PROCESAMIENTO POR LOTES

PROCESAMIENTO POR LOTES

Como ya vimos en la presentación: “Del dato a la Información”, en la caracterización del dato en cuanto a su latencia, hablábamos de “**el tiempo que transcurre entre la solicitud de un dato y su disponibilidad para su uso”**.

Llevándonos a dos escenarios:

- Tiempo Real
- Datos en Lotes

PROCESAMIENTO POR LOTES

CARACTERÍSTICAS	
PROCESAMIENTO POR LOTES	**PROCESAMIENTO EN TIEMPO REAL**
Los datos deben ser contextualizados en un marco temporal para ser tratados de forma conjunta.	Es necesario gestionar eventos de forma individual, tan pronto como se generan.
Pueden ser almacenados temporalmente en varias ocasiones, los datos durante el procesamiento.	Los datos asociados a los eventos son gestionados en memoria, con una persistencia mínima o nula.
Los procesos de tratamiento suelen estar planificados o pueden desencadenarse por algún evento.	El procesamiento de eventos se realiza de forma continua e ininterrumpida, con la mínima demora.
Las operaciones y análisis deben adaptarse a las dependencias de los datos de negocio que la ventana de ejecución de los procesos o viceversa.	La captura, transformación y análisis del dato conforman un flujo continuo desde los orígenes de los eventos hasta las aplicaciones consumidoras finales.
	Un tratamiento intensivo del dato impactará en el tiempo de entrega.

PROCESAMIENTO POR LOTES

EJEMPLOS	
PROCESAMIENTO POR LOTES	**PROCESAMIENTO EN TIEMPO REAL**
Generación de informes diarios, semanales o mensuales.	Monitoreo de sistemas.
Procesamiento de transacciones bancarias al final del día.	Detección de fraude.
Cálculos de nómina.	Aplicaciones de streaming.
Ejecución de modelos de aprendizaje automático para análisis predictivos.	Juegos en línea.
Análisis de tendencias de ventas.	Sistemas de control industrial.

PROCESAMIENTO POR LOTES

VENTAJAS	
PROCESAMIENTO POR LOTES	**PROCESAMIENTO EN TIEMPO REAL**
Eficiencia: el procesamiento por lotes puede ser muy eficiente para procesar grandes volúmenes de datos, ya que los datos se procesan en grupos.	**Inmediatez**: los resultados están disponibles de inmediato, lo que permite una toma de decisiones más rápida.
Costo-efectividad: generalmente es más económico que el procesamiento en tiempo real, ya que no requiere una infraestructura de procesamiento continuo.	**Mejor experiencia del usuario**: las aplicaciones en tiempo real pueden proporcionar una experiencia más interactiva y atractiva para los usuarios.
Simplicidad: los sistemas de procesamiento por lotes suelen ser más simples de diseñar e implementar que los sistemas en tiempo real.	**Mayor flexibilidad**: permite adaptar el procesamiento de datos a las condiciones cambiantes.

