MAILAM Engineering College

Approved by AICTE, New Delhi, affiliated to Anna University, Chennai, Accredited by NBA & TCS

CCS341 – DATA WAREHOUSING

[REGULATION-2021]

STUDY MATERIAL

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

NAME OF THE STUDENT:……

REGISTER NUMBER:……

YEAR / SEM:……

ACADEMIC YEAR:……

PREPARED BY

Mrs. G. Vasanthi, ASP/AI & DS

MAILAM
Engineering College
Approved by AICTE, New Delhi, affiliated to Anna University, Chennai, Accredited by NBA & TCS

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

CCS341 – DATA WAREHOUSING

III Yr. / VI SEM

SYLLABUS

COURSE OBJECTIVES:

- To know the details of data warehouse Architecture
- To understand the OLAP Technology
- To understand the partitioning strategy
- To differentiate various schema
- To understand the roles of process manager & system manager

UNIT I INTRODUCTION TO DATA WAREHOUSE 5

Data warehouse Introduction - Data warehouse components- operational database Vs data warehouse – Data warehouse Architecture – Three-tier Data Warehouse Architecture - Autonomous Data Warehouse- Autonomous Data Warehouse Vs Snowflake - Modern Data Warehouse.

UNIT II ETL AND OLAP TECHNOLOGY 6

What is ETL – ETL Vs ELT – Types of Data warehouses - Data warehouse Design and Modeling - Delivery Process - Online Analytical Processing (OLAP) - Characteristics of OLAP - Online Transaction Processing (OLTP) Vs OLAP - OLAP operations- Types of OLAP- ROLAP Vs MOLAP Vs HOLAP.

UNIT III META DATA, DATA MART AND PARTITION STRATEGY 7

Meta Data – Categories of Metadata – Role of Metadata – Metadata Repository – Challenges for Meta Management - Data Mart – Need of Data Mart- Cost Effective Data Mart- Designing Data Marts- Cost of Data Marts- Partitioning Strategy – Vertical partition – Normalization – Row Splitting – Horizontal Partition.

UNIT IV SYSTEM & PROCESS MANAGERS 6

Dimensional Modeling- Multi-Dimensional Data Modeling – Data Cube- Star Schema- Snowflake schema- Star Vs Snowflake schema- Fact constellation Schema- Schema Definition - Process Architecture- Types of Data Base Parallelism – Datawarehouse Tools.

UNIT V PREDICTIVE ANALYTICS 6

Data Warehousing System Managers: System Configuration Manager- System Scheduling Manager - System Event Manager - System Database Manager - System Backup Recovery Manager - Data Warehousing Process Managers: Load Manager – Warehouse Manager- Query Manager – Tuning – Testing.

TOTAL: 30 PERIODS

COURSE OUTCOMES:

CO1: Design data warehouse architecture for various Problems

CO2: Apply the OLAP Technology

CO3: Analyse the partitioning strategy

CO4: Critically analyze the differentiation of various schema for given problem

CO5: Frame roles of process manager & system manager

TEXT BOOKS

1. Alex Berson and Stephen J. Smith "Data Warehousing, Data Mining & OLAP", Tata McGraw – Hill Edition, Thirteenth Reprint 2008.
2. Ralph Kimball, "The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling", Third edition, 2013.

REFERENCES

1. Paul Raj Ponniah, "Data warehousing fundamentals for IT Professionals", 2012.
2. K.P. Soman, ShyamDiwakar and V. Ajay "Insight into Data mining Theory and Practice", Easter Economy Edition, Prentice Hall of India, 2006.

ANNA UNIVERSITY UPDATED QP – AP 2023, ND 2023, ND 2024

Prepared by –

Mrs.G.Vasanthi, ASP/AI&DS

Verified By: Dr. S. Artheeswari, HOD/AI&DS

PRINCIPAL

CCS341 DATA WAREHOUSING

SYLLABUS

UNIT I INTRODUCTION TO DATA WAREHOUSE 5

Data warehouse Introduction - Data warehouse components- operational database Vs data warehouse – Data warehouse Architecture – Three-tier Data Warehouse Architecture - Autonomous Data Warehouse- Autonomous Data Warehouse Vs Snowflake - Modern Data Warehouse

UNIT II ETL AND OLAP TECHNOLOGY 6

What is ETL – ETL Vs ELT – Types of Data warehouses - Data warehouse Design and Modeling - Delivery Process - Online Analytical Processing (OLAP) - Characteristics of OLAP – Online Transaction Processing (OLTP) Vs OLAP - OLAP operations- Types of OLAP- ROLAP Vs MOLAP Vs HOLAP.

UNIT III META DATA, DATA MART AND PARTITION STRATEGY 7

Meta Data – Categories of Metadata – Role of Metadata – Metadata Repository – Challenges for Meta Management - Data Mart – Need of Data Mart- Cost Effective Data Mart- Designing Data Marts- Cost of Data Marts- Partitioning Strategy – Vertical partition – Normalization – Row Splitting – Horizontal Partition

UNIT IV DIMENSIONAL MODELING AND SCHEMA 6

Dimensional Modeling- Multi-Dimensional Data Modeling – Data Cube- Star Schema- Snowflake schema- Star Vs Snowflake schema- Fact constellation Schema- Schema Definition – Process Architecture- Types of Data Base Parallelism – Data warehouse Tools

UNIT V SYSTEM & PROCESS MANAGERS 6

Data Warehousing System Managers: System Configuration Manager- System Scheduling Manager - System Event Manager - System Database Manager - System Backup Recovery Manager - Data Warehousing Process Managers: Load Manager – Warehouse Manager- Query Manager – Tuning – Testing

Total : 30 Periods

UNIT I INTRODUCTION TO DATA WAREHOUSE 5

Data warehouse Introduction - Data warehouse components- operational database Vs data warehouse – Data warehouse Architecture – Three-tier Data Warehouse Architecture - Autonomous Data Warehouse- Autonomous Data Warehouse Vs Snowflake - Modern Data Warehouse

PART A

1. Write the different steps in Knowledge Discovery in Databases. [Nov 2024]

- Understanding the Data Set.
- Data Selection.
- Cleaning and Pre-processing.
- Data Transformation.
- Select the Appropriate Data Mining Task.
- Choice of Data Mining Algorithms.
- Application of Data Mining Algorithms.
- Evaluation.

2. How is Data warehouse different from a database? Identify the similarity.

[Nov 2024]

- A database stores the current data required to power an application whereas a data warehouse stores current and historical data for one or more systems in a predefined and fixed schema for the purpose of analyzing the data.

3. What is meant Data Warehouse? [NOV 2023]

- A **Data Warehouse** is a system that is used by the users or knowledge managers for data analysis and decision-making.
- It can construct and present the data in a certain structure to fulfill the diverse requirements of several users.
- Data warehouses are also known as **Online Analytical Processing (OLAP) Systems.**

4. What is an Operational Database?

- The type of database system that stores information related to operations of an enterprise is referred to as an **operational database.**
- Operational databases are required for functional lines like marketing, employee relations, customer service etc.

- Operational databases are basically the sources of data for the data warehouses because they contain detailed data required for the normal operations of the business.
- In an operational database, the data changes when updates are created and shows the latest value of the final transaction.
- They are also known as OLTP **(Online Transactions Processing Databases).** These databases are used to manage dynamic data in real-time.

5. Differentiate Operational Databases Vs Data Warehouse.

- A **data warehouse** is a repository for structured, filtered data that has already been processed for a specific purpose. It collects the data from multiple sources and transforms the data using ETL process, then loads it to the Data Warehouse for business purpose.
- An **operational database,** on the other hand, is a database where the data changes frequently. They are mainly designed for high volume of data transaction. They are the source database for the data warehouse. Operational databases are used for recording online transactions and maintaining integrity in multi-access environments.

6. What is meant Operational System?

- An **operational system** is a method used in data warehousing to refer to a **system** that is used to process the day-to-day transactions of an organization.

7. What is meant Flat Files?

- A **Flat file** system is a system of files in which transactional data is stored, and every file in the system must have a different name.

8. What is meant Meta Data?

- A set of data that defines and gives information about other data.
- Meta Data used in Data Warehouse for a variety of purpose, including:
- Meta Data summarizes necessary information about data, which can make finding and work with particular instances of data more accessible. For

example, author, data build, and data changed, and file size are examples of very basic document metadata.

- Metadata is used to direct a query to the most appropriate data source.

9. Difference between Data Warehouse and Operational Database.

Key	Data Warehouse	Operational Database
Basic	A data warehouse is a repository for structured, filtered data that has already been processed for a specific purpose.	Operational Database are those databases where data changes frequently.
Data Structure	Data warehouse has de-normalized schema.	It has normalized schema.
Performance	It is fast for analysis queries.	It is slow for analytics queries.
Type of Data	It focuses on historical data.	It focuses on current transactional data.
Use Case	It is used for OLAP.	It is used for OLTP.

10. What is the difference between autonomous and Snowflake?

- With Cloud Customer solutions, Autonomous Data Warehouse can be deployed as a fully managed cloud service in a customer's own data centers to address data residency and security requirements. Snowflake only runs in the public cloud.

11. What is Modern Data Warehouse & its uses?

- A Modern Data Warehouse is a cloud-based solution that helps organizations gather, store, and process data to help you make intelligent decisions.
- A variety of organizations can use a modern data warehouse to improve business processes such as finances, human resources, and operations.

12. What are the five different components of a Modern Data Warehouse?

Level 1: Data Acquisition

Level 2: Data Engineering

Level 3: Data Management Governance

Level 4: Reporting and Business Intelligence

Level 5: Data Science

13. What are the various end-user access tools?

- Reporting and Query Tools
- Application Development Tools
- Executive Information Systems Tools
- Online Analytical Processing Tools
- Data Mining Tools

14. What are the Properties of Data Warehouse Architectures?

- Separation
- Scalability
- Extensibility
- Security
- Administer ability

15. What are the various Data acquisition sources?

- IoT devices
- Social media posts

- YouTube videos
- Website content
- Customer data
- Enterprise Resource Planning
- Legacy data stores

16. What are the various Data Flow Stages?

- Source layer
- Data Staging
- Data Warehouse layer
- Analysis

17. Write short notes on Data Warehouse.

- A data warehouse is subject-oriented since it provides topic-wise information rather than the overall processes of a business.
- Such subjects may be sales, promotion, inventory, etc.
- For example, if you want to analyze your company's sales data, you need to build a data warehouse that concentrates on sales.

18. What are types of data warehouse?

- The three main types of data warehouses are enterprise data warehouse (EDW), operational data store (ODS), and data mart.

19. What are the 4 characteristics of data warehouse?

- Subject oriented
- Time variant
- Integrated
- Non-volatile

20. What is a data warehouse vs database?

- A database stores the current data required to power an application whereas a data warehouse stores current and historical data for one or more systems in a predefined and fixed schema for the purpose of analyzing the data.

21. What is the process of data warehouse?

- Data warehousing is a process used to collect and manage data from multiple sources into a centralized repository to drive actionable business insights.

22. What is autonomous data warehouse?

- Autonomous Data Warehouse continuously monitors all aspects of system performance. It adjusts autonomously to ensure consistent high performance even as workloads, query types, and the number of users vary over time.

23. What is the benefit of autonomous database?

- Autonomous databases offer greater visibility into database performance and allow you to improve database security. You can use them to power new digital business applications and services that require fast scalability and change management so you can meet increased customer demand.

24. What is autonomous used for?

- An autonomous system is one that can achieve a given set of goals in a changing environment—gathering information about the environment and working for an extended period of time without human control or intervention.
- Driverless cars and autonomous mobile robots (AMRs) used in warehouses are two common examples.

25. What is an example of modern data warehouse?

- Some of the more notable cloud data warehouses in the market include Amazon Redshift, Google Big Query, Snowflake, and Microsoft Azure SQL Data Warehouse.

26. What are the components of a modern data warehouse?

- A typical data warehouse has four main components: a central database, ETL (extract, transform, load) tools, metadata, and access tools.

- All of these components are engineered for speed so that you can get results quickly and analyze data on the fly.

27. What is the difference between traditional and modern data warehouse?

- Modern data warehouses differ from traditional warehouses in the following ways: There is no need to purchase physical hardware.
- They are less complex to set up. It is much easier to prototype and provide business value without having to build out the ETL processes right away.

28. Why we need a separate Data Warehouse?

- Data Warehouse queries are complex because they involve the computation of large groups of data at summarized levels.
- It may require the use of distinctive data organization, access, and implementation method based on multidimensional views.
- Performing OLAP queries in operational database degrade the performance of functional tasks.
- Data Warehouse is used for analysis and decision making in which extensive database is required, including historical data, which operational database does not typically maintain.
- The separation of an operational database from data warehouses is based on the different structures and uses of data in these systems.
- Because the two systems provide different functionalities and require different kinds of data, it is necessary to maintain separate databases.

29. List out the benefits of Data warehousing, [NOV/DEC 2023]

- Maintain data quality and consistency. ...
- Combine data from diverse sources. ...
- Eliminate data silos. ...
- Enable business automation. ...
- Learn more about your customers. ...
- Gain historical intelligence of your business activity. ...

- Increase data security.

30. How is data warehouse different from a database? Identify the similarity.

Difference	Data warehouse	Database
Purpose	Analysis of data	Recording data
Data Type	Historical Data (often summarized)	Real Time (Detailed data including metadata)
Processing Method	OLAP (online analytical processing)	OLTP (online transactional processing)
Type of collection	Subject-oriented	Application-oriented
Users	Limited	Can vary from 00's to 000's and more
Query	Complex analytical queries	Transaction queries (CRUD)
Service Level Agreement (SLA)	99.99 upwards for mission critical apps	Flexible (refreshes usually occur once a day)

Similarities:

1. **Data Storage:**
   - Both store data persistently using structured formats (tables) and manage data integrity.
2. **Query Language:**
   - Both use SQL (Structured Query Language) for querying and manipulating data.
3. **Data Management:**
   - Both employ indexing and data management techniques to optimize data retrieval and ensure data consistency.

PART B

1. Explain in detail about Data Warehouse.

Data Warehouse

- A **Data Warehouse** is a system that is used by the users or knowledge managers for data analysis and decision-making. It can construct and present the data in a certain structure to fulfill the diverse requirements of several users. Data warehouses are also known as **Online Analytical Processing (OLAP) Systems.**
- In a data warehouse or OLAP system, the data is saved in a format that allows the effective creation of data mining documents. The data structure in a data warehousing has denormalized schema. Performance-wise, data warehouses are quite fast when it comes to analyzing queries.
- Data warehouse systems do the integration of several application systems. These systems then provide data processing by supporting a solid platform of consolidated historical data for analysis.

Operational Database

- The type of database system that stores information related to operations of an enterprise is referred to as an **operational database.** Operational databases are required for functional lines like marketing, employee relations, customer service etc. Operational databases are basically the sources of data for the data warehouses because they contain detailed data required for the normal operations of the business.
- In an operational database, the data changes when updates are created and shows the latest value of the final transaction. They are also known as **OLTP (Online Transactions Processing Databases).** These databases are used to manage dynamic data in real-time.

Operational Databases Vs Data Warehouse

- A **data warehouse** is a repository for structured, filtered data that has already been processed for a specific purpose. It collects the data from multiple sources

and transforms the data using ETL process, then loads it to the Data Warehouse for business purpose.

- An **operational database,** on the other hand, is a database where the data changes frequently. They are mainly designed for high volume of data transaction. They are the source database for the data warehouse. Operational databases are used for recording online transactions and maintaining integrity in multi-access environments.

Difference between Data Warehouse and Operational Database

Key	**Data Warehouse**	**Operational Database**
Basic	A data warehouse is a repository for structured, filtered data that has already been processed for a specific purpose.	Operational Database are those databases where data changes frequently.
Data Structure	Data warehouse has de-normalized schema.	It has normalized schema.
Performance	It is fast for analysis queries.	It is slow for analytics queries.
Type of Data	It focuses on historical data.	It focuses on current transactional data.
Use Case	It is used for OLAP.	It is used for OLTP.

Some key differences between operational database systems and data warehouses include:

- **Purpose:** Operational database systems are used to support day-to-day operations of an organization, while data warehouses are used to support decision-making and analysis activities.
- **Data Structure:** Operational database systems typically have a normalized data structure, which means that the data is organized into many related tables to reduce data redundancy and improve data consistency. Data warehouses, on the other hand, typically have a denormalized data structure, which means that the data is organized into fewer tables optimized for reporting and analysis.
- **Data Volume:** Operational database systems typically store a smaller volume of data compared to data warehouses, which may store years of historical data.
- **Performance:** Operational database systems are optimized for transaction processing and are designed to support high-volume, high-speed transaction processing. Data warehouses, on the other hand, are optimized for querying and reporting and are designed to support complex analytical queries that may involve large volumes of data.

2. Explain in detail about the Components or Building Blocks of Data Warehouse.

- Architecture is the proper arrangement of the elements. We build a data warehouse with software and hardware components.
- To suit the requirements of our organizations, we arrange these building we may want to boost up another part with extra tools and services. All of these depends on our circumstances.

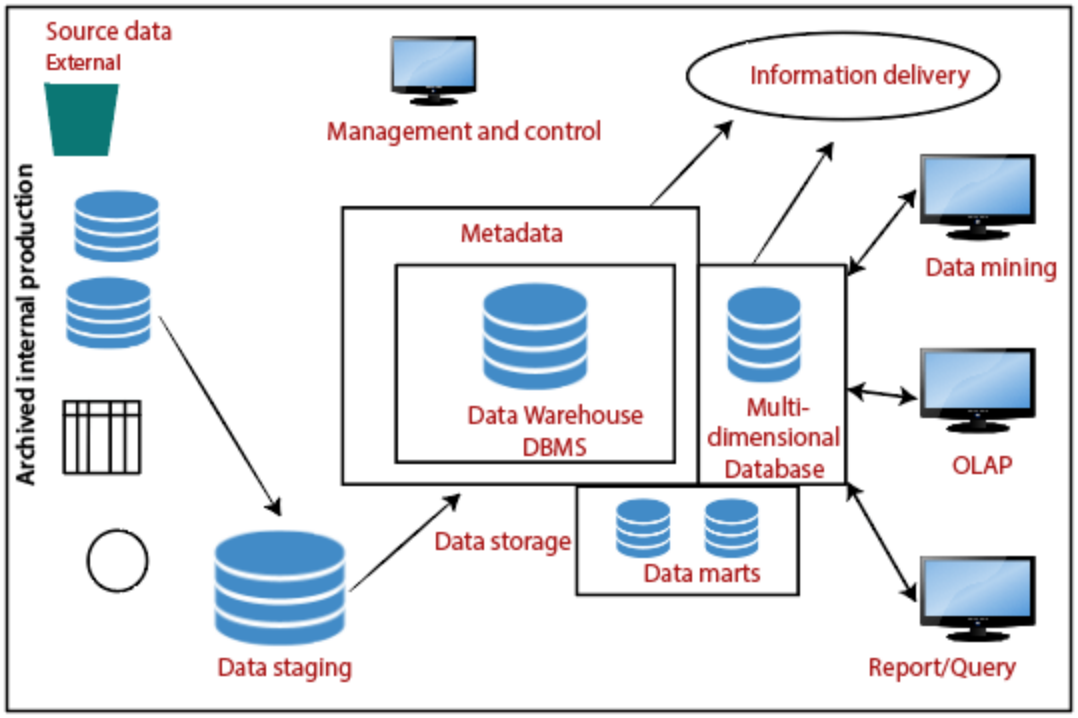


Fig.1.1 Components of a Data Warehouse

- The figure 1.1 shows the essential elements of a typical warehouse. We see the Source Data component shows on the left.
- The Data staging element serves as the next building block. In the middle, we see the Data Storage component that handles the data warehouses data.
- This element not only stores and manages the data; it also keeps track of data using the metadata repository.
- The Information Delivery component shows on the right consists of all the different ways of making the information from the data warehouses available to the users.

Source Data Component

Source data coming into the data warehouses may be grouped into four broad categories:

Production Data:

- This type of data comes from the different operating systems of the enterprise.

- Based on the data requirements in the data warehouse, we choose segments of the data from the various operational modes.

Internal Data:

- In each organization, the client keeps their "private" spreadsheets, reports, customer profiles, and sometimes even department databases. This is the internal data, part of which could be useful in a data warehouse.

Archived Data:

- Operational systems are mainly intended to run the current business. In every operational system, we periodically take the old data and store it in achieved files.

External Data:

- Most executives depend on information from external sources for a large percentage of the information they use. They use statistics associating to their industry produced by the external department.

Data Staging Component

- After we have been extracted data from various operational systems and external sources, we have to prepare the files for storing in the data warehouse.
- The extracted data coming from several different sources need to be changed, converted, and made ready in a format that is relevant to be saved for querying and analysis.

The three primary functions in the staging area

1) Data Extraction:

- This method has to deal with numerous data sources. We have to employ the appropriate techniques for each data source.

