I.T.S
GHAZIABAD

Data Warehousing & Mining

UNIT – I

I.T.S GHAZIABAD

Syllabus of Unit - I

- DSS-Uses, definition, Operational Database.
- Introduction to DATA Warehousing. Data-Mart,
- Concept of Data-Warehousing,
- Multi Dimensional Database Structures.
- Client/Server Computing Model & Data Warehousing
- Parallel Processors & Cluster Systems. Distributed DBMS implementations.

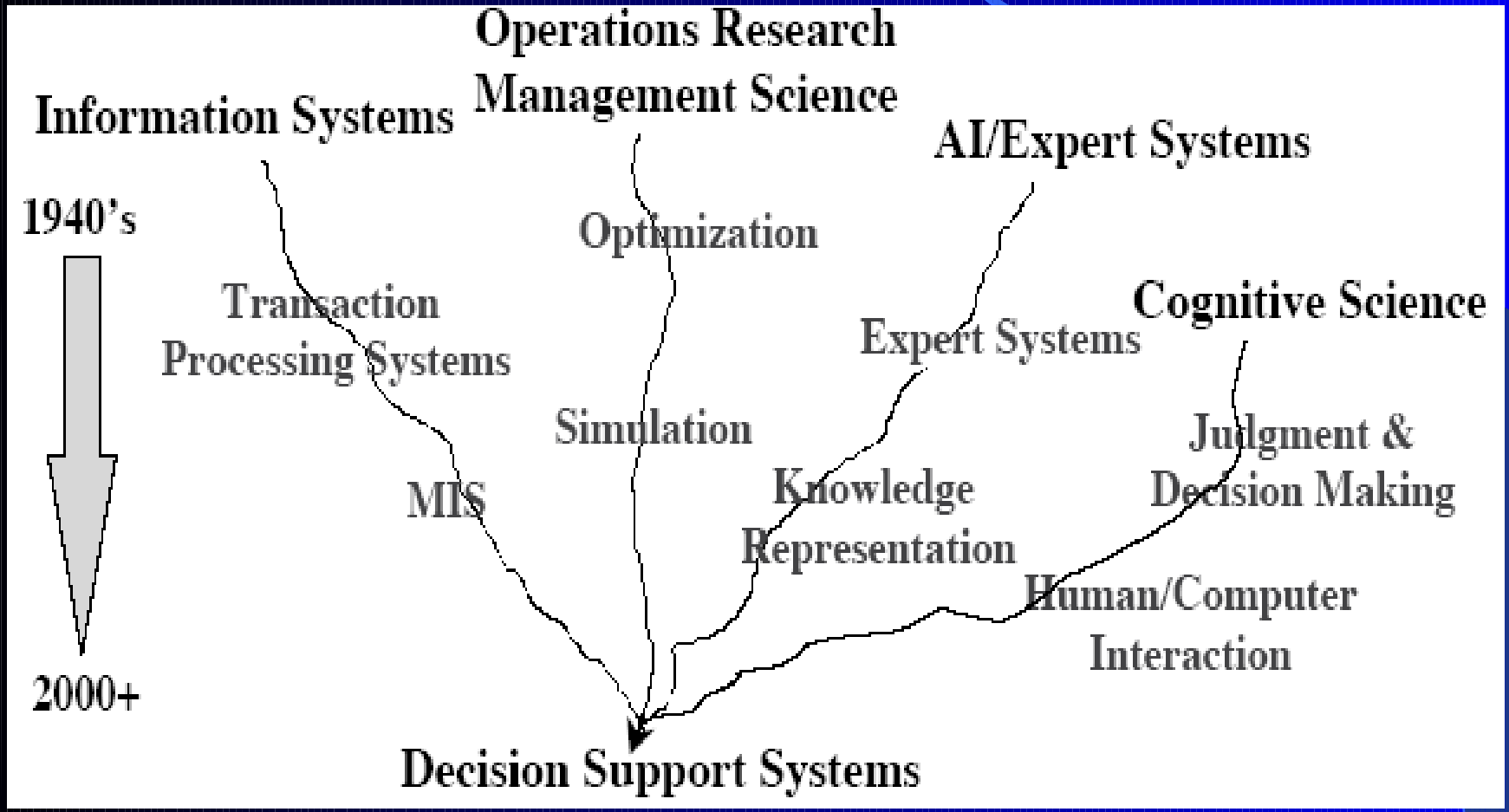
Introduction – Decision Support System (DSS)

- A **Decision Support System (DSS)** is an interactive computer-based system or subsystem intended to help decision makers use communications technologies, data, documents, knowledge and/or models to identify and solve problems, complete decision process tasks, and make decisions.
- It is clear that DSS belong to an environment with multidisciplinary foundations, including (but not exclusively):
  - Database research,
  - Artificial intelligence,
  - Human-computer interaction,
  - Simulation methods,
  - Software engineering, and
  - Telecommunications.

DSS

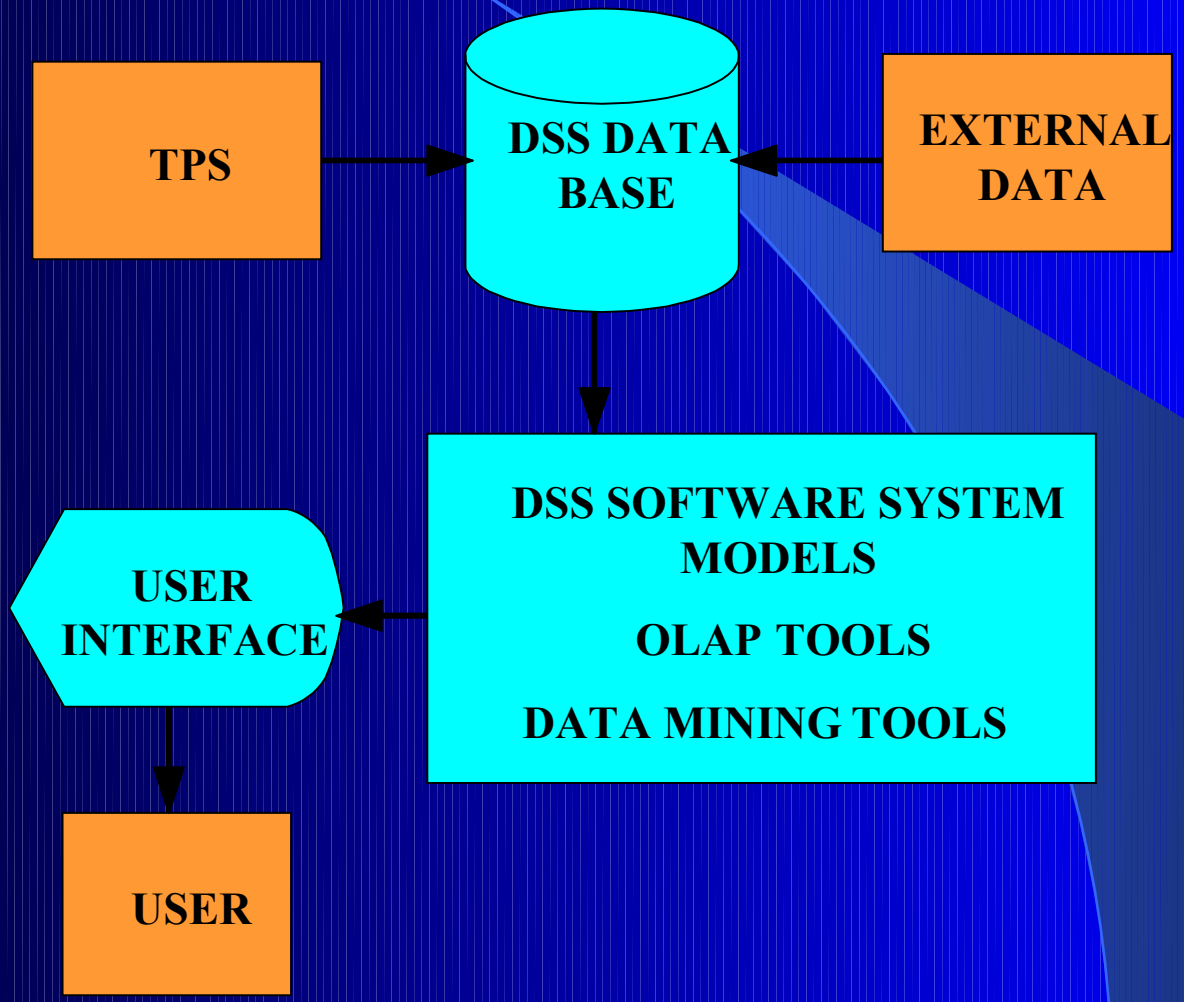
- A **Decision Support System (DSS)** is a computer-based information system that supports business or organizational decision-making activities.
- DSSs serve the management, operations, and planning levels of an organization (usually mid and higher management) and help to make decisions, which may be rapidly changing and not easily specified in advance (Unstructured and Semi-Structured decision problems).
- Decision support systems can be either fully computerized, human or a combination of both.