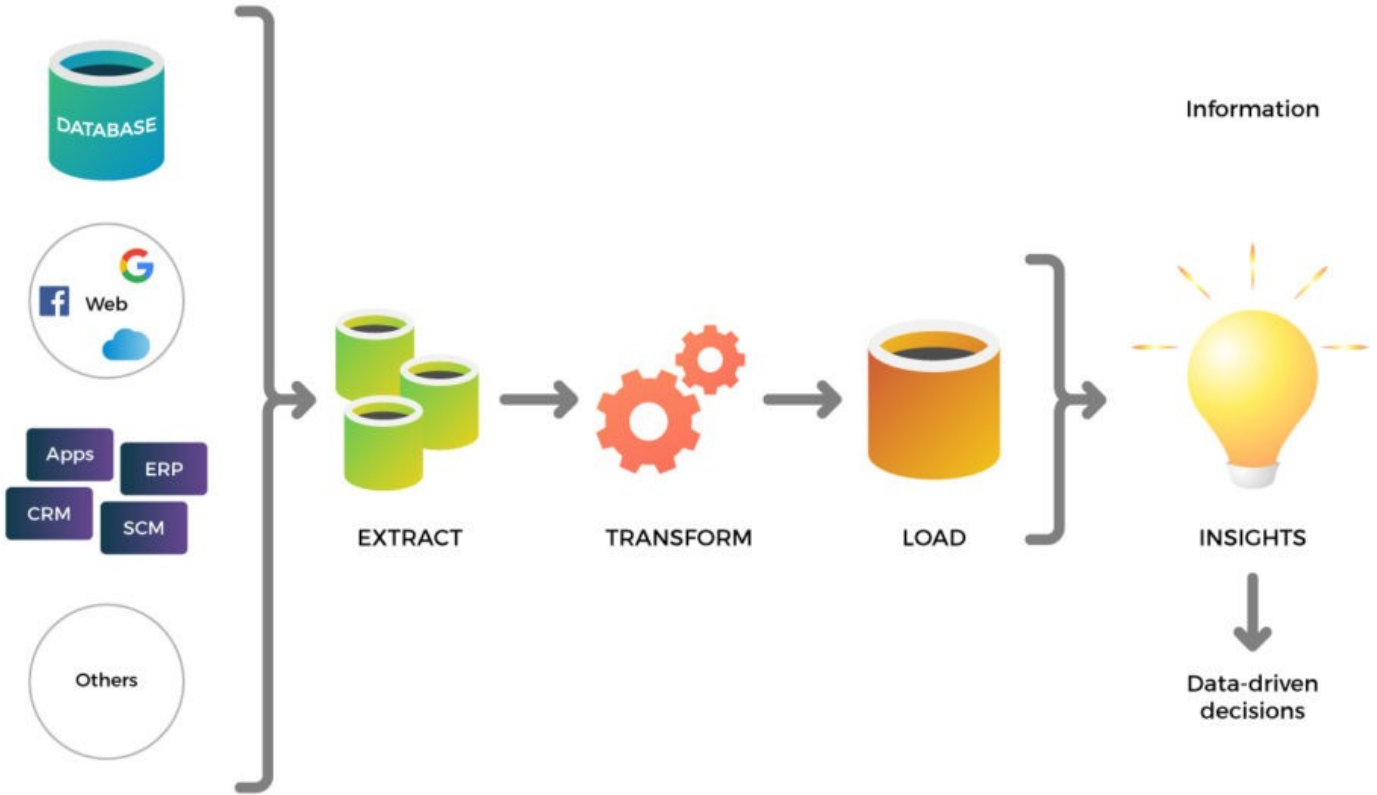
PROCESAMIENTO POR LOTES

DESVENTAJAS	
PROCESAMIENTO POR LOTES	**PROCESAMIENTO EN TIEMPO REAL**
Latencia: hay un retraso inherente en el procesamiento por lotes, ya que los datos se procesan en lotes.	**Complejidad**: los sistemas en tiempo real son más complejos de diseñar e implementar que los sistemas de procesamiento por lotes.
Falta de inmediatez: los resultados no están disponibles de inmediato, lo que puede ser un problema para las aplicaciones que requieren información en tiempo real.	**Costo**: es más costoso que el procesamiento por lotes, ya que requiere una infraestructura de procesamiento continuo.
Menos flexible: es menos flexible que el procesamiento en tiempo real, ya que los cambios en el proceso de procesamiento pueden requerir reprocesar todo el lote de datos.	**Escalabilidad**: puede ser difícil escalar los sistemas en tiempo real para manejar grandes volúmenes de datos.

Extracción, Transformación y Carga

ETL (*Extracción, Transformación* y *Carga*) es un proceso fundamental en la gestión integral de datos, que se utiliza para recopilar, procesar y cargar datos desde diferentes fuentes a un destino final, como un almacén de datos o un data lake.

Extracción, Transformación y Carga



Fuente: https://todoingenierias.com/proceso-etl-extraccion-transformacion-y-carga-de-datos/

Extracción

La etapa de extracción es **fundamental** porque:

- Sienta las bases para las etapas posteriores de transformación y carga.
- Permite obtener los datos en un formato compatible con la etapa de transformación.
- Es el punto de partida para la gestión integral de datos, abarcando desde la infraestructura de datos hasta la modelización y el análisis de la información.

