

ANEXO I – CODIGO PARA CREACION EN LA BASE DE DATOS

```
CREATE DATABASE tarea1;
```

-- Creación de Entidades

```
CREATE TABLE empleados (ID_empleados int PRIMARY KEY,  
nombre VARCHAR (45), cargo VARCHAR (20), departamento VARCHAR (40)  
);  
  
CREATE TABLE clientes (ID_cliente int PRIMARY KEY,  
nombre VARCHAR (45), ubicacion VARCHAR (200), genero VARCHAR (10), edad INT  
);  
  
CREATE TABLE promocion (ID_promocion int PRIMARY KEY,  
nombre VARCHAR (45), fecha_inicio DATE, fecha_fin DATE, descuento NUMERIC (3,2), medio  
VARCHAR (20)  
);  
  
--Instead of DECIMAL (1,2)  
  
CREATE TABLE canal_ventas (ID_canal_ventas int PRIMARY KEY,  
nombre VARCHAR (45), descripcion VARCHAR (50)  
);  
  
CREATE TABLE tienda (ID_tienda int PRIMARY KEY,  
nombre VARCHAR (45), ubicacion VARCHAR (50), ID_canal_ventas int, FOREIGN KEY  
(ID_canal_ventas) references canal_ventas (ID_canal_ventas)  
);  
  
CREATE TABLE categoria_producto (ID_categoria_producto int PRIMARY KEY,  
nombre VARCHAR (45), descripcion TEXT  
);  
  
CREATE TABLE proveedores (ID_proveedor int PRIMARY KEY,  
nombre VARCHAR (45), telefono VARCHAR (50), ubicacion VARCHAR (50)  
);
```

```

CREATE TABLE producto (ID_producto int PRIMARY KEY,
nombre VARCHAR (45), descripcion VARCHAR (50), marca VARCHAR (15), costo_de_adquisicion
DECIMAL (6,2),
precio_de_venta DECIMAL (7,2), ID_categoria_producto int, FOREIGN KEY
(ID_categoria_producto) references categoria_producto(ID_categoria_producto),
ID_proveedor int, FOREIGN KEY (ID_proveedor)
references proveedores (ID_proveedor)
);

```

--Creacion de Tablas de Hechos

```

CREATE TABLE fact_ventas (ID_venta int PRIMARY KEY,
fecha DATE,
hora TIME,
monto_total DECIMAL (9,2),
metodo_de_pago VARCHAR (50),
ID_producto int, FOREIGN KEY (ID_producto)
references producto (ID_producto),
ID_empleados int, FOREIGN KEY (ID_empleados)
references empleados (ID_empleados),
ID_cliente int, FOREIGN KEY (ID_cliente)
references clientes (ID_cliente),
ID_promocion int, FOREIGN KEY (ID_promocion)
references promocion (ID_promocion),
ID_tienda int, FOREIGN KEY (ID_tienda)
references tienda (ID_tienda)
);

```

```

CREATE TABLE fact_inventario (ID_inventario int PRIMARY KEY,
fecha_actualizacion DATE,
stock_sistema int,
stock_fisico int,

```

```

                                estado_producto VARCHAR (15),
                                ID_producto int, FOREIGN KEY
(ID_producto) references producto (ID_producto),

                                ID_empleados int, FOREIGN KEY
(ID_empleados) references empleados (ID_empleados),

                                ID_tienda int, FOREIGN KEY (ID_tienda)
references tienda (ID_tienda)
);

```

--Nota: Algunas tablas en el código original fueron modificadas con el siguiente código para que su formato fuera acorde con la data dummy.

-- Este código fue utilizado para modificar el tipo de datos de las columnas. Sin embargo, en el código anterior se realizan las modificaciones respectivas desde el principio.

```
ALTER TABLE promocion
```

```
ALTER COLUMN descuento TYPE numeric (3,2);
```

```
ALTER TABLE empleados
```

```
ALTER COLUMN departamento TYPE VARCHAR (40);
```

```
ALTER TABLE categoria_producto
```

```
ALTER COLUMN descripcion TYPE TEXT;
```

```
ALTER TABLE fact_ventas
```

```
ALTER COLUMN metodo_de_pago TYPE VARCHAR(50);
```

```
ALTER TABLE fact_inventario
```

```
ALTER COLUMN estado_producto TYPE VARCHAR(15);
```

