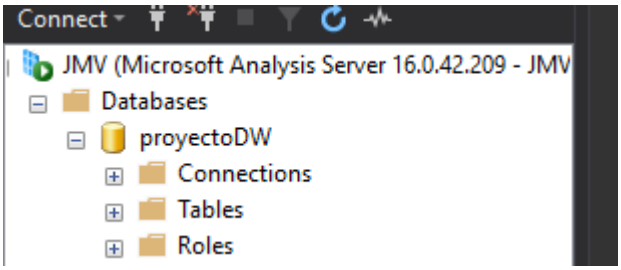
Para esta tabla se utilizaron los siguientes valores: ventas_id, fecha_key, cliente_key, producto_key, empleado_key, transportista_key, cantidad, precio_unitario, descuento y total_ventas.

Se elimina proveedor_key, ya se elimino la tabla de dimensiones de proveedor.

Luego a esta se le añadieron sus respectivas llaves foráneas para vincularla con las tablas de hechos.

Una vez creadas las tablas y las vistas, se cargar los todos los registros. Y una vez terminado, pasamos a crear el análisis service en Visual Studio.

Para esto creamos un proyecto de análisis service, y le vinculamos como data sourse nuestra base de datos de warehouse, una vez creado este análisis service, se puede ver refleado al entrar al management studio en la sección de análisis services:

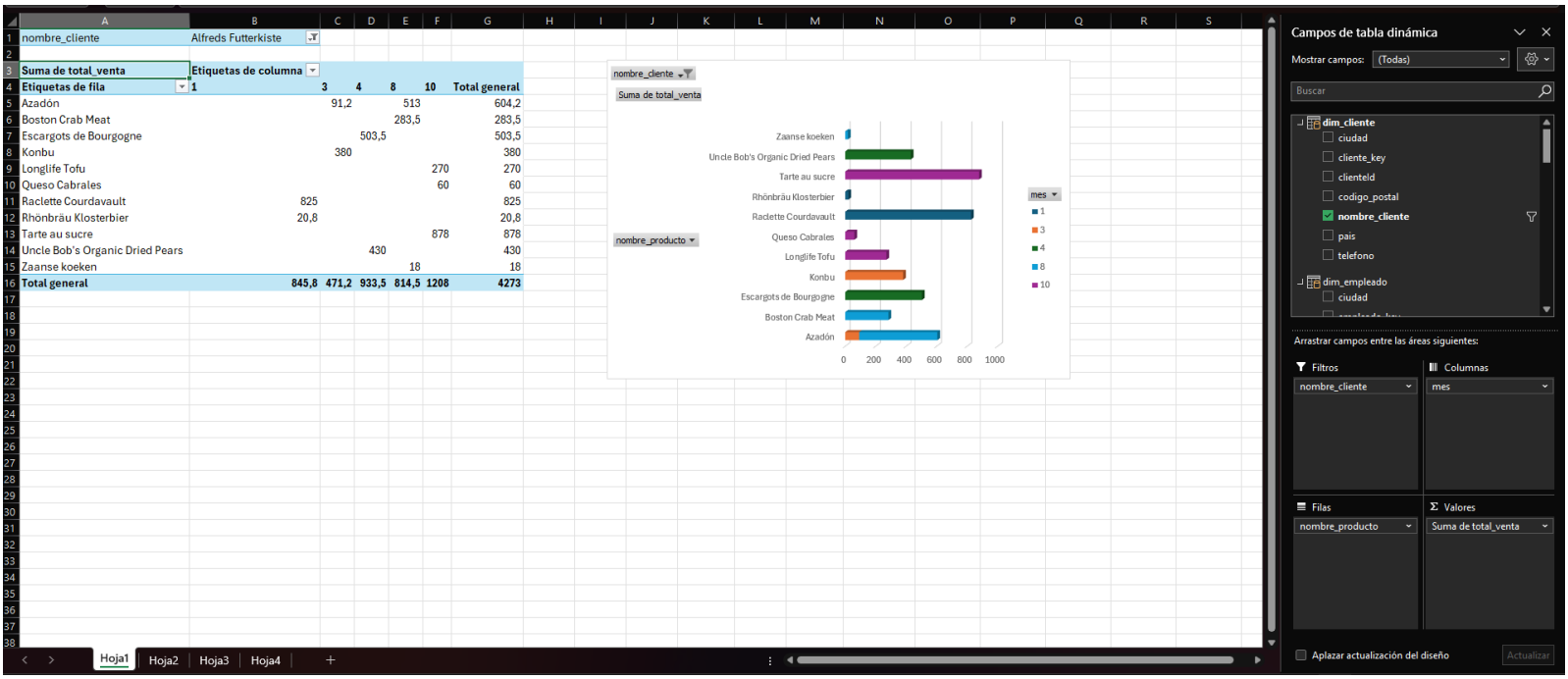


Ahora pasamos a Excel donde importamos datos a través de este análisis service, y creamos los siguientes gráficos a partir de nuestra información:

Consultas de Excel

- **Ventas Totales por Producto, Cliente y Mes**

Esta consulta muestra el total de ventas para cada combinación de producto y mes, filtrado por clientes, permitiendo analizar cuáles productos generan más ingresos y qué clientes contribuyen más a las ventas en ciertos periodos.



nombre_cliente	Alfreds Futterkiste					
Suma de total_venta	Etiquetas de columna					
Etiquetas de fila	1	3	4	8	10	Total general
Azadón		91,2		513		604,2
Boston Crab Meat				283,5		283,5
Escargots de Bourgogne			503,5			503,5
Konbu		380				380
Longlife Tofu					270	270
Queso Cabrales					60	60
Raclette Courdavault	825					825
Rhönbräu Klosterbier	20,8					20,8
Tarte au sucre					878	878
Uncle Bob's Organic Dried Pears			430			430
Zaanse koeken				18		18
Total general	845,8	471,2	933,5	814,5	1208	4273

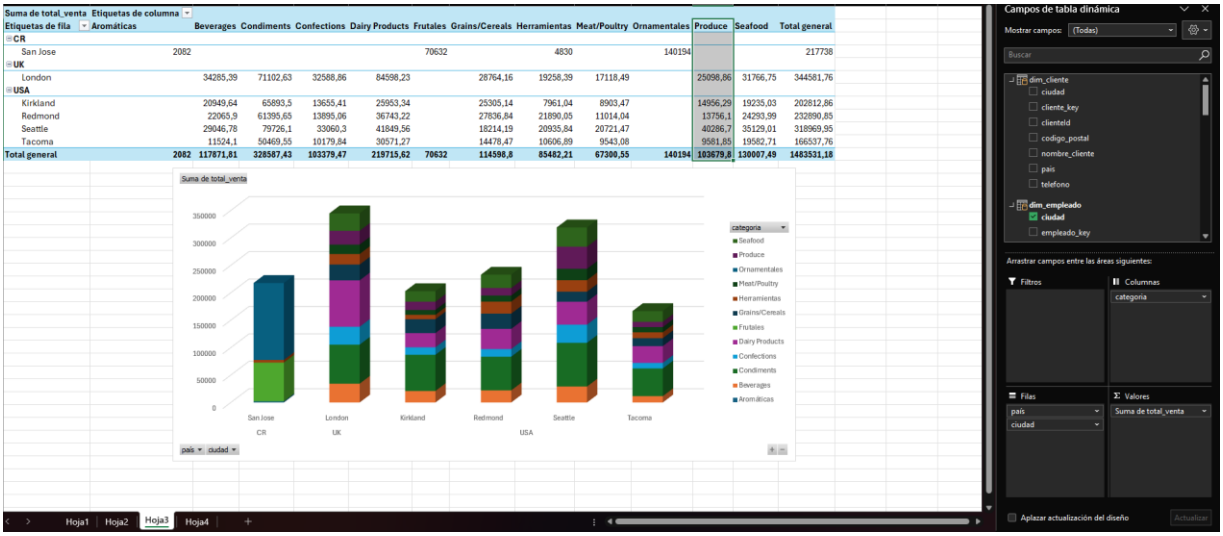
- **Ventas por Empleado, Producto y Trimestre**

Esta consulta permite ver el rendimiento de cada empleado por producto y año, lo cual es útil para evaluar el desempeño en diferentes periodos y los productos que mejor venden cada año.

