**UNIVERSIDAD TÉCNICA NACIONAL**

**INGENIERIA DE SOFTWARE**

**MINERIA DE DATOS**



***ETL y Datawarehouse***

Realizado por:

Jose Arturo Quirós Araya

Jose Alí Rivas Gómez

Hanzel

Docente

Freddy

CONTENIDO

[**INTRODUCCIÓN** 3](#_Toc129513199)

[**ENUNCIADO DEL PROBLEMA** 4](#_Toc129513200)

[**OBJETIVO GENERAL** 5](#_Toc129513201)

[**OBJETIVOS ESPECÍFICOS** 5](#_Toc129513202)

[**DESCRIPCIÓN DE LA SOLUCIÓN** 6](#_Toc129513203)

[**APARTADO DOCUMENTAL** 6](#_Toc129513204)

[**ESTRATEGIAS PARA LA CREACIÓN DE UN PROYECTO ETL** 6](#_Toc129513205)

[**ETAPAS DE ETL** 7](#_Toc129513206)

[**METODOLOGIAS** 8](#_Toc129513207)

[**APARTADO PRACTICO** 11](#_Toc129513208)

[**AMBIENTE DE PRODUCCIÓN** 11](#_Toc129513209)

[**AREA DE PREPARACIÓN (STAGING AREA)** 12](#_Toc129513210)

[**DATAWAREHOUSE** 16](#_Toc129513211)

[**CONCLUSIÓN** 26](#_Toc129513212)

[**BIBLIOGRAFÍA** 27](#_Toc129513213)

# **INTRODUCCIÓN**

En un mundo altamente digitalizado como el que vivimos la información es un insumo que se genera a gigantescas cantidad y es ahí donde el análisis de estos ser convierte en un reto.

De este modo, en la empresa Northwind posee una amplia Base de Datos producto de sus operaciones diarias en venta de comidas, sin embargo, el análisis de esta información es realizado mediante reportes estáticos y el uso de herramientas de ofimática como Excel lo que limita el potencial de los insumos que se pueden desarrollar producto de este análisis.

Por lo tanto, mediante el desarrollo de un proceso de ETL (Extracción, Transformación y Limpieza) de datos y la construcción de un Datawarehouse que sirva de insumo para la posterior construcción de tableros con KPI’s importantes para Northwind.

# **ENUNCIADO DEL PROBLEMA**

La construcción de un datawarehouse es un proceso que no debe ser tomado a la ligera debido a que su uso está ortientado a la toma de decisiones y la incorrecta presentación de la información podría ocasionar la ejecución de decisiones con impactos altos en la operativa.

Para este proceso, es necesario realizar un análisis a fondos de la estructura presente en la Base de Datos de Producción, y luego en conjunto con las áreas de negocio identificar aquella información que es relevante para las partes interesadas encargadas de la dirección del mismo con el fin de establecer el contenido que posteriormente formará parte del Datawarehouse.

Para la construcción del Datawarehouse es necesario el desarrollo e implementación de un Staging Area que sirva de insumo para este, de modo que se pueda acceder a información relativamente similar a la de producción pero sin afectar este ambiente.

Finalmente se deberá realizar un proceso de ETL desde el SA hacia el DW para cargarlos con la información correcta que posteriormente servirá de insumo para el consumo de herramientas de Análisis de Datos.

# **OBJETIVO GENERAL**

Desarrollar un Datawarehouse a partir de la base de datos Northwind mediante un proceso de ETL.

# **OBJETIVOS ESPECÍFICOS**

* Identificar las reglas de negocio para la construcción de un Datawarehouse.
* Investigar las mejores practicas acerca de un proceso de ETL y Datawarehouse.
* Desarrollo de un ambiente de Stagin Area (SA).
* Desarrollo de un proceso de ETL.
* Desarrollo de un Datawarehoouse.

# **DESCRIPCIÓN DE LA SOLUCIÓN**

## **APARTADO DOCUMENTAL**

### **ESTRATEGIAS PARA LA CREACIÓN DE UN PROYECTO ETL**

Los procesos ETL (Extract, Transform, Load) consisten en un conjunto de técnicas, herramientas y tecnologías que permiten extraer datos de varias fuentes, transformarlos de forma que sean veraces y útiles, y cargarlos en otros sistemas con el fin de que puedan ser accesibles por los niveles de la organización que así lo requieran con el fin de aportar a la toma de decisiones.