Historical Evolution of DSS



Typical DSS Architecture

- **TPS**: transaction processing system
- MODEL: representation of a problem
- **OLAP:** on-line analytical processing
- **USER INTERFACE:** how user enters problem & receives answers
- **DSS DATABASE:** current data from applications or groups
- **DATA MINING:** technology for finding relationships in large data bases for prediction

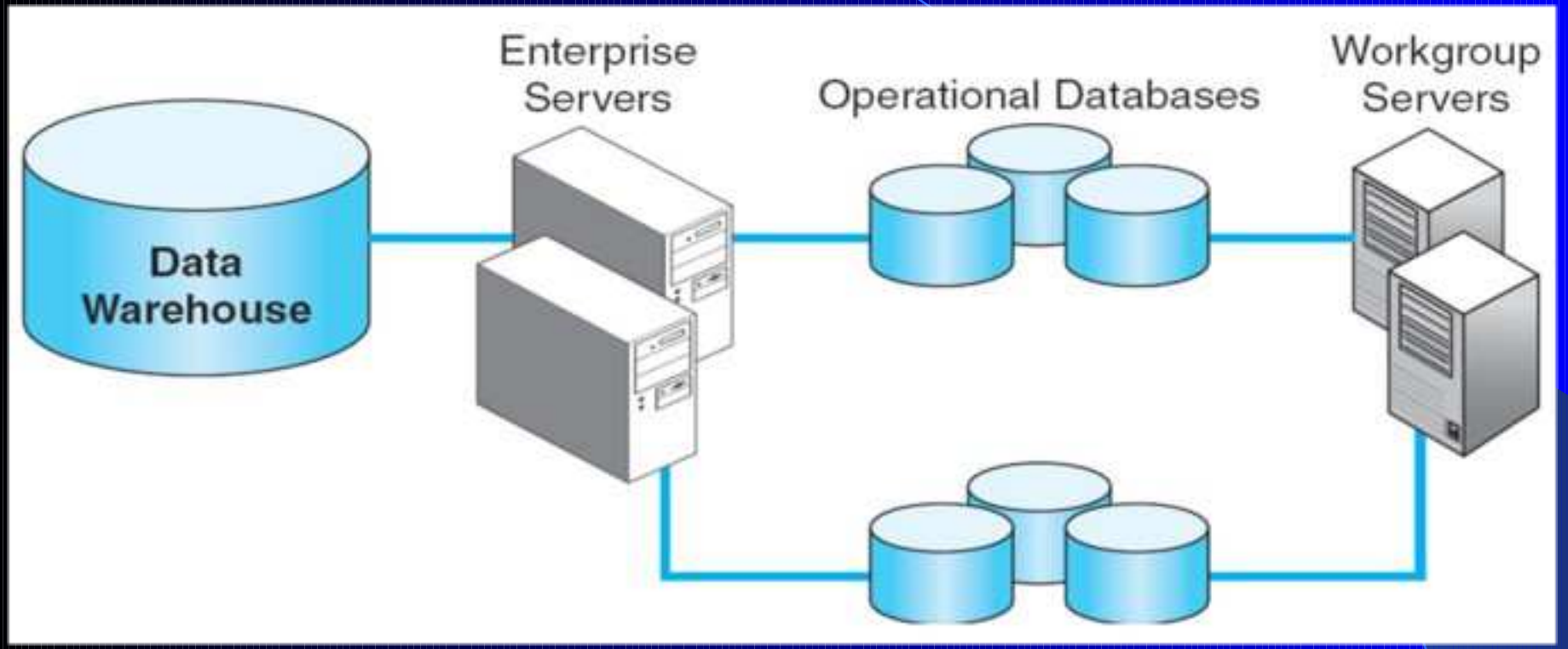


Why DSS?

- Increasing complexity of decisions
  - Technology
  - Information:
    - "Data, data everywhere, and not the time to think!"
  - Number and complexity of options
  - Pace of change
- Increasing availability of computerized support
  - Inexpensive high-powered computing
  - Better software
  - More efficient software development process
- Increasing usability of computers

Operational Databases

- Operational database management systems (also referred to as OLTP databases), are used to manage dynamic data in real-time.
- These types of databases allow you to do more than simply view archived data. Operational databases allows to modify that data (add, change or delete data), doing it in real-time.
- Since the early 90's, the operational database software market has been largely taken over by SQL engines.
- Today, the operational DBMS market (formerly OLTP) is evolving dramatically, with new, innovative entrants and incumbents supporting the growing use of unstructured data and NoSQL DBMS engines, as well as XML databases and NewSQL databases.
- Operational databases are increasingly supporting distributed database architecture that provides high availability and fault tolerance through replication and scale out ability.


Enterprise
Servers
Operational Databases
Workgroup
Servers
Data
Warehouse

Differences between the Databases and Data Warehouses

FEATURES	DATABASE	DATA WAREHOUSE
Characteristic	It is based on Operational Processing.	It is based on Informational Processing.
Data	It mainly stores the Current data which always guaranteed to be up-to-date.	It usually stores the Historical data whose accuracy is maintained over time.
Function	It is used for day-to-day operations.	It is used for long-term informational requirements and decision support.
User	The common users are clerk, DBA, database professional.	The common users are knowledge worker (e.g., manager, executive, analyst)
Unit of work	Its work consists of short and simple transaction.	The operations on it consists of complex queries..
Focus	The focus is on “Data IN”	The focus is on “Information OUT”
Orientation	The orientation is on Transaction.	The orientation is on Analysis.
DB design	The designing of database is ER based and application-oriented.	The designing is done using star/snowflake schema and its subject-oriented.
Summarization	The data is primitive and highly detailed.	The data is summarized and in consolidated form.
View	The view of the data is flat relational.	The view of the data is multidimensional.

I.T.S
GHAZIABAD

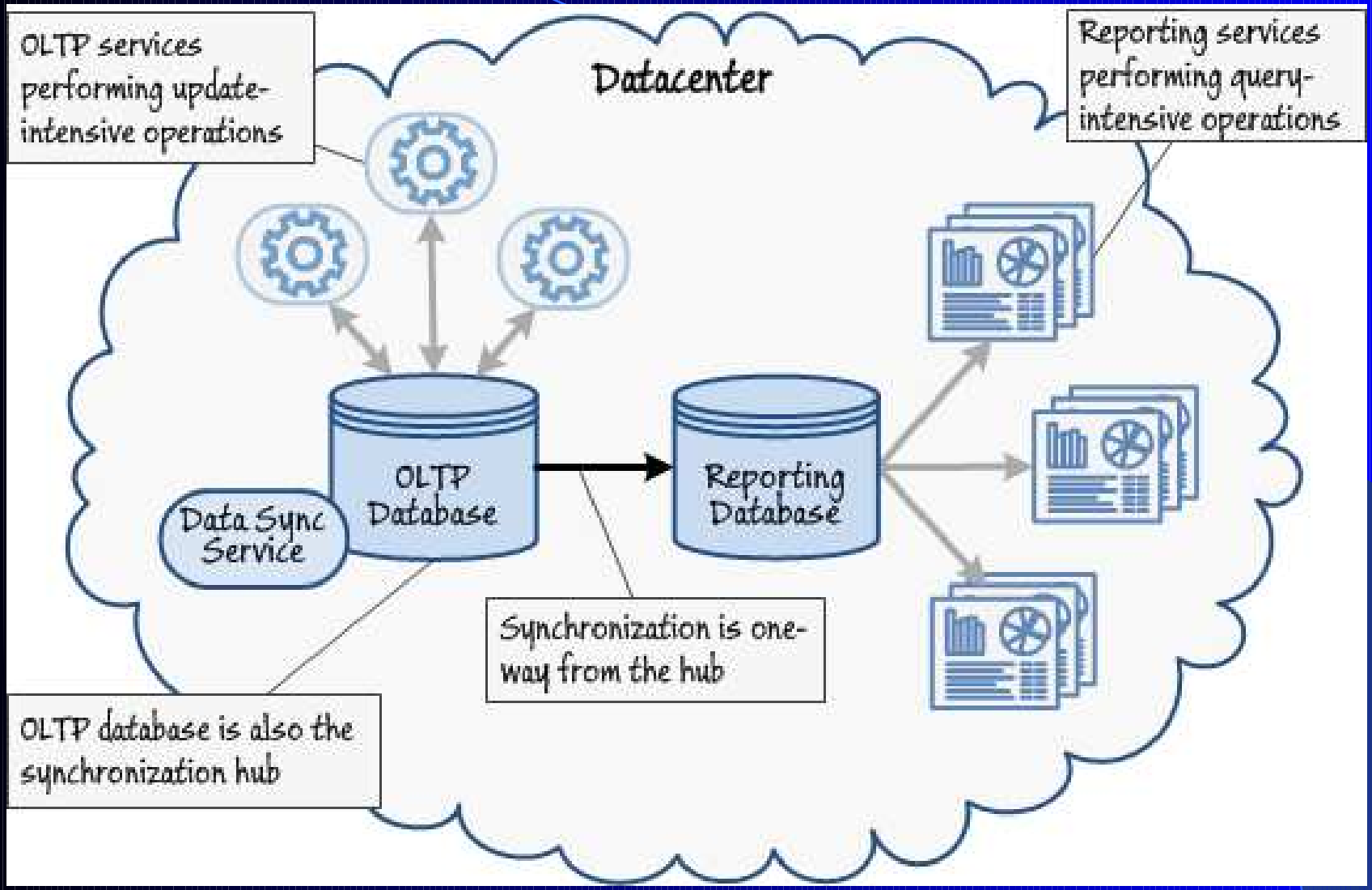
FEATURES	DATABASE	DATA WAREHOUSE
Function	It is used for day-to-day operations.	It is used for long-term informational requirements and decision support.
User	The common users are clerk, DBA, database professional.	The common users are knowledge worker (e.g., manager, executive, analyst)
Access	The most frequent type of access type is read/write.	It mostly use the read access for the stored data.
Operations	The main operation is index/hash on primary key.	For any operation it needs a lot of scans.
Number of records accessed	A few tens of records.	A bunch of millions of records.
Number of users	In order of thousands.	In the order of hundreds only.
DB size	100 MB to GB.	100 GB to TB.
Priority	High performance, high availability	High flexibility, end-user autonomy
Metric	To measure the efficiency, transaction throughput is measured.	To measure the efficiency, query throughput and response time is measured.

