

B2B sell-out manufacturing

Javier Acevedo, Jose Ignacio Arguello, Willian Lopez

Noviembre 29, 2024

Contenido

1. Introducción.....	4
2. Características del proyecto de investigación que hace uso de Bases de Datos.....	4
2.1. Título proyecto.....	6
2.2. Objetivo General.....	6
2.2.1. Objetivos Específicos.....	7
2.3. Alcance.....	7
2.4. Pregunta de investigación.....	7
2.5. Hipótesis.....	7
3. Reflexiones Sobre El Origen De La Información.....	8
3.1. Cual es el origen de los datos e información.....	9
3.2. Cuales son la consideraciones legales o éticas sobre el origen de la información.....	9
3.3. Cuales son los retos de la información y los datos que utilizará en la base de datos en términos de la calidad y la consolidación.....	9
3.4. Qué espera de la utilización de un sistema de Bases de Datos para su proyecto.....	9
4. Diseño del Modelo de Datos del SMBD (Sistema Manejador de Bases de Datos).....	10
4.1. Características del SMBD (Sistema Manejador de Bases de Datos) para el proyecto.....	10
4.2. Diagrama modelo de datos.....	11
4.3. Imágenes de la Base de Datos.....	12
4.4. Código SQL - lenguaje de definición de datos (DDL).....	12
4.5. Código SQL - Manipulación de datos (DML).....	14
4.6. Código SQL + Resultados: Vistas.....	16
4.7. Código SQL + Resultados: Triggers.....	16
4.8. Código SQL + Resultados: Funciones.....	18
4.9. Código SQL + Resultados: procedimientos almacenados.....	19
5. Bases de Datos No-SQL.....	20
5.1. Diagrama Bases de Datos No-SQL.....	21
5.2. SMBD utilizado para la Base de Datos No-SQL.....	21
6. Aplicación de ETL (Extract, Transform, Load) y Bodega de Datos.....	21

6.1. Ejemplo de aplicación de ETL y Bodega de Datos.....	21
6.2. Automatización de Datos.....	21
6.3. Integración de Datos.....	21
7. Próximos pasos.....	21
7.1. Integración de sistemas y datos en tiempo real.....	21
7.2. Desarrollo de analítica descriptiva y visualización.....	22
8. Lecciones aprendidas.....	23
9. Bibliografía.....	24

1. Introducción

Este proyecto de bases de datos tiene como objetivo mejorar la gestión y eficiencia de la información en una empresa B2B que ofrece productos y servicios a otras organizaciones. El sistema propuesto está diseñado para gestionar de manera centralizada y optimizada las interacciones entre la empresa y sus clientes finales, garantizando un acceso rápido y fiable a datos clave, como historial de compras, preferencias, facturación y soporte. (Li, Tse y Chen, 2024),

El enfoque principal es desarrollar una base de datos robusta y escalable que permita almacenar grandes volúmenes de información relacionada con transacciones, contratos, comunicaciones y performance de productos o servicios, facilitando el análisis de datos y la toma de decisiones estratégicas. Además, el sistema facilitará la automatización de procesos, como la actualización de inventarios, el seguimiento de pedidos y la gestión de incidencias, mejorando así la experiencia del cliente y reduciendo el tiempo de respuesta.(DCKAP, 2024)

Este proyecto no solo busca optimizar la gestión interna de la empresa, sino también fortalecer las relaciones comerciales con los clientes finales al proporcionarles un servicio más ágil y personalizado. A través de una infraestructura de base de datos eficiente y flexible, se busca mejorar la capacidad de la empresa para adaptarse a las demandas del mercado y generar valor agregado en sus operaciones B2B.

En el contexto de una empresa manufacturera como la nuestra, donde el 80% de las ventas se concentra a través de distribuidores y mayoristas, es fundamental entender el comportamiento de ventas del cliente final. Dado que nuestra fortaleza radica en la fabricación y no en la distribución, el seguimiento de las ventas hasta el cliente final (sell-out) se convierte en una pieza clave para la optimización de recursos, la estrategia de mercado y la toma de decisiones comerciales. Este documento aborda la necesidad de desarrollar una base de datos robusta que nos permita analizar detalladamente el comportamiento de las ventas a nivel geográfico y de producto, con el fin de fortalecer nuestra posición en el mercado y optimizar nuestras estrategias de comercialización B2B.

2. Características del proyecto de investigación que hace uso de Bases de Datos

Rastreo Detallado del Sell-Out: La base de datos debe permitir capturar información precisa y detallada sobre las ventas al cliente final, incluso cuando estas se canalizan a través de distribuidores y mayoristas. Esto incluirá el seguimiento de cada transacción, desde el punto de venta hasta el destino final,

proporcionando visibilidad sobre los patrones de compra de los consumidores y las tendencias regionales de demanda.

Segmentación de Clientes y Canales: El sistema debe permitir la segmentación avanzada de clientes y canales de venta. Al poder clasificar a los distribuidores y mayoristas según criterios como volumen de ventas, localización geográfica y tipo de cliente final, la empresa podrá desarrollar estrategias más efectivas de marketing y ventas, ajustadas a las características de cada segmento.

Integración con Sistemas de Distribución: Dado que el 80% de las ventas se canaliza a través de distribuidores, la base de datos debe ser capaz de integrarse con los sistemas de gestión de estos distribuidores, para poder obtener datos en tiempo real sobre las ventas realizadas y los inventarios disponibles. Esto mejorará la capacidad de monitorear las dinámicas de la cadena de suministro y las fluctuaciones de la demanda de forma continua.

Análisis de Rendimiento por Producto y Canal: La base de datos debe incluir funcionalidades avanzadas de análisis para medir el rendimiento de cada producto en diferentes canales de distribución. Esto permitirá identificar productos más vendidos, detectar caídas en las ventas y realizar proyecciones sobre la demanda futura, optimizando la estrategia de inventarios y la distribución de recursos.

Previsión de Demanda: Integrar herramientas de análisis predictivo y algoritmos de machine learning para prever la demanda futura en función de las tendencias históricas, estacionales y los comportamientos de compra de los clientes finales. Esto optimizará los niveles de inventario y reducirá el riesgo de desabastecimiento o exceso de stock.

Visualización de Datos y Dashboard Interactivo: La plataforma debe ofrecer una interfaz visual de fácil comprensión, con paneles de control (dashboards) interactivos que permitan a los usuarios visualizar datos de ventas, inventarios, comportamiento del cliente y otros KPIs clave en tiempo real. Esto facilitará la toma de decisiones y la identificación rápida de áreas de mejora en la estrategia comercial.

Automatización de Reportes y Alertas: La base de datos deberá incluir la capacidad de generar reportes automáticos sobre el rendimiento de ventas, destacando métricas clave como el sell-out por canal, área geográfica y tipo de cliente. Además, se configurarán alertas automáticas para notificar sobre posibles caídas en las ventas o anomalías en los patrones de compra, permitiendo una respuesta proactiva.

Optimización de Estrategias de Marketing: Al contar con datos detallados sobre las preferencias y comportamientos de compra de los clientes finales, el sistema

permitirá diseñar campañas de marketing más focalizadas, tanto a nivel de producto como de canal. Además, se podrá medir el impacto de las campañas de manera más precisa, ajustando las estrategias de manera ágil.

Acceso Multiusuario y Control de Permisos: La base de datos debe permitir un acceso controlado según roles, para garantizar que solo los usuarios autorizados puedan acceder a datos sensibles o realizar modificaciones en los informes y configuraciones. Esto asegura la seguridad de la información y la trazabilidad de las acciones dentro del sistema.

Soporte para Análisis en Tiempo Real: A medida que las ventas se realizan a través de distribuidores y mayoristas, el sistema debe ser capaz de recibir y procesar los datos en tiempo real, garantizando que la información sobre las transacciones esté siempre actualizada y disponible para la toma de decisiones en el momento adecuado.

Optimización de la Cadena de Suministro: La base de datos debe permitir un mejor análisis de la cadena de suministro, optimizando los niveles de inventarios en función de las ventas reales al cliente final. Esto ayudará a reducir costos operativos, minimizar pérdidas por obsolescencia de inventarios y mejorar los tiempos de entrega.

Escalabilidad para Nuevas Áreas de Expansión: A medida que la empresa crezca y se expanda a nuevos mercados o productos, la base de datos debe ser capaz de escalar y adaptarse a las nuevas necesidades sin perder eficiencia. Esto incluye la capacidad de incorporar nuevas fuentes de datos, como ventas online o en mercados internacionales.

El proyecto se centra en el desarrollo de una base de datos integral que capture la información de sell-out en el contexto de una empresa manufacturera de PVC en Colombia. En un entorno donde el 80% de las ventas se canaliza a través de distribuidores y mayoristas, se hace necesario comprender mejor el comportamiento de las ventas al cliente final. La capacidad de rastrear y analizar esta información es crucial para optimizar los recursos y fortalecer las estrategias de comercialización.

