

An Event-Based Near Real-Time Data Integration Architecture

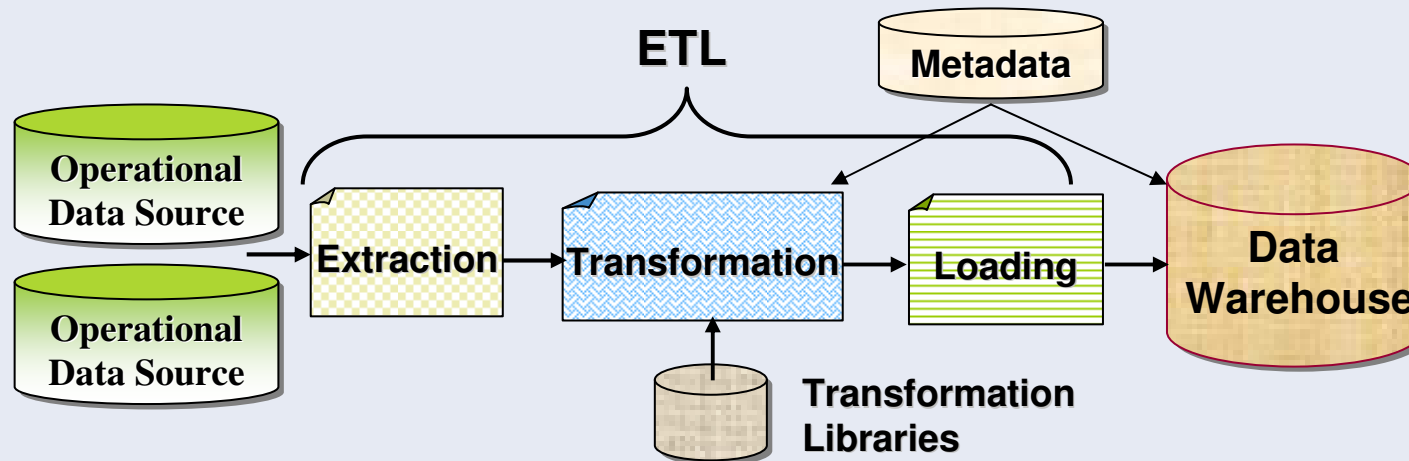
**Muhammad Asif Naeem,
Gill Dobbie, Gerald Weber**

**Department of Computer Science
The University of Auckland, New Zealand**



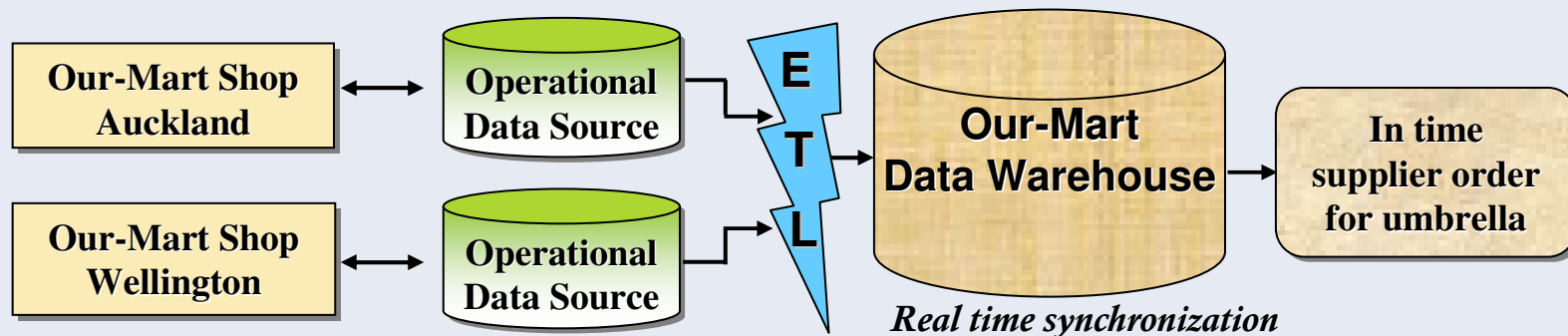
Data Warehouse using Proprietary ETL Tools

- Traditional data warehouses do not have up-to-the-minute data.
- Updates on nightly or even weekly basis.
- Operational systems have to go offline during extraction process.
- Proprietary ETL tools work on pull technology principle.



