Universitas Esa Unggul

Smart, Creative and Entrepreneurial

Data Warehouse

Munawar, PhD

Session 09

ETL
(Extract Transform Loading)

Agenda

- What is ETL?
- Extraction
- Transformation
- Loading

What is ETL ?

ETL Overview

- Extraction Transformation Loading – ETL
- To get data out of the source and load it into the data warehouse – simply a process of copying data from one database to other
- Data is extracted from an OLTP database, transformed to match the data warehouse schema and loaded into the data warehouse database
- Many data warehouses also incorporate data from non-OLTP systems such as text files, legacy systems, and spreadsheets; such data also requires extraction, transformation, and loading
- When defining ETL for a data warehouse, it is important to think of ETL as a process, not a physical implementation

ETL …

- ETL is often a complex combination of process and technology that consumes a significant portion of the data warehouse development efforts and requires the skills of business analysts, database designers, and application developers
- It is not a one time event as new data is added to the Data Warehouse periodically – monthly, daily, hourly
- Because ETL is an integral, ongoing, and recurring part of a data warehouse
  - Automated
  - Well documented
  - Easily changeable

When Should we ETL?

- **When** should we ETL?
  - Periodically (e.g., every night, every week) or after significant events
  - Refresh policy set by administrator based on user needs and traffic
  - Possibly different policies for different sources
  - Rarely, on every update (real-time DW)
    - Not warranted unless warehouse data require current data
      (up to the minute stock quotes)

ETL...

- ETL is used to integrate heterogeneous systems
  - With different DBMS, operating system, hardware, communication protocols
- ETL challenges
  - Getting the data from the source to target as fast as possible
  - Allow recovery from failure without restarting the whole process
- This leads to balance between **writing data** to staging tables or **keeping it in memory**

Staging Area

- **Staging area,** basic rules
  - Data in the staging area is owned by the ETL team
    - Users are not allowed in the staging area at any time
  - Reports cannot access data from the staging area
  - Only ETL processes can write to and read from the staging area

Staging Area

- Staging area **structures** for holding data
  - Flat files
  - XML data sets
  - Relational tables

Data Structure

- **Flat files**
  - ETL tools based on scripts, such as Perl, VBScript or JavaScript
  - Advantages
    - No overhead of maintaining metadata as DBMS does
    - Sorting, merging, deleting, replacing and other data-migration functions are much faster outside the DBMS
  - Disadvantages
    - No concept of updating
    - Queries and random access lookups are not well supported by the operating system
    - Flat files cannot be **indexed for fast lookups**

Data Structure…

- When should **flat files** be used?
  - Staging source data for safekeeping and recovery
    - Best approach to restart a failed process is by having data dumped in a flat file
  - Sorting data
    - Sorting data in a file system may be more efficient as performing it in a DBMS with Order By clause
    - **Sorting is important:** a huge portion of the ETL processing cycles goes to sorting

Data Structure …

- When should **flat files** be used?
  - Filtering
    - Using grep-like functionality
  - Replacing text strings
    - Sequential file processing is much faster at system-level than using a database

Data Structure

- **XML Data** sets
  - Used as common format for both input and output from the ETL system
  - Generally, not used for persistent staging
  - Useful mechanisms
    - XML schema (successor of DTD)
    - XQuery, XPath
    - XSLT

Data Structure…

- **Relational tables**
  - Using tables is most appropriate especially when there are no dedicated ETL tools
  - Advantages
    - Apparent metadata: column names data types and lengths, cardinality, etc.
    - Relational abilities: data integrity as well as normalized staging
    - Open repository/SQL interface: easy to access by any SQL compliant tool
  - Disadvantages
    - Sometimes slower than the operating file system

Staging Area - Storage

- **How is the staging area designed?**
  - Staging database, file system, and directory structures are set up by the DB and OS administrators based on ETL architect estimations e.g., **tables volumetric worksheet**

Table Name	Update strategy	Load frequency	ETL Job	Initial row count	Avg row length	Grows with	Expected rows/mo	Expected bytes/mo	Initial table size	Table Size 6 mo. (MB)
S_ACC	Truncate/ Reload	Daily	SAcc	39,933	27	New account	9,983	269,548	1,078,191	2.57
S_ASSETS	Insert/ Delete	Daily	SAssets	771,500	75	New assets	192,875	15,044,250	60,177,000	143.47
S_DEFECT	Truncate/ Reload	On demand	SDefect	84	27	New defect	21	567	2,268	0.01