2.1. Título proyecto

B2B sell-out manufacturing

2.2. Objetivo General

Desarrollar una base de datos integral que capture la información de sell-out, estructurada para permitir un análisis detallado a nivel geográfico (ciudades,

departamentos, localidades, barrios) y de producto (categorías, subcategorías y materiales), con el fin de facilitar un análisis del comportamiento de ventas.

2.2.1. Objetivos Específicos

- Estructurar la base de datos de sell-out que permite desglosar las ventas por ubicación geográfica, utilizando jerarquías claras como ciudades, departamentos, localidades y barrios.
- Organizar la información de producto en la base de datos bajo jerarquías definidas, como categorías, subcategorías y materiales, para facilitar un análisis profundo por tipo de producto.
- Generar reportes de comportamiento de ventas del cliente final que ofrezcan una visión clara y precisa de los patrones de compra por cliente, ciudad, producto en sus diferentes jerarquías.

2.3. Alcance

Este proyecto tiene como alcance el desarrollo e implementación de una base de datos integral para la captura, almacenamiento y análisis de información de ventas al cliente final (sell-out) en una empresa manufacturera de PVC en Colombia. El sistema permitirá rastrear las ventas realizadas a través de distribuidores y mayoristas, proporcionando datos detallados sobre el comportamiento de compra del cliente final, incluyendo tendencias, preferencias y patrones geográficos.

El proyecto no cubrirá la modificación de los sistemas de los distribuidores ni la implementación de nuevas infraestructuras tecnológicas. El enfoque principal será el diseño e implementación de la base de datos y las herramientas analíticas para optimizar las estrategias comerciales y de distribución.

El ejercicio está dado por una base de datos a partir de facturas de ventas de un fabricante (sell-in) y facturas de venta de un distribuidor mayorista al cliente final (sell-out).

2.4. Pregunta de investigación

Cómo almacenar datos para entender el comportamiento de los clientes finales de una fábrica de PVC colombiana a través del sell-out y que a partir de esto se pueda generar análisis y acompañar la estrategia de crecimiento en ventas.

2.5. Hipótesis

La implementación de una base de datos centralizada permitirá a la fábrica identificar oportunidades de expansión y mejorar la eficiencia operativa al optimizar la gestión de la relación con los mayoristas y los clientes finales.

Justificación: Al consolidar toda la información de ventas y productos en una única base de datos, se podrá tener una visión completa del rendimiento de los mayoristas, detectar áreas de mejora en la operación y crear estrategias más efectivas para fortalecer la relación comercial y expandir la presencia en el mercado final.

3. Reflexiones Sobre El Origen De La Información

El principal reto de este proyecto, en el que la información proviene de varios mayoristas con sus propios sistemas de facturación, es garantizar la homogeneidad y calidad de los datos. Cada mayorista maneja distintos criterios de codificación y categorización de clientes, productos y ventas, lo que exige un proceso riguroso de normalización. La diversidad en los formatos y estructuras de los datos puede generar inconsistencias, duplicaciones y errores si no se gestionan correctamente. Un ejemplo clave es la duplicidad de clientes: un mismo cliente final puede comprar a diferentes mayoristas, y cada uno puede asignarle una identificación distinta. Garantizar la unicidad de los clientes es esencial para un análisis preciso del comportamiento de compra, lo que a su vez permite identificar oportunidades de venta cruzada o personalización de ofertas.

Además, tras la facturación de la fábrica, los mayoristas pueden modificar productos o embalajes, lo que añade complejidad a la gestión de datos. Es necesario convertir esta información al lenguaje de la fábrica, creando equivalencias entre los códigos y descripciones de productos utilizados por los mayoristas y los propios de la fábrica. Sin esta conversión precisa, es probable que se cometan errores en la categorización de productos, lo que afectaría los análisis posteriores de ventas.

Otro reto importante es la obtención de datos detallados para el análisis geográfico. Los mayoristas deben proporcionar información con suficiente granularidad para permitir la segmentación por regiones, ciudades, localidades e incluso barrios. Este nivel de detalle es fundamental para identificar patrones de comportamiento por zona geográfica y ajustar las estrategias de distribución y marketing a las necesidades de cada mercado local. No obstante, la falta de estandarización entre los mayoristas respecto a la geolocalización de los clientes puede dificultar la implementación de una segmentación precisa.

Por lo tanto, un sistema de bases de datos eficiente debe integrar datos de diversas fuentes y gestionar la heterogeneidad de los mismos, asegurando su calidad para sustentar una toma de decisiones estratégica basada en información confiable. La clave es unificar, limpiar y estandarizar los datos, garantizando su unicidad y consistencia, además de traducirlos al lenguaje de la fábrica para hacer un análisis preciso de productos, clientes y zonas geográficas. Esto permitirá a la

fábrica identificar oportunidades y amenazas en el mercado, optimizar la estrategia de *go to market* y fortalecer la relación con los mayoristas.

3.1. Cual es el origen de los datos e información

La base de datos de sell-out proviene de los datos de facturación suministrados por los mayoristas bajo acuerdos de confidencialidad. Esta información proporciona a la fábrica detalles clave sobre las ventas a clientes finales, permitiendo un análisis preciso del comportamiento de mercado. Los datos se estructuran para ofrecer insights geográficos (ciudades, departamentos, localidades, barrios) y jerárquicos del producto (categorías, subcategorías, materiales). Con esta base de datos, la fábrica puede tomar decisiones estratégicas, optimizando su estrategia de *go to market* y fortaleciendo la gestión comercial, a pesar de que el músculo logístico no es su principal fortaleza.

3.2. Cuales son la consideraciones legales o éticas sobre el origen de la información

Esta información es confidencial por consiguiente será usada con fines académicos únicamente.

3.3. Cuales son los retos de la información y los datos que utilizará en la base de datos en términos de la calidad y la consolidación

Uno de los principales retos de la base de datos es asegurar la limpieza y consistencia de los datos, garantizando la unicidad e identificación precisa de los clientes finales. Dado que un cliente final puede comprar a múltiples mayoristas, y cada uno puede codificarlo de manera diferente, lo que complica la consolidación de la información. Además, es fundamental convertir los datos suministrados por los mayoristas al formato estándar de la fábrica, ya que, tras la facturación, pueden ocurrir variaciones en los productos o embalajes. Estos desafíos requieren un enfoque robusto en la normalización y estandarización de los datos para obtener análisis precisos.

3.4. Qué espera de la utilización de un sistema de Bases de Datos para su proyecto

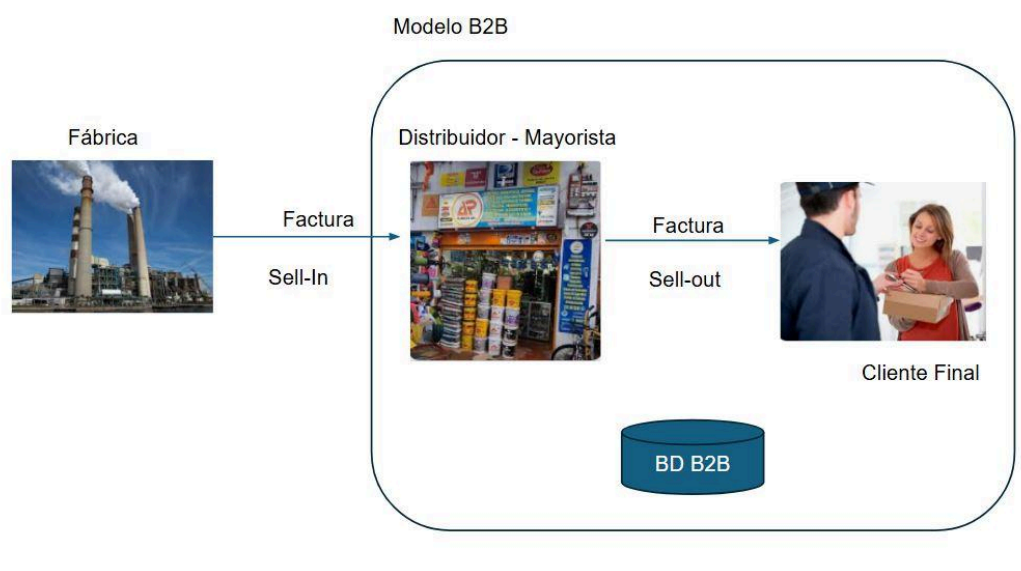
El objetivo de utilizar un sistema de bases de datos en el proyecto es consolidar la información proveniente de diferentes mayoristas, lo que permitirá realizar un análisis profundo del mercado. A través de esta base de datos, se espera identificar oportunidades y amenazas mediante el análisis detallado de productos, segmentos de clientes y zonas geográficas. Esto facilitará la segmentación precisa del mercado, la detección de patrones de compra y la optimización de estrategias

comerciales. Además, permitirá el acceso a datos confiables y actualizados, esenciales para la toma de decisiones informadas y para mejorar la competitividad en un entorno dinámico.

