

INFORME DE NEGOCIO: Fast Food

Nombre del autor: Zambón Enzo Agustín

Email: eazambón97@gmail.com

Cohorte: FT08

Fecha de entrega: 10/10/2024

Institución: Fast Food

Módulo 2

INDICE

Introducción 4

Desarrollo del proyecto..... 5

Resultados y consultas..... 20

Hallazgos clave..... 25

Recomendaciones estratégicas 26

Optimización y sostenibilidad 27

Desafíos y soluciones..... 29

Reflexión personal..... 30

EXTRA CREDIT 31

Fast Food

La historia de "Fast Food" comienza en el corazón de una pequeña ciudad, en la Sucursal Central, ubicada en "1234 Main St" en el año 2018. Desde inicios, el menú ofrecía una selección limitada de categorías de productos que incluían lo básico de la comida rápida: hamburguesas, pizzas, postres y bebidas. Con el paso del tiempo, la empresa fue expandiendo su menú, incorporando opciones más saludables como ensaladas y comidas veganas, lo que atrajo una nueva base de clientes.

Los productos estrella, como la Hamburguesa Deluxe y la Pizza Pepperoni, rápidamente ganaron popularidad, lo que impulsó las ventas. Gracias a un equipo dedicado de empleados, liderado por el gerente John Doe y un dinámico equipo de cocineros, cajeros y repartidores, la Sucursal Central se convirtió en el punto neurálgico de la empresa. Sin embargo, la verdadera clave del éxito fue la implementación de una eficaz red de mensajeros, quienes, tanto internos como externos, aseguraron que los pedidos llegaran puntuales a las puertas de sus clientes.

La expansión no tardó en llegar y para el 2024, la empresa había abierto sucursales en todas las áreas de la ciudad, como la Sucursal Norte, la Sucursal Este, y la popular Sucursal Playa, que se convirtió en el lugar preferido de los turistas.

Con el incremento constante en ventas y la expansión acelerada de la empresa, los directivos notaron que optimizar los procesos y mejorar la eficiencia operativa era esencial para mantener el crecimiento. Los informes de ventas y los datos recogidos de las sucursales eran vastos, pero no estaban siendo explotados al máximo. Para ello, decidieron contratar a un **data analyst**.



Introducción

El proyecto tiene como objetivo diseñar y desarrollar una base de datos relacional que permita gestionar de manera eficiente la información de “FastFood”, facilitando el acceso a datos clave sobre productos, clientes, órdenes y empleados. A través de la implementación de un esquema de base de datos bien estructurado, se busca optimizar las operaciones del negocio y mejorar la toma de decisiones estratégicas basadas en análisis de datos.

Objetivos Organizacionales Alcanzados:

- Se creó un modelo relacional que identifica las tablas y sus relaciones, logrando una estructura normalizada que minimiza la redundancia de datos. Además, se establecieron siete tablas independientes (Categorías, Sucursales, Empleados, Clientes, Mensajeros, Orígenes y Tipos de Pago) y tres tablas dependientes (Órdenes, Detalle de la Orden y Productos).
- Las tablas fueron llenadas con datos relevantes, permitiendo la manipulación de información a través de sentencias SQL (INSERT, UPDATE, DELETE). Se realizó un seguimiento exhaustivo para asegurar que las tablas fueron correctamente pobladas y mantenidas.
- Se realizaron múltiples consultas, tanto básicas como complejas, para extraer información valiosa, incluyendo análisis de ventas, rendimiento de sucursales y eficiencia de empleados.
- La información analizada a partir de las consultas permitió a la organización identificar áreas de mejora, como la optimización de la logística de entrega, el aumento de ingresos a través de la aplicación móvil y la capacitación del personal.

En resumen, el desarrollo de la base de datos relacional ha proporcionado a la organización una herramienta poderosa para la gestión y análisis de información, permitiendo una mejor planificación y ejecución de estrategias operativas. La estructura creada no solo logró facilitar el manejo eficiente de los datos, sino que también promueve la sostenibilidad y escalabilidad del sistema en el futuro.

Desarrollo del proyecto

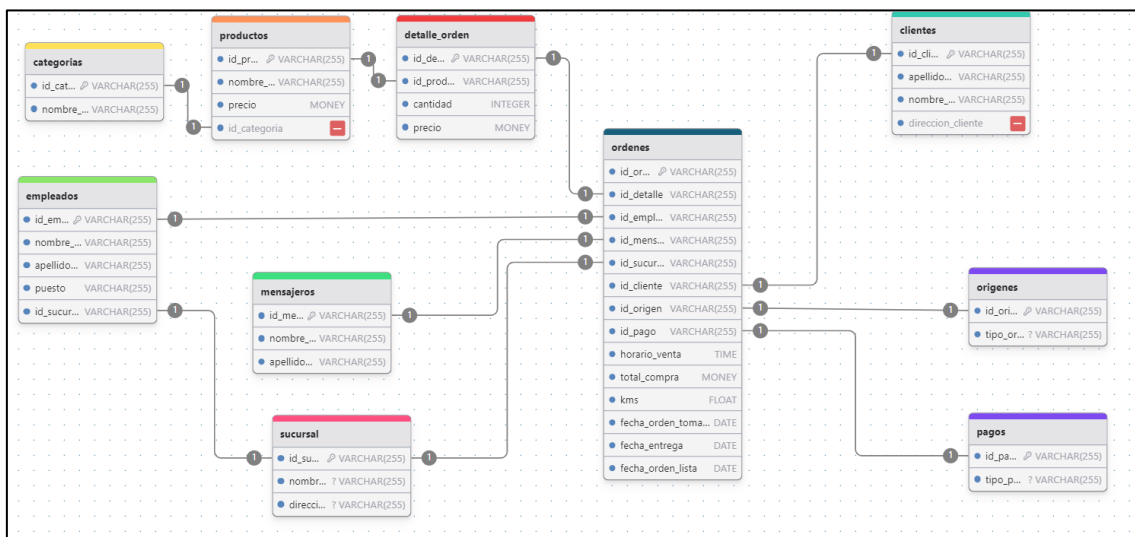
Avance 1: Diseño del esquema de la base de datos

En primer lugar, es útil pensar en cómo se vería nuestro esquema de relaciones dentro de la base de datos que vamos a estructurar. Para ello, diseñé un modelo relacional de acuerdo con la información suministrada por el Modelo Semántico en la capsula de Henry, identificando las posibles relaciones entre las tablas a partir de elementos en común.

En dicho modelo fue posible identificar siete tablas independientes (aquellas que no dependen de otras tablas para su existencia, es decir, no tienen claves foráneas hacia otras tablas): Categorías, Sucursales, Empleados, Clientes, Mensajeros, Orígenes y Tipos de Pago. A su vez, se identifican también tres tablas dependientes: Ordenes, Detalle de la Orden y Productos.

Cabe aclarar que la tabla de Detalle de la Orden no estaba mencionada en el enunciado, sin embargo, fue necesario incorporarla para incluir un detalle más preciso acerca de los pedidos realizados por los clientes, ya que existe la posibilidad de que estos demanden más de un producto o diversas cantidades del mismo, con lo cual resulta necesario agregar esta información al esquema, así como también un importe total (en \$) de toda la operación.

Ilustración 1: Modelo Relacional (Esquema Imaginado)



Módulo 2

Luego de haber diseñado mi esquema, me dispuse construir la base de datos en SQL server, aplicando las siguientes sentencias:

1. Creación y selección de la base de datos a utilizar:

```
CREATE DATABASE Fast_Food_DB
USE Fast_Food_DB
```

2. Creación de las tablas con sus respectivas columnas y relaciones (a partir del uso de claves foráneas):

```
-- Crear Tabla Categorías
CREATE TABLE Categorías(
  Id_categoria INT IDENTITY (1,1) NOT NULL PRIMARY KEY,
  Categoría VARCHAR (100));

-- Crear Tabla sobre los Productos
CREATE TABLE Productos (
  Id_producto INT IDENTITY (1,1) NOT NULL PRIMARY KEY,
  Producto VARCHAR (100),
  Id_categoria INT FOREIGN KEY (Id_categoria) REFERENCES Categorías (Id_categoria),
  Precio DECIMAL);

-- Crear Tabla sobre las Sucursales
CREATE TABLE Sucursales (
  Id_sucursal INT IDENTITY (1,1) NOT NULL PRIMARY KEY,
  Sucursal VARCHAR (100),
  Dirección VARCHAR (200));

-- Crear Tabla sobre los Empleados
CREATE TABLE Empleados(
  Id_empleado INT IDENTITY (1,1) NOT NULL PRIMARY KEY,
  Nombre_empleado VARCHAR (100),
  Posición VARCHAR (50),
  Departamento VARCHAR (50),
  Id_sucursal INT FOREIGN KEY (Id_sucursal) REFERENCES Sucursales (Id_sucursal),
  Rol VARCHAR (50),);

-- Crear Tabla sobre los Clientes
CREATE TABLE Clientes(
  Id_cliente INT IDENTITY (1,1) NOT NULL PRIMARY KEY,
  Nombre_cliente VARCHAR (100),
  Dirección VARCHAR (100));
```

```
-- Crear Tabla de Orígenes del Pedido
CREATE TABLE Orígenes(
  Id_origen INT IDENTITY (1,1) NOT NULL PRIMARY KEY,
  Origen VARCHAR (50));

-- Crear Tabla de Tipos de Pago
CREATE TABLE Tipos_Pago(
  Id_pago INT IDENTITY (1,1) NOT NULL PRIMARY KEY,
  Descripción VARCHAR (50));
```

