SQL

Alumno: Omar Miguel Cabbad

Profesor: Emilio D Aguero

Comisión: 81820

Tabla de contenido

1. Introducción

En este mundo globalizado, las empresas deben estudiar cuidadosamente los caminos a seguir para consolidarse y diferenciarse de la competencia. Para ello, enfrentan desafíos como conocer el comportamiento de sus clientes, optimizar sus estrategias de ventas y gestionar eficientemente sus productos y sucursales. Un análisis adecuado de las ventas y la segmentación de clientes es clave para mejorar la rentabilidad y la satisfacción del consumidor.

En este proyecto se utilizará un dataset obtenido de Kaggle, una plataforma online ampliamente reconocida donde se comparten datasets, se realizan competencias de machine learning y se aprende sobre ciencia de datos. El dataset puede descargarse desde el siguiente enlace: https://www.kaggle.com/datasets/faresashraf1001/supermarket-sales

Este conjunto de datos contiene información detallada sobre ventas en supermercados, incluyendo datos de facturación, sucursal, ciudad, tipo de cliente, género, línea de productos, precios, cantidades, métodos de pago y márgenes de ganancia. Esto permitirá desarrollar análisis para identificar patrones de compra, evaluar el rendimiento de productos y clientes, y tomar decisiones estratégicas que contribuyan al logro de los objetivos empresariales.

2. Descripción de la temática de los datos

La temática de la base de datos se centra en el funcionamiento de una cadena de supermercados y en cómo diversos factores influyen en su desempeño diario. La información incluye datos detallados sobre ventas, sucursales, ciudades, tipos de clientes, géneros, líneas de productos, precios, cantidades, métodos de pago y márgenes de ganancia.

El propósito de esta base de datos es organizar toda esta información de manera estructurada, permitiendo analizar el comportamiento de las ventas según diferentes variables, como el tipo de cliente, la ubicación, el producto o la forma de pago.

Contar con esta base de datos facilita el análisis del negocio desde múltiples perspectivas: identificar patrones de compra, evaluar el rendimiento de productos y sucursales, detectar oportunidades para mejorar la experiencia del cliente y optimizar las estrategias comerciales.

A través del uso de SQL, será posible realizar consultas que ayuden a comprender mejor la operación diaria, generar reportes relevantes para distintas áreas de la empresa y apoyar la toma de decisiones para aumentar la eficiencia y rentabilidad del supermercado.

3. Problemática

Es importante implementar una base de datos para resolver problemáticas comunes en la gestión de ventas y operaciones en supermercados. En el mundo actual, las empresas necesitan manejar de forma eficiente la información relacionada con las ventas, clientes, productos y sucursales, optimizando los procesos desde la compra hasta la atención al cliente. Sin embargo, enfrentan dificultades para analizar el comportamiento de los consumidores y prever la demanda de productos.

Sin una base de datos centralizada y bien estructurada, pueden presentarse problemas que afectan el desempeño del negocio, tales como:

- Falta de datos consolidados sobre ventas por sucursal, tipo de cliente y línea de productos.
- Dificultad para evaluar el impacto de variables como métodos de pago, horarios o características del cliente.
- Limitada capacidad para tomar decisiones comerciales basadas en información precisa y actualizada.
- Necesidad de mejorar la formación del personal para el manejo adecuado de tecnologías y análisis de datos.

4. Objetivos

Objetivo General:

El objetivo general del proyecto es lograr una mejor gestión de ventas y desempeño de los supermercados mediante el análisis de datos históricos de transacciones y variables asociadas al cliente y al producto. Esto permitirá apoyar decisiones estratégicas, optimizar recursos y mejorar la planificación comercial y operativa de cada sucursal.

Objetivos Específicos:

• Analizar los datos históricos de ventas por sucursal, línea de productos, tipo de cliente y método de pago.
• Evaluar el impacto de factores como horarios, días de la semana, promociones y características del cliente sobre las ventas.
• Identificar patrones de comportamiento de compra que permitan optimizar la planificación de productos y stock.
• Integrar información de distintas áreas (ventas, logística, contabilidad) para apoyar la toma de decisiones estratégicas.
• Proponer estrategias basadas en los análisis realizados que contribuyan a mejorar la eficiencia, rentabilidad y experiencia del cliente en los supermercados.

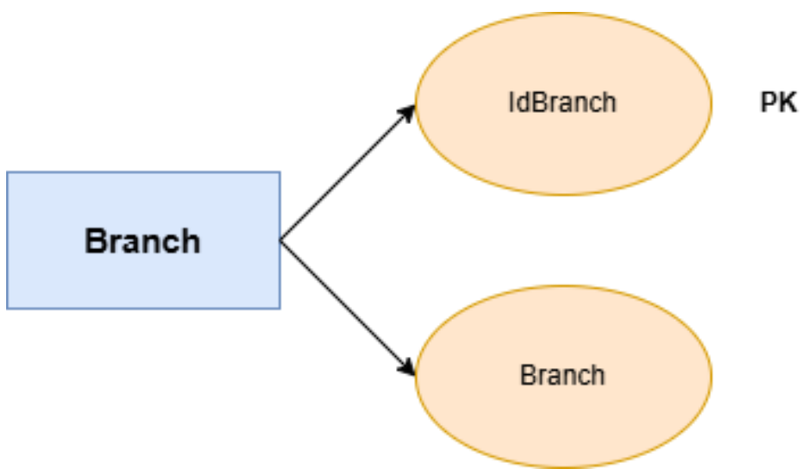
5. Diagrama Entidad-Relación

El modelo Entidad-Relación (ER) es un modelo de datos utilizado en el diseño de bases de datos. Representa entidades y sus relaciones en el mundo real.

