Unit 4 : Data Warehousing Technologies and Implementation

Lecturer : Bijay Mishra

Design of a Data Warehouse: A Business Analysis Framework

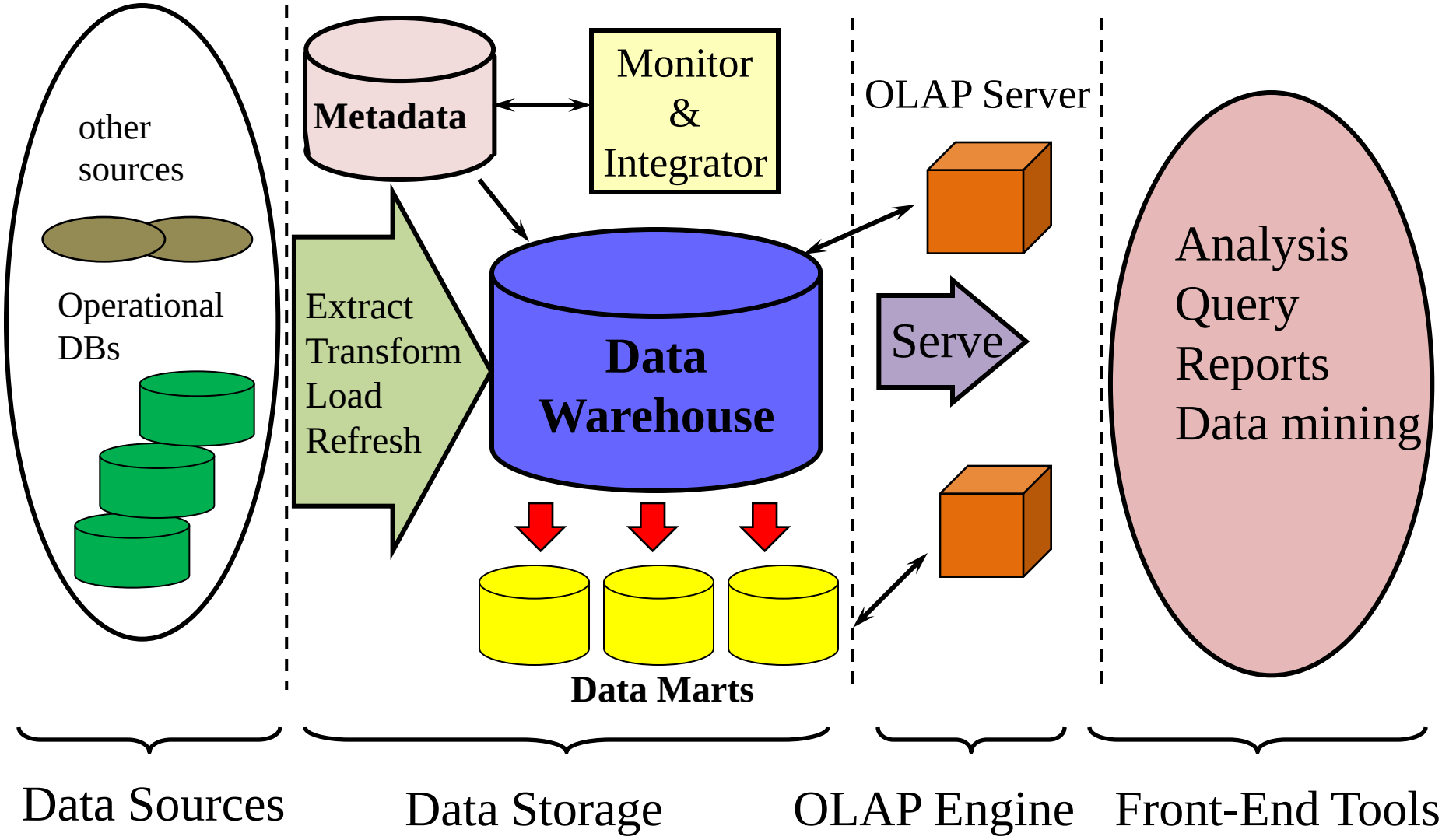
Four views regarding the design of a data warehouse

- **Top-down view**
  - allows selection of the relevant information necessary for the data warehouse
- **Data source view**
  - exposes the information being captured, stored, and managed by operational systems
- **Data warehouse view**
  - consists of fact tables and dimension tables
- **Business query view**
  - sees the perspectives of data in the warehouse from the view of end-user

Data Warehouse Design Process

- Top-down, bottom-up approaches or a combination of both
  - **Top-down:** Starts with overall design and planning (mature)
  - **Bottom-up:** Starts with experiments and prototypes (rapid)
- From software engineering point of view
  - **Waterfall:** structured and systematic analysis at each step before proceeding to the next
  - **Spiral:** rapid generation of increasingly functional systems, short turn around time, quick turn around
- Typical data warehouse design process
  - Choose a business process to model, e.g., orders, invoices, etc.
  - Choose the *grain* (*atomic level of data*) of the business process
  - Choose the dimensions that will apply to each fact table record
  - Choose the measure that will populate each fact table record

Multi-Tiered Architecture



Design of a Data Warehouse: Three Data Warehouse Models

- **Enterprise warehouse**
  - collects all of the information about subjects spanning the entire organization
  - top down approach
  - the W. Inmon methodology
- **Data Mart**
  - a subset of corporate-wide data that is of value to a specific groups of users. Its scope is confined to specific, selected groups, such as marketing data mart
    - Independent vs. dependent (directly from warehouse) data mart
    - bottom up approach
    - the R. Kimball methodology
- **Virtual warehouse**
  - A set of views over operational databases
  - Only some of the possible summary views may be materialized

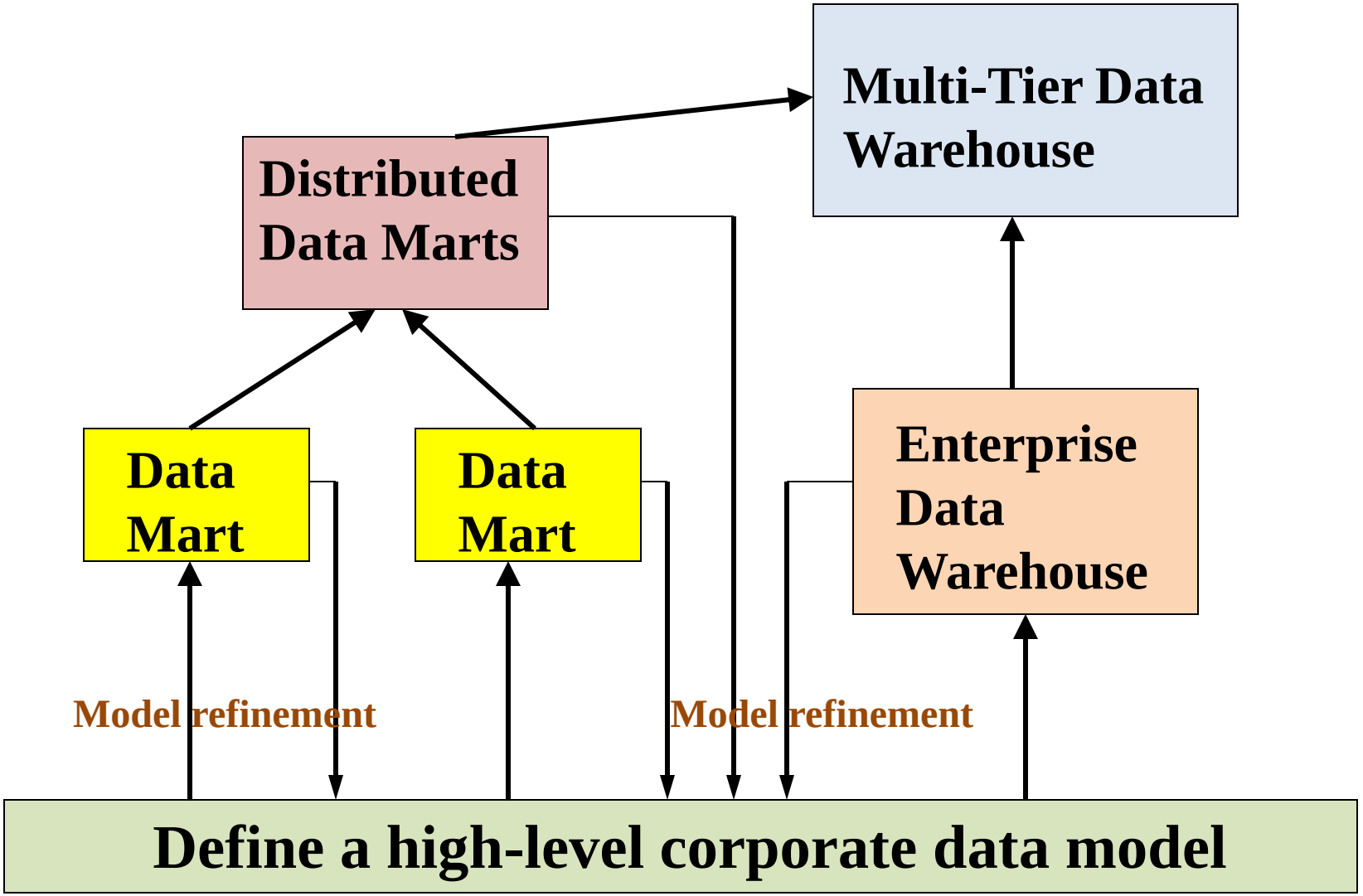
The Data Mart Strategy

- The most common approach
- Begins with a single mart and architected marts are added over time for more subject areas
- Relatively inexpensive and easy to implement
- Can be used as a proof of concept for data warehousing
- Can postpone difficult decisions and activities
- Requires an overall integration plan
- The key is to have an overall plan, processes, and technologies for integrating the different marts.
- The marts may be logically rather than physically separate.

Enterprise Warehouse Strategy

- A comprehensive warehouse is built initially
- An initial dependent data mart is built using a subset of the data in the warehouse
- Additional data marts are built using subsets of the data in the warehouse
- Like all complex projects, it is expensive, time consuming, and prone to failure
- When successful, it results in an integrated, scalable warehouse
- Even with the enterprise-wide strategy, the warehouse is developed in phases and each phase should be designed to deliver business value.

Data Warehouse Development: A Recommended Approach



Extract, Transform and Load (ETL) Definition

Three separate functions combined into one development tool:

1. **Extract** - Reads data from a specified source and extracts a desired subset of data.

