# **UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN**

Facultad Ciencias Físico Matemáticas

Base de Datos Multimedia

**2º Avance**

##### **Profesor: M.D. JUAN ALEJANDRO VILLARREAL MOJICA**

##### **Equipo:**

##### **° Alexis Urdiales Garza 1604065 º María Paula Rodríguez Escamilla 1691594 Gpo:01**

San Nicolás de los Garza, Nuevo León, México A 20 de Octubre de 2022

Contenido

[**UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN** 1](#_Toc117202104)

[Diagrama de MySQL 3](#_Toc117202105)

[Diccionario de datos 4](#_Toc117202106)

[Query para creacion de diccionario de datos 6](#_Toc117202107)

[Query para creacion de tablas y base de datos 11](#_Toc117202108)

# Diagrama de MySQL

# Diccionario de datos





# Query para creacion de diccionario de datos

------------------------------------------------------------------------------------

-- Data Dictionary Dump:

-- This SQL script will dump table, column, key, and description design related

-- metadata so that you can copy-paste or export to Excel as a Data Dictionary.

------------------------------------------------------------------------------------

-- Platform: MySQL

-- Author: DataResearchLabs

-- GitHub: https://github.com/DataResearchLabs/sql\_scripts

-- YouTube Tutorials: https://www.youtube.com/channel/UCQciXv3xaBykeUFc04GxSXA

----------------------------------------------------------------------------------

USE bdmm\_db2 -- <<<<<<<<<<<<< Change schema here

;

WITH vars

AS (

SELECT

DATABASE() AS v\_SchemaName -- Do not change this value...it is changed up above on line 11 and passed thru to here

, 'NO' AS v\_TablesOnly -- YES=Limit To Tables only; NO=Include views too

)

, baseTbl

AS (

SELECT table\_schema AS SchemaName

, table\_catalog

, table\_type, table\_name, table\_schema

FROM INFORMATION\_SCHEMA.TABLES

WHERE TABLE\_SCHEMA = (SELECT v\_SchemaName FROM vars)

AND ( (TABLE\_TYPE = 'BASE TABLE')

OR ((SELECT v\_TablesOnly FROM vars) = 'NO')

)

)

, metadata

AS (

SELECT

bt.SchemaName AS schema\_nm

, bt.table\_name AS table\_nm

, CASE WHEN bt.TABLE\_TYPE = 'BASE TABLE' THEN 'TBL'

WHEN bt.TABLE\_TYPE = 'VIEW' THEN 'VW'

ELSE 'UK'

END AS obj\_typ

, tut.ordinal\_position AS ord\_pos

, tut.column\_name AS column\_nm

, CONCAT(COALESCE(tut.data\_type, 'unknown'),

CASE WHEN tut.data\_type IN('varchar','char') THEN CONCAT('(', tut.CHARACTER\_MAXIMUM\_LENGTH, ')')

WHEN tut.data\_type IN('date','time') THEN CONCAT('(3)')

WHEN tut.data\_type = 'datetime' THEN CONCAT('(8)')

WHEN tut.data\_type = 'timestamp' THEN CONCAT('(4)')

WHEN tut.data\_type in('tinyint','smallint','mediumint','int','bigint') THEN CONCAT('(', tut.NUMERIC\_PRECISION, ')')

WHEN tut.data\_type = 'decimal' THEN CONCAT('(', tut.NUMERIC\_PRECISION, ',', tut.NUMERIC\_SCALE, ')')

WHEN tut.CHARACTER\_MAXIMUM\_LENGTH IS NOT NULL THEN CONCAT('(', tut.CHARACTER\_MAXIMUM\_LENGTH, ')')

WHEN tut.DATETIME\_PRECISION IS NOT NULL THEN CONCAT('(', tut.DATETIME\_PRECISION, ')')

WHEN tut.NUMERIC\_PRECISION IS NOT NULL

AND tut.NUMERIC\_SCALE IS NULL THEN CONCAT('(', tut.NUMERIC\_PRECISION, ')')

WHEN tut.NUMERIC\_PRECISION IS NOT NULL

AND tut.NUMERIC\_SCALE IS NOT NULL THEN CONCAT('(', tut.NUMERIC\_PRECISION, ',', tut.NUMERIC\_SCALE, ')')

ELSE ''

END ) AS data\_typ

, CASE WHEN tut.IS\_NULLABLE = 'YES' THEN 'NULL' ELSE 'NOT NULL' END AS nullable

FROM INFORMATION\_SCHEMA.COLUMNS tut

INNER JOIN baseTbl bt ON bt.table\_catalog = tut.TABLE\_CATALOG AND bt.table\_name = tut.table\_name

)