I.T.S
GHAZIABAD

DATA Warehousing - Introduction

A data warehouse is a subject-oriented, integrated, nonvolatile, time-variant collection of data in support of management's decisions.

- WH Inmon


Datacenter
OLTP services performing update-intensive operations
Reporting services performing query-intensive operations
OLTP Database
Reporting Database
Data Sync Service
Synchronization is one-way from the hub
OLTP database is also the synchronization hub

Data Warehouse Usage

- Three kinds of data warehouse applications
  - **Information processing**
    - supports querying, basic statistical analysis, and reporting using crosstabs, tables, charts and graphs
  - **Analytical processing**
    - multidimensional analysis of data warehouse data
    - supports basic OLAP operations, slice-dice, drilling, pivoting
  - **Data mining**
    - knowledge discovery from hidden patterns
    - supports associations, constructing analytical models, performing classification and prediction, and presenting the mining results using visualization tools.
- Differences among the three tasks

Data Warehouse: Subject-Oriented

- Organized around major subjects, such as customer, product, sales.
- **Focusing on the modeling and analysis of data for decision makers, not on daily operations or transaction processing.**
- Provide a **simple and concise view around particular subject issues by excluding data that are not useful in the decision support process.**

Data Warehouse—Integrated

- Constructed by integrating multiple, heterogeneous data sources
  - relational databases, flat files, on-line transaction records
- Data cleaning and data integration techniques are applied.
  - Ensure consistency in naming conventions, encoding structures, attribute measures, etc. among different data sources
    - E.g., Hotel price: currency, tax, breakfast covered, etc.
  - When data is moved to the warehouse, it is converted.

Data Warehouse—Time Variant

- The time horizon for the data warehouse is significantly longer than that of operational systems.
  - Operational database: current value data.
  - Data warehouse data: provide information from a historical perspective (e.g., past 5-10 years)
- Every key structure in the data warehouse
  - Contains an element of time, explicitly or implicitly
  - But the key of operational data may or may not contain "time element".

Data Warehouse—Non-Volatile

- A **physically separate** store of data transformed from the operational environment.
- Operational update of data does not occur in the data warehouse environment.
  - Does not require transaction processing, recovery, and concurrency control mechanisms
  - Requires only two operations in data accessing:
    - *initial loading of data* and *access of data*.

Data Warehouse vs. Heterogeneous DBMS

- Traditional heterogeneous DB integration:
  - Build wrappers/mediators on top of heterogeneous databases
  - Query driven approach
    - When a query is posed to a client site, a meta-dictionary is used to translate the query into queries appropriate for individual heterogeneous sites involved, and the results are integrated into a global answer set
    - Complex information filtering, compete for resources
- Data warehouse: update-driven, high performance
  - Information from heterogeneous sources is integrated in advance and stored in warehouses for direct query and analysis

Data Warehouse vs. Operational DBMS

- OLTP (on-line transaction processing)
  - Major task of traditional relational DBMS
  - Day-to-day operations: purchasing, inventory, banking, manufacturing, payroll, registration, accounting, etc.
- OLAP (on-line analytical processing)
  - Major task of data warehouse system
  - Data analysis and decision making
- Distinct features (OLTP vs. OLAP):
  - User and system orientation: customer vs. market
  - Data contents: current, detailed vs. historical, consolidated
  - Database design: ER + application vs. star + subject
  - View: current, local vs. evolutionary, integrated
  - Access patterns: update vs. read-only but complex queries

I.T.S GHAZIABAD

OLTP vs. OLAP

	OLTP	OLAP
users	clerk, IT professional	knowledge worker
function	day to day operations	decision support
DB design	application-oriented	subject-oriented
data	current, up-to-date detailed, flat relational isolated	historical, summarized, multidimensional integrated, consolidated
usage	repetitive	ad-hoc
access	read/write index/hash on prim. key	lots of scans
unit of work	short, simple transaction	complex query
# records accessed	tens	millions
#users	thousands	hundreds
DB size	100MB-GB	100GB-TB
metric	transaction throughput	query throughput, response

