

Full Notes

Objective

- Introduce the basic concept and terminology relating to Data Warehousing

Outcome (what you can understand)

- Meaning of Data warehouse
- Evolution of Data warehouse

A data warehouse is a collection of corporate information, derived directly from operational system and some external data sources

Introduction

- List all kind of data being generated in your organization
- Do you know a tool/concept to organize (capture, process and store) all those data?
- The new technology that does the above is "Data Warehouse"
- One of the earliest predecessor to Data Warehouse is "Lotus"
- Why I need to organize all data?

Meaning of Data Warehousing

- Any centralized data repository which can be queried for business benefit
- It is possible to extract archived operational data and overcome inconsistencies between different legacy data formats as well as integrating data throughout an enterprise regardless of location format or communication requirements

- The logical link between what the managers see in their decision Support EIS application and the company's operational activities
- The data warehouse provides data that is already transformed and summarized therefore making it an appropriate environment for the more efficient DSS and EIS applications

History of Data warehousing

(Review of historical management schemes of the analysis data)

- Throughout the history of systems development - the primary emphasis had been given to the operational systems and the data they process
- The fundamental requirements of the operational and analysis systems are different: the operational systems need performance, whereas the analysis systems need flexibility and broad scope
- Data from Legacy Systems:
 - 1970 - IBM Mainframe - Cobol, CICS, IMS, DB2
 - 1980 - AS400, VAS/VMS
 - 1985 - UNIX servers
- The data stored in such legacy systems ultimately becomes remote and becomes difficult to get at.
- Personal Computers – use desktop database programs, spreadsheets for business analysis and graphical representation
- The disadvantage of the above is that it leaves the data fragmented and oriented towards very specific needs.
- Each individual user has obtained only the information that she/he requires.
- The extracts are unable to address the requirements of multiple users and use.
- The time and cost involved in addressing the requirements of only one user are large.
- The disadvantages faced it led to the development of **Data Warehousing**

Factors, which lead to Data Warehousing

- The most important factor - advancement in hardware and software technologies.
 - Powerful Preprocessors
 - Inexpensive disks
 - Desktop powerful for analysis tools
 - Server software
- Availability of affordable and easy-to-use reporting and analysis tools
- Emergence of standard business applications – SAP AG, Baan, Peoplesoft, Oracle
- Technology oriented end users

What is a Data Warehouse?

- A data warehouse is a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision-making process
- A decision support database that is maintained separately from the organization's operational database
- Support information processing by providing a solid platform of consolidated, historical data for analysis
- Data Warehousing: the process of constructing and using data warehouses

Operational vs Informational Systems

- "Operational systems" run the enterprise day-to-day operations.
Ex: "order entry", "inventory", "manufacturing", "payroll", "accounting"
- Other Functions of Enterprise: planning, forecasting, and managing the organization. Ex: "marketing planning", "engineering planning" and "financial analysis"
- The above functions require the support of information systems

- “Informational systems” have to do with analyzing data and making decisions, often major decisions about how the enterprise will operate, now and in the future
 - Not only do informational systems have a different focus from operational ones, they often have a different scope
 - Where operational data needs are normally focused upon a single area, informational data needs often span a number of different areas and need large amounts of related operational data.
 - Data Warehousing has grown rapidly from a set of related ideas into architecture for data delivery for enterprise end-user computing.
 - Data warehouse support high-performance demands on an organization’s data and information - OLAP , DSS and Data Mining applications.
 - Traditional databases support On-Line Transaction Processing (OLTP), which includes insertions, updates, and deletions, while also supporting information query requirements
 - Traditional relational databases are optimized to process queries that may touch a small part of the database and transactions that deal with insertions or updates of a few tuples per relation to process.
 - By contrast, data warehouses are designed precisely to support efficient extraction, processing, and presentation for analytic and decision-making purposes
-

Objective

- To introduce OLTP
- To compare OLTP and Data warehouse
- To introduce OLAP

OLTP vs Datawarehouse

- A database which is built for online transaction processing, OLTP, is generally regarded as inappropriate for warehousing as they have been designed with a

different set of need in mind i.e., maximizing transaction capacity and typically having hundreds of table in order not to look out user, etc...

- Data warehouses are interested in query processing as opposed to transaction processing.
- OLTP systems cannot be receptacle stores of repositories of facts and historical data for business analysis.
- Basically OLTP offers large amounts of raw data, which is not easily understood
- OLTP cannot quickly answer ad-hoc queries where rapid retrieval is almost impossible.
- OLTP deals with the data which is inconsistent and changing, duplicate entries exist, entries can be missing and there is an absence of historical data, which is necessary to analyses trends
- The data warehouse offers the potential to retrieve and analyze information quickly and easily.

	OLTP	Data Warehouse
Purpose	Run day-to-day operation	Information retrieval and analysis
Structure	RDBMS	RDBMS
Data Model	Normalized	Multi-dimensional
Access	SQL	SQL plus data analysis extensions
Type of Data	Data that run the business	Data that analyses the business
Condition of Data	Changing incomplete	Historical descriptive

“The data warehouse serve a different purpose from that of OLTP systems by allowing business analysis queries to be answered as opposed to “simple aggregation” such as ‘what is the current account balance for this customer?’ Typical data warehouse queries include such things as ‘which product line sells best in middle America and how dose this correlate to demographic data? “

Processes in Data warehousing OLTP

- The first step in data warehousing is to “insulate” your current operational information, i.e. to preserve the security and integrity of mission-critical OLTP applications, while giving you access to the broadest possible base of data.
- Data warehousing needs to store and retrieve massive amounts of information. Increasingly, large organizations have found that only parallel processing systems offer sufficient bandwidth.
- The data warehouse thus retrieves data from a varsity of a heterogeneous operational database.
- The data is then transformed and delivered to the data warehouse/ store based in a selected modal (or mapping definition).
- The information that describes the modal metadata is the means by which the end-user finds and understands the data in the warehouse
- The metadata should at least contain
 - Structure of the data;
 - Algorithm used for summarization;
 - Mapping from the operational environment to the data warehouse.
- Data Cleansing - the removal of creation aspects Operational data such as low-level transaction information which slows down the query times.
- The cleansing stage should accommodate all types of queries even those, which may require low-level information.

- Data should be extracted from production sources at regular intervals and pooled centrally but the cleansing process has to remove duplication and reconcile differences between various styles of data collection.
 - Once the data has been cleaned it is then transferred to the data warehouse, which typically is a large database on a high-performance box, either SMP Symmetric Multi-Processing or MPP, Massively parallel Processing
 - Parallel Processing is another important aspect of data warehousing because of the complexity involved in processing ad-hoc queries and because of the vast quantities of data that the organization wants to use in the warehouse.
 - A data warehouse can be a central store or data mart.
 - A data mart - provide subsets of the main store and summarized information depending on the requirements of a specific group/ department.
 - The central stores approach generally uses simple data structure with very little assumptions about the relationships between data
 - Marts often uses a multidimensional database that can speed up query processing as it can have data structures that reflect the most likely questions.
-

Objective

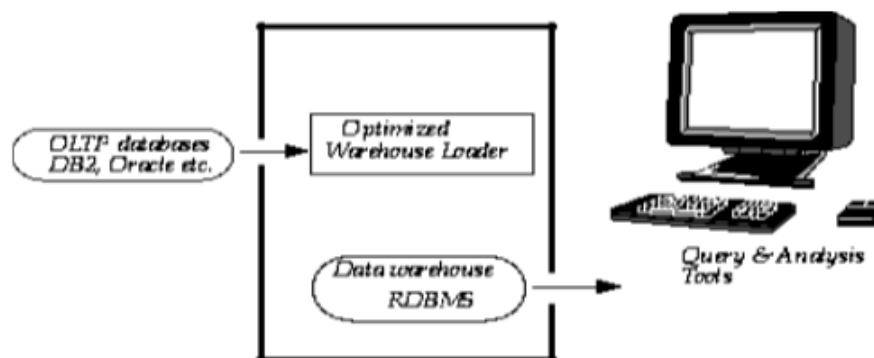
- To understand data warehouse models
- To explain multidimensional models and schemas

Data Warehousing

“Data warehousing is the process of extracting and transforming operational data into informational data and loading it into a central data store or warehouse

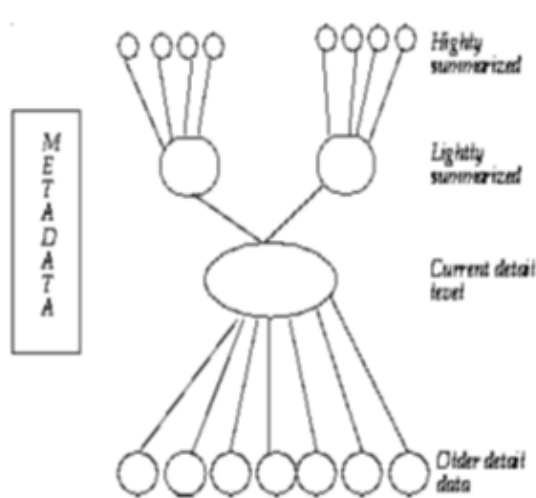
Once the data is loaded it is accessible via desktop query and analysis tools by the decision makers“

Data Warehouse Model



The data within the actual warehouse itself has a distinct structure with the emphasis on different levels of summarization

Structure of data inside the data warehouse



The current detail data is central as it :

- Reflects the most recent happenings, which are usually the most interesting;
- It is voluminous as it is stored at the lowest level of granularity;
- it is always (almost) stored on disk storage which is fast to access but expensive and complex to manage.

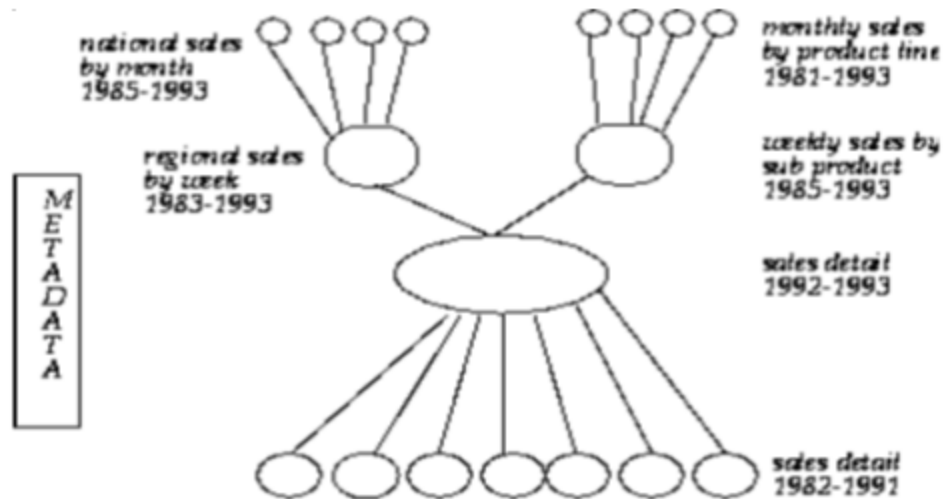
Older detail data is stored on some form of mass storage, it is infrequently accessed and stored at a level detail consistent with current detailed data

- Lightly summarized data is data distilled from the low level of detail found at the current detailed level and generally is stored on disk storage.
- When building the data warehouse, things to be considered –the unit of time needed for summarization, the contents to be summarized or the attributes of the summarized data.
- Highly summarized data is compact and easily accessible and can even be found outside the warehouse.

