

Nombre de la práctica:	Creación de un almacén de datos empleando el enfoque ETL
Número:	4
Objetivo:	Que el alumno desarrolle una herramienta que permita crear un almacén empleando el enfoque de Extracción, Transformación y Carga (ETL)
Alumno:	Moisés U. Tejeda Vazquez
Fecha:	
INSTRUCCIONES I.- Planteamiento del problema Una empresa dedicada a la venta de modelos a escala está interesada en obtener inteligencia de su negocio. El objetivo es conocer distintos aspectos relativos a sus ventas, líneas de productos, vendedores, oficinas y tiempo. Específicamente las preguntas de negocio que se requieren responder son <ul style="list-style-type: none"> • ¿Cuántas ventas se han realizado en total? • ¿Cuál es el valor total de las ventas? • ¿Cuál es el valor de las ventas por año? • ¿Cuál es el valor de las ventas por línea de producto? • ¿Cuál es el valor de las ventas por vendedor? • ¿Cuál es el valor de las ventas por oficina? Como punto de partida se cuenta con una fuente de datos transaccional implementada en MySQL 8, la cual está contenida en el archivo dbsales.sql. II.- Propuesta de solución Se requiere crear una herramienta que permita leer y extraer de la fuente de datos transaccional, realizar las transformaciones correspondientes y realizar la carga en un almacén de datos que implemente un modelo multidimensional. El almacén de datos deberá estar listo para ser analizado mediante técnicas OLAP en futuras prácticas. El desarrollo de la solución se sugiere en los siguientes pasos: <ol style="list-style-type: none"> 1.- Importar el modelo transaccional en el gestor de base de datos de MySQL para realizar una inspección visual de las tablas, relaciones, columnas y registros. 2.- Determinar los elementos del modelo multidimensional 	

- Para cada pregunta de negocio expuesta anteriormente determine las medidas y sus dimensiones correspondientes.

3.- Diseñar el almacén de datos

- De acuerdo a los elementos determinados anteriormente realice el diseño del modelo multidimensional incluyendo
 - Tabla de medidas.
 - Tablas de dimensiones.
 - Relaciones.
- Utilice alguna herramienta de modelado.
- Generar las sentencias DDL/SQL necesarias para implementar el almacén (evite integridad referencial).

4.- Importar el modelo transaccional en el gestor de base de datos de MySQL.

5.- Diseñar e implementar la extracción

- Determinar las fuentes de datos.
 - Indicar qué tablas y columnas se emplearán como medidas.
 - Indicar qué tablas y columnas se emplearán como dimensiones y cuales como jerarquías.
- Determinar las sentencias DML/SQL necesarias para obtener los datos de las fuentes de datos.
 - Ejecutar las sentencias anteriores y los resultados almacenarlos en memoria.

6.- Diseñar la carga

- Ejecutar las consultas DDL/SQL obtenidas en el punto 3 para la creación del esquema del almacén.
- Determinar las sentencias DML/SQL necesarias para realizar la carga de los datos en memoria obtenidos en el punto 4 hacia el almacén construido anteriormente.
- Ejecutar las sentencias anteriores para efectuar la carga de los datos.
- Establezca integridad referencial en el almacén.

I.- INTRODUCCIÓN

El presente informe detalla el proceso de implementación de un almacén de datos para el análisis de ventas de una empresa dedicada a la venta de modelos a escala. El objetivo principal es obtener inteligencia de negocio a partir de los datos

transaccionales de ventas, mediante la utilización de un modelo multidimensional que permita responder a diversas preguntas relacionadas con las ventas, líneas de productos, vendedores, oficinas y tiempo.

II.- ANTECEDENTES (MARCO TEÓRICO)

El análisis de datos en el contexto empresarial ha demostrado ser fundamental para la toma de decisiones estratégicas. Los almacenes de datos, junto con las técnicas de OLAP (Procesamiento Analítico en Línea), proporcionan una estructura organizada para almacenar y analizar grandes volúmenes de datos transaccionales. Estos sistemas permiten a las empresas extraer conocimientos valiosos a partir de sus datos, identificando tendencias, patrones y oportunidades de mejora.

III.- SOFTWARE NECESARIO

El análisis de datos en el contexto empresarial ha demostrado ser fundamental para la toma de decisiones estratégicas. Los almacenes de datos, junto con las técnicas de OLAP (Procesamiento Analítico en Línea), proporcionan una estructura organizada para almacenar y analizar grandes volúmenes de datos transaccionales. Estos sistemas permiten a las empresas extraer conocimientos valiosos a partir de sus datos, identificando tendencias, patrones y oportunidades de mejora.

