Reto BD

Barbería (Ejercicio A)

Entregado A

Juanes Pineda

Presentado

Yeison Ferney Osorio Buitrago

SOFKAU

SAN JOSE DE CUCUTA

Contenido

[Indicar que ejercicio fue asignado 3](#_Toc127530581)

[Realizar el modelo E-R 5](#_Toc127530582)

[Realizar el modelo relacional 7](#_Toc127530583)

[Normalizar correctamente 8](#_Toc127530584)

[Escribir con sentencias SQL toda la definición de la base de datos. 9](#_Toc127530585)

[Escribir consultas que me permitan ver la información de cada tabla o de varias tablas (10). 12](#_Toc127530586)

[Generar de 4 a 6 vistas donde se evidencie lo más importante de cada ejercicio (haga una selección muy responsable de la información realmente importante según el contexto). 16](#_Toc127530587)

[Generar al menos 4 procedimientos almacenados. 18](#_Toc127530588)

[Generar al menos 4 triggers 19](#_Toc127530589)

[Poblar la base de datos (50 registros por tabla) utilizando una conexión desde Java. 20](#_Toc127530590)

[Al terminar el ejercicio responda ¿ Está conforme con el resultado obtenido según el contexto o cree que hubiera obtenido un mejor resultado con una base de datos no relacional? 20](#_Toc127530591)

# Indicar que ejercicio fue asignado

Enunciado



BARBER YEIO es una barbería que ofrece la posibilidad de realizar reservas para sus clientes. Cuando un cliente realiza una reserva, se genera una cita con identificador único en la que se indica la fecha y hora en la que será atendido por un empleado de la barbería. Cada cita está asociada a un servicio en el que se indica el tipo de servicio a realizar y un identificador único.

Cada servicio genera un gasto de insumos, que son los productos necesarios para realizar el servicio, y que están disponibles en la barbería. Los insumos son facturados junto con el servicio prestado. Además, los clientes también tienen la posibilidad de comprar productos disponibles en la barbería, que son distribuidos por diferentes proveedores. Estos productos son vendidos por un empleado de la barbería el cual gana una pequeña comisión por venta realizada que en la factura se refleja el empleado que vendió dicho producto

La barbería cuenta con un registro de clientes en el que se almacenan sus datos personales, como su identificación, nombre, profesión, teléfono y correo. También se guarda un historial de los servicios prestados a cada cliente, para llevar un control detallado de las citas y servicios realizados.

En resumen, BARBER YEIO cuenta con diferentes entidades como reserva, cita, servicio, insumos, factura, productos, proveedor, cliente e historial de servicio, que están relacionadas entre sí de diferentes maneras para llevar un control detallado de los servicios prestados y los productos disponibles en la barbería.

Se desea almacenar la siguiente información:

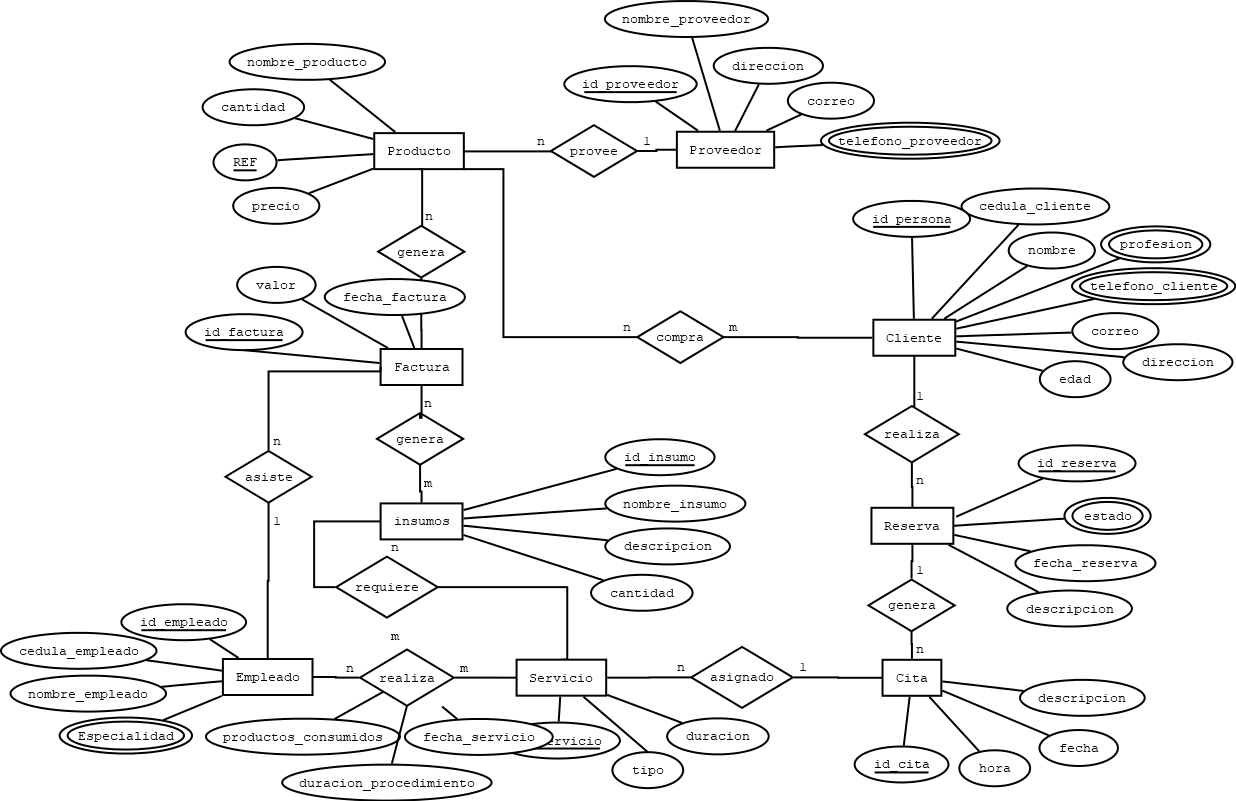
* Empleados: ID, cedula, Nombre, Especialidad (Masaje, Corte, Cejas, etc.)
* Clientes: Datos personales (ID, cedula, Nombre, Profesión, Teléfono, correo, edad y Dirección).
* Historial de Servicios prestados por la barbería: Un registro para saber información del servicio prestado por un empleado a un cliente, productos consumidos, duración del procedimiento y fecha.
* Citas: ID, Fecha y Hora y una descripción en la que se cita al cliente y el barbero que realizará el servicio.
* Productos vendidos por la barbería: REF, Nombre, Cantidad y Precio.
* Proveedor: los productos vendidos deben tener una fuente.
* Registro de Ventas: Si un barbero vende un producto a un cliente, termina obteniendo una “liga” ganancia ocasional.
* Insumo: ID, nombre, descripción, cantidad
* Reserva: ID, estado (confirmada, pendiente, cancelada.), fechas reserva, descripción
* Factura: ID, valor, fecha

