

# BASE DE DATOS

## *ARCOR MARKET*

AUTOR: MIKHAIL ACEVEDO

Documentación del Desarrollo de la base de datos

ARCOR MARKET” creada como ejemplo para la clase de SQL

De Coderhouse.

## INDICE

Introducción.....	3
Resumen.....	3
Objetivo.....	3
Modelo de negocio.....	3
Creación de la base de datos ARCOR.....	4
Insert de datos en la DB.....	4
Apartado de Vistas.....	4
Apartado de Stored Procedures.....	5
Apartado de Funciones.....	5
Apartado de Triggers.....	6
Diagrama Entidad-Relación (Conceptual).....	6
Diagrama Entidad-Relación (Esquemático).....	7
Tablas Bitacora.....	7
Lista de tablas.....	8
Inteligencia empresarial.....	11
Tecnologías utilizadas.....	11

## **INTRODUCCIÓN**

En el mundo actual, diversos modelos de negocios, como supermercados o markets, generan y acumulan una gran cantidad de datos que, adecuadamente estructurados, ofrecen la oportunidad de realizar análisis y tomar decisiones fundamentadas para mejorar su funcionamiento. En este documento, se presenta detalladamente el proceso de creación de una base de datos para un supermercado imaginario, con el objetivo de proporcionar a sus dueños herramientas que les permitan tomar decisiones más informadas en cuanto a operaciones, inventario y gestión de recursos.

## **RESUMEN**

En este documento se presenta el modelo de base de datos para un mercado, donde se describen el objetivo del proyecto y los aspectos técnicos relacionados. Se incluyen los Diagramas de Entidad-Relación que representan la estructura de las tablas utilizadas en la base de datos. Este modelo de base de datos proporciona una solución integral para gestionar eficientemente las operaciones del mercado y facilitar el manejo de la información relacionada con clientes, empleados, proveedores, productos, pedidos y otros aspectos cruciales para el funcionamiento del negocio. Con este modelo, se busca optimizar la organización y el acceso a los datos, garantizando la integridad y consistencia de la información almacenada en la base de datos.

## **OBJETIVO**

El objetivo de este proyecto es diseñar y crear una base de datos eficiente y estructurada para un supermercado, que permita recopilar, organizar y gestionar de manera óptima la información relacionada con clientes, empleados, proveedores, productos y pedidos. Con esta base de datos, buscamos proporcionar herramientas que ayuden a los dueños y administradores del supermercado a tomar decisiones informadas y estratégicas para mejorar las operaciones diarias, optimizar el inventario, y ofrecer un mejor servicio a sus clientes.

## **MODELO DE NEGOCIO**

El modelo de negocio de nuestro supermercado se basa en una base de datos bien estructurada y eficiente que recopila y gestiona datos cruciales. Con un enfoque centrado en el cliente, optimizamos operaciones, inventario y recursos para brindar un servicio excepcional. Con información organizada sobre clientes, empleados, proveedores y productos, tomamos decisiones informadas para mejorar continuamente nuestro negocio y garantizar una experiencia de compra satisfactoria.

## CREACION Y DISEÑO DE LA BASE DE DATOS EMPRESARIAL

Esta colección de comandos SQL representa el diseño de una base de datos empresarial integral. Esta base de datos está diseñada para administrar y organizar información crucial relacionada con la gestión de clientes, empleados, productos, órdenes de venta, ventas, devoluciones y sueldos en el contexto de una empresa o negocio.

### INSERT EN LA DB

Se realizaron inserciones de datos en las tablas para clientes, empleados, productos, órdenes de venta, ventas y devoluciones. Los comandos INSERT INTO se utilizaron para agregar registros con información relevante en cada una de estas tablas, llenando así la base de datos con datos.

### APARTADO DE VISTAS

#### Primera vista: Productos por Categoría #1

Este código crea una vista llamada "vista\_productos\_por\_categoria" que muestra los nombres de los productos y sus categorías correspondientes. Utiliza una consulta SQL para combinar la información de las tablas "productos" y "categorias" a través de una operación JOIN.