4. Diseño del Modelo de Datos del SMBD (Sistema Manejador de Bases de Datos)

En la Figura 1 se muestra el modelo B2B, a partir del cual se pretende obtener el sistema manejador de bases de datos (SMBD).

Figura 1. Modelo B2B

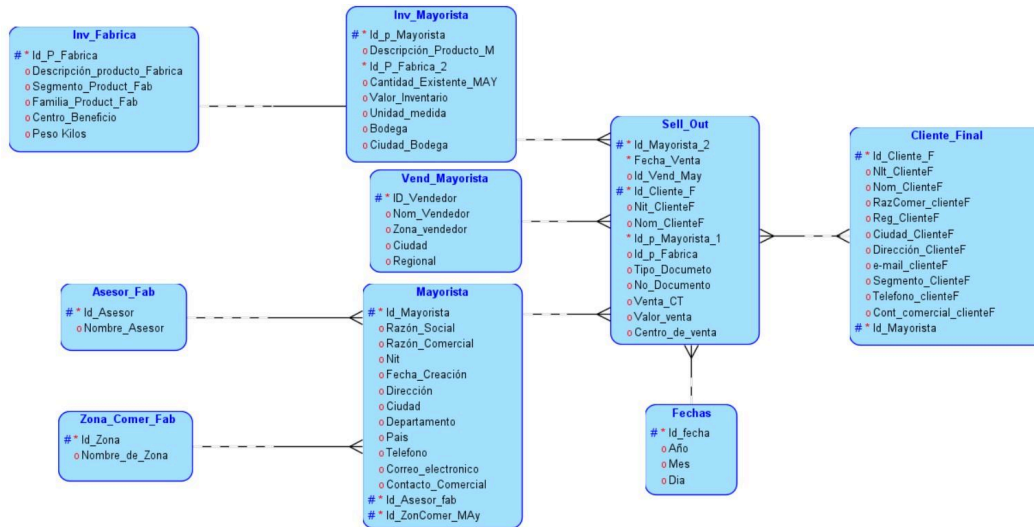


Fuente: Elaboración propia

4.1. Características del SMBD (Sistema Manejador de Bases de Datos) para el proyecto

En la Figura 2 se observa el esquema del sistema manejador de bases de datos, a partir del cual se creó la base de datos y en donde se le asignaron características propias.

Figura 2. Características del SMBD

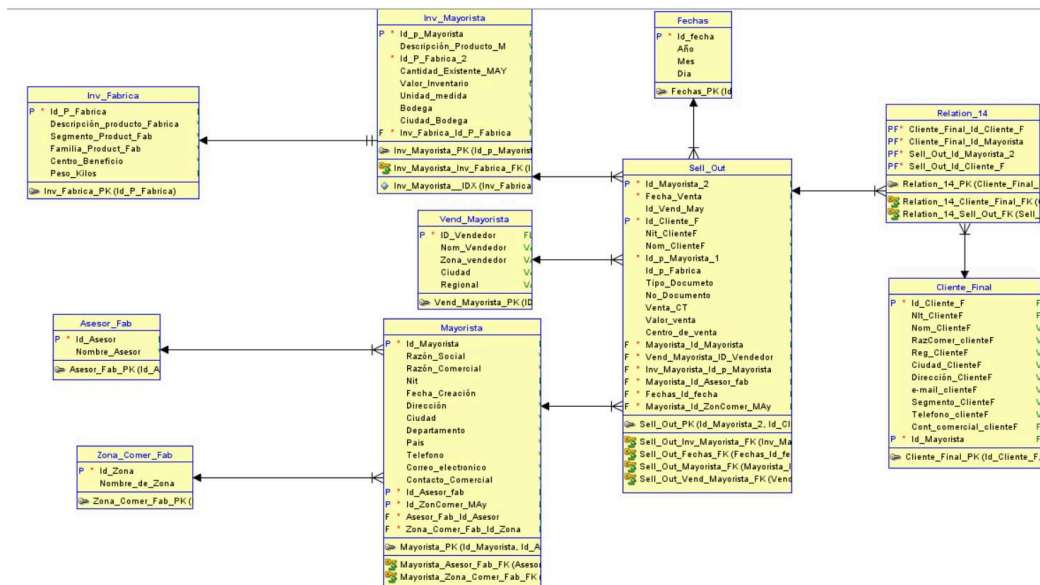


Fuente: Elaboración propia

4.2. Diagrama modelo de datos

En la Figura 3, se muestra el modelo entidad relación que muestra cómo se relacionan las entidades entre sí con sus respectivos atributos, además es posible ver el flujo de información en los procesos.

Figura 3. Modelo Entidad Relación

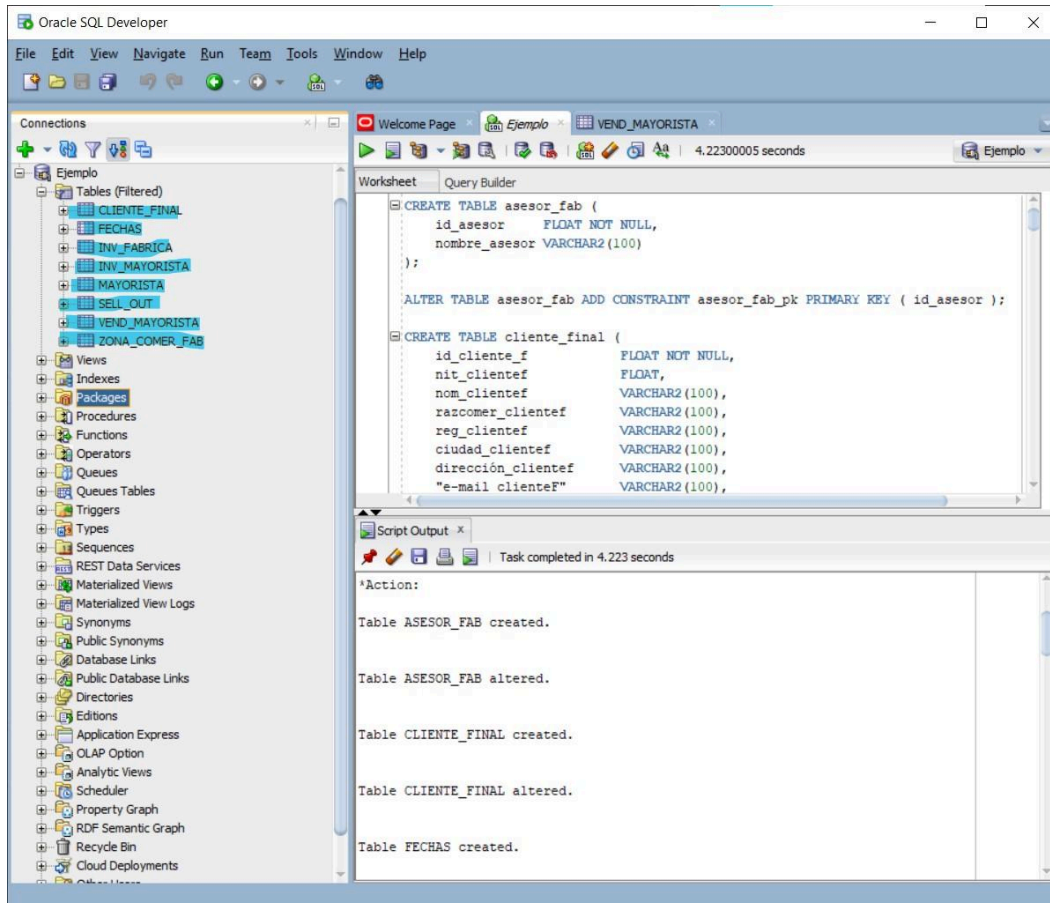


Fuente: Elaboración propia

4.3. Imágenes de la Base de Datos

Después de haber ejecutado el código DDL, en Data Developer, obtenido a partir del diseño de Data Modeler, se crearon las tablas, como se muestra en la Figura 4.