Metadata is the final component of the data warehouse and is really of a different dimension in that it is not the same as data drawn from the operational environment but is used as:

- a directory to help the DSS analyst locate the contents of the data warehouse,
- a guide to the mapping of data as the data is transformed from the operational environment to the data warehouse environment,
- a guide to the algorithms used for summarization between the current detailed data and the lightly summarized data and the lightly summarized data and the highly summarized data, etc.

An example of levels of summarization of data inside the data warehouse



Data Modelling for Data Warehouses

- **Multidimensional models** take advantage of inherent relationships in data to populate data in multidimensional matrices called data cubes
- These may be called hypercube if they have more than three dimensions
- For data that lend themselves to dimensional Formatting, query performance in multidimensional matrices can be much better than in the relational data model.
- The example of 3 dimensions in a corporate data warehouse would be the corporation's fiscal periods, products, and regions.

Three Dimensions Data Modelling

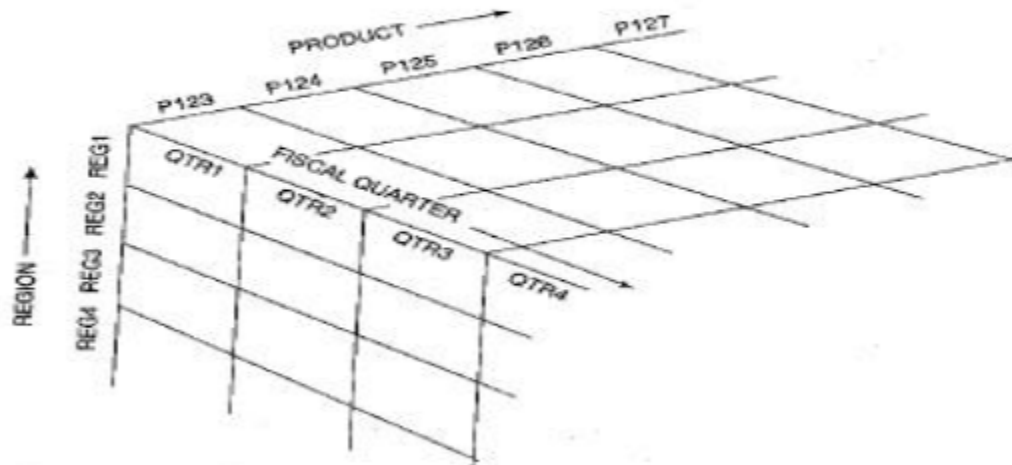


Figure 26.4 Pivoted version of the data cube.

Features of Multidimensional Data Models

- Changing from one dimensional hierarchy -(orientation) to another is easily accomplished in a data cube by a technique called pivoting (also called rotation).
- In this technique, the data cube can be thought of as rotating to show a different orientation of the axes.
- For example, you might pivot the data cube to show regional sales revenues as rows, the fiscal quarter revenue totals as columns, and company's products in the third dimension.
- This technique is equivalent to having a regional sales table for each product separately

Multidimensional models lend themselves readily to hierarchical views in what is known as **roll-up display and drill-down display**.

- **Roll-up display** moves up the hierarchy, grouping into larger units along a dimension (e.g., summing weekly data by quarter, or by year). One of the

above figures shows a rollup display that moves from individual products to a coarser grain of product categories.

- A **drill-down display** provides the opposite capability, furnishing a finer-grained view, perhaps disaggregating country sales by region and then regional sales by sub region and also breaking up products by styles.

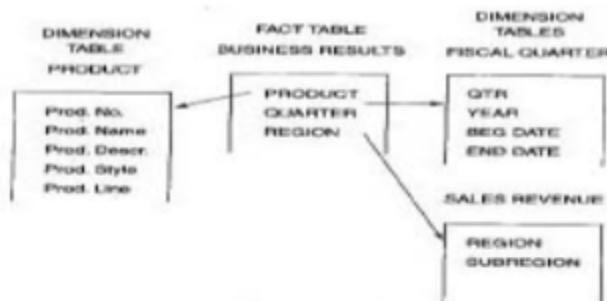


Figure 26.7 A star schema with fact and dimensional tables.

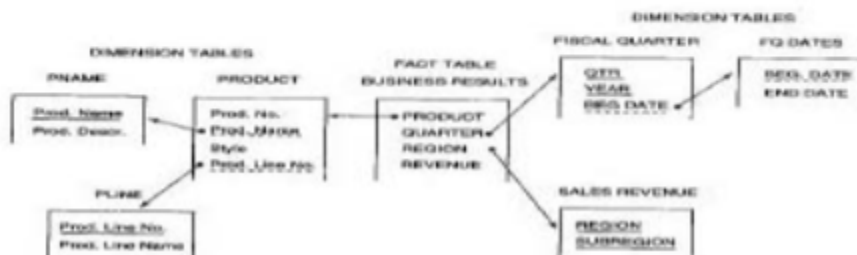


Figure 26.8 A snowflake schema.

Fact Constellation

- A fact constellation is a set of fact tables that share some dimension tables
- Following figure shows a fact constellation with two fact tables, business results and business forecast. These share the dimension table called product
- Fact constellations limit the possible queries for the ware-house.

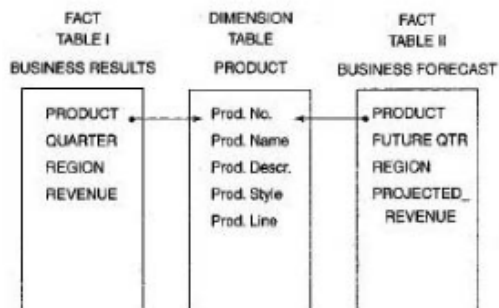


Figure 26.9 A fact constellation.

- A fact constellation is a set of fact tables that share some dimension tables
- Following figure shows a fact constellation with two fact tables, business results and business forecast. These share the dimension table called product
- Fact constellations limit the possible queries for the ware-house.

Bitmap Indexing

- Data warehouse storage also utilizes indexing techniques to support high performance access.
- A technique called bitmap indexing constructs a bit vector for each value in a domain (column) being indexed.
- It works very well for domains of low-cardinality.
- There is a 1 bit placed in the jth position in the vector if the jth row contains the value being indexed.
- For example, imagine an inventory of 100,000 cars with a bitmap index on car size.
- If there are four-car sizes--economy, compact, midsize, and full size--there will be four bit vectors, each containing 100,000 bits (12.5 K) for a total index size of 50K.
- Bitmap indexing can provide considerable input/output and storage space advantages in low-cardinality domains.
- With bit vectors a bitmap index can provide dramatic improvements in comparison, aggregation, and join performance
- In a star schema, dimensional data can be indexed to tuples in the fact table by join indexing

- Join indexes are traditional indexes to maintain relationship between primary key and foreign key values
 - They relate the values of a dimension of a star schema to rows in the fact table
 - For example, consider a sales fact table that has city and fiscal quarter as dimensions. If there is a join index on city for each city the join index maintains the tuple IDs of tuples containing that city.
 - Join indexes may involve multiple dimensions.
 - Data warehouse storage can facilitate access to summary data by taking further advantage of the nonvolatility of data warehouses and a degree of predictability of the analyses that will be performed using them. Two approaches have been used.
 - (1) smaller tables including summary data such as quarterly sales or revenue by product line
 - (2) encoding of level (e.g., weekly, quarterly, annual) into existing tables.
 - By comparison, the overhead of creating and maintaining such aggregations would likely be excessive in a volatile, transaction-oriented database.
-

Objective

- To learn the basic structure of a Data warehouse
- To understand data warehouse physical architecture
- To know various principles of a Data warehousing

“A data warehouse is a data base that collects current information, transforms it to ways it can be used by the warehouse owner, transforms that information for clients, and offers portals of access to members of your firm to help them make decisions and future plans.”

“Data warehousing is the technology trend most often associated with enterprise computing today. The term conjures up images of vast data banks fed from systems all over the globe, with legions of corporate analysts mining them for golden nuggets of information that will make their companies more profitable“

Purpose of Data warehouse

“Essentially, a data warehouse provides historical data for decision-support applications. Such applications include reporting, online analytical processing (OLAP), executive information systems (EIS), and data mining.”

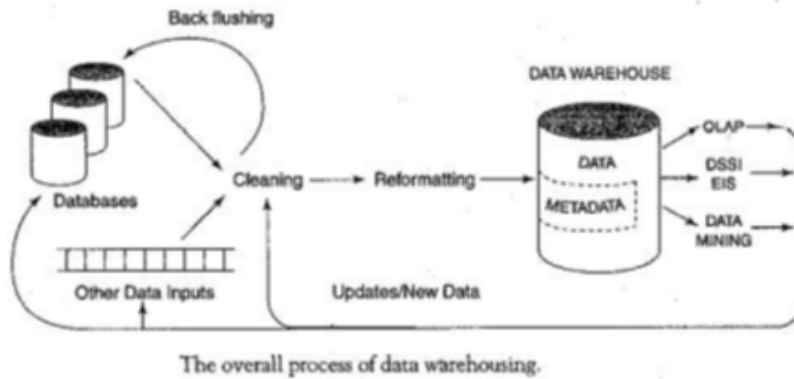
- According to W. H. Inmon, the man who originally came up with the term, a data warehouse is a centralized, integrated repository of information.
- Here, integrated means cleaned up, merged, and redesigned. This may be more or less complicated depending on how many systems feed into a warehouse and how widely they differ in handling similar information.

