DataWarehouse and OLAP Chapter 3

CSE521 DATA MINING

Professor Anita Wasilewska
Computer Science Department
Stony Brook University

References

- **CHAPTER 3 Data Warehouse and OLAP Technology: An Overview.**

- Online Sources:

https://www.betterbuys.com/wp-content/uploads/2015/01/Data-Warehouse-Graphic.png
http://bi-dwblog.blogspot.com/2010/07/dimensional-model.htm
https://www.mytechlogy.com/IT-blogs/20762/the-difference-between-data-warehouses-and-data-marts/#.WsGrgMj_quU
http://www2.cs.uregina.ca/~dbd/cs831/notes/dcubes/dcubes.html
https://image.slidesharecdn.com/datacubes-110511072337-phpapp01/95/data-cubes-26-728.jpg?cb=1387530899
https://www.tutorialspoint.com/dwh/dwh_olap.html

Research Paper

- Title : A Data-Warehouse / OLAP Framework for Scalable Telecommunication Tandem Traffic Analysis
- Authors: Qiming Chen, Meichun Hsu, Umesh Dayal
- Conference : 16th International Conference on Data Engineering (Cat. No. 00CB37073)
- Place : San Diego, California
- Date of Publication : 3 March 2000
- Research Paper Link: http://ieeexplore.ieee.org/ stamp /stamp.jsp?tp=&arnumber=839413

In Brief: The Overview

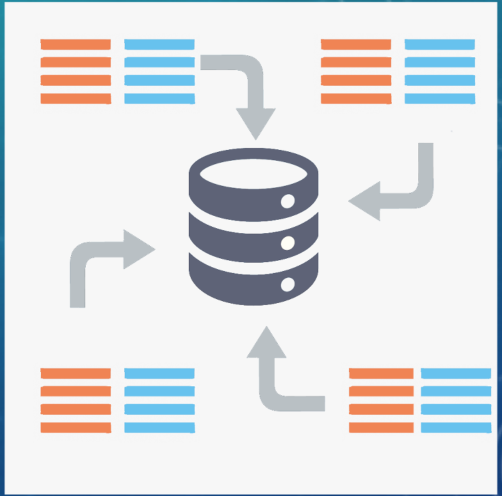
- What is Data Warehouse?
- Data Warehouse Architecture
- Dimensional Data Modelling
- Data Marts
- Data Cube
- Operations on a data cube
- Type of OLAP Server Tools

What is a Data Warehouse ?

DATA WAREHOUSE is a decision support database that is maintained **separately** from the organization's **operational databases**

- "A data warehouse is a **subject-oriented, integrated, time-variant, and nonvolatile** collection of data in **support** of management's decision making process"

  \- William H. Inmon



Source: https://www.betterbuys.com/wp-content/uploads/2015/01/Data-Warehouse-Graphic.png

Data Warehouse - Subject Oriented

- Data Warehouse **is organized** around major subjects
- such as customer, supplier, product, and sales

- Data Warehouse **focuses** on the modeling and analysis of data
- for decision makers **instead of** concentrating on the
- day-to-day operations and transaction processing of an organization

- Data Warehouse **provides** a simple and concise view around
- particular subject issues by excluding data that is not useful in the **decision support** process

Data Warehouse - Integrated

Data Warehouse is constructed by **integrating** multiple heterogeneous sources:

Relational databases

Flat files

Online transaction records

Data Warehouse **applies** data cleaning and data integration techniques

to ensure:

consistency in naming conventions

encoding structures

attribute measures, etc….

Data Warehouse - Time Variant

Data is stored to provide information from a

- **historical** perspective over sometimes many years

- Every **key structure** in the data warehouse contains,

- either implicitly or explicitly, an element of time

Data Warehouse - Non Volatile

- Physically separate **store of data** transformed from the
- **application data** found in the **operational** environment
- **Does not** require transaction processing, recovery and
- concurrency control mechanisms
- **Requires** only two operations in **data accessing**

Initial loading of data and **access** of data

Heterogeneous Database Integration – Traditional Approach vs Data Warehouse

Traditional Approach:

- It is a query driven approach
- Builds wrappers/mediators on top of heterogeneous databases
- When a query is posed to a client site, a meta-dictionary **is used** to **translate** the query into queries appropriate for individual heterogeneous sites involved, and the results are integrated into a global answer set
- It is a **complex** information filtering, competing for resources**, expensive**

Data warehouse:

- Update-driven, high performance
- Information is integrated in advance and stored in warehouses
- for direct query and analysis
- **Do not** contain the most current information
- **Supports** complex multi-dimensional queries

Operational Data Base Systems and Data Warehouses

- The **major** task on-line operational data base systems is to **perform** on-line transaction and query processing
- These systems are called
- **OLTP** - **On-Line Transaction Processing** systems

- **Data Warehouse** systems serve users or knowledge workers providing **data analysis** and acting as **decision support** systems
- These systems are called
- **OLAP** - **On-Line Analytical Processing** systems

Data Warehouse vs. Operational DB Systems

Table 3.1 Comparison between OLTP and OLAP systems.

Feature	*OLTP*	*OLAP*
Characteristic	operational processing	informational processing
Orientation	transaction	analysis
User	clerk, DBA, database professional	knowledge worker (e.g., manager, executive, analyst)
Function	day-to-day operations	long-term informational requirements, decision support
DB design	ER based, application-oriented	star/snowflake, subject-oriented
Data	current; guaranteed up-to-date	historical; accuracy maintained over time
Summarization	primitive, highly detailed	summarized, consolidated
View	detailed, flat relational	summarized, multidimensional
Unit of work	short, simple transaction	complex query
Access	read/write	mostly read
Focus	data in	information out
Operations	index/hash on primary key	lots of scans
Number of records accessed	tens	millions
Number of users	thousands	hundreds
DB size	100 MB to GB	100 GB to TB
Priority	high performance, high availability	high flexibility, end-user autonomy
Metric	transaction throughput	query throughput, response time

NOTE: Table is partially based on [CD97].

Source: Data Warehouse and OLAP Technology: An Overview.

Why Have a Separate Data Warehouse ?

