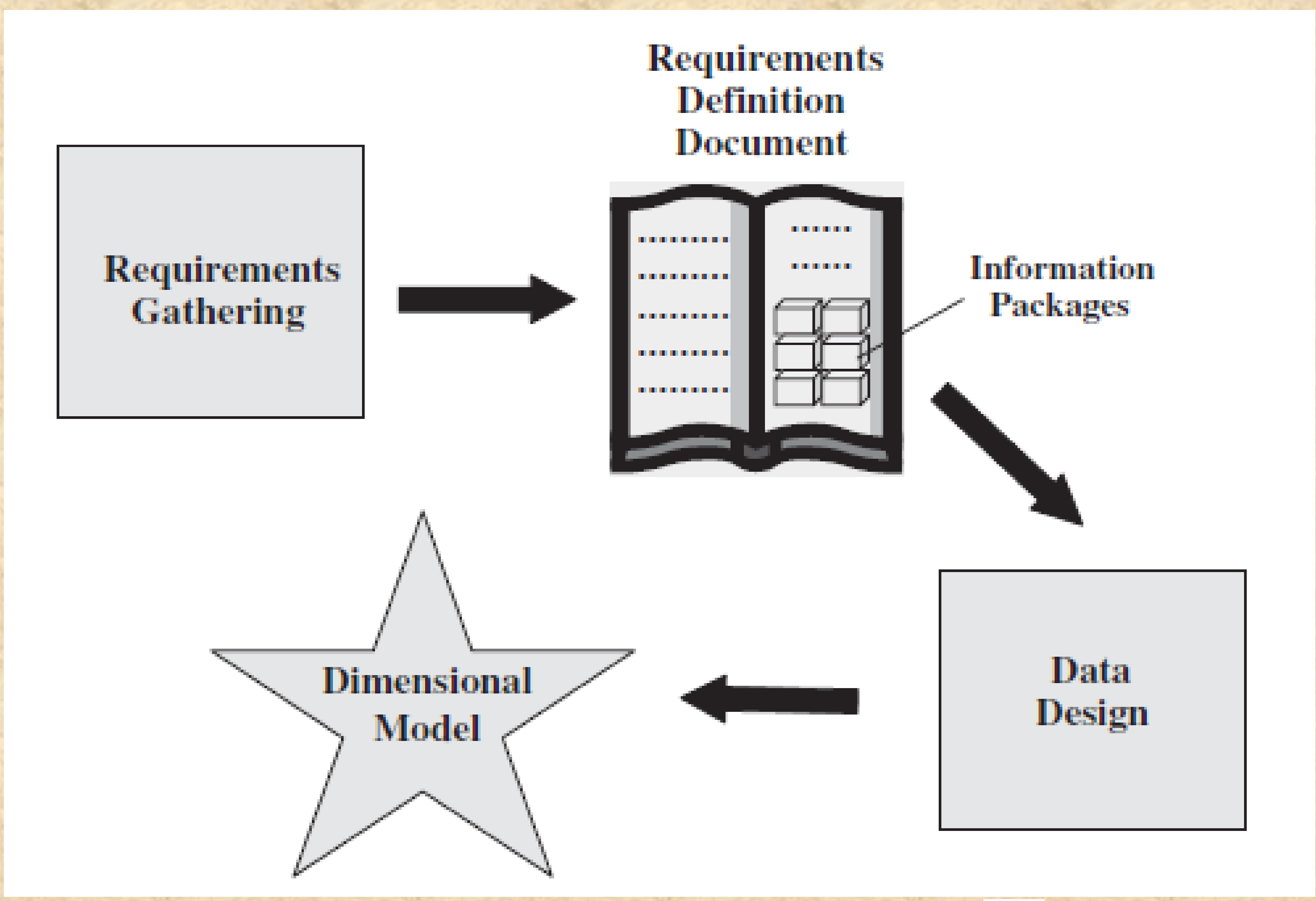
Building the Data Warehouse

Principles of Dimensional Modeling

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

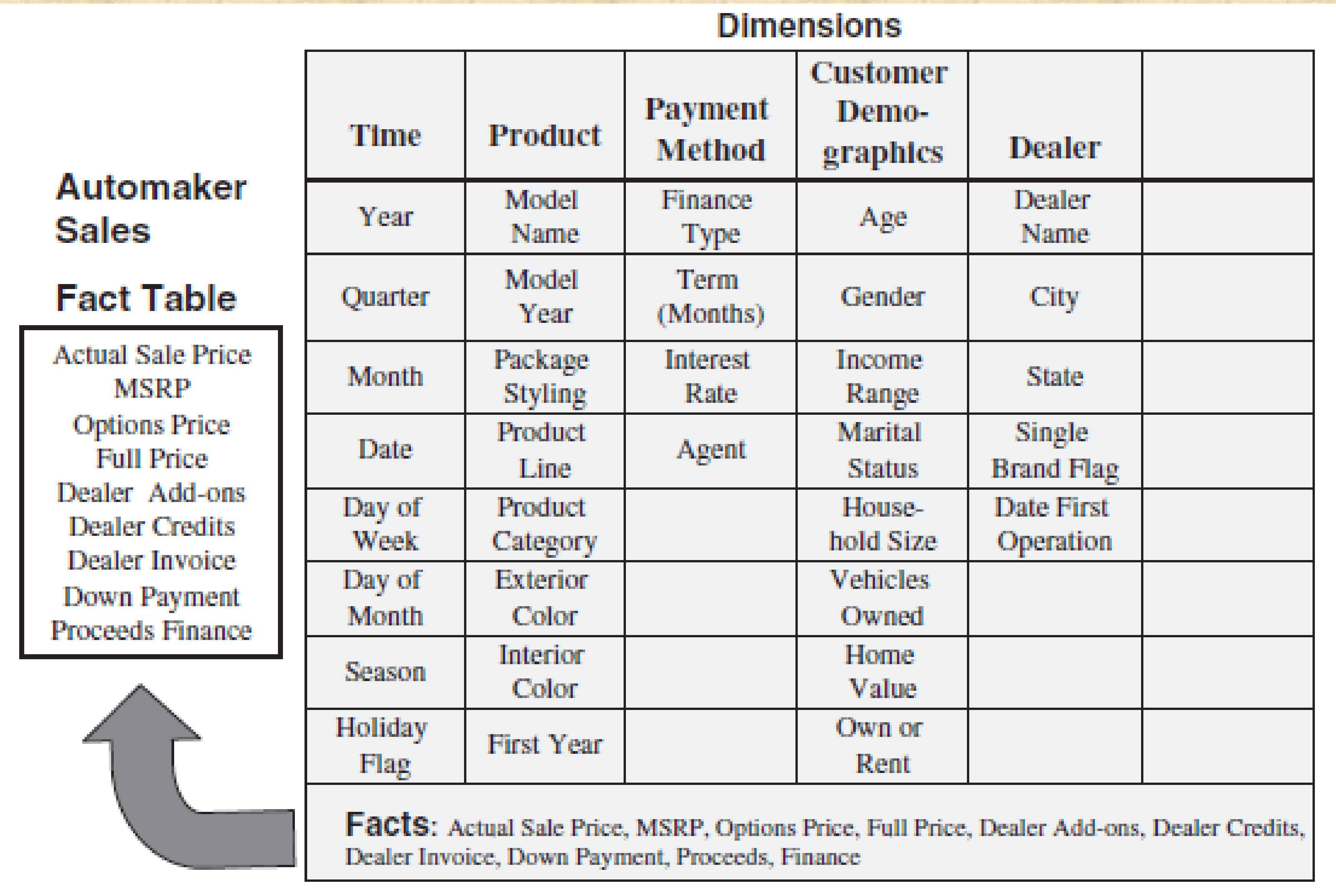
FROM REQUIREMENTS TO DATA DESIGN



Dimensional Modeling Basics

- logical design technique to structure the business dimensions and the metrics
- based on a multidimensional data model which views data in the form of a data cube
- Terminology :
  - **Dimension tables : subjects**
  - **Fact table : units/metrics**

