NATIONAL ENGINEERING CENTER
University of the Philippines
Diliman, Quezon City

6.0 ETL: Extraction, Transformation and Loading

Eugene Rex L. Jalao, Ph.D.

Associate Professor

Department Industrial Engineering and Operations Research

University of the Philippines Diliman

Module 2 of the Business Intelligence and Analytics Track of UP NEC and the UP Center of Business Intelligence

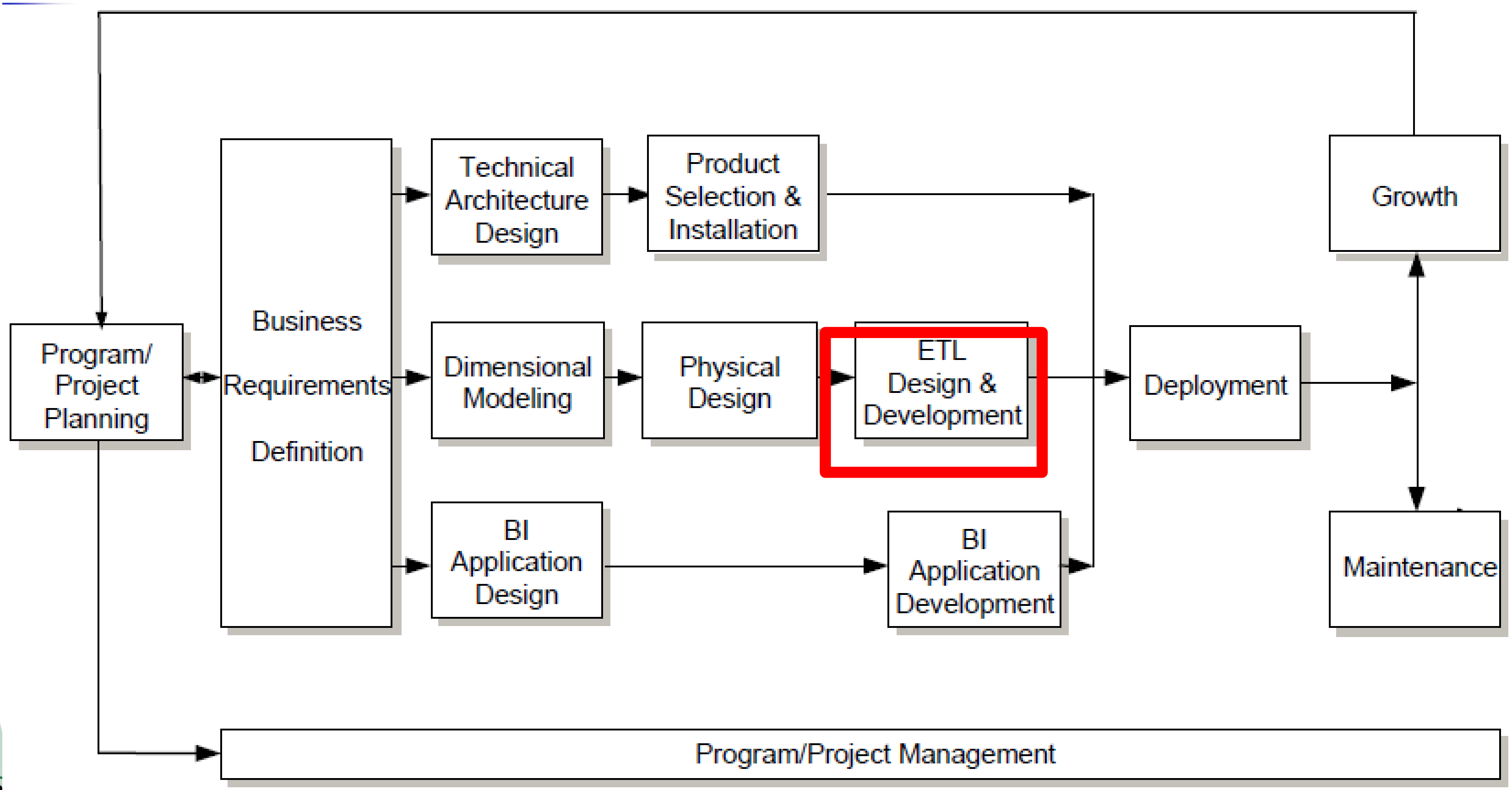
Outline for This Training

1. Introduction to Data Warehousing
2. DW Lifecycle and Project Management
   - Case Study on DW PM
3. Dimensional Modeling
4. Designing Fact Tables
5. Designing Dimension Tables
   - Case Study on Dimension Modeling
6. **Extraction Transformation and Loading**
   - **Case Study on ETL Planning**
7. Transformation and Loading Methodologies
   - Case Study on ETL

Outline for This Session

- What is ETL?
- ETL Process
  - Kimball’s 38 Step Method
  - Schmitz ETL Roadmap
- HW/DB Architecture Considerations
- ETL Tools
- Extraction
  - Data Profiling
  - Source-to-Target Mapping
- Case Study

The Kimball Lifecycle


Program/ Project Planning
Business Requirements Definition
Technical Architecture Design
Product Selection & Installation
Dimensional Modeling
Physical Design
ETL Design & Development
BI Application Design
BI Application Development
Deployment
Growth
Maintenance
Program/Project Management

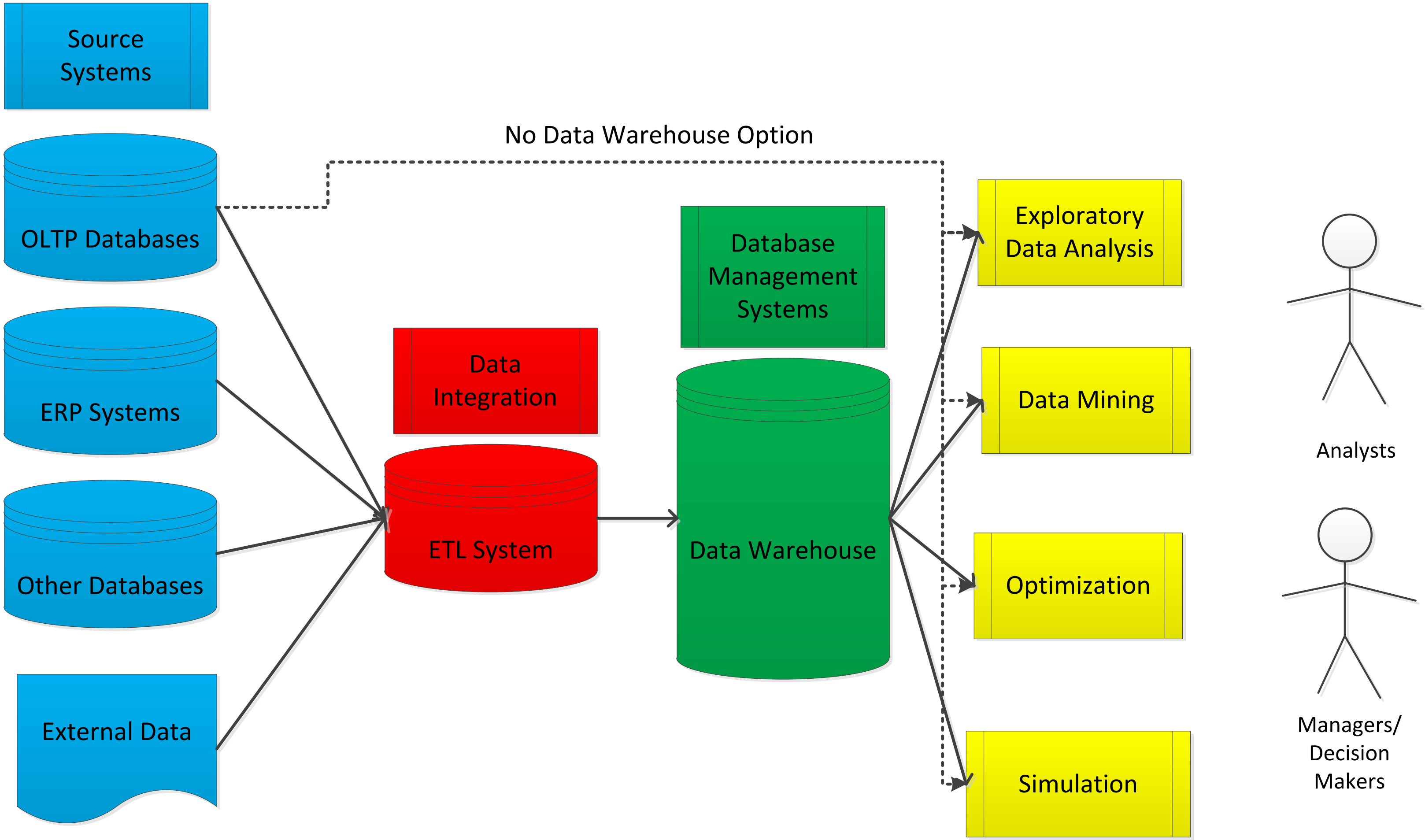
Mindset

- The importance of data quality
- Scalability
- Maintainability
- It is cheaper and faster to do things right the first time

What is ETL?

- Extracting and cleaning data from source systems, transforming it to the desired dimensional form, and loading it into the target data warehouse structure (dimensional)
- According to Kimball: Extract, Clean and Conform, Deliver, Manage
- Also According to Schmitz : Extract, Clean, Transform, and Load

ETL System


Source Systems
No Data Warehouse Option
OLTP Databases
ERP Systems
Other Databases
External Data
Data Integration
ETL System
Database Management Systems
Data Warehouse
Exploratory Data Analysis
Data Mining
Optimization
Simulation
Analysts
Managers/ Decision Makers

Disclaimer

- The target is not a copy of operational data
  - Content enhanced
  - It is Cleaned
  - It is Conformed
  - Integrated
  - Historical context added to historical transaction or event data

The Harsh Reality

- ETL is a major failure point in data warehousing
- Underestimating the effort involved in the ETL process is rampant
- Underestimating data quality problems is a prime culprit
- Providing for contextual history is another
  - Have management make the decision whether the goal is to provide optimal analytical capabilities to the business or lessen the ETL effort
- Scalability and performance are crucial

ETL Overview

- It is not a one time event as new data is added to the Data Warehouse **periodically** – monthly, daily, hourly
- Because ETL is an integral, ongoing, and recurring part of a data warehouse
  - Automated
  - Well documented
  - Easily changeable

The Good News

- A standardized approach and proven techniques and templates can exponentially lessen the amount of effort required and can ensure scalability and performance
- Must start from here, and probably won’t be able to go back and re-design or code

Where do We Start

- Understand Our Target Data Warehouse Environment
  - The Business Accessible Data Warehouse
  - Dimensional Schema Basics
  - Dimension Warehouse Keys
  - Dimension Attribute History Tracking
  - ETL in the DW Project

Outline for This Session

- What is ETL?
- **ETL Process**
  - **Kimball's 38 Step Method**
  - **Schmitz ETL Roadmap**
- HW/DB Architecture Considerations
- ETL Tools
- Extraction
  - Data Profiling
  - Source-to-Target Mapping
- Case Study

