Business intelligence

Unit 2 – Datawarehouse

S2-3 – Data integration - ETL

Data integration

- Data come from different systems.
  - Need for integration.
  - Very different source systems
- Data stored at least in two places:
  - The source system.
  - The datawarehouse.
- DWH features
  - Exchange, integrate and consolidate data from many systems.
  - It provides a new unified database.
  - Volume tends to be very large.

Data integration - ETL

- The maintenance of the data warehouse is done by means of the ETL (Extraction - Transformation - Load) process
  - is the responsibility of the team developing the data warehouse.
  - is built specifically for each DW.
  - you can use market tools or programs designed specifically.
  - It is very time consuming when building our DW/BI solution.
- General steps:
  - Initial load
  - Refreshment: daily, weekly, monthly, … periodic maintenance

Importance of ETL

- It adds value to data
  - Metadata: document data quality
  - Captures data flows (processes)
  - Increases quality: less mistakes
  - Conformed data between departments

ETL

- ETL description
  - The data of one or more operating systems need accessible from the DWH.
    - Eventually data can be replicated in the DWH.
  - ETL is the process of extracting data from source systems and bringing them to DWH
    - Load DWH regularly.
  - It consists of several steps
    - extraction
    - cleaning
    - transformation
    - load

ETL

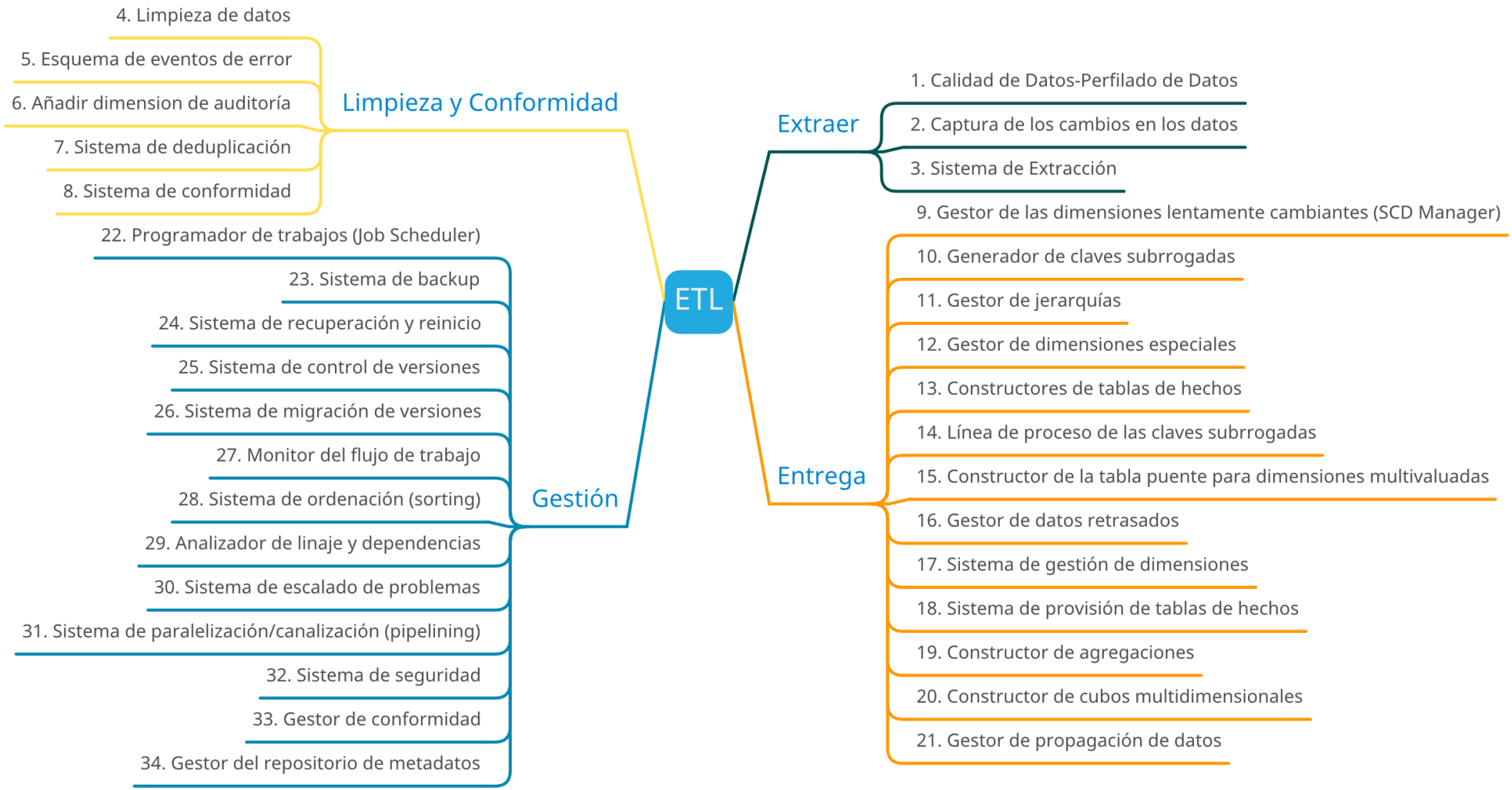
Requisitos: Paso necesario para establecer la estructura de un sistema ETL

- **Requisitos del negocio**: las necesidades del negocio determina la elección de las fuentes de datos y su transformación dentro de ETL.
- **Requisitos de conformidad**: El cumplimiento de requisitos legales/regulatorios (ej: telecomunicaciones) y de generación de reportes se incluyen en este apartado. Debemos identificar los datos y reportes sobre los que debemos mantener la cadena de custodia y los marcos legales vigentes.
- **Requisitos de calidad de datos**: Debemos identificar aquellos elementos que presentan problemas de calidad y evaluar si los sistemas operacionales pueden ser modificados para resolverlos antes de proceder a su extracción. Incluye seguimiento de calidad.
- **Requisitos de seguridad**: Debemos determinar el nivel de sensibilidad de los datos. Está relacionado con el apartado de conformidad..
- **Requisitos de las interfaces BI**: los datos deben llegar a las aplicaciones de forma rápida, simple y fiable. Debemos identificar los hechos y dimensiones que vamos a exponer a las herramientas. Esto implica una interacción entre el equipo ETL, los arquitectos de datos y los desarrolladores de aplicaciones para determinar estos requisitos.

ETL

- **Req. de integración de datos**: debemos identificar dimensiones y atributos asociados comunes (compartidos por varios esquemas estrella) dentro de los procesos del negocio para desarrollar métricas comunes (ej. KPIs). Queremos que todos los sistemas puedan trabajar juntos y ello implica también el desarrollo de tablas de hechos comunes (estandarizadas con las mismas unidades y granuralidad).
- **Req. de latencia**: Debemos conocer la rapidez con la que los datos de fuente deben llegar al usuario del DW. Esto influye de manera importante en la arquitectura ETL.
- **Req. de archivado y linaje**: Existen requisitos legales y de buenas prácticas, pero incluso sin ellos, debemos identificar los requisitos de archivado (#copias, política de retención,…) y linaje de datos, que usan metadatos para describir el origen, cambios, y la calidad de los datos.
- **Requisitos de habilidades disponibles**: Debemos implementar ETL de acuerdo con la familiaridad de la empresa/trabajadores con las tecnologías a aplicar. Esto implica conocer las habilidades disponibles para gestionarlas mejor.
- **Licencias de sistemas legados**: En muchas ocasiones nos vemos obligados a trabajar en entornos legados en los cuales no podemos seleccionar la tecnología y metodologías más ventajosas. Esto puede suponer una restricción importante y hay que saber diferenciar entre su uso obligatorio o meramente recomendable.