2) Data Transformation:

- As we know, data for a data warehouse comes from many different sources.
- If data extraction for a data warehouse posture big challenges, data transformation present even significant challenges.

- We perform several individual tasks as part of data transformation.
- First, we clean the data extracted from each source. Cleaning may be the correction of misspellings or may deal with providing default values for missing data elements, or elimination of duplicates when we bring in the same data from various source systems.
- Standardization of data components forms a large part of data transformation. Data transformation contains many forms of combining pieces of data from different sources. We combine data from single source record or related data parts from many source records.
- On the other hand, data transformation also contains purging source data that is not useful and separating outsource records into new combinations.
- Sorting and merging of data take place on a large scale in the data staging area. When the data transformation function ends, we have a collection of integrated data that is cleaned, standardized, and summarized.

3) Data Loading:

- Two distinct categories of tasks form data loading functions.
- When we complete the structure and construction of the data warehouse and go live for the first time, we do the initial loading of the information into the data warehouse storage.
- The initial load moves high volumes of data using up a substantial amount of time.

Data Storage Components

- Data storage for the data warehousing is a split repository.
- The data repositories for the operational systems generally include only the current data. Also, these data repositories include the data structured in highly normalized for fast and efficient processing.

Information Delivery Component

- The information delivery element as shown in figure 1.2 is used to enable the process of subscribing for data warehouse files and having it transferred to one or more destinations according to some customer-specified scheduling algorithm.

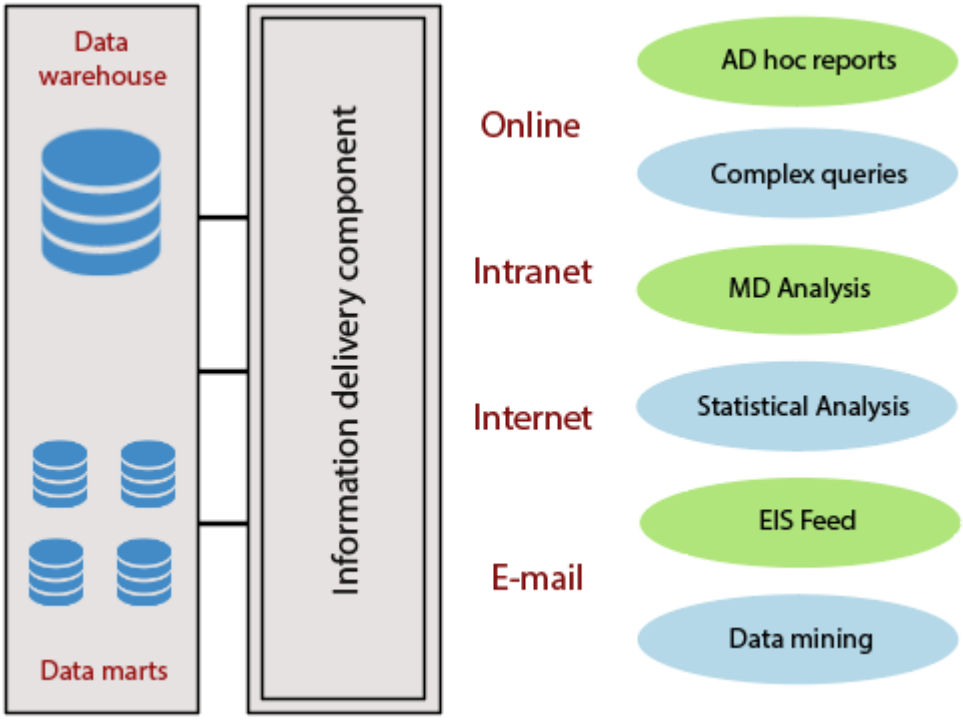


Fig.1.2 Information Delivery Component

Metadata Component

- Metadata in a data warehouse is equal to the data dictionary or the data catalog in a database management system.
- In the data dictionary, we keep the data about the logical data structures, the data about the records and addresses, the information about the indexes, and so on.

Data Marts

- It includes a subset of corporate-wide data that is of value to a specific group of users. The scope is confined to particular selected subjects.

- Data in a data warehouse should be a fairly current, but not mainly up to the minute, although development in the data warehouse industry has made standard and incremental data dumps more achievable.
- Data marts are lower than data warehouses and usually contain organization.
- The current trends in data warehousing are to developed a data warehouse with several smaller related data marts for particular kinds of queries and reports.

Management and Control Component

- The management and control elements coordinate the services and functions within the data warehouse.
- These components control the data transformation and the data transfer into the data warehouse storage.
- On the other hand, it moderates the data delivery to the clients.
- Its work with the database management systems and authorizes data to be correctly saved in the repositories.
- It monitors the movement of information into the staging method and from there into the data warehouses storage itself.

3. Difference between Operational Database and Data warehouse.

Operational Database	Data Warehouse
Operational systems are designed to support high-volume transaction processing.	Data warehousing systems are typically designed to support high-volume analytical processing (i.e., OLAP).
Operational systems are usually concerned with current data.	Data warehousing systems are usually concerned with historical data.
Data within operational systems are mainly updated regularly according to need.	Non-volatile, new data may be added regularly. Once Added rarely changed.
It is designed for real-time business dealing and processes.	It is designed for analysis of business measures by subject area, categories,

	and attributes.
It is optimized for a simple set of transactions, generally adding or retrieving a single row at a time per table.	It is optimized for extent loads and high, complex, unpredictable queries that access many rows per table.
It is optimized for validation of incoming information during transactions, uses validation data tables.	Loaded with consistent, valid information, requires no real-time validation.
It supports thousands of concurrent clients.	It supports a few concurrent clients relative to OLTP.
Operational systems are widely process-oriented.	Data warehousing systems are widely subject-oriented
Operational systems are usually optimized to perform fast inserts and updates of associatively small volumes of data.	Data warehousing systems are usually optimized to perform fast retrievals of relatively high volumes of data.
Data In	Data Out
Less Number of data accessed.	Large Number of data accessed.
Relational databases are created for on-line transactional Processing (OLTP)	Data Warehouse designed for on-line Analytical Processing (OLAP)

4. Explain in detail about Data Warehouse Architecture.

Data Warehouse Architecture

- A data warehouse architecture is a method of defining the overall architecture of data communication processing and presentation that exist for end-clients computing within the enterprise. Each data warehouse is different, but all are characterized by standard vital components.
- Production applications such as payroll accounts payable product purchasing and inventory control are designed for online transaction processing **(OLTP)**. Such applications gather detailed data from day to day operations.

- Data Warehouse applications are designed to support the user ad-hoc data requirements, an activity recently dubbed online analytical processing (OLAP). These include applications such as forecasting, profiling, summary reporting, and trend analysis.
- Production databases are updated continuously by either by hand or via OLTP applications. In contrast, a warehouse database is updated from operational systems periodically, usually during off-hours. As OLTP data accumulates in production databases, it is regularly extracted, filtered, and then loaded into a dedicated warehouse server that is accessible to users.
- As the warehouse is populated, it must be restructured tables de-normalized, data cleansed of errors and redundancies and new fields and keys added to reflect the needs to the user for sorting, combining, and summarizing data.
- Data warehouses and their architectures very depending upon the elements of an organization's situation.

Three common architectures are:

1. Data Warehouse Architecture: Basic – Figure 1.3
2. Data Warehouse Architecture: With Staging Area
3. Data Warehouse Architecture: With Staging Area and Data Marts

1. Data Warehouse Architecture: Basic

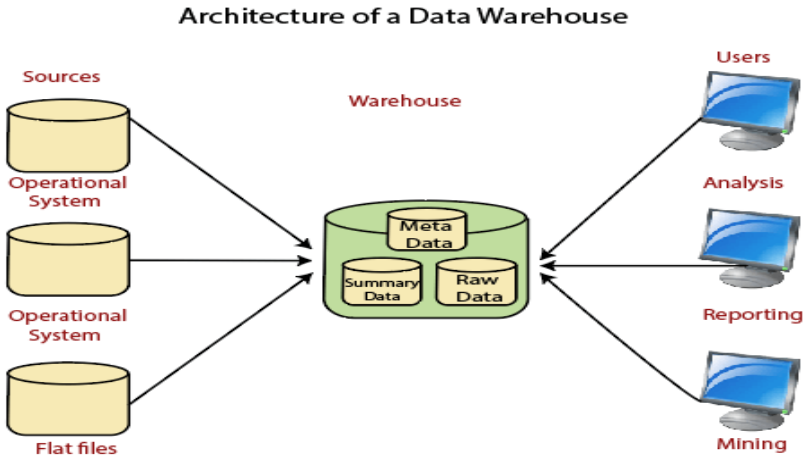


Fig.1.3 Architecture of a Data Warehouse - Basic

Operational System

- An **operational system** is a method used in data warehousing to refer to a **system** that is used to process the day-to-day transactions of an organization.

Flat Files

- A **Flat file** system is a system of files in which transactional data is stored, and every file in the system must have a different name.

Meta Data

- A set of data that defines and gives information about other data.
- Meta Data used in Data Warehouse for a variety of purpose, including:
- Meta Data summarizes necessary information about data, which can make finding and work with particular instances of data more accessible. For example, author, data build, and data changed, and file size are examples of very basic document metadata.
- Metadata is used to direct a query to the most appropriate data source.

Lightly and Highly Summarized data

- The area of the data warehouse saves all the predefined lightly and highly summarized (aggregated) data generated by the warehouse manager.
- The goals of the summarized information are to speed up query performance. The summarized record is updated continuously as new information is loaded into the warehouse.

End-User access Tools

- The principal purpose of a data warehouse is to provide information to the business managers for strategic decision-making. These customers interact with the warehouse using end-client access tools.

The examples of some of the end-user access tools can be:

  - Reporting and Query Tools
  - Application Development Tools

- Executive Information Systems Tools
- Online Analytical Processing Tools
- Data Mining Tools

2. Data Warehouse Architecture: With Staging Area

- We must clean and process your operational information before put it into the warehouse.
- We can do this programmatically, although data warehouses uses a **staging area** (A place where data is processed before entering the warehouse).
- A staging area simplifies data cleansing and consolidation for operational method coming from multiple source systems, especially for enterprise data warehouses where all relevant data of an enterprise is consolidated as shown in figure 1.4.

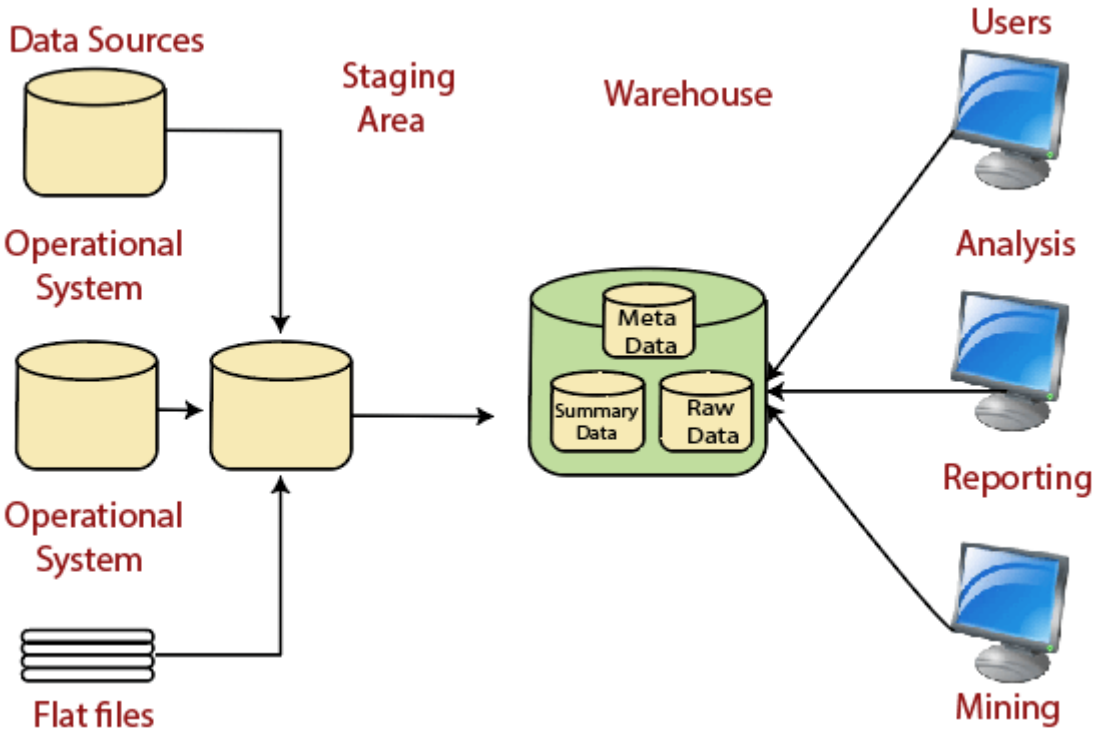


Fig.1.4. Architecture of a Data Warehouse with a Staging Area

Data Warehouse Staging Area is a temporary location where a record from source systems is copied as in figure .

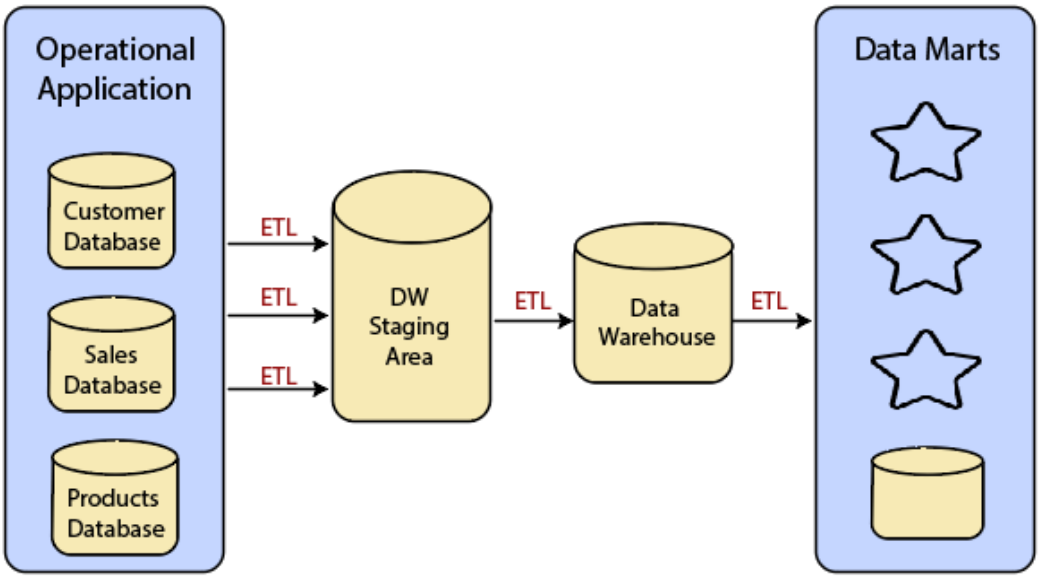


Fig.1.5. Architecture of a Data Warehouse with a Staging Area

3. Data Warehouse Architecture: With Staging Area and Data Marts

- We may want to customize our warehouse's architecture for multiple groups within our organization.
- We can do this by adding data marts. A data mart is a segment of a data warehouses that can provided information for reporting and analysis on a section, unit, department or operation in the company, e.g., sales, payroll, production, etc.
- The figure 1.6 illustrates an example where purchasing, sales, and stocks are separated. In this example, a financial analyst wants to analyze historical data for purchases and sales or mine historical information to make predictions about customer behavior.

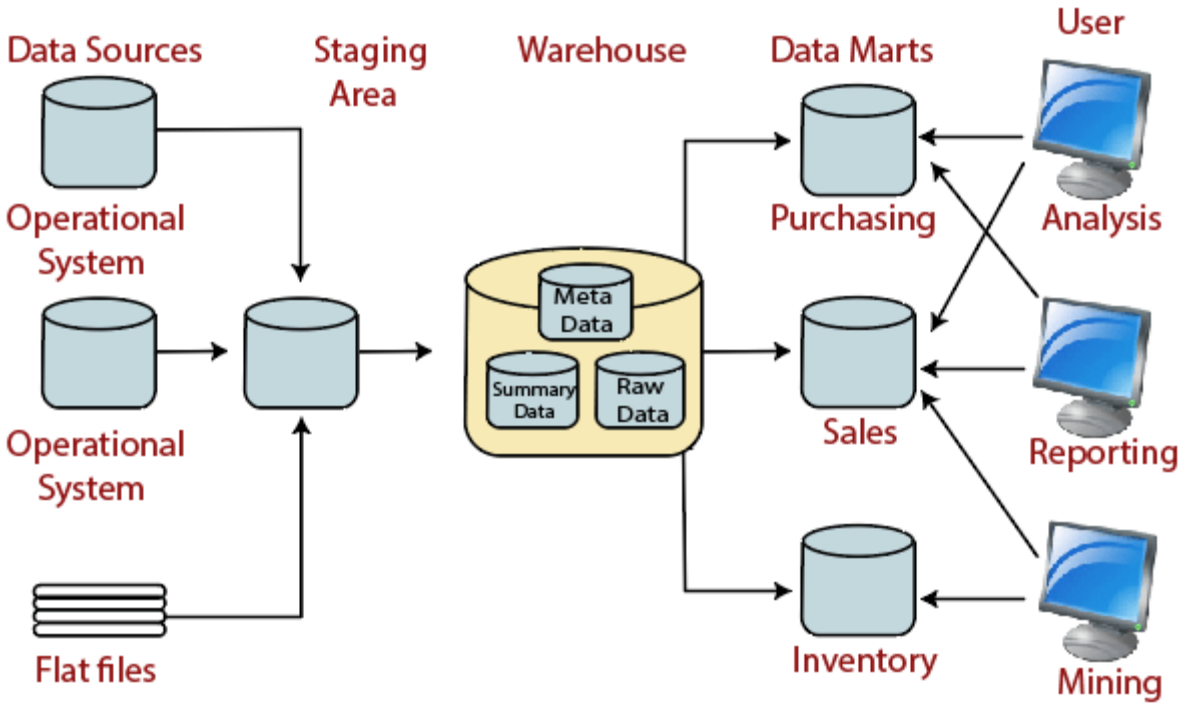


Fig.1.6. Architecture of a Data Warehouse with a Staging Area and Data Marts

Properties of Data Warehouse Architectures

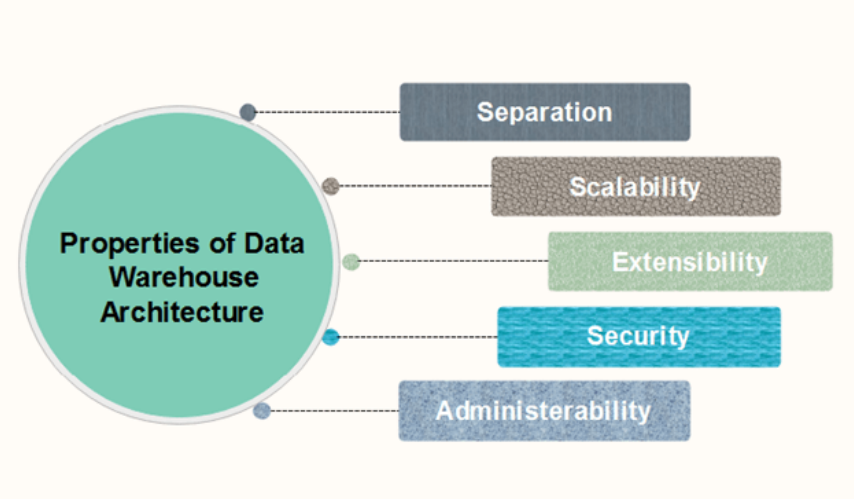


Fig.1.5 Properties of Data Warehouse Architecture

1. **Separation:** Analytical and transactional processing should be keep apart as much as possible.

2. Scalability: Hardware and software architectures should be simple to upgrade the data volume, which has to be managed and processed, and the number of user's requirements, which have to be met, progressively increase.

3. Extensibility: The architecture should be able to perform new operations and technologies without redesigning the whole system.