Figura 4. Imagen de las tablas creadas en la base de datos



Fuente: Elaboración propia

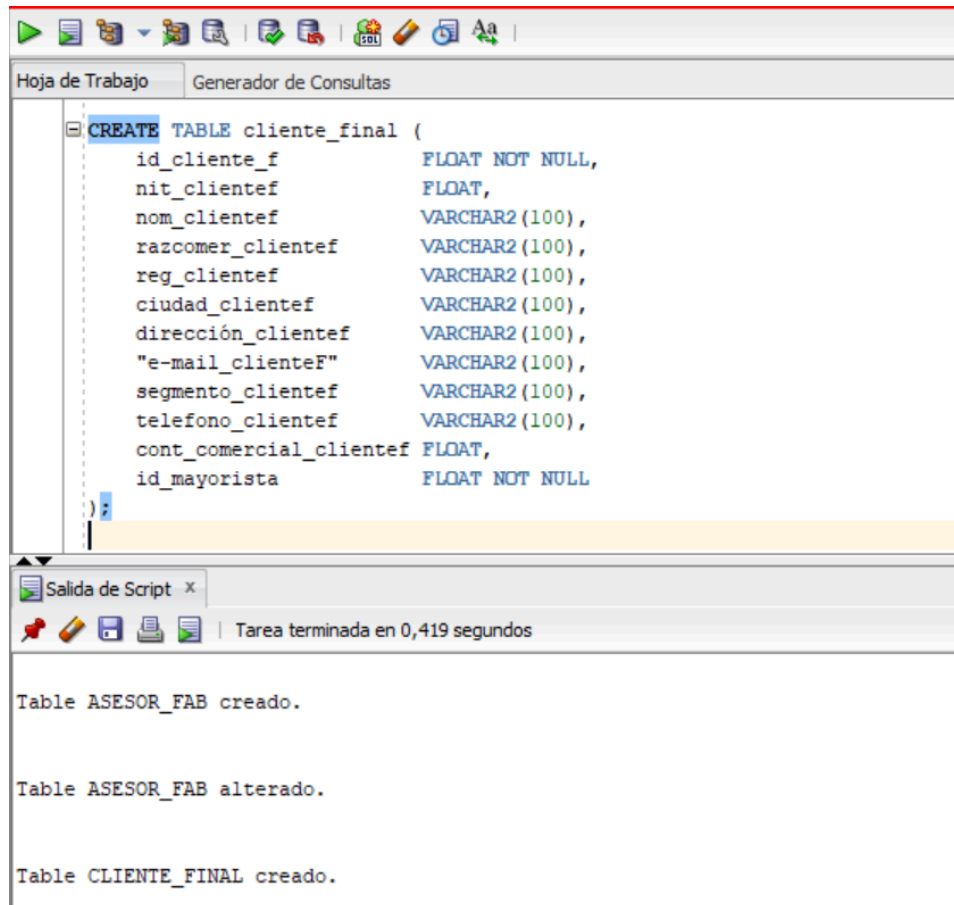
4.4. Código SQL - lenguaje de definición de datos (DDL)

En la siguiente sección se observa un fragmento de código y cómo se genera a partir de ese código las tablas, Figura 5, todo esto con código lenguaje de definición de datos (DDL), un conjunto de instrucciones o comandos en un sistema de gestión de bases de datos (SGBD) que permiten definir, modificar y eliminar estructuras de datos, como tablas, vistas, índices y esquemas.

Unset

```
CREATE TABLE asesor_fab (  
    id_asesor      FLOAT NOT NULL,  
    nombre_asesor  VARCHAR2(100)  
);  
  
ALTER TABLE asesor_fab ADD CONSTRAINT asesor_fab_pk PRIMARY KEY ( id_asesor  
);  
  
CREATE TABLE cliente_final (  
    id_cliente_f      FLOAT NOT NULL,  
    nit_clientef       FLOAT,  
    nom_clientef       VARCHAR2(100),  
    razcomer_clientef  VARCHAR2(100),  
    reg_clientef       VARCHAR2(100),  
    ciudad_clientef    VARCHAR2(100),  
    dirección_clientef VARCHAR2(100),  
    "e-mail_clientef"  VARCHAR2(100),  
    segmento_clientef  VARCHAR2(100),  
    telefono_clientef  VARCHAR2(100),  
    cont_comercial_clientef FLOAT,  
    id_mayorista       FLOAT NOT NULL  
);
```

Figura 5. Resultado ejecucion código DDL



Fuente: Elaboración propia

4.5. Código SQL - Manipulación de datos (DML)

Lenguaje de Manipulación de Datos (DML), se refiere a un conjunto de comandos que se utilizan para manipular o modificar los datos dentro de las tablas de una base de datos. A diferencia de DDL, que se enfoca en la estructura de la base de datos, DML se centra en los datos que están almacenados en ella. En la sección enseguida se observa el fragmento de código con este lenguaje y el resultado de la ejecución, Figura 6.

```
Unset
INSERT INTO asesor_fab (id_asesor, nombre_asesor) VALUES (101020, 'Armando Romero');
INSERT INTO asesor_fab (id_asesor, nombre_asesor) VALUES (101030, 'Juan Carlos Correa');
INSERT INTO asesor_fab (id_asesor, nombre_asesor) VALUES (101040, 'Yolanda Fula');
```

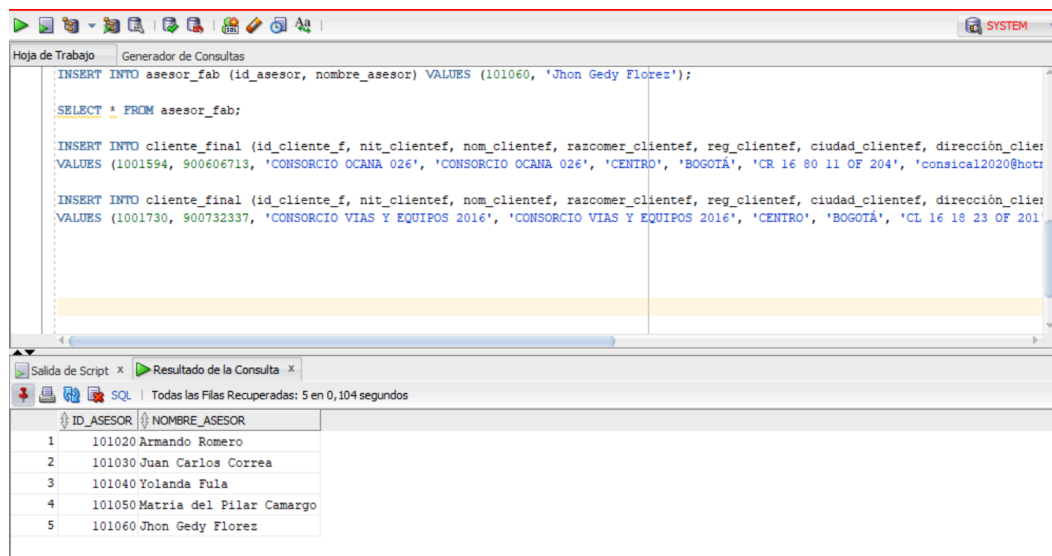
```
INSERT INTO asesor_fab (id_asesor, nombre_asesor) VALUES (101050, 'Matria
del Pilar Camargo');
INSERT INTO asesor_fab (id_asesor, nombre_asesor) VALUES (101060, 'Jhon
Gedy Florez');
```

```
SELECT * FROM asesor_fab;
```

```
INSERT INTO cliente_final (id_cliente_f, nit_cliente_f, nom_cliente_f,
razcomer_cliente_f, reg_cliente_f, ciudad_cliente_f, dirección_cliente_f,
"e-mail_cliente_f", segmento_cliente_f, telefono_cliente_f,
cont_comercial_cliente_f, id_mayorista)
VALUES (1001594, 900606713, 'CONSORCIO OCANA 026', 'CONSORCIO OCANA 026',
'CENTRO', 'BOGOTÁ', 'CR 16 80 11 OF 204', 'consical2020@hotmail.com',
'Consortios y Uniones', '7148058293', 7148058293, 1000002);
```

```
INSERT INTO cliente_final (id_cliente_f, nit_cliente_f, nom_cliente_f,
razcomer_cliente_f, reg_cliente_f, ciudad_cliente_f, dirección_cliente_f,
"e-mail_cliente_f", segmento_cliente_f, telefono_cliente_f,
cont_comercial_cliente_f, id_mayorista)
VALUES (1001730, 900732337, 'CONSORCIO VIAS Y EQUIPOS 2016', 'CONSORCIO
VIAS Y EQUIPOS 2016', 'CENTRO', 'BOGOTÁ', 'CL 16 18 23 OF 201',
'juancarlostorogil2040@yahoo.es', 'Consortios y Uniones', '713203625',
713203625, 1000052);
```

Figura 6. Resultado ejecución código DML



Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 5 en 0,104 segundos

ID_ASESOR	NOMBRE_ASESOR
1	101020 Armando Romero
2	101030 Juan Carlos Correa
3	101040 Yolanda Fula
4	101050 Matria del Pilar Camargo
5	101060 Jhon Gedy Florez

Fuente: Elaboración propia

4.6. Código SQL + Resultados: Vistas

Dado que se considera necesario e importante ver la información de contacto de los mayoristas se decidió crear una vista a partir del siguiente código:

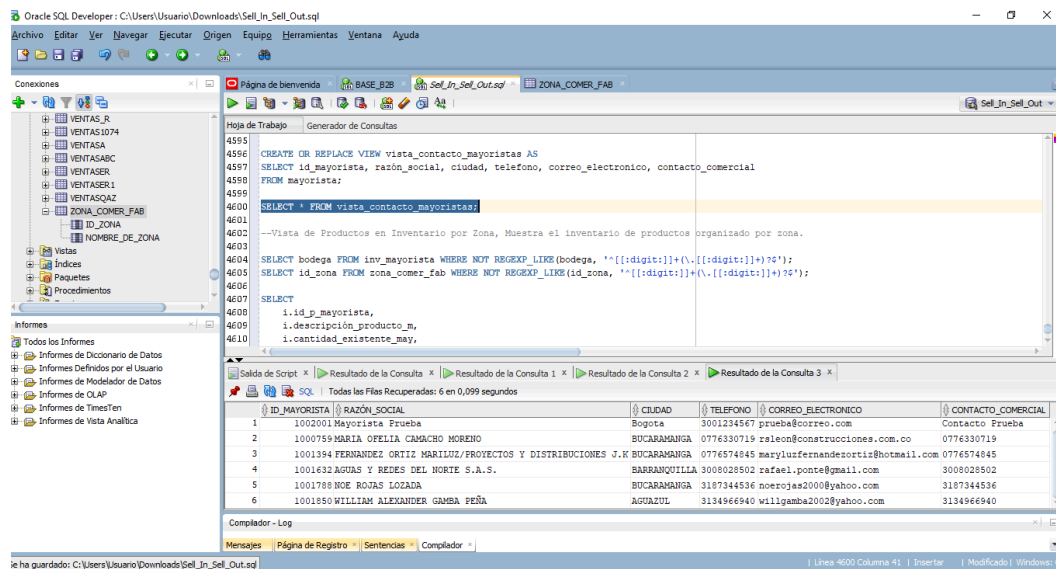
```
Unset

--Vista de Contacto de Mayoristas, Muestra información de contacto de los
mayoristas.

CREATE OR REPLACE VIEW vista_contacto_mayoristas AS
SELECT id_mayorista, razón_social, ciudad, telefono, correo_electronico,
contacto_comercial
FROM mayorista;
```

En la Figura 7 que se muestra a continuación se aprecian los resultados de ejecutar dicha línea de código con la vista.

Figura 7. Resultado ejecución código vistas



ID_MAYORISTA	RAZÓN_SOCIAL	CIUDAD	TELEFONO	CORREO_ELECTRONICO	CONTACTO_COMERCIAL
1	1002001 Mayorista Prueba	Bogota	3001234567	prueba@correo.com	Contacto Prueba
2	1000759 MARIA OFELIA CAMACHO MORENO	BUCARANGA	0776330719	rsleon@construcciones.com.co	0776330719
3	1001354 FERNANDEZ ORTIZ MARILUZ/PROYECTOS Y DISTRIBUCIONES J.K	BUCARANGA	0776574845	maryluzfernandesortiz@hotmail.com	0776574845
4	1001632 AGUAS Y REDES DEL NORTE S.A.S.	BAHRANQUILLA	3008028502	rafael.ponte@gmail.com	3008028502
5	1001788 NOE ROJAS LOZADA	BUCARANGA	3187344536	noerojas2000@yahoo.com	3187344536
6	1001850 WILLIAM ALEXANDER GAMBA PEÑA	AGUILA	3134966940	wiliamba2002@yahoo.com	3134966940

Fuente: Elaboración propia

4.7. Código SQL + Resultados: Triggers

Dada la necesidad de actualizar el inventario de la empresa se generó un código con un Trigger que hace este proceso automáticamente.

```

Unset
CREATE OR REPLACE TRIGGER trg_update_inventario
BEFORE UPDATE OF cantidad_existente_may ON inv_mayorista
FOR EACH ROW
DECLARE
    v_precio NUMBER; -- Variable para almacenar el precio unitario
BEGIN
    -- Intentar obtener el precio desde la tabla productos
    BEGIN
        SELECT precio
        INTO v_precio
        FROM productos
        WHERE producto_id = :NEW.id_p_fabrica_2; -- Usa id_p_fabrica_2 para
relacionar con producto_id

        -- Calcular el nuevo valor de inventario
        :NEW.valor_inventario := :NEW.cantidad_existente_may * v_precio;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            -- Si no se encuentra el producto, asignar un valor nulo o
realizar una acción alternativa
            :NEW.valor_inventario := NULL; -- 0 cualquier otra acción que
consideres apropiada
    END;
END;
/

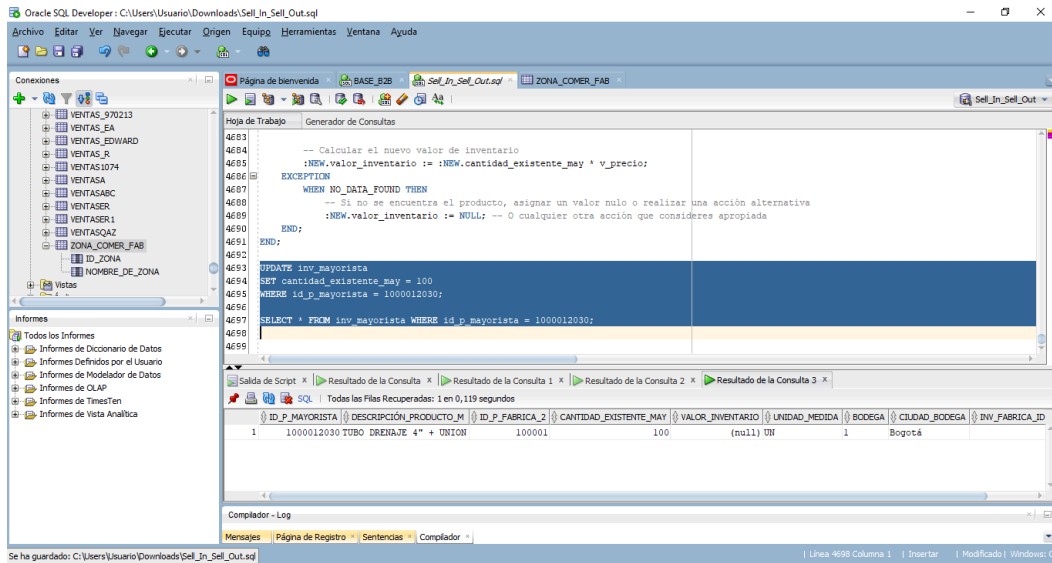
UPDATE inv_mayorista
SET cantidad_existente_may = 100
WHERE id_p_mayorista = 1000012030;

SELECT * FROM inv_mayorista WHERE id_p_mayorista = 1000012030;

```

Es posible ver el resultado de la ejecución del código del Trigger, en la Figura 8.

Figura 8. Resultado ejecución código Triggers



Fuente: Elaboración propia

4.8. Código SQL + Resultados: Funciones

A partir de la necesidad de contar los productos por segmentos se creo una función que es posible ver en la siguiente sección.

```

Unset
CREATE OR REPLACE FUNCTION contar_productos_por_segmento(p_segmento
VARCHAR2)
RETURN NUMBER IS
    v_count NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO v_count
    FROM inv_fabrica
    WHERE segmento_product_fab = p_segmento;

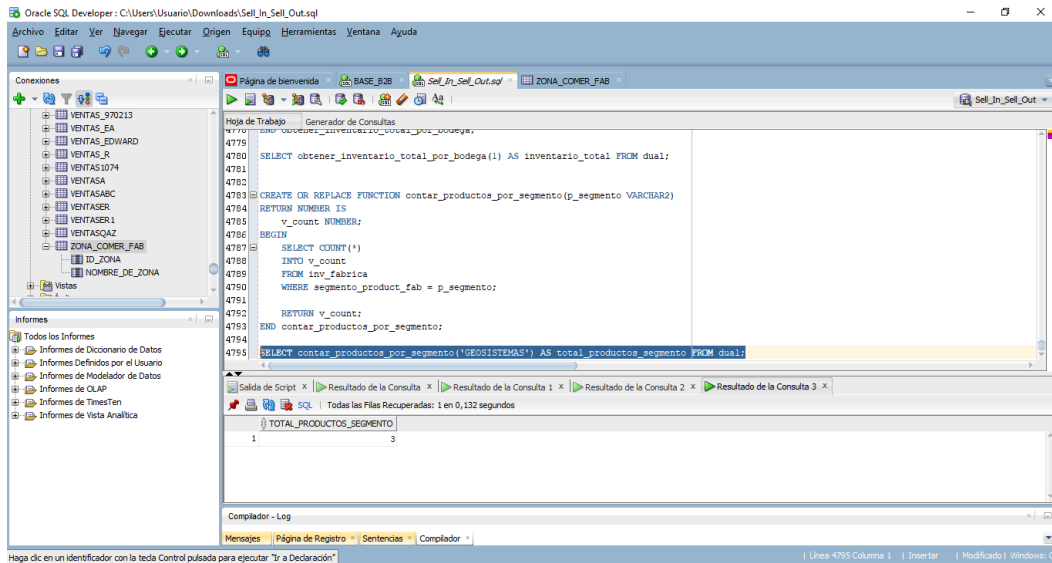
    RETURN v_count;
END contar_productos_por_segmento;

SELECT          contar_productos_por_segmento('GEOSISTEMAS')      AS
total_productos_segmento FROM dual;

```

En la Figura 9 que se muestra a continuación se aprecian los resultados de ejecutar dicha línea de código de la función.