Data Warehouse goal is to help to **promote** a high performance for **both** systems:

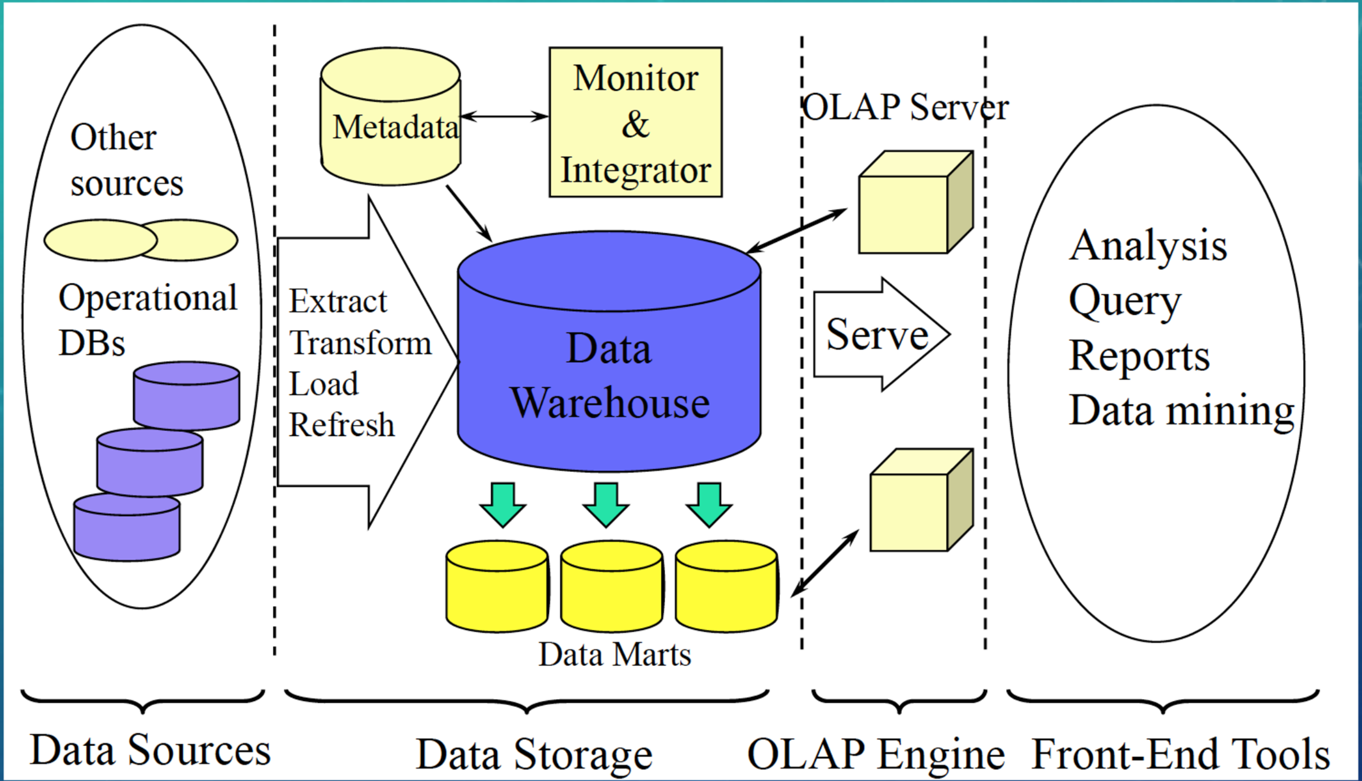
- DBMS (tuned for OLTP): access methods, indexing, concurrency control, recovery
- Data Warehouse (tuned for OLAP): complex OLAP queries, multidimensional view, and consolidation
- Processing OLAP queries in operational databases would substantially **degrade** the performance of operational tasks

Data Warehouse involves the computation of **large groups** of data at summarized levels and may require the use of **special data organization**, access and **implementation methods** based on multidimensional views

Data Warehouse Architecture

- Data Warehouse systems have the following layers:

1. Data Source Layer
2. Data Extraction Layer
3. Staging Area
4. ETL (Extract –Transform- Load) Layer
5. Data Storage Layer
6. Data Logic Layer
7. Data Presentation Layer
8. Metadata Layer
9. System Operations Layer



Source: http://bi-dwblog.blogspot.com/2010/07/dimensional-model.htm

Dimensional Data Modelling

Basic terms

- Dimension: A category of **information**
- For example, the time dimension

- Attribute: A **unique level** within a **dimension**
- For example, Month is an attribute in the Time Dimension

- Hierarchy: The **specification of levels** that represents **relationship** between different attributes within a dimension
- For example, one possible **hierarchy** in the Time dimension is Year → Quarter → Month → Day

Dimensional Data Modelling

The **data** that can be **measured** are called the Facts

Facts are **stored** with the things that can be measured by, which are called the **Dimensions**

Steps to Dimensional Modeling:

- Identify business process/ source of measurements
- Identify the grain
- Identify the dimensions
- Identify the facts

Granularity mean the lowest level of **information** to be **stored** in the fact table

