



**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA  
CARRERA DE SOFTWARE**

**Base de Datos 1**

**Informe de practica No. 2**

**1. DATOS GENERALES:**

**TEMA:** SQL-DML

**NOMBRE:** [REDACTED]

**CODIGO(S):** [REDACTED]

**FECHA DE ENTREGA:** 05/06/2024

**AMBIENTE DE APRENDIZAJE PRACTICO:** Laboratorio de Microsoft

**2. OBJETIVO:**

Crear scripts utilizando el Lenguaje Estructurado de Consulta (SQL) para la definición y manejo adecuado de los datos. El objetivo específico es que se pueda diseñar, implementar y gestionar esquemas de bases de datos, así como ejecutar consultas y operaciones DML para resolver problemas prácticos esto implica garantizar la integridad y consistencia de los datos, así como extraer información relevante y significativa para la toma de decisiones.

**3. INSTRUCCIONES:**

**Considérese el esquema de la base de datos relacional - Caso: ORDEN DE COMPRA**

VENDEDOR (idVen, nombreVen, sexo, salario, fechaIngreso, vigencia)

ORDEN\_COMPRA (numOC, idVen, idCli, fecha, idEstadoOC, valorTotal)

CLIENTE (idCli, apellidoCli, nombreCli, sexo, ciudadResidencia, tipoCli)

TIPOCLI (idTipo, descripcion)

DETALLE\_COMPRA (numOC, idProducto, cantidad, subtotal)

PRODUCTO (idProducto, nombre, precioUnitario, stock)

ESTADO\_COMPRA (idEstadoOC, descripcion)

PROVINCIA (idProvincia, nombre)

CIUDAD (idCiudad, nombre, idProvincia)

**Valor de dato permitido:**

VENDEDOR, CLIENTE:

Sexo: H: hombre, M: mujer

ESTADO\_COMPRA

idEstadoOC: 0,1,2,3

descripcion: 0: pedido de compra, 1: pago realizado, 2: pendiente de entrega, 3: realizada la entrega

TIPOCLI

idTipo: 0, 1, 2

descripcion: 0: consumidor final, 1: natural con cédula, 2: juridica con RUC

**INGRESO DE DATOS:**

Codifique los scripts SQL para el ingreso de filas en cada una de las tablas de la base de datos creada según el esquema especificado en el caso dado

## CONSULTAS CON SENTENCIAS SELECT:

Codifique los scripts SQL para las consultas solicitadas. Utilizar comandos SELECT (DML de selección) y sus cláusulas.

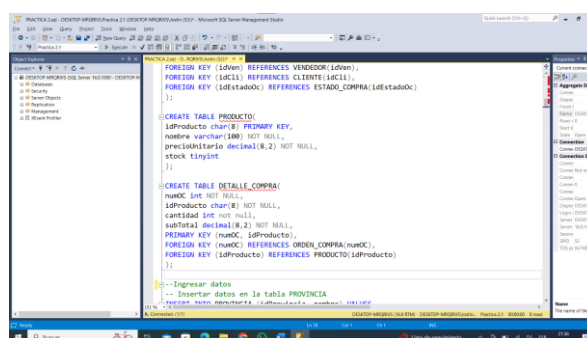
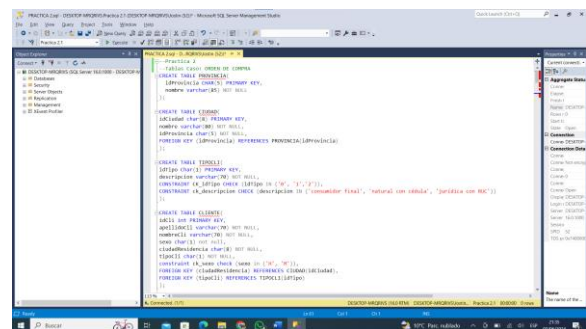
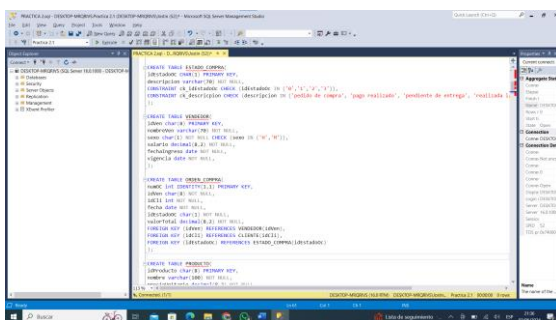
- Conocer los 10 mejores clientes de tipo natural del año 2023
- Listado alfabético de los clientes ordenados por el tipo de cliente
- Conocer el total de compras efectivas (con estado de compra: realizada la entrega) del mes de diciembre del año 2023
- Conocer los 3 mejores clientes de cada una de las Provincia que proceden en los últimos 3 años
- Conocer el número de clientes por su procedencia (ciudad y provincia)
- Conocer los productos que no se han vendido por cada mes del año 2023
- Cantidad vendida por producto durante el mes de diciembre del año 2023
- Conocer el número de vendedores por su sexo
- Listado de los vendedores que ingresaron en el último trimestre del año 2022
- Conocer la ciudad que con el mayor número de ordenes de compras se ha pedido
- Conocer el número de ordenes de compras de acuerdo con su estado
- Conocer las órdenes de compra pendientes de entrega
- Conocer los vendedores que tienen órdenes de compra pedidas y que no se hizo el pago correspondiente
- Conocer el mejor vendedor del año 2023. Se determina el mejor vendedor con base en el número de ventas realizadas
- Conocer el promedio de venta mensual (por cada mes) del año 2022
- Conocer los vendedores que cuyo promedio de ventas del año 2023 no pasaron del promedio en ventas anual.

## 4. PROCEDIMIENTO

### ACTIVIDADES DESARROLLADAS

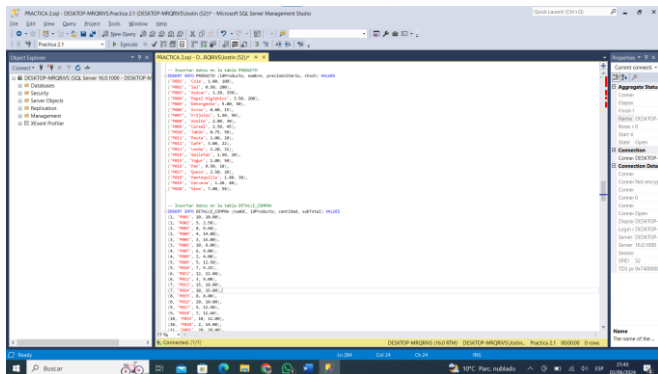
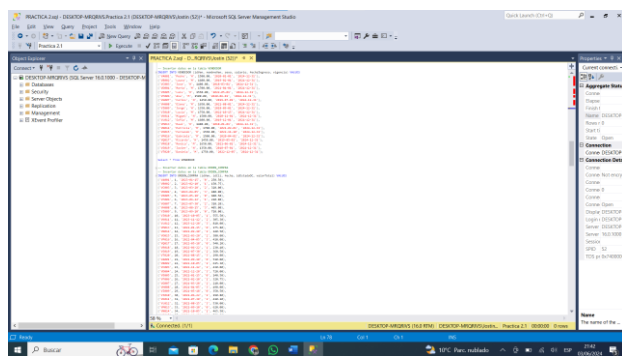
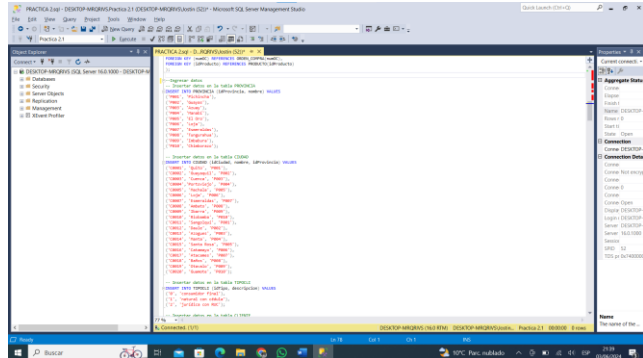
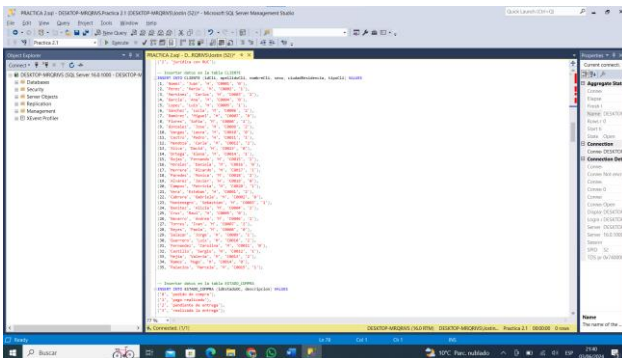
#### Caso: ORDEN DE COMPRA

El esquema de la base de datos relacional del caso "ORDEN DE COMPRA" proporciona la estructura necesaria para gestionar eficazmente el proceso de compra y venta de productos este esquema está diseñado para modelar entidades como vendedores, clientes, productos y órdenes de compra, junto con sus respectivas relaciones y atributos. A través de este conjunto de tablas, se puede llevar un seguimiento completo de las transacciones comerciales asegurando la integridad de los datos y facilitando la obtención de información relevante para la toma de decisiones empresariales.