Extraction

Data Extraction

- **Data Extraction**
  - Data needs to be taken from a data source so that it can be put into the DW
    - Internal scripts/tools at the data source, which **export** the data to be used
    - External programs, which **extract** the data from the source
  - If the data is exported, it is typically exported into a **text file** that can then be brought into an intermediary database
  - If the data is extracted from the source, it is typically transferred directly into an **intermediary database**

Data Extraction…

- **Steps** in data extraction
  - Initial extraction
    - Preparing the **logical map**
    - First time data extraction
  - Ongoing extraction
    - Just **new** data
    - **Changed** data
    - Or even **deleted** data

Data Extraction…

- **Logical map** connects the original source data to the final data
- **Building the logical map**: first identify the data sources
  - Data discovery phase
    - Collecting and documenting source systems: databases, tables, relations, cardinality, keys, data types, etc.
  - Anomaly detection phase
    - NULL values can destroy any ETL process, e.g., if a foreign key is NULL, joining tables on a NULL column results in data loss, because in RDB NULL ≠ NULL

Data Extraction…

- Example of solving NULL anomalies
  - If NULL on **foreign key** then use **outer joins**
  - If NULL on other columns then create a **business rule** to replace NULLs while loading data in DW
- Ongoing Extraction
  - Data needs to be **maintained** in the DW also after the initial load
    - Extraction is performed on a regular basis
    - Only changes are extracted after the first time
- **How do we detect changes?**

Ongoing Extraction

- Detecting changes (new/changed data)
  - Using audit columns
  - Database log scraping or sniffing
  - Process of elimination
- Using **audit columns**
  - Store the date and time a record has been added or modified at
  - Detect changes based on **date stamps higher than the last extraction date**

Detecting Changes

- **Log scraping**
  - Takes a snapshot of the database **redo log** at a certain time (e.g., midnight) and finds the transactions affecting the tables ETL is interested in
  - It can be problematic when the redo log gets full and is emptied by the DBA
- **Log sniffing**
  - Pooling the redo log at small time granularity, capturing the transactions on the fly
  - The better choice: suitable also for real-time ETL

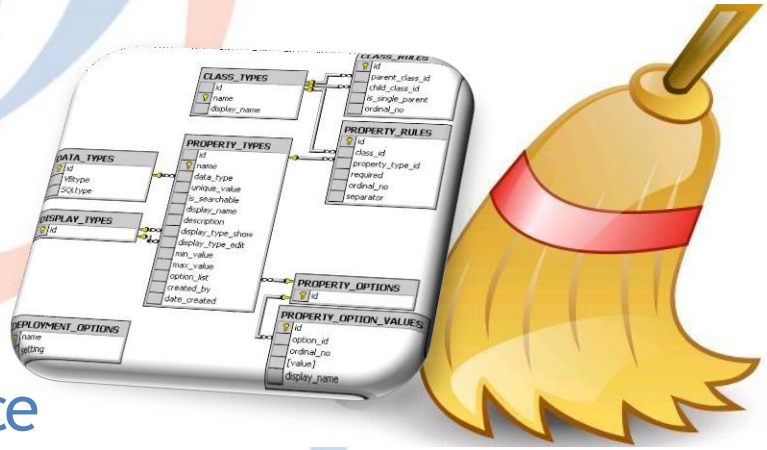
Detecting Changes…

- **Process of elimination**
  - Preserves exactly one copy of each previous extraction
  - During next run, it compares the entire source tables against the extraction copy
  - Only differences are sent to the DW
  - Advantages
    - Because the process makes row by row comparisons, it is **impossible to miss data**
    - **It can also detect deleted rows**

Transformation

Data Transformation

- **Data transformation**
  - Uses **rules**, **lookup tables**, or combinations with other data, to convert source data to the desired state
- 2 major steps
  - Data Cleaning
    - May involve manual work
    - Assisted by artificial intelligence algorithms and pattern recognition
  - Data Integration
    - May also involve manual work



Data Cleansing

- Extracted data can be dirty. How does clean data look like?
- **Data Quality** characteristics:
  - **Correct**: values and descriptions in data represent their associated objects truthfully
    - E.g., if the city in which store A is located is Braunschweig, then the address should not report Paris
  - **Unambiguous**: the values and descriptions in data can be taken to have only one meaning

