



Data Warehousing and Multidimensional Data Model

Presentation
Prof Anita Wasilewska
SUNY Stony Brook

Presented By:
Nidhi Pai (106674895)
Ujesh Nambiar (106866423)

References

- <http://infolab.stanford.edu/warehousing/>
- <http://www.kimballgroup.com/>
- http://en.wikipedia.org/wiki/Data_warehouse
- <http://infolab.stanford.edu/infoseminar/Archive/FallY99/>
- www.cacs.louisiana.edu/~cmps566/Kishore-jaladi-DW.ppt
- http://en.wikipedia.org/wiki/Data_model
- <http://infolab.stanford.edu/infoseminar/Archive/FallY99/russakovskii-slides/sld007.htm>
- www.it.kmitl.ac.th/~pattarachai/DB/PDF/P2DataWarehouse.pdf
- wwwbayer.in.tum.de/lehre/SS2002/DAWA-bayer/DWH-Ch3-1.ppt
- <http://www.stanford.edu/dept/itss/docs/oracle/10g/olap.101/b10333/multimodel.htm>
- Data Mining Concepts and Techniques – Jiawei Han and Micheline Kamber – Book Slides

Overview

- What is a data warehouse?
- History of Data Warehouse
- Components of Data Warehouse
- Why a warehouse?
- OLTP vs OLAP



Definitions of a Data Warehouse

“A **subject-oriented, integrated, time-variant** and **non-volatile** collection of data in support of management's decision making process”

- W.H. Inmon

“A **copy of transaction data**, specifically structured for **query and analysis**”

- Ralph Kimball

Understanding the term Data Warehousing

- **Subject Oriented:**

Data that gives information about a particular subject instead of about a company's ongoing operations.

- **Integrated:**

Data that is gathered into the data warehouse from a variety of sources and merged into a coherent whole.

- **Time-variant:**

All data in the data warehouse is identified with a particular time period.

- **Non-volatile**

Data is stable in a data warehouse. More data is added but data is never removed. This enables management to gain a consistent picture of the business.

History of Data Warehouse

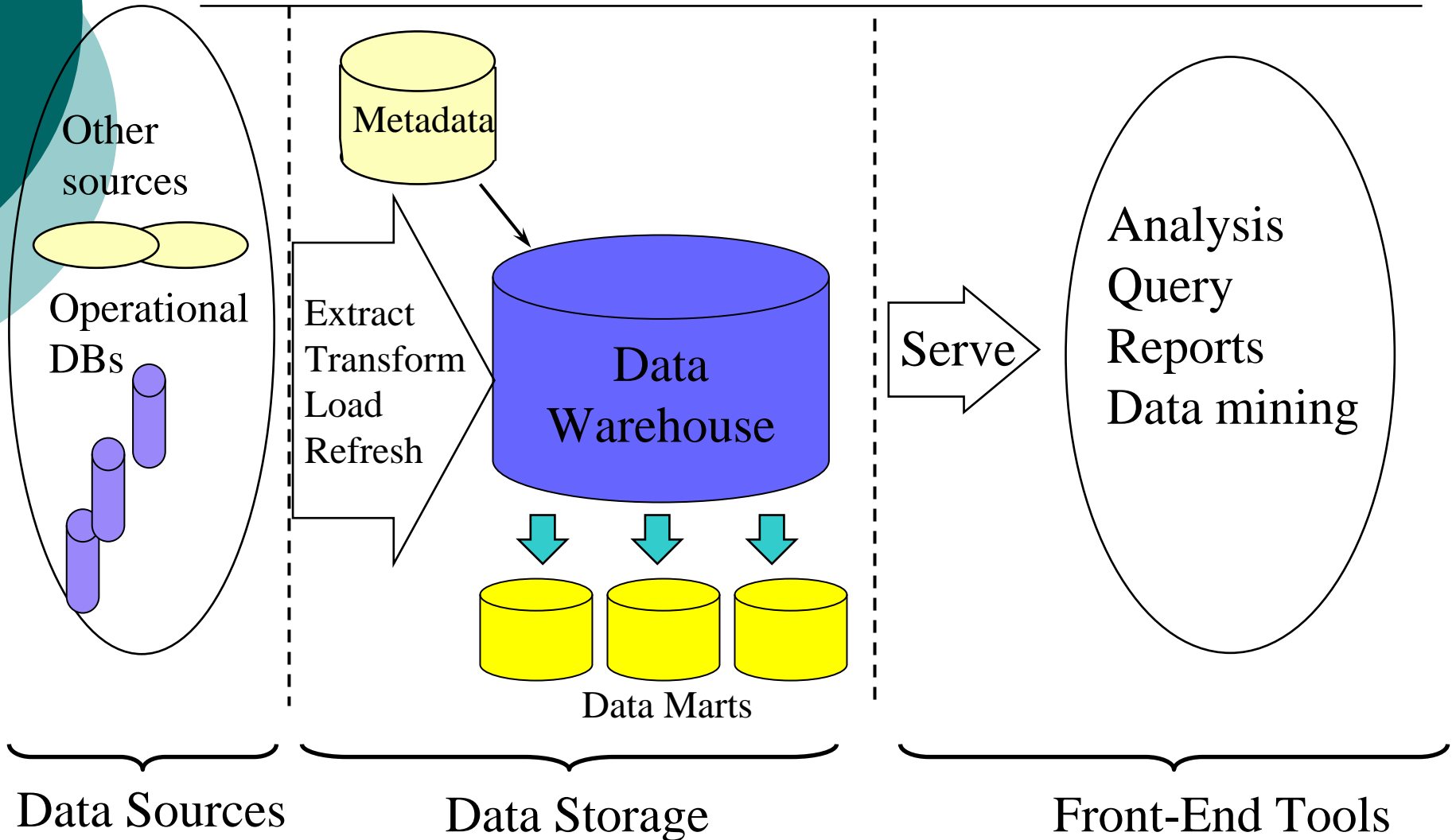
- The concept of data warehousing dates back to the late 1980s when IBM researchers Barry Devlin and Paul Murphy developed the "business data warehouse".
[Intended to provide an architectural model for the flow of data from operational systems to decision support environments]
Key developments in early years of data warehousing were:
 - 1960s - General Mills and Dartmouth College, in a joint research project, develop the terms *dimensions* and *facts*.
 - 1970s - ACNielsen and IRI provide dimensional data marts for retail sales.
 - 1983 – Teradata introduces a database management system specifically designed for decision support.
 - 1995 - The Data Warehousing Institute, a for-profit organization that promotes data warehousing, is founded.
 - 1997 - Oracle 8, with support for star queries, is released.



