PROYECTO DE EVALUACIÓN - DATA ANALYST JR - GRUPO 02

URL DE GITHUB: https://github.com/hral-work/PROYECTO_RSM-KODIGO_DAJ10.git

ALUMNOS:
Katia Elizabeth Martínez - k00002733.
Rafael Alexander Martínez - k00002735.
Javier Antonio Valle - k00002730.
Hugo Robin Aparicio - k00002729.

ENTREGABLES DEL AVANCE 1**:** 21 de noviembre de 2024.
Diseño de base de datos | Extracción y Manipulación de datos.

---

1. GENERALIDADES DEL PROYECTO Y REQUERIMIENTOS

Como proyecto final, los alumnos del bootcamp tienen que realizar un proyecto que consiste en optimizar las ventas de una tienda en línea mediante el análisis de datos, se debe diseñar una base de datos y analizar la información de ventas, clientes y productos.

El proyecto se divide en varias fases, como se indica a continuación:

1. Diseño de Base de Datos:
   Crear un diagrama entidad-relación (ERD), definir atributos clave y relaciones, y transformar el ERD en sentencias SQL para implementar las tablas en un DBMS.

2. Extracción y Manipulación de Datos:
   Importar datos de archivos CSV a la base de datos, realizar validaciones y escribir consultas SQL para obtener información relevante.

3. Análisis Exploratorio de Datos:
   Usar estadísticas descriptivas y técnicas de análisis para identificar patrones, tendencias y anomalías en las ventas.

4. Creación de Dashboard:
   Desarrollar un dashboard interactivo en Power BI para visualizar KPIs, tendencias de ventas, y análisis de cestas de compra.

5. Modelo Predictivo (Opcional):
   Construir y evaluar un modelo de machine learning para prever tendencias de ventas futuras.

6. Reporte y Presentación:
   Preparar un informe escrito y una presentación para resumir los hallazgos, métodos utilizados y recomendaciones basadas en el análisis de datos. Además, se incluyen buenas prácticas de codificación y optimización de consultas SQL, así como la documentación y entrega del proyecto en GitHub.

El objetivo final es proporcionar estrategias basadas en datos para mejorar las ventas del cliente.

2. DATOS RELEVANTES, DESCRIPCION Y REVISION DE LOS INSUMOS

Por tanto, iniciando de lo más básico, en la tabla número 1, se describen los datos y el contenido de los tres Data Set que han sido proporcionados para proyecto (la revisión se hizo explorando el archivo en texto plano).

2.1 Descripción del contenido

Nombre del Data Set	Descripción
Data Set Clientes	Este dataset contiene información sobre los clientes de una empresa, con los siguientes campos: • ClienteID: Identificador único del cliente. • NombreCliente: Nombre del cliente. • Email: Correo electrónico del cliente. • Telefono: Número de teléfono del cliente. • Direccion: Dirección del cliente.
Data Set Productos	Este dataset incluye información sobre los productos ofrecidos por la empresa: • ProductoID: Identificador único del producto. • NombreProducto: Nombre del producto. • Categoria: Categoría a la que pertenece el producto. • PrecioUnitario: Precio unitario del producto.
Data Set Ventas	Este dataset contiene registros de ventas, con los siguientes campos: • VentaID: Identificador único de la venta. • ClienteID: Identificador del cliente que realizó la compra. • ProductoID: Identificador del producto comprado. • Cantidad: Cantidad del producto comprado. • FechaVenta: Fecha en que se realizó la venta. • Region: Región donde se realizó la venta.

Tabla 1, Descripción del contenido de los Data Set del proyecto.

2.2 Relevancia de cada Data Set

Clientes:
Este dataset es crucial para entender quiénes son los clientes de la empresa. Es la base para cualquier análisis de comportamiento del cliente, segmentación de mercado, y estrategias de marketing personalizadas.

Productos:
Es fundamental para el análisis de inventarios, planificación de estrategias de precios, y análisis de ventas por categoría de producto.

Ventas:
Este dataset es esencial para el análisis de rendimiento de ventas, tendencias de compra, y evaluación del impacto de las campañas de marketing.

2.3 Hallazgos Clave y Errores a Revisar

Clientes:
- Duplicados: Verificar si hay clientes duplicados, especialmente en el campo Email o Teléfono.
- Campos Vacíos o Nulos: Revisar que todos los campos estén completos.
- Formatos: Asegurar que los correos electrónicos y números de teléfono tengan el formato correcto.

