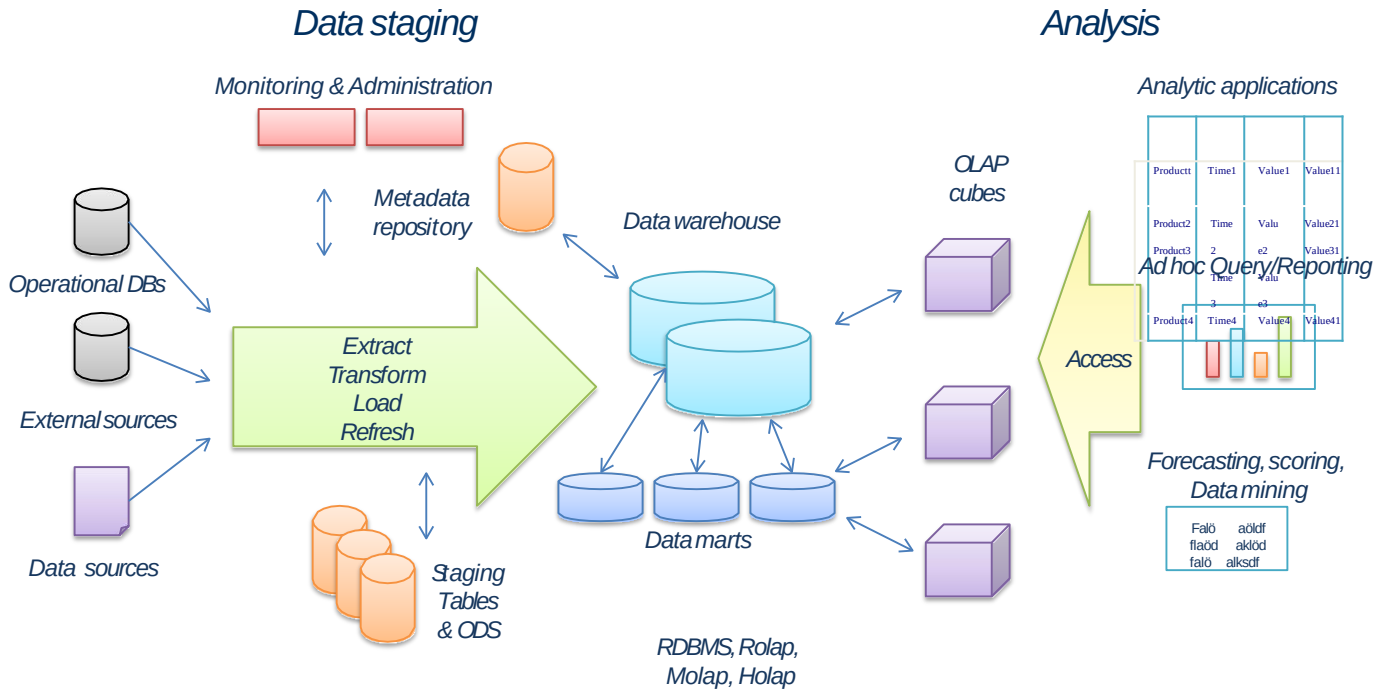
Presentation:
ETL

Erik Perjons

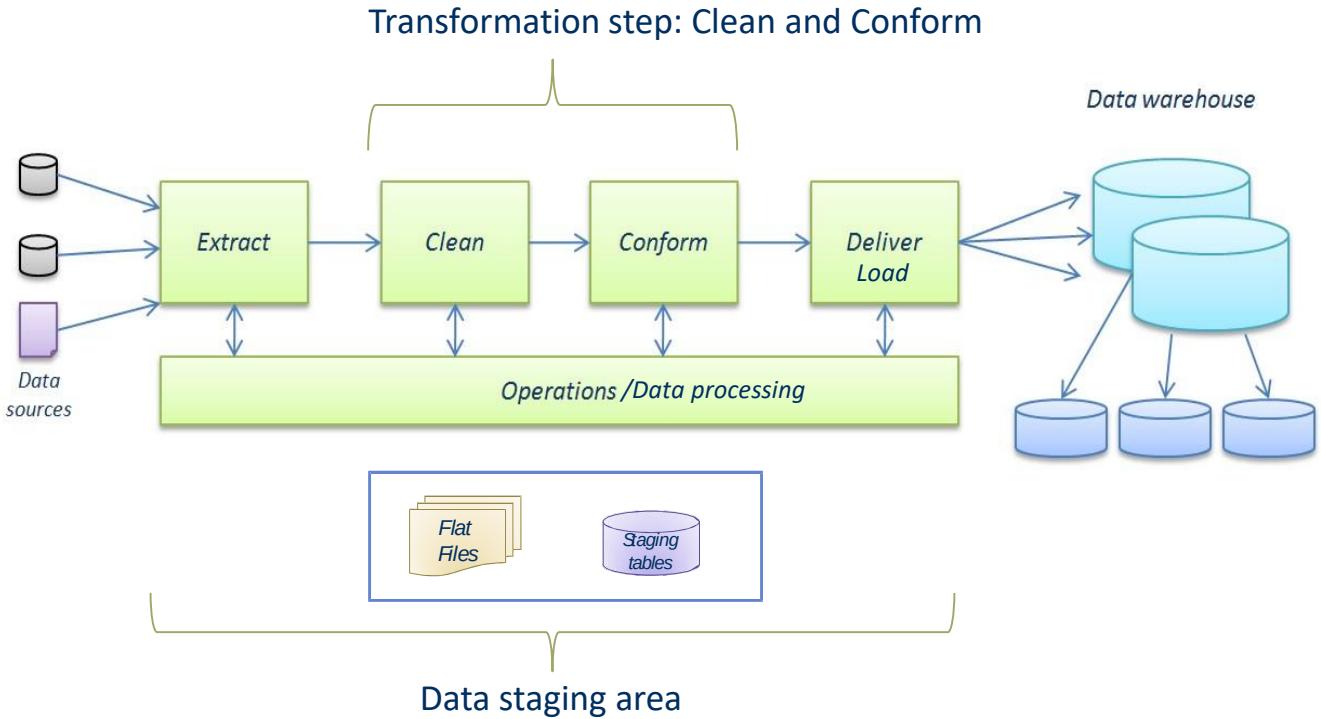
DSV, Stockholm University

DW/BI Lifecycle Overview

DW Architecture

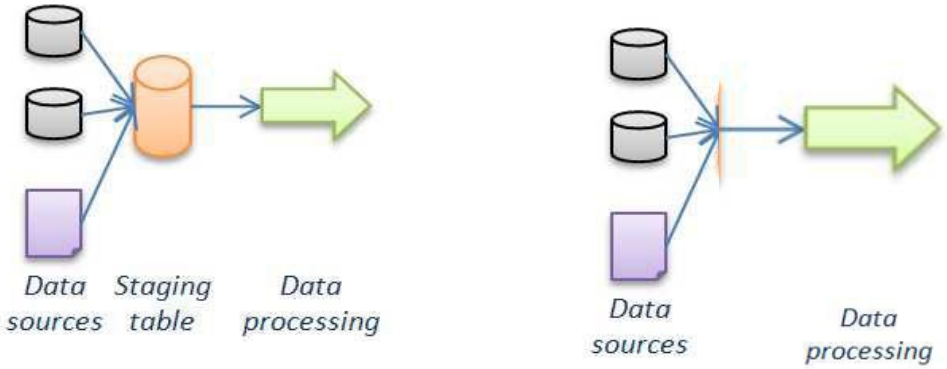


The ETL Process Steps



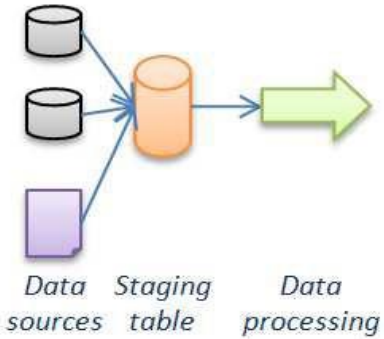
ETL design decision

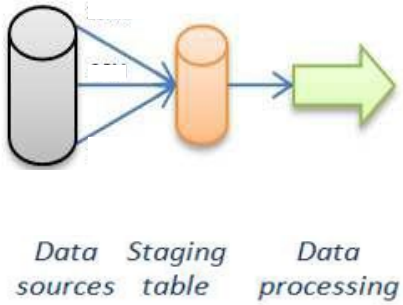
- Use a staging table or not



ETL design decision