ETL: Subsistemas (Kimball)



ETL: Comp 1: Extraer

Comp 1: Extraer

- **Subsist 1**: Perfilado de datos: Analizamos los datos para conocer su contenido, consistencia y estructura. Nos permite hacernos una idea de los hechos y dimensiones, y del mapeado a implementar.
- **Subsist 2**: Sist de captura de los cambios en los datos: Podemos cargar todos los datos de las fuentes, pero con los volúmenes de datos de hoy en día eso en ocasiones no es factible. Debemos tener la capacidad de solo transferir aquellos cambios desde la última extracción. Técnicas comunes implican chequear los logs de transacciones, detectar cambios con CRC/Hash, por el timestamp un record ha sido creado/modificado, etc.

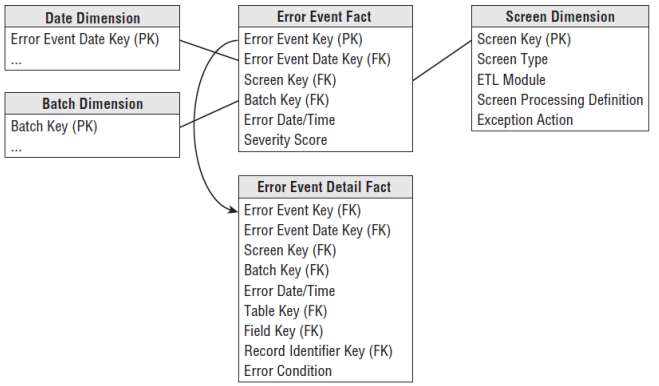
- **Subsist 3**: Sistema de Extracción: Relacionado con la obtención de datos de los sistemas fuente. Cada una de ellas podría tener estar en un entorno/BBDD diferente. Podemos obtener los datos en forma de ficheros (XML, csv, txt,…) o a través de un flujo de datos que podemos iniciar a través de una consulta, acceso a un servicio web, etc. Con un fichero no necesitamos re-consultar la fuente en caso de fallo. Aquí podemos decidir comprimir los datos para acelerar la transmisión. Debemos evaluar la necesidad de encriptar la inomación en tránsito.

ETL: Comp 2: Conformidad

Comp 2: Limpieza y conformidad de datos (la T de ETL)

- **Subsist 4**: Limpieza de datos: Desarrollar una arquitectura para diagnosticar problemas, determinar métricas sobre la calidad de los datos, y responder a problemas (ej. resolviéndolo, reportándolo, abortando, continuando,…). Los procesos de filtrado se implementan en el pipeline de datos en la forma de chequeos de calidad, que pueden ser: de columnas (datos dentro de una misma columna), estructurales (relaciones entre los datos en distintas columnas) y de las reglas de negocio (ej. un cliente VIP ha realizado unas compras por encima de un umbral).
- **Subsist 5**: Esquema de eventos de error: Un esquema dimensional centralizado para almacenar los eventos de los chequeos de calidad.

Cada error produce una entrada en Error Event Fact
Los procesos de filtrado son los responsables de añadir records a estas tablas.

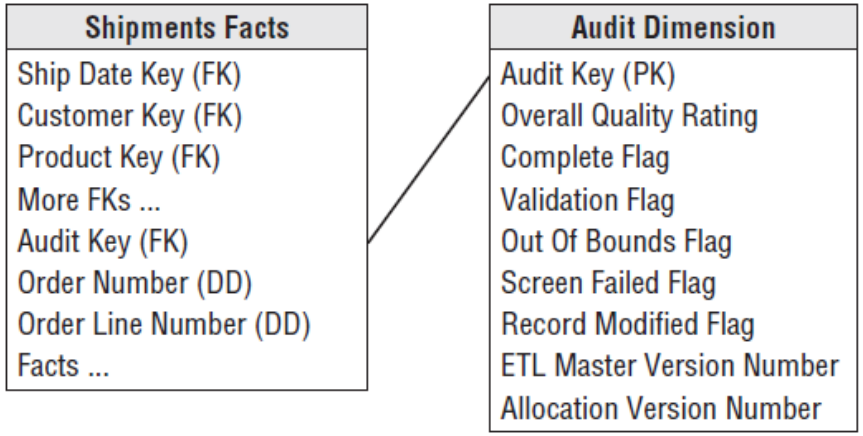


Un error que añade una fila e Error Event Fact podría generar muchas entradas en la Error Event Detail Fact

ETL: Comp 2: Conformidad

Comp 2: Limpieza y conformidad de datos (la T de ETL)

- **Subsist 6**: Añadir dimension de auditoría: Dimensión añadida por el sistems ETL con el propósito de evaluar los resultados de los procesos de filtrado. Si todo va bien, solo se genera una fila en la tabla Audit Dimension, y la correspondiente FK será añadida a la tabla de hechos. Pero si tenemos condiciones de error, necesitaríamos más entradas en la tabla de dimensiones



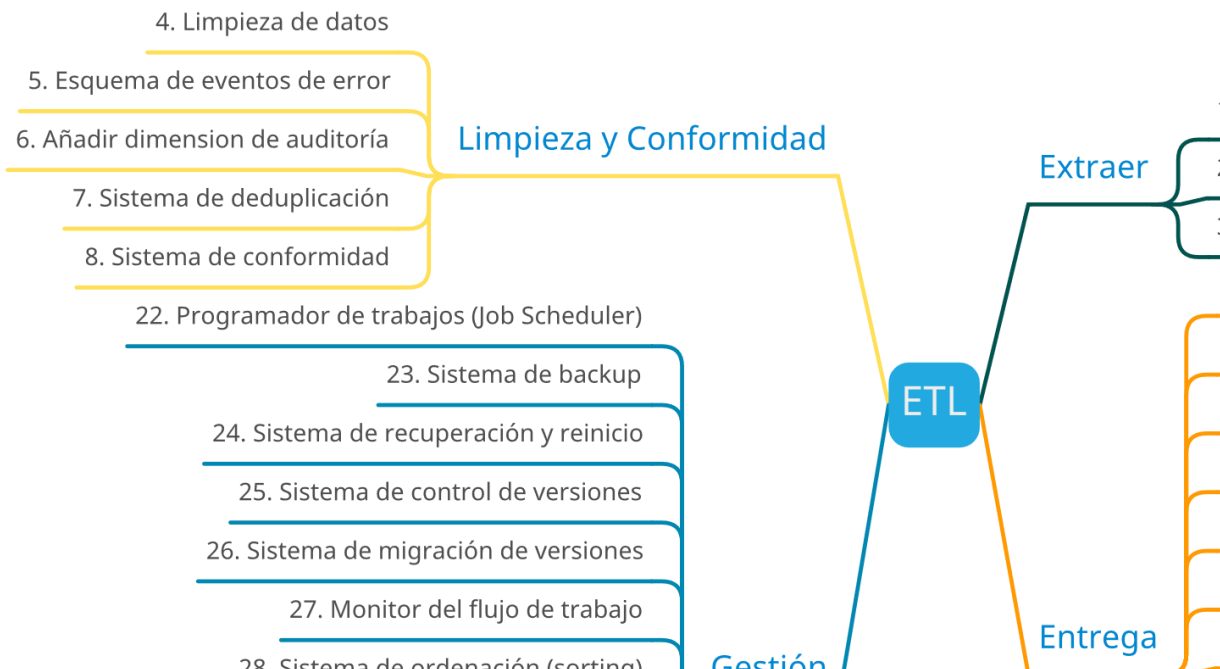
- **Subsist 7**: Sistema de deduplicación

En muchas ocasiones las dimensiones proceden de varias fuentes. Necesitamos combinarlas para crear una imagen que combine las columnas de más alta calidad para cada fila (proceso de supervivencia). Este proceso incluye reglas claras de negocio para decidir la prioridad de las diferentes fuentes a la hora de construir cada fila.

