Prueba Técnica

Objetivo: **Asignación de retos comerciales para la variable de GCAR del negocio empresarial por gerente comercial y análisis de tamaño comercial.**

Tener en cuenta:

- El reto para la variable de GCAR de la Vicepresidencia Empresarial para 2024 es un crecimiento del 15% adicional vs el año anterior.
- La Vicepresidencia Empresarial consta de 8 zonas y 78 gerentes comerciales.
- En la hoja datos se encuentran los datos codificados de la GCAR y tamaño comercial por gerente comercial y zona de enero 2022 a diciembre 2023. El dato es generado mensualmente, es decir, no se encuentra acumulado.
- En la hoja Zonas códigos se encuentra la relación del código de zona y el nombre de la zona. En la hoja Rubros códigos se encuentra el código del rubro y la descripción del mismo.
- Para encontrar el valor del reto de GCAR se debe acumular los resultados del año 2023 y aplicar la tasa de crecimiento para el año 2024.
- Recordar que el TC se acumula como el promedio de los meses y la GCAR en suma. El cierre de cada año es acumulado de enero a diciembre.

Actividad 1: Responda las siguientes preguntas:

1. Cuál fue el gerente con mayor crecimiento en su Tamaño comercial entre 2022 y 2023? **//** el gerente con mayor crecimiento comercial es el **9116** con un crecimiento de **67.817,44**
2. Cuánto fue el crecimiento de la zona Antioquia 1 en GCAR entre 2022 y 2023? **//** el crecimiento de la zona Antioquia 1 en GCAR fue de **993.99**
3. Cuál fue la zona con menor GCAR en el segundo semestre de 2022 y la de mayor tamaño comercial en todo 2023? **//** la zona de menor GCAR es la zona **22300** "VP Empresas zona Centro" y la zona **22220** "VP Empresas zona Bogotá 2" fue la de mayor tamaño comercial en el 2023

In [1]:
```
# Importamos las libreria pandas para trabajar con los datos y responder las preguntas de la actividad 1
import pandas as pd
pd.options.display.float_format = '{:.2f}'.format
```

In [2]:
```
# Importamos los datos necesarios del archivo de excel para resolver las preguntas de la actividad 1

df_datos = pd.read_excel(r'Prueba Bancolombia 2024 (analitico).xlsx', sheet_name='Datos')
df_rubros_codigos = pd.read_excel(r'Prueba Bancolombia 2024 (analitico).xlsx', sheet_name='Rubros codigos')
df_zonas_codigos = pd.read_excel(r'Prueba Bancolombia 2024 (analitico).xlsx', sheet_name='Zonas códigos')
```

Creamos 2 columnas para saber los acumulados del 2022 y 2023 siguiendo las reglas de la actividad 1, donde se menciona que el acumulado de GCAR se suma y el de TC se promedia

In [3]:
```
df_datos.loc[df_datos['cod_rubro'] == 95999, 'acumulado_2022'] = df_datos.loc[df_datos['cod_rubro'] == 95999].filter(like='2022').sum(axis=1)
df_datos.loc[df_datos['cod_rubro'] == 96450, 'acumulado_2022'] = df_datos.loc[df_datos['cod_rubro'] == 96450].filter(like='2022').mean(axis=1)

df_datos.loc[df_datos['cod_rubro'] == 95999, 'acumulado_2023'] = df_datos.loc[df_datos['cod_rubro'] == 95999].filter(like='2023').sum(axis=1)
df_datos.loc[df_datos['cod_rubro'] == 96450, 'acumulado_2023'] = df_datos.loc[df_datos['cod_rubro'] == 96450].filter(like='2023').mean(axis=1)
```

In [4]:
```
# Creamos una columna para saber el crecimiento restando los totales del 2023 - 2022

df_datos['Crecimiento'] = df_datos['acumulado_2023'] - df_datos['acumulado_2022']
```

Podemos ver las columnas **"acumulado_2022"**, **"acumulado_2023"** y **"Crecimiento"** creadas al final del DataFrame omitiendo los datos mensuales

In [5]:
```
df_datos.loc[: , ['zona', 'gerente', 'cod_rubro', 'acumulado_2022', 'acumulado_2023', 'Crecimiento']]
```

Out[5]:

	zona	gerente	cod_rubro	acumulado_2022	acumulado_2023	Crecimiento
0	22110	9102	95999	6893.76	7932.29	1038.53
1	22110	9125	95999	7983.04	8027.06	44.02
2	22110	9166	95999	9072.92	7917.03	-1155.89
3	22110	9184	95999	7894.10	8752.69	858.59
4	22110	9198	95999	8232.61	6980.49	-1252.12
...	...	...	...	...	...	...
147	22500	9179	96450	136708.79	160233.37	23524.58
148	22500	9410	96450	121289.90	117929.00	-3360.90
149	22500	9411	96450	187765.81	192539.09	4773.28
150	22500	9901	96450	122541.93	125084.67	2542.74
151	22500	9903	96450	135007.40	125604.27	-9403.13

152 rows × 6 columns

1. Cuál fue el gerente con mayor crecimiento en su Tamaño comercial entre 2022 y 2023?

Para responder esta pregunta inicialmente Nos fijamos en el dataFrame de los rubros para identificar el codigo del Tamaño comercial

In [6]:
```
df_rubros_codigos
```

Out[6]:

	descri_rubro	cod_rubro
0	GCAR	95999
1	Tamaño comercial	96450

Una vez hecho esto, solo filtramos el codigo 96450 y agrupamos por gerente

In [7]:
```
df_temp = df_datos.loc[df_datos['cod_rubro'] == 96450].groupby('gerente').agg({'acumulado_2022': 'sum', 'acumulado_2023': 'sum', 'Crecimiento': '
```