The Goal: Near Real Time DW

- **Company has two branches at different locations.**
 - Our-Mart Shop in Auckland
 - Our-Mart Shop in Wellington
- **Information is updated on a near real time basis.**
- **Fresh data will be available for analysis to make business related decisions.**
 - E.g. In-time supplier order of umbrellas from analyzing buying trends in data warehouse



Kinds of Data

➤ Transactional Data

- ▮ Data relating to the day-to-day transactions
e.g. shopping cart data.
- ▮ Orders of Magnitude more than Master Data
- ▮ Consolidates in the **Fact Table** in DWH using **Star Schema**

➤ Master Data

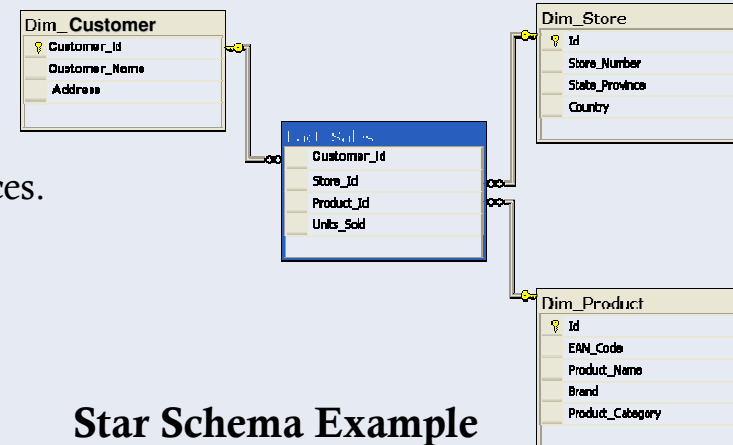
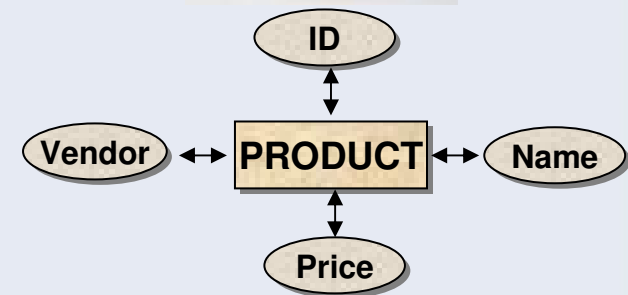
- ▮ Element along with their associated attributes
e.g. Customer, Product, Store etc.
- ▮ Typically slowly changing
- ▮ Consolidated in **Dimension Tables** in DWH

➤ Other Data

- ▮ Aggregation of transactional data
e.g. items on hand, seats available, and account balances.

➤ Metadata

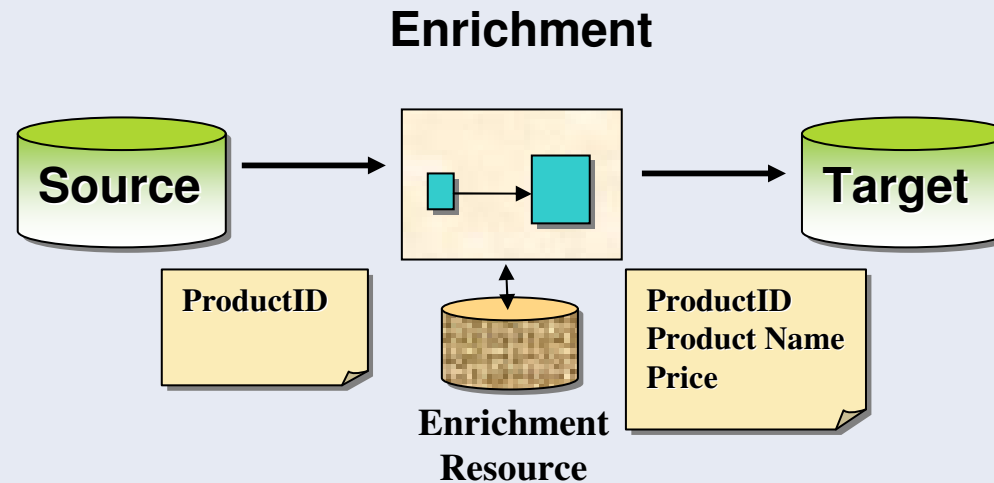
- ▮ Representing the Information System itself
e.g. type and size of attributes, constraints, etc.