Need for an ETL Process

- 70% of the effort to build the data warehouse is ETL
- Too many data warehouses are built by trial and error ETL
- Kimball and Schmitz say you need an ETL methodology for consistency and productivity
- Kimball defines 38 steps in his high level ETL methodology
- Schmitz's Methodology focuses on HOW to build the tables using intermediate tables

ETL Process Parts

- Data Sourcing
  - Document Source Data
  - Analyze Source Data (format, values, quality)
  - Determine the amount of history to be initially loaded
    - Is it all in the same form
    - Is there contextual history
  - Determine the currency requirement
  - Determine how to incrementally extract new and changed data

ETL Process Parts

- Model Target
- Map source elements to target elements
- Define transformation rules
- Develop the extraction and transport processes
- Develop the transformation processes
- Develop the load and update processes

The Historical Load

- Understand if all historical data is in the same format or not
- Test data quality and processes extensively with smaller samples of data
- With the caveat that you must do volume testing to make sure that your processes are scalable
- Break up the production load into manageable pieces

Dimension and Fact ETL Processes

- Facts and Dimensions
  - Should be designed and built separately
  - Design Dimensions First -> Keys For Fact Tables
  - Some special dimensions like transactions may be processed during the fact table process

Kimball's 38 Subsystems

- Recommended **Best Practices List** of the components of an ETL System for any Data Warehouse
- Bad News: 38 Components!
- Good News: Exhaustive
- See **Exhibit 9**

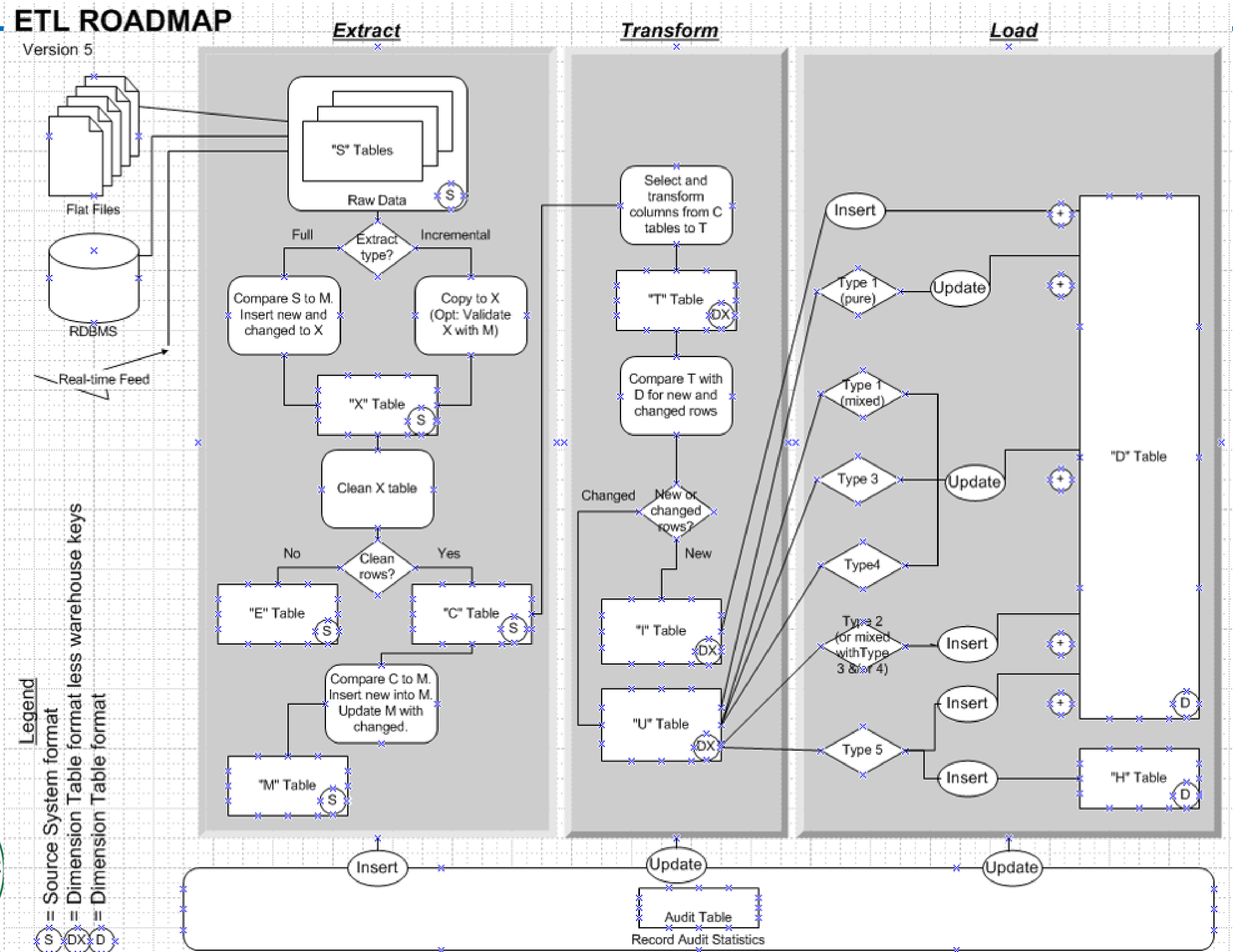
Kimball’s 38 Subsystems

- Extract – 3 steps
  - Gathering raw data and writing to disk before any processing is performed
- Transform (Clean and Conform) – 5 Steps
  - Cleaning the data and creating conformed dimensions and facts
- Load (Deliver) – 17 Processes
  - Structuring and loading the data into relational and multidimensional structured dimensional databases
- Manage – 13 Processes
  - Managing the entire ongoing ETL process – automatically and manually

Schmitz' ETL Method

- One-page template provides a straightforward roadmap that can be reused to build and maintain all dimensions.
- Uses intermediate tables in staging area to organize and stage data as it goes through ETL process
- ETL process can be hand-coded or used with ETL tools

Schmitz' Generic ETL Roadmap



Intermediate Tables with Naming Convention

- D: Dimension table
- F: Fact table
- S: Source table – contains all data copied directly from a source file
- X: eXtract table – contains changed source data only. Changes may be from an incremental extract or derived from a full extract.
- C: Clean table – contains source data rows that have been cleaned

Intermediate Tables with Naming Convention

- E: Error table – contains error rows found in source data
- M: Master table – maintains history of all clean source rows
- T: Transform table – contains the data resulting from a transformation of source data (merges, splits, transformations of clean rows from one or more source tables)

Intermediate Tables with Naming Convention

- I: Insert table – contains new data to be inserted in dimension tables
- U: Update table – contains changed data to update dimension tables
- H: History table – contains dimension history table data

Intermediate Table Formats

- S, X, C, E, and M tables are in source table (S table) format
- T, U, and I tables are in target table (D table) format without the warehouse key
- D and H tables are in D table format with the warehouse key
- F tables are in F table format

Why Intermediate Tables

- It's a tradeoff between
  - Development time
  - Availability
  - Restartability
  - Auditing
  - Performance
- Data arrival time differences

We Want the Optimal Balance

- Breaks development into logical, debugable steps
- Provides restart points
- Allows processing do be done in steps as data is available
- Allows detailed auditing of each step in the process
- Does impact performance so it needs to be done optimally

Outline for This Session

- What is ETL?
- ETL Process
  - Kimball's 38 Step Method
  - Schmitz ETL Roadmap
- **HW/DB Architecture Considerations**
- ETL Tools
- Extraction
  - Data Profiling
  - Source-to-Target Mapping
- Case Study

Architectural Considerations

- Hardware and Database Architecture
- Amount of processing done on source system
- What types of tools will be used

Hardware and Database Architecture Considerations

- Separate Hardware Server for ETL processing (or partitioned server)
- Ideal would be to devote extra resources to DW when ETL processing is not happening
- Same OS type for publishing efficiency

Hardware and Database Architecture Considerations

- Same Server for both ETL and DW (DW access definitely impacted during ETL processing)
- Separate database instances
  - Pro – can configure db parms for ETL, not really necessary most of the time
  - Cons – more systems resources used
- Same database instance
  - Pro – less system resources used
  - Con – Logical separation by schema owner

Source Processing

- If operational system is constrained get data as efficiently as possible
- If extract window on operational system is constrained make sure that nothing can prevent the extract from running (even a down network) during its window
- Do as much cleaning as possible here (may be better feedback to encourage source system fixes)

Source System Quality Enhancers

- Establish enforcement of required input fields like “Zip Code”
- Provide drop down data windows/lookups
- Online De-duplication
- A new entry looks for potential matches and prompts to continue with the add or accept one of the matches

Outline for This Session

- What is ETL?
- ETL Process
  - Kimball's 34 Step Method
  - Schmitz ETL Roadmap
- HW/DB Architecture Considerations
- **ETL Tools**
- Extraction
  - Data Profiling
  - Source-to-Target Mapping
- Case Study

Source to Target Packages

- Several companies have a strong ETL tools and a fairly complete suite of supplementary tools
- Three general types of Source to Target tools
  - Code generators
    - actually compile ETL code, typically COBOL which is used by several large companies that use mainframe
  - Engine based
    - easy-to-use graphic interfaces and interpreter style programs
  - Database based
    - manual coding using SQL statements augmented by scripts.

Major Advantages

- Automatic meta data capture
  - After initial development is key
- Myths
  - Standardization
    - ETL Programmers may not be consistent with process. Better to follow standard processes
  - Faster Development
    - Need to learn the tool first
  - Cheaper developers
    - Experienced developers with better ETL tools are higher paid

Disadvantages

- Performance
  - But difference is easing as ETL tools mature
- Databases are getting more native capabilities
  - Microsoft has their own ETL system
- Custom programming can be poor performing and non-scalable also
- Speed of development decreases when project gets complex

Well Known ETL Tools

- Commercial
  - Ab initio
  - IBM DataStage
  - Informatica PowerCenter
  - Microsoft Data Integration Services
  - Oracle Data Integrator
  - SAP Business Objects – Data Integrator
  - SAS Data Integration Studio

Well Known ETL Tools

- Open-Source Based
  - Adeptia Integration Suite
  - Apatar
  - CloverETL
  - Pentaho Data Integration (Kettle)
  - Talend Open Studio/Integration Suite
  - R/R Studio

ETL Recommendations

- Set Standards and develop highly scalable templates
- Integrate DB stored procedure use
- Ensure that the tool can use native database fast bulk load capabilities (nologging)
- Get supplemental non-vendor qualified training Use high performing scalable methods and standards regardless of whether you use an ETL tool or use custom coding

ETL Tools

- The "best" tool does not exist
- Choose based on your **own needs**
- Check first if the "standard tools" from the big vendors are ok

Outline for This Session

- What is ETL?
- ETL Process
  - Kimball’s 38 Step Method
  - Schmitz ETL Roadmap
- HW/DB Architecture Considerations
- ETL Tools
- **Extraction**
  - Data Profiling
  - Source-to-Target Mapping
- Case Study

Extraction Considerations

- Availability: Not available due to non existent data, or poor data quality
- Restartability: Restart from Beginning
- Performance: Processing Options
- Scalability: Scaling-up for future expansion
- Auditing: Tracking errors
- Responsibility: Have source system stewards be responsible for extraction

List of Data Extraction Issues

- Source Identification —identify source applications and source structures.
- Method of extraction —define whether the extraction process is manual or tool-based.
- Extraction frequency— establish how frequently the data extraction must by done—daily, weekly and so on.
- Time window —for each data source, denote the time window for the extraction process.

