

EVIDENCIA DE APRENDIZAJE 3
TRANSFORMACIÓN DE DATOS Y CARGA EN EL DATA MART FINAL

DAVID ALEJANDRO AGUDELO MENESES
FERNEY DE JESUS ECHEVERRI ECHEVERRI
ESTEBAN URRUTIA BERMUDEZ

Asesor

ANTONIO JESÚS VALDERRAMA JARAMILLO

Institución Universitaria Digital de Antioquia

Facultad de Ingenierías y Ciencias

Ingeniería Software y Datos

Grupo 2

2025

Resumen

La gestión de datos en los diferentes entornos de data Warehouse requiere de procesos intermedios que garanticen la calidad, consistencia y organización de la información antes de su análisis. En este trabajo se aborda la construcción de una base de datos Staging a partir de la base de datos Jardinería, con el fin de estructurar, depurar y trasladar datos relevantes que permitan soportar futuros procesos analíticos.

El desarrollo de esta actividad incluye un análisis de la información disponible, la creación de consultas SQL para migrar los registros y la validación de la integridad de los datos.

De igual manera se busca desarrollar competencias en el diseño, construcción y documentación de soluciones de bases de datos bajo un enfoque práctico y analítico.

Tabla de Contenido

Introducción.....	4
Justificación.....	5
Objetivos.....	6
Planteamiento del problema.....	7
Propuesta de la solución.....	9
Modelo estrella actualizado.....	10
Análisis realizado a los datos Jardinería e implementación de ETL.....	11
Validación de la extracción de datos de Jardinería a la Base de Datos Staging.....	11
Conclusiones.....	18
Anexos.....	19
Referencias Bibliográficas.....	20

Introducción

En la actualidad, los datos representan un recurso estratégico indispensable para las organizaciones, ya que de ellos depende en gran medida la calidad de la toma de decisiones. Sin embargo, antes de que esta información pueda ser utilizada en sistemas de inteligencia de negocios o en procesos de análisis avanzado, es necesario que atraviese un conjunto de etapas que garanticen su integridad, consistencia y confiabilidad. Este proceso se conoce como ETL (Extract, Transform, Load).

El presente documento tiene como propósito aplicar un proceso ETL al modelo relacional Jardinería, integrando los conocimientos adquiridos en diseño de bases de datos, normalización y construcción de consultas SQL. De esta manera, se busca no solo reforzar la teoría aprendida, sino también llevarla a un ejercicio práctico que sirva como preparación para los desafíos reales que plantea la gestión y el aprovechamiento de los datos en entornos profesionales.

Justificación

La implementación de procesos ETL (Extract, Transform, Load) resulta fundamental en la gestión de datos, ya que permiten consolidar información proveniente de diferentes fuentes, garantizar su calidad y prepararla para posteriores análisis. Dentro de este proceso, la creación de una base de datos Staging constituye una etapa clave, pues actúa como un espacio intermedio donde los datos son almacenados temporalmente para ser validados, transformados y depurados antes de su carga en el sistema de destino.

En entornos reales, la información suele encontrarse dispersa, incompleta o con inconsistencias que de no ser tratadas adecuadamente pueden comprometer la fiabilidad de los resultados en herramientas de inteligencia de negocios o en procesos analíticos. Por ello los ETL proporcionan un marco estructurado que no solo mejora la calidad de los datos, sino que también optimiza su integración y facilita la toma de decisiones estratégicas basadas en información confiable.

Objetivos

Objetivo General

Diseñar y ejecutar un proceso ETL (Extract, Transform, Load) que permita extraer los datos desde la base de datos origen, transformarlos en la base de datos de Staging y cargarlos en el Data Mart final, asegurando su calidad e integridad para apoyar los procesos de análisis y toma de decisiones.

Objetivos Específicos

Aplicar consultas SQL que faciliten la extracción y validación de los datos desde la base de datos origen.

Implementar transformaciones mediante SQL que permitan depurar, normalizar y organizar la información en el entorno de Staging.

Realizar la carga estructurada y precisa de los datos hacia el Data Mart, garantizando su disponibilidad para procesos analíticos.

Demostrar cómo los procesos ETL fortalecen la confiabilidad de la información en contextos de inteligencia de negocios.

Planteamiento del problema.