Extracción

El **proceso de extracción** implica una serie de pasos, algunos de estos pasos son:

- Acceso a la fuente de datos: ya sea una tabla de origen, una API o un archivo.
- Inventario de los orígenes de datos: para tener un control de las fuentes y detectar posibles problemas.
- Filtrado de datos: para seleccionar solo los datos necesarios.
- Agregación de datos: para combinar datos de diferentes fuentes.
- División de datos: para separar los datos en diferentes conjuntos.

Extracción

Antes de implementar la etapa de extracción, es fundamental considerar los siguientes aspectos:

- **Variedad de los Orígenes de Datos**: Es crucial considerar que los datos pueden originarse de diversas fuentes y en una variedad de formatos. Esto podría requerir el uso de tecnologías y técnicas de extracción específicas para cada caso. Por ejemplo, la extracción de datos de una base de datos relacional requerirá el uso de un controlador **JDBC**, mientras que la extracción de datos de un archivo plano requerirá el uso de un analizador de archivos.

Extracción

Algunos ejemplos comunes son:

- **Bases de datos relacionales**: Son el tipo de base de datos más común y almacenan los datos en tablas con filas y columnas.
- **Archivos planos**: Son archivos de texto sin formato que almacenan los datos en un formato delimitado, como comas o tabulaciones.
- **APIs** (Interfaces de Programación de Aplicaciones): Permiten acceder a datos de aplicaciones de terceros, como redes sociales, servicios web, etc.

Extracción

Antes de implementar la etapa de extracción, es fundamental considerar los siguientes aspectos:

- **No Interferir con los Sistemas Operacionales**: La extracción de datos no debe afectar negativamente el funcionamiento normal de los sistemas de origen. Por ejemplo, si se extraen datos de un sistema de producción, es importante asegurarse de que la extracción no afecte el rendimiento del sistema.
- Frecuencia de las Extracciones: La frecuencia con la que se extraen los datos dependerá de las necesidades

Extracción

Antes de implementar la etapa de extracción, es fundamental considerar los siguientes aspectos:

- **Frecuencia de las Extracciones**: La frecuencia con la que se extraen los datos dependerá de las necesidades de la aplicación. Por ejemplo, si se necesita información actualizada diariamente, las extracciones se deben realizar diariamente. Si la información solo se necesita mensualmente, las extracciones pueden ser mensuales.

Extracción

Antes de implementar la etapa de extracción, es fundamental considerar los siguientes aspectos:

- **Recuperación de los Datos Extraídos**: Siempre se debe tener en cuenta la posibilidad de que alguna etapa posterior del proceso ETL falle. Por lo tanto, es fundamental garantizar la posibilidad de recuperar los datos extraídos en caso de que esto ocurra. Esto se puede lograr almacenando los datos extraídos en un almacenamiento intermedio.

Extracción

Antes de implementar la etapa de extracción, es fundamental considerar los siguientes aspectos:

- **Creación de Metadatos**: Los metadatos son datos que describen otros datos, como el origen, la fecha de extracción, el formato, etc. La creación de metadatos es fundamental para el seguimiento del origen de los datos y para garantizar su calidad.

Extracción

ESTRATEGIAS DE EXTRACCIÓN EN ETL

Las **estrategias de extracción** se refieren a los métodos utilizados para recopilar datos de diferentes fuentes durante la etapa inicial del proceso ETL (Extracción, Transformación y Carga). El objetivo principal es obtener los datos necesarios para su posterior análisis y procesamiento.

Extracción

Exiten dos estrategias principales para la extracción de datos:

- **Extracciones Totales**: Esta estrategia consiste en extraer todos los datos de la fuente en cada ejecución del proceso ETL. Es un enfoque simple, pero puede ser ineficiente si el volumen de datos es grande, ya que implica procesar todos los datos, incluso aquellos que no han cambiado desde la última extracción.