ETL: Comp 2: Conformidad

Comp 2: Limpieza y conformidad de datos (la T de ETL)

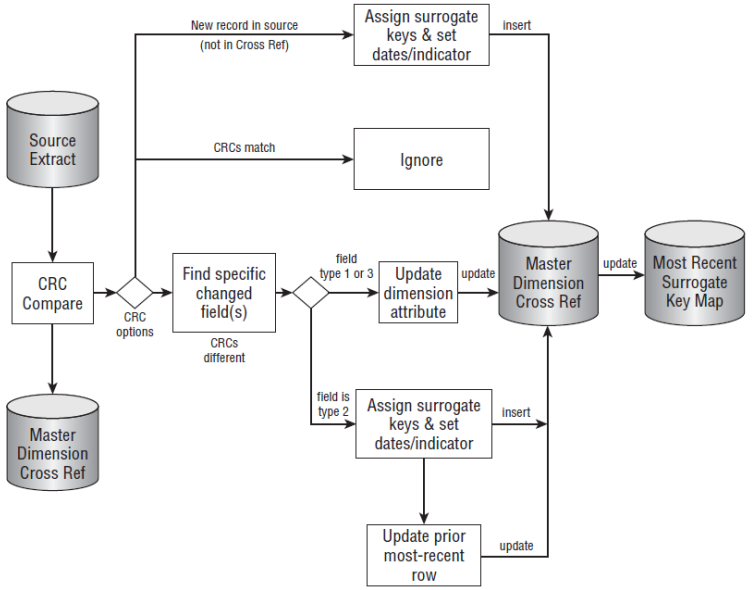
- **Subsist 8**: Sistema de conformidad: Encargado de definir y crear dimensiones y hechos estructuralmente idénticos a partir de la combinación e integración de datos procedentes de diversos sistemas que se han deduplicado, filtrados para eliminar datos inválidos y estandarizados.



ETL: Comp 3: Entrega

- **Comp 3: Entrega** (la L de ETL)
  - **Subsist 9:** Gestor de las dimensiones lentamente cambiantes (SCD Manager): El sistema ETL tiene que gestionar el valor de un atributo que ha cambiado respecto al valor almacenado en el DW. Las respuestas típicas son: sobreescribir (tipo 1), añadir una nueva fila (tipo 2), y añadir una nueva columna (tipo 3)

- **Tipo 1-** Cambiamos la dirección de un cliente. Solo actualizamos la fila sin tocar las claves de la tabla de dimensión ni la de la tabla de hechos.
- **Tipo 2-** Ahora imaginemos que queremos mantener el historial de cambios y, por tanto, al cambiar la dirección, necesitamos añadir una nueva fila. Seguimos sin tocar la tabla de hechos, pero usaremos ahora una nueva FK para ese cliente.
- **Tipo 3-** Añadimos una columna nueva para almacenar los cambios. Por ejemplo, en el caso anterior, añadimos una nueva columna para incluir la dirección vieja.



ETL: Comp 3: Entrega

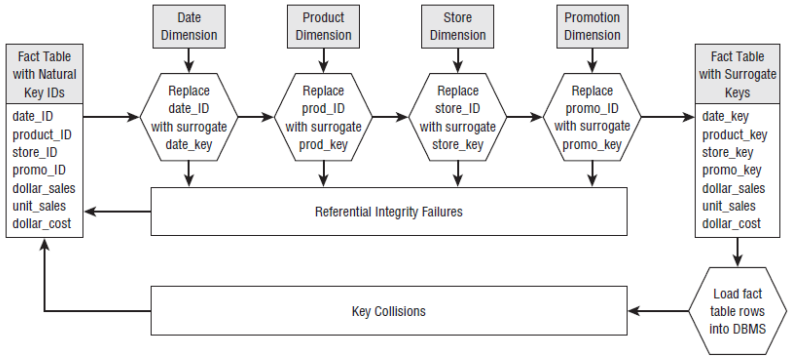
Comp 3: Entrega (la L de ETL)

- **Subsist 10**:Generador de claves subrrogadas: Se recomienda usar este tipo de claves como PK de las tablas de dimensiones. Necesitamos un mecanismo robusto para generarlas en entornos ETL. En PostgreSQL, una secuencia valdría.
- **Subsist 11**: Gestor de jerarquías: Es normal encontrarnos con jerarquías en las dimensiones. Hay 2 tipos:
  - **Fijas**: número estable de niveles que se modelan como atributos de la dimensión. Ej: producto->fabricante
  - **Desiguales (ragged)**: número variable de niveles. Podemos recurrir a un modelo copo de nieve para modelarlas. A veces , las direcciones postales son desiguales

- **Subsist 12**: Gestor de dimensiones especiales: Apoya las características de diseño de las dimensiones para la organización, pudiendo abarcar todas o algunas de:
- Dimensiones de tiempo y fecha
- Dimensiones basura (junk)
- Dimensiones “encogidas” (shrunken)- subconjuntos de dimensiones mayores
- Dimensiones pequeñas estáticas- normalmente tablas lookup sin fuente de referencia

ETL: Comp 3: Entrega

- **Comp 3: Entrega** (la L de ETL)
  - **Subsist 13:**Constructores de tablas de hechos: Enfocado a la construcción de los diferentes tipos de tablas de hechos:
    - **Transaccionales:** Se cargan cuando la transacción ocurre o en intervalos a la máxima granuralidad . Se cargan los datos y, a veces, si se requiere actualizar datos ya almacenados, primero se insertan y después de borran los viejos.
    - **Instantáneas (snapshot) periódicas:** Se cargan a intervalos regulares y representan períodos. Ej: balances mensuales de un banco.
    - **Instantáneas acumuladas:** para representar eventos finitos que evolucionan con un inicio y final bien definidos. Ej: procesar un pedido: tenemos primero la fecha de pedido, pero no sabemos la de envío, recogida... A medida que las sabemos tenemos que ir actualizándolas.



- **Subsist 14:** Línea de proceso de las claves subrrogadas: Reemplazar las claves operacionales naturales del registro de la tabla de hechos con las subrrogadas de la dimensión.

ETL: Comp 3: Entrega

- **Comp 3: Entrega** (la L de ETL)
  - **Subsist 15:** Constructor de la tabla puente para dimensiones multivaluadas: Soporta relaciones Many-to-Many entre hechos y dimensiones. Ej: hecho: compra (cantidad, valor); dimensión: razones de la compra. Otras: pacientes/diagnósticos, clases/profesores,…
  - **Subsist 16:** Gestor de datos retrasados: En realidad, no todos los datos llegan al mismo tiempo. Ej: odemos tener datos de ventas diarios y datos de personal mensuales. El sistema ETL debe solucionar estas situaciones en las que hechos y dimensiones pueden llegar tarde para mantener la integriad referencial. Una posible solución podría ser asignar un valor por defecto para la dimensión hasta que se conoce ésta. Ej: Nombre cliente: TBD
  - **Subsist 17:** Sistema de gestión de dimensiones: Autoridad centralizada que prepara y publica las dimensiones conformadas (dimensiones cuya semántica, estructura y uso conviene a toda la empresa) a la comunidad del DW. Entre sus responsabilidades:
    - Implementar etiquetas descriptivas para los atributos
    - Gestionar los cambios en los atributos
    - Añadir nuevas filas a la dimensión conformada, generando nuevas claves subrrogadas
    - Distribuir las actualizaciones dimensionales

ETL: Comp 3: Entrega

Comp 3: Entrega (la L de ETL)

- **Subsist 18:** Sistema de provisión de tablas de hechos: Administra una o varias tablas de hechos, siendo responsable de su creación, mantenimiento y uso. Responsabilidades:
- Recibe las dimensiones replicadas del gestor de dimensiones.
- Añade y actualiza las filas de las tablas de hechos
- Asegura la calidad de los hechos
- Recalcula los agregados
- Notifica a los usuarios de cambios, actualizaciones y problemas.