List of Data Extraction Issues

- Job sequencing —determine whether the beginning of one job in an extraction job stream has to wait until the previous job has finished successfully.
- Exception handling —determine how to handle input records that cannot be extracted

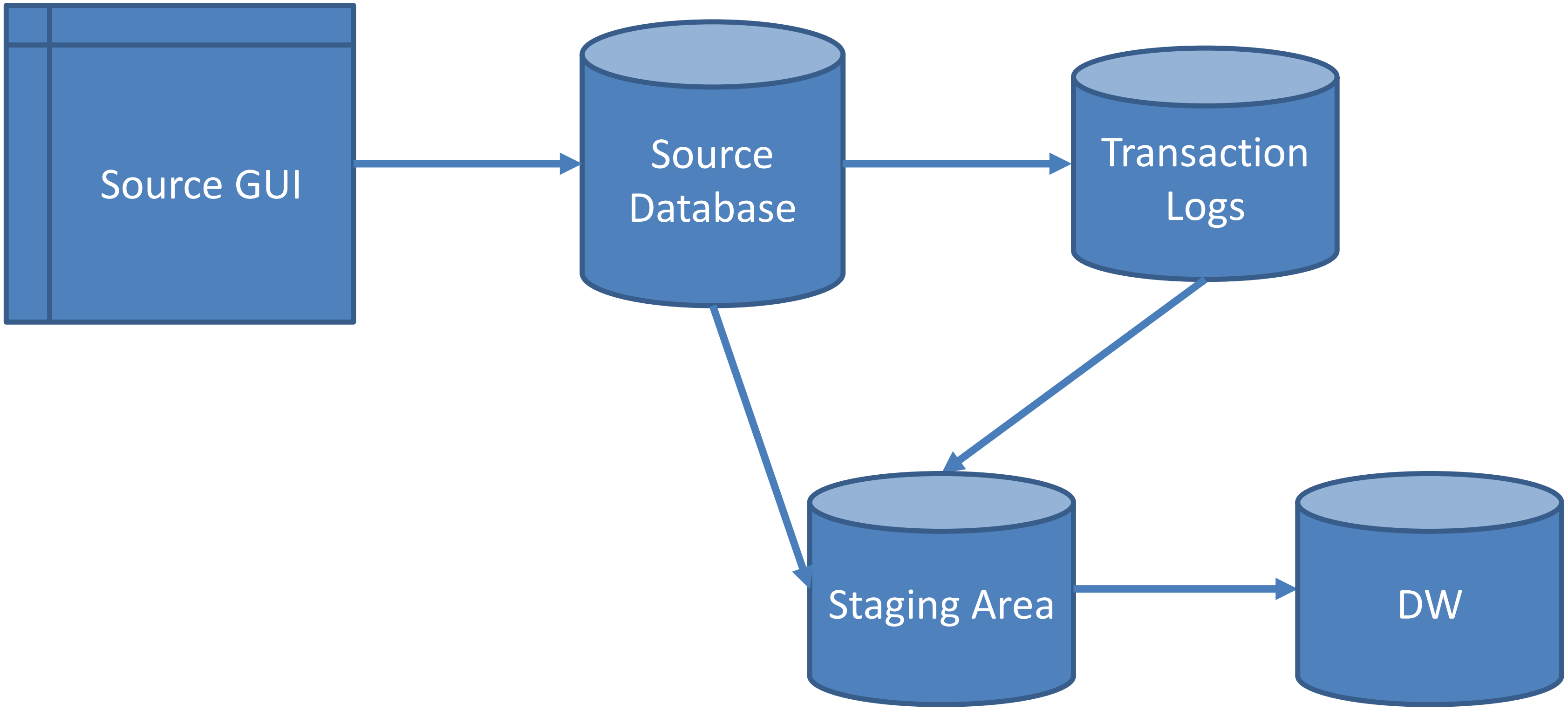
Options for Data Extraction

- Immediate Data Extraction.
  - Capture through Transaction Logs.
  - Capture through Database Triggers.
  - Capture in Source Applications.
- Deferred Data Extraction.
  - Capture Based on Date and Time Stamp.
  - Capture by Comparing Files.

Log Mining

- DBMSs keep detailed before and after records related to each transaction
- Use these logs for extracting new or changed data
  - Use of on-line logs for current changes
  - Use of archived logs for older changes

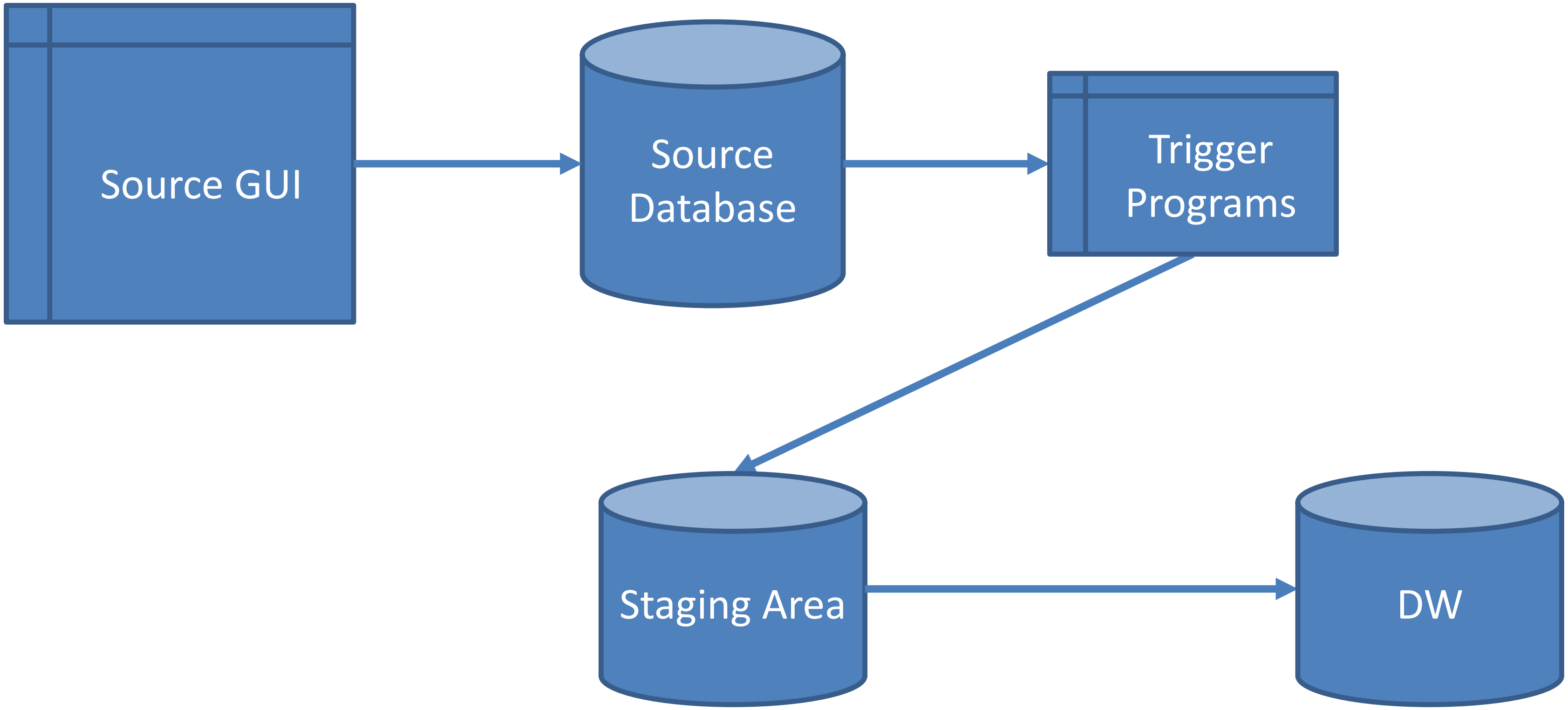
Log Mining


Source GUI
Source Database
Transaction Logs
Staging Area
DW

Database Triggers

- Occurs right at the source and is therefore quite reliable.
- You can capture both before and after images.
- Building and maintaining trigger programs puts an additional burden on the development effort.
- Execution of trigger procedures during transaction processing of the source systems puts additional overhead on the source systems.
- This option is applicable only for source data in databases.

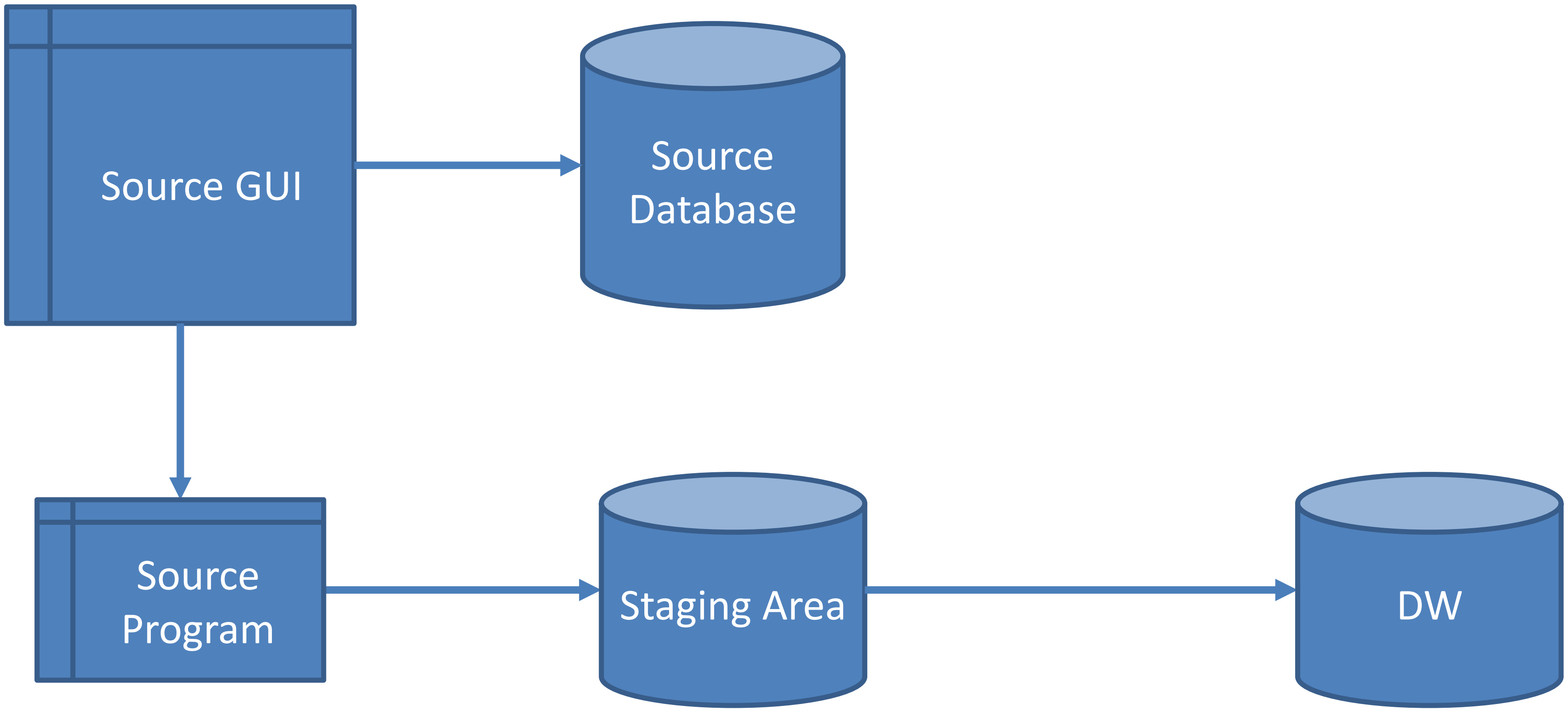
Database Triggers



Capture in Source Applications

- Once updated in the source applications e.g. ERP or POS systems, copies of new or changed data forwarded to data staging area immediately or by batch.
- Additional burden on developers
- Additional performance overhead on live applications