La empresa Jardinería administra sus operaciones comerciales mediante una base de datos transaccional que registra información relacionada con clientes, pedidos, pagos, productos, oficinas y empleados. Si bien este sistema es eficiente para la gestión operativa diaria, no está diseñado ni optimizado para responder preguntas estratégicas ni para soportar análisis empresariales de mayor complejidad.

El desafío se presenta cuando la organización necesita obtener indicadores clave de negocio, tales como las ventas mensuales, los clientes más rentables, los productos con mayor demanda o el nivel de cobranza. Consultar directamente la base de datos transaccional implica ejecutar procesos costosos y lentos que pueden afectar el rendimiento del sistema operativo, además de generar consultas difíciles de mantener en el tiempo.

Ante esta situación, se hace indispensable implementar un proceso ETL (Extract, Transform, Load) que permita extraer los datos desde la base de origen, transformarlos en un entorno Staging (jardineria_stg) para su depuración, organización y validación, y finalmente cargarlos en un Data Mart optimizado para el análisis de información.

De esta manera, el Data Mart final proporcionará a la empresa Jardinería una estructura confiable y eficiente para realizar reportes, generar indicadores y apoyar procesos de inteligencia de negocios, garantizando que la información utilizada sea consistente y útil para la toma de decisiones estratégicas.

Análisis del problema

Limitaciones de la base transaccional (Jardinería)

- Está diseñada para registrar operaciones del día a día (altas, bajas, modificaciones) y no para responder consultas analíticas.
- Su estructura altamente normalizada dificulta la construcción de reportes integrales que combinen información de múltiples tablas.
- Consultas frecuentes sobre ventas, clientes o productos pueden degradar el rendimiento del sistema operativo.

Necesidad de un proceso ETL hacia un Data Mart final

- Extraer los datos relevantes (clientes, productos, pedidos, pagos, empleados, oficinas) de la base transaccional.
- Transformar la información aplicando reglas de limpieza, integración y consolidación que garanticen su calidad.
- Cargar los datos en un Data Mart diseñado bajo un esquema optimizado para el análisis (estrella o copo de nieve).

Objetivos que resuelve el Data Mart final

- Velocidad: agilizar los tiempos de respuesta en las consultas analíticas.
- Calidad de datos: asegurar consistencia, detectar registros incompletos o inválidos.
- Seguridad: evitar que la ejecución de reportes afecte el rendimiento de la base transaccional.
- Escalabilidad: servir como base para ampliar el modelo analítico y dar soporte a futuras necesidades de inteligencia de negocios.

Propuesta de la solución

Se busca desarrollar un proceso de transformación y carga de datos, de forma tal que logre asegurar que dichos datos estén disponibles y preparados para su análisis.

Lo anterior implica tomar los datos de la base de datos de origen, pasar un un Staging y finalmente un Data Mart, el cual será objeto de las respectivas consultas y su posterior análisis.