- **Subsist 19:** Constructor de agregaciones: Creación y mantenimiento de estructuras físicas agregadas en la BBDD que ayudan a mejorar el rendimiento. Debemos controlar cuidadosamente cuales necesitamos, sin quedarnos cortos ni pasarnos. Los hechos/dimensiones sumario se crean a partir de los hechos/dimensiones base.
- **Subsist 20:** Constructor de cubos multidimensionales: Crear y mantener los esquemas ROLAP estrella que permiten crear los cubos. Los cubos MOLAP presentan datos multidimensionales fáciles de explorar. Los cubos deben actualizarse si los hechos o dimensiones cambian.

ETL: Comp 3: Entrega

- **Comp 3: Entrega** (la L de ETL)
  - **Subsist 21:** Gestor de propagación de datos: Se encarga de mover datos del DW a otros entornos, como sistemas de minería de datos, o de auditoría.

ETL: Comp 4: Gestión Entorno

Comp 4: Gestionar el entorno ETL

- El entorno ETL debe ser fiable, disponible según los SLA, y adaptable según la evolución del negocio. Para ellos necesitamos los siguientes subsistemas:
  - **Subsist 22:** Programador de trabajos (Job Scheduler): Sistema para programar y lanzar los trabajos ETL. Debe conocer y controlar las dependencias entre trabajos ETL. Responsable de definir trabajos, programarlos, capturar metadatos y volcarlos en ficheros log así como notificar problemas.
  - **Subsist 23:** Sistema de backup: Almacenar, archivar y extraer elementos del sistema ETL.
  - **Subsist 24:** Sistema de recuperación y reinicio: Los trabajos deben recuperarse en caso de errores y tener la capacidad de reiniciarse.
  - **Subsist 25:** Sistema de control de versiones: Almacenamos información sobre las diferentes versiones (scripts, SQL, Jobs,…) en el flujo de procesos de ETL: Git, SVN, …

ETL: Comp 4: Gestión Entorno

Comp 4: Gestionar el entorno ETL

- **Subsist 26:** Sistema de migración de versiones: Transferir los cambios desde la etapa de desarrollo, a testeo, y después a producción
- **Subsist 27:** Monitor del flujo de trabajo: Panel de control y sistema de reporte de todos los trabajos iniciados por el programador de trabajos. Incluye un sistema de auditoría, los logs del ETL y la monitorización de la BBDD.
- **Subsist 28:** Sistema de ordenación (sorting): Ordenar es una capacidad muy importante en ETL, usada muy a menudo en agregaciones y joins.Es importante que se pueda aplicar de forma rápida y eficiente.
- **Subsist 29:** Analizador de linaje y dependencias: (linaje) ->Representa las fuentes físicas y las transformaciones de cualquier elemento de datos, en cualquier etapa del proceso ETL. (dependencia) ->Representar todos los elementos de datos ”downstream” y reportes finales que se verían afectados por un cambio en un elemento “upstream”. Ej dependencia: identificar los cubos y esquemas estrella que usan un elemento de la fuente.
- **Subsist 30:** Sistema de escalado de problemas: Sistema automático/manual donde una condición de error se eleva al nivel correspondiente para analizarlo y resolverlo. Va desde simples entradas en los logs, hasta a notificar a los operadores, supervisor o desarrollador.

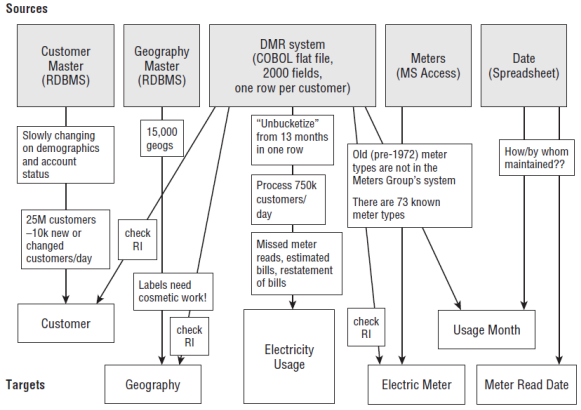
ETL: Comp 4: Gestión Entorno

Comp 4: Gestionar el entorno ETL

- **Subsist 31:** Sistema de paralelización/canalización (pipelining): Normalmente las restricciones temporales nos obligan a aprovecharnos de la disponibilidad de recursos computacionales para aquellas tareas que se pueden aprovechar de ello.
- **Subsist 32:** Sistema de seguridad: Gestionar la seguridad a nivel individual y rol de todos los datos y metadatos en ETL. Esto también incluye el sistema de backup.
- **Subsist 33:** Gestor de conformidad: Mantener la cadena de custodia de los datos en entornos regulados (ej: bancos, salud,…) y registrar quién ha accedido de forma autorizada a los datos. Los cambios en los datos deben de reportarse.
- **Subsist 34:** Gestor del repositorio de metadatos: Administra todos los metadatos de ETL. Esto incluye la captura y mantenimiento, incluyendo toda la lógica de las transformaciones. Incluye metadatos de procesos, técnicos y de negocio.

Planificar e implementar ETL

- **Pasos preliminares (Kimball):**
  - Diseño lógico
  - Comprender la arquitectura DW/BI
  - Mapear Fuente – Destino para todos los elementos de datos
- **Etapas planificación ETL:**
  1. Esbozar un plan de alto nivel: Identificar fuentes y destinos (dimensiones y tablas hecho)



ETL PROCESS STEP	ETL SUBSYSTEM: EXTRACTING DATA	CLEANING AND CONFORMING	DELIVERING FOR PRESENTATION	MANAGING THE ETL ENVIRONMENT
Plan				
Create a high level, one-page schematic of the source-to-target flow.	1			
Test, choose, and implement an ETL tool (Chapter 5).				
Develop default strategies for dimension management, error handling, and other processes.	3	4, 5, 6	10	
Drill down by target table, graphically sketching any complex data restructuring or transformations, and develop preliminary job sequencing.		4, 5, 6	11	22
Develop One-Time Historic Load Process				
Build and test the historic dimension table loads.	3	4, 7, 8	9, 10, 11, 12, 15	
Build and test the historic fact table loads, including surrogate key lookup and substitution.	3	4, 5, 8	13, 14	
Develop Incremental Load Process				
Build and test the dimension table incremental load processes.	2, 3	4, 7, 8	9, 10, 11, 12, 15, 16, 17	
Build and test the fact table incremental load processes	2, 3	4, 5, 8	13, 14, 16, 18	
Build and test aggregate table loads and/or OLAP processing.			19, 20	
Design, build, and test the ETL system automation.		6	17, 18, 21	22, 23, 24, 30

Planificar e implementar ETL

Etapas planificación ETL:

2. Seleccionar una herramienta ETL: Existen muchas opciones que pueden leer de diversas fuentes. Podemos desarrollarla o bien emplear herramientas GUI.
3. Desarrollar estrategias para las actividades más comunes en el sistema ETL. Incluye:
   - Extracción de datos y archivado
   - Evaluar calidad de dimensiones y hechos
   - Administrar cambios en atributos de dimensiones
   - Asegurar cumplimiento requisitos de disponibilidad del DW y ETL
   - Diseñar sistema de auditoría de datos
   - Organizar el almacenamiento de datos intermedio (staging area) para almacenamiento temporal.