4. Security: Monitoring accesses are necessary because of the strategic data stored in the data warehouses.

5. Administer ability: Data Warehouse management should not be complicated.

4. Explain in detail about Three Tier Data Warehouse Architectures

With a neat sketch, explain the steps for design and construction of Data warehouses and explain with three tier architecture. [Nov 2024]

Types of Data Warehouse Architectures

- Refer Figure 1.7

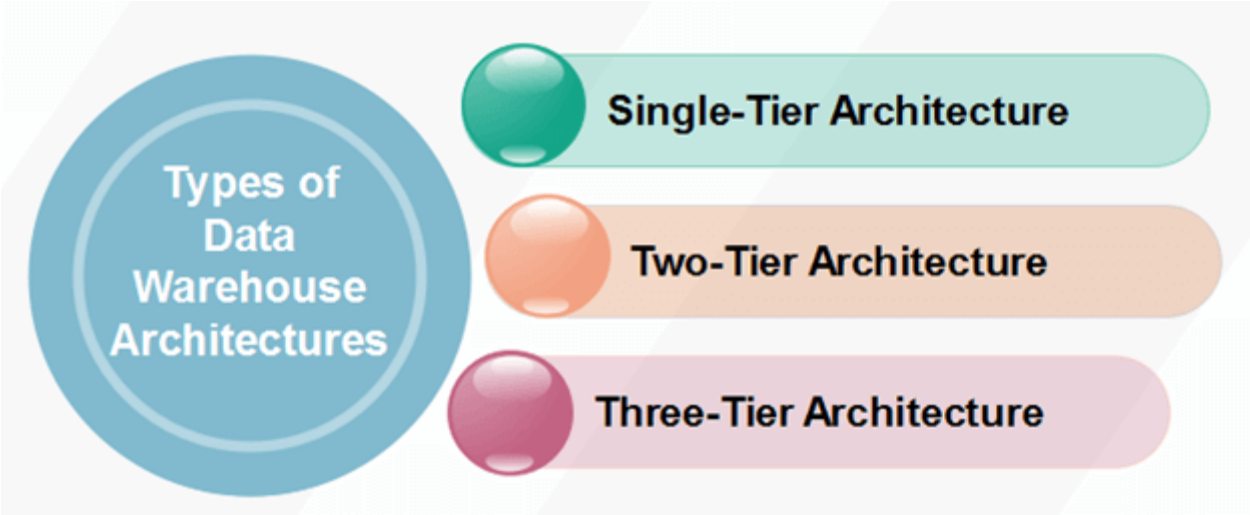


Fig.1.7 Types of Data Warehouse Architecture

Single-Tier Architecture

- Single-Tier architecture is not periodically used in practice. Its purpose is to minimize the amount of data stored to reach this goal; it removes data redundancies.

- The figure 1.8 shows the only layer physically available is the source layer. In this method, data warehouses are virtual.
- This means that the data warehouse is implemented as a multidimensional view of operational data created by specific middleware, or an intermediate processing layer.

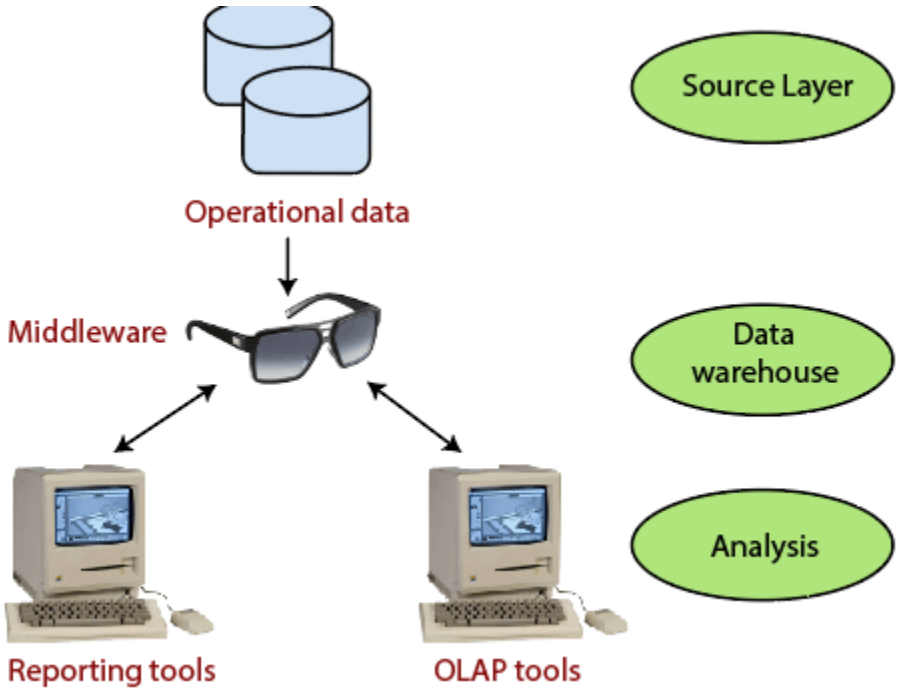


Fig.1.8 Single Tier Data Warehouse Architecture

- The vulnerability of this architecture lies in its failure to meet the requirement for separation between analytical and transactional processing.
- Analysis queries are agreed to operational data after the middleware interprets them. In this way, queries affect transactional workloads.

Two-Tier Architecture

- The requirement for separation plays an essential role in defining the two-tier architecture for a data warehouse system, as shown in figure 1.9:

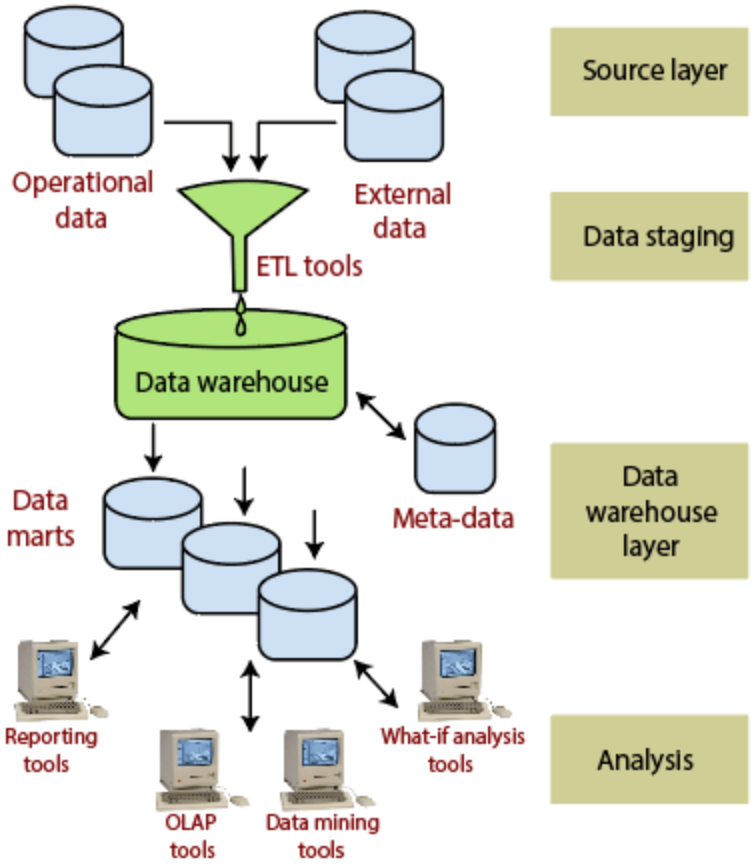


Fig.1.9 Two Tier Data Warehouse Architecture

Four subsequent data flow stages:

1. **Source layer:**
   - A data warehouse system uses a heterogeneous source of data.
   - That data is stored initially to corporate relational databases or legacy databases, or it may come from an information system outside the corporate walls.
2. **Data Staging:**
   - The data stored to the source should be extracted, cleansed to remove inconsistencies and fill gaps, and integrated to merge heterogeneous sources into one standard schema.

- The so-named **Extraction**, **Transformation**, and **Loading Tools (ETL)** can combine heterogeneous schemata, extract, transform, cleanse, validate, filter, and load source data into a data warehouse.

3. Data Warehouse layer:

- Information is saved to one logically centralized individual repository: a data warehouse.
- The data warehouses can be directly accessed, but it can also be used as a source for creating data marts, which partially replicate data warehouse contents and are designed for specific enterprise departments.
- Meta-data repositories store information on sources, access procedures, data staging, users, data mart schema, and so on.

4. Analysis:

- In this layer, integrated data is efficiently, and flexible accessed to issue reports, dynamically analyze information, and simulate hypothetical business scenarios.
- It should feature aggregate information navigators, complex query optimizers, and customer-friendly GUIs.

Three-Tier Architecture

- The three-tier architecture as in figure 1.10 consists of the **source layer** (containing multiple source system), the **reconciled layer** and the **data warehouse layer** (containing both data warehouses and data marts). The reconciled layer sits between the source data and data warehouse.
- The main **advantage** of the **reconciled layer** is that it creates a standard reference data model for a whole enterprise. At the same time, it separates the problems of source data extraction and integration from those of data warehouse population.
- In some cases, the **reconciled layer** is also directly used to accomplish better some operational tasks, such as producing daily reports that cannot be satisfactorily prepared using the corporate applications or generating data flows to feed external processes periodically to benefit from cleaning and integration.

- This architecture is especially useful for the extensive, enterprise-wide systems. A **disadvantage** of this structure is the extra file storage space used through the extra redundant reconciled layer. It also makes the analytical tools a little further away from being real-time.

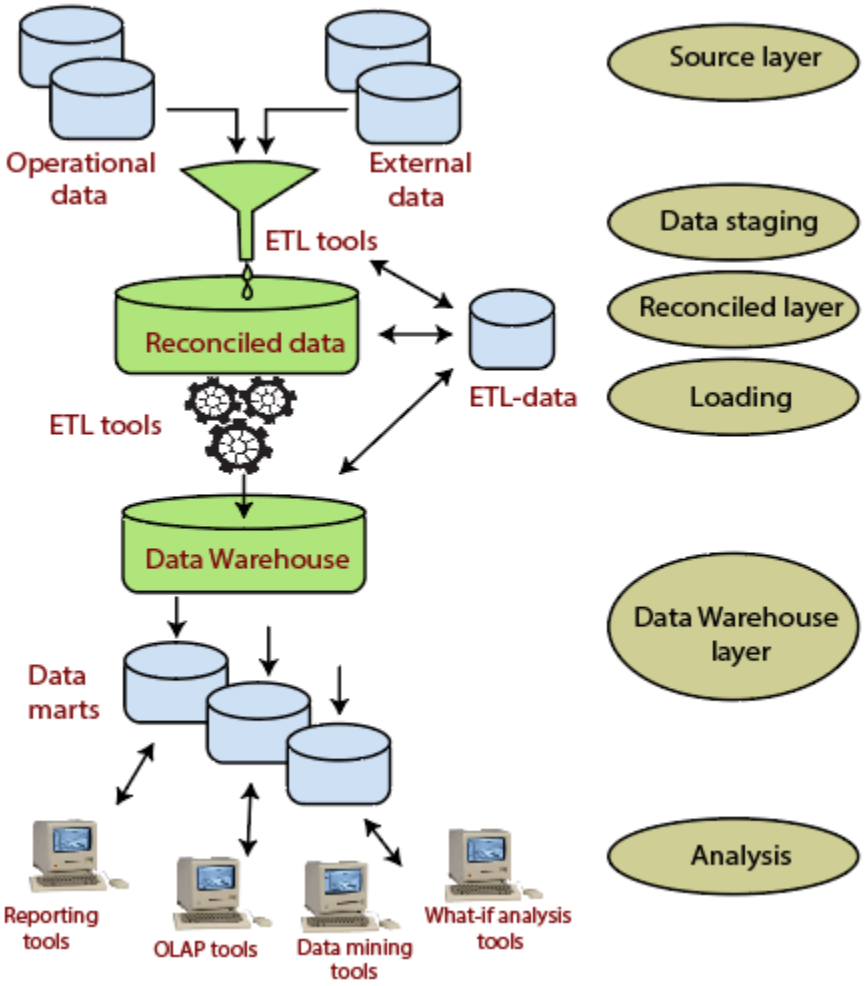


Fig.1.10. Three Tier Data Warehouse Architecture

5. Explain in detail about Three Tier Architecture of Data Warehouse. [NOV/DEC 2023]

- The Three-Tier Data Warehouse Architecture is the commonly used Data Warehouse design in order to build a Data Warehouse by including the required Data Warehouse Schema Model, the required OLAP server type, and the required front-end tools for Reporting or Analysis purposes, which as the name suggests contains three tiers such as Top tier, Bottom Tier and the Middle Tier

that are procedurally linked with one another from Bottom tier(data sources) through Middle tier(OLAP servers) to the Top tier(Front-end tools).

- Data Warehouse Architecture is the design based on which a Data Warehouse is built, to accommodate the desired type of Data Warehouse Schema, user interface application and database management system, for data organization and repository structure.
- The type of Architecture is chosen based on the requirement provided by the project team.
- Three-tier Data Warehouse Architecture is the commonly used choice, due to its detailing in the structure.

The three different tiers in figure 1.11 are termed as:

- Top-Tier
- Middle-Tier
- Bottom-Tier

Each Tier can have different components based on the prerequisites presented by the decision-makers of the project but are subject to the novelty of their respective tier.

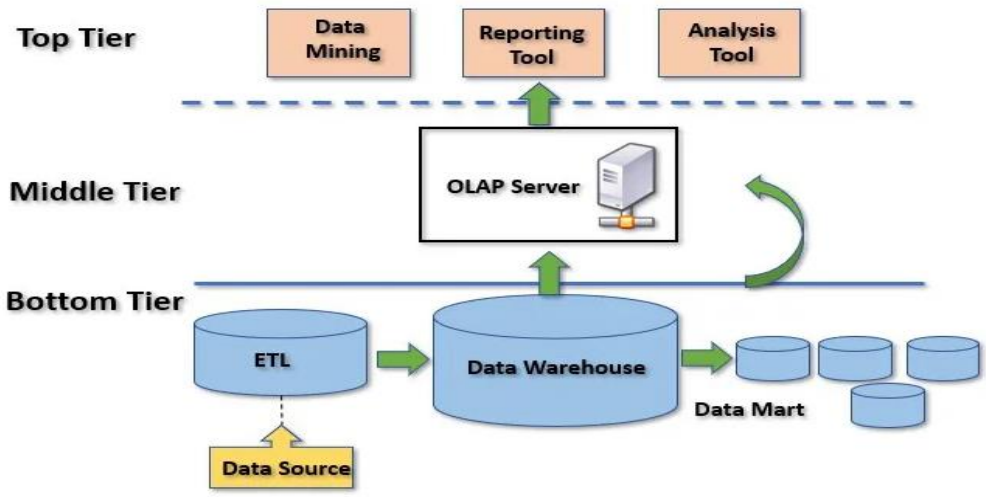


Fig.1.11 Three Tier Architecture

1. Bottom Tier

- The Bottom Tier in the three-tier architecture of a data warehouse consists of the Data Repository.
- Data Repository is the storage space for the data extracted from various data sources, which undergoes a series of activities as a part of the ETL process.
- ETL stands for Extract, Transform and Load. As a preliminary process, before the data is loaded into the repository, all the data relevant and required are identified from several sources of the system.
- These data are then cleaned up, to avoid repeating or junk data from its current storage units.
- The next step is to transform all these data into a single format of storage.
- The final step of ETL is to Load the data on the repository. Few commonly used ETL tools are:
    - Informatica
    - Microsoft SSIS
    - Snaplogic
    - Confluent
    - Apache Kafka
    - Alooma
    - Ab Initio
    - IBM Infosphere
- The storage type of the repository can be a relational database management system or a multidimensional database management system.
- A relational database system can hold simple relational data, whereas a multidimensional database system can hold data that more than one dimension.
- Whenever the Repository includes both relational and multidimensional database management systems, there exists a metadata unit.

- The metadata unit consists of all the metadata fetched from both the relational database and multidimensional database systems.
- This Metadata unit provides incoming data to the next tier, that is, the middle tier.
- From the user's standpoint, the data from the bottom tier can be accessed only with the use of SQL queries.
- The complexity of the queries depends on the type of database.
- Data from the relational database system can be retrieved using simple queries, whereas the multidimensional database system demands complex queries with multiple joins and conditional statements.

2. Middle Tier

- The Middle tier here is the tier with the OLAP servers.
- The Data Warehouse can have more than one OLAP server, and it can have more than one type of OLAP server model as well, which depends on the volume of the data to be processed and the type of data held in the bottom tier.
- There are three types of OLAP server models, such as:

ROLAP

- Relational online analytical processing is a model of online analytical processing which carries out an active multidimensional breakdown of data stored in a relational database, instead of redesigning a relational database into a multidimensional database.
- This is applied when the repository consists of only the relational database system in it.

MOLAP

- Multidimensional online analytical processing is another model of online analytical processing that catalogs and comprises of directories directly on its multidimensional database system.
- This is applied when the repository consists of only the multidimensional database system in it.

HOLAP

- Hybrid online analytical processing is a hybrid of both relational and multidimensional online analytical processing models.
- When the repository contains both the relational database management system and the multidimensional database management system, HOLAP is the best solution for a smooth functional flow between the database systems. HOLAP allows storing data in both the relational and the multidimensional formats.
- The Middle Tier acts as an intermediary component between the top tier and the data repository, that is, the top tier and the bottom tier respectively.
- From the user's standpoint, the middle tier gives an idea about the conceptual outlook of the database.

3. Top Tier

- The Top Tier is a front-end layer, that is, the user interface that allows the user to connect with the database systems.
- This user interface is usually a tool or an API call, which is used to fetch the required data for Reporting, Analysis, and Data Mining purposes.
- The type of tool depends purely on the form of outcome expected. It could be a Reporting tool, an Analysis tool, a Query tool or a Data mining tool.

The few commonly used Top Tier tools

- IBM Cognos
- Microsoft BI Platform
- SAP Business Objects Web
- Pentaho
- Crystal Reports
- SAP BW
- SAS Business Intelligence

5. Explain in detail about Autonomous Data Warehouse.

Discuss on autonomous data warehouse. [Nov 2024]

Autonomous Data Warehouse(ADW)

- Oracle Autonomous Data Warehouse is the world's first and only autonomous database optimized for analytic workloads, including data marts, data warehouses, data lakes, and data lake houses.
- With Autonomous Data Warehouse, data scientists, business analysts, and nonexperts can rapidly, easily, and cost-effectively discover business insights using data of any size and type.
- Built for the cloud and optimized using Oracle Exadata, Autonomous Data Warehouse benefits from faster performance and, according to an IDC report, lowers operational costs by an average of 63%.

Autonomous Data warehouse Vs Snowflake

- **Five key areas** that organizations assign top
- priority – cost governance, real-time DW workloads, data integrity, ML integration, and deployment flexibility – in comparison to the Snowflake Cloud Data Platform.

First

- In terms of cost governance Oracle ADW provides granular compute sizing, whereas Snowflake requires customers to double in size and cost for every step up to meet their expanding compute size requirements.
- In parallel, Oracle ADW auto-scales instantaneously online and uses governed storage, however I see Snowflake lacking the oversight mechanisms needed to control storage costs.

Second

- Oracle ADW as providing a definitive edge in the processing of real-time and operational DW workloads.
- For example, Oracle ADW supports large numbers of concurrent queries, whereas the Snowflake platform defaults to eight concurrent queries per clusters.

- Oracle ADW furnishes indexes for rapid lookups – a capability that Snowflake completely lacks.
- In addition, Oracle ADW assures efficient updates, while Snowflake offers only an append-only architecture that impedes real-time updates.

Third

- Oracle ADW ensures data integrity by applying enforced unique/primary key, foreign key and check constraints to ensure that the data is correct by preventing simple mistakes like duplicate records.
- In contrast, the Snowflake Data Cloud Platform does not enforce such meaningful constraints. As a result, Snowflake customers do not have full assurances about the integrity and correctness of their data.

Fourth

- Oracle ADW supports and integrates a wide array of built-in, self-service ML algorithms.
- With Snowflake, customers must license third-party ML tools, install, manage, and learn how to use the tools, delaying time to insight and increasing overall costs.
- Oracle ADW also includes APEX, a popular no-code/low-code environment that significantly accelerates application development.
- I do not see Snowflake having equivalent of Oracle APEX.

Fifth

- Oracle ADW is available in OCI and on-premises in Oracle Exadata Cloud@Customer and Dedicated Region Cloud@Customer with complete architectural identicality across deployment models.
- Moving Oracle Database data does not require transformations or re-formatting, Further, Oracle's Cloud@Customer options enable customers to meet data sovereignty and regulatory requirements. Snowflake only runs in the public cloud and from my perspective does not address the requirements for data sovereignty and regulatory compliance.

- Overall, DB/DW decision makers must prioritize these selection criteria in the evaluation process and direct comparison of the Oracle ADW and Snowflake propositions.
- Organizations need to consider the full spectrum of DB and DW requirements or risk selecting a solution that curtails their ability to perform analytics in real-time, lacks fine-grained elastic scaling, does not provide full data integrity, has insufficient tools, and limits deployment flexibility in advancing their cloud DW journey.
- By separating hype from the underlying realities of DW optimization and administration, organizations can avoid spinning their wheels on subpar outcomes and getting ensnared in data warehousing snowdrifts.

Difference between autonomous and Snowflake

- With Cloud Customer solutions, Autonomous Data Warehouse can be deployed as a fully managed cloud service in a customer's own data centers to address data residency and security requirements. Snowflake only runs in the public cloud.

6. Explain in detail about Modern Data Warehouse.

Modern Data Warehouse

- A Modern Data Warehouse is a cloud-based solution that helps organizations gather, store, and process data to help you make intelligent decisions.