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

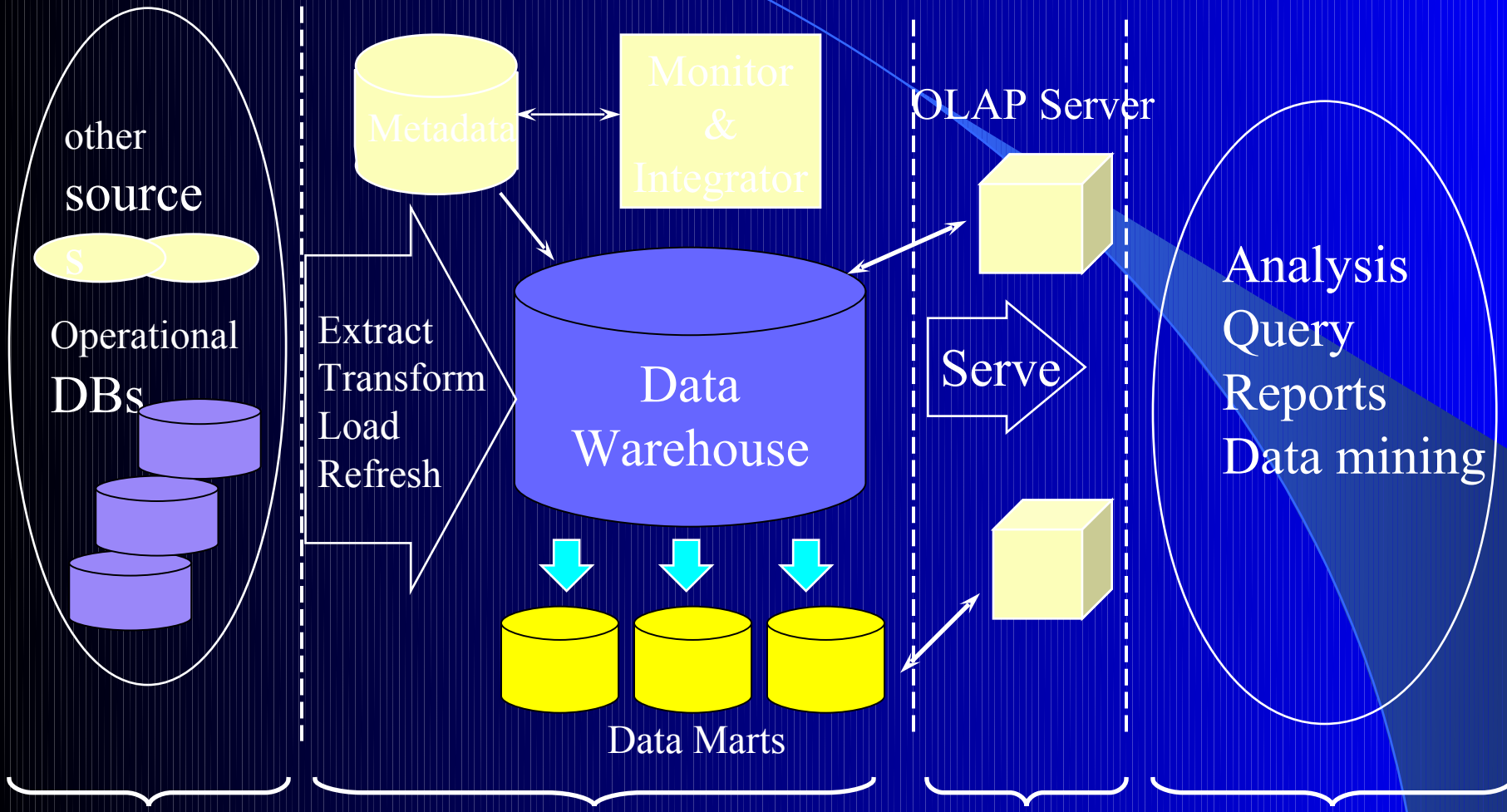
I.T.S
GHAZIABAD

Data Mart

I.T.S
GHAZIABAD

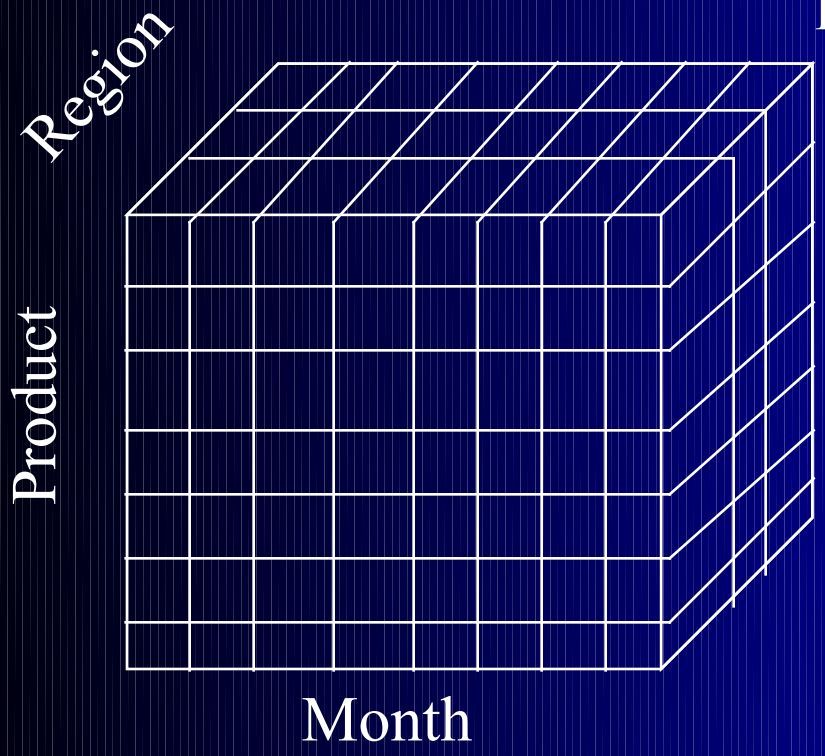
Concept of Data-Warehousing

Multi-Tiered Architecture

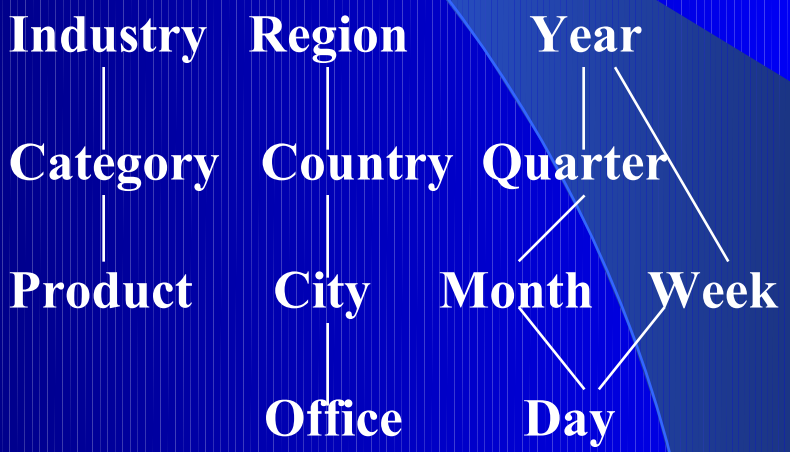


Multi Dimensional Database Structures

- Sales volume as a function of product, month, and region



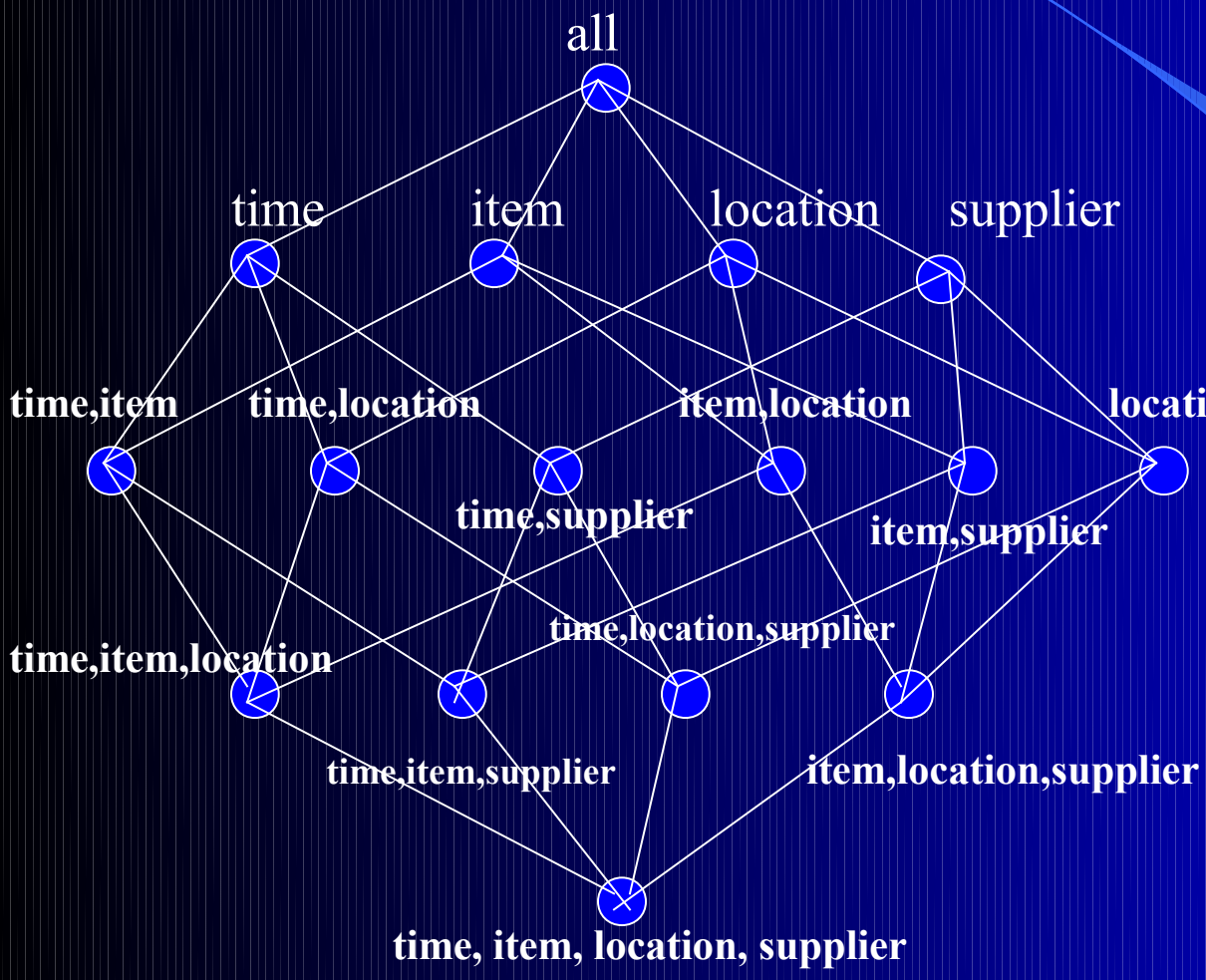
Dimensions: Product, Location, Time
Hierarchical summarization paths



From Tables and Spreadsheets to Data Cubes

- A data warehouse is based on a **multidimensional data model** which views data in the form of a data cube
- A data cube, such as sales, allows data to be modeled and viewed in multiple dimensions
  - Dimension tables, such as item (item_name, brand, type), or time(day, week, month, quarter, year)
  - Fact table contains measures (such as dollars_sold) and keys to each of the related dimension tables
- In data warehousing literature, an n-D base cube is called a **base cuboid**. The top most 0-D cuboid, which holds the highest-level of summarization, is called the **apex cuboid**. The lattice of cuboids forms a **data cube**.

