

Informe Bases de Datos Relacionales

Harold Mosquera

Instituto Tecnológico del Putumayo

Facultad de Ingeniería y Ciencias Básicas

Índice

Índice	2
Resumen	3
Introducción	4
Metodología	5
Desarrollo del Informe.....	6
Consultas SQL.....	8
Subconsultas	10
Explicación de Consultas	11
Análisis y Discusión de Resultados.....	12
Conclusiones	13

Resumen

El informe describe y analiza una base de datos de una tienda, compuesta por tablas relacionadas como cliente, factura, producto, entre otras. Se detallaron las claves primarias y las relaciones entre las tablas, mostrando cómo se conectan los datos dentro del sistema.

Se realizaron diversas consultas SQL, incluyendo la obtención de clientes y sus facturas, y los productos vendidos en una factura específica. Los resultados reflejan correctamente las relaciones entre las tablas y permiten visualizar la información almacenada.

Una consulta adicional identificó el tipo de producto con el mayor y menor costo promedio, proporcionando un análisis clave sobre los costos de los productos en la tienda. Estos datos son útiles para tomar decisiones comerciales basadas en precios.

Introducción

Las bases de datos relacionales son una tecnología fundamental en el mundo de la informática y la gestión de datos. Introducidas por Edgar F. Codd en 1970, estas bases de datos se basan en el modelo relacional, que organiza los datos en tablas (o relaciones) que pueden ser fácilmente manipuladas y consultadas mediante el lenguaje SQL (Structured Query Language).

Una de las principales ventajas de las bases de datos relacionales es su capacidad para manejar grandes volúmenes de datos de manera eficiente y estructurada. Cada tabla en una base de datos relacional contiene filas (registros) y columnas (campos), donde cada columna representa un atributo de los datos y cada fila una instancia de esos datos. Esta estructura permite realizar consultas complejas y obtener información precisa y relevante de manera rápida.

Además, las bases de datos relacionales aseguran la integridad y consistencia de los datos a través de restricciones y reglas de integridad, como las claves primarias y foráneas, que garantizan que las relaciones entre las tablas sean válidas y coherentes. Esto es crucial para aplicaciones críticas donde la precisión de los datos es esencial.

