Carrera: Data Analytics

Módulo 2

Fast Food Restaurant Database

Nombre del autor: Gerardo Morsucci

Email: geramorsucci@gmail.com

Cohorte: DAPT-01

Fecha de entrega: 02/05/2024

Institución: Fast Food Restaurant

Introducción

Se me solicita migrar los datos de Fast Food Restaurant hacia una base de datos consolidada, escalable y eficiente, logrando llevarlo a cabo a través de los datos brindados y el lenguaje DDL. También se genera un informe respondiendo preguntas mediante consultas generadas con lenguaje DML,. Una vez respondidas las preguntas se elaboran análisis y recomendaciones.

Desarrollo del proyecto

- ### Avance nº1:

Se creó la base de datos, se definieron sus parámetros de tamaño teniendo en cuenta un crecimiento a futuro de la base de datos misma y su logfile (registra todas las operaciones que modifican los datos o la estructura de la base de datos).

El enfoque principal fue el diseño del esquema de la base de datos, utilizando sentencias del lenguaje de definición de datos (DDL). Este proceso incluirá la creación de tablas, definición de campos y establecimiento de relaciones. Para darle uniformidad a las sentencias se decidió usar PascalCase.

A continuación se mostrará la sentencia con la que se creó la tabla DetallesOrden:

```
-- Tabla Detalles Orden
CREATE TABLE DetallesOrden (
    OrdenID INT,
    ProductoID INT,
    Cantidad INT,
    Precio DECIMAL(10,2),
    PRIMARY KEY (OrdenID,ProductoID),
    FOREIGN KEY (OrdenID) REFERENCES Ordenes(OrdenID),
    FOREIGN KEY (ProductoID) REFERENCES Productos
);
```

La tabla DetallesOrden contiene una optimización mediante una llave primaria compuesta, la cuál se crea con los campos OrdenID y ProductoID.

En toda la base de datos se utilizó un criterio en los campos formateados como DECIMAL de hasta 2 decimales.

A continuación la sentencia con la que se creó la tabla Mensajeros:

```
-- Tabla Mensajeros
CREATE TABLE Mensajeros (
    MensajeroID INT PRIMARY KEY IDENTITY,
    Nombre VARCHAR(100) NOT NULL,
    EsExterno BIT NOT NULL
);
```

Las llaves primarias en todas las tablas fueron creadas con la propiedad IDENTITY. Esta sentencia se usa para generar valores de clave.

La propiedad de identidad de una columna garantiza las siguientes condiciones:

- Cada nuevo valor se genera basándose en el valor actual de inicialización e incremento.
- Cada nuevo valor de una transacción determinada es diferente de otras transacciones simultáneas de la tabla.

Se estableció el criterio de limitar hasta 100 caracteres los campos formateados con VARCHAR. Se optó por VARCHAR debido a que NVARCHAR puede utilizar caracteres especiales unicode lo cual no es necesario debido a nuestros datos. Estas dos decisiones optimizan nuestra base de datos.

A lo largo de la base de datos se realizó un manejo de prevención de datos nulos con la sentencia NOT NULL. La misma impide que se ingresen valores nulos en una columna, garantizando la integridad de los datos (calidad y confiabilidad de los datos).

Sentencia con la que se creó la tabla Categorías y Productos:

```
-- Tabla Categorías
CREATE TABLE Categorias (
    CategoriaID INT PRIMARY KEY IDENTITY,
    Nombre VARCHAR(100) NOT NULL
);


-- Tabla productos
CREATE TABLE Productos (
    ProductoID INT PRIMARY KEY IDENTITY,
    Nombre VARCHAR(100),
    CategoriaID INT,
    Precio DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (CategoriaID) REFERENCES Categorias(CategoriaID)
);
```

Ejemplo de cómo se fueron creando las llaves foráneas. Se crean para mantener enlaces entre tablas y así evitar que accidentalmente se destruyan los datos en ocasiones de mala manipulación de los mismos. Mediante REFERENCES se establecieron las relaciones entre las tablas.

Luego de crear todas las tablas y formatearlas correctamente, se creó el esquema relacional. Son herramientas extremadamente útiles durante la fase de diseño de una base de datos, ya que permiten a los desarrolladores y otros interesados entender la organización y las relaciones entre diferentes conjuntos de datos.

- ## Avance n°2:

Se poblaron las tablas con los datos pertinentes mediante sentencias INSERT.

. Se realizaron actualizaciones de datos mediante sentencias UPDATE, en primera instancia se aumentó en 1 los precios de la categoría 'Comida rápida' (id = 1). Para lograr afectar solo los registros deseados se utilizó WHERE (Sentencia utilizada para realizar filtrados).

Se actualizó la posición de todos los empleados del departamento 'Cocina' a 'Chef' y también se actualizó la dirección de una sucursal específica (SucursalID = 1).

Se hizo un DELETE de la OrdenID = 10. Esta operación eliminará la orden junto con sus detalles asociados, también se verificó que esta eliminación no afecte la integridad de los datos.

También se eliminaron todos los productos de la categoría 'Postres' (CategoriaID = 2) para realizarlo sin problema se deshabilita temporalmente la restricción de la foreign key NOCHECK CONSTRAINT sobre los datos afectados. Luego se volvieron a activar las restricciones sobre estos datos con CHECK CONSTRAINT.