#### Segunda vista: Clientes en Venezuela #2

Este código crea una vista llamada "vista\_clientes\_venezuela" que contiene todos los registros de la tabla "clientes" donde el campo "pais" es igual a 'Venezuela'. Es una forma de filtrar y ver solo los clientes de Venezuela.

#### Tercera vista: Detalles de Órdenes con Nombres de Productos #3

Esta vista, llamada "vista\_detalle\_ordenes\_productos", muestra los identificadores de órdenes y los nombres de productos correspondientes. Se basa en una operación JOIN entre las tablas "detalle\_ordenes" y "productos".

#### Cuarta vista Clientes con Órdenes #4

El código crea una vista llamada "vista\_clientes\_con\_ordenes" que lista los clientes con sus nombres y la cantidad de órdenes que han realizado. Utiliza una subconsulta para contar las órdenes asociadas a cada cliente.

### **Quinta vista: Proveedores con Más Demanda #5**

Esta vista, denominada "vista\_proveedores\_con\_mas\_demanda", muestra los cinco proveedores con la mayor demanda de productos. Realiza una consulta compleja que implica la tabla "proveedores" y "detalles\_ordenes" para contar cuántos productos de cada proveedor se han ordenado y luego los ordena en orden descendente.

## **APARTADO DE STORED PROCEDURES**

### **Procedimiento cantidad\_x\_letra**

Este procedimiento toma una letra como entrada y devuelve la cantidad de clientes cuyos nombres contienen esa letra. Utiliza una consulta SQL con un patrón LIKE en la tabla "clientes".

### **Procedimiento OrdenarTabla**

Este procedimiento permite ordenar una tabla específica en función de un campo dado en orden ascendente o descendente. La consulta dinámica se construye usando los parámetros proporcionados y se ejecuta para ordenar la tabla.

### **Procedimiento actualizacion\_preio**

Este procedimiento toma un ID de categoría como entrada y actualiza los precios de los productos en esa categoría. Calcula el nuevo precio multiplicando el precio actual por 1.05 (aumento del 5%) y luego actualiza los registros en la tabla "productos".

## **APARTADO DE FUNCIONES**

### **Función iva\_21**

Esta función calcula el 21% de IVA a partir de un monto dado y devuelve el resultado.

### **Función precio\_netio**

La función "precio\_netio" calcula el precio neto a partir del precio con IVA. Divide el monto dado por 1.21 (para eliminar el IVA) y devuelve el resultado.

### **Función paises\_clientes**

Esta función encuentra el país con la mayor cantidad de clientes y lo devuelve como resultado. Utiliza una subconsulta para contar los clientes en cada país y luego selecciona el país con la mayor cantidad.

### **Función edad\_empleados**

La función "edad\_empleados" calcula la edad de un empleado en función de su fecha de nacimiento. Utiliza la función TIMESTAMPDIFF para calcular la diferencia en años entre la fecha de nacimiento y la fecha actual.

## APARTADO DE TRIGGERS

### Trigger log\_proveedores

Este trigger se dispara después de insertar un registro en la tabla "proveedores". Registra los detalles de la inserción en la tabla "proveedores\_log", incluyendo la fecha y hora, el nombre de usuario, y el ID del proveedor insertado.