Módulo 2

```
-- Crear Tabla de Mensajeros
CREATE TABLE Mensajeros(
  Id_mensajero INT IDENTITY (1,1) NOT NULL PRIMARY KEY,
  Nombre_mensajero VARCHAR (100),
  EsExterno BIT NOT NULL); --1 si es externo , 0 si es empleado

-- Crear Tabla de Ordenes de Pedido
CREATE TABLE Ordenes(
  Id_orden INT IDENTITY (1,1) PRIMARY KEY,
  Id_cliente INT FOREIGN KEY (Id_cliente) REFERENCES Clientes (Id_cliente),
  Id_empleado INT FOREIGN KEY (Id_empleado) REFERENCES Empleados (Id_empleado),
  Id_sucursal INT FOREIGN KEY (Id_sucursal) REFERENCES Sucursales (Id_sucursal),
  Id_mensajero INT FOREIGN KEY (Id_mensajero) REFERENCES Mensajeros (Id_mensajero),
  Id_origen INT FOREIGN KEY (Id_origen) REFERENCES Origenes (Id_origen),
  Id_pago INT FOREIGN KEY (Id_pago) REFERENCES Tipos_Pago (Id_pago),
  Horario_venta VARCHAR (20),
  Total_compra DECIMAL (10,2),
  Kms_recorrer DECIMAL (10,2),
  Fecha_despacho DATETIME,
  Fecha_entrega DATETIME,
  Fecha_orden_tomada DATETIME,
  Fecha_orden_lista DATETIME);

-- Crear Tabla Detalle de las Ordenes
CREATE TABLE Detalle_Ordenes(
  Id_detalle INT IDENTITY (1,1) NOT NULL PRIMARY KEY,
  Id_orden INT FOREIGN KEY (Id_orden) REFERENCES Ordenes (Id_orden),
  Id_producto INT FOREIGN KEY (Id_producto) REFERENCES Productos (Id_producto),
  Cantidad INT NOT NULL,
  Precio DECIMAL (10,2) NOT NULL);
```

Donde:

- INT: Indica que el tipo de dato es un número entero.
- IDENTITY (1,1): Significa que la primera fila tendrá el valor 1 y cada fila subsiguiente incrementará este valor en 1.
- NOT NULL: Esta restricción se aplica a una columna para asegurar que no puede contener valores nulos.
- PRIMARY KEY: Esta restricción se usa para identificar de manera única cada fila en una tabla.
- VARCHAR: Es un tipo de dato que se utiliza para almacenar cadenas de texto de longitud variable.
- FOREIGN KEY: Esta es una columna que establece una relación con otra tabla.
- REFERENCES: Se utiliza en la definición de una clave foránea para indicar a qué tabla y columna se está haciendo referencia.
- DECIMAL: Es un tipo de dato que se utiliza para almacenar números decimales con precisión.
- DATETIME: Es un tipo de dato que se utiliza para almacenar fechas y horas.

Módulo 2

Avance 2: Población de tablas mediante queries

El siguiente avance consistió en llenar las tablas previamente creadas con datos y gestionar dicha información utilizando sentencias DML.

En primer lugar, utilice INSERT para agregar los datos disponibles en la lección 4 del Módulo 2 de Henry a las tablas en forma específica, asegurándome de que los nombres de las columnas sean los correctos y que el orden de los datos sea el adecuado:

```
-- Insertar datos en Categorías
INSERT INTO Categorías (Categoría) VALUES
('Comida Rápida'), ('Postres'), ('Bebidas'), ('Ensaladas'), ('Desayunos'),
('Cafetería'), ('Helados'), ('Comida Vegana'), ('Carnes'), ('Pizzas');

-- Insertar datos en Productos
INSERT INTO Productos (Producto, Id_categoria, Precio) VALUES
('Hamburguesa Deluxe', 1, 9.99), ('Cheeseburger', 1, 7.99), ('Pizza Margarita', 10, 11.99), ('Pizza Pepperoni', 10, 12.99), ('Helado de Chocolate', 7, 2.99),
('Helado de Vainilla', 7, 2.99), ('Ensalada César', 4, 5.99), ('Ensalada Griega', 4, 6.99), ('Pastel de Zanahoria', 2, 3.99), ('Brownie', 2, 2.99);

-- Insertar datos en Sucursales
INSERT INTO Sucursales (Sucursal, Direccion) VALUES
('Sucursal Central', '1234 Main St'), ('Sucursal Norte', '5678 North St'), ('Sucursal Este', '9101 East St'), ('Sucursal Oeste', '1121 West St'), ('Sucursal Sur', '1234 South St'),
('Sucursal Playa', '1516 Beach St'), ('Sucursal Montaña', '1718 Mountain St'), ('Sucursal Valle', '1920 Valley St'), ('Sucursal Lago', '2122 Lake St'), ('Sucursal Bosque', '2324 Forest St');

-- Insertar datos en Empleados
INSERT INTO Empleados (Nombre_empleado, Posicion, Departamento, Id_sucursal, Rol) VALUES
('John Doe', 'Gerente', 'Administración', 1, 'Vendedor'), ('Jane Smith', 'Subgerente', 'Ventas', 1, 'Vendedor'), ('Bill Jones', 'Cajero', 'Ventas', 1, 'Vendedor'),
('Emma Davis', 'Repartidor', 'Logística', 1, 'Mensajero'), ('Lucas Miller', 'Atención al Cliente', 'Servicio', 1, 'Vendedor'), ('Olivia García', 'Encargado de Turno', 'Ventas', 1, 'Vendedor');

-- Insertar datos en Clientes
INSERT INTO Clientes (Nombre_cliente, Direccion) VALUES
('Cliente Uno', '1000 A Street'), ('Cliente Dos', '1001 B Street'), ('Cliente Tres', '1002 C Street'), ('Cliente Cuatro', '1003 D Street'), ('Cliente Cinco', '1004 E Street'),
('Cliente Seis', '1005 F Street'), ('Cliente Siete', '1006 G Street'), ('Cliente Ocho', '1007 H Street'), ('Cliente Nueve', '1008 I Street'), ('Cliente Diez', '1009 J Street');

-- Insertar datos en OrigenesOrden
INSERT INTO Origenes (Origen) VALUES
('En línea'), ('Presencial'), ('Teléfono'), ('Drive Thru'), ('App Móvil'),
('Redes Sociales'), ('Correo Electrónico'), ('Publicidad'), ('Recomendación'), ('Evento');
```

```
-- Insertar datos en TiposPago
INSERT INTO TiposPago (Descripción) VALUES
('Efectivo'), ('Tarjeta de Crédito'), ('Tarjeta de Débito'), ('PayPal'), ('Transferencia Bancaria'),
('Criptomonedas'), ('Cheque'), ('Vale de Comida'), ('Cupón de Descuento'), ('Pago Móvil');

-- Insertar datos en Mensajeros
INSERT INTO Mensajeros (Nombre_mensajero, EsExterno) VALUES
('Mensajero Uno', 0), ('Mensajero Dos', 1), ('Mensajero Tres', 0), ('Mensajero Cuatro', 1), ('Mensajero Cinco', 0),
('Mensajero Seis', 1), ('Mensajero Siete', 0), ('Mensajero Ocho', 1), ('Mensajero Nueve', 0), ('Mensajero Diez', 1);

INSERT INTO Ordenes (Id_cliente, Id_empleado, Id_sucursal, Id_mensajero, Id_pago, Id_origen, Horario_venta, Total_compra, Kms_recorrer, Fecha_despacho, Fecha_entrega)
VALUES
(1, 1, 1, 1, 1, 'Mañana', 1053.51, 5.5, '2023-01-10 08:30:00', '2023-01-10 09:00:00', '2023-01-10 08:00:00', '2023-01-10 08:15:00'),
(2, 2, 2, 2, 2, 'Tarde', 1075.00, 10.0, '2023-02-15 14:30:00', '2023-02-15 15:00:00', '2023-02-15 13:30:00', '2023-02-15 14:00:00'),
(3, 3, 3, 3, 3, 'Noche', 920.00, 2.0, '2023-03-20 19:30:00', '2023-03-20 20:00:00', '2023-03-20 19:00:00', '2023-03-20 19:15:00'),
(4, 4, 4, 4, 4, 'Mañana', 930.00, 0.5, '2023-04-25 09:30:00', '2023-04-25 10:00:00', '2023-04-25 09:00:00', '2023-04-25 09:15:00'),
(5, 5, 5, 5, 5, 'Tarde', 955.00, 8.0, '2023-05-30 15:30:00', '2023-05-30 16:00:00', '2023-05-30 15:00:00', '2023-05-30 15:15:00'),
(6, 6, 6, 6, 1, 'Noche', 945.00, 12.5, '2023-06-05 20:30:00', '2023-06-05 21:00:00', '2023-06-05 20:00:00', '2023-06-05 20:15:00'),
(7, 7, 7, 7, 2, 'Mañana', 1065.00, 7.5, '2023-07-10 08:30:00', '2023-07-10 09:00:00', '2023-07-10 08:00:00', '2023-07-10 08:15:00'),
(8, 8, 8, 8, 3, 'Tarde', 1085.00, 9.5, '2023-08-15 14:30:00', '2023-08-15 15:00:00', '2023-08-15 14:00:00', '2023-08-15 14:15:00'),
(9, 9, 9, 9, 4, 'Noche', 1095.00, 3.0, '2023-09-20 19:30:00', '2023-09-20 20:00:00', '2023-09-20 19:00:00', '2023-09-20 19:15:00'),
(10, 10, 10, 10, 5, 'Mañana', 900.00, 15.0, '2023-10-25 09:30:00', '2023-10-25 10:00:00', '2023-10-25 09:00:00', '2023-10-25 09:15:00');

-- Insertar datos en DetalleOrdenes
INSERT INTO DetalleOrdenes (Id_orden, Id_producto, Cantidad, Precio) VALUES
(1, 1, 3, 23.44),
(1, 2, 5, 45.14),
(1, 3, 4, 46.37),
(1, 4, 4, 42.34),
(1, 5, 1, 18.25),
(1, 6, 4, 20.08),
(1, 7, 2, 13.31),
(1, 8, 2, 20.96),
(1, 9, 4, 30.13),
(1, 10, 3, 38.34);
```