Finalmente ordenamos los datos de mayor a menor utilizando la columna **"Crecimiento"**, para llegar a la conclusion de que el gerente con mayor crecimiento comercial es el 9116 con un crecimiento de 67.817,44

In [8]:
```
df_temp.sort_values(by='Crecimiento', ascending=False)
```

Out[8]:

	gerente	acumulado_2022	acumulado_2023	Crecimiento
1	9116	240593.26	308410.70	67817.44
75	9922	234209.13	272292.36	38083.23
35	9218	99092.75	134862.52	35769.76
38	9274	158486.95	190276.23	31789.29
19	9160	138616.89	168133.90	29517.01
...	...	...	...	...
70	9909	62137.05	51351.09	-10785.96
50	9334	132056.62	120047.58	-12009.04
46	9310	207948.68	195434.89	-12513.79
11	9144	203038.00	188870.57	-14167.43
14	9149	114581.07	98697.03	-15884.04

76 rows × 4 columns

2. Cuánto fue el crecimiento de la zona Antioquia 1 en GCAR entre 2022 y 2023?

Inicialmente para resolver esta pregunta identificamos el codigo de la zona Antioquia 1 del dataFrame df_zonas_codigos el cual corresponde a 22110

In [9]:
```
df_zonas_codigos
```

Out[9]:

	zona	Desc
0	22110	VP Empresas zona Antioquia 1
1	22120	VP Empresas zona Antioquia 2
2	22210	VP Empresas zona Bogotá 1
3	22210	VP Empresas zona Bogotá 1
4	22220	VP Empresas zona Bogotá 2
5	22230	VP Empresas zona Bogotá 3
6	22300	VP Empresas zona Centro
7	22400	VP Empresas zona Sur
8	22500	VP Empresas zona Caribe

Buscamos el codigo de GCAR en el dataFrame df_rubros_codigos el cual corresponde a 95999

In [10]:
```
df_rubros_codigos
```

Out[10]:

	descri_rubro	cod_rubro
0	GCAR	95999
1	Tamaño comercial	96450

filtramos la zona 22110 y el codigo de rubro 95999 y sumamos los totales y el crecimiento para concluir que el crecimiento de la zona Antioquia 1 en GCAR fue de 993.99

In [11]:
```
df_temp = df_datos.loc[(df_datos['cod_rubro'] == 95999) & (df_datos['zona'] == 22110)].groupby('zona').agg({'acumulado_2022': 'sum', 'acumulado_2
df_temp
```

Out[11]:

	zona	acumulado_2022	acumulado_2023	Crecimiento
0	22110	58943.38	59937.37	993.99

Calculamos el porcentaje de crecimiento equivalente a 1.69%

In [12]:
```
df_temp['Porcentaje_Crecimiento'] = (df_temp['Crecimiento'] / df_temp['acumulado_2022']) * 100
df_temp['Porcentaje_Crecimiento']
```

Out[12]:
```
0   1.69
Name: Porcentaje_Crecimiento, dtype: float64
```

3. Cuál fue la zona con menor GCAR en el segundo semestre de 2022 y la de mayor tamaño comercial en todo 2023?

Iniciamos verificando cual fue la zona con menor GCAR, para esto creamos una nueva columna que nos sume el GCAR para el segundo semestre del 2022

In [13]:
```
df_datos['semestre2_2022'] = df_datos.filter(like='2022').iloc[:, 6:12].sum(axis=1)
df_datos.loc[: , [ 'gerente', 'zona', 'cod_rubro', 'acumulado_2022', 'acumulado_2023', 'Crecimiento', 'semestre2_2022']].head(3)
```

Out[13]:

	gerente	zona	cod_rubro	acumulado_2022	acumulado_2023	Crecimiento	semestre2_2022
0	9102	22110	95999	6893.76	7932.29	1038.53	3943.64
1	9125	22110	95999	7983.04	8027.06	44.02	3858.64
2	9166	22110	95999	9072.92	7917.03	-1155.89	5023.96

Posteriormente filtramos el codigo 95999 correspondiente al GCAR, agrupamos por la zona y ordenamos de menor a mayor para encontrar la zona de menor GCAR, y concluir que la zona de menor crecimiento es la zona 22300 "VP Empresas zona Centro"

In [14]:
```
df_datos.loc[df_datos['cod_rubro'] == 95999].groupby('zona').agg({'semestre2_2022': 'sum'}).reset_index().sort_values(by='semestre2_2022', ascend
```

Out[14]:

	zona	semestre2_2022
5	22300	31539.28
0	22110	32105.78
6	22400	33722.31
7	22500	35164.94
1	22120	37358.50
4	22230	41425.59
2	22210	47342.14
3	22220	50881.84

Para encontrar la zona de mayor tamaño comercial en todo 2023 filtramos el codigo 96450, y agrupamos por zona sumando la columna "acumulado_2023" que calculamos al comienzo. Se concluye que la zona 22220 "VP Empresas zona Bogotá 2" fue la de mayor tamaño comercial en el 2023.