Se seba que:

* Los clientes pueden realizar una reserva para programar una cita, si no realiza la reserva no se le atenderá, las políticas de la compañía son estrictas. Además, los clientes tienen la opción de comprar productos, lo cual le genera una factura.
* Un empleado de la barbería realiza un servicio para el cual se le asigna una cita con un cliente en particular. Durante el servicio, se pueden utilizar insumos que generan un costo que se cargará a la factura. El historial del servicio se guarda para llevar registro del empleado que atendió al cliente.
* Si un empleado vende un producto a un cliente, se le otorgará una ganancia o "liga".
* Los productos son provistos por un proveedor externo.

# Realizar el modelo E-R

Modelo Entidad Relación Barberia



**Relación Cliente – Reserva (1-N)**

Cardinalidad: Un cliente puede realizar una o mucha reserva, una reserva puede tener muchas reservas

**Relación Reserva – Cita (1-N)**

Cardinalidad: Una Reserva puede apartar una o muchas citas y una cita pertenece a una reserva

**Relación Cita – Servicio(1-N)**

Cardinalidad: Una cita puede tener uno o muchos servicios y estos servicios están asociados a una o muchas facturas

**Relación Servicio – Insumo(N-M)**

Cardinalidad: un servicio puede tener uno o muchos insumos y un insumo puede ser utilizado por uno o muchos servicios

**Relación Insumo – Factura(N-M)**

Cardinalidad: una factura tiene uno o muchos insumos y un insumo puede estar en uno o muchas facturas

**Relación Empleado -Servicio(N-M)**

Cardinalidad: El Empleado puede realizar uno a muchos servicios y un servicio puede tener muchos empleados por ejemplo un corte de cabello y un corte de uñas

**Relación Empleado – Factura(1-N)**

Cardinalidad: un empleado puede asesorar uno o mucha compra y una compra es asesorada por un empleado

**Relación Factura – Producto(N-M)**

Cardinalidad: una factura tiene uno o muchos productos y un producto puede pertenecer a una o muchas facturas una relación n:m

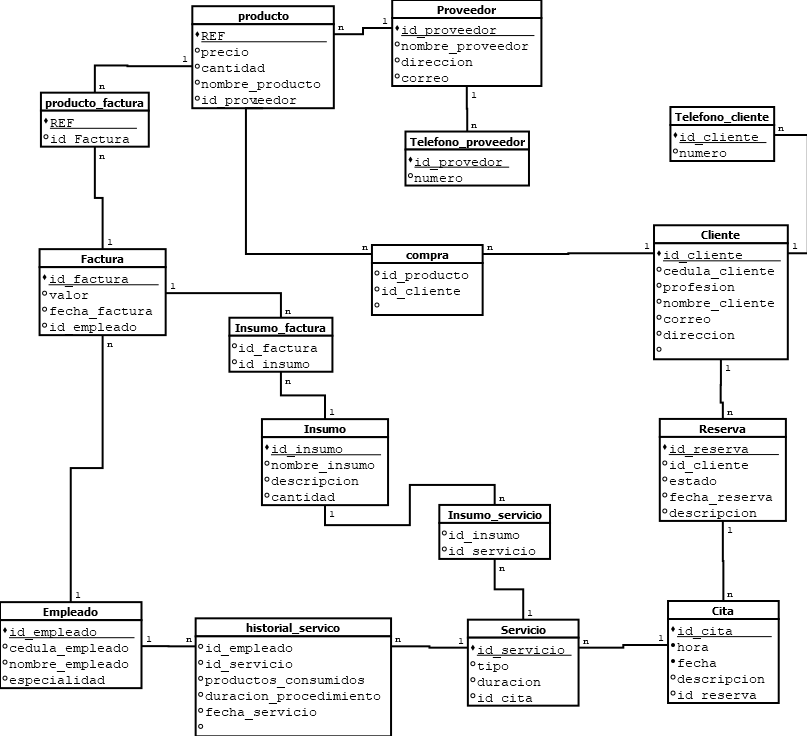
**Relación Cliente – Producto(N-M)**

Cardinalidad: un cliente puede comprar uno o muchos productos y un producto puede ser comprando por uno o muchos clientes

