COP 4710: Database Systems
Spring 2008

CHAPTER 25 – Data Warehousing

Instructor : Dr. Mark Llewellyn
markl@cs.ucf.edu
HEC 236 242, 407-823-2790
http://www.cs.ucf.edu/courses/cop4710/spr2008

School of Electrical Engineering and Computer Science
University of Central Florida

Introduction to Decision Support Systems

- Organizations tend to grow and prosper as they gain a better understanding of their environment. Typically, business managers must be able to track daily transactions to evaluate how the business is performing.
- By tapping into the operational database, management can develop strategies to meet organizational goals. In addition, data analysis can provide information about short-term tactical evaluations and strategies, such as: Are our sales promotions working? What market percentage are we controlling? Are we attracting new customers?
- Managers understand that the business climate is very dynamic, and this mandates their prompt reaction to change in order to remain competitive.
- The modern business climate requires that managers approach increasingly complex problems involving a rapidly growing number of internal and external variables.

Introduction to Decision Support Systems (cont.)

- It should come as no surprise that there is a growing interest in creating support systems dedicated to facilitating quick decision making in a complex environment.
- Different managerial levels require different decision support needs.
  - For example, transaction-processing systems based on operational databases, are tailored to serve the information needs of people who deal with short-term inventory, accounts payable, or purchasing.
  - Middle-level managers and on up, focus on strategic and tactical decision making. Such managers require detailed information designed to help them make complex decisions in the face of a complex data and analysis environment.
- To support middle and upper management, information systems departments have created a number of decision support systems (DSSs).

Introduction to Decision Support Systems (cont.)

- Decision support is a methodology (or a series of methodologies) designed to extract information from data and to use such information as a basis for decision making. A decision support system (DSS) is an arrangement of computerized tools used to assist managerial decision making within a business.
- A DSS usually requires extensive data "massaging" to produce the required information.
- Once constructed the DSS is used at all levels within an organization and is often tailored to focus on specific business areas or problems such as finance, insurance, healthcare, banking, sales, and manufacturing.
- The DSS is interactive and provides ad hoc query tools to retrieve data and to display data in different formats. For example a user might:
  - Compare the relative rates of productivity growth by company division over some specified period of time.
  - Define the relationship between advertising types and sales levels. This relationship might then be used for forecasting purposes.

Introduction to Decision Support Systems (cont.)

- The DSS answers queries such as those on the previous page by combining historical operational data with business models that reflect the business activities.

- A typical DSS consists of four main components: a data store component, a data extraction and filtering component, an end-user query tool, and an end-user presentation tool.

  – The data store component is the data warehouse. Data warehouses differ from conventional databases in the types of data which are stored in them. Certainly a major component of the data warehouse is the operational database, but it goes well beyond that to include many different forms of data including external data (data from outside of the company).

  – The data extraction and filtering component is used to extract and validate data pulled from both the operational database as well as external sources. DSS data differs from purely operational data in three main areas: (1) time span, (2) granularity, and (3) dimensionality. We'll look at these in more detail later.

Operational Data vs. Decision Support Data

- Operational data and DSS data serve different purposes.
  - Most operational data are stored in a relational database in highly normalized fashion. Operational data storage is optimized to support transactions that represent daily operations. Operational data is frequently updated.
- DSS data give tactical and strategic business meaning to operational data. DSS data differs from operational data in three main areas: time span, granularity, and dimensionality.
  - Time span: operational data represent current transactions and represent relatively short time spans. DSS data represents a longer time frame. Managers are typically not interested in a particular sale to customer X, rather they tend to focus on sales generated in the last month or last year, or last five years. They are interested in the buying patterns of a customer or group of customers. The data tends to be historic in nature. The DSS data represents company transactions up to a given point in time: yesterday, last week, last month and so on. Data analysts should be aware that the sales invoice generated two minutes ago is not likely to be found in the DSS database.

Operational Data vs. Decision Support Data (cont.)

- Granularity (level of aggregation): DSS data must be presented at different levels of aggregation, from highly summarized to near-atomic. Managers at different levels in the organization require data with different levels of aggregation. It is also possible that a single problem requires data with different summarization levels. For example, if a manager must analyze sales region, they must be able to access data showing the sales by region, by city within a region, by store within a city within a region, and so on.

  Drilling down data refers decomposing data into finer granularity.

  Rolling up data refers to aggregating data to a higher level or more coarse granularity.

- Dimensionality: This is probably the most distinguishing characteristic of DSS data. From the data analysts point of view, the data are always related in many different ways. For example, if we analyze product sales to a customer during a given time span, we might as "how many widgets of type X were sold to customer Y during the last six months?" This question tends to expand quickly to include many different data slices. For instance, we might want to know how product X fared compared to product Z during the past six months, by region, state, city, store, and customer. Both time and location become part of the picture.

Operational Data vs. Decision Support Data (cont.)

- Data analysts are always interested in developing the larger picture.
- Data analysts tend to include data from many data dimensions, a multi-dimensional view of the data.
- Operational data represent transaction as they happen, in real time. DSS data are a snapshot of the operational data at some point in time. Thus, DSS data are historic, representing a time slice of the operational data.
- Operational data and DSS data also differ in terms of transaction type and transaction volume. Operational data are characterized by update transactions. DSS data are characterized by query operations. DSS data also require periodic updates to load new summary data from operational data. Transaction volume tends to be high for operational data and low for DSS data.

Operational Data vs. Decision Support Data Summary

Characteristic	Operational Data	DSS Data
Data currency	current operations – real time data	historic data, snapshot in time, time component
Granularity	atomic – detailed data	summarized data
Summarization level	low, some aggregation possible	high, many aggregation levels
Data model	highly normalized, mostly relational	non-normalized, complex structures, mostly multidimensional DBMS
Transaction type	mostly updates	mostly queries
Transaction volume	high update volumes, low query	periodic loads and summary calculations
Transaction speed	update critical – tuned for updates	retrieval critical
Query activity	low to medium in volume	high query volume
Query complexity	simple to medium	high to very complex
Data volumes	Hundreds of megabytes to gigabytes and up	Hundreds of gigabytes to terabytes and up

Introduction to Data Warehousing

- A data warehouse holds data drawn from several data sources, maintained by different operating units within the organization, together with historical and summary transformations.
- The data warehouse is based upon extended database technology to provide the management of the data store. VLDB technology is required.
- The decision making process also requires fairly sophisticated and powerful analysis tools. Two main types of analysis tools have emerged in the last few years: On-Line Analytical Processing (OLAP) tools and data mining tools.
- Data warehousing is an extremely complex subject, an entire course could be devoted to the subject. We will cover enough of the subject to give you some familiarity with the topic and an idea of how they are utilized. In fact, a more recent trend has been toward the *data webhouse* which is a data warehouse which is implemented over a network (the most common being the Internet) with no central data repository.

Introduction to Data Warehousing (cont.)