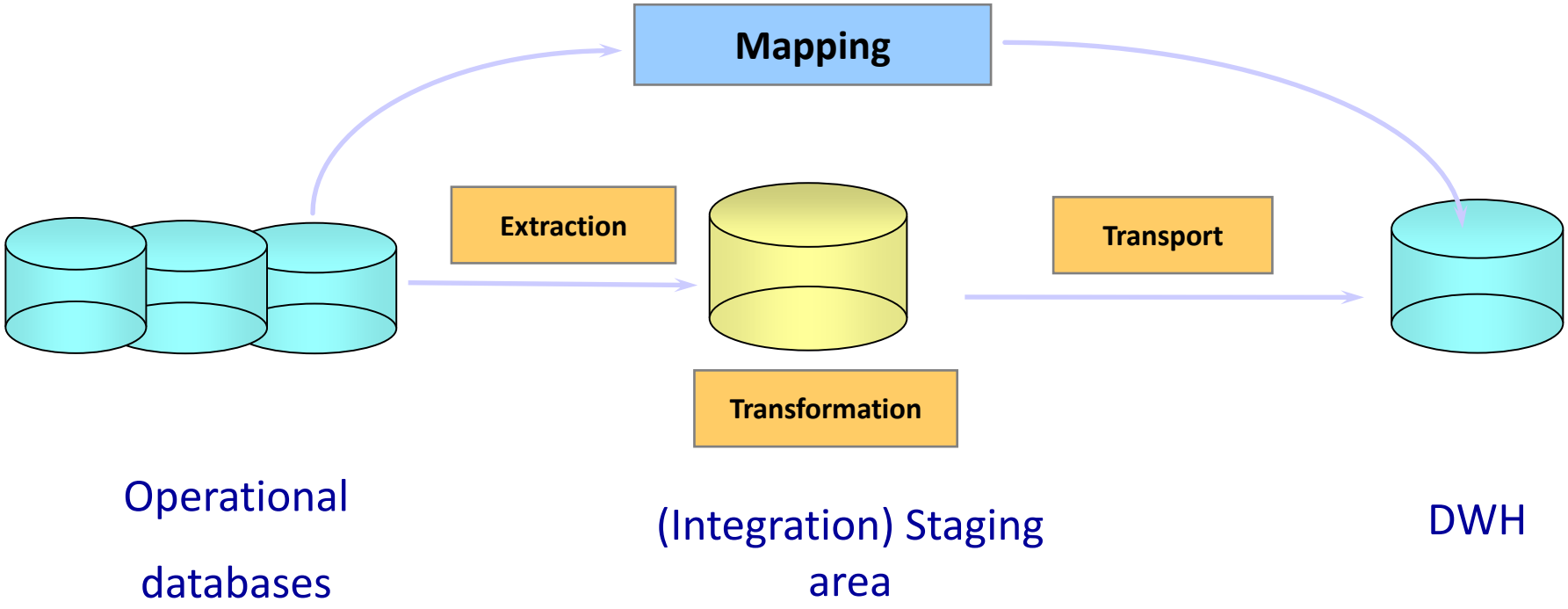
Planificar e implementar ETL

- **Etapas planificación ETL:**
  4. Mapear de forma detallada el flujo de datos desde las fuentes a las tablas de dimensiones y hechos. Podemos hacer uso de diagramas de flujo o esquemas que nos ayuden a realizar esta labor.
  5. Cargar las tablas de dimensiones con datos históricos: Este es un proceso que hacemos una vez y que difiere de la carga incremental. Normalmente comenzamos con las tablas más simples (ej: tipo 1). Después podemos transformar los datos : convertir tipos, combinarlos de fuentes diferentes, asignarles claves subrrogadas, etc.
  6. Cargar la tabla de hechos con datos históricos: Podemos realizar transformaciones: sustituir valores numéricos “especiales” por NULL en hechos aditivos o semiaditivos, calcular hechos complejos, añadir una clave de auditoría, etc.
  7. Procesado incremental de la tabla de dimensiones: A veces podemos usar la misma lógica que en la carga de datos históricos. Un reto es identificar los datos nuevos/actualizados.

Planificar e implementar ETL

- **Etapas planificación ETL:**
  8. Procesado incremental de la tabla de hechos: Esta fase normalmente requiere automatizar el proceso y puede requerir procesado en paralelo y chequeo de la calidad de datos.
  9. Agregar carga de tablas y OLAP: A veces necesitamos procesado adicionar más allá del esquema estrella. Necesitamos actualizar MOLAP, las vistas (incluyendo las materializadas) y las tablas sumario con agregaciones.
  10. Operación del sistema ETL y automatización: en esta etapa necesitamos programar los trabajos, gestionar los errores/excepciones, operaciones sobre las BBDD(actualizar las estadísticas, recrear los índices, limpiar datos,…)

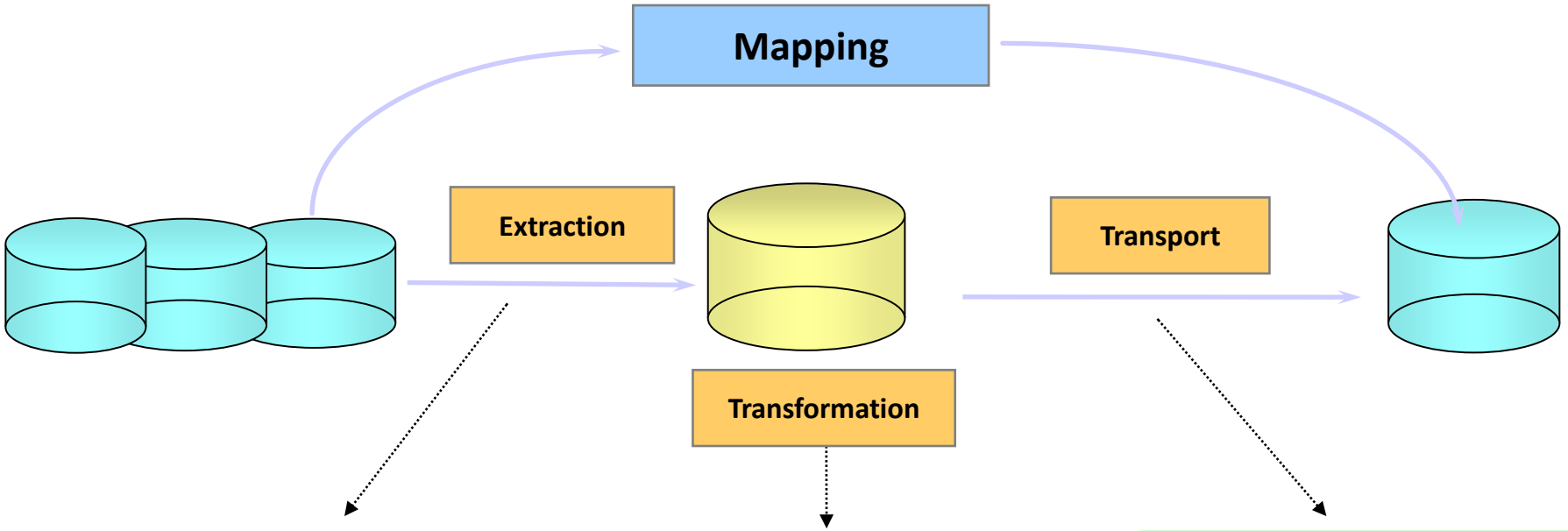
ETL



The Staging area:

- allows transformations without paralyzing the operational databases and the datawarehouse.
- Allows storing metadata.
- Facilitates the integration of external sources.
- Not necessarily stores data in RM