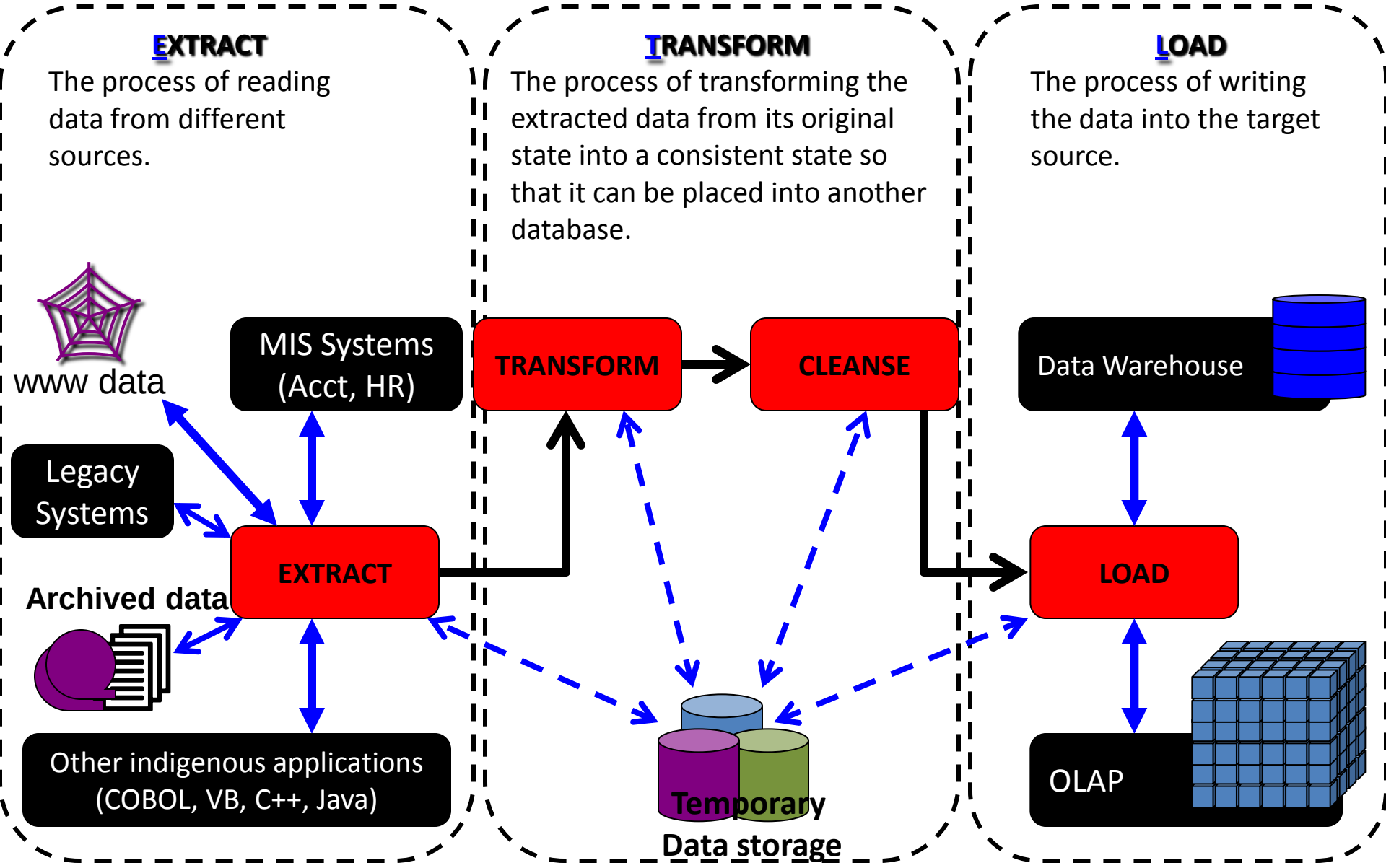
2. **Transform** - Uses rules or lookup tables, or creating combinations with other data, to convert source data to the desired state.

3. **Load** - Writes the resulting data to a target database

ETL Overview

- ETL, Short for *extract, transform, and load* are the database functions that are combined into one tool.
- ETL is used to migrate data from one database to another, to form data marts and data warehouses and also to convert databases from one format or type to another.
- 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

The ETL Cycle

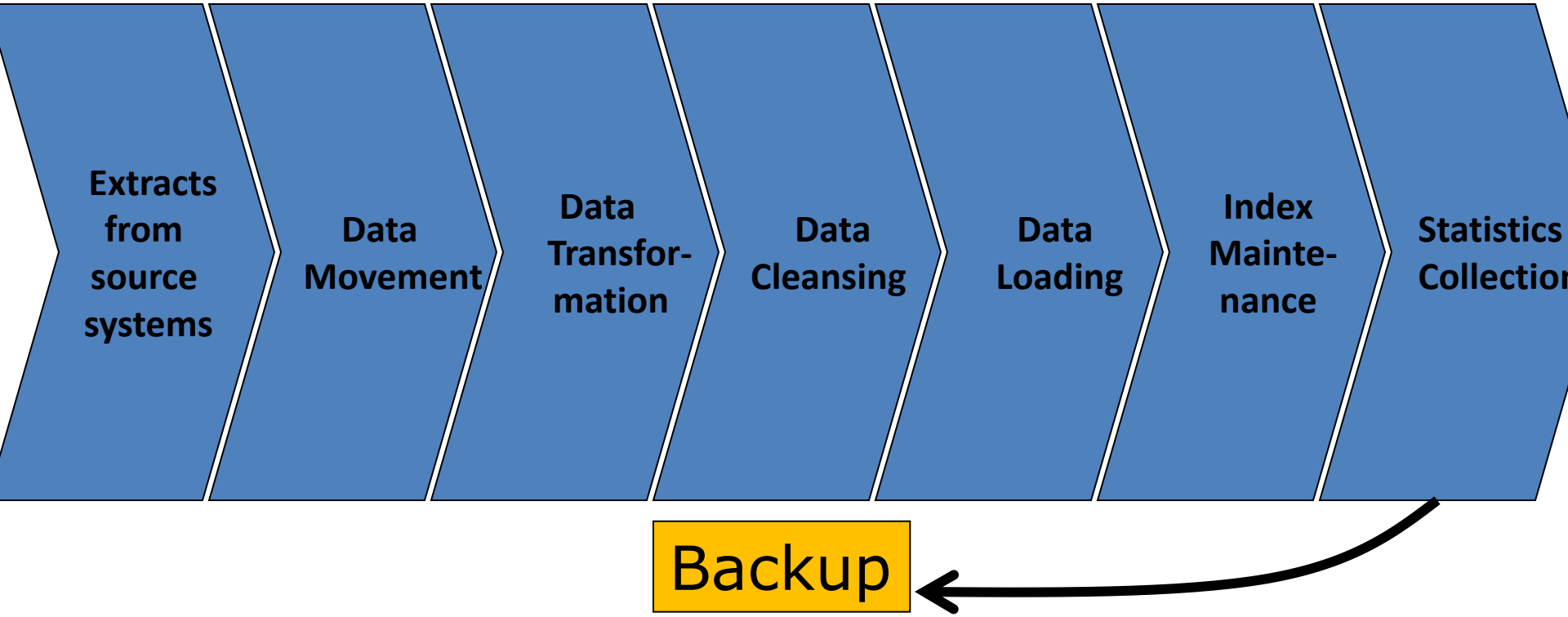


ETL Processing

ETL is independent yet interrelated steps.

It is important to look at the big picture.

Data acquisition time may include…



Back-up is a major task, it is a Data Warehouse not a cube

- 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 – i.e. monthly, daily, hourly
- Because ETL is an integral, ongoing, and recurring part of a data warehouse. It may be:
  - Automated
  - Well documented
  - Easily changeable
- When defining ETL for a data warehouse, it is important to think of ETL as a process, not a physical implementation

Extraction, Transformation, and Loading (ETL) Processes

Data Extraction

Data Cleansing

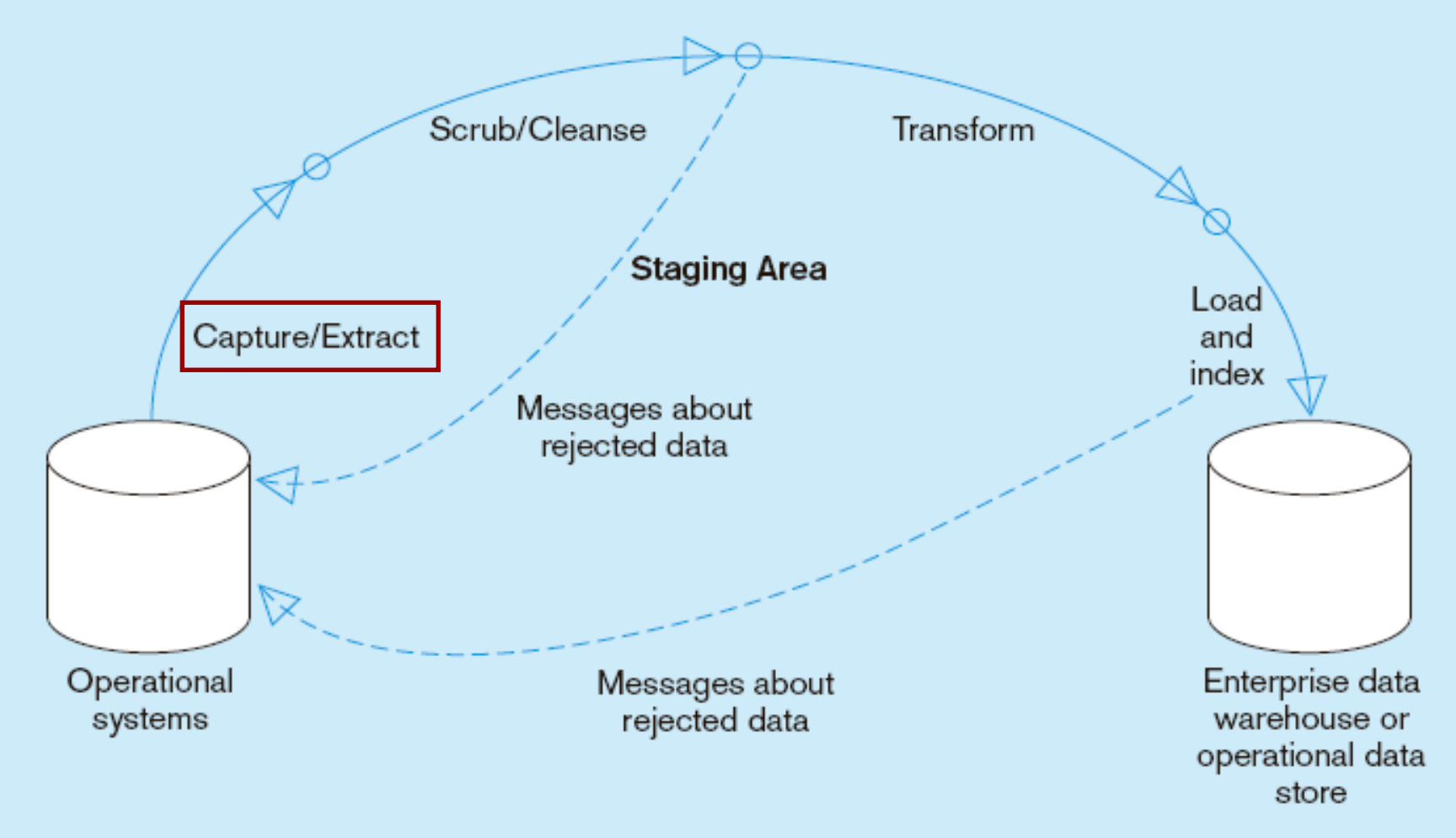
Data Transformation

Data Loading

Data Refreshing

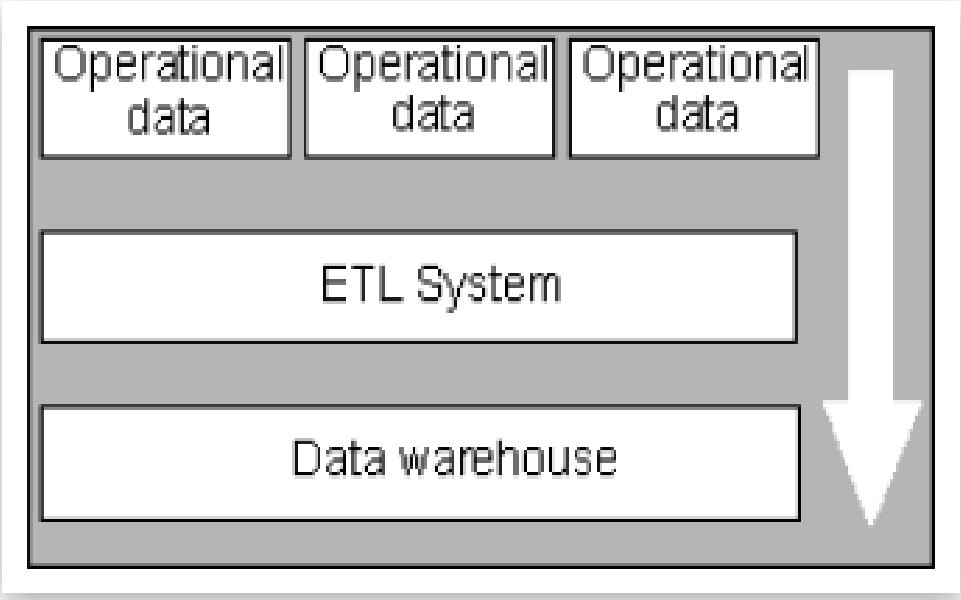
Data Extraction

Capture/Extract…obtaining a snapshot of a chosen subset of the source data for loading into the data warehouse



Static extract = capturing a snapshot of the source data at a point in time

Incremental extract = capturing changes that have occurred since the last static extract



- Data is extracted from heterogeneous data sources
- Each data source has its distinct set of characteristics that need to be managed and integrated into the ETL system in order to effectively extract data.

