Getting data into the data warehouse

Data Warehouse involves following functions

- Data extraction
- Transformation
- Cleansing the data
- summarization
- Loading data

Data Extraction

- It is necessary to extract the relevant data from the operational database before bringing it into the data warehouse.
- Many commercial tools are available to help with the extraction process.

[Paulraj pg.no. 37 pg. no. 281]

Transformation

- Inconsistency is an indicator of poor data quality.
- In the case of multiple input sources to a data warehouse, inconsistency can sometimes make data unusable.
- Transformation is the process of dealing with these inconsistencies.
- Once the data elements have the right names, they must be converted to common formats.
- All the transformations can be automated.
- Basic tasks:
  - Selection : extract the whole record or portions then select
  - Splitting/joining : splitting or joining part of data from diff sources.
  - Conversion: standardization & fields useful and understandable to the user
  - Summarization : storing aggregates only
  - Enrichment : rearrangement or simplification of individual fields

[Paulraj pg.no. 296]

Cleansing

- Information quality is a key consideration in determining the value of that information.
- Information of high quality leads to high quality decisions.
- Data entered into the data warehouse must be error-free.
- This process is known as data cleansing.
- These include missing data & incorrect data in one source; inconsistent data & conflicting data when two or more sources are involved
- Many type of data cleansing can be automated.
- Data cleansing software can often suggest areas to check for poor data quality even if can't figure out how to fix them.

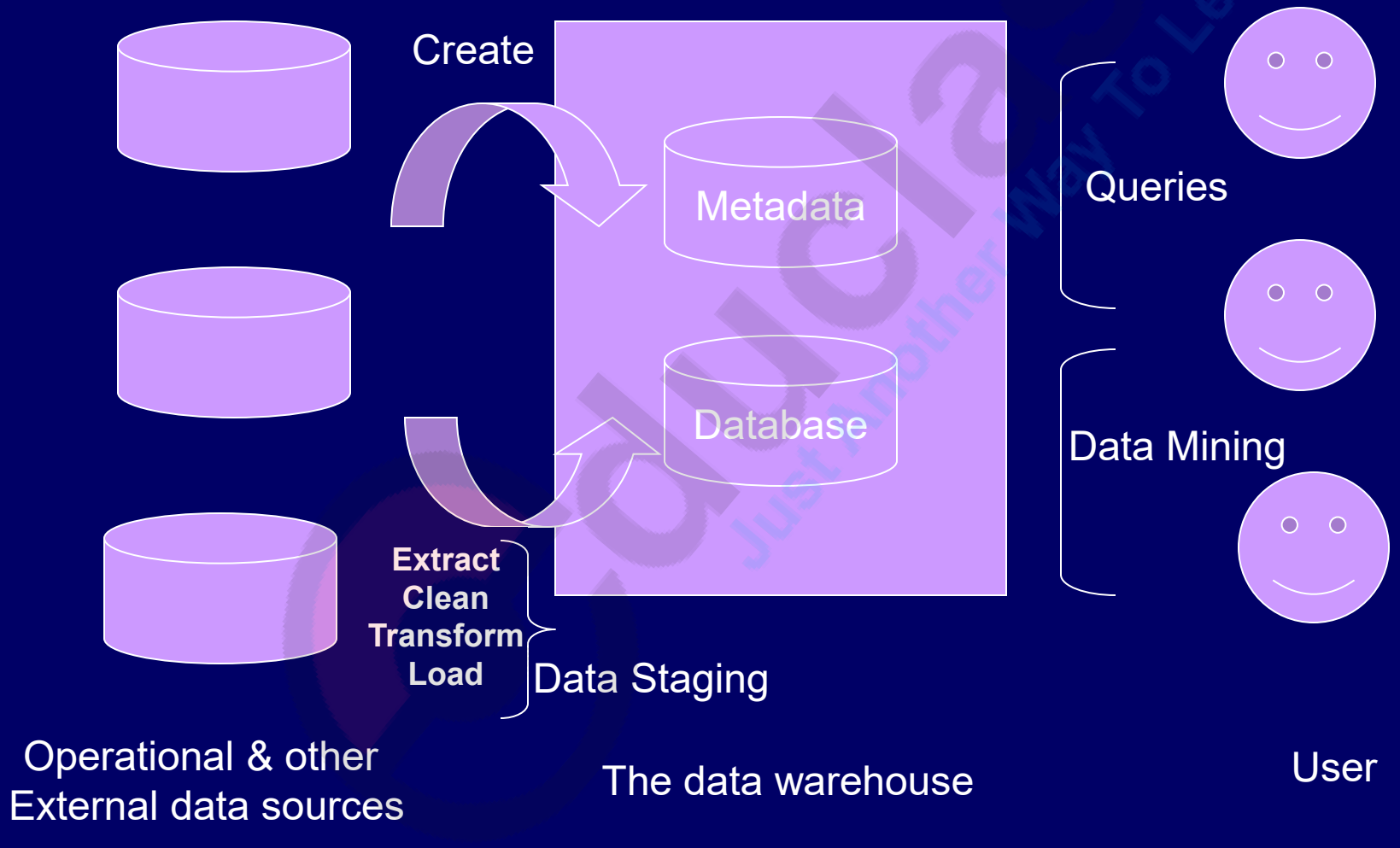
Loading

- Once these steps have been successfully performed, it is finally possible to load the new data into the data warehouse database.
- Loading implies physical movement of the data from the computer(s) storing the source database(s) to that which will store the data warehouse database.
- The most common channel for the data movement process is a high-speed communication link.

Summarization

- Summary data is one of the levels of data in a data warehouse.
- Once the data warehouse database has been loaded it is possible to create these summaries.
- Summaries must usually be re-created after every incremental update,as any changes in the underlying data may impact them.

