INFORME REPASO SQL

Estudiante:

YEINER ADRIAN ANGULO ORDOÑEZ

Docente:

BRAYAN ARCOS

INSTITUTO TECNOLOGICO DEL PUTUMAYO

MOCOA-PUTUMAYO

2024

Contenido

[RESUMEN EJECUTIVO 3](#_Toc176337066)

[INTRODUCCION 3](#_Toc176337067)

[METODOLOGIA 4](#_Toc176337068)

[HERRAMIENTAS USADAS: 4](#_Toc176337069)

[PROCEDIMIENTOS: 4](#_Toc176337070)

[LENGUAJE DE MANIPULACION DE DATOS (DML) 4](#_Toc176337071)

[MySQL 4](#_Toc176337072)

[LENGUAJE DE DEFINICION DE DATOS (DDL) 6](#_Toc176337073)

[MYySQL 6](#_Toc176337074)

[TIPOS DE DATOS 8](#_Toc176337075)

[FUNCIONES Y OPERACIONES 9](#_Toc176337076)

[MySQL FUNCIONES 9](#_Toc176337077)

[MySQL OPERADORES 13](#_Toc176337078)

[CONSULTAS AVANZADAS 15](#_Toc176337079)

[MySQL 15](#_Toc176337080)

[DISEÑO DE BASES DE DATOS 21](#_Toc176337081)

[MODELO DE DATOS: 21](#_Toc176337082)

[CONSIDERACION DE DISEÑO 21](#_Toc176337083)

[ELECCION DE CLAVES PRIMARIAS: 21](#_Toc176337084)

[FOREIGN KEY (LLAVES FOREANEAS) 21](#_Toc176337085)

[ANALISIS Y DISCUSIÓN: 22](#_Toc176337086)

[CONCLUCION 23](#_Toc176337087)

[RECOMENDACIONES 23](#_Toc176337088)

[REFERENCIAS 23](#_Toc176337089)

# RESUMEN EJECUTIVO

El siguiente informe que se presentara es un análisis detallado sobre los conceptos fundamentales de las bases de datos relacionales en MySQL. El informe abordara aspectos como es el lenguaje de manipulación de datos (DML), el lenguaje de definición de datos (DDL), tipos de datos, funciones y operadores SQL, así como las consultas avanzadas en SQL en un escenario de desarrollo de software

# INTRODUCCION

En el siguiente informe se realizará para explorar a fondo MySQL, una de las bases de datos relacionales más utilizadas en el mundo. Es esencial que se comprenda las capacidades y limitaciones que tiene para su implementación en algún proyecto de desarrollo de software.

Se van a cubrir aspectos fundamentales como la creación y manipulación de datos entra vez de las consultas SQL, también en el diseño de bases de datos relacionales, la optimización de consultas, y las técnicas avanzadas como joins y subconsultas.

Para llegar al objetivo principal que es proporcionar un marco sobre el entendimiento y la utilización de MySQL de la manera más eficiente, mejorando el rendimiento y la organización de los datos en proyectos reales.

# METODOLOGIA

## HERRAMIENTAS USADAS:

La herramienta usada va a ser MySQL Workbench se utilizó como herramienta de creación, manipulación y para el análisis de bases de datos en MySQL.

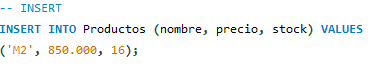
## PROCEDIMIENTOS:

Se llevo a cabo diversas consultas SQL para hacer la demostración de la manipulación de datos, el diseño de la base de datos y la optimización de rendimiento haciendo uso de un modelo detallado.

# LENGUAJE DE MANIPULACION DE DATOS (DML)

## MySQL

\*INSERT: Es utilizado para agregar filas en una tabla en este caso se usó la tabla llamada productos que contaba con 10 productos ya creados, la tabla pertenece a la base de datos denominada “Almacen\_adrian” para realizar la consulta insert que se le agregara o insertar un nuevo producto a la tabla productos.



Resultado de la consulta:

Al haber ejecutado la consulta nos arrojó que se había agregado un nuevo producto a la tabla productos que sería “un M2 de valor 850.000 y con un stock de 16) evidencia a continuación:



\*UPDATE: Es utilizado para actualizar filas en una tabla en este caso se usó la tabla llamada productos que contaba con 10 productos ya creados, la tabla pertenece a la base de datos denominada “Almacen\_adrian” para realizar la consulta update se optó por buscar el producto por su nombre para hacer la respectiva actualización o modificación del producto en la tabla productos.



