# DATA WAREHOUSE

***MSBA 7024 / MACC 7020***

***Database Design and Management***

# Objectives

- Definition of terms
- Reasons for information gap between information needs and availability
- Reasons for need of data warehousing
- Describe three levels of data warehouse architectures
- List four steps of data reconciliation
- Describe two components of star schema
- Compare star schema with snowflake and galaxy schemas
- Discuss issues related to the successful implementation of data warehousing
- Explain the differences between a data warehouse and a data lake

# Definition

- **Data Warehouse**:
  - A subject-oriented, integrated, time-variant, non-volatile collection of data used in support of management decision-making processes
  - ***Subject-oriented:*** e.g. customers, patients, students, products
  - ***Integrated:*** Consistent naming conventions, formats, encoding structures; from multiple data sources
  - ***Time-variant:*** Can study trends and changes
  - ***Non-volatile:*** Read-only, periodically refreshed
- **Data Mart**:
  - A data warehouse that is limited in scope

# Separating Operational and Informational Systems

- **Operational system** – a system that is used to run a business in real time, based on current data; also called a system of record

- **Informational system (decision support system)** – a system designed to support decision making based on historical point-in-time and prediction data for complex queries or data-mining applications

# Operational vs. Informational Systems

| Characteristic | Operational Systems | Informational Systems |
|---|---|---|
| Primary purpose | Run the business on a current basis | Support managerial decision making |
| Type of data | Current representation of state of the business | Historical point-in-time (snapshots) and predictions |
| Primary users | Clerks, salespersons, administrators | Managers, business analysts, customers |
| Scope of usage | Narrow, planned, and simple updates and queries | Broad, ad hoc, complex queries and analysis |
| Design goal | Performance: throughput, availability | Ease of flexible access and use |
| Volume | Many constant updates and queries on one or a few table rows | Periodic batch updates and queries requiring many or all rows |

# Issues with Company-Wide View

- ✖Inconsistent key structures
- ✖Synonyms
- ✖Free-form vs. structured fields
- ✖Inconsistent data values
- ✖Missing data

# Examples of heterogeneous data

STUDENT DATA

| StudentNo | LastName | MI | FirstName | Telephone | Status | • • • |
|---|---|---|---|---|---|---|
| 123-45-6789 | Enright | T | Mark | 483-1967 | Soph | |
| 389-21-4062 | Smith | R | Elaine | 283-4195 | Jr | |

STUDENT EMPLOYEE

| StudentID | Address | Dept | Hours | • • • |
|---|---|---|---|---|
| 123-45-6789 | 1218 Elk Drive, Phoenix, AZ 91304 | Soc | 8 | |
| 389-21-4062 | 134 Mesa Road, Tempe, AZ 90142 | Math | 10 | |

STUDENT HEALTH

| StudentName | Telephone | Insurance | ID | • • • |
|---|---|---|---|---|
| Mark T. Enright | 483-1967 | Blue Cross | 123-45-6789 | |
| Elaine R. Smith | 555-7828 | ? | 389-21-4062 | |

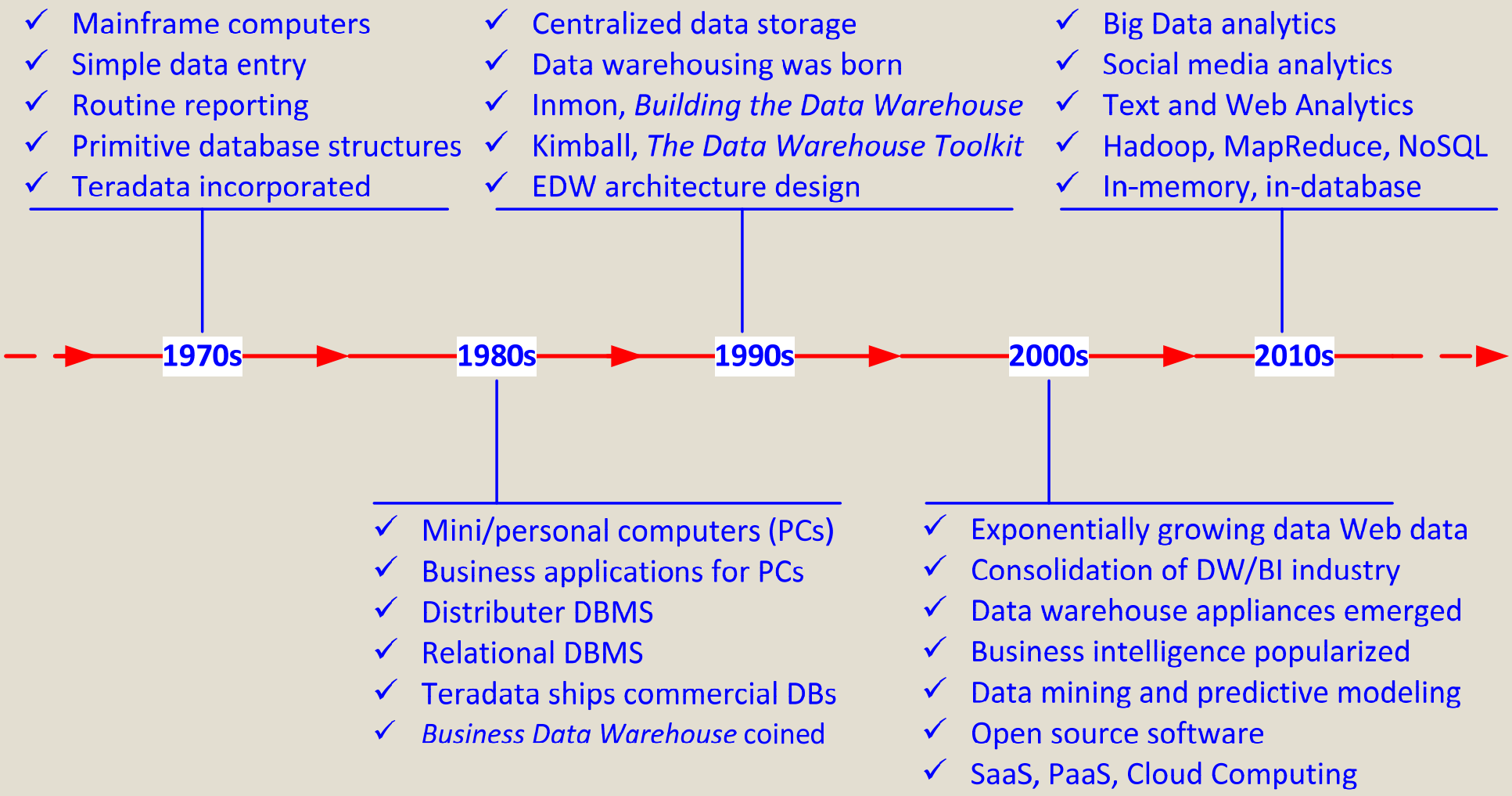
# Organizational Trends Motivating Data Warehouses

- No single system of records
- Multiple systems not synchronized
- Organizational need to analyze activities in a balanced way
- Customer relationship management
- Supplier relationship management

# Characteristics of DWs

- Subject oriented
- Integrated
- Time-variant (time series)
- Non-volatile
- Summarized
- Not normalized
- Metadata
- Relational/multidimensional
- Web based, client/server, cloud-based
- Real-time/right-time/active...

# A Historical Perspective to Data Warehousing



# Purpose of Data Warehouse

- Support business performance management (business intelligence)
- Support knowledge discovery and data mining (business analytics)

# Business Performance Mgmt (BPM)

## Sample Dashboard

BPM systems allow managers to measure, monitor, and manage key activities and processes to achieve organizational goals. Dashboards are often used to provide an information system in support of BPM.


Charts like these are examples of **data visualization**, the representation of data in graphical and multimedia formats for human analysis.

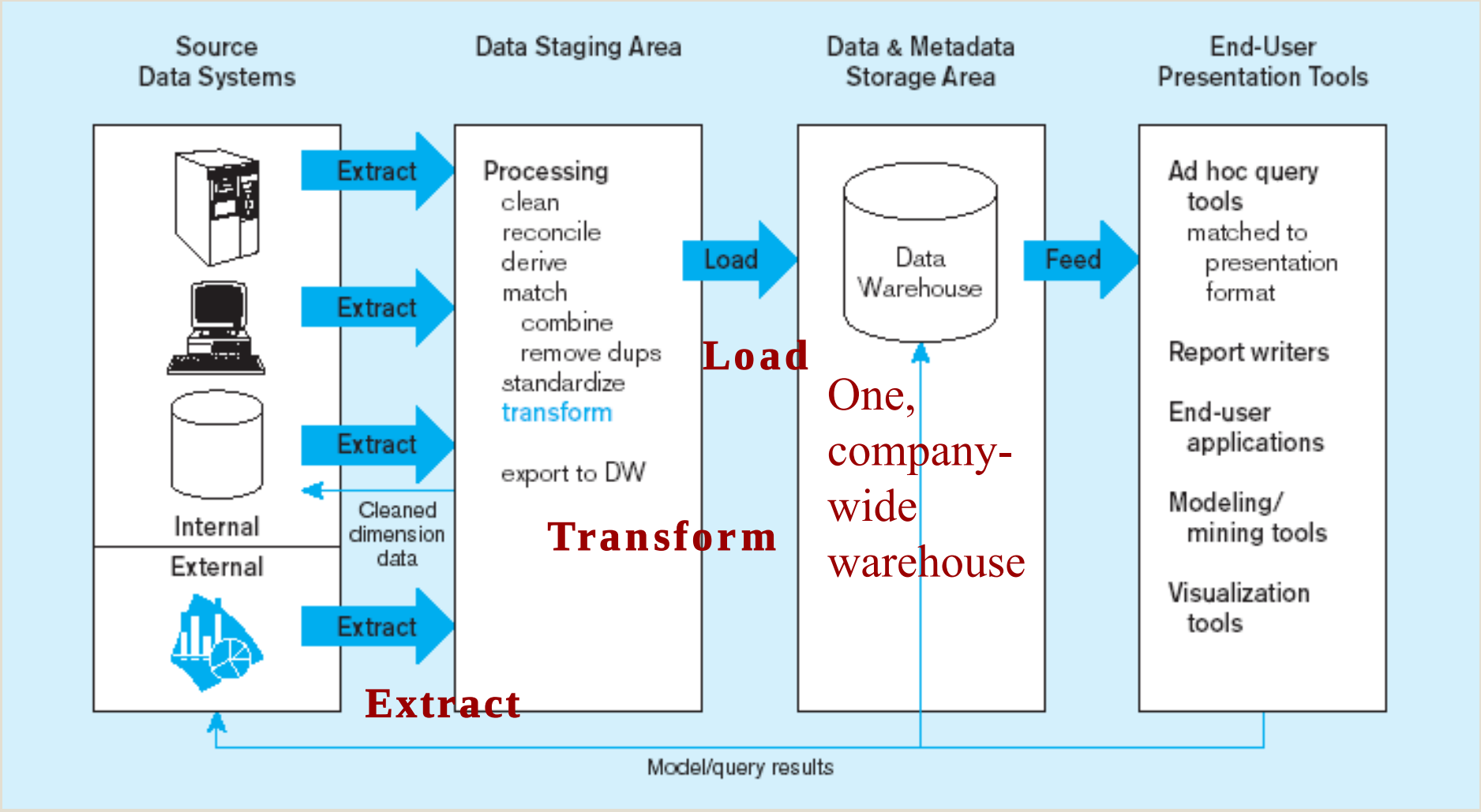
# Data Mining

| Technique | Function |
|---|---|
| Regression | Test or discover relationships from historical data |
| Decision tree induction | Test or discover if . . . then rules for decision propensity |
| Clustering and signal processing | Discover subgroups or segments |
| Affinity | Discover strong mutual relationships |
| Sequence association | Discover cycles of events and behaviors |
| Case-based reasoning | Derive rules from real-world case examples |
| Rule discovery | Search for patterns and correlations in large data sets |
| Fractals | Compress large databases without losing information |
| Neural nets | Develop predictive models based on principles modeled after the human brain |

# Data Warehouse Architectures

- Generic Architecture
- Independent Data Mart
- Dependent Data Mart and Operational Data Store
- Logical Data Mart and Real-Time Data Warehouse
- Three-Layer architecture