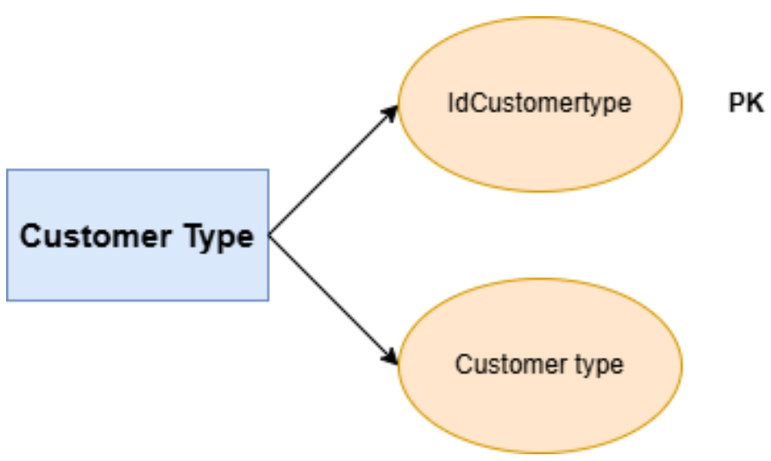
Para garantizar la correcta organización de los datos y permitir establecer relaciones entre las diferentes entidades del modelo, se incorporará un identificador único (ID**)** en cada tabla principal permitiendo una clave primaria en cada tabla (PK), asegurando un registro único, creación de claves foráneas (FK) y logrando una integridad diferencial.

Para nuestro proyecto tendremos lo siguiente:

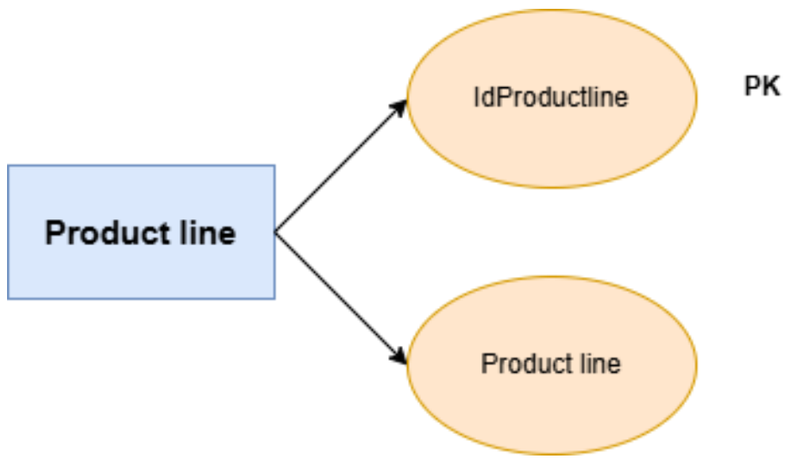
- La tabla **BRANCH** contiene la información sobre las distintas sucursales en donde se comercializa los productos. Incluye un ID para cada sucursal y el nombre de la sucursal.



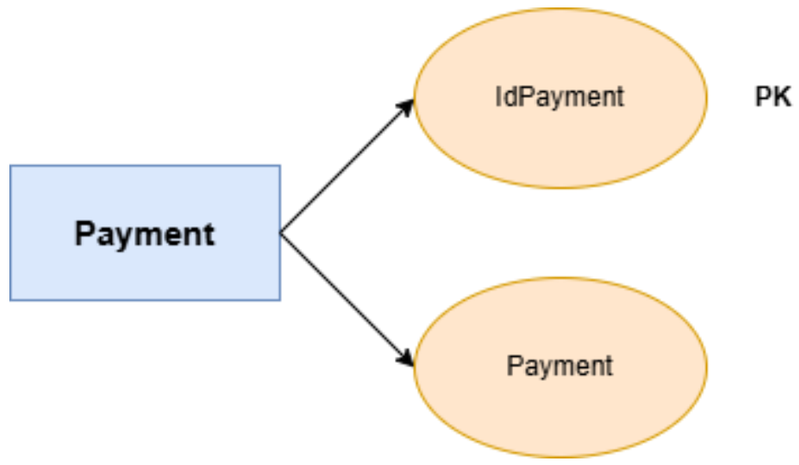
- La tabla **CUSTOMER TYPE** contiene la información sobre el tipo de cliente que compra el producto en el supermercado. Incluye un ID para cada tipo de cliente y el nombre del tipo de cliente.