Extracción

Exiten dos estrategias principales para la extracción de datos:

- **Extracciones Diferenciales**: A diferencia de las extracciones totales, las extracciones diferenciales solo extraen los datos que han sido modificados o añadidos desde la última extracción. Esta estrategia es más eficiente, ya que solo se procesan los datos nuevos o actualizados, lo que reduce el tiempo de procesamiento y el consumo de recursos.

Extracción

La elección de la estrategia de extracción adecuada depende de varios factores, entre ellos:

- **Frecuencia de actualización**: Si los datos se actualizan con frecuencia, las extracciones diferenciales son más adecuadas, ya que solo procesan los cambios.
- **Complejidad del sistema**: La complejidad del sistema de origen también influye en la elección de la estrategia. Si el sistema es complejo, las extracciones diferenciales pueden ser más difíciles de implementar.
- **Volumen de datos**: Si el volumen de datos es pequeño, las extracciones totales pueden ser una opción viable. Sin embargo, si el volumen de datos es grande, las extracciones diferenciales son más eficientes.

Extracción

Una estrategia de extracción correcta es fundamental para el éxito del proceso ETL. Una estrategia inadecuada puede resultar en:

- **Ineficiencia en el procesamiento**: Si se extraen más datos de los necesarios, se desperdician recursos y tiempo.
- **Inconsistencias en los datos**: Si la estrategia no captura todos los cambios, los datos pueden ser inconsistentes.
- **Mayor complejidad en la implementación**: Algunas estrategias de extracción diferencial pueden ser más complejas de implementar que las extracciones totales.

Extracción

El **inventariado** y **perfilado** de datos son procesos esenciales en la etapa de extracción del proceso ETL.

El **inventario** de datos consiste en identificar y documentar todas las fuentes de datos que se utilizarán en el proceso ETL. Esto incluye bases de datos, archivos planos, APIs, sistemas de archivos distribuidos, páginas web, etc. El objetivo es tener un control exhaustivo de las fuentes de datos y poder detectar posibles problemas.

Extracción

El **perfilado de datos** (*data profiling*) es un proceso que se utiliza para analizar y auditar la estructura y la calidad de los datos. Permite obtener estadísticas, identificar valores anómalos, inconsistencias en la codificación, o posibles relaciones entre valores de diferentes orígenes. Esta información es fundamental para poder realizar acciones correctoras en la etapa de transformación.

Extracción

Beneficios del Inventariado y Perfilado de Datos

- Mejor comprensión de los datos: El inventario y el perfilado proporcionan una visión completa de los datos, incluyendo su origen, formato, calidad y contenido.
- Detección temprana de problemas: Permite identificar problemas en los datos, como valores faltantes, duplicados o inconsistencias, antes de que afecten las etapas posteriores del proceso ETL.
- Toma de decisiones informadas: La información obtenida a través del inventario y el perfilado permite tomar decisiones más informadas sobre cómo limpiar, transformar y cargar los datos.

Extracción

Beneficios del Inventariado y Perfilado de Datos

- Mejora de la calidad de los datos: El perfilado de datos ayuda a identificar y corregir problemas de calidad, lo que mejora la calidad general de los datos.
- Reducción de costes: La detección temprana de problemas en los datos puede ayudar a reducir costes asociados a la limpieza y corrección de datos en etapas posteriores.

Extracción

Supongamos que estamos analizando una base de datos de clientes. El proceso de perfilado de datos podría incluir:

- **Análisis de la estructura de la tabla**: Determinar el número de columnas, sus nombres y tipos de datos.
- **Análisis del contenido de las columnas**: Obtener estadísticas como la media, la mediana, el valor mínimo y máximo, la frecuencia de valores nulos, etc.

Extracción

- **Identificación de valores atípicos**: Detectar valores que se desvían significativamente de la norma.
- **Análisis de la consistencia de los datos**: Verificar si los datos cumplen con las reglas de negocio y las restricciones definidas.

Transformación

La transformación es una etapa crucial en el proceso de extracción, transformación y carga (ETL) de datos, especialmente en el contexto de Big Data. **Este proceso busca preparar los datos para su uso efectivo en análisis y toma de decisiones**.