Debido a que la Sucursal Bosque (SucursalID = 10) cerró, se eliminaron los empleados de la misma.

A través de funciones de agregación se contestaron preguntas básicas:

¿Cuántos productos diferentes hay en la orden 1?

Verificamos que la OrdenID = 1 **tiene 10 productos**.

Usando COUNT (función de conteo) acompañada de DISTINCT (para evitar datos duplicados)

```
SELECT OrdenID, COUNT(DISTINCT ProductoID) AS Conteo
from DetallesOrden
where OrdenID = 1
group by OrdenID;
```

¿Cuál es la cantidad total de registros únicos en la tabla de órdenes?

También con un simple COUNT verificamos que **tiene 9 registros únicos.**

```
SELECT COUNT(OrdenID) AS 'Regristos unicos'
FROM Ordenes;
```

Regristos unicos
9

A pesar de sus pocos registros, la base de datos actual está preparada para soportar gran cantidad de datos.

¿Cuántos empleados existen en cada departamento?

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

La cantidad de empleados en cada departamento está distribuida de forma correcta, en un futuro para tener más sucursales abiertas se deberá incorporar nuevos empleados.

```
select Departamento,
    COUNT(EmpleadoID) AS Cantidad
from Empleados
group by Departamento
order by Cantidad DESC;
```

Se ordenó de forma descendente con ORDER BY

La cláusula GROUP BY combina registros con valores idénticos en la lista de campos especificados en un único registro. Si en la instrucción SELECT se incluye una función de agregado de SQL, como SUM o COUNT, se creará un valor de resumen para cada registro. Si no usaramos GROUP BY en esta consulta, nos arrojaría un error.

¿Cuántos productos hay por código de categoría?

	CategoriaID	Cantidad
1	1	2
2	4	2
3	7	2
4	10	2

No hay datos de productos para todas las categorías existentes, se debe considerar incorporar más productos para abarcar todas las categorías.

```
SELECT CategoriaID, COUNT(ProductoID) AS Cantidad
FROM Productos
GROUP BY CategoriaID
HAVING count(ProductoID) >= 2;
```

Para responder se usó un filtrado con HAVING

Una cláusula HAVING es como una cláusula WHERE , pero que solo se aplica a los grupos en su totalidad (es decir, a las filas del conjunto de resultados que representa los grupos), a diferencia de la cláusula WHERE , que se aplica a filas individuales.

¿Cuántos clientes se han importado a la tabla de clientes?

Se han importado 10 clientes

```
SELECT COUNT(*) Conteo
FROM Clientes;
```

¿Cuáles son las sucursales con un promedio de ventas por orden superior a un valor específico, ordenadas por el promedio de kilómetros recorridos para las entregas de mayor a menor?

La sucursal (id = 9) con menos cantidad de kilómetros recorridos fue la que **más promedio de venta total consiguió (95)**, y **la sucursal (id = 6)** con la mayor cantidad de kilómetros recorridos fue la que **menos promedio de venta total consiguió (45)**

Se recomienda reconsiderar costos de traslado actuales para mejorar la ganancia en mayores distancias, es decir mejorar la rentabilidad.

SucursalID	PromedioVenta	PromedioKilometros
6	45.00	12.50
2	75.00	10.00
8	85.00	9.50
5	55.00	8.00
7	65.00	7.50
1	50.00	5.50
9	95.00	3.00

```
select SucursalID,
    CAST(AVG(TotalCompra) AS DECIMAL(10,2)) PromedioVenta,
    FORMAT(AVG(KilometrosRecorrer), '0.00') PromedioKilometros
from Ordenes
group by SucursalID
having AVG(TotalCompra) > 40
order by AVG(KilometrosRecorrer) DESC;
```

- **Avance nº3:**

Nos enfocaremos en consultas más avanzadas utilizando funciones de agregación.

¿Cuál es el total de ventas (TotalCompra) a nivel global?

VentasNivelGlobal
520.00

```
SELECT SUM(TotalCompra) AS VentasNivelGlobal
FROM Ordenes;
```

Se recomienda tener en cuenta este dato a futuro, para que cuando transcurra la misma cantidad de tiempo que tiene el dato más viejo de la tabla, se realicen cálculos de crecimiento de ventas.

¿Cuál es el precio promedio de los productos dentro de cada categoría?

CategoriaID	Precio Promedio
10	12.49
1	9.99
4	6.49
7	2.99

La categoría 'Pizzas' se queda con el promedio más elevado, y la categoría 'Helados' con el promedio menos elevado. Se recomienda comparar precios actuales con los de otras empresas referentes.

```
SELECT CategoriaID, CAST(AVG(Precio) AS DECIMAL(10,2)) 'Precio Promedio'
 FROM Productos
 GROUP BY CategoriaID
 ORDER BY 'Precio promedio' DESC;
```

¿Cuál es el valor de la orden mínima y máxima por cada sucursal?

SucursalID	Minimo	Maximo
1	50.00	50.00
2	75.00	75.00
3	20.00	20.00
4	30.00	30.00
5	55.00	55.00
6	45.00	45.00
7	65.00	65.00
8	85.00	85.00
9	95.00	95.00

La tabla está mostrando datos poco confiables debido a la poca cantidad de datos (un solo registro de venta por sucursal)(Underfitting). Esto se puede ver debido a que nuestros datos comparten el mismo minimo y maximo.

```
SELECT SucursalID, MIN(TotalCompra) AS Minimo, MAX(TotalCompra) AS Maximo
FROM Ordenes
GROUP BY SucursalID;
```

¿Cuál es el mayor número de kilómetros recorridos para una entrega?