ANEXO II – CODIGO PARA CREACION DE LAS VISTAS

-- Crea la vista para análisis de promociones

```
CREATE VIEW vista_promociones AS

SELECT fv.id_venta,
       fv.fecha,
       fv.monto_total,
       prom.nombre AS promo,
       prom.descuento,
       prom.fecha_inicio AS inicio_promo,
       prom.fecha_fin AS fin_promo,
       prom.medio,
       cl.nombre AS cliente,
       cl.ubicacion,
       cl.genero,
       cl.edad,
       pr.nombre AS producto,
       cp.nombre AS categoria_producto,
       pr.precio_de_venta
FROM FACT_VENTAS fv,
     CLIENTES cl,
     TIENDA td,
     PRODUCTO pr,
     CATEGORIA_PRODUCTO cp,
     PROMOCION prom
WHERE fv.id_cliente = cl.id_cliente
AND   fv.id_tienda = td.id_tienda
AND   fv.id_producto = pr.id_producto
AND   pr.id_categoria_producto=cp.id_categoria_producto
AND   fv.id_promocion = prom.id_promocion;
```

--Crea vista para análisis de ventas

CREATE VIEW ventas AS

SELECT fv.id_venta,

fv.fecha,

fv.monto_total,

fv.metodo_de_pago,

cl.nombre AS cliente,

cl.ubicacion,

cl.genero,

cl.edad,

pr.nombre AS producto,

pr.precio_de_venta,

td.nombre AS nombre_tienda,

td.ubicacion AS ubicacion_tienda,

cv.nombre AS canal_venta

FROM FACT_VENTAS fv,

CLIENTES cl,

TIENDA td,

PRODUCTO pr,

CANAL_VENTAS cv

WHERE fv.id_cliente = cl.id_cliente

AND fv.id_tienda = td.id_tienda

AND fv.id_producto = pr.id_producto

AND td.id_canal_ventas = cv.id_canal_ventas;

--Crea vista para análisis del inventario

```
CREATE VIEW vista_inventario AS

SELECT fv.id_venta,
       fv.fecha,
       td.nombre AS tienda,
       pr.id_producto,
       pr.nombre AS producto,
       cp.nombre AS categoria_producto,
       prov.nombre AS proveedor,
       fi.fecha_actualizacion AS fecha_ultimo_conteo,
       fi.stock_sistema,
       fi.stock_fisico,
       fi.estado_producto,
       pr.precio_de_venta,
       pr.costo_de_adquisicion
FROM fact_inventario fi,
     fact_ventas fv,
     tienda td,
     producto pr,
     categoria_producto cp,
     proveedores prov
WHERE fi.id_inventario = fv.id_producto
AND fi.id_tienda = td.id_tienda
AND fi.id_producto = pr.id_producto
AND pr.id_categoria_producto = cp.id_categoria_producto
AND pr.id_proveedor = prov.id_proveedor;
```

ANEXO III – CODIGO PARA LA EJECUCION DE CONSULTAS Y ANALISIS DE DATOS

/*La compañía fue fundada en el mes de enero de 2023, iniciando operaciones en esa fecha*/

--ANÁLISIS DE DATOS DE VENTAS

-- Que producto se vende más?

```
SELECT PRODUCTO,  
COUNT(*) AS cantidad FROM ventas  
GROUP BY producto  
ORDER BY COUNT(*) DESC;
```

/*El producto que más se vende es 'Edición Limitada' con 12 unidades vendidas en total desde el inicio de operaciones */

-- Qué tipo de clientes les vendemos?

```
SELECT ubicacion,  
COUNT(*) AS cantidad FROM ventas  
GROUP BY ubicacion  
ORDER BY COUNT(*) DESC;
```

/*Los clientes que más compran por ubicación son los de Monterrey */

```
SELECT genero,  
COUNT(*) AS cantidad FROM ventas  
GROUP BY genero  
ORDER BY COUNT(*) DESC;
```

/*Los clientes que más compran son del género masculino */

```
SELECT edad,  
COUNT(*) AS cantidad FROM ventas  
GROUP BY edad
```

```
ORDER BY COUNT(*) DESC;
```

```
/*Los clientes que más compran son de 35 años de edad */
```