- ETL process needs to effectively integrate systems that have different:
  - DBMS
  - Operating Systems
  - Hardware
  - Communication protocols

- Need to have a logical data map before the physical data can be transformed

- The logical data map describes the relationship between the extreme starting points and the extreme ending points of your ETL system usually presented in a table or spreadsheet

Target			Source			Transformation
Table Name	Column Name	Data Type	Table Name	Column Name	Data Type	

- The content of the logical data mapping document has been proven to be the critical element required to efficiently plan ETL processes.
- The table type gives us our queue for the ordinal position of our data load processes—first dimensions, then facts.
- This table must depict, without question, the course of action involved in the transformation process
- The transformation can contain anything from the absolute solution to nothing at all. Most often, the transformation can be expressed in SQL. The SQL may or may not be the complete statement

Some ETL Tools

Tool	Vendor
Oracle Warehouse Builder (OWB)	Oracle
Data Integrator (BODI)	Business Objects
IBM Information Server (Ascential)	IBM
SAS Data Integration Studio	SAS Institute
PowerCenter	Informatica
Oracle Data Integrator (Sunopsis)	Oracle
Data Migrator	Information Builders
Integration Services	Microsoft
Talend Open Studio	Talend
DataFlow	Group 1 Software (Sagent)
Data Integrator	Pervasive
Transformation Server	DataMirror
Transformation Manager	ETL Solutions Ltd.
Data Manager	Cognos
DT/Studio	Embarcadero Technologies
ETL4ALL	IKAN
DB2 Warehouse Edition	IBM
Jitterbit	Jitterbit
Pentaho Data Integration	Pentaho

Data Cleansing

Data Cleansing

- Data Warehouse is NOT just about arranging data, but should be clean for overall health of organization. **"We drink clean water"!**
- Sometime called as **Data Scrubbing** or **Cleaning**.
- ETL software contains rudimentary data cleansing capabilities
- Specialized data cleansing software is often used. Leading data cleansing vendors include Vality (Integrity), Harte-Hanks (Trillium), and Firstlogic (i.d.Centric)

Why Cleansing?

- Data warehouse contains data that is analyzed for business decisions
- Source systems contain “dirty data” that must be cleansed.
- More data and multiple sources could mean more errors in the data and harder to trace such errors
- Results in incorrect analysis
- Enormous problem, as most data is dirty. **(GIGO)**

Reasons for “Dirty” Data

- Dummy Values
- Absence of Data
- Multipurpose Fields
- Cryptic Data
- Contradicting Data
- Inappropriate Use of Address Lines
- Violation of Business Rules
- Reused Primary Keys,
- Non-Unique Identifiers
- Data Integration Problems

Examples:

- Dummy Data Problem:

  A clerk enters 999-99-9999 as a SSN rather than asking the customer for theirs.

- Reused Primary Keys:

  A branch bank is closed. Several years later, a new branch is opened, and the old identifier is used again.

Inconsistent Data Representations

Same data, different representation

Date value representations

Examples:

970314 1997-03-14

03/14/1997 14-MAR-1997

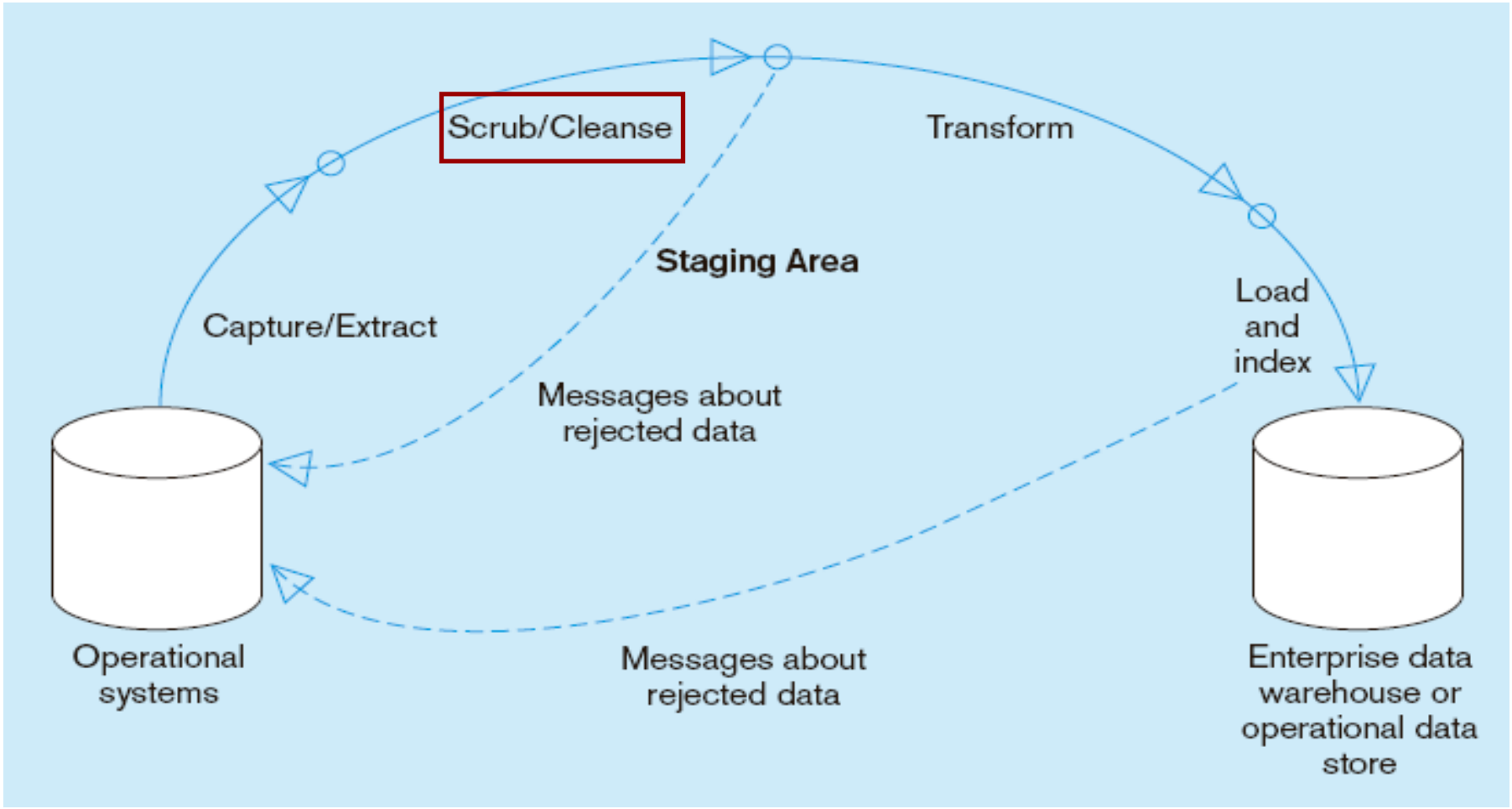
March 14 1997 2450521.5 (Julian date format)

Gender value representations

Examples:

- Male/Female - M/F
- 0/1

Scrub/Cleanse…uses pattern recognition and AI techniques to upgrade data quality



Fixing errors: misspellings, erroneous dates, incorrect field usage, mismatched addresses, missing data, duplicate data, inconsistencies

Also: decoding, reformatting, time stamping, conversion, key generation, merging, error detection/logging, locating missing data

Two Classes of Anomalies

- **Coverage Problems**
  - **Missing values**
  - **Missing Tuples or records**

- **Key-based classification problems**
  - **Primary key problems**
  - **Non-Primary key problems**

1. Coverage Problems

- **Missing Attribute**

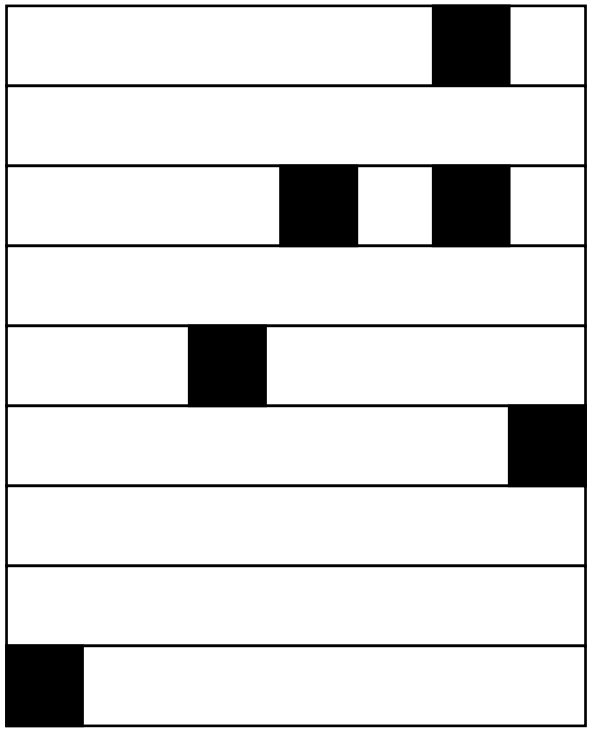
  Result of omissions while collecting the data.

  A constraint violation if we have null values for attributes where NOT NULL constraint exists.

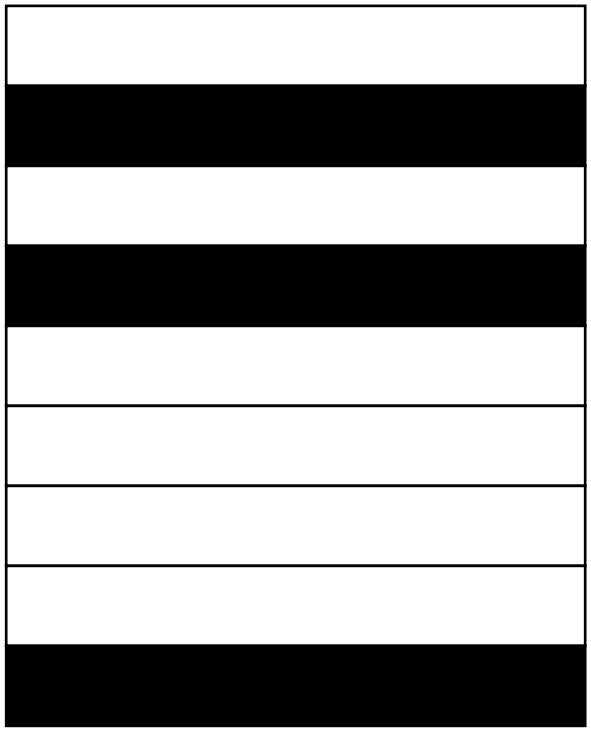
  Case more complicated where no such constraint exists.