A la hora de desarrollar un proceso de ETL el primer paso es el desarrollo de una estrategia en conjunto con el área de negocio de la empresa u organización para la cual se va a desarrollar el proceso.

Para el desarrollo de una estrategia de ETL, el principal insumo que se va a requerir es el objetivo principal, por ejemplo: “Analizar las ventas y sus indicadores asociados para determinar los mejores vendedores y clientes”.

Con este insumo, el encargado o líder del proyecto deberá comenzar a realizar un análisis de las diferentes fuentes de información con las que cuenta la organización y cuáles son sus características para dar comienzo al proyecto.

### **ETAPAS DE ETL**

#### **ANÁLISIS**

En la etapa de análisis el líder del proyecto (el cual deberá ser una persona con suficiente conocimiento técnico y con amplio conocimiento del área de negocio) en conjunto con las partes interesadas realizará la identificación y establecimiento de los objetivos de la organización.

En base a estos objetivos, establecidos generalmente por una alta dirección o personal de nivel gerencial, el líder del proyecto realizará una identificación de las posibles fuentes de datos que servirán de insumo.

#### **DISEÑO**

En esta etapa, el líder del proyecto comenzará con el diseño de un área de preparación, en la cual se cargará la información proveniente de producción, posteriormente realizará un aálisis de la información actual con el fin de identificar si cumple con los requisitos necesarios para ser utilizada, finalmente diseñará la estructura del datawarehouse tomando en cuenta los datos que este contendrá, su estructura y su contenido.

#### **CONSTRUCCIÓN**

Al comienzo, se identifican todas las fuentes de datos y se extraen los mismos al área de preparación (Stagin Area) esto corresponde a la etapa de Extracción.

Posteriormente, los datos extraídos de las fuentes se compilan, convierten, reformatean y limpian en el área de preparación (Stagin Area) para alimentar la base de datos de destino (Datawarehouse), esta es la etapa de **Transformación y Limpieza.**

Finalmente, los datos procedentes de la fase anterior (fase de Transformación y Limpieza) son cargados en el sistema de destino que formará nuestro Datawarehouse.

### **METODOLOGIAS**

Al diseñar una solución de Datawarehouse, las organizaciones se topan con distintas metodologías a seguir, cada una con sus ventajas. Así como ciertas características limitantes. Las cuales se deberán evaluar para seleccionar la que mejor se adapte a los requisitos del proyecto y la organización.

#### **RALPH KIMBALL - MULTIDIMENSIONAL**

La metodología multidimensional o metodología de **Ralph Kimball**, mantiene un diseño ascendente. Por lo que los **Data Marts** son los primeros en crearse y después se integran al **Data Warehouse**, donde se busca que el almacenamiento de datos de los usuarios se ejecute de la forma más rápida posible.

Según Kimball, un Data Warehouse es la copia de los datos transaccionales específicamente estructurados para consultas analíticas e informes, con el fin de apoyar la toma de decisiones. Con esta metodología, al crear primero los **Data Marts** se proporcionan capacidades analíticas de informes para procesos específicos de negocio y funcionales.

**Principales Ventajas:**

* No requiere un equipo muy grande de desarrolladores y arquitectos de datos para mantener el Data Warehouse (menor costo).
* Brinda buena funcionalidad y seguimiento de las métricas o KPI´s, orienta los Data Marts a informes en cuanto a procesos de departamento o de negocios.
* Administración más “simplificada” al estar concentrado en los procesos y las áreas individuales en vez de toda la organización.
* La optimización de consultas es sencilla, predecible y controlable.

**Principales Desventajas:**

* Por su enfoque en procesos y áreas, puede no llegar a cubrir o manejar todos los requisitos en los informes.
* Consta de una menor flexibilidad de modificación.

Diagrama

Descripción generada automáticamente

#### **RALPH KIMBALL – RELACIONAL**

La metodología relacional de Bill Inmon muestra un diseño descendente, donde se construye primero el Data Warehouse y posteriormente los Data Marts. Ubicando el Data Warehouse en el centro de la información corporativa lo que asegura un marco lógico en los datos.

Crea una estructura de entidades procurando que no se repitan datos. Este modelo crea una única fuente de verdad para todo el negocio.

La carga de datos se vuelve menos compleja debido a la estructura normalizada del modelo. Sin embargo, el uso de esta disposición para realizar consultas es complicado; ya que incluye gran cantidad de tablas y vínculos.