Data Warehouse Architecture



Data Warehouse Architecture

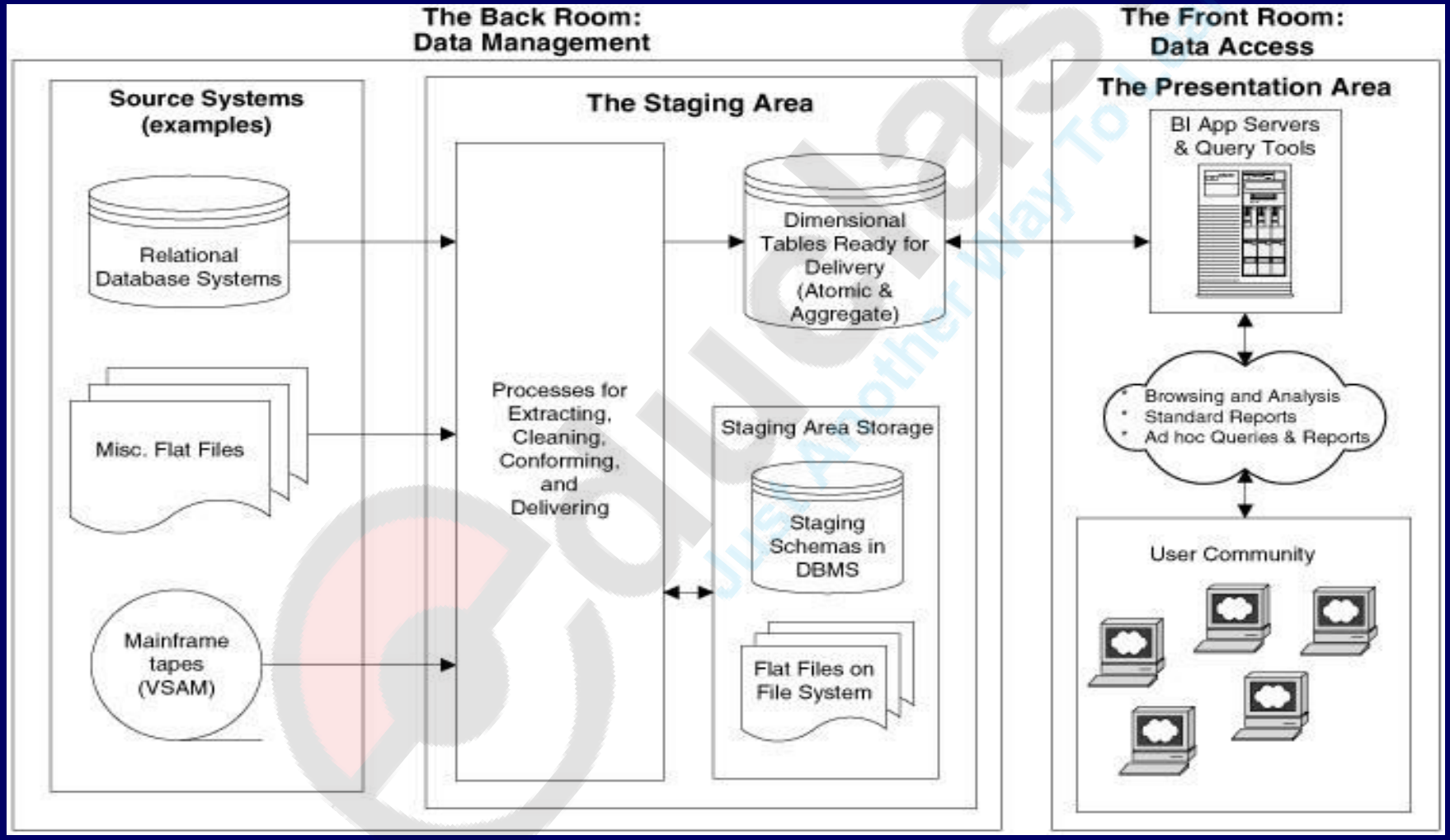
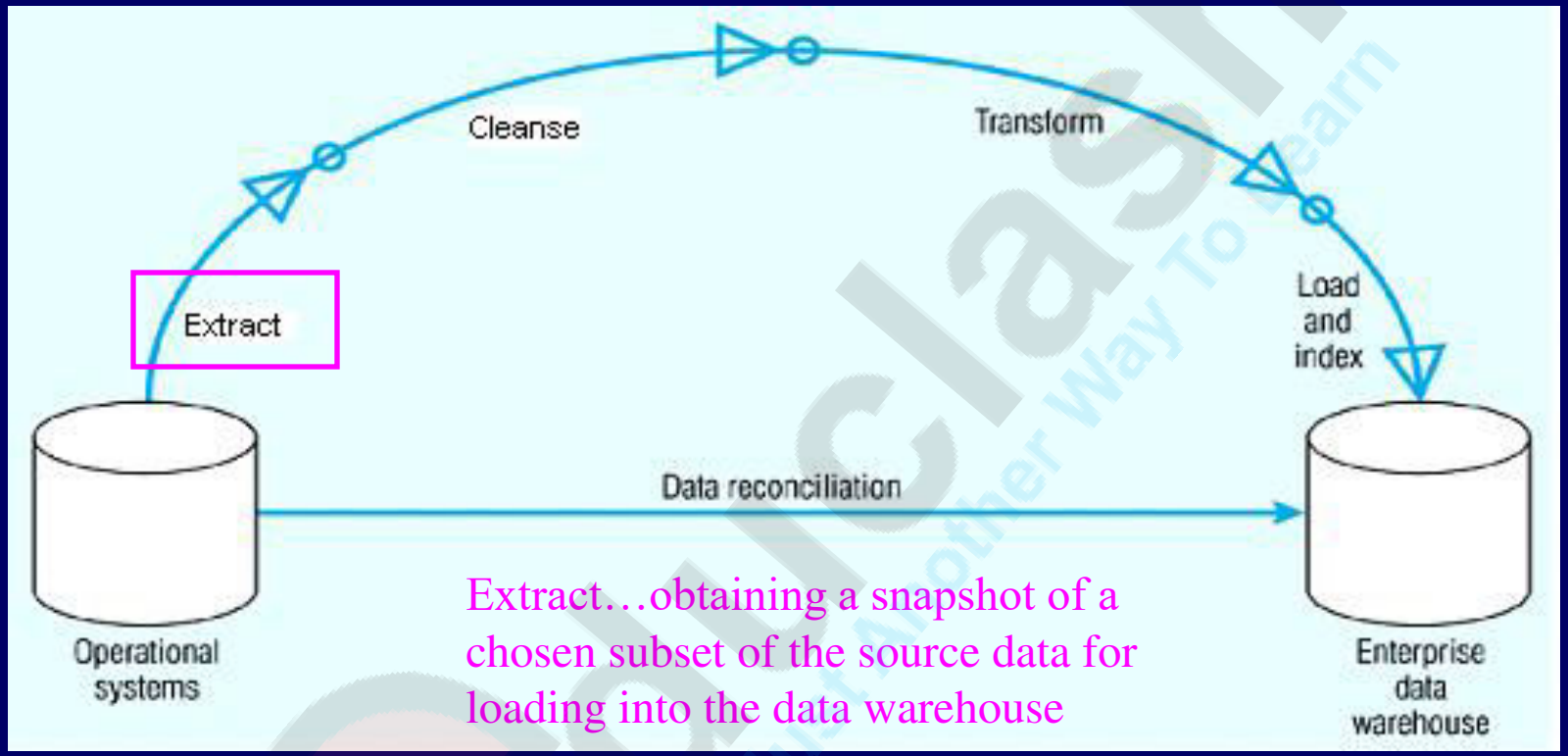


Figure 11-10: Steps in data reconciliation



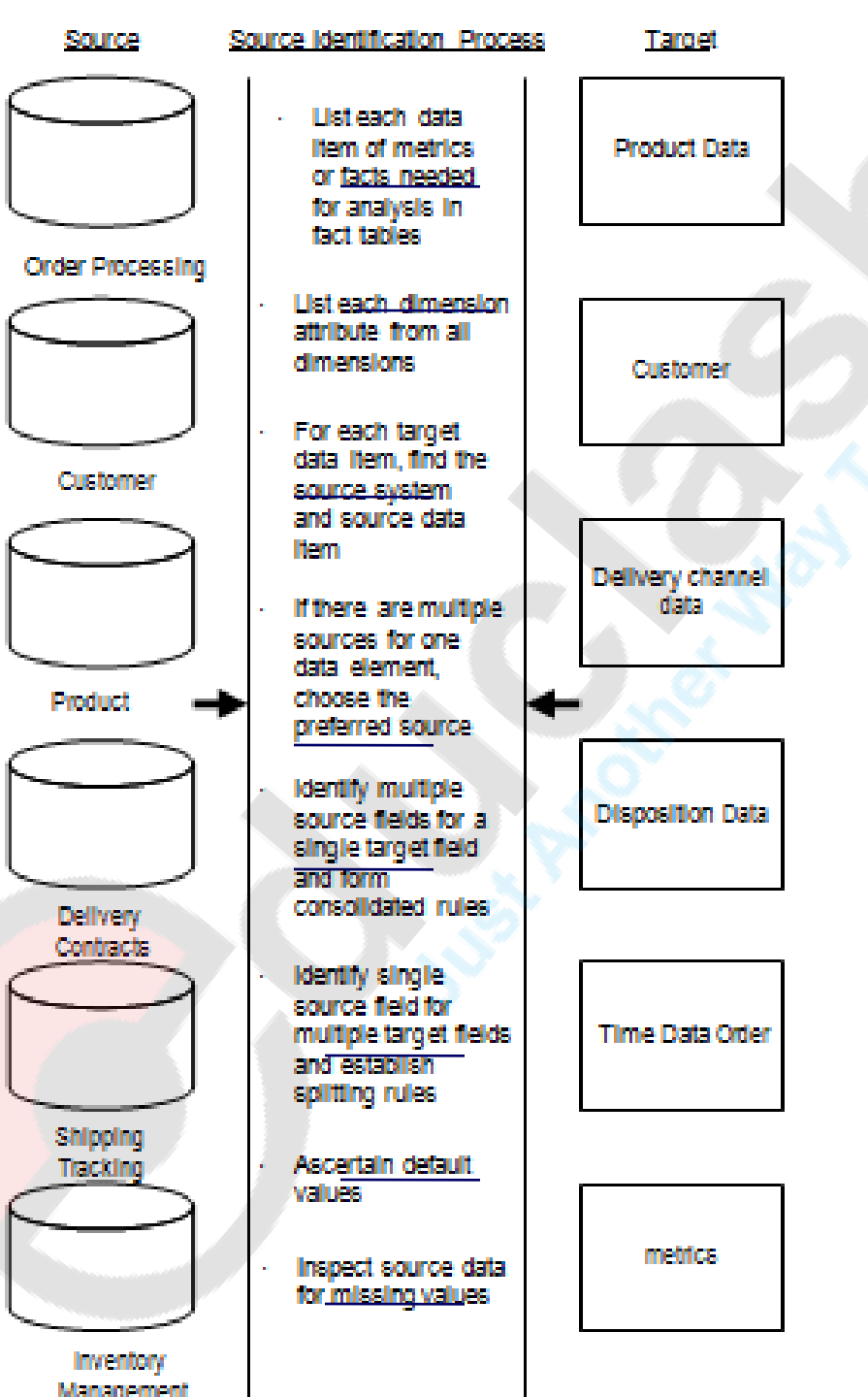
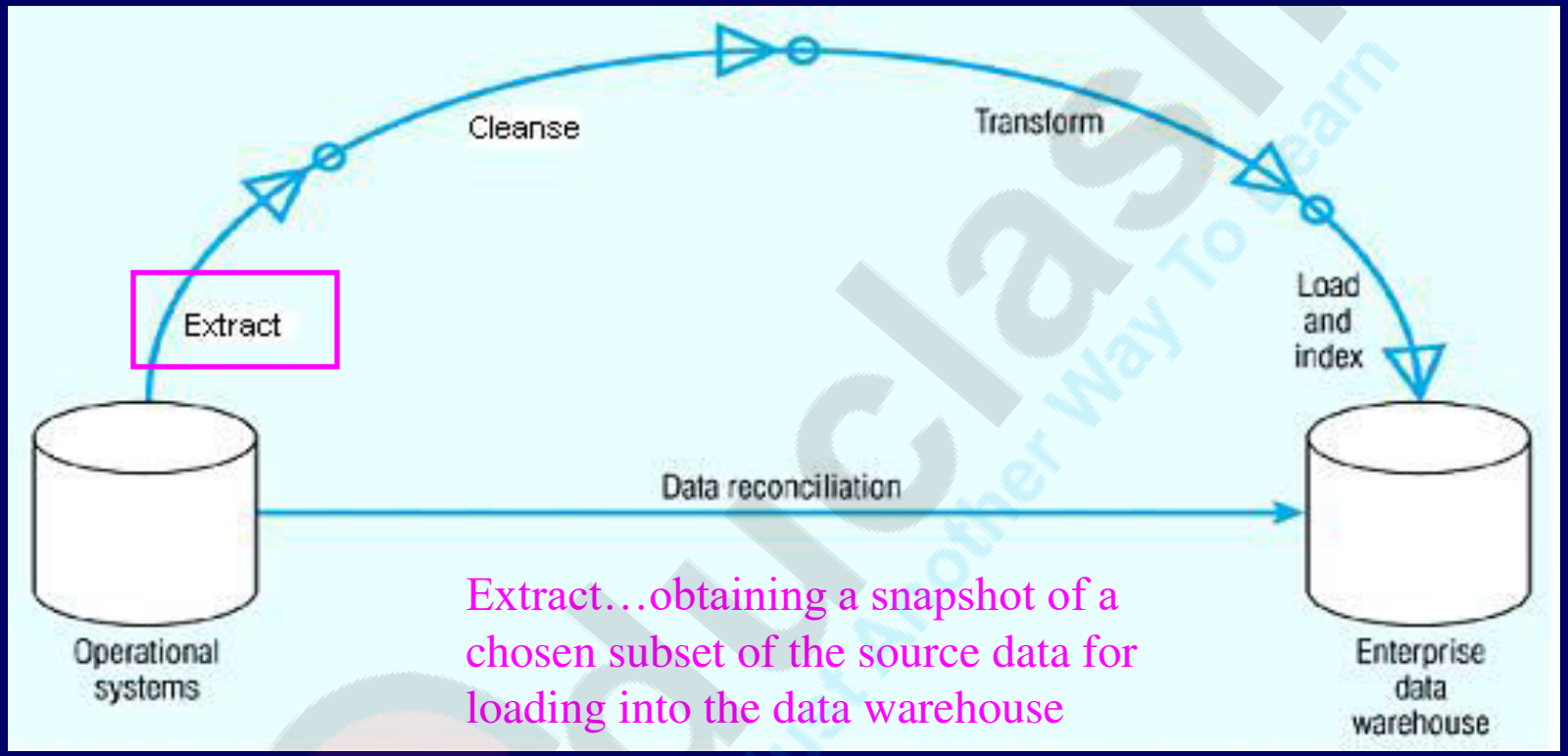