In [15]:
```
df_datos.loc[df_datos['cod_rubro'] == 96450].groupby('zona').agg({'acumulado_2023': 'sum'}).reset_index().sort_values(by='acumulado_2023', ascend
```

Out[15]:

	zona	acumulado_2023
3	22220	1518174.07
1	22120	1440314.17
2	22210	1368384.32
4	22230	1295494.78
0	22110	1276101.99
6	22400	1270682.69
7	22500	1198621.00
5	22300	1116107.31

Actividad 2: En la hoja Retos 2024 asigne los retos en GCAR para cada gerente a Diciembre de 2024, y responda las siguientes preguntas en el word que anexará:

1. Qué técnicas de modelación considera pueden ser pertinentes para asignar los retos:
2. Cuál de ellas utilizó y por qué?
3. Qué herramientas conoce y usó para realizar esta prueba?
4. Asumiendo que las hojas de Datos, Rubros códigos y Zonas códigos, son tablas en zonas de resultados (tablas sql), diseñe un query que las combine y genere el resultado agregado el nombre de la zona, el rubro y el valor de cierre del Tamaño comercial para la zona del 2023. (Dejar expresadas las sentencias de SQL (pseudocódigo) en la respuesta)

Asignación en la hoja Retos 2024

Para comenzar importamos la hoja 'Retos 2024' desde el excel

In [16]:
```
df_retos_2024 = pd.read_excel(r'Prueba Bancolombia 2024 (analitico).xlsx', sheet_name='Retos 2024')
df_retos_2024
```

Out[16]:

	zona	gerente	GCAR 2024
0	VP Empresas zona Antioquia 1	9102	NaN
1	VP Empresas zona Antioquia 1	9125	NaN
2	VP Empresas zona Antioquia 1	9166	NaN
3	VP Empresas zona Antioquia 1	9184	NaN
4	VP Empresas zona Antioquia 1	9198	NaN
...	...	...	...
74	VP Empresas zona Sur	9203	NaN
75	VP Empresas zona Sur	9307	NaN
76	VP Empresas zona Sur	9310	NaN
77	VP Empresas zona Sur	9334	NaN
78	VP Empresas zona Sur	9915	NaN

79 rows × 3 columns

Creamos un nuevo dataFrame apartir del df_datos, solo con los datos de GCAR realizando el filtro por el codigo de rubro 95999 y solo las columnas necesarias

In [17]:
```
df_datos_GCAR = df_datos.loc[df_datos['cod_rubro'] == 95999].copy()
df_datos_GCAR = df_datos_GCAR.loc[: , ['zona', 'gerente', 'cod_rubro', 'acumulado_2023']]
df_datos_GCAR
```

Out[17]:

	zona	gerente	cod_rubro	acumulado_2023
0	22110	9102	95999	7932.29
1	22110	9125	95999	8027.06
2	22110	9166	95999	7917.03
3	22110	9184	95999	8752.69
4	22110	9198	95999	6980.49
...	...	...	...	...
71	22500	9179	95999	7952.17
72	22500	9410	95999	6145.22
73	22500	9411	95999	8266.54
74	22500	9901	95999	5567.41
75	22500	9903	95999	8080.75

76 rows × 4 columns

Creamos una columna con el reto para el 2024, siendo igual al acumulado del 2023 en GCAR + un 15%

In [18]:
```
df_datos_GCAR['GCAR 2024'] = df_datos_GCAR['acumulado_2023'] * 1.15
df_datos_GCAR
```

Out[18]:

	zona	gerente	cod_rubro	acumulado_2023	GCAR 2024
0	22110	9102	95999	7932.29	9122.13
1	22110	9125	95999	8027.06	9231.12
2	22110	9166	95999	7917.03	9104.59
3	22110	9184	95999	8752.69	10065.59
4	22110	9198	95999	6980.49	8027.56
...	...	...	...	...	...
71	22500	9179	95999	7952.17	9145.00
72	22500	9410	95999	6145.22	7067.01
73	22500	9411	95999	8266.54	9506.52
74	22500	9901	95999	5567.41	6402.52
75	22500	9903	95999	8080.75	9292.86

76 rows × 5 columns

creamos una columna con la llave de cruce para unir los datos de los retos 2024 con los datos de GCAR 2024, la llave es la concatenación del código de la zona y el gerente

In [19]:
```
df_datos_GCAR['llave'] = df_datos_GCAR['zona'].astype(str) + df_datos_GCAR['gerente'].astype(str)
df_datos_GCAR
```

Out[19]:

	zona	gerente	cod_rubro	acumulado_2023	GCAR 2024	llave
0	22110	9102	95999	7932.29	9122.13	221109102
1	22110	9125	95999	8027.06	9231.12	221109125
2	22110	9166	95999	7917.03	9104.59	221109166
3	22110	9184	95999	8752.69	10065.59	221109184
4	22110	9198	95999	6980.49	8027.56	221109198
...	...	...	...	...	...	...
71	22500	9179	95999	7952.17	9145.00	225009179
72	22500	9410	95999	6145.22	7067.01	225009410
73	22500	9411	95999	8266.54	9506.52	225009411
74	22500	9901	95999	5567.41	6402.52	225009901
75	22500	9903	95999	8080.75	9292.86	225009903

76 rows × 6 columns

Para crear la misma llave en el dataFrame df_retos_2024 primero debemos obtener el código de la zona

In [20]:
```
df_zonas_codigos
```

Out[20]:

	zona	Desc
0	22110	VP Empresas zona Antioquia 1
1	22120	VP Empresas zona Antioquia 2
2	22210	VP Empresas zona Bogotá 1
3	22210	VP Empresas zona Bogotá 1
4	22220	VP Empresas zona Bogotá 2
5	22230	VP Empresas zona Bogotá 3
6	22300	VP Empresas zona Centro
7	22400	VP Empresas zona Sur
8	22500	VP Empresas zona Caribe

Para esto realizamos un Left Join con el campo descripción de la zona del df_zonas_codigos

In [21]:
```
df_temp = pd.merge(df_retos_2024, df_zonas_codigos, how='left', left_on='zona', right_on='Desc')
df_temp.rename(columns={'zona_y': 'Zona'}, inplace=True)
```

```
df_temp = df_temp.loc[: , ['Zona', 'Desc', 'gerente', 'GCAR 2024']].drop_duplicates(subset=['Zona', 'Desc', 'gerente'])
df_temp
```

Out[21]:

	Zona	Desc	gerente	GCAR 2024
0	22110	VP Empresas zona Antioquia 1	9102	NaN
1	22110	VP Empresas zona Antioquia 1	9125	NaN
2	22110	VP Empresas zona Antioquia 1	9166	NaN
3	22110	VP Empresas zona Antioquia 1	9184	NaN
4	22110	VP Empresas zona Antioquia 1	9198	NaN
...	...	...	...	...
86	22400	VP Empresas zona Sur	9203	NaN
87	22400	VP Empresas zona Sur	9307	NaN
88	22400	VP Empresas zona Sur	9310	NaN
89	22400	VP Empresas zona Sur	9334	NaN
90	22400	VP Empresas zona Sur	9915	NaN

79 rows × 4 columns

Ahora que tenemos el código de la zona creamos la llave concatenando el codigo con el gerente

In [22]:
```
df_temp['llave'] = df_temp['Zona'].astype(str) + df_temp['gerente'].astype(str)
df_temp
```

Out[22]:

	Zona	Desc	gerente	GCAR 2024	llave
0	22110	VP Empresas zona Antioquia 1	9102	NaN	221109102
1	22110	VP Empresas zona Antioquia 1	9125	NaN	221109125
2	22110	VP Empresas zona Antioquia 1	9166	NaN	221109166
3	22110	VP Empresas zona Antioquia 1	9184	NaN	221109184
4	22110	VP Empresas zona Antioquia 1	9198	NaN	221109198
...	...	...	...	...	...
86	22400	VP Empresas zona Sur	9203	NaN	224009203
87	22400	VP Empresas zona Sur	9307	NaN	224009307
88	22400	VP Empresas zona Sur	9310	NaN	224009310
89	22400	VP Empresas zona Sur	9334	NaN	224009334
90	22400	VP Empresas zona Sur	9915	NaN	224009915

79 rows × 5 columns

Realizamos un left Join entre el DataFrame df_datos_GCAR y el df_temp, para rellenar el reto GCAR 2024 para cada gerente

In [23]:
```
df_retos_2024 = pd.merge(df_temp, df_datos_GCAR, how='left', on='llave').loc[: , ['Desc', 'gerente_x', 'GCAR 2024_y']].rename(columns={'gerente_x
df_retos_2024
```

Out[23]:

	Desc	gerente	GCAR 2024
0	VP Empresas zona Antioquia 1	9102	9122.13
1	VP Empresas zona Antioquia 1	9125	9231.12
2	VP Empresas zona Antioquia 1	9166	9104.59
3	VP Empresas zona Antioquia 1	9184	10065.59
4	VP Empresas zona Antioquia 1	9198	8027.56
...	...	...	...
74	VP Empresas zona Sur	9203	6976.83
75	VP Empresas zona Sur	9307	11140.37
76	VP Empresas zona Sur	9310	8254.52
77	VP Empresas zona Sur	9334	7068.49
78	VP Empresas zona Sur	9915	5326.18

79 rows × 3 columns

Nota: estos 5 gerentes y zonas quedaron sin reto asignado debido a que en la hoja datos no esta la información del 2022 y 2023 para poder calcular el reto para el 2024.

In [24]:
```
df_retos_2024.loc[df_retos_2024['GCAR 2024'].isna()]
```

Out[24]:

	Desc	gerente	GCAR 2024
7	VP Empresas zona Antioquia 1	9298	NaN
12	VP Empresas zona Antioquia 2	9210	NaN
51	VP Empresas zona Caribe	9142	NaN
56	VP Empresas zona Caribe	9189	NaN
72	VP Empresas zona Sur	9182	NaN

Se exporta el dataFrame a excel

In [25]:
```
df_retos_2024.to_excel('Retos 2024.xlsx', index=False)
```

1. Qué técnicas de modelación considera pueden ser pertinentes para asignar los retos:

Las tecnicas de modelación que pueden ser pertinentes para el congunto de datos suministrado son:

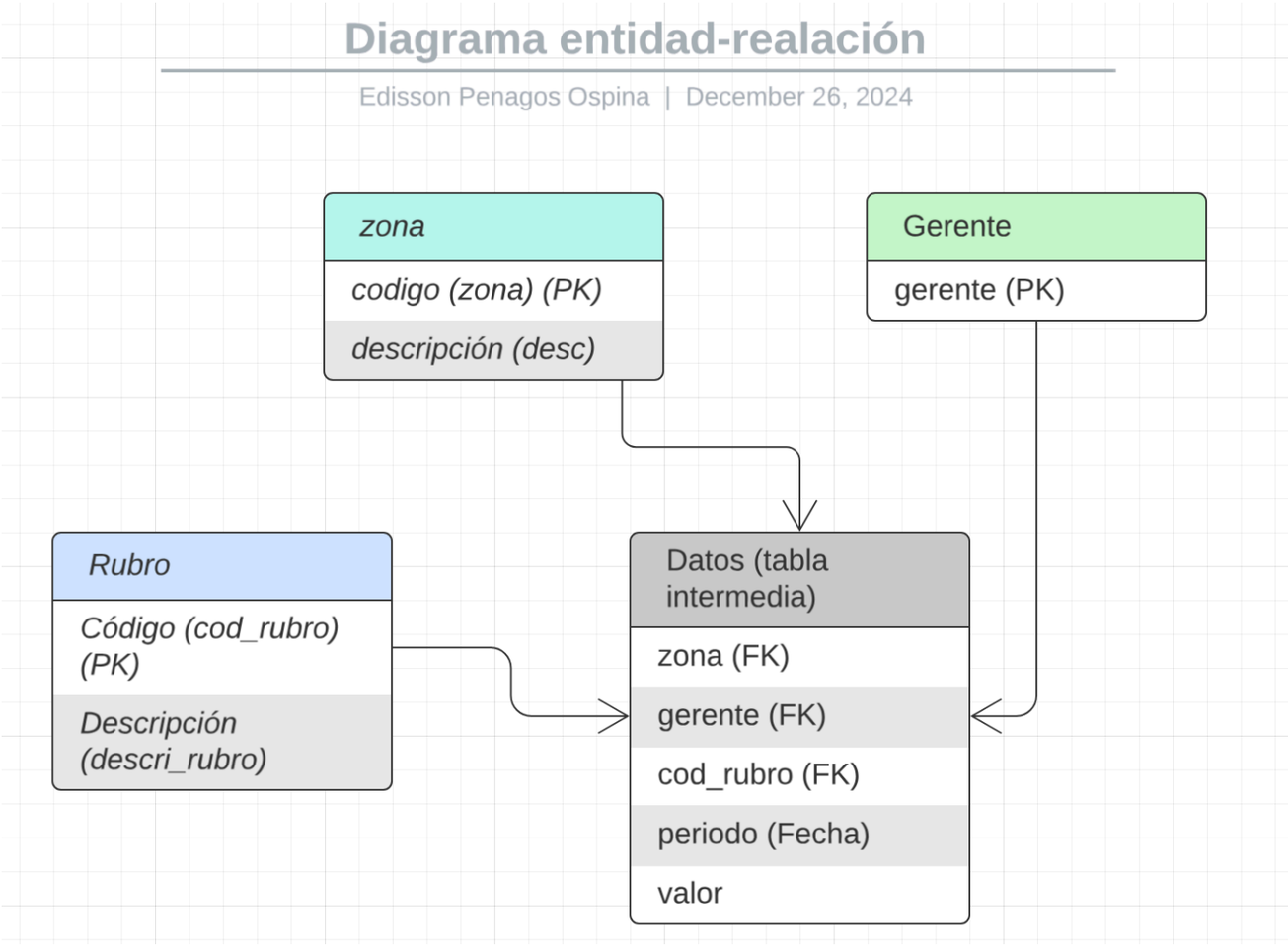
- Modelado de datos entidad-relación

- Modelado de datos orientado a objetos
- Modelado de datos como series de tiempo

2. Cuál de ellas utilizó y por qué?

Se utilizo el modelado de datos entidad relación, debido a que este es el mas comun para modelar datos estructurados como los suministrados, aunque se pudo usar perfectamente el enfoque como series de tiempo debido a la periodicidad de los datos de manera mensual

Se realizo un diagrama entidad realcion simple para entender como se relacionaban los datos de las diferentes hojas



3. Qué herramientas conoce y usó para realizar esta prueba?

Las herramientas que utilice para realizar esta prueba fueron excel, python con la libreria pandas y jupyter notebook. Ademas de git y github para guardar el proyecto.

4. Asumiendo que las hojas de Datos, Rubros códigos y Zonas códigos, son tablas en zonas de resultados (tablas sql), diseñe un query que las combine y genere el resultado agregado el nombre de la zona, el rubro y el valor de cierre del Tamaño comercial para la zona del 2023. (Dejar expresadas las sentencias de SQL (pseudocódigo) en la respuesta)

Para ser lo mas exactos posible con la query, voy a subir la información de las hojas Datos, Rubros códigos y Zonas códigos a la LZ a la zona de procesos

Para esto voy a utilizar sparky_bc para subir las tablas y impala helper para realizar consultas SQL

In [26]:
```
import pandas as pd

# Se importa la clase Helper del módulo helper.helper
from helper.helper import Helper

# Se importa la clase Sparky del módulo sparky_bc
from sparky_bc import Sparky

# Se instancia el sparky reemplazando 'nombre_usuario' por tu usuario y 'dsn' por el DSN configurado en la maquina
# (dsn a partir de la version 2.1.0 de la libreria)
sp = Sparky('epenago', 'IMPALA_PROD', hostname = "sbmdeblze004.bancolombia.corp")

# Se instancia el helper
ih = Helper(dsn="IMPALA_PROD")
```

```
2024-12-27 09:38:49 - [WARNING] - No se encontro la carpeta "c:\Users\epenago\Documents\Prueba_técnica\prueba_tecnica_analitico\logs" para guarda
r los logs
```

```
 ___  _____ __   __  ___ _____ _____
|  _ \| ____|  \/  |/ _ \_   _| ____|
| |_) |  _| | |\/| | | | || | |  _|
|  _ <| |___| |  | | |_| || | | |___
|_| \_\_____|_|  |_|\___/ |_| |_____|

 ___ ___  _   ___ _  _
/ ___||  _ \ / \  |  _ \| |/ /
\___ \| |_) / _ \ | |_) | ' /
 ___) |  __/ ___ \|  _ <| . \
|____/|_| /_/   \_\_| \_\_|\_\
```

```
2024-12-27 09:39:06 - [WARNING] - No se encontro la carpeta "c:\Users\epenago\Documents\Prueba_técnica\prueba_tecnica_analitico\logs" para guarda
r los logs
```

```
 ___ __ __ ____  _   _       _
|_ _|  \/  |  _ \ / \  | |     / \
 | || |\/| | |_) / _ \ | |    / _ \
 | || |  | |  __/ ___ \| |___ / ___ \
|___|_|  |_|_| /_/   \_\_____/_/   \_\

 _   _ _____ _     ____  _____ ____
| | | | ____| |   |  _ \| ____|  _ \
| |_| |  _| | |   | |_) |  _| | |_) |
|  _  | |___| |___|  __/| |___|  _ <
|_| |_|_____|_____|_|   |_____|_| \_\
```

Cargo de nuevo las hojas Datos, Rubros códigos y Zonas códigos, para evitar subir a la LZ las modificaciones que se realizaron para resolver las preguntas anteriores