Este modelo propone la construcción de Data Marts por separado para cada departamento. Todos los datos que entran en el Data Warehouse están integrados para garantizar la integridad y la coherencia en toda la empresa. Actuando el Data Warehouse como el único origen de datos.

**Principales Ventajas:**

* El Data Warehouse proporciona una única versión de la verdad, al ser el único origen de datos para los Data Marts.
* Tiene una mayor facilidad de comprensión de los procesos empresariales para los usuarios, ya que el modelo lógico representa entidades empresariales detalladas.
* Resulta más fácil y menos propenso al fracaso el proceso de ETL, puesto que en la actualización de los datos y las anomalías se evitan al contar con una redundancia muy baja.
* Mayor flexibilidad a cambios por necesidades analíticas, de negocio y/o por fuentes de datos.

**Principales Desventajas:**

* De mayor complejidad, se requieren recursos con mayor capacidad en modelado y almacenamiento de datos (generalmente de mayor costo).
* Suele requerir de tiempos más largos para dado su procesos y entrega.

Diagrama

Descripción generada automáticamente

## **APARTADO PRACTICO**

### **AMBIENTE DE PRODUCCIÓN**

La base de datos Northwind es una base de datos de muestra creada originalmente por Microsoft y utilizada como base para sus tutoriales en una variedad de productos de bases de datos durante décadas. La base de datos de Northwind contiene los datos de ventas de una empresa ficticia llamada "Northwind Traders", que importa y exporta alimentos especiales de todo el mundo. Esta base de datos fue replicada en SQL Server y fue llamada NorwhtindDB

Diagrama, Esquemático

Descripción generada automáticamente

### **AREA DE PREPARACIÓN (STAGING AREA)**

Posteriormente, en SQL Server se desarrolló la creación de la Base de Datos de NorthwindSA

Diagrama, Esquemático

Descripción generada automáticamente

Esto mediante los siguientes scripts

*-- Categories*

CREATE TABLE Categories(

    CategoryID INT,

    CategoryName VARCHAR(15),

    Description VARCHAR(MAX),

    Picture VARCHAR(MAX)

    )

*-- Customers*

CREATE TABLE Customers(

    CustomerID VARCHAR(5),

    CompanyName VARCHAR(40),

    ContactName VARCHAR(30),

    ContactTitle VARCHAR(30),

    Address VARCHAR(60),

    City VARCHAR(15),

    Region VARCHAR(15),

    PostalCode VARCHAR(10),

    Country VARCHAR(15),

    Phone VARCHAR(24),

    Fax VARCHAR(24)

    )

*-- Employees*

CREATE TABLE Employees(

    EmployeeID INT,

    LastName VARCHAR(20),

    FirstName VARCHAR(10),

    Title VARCHAR(30),

    TitleOfCourtesy VARCHAR(25),

    BirthDate DATETIME ,

    HireDate DATETIME ,

    Address VARCHAR(60),

    City VARCHAR(15),

    Region VARCHAR(15),

    PostalCode VARCHAR(10),

    Country VARCHAR(15),

    HomePhone VARCHAR(24),

    Extension VARCHAR(4),

    Photo VARCHAR(MAX),

    Notes VARCHAR(MAX),

    ReportsTo INT,

    PhotoPath VARCHAR(255)

    )

*-- EmployeeTerritories*

CREATE TABLE EmployeeTerritories(

    EmployeeID INT,

    TerritoryID VARCHAR(20)

    )

*-- OrderDetails*

CREATE TABLE OrderDetails(

    OrderID INT,

    ProductID INT,

    UnitPrice DECIMAL(25,2),

    Quantity INT,

    Discount DECIMAL(25,2)

    )

*-- Orders*

CREATE TABLE Orders(

    OrderID INT,

    CustomerID VARCHAR(5),

    EmployeeID INT,

    OrderDate DATETIME ,

    RequiredDate DATETIME ,

    ShippedDate DATETIME ,

    ShipVia INT,

    Freight DECIMAL(25,2),

    ShipName VARCHAR(40),

    ShipAddress VARCHAR(60),

    ShipCity VARCHAR(15),

    ShipRegion VARCHAR(15),

    ShipPostalCode VARCHAR(10),

    ShipCountry VARCHAR(15)

    )