Example



Automaker Sales

Fact Table

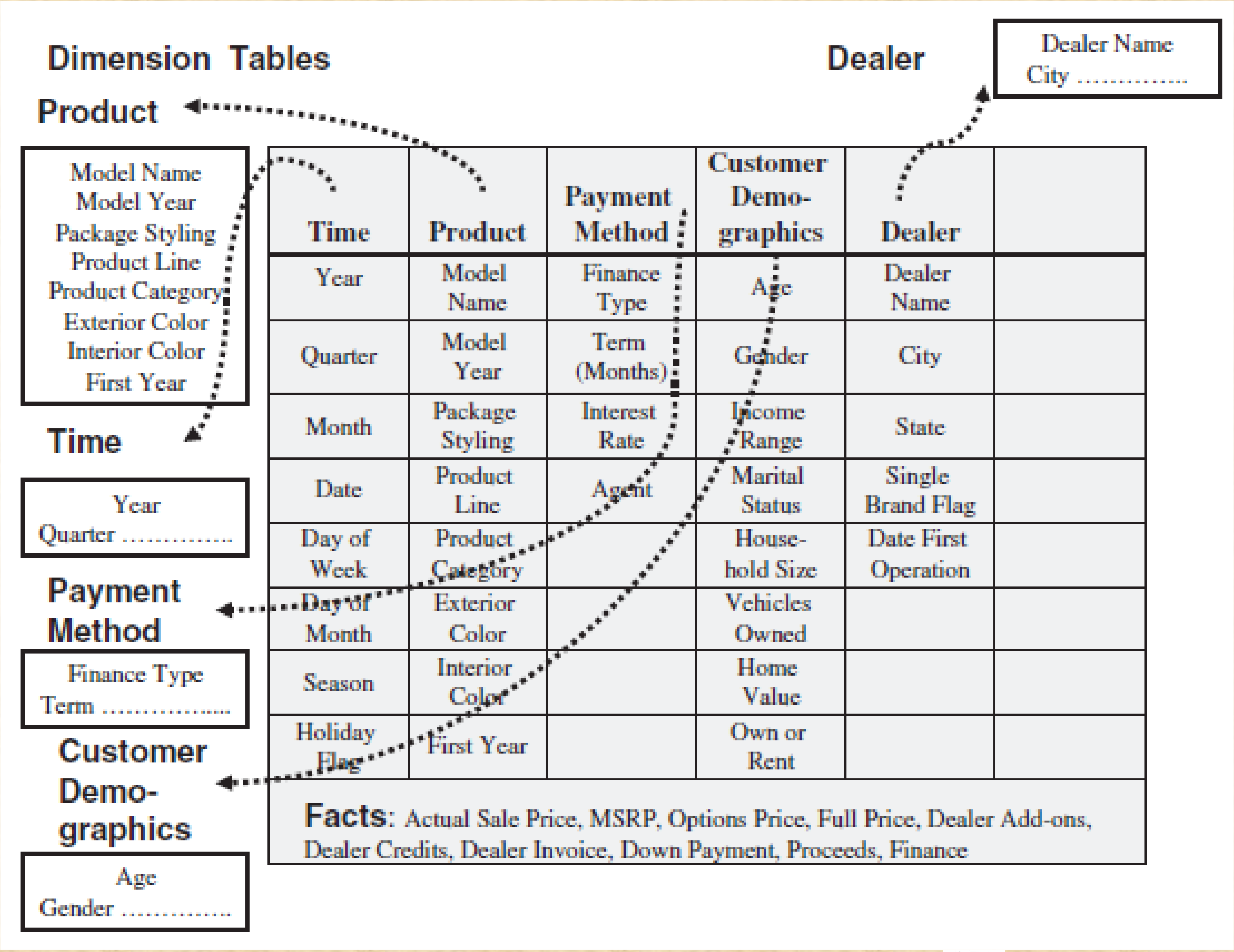
Actual Sale Price
MSRP
Options Price
Full Price
Dealer Add-ons
Dealer Credits
Dealer Invoice
Down Payment
Proceeds Finance

Dimensions

Time	Product	Payment Method	Customer Demo-graphics	Dealer	
Year	Model Name	Finance Type	Age	Dealer Name	
Quarter	Model Year	Term (Months)	Gender	City	
Month	Package Styling	Interest Rate	Income Range	State	
Date	Product Line	Agent	Marital Status	Single Brand Flag	
Day of Week	Product Category		House-hold Size	Date First Operation	
Day of Month	Exterior Color		Vehicles Owned		
Season	Interior Color		Home Value		
Holiday Flag	First Year		Own or Rent		

Facts: Actual Sale Price, MSRP, Options Price, Full Price, Dealer Add-ons, Dealer Credits, Dealer Invoice, Down Payment, Proceeds, Finance

Dimension Tables



Dimension Tables

Dealer

Dealer Name
City ....

Product

Model Name
Model Year
Package Styling
Product Line
Product Category
Exterior Color
Interior Color
First Year

Time

Year
Quarter ....

Payment Method

Finance Type
Term ....

Customer Demo-graphics

Age
Gender ....

Time	Product	Payment Method	Customer Demo-graphics	Dealer	
Year	Model Name	Finance Type	Age	Dealer Name	
Quarter	Model Year	Term (Months)	Gender	City	
Month	Package Styling	Interest Rate	Income Range	State	
Date	Product Line	Agent	Marital Status	Single Brand Flag	
Day of Week	Product Category		House-hold Size	Date First Operation	
Day of Month	Exterior Color		Vehicles Owned		
Season	Interior Color		Home Value		
Holiday Flag	First Year		Own or Rent		

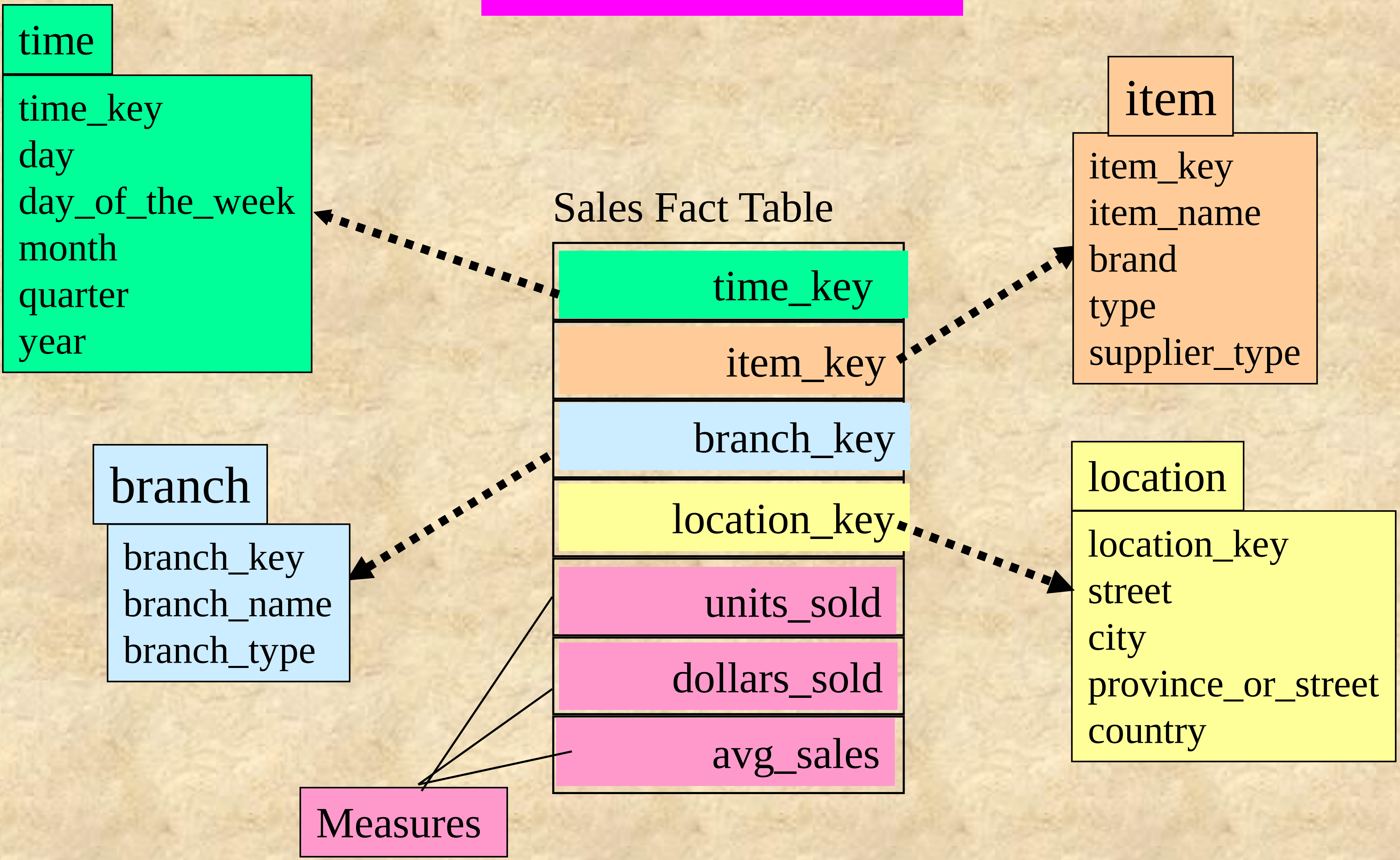
Facts: Actual Sale Price, MSRP, Options Price, Full Price, Dealer Add-ons, Dealer Credits, Dealer Invoice, Down Payment, Proceeds, Finance