ETL

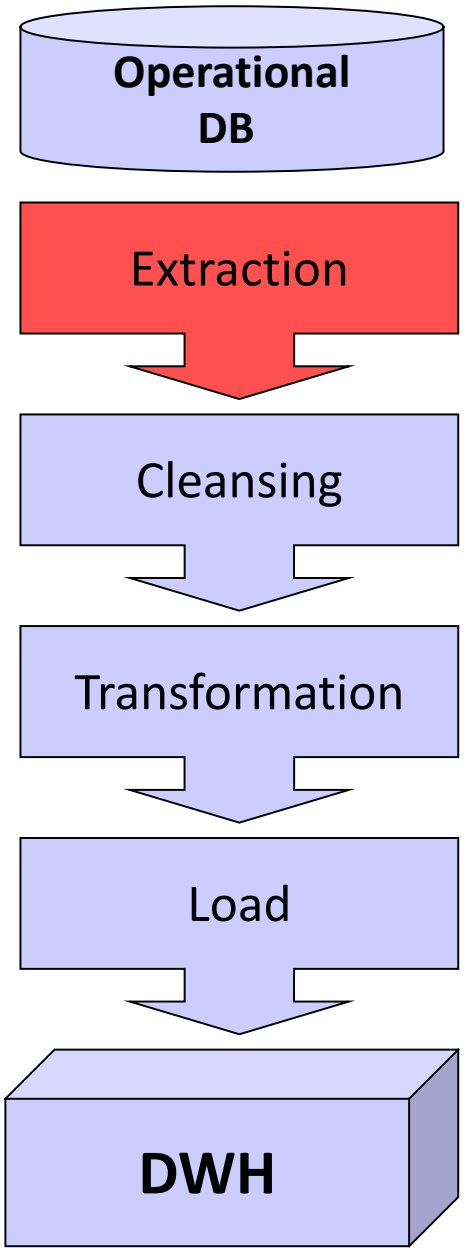


- ✓Identifying data that has changed
- ✓Extraction (read) data.
- ✓Aggregates definition
- ✓Maintenance of metadata

- ✓Data cleansing and Transformation
- ✓Data integration (calculation of derived data)
- ✓Key creation
- ✓Obtaining aggregates
- ✓Maintenance of metadata

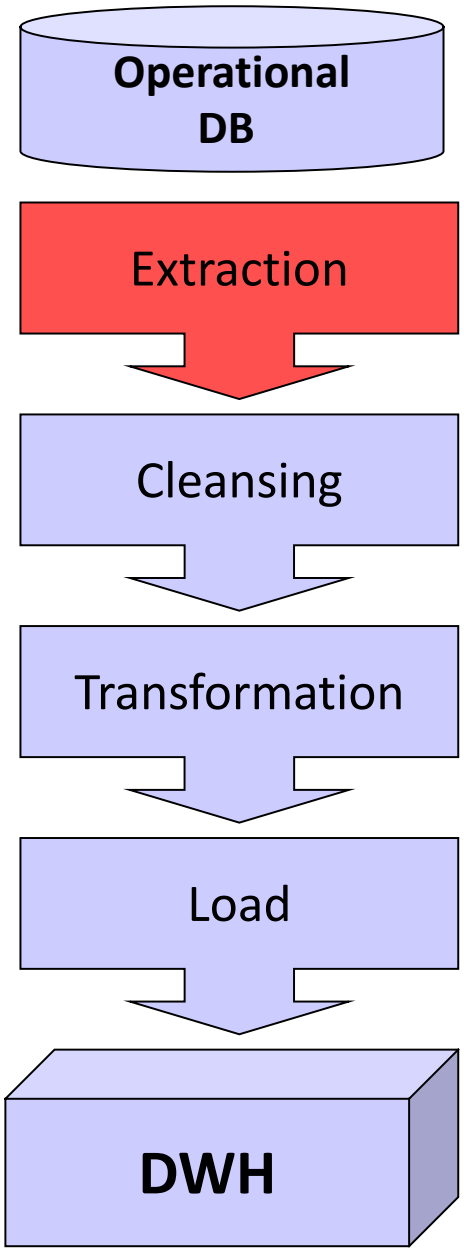
- ✓ Load data
- ✓ Indexing
- ✓ Loading aggregated data
- ✓ Quality test.
- ✓ Exception management
- ✓Maintenance of metadata

ETL

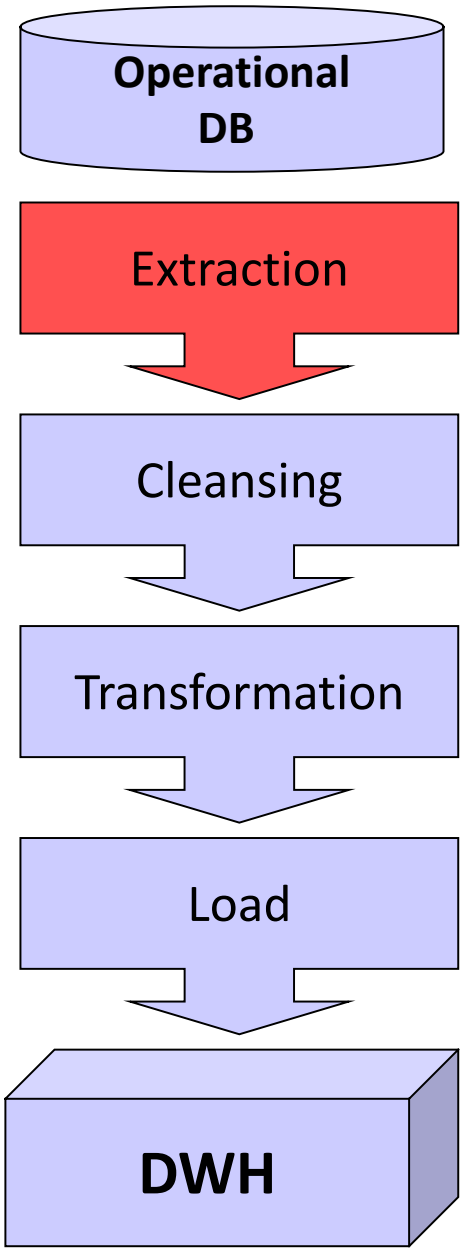


- Extraction = obtaining an image of a given subset of the source data to be loaded into the DWH
- Logical and physical extraction.
  - Logical: Complete vs incremental
  - Physical: Online vs Offline
- Also “Ingestion” in BigData

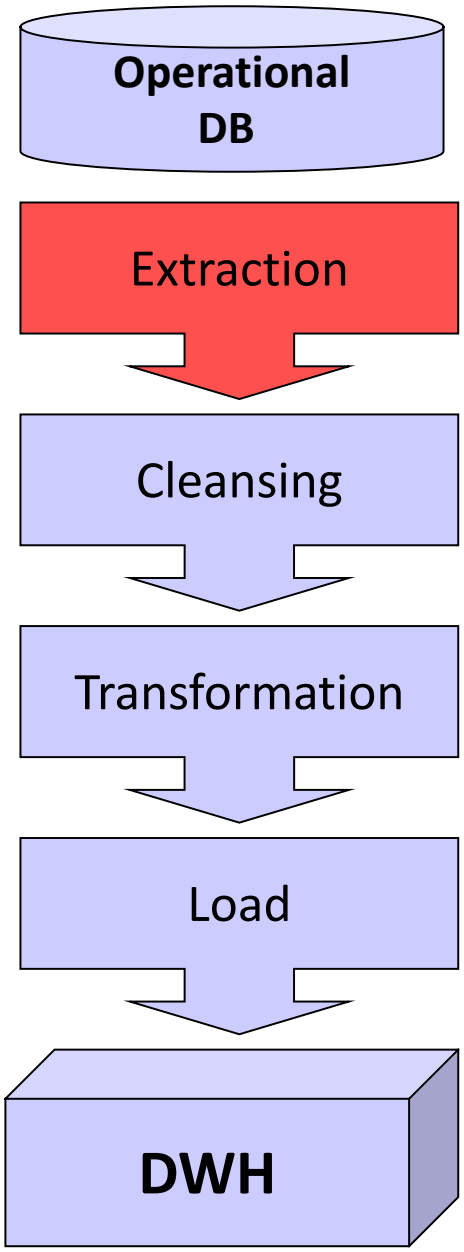
ETL



- Complete extraction
  - Capture all the source data in an instant of time
  - The extraction reflects all available data in the source system, it is not necessary to look at the source changes from the last extraction.
  - The source data are taken “as if”, and it does not require additional logic (e.g. timestamps).
  - Example: dumping one table, or SQL that retrieves the entire table.
  - Moving from relational to hadoop: Sqoop