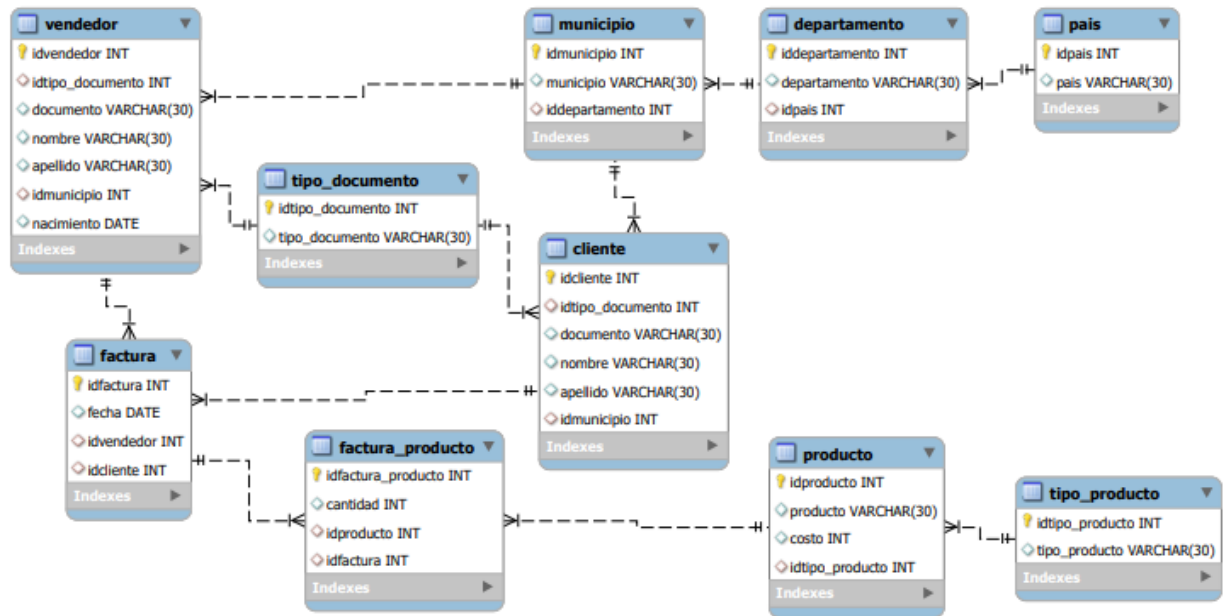
Metodología

La metodología seguida para este trabajo consistió en el análisis y optimización de consultas SQL utilizando JOINS y subconsultas, así como en la representación del modelo entidad-relación (ER). Se empleó HeidiSQL para la ejecución de consultas y análisis de rendimiento, y WorkBench para la creación del diagrama ER. Inicialmente, se identificaron las principales entidades y relaciones de la base de datos, y se diseñó un modelo ER que reflejara dichas conexiones. Posteriormente, se desarrollaron consultas básicas para cada entidad, seguidas de consultas más complejas con JOINS y subconsultas anidadas, necesarias para recuperar información de múltiples tablas. Las consultas fueron probadas en HeidiSQL, verificando tanto la exactitud de los resultados como su eficiencia, lo que permitió aplicar técnicas de optimización como el uso de índices. Finalmente, se documentaron los resultados, incluyendo el diagrama ER y ejemplos de las consultas desarrolladas, junto con recomendaciones para su mejora en futuras implementaciones.

Desarrollo del Informe

Figura 1.

Modelo entidad relación



La base de datos está compuesta por las siguientes tablas:

cliente: Registra la información de los clientes, incluyendo el tipo de documento, nombre, apellido y municipio de residencia.

departamento: Almacena los departamentos (regiones) y está relacionada con la tabla de municipios.

factura: Registra las ventas realizadas, asociadas a un cliente y un vendedor.

factura_producto: Almacena el detalle de los productos que forman parte de cada factura.

municipio: Representa los municipios y está relacionado con los departamentos.

pais: Guarda los países a los que pertenecen los departamentos.

producto: Almacena la información de los productos vendidos, incluyendo su costo y tipo.

tipo_documento: Registra los diferentes tipos de documentos identificativos de clientes y vendedores

tipo_producto: Clasifica los productos.

vendedor: Contiene la información de los vendedores.

Las claves principales y relaciones son las siguientes:

cliente está relacionado con factura por idcliente.

factura está relacionado con factura_producto por idfactura.

producto está relacionado con factura_producto por idproducto.

municipio está relacionado con departamento por iddepartamento.

departamento está relacionado con pais por idpais.

vendedor está relacionado con factura por idvendedor.

Consultas SQL

Consulta de clientes y sus facturas

Figura 2.

