Ejercicio Data Engineering

Para desenvolverse de forma ágil en el puesto, poder manejarse con Python y, en general, ser capaz de manipular datos con SQL resultan habilidades esenciales.

Se proponen una serie de ejercicios que nos permitan entender un poco más de qué manera se encararía la solución de algunos requerimientos que pueden surgir en el día a día, se recomienda entregar un script de manera de poder pasarle parámetros al ejecutarlo de la misma manera que sucedería en un entorno productivo.

Ejercicio 1 - Manejo de datos (sintaxis Redshift)

Ejercicio 1 - Manejo de Datos (Usando sintaxis Redshift)

Esquema de Tablas:

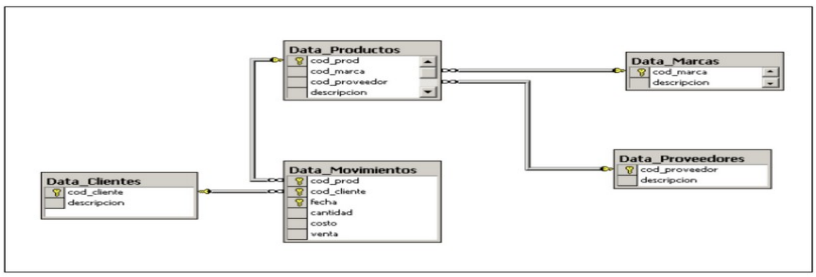
Data_Productos
Cod_Prod (Numérico)
Cod_Marca (Numérico)
Cod_Proveedor (Numérico)
Descripcion (Texto)

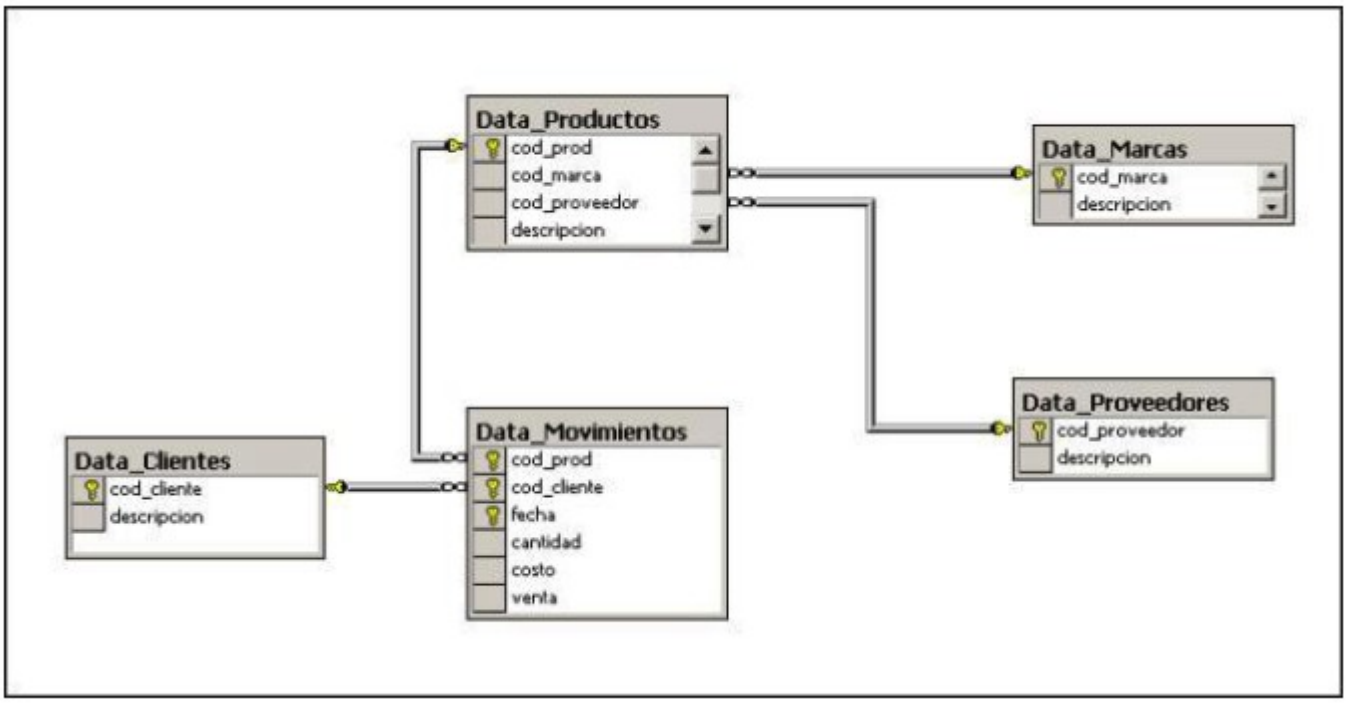
Data_Marcas
Cod_Marca (Numérico)
Descripcion (Texto)

Data_Proveeedores
Cod_Proveedor (Numérico)
Descripcion (Texto)

Data_Clientes
Cod_Cliente (Numérico)
Descripcion (Texto)

Data_Movimientos
Cod_Prod (Numérico)
Cod_Cliente (Numérico)
Fecha (DateTime)
Cantidad (Numérico)
Costo (Numérico)
Venta (Numérico)





Nota:
En la tabla *Data_Movimientos*, el campo *Venta* es el precio total de la venta, por lo que costo/cantidad da el costo unitario de cada producto en cada operación, y venta/cantidad da el precio de venta unitario en cada operación.

a) Crear la tabla “movimientos”, con el listado de todos los Movimientos, con el siguiente contenido : . Fecha . Descripción de Cliente . Descripción de Proveedor . Descripción de Producto . Descripción de Marca . Cantidad . Costo . Venta . Ganancia Neta

b) En base a la tabla generada en a), consultar, ordenando por fecha y descripción del cliente: . fecha, descripción de cliente y ganancia de las primeras 3 operaciones.

c) Dado que las empresas en la base de datos pertenecen en su mayoría a un rubro con una alta curva de aprendizaje, es usual que las empresas no tengan bien calculados los costos y presenten ganancia negativa durante sus primeras operaciones.