  Have to decide whether the value exists in the real world and has to be deduced here or not.

Coverage



Missing values



Missing records

Why Missing Rows/Value?

- Equipment malfunction (bar code reader, keyboard etc.)
- Inconsistent with other recorded data and thus deleted.
- Data not entered due to misunderstanding/illegibility.
- Data not considered important at the time of entry (e.g. Y2K).

Handling missing data

- Dropping records.
- “Manually” filling missing values.
- Using a global constant as filler.
- Using the attribute mean (or median) as filler.
- Using the most probable value as filler.

2. Key-based Classification Problems

Primary key problems

- Same PK but different data.
- Same entity with different keys.
- PK in one system but not in other.
- Same PK but in different formats.

Non primary key problems

- Different encoding in different sources.
- Multiple ways to represent the same information.
- Sources might contain invalid data.
- Two fields with different data but same name.
- Required fields left blank.
- Data erroneous or incomplete.
- Data contains null values.

Data Quality paradigm

- Correct
- Unambiguous
- Consistent
- Complete

Data quality checks are run at 2 places - after extraction and after cleaning and confirming additional check are run at this point

Steps in Data Cleansing

- Parsing
- Correcting
- Standardizing
- Matching
- Consolidating

Parsing

- The record is broken down into atomic data elements.
- Parsing locates and identifies individual data elements in the source files and then isolates these data elements in the target files.
- Examples include parsing the first, middle, and last name; street number and street name; and city and state.

Correcting

- External data, such as census data, is often used in this process.
- Corrects parsed individual data components using sophisticated data algorithms and secondary data sources.
- Example include replacing a vanity address and adding a zip code.

Standardizing

- Companies decide on the standards that they want to use.
- Standardizing applies conversion routines to transform data into its preferred (and consistent) format using both standard and custom business rules.
- Examples include adding a pre name, replacing a nickname, and using a preferred street name.

Matching

- Commercial data cleansing software often uses AI techniques to match records.
- Searching and matching records within and across the parsed, corrected and standardized data based on predefined business rules to eliminate duplications.
- Examples include identifying similar names and addresses.

Consolidating

- All of the data are now combined in a standard format.
- Analyzing and identifying relationships between matched records and consolidating/merging them into ONE representation.

Data Staging

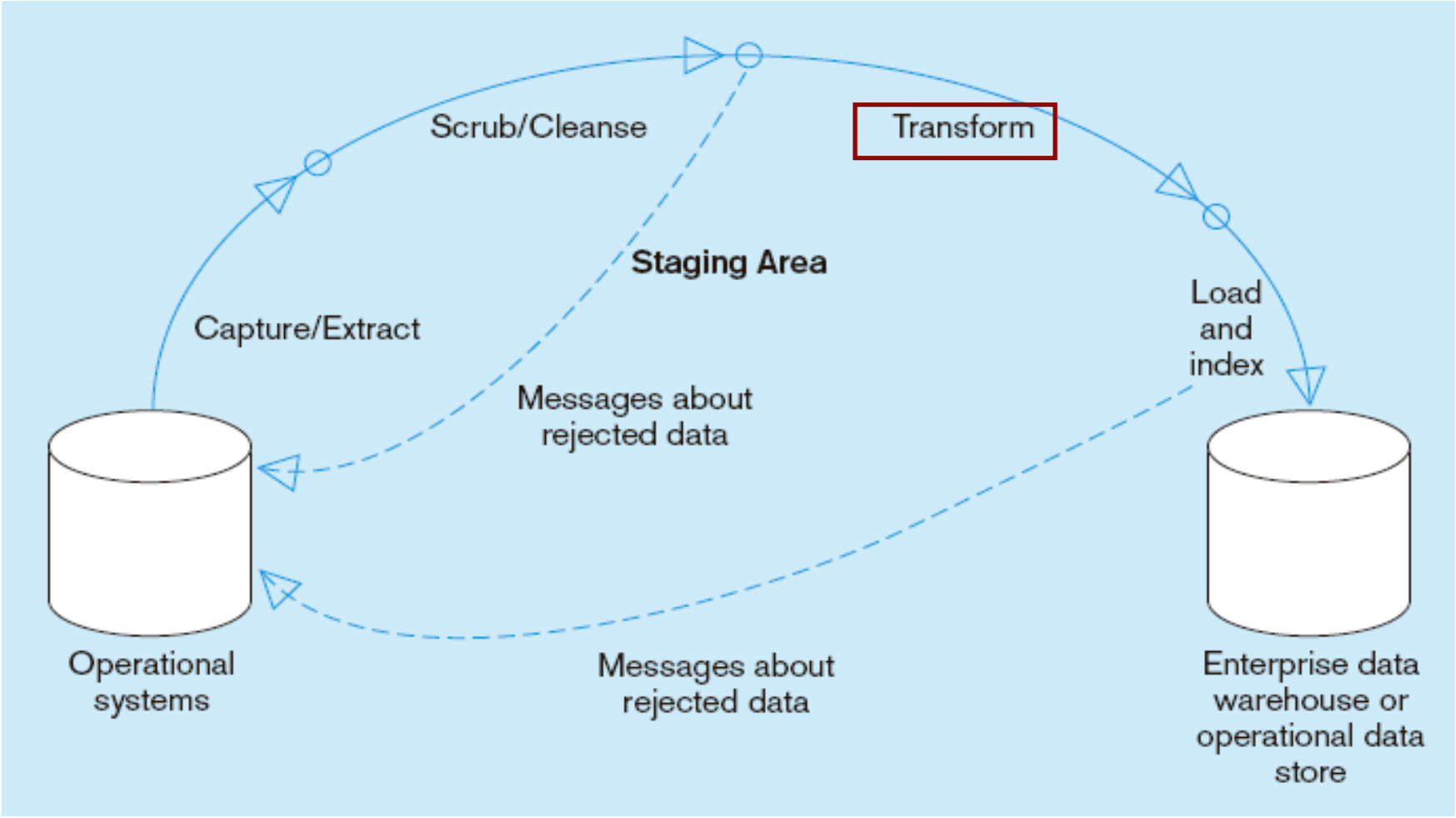
- Data staging is used in cleansing, transforming, and integrating the data.
- Often used as an interim step between data extraction and later steps
- Accumulates data from asynchronous sources using native interfaces, flat files, FTP sessions, or other processes
- At a predefined cutoff time, data in the staging file is transformed and loaded to the warehouse
- There is usually no end user access to the staging file
- An operational data store may be used for data staging

Data Transformation

Data Transformation

- It is the main step where the ETL adds value.
- Actually changes data and provides guidance whether data can be used for its intended purposes.
- Performed in staging area.

Transform = convert data from format of operational system to format of data warehouse



Record-level:

Selection–data partitioning

Joining–data combining

Aggregation–data summarization

Field-level:

single-field–from one field to one field

multi-field–from many fields to one, or one field to many

Basic Tasks

1. Selection
2. Splitting/Joining
3. Conversion
4. Summarization
5. Enrichment

Data Transformation : Conversion

- Convert common data elements into a consistent form i.e. name and address.

Field format		Field data
First-Family-title	→	Bijay Mishra, Lecturer
Family-title-comma-first	→	Mishra Lecturer, Bijay
Family-comma-first-title	→	Mishra, Bijay Lecturer

- Translation of dissimilar codes into a standard code.

Natl. ID	NID
National ID	NID

F/NO-2
F-2
FL.NO.2
FL.2
FL/NO.2
FL-2 → FLAT No. 2
FLAT-2
FLAT#
FLAT,2
FLAT-NO-2
FL-NO.2

- Data representation change
  - EBCIDIC to ASCII
- Operating System Change
  - Mainframe (MVS) to UNIX
  - UNIX to NT or XP
- Data type change
  - Program (Excel to Access), database format (FoxPro to Access).
  - Character, numeric and date type.
  - Fixed and variable length.

Data Transformation : Summarization

Values are summarized to obtain total figures which are subsequently calculated and stored at multiple levels as business fact in multidimensional fact tables.

Data Transformation : Enrichment

- Data elements are mapped from source tables and files to destination fact and dimension tables.

Input Data

BIJAY MISHRA LECTURER
WHITEHOUSE INTERNATIONAL COLLEGE,
KHUMALTAR, LALITPUR
BAGMATI, 9841695609

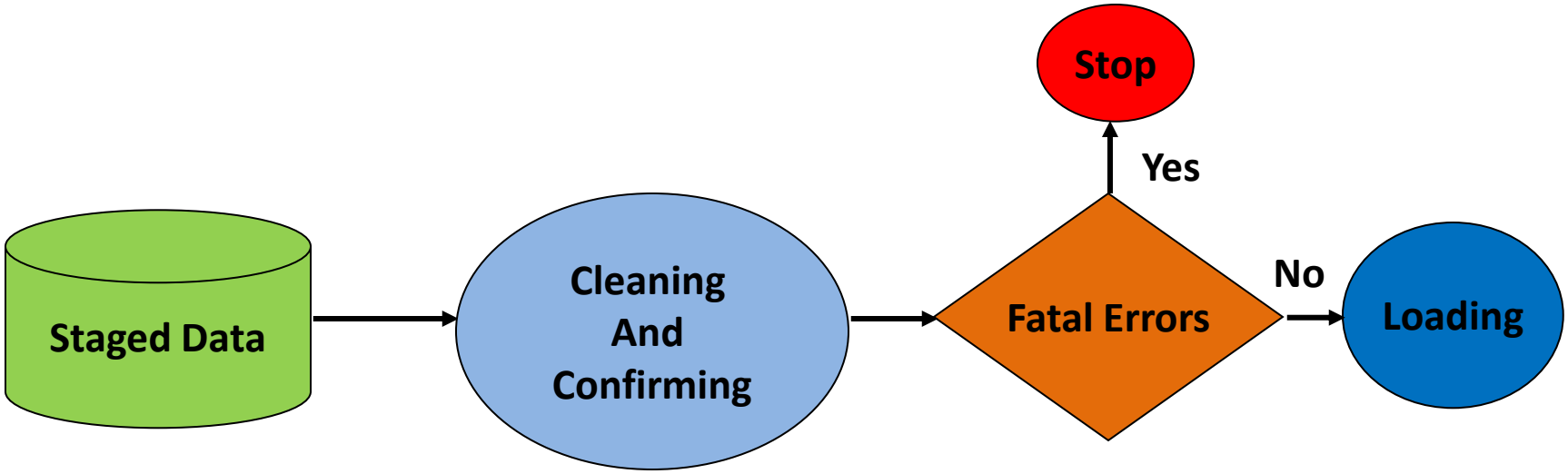
Parsed Data

First Name:	BIJAY
Family Name:	MISHRA
Title:	LECTURER
College:	WHITEHOUSE INTERNATIONAL COLLEGE,
College Location:	KHUMALTAR, LALITPUR
Zone:	RAWALPINDI
Mobile:	9841695609
Code:	46200