Conceptual Modeling of Data Warehouses

- Modeling data warehouses: dimensions & measures
  - **Star schema:** A fact table in the middle connected to a set of dimension tables
  - **Snowflake schema:** A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake
  - **Fact constellations:** Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation

Example of Star Schema

Sales Data Mart



time

time_key
day
day_of_the_week
month
quarter
year

Sales Fact Table

time_key
item_key
branch_key
location_key
units_sold
dollars_sold
avg_sales

item

item_key
item_name
brand
type
supplier_type

branch

branch_key
branch_name
branch_type

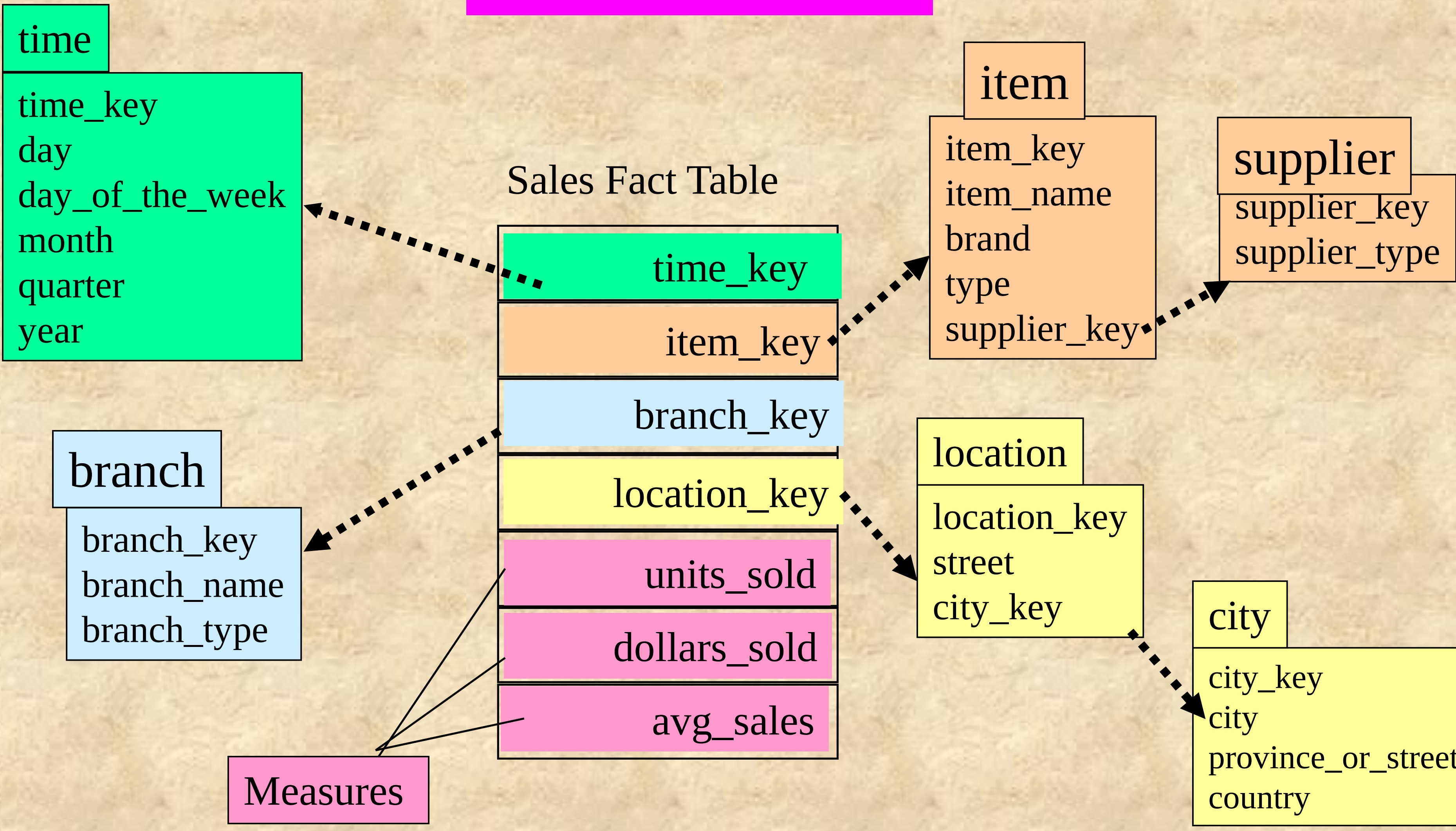
location

location_key
street
city
province_or_street
country

Measures

Example of Snowflake Schema

Sales Data Mart



time

time_key
day
day_of_the_week
month
quarter
year

Sales Fact Table

time_key
item_key
branch_key
location_key
units_sold
dollars_sold
avg_sales

Measures

item

item_key
item_name
brand
type
supplier_key

supplier

supplier_key
supplier_type

branch

branch_key
branch_name
branch_type

location

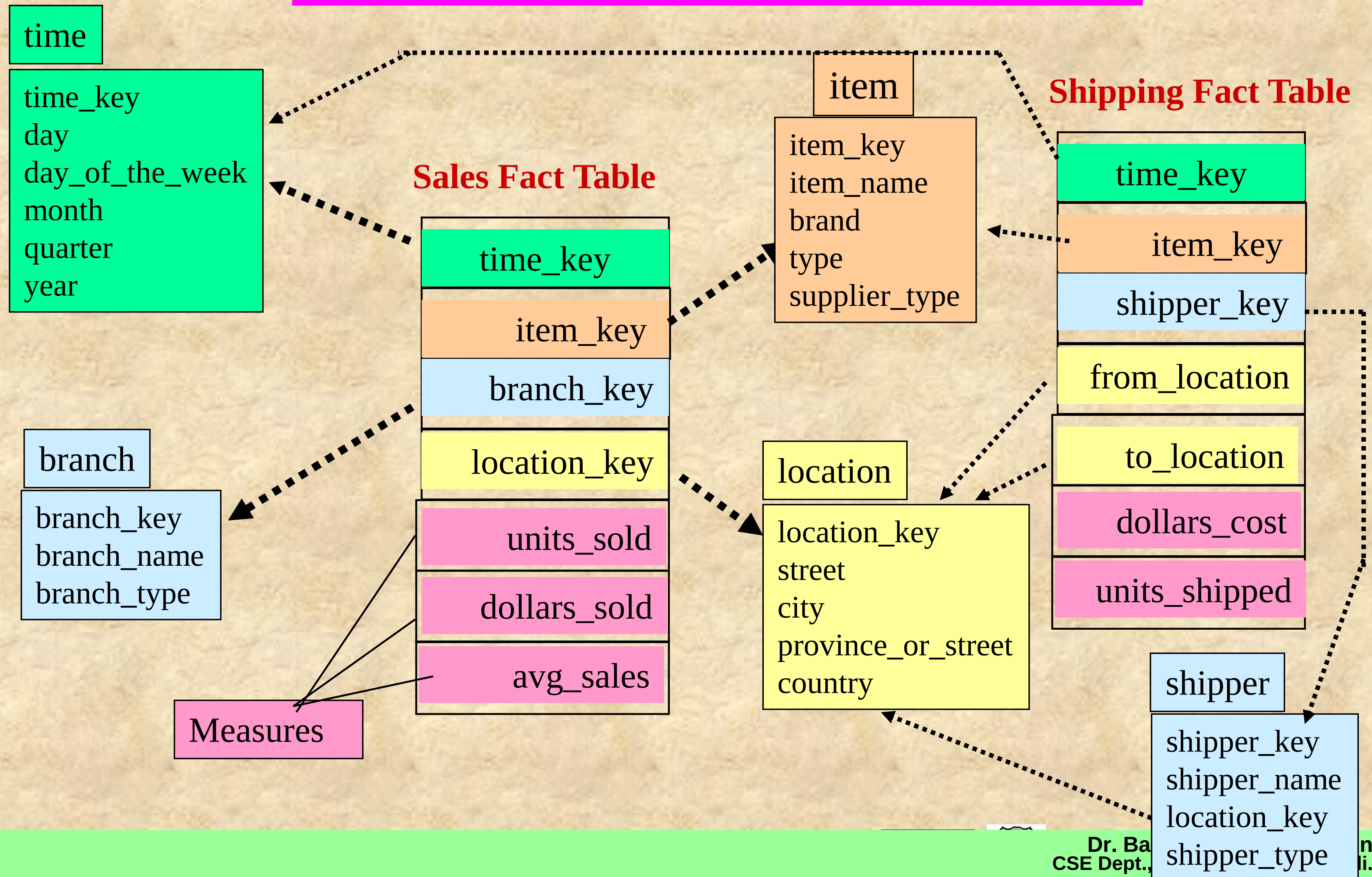
location_key
street
city_key

city

city_key
city
province_or_street
country

Example of Fact Constellation

Sales (with Shipping) Data Mart



time

time_key
day
day_of_the_week
month
quarter
year

Sales Fact Table

time_key
item_key
branch_key
location_key
units_sold
dollars_sold
avg_sales

Measures

branch

branch_key
branch_name
branch_type

item

item_key
item_name
brand
type
supplier_type

location

location_key
street
city
province_or_street
country

Shipping Fact Table

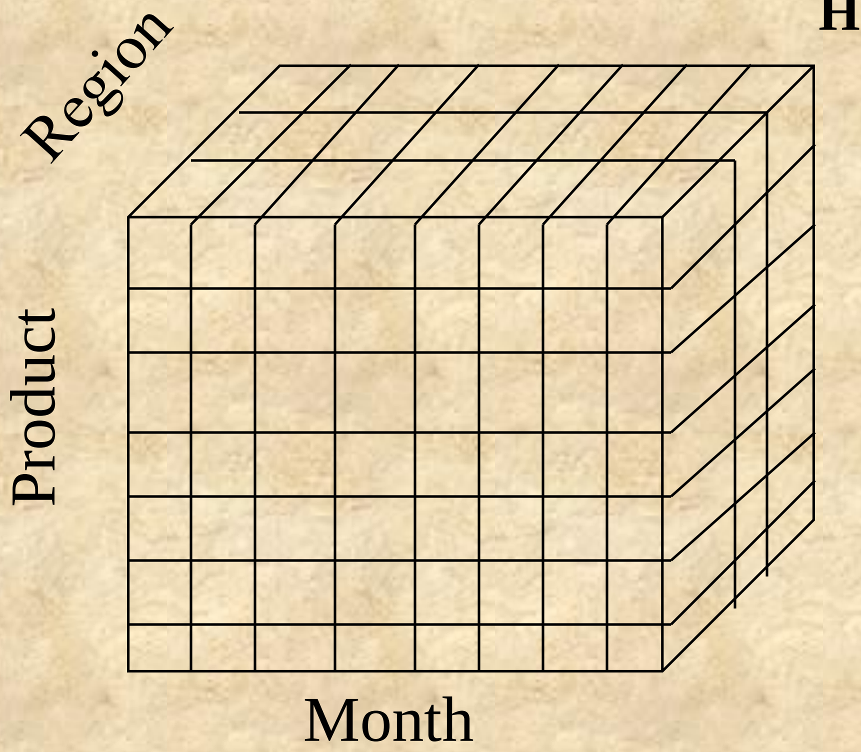
time_key
item_key
shipper_key
from_location
to_location
dollars_cost
units_shipped

shipper

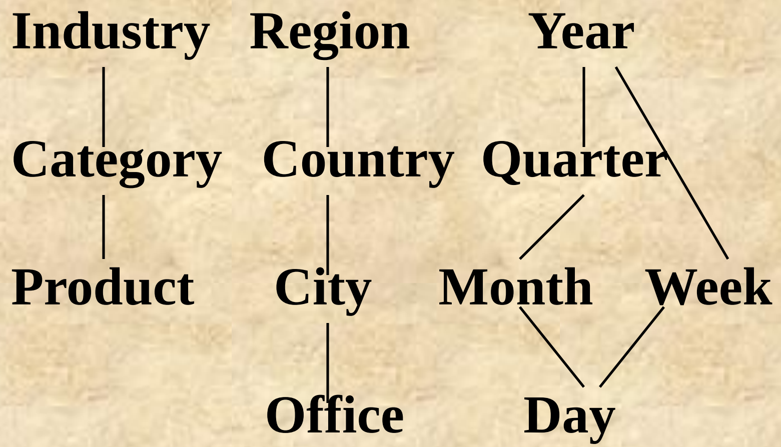
shipper_key
shipper_name
location_key
shipper_type