Cube: A Lattice of Cuboids



0-D(apex) cuboid

1-D cuboids

2-D cuboids

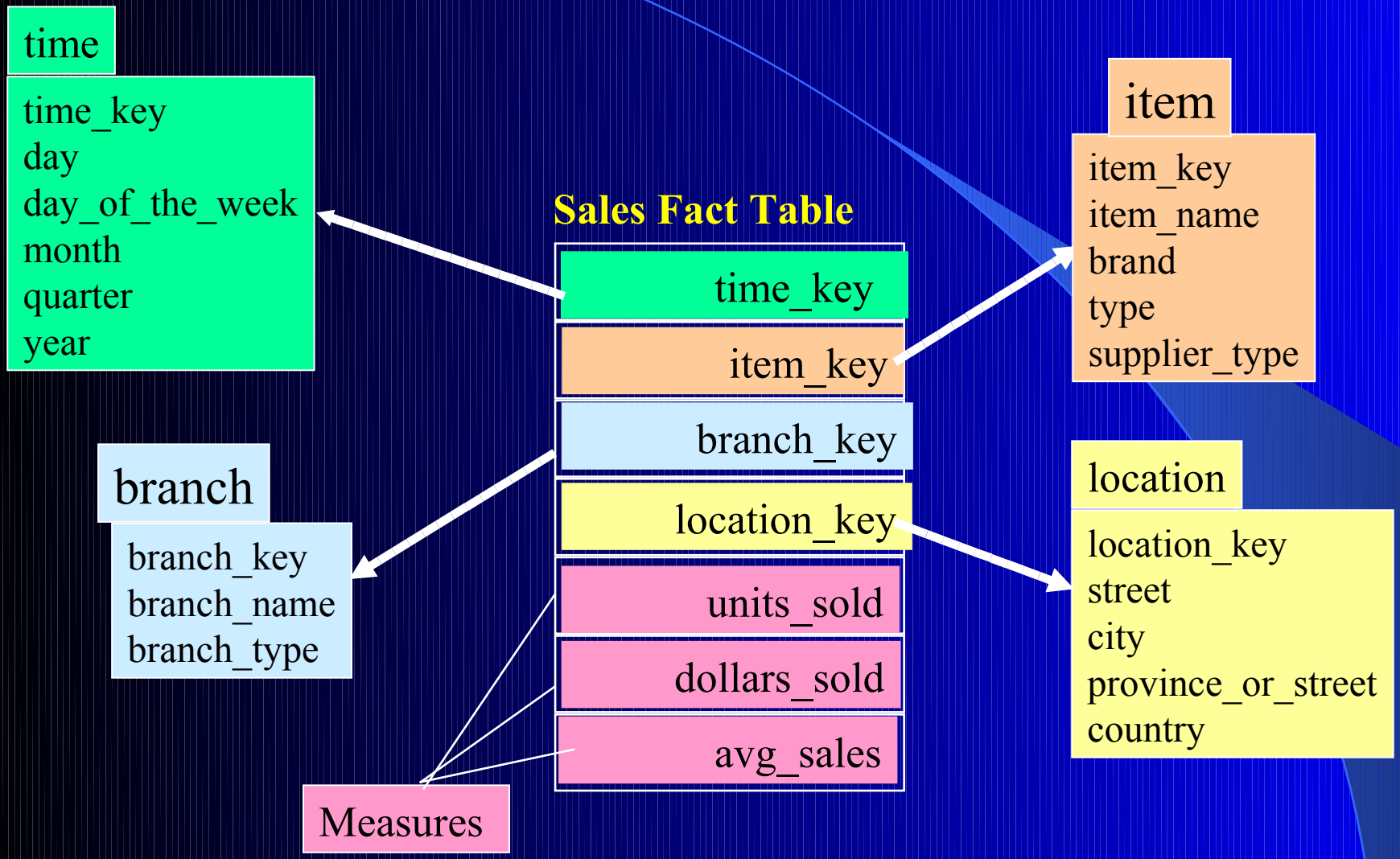
3-D cuboids

4-D(base) cuboid

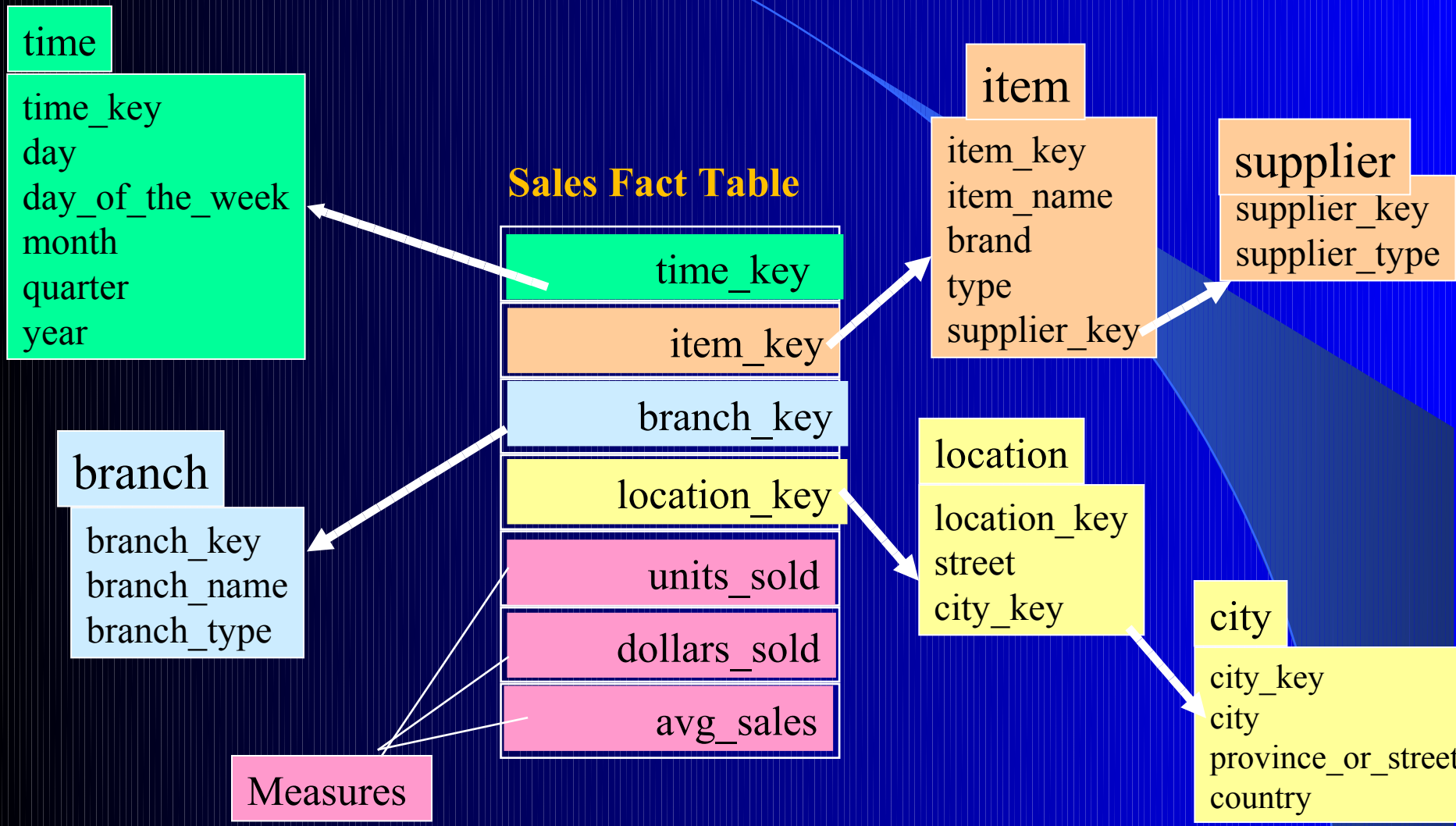
Conceptual Modeling of Data Warehouses

- Modeling data warehouses: dimensions & measures
  - **Star schema:** A fact table in the middle connected to a set of dimension tables
  - **Snowflake schema:** A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake
  - **Fact constellations: Multiple fact tables share dimension tables**, viewed as a collection of stars, therefore called galaxy schema or fact constellation

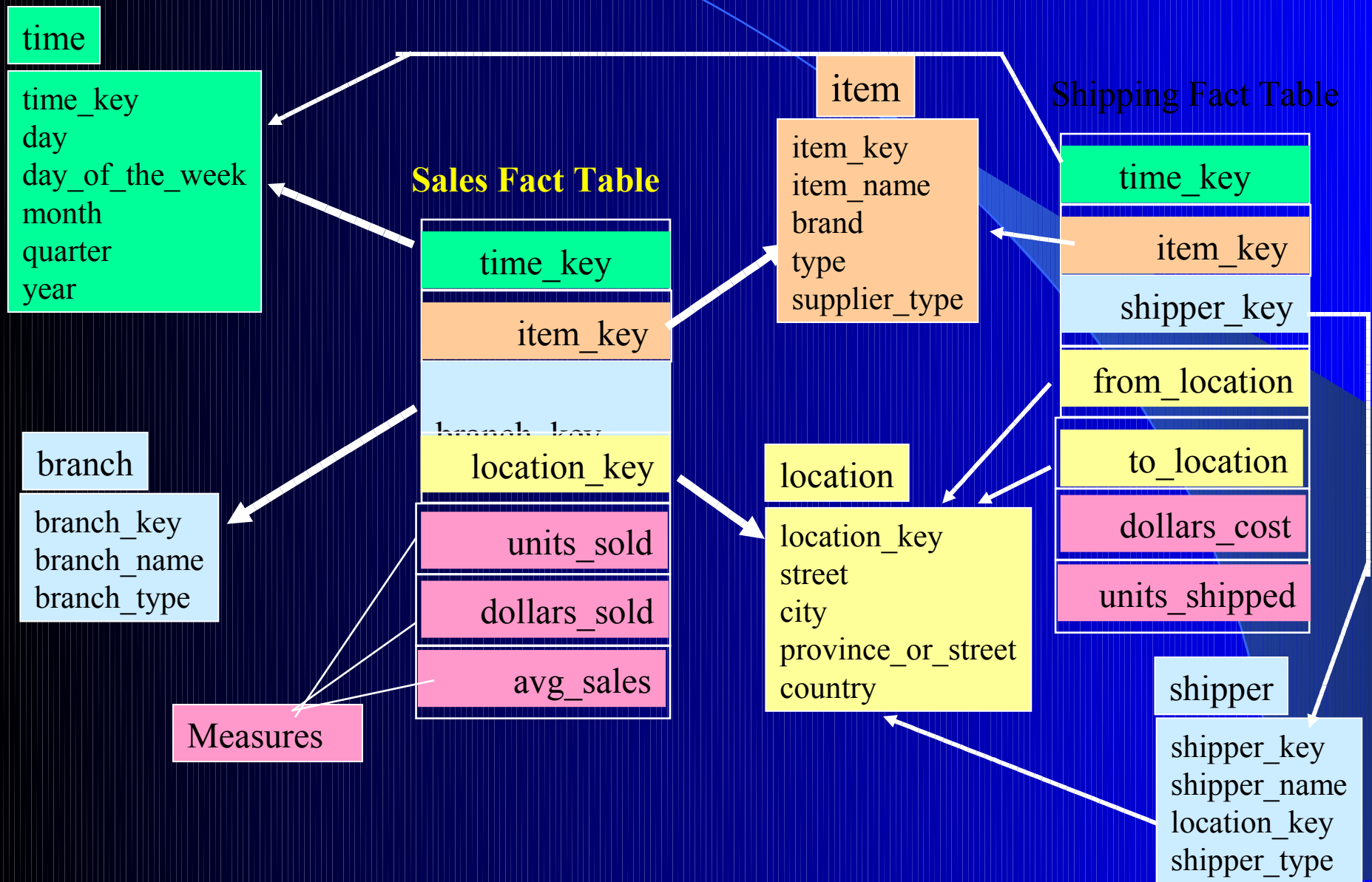
Example of Star Schema



Example of Snowflake Schema



Example of Fact Constellation



Client/Server Computing Model & Data Warehousing

- The fundamental characteristic of client/server computing is distribution of computing resources (e.g. data, compute power) across different computers.
- The idea is to divide applications into logical segments (tasks) so that they are then performed on platforms most appropriate.
- A **client/server database system** increases processing power by separating the database management system from the application; the client as the front-end system handling the user interface and the server as the back-end system accessing the database, which cooperate to run an application.