, meta\_for\_keys

AS (

SELECT schema\_nm, table\_nm, column\_nm

, GROUP\_CONCAT(is\_key ORDER BY is\_key SEPARATOR ',') AS is\_key

FROM (

SELECT cons.TABLE\_SCHEMA AS schema\_nm

, cons.TABLE\_NAME AS table\_nm

, kcu.COLUMN\_NAME AS column\_nm

, CASE WHEN cons.constraint\_type = 'PRIMARY KEY' THEN 'PK'

WHEN cons.constraint\_type = 'UNIQUE' THEN 'UK'

WHEN cons.constraint\_type = 'FOREIGN KEY' THEN 'FK'

ELSE 'X'

END AS is\_key

FROM INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS cons

INNER JOIN INFORMATION\_SCHEMA.KEY\_COLUMN\_USAGE kcu

ON cons.TABLE\_SCHEMA = kcu.TABLE\_SCHEMA

AND cons.TABLE\_NAME = kcu.TABLE\_NAME

AND cons.CONSTRAINT\_NAME = kcu.CONSTRAINT\_NAME

WHERE cons.table\_schema = (SELECT v\_SchemaName FROM vars)

AND cons.table\_name IN(SELECT DISTINCT table\_name FROM baseTbl)

AND cons.constraint\_type IN('PRIMARY KEY','FOREIGN KEY','UNIQUE')

GROUP BY cons.TABLE\_SCHEMA, cons.TABLE\_NAME, kcu.COLUMN\_NAME, cons.constraint\_type

) t

GROUP BY schema\_nm, table\_nm, column\_nm

)

, col\_comm

AS (

SELECT TABLE\_SCHEMA AS SCHEMA\_NM

, TABLE\_NAME AS TABLE\_NM

, COLUMN\_NAME AS COLUMN\_NM

, COLUMN\_COMMENT AS column\_descr

FROM INFORMATION\_SCHEMA.COLUMNS

WHERE table\_schema = (SELECT v\_SchemaName FROM vars)

AND table\_name IN(SELECT DISTINCT table\_name FROM baseTbl)

)

SELECT md.SCHEMA\_NM, md.TABLE\_NM, md.OBJ\_TYP

, md.ORD\_POS AS ord

, COALESCE(pk.is\_key, ' ') AS is\_key

, md.COLUMN\_NM, md.DATA\_TYP, md.NULLABLE, c.column\_descr

FROM metadata md

LEFT JOIN meta\_for\_keys pk ON pk.SCHEMA\_NM = md.SCHEMA\_NM AND pk.TABLE\_NM = md.TABLE\_NM AND pk.COLUMN\_NM = md.COLUMN\_NM

LEFT JOIN col\_comm c ON c.SCHEMA\_NM = md.SCHEMA\_NM AND c.TABLE\_NM = md.TABLE\_NM AND c.COLUMN\_NM = md.COLUMN\_NM

ORDER BY md.SCHEMA\_NM, md.TABLE\_NM, md.ORD\_POS

# Query para creacion de tablas y base de datos

CREATE DATABASE BDMM\_DB2;

USE BDMM\_DB2;

CREATE TABLE usuarios (

Usuario\_Id INT UNIQUE AUTO\_INCREMENT COMMENT 'Identificacion numerica para los usuarios de la tienda online',

Usuario\_NombreUsuario VARCHAR(255) COMMENT "Nombre de Usuario",

Usuario\_Rol VARCHAR(255) COMMENT "Rol del usuario, este puede ser de Administrador, Vendedor y Comprador. Tambien se puede que sean Vendedor y Comprador al mismo tiempo",

Usuario\_Email VARCHAR(255) COMMENT "Email del usuario",

Usuario\_Contraseña VARCHAR(255) COMMENT "Contraseña del usuario. Esta debe ser min. de 8 caracteres y con al menos una mayuscula, minuscula, numero y caracter especial",

Usuario\_Nombre VARCHAR(255) COMMENT "Nombre(s) del usuario",

Usuario\_ApellidoP VARCHAR(255) COMMENT "Apellido Paterno del usuario",

Usuario\_ApellidoM VARCHAR(255) COMMENT "Apellido Materno del usuario",

Usuario\_FechaNacimiento DATE COMMENT "Fecha de nacimiento del usuario",

Usuario\_Genero bit NOT NULL COMMENT "Genero del usuario. Se puede escoger de la lista.",

Usuario\_Imagen BLOB COMMENT "Imagen de perfil del usuario",

Usuario\_FechaCreacion DATE COMMENT "Fecha en el que se creo el usuario",

PRIMARY KEY (Usuario\_Id)

);

CREATE TABLE Producto (

Producto\_Id INT UNIQUE AUTO\_INCREMENT COMMENT "El numero de registro para los productos",

Producto\_Usuario\_id int COMMENT "El numero de registro para el usuario",

Producto\_Nombre VARCHAR(255) COMMENT "Nombre del producto disponible",

Producto\_Descripcion VARCHAR(255) COMMENT "Informacion general o tecnica del producto",

