Taller Bases de Datos Relacionales

Juan Pablo Guzmán Martínez

ID: 1036822

Corporación Universitaria Minuto de Dios

Facultas Ingeniería de Sistemas

NRC-10-73461

Bogotá, Colombia

02 de Marzo de 2025

Introducción

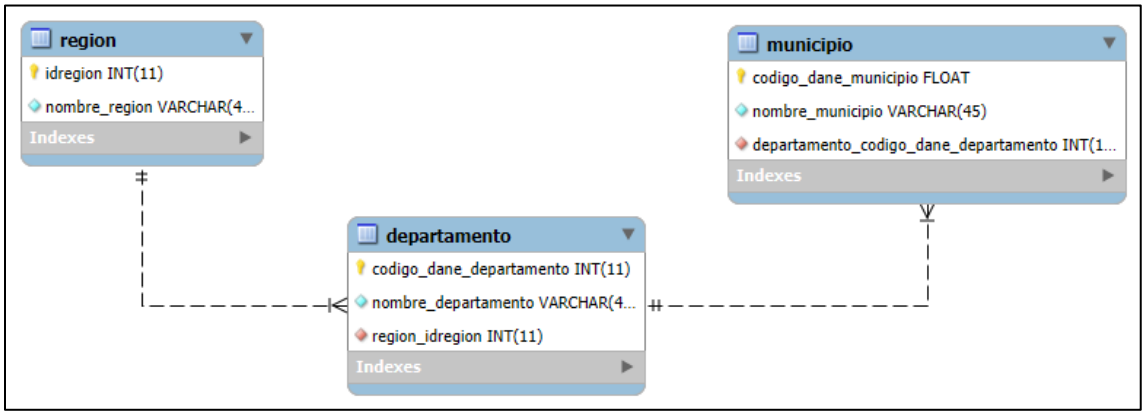
Este documento ayuda a presentar como es el desarrollo de una base de datos relacional, donde se incluyen su diagrama relacional, los scripts necesarios para su implementación y una explicación detallada de cada una de las realizaciones. El objetivo principal es poder demostrar el diseño y funcionamiento de una base de datos que es optimizada para su uso en entornos reales o en el mundo real. A lo largo del documento se muestran las relaciones entre las entidades, las estructuras de las tablas y los procedimientos necesarios para poder gestionar la información. Finalmente, se presentan las conclusiones basadas en la experiencia obtenida durante el desarrollo de este trabajo.