**Relación Proveedor- Producto(N-M)**

Cardinalidad: un proveedor puede proveer uno a muchos productos y un producto es proveído por un proveedor

# Realizar el modelo relacional



# Normalizar correctamente

Normalización

Primera Forma Normal 1FN

* Para cumplir con la función debimos crear tabla para los valores multivaluados y de esta manera cumplir con los requerimientos.
* No deben existir registros duplicados
* Todas las tablas tienen llave primaria

Se cumple con la 2FN.

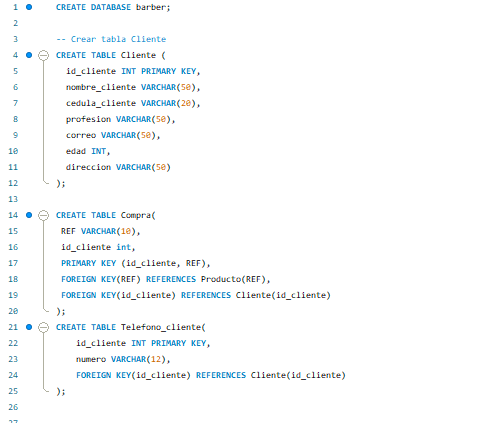
* Cumple con 1FN
* Todos los valores de las columnas dependen únicamente de la llave primaria de la tabla.
* Las tablas tienen una única llave primaria que identifica a la tabla y sus atributos dependen de ella. Esto significa que, si se conoce la llave primaria, se pueden determinar todos los demás valores en la tabla.

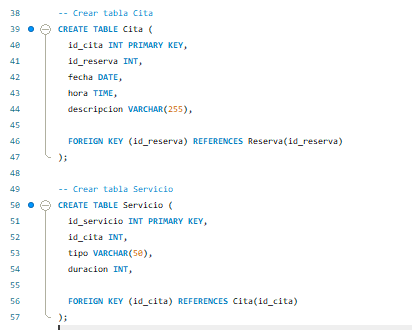
Se cumple con la 3FN.

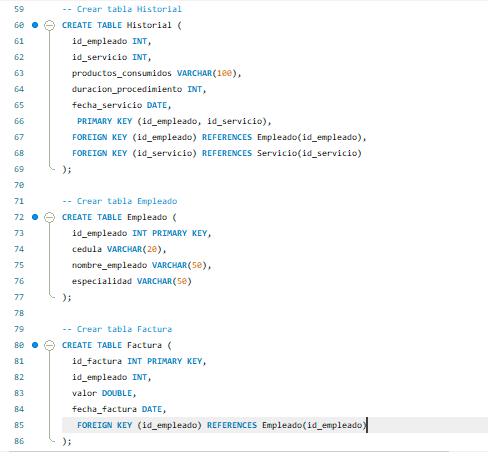
* Estar en 2FN
* Cada atributo que no está incluido en la clave primaria no depende de la clave primaria. Esto significa que un atributo no debe depender de otro atributo que no forme parte de la clave primaria.

# Escribir con sentencias SQL toda la definición de la base de datos.

Pantallazo del script de la base de datos, archivo anexo en el repo

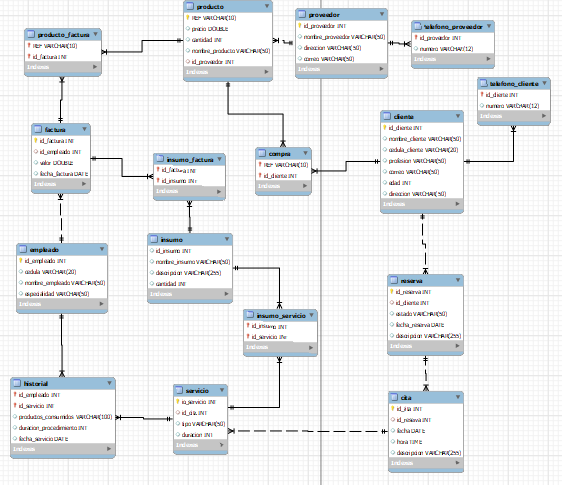








Modelo resultante de la creación de la base de datos



# Escribir consultas que me permitan ver la información de cada tabla o de varias tablas (10).

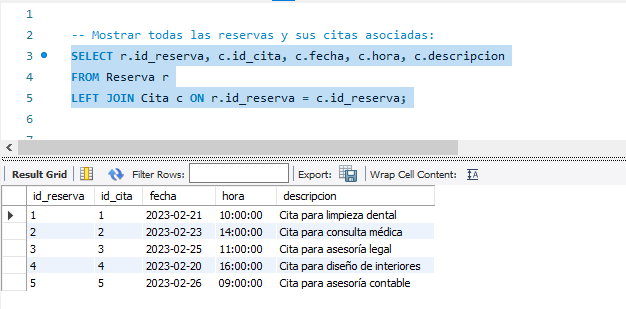
1. La siguiente consulta muestra todas las reservas y sus citas asociadas:

La consulta realiza un JOIN entre las tablas reserva y cita, relacionando las reservas con sus citas asociadas, y selecciona los campos id\_reserva y fecha de la tabla reserva, y los campos fecha, hora y descripción de la tabla cita.

SELECT r.id\_reserva, c.id\_cita, c.fecha, c.hora, c.descripcion

FROM Reserva r

LEFT JOIN Cita c ON r.id\_reserva = c.id\_reserva;



1. La consulta SQL que me permitiría obtener el cliente con sus datos personales, la fecha y hora de su cita, la descripción del servicio y su tipo, y el nombre del empleado que lo atendió:

SELECT cliente.nombre\_cliente, cliente.cedula\_cliente, cliente.correo, cita.fecha, cita.hora,

cita.descripcion, servicio.tipo, empleado.nombre\_empleado

FROM servicio

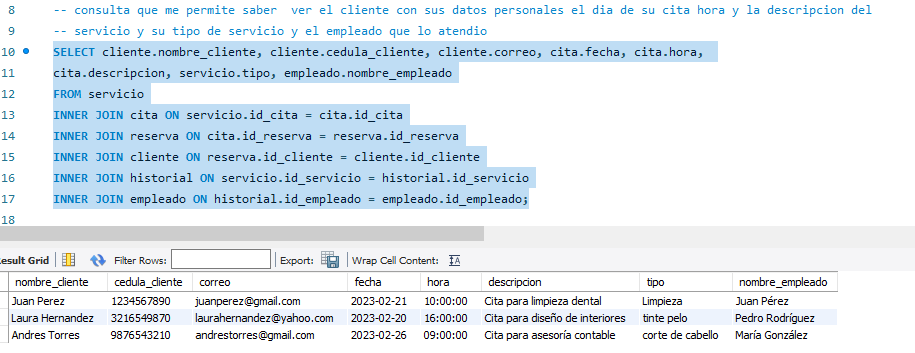
INNER JOIN cita ON servicio.id\_cita = cita.id\_cita

INNER JOIN reserva ON cita.id\_reserva = reserva.id\_reserva

INNER JOIN cliente ON reserva.id\_cliente = cliente.id\_cliente

INNER JOIN historial ON servicio.id\_servicio = historial.id\_servicio

INNER JOIN empleado ON historial.id\_empleado = empleado.id\_empleado;



1. Esta consulta te permitirá obtener información del cliente (nombre, cédula, correo), fecha y hora de la cita, la descripción del servicio y su tipo.

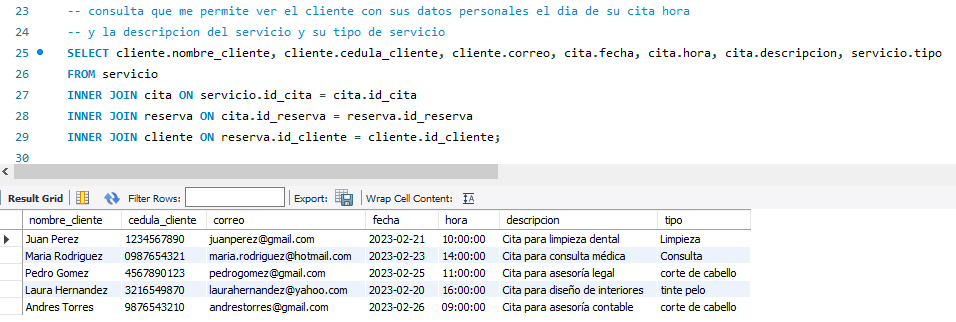
SELECT cliente.nombre\_cliente, cliente.cedula\_cliente, cliente.correo, cita.fecha, cita.hora, cita.descripcion, servicio.tipo

FROM servicio

INNER JOIN cita ON servicio.id\_cita = cita.id\_cita

INNER JOIN reserva ON cita.id\_reserva = reserva.id\_reserva

INNER JOIN cliente ON reserva.id\_cliente = cliente.id\_cliente;



1. consulta que permite conocer la cantidad de productos comprados por cada cliente, incluyendo su nombre:

SELECT COUNT(\*) AS cantidad\_productos, cliente.nombre\_cliente, factura.valor, factura.fecha\_factura, producto.nombre\_producto