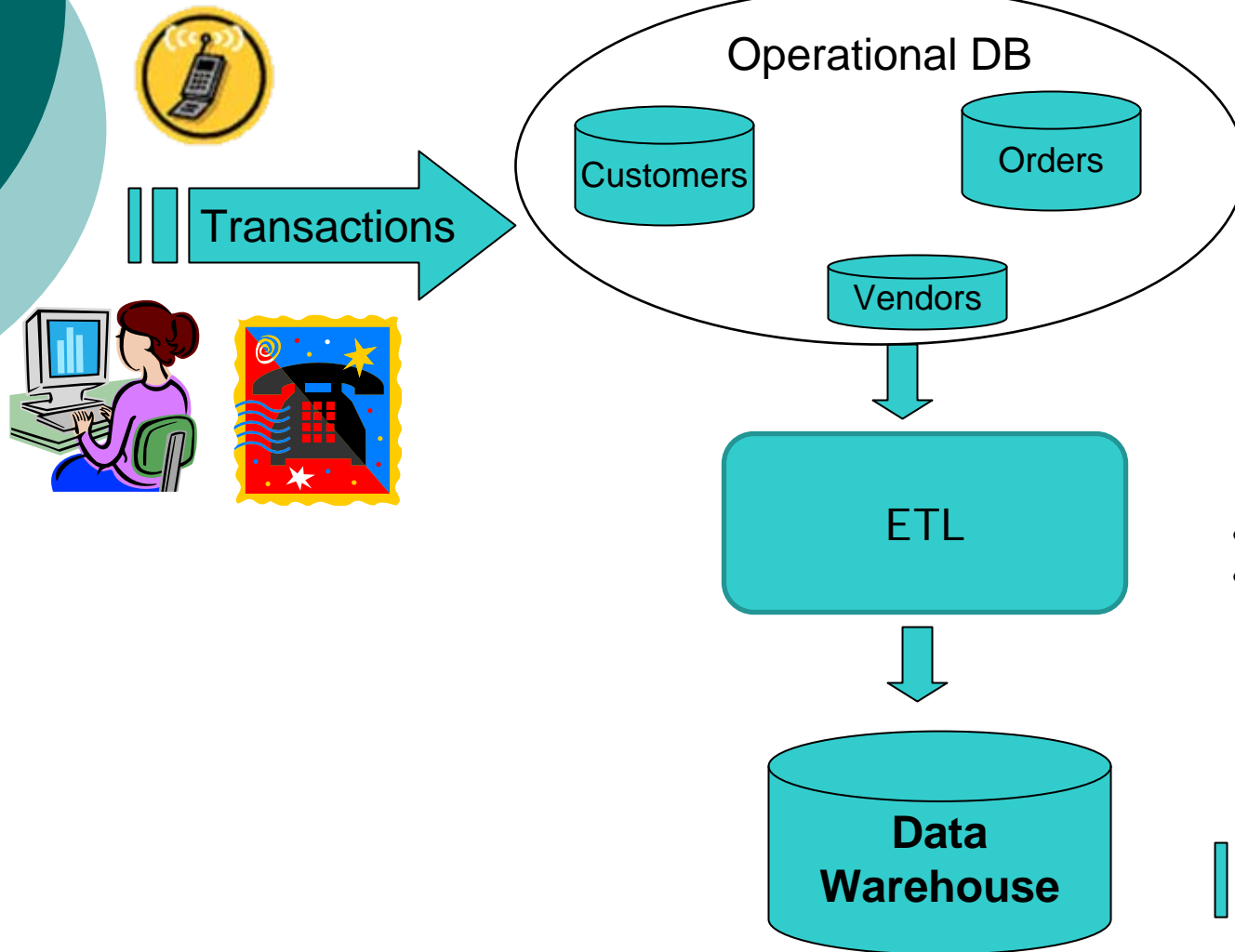
Components of Data Warehouse

- Input sources: e.g. Operational Database, RDBMS, Flat files
- ETL
- Metadata
- Data Warehouse Database
- Data Marts
- Information Delivery System

Data Warehouse Architecture



Data Warehouse



- Data Miners:
- “Farmers” – they know
 - “Explorers” - unpredictable



Why Separate Data Warehouse?

- High performance for both systems
 - DBMS - Tuned for Online Transaction Processing Systems
 - Warehouse - Tuned for Online Analytical Processing systems involving complex OLAP queries
 - Processing OLAP queries would degrade DBMS performance of operational tasks.
- Decision support requires historical data which operational Databases do not typically maintain.
- Decision Support requires consolidation of data from heterogeneous sources.
- **Solution**
 - To maintain separate database systems (Data Warehouse database) which support special primitives and structures suitable to store, access and process OLAP specific data.

OLTP vs. OLAP

	OLTP	OLAP
User	Clerk, IT Professional	Knowledge worker
Function	Day to day operations	Decision support
Schema	Application-oriented (E-R based)	Subject-oriented (Star, snowflake)
Data	Current, Isolated	Historical, Consolidated
View	Detailed, Flat relational	Summarized, Multidimensional
Unit of work	Short, Simple transaction	Complex query
Db size	100 MB-GB	100GB-TB

Advantages

- A data warehouse provides a common data model for all data of interest regardless of the data's source.
- Prior to loading data into the data warehouse, inconsistencies are identified and resolved. This greatly simplifies reporting and analysis.
- Information in the data warehouse is under the control of data warehouse users so that, even if the source system data is purged over time, the information in the warehouse can be stored safely for extended periods of time.
- Data warehouses can work in conjunction with and, hence, enhance the value of operational business applications, notably customer relationship management (CRM) systems.