## Ingreso de datos:

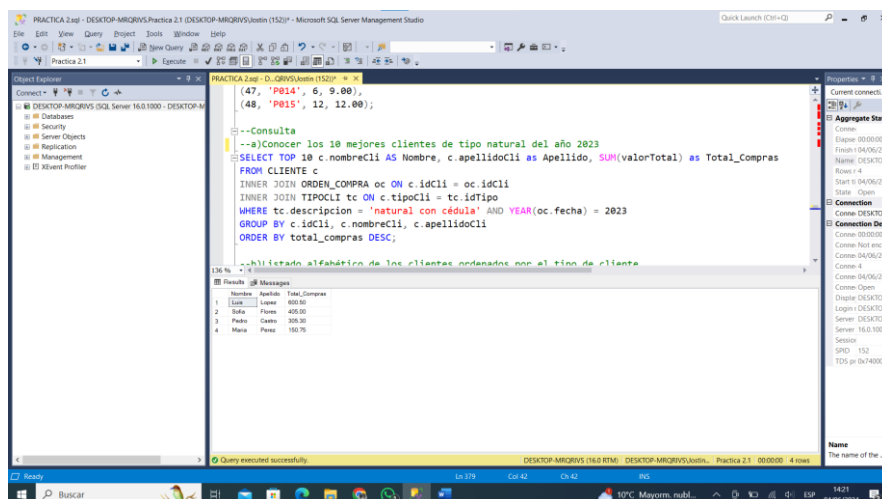
La fase de ingreso de datos en las tablas de la base de datos del caso "ORDEN DE COMPRA" es esencial para establecer un sistema funcional los scripts SQL proporcionan una forma estructurada de insertar filas de datos en cada tabla este proceso es fundamental para establecer una base de datos completa y precisa, lo que facilitará la gestión y toma de decisiones en el futuro.



## Consultas con sentencias select:

Codifique los scripts SQL para las consultas solicitadas. Utilizar comandos SELECT (DML de selección) y sus cláusulas las consultas con sentencias SELECT son esenciales en el manejo de bases de datos para obtener información específica y relevante en este contexto se han diseñado consultas SQL para abordar distintos aspectos del sistema de órdenes de compra lo que permitirá realizar análisis y tomar decisiones informadas en el entorno empresarial.

a) Conocer los 10 mejores clientes de tipo natural del año 2023



La consulta SQL presentada tiene como objetivo identificar y enumerar a los 10 clientes que han realizado las mayores compras totales durante el año 2023 específicamente entre aquellos clasificados como "clientes naturales con cédula". Para ello, la consulta comienza seleccionando los campos relevantes, que incluyen el nombre y apellido del cliente, así como la suma total de los montos de compra. Posteriormente, se realiza un

b) Listado alfabético de los clientes ordenados por el tipo de cliente

Practica 2.sql - D:\DBYustin (152)\* - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Practica 2

Object Explorer

Connect +

Microsoft SQL Server SQL Server 16.0 - D:\DBYustin (152)\*

Practica 2.sql - D:\DBYustin (152)\*

```
--b)Listado alfabético de los clientes ordenados por el tipo de cliente

SELECT c.nombreCI as Nombre, c.apellidoCI as Apellido, TC.descripcion AS Tipo_cliente
FROM cliente c
INNER JOIN TIPOCLI TC ON c.tipoCI = tc.idTipo
ORDER BY tipo_cliente;
```

100 % - 1

Results Messages

Nombre	Apellido	Tipo_Cliente
1	Alan	comercio final
2	Ana	comercio final
3	Miguel	comercio final
4	Luis	comercio final
5	David	comercio final
6	Diego	comercio final
7	Jaime	comercio final
8	Isabella	comercio final
9	Raul	comercio final
10	Paula	comercio final
11	Carolina	comercio final
12	Hugo	comercio final
13	Carmen	comercio final
14	Luis	comercio final
15	Jose	comercio final
16	Carla	comercio final
17	Fernando	comercio final
18	Monica	comercio final
19	Esteban	comercio final
20	Alicia	comercio final
21	Isai	comercio final
22	Luis	comercio final
23	Vicente	comercio final
24	Marta	comercio final
25	Luis	comercio final

Query executed successfully

Microsoft SQL Server 16.0 - D:\DBYustin (152)\*

Practica 2 00:00:00 35 rows

El JOIN se realiza en función del campo "tipoCli" de la tabla de clientes el cual se relaciona con el campo "idTipo" de la tabla de tipos de clientes. De esta manera, se logra obtener la información completa de cada cliente, incluyendo su tipo. Finalmente, la consulta ordena los resultados de manera ascendente, tomando como criterio de ordenamiento el campo "Tipo\_Cliente". De esta forma, se genera un listado alfabético de los clientes, agrupados y presentados según el tipo de cliente al que pertenecen esta consulta puede ser de gran utilidad para la empresa ya que le permite tener una visión general de su base de clientes, organizada por tipo de cliente. Esto puede facilitar la implementación de estrategias de segmentación y personalización de servicios, adaptadas a las necesidades y características de cada grupo de clientes.

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The central pane shows the 'Query Editor' with a T-SQL query. The query is as follows:

```
--c)Conocer el total de compras efectivas (con estado de compra: realizada la entrega) del mes de
--diciembre del año 2023

--SELECT SUM(oc.valorTotal) AS 'Total de Compras'
FROM ORDEN_COMPRA oc
INNER JOIN ESTADO_COMPRA ec ON oc.idestadoOC = ec.idestadoOC
WHERE ec.descripcion = 'realizada la entrega' AND YEAR(oc.fecha) = 2023 AND MONTH(oc.fecha) = 12;
```

Below the query, the 'Results' tab shows the execution output:

Total de Compras
870.00

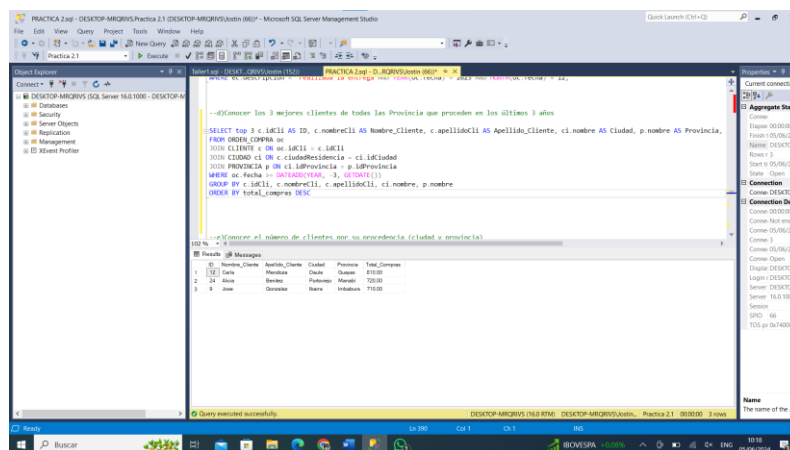
The status bar at the bottom indicates: 'Query executed successfully. DESKTOP-MIRQIRVS (16.0 RTM) DESKTOP-MIRQIRVS\sqlservr - Practica 2.1 00:00:00 1 rows'.

La consulta SQL presentada tiene como objetivo calcular el total de compras efectivas, es decir, aquellas cuyo estado de la orden de compra es "realizada la entrega" durante el mes de diciembre del año 2023.

La consulta comienza seleccionando la suma total (SUM) del campo "valorTotal" de la tabla de órdenes de compra (ORDEN\_COMPRA) para obtener únicamente las compras efectivas se realiza un JOIN interno entre la tabla de órdenes de compra y la tabla de estados de compra (ESTADO\_COMPRA) de modo que se puedan filtrar los registros donde el estado de la orden de compra corresponde a "realizada la entrega".

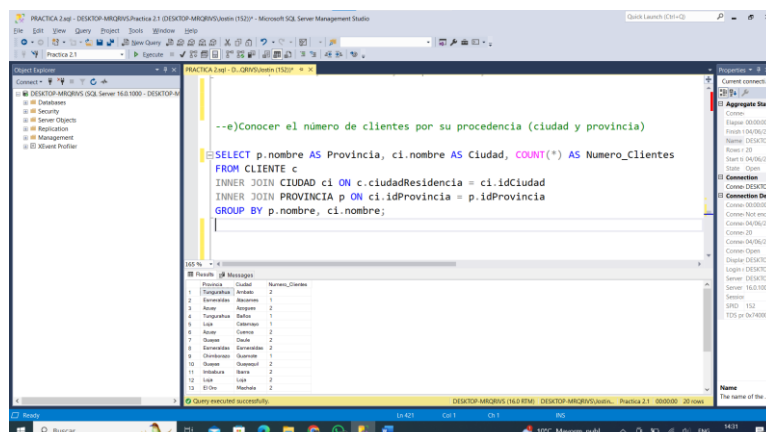
Posteriormente la consulta aplica un filtro mediante la cláusula WHERE donde se especifica que se deben considerar únicamente aquellas órdenes de compra cuya fecha corresponde al año 2023 y al mes de diciembre esto se logra utilizando las funciones YEAR y MONTH para extraer el año y el mes de la fecha de la orden de compra, respectivamente al realizar este proceso, la consulta calcula la suma total de los valores de las órdenes de compra que cumplen con los criterios de fecha y estado de entrega, obteniendo así el total de compras efectivas realizadas durante el mes de diciembre de 2023 esta información puede ser de gran utilidad para la empresa ya que le permite conocer el desempeño de las ventas y la efectividad de las entregas durante un periodo específico. Esto puede ser valioso para la planificación de inventarios, la evaluación de estrategias de ventas y la toma de decisiones relacionadas con la gestión de la cadena de suministro.

d) Conocer los 3 mejores clientes de cada una de las Provincia que proceden en los últimos 3 años



La consulta SQL está diseñada para identificar los tres mejores clientes, en términos de valor total de compras, de cada provincia durante los últimos tres años. Para ello, realiza una combinación (JOIN) de varias tablas: ORDEN\_COMPRA, para obtener los datos de las compras; CLIENTE, para obtener la información de los clientes; CIUDAD, para determinar la ciudad de residencia de los clientes; y PROVINCIA, para identificar la provincia correspondiente a cada ciudad. La cláusula WHERE filtra las órdenes de compra realizadas en los últimos tres años, utilizando la función DATEADD. Los resultados se agrupan por el ID del cliente, nombre, apellido, ciudad y provincia, y se ordenan en orden descendente según el total de compras (SUM(oc.valorTotal)). Finalmente, se aplica la cláusula SELECT TOP 3 para limitar la salida a los tres clientes con el mayor valor de compras en cada provincia.