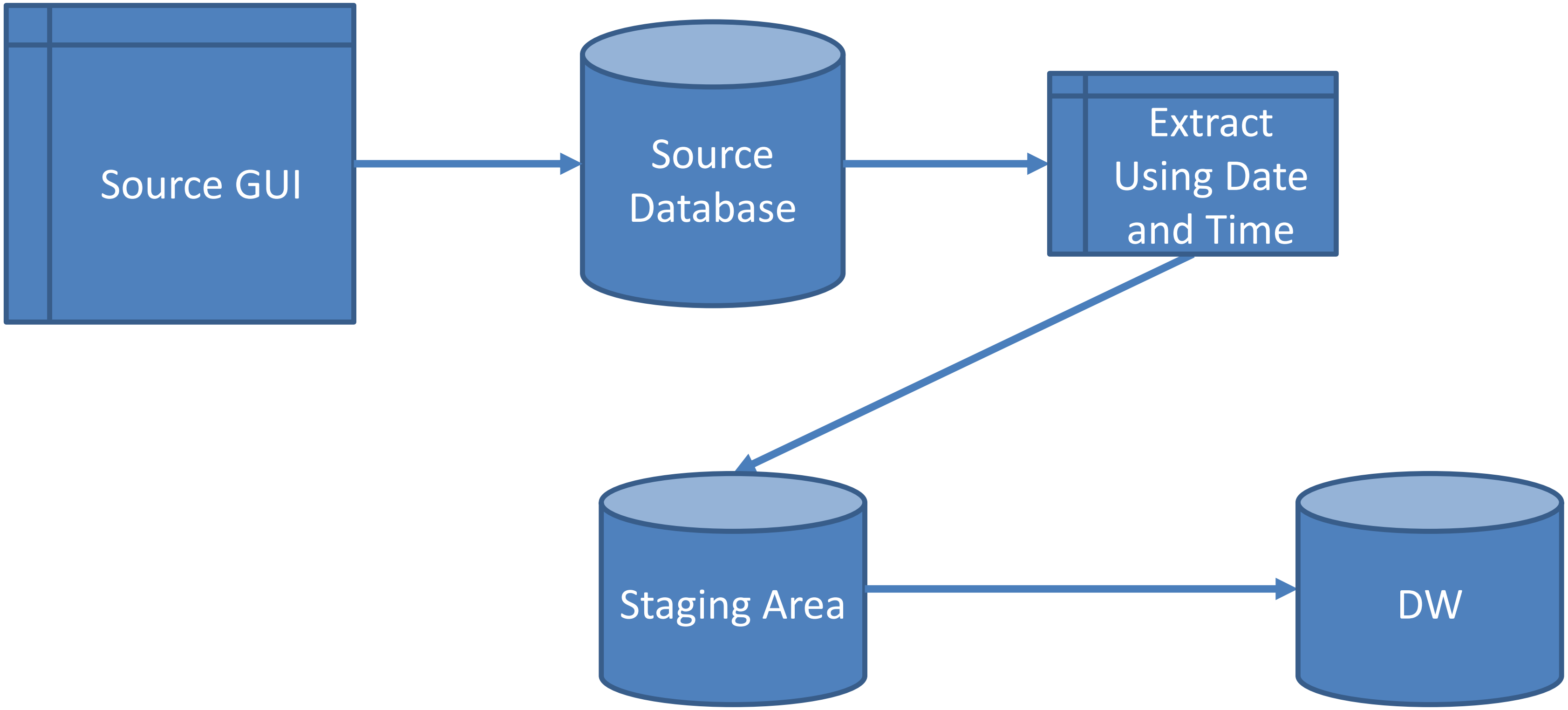
Database Triggers



Capture Based on Date and Time Stamp

- If a source record gets deleted in between two extract runs, the information about the delete is not detected.
- You can get around this by marking the source record for delete first, do the extraction run, and then go ahead and physically delete the record.
- This means you have to add more logic to the source applications.

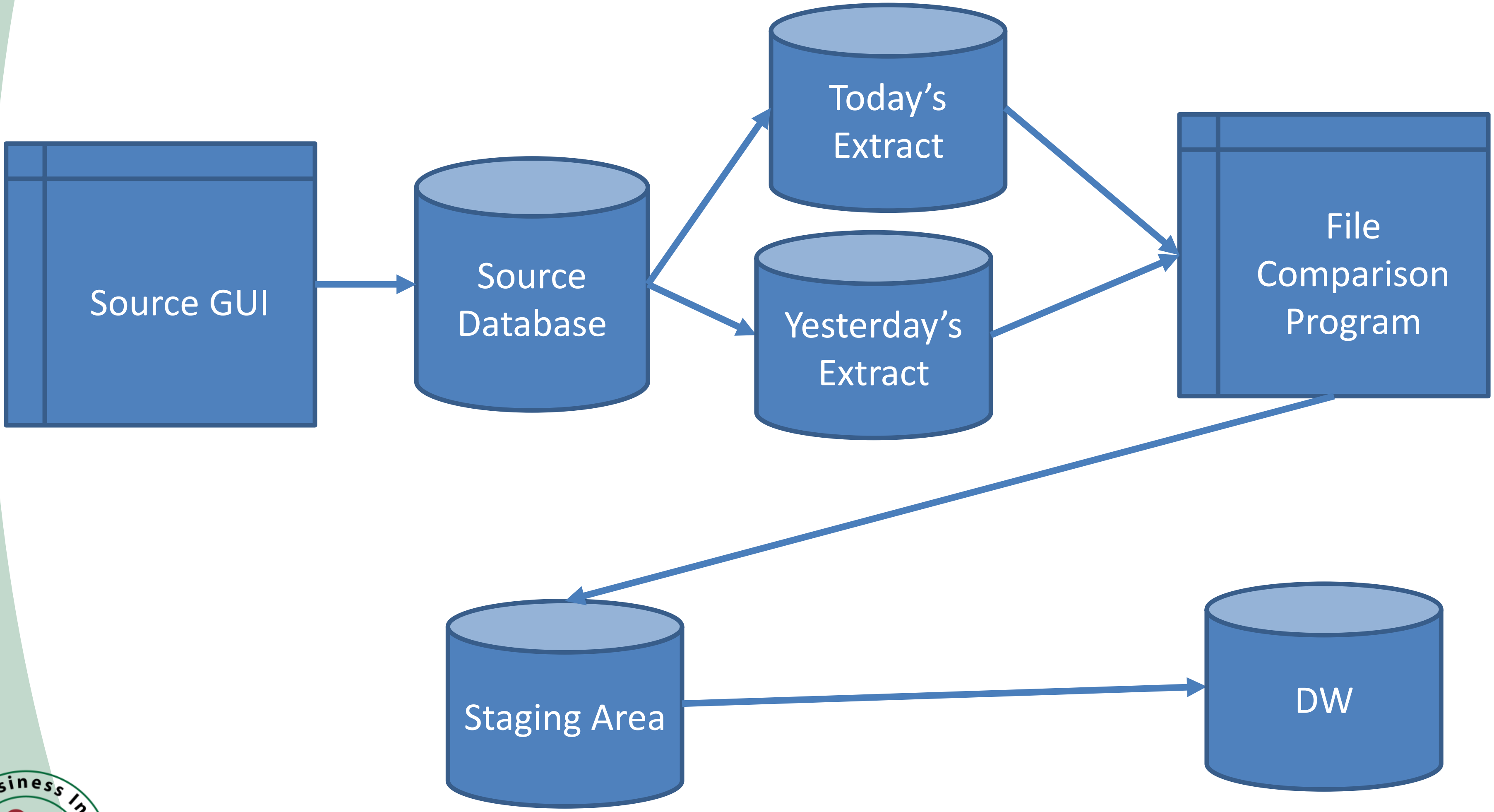
Capture Based on Date and Time Stamp



Capture by Comparing Files

- If none of the above techniques are feasible for specific source files in your environment, then consider this technique as the last resort.
- This technique is also called the snapshot differential technique because it compares two snapshots of the source data.