Data Cleansing…

- **Consistent:** values and descriptions in data use one constant notational convention
  - E.g., Jakarta can be expressed as Jkt or DKI, by our employees. Consistency means using just Jakarta in all our data
- **Complete**
  - Individual values and descriptors in data have a value (not null)

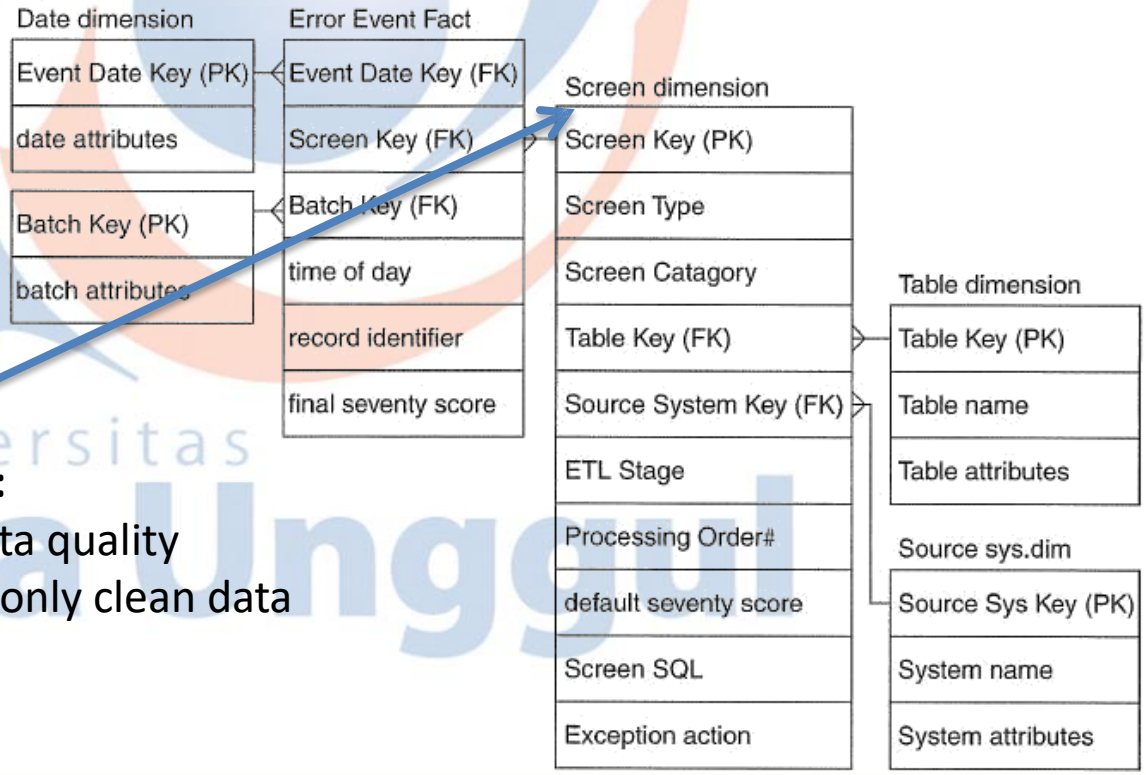
Data Cleansing…

- The **data cleaning engine** produces 3 main deliverables:
  - **Data-profiling results:**
    - Meta-data repository describing schema definitions, business objects, domains, data sources, table definitions, data rules, value rules, etc.
    - Represents a quantitative assessment of original data sources

Cleaning Deliverables

– Error event table

- Structured as a dimensional star schema
- Each data quality error identified by the cleaning subsystem is inserted as a row in the error event fact table



Data Quality Screen:
- Status report on data quality
-Gateway which lets only clean data go through