Tabla de contenido

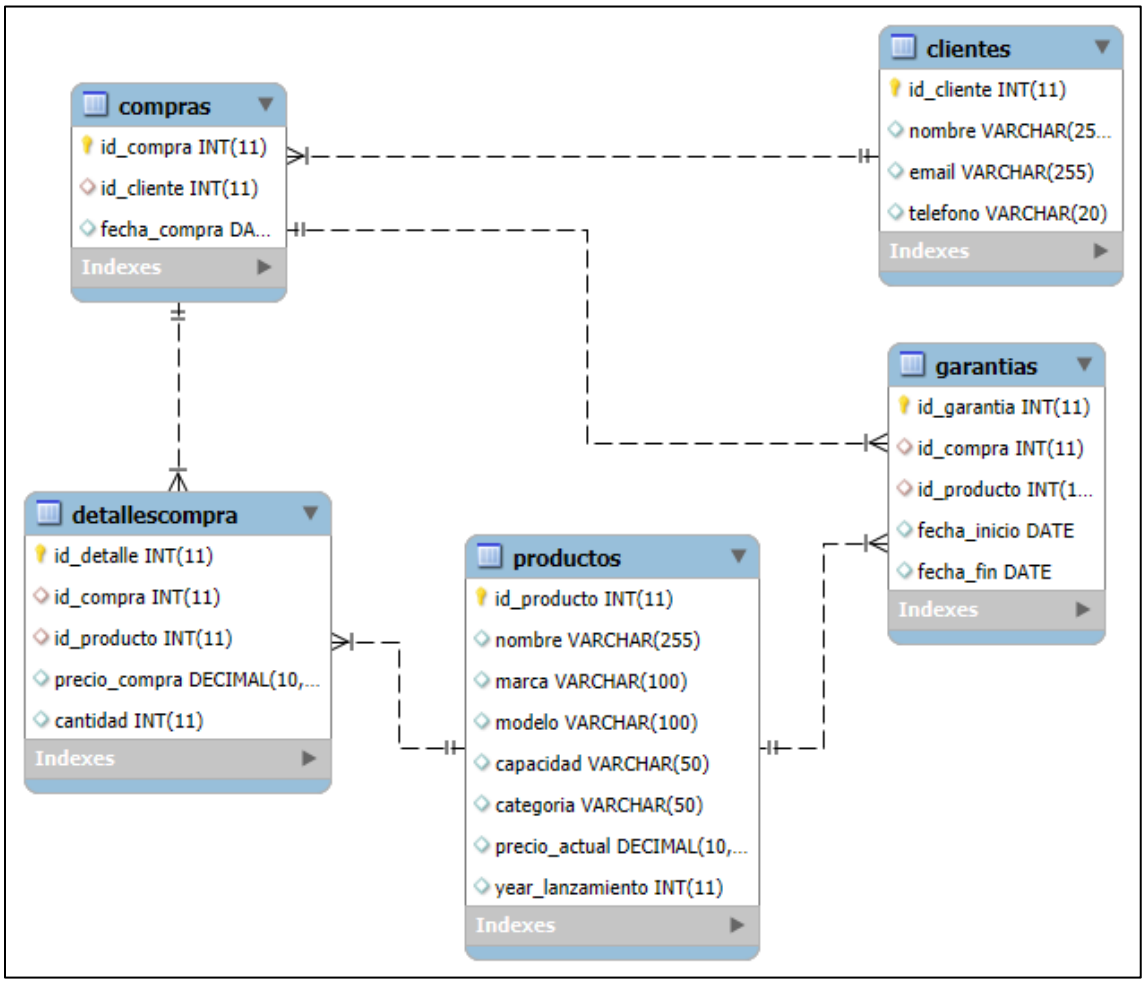
Desarrollo

Diagrama Relacional de las diferentes bases de datos relacionales solicitadas.

Base de Datos Relacional regiones, departamentos y municipios de Colombia:



Base de Datos Relacional Procesos de Facturación:



Scripts solicitados y necesarios para la defensa de la entrega.