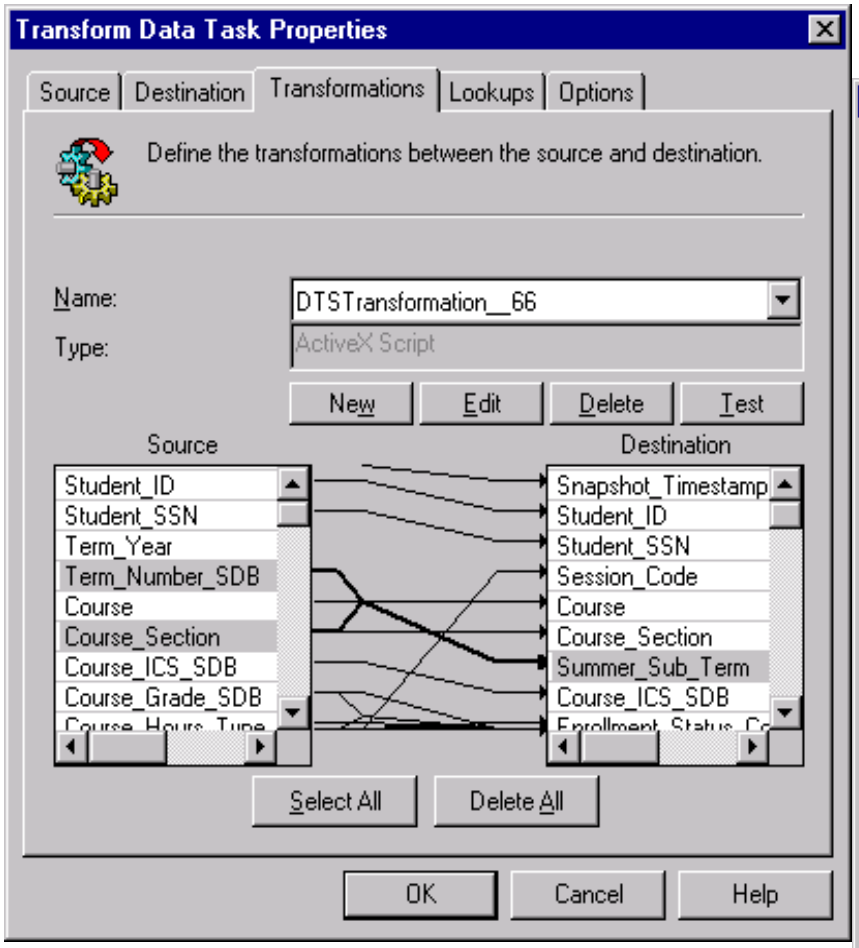
- Default values are used in the absence of source data.
- Fields are added for unique keys and time elements.

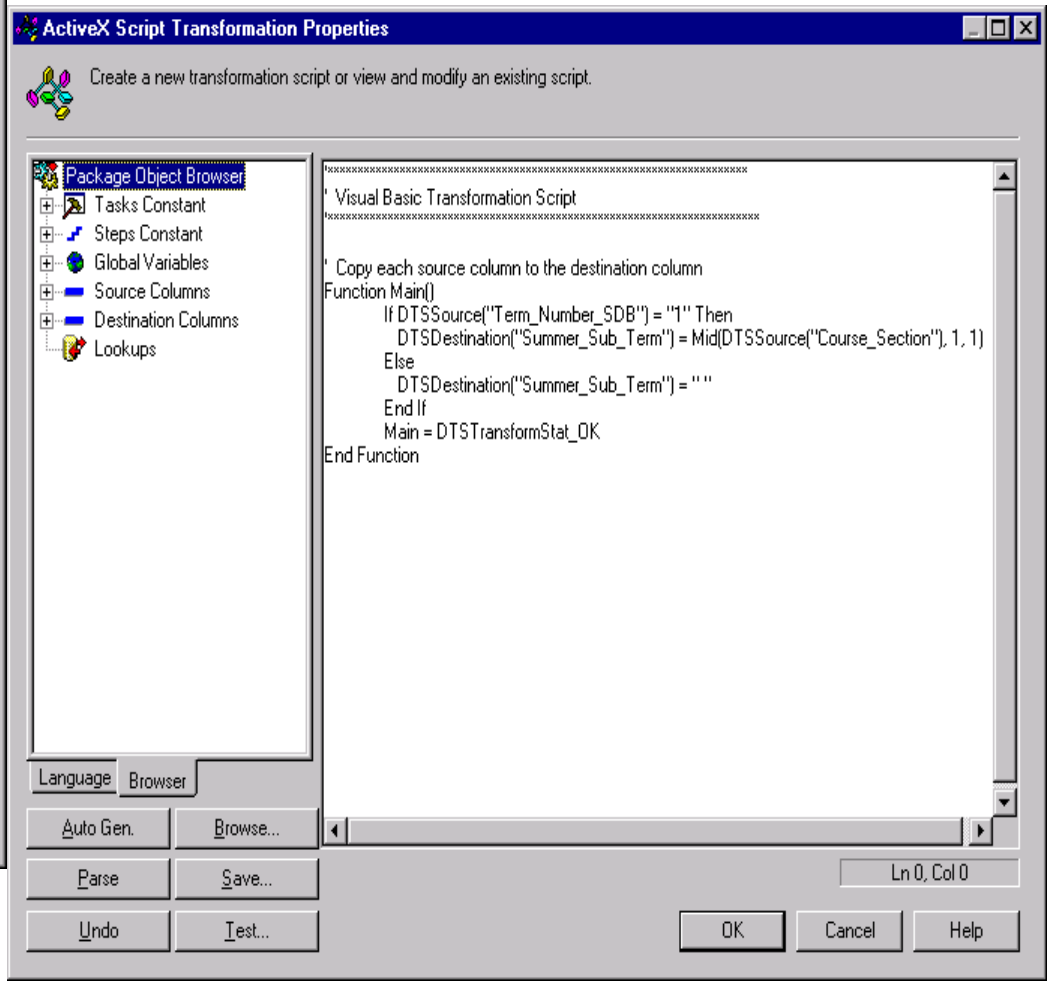
Transformation - Confirming

- Structure Enforcement
  - Tables have proper primary and foreign keys
  - Obey referential integrity
- Data and Rule value enforcement
  - Simple business rules
  - Logical data checks


Stop
Yes
Staged Data
Cleaning
And
Confirming
Fatal Errors
No
Loading

Data Warehouse
Data Transformation Services



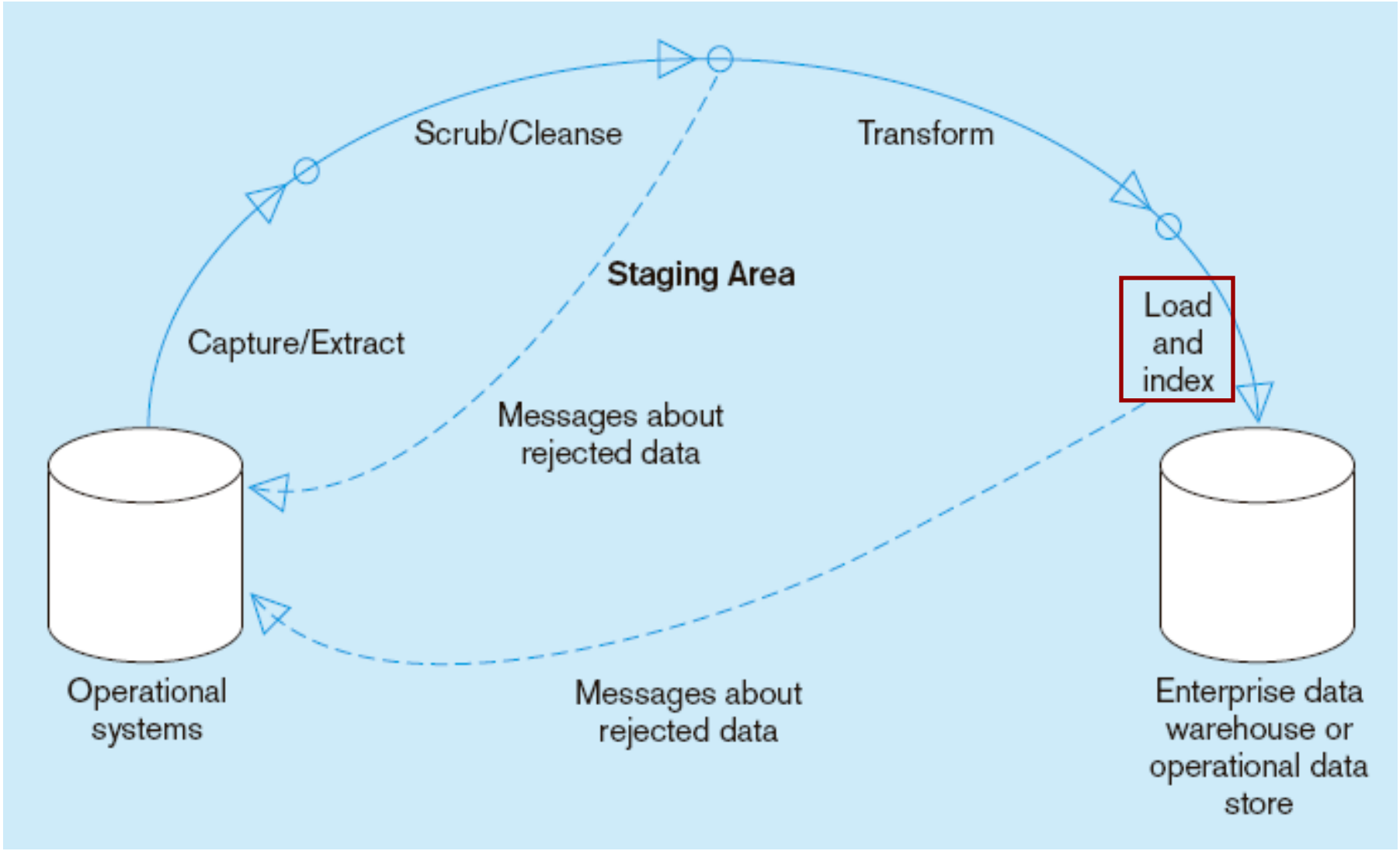


Data Loading

Data Loading

- Most loads involve only change data rather than a bulk reloading of all of the data in the warehouse.
- Data are physically moved to the data warehouse
- The loading takes place within a “load window”
- The trend is to near real time updates of the data warehouse as the warehouse is increasingly used for operational applications