--Resumen de ventas por canal de venta, se observa que las tiendas físicas tiene la mayoría de ventas

```
SELECT canal_venta,  
       SUM(monto_total) AS suma_total  
FROM ventas  
GROUP BY canal_venta  
ORDER BY suma_total DESC;
```

--Resumen de ventas por ubicacion y nombre de tienda:

```
SELECT ubicacion_tienda,  
       nombre_tienda,  
       SUM(monto_total) AS suma_total  
FROM ventas  
GROUP BY ubicacion_tienda, nombre_tienda  
ORDER BY suma_total DESC;
```

--TENDENCIAS Y PATRONES DE COMPRA

-- Tendencia: ¿Qué tipo de producto se vende más?

```
SELECT vi.categoria_producto,  
       COUNT (v.*)  
FROM VENTAS V  
JOIN vista_inventario vi ON v.id_venta=vi.id_venta  
group by vi.categoria_producto  
ORDER BY COUNT(v.*) DESC;
```

```
/*Los productos que más se han vendido son los de categoría tecnología en la historia de la compañía*/
```


-- Patrón: ¿En qué mes se vendió más de la categoría Tecnología?

```
SELECT vi.categoria_producto,EXTRACT(MONTH FROM v.fecha) AS mes,  
COUNT (v.*)  
FROM VENTAS V  
JOIN vista_inventario vi ON v.id_venta=vi.id_venta  
WHERE vi.categoria_producto='Tecnologia'  
group by vi.categoria_producto, mes  
ORDER BY COUNT(v.*) DESC;
```

*/*El mes en el que más se venden productos de tecnología es el mes de Junio (mes 6) */*

-- Patrón: Mes que más se ha vendido en todas las categorías

```
SELECT EXTRACT(MONTH FROM fecha) AS mes,  
COUNT (*)  
FROM VENTAS  
group by mes  
ORDER BY COUNT(*) DESC;
```

*/*Nuestro negocio no muestra un patrón de ventas estacionales ni de temporada al considerar todos los productos*/*

--CAMPAÑAS PROMOCIONALES

-- Impacto bruto y global de las campañas sobre el total de ventas.

```
SELECT SUM(v.monto_total)-SUM(pro.monto_total) AS venta_sin_promo,  
SUM(pro.monto_total) AS ventas_promo
```

```
FROM ventas v
```

```
LEFT JOIN vista_promociones pro ON v.id_venta=pro.id_venta;
```

-- que promociones han tenido mayor impacto?

```
SELECT promo, SUM(monto_total) AS venta_total  
FROM vista_promociones  
GROUP BY promo
```

```
ORDER BY venta_total DESC;

/*Venta total por promocion*/
```

--Suma de ventas por promoción, fechas de promoción y medio

```
SELECT promo AS nombre_promocion,
        inicio_promo, fin_promo,
        medio,
        SUM(monto_total) AS suma_monto_total
FROM vista_promociones
GROUP BY promo, inicio_promo, fin_promo, medio
ORDER BY suma_monto_total DESC;
```

--INVENTARIOS

--Estado del inventario por categoria de producto para el total del inventario en stock

```
SELECT categoria_producto,
        estado_producto,
        SUM(stock_sistema) AS total_stock_sistema,
        SUM(stock_fisico) AS total_stock_fisico,
        SUM(costo_de_adquisicion) AS costo_total,
        SUM (precio_de_venta) AS precio_total
FROM vista_inventario
GROUP BY categoria_producto, estado_producto
ORDER BY categoria_producto DESC;
```

--Costo total de los productos en el inventario por categoría:

```
SELECT categoria_producto,
        SUM(costo_de_adquisicion) AS costo_total
FROM vista_inventario
GROUP BY categoria_producto
ORDER BY SUM(costo_de_adquisicion) DESC;
```

--Costo total de los productos en los inventarios por nombre de proveedor y categoría de producto:

```
SELECT proveedor,  
       categoria_producto,  
       SUM(costo_de_adquisicion) AS costo_total  
FROM vista_inventario  
GROUP BY proveedor, categoria_producto  
ORDER BY SUM(costo_de_adquisicion) DESC;
```