Capture by Comparing Files



Comparison of Methodologies

Methodology	Advantages	Disadvantages
Log Mining	• Performance of source system not affected • No revisions of existing source applications • No internal Costs	• Not much flexibility for source capture • Cannot be used on file-based systems
Database Triggers	• No Revisions of Existing Source Applications • No internal Costs	• Performance of Source Systems Affected a Bit • Not much flexibility for source capture • Cannot be used on File-Based Systems
Capture in Source Applications	• Good Flexibility for Capture Specifications • Can be used on File-Based Systems	• Performance of Source Systems Affected a Bit • High Internal Costs due to Development • Major Revisions on Existing Systems

Comparison of Methodologies

Methodology	Advantages	Disadvantages
Capture Based on Date and Time Stamp	• Good Flexibility for Capture Specifications • Performance of source system not affected • Can be used on File-Based Systems • Little or No internal Costs	• Major Revisions on Existing Systems
Capture by Comparing Files	• Good Flexibility for Capture Specifications • Performance of source system not affected • No Revisions of Existing Source Applications • Little or No internal Costs	• Major Revisions on Existing Systems

General Processing Options

- ETL Tools
  - Most Common
- Programs with database calls
  - Heavy Reliance on SQL Statements
- SQL set processing statements
- Programs or utilities using sorted flat files

Set Based v Cursor Based SQL Operations

- Use set-based whenever possible
- Cursor based can cause severe performance degradation
  - processes transactions one row at a time

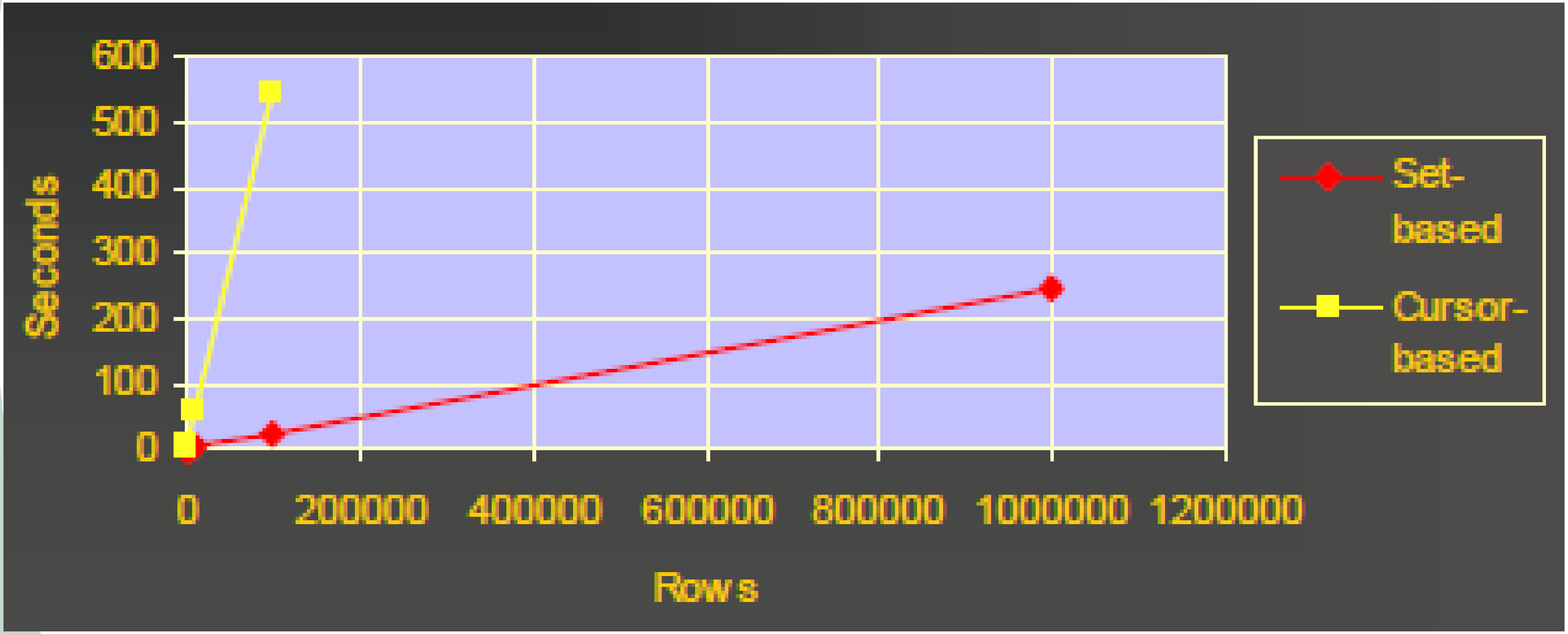
Cursor-Based Example

```
Open cursor for select from input
Loop: Fetch input row
     SELECT FROM dimension table
     WHERE input.operational id =
          dim.operational id
     If found
          If changed
               UPDATE dimension table
          else ignore
     Else INSERT new dimension row
```

Set-Based Example

```
UPDATE dim_table dim SET VALUES =
   (SELECT statement FROM input_table
   WHERE input.operational_id =
   dim.operational_id);
INSERT INTO dim_table (SELECT
   statement FROM input_table WHERE
   NOT EXISTS (SELECT operational_id
   FROM dim_table);
```

Set-based vs Row -based



Set Based v Cursor Based SQL Operations

- Set based
- Pros
  - Processing efficiency (~10X)
  - Scalable (nologging/parallel)
- Cons
  - Must be sure to construct auditable processes
  - Multiple steps often required (still faster than issuing thousands of logged SQL instructions)
  - Complex logic not straightforward

Set Based v Cursor Based SQL Operations

- Cursor based
- Pros
  - Most programmers are skilled at this
  - Complex logic fairly straightforward
- Cons
  - Performance
  - Scalability

Outline for This Session

- What is ETL?
- ETL Process
  - Kimball's 38 Step Method
  - Schmitz ETL Roadmap
- HW/DB Architecture Considerations
- ETL Tools
- Extraction
  - **Data Profiling**
  - Source-to-Target Mapping
- Case Study

Data Quality Issues

- Three drivers of need for data quality
  - “If only I could see the data, I could manage it better” heard from all levels
  - Integration of disparate data, often from distributed systems scattered worldwide
  - Increased demand for compliance
- Some data can be corrected in ETL process, but not all; some can only be corrected by source stewards
- If data is too dirty, DW project may have to be put on Hold.

About Data Profiling

- Each source must be profiled, that is, examined for data quality issues.
- This applies to each table or file and the database structure itself. This is the basis for cleaning data during the ETL process.
- Failure to identify these issues results in embarrassingly long and even fatal ETL processes and/or data warehouse efforts that fail due to inconsistent data.

About Data Profiling

- Generally there are more issues with older mainframe databases than with newer relational databases, but both often have many problems.
- Data sources can be profiled manually, usually using SQL queries or Excel AutoFilters.
- For larger or more complex sources, you may wish to employ a profiling tool such as Pentaho, Evoke, SAS DataFlux, Ascential Discovery, or others.

Common Data Quality Issues: Tables and Files

- Empty source tables (no data)
- Empty table columns
- Duplicate rows or natural keys
- Blank or Null values in text/character columns
- Inconsistent values in columns
- Invalid values in columns
- Unparsed fields such as City/State/Postal Code
- Codes about a single item from two or more sources that do not match, such as a customer or a product

Common Data Quality Issues: Database Structures

- Incorrect relationships between tables thereby violating 3NF or referential integrity
- Database structures that do not represent current business rules
- Data in two or more different files about a single subject (such as customers) where the unique identifiers for an individual instance (a customer) do not match

Some Options

- Examine manually, correct at source
- Write code to analyze automatically – may identify 75%
- Purchase package to analyze – 90%-95%

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

Data Cleansing

- Source systems contain “dirty data” that must be cleansed
- ETL software contains rudimentary data cleansing capabilities
- Specialized data cleansing software is often used. Important for performing name and address correction and householding functions
- Leading data cleansing vendors include Vality (Integrity), Harte-Hanks (Trillium), and Firstlogic (i.d.Centric)

Steps in Data Cleansing

- Parsing
- Correcting
- Standardizing
- Matching
- Consolidating

Parsing

- 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

- 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

- 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

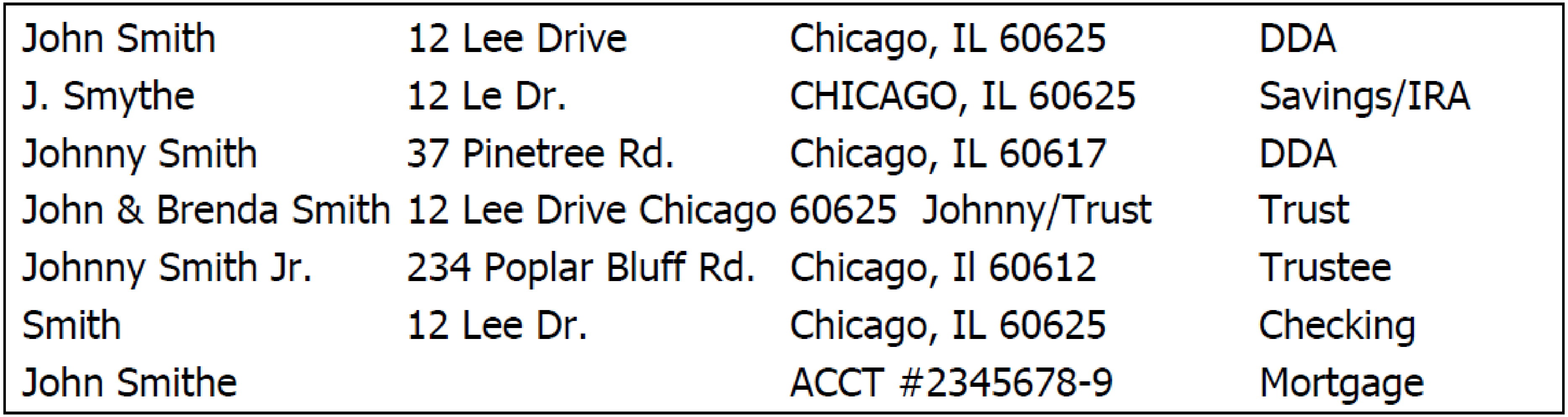
- Searching and matching records within and across the parsed, corrected and standardized data based on predefined business rules to eliminate duplications.
- Examples include identifying similar names and addresses.

Consolidating

- Analyzing and identifying relationships between matched records and consolidating/merging them into ONE representation.

Data Quality Exercise

- Assume the legacy data below is source data for a data warehouse you are building
  - What problems are there with this data?
  - What would you do to solve these problems?



John Smith	12 Lee Drive	Chicago, IL 60625	DDA
J. Smythe	12 Le Dr.	CHICAGO, IL 60625	Savings/IRA
Johnny Smith	37 Pinetree Rd.	Chicago, IL 60617	DDA
John & Brenda Smith	12 Lee Drive Chicago	60625 Johnny/Trust	Trust
Johnny Smith Jr.	234 Poplar Bluff Rd.	Chicago, Il 60612	Trustee
Smith	12 Lee Dr.	Chicago, IL 60625	Checking
John Smithe		ACCT #2345678-9	Mortgage