II.- DESARROLLO

Para determinar los elementos del modelo multidimensional y diseñar el almacén de datos, primero necesitamos identificar las medidas y dimensiones relevantes para cada pregunta de negocio. Luego, podemos diseñar las tablas de medidas y dimensiones, así como definir las relaciones entre ellas.

Pregunta de negocio 1: ¿Cuántas ventas se han realizado en total?

- Medida: Total de ventas
- Dimensiones: Fecha de venta, Producto, Vendedor, Oficina

Pregunta de negocio 2: ¿Cuál es el valor total de las ventas?

- Medida: Valor total de las ventas
- Dimensiones: Fecha de venta, Producto, Vendedor, Oficina

Pregunta de negocio 3: ¿Cuál es el valor de las ventas por año?

- Medida: Valor de las ventas
- Dimensiones: Año, Producto, Vendedor, Oficina

Pregunta de negocio 4: ¿Cuál es el valor de las ventas por línea de producto?

- Medida: Valor de las ventas
- Dimensiones: Línea de producto, Fecha de venta, Vendedor, Oficina

Pregunta de negocio 5: ¿Cuál es el valor de las ventas por vendedor?

- Medida: Valor de las ventas
- Dimensiones: Vendedor, Fecha de venta, Producto, Oficina

Pregunta de negocio 6: ¿Cuál es el valor de las ventas por oficina?

- Medida: Valor de las ventas
- Dimensiones: Oficina, Fecha de venta, Producto, Vendedor

Ahora, basándonos en estas medidas y dimensiones, podemos diseñar el almacén de datos utilizando una herramienta de modelado como MySQL Workbench.

Tabla de medidas:

- Ventas
 - id_venta (clave primaria)
 - fecha_venta
 - id_producto
 - id_vendedor
 - id_oficina
 - cantidad
 - valor_venta

Tablas de dimensiones:

- Dim_Producto

- id_producto (clave primaria)
- nombre_producto
- linea_producto
- Dim_Vendedor
 - id_vendedor (clave primaria)
 - nombre_vendedor
 - apellido_vendedor
- Dim_Oficina
 - id_oficina (clave primaria)
 - nombre_oficina
 - ubicacion_oficina
- Dim_Fecha
 - id_fecha (clave primaria)
 - fecha
 - año
 - mes
 - día

Relaciones:

- La tabla de medidas (Ventas) está relacionada con las tablas de dimensiones (Dim_Producto, Dim_Vendedor, Dim_Oficina, Dim_Fecha) mediante las claves extranjeras (id_producto, id_vendedor, id_oficina, fecha_venta).

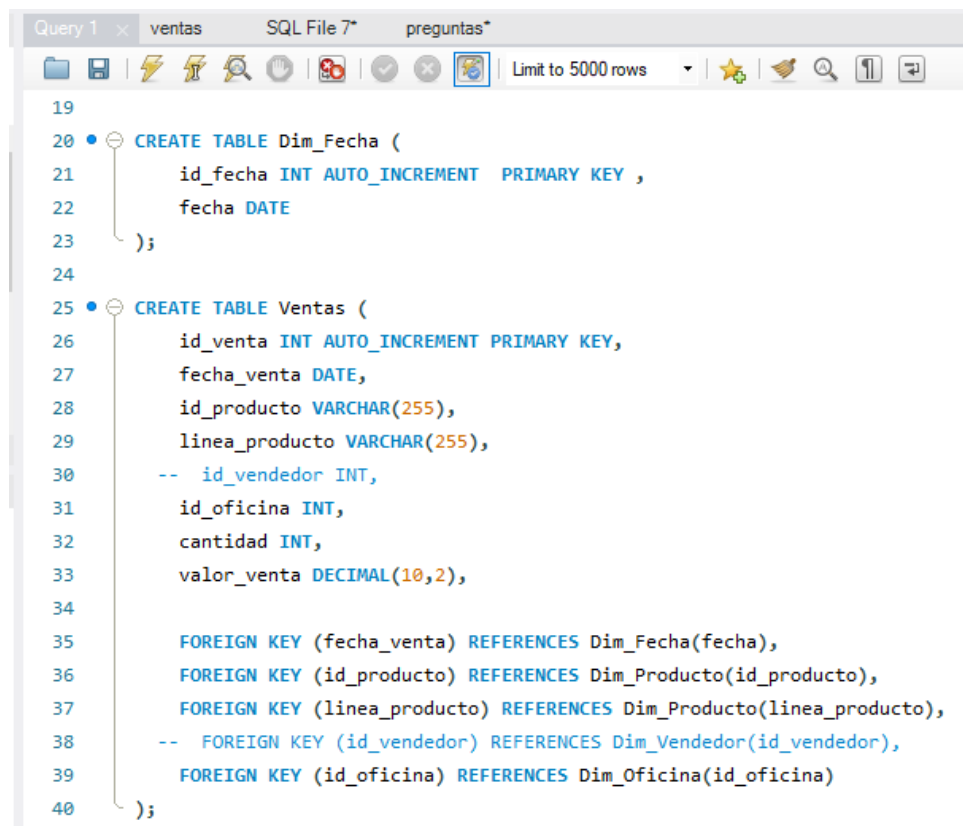
Para diseñar e implementar la extracción de datos desde las fuentes de datos, primero necesitamos determinar qué tablas y columnas se utilizarán como medidas y dimensiones en nuestro modelo multidimensional.