Script de mi base de datos db_municipios:

```
      -- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERR
OR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----------------------------------------------------
-- Schema mydb
-- -----------------------------------------------------
-- -----------------------------------------------------
-- Schema fabrica
-- -----------------------------------------------------

-- -----------------------------------------------------
-- Schema fabrica
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `fabrica` DEFAULT CHARACTER SET utf8mb4 ;
-- -----------------------------------------------------
-- Schema db_municipios
-- -----------------------------------------------------

-- -----------------------------------------------------
-- Schema db_municipios
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `db_municipios` DEFAULT CHARACTER SET utf8mb4 COLLATE
utf8mb4_spanish_ci ;
USE `fabrica` ;

-- -----------------------------------------------------
-- Table `fabrica`.`instructores`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `fabrica`.`instructores` (
  `IdInstructor` INT(11) NOT NULL AUTO_INCREMENT,
  `NombreInstructor` VARCHAR(255) NULL DEFAULT NULL,
  `ApellidoInstructor` VARCHAR(255) NULL DEFAULT NULL,
  `TipoIdentificacion` ENUM('CC', 'TI') NULL DEFAULT NULL,
  `NumeroIdentificacion` INT(11) NULL DEFAULT NULL,
  `CorreoInstructor` VARCHAR(255) NULL DEFAULT NULL,
  `CelularInstructor` INT(11) NULL DEFAULT NULL,
```

```
  PRIMARY KEY (`IdInstructor`))
ENGINE = InnoDB
AUTO_INCREMENT = 3
DEFAULT CHARACTER SET = utf8mb4;


-- -----------------------------------------------------
-- Table `fabrica`.`productosgenerales`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `fabrica`.`productosgenerales` (
  `IdProducto` INT(11) NOT NULL AUTO_INCREMENT,
  `NombreProducto` VARCHAR(225) NULL DEFAULT NULL,
  `DescripcionProducto` VARCHAR(225) NULL DEFAULT NULL,
  `TipoProducto` VARCHAR(20) NULL DEFAULT NULL,
  `CantidadProducto` INT(11) NULL DEFAULT NULL,
  `ObservacionesProducto` VARCHAR(225) NULL DEFAULT NULL,
  PRIMARY KEY (`IdProducto`))
ENGINE = InnoDB
AUTO_INCREMENT = 6
DEFAULT CHARACTER SET = utf8mb4;


-- -----------------------------------------------------
-- Table `fabrica`.`prestamos`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `fabrica`.`prestamos` (
  `IdPrestamo` INT(11) NOT NULL AUTO_INCREMENT,
  `IdInstructor` INT(11) NULL DEFAULT NULL,
  `IdProducto` INT(11) NULL DEFAULT NULL,
  `FechaHoraPrestamo` DATETIME NULL DEFAULT NULL,
  `CantidadPrestamo` INT(11) NULL DEFAULT NULL,
  `EstadoPrestamo` ENUM('En curso', 'Culminados') NULL DEFAULT NULL,
  `ObservacionesPrestamo` VARCHAR(225) NULL DEFAULT NULL,
  PRIMARY KEY (`IdPrestamo`),
  INDEX `IdInstructor` (`IdInstructor` ASC) VISIBLE,
  INDEX `IdProducto` (`IdProducto` ASC) VISIBLE,
  CONSTRAINT `prestamos_ibfk_1`
    FOREIGN KEY (`IdInstructor`)
    REFERENCES `fabrica`.`instructores` (`IdInstructor`),
  CONSTRAINT `prestamos_ibfk_2`
    FOREIGN KEY (`IdProducto`)
    REFERENCES `fabrica`.`productosgenerales` (`IdProducto`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4;
```

```
-- -----------------------------------------------------
-- Table `fabrica`.`devoluciones`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `fabrica`.`devoluciones` (
  `IdDevoluciones` INT(11) NOT NULL AUTO_INCREMENT,
  `IdInstructor` INT(11) NULL DEFAULT NULL,
  `IdPrestamo` INT(11) NULL DEFAULT NULL,
  `IdProducto` INT(11) NULL DEFAULT NULL,
  `FechaHoraDevolucion` DATETIME NULL DEFAULT NULL,
  `EstadoDevolucion` ENUM('Bueno', 'Mal Estado') NULL DEFAULT NULL,
  `Observaciones` VARCHAR(225) NULL DEFAULT NULL,
  `ModoTiempoLugar` VARCHAR(225) NULL DEFAULT NULL,
  PRIMARY KEY (`IdDevoluciones`),
  INDEX `IdInstructor` (`IdInstructor` ASC) VISIBLE,
  INDEX `IdPrestamo` (`IdPrestamo` ASC) VISIBLE,
  INDEX `IdProducto` (`IdProducto` ASC) VISIBLE,
  CONSTRAINT `devoluciones_ibfk_1`
    FOREIGN KEY (`IdInstructor`)
    REFERENCES `fabrica`.`instructores` (`IdInstructor`),
  CONSTRAINT `devoluciones_ibfk_2`
    FOREIGN KEY (`IdPrestamo`)
    REFERENCES `fabrica`.`prestamos` (`IdPrestamo`),
  CONSTRAINT `devoluciones_ibfk_3`
    FOREIGN KEY (`IdProducto`)
    REFERENCES `fabrica`.`productosgenerales` (`IdProducto`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4;


-- -----------------------------------------------------
-- Table `fabrica`.`usuarios`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `fabrica`.`usuarios` (
  `IdUsuario` INT(11) NOT NULL AUTO_INCREMENT,
  `NombreUsuario` VARCHAR(255) NULL DEFAULT NULL,
  `ApellidoUsuario` VARCHAR(255) NULL DEFAULT NULL,
  `TipoIdentificacion` ENUM('CC', 'TI') NULL DEFAULT NULL,
  `NumeroIdentificacion` INT(11) NULL DEFAULT NULL,
  `CorreoUsuario` VARCHAR(255) NULL DEFAULT NULL,
  `CelularUsuario` INT(11) NULL DEFAULT NULL,
  `ContrasenaUsuario` VARCHAR(50) NULL DEFAULT NULL,
  PRIMARY KEY (`IdUsuario`))
ENGINE = InnoDB
AUTO_INCREMENT = 3
DEFAULT CHARACTER SET = utf8mb4;
```

```
USE `db_municipios` ;

-- -----------------------------------------------------
-- Table `db_municipios`.`region`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `db_municipios`.`region` (
  `idregion` INT(11) NOT NULL AUTO_INCREMENT,
  `nombre_region` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idregion`),
  UNIQUE INDEX `nombre_region` (`nombre_region` ASC) VISIBLE)
ENGINE = InnoDB
AUTO_INCREMENT = 1124
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_spanish_ci;


-- -----------------------------------------------------
-- Table `db_municipios`.`departamento`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `db_municipios`.`departamento` (
  `codigo_dane_departamento` INT(11) NOT NULL AUTO_INCREMENT,
  `nombre_departamento` VARCHAR(45) NOT NULL,
  `region_idregion` INT(11) NOT NULL,
  PRIMARY KEY (`codigo_dane_departamento`),
  INDEX `fk_departamento_region1_idx` (`region_idregion` ASC) VISIBLE,
  CONSTRAINT `fk_departamento_region1`
    FOREIGN KEY (`region_idregion`)
    REFERENCES `db_municipios`.`region` (`idregion`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB
AUTO_INCREMENT = 100
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_spanish_ci;


-- -----------------------------------------------------
-- Table `db_municipios`.`municipio`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `db_municipios`.`municipio` (
  `codigo_dane_municipio` FLOAT NOT NULL AUTO_INCREMENT,
  `nombre_municipio` VARCHAR(45) NOT NULL,
  `departamento_codigo_dane_departamento` INT(11) NOT NULL,
  PRIMARY KEY (`codigo_dane_municipio`),
```

```
  INDEX `fk_municipio_departamento_idx` (`departamento_codigo_dane_departamento`
ASC) VISIBLE,
  CONSTRAINT `fk_municipio_departamento`
    FOREIGN KEY (`departamento_codigo_dane_departamento`)
    REFERENCES `db_municipios`.`departamento` (`codigo_dane_departamento`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB
AUTO_INCREMENT = 100
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_spanish_ci;


SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Script de mi base de datos databaseJumbo:

```sql
CREATE DATABASE IF NOT EXISTS databaseJumbo;

USE databaseJumbo;

CREATE TABLE Productos (
    id_producto INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(255),
    marca VARCHAR(100),
    modelo VARCHAR(100),
    capacidad VARCHAR(50),
    categoria VARCHAR(50),
    precio_actual DECIMAL(10,2),
    year_lanzamiento INT
) ENGINE=InnoDB;

CREATE TABLE Clientes (
    id_cliente INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(255),
    email VARCHAR(255),
    telefono VARCHAR(20)
) ENGINE=InnoDB;

CREATE TABLE Compras (
    id_compra INT AUTO_INCREMENT PRIMARY KEY,
    id_cliente INT,
    fecha_compra DATE,
    FOREIGN KEY (id_cliente) REFERENCES Clientes(id_cliente)
) ENGINE=InnoDB;

CREATE TABLE DetallesCompra (
    id_detalle INT AUTO_INCREMENT PRIMARY KEY,
    id_compra INT,
    id_producto INT,
    precio_compra DECIMAL(10,2),
    cantidad INT,
    FOREIGN KEY (id_compra) REFERENCES Compras(id_compra),
    FOREIGN KEY (id_producto) REFERENCES Productos(id_producto)
) ENGINE=InnoDB;

CREATE TABLE Garantias (
    id_garantia INT AUTO_INCREMENT PRIMARY KEY,
    id_compra INT,
    id_producto INT,
```

```
    fecha_inicio DATE,
    fecha_fin DATE,
    FOREIGN KEY (id_compra) REFERENCES Compras(id_compra) ON DELETE CASCADE,
    FOREIGN KEY (id_producto) REFERENCES Productos(id_producto) ON DELETE CASCADE
) ENGINE=InnoDB;

-- Inserciones a las tablasd
INSERT INTO Productos (nombre, marca, modelo, capacidad, categoria,
precio_actual, year_lanzamiento)
VALUES
('Nevera Mabe 300L', 'Mabe', 'MB300', '300 Litros', 'Nevera', 1800000, 2023),
('Televisor Samsung 55"', 'Samsung', 'QLED55', '55 Pulgadas', 'Televisor',
3000000, 2023);

INSERT INTO Clientes (nombre, email, telefono)
VALUES
('Juan Pérez', 'juanperez@email.com', '3001234567');

INSERT INTO Compras (id_cliente, fecha_compra)
VALUES (1, '2023-06-15');

INSERT INTO DetallesCompra (id_compra, id_producto, precio_compra, cantidad)
VALUES (1, 1, 1800000, 1);

INSERT INTO Garantias (id_compra, id_producto, fecha_inicio, fecha_fin)
VALUES (1, 1, '2023-06-15', '2025-06-15');