nombre_producto	All				
Suma de total_venta	Etiquetas de columna				
Etiquetas de fila	2020	2021	2022	2023	Total general
Andrew Fuller	21757,06	70444,14	74336,56		166537,76
Anne Dodsworth	9894,52	26310,39	41103,17		77308,08
Emmanuel Magaña Perez				3319	3319
Felipe Rosas Marquez			1360	17902	19262
Janet Leverling	18223,96	108026,16	76562,74		202812,86
José Manuel Martinez De la Osa				504	504
Julian Bellinelli			6883	13687	20570
Laura Callahan	22240,12	56032,63	48589,55		126862,3
Lorena Paxton	1567	84960	16492	2624	105643
Lucio Campoamor Martín			18979	10972	29951
Margaret Peacock	49945,12	128809,79	54135,94		232890,85
Mariano López Murcia			4964	1442	6406
Mariko Kishi	553	859	2056	1024	4492
Michael Bolton			2100	1400	3500
Michael Suyama	16642,61	43126,37	14144,16		73913,14
Nancy Davolio	35764,51	93148,12	63195,02		192107,65
Robert King	15232,16	60471,19	48864,89		124568,24
Steven Buchanan	18383,92	30716,48	19691,9		68792,3
Walter Santiago Sanchez Lopez			5000	19091	24091
Total general	210203,98	702904,27	498457,93	71965	1483531,18

- **Ventas Totales por País, Ciudad y Categoría de Producto**

Esta consulta permite ver las ventas organizadas por país, ciudad, y categoría de producto. Esto es útil para analizar la demanda por ubicación geográfica y tipo de producto.