Production Databases vs Data warehouses

- Data warehouses differ from production databases, or online transaction-processing (OLTP) systems, in their purpose and design.
- An OLTP system is designed and optimized for data entry and updates, whereas a data warehouse is optimized for data retrieval and reporting, and it is usually a read-only system.
- An OLTP system contains data needed for running the day-today operations of a business but a data warehouse contains data used for analyzing the business.
- The data in an OLTP system is current and highly volatile, which data elements that may be incomplete or unknown at the time of entry. A warehouse contains historical, nonvolatile data that has been adjusted for transactions errors.
- OLTP systems and data warehouses use different data-modeling strategies.

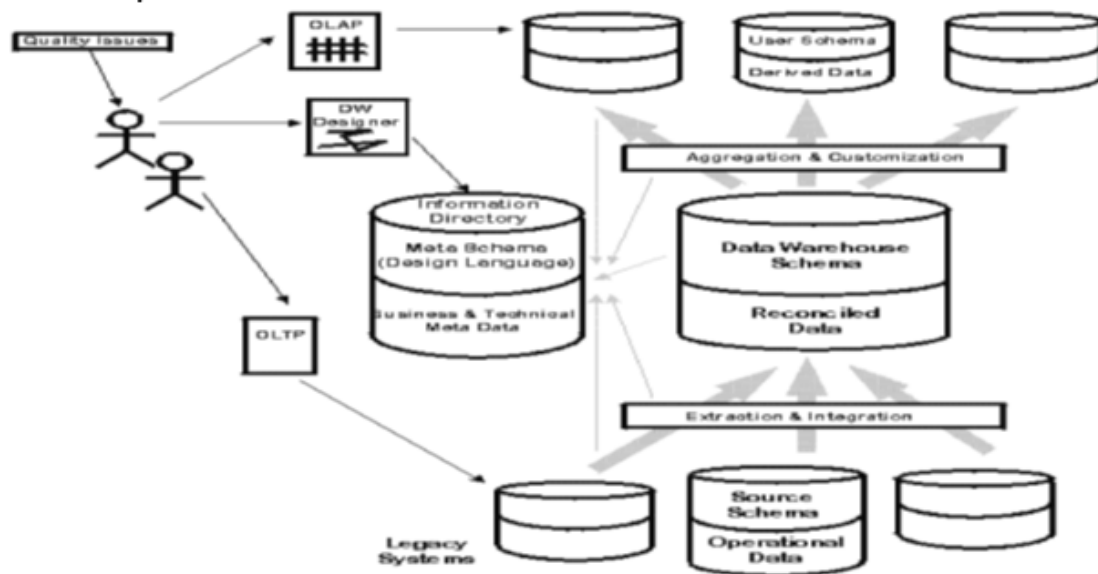
- Redundancy is almost nonexistent in OLTP systems, since redundant data complicates updates. But redundancy is desirable in a data warehouse, since it simplifies user access and enhances performance by minimizing the number of tables that have to be joined.
- OLTP systems are highly normalized and are usually based on a relational model. Some data warehouses don't use a relational model at all, preferring a multidimensional design instead.
- The multidimensional data model is a good fit for OLAP and decision-support technologies.
- In contrast to multi-databases, which provide access to disjoint and usually heterogeneous databases, a data warehouse is frequently a store of integrated data from multiple sources, processed for storage in a multidimensional model.
- Unlike most transactional databases, data warehouses typically support time-series and trend analysis, both of which requires more historical data than are generally maintained in transactional databases.
- Compared with transactional databases, data warehouses are nonvolatile. That means that information in the data warehouse changes far less often and may be regarded as non-real-time with periodic updating
- In transactional systems, transactions are the unit and are the agent of change in a database; by contrast, data warehouse information is much more coarse grained and is refreshed according to a careful choice of refresh policy, usually incremental.
- Warehouse updates are handled by the warehouse's acquisition component that provides all required preprocessing.

Conceptual structure of a data warehouse



Being basically dependent on architecture in concept, a Data Warehouse - or an OLAP system - is designed by applying data warehousing concepts on traditional database systems and using appropriate design tools.

Example –IMF Data Warehouse



Example - IMF Data Warehouse

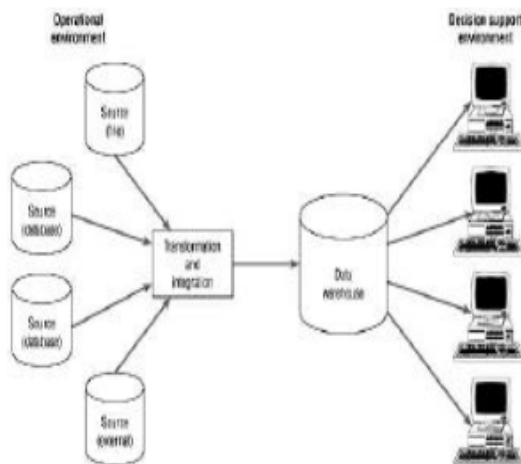
- Data Warehouses and OLAP applications designed and implemented comply with the adopted methodology by IMF.

- The final deployment takes place through the use of specialized data warehouse and OLAP systems, namely MicroStrategy's DSS Series.
- MicroStrategy Inc. is one of the most prominent and accepted international players on data warehousing systems and tools, offering solutions for every single layer of the DW architecture hierarchy.

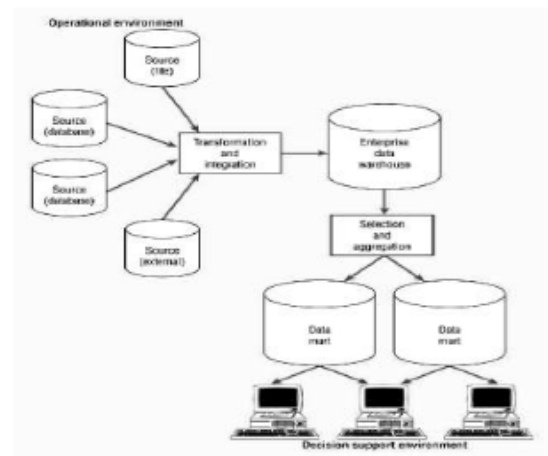
Data Warehouse Physical Architectures

- Generic Two-Level
- Expanded Three-Level
- Enterprise data warehouse (EDW) - single source of data for decision making
- Data marts - limited scope; data selected from EDW

Generic Two Level Architecture



Expanded Three Level Architecture

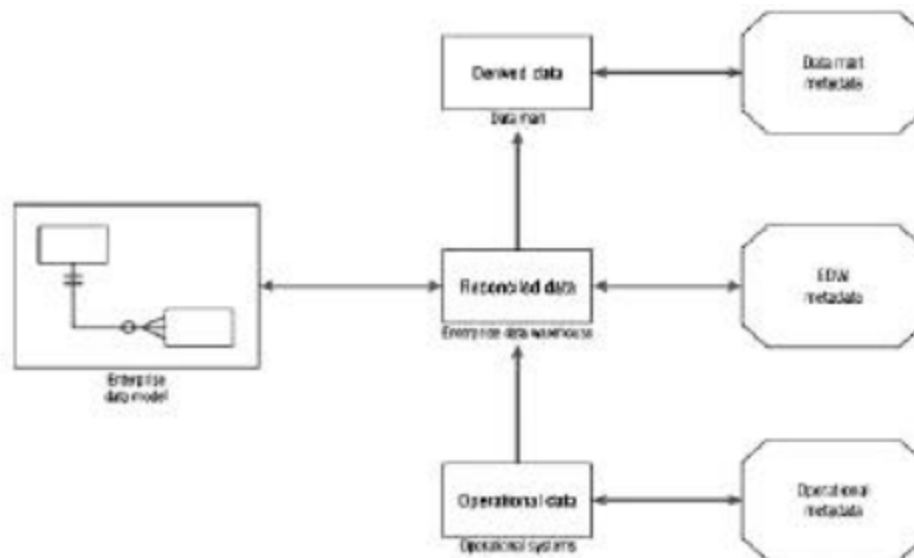


Three-Layer Data Architecture

Associated with the three-level physical architecture

- Operational Data

- Stored in the various operational systems throughout the organization
- Reconciled Data
 - The data stored in the enterprise data warehouse
 - Generally not intended for direct access by end users
- Derived Data
 - The data stored in the data marts
 - Selected, formatted, and aggregated for end user decision-support applications



Principles of a Data Warehousing

- **Load Performance**

Data warehouses require increase loading of new data on a periodic basic within narrow time windows; performance on the load process should be measured in hundreds of millions of rows and gigabytes per hour and must not artificially constrain the volume of data business.

- **Load Processing**

Many steps must be taken to load new or update data into the data warehouse including data conversion, filtering, reformatting, indexing and metadata update.

- **Data Quality Management**

Fact-based management demands the highest data quality. The warehouse must ensure local consistency, global consistency, and referential integrity despite “dirty” sources and massive database size.

- **Query Performance**

Fact-based management must not be slowed by the performance of the data warehouse RDBMS; large, complex queries must be complete in seconds not days.

- **Terabyte Scalability**

Data warehouse sizes are growing at astonishing rates. Today these range from a few to hundreds of gigabytes and terabyte-sized data warehouses

Objective

- To learn the basic structure of a Data warehouse
- To understand data warehouse physical architecture
- To know various principles of a Data warehousing

Data Warehouse

- "A data warehouse is a database that collects current information, transforms it to ways it can be used by the warehouse owner, transforms that information for clients, and offers portals of access to members of your firm to help them make decisions and future plans."

Data Warehousing

- Data warehousing is the technology trend most often associated with enterprise computing today. The term conjures up images of vast data banks fed from systems all over the globe, with legions of corporate analysts mining them for golden nuggets of information that will make their companies more profitable

Purpose of Data Warehouse

“Essentially, a data warehouse provides historical data for decision-support applications. Such applications include reporting, online analytical processing (OLAP), executive information systems (EIS), and data mining.”