Disadvantages

- Because data must be extracted, transformed and loaded into the warehouse, there is an element of **latency** in data warehouse data.
- Over their life, data warehouses can have **high costs**. The data warehouse is usually not static. Maintenance costs are high.
- Data warehouses can get **outdated** relatively quickly. There is a cost of delivering suboptimal information to the organization.

Overview

- Multidimensional Data Model
- Dimensions and Facts
- Star Schema
- Snow Flake schema
- Fact Constellation

Multidimensional Data Model

- **Data Model:**

A data model^[1] in software engineering is an abstract model that describes how data is represented and accessed.

- **Multidimensional Data Model:**

- A Data warehouse is based on multidimensional data model, which views data in the form of a data cube.

- [2]The multidimensional data model represents the data of interest by means of multi dimensions.

For e.g. Kodak Management wants to view the report of sales (revenue) of different Digital Cameras per region per year.

Dimensions – Product Type, City, Year

Data of Interest (Fact or Measure) – Revenue in Dollars

[1] – http://en.wikipedia.org/wiki/Data_model

[2] - <http://infolab.stanford.edu/infoseminar/Archive/FallY99/russakovskii-slides/sld007.htm>

Multidimensional Representation of 2-dim Data as 2-D Matrix

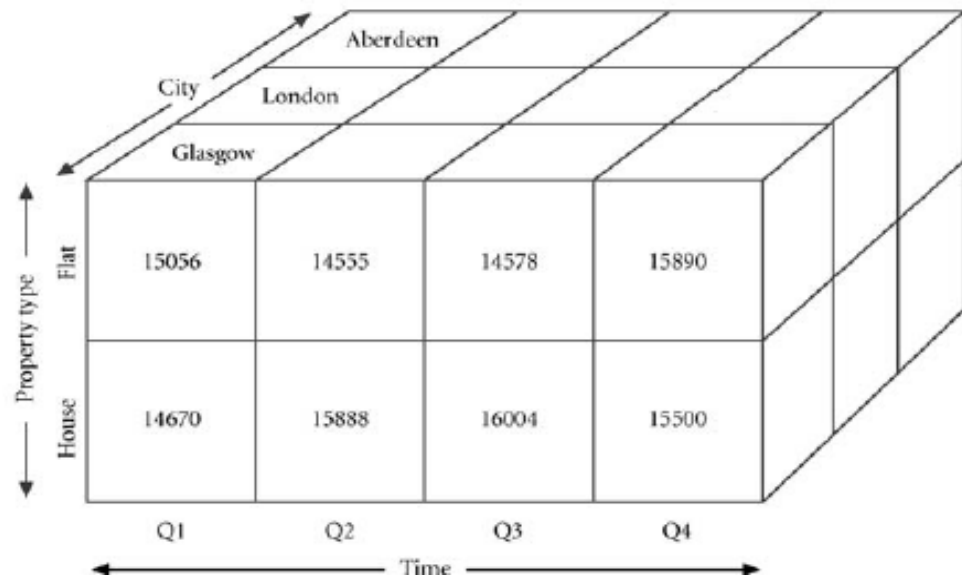
City	Time	Total Revenue
Glasgow	Q1	29726
Glasgow	Q2	30443
Glasgow	Q3	30582
Glasgow	Q4	31390
London	Q1	43555
London	Q2	48244
London	Q3	56222
London	Q4	45632
Aberdeen	Q1	53210
Aberdeen	Q2	34567
Aberdeen	Q3	45677
Aberdeen	Q4	50056
.....
.....

		City			
		Glasgow	London	Aberdeen
Time	Quarter				
	Q1	29726	43555	53210
	Q2	30443	48244	34567
	Q3	30582	56222	45677
	Q4	31390	45632	50056

Dimensions: City, Time (so 2-D)
Fact: Total Revenue

Multidimensional Representation of 3-dim Data as a Cuboid

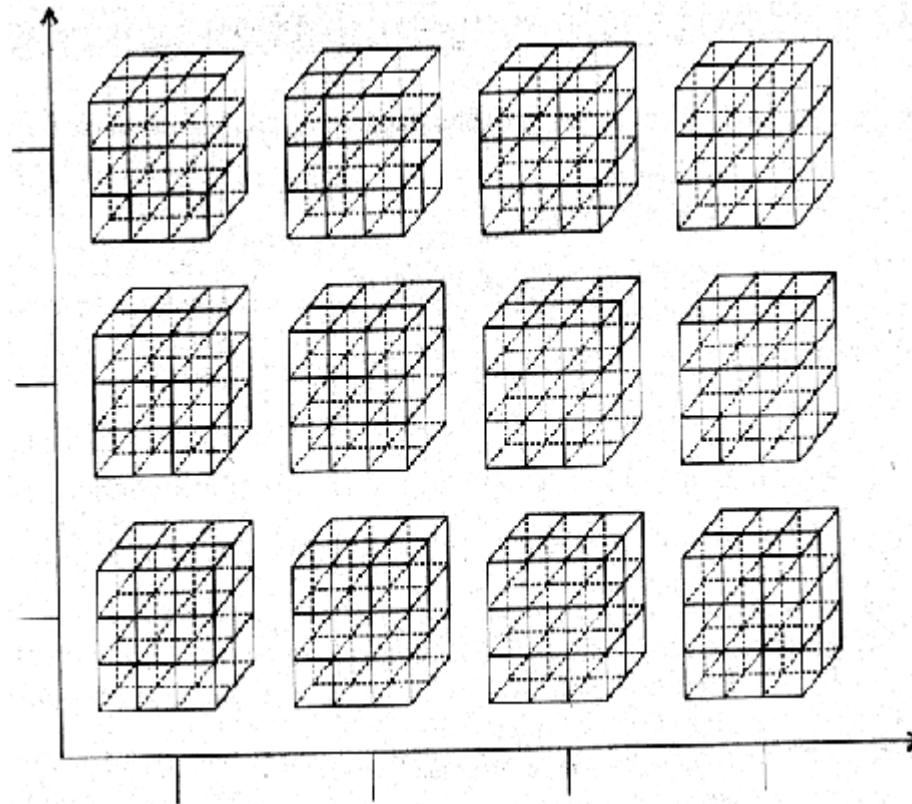
Property Type	City	Time	Total Revenue
Flat	Glasgow	Q1	15056
House	Glasgow	Q1	14670
Flat	Glasgow	Q2	14555
House	Glasgow	Q2	15888
Flat	Glasgow	Q3	14578
House	Glasgow	Q3	16004
Flat	Glasgow	Q4	15890
House	Glasgow	Q4	15500
Flat	London	Q1	19678
House	London	Q1	23877
Flat	London	Q2	19567
House	London	Q2	28677
.....
.....



