DOCUMENTACION

ANDERSON DANIEL CERON ASTAIZA

PROFESOR: BAYRON ARCOS

INSTITUTO TECNOLOGICO DEL PUTUMAYO

DESARROLLO DE BASE DE DATOS II

MOCOA-PUTUMAYO

11/09/2024

1. Cambio de nombre a la base de datos de “tienda” a “store”

Para hacer este cambio primero cree una nueva base de datos llamada Store

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Después renombre las tablas de tienda por su mismo nombre pero en ingles

Texto

Descripción generada automáticamente

1. Después Para mover las tablas de tienda a store con los nombres en inglés, usa el siguiente comando

Texto

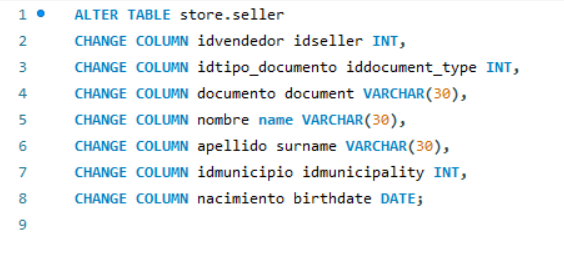
Descripción generada automáticamente

1. Luego comienzo a traducir cada atributo de cada tabla :
2. customer

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Seller:



1. Department

Texto, Carta

Descripción generada automáticamente

1. Country

Texto

Descripción generada automáticamente

1. Document\_type

Texto

Descripción generada automáticamente

1. Invoice

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Invoice\_product

Texto, Carta

Descripción generada automáticamente

1. Municipality

Texto

Descripción generada automáticamente

1. Product

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Product\_type

Texto

Descripción generada automáticamente

1. Luego de haber editado toda nuestra base de datos quedara de esta forma:

Diagrama

Descripción generada automáticamente

1. Mejora a la base de datos con la eliminación de tablas como Seller, customer, supplier y creación de nuevas tablas llamadas person, person\_role y role.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza baja

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

1. Corrección de error id\_nombre\_de\_la \_tabla a solo id

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Consulta INNER JOIN
2. con la tabla person:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. Consulta INNER JOIN con la tabla product:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. Consulta INNER JOIN con la tabla munipality:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Consulta LEFT JOIN
2. Consulta con LEFT JOINpara obtener personas y su rol, (si lo tienen )

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. Consulta con LEFT JOIN para obtener todos los productos y las facturas asociadas (si existen):

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

1. Consulta con LEFT JOIN
2. Consulta con RIGHT JOIN para obtener todas las personas y su rol (incluso si no tienen rol asignado)

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. Consulta con RIGHT JOIN para obtener el tipo de producto, aunque no esté en una factura

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Consulta con CROSS JOIN
2. Consulta con CROSS JOINpara combinar todas las personas con todos los roles

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. Consulta con CROSS JOIN para combinar todos los productos con todas las facturas