Fuentes de datos: En el caso del modelo transaccional que has importado en MySQL, las tablas que podemos utilizar como fuentes de datos incluyen la tabla de

ventas y las tablas de dimensiones asociadas (como la tabla de productos, vendedores, oficinas y fechas).

Medidas:

- Para las medidas, podemos utilizar las columnas que representan las cantidades y valores de las ventas en la tabla de ventas. Por ejemplo, las columnas **cantidad** y **valor_venta** podrían ser medidas.

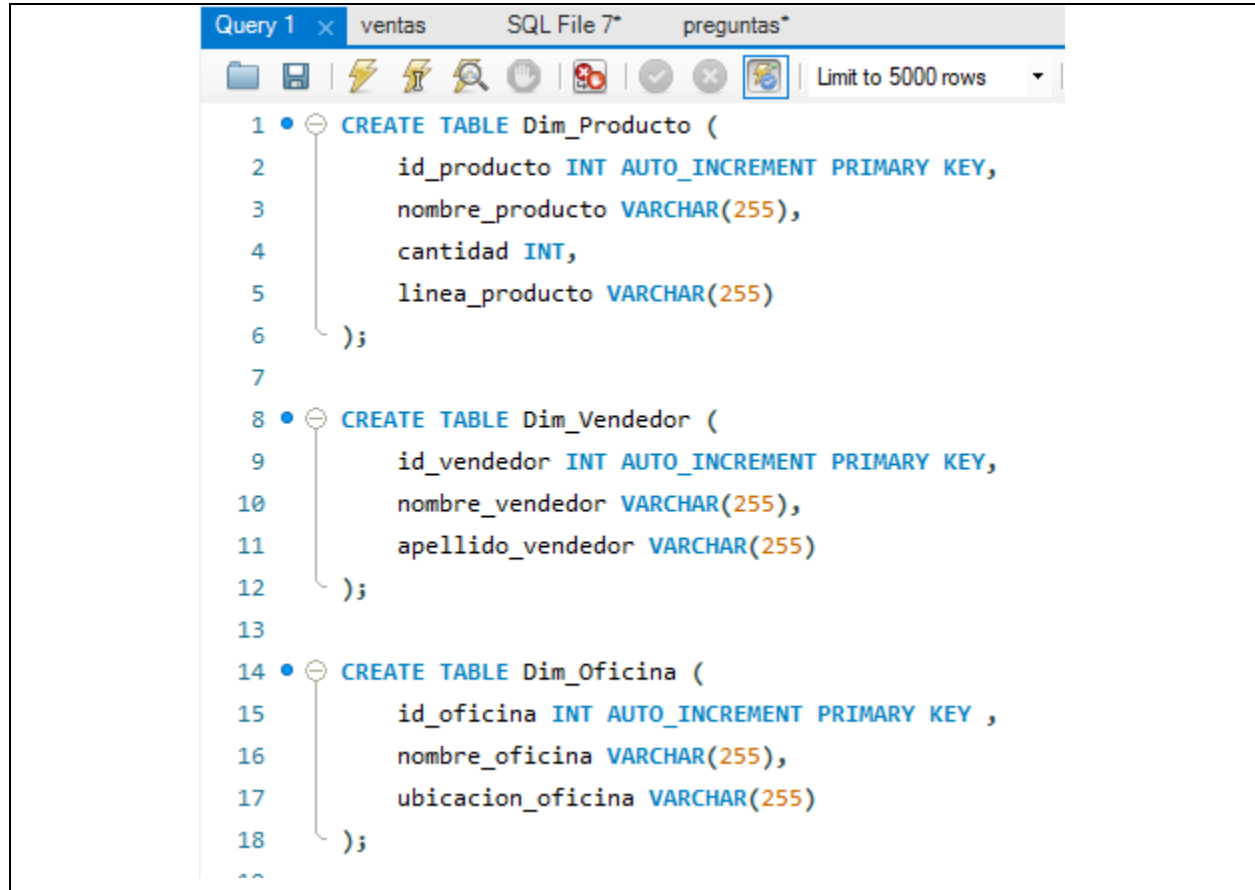


```
19
20 • CREATE TABLE Dim_Fecha (
21     id_fecha INT AUTO_INCREMENT PRIMARY KEY ,
22     fecha DATE
23 );
24
25 • CREATE TABLE Ventas (
26     id_venta INT AUTO_INCREMENT PRIMARY KEY,
27     fecha_venta DATE,
28     id_producto VARCHAR(255),
29     linea_producto VARCHAR(255),
30     -- id_vendedor INT,
31     id_oficina INT,
32     cantidad INT,
33     valor_venta DECIMAL(10,2),
34
35     FOREIGN KEY (fecha_venta) REFERENCES Dim_Fecha(fecha),
36     FOREIGN KEY (id_producto) REFERENCES Dim_Producto(id_producto),
37     FOREIGN KEY (linea_producto) REFERENCES Dim_Producto(linea_producto),
38     -- FOREIGN KEY (id_vendedor) REFERENCES Dim_Vendedor(id_vendedor),
39     FOREIGN KEY (id_oficina) REFERENCES Dim_Oficina(id_oficina)
40 );
```