Figura 9. Resultado ejecucion código Función



Fuente: Elaboración propia

4.9. Código SQL + Resultados: procedimientos almacenados

A partir de la necesidad de actualizar los inventarios se creó un procedimiento almacenado que es posible ver en la siguiente sección.

```

Unset

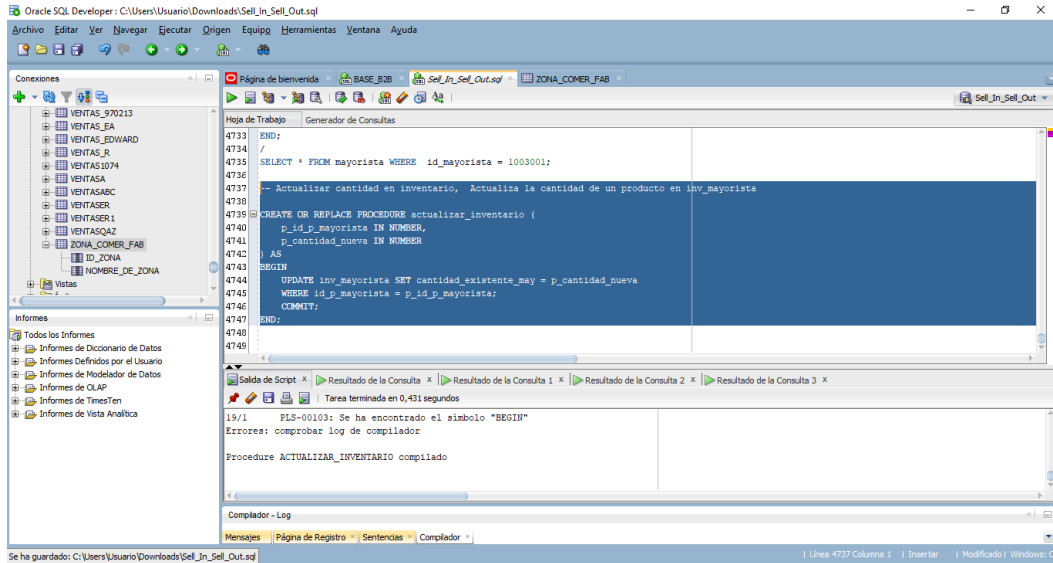
-- Actualizar cantidad en inventario, Actualiza la cantidad de un producto
en inv_mayorista

CREATE OR REPLACE PROCEDURE actualizar_inventario (
    p_id_p_mayorista IN NUMBER,
    p_cantidad_nueva IN NUMBER
) AS
BEGIN
    UPDATE inv_mayorista SET cantidad_existente_may = p_cantidad_nueva
    WHERE id_p_mayorista = p_id_p_mayorista;
    COMMIT;
END;

```

En la Figura 10 que se muestra a continuación se aprecian los resultados de ejecutar dicha línea de código del procedimiento almacenado.

Figura 10. Resultado ejecución código Procedimiento almacenado

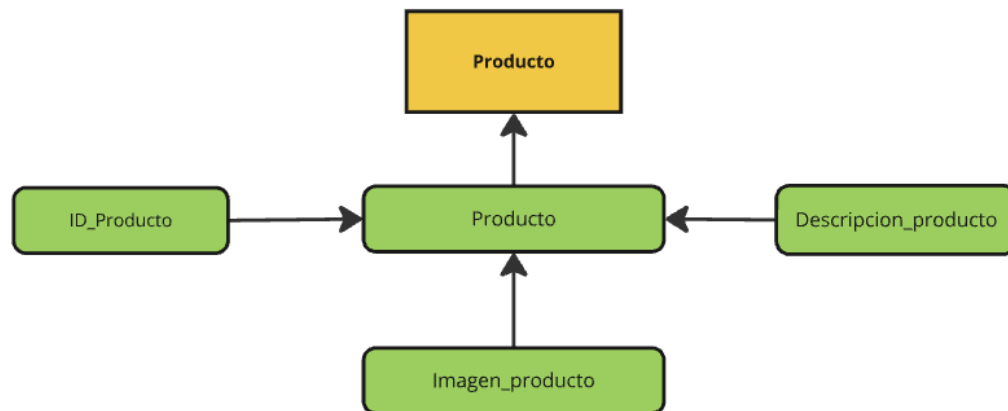


Fuente: Elaboración propia

5. Bases de Datos No-SQL

A continuación en la Figura 11, se muestra el diseño de la base de datos NoSQL, es el modelo conceptual como se cargan imágenes de los productos que existen para la base de datos construida.

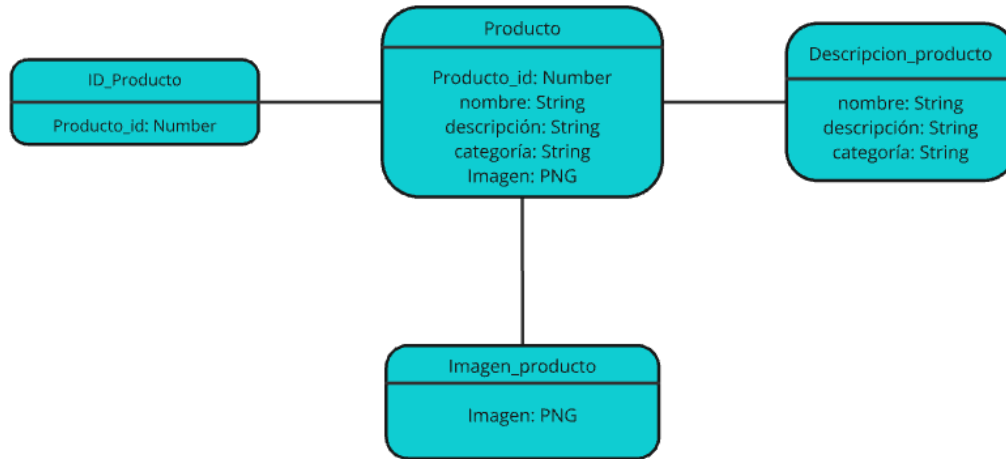
Figura 11. Modelo conceptual SMBD No-SQL



Fuente: Elaboración propia

5.1. Diagrama Bases de Datos No-SQL

Figura 12. Modelo logico SMBD No-SQL

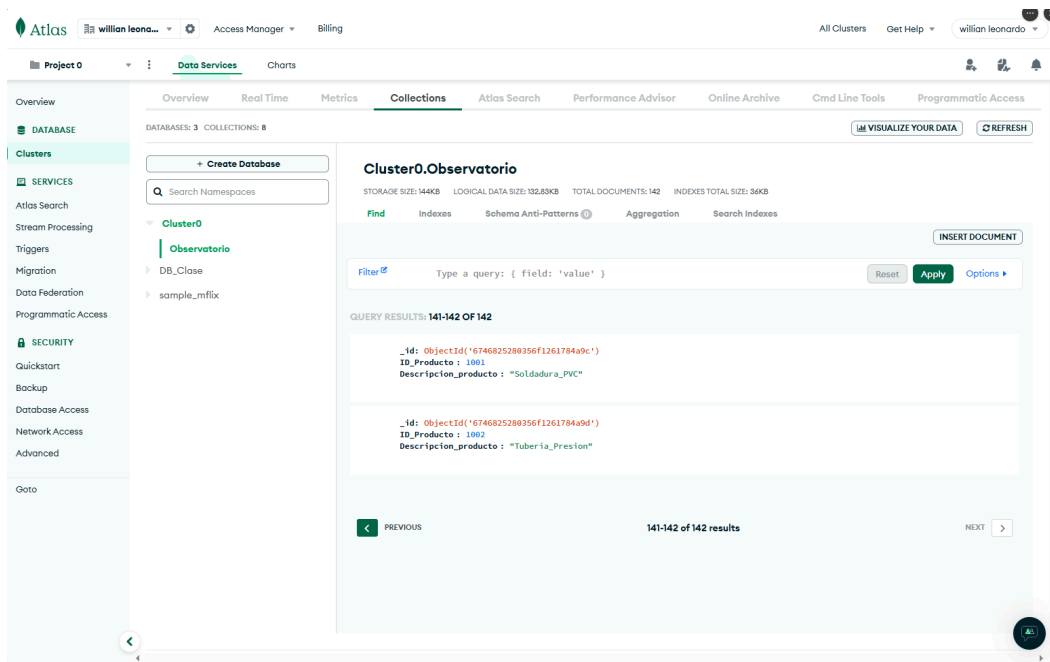


Fuente: Elaboración propia

5.2. SMBD utilizado para la Base de Datos No-SQL

En la Figura 13, se observa cómo quedó una parte de las imágenes con unas características en MongoDB Atlas.