Interfaz de usuario gráfica, Texto, Aplicación

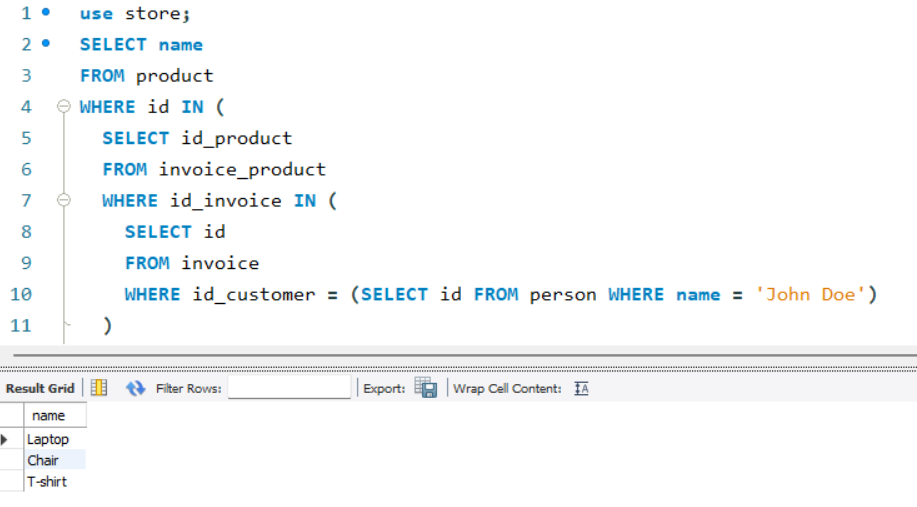
Descripción generada automáticamente

1. Diagrama de la Nueva Base de Datos

Diagrama

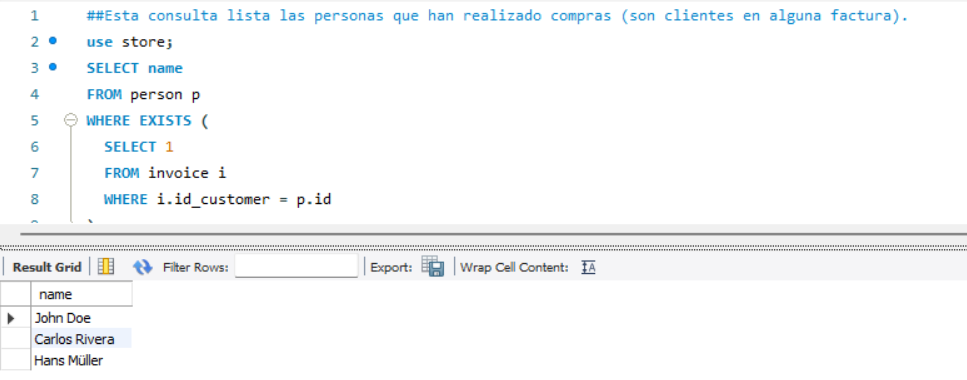
Descripción generada automáticamente

1. Subconsultas:
2. IN:



* La subconsulta interior obtiene las facturas de **John Doe** .
* Luego, la subconsulta intermedia obtiene los productos relacionados con esas facturas.
* Finalmente, la consulta exterior lista los nombres de esos productos.

1. EXISTS:



* La subconsulta EXISTSverifica si una persona ha realizado una compra, es decir, si existe al menos una factura con esa persona como cliente.
* La consulta principal lista los nombres de dichas personas.

1. ANY:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* La subconsulta selecciona los precios de todos los productos en la categoría de **Mobiliario** .
* La consulta principal busca productos cuyo precio sea mayor que el precio de **cualquier** producto en esa categoría.

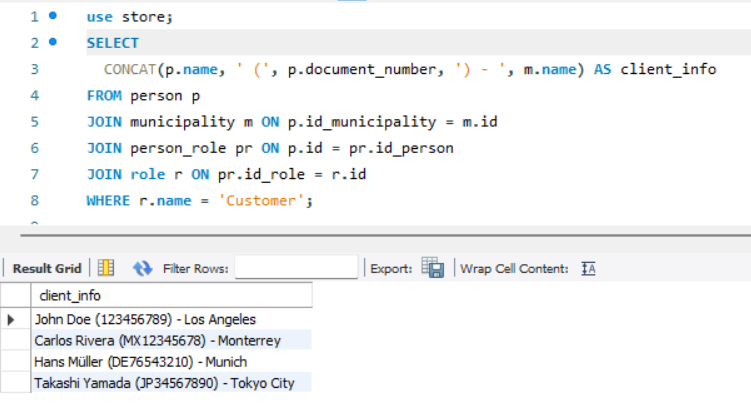
1. ALL:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* La subconsulta selecciona los precios de todos los productos en la categoría de **Mobiliario** .
* La consulta principal busca productos cuyo precio sea mayor que **todos** los precios en esa categoría, es decir, aquellos productos que sean más caros que el más caro de la categoría "Muebles".

1. Manipulación de cadena de textos:
2. CONCAT: Combina cadenas.



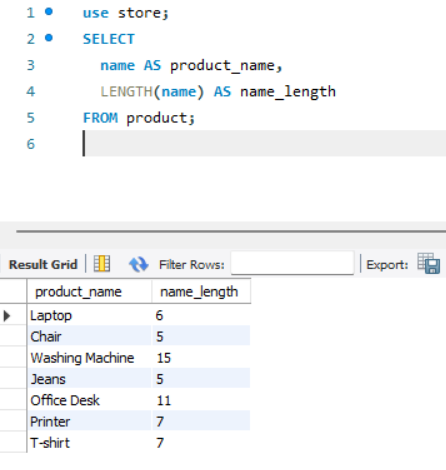
* **CONCAT**: Combina el nombre del cliente, el número de documento y la ciudad en una sola columna llamada client\_info.
* **JOIN**:

Se hace un JOIN entre la tabla person y municipality para obtener la ciudad donde vive cada persona.

