2. INTEGRACIÓN DE DATOS

Como hemos comentado anteriormente, las herramientas ETL han evolucionado, y esta evolución es debida a diferentes causas. Una de estas es que existen diferentes tipos de datos y que nuestras necesidades de negocio también van cambiando.

Veamos los tipos de datos:

- **Estructurados**: son aquellos contenidos en bases de datos, son los que tienen un formato ya predefinido, en el que los campos ocupan un sitio fijo y, por lo tanto, conocemos de forma anticipada su organización, tipo, etc. Se almacenan en tablas y la información se representa por datos elementales. Pueden ser recibos, facturas en tablas, etc.
- **Semiestructurados**: son aquellos que constan de un formato legible para máquinas, pero no son completamente estructurados. Son datos que sí tienen alguna estructura autodefinida pero, no tienen una estructura fija. Ofrecen información poco regular ya que a veces, debido a su complejidad en el proceso de carga, se pueden perder datos. Pueden ser documentos Excel, HTML tabulado, etc.
- **No estructurados**: son aquellos en formato legible para humanos, pero no para máquinas. Son aquellos que no tienen una estructura específica. Manipular estos es algo más complejo que los anteriores y no se pueden almacenar en una tabla como sí sucede con los estructurados. Podrían ser los archivos Word, PDF, HTML no tabulado, etc. Estos pueden obtenerse mediante text mining, por ejemplo.

Por esta razón se necesita llevar a cabo la **integración de datos** que consiste en realizar aplicaciones, técnicas, productos y tecnologías que nos permitan obtener una única visión consistente de nuestros datos de negocio.

2.1 TÉCNICAS DE INTEGRACIÓN DE DATOS

Existen distintas técnicas de integración de datos:

- **Propagación de datos**: se copian los datos de un lugar origen a un destino local o remoto. Con programas que generan un fichero, pueden extraerse los datos del origen, que serán transportados al destino, donde se usarán como fichero de entrada para cargar en la base de datos de destino. Sin embargo, es más eficiente descargar únicamente los datos que han cambiado en origen respecto a la última propagación realizada, creando un fichero de carga incremental que será transportado al destino. Estos procesos suelen ser de tipo en línea y trabajan con arquitectura de push. Puede realizarse como:
  - Intercambio bidireccional.
  - Distribución.
- **Consolidación de datos**: se capturan las modificaciones realizadas en distintos entornos origen y se propagan a un único entorno destino. En este último, se almacenará una copia de todos los datos. Con esta técnica es complicado trabajar con tiempos de latencia (retardos temporales dentro de una red) bajos:
  - Si no se requiere latencia baja, mediante procesos batch, se proveen los datos en intervalos prefijados superiores a varias horas. Para conseguir los datos se hace mediante técnica pull que consiste en usar consultas SQL.
  - Si se requiere baja latencia, se usa la técnica push. Hay que identificar los cambios producidos en origen para transmitir únicamente esos cambios. Normalmente se emplea alguna técnica de tipo CDC (change data capture).
- **Federación de datos**: permite acceder a distintos entornos origen de datos, que pueden estar en diferentes o en los mismos gestores de datos y máquinas y crear una visión del conjunto como si fuese una única base de datos integrada. El motor de federación de datos, cuando una aplicación lanza una consulta SQL contra la vista virtual, descompone la consulta en consultas

individuales para cada uno de los orígenes de datos físicos y la lanza contra cada uno de estos datos involucrados. Cuando recibe todas las respuestas, integra todos los resultados en uno único, realizando agregaciones y/o ordenaciones, sumarizaciones, y devuelve los datos a la aplicación que lanzó la consulta original. El catálogo de datos común es uno de los elementos clave del motor de federación, puesto que contiene información sobre los datos (estructura, localización, demografía, etc.). Esto permite que se elija el camino más eficiente de acceso a los datos, optimizando la división de la consulta original al enviarla a los gestores de bases de datos

- **CDC (Change Data Capture):** se atrapan o capturan los cambios producidos por las aplicaciones operacionales en las bases de datos de origen, así pueden ser propagados y/o almacenados en los entornos destino para que estos mantengan la consistencia con los entornos origen. Veamos las principales técnicas de change data capture:
    - **CDC por aplicación:** es la misma aplicación la que genera la actualización de datos en origen, y actualiza los entornos destino o almacena localmente los cambios en una tabla de paso (staging) por medio de una operación de INSERT dentro de la misma unidad lógica de trabajo.
    - **CDC por timestamp:** se puede usar cuando los datos de origen incorporan un timestamp (por ejemplo a nivel de fila si el origen es una tabla relacional) de la última actualización de esta. El change data capture se limitará a escanear los datos de origen para extraer así los datos que posean un timestamp posterior al de la última vez que se ejecutó el proceso de CDC. Son estos datos los que hay que actualizar en los entornos destino porque son los que han cambiado desde la última obtención de datos.
    - **CDC por triggers:** los disparadores o triggers son acciones que se ejecutan al actualizarse (mediante UPDATE, DELETE o INSERT) los datos de cierta tabla sobre la que están definidos. Estos disparadores pueden usar estos datos de la actualización en sentencias SQL para generar

cambios SQL en otras tablas remotas o locales. Así, un modo de obtener cambios es crear triggers o disparadores sobre las tablas de origen, las acciones de los cuales modifiquen los datos de las tablas destino.