Genere una consulta que devuelva las marcas con pérdidas en cada una de sus primeras 3 operaciones

Una consulta con las que tuvieron pérdidas en sus primeras tres operaciones pero no en la cuarta.

Ejercicio 2 - Python

1) El área de marketing acordó con una empresa de publicidad la disponibilización diaria de un archivo tsv y encoding UTF-16LE con los datos de aquellas personas que se comunicaron para adquirir los productos de Etermax. Lamentablemente para usted, los ingenieros de datos de la empresa en cuestión están dando sus primeros pasos con AWS, por lo que el path donde se encuentre el archivo siempre sera *'data/datos_data_engineer.tsv'* pero bucket donde lo disponibilizan puede variar entre ejecuciones.Desde marketing le encargaron que descargue el archivo en cuestión y lo vuelva a subir al mismo bucket pero al path *"nombre-apellido/python1/parsed.csv'* haciendo las correcciones que usted considere necesarias. Es indispensable que el archivo se encuentre en formato csv y utilice un pipe (|) como separador. Para esta primer ejecución, el bucket a utilizar tiene el siguiente arn : arn:aws:s3:::etermax-challenge-engineering
2) Se acerca la fiesta de fin de año y el equipo de marketing desea saber la preferencia de los empleados de Etermax respecto a sus cervezas preferidas. Para eso le encargan extraer información de la siguiente api, a fin de poder realizar la planificación: Punk API - Documentation . Se le solicita que suba al mismo bucket del caso anterior pero al path *"nombre-apellido/python2/"*, un archivo json que contenga la siguiente información:
    2.a)id
    2.b)name
    2.c)first_brewed
    2.d)abv
    2.e)ibu
    2.f)ph
    2.g)principal malt (malta de mayor peso en kg dentro de las presentes entre los ingredientes)
    2.h)ph_type: en función del ph, clasificar las cervezas de la siguiente manera:
        2.h.i)base: si tiene ph<7
        2.h.ii)neutral: ph=7
        2.h.iii)sour: ph>7
    2.i)alcohol_type:
        2.i.i)strong: abv mayor a 7
        2.i.ii)medium: abv entre 5 y 7
        2.i.iii)low: abv menor a 5

No le informaron la cantidad de cervezas que debe extraer, por lo tanto debe variabilizar

la cantidad de registros.

Opcional:

El equipo de marketing no logra decidirse acerca de donde le es más cómodo tener la data disponible. Debido a esto, le solicitan levantar una instancia de una base de datos postgres en un contenedor local y cargar el resultado del punto dos en la tabla en cuestión. En caso

de querer resolver este ejercicio, debe modificar el archivo del punto dos para que permita elegir el destino final de la data (bucket o bbdd).

Restricciones:

- Se puede usar la librería que le parezca más conveniente siempre que esté justificado su uso.
- Queda a su criterio cuáles son las buenas prácticas a seguir y que debe o no hacer en el presente desafío.
- Los ejercicios 1 y 2 de Python deben estar resueltos cada uno en un script (sin contar módulos o archivos/librerías auxiliares).
- Los ejercicios 1 y 2 de Python no pueden ser resueltos con un notebook.
- Se debe agregar un Readme describiendo cómo ejecutar sus soluciones.

Credenciales necesarias para acceder al bucket:

aws_access_key_id = AKIAW7Y3QSZOB53T7B5Q
aws_secret_accces_key = 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

Teórica:

1. Si una instancia de Redshift utilizada para reporting se está quedando sin espacio y se impone la necesidad de sacar algunos datos antiguos de la base, pero a pesar de que los datos de más de seis meses de antigüedad no se utilicen para reporting, se los requiere para entrenar y validar modelos predictivos, además de hacer algunos análisis ad-hoc en SQL **a un precio razonable considerando tanto infraestructura como costos de consultas** ¿Que tipo de solución propondría para poder consultar los datos usando servicios cloud en AWS? ***Intentar ser lo más descriptivo posible.***
2. Dada su experiencia usando los servicios de Amazon, el equipo de ingeniería le solicita que diseñe la arquitectura del proceso que va desde la recepción del archivo en el bucket todos los días a las 8 am, la subida del archivo parseado en el bucket y también la carga en una base de datos disponible en AWS. Asuma todo lo que considere necesario.
3. Suponiendo que usted usa Redshift y que la tabla Data_Movimientos del ejercicio es una tabla externa (alojada no en Redshift sino en S3) creada con la siguiente sentencia:

```
CREATE EXTERNAL TABLE my_spectrum.Data_Movimientos(
Cod_Prod integer,
Cod_Cliente integer,
Cantidad integer,
Costo decimal(8,2),
Venta decimal(8,2)
)
PARTITIONED BY (Fecha char(8))   -- YYYYMMDD

ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
STORED AS textfile
```

LOCATION 's3://bucket/spectrum/Data_Movimientos/‘

y responde a la siguiente consulta: SELECT * FROM my_sepectrum.Data_Movimientos

WHERE cast(substring(Fecha, 5, 2) as int) >= 4 AND cast(substring(Fecha, 5, 2) as int) <= 10

¿Sugeriría algún cambio a la definición de la tabla o de la consulta, para que la query fuese más performante? ¿Cuál? ¿Por qué?