Productos:
- Precios Negativos o Inválidos: Verificar que todos los precios sean positivos y válidos.
- Categorías Correctas: Asegurar que cada producto esté correctamente categorizado.
- Duplicados: Revisar que no haya productos duplicados.

Ventas:
- Fechas Inválidas: Verificar que todas las fechas sean válidas y en el formato correcto.
- Inconsistencias en IDs: Asegurar que ClienteID y ProductoID correspondan a registros válidos en los otros datasets.
- Regiones Correctas: Verificar que las regiones sean válidas y consistentes.

3. DISEÑO DE LA BASE DE DATOS

3.1 Diagrama Entidad-Relación

Para la creación del diagrama entidad-relación (ERD) con base en los datasets proporcionados ("clientes", "productos" y "ventas"), identificamos las entidades principales, sus atributos, y las relaciones entre ellas. A continuación, se detalla cómo se puede construir el ERD y qué elementos clave se deben considerar.

Entidades y Atributos

1. Clientes
    - ClienteID (llave primaria)
    - NombreCliente
    - Email
    - Telefono
    - Direccion

2. Productos
    - ProductoID (llave primaria)
    - NombreProducto
    - Categoria
    - PrecioUnitario

3. Ventas
    - VentaID (llave primaria)
    - ClienteID (llave foránea)
    - ProductoID (llave foránea)
    - Cantidad
    - FechaVenta
    - Región

Relaciones

1. Relación entre Clientes y Ventas:
    - Un cliente puede tener asociadas muchas ventas.
    - Cardinalidad: Uno a Muchos (1:N) desde Clientes a Ventas.
    - ClienteID en la tabla Ventas es una llave foránea que referencia a ClienteID en la tabla Clientes.

2. Relación entre Productos y Ventas:
    - Un producto puede estar asociado con muchas ventas.
    - Cardinalidad: Uno a Muchos (1:N) desde Productos a Ventas.
    - ProductoID en la tabla Ventas es una llave foránea que referencia a ProductoID en la tabla Productos.

Llaves y Cardinalidad

- Llaves Primarias: ClienteID en Clientes, ProductoID en Productos, y VentaID en Ventas son llaves primarias que identifican de manera única a cada registro en sus respectivas tablas.

- Llaves Foráneas: ClienteID y ProductoID en la tabla Ventas son llaves foráneas que crean relaciones entre las tablas, permitiendo la integridad referencial y conexiones lógicas entre datos.

- Cardinalidad: Definida por las relaciones 1:N, ya que un cliente puede tener múltiples ventas y un producto puede ser vendido en múltiples transacciones.