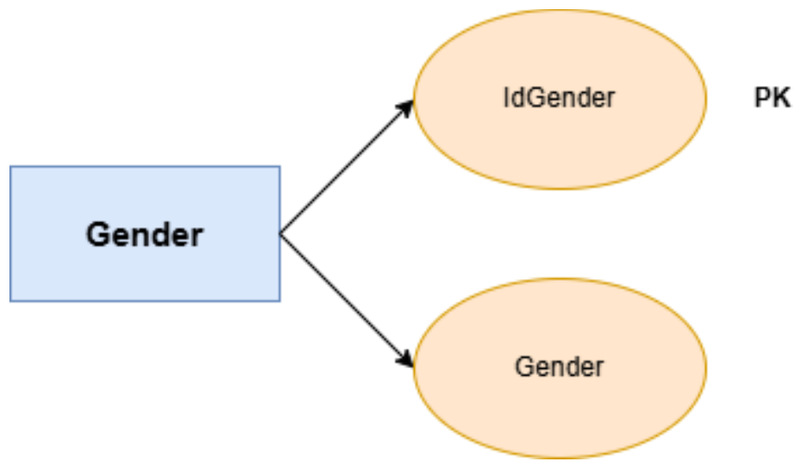
- La tabla **PRODUCT LINE** contiene la información sobre qué línea de producto compra el cliente en el supermercado. Incluye un ID para cada tipo de cliente y el nombre del tipo de cliente.



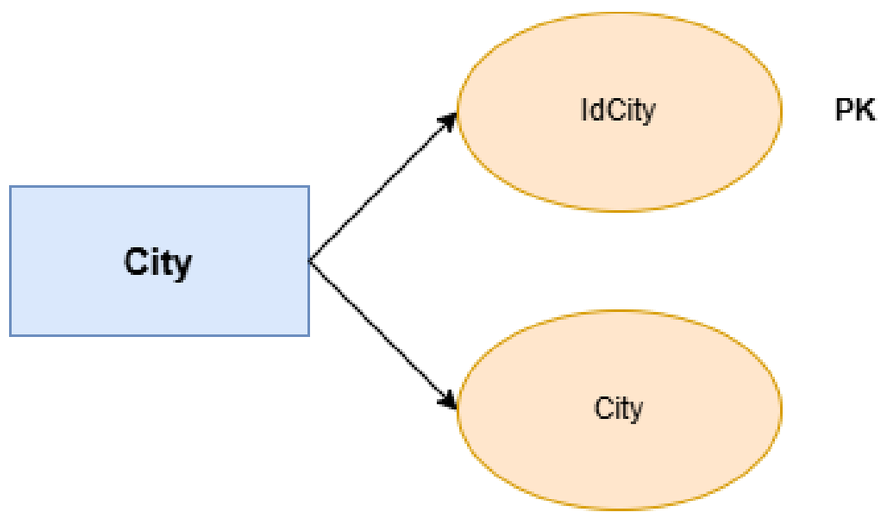
- La tabla **PAYMENT** contiene la información sobre la forma de pago en el cual se adquieren los productos del supermercado. Incluye un ID para cada método de pago y el nombre de dicho método.



- La tabla **Gender** brinda información sobre los géneros que adquieren productos en el supermercado. Incluye un ID para el tipo de género y el nombre del tipo de género.



- La tabla **City** brinda información sobre las ciudades donde se comercializan los productos del supermercado. Incluye un ID para el tipo de ciudad y el nombre de la ciudad.