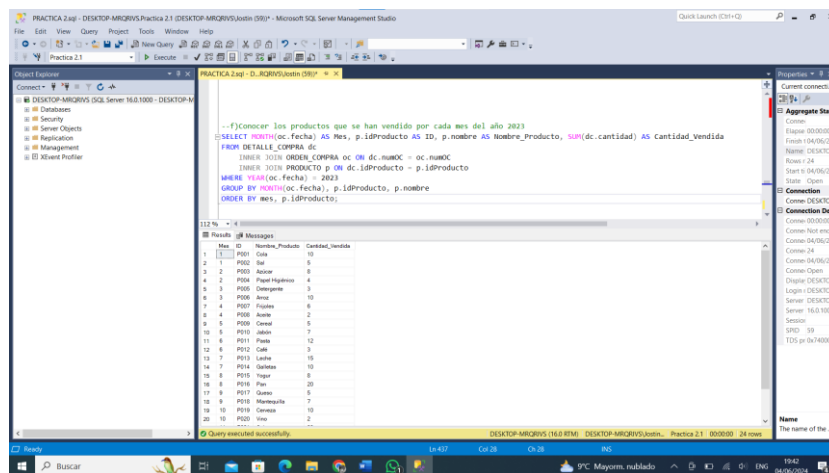
e) Conocer el número de clientes por su procedencia (ciudad y provincia)



La consulta SQL presentada tiene como objetivo obtener información sobre el número de clientes por su procedencia, es decir, la ciudad y provincia donde residen. La consulta comienza seleccionando los campos de nombre de provincia, nombre de ciudad y el conteo de clientes (COUNT(\*)) como Número\_Clientes. Estos datos se obtienen a partir de la tabla de clientes (CLIENTE), mediante la utilización de dos JOIN internos.

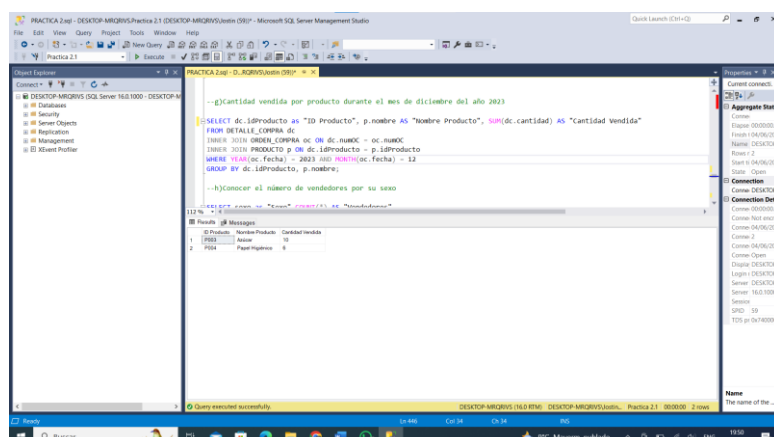
El primer JOIN se realiza entre la tabla de clientes y la tabla de ciudades (CIUDAD), utilizando el campo "ciudadResidencia" de la tabla de clientes, el cual se relaciona con el campo "idCiudad" de la tabla de ciudades de esta manera, se obtiene la información de la ciudad de residencia de cada cliente. Luego, se realiza un segundo JOIN entre la tabla de ciudades y la tabla de provincias (PROVINCIA) utilizando el campo "idProvincia" de la tabla de ciudades el cual se relaciona con el campo "idProvincia" de la tabla de provincias. Esto permite obtener la información de la provincia a la que pertenece cada ciudad. Finalmente, la consulta agrupa los resultados por provincia y ciudad, y aplica la función COUNT(\*) para obtener el número de clientes por cada combinación de provincia y ciudad. De esta forma se genera un reporte que muestra el número de clientes por su procedencia, desglosado por provincia y ciudad.

f) Conocer los productos que no se han vendido por cada mes del año 2023



Para obtener estos datos la consulta realiza un análisis exhaustivo de la información contenida en diversas tablas de la base de datos, como Detalle de Compra, Orden de Compra y Producto. Mediante la utilización de JOIN internos se logra establecer las relaciones necesarias entre estos objetos de datos permitiendo extraer los campos relevantes para el informe requerido la consulta comienza seleccionando el mes de la fecha de la orden de compra el identificador y el nombre del producto, así como la suma total de las cantidades vendidas. Posteriormente, se aplica un filtro que considera únicamente las órdenes de compra realizadas durante el año 2023. Finalmente, los resultados se agrupan por mes y por producto, y se ordenan en función de estas variables de este modo se genera un reporte que brinda una visión detallada de la demanda de los diferentes productos a lo largo del año 2023. Esta información puede resultar sumamente útil para la empresa ya que le permite planificar la producción gestionar los inventarios y tomar decisiones estratégicas relacionadas con la oferta de productos de manera más informada y fundamentada.

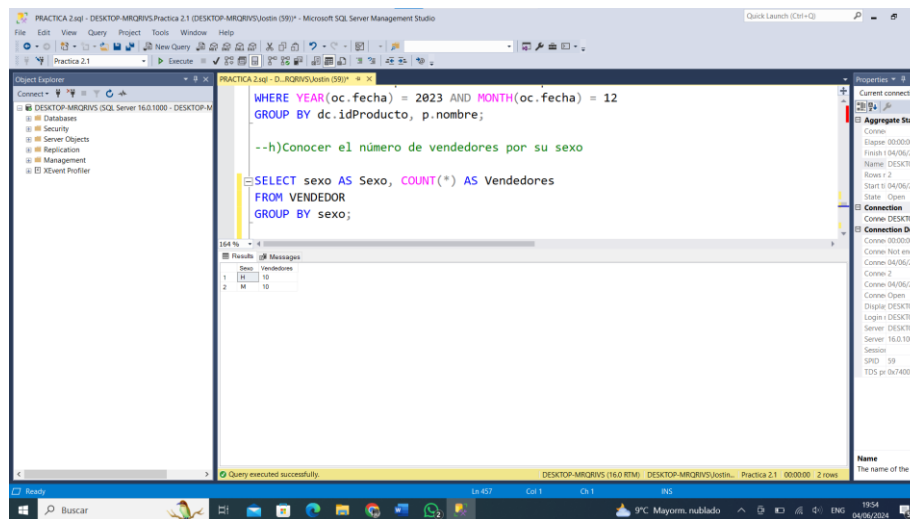
g) Cantidad vendida por producto durante el mes de diciembre del año 2023





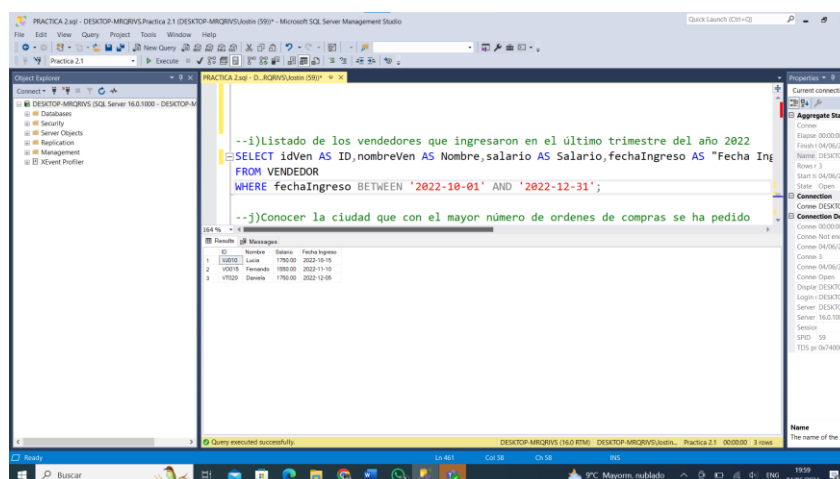
La consulta realiza una serie de operaciones sobre las tablas Detalle de Compra, Orden de Compra y Producto. Primero, se seleccionan los campos relevantes: el identificador del producto, el nombre del producto y la suma de las cantidades vendidas. Luego, se utilizan JOIN internos para establecer las relaciones entre estas tablas y obtener la información necesaria. La cláusula WHERE se encarga de filtrar los resultados considerando únicamente las órdenes de compra realizadas durante el año 2023 y en el mes de diciembre. Finalmente, los datos se agrupan por el identificador y el nombre del producto de modo que se pueda calcular la cantidad total vendida de cada uno. De esta manera el reporte generado proporciona información detallada sobre la cantidad de cada producto vendida durante el mes de diciembre de 2023 esta información puede ser de gran utilidad para la empresa ya que le permite analizar el rendimiento de sus productos en períodos específicos lo que puede ser fundamental para la planificación de la producción, la gestión de inventarios y la toma de decisiones estratégicas relacionadas con la oferta de productos.

h) Conocer el número de vendedores por su sexo



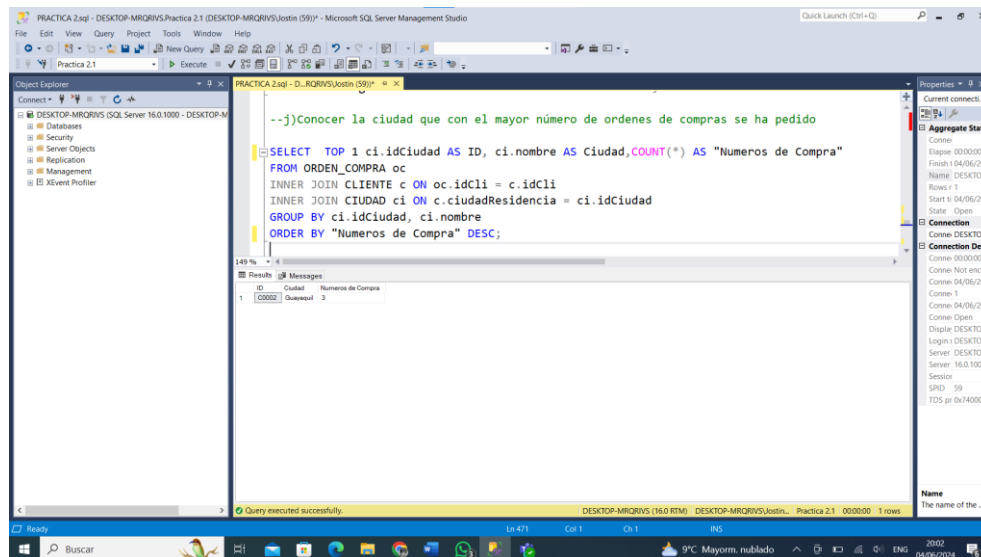
La consulta SQL presenta una manera eficiente de analizar la distribución de vendedores según su sexo en una base de datos al seleccionar el campo "sexo" de la tabla VENDEDOR y utilizarlo como criterio de agrupación, la consulta forma grupos distintos para cada valor único de sexo. Posteriormente, la función de agregación COUNT(\*) calcula el número de vendedores en cada grupo. Al final, la instrucción GROUP BY asegura que los resultados se presenten de forma ordenada y clara proporcionando así una visión detallada y cuantitativa de la composición de los vendedores en función de su sexo esta consulta es una forma efectiva y directa de obtener información sobre la distribución de género dentro de la fuerza de ventas sin necesidad de recurrir a expresiones condicionales adicionales.