Resultado de la consulta:

Al haber ejecutado la consulta nos arrojó que se había actualizado el valor de un producto de la tabla productos que sería “un M2 de valor 850.000 y con un stock de 16” a “un M2 de valor 750.000 y con un stock de 16” evidencia a continuación:

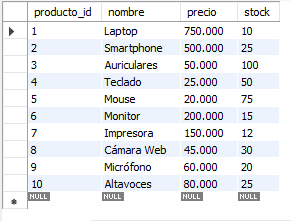


\*DELETE: Es utilizado para eliminar filas en una tabla en este caso se usó la tabla llamada productos que contaba con 10 productos ya creados, la tabla pertenece a la base de datos denominada “Almacen\_adrian” para realizar la consulta delete se optó por buscar el producto por el nombre para hacer la respectiva eliminación del producto de la tabla productos.



Resultado de la consulta:

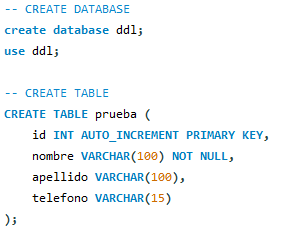
Al haber ejecutado la consulta nos arrojó que se había eliminado un producto de la tabla productos que sería un "M2 de valor 750.000 y con un stock de 16” evidencia a continuación:



# LENGUAJE DE DEFINICION DE DATOS (DDL)

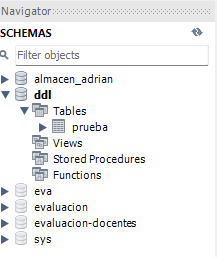
## MYySQL

\*CREATE TABLE: se utiliza para crear tablas en una base de datos, primero se tiene que crear una base de datos para poder realizar la consulta “créate table” por consiguiente se creó una base de datos llamada DDL para crear una tabla llamada prueba.



Resultado de la consulta:

Al haber ejecutado la consulta nos arrojó que se había creado una tabla llamada prueba en la base de datos ddl la evidencia esta a continuación:

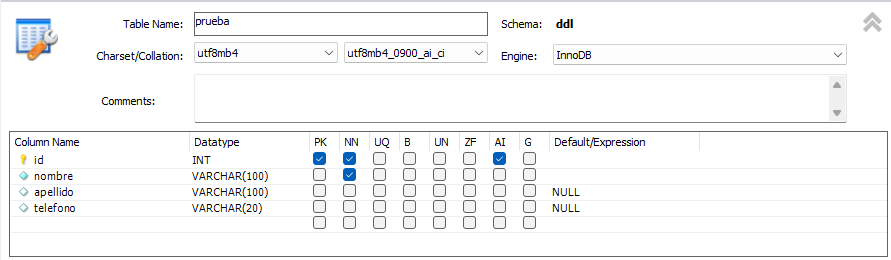


\*ALTER TABLE: se utiliza para modificar los datos de una tabla por consiguiente se usó la tabla prueba para hacer una modificación en los datos de la tabla que se encuentra en una base de datos llamada DDL.



Resultado de la consulta:

Al haber ejecutado la consulta nos arrojó que se había modificado los datos de la columna teléfono en su tamaño de almacenamiento de “15 a “20” la evidencia a continuación:

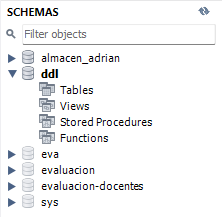


\*DROP TABLE: se utiliza para eliminar tablas de una base de datos para realizar el ejemplo vamos a usar la tabla prueba para eliminarla de la base de datos que tiene como nombre DDL



Resultado de la consulta:

Al haber ejecutado la consulta nos arrojó que se había eliminado la tabla prueba de la base de datos llamada DDL la evidencia a continuación:



# TIPOS DE DATOS

|  |  |  |  |
| --- | --- | --- | --- |
| Categoría | Tipo de dato | Descripción | Rango |
| numéricos | Int | Entero con signo de 4 bits | -2,147,483,648 a 2,147,483,647. |
|  | Tinyint | Entero pequeño de 1 byte. | -128 a 127. |
|  | Smallint | Entero pequeño de 2 bytes. | -32,768 a 32,767 |
|  | Mediumint | Entero medio de 3 bytes | -8,388,608 a 8,388,607 |
|  | Bigint | Entero grande de 8 bytes | -9,223,372,036,854,775,808 a 9,223,372,036,854,775,807. |
|  | Float | |  | | --- | |  |  |  | | --- | | Número de coma flotante de precisión simple. | | |
|  | Double | Número de coma flotante de doble precisión. | |
|  | Decimal | Número decimal con precisión fija. Utilizado para valores financieros. | |
| Cadena | Char | Cadena de longitud fija | 1 a 255 caracteres |
|  | Varchar | Cadena de longitud variable | 1 a 65,535 caracteres |
|  | Text | Cadena de texto de longitud variable | hasta 65,535 caracteres |
|  | Mediumtext | Cadena de texto de longitud variable | hasta 16,777,215 caracteres |
|  | Longtext | Cadena de texto de longitud variable | hasta 4,294,967,295 caracteres |
| Fecha y hora | Date | Fecha en formato 'YYYY-MM-DD'. | |
|  | Datetime | Fecha y hora en formato 'YYYY-MM-DD HH:MM'. | |
|  | Timestamp | Marca de tiempo con fecha y hora, desde '1970-01-01 00:00:01' UTC hasta '2038-01-19 03:14:07' UTC. | |
|  | Time | Hora en formato 'HH:MM'. | |
|  | Year | |  | | --- | |  |  |  | | --- | | Año en formato 'YYYY'. | | |
| Binario | Binary | Datos binarios de longitud fija. | |
|  | Varbinary | Datos binarios de longitud variable. | |
|  | Blob | Objeto binario grande | 65,535 bytes de datos |
|  | Mediumblob | Objeto binario grande | hasta 16,777,215 bytes de datos |
|  | Longblob | Objeto binario grande | hasta 4,294,967,295 bytes de datos |
| Otros | Enum | |  | | --- | |  |  |  | | --- | | Enumeración, lista de valores predefinidos. | | |
|  | Set | |  | | --- | |  |  |  | | --- | | Conjunto de valores predefinidos, permite múltiples valores. | | |
|  | Json | Almacena datos en formato JSON. | |

# FUNCIONES Y OPERACIONES

## MySQL FUNCIONES

\*COUNT: la siguiente función cuenta el número de filas en la tabla en el siguiente caso va a contar a los clientes de la tabla “clientes” de la base de datos “Almacen\_adrian”



Resultado consulta: La función devuelve el número total de clientes registrados en la base de datos.



\*SUM: la siguiente función calcula el total en el siguiente caso va a Multiplica el precio de cada producto por la cantidad vendida y suma todos esos valores para obtener el total de ingresos generados por las ventas.



Resultado consulta: La función devuelve el total de los ingresos que se generaron en ventas.



\*AVG: La siguiente función calcula el promedio de un conjunto de valores en el siguiente caso se utiliza para calcular el precio promedio de todos los productos en la tabla Productos.



Resultado consulta: La función devuelve el promedio de los productos de la tabla productos.



\*MAX: La siguiente función devuelve el valor máximo en un conjunto de datos en este caso se usará para buscar el producto mas caro de la tabla “productos”.



Resultado consulta: devuelve el valor máximo de la tabla productos.



\*MIN: La siguiente función devuelve el valor mínimo en un conjunto de datos en este caso se va a buscar el producto con la menor capacidad en el stock.



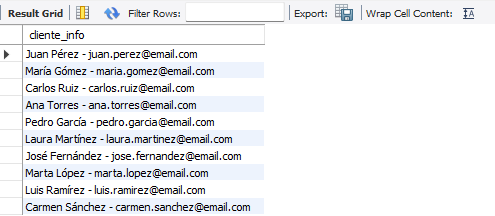
Resultado consulta: devuelve el producto con menor capacidad en el stock.