Figure 11-10: Steps in data reconciliation

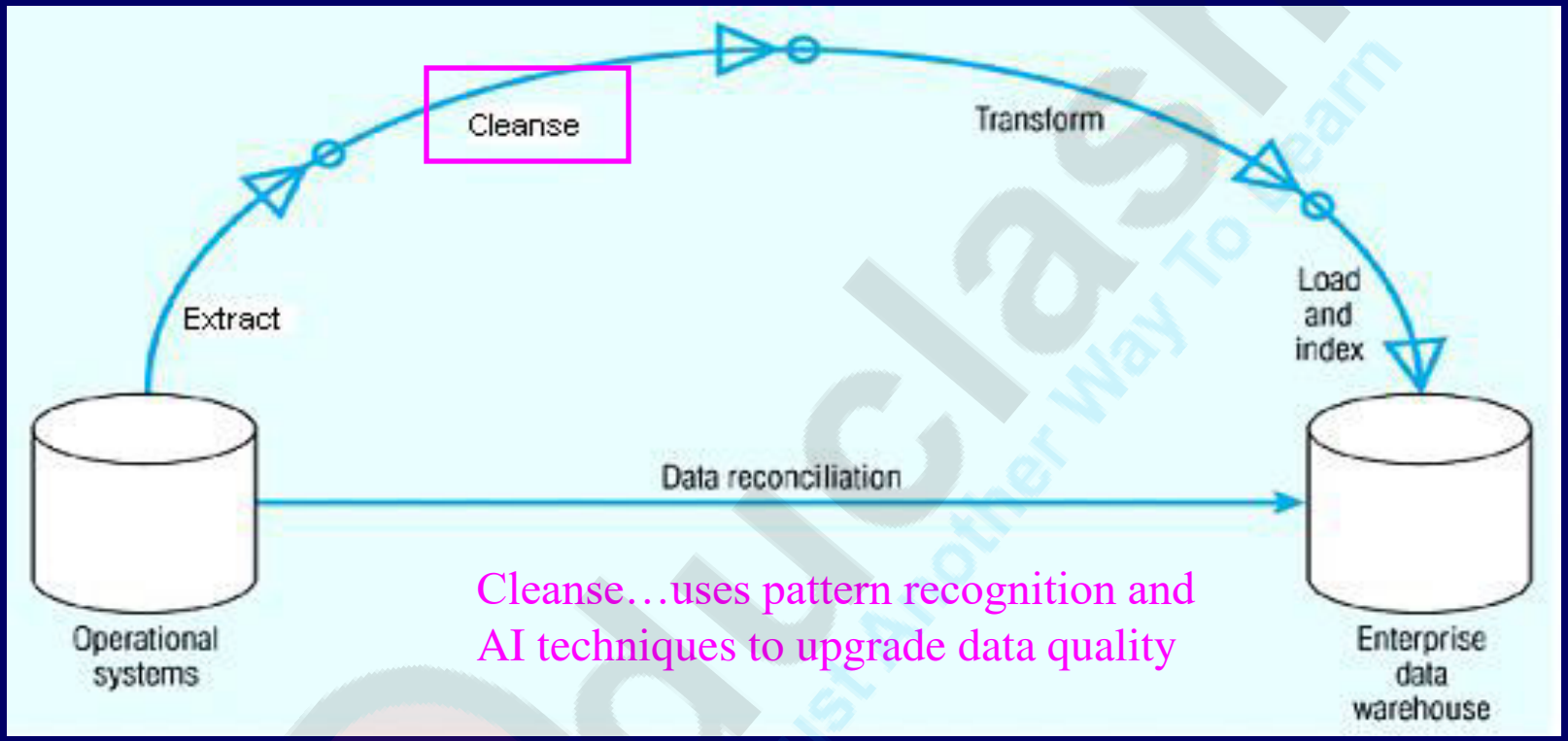


Static extract = capturing a snapshot of the source data at a point in time

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

- **Extraction Process :**
  - Immediate data extraction : real time
    - Capture through transaction logs
    - Capture through database triggers
    - Capture in source applications
  - Deferred data extraction : do not capture changes in real time
    - Capture based on date and time stamp
    - Capture by comparing files by comparing two snapshots

Figure 11-10: Steps in data reconciliation (continued)



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

Figure 11-10: Steps in data reconciliation (continued)