# Generic data warehousing architecture

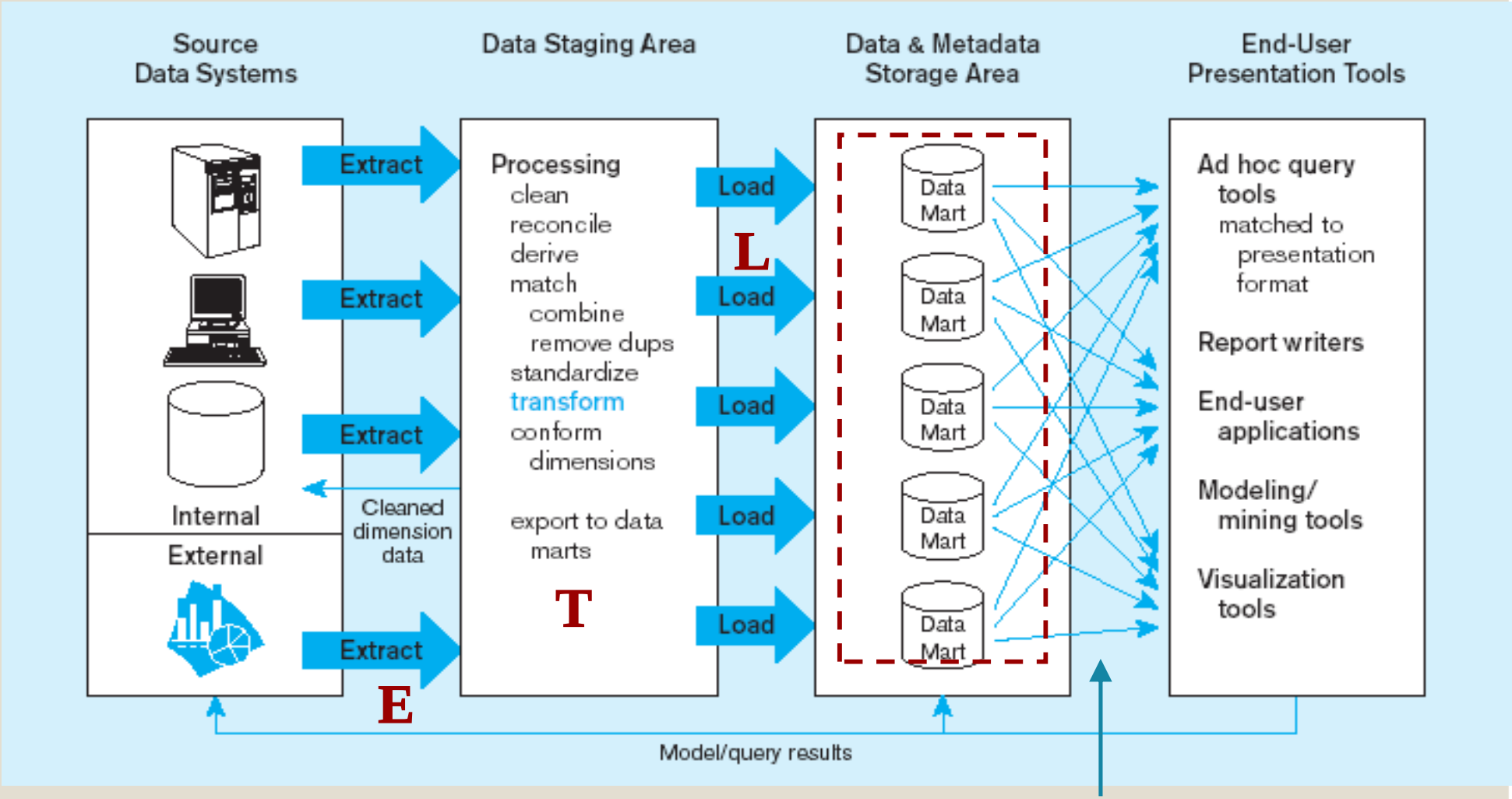



Periodic extraction ➔ data is not completely current in warehouse

Handout 6

# Independent data mart data warehousing architecture

**Data marts:**
Mini-warehouses, limited in scope



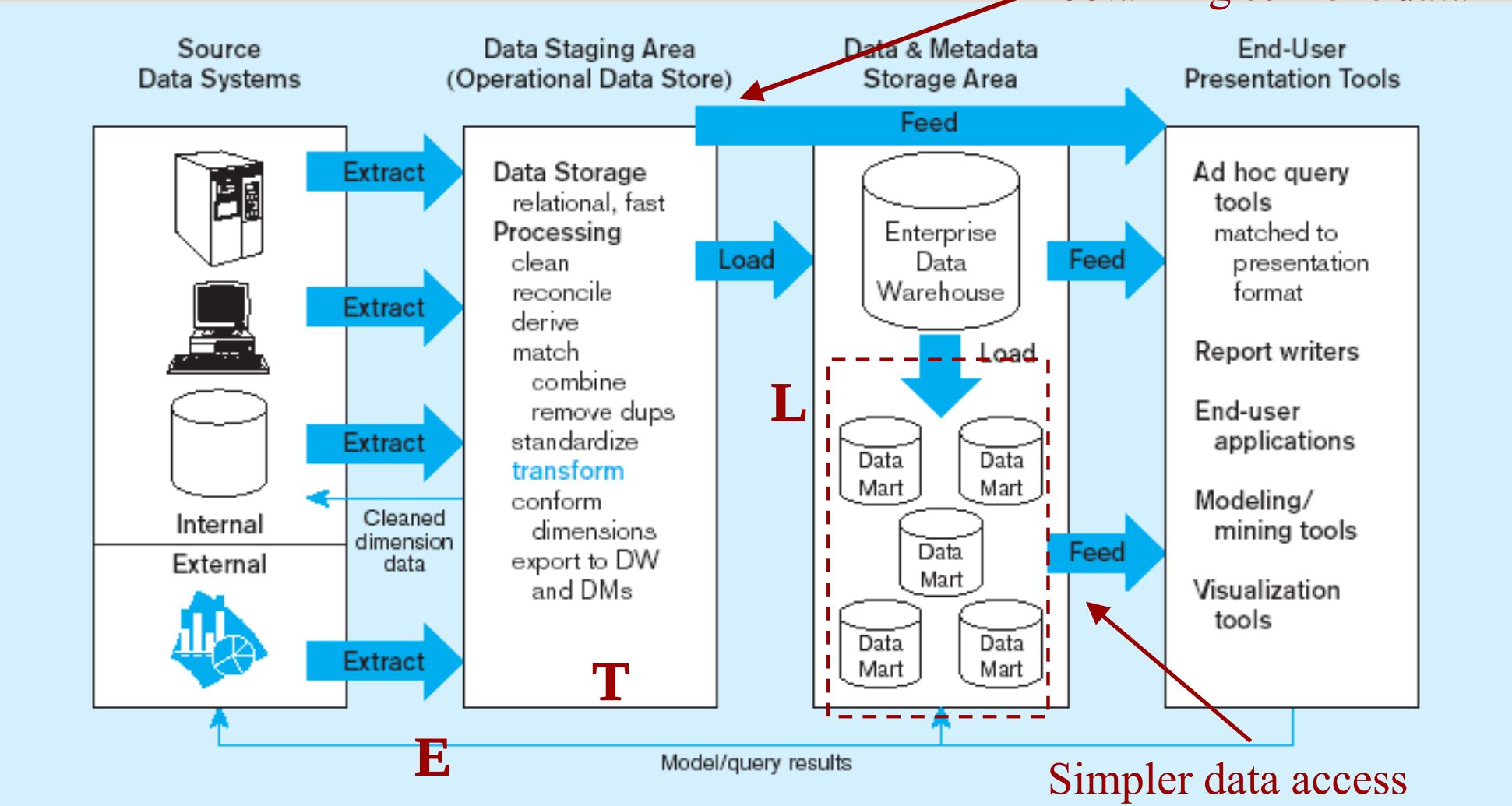

Separate ETL for each ***independent*** data mart

Data access complexity due to ***multiple*** data marts

Handout 6

# Dependent data mart with operational data store: a three-level architecture

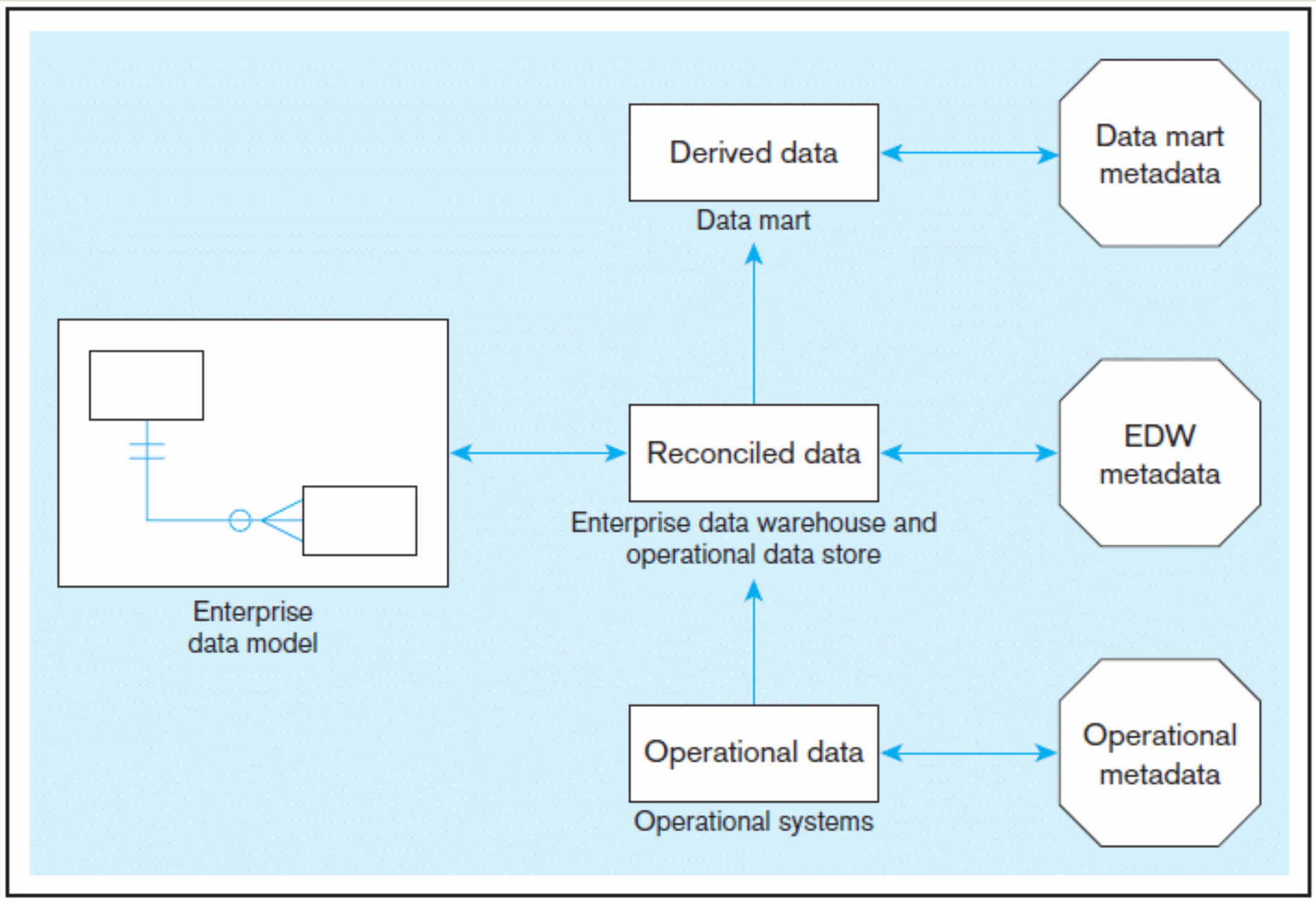
**ODS** provides option for obtaining ***current*** data