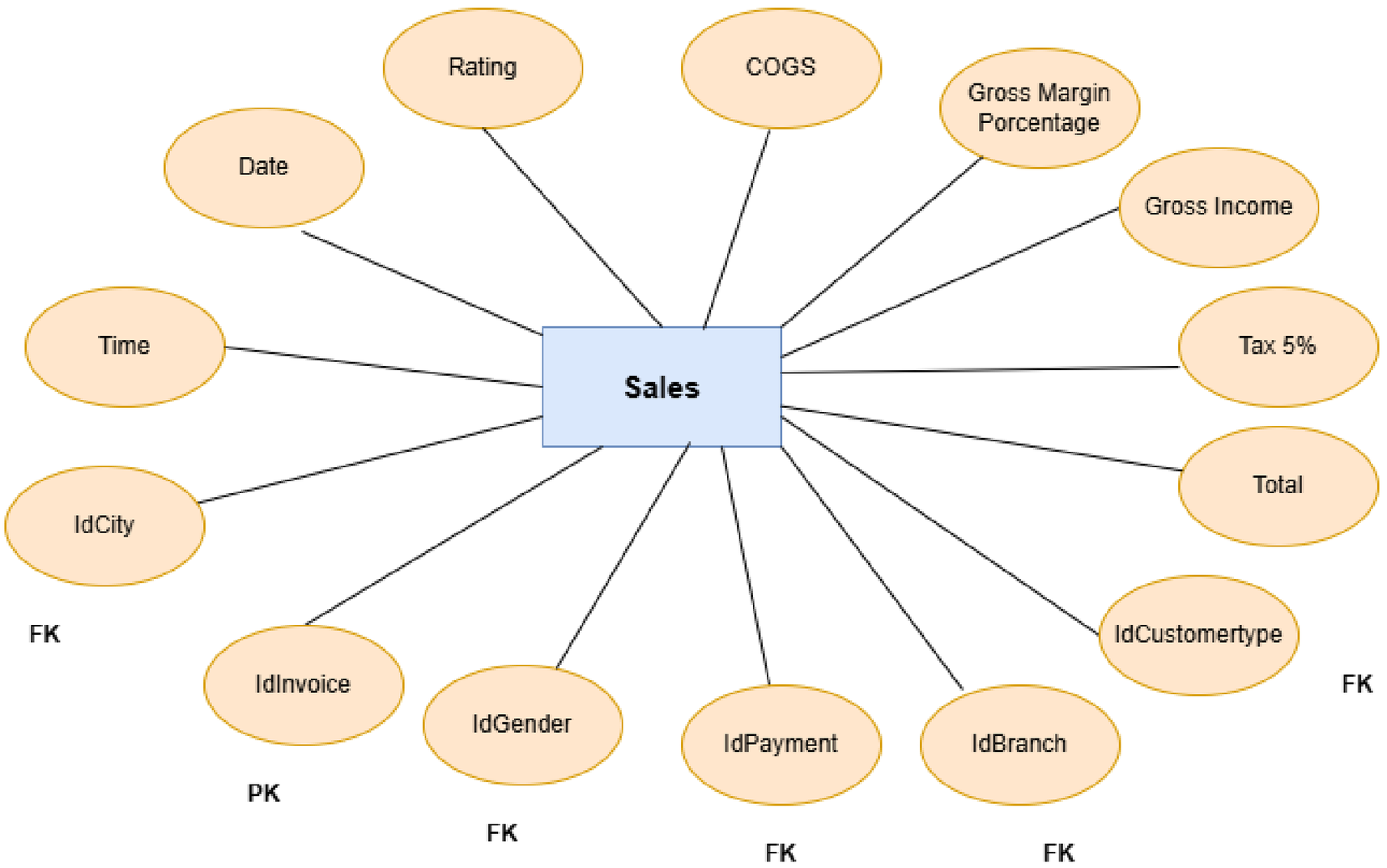
Modelo Cabecera-detalle

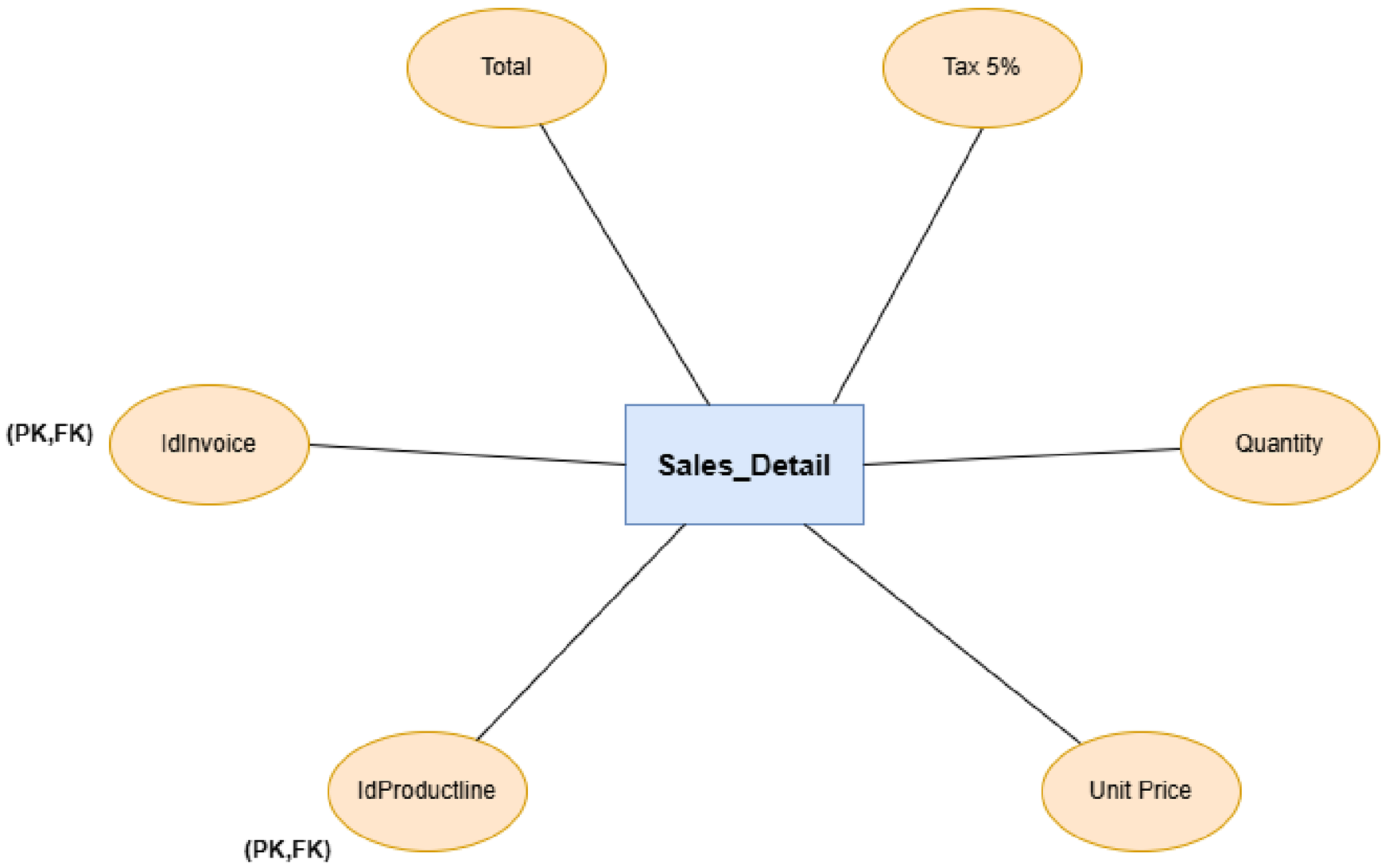
Se optó por dividir toda la información en dos tablas (Sales y Sales_Detail) para evitar redundancia en la factura y que la misma pueda tener múltiples productos. De esta manera la factura queda mucho mal limpia y no tenga que repetirse, por ejemplo, la fecha en cada producto que compro el cliente.