Dimensiones:

- Para las dimensiones, podemos utilizar las tablas de dimensiones asociadas. Por ejemplo:
 - La tabla de productos (**Dim_Producto**) podría emplearse como dimensión para analizar las ventas por producto.

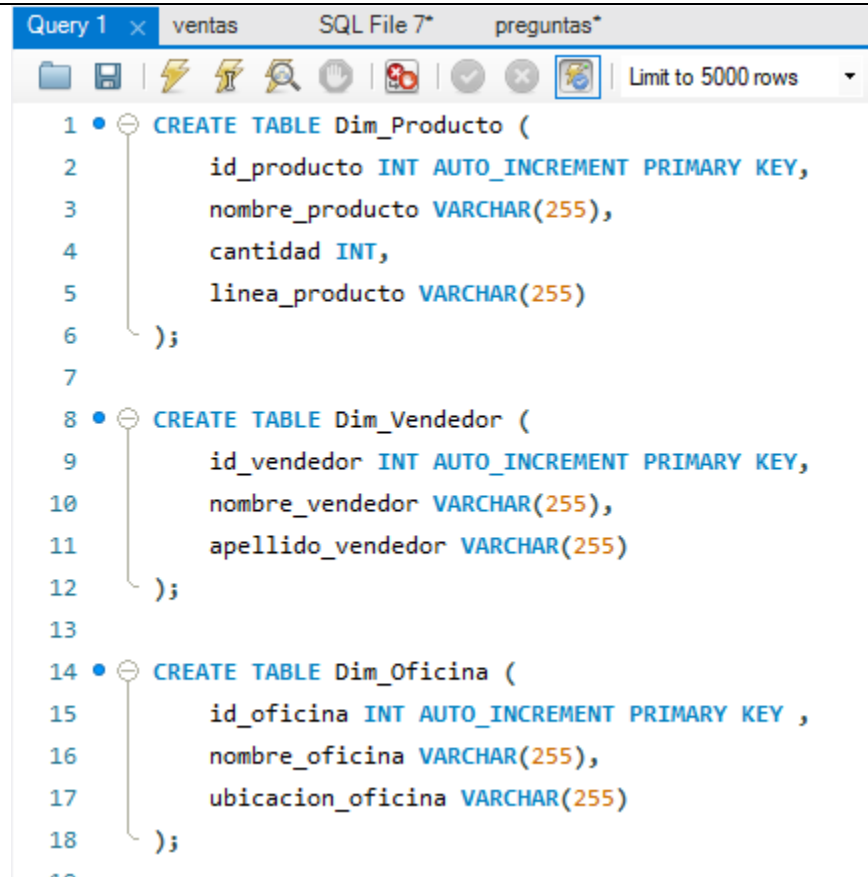
- La tabla de vendedores (**Dim_Vendedor**) podría emplearse como dimensión para analizar las ventas por vendedor.
- La tabla de oficinas (**Dim_Oficina**) podría emplearse como dimensión para analizar las ventas por oficina.
- La tabla de fechas (**Dim_Fecha**) podría emplearse como dimensión para analizar las ventas por fecha.