*-- Products*

CREATE TABLE Products(

    ProductID INT,

    ProductName VARCHAR(40),

    SupplierID INT,

    CategoryID INT,

    QuantityPerUnit VARCHAR(20),

    UnitPrice DECIMAL(25,2),

    UnitsInStock INT,

    UnitsOnOrder INT,

    ReorderLevel INT,

    Discontinued INT

    )

*-- Region*

CREATE TABLE Region(

    RegionID INT,

    RegionDescription VARCHAR(50)

    )

*-- Shippers*

CREATE TABLE Shippers(

    ShipperID INT,

    CompanyName VARCHAR(40),

    Phone VARCHAR(24)

    )

*-- Suppliers*

CREATE TABLE Suppliers(

    SupplierID INT,

    CompanyName VARCHAR(40),

    ContactName VARCHAR(30),

    ContactTitle VARCHAR(30),

    Address VARCHAR(60),

    City VARCHAR(15),

    Region VARCHAR(15),

    PostalCode VARCHAR(10),

    Country VARCHAR(15),

    Phone VARCHAR(24),

    Fax VARCHAR(24),

    HomePage VARCHAR(MAX)

    )

*-- Territories*

CREATE TABLE Territories(

    TerritoryID VARCHAR(20),

    TerritoryDescription VARCHAR(50),

    RegionID INT

    )

### **DATAWAREHOUSE**

Finalmente, en SQL Server se desarrolló la creación de la Base de Datos de NorthwindDW mediante un esquema de estrella.

Diagrama

Descripción generada automáticamente

Dimensiones:

DIM\_EMPLOYEE\_TERRITORIES CON REGION

DIM\_EMPLOYEES SIN TERRITORY

DIM\_ORDERS CON SHIPPER Y PHONE

DIM\_CUSTOMERS

DIM\_PRODUCTS

DIM\_CATEGORIES

DIM\_SUPPLIERS

DIM\_TIME 1996-1998

FACT\_SALES CON FECHA (ORDER\_DATE), UNIT\_PRICE, QUANTITY, DISCOUNT

DROP TABLE DIM\_EMPLOYEE\_TERRITORIES;

DROP TABLE DIM\_EMPLOYEES;

DROP TABLE DIM\_ORDERS;

DROP TABLE DIM\_CUSTOMERS;

DROP TABLE DIM\_PRODUCTS;

DROP TABLE DIM\_CATEGORIES;

DROP TABLE DIM\_SUPPLIERS;

DROP TABLE DIM\_TIME;

DROP TABLE FACT\_SALES;

CREATE TABLE DIM\_TIME(

    ID\_TIME int IDENTITY NOT NULL *CONSTRAINT* PK\_TIME *PRIMARY KEY*,

    DATE date NULL,

    YEAR int NULL,

    MONTH int NULL,

    DAY int NULL,

    NDAY varchar(30) NULL,

    NMONTH varchar(20) NULL,

    DAY\_YEAR int NULL,

    PERIOD varchar(30) NULL

)

select \* from DIM\_TIME

delete from DIM\_TIME

DBCC CHECKIDENT (DIM\_TIME, RESEED, 0)

BEGIN

SET LANGUAGE English;

DECLARE @FECHA\_INICIO DATE;

DECLARE @FECHA\_FINAL DATE;

SET @FECHA\_INICIO='1996-01-01';

SET @FECHA\_FINAL ='1998-12-31';

DECLARE @PFECHA DATE;

DECLARE @PANNIO INT;

DECLARE @PMES INT;

DECLARE @PNMES VARCHAR(20);

DECLARE @PDIA  INT;

DECLARE @PDIA\_ANNIO INT  ;

DECLARE @PERIODO VARCHAR(30);

DECLARE @PNDIA VARCHAR(30);

WHILE @FECHA\_INICIO<=@FECHA\_FINAL

    BEGIN

        SET @PFECHA      = @FECHA\_INICIO;

        SET @PANNIO      = YEAR(@FECHA\_INICIO);

        SET @PMES        = MONTH(@FECHA\_INICIO);

        SET @PNMES       = DATENAME(MONTH, @FECHA\_INICIO);

        SET @PNDIA       = DATENAME(dw,@FECHA\_INICIO);

        SET @PDIA        = DAY(@FECHA\_INICIO);

        SET @PDIA\_ANNIO  = DATENAME(dayofyear, @FECHA\_INICIO) ;