Máximo
12.50

```
SELECT MAX(KilometrosRecorrer) AS 'Máximo'
FROM Ordenes;
```

Se debe evaluar si es rentable realizar tanto recorrido por una sola entrega (en el caso de un pedido grande puede llegar a ser rentable)

¿Cuál es la cantidad promedio de productos por orden?

OrdenID	PromedioCantidadProducto
1	3

```
SELECT OrdenID, AVG(Cantidad) AS PromedioCantidadProducto
FROM DetallesOrden
GROUP BY OrdenID;
```

Pueden realizarse promociones para incentivar compras grandes

¿Cuál es el total de ventas por cada tipo de pago?

TipoPagoID	TotalVenta
9	95.00
8	85.00
2	75.00
7	65.00
5	55.00
1	50.00
6	45.00
4	30.00
3	20.00

El tipo de pago con ID = 9 es 'Cupones de descuento', podemos concluir que a la gente le atraen los actuales descuentos de la tienda, se pueden crear nuevos cupones en base a los anteriores.

```
SELECT TipoPagoID, SUM(TotalCompra) AS TotalVenta
FROM Ordenes
GROUP BY TipoPagoID
ORDER BY TotalVenta DESC;
```

¿Cuál sucursal tiene la venta promedio más alta?

SucursalID	PromedioVenta
9	95.00

```
SELECT TOP 1 SucursalID, CAST(AVG(TotalCompra) AS DECIMAL(10,2)) PromedioVenta
FROM Ordenes
GROUP BY SucursalID
ORDER BY PromedioVenta DESC;
```

La sucursal con ID = 9 es la 'Sucursal Lago', tiene las ventas más altas, se debe analizar su ubicación para encontrar algún patrón que se pueda replicar en otras sucursales.

¿Cuáles son las sucursales que han generado ventas por orden por encima de $50?

	SucursalID	TotalVentas	NumeroOrdenes	PromedioVentaXOrden
1	9	95.00	1	95.00
2	8	85.00	1	85.00
3	2	75.00	1	75.00
4	7	65.00	1	65.00
5	5	55.00	1	55.00

Cuando la base de datos crezca se podrán hacer análisis más complejos, por lo pronto el promedio es igual al MÁXIMO de ventas, esto nos dice que contamos con pocos datos.

```
SELECT SucursalID,
    SUM(TotalCompra) AS TotalVentas,
    COUNT(OrdenID) AS NumeroOrdenes,
    CAST(SUM(TotalCompra)/COUNT(OrdenID) AS DECIMAL(10,2)) AS PromedioVentaXOrden
FROM Ordenes
GROUP BY SucursalID
HAVING SUM(TotalCompra) > 50
ORDER BY TotalVentas DESC;
```

¿Cómo se comparan las ventas promedio antes y después del 1 de julio de 2023?

	Período	TotalVenta
1	Antes del 1 de julio de 2023	45.83
2	Después del 1 de julio de 2023	81.67

En la segunda mitad del año, las ventas se vieron incrementadas en un 56.11%

Se deben analizar los productos vendidos antes y después del 1 de julio para hallar una posible razón del incremento de las ventas.

```
SELECT 'Antes del 1 de julio de 2023' AS 'Período',
    CAST(AVG(TotalCompra) AS DECIMAL(10,2)) TotalVenta
FROM Ordenes
WHERE FechaOrdenTomada < '2023-07-01'
UNION  --Para usar el union las dos querys deben tener la misma estructura y el ALL incluye duplicados
SELECT 'Después del 1 de julio de 2023' AS 'Período',
    CAST(AVG(TotalCompra) AS DECIMAL(10,2)) TotalVenta
FROM Ordenes
WHERE FechaOrdenTomada > '2023-07-01';
```

¿Durante qué horario del día (mañana, tarde, noche) se registra la mayor cantidad de ventas, cuál es el valor promedio de estas ventas, y cuál ha sido la venta máxima alcanzada?

	HorarioVenta	CantidadVentas	PromedioVentas	VentaMaxima
1	Mañana	3	48.33	65.00
2	Noche	3	53.33	95.00
3	Tarde	3	71.67	85.00

Se encontró que la cantidad de ventas fue pareja para los 3 horarios (Mañana, Noche, Tarde) La venta máxima la realizó el turno noche, pero la diferencia no varía mucho entre noche y tarde. Se puede realizar un trabajo de marketing por la mañana para incrementar las ventas en este horario..

```
SELECT HorarioVenta,
    COUNT(*) as CantidadVentas,
    CAST(AVG(TotalCompra) AS DECIMAL(10,2)) PromedioVentas,
    MAX(TotalCompra) VentaMaxima
FROM Ordenes
GROUP BY HorarioVenta
ORDER BY CantidadVentas DESC;
```

- ## Avance n°4

Se combina información de diferentes tablas para obtener resultados más completos en las consultas.

¿Cómo puedo obtener una lista de todos los productos junto con sus categorías?