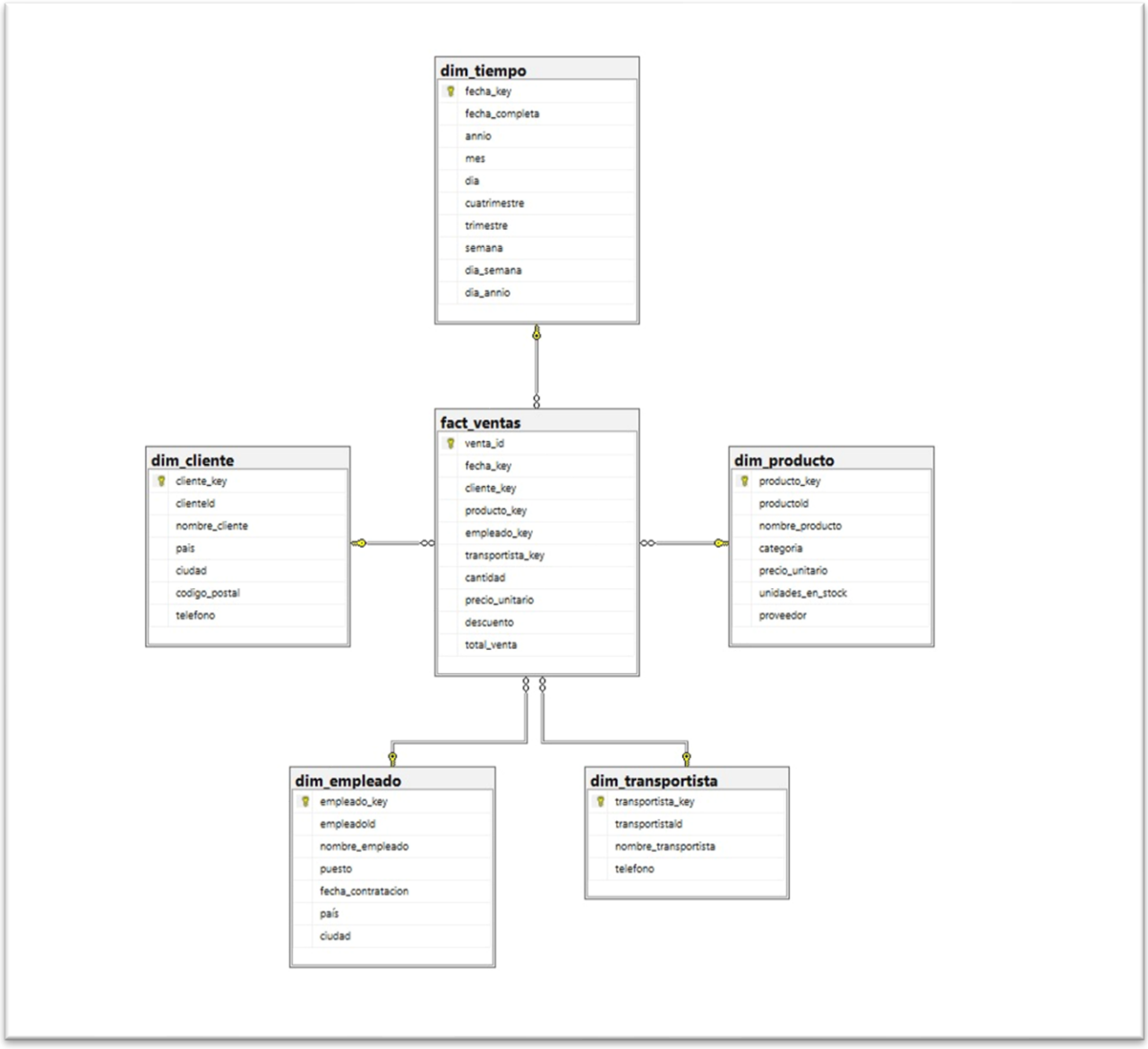
Suma de total_venta	Etiquetas de columna											
Etiquetas de fila	Aromáticas	Beverages	Condiments	Confections	Dairy Products	Frutales	Grains/Cereals	Herramientas	Meat/Poultry	Ornamentales	Produce	Seafood	Total general
CR													
San Jose	2082					70632		4830		140194			217738
UK													
London		34285,39	71102,63	32588,86	84998,23		28764,16	19258,39	17118,49		25098,86	31766,75	344581,76
USA													
Kirkland		20949,64	65893,5	13655,41	25953,34		25305,14	7961,04	8903,47		14956,29	19235,03	202812,86
Redmond		22065,9	61395,65	13895,06	36743,22		27836,84	21890,05	11014,04		13756,1	24293,99	232890,85
Seattle		29046,78	79726,1	33060,3	41849,56		18214,19	20935,84	20721,47		40286,7	35129,01	318969,95
Tacoma		11524,1	50469,55	10179,84	30571,27		14478,47	10606,89	9543,08		9581,85	19582,71	166537,76
Total general	2082	117871,81	328587,43	103379,47	219715,62	70632	114598,8	85482,21	67300,55	140194	103679,8	130007,49	1483531,18

- **Comparación de Ventas por Empleado y Transportista**

Esta consulta te permite ver cómo cada empleado contribuye a las ventas en combinación con el transportista, para analizar si ciertos transportistas están más asociados con ciertos empleados y si eso impacta en las ventas.

Etiquetas de fila	Suma de total_venta
Andrew Fuller	
Federal Shipping	41277,99
Speedy Express	63299,87
United Package	61959,9
Anne Dodsworth	
Federal Shipping	22718,2
Speedy Express	17694,56
United Package	36895,32
Emmanuel Magaña Perez	
Transporte Jardineria	3319
Felipe Rosas Marquez	
Transporte Jardineria	19262
Janet Leverling	
Federal Shipping	64357,33
Speedy Express	50455,87
United Package	87999,66
José Manuel Martinez De la Osa	
Transporte Jardineria	504
Julian Bellinelli	
Transporte Jardineria	20570
Laura Callahan	
Federal Shipping	35529,23
Speedy Express	33963
United Package	57370,07
Lorena Paxton	
Transporte Jardineria	105643
Lucio Campoamor Martín	
Transporte Jardineria	29951
Margaret Peacock	
Federal Shipping	67736,42
Speedy Express	58511,9
United Package	106642,53
Mariano López Murcia	
Transporte Jardineria	6406
Mariko Kishi	
Transporte Jardineria	4492
Michael Bolton	
Transporte Jardineria	3500