Diagrama Visual

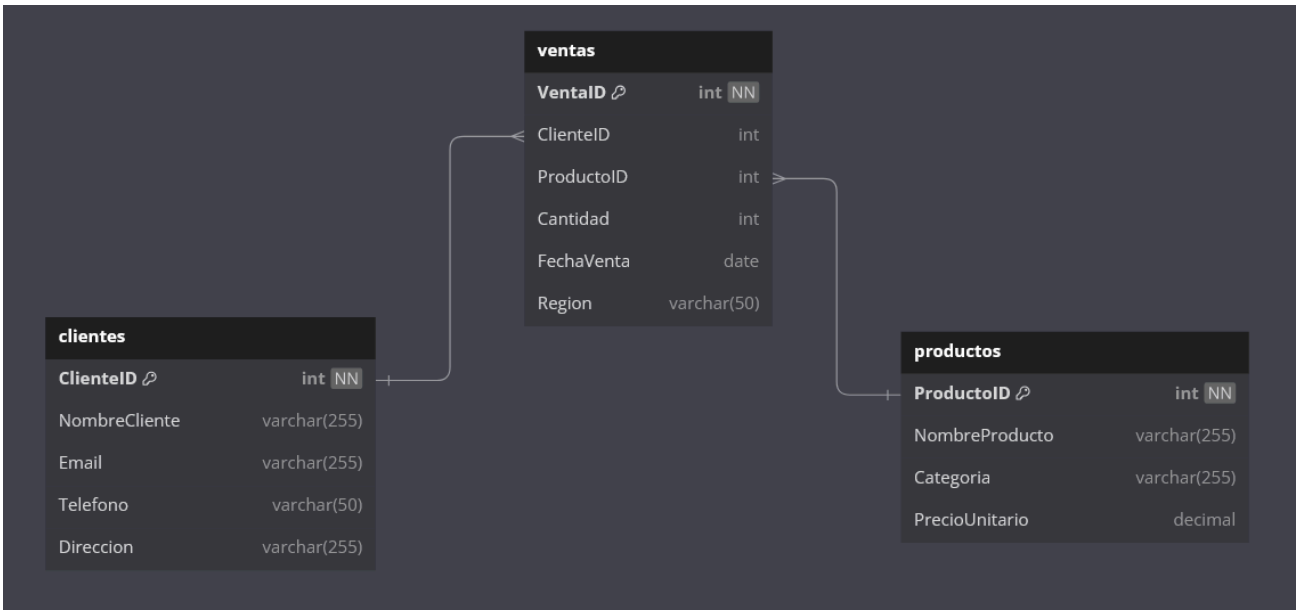

Diagrama 1, diagrama Entidad-Relación.

Para generar el diagrama anterior se utilizó el lenguaje DBML (Database Markup Language)**:**

https://dbdiagram.io/d/PROYECTO_FINAL-673c1b28e9daa85acaea168c

```
Table clientes {
  ClienteID int [pk, not null]
  NombreCliente varchar(255)
  Email varchar(255)
  Telefono varchar(50)
  Direccion varchar(255)
}

Table productos {
  ProductoID int [pk, not null]
  NombreProducto varchar(255)
  Categoria varchar(255)
  PrecioUnitario decimal
}

Table ventas {
  VentaID int [pk, not null]
  ClienteID int [ref: > clientes.ClienteID]
  ProductoID int [ref: > productos.ProductoID]
  Cantidad int
  FechaVenta date
  Region varchar(50)
}
```

3.2 Implementación en el DBMS

Previo al análisis y normalización de datos es necesario que se ejecute la creación de la base de datos y tablas, una vez lista esa parte, se procede a cargar la información de los Datasets en el ambiente habilitado. Se decidió utilizar como motor de base datos SQLServer 2019 por la facilidad que se tiene en cuanto a disponibilidad y acceso al uso de un ambiente de desarrollo.

Los pasos requeridos en esta fase pueden resumirse así:

1. Crear la Base de Datos: Crear la base de datos RSMDB.
2. Crear Usuario y Roles: Crear un usuario rsmuser y asignarle roles de lectura y escritura.
3. Crear las Tablas: Crear las tablas clientes, productos y ventas con sus respectivas relaciones.
4. Verificar: Asegurar que las tablas y relaciones se han creado correctamente mediante consultas de verificación.

SCRIPT creacion_db_user_tablas.sql:

```
-- 1. Creación de la Base de Datos
CREATE DATABASE RSMDB;
GO
```

```
-- 2. Creación del Usuario y Asignación de Permisos
USE RSMDB;
GO

-- Crear usuario
CREATE USER rsmuser WITH PASSWORD = 'password';
GO

-- Crear roles
CREATE ROLE db_datareader;
CREATE ROLE db_datawriter;
GO

-- Asignar permisos al usuario
ALTER ROLE db_datareader ADD MEMBER rsmuser;
ALTER ROLE db_datawriter ADD MEMBER rsmuser;
GO

-- 3. Creación de las Tablas
CREATE TABLE clientes (
  ClienteID int PRIMARY KEY NOT NULL,
  NombreCliente varchar(255),
  Email varchar(255),
  Telefono varchar(50),
  Direccion varchar(255)
);
GO

CREATE TABLE productos (
  ProductoID int PRIMARY KEY NOT NULL,
  NombreProducto varchar(255),
  Categoria varchar(255),
  PrecioUnitario decimal(10, 2)
);
GO

CREATE TABLE ventas (
  VentaID int PRIMARY KEY NOT NULL,
  ClienteID int,
  ProductoID int,
  Cantidad int,
  FechaVenta date,
  Region varchar(50),
  FOREIGN KEY (ClienteID) REFERENCES clientes (ClienteID),
  FOREIGN KEY (ProductoID) REFERENCES productos (ProductoID)
);
GO
```

4. EXTRACCION Y MANIPULACION DE DATOS

4.1 Importación de datos (paso inicial necesario)

SQL Server ofrece un asistente de Importación y Exportación para cargar datos desde archivos CSV a las tablas de SQL Server.