Star Schema Example

Content Enrichment

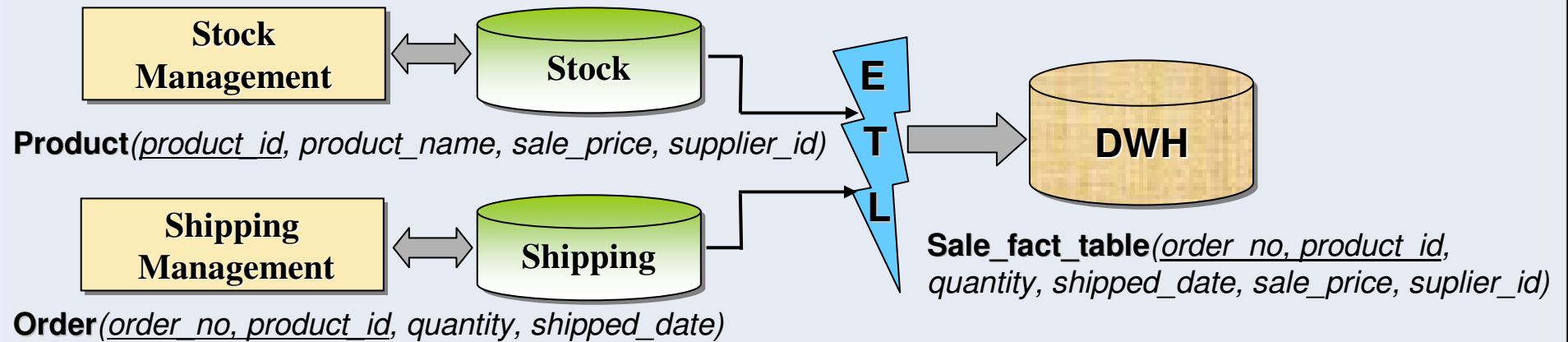
- A special case of data transformation in which some additional information is injected into the current message.