Diagrama E/R


dim_tiempo
fecha_key
fecha_completa
annio
mes
dia
cuatrimestre
trimestre
semana
dia_semana
dia_annio
fact_ventas
venta_id
fecha_key
cliente_key
producto_key
empleado_key
transportista_key
cantidad
precio_unitario
descuento
total_venta
dim_cliente
cliente_key
clienteId
nombre_cliente
pais
ciudad
codigo_postal
telefono
dim_producto
producto_key
productoId
nombre_producto
categoria
precio_unitario
unidades_en_stock
proveedor
dim_empleado
empleado_key
empleadoId
nombre_empleado
puesto
fecha_contratacion
país
ciudad
dim_transportista
transportista_key
transportistaId
nombre_transportista
telefono

Scripts Adjuntos

Códigos de las Tablas de Dimensión

- **Tabla de Transportistas del warehouse**

```
create table dim_transportista (
    transportista_key int identity(1,1) constraint
pk_transportista primary key,
    transportistaId int,
    nombre_transportista nvarchar(100),
    telefono nvarchar(20)
);
```

- **Tabla de Productos del warehouse**

```
create table dim_producto (
    producto_key int identity(1,1) constraint
pk_prodcuto primary key,
    productoId int,
    nombre_producto nvarchar(100),
    categoria nvarchar(20),
     proveedor_key int,
    cantidad_por_unidad nvarchar(50),
    precio_unitario decimal(10,2),
    unidades_en_stock int,
    unidades_en_orden int
);
```

- **Tabla de Empleado del warehouse**

```
create table dim_empleado (
    empleado_key int identity(1,1) constraint pk_empleado
primary key,
    empleadoId int,
    nombre_empleado nvarchar(100),
    puesto nvarchar(50),
    fecha_contratacion date,
    país nvarchar(50),
    ciudad nvarchar(50),
    jefatura varchar(80),
    salario numeric(25,2)
);
```

- **Tabla de Cliente del warehouse**

```
create table dim_cliente (
    cliente_key int identity(1,1) constraint  pk_cliente
primary key,
    clienteId nvarchar(5),
    nombre_cliente nvarchar(100),
     sexo varchar(15),
     fecha_nacimiento date,
    pais nvarchar(50),
    ciudad nvarchar(50),
    codigo_postal nvarchar(10),
    telefono nvarchar(20),
     categoria nvarchar(20),
     asesor  nvarchar(80)
);
```

- **Tabla de Ventas del warehouse**

```
create table fact_ventas(
    venta_id int identity(1,1) constraint pk_fac_ventas
primary key,
    fecha_key int,
    cliente_key int,
    producto_key int,
    empleado_key int,
     proveedor_key int,
    transportista_key int,
    cantidad int,
    precio_unitario decimal(10,2),
    descuento decimal(5,2),
    total_venta decimal(12,2)
);
```

- **Tabla de Tiempo del warehouse**

```
create table dim_tiempo (
    fecha_key int constraint  pk_tiempo primary key
identity,
    fecha_completa date,
    annio int,
    mes int,
    dia int,
    cuatrimestre int,
    trimestre int,
    semana int,
    dia_semana nvarchar(10),
    dia_annio int);
```

Código de las vistas implementadas

- **Vista de datos transportistas**

```
Alter view  v_dim_transportista
AS
SELECT ShipperID as transportistaId,
          CompanyName as nombre_transportista,
          Phone as telefono
FROM staging.dbo.SHIPPERS
UNION ALL
SELECT 4 AS transportistaId,
       'Transporte Jardineria' AS nombre_transportista,
       '(506) 8724-6262' AS telefono;
```