También se hace JOIN con person\_role y role para filtrar solo aquellos que tienen el rol de "Customer" (clientes).

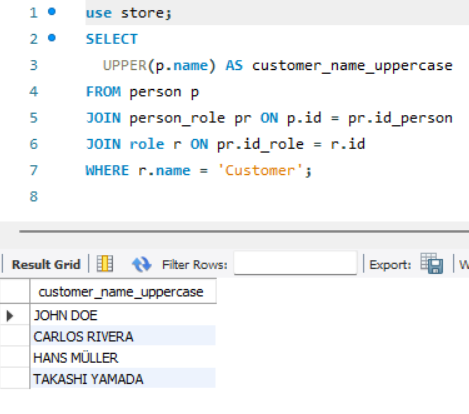
* **WHERE**: Filtra solo las personas que tienen el rol de "Customer" (clientes).

1. LENGHT: Longitud de una cadena.



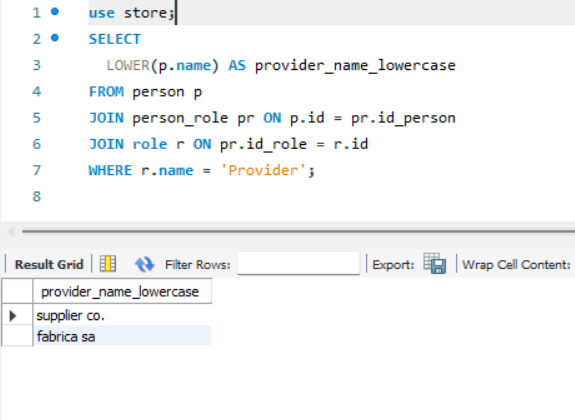
* **LENGTH**: Calcula la longitud del nombre del producto (cantidad de caracteres).
* **SELECT**: Muestra el nombre del producto y el valor calculado por LENGTH como name\_length.
* **FROM product**: La consulta trabaja sobre la tabla product.

1. UPPER: Convierte a mayúsculas.



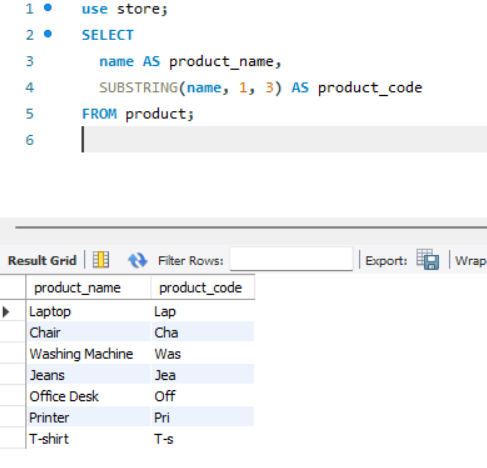
* **UPPER**: Convierte el nombre del cliente a mayúsculas.
* **SELECT**: Muestra el nombre del cliente en mayúsculas con el alias customer\_name\_uppercase.
* **JOIN**: Se hace un JOIN con person\_role y role para obtener solo las personas que tienen el rol de "Customer".
* **WHERE**: Filtra solo las personas con el rol de "Customer" (clientes).

1. LOWER: Convierte a minúsculas.



* **LOWER**: Convierte el nombre del proveedor a minúsculas.
* **SELECT**: Muestra el nombre del proveedor en minúsculas con el alias provider\_name\_lowercase.
* **JOIN**: Se hace un JOIN con person\_role y role para obtener solo las personas que tienen el rol de "Provider".
* **WHERE**: Filtra solo las personas con el rol de "Provider" (proveedores).

1. SUBSTRING: Extrae parte de una cadena.



* **SUBSTRING(name, 1, 3)**: Extrae una subcadena de la columna name, comenzando desde el primer carácter y tomando los tres primeros.
* **SELECT**: Muestra el nombre del producto completo y la subcadena extraída como product\_code.
* **FROM product**: La consulta se realiza sobre la tabla product.