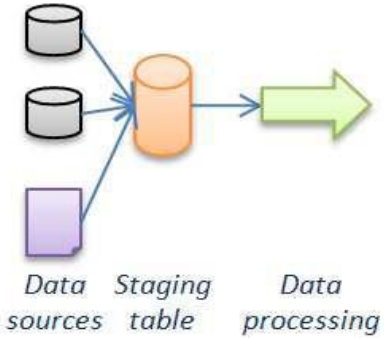
- Use one staging table for several sources or one staging table for one source

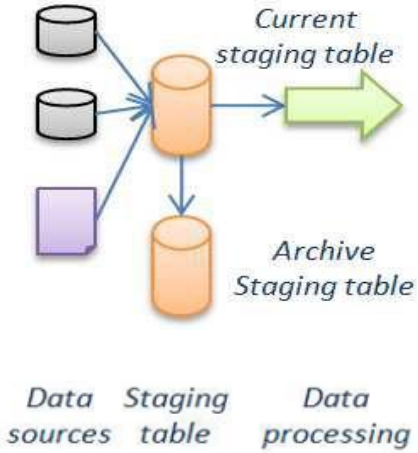




ETL design decision

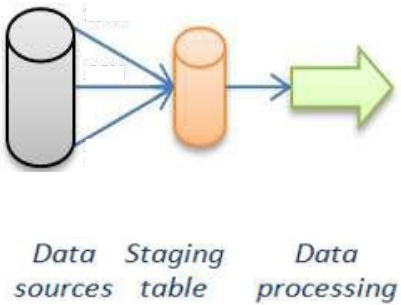
- Archive the data in the staging table in another database or not

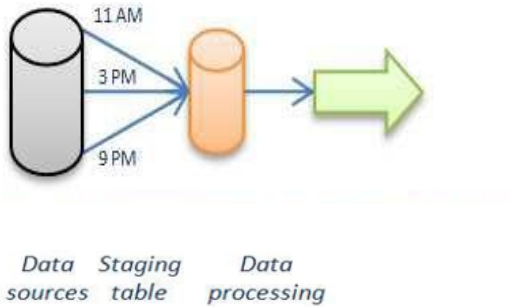




ETL design decision

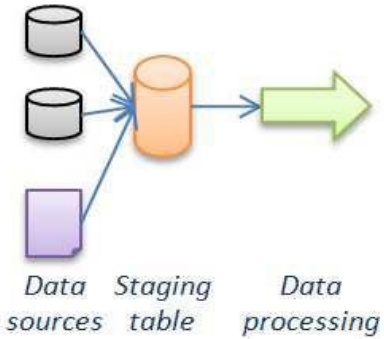