- **Vista de datos Producto**

```
CREATE VIEW v_dim_producto
AS
SELECT
     CAST(P.ProductID AS varchar(10)) as productoId,
    P.ProductName AS nombre_producto,
    CASE
        WHEN C.CategoryName = 'Beverages' THEN 'Bebidas'
        WHEN C.CategoryName = 'Condiments' THEN
'Condimentos'
        WHEN C.CategoryName = 'Confections' THEN
'Confites'
        WHEN C.CategoryName = 'Dairy Products' THEN
'Productos Lácteos'
        WHEN C.CategoryName = 'Grains/Cereals' THEN
'Granos/Cereales'
```

```
        WHEN C.CategoryName = 'Meat/Poultry' THEN
'Carne/Aves'
        WHEN C.CategoryName = 'Produce' THEN 'Producir'
        WHEN C.CategoryName = 'Seafood' THEN 'Mariscos'
        ELSE C.CategoryName
    END AS categoria,
    P.UnitPrice AS precio_unitario,
    P.UnitsInStock AS unidades_en_stock,
    S.CompanyName AS proveedor
FROM
    staging.dbo.PRODUCTS P
INNER JOIN
    staging.dbo.CATEGORIES C ON P.CategoryID =
C.CategoryID
INNER JOIN
    staging.dbo.suppliers S ON S.SupplierID =
P.SupplierID


UNION


SELECT
     P.CODIGO_PRODUCTO AS productoId,
    P.NOMBRE AS nombre_producto,
    P.GAMA AS categoria,
    P.PRECIO_VENTA AS precio_unitario,
    P.CANTIDAD_EN_STOCK AS unidades_en_stock,
    P.PROVEEDOR AS proveedor
FROM
    Staging.DBO.producto P;
```

- **Vista de datos de empleado**

```
alter VIEW v_dim_empleado

AS

SELECT

     CONCAT('JD_',CODIGO_EMPLEADO) AS empleadoId, ---
se usa un prefijo para identifar de donde viene y que no
hayan coincidencias en las ids

    CONCAT(E.Nombre, ' ', E.Apellido1, ' ', E.Apellido2)
AS nombre_empleado,

    E.puesto AS puesto,

    '2020-01-01' AS fecha_contratacion,  -- se establace
la primera fecha como predeterminada

    'CR' AS pais,                -- Costa rica como pais
predeterminado

    'San Jose' AS ciudad              -- San José como
ciudad predeterminada

FROM

    staging.dbo.empleado E

UNION

SELECT

    CONCAT('NW_', EmployeeID) AS empleadoId,

    CONCAT(e.FirstName, ' ', e.LastName) AS nombre_empleado,

    CASE

        WHEN e.Title = 'Inside Sales Coordinator' THEN 'Director
Oficina'

        WHEN e.Title = 'Sales Manager' THEN 'Director General'

        WHEN e.Title = 'Sales Representative' THEN 'Representante
Ventas'

        WHEN e.Title = 'Vice President, Sales' THEN 'Subdirector
Ventas'

        ELSE e.Title
```

```
    END AS puesto,
    e.HireDate AS fecha_contratacion,
    e.Country AS pais,
    e.City AS ciudad
FROM
    Staging.DBO.employees e;
```

❖ **Vista de datos de cliente**

```
CREATE VIEW v_dim_cliente
AS
select CustomerID as clienteId,
  c.CompanyName as nombre_cliente,
  CASE
        WHEN c.Country = 'USA' THEN 'US'
        ELSE c.Country
    END AS pais,
  c.City as ciudad,
  c.PostalCode as codigo_postal,
  c.Phone as telefono
from staging.dbo.CUSTOMERS c

union all

select cast(c.CODIGO_CLIENTE as varchar(10)) as
clienteId,
  c.NOMBRE_CLIENTE as nombre_cliente,
  CASE
        WHEN c.PAIS = 'USA' THEN 'US'
        ELSE c.PAIS
```

```
    END AS pais,
  c.CIUDAD as ciudad,
  c.CODIGO_POSTAL as codigo_postal,
  c.TELEFONO as telefono
from staging.dbo.CLIENTE c;
```