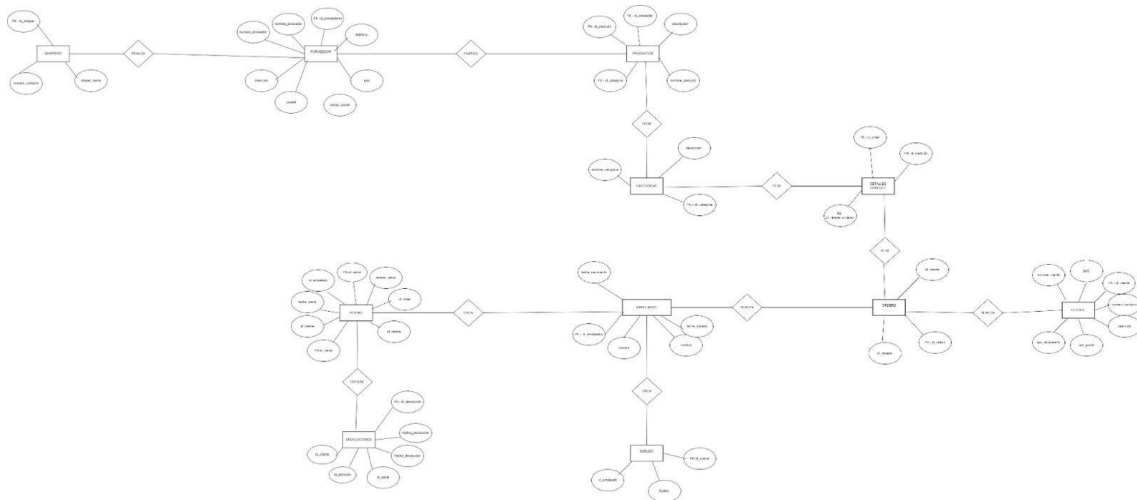
### Trigger orders\_transaction\_log

Este trigger se dispara después de insertar un registro en la tabla "orders". Registra los detalles de la inserción en la tabla "ordenes\_log", incluyendo la fecha y hora, el ID de la orden, el ID del cliente, el ID del empleado y el ID del shipper, junto con una descripción.

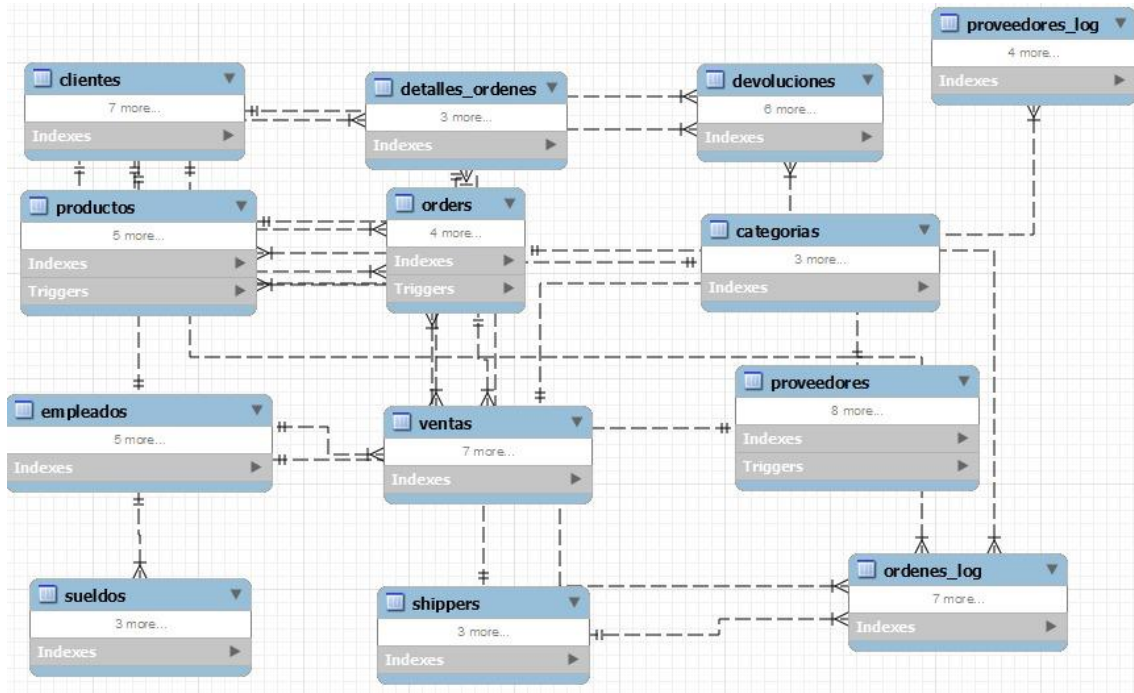
### Trigger validar\_producto\_existente