Transformación

Esta acción sería inicialmente mínima o inexistente en una *data lake*, en un *data warehouse* estaría poblada de actividades. Lo mencionado anteriormente es consistente debido a que en el primero se guardan los datos en **crudo** y en el segundo hay que hacerlo de forma **conformada** y **estandarizada**.

Transformación

En el caso de un data lake, las necesidades de transformación empiezan precisamente una vez los datos están en el repositorio. En este caso, es más propio hablar de **extracción**, **carga** y **transformación** (ELT).

Estén donde estén los datos, el objetivo es acondicionarlos para su consumo, medie o no un punto de almacenaje.

Transformación

Vamos a ver las etapas de la transformación:

- **Definición**: La transformación implica convertir los datos extraídos de diversas fuentes a un formato compatible con el modelo de datos destino.
- **Objetivo**: El objetivo principal es asegurar que los datos sean consistentes, precisos y estén estructurados de manera que se ajusten a las necesidades del sistema de destino, como un data lake o data warehouse.

Transformación

- **Tareas Principales**:
  - **Limpieza**: Eliminar datos duplicados, gestionar valores erróneos u omitidos, y estandarizar valores para garantizar la calidad de los datos.
  - **Normalización:** Ajustar los datos a un formato estándar para facilitar su análisis y comparación.
  - **Combinación y Agregación**: Unir datos de diferentes fuentes y agregarlos para obtener información resumida.

Transformación

- **Tareas Principales**:
  - **Filtrado**: Seleccionar solo los datos relevantes para el análisis, descartando los que no son necesarios.
  - **Cálculo y Derivación**: Realizar cálculos y derivar nuevas variables a partir de los datos existentes.
  - **Conformación**: Adaptar los datos a la estructura del modelo de datos de destino.
  - **Formateado**: Darle a los datos un formato final adecuado para su consumo.

Transformación

- **Complejidad:** La amplitud de la transformación varía dependiendo de factores como el tamaño de los datos, la calidad inicial de los datos y las necesidades del sistema de destino. Los data lakes, por ejemplo, suelen requerir una transformación mínima inicial, mientras que los data warehouses necesitan una transformación más completa.

Transformación

- **Ubicación:** La transformación se realiza en un área intermedia, después de la extracción de los datos y antes de la carga en el sistema de destino.
- **Implementación:** La transformación se puede llevar a cabo utilizando diversas tecnologías, desde lenguajes de programación como Python hasta plataformas especializadas en Big Data como Spark.

Transformación

Ya se ha comentado que las operaciones de extracción, transformación y carga se llevan a cabo en un área intermedia por un sistema dedicado. Estas operaciones se pueden realizar empleando distintos lenguajes de programación e interrogación de propósito general (Python, Scala, SQL, etc.).

Transformación

Se ha de poner énfasis en que la etapa de transformación es fundamental para garantizar la calidad y utilidad de los datos en un sistema de Big Data. Un proceso de transformación bien diseñado asegura que los datos sean confiables, consistentes y estén listos para su análisis, lo que permite a las organizaciones obtener información valiosa para la toma de decisiones.

Carga

La fase de **carga** es la etapa final del proceso **ETL** (Extracción, Transformación y Carga). Una vez que los datos han sido extraídos de sus fuentes originales y transformados a un formato adecuado, la fase de carga se encarga de insertarlos en el sistema de destino. Este destino puede ser un simple archivo, una base de datos transaccional, un data warehouse o un data lake.

Carga

Siguiendo con el razonamiento del apartado anterior, en el caso de un *data lake* la carga consistirá prácticamente en un volcado de nuevos datos en el repositorio.

En estos casos donde el volumen de transformaciones es muy pequeño, la tecnología de **replicación de datos** (*data replication*) puede ser muy interesante.

¿En qué consiste esta tecnología?