FROM producto

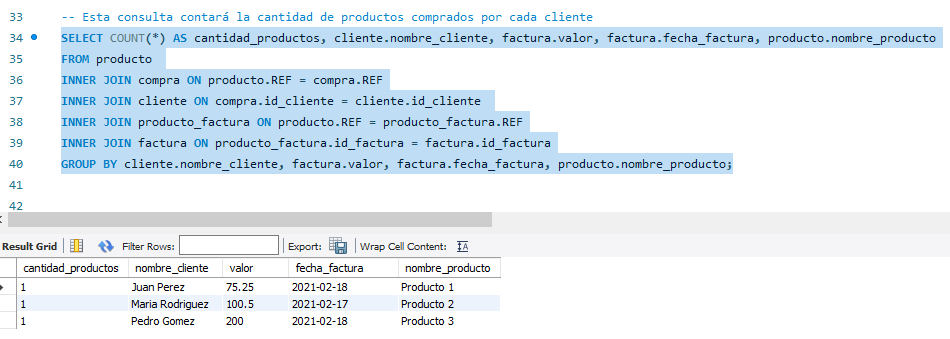
INNER JOIN compra ON producto.REF = compra.REF

INNER JOIN cliente ON compra.id\_cliente = cliente.id\_cliente

INNER JOIN producto\_factura ON producto.REF = producto\_factura.REF

INNER JOIN factura ON producto\_factura.id\_factura = factura.id\_factura

GROUP BY cliente.nombre\_cliente, factura.valor, factura.fecha\_factura, producto.nombre\_producto;



1. Esta consulta permite visualizar los productos comprados por un cliente y el empleado que lo asesoró. Se presenta el nombre del cliente, el nombre del empleado, el nombre del producto, la fecha de la compra y su precio.

SELECT cliente.nombre\_cliente, factura.valor, factura.fecha\_factura, producto.nombre\_producto, empleado.nombre\_empleado

FROM producto

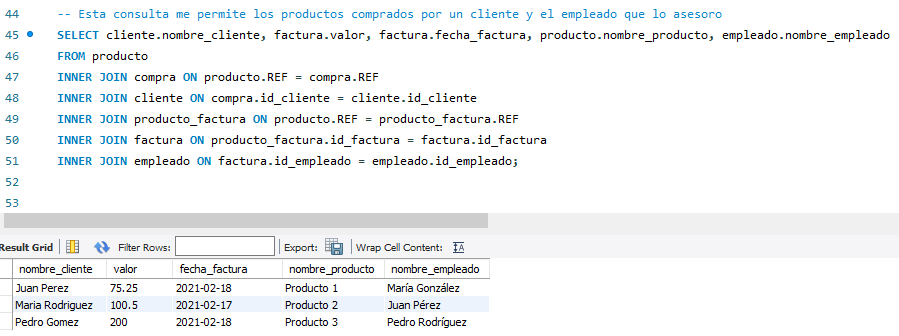
INNER JOIN compra ON producto.REF = compra.REF

INNER JOIN cliente ON compra.id\_cliente = cliente.id\_cliente

INNER JOIN producto\_factura ON producto.REF = producto\_factura.REF

INNER JOIN factura ON producto\_factura.id\_factura = factura.id\_factura

INNER JOIN empleado ON factura.id\_empleado = empleado.id\_empleado;



1. Consulta que me permite obtener información sobre las reservas de los clientes, incluyendo la fecha y hora de las citas, los empleados asignados a los servicios, los insumos utilizados y las facturas correspondientes. En particular, se mostrará la reserva que tiene el cliente, con detalles sobre la fecha y hora de la cita, el empleado asignado para realizar el servicio, los insumos que se utilizaron en dicho servicio, y la factura correspondiente.

SELECT cliente.nombre\_cliente, cliente.cedula\_cliente, cliente.correo, cita.fecha, cita.hora,

cita.descripcion, servicio.tipo, empleado.nombre\_empleado, factura.valor, factura.fecha\_factura

FROM servicio

INNER JOIN cita ON servicio.id\_cita = cita.id\_cita

INNER JOIN reserva ON cita.id\_reserva = reserva.id\_reserva

INNER JOIN cliente ON reserva.id\_cliente = cliente.id\_cliente

INNER JOIN historial ON servicio.id\_servicio = historial.id\_servicio

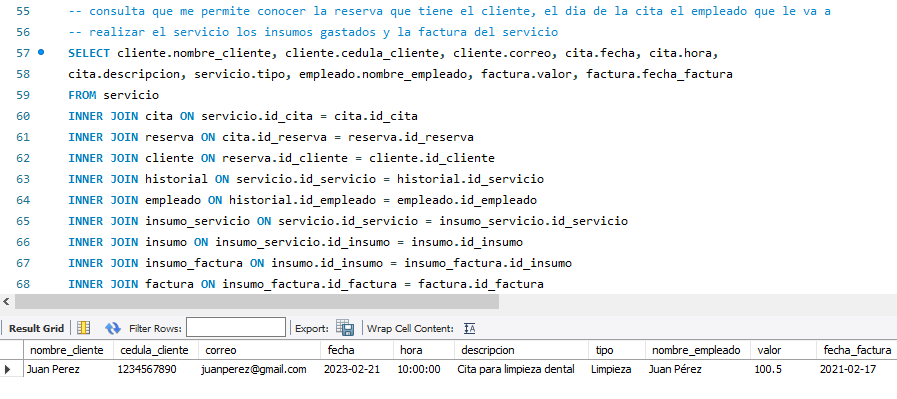
INNER JOIN empleado ON historial.id\_empleado = empleado.id\_empleado

