

## Documentación de la Base de Datos: inventario

La base de datos inventario está diseñada para gestionar los productos, empleados, ventas y proveedores de un sistema de control de inventario. A continuación, se presenta una descripción detallada de las tablas que componen el esquema inventario, junto con sus columnas, claves primarias, claves foráneas y relaciones entre ellas.

### 1. Tabla local

#### Descripción:

La tabla local almacena información sobre los diferentes locales o puntos de venta de la empresa. Cada local puede ser una tienda, una oficina o cualquier otro establecimiento físico donde se gestionen productos o ventas. Esta tabla permite almacenar datos clave como la ubicación y los datos de contacto de cada local.

#### Estructura:

- **id\_local** (INT, AUTO\_INCREMENT): Identificador único para cada local. Es la clave primaria de la tabla y se autoincrementa automáticamente con cada nuevo registro.
- **tipo\_local** (VARCHAR(50)): Tipo o categoría del local, como "tienda", "oficina", "almacén", etc.
- **nombre** (VARCHAR(200)): Nombre comercial del local, por ejemplo, "Sucursal Centro" o "Tienda San Sebastián".
- **direccion** (VARCHAR(200)): Dirección física del local.
- **telefono** (VARCHAR(12)): Número de teléfono de contacto del local.
- **email** (VARCHAR(200)): Correo electrónico de contacto del local.

#### Clave primaria:

- **id\_local**: Es la clave primaria, identificando de forma única cada local.

## 2. Tabla empleado

### Descripción:

La tabla empleado contiene información sobre los empleados de la empresa. Esta tabla no solo almacena los datos personales de los empleados, sino también su relación con los locales en los que trabajan, permitiendo así gestionar las funciones laborales de cada uno dentro de un establecimiento determinado.

### Estructura:

- **id\_empleado** (INT, AUTO\_INCREMENT): Identificador único para cada empleado. Es la clave primaria de la tabla.
- **rut** (VARCHAR(12)): El RUT (Rol Único Tributario) del empleado, utilizado en algunos países como identificador fiscal.
- **tipo\_empleado** (VARCHAR(45)): El tipo de empleado, como "vendedor", "gerente", "administrativo", etc.
- **nombre** (VARCHAR(200)): Nombre del empleado.
- **apellido** (VARCHAR(200)): Apellido del empleado.
- **direccion\_empleado** (VARCHAR(200)): Dirección personal del empleado.
- **telefono** (VARCHAR(12)): Teléfono de contacto del empleado.
- **email** (VARCHAR(200)): Correo electrónico del empleado.
- **local\_id** (INT): ID del local donde el empleado trabaja. Este campo es una clave foránea que hace referencia a la tabla local.

### Clave primaria:

- **id\_empleado**: Identificador único de cada empleado.

### Clave foránea:

- **local\_id**: Hace referencia a **id\_local** de la tabla local. Esta relación establece que un empleado está asociado a un único local.

### 3. Tabla venta

#### Descripción:

La tabla venta almacena las transacciones de venta realizadas en los locales de la empresa. Cada venta tiene un total de venta, una fecha y un vínculo con el empleado que realizó la venta.

#### Estructura:

- **id\_venta** (INT, AUTO\_INCREMENT): Identificador único de cada venta. Es la clave primaria de la tabla.
- **total\_venta** (INT): El monto total de la venta en unidades monetarias.
- **fecha\_venta** (DATETIME): Fecha y hora exacta en la que se realizó la venta.
- **id\_empleado** (INT): El ID del empleado que realizó la venta. Es una clave foránea que hace referencia a la tabla empleado.

#### Clave primaria:

- **id\_venta**: Identificador único de la venta.

#### Clave foránea:

- **id\_empleado**: Hace referencia a **id\_empleado** de la tabla empleado, estableciendo que cada venta está asociada con un empleado específico.

## 4. Tabla categoria

### Descripción:

La tabla categoria organiza los productos en categorías, lo que facilita la clasificación y búsqueda de productos dentro del inventario. Una categoría puede tener una categoría principal (padre), lo que permite la creación de subcategorías.

### Estructura:

- **id\_categoria** (INT, AUTO\_INCREMENT): Identificador único de cada categoría. Es la clave primaria de la tabla.
- **nombre** (VARCHAR(200)): El nombre de la categoría, como "Electrónica", "Ropa", etc.
- **categoria\_padre\_id** (INT, NULL): Hace referencia a una categoría principal (padre) para las subcategorías. Si el valor es NULL, la categoría es de nivel superior.

### Clave primaria:

- **id\_categoria**: Identificador único de la categoría.

### Clave foránea:

- **categoria\_padre\_id**: Hace referencia a **id\_categoria** de la misma tabla, permitiendo la jerarquización de categorías en subcategorías.