Producto\_Precio FLOAT4 COMMENT "Precio del producto",

Producto\_Imagen BLOB COMMENT "Imagen(es) del producto",

Producto\_Video VARCHAR(255) COMMENT "Video para mostrar el producto",

Producto\_CategoriaId INT COMMENT "Numero de registro para las categorias",

Producto\_PalabraClave1 VARCHAR(255) COMMENT "Palabra clave para busqueda del producto #1",

Producto\_PalabraClave2 VARCHAR(255) COMMENT "Palabra clave para busqueda del producto #2",

Producto\_PalabraClave3 VARCHAR(255) COMMENT "Palabra clave para busqueda del producto #3",

Producto\_Subasta BOOL COMMENT "Indica si el producto esta disponible para subasta",

Producto\_Cantidad INT COMMENT "Cantidad de producto disponible",

Producto\_Valoracion INT COMMENT "Calificacion promedio del producto. Este se calcula por medio de todo sumar los comentarios.",

Producto\_Comentarios VARCHAR(255) COMMENT "Comentarios acerca del producto",

PRIMARY KEY (Producto\_Id),

FOREIGN KEY (Producto\_CategoriaId)

REFERENCES Categoria (Categoria\_Id),

FOREIGN KEY (Producto\_Usuario\_id)

REFERENCES usuarios (Usuario\_id)

);

CREATE TABLE MetodoPago (

Metodo\_Id INT UNIQUE AUTO\_INCREMENT COMMENT "Numero de registro para los metodos de pago",

Metodo\_Usuario INT COMMENT "Numero de registro de los usuarios",

Metodo\_tipo int NOT NULL COMMENT "Que tipo de pago seria el que se registrara para el usuario. 1.Tarjeta de Credito, 2.Tarjeta de Debito, 3.PayPal",

Metodo\_numerotarjeta int COMMENT "Numero de la tarjeta(si disponible) del usuario",

Metodo\_fechatarjeta date COMMENT "Fecha en el que se usa la tarjeta",

Metodo\_cvvtarjeta smallint COMMENT "Numero CVV de la tarjeta",

Metodo\_Nombre VARCHAR(255) COMMENT "Nombre costumatizable para el metodo de pago",

PRIMARY KEY (Metodo\_Id),

FOREIGN KEY (Metodo\_Usuario)

REFERENCES usuarios (Usuario\_Id)

);

CREATE TABLE Lista (

Lista\_Id int UNIQUE AUTO\_INCREMENT COMMENT "Numero de registro para las listas",

Lista\_Usuario int NOT NULL COMMENT "Numero de registro para los usuarios",

Lista\_Nombre VARCHAR (255) COMMENT "Nombre de la lista",

Lista\_Descripcion VARCHAR(255) COMMENT "Descripcion acerca de la lista",

Lista\_Imagen VARCHAR(255) COMMENT "Imagen adjuntada para la lista",

PRIMARY KEY (Lista\_Nombre),

FOREIGN KEY (Lista\_Usuario)

REFERENCES usuarios (Usuario\_Id)

);

CREATE TABLE Categoria (

Categoria\_Id INT UNIQUE AUTO\_INCREMENT COMMENT "Numero de registro para las categorias",

Categoria\_Nombre VARCHAR(255) COMMENT "Nombre de la categoria",

Categoria\_Descripcion VARCHAR(255) COMMENT "Descripcion de la categoria",

PRIMARY KEY (Categoria\_Id)

);

CREATE TABLE Cotizacion (

Cotizacion\_Id INT UNIQUE AUTO\_INCREMENT COMMENT "Numero de registro para las cotizaciones hechas",

Cotizacion\_Precio FLOAT4 COMMENT "Precio establecido por parte del vendedor para ofrecer al comprador",

Cotizacion\_Usuario int NOT NULL COMMENT "Numero de registro para los usuarios",

Cotizacion\_Producto int NOT NULL COMMENT "Numero de registro para los productos",

PRIMARY KEY (Cotizacion\_Id),

FOREIGN KEY (Cotizacion\_Producto)

REFERENCES Producto (Producto\_Id),

FOREIGN KEY (Cotizacion\_Usuario)

REFERENCES usuarios (Usuario\_Id)

);

CREATE TABLE Carrito (

Carrito\_Id INT UNIQUE AUTO\_INCREMENT COMMENT "Numero de registro para el carrito del comprador",

Carrito\_Usuario INT NOT NULL COMMENT "Numero de registro para los usuarios",

Carrito\_Subtotal FLOAT4 NOT NULL COMMENT "Resultado subtotal para los productos en el carrito",

Carrito\_IVA FLOAT4 NOT NULL COMMENT "Resultado estatico para el IVA",

Carrito\_Total FLOAT4 NOT NULL COMMENT "Resultado total para los productos en el carrito",