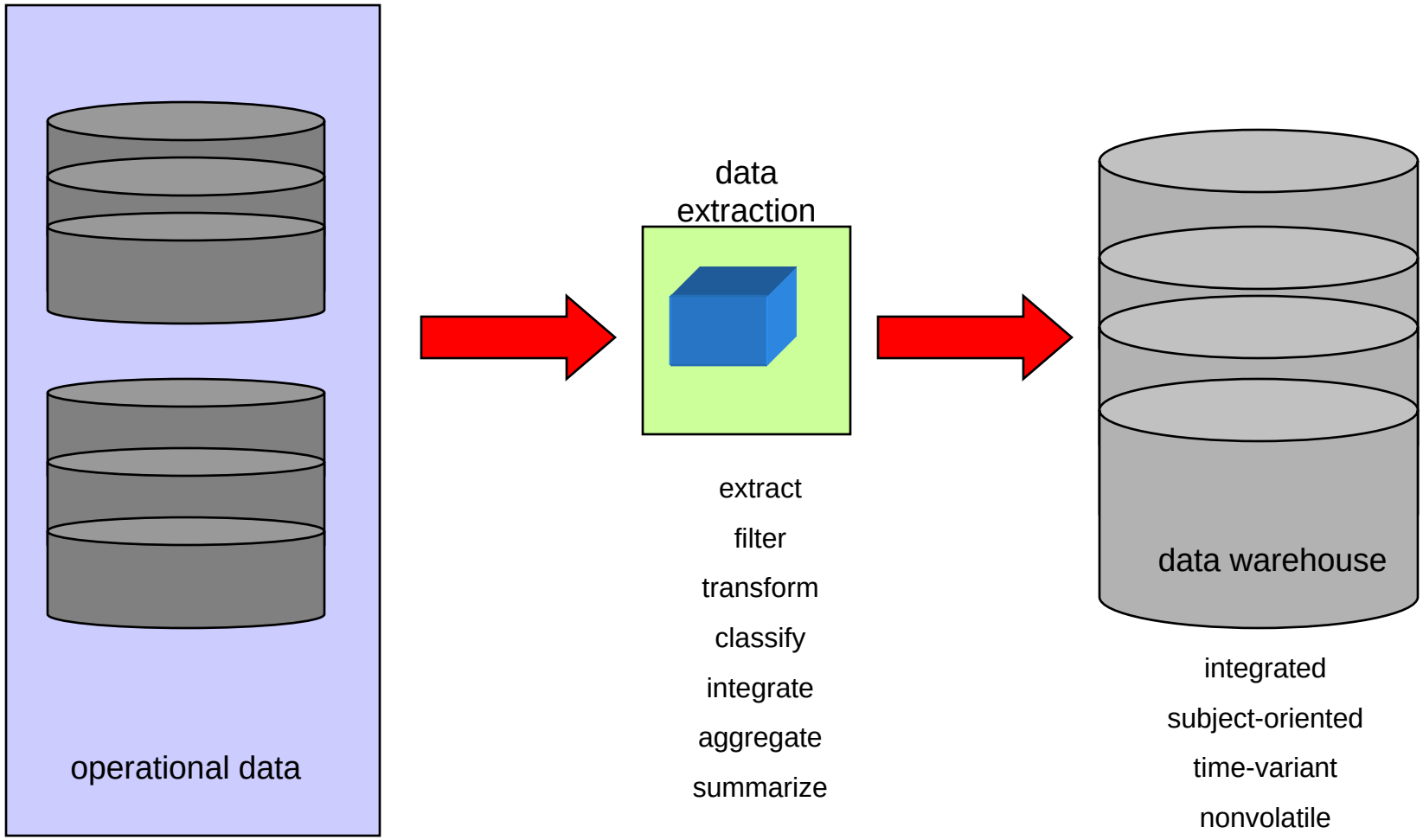
- Bill Inmon is the acknowledged father of the data warehouse. He defines a data warehouse as an integrated, subject-oriented, time-variant, nonvolatile database that provides support for decision making.
  - Subject-oriented – the warehouse is organized around the major subjects of the enterprise (such as customers, products, and sales) rather than the major application areas (such as customer invoicing, stock control, and product sales). This is reflected in the need to store decision-support data rather than application-oriented data.
  - Integrated – the warehouse houses data from various enterprise-wide sources. The source data is often inconsistent using, for example, different formats. The integrated data source must be made consistent in order to present a unified view of the data to the users.
  - Time-variant – the data in the warehouse is only accurate and valid at some point in time or over some time interval. The time-variance of the data warehouse is also shown in the extended time that the data is held, the implicit or explicit association of time with all data, and the fact that the data represents a series of snapshots.

Introduction to Data Warehousing (cont.)

- Non-volatile – the data in the warehouse is not updated in real-time but is refreshed from operational systems on a regular basis. New data is always added as a supplement to the database, rather than as a replacement. The database continuously absorbs this new data, incrementally integrating it with the previous data.

- Depending upon who you talk to or which text on the subject you happen to read, you will probably find a slightly different definition of data warehousing. In short, data warehousing is a combination of data management and data analysis technology. Regardless of the definition, the ultimate goal of data warehousing is to integrate enterprise-wide corporate data into a single repository from which users can easily run queries, produce reports, and perform analysis.

Creating a Data Warehouse



Some Issues of Data Warehousing

- While the concept of data warehousing sounds simple enough, there are many problems associated with implementing and maintaining such a system. We'll highlight a few of the more obvious problems in this section of the notes.

- Underestimation of resources for data loading – Many developers underestimate the time required to extract, clean, and load the data into the warehouse. This process may account for a significant portion of the total development time, although better data cleansing and management tools should ultimately reduce the time and effort spent on data loading.

- Hidden problems with source systems – Hidden problems with the source systems feeding the warehouse may be identified, possibly after years of being undetected. The developer must decide whether to fix the problem in the warehouse and/or fix the source system. For example, when entering the details of a new product, certain fields may allow null values, which may result in entering a null value for such a field even though the data is available and applicable.

Some Issues of Data Warehousing (cont.)

- Required data is not captured – Warehouse projects often highlight a requirement for data not being captured by the existing source systems. The organization must decide whether to modify the OLTP system or create a system dedicated to capturing the missing data.

- Increased end-user demands – After end-users receive query and reporting tools, request for support from IS staff may increase rather than decrease. This is typically caused by an increasing awareness of the users on the capabilities and value of the warehouse. This problem can be partially alleviated by investing in easier-to-use, more powerful tools, or in providing better training for the users. A further reason for increasing demand on IS staff is that once a warehouse is online, it is often the case that the number of users and queries increase together with requests for answers to more and more complex queries.

- Data homogenization – Large-scale warehousing can become an exercise in data homogenization that lessens the value of the data. For example, in producing a consolidated and integrated view of the organization's data, the warehouse designer may be tempted to emphasize similarities rather than differences in the data used by different application areas such as product sales and product inventory.

Some Issues of Data Warehousing (cont.)

- High demand for resources – The warehouse can use huge amounts of disk space. Many relational databases used for decision support are designed around star, snowflake, and starflake schemas (these are schemas in which a central schema spawns related schemas which radiate out from the central schema). These schema designs tend to result in the creation of very large fact tables. If there are many dimensions to the factual data, the combination of aggregate tables and indices to the fact tables can require more space than the data itself.

- Data ownership – Warehousing may change the attitude of the end-users to the ownership of the data. Sensitive data that was originally viewed and used only by a particular department or business area such as in sales or marketing, may now be made accessible to others in the organization. Indeed, some departments or areas may be unaware of the existence of the warehouse.

- High maintenance – Warehouses are high maintenance systems. Any reorganization of the business processes and the source systems may affect the warehouse. To remain a valuable resource, the warehouse must remain consistent with the organization that it supports.

Some Issues of Data Warehousing (cont.)