Luego, me aseguré de que las tablas hayan quedado completamente pobladas mediante la sentencia SELECT * FROM (nombre de tabla), como se observa a continuación:

SELECT * FROM Categorías y SELECT * FROM Productos:

Módulo 2

Id_categoria	Categoria	Id_producto	Producto	Id_categoria	Precio
1	Comida Rápida	1	Hamburguesa Deluxe	1	10
2	Postres	2	Cheeseburger	1	8
3	Bebidas	3	Pizza Margarita	10	12
4	Ensaladas	4	Pizza Pepperoni	10	13
5	Desayunos	5	Helado de Chocolate	7	3
6	Cafetería	6	Helado de Vainilla	7	3
7	Helados	7	Ensalada César	4	6
8	Comida Vegana	8	Ensalada Griega	4	7
9	Carnes	9	Pastel de Zanahoria	2	4
10	Pizzas	10	Brownie	2	3

SELECT * FROM Sucursales:

	Id_sucursal	Sucursal	Direccion
1	1	Sucursal Central	1234 Main St
2	2	Sucursal Norte	5678 North St
3	3	Sucursal Este	9101 East St
4	4	Sucursal Oeste	1121 West St
5	5	Sucursal Sur	3141 South St
6	6	Sucursal Playa	1516 Beach St
7	7	Sucursal Montaña	1718 Mountain St
8	8	Sucursal Valle	1920 Valley St
9	9	Sucursal Lago	2122 Lake St
10	10	Sucursal Bosque	2324 Forest St

SELECT * FROM Empleados:

	Id_empleado	Nombre_empleado	Posicion	Departamento	Id_sucursal	Rol
1	1	John Doe	Gerente	Administración	1	Vendedor
2	2	Jane Smith	Subgerente	Ventas	1	Vendedor
3	3	Bill Jones	Cajero	Ventas	1	Vendedor
4	4	Alice Johnson	Cocinero	Cocina	1	Vendedor
5	5	Tom Brown	Barista	Cafetería	1	Vendedor
6	6	Emma Davis	Repartidor	Logística	1	Mensajero
7	7	Lucas Miller	Atención al Cliente	Servicio	1	Vendedor
8	8	Olivia García	Encargado de Turno	Administración	1	Vendedor
9	9	Ethan Martinez	Mesero	Restaurante	1	Vendedor
10	10	Sophia Rodriguez	Auxiliar de Limpieza	Mantenimiento	1	Vendedor

SELECT * FROM Clientes:

	Id_cliente	Nombre_cliente	Direccion		Id_origen	Origen
1	1	Cliente Uno	1000 A Street	1	1	En línea
2	2	Cliente Dos	1001 B Street	2	2	Presencial
3	3	Cliente Tres	1002 C Street	3	3	Teléfono
4	4	Cliente Cuatro	1003 D Street	4	4	Drive Thru
5	5	Cliente Cinco	1004 E Street	5	5	App Móvil
6	6	Cliente Seis	1005 F Street	6	6	Redes Sociales
7	7	Cliente Siete	1006 G Street	7	7	Correo Electrónico
8	8	Cliente Ocho	1007 H Street	8	8	Publicidad
9	9	Cliente Nueve	1008 I Street	9	9	Recomendación
10	10	Cliente Diez	1009 J Street	10	10	Evento

Módulo 2

SELECT * FROM Tipos_Pago y SELECT * FROM Mensajeros:

	id_pago	Descripcion		Id_mensajero	Nombre_mensajero	EsExterno
1	1	Efectivo	1	1	Mensajero Uno	0
2	2	Tarjeta de Crédito	2	2	Mensajero Dos	1
3	3	Tarjeta de Débito	3	3	Mensajero Tres	0
4	4	PayPal	4	4	Mensajero Cuatro	1
5	5	Transferencia Bancaria	5	5	Mensajero Cinco	0
6	6	Criptomonedas	6	6	Mensajero Seis	1
7	7	Cheque	7	7	Mensajero Siete	0
8	8	Vale de Comida	8	8	Mensajero Ocho	1
9	9	Cupón de Descuento	9	9	Mensajero Nueve	0
10	10	Pago Móvil	10	10	Mensajero Diez	1

SELECT * FROM Ordenes:

Id_orden	Id_cliente	Id_empleado	Id_sucursal	Id_mensajero	Id_origen	Id_pago	Horario_venta	Total_compra	Kms_recorrer	Fecha_despacho	Fecha_entrega	Fecha_orden_tomada	Fecha_orden_lista
1	1	1	1	1	1	1	Mañana	1053.51	5.50	2023-01-10 08:30:00.000	2023-01-10 09:00:00.000	2023-01-10 08:00:00.000	2023-01-10 08:15:00.000
2	2	2	2	2	2	2	Tarde	1075.00	10.00	2023-02-15 14:30:00.000	2023-02-15 15:00:00.000	2023-02-15 13:30:00.000	2023-02-15 14:00:00.000
3	3	3	3	3	3	3	Noche	920.00	2.00	2023-03-20 19:30:00.000	2023-03-20 20:00:00.000	2023-03-20 19:00:00.000	2023-03-20 19:15:00.000
4	4	4	4	4	4	4	Mañana	930.00	0.50	2023-04-25 09:30:00.000	2023-04-25 10:00:00.000	2023-04-25 09:00:00.000	2023-04-25 09:15:00.000
5	5	5	5	5	5	5	Tarde	955.00	8.00	2023-05-30 15:30:00.000	2023-05-30 16:00:00.000	2023-05-30 15:00:00.000	2023-05-30 15:15:00.000
6	6	6	6	6	1	6	Noche	945.00	12.50	2023-06-05 20:30:00.000	2023-06-05 21:00:00.000	2023-06-05 20:00:00.000	2023-06-05 20:15:00.000
7	7	7	7	7	2	7	Mañana	1065.00	7.50	2023-07-10 08:30:00.000	2023-07-10 09:00:00.000	2023-07-10 08:00:00.000	2023-07-10 08:15:00.000
8	8	8	8	8	3	8	Tarde	1085.00	9.50	2023-08-15 14:30:00.000	2023-08-15 15:00:00.000	2023-08-15 14:00:00.000	2023-08-15 14:15:00.000
9	9	9	9	9	4	9	Noche	1095.00	3.00	2023-09-20 19:30:00.000	2023-09-20 20:00:00.000	2023-09-20 19:00:00.000	2023-09-20 19:15:00.000
10	10	10	10	10	5	10	Mañana	900.00	15.00	2023-10-25 09:30:00.000	2023-10-25 10:00:00.000	2023-10-25 09:00:00.000	2023-10-25 09:15:00.000

SELECT * FROM Detalle_Ordenes:

	Id_detalle	Id_orden	Id_producto	Cantidad	Precio
1	1	1	1	3	23.44
2	2	1	2	5	45.14
3	3	1	3	4	46.37
4	4	1	4	4	42.34
5	5	1	5	1	18.25
6	6	1	6	4	20.08
7	7	1	7	2	13.31
8	8	1	8	2	20.96
9	9	1	9	4	30.13
10	10	1	10	3	38.34

Posteriormente realicé modificaciones de mi información a través de las sentencias UPDATE y DELETE, las cuales también verifiqué con SELECT * FROM (nombre de tabla):

1. Ejemplos de actualizaciones con UPDATE:

```
-- Aumentar el precio de todos los productos en la categoría 1
UPDATE Productos SET Precio = Precio + 1 WHERE Id_categoria = 1;

-- Cambiar la posición de todos los empleados del departamento 'Cocina' a 'Chef'
UPDATE Empleados SET Posicion = 'Chef' WHERE Departamento = 'Cocina';

-- Actualizar la dirección de una sucursal específica
UPDATE Sucursales SET Direccion = '1234 New Address St' WHERE Id_Sucursal = 1;
```