Producto	Categoria
Hamburguesa Deluxe	Comida Rápida
Cheeseburger	Comida Rápida
Pizza Margarita	Pizzas
Pizza Pepperoni	Pizzas
Helado de Chocolate	Helados
Helado de Vainilla	Helados
Ensalada César	Ensaladas
Ensalada Griega	Ensaladas

```
SELECT P.Nombre Producto, C.Nombre Categoria
FROM Productos P
INNER JOIN Categorias C ON (P.CategoriaID = C.CategoriaID);
```

Logramos obtener la lista utilizando un INNER JOIN que combina los registros de dos tablas si hay valores coincidentes en un campo común.

Se observa que hay dos productos por categoría por ende en un futuro habría que agregar productos para llenar las demás categorías existentes.

¿Cómo puedo saber a qué sucursal está asignado cada empleado?

	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

Se observa que todos los empleados actuales pertenecen a 'Sucursal Central'.

```
SELECT E.Nombre Empleado, S.Nombre Sucursal
FROM Empleados E
LEFT JOIN Sucursales S ON (E.SucursalID = S.SucursalID);
```

Use una operación LEFT JOIN para crear un combinación externa izquierda. En las combinaciones externas izquierdas se incluyen todos los registros de la primera tabla (izquierda), incluso si no existen valores coincidentes para registros en la segunda tabla (derecha).

¿Existen productos que no tienen una categoría asignada?

```
SELECT P.Nombre, C.CategoriaID
FROM Productos P
LEFT JOIN Categorias C ON (P.CategoriaID = C.CategoriaID)
WHERE C.CategoriaID IS NULL;
```

A través de la siguiente consulta se verificó que no existen productos que no tengan categoría asignada.

¿Cómo puedo obtener un detalle completo de las órdenes, incluyendo cliente, empleado que tomó la orden, y el mensajero que la entregó?

OrdenID	Cliente	Empleado	Mensajero
1	Cliente Uno	John Doe	Mensajero Uno
2	Cliente Dos	Jane Smith	Mensajero Dos
3	Cliente Tres	Bill Jones	Mensajero Tres
4	Cliente Cuatro	Alice Johnson	Mensajero Cuatro
5	Cliente Cinco	Tom Brown	Mensajero Cinco
6	Cliente Seis	Emma Davis	Mensajero Seis
7	Cliente Siete	Lucas Miller	Mensajero Siete
8	Cliente Ocho	Olivia García	Mensajero Ocho
9	Cliente Nueve	Ethan Martinez	Mensajero Nueve

```
SELECT O.OrdenID, C.Nombre Cliente, E.Nombre Empleado, M.Nombre Mensajero
FROM Ordenes O
LEFT JOIN Clientes C ON (O.ClienteID = C.ClienteID)
LEFT JOIN Empleados E ON (O.EmpleadoID = E.EmpleadoID)
LEFT JOIN Mensajeros M ON (O.MensajeroID = M.MensajeroID);
```

Para obtener el detalle completo solicitado, se combinaron cuatro tablas distintas mediante tres LEFT JOIN. Las tablas son: 'Ordenes', 'Clientes', 'Empleados' y 'Mensajeros'.

¿Cuántos productos de cada tipo se han vendido en cada sucursal?