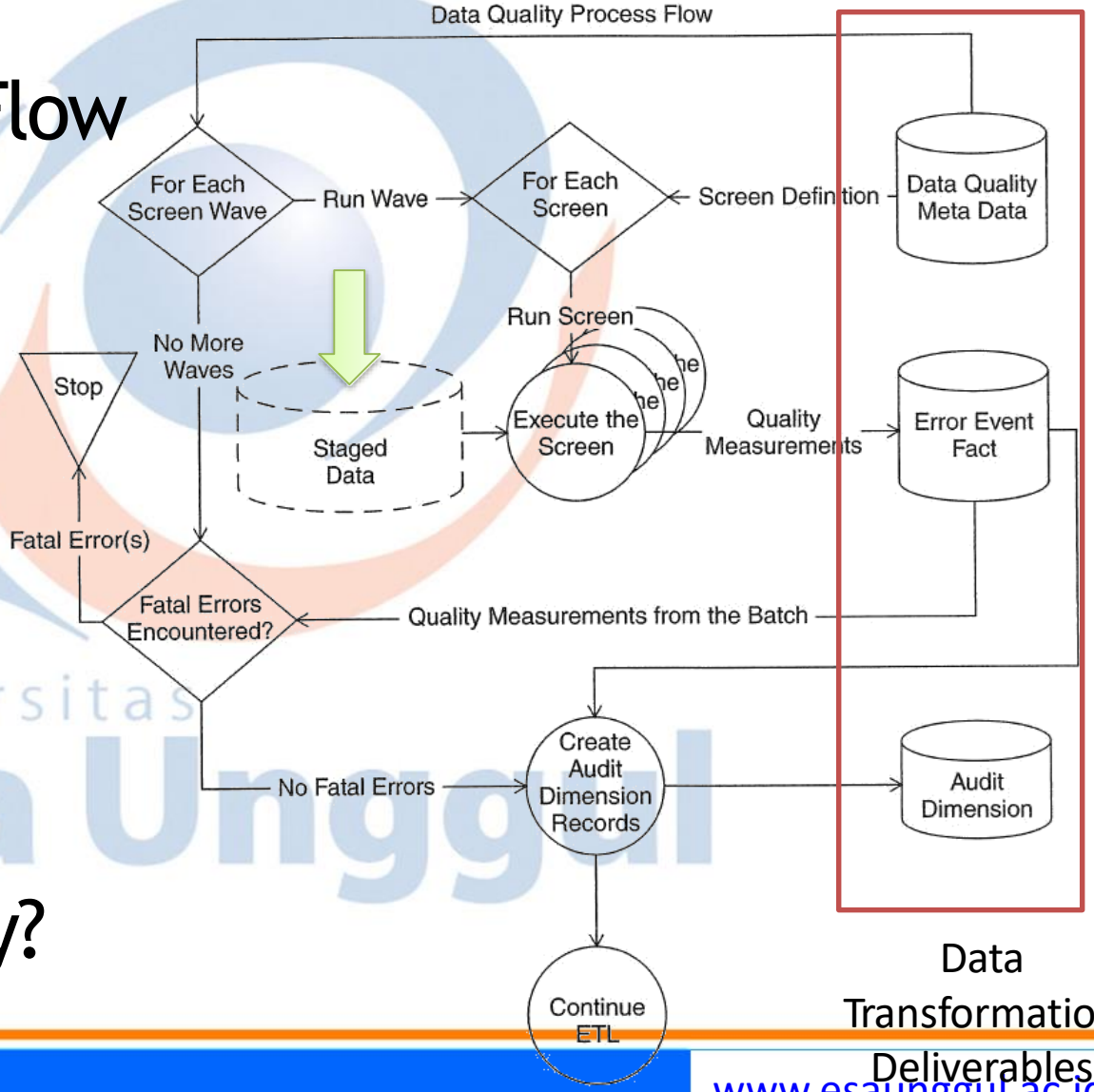
Cleaning Deliverables…

- **Audit dimension**
  - Describes the **data-quality** context of a fact table record being loaded into the DW
  - Attached to **each fact record**
  - Aggregates the information from the error event table on a per record basis

Audit key (PK)
Completeness category (text)
Completeness score (integer)
Number screens failed
Max severity score
Extract timestamp
Clean timestamp
…