- **CDC por captura de log:** el fichero de log de la base de datos se examina constantemente en busca de algún cambio en las tablas que se deben monitorizar. Aquí la obtención de información no afecta al rendimiento del gestor relacional debido a que no se necesita acceso al disco duro que contiene el fichero de log, sino que se basa en la lectura de los buffers de memoria de escritura en el log.

- **Técnicas híbridas**: se suelen emplear varias técnicas de integración constituyendo así lo que llamamos técnica híbrida. La elección de la técnica va a depender de nuestros requisitos de negocio y tecnológicos y de las posibles y probables restricciones presupuestarias.

2.2 TECNOLOGÍAS DE INTEGRACIÓN DE DATOS

Existen distintas tecnologías de integración de datos basadas en las técnicas comentadas en el apartado anterior:

- **ETL de generación de código**: tienen un entorno gráfico en el que se diseñan y especifican los datos de origen, sus transformaciones y los entornos destino. El resultado que se genera es un programa de tercera generación (típicamente COBOL) que permite llevar a cabo las transformaciones de los datos. Estos programas simplifican el proceso ETL, sin embargo, añaden pocas mejorías en cuanto al establecimiento y la automatización de los flujos de procesos que se requieren para realizar la ETL. Los administradores de datos, normalmente, son los que se encargan de administrar y distribuir el código compilado, planificar y ejecutar los procesos de lotes, y llevar a cabo el transporte de los datos.
- **ETL basados en motor**: permite crear flujos de trabajo en tiempo de ejecución definidos por medio de herramientas gráficas. El entorno gráfico permite hacer un mapping de los entornos de origen y destino, las transformaciones de datos que se requieren, los procesos por lotes necesarios, y el flujo de procesos. La

información que hace referencia a diseño y procesos de ETL se almacena en el repositorio del catálogo de metadatos. Está compuesto por distintos motores:

- **Motor de extracción**: usa adaptadores de ficheros planos, adaptadores como ODBC, JDBC, JNDI, SQL nativo u otros. Los datos pueden ser extraídos en modo push o pull. En modo pull planificado, soportando técnicas de consolidación en procesos por lotes. En modo push, usando técnicas de propagación en procesos de tipo en línea. En los dos casos se pueden usar las técnicas de changed data capture (CDC) que hemos comentado antes.
- **Motor de transformación**: proporciona una librería de objetos, la cual permite a los desarrolladores transformar los datos de origen para adaptarse así a las estructuras de datos de destino. De este modo permite, por ejemplo, la sumarización de los datos en destino en tablas resumen.
- **Motor de carga**: usa adaptadores a los datos de destino, como el SQL nativo, o cargadores masivos de datos para actualizar o insertar los datos en los ficheros de destino o bases de datos.
- **Servicios de administración y operación**: permiten planificar, ejecutar y monitorizar los procesos de ETL. También permiten la visualización de eventos y la recepción y resolución de errores en los procesos.

- **ETL integrado en la base de datos**: dentro del motor de la base de datos algunos fabricantes incluyen capacidades ETL. Estas, en general, tienen menos funcionalidades y son menos complejas y también menos completas que los ETL comerciales de generación de código o basados en motor. Por esto, a los ETL integrados en las bases de datos los clasificamos en tres clases en relación con los ETL comerciales que se basan en motor o de generación de código:
  - **ETL cooperativos**: para mejorar los procesos de ETL, los productos comerciales pueden utilizar funciones avanzadas del gestor de base de datos. Ejemplos de ETL cooperativos son los que pueden usar procesos almacenados y SQL complejo para llevar a cabo las transformaciones de

los datos en origen de un modo más eficiente, o usar paralelismo de CPU en consultar para minimizar asi el tiempo de los procesos ETL.

- **ETL complementarios**: reciben este nombre cuando a los ETL comerciales les ofrecen funcionalidades complementarias a los ETL de bases de datos. Por ejemplo, algunos gestores de bases de datos ofrecen soporte a MQT (Materializad Query Tables) o vistas de sumarización precalculadas, mantenidas y almacenadas por el gestor que pueden utilizarse para evitar transformaciones de datos realizadas por el ETL comercial. También hay gestores que permiten la interacción directa a través de SQL con middleware de gestión de mensajes (por ejemplo, permitiendo la inserción de nuevos mensajes en colas por medio de SQL) o con aplicaciones que se comunican a través de web services.
- **ETL competitivos:** hay gestores que ofrecen herramientas gráficas integradas que explotan sus capacidades ETL en lo que es competencia con los ETL comerciales.

2.3 USOS DE LA INTEGRACIÓN DE DATOS

Los procesos de integración de datos se usan en diferentes tipos de proyectos. Destacamos los siguientes:

- Migración de datos.
- Corporate Performance Management (CPM).
- Business Intelligence (BI).
- Master Data Management (MDM).
- Customer Data Integration (CDI).
- Procesos de calidad de datos.
- Product Information Management (PIM).
- Enterprise Information Management (EIM).
- Data Warehousing.