- Load the data staging table at one or several occasions

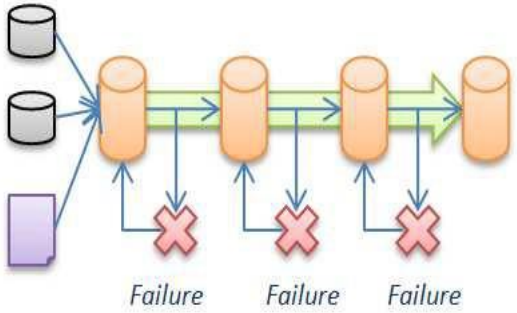




ETL design decision

- Use one or several data staging tables during data processing

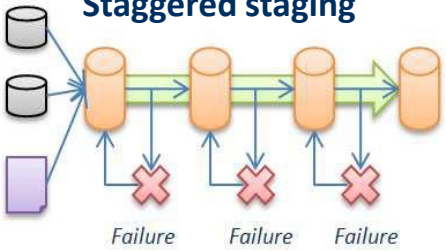




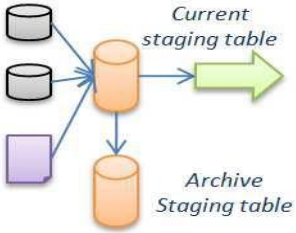
Data staging pattern

- The different design decision can result in a number of well-known data staging pattern