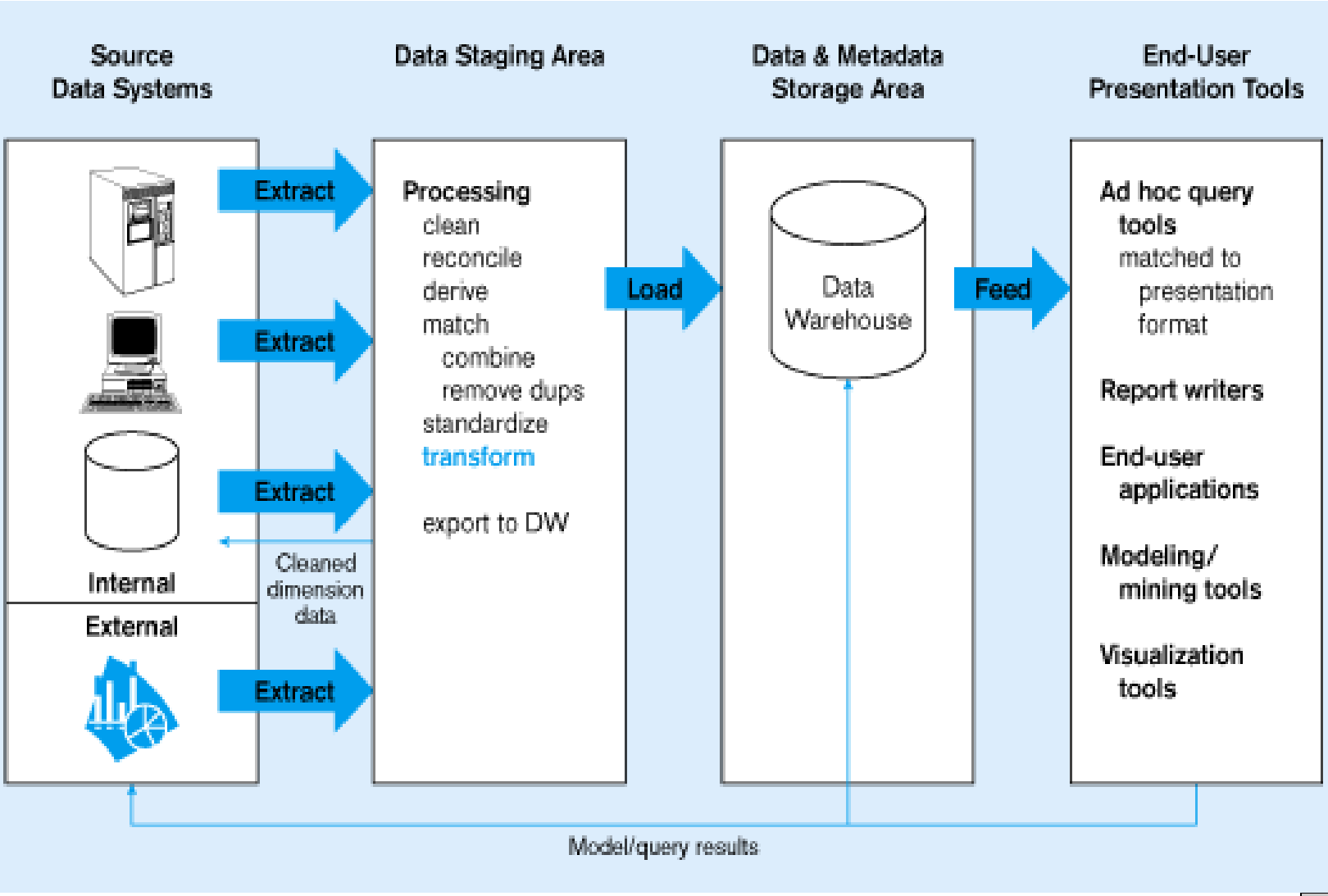
- Long-duration projects – A warehouse represents a single data resource for the organization. However, the building of a warehouse can take up to three years, which is why some organizations are building data marts. Data marts support only the requirements of a particular department or functional area and can therefore be built much more rapidly.

- Complexity of integration – The most important area for the management of a data warehouse is the integration capabilities. This means an organization must spend a significant amount of time determining how well the various warehousing tools can be integrated into the overall solution that is needed. This can be a very difficult task, as there are a number of tools for every operation of the warehouse, which must integrate well in order that the warehouse works to the organization's benefit.

Summary of Differences in Operational Databases and Data Warehouses

Characteristic	Operational DB	Data Warehouse
Primary purpose	Run the business on a real-time basis (current basis)	Support managerial decision making
Type of data	Current representation of the state of the business	Historical point in time (snapshots) and predictions
Primary users	Clerks, salespersons, administrators	Managers, business analysts, customers
Scope of usage	Narrow, planned, simple updates and queries	Broad, ad hoc, complex queries and analysis
Design goal	Performance, throughput, availability	Ease of flexible access and use
Volume	Many, constant updates and queries on one or a few table rows	Periodic batch updates and queries involving many or all rows

Generic Two-level Data Warehouse



Generic Two-level Data Warehouse (cont.)

- Building a data warehouse, like that shown in the previous slide requires four basic steps (moving left to right in the picture):

1. Data are extracted from the various internal and external source files and databases. In large organizations there may be dozens or hundreds of such sources.
2. The data from the various sources are transformed and integrated before being loaded into the warehouse. Transactions may be sent to source systems to correct errors discovered in data staging.
3. The data warehouse is organized for decision support. It contains both detailed and summary data.
4. Users access the warehouse by means of a variety of query languages and analytical tools. Results (e.g., predictions, forecasts) may be fed back into the warehouse and operational databases.

Introduction to OnLine Analytical Processing

- The need for more intensive decision support prompted the introduction of a new generation of tools. These new tools, called online analytical processing (OLAP), create an advanced data analysis environment that supports decision making, business modeling, and operations research.

- OLAP systems share four main characteristics:

  1. Use multidimensional data analysis techniques.
  2. Provide advanced database support.
  3. Provide easy-to-use end-user interfaces.
  4. Support client/server architectures.

Multidimensional Data Analysis Techniques

- The most distinct characteristic of OLAP tools is their capacity for multidimensional analysis. In multidimensional analysis, data are processed and viewed as part of a multidimensional structure. This view of data analysis is particularly attractive to business decision makers because they tend to view business data as data that are related to other business data.

- Multidimensional analysis techniques are augmented by:

  - Advanced data presentation functions: 3D graphics, pivot tables, crosstabs, data rotation, three-dimensional cubes, and so on.

  - Advanced data aggregation, consolidation, and classification functions that all the business data analyst to create multiple data aggregation levels, slice and dice, and drill down and roll up data across different dimensions and aggregation level.s. For example aggregating data across the time dimension (by day, week, month, quarter, year) allows the analyst to drill down and roll up across time dimensions.

  - Advanced computational functions: business-oriented variables (market share, period comparisons, sales margins), financial and accounting ratios (profitability, overhead, cost allocations, returns, etc.).

  - Advanced data modeling functions: support for "what-if" scenarios, variable assessment, linear programming, variable contributions to outcome, etc.

Advanced Database Support

- OLAP tools must have many advanced data access features. These features include:
  - Access to many different kinds of DBMSs, flat files, and internal and external data sources.
  - Access to aggregated data warehouse data as well as to the detailed data found in operational databases.
  - Rapid and consistent query response times.
  - The ability to map end-user requests, expressed in either business or model terms, to the appropriate data source and then to the proper data access language (typically SQL). The query code must be optimized to match the data source, regardless of whether the source is operational or warehouse data.
  - Support for VLDBs (Very Large Databases).