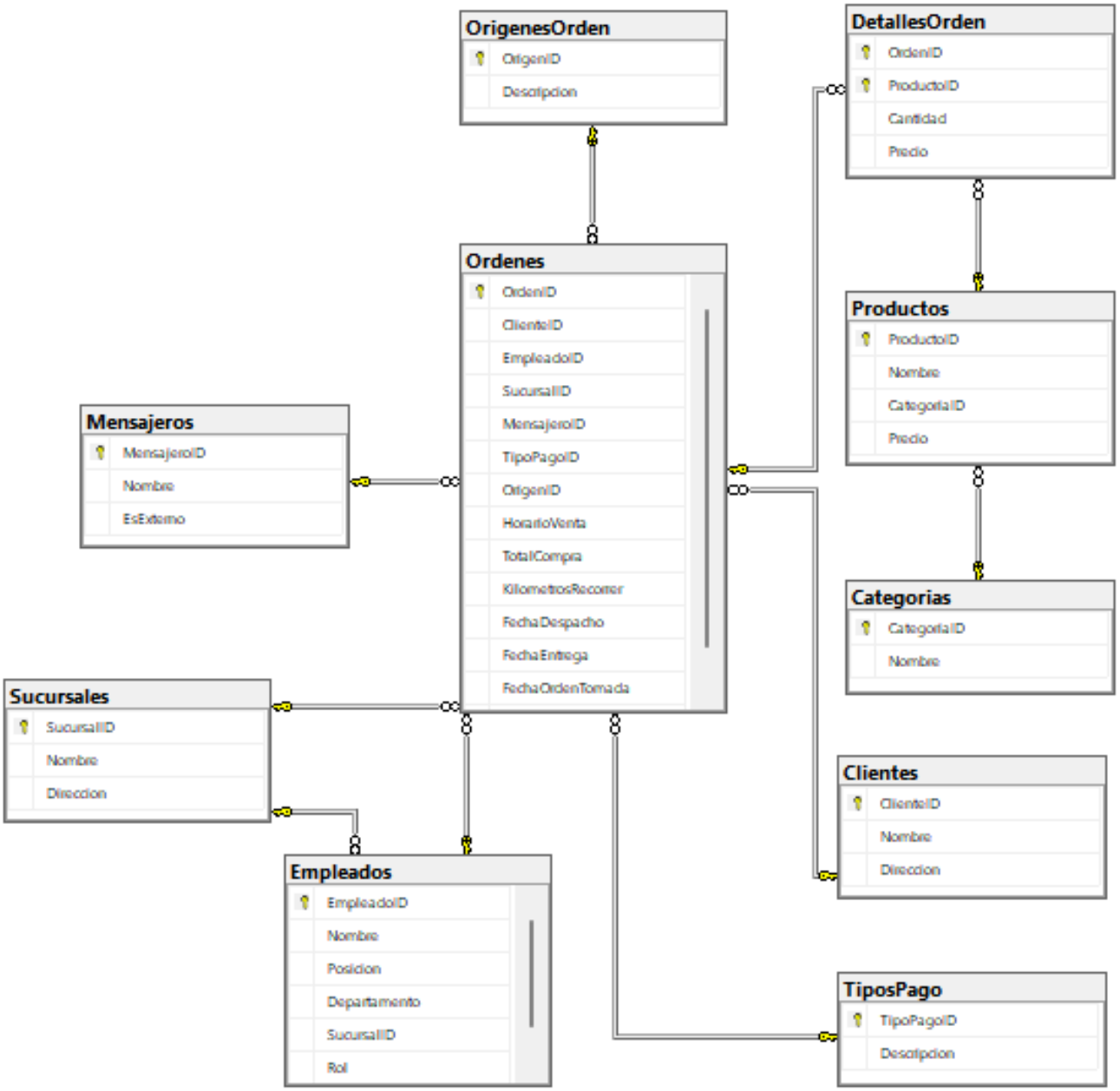
	Sucursal	Producto	ProductosVendidos
1	Sucursal Central	Cheeseburger	5
2	Sucursal Central	Helado de Vainilla	4
3	Sucursal Central	Pizza Margarita	4
4	Sucursal Central	Pizza Pepperoni	4
5	Sucursal Central	Hamburguesa Deluxe	3
6	Sucursal Central	Ensalada César	2
7	Sucursal Central	Ensalada Griega	2
8	Sucursal Central	Helado de Chocolate	1

```
SELECT S.Nombre Sucursal, P.Nombre Producto, SUM(DO.Cantidad) ProductosVendidos
FROM Ordenes O
INNER JOIN DetallesOrden DO ON (O.OrdenID = DO.OrdenID)
INNER JOIN Productos P ON (DO.ProductoID = P.ProductoID)
INNER JOIN Sucursales S ON (O.SucursalID = S.SucursalID)
GROUP BY S.Nombre, P.Nombre
ORDER BY ProductosVendidos DESC;
```

La query está preparada para que en un futuro, cuando se ingresen más datos, se diferencien las cantidades de productos vendidos por sucursal, por lo pronto **solo hay productos vendidos en la Sucursal Central.**

RESULTADOS Y CONSULTAS:

ESQUEMA RELACIONAL



Se puede observar un ciclo dentro del esquema, protagonizado por las tablas Ordenes, Empleados y Sucursales. Esto ocurre porque las Sucursales adoptan una posición como entidad independiente, y se relaciona directamente con Ordenes y Empleados.

Consultas finales:

- ¿Cuál es el tiempo promedio desde el despacho hasta la entrega de los pedidos por los mensajeros?

PromedioTiempoEntrega
30

```
SELECT AVG(DATEDIFF (MINUTE, FechaDespacho, FechaEntrega)) PromedioTiempoEntrega
FROM Ordenes
WHERE MensajeroID IS NOT NULL;
```

Hallazgos clave:

El tiempo de entrega estándar, en general, no debe ser más de 30 minutos, esto puede variar dependiendo de la complejidad del pedido. Los datos actualmente nos muestran que la compañía está situada en el límite estándar.

Recomendaciones estratégicas:

Para poder mejorar el tiempo promedio se recomienda monitorear el tiempo de pedido promedio a diario para identificar tendencias e identificar áreas de mejora.

Analizar el tiempo de pedido promedio por ubicación para determinar si ciertas áreas son más eficientes que otras.

Comparar el tiempo de pedido promedio con los puntos de referencia de la industria para garantizar que el negocio cumpla con las expectativas del cliente.

Consultas finales:

- ¿Qué canal de ventas genera más ingresos?

Canal	TotalVenta
Presencial	140.00

```
SELECT TOP 1 OO.Descripcion Canal, SUM(O.TotalCompra) TotalVenta
 FROM Ordenes O
 INNER JOIN OrigenesOrden OO ON(O.OrigenID = OO.OrigenID)
 GROUP BY OO.Descripcion
 ORDER BY TotalVenta desc
```

Hallazgos clave:

Al ser el canal presencial el que más vende, esto nos genera puntos de inflexión en los demás canales, en los cuales se va a tener que revisar estrategias de venta.

Recomendaciones estratégicas:

Ya sabiendo que el tiempo de entrega está en el límite estandarizado, hay que hacerle más énfasis en mejorarlo, ya que el canal presencial es el que menos tiene que fallar en ese aspecto. No hay que descuidar a la gente que opta por el canal presencial

Hay que realizar una investigación sobre qué es lo que le impide a los demás canales igualar al presencial.