SELECT * FROM Productos:

Módulo 2

	Id_producto	Producto	Id_categoria	Precio
1	1	Hamburguesa Deluxe	1	11
2	2	Cheeseburger	1	9
3	3	Pizza Margarita	10	12
4	4	Pizza Pepperoni	10	13
5	5	Helado de Chocolate	7	3
6	6	Helado de Vainilla	7	3
7	7	Ensalada César	4	6
8	8	Ensalada Griega	4	7
9	9	Pastel de Zanahoria	2	4
10	10	Brownie	2	3

SELECT * FROM Empleados:

	Id_empleado	Nombre_empleado	Posicion	Departamento	Id_sucursal	Rol
1	1	John Doe	Gerente	Administración	1	Vendedor
2	2	Jane Smith	Subgerente	Ventas	1	Vendedor
3	3	Bill Jones	Cajero	Ventas	1	Vendedor
4	4	Alice Johnson	Chef	Cocina	1	Vendedor
5	5	Tom Brown	Barista	Cafetería	1	Vendedor
6	6	Emma Davis	Repartidor	Logística	1	Mensajero
7	7	Lucas Miller	Atención al Cliente	Servicio	1	Vendedor
8	8	Olivia García	Encargado de Turno	Administración	1	Vendedor
9	9	Ethan Martinez	Mesero	Restaurante	1	Vendedor
10	10	Sophia Rodriguez	Auxiliar de Limpieza	Mantenimiento	1	Vendedor

SELECT * FROM Sucursales:

	Id_sucursal	Sucursal	Direccion
1	1	Sucursal Central	1234 New Address St
2	2	Sucursal Norte	5678 North St
3	3	Sucursal Este	9101 East St
4	4	Sucursal Oeste	1121 West St
5	5	Sucursal Sur	3141 South St
6	6	Sucursal Playa	1516 Beach St
7	7	Sucursal Montaña	1718 Mountain St
8	8	Sucursal Valle	1920 Valley St
9	9	Sucursal Lago	2122 Lake St
10	10	Sucursal Bosque	2324 Forest St

2. Ejemplos de eliminaciones con DELETE:

a. Eliminar una Orden y su respectivo detalle:

```
DELETE FROM Detalle_Ordenes WHERE Id_orden = 10;
DELETE FROM Ordenes WHERE Id_orden = 10;
```

SELECT * FROM Ordenes y Detalle_Ordenes:

	Id_orden	Id_cliente	Id_empleado	Id_sucursal	Id_mensajero	Id_origen	Id_tipo	Horario_venta	Total_compra	Kms_recorrer	Fecha_despacho	Fecha_entrega	Fecha_orden_tomada	Fecha_orden_lista
1	1	1	1	1	1	1	1	Mañana	1053.51	5.50	2023-01-10 08:30:00.000	2023-01-10 09:00:00.000	2023-01-10 08:00:00.000	2023-01-10 08:15:00.000
2	2	2	2	2	2	2	2	Tarde	1075.00	10.00	2023-02-15 14:30:00.000	2023-02-15 15:00:00.000	2023-02-15 13:30:00.000	2023-02-15 14:00:00.000
3	3	3	3	3	3	3	3	Noche	920.00	2.00	2023-03-20 19:30:00.000	2023-03-20 20:00:00.000	2023-03-20 19:00:00.000	2023-03-20 19:15:00.000
4	4	4	4	4	4	4	4	Mañana	930.00	0.50	2023-04-25 09:30:00.000	2023-04-25 10:00:00.000	2023-04-25 09:00:00.000	2023-04-25 09:15:00.000
5	5	5	5	5	5	5	5	Tarde	955.00	8.00	2023-05-30 15:30:00.000	2023-05-30 16:00:00.000	2023-05-30 15:00:00.000	2023-05-30 15:15:00.000
6	6	6	6	6	6	1	6	Noche	945.00	12.50	2023-06-05 20:30:00.000	2023-06-05 21:00:00.000	2023-06-05 20:00:00.000	2023-06-05 20:15:00.000
7	7	7	7	7	2	7	7	Mañana	1065.00	7.50	2023-07-10 08:30:00.000	2023-07-10 09:00:00.000	2023-07-10 08:00:00.000	2023-07-10 08:15:00.000
8	8	8	8	8	8	3	8	Tarde	1085.00	9.50	2023-08-15 14:30:00.000	2023-08-15 15:00:00.000	2023-08-15 14:00:00.000	2023-08-15 14:15:00.000
9	9	9	9	9	9	4	9	Noche	1095.00	3.00	2023-09-20 19:30:00.000	2023-09-20 20:00:00.000	2023-09-20 19:00:00.000	2023-09-20 19:15:00.000

Módulo 2

	Id_detalle	Id_orden	Id_producto	Cantidad	Precio
1	1	1	1	3	23.44
2	2	1	2	5	45.14
3	3	1	3	4	46.37
4	4	1	4	4	42.34
5	5	1	5	1	18.25
6	6	1	6	4	20.08
7	7	1	7	2	13.31
8	8	1	8	2	20.96
9	9	1	9	4	30.13

- b. Eliminar todos los productos de una categoría específica:

```
-- Eliminar los detalles de ordenes asociados a los productos de categoria 2
DELETE FROM Detalle_Ordenes
WHERE Id_producto IN (
    SELECT Id_Producto
    FROM Productos
    WHERE Id_categoria = 2);

-- Eliminar los productos asociados con la categoria 2
DELETE FROM Productos WHERE Id_categoria = 2;
```

SELECT * FROM Detalle_Ordenes y Productos:

Results		Messages			
	Id_detalle	Id_orden	Id_producto	Cantidad	Precio
1	1	1	1	3	23.44
2	2	1	2	5	45.14
3	3	1	3	4	46.37
4	4	1	4	4	42.34
5	5	1	5	1	18.25
6	6	1	6	4	20.08
7	7	1	7	2	13.31
8	8	1	8	2	20.96

Results

Messages

	Id_producto	Producto	Id_categoria	Precio
1	1	Hamburguesa Deluxe	1	11
2	2	Cheeseburger	1	9
3	3	Pizza Margarita	10	12
4	4	Pizza Pepperoni	10	13
5	5	Helado de Chocolate	7	3
6	6	Helado de Vainilla	7	3
7	7	Ensalada César	4	6
8	8	Ensalada Griega	4	7

- c. Elimina los empleados de una sucursal que cerró. También eliminé las ordenes de esta sucursal y la sucursal de la tabla de Sucursales, siendo esta última tabla la única en modificarse:

```
DELETE FROM Empleados WHERE Id_sucursal = 10;
DELETE FROM Ordenes WHERE Id_sucursal = 10;
DELETE FROM Sucursales WHERE Id_sucursal = 10;
```

SELECT * FROM Sucursales:

	Id_sucursal	Sucursal	Direccion
1	1	Sucursal Central	1234 New Address St
2	2	Sucursal Norte	5678 North St
3	3	Sucursal Este	9101 East St
4	4	Sucursal Oeste	1121 West St
5	5	Sucursal Sur	3141 South St
6	6	Sucursal Playa	1516 Beach St
7	7	Sucursal Montaña	1718 Mountain St
8	8	Sucursal Valle	1920 Valley St
9	9	Sucursal Lago	2122 Lake St

Consultas Básicas realizadas:

1. Registros únicos: ¿Cuál es la cantidad total de registros únicos en la tabla de órdenes?

```
--1 REGISTROS UNICOS:
--Pregunta: ¿Cuál es la cantidad total de registros únicos en la tabla de órdenes?
SELECT COUNT (*) AS Registros_unicos
FROM (SELECT DISTINCT * FROM Ordenes) AS Subconsulta
-- Respuesta: En total los registros únicos en la tabla órdenes (Ordenes) son 9
```

	Registros_unicos
1	9

2. Empleados por departamento: ¿Cuántos empleados existen en cada departamento?

```
--2 EMPLEADOS POR DEPARTAMENTO:
--Pregunta: ¿Cuántos empleados existen en cada departamento?
SELECT Departamento, COUNT (Id_empleado) as Cantidad_empleados
FROM Empleados
GROUP BY Departamento
--Respuesta: Existen 2 empleados en los departamentos Administración y Ventas y 1 en los demás
```

	Departamento	Cantidad_empleados
1	Administración	2
2	Cafetería	1
3	Cocina	1
4	Logística	1
5	Mantenimiento	1
6	Restaurante	1
7	Servicio	1
8	Ventas	2

3. Productos por categoría: ¿Cuántos productos hay por código de categoría?

Módulo 2

```
--3 PRODUCTOS POR CATEGORÍA:
--Pregunta: ¿Cuántos productos hay por código de categoría?
SELECT Id_categoria, COUNT (Id_categoria) as Cantidad_productos
FROM Productos
GROUP BY Id_categoria
--Respuesta: Hay 2 productos en las categorías 1,4,7 y 10 y en las demás no hay
```

	Id_categoria	Cantidad_productos
1	1	2
2	4	2
3	7	2
4	10	2

4. Importación de clientes: ¿Cuántos clientes se han importado a la tabla de clientes?

```
--4 IMPORTACIÓN DE CLIENTES:
--Pregunta: ¿Cuántos clientes se han importado a la tabla de clientes?
SELECT COUNT (Id_cliente) as Total_clientes
FROM Clientes
-- Respuesta: Se han importado 10 Clientes
```