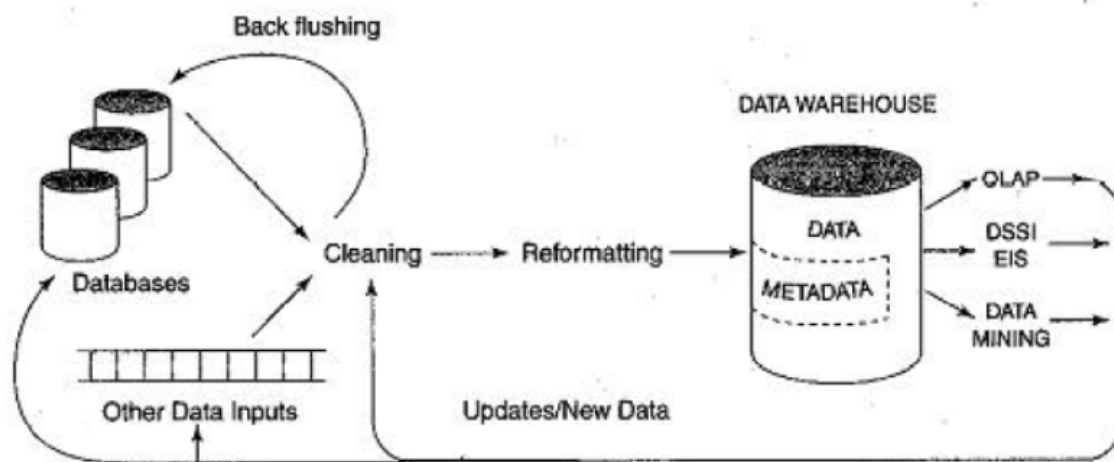
- According to W. H. Inmon, the man who originally came up with the term, a data warehouse is a centralized, integrated repository of information.
- Here, integrated means cleaned up, merged, and redesigned. This may be more or less complicated depending on how many systems feed into a warehouse and how widely they differ in handling similar information.

Production Databases Vs Data Warehouses

- Data warehouses differ from production databases, or online transaction processing (OLTP) systems, in their purpose and design.
- An OLTP system is designed and optimized for data entry and updates, whereas a data warehouse is optimized for data retrieval and reporting, and it is usually a read-only system.
- An OLTP system contains data needed for running the day-to-day operations of a business but a data warehouse contains data used for analyzing the business.
- The data in an OLTP system is current and highly volatile, which data elements that may be incomplete or unknown at the time of entry. A warehouse contains historical, nonvolatile data that has been adjusted for transaction errors.

- OLTP systems and data warehouses use different data-modeling strategies.
- Redundancy is almost nonexistent in OLTP systems since redundant data complicates updates. But redundancy is desirable in a data warehouse since it simplifies user access and enhances performance by minimizing the number of tables that have to be joined.
- OLTP systems are highly normalized and are usually based on a relational model. Some data warehouses don't use a relational model at all, preferring a multidimensional design instead.
- The multidimensional data model is a good fit for OLAP and decision-support technologies.
- In contrast to multi-databases, which provide access to disjoint and usually heterogeneous databases, a data warehouse is frequently a store of integrated data from multiple sources, processed for storage in a multidimensional model.
- Unlike most transactional databases, data warehouses typically support time-series and trend analysis, both of which require more historical data than are generally maintained in transactional databases.
- Compared with transactional databases, data warehouses are nonvolatile. That means that information in the data warehouse changes far less often and may be regarded as non-real-time with periodic updating
- In transactional systems, transactions are the unit and are the agent of change a database; by contrast, data warehouse information is much more coarse-grained and is refreshed according to a careful choice of refresh policy, usually incremental.
- Warehouse updates are handled by the warehouse's acquisition component that provides all required preprocessing.

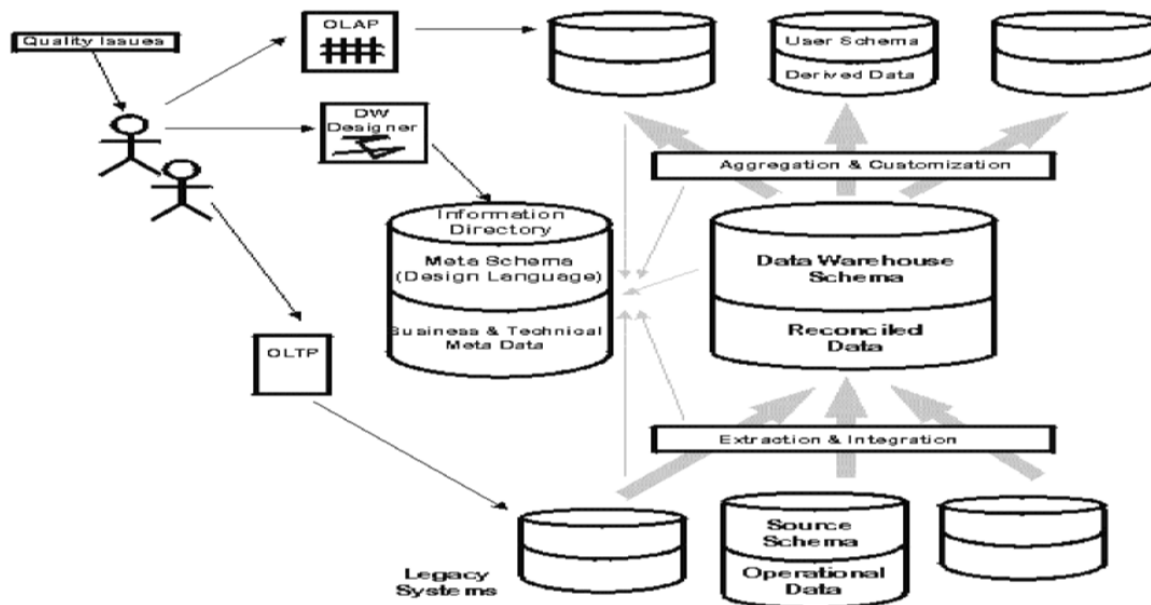
Conceptual Structure of a Data Warehouse



The overall process of data warehousing.

- Being basically dependent on architecture in concept, a Data Warehouse - or an OLAP system - is designed by applying data warehousing concepts on traditional database systems and using appropriate design tools.

Example-IMF Data Warehouse

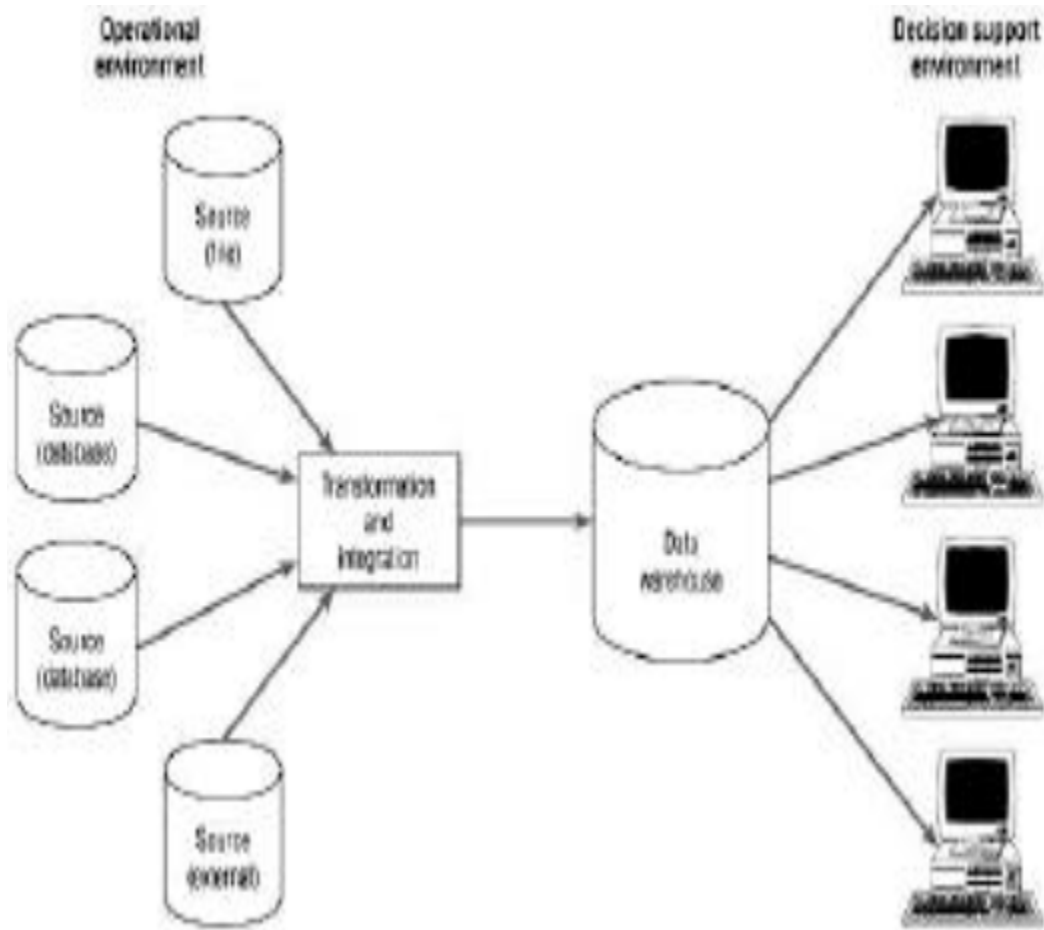


- Data Warehouses and OLAP applications designed and implemented comply with the adopted methodology by IMF.
- The final deployment takes place through the use of specialized data warehouse and OLAP systems, namely MicroStrategy's DSS Series.
- MicroStrategy Inc. is one of the most prominent and accepted international players on data warehousing systems and tools, offering solutions for every single layer of the DW architecture hierarchy.