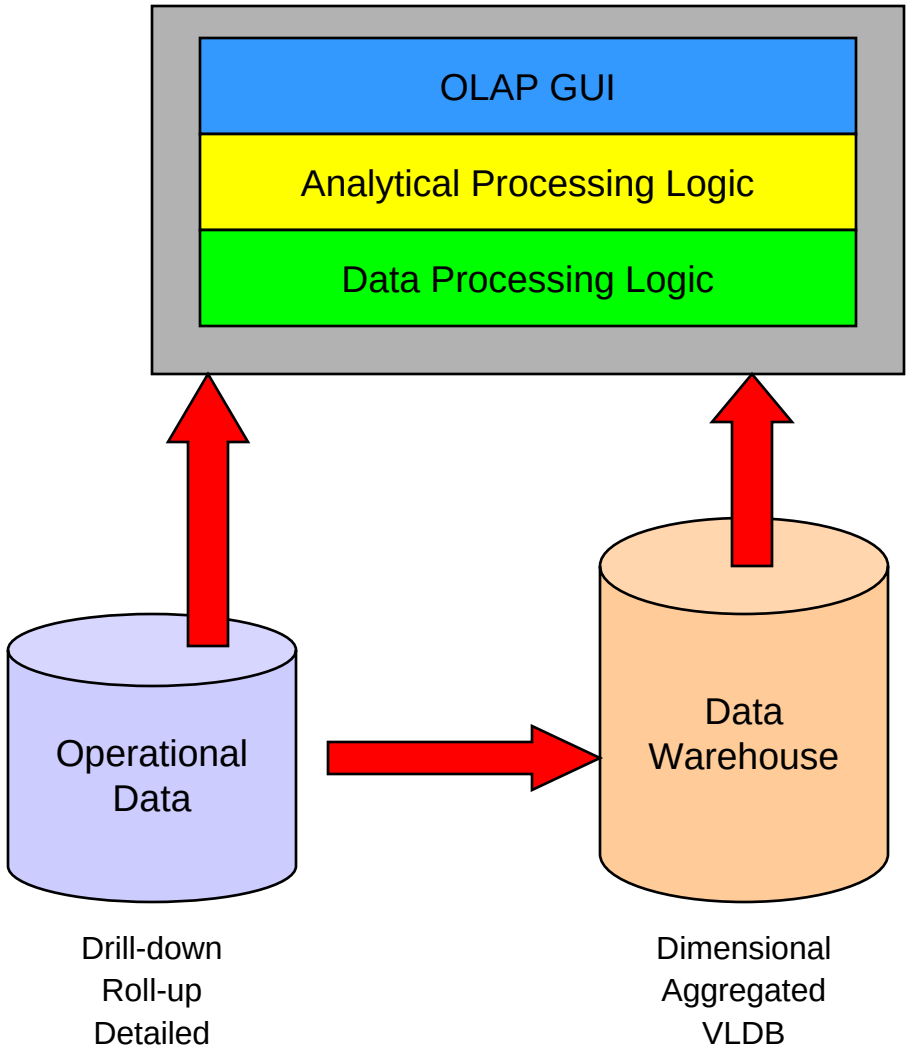
Easy to Use End User Interface

- Developers of OLAP tools learned very early in the game that OLAP tools are much more useful if access to them is kept simple.
- Most of the commercially available OLAP tools have easy to user GUIs and many of the their features have been borrowed from previous generations of data analysis tools that are already familiar to end users.
- More information about various OLAP tools can be obtained from www.olapreport.com. (This is a subscription site, but you can see many details without a subscription.)

Client/Server Architecture

- Client/server architecture provides a framework within which new systems can be designed, developed, and implemented.
- The client/server environment allows us to look at an OLAP system as if it consists of several components that define its architecture.
- The components of the OLAP can be placed on a single computer system or distributed among several computers.
- The OLAP operational characteristics can be divided into three main modules:
  - GUI (graphical user interface).
  - Analytical processing logic.
  - Data-processing logic.