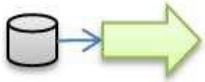
Staggered staging



Persistent staging

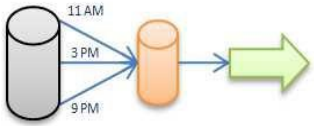


Pipeline staging



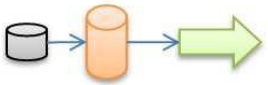
Data is stored in memory buffers during process execution

Chuncked ackumulated staging



Chunks of data extracted at different times

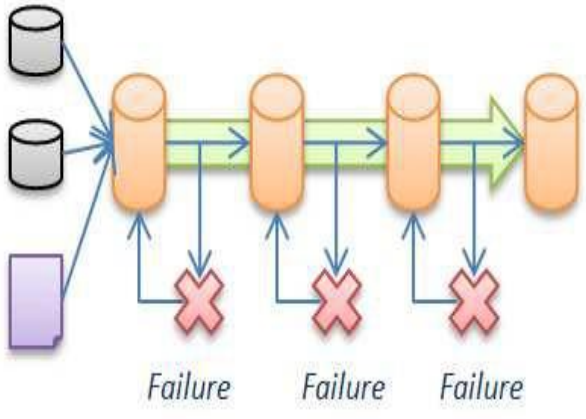
Ackumulated staging



Data events ackumulated over time

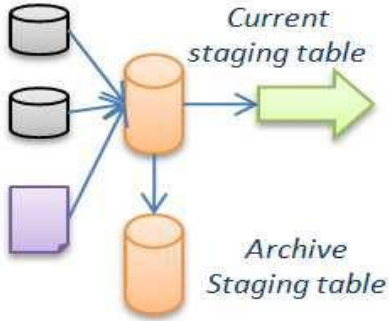
Staggered staging

- Data is stored in staging tables throughout the ETL process, where each staging table contains data with higher value
- This pattern can support recovery, that is, the whole ETL process does not need to be processed again when a failure is happening



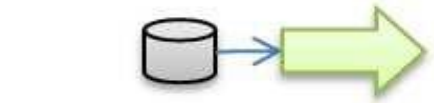
Persistent staging

- A copy of the data in the staging table is stored in another database
- This pattern is used for data auditing and other reasons



Pipeline staging

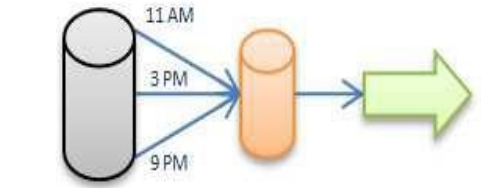
- Data are not stored in data staging tables, instead data are stored in memory bufferts during processing
- This pattern can be used for higher data throughput, but descrease the ability to handle changes in the data



Data is stored in memory buffers during process execution

Chunked Accumulated Staging

- Data is extracted from source system in smaller chunks at different times
- This pattern is often used in high transaction environments



Chunks of data extracted at different times

Accumulated Staging

- The staging table is used for detect and accumulate data changes from extracted data
- This pattern is often used when the source systems cannot provide info of which data that are new since last extraction

Different aspects of ETL

Stockholms universitet

- **Design the ETL architecture, staging tables, script and transformation rules –** to enable the execution of the ETL process
- **One time historic load processing** – when the DW is loaded for the first time with records from the source systems
- **Incremental load processing** – when the DW is loaded daily with new records daily (or using some other time interval for load)

ETL process issues

- Kimball/Ross presents ***34 ETL Subsystem and Techniques*** to be applied in order to manage different kinds of issues related to the ETL process

Source system issues

- **Issue: How to check data quality in the source system?**
- **Solution:** Carry out **data profiling activities** using **a data profiling tool**
    - A data profiling tool can be used to measure the **conformance of data to data quality rules**, which need to be specified.
    - However, a data profiling tool can also be used to **discover possible data quality rules**, for example, the tool can detect that a column in the database always has a unique value, therefore, the tool suggests that that column should have a uniqueness data quality rule

Source system issues

- **Issue**: How to handle **low data quality identified** in the source systems?
- **Solutions:**
  - Send the source system owner a request for **improvement of data quality in the source system**
  - Specify requirements and **implement data quality improvement in the ETL process**