Carga

Data Replication:

Mediante un modelo de publicación-subscripción, la replicación de datos permite tener en sincronía estructuras de datos que residen en repositorios diferentes y heterogéneos, de manera que se mantengan consistentes prácticamente en tiempo real. Minimizando el impacto en los sistemas de origen, ya que el control de cambios se realizan en el registro de operaciones (log), sin necesidad de acceder a las tablas.

Carga

Aunque son similares, la replicación de datos y el control de cambios (CDC) tienen propósitos distintos. Es importante decir que el CDC se utiliza fundamentalmente en la etapa de extracción como mecanismo de detección, ya que nos permite identificar los cambios y acceder a ellos para luego gestionarlos en la etapa de transformación. Por el contrario, la replicación la ubicamos en la etapa de carga para aplicar en destino los cambios que se van produciendo en origen o en el área intermedia.

Carga

Todo lo anteriormente dicho tiene su importancia ya que, es posible que convivan ambas tecnologías, replicación de datos y CDC; con la primera para el movimiento constante de datos entre origen y destino, y la segunda para auditar los cambios.

Carga

Vamos a ver los puntos importantes de la fase de carga:

- El **lugar** donde se cargan los datos depende de las necesidades del proyecto pudiendo ser:
  - **Archivo**: una forma sencilla de almacenar datos, pero con limitaciones en términos de análisis.
  - **Base de datos transaccional**: ideal para datos que necesitan ser actualizados con frecuencia, como los datos de transacciones comerciales.

Carga

- **Data warehouse:** diseñado para análisis y almacenamiento de grandes volúmenes de datos históricos.
- **Data lake:** un repositorio centralizado que almacena datos en su formato raw, permitiendo flexibilidad en su posterior uso.

Carga

- **Tipos de carga**:
  - **Carga completa (full load):** Se purgan todos los datos existentes en el destino y se carga el conjunto completo de datos. Este método es sencillo pero poco eficiente si solo se han modificado algunos datos.
  - **Carga incremental:** Solo se cargan los datos nuevos o modificados desde la última carga. Esto mejora la eficiencia y reduce el tiempo de carga.

Carga

- **Programación de la carga:** La frecuencia de la carga depende de los requisitos del proyecto. Puede ser por hora, diaria, semanal, mensual o incluso anual.
- **Impacto en el rendimiento:** Una carga bien diseñada minimiza el impacto en el rendimiento del sistema de destino, especialmente en el caso de data warehouses, donde el volumen de datos suele ser muy grande.

Carga

- **CDC (Change Data Capture):** Esta técnica se utiliza para identificar los cambios realizados en los datos y solo cargar las modificaciones, lo que optimiza aún más la carga incremental**.**

Carga

Consideraciones Adicionales:

- La fase de carga debe tener en cuenta la conectividad del sistema de destino. Si la conectividad es limitada, puede ser necesario implementar mecanismos para asegurar la integridad de los datos durante la carga.
- El diseño del data warehouse influye en la complejidad de la carga. En data warehouses con estructuras dimensionales, la carga puede requerir actualizaciones en varias tablas (tabla de hechos y tablas de dimensiones).

Carga

Consideraciones Adicionales:

- La elección entre carga completa e incremental se basa en la frecuencia de las actualizaciones, el volumen de datos y el impacto en el rendimiento del sistema de destino.

El diseño e implementación de la fase de carga deben considerar las necesidades del proyecto, el tipo de destino y el rendimiento general del sistema.

Carga

Consideraciones Adicionales:

- La gestión de los metadatos. Todas las etapas que hemos contemplado son generadoras de gran cantidad de metadatos de proceso, conteniendo información sobre la tipología y número de registros extraídos, transformados, rechazados, cargados, tiempos de ejecución de cada flujo, errores, etc. Todos estos metadatos se unen a los metadatos operacionales y técnicos, permitiendo dibujar todo el ciclo de vida de los datos y realizar análisis de impacto de los diferentes elementos.

Carga

Consideraciones Adicionales:

- La gestión de los metadatos. Toda **área intermedia** debería contar con un repositorio de metadatos dirigido a los ingenieros de datos, con el fin de soportar tareas de monitorización, análisis comparativo y auditoría.