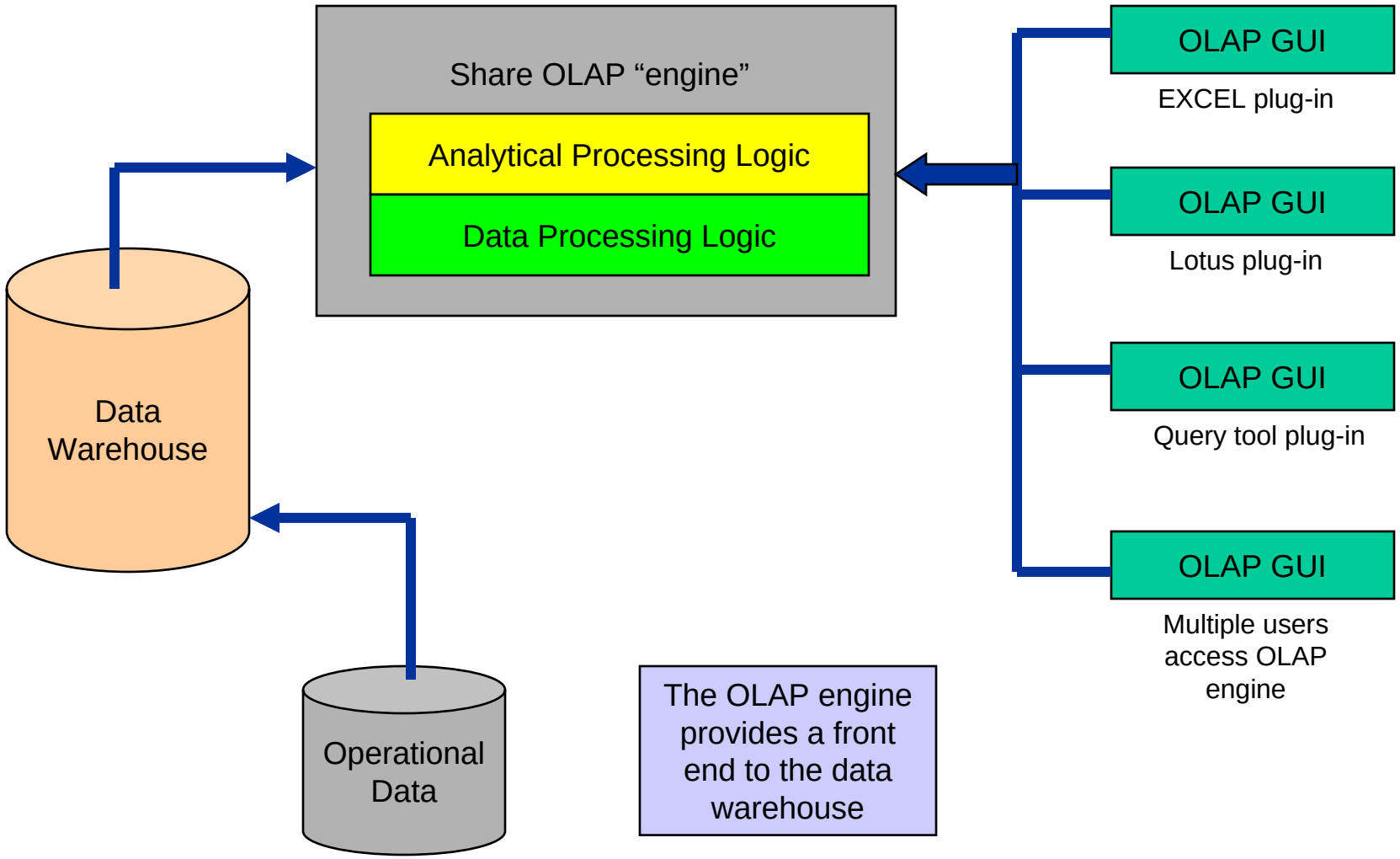
OLAP Client/Server Architecture



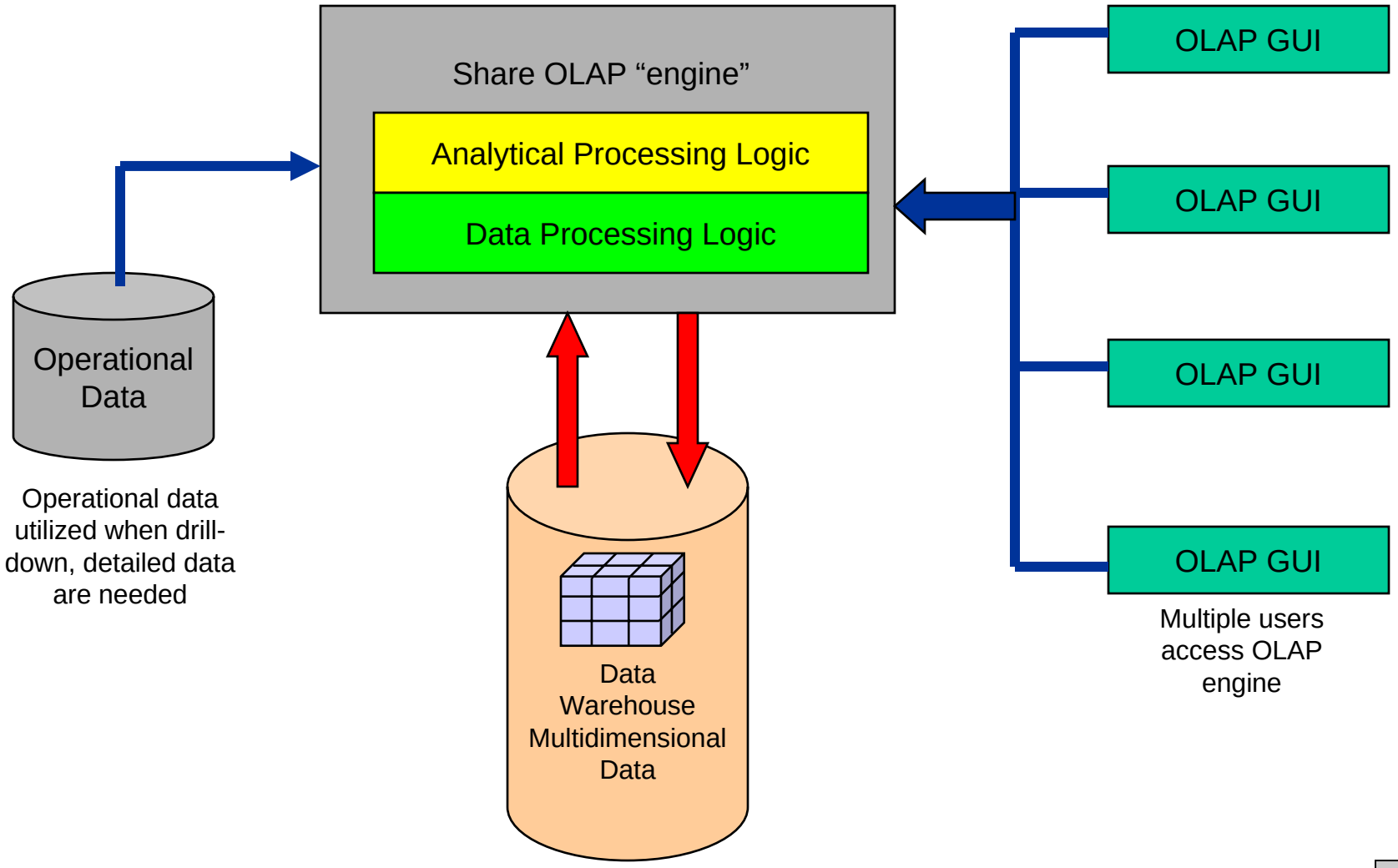
OLAP System exhibits

- Client/Server Architecture
- Easy-to-use GUI
  - Dimensional presentation
  - Dimensional modeling
  - Dimensional analysis
- Multidimensional data
  - Analysis
  - Manipulation
  - Structure
- Database support
  - Data warehouse
  - Operational database
  - Relational
  - Multidimensional

OLAP Server Arrangement



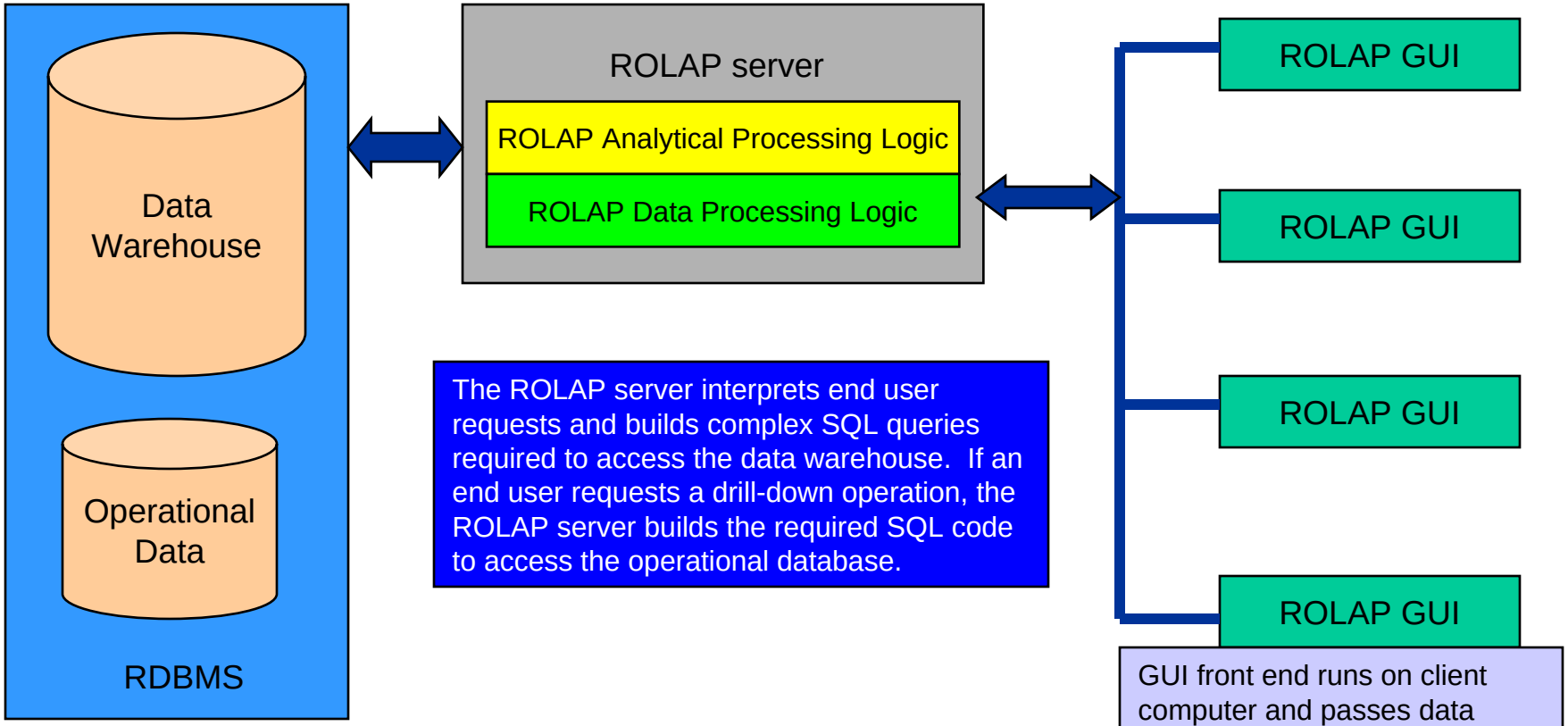
OLAP Server with Multidimensional Data Store Arrangement



Relational OnLine Analytical Processing (ROLAP)

- Relational OnLine Analytical Processing (ROLAP) provides OLAP functionality by using relational databases and familiar relational query tools to store and analyze multidimensional data.
- This approach builds on existing relational technologies and represents a natural extension for relational database vendors.
- ROLAP adds the following extensions to traditional RDBMS technology:
  - Multidimensional data schema support within the RDBMS.
  - Data access language and query performance optimized for multidimensional data.
  - Support for VLDBs.

ROLAP System



The ROLAP server interprets end user requests and builds complex SQL queries required to access the data warehouse. If an end user requests a drill-down operation, the ROLAP server builds the required SQL code to access the operational database.

An RDBMS is used to manage both the data warehouse and the operational data. The RDBMS executes data requests and sends back the data sets.

GUI front end runs on client computer and passes data analysis to the ROLAP server. The GUI receives data replies from the ROLAP server and formats them according to the end user's presentation needs.

Relational OnLine Analytical Processing (ROLAP)

- Relational technology utilizes normalized tables to store data. This reliance on normalized data, while a benefit to the normal relational system, is viewed as a stumbling block in OLAP systems.
- As you will recall, normalization divides tables into smaller pieces to produce the normalized tables. Normalization is useful for reducing redundancies and eliminating certain types of data anomalies.
- Unfortunately, for decision support purposes, it is easier to understand data when they are seen with respect to other data. Normalization tends to preclude this possibility.
- Fortunately, particularly for those businesses which are heavily invested in relational technology, ROLAP uses a special design technique to enable RDBMS technology to support multidimensional data representations. This technique is called the star schema.