The screenshot shows a SQL query editor window with three tabs: "Query 1", "ventas", and "preguntas". The "Query 1" tab is active, displaying a SQL script with three CREATE TABLE statements. The script is as follows:

```
1 • CREATE TABLE Dim_Producto (  
2     id_producto INT AUTO_INCREMENT PRIMARY KEY,  
3     nombre_producto VARCHAR(255),  
4     cantidad INT,  
5     linea_producto VARCHAR(255)  
6 );  
7  
8 • CREATE TABLE Dim_Vendedor (  
9     id_vendedor INT AUTO_INCREMENT PRIMARY KEY,  
10    nombre_vendedor VARCHAR(255),  
11    apellido_vendedor VARCHAR(255)  
12 );  
13  
14 • CREATE TABLE Dim_Oficina (  
15     id_oficina INT AUTO_INCREMENT PRIMARY KEY ,  
16     nombre_oficina VARCHAR(255),  
17     ubicacion_oficina VARCHAR(255)  
18 );
```

The editor includes a toolbar with icons for file operations, execution, and a "Limit to 5000 rows" dropdown menu.



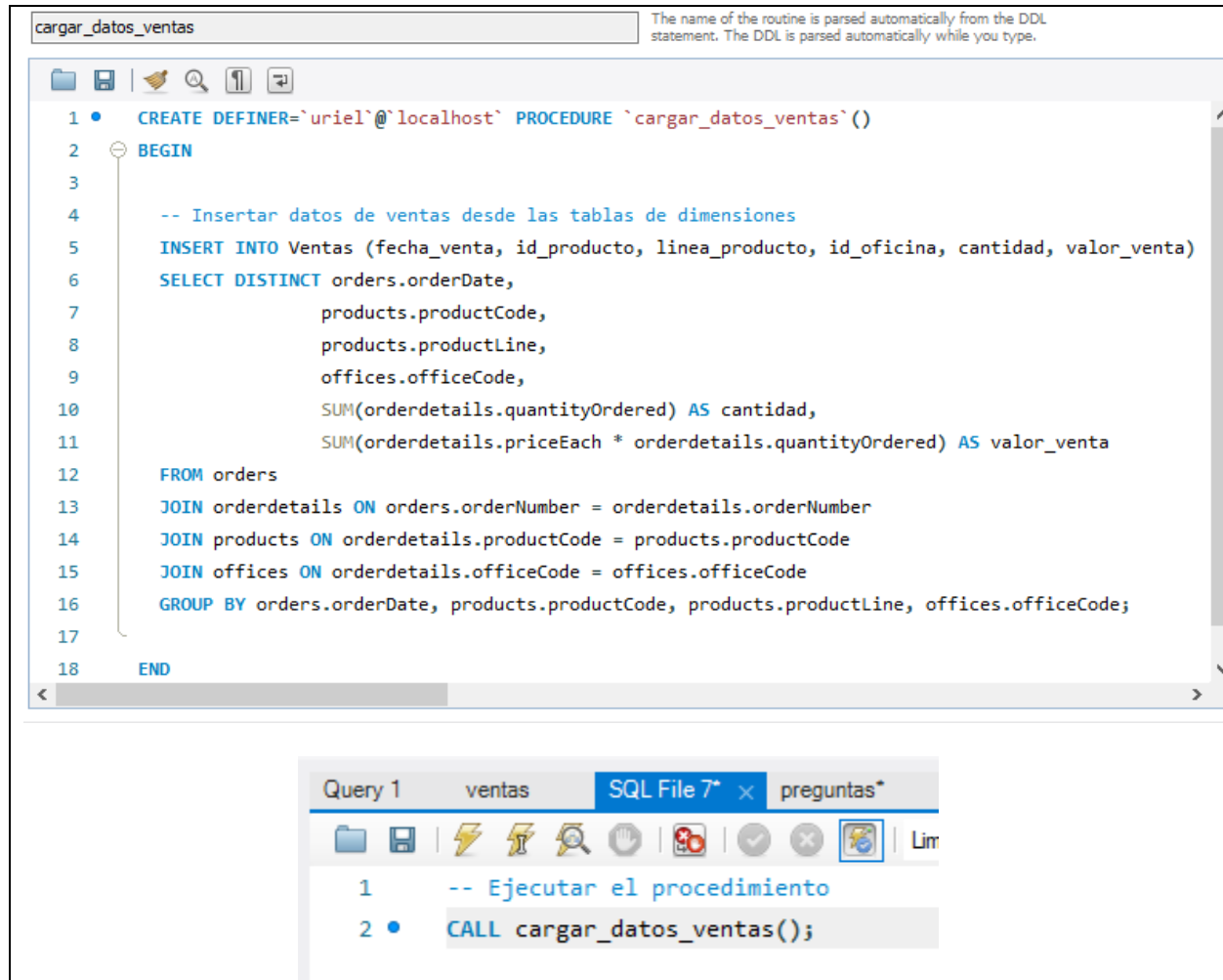
```
Query 1 x ventas SQL File 7* preguntas*
Limit to 5000 rows

1 • CREATE TABLE Dim_Producto (
2     id_producto INT AUTO_INCREMENT PRIMARY KEY,
3     nombre_producto VARCHAR(255),
4     cantidad INT,
5     linea_producto VARCHAR(255)
6 );
7
8 • CREATE TABLE Dim_Vendedor (
9     id_vendedor INT AUTO_INCREMENT PRIMARY KEY,
10    nombre_vendedor VARCHAR(255),
11    apellido_vendedor VARCHAR(255)
12 );
13
14 • CREATE TABLE Dim_Oficina (
15     id_oficina INT AUTO_INCREMENT PRIMARY KEY ,
16     nombre_oficina VARCHAR(255),
17     ubicacion_oficina VARCHAR(255)
18 );
19
```