\*CONCAT: La siguiente función combina varias cadenas en una sola en este caso vamos a combinar el nombre y el correo de cada cliente y van a ser separados por un “-“ en una sola columna que va a llevar por nombre “cliente\_info”



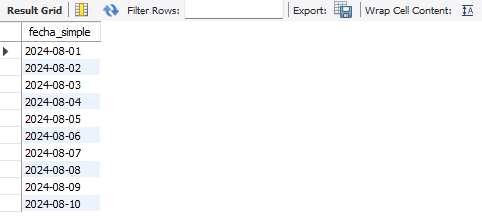
Resultado consulta: nos devuelve una combinación de nombre y correo de los clientes y separados por un “-“.



\*DATE: La siguiente función extrae solo la parte de la fecha de una columna de tipo fecha y hora (timestamp) en este caso vamos a convertir la fecha en una columna.



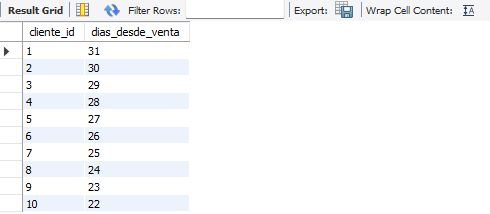
Resultado consulta: nos devuelve la fecha de ventas.



\*DATEDIFF: La siguiente función calcula la diferencia en días entre dos fechas en este caso vamos a calcular la cantidad de días que han pasado desde cada venta hasta la fecha actual.



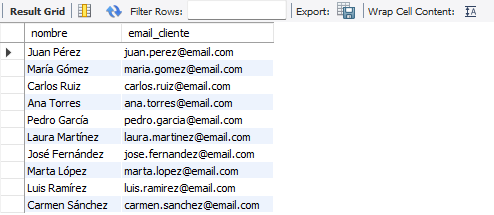
Resultado consulta: nos devuelve los días que han pasado desde cada venta hasta hoy.



\*IFNULL: La siguiente función devuelve el primer argumento si no es NULL; de lo contrario, devuelve el segundo argumento en este caso un cliente no tiene email registrado.



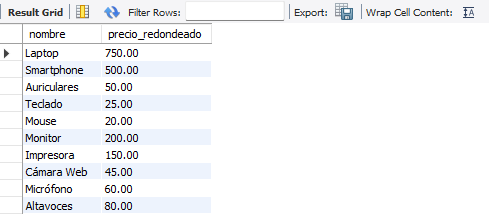
Resultado consulta: nos va a devolver todos los clientes y si alguno no tiene email nos va a mostrar no registrado, pero no hay ninguno todos están registrados.



\*ROUND: La siguiente función redondea un número a un número específico de decimales en este caso se va redondear el precio de cada producto a dos decimales.



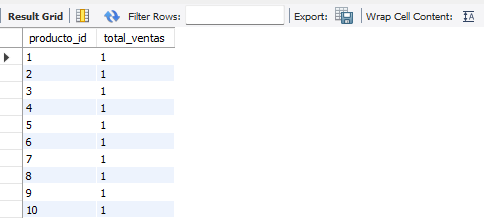
Resultado consulta: devuelve el precio redondeado a dos decimales de cada producto.



\*GROUP BY y COUNT: La siguiente funcion agrupa las filas que tienen los mismos valores en las columnas especificadas y luego realiza cálculos agregados sobre cada grupo en este caso se agruparan las ventas por “producto\_id” y se va a contar cuantas veces se ha vendido cada producto.



Resultado consulta: devuelve la agrupación y la venta de cada producto.

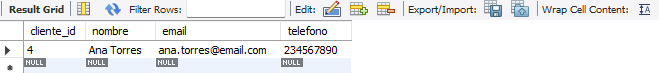


## MySQL OPERADORES

\*LIKE: El siguiente operador se utiliza para buscar un patrón específico en una columna en este caso se buscan todos los clientes cuyos nombres comienzan con la letra 'A'.



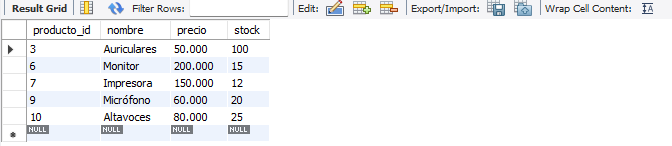
Resultado consulta: El operador devuelve los nombres de los clientes que su nombre empieza por la letra A.



\*BETWEEN: El siguiente operador se utiliza para seleccionar valores dentro de un rango dado en este caso se va a hacer una consulta que me devuelva los productos que el precio este en el rango de 50 a 200.



