

Desafío Alegra BI DEV

Faustino Correa Muñoz



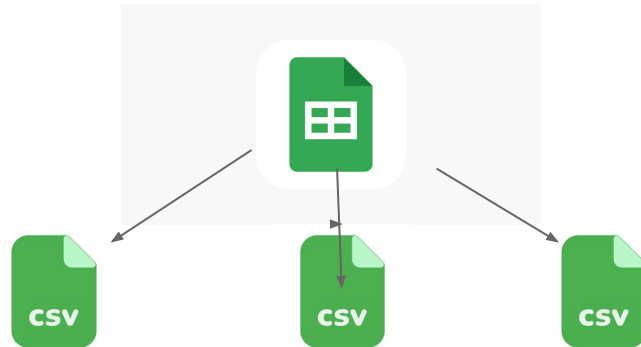
1

Flujo de Trabajo



Transformación usando python

- El archivo original “[BD - Test Business Intelligence Developer](#)” fue transformado usando python y la librería de pandas.
- La transformación consistió en separar las 3 hojas con datos del cuaderno excel en archivos .csv



Implementación en python

```
import pandas as pd
from pathlib import Path

#Constantes
ROOT = Path('.').resolve()
DATA_RAW_PATH = ROOT.joinpath('data', 'raw')
DATA_RAW = str(list(DATA_RAW_PATH.iterdir())[0])
SPLIT_DF = ROOT.joinpath('data', 'split')

def read_excel(file: str, sheet_name: str) → pd.DataFrame:
    df = pd.read_excel(file, sheet_name=sheet_name)
    return df

cliente = read_excel(DATA_RAW, 'Cliente')
descuento = read_excel(DATA_RAW, 'Descuento')
ventas = read_excel(DATA_RAW, 'Ventas')

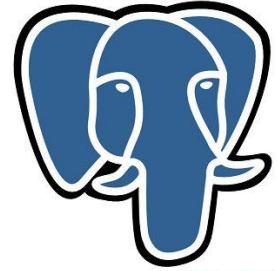
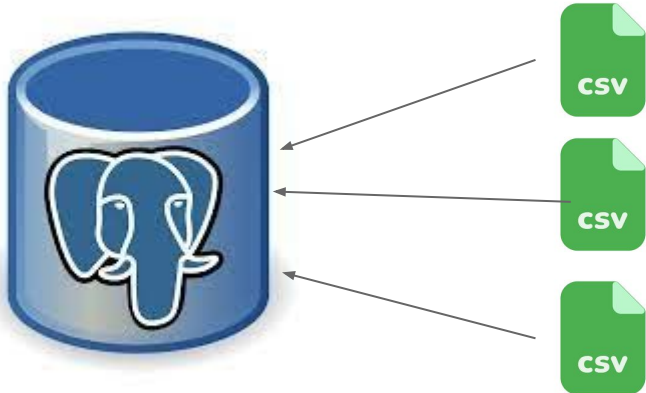
ventas['ID'] = pd.Series([i for i in range(1, len(ventas)+1)])

cliente.to_csv(str(SPLIT_DF)+'/cliente.csv', index=False)
descuento.to_csv(str(SPLIT_DF)+'/descuento.csv', index=False)
ventas.to_csv(str(SPLIT_DF)+'/ventas.csv', index=False)
```



Carga usando PostgreSQL y Pgadmin4

- La carga de archivos se realizó usando comandos del motor de bases de datos PostgreSQL en la plataforma de desarrollo pdAdmin



PostgreSQL



pgAdmin Version 4
Management Tools for PostgreSQL

Creando las tablas dentro de postgresQL

```
CREATE TABLE PUBLIC.cliente(  
    id INTEGER PRIMARY KEY,  
    nombre VARCHAR(50),  
    edad INTEGER,  
    genero VARCHAR(1)  
);
```

```
CREATE TABLE PUBLIC.descuento(  
    id INTEGER PRIMARY KEY,  
    descuento REAL  
);
```

```
CREATE TABLE PUBLIC.ventas(  
    Fecha_de_Venta TIMESTAMP,  
    ClienteID INTEGER,  
    Cantidad INTEGER,  
    Precio_Unitario REAL,  
    DescuentoID INTEGER,  
    Prioridad VARCHAR(20),  
    Medio_de_Envio VARCHAR(20),  
    Segmento VARCHAR(20),  
    Provincia VARCHAR(50),  
    ID INTEGER PRIMARY KEY  
);
```

Insertando datos

```
COPY PUBLIC.cliente
FROM
'C:\Users\Faustino\Documents\GitHub\Desafio-Alegra-BI-DEV\data\split\cliente.
csv'
DELIMITER ','
CSV HEADER;

COPY PUBLIC.descuento
FROM
'C:\Users\Faustino\Documents\GitHub\Desafio-Alegra-BI-DEV\data\split\descuento.
csv'
DELIMITER ','
CSV HEADER;

COPY PUBLIC.ventas| You, hace 11 horas • insert data from csv ...
FROM
'C:\Users\Faustino\Documents\GitHub\Desafio-Alegra-BI-DEV\data\split\ventas.csv'
DELIMITER ','
CSV HEADER;
```