Outline for This Session

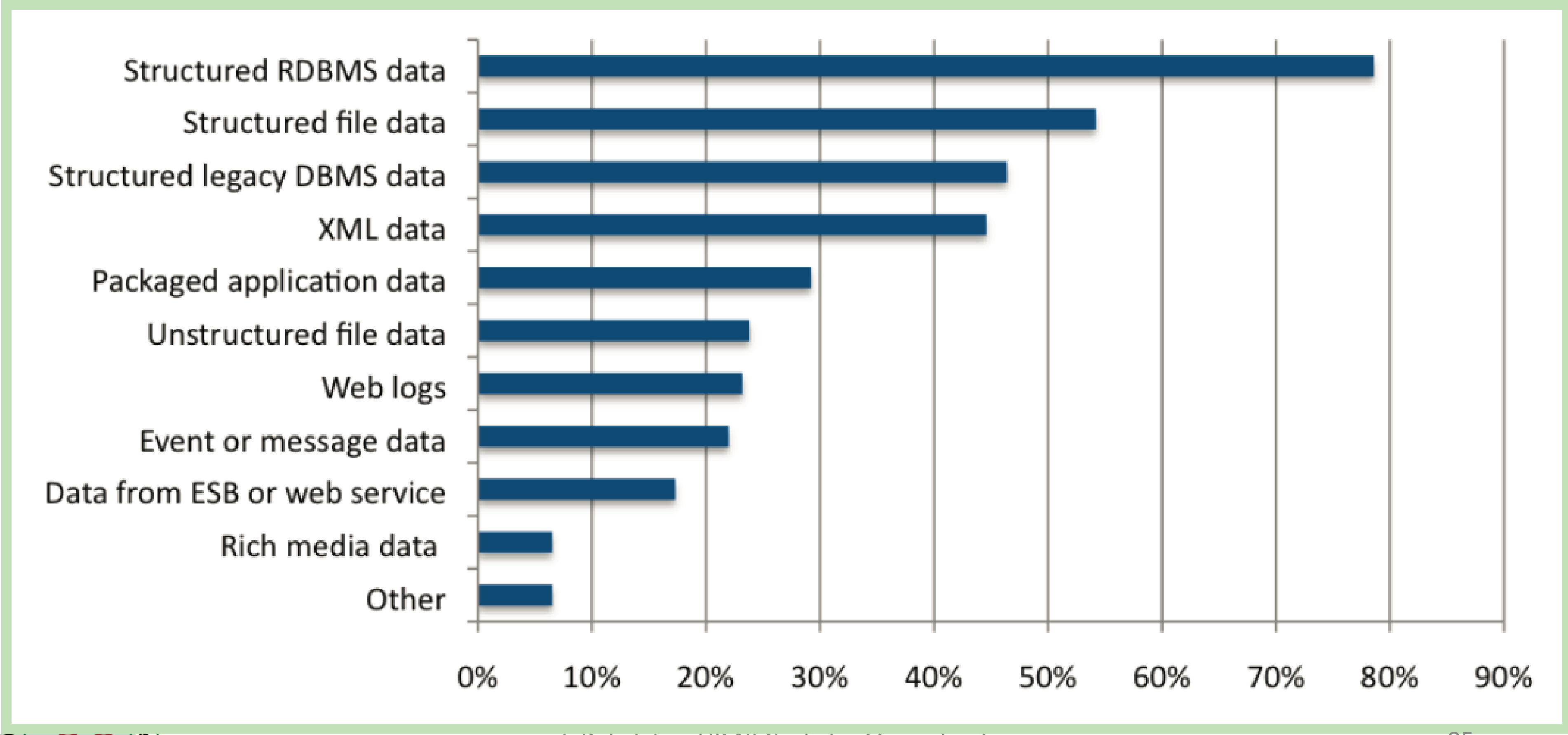
- What is ETL?
- ETL Process
  - Kimball's 34 Step Method
  - Schmitz ETL Roadmap
- HW/DB Architecture Considerations
- ETL Tools
- Extraction
  - Data Profiling
  - **Source-to-Target Mapping**
- Case Study

Some Examples of Data Source Origins

- Home grown operational applications
- Home grown departmental applications
- Home grown spreadsheets, Access databases, or personal databases
- External purchased data
- Industry applications
- Third party ERP, SCM, SFA, HR, and Web analytics applications

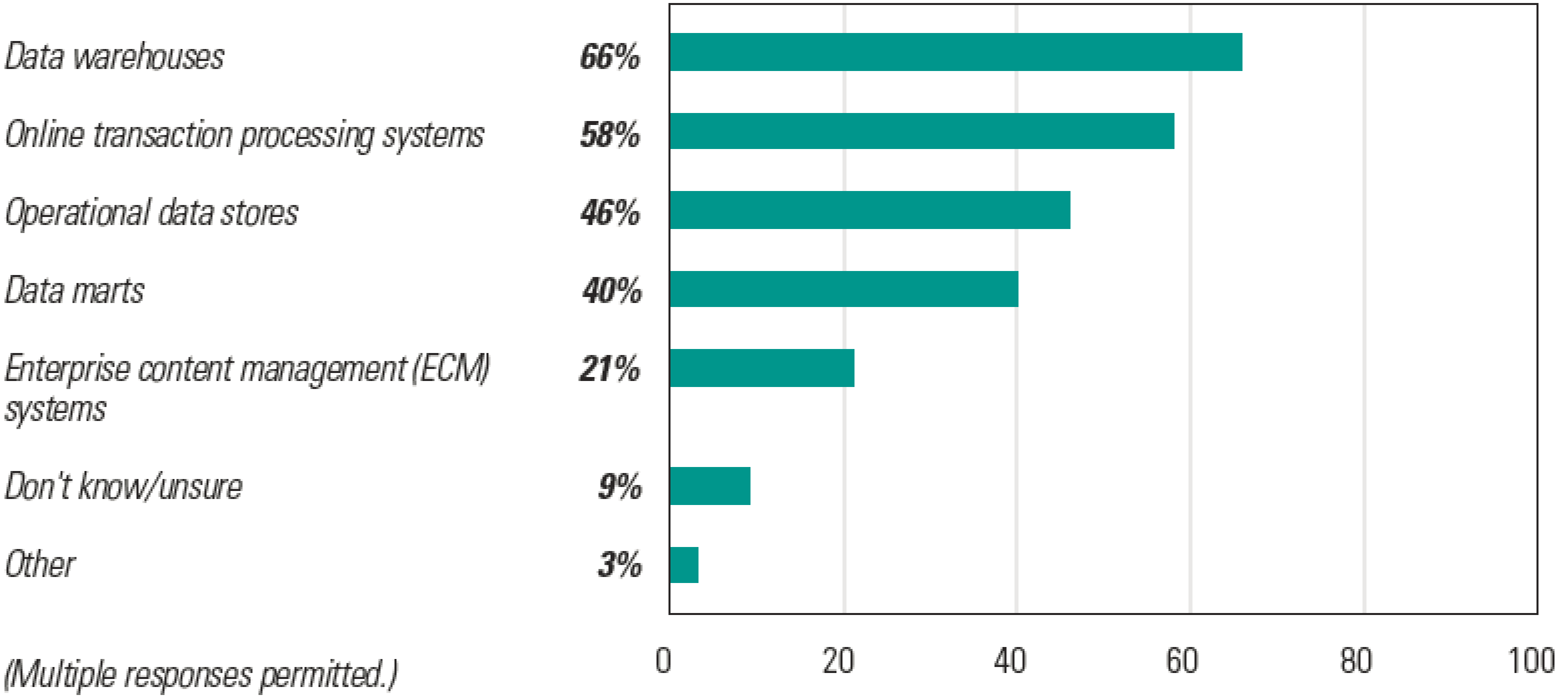
TechTarget, BI Research, IT Market Strategy (2010)

- Sources of Data



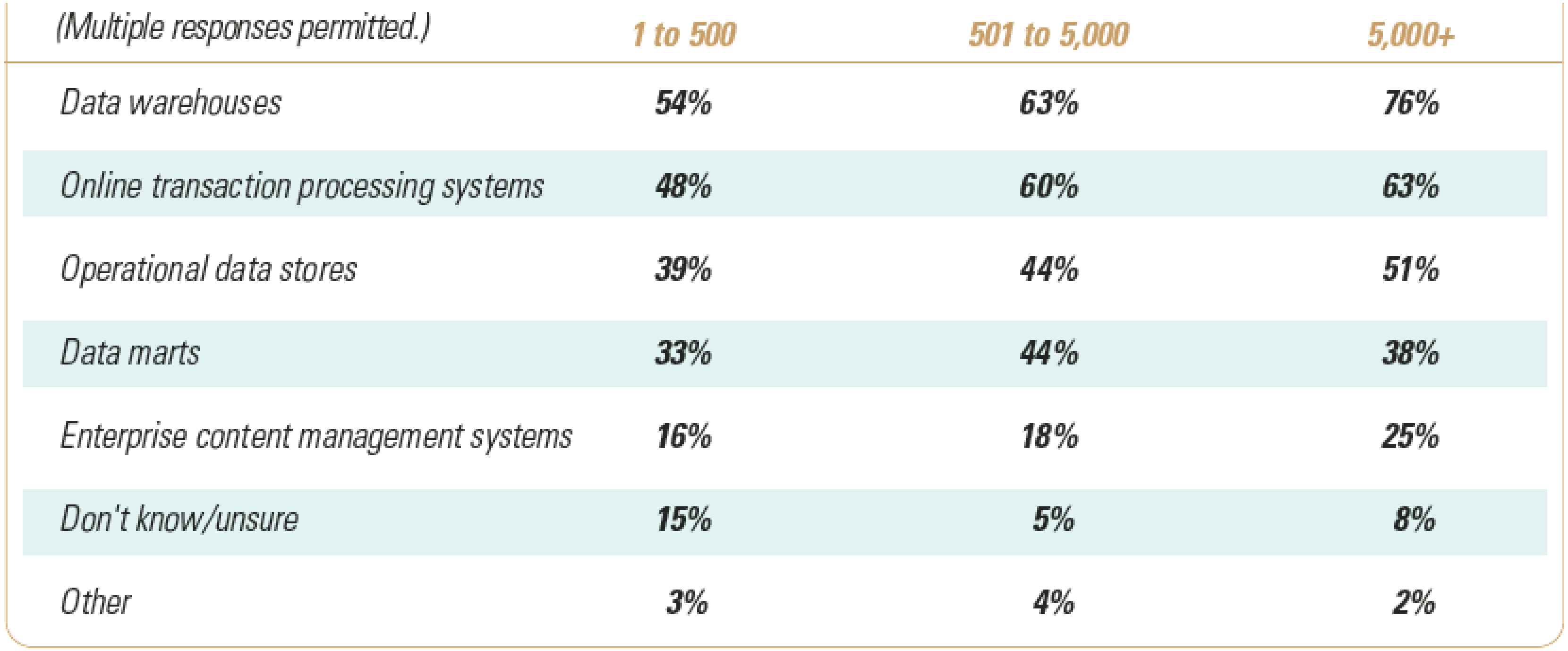
A New Dimension To Data Warehousing: 2011 Ioug Data Warehousing Survey