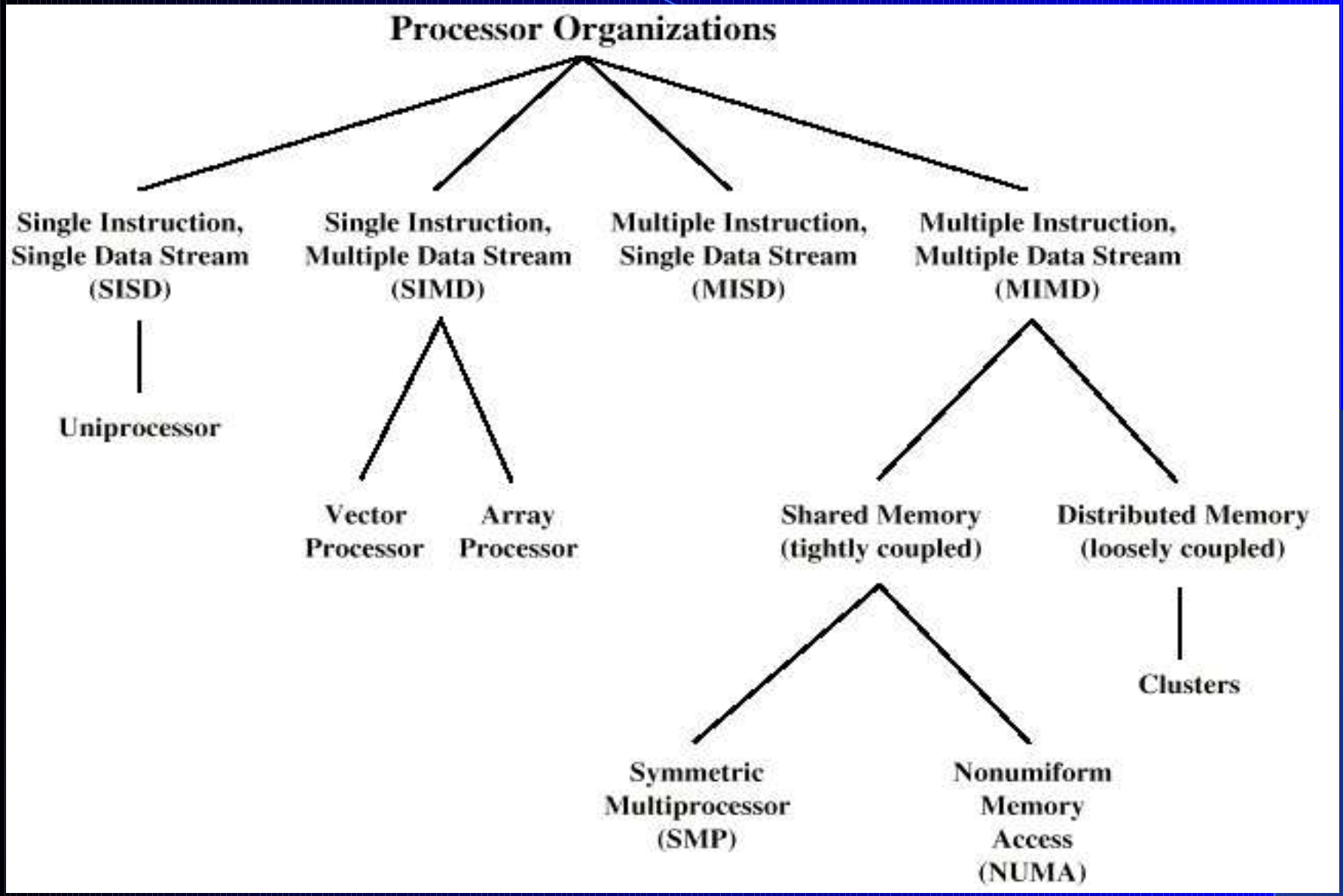
Contd....

- Data Warehousing is a continual process which enables a corporation to assemble operational and other data from a variety of internal and external sources, and transform that data into consistent, high-quality, business information, distribute that information to the points of maximum value within the organizations, and provide easy, flexible and fast access for busy non-technical users.

Reasons for using client/server

- Exploitation of centralised computing power /data capacity
- Scalability
- Performance
- Flexibility (in order to adjust to changing demands)
- GUI on desktop
- Protection of investment, strategic software, strategic data
- Client/server provides an integrated solution.

Parallel Processors & Cluster Systems



Loosely Coupled - Clusters

- Collection of independent whole uni-processors or SMPs
  - Usually called nodes
- Interconnected to form a cluster
- Working together as unified resource
  - Illusion of being one machine
- Communication via fixed path or network connections

Cluster Benefits

- Absolute scalability
- Incremental scalability
- High availability
- Superior price/performance

Distributed DBMS implementations

I.T.S
GHAZIABAD

Data Warehousing & Mining

UNIT – II

I.T.S GHAZIABAD

Syllabus of Unit - II

- DATA Warehousing
- Data Warehousing Components
- Building a Data Warehouse
- Warehouse Database
- Mapping the Data Warehouse to a Multiprocessor Architecture
- DBMS Schemas for Decision Support
- Data Extraction, Cleanup & Transformation Tools
- Metadata.

Data Warehouse

- The Data warehouse is an environment, not a product.
- It is an architectural construct of an information system that provides users with current and historical decision support information that is hard to access or present in traditional operational data store.
- Data warehousing is a blend of technologies and components aimed at effective integration of operation database into an environment that enables strategic use of data.
- These technologies include relational and multi-dimensional database management system, client/ server architecture, meta-data modeling and repositories, graphical user interface etc.

I.T.S
GHAZIABAD

Data Warehousing Components

Data Warehousing Components

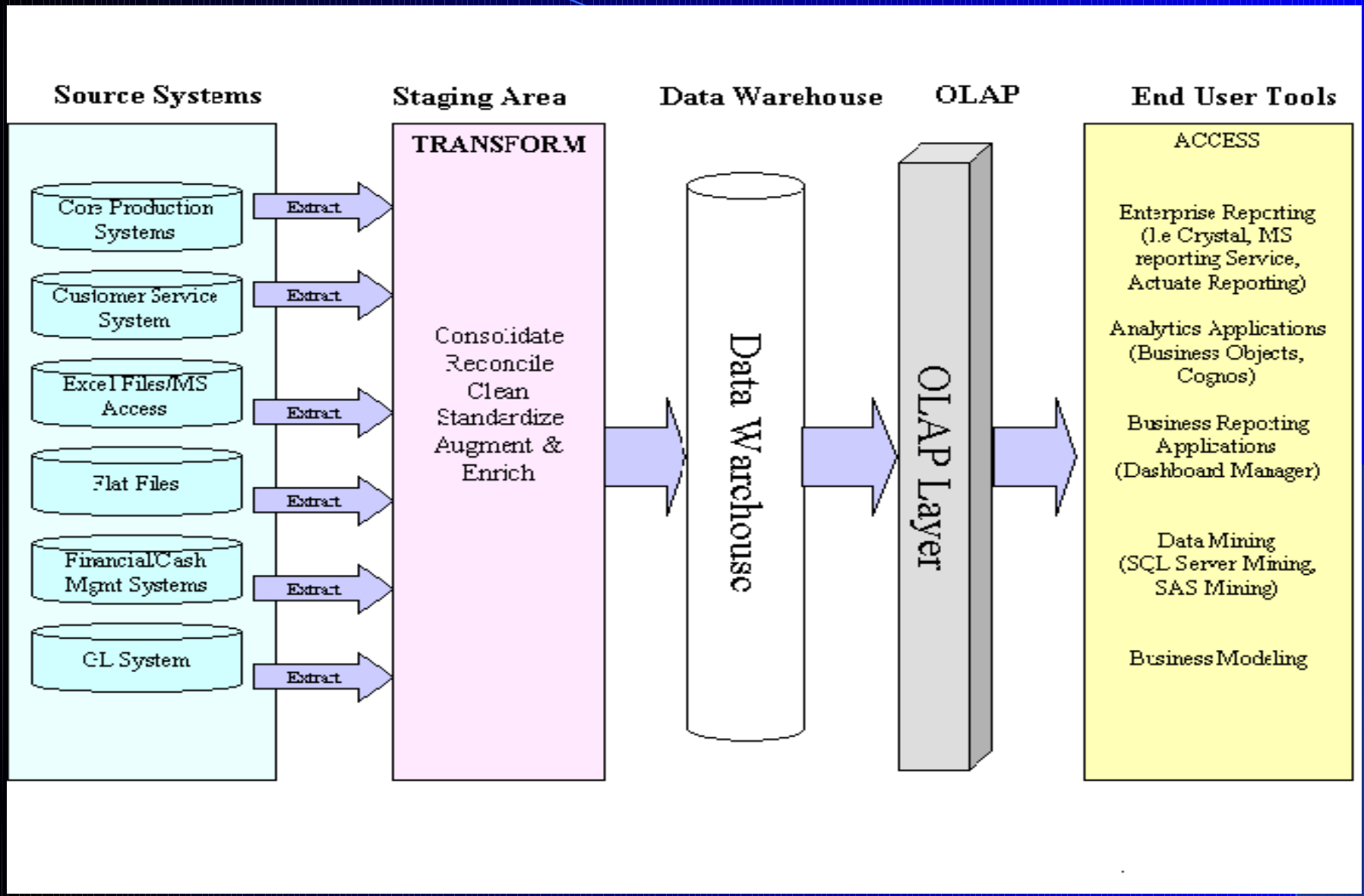
- The data warehouse architecture is based on a relational database management system server that functions as the central repository for informational data. Operational data and processing is completely separated from data warehouse processing. This central information repository is surrounded by a number of key components designed to make the entire environment functional, manageable and accessible by both the operational systems that source data into the warehouse and by end-user query and analysis tools.