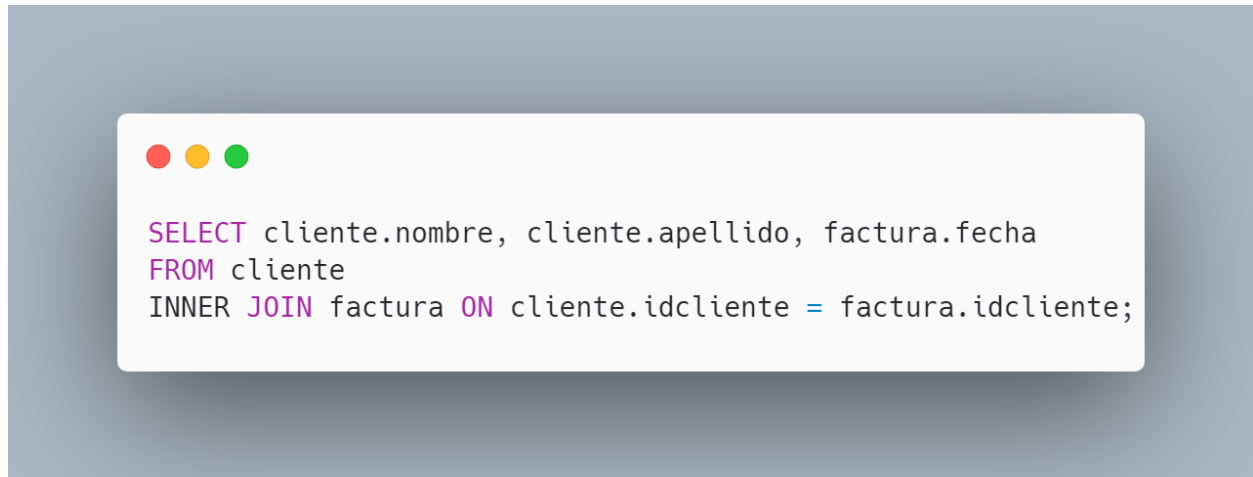
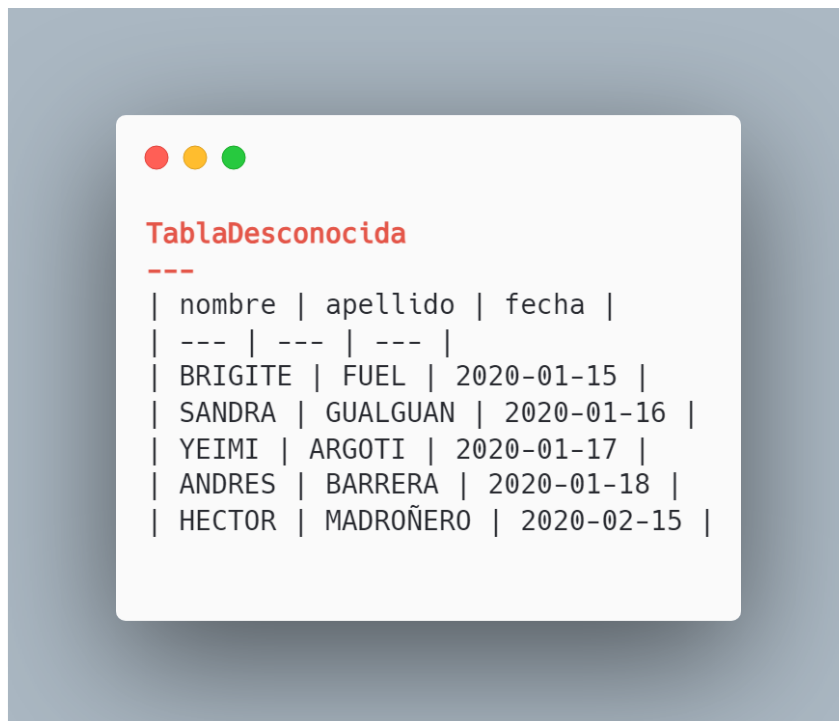


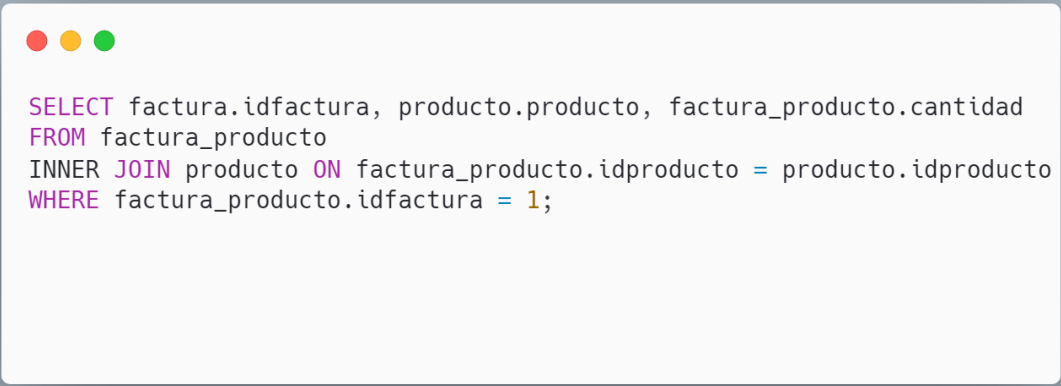
Figura 3.

Resultado de la consulta



Consulta de productos vendidos en una factura específica

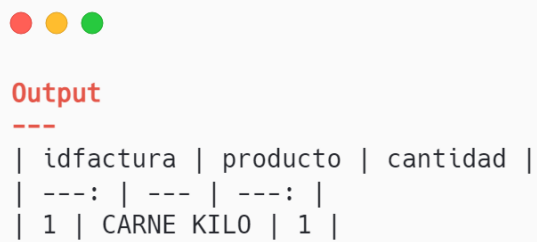
Figura 4.

A screenshot of a SQL query window with a light gray background and three colored window control buttons (red, yellow, green) in the top left corner. The query is written in a monospaced font with syntax highlighting: keywords are in purple, table names in pink, and values in yellow.

```
SELECT factura.idfactura, producto.producto, factura_producto.cantidad
FROM factura_producto
INNER JOIN producto ON factura_producto.idproducto = producto.idproducto
WHERE factura_producto.idfactura = 1;
```

Figura 5.