- **Vista de datos de ventas**

```
CREATE VIEW v_fact_ventas
AS
select
    (select fecha_key from dim_tiempo where
fecha_completa = dateadd(year, 24, cast(o.orderdate as
date))) fecha_key,
    (select cliente_key from dim_cliente where clienteId
= o.customerid) cliente_key,
     (select producto_key from dim_producto where
isnumeric(producto_key)=1
     and cast(producto_key as varchar(10)) =
cast(od.productid as varchar(10)))producto_key,
     (select empleado_key from dim_empleado where
substring(empleadoId,4,3) = CAST(o.employeeid AS
varchar(10))
     and substring(empleadoId,1,3)='NW_') as
empleado_key,
    (select transportista_key from dim_transportista
where transportista_key = o.shipvia) transportista_key,
     od.quantity as cantidad,
    od.unitprice as precio_unitario,
    od.discount as descuento,
    od.quantity * od.unitprice * (1 - od.discount) as
total_venta
```

```
from staging.dbo.orders o

inner join staging.dbo.[orderdetails] od on o.orderid =
od.orderid -- se cambiar a orderdetails

inner join staging.dbo.products p on od.productid =
p.productid

union all

select

     (select fecha_key from dim_tiempo where
fecha_completa = p.fecha_pedido)fecha_key,

    (select cliente_key from dim_cliente where clienteId
= CAST(p.codigo_cliente AS VARCHAR(10))) AS cliente_key,
-- Se ajustan las referencias y se castea a varchar para
que sean compatibles

     (select producto_key from dim_producto where --
isnumeric(producto_key)=0 and

     productoId = dp.codigo_producto)producto_key,    -
- se ajustan las referencias

          (select empleado_key from dim_empleado
     where substring(empleadoId,4,3) =
cl.codigo_empleado_rep_ventas -- se adapto para que
funcionara con string y los prefijos

     and substring(empleadoId,1,3)='JD_')empleado_key,

    4 transportista_key,

     dp.cantidad,

    dp.precio_unidad,

    0 as descuento,

    dp.cantidad * dp.precio_unidad as total_venta

from staging.dbo.pedido p

inner join staging.dbo.detalle_pedido dp

on p.codigo_pedido=dp.codigo_pedido

left join staging.dbo.cliente cl

on cl.codigo_cliente=p.codigo_cliente;
```

❖ **Método de llenado de la tabla tiempo**

```
declare @fecha_inicial date = '2020-01-01';  -- fecha
inicial

declare @fecha_final date = '2025-12-31';    -- fecha
final


while @fecha_inicial <= @fecha_final

begin

    insert into dim_tiempo

    (fecha_completa, annio, mes, dia, cuatrimestre,

    trimestre, semana, dia_semana,dia_annio)

    values (

        @fecha_inicial,                                 --
fecha_completa

        year(@fecha_inicial),                           --
año

        month(@fecha_inicial),                          --
mes

        day(@fecha_inicial),                            --
día

        case                                           --
cuatrimestre

            when month(@fecha_inicial) between 1 and 3
then 1

            when month(@fecha_inicial) between 4 and 6
then 2

            when month(@fecha_inicial) between 7 and 9
then 3
```

```
            else 4
        end,
        case                                            --
trimestre
            when month(@fecha_inicial) between 1 and 3
then 1
            when month(@fecha_inicial) between 4 and 6
then 2
            when month(@fecha_inicial) between 7 and 9
then 3
            else 4
        end,
        datepart(week, @fecha_inicial),              --
semana (número de la semana)
        datename(weekday, @fecha_inicial) ,           -
- día_semana (nombre del día de la semana)
        datepart(dayofyear,@fecha_inicial)
    );


    -- incrementar la fecha actual en un día
    set @fecha_inicial = dateadd(day, 1,
@fecha_inicial);
end;
```

Conclusión

El desarrollo de un data warehouse para Tecnologies Inc. S.A manifiesta un avance representativo en la capacidad de la empresa para administrar y analizar grandes cantidades de datos de manera agrupada y eficaz. Por medio del proceso de extracción, transformación y carga, se ha logrado incluir datos derivados de diferentes fuentes, asegurando su calidad y consistencia, los cuales son aspectos cruciales para la confiabilidad en el análisis de la información.

En este presente proyecto se logró fortalecer un sistema de almacenamiento estructurado, además, se ha establecido una base sólida para futuras expansiones en el análisis de datos y generación de reportes. La creación de consultas determinadas y bien diseñadas favorece a los usuarios acceder rápidamente a la información importante, mejorando así los procesos de toma de decisiones estratégicas y operativas de la empresa.

En conclusión, la implementación de este data warehouse posibilitara a la empresa enfrentar de manera eficaz los desafíos de un entorno empresarial dinámico, garantizando una ventaja competitiva mediante el beneficio óptimo de los datos.

Referencias

Oracle. (2024). *OCI*. Obtenido de https://www.oracle.com/database/what-is-a-data-warehouse/