Dimensions: Property Type, City, Time (so 3-D)

Fact: Total Revenue

Multidimensional Representation of 5-dim Data



Multidimensional Data Model Continued...

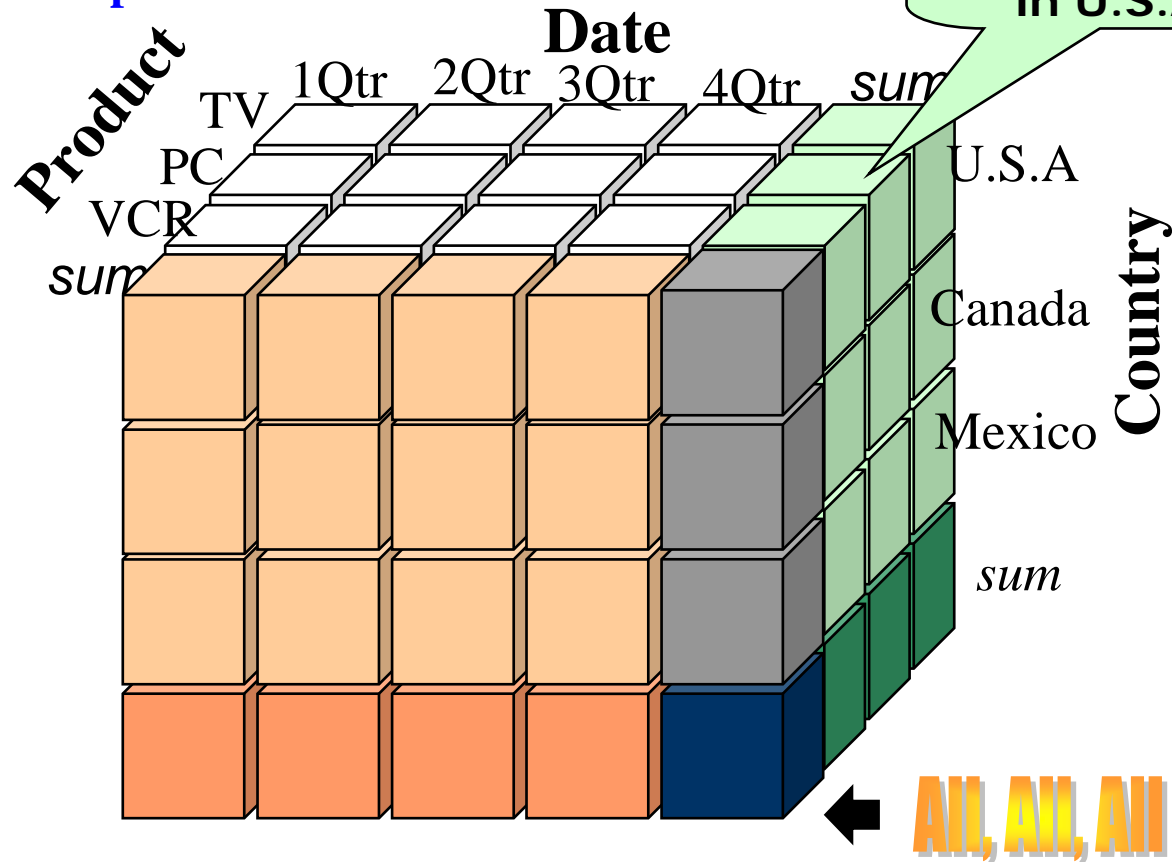
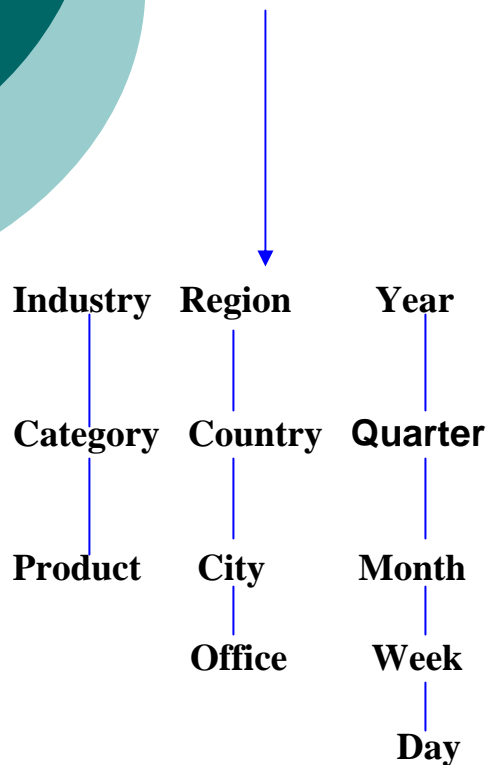
- The multidimensional data model^[1] is composed of logical cubes, measures, dimensions, hierarchies, and attributes.
 - Dimensions^[1]: They are entities with respect to which an organization wants to keep records.
 - Facts/Measures^[1]: It is a subject of decision oriented analysis such as [dollars_sold](#) or [units_sold](#).
 - Facts are numerical measures.
 - Quantities by which we want to analyze relationship between dimensions
 - Hierarchies^[2]: A hierarchy is a way to organize data at different levels of aggregation. In viewing data, analysts use dimension hierarchies to recognize trends at one level, drill down to lower levels.
 - Attribute^[2]: An attribute provides additional information about the data such as color, size.