Dimensional Table & Fact Table

Dimension tables hold the information necessary to allow us **to query** it

Dimensions are categories by which **summarized data** can be viewed

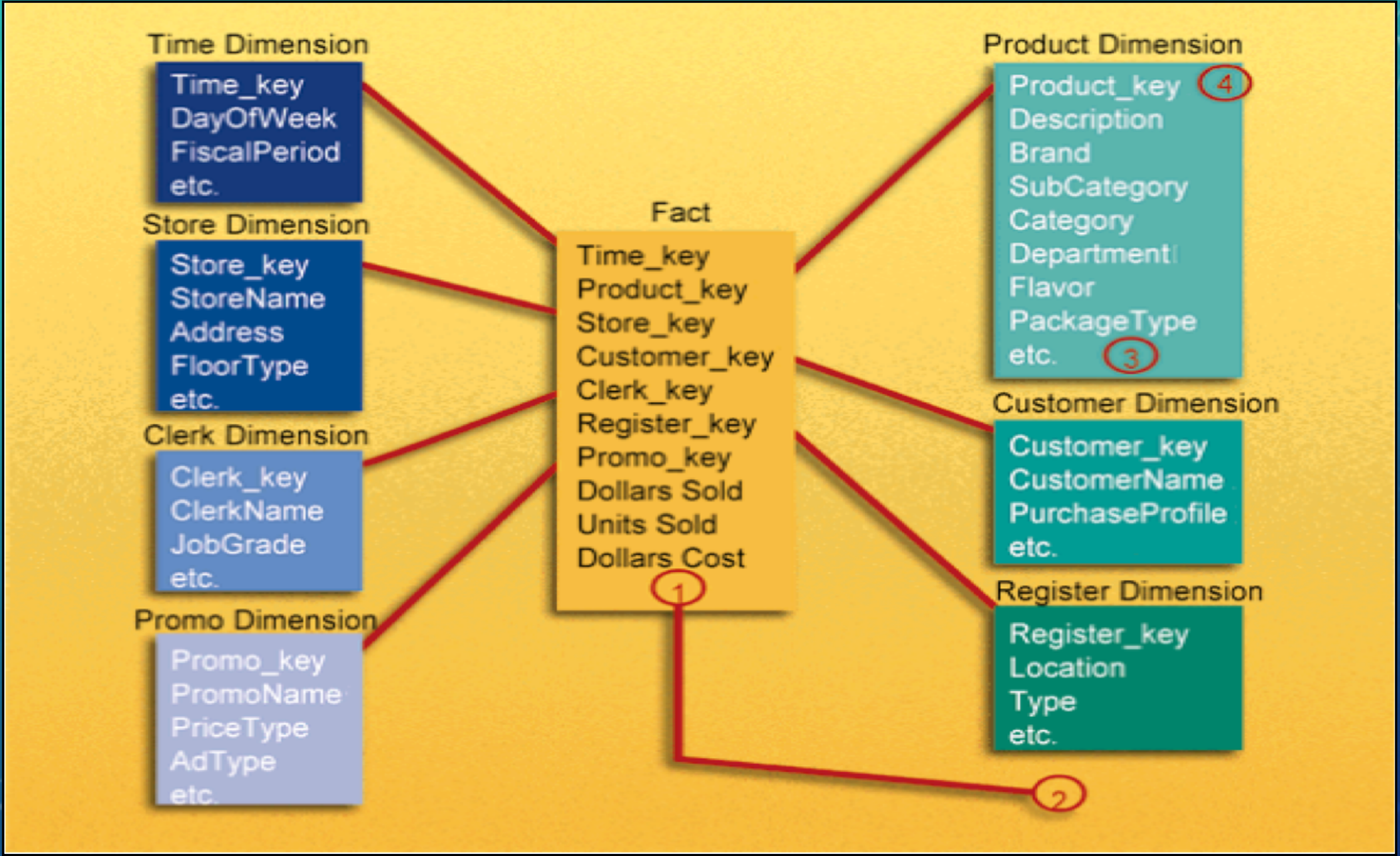
Dimension tables are **referenced** by fact tables using **keys**

The **surrogate key** is used as the primary key in the dimension table

A **fact table** is a table that contains the **measures** of interest

For example, sales amount would be such measure of interest

This **measure** is stored in the fact table with the appropriate **granularity**

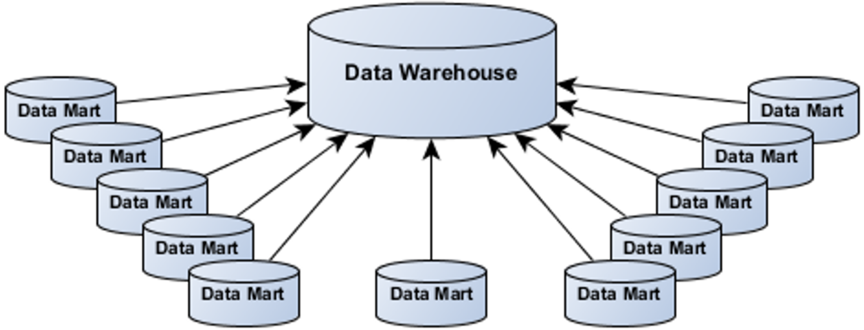


Source: http://bi-dwblog.blogspot.com/2010/07/dimensional-model.html

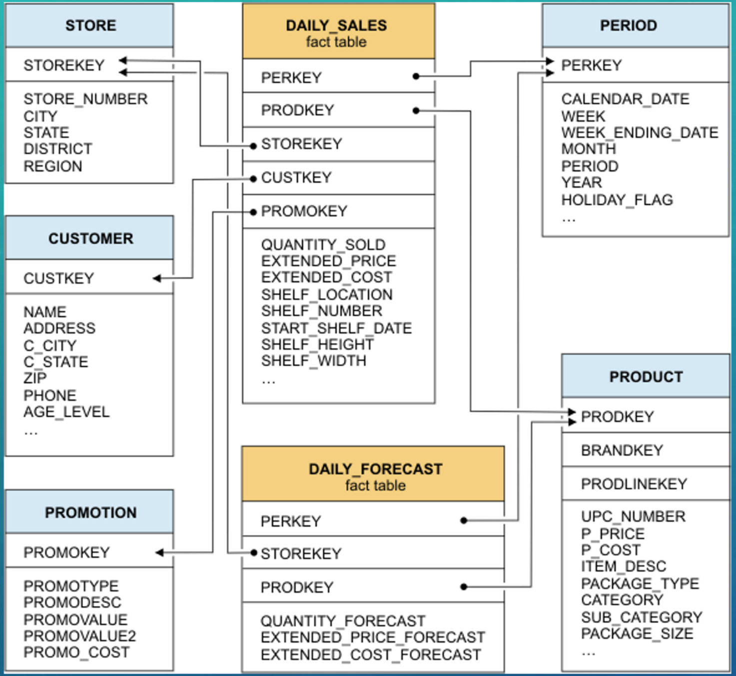
Data Marts

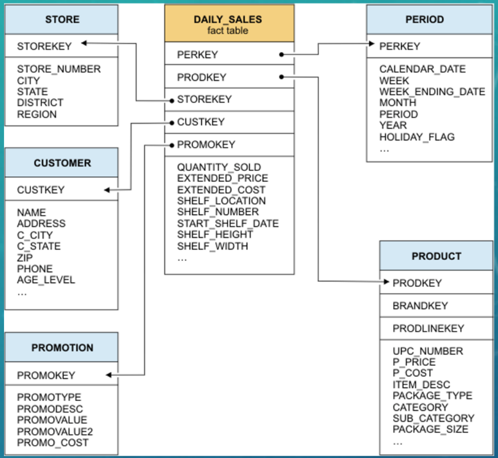
Data mart is a subset of a Data Warehouse where all the **information related** to specific business area is **stored**