Data Cleaning

- Overall Process Flow

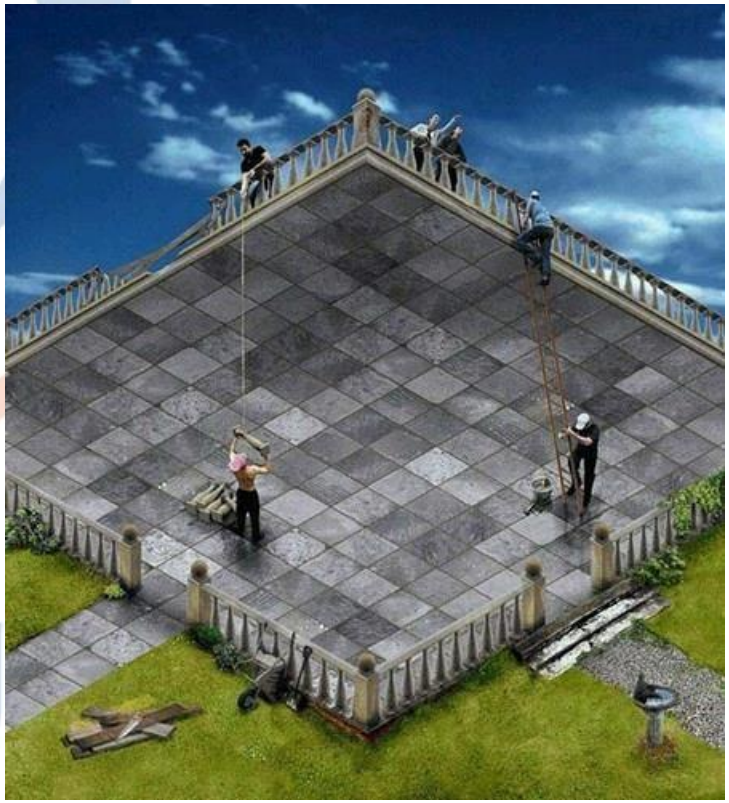


- Is data really dirty?

Data Transformation Deliverables

Cleaning Engine

- Core of the data cleaning engine: anomaly detection phase
  - **Data anomaly** is a piece of data which doesn't fit into the domain of the rest of the data it is stored with
    - E.g., Male named Endang ???



Anomaly Detection

- Anomaly detection
  - Count the rows in a table while grouping on the column in question e.g.,
    - SELECT city, count(*) FROM order_detail GROUP BY city

City	Count(*)
Bremen	2
Berlin	3
WOB	4,500
BS	12,000
HAN	46,000
…	

Anomaly Detection…

- What if our table has 100 million rows with 250,000 distinct values?
  - Use **data sampling** e.g.,
    - Divide the whole data into 1000 pieces, and choose 1 record from each
    - Add a random number column to the data, sort it an take the first 1000 records
    - Etc.
  - **Common mistake is to select a range of dates**
    - Most anomalies happen **temporarily**

Loading

Loading

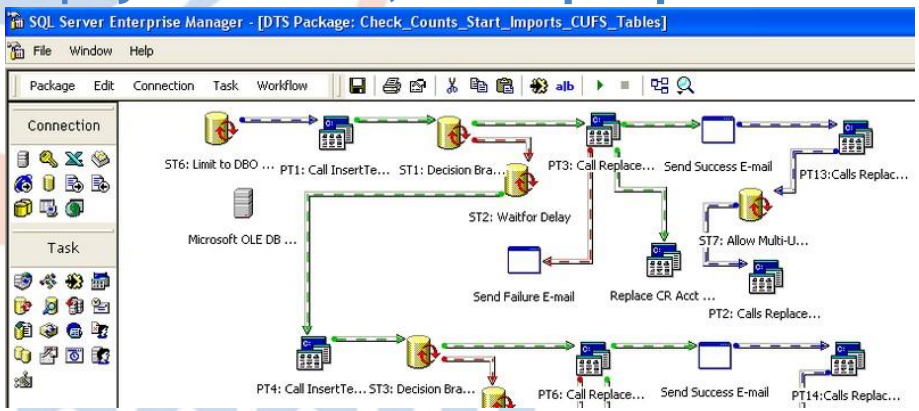
- The **loading** process can be broken down into 2 different types:
  - Initial load
  - Continuous load (loading over time)

Loading…

- **Issues**
  - Huge volumes of data to be loaded
  - Small time window available when warehouse can be taken off line (usually nights)
  - When to build index and aggregated tables
  - Allow system administrators to monitor, cancel, resume, change load rates
  - **Recover gracefully** - restart after failure from where you were and without loss of data integrity