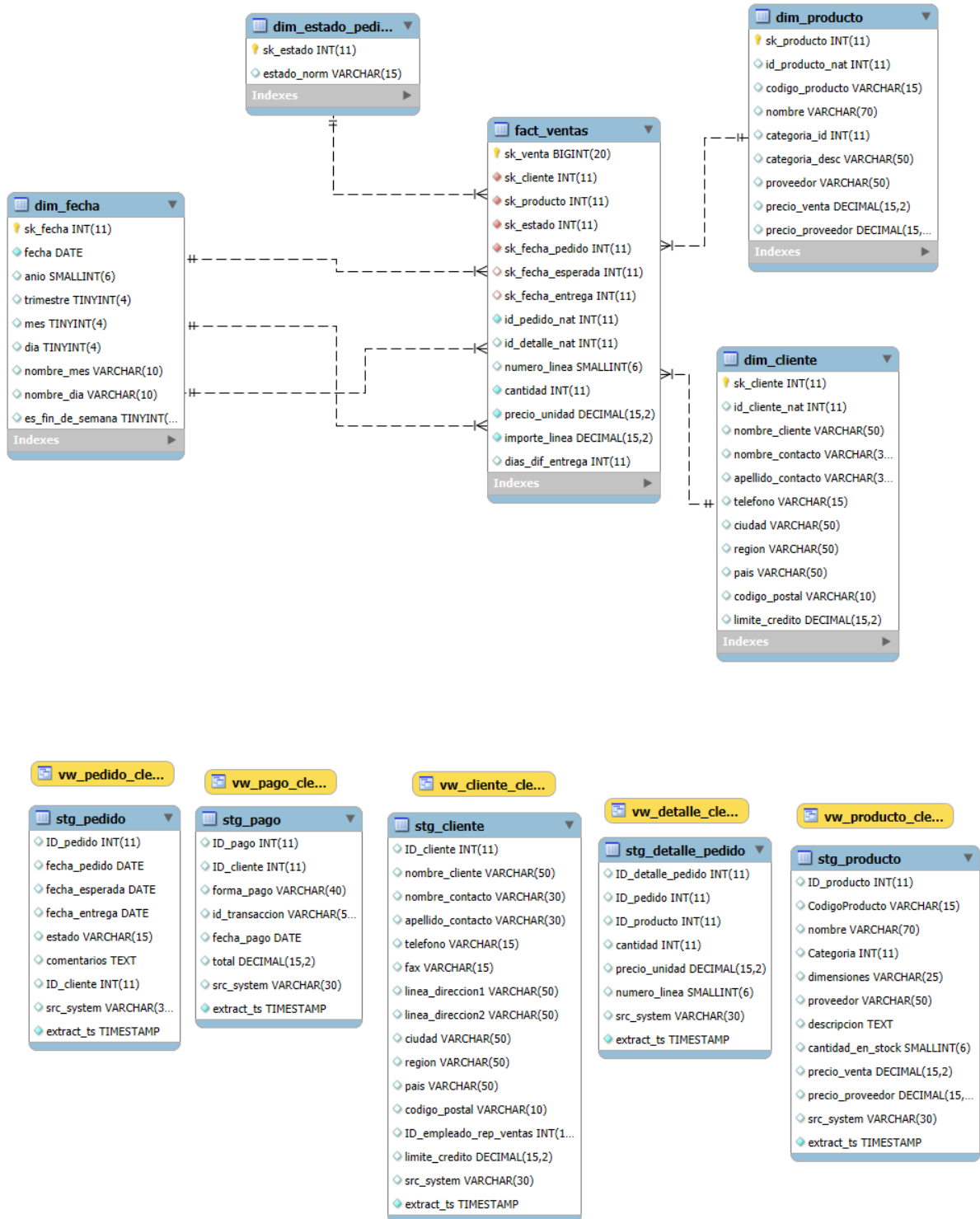
La base de datos jardineria_stg se diseña como un conjunto de tablas espejo de la base de datos Jardinería, que contienen únicamente los datos relevantes para análisis. Estas tablas mantienen la estructura básica del origen, pero se complementan con columnas de auditoría:

stg_batch_id: Identificador del lote de carga.

stg_load_ts: Fecha y hora en que se insertaron los datos.

stg_source: Origen de los datos (ejemplo: “Jardinería”).

Modelo estrella actualizado



Análisis realizado a los datos Jardinería e implementación de ETL.

Se implementó un proceso ETL para separar la operación transaccional (OLTP) de la analítica. Los datos se trasladaron a un entorno de Staging (jardineria_stg) con control de lotes y metadatos, se normalizaron mediante Vistas Clean y se proyectaron a un Data Mart dimensional orientado a reportes. Esto habilita consultas de negocio consistentes sin afectar el rendimiento del sistema operativo.

La implementación de las vistas de limpieza ayuda en la separación entre operación y analítica mediante Staging y Vistas Clean permite trasladar datos de forma controlada, auditable y confiable. El Data Mart resultante habilita indicadores y reportes de negocio sin penalizar el Procesamiento de Transacciones en Línea (OLTP), y sienta las bases para iniciativas de inteligencia de negocios.

Validación de la extracción de datos de Jardinería a la Base de Datos Staging.

Imagen 1. Consulta de la vista de limpieza vw_detalle_clean uniendo con vw_pedido_clean en la base de datos jardineria_stg filtrando todos los datos de las tiendas.

