Responda las siguientes preguntas:

¿En qué consisten datos estructurados, semiestructurados y no estructurados? Comente ejemplos de estos tipos de datos.

- Datos estructurados: Son datos que encajan dentro de una plantilla o estructura que se mantiene constante a través del tiempo, por ejemplo información básica de una persona, o un inventario dentro de una biblioteca.
- Datos semiestructurados: Son datos que no están dentro de una estructura definida pero contienen etiquetas que permiten crear una jerarquía que les dá órden, por ejemplo los archivos JSON.
- Datos no estructurados: Son datos cuya estructura no se encuentra definida para encajar con una plantilla o que cambia a través del tiempo constantemente, por ejemplo los datos que se almacenan en una base de datos NoSQL como el tránsito dentro de una página web, o información de compras.

¿En qué consisten datos de series de tiempo? ¿Se consideran logs, datos de series de tiempo?

Los datos de series de tiempo consisten en un conjunto de datos que van cambiando a través del tiempo, por lo que son una forma estructurada de representar los datos.
Por lo que sí se puede considerar que los logs sí sin series de tiempo, ya que estos tienen una estructura definida donde almacenan distintos tipos de datos que van variando durante el tiempo de uso.

¿Comente diferencias entre Lake house, Data warehouse y Data mart?

Lake house:

- Permite consultar datos a travez de los almacenes de datos, lagos de datos y bases de datos operacionales para obtener resultados más precisos de manera más rápida.
- Con esta estructura se pueden almacenar datos en un formato de archivos abiertos dentro de el lago de datos y consultarlos mientras se unen con los datos dentro del almacén de datos.
- Los datos son de fácil acceso para análisis y herramientas de aprendizaje automático.

Data warehouse:

- Al usar almacenes de datos se pueden obtener analíticas de manera más rápida y con mayor cantidad de datos, así como descubrir patrones ocultos en los datos mediante la reutilización de herramientas de inteligencia de negocios.
- Los científicos de datos pueden consultar almacenes de datos para realizar análisis e identificar tendencias de manera local.
- Usuarios a lo largo de la corporación consumen datos mediante consultas SQL,

reportes periódicos, y páneles de información como sea necesario para realizar decisiones de negocio críticas.

Data mart:

- Es una forma simple de un almacén de datos, solamente que estos están orientados a un área funcional o tema de interés.
- Se pueden construir mercados de datos a partir de grandes almacenes de datos, almacenamientos operacionales, o un híbrido entre ambos.
- Los mercados de datos son sencillos de diseñar, construir y administrar.
- Las consultas entre áreas funcionales pueden ser complejas debido a la distribución.

¿En qué consiste Row-oriented Column-oriented databases? Suponiendo que existe una tabla en una base de datos relacional con 10 columnas cuyos nombres son column1, column2, …., column10, ¿Una consulta como "SELECT column1, colum2 FROM tabla" se vería mas beneficiada por Row-oriented o Column-oriented? Explique.

- Row oriented DB: Las bases de datos orientadas a filas almacenan filas enteras de un bloque físico, este tipo de bases de tiene un alto rendimiento para operaciones de lectura, lo cual es posible por índices secundarios. Algunos ejemplos de este tipo de bases de datos son: MySQL, PostgreSQL y Oracle Database Server, entre otros.
  Estos sistemas son usados mayormente como almacenes de datos, sin embargo, están mejor diseñados para procesamiento de transacciones que pasa análisis.
  Los desarrolladores utilizan distintas técnicas para optimizar el rendimiento de este tipo de sistemas, por ejemplo: Construyendo vistas materializadas, creando tablas desplegables preagregadas, construyendo índices para cada posible combinación de predicados, implementando particiones de datos para aprovechar la función de optimizar particiones del optimizador de consultas.
  Los almacenes de datos orientados a filas son limitados por los recursos que se encuentran disponibles en una sola máquina. los mercados de datos alivianan el problema usando fragmentación funcional, lo que permite separar el almacén de datos en múltiples almacenes de datos, cada uno satisfaciendo las necesidades de un área funcional en específico, no obstante, los mercados de datos crecen mucho con el tiempo, por lo que el procesamiento de los datos se vuelve cada vez más lento.
  En un almacén de datos basado en filas, cada consulta debe ser leída a través de cada una de las columnas por cada una de las filas en los bloques que satisfacen el predicado de la consulta, incluyendo columnas que no se han seleccionado por el usuario, este tipo de acercamiento crea un cuello de botella significante en rendimiento dentro de los almacenes de datos donde las tablas tienen más columnas pero la consulta usa solo unas pocas.
- Column oriented DB: Este tipo de bases de datos organiza cada columna en su propio conjunto de bloques físicos, en vez de agruparlas como filas completas dentro de un bloque, esta funcionalidad permite que sean más eficientes para datos de entrada y salida en las consultas de lectura, debido a que solamente deben leer las columnas accesadas por la consulta en disco.

Esto hace que las bases de datos orientadas a columnas sean más eficientes para los almacenes de datos que las bases de datos orientadas a filas, otro beneficio de este tipo de sistemas es que el comprimir los datos está mejorado, esto debido a que cala columna está almacenada en su propio conjunto de bloques del mismo tipo, esto permite que se utilice menos espacio comparado a las bases de datos orientadas a las filas.

- Por lo que responiendo a la pregunta sobre si la consulta de "SELECT column1, colum2 FROM tabla" se vería mas beneficiada por Row-oriented o Column-oriented, podemos afirmar que sería más eficiente usar una base de datos orientada a columnas, la primera razón es que este tipo de consulta está solicitando datos específicos de dos columnas dentro de una tabla que contiene más de las columnas solicitadas, por lo que si usamos una base de datos orientada a columnas solamente se revisa la información dentro de las columnas específicadas, además de que se obtiene el bloque de la columna entero, en vez de hacerlo por fila y en todas las columnas de la tabla como sería en una base de datos orientada a filas.