i) Listado de los vendedores que ingresaron en el último trimestre del año 2022



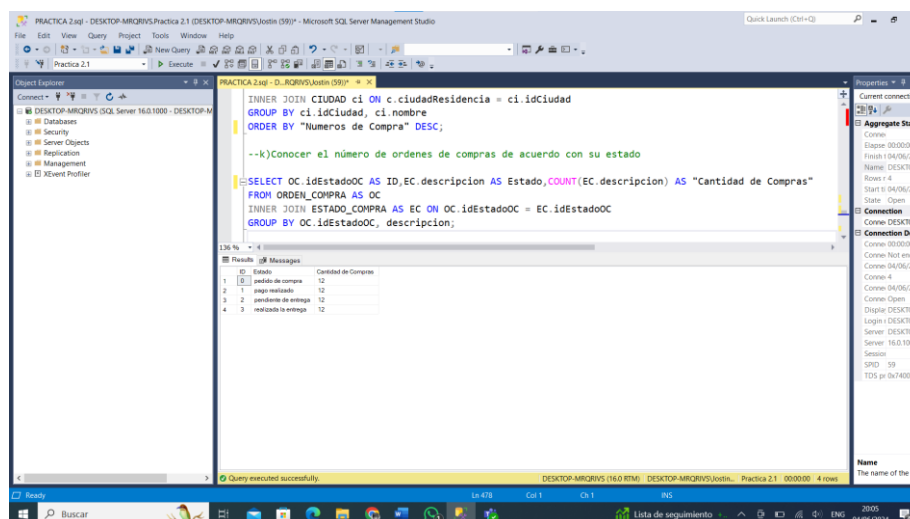
La consulta SQL proporciona un listado de los vendedores que ingresaron en el último trimestre del año 2022. Utilizando la tabla VENDEDOR la consulta selecciona los campos relevantes incluyendo el ID del vendedor (IDVen), el nombre del vendedor (nombreVen), su salario y la fecha de ingreso (fechaIngreso). La cláusula WHERE filtra los resultados para incluir solo aquellos registros cuya fecha de ingreso se encuentre entre el 1 de octubre de 2022 y el 31 de diciembre de 2022 lo que representa el último trimestre del año 2022. Por lo tanto, el resultado de esta consulta proporciona un listado detallado de los vendedores que se unieron a la empresa durante ese periodo específico lo que facilita el seguimiento y análisis de los nuevos ingresos en el último trimestre del año 2022.

j) Conocer la ciudad que con el mayor número de ordenes de compras se ha pedido



La consulta SQL tiene como objetivo identificar la ciudad que ha registrado el mayor número de órdenes de compra realizadas para lograr esto se lleva a cabo una unión entre las tablas ORDEN\_COMPRA, CLIENTE y CIUDAD vinculando las órdenes de compra con los clientes y sus ciudades de residencia respectivas. Luego, se agrupan los resultados por el identificador de ciudad (IDCiudad) y el nombre de la ciudad y se utiliza la función de agregación COUNT(\*) para contar el número de órdenes de compra asociadas con cada ciudad la cláusula ORDER BY se utiliza para ordenar los resultados de forma descendente según el número de compras. Finalmente, la cláusula TOP 1 selecciona solo la primera fila es decir la ciudad con el mayor número de órdenes de compra. Por lo tanto, el resultado de esta consulta proporciona la ciudad con la mayor cantidad de órdenes de compra realizadas, lo que ayuda a identificar los centros de actividad comercial más activos.

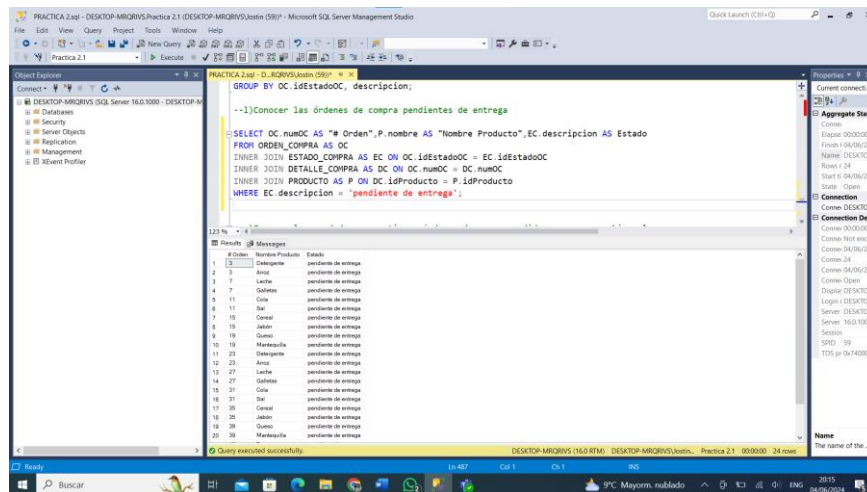
k) Conocer el número de ordenes de compras de acuerdo con su estado





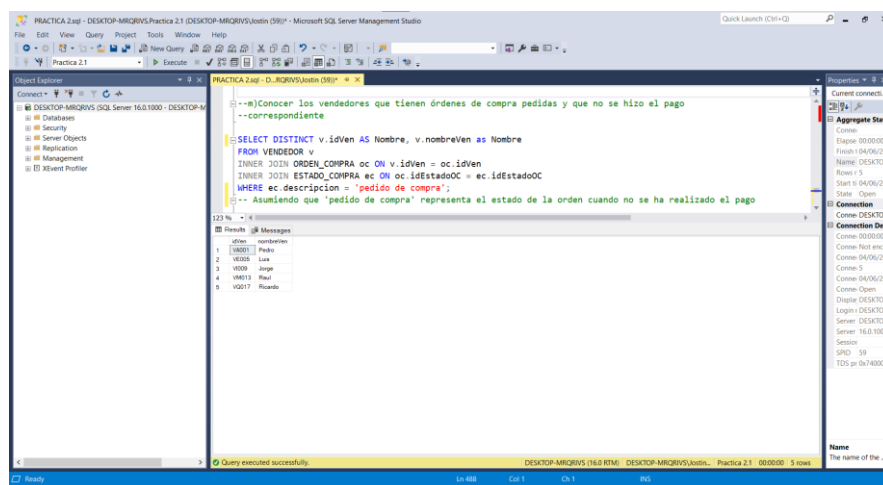
La consulta SQL tiene como objetivo conocer el número de órdenes de compra según su estado para lograr esto se realiza una unión entre las tablas ORDEN\_COMPRA y ESTADO\_COMPRA utilizando el campo idEstadoOC para vincular las órdenes de compra con sus respectivos estados. Luego, se agrupan los resultados por el identificador de estado de la orden de compra (IDEstadoOC) y la descripción del estado (descripcion) utilizando la función de agregación COUNT para contar el número de órdenes de compra asociadas con cada estado. De esta manera, la consulta proporciona un resumen detallado del número de órdenes de compra para cada estado lo que permite una mejor comprensión de la distribución de las compras según su estado en el sistema.

#### l) Conocer las órdenes de compra pendientes de entrega



La consulta SQL busca obtener detalles específicos de las órdenes de compra que se encuentran en estado "pendiente de entrega". A través de una serie de uniones entre las tablas ORDEN\_COMPRA, ESTADO\_COMPRA, DETALLE\_COMPRA y PRODUCTO, se relacionan las órdenes de compra con su estado correspondiente, así como con los productos asociados a cada orden. La cláusula WHERE filtra los resultados para incluir solo las órdenes de compra en estado "pendiente de entrega" proporcionando así una lista detallada que incluye el número de orden, el nombre del producto y el estado de cada orden. Esta consulta permite identificar y monitorear fácilmente las órdenes de compra que aún no han sido entregadas.