Single ETL for ***enterprise data warehouse (EDW)***

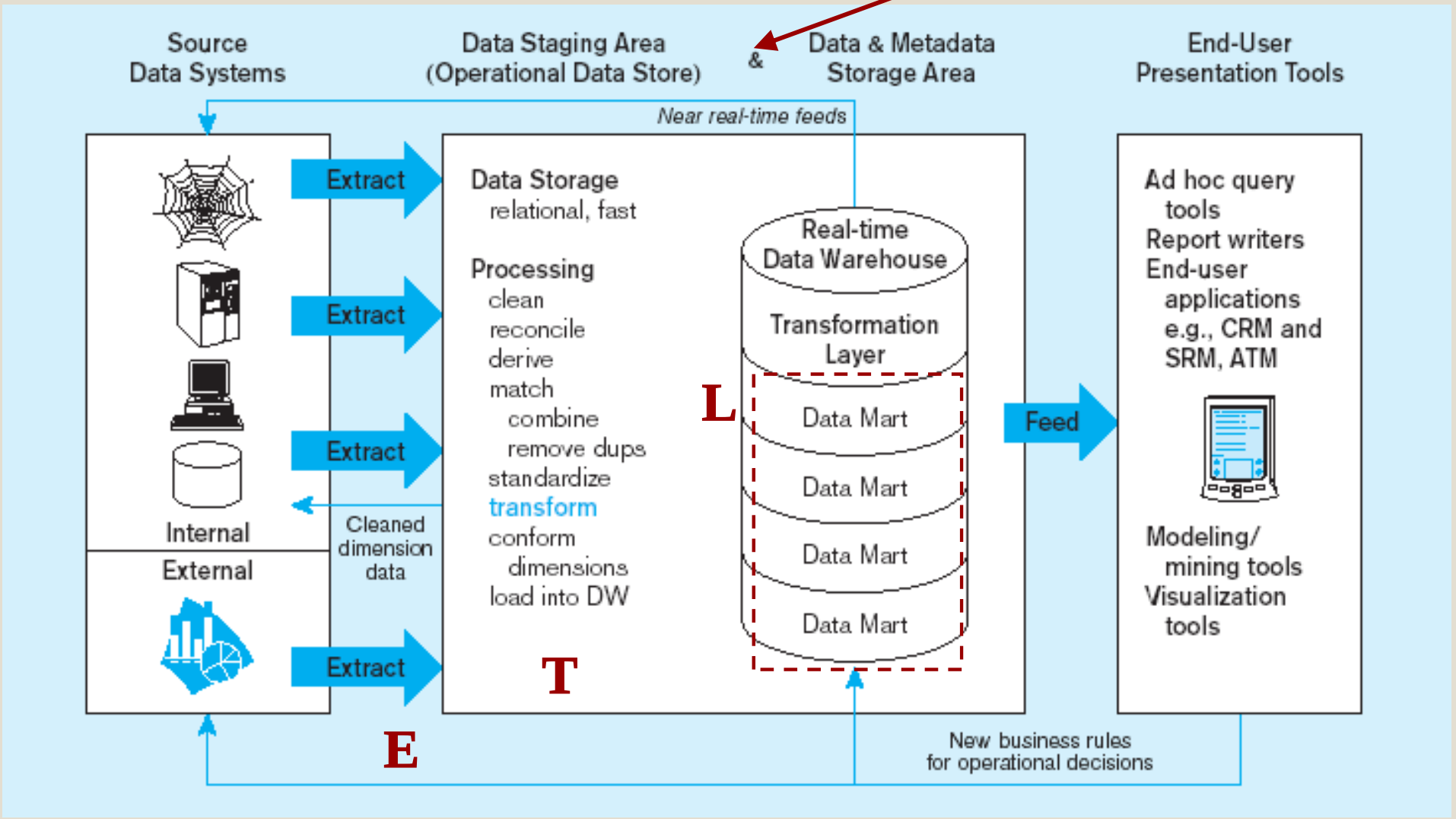
***Dependent*** data marts loaded from EDW

# Three-layer data architecture for a data warehouse



# Logical data mart and real time warehouse architecture

**ODS** and **data warehouse** are one and the same

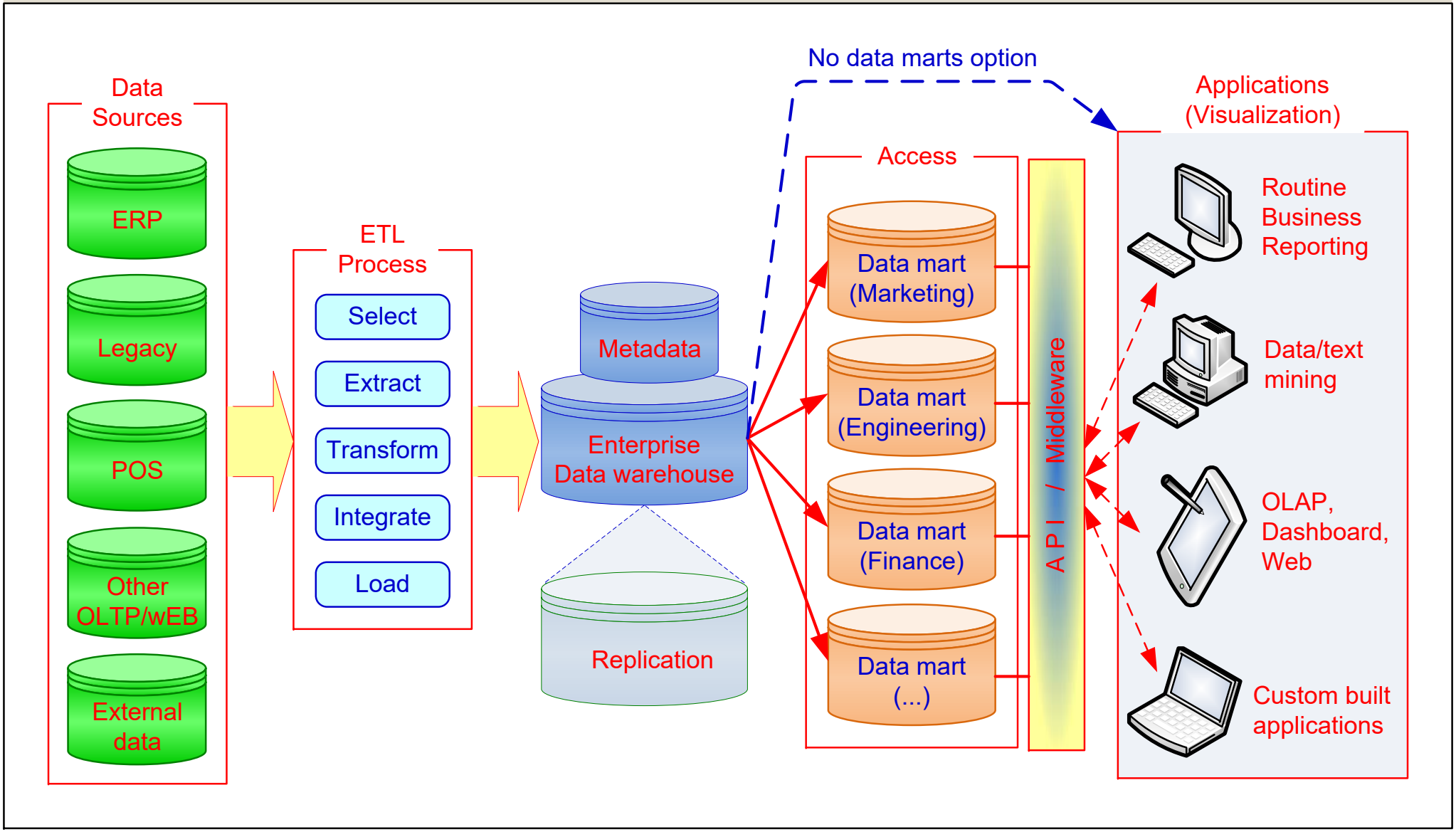



Near real-time ETL for ***Data Warehouse***

Data marts here are NOT separate databases, but logical ***views*** of the data warehouse

➔ Easier to create new data marts

# Another Look



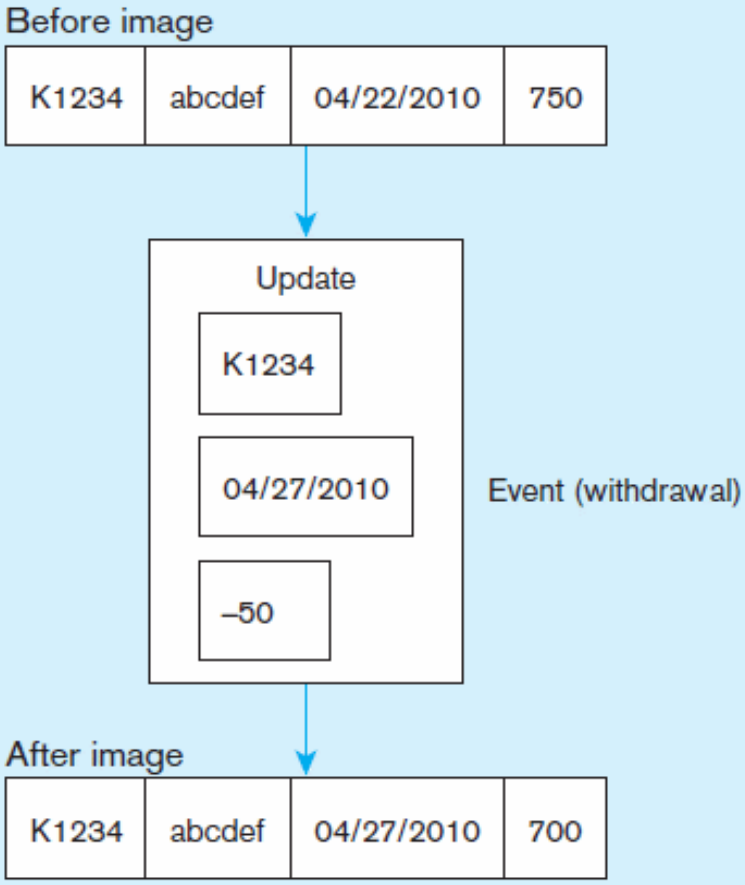
# Data Warehouse vs Data Mart

| Data Warehouse | Data Mart |
|---|---|
| **Scope** | **Scope** |
| • Application independent<br>• Centralized, possibly enterprise-wide<br>• Planned | • Specific DSS application<br>• Decentralized by user area<br>• Organic, possibly not planned |
| **Data** | **Data** |
| • Historical, detailed, and summarized<br>• Lightly denormalized | • Some history, detailed, and summarized<br>• Highly denormalized |
| **Subjects** | **Subjects** |
| • Multiple subjects | • One central subject of concern to users |
| **Sources** | **Sources** |
| • Many internal and external sources | • Few internal and external sources |
| **Other Characteristics** | **Other Characteristics** |
| • Flexible<br>• Data oriented<br>• Long life<br>• Large<br>• Single complex structure | • Restrictive<br>• Project oriented<br>• Short life<br>• Start small, becomes large<br>• Multi, semi-complex structures, together complex |

Handout 6

# Data Characteristics
## Status vs. Event Data

Example of DBMS log entry




Status

Event = a database action (create/update/delete) that results from a transaction

Status

Transient operational data

# Data Characteristics

## Transient vs. Periodic Data

Table X (10/05)

| Key | A | B |
|---|---|---|
| 001 | a | b |
| 002 | c | d |
| 003 | e | f |
| 004 | g | h |

Table X (10/06)

| Key | A | B |
|---|---|---|
| 001 | a | b |
| 002 | r | d |
| 003 | e | f |
| 004 | y | h |
| 005 | m | n |

Table X (10/07)

| Key | A | B |
|---|---|---|
| 001 | a | b |
| 002 | r | d |
| 003 | e | t |
| | | |
| 005 | m | n |

With transient data, changes to existing records are written over previous records, thus destroying the previous data content

Handout 6

Periodic warehouse data

# Data Characteristics

## Transient vs. Periodic Data

Table X (10/05)

| Key | Date | A | B | Action |
|---|---|---|---|---|
| 001 | 10/03 | a | b | C |
| 002 | 10/03 | c | d | C |
| 003 | 10/03 | e | f | C |
| 004 | 10/03 | g | h | C |

Table X (10/06)

| Key | Date | A | B | Action |
|---|---|---|---|---|
| 001 | 10/05 | a | b | C |
| 002 | 10/05 | c | d | C |
| 002 | 10/06 | r | d | U |
| 003 | 10/05 | e | f | C |
| 004 | 10/05 | g | h | C |
| 004 | 10/06 | y | h | U |
| 005 | 10/06 | m | n | C |

Table X (10/07)

| Key | Date | A | B | Action |
|---|---|---|---|---|
| 001 | 10/05 | a | b | C |
| 002 | 10/05 | c | d | C |
| 002 | 10/06 | r | d | U |
| 003 | 10/05 | e | f | C |
| 003 | 10/07 | e | t | U |
| 004 | 10/05 | g | h | C |
| 004 | 10/06 | y | h | U |
| 004 | 10/07 | y | h | D |
| 005 | 10/06 | m | n | C |

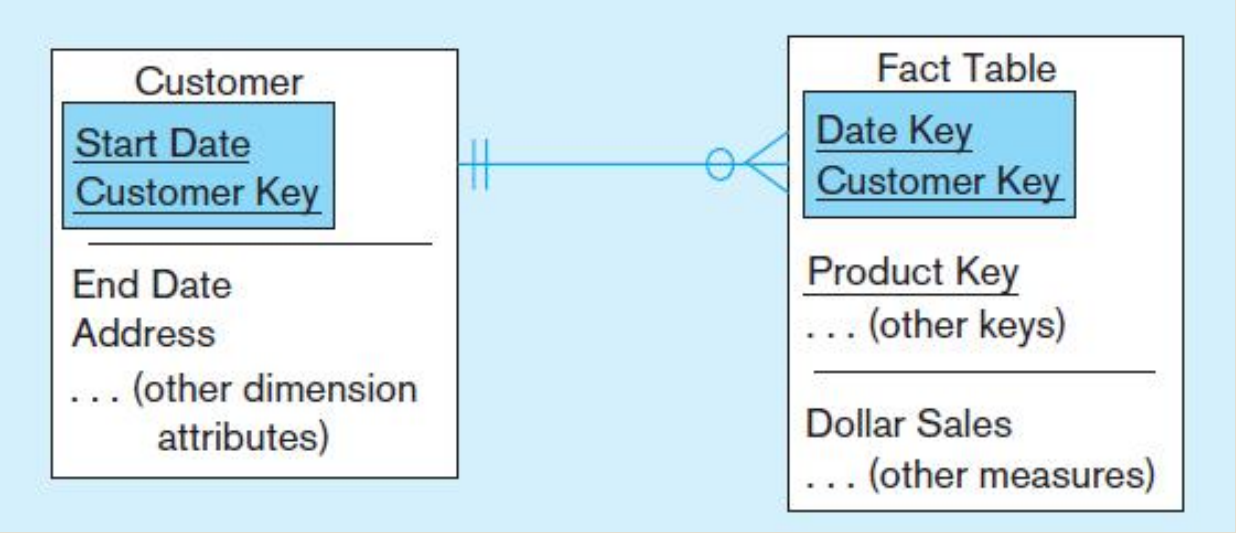
Periodic data are never physically altered or deleted once they have been added to the store

Handout 6

# Slowly Changing Dimensions (SCD)

- How to maintain knowledge of the past
- SCD Types:
  - Type 1: just replace old data with new (lose historical data)
  - Type 2: create a new dimension table row each time the dimension object changes, with all dimension characteristics at the time of change. Most common approach.
  - Type 3: for each changing attribute, create a current value field and several old-valued fields
  - Type 4: add a history table
  - Type 6: hybrid

# Example of Type 2 SCD Customer dimension table




The dimension table contains several records for the same customer. The specific customer record to use depends on the key and the date of the fact, which should be between start and end dates of the SCD customer record.