La relación que existe entre ambas es de 1:N. Esto significa que por cada registro de la tabla Sales, pueden existir múltiples registros de la tabla Sales_Detail. El campo que une a ambas es IDInvoice.

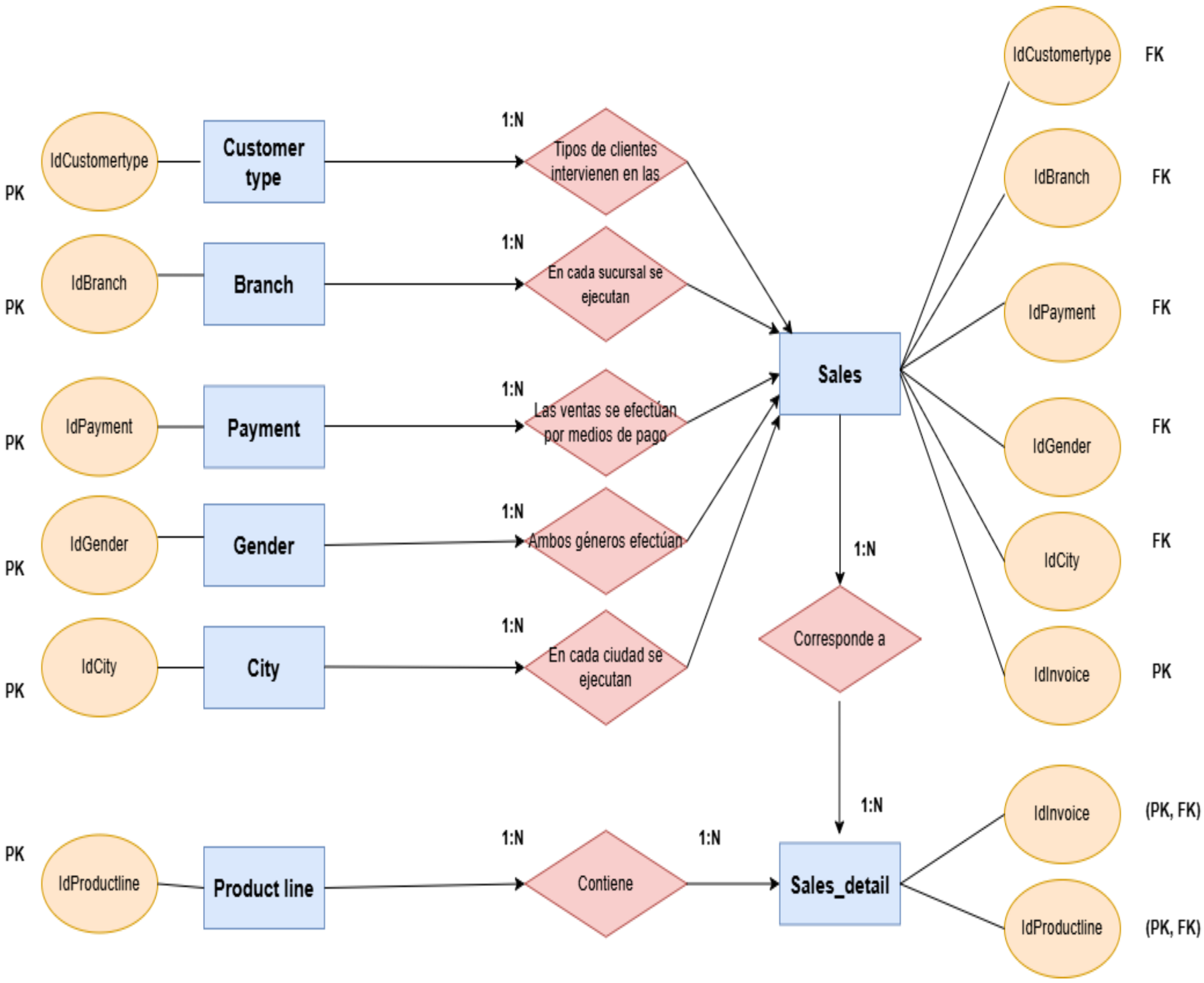
En Sales el identificador único es IDInvoice, mientras que en Sales_Detail se aplicó una llave compuesta porque en cada producto que compra ya sea uno o varios, tendremos en cada “renglón” un identificador único de la factura con su respectivo producto evitando que se duplique el mismo producto dentro de una misma factura.