Multidimensional Data : Cube

- Sales volume as a function of product, month, and region



Dimensions: Product, Location, Time
Hierarchical summarization paths



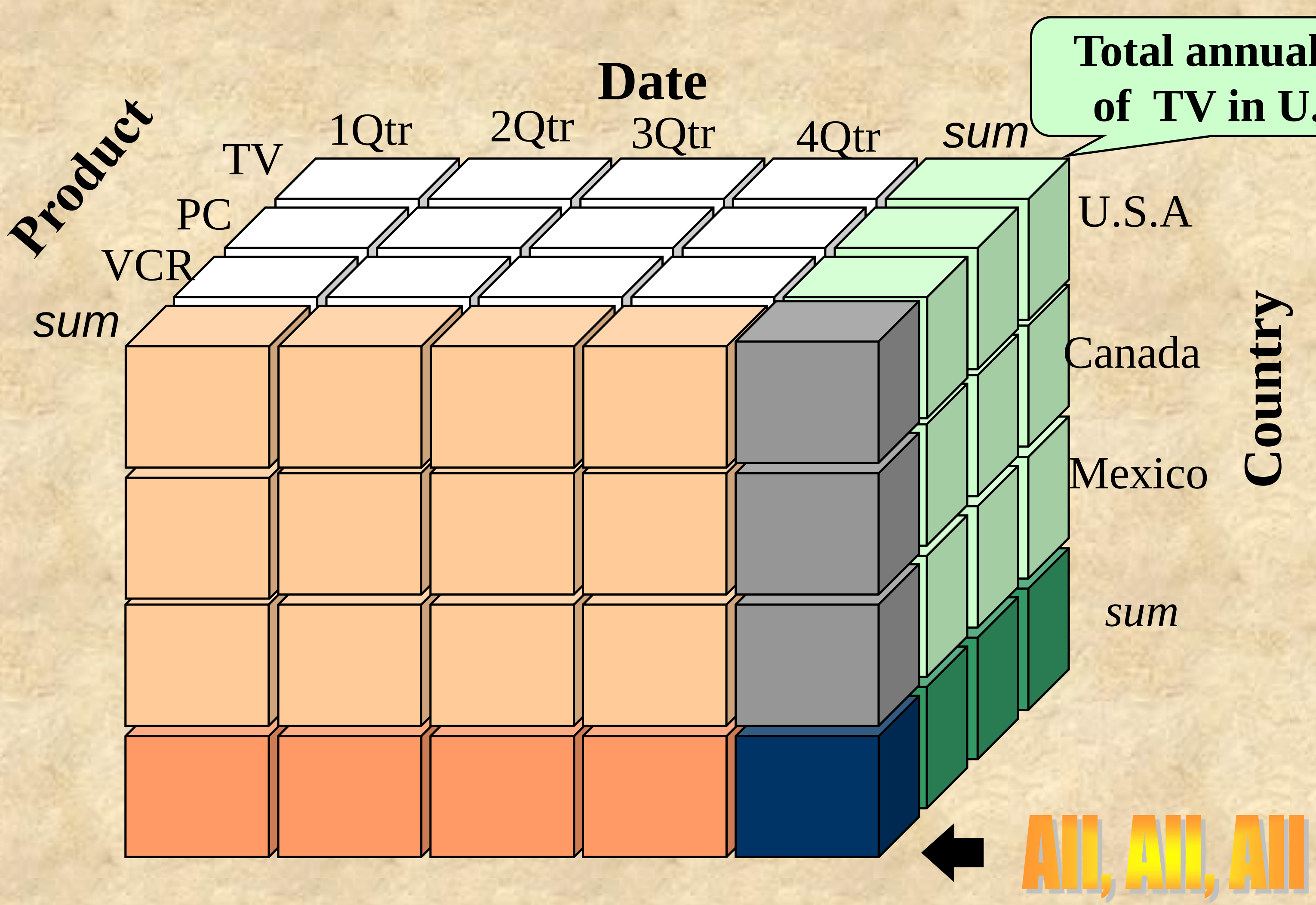
Industry Region Year

Category Country Quarter

Product City Month Week

Office Day

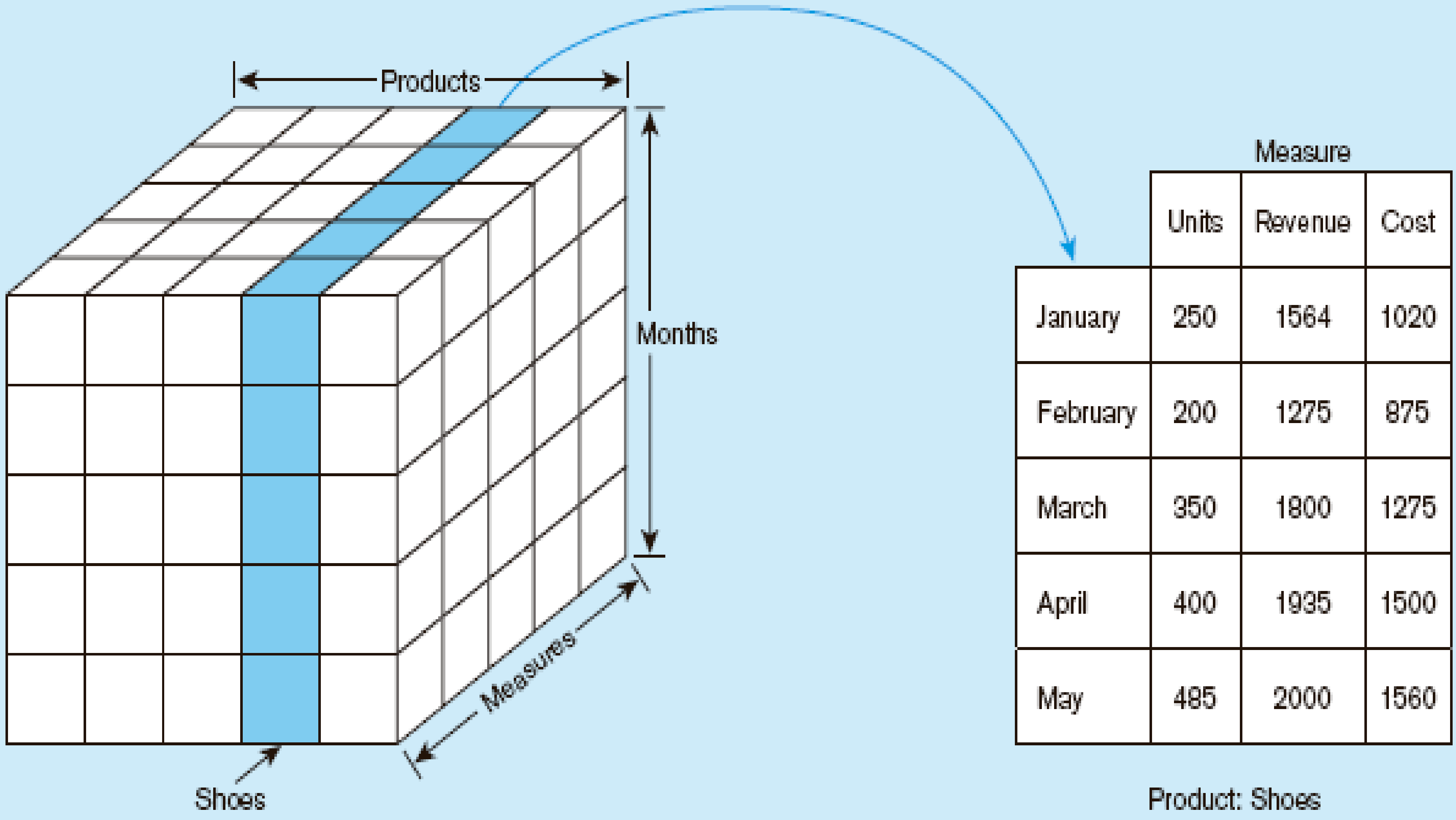
A Sample Data Cube



Typical Cube Operations

- **Roll up (drill-up):** summarize data
  - *by climbing up hierarchy or by dimension reduction*
- **Drill down (roll down):** reverse of roll-up
  - *from higher level summary to lower level summary or detailed data, or introducing new dimensions*
- **Slice and dice:**
  - *project and select*
- **Pivot (rotate):**
  - *reorient the cube, visualization, 3D to series of 2D planes.*
- **Other operations**
  - *drill across: involving (across) more than one fact table*
  - *drill through: through the bottom level of the cube to its back-end relational tables (using SQL)*

Example : Slicing a data cube



	Measure		
	Units	Revenue	Cost
January	250	1564	1020
February	200	1275	875
March	350	1800	1275
April	400	1935	1500
May	485	2000	1560

Product: Shoes