Uses

- A variety of organizations can use a modern data warehouse to improve business processes such as finances, human resources, and operations.
- Modern Data Warehouse Pyramid

There are five different components of a Modern Data Warehouse.

Level 1: Data Acquisition

Data acquisition can come from a variety of sources such as:

- IoT devices
- Social media posts

- YouTube videos
- Website content
- Customer data
- Enterprise Resource Planning
- Legacy data stores

Level 2: Data Engineering

- Once you acquired it, you need to upload it into the data warehouse. Data engineering uses pipelines and ETL (extract, transform, load) tools.
- Using these different tools, you can upload that information to a data warehouse similar to a factory.
- Data engineering is similar to a truck bringing raw materials into a factory.

Level 3: Data Management Governance

- Once the data comes into the factory, you need someone to evaluate the quality of the data. You then need to steward that data because security and privacy must be considered.
- Data governance helps ensure the quality of the info by stewarding, prepping, and cleaning the data to ensure it is ready for analysis.

Level 4: Reporting and Business Intelligence

- Once you prep and clean the data, you can start using factory analysis to take that raw material(data) and turn it into a finished good (business intelligence).
- For our purposes, we will use Microsoft Power BIto help you visualize the information by using advanced analytics, KPIs, and workflow automation.
- When you are finished, you can see exactly what's going on with your data.

Level 5: Data Science

- Modern Data Warehouse is about more than seeing the information; it's about using the data to make smarter decisions.

  There are several different programs helps leverage the data benefit, including:

  - AI

- Deep learning
- Machine learning
- Statistical modeling
- Natural language processing (NLP)

- Modern Data warehouse comprised of multiple programs impervious to User. Polyglot persistence encourages the most suitable data storage technology based on data.
- This "best-fit engineering" aligns multi-structure data into data lakes and considers NoSQL solutions for JSON formats.
- Pursuing a polyglot persistence data strategy benefits from virtualization and takes advantage of the different infrastructure.
- Modern DW requires Petabytes of storage and more optimized techniques to run complex analytic queries.
- The traditional methods are relatively less efficient and not cost-effective to fit into the modern day Data Warehousing needs.
- There are tons of Cloud solutions to build data warehouses performance optimized, inexpensive, and support parallel query execution.
    - Incorporate Hadoop, traditional data warehouse, and other data stores.
    - Includes multiple repositories may reside in different locations.
    - Include Data from mobile devices, sensors, cloud and the Internet of Things.
    - Includes structure/semi-structured/unstructured, raw data.
    - Inexpensive commodity hardware in cluster mode.
- Data Warehousing is processing for gathering and handling data from various sources to provide essential business insights. Source: Data Warehouse Modernization

Working architecture of Modern Data Warehouse

- The working architecture of real-time Modern Data Warehouse is mentioned below:

Multiple Parallel Processing (MPP) Architectures

- MPP architecture enables a mighty scale and Distributed Computing.
- Resources add for a linear scale-out to the largest Data Warehousing projects.
- Multiple parallel processing architecture uses a "shared-nothing". There are numerous physical nodes, each runs its instance. This results from performance many times faster than traditional architectures.

Multi-Structured Data

- Define Big Data & Analytics Infrastructure for multiple storage data with a polyglot persistence strategy.
- Integrate portions of the data into the Data Warehouse.
- Federated query access.

Lambda Architecture

- In lambda, architecture defines three layers -
- Speed Layer - Low latency data.
- Batch Layer - Raw Data processing to support complex analysis.
- Serving Layer - Response to queries.

Hybrid Architecture

- Scale up MPP compute nodes during -
- Peak ETL data loads.
- High query volumes.
- Utilize existing On-Premises data structures.
- Use Cloud services for Advanced Analytics.
- A mini Data Warehouse design that shows the contents to be needed only to the client-side, i.e. it holds the overview of the data. Click to explore about, Data Mart a Subset of The Data Warehouse

Importance of Modern Data Warehouse

It solves the problems for various businesses such as:

- **Data Lakes -** Instead of storing in hierarchical files and folders, as traditional data warehouse do, a data lake is the repository that holds a vast amount of raw data in its native format until needed.
- **Data Divided Across Organizations** - Modern Data Warehousing allows for quicker information Assortment and Analysis across organizations and divisions. It keeps the Agility model and promotes more alignment and sooner effect.
- **IoT Streaming Data** - The Internet of Things has completely transformed the scenario, units, etc. share and stock data across multiple devices.

Business Challenges

- Reduce the cost to store and manage data growth.
- Business demand to analyze new data sources requires investment in technologies to process all data formats.
- Current Data Warehouses are good for Multidimensional Analytics but not suited for Image, Video or other new types of analytics.
- The core process used to manage, centralize, and organize data according to business marketing and operations. Source: Master Data Management

Adoption of Modern Data Warehouse

The steps to adopt it are described below:

- Growing an Existing DW Environment
- Internal to the Data Warehouse
- Data modeling strategies
- Partitioning
- Clustered column store index
- In-memory structure

- MPP

Augment the Data Warehouse

- Complementary Data Storage & Analytical solutions.
- Cloud & Hybrid solutions.
- Data Virtualization/ Virtual DW.

Features of Modern Data Warehouse

- Variety of subject areas & data sources for analysis with the capability to handle the large volume of data.
- Expansion beyond a single relational DW/Data Mart structure to include Data Lake.
- Logical design across multi-platform architecture balancing performance & scalability.
- Data virtualization in addition to Data Integration.
- Support for all type & levels of users.
- Flexible deployment decoupled from the tool used for development.
- Governance model to support security and trust, and Master Data Management.
- Support for promoting the self-service solution to the corporate environment.
- Ability to facilitate Real-Time analysis of high-velocity data.
- Support for Advanced Analytics.
- Agile Delivery approach with the fast delivery cycle.
- Hybrid Integration with Cloud services.
- APIs for downstream access to data.
- Some DW automation to improve speed, consistency, business terminology.
- An analytics sandbox or workbench area to facilitate agility within a BI environment.

- Support for self-service BI to augment corporate BI; Data discovery, Data Exploration, Self-service Data preparation.
- The Concept of Database designing is key, whereas the SQL queries part is relatively very simple. Click to explore about our, Data Warehouse Database Design Architecture

Benefits of Modern Data Warehouse

- Rapid integration of data into the environment.
- Improved efficiency in integration reducing time, cost and efforts.
- Opportunity to enable innovative new data models.
- Potential for new insights into the data that provide Preventive analysis and Predictive Analysis.
- Ability to have more extensive datasets for analysis as the data collected and stored continues to grow exponentially.
- Cost advantages of Open source software & Commodity hardware.

7. Describe the technologies used to improve the performance in data warehouse environment. Mention a few alternate technologies also. [NOV/DEC 2023]

Improving performance in a data warehouse environment involves implementing various technologies and strategies to optimize data storage, processing, and query performance. Here are some key technologies commonly used:

1. **Columnar Storage:**
    - Data warehouses often use columnar storage formats (e.g., Parquet, ORC) where data is stored by columns rather than rows. This can significantly improve query performance for analytical workloads because it allows for efficient compression and retrieval of specific columns relevant to a query.

2. **Compression Techniques:**

    - Compression algorithms (e.g., Snappy, LZ4) are used to reduce the storage footprint of data while maintaining fast query performance. Compressed data requires less I/O bandwidth and storage space, which speeds up data access and retrieval.

3. **Partitioning:**

    - Partitioning divides large tables into smaller, more manageable parts based on certain criteria (e.g., date ranges, key ranges). This technique improves query performance by allowing the query engine to scan only relevant partitions instead of the entire dataset.

4. **Indexing:**

    - While not as commonly used in data warehouses compared to OLTP databases, indexing on frequently queried columns can improve query performance by facilitating faster data retrieval.

5. **In-Memory Processing:**

    - Some data warehouse solutions utilize in-memory processing (e.g., Apache Spark with in-memory caching) to reduce query latency by keeping frequently accessed data in memory. This approach accelerates query performance by avoiding disk I/O operations.

6. **Parallel Processing:**

    - Data warehouses leverage parallel processing techniques to distribute query workload across multiple nodes or cores. This allows queries to be executed in parallel, speeding up overall query response times.

7. **Query Optimization:**

    - Advanced query optimization techniques, such as cost-based optimization and query rewriting, are used to generate efficient query

execution plans. These techniques analyze query patterns and data statistics to optimize query performance.

8. **Data Partitioning and Sharding:**

    - Partitioning and sharding techniques distribute data across multiple nodes or servers to parallelize data access and processing. This horizontal scaling approach helps handle large volumes of data and improve overall system performance.

Alternate Technologies:

- **Data Lake:** While not a direct performance improvement technology for data warehouses, a data lake (using technologies like Apache Hadoop or AWS S3) complements data warehouses by storing raw, unstructured or semi-structured data. It serves as a scalable repository for diverse data types that can be processed and integrated into the data warehouse for analytics.
- **Apache Hadoop:** Beyond just a data lake, Hadoop can also support distributed processing frameworks like Apache Spark or Hive, which can be used for large-scale data processing and analytics, potentially integrating with or augmenting traditional data warehouse solutions.
- **NoSQL Databases:** For specific use cases where semi-structured or unstructured data needs to be processed at scale, NoSQL databases like MongoDB or Cassandra provide flexible data models and horizontal scalability, although they serve different purposes than traditional SQL-based data warehouses.
- **Cloud-based Data Warehouses:** Services like Amazon Redshift, Google Big Query, or Snowflake provide scalable, managed data warehousing solutions in the cloud, leveraging distributed computing resources and optimized storage architectures to improve performance and scalability.

These technologies and strategies contribute to optimizing performance in data warehouse environments by addressing data storage, retrieval, and processing challenges inherent in large-scale analytics and reporting tasks.

8. Describe the three layers of snowflake architecture. [Nov 2024]

What is the Snowflake Data Warehouse?

- Snowflake Data Warehouse is a cloud-based data warehousing platform that is designed for scalable and efficient storage and analysis of datasets. It contains a unique architecture with separate storage and computing resources.
- Snowflake supports multi-cluster, multi-cloud deployment which enables us to choose a preferred cloud provider.
- It offers robust security measures, including encryption and role-based access control. With features like zero-copy cloning and data sharing, Snowflake facilitates agile data management and collaboration.

Snowflake's Architecture

Snowflake's architecture in figure 1.12 mainly consists of three layers.

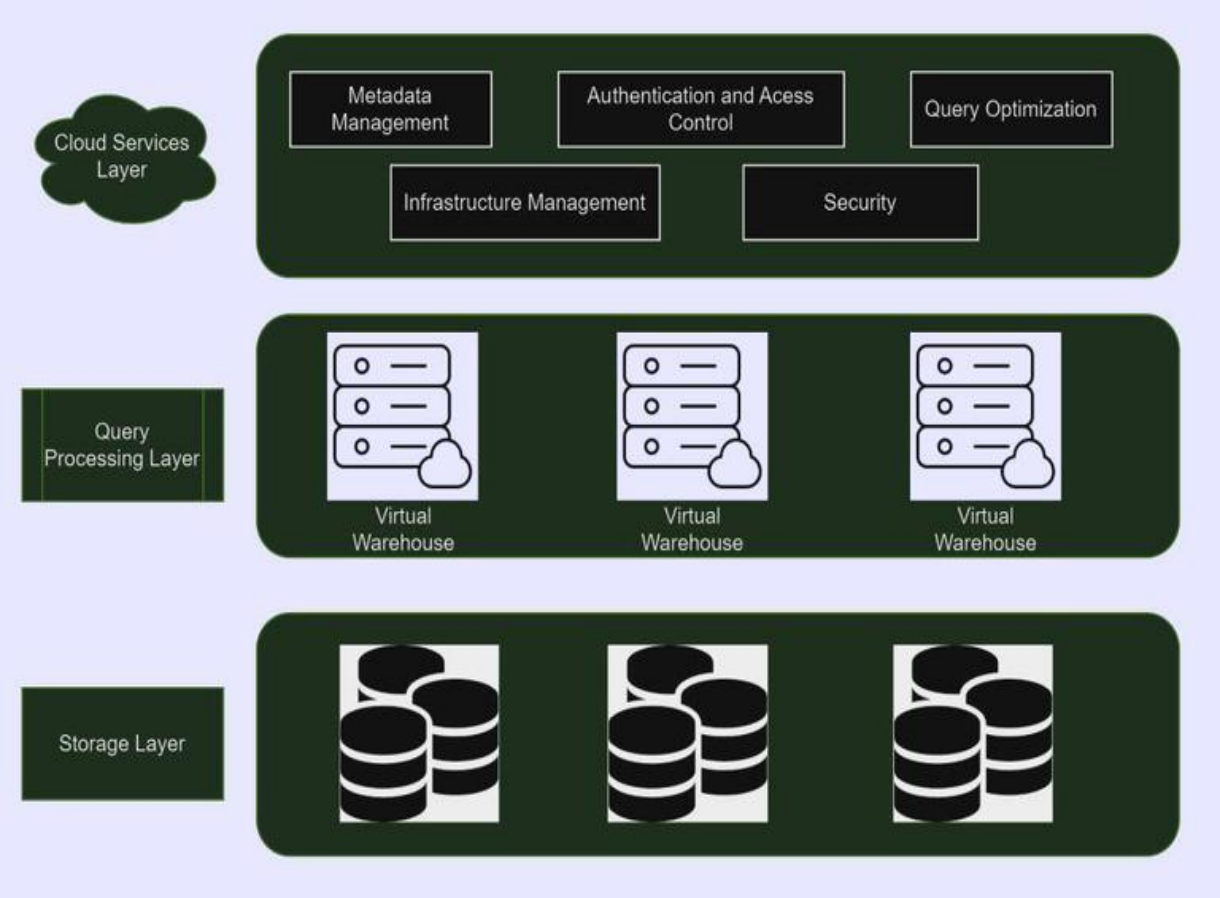


Fig.1.12 Snowflake Architecture

1. Storage Layer

The Storage layer in snowflake architecture is responsible for managing and storing data in an effective manner. The functionalities that were supported by the storage layer are:

Elasticity: Snowflake's storage layer is elastic, allowing organizations to scale their storage needs independent of compute resources. It ensures to handle various data volumes without affecting performance.

Cloud Based Object Storage: Snowflake uses cloud based object storage to store data. This separation of storage and compute allows for cost-effective and scalable data storage.

Data Clustering: Snowflake organizes data into micro partitions within the object storage, and these micro partitions are clustered based on metadata. This clustering enhances query performance by minimizing the amount of data that needs to be scanned.

Zero Copy Cloning: Snowflake enables efficient data cloning through zero-copy cloning technology. This feature allows users to create a copy of a dataset instantly without duplicating the actual data, saving both time and storage costs.

2. Query Processing Layer

- The SQL query execution is handled by Snowflake's Query Processing Layer, which dynamically optimizes and parallelizes queries over several compute clusters. It ensures great performance and scalability by decoupling computation and storage, allowing for on-demand resource allocation based on query complexity and workload.

Functionalities of Query Processing Layer are:

Automatic Query Processing: Snowflake's Query Processing Layer optimizes SQL queries automatically, modifying execution plans based on underlying data distribution and query complexity to ensure efficient processing.

Parallel Execution across Clusters: Query execution is performed in parallel across many compute clusters, leveraging Snowflake's multi-cluster architecture to achieve high concurrency and faster results for complex analytical workloads.

On Demand Resource Allocation: Depending on the complexity and number of queries, the Query Processing Layer dynamically distributes computational resources as needed. This on-demand resource distribution provides peak performance and cost efficiency.

Compute and Storage Separation: Snowflake's architecture separates computing and storage, allowing the Query Processing Layer to expand compute resources independently. This separation improves flexibility by allowing enterprises to change computer power without affecting stored data, so optimizing both performance and prices.

3. Cloud Services Layer

- In Snowflake's architecture, the Cloud Services Layer serves as the control plane, managing information, security, and user access. It serves as a centralized platform for administration, authentication, and activity coordination across the data warehouse. This layer ensures that users and the underlying computation and storage resources in a cloud environment interact seamlessly.

The functionalities of Cloud Services Layer are:

Metadata Management: Snowflake's metadata management involves storing comprehensive information about data objects, structures, and statistics, facilitating efficient query optimization. This metadata layer is crucial for dynamically organizing and processing data within the cloud-based data warehousing platform.

Authentication and Access Control: Snowflake employs robust authentication methods, including multi-factor authentication, to secure user access. Access control is granular, with role-based permissions and policies ensuring fine-tuned control over data and system resources.

Query Optimization: Snowflake's query optimization dynamically adjusts execution plans based on data distribution and complexity, ensuring efficient processing of SQL queries. It leverages a multi-cluster, parallel processing architecture for faster and scalable query performance.

Infrastructure Management: Snowflake automates infrastructure management by dynamically allocating and deallocating computing resources based on workload demands, ensuring optimal performance and cost efficiency. This approach simplifies operations for users by abstracting the complexities of underlying cloud infrastructure.

Security: Snowflake prioritizes security with end-to-end encryption, role-based access controls, and features like data masking, ensuring comprehensive protection of sensitive data within the cloud-based data warehousing platform. Security measures are integrated at every level, safeguarding against unauthorized access and data breaches.

9. Suppose that a data warehouse consists of four dimensions customer, product, salesperson, and sales time and the three measure sales Amount (in rupees), VAT (in rupees) and payment type (in rupees). Draw the different classes of schemas that are popularly used for modeling data warehouses and explain it. (8)

- Schema is a logical description of the entire database. It includes the name and description of records of all record types including all associated data-items and aggregates.
- Much like a database, a data warehouse also requires to maintain a schema. A database uses relational model, while a data warehouse uses Star, Snowflake, and Fact Constellation schema.

Star Schema

- Each dimension in a star schema is represented with only one-dimension table.
- This dimension table contains the set of attributes.

The following figure 1.13 shows the sales data of a company with respect to the four dimensions, namely time, item, branch, and location.

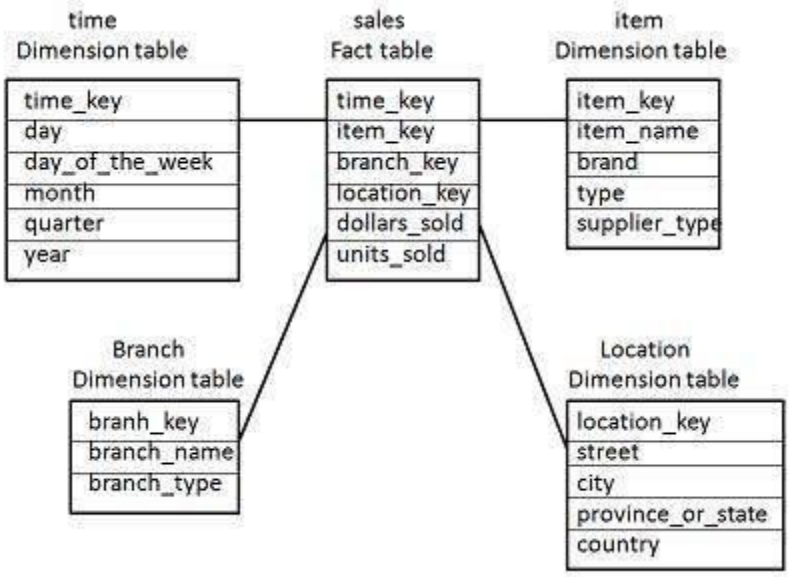


Fig.1.13 Star Schema

There is a fact table at the center. It contains the keys to each of four dimensions.

The fact table also contains the attributes, namely dollars sold and units sold.

Note –

- Each dimension has only one dimension table and each table holds a set of attributes. For example, the location dimension table contains the attribute set {location_key, street, city, province_or_state,country}.
- This constraint may cause data redundancy.
- For example, "Vancouver" and "Victoria" both the cities are in the Canadian province of British Columbia.
- The entries for such cities may cause data redundancy along the attributes province_or_state and country.

Snowflake Schema

- Some dimension tables in the Snowflake schema are normalized.
- The normalization splits up the data into additional tables as shown in figure 1.14.

- Unlike Star schema, the dimensions table in a snowflake schema are normalized. For example, the item dimension table in star schema is normalized and split into two dimension tables, namely item and supplier table.