Los datos de transacción total, como el COGS, Gross income y rating se ubicaron en la tabla Sales. En cambio, en la tabla de detalle guardamos los valores que pueden variar por cada ítem, como por ejemplo quantity y unit Price.




Total
Tax 5%
(PK,FK)
IdInvoice
Sales_Detail
Quantity
IdProductline
Unit Price
(PK,FK)

- El modelo Entidad-Relación (ER) final quedaría de la siguiente manera:



6. Listado de campos por tablas

En el siguiente apartado, procederemos a proporcionar un exhaustivo desglose de estructura de cada tabla, incluyendo la enumeración de sus columnas, la especificación detallada de los tipos de datos asignados a cada una de ellas y, además, la clara identificación de los tipos de clave que se han implementado en dichas tablas. Este análisis minucioso de la disposición y características de los datos permitirá una comprensión más profunda y completa de la base de datos en cuestión, brindando una visión integral de su diseño y funcionamiento.

Tabla: BRANCH		
Campos	**Tipo de datos**	**Claves**
IdBranch	INT	PK
Branch	VARCHAR	

Tabla: CUSTOMER TYPE		
Campos	**Tipo de datos**	**Claves**
IdCustomertype	INT	PK
Customer type	VARCHAR	

Tabla: PRODUCT LINE		
Campos	**Tipo de datos**	**Claves**
IdProductline	INT	PK
Product Line	VARCHAR	

Tabla: PAYMENT		
Campos	**Tipo de datos**	**Claves**
IdPayment	INT	PK
Payment	VARCHAR	

Tabla: GENDER		
Campos	**Tipo de datos**	**Claves**
IdGender	INT	PK
Gender	VARCHAR	

Tabla: SALES		
Campos	Tipo de datos	Claves
IdBranch	INT	FK
IdCustomertype	INT	FK
IdPayment	INT	FK
IdGender	INT	FK
IdInvoice	VARCHAR	PK
IdCity	INT	FK
Time	TIME	
Date	DATE	
Tax 5%	DECIMAL(10,3)	
Total	DECIMAL(10,3)	
COGS	DECIMAL(10,2)	
Gross Margin	DECIMAL(10,9)	
Gross Income	DECIMAL(10,3)	
Rating	DECIMAL(3,1)	

Tabla: SALES_DETAIL		
Campos	Tipo de datos	Claves
IdProductline	INT	(PK,FK)
IdInvoice	VARCHAR	(PK,FK)
Unit price	DECIMAL(10,2)	
Quantity	INT	
Tax 5%	DECIMAL(10,3)	
Total	DECIMAL(10,3)	

7. Ingeniería de Datos y Normalización

1. Origen y Extracción de Datos

El dataset original “Supermarket Sales” se utilizo de la plataforma Kaggle, el cual contiene datos históricos de ventas de una empresa de supermercados recolectados en tres sucursales distintas en un periodo de 3 meses.

Se llevo a cabo un almacenamiento local, donde fue descargado y procesado localmente desde la ruta C:\Users\carok\Desktop\Supermarket Sales.csv, realizando las modificaciones correspondientes para una importación exitosa al MySQL Workbench.

2. Fase de Transformación y Normalización (Excel)

Esta fue la etapa de mayor trabajo analítico. El objetivo no fue solo limpiar datos, sino rediseñar la estructura para pasar de un archivo plano a un modelo relacional

Se identificó las categorías que se repetían (Ciudad, Sucursal, Género, etc.) y las separé en tablas independientes. A cada elemento le asigné un Id numérico único logrando de esta manera profesionalizar la base y evitar errores de escritura

Utilicé la función BuscarV, donde reemplacé cada palabra por su ID correspondiente. Por ejemplo, donde decía "Member" ahora dice "1". Una vez hecho esto, convertí las fórmulas en valores fijos para que la tabla fuera estática.

Se realizo la estandarización de formatos.

Fechas: Cambié el formato original al estándar de base de datos AAAA-MM-DD.

Decimales: Realicé un "Buscar y Reemplazar" para convertir las comas en puntos, asegurando que MySQL no tuviera problemas al interpretar los valores monetarios.

El desafío de los decimales (Gross Margin % y Total): Durante la limpieza, el "Buscar y Reemplazar" estándar de Excel falló en columnas críticas como el porcentaje de margen. Para solucionar esto, implementé una solución técnica avanzada: utilicé la función =SUSTITUIR(celda; "."; ",") para forzar el cambio de formato, logrando que Excel reconociera los números correctamente. Para mantener la limpieza del archivo, oculté las columnas originales con errores y trabajé únicamente con las columnas procesadas.