Data marts are based on a snowflake or a star schema

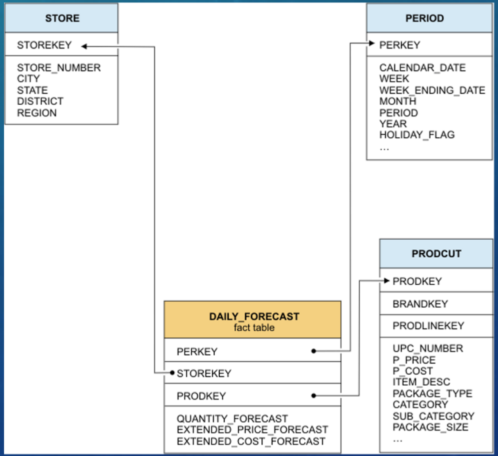


Source: https://www.mytechlogy.com/IT-blogs/20762/the-difference-between-data-warehouses-and-data-marts/#.WsGrgMj_quU





Data Mart 1



Data Mart 2

Data Warehouse

Source: https://www.ibm.com/support/knowledgecenter/en/SSGU8G_12.1.0/com.ibm.acc.doc/ids_acc_012.htm

Data Cube

Data Cube (can be multi-dimensional) is used to **represent** data along some measure of interest

Each dimension **represents** some **attribute** in the database and the cells in the **data cube** represent the facts of interest

For example, they could contain a count for the number of times that **attribute** combination **occurs** in the database ,or the minimum, maximum, sum or average value of some **attribute**

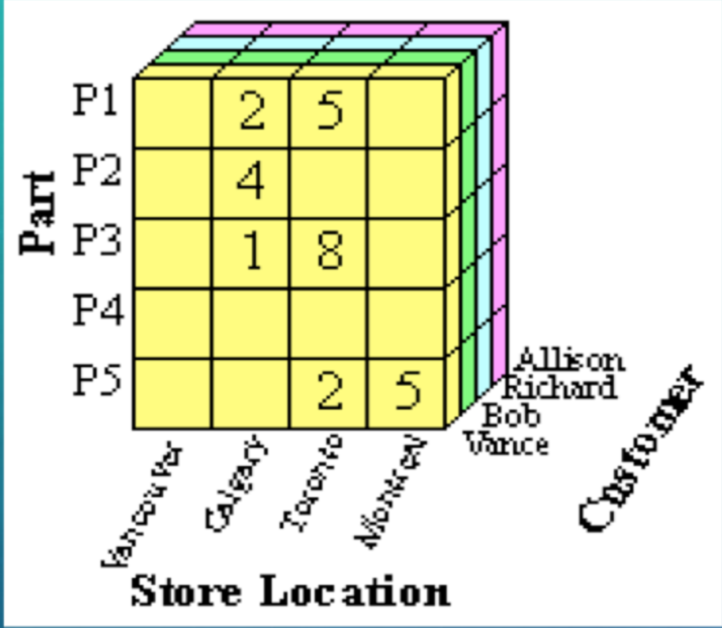
A Simple Example

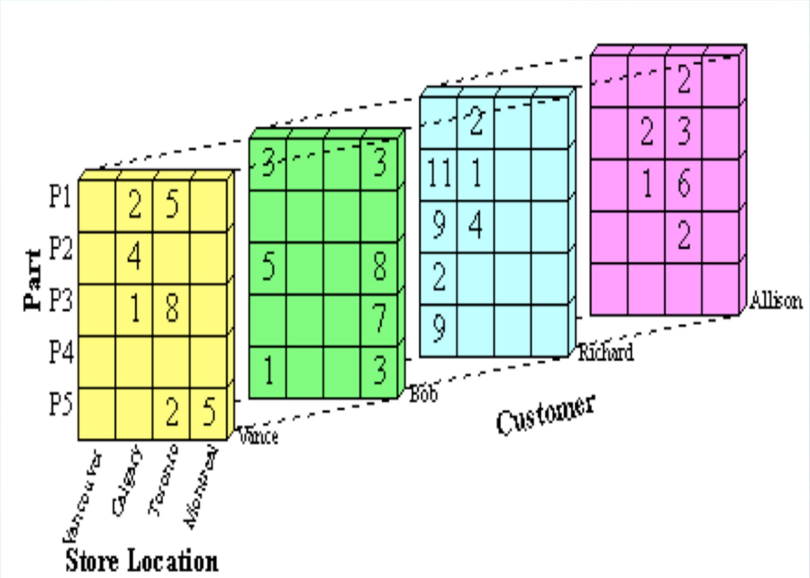
Consider a database that contains **transaction information** relating company sales of a part to a **customer** at a store location

Each cell of this 3-dimensional cube will **represent** insights about the

units of a part purchased by a customer at a particular store

The **cube** can then be **used** to **retrieve** information within the **database** about, for example, which store should be given a certain **part to sell** in order to make the greatest sales





Source: http://www2.cs.uregina.ca/~dbd/cs831/notes/dcubes/dcubes.html

Pre-Compute the results

The goal of DATA WAREHOUSE is to **retrieve** the decision support information from the data cube in the most **efficient** way possible

Three possible **solutions** are:

- Pre-compute all cells in the cube
- Pre-compute no cells
- Pre-compute some of the cells

If the whole cube is **pre-computed**, then queries run on the cube will be **very fast**

The disadvantage is that the pre-computed cube requires a lot of **memory**

The **size** increases exponentially with the number of **attributes** and linearly with the cardinalities of those **attributes**

Representation of a data cube

m-Dimensional Array

A **data cube** built from *m* attributes can be stored as an ***m*-dimensional array**

Each **element** of the array contains the **measure value**, such as **count**

The array itself can be represented as a 1-dimensional array

For example, a 2-dimensional array of size X x Y can be **stored** as a 1-dimensional array of size $|X|x|Y|$

The disadvantage of storing the cube directly as an array is that most data cubes are sparse, so the array will contain many empty elements (zero values).