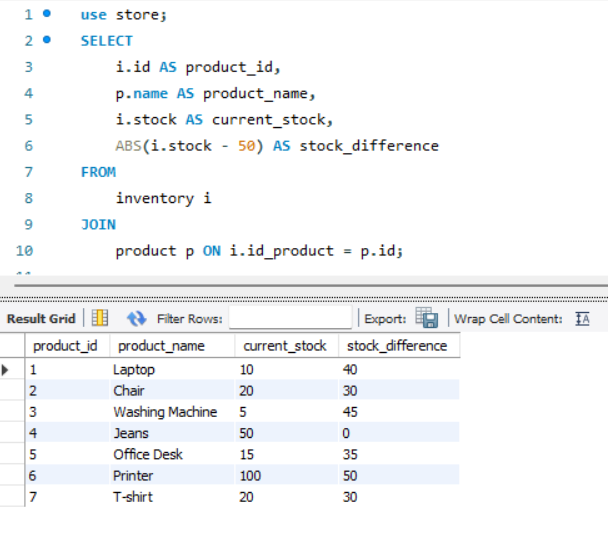
1. FUNCIONES MATEMATICAS EN SQL:
2. ROUND: Redondea un número.

Tabla

Descripción generada automáticamente con confianza media

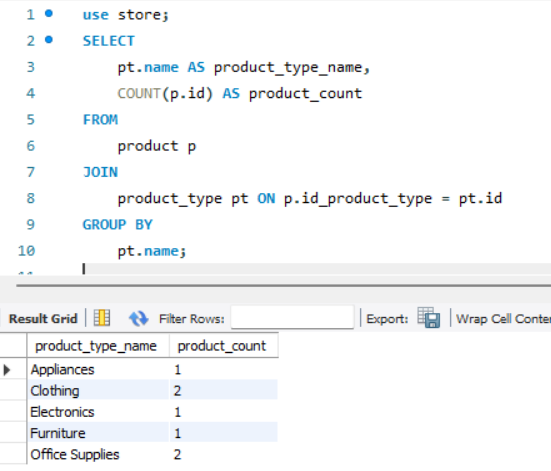
* **ROUND(price, 2)**: Redondea el valor de la columna price a 2 decimales.
* **SELECT**: Muestra el nombre del producto, el precio original, y el precio redondeado como rounded\_price.
* **FROM product**: La consulta trabaja sobre la tabla product.

1. ABS: Valor absoluto: es decir, convierte números negativos en su equivalente positivo.



* La consulta selecciona el id del producto, su nombre (product\_name), el stock actual (current\_stock), y la diferencia absoluta entre el stock y 50 unidades.
* Se utiliza ABS(i.stock - 50) para calcular la diferencia y asegurarse de que sea un valor positivo.

1. FUNCIONES AGREGADAS EN SQL
2. COUNT: Cuenta el número de filas que cumplen una condición



* COUNT(p.id) cuenta cuántos productos están en cada tipo.
* Se usa GROUP BY para agrupar los productos por el tipo de producto (product\_type), y el resultado te dará una lista con el nombre del tipo de producto y la cantidad de productos asociados a ese tipo.

1. SUM: Suma los valores de una columna numérica.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* SUM(total) suma el valor de la columna total de todas las filas en la tabla invoice.
* Esto te devolverá el monto total de todas las ventas/facturas.

1. AVG: Calcula el promedio de los valores de una columna numérica.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* **AVG(price)** calcula el promedio del valor de la columna **price** en la tabla **product**.
* Esto te devolverá el precio promedio de todos los productos que tienes en la tabla **product**.

1. MIN: Encuentra el valor mínimo en una columna

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* **MIN(price)** obtiene el valor más bajo (mínimo) de la columna **price** en la tabla **product**.
* Esto te devolverá el precio más barato de todos los productos.

1. MAX: Encuentra el valor máximo en una columna.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* **MAX(price)** obtiene el valor más alto (máximo) de la columna **price** en la tabla **product**.
* Esto te devolverá el precio más caro de todos los productos.