SELECT * FROM Productos;
SELECT * FROM Clientes;
SELECT * FROM Compras;
SELECT * FROM DetallesCompra;
SELECT * FROM Garantias;

DELIMITER $$

CREATE PROCEDURE depreciar_precios()
BEGIN
    UPDATE Productos
    SET precio_actual = precio_actual * (1 - 0.10);
END$$

DELIMITER ;

CREATE EVENT IF NOT EXISTS depreciar_precios_event
ON SCHEDULE EVERY 1 YEAR
```

```
STARTS '2025-02-17 12:12:30'
DO
  CALL depreciar_precios();

SELECT * FROM Productos;


SELECT g.id_garantia, c.id_compra, p.nombre, p.marca, p.modelo, dc.precio_compra
    AS precio_original, p.precio_actual AS precio_actual
FROM Garantias g
JOIN Compras c ON g.id_compra = c.id_compra
JOIN DetallesCompra dc ON c.id_compra = dc.id_compra
JOIN Productos p ON dc.id_producto = p.id_producto
WHERE c.id_cliente = 1 AND p.id_producto = 1;
```

Descripciones o explicaciones de los scripts solicitados.

Tuve un problema en la instalación con el Docker en mi equipo, sin embargo, intenté hacerlo por medio de una maquina virtual con Docker en la nube de AWS, y tampoco fue posible.

De todos modos, tengo los scripts para indexar los datos a del archivo CSV a las tablas de "*db_municipios*":

```
Administrador: XAMPP for Windows - mysql  -u root -p
MariaDB [db_municipios]> SOURCE C:/Users/user/Documents/DB_masivas/databases/db_municipios.sql;
Query OK, 0 rows affected (0.081 sec)

Query OK, 0 rows affected (0.001 sec)

Query OK, 0 rows affected (0.004 sec)

Query OK, 0 rows affected, 1 warning (0.025 sec)

Database changed
Query OK, 0 rows affected (0.294 sec)

Query OK, 0 rows affected (0.062 sec)

Query OK, 0 rows affected (0.045 sec)

Query OK, 0 rows affected (0.001 sec)

Query OK, 0 rows affected (0.001 sec)

Query OK, 0 rows affected (0.000 sec)
```

Regiones:

```
LOAD DATA INFILE 'C:/Users/user/Documents/DB_masivas/datos_municipios.csv'
REPLACE INTO TABLE region
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(@nombre_region) SET nombre_region = TRIM(@nombre_region);
```

```
Administrador: XAMPP for Windows - mysql  -u root -p
MariaDB [db_municipios]> SOURCE C:/Users/user/Documents/DB_masivas/databases/indexacion_regiones.sql;
Query OK, 2240 rows affected, 1123 warnings (2.557 sec)
Records: 1123  Deleted: 1117  Skipped: 0  Warnings: 1123
```

```
Administrador: XAMPP for Windows - mysql  -u root -p
MariaDB [db_municipios]> Select * from region;
+----------+---------------------------------+
| idregion | nombre_region                   |
+----------+---------------------------------+
|     1123 | Regi├│n Caribe                   |
|     1117 | Regi├│n Centro Oriente           |
|     1116 | Regi├│n Centro Sur               |
|     1119 | Regi├│n Eje Cafetero - Antioquia |
|     1112 | Regi├│n Llano                    |
|     1121 | Regi├│n Pac├¡fico                 |
+----------+---------------------------------+
6 rows in set (0.052 sec)
```

Departamentos:

```
LOAD DATA INFILE 'C:/Users/user/Documents/DB_masivas/datos_municipios.csv'
REPLACE INTO TABLE departamento
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(@var1, @var2, @var3, @dummy1, @dummy2)
SET
    codigo_dane_departamento = TRIM(@var2),
    nombre_departamento = TRIM(@var3),
    region_idregion = (SELECT idregion FROM region WHERE TRIM(nombre_region) =
TRIM(@var1) LIMIT 1);
```

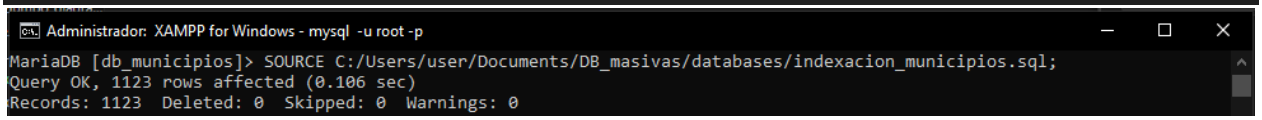
```
Administrador: XAMPP for Windows - mysql  -u root -p
MariaDB [db_municipios]> SOURCE C:/Users/user/Documents/DB_masivas/databases/indexacion_departamentos.sql;
Query OK, 2213 rows affected, 2 warnings (0.109 sec)
Records: 1123  Deleted: 1090  Skipped: 0  Warnings: 2
```

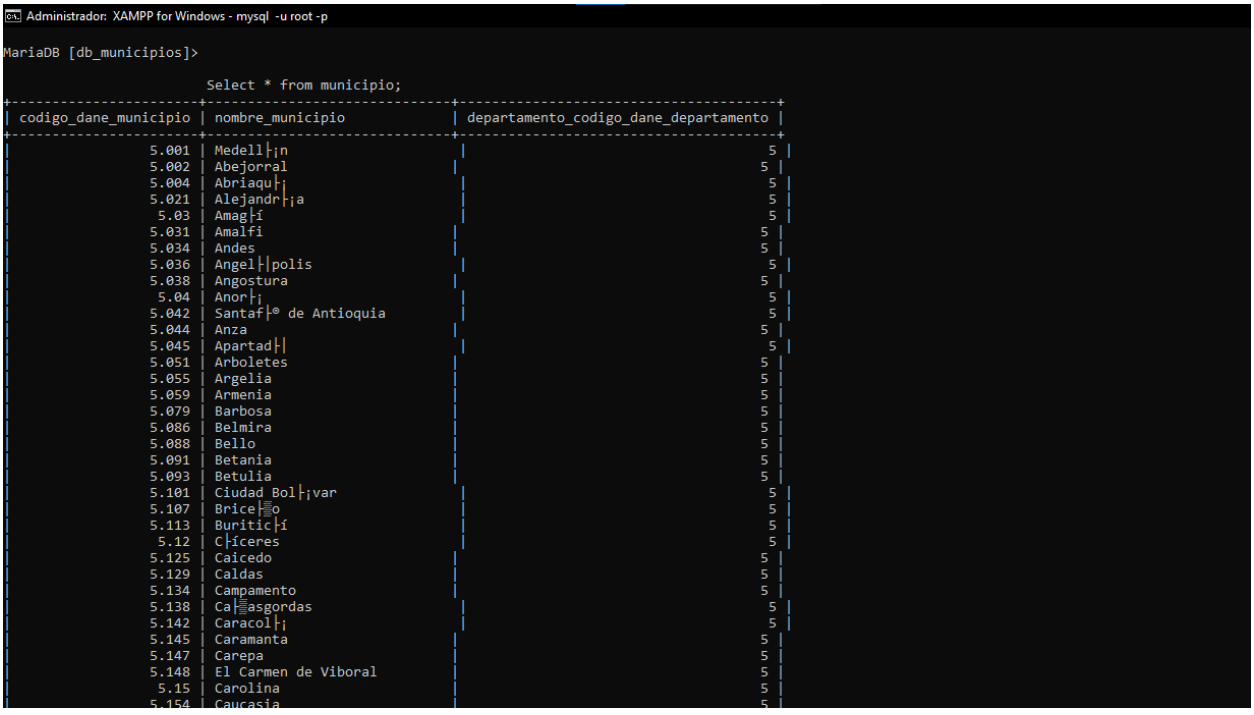
```
Administrador: XAMPP for Windows - mysql  -u root -p