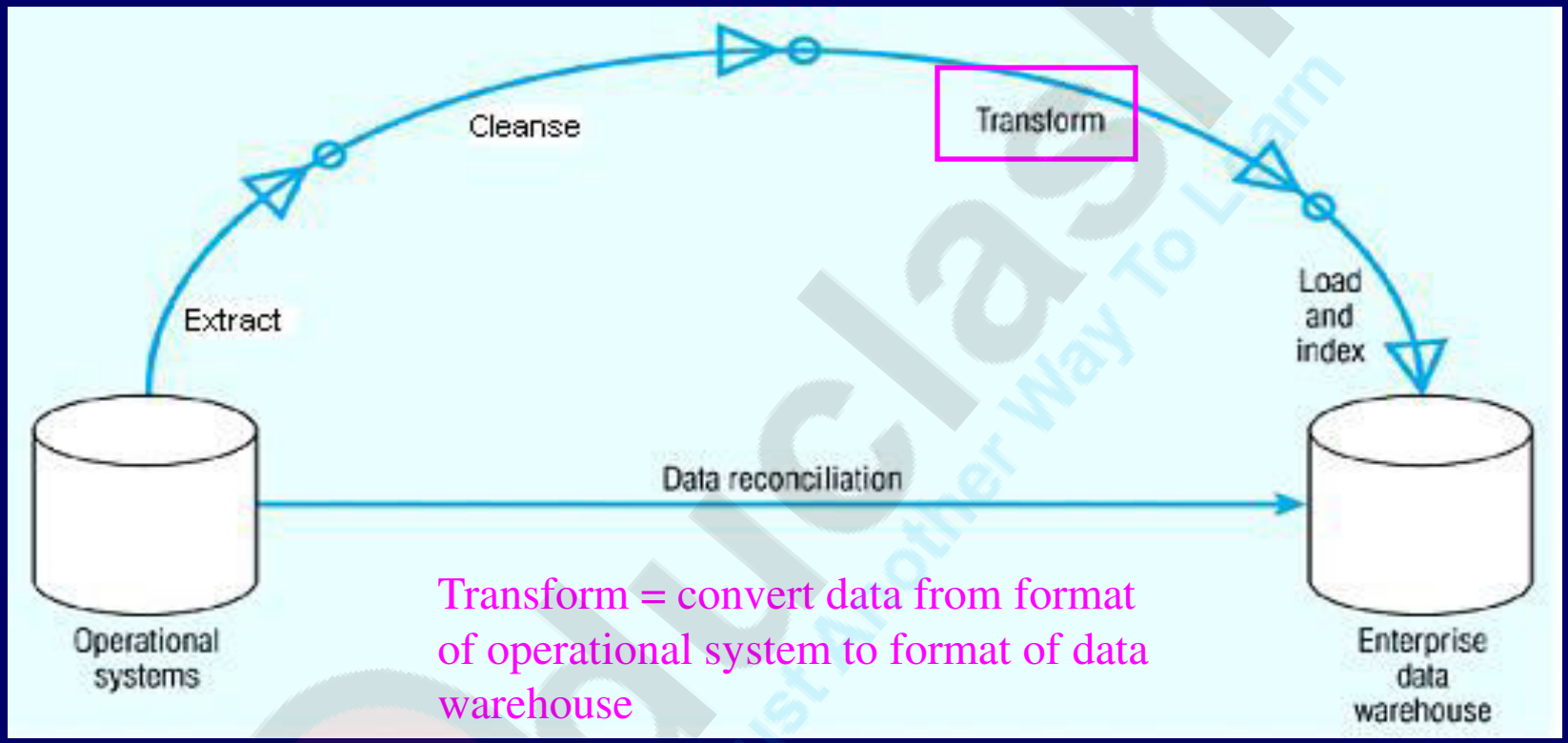
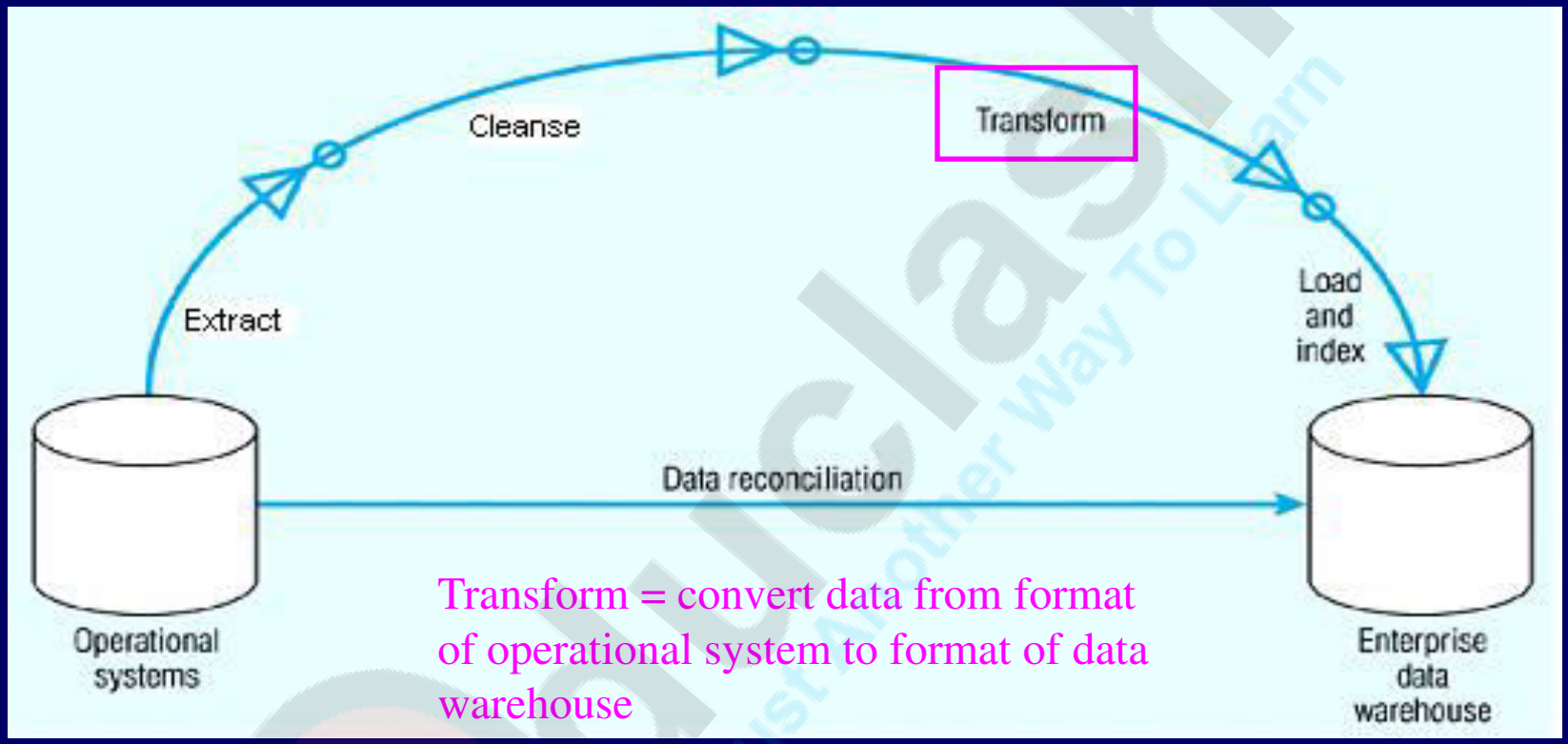


Figure 11-10: Steps in data reconciliation (continued)



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

Transformation

- Basic tasks:
  - Selection : extract the whole record or portions then select
  - Splitting/joining : splitting or joining part of data from diff sources.
  - Conversion: standardization & fields useful and understandable to the user
  - Summarization : storing aggregates only
  - Enrichment : rearrangement or simplification of individual fields

[Paulraj pg.no. 296]

Transformation

- Major transformation tasks:
  - Format revision
  - Decoding of fields
  - Calculated and derived values
  - Splitting
  - Merging
  - Character set conversion
  - Units
  - Date /time
  - Summarization
  - Key restructuring
  - Deduplication

[Paulraj pg.no. 298]