Load/Index= place transformed data into the warehouse and create indexes



Refresh mode: bulk rewriting of target data at periodic intervals

Update mode: only changes in source data are written to data warehouse

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

Initial Load

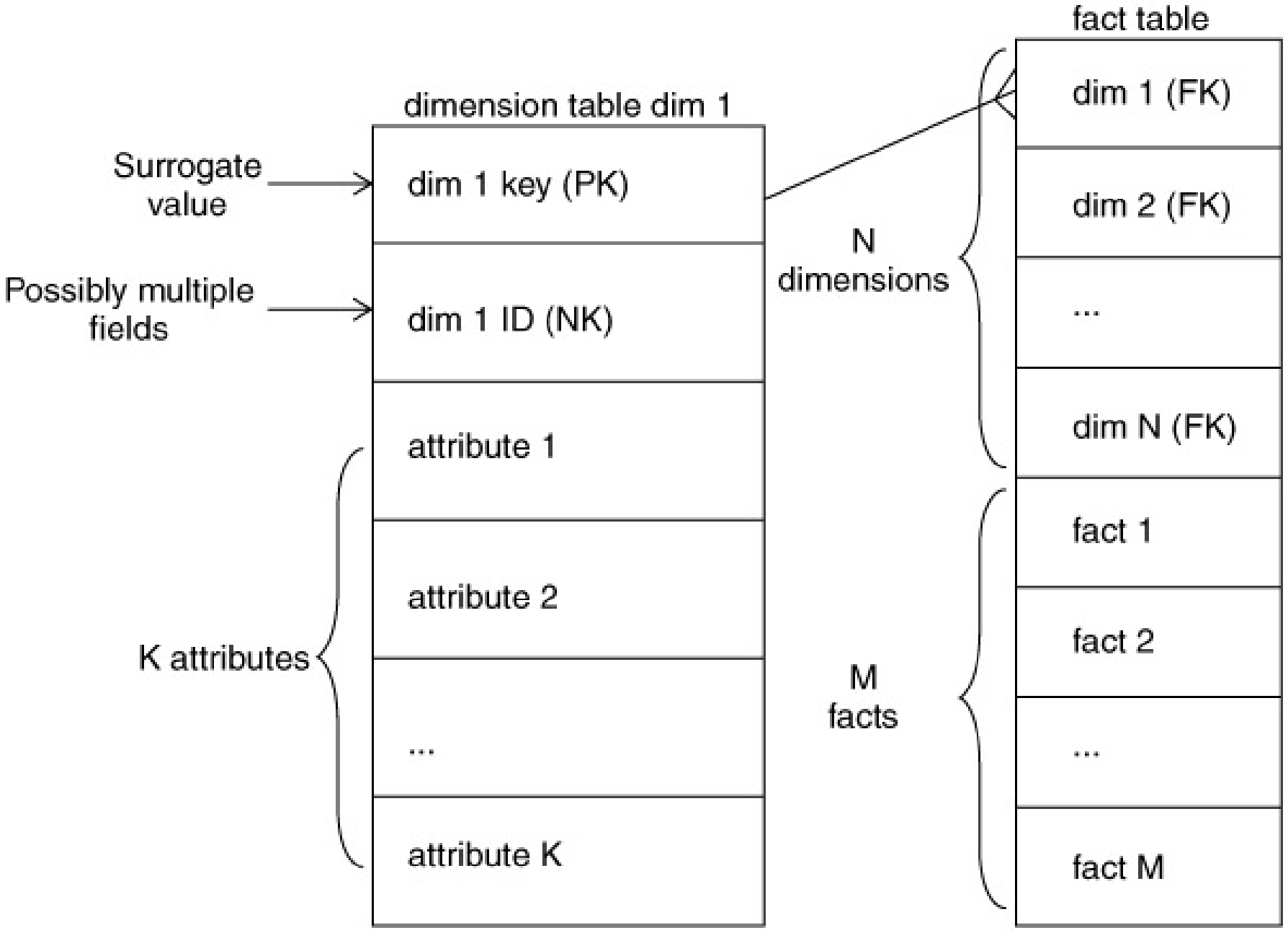
- Consists of populating tables in warehouse schema and verifying data readiness
- Examples:
  - DTS – data transformation services
  - Backup utility – batch copy
  - SQL*Loader
  - Native Database Languages (T-SQL, PL/SQL, etc.)

Continuous Loads

- Must be scheduled and processed in a specific order to maintain integrity, completeness, and a satisfactory level of trust
- Should be the most carefully planned step in data warehousing or can lead to:
  - Error duplication
  - Exaggeration of inconsistencies in data
- Must be during a fixed batch window (usually overnight)
- Must maximize system resources to load data efficiently in allotted time
  - Ex. *Red Brick Loader* can validate, load, and index up to 12GB of data per hour on an SMP system

Loading Dimensions

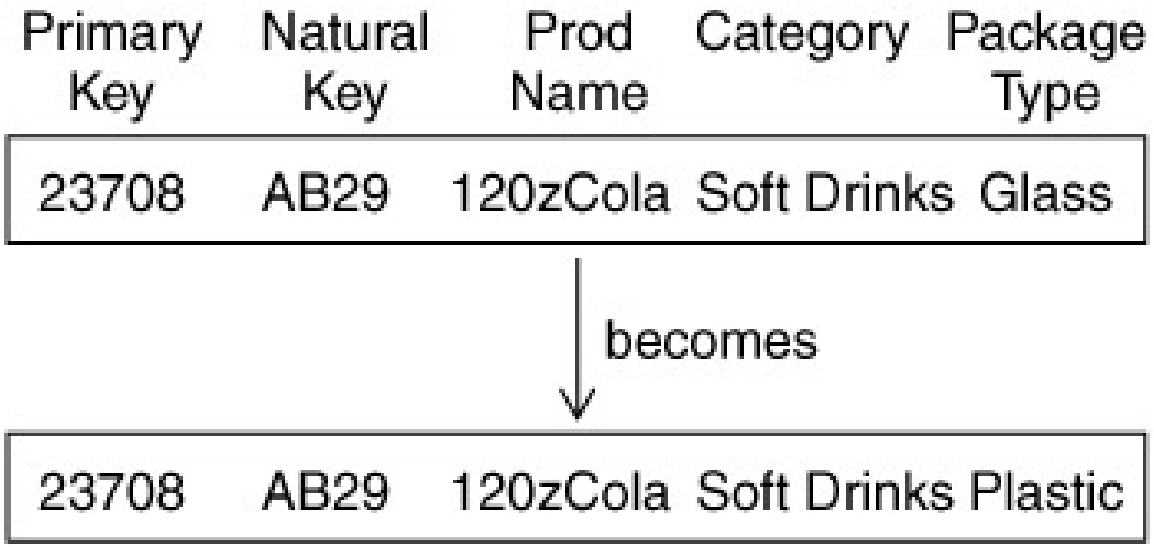
- Physically built to have the minimal sets of components
- The primary key is a single field containing meaningless unique integer – Surrogate Keys
- The DW owns these keys and never allows any other entity to assign them
- De-normalized flat tables – all attributes in a dimension must take on a single value in the presence of a dimension primary key.
- Should possess one or more other fields that compose the natural key of the dimension


fact table
dim 1 (FK)
dim 2 (FK)
...
dim N (FK)
fact 1
fact 2
...
fact M
N
dimensions
M
facts
dimension table dim 1
dim 1 key (PK)
dim 1 ID (NK)
attribute 1
attribute 2
...
attribute K
Surrogate
value
Possibly multiple
fields
K attributes

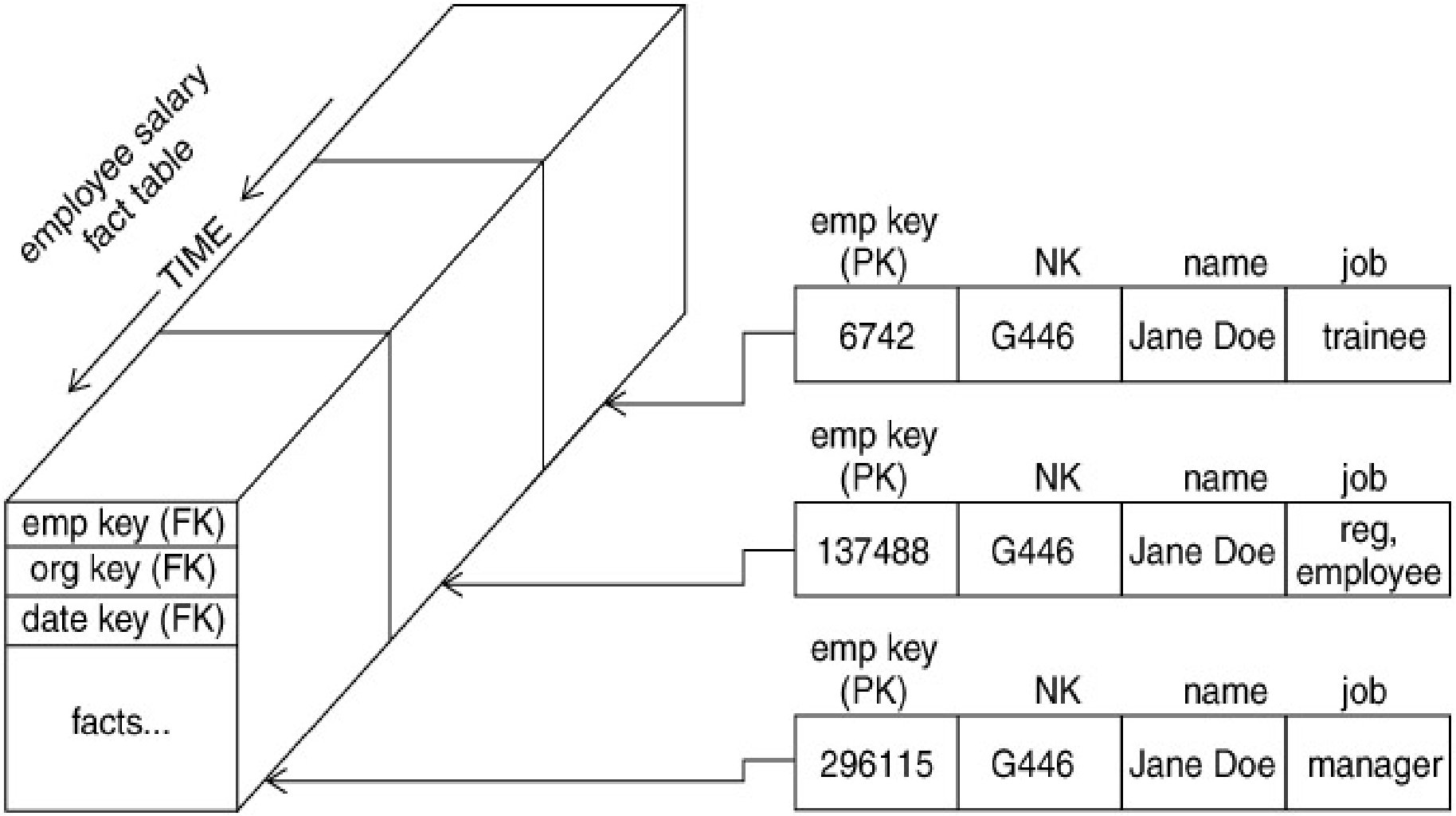
- The data loading module consists of all the steps required to administer slowly changing dimensions (SCD) and write the dimension to disk as a physical table in the proper dimensional format with correct primary keys, correct natural keys, and final descriptive attributes.
- Creating and assigning the surrogate keys occur in this module.
- The table is definitely staged, since it is the object to be loaded into the presentation system of the data warehouse.

- When DW receives notification that an existing row in dimension has changed it gives out 3 types of responses:
  - Type 1
  - Type 2
  - Type 3