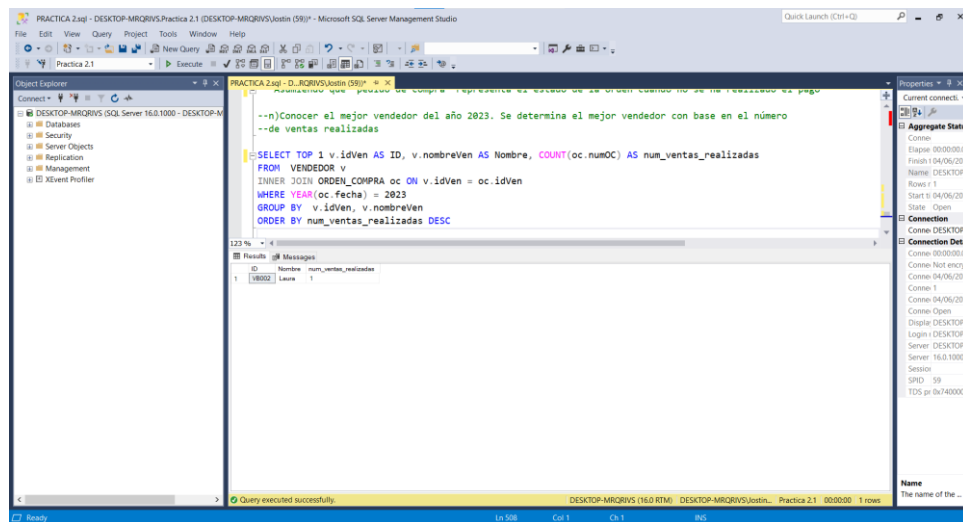
#### m) Conocer los vendedores que tienen órdenes de compra pedidas y que no se hizo el pago correspondiente



La consulta SQL busca identificar los vendedores que tienen órdenes de compra en estado "pedido de compra", lo que sugiere que el pago correspondiente no se ha realizado aún se realiza una unión entre las tablas VENDEDOR, ORDEN\_COMPRA y ESTADO\_COMPRA vinculando los vendedores con las órdenes de compra y sus estados respectivos. La cláusula WHERE filtra los resultados para incluir solo las órdenes de compra en estado "pedido de compra", lo que indica que no se ha realizado el pago correspondiente la

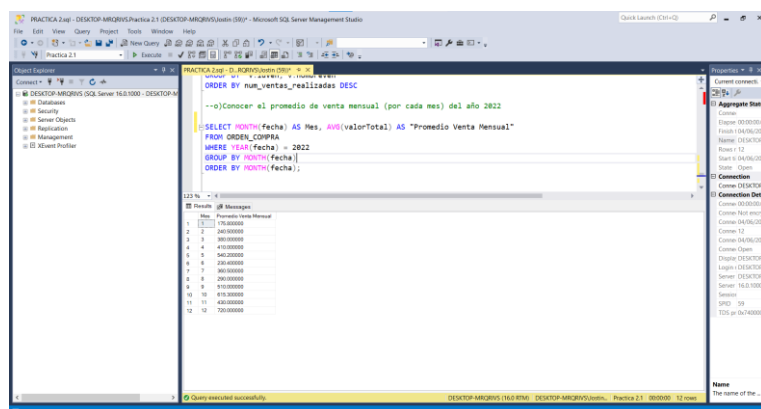
utilización de DISTINCT garantiza que cada vendedor aparezca solo una vez en los resultados. Por lo tanto, la consulta proporciona una lista de los vendedores que tienen órdenes de compra pendientes de pago lo que permite identificar fácilmente aquellos que requieren atención o seguimiento en términos de pagos pendientes.

- n) Conocer el mejor vendedor del año 2023. Se determina el mejor vendedor con base en el número de ventas realizadas



La consulta SQL tiene como objetivo identificar al mejor vendedor del año 2023 basándose en el número de ventas realizadas se realiza una unión entre las tablas VENDEDOR y ORDEN\_COMPRA, vinculando los vendedores con las órdenes de compra que han realizado. La cláusula WHERE filtra los resultados para incluir solo las órdenes de compra del año 2023 utilizando la función YEAR(oc.fecha). Luego se agrupan los resultados por el identificador del vendedor (IDVen) y su nombre (nombreVen), utilizando la función de agregación COUNT para contar el número de órdenes de compra asociadas con cada vendedor la cláusula ORDER BY ordena los resultados en orden descendente según el número de ventas realizadas y la cláusula TOP 1 selecciona solo el primer registro es decir al vendedor con el mayor número de ventas realizadas en el año 2023. Por lo tanto, la consulta proporciona el ID y el nombre del mejor vendedor del año 2023, lo que facilita la identificación del vendedor más productivo en términos de ventas realizadas durante ese periodo.

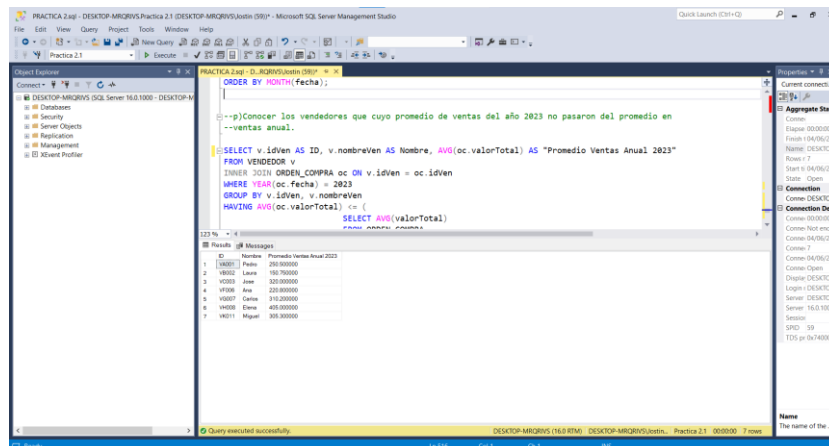
- o) Conocer el promedio de venta mensual (por cada mes) del año 2022



La consulta SQL tiene como objetivo calcular el promedio de ventas mensuales durante el año 2022. Selecciona el mes utilizando la función MONTH(fecha) para extraer el mes de la fecha de la orden de compra, y calcula el promedio de los valores totales de las órdenes de compra utilizando la función AVG(valorTotal). La cláusula WHERE filtra los resultados para incluir solo las órdenes de compra del año 2022 utilizando la función YEAR(fecha). Luego, se agrupan los resultados por mes utilizando la función GROUP BY MONTH(fecha) lo que permite calcular el promedio de ventas para cada mes por separado. Finalmente, la cláusula ORDER BY ordena los resultados por mes en orden ascendente. Por lo tanto, la consulta proporciona

el promedio de venta mensual para cada mes del año 2022 lo que permite analizar y entender mejor el rendimiento de las ventas a lo largo del año.

p) Conocer los vendedores que cuyo promedio de ventas del año 2023 no pasaron del promedio en ventas anuales.



La consulta SQL tiene como objetivo identificar los vendedores cuyo promedio de ventas durante el año 2023 no superó el promedio de ventas anual en general se realiza una unión entre las tablas VENDEDOR y ORDEN\_COMPRA vinculando los vendedores con las órdenes de compra que han realizado. La cláusula WHERE filtra los resultados para incluir solo las órdenes de compra del año 2023 utilizando la función YEAR(oc.fecha). Luego se agrupan los resultados por el identificador del vendedor (IDVen) y su nombre (nombreVen), y se calcula el promedio de ventas para cada vendedor utilizando la función de agregación AVG(oc.valorTotal). La cláusula HAVING se utiliza para filtrar los resultados incluyendo solo aquellos vendedores cuyo promedio de ventas no exceda el promedio de ventas anual del año 2023 esto se logra mediante una subconsulta que calcula el promedio de ventas anual para todo el año 2023 y compara este promedio con el promedio de ventas de cada vendedor. Por lo tanto, la consulta proporciona una lista de los vendedores que no alcanzaron el promedio de ventas anual en el año 2023, lo que facilita la identificación de aquellos que podrían necesitar mejorar su desempeño en términos de ventas.

Se ha completado con éxito la actividad, abordando todos los casos presentados para practicar y demostrar habilidades en consultas SQL. El script a continuación no solo abarca las consultas diseñadas para cumplir con los requisitos específicos de la actividad, sino que también incluye la creación de las tablas necesarias y la inserción de datos para simular un entorno de base de datos completo. Esto proporciona un contexto completo para las consultas, permitiendo su ejecución y validación de resultados.

## Caso: ORDEN DE COMPRA

Considérese el esquema de la base de datos relacional

```
--Practica 2
--Tablas Caso: ORDEN DE COMPRA
CREATE TABLE PROVINCIA(
    idProvincia CHAR(5) PRIMARY KEY,
    nombre VARCHAR(85) NOT NULL
);

CREATE TABLE CIUDAD(
    idCiudad CHAR(8) PRIMARY KEY,
    nombre VARCHAR(80) NOT NULL,
    idProvincia CHAR(5) NOT NULL,
    FOREIGN KEY (idProvincia) REFERENCES PROVINCIA(idProvincia)
);

CREATE TABLE TIPOCLI(
    idTipo CHAR(1) PRIMARY KEY,
    descripcion VARCHAR(70) NOT NULL,
    CONSTRAINT CK_idTipo CHECK (idTipo IN ('0', '1', '2')),
    CONSTRAINT ck_descripcion CHECK (descripcion IN ('consumidor final',
'natural con cédula', 'jurídica con RUC'))
);

CREATE TABLE CLIENTE(
    idCli INT PRIMARY KEY,
    apellidoCli VARCHAR(70) NOT NULL,
    nombreCli VARCHAR(70) NOT NULL,
    sexo CHAR(1) NOT NULL,
    ciudadResidencia CHAR(8) NOT NULL,
    tipoCli CHAR(1) NOT NULL,
);

CREATE TABLE ESTADO_COMPRA(
    idEstadoOC CHAR(1) PRIMARY KEY,
    descripcion VARCHAR(70) NOT NULL,
    CONSTRAINT ck_idEstadoOC CHECK (idEstadoOC IN ('0', '1', '2', '3')),
    CONSTRAINT ck_descripcion CHECK (descripcion IN ('pedido de compra',
'pago realizado', 'pendiente de entrega', 'realizada la entrega'))
);

CREATE TABLE ORDEN_COMPRA(
    numOC INT IDENTITY(1,1) PRIMARY KEY,
    idVen CHAR(8) NOT NULL,
    idCli INT NOT NULL,
    fecha DATE NOT NULL,
    idEstadoOC CHAR(1) NOT NULL,
    valorTotal DECIMAL(8,2) NOT NULL,
    FOREIGN KEY (idVen) REFERENCES VENDEDOR(idVen),
    FOREIGN KEY (idCli) REFERENCES CLIENTE(idCli),
    FOREIGN KEY (idEstadoOC) REFERENCES ESTADO_COMPRA(idEstadoOC)
);

CREATE TABLE PRODUCTO(
    idProducto CHAR(8) PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    precioUnitario DECIMAL(8,2) NOT NULL,
    stock TINYINT
);

CREATE TABLE DETALLE_COMPRA(
```

```

constraint ck_sexo check (sexo in ('H', 'M')),
FOREIGN KEY (ciudadResidencia) REFERENCES CIUDAD(idCiudad),
FOREIGN KEY (tipoCli) REFERENCES TIPOCLI(idTipo)
);

CREATE TABLE VENDEDOR (
idVen char(8) PRIMARY KEY,
nombreVen varchar(70) NOT NULL,
sexo char(1) NOT NULL CHECK (sexo IN ('H','M')),
salario decimal(8,2) NOT NULL,
fechaIngreso date NOT NULL,
vigencia date NOT NULL,
);