# The ETL Process

- Capture/Extract
- Scrub or data cleansing
- Transform
- Load and Index

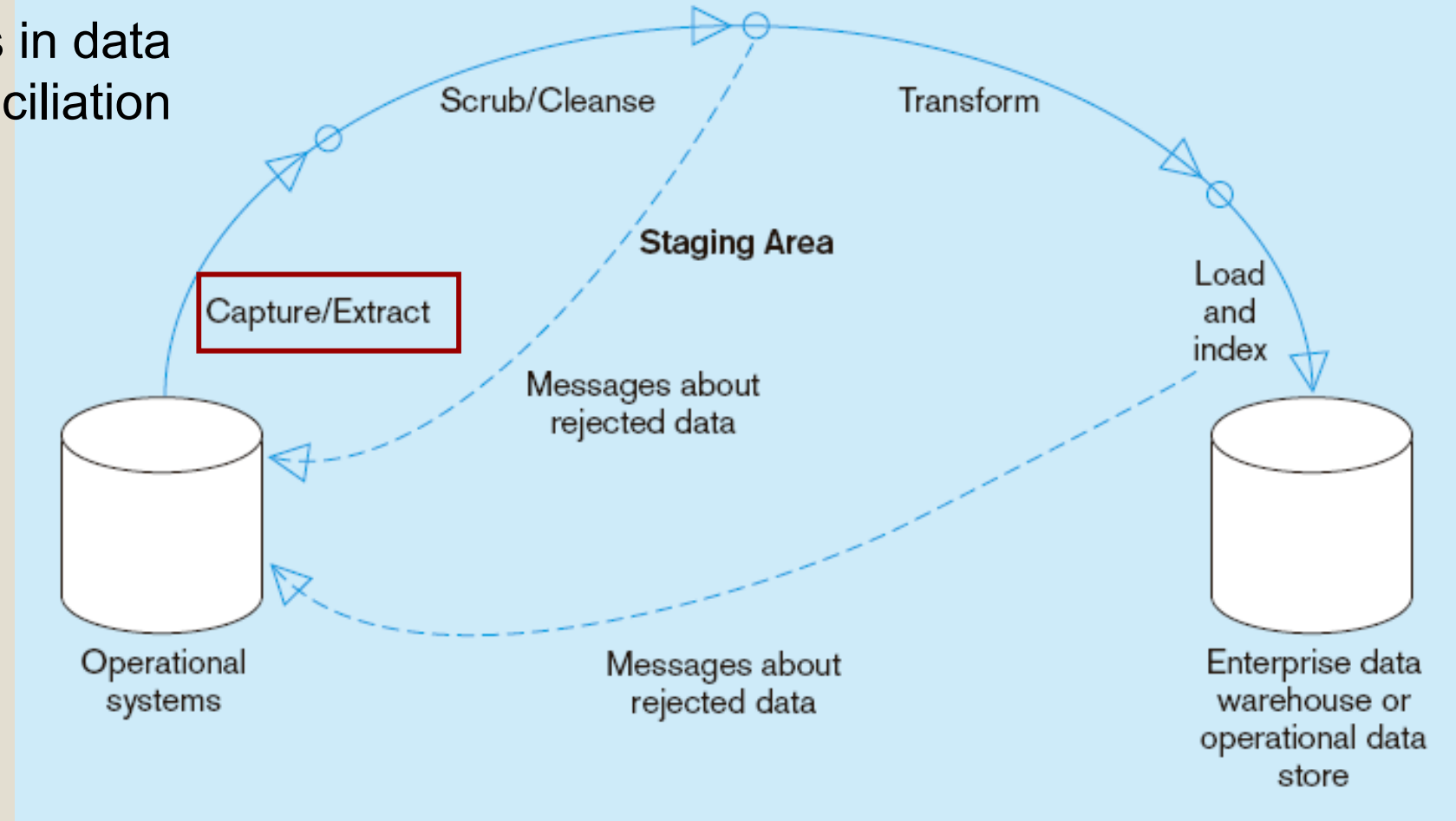
**ETL = Extract, transform, and load**

# The Reconciled Data Layer

- Typical operational data is:
  - Transient–not historical
  - Restricted in scope–not comprehensive
  - Sometimes poor quality–inconsistencies and errors
- After ETL, data should be:
  - Detailed–not summarized yet
  - Historical–periodic
  - Comprehensive–enterprise-wide perspective
  - Timely–data should be current enough to assist decision-making
  - Quality controlled–accurate with full integrity

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

Steps in data reconciliation

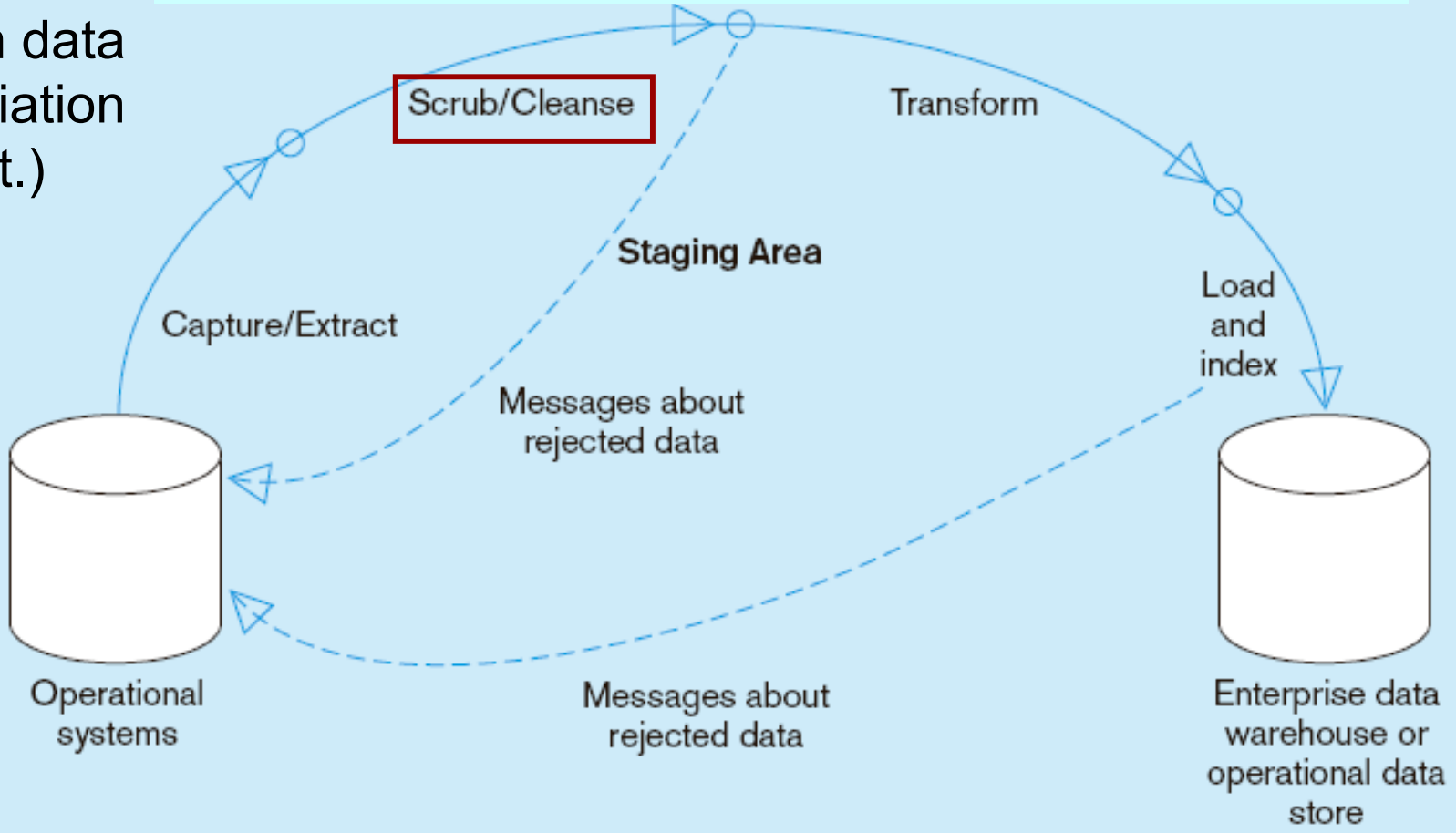



**Static extract** = capturing a snapshot of the source data at a point in time

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

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

Steps in data reconciliation (cont.)




**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

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

Steps in data reconciliation (cont.)

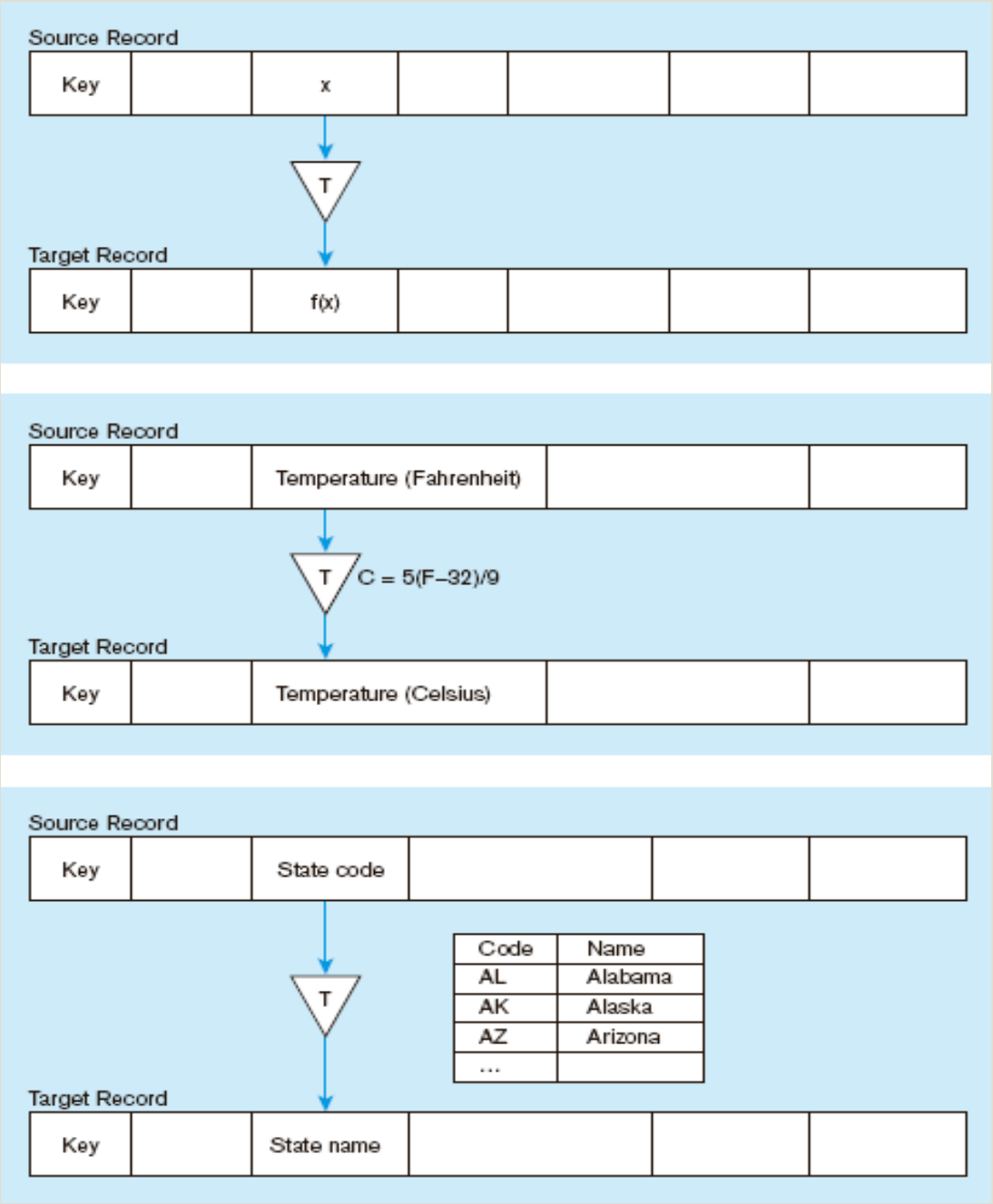

**Record-level:**

*Selection*–data partitioning (e.g., only data created after 2018)
*Joining*–data combining
*Aggregation*–data summarization