An Aside On The Star Schema

- The star schema is a data modeling technique used to map multidimensional decision support data into a relational database. In effect, the star schema creates the near equivalent of a multidimensional database schema from the existing relational database.
- Star schemas yield an easily implemented model for multidimensional data analysis, while still preserving the relational structures on which the operational database is built.
- The basic star schema has four components:
  - facts
  - dimensions
  - attributes
  - attribute hierarchies.

An Aside On The Star Schema (cont.)

- Facts are numeric measurements (values) that represent a specific business aspect or activity. For example, sales figures. Facts are normally stored in a fact table that is the center of the star schema. The fact table contains facts that are linked through their dimensions.

- Dimensions are qualifying characteristics that provide additional perspectives to a given fact. Dimensional data is stored in dimension tables. Recall that DSS data are almost always viewed in relation to other data. For instance, sales might be compared by product from region to region, and from one time period to the next.

  – In effect, dimensions are the magnifying glass through which the facts are studied.

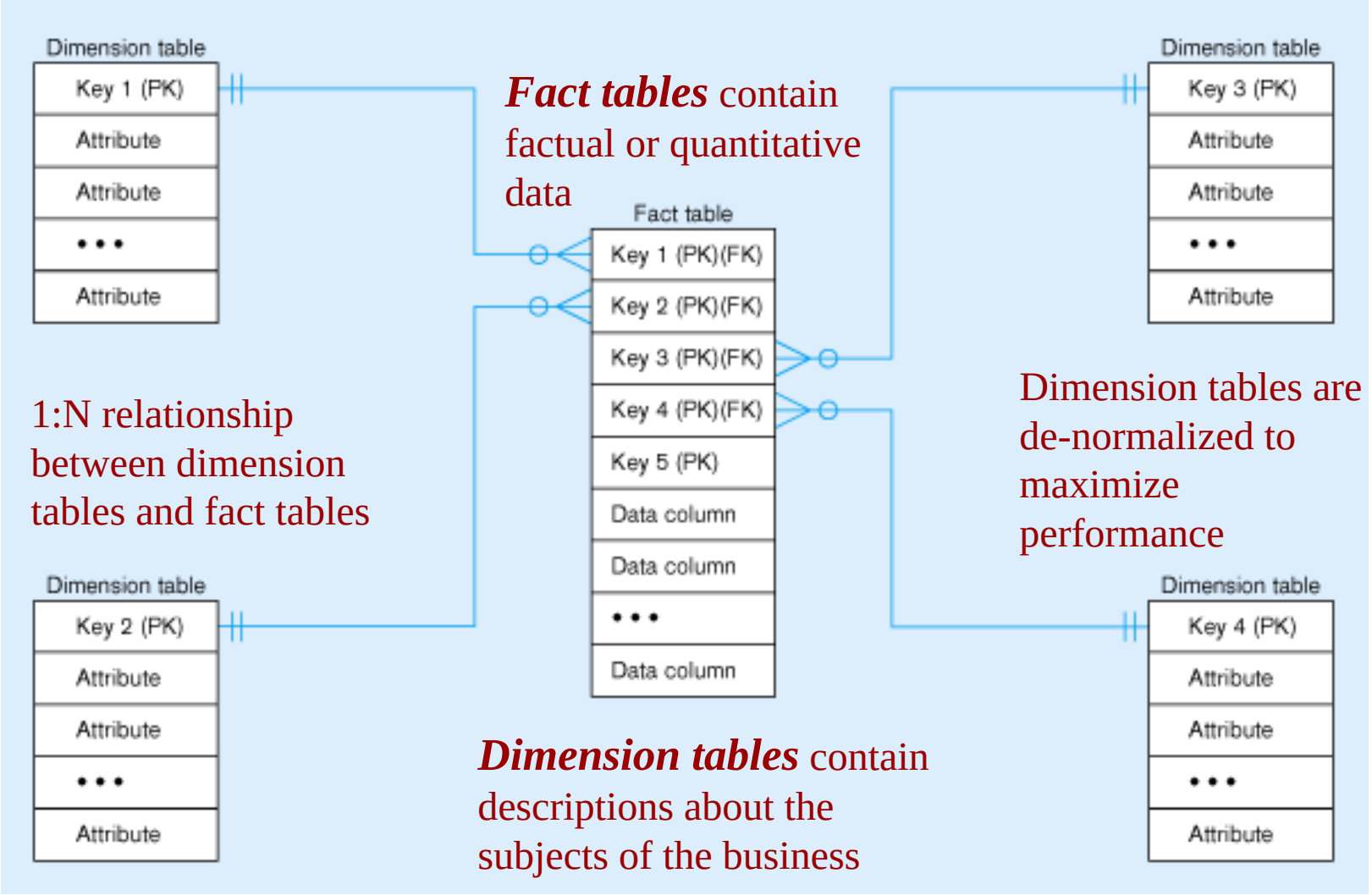
An Aside On The Star Schema (cont.)

- Attributes are often used to search, filter, or classify facts. Dimensions provide descriptive characteristics about the facts through their attributes. The data warehouse designer must define common business attributes that will be used by the data analyst to narrow a search, group information, or describe dimensions.
  - Example: Consider sales. Some possible attributes for the dimensions of sales might be: location, product, and time. These attributes add a business perspective to the sales facts. The data analyst can now group the sales figures for a given product, in a give region, and at a given time.
- The star schema, through its facts and dimensions, can provide the data when needed and in the required format. It can do this without imposing the burden of the additional and unnecessary data (such as order number, purchase order number, status, etc.) that commonly exist in the operational database.

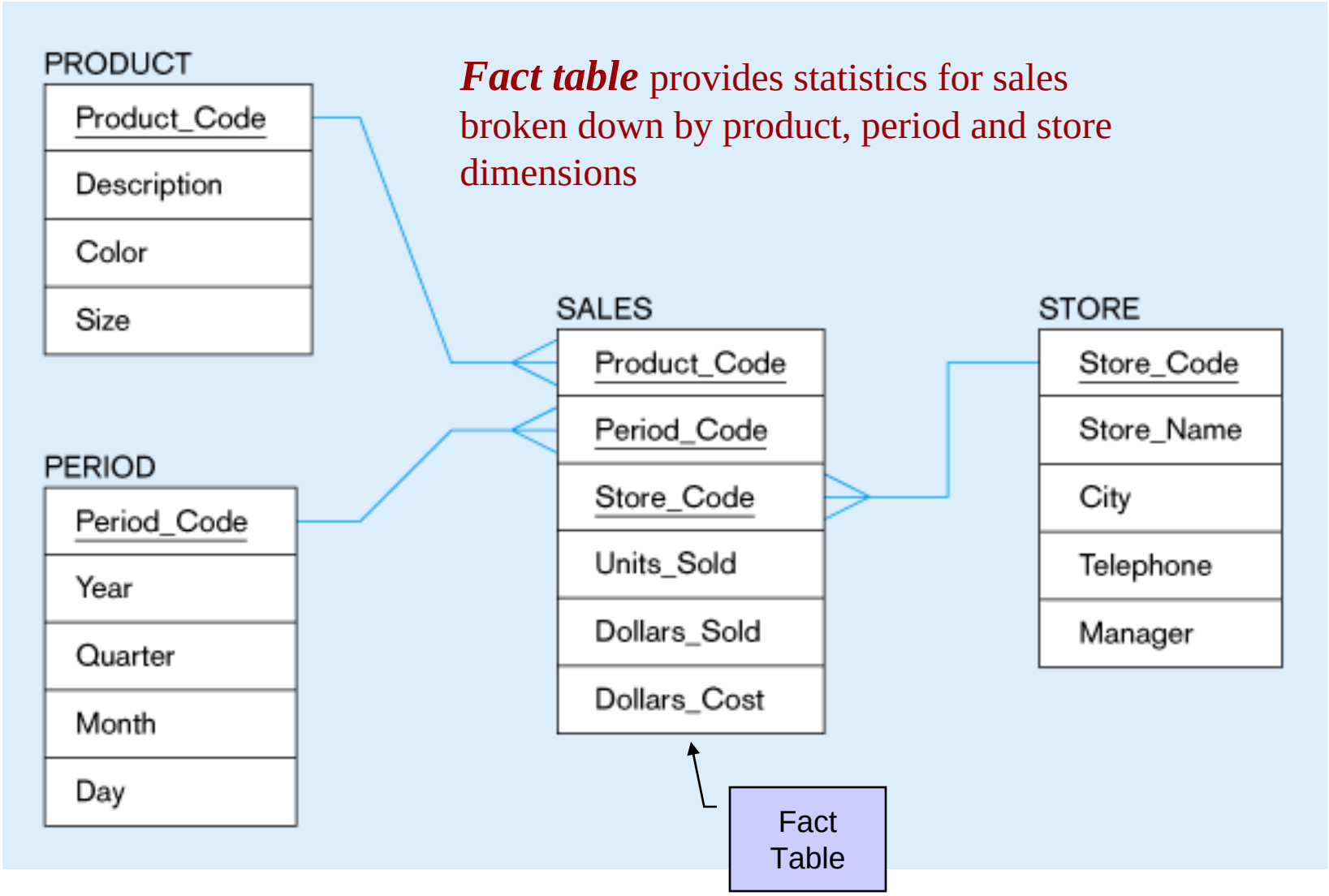
An Aside on the Star Schema (cont.)