Consultas finales:

- ¿Cuál es el volumen de ventas promedio gestionado por empleado?

	Empleado	VolumenVentaPromedio
1	Ethan Martinez	95.00
2	Olivia García	85.00
3	Jane Smith	75.00
4	Lucas Miller	65.00
5	Tom Brown	55.00
6	John Doe	50.00
7	Emma Davis	45.00
8	Alice Johnson	30.00
9	Bill Jones	20.00

```
SELECT E.Nombre Empleado, CAST(AVG(TotalCompra) AS DECIMAL (10,2)) VolumenVentaPromedio
FROM Ordenes O
INNER JOIN Empleados E ON (O.EmpleadoID = E.EmpleadoID)
GROUP BY E.Nombre
ORDER BY VolumenVentaPromedio DESC;
```

Hallazgos clave:

Podemos observar que hay mucha diferencia de volumen de venta promedio entre empleados, esto obliga a la empresa a averiguar por qué se producen estos desniveles de rendimiento.

Recomendaciones estratégicas:

Realizar rotaciones de horario entre empleados para verificar cómo varía su rendimiento, así averiguar si la hora tiene algo que ver con el caudal de ventas de cada empleado

Promover la comunicación entre empleados y la empresa para averiguar si sus rendimientos arrastran algún condicionamiento.

Consultas finales:

- ¿Cómo varía la demanda de productos a lo largo del día?

	Horario	Producto	Demanda
1	Mañana	Cheeseburger	5
2	Mañana	Helado de Vainilla	4
3	Mañana	Pizza Margarita	4
4	Mañana	Pizza Pepperoni	4
5	Mañana	Hamburguesa Deluxe	3
6	Mañana	Ensalada César	2
7	Mañana	Ensalada Griega	2
8	Mañana	Helado de Chocolate	1

```
SELECT O.HorarioVenta Horario, P.Nombre Producto, SUM(DO.Cantidad) Demanda
FROM Ordenes O
INNER JOIN DetalleSOrden DO ON (O.OrdenID = DO.OrdenID)
INNER JOIN Productos P ON (DO.ProductoID = P.ProductoID)
GROUP BY O.HorarioVenta, P.Nombre
ORDER BY O.HorarioVenta, Demanda DESC
;
```

Hallazgos clave:

Las ventas están concentradas altamente en la mañana. Las Cheeseburger lideran la demanda actual.

En la consulta anterior se verifica el volumen de ventas promedio por empleado, sabiendo que las ventas se concentran plenamente en la mañana, eso nos dice que el horario no es la razón del bajo rendimiento de los empleados ubicados en la parte inferior de la tabla.

Recomendaciones estratégicas:

Realizar promociones para promover las ventas en horarios distintos a la mañana.

Revisar si los precios de los productos menos demandados son correctos en base a lo que ofrece la competencia.

Generar nuevos productos o combos en base a lo más demandado.

Consultas finales:

- ¿Cómo se comparan las ventas mensuales de este año con el año anterior?

	Año	Mes	Venta
1	2023	9	95.00
2	2023	8	85.00
3	2023	2	75.00
4	2023	7	65.00
5	2023	5	55.00
6	2023	1	50.00
7	2023	6	45.00
8	2023	4	30.00
9	2023	3	20.00

```
SELECT Year(FechaOrdenTomada) 'Año', MONTH (FechaOrdenTomada) Mes,
  SUM (TotalCompra) Venta
FROM Ordenes
WHERE YEAR (FechaOrdenTomada) >= 2023 AND YEAR (FechaOrdenTomada) <= 2024
GROUP BY YEAR (FechaOrdenTomada), MONTH (FechaOrdenTomada)
ORDER BY Venta DESC;
```

Hallazgos clave:

Como no se registran ventas el año anterior, se puede concluir que el negocio está actualmente registrando su mayor volumen de ventas.

Recomendaciones estratégicas:

Realizar un estudio con el enfoque de averiguar la razón de la decadencia de ventas en los meses 3 y 4.

Consultas finales:

- ¿Qué porcentaje de clientes son recurrentes versus nuevos clientes cada mes?

	Nombre	NumeroOrdenes
1	Cliente Cinco	1
2	Cliente Cuatro	1
3	Cliente Dos	1
4	Cliente Nueve	1
5	Cliente Ocho	1
6	Cliente Seis	1
7	Cliente Siete	1
8	Cliente Tres	1
9	Cliente Uno	1

```
SELECT C.Nombre, COUNT(*) NumeroOrdenes
FROM Ordenes O
INNER JOIN Clientes C ON (O.ClienteID = C.ClienteID)
GROUP BY C.Nombre
ORDER BY NumeroOrdenes DESC
;
```

Hallazgos clave:

Todos los clientes actuales de la empresa son clientes nuevos

Recomendaciones estratégicas:

Ofrecer a los clientes cupones de descuento para sus próximas compras, esto nos sirve para afianzar las relaciones con el cliente y mantenerlo fiel a la empresa.

Carrera: Data Analytics

Módulo 2

Optimización y sostenibilidad

La primera medida que se tomó fue la de brindarle espacio de almacenamiento suficiente en el disco a la base de datos y al logfile. Esto nos garantiza sostenibilidad a futuro en caso de que la base de datos crezca lo suficiente.