- Sources of Data

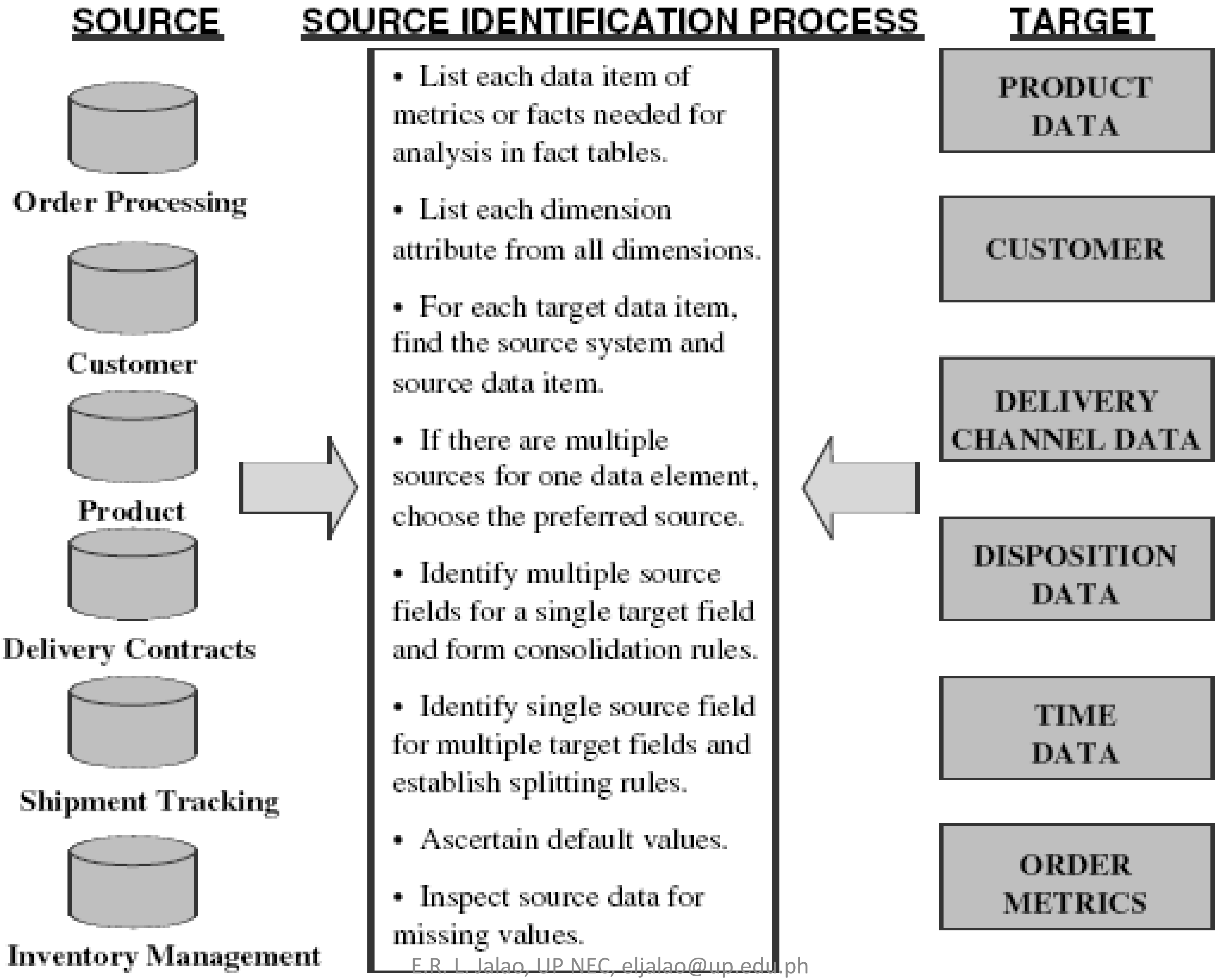


A New Dimension To Data Warehousing: 2011 Ioug Data Warehousing Survey

- Based on Total Number of Employees



(Multiple responses permitted.)	1 to 500	501 to 5,000	5,000+
Data warehouses	54%	63%	76%
Online transaction processing systems	48%	60%	63%
Operational data stores	39%	44%	51%
Data marts	33%	44%	38%
Enterprise content management systems	16%	18%	25%
Don't know/unsure	15%	5%	8%
Other	3%	4%	2%



SOURCE

Order Processing

Customer

Product

Delivery Contracts

Shipment Tracking

Inventory Management

SOURCE IDENTIFICATION PROCESS

- List each data item of metrics or facts needed for analysis in fact tables.
- List each dimension attribute from all dimensions.
- For each target data item, find the source system and source data item.
- If there are multiple sources for one data element, choose the preferred source.
- Identify multiple source fields for a single target field and form consolidation rules.
- Identify single source field for multiple target fields and establish splitting rules.
- Ascertain default values.
- Inspect source data for missing values.

TARGET

PRODUCT DATA

CUSTOMER

DELIVERY CHANNEL DATA

DISPOSITION DATA

TIME DATA

ORDER METRICS

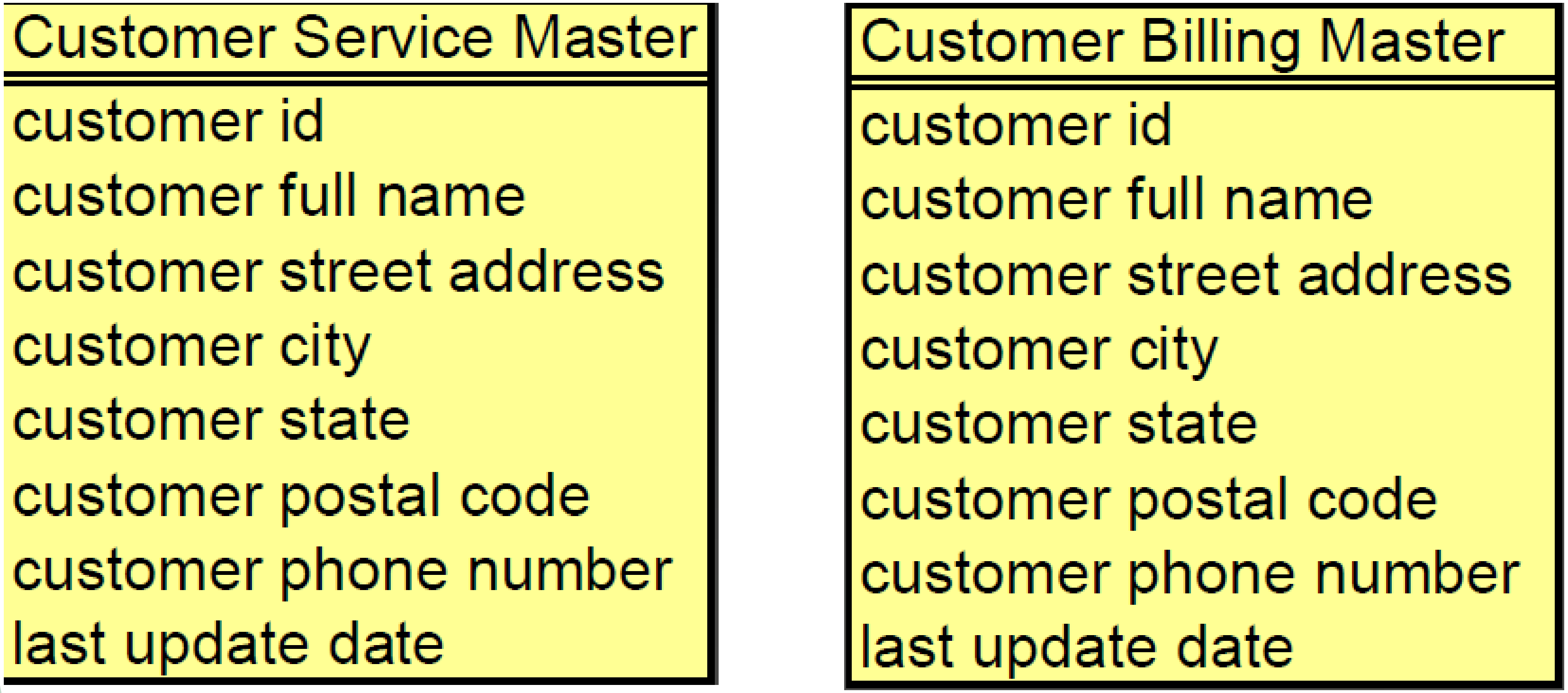
Full or Incremental Extracts

- Dimension changes may be hard to detect
  - no update timestamp or dimension transaction history
- Fact Table data – transaction and event based records are usually easy to detect and extract incrementally
  - Not always – log mining and replication may help here

Avoid a Complete Refresh of a Dimension

- You will usually lose some type of history
- Consumes a lot of extra computing resource
- Should only be used in special cases
- Usually only applicable in very small data warehouses where the operational system retains all history
- You will probably still lose some contextual history

Multiple Dimension Sources: Where to Get Data?



Customer Service Master
customer id
customer full name
customer street address
customer city
customer state
customer postal code
customer phone number
last update date

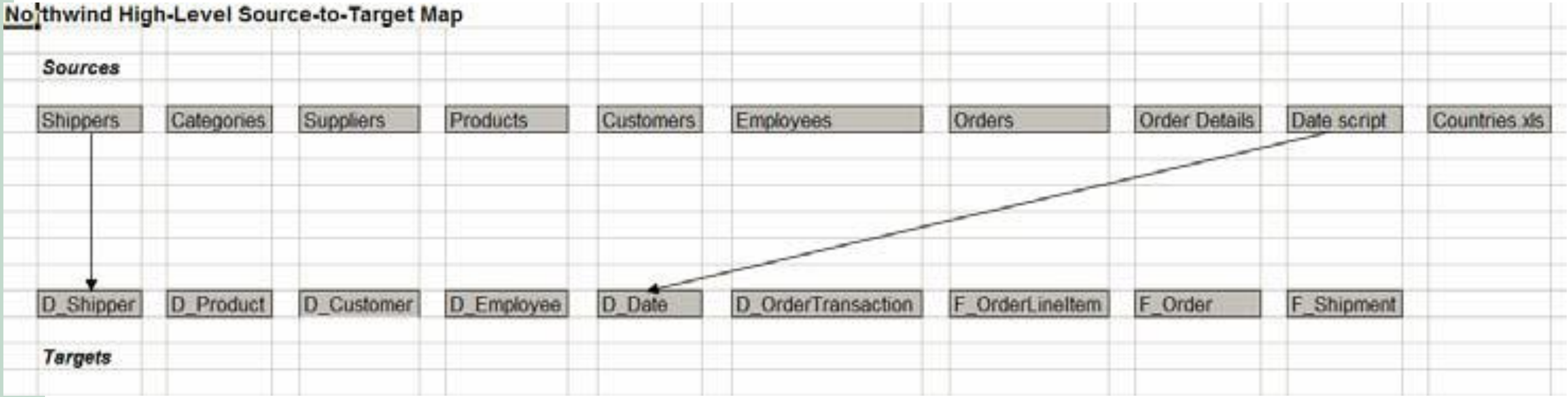
Customer Billing Master
customer id
customer full name
customer street address
customer city
customer state
customer postal code
customer phone number
last update date

Fact Table Sources

- Mostly transactional and event based tables or records
- New records are usually easy to detect
- Corrected or updated transaction records may not be easy to detect

High Level Source-to-Target Map (Template)