Components of Data Warehouse continued…

- There are following **seven** components of a Data Warehouse:
  - **Data Warehouse Database**
  - **Sourcing, Acquisition, Cleanup and Transformation Tools**
  - **Meta Data**
  - **Access (Query) Tools**

    The **query tool** allows executives and other users real-time access to the Data Warehouse database for query generation, result displays, reports and data exports
  - **Data Marts**
  - **Data Warehouse Administration and Management**
  - **Information Delivery System**

Components & Framework



1. Data Warehouse Database

The central data warehouse database is the cornerstone of the data warehousing environment. Certain data warehouse attributes, such as very large database size, ad hoc query processing and the need for flexible user view creation including aggregates, multi-table joins and drill-downs, have become drivers for different technological approaches to the data warehouse database. These approaches include:

- Parallel relational database designs for scalability that include shared-memory, shared disk, or shared-nothing models implemented on various multiprocessor configurations (symmetric multiprocessors or SMP, massively parallel processors or MPP, and/or clusters of uni- or multiprocessors).
- An innovative approach to speed up a traditional RDBMS by using new index structures to bypass relational table scans.
- Multidimensional databases (MDDBs) that are based on proprietary database technology. Multi-dimensional databases are designed to overcome any limitations placed on the warehouse by the nature of the relational data model. MDDBs enable on-line analytical processing (OLAP) tools that architecturally belong to a group of data warehousing components jointly categorized as the data query, reporting, analysis and mining tools.

2. Sourcing, Acquisition, Cleanup and Transformation Tools

The data sourcing, cleanup, transformation and migration tools perform all of the conversions, summarizations, key changes, structural changes and condensations needed to transform disparate data into information that can be used by the decision support tool. They produce the programs and control statements, including the COBOL programs, MVS job-control language (JCL), UNIX scripts, and SQL data definition language (DDL) needed to move data into the data warehouse for multiple operational systems. These tools also maintain the meta data. The functionality includes:

- Removing unwanted data from operational databases
- Converting to common data names and definitions
- Establishing defaults for missing data
- Accommodating source data definition changes

ETL Tools

- **ETL** tools are the equivalent of **schema mappings** in virtual integration, but are more powerful
- **Some of the Well Known ETL Tools**

  The most well known commercial tools are Ab Initio, IBM InfoSphere DataStage, Informatica, Oracle Data Integrator and SAP Data Integrator.

  There are several open source ETL tools, among others:

  Apatar, CloverETL, Pentaho and Talend.
- Arbitrary pieces of code to take data from a source, convert it into data for the warehouse:
  - **Import filters** – read and convert from data sources
  - **Data Transformations** – join, aggregate, filter, convert data
  - **De-duplication** – finds multiple records referring to the same entity, merges them
  - **Profiling** – builds tables, histograms, etc. to summarize data
  - **Quality management** – test against master values, known business rules, constraints, etc.

3. Meta Data

Meta data is data about data that describes the data warehouse. It is used for building, maintaining, managing and using the data warehouse. Meta data can be classified into:

- **Technical meta data**, which contains information about warehouse data for use by warehouse designers and administrators when carrying out warehouse development and management tasks.
- **Business meta data,** which contains information that gives users an easy-to-understand perspective of the information stored in the data warehouse.

4. Access (Query) Tools

Query and Reporting tools can be divided into two groups:

- **Reporting Tools** and **Managed Query Tools**
- ***Reporting tools*** can be further divided into **production reporting** tools and **report writers**.
  - **Production reporting** tools let companies generate regular operational reports or support high-volume batch jobs such as calculating and printing paychecks.
  - **Report writers**, on the other hand, are inexpensive desktop tools designed for end-users.
- ***Managed query tools*** shield end users from the complexities of SQL and database structures by inserting a meta-layer between users and the database. These tools are designed for easy-to-use, point-and-click operations that either accept SQL or generate SQL database queries.

5. Data Mart

- the term data mart means different things to different people. A rigorous definition of this term is a data store that is subsidiary to a data warehouse of integrated data. The data mart is directed at a partition of data (often called a subject area) that is created for the use of a dedicated group of users. These could be classified in two categories:
  - Dependent Data Marts
  - Independent Data Marts

Dependent Data Marts: These types of data marts, data is sourced from the data warehouse, have a high value because no matter how they are deployed and how many different enabling technologies are used, different users are all accessing the information views derived from the single integrated version of the data.

Independent Data Marts: Unfortunately, the misleading statements about the simplicity and low cost of data marts sometimes result in organizations or vendors incorrectly positioning them as an alternative to the data warehouse. This viewpoint defines independent data marts that in fact, represent fragmented point solutions to a range of business problems in the enterprise. This type of implementation should be rarely deployed in the context of an overall technology or applications architecture. Indeed, it is missing the ingredient that is at the heart of the data warehousing concept -- that of data integration.

6. Data Warehouse Administration and Management

Managing data warehouses includes:

1. Security and priority management
2. Monitoring updates from the multiple sources
3. Data quality checks
4. Managing and updating meta data
5. Auditing and reporting data warehouse usage and status
6. Purging data
7. Replicating, sub-setting and distributing data
8. Backup and Recovery and
9. Data warehouse storage management.

7. Information Delivery System

- The information delivery component is used to enable the process of subscribing for data warehouse information and having it delivered to one or more destinations according to some user-specified scheduling algorithm.

- In other words, the information delivery system distributes warehouse-stored data and other information objects to other data warehouses and end-user products such as spreadsheets and local databases.

- Delivery of information may be based on time of day or on the completion of an external event.

- The rationale for the delivery systems component is based on the fact that once the data warehouse is installed and operational, its users don't have to be aware of its location and maintenance.

Building a Data Warehouse

Why a Data Warehouse Application – Business Perspectives

There are several reasons why organizations consider Data Warehousing a critical need. From a business prospective, to strive and succeed in today's highly competetive global environment, business users demand business answers mainly because:

- Decisions need to be made quickly and correctly, using all available data
- Users are business domain experts, not computer professionals
- The amount of data increasing in the data stores, which affects response time and the sheer ability to comprehend its content.
- Competitions is heating up in the areas of business intelligence and added information value.

Building a Data Warehouse

Why a Data Warehouse Application – Technology Perspectives

- There are several technology reasons also for existence of Data Warehousing.
  - First, the Data Warehouse is designed to address the incompatibility of informational and operational transactional systems. These two classes of information systems are designed to satisfy different , often incompatible, requirements.
  - Secondly, the IT infrastructure is changing rapidly, and its capabilities are increasing, as evidenced by the following:
    - The prices of MIPS continues to decline, while the power of processors doubles every 2 years
    - The prices 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

Building a Data Warehouse

1. Business Considerations (Return on Investment)
2. Design Considerations
3. Technical Considerations
4. Implementation Considerations
5. Integrated Solutions
6. Benefits of Data Warehousing

Building a Data Warehouse Contd..

1. Business Considerations (Return on Investment)

1. Approach

- 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 in developing individual data marts, which are then integrated into enterprise data warehouse.

1. Organizational Issues

A Data Warehouse, in general, 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.

Building a Data Warehouse Contd..

2. Design Consideration

To be a successful, a data warehouse designer must take a holistic approach – consider all data warehouse components as parts of a single complex system and take into the account all possible data stores 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 sources, or a selected access tool. This is also one of the reasons why a data warehouse is rather difficult to build. The main factors include:

- **Heterogeneity of Data sources, which affects data conversion, quality, timeliness**
- **Use of historical data, while implies that data may be "old".**
- **Tendency of databases to grow very large**

Building a Data Warehouse Contd..

2. Design Consideration - In addition to the general considerations, there are several specific points relevant to the data warehouse design:

- Data Content
- Metadata
- Data Distribution

  One of the biggest challenge when designing a data warehouse is the data placement and distribution strategy.
- Tools

  These tools provide facilities for defining the transformation and cleanup rules, data movement (from operational sources to the warehouses, end-user query, reporting, and data analysis.
- Performance consideration

Building a Data Warehouse Contd..

3. Technical Considerations

A number of technical issues are to be considered when designing and implementing a Data Warehouse environment.

1. The Hardware Platform that would house the Data Warehouse for parallel query scalability. (Uni-Processor, Multi-processor, etc)

2. The DBMS that supports the warehouse database

3. The communication infrastructure that connects the warehouse, data marts, operational systems, and end users

4. The hardware platform and software to support the metadata repository

5. The systems management framework that enables centralized management and administration to the entire environment.

Building a Data Warehouse Contd..

4. Implementation Considerations

i. Access Tools

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

a. Simple Tabular for reporting
b. Ranking
c. Multi-variable Analysis
d. Time Series Analysis
e. Data Visualization, Graphing, Charting and pivoting
f. Complex Textual Search
g. Statistical Analysis
h. AI Techniques for testing of hypothesis, trends discovery, definition, validation of Data Clusters and segments
i. Information Mapping (i.e. mapping of Spatial Data in geographic information systems)
j. Ad-hoc User Specified Queries
k. Pre-defined repeatable queries
l. Interactive drill-down reporting and analysis
m. Complex queries with multiple joins, multi-level subquesries, and sophisticated search criteria.

Building a Data Warehouse Contd..

4. Implementation Considerations

ii. Data Extraction, Cleanup, Transformation, and Migration

As a components of the Data Warehouse architecture, proper attention must be given to Data Extraction, which represents a critical success factor for a data warehouse architecture.