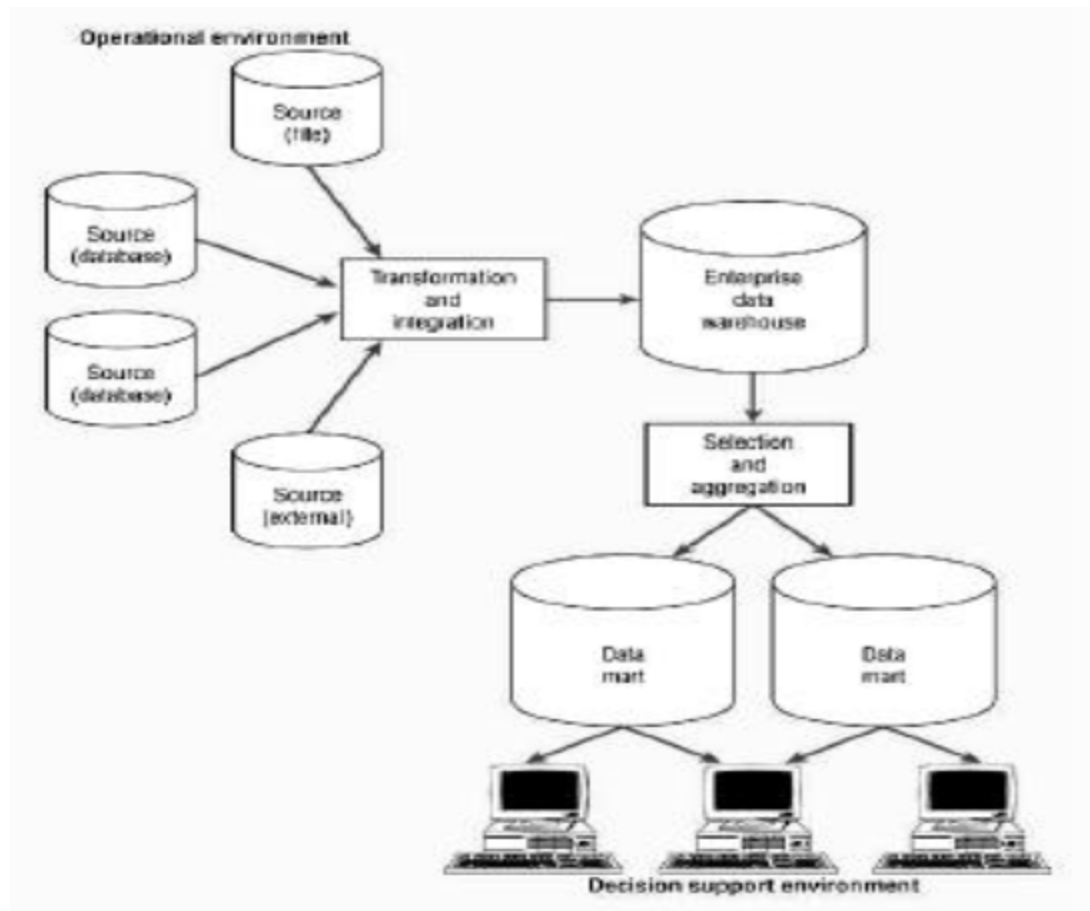
Data Warehouse Physical Architectures

- Generic Two-Level
- Expanded Three-Level
- Enterprise data warehouse (EDW) - single source of data for decision making
- Data marts - limited scope; data selected from EDW

Generic Two Level Architecture



Expanded Three Level Architecture

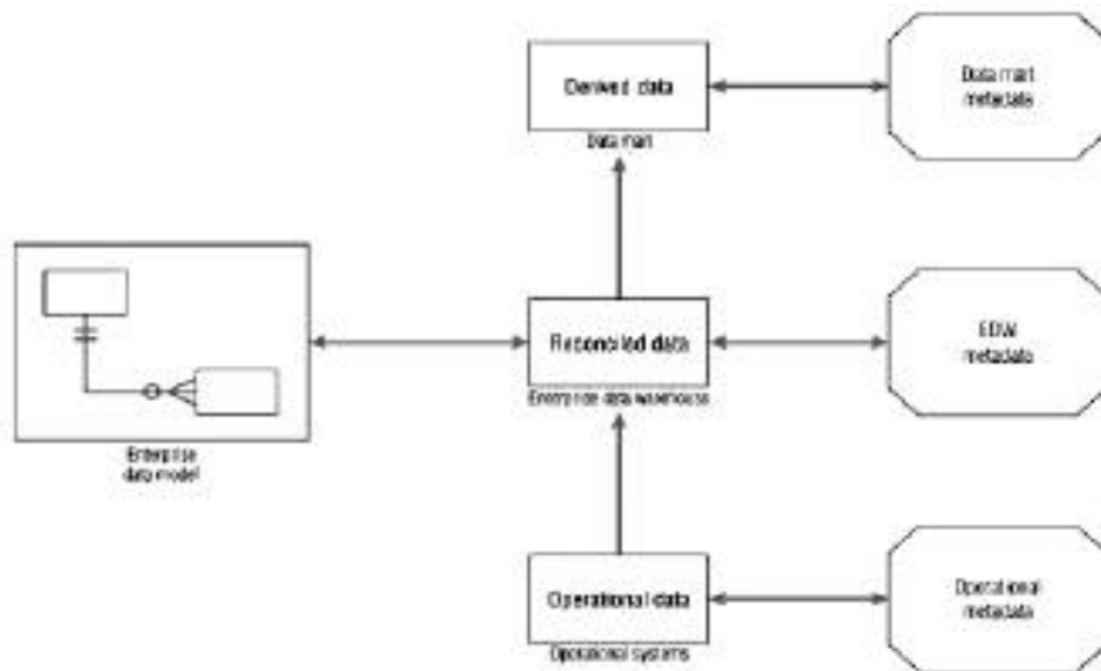


Three-Layer Data Architecture

Associated with the three-level physical architecture

- **Operational Data**
 - Stored in the various operational systems throughout the organization
- **Reconciled Data**
 - The data stored in the enterprise data warehouse
 - Generally not intended for direct access by end-users
- **Derived Data**
 - The data stored in the data marts

- Selected, formatted, and aggregated for end-user decision-support applications



Principles of a Data Warehousing

•Load Performance

Data warehouses require increased loading of new data on a periodic basis within narrow time windows; performance on the load process should be measured in hundreds of millions of rows and gigabytes per hour and must not artificially constrain the volume of data business.

•Load Processing

Many steps must be taken to load new or update data into the data warehouse including data conversion, filtering, reformatting, indexing, and metadata update.

•Data Quality Management

Fact-based management demands the highest data quality. The warehouse must ensure local consistency, global consistency, and referential integrity

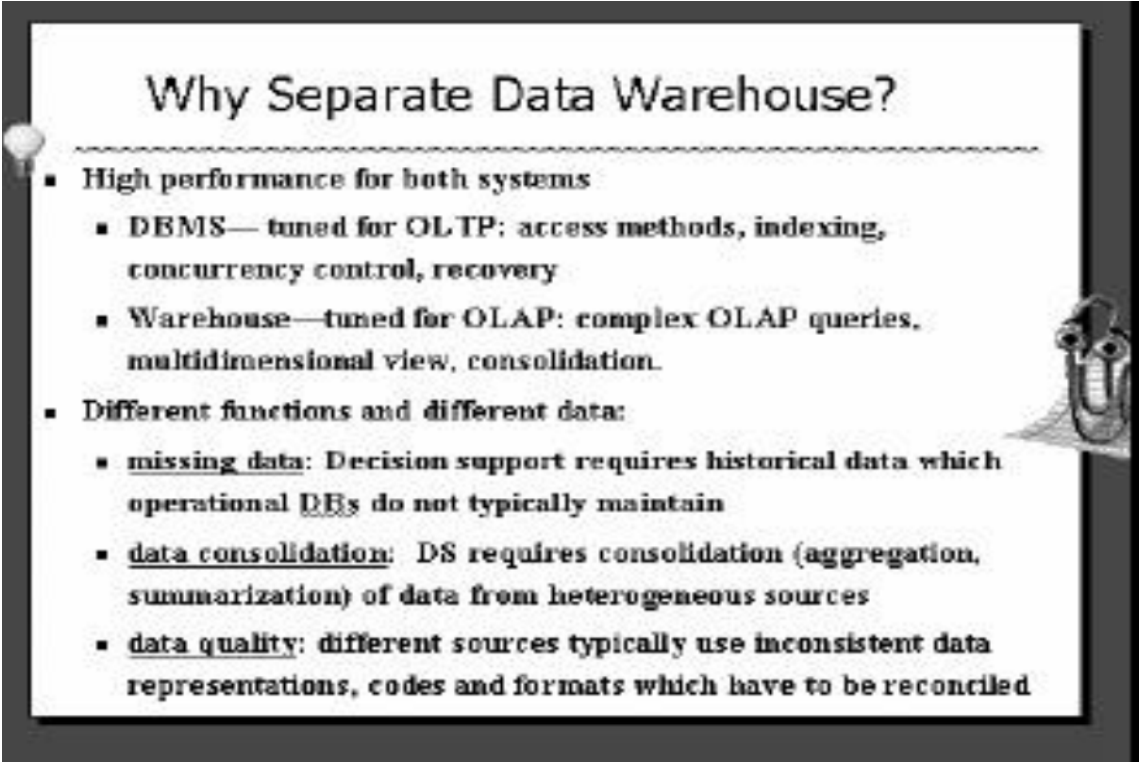
despite “dirty” sources and massive database size.

•Query Performance

Fact-based management must not be slowed by the performance of the data warehouse RDBMS; large, complex queries must be complete in seconds not days.

•Terabyte Scalability

Data warehouse sizes are growing at astonishing rates. Today these range from a few to hundreds of gigabytes and terabyte-sized data warehouses



Why Separate Data Warehouse?

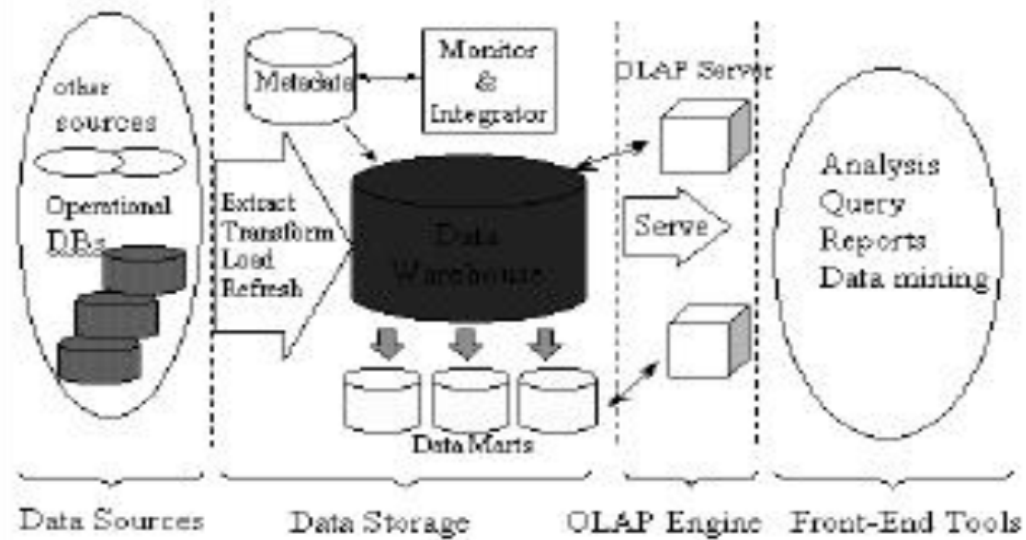
- High performance for both systems
 - DBMS—tuned for OLTP: access methods, indexing, concurrency control, recovery
 - Warehouse—tuned for OLAP: complex OLAP queries, multidimensional view, consolidation.
- Different functions and different data:
 - missing data: Decision support requires historical data which operational DBs do not typically maintain
 - data consolidation: DS requires consolidation (aggregation, summarization) of data from heterogeneous sources
 - data quality: different sources typically use inconsistent data representations, codes and formats which have to be reconciled

- 

POLYMER LETTERS



Multi-Tiered Architecture



DW Architecture

Source Systems



Metadata Mgmt.