Representation of Totals

Repeesentaion of Totals is another **aspect** of data cube **representation**

A **simple data cube** does not contain **TOTALS** as the storage of totals **increases**

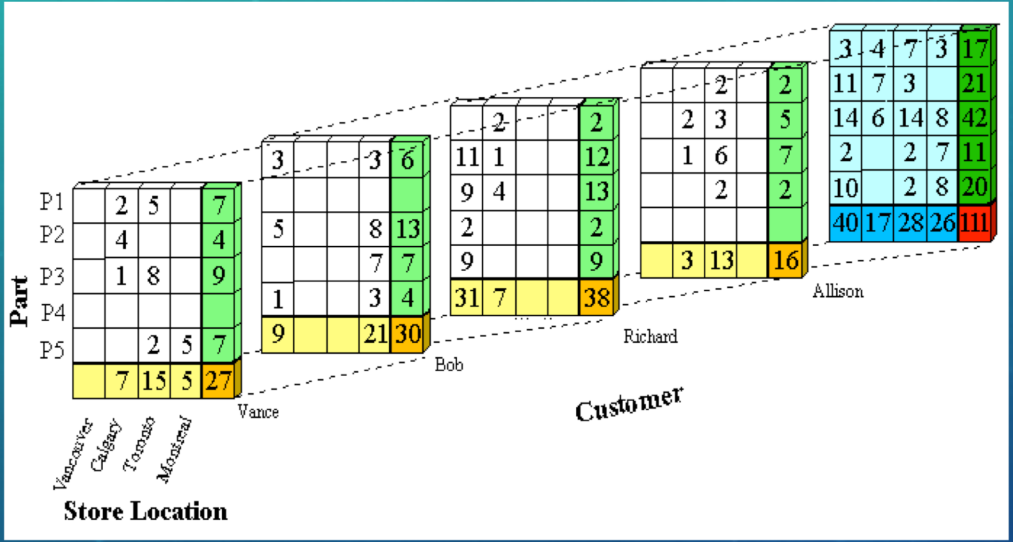
the size of the data cube but can also decrease the time to make **total-based queries**

A simple way to **represent totals** is to **add** an additional layer on *n* sides of the *n*-dimensional data cube

This can be easily **visualized** with the 3-dimensional data cube introduced in next slide.

The **TOTALS** represent the sum of all values in one **horizontal row**, vertical row (column) or depth row of the **data cube**

Representation of Totals



White: Original values

Light yellow: Total for one customer and one store location

Light green: Total for one customer and one part

Light blue: Total for one part and one store location

Dark yellow: Total for one customer

Dark green: Total for one part

Dark blue: Total for one store location

Red: Total number of transactions in all

Source: https://image.slidesharecdn.com/datacubes-110511072337-phpapp01/95/data-cubes-26-728.jpg?cb=1387530899

Operations on a Data Cube

(A) ROLLUP

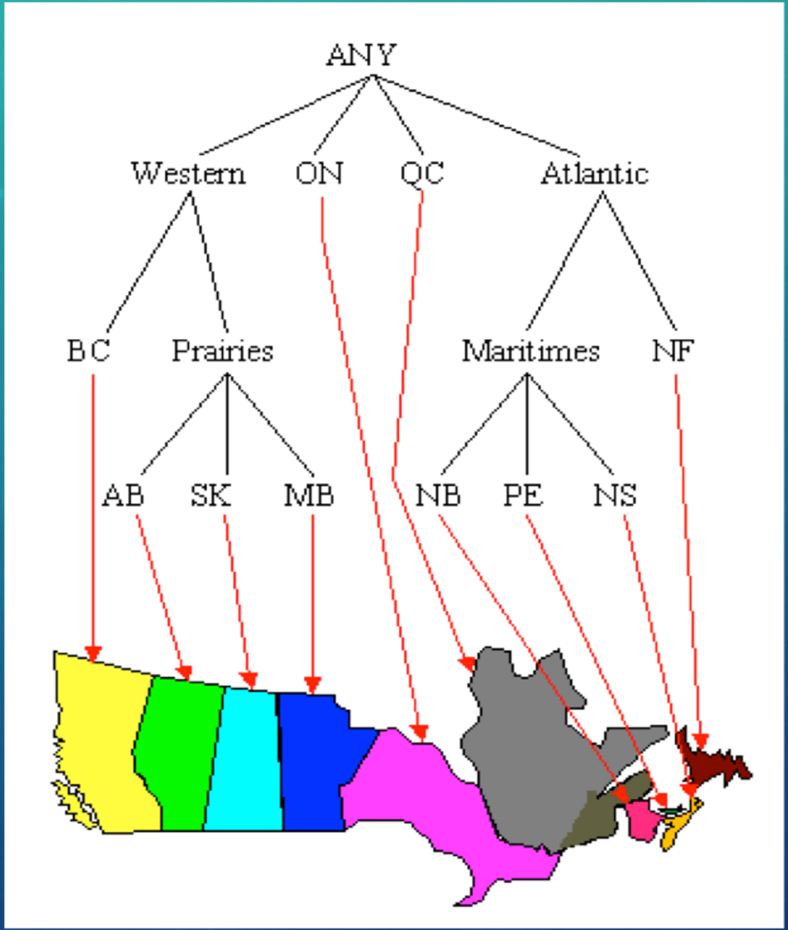
Rollup or **summarization** of the **data cube** can be done by traversing upward through a **concept hierarchy**

