UNIVERSIDAD GALILEO

Postgrado en Análisis y Predicción de Datos

Curso: Diseño y Construcción de Data Warehouses Horario: Lun: 18:00 – 21:00

Tutor: Msc. José Rolando Lucero Morataya

**Diseño y gestión de Data Warahouses**

**Semana 4**

Nombre: Andrea María Hernández Marroquín

Carne: 240110074

Guatemala, 24 de marzo de 2024

# **INDICE**

Contenido

[**INDICE** 2](#_Toc162183035)

[**INTRODUCCION:** 3](#_Toc162183036)

[**DISEÑO DEL DATA WAREHOUSE** 4](#_Toc162183037)

[**DISEÑO DIMENSIONAL** 6](#_Toc162183038)

[**SQL** 7](#_Toc162183039)

[**CONCLUSIONES** 12](#_Toc162183040)

# **INTRODUCCION:**

El diseño dimensional en un Data Warehouse (DW) es un enfoque de modelado que se utiliza para organizar y estructurar los datos de manera que sean óptimos para el análisis y la generación de informes. A diferencia de los modelos relacionales tradicionales, que están diseñados para transacciones y operaciones de base de datos, el diseño dimensional se centra en facilitar el análisis de datos mediante la creación de estructuras fáciles de entender y consultar.

En un diseño dimensional, los datos se organizan en dos tipos principales de tablas: tablas de hechos y tablas de dimensiones. Las tablas de hechos contienen métricas o medidas cuantitativas que se están analizando, mientras que las tablas de dimensiones contienen descripciones o atributos que proporcionan contexto a las métricas. Este enfoque permite a los usuarios realizar consultas complejas y analizar datos desde diferentes perspectivas.

La presente tarea se presenta un detallado diagrama dimensional para una empresa que busca analizar información de un supermercado para poder tomar decisiones.

## **DISEÑO DEL DATA WAREHOUSE**

Para realizar el diseño dimensional para un Supermercado se propone lo siguiente la creación de las siguientes tablas:

1. Tabla employee\_dimension

Se decidió reemplazar la tabla de cashier\_dimension por la de employee\_dimension en esta tabla se establecen campos como el tipo de empleado y la tienda a la que pertenece para poder identificar de mejor forma al empleado que despacho en caja.

1. Tabla store\_dimension

En el diseño original se consideró colocar en esta sola tabla las características de la tienda, sin embargo, por la cantidad de tiendas con las que cuenta el supermercado y la cantidad de estados en las que se encuentran distribuidas se consideró necesario realizar un catálogo de distritos y otro catálogo de regiones.

1. Tabla product\_dimension

En el diseño original se consideró realizar en una sola tabla las características del producto, sin embargo, se agrego la tabla supplier\_dimension, category\_dimension y brand\_catalog ya que en el diseño original no se consideró la realización de estas tablas sin embargo son importantes ya que identifican las diferentes categorías de los productos, el proveedor del producto y la marca

1. Tabla payment\_method\_dimension
2. Tabla promotion\_dimention

En el diseño original solo se estableció que en una tabla se almacenara todos los tipos de promoción, sin embargo, se consideró necesario la realización de la tabla promotion\_media\_type en la que se almacenen las diferentes categorías o medios de promoción. Así también en la tabla promotion\_dimention se le agrego el campo promotion\_discount en el que se almacena el descuento que se realizará en la promoción.

1. Tabla payment\_method\_dimension
2. Tabla rental\_sales\_facts se estableció como la tabla de hechos
3. En la tabla fact\_sales\_snapshot se almacenarán las ventas por facturas por lo que el diseño dimensional quedo de la siguiente manera:

## **DISEÑO DIMENSIONAL**

**Captura de pantalla de un celular

Descripción generada automáticamente**

## **SQL**

CREATE DATABASE IF NOT EXISTS retail\_sales;

USE retail\_sales;

CREATE TABLE employee\_dimension (

employee\_id INT PRIMARY KEY,

store\_key INT,

employee\_name VARCHAR(100),

employee\_direction VARCHAR(255),

employee\_telephone\_area INT,

employee\_telephone INT, -- Cambiado a VARCHAR(15)

FOREIGN KEY (store\_key) REFERENCES store\_dimension(store\_key)

);

CREATE TABLE district\_catalog (

district\_id INT PRIMARY KEY,

district\_name VARCHAR(50) NOT NULL

);