## 5. Tabla productos

### Descripción:

La tabla productos contiene todos los productos disponibles en el inventario. Cada producto está asociado a una categoría y tiene atributos como el precio, el stock disponible y la fecha de caducidad (si aplica).

### Estructura:

- **id\_productos** (INT, AUTO\_INCREMENT): Identificador único de cada producto. Es la clave primaria de la tabla.
- **nombre** (VARCHAR(200)): Nombre del producto.
- **precio** (INT): Precio del producto.
- **stock** (INT): Cantidad disponible del producto en inventario.
- **fecha\_caducidad** (DATETIME): Fecha de caducidad del producto, si aplica.
- **codigo\_barra** (VARCHAR(200)): Código de barras del producto.
- **categoria\_id** (INT): ID de la categoría del producto. Es una clave foránea que hace referencia a la tabla categoria.

### Clave primaria:

- **id\_productos**: Identificador único de cada producto.

### Clave foránea:

- **categoria\_id**: Hace referencia a **id\_categoria** de la tabla categoria, asignando cada producto a una categoría específica.

## 6. Tabla proveedor

### Descripción:

La tabla proveedor almacena información sobre los proveedores de los productos. Cada proveedor tiene un nombre y un número de contacto.

### Estructura:

- **id\_proveedor** (INT, AUTO\_INCREMENT): Identificador único del proveedor. Es la clave primaria de la tabla.
- **nombre\_proveedor** (VARCHAR(200)): Nombre del proveedor.
- **numero\_movil** (VARCHAR(12)): Número de teléfono móvil del proveedor.

### Clave primaria:

- **id\_proveedor**: Identificador único del proveedor.

## 7. Tabla **venta\_productos**

### **Descripción:**

La tabla **venta\_productos** almacena los productos vendidos en cada transacción de venta. Esta tabla actúa como una relación de muchos a muchos entre las tablas **venta** y **productos**, detallando qué productos se vendieron en qué venta y en qué cantidad.

### **Estructura:**

- **venta\_id** (INT): ID de la venta. Es una clave foránea que hace referencia a la tabla **venta**.
- **productos\_id** (INT): ID del producto vendido. Es una clave foránea que hace referencia a la tabla **productos**.
- **cantidad** (INT): Cantidad de productos vendidos.
- **precio\_unitario** (INT): Precio unitario del producto en el momento de la venta.

### **Claves primarias:**

- Combinación de **venta\_id** y **productos\_id**.

### **Claves foráneas:**

- **venta\_id**: Hace referencia a **id\_venta** de la tabla **venta**.
- **productos\_id**: Hace referencia a **id\_productos** de la tabla **productos**.

## 8. Tabla usuarios

### Descripción:

La tabla usuarios almacena las credenciales de los usuarios del sistema, vinculándolos con un empleado específico. Esta tabla es útil para la autenticación y autorización de acceso al sistema.

### Estructura:

- **id\_usuarios** (INT, AUTO\_INCREMENT): Identificador único de cada usuario. Es la clave primaria de la tabla.
- **user** (VARCHAR(45)): Nombre de usuario para la autenticación.
- **pass** (VARCHAR(255)): Contraseña del usuario.
- **tipo\_empleado** (VARCHAR(45)): Tipo de empleado que tiene acceso a este usuario.
- **empleado\_id** (INT): ID del empleado asociado a este usuario. Es una clave foránea que hace referencia a la tabla empleado.

### Clave primaria:

- **id\_usuarios**: Identificador único del usuario.

### Clave foránea:

- **empleado\_id**: Hace referencia a **id\_empleado** de la tabla empleado.



## 9. Tabla **productos\_has\_proveedor**

### **Descripción:**

La tabla **productos\_has\_proveedor** establece una relación muchos a muchos entre los productos y los proveedores. Un producto puede ser suministrado por varios proveedores, y un proveedor puede suministrar varios productos.

### **Estructura:**

- **productos\_id** (INT): ID del producto. Es una clave foránea que hace referencia a la tabla **productos**.
- **proveedor\_id** (INT): ID del proveedor. Es una clave foránea que hace referencia a la tabla **proveedor**.

### **Claves primarias:**

- Combinación de **productos\_id** y **proveedor\_id**.

### **Claves foráneas:**

- **productos\_id**: Hace referencia a **id\_productos** de la tabla **productos**.
- **proveedor\_id**: Hace referencia a **id\_proveedor** de la tabla **proveedor**.

## 10. Tabla producto\_control

### Descripción:

La tabla producto\_control se utiliza para controlar el estado de los productos en el inventario. Permite gestionar el stock y registrar si un producto está en buen estado, caducado o en merma.