- Important example: surrogate key insertion

Problems with Traditional ETL

Example of Our-Mart Inventory System without enrichment and data distinction



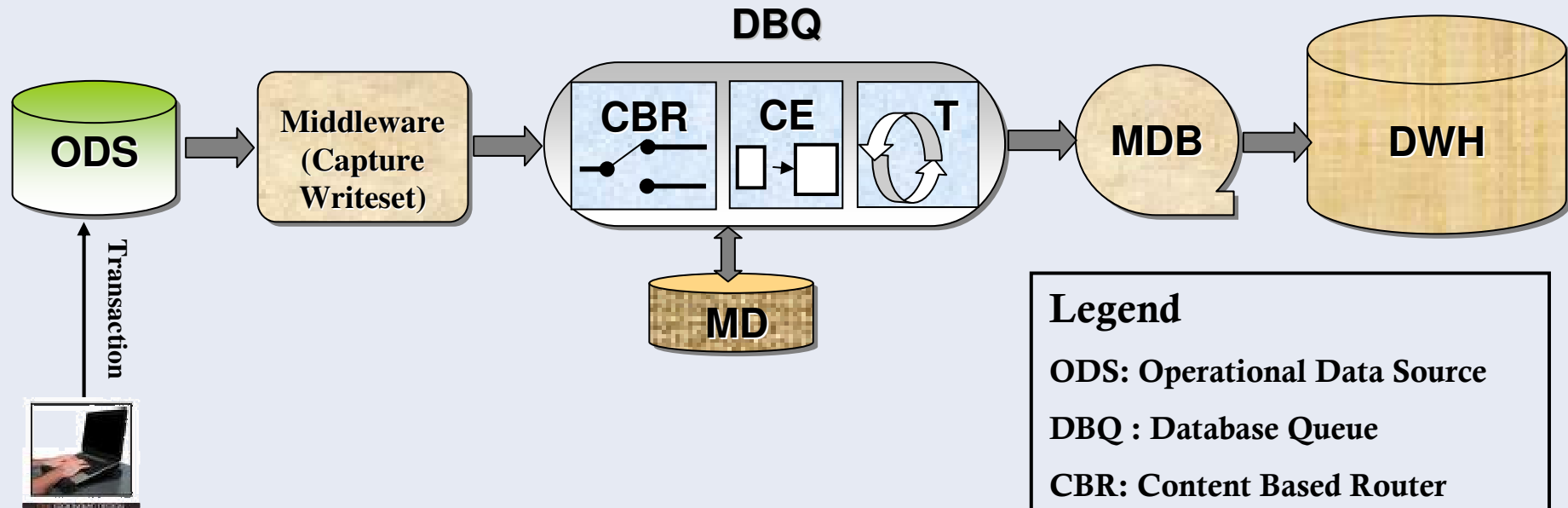
Observations

- ▀ Extraction of Master Data on each data loading window
- ▀ Might be feasible in batch processing system
- ▀ Not appropriate in real-time scenario

Proposed Solution

- **Identification of master and transaction data**
- **Enrichment with master data on-the-fly**
- **Event driven database queue to perform ETL functions**
- **Lightweight coupling of data sources and sinks.**
- **Standard Enterprise Service Bus architecture**

Proposed Architecture



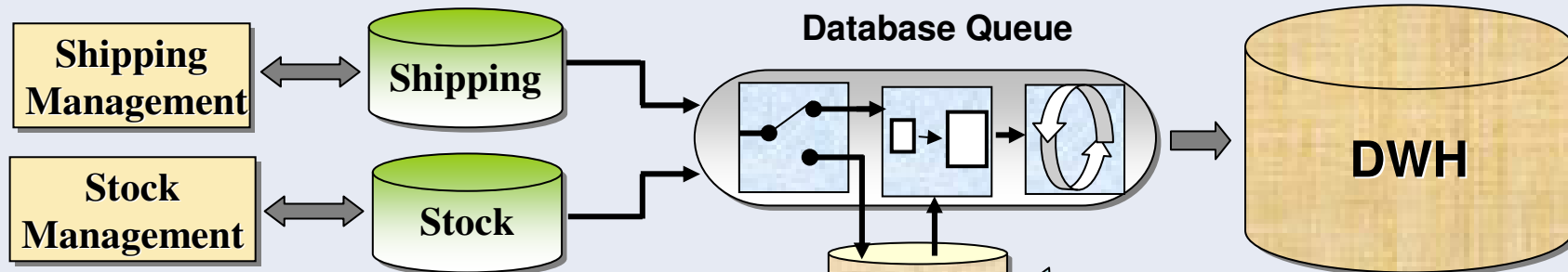
Legend

- ODS: Operational Data Source
- DBQ : Database Queue
- CBR: Content Based Router
- CE: Content Enrichment
- T: Transformation
- MD: Master Data
- MDB: Message Driven Bean
- DWH: Data Warehouse

Reconsider the Our-Mart Inventory Example

Example of Our-Mart inventory system with enrichment and data distinction

Order(*order_no*, *product_id*, *quantity*, *shipped_date*)



Product(*product_id*, *product_name*, *sale_price*, *supplier_id*)

Sale_fact_table(*order_no*, *product_id*, *quantity*, *shipped_date*, *sale_price*, *supplier_id*)

Merits

- No need to transfer Master Data for each transaction level.
- Transfer only if it is changed.
- Only updates will be transferred.