	Total_clientes
1	10

5. Análisis de desempeño de sucursales: ¿Cuáles son las sucursales con un promedio de Facturación/Ingresos superior a 1000.00 y que minimizan sus costos en base al promedio de kilómetros recorridos de todas de sus entregas gestionadas? Para un mejor relevamiento, ordene el listado por el Promedio Km Recorridos.

```
-- Forma 1:
SELECT Id_sucursal,
       AVG(total_compra) AS Facturacion_promedio,
       AVG(Kms_recorrer) AS Kms_promedio
FROM Ordenes
GROUP BY Id_sucursal
HAVING AVG(total_compra) > 1000.00
ORDER BY Kms_promedio ASC;
-- Respuesta: Las Sucursales son 9-1-7-8-2
```

	Id_sucursal	Facturacion_promedio	Kms_promedio
1	9	1095.000000	3.000000
2	1	1053.510000	5.500000
3	7	1065.000000	7.500000
4	8	1085.000000	9.500000
5	2	1075.000000	10.000000

```
-- Forma 2:
SELECT Id_sucursal, Total_compra, Kms_recorrer
FROM Ordenes
WHERE Kms_recorrer < (SELECT AVG(Kms_recorrer) FROM Ordenes) AND Total_compra > 1000.00
ORDER BY Kms_recorrer ASC;
-- Respuestas: Las sucursales con ese nivel de eficiencia son la 9 (Sucursal Lago) y la 1 (Sucursal Central)
```

	Id_sucursal	Total_compra	Kms_recorrer
1	9	1095.00	3.00
2	1	1053.51	5.50

Módulo 2**Avance 3: Consultas con preguntas de agregación**

Utilizando funciones de agregación (SUM, AVG, MAX, MIN) es posible extraer información valiosa de la Base de Datos. De esta forma se pudieron resolver las siguientes consultas:

1. Total de ventas globales:

```
-- 1. Pregunta: ¿Cuál es el total de ventas (TotalCompra) a nivel global?  
SELECT SUM (Total_compra) As Total_ventas  
FROM Ordenes;  
-- Respuesta: $9123.51
```

	Total_ventas
1	9123.51

2. Promedio de precios de productos por categoría:

```
-- 2. Pregunta: ¿Cuál es el precio promedio de los productos dentro de cada categoría?  
SELECT Id_categoria, AVG (Precio) as Promedio  
FROM Productos  
GROUP BY Id_categoria
```

	Id_categoria	Promedio
1	1	10.000000
2	4	6.500000
3	7	3.000000
4	10	12.500000

3. Orden mínima y máxima por sucursal:

```
-- 3. Pregunta: ¿Cuál es el valor de la orden mínima y máxima por cada sucursal?  
SELECT Id_sucursal, MIN (Total_compra) as Importe_orden_minima, MAX (Total_compra) as Importe_orden_maxima  
FROM Ordenes  
GROUP BY Id_sucursal
```

	Id_sucursal	Importe_orden_minima	Importe_orden_maxima
1	1	1053.51	1053.51
2	2	1075.00	1075.00
3	3	920.00	920.00
4	4	930.00	930.00
5	5	955.00	955.00
6	6	945.00	945.00
7	7	1065.00	1065.00
8	8	1085.00	1085.00
9	9	1095.00	1095.00

4. Mayor número de kilómetros recorridos para entrega:

```
-- 4. Pregunta: ¿Cuál es el mayor número de kilómetros recorridos para una entrega?  
SELECT Max (Kms_recorrer) as Maximo_Kms  
FROM Ordenes
```

	Maximo_Kms
1	12.50

Módulo 2

5. Promedio de cantidad de productos por orden: ¿Cuál es la cantidad promedio de productos por orden?

```
-- 5. Pregunta: ¿Cuál es la cantidad promedio de productos por orden?
SELECT Id_orden, AVG (Cantidad) as Promedio_productos
FROM Detalle Ordenes
GROUP BY Id_orden
```

	Id_orden	Promedio_productos
1	1	3

6. Total de ventas por tipo de pago:

```
-- 6. ¿Cómo se distribuye la Facturación Total del Negocio de acuerdo a los métodos de pago?
SELECT Id_pago, SUM (Total_compra) as Total_Compra
FROM Ordenes
GROUP BY Id_pago
```

	Id_pago	Total_Compra
1	1	1053.51
2	2	1075.00
3	3	920.00
4	4	930.00
5	5	955.00
6	6	945.00
7	7	1065.00
8	8	1085.00
9	9	1095.00

7. Sucursal con la venta promedio más alta:

```
-- 7. Pregunta: ¿Cuál Sucursal tiene el ingreso promedio más alto?
SELECT TOP (1) Id_sucursal, AVG (Total_compra) as Ingreso_promedio
FROM Ordenes
GROUP BY Id_sucursal
```

	Id_sucursal	Ingreso_promedio
1	1	1053.510000

8. Sucursal con la mayor cantidad de ventas por encima de un umbral:

```
-- 8. ¿Cuáles son las sucursales que han generado ventas totales por encima de $ 1000?
SELECT Id_sucursal, Total_compra
FROM Ordenes
WHERE Total_compra>1000.00
ORDER BY Total_compra DESC
```

	Id_sucursal	Total_compra
1	9	1095.00
2	8	1085.00
3	2	1075.00
4	7	1065.00
5	1	1053.51

Módulo 2

9. Comparación de ventas promedio antes y después de una fecha específica:

```
-- 9. ¿Cómo se comparan las ventas promedio antes y después del 1 de julio de 2023?  
  
SELECT 'Antes de Julio' as Periodo,  
AVG (Total_compra) as Promedio_ventas  
FROM Ordenes  
WHERE Fecha_entrega < '2023-07-01'  
UNION  
SELECT 'Despues de Julio' as Periodo,  
AVG (Total_compra) as Promedio_ventas  
FROM Ordenes  
WHERE Fecha_entrega >= '2023-07-01'
```

	Periodo	Promedio_ventas
1	Antes de Julio	979.751666
2	Despues de Julio	1081.666666

10. Análisis de actividad de ventas por horario:

```
-- 10 ¿Durante qué horario del día (mañana, tarde, noche) se registra la mayor cantidad de ventas,  
-- cuál es el ingreso promedio de estas ventas, y cuál ha sido el importe máximo alcanzado por una orden en dicha jornada?  
  
WITH VentasPorHorario AS(  
    SELECT Horario_venta,  
           COUNT(*) AS CantidadVentas,  
           AVG(Total_compra) AS IngresoPromedio,  
           MAX(Total_compra) AS ImporteMaximo  
    FROM Ordenes  
    GROUP BY Horario_venta)  
  
SELECT Horario_venta,  
       CantidadVentas,  
       IngresoPromedio,  
       ImporteMaximo  
FROM VentasPorHorario  
WHERE CantidadVentas = (  
    SELECT MAX(CantidadVentas)  
    FROM VentasPorHorario);  
  
-- Debio ser necesario utilizar una consulta temporal (WITH) que permitió definir variables como las cantidades vendidas, el ingreso promedio y el importe máximo  
-- y una subconsulta (WHERE) para seleccionar el maximo de ventas por horario, lo que permitió a la tabla brindar todas las especificaciones en conjunto
```

	Horario_venta	CantidadVentas	IngresoPromedio	ImporteMaximo
1	Mañana	3	1016.170000	1065.00
2	Noche	3	986.666666	1095.00
3	Tarde	3	1038.333333	1085.00

Avance 4: Consultas con Múltiples Tablas y JOINS

Preguntas para consultas:

1. Listar todos los productos y sus categorías:

```
--1 Listar todos los productos y sus categorías  
--Pregunta: ¿Cómo puedo obtener una lista de todos los productos junto con sus categorías?
```

```
SELECT v.Producto, a.Categoria as Categoria  
FROM Productos v  
INNER JOIN Categorías a ON v.Id_categoria = a.Id_categoria
```

	Producto	Categoria
1	Hamburguesa Deluxe	Comida Rápida
2	Cheeseburger	Comida Rápida
3	Pizza Margarita	Pizzas
4	Pizza Pepperoni	Pizzas
5	Helado de Chocolate	Helados
6	Helado de Vainilla	Helados
7	Ensalada César	Ensaladas
8	Ensalada Griega	Ensaladas

2. Obtener empleados y su sucursal asignada:

```
--2 Obtener empleados y su sucursal asignada  
--Pregunta: ¿Cómo puedo saber a qué sucursal está asignado cada empleado?
```

```
SELECT v.Nombre_empleado, a.Sucursal as Sucursal  
FROM Empleados v  
INNER JOIN Sucursales a ON v.Id_sucursal = a.Id_sucursal
```

	Nombre_empleado	Sucursal
1	John Doe	Sucursal Central
2	Jane Smith	Sucursal Central
3	Bill Jones	Sucursal Central
4	Alice Johnson	Sucursal Central
5	Tom Brown	Sucursal Central
6	Emma Davis	Sucursal Central
7	Lucas Miller	Sucursal Central
8	Olivia García	Sucursal Central
9	Ethan Martinez	Sucursal Central
10	Sophia Rodriguez	Sucursal Central