Exercise 2.1

Design a STAR schema for a retail company to track the sales units and sales dollars with three dimension tables.

Building the Data Warehouse

Data Extraction, Transformation & Loading [ETL]

Data Warehouse Back-End Tools and Utilities

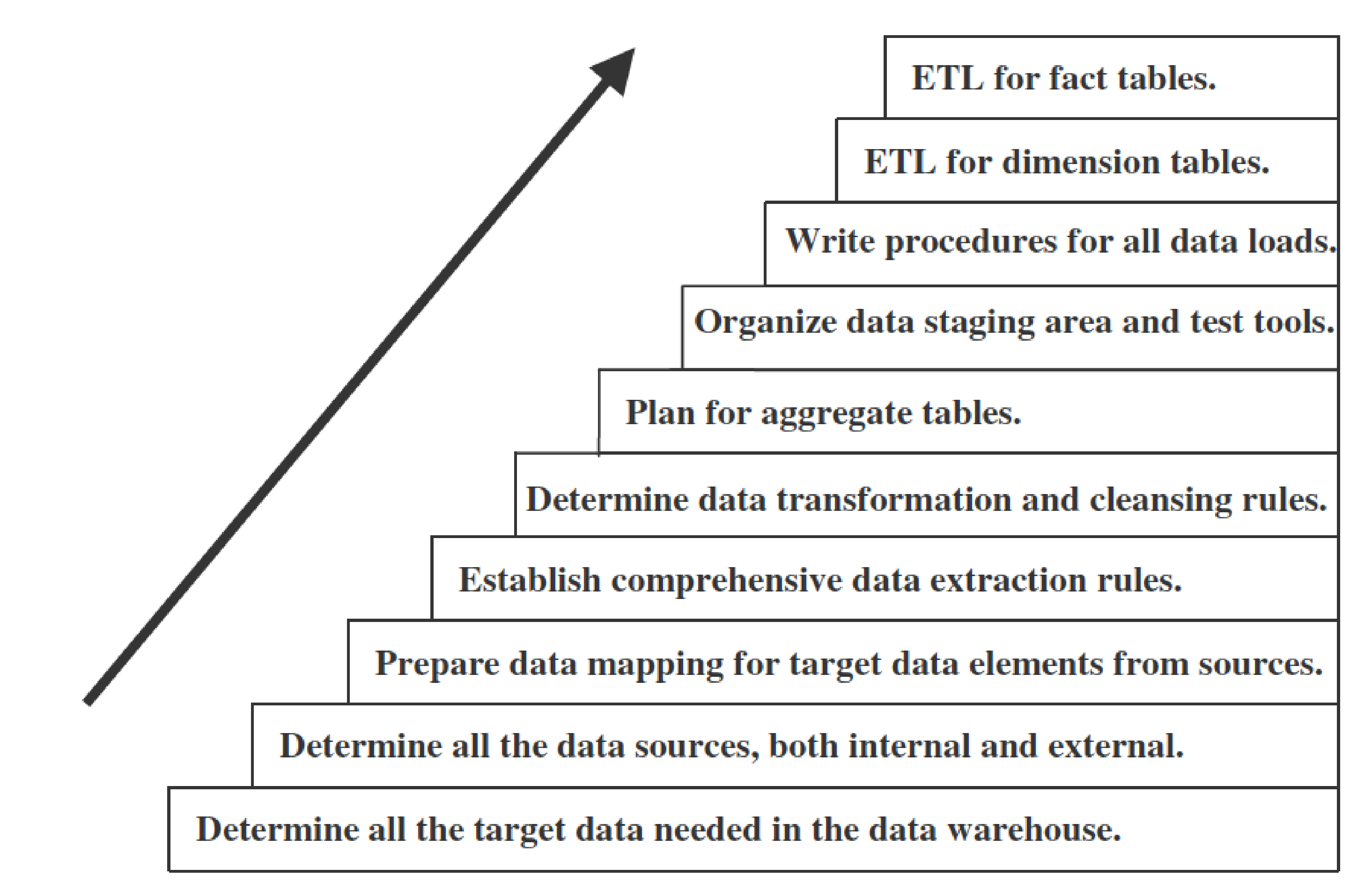
- **Data extraction:**
  - get data from multiple, heterogeneous, and external sources
- **Data cleaning:**
  - detect errors in the data and rectify them when possible
- **Data transformation:**
  - convert data from legacy or host format to warehouse format
- **Load:**
  - sort, summarize, consolidate, compute views, check integrity, and build indicies and partitions
- **Refresh :**
  - propagate the updates from the data sources to the warehouse

Extract, Transform, Load (ETL)

- Extract only relevant data from the internal source systems or external systems, instead of dumping all data (“data junkhouse”)
- The ETL completion can take up to 50-70% of your total effort while developing a data warehouse.
- These ETL efforts depends on various factors

Certified for IBM DB2 e-business software

Major steps in ETL



ETL for fact tables.