Este trigger se dispara antes de insertar un nuevo registro en la tabla "productos". Su objetivo es verificar si ya existe un producto con el mismo nombre en la tabla. Si existe, lanza un error que impide la inserción del nuevo producto.

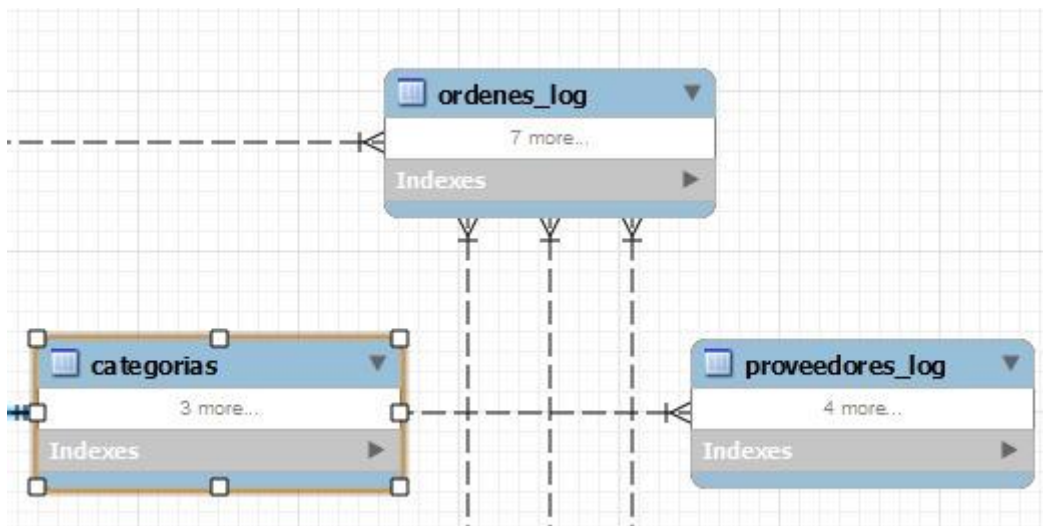
## DIAGRAMA ENTIDAD-RELACIÓN (CONCEPTUAL)



## DIAGRAMA ENTIDAD-RELACIÓN (ESQUEMÁTICO)



## TABLAS BITACORA



## LISTA DE TABLAS

A continuación, se desglosarán cómo fueron estructuradas y ordenadas las tablas:

Tabla	CLIENTES					
Desc	Datos de clientes					
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT
PK	Id_cliente	INT		NOT NULL	SI	AUTO_INCREMENT
	Nombre_cliente	TEXT		NOT NULL	NO	
	Numero_contacto	INT		NOT NULL	NO	
	Dirección	VARCHAR		NOT NULL	NO	
	Ciudad	TEXT		NOT NULL	NO	
	Código postal	VARCHAR		NOT NULL	NO	
	País	TEXT		NOT NULL	NO	

Tabla	EMPLEADOS					
Desc	Datos de empleados					
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT
PK	id_empleado	INT		NOT NULL	SI	AUTO_INCREMENT
	Nombre	TEXT		NOT NULL	NO	
	Apellido	TEXT		NOT NULL	NO	
	Fecha_nacimiento	DATE		NULL	NO	

Tabla	SHIPPERS					
Desc	Datos de shipper					
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT
PK	ID_SHIPPERS	INT		NOT NULL	SI	AUTO_INCREMENT
	SHIPPERS_NAME	VARCHAR	30	NOT NULL	NO	
	TELEFONO	INT		NOT NULL	NO	