INNER JOIN insumo\_servicio ON servicio.id\_servicio = insumo\_servicio.id\_servicio

INNER JOIN insumo ON insumo\_servicio.id\_insumo = insumo.id\_insumo

INNER JOIN insumo\_factura ON insumo.id\_insumo = insumo\_factura.id\_insumo

INNER JOIN factura ON insumo\_factura.id\_factura = factura.id\_factura;



1. Consulta para saber cuál es el empleado que ha realizado más servicios:

* Selecciona el nombre del empleado y cuenta el número de servicios que ha realizado.
* La tabla "historial" se une con la tabla "empleado" usando el ID del empleado.
* Los resultados se agrupan por el nombre del empleado.
* Los resultados se ordenan de manera descendente por la cantidad de servicios realizados.
* Se limita la consulta a mostrar solo el primer resultado, que sería el empleado con mayor cantidad de servicios realizados.

SELECT empleado.nombre\_empleado, COUNT(\*) as cantidad\_servicios

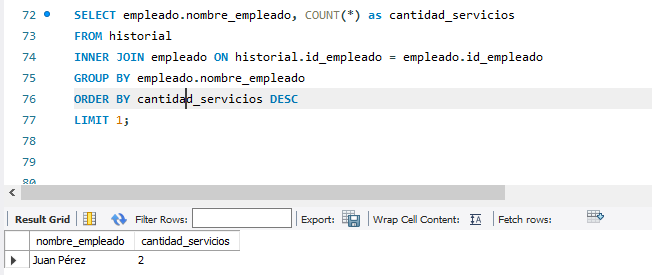
FROM historial

INNER JOIN empleado ON historial.id\_empleado = empleado.id\_empleado

GROUP BY empleado.nombre\_empleado

ORDER BY cantidad\_servicios DESC

LIMIT 1;



1. Cliente que más ha asistido a servicios, podemos contar el número de servicios que ha tenido cada cliente y luego ordenarlos en orden descendente.

SELECT cliente.nombre\_cliente, COUNT(\*) AS num\_servicios

FROM cliente

INNER JOIN reserva ON cliente.id\_cliente = reserva.id\_cliente

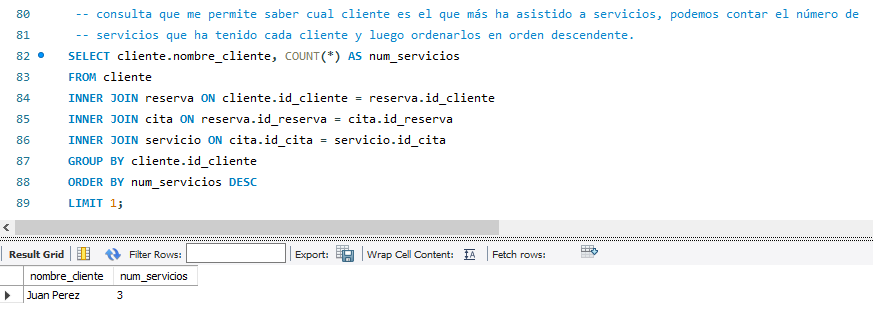
INNER JOIN cita ON reserva.id\_reserva = cita.id\_reserva

INNER JOIN servicio ON cita.id\_cita = servicio.id\_cita

GROUP BY cliente.id\_cliente

ORDER BY num\_servicios DESC

LIMIT 1;



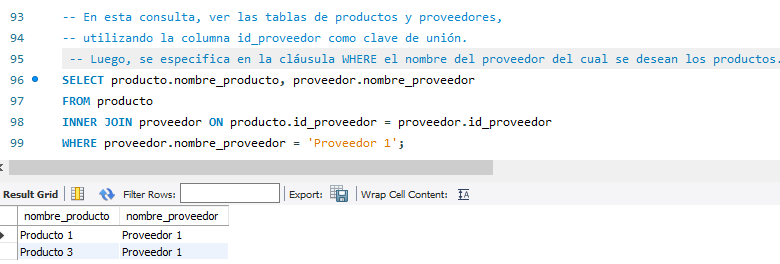
1. En esta consulta, ver las tablas de productos y proveedores, utilizando la columna id\_proveedor como clave de unión. Luego, se especifica en la cláusula WHERE el nombre del proveedor del cual se desean los productos.

SELECT producto.nombre\_producto, proveedor.nombre\_proveedor

FROM producto

INNER JOIN proveedor ON producto.id\_proveedor = proveedor.id\_proveedor

WHERE proveedor.nombre\_proveedor = 'Proveedor 1';

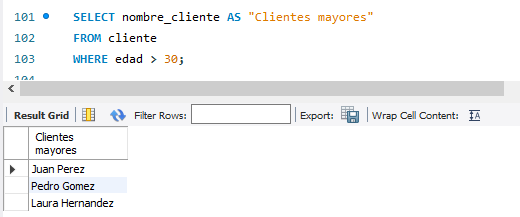


1. Consulta que me permite obtener los nombres de los clientes mayores de 30 años podrías hacer

SELECT nombre\_cliente AS "Clientes mayores"