1. The ability to identify data in the data source environments that can be read by conversion tool is important. This additional step may affect the timeliness of data delivery to the warehouse.
2. Support for the flat files. (VSAM, IMS, IDMS) is critical, since bulk of the corporate data is still maintained in this type of data storage.
3. The capability to merge data from multiple data stores is required in many installations.
4. The specification interface to indicate the data to extracted and the conversion criteria is important.
5. The ability to read information from data dictionaries or import information from repository product is desired.
6. The ability to perform data-type and character-set translation is a requirement when moving data moving between incompatible systems.
7. The capability to create summarization, aggregation, and derivation records and fields is very important.

Building a Data Warehouse Contd..

4. Implementation Considerations

iii. Data Placement Strategies

As Data Warehouse grows, there are at least two options for Data Placement. One is to put some of the data in the data warehouse into another storage media (WORM, RAID). Second option is to distribute data in data warehouse across multiple servers. Some criteria must be established for dividing it over the servers – by geography, organization unit, time, function, etc. However, the data is divided, a single source of meta data across the entire organization is required. Hence this configuration requires both corporation-wide and the meta data managed for any given server.

Building a Data Warehouse Contd..

4. Implementation Considerations

iv. Metadata

A frequently occurring problem in Data Warehouse is the problem of communicating to the end user what information resides in the data warehouse and how it can be accessed. The key to providing users and applications with a roadmap to the information stored in the warehouse is the **metadata**. It can define all data elements and their attributes, data sources and timing, and the rules that govern data use and data transformations. Meta data needs to be collected as the warehouse is designed and built.

Building a Data Warehouse Contd..

4. Implementation Considerations

v. User Sophistication Levels

Data Warehousing is relatively new phenomenon, and a certain degree of sophistication is required on the end user's part to effectively use the warehouse. The users can be classified on the basis of their skill level in accessing the warehouse:

1. Casual Users: These users are most comfortable retrieving information from the warehouse in pre-defined formats, and running preexisting queries and reports.

2. Power Users: In their delay activities, these users typically combine predefined queries with some relatively simple and ad-hoc queries that they create themselves. These users need access tools that combine the simplicity of pre-defined queries and reports with a certain degree of flexibility.

3. Experts: These users tend to create their own queries and perform sophisticated analysis on the information they retrieve from the warehouse. These users know the data, tools and database well enough to demand tools that allow for maximum flexibility and adaptability.

Benefits of Data Warehouse

Successfully implemented data warehousing can realize some significance benefits which can be categorized in two categories:

1. Tangible Benefits:

1. Product inventory turnover is improved
2. Costs of product introduction are decreased with improved selection of target markets.
3. More cost effective decision making is enabled by separating (ad-hoc) query processing from running against operational database.
4. Better business intelligence is enabled by increased quality and flexibility of market analysis available through multi-level data structures, which may range from detailed to highly summarized.

2. Intangible Benefits:

1. Improved productivity
2. Reduced redundant processing, support, and software to support overlapping decision support applications
3. Enhanced Customer relations through improved knowledge of individual requirements and trends, through customization, improved communications, and tailored product offerings.
4. Enabling business process reengineering – data warehousing can provide useful insights into work process themselves,

Warehouse Database

- The organizations that embarked on data warehousing development deal with ever increasing amounts of data. Generally speaking, the size of a data warehouse rapidly approaches the point where the search for better performance and scalability becomes a real necessity. This search aims to pursue two goals:
  - ***Speed-up:*** the ability to execute the same request on the same amount data in less time
  - ***Scale-up:*** the ability to obtain the same performance on the same request as the database size increases.

  An additional and important goal is to achieve ***linear*** speed-up and scale-up, doubling the number of processors cuts the response time in half (linear speed-up) or provides the same performance on twice as much data (linear scale-up).

Mapping the Data Warehouse to a Multiprocessor Architecture

- The goals of linear performance and scalability (discussed in previous slide) can be satisfied by parallel hardware architectures, parallel operating systems, and parallel DBMSs. Parallel hardware architectures are based on Multi-processor systems designed as a Shared-memory model (symmetric multiprocessors), Shared-disk model or distributed-memory model (MPP and Clusters of SMPs). Parallelism can be achieved in two different ways:
  - Horizontal Parallelism (Database is partitioned across different disks)
  - Vertical Parallelism (occurs among different tasks – all components query operations i.e. scans, join, sort)
  - Data Partitioning

Database Architectures for Parallel Processing

- Shared-memory Architecture
- Shared Disk Architecture
- Shared-nothing Architecture
- Combined Architecture

Parallel RDBMS Features

- Data Warehouse development requires a good understanding of all architectural components, including the data warehouse DBMS Platform. Understanding the basic architecture of Warehouse database is the first step in evaluating and selecting a product.
- State of the art parallel features the developers and users of the Warehouse should demand from the DBMS vendor:
  - Scope and techniques of Parallel DBMS
    - Queries (Insert/ Update/Delete)
    - DBMS that supports parallel database load, backup, reorganization and recovery is much better positioned for VLDBs.
  - Optimizer Implementation
  - Application Transparency
  - The Parallel environment
  - DBMS Management Tools
  - Price/ Performance

Parallel DBMS Vendors

- **ORACLE** – Oracle supports Parallel Database processing with its add-on **Oracle Parallel Server Option (OPS)** and **Parallel Query Option (PQO)** with Query Coordinator.
- **Informix** – Informix developed its **Dynamic Scalable Architecture (DSA) to support Shared-Memory, Shared-Disk, and Shared-Nothing Models**. Informix OnLine release 8, also known as XPS (eXtended Parallel Server), supports MPP Hardware platforms that include IBM, SP, AT & T, Sun, HP, ICL Goldrush, with sequent, Siemens, Pyramid etc.
- **IBM** – DB2 Parallel Edition **(DB2 PE)**, a Database based on DB2/6000 Server Architecture; latest version is **DB2 Universal Database**.
- **Sybase** – Sybase implemented its parallel DBMS functionality in a product called **SYBASE MPP** (formerly Navigational Server). It was jointly developed by Sybase and NCR (formerly AT&T GIS), and its first release was targeted for the AT&T 3400, 3500 (both SMP) and 3600 (MPP) Platforms.
- Other RDBMS Products **i.** NCR Teradata **ii.** Tandem NonStop SQL/MP
- Specialized Database Products - **i.** Red Brick Systems
  ii. White Cross Systems Inc.

DBMS Schemas for Decision Support

- Data Warehousing projects were forced to choose between a data model and a corresponding database schema that is intuitive for analysis but performs poorly and a model-schema that performs better but is not well suited for analysis.
- As Data Warehousing continued to mature, new approaches to schema design resulted in schemas better suited to business analysis that is so crucial to successful data warehousing.
- The schema methodology that is gaining widespread acceptance for Data Warehousing is the **Star Schema**.

Data Layout for best Access

- The original objective in developing an abstract model known as Relational Model were to address a number of shortcomings of non-relational DBMS and application development.
- The typical requirements for the RDBMS supporting operational systems are based on the need to effectively support a large number of small but simultaneous read and write requests.
- The demand placed on the RDBMS by a Data Warehouse are very different. A data warehouse RDBMS typically needs to process queries that are large, complex, ad-hoc and data intensive.
- Solving modern business problems such as market analysis and financial forecasting requires query-centric database schemas that are array-oriented and multi-dimensional in nature.

Multi-dimensional Data Model

- The Multi-dimensional nature of business questions is reflected in the fact that, for example, marketing managers are no longer satisfied by asking simple one-dimensional questions such as "How much revenue did the new product generate by month, in northeastern division, broken down by user demographic, by sales office, relative to the previous version of the product, compared with the plan?" – a six dimensional question.

STAR SCHEMA

- The Multi-dimensional view of Data that is expressed using relational database semantics is provided by the database schema design called Star Schema.
- The basic premise of Star Schema is that information can be classified into two groups: **facts** and **dimensions**.
- **Facts** are the core Data element being analyzed. For example, units of individual items sold are facts.
- **Dimensions** are attributes about the facts. For example, dimensions are the product types purchased and date of purchase.

Data Extraction, Cleanup & Transformation Tools

- The task of capturing data from a source data system, cleaning and transforming it and then loading the results into a target data system can be carried out either by separate products, or by a single integrated solution. More contemporary integrated solutions can fall into one of the categories described below:
  - Code Generators
  - Database data Replications
  - Rule-driven Dynamic Transformation Engines (Data Mart Builders)

Code Generator

- It creates 3GL/4GL transformation programs based on source and target data definitions, and data transformation and enhancement rules defined by the developer.
- This approach reduces the need for an organization to write its own data capture, transformation, and load programs. These products employ DML Statements to capture a set of the data from source system.
- These are used for data conversion projects, and for building an enterprise-wide data warehouse, when there is a significant amount of data transformation to be done involving a variety of different flat files, non-relational, and relational data sources.

Database Data Replication Tools

- These tools employ database triggers or a recovery log to capture changes to a single data source on one system and apply the changes to a copy of the data source data located on a different system.
- Most replication products do not support the capture of changes to non-relational files and databases, and often do not provide facilities for significant data transformation and enhancement.
- These point-to-point tools are used for disaster recovery and to build an operational data store, a data warehouse, or a data mart when the number of data sources involved are small and a limited amount of data transformation and enhancement is required.

Rule-driven Dynamic Transformation Engines

- They are also known as Data Mart Builders and capture data from a source system at User-defined intervals, transform data, and then send and load the results into a target environment, typically a data mart.
- To date most of the products of this category support only relational data sources, though now this trend have started changing.
- Data to be captured from source system is usually defined using query language statements, and data transformation and enhancement is done on a script or a function logic defined to the tool.
- With most tools in this category, data flows from source systems to target systems through one or more servers, which perform the data transformation and enhancement. These transformation servers can usually be controlled from a single location, making the job of such environment much easier.