Loading…

- **Initial Load**
  - Deliver dimensions tables
    - Create and assign surrogate **keys**, each time a new cleaned and conformed dimension record has to be loaded
    - Write dimensions to disk as physical tables, in the **proper dimensional format**
  - Deliver fact tables
    - Utilize bulk-load utilities
    - Load in parallel
  - Tools
    - DTS – DataTransformation Services (set of tools)
    - bcp utility – batch copy
    - SQL* Loader



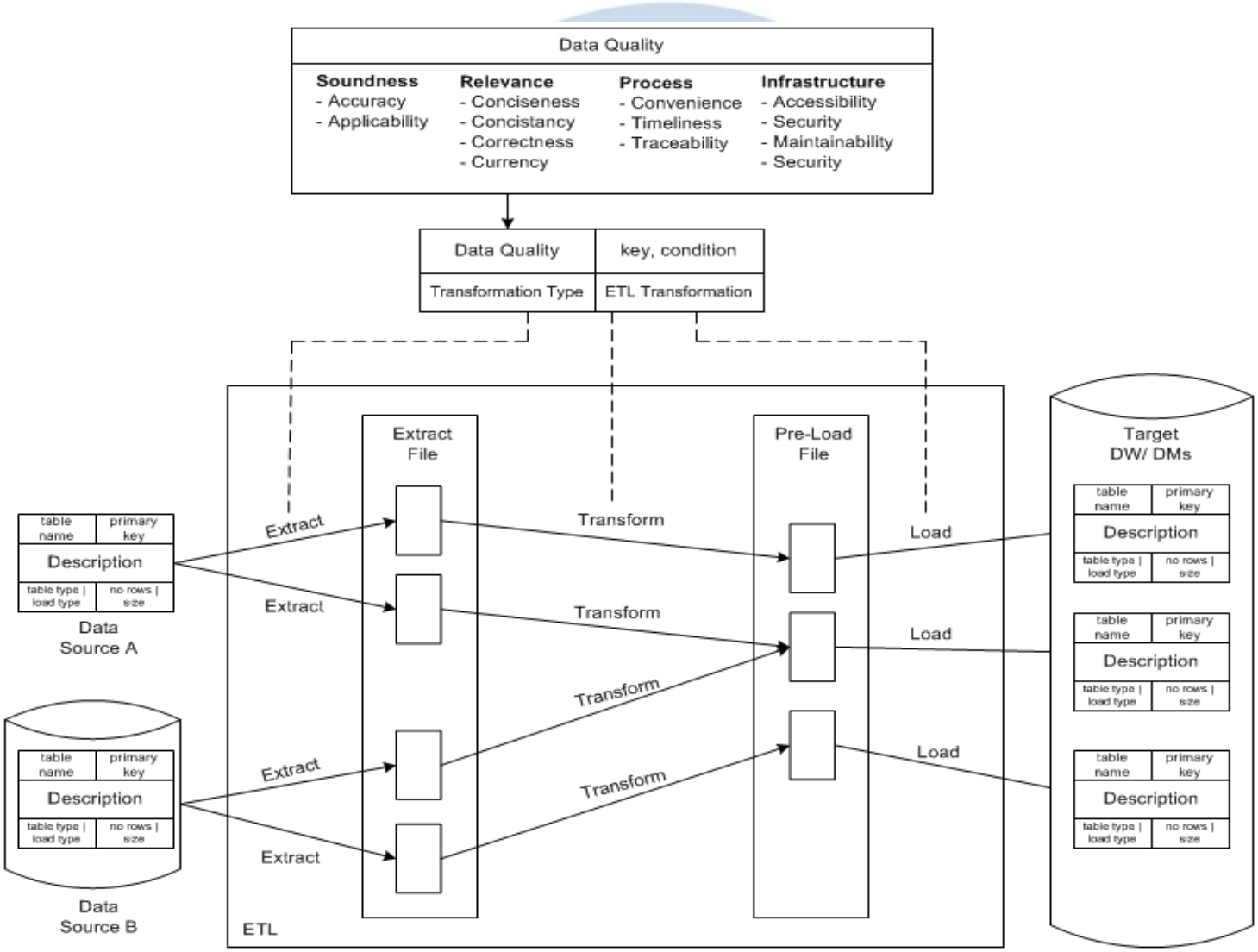
Loading…

- **Continuous load** (loading over time)
  - Must be scheduled and processed in a specific order to maintain integrity, completeness, and a satisfactory level of trust (if done once a year… the data is obsolete)
  - Should be the most carefully planned step in data warehousing or can lead to:
    - Error duplication
    - Exaggeration of inconsistencies in data