Tabla	PROVEEDOR					
Desc	Datos de proveedores					
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT
PK	ID_PROVEEDOR	INT		NOT NULL	SI	AUTO_INCREMENT
	NOMBRE_PROVEEDOR	VARCHAR	30	NOT NULL	NO	
	NUMERO_PROVEEDOR			NOT NULL	NO	
	DIRECCION	VARCHAR	30	NOT NULL	NO	
	CIUDAD			NOT NULL	NO	
	CODIGO_POSTAL	VARCHAR	30	NOT NULL	NO	
	PAIS	TEX		NOT NULL	NO	
	TELEFONO	INT		NOT NULL	NO	



Tabla	CATEGORIAS					
Desc	Datos de las categorías					
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT
PK	ID_CATEGORIA	INT		NOT NULL	SI	AUTO_INCREMENT
	NOMBRE_CATEGORIA	VARCHAR	30	NOT NULL	NO	
	DESCRIPCION	VARCHAR	30	NOT NULL	NO	

Tabla	PRODUCTOS					
Desc	Datos de productos					
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT
PK	ID_PRODUCTO	INT		NOT NULL	SI	AUTO_INCREMENT
	NOMBRE_PRODUCTO	VACHAR	40	NOT NULL	NO	
FK	ID_PROVEEDOR	INT		NOT NULL		
FK	ID_CATEGORIA	VARCHAR	30	NOT NULL	SI	
	PRECIO	DECIMAL(M,N)		NOT NULL	SI	

Tabla	ORDERS					
Desc	Datos de orders					
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT
PK	ID_ORDER	INT		NOT NULL	SI	AUTO_INCREMENT
FK	Id_cliente	INT		NOT NULL	SI	
FK	id_empleado	INT		NOT NULL	SI	
FK	ID_SHIPPERS	INT		NOT NULL	SI	

Tabla	DETALLE_ORDENES					
Desc	Datos de ordenes					
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT
PK	ID_DETALLE_ORDEN	INT		NOT NULL	SI	AUTO_INCREMENT
FK	ID_ORDER	INT		NOT NULL	SI	
FK	ID_PRODUC	INT		NOT NULL	SI	

Tabla	PROYECTOS					
Desc	Datos ordenes_log					
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT
PK	id_proyecto	INT			No	AUTO_INCREMENT
	nombre_proyecto	VARCHAR(50)	50	NOT NULL	NO	
	descripcion	TEXT			NO	
	fecha_inicio	DATE			NO	
	fecha_fin	DATE			NO	
	responsable	VARCHAR(30)	30		NO	

Tabla	SUELDO					
Desc	Datos sueldo					
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT
PK	id_sueldo	INT		NOT NULL	No	AUTO_INCREMENT
	sueldo	VARCHAR(30)	50	NOT NULL	NO	
PK	id_empleado	VARCHAR(30)		NOT NULL	NO	

Tabla	DEVOLUCIONES					
Desc	Datos devoluciones					
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT
PK	id_devolucion	INT		NOT NULL		
FK	id_venta	INT		NOT NULL		
FK	id_producto	INT		NOT NULL		
FK	id_cliente	INT		NOT NULL		
	fecha_devolucion	DATE		NOT NULL		
	motivo_devolucion	TEXT				

Tabla	VENTA					
Desc	Datos ventas					
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT
PK	id_venta	INT		NOT NULL		
FK	id_order	INT		NOT NULL		
FK	id_empleado	INT		NOT NULL		
FK	id_cliente	INT		NOT NULL		
	fecha_venta	DATE		NOT NULL		
	estado_venta	DATE				

## INTELIGENCIA EMPRESARIAL



## TECNOLOGÍAS UTILIZADAS

SISTEMA	PRODUCTO	VERSION
DataBase	MySQL Community Server-GPL	8.0.33 for Win64 on x86_64
BI	PowerBI	Desktop