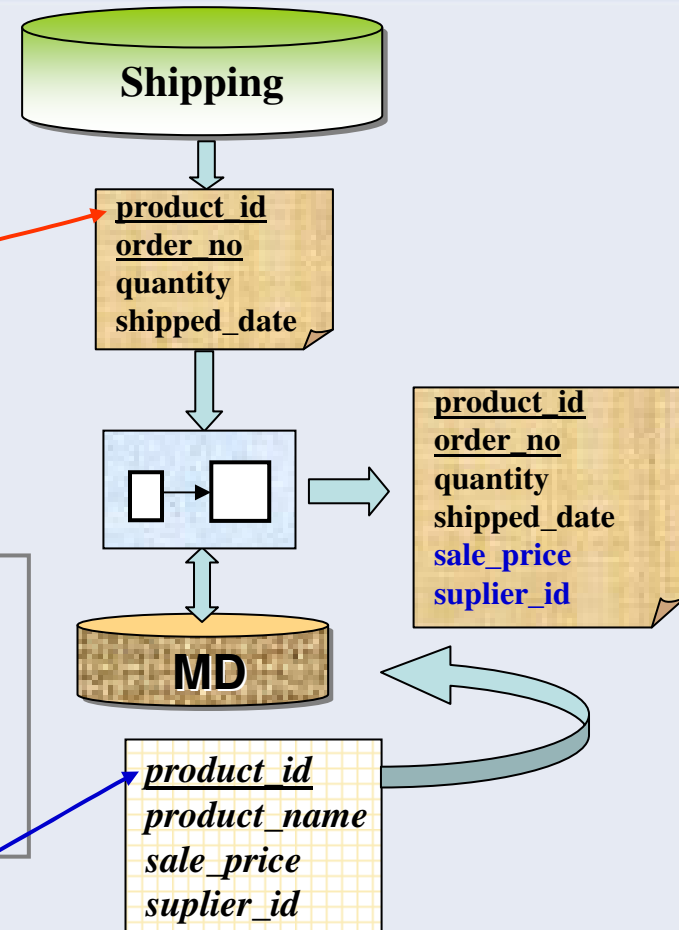
Reconsider the Our-Mart Inventory Example

Enrichment of attributes (*sale_price*, *supplier_id*)

Using Index Loop Join

```
APPEND Message
AS SELECT sale_price, supplier_id
FROM Product_master
WHERE product_id = Product_master.product_id;
```

Index Loop Join



Features of Database Queue

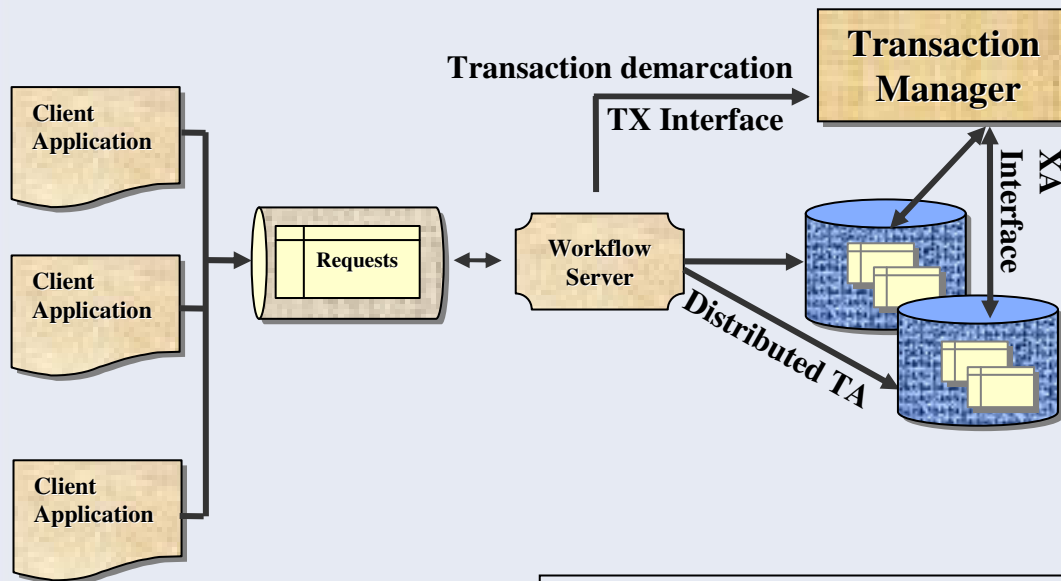
✦ **Features of the message queue**

- ▀ Message will be processed once it is placed in the queue (Transactionality).
- ▀ Multiple workflow servers can work on same queue.
- ▀ Feasible for workflow disconnected applications
- ▀ No chance of data/request loss