MariaDB [db_municipios]> Select * from departamento;
+-------------------------+-------------------------------------------------+-----------------+
| codigo_dane_departamento | nombre_departamento                             | region_idregion |
+-------------------------+-------------------------------------------------+-----------------+
|                       5 | Antioquia                                       |            1119 |
|                       8 | Atl├íntico                                       |            1123 |
|                      11 | Bogot├í D.C.                                     |            1117 |
|                      13 | Bol├¡var                                         |            1123 |
|                      15 | Boyac├í                                          |            1117 |
|                      17 | Caldas                                          |            1119 |
|                      18 | Caquet├í                                         |            1116 |
|                      19 | Cauca                                           |            1121 |
|                      20 | Cesar                                           |            1123 |
|                      23 | C├│rdoba                                         |            1123 |
|                      25 | Cundinamarca                                    |            1117 |
|                      27 | Choc├│                                           |            1121 |
|                      41 | Huila                                           |            1116 |
|                      44 | La Guajira                                      |            1123 |
|                      47 | Magdalena                                       |            1123 |
|                      50 | Meta                                            |            1112 |
|                      52 | Nari├▒o                                          |            1121 |
|                      54 | Norte de Santander                              |            1117 |
|                      63 | Quind├¡o                                         |            1119 |
|                      66 | Risaralda                                       |            1119 |
|                      68 | Santander                                       |            1117 |
|                      70 | Sucre                                           |            1123 |
|                      73 | Tolima                                          |            1116 |
|                      76 | Valle del Cauca                                 |            1121 |
|                      81 | Arauca                                          |            1112 |
|                      85 | Casanare                                        |            1112 |
|                      86 | Putumayo                                        |            1116 |
|                      88 | Archipi├®lago de San Andr├®s, Providencia y San  |            1123 |
|                      91 | Amazonas                                        |            1116 |
|                      94 | Guain├¡a                                         |            1112 |
|                      95 | Guaviare                                        |            1112 |
|                      97 | Vaup├®s                                          |            1116 |
|                      99 | Vichada                                         |            1112 |
+-------------------------+-------------------------------------------------+-----------------+
33 rows in set (0.088 sec)
```

Municipios:

```
LOAD DATA INFILE 'C:/Users/user/Documents/DB_masivas/datos_municipios.csv'
REPLACE INTO TABLE municipio
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(@var1, @var2, @dummy1, @var4, @var5)
SET
    codigo_dane_municipio = TRIM(@var4),   -- Código DANE del municipio
    nombre_municipio = TRIM(@var5),        -- Nombre del municipio
    departamento_codigo_dane_departamento = (
        SELECT codigo_dane_departamento
        FROM departamento
        WHERE TRIM(codigo_dane_departamento) = TRIM(@var2)
        LIMIT 1
    );
```



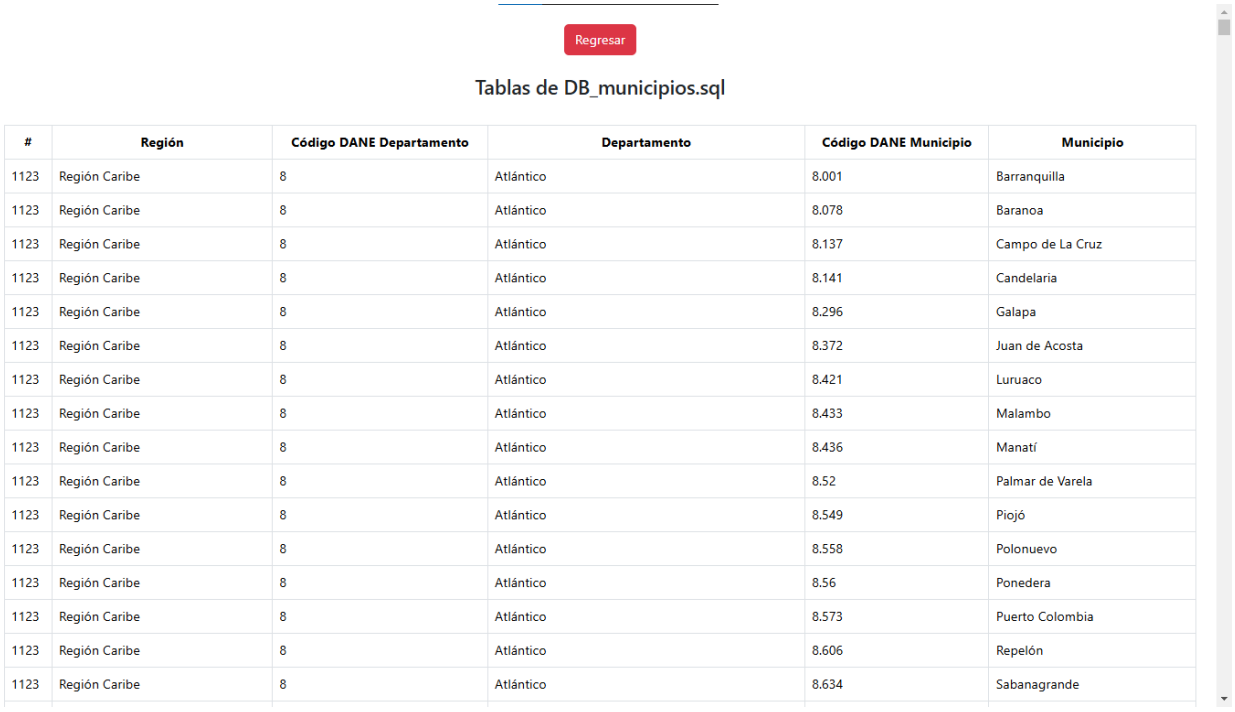


…

Conclusiones Scripts Load Data Infile

Lo que hace este código de Load Data Infile es leer las filas que se escriben por mediante el script en la instrucción de *SELECT INTO OUTFILE*, después de esto ya puede volver a leer los archivos en las tablas por medio de la instrucción *LOAD DATA INFILE*. Las instrucciones *FIELDS* y *LINES* son las mismas instrucciones que de la misma manera predeterminada hace que se esperen que los campos terminen en tabulaciones y con nuevas líneas (*n*). Cuando se ejecuta esta declaración se activan los *INSERT* en los activadores.

Se tiene que tener el privilegio y el permiso de *FILE* para que se pueda ejecutar el *LOAD DATA INFILE*. Esto nos ayuda a que los usuarios normales no puedan leer los archivos del sistema. *LOAD DATA LOCAL INFILE* no tienen ese requisito. De esta forma queda la indexación como una tabla igual que en el CSV:



Regresar

Tablas de DB_municipios.sql

#	Región	Código DANE Departamento	Departamento	Código DANE Municipio	Municipio
1123	Región Caribe	8	Atlántico	8.001	Barranquilla
1123	Región Caribe	8	Atlántico	8.078	Baranoa
1123	Región Caribe	8	Atlántico	8.137	Campo de La Cruz
1123	Región Caribe	8	Atlántico	8.141	Candelaria
1123	Región Caribe	8	Atlántico	8.296	Galapa
1123	Región Caribe	8	Atlántico	8.372	Juan de Acosta
1123	Región Caribe	8	Atlántico	8.421	Luruaco
1123	Región Caribe	8	Atlántico	8.433	Malambo
1123	Región Caribe	8	Atlántico	8.436	Manatí
1123	Región Caribe	8	Atlántico	8.52	Palmar de Varela
1123	Región Caribe	8	Atlántico	8.549	Piojó
1123	Región Caribe	8	Atlántico	8.558	Polonuevo
1123	Región Caribe	8	Atlántico	8.56	Ponedera
1123	Región Caribe	8	Atlántico	8.573	Puerto Colombia
1123	Región Caribe	8	Atlántico	8.606	Repelón
1123	Región Caribe	8	Atlántico	8.634	Sabanagrande

Conclusiones Script databaseJumbo

En el script de mirar la garantía con precio de compra, precio original y precio actual