3. Identificar productos sin categoría asignada:

```
-- 3 Identificar productos sin categoría asignada  
-- Pregunta: ¿Existen productos que no tienen una categoría asignada?  
--  
SELECT v.Producto, a.Id_categoria AS Categoria  
FROM Productos v  
FULL JOIN Categorías a ON v.Id_categoria = a.Id_categoria  
-- Respuesta: No existen productos sin categoría, solo existen categorías sin productos asignados
```

Módulo 2

	Producto	Categoria
1	Hamburguesa Deluxe	1
2	Cheeseburger	1
3	Pizza Margarita	10
4	Pizza Pepperoni	10
5	Helado de Chocolate	7
6	Helado de Vainilla	7
7	Ensalada César	4
8	Ensalada Griega	4
9	NULL	2
10	NULL	3
11	NULL	5
12	NULL	6
13	NULL	8
14	NULL	9

4. Detalle completo de órdenes:

```
--4 Detalle completo de órdenes
-- Pregunta: ¿Cómo puedo obtener un detalle completo de las órdenes, incluyendo el Nombre del cliente,
-- Nombre del empleado que tomó la orden, y Nombre del mensajero que la entregó?
```

```
SELECT t1.Id_orden,
t2.Nombre_cliente, t3.Nombre_empleado, t4.Nombre_mensajero, t1.Horario_venta, t1.Total_compra,
t1.Kms_recorrer, t1.Fecha_despacho, t1.Fecha_entrega, t1.Fecha_orden_lista, t1.Fecha_orden_tomada
FROM Ordenes as t1
JOIN Clientes as t2
    ON t1.Id_cliente = t2.Id_cliente
JOIN Empleados as t3
    ON t1.Id_empleado = t3.Id_empleado
JOIN Mensajeros as t4
    ON t1.Id_mensajero = t4.Id_mensajero
```

Id_orden	Nombre_cliente	Nombre_empleado	Nombre_mensajero	Horario_venta	Total_compra	Kms_recorrer	Fecha_despacho	Fecha_entrega	Fecha_orden_lista	Fecha_orden_tomada
1	Cliente Uno	John Doe	Mensajero Uno	Mañana	1053.51	5.50	2023-01-10 08:30:00.000	2023-01-10 09:00:00.000	2023-01-10 08:15:00.000	2023-01-10 08:00:00.000
2	Cliente Dos	Jana Smith	Mensajero Dos	Tarde	1075.00	10.00	2023-02-15 14:30:00.000	2023-02-15 15:00:00.000	2023-02-15 14:00:00.000	2023-02-15 13:30:00.000
3	Cliente Tres	Bill Jones	Mensajero Tres	Noche	920.00	2.00	2023-03-20 19:30:00.000	2023-03-20 20:00:00.000	2023-03-20 19:15:00.000	2023-03-20 19:00:00.000
4	Cliente Cuatro	Alice Johnson	Mensajero Cuatro	Mañana	930.00	0.50	2023-04-25 09:30:00.000	2023-04-25 10:00:00.000	2023-04-25 09:15:00.000	2023-04-25 09:00:00.000
5	Cliente Cinco	Tom Brown	Mensajero Cinco	Tarde	955.00	8.00	2023-05-30 15:30:00.000	2023-05-30 16:00:00.000	2023-05-30 15:15:00.000	2023-05-30 15:00:00.000
6	Cliente Seis	Emma Davis	Mensajero Seis	Noche	945.00	12.50	2023-06-05 20:30:00.000	2023-06-05 21:00:00.000	2023-06-05 20:15:00.000	2023-06-05 20:00:00.000
7	Cliente Siete	Lucas Miller	Mensajero Siete	Mañana	1065.00	7.50	2023-07-10 08:30:00.000	2023-07-10 09:00:00.000	2023-07-10 08:15:00.000	2023-07-10 08:00:00.000
8	Cliente Ocho	Olivia Garcia	Mensajero Ocho	Tarde	1085.00	9.50	2023-08-15 14:30:00.000	2023-08-15 15:00:00.000	2023-08-15 14:15:00.000	2023-08-15 14:00:00.000
9	Cliente Nueve	Ethan Martinez	Mensajero Nueve	Noche	1095.00	3.00	2023-09-20 19:30:00.000	2023-09-20 20:00:00.000	2023-09-20 19:15:00.000	2023-09-20 19:00:00.000

5. Productos vendidos por sucursal:

```
-- 5 Productos vendidos por sucursal
-- Pregunta: ¿Cuántos artículos correspondientes a cada Categoría de Productos se han vendido en cada sucursal?
```

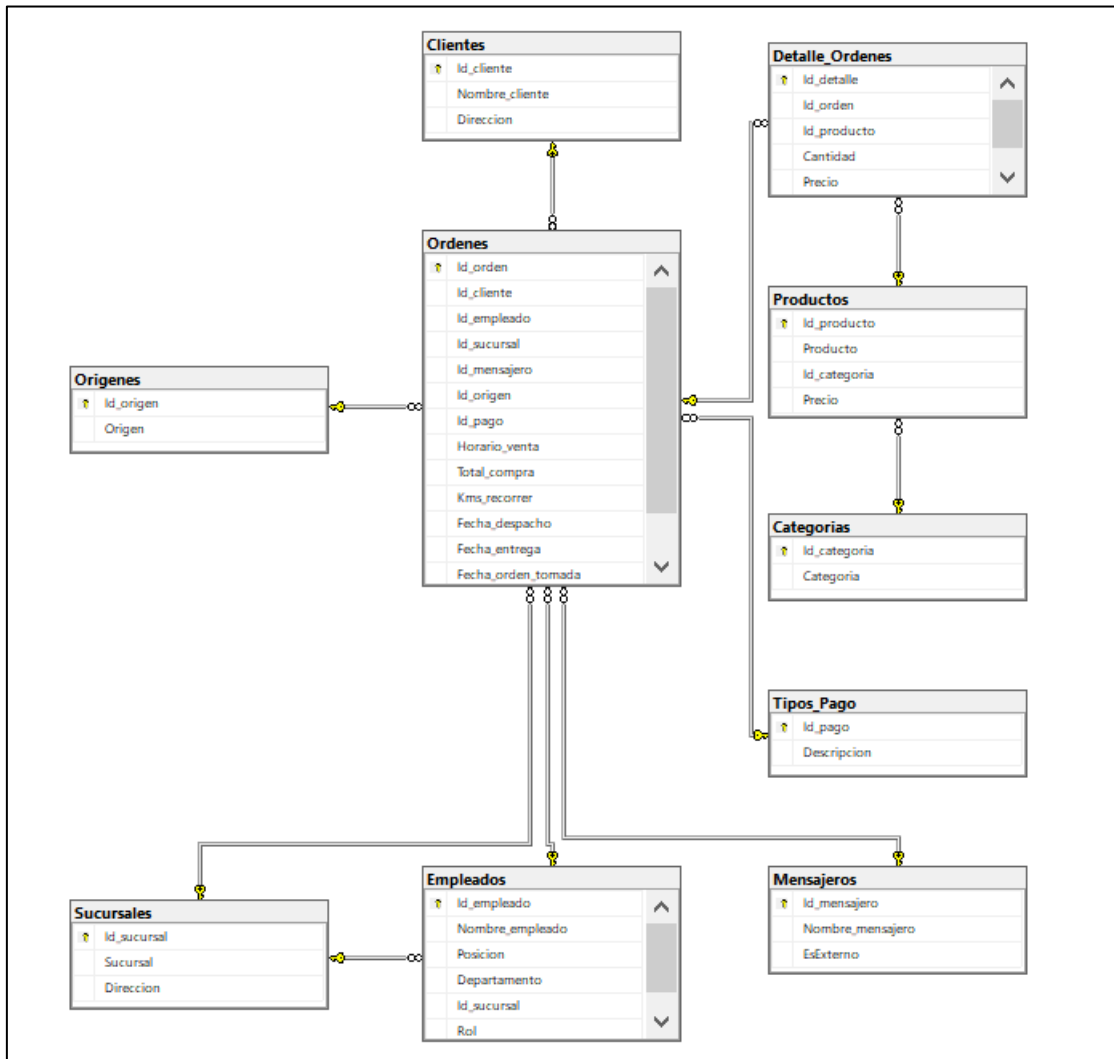
```
SELECT Sucursales.Sucursal,
Categorias.Categoria,
SUM (Detalle_Ordenes.Cantidad) AS TotalArticulosVendidos
FROM Detalle_Ordenes
JOIN Productos ON Detalle_Ordenes.Id_producto = Productos.Id_producto
JOIN Categorias ON Productos.Id_categoria = Categorias.Id_categoria
JOIN Ordenes ON Detalle_Ordenes.Id_orden = Ordenes.Id_orden
JOIN Sucursales ON Ordenes.Id_sucursal = Sucursales.Id_sucursal
GROUP BY Sucursales.Sucursal, Categorias.Categoria
```

	Sucursal	Categoria	TotalArticulosVendidos
1	Sucursal Central	Comida Rápida	8
2	Sucursal Central	Ensaladas	4
3	Sucursal Central	Helados	5
4	Sucursal Central	Pizzas	8