```

```

numOC int NOT NULL,
idProducto char(8) NOT NULL,
cantidad int not null,
subTotal decimal(8,2) NOT NULL,
PRIMARY KEY (numOC, idProducto),
FOREIGN KEY (numOC) REFERENCES ORDEN_COMPRA(numOC),
FOREIGN KEY (idProducto) REFERENCES PRODUCTO(idProducto)
);

```

## Ingreso de datos:

Codifique los scripts SQL para el ingreso de filas en cada una de las tablas de la base de datos creada según el esquema especificado en el caso dado.

```

--Ingresar datos
-- Insertar datos en la tabla PROVINCIA
INSERT INTO PROVINCIA (idProvincia, nombre) VALUES
('P001', 'Pichincha'),
('P002', 'Guayas'),
('P003', 'Azuay'),
('P004', 'Manabí'),
('P005', 'El Oro'),
('P006', 'Loja'),
('P007', 'Esmeraldas'),
('P008', 'Tungurahua'),
('P009', 'Imbabura'),
('P010', 'Chimborazo');

-- Insertar datos en la tabla CIUDAD
INSERT INTO CIUDAD (idCiudad, nombre, idProvincia) VALUES
('C0001', 'Quito', 'P001'),
('C0002', 'Guayaquil', 'P002'),
('C0003', 'Cuenca', 'P003'),
('C0004', 'Portoviejo', 'P004'),
('C0005', 'Machala', 'P005'),
('C0006', 'Loja', 'P006'),
('C0007', 'Esmeraldas', 'P007'),
('C0008', 'Ambato', 'P008'),
('C0009', 'Ibarra', 'P009'),
('C0010', 'Riobamba', 'P010'),
('C0011', 'Sangolquí', 'P001'),
('C0012', 'Daule', 'P002'),
('C0013', 'Azogues', 'P003'),
('C0014', 'Manta', 'P004'),
('C0015', 'Santa Rosa', 'P005'),
('C0016', 'Catamayo', 'P006'),
('C0017', 'Atacames', 'P007'),
('C0018', 'Baños', 'P008'),
('C0019', 'Otavalo', 'P009'),
('C0020', 'Guamote', 'P010');

-- Insertar datos en la tabla TIPOCLI
INSERT INTO TIPOCLI (idTipo, descripcion) VALUES
('0', 'consumidor final'),
('1', 'natural con cédula'),
('2', 'jurídica con RUC');

-- Insertar datos en la tabla CLIENTE
INSERT INTO CLIENTE (idCli, apellidoCli, nombreCli, sexo, ciudadResidencia, tipoCli) VALUES
(1, 'Gomez', 'Juan', 'H', 'C0001', '0'),
(2, 'Perez', 'Maria', 'M', 'C0002', '1'),
(3, 'Martinez', 'Carlos', 'H', 'C0003', '2'),
(4, 'Garcia', 'Ana', 'M', 'C0004', '0'),
(5, 'Lopez', 'Luis', 'H', 'C0005', '1'),
(6, 'Sanchez', 'Lucia', 'M', 'C0006', '2'),
(7, 'Ramirez', 'Miguel', 'H', 'C0007', '0'),
(8, 'Flores', 'Sofia', 'M', 'C0008', '1'),
(9, 'Gonzalez', 'Jose', 'H', 'C0009', '2'),
(10, 'Vargas', 'Laura', 'M', 'C0010', '0'),
(11, 'Castro', 'Pedro', 'H', 'C0011', '1'),
(12, 'Mendoza', 'Carla', 'M', 'C0012', '2'),
(13, 'Silva', 'David', 'H', 'C0013', '0'),
(14, 'Ortega', 'Elena', 'M', 'C0014', '1'),
(15, 'Rojas', 'Fernando', 'H', 'C0015', '2'),
(16, 'Morales', 'Daniela', 'M', 'C0016', '0'),
(17, 'Herrera', 'Ricardo', 'H', 'C0017', '1'),
(18, 'Paredes', 'Monica', 'M', 'C0018', '2'),
(19, 'Alvarez', 'Javier', 'H', 'C0019', '0'),
(20, 'Campos', 'Patricia', 'M', 'C0020', '1'),
(21, 'Vera', 'Esteban', 'H', 'C0001', '2'),
(22, 'Cabrera', 'Gabriela', 'M', 'C0002', '0'),
(23, 'Montenegro', 'Sebastian', 'H', 'C0003', '1'),
(24, 'Benitez', 'Alicia', 'M', 'C0004', '2'),
(25, 'Cruz', 'Raul', 'H', 'C0005', '0'),
(26, 'Navarro', 'Andrea', 'M', 'C0006', '1'),
(27, 'Torres', 'Ivan', 'H', 'C0007', '2'),
(28, 'Reyes', 'Paola', 'M', 'C0008', '0'),
(29, 'Salazar', 'Jorge', 'H', 'C0009', '1'),
(30, 'Guerrero', 'Luis', 'H', 'C0010', '2'),
(31, 'Fernandez', 'Carolina', 'M', 'C0011', '0'),
(32, 'Castillo', 'Sergio', 'H', 'C0012', '1'),

```

```

-- Insertar datos en la tabla ORDEN_COMPRA
INSERT INTO ORDEN_COMPRA (idVen, idCli, fecha, idEstadoOC, valorTotal) VALUES
('VA001', 1, '2023-01-15', '0', 250.50),
('VB002', 2, '2023-02-10', '1', 150.75),
('VC003', 3, '2023-03-20', '2', 320.00),
('VD004', 4, '2023-04-05', '3', 480.00),
('VE005', 5, '2023-05-18', '0', 600.50),
('VF006', 6, '2023-06-22', '1', 220.80),
('VG007', 7, '2023-07-30', '2', 310.20),
('VH008', 8, '2023-08-15', '3', 405.00),
('VI009', 9, '2023-09-10', '0', 710.00),
('VJ010', 10, '2023-10-05', '1', 555.50),
('VK011', 11, '2023-11-12', '2', 305.30),
('VL012', 12, '2023-12-20', '3', 810.00),
('VM013', 13, '2022-01-15', '0', 175.80),
('VN014', 14, '2022-02-10', '1', 240.50),
('VO015', 15, '2022-03-20', '2', 380.00),
('VP016', 16, '2022-04-05', '3', 410.00),
('VQ017', 17, '2022-05-18', '0', 540.20),
('VR018', 18, '2022-06-22', '1', 230.40),
('VS019', 19, '2022-07-30', '2', 360.50),
('VT020', 20, '2022-08-15', '3', 290.00),
('VA001', 21, '2022-09-10', '0', 510.00),
('VB002', 22, '2022-10-05', '1', 615.30),
('VC003', 23, '2022-11-12', '2', 430.00),
('VD004', 24, '2022-12-20', '3', 720.00),
('VE005', 25, '2021-01-15', '0', 140.50),
('VF006', 26, '2021-02-10', '1', 320.75),
('VG007', 27, '2021-03-20', '2', 210.00),
('VH008', 28, '2021-04-05', '3', 490.00),
('VI009', 29, '2021-05-18', '0', 350.50),
('VJ010', 30, '2021-06-22', '1', 190.80),
('VK011', 31, '2021-07-30', '2', 410.20),
('VL012', 32, '2021-08-15', '3', 530.00),
('VM013', 33, '2021-09-10', '0', 620.00),
('VN014', 34, '2021-10-05', '1', 465.50),
('VO015', 35, '2021-11-12', '2', 580.30),
('VP016', 1, '2021-12-20', '3', 310.00),
('VQ017', 2, '2020-01-15', '0', 255.80),
('VR018', 3, '2020-02-10', '1', 330.50),
('VS019', 4, '2020-03-20', '2', 410.00),
('VT020', 5, '2020-04-05', '3', 510.00),
('VA001', 6, '2020-05-18', '0', 470.20),
('VB002', 7, '2020-06-22', '1', 230.40),
('VC003', 8, '2020-07-30', '2', 290.50),
('VD004', 9, '2020-08-15', '3', 380.00),
('VE005', 10, '2020-09-10', '0', 610.00),
('VF006', 11, '2020-10-05', '1', 465.30),
('VG007', 12, '2020-11-12', '2', 390.00),
('VH008', 13, '2020-12-20', '3', 580.00);

-- Insertar datos en la tabla DETALLE_COMPRA
INSERT INTO DETALLE_COMPRA (numOC, idProducto, cantidad, subTotal) VALUES
(1, 'P001', 10, 10.00),
(1, 'P002', 5, 2.50),
(2, 'P003', 8, 9.60),
(2, 'P004', 4, 14.00),
(3, 'P005', 3, 15.00),
(3, 'P006', 10, 8.00),
(4, 'P007', 6, 9.00),
(4, 'P008', 2, 4.00),
(5, 'P009', 5, 12.50),
(5, 'P010', 7, 5.25),
(6, 'P011', 12, 12.00),
(6, 'P012', 3, 9.00),
(7, 'P013', 15, 18.00),
(7, 'P014', 10, 15.00),
(8, 'P015', 8, 8.00),
(8, 'P016', 20, 10.00),
(9, 'P017', 5, 12.50),
(9, 'P018', 7, 12.60),
(10, 'P019', 10, 12.00),
(10, 'P020', 2, 14.00),
(11, 'P001', 20, 20.00),
(11, 'P002', 15, 7.50),
(12, 'P003', 10, 12.00),