✦ **Natural solution: queues are databases**

- ▀ All features of database are available to use.
- ▀ No need to write additional code for create(), enqueue(), dequeue(), poll() and destroy().
- ▀ ACID properties are handled by DBMS.

Unsing Transaction Services for higher QoS



X/Open Interfaces:
TX: Extended Transaction
XA: Extended Architecture
TA: Transaction Architecture

```

Interface (3) [Java Application] C:\Program Files\Java\jre1.5.0_14\bin\javaw.exe (Jul 6, 2008 4:17:54 PM)
Message Queue is Successfully Connected
Transaction is propogated to Message Queue successfully
From: 100      To: 101      Amount: 10.0
Src account:942.0
Update in source is done successfully
Update in target is done successfully
Src account:932.0
Source Insertion successfully Executed
Trgt account:3558.0
Target Insertion successfully Executed

Transaction is propogated to Message Queue successfully
From: 100      To: 101      Amount: 15.0
Src account:932.0
Update in source is done successfully
Update in target is done successfully
Src account:917.0
Source Insertion successfully Executed
Trgt account:3573.0
Target Insertion successfully Executed

Transaction is propogated to Message Queue successfully
From: 100      To: 101      Amount: 4.0
Src account:917.0
Update in source is done successfully
Update in target is done successfully
Src account:913.0
Source Insertion successfully Executed
Trgt account:3577.0
Target Insertion successfully Executed
    
```

Related work

- Most research has been done related to proprietary data warehouse. (Galhardas H., Florescu D., 2000),(Wilburt Labio, Jun Yang, 2000),(Wilburt Labio, Hector Garcia-Molina, 1996, Wilburt Labio, Janet L.,2000, Vijayshankar Raman, Joseph M. , 2001).
- NCR's Teradata introduced the concept of Active Data Warehousing (ADWH) in 2002 with different continuous data integration utilities (e.g. FastLoad, MultiLoad).
- Bruckner, R. used Event-Condition-Action (ECA) rules and Microsoft Message Queue (MSMQ) to implement the (ADWH), 2002.
- Araque F. proposed the architecture of Real-time Data Warehousing for Web Repositories, 2003.
- Ian Gorton, Anna Liu proposed an Enterprise Application Integration Architecture (EAI) using hub and spoke (2004).
- Alexandros Karakasidis, introduce the concept of queue networks to implement the near real-time ETL(2005).
- Neoklis Polyzotics, Spiros Shiadopoulos proposed an algorithm (MESHJOIN) to support continuous streaming in ADWH, 2007.



Conclusion

+ Novel Features

- Distinction between Master and Transactional Data
- Complicated transformations such as Content Enrichment
- Uses push or event-driven principle

+ Advantages

- Increase the freshness level of data warehouse
- Availability of up-to-date information to make good decisions
- Only updates are transferred instead of whole window
- No need for operational data sources to go offline
- More reliable because of database queue



Future Directions

➤ **Writeset Propagation**

- ▮ Extraction of writeset at transactional level
- ▮ Transmission of writeset in database queue

➤ **Identification of Master Data**

- ▮ Separation of Master Data and Transactional Data
- ▮ Loading Master Data into separate repository

➤ **Content Enrichment**

- ▮ Integration with content-based routing enrichment
- ▮ Enrichment with Master Data using Index Loop Join

➤ **Transformation**

- ▮ Data cleaning
- ▮ Preparation of data w.r.t format & data type
- ▮ Source to target mapping

➤ **Loading**

➤ **Performance Evaluation**