Figure 11-11: Single-field transformation

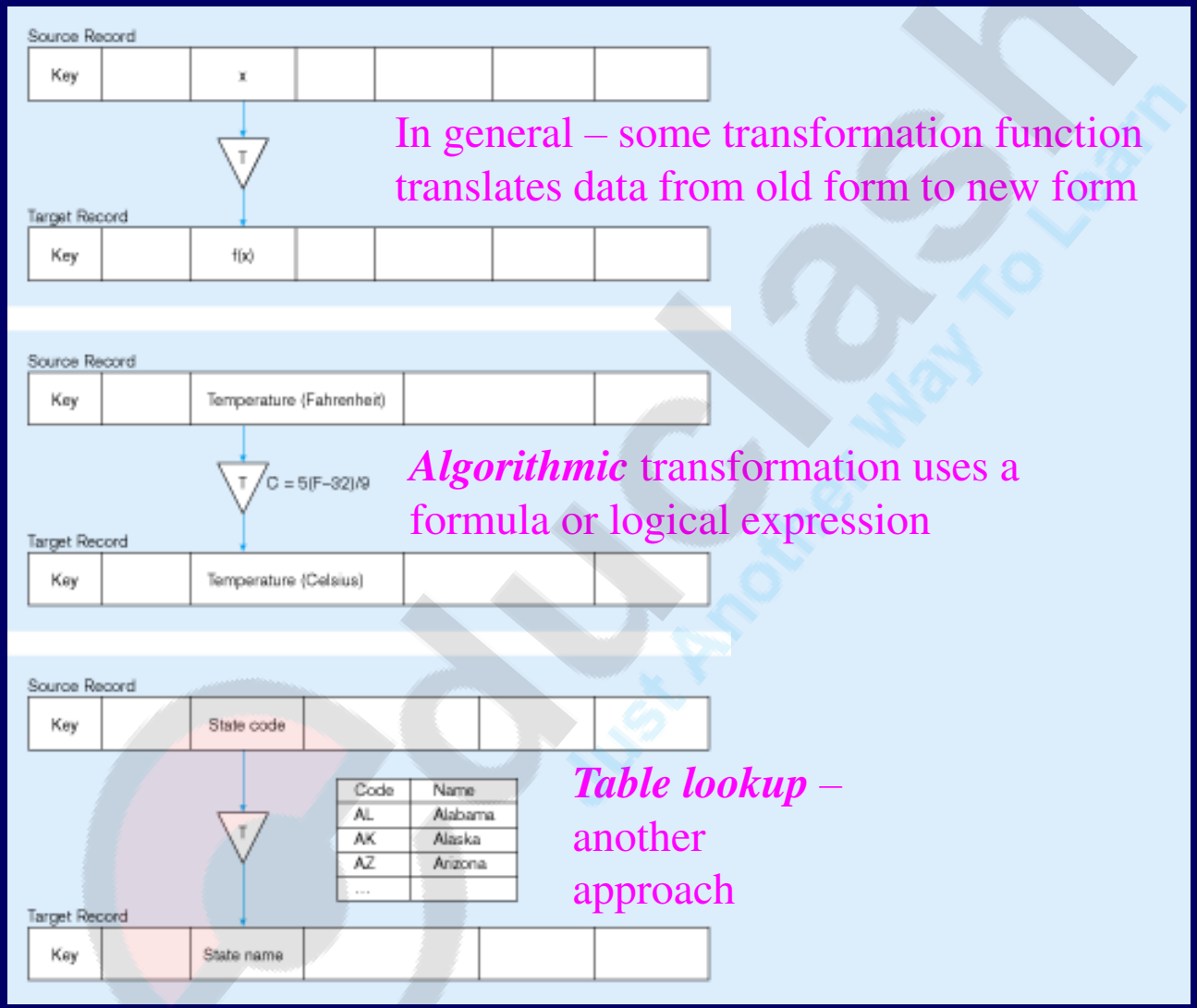
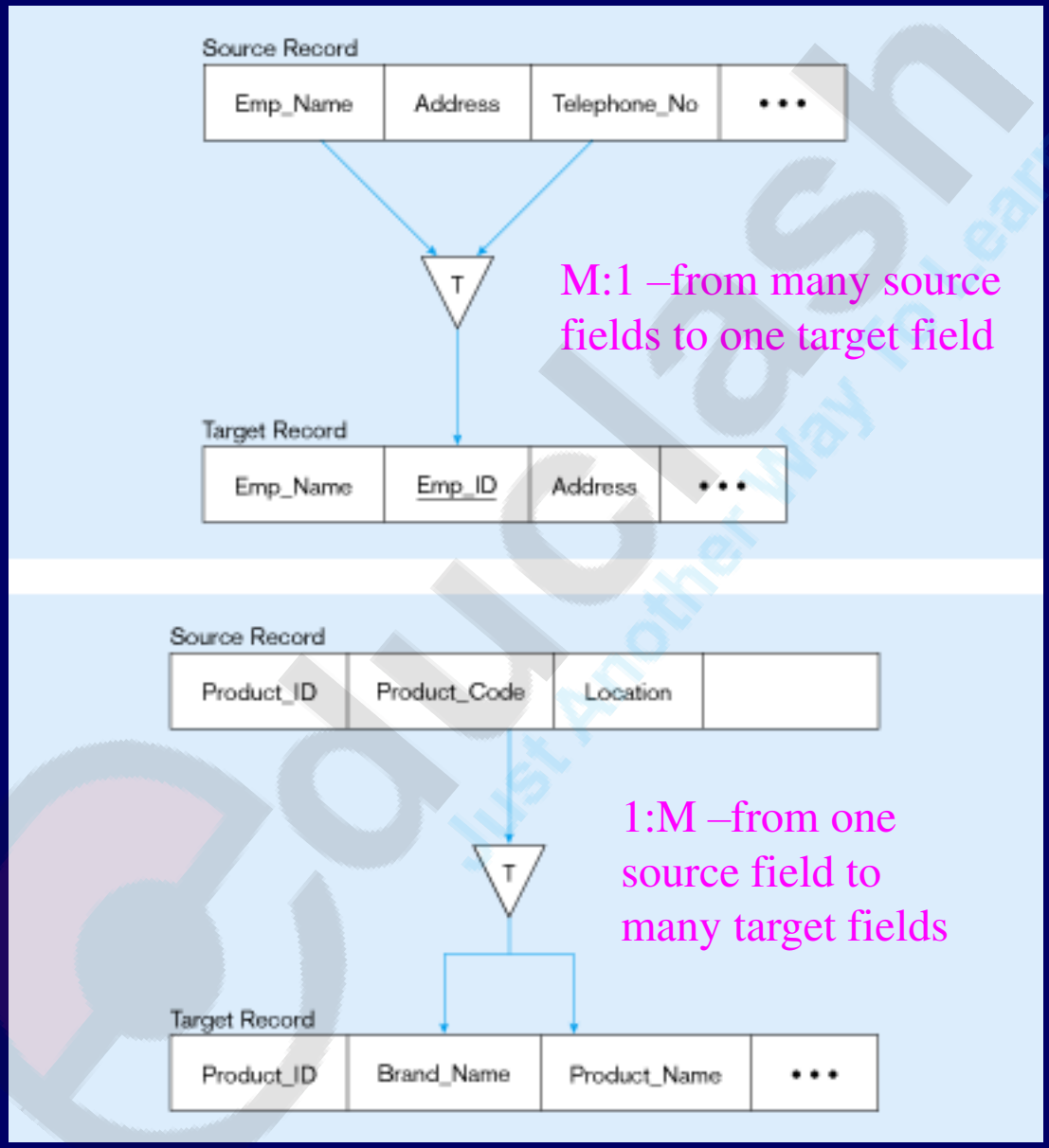


Figure 11-12: Multifield transformation



Data Integrity Problems

- Same person, different spellings
  - Agarwal, Agrawal, Aggarwal etc...
- Multiple ways to denote company name
  - Persistent Systems, PSPL, Persistent Pvt. LTD.
- Use of different names
  - mumbai, bombay
- Different account numbers generated by different applications for the same customer
- Required fields left blank
- Invalid product codes collected at point of sale
  - manual entry leads to mistakes
  - "in case of a problem use 9999999"

Data Transformation Example

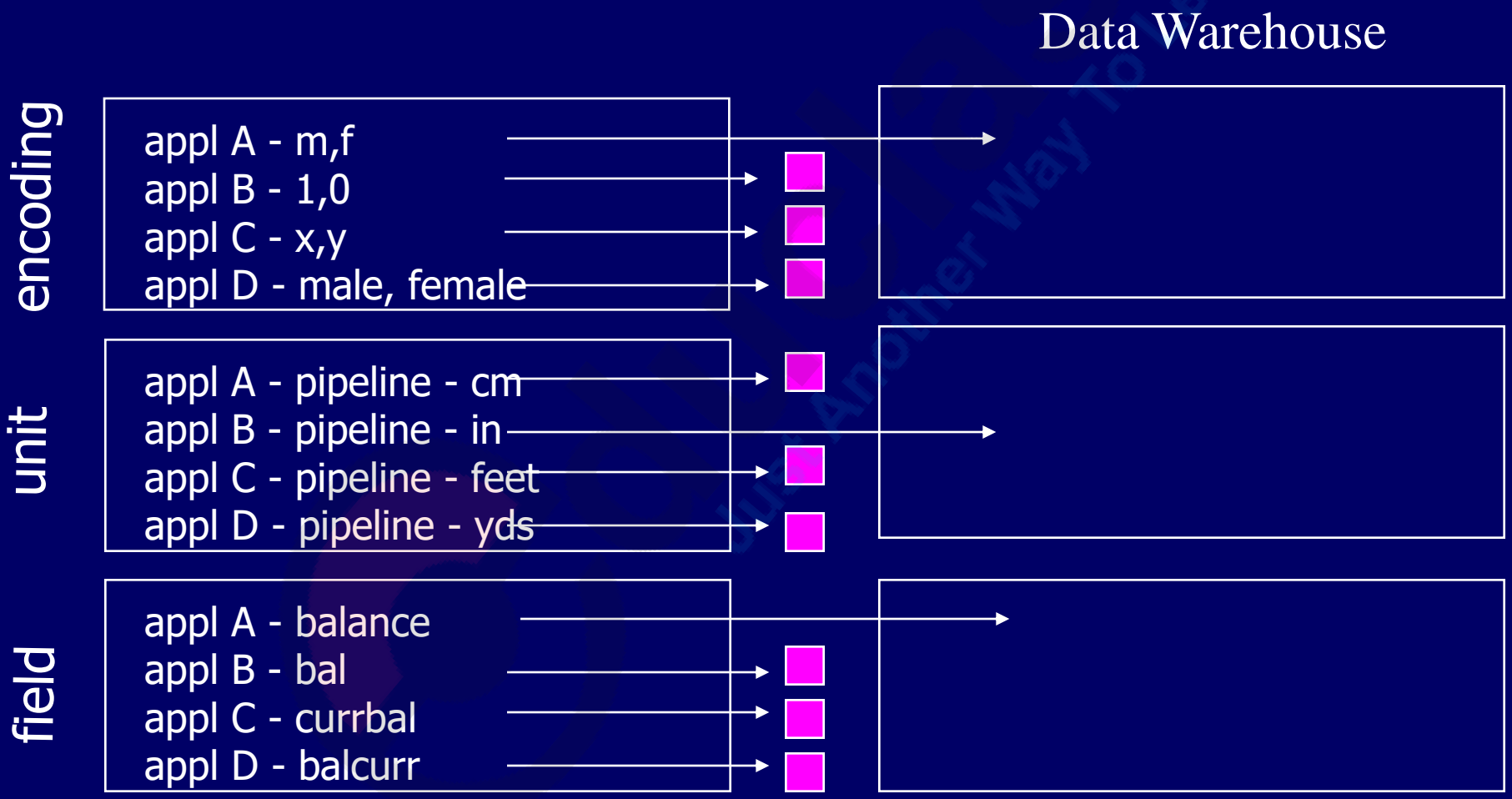
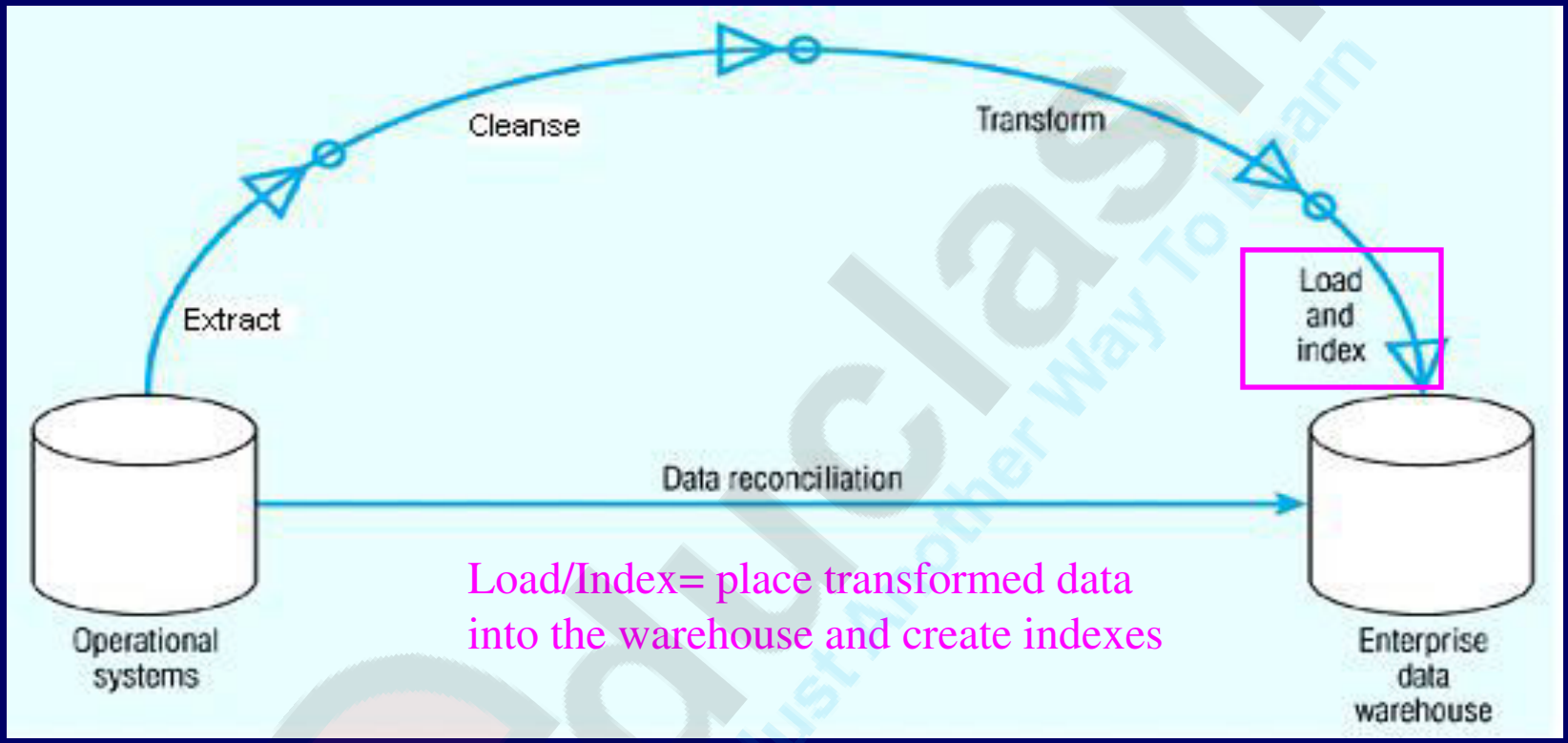