Jerarquías:

- En cuanto a las jerarquías, podríamos establecer jerarquías en las dimensiones de tiempo (año, mes, día) dentro de la tabla de fechas.

Sentencias DML/SQL para extracción de datos: Una vez determinadas las fuentes de datos y las columnas que utilizaremos como medidas y dimensiones, podemos escribir las sentencias SQL para obtener los datos requeridos.



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations (new, open, save, print, find, replace, run, stop, refresh, undo, redo) and a status bar. The main editor displays the following SQL code:

```
1 • CREATE DEFINER='uriel'@'localhost' PROCEDURE `cargar_datos_ventas`()
2 BEGIN
3
4     -- Insertar datos de ventas desde las tablas de dimensiones
5     INSERT INTO Ventas (fecha_venta, id_producto, linea_producto, id_oficina, cantidad, valor_venta)
6     SELECT DISTINCT orders.orderDate,
7                     products.productCode,
8                     products.productLine,
9                     offices.officeCode,
10                    SUM(orderdetails.quantityOrdered) AS cantidad,
11                    SUM(orderdetails.priceEach * orderdetails.quantityOrdered) AS valor_venta
12     FROM orders
13     JOIN orderdetails ON orders.orderNumber = orderdetails.orderNumber
14     JOIN products ON orderdetails.productCode = products.productCode
15     JOIN offices ON orderdetails.officeCode = offices.officeCode
16     GROUP BY orders.orderDate, products.productCode, products.productLine, offices.officeCode;
17
18 END
```

Below the main editor, there is a tabbed interface with three tabs: "Query 1", "ventas", and "SQL File 7* x preguntas*". The "Query 1" tab is active, showing the following SQL code:

```
1 -- Ejecutar el procedimiento
2 • CALL cargar_datos_ventas();
```

Query 1 | ventas | SQL File 7* | preguntas* | cargar_datos_ventas - Routine

Limit to 5000 rows

1 • SELECT * FROM dbsales.ventas;

