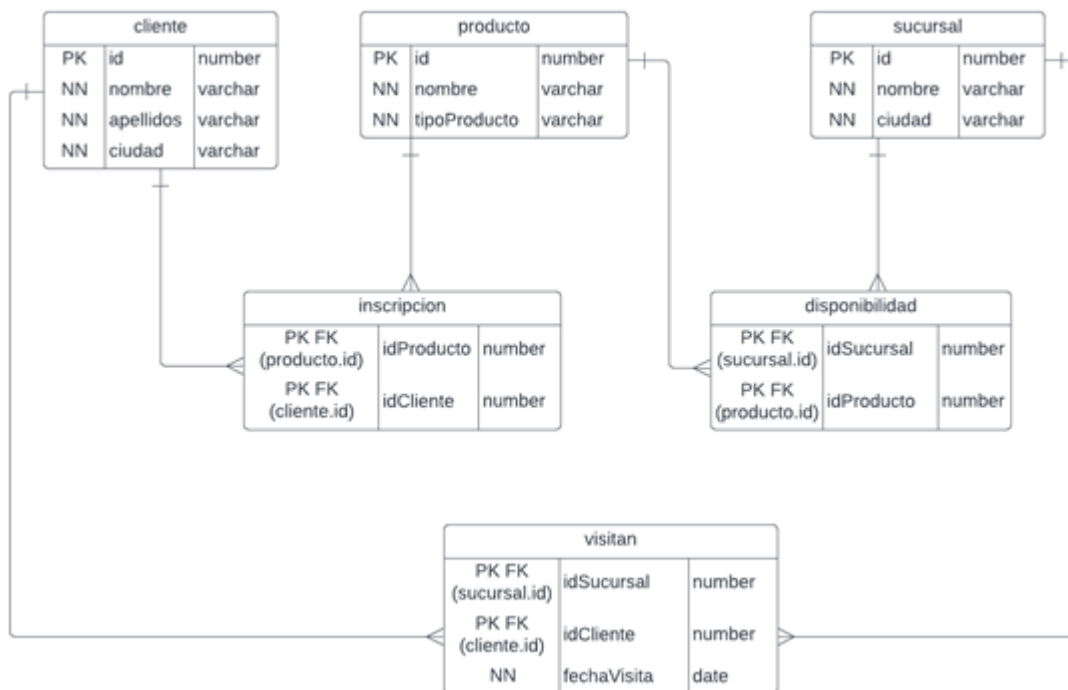


Ejercicio 1

Escriba las sentencias SQL en la creación del modelo relacional y las consultas correspondientes de la forma más eficiente posible, para ello, tenga en cuenta la base de datos llamada “BTG” la cual representa un banco que ofrece diferentes productos en sus sucursales, las cuales visitan los clientes para inscribirse a dichos productos. (tenga en cuenta que se puede presentar el caso en que no todas las sucursales ofrecen los mismos productos).

1. Listar los clientes que han visitado más de 3 sucursales.
2. Obtener los productos que están disponibles en todas las sucursales.
3. Listar las fechas de visita en que un cliente visitó una sucursal que no es de su ciudad.
4. Obtener los nombres de los clientes los cuales tienen inscrito algún producto disponible sólo en las sucursales que visitan.
5. ¿Qué recomendaciones podrías hacer al modelo para mejorarlo?



Ejercicio 2

Una compañía comercializadora de energía compra la electricidad a los generadores en el mercado mayoritario, donde después de una serie de contratos y control riesgos de precios esta se vende a los usuarios finales que pueden ser clientes residenciales, comerciales o industriales.

El sistema de la compañía que administra este producto tiene la capacidad de exportar la información de proveedores, clientes y transacciones en archivos CSV.

Requisitos técnicos:

1. Crear una estrategia de datalake en s3 con las capas que usted considere necesario tener y cargue esta información de manera automática y periódica. Los archivos deben particionarse por fecha de carga.
2. Realice 3 transformaciones básicas de datos utilizando AWS Glue y transforme la información para que esta sea almacenada en formato parquet en una zona procesada.
3. Utilizando AWS Glue, crea un proceso que detecte y catalogue automáticamente los esquemas de los datos almacenados en el datalake.
4. Utilizando Python, realiza consultas SQL básicas sobre los datos que han sido transformados.

Documentación:

1. Realiza una descripción detallada del pipeline de datos construido.
2. Indica que proceso es necesario seguir para configurar permisos y políticas para los diferentes servicios de AWS utilizados.

Puntos adicionales (plus)

1. Crea la IaC (Infraestructura como código) necesaria para desplegar esta solución en AWS.
2. Configure AWS Lakeformation para centralizar el gobierno, la seguridad y compartir los datos alojados en el datalake creado.
3. Construya un pipeline de datos que permita cargar esta información desde el datalake en la zona procesada a un datawarehouse en redshift.

Utilice información ficticia en los archivos csv que se simula entregue el sistema transaccional. A continuación, tendrá una recomendación de columnas de estos archivos, pero podrá ajustarla a sus necesidades:



Archivo proveedores: nombre de proveedor, tipo de energía (eólica, hidroeléctrica, nuclear).

Archivo clientes: tipo de identificación, identificación, nombre, ciudad.

Archivo transacciones: tipo de transacción (venta o compra), nombre del cliente/proveedor, cantidad comprada, precio, tipo de energía.

Importante: El código fuente creado deberá ser desplegado en una herramienta de control de versiones como github, azure devops, gitlab o similares.

Ejercicio 3

Supongamos que hoy haces parte del equipo de Datos & Analítica y requerimos concientizar a las áreas del negocio de su relevancia, del porqué aplicar calidad, gobierno y centralización de datos es importante, así como explicar de forma no técnica conceptos como que es un “datalake”, que es una “etl”, que es “gobierno de datos”, entre otras. Crea un paper de publicación que podamos enviar a todas las áreas (por ejemplo, un newsletter, boletín, infografía, entre otros).