Extract issues

Stockholms universitet

- **Issue:** How **identify changes in the sources system to be extracted and loaded** into the data staging area, that is, how to perform so called **change data capture (CDC)**

- **Solution:**
  - **source system can include audit files** (created by for example triggers when data is inserted or changed) with date and time when a record was added or modified
  - **compare a yesterdays data with the today's data, record by record** – can be carried out in the source system (preferable) or inte data staging area (may require that the whole database to betransferred into the data staging area)
  - **use a checksum algoritm (CRC)** to check if a complex record is changed

Extract issues

- **Issue:** Which method to use for transfering data – **batch/file or streaming?**
- **Solution:** Extracts in form of **files has many advantages over streaming**:
  - the extraction is easy to rerun due to failure
  - the extraction is easy to encrypt and compress,
  - it is easy to verify that all data have been transfered

Data cleansing issues

Stockholms universitet

- **Issue:** How to **identify data quality issues** in the data loaded into **the data staging area**?
- **Solution: Carry out data profiling analysis** in form of **quality screens**, which are diagnostic filters in the data flow pipelines. Each quality screen is a test and if the test fail, an error event is dropped, such as identified NULL values (column screen, test data in one column), violation of referential integrity (structure screen, i.e., test relationship of data accross columns), or violation of business rules (business rules screen)

Data cleansing issues

- **Issue:** How to **handle data quality issues identified** in the data loaded into the data staging area?
- **Solution:** If an quality issue occures (that is, an error event is dropped) the following can be done:
    - Halt the data flow/process
    - Send the data into suspension
    - Tag the data (the preferred option, according to Kimball/Ross)

Data cleansing issues

- **Issue:** How to **document identified error events**?
- **Solutions:**
  - **Create a star join schema to be used in the data staging area** with an error event fact table and dimensions such as Batch dimension, Data dimensions, and Scrre dimension?
  - **Add an audit dimensions for each star join schema** with info about errors

Dimension building issues

- **Issue:** How to **mange attribute values that have changed** and thereby **differ from the values already stored in the DW**?
- **Solution:** Select and apply appropriate **Slowly changing dimension (SCD) techniques**, or select a **Minidimension**

Dimension building issues

Stockholms universitet

- **Issue:** How to **add a surrogate key** to the rows in the dimensions?
- **Solution:**
  - Introduce a robust mechanism for producing surrogate key, for example a **surrogate key generator** that independently generate keys for every dimensions.
  - Make use of a table in the data staging area where the natural keys and surrogate keys are mapped. This can be used to identify new rows to be added in the dimensional table: a new natural key value, means a row that needs to be added to the DW

Dimension building issues

- **Issue:** How to **create conformed dimensions from different source systems**?
- **Solution:** Data from multiple systems need to be combined and integrated according to some strategy, for example applying attribute mapping model

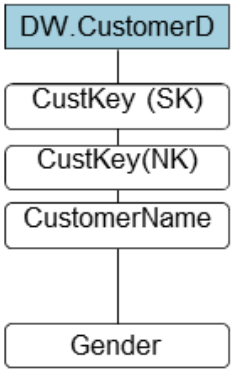
Attribute mapping model

- Attribute Mapping Model - make one attribute mapping model for each dimension
- Carry out the following steps:
  - Step 1: List the attributes for a dimensional table
  - Step 2: List the relevant attributes in the source databases
  - Step 3: Map the source attributes to the attributes in the dimensional table.
  - Step 4: Add transformation rules for the source attributes

Attribute mapping model

Stockholms universitet

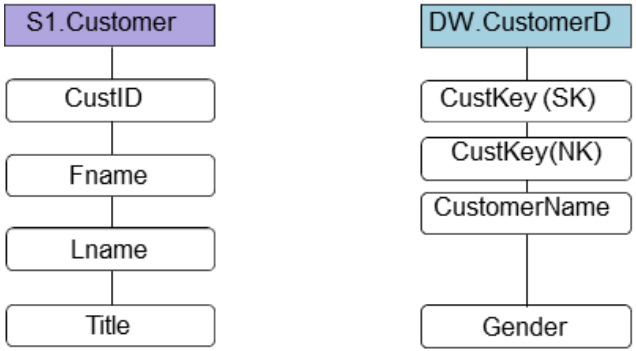
Step 1: List the attributes for a dimensional table