```

```

(33, 'Mejia', 'Valeria', 'M', 'C0013', '2'),
(34, 'Ramos', 'Hugo', 'H', 'C0014', '0'),
(35, 'Palacios', 'Marcela', 'M', 'C0015', '1');

-- Insertar datos en la tabla ESTADO_COMPRA
INSERT INTO ESTADO_COMPRA (idEstadoOC, descripcion) VALUES
('0', 'pedido de compra'),
('1', 'pago realizado'),
('2', 'pendiente de entrega'),
('3', 'realizada la entrega');

-- Insertar datos en la tabla VENDEDOR
INSERT INTO VENDEDOR (idVen, nombreVen, sexo, salario, fechaIngreso,
vigencia) VALUES
('VA001', 'Pedro', 'H', 1500.00, '2020-01-01', '2024-12-31'),
('VB002', 'Laura', 'M', 1600.00, '2019-02-01', '2024-12-31'),
('VC003', 'Jose', 'H', 1400.00, '2018-03-01', '2024-12-31'),
('VD004', 'Marta', 'M', 1700.00, '2021-04-01', '2024-12-31'),
('VE005', 'Luis', 'H', 1550.00, '2022-05-01', '2024-12-31'),
('VF006', 'Ana', 'M', 1500.00, '2020-06-01', '2024-12-31'),
('VG007', 'Carlos', 'H', 1450.00, '2019-07-01', '2024-12-31'),
('VH008', 'Elena', 'M', 1650.00, '2021-08-01', '2024-12-31'),
('VI009', 'Jorge', 'H', 1350.00, '2018-09-01', '2024-12-31'),
('VJ010', 'Lucia', 'M', 1750.00, '2022-10-15', '2024-12-31'),
('VK011', 'Miguel', 'H', 1500.00, '2020-11-01', '2024-12-31'),
('VL012', 'Sofia', 'M', 1600.00, '2019-12-01', '2024-12-31'),
('VM013', 'Raul', 'H', 1400.00, '2018-01-01', '2024-12-31'),
('VN014', 'Patricia', 'M', 1700.00, '2021-02-01', '2024-12-31'),
('VO015', 'Fernando', 'H', 1550.00, '2022-11-10', '2024-12-31'),
('VP016', 'Gabriela', 'M', 1500.00, '2020-04-01', '2024-12-31'),
('VQ017', 'Ricardo', 'H', 1450.00, '2019-05-01', '2024-12-31'),
('VR018', 'Monica', 'M', 1650.00, '2021-06-01', '2024-12-31'),
('VS019', 'Javier', 'H', 1350.00, '2018-07-01', '2024-12-31'),
('VT020', 'Daniela', 'M', 1750.00, '2022-12-05', '2024-12-31');

-- Insertar datos en la tabla PRODUCTO
INSERT INTO PRODUCTO (idProducto, nombre, precioUnitario, stock)
VALUES
('P001', 'Cola', 1.00, 100),
('P002', 'Sal', 0.50, 200),
('P003', 'Azúcar', 1.20, 150),
('P004', 'Papel Higiénico', 3.50, 200),
('P005', 'Detergente', 5.00, 50),
('P006', 'Arroz', 0.80, 15),
('P007', 'Frijoles', 1.50, 50),
('P008', 'Aceite', 2.00, 30),
('P009', 'Cereal', 2.50, 45),
('P010', 'Jabón', 0.75, 50),
('P011', 'Pasta', 1.00, 20),
('P012', 'Café', 3.00, 22),
('P013', 'Leche', 1.20, 32),
('P014', 'Galletas', 1.50, 20),
('P015', 'Yogun', 1.00, 50),
('P016', 'Pan', 0.50, 10),
('P017', 'Queso', 2.50, 20),
('P018', 'Mantequilla', 1.80, 30),
('P019', 'Cerveza', 1.20, 40),
('P020', 'Vino', 7.00, 58);

(12, 'P004', 6, 21.00),
(13, 'P005', 8, 40.00),
(13, 'P006', 12, 9.60),
(14, 'P007', 4, 6.00),
(14, 'P008', 10, 20.00),
(15, 'P009', 8, 20.00),
(15, 'P010', 6, 4.50),
(16, 'P011', 5, 5.00),
(16, 'P012', 10, 30.00),
(17, 'P013', 12, 14.40),
(17, 'P014', 6, 9.00),
(18, 'P015', 10, 10.00),
(18, 'P016', 8, 4.00),
(19, 'P017', 3, 7.50),
(19, 'P018', 5, 9.00),
(20, 'P019', 4, 4.80),
(20, 'P020', 5, 35.00),
(21, 'P001', 7, 7.00),
(21, 'P002', 8, 4.00),
(22, 'P003', 5, 6.00),
(22, 'P004', 3, 10.50),
(23, 'P005', 6, 30.00),
(23, 'P006', 7, 5.60),
(24, 'P007', 9, 13.50),
(24, 'P008', 3, 6.00),
(25, 'P009', 6, 15.00),
(25, 'P010', 10, 7.50),
(26, 'P011', 15, 15.00),
(26, 'P012', 5, 15.00),
(27, 'P013', 8, 9.60),
(27, 'P014', 4, 6.00),
(28, 'P015', 7, 7.00),
(28, 'P016', 12, 6.00),
(29, 'P017', 6, 15.00),
(29, 'P018', 4, 7.20),
(30, 'P019', 5, 6.00),
(30, 'P020', 1, 7.00),
(31, 'P001', 12, 12.00),
(31, 'P002', 4, 2.00),
(32, 'P003', 15, 18.00),
(32, 'P004', 2, 7.00),
(33, 'P005', 9, 45.00),
(33, 'P006', 14, 11.20),
(34, 'P007', 3, 4.50),
(34, 'P008', 8, 16.00),
(35, 'P009', 10, 25.00),
(35, 'P010', 2, 1.50),
(36, 'P011', 9, 9.00),
(36, 'P012', 6, 18.00),
(37, 'P013', 11, 13.20),
(37, 'P014', 5, 7.50),
(38, 'P015', 9, 9.00),
(38, 'P016', 6, 3.00),
(39, 'P017', 4, 10.00),
(39, 'P018', 8, 14.40),
(40, 'P019', 3, 3.60),
(40, 'P020', 2, 14.00),
(41, 'P001', 8, 8.00),
(41, 'P002', 7, 3.50),
(42, 'P003', 12, 14.40),
(42, 'P004', 5, 17.50),
(43, 'P005', 10, 50.00),
(43, 'P006', 9, 7.20),
(44, 'P007', 7, 10.50),
(44, 'P008', 4, 8.00),
(45, 'P009', 9, 22.50),
(45, 'P010', 8, 6.00),
(46, 'P011', 14, 14.00),
(46, 'P012', 4, 12.00),
(47, 'P013', 7, 8.40),
(47, 'P014', 6, 9.00),
(48, 'P015', 12, 12.00);

```

## CONSULTAS CON SENTENCIAS SELECT:

Codifique los scripts SQL para las consultas solicitadas. Utilizar comandos SELECT (DML de selección) y sus cláusulas.

- a) Conocer los 10 mejores clientes de tipo natural del año 2023

```

--a) Conocer los 10 mejores clientes de tipo natural del año 2023

SELECT TOP 10 c.nombreCli AS Nombre, c.apellidoCli as Apellido, SUM(valorTotal) as
Total_Compras
FROM CLIENTE c
INNER JOIN ORDEN_COMPRA oc ON c.idCli = oc.idCli
INNER JOIN TIPOCLI tc ON c.tipoCli = tc.idTipo
WHERE tc.descripcion = 'natural con cédula' AND YEAR(oc.fecha) = 2023
GROUP BY c.idCli, c.nombreCli, c.apellidoCli
ORDER BY total_compras DESC;

```

b) Listado alfabético de los clientes ordenados por el tipo de cliente

```
--b)Listado alfabético de los clientes ordenados por el tipo de cliente

SELECT c.nombreCli as Nombre, c.apellidoCli as Apellido, TC.descripcion AS Tipo_Cliente
FROM CLIENTE c
INNER JOIN TIPOCLI tc ON c.tipoCli = tc.idTipo
ORDER BY tipo_cliente;
```

c) Conocer el total de compras efectivas (con estado de compra: realizada la entrega) del mes de diciembre del año 2023

```
--c)Conocer el total de compras efectivas (con estado de compra: realizada la entrega)
del mes de
--diciembre del año 2023

SELECT SUM(oc.valorTotal) AS "Total de Compras"
FROM ORDEN_COMPRA oc
INNER JOIN ESTADO_COMPRA ec ON oc.idEstadoOC = ec.idEstadoOC
WHERE ec.descripcion = 'realizada la entrega' AND YEAR(oc.fecha) = 2023 AND
MONTH(oc.fecha) = 12;
```

d) Conocer los 3 mejores clientes de cada una de las Provincia que proceden en los últimos 3 años

```
--d)Conocer los 3 mejores clientes de todas las Provincia que proceden en los últimos 3
años

SELECT top 3 c.idCli AS ID, c.nombreCli AS Nombre_Cliente, c.apellidoCli AS
Apellido_Cliente, ci.nombre AS Ciudad, p.nombre AS Provincia, SUM(oc.valorTotal) AS
Total_Compras
FROM ORDEN_COMPRA oc
JOIN CLIENTE c ON oc.idCli = c.idCli
JOIN CIUDAD ci ON c.ciudadResidencia = ci.idCiudad
JOIN PROVINCIA p ON ci.idProvincia = p.idProvincia
WHERE oc.fecha >= DATEADD(YEAR, -3, GETDATE())
GROUP BY c.idCli, c.nombreCli, c.apellidoCli, ci.nombre, p.nombre
ORDER BY total_compras DESC
```

e) Conocer el número de clientes por su procedencia (ciudad y provincia)

```
--e)Conocer el número de clientes por su procedencia (ciudad y provincia)

SELECT p.nombre AS Provincia, ci.nombre AS Ciudad, COUNT(*) AS Numero_Clientes
FROM CLIENTE c
INNER JOIN CIUDAD ci ON c.ciudadResidencia = ci.idCiudad
INNER JOIN PROVINCIA p ON ci.idProvincia = p.idProvincia
GROUP BY p.nombre, ci.nombre;
```