2

Consultas SQL



Pregunta 01: enunciado

- ¿Cuál ha sido el porcentaje de crecimiento de las ventas por año?
Agrupe por año las ventas brutas (precio unitario* cantidad) y las ventas netas (restando los descuentos) por año, obtenga la variación en números absolutos y en tasa de crecimiento respecto al año anterior. También incluya el número total de productos vendidos (suma cantidades) y número de pedidos (ventas únicas) por año.
Nota: La tabla ventas no cuenta aún con id, incluya un índice que le permita usar como identificador para contar las ventas únicas.

Pregunta 01: resultados

Año	Ventas brutas	Ventas netas	total productos vendidos	número de pedidos	Taza de crecimiento ventas netas
2009	4355150	3650890	54380	2153	-
2010	3685916	3073398	54384	2142	-15.82%
2011	3597484	3021678	51564	2002	-1.68%
2012	3868625	3225687	54449	2102	6.75%

Pregunta 01: consulta SQL

```
SELECT      EXTRACT(YEAR FROM VENTAS.fecha_de_venta) AS "Year",  
            ROUND(SUM(cantidad*precio_unitario)) AS "Ventas brutas",  
            ROUND(SUM(cantidad*precio_unitario*(1.-DESCUENTO.descuento))) AS  
            "ventas netas",  
            SUM(cantidad) AS "total productos vendidos",  
            COUNT(VENTAS.ID) AS "Número de pedidos"  
FROM        VENTAS  
LEFT JOIN   DESCUENTO  
ON          VENTAS.descuentoid = DESCUENTO.id  
GROUP BY   EXTRACT(YEAR FROM VENTAS.fecha_de_venta)  
ORDER BY   EXTRACT(YEAR FROM VENTAS.fecha_de_venta);
```

Pregunta 02: enunciado

Se está pensando darle mayor atención a los medios de envío con mayor prioridad. Identifique el medio de envío con mayor cantidad de pedidos con Prioridad High o Critical, y compare la cantidad de pedidos y valores totales de este medio de envío respecto al resto de medios, obteniendo los resultados en porcentajes. Este análisis se desea hacer para la provincia con mayor de número de ventas (cantidad de ventas únicas).

Pregunta 02: resultados

Medio de Envío	Pedidos	Pedidos %	Ventas netas
Regular Air	540	72.78%	625036
Delivery Truck	113	15.23%	543399
Express Air	89	11.99%	72672

Pregunta 02: consulta SQL

```
SELECT medio_de_envio, COUNT(VENTAS.ID) AS "pedidos",  
       ROUND(SUM(cantidad*precio_unitario*(1.-DESCUENTO.descuento))) AS  
       "ventas netas"  
FROM VENTAS  
LEFT JOIN DESCUENTO  
ON VENTAS.descuentoid = DESCUENTO.id  
WHERE provincia = (  
    SELECT provincia  
    FROM VENTAS  
    GROUP BY provincia  
    ORDER BY COUNT(ID) DESC  
    LIMIT 1) AND prioridad IN ('High', 'Critical')  
GROUP BY medio_de_envio  
ORDER BY "pedidos" DESC  
;
```

You, hace 9 horas • question2

Pregunta 03: enunciado

La empresa quiere analizar las pérdidas por descuentos realizados. Por cada "Segmento" que atiende la empresa muestre el dinero perdido por descuentos, además el porcentaje que representa respecto al valor total de las ventas sin descuentos ($\text{valor total descuentos} / \text{valor total de ventas sin descuentos}$) y el número de ventas únicas que se han atendido por segmento. Por tratarse de un análisis muy específico, sólo debe considerar en la consulta los hombres (M) desde 30 años en adelante. Ordene su tabla por el número de ventas mostrando primero el segmento con mayor número de éstas. Puede utilizar subconsultas.

Pregunta 03: resultado

Segmento	Pérdidas por descuento	Ratio pérdidas	Número de pedidos
Corporate	464640	17%	1435
Home Office	230892	16%	927
Consumer	220328	15%	756
Small Business	222259	17%	753

Pregunta 03: consulta SQL

```
SELECT segmento AS "Segmento",  
    ROUND(SUM(cantidad*precio_unitario*DESCUENTO.descuento)) AS "perdidas por  
    descuento",  
    ROUND(100.*SUM(cantidad*precio_unitario*DESCUENTO.descuento)/SUM  
    (cantidad*precio_unitario)) AS "ratio",  
    COUNT(VENTAS.ID) AS "Número de pedidos"  
FROM VENTAS  
LEFT JOIN DESCUENTO  
ON VENTAS.descuentoid = DESCUENTO.id  
LEFT JOIN CLIENTE  
ON CLIENTE.id = VENTAS.clienteid  
WHERE CLIENTE.edad ≥ 30 AND CLIENTE.genero = 'M'  
GROUP BY segmento  
ORDER BY "Número de pedidos" DESC  
;|
```

You, hace 9 horas • question 3

Consulta 04: enunciado

La empresa desea premiar a sus mejores clientes durante el último año, 2012 , como estrategia de retención. Por ello, define los siguientes obsequios.