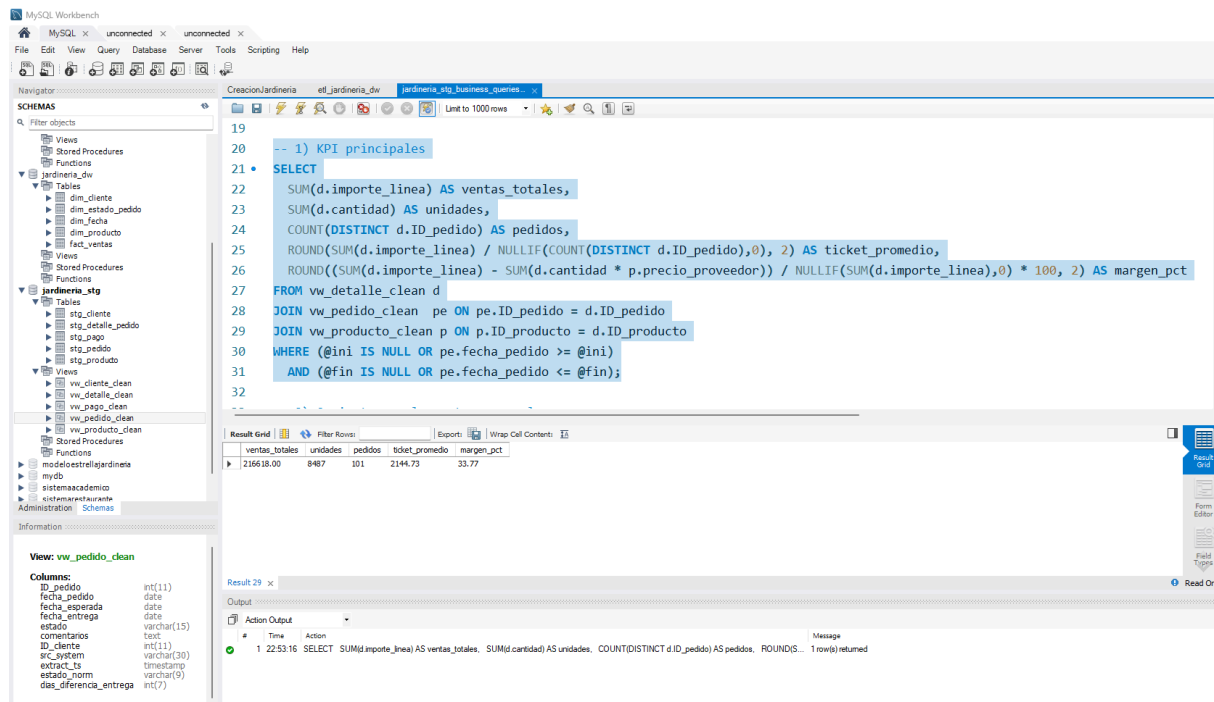
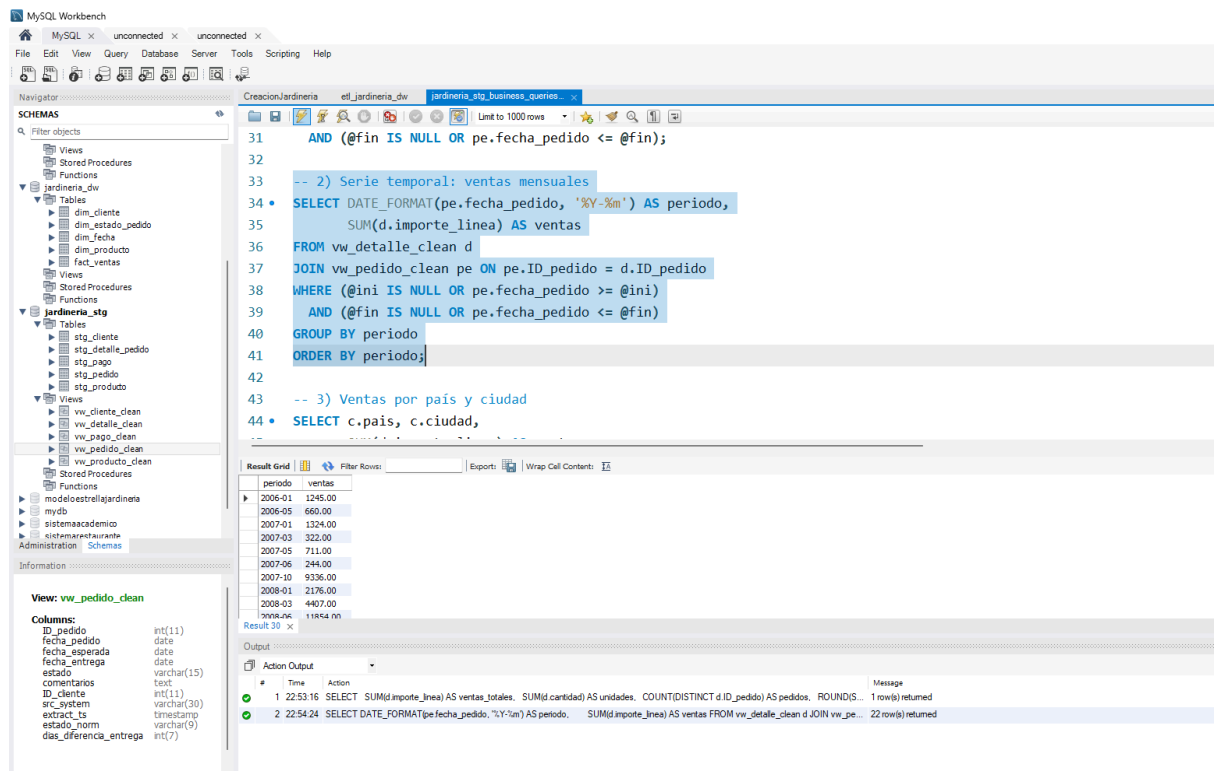