3. Implementación y Carga en MySQL

Tablas Maestras (Script de Inserción): Para tablas como Gender, Payment, City, Branch, Customer_type y Product_line, se provee un script SQL de inserción manual (Carga_Datos_Cabbad.sql). Esto asegura que los IDs coincidan exactamente con la lógica de las funciones y procedimientos, sin depender del AUTO_INCREMENT, garantizando que cada registro tenga un valor predecible en cualquier entorno.

Tablas Transaccionales (Data Import Wizard): Para las tablas Sales y Sales_detail, se utilizó el Table Data Import Wizard para procesar los 1.000 registros del dataset original (archivo supermarket_sales.csv adjunto).

Limpieza y Optimización: Mediante comandos SQL post-importación, eliminé caracteres residuales y transformé las columnas a sus tipos definitivos: DECIMAL e INT, optimizando el rendimiento de la base de datos.

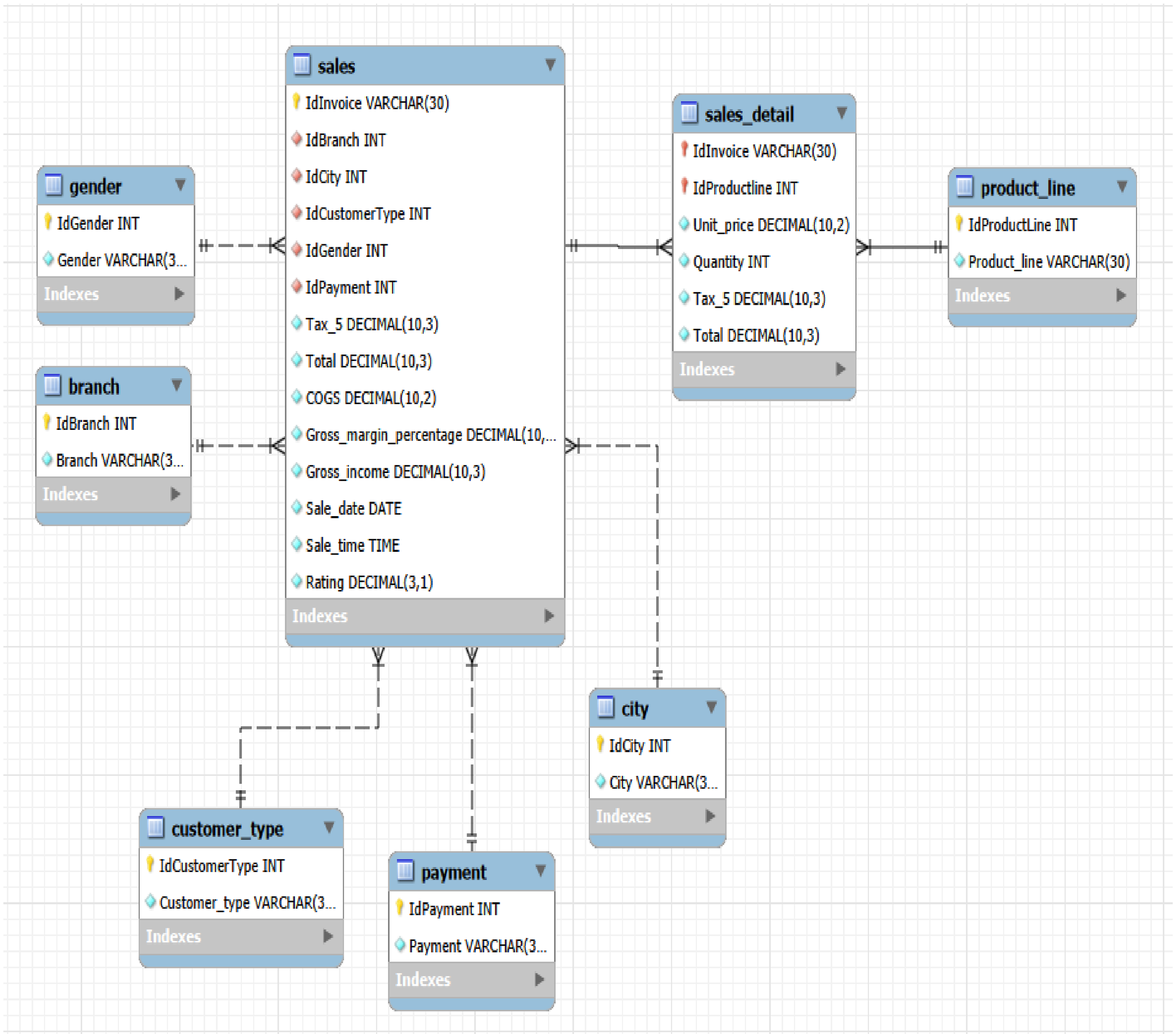
4. Arquitectura del Modelo: El Diagrama EER

Una vez cargadas las tablas, establecí las relaciones mediante Llaves Foráneas (Foreign Keys). El resultado es el EER Diagram (Diagrama Entidad-Relación) que se detalla a continuación:

Análisis del Diagrama:

- Esquema en Estrella: Tenemos la tabla Sales en el centro como nuestra Tabla de Hechos (Fact Table).
- Tablas de Dimensiones: A su alrededor, conectadas por flechas, están las tablas city, branch, product_line, gender, customer_type y payment.