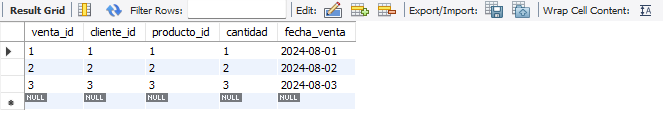
Resultado consulta: El operador devuelve los productos que están en el rango.



\*IN: El siguiente operador se utiliza para comprobar si un valor existe dentro de un conjunto de valores en el siguiente caso se va a seleccionar las ventas que el “producto id” es 1,2 o 3.



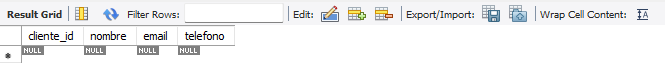
Resultado consulta: El operador devuelve las ventas donde “producto id” es 1,2 o 3.



\*IS NULL: El siguiente operador se utiliza para verificar si un valor es null en este caso vamos a seleccionar a los clientes que no tienen correo electrónico registrado.



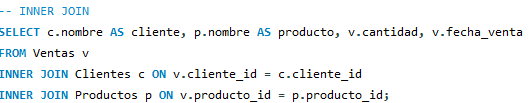
Resultado consulta: El operador devolverá los clientes que no tiene correo registrado.



# CONSULTAS AVANZADAS

## MySQL

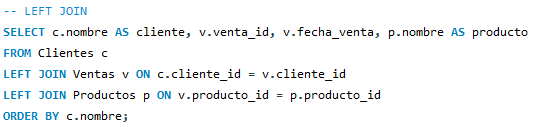
\*INNER JOIN: Es utilizada para combinar tablas en este caso se va a obtener los nombres de los clientes y productos, junto con la cantidad y la fecha de la venta.



Resultado consulta: se obtiene las filas que tienen coincidencias en ambas tablas.



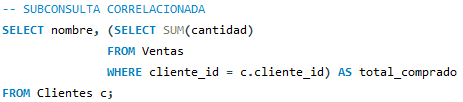
\*LEFT JOIN: se utiliza para devolver todos los registros en este caso se va devolver todos los registros de la tabla Clientes y los registros coincidentes en Ventas y Productos.



Resultado consulta: no aparece ningún resultado como null.



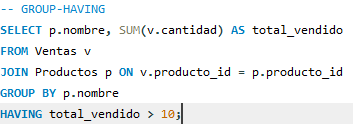
\*SUBCONSULTA CORRELACIONADA: es una consulta que esta relacionada en este caso los clientes para cada cliente en la tabla cliente.



Resultado consulta: el total de productos que han comprado sumando la cantidad de ventas correspondientes en la tabla Ventas.



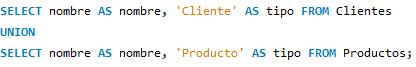
\*GROUP BY y HAVING: La consulta agrupa y de HAVING filtra los resultados en este caso va a agrupar las ventas por producto y calcula la cantidad total vendida para cada producto.



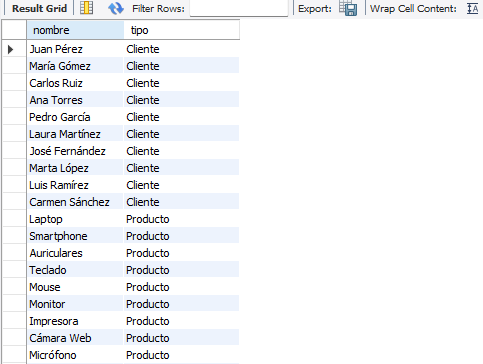
Resultado consulta: No va a mostrar nada porque no hay productos que hayan vendido más de 10 veces.



\*UNION: combina los resultados de dos consultas en un solo conjunto de resultados en este caso nos va a combinar los nombres de los clientes incluyendo los productos.



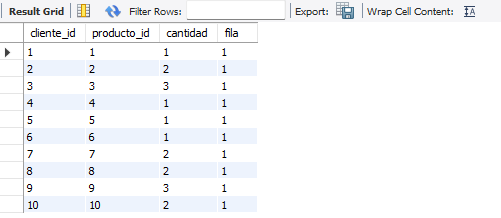
Resultado consulta: Esta consulta devuelve una lista de clientes y productos, con una columna adicional tipo que indica si el nombre corresponde a un cliente o a un producto.