Se implementaron muchas estrategias de optimización. Lo fundamental en esto fue mejorar el rendimiento y la eficiencia en el manejo de datos. Los campos de las columnas fueron formateados con criterio, en base a la naturaleza y utilidad de los datos a ingresar. Esto nos facilita su manipulación y análisis.

Para preservar la integridad de los datos en la eliminación de registros requerida, se optó por desactivar, eliminar los datos y luego reactivar la foreign key pertinente. Esto permitió no afectar a las otras tablas relacionadas durante la eliminación.

Se manejaron correctamente los datos nulos estableciendo restricciones previas donde se ameritaba. Esto se realizó con la sentencia NOT NULL que nos impide insertar datos nulos.

La tabla DetallesOrdenes se llevó una de las mejores optimizaciones, fue creada con una clave primaria compuesta la cual nos da la ventaja con su capacidad para garantizar la integridad de los datos dentro de una base de datos. Al incorporar varias columnas en la clave principal, resulta más difícil ingresar al sistema datos duplicados o inconsistentes. Las claves compuestas también pueden mejorar significativamente el rendimiento de las operaciones de la base de datos.

Además se generó velocidad y eficiencia en consultas formateando los datos arrojados con CAST o FORMAT, para mantener dos decimales en los datos consultados.

Desafíos y soluciones

Hubieron dos desafíos fundamentales para el éxito del proyecto:

- Lograr análisis y recomendaciones tan potentes como las entregadas para la empresa, contando con poca cantidad de datos. Esto lo solucione visualizando los datos de muchas maneras y dándole a los datos una estructura bien armada con relaciones bien forjadas entre tablas.
- Brindar los resultados de forma clara y concisa para cualquier persona. Se llevó a cabo manteniendo un lenguaje general para todo público, dándole un orden al código e informe para que nunca pierda coherencia. También se utilizaron anotaciones a modo de guía.

Haber resuelto estas dos problemáticas de forma correcta garantizo el éxito del proyecto.

Reflexión personal

Adquirí muchas habilidades nuevas:

- Encarar problemas y escenarios nuevos.
- Crear y manipular bases de datos de grandes volumenes.
- Extraer información y generar insights mediante consultas complejas.

Por otro lado pulí mi visión a la hora de tomar decisiones tanto de negocio como de workflow. También generé una forma de trabajo eficiente desglosando las tareas que debia realizar. Todo esto gracias al entorno que plantea el proyecto.

Respondiendo las preguntas, pienso que encararía desde un principio todas las tareas con mi nueva capacidad para optimizar tanto mi tiempo, la eficiencia e integridad de la base de datos.

EXTRA CREDIT

Se agregó información extra mediante la herramienta mockaroo (generador de datos ficticios https://www.mockaroo.com/).

INSERT INTO Productos (Nombre, CategoriaID, Precio) VALUES

('Wine - Jaboulet Cotes Du Rhone',3,9),

('Muffins - Assorted',5,2),

('Tomatoes - Vine Ripe, Yellow',8,15),

('Mushroom - Morel Frozen',8,10),

('Beef - Tenderloin - Aa',9,13),

('Wine - Toasted Head',3,9),

('Crackers - Graham',5,10),

('Muffin Orange Individual',6,4),

('Coffee Decaf Colombian',6,8),

('Pie Filling - Apple',2,8),

('Cheese - Perron Cheddar',2,14),

('Water - Mineral Natural',3,2),

('Pastry - Baked Scones - Mini',2,11),

('Beef - Rouladin, Sliced',9,10),

('Appetizer - Seafood Assortment',3,1),

('Juice Peach Nectar',3,12),

('Pasta - Cheese / Spinach Bauletti',1,11),

('Bacardi Mojito',3,10),

('Egg - Salad Premix',4,9),

('Fruit Salad Deluxe',4,8),

('Crackers - Melba Toast',5,8),

('Roe - Flying Fish',9,4),

('Trout - Smoked',9,13),

('Cookies - Oreo, 4 Pack',5,13),

('Wine - Stoneliegh Sauvignon',3,10),

('Beer - Steamwhistle',3,8),

('Bread - White, Sliced',6,2),

('Bread - Petit Baguette',6,3),

('Wine - Sake',3,11);

Así se ve la tabla con los datos nuevos:

	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
9	Wine - Jaboulet Cotes Du Rhone	Bebidas
10	Muffins - Assorted	Desayunos
11	Tomatoes - Vine Ripe, Yellow	Comida Vegana
12	Mushroom - Morel Frozen	Comida Vegana
13	Beef - Tenderloin - Aa	Carnes
14	Wine - Toasted Head	Bebidas
15	Crackers - Graham	Desayunos
16	Muffin Orange Individual	Cafetería
17	Coffee Decaf Colombian	Cafetería
18	Pie Filling - Apple	Postres
19	Cheese - Perron Cheddar	Postres
20	Water - Mineral Natural	Bebidas
21	Pastry - Baked Scones - Mini	Postres
22	Beef - Rouladin, Sliced	Carnes
23	Appetizer - Seafood Assortment	Bebidas
24	Juice Peach Nectar	Bebidas
25	Pasta - Cheese / Spinach Baul...	Comida Rápida
26	Bacardi Mojito	Bebidas
27	Egg - Salad Premix	Ensaladas
28	Fruit Salad Deluxe	Ensaladas
29	Crackers - Melba Toast	Desayunos
30	Roe - Flying Fish	Carnes
31	Trout - Smoked	Carnes
32	Cookies - Oreo, 4 Pack	Desayunos
33	Wine - Stoneliegh Sauvignon	Bebidas
34	Beer - Steamwhistle	Bebidas
35	Bread - White, Sliced	Cafetería
36	Bread - Petit Baguette	Cafetería
37	Wine - Sake	Bebidas

Nuestra base de datos ahora contiene 37 productos en total.