Figure 11-10: Steps in data reconciliation (continued)



Refresh mode: bulk rewriting of target data at periodic intervals

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

Loads

- After extracting, cleaning, validating etc. need to load the data into the warehouse
- Issues
  - huge volumes of data to be loaded
  - small time window available when warehouse can be taken off line (usually nights)
  - when to build index and summary tables
  - allow system administrators to monitor, cancel, resume, change load rates
  - Recover gracefully -- restart after failure from where you were and without loss of data integrity

Loading terminology

- Initial load
  - Populating all the data warehouse tables for the first time
- Incremental load
  - Applying ongoing changes as necessary in a periodic manner
- Full refresh
  - Completely erasing the contents of one or more tables and reloading with fresh data

Loading

- During loads DW or part of DW would be offline
- Schedule loads without affecting the user
- Have to consider bandwidth needed and impact of the transmission on network
- Data compression-some contingency plan
- Use batch load utility

Modes of data loading

- Load
  - Existing data is wiped out
- Append
  - Adds the incoming data
- Destructive merge:
  - Key 123 data PPP → Key 123 data AAA
- Constructive merge
  - Key 123 data PPP → Key 123 data PPP Key 123 data AAA

Data Marts

Data Mart

- A Data Mart is a smaller, more focused Data Warehouse – a mini-warehouse.
- It is a subset of a data warehouse that supports the requirements of particular department or business function

Reasons for creating a data mart

- To give users access to the data they need to analyze most often
- To provide data in a form that matches the collective view of the data by a group of users in a department or business function
- To improve end-user response time due to the reduction in the volume of data to be accessed
- To provide appropriately structured data as dictated by the requirements of end-user access tools


The characteristics :

- a data mart focuses on only the requirements of users associated with one department or business function
- data marts do not normally contain detailed operational data, unlike data warehouses
- as data marts contain less data compared with data warehouses, data marts are more easily understood and managed


- use less data so tasks such as data cleansing, loading, transformation, and integration are far easier, and hence implementing and setting up a data mart is simpler than establishing a corporate data warehouse

- The cost of implementing data marts is normally less than that required to establish a data warehouse

- The potential uses of a data mart are more clearly defined and can be more easily targeted to obtain support for a data mart project rather than a corporate data warehouse project

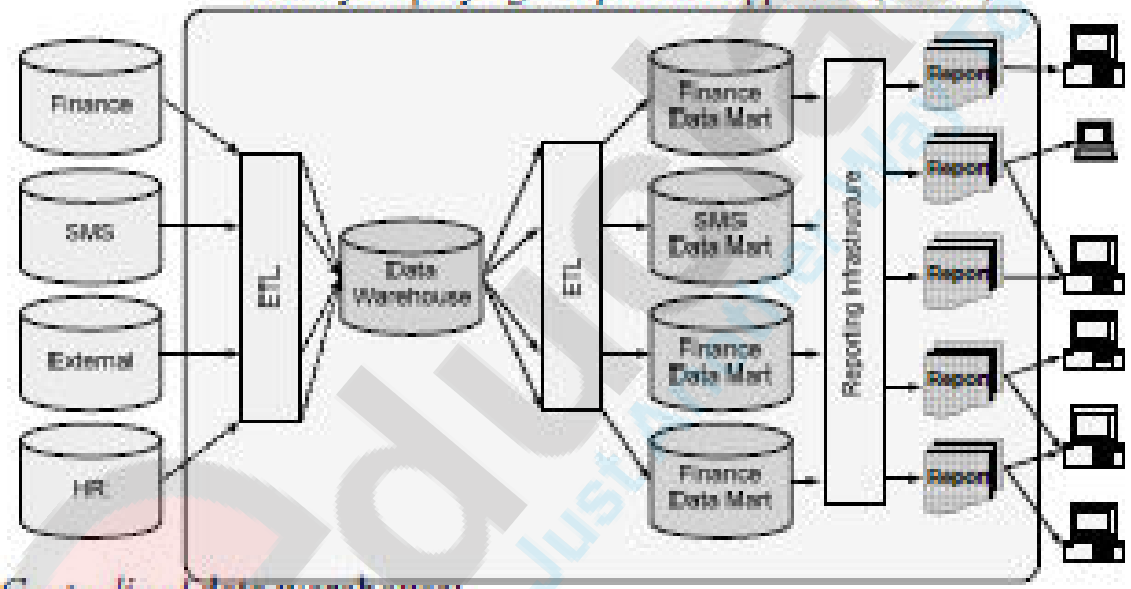
Type of Data Mart

There are two kinds of data marts—

- **Dependent** :A subset that is created directly from a data warehouse
- **Independent** :A small data warehouse designed for a strategic business unit or a department

Dependent Data Marts / Hub & Spoke

Usually employing a *Top-Down* approach (Inmon)



Centralized data warehouse:

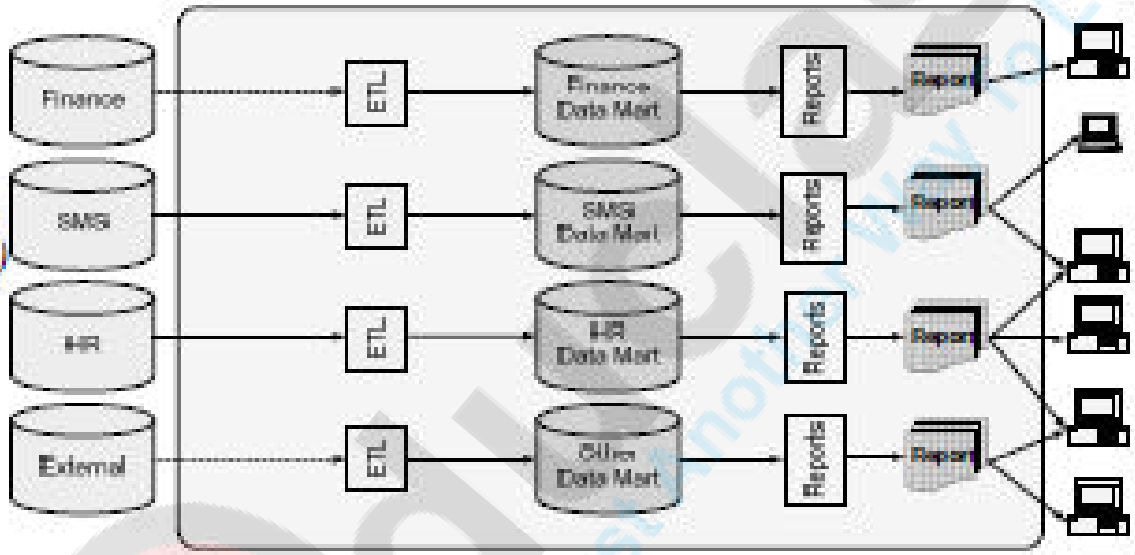
An approach also used in the early days, but refined over time

- Originally suggested extensive effort in building the DW
- Now recommends building DW incrementally

Dependent Data mart

- A dependent data mart is one whose source is a data warehouse.
- All dependent data marts are fed by the same source--the data warehouse.
- Dependent data marts are architecturally and structurally sound.