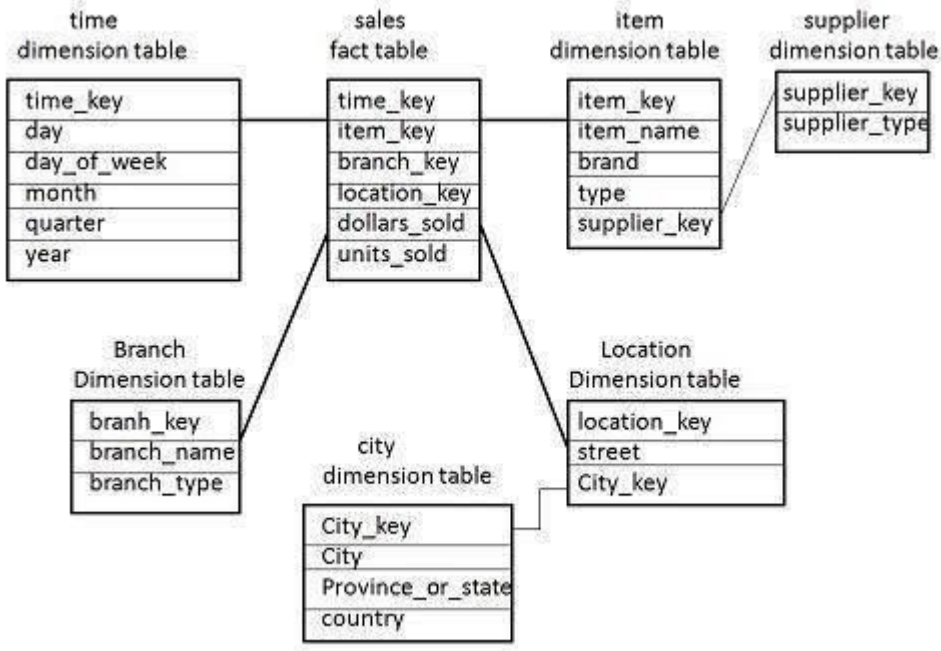


Fig.1.14 Snowflake Schema

- Now the item dimension table contains the attributes item_key, item_name, type, brand, and supplier-key.
- The supplier key is linked to the supplier dimension table. The supplier dimension table contains the attributes supplier_key and supplier_type.
- **Note** – Due to normalization in the Snowflake schema, the redundancy is reduced and therefore, it becomes easy to maintain and the save storage space.

Fact Constellation Schema

- A fact constellation has multiple fact tables. It is also known as galaxy schema.

  The following figure 1.15 shows two fact tables, namely sales and shipping.

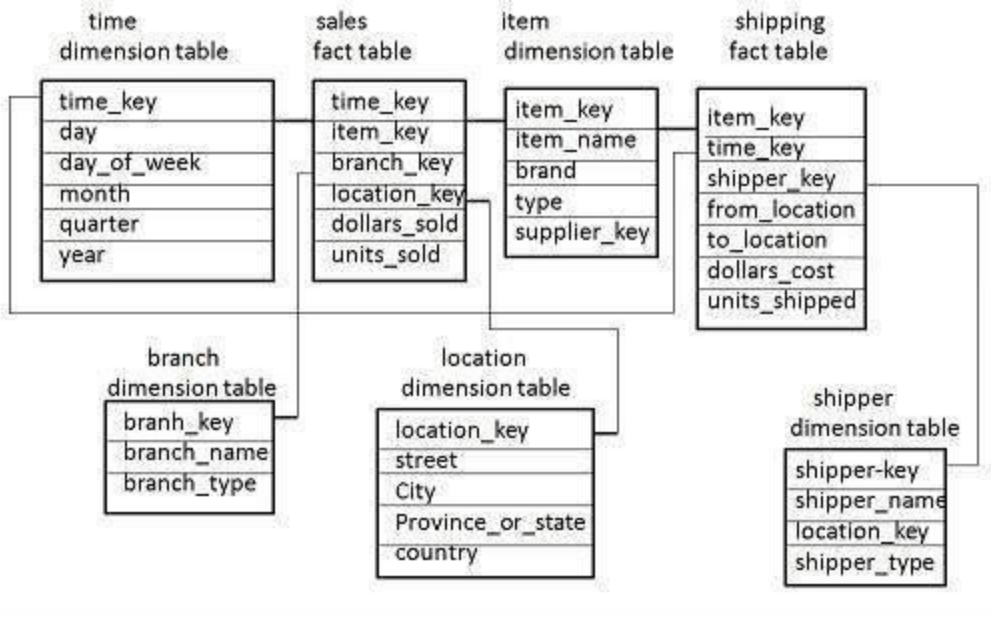


Fig.1.15 Fact Constellation Schema

- The sales fact table is same as that in the star schema.
- The shipping fact table has the five dimensions, namely item_key, time_key, shipper_key, from_location, to_location.
- The shipping fact table also contains two measures, namely dollars sold and units sold.
- It is also possible to share dimension tables between fact tables. For example, time, item, and location dimension tables are shared between the sales and shipping fact table.

Schema Definition

- Multidimensional schema is defined using Data Mining Query Language (DMQL). The two primitives, cube definition and dimension definition, can be used for defining the data warehouses and data marts.

Syntax for Cube Definition

define cube < cube_name > [< dimension-list > }: < measure_list >

Syntax for Dimension Definition

define dimension < dimension_name > as (< attribute_or_dimension_list >)

Star Schema Definition

The star schema that we have discussed can be defined using Data Mining Query Language (DMQL) as follows –

define cube sales star [time, item, branch, location]:
dollars sold = sum(sales in dollars), units sold = count(*)
define dimension time as (time key, day, day of week, month, quarter, year)
define dimension item as (item key, item name, brand, type, supplier type)
define dimension branch as (branch key, branch name, branch type)
define dimension location as (location key, street, city, province or state, country)

Snowflake Schema Definition

Snowflake schema can be defined using DMQL as follows –

define cube sales snowflake [time, item, branch, location]:
dollars sold = sum(sales in dollars), units sold = count(*)
define dimension time as (time key, day, day of week, month, quarter, year)
define dimension item as (item key, item name, brand, type, supplier (supplier key, supplier type))
define dimension branch as (branch key, branch name, branch type)
define dimension location as (location key, street, city (city key, city, province or state, country))

Fact Constellation Schema Definition

Fact constellation schema can be defined using DMQL as follows –

define cube sales [time, item, branch, location]:
dollars sold = sum(sales in dollars), units sold = count(*)
define dimension time as (time key, day, day of week, month, quarter, year)
define dimension item as (item key, item name, brand, type, supplier type)
define dimension branch as (branch key, branch name, branch type)

define dimension location as (location key, street, city, province or state,country)
define cube shipping [time, item, shipper, from location, to location]:
dollars cost = sum(cost in dollars), units shipped = count(*)
define dimension time as time in cube sales
define dimension item as item in cube sales
define dimension shipper as (shipper key, shipper name, location as location in cube sales, shipper type)
define dimension from location as location in cube sales
define dimension to location as location in cube sales

UNIT II ETL AND OLAP TECHNOLOGY 6

What is ETL – ETL Vs ELT – Types of Data warehouses - Data warehouse Design and Modeling - Delivery Process - Online Analytical Processing (OLAP) - Characteristics of OLAP – Online Transaction Processing (OLTP) Vs OLAP - OLAP operations- Types of OLAP- ROLAP Vs MOLAP Vs HOLAP.

PART A

1. What is meant ETL? [Nov 2024]

- ETL is a process in Data Warehousing and it stands for Extract, Transform and Load.
- It is a process in which an ETL tool extracts the data from various data source systems, transforms it in the staging area, and then finally, loads it into the Data Warehouse system.

2. What are characteristics of OLAP?

Outline the characteristics of OLAP. [Nov 2024]

- OLAP provides for distinguishing between zero values and missing values so that aggregates are computed correctly.
- OLAP system should ignore all missing values and compute correct aggregate values. OLAP facilitate interactive query and complex analysis for the users.

3. What is data warehouse Modelling?

- Data warehouse modeling is the process of designing and organizing your data models within your data warehouse platform.
- The design and organization process consists of setting up the appropriate databases and schemas so that the data can be transformed and then stored in a way that makes sense to the end user.

4. What are the various steps of data modeling?

- Choose a Data Source.
- Selection of Data Sets.
- Selection of Attributes, Columns and Metrics.

- Relationship Tool.
- Hierarchies.
- Roles & Permissions.
- Finalization and Deployment.

5. What is data modelling in ETL?

- ETL data modeling is a process that generates theoretical representations of various data objects, figures, and rules on how to store them in a data warehouse.
- It is a critical part of the ETL process and, at the same time, has no purpose without ETL.

6. What is delivery method in data warehouse?

- The delivery method is a variant of the joint application development approach adopted for the delivery of a data warehouse. We have staged the data warehouse delivery process to minimize risks.

7. What are the three 3 process used in a data warehouse?

- Process Flow in Data Warehouse
- Extract and load the data. Cleaning and transforming the data. Backup and archive the data.

8. What is the ETL loading process?

- Extract, transform, and load (ETL) is the process of combining data from multiple sources into a large, central repository called a data warehouse.
- ETL uses a set of business rules to clean and organize raw data and prepare it for storage, data analytics, and machine learning (ML).

9. What is OLAP and OLTP?

- The primary purpose of online analytical processing (OLAP) is to analyze aggregated data, while the primary purpose of online transaction processing (OLTP) is to process database transactions.
- You use OLAP systems to generate reports, perform complex data analysis, and identify trends.

10. What are the functions of OLAP?

- Oracle OLAP functions extend the functionality of standard SQL analytic functions by providing capabilities to compute aggregate values based on a group of rows.

- You can apply the OLAP functions to logically partitioned sets of results within the scope of a single query expression.

11. What is OLAP and where it is used?

- Online analytical processing (OLAP) is a technology that organizes large business databases and supports complex analysis.
- It can be used to perform complex analytical queries without negatively affecting transactional systems.

12. What is OLAP vs OLTP vs data warehouse?

- Data Warehouse is the example of OLAP system. OLTP stands for On-Line Transactional processing.
- It is used for maintaining the online transaction and record integrity in multiple access environments.
- OLTP is a system that manages very large number of short online transactions for example, ATM.

13. What is OLAP and its functions?

- OLAP (online analytical processing) is a computing method that enables users to easily and selectively extract and query data in order to analyze it from different points of view.

14. What are the advantages of OLAP?

- Analyze reports at the "speed of thought", and manipulate them in real time.
- Share Intelligent Cube data securely.
- Schedule Intelligent Cube execution and maintenance.

15. What are the limitations of OLAP?

- Some of the disadvantages of OLAP are pre-modeling, which is a must, great dependence on IT, poor computation capability, slow in reacting, short of Interactive analysis ability, abstract model, and great potential risk.

16. When should OLAP be used?

- OLAP is ideal for data mining, business intelligence and complex analytical calculations, as well as business reporting functions like financial analysis, budgeting and sales forecasting.
- The core of most OLAP databases is the OLAP cube, which allows you to quickly query, report on and analyze multidimensional data.

17. What is a pivot in OLAP?

- This term refers to a new view of data available within a Slice of a multidimensional OLAP Cube.
- As an example: a financial analyst might want to view or "pivot" data in various ways, such as displaying all the cities down the page and all the products across a page.

18. What are the operations of OLAP?

- It allows us to gain insight into the data through special data structures known as OLAP cubes and operations such as drill-down, roll-up, slicing, dicing, and pivot.

19. What is OLTP operations in data mining?

- OLTP (Online Transactional Processing) is a type of data processing that executes transaction-focused tasks.
- It involves inserting, deleting, or updating small quantities of database data. It is often used for financial transactions, order entry, retail sales and CRM.

20. How many types of OLAP are there?

- ROLAP stands for Relational OLAP, an application based on relational DBMSs.
- MOLAP stands for Multidimensional OLAP, an application based on multidimensional DBMSs.
- HOLAP stands for Hybrid OLAP, an application using both relational and multidimensional techniques.

21. What is the difference between MOLAP and ROLAP servers?

- MOLAP is used for limited data volumes and in this data is stored in multidimensional array.
- In MOLAP, Dynamic multidimensional view of data is created. The main difference between ROLAP and MOLAP is that, In ROLAP, Data is fetched from data-warehouse.
- On the other hand, in MOLAP, Data is fetched from MDDBs database.

22. Is Snowflake a MOLAP or ROLAP?

- Snowflake uses OLAP as a foundational part of its database schema and acts as a single, governed, and immediately queryable source for your data.

23. What is ETL and ELT difference?

- The ETL process transforms data on a secondary processing server.

- In contrast, the ELT process loads raw data directly into the target data warehouse. Once there, you can transform the data whenever you need it.

24. Which is better ETL or ELT?

- ETL is most appropriate for processing smaller, relational data sets which require complex transformations and have been predetermined as being relevant to the analysis goals.
- ELT can handle any size or type of data and is well suited for processing both structured and unstructured big data.

25. Why is ELT preferred over ETL?

- The primary advantage of ELT over ETL relates to flexibility and ease of storing new, unstructured data.
- With ELT, you can save any type of information—even if you don't have the time or ability to transform and structure it first—providing immediate access to all of your information whenever you want it.

26. Is ELT replacing ETL?

- Whether ELT replaces ETL depends on the use case.
- While ELT is adopted by businesses that work with big data, ETL is still the method of choice for businesses that process data from on-premises to the cloud.
- It is obvious that data is expanding and pervasive.

27. What is an example of ELT?

- An example of this is the stock market, which generates large amounts of data that is consumed in real-time.
- In scenarios such as this, ELT is the solution of choice because the transformation occurs after the data reaches its destination.

28. Compare OLTP and OLAP Systems. [NOV/DEC 2023]

- **Data Usage:** OLTP manages current, operational data for transactional processing, while OLAP analyzes historical, aggregated data for decision support.
- **Schema:** OLTP uses normalized schemas for data integrity, while OLAP uses denormalized or dimensional schemas for efficient querying and analysis.
- **Query Types:** OLTP handles simple, real-time transactional queries, while OLAP supports complex, analytical queries.

- **Performance Focus:** OLTP emphasizes fast transaction processing and high throughput, while OLAP focuses on fast query response times and scalability for analytical processing.
- **Examples:** OLTP includes systems like banking transactions and order processing, while OLAP includes data warehouses and BI platforms for reporting and analysis.

29. List out the views in the design of a data warehouse. [NOV/DEC 2023]

- Top down approach
- Bottom Up approach

PART B

1. Explain in detail about ETL.

ETL

- ETL is a process in Data Warehousing and it stands for Extract, Transform and Load.
- It is a process in which an ETL tool extracts the data from various data source systems, transforms it in the staging area, and then finally, loads it into the Data Warehouse system.
- ETL stands for Extract, Transform, Load and it is a process used in data warehousing to extract data from various sources, transform it into a format suitable for loading into a data warehouse, and then load it into the warehouse. The process of ETL can be broken down into the following three stages:
- **Extract:** The first stage in the ETL process is to extract data from various sources such as transactional systems, spreadsheets, and flat files. This step involves reading data from the source systems and storing it in a staging area.
- **Transform:** In this stage, the extracted data is transformed into a format that is suitable for loading into the data warehouse. This may involve cleaning and validating the data, converting data types, combining data from multiple sources, and creating new data fields.
- **Load:** After the data is transformed, it is loaded into the data warehouse. This step involves creating the physical data structures and loading the data into the warehouse.
- The ETL process as shown in figure 2.1 is an iterative process that is repeated as new data is added to the warehouse.
- The process is important because it ensures that the data in the data warehouse is accurate, complete, and up-to-date.
- It also helps to ensure that the data is in the format required for data mining and reporting.

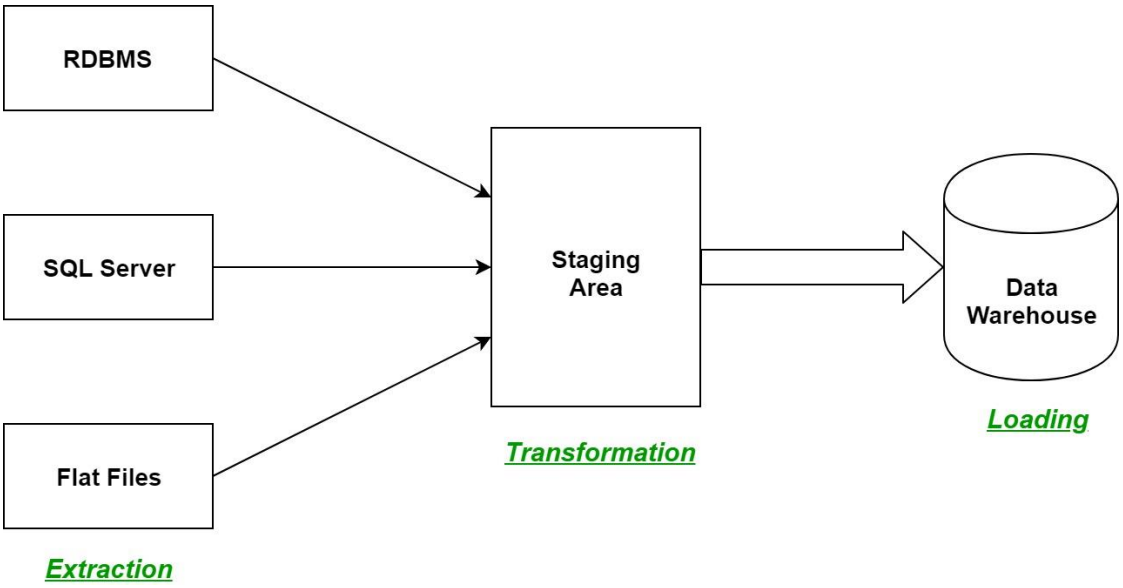


Fig.2.1 ETL

Extraction:

- The first step of the ETL process is extraction. In this step, data from various source systems is extracted which can be in various formats like relational databases, No SQL, XML, and flat files into the staging area.
- It is important to extract the data from various source systems and store it into the staging area first and not directly into the data warehouse because the extracted data is in various formats and can be corrupted also.
- Hence loading it directly into the data warehouse may damage it and rollback will be much more difficult. Therefore, this is one of the most important steps of ETL process.

Transformation:

- The second step of the ETL process is transformation. In this step, a set of rules or functions are applied on the extracted data to convert it into a single standard format. It may involve following processes/tasks:
- Filtering – loading only certain attributes into the data warehouse.
- Cleaning – filling up the NULL values with some default values, mapping U.S.A, United States, and America into USA, etc.

- Joining – joining multiple attributes into one.
- Splitting – splitting a single attribute into multiple attributes.
- Sorting – sorting tuples on the basis of some attribute (generally key-attribute).

Loading:

- The third and final step of the ETL process is loading. In this step, the transformed data is finally loaded into the data warehouse.
- Sometimes the data is updated by loading into the data warehouse very frequently and sometimes it is done after longer but regular intervals.
- The rate and period of loading solely depends on the requirements and varies from system to system.
- ETL process can also use the pipelining concept i.e. as soon as some data is extracted, it can transformed and during that period some new data can be extracted.
- And while the transformed data is being loaded into the data warehouse, the already extracted data can be transformed.
- The block diagram of the pipelining of ETL process is shown in figure 2.2:

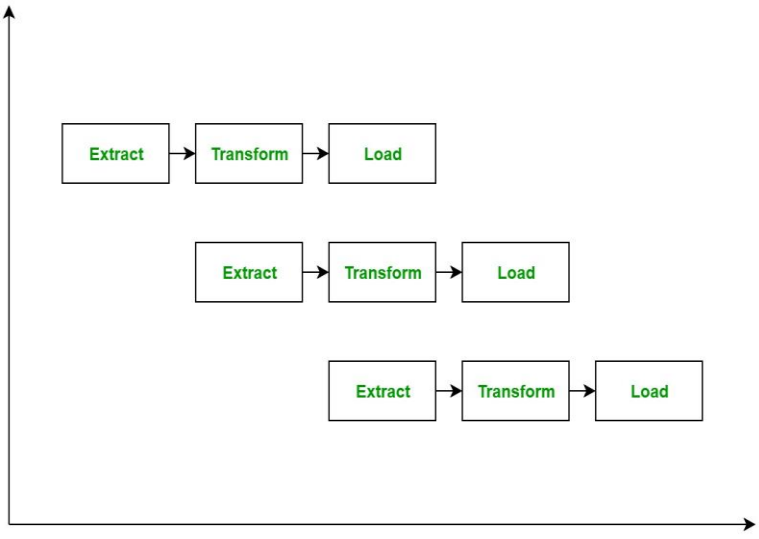


Fig.2.2. ETL Process

ETL Tools:

- ETL tools are Hevo, Sybase, Oracle Warehouse builder, CloverETL, and MarkLogic.

Data Warehouses:

- Data Warehouses are Snowflake, Redshift, BigQuery, and Firebolt.

Advantages of ETL process in data warehousing:

- **Improved data quality**
    - ETL process ensures that the data in the data warehouse is accurate, complete, and up-to-date.
- **Better data integration**
    - ETL process helps to integrate data from multiple sources and systems, making it more accessible and usable.
- **Increased data security**
    - ETL process can help to improve data security by controlling access to the data warehouse and ensuring that only authorized users can access the data.
- **Improved scalability**
    - ETL process can help to improve scalability by providing a way to manage and analyze large amounts of data.
- **Increased automation**
    - ETL tools and technologies can automate and simplify the ETL process, reducing the time and effort required to load and update data in the warehouse.