Gracias a esto ahora podemos ofrecer una amplia variedad de productos para todas las exigencias, esperando un aumento en las ventas.

¿Cuáles son los precios de los productos y a qué categorías pertenecen?

	Precio	Nombre Producto	Nombre Categoria
1	15.00	Tomatoes - Vine Ripe, Yellow	Comida Vegana
2	14.00	Cheese - Perron Cheddar	Postres
3	13.00	Cookies - Oreo, 4 Pack	Desayunos
4	13.00	Beef - Tenderloin - Aa	Carnes
5	13.00	Trout - Smoked	Carnes
6	12.99	Pizza Pepperoni	Pizzas
7	12.00	Juice Peach Nectar	Bebidas
8	11.99	Pizza Margarita	Pizzas
9	11.00	Pastry - Baked Scones - Mini	Postres
10	11.00	Pasta - Cheese / Spinach Bauletti	Comida Rápida
11	11.00	Wine - Sake	Bebidas
12	10.99	Hamburguesa Deluxe	Comida Rápida
13	10.00	Beef - Rouladin, Sliced	Carnes
14	10.00	Mushroom - Morel Frozen	Comida Vegana
15	10.00	Crackers - Graham	Desayunos
16	10.00	Wine - Stoneliegh Sauvignon	Bebidas
17	10.00	Bacardi Mojito	Bebidas
18	9.00	Wine - Toasted Head	Bebidas
19	9.00	Wine - Jaboulet Cotes Du Rhone	Bebidas
20	9.00	Egg - Salad Premix	Ensaladas
21	8.99	Cheeseburger	Comida Rápida
22	8.00	Coffee Decaf Colombian	Cafetería
23	8.00	Beer - Steamwhistle	Bebidas
24	8.00	Fruit Salad Deluxe	Ensaladas
25	8.00	Pie Filling - Apple	Postres
26	8.00	Crackers - Melba Toast	Desayunos
27	6.99	Ensalada Griega	Ensaladas
28	5.99	Ensalada César	Ensaladas
29	4.00	Muffin Orange Individual	Cafetería
30	4.00	Roe - Flying Fish	Carnes
31	3.00	Bread - Petit Baguette	Cafetería
32	2.99	Helado de Chocolate	Helados
33	2.99	Helado de Vainilla	Helados
34	2.00	Muffins - Assorted	Desayunos
35	2.00	Bread - White, Sliced	Cafetería
36	2.00	Water - Mineral Natural	Bebidas
37	1.00	Appetizer - Seafood Assortment	Bebidas

El precio más alto es de 15, el más bajo 1. Hay un rango de precios muy variado.

Al incorporar productos nuevos de alto costo, se busca captar todo tipo de público, desde gente adinerada hasta gente de mediana clase

¿Qué query se utilizó para la última tabla?

```
SELECT CAST(AVG(P.Precio)AS DECIMAL(10,2)) AS Precio, P.Nombre 'Nombre Producto', C.Nombre 'Nombre Categoria'
FROM Productos P
LEFT JOIN Categorias C ON (P.CategoriaID = C.CategoriaID)
GROUP BY P.Nombre, C.Nombre
ORDER BY CAST(AVG(P.Precio)AS DECIMAL(10,2)) DESC;
```

Se hizo una query optimizada con CAST para mantener el formato de hasta dos decimales.

¿Qué cantidad de productos hay por categoría?

	Cantidad de productos por categoría	Nombre Categoria
1	9.00	Bebidas
2	4.00	Cafetería
3	4.00	Carnes
4	4.00	Desayunos
5	4.00	Ensaladas
6	3.00	Comida Rápida
7	3.00	Postres
8	2.00	Comida Vegana
9	2.00	Helados
10	2.00	Pizzas

Como se puede observar ahora todas las categorías de nuestra base de datos contienen productos.

Las bebidas tienen la mayor variedad de artículos.

```
SELECT CAST(COUNT(C.CategoriaID)AS DECIMAL(10,2)) AS 'Cantidad de productos por categoría', C.Nombre 'Nombre Categoria'
FROM Categorias C
LEFT JOIN Productos P ON (P.CategoriaID = C.CategoriaID)
GROUP BY C.Nombre
ORDER BY CAST(COUNT(C.CategoriaID)AS DECIMAL(10,2)) DESC;
```

Mejora:

Una mejora que puede hacerse es la de indexar el campo que suele ser más usado en las querys. Este campo es TotalCompra de la tabla Ordenes.

La query usada fue la siguiente:

CREATE INDEX IndiceTotalVentas

ON Ordenes (TotalCompra);

Un índice es una estructura de disco asociada con una tabla o una vista que acelera la recuperación de filas de la tabla o de la vista.