# Lectura 1: Data Warehousing on AWS

Diego Granados Retana 2022158363

Bases de datos II

8 de agosto, 2023

**Responda las siguientes preguntas:**

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

Los datos estructurados son los datos que pueden ser contenidos en filas y columnas y campos predefinidos (Marr, 2019). Se pueden agrupar entidades para formar relaciones. Es muy fácil de almacenar, analizar y realizar búsquedas. Usualmente se maneja con Structured Query Language. Puede ser creada por máquinas o humanos. Se pueden hacer consultas con uniones complejas. Algunos ejemplos incluyen datos financieros, información de direcciones, información demográfica, registros de bitácora, entre otros. Los datos semiestructurados son aquellos que tienen características consistentes y que los definen, como etiquetas o metadatos, pero no necesariamente se puede fijar en una estructura rígida (Marr, 2019). Un ejemplo puede ser un correo electrónico. El cuerpo del correo es variante, pero siempre hay elementos fijos, como el correo emisor, el receptor, el asunto y la hora enviada. Otro ejemplo puede ser una fotografía. El contenido de la imagen no es estructurado, pero tiene información, como la hora en la que fue tomada, la ubicación donde se tomó, el dispositivo que la tomó e incluso se puede atribuir una etiqueta para organizarla mejor, como por ejemplo si la foto es de un perro o de una casa. Otro ejemplo son datos de XML. Los datos no estructurados son aquellos que no se pueden asociar a un modelo de datos (Marr, 2019). Esto los hacía más difíciles de buscar, administrar y analizar. No obstante, con la inteligencia artificial se ha logrado procesar más fácilmente. Estos datos se pueden almacenar en data lakes, bases de datos NoSQL y data warehouses. Algunos ejemplos son videos, audios, contenido de redes sociales, imágenes satelitales y páginas web.

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

Una serie temporal es "una colección ordenada de mediciones tomadas en intervalos regulares." (IBM, 2021) Los datos pueden ser de algo que se quiera analizar. Los intervalos pueden ser de cualquier unidad de tiempo, pero deberían ser el mismo. Si un intervalo no tiene datos, se considera en el valor perdido. Se pueden clasificar como:

- Dependiente: una serie de la cual se quiere hacer un pronóstico.
- Predictora: una serie que ayuda a explicar el objetivo, como utilizar los resultados de la inversión en mercadeo para predecir ventas.
- Evento: Una serie que se utiliza para predecir incidentes recurrentes.
- Intervención: Serie predictora que se utiliza para tener en cuenta incidentes específicos que ocurrieron antes, como la pérdida de electricidad.

Yo creo que un log sí podría considerarse un dato de series de tiempo dependiendo de lo que registren. Como los logs se registran constantemente a lo largo del tiempo, se pueden ver los cambios en el sistema y las tendencias que surgen. Por ejemplo, si un log registra que se vendió un producto, esos logs se pueden usar para una serie de tiempo que trate de predecir la cantidad de ventas. No obstante, también considero

que depende de la información del log. Como usualmente son semiestructurados, si la información que brinda no tiene mucha relevancia, como que por ejemplo un usuario se conectó al sistema o se realizó una operación administrativa, el log no puede ser útil para una serie de tiempo que intenta predecir algo para la toma de decisiones de negocios.

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

Un data warehouse es una base de datos relacional que almacena datos de sistemas transaccionales y aplicaciones de funciones de negocios. Todos los datos son estructurados y está diseñada para optimizar las consultas SQL (AWS, s.f.). Un data mart es una forma simple de data warehouse que está enfocada en un solo asunto, como ventas, finanzas o mercadeo (Oracle, s.f.). Un data lake es un repositorio de almacenamiento centralizado y flexible que guarda datos estructurados y no estructurados en su forma original, sin formatear (Kutay, 2023). No se define un schema. La data se extrae, carga y transforma para analizar. Un data lakehouse es una arquitectura que combina data warehouses y data lakes. Hay un solo repositorio para todos los datos, sean estructurados, semiestructurados o no estructurados. Permiten almacenar datos en formatos abiertos y hacer consultas con uniones a la data del data warehouse. Esto permite que la información sea más accesible para analizar y usarse en herramientas de machine learning.