ETL for dimension tables.

Write procedures for all data loads.

Organize data staging area and test tools.

Plan for aggregate tables.

Determine data transformation and cleansing rules.

Establish comprehensive data extraction rules.

Prepare data mapping for target data elements from sources.

Determine all the data sources, both internal and external.

Determine all the target data needed in the data warehouse.

Data Extraction

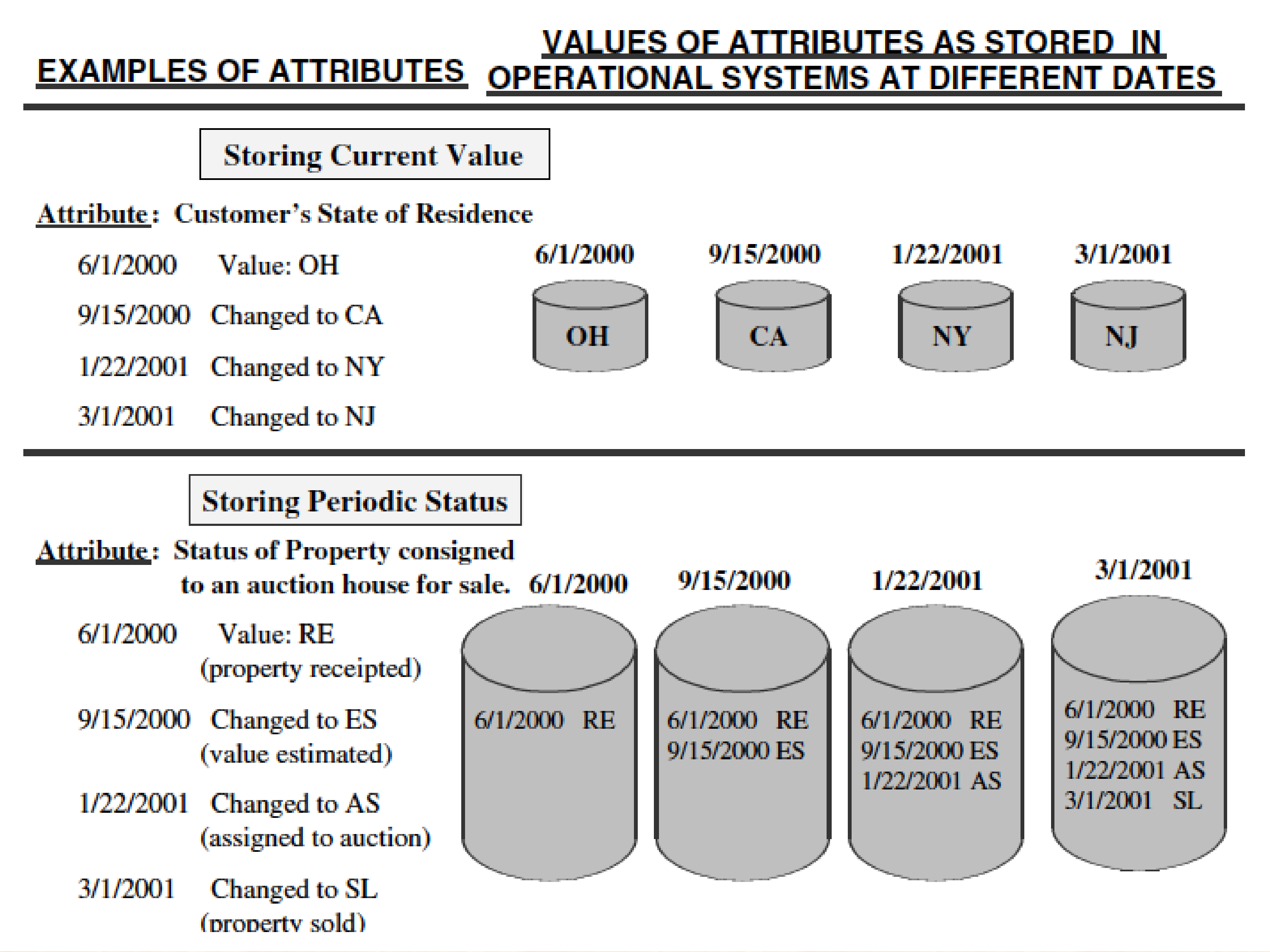
- Data can be extracted using third party tools or in-house programs or scripts
- Data extraction issues:

1. Identify sources
2. Method of extraction for each source (manual, automated)
3. When and how much frequently data will be extracted for each source
4. Time window
5. Sequencing of extraction processes

How data is stored in operational systems

- Current value: Values continue to changes as daily transactions are performed. We need to monitor these changes to maintain history for decision making process, e.g., bank balance, customer address, etc.
- Periodic status: sometimes the history of changes is maintained in the source system

Example



EXAMPLES OF ATTRIBUTES

VALUES OF ATTRIBUTES AS STORED IN OPERATIONAL SYSTEMS AT DIFFERENT DATES

Storing Current Value

Attribute: Customer's State of Residence

6/1/2000 Value: OH

9/15/2000 Changed to CA

1/22/2001 Changed to NY

3/1/2001 Changed to NJ

Storing Periodic Status

Attribute: Status of Property consigned to an auction house for sale.

6/1/2000 Value: RE (property receipted)

9/15/2000 Changed to ES (value estimated)

1/22/2001 Changed to AS (assigned to auction)

3/1/2001 Changed to SL (property sold)

Data Extraction Method

- **Static data extraction:**

1. Extract the data at a certain time point.
2. It will include all transient data and periodic data along with its time/date status at the extraction time point
3. Used for initial data loading

- **Data of revisions**

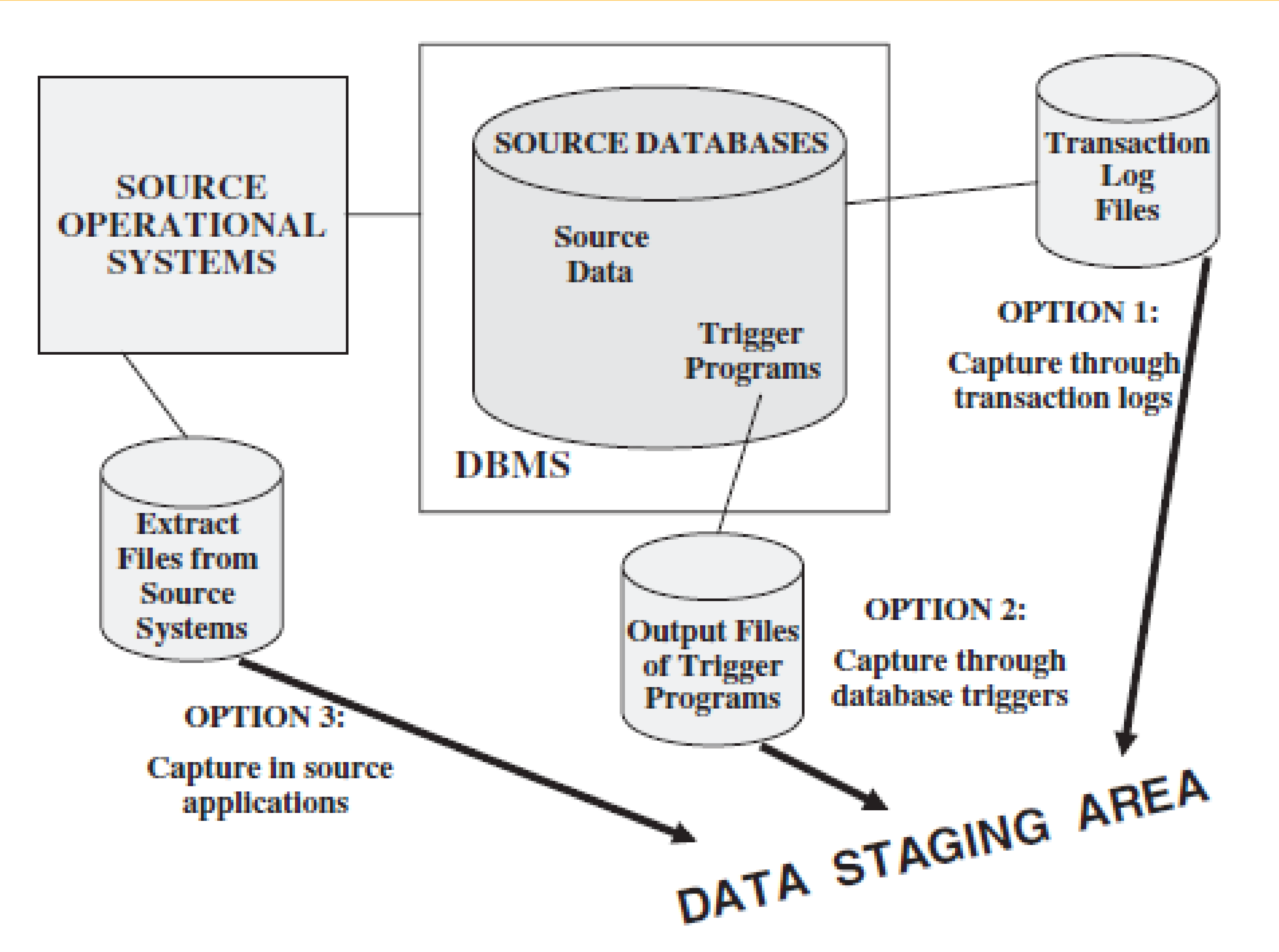
1. Data is loaded in increments thus preserving history of both changing and periodic data