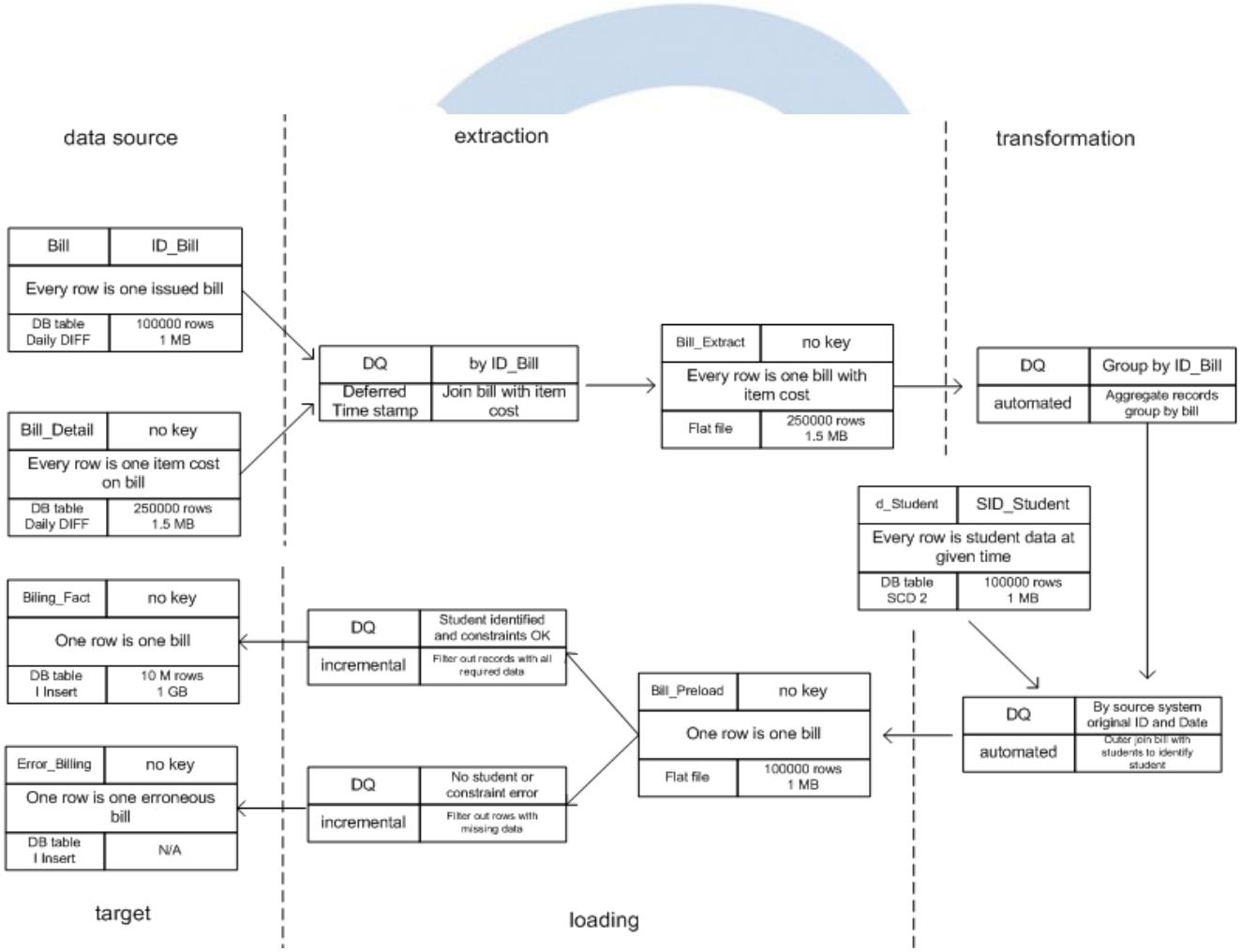
Loading…

- Continuous load of **facts**
  - Separate updates from inserts
  - Drop any indexes not required to support updates
  - Load updates
  - Drop all remaining indexes
  - Load inserts through bulk loaders
  - Rebuild indexes

Incorporating DQ in DW



Sample ETL for Student Billing



Thank You…