[1] - Data Mining Concepts and Techniques – Jiawei Han and Micheline Kamber – Book Slides

[2] - <http://www.stanford.edu/dept/itss/docs/oracle/10g/olap.101/b10333/multimodel.htm>

Sales volume as a function of product, Date, Country

Dimensions: Product, Location, Time
 Hierarchical summarization paths



The Relational Implementation of the Model

The relational implementation of the multidimensional data model is typically:

- Star Schema
- Snow Flake schema
- Fact Constellation

Example

Consider a database of sales, perhaps from a store chain, classified by date, branch, product (item), and location.

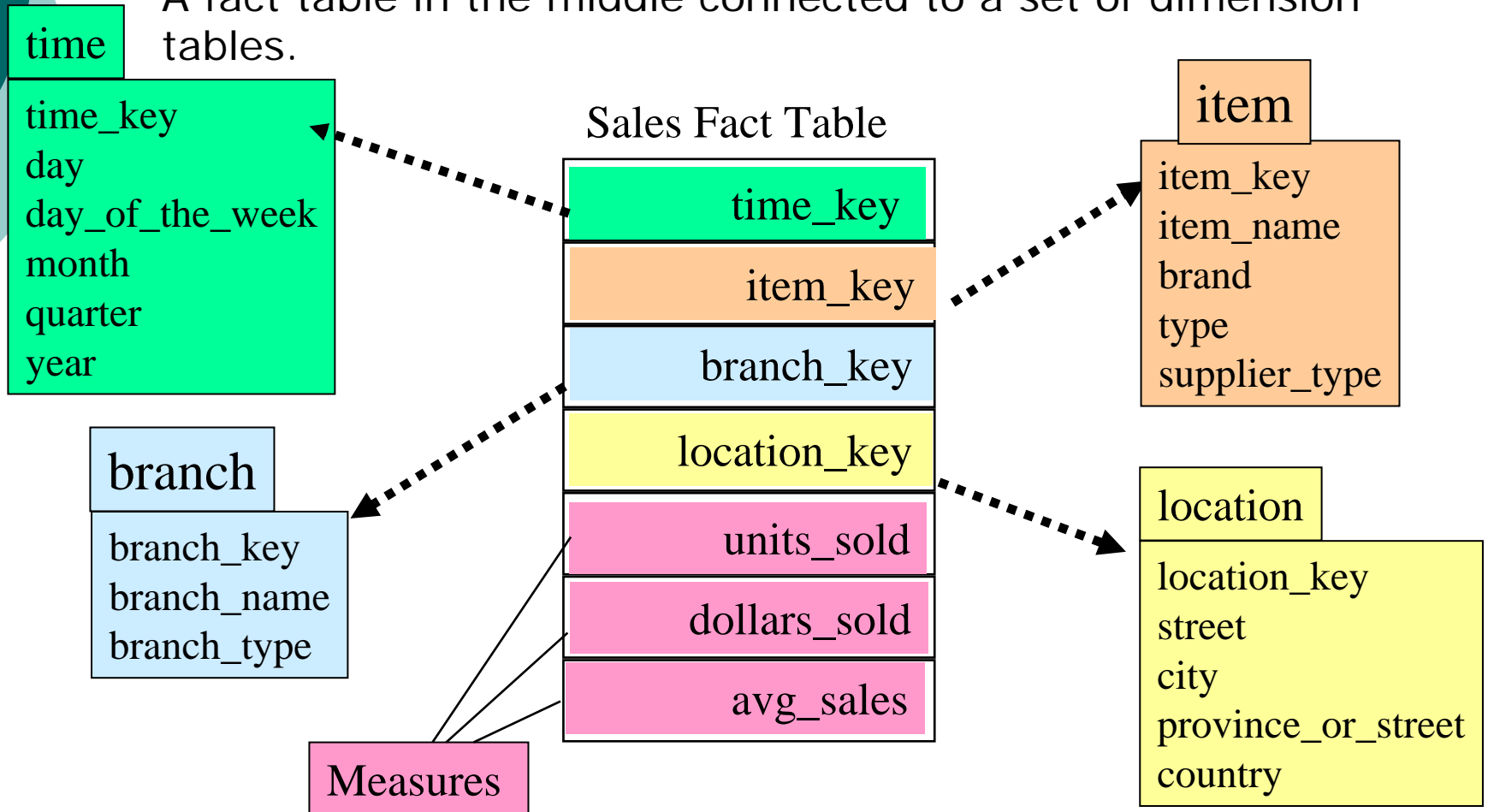
And you have been asked to create a Star Schema for the above mentioned DB to retrieve the units sold, total sale, avg sales classified by time, branch, item, and location.

Example Continued...

- Dimensions:
 - Dimension are - Item, Time, branch, location
 - Create Dimension table for each of these dimensions having dimension attributes
- Facts/Measures:
 - Facts are – Units sold, Sales, Avg Sales
 - Create a fact table with above mentioned facts as attributes in addition to the foreign keys to each of the dimension table