Resultados y consultas

El Esquema Relacional arrojado por SQL Server se asemeja al Esquema Imaginado:

Ilustración 2: Modelo Relacional (SQL Server)



Análisis general de la empresa:

El negocio parece estar optimizando su oferta de productos y ajustando su estructura interna y geográfica, mientras mantiene una amplia accesibilidad a través de múltiples canales de venta y métodos de pago, lo que sugiere un enfoque en mejorar la rentabilidad y la eficiencia operativa. A continuación, se detallan las principales observaciones clave:

- a. **Diversificación de productos y categorías:** El negocio tiene una oferta variada que incluye comida rápida, postres, bebidas, ensaladas, pizzas y más, lo que sugiere una amplia gama de opciones para atraer diferentes tipos de clientes.
- b. **Distribución geográfica y logística:** Con sucursales distribuidas en varias ubicaciones (ciudad, playa, montaña, lago, bosque, etc.), el negocio parece estar bien establecido en distintas zonas geográficas, lo que le permite atender tanto a áreas urbanas como a zonas más alejadas.
- c. **Optimización interna:** Se observan ajustes internos como el aumento de precios en productos específicos y la reasignación de roles (por ejemplo, todos los cocineros ahora son "Chefs"). Esto indica un esfuerzo por mejorar la eficiencia operativa y la estructura del negocio.
- d. **Adaptabilidad y decisiones estratégicas:** La eliminación de categorías de productos (postres) y la clausura de sucursales (Sucursal Bosque) muestra una gestión proactiva, eliminando elementos que podrían no ser rentables o relevantes para la estrategia actual del negocio.
- e. **Integridad de la base de datos:** Se asegura que, al realizar eliminaciones (órdenes, productos, sucursales), las relaciones referenciales se mantengan correctas, eliminando también registros relacionados, lo cual es clave para la coherencia de la base de datos.
- f. **Diversificación de canales de venta y métodos de pago:** El negocio permite compras tanto de forma presencial como en línea, vía móvil, redes sociales, etc., y ofrece varios métodos de pago, lo que demuestra una estrategia de accesibilidad y conveniencia para el cliente.

Análisis de los resultados hallados por medio de consultas:

- a. En primer lugar, me parece relevante destacar que parece que los datos más completos son de la Sucursal 1, mientras que la información de las demás sucursales es limitada y solo incluye aspectos básicos como su ubicación y algunas órdenes asociadas. Esto se puede observar en las tablas de Empleados (donde todos están asignados a la sucursal 1) y en la tabla ordenes que, aunque se cargaron órdenes para varias sucursales (por ejemplo, Sucursal 2, Sucursal 3, etc.), los empleados involucrados en las órdenes son de la sucursal 1, lo que parece sugerir que la Sucursal 1 es la principal o central, o simplemente que el sistema aún no ha capturado información detallada de las otras sucursales.
- b. Existe una baja diversificación de productos por categoría (solo hay 2).
- c. La distribución del personal en las áreas parece bastante eficiente a simple vista.
- d. Las sucursales más eficientes son la 9 (Sucursal Lago) y la 1 (Sucursal Central), ya que cuentan con un promedio de ventas mayor a \$1000.00 y se encuentran por debajo del promedio de kilómetros totales de todas las sucursales (6.5 km).
- e. El total de ventas fue de \$ 9123.51.
- f. La categoría "Pizzas" tiene el precio promedio más alto (\$12.5) y la categoría "Helados" el más bajo (\$3).
- g. La orden mínima tuvo un importe de \$920.00 y fue tomada por la sucursal 3, mientras que la orden máxima fue de \$1075.00, tomada por la sucursal 2.
- h. La mayor distancia recorrida de una entrega fue de 12.50 kms.
- i. El promedio de productos por orden es de 3.
- j. Los tres primeros medios de pago mayormente utilizados por los clientes son Cupón de Descuento, Vale de Comida y Tarjeta de Crédito.
- k. Después del mes de julio las ventas subieron, en promedio, \$101.91
- l. El mayor volumen de ingresos se registra por la tarde, luego en la mañana y finalmente en la noche.
- m. En la Sucursal Central, lo que más se vendió fueron pizzas y comida rápida, luego helados y, por último, ensaladas.

Consultas finales:

1. Eficiencia de los mensajeros: ¿Cuál es el tiempo promedio desde el despacho hasta la entrega de los pedidos gestionados por todo el equipo de mensajería?

```
SELECT AVG (DATEDIFF (MINUTE, Fecha_despacho, Fecha_entrega)) as Minutos_promedio
FROM Ordenes
```

Minutos_promedio
30

2. Análisis de Ventas por Origen de Orden: ¿Qué canal de ventas genera más ingresos?

```
SELECT a.Origen AS Canal_ventas, SUM(o.Total_compra) AS Ingresos_totales
FROM Ordenes o
JOIN Origenes a ON o.Id_origen = a.Id_origen
GROUP BY a.Origen
ORDER BY Ingresos_totales DESC;
```

	Canal_ventas	Ingresos_totales
1	Presencial	2140.00
2	Drive Thru	2025.00
3	Teléfono	2005.00
4	En línea	1998.51
5	App Móvil	955.00

3. Productividad de los Empleados: ¿Cuál es el nivel de ingreso generado por Empleado?

```
SELECT a.nombre_empleado AS Empleado, SUM(o.Total_compra) AS Ingresos_totales
FROM Ordenes o
JOIN Empleados a ON o.Id_empleado = a.Id_empleado
GROUP BY a.nombre_empleado
ORDER BY Ingresos_totales DESC;
```

	Empleado	Ingresos_totales
1	Ethan Martinez	1095.00
2	Olivia Garcia	1085.00
3	Jane Smith	1075.00
4	Lucas Miller	1065.00
5	John Doe	1053.51
6	Tom Brown	955.00
7	Emma Davis	945.00
8	Alice Johnson	930.00
9	Bill Jones	920.00

4. Análisis de Demanda por Horario y Día: ¿Cómo varía la demanda de productos a lo largo del día? NOTA: Esta consulta no puede ser implementada sin una definición clara del horario (mañana, tarde, noche) en la base de datos existente. Asumiremos que HorarioVenta refleja esta información correctamente.

Módulo 2

```

SELECT
    t1.Horario_venta, SUM(t2.Cantidad) as Demanda,
    COUNT(t1.Id_orden) as Cantidad_ordenes
FROM
    Ordenes as t1
    JOIN Detalle_ordenes as t2
    ON t1.Id_orden = t2.Id_orden
GROUP BY
    t1.Horario_venta;

```

	Horario_venta	Demanda	Cantidad_ordenes
1	Mañana	25	8

5. Comparación de Ventas Mensuales: ¿Cuál es la tendencia de los ingresos generados en cada periodo mensual?

```

SELECT
    YEAR(Fecha_entrega) AS Año,
    MONTH(Fecha_entrega) AS Mes,
    SUM(Total_compra) AS Ingresos_mensuales
FROM
    Ordenes
GROUP BY
    YEAR(Fecha_entrega),
    MONTH(Fecha_entrega)
ORDER BY
    Mes ASC;

```

	Año	Mes	Ingresos_mensuales
1	2023	1	1053.51
2	2023	2	1075.00
3	2023	3	920.00
4	2023	4	930.00
5	2023	5	955.00
6	2023	6	945.00
7	2023	7	1065.00
8	2023	8	1085.00
9	2023	9	1095.00

6. Análisis de Fidelidad del Cliente: ¿Qué porcentaje de clientes son recurrentes versus nuevos clientes cada mes? NOTA: La consulta se enfocaría en la frecuencia de órdenes por cliente para inferir la fidelidad.

```

SELECT Id_cliente, COUNT(Id_orden) as Cantidad
FROM Ordenes
GROUP BY Id_cliente;

```

	Id_cliente	Cantidad
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1
6	6	1
7	7	1
8	8	1
9	9	1

Hallazgos clave

- a. El tiempo promedio de entrega de los mensajeros es de 30 minutos.
- b. La mayor cantidad de ventas se realiza en modo presencial, en segundo lugar, por Drive Thru y luego por Teléfono. La peor performance se la lleva la Aplicación Movil.
- c. Los empleados menos productivos son Tom Brown, Emma Davis, Alice Jhonson y Bill Jones.
- d. No es posible realizar un análisis adecuado de la demanda porque solo tenemos en detalle de ordenes el registro de 1 sola orden, que señala que las compras se realizaron durante la mañana.
- e. Durante los meses de marzo a junio los ingresos se reducen significativamente.
- f. No podemos determinar la frecuencia de los clientes porque todos los clientes registrados hicieron 1 sola compra. Para poder realizar el análisis de frecuencia de clientes necesitaríamos ingresos mas datos.