Figura 13. Datos cargados de los productos cargados en MongoDB Atlas



Fuente: Elaboración propia

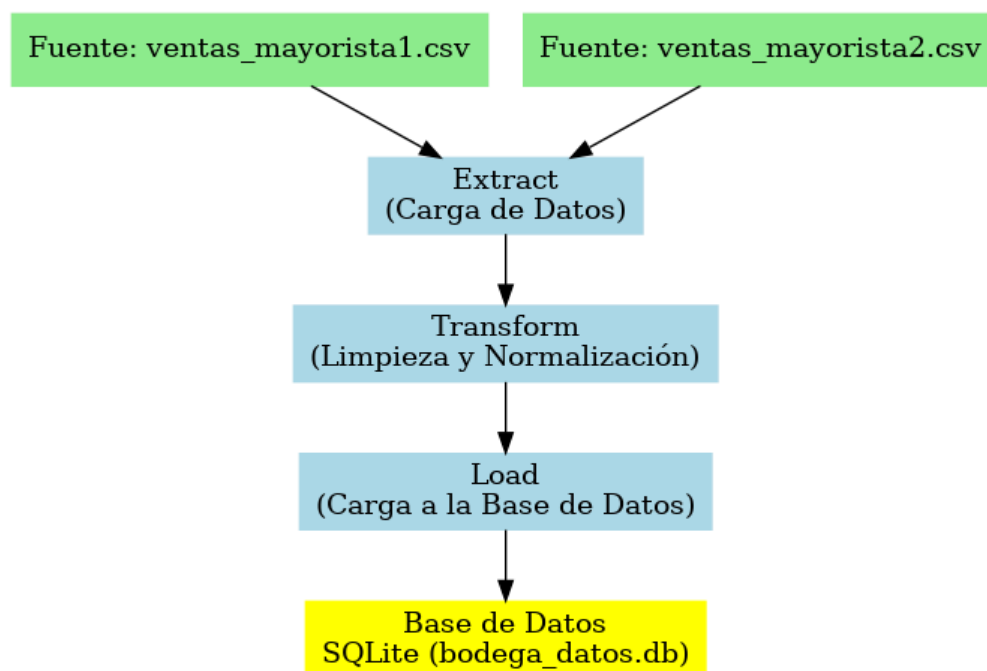
6. Aplicación de ETL (Extract, Transform, Load) y Bodega de Datos

La arquitectura ETL y la bodega de datos son fundamentales en el mundo de la inteligencia empresarial. Nos permite la integración, procesamiento y almacenamiento de datos desde diversas fuentes para su posterior análisis. A continuación se explica cómo se aplican ambos conceptos en el contexto de nuestro proyecto B2B.

6.1. Ejemplo de aplicación de ETL y Bodega de Datos

A continuación se observa el modelo creado para los datos de ventas de mayoristas de extracción, transformación y carga de datos en la Figura 14.

Figura 14. Diagrama modelo arquitectura ETL



Fuente: Elaboración propia

A partir del anterior diagrama se generó el siguiente código en python para desarrollar la arquitectura ETL propuesta.

```
Python
import pandas as pd
```

```

from sqlalchemy import create_engine

# Paso 1: Extract - Cargar datos desde fuentes (simulados con archivos CSV)
def extract_data(file_paths):
    dataframes = [pd.read_csv(file) for file in file_paths]
    return pd.concat(dataframes, ignore_index=True)

# Paso 2: Transform - Limpiar y estandarizar los datos
def transform_data(df):
    # Normalización de columnas
    df.columns = [col.lower().replace(" ", "_") for col in df.columns]

    # Manejo de valores nulos
    df.fillna({'cliente_final': 'Desconocido', 'monto_venta': 0},
inplace=True)

    # Conversión de formatos
    df['fecha_venta'] = pd.to_datetime(df['fecha_venta'])

    # Agregar columna calculada (ejemplo: región por cliente)
    df['region'] = df['ciudad'].map(lambda x: 'Norte' if x in ['Bogotá',
'Medellín'] else 'Sur')

    return df

# Paso 3: Load - Guardar los datos transformados en la base de datos
def load_data(df, db_url, table_name):
    engine = create_engine(db_url)
    with engine.connect() as connection:
        df.to_sql(table_name, con=connection, if_exists='replace',
index=False)

# Archivo de entrada
file_paths = ['ventas_mayorista1.csv', 'ventas_mayorista2.csv']

# Base de datos destino (SQLite para ejemplo)
db_url = 'sqlite:///bodega_datos.db'
table_name = 'ventas_procesadas'

```

```

# Pipeline ETL
def etl_pipeline(file_paths, db_url, table_name):
    print("Iniciando proceso ETL...")
    # Extracción
    raw_data = extract_data(file_paths)
    print("Datos extraídos.")

    # Transformación
    processed_data = transform_data(raw_data)
    print("Datos transformados.")

    # Carga
    load_data(processed_data, db_url, table_name)
    print(f"Datos cargados en la tabla {table_name}.")

etl_pipeline(file_paths, db_url, table_name)

```

6.2. Automatización de Datos

Para automatizar tareas de recolección, procesamiento, análisis y carga de datos, en pro de mejorar la eficiencia, reducir errores y hacer que los procesos de datos sean más rápidos y escalables. Se generó un código en tipo bat, que automatiza la recolección, transformación, limpieza y carga de datos.

```

Unset
@echo off
REM Ejecutar el proceso automatizado ETL en Python
python -c "
import pandas as pd
from sqlalchemy import create_engine

def extract_data(file_paths):
    dataframes = [pd.read_csv(file) for file in file_paths]
    return pd.concat(dataframes, ignore_index=True)

def transform_data(df):
    df.columns = [col.lower().replace(' ', '_') for col in df.columns]

```

```

        df.fillna({'cliente_final': 'Desconocido', 'monto_venta': 0},
inplace=True)
        df['fecha_venta'] = pd.to_datetime(df['fecha_venta'])
        df['region'] = df['ciudad'].map(lambda x: 'Norte' if x in ['Bogotá',
'Medellín'] else 'Sur')
        return df

def load_data(df, db_url, table_name):
    engine = create_engine(db_url)
    with engine.connect() as connection:
        df.to_sql(table_name, con=connection, if_exists='replace',
index=False)

file_paths = ['ventas_mayorista1.csv', 'ventas_mayorista2.csv']
db_url = 'sqlite:///bodega_datos.db'
table_name = 'ventas_procesadas'

raw_data = extract_data(file_paths)
processed_data = transform_data(raw_data)
load_data(processed_data, db_url, table_name)

print('ETL completado.')
"
pause

```

6.3. Integración de Datos

Dado la anterior sección de automatización para la recolección, transformación y limpieza de los datos recolectados se procede a unir estos datos recolectados de diferentes fuentes, y adicional a esto se incluyó la generación de informes y análisis con el código que se muestra a continuación.

```

Unset
let
    Source1 = Csv.Document(File.Contents("ventas_mayorista1.csv"),
[Delimiter=",", Columns=6, Encoding=65001, QuoteStyle=QuoteStyle.None]),
    Source2 = Csv.Document(File.Contents("ventas_mayorista2.csv"),
[Delimiter=",", Columns=6, Encoding=65001, QuoteStyle=QuoteStyle.None]),
    CombinedData = Table.Combine({Source1, Source2}),
    PromotedHeaders = Table.PromoteHeaders(CombinedData,
[PromoteAllScalars=true]),

```



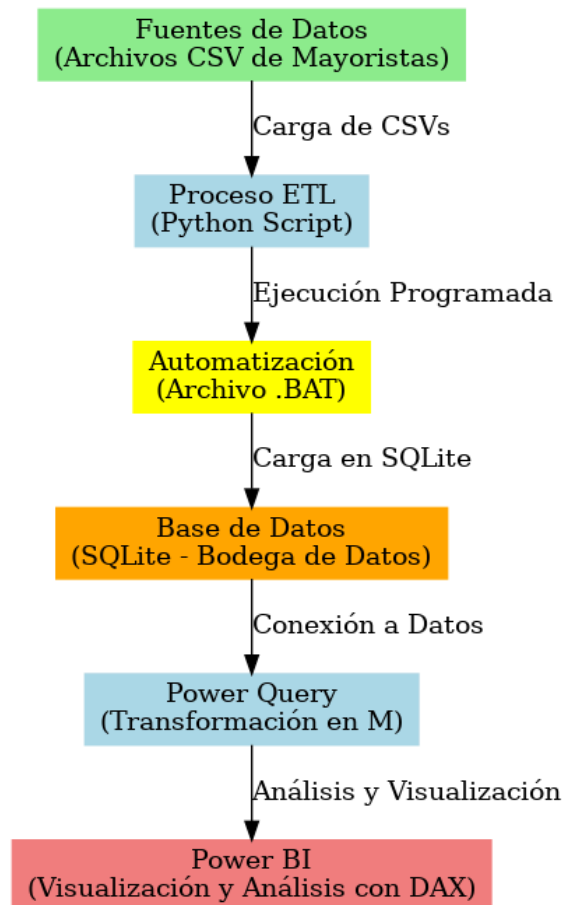
```

        ReplacedNulls = Table.ReplaceValue(PromotedHeaders, null,
"Desconocido", Replacer.ReplaceValue, {"cliente_final"}),
        AddedRegion = Table.AddColumn(ReplacedNulls, "region", each if [ciudad]
= "Bogotá" or [ciudad] = "Medellín" then "Norte" else "Sur"),
        ChangedTypes = Table.TransformColumnTypes(AddedRegion, {"fecha_venta",
type date}, {"monto_venta", type number}))
in
    ChangedTypes

```

En la Figura 15, se observa una aplicación del modelo general con la arquitectura ETL y bodega de datos, incluido la automatización e integración de datos.