Disadvantages of ETL process in data warehousing:

- **High cost**
    - ETL process can be expensive to implement and maintain, especially for organizations with limited resources.

- **Complexity**
    - ETL process can be complex and difficult to implement, especially for organizations that lack the necessary expertise or resources.
- **Limited flexibility**
    - ETL process can be limited in terms of flexibility, as it may not be able to handle unstructured data or real-time data streams.
- **Limited scalability**
    - ETL process can be limited in terms of scalability, as it may not be able to handle very large amounts of data.
- **Data privacy concerns**
    - ETL process can raise concerns about data privacy, as large amounts of data are collected, stored, and analyzed.

2. Explain in detail about ETL Vs ELT.

ELT

Extraction, Load and Transform (ELT):

- Extraction, Load and Transform (ELT) is the technique of extracting raw data from the source and storing it in data warehouse of the target server and preparing it for end stream users.

  ELT comprises of 3 different operations performed on the data as shown in figure 2.3:

1. Extract

- Extracting data is the technique of identifying data from one or more sources. The sources may be databases, files, ERP, CRM or any other useful source of data.

2. Load

- Loading is the process of storing the extracted raw data in data warehouse or data lakes.

3. Transform

- Data transformation is the process in which the raw data source is transformed to the target format required for analysis.

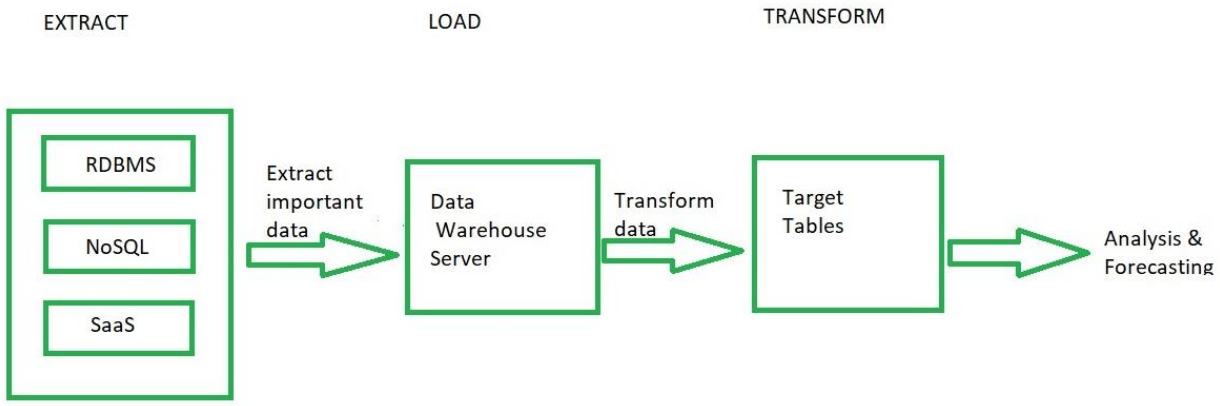


Fig.2.3. ELT

- Data from the sources are extracted and stored in the data warehouse.
- The entire data is not transformed but only the required transformation is done when necessary.
- Raw data can be retrieved from the warehouse anytime when required.
- The data transformed as required is then sent forward for analysis.
- When you use ELT, you move the entire data set as it exists in the source systems to the target.
- This means that you have the raw data at your disposal in the data warehouse, in contrast to the ETL approach.

Extraction, Transform and Load (ETL):

- ETL is the traditional technique of extracting raw data, transforming it for the users as required and storing it in data warehouses as in figure 2.4.
- ELT was later developed, having ETL as its base.
- The three operations happening in ETL and ELT are the same except that their order of processing is slightly varied.

- This change in sequence was made to overcome some drawbacks.

1. Extract

- It is the process of extracting raw data from all available data sources such as databases, files, ERP, CRM or any other.

2. Transform

- The extracted data is immediately transformed as required by the user.

3. Load

- The transformed data is then loaded into the data warehouse from where the users can access it.

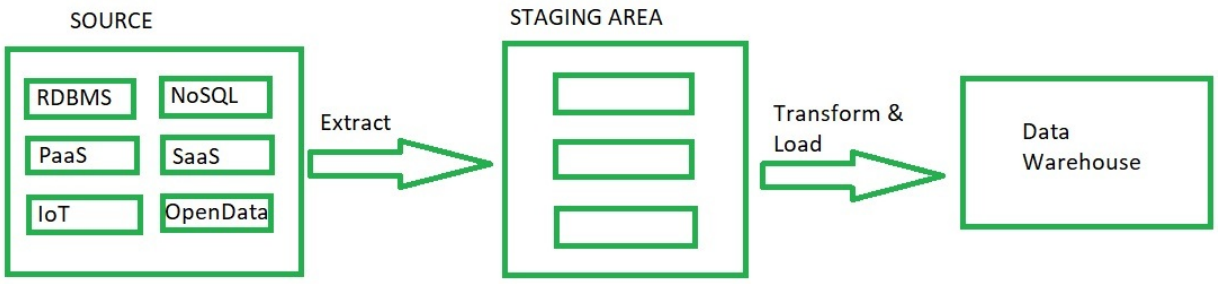


Fig.2.4 ETL

- The data collected from the sources are directly stored in the staging area.
- The transformations required are performed on the data in the staging area.
- Once the data is transformed, the resultant data is stored in the data warehouse.
- The main drawback of ETL architecture is that once the transformed data is stored in the warehouse, it cannot be modified again whereas in ELT, a copy of the raw data is always available in the warehouse and only the required data is transformed when needed.

Difference between ELT and ETL:

ELT	ETL
ELT tools do not require additional hardware	ETL tools require specific hardware with their own engines to perform transformations
Mostly Hadoop or NoSQL database to store data. Rarely RDBMS is used	RDBMS is used exclusively to store data
As all components are in one system, loading is done only once	As ETL uses staging area, extra time is required to load the data
Time to transform data is independent of the size of data	The system has to wait for large sizes of data. As the size of data increases, transformation time also increases
It is cost effective and available to all business using SaaS solution	Not cost effective for small and medium business
The data transformed is used by data scientists and advanced analysts	The data transformed is used by users reading report and SQL coders
Creates ad hoc views. Low cost for building and maintaining	Views are created based on multiple scripts. Deleting view means deleting data

ELT	ETL
Best for unstructured and non-relational data. Ideal for data lakes. Suited for very large amounts of data	Best for relational and structured data. Better for small to medium amounts of data

4. Explain in detail about the Types of Data Warehouses.

Types of Data Warehouses

- The three main types of data warehouses are
    - Enterprise Data Warehouse (EDW)
    - Operational Data Store (ODS)
    - Data Mart

Enterprise Data Warehouse (EDW)

- An enterprise data warehouse (EDW) is a centralized warehouse that provides decision support services across the enterprise.
- EDWs are usually a collection of databases that offer a unified approach for organizing data and classifying data according to subject.

Operational Data Store (ODS)

- An operational data store (ODS) is a central database used for operational reporting as a data source for the enterprise data warehouse described above.
- An ODS is a complementary element to an EDW and is used for operational reporting, controls, and decision making.
- An ODS is refreshed in real-time, making it preferable for routine activities such as storing employee records.
- An EDW, on the other hand, is used for tactical and strategic decision support.

Data Mart

- A data mart is considered a subset of a data warehouse and is usually oriented to a specific team or business line, such as finance or sales.
- It is subject-oriented, making specific data available to a defined group of users more quickly, providing them with critical insights.
- The availability of specific data ensures that they do not need to waste time searching through an entire data warehouse.

5. Explain in detail about Data Warehouse Modeling.

Data Warehouse Modeling

- Data warehouse modeling is the process of designing the schemas of the detailed and summarized information of the data warehouse as in figure 2.5
- The **goal of data warehouse modeling** is to develop a schema describing the reality, or at least a part of the fact, which the data warehouse is needed to support.
- Data warehouse modeling is an essential stage of building a data warehouse for **two main reasons.**
- **Firstly,** through the schema, data warehouse clients can visualize the relationships among the warehouse data, to use them with greater ease.
- **Secondly,** a well-designed schema allows an effective data warehouse structure to emerge, to help decrease the cost of implementing the warehouse and improve the efficiency of using it.
- Data modeling in data warehouses is different from data modeling in operational database systems .
- The primary function of data warehouses is to support DSS processes.
- Thus, the objective of data warehouse modeling is to make the data warehouse efficiently support complex queries on long term information.
- In contrast, data modeling in operational database systems targets efficiently supporting simple transactions in the database such as retrieving, inserting, deleting, and changing data.

- Moreover, data warehouses are designed for the customer with general information knowledge about the enterprise, whereas operational database systems are more oriented toward use by software specialists for creating distinct applications.

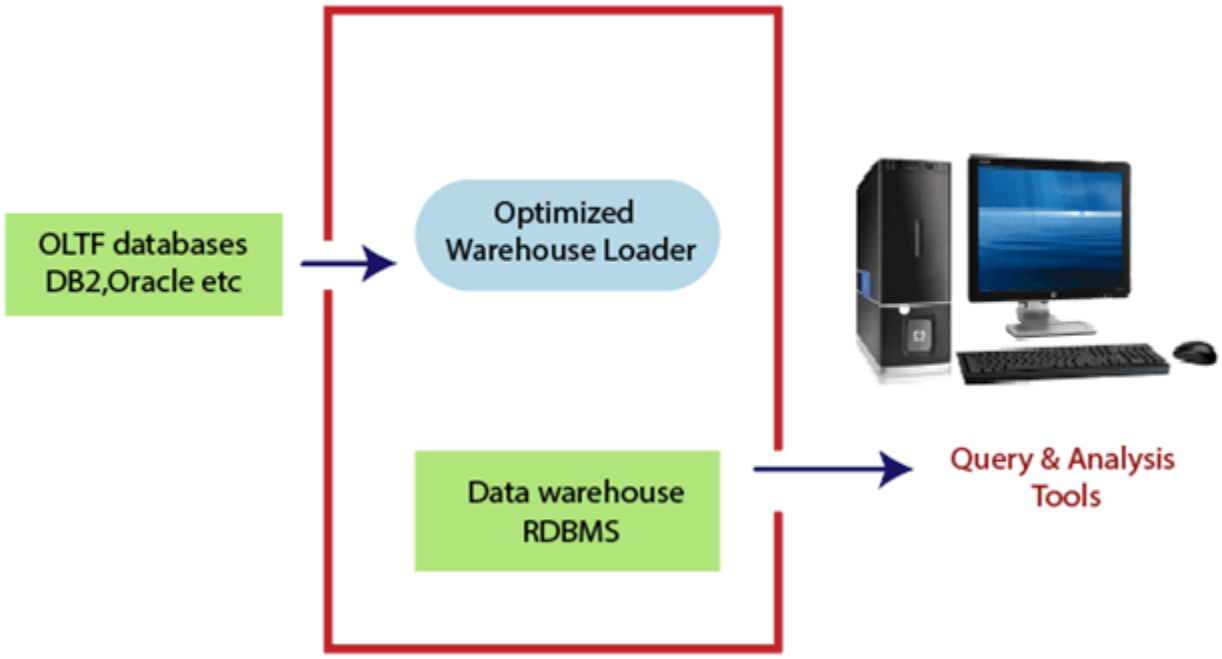


Fig.2.5 Data Warehouse Model

- The data within the specific warehouse itself has a particular architecture with the emphasis on various levels of summarization, as shown in figure 2.6:

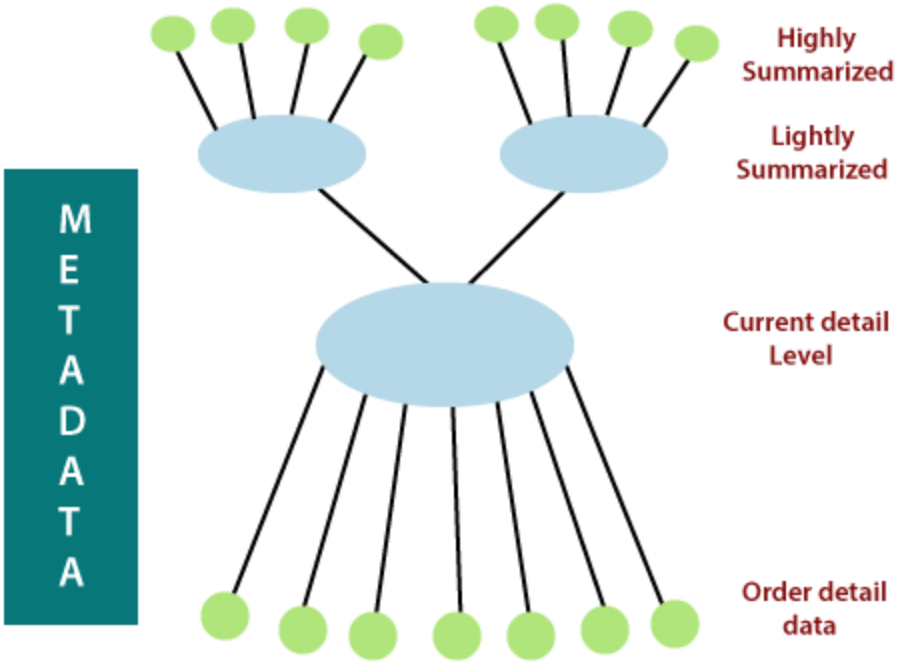


Fig.2.6 Structure of data inside the Data Warehouse

- **Current detail record** is central in importance as it:
  - Reflects the most current happenings, which are commonly the most stimulating.
  - It is numerous as it is saved at the lowest method of the Granularity.
  - It is always (almost) saved on disk storage, which is fast to access but expensive and difficult to manage.
- **Older detail data** is stored in some form of mass storage, and it is infrequently accessed and kept at a level detail consistent with current detailed data.
- **Lightly summarized data** is data extract from the low level of detail found at the current, detailed level and usually is stored on disk storage. When building the data warehouse have to remember what unit of time is summarization done over and also the components or what attributes the summarized data will contain.
- **Highly summarized data** is compact and directly available and can even be found outside the warehouse.

- **Metadata** is the final element of the data warehouses and is really of various dimensions in which it is not the same as file drawn from the operational data, but it is used as:-
  - A directory to help the DSS investigator locate the items of the data warehouse.
  - A guide to the mapping of record as the data is changed from the operational data to the data warehouse environment.
  - A guide to the method used for summarization between the current, accurate data and the lightly summarized information and the highly summarized data, etc.

6. Explain in detail about Data Modeling Life Cycle

- It is a straight forward process of transforming the business requirements to fulfill the goals for storing, maintaining, and accessing the data within IT systems.
- The result is a logical and physical data model for an enterprise data warehouse.
- The objective of the data modeling life cycle is primarily the creation of a storage area for business information.
- That area comes from the logical and physical data modeling stages, as shown in Figure 2.7:

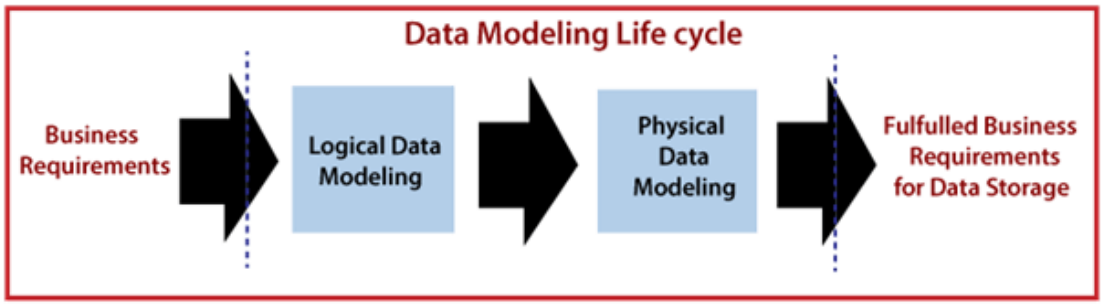


Fig.2.7 Data Modeling Life Cycle

Conceptual Data Model

- A conceptual data model recognizes the highest-level relationships between the different entities. Refer figure 2.8

Characteristics of the conceptual data model

- It contains the essential entities and the relationships among them.
- No attribute is specified.
- No primary key is specified.

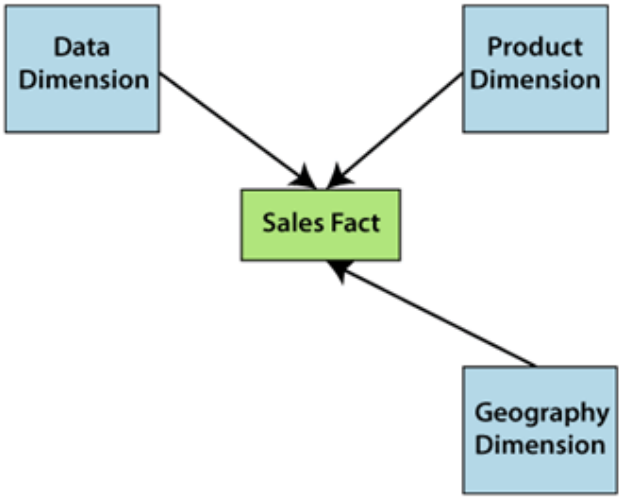


Fig.2.8 Example of Conceptual Data Model

Logical Data Model

- A logical data model defines the information in as much structure as possible, without observing how they will be physically achieved in the database.
- The primary objective of logical data modeling is to document the business data structures, processes, rules, and relationships by a single view - the logical data model. Refer figure 2.9

Features of a logical data model

- It involves all entities and relationships among them.
- All attributes for each entity are specified.
- The primary key for each entity is stated.
- Referential Integrity is specified (FK Relation).

The phase for designing the logical data model which are as follows:

- Specify primary keys for all entities.

- List the relationships between different entities.
- List all attributes for each entity.
- Normalization.
- No data types are listed

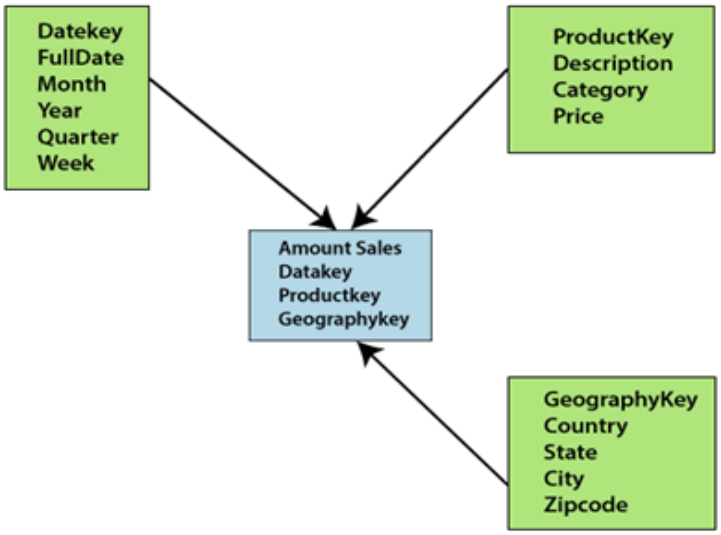


Fig.2.9 Example of Logical Data Model

Physical Data Model

- Physical data model describes how the model will be presented in the database.
- A physical database model demonstrates all table structures, column names, data types, constraints, primary key, foreign key, and relationships between tables.
- The purpose of physical data modeling is the mapping of the logical data model to the physical structures of the RDBMS system hosting the data warehouse.
- This contains defining physical RDBMS structures, such as tables and data types to use when storing the information.
- It may also include the definition of new data structures for enhancing query performance. Refer figure 2.10

Characteristics of a physical data model

- Specification all tables and columns.
- Foreign keys are used to recognize relationships between tables.

The steps for physical data model design which are as follows:

- Convert entities to tables.
- Convert relationships to foreign keys.
- Convert attributes to columns.