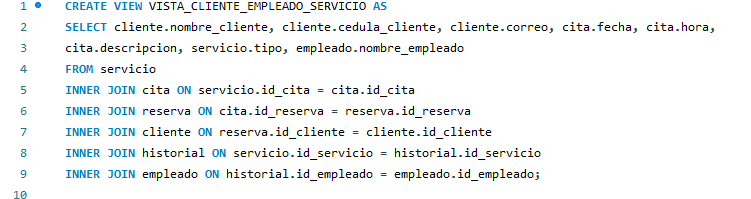
FROM cliente

WHERE edad > 30;

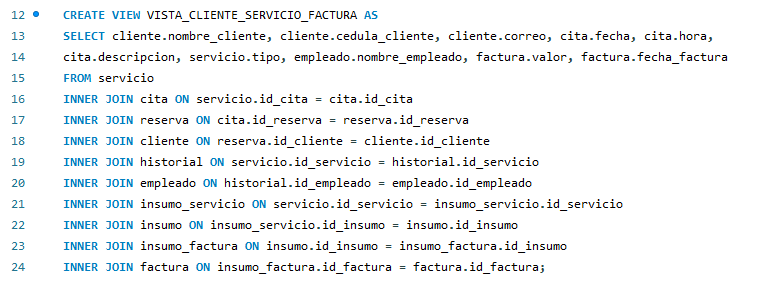


# Generar de 4 a 6 vistas donde se evidencie lo más importante de cada ejercicio (haga una selección muy responsable de la información realmente importante según el contexto).

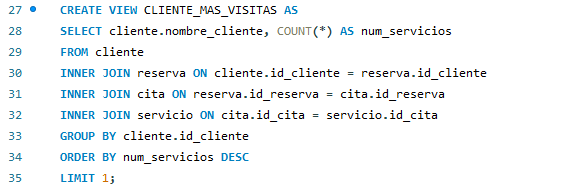
1. Vista que me permite ver el empleado que atendió un cliente



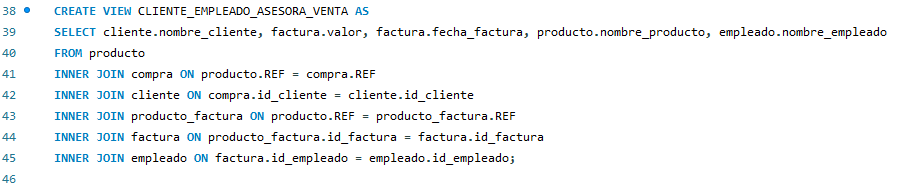
1. Vista que me permite ver todo el ciclo de cita, reserva, servicio, insumo, factura me muestra los datos correspondientes



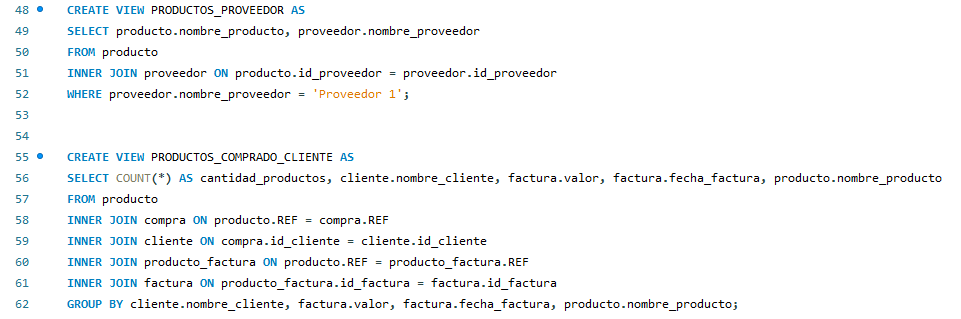
1. Vista que me permite ver el cliente que mas veces me ha visita y cuantas veces



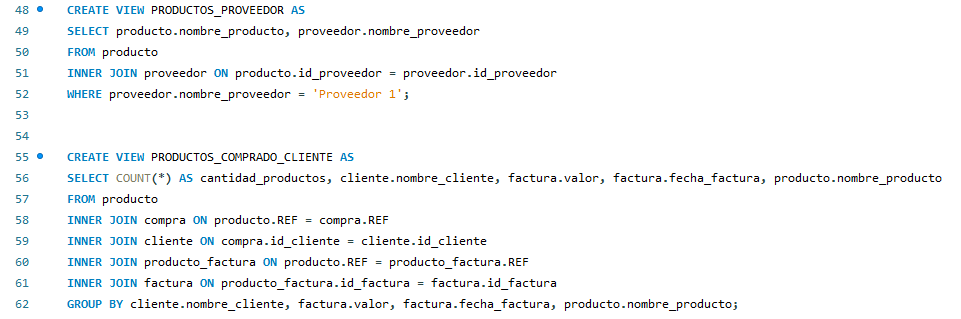
1. Vista que me permite ver que empleado asesoro a un cliente en una compra de un producto