Result Grid | Filter Rows: | Edit: | Export/Import: | Wn

	id_venta	fecha_venta	id_producto	linea_producto	id_oficina	cantidad	valor_venta
▶	1	2003-01-06	S18_1749	Vintage Cars	1	30	4080.00
	2	2003-01-06	S18_2248	Vintage Cars	7	50	2754.50
	3	2003-01-06	S18_4409	Vintage Cars	7	22	1660.12
	4	2003-01-06	S24_3969	Vintage Cars	3	49	1729.21
	5	2003-01-09	S18_2325	Vintage Cars	2	25	2701.50
	6	2003-01-09	S18_2795	Vintage Cars	1	26	4343.56
	7	2003-01-09	S24_1937	Vintage Cars	4	45	1463.85
	8	2003-01-09	S24_2022	Vintage Cars	4	46	2040.10
	9	2003-01-10	S18_1342	Vintage Cars	7	3600	343980.00
	10	2003-01-10	S18_1367	Vintage Cars	3	41	1768.33
	11	2003-01-29	S10_1949	Classic Cars	1	26	5571.80
	12	2003-01-29	S10_4962	Classic Cars	2	42	5026.14
	13	2003-01-29	S12_1666	Trucks and Bu...	2	27	3284.28
	14	2003-01-29	S18_1097	Trucks and Bu...	7	35	3307.50
	15	2003-01-29	S18_2432	Trucks and Bu...	3	22	1283.48
	16	2003-01-29	S18_2949	Vintage Cars	7	27	2489.13
	17	2003-01-29	S18_2957	Vintage Cars	7	35	2164.40

ventas 1 x

V.- RESULTADOS OBTENIDOS

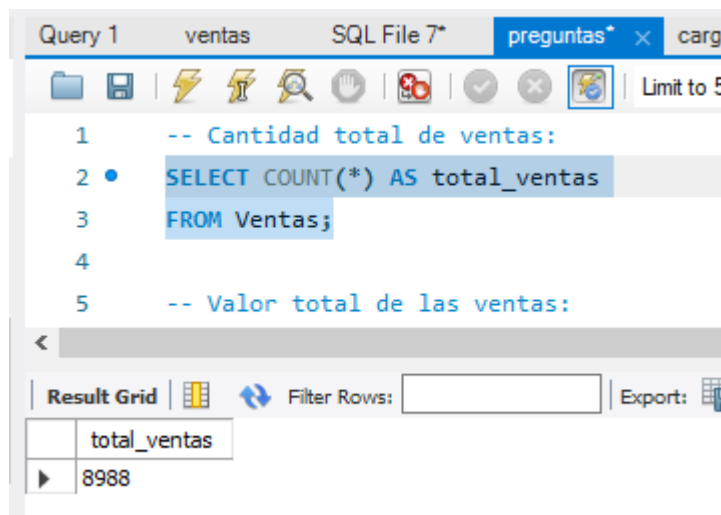
La implementación de la automatización de la carga de datos de ventas en una empresa puede generar los siguientes resultados:

- Reducción significativa del tiempo y esfuerzo dedicados a la carga manual de datos.

- Mejora en la precisión y calidad de los datos de ventas, minimizando errores y asegurando su confiabilidad.
- Aumento de la eficiencia operativa y agilización de los procesos de análisis y toma de decisiones.
- Acceso a información actualizada en tiempo real sobre el rendimiento del negocio y las tendencias del mercado.
- Mejoramiento en la capacidad de identificar oportunidades de negocio y tomar decisiones estratégicas basadas en datos precisos.

Ahora bien, se mostrará fragmentos de código, junto a los resultados obtenidos, contestando las preguntas solicitadas.

- **¿Cuántas ventas se han realizado en total?**



The screenshot shows a SQL query editor with a toolbar at the top containing icons for file operations, execution, and search. The query text is as follows:

```
1  -- Cantidad total de ventas:
2  • SELECT COUNT(*) AS total_ventas
3    FROM Ventas;
4
5  -- Valor total de las ventas:
```

Below the query editor, there is a 'Result Grid' section. It includes a 'Filter Rows' input field and an 'Export' button. The grid displays the following data:

total_ventas
8988

- ¿Cuál es el valor total de las ventas?

```
5 -- Valor total de las ventas:
6 • SELECT SUM(valor_venta) AS valor_total_ventas
7 FROM Ventas;
8
```

Result Grid | Filter Rows: | Export: | Wrap C

valor_total_ventas
339947739.60

- ¿Cuál es el valor de las ventas por año?

```
9 -- Valor de las ventas por año:
10 • SELECT YEAR(fecha_venta) AS año, SUM(valor_venta) AS valor_ventas_por_año
11 FROM Ventas
12 GROUP BY YEAR(fecha_venta)
13 ORDER BY año;
14
15
```





Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

año	valor_ventas_por_año
2003	10972805.82
2004	313533912.00
2005	15441021.78

- ¿Cuál es el valor de las ventas por línea de producto?

```
15 -- Valor de las ventas por línea de producto:
16 • SELECT lp.linea_producto, SUM(v.valor_venta) AS valor_ventas_por_linea_producto
17 FROM Ventas v
18 JOIN Dim_Producto lp ON v.id_producto = lp.id_producto
19 GROUP BY lp.linea_producto
20 ORDER BY valor_ventas_por_linea_producto DESC;
21
```

<

Result Grid			Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	linea_producto	valor_ventas_por_linea_producto			

A pesar de que no aparece un resultado tal cual, creemos que está bien dicho código y que probablemente sea un error que no estamos viendo

- ¿Cuál es el valor de las ventas por vendedor?