Incremental data extraction

- **Immediate data extraction: involves data extraction in real time.**
- **Possible options:**

1. Capture through transactions logs
2. Make triggers/Stored procedures
3. Capture via source application
4. Capture on the basis of time and date stamps
5. Capture by comparing files

Options for Immediate Extraction


SOURCE OPERATIONAL SYSTEMS
SOURCE DATABASES
Source Data
Trigger Programs
DBMS
Transaction Log Files
OPTION 1:
Capture through transaction logs
Extract Files from Source Systems
Output Files of Trigger Programs
OPTION 2:
Capture through database triggers
OPTION 3:
Capture in source applications
DATA STAGING AREA

Data Transformation

- **Transformation means to integrate or consolidate data from various sources**
- Major tasks:

1. Format conversions (change in data type, length)
2. Decoding of fields (1,0 → male, female)
3. Calculated and derived values (units sold, price, cost→ profit)
4. Splitting of single fields (House No 11, ABC Road, Sangli, Maharashtra State, INDIA)
5. Merging of information (information from different sources regarding any entity, attribute)
6. Character set conversion

Data Transformation (Cont.)

8. Conversion of unit of measures
9. Date/time conversion
10. Key restructuring
11. De-duplication
12. Entity identification
13. Multiple source problem

Data Loading

- **Determine when (time) and how (as a whole or in chunks) to load data**
- **Four modes to load data**

1. **Load**: removes old data if available otherwise load data
2. **Append**: The old data is not removed, the new data is appended with the old data
3. **Destructive Merge**: If primary key of the new record matched with the primary key of old record then update old record
4. **Constructive Merge**: If primary key of the new record matched with the primary key of old record then do not update old record just add the new record and mark it as superseding record

Refresh / Update

After the initial load, the data warehouse is kept up-to-date by

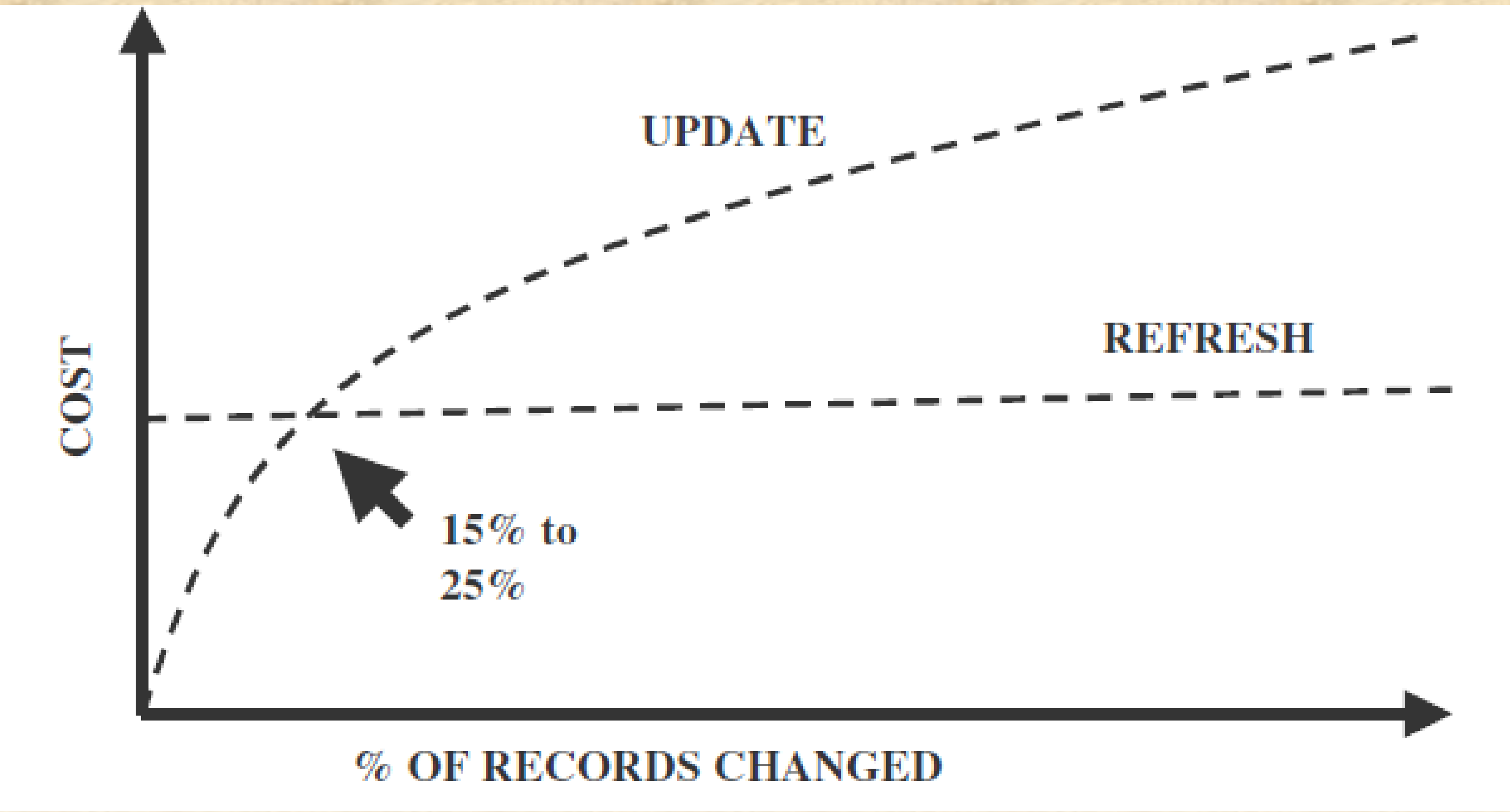
REFRESH - complete reload at specified intervals

UPDATE - application of incremental changes

Data Loading (Cont.)