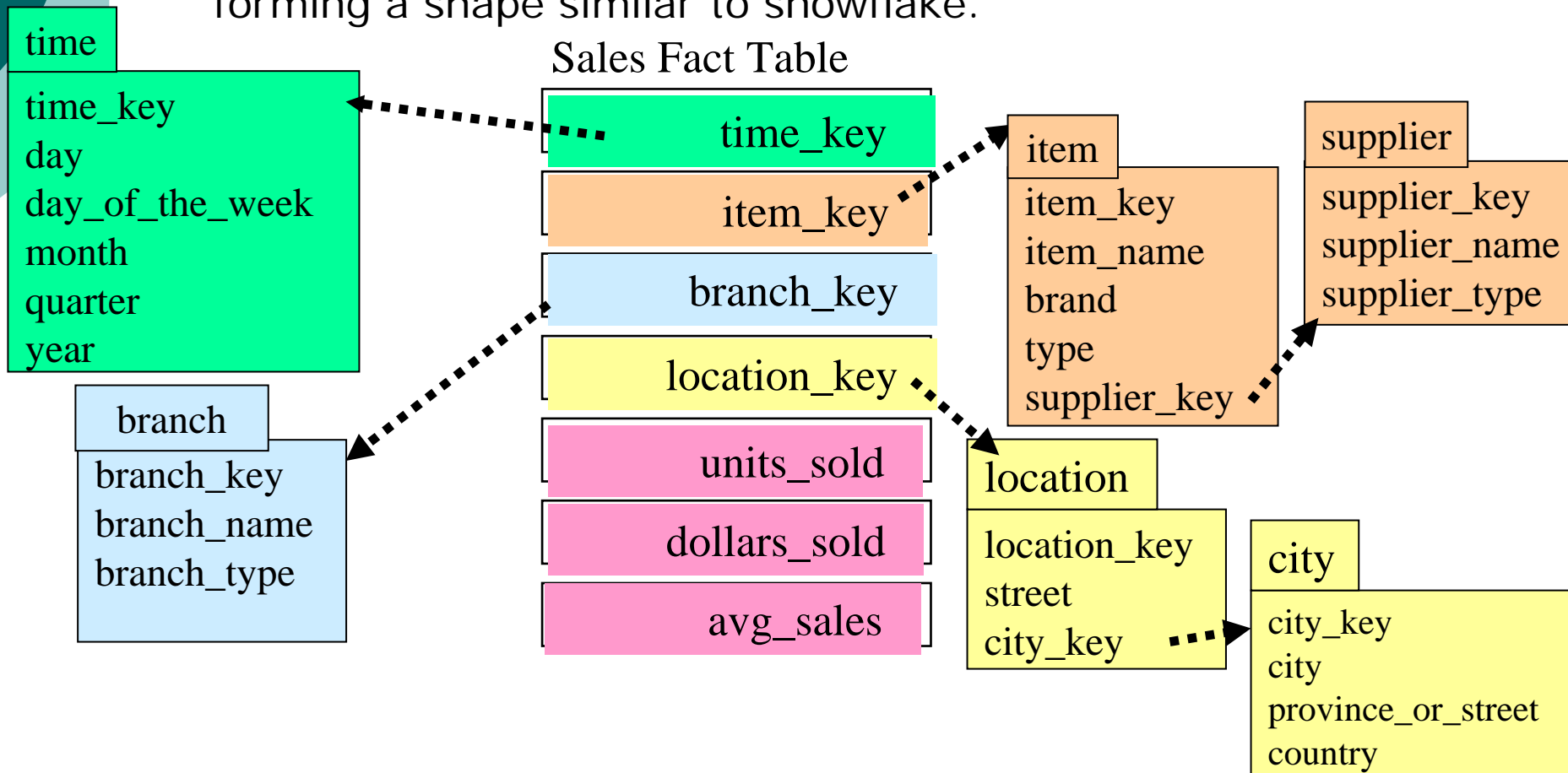
Star Schema

A fact table in the middle connected to a set of dimension tables.