Attribute mapping model

Stockholms universitet

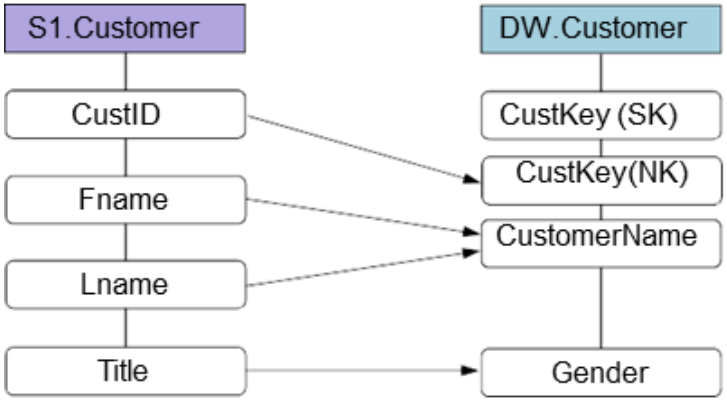
Step 2: List the relevant attributes in the source databases



Attribute mapping model

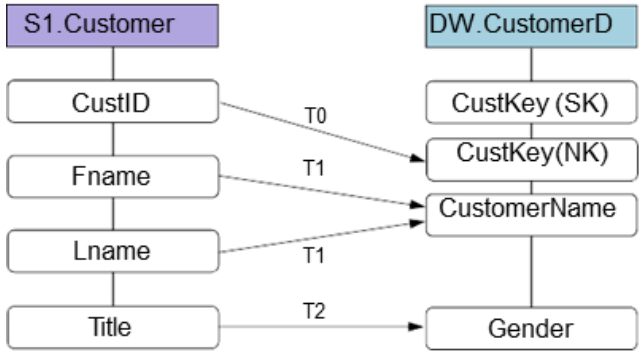
Stockholms universitet

Step 3: Map the source attributes to the attributes in the dimensional table



Attribute mapping model

Step 4: Add transformation rules for the source attributes

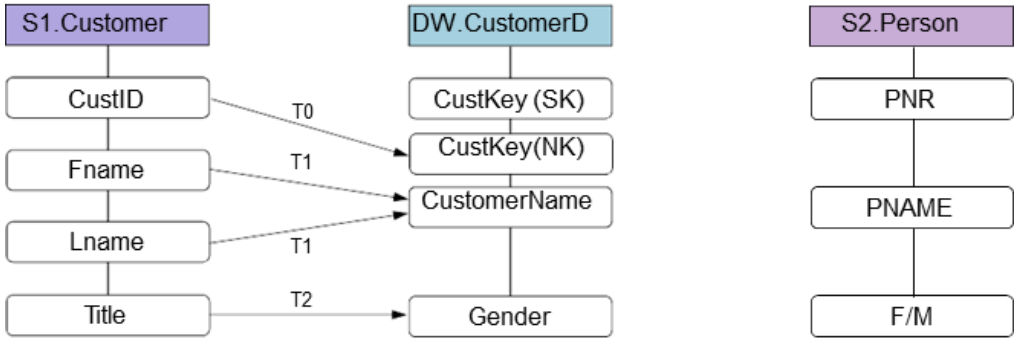


Transformation rules

T0 – move
T1 – concatenate (e.g. 'Anna' and 'Jennings' to 'Anna Jennings')
T2 – replace with a default value (e.g. 'Ms' and 'F' with 'female')

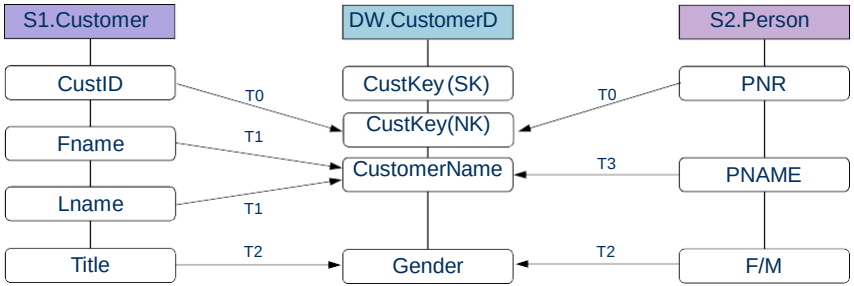
Attribute mapping model

Step 2: Add additional source: List the relevant attributes in the source databases