In [27]:
```
df_datos = pd.read_excel(r'Prueba Bancolombia 2024 (analitico).xlsx', sheet_name='Datos')
df_rubros_codigos = pd.read_excel(r'Prueba Bancolombia 2024 (analitico).xlsx', sheet_name='Rubros codigos')
df_zonas_codigos = pd.read_excel(r'Prueba Bancolombia 2024 (analitico).xlsx', sheet_name='Zonas códigos')
```

Antes de subirse a la lz se eliminan los duplicados en el df_zonas_codigos

In [28]:
```
df_zonas_codigos.drop_duplicates(subset=['Desc'], inplace=True)
df_zonas_codigos.reset_index(drop=True, inplace=True)
df_zonas_codigos
```

Out[28]:

	zona	Desc
0	22110	VP Empresas zona Antioquia 1
1	22120	VP Empresas zona Antioquia 2
2	22210	VP Empresas zona Bogotá 1
3	22220	VP Empresas zona Bogotá 2
4	22230	VP Empresas zona Bogotá 3
5	22300	VP Empresas zona Centro
6	22400	VP Empresas zona Sur
7	22500	VP Empresas zona Caribe

In [29]:
```
# Subir tabla de pandas (DataFrame to LZ)
ih.ejecutar_consulta("DROP TABLE IF EXISTS proceso_vsa.datos_PT PURGE")
sp.subir_df(df_datos, 'proceso_vsa.datos_PT', 'proceso_vsa')

ih.ejecutar_consulta("DROP TABLE IF EXISTS proceso_vsa.rubros_codigos_PT PURGE")
sp.subir_df(df_rubros_codigos, 'proceso_vsa.rubros_codigos_PT', 'proceso_vsa')

ih.ejecutar_consulta("DROP TABLE IF EXISTS proceso_vsa.zonas_codigos_PT PURGE")
sp.subir_df(df_zonas_codigos, 'proceso_vsa.zonas_codigos_PT', 'proceso_vsa')
```

```
2024-12-27 09:39:09 - [INFO] - Transcurrido: 1735310349, Tiempo de Refresco = 1000
------------------------------------------------------------------------
  i  tipo        nombre           estado     hora_inicio   duracion
------------------------------------------------------------------------
 1/1 DROP proceso_vsa.datos_PT   finalizado   09:39:11 AM     00:08.5
------------------------------------------------------------------------
------------------------------------------------------------------------
  i   tipo          nombre           estado     hora_inicio   duracion
------------------------------------------------------------------------
 1/1 DF A LZ proceso_vsa.datos_PT   ejecutando   09:39:19 AM
2024-12-27 09:39:26 - [INFO] - Intento 1 de 3
 1/1 DF A LZ proceso_vsa.datos_PT     subir_lz   09:39:19 AM
2024-12-27 09:40:19 - [INFO] - Transcurrido: 1735310419, Tiempo de Refresco = 1000
 1/1 DF A LZ proceso_vsa.datos_PT   finalizado   09:39:19 AM     06:37.7
------------------------------------------------------------------------
------------------------------------------------------------------------------
  i  tipo           nombre               estado     hora_inicio   duracion
------------------------------------------------------------------------------
 2/2 DROP proceso_vsa.rubros_codigos_PT   finalizado   09:45:57 AM     00:01.4
------------------------------------------------------------------------------
------------------------------------------------------------------------------
  i   tipo             nombre               estado     hora_inicio   duracion
------------------------------------------------------------------------------
 2/2 DF A LZ proceso_vsa.rubros_codigos_PT   ejecutando   09:45:58 AM
2024-12-27 09:46:06 - [INFO] - Intento 1 de 3
 2/2 DF A LZ proceso_vsa.rubros_codigos_PT   finalizado   09:45:58 AM     05:23.1
------------------------------------------------------------------------------
------------------------------------------------------------------------------
  i  tipo           nombre               estado     hora_inicio   duracion
------------------------------------------------------------------------------
 3/3 DROP  proceso_vsa.zonas_codigos_PT   finalizado   09:51:21 AM     00:00.5
------------------------------------------------------------------------------
------------------------------------------------------------------------------
  i   tipo             nombre               estado     hora_inicio   duracion
------------------------------------------------------------------------------
 3/3 DF A LZ  proceso_vsa.zonas_codigos_PT   ejecutando   09:51:22 AM
2024-12-27 09:51:30 - [INFO] - Intento 1 de 3
 3/3 DF A LZ  proceso_vsa.zonas_codigos_PT   finalizado   09:51:22 AM     02:12.6
------------------------------------------------------------------------------
```

In [30]:
```
consulta = """
SELECT
    z.`Desc` AS zona_nombre,
    z. zona,
    r.descri_rubro AS rubro_nombre,
    (SUM(d.`202301`) + SUM(d.`202302`) + SUM(d.`202303`) + SUM(d.`202304`) +
     SUM(d.`202305`) + SUM(d.`202306`) + SUM(d.`202307`) + SUM(d.`202308`) +
     SUM(d.`202309`) + SUM(d.`202310`) + SUM(d.`202311`) + SUM(d.`202312`)) / 12 AS valor_cierre_2023
FROM
    proceso_vsa.datos_PT d
JOIN
    proceso_vsa.zonas_codigos_PT z ON d.zona = z.zona
JOIN
    proceso_vsa.rubros_codigos_PT r ON d.cod_rubro = r.cod_rubro
WHERE
    d.cod_rubro = 96450
GROUP BY
    Z. zona, z.`Desc`, r.descri_rubro;"""
```

```
df = ih.obtener_dataframe(consulta)
df
```

```
-------------------------------------------------------------------------------
  i    tipo               nombre                estado     hora_inicio   duracion
-------------------------------------------------------------------------------
 4/4 DATAFRAME                                descargando   09:53:35 AM
2024-12-27 09:53:37 - [INFO] - 8 filas, 4 columnas, 00:01.2 consultando, 00:00.3 descargando, 00:00.0 convirtiendo
 4/4 DATAFRAME                                finalizado    09:53:35 AM     00:01.7
-------------------------------------------------------------------------------
```