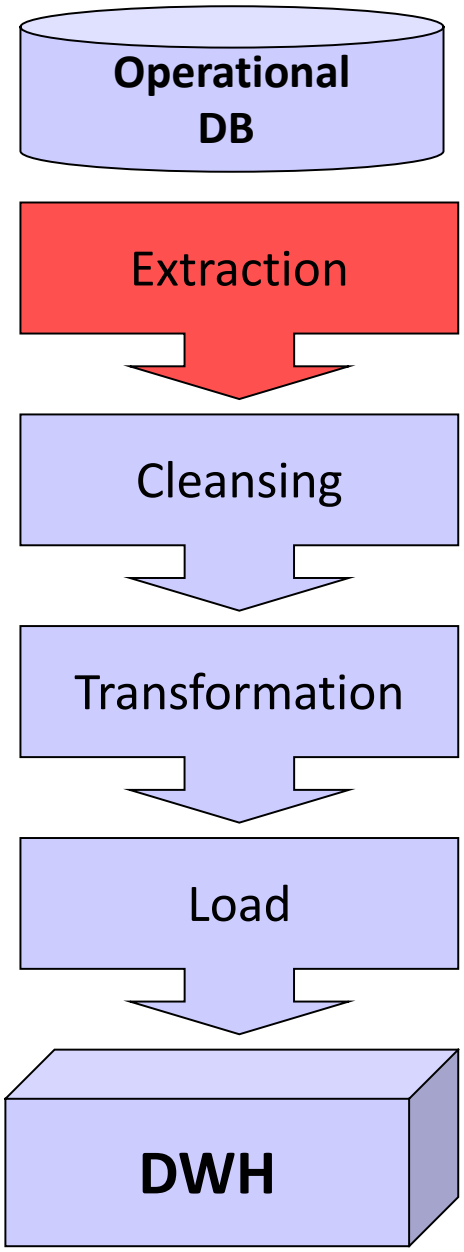


- Incremental extraction
  - At a specific point in time. One event triggers it.
  - Only changed data will be extracted.
  - Identify the information from one point changing in time CDC (Change Data Capture):
    - Timestamps on row
    - Version number on row
    - Status on row
    - Triggers (on database or application)
    - Logs (on database or application)
      - Powerful tools for this choice
  - Related to Slowly Changing Dimension
  - Pull Vs Push
  - Real time or event oriented processing: eg. Flume



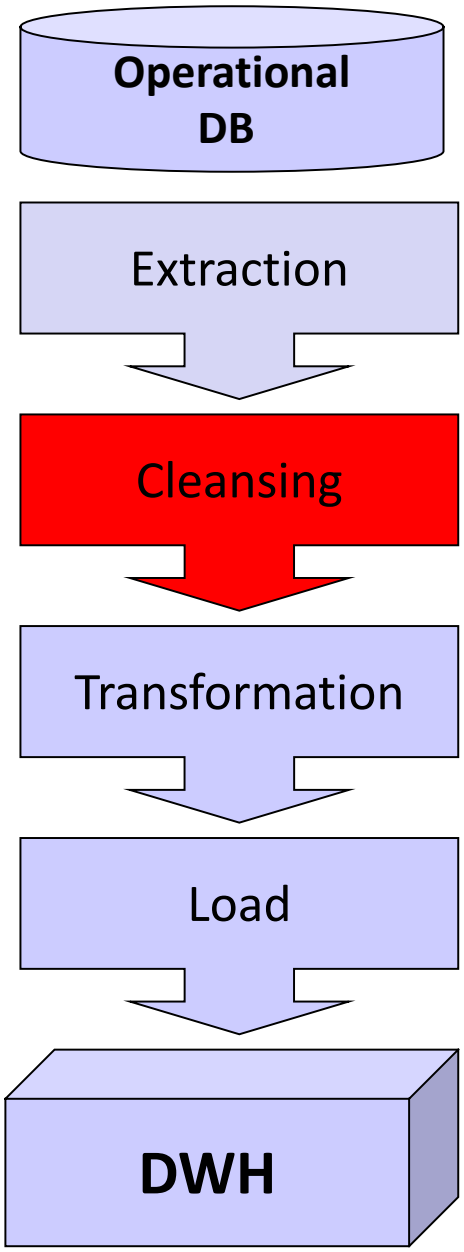
- Online extraction
  - Connect directly to the source or an intermediate system that stores data (eg, snapshots or change tables).
  - The intermediate system is not necessarily different from the source system.
  - (See previous Incremental extraction)

ETL



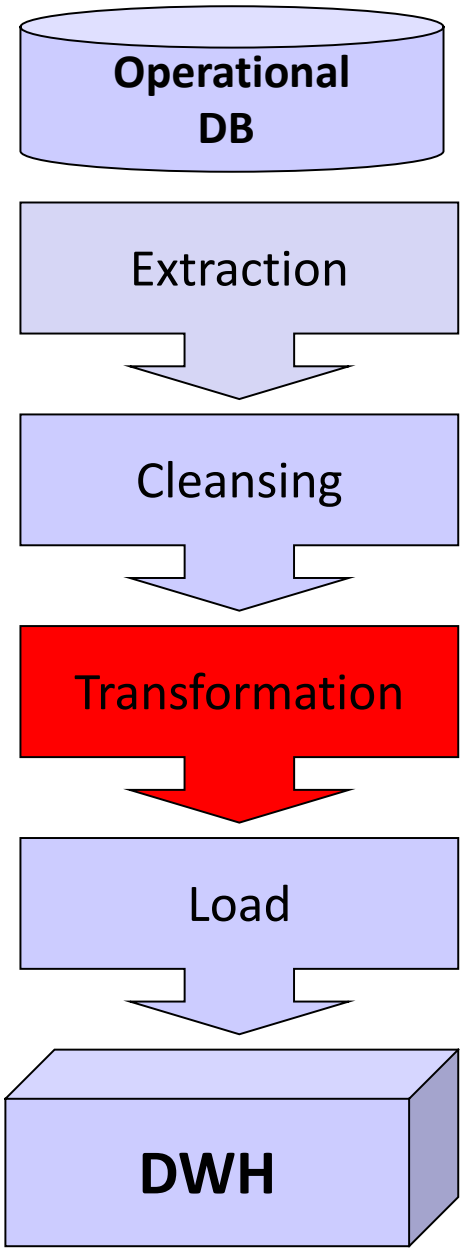
- Offline extraction
  - Data not extracted directly from the source system. It runs outside the original source system.
  - The data have an existing structure or are created by a dumping routine.
    - Flat files. Data in a generic format, eg csv. Includes Extra info on format.
    - Dump files. DBMS specific format. Extra info on SQL + format.
    - Redo logs and archive logs. Transportable tablespaces.