Extensible DW Model Design

- DW projects typically can't include data from all BU apps right from start
 - Build framework for enterprise DW
 - Implement initial valued sources
 - Add additional applications as business case can be made



What it is Not

- Data warehousing is not a silver bullet
 - It will not lower your IT costs
 - It will not allow you to cut resource requirements
 - It will not fix bad data models, poor system design or data problems

Objective

- To introduce with the basic concepts behind building a data warehouse

Building a Data Warehouse

- In constructing a data warehouse, builders should take a broad view of the anticipated use of the warehouse
- There is no way to anticipate all possible queries or analyses during the design phase.
- However, the design should specifically support ad-hoc querying, that is, accessing data with any meaningful combination of values for the attributes in the dimension or fact tables
- For example; a marketing-intensive consumer-products company would require different ways of organizing the data warehouse than would a nonprofit charity focused on fundraising.
- An appropriate schema should be chosen that reflects anticipated usage.

Acquisition of data for the warehouse involves the following steps:

- The data must be extracted from multiple, heterogeneous sources; for example, databases or other data feeds such as those containing financial market data or environmental data
- Data must be formatted for consistency within the warehouse. Names, meanings, and domains of data from unrelated sources must be reconciled.
- For instance, subsidiary companies of a large corporation may have different fiscal calendars with quarters ending on different dates, making it difficult to aggregate financial data by quarter.
- Various credit cards may report their transactions differently, making it difficult to compute all credit sales.

- These format inconsistencies must be resolved.
- The data must be cleaned to ensure validity. Data cleaning is an involved and complex process that has been identified as the largest labor-demanding component of data warehouse construction
- For input data, cleaning must occur before the data are loaded into the warehouse
- Since input data must be examined and formatted consistently, data warehouse builders should take this opportunity to check for validity and quality.
- Recognizing erroneous and incomplete data is difficult to automate, and cleaning that requires automatic error correction can be even tougher
- Some aspects, such as domain checking, are easily coded into data cleaning routines, but automatic recognition of other data problems can be more challenging.
- After such problems have been taken care of, similar data from different sources must be coordinated for loading into the warehouse.
- As data managers in the organization discover that their data are being cleaned for input into the warehouse; they will be likely to upgrade their data source with the cleaned data.
- The process of returning cleaned data to the source is called **backflushing**.
- The data must be fitted into the data model of the warehouse. The data may have to be converted from relational, object-oriented, or legacy databases (network and/or hierarchical) to a multidimensional model
- The data must be loaded into the warehouse. The sheer volume of data in the warehouse makes loading the data a significant task.
- Monitoring tools for loads as well as methods to recover from incomplete or incorrect loads are required.
- With the huge volume of data in the warehouse, incremental updating is usually the only feasible approach

- The refresh policy answers to the following questions:
 - How up-to-date must the data be?
 - Can the warehouse go off-line, and for how long?
 - What are the data inter-dependencies?
 - What is the storage availability?
 - What are the distribution requirements (such as for replication and partitioning)?
 - What is the loading time (including cleaning, formatting, copying, transmitting, and overhead such as index rebuilding)?

“Databases must strike a balance between efficiency in transaction processing and supporting query requirements (ad hoc user requests), but a data warehouse is typically optimized for access from a decision maker’s, needs”

Data Storage

Data storage in a data warehouse involves the following processes:

- Storing the data according to the data model of the warehouse.
- Creating and maintaining, required data structures.
- Creating and maintaining appropriate access paths.
- Providing for time-variant data as new data are added.
- Supporting the updating of warehouse data.
- Refreshing the data.
- Purging data

Reloading Data Warehouse

- The sheer volume of data in the warehouse generally makes it impossible to simply reload the warehouse in its entirety later on.
- Alternatives include selective (partial) refreshing of data and separate warehouse versions (requiring, double, storage capacity for the warehouse)
- When the warehouse uses an incremental data refreshing mechanism, data may need to be periodically purged
- For example, a warehouse that maintains data on the previous twelve business quarters may periodically purge its data each year.

Important Design Considerations, Data Warehouse Environment

- Usage projections.
- The fit of the data model.
- Characteristics of available sources.
- Design of the metadata component.
- Modular component design.
- Design for manageability and change.
- Considerations of distributed and parallel architecture

Metadata

- Metadata is defined as - description of a database including its schema definition.
- **The metadata repository** is a key data warehouse component.
- The metadata repository includes both technical and business metadata.
- The first, technical metadata, covers details of acquisition processing, storage structures, data descriptions, warehouse operations, and

maintenance, and access support functionality

- The second, business metadata, include the relevant business rules and organizational details supporting the warehouse.

Architecture of Distributed Computing Environment

- There are two basic distributed architectures: the **distributed warehouse** and the **federated warehouse**
- For a distributed warehouse, all the issues of distributed databases are relevant, for example, replication, partitioning, communications, and consistency concerns.
- A distributed architecture can provide benefits particularly important to warehouse performance, such as improved load balancing, scalability of performance, and higher availability.
- A single replicated metadata repository would reside at each distribution site.
- The idea of the federated warehouse is like that of the federated database: a decentralized confederation of autonomous data warehouses, each with its own metadata repository.
- Given the magnitude of the challenge inherent to data warehouses, it is likely that such federations will consist of smaller-scale components, such as data marts.
- Large organizations may choose to federate data marts rather than build huge data warehouses.

Nine Decisions in the design of a Data Warehouse

1. Choosing the subject matter
2. Deciding what a fact table represents

3. Identifying and confirming the dimensions.
 4. Choosing the facts
 5. Storing pre-calculations in the fact table.
 6. Rounding out the dimension tables
 7. Choosing the duration of the database
 8. The need to track slowly changing dimensions
 9. Deciding the query priorities and the query modes
-

BUILDING A DATA WARHOUSE-2

Objective

To study about Datawarehouse applications and various approaches that are used to build a Data warehouse

Data Warehouse Application

- Data warehouse application deals with large amounts of data, which is aggregated in nature.
- A data warehouse application answers questions like
 - What is the average deposit by branch?
 - Which day of the week is busiest?
 - Which customers with high average balances currently are not participating in a checking- plus account)
- Each query is unique, the end-user interface must be flexible by design.
- A key issue in the industry today is which approach should you take when building a Decision Support System?

Approaches used to build a Data Warehouse

Top-Down Approach

- **Top-Down Approach**, meaning that an organization has developed an enterprise data model, collected enterprise-wide business requirements, and decided to build an enterprise data warehouse with subset data marts.
- In this approach, we need to spend the extra time and build a core data warehouse first, and then use this as the basis to quickly spin off many data marts.
- **The disadvantage** is this approach takes longer to build initially since time has to be spent analyzing data requirements in the full-blown warehouse, identifying the data elements that will be used in numerous marts down the road
- **The advantage** is that once you go to build the data mart, you already have the warehouse to draw from.

Bottom-Up Approach

- **Bottom-Up Approach**, implying that the business priorities resulted in developing individual data marts, which are then integrated into the enterprise data warehouse
- In this approach, we need to build a workgroup specific data mart first
- Disadvantage –This approach gets data into your user's hands quicker but the work it takes to get the information into the data mart may not be reusable when moving the same data into a warehouse or trying to use similar data in the different data mart.
- Advantage is you gain speed but not portability

Important Considerations

- Tighter Integration

Important Considerations

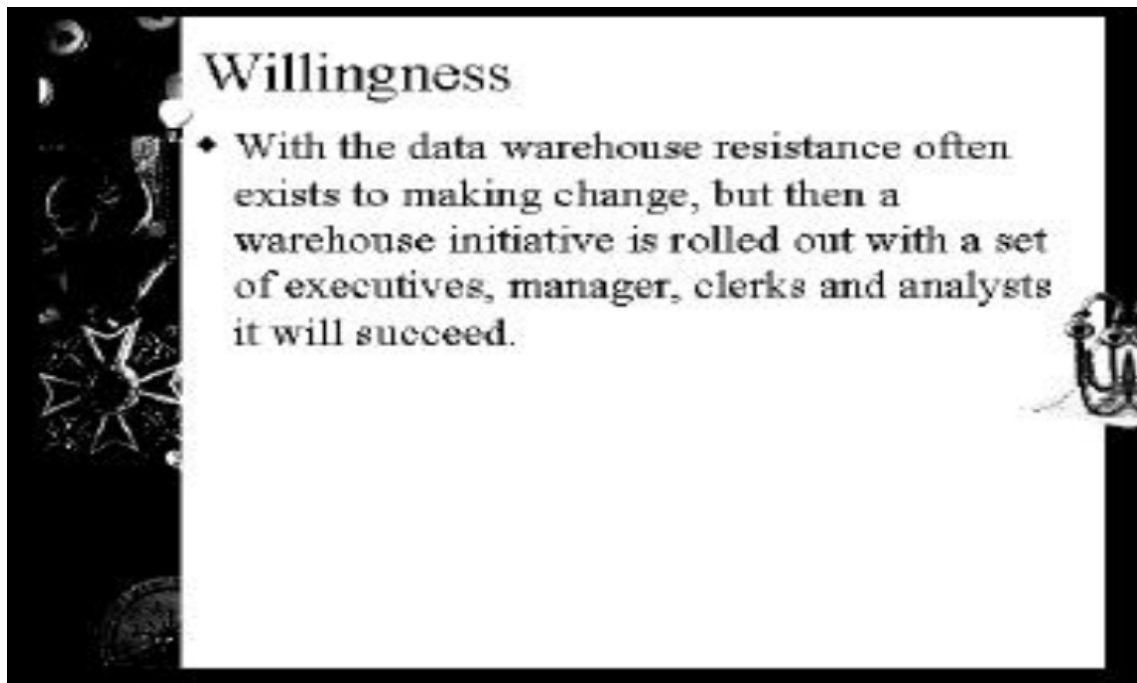
- ♦ Tighter Integration
 - The term back end describes the data repository used to support the data warehouse coupled with the software that supports the repository. For e.g., Oracle 7 Cooperative Server. Front end describes the tools used by the warehouse end-users to support their decision making activities. This tighter integration between front end and back end requires continual communication between the data warehouse project team players

- Empowerment

Empowerment

- ♦ Users to control their own destiny – no running to the programmers asking for reports programmed to satisfy burning needs, how many times in operational environments are new reports identified, scoped and programmed. An inevitable time delay exists between the identification of new requirements and their delivery.