1. Abrir SQL Server Management Studio (SSMS).
2. Conectarse a la instancia de SQL Server.
3. Hacer clic derecho en la base de datos RSMDB y seleccionar Tasks -> Import Data.
4. Elegir el origen de datos:
    - Source: Flat File Source
    - File name: Seleccionamos el archivo CSV correspondiente (clientes.csv, productos.csv, o ventas.csv).
    - Configuramos los delimitadores y otros parámetros según sea necesario.
5. Elegir el destino de datos:
    - Destination: SQL Server Native Client
    - Server name: SQLSRV.
    - Database: RSMDB.
6. Configurar las opciones de mapeo de columnas para asegurar que las columnas de los archivos CSV correspondan correctamente a las columnas de las tablas SQL.
7. Ejecutar el proceso de importación.

Una vez que los datos se hayan importado, verifica que estén correctamente cargados ejecutando algunas consultas básicas en la base de datos:

–Verificar que las tablas estén pobladas
*SELECT * FROM clientes;*
*SELECT * FROM productos;*
*SELECT * FROM ventas;*

Como buenas prácticas, durante y posteriormente a la importación de datos se recomienda aplicar estos controles:

- Limpieza de Datos: Asegúrate de eliminar duplicados, manejar valores nulos y corregir errores en los datos.
- Integridad Referencial: Garantizar que todas las llaves foráneas correspondan a registros válidos en sus tablas de origen.
- Normalización: Mantener una estructura de datos normalizada para evitar redundancias y asegurar un almacenamiento eficiente.
- Indexación: Crear índices en las columnas utilizadas frecuentemente en las consultas para mejorar el rendimiento.

4.2 Consultas SQL para Extracción de Información

La siguientes, son consultas SQL para extraer la información solicitada, basadas en los datos de los datasets y el procedimiento de importación descritos anteriormente:

1. Ventas Totales por Categoría de Producto (ventas_totales_por_categoria.sql)
Esta consulta calcula la suma de las ventas agrupadas por categoría de producto.

```
SELECT
   p.Categoria,
   SUM(v.Cantidad * p.PrecioUnitario) AS VentasTotales
FROM
   ventas v
JOIN
   productos p ON v.ProductoID = p.ProductoID
GROUP BY
   p.Categoria;
```

2. Clientes con Mayor Valor de Compra (clientes_mayor_valor_compra.sql)

Esta consulta calcula el total gastado por cada cliente y ordena los resultados de mayor a menor.

```
SELECT
   c.ClienteID,
   c.NombreCliente,
   SUM(v.Cantidad * p.PrecioUnitario) AS TotalGastado
FROM
   ventas v
JOIN
   clientes c ON v.ClienteID = c.ClienteID
JOIN
   productos p ON v.ProductoID = p.ProductoID
GROUP BY
   c.ClienteID, c.NombreCliente
ORDER BY
   TotalGastado DESC;
```

3. Productos Más Vendidos por Región (productos_mas_vendidos_por_region.sql)

Esta consulta determina los productos más populares en cada región, basándose en la cantidad total vendida.

```
SELECT
   v.Region,
   p.NombreProducto,
   SUM(v.Cantidad) AS CantidadTotalVendida
FROM
   ventas v
JOIN
   productos p ON v.ProductoID = p.ProductoID
GROUP BY
   v.Region, p.NombreProducto
ORDER BY
   v.Region, CantidadTotalVendida DESC;
```

Resultado de consulta ventas_totales_por_categoria.sql:

```sql
-- Esta consulta determina los productos más populares en cada región, basándose en la cantidad total vendida.

SELECT
    v.Region,
    p.NombreProducto,
    SUM(v.Cantidad) AS CantidadTotalVendida
FROM
    ventas v
JOIN
    productos p ON v.ProductoID = p.ProductoID
GROUP BY
    v.Region, p.NombreProducto
ORDER BY
    v.Region, CantidadTotalVendida DESC;
```