- Relación Uno a Muchos: Las líneas indican que, por ejemplo, una ciudad puede tener muchas ventas registradas, pero cada venta solo puede pertenecer a una ciudad.
- Sales_Detail: Esta tabla está vinculada por el IdInvoice, permitiendo separar la información de la transacción general (sucursal, fecha, cliente) del desglose específico de los productos comprados.



8- Vistas

Estas tablas virtuales nos permitirán unificar la información de las distintas tablas en reportes simplificados. Su uso es clave para evitar la repetición de consultas complejas (JOINs), proteger datos sensibles y garantizar que la información se presente de forma organizada, segura y eficiente para la toma de decisiones.

1. Desempeño por Ciudad

Cuál es el desempeño de la ventas en las distintas ciudades?

La idea de esta vista es ver rápido qué ciudades están rindiendo mejor, logrando de esta manera saber cuál está vendiendo mas cantidad de productos. Se apoya en Sales para los números y en City para ponerle nombre a cada lugar.

-- El mayor desempeño representa a Naypyitaw con 110568.748, seguido por Yangon con 106200.409 y por ultimo Mandalay con 106197.709 --

2. Ventas de Enero en Mandalay

Cuanto se facturo en el primer mes de 2019 en la ciudad de Mandalay?

Esta es bien específica. La armé para ver cómo nos fue en el primer mes del año en una de las sucursales clave (Mandalay). Sirve para chequear si se llega a los objetivos del mes, por ejemplo. Cruza la tabla de Sales con la de City.

El total de facturación en enero en la sucursal que se encuentra en Mandalay es de 37176.068

3. Productos que más dejan dinero

Que Línea de producto da mayor margen de ganancia a la empresa (Gross income)?

Acá es donde vemos que se está vendiendo mejor. Nos muestra qué categorías de productos nos dejan más ganancia bruta, para saber qué conviene promocionar más. Junta Sales, Sales_Detail y Product_line.

Salud y belleza es el tipo de producto que genera mayor margen de ganancia con 8313.705

4. **Compras por Género**

¿Cuántas ventas se realizaron por cada género, incluyendo aquellos que no compraron nada?

Queremos saber quiénes nos compran más, si hombres o mujeres. Lo bueno es que, al usar un RIGHT JOIN, si llegáramos a tener un género que todavía no compró nada, igual aparecería en la lista con un cero, para que no nos falte nadie en el reporte. Se usa Sales y la tabla maestra Gender.

Las ventas de mujeres representan 501 mientras que los hombres 499

5. Ranking de Medios de Pago

¿Cuál es el metodo de pago que más ha recaudado? ¿cual fue el de mayor uso?

Esta es fundamental para entender cómo paga la gente. Nos permite ver si prefieren usar la billetera virtual (Ewallet), efectivo o tarjeta. Conecta la tabla de ventas Sales con la de Payment.

El método más utilizado fue la billetera electrónica con 345 operaciones. Sin Embargo, el que más recaudo fue el efectivo con 112206.610

9- Funciones

Nos permitirán automatizar cálculos específicos y lógicos dentro de las consultas, como el cálculo de impuestos o comisiones por venta. Su objetivo es reutilizar operaciones matemáticas en múltiples reportes, garantizando que el resultado sea siempre exacto y consistente.

1) ¿Cuántas unidades de producto se adquirieron según el tipo de factura?

Para responder esta pregunta, utilizamos una función para automatizar el cruce de datos de las tablas Quantity e IdInvoice permitiendo de esta manera obtener información que puede ser muy relevante para la gerencia ante cualquier imprevisto que pueda surgir ante una mal trabajo del empleado u error del sistema en general

2) ¿Cuál es el promedio de facturación total según el tipo de producto?

Utilizamos una función para automatizar el cruce de datos entre total, product_line e idproductline, en el que se hizo un puente entre las tablas sales, sales_detail e productline a través de join. Esto nos permitirá obtener información relevante para determinar campañas de marketing, donde centrarse y que productos comprar a nivel general.

10- Stored procedures

Los Stored Procedures (Procedimientos Almacenados) se utilizan para encapsular lógica de negocio compleja que el usuario o la aplicación pueden ejecutar a demanda. A diferencia de las consultas simples, estos permiten procesar parámetros y ejecutar código dinámico, automatizando tareas repetitivas y garantizando la consistencia de los reportes.

1) Reporte de Ventas Flexible: Se genera un reporte que permite al usuario decidir el orden de los datos en tiempo real mediante un Stored Procedure con SQL dinámico, facilitando la visualización personalizada de los 1,000 registros.

2) Reporte de Segmentación Avanzada: Se genera un reporte filtrado por categoría mediante un Stored Procedure que aplica una cláusula WHERE y un ORDER BY dinámico. Este proceso integra funciones personalizadas para calcular métricas en tiempo real (como totales e impuestos), permitiendo un análisis profundo de cada segmento del supermercado. De esta manera, el sistema automatiza el cruce de datos complejos entre Sales y ProductLine sin necesidad de escribir consultas manuales repetitivas.