1. Vista que me permite ver los proveedores según mi criterio de búsqueda en este caso por nombre del proveedor

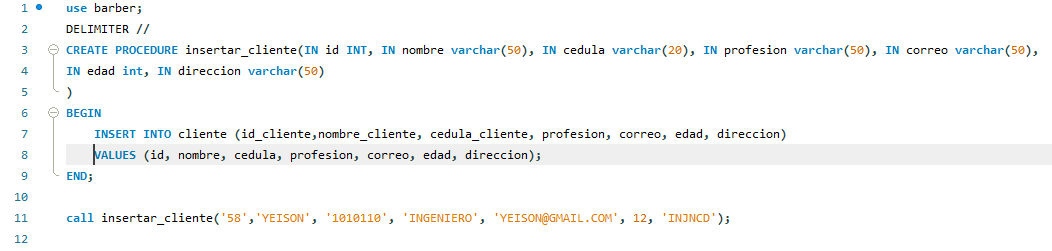


1. Vista que me permite ver la cantidad de productos que compro un cliente y su factura asociada a dicha compra.

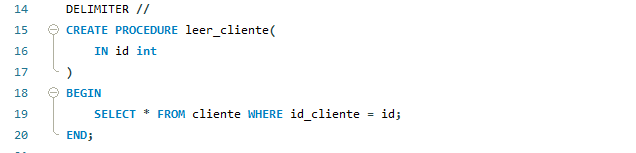


# Generar al menos 4 procedimientos almacenados.

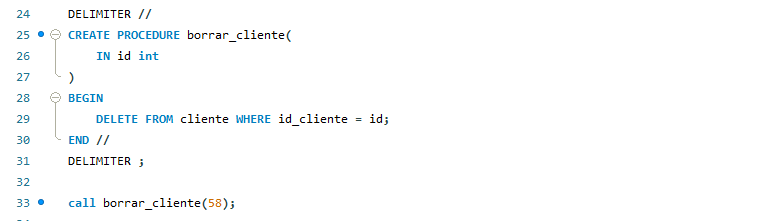
Procedimiento insertar un cliente



Procedimiento leer cliente



Procedimiento borrar un cliente

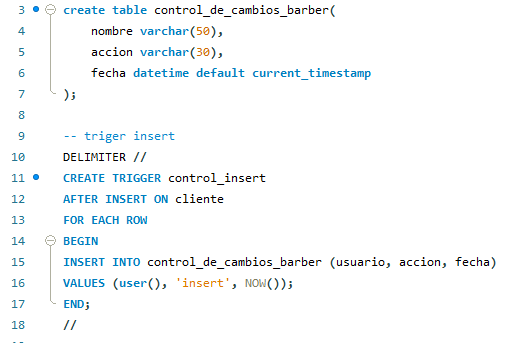


Procedimiento para actualizar cliente

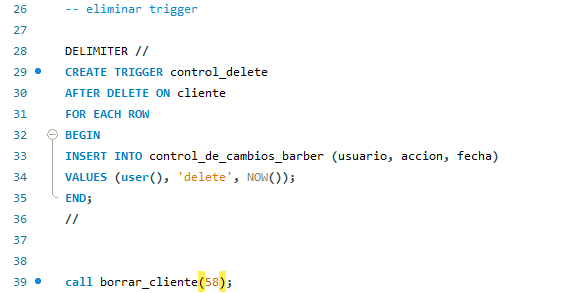


# Generar al menos 4 triggers

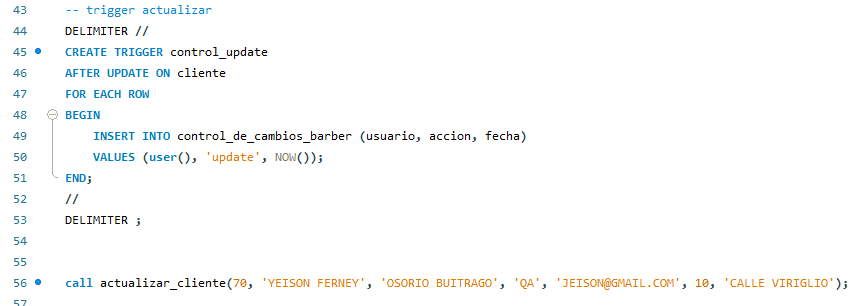
Trigger para insert en la tabla cliente



Trigger para eliminar en la tabla cliente



Trigger para actualizar en la tabla paciente



# Poblar la base de datos (50 registros por tabla) utilizando una conexión desde Java.

Proyecto

# Al terminar el ejercicio responda ¿ Está conforme con el resultado obtenido según el contexto o cree que hubiera obtenido un mejor resultado con una base de datos no relacional?

Si estoy conforme con el modelo y creo que fui un poco más allá de la información especificada ya que el modelo me permite ver desde el cliente que realiza una reserva para una cita de un servicio en la Barberia y el precio de ese servicio y esos insumos y la factura que le genera, cubre todo el Core del negocio.

Crear la base de datos en un modelo no relacional seria mas sencillo ya que no necesita modelar la base de datos, pero no sería escalable al largo plazo ya que las bases de datos no relaciones no son tan escalables y no importa muchos la consistencia de los datos al contrario de las relacional ya que nos permite tener una integridad de la información y escalar de forma lógica la base de datos y crear unas consultas complejas en el modelo.

documente muy bien su proceso (paso a paso) en un archivo PDF escriba todas las aclaraciones o especificaciones necesarias para realizar el ejercicio