Imagen 2. Consulta de tabla temporal `vw_detalle_clean` uniendo con `vw_pedido_clean` en la base de datos `jardineria_stg` para filtrar ventas por periodos mensuales.



Análisis de calidad de datos y validaciones típicas

Imagen 3. Consulta de SQL Verificación de ventas por país y ciudad, garantizando la coherencia temporal en la base de datos Staging.

The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'SCHEMAS' tree with 'jardineria_stg' selected. The main editor shows a SQL query for verifying sales data. The 'Result Grid' at the bottom displays a table with columns: país, ciudad, ventas, and pedidos. The 'Action Output' pane at the bottom shows the execution of three SQL statements.

```
49 JOIN vw_cliente_clean c ON c.ID_cliente = pe.ID_cliente
50 WHERE (@ini IS NULL OR pe.fecha_pedido >= @ini)
51 AND (@fin IS NULL OR pe.fecha_pedido <= @fin)
52 GROUP BY c.país, c.ciudad
53 ORDER BY ventas DESC, pedidos DESC;
54
55 -- 4) Top 10 productos por ventas
56 • SELECT p.CodigoProducto, p.nombre,
57 SUM(d.importe_linea) AS ventas,
58 SUM(d.cantidad) AS unidades
```

país	ciudad	ventas	pedidos
SPAIN	MADRID	103555.00	25
SPAIN	FUENLABRADA	38282.00	17
USA	NEW YORK	11826.00	8
SPAIN	GETAFE	11441.00	5
SPAIN	MONTEÑONES DEL VALLES	10427.00	7
AUSTRALIA	SYDNEY	9509.00	8
USA	SAN FRANCISCO	7556.00	9
USA	MIAMI	6900.00	5
UNITED KINGDOM	LONDON	5523.00	4
SPAIN	HUMANES	4650.00	7
SPAIN	BARCELONA	4489.00	5
FRANCE	PARIS	2460.00	1

View: vw_pedido_clean

Columns:

- ID_pedido: int(11)
- fecha_pedido: date
- fecha_esperada: date
- fecha_entrega: date
- estado: varchar(15)
- comentarios: text
- ID_cliente: int(11)
- src_system: varchar(30)
- extract_ts: timestamp
- estado_norm: varchar(9)
- das_diferencia_entrega: int(7)