**Field-level:**

*single-field*–from one field to one field (e.g., Celsius to Fahrenheit)
*multi-field*–from many fields to one, or one field to many

# Single-field transformation



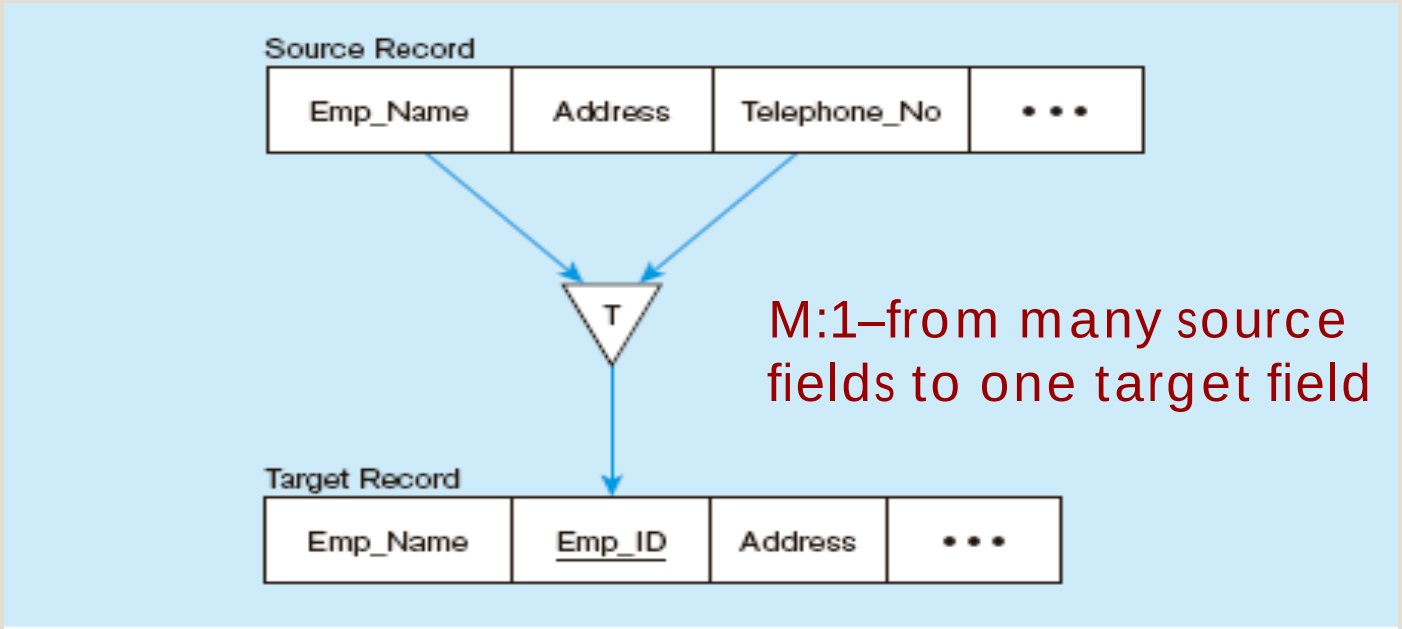

In general–some transformation function translates data from old form to new form

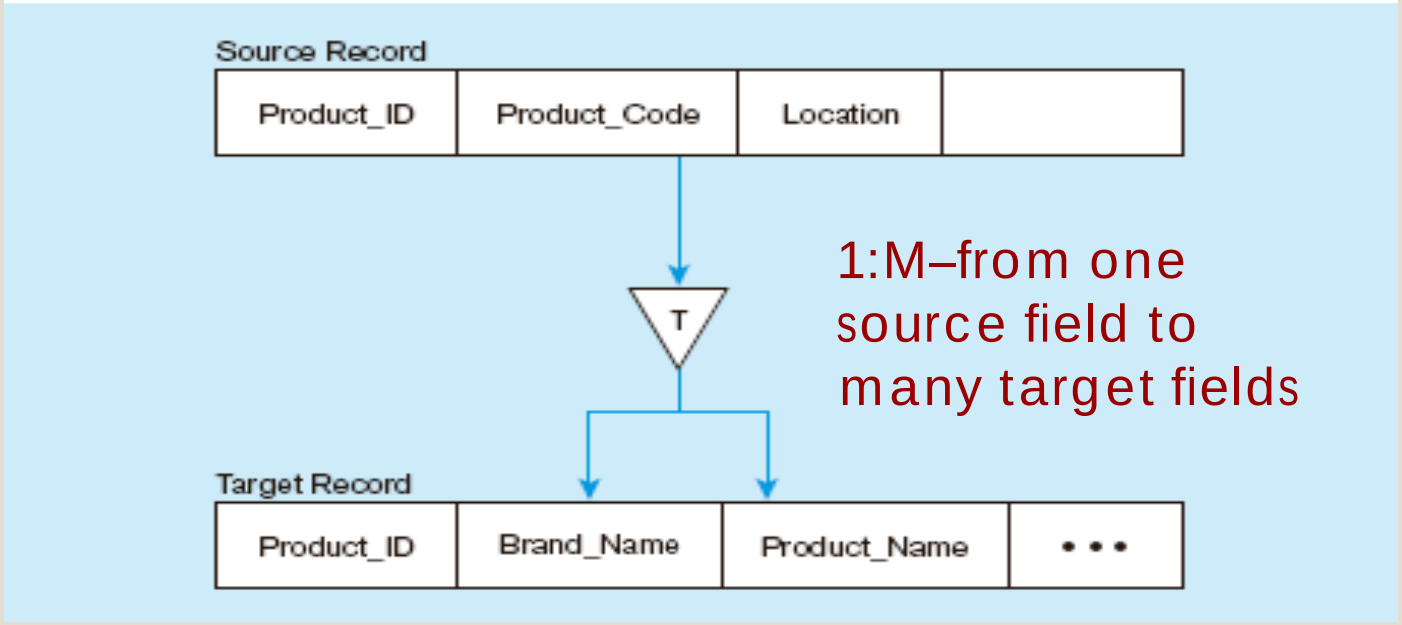
***Algorithmic*** transformation uses a formula or logical expression

***Table lookup***–another approach, uses a separate table keyed by source record code

Handout 6

# Multi-field transformation

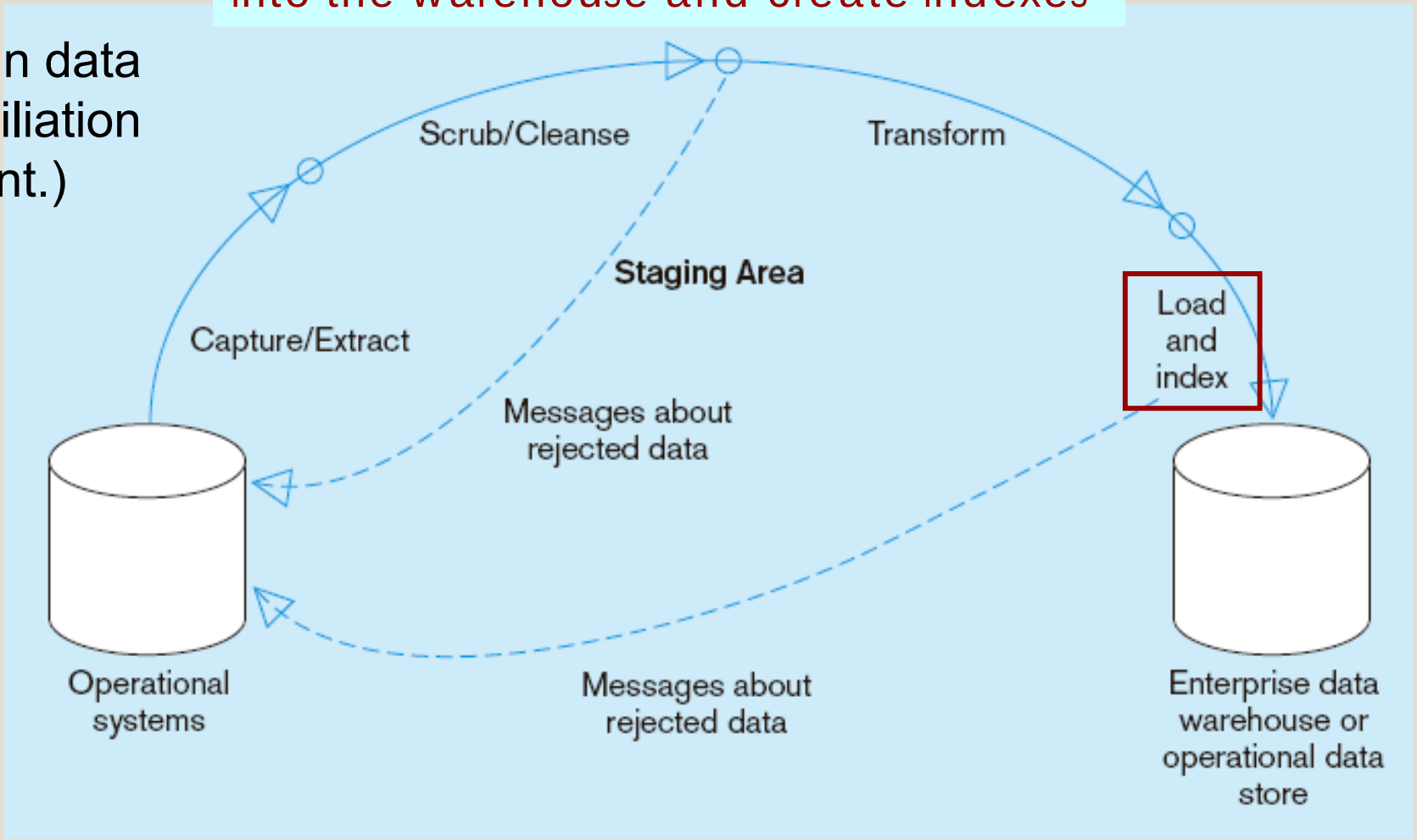



Handout 6

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

Steps in data reconciliation (cont.)