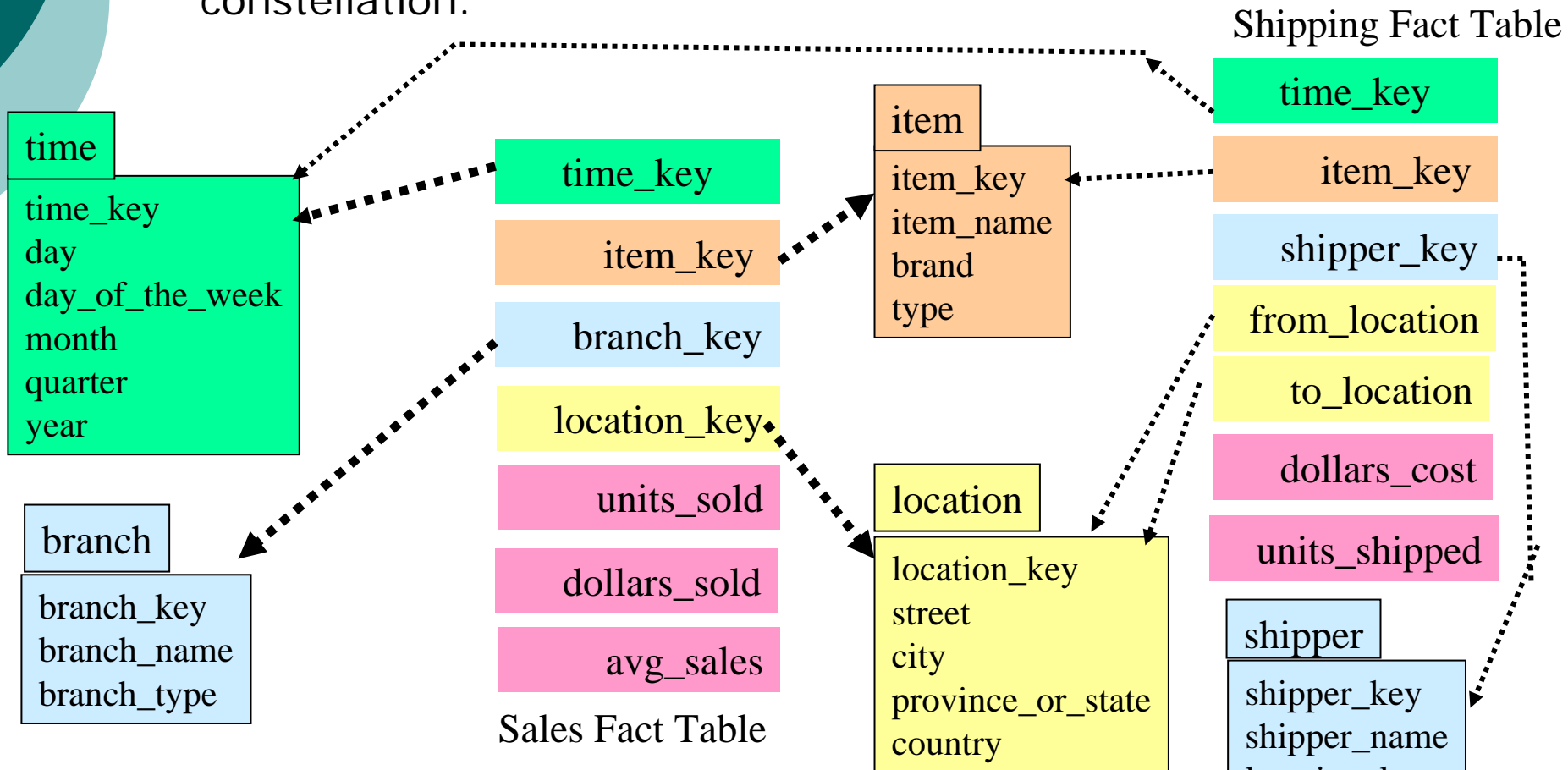
Snowflake schema

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 Constellation

Multiple facts tables share dimension tables, viewed as collection of stars, therefore called galaxy schema or fact constellation.





Research Problems in Data Warehousing

Jennifer Widom
Dept of Computer Science
Stanford University
widom@db.stanford.edu

Proc. of 4th Int'l Conference on Information and Knowledge Management (CIKM), Nov 1995

References

- <http://infolab.stanford.edu/pub/paper/warehouse-research.ps>
- Chawate, Gracia, Ullman, Widom, Hammer – *Integration of Heterogeneous information sources, Oct 1994*
- IEEE Computers – *Special Issue on Heterogeneous Distributed Database Systems, 24 (12), Dec 1991*
- Gupta, Mumick, Jagadish – *Maintenance of materialized views: problems, techniques, and applications*

Abstract

In this paper, the author – Jennifer Widom:

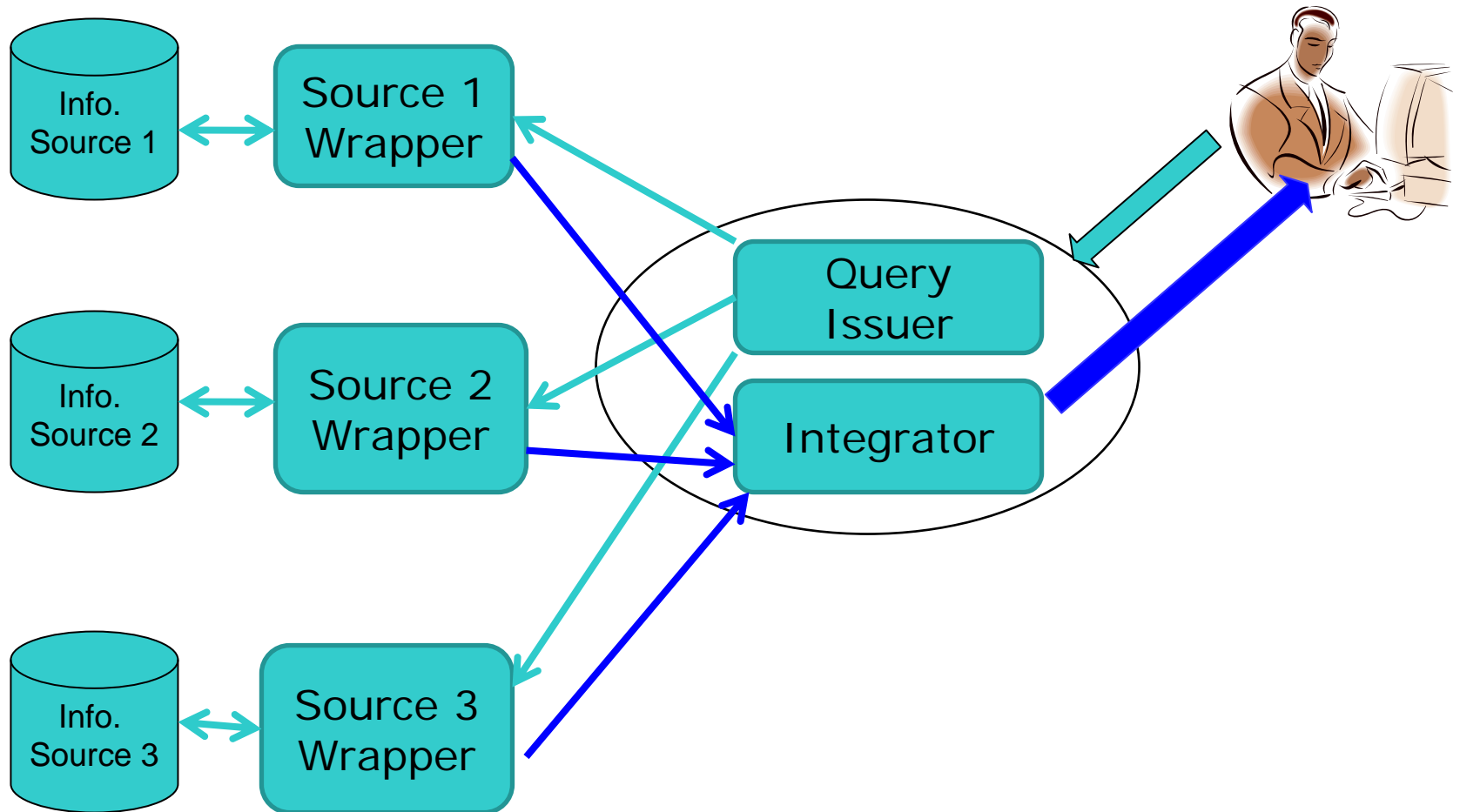
- Motivates the concept of a data warehouse
- Outlines a general data warehousing architecture
- Proposes a number of technical issues arising from the architecture



Two Approaches of Data Integration

- On Demand
- In Advance (Data warehousing)

On Demand (Lazy Approach)



On Demand (Lazy Approach)

It is based on two steps:

1) Accept a query, determine the appropriate sources and generate the sub-queries (commands) to each information source.

2) Perform appropriate translation, filtering and merging on the results obtained from sources and return final result to the client.