- **Data Refresh Vs. Data Update**

Full refresh reloads whole data after deleting old data and data updates are used to update the changing attributes



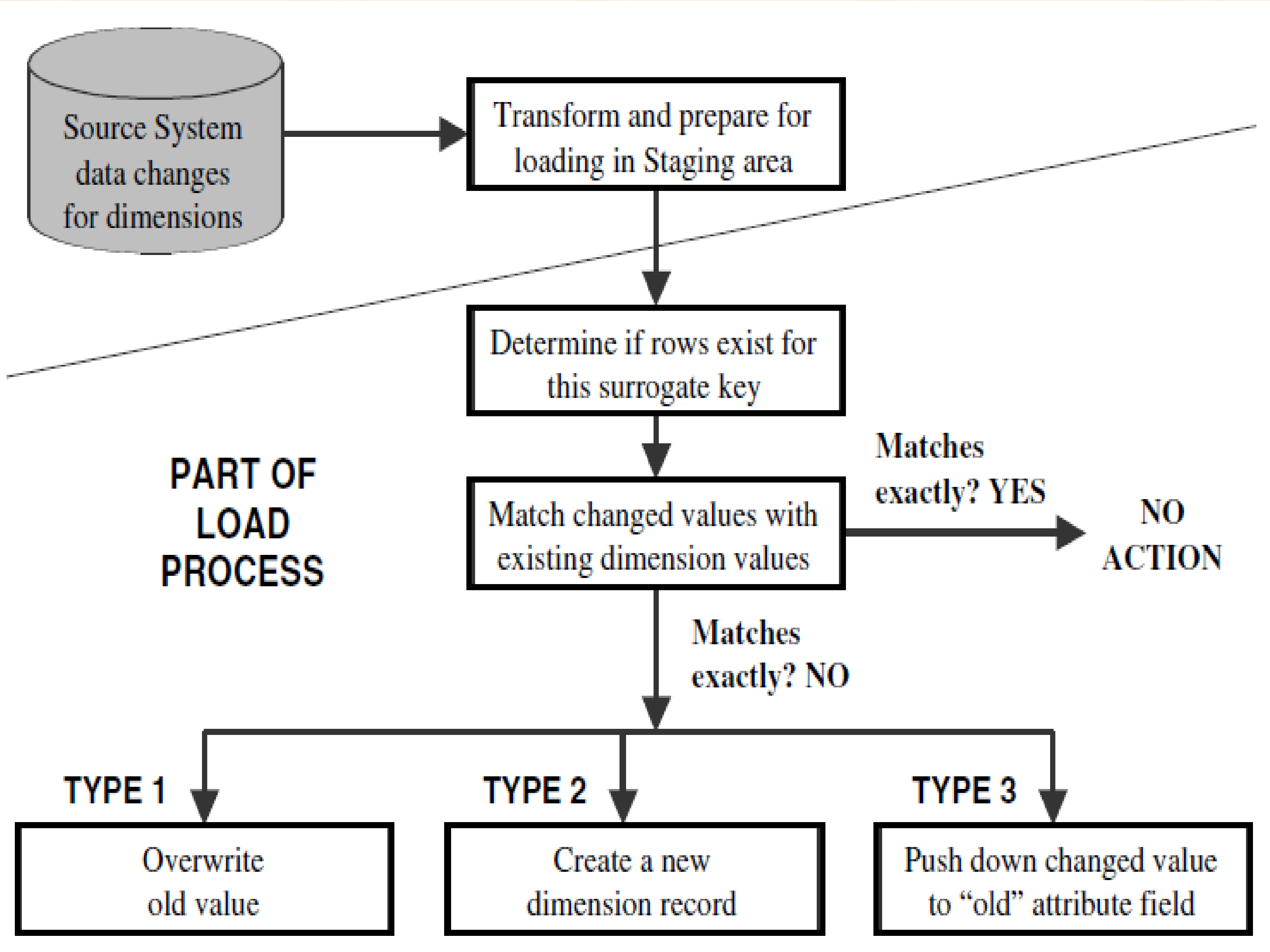
Data Loading (Cont.)

- **Loading for dimensional tables**:

You need to define a mapping between source system key and system generated key in data warehouse, otherwise you will not be able to load/update data correctly

Data Loading (Cont.)

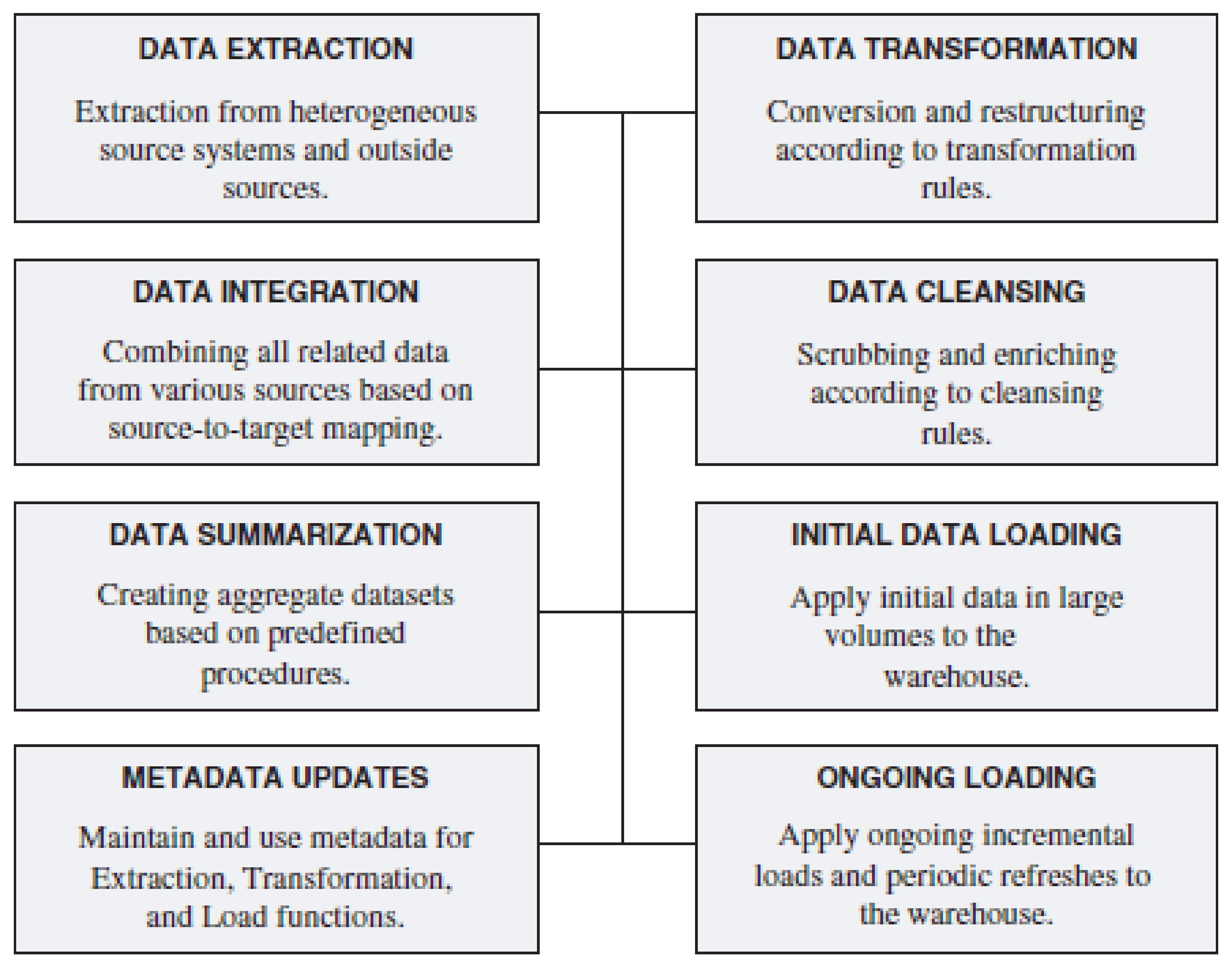
- **Updates to dimension table**


Source System data changes for dimensions
Transform and prepare for loading in Staging area
Determine if rows exist for this surrogate key
PART OF LOAD PROCESS
Match changed values with existing dimension values
Matches exactly? YES
NO ACTION
Matches exactly? NO
TYPE 1
TYPE 2
TYPE 3
Overwrite old value
Create a new dimension record
Push down changed value to "old" attribute field

Loading Fact Table

- Concatenation of the keys of dimensional table
- Load dimension records first
- Create concatenated key for the fact table record from the keys of the corresponding dimension record
- History load :
  - Loads historical data useful and interesting
- Incremental load :
  - Load as frequently as possible
- Use partitioned files/indexes, parallel processing

ETL Summary



DATA EXTRACTION

Extraction from heterogeneous source systems and outside sources.

DATA TRANSFORMATION

Conversion and restructuring according to transformation rules.

DATA INTEGRATION

Combining all related data from various sources based on source-to-target mapping.

DATA CLEANSING

Scrubbing and enriching according to cleansing rules.

DATA SUMMARIZATION

Creating aggregate datasets based on predefined procedures.

INITIAL DATA LOADING

Apply initial data in large volumes to the warehouse.

METADATA UPDATES

Maintain and use metadata for Extraction, Transformation, and Load functions.

ONGOING LOADING

Apply ongoing incremental loads and periodic refreshes to the warehouse.