CREATE TABLE region\_catalog (

region\_id INT PRIMARY KEY,

region\_name VARCHAR(50) NOT NULL

);

CREATE TABLE store\_dimension (

store\_key INT PRIMARY KEY,

store\_number INT NOT NULL,

store\_name VARCHAR(255) NOT NULL,

district\_id INT,

region\_id INT,

store\_address VARCHAR(255),

FOREIGN KEY (district\_id) REFERENCES district\_catalog(district\_id),

FOREIGN KEY (region\_id) REFERENCES region\_catalog(region\_id)

);

CREATE TABLE promotion\_dimention(

promotion\_key INT AUTO\_INCREMENT PRIMARY KEY,

promotion\_code VARCHAR(20) NOT NULL,

promotion\_name VARCHAR(20) NOT NULL,

promotion\_media\_type\_id VARCHAR(100) NOT NULL,

promotion\_begin\_date DATETIME NOT NULL,

promotion\_discount DECIMAL(10, 2)

UNIQUE(promotion\_key)

);

CREATE TABLE promotion\_media\_type(

promotion\_media\_type\_id INT AUTO\_INCREMENT PRIMARY KEY,

media\_type\_description varchar NOT NULL

);

CREATE TABLE payment\_method\_dimension(

payment\_method\_key INT AUTO\_INCREMENT PRIMARY KEY,

payment\_method\_description VARCHAR(100) NOT NULL,

payment\_method\_group VARCHAR(100) NOT NULL,

UNIQUE(payment\_method\_key)

);

CREATE TABLE brand\_dimension (

brand\_id INT AUTO\_INCREMENT PRIMARY KEY,

brand\_description VARCHAR(100) NOT NULL,

UNIQUE(brand\_description)

);

CREATE TABLE category\_dimension (

category\_id INT AUTO\_INCREMENT PRIMARY KEY,

category\_description VARCHAR(100) NOT NULL,

UNIQUE(category\_description)

);

CREATE TABLE supplier\_dimension (

supplier\_id INT AUTO\_INCREMENT PRIMARY KEY,

supplier\_name VARCHAR(100) NOT NULL,

supplier\_address VARCHAR(255),

supplier\_phone VARCHAR(20),

UNIQUE(supplier\_name)

);

CREATE TABLE product\_dimension (

product\_key INT AUTO\_INCREMENT PRIMARY KEY,

sku\_number VARCHAR(20) NOT NULL,

product\_description VARCHAR(100) NOT NULL,

brand\_id INT NOT NULL,

category\_id INT NOT NULL,

supplier\_id INT,

FOREIGN KEY (brand\_id) REFERENCES brand\_dimension(brand\_id),

FOREIGN KEY (category\_id) REFERENCES category\_dimension(category\_id),

FOREIGN KEY (supplier\_id) REFERENCES supplier\_dimension(supplier\_id)

);

CREATE TABLE rental\_sales\_facts (

sales\_id INT AUTO\_INCREMENT PRIMARY KEY,

product\_key INT,

store\_key INT,

promotion\_key INT,

employee\_key INT,

payment\_method\_key INT,

sales\_quantity INT,

regular\_unit\_price DECIMAL(10, 2),

discount\_unit\_price DECIMAL(10, 2),

net\_unit\_price DECIMAL(10, 2),

extended\_discount\_dollar\_amount DECIMAL(10, 2),

extended\_sales\_dollar\_amount DECIMAL(10, 2),

extended\_cost\_dollar\_amount DECIMAL(10, 2),

extended\_gross\_profit\_dollar\_amount DECIMAL(10, 2),

FOREIGN KEY (product\_key) REFERENCES product\_dimension(product\_key),

FOREIGN KEY (store\_key) REFERENCES store\_dimension(store\_key),

FOREIGN KEY (promotion\_key) REFERENCES promotion\_dimention(promotion\_key),

FOREIGN KEY (employee\_id) REFERENCES employee\_dimension(employee\_id)

);

# **CONCLUSIONES**

Con el diseño del Data Warehouse anterior se desea convertir los datos en información valiosa y accionable para la toma de decisiones empresariales. Al proporcionar una estructura clara y organizada para los datos, se facilita el análisis, se mejora el rendimiento y se impulsa la colaboración en toda la organización.