Out[30]:

	zona_nombre	zona	rubro_nombre	valor_cierre_2023
0	VP Empresas zona Antioquia 2	22120	Tamaño comercial	1440314.17
1	VP Empresas zona Sur	22400	Tamaño comercial	1270682.69
2	VP Empresas zona Bogotá 3	22230	Tamaño comercial	1295494.78
3	VP Empresas zona Bogotá 1	22210	Tamaño comercial	1368384.32
4	VP Empresas zona Caribe	22500	Tamaño comercial	1198621.00
5	VP Empresas zona Centro	22300	Tamaño comercial	1116107.31
6	VP Empresas zona Bogotá 2	22220	Tamaño comercial	1518174.07
7	VP Empresas zona Antioquia 1	22110	Tamaño comercial	1276101.99

Actividad 3:

El Vicepresidente del Negocio de Independientes y su Director, al recibir por parte del equipo de analítica de negocios los retos de sus vicepresidencias regionales manifiestan inquietudes e incoherencias frente a la definición del reto y la distribución del mismo tanto en Tamaño comercial como en Gestión Comercial Ajustada por Riesgos. Programa un espacio para revisar con ambos la razón de las inconsistencias, incoherencias y pasos a seguir.

Los calculos realizados para corregir las inconsistencias se encuentran en el archivo de excel en la hoja **Datos_Actividad3**

```
Inicialmente se analizan los datos para detectar si realmente hay alguna incoherencia en la asignación de los retos, antes de la reunion
```

Para el reto GCAR se identifica que los valores que se asignaron para cada region no coinsiden con el reto global, siendo el valor del reto para el 2024 de **861.943,72** y el de la suma de las regiones de **869.528,72** presentando una diferencia de **7.585,00**

```
Para corregir esta inconsistencia se recalculan los valores y los porcentajes de los retos para cada region, ademas de colocar las cifras de los retos mensuales sin acumularse
```

Para el reto del tamaño comercial se identifica que tanto los porcentajes de la region como los valores estan correctos, la diferencia radica es en el reto mensual propuesto por el equipo analitico, que no concuerda con el global siendo este por un valor de **21.017.318,9** y el asignado mensual por **20.135.687** teniendo una diferencia de **881.631,5**

```
Para corregir esta inconsistencia se calculan de nuevo los retos mensuales teniendo en cuenta el historico de datos
```

Agenda de la Reunión (60 minutos)

Introducción (5 minutos):

- Propósito: Identificar y resolver inconsistencias en los retos de GCAR y Tamaño Comercial.
- Escuchar al vicepresidente y su director sobre los motivos de las inconformidades.

Análisis de las inconsistencias (25 minutos):

1. Reto GCAR (12 minutos):
   - Presentación de los valores globales y regionales.
   - Discusión sobre la diferencia de 7.585,00 y sus posibles causas.
2. Reto Tamaño Comercial (13 minutos):
   - Revisión de los porcentajes y su coherencia.
   - Evaluación de la diferencia mensual de 881.631,5 y análisis de su impacto.

Propuesta de soluciones (15 minutos):

- Recalculo de valores y porcentajes de los retos GCAR para cada región.
- Ajuste de los retos mensuales en Tamaño Comercial con base en datos históricos.
- Validación de las soluciones propuestas por los participantes.

Definición de pasos a seguir (10 minutos):

- Aprobación de los ajustes necesarios.
- Establecimiento de un calendario para implementar los cambios.
- Asignación de responsables para las acciones definidas.

Cierre y conclusiones (5 minutos):

- Resumen de los acuerdos alcanzados.
- Confirmación de los próximos pasos y fechas de seguimiento.

Actividad 4:

Teniendo en cuenta el siguiente diagrama entidad relación y los conceptos básicos de la teoría básica de conjuntos y cálculo relacional, construya o genere las consultas SQL respectivas de acuerdo a las siguientes preguntas:

1. Cuáles son las ventas de cada uno de los productos vendidos por categoría y por cada uno de los vendedores, indique aquí los nombres de estado civil sexo y tipo de identificación de cada vendedor en la consulta.
2. Cuáles son los productos que han tenido mayor venta y a qué vendedor pertenece?
3. Construya una consulta general que involucre todas las tablas del Modelo Relacional y permita visualizar totales en ella.

Nota: Dejar expresadas las sentencias de SQL (pseudocódigo) en la respuesta ya que no se cuenta con tablas de datos para este enunciado.

1. Ventas de cada producto por categoría y por vendedor

Esta consulta muestra las ventas agrupadas por categoría, producto y vendedor, incluyendo detalles del vendedor como estado civil, sexo y tipo de identificación.