### Estructura:

- **id\_control** (INT, AUTO\_INCREMENT): Identificador único del control de producto.
- **nombre** (VARCHAR(200)): Nombre del producto en control.
- **precio** (INT): Precio del producto.
- **stock** (INT): Stock disponible del producto.
- **fecha\_caducidad** (DATETIME): Fecha de caducidad del producto.
- **estado** (ENUM('activo', 'caducado', 'merma')): Estado del producto, indicando si está activo, caducado o en merma.
- **fecha\_registro** (TIMESTAMP): Fecha en que se registró el control del producto.
- **id\_productos** (INT): ID del producto al que pertenece este control. Es una clave foránea que hace referencia a la tabla productos.

### Clave primaria:

- **id\_control**: Identificador único del registro de control.

### Clave foránea:

- **id\_productos**: Hace referencia a **id\_productos** de la tabla productos.

## 11. Tabla historial\_movimiento

### Descripción:

La tabla historial\_movimiento registra los movimientos de los productos dentro del inventario. Esto incluye entradas y salidas de productos, como compras, ventas, devoluciones, etc.

### Estructura:

- **id\_movimiento** (INT, AUTO\_INCREMENT): Identificador único del movimiento. Es la clave primaria de la tabla.
- **fecha\_movimiento** (TIMESTAMP): Fecha y hora en que ocurrió el movimiento del producto.
- **cantidad** (INT): Cantidad de productos que se movieron (ya sea entrada o salida).
- **tipo\_movimiento** (VARCHAR(200)): Tipo de movimiento, como "entrada", "salida", "venta", "compra", etc.
- **usuario** (VARCHAR(100)): Nombre del usuario que realizó el movimiento.
- **id\_productos** (INT): ID del producto que fue movido. Es una clave foránea que hace referencia a la tabla productos.

### Clave primaria:

- **id\_movimiento**: Identificador único del movimiento.

### Clave foránea:

- **id\_productos**: Hace referencia a **id\_productos** de la tabla productos.

## Script

-- 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,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SU  
BSTITUTION';

-- -----

-- Schema inventario

-- -----

-- -----

-- Schema inventario

-- -----

CREATE SCHEMA IF NOT EXISTS `inventario` DEFAULT CHARACTER SET utf8  
;

USE `inventario` ;

-- -----

-- Table `inventario`.`local`

-- -----

CREATE TABLE IF NOT EXISTS `inventario`.`local` (  
 `id\_local` INT NOT NULL AUTO\_INCREMENT,  
 `tipo\_local` VARCHAR(50) NOT NULL,  
 `nombre` VARCHAR(200) NOT NULL,

```
`direccion` VARCHAR(200) NOT NULL,  
`telefono` VARCHAR(12) NOT NULL,  
`email` VARCHAR(200) NOT NULL,  
PRIMARY KEY (`id_local`))  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `inventario`.`empleado`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `inventario`.`empleado` (  
  `id_empleado` INT NOT NULL AUTO_INCREMENT,  
  `rut` VARCHAR(12) NOT NULL,  
  `tipo_empleado` VARCHAR(45) NOT NULL,  
  `nombre` VARCHAR(200) NOT NULL,  
  `apellido` VARCHAR(200) NOT NULL,  
  `direccion_empleado` VARCHAR(200) NOT NULL,  
  `telefono` VARCHAR(12) NOT NULL,  
  `email` VARCHAR(200) NOT NULL,  
  `local_id` INT NOT NULL,  
  PRIMARY KEY (`id_empleado`),  
  CONSTRAINT `fk_empleado_local`  
    FOREIGN KEY (`local_id`)  
    REFERENCES `inventario`.`local` (`id_local`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```

-----
-- Table `inventario`.`venta`
-----

CREATE TABLE IF NOT EXISTS `inventario`.`venta` (
  `id_venta` INT NOT NULL AUTO_INCREMENT,
  `total_venta` INT NOT NULL,
  `fecha_venta` DATETIME NOT NULL,
  `id_empleado` INT NOT NULL,
  PRIMARY KEY (`id_venta`),
  CONSTRAINT `fk_venta_empleado1`
    FOREIGN KEY (`id_empleado`)
    REFERENCES `inventario`.`empleado` (`id_empleado`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `inventario`.`categoria`
-----

CREATE TABLE IF NOT EXISTS `inventario`.`categoria` (
  `id_categoria` INT NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(200) NOT NULL,
  `categoria_padre_id` INT NULL,
  PRIMARY KEY (`id_categoria`),
  CONSTRAINT `fk_categoria_categoria1`
    FOREIGN KEY (`categoria_padre_id`)