ETL



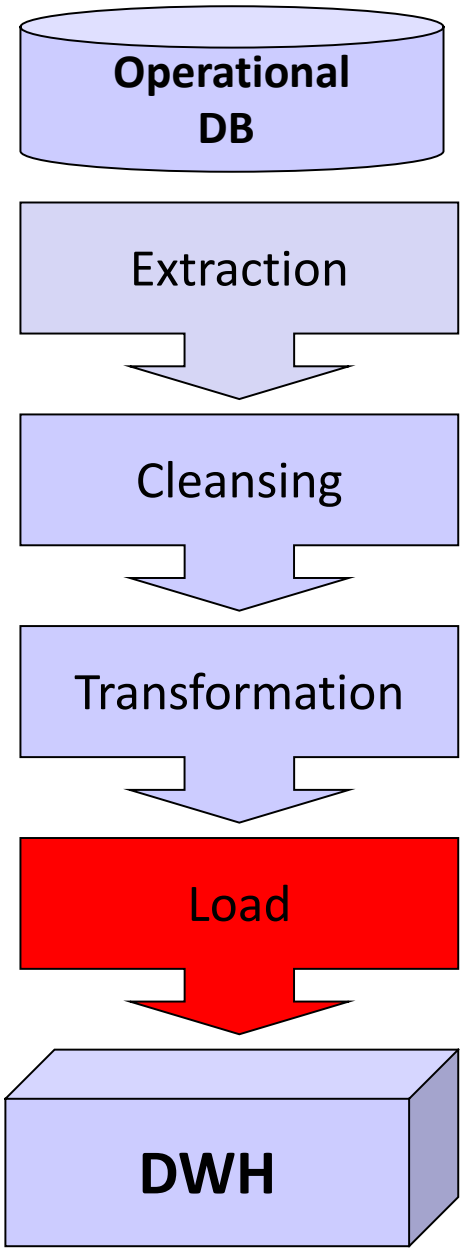
- Cleansing = ensure data quality
- Correct mistakes: format errors, wrong dates, misuse of fields, duplicate data, inconsistencies, restore referential integrity
- Also: decoding, reformatting, add timestamps, converting data, key generation, etc.
- Other words: Wrangling, munging, tidying

ETL



- Transform = convert data from the format of an operational system to the DWH format
  - Record Level:
    - Selection - partitioning
    - Joining - combination
    - Aggregation - precalculation
  - Field level:
    - Single field - from field to field
    - Multiple field - from multiple fields to a single field or vice versa
- Type of data transformation
  - Gradual
  - Pipelined
    - Multiple steps grouped.
    - It is not necessary to use intermediate tables.

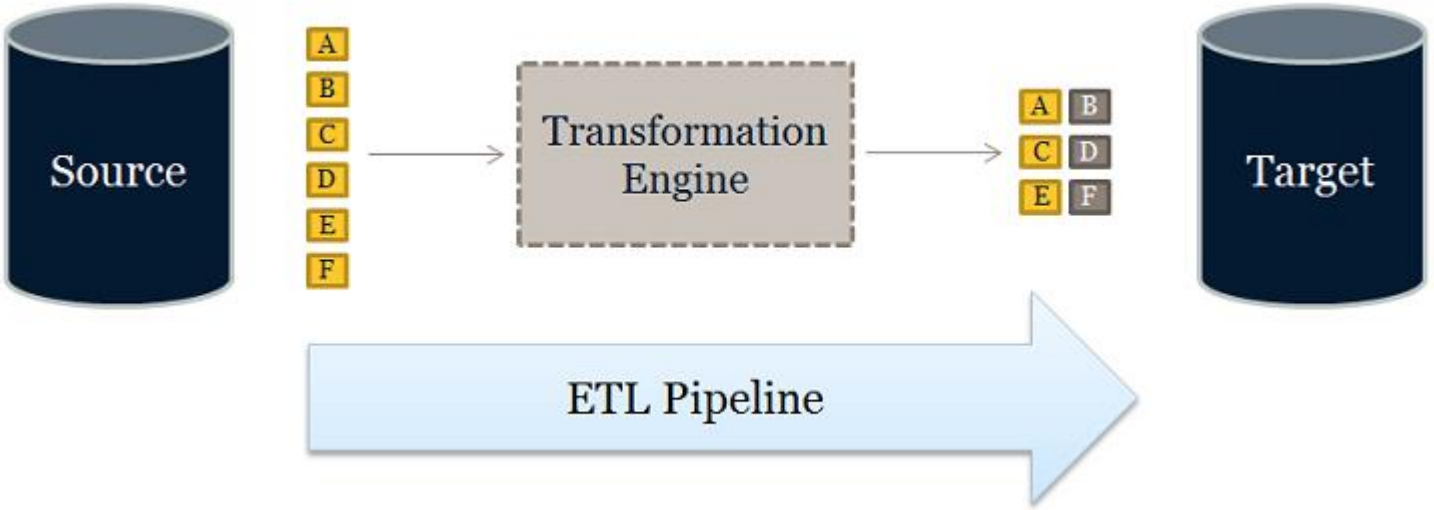
ETL

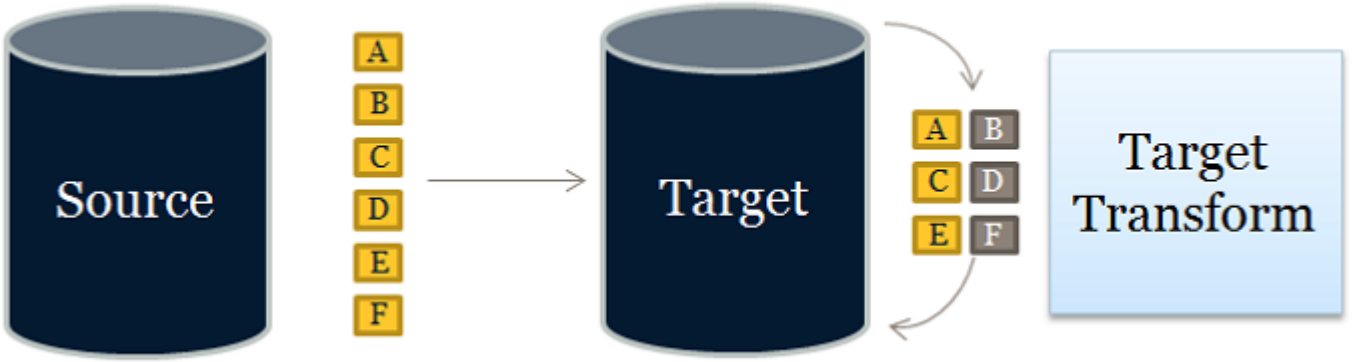


- Load = enter data into the DW and create indexes
- Refresh method: determination of refresh intervals
  - Remove unused old data
- Update method: only changes in the sources are taken into account
- Loading mechanisms
  - SQL * Loader (pgload and COPY in Postgres)
  - External tables
  - OCI API and Direct-Path
  - Export / Import (pgdump in Postgres)
  - Sqoop

ETL vs ELT (note the order of letters)

- ETL Classic vs ELT “Big Data” approach





ETL vs ELT (note the order of letters)

- ELT:
  - Data lakes
    - Raw data in staging in DW (only filter unneeded): Extra space
    - Faster
    - For future there are more data (not only the datamart)
  - Schemaless / schema on read
  - Less processing on the source since no transforming is being done
  - More processing when consuming (need to transform)
  - Languages with limited processing: hive, pig, …
  - More scalability and better I/O than ETL if classic db is used
  - Batch or event oriented

ETL vs ELT (note the order of letters)

- Classic ETL
  - Data marts: Cubes / Schema on write
  - Only clean and quality data
  - Staging not needed: Less space
  - More processing on extraction: a pipeline
    - Much slower
  - GUI tools: abstraction independent from language and platform, very rich and powerful transformations

ETL tools

- An integration tool needs:
  - Many interfaces
  - Transformations
  - Quality: validation, filters, …
  - Open Source:
    - Apache Airflow, Apache Kafka, Apache Nifi, Pentaho Data Integration, Talend OS, …
  - Commercial:
    - Hevo Data, Informatica Power Center, SnapLogic, Jitterbit, Oracle Data Integrator, Mulesoft, IBM Datastage, Microsoft SQL Server Integration Services, …