        SET @PERIODO     =CONCAT( CASE WHEN @PMES BETWEEN 1 and 9  THEN

                            CONCAT('0',cast(@PMES as varchar(1))) ELSE

                                cast( @PMES as varchar(2)) END ,' - ',DATENAME(MONTH, @FECHA\_INICIO))

        INSERT INTO DIM\_TIME (DATE,

                                YEAR,

                                MONTH,

                                DAY,

                                NDAY,

                                NMONTH,

                                DAY\_YEAR,

                                PERIOD

                                )

                                VALUES

                                (@PFECHA ,

                                @PANNIO ,

                                @PMES ,

                                @PDIA,

                                upper(@PNDIA),

                                upper(@PNMES),

                                @PDIA\_ANNIO ,

                                upper(@PERIODO)

                                )

                SET @FECHA\_INICIO=DATEADD(DAY,1,@FECHA\_INICIO);

    END ;

END ;

CREATE TABLE DIM\_EMPLOYEE\_TERRITORIES(

    ID\_EMPLOYEE\_TERRITORY int IDENTITY NOT NULL *CONSTRAINT* PK\_EMPLOYEE\_TERRITORY *PRIMARY KEY*,

    TERRITORY\_ID nvarchar(10) NOT NULL,

    EMPLOYEE\_ID INT NOT NULL,

    TERRITORY nchar(50) NOT NULL,

    EMPLOYEE nvarchar(80) NOT NULL,

    REGION nchar(30) NOT NULL

)

CREATE TABLE DIM\_EMPLOYEES(

    ID\_EMPLOYEE int IDENTITY NOT NULL *CONSTRAINT* PK\_EMPLOYEE *PRIMARY KEY*,

    EMPLOYEE\_ID INT NOT NULL,

    EMPLOYEE\_NAME nvarchar(80) NOT NULL,

    TITLE nvarchar(40) NOT NULL,

    BIRTH\_DATE DATETIME NOT NULL,

    HIRE\_DATE DATETIME NOT NULL,

    CITY nvarchar(20) NOT NULL,

    REGION nvarchar(20) NOT NULL,

    COUNTRY nvarchar(20) NOT NULL,

    REPORTS\_TO nvarchar(60) NOT NULL

)

CREATE TABLE DIM\_ORDERS(

    ID\_ORDER int IDENTITY NOT NULL *CONSTRAINT* PK\_ORDER *PRIMARY KEY*,

    ORDER\_ID int NOT NULL,

    CUSTOMER nvarchar(80) NOT NULL,

    EMPLOYEE nvarchar(80) NOT NULL,

    ORDER\_DATE DATETIME NOT NULL,

    REQUIRED\_DATE DATETIME NOT NULL,

    SHIPPED\_DATE DATETIME NOT NULL,

    SHIPPER nvarchar(30) NOT NULL,

    FREIGHT money NOT NULL,

    SHIP\_CITY nvarchar(30) NOT NULL,

    SHIP\_REGION nvarchar(20) NOT NULL,

    SHIP\_COUNTRY nvarchar(30) NOT NULL

)

CREATE TABLE DIM\_CUSTOMERS(

    ID\_CUSTOMER int IDENTITY NOT NULL *CONSTRAINT* PK\_CUSTOMER *PRIMARY KEY*,

    CUSTOMER\_ID nchar(15) NOT NULL,

    COMPANY\_NAME nvarchar(80) NOT NULL,

    CONTACT\_NAME nvarchar(80) NOT NULL,

    CONTACT\_TITLE nvarchar(80) NOT NULL,

    CITY nvarchar(40) NOT NULL,

    REGION nvarchar(40) NOT NULL,

    COUNTRY nvarchar(40) NOT NULL,

    CATEGORY varchar(40) NOT NULL,

    ENTRY\_DATE DATETIME NOT NULL

)

CREATE TABLE DIM\_PRODUCTS(

    ID\_PRODUCT int IDENTITY NOT NULL *CONSTRAINT* PK\_PRODUCT *PRIMARY KEY*,

    PRODUCT\_ID INT NOT NULL,

    PRODUCT\_NAME nvarchar(80) NOT NULL,

    CATEGORY nvarchar(30) NOT NULL,

    SUPPLIER nvarchar(100) NOT NULL,

    REORDER\_LEVEL smallint NOT NULL

)