A concept hierarchy maps a set of lower level concepts to higher level, more general concepts and is used to **summarize information** in the **data cube**

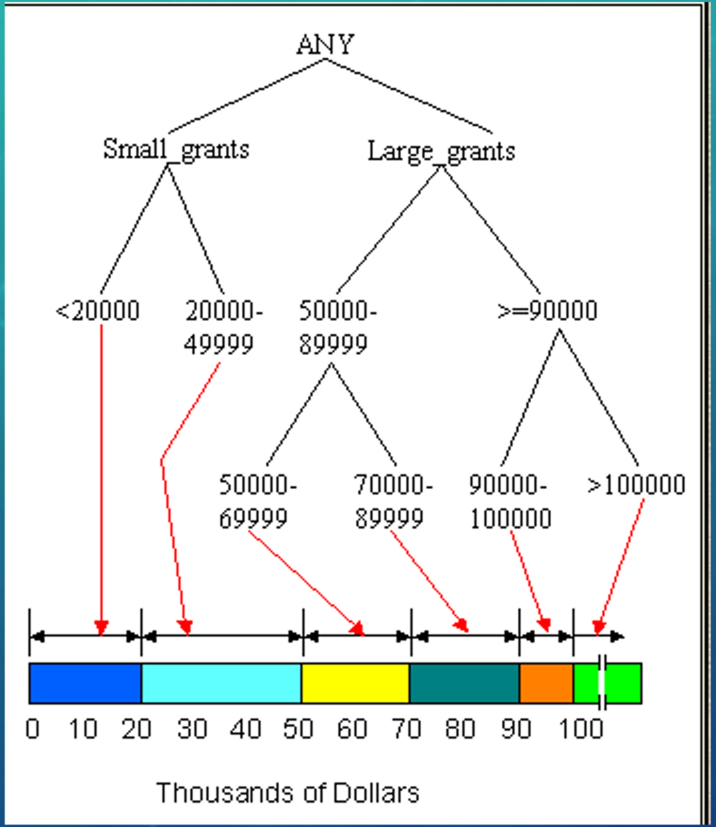
As the values of generalized attributes are combined, cardinalities shrink and

the **cube** gets smaller

Generalizing can be thought of as computing some of the **summary total cells** and storing those in favour of the original cells



Source: http://www2.cs.uregina.ca/~dbd/cs831/notes/dcubes/dcubes.html



Source: http://www2.cs.uregina.ca/~dbd/cs831/notes/dcubes/dcubes.html

(B) DRILL DOWN

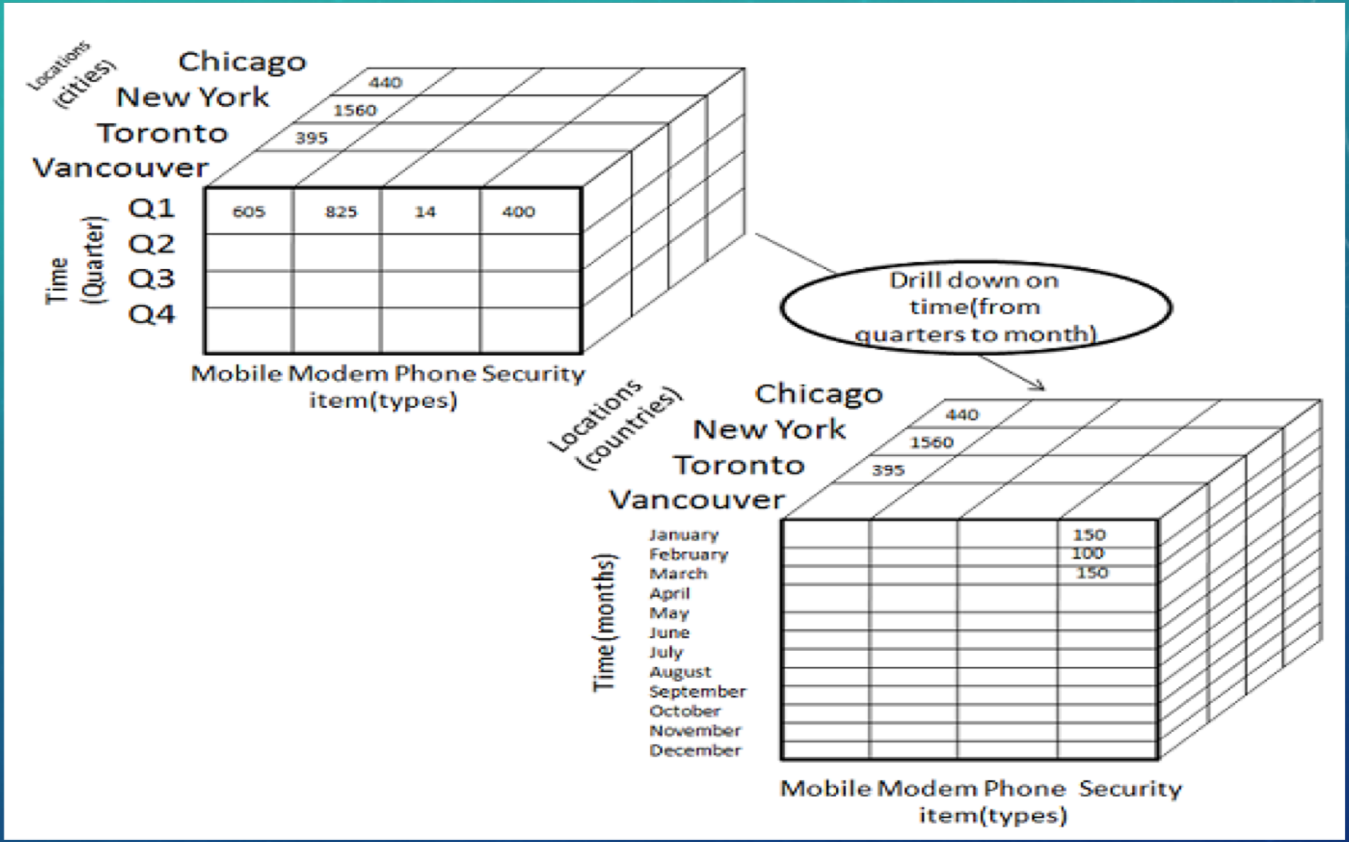
Drill-down is similar to Rollup, but is done **in reverse**