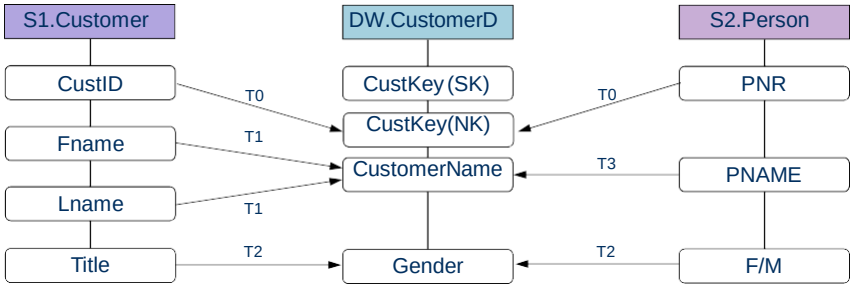
Attribute mapping model

Step 3: Map the source attributes to the attributes in the dimensional table



Attribute mapping model

Step 4: Add transformation rules for the source attributes



Transformation rules

T0 – move
T1 – concatenate (e.g. 'Anna' and 'Jennings' to 'Anna Jennings')
T2 – replace with a default value (e.g. 'Ms' and 'F' with 'female')
T3 – parse, clean, change case, concatenate (e.g. 'JENNINGS,ANNA' to 'Anna Jennings')

Fact building issues

- **Issue:** How to **add a new facts in the fact tables**?
- **Solution: Bulk loading (batch) new rows** in the fact table

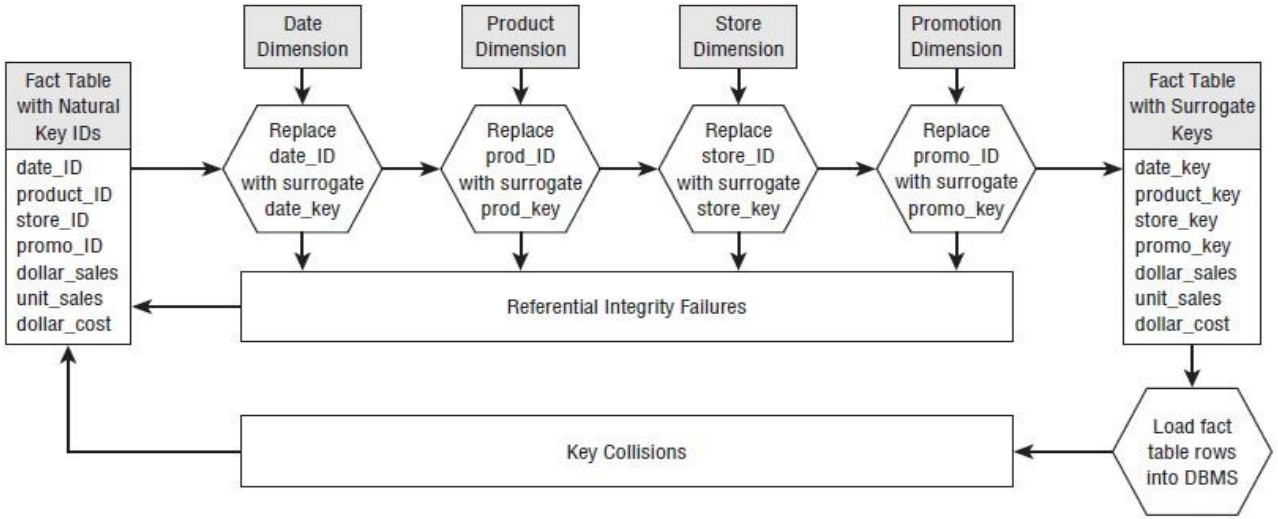
Fact building issues

- **Issue:** How to **maintain referential integrity** with the associated dimensional tables when loading the fact table with facts?
- **Solution: Replace the natural keys with appropate surrogate key existing in the dimensions**