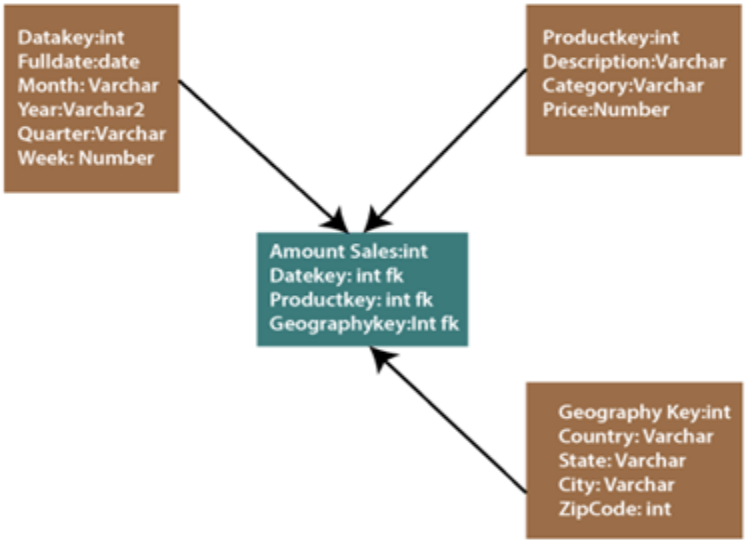


Fig.2.10 Example of Physical Data Model

7. Explain in detail about Types of Data Warehouse Models.

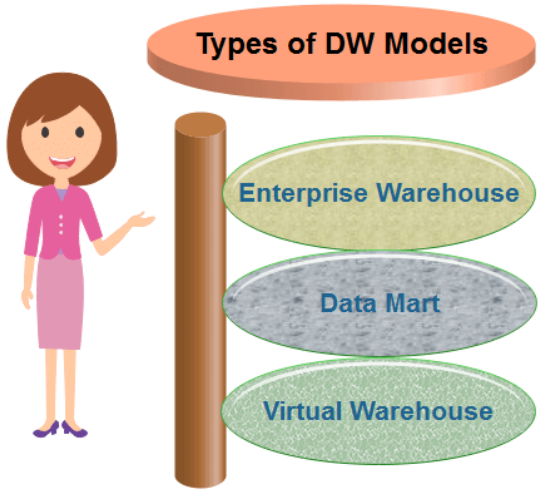


Fig.2.11 Types of DW models

Enterprise Warehouse

- An Enterprise warehouse collects all of the records about subjects spanning the entire organization.
- It supports corporate-wide data integration, usually from one or more operational systems or external data providers, and it's cross-functional in scope.
- It generally contains detailed information as well as summarized information and can range in estimate from a few gigabyte to hundreds of gigabytes, terabytes, or beyond.
- An enterprise data warehouse may be accomplished on traditional mainframes, UNIX super servers, or parallel architecture platforms. It required extensive business modeling and may take years to develop and build.

Data Mart

- A data mart includes a subset of corporate-wide data that is of value to a specific collection of users.
- The scope is confined to particular selected subjects.
- For example, a marketing data mart may restrict its subjects to the customer, items, and sales. The data contained in the data marts tend to be summarized.

Data Marts is divided into two parts:

- **Independent Data Mart:** Independent data mart is sourced from data captured from one or more operational systems or external data providers, or data generally locally within a different department or geographic area.
- **Dependent Data Mart:** Dependent data marts are sourced exactly from enterprise data-warehouses.

Virtual Warehouses

- Virtual Data Warehouses is a set of perception over the operational database.
- For effective query processing, only some of the possible summary vision may be materialized.
- A virtual warehouse is simple to build but required excess capacity on operational database servers.

8. Explain in detail about Data Warehouse Design.

- A data warehouse is a single data repository where a record from multiple data sources is integrated for online business analytical processing (OLAP).
- This implies a data warehouse needs to meet the requirements from all the business stages within the entire organization.
- Thus, data warehouse design is a hugely complex, lengthy, and hence error-prone process.
- Furthermore, business analytical functions change over time, which results in changes in the requirements for the systems.
- Therefore, data warehouse and OLAP systems are dynamic, and the design process is continuous.
- Data warehouse design takes a method different from view materialization in the industries.
- It sees data warehouses as database systems with particular needs such as answering management related queries.
- The target of the design becomes how the record from multiple data sources should be extracted, transformed, and loaded (ETL) to be organized in a database as the data warehouse.

There are two approaches

1. "top-down" approach
2. "bottom-up" approach

Top-down Design Approach

- In the "Top-Down" design approach, a data warehouse is described as a subject-oriented, time-variant, non-volatile and integrated data repository for the entire enterprise data from different sources are validated, reformatted and saved in a normalized (up to 3NF) database as the data warehouse. Refer Figure 2.12
- The data warehouse stores "atomic" information, the data at the lowest level of granularity, from where dimensional data marts can be built by selecting the data required for specific business subjects or particular departments.

- An approach is a data-driven approach as the information is gathered and integrated first and then business requirements by subjects for building data marts are formulated.
- The advantage of this method is which it supports a single integrated data source. Thus data marts built from it will have consistency when they overlap.

Advantages of top-down design

- Data Marts are loaded from the data warehouses.
- Developing new data mart from the data warehouse is very easy.

Disadvantages of top-down design

- This technique is inflexible to changing departmental needs.
- The cost of implementing the project is high.

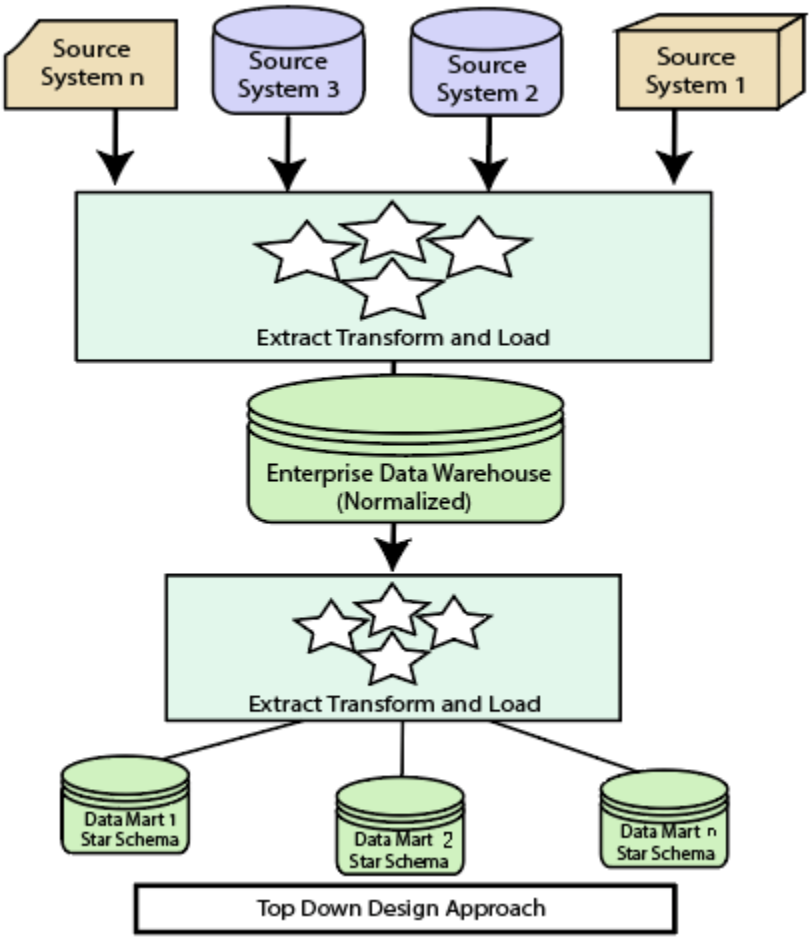


Fig.2.12 Top Down Design Approach

Bottom-Up Design Approach

- In the "Bottom-Up" approach, a data warehouse is described as "a copy of transaction data specifically architecture for query and analysis," term the star schema. In this approach, a data mart is created first to necessary reporting and analytical capabilities for particular business processes (or subjects). Refer figure 2.13
- Thus it is needed to be a business-driven approach in contrast to Inmon's data-driven approach.
- Data marts include the lowest grain data and, if needed, aggregated data too.
- Instead of a normalized database for the data warehouse, a denormalized dimensional database is adapted to meet the data delivery requirements of data warehouses.
- Using this method, to use the set of data marts as the enterprise data warehouse, data marts should be built with conformed dimensions in mind, defining that ordinary objects are represented the same in different data marts.
- The conformed dimensions connected the data marts to form a data warehouse, which is generally called a virtual data warehouse.
- The **advantage** of the "bottom-up" design approach is that it has quick ROI, as developing a data mart, a data warehouse for a single subject, takes far less time and effort than developing an enterprise-wide data warehouse.
- Also, the risk of failure is even less.
- This method is inherently incremental. This method allows the project team to learn and grow.

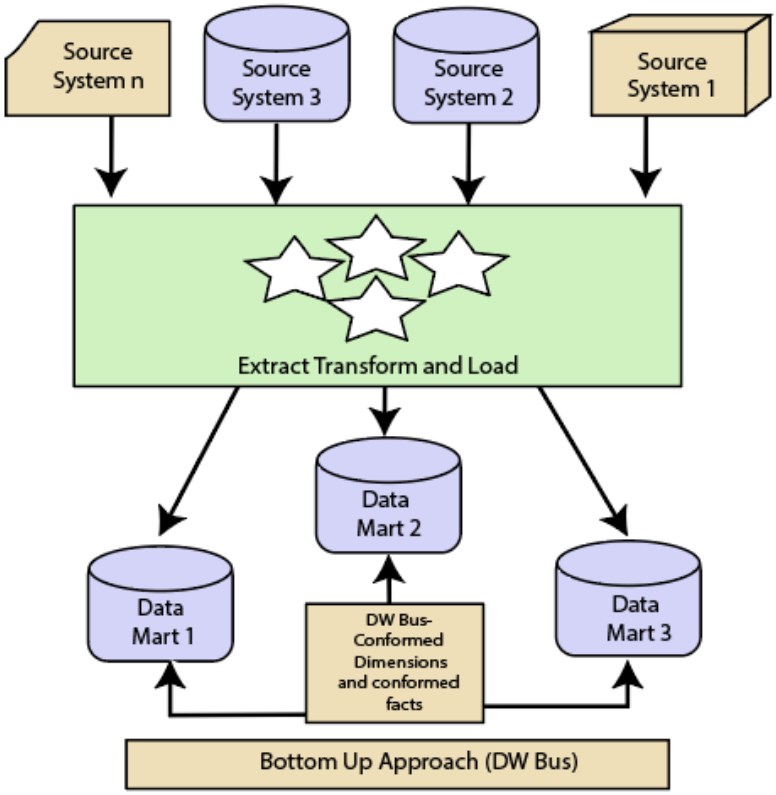


Fig.2.13 Bottom Up Design Approach

Advantages of bottom-up design

- Documents can be generated quickly.
- The data warehouse can be extended to accommodate new business units.
- It is just developing new data marts and then integrating with other data marts.

Disadvantages of bottom-up design

- The locations of the data warehouse and the data marts are reversed in the bottom-up approach design.

9. Differentiate between Top-Down Design Approach and Bottom-Up Design Approach.

Top-Down Design Approach	Bottom-Up Design Approach
Breaks the vast problem into smaller sub problems.	Solves the essential low-level problem and integrates them into a higher one.
Inherently architected- not a union of several data marts.	Inherently incremental; can schedule essential data marts first.
Single, central storage of information about the content.	Departmental information stored.
Centralized rules and control.	Departmental rules and control.
It includes redundant information.	Redundancy can be removed.
It may see quick results if implemented with repetitions.	Less risk of failure, favorable return on investment, and proof of techniques.

10. Explain in detail about Data Warehouse Delivery Process.

- Main steps used in data warehouse delivery process which are as follows in figure 2.14:

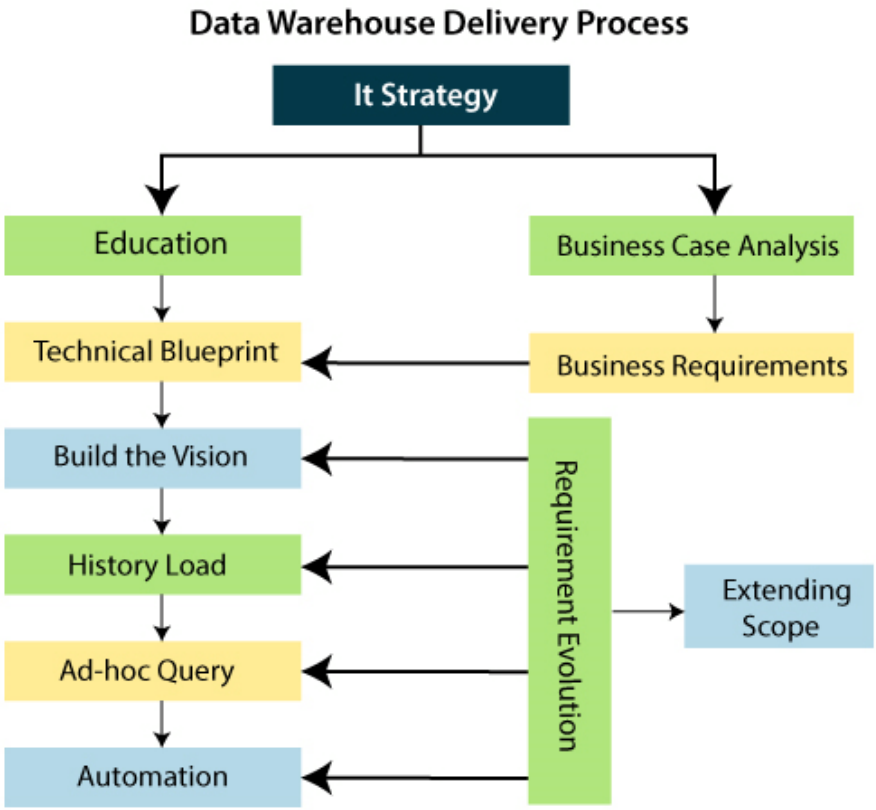


Fig.2.14 Data Warehouse Delivery Process

IT Strategy:

- DWH project must contain IT strategy for procuring and retaining funding.

Business Case Analysis:

- After the IT strategy has been designed, the next step is the business case.
- It is essential to understand the level of investment that can be justified and to recognize the projected business benefits which should be derived from using the data warehouse.

Education & Prototyping:

- Company will experiment with the ideas of data analysis and educate themselves on the value of the data warehouse.

- This is valuable and should be required if this is the company first exposure to the benefits of the DS record.
- Prototyping method can progress the growth of education. It is better than working models. Prototyping requires business requirement, technical blueprint, and structures.

Business Requirement:

It contains such as

- The logical model for data within the data warehouse.
- The source system that provides this data (mapping rules)
- The business rules to be applied to information.
- The query profiles for the immediate requirement

Technical blueprint:

- It arranges the architecture of the warehouse. Technical blueprint of the delivery process makes an architecture plan which satisfies long-term requirements.
- It lays server and data mart architecture and essential components of database design.

Building the vision:

- It is the phase where the first production deliverable is produced.
- This stage will probably create significant infrastructure elements for extracting and loading information but limit them to the extraction and load of information sources.

History Load:

- The next step is one where the remainder of the required history is loaded into the data warehouse.
- This means that the new entities would not be added to the data warehouse, but additional physical tables would probably be created to save the increased record volumes.

AD-Hoc Query:

- In this step, we configure an ad-hoc query tool to operate against the data warehouse.
- These end-customer access tools are capable of automatically generating the database query that answers any question posed by the user.

Automation:

- The automation phase is where many of the operational management processes are fully automated within the DWH. These would include:
  - Extracting & loading the data from a variety of sources systems
  - Transforming the information into a form suitable for analysis
  - Backing up, restoring & archiving data
  - Generating aggregations from predefined definitions within the Data Warehouse.
  - Monitoring query profiles & determining the appropriate aggregates to maintain system performance.

Extending Scope:

- In this phase, the scope of DWH is extended to address a new set of business requirements.
- This involves the loading of additional data sources into the DWH i.e. the introduction of new data marts.

Requirement Evolution:

- This is the last step of the delivery process of a data warehouse.
- As we all know that requirements are not static and evolve continuously.
- As the business requirements will change it supports to be reflected in the system.

11. Explain in detail about OLAP (Online Analytical Processing).

- **OLAP** stands for **On-Line Analytical Processing**.
- OLAP is a classification of software technology which authorizes analysts, managers, and executives to gain insight into information through fast, consistent, interactive access in a wide variety of possible views of data that has been transformed from raw information to reflect the real dimensionality of the enterprise as understood by the clients.
- **OLAP** implement the multidimensional analysis of business information and support the capability for complex estimations, trend analysis, and sophisticated data modeling.
- It is rapidly enhancing the essential foundation for Intelligent Solutions containing Business Performance Management, Planning, Budgeting, Forecasting, Financial Documenting, Analysis, Simulation-Models, Knowledge Discovery, and Data Warehouses Reporting.
- OLAP enables end-clients to perform ad hoc analysis of record in multiple dimensions, providing the insight and understanding they require for better decision making.

Who uses OLAP and Why

- OLAP applications are used by a variety of the functions of an organization.

Finance and accounting:

- Budgeting
- Activity-based costing
- Financial performance analysis
- And financial modeling

Sales and Marketing

- Sales analysis and forecasting
- Market research analysis
- Promotion analysis

- Customer analysis
- Market and customer segmentation

Production

- Production planning
- Defect analysis

- OLAP cubes have two main purposes.
  - The first is to provide business users with a data model more intuitive to them than a tabular model. This model is called a Dimensional Model.
  - The second purpose is to enable fast query response that is usually difficult to achieve using tabular models.

How OLAP Works?

- OLAP has a very simple concept.
- It pre-calculates most of the queries that are typically very hard to execute over tabular databases, namely aggregation, joining, and grouping.
- These queries are calculated during a process that is usually called 'building' or 'processing' of the OLAP cube.
- This process happens overnight, and by the time end users get to work - data will have been updated.

OLAP Guidelines (Dr.E.F.Codd Rule)

- Dr E.F. Codd, the "father" of the relational model, has formulated a list of 12 guidelines and requirements as the basis for selecting OLAP systems as in figure 2.15:

Fig.2.15 OLAP Guidelines

1) Multidimensional Conceptual View:

- This is the central features of an OLAP system. By needing a multidimensional view, it is possible to carry out methods like slice and dice.

2) Transparency:

- Make the technology, underlying information repository, computing operations, and the dissimilar nature of source data totally transparent to users. Such transparency helps to improve the efficiency and productivity of the users.

3) Accessibility:

- It provides access only to the data that is actually required to perform the particular analysis, present a single, coherent, and consistent view to the clients.
- The OLAP system must map its own logical schema to the heterogeneous physical data stores and perform any necessary transformations.
- The OLAP operations should be sitting between data sources (e.g., data warehouses) and an OLAP front-end.

4) Consistent Reporting Performance:

- To make sure that the users do not feel any significant degradation in documenting performance as the number of dimensions or the size of the database increases.

- That is, the performance of OLAP should not suffer as the number of dimensions is increased. Users must observe consistent run time, response time, or machine utilization every time a given query is run.

5) Client/Server Architecture:

- Make the server component of OLAP tools sufficiently intelligent that the various clients to be attached with a minimum of effort and integration programming.
- The server should be capable of mapping and consolidating data between dissimilar databases.

6) Generic Dimensionality:

- An OLAP method should treat each dimension as equivalent in both is structure and operational capabilities.
- Additional operational capabilities may be allowed to selected dimensions, but such additional tasks should be grantable to any dimension.

7) Dynamic Sparse Matrix Handling:

- To adapt the physical schema to the specific analytical model being created and loaded that optimizes sparse matrix handling.
- When encountering the sparse matrix, the system must be easy to dynamically assume the distribution of the information and adjust the storage and access to obtain and maintain a consistent level of performance.

8) Multiuser Support:

- OLAP tools must provide concurrent data access, data integrity, and access security.

9) Unrestricted cross-dimensional Operations:

- It provides the ability for the methods to identify dimensional order and necessarily functions roll-up and drill-down methods within a dimension or across the dimension.

10) Intuitive Data Manipulation:

- Data Manipulation fundamental the consolidation direction like as reorientation (pivoting), drill-down and roll-up, and another manipulation to be accomplished

naturally and precisely via point-and-click and drag and drop methods on the cells of the scientific model.

- It avoids the use of a menu or multiple trips to a user interface.

11) Flexible Reporting:

- It implements efficiency to the business clients to organize columns, rows, and cells in a manner that facilitates simple manipulation, analysis, and synthesis of data.

12) Unlimited Dimensions and Aggregation Levels:

- The number of data dimensions should be unlimited. Each of these common dimensions must allow a practically unlimited number of customer-defined aggregation levels within any given consolidation path.

12. Explain in detail about Characteristics of OLAP.

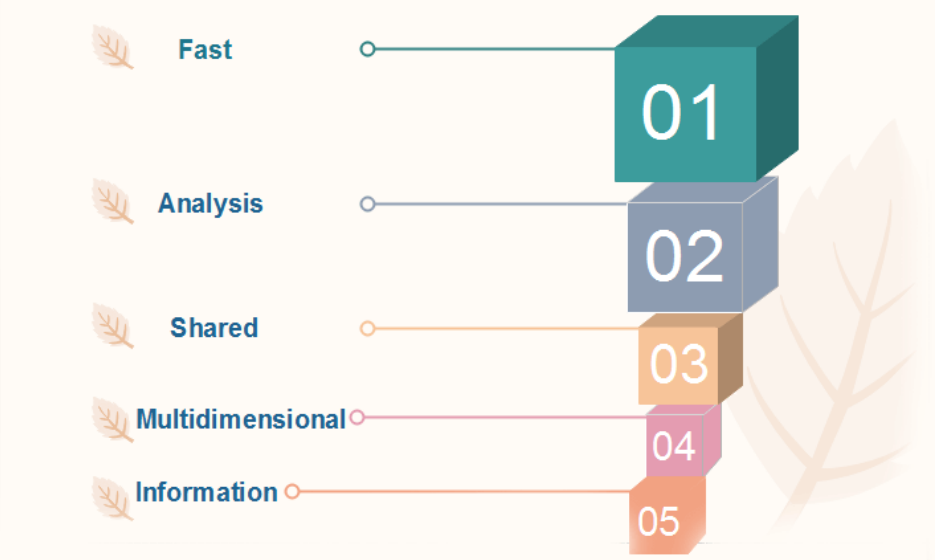


Fig.2.15 Characteristics of OLAP

Fast

- It defines which the system targeted to deliver the most feedback to the client within about five seconds, with the elementary analysis taking no more than one second and very few taking more than 20 seconds.