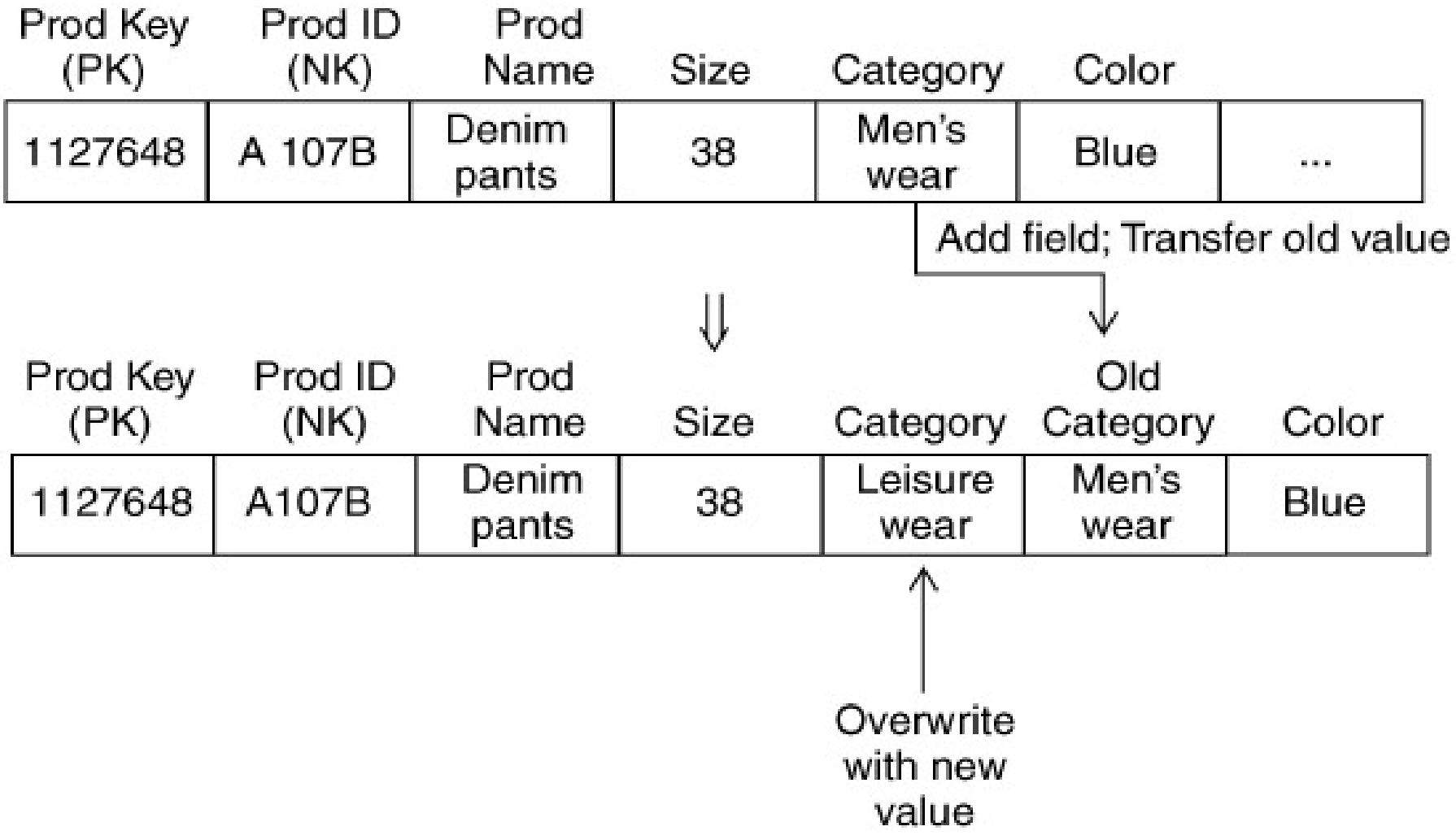
Type 1 Dimension



Type 2 Dimension



Type 3 Dimensions

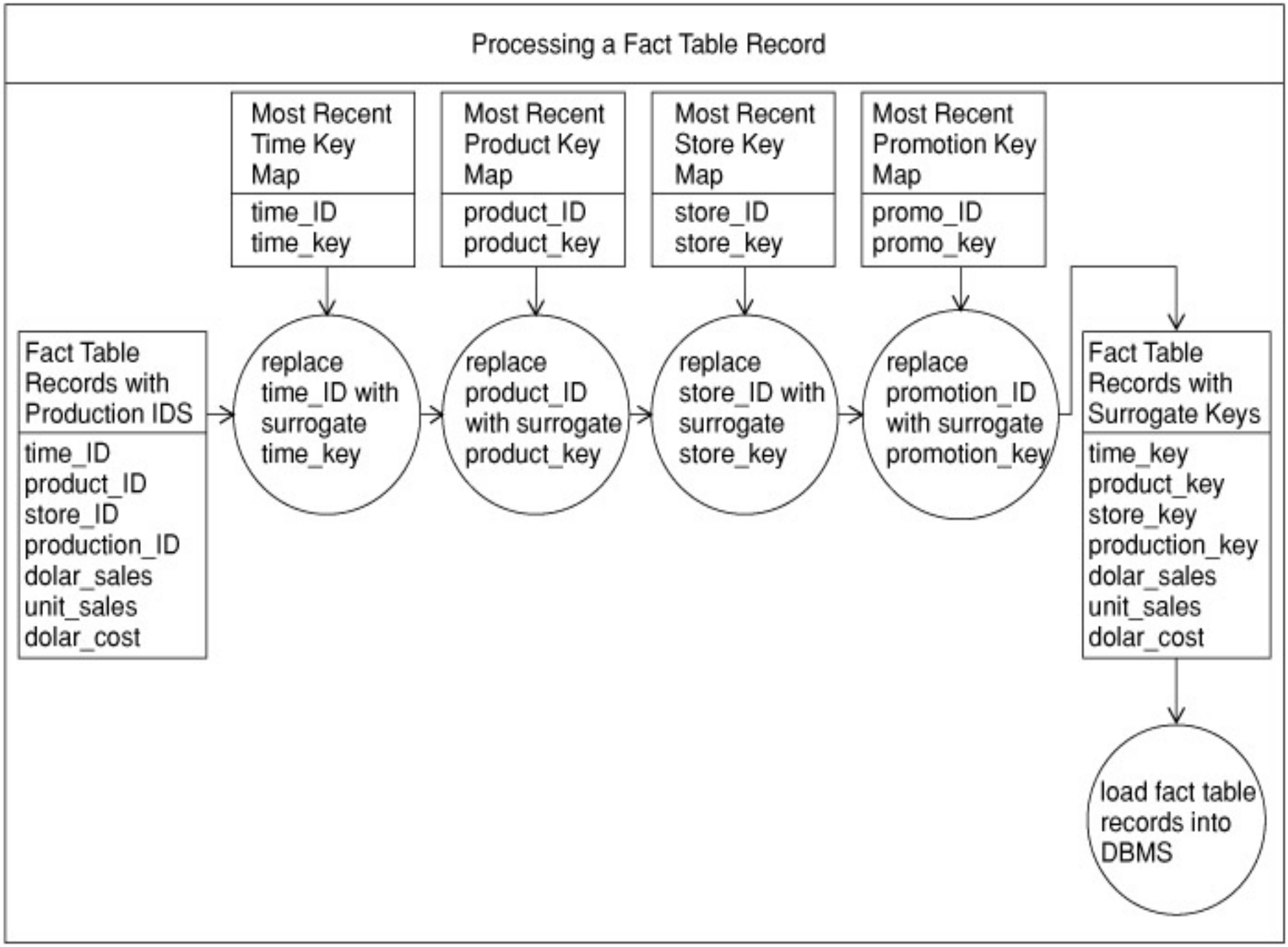


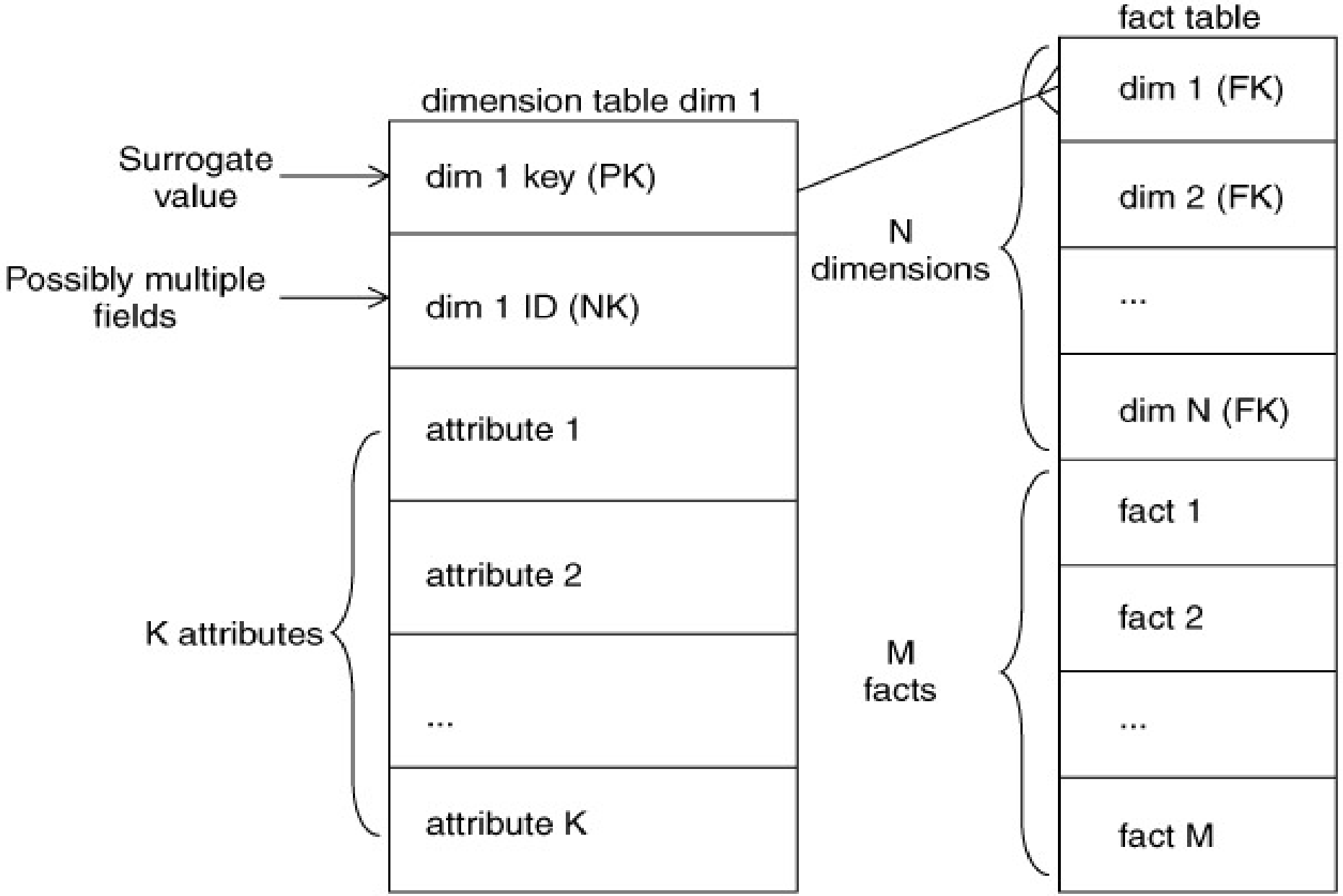
Loading Facts

- Fact tables hold the measurements of an enterprise.
- The relationship between fact tables and measurements is extremely simple.
- If a measurement exists, it can be modeled as a fact table row.
- If a fact table row exists, it is a measurement

Key Building Process - Facts

- When building a fact table, the final ETL step is converting the natural keys in the new input records into the correct, contemporary surrogate keys
- ETL maintains a special surrogate key lookup table for each dimension. This table is updated whenever a new dimension entity is created and whenever a Type 2 change occurs on an existing dimension entity
- All of the required lookup tables should be pinned in memory so that they can be randomly accessed as each incoming fact record presents its natural keys. This is one of the reasons for making the lookup tables separate from the original data warehouse dimension tables.


Processing a Fact Table Record
Most Recent Time Key Map
time_ID
time_key
Most Recent Product Key Map
product_ID
product_key
Most Recent Store Key Map
store_ID
store_key
Most Recent Promotion Key Map
promo_ID
promo_key
Fact Table Records with Production IDS
time_ID
product_ID
store_ID
production_ID
dolar_sales
unit_sales
dolar_cost
replace time_ID with surrogate time_key
replace product_ID with surrogate product_key
replace store_ID with surrogate store_key
replace promotion_ID with surrogate promotion_key
Fact Table Records with Surrogate Keys
time_key
product_key
store_key
production_key
dolar_sales
unit_sales
dolar_cost
load fact table records into DBMS


fact table
dimension table dim 1
Surrogate value
dim 1 key (PK)
dim 1 (FK)
dim 2 (FK)
N dimensions
Possibly multiple fields
dim 1 ID (NK)
...
dim N (FK)
attribute 1
fact 1
attribute 2
fact 2
K attributes
M facts
...
...
attribute K
fact M

Loading Fact Tables

- Managing Indexes
  - Performance Killers at load time
  - Drop all indexes in pre-load time
  - Segregate Updates from inserts
  - Load updates
  - Rebuild indexes

❖Managing Partitions

- Partitions allow a table (and its indexes) to be physically divided into *minitables* for administrative purposes and to improve query performance
- The most common partitioning strategy on fact tables is to partition the table by the date key. Because the date dimension is preloaded and static, you know exactly what the surrogate keys are
- Need to partition the fact table on the key that joins to the date dimension for the optimizer to recognize the constraint.
- The ETL team must be advised of any table partitions that need to be maintained.