Maintain referential integrity

Important steps:

- Assign surrogate key to new rows in the dimensional tables
- Populate data in dimensions
- Substitute natural key in fact table rows with proper surrogate key



Fact and dimension building issues

- **Issue:** How to **handle fact table data arriving much later than dimensional table data**?
- Solutions:
  - Search back in history to find the correct dimension keys and, thereby, rows

Fact and dimension building issues

Stockholms universitet

- **Issue:** How to **handle dimensional table data arriving later than fact table data**?
- Solutions:
  - Wait to load the facts table data if possible
  - Create a new customer dimension row with a surrogate key with a set of dummy attributes, that later will be changed (type 1 overwrite) when you get complete information of the customer
  - ….

Kimball & Ross: ETL process

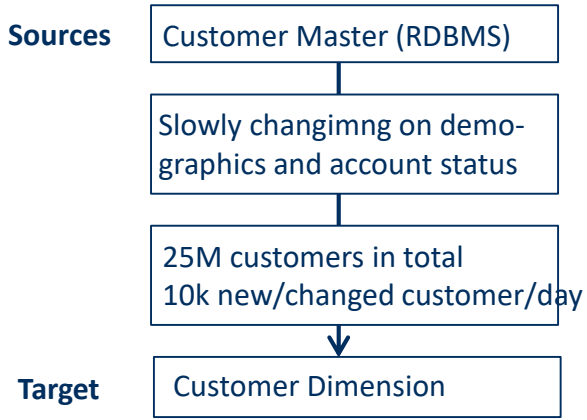
- Kimball/Ross suggest a ETL process, consisting of three ”subprocesses”, which consist of a number of steps
- **Subprocess 1:** Develop a Plan (Step 1-4)
- **Subprocess 2:** Develop One-Time Historic Load Processing (Step 5-6)
- **Subprocess 3:** Develop Incremental ETL Processing (Step 7-10

Subprocess 1: Develop a Plan (Step 1-4)

Step 1: Draw the High Level Plan

Stockholms universitet

- Draw a high-level plan showing sources and targets and imporant info about this transformation
- Document the overall ETL strategy
- Highlight issues and open problems



Step 2: Choose an ETL Tool

- Tools offers support for advanced and standardized ETL processes
- Some commercial tools:
  - Oracle Warehouse Builder
  - IBM DB2 Warehouse Manager
  - Microsoft Integration Services
- Some open source tools:
  - Talend
  - Enhydra Octopus
  - Clover.ETL

Step 3: Develop Default Strategies

Stockholms universitet

- Decide how extract from each major source system
- Decide staging design/patterns
- Decide how to handle data profiling
- Decide how to manage change in dimension attributes
- Decide how to monitoring performance

Step 4: Drill down by Target Tables

- Drill down to detailed descriptions of how to transform the data in the source tables to the data in the target table
- Design necessary transformations
  - How to transform source data to target data, including combining separate sources
  - How to mange hierachies
  - How to decode production codes with text descriptions
  - How to assign surrogate keys
  - How to handle NULL values
  - How to achieve referential integrity

Stockholms universitet

Subprocess 2: Develop One-Time Historic Load Processing (Step 5-6)

Step 5: Populate Dimension Tables with Historic Data

- Carry out extract, transform and load of data for the the dimensional tables

Step 6: Perform the Fact Table Historic Load

- Carry out extract, transform and load of data to the fact tables
- Create indexes

Stockholms universitet

Subprocess 3: Develop Incremental ETL Processing (Step 7-10)

Step 7: Dimensional Table Incremental Processesing

- Carry out extract, transform and load of data of new and changed rows for the dimensional tables
  - Identify new and changed dimensional rows/column values
  - Manage changed values of dimensional attributes
  - Add surrogate key

Step 8: Fact Table Incremental Processing

- Carry out extract, transform and load of data in form of new rows for the fact tables
- Handling data quality (data profiling)
- Manage referential integrity
- Create indexes

Step 9: Load Aggregate Table and OLAP load

- The aggregate tables and OLAP cubes are loaded with data created by carrying out queries that aggregate data in different dimensions

Step 10: ETL System Operations and Automation

- The ETL operations should ideally be run without human intervention – this is hard to attain but not impossible to get close