1. Agrupación de Resultados con GROUP BY

¿ bvInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* **SELECT p.id\_municipality**: Selecciona el **municipio** de las personas. Se tomará esta columna de la tabla **person** para agrupar las facturas por municipio.
* **AVG(i.total)**: Calcula el promedio de los totales de las facturas de la columna **total** en la tabla **invoice**.
* **FROM invoice i**: Indica que los datos de las facturas están en la tabla **invoice**.
* **JOIN person p ON i.id\_customer = p.id**: Conecta la tabla **invoice** con la tabla **person** para obtener el municipio de la persona que realizó la factura, relacionando **id\_customer** en la tabla **invoice** con **id** en la tabla **person**.
* **GROUP BY p.id\_municipality**: Agrupa las facturas por municipio. Esto hace que el promedio del total se calcule por cada municipio diferente.

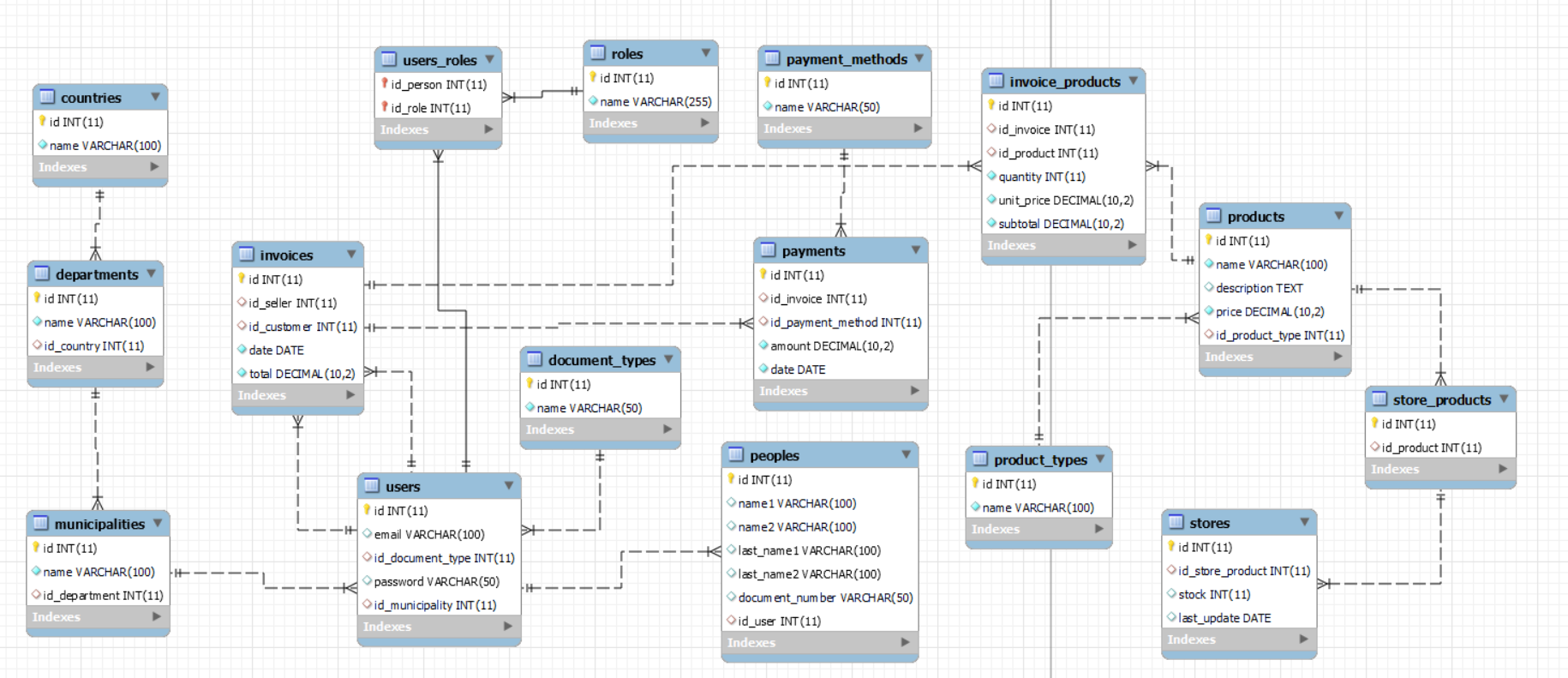
1. Filtrado de Grupos con HAVING

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* SELECT p.name, SUM(i.stock) as total\_stock: selecciona el nombre del producto y la suma total del inventario para cada producto.
* FROM product p JOIN inventory i ON p.id = i.id\_product: une la tabla de productos con la de inventario según el ID de producto.
* GROUP BY p.name: agrupa los resultados por el nombre de cada producto.
* HAVING total\_stock > 50: filtra los productos que tienen más de 50 unidades en stock.