- Willingness



Business Considerations: Return on Investment Design Considerations

Objective

- The objective of this lesson is to learn about the need of a data warehouse
- It also includes topics related to various business considerations and performance considerations

Introduction

- The data warehouse is an environment, not a product.
- It is an architectural construct of information systems that provides users with current and historical decision-support information that is hard to access or present in traditional operational data stores

- In fact, the data warehouse is a cornerstone of the organization's ability to do effective information processing, which, among other things, can enable and shear the discovery and exploration of important business trends and dependencies that otherwise would have gone unnoticed.
- There are several reasons why organizations consider data warehousing a critical need.

Need of a Data Warehouse

From a business perspective, to survive and succeed in today's highly competitive global environment

- Decisions need to be made quickly and correctly, using all available data. Users are business domain experts, not computer professionals.
- The amount of data is doubling every 18 months, which affects response time and the sheer ability to comprehend its content.
- Competition is heating up in the areas of business intelligence. And added Information value.
- In addition, the necessity for data warehouses has increased as organizations distribute control away from the middle management layer, which has traditionally provided and screened business information

Technology reasons for the existence of data warehousing

- First, the data warehouse is designed to address the incompatibility of informational and operational transactional systems
- The IT infrastructure is changing rapidly, and its capabilities are increasing, as evidenced by the following:
 - The price of MIPS (computer processing speed) continues to decline, while the power of microprocessors doubles every 2 years.

- The price of digital storage is rapidly dropping.
- Network bandwidth is increasing, while the price of high bandwidth is decreasing.
- The workplace is increasingly heterogeneous with respect to both hardware and software.
- Legacy systems need to, and can, be integrated with new applications.

Business Considerations: Return on investment

Approach

- The information scope of the data warehouse varies with the business requirements, business priorities, and even the magnitude of the problem
- The subject-oriented nature of the data warehouse means that the nature of the subject determines the scope (or the coverage) of the warehoused information.
- Specifically, if the data warehouse is implemented to satisfy a specific subject area (e.g., human resources), such a warehouse is expressly designed to solve business problems related to personnel
- An organization may choose to build another warehouse for its marketing department.
- These two warehouses could be implemented independently and be completely stand-alone warehouses (DataMart), and interact with each other

Two Approaches for Data Warehousing

- The top-down approach, meaning that an organization has developed an enterprise data model, collected enterprise-wide business requirements

and decided to build an enterprise data warehouse with subset data marts

- The bottom-up approach, implying that the business priorities resulted i.e. developing individual data marts, which are then integrated into the enterprise data warehouse
- The bottom-up approach is probably more realistic, but the complexity of the integration may become a serious obstacle, and the warehouse designers should carefully analyze each data mart for integration affinity.

Organizational Issues

- Most IS an organization has considerable expertise in developing operational systems.
- However; the requirements and environments associated with the informational applications of a data warehouse are different.
- Therefore, an organization will need to employ different development practices than the ones it uses for operational applications.
- The IS department will need to bring together data that cuts across a company's operational systems as well as data from outside the company.
- But users will also need to be involved with a data warehouse implementation since they are closest to the data.
- In many ways, a data warehouse implementation is not truly a technological issue; rather, it should be more concerned with identifying and establishing information requirements, the data sources to fulfill these requirements and timeliness.

Design Considerations

- To be successful, a data warehouse designer must adopt a holistic approach, consider all data warehouse components as parts of a single complex system and take into account all possible data sources and all known usage requirements.

- Failing to do so may easily result in a data warehouse design that is skewed toward a particular business requirement, a particular data source, or a selected access tool.
- In general, a data warehouse's design point is to consolidate data from multiple, often heterogeneous, sources into a query database.

Difficulty in building data warehousing

- Heterogeneity of data sources, which affects data conversion, quality, and timelines.
 - Use of historical data, which implies that data may be "old".
 - Tendency of databases to grow very large.
 - Another important point concerns the experience and accepted practices.
 - Indeed, the data warehouse is business-driven (not IS-driven, as in OLTP), requires continuous interactions with end-users, and is never finished, since both requirements and data sources change.
-
- Understanding these points allows developers to avoid a number of pitfalls relevant to data warehouse development, and justifies a new approach to data warehouse design: a business-driven, continuous, iterative warehouse engineering approach.
 - In addition to these general considerations, there are several specific points relevant to the data warehouse design.
 - **Data content**
 - **Metadata**
 - **Data Distribution**
 - **Tools**

Data Content

- One common misconception about data warehouses is that they should not contain as much detail level data as operational systems used to source this data.
- In reality, however, while the data in the warehouse is formatted differently from the operational data, it may be just as detailed
- Typically, a data warehouse may contain detailed data, but the data is cleaned up and-transformed to fit the ware-house model, and certain transactional attributes of the data are filtered out.
- These attributes are mostly the ones used for the internal transaction system logic, and they are not meaningful in the context of analysis and decision-making.
- The content and structure of the data warehouse are reflected in its data model
- The data model is the template that describes how the information will be organized within the integrated warehouse framework
- It identifies major subjects and relationships of the model, including keys, attributes, and attribute groupings.
- In addition, a designer should always remember that decision support queries, because of their broad scope and analytical intensity, require data models to be optimized to improve query performance.
- In addition to its effect on query performance, the data model affects data storage requirements and data loading performance.
- Additionally, the data model for the data warehouse may be (and quite often is) different from the data models for data marts.
- The data marts, discussed in the previous chapter, are sourced from the data warehouse and may contain highly aggregated and summarized data in the form of a specialized demoralized relational schema (star schema) or as a multidimensional data cube.
- The key point is, however, that in a dependent data mart environment, the data mart data is cleaned up, is transformed, and is consistent with the data warehouse and other data marts sourced from the same warehouse.

Metadata

- As already discussed, metadata defines the contents and location of data (data model) in the warehouse, relationships between the operational databases and the data warehouse, and the business views of the warehouse data that are accessible by end-user tools.
- Metadata is searched by users to find data definitions or subject areas.
- In other words, metadata provides decision-support oriented pointers to warehouse data, and thus provides a logical link between warehouse data and the decision support application.
- A data warehouse design should ensure that there are mechanisms that populate and maintain the metadata repository and that all access paths to the data warehouse have metadata as an entry point.
- To put it another way, the warehouse design should prevent any direct access to the warehouse data (especially updates) if it does not use metadata definitions to gain access.

Data Distribution

- One of the biggest challenges when designing a data warehouse is the data placement and distribution strategy.
- This follows from the fact that as the data volumes continue to grow; the database size may rapidly outgrow a single server.
- Therefore, it becomes necessary to know how the data should be divided across multiple servers, and which users should get access to which types of data.
- The data placement and distribution design should consider several options, including data distribution by subject area (e.g., human resources, marketing), location (e.g., geographic regions), or time (e.g., current, monthly, Quarterly).

- The designers should be aware that, while the distribution solves a number of problems, it may also create a few of its own; for example, if the warehouse servers are distributed across multiple locations, a query that spans several servers across the LAN or WAN may flood the network with a large amount of data
- Therefore, any distribution strategy should take into account all possible access needs for the warehouse data.

Tools

- A number of tools available today are specifically designed to help in the implementation of a data warehouse
- These tools provide facilities for defining the transformation and cleanup rules, data movement (from operational sources into the warehouse), end-user query, reporting, and data analysis
- Each tool takes a slightly different approach to data warehousing and often maintains its own version of the metadata, which is placed in a tool-specific, proprietary meta-data repository
- Data warehouse designers have to be careful not to sacrifice the overall design to fit a specific tool.
- At the same time, the designers have to make sure that all selected tools are compatible with the given data warehouse environment and with each other.
- That means that all selected tools can use a common metadata repository
- Alternatively, the tools should be able to source the metadata from the warehouse data dictionary (if it exists) or from a CASE tool used to design the warehouse database
- Another option is to use metadata gateways that translate one tool's metadata into another tool's format.

- If these requirements are not satisfied, the resulting warehouse environment may rapidly become unmanageable, since every modification to the warehouse data model may involve some significant and labor-intensive changes to the meta-data' definitions for every tool in the environment
- And then, these changes would have to be verified for consistency and integrity

Performance Considerations

- Although the data warehouse design point does not include sub-second response times typical of OLTP systems, it is nevertheless a clear business requirement that an ideal data warehouse environment should support interactive query processing
- In fact, the majority of end-user tools are designed as interactive applications
- Therefore, "rapid" query processing is a highly desired feature that should be designed into the data warehouse.
- Of course, the actual performance levels are business dependent and vary widely from one environment to another.
- Unfortunately, it is relatively difficult to predict the performance of a typical data warehouse.
- One of the reasons for this is the unpredictable usage pattern against the data.
- Thus, traditional database design and tuning techniques don't always work in the data warehouse arena.
- When designing a data warehouse, therefore, the need to clearly understand users informational requirements becomes mandatory
- Specifically, knowing how end users need to access various data can help design warehouse data-bases to avoid the majority of the most expensive operations such as multi-table scans and joins

- For example, one design technique is to populate the ware-house with a number of demoralized views containing summarized, derived, and aggregated data
- If done correctly, many end user queries may execute directly against these views, thus maintaining appropriate overall performance levels.

Technical Considerations, Implementation Considerations

Objective

The purpose of this lessons is to take a close look at what it takes to build a successful data warehouse

Reasons for the Existence of Data Warehousing

- First, the data warehouse is designed to address the incompatibility of informational and operational transaction system
- These two classes of information systems are designed to satisfy different, often incompatible, requirements
- At the same time, the IT infrastructure is changing rapidly, and its capabilities are increasing, evidenced by the following
 - The price of MIPS (computer processing speed) continues to decline, while
 - The power of microprocessors doubles every 2 years.
 - The price of digital storage is rapidly dropping.
- Network bandwidth is increasing, while the price of high bandwidth is decreasing.
- The workplace is increasingly heterogeneous with respect to both the hardware and software.
- Legacy systems need to, and can, be integrated with new applications.

These business and technology drivers often make building a data warehouse a strategic imperative.

Technical Considerations

- The hardware platform that would house the data warehouse
- The database management system that supports the warehouse database
- The communications infrastructure that connects the warehouse, data marts, operational systems, and end users
- The hardware platform and software to support the metadata repository
- The systems management framework that enables centralized management and administration of the entire environment.