CREATE TABLE DIM\_CATEGORIES(

    ID\_CATEGORY int IDENTITY NOT NULL *CONSTRAINT* PK\_CATEGORY *PRIMARY KEY*,

    CATEGORY\_ID INT NOT NULL,

    CATEGORY\_NAME nvarchar(50) NOT NULL,

    DESCRIPTION nvarchar(MAX) NOT NULL

)

CREATE TABLE DIM\_SUPPLIERS(

    ID\_SUPPLIER int IDENTITY NOT NULL *CONSTRAINT* PK\_SUPPLIER *PRIMARY KEY*,

    SUPPLIER\_ID INT NOT NULL,

    COMPANY\_NAME nvarchar(80) NOT NULL,

    CONTACT\_NAME nvarchar(80) NOT NULL,

    CONTACT\_TITLE nvarchar(80) NOT NULL,

    CITY nvarchar(40) NOT NULL,

    REGION nvarchar(40) NOT NULL,

    COUNTRY nvarchar(40) NOT NULL

)

CREATE TABLE FACT\_SALES(

    ID\_SALE int IDENTITY NOT NULL *CONSTRAINT* PK\_SALE *PRIMARY KEY* CLUSTERED,

    ID\_EMPLOYEE int NOT NULL,

    ID\_ORDER int NOT NULL,

    ID\_CUSTOMER int NOT NULL,

    ID\_PRODUCT int NOT NULL,

    ID\_CATEGORY INT NOT NULL,

    ID\_SUPPLIER int NOT NULL,

    ID\_TIME INT NOT NULL,

    QUANTITY smallint NOT NULL,

    UNIT\_PRICE money NOT NULL,

    DISCOUNT real NOT NULL,

    ORDER\_DATE DATETIME NOT NULL

)

CREATE TABLE ORDER\_DETAILS (

    OrderID int NOT NULL,

    ProductID int NOT NULL,

    UnitPrice money NOT NULL,

    Quantity smallint NOT NULL,

    Discount real NOT NULL

)

ALTER TABLE FACT\_SALES  WITH *CHECK* ADD  *CONSTRAINT* FK\_SALES\_EMPLOYEES *FOREIGN KEY*(ID\_EMPLOYEE)

*REFERENCES* DIM\_EMPLOYEES (ID\_EMPLOYEE)

ALTER TABLE FACT\_SALES  WITH *CHECK* ADD  *CONSTRAINT* FK\_SALES\_ORDERS *FOREIGN KEY*(ID\_ORDER)

*REFERENCES* DIM\_ORDERS (ID\_ORDER)

ALTER TABLE FACT\_SALES  WITH *CHECK* ADD  *CONSTRAINT* FK\_SALES\_CUSTOMERS *FOREIGN KEY*(ID\_CUSTOMER)

*REFERENCES* DIM\_CUSTOMERS (ID\_CUSTOMER)

ALTER TABLE FACT\_SALES  WITH *CHECK* ADD  *CONSTRAINT* FK\_SALES\_PRODUCTS *FOREIGN KEY*(ID\_PRODUCT)

*REFERENCES* DIM\_PRODUCTS (ID\_PRODUCT)

ALTER TABLE FACT\_SALES  WITH *CHECK* ADD  *CONSTRAINT* FK\_SALES\_CATEGORIES *FOREIGN KEY*(ID\_CATEGORY)

*REFERENCES* DIM\_CATEGORIES (ID\_CATEGORY)

ALTER TABLE FACT\_SALES  WITH *CHECK* ADD  *CONSTRAINT* FK\_SALES\_SUPPLIERS *FOREIGN KEY*(ID\_SUPPLIER)

*REFERENCES* DIM\_SUPPLIERS (ID\_SUPPLIER)

ALTER TABLE FACT\_SALES  WITH *CHECK* ADD  *CONSTRAINT* FK\_SALES\_TIME *FOREIGN KEY*(ID\_TIME)

*REFERENCES* DIM\_TIME (ID\_TIME)

DIM\_EMPLOYEE\_TERRITORIES:

SELECT

UPPER(T.TERRITORYID) TERRITORY\_ID,

UPPER(E.EMPLOYEEID) EMPLOYEE\_ID,

UPPER(T.TERRITORYDESCRIPTION) TERRITORY\_DESCRIPTION,

UPPER(CONCAT(E.FIRSTNAME, ' ', E.LASTNAME)) EMPLOYEE,

UPPER(REGIONDESCRIPTION) REGION

FROM EMPLOYEES E

INNER JOIN EMPLOYEETERRITORIES ET

ON E.EMPLOYEEID = ET.EMPLOYEEID

INNER JOIN TERRITORIES T

ON T.TERRITORYID = ET.TERRITORYID

INNER JOIN REGION G

ON G.REGIONID = T.REGIONID

ORDER BY EMPLOYEE\_ID

DIM\_EMPLOYEES:

SELECT

UPPER(E.EMPLOYEEID) EMPLOYEE\_ID,

UPPER(CONCAT(E.FIRSTNAME, ' ', E.LASTNAME)) EMPLOYEE\_NAME,

UPPER(E.TITLE) TITLE,

E.BIRTHDATE BIRTH\_DATE,

E.HIREDATE HIRE\_DATE,

UPPER(E.CITY) CITY,

UPPER(isnull(E.REGION,'NO REGION')) REGION,

UPPER(E.COUNTRY) COUNTRY,

COALESCE(NULLIF(

UPPER(CONCAT(J.FIRSTNAME, ' ', J.LASTNAME)),''), 'NO BOSS') REPORTS\_TO

FROM EMPLOYEES E

LEFT JOIN EMPLOYEES J

ON E.REPORTSTO = J.EMPLOYEEID

ORDER BY EMPLOYEE\_ID

DIM\_ORDERS:

SELECT

UPPER(O.ORDERID) ORDER\_ID,

UPPER(C.COMPANYNAME) CUSTOMER,

UPPER(CONCAT(E.FIRSTNAME, ' ', E.LASTNAME)) EMPLOYEE,

O.ORDERDATE ORDER\_DATE,

O.REQUIREDDATE REQUIRED\_DATE,

isnull(O.SHIPPEDDATE,cast(getdate() as date)) SHIPPED\_DATE,

UPPER(S.COMPANYNAME) SHIPPER,

UPPER(O.FREIGHT) FREIGHT,

UPPER(O.SHIPCITY) SHIP\_CITY,

UPPER(isnull(O.SHIPREGION,'NO SHIP REGION')) SHIP\_REGION,

UPPER(O.SHIPCOUNTRY) SHIP\_COUNTRY

FROM ORDERS O

INNER JOIN EMPLOYEES E

ON E.EMPLOYEEID = O.EMPLOYEEID

INNER JOIN CUSTOMERS C

ON C.CUSTOMERID = O.CUSTOMERID

INNER JOIN SHIPPERS S

ON S.SHIPPERID = O.SHIPVIA

ORDER BY ORDER\_ID

DIM\_CUSTOMERS:

SELECT

UPPER(C.CUSTOMERID) CUSTOMER\_ID,

UPPER(C.COMPANYNAME) COMPANY\_NAME,

UPPER(C.CONTACTNAME) CONTACT\_NAME,

UPPER(C.CONTACTTITLE) CONTACT\_TITLE,

UPPER(C.CITY) CITY,

UPPER(isnull(C.REGION,'NO REGION')) REGION,

UPPER(C.COUNTRY) COUNTRY,

isnull(

case when cc.grupo=1 then

'A'

when cc.grupo=2 then

'B'

when cc.grupo=3 then

'C'

when cc.grupo=4 then

'D'

end ,'E' ) CATEGORY,

isnull(IC.ENTRY,cast(getdate() as date)) ENTRY\_DATE

FROM CUSTOMERS C

LEFT JOIN (

SELECT MIN(ORDERDATE) ENTRY, CUSTOMERID FROM ORDERS

GROUP BY CUSTOMERID

) IC

ON C.CUSTOMERID=IC.CUSTOMERID

left join (

 select CUSTOMERID, sum(QUANTITY\*UNITPRICE) SALES,

 NTILE(4) OVER(ORDER BY sum(QUANTITY\*UNITPRICE) DESC) AS grupo

 from ORDERS O

 inner join [Order Details] OD

 on O.ORDERID = OD.ORDERID

 where SHIPPEDDATE IS NOT NULL

 group by CUSTOMERID) CC

 on C.CUSTOMERID = CC.CUSTOMERID

ORDER BY CUSTOMER\_ID

DIM\_PRODUCTS:

SELECT

UPPER(P.PRODUCTID) PRODUCT\_ID,

UPPER(P.PRODUCTNAME) PRODUCT\_NAME,

UPPER(C.CATEGORYNAME) CATEGORY,

UPPER(S.COMPANYNAME) SUPPLIER,

UPPER(P.REORDERLEVEL) REORDER\_LEVEL

FROM PRODUCTS P

INNER JOIN SUPPLIERS S

ON S.SUPPLIERID = P.SUPPLIERID

INNER JOIN CATEGORIES C

ON C.CATEGORYID = P.CATEGORYID

ORDER BY PRODUCT\_ID

DIM\_CATEGORIES:

SELECT

UPPER(CATEGORYID) CATEGORY\_ID,

UPPER(CATEGORYNAME) CATEGORY\_NAME,

UPPER(DESCRIPTION) DESCRIPTION

FROM CATEGORIES

ORDER BY CATEGORY\_ID

DIM\_SUPPLIERS:

SELECT

UPPER(SUPPLIERID) SUPPLIER\_ID,

UPPER(COMPANYNAME) COMPANY\_NAME,

UPPER(CONTACTNAME) CONTACT\_NAME,

UPPER(CONTACTTITLE) CONTACT\_TITLE,

UPPER(CITY) CITY,

UPPER(isnull(REGION,'NO REGION')) REGION,

UPPER(COUNTRY) COUNTRY

FROM SUPPLIERS

ORDER BY SUPPLIER\_ID

FACT\_SALES:

SELECT

E.ID\_EMPLOYEE,

O.ID\_ORDER,

C.ID\_CUSTOMER,

PR.ID\_PRODUCT,

CA.ID\_CATEGORY,

S.ID\_SUPPLIER,

TI.ID\_TIME,

OD.QUANTITY,

OD.UNITPRICE UNIT\_PRICE,

OD.DISCOUNT,

O.ORDER\_DATE

FROM ORDER\_DETAILS OD

INNER JOIN DIM\_ORDERS O

ON OD.ORDERID = O.ORDER\_ID

INNER JOIN DIM\_PRODUCTS PR

ON OD.PRODUCTID = PR.PRODUCT\_ID

INNER JOIN DIM\_TIME TI

ON TI.DATE = O.ORDER\_DATE

INNER JOIN DIM\_CUSTOMERS C

ON C.COMPANY\_NAME = O.CUSTOMER

INNER JOIN DIM\_EMPLOYEES E

ON E.EMPLOYEE\_NAME = O.EMPLOYEE

INNER JOIN DIM\_CATEGORIES CA

ON CA.CATEGORY\_NAME = PR.CATEGORY

INNER JOIN DIM\_SUPPLIERS S

ON S.COMPANY\_NAME = PR.SUPPLIER

# **CONCLUSIÓN**

El desarrollo de un datawarehouse es un proceso técnico sumamente complejo pero que se ve apoyado de manera importante de un área de negocio mediante el establecimiento de los objetivos y la dirección de tal manera que se convierte en una relación bidireccional.

Para esto existen diversas técnicas, herramientas, tecnologías que pueden ser utilizadas y no existe una perfecta o única; por el contrario, estas deben ser elegidas cuidadosamente según las necesidades del proyecto o de la organización.

Durante la ejecución de este proceso, es imperativo realizar un análisis de la estructura que posee actualmente la organización con el fin de estandarizarla mediante técnicas de ETL para asegurar el cumplimiento de los objetivos.

# **BIBLIOGRAFÍA**

Fatima, N. (2022, 24 noviembre). *Proceso ETL y los pasos para su implementación*. Astera. Recuperado 12 de marzo de 2023, de <https://www.astera.com/es/type/blog/etl-process-and-steps/>

Alfaro, F. R. (2021, 21 junio). *Proceso ETL con SQL Server Integration Services - SSIS*. Data Management. Recuperado 12 de marzo de 2023, de https://datamanagement.es/2020/04/06/proceso-etl-con-sql-server-integration-services-ssis/

Mendoza A. (2022, 23 noviembre). *MetodologíasData Warehouse | Gravitar Business Intelligence*. Business Intelligence, Data Warehouse, Monterrey, México : Gravitar. Recuperado 12 de marzo de 2023, de https://gravitar.biz/datawarehouse/metodologias-data-warehouse/

Ramirez J. (2022, 10 agosto). Recuperado 12 de marzo de 2023, de https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/qu-son-los-procesos-etl