- The star schema is a database design which is especially well-suited to ad-hoc queries in which dimensional data (describing how data are commonly aggregated) are separated from fact or event data (describing individual transactions).
- The star schema is not well-suited to on-line transaction processing and therefore is not typically used in operational databases.

An Aside on the Star Schema (cont.)

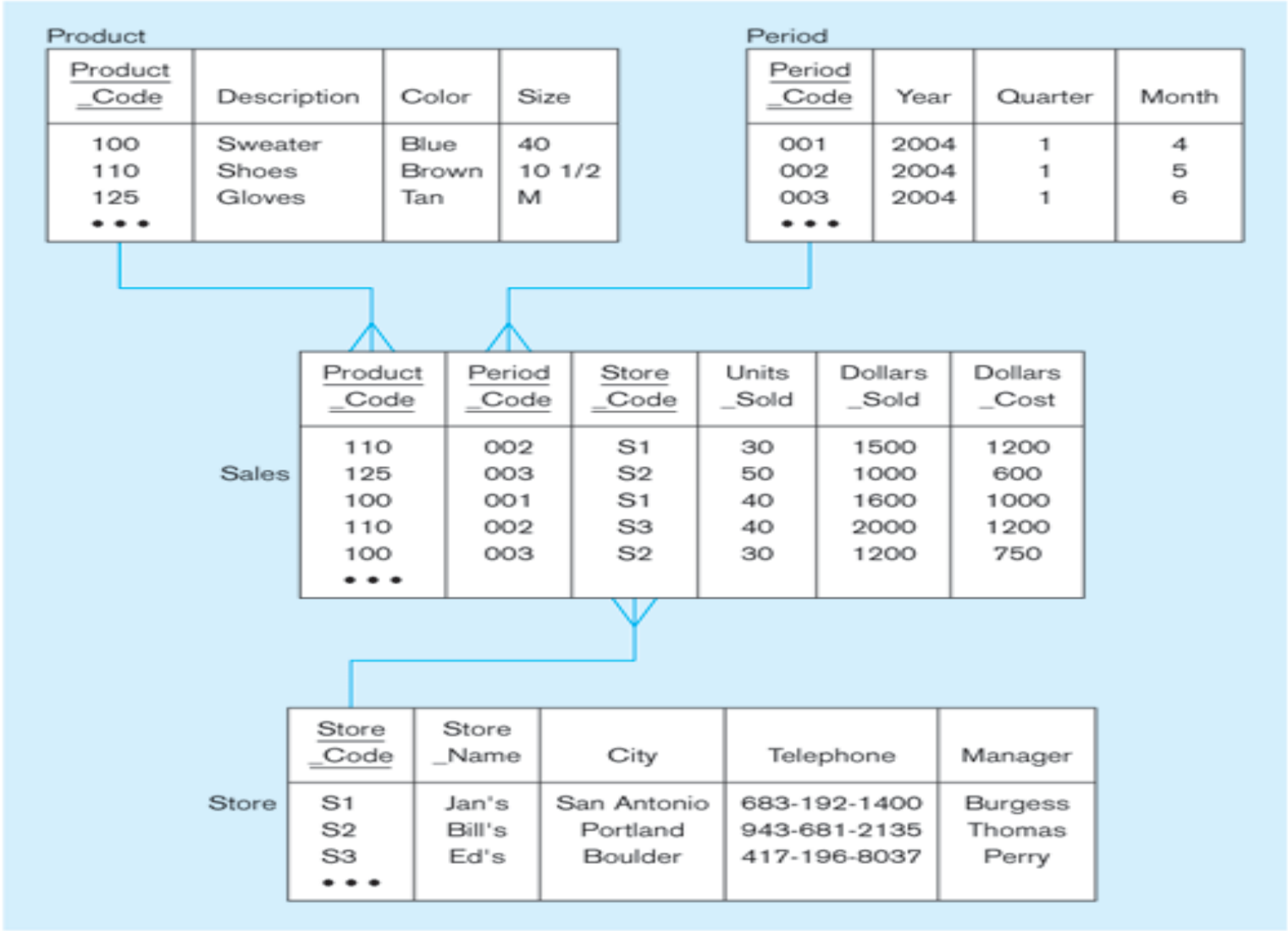


An Aside on the Star Schema (cont.)



Fact table provides statistics for sales broken down by product, period and store dimensions

An Aside on the Star Schema (cont.)



Product

Product_Code	Description	Color	Size
100	Sweater	Blue	40
110	Shoes	Brown	10 1/2
125	Gloves	Tan	M
• • •			

Period

Period_Code	Year	Quarter	Month
001	2004	1	4
002	2004	1	5
003	2004	1	6
• • •			

Sales

Product_Code	Period_Code	Store_Code	Units_Sold	Dollars_Sold	Dollars_Cost
110	002	S1	30	1500	1200
125	003	S2	50	1000	600
100	001	S1	40	1600	1000
110	002	S3	40	2000	1200
100	003	S2	30	1200	750
• • •					

Store

Store_Code	Store_Name	City	Telephone	Manager
S1	Jan's	San Antonio	683-192-1400	Burgess
S2	Bill's	Portland	943-681-2135	Thomas
S3	Ed's	Boulder	417-196-8037	Perry
• • •				

An Aside on the Star Schema (cont.)

- Dimension table keys must be ***surrogate*** (non-intelligent and non-business related), because:
  - Keys may change over time.
  - Length/format consistency.
- Granularity of Fact Table – what level of detail do you want?
  - Transactional grain – finest level.
  - Aggregated grain – more summarized.
  - Finer grain implies a better ***market basket analysis*** capability.
  - Finer grain implies more dimension tables, more rows in fact table.
- Duration of the database – how much history should be kept?
  - Natural duration – 13 months or 5 quarters.
  - Financial institutions may need longer duration.
  - Older data is more difficult to source and cleanse.

Relational OnLine Analytical Processing (ROLAP)

- The star schema is designed to optimize data query operations rather than data update operations. Naturally, changing the data design foundation means that the tools used to access such data will have to change. End users familiar with the traditional relational query tools will discover that these tools will not work efficiently with the star schema.
- ROLAP, however, saves the day by adding support for the star schema to use familiar query tools.
- ROLAP provides advanced data analysis functions, and improves query optimization and data visualization methods.
- Another criticism of RDBMs is that SQL is not suited for performing advanced data analysis. Most of the decision support data requests require the use of multiple-pass SQL queries or multiple nested SQL statements.