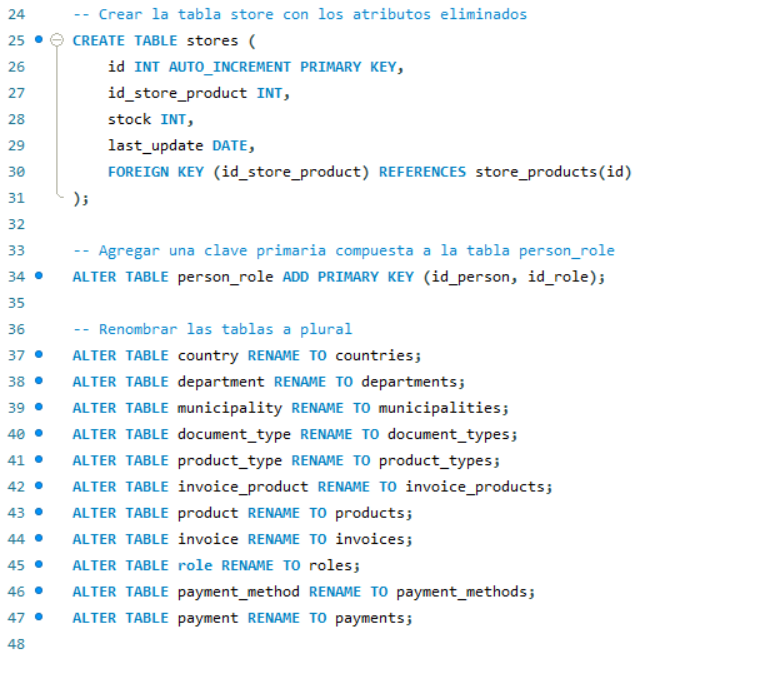
1. Modificación de la base de datos agregándole nuevas tablas como users y users\_roles también store\_products y store:



* 1. Proceso:

Texto

Descripción generada automáticamente



VISTAS, PROCEDIMIENTOS ALMACENADOS Y TRIGGERS

1. Esta vista incluye solo el nombre y apellido de los clientes, junto con su correo electrónico en users.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Resultado de la vista:

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

* **peoples**: Obtenemos los campos name1y last\_name1de esta tabla para el nombre y apellido del cliente.
* **users**: Usamos emailesta tabla para mostrar el correo electrónico del cliente.
* **Relación** : Las tablas se relacionan usando p.id\_user = u.id.

1. **trigger**

se activa al insertar un nuevo registro en la tabla invoices. Este disparador actualizará automáticamente el campo last\_updatede la tabla stores cada vez que se inserte una nueva factura. Esto puede ser útil para llevar un control de cuándo fue la última vez que se realizó una venta en la tienda.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* **Trigger** : update\_last\_update\_on\_new\_invoicese activa después de una inserción ( AFTER INSERT) en la tabla invoices.
* **Tabla Afectada** : stores- se actualizará el campo last\_updatecon la fecha y hora actuales ( NOW()).
* **Condición** : Solo se actualiza la tienda correspondiente al vendedor ( id\_seller) especificado en la nueva factura.

1. **procedimiento almacenado**

inserta un nuevo producto en la tabla products. Este procedimiento recibe como parámetros el nombre, descripción, precio y tipo de producto, e inserta estos valores en la tabla.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* **Parámetros de Entrada (IN)** : Los parámetros product\_name, product\_description, product\_pricey product\_type\_idson datos que se pasan al procedimiento para insertar un nuevo producto.
* **Inserción** : El procedimiento ejecuta una instrucción INSERT INTOen la tabla products, usando los valores proporcionados en los parámetros.
* **DELIMITER** : Cambia el delimitador para definir correctamente el bloque de código.

LINK GITHUB: <https://github.com/AndersonC09/Mysql_Anderson_ceron.git>