- See Exhibit 7



Northwind High-Level Source-to-Target Map

Sources

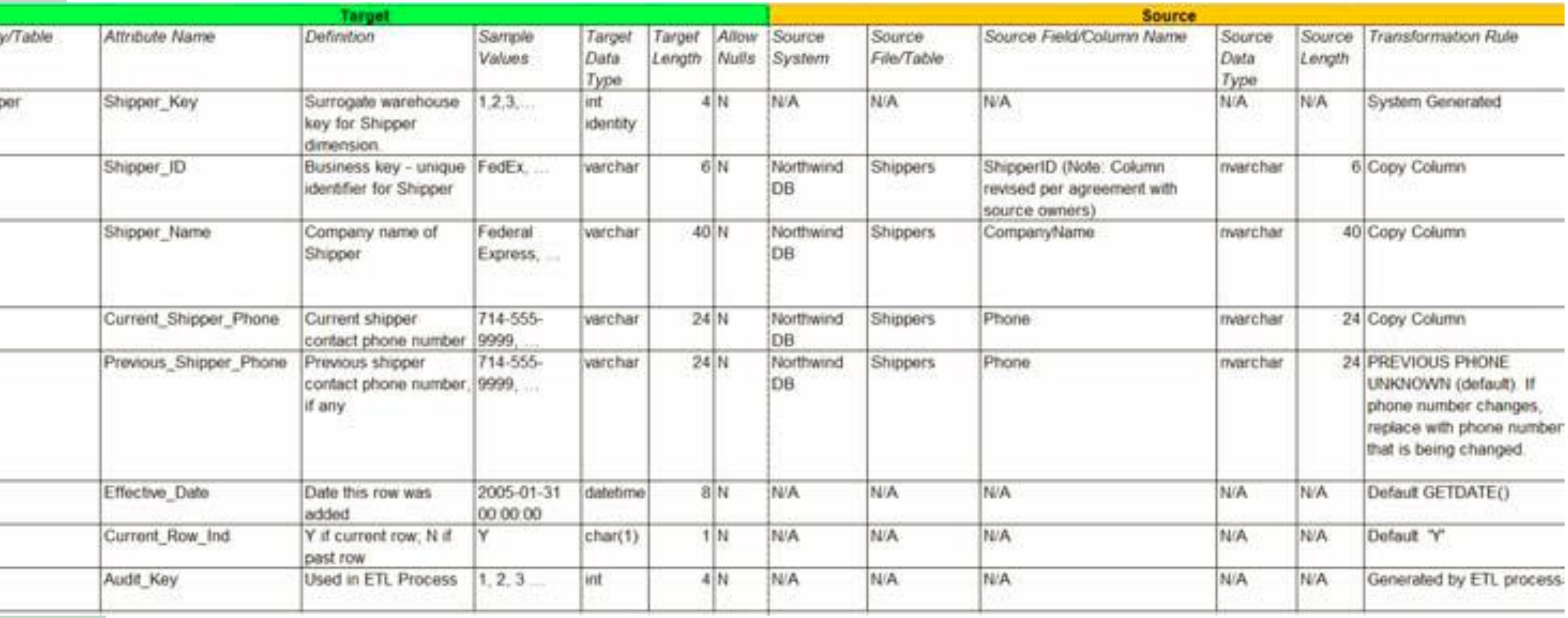
Shippers | Categories | Suppliers | Products | Customers | Employees | Orders | Order Details | Date script | Countries xls

D_Shipper | D_Product | D_Customer | D_Employee | D_Date | D_OrderTransaction | F_OrderLineItem | F_Order | F_Shipment

Targets

Develop Detailed Source-To-Target Map

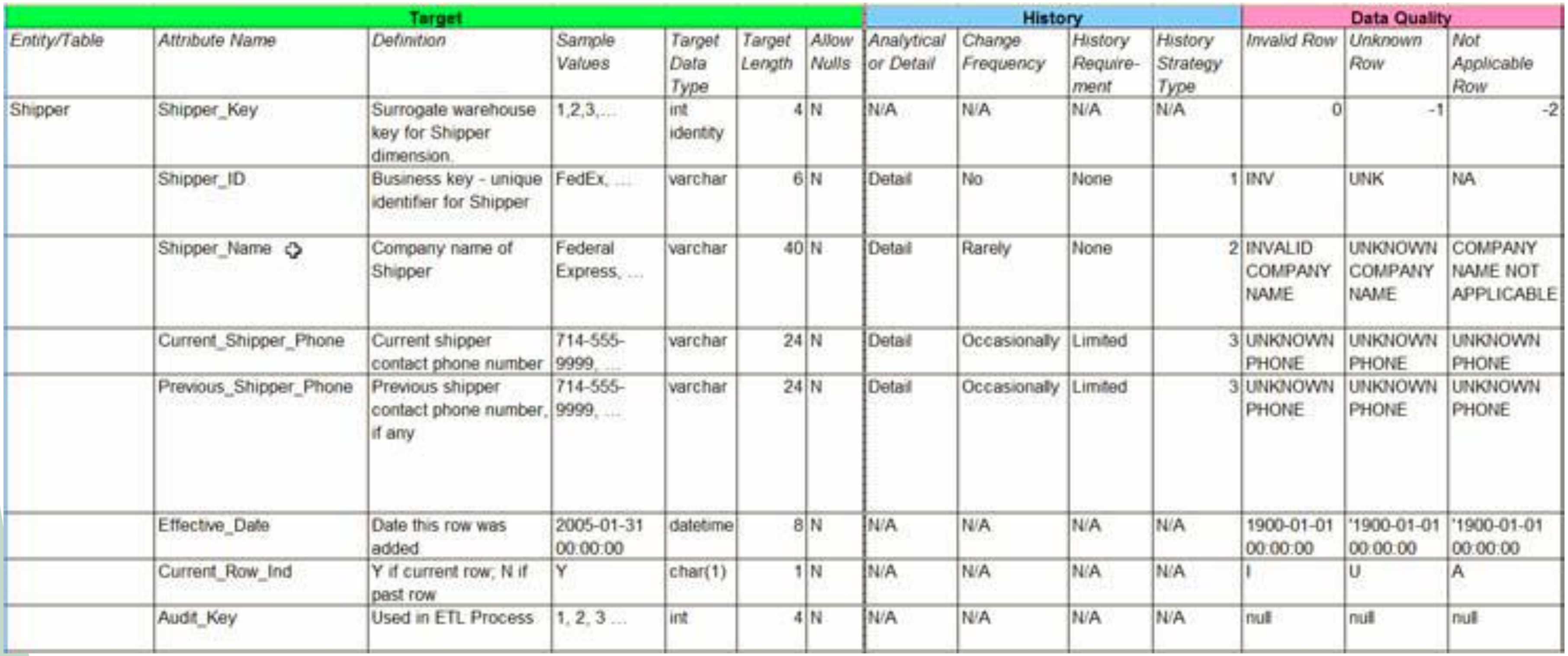
- See Exhibit 7



Target							Source					
y/Table	Attribute Name	Definition	Sample Values	Target Data Type	Target Length	Allow Nulls	Source System	Source File/Table	Source Field/Column Name	Source Data Type	Source Length	Transformation Rule
per	Shipper_Key	Surrogate warehouse key for Shipper dimension	1,2,3,...	int identity	4	N	N/A	N/A	N/A	N/A	N/A	System Generated
	Shipper_ID	Business key - unique identifier for Shipper	FedEx, ...	varchar	6	N	Northwind DB	Shippers	ShipperID (Note: Column revised per agreement with source owners)	nvarchar	6	Copy Column
	Shipper_Name	Company name of Shipper	Federal Express, ...	varchar	40	N	Northwind DB	Shippers	CompanyName	nvarchar	40	Copy Column
	Current_Shipper_Phone	Current shipper contact phone number	714-555-9999, ...	varchar	24	N	Northwind DB	Shippers	Phone	nvarchar	24	Copy Column
	Previous_Shipper_Phone	Previous shipper contact phone number, if any	714-555-9999, ...	varchar	24	N	Northwind DB	Shippers	Phone	nvarchar	24	PREVIOUS PHONE UNKNOWN (default). If phone number changes, replace with phone number that is being changed.
	Effective_Date	Date this row was added	2005-01-31 00:00:00	datetime	8	N	N/A	N/A	N/A	N/A	N/A	Default GETDATE()
	Current_Row_Ind	Y if current row; N if past row	Y	char(1)	1	N	N/A	N/A	N/A	N/A	N/A	Default 'Y'
	Audit_Key	Used in ETL Process	1, 2, 3 ...	int	4	N	N/A	N/A	N/A	N/A	N/A	Generated by ETL process.

Detailed Source-to-Target Map: Target, History, Data Quality

- See Exhibit 7



Target							History				Data Quality		
Entity/Table	*Attribute Name*	*Definition*	*Sample Values*	*Target Data Type*	*Target Length*	*Allow Nulls*	*Analytical or Detail*	*Change Frequency*	*History Require-ment*	*History Strategy Type*	*Invalid Row*	*Unknown Row*	*Not Applicable Row*
Shipper	Shipper_Key	Surrogate warehouse key for Shipper dimension.	1,2,3,...	int identity	4	N	N/A	N/A	N/A	N/A	0	-1	-2
	Shipper_ID	Business key - unique identifier for Shipper	FedEx, ...	varchar	6	N	Detail	No	None	1	INV	UNK	NA
	Shipper_Name	Company name of Shipper	Federal Express, ...	varchar	40	N	Detail	Rarely	None	2	INVALID COMPANY NAME	UNKNOWN COMPANY NAME	COMPANY NAME NOT APPLICABLE
	Current_Shipper_Phone	Current shipper contact phone number	714-555-9999, ...	varchar	24	N	Detail	Occasionally	Limited	3	UNKNOWN PHONE	UNKNOWN PHONE	UNKNOWN PHONE
	Previous_Shipper_Phone	Previous shipper contact phone number, if any	714-555-9999, ...	varchar	24	N	Detail	Occasionally	Limited	3	UNKNOWN PHONE	UNKNOWN PHONE	UNKNOWN PHONE
	Effective_Date	Date this row was added	2005-01-31 00:00:00	datetime	8	N	N/A	N/A	N/A	N/A	1900-01-01 00:00:00	'1900-01-01 00:00:00	'1900-01-01 00:00:00
	Current_Row_Ind	Y if current row; N if past row	Y	char(1)	1	N	N/A	N/A	N/A	N/A	I	U	A
	Audit_Key	Used in ETL Process	1, 2, 3 ...	int	4	N	N/A	N/A	N/A	N/A	null	null	null

Outline for This Session

- What is ETL?
- ETL Process
  - Kimball's 34 Step Method
  - Schmitz ETL Roadmap
- HW/DB Architecture Considerations
- ETL Tools
- Extraction
  - Data Profiling
  - Source-to-Target Mapping
- **Case Study**

Case Study: Data Profiling and Source-To-Target Mapping

- Northwind Database

References

- Kimball, Ralph, Margy Ross, Warren Thornthwaite, Joy Mundy, and Bob Becker, *The Data Warehouse Life Cycle Toolkit, Second Edition*, Wiley, 2008, ISBN 978-0-470-14977-5
- Schmitz, Michael D. UCI Irvine Data Warehousing Notes (2014), High Performance Data Warehousing
- http://hornad.fei.tuke.sk/~genci/Vyucba/OOBDS/Archiv/Prednasky/2008/03-ETL-081028-2055.ppt
- Simon, Alan. CIS 391 PPT Slides
- Jeltema ,Bernie, UCI Irvine Data Warehousing Notes (2014), Strategic Frameworks, Inc.