Recomendaciones estratégicas

- a. Podría optimizarse el recorrido a realizar por parte de los mensajeros para reducir los minutos recorridos. Esto puede hacerse mediante la implementación de sistemas de GPS o aplicaciones que optimicen las rutas de entrega en función del tráfico, distancia o número de órdenes. Otra opción podría ser agrupar varias entregas en un mensajero si es que varios pedidos están cerca entre sí. Finalmente podría evaluarse la opción de incorporar vehículos de reparto más ágiles.
- b. Podrían mejorarse los ingresos obtenidos por la Aplicación Móvil lanzando una campaña de promoción de la misma a través de distintos medios e incorporándole diversas promociones que incentiven a instalarla en el celular. También podría mejorarse la experiencia del usuario en la app.
- c. Podrían realizarse capacitaciones para incrementar la productividad de los empleados. También implementar algún tipo de incentivo (premios, beneficios extra, etc.) para que los empleados quieran superarse día a día.
- d. Se podrían implementar diversas estrategias para atraer mas clientes durante los meses de marzo a junio, por ejemplo, promociones estacionales, o para celebrar eventos especiales. También podrían lanzarse menús o productos temporales. Otras opciones serían bonificaciones para las compras frecuentes u ofertas para los nuevos clientes.

Optimización y sostenibilidad

La base de datos se estructuró utilizando un modelo normalizado, dividiendo los datos en tablas lógicas que representan entidades clave (productos, categorías, clientes, órdenes, etc.), lo que minimiza la redundancia y asegura que los datos sean fácilmente accesibles y mantenibles.

La información de productos, categorías, empleados, clientes, sucursales, métodos de pago y otros elementos están claramente separados en sus respectivas tablas, facilitando las consultas y análisis de datos para informes externos sin tener que lidiar con información redundante.

Se añadieron índices a las columnas más consultadas, como:

- Claves primarias: Todas las tablas cuentan con claves primarias (Id_producto, Id_orden, etc.) para asegurar el acceso rápido a los registros.
- Índices en claves foráneas: Se establecieron relaciones entre las tablas clave mediante el uso de claves foráneas (Id_cliente, Id_producto, Id_empleado, etc.), facilitando la extracción de datos a través de uniones entre tablas, permitiendo informes complejos de forma más eficiente.

También se implementaron consultas agregadas optimizadas para el análisis externo, que agrupan datos por intervalos de tiempo, productos o sucursales, permitiendo generar reportes rápidos sobre ventas, productos populares y rendimiento de sucursales y se minimizaron las operaciones redundantes en consultas de gran volumen, eliminando subconsultas innecesarias y asegurando el uso de operaciones de agregado eficientes como SUM() y COUNT().

Además, las tablas relacionadas con órdenes y ventas fueron particionadas por fecha, de forma que el rendimiento de las consultas que manejan grandes volúmenes de datos históricos sea más eficiente. Esto permite ejecutar consultas basadas en fechas (como entre marzo y junio) sin afectar el rendimiento general de la base de datos.

En cuanto a la sostenibilidad de largo plazo, las claves foráneas garantizarán la integridad de los datos entre las tablas relacionadas. Esto asegura que no existan registros huérfanos o inconsistencias en las referencias de productos, clientes, empleados o sucursales. Cualquier operación de eliminación o actualización debe estar controlada para preservar la consistencia de la base de datos.

Es bueno implementar una política de backups automáticos regulares, tanto completos como incrementales, para garantizar la recuperación de datos en caso de fallos. Esto protege contra la pérdida de información crítica y permite la restauración de versiones anteriores de la base de datos si fuera necesario.

En cuanto a la escalabilidad, la base de datos está diseñada para crecer de manera escalable. Esto quiere decir que, si se añaden más productos,

Módulo 2

sucursales o empleados en el futuro, las tablas y relaciones pueden extenderse sin comprometer el rendimiento.

Finalmente, se destacar que existe cierta optimización en el uso de los tipos de datos, como, por ejemplo, se ha utilizado el formato VARCHAR para las columnas que contienen texto (como Categoría, Dirección, etc.). Esto es así porque, a diferencia del CHAR, que utiliza un tamaño fijo, VARCHAR ajusta el espacio a la longitud real del contenido, ahorrando memoria. Así mismo, el uso de DECIMAL para los precios es adecuado, ya que permite representar números decimales con precisión, lo cual es crucial en aplicaciones financieras.

A pesar de ello, se considera que siempre existe un margen para mejorar y optimizar aún más los tipos de datos utilizados, considerando la naturaleza y las características específicas de cada campo para maximizar la eficiencia en el almacenamiento y el rendimiento en la consulta.

Desafíos y soluciones

El primer desafío fue definir correctamente las tablas y sus relaciones, asegurando la normalización de la misma y evitando redundancias. Para ello, se realizó un diseño lógico que incluyó tablas claramente definidas (categorías, productos, sucursales, empleados, clientes, etc.) con claves primarias y foráneas, ayudando a mantener la integridad de los datos.

Al mismo tiempo, se debió tener en cuenta la integridad de los datos al establecer las relaciones y la forma en la que las actualizaciones o modificaciones de los mismos afectan a las tablas relacionadas.

En la inserción de datos se consideró que los mismos sean consistentes con los tipos de datos de cada tabla y se respeten las relaciones.

Por último, en cuanto a desarrollo de consultas, se tuvo en cuenta que las mismas debían extraer información útil de la base de datos. Para ello, se estructuraron esquemas lógicos, incorporando funciones de agregación, combinación, agrupamiento y orden, fechas y condicionales, que aseguren que las consultas sean informativas y precisas.

Reflexión personal

Esta fue mi primera experiencia trabajando con bases de datos relacionales en SQL. Al comenzar a diseñar la estructura me di cuenta de la importancia que tiene la buena planificación y la esquematización de lo que vamos a desarrollar. Por ello es que me resultó muy útil el diseño previo del modelo relacional, que me permitió tener un esquema visual y mental de la estructura e información de la base.

Al cargar los datos entendí que hay que hacerlo de forma organizada y consistente para conservar la uniformidad de los datos, de tal forma que la información que ésta suministre sea coherente.

En cuanto a la realización de consultas, me sorprendió la variedad de funcionalidades que existen para trabajar con los datos. Algo que me pareció importante es tener en cuenta el impacto de la actualización de los datos al momento de estructurar nuestras queries, de tal forma que se garantice la escalabilidad de los mismos.

Armando queries me di cuenta de la facilidad y el potencial que tiene SQL para trabajar con grandes volúmenes de información. Sin embargo, no solo se trata de aprender a estructurar bases y realizar consultas, sino que también se debe entender como los datos se relacionan con el funcionamiento del negocio. En este sentido, cada consulta y cada análisis se convierten en una herramienta poderosa para impulsar estrategias, optimizar recursos y, en última instancia, contribuir al crecimiento y éxito de la organización.

En definitiva, SQL me pareció un programa muy práctico para la extracción de información relevante, y el armado de queries me pareció entretenido, algo que voy a continuar practicando.

En cuanto al proyecto, quizás lo único que le agregaría sería más información sobre otras sucursales (para que la estructura quede con un mayor grado de detalle) además de un poco mas de datos en general, para complejizar levemente el análisis de la información. Sin embargo, en términos generales, me pareció bastante completo y entretenido.

EXTRA CREDIT

Nuevas consultas estratégicas:

1. Lista de clientes y sus respectivas compras.

```
SELECT
  c.Id_cliente,
  c.nombre_cliente,
  o.Id_orden,
  p.producto,
  d.cantidad,
  o.Fecha_orden_tomada
FROM
  clientes c
INNER JOIN
  ordenes o ON c.Id_cliente = o.Id_cliente
INNER JOIN
  Detalle_Ordenes d ON o.Id_orden = d.Id_orden
INNER JOIN
  productos p ON d.Id_producto = p.Id_producto
ORDER BY
  c.nombre_cliente, o.Fecha_orden_tomada;
```

	Id_cliente	nombre_cliente	Id_orden	producto	cantidad	Fecha_orden_tomada
1	1	Cliente Uno	1	Hamburguesa Deluxe	3	2023-01-10 08:00:00.000
2	1	Cliente Uno	1	Cheeseburger	5	2023-01-10 08:00:00.000
3	1	Cliente Uno	1	Pizza Margarita	4	2023-01-10 08:00:00.000
4	1	Cliente Uno	1	Pizza Pepperoni	4	2023-01-10 08:00:00.000
5	1	Cliente Uno	1	Helado de Chocolate	1	2023-01-10 08:00:00.000
6	1	Cliente Uno	1	Helado de Vainilla	4	2023-01-10 08:00:00.000
7	1	Cliente Uno	1	Ensalada César	2	2023-01-10 08:00:00.000
8	1	Cliente Uno	1	Ensalada Griega	2	2023-01-10 08:00:00.000

2. Productos con bajo stock.
3. Historial de entradas y salidas de productos.
4. Se podría incorporar otras tablas: Proveedores, Calificación de los Clientes, Costos, etc.

Optimización de rendimiento:

Como mencioné anteriormente, la base de datos podría continuar optimizándose. Algunas opciones para esto son las siguientes:

- Si las categorías tienen una longitud predefinida y limitada, como "Comida Rápida", "Postres", "Desayunos", se podría establecer un VARCHAR(30) o un valor adecuado en lugar de dejarlo abierto, lo que reduce el espacio ocupado.
- Columna Id_categoria (INT): Aquí es posible usar un tipo más pequeño como TINYINT o SMALLINT, ya que solo hay 10 categorías en la base de datos, lo que ahorra espacio considerablemente frente a un INT (que usa 4 bytes).