Independent Data Marts



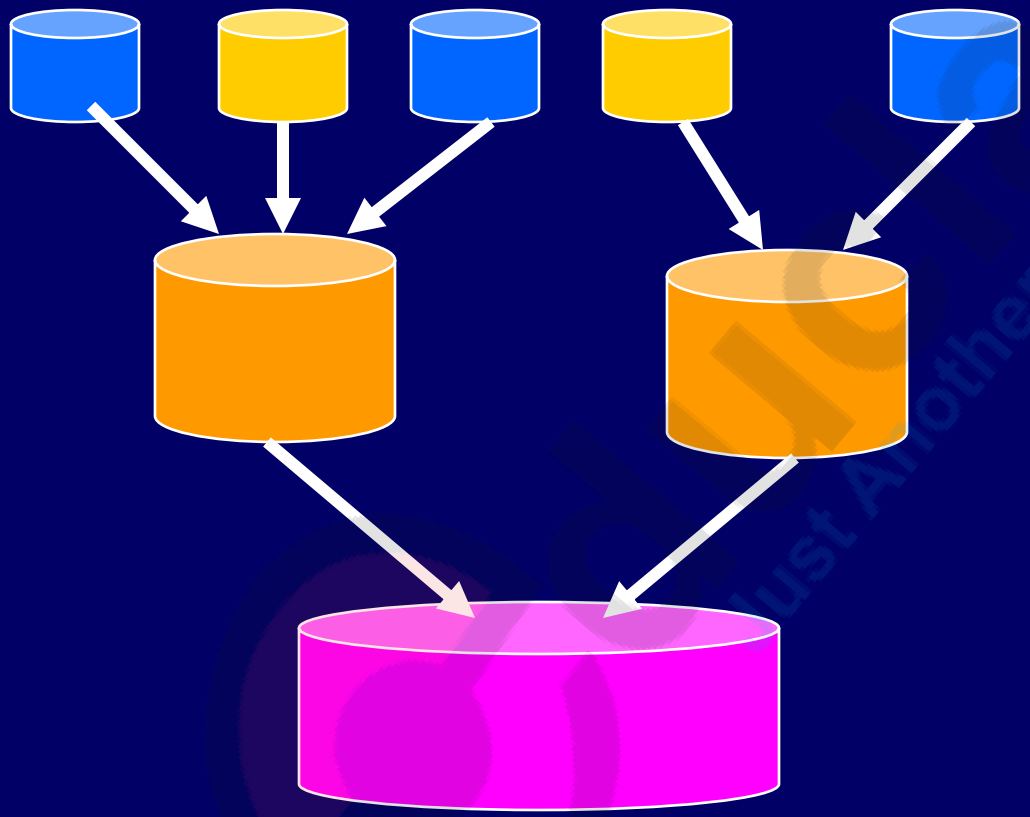
How Data Warehousing was often performed in the early days

- Individual projects developing solutions into functional silos
- No program / enterprise perspective
- No conformed dimensions

Independent Data Mart

- An independent data mart is one whose source is the legacy applications environment.
- Each independent data mart is fed uniquely and separately by the legacy applications environment.
- Independent data marts are unstable and architecturally unsound, at least for the long haul.
- The problem with independent data marts is that their deficiencies do not make themselves manifest until the organization has built multiple independent data marts.

Data Mart Centric

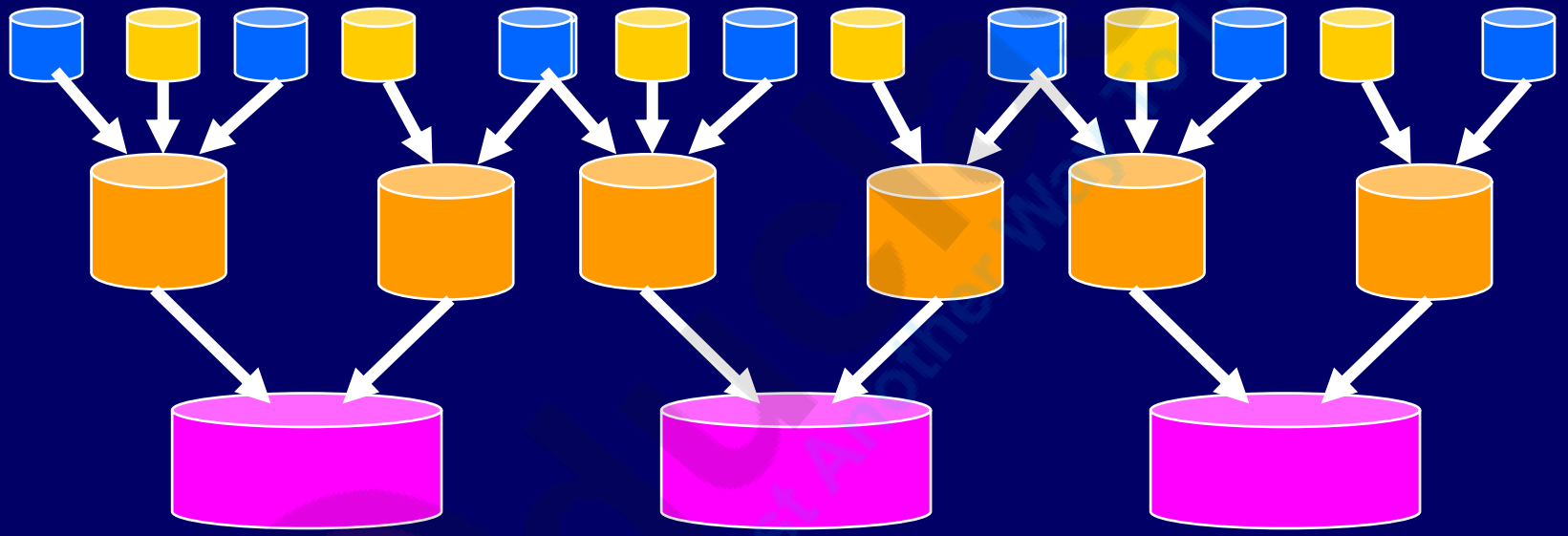


Data Sources

Data Marts

Data Warehouse

Problems with Data Mart Centric Solution



If you end up creating multiple warehouses, integrating them is a problem

True Warehouse

Data Sources

Data Warehouse

Data Marts

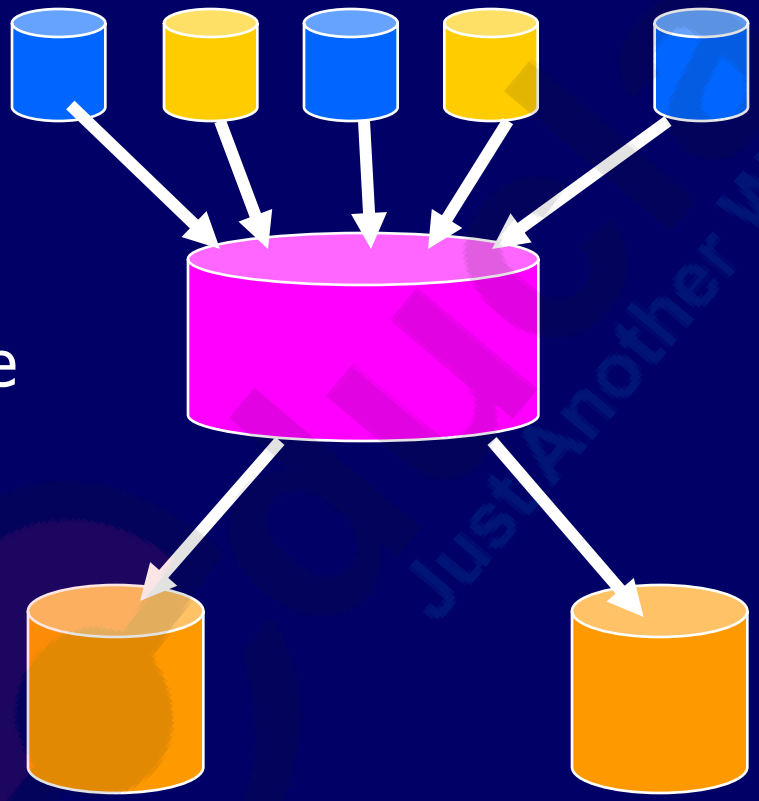


Table 11-2 Data Warehouse Versus Data Mart

Data Warehouse	*Data Mart*
Scope	*Scope*
• Application independent	• Specific DSS application
• Centralized, possibly enterprise-wide	• Decentralized by user area
• Planned	• Organic, possibly not planned
Data	*Data*
• Historical, detailed, and summarized	• Some history, detailed, and summarized
• Lightly denormalized	• 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	• Restrictive
• Data-oriented	• Project-oriented
• Long life	• Short life
• Large	• Start small, becomes large