Los data warehouses tienen múltiples fuentes de datos, sean internas o externas (AWS, s.f.). Se puede extraer datos de cualquier lado y si se puede estructurar, se puede poner en el warehouse. Un data mart tiene menos fuentes y suele ser más pequeño. Los data warehouses se utilizan para múltiples funciones de negocios, mientras que los data marts solo se enfocan en un aspecto. Los data warehouses usualmente se utilizan por mucho tiempo mientras que los data marts se utilizan mientras se necesiten y luego se eliminan. Los data lakes y warehouses pueden recibir información de cualquier fuente, pero los warehouses solo pueden almacenar la que sea estructurada. Se debe definir un schema. Los data lakes pueden almacenar cualquier cosa. Los data warehouses tienen información más confiable. Se pueden eliminar la información duplicada, ordenar, resumir y verificar. En un data lake es más difícil. Un data warehouse también tiene el mejor rendimiento y velocidad de consultas. Un data lake prioriza el volumen que puede almacenar sobre rendimiento. Un data lakehouse reduce la duplicación de datos al tener solo una plataforma que almacena todo (Kutay, 2023). Si se tiene tanto un warehouse y un lake, es posible que haya duplicación. Los lakehouses permiten un acceso más directo a herramientas de inteligencia de negocios. Como sí tiene un schema, los datos son más robustos, seguros y se facilita la administración. Los data lakehouses son tecnología nueva, entonces es posible que todavía en algunos casos sea mejor utilizar los data warehouses y data lakes, que tienen más tiempo de uso.

**¿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 más beneficiada por Row-oriented o Column-oriented? Explique.**

Los row-oriented databases almacenan filas o registros enteros en un solo bloque (AWS Whitepaper, 2023). Los índices permiten que haya un alto rendimiento. Se utilizan en data warehouses y procesamiento de transacciones online y no son tan buenas para análisis. Son limitadas por los recursos disponibles en una sola máquina. Cada consulta tiene que leer todas las columnas para los registros que estamos buscando, aún si no las escogimos. Esto hace que haya muchos problemas de rendimiento si hay muchas columnas. Los column-oriented databases almacenan cada columna en su propio bloque. Son más eficientes para operaciones de lectura porque solo acceden a las columnas que pide el query. Son mejores para data warehousing. Además, se puede comprimir mejor, ya que cada bloque contiene el mismo tipo de datos. Se necesita menos espacio

de almacenamiento. Yo considero que, para esta consulta, lo que serviría mejor es un column-oriented database, ya que en el row-oriented database se tienen que leer todas las columnas del registro, aún si no se usan. En esta consulta se tienen que leer todos los registros de la tabla, por lo que no se ganaría nada con la capacidad de leer solo algunos registros. Si fuera row-oriented, se estaría desperdiciando la lectura de ocho columnas. En cambio, con column-oriented, solo se leerían las dos columnas que necesitamos completas, por lo que es la mejor opción.

## Bibliografía

AWS. (s. f.). Data Lake vs Data Warehouse vs Data Mart - Difference between Cloud Storage Solutions - AWS. Amazon Web Services, Inc. Recuperado 5 de agosto de 2023, de https://aws.amazon.com/compare/the-difference-between-a-data-warehouse-data-lake-and-data-mart/ AWS Whitepaper. (2023). Data Warehousing on AWS. perusall.com. Recuperado 5 de agosto de 2023, de https://app.perusall.com/courses/ic4302-2023-02/data-warehousing-on-aws IBM. (2021, 17 agosto). Datos de series temporales. IBM.com. Recuperado 5 de agosto de 2023, de https://www.ibm.com/docs/es/spss-modeler/saas?topic=models-time-series-data Kutay, J. (2023). Data Warehouse vs. Data Lake vs. Data Lakehouse: An overview of three cloud data storage patterns. Striim. https://www.striim.com/blog/data-warehouse-vs-data-lake-vs-data-lakehouse-an-overview/#dlh Marr, B. (2019, 18 octubre). What's The Difference Between Structured, Semi-Structured And Unstructured Data?. Forbes. https://www.forbes.com/sites/bernardmarr/2019/10/18/whats-the-difference-between-structured-semi-structured-and-unstructured-data/?sh=635cde2b2b4d Oracle. (s. f.). What is a Data Mart? | Oracle. oracle.com. Recuperado 5 de agosto de 2023, de https://www.oracle.com/autonomous-database/what-is-data-mart/#:~:text=A%20data%20mart%20is%20a%20simple%20form%20of%20a%20data,fewer%20sources%20than%20data%20warehouses.