Result 31

Action Output

#	Time	Action	Message
1	22.53.16	SELECT SUM(d.importe_linea) AS ventas_totales, SUM(d.cantidad) AS unidades, COUNT(DISTINCT d.ID_pedido) AS pedidos, ROUND(SUM(d.importe_linea) / COUNT(DISTINCT d.ID_pedido), 2) AS promedio AS ventas_promedio FROM vw_pedido_clean d JOIN vw_cliente_clean c ON c.ID_cliente = d.ID_cliente WHERE (@ini IS NULL OR d.fecha_pedido >= @ini) AND (@fin IS NULL OR d.fecha_pedido <= @fin) GROUP BY c.país, c.ciudad ORDER BY ventas_totales DESC, unidades DESC, promedio DESC;	1 row(s) returned
2	22.54.24	SELECT DATE_FORMAT(pe.fecha_pedido, '%Y-%m') AS periodo, SUM(d.importe_linea) AS ventas FROM vw_pedido_clean d JOIN vw_cliente_clean c ON c.ID_cliente = d.ID_cliente WHERE (@ini IS NULL OR pe.fecha_pedido >= @ini) AND (@fin IS NULL OR pe.fecha_pedido <= @fin) GROUP BY c.país, c.ciudad, periodo ORDER BY ventas DESC, pedidos DESC;	22 row(s) returned
3	22.55.01	SELECT c.país, c.ciudad, SUM(d.importe_linea) AS ventas, COUNT(DISTINCT d.ID_pedido) AS pedidos FROM vw_pedido_clean d JOIN vw_cliente_clean c ON c.ID_cliente = d.ID_cliente WHERE (@ini IS NULL OR d.fecha_pedido >= @ini) AND (@fin IS NULL OR d.fecha_pedido <= @fin) GROUP BY c.país, c.ciudad ORDER BY ventas DESC, pedidos DESC;	12 row(s) returned

Imagen 4 . Consulta de SQL Consulta de 10 productos por ventas.

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

Schemas

- Views
- Stored Procedures
- Functions
- jardineria_dw
 - Tables
 - dim_cliente
 - dim_estado_pedido
 - dim_fecha
 - dim_producto
 - fact_ventas
 - Views
 - Stored Procedures
 - Functions
- jardineria_stg
 - Tables
 - stg_cliente
 - stg_detalle_pedido
 - stg_pago
 - stg_pedido
 - stg_producto
 - Views
 - vw_cliente_clean
 - vw_detalle_clean
 - vw_pago_clean
 - vw_pedido_clean
 - vw_producto_clean
 - Stored Procedures
 - Functions
- modelostrallajardineria
- mydb
- sistemasacademico
- sistemasadministrativo

Information

View: vw_pedido_clean

Columns:

- ID_pedido int(11)
- fecha_pedido date
- fecha_esperada date
- estado varchar(15)
- comentarios text
- ID_cliente int(11)
- src_system varchar(30)
- extract_ts timestamp
- estado_norm varchar(9)
- dias_diferencia_entrega int(7)

Object Info Session

CreacionJardineria etl_jardineria_dw jardineria_stg_business_queries

```
49 JOIN vw_cliente_clean c ON c.ID_cliente = pe.ID_cliente
50 WHERE (@ini IS NULL OR pe.fecha_pedido >= @ini)
51 AND (@fin IS NULL OR pe.fecha_pedido <= @fin)
52 GROUP BY c.pais, c.ciudad
53 ORDER BY ventas DESC, pedidos DESC;
54
55 -- 4) Top 10 productos por ventas
56 SELECT p.CodigoProducto, p.nombre,
57 SUM(d.importe_linea) AS ventas,
58 SUM(d.cantidad) AS unidades
```

Result Grid

CodigoProducto	nombre	ventas	unidades
OR-250	Yucca Jewel	73510.00	279
FR-67	Cerezo	19230.00	275
OR-230	Cordylne Australis -ORACAENA	15104.00	235
FR-11	Lmonero 30/40	13092.00	131
OR-216	Brahea Edulis	9310.00	35
FR-85	Kaki	5320.00	76
OR-214	Brahea Armata	4130.00	70
OR-237	Livistonia Australis	3520.00	55
OR-213	Bismarckia Nobilis	3120.00	80
OR-128	Camelia japonica ejemplar	2900.00	29