**Refresh mode:** bulk rewriting of target data at periodic intervals

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

Handout 6

# Derived Data

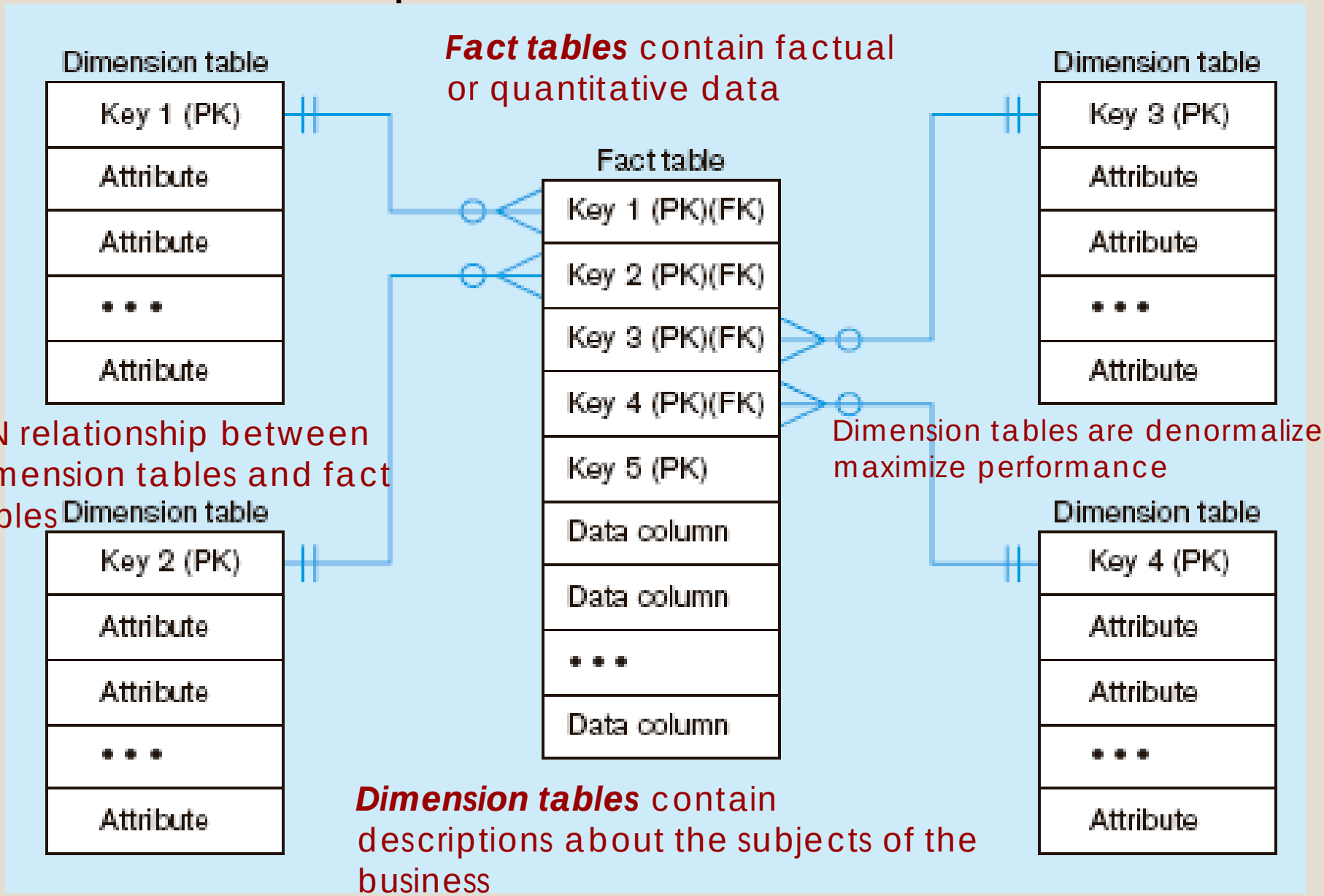
- The data layer associated with logical or physical data marts
- Users normally interact with this layer for their decision support applications.
- Objectives
  - Ease of use for decision support applications
  - Fast response to predefined user queries
  - Customized data for particular target audiences
  - Ad-hoc query support
  - Data mining capabilities

# Derived Data

- Characteristics
  - Detailed (mostly periodic) data
  - Aggregate (for summary)
  - Distributed (to departmental servers)

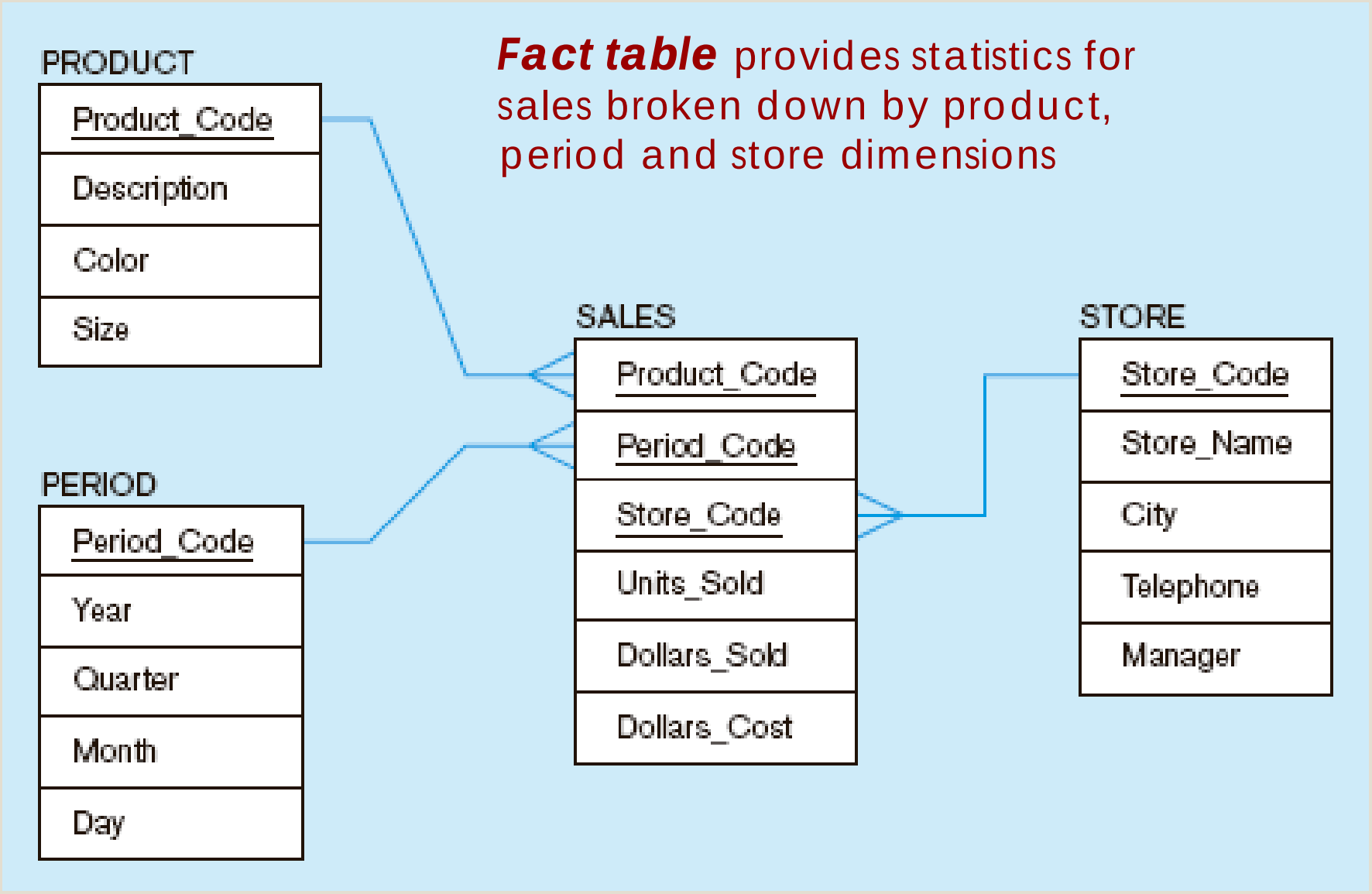
Most common data model = **star schema**
(also called “dimensional model”)

# Components of a **star schema**

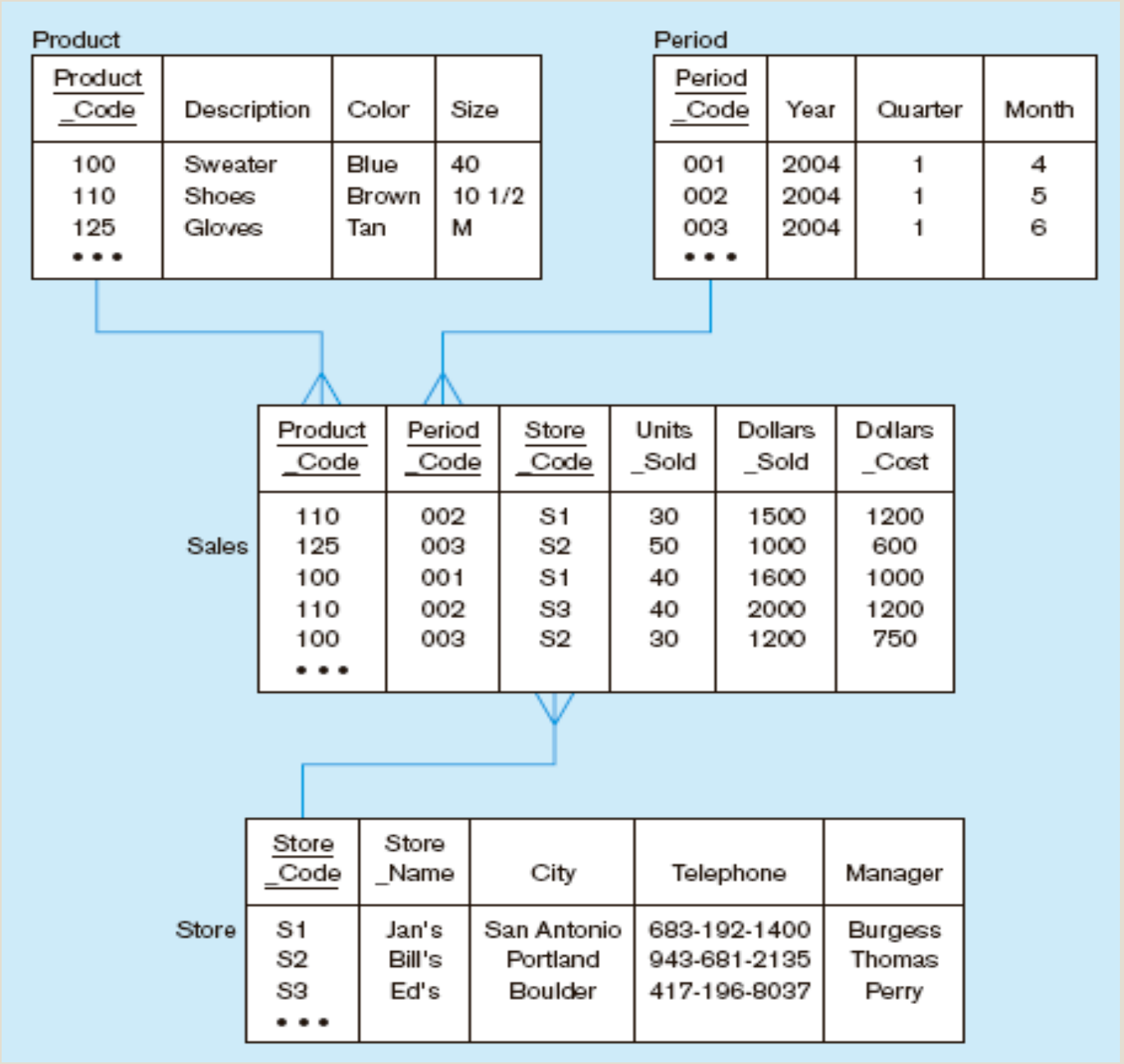



Excellent for ad-hoc queries, but bad for online transaction processing

# Star schema example



# Star schema with sample data



Product

| Product_Code | Description | Color | Size |
|---|---|---|---|
| 100 | Sweater | Blue | 40 |
| 110 | Shoes | Brown | 10 1/2 |
| 125 | Gloves | Tan | M |
| • • • | | | |

Period

| Period_Code | Year | Quarter | Month |
|---|---|---|---|
| 001 | 2004 | 1 | 4 |
| 002 | 2004 | 1 | 5 |
| 003 | 2004 | 1 | 6 |
| • • • | | | |