\*ROW\_NUMBER: asigna un número de fila a cada fila dentro de su grupo de PARTITION en el siguiente caso se va a particionar los resultados de clientes y se va a ordenar la fecha de venta en forma descendente.



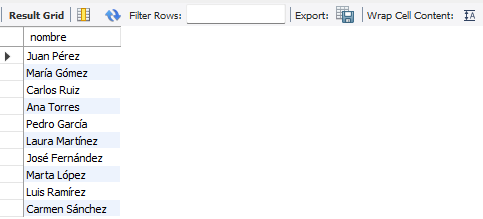
Resultado consulta: se va a enumerar cada compra de cada cliente.



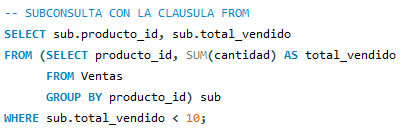
\*EXISTS: se utiliza para verificar si una subconsulta devuelve algún resultado en este caso se va a seleccionar a todos los clientes que han hecho al menos una compra.



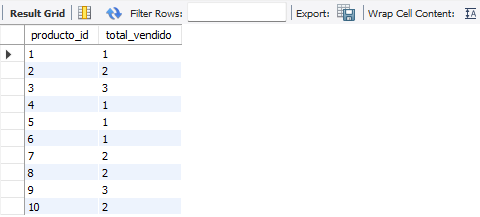
Resultado consulta: devolverá los clientes que han hecho al menos una compra.



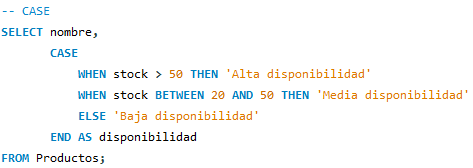
\*SUBCONSULTA EN LA CLÁUSULA FROM: se utiliza una subconsulta dentro de la cláusula FROM. Las consultas van a ser agrupadas por el “producto\_id” y calcula la cantidad total vendida y que han vendido menos de 10 unidades.



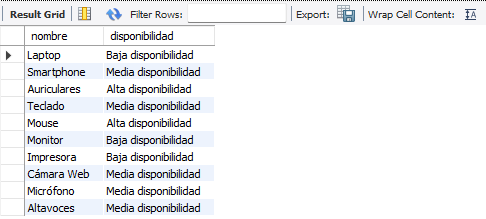
Resultado consulta: devuelve los productos que han vendido menos de 10 unidades.



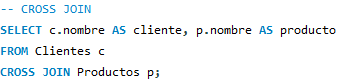
\*CASE: se utiliza para devolver diferentes valores basados en condiciones en este caso se van a clasificar los productos según la cantidad del stock disponible como alta, media y baja.



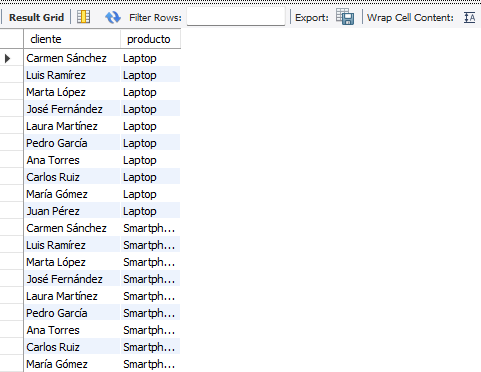
Resultado consulta: devolverá la categorización del stock



\*CROSS JOIN: devuelve el producto cartesiano de las tablas involucradas en este caso combina todas las filas de la tabla Clientes con todas las filas de la tabla Productos



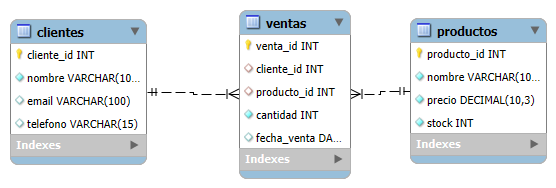
Resultado consulta: genera todas las combinaciones posibles de clientes y productos.