Result 32 x

Output

Action Output

#	Time	Action	Message
1	22:53:16	SELECT SUM(d.importe_linea) AS ventas_totales, SUM(d.cantidad) AS unidades, COUNT(DISTINCT d.ID_pedido) AS pedidos, ROUND(SUM(d.importe_linea) / COUNT(DISTINCT d.ID_pedido), 2) AS promedio AS promedio	1 row(s) returned
2	22:54:24	SELECT DATE_FORMAT(pe.fecha_pedido, '%Y-%m') AS periodo, SUM(d.importe_linea) AS ventas FROM vw_detalle_clean d JOIN vw_pedido_clean pe ON d.ID_pedido = pe.ID_pedido GROUP BY periodo	22 row(s) returned
3	22:55:01	SELECT c.pais, c.ciudad, SUM(d.importe_linea) AS ventas, COUNT(DISTINCT d.ID_pedido) AS pedidos FROM vw_detalle_clean d JOIN vw_cliente_clean c ON d.ID_cliente = c.ID_cliente GROUP BY c.pais, c.ciudad	12 row(s) returned
4	22:56:03	SELECT p.CodigoProducto, p.nombre, SUM(d.importe_linea) AS ventas, SUM(d.cantidad) AS unidades FROM vw_detalle_clean d JOIN vw_producto_clean p ON d.CodigoProducto = p.CodigoProducto GROUP BY p.CodigoProducto, p.nombre ORDER BY ventas DESC, unidades DESC	10 row(s) returned

Imagen 5. Consulta de SQL Validación de 10 productos por margen.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the following structure:

- Views
- Stored Procedures
- Functions
- jardineria_dw
 - Tables
 - dim_cliente
 - dim_estado_pedido
 - dim_fecha
 - dim_producto
 - fact_ventas
 - Views
 - Stored Procedures
 - Functions
- jardineria_stg
 - Tables
 - stg_cliente
 - stg_detalle_pedido
 - stg_pago
 - stg_pedido
 - stg_producto
 - Views
 - vw_cliente_clean
 - vw_detalle_clean
 - vw_pago_clean
 - vw_pedido_clean
 - vw_producto_clean
 - Stored Procedures
 - Functions
- modeloestrellajardineria
- mydb
- sistemaacademico
- sistemarestaurante

The main editor shows a SQL query with the following lines highlighted:

```
77 AND (@fin IS NULL OR pe.fecha_pedido <= @fin)
78 GROUP BY p.ID_producto, p.CodigoProducto, p.nombre
79 ORDER BY margen DESC
80 LIMIT 10;
81
82 -- 6) Ventas por categoría
83 • SELECT p.Categoria AS categoria_id,
84         SUM(d.importe_linea) AS ventas,
85         SUM(d.cantidad) AS unidades
86 FROM vw_detalle_clean d
```

The 'Result Grid' at the bottom displays the following data:

	CodigoProducto	nombre	margen	ventas	unidades
▶	OR-250	Yucca Jewel	71278.00	73510.00	279
	OR-216	Brahea Edulis	8785.00	9310.00	35
	OR-230	Cordylne Australis -DRACAENA	8024.00	15104.00	236
	FR-67	Cerezo	3850.00	19250.00	275
	OR-214	Brahea Armata	3570.00	4130.00	70
	OR-237	Livistonia Australis	2695.00	3520.00	55
	FR-11	Limonero 30/40	2612.00	13092.00	131
	OR-179	Sesbania Punicea	1650.00	2250.00	150
	OR-225	Chamaerops Humilis	1071.00	1239.00	21
	FR-85	Kaki	1064.00	5320.00	76

The 'Information' panel at the bottom left shows the details for the view 'vw_pedido_clean':

View: vw_pedido_clean

Columns:

- ID_pedido: int(11)
- fecha_pedido: date

Imagen 6. Consulta de SQL Selección de la vista de limpieza vw_detalle_clean uniendo 2 vistas más para filtrar información relevante a las ventas por categoría.

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays a tree view of databases and tables. The 'jardineria_stg' database is selected, showing tables like 'stg_cliente', 'stg_detalle_pedido', 'stg_pago', 'stg_pedido', and 'stg_producto'. Below this, the 'vw_detalle_clean' view is highlighted. The main editor pane contains a SQL query (lines 81-90) that selects data from 'vw_detalle_clean' and joins it with 'vw_producto_clean' and 'vw_pedido_clean'. The query filters for specific date ranges using '@ini' and '@fin' variables. The 'Result Grid' at the bottom shows the output of the query, which is a table with three columns: 'categoria_id', 'ventas', and 'unidades'. The results are as follows:

categoria_id	ventas	unidades
5	139774.00	3131
4	69932.00	2912
2	4830.00	387
3	2082.00	2057

The 'Information' pane at the bottom left shows the details of the selected view 'vw_pedido_clean', including its columns: 'ID_pedido' (int(11)) and 'fecha_pedido' (date).