Architectural types

- Indicates how data is stored
- Relationalship between DW and Data marts

Data Warehouse Architecture: Types

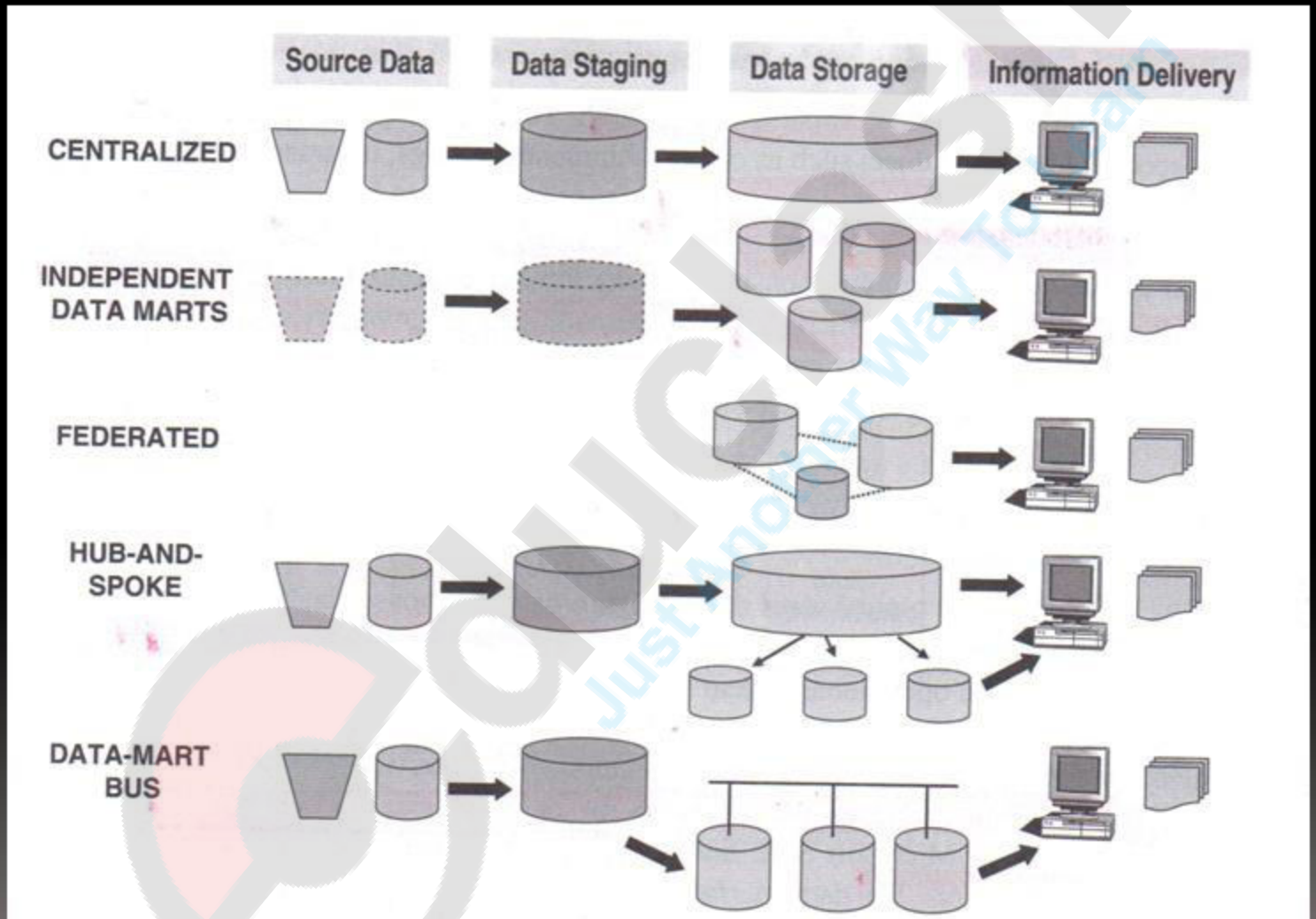


Figure 2-6 Data warehouse architectural types.

[Paulraj pg.no.33&156]

Granularity

- Level of detailed data
- Low level – fine grain
  - Operational level managers
  - Lot of data to be stored
- Middle level-Coarse grain
  - Middle level
- High level – high grain
  - GM
- Decide on the granularity level based on data types and system performance

[Paulraj pg.no. 28]

DATA GRANULARITY

Data granularity refers to the level of details of data in data warehouse.
The lower the level of details, the finer is the data granularity.

THREE DATA LEVELS IN A BANKING DATA WAREHOUSE

Daily Detail	Monthly Summary	Quarterly Summary
Account	Account	Account
Activity Date	Month	Month
Amount	Number of transactions	Number of transactions
Deposit/Withdrawal	Withdrawals	Withdrawals
	Deposits	Deposits
	Beginning Balance	Beginning Balance
	Ending Balance	Ending Balance

Data granularity refers to the level of detail. Depending on the requirements, multiple levels of detail may be present. Many data warehouses have at least dual levels of granularity.

Figure 2-4 Data granularity.

Approaches for building DW

Whether to build data mart or DW first?

- Top-down, bottom-up approaches or a combination of both
- Top down approach
  - Starts with overall design and planning (mature)
- Bottom up approach
  - Starts with experiments and prototypes (rapid)

- 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

Paulraj pg. no. 29

Approaches for building DW

- Top down approach
  - Build a mammoth DW, Require corporate effort.
  - Centralized repository for entire enterprise
  - Data in DW stored in lowest level of granularity based on normalized data model.
  - Organized on ER model
  - Centralized rules and control
  - Data received from data staging area
  - Takes longer to build
  - Risk of failure high
  - Needs high level of cross functional skills
  - Difficult to sell to senior mngt. And sponsors

Approaches for building DW

- Bottom up approach
  - Look at local departmental requirement with Data marts
  - A single business process
  - STAR joins
  - Technology optimal for data access and analysis
  - Structure to suit the departmental view of data
  - Faster and easier implementation of manageable pieces
  - Favorable ROI
  - Less risk of failure
  - Inherently incremental , can schedule imp data marts first
  - Allows project team to learn and grow
  - Each Data mart has narrow view of its own data
  - Redundant data in each data marts
  - Inconsistent
  - Unmanageable interfaces

The difference bewteen OLTP and data warehousin

- A DBMS built for online transaction processing (OLTP) is generally regarded as unsuitable for data warehousing because each system is designed with a differing set of requirements in mind

- example: OLTP systems are design to maximize the transaction processing capacity, while data warehouses are designed to support ad hoc query processing

OLTP vs Data Warehouse

- OLTP
  - Data set organized around individual applications to support those particular operational systems.
  - Application oriented
  - E.g. order processing, customer billing….
  - Support day to day decisions
- Warehouse
  - Data set organized in a way that all data relating to the same real world business subject are tied together.
  - Subject oriented
  - Eg. Sales, shipment, inventory…..
  - Support strategic decisions

OLTP vs Data Warehouse

- OLTP
  - Used to run business
  - Detailed data
  - Current up to date
  - Repetitive access
  - Clerical User
- Warehouse
  - Used to analyze business
  - Summarized and refined
  - Snapshot data, Integrated Data, Current +history data
  - Ad-hoc access
  - Knowledge User (Manager)

OLTP vs Data Warehouse

- OLTP
  - Performance Sensitive
  - Few Records accessed at a time (tens)
  - Read/ Update Access
  - No data redundancy
  - Database Size
    100MB -100 GB
  - Design is normalized
- Data Warehouse
  - Performance relaxed
  - Large volumes accessed at a time(millions)
  - Mostly Read (Batch Update)
  - Redundancy present
  - Database Size
    100 GB - few terabytes
  - Design is de-normalized

OLTP vs Data Warehouse

- OLTP
  - Transaction throughput is the performance metric
  - Simple queries
  - Thousands of users
  - Internal data source
  - Time element not necessary
  - Data granularity is low
- Data Warehouse
  - Query throughput is the performance metric
  - Complex queries
  - Hundreds of users
  - Internal and external data source
  - Every data structure has a time element
  - At least dual granularity is maintained

To summarize …

- OLTP Systems are used to *"run"* a business
- The Data Warehouse helps to *"optimize"* the business

syllabus

- Dimensional analysis[ch 5 paulraj]
- Define cubes.
- Drill- down and roll- up – slice and dice or rotation
- OLAP models- ROLAP and MOLAP[ch 15 paulraj]
- Define Schemas- Star, snowflake and fact constellations [chapt 10&11 paulraj] [ch 3.2 Han Kamber]

Dimensional analysis[ch 5 paulraj]