Obsequio A: El mejor cliente. El cliente con la compra más alta (quien realizó la compra neta con mayor valor).

Obsequio B: Cliente más fiel. El cliente con mayor cantidad de compras únicas.

Obsequio C: Clientes estrellas. Ranking con los 3 clientes que dejaron más ganancias.

El obsequio consiste en un bonus para comprar cualquier producto de la tienda, en función del total de sus compras (valor neto) durante el 2012.

Pregunta 04: resultado obsequio A

ID cliente	Nombre	Edad	Género	venta neta	Obsequio A
16	Annie Cyprus	57	F	37985	1086

Pregunta 04: consulta SQL obsequio A

```
SELECT      CLIENTE.ID, CLIENTE.nombre, CLIENTE.edad, CLIENTE.genero,
            ROUND(cantidad*precio_unitario*(1.-DESCUENTO.descuento)) AS "venta
            neta"
FROM VENTAS
LEFT JOIN DESCUENTO
ON VENTAS.descuentoid = DESCUENTO.id
LEFT JOIN CLIENTE
ON CLIENTE.id = VENTAS.clienteid
WHERE EXTRACT(YEAR FROM VENTAS.fecha_de_venta) = 2012
ORDER BY "venta neta" DESC
LIMIT 1
;

SELECT      ROUND(SUM(cantidad*precio_unitario*(1.-DESCUENTO.descuento)))*0.01
AS "obsequio A"
FROM VENTAS
LEFT JOIN DESCUENTO
ON VENTAS.descuentoid = DESCUENTO.id
LEFT JOIN CLIENTE
ON CLIENTE.id = VENTAS.clienteid
WHERE EXTRACT(YEAR FROM VENTAS.fecha_de_venta) = 2012 AND VENTAS.clienteid=16
GROUP BY VENTAS.clienteid
;| You, hace 1 segundo • Uncommitted changes
```

Pregunta 04: resultado obsequio B

ID cliente	Nombre	Edad	Género	Total de compras	Obsequio B
17	Grant Carroll	42	M	65	2009

Pregunta 04: consulta SQL obsequio B

```
SELECT      clienteid AS "id",
            COUNT(VENTAS.ID) AS "Total_compras",
            ROUND(SUM(cantidad*precio_unitario*(1.-DESCUENTO.descuento))*0.02)
            AS "obsequio_B"
FROM VENTAS
LEFT JOIN DESCUENTO
ON VENTAS.descuentoid = DESCUENTO.id
LEFT JOIN CLIENTE
ON CLIENTE.id = VENTAS.clienteid
WHERE EXTRACT(YEAR FROM VENTAS.fecha_de_venta) = 2012
GROUP BY clienteid
ORDER BY "Total_compras" DESC
LIMIT 1;

SELECT CLIENTE.ID, CLIENTE.nombre, CLIENTE.edad, CLIENTE.genero
FROM CLIENTE
WHERE CLIENTE.ID = 17;
```

Pregunta 04: resultado obsequio C

ID cliente	Nombre	Edad	Género	Total compras	Obsequio C
12	Sylvia Foulston	22	F	150423	6017
13	Jim Radford	37	M	128599	5144
27	Don Jones	47	M	114781	4591

Pregunta 04: consulta SQL obsequio C

```
SELECT      clienteid AS "id",
            ROUND(SUM(cantidad*precio_unitario*(1.-DESCUENTO.descuento))) AS
            "Total_compras",
            ROUND(SUM(cantidad*precio_unitario*(1.-DESCUENTO.descuento))*0.04)
            AS "obsequio_C"
FROM VENTAS
LEFT JOIN DESCUENTO
ON VENTAS.descuentoid = DESCUENTO.id
LEFT JOIN CLIENTE
ON CLIENTE.id = VENTAS.clienteid
WHERE EXTRACT(YEAR FROM VENTAS.fecha_de_venta) = 2012
GROUP BY clienteid
ORDER BY "Total_compras" DESC
LIMIT 3;

SELECT CLIENTE.ID, CLIENTE.nombre, CLIENTE.edad, CLIENTE.genero
FROM CLIENTE
WHERE CLIENTE.ID IN (13, 12, 27);
```




Conclusiones

- Puedes encontrar todo el código utilizado en el repositorio de Github:
- <https://github.com/FaustinoAndres/Desafio-Alegra-BI-DEV>



¡Gracias por la atención!

¿te quedaste con alguna pregunta?

 faustino.correa.m@gmail.com