Hardware Platforms

- An important consideration when choosing a data warehouse server is its capacity for handling the volumes of data required by decision support applications, some of which may require a significant amount of historical (e.g., up to 10 years) data.
- This capacity requirement can be quite large. For example, in general, disk storage allocated for the warehouse should be 2 to 3 times the size of the data component of the warehouse to accommodate DSS processing, such as sorting, storing of intermediate results, summarization, join, and formatting.
- Often, the platform choice is the choice between a mainframe and non-MVS (UNIX or Windows NT) server.
- Of course, a number of arguments can be made for and against each of these choices
- For example, a mainframe is based on a proven technology; has large data and throughput capacity; is reliable, available, and serviceable; and may support the legacy databases that are used as sources for the data warehouse

- The data warehouse residing on the mainframe is best suited for situations in which large amounts of legacy data need to be stored in the data warehouse.
- A mainframe system, however, is not as open and flexible as a contemporary client/server system and is not optimized for ad hoc query processing
- A modern server (no mainframe) can also support large data volumes and a large number of flexible GUI-based end-user tools and can relieve the mainframe from ad hoc query processing
- However, in general, non-MVS servers are not as reliable as mainframes, are more difficult to manage and integrate into the existing environment, and may require new skills and even new organizational structures
- From the architectural viewpoint, however, the data warehouse server has to be specialized for the tasks associated with the data warehouse, an mainframe can be well suited to be a data warehouse server.
- Let's look at the hardware features that make a server-whether it is the mainframe, UNIX, or NT-based an appropriate technical solution for the data warehouse.
- To begin with, the data warehouse server has to be able to support large data volumes and complex query processing.
- In addition, it has to be scalable, since the data warehouse is never finished, as new user requirements, new data sour and more historical data are continuously incorporated into the warehouse, a clear as the user population of the data warehouse continues to grow.
- Therefore, a clear requirement for the data warehouse server is the scalable high-performance data loading and ad hoc query processing as well as the ability to support databases in a reliable, efficient fashion.
- An important design point when selecting a scalable computing platform is the right balance between all computing component example, between the number of processors in a multiprocessor system an the I/O bandwidth
- Remember that the lack of balance in a system inevitably results in a bottleneck!

- Typically, when a hardware platform is sized to accommodate the data house, this sizing is frequently focused on the number and size of disks
- A typical disk configuration. Includes 2.5 to 3 times the amount of raw important consideration disk throughput comes from the actual number of disks, and not the total disk space.
- Thus, the number of disks has direct on data parallelism.
- To balance the system, it is very important to correct number of processors to efficiently handle all disk I/O operations.
- If this allocation is not balanced, an expensive data warehouse platform can rapidly become CPU-bound.
- Indeed, since various processors have widely performance ratings and thus can support a different number of CPU, data warehouse designers should carefully analyze the disk I/O processor capabilities to derive an efficient system configuration
- For if it takes a CPU rated at 10 SPECint to efficiently handle one 3-Glry- _ drive, then a single 30 SPECint processor in a multiprocessor system can handle three disk drives
- Knowing how much data needs to be processed, should give you an idea of how big the multiprocessor 'system should be
- A consideration is related to disk controllers. A disk controller can support a amount of data throughput (e.g., 20 Mbytes/s).
- Knowing the per-disk through-put ratio and the total number of disks can tell you how many controller given type should be configured in the system.
- The idea of a balanced approach can (and should) be carefully extended to all system components
- The resulting system configuration will easily handle known workloads and provide a balanced and scalable computing platform for future growth

Optimal hardware architecture for parallel query scalability

- An important consideration when selecting a hardware platform for a data warehouse is the ability of scalability
- Therefore, a frequent approach to system selection is to take of hardware parallelism that comes in the form of shared-memory symmetric multiprocessors (SMPs), clusters, and shared-nothing distributed-memory system terns (MPPs)
- The scalability of these systems can be seriously affected by the system-architecture-induced data skew.
- This architecture induced data skew is more severe in the low-density asymmetric connection architectures (e.g., daisy-chained, 2-D and 3-D mesh), and is virtually nonexistent in symmetric connection architectures (e.g., cross-bar switch).

“Thus, when selecting a hardware platform for a data warehouse, take into account the fact that the system architecture induced data skew can overpower even the best data layout for parallel query execution, and can force an expensive parallel computing system to process queries serially.”

Data Warehouse and DBMS Specialization

- To reiterate, the two important challenges facing the developers of data ware-houses are the very large size of the databases and the need to process complex ad hoc queries in a relatively short time
- Therefore, among the most important requirements for the data warehouse DBMS are performance, throughput, and scalability.
- The majority of established RDBMS vendors have implemented various degrees of parallelism in their respective products

- Although any relational database management system—such as DB2, Oracle, Informix, or Sybase—supports parallel database processing, some of these products have been architect to better suit the specialized requirements of the data warehouse
- In addition to the “traditional” relational DBMSs, there are databases that have been optimized specifically for data warehousing, such as Red Brick Warehouse from Red Brick Systems

Communications Infrastructure

- When planning for a data warehouse, one often-neglected aspect of the architecture is the cost and efforts associated with bringing access to corporate data directly to the desktop
- These costs and efforts could be significant, since many large organizations do not have a large user population with direct electronic access to information, and since a typical data warehouse user requires a relatively large bandwidth to interact with the data warehouse and retrieve a significant amount of data for the analysis.
- This may mean that communications networks have to be expanded, and new hardware and software may have to be purchased

Implementation Considerations

- A data warehouse cannot be simply bought and installed—its implementation requires the integration of many products within a data warehouse
- The caveat here is that the necessary customization drives up the cost of implementing a data warehouse.
- To illustrate the complexity of the data warehouse implementation, let’s discuss the logical steps needed to build a data warehouse:
- Collect and analyze business requirements.
- Create a data model and a physical design for the data warehouse.
- Define data sources.

- Choose the database technology and platform for the warehouse
- Extract the data from the operational databases, transform it, and clean it up.
- And load it into the database.
- Choose database access and reporting tools.
- Choose database connectivity software.
- Choose data analysis and presentation software.
- Update the data warehouse.

“When building the warehouse, these steps must be performed within the constraints of the current state of data warehouse technologies”

Access Tools

Currently, no single tool on the market can handle all possible data warehouse Access needs. Therefore, most implementations rely on a suite of tools.

- The best way to choose this suite includes the definition of different types of access.
 - Simple tabular form reporting
 - Ranking.
 - Multi variable analysis
 - Time series analysis
 - Data visualization, graphing, charting, and pivoting Complex textual search
 - Statistical analysis
 - Artificial intelligence techniques for testing of hypothesis, trends discovery definition, and validation of data clusters and segments
 - Information mapping (i.e., mapping of spatial data in geographic' information systems)
 - Ad hoc user-specified queries

- Predefined repeatable queries
- Interactive drill-down reporting and analysis
- Complex queries with multi-table joins, multilevel subqueries, and sophisticated search criteria

In addition, certain business requirements often exceed existing tool capabilities and may require building sophisticated applications to retrieve and analyze warehouse data.

- These applications often take the form of custom-developed screens and reports that retrieve frequently used data and format it in a pre-defined standardized way.
- This approach may be very useful for those data warehouse users who are not yet comfortable with ad hoc queries
- There are a number of query tools on the market today. Many of these tools are designed to easily compose and execute ad hoc queries and build customized reports with little knowledge of the underlying database technology, SQL, or even the data model (i.e., Impromptu from Cognos, Business Objects, etc.), while others (e.g., Andyne's GQL) provide relatively low-level capabilities for an expert user to develop complex ad hoc queries in a fashion similar to developing SQL queries for relational databases.
- Business requirements that exceed the capabilities of ad hoc query and reporting tools are fulfilled by different classes of tools: OLAP and data mining tools.

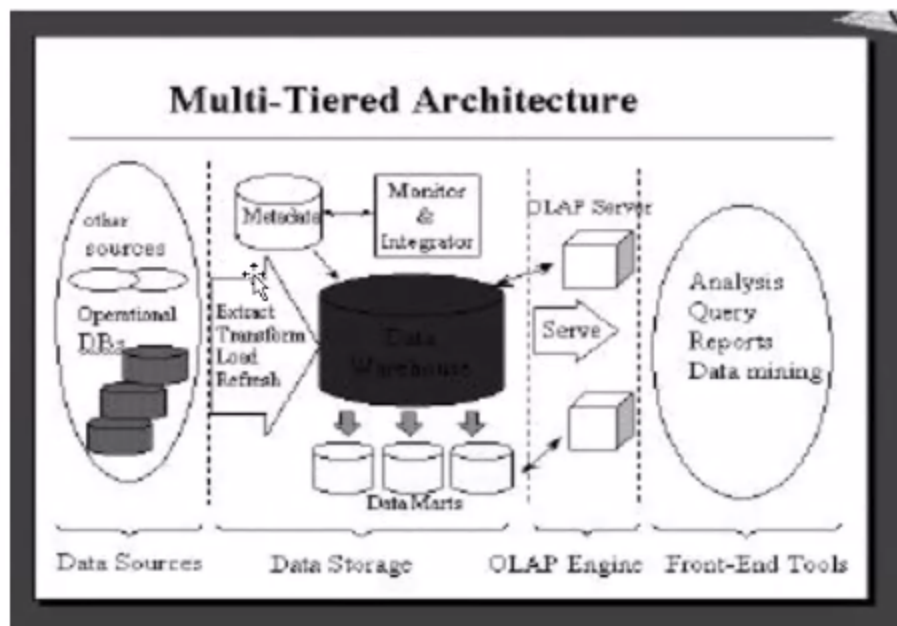
Basic Concepts of Data Warehousing

- Introduction, Meaning and characteristics of Data Warehousing, Online Transaction Processing (OLTP), Data Warehousing Models, Data Warehouse Architecture, and Principles of Data Warehousing Data Mining

Building a Data Warehouse

- Project Structure of the Data Warehouse, Data Warehousing, and Operational Systems, organizing for building data warehousing, Important considerations - Tighter integration, Empowerment, Willingness Business Considerations: Return on Investment, Design Considerations, Technical Consideration, Implementation Consideration, Benefits of Data Warehousing

Multi-Tiered Architecture



- **Extract:** take only the information needed
 - e.g. if there's name and id, take only the id
- **Transform:** data cleaning, eliminate consistency, convert in the common format
 - e.g. some attendance using 1 and O or A and P. So, we can convert to attend and absent.
- **Load:** load the data into the data storage
- **Refresh:** check if the data is updated, define the period of the data update, define which part of the data to be updated