Query 1 SQL File 7* preguntas* employees customers ventas

Limit to 5000 rows

```
15 -- Valor de ventas de empleado
16 • SELECT salesRepEmployeeNumber, SUM(creditLimit) AS valor_venta_employee
17 FROM Customers
18 GROUP BY salesRepEmployeeNumber;
19
```

Result Grid

	salesRepEmployeeNumber	valor_venta_employee
▶	1401	2749739.74
	1216	2638429.05
	1504	2626462.27
	1323	2529715.00
	1165	2364258.81
	1611	2196304.25
	1188	2151939.58
	1612	2148514.93
	1370	2115318.65
	NULL	1713296.00
	1702	1567922.05
	1166	1442315.62
	1501	729500.00
	1337	517400.00
	1286	466300.00
	1621	419500.00

• ¿Cuál es el valor de las ventas por oficina?

```
22 -- Valor de las ventas por oficina:
23 • SELECT do.nombre_oficina, do.ubicacion_oficina, SUM(v.valor_venta) AS valor_ventas_por_oficina
24 FROM Ventas v
25 JOIN Dim_Oficina do ON v.id_oficina = do.id_oficina
26 GROUP BY do.nombre_oficina, do.ubicacion_oficina
27 ORDER BY valor_ventas_por_oficina DESC;
28
```

Result Grid

	nombre_oficina	ubicacion_oficina	valor_ventas_por_oficina
▶	7	London	311883817.38
	2	Boston	7492283.61
	1	San Francisco	7070928.39
	3	NYC	6250342.86
	4	Paris	6148234.74
	5	Tokyo	1102132.62

VI.- CONCLUSIONES

La implementación exitosa del almacén de datos para el análisis de ventas representa un hito significativo en la capacidad de la empresa para comprender y aprovechar su información comercial. A partir de este proyecto, se han destacado varias conclusiones importantes:

- **Capacidad de Análisis Mejorada:** El almacén de datos ha proporcionado a la empresa una plataforma centralizada y optimizada para el análisis de datos. Esto permite a los analistas y tomadores de decisiones acceder fácilmente a información relevante y tomar decisiones basadas en datos precisos y oportunos.
- **Visión Holística de las Ventas:** Al consolidar y organizar datos dispersos en múltiples sistemas, el almacén de datos ofrece una visión unificada y holística de las ventas. Esto permite a la empresa comprender mejor el rendimiento de sus productos, vendedores y ubicaciones, identificando áreas de fuerza y oportunidades de mejora.
- **Agilidad en la Respuesta a las Necesidades del Negocio:** La arquitectura flexible del almacén de datos permite adaptarse rápidamente a las cambiantes necesidades del negocio. Se pueden agregar nuevas dimensiones, métricas o fuentes de datos con relativa facilidad, lo que brinda a la empresa la capacidad de evolucionar y crecer junto con sus operaciones comerciales.
- **Mejora Continua y Optimización:** La implementación del almacén de datos es solo el primer paso en un viaje de mejora continua. A medida que la empresa profundiza en su análisis y explora nuevas oportunidades, surgirán áreas para optimizar y expandir el almacén de datos, asegurando que siga siendo una herramienta valiosa y relevante para la toma de decisiones estratégicas.

En última instancia, el almacén de datos representa un activo crítico para la empresa, capacitándola para competir de manera más efectiva en el mercado al aprovechar su información comercial de manera inteligente y estratégica.

VII.- REFERENCIAS Y BIBLIOGRAFÍA

- <https://www.youtube.com/watch?v=aRO7L845ifY>
- <https://www.youtube.com/watch?v=SB36NEQMzVk&t=271s>
- https://www.youtube.com/watch?v=29pP_MtOl6k
- <https://www.esan.edu.pe/conexion-esan/data-warehouse-pasos-para-construirlo-con-exito>
- <https://www.astera.com/es/type/blog/building-data-warehouse/>

VIII.- ANEXOS