Data Refreshing

Data Refresh

- Propogate updates from sources to the warehouse
- **Issues:**
  - when to refresh
  - how to refresh -- refresh techniques
- Set by administrator depending on user needs and traffic

When to Refresh?

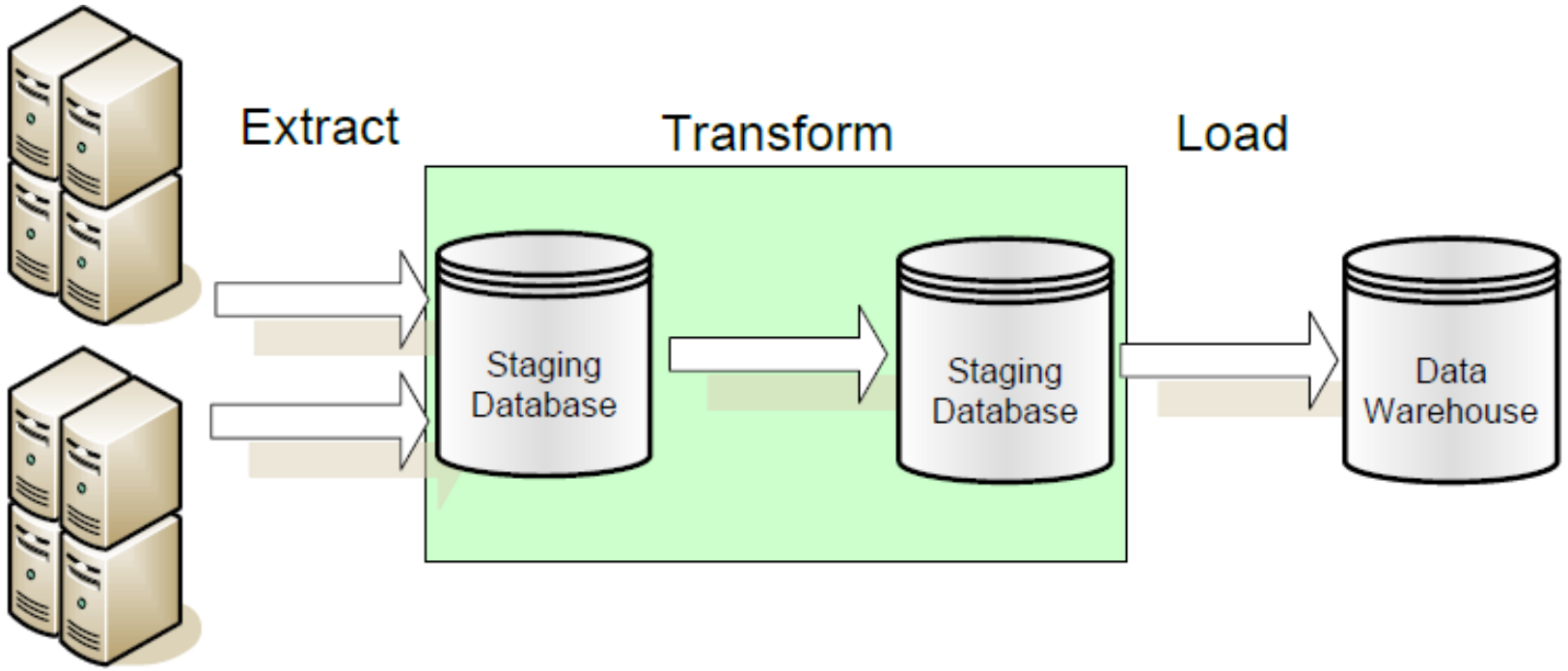
- periodically (e.g., every night, every week) or after significant events
- on every update: not warranted unless warehouse data require current data (up to the minute stock quotes)
- refresh policy set by administrator based on user needs and traffic
- possibly different policies for different sources

Refresh Techniques

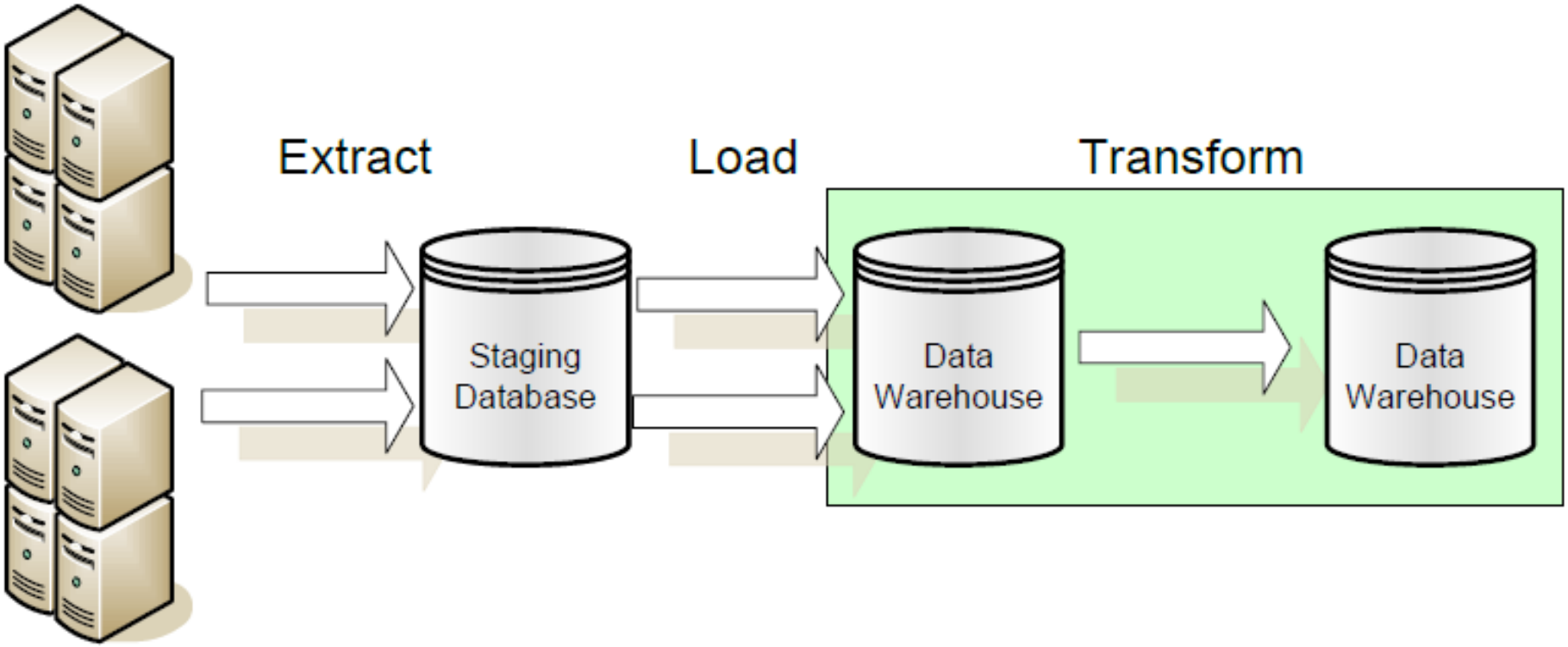
- Full Extract from base tables
  - read entire source table: too expensive
  - maybe the only choice for legacy systems

ETL vs. ELT

ETL: Extract, Transform, Load in which data transformation takes place on a separate transformation server.



ELT: Extract, Load, Transform in which data transformation takes place on the data warehouse server.

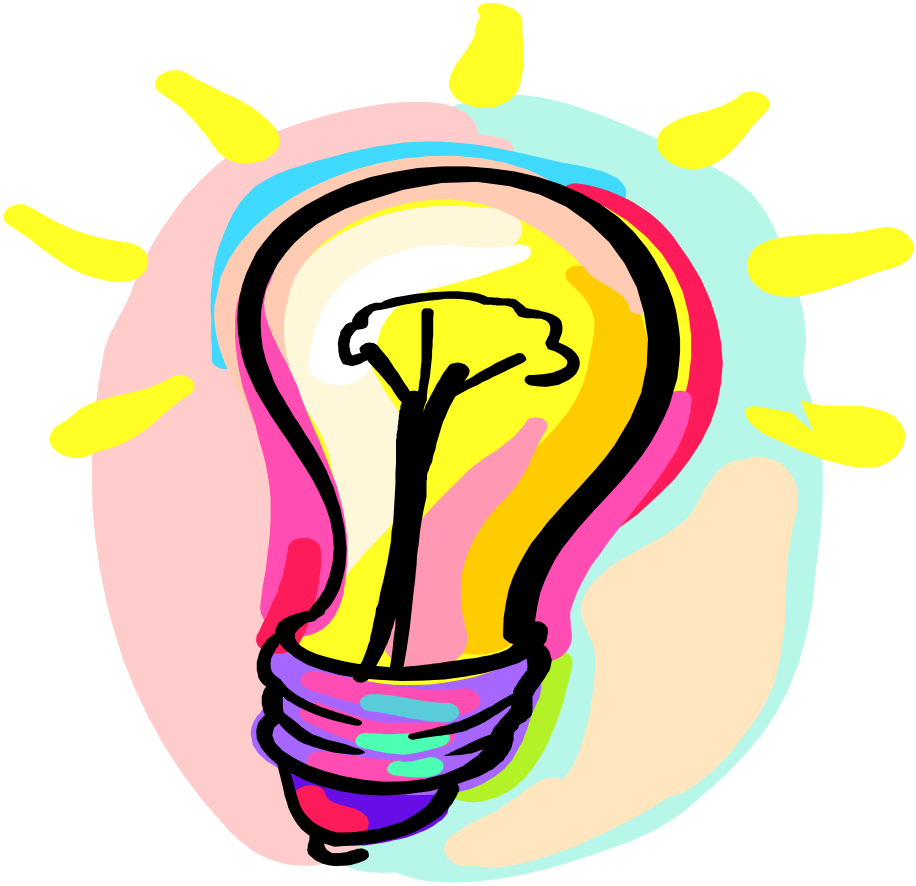


Data warehouse support in SQL Server 2008/Oracle 11g

- **Oracle** supports the ETL process with their "Oracle Warehouse Builder" product. Many new features in the Oracle9i database will also make ETL processing easier.
- Data Warehouse Builder (or Oracle Data Mart builder), Oracle Designer, Oracle Express, Express Objects, etc. tools can be used to design and build a warehouse.

- **SQL Server 2008** introduced what we call the Management Data Warehouse.
- The Management Data Warehouse is a relational database that contains data that is collected from a server using the new SQL Server 2008 Data Collection mechanism.
- The Warehouse consists primarily of the following components:
  - An extensible data collector
  - Stored procedures which allow the DBA to create their own data collection set and own the resultant data collection items
  - Three Data Collections Sets which are delivered with SQL Server 2008 and which can be enabled at any time
  - Standard reports delivered with SQL Server 2008 Management Studio display data collected by the three predefined Data Collection Sets

Questions?



References

1. D. Pyle. Data Preparation for Data Mining. Morgan Kaufmann, 1999.
2. R. Wang, V. Storey, and C. Firth. A framework for analysis of data quality research. IEEE Trans. Knowledge and Data Engineering, 7:623-640, 1995.
3. J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2000.
4. C. Imhoff, N. Galemmo, and J. G. Geiger. Mastering Data Warehouse Design: Relational and Dimensional Techniques. John Wiley, 2003
5. W. H. Inmon. Building the Data Warehouse. John Wiley, 1996
6. R. Kimball and M. Ross. The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling. 2ed. John Wiley, 2002
7. http://blogs.msdn.com

End of Unit 4

Thank you !!!