Resultados de la consulta

A screenshot of a query output window with a light gray background and three colored window control buttons (red, yellow, green) in the top left corner. The word "Output" is in red, followed by a red dashed line. The results are displayed in a table with three columns: idfactura, producto, and cantidad.

```
Output
---
```

idfactura	producto	cantidad
1	CARNE KILO	1

Subconsultas

Figura 6.

```
select round(AVG(PRO.costo),0), TP.tipo_producto, "Mayor costo promedio" from producto AS PRO
INNER JOIN tipo_producto AS TP ON PRO.idtipo_producto = TP.idtipo_producto
GROUP by TP.tipo_producto
having round(AVG(PRO.costo),0) = (
SELECT MAX(SQ1.promedio) FROM (
select round(AVG(PRO.costo),0) AS promedio from producto AS PRO
INNER JOIN tipo_producto AS TP ON PRO.idtipo_producto = TP.idtipo_producto
GROUP by TP.tipo_producto
) AS SQ1
)
union
select round(AVG(PRO.costo),0), TP.tipo_producto, "menor costo promedio" from producto AS PRO
INNER JOIN tipo_producto AS TP ON PRO.idtipo_producto = TP.idtipo_producto
GROUP by TP.tipo_producto
having round(AVG(PRO.costo),0) = (
SELECT MIN(SQ1.promedio) FROM (
select round(AVG(PRO.costo),0) AS promedio from producto AS PRO
INNER JOIN tipo_producto AS TP ON PRO.idtipo_producto = TP.idtipo_producto
GROUP by TP.tipo_producto
) AS SQ1
)
)
```

Figura 7.

Resultados de la consulta

```
producto
---
| round(AVG(PRO.costo),0) | tipo_producto | Mayor costo promedio |
| ---: | --- | --- |
| 12000 | carnico | Mayor costo promedio |
| 2425 | mecato | menor costo promedio |
```

Explicación de Consultas

- I. En la primera consulta, se realiza un INNER JOIN entre la tabla cliente y factura utilizando la clave foránea idcliente, lo que permite listar los clientes con sus respectivas facturas
- II. En la **segunda consulta**, se une factura_producto con producto mediante idproducto para mostrar los productos vendidos en una factura específica
- III. Este query busca obtener el tipo de producto con el mayor y el menor costo promedio. Usa UNION para combinar dos subconsultas: una que calcula el mayor costo promedio de productos agrupados por tipo y otra que calcula el menor costo promedio, comparando los promedios en cada caso para mostrar el resultado

Análisis y Discusión de Resultados

La primera consulta nos permitió obtener una lista de clientes junto con las fechas de sus respectivas facturas. El objetivo fue demostrar la relación entre la tabla de clientes y facturas, validando que un cliente puede tener múltiples facturas. Esta consulta es útil para entender la frecuencia de compras de cada cliente y el comportamiento de los mismos respecto a sus interacciones con el sistema de ventas.

En la segunda consulta, nos enfocamos en recuperar los productos vendidos en una factura específica, uniendo la tabla de productos con la de factura_producto. Los resultados muestran los detalles de las ventas asociadas a la factura 1, incluyendo el nombre del producto y la cantidad vendida.

Este tipo de consulta es esencial para la gestión de inventarios y para hacer un seguimiento preciso de qué productos han sido comprados por los clientes.

En la tercera consulta nos enfocamos para encontrar el tipo de producto con el mayor y menor costo promedio. Este análisis es útil para conocer qué categorías de productos tienen un precio promedio más alto o bajo, lo cual puede influir en decisiones comerciales y de marketing

esta consulta es particularmente interesante ya que permite identificar qué tipo de productos tienen un mayor impacto en términos de costos dentro del inventario. Esto puede influir en decisiones estratégicas como promociones, descuentos o marketing dirigido a productos de mayor o menor precio. La lógica detrás de la consulta utiliza subconsultas para calcular el promedio de cada tipo de producto y luego compara estos promedios con los valores máximos y mínimos

Conclusiones

Los resultados de las consultas SQL ejecutadas proporcionan una visión detallada de cómo se interrelacionan los datos en la base de datos "Tienda". Las relaciones entre clientes, facturas y productos han sido validadas correctamente, y el análisis de costos promedio de productos por tipo destaca información valiosa para la toma de decisiones comerciales.

Este análisis ha demostrado la eficiencia y efectividad de la estructura de la base de datos, así como la importancia de contar con consultas optimizadas para obtener información valiosa de los datos almacenados