PRIMARY KEY (Carrito\_Id),

FOREIGN KEY (Carrito\_Usuario)

REFERENCES usuarios (Usuario\_Id)

);

CREATE TABLE Venta (

Venta\_Id INT UNIQUE AUTO\_INCREMENT COMMENT "Numero de registro para las Ventas realizadas",

Venta\_Producto INT COMMENT "Numero de registro de los productos",

PRIMARY KEY (Venta\_Id),

FOREIGN KEY (Venta\_Producto)

REFERENCES Producto (Producto\_Id)

);

CREATE TABLE Recibo (

Recibo\_Id INT UNIQUE AUTO\_INCREMENT COMMENT "Numero de registro para los recibos realizados",

Recibo\_CarritoId INT COMMENT "Numero de registro para el carrito del comprador",

Recibo\_UsuarioId INT COMMENT "Numero de registro para los usuario registrados",

Recibo\_Fecha DATE COMMENT "Fecha en el que se crea el recibo",

Recibo\_Subtotal FLOAT4 COMMENT "Subtotal del carrito mostrado en el recibo",

Recibo\_IVA FLOAT4 COMMENT "Resultado estatico para el IVA",

Recibo\_Total FLOAT4 COMMENT "Total del carrito mostrado en el recibo",

PRIMARY KEY (Recibo\_Id),

FOREIGN KEY (Recibo\_UsuarioId)

REFERENCES usuarios (Usuario\_Id),

FOREIGN KEY (Recibo\_CarritoId)

REFERENCES Carrito (Carrito\_Id)

);

CREATE TABLE Carrito\_Producto(

CarritoP\_CarritoId int COMMENT "Numero de registro para el carrito del cliente",

CarritoP\_ProductoId int COMMENT "Numero de registro para el producto",

CarritoP\_Cantidad int COMMENT "Cantidad de productos puestos en el carrito",

FOREIGN KEY (CarritoP\_CarritoId)

REFERENCES Carrito (Carrito\_Id),

FOREIGN KEY (CarritoP\_ProductoId)

REFERENCES Producto (Producto\_Id)

);

CREATE TABLE Venta\_Producto(

VentaP\_VentaId int COMMENT "Numero de registro para las ventas realizadas",

VentaP\_ProductoId int COMMENT "Numero de registro para los productos",

VentaP\_Cantidad int COMMENT "Cantidad de productos vendidos",

VentaP\_Subtotal float4 NOT NULL COMMENT "Subtotal de la venta del producto",

VentaP\_IVA FLOAT4 NOT NULL COMMENT "Resultado estatico para el IVA",

VentaP\_Total float4 NOT NULL COMMENT "Total de la venta del producto",

FOREIGN KEY (VentaP\_VentaId)

REFERENCES Venta (Venta\_Id),

FOREIGN KEY (VentaP\_ProductoId)

REFERENCES Producto (Producto\_Id)

);

CREATE TABLE Lista\_Producto(

ListaP\_ListaId int COMMENT "Numero de registro de las listas creadas",

ListaP\_ProductoId int COMMENT "Numero de registro para los productos",

FOREIGN KEY (ListaP\_ListaId)

REFERENCES Lista (Lista\_Id),

FOREIGN KEY (ListaP\_ProductoId)

REFERENCES Producto (Producto\_Id)

);

CREATE TABLE Comentario\_Producto(

ComentarioP\_Id int UNIQUE AUTO\_INCREMENT COMMENT "Numero de Registro para los comentarios hechos en el sitio web",

ComentarioP\_UsuarioId INT COMMENT "Numero de registro para los usuarios registrados",

ComentarioP\_ProductoId INT COMMENT "Numero de registro para los productos",

ComentarioP\_texto VARCHAR(255),

PRIMARY KEY (ComentarioP\_Id),

FOREIGN KEY (ComentarioP\_UsuarioId)

REFERENCES usuarios (Usuario\_Id),

FOREIGN KEY (ComentarioP\_ProductoId)

REFERENCES Producto (Producto\_Id)

);

CREATE TABLE Valoracion\_Producto(

Valoracion\_Id int UNIQUE AUTO\_INCREMENT COMMENT "Numero de registro para las valoraciones hechas a los productos",

Valoracion\_rate int COMMENT "Calificacion dada a los productos por medio de estrellas. Rango de 1 a 5",

Valoracion\_ProductoId int COMMENT "Numero de registro para los productos",

Valoracion\_UsuarioId int COMMENT "Numero de registro para los usuarios registrados",

PRIMARY KEY (Valoracion\_Id),

FOREIGN KEY (Valoracion\_ProductoId)

REFERENCES Producto (Producto\_Id),

FOREIGN KEY (Valoracion\_UsuarioId)

REFERENCES usuarios (Usuario\_Id)

);