Sales

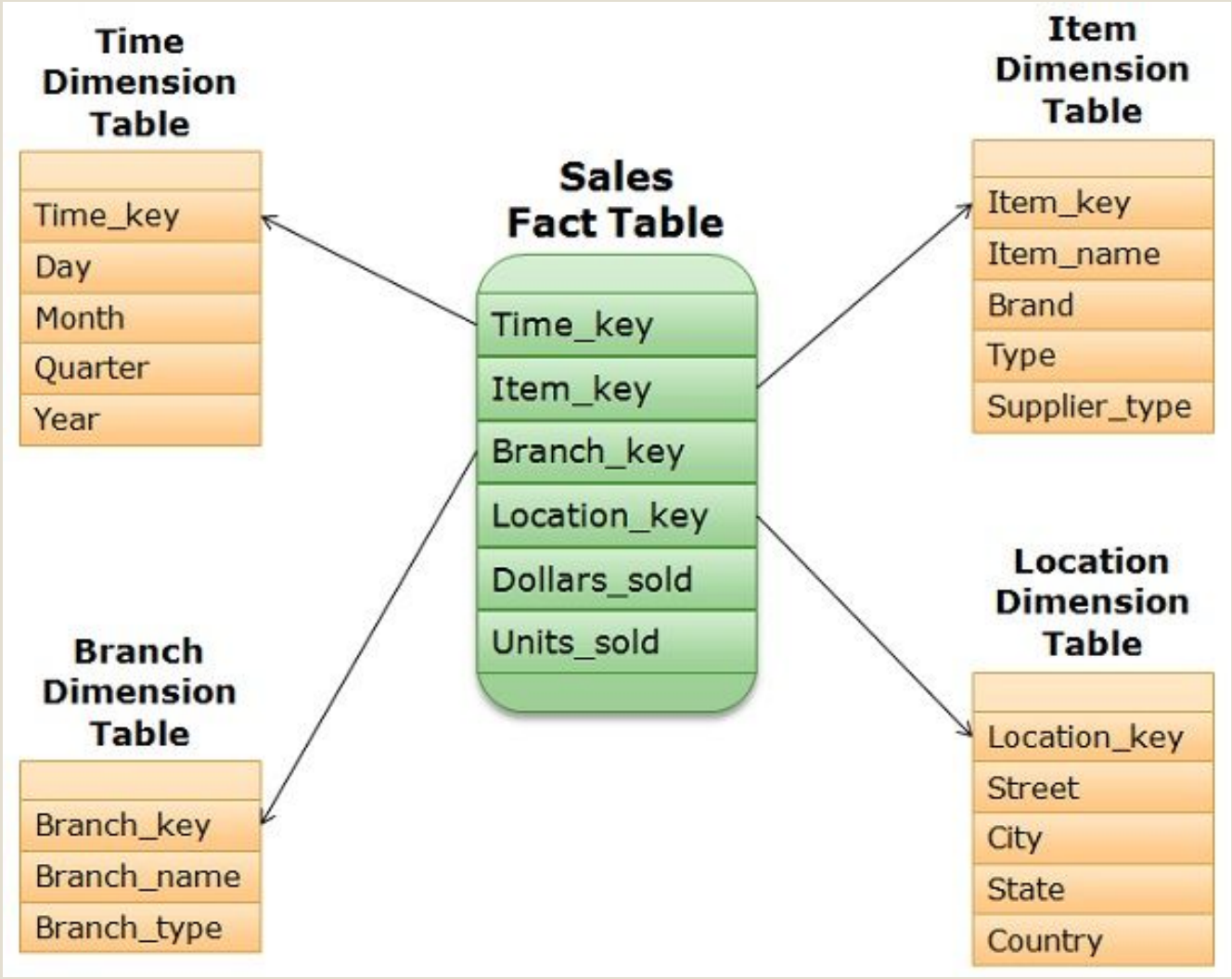
| Product_Code | Period_Code | Store_Code | Units_Sold | Dollars_Sold | Dollars_Cost |
|---|---|---|---|---|---|
| 110 | 002 | S1 | 30 | 1500 | 1200 |
| 125 | 003 | S2 | 50 | 1000 | 600 |
| 100 | 001 | S1 | 40 | 1600 | 1000 |
| 110 | 002 | S3 | 40 | 2000 | 1200 |
| 100 | 003 | S2 | 30 | 1200 | 750 |
| • • • | | | | | |

Store

| Store_Code | Store_Name | City | Telephone | Manager |
|---|---|---|---|---|
| S1 | Jan's | San Antonio | 683-192-1400 | Burgess |
| S2 | Bill's | Portland | 943-681-2135 | Thomas |
| S3 | Ed's | Boulder | 417-196-8037 | Perry |
| • • • | | | | |

Handout 6

# Star Schema: Another Example



# Characteristics of Star Schema

- Every dimension in a star schema is represented with the only one-dimension table.
- The dimension table should contain the set of attributes.
- The dimension table is joined to the fact table using a foreign key, but not to other dimension table
- Fact table would contain key and measure.
- The star schema is easy to understand and provides optimal disk usage.
- The dimension tables are not normalized. For instance, in the above figure, Country_ID does not have Country lookup table as an OLTP design would have.
- The schema is widely supported by BI Tools

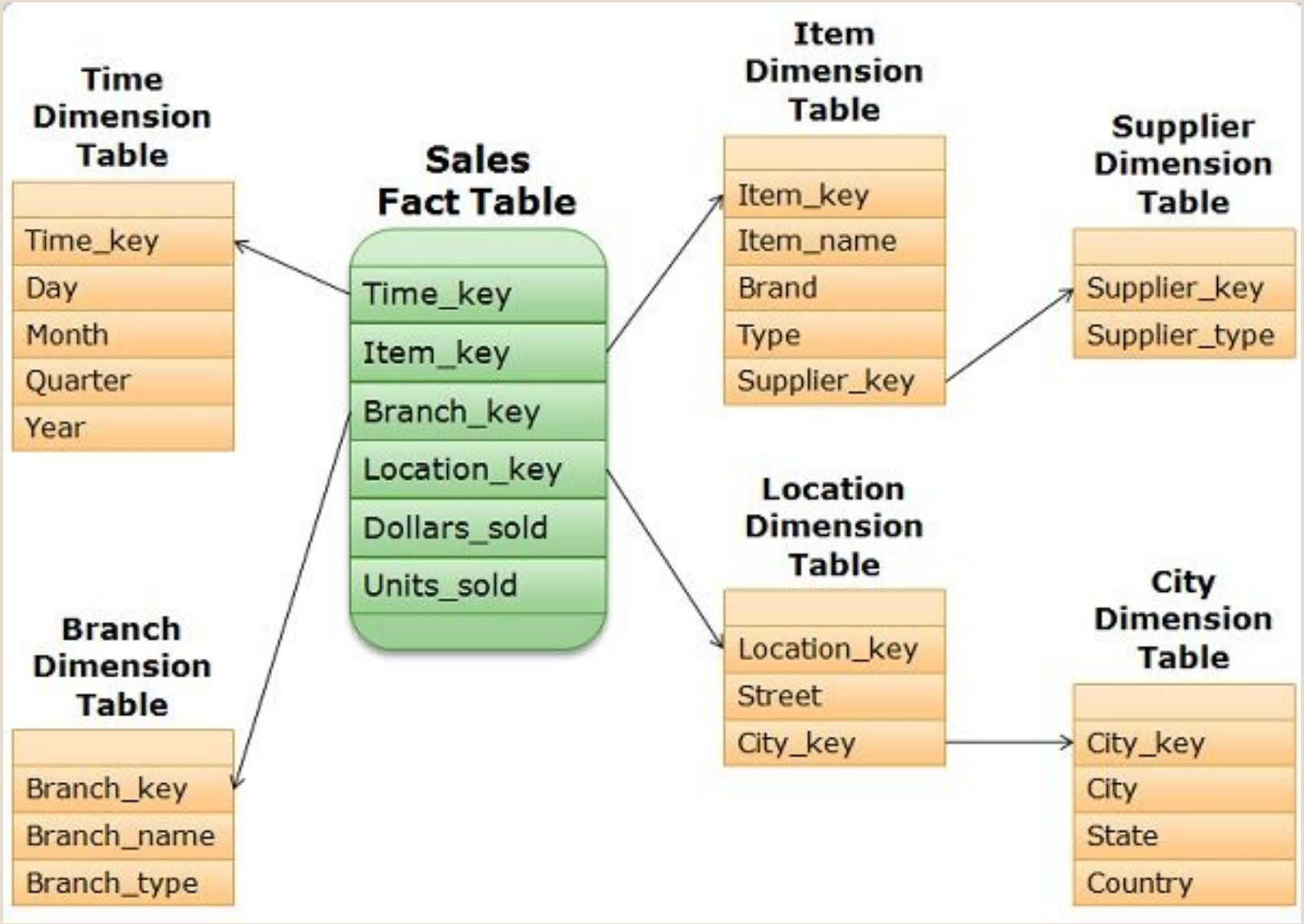
# Issues Regarding Star Schema

- Dimension table keys must be ***surrogate*** (non-intelligent and non-business related), because:
  - Keys may change over time
  - Length/format consistency
- Granularity of Fact Table – what level of detail do you want?
  - Transactional grain – finest level
  - Aggregated grain – more summarized
  - Finer grain ➔ better ***market basket analysis*** capability
  - Finer grain ➔ more dimension tables, more rows in fact table
- Duration of the database–how much history should be kept?
  - Natural duration–13 months or 5 quarters
  - Financial institutions may need longer duration
  - Older data is more difficult to source and cleanse

# Other Extensions

- Snowflake schema
- Galaxy schema

# Snowflake Schema



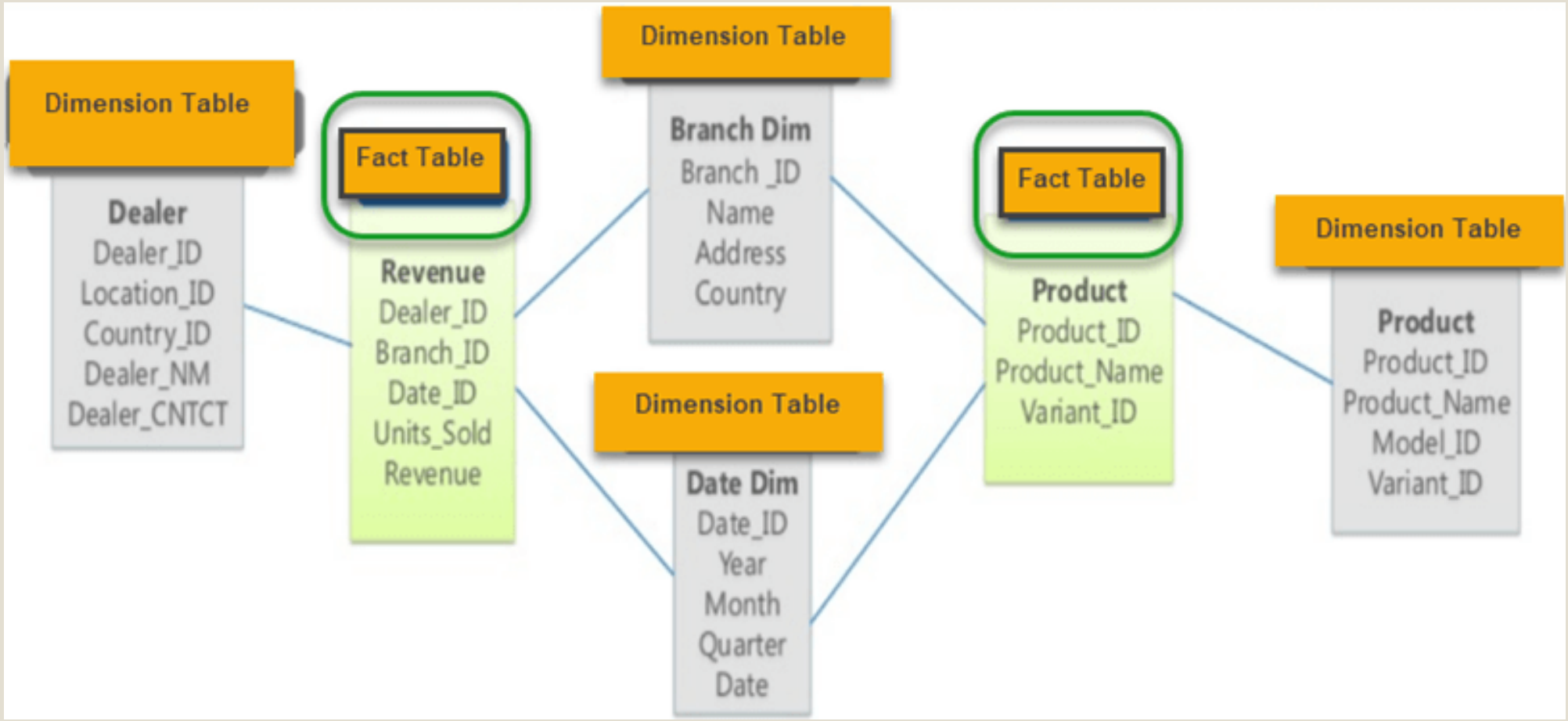
# Characteristics of Snowflake Schema

- Extension of the star schema
- The main benefit is that it uses smaller disk space.
- Easier to implement when a dimension is added to the schema
- Due to multiple tables, query performance is reduced
- Data schema is more complex