Figura 15. Modelo de integración y analítica en tiempo real



Fuente: Elaboración propia

La implementación del modelo ETL centraliza la información de múltiples fuentes (como archivos CSV) en una base de datos única, asegurando datos limpios, organizados y listos para el análisis, sin redundancias ni errores. El uso de archivos .BAT automatiza el proceso ETL, garantizando una carga y transformación regular de datos sin intervención manual, mejorando la consistencia y reduciendo el tiempo en tareas repetitivas. El modelo es flexible y escalable, permitiendo la incorporación de nuevas fuentes de datos sin reestructurar el flujo. La integración con Power Query y Power BI facilita transformaciones adicionales y la creación de dashboards interactivos, apoyando la toma de decisiones estratégicas mediante análisis en tiempo real. La limpieza y normalización de los datos, junto con las transformaciones en Power Query, aseguran alta calidad y confiabilidad en los análisis.

7. Próximos pasos

7.1. Integración de sistemas y datos en tiempo real

El primer paso crucial para avanzar en este proyecto es conectar los sistemas de los mayoristas con la base de datos central de la fábrica, estableciendo una integración mediante interfaces API. Esto permitirá la sincronización en tiempo real de datos clave como ventas, inventarios y comportamiento del cliente final. La automatización de esta integración reducirá errores manuales, garantizará datos actualizados y aumentará la capacidad de respuesta ante fluctuaciones en la demanda. Además, se establecerán estándares de calidad y homologación de datos para superar los retos de normalización identificados.

7.2. Desarrollo de analítica descriptiva y visualización

Una vez consolidada la base de datos, se implementarán herramientas de analítica descriptiva para generar dashboards interactivos que permitan visualizar indicadores clave (KPIs) en tiempo real. Estos paneles incluirán métricas como rotación de inventarios, sell-out por región y segmento, y análisis de productos más vendidos. La visualización de estos datos facilitará la identificación de patrones de comportamiento y oportunidades inmediatas para ajustes en las estrategias comerciales, es lo que se está haciendo en el presente proyecto, sin embargo la obtención de los datos sería en modo automático, reduciendo tiempos y mejorando la calidad de los mismos.

7.3. Evolución hacia la analítica predictiva

Para optimizar la planificación de la demanda, se desarrollarán modelos de machine learning basados en series temporales y variables contextuales. Estos modelos preverán la demanda futura considerando tendencias históricas,

estacionalidad y comportamientos geográficos de los clientes. Esta capacidad permitirá a la fábrica ajustar la producción y distribución de manera más precisa, reduciendo costos por exceso de inventario o desabastecimiento.

7.4. Analítica prescriptiva y toma de decisiones estratégica

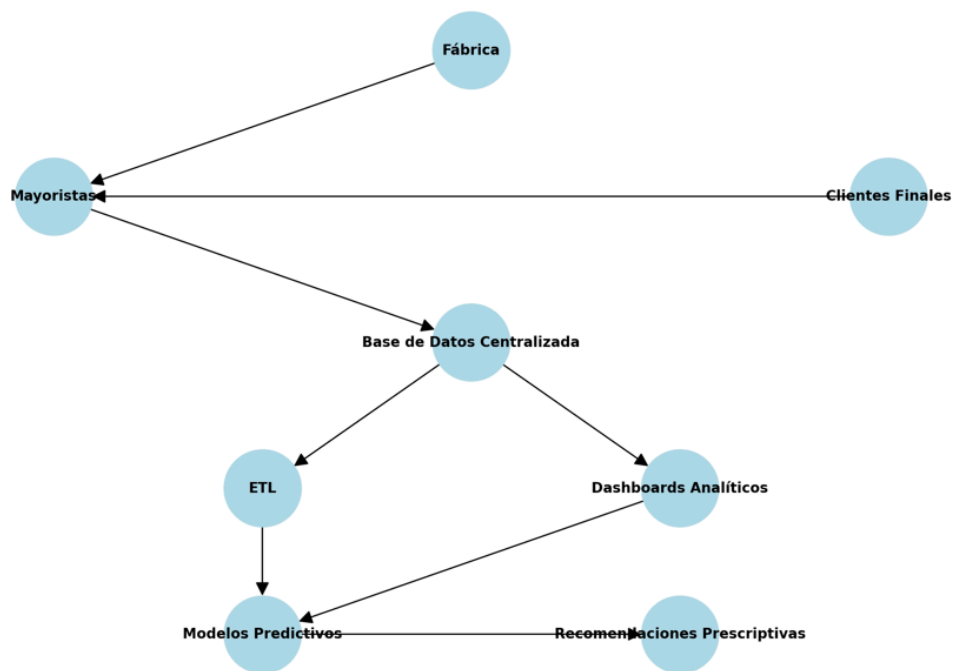
La siguiente etapa será implementar herramientas de analítica prescriptiva que sugieran acciones específicas para maximizar la eficiencia operativa y comercial. Por ejemplo, se crearán recomendaciones automáticas para la asignación óptima de inventarios entre mayoristas, basadas en los datos de rotación y comportamiento del cliente final. Estas recomendaciones estarán alineadas con los objetivos estratégicos de la fábrica, como aumentar la penetración en nuevos mercados o fortalecer relaciones con mayoristas clave.

7.5. Expansión y escalabilidad del modelo

Finalmente, el modelo se diseñará para ser escalable y adaptable a futuras necesidades, como incorporar nuevas líneas de productos, explorar mercados internacionales o conectar con canales de venta online. También se evaluará la implementación de soluciones avanzadas, como H2O.ai, para robustecer los modelos predictivos y prescriptivos. Esto garantizará que el sistema continúe generando valor a medida que la empresa crezca.

7.6. Modelo sugerido para la solución

Figura 16. Modelo de integración y analítica en tiempo real



Fuente: Elaboración propia

El diagrama muestra el flujo de información desde la fábrica y los mayoristas hacia una base de datos centralizada, habilitada con procesos ETL. Este sistema conecta herramientas de visualización (dashboards) con modelos predictivos y prescriptivos, optimizando la toma de decisiones estratégicas en tiempo real. Este enfoque asegura un ciclo continuo de análisis, planificación y mejora operativa.

8. Lecciones aprendidas

1. La importancia de un diseño sólido de bases de datos a lo largo del proceso, quedó claro que una base de datos bien diseñada es el pilar para cualquier análisis exitoso. Garantizar la normalización, la integración de datos heterogéneos y la limpieza de la información permitió sentar las bases para un análisis confiable y preciso.
2. La sinergia entre bases de datos y analítica, Este proyecto evidenció cómo las bases de datos no solo son depósitos de información, sino también habilitadores de la analítica avanzada y la inteligencia artificial. Desde modelos descriptivos hasta predictivos y prescriptivos, las bases de datos sirven como un motor fundamental para impulsar decisiones estratégicas basadas en datos.

3. Adaptación a un mercado dinámico los jugadores del mercado cambian constantemente, lo que exige un sistema flexible y escalable que permita reaccionar a nuevas demandas, comportamientos del cliente y tendencias. Incorporar datos en tiempo real fue clave para mejorar la capacidad de respuesta de la organización.
4. Eficiencia en performance y costos lograr un equilibrio entre un sistema de alto rendimiento y costos controlados fue un desafío clave. A través de la optimización de procesos y herramientas modernas, se logró maximizar la eficiencia operativa sin comprometer el presupuesto del proyecto.
5. Importancia de estar actualizado con las tendencias tecnológicas. El proyecto destacó la necesidad de mantenernos informados sobre las últimas tendencias en bases de datos, analítica e inteligencia artificial. Esto incluyó explorar nuevas herramientas, como H2O.ai, para aprovechar las ventajas competitivas de estas tecnologías y mantener la relevancia en un entorno en constante evolución para poder visualizar los próximos pasos.

9. Bibliografía

- Li, W., Tse, W. K., & Chen, J. (2024). Privacy and Security Mechanisms for B2B Data Sharing: A Conceptual Framework. *Information*, 15(6), 308. <https://doi.org/10.3390/info15060308>
- DCKAP. (2024). Benefits of Centralized Data Management for Decision-Making. Recuperado de <https://www.dckap.com>