# DISEÑO DE BASES DE DATOS

## MODELO DE DATOS:

Diagrama entidad-relación (ERD):



La tabla ventas tendría una relación de muchos a uno (N:1) por lo que un cliente puede tener muchas compras, pero cada compra esta asociada un solo cliente, también tendría una relación de muchos a uno (N:1) con la tabla productos porque un producto se puede comprar varias veces y cada compra hace referencia a un solo producto.

## CONSIDERACION DE DISEÑO

## ELECCION DE CLAVES PRIMARIAS:

\*Cliente\_id en la tabla Clientes para que cada cliente tenga un identificador único es decir como el numero de la cedula.

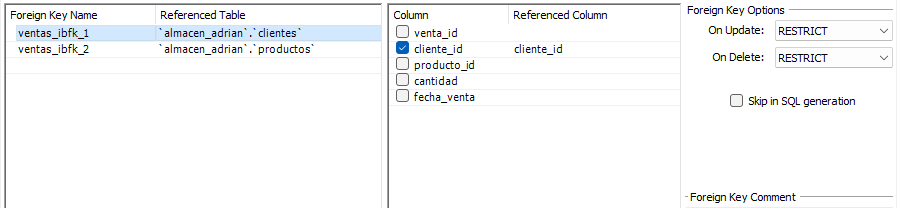
\*Producto\_id en la tabla Productos nos va a asegurar que cada producto sea único.

\*Venta\_id en la tabla Ventas en este caso nos va a garantizar que cada venta sea única.

## FOREIGN KEY (LLAVES FOREANEAS)

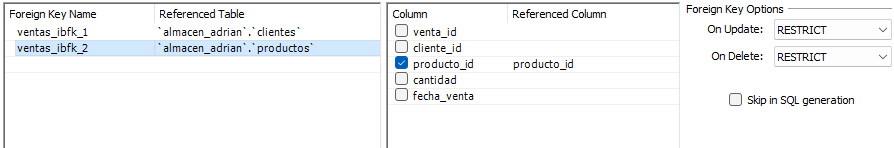
Ventas-Clientes

Descripción: Es una clave foránea que hace referencia en clientes estas relaciones aseguran la referencia para que la tabla ventas se relacione con los registros existentes.



Ventas-Productos:

Descripción: Es una clave foránea que hace referencia en productos estas relaciones aseguran la referencia para que la tabla ventas se relacione con los registros existentes.



## ANALISIS Y DISCUSIÓN:

Los resultados que se obtuvieron con las consultas realizadas en la base de datos nombrada “Almacen\_adrian” nos ha mostrado información valiosa sobre lo que es el comportamiento de las ventas, la relación que esta con los clientes y la gestión con los productos, Al implementar los resultados que no solo cumplen con los objetivos del análisis, sino que también ofrece mejoras solidas en la eficiencia operativa y la toma de dediciones en la gestión del almacén.

# CONCLUCION

MySQL y MongoDB son herramientas poderosas para manejar datos, con un enfoque fundamentalmente diferente. MySQL se basa en una estructura relacional que es estricta con las tablas y con esquemas definidos, por otro lado, MongoDB ofrece flexibilidad con los documentos que pueden variar en su estructura. A la hora de seleccionar entre uno u otro va a depender de los requisitos del proyecto, como la necesidad de transacciones complejas o la capacidad horizontalmente.

# RECOMENDACIONES

Se les ofrece algunas recomendaciones especificas sobre el manejo del diseño y rendimiento de las bases de datos en MySQL, considerando la optimización de consultas y la gestión de transacciones.

# REFERENCIAS

\*Urtiaga, G. G. (2020). *Administrar MySQL y MariaDB: Aprende a administrar MySQL y MariaDB fácilmente*. AprendeIT.

\*Inan, D. I., & Juita, R. (2011). Analysis and design complex and large data base using MySQL workbench. *International Journal of Computer Science & Information Technology*, *3*(5), 173.

\*Letkowski, J. (2015). Doing database design with MySQL. *Journal of Technology Research*, *6*, 1.

\*Yang, L., & Cao, L. (2016). The effect of MySQL Workbench in teaching entity-relationship diagram (ERD) to relational schema mapping. *International journal of modern education and computer science*, *8*(7), 1.

\*bin Uzayr, S. (2022). *Mastering MySQL for Web: A Beginner's Guide*. CRC Press.

Repositorio de alojamiento de archivos:

<https://github.com/YAAO16/mysql_adrian_o.git>