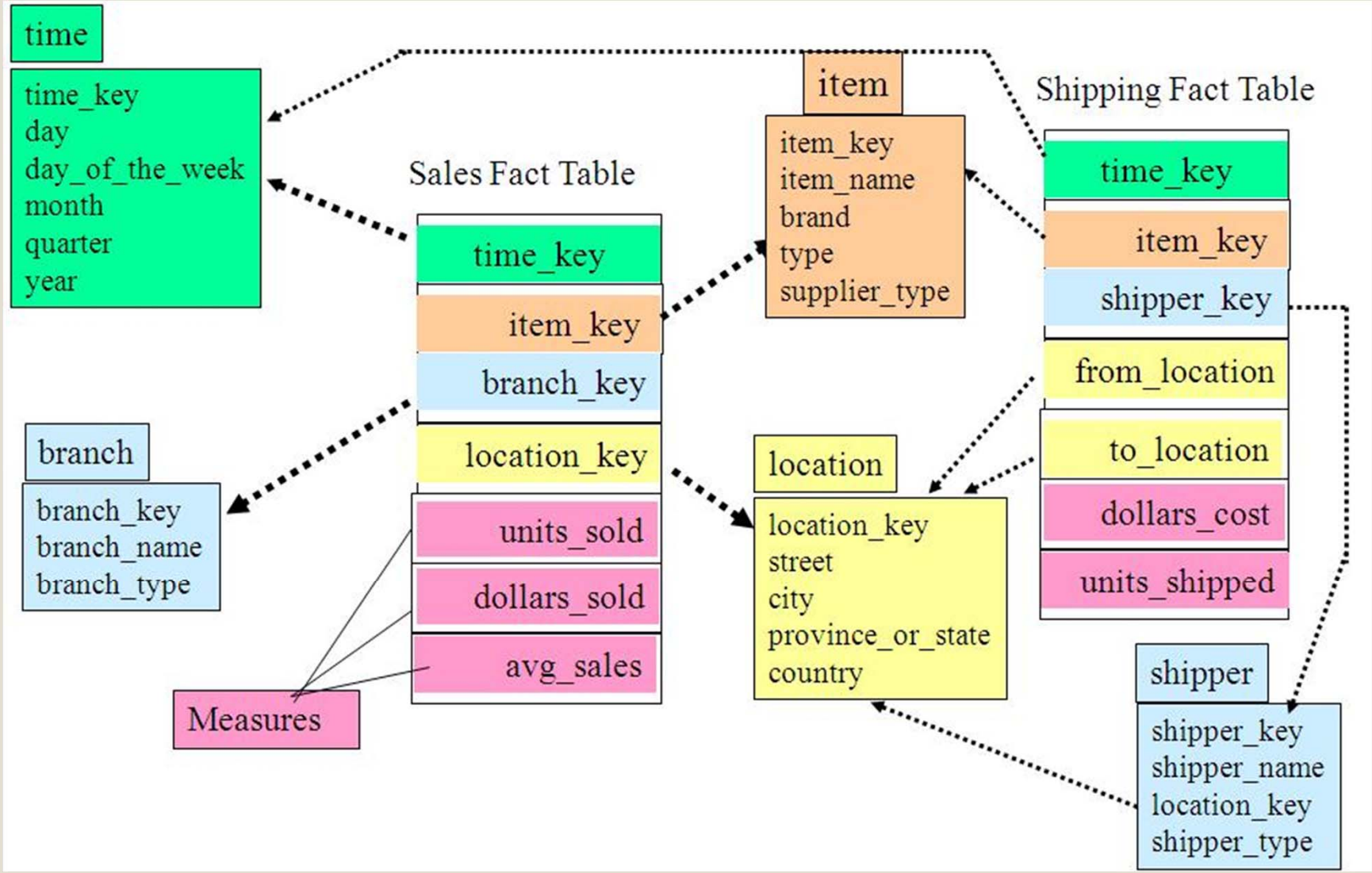
# Star vs. Snowflake Schema

| Star Schema | Snowflake Schema |
|---|---|
| Hierarchies for the dimensions are stored in the dimensional table. | Hierarchies are divided into separate tables. |
| It contains a fact table surrounded by dimension tables. | One fact table surrounded by dimension tables which are in turn surrounded by dimension tables |
| Only a single join is need to link between the fact table and any dimension tables. | A snowflake schema requires many joins to fetch the data. |
| Simple DB design. | More complex DB design. |
| Denormalized data structure and query also run faster. | Normalized data structure. |
| High level of data redundancy | Very low-level data redundancy |
| Single dimension table contains aggregated data. | Data split into different dimension tables. |
| Cube processing is faster. | Cube processing might be slow because of the complex join. |

# Galaxy Schema



# Galaxy Schema: Another Example



# Characteristics of Galaxy Schema

- Contains two or more fact tables that share dimension tables between them. It is also called the Fact Constellation Schema.
- Can be built by splitting the one-star schema into more star schemes.
- Helpful for aggregating fact tables for better understanding.

# 10 Essential Rules for Dimensional Modeling

- Use atomic facts
  - Users want detailed data eventually
- Create single-process fact tables
  - Each fact table should address the important measures of a business process
- Include a date dimension for each fact table
- Enforce consistent grain
- Disallow null keys in fact tables

# 10 Essential Rules for Dimensional Modeling

- Honor hierarchies
- Decode dimension tables
- Use surrogate keys
- Conform dimensions
  - A conformed dimension should be used across multiple fact tables.
- Balance requirements with actual data

# On-Line Analytical Processing (OLAP) Tools

- The use of a set of graphical tools that provides users with multidimensional views of their data and allows them to analyze the data using simple windowing techniques
- ***Relational OLAP (ROLAP)***
  - Traditional relational representation
- ***Multidimensional OLAP (MOLAP)***
  - **Cube** structure

# Multidimensionality

The ability to organize, present, and analyze data by several dimensions, such as sales by region, by product, by salesperson, and by time (four dimensions)

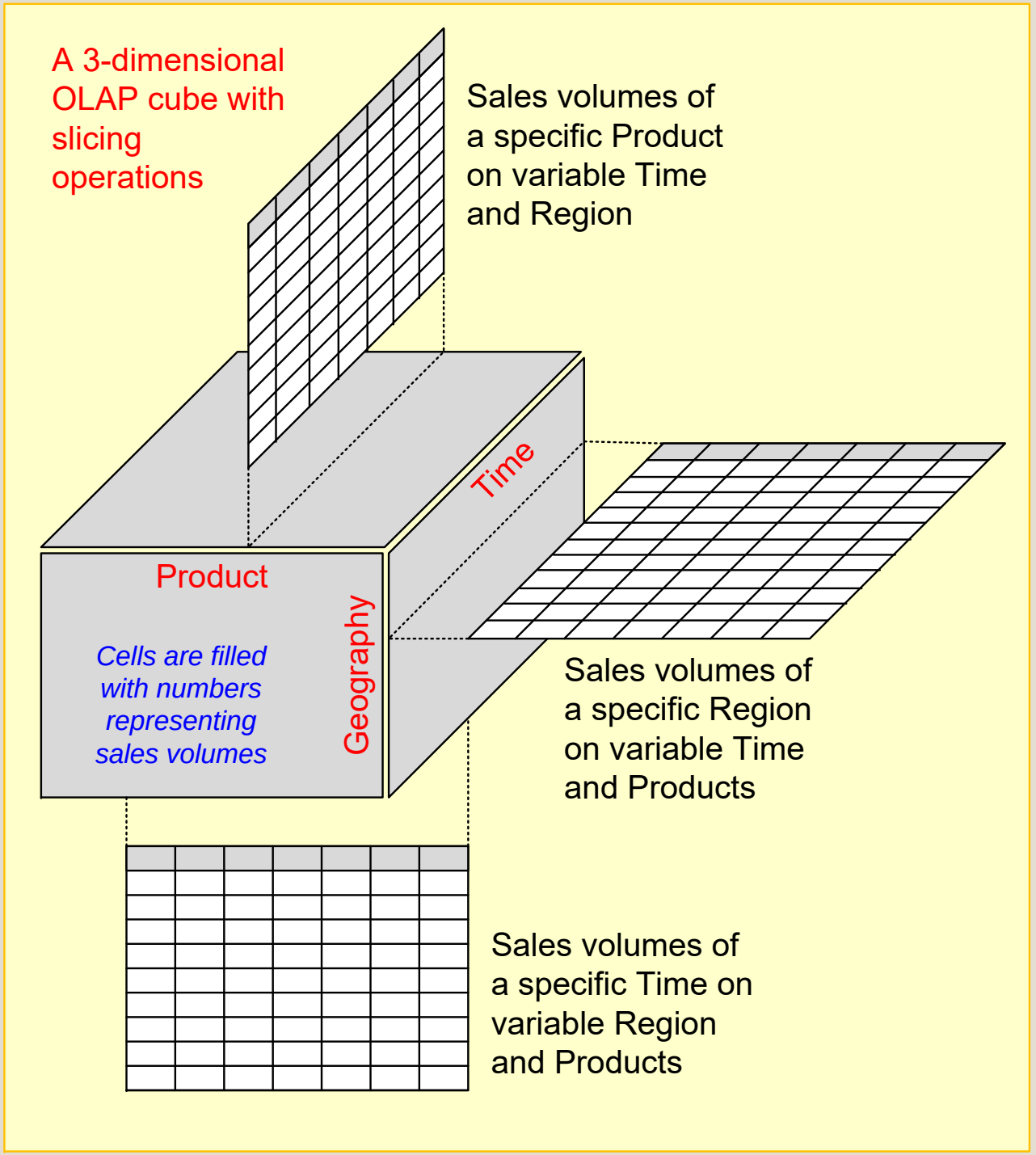
- Multidimensional presentation
  - Dimensions: products, salespeople, market segments, business units, geographical locations, distribution channels, country, or industry
  - Time dimension: daily, weekly, monthly, quarterly, or yearly
  - Measures: money, sales volume, head count, inventory profit, actual versus forecast

# OLAP Operations

- Slice - a subset of a multidimensional array
- Dice - a slice on more than two dimensions
- Drill Down/Up - navigating among levels of data ranging from the most summarized (up) to the most detailed (down)
- Roll Up - computing all of the data relationships for one or more dimensions
- Pivot - used to change the dimensional orientation of a report or an ad hoc query-page display

# OLAP

## Slicing Operations on a Simple Three-Dimensional Data Cube



# Example of drill-down

Starting with summary data, users can obtain details for particular cells

## Summary report

| Brand | Package size | Sales |
|---|---|---|
| SofTowel | 2-pack | $75 |
| SofTowel | 3-pack | $100 |
| SofTowel | 6-pack | $50 |

## Drill-down with color added

| Brand | Package size | Color | Sales |
|---|---|---|---|
| SofTowel | 2-pack | White | $30 |
| SofTowel | 2-pack | Yellow | $25 |
| SofTowel | 2-pack | Pink | $20 |
| SofTowel | 3-pack | White | $50 |
| SofTowel | 3-pack | Green | $25 |
| SofTowel | 3-pack | Yellow | $25 |
| SofTowel | 6-pack | White | $30 |
| SofTowel | 6-pack | Yellow | $20 |

# DW Data Catalog

- Identify subjects of the data mart
- Identify dimensions and facts
- Indicate how data is derived from enterprise data warehouses, including derivation rules
- Indicate how data is derived from operational data store, including derivation rules
- Identify available reports and predefined queries
- Identify data analysis techniques (e.g. drill-down)
- Identify responsible people

# DW Implementation Issues

- Identification of data sources and governance
  - Data silos!
  - Privacy
- Data quality planning, data model design
- ETL tool selection
- Establishment of service-level agreements
- Data transport, data conversion
- End-user support

# DW Implementation Issues

- On-premises vs Cloud-based
- On-premises data warehouses
  - Complete control over the tech stack
  - Local speed and performance; low network latency
  - Data governance and regulatory compliance
- Cloud data warehouses (e.g., Google BigQuery, Snowflake, Amazon Redshift, MS Azure SQL Data Warehouse)
  - On-demand scalability
  - Cost efficiency
  - Bundled capabilities such as access management and analytics
  - System uptime and availability

# Real-Time/Active DW/BI

- Enabling real-time data updates for real-time analysis and real-time decision making is growing rapidly
  - Push vs. Pull (of data)
- Is real time BI always better?

# Failure Factors in DW Projects

- Lack of executive sponsorship
- Unclear business objectives
- Setting expectations that you cannot meet
- Cultural issues being ignored
  - Change management
- Inappropriate architecture
- Low data quality / missing information
- Loading the data warehouse with information just because it is available
- Choosing a data warehouse manager who is technology oriented rather than user oriented

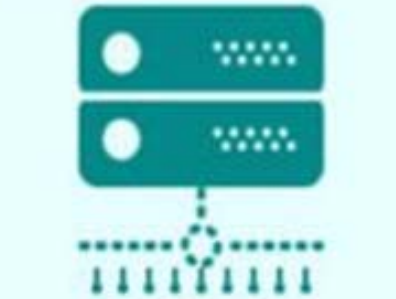
# Data Lake

- Big data are usually stored in a data lake (usually on a cloud based on Hadoop or other distributed storage, such as Azure Data Lake, AWS S3)
- Stores data of all types (both structured and unstructured) that have been generated from different sources.
- Stores data in its rawest form.

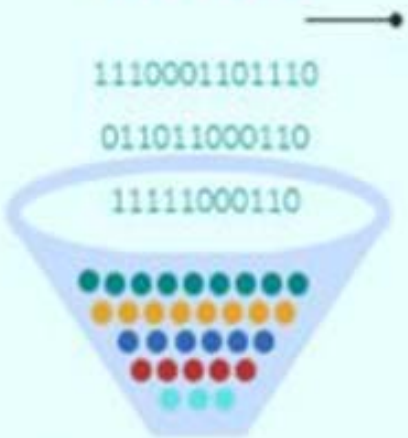
# DATA WAREHOUSE

VS

# DATA LAKE

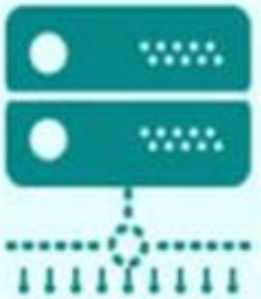


Data is processed and organized into a single schema before being put into the warehouse




The analysis is done on the cleansed data in the warehouse

Raw and unstructured data goes into a data lake




Data is selected and organized as and when needed

# Data Lakehouse