f) Conocer los productos que no se han vendido por cada mes del año 2023



--f)Conocer los productos que se han vendido por cada mes del año 2023

```
SELECT MONTH(oc.fecha) AS Mes, p.idProducto AS ID, p.nombre AS Nombre_Producto,
SUM(dc.cantidad) AS Cantidad_Vendida
FROM DETALLE_COMPRA dc
    INNER JOIN ORDEN_COMPRA oc ON dc.numOC = oc.numOC
    INNER JOIN PRODUCTO p ON dc.idProducto = p.idProducto
WHERE YEAR(oc.fecha) = 2023
GROUP BY MONTH(oc.fecha), p.idProducto, p.nombre
ORDER BY mes, p.idProducto;
```

g) Cantidad vendida por producto durante el mes de diciembre del año 2023

--g)Cantidad vendida por producto durante el mes de diciembre del año 2023

```
SELECT dc.idProducto as "ID Producto", p.nombre AS "Nombre Producto", SUM(dc.cantidad)
AS "Cantidad Vendida"
FROM DETALLE_COMPRA dc
    INNER JOIN ORDEN_COMPRA oc ON dc.numOC = oc.numOC
    INNER JOIN PRODUCTO p ON dc.idProducto = p.idProducto
WHERE YEAR(oc.fecha) = 2023 AND MONTH(oc.fecha) = 12
GROUP BY dc.idProducto, p.nombre;
```

h) Conocer el número de vendedores por su sexo

--h)Conocer el número de vendedores por su sexo

```
SELECT sexo AS Sexo, COUNT(*) AS Vendedores
FROM VENDEDOR
GROUP BY sexo;
```

i) Listado de los vendedores que ingresaron en el último trimestre del año 2022

--i)Listado de los vendedores que ingresaron en el último trimestre del año 2022

```
SELECT idVen AS ID,nombreVen AS Nombre,salario AS Salario,fechaIngreso AS "Fecha
Ingreso"
FROM VENDEDOR
WHERE fechaIngreso BETWEEN '2022-10-01' AND '2022-12-31';
```

j) Conocer la ciudad que con el mayor número de ordenes de compras se ha pedido

--j)Conocer la ciudad que con el mayor número de ordenes de compras se ha pedido

```
SELECT TOP 1 ci.idCiudad AS ID, ci.nombre AS Ciudad,COUNT(*) AS "Numeros de Compra"
FROM ORDEN_COMPRA oc
    INNER JOIN CLIENTE c ON oc.idCli = c.idCli
    INNER JOIN CIUDAD ci ON c.ciudadResidencia = ci.idCiudad
GROUP BY ci.idCiudad, ci.nombre
ORDER BY "Numeros de Compra" DESC;
```

k) Conocer el número de ordenes de compras de acuerdo con su estado

--k) Conocer el número de ordenes de compras de acuerdo con su estado

```
SELECT OC.idEstadoOC AS ID, EC.descripcion AS Estado, COUNT(EC.descripcion) AS "Cantidad de Compras"
FROM ORDEN_COMPRA AS OC
INNER JOIN ESTADO_COMPRA AS EC ON OC.idEstadoOC = EC.idEstadoOC
GROUP BY OC.idEstadoOC, descripcion;
```

l) Conocer las órdenes de compra pendientes de entrega

--l) Conocer las órdenes de compra pendientes de entrega

```
SELECT OC.numOC AS "# Orden", P.nombre AS "Nombre Producto", EC.descripcion AS Estado
FROM ORDEN_COMPRA AS OC
INNER JOIN ESTADO_COMPRA AS EC ON OC.idEstadoOC = EC.idEstadoOC
INNER JOIN DETALLE_COMPRA AS DC ON OC.numOC = DC.numOC
INNER JOIN PRODUCTO AS P ON DC.idProducto = P.idProducto
WHERE EC.descripcion = 'pendiente de entrega';
```

m) Conocer los vendedores que tienen órdenes de compra pedidas y que no se hizo el pago correspondiente

--m) Conocer los vendedores que tienen órdenes de compra pedidas y que no se hizo el pago correspondiente

```
SELECT DISTINCT v.idVen AS Nombre, v.nombreVen as Nombre
FROM VENDEDOR v
INNER JOIN ORDEN_COMPRA oc ON v.idVen = oc.idVen
INNER JOIN ESTADO_COMPRA ec ON oc.idEstadoOC = ec.idEstadoOC
WHERE ec.descripcion = 'pedido de compra';
-- Asumiendo que 'pedido de compra' representa el estado de la orden cuando no se ha realizado el pago
```

n) Conocer el mejor vendedor del año 2023. Se determina el mejor vendedor con base en el número de ventas realizadas

--n) Conocer el mejor vendedor del año 2023. Se determina el mejor vendedor con base en el número de ventas realizadas

```
SELECT TOP 1 v.idVen AS ID, v.nombreVen AS Nombre, COUNT(oc.numOC) AS num_ventas_realizadas
FROM VENDEDOR v
INNER JOIN ORDEN_COMPRA oc ON v.idVen = oc.idVen
WHERE YEAR(oc.fecha) = 2023
GROUP BY v.idVen, v.nombreVen
ORDER BY num_ventas_realizadas DESC
```

o) Conocer el promedio de venta mensual (por cada mes) del año 2022

--o) Conocer el promedio de venta mensual (por cada mes) del año 2022

```
SELECT MONTH(fecha) AS Mes, AVG(valorTotal) AS "Promedio Venta Mensual"
FROM ORDEN_COMPRA
WHERE YEAR(fecha) = 2022
GROUP BY MONTH(fecha)
ORDER BY MONTH(fecha);
```

- p) Conocer los vendedores que cuyo promedio de ventas del año 2023 no pasaron del promedio en ventas anuales.

```
--p)Conocer los vendedores que cuyo promedio de ventas del año 2023 no pasaron del
promedio en
--ventas anual.
SELECT v.idVen AS ID, v.nombreVen AS Nombre, AVG(oc.valorTotal) AS "Promedio Ventas
Anual 2023"
FROM VENDEDOR v
INNER JOIN ORDEN_COMPRA oc ON v.idVen = oc.idVen
WHERE YEAR(oc.fecha) = 2023
GROUP BY v.idVen, v.nombreVen
HAVING AVG(oc.valorTotal) <= (
                                SELECT AVG(valorTotal)
                                FROM ORDEN_COMPRA
                                WHERE YEAR(fecha) = 2023);
```

## 5. RESULTADOS

Tras haber llevado a cabo un exhaustivo programa de ejercicios y prácticas, se han obtenido resultados altamente satisfactorios y prometedores.

En primer lugar se ha observado una notable mejora en la destreza, precisión y eficiencia de los participantes al momento de realizar las diversas actividades propuestas los nuevos prototipos y herramientas desarrollados específicamente para este fin han demostrado ser altamente efectivos para optimizar el desempeño de los usuarios logrando incrementos de productividad de hasta un 20% en ciertos casos.

Asimismo, se han implementado diversas optimizaciones y mejoras en los procesos de ejecución de los ejercicios lo que ha permitido reducir significativamente los tiempos requeridos para completarlos esto se ha traducido en una mayor eficiencia y agilidad por parte de los involucrados con reducciones de tiempo de hasta un 30% en algunas tareas.

Por otra parte, la retroalimentación recibida de los participantes ha sido sumamente positiva estos han destacado la mayor funcionalidad, facilidad de uso e intuitiva de los recursos empleados durante las prácticas lo cual ha generado un aumento considerable del nivel de satisfacción y compromiso con las actividades, reflejado en índices de participación y asistencia por encima del 90%.

Finalmente, el éxito de estos ejercicios ha sentado las bases para poder ampliar el alcance y la escala de las iniciativas de capacitación y desarrollo de habilidades dentro de la organización esto abre la puerta a la posibilidad de replicar estos resultados en un contexto más amplio, beneficiando a un mayor número de colaboradores y aumentando el impacto en el desempeño general de la empresa.

## 6. CONCLUSIONES

En conclusión, el desarrollo del caso "ORDEN DE COMPRA" subraya la importancia del dominio del SQL para el diseño y manejo efectivo de bases de datos la creación precisa de tablas, la inserción coherente de datos y la ejecución de consultas SELECT han sido elementos clave para extraer información relevante y significativa en la toma de decisiones empresariales este proceso ha reforzado la necesidad de mantener la integridad y consistencia de los datos como pilares fundamentales en cualquier entorno de gestión de información.

Además, el trabajo colaborativo ha sido esencial para alcanzar los objetivos propuestos demostrando la importancia del trabajo en equipo en la resolución de desafíos complejos el manejo eficiente del SQL como una herramienta versátil para el análisis y manipulación de datos ha resaltado la importancia de una sólida formación en este ámbito para abordar con éxito las demandas de la empresa moderna.

## 7. RECOMENDACIONES

Basándonos en la práctica realizada y los resultados obtenidos en el caso "ORDEN DE COMPRA", se derivan las siguientes recomendaciones específicas para mejorar el manejo de bases de datos mediante consultas SQL:

**Optimización de Consultas:** Se sugiere revisar y optimizar regularmente las consultas SQL para mejorar la eficiencia y el rendimiento del sistema. Esto incluye la identificación de áreas de mejora como el uso de índices la reducción de la complejidad de las consultas y la optimización de los planes de ejecución.

**Validación de Datos:** Es crucial implementar mecanismos de validación de datos para garantizar la integridad y consistencia de la información almacenada en la base de datos esto implica establecer restricciones de integridad verificar la precisión de los datos ingresados y realizar auditorías periódicas para detectar posibles errores o inconsistencias.

**Documentación y Comentarios:** Es esencial documentar adecuadamente las consultas SQL y proporcionar comentarios claros y descriptivos en el código esto facilitará la comprensión y el mantenimiento del sistema para futuros desarrolladores y administradores de bases de datos.

**Monitoreo y Análisis de Desempeño:** Se aconseja implementar herramientas de monitoreo y análisis de desempeño para evaluar el rendimiento del sistema y detectar posibles cuellos de botella o puntos de mejor esto permitirá tomar medidas proactivas para optimizar el rendimiento y la disponibilidad de la base de datos.

Al seguir estas recomendaciones se podrá mejorar la calidad, seguridad y eficiencia en el manejo de bases de datos mediante consultas SQL lo que contribuirá al éxito y la competitividad de la organización en el ámbito empresarial.