100 %

Results Messages

	Region	NombreProducto	CantidadTotalVendida
1	Este	Producto 5	75
2	Este	Producto 6	69
3	Este	Producto 10	69
4	Este	Producto 30	67
5	Este	Producto 15	67
6	Este	Producto 20	65
7	Este	Producto 26	62
8	Este	Producto 4	59
9	Este	Producto 2	59
10	Este	Producto 11	55
11	Este	Producto 9	55
12	Este	Producto 7	54
13	Este	Producto 25	52
14	Este	Producto 27	51
15	Este	Producto 24	50
16	Este	Producto 18	49
17	Este	Producto 13	48
18	Este	Producto 17	48
19	Este	Producto 8	43
20	Este	Producto 22	36
21	Este	Producto 23	35
22	Este	Producto 19	34
23	Este	Producto 29	30
24	Este	Producto 14	25
25	Este	Producto 21	20
26	Este	Producto 16	19
27	Este	Producto 12	18
28	Este	Producto 3	17
29	Este	Producto 1	14
30	Este	Producto 28	5

Query executed successfully.

Resultado de la consulta clientes_mayor_valor_compra.sql:

```sql
--Esta consulta calcula el total gastado por cada cliente y ordena los resultados de mayor a menor.
SELECT
    c.ClienteID,
    c.NombreCliente,
    SUM(v.Cantidad * p.PrecioUnitario) AS TotalGastado
FROM
    ventas v
JOIN
    clientes c ON v.ClienteID = c.ClienteID
JOIN
    productos p ON v.ProductoID = p.ProductoID
GROUP BY
    c.ClienteID, c.NombreCliente
ORDER BY
    TotalGastado DESC;
```

100 %

Results Messages

	ClienteID	NombreCliente	TotalGastado
1	1043	Cliente 43	110127.57
2	1063	Cliente 63	94780.25
3	1005	Cliente 5	85356.86
4	1070	Cliente 70	82935.08
5	1030	Cliente 30	81736.62
6	1095	Cliente 95	81139.11
7	1024	Cliente 24	75599.88
8	1058	Cliente 58	75347.86
9	1075	Cliente 75	72330.36
10	1098	Cliente 98	71991.52
11	1010	Cliente 10	69773.41
12	1061	Cliente 61	68949.07
13	1041	Cliente 41	68728.42
14	1073	Cliente 73	68687.97
15	1022	Cliente 22	68026.40
16	1027	Cliente 27	66659.18
17	1082	Cliente 82	65104.78
18	1035	Cliente 35	63724.44
19	1085	Cliente 85	63545.22
20	1029	Cliente 29	62798.98
21	1006	Cliente 6	61443.68
22	1031	Cliente 31	61443.18
23	1091	Cliente 91	61242.86
24	1062	Cliente 62	61117.13
25	1037	Cliente 37	60240.61
26	1078	Cliente 78	60234.20
27	1066	Cliente 66	60035.61
28	1011	Cliente 11	59817.61
29	1016	Cliente 16	58591.13
30	1039	Cliente 39	57414.60

Query executed successfully.

Resultado de la consulta productos_mas_vendidos_por_region.sql:

SQLQuery1.sql - DW...(kodigo_user (56))*

```sql
-- Esta consulta determina los productos más populares en cada región, basándose en la cantidad total vendida.

SELECT
    v.Region,
    p.NombreProducto,
    SUM(v.Cantidad) AS CantidadTotalVendida
FROM
    ventas v
JOIN
    productos p ON v.ProductoID = p.ProductoID
GROUP BY
    v.Region, p.NombreProducto
ORDER BY
    v.Region, CantidadTotalVendida DESC;
```

100 %

Results | Messages

	Region	NombreProducto	CantidadTotalVendida
1	Este	Producto 5	75
2	Este	Producto 6	69
3	Este	Producto 10	69
4	Este	Producto 30	67
5	Este	Producto 15	67
6	Este	Producto 20	65
7	Este	Producto 26	62
8	Este	Producto 4	59
9	Este	Producto 2	59
10	Este	Producto 11	55
11	Este	Producto 9	55
12	Este	Producto 7	54
13	Este	Producto 25	52
14	Este	Producto 27	51
15	Este	Producto 24	50
16	Este	Producto 18	49
17	Este	Producto 13	48
18	Este	Producto 17	48
19	Este	Producto 8	43
20	Este	Producto 22	36
21	Este	Producto 23	35
22	Este	Producto 19	34
23	Este	Producto 29	30
24	Este	Producto 14	25
25	Este	Producto 21	20
26	Este	Producto 16	19
27	Este	Producto 12	18
28	Este	Producto 3	17
29	Este	Producto 1	14
30	Este	Producto 28	5

Query executed successfully.