```
SELECT g.id_garantia, c.id_compra, p.nombre, p.marca, p.modelo, dc.precio_compra
    AS precio_original, p.precio_actual AS precio_actual
FROM Garantias g
JOIN Compras c ON g.id_compra = c.id_compra
JOIN DetallesCompra dc ON c.id_compra = dc.id_compra
JOIN Productos p ON dc.id_producto = p.id_producto
WHERE c.id_cliente = 1 AND p.id_producto = 1;
```

Nos da la posibilidad de mirar cómo se recupera la información sobre la garantía de un producto en específico que fué comprado por un cliente en particular. Si el cliente con id_cliente = 1 compró un producto con id_producto = 1, la consulta devolverá información sobre:

1. La garantía a la cual está asociada a esa compra
2. Los detalles del producto como el nombre, la marca y el modelo
3. El precio original al que se compró el producto
4. El precio actual que sirve para ver si ha cambiado desde que se hizo la compra

Conclusiones

Es un trabajo orgánico que puede suceder el cualquier entorno real, el cuál nos enseña aún más a optimizar las tareas de datos por medio de entradas de SQL las cuales nos hacen la vida mucho más fácil, espero aprender mucho más de ello para poder implementar en un futuro el cuál sé que necesitaré implementarlo. Esto nos ha ayudad mucho a tener bases de datos masivas que permiten almacenar, procesar y analizar grandes cantidades de información de manera estructurada, lo cual no facilita mucho en la toma de decisiones en diferentes sectores como la industria tecnológica, en la salud y en el comercio.