A **drill-down** goes from less detailed data to more detailed data

To **drill-down**, we can either traverse down a **concept hierarchy** or **add** another dimension to the **data cube**

For example, given the data shown, a **drill-down** on the Province attribute would result in more detailed information about the location

The value Prairies would be replaced by the **more detailed** values of AB –Alberta, SK- Saskatchewan and MB -Manitoba



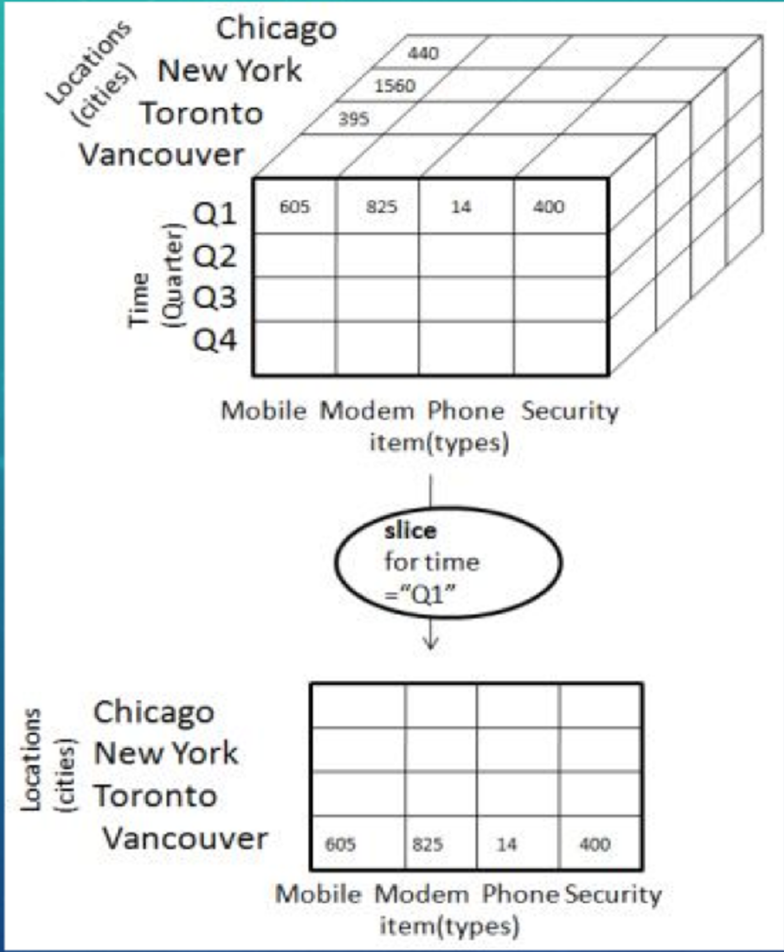
Source: https://www.tutorialspoint.com/dwh/dwh_olap.htm

(C) SLICE AND DICE

Slice and **Dice** refers to a **strategy** for segmenting, viewing and understanding data in a database

Users slice and dice by **cutting** a large segment of data into smaller parts, and **repeating** this process until arriving at the **right leve**l of detail for analysis

Slicing and dicing helps **provide** a closer view of data for analysis and **presents** data in new and diverse **perspectives**



Source: https://www.tutorialspoint.com/dwh/dwh_olap.htm

Types of OLAP Server Tools

1. Relational OLAP (ROLAP)
   - **Relational** and **specialized** relational **DBMS** to store and manage warehouse data
   - OLAP middleware to **support** missing pieces

1. Multidimensional OLAP (MOLAP)
   - Array-based **storage** structures
   - **Direct access** to array data structures

1. Hybrid OLAP (HOLAP)
   - **Storing detailed** data in **RDBMS**
   - **Storing aggregate** data in Multi-dimensional **DBMS**
   - **User access** via MOLAP tools

Research Paper

A Data-Warehouse / OLAP Framework for Scalable Telecommunication Tandem Traffic Analysis

Authors : Qiming Chen, Meichun Hsu, Umesh Dayal
16th International Conference on Data Engineering
March 2000
San Diego, California
Research Paper Link

MOTIVATION

- Telecommunication business intelligence applications require the
- **mining** of large volumes of **Call Detail Records** (CDRs) to **generate**
- system and customer behaviour pattern
- Tandem Traffic Analysis is an **example** of such application which poses several challenges:
  - **Dealing** with the large data volumes and data flow rates
  - Continuous **analysis** and **mining** of CDRs
  - **Tackling storage** constraints that arise due to the massive size of input data
  - **Increasing performance** of the entire system by scaling them to **match** the input data rates

FEATURES OF IMPLEMENTED FRAMEWORK

- **Integration** of data warehousing and OLAP technologies to provide a scalable data management and data mining framework

- **Dynamic** data warehousing to **handle** data staging and retirement
- **Parallel** and **incremental architecture** to scale up OLAP

- **Use of optimizations** like direct-binning and some application specific methods to **reduce** computational load.

WHAT IS TANDEM TRAFFIC ANALYSIS?

- Voice trunks connect EOs which are controlled by and connected to SS7 **signalling network**.
- By monitoring SS7 network, CDRs are generated to **represent** information of each call attempt.
- Each CDR typically consists of calling phone number, **called phone** number, time duration of call , a OPC and a DPC.
- **Tandem traffic analysis** involves studying traffic volume between pairs of EOs.