In [ ]:
```
SELECT
    TblVendedor.Nombre1 AS NombreVendedor,
    TblVendedor.Nombre2 AS SegundoNombre,
    TblVendedor.Apellido1,
    TblVendedor.Apellido2,
    TblConceptoDetalle.Descripcion AS EstadoCivil,
    CASE
        WHEN TblVendedor.Sexo = 1 THEN 'Masculino'
        ELSE 'Femenino'
    END AS Sexo,
    TblConcepto.Descripcion AS TipoIdentificacion,
    TblCategoria.Descripcion AS Categoria,
    TblProducto.Nombre AS Producto,
    SUM(TblDetalleVenta.Cantidad) AS CantidadVendida,
    SUM(TblDetalleVenta.Total) AS TotalVenta
FROM
    TblVendedor
INNER JOIN
    TblVenta ON TblVendedor.Identificacion = TblVenta.Vendedor
INNER JOIN
    TblDetalleVenta ON TblVenta.IdFactura = TblDetalleVenta.IdFactura
INNER JOIN
    TblProducto ON TblDetalleVenta.IdProducto = TblProducto.IdProducto
INNER JOIN
    TblCategoria ON TblProducto.IdCategoria = TblCategoria.IdCategoria
INNER JOIN
    TblConceptoDetalle ON TblVendedor.EstadoCivil = TblConceptoDetalle.IdDetalleConcepto
INNER JOIN
    TblConcepto ON TblVendedor.TipoDeIdentificacion = TblConcepto.IdConcepto
GROUP BY
    TblVendedor.Nombre1, TblVendedor.Nombre2, TblVendedor.Apellido1, TblVendedor.Apellido2,
    TblConceptoDetalle.Descripcion, TblVendedor.Sexo, TblConcepto.Descripcion, TblCategoria.Descripcion, TblProducto.Nombre
ORDER BY
    TblCategoria.Descripcion, TblVendedor.Nombre1, TblProducto.Nombre;
```

2. Productos con mayor venta y su vendedor

Esta consulta identifica los productos con mayores ventas totales y el vendedor asociado.

In [ ]:
```
SELECT
    TOP 1 WITH TIES
    TblProducto.Nombre AS Producto,
    TblVendedor.Nombre1 AS NombreVendedor,
    TblVendedor.Apellido1 AS ApellidoVendedor,
    SUM(TblDetalleVenta.Cantidad) AS CantidadVendida,
    SUM(TblDetalleVenta.Total) AS TotalVenta
FROM
    TblDetalleVenta
INNER JOIN
    TblVenta ON TblDetalleVenta.IdFactura = TblVenta.IdFactura
INNER JOIN
    TblProducto ON TblDetalleVenta.IdProducto = TblProducto.IdProducto
INNER JOIN
    TblVendedor ON TblVenta.Vendedor = TblVendedor.Identificacion
GROUP BY
    TblProducto.Nombre, TblVendedor.Nombre1, TblVendedor.Apellido1
ORDER BY
    SUM(TblDetalleVenta.Total) DESC;
```

3. Consulta general que involucra todas las tablas con totales

Esta consulta combina datos de todas las tablas en el modelo relacional para mostrar detalles completos y totales.

In [ ]:
```
SELECT
    TblVenta.IdFactura AS Factura,
    TblVenta.Fechas AS FechaVenta,
    TblVendedor.Nombre1 AS NombreVendedor,
    TblVendedor.Apellido1 AS ApellidoVendedor,
    TblConcepto.Descripcion AS TipoIdentificacion,
    TblProducto.Nombre AS Producto,
    TblProducto.Descripcion AS DescripcionProducto,
    TblCategoria.Descripcion AS Categoria,
    TblDetalleVenta.Cantidad AS CantidadVendida,
    TblDetalleVenta.Total AS TotalProducto,
    TblVenta.Iva AS IVA,
    SUM(TblDetalleVenta.Total + TblVenta.Iva) OVER (PARTITION BY TblVenta.IdFactura) AS TotalFactura
FROM
    TblVenta
INNER JOIN
    TblVendedor ON TblVenta.Vendedor = TblVendedor.Identificacion
INNER JOIN
    TblDetalleVenta ON TblVenta.IdFactura = TblDetalleVenta.IdFactura
INNER JOIN
    TblProducto ON TblDetalleVenta.IdProducto = TblProducto.IdProducto
INNER JOIN
    TblCategoria ON TblProducto.IdCategoria = TblCategoria.IdCategoria
INNER JOIN
    TblConceptoDetalle ON TblVendedor.EstadoCivil = TblConceptoDetalle.IdDetalleConcepto
INNER JOIN
    TblConcepto ON TblVendedor.TipoDeIdentificacion = TblConcepto.IdConcepto
ORDER BY
    TblVenta.IdFactura, TblVendedor.Nombre1, TblProducto.Nombre;
```

Preguntas bonus

1. ¿Cuáles son algunas estrategias que usarías para optimizar el rendimiento de consultas SQL en grandes conjuntos de datos?

Para optimizar el rendimiento seria enfatico en el uso de las buenas practicas

A nivel de creación de tablar y bases de datos:

- Normalización: Diseña tablas bien normalizadas para evitar redundancias y mantener la consistencia de los datos
- Particionamiento: Divide grandes tablas en particiones horizontales o verticales para distribuir la carga de trabajo.

A nivel de consultas:

- Seleccionar solo las columnas necesarias: Evita el uso de SELECT *; selecciona únicamente las columnas requeridas.
- Filtrar eficientemente: Usa cláusulas WHERE para limitar los datos procesados.

2. Se tiene como resultado de un modelo de Clustering para clientes del segmento Personas un total de 12 grupos, donde existen 3 grupos que abarcan aproximadamente el 53% de la muestra tenida en cuenta. ¿Sí se requieren tener grupos más balanceados, qué metodologías utilizarías para balancear la composición de cada Cluster?

Cluster	Composición
1	1,748
2	3,496
3	5,244
4	6,992
5	8,740
20	34,959
22	38,455
24	41,951
9	15,732
10	17,480
11	19,228
12	20,976

Para tratar de balancear los clusters:

- Iniciaria tratande a de ajustar los Parámetros en el Algoritmo de Clustering
- De ser posible modificar el número de grupos combinar clústeres pequeños que tengan características similares.
- Modificando la muestra, reduciendo el tamaño de los clústeres más grandes (undersampling) o Aumentar el tamaño de los clústeres más pequeños (oversampling)