```

```
REFERENCES `inventario`.`categoria` (`id_categoria`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- Table `inventario`.`productos`
```

```
CREATE TABLE IF NOT EXISTS `inventario`.`productos` (
  `id_productos` INT NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(200) NOT NULL,
  `precio` INT NOT NULL,
  `stock` INT NOT NULL,
  `fecha_caducidad` DATETIME NOT NULL,
  `codigo_barra` VARCHAR(200) NOT NULL,
  `categoria_id` INT NOT NULL,
  PRIMARY KEY (`id_productos`),
  CONSTRAINT `fk_productos_categoria1`
  FOREIGN KEY (`categoria_id`)
  REFERENCES `inventario`.`categoria` (`id_categoria`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- Table `inventario`.`proveedor`
```

```

-----
CREATE TABLE IF NOT EXISTS `inventario`.`proveedor` (
  `id_proveedor` INT NOT NULL AUTO_INCREMENT,
  `nombre_proveedor` VARCHAR(200) NOT NULL,
  `numero_movil` VARCHAR(12) NOT NULL,
  PRIMARY KEY (`id_proveedor`))
ENGINE = InnoDB;

```

```

-----
-- Table `inventario`.`venta_productos`
-----

```

```

CREATE TABLE IF NOT EXISTS `inventario`.`venta_productos` (
  `venta_id` INT NOT NULL,
  `productos_id` INT NOT NULL,
  `cantidad` INT NOT NULL,
  `precio_unitario` INT NOT NULL,
  PRIMARY KEY (`venta_id`, `productos_id`),
  CONSTRAINT `fk_venta_has_productos_venta1`
    FOREIGN KEY (`venta_id`)
      REFERENCES `inventario`.`venta` (`id_venta`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_venta_has_productos_productos1`
    FOREIGN KEY (`productos_id`)
      REFERENCES `inventario`.`productos` (`id_productos`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)

```



ENGINE = InnoDB;

-- -----

-- Table `inventario`.`usuarios`

-- -----

```
CREATE TABLE IF NOT EXISTS `inventario`.`usuarios` (  
  `id_usuarios` INT NOT NULL AUTO_INCREMENT,  
  `user` VARCHAR(45) NOT NULL,  
  `pass` VARCHAR(255) NOT NULL,  
  `tipo_empleado` VARCHAR(45) NOT NULL,  
  `empleado_id` INT NOT NULL,  
  PRIMARY KEY (`id_usuarios`),  
  CONSTRAINT `fk_usuarios_empleado1`  
    FOREIGN KEY (`empleado_id`)  
    REFERENCES `inventario`.`empleado` (`id_empleado`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-- -----

-- Table `inventario`.`productos\_has\_proveedor`

-- -----

```
CREATE TABLE IF NOT EXISTS `inventario`.`productos_has_proveedor` (  
  `productos_id` INT NOT NULL,  
  `proveedor_id` INT NOT NULL,  
  PRIMARY KEY (`productos_id`, `proveedor_id`),
```

```

CONSTRAINT `fk_productos_has_proveedor_productos1`
  FOREIGN KEY (`productos_id`)
  REFERENCES `inventario`.`productos` (`id_productos`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_productos_has_proveedor_proveedor1`
  FOREIGN KEY (`proveedor_id`)
  REFERENCES `inventario`.`proveedor` (`id_proveedor`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-- -----
-- Table `inventario`.`producto_control`
-- -----

CREATE TABLE IF NOT EXISTS `inventario`.`producto_control` (
  `id_control` INT NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(200) NOT NULL,
  `precio` INT NOT NULL,
  `stock` INT NOT NULL,
  `fecha_caducidad` DATETIME NOT NULL,
  `estado` ENUM('activo', 'caducado', 'merma') NOT NULL,
  `fecha_registro` TIMESTAMP NOT NULL,
  `id_productos` INT NOT NULL,
  PRIMARY KEY (`id_control`),
  CONSTRAINT `fk_producto_control_productos1`
    FOREIGN KEY (`id_productos`)

```

```
REFERENCES `inventario`.`productos` (`id_productos`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----
-- Table `inventario`.`historial_movimiento`
-- -----

CREATE TABLE IF NOT EXISTS `inventario`.`historial_movimiento` (
  `id_movimiento` INT NOT NULL AUTO_INCREMENT,
  `fecha_movimiento` TIMESTAMP NOT NULL,
  `cantidad` INT NOT NULL,
  `tipo_movimiento` VARCHAR(200) NOT NULL,
  `usuario` VARCHAR(100) NOT NULL,
  `id_productos` INT NOT NULL,
  PRIMARY KEY (`id_movimiento`),
  CONSTRAINT `fk_historial_movimiento_productos1`
    FOREIGN KEY (`id_productos`)
      REFERENCES `inventario`.`productos` (`id_productos`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

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