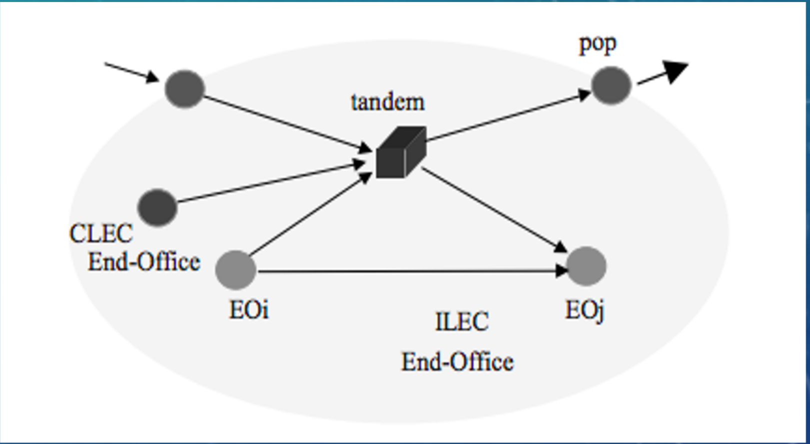


Image source:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=839413

TANDEM TRAFFIC ANALYSIS GOALS

- **Monitoring** network configuration
- **Maximizing** trunk group usage and avoiding traffic jams
- **Discovering** reasons for high tandem load
- **Improving** the quality of service by better business and network planning

Two aspects of **Tandem Traffic Analysis** pose complications as well as optimization opportunities, viz.

- Duplicate CDRs and multiple legs of same call
  - While **monitoring** both inbound and outbound traffic at an EO, duplicate CDRs are **generated** with slightly different timestamps
  - Separate CDRs are generated by each leg of a call with different OPC and DPC
- Mapping between phone numbers and point codes

DATA WAREHOUSE/OLAP BASED TANDEM ANALYSIS FRAMEWORK

- **Centum Call Seconds** (CCS) and other summary information is **represented** in form of **Cubes.**
- A *cube C* has a set of underlying *dimensions D1, ….., DN* and is used to represent a multidimensional *measure*.
- A sub-cube of C can be formed by limiting its dimensions or by taking a **subset** of the domain of dimensions.
- The **OLAP servers** act as engines for creating and updating the CCS and other summary cubes, **deriving** patterns from these cubes and **analyzing** them.
- The **infrastructure** is built on top of Oracle-8 data-warehouse and Oracle Express OLAP server.
- **CDRs** and other summary data is **fed** to the data warehouse **continuously** or **periodically** and dumped to archive adhering to certain data retirement constraints

BASIC FUNCTIONS OF DATA WAREHOUSE/OLAP FRAMEWORK

- **Building** the CCS and other summary data cubes by processing CDRs in the data warehouse **using** OLAP servers.
- **Deriving** multidimensional and multilevel **patterns** from the resulting cubes for analysis
- **Staging** CCS and other summary data between the data warehouse and **OLAP Multidimensional Database** (MDB)

CCS AND OTHER SUMMARY DATA CUBES

- Three kinds of **measures/cubes** :
  - CCS (Centum Call Seconds)
  - NC (Number of calls)
  - NCA (Number of calls answered)
- All measures are **dimensioned** as follows:
  - Epc.o (origin point-code)
  - Epc.d (destination point-code)
  - Tpc (tandem point-code, if any)
  - Day
  - Hour
- These **measures** can be **expressed** in Oracle Express Language.
  - E.g.- define CCS variable int <sparse <epc.o, epc.d, tpc, day, hour>> inplace

MULTILEVEL TANDEM ANALYSIS

- **Values** of dimensions (attributes) can form a hierarchy.
- The values of dimensions can be rolled up the **hierarchy** such that the upper levels can contain the sum of the lower levels.
- Hierarchical dimensions have different **values** at different **levels of abstraction**.
  - DL - Dimension Level
  - DL_D - mapping of DL and D
  - D_D - child & parent mapping

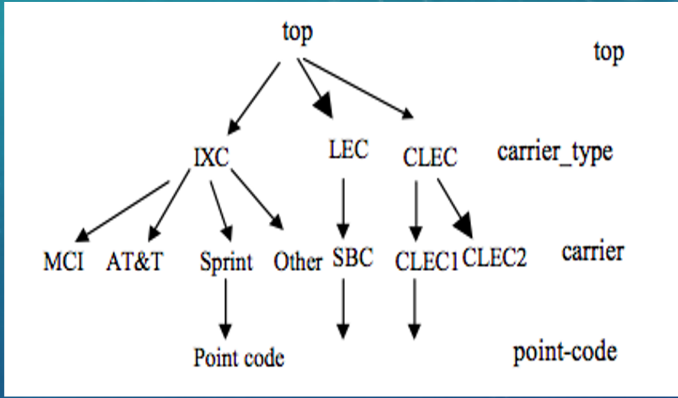


Image source:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=839413

ARCHITECTURE BASED SCALABILITY ENHANCEMENTS

- There are **two basic operations** : Loading data to MDB to **form cubes** and using these cubes to **form patterns**.
- To **reduce** data transfer between RDB and MDB we use the method of *direct binning.*
- **Direct binning** involves forming data cubes **directly** from data which is **retrieved** from the relational database which is a simple and significant solution.
- There are **two ways** to analyze the formed cubes : one-shot analysis and incremental and continuous analysis.
- The later is also called Dynamic Tandem Analysis.

DYNAMIC TANDEM ANALYSIS BENEFITS

- **Provides** dynamic and almost real-time system monitoring
- It **enables** multi level tandem analysis which requires summarizing multiple partial results.
- It **enhances** the scalability since it **does not** involve mining CDRs of arbitrary size.
- **Dynamic Warehousing** also helps in:
  - Incremental data reduction using OLAP servers
  - **Handling** FIFO data with different life-spans
  - **Control** of data operations based on information state
  - **Enabling** the implementation of Parallel OLAP

CONCLUSION

- The implementation of the data-warehouse / OLAP framework for tandem traffic analysis enabled the authors to tackle several issues that hinder the scalability of the telecommunication systems.
- A prototype was used to analyze real time data, which showed improved scalability, maintainability and performance.
- Unlike most applications where OLAP servers are used only as a front-end tool, the authors used it as a computational engine and support information staging between RDB and MDB.
- They have also tackled the problem of data storage by implementing dynamic warehousing using improved retiring rules of old data.
- They achieved processing rates of 1 million CDRs per hour to generate a set of CCS, NC and NCA cubes which was about 45% faster than the existing frameworks.