Relational OnLine Analytical Processing (ROLAP)

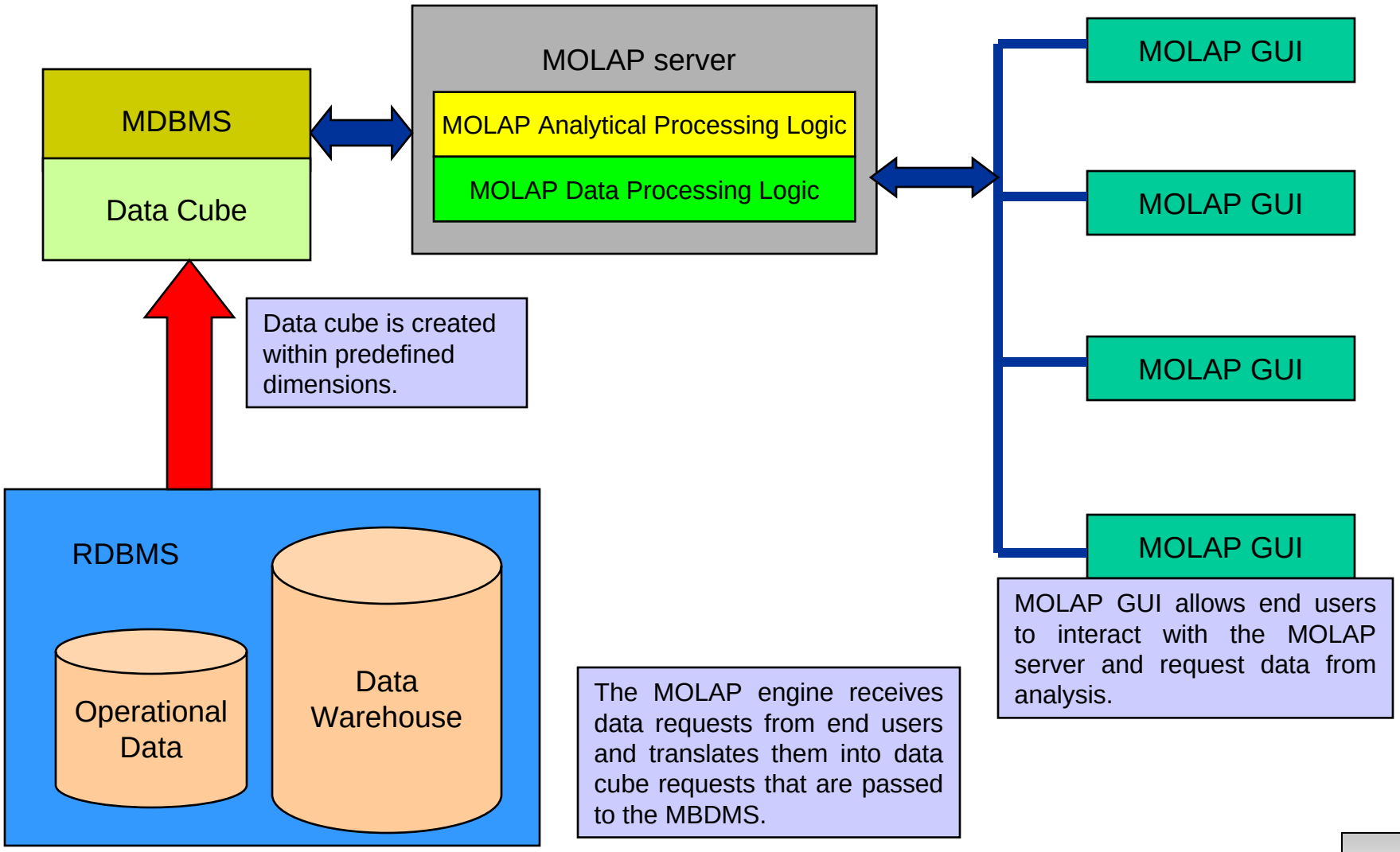
- To answer this criticism, ROLAP extends SQL so that it can differentiate between access requirements for data warehouse data (based on the star schema) and operational data (based on normalized tables). In this fashion, a ROLAP system can properly generate the SQL code required to access the star schema data.

- Query performance is also enhanced because the query optimizer is modified so that it can identify the SQL-code's intended query targets. For example, if the query target is the data warehouse, the optimizer passes the request to the data warehouse. However, if the end user performs drill-down queries against operational data, the query optimizer identifies this operation and properly optimizes the SQL request before passing them through to the operational DBMS.

Multidimensional OnLine Analytical Processing (MOLAP)

- Multidimensional OnLine Analytical Processing (MOLAP) extends OLAP functionality to multidimensional database management systems (MDBMSs).
- An MDBMS typically employs proprietary techniques to store data in matrix-like *n*-dimensional arrays.
- Many of the techniques in MDBMS are derived from CAD/CAM techniques and GIS (Geographic Information Systems).
- Conceptually, MDBMS end users visualize the stored data as a three-dimensional cube known as a data cube. The location of each data value in the data cube is a function of the x, y, and z axes in three-dimensional space.
- The data cubes can grow to *n*-dimensions, thus becoming hypercubes.
- Data cubes are created by extracting data from operational databases or from the data warehouse. An important characteristic of a data cube is that it is static. They are not subject to change and must be created before use. They cannot be created by ad hoc queries.

MOLAP System



Relational vs. Multidimensional OLAP

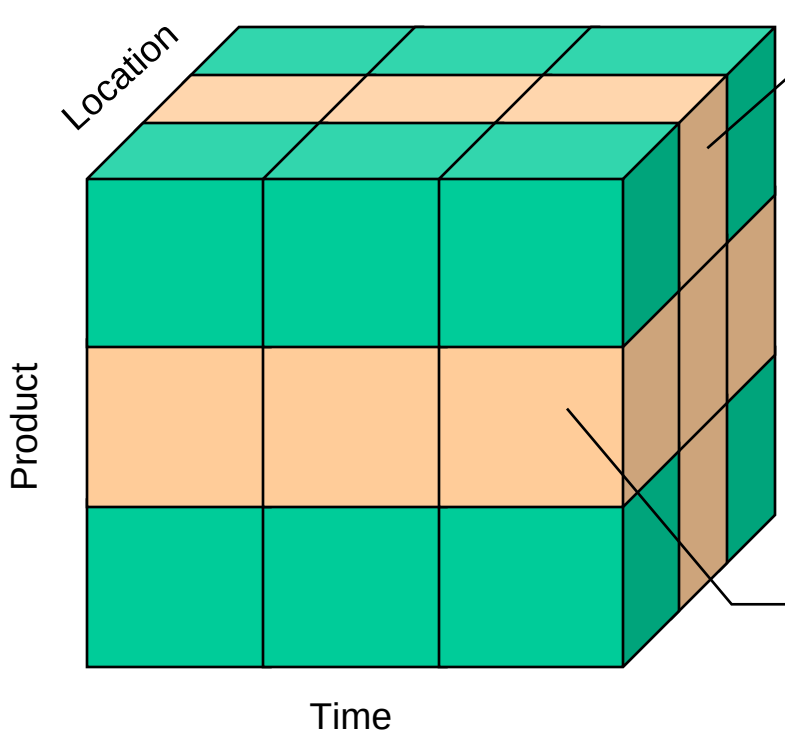
Characteristic	ROLAP	MOLAP
Schema	Uses star schema. Additional dimensions added dynamically	Uses data cubes Additional dimensions require re-creation of the data cube
Database Size	Medium to large	Small to medium
Architecture	Client/server Standards based Open	Client/server Proprietary
Access	Supports ad hoc requests Unlimited dimensions	Limited to pre-defined dimensions
Resources	High	Very high
Flexibility	High	Low
Scalability	High	Low
Speed	Good with small data data sets; average for medium to large data sets	Faster for small to medium data sets; average for large data sets.

Three Dimensional View of Data

Location: possible attributes – region, state, city, store, etc.

Product: possible attributes – product type, id, brand, color, size.

Time: possible attributes – year, quarter, month, week, day, time of day, etc.



Sales manager's view of sales data

Product manager's view of sales data

Slice and Dice Operation