Imagen 7. Consulta de SQL para obtener la información correspondiente al filtrado de la información de los pedidos por estados, usando la vista vw_pedido_clean.

The screenshot displays the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows the database structure, including tables like 'dim_cliente', 'dim_estado_pedido', 'dim_fecha', 'dim_producto', 'fact_ventas', and views like 'vw_cliente_clean', 'vw_detalle_clean', 'vw_pago_clean', 'vw_pedido_clean', and 'vw_producto_clean'. The 'vw_pedido_clean' view is selected.

The main editor shows a SQL query (lines 94-103) that filters orders by state. The query is as follows:

```

94 -- 7) Filtrar por estado
95 • SELECT pe.estado_norm,
96        COUNT(DISTINCT pe.ID_pedido) AS pedidos,
97        SUM(d.importe_linea) AS ventas
98 FROM vw_pedido_clean pe
99 LEFT JOIN vw_detalle_clean d ON d.ID_pedido = pe.ID_pedido
100 WHERE (@ini IS NULL OR pe.fecha_pedido >= @ini)
101        AND (@fin IS NULL OR pe.fecha_pedido <= @fin)
102 GROUP BY pe.estado_norm WITH ROLLUP;
103

```

The 'Result Grid' shows the output of the query, filtered by state. The results are as follows:

estado_norm	pedidos	ventas
ENTREGADO	61	147139.00
PENDIENTE	30	32948.00
RECHAZADO	24	36531.00
TOTAL	115	216618.00

The 'Information' pane at the bottom left shows the columns of the 'vw_pedido_clean' view:

```

Columns:
ID_pedido          int(11)
fecha_pedido       date
fecha_esperada     date
fecha_entrega      date
estado             varchar(15)
comentarios        text
ID_cliente         int(11)
src_system         varchar(30)
extract_ts         timestamp
estado_norm        varchar(9)
dias_diferencia_entrega int(7)

```

The 'Result 35' pane at the bottom right shows the execution log, indicating that the query was executed successfully and returned 4 rows.

Conclusiones

El ejercicio permitió aplicar consultas SQL que faciliten la extracción y validación de los datos desde la base de datos origen, así como la implementación de transformaciones mediante SQL que permitieron depurar, normalizar y organizar la información en el entorno de Staging.

Las consultas implementadas en SQL y cuyo resultado se encuentra ilustrado, permitió la carga de datos de manera organizada y precisa que permitió analizar los datos para la toma de decisiones.

La realización de esta actividad, más allá de tratarse de una simple transformación, permitió evidenciar que los datos se encuentren preparados para el análisis, a través de la respectivas consultas, cuyos resultados son relevantes como parte de la información y conocimiento requerido para analizar los datos de acuerdo con el negocio al que pertenecen.

Mediante la presente actividad se aplicó un proceso ETL al modelo relacional Jardinería, en el cual se integraron los conocimientos adquiridos en diseño de bases de datos, normalización y construcción de consultas SQL. Con ésto no sólo se buscó no solo reforzar la teoría aprendida, sino también llevarla a un ejercicio práctico que sirva como preparación para los desafíos reales que plantea la gestión y el aprovechamiento de los datos en entornos profesionales

Anexos

Query de creación de base de datos jardineria_dw y jardineria_stg con vistas de limpieza,
Script de inserción de datos, documento del script de las consultas ETL_Jardineria y Backup
de las tres bases de datos unificadas en un sql.

[https://drive.google.com/file/d/1ap7O0P3ZzVrA3qvHvMjHwVz4NygesFG3/view?usp=shari
ng](https://drive.google.com/file/d/1ap7O0P3ZzVrA3qvHvMjHwVz4NygesFG3/view?usp=sharing)

Referencias Bibliográficas

IUDigital de Antioquia (2025). *Programación orientada a objetos II. Unidad 3*.

Star Data Engineering (15 sep 2025). *What is the staging area?* [Video]. Youtube.

<https://www.youtube.com/watch?v=4Q4FT594ZLA>