The lazy approach to integration is appropriate for:

- Information changes rapidly.
- Clients have un-predictable needs.

Lazy approach incurs inefficiency and delay in query processing:

- Queries are issued multiple times
- Information sources are slow, periodically unavailable.
- Significant processing is required for translation, filtering and merging steps.

In-Advance Approach (Data Warehouse)

The Alternative to On-Demand approach is In-advance in which:

- Information of interest from sources is extracted, translated, filtered and merged with appropriate information, and stored in a centralized repository in advance.
- Query is directly evaluated at the repository.
- This approach is commonly referred as data warehousing.

Basic Architecture presented in paper

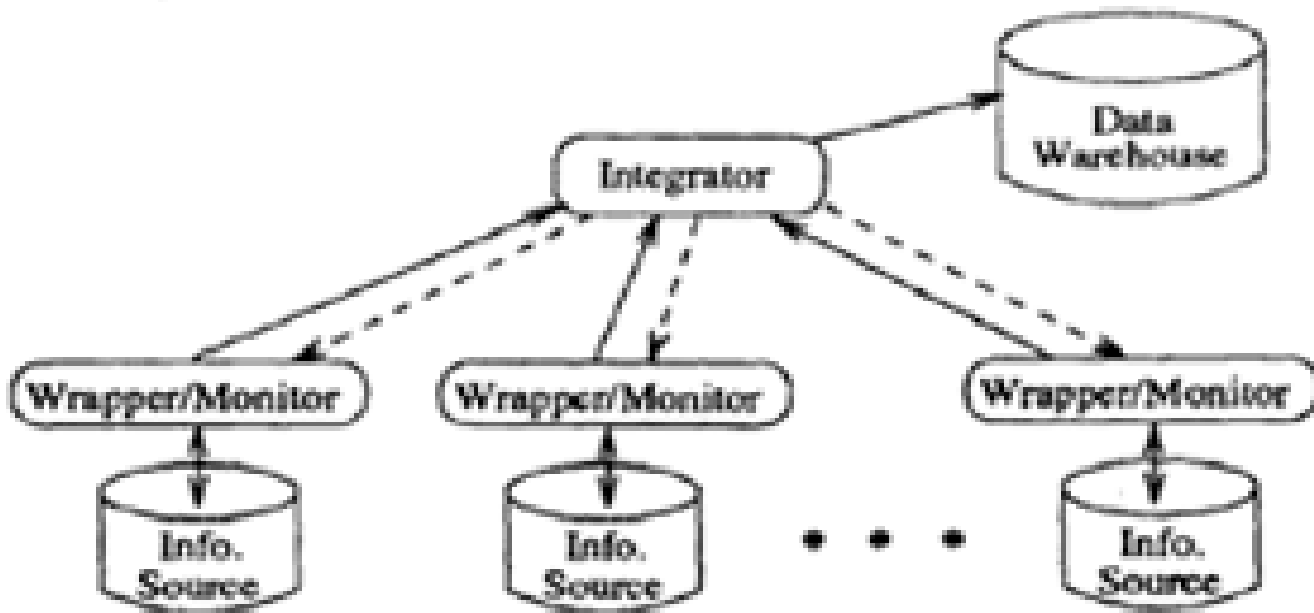


Figure 1: Basic architecture of a data warehousing system

Components of Data Warehouse

➤ Information Sources:

Flat files, knowledge bases, legacy system, operational database and so on.

➤ Wrapper/ Monitor:

The **wrapper** component is responsible for translating information from the native format of the source into the format and data model used by the warehousing system.

Monitor component is responsible for automatically detecting changes of interest in the source data and reporting them to the integrator.

➤ Integrator:

It is responsible for installing the information in the warehouse, which may include filtering the information, summarizing it, or merging it with information from other sources.

Research Problems in Data Warehouse

○ **Translation:**

- If the information source consists of a set of flat files but the warehouse model is relational, then wrapper/monitor must support an interface that presents the data from information sources as if it were relational.
- Incompatibility of data types and size of data type. For e.g. If in the source, one of the text field's size is 12000 bytes, while maximum text size in the data warehouse database is 8000 bytes, then after transformation there will be data loss.

○ **Change Detection:**

The other problem in maintaining data warehouse is Change Detection - means monitoring the source information for the changes that are relevant to data warehouse and propagating the same to data warehouse.



Solutions proposed for Change Detection

1. Ignore the change detection altogether and simply propagate the entire copies of the relevant data from the information source to the Data Warehouse periodically. The integrator can combine this data with the existing warehouse.
2. Integrator can request complete information from all the sources and re-compute the warehouse from the scratch.

Types of Information sources

1. Co operative Sources: These sources provide triggers to notify changes automatically .
2. Logged Sources: Maintenance of logs that stores changes.
3. Query able Sources: Wrapper/monitor performs periodic polling to detect changes.
(Issue: Performance issues related to polling frequency)
4. Snapshot Source: Periodic dumps of data are provided offline.
(Issue: Challenge of comparing very large data dumps)

Note: Translation is the common problem in all these sources.

View maintenance issue

- Integrator accepts change notifications from wrapper and monitor and reflects them into the data warehouse.
- **Data in the warehouse can be seen as a materialized view.** Hence the job of integrator is to perform materialized view maintenance.

Issue with view maintenance: -

1. Base data may come from legacy systems that are unable or unwilling to participate in view maintenance.
2. Most materialized view maintenance techniques rely on the fact that base data update is closely tied to the view maintenance machinery and view modification occurs within the same transaction as updates. Where as in warehouse, the system maintaining the view (integrator) is loosely coupled with base data.

Miscellaneous issues

➤ **Source and warehouse evolution:**

The schema changes at the source should be handled with as few disruptions or modifications to other components of warehousing systems as possible.

➤ **Duplicate and inconsistent information:**

Due to multiple and heterogeneous information sources, there is high likelihood of encountering copies of the same information from multiple sources.

➤ **Outdated information:**

Techniques are needed to ensure that outdated information is efficiently purged from the data warehouse.



Thank You