Analysis

- It defines which the method can cope with any business logic and statistical analysis that is relevant for the function and the user, keep it easy enough for the target client.

Share

- It defines which the system tools all the security requirements for understanding and, if multiple write connection is needed, concurrent update location at an appropriated level, not all functions need customer to write data back, but for the increasing number which does, the system should be able to manage multiple updates in a timely, secure manner.

Multidimensional

- This is the basic requirement. OLAP system must provide a multidimensional conceptual view of the data, including full support for hierarchies, as this is certainly the most logical method to analyze business and organizations.

Information

- The system should be able to hold all the data needed by the applications. Data sparsity should be handled in an efficient manner.

The main characteristics of OLAP are as follows:

1. **Multidimensional conceptual view:** OLAP systems let business users have a dimensional and logical view of the data in the data warehouse. It helps in carrying slice and dice operations.
2. **Multi-User Support:** Since the OLAP techniques are shared, the OLAP operation should provide normal database operations, containing retrieval, update, adequacy control, integrity, and security.
3. **Accessibility:** OLAP acts as a mediator between data warehouses and front-end. The OLAP operations should be sitting between data sources (e.g., data warehouses) and an OLAP front-end.
4. **Storing OLAP results:** OLAP results are kept separate from data sources.

5. **Uniform documenting performance:** Increasing the number of dimensions or database size should not significantly degrade the reporting performance of the OLAP system.
6. OLAP provides for distinguishing between zero values and missing values so that aggregates are computed correctly.
7. OLAP system should ignore all missing values and compute correct aggregate values.
8. OLAP facilitate interactive query and complex analysis for the users.
9. OLAP allows users to drill down for greater details or roll up for aggregations of metrics along a single business dimension or across multiple dimension.
10. OLAP provides the ability to perform intricate calculations and comparisons.
11. OLAP presents results in a number of meaningful ways, including charts and graphs.

Benefits of OLAP

OLAP holds several benefits for businesses: -

1. OLAP helps managers in decision-making through the multidimensional record views that it is efficient in providing, thus increasing their productivity.
2. OLAP functions are self-sufficient owing to the inherent flexibility support to the organized databases.
3. It facilitates simulation of business models and problems, through extensive management of analysis-capabilities.
4. In conjunction with data warehouse, OLAP can be used to support a reduction in the application backlog, faster data retrieval, and reduction in query drag.

Motivations for using OLAP

1) Understanding and improving sales:

- For enterprises that have much products and benefit a number of channels for selling the product, OLAP can help in finding the most suitable products and the most famous channels.

- In some methods, it may be feasible to find the most profitable users.
- **For example,** considering the telecommunication industry and considering only one product, communication minutes, there is a high amount of record if a company want to analyze the sales of products for every hour of the day (24 hours), difference between weekdays and weekends (2 values) and split regions to which calls are made into 50 region.

2) Understanding and decreasing costs of doing business:

- Improving sales is one method of improving a business, the other method is to analyze cost and to control them as much as suitable without affecting sales.
- OLAP can assist in analyzing the costs related to sales. In some methods, it may also be feasible to identify expenditures which produce a high return on investments (ROI).
- **For example,** recruiting a top salesperson may contain high costs, but the revenue generated by the salesperson may justify the investment.

Difference between OLTP and OLAP

- **OLTP (On-Line Transaction Processing)** is featured by a large number of short on-line transactions (INSERT, UPDATE, and DELETE).
- The primary significance of OLTP operations is put on very rapid query processing, maintaining record integrity in multi-access environments, and effectiveness consistent by the number of transactions per second.
- In the OLTP database, there is an accurate and current record, and schema used to save transactional database is the entity model (usually 3NF).
- **OLAP (On-line Analytical Processing)** is represented by a relatively low volume of transactions. Queries are very difficult and involve aggregations. For OLAP operations, response time is an effectiveness measure.
- OLAP applications are generally used by Data Mining techniques. In OLAP database there is aggregated, historical information, stored in multi-dimensional schemas (generally star schema).

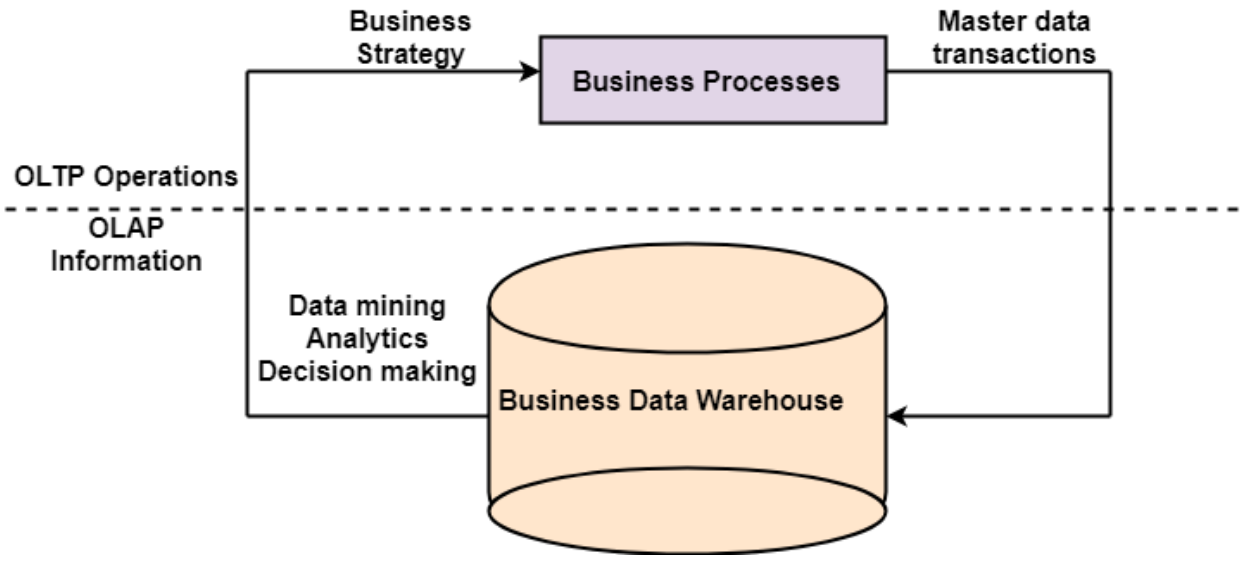


Fig.2.16 Difference between OLAP and OLTP

Following are the difference between OLAP and OLTP system.

1) Users:

- **OLTP** systems are designed for office worker while the OLAP systems are designed for decision-makers.
- Therefore while an OLTP method may be accessed by hundreds or even thousands of clients in a huge enterprise, an OLAP system is suitable to be accessed only by a select class of manager and may be used only by dozens of users.

2) Functions:

- **OLTP** systems are mission-critical. They provide day-to-day operations of an enterprise and are largely performance and availability driven.
- These operations carry out simple repetitive operations. **OLAP** systems are management-critical to support the decision of enterprise support tasks using detailed investigation.

3) Nature:

- Although **SQL** queries return a set of data, OLTP methods are designed to step one record at the time, for example, a data related to the user who may be on the phone or in the store. OLAP system is not designed to deal with individual customer records.

- Instead, they include queries that deal with many data at a time and provide summary or aggregate information to a manager.
- OLAP applications include data stored in a data warehouses that have been extracted from many tables and possibly from more than one enterprise database.

4) Design:

- **OLTP** database operations are designed to be application-oriented while **OLAP** operations are designed to be subject-oriented.
- OLTP systems view the enterprise record as a collection of tables (possibly based on an entity-relationship model).
- **OLAP** operations view enterprise information as multidimensional).

5) Data:

- OLTP systems usually deal only with the current status of data.
- For example, a record about an employee who left three years ago may not be feasible on the Human Resources System.
- The old data may have been achieved on some type of stable storage media and may not be accessible online.
- On the other hand, OLAP systems needed historical data over several years since trends are often essential in decision making.

6) Kind of use:

- **OLTP** methods are used for reading and writing operations while OLAP methods usually do not update the data.

7) View:

- An **OLTP** system focuses primarily on the current data within an enterprise or department, which does not refer to historical data or data in various organizations.
- In contrast, an **OLAP** system spans multiple version of a database schema, due to the evolutionary process of an organization.
- OLAP system also deals with information that originates from different organizations, integrating information from many data stores.

- Because of their huge volume, these are stored on multiple storage media.

8) Access Patterns:

- The access pattern of an OLTP system consist primarily of short, atomic transactions.
- Such a system needed concurrency control and recovery techniques.
- However, access to OLAP systems is mostly read-only operations because these data warehouses store historical information
- The biggest difference between an OLTP and OLAP system is the amount of data analyzed in a single transaction.
- Whereas an OLTP handles many concurrent customers and queries touching only a single data or limited collection of records at a time, an OLAP system must have the efficiency to operate on millions of data to answer a single query.

13. Explain in detail about OLAP Operations in the Multidimensional Data Model.

Discuss the typical OLAP operations with an example. (KDD) [Nov 2024]

- In the multidimensional model, the records are organized into various dimensions, and each dimension includes multiple levels of abstraction described by concept hierarchies.
- This organization support users with the flexibility to view data from various perspectives.
- A number of OLAP data cube operation exist to demonstrate these different views, allowing interactive queries and search of the record at hand.
- Hence, OLAP supports a user-friendly environment for interactive data analysis.
- Consider the OLAP operations which are to be performed on multidimensional data. The figure shows data cubes for sales of a shop.
- The cube contains the dimensions, location, and time and item, where the **location** is aggregated with regard to city values, **time** is aggregated with respect to quarters, and an **item** is aggregated with respect to item types.

1. Roll-Up Operation

- The roll-up operation **(also known as drill-up or aggregation operation)** performs aggregation on a data cube, by climbing down concept hierarchies, i.e., dimension reduction.
- Roll-up is like **zooming-out** on the data cubes. Figure shows the result of roll-up operations performed on the dimension location.
- The hierarchy for the location is defined as the Order Street, city, province, or state, country.
- The roll-up operation aggregates the data by ascending the location hierarchy from the level of the city to the level of the country.
- When a roll-up is performed by dimensions reduction, one or more dimensions are removed from the cube.
- For example, consider a sales data cube having two dimensions, location and time.
- Roll-up may be performed by removing, the time dimensions, appearing in an aggregation of the total sales by location, relatively than by location and by time.

Example

Consider the following cubes illustrating temperature of certain days recorded weekly:

Temperature	64	65	68	69	70	71	72	75	80	81	83	85
Week1	1	0	1	0	1	0	0	0	0	0	1	0
Week2	0	0	0	1	0	0	1	2	0	1	0	0

- Consider that we want to set up levels (hot (80-85), mild (70-75), cool (64-69)) in temperature from the above cubes.
- To do this, we have to group column and add up the value according to the concept hierarchies. This operation is known as a roll-up.

- By doing this, we contain the following cube:

Temperature	cool	mild	hot
Week1	2	1	1
Week2	2	1	1

The roll-up operation groups the information by levels of temperature

The following figure 2.16 illustrates how roll-up works.

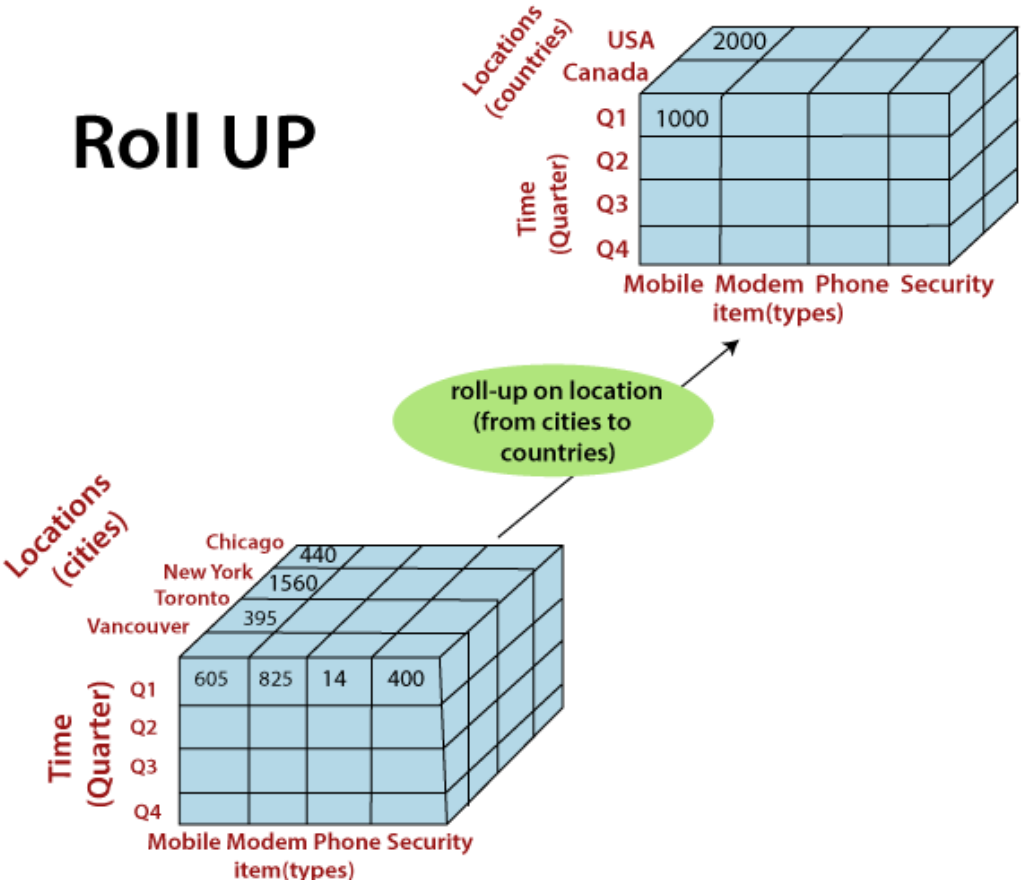


Fig.2.16 Roll Up Operation

2. Drill-Down Operation

- The drill-down operation **(also called roll-down)** is the reverse operation of **roll-up**.
- Drill-down is like **zooming-in** on the data cube. It navigates from less detailed record to more detailed data.
- Drill-down can be performed by either **stepping down** a concept hierarchy for a dimension or adding additional dimensions.
- Figure shows a drill-down operation performed on the dimension time by stepping down a concept hierarchy which is defined as day, month, quarter, and year.
- Drill-down appears by descending the time hierarchy from the level of the quarter to a more detailed level of the month.
- Because a drill-down adds more details to the given data, it can also be performed by adding a new dimension to a cube.
- For example, a drill-down on the central cubes of the figure can occur by introducing an additional dimension, such as a customer group.

Example

- Drill-down adds more details to the given data

Temperature	cool	mild	hot
Day 1	0	0	0
Day 2	0	0	0
Day 3	0	0	1
Day 4	0	1	0
Day 5	1	0	0

Day 6	0	0	0
Day 7	1	0	0
Day 8	0	0	0
Day 9	1	0	0
Day 10	0	1	0
Day 11	0	1	0
Day 12	0	1	0
Day 13	0	0	1
Day 14	0	0	0

The following figure 2.17 illustrates how Drill-down works.

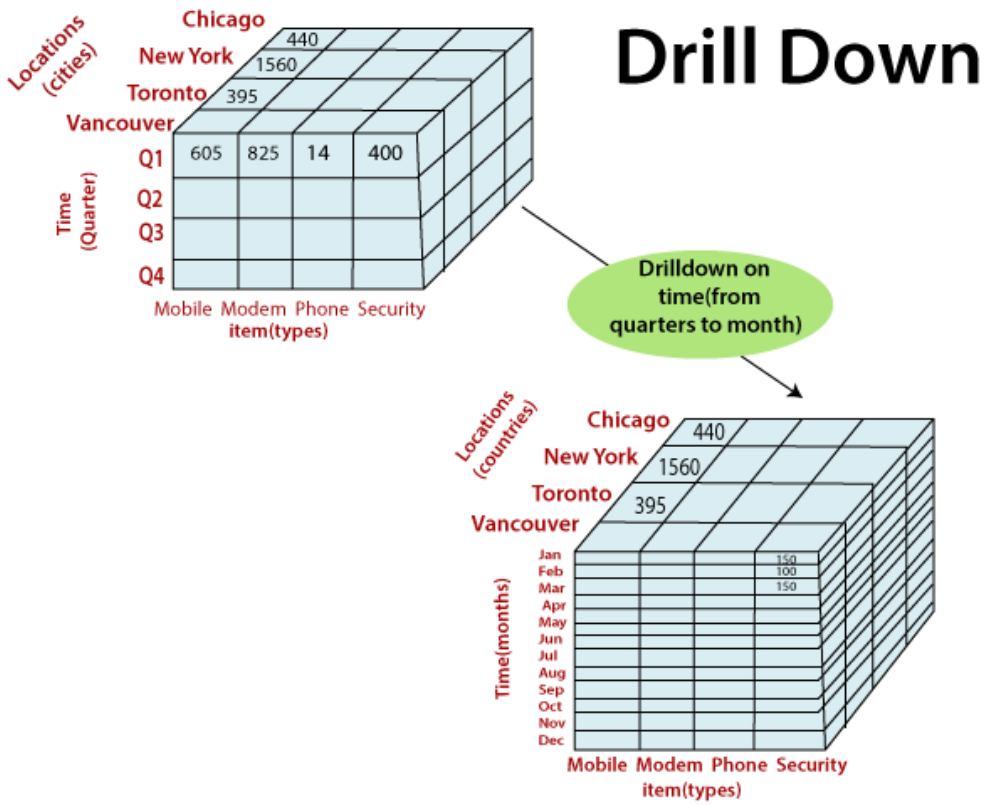


Fig.2.17 Drill Down

3. Slice Operation

- A **slice** is a subset of the cubes corresponding to a single value for one or more members of the dimension.
- For example, a slice operation is executed when the customer wants a selection on one dimension of a three-dimensional cube resulting in a two-dimensional site.
- So, the Slice operations perform a selection on one dimension of the given cube, thus resulting in a subcube.
- For example, if we make the selection, temperature=cool we will obtain the following cube:

Temperature	cool
Day 1	0
Day 2	0
Day 3	0
Day 4	0
Day 5	1
Day 6	1
Day 7	1
Day 8	1
Day 9	1
Day 11	0
Day 12	0
Day 13	0
Day 14	0

The following figure 2.18 illustrates how Slice works.

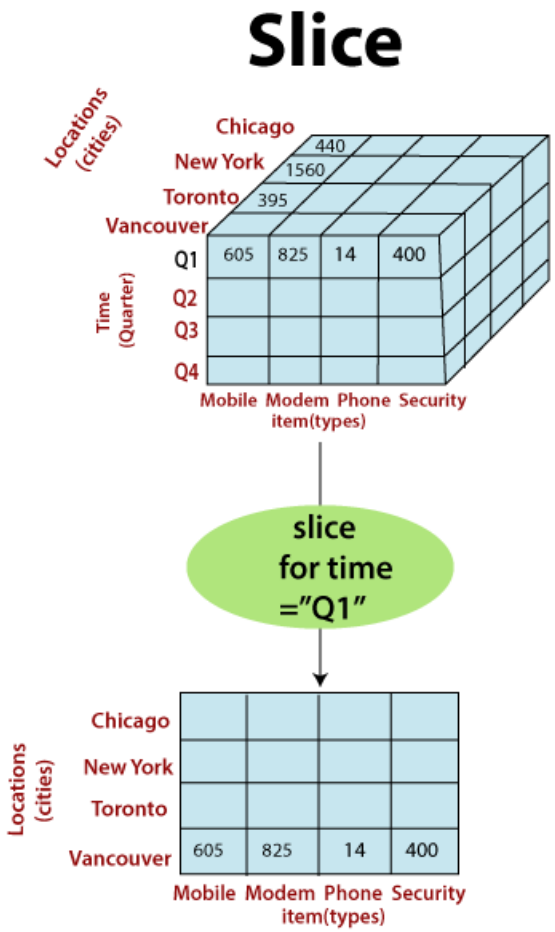


Fig.2.18 Slice

- Here Slice is functioning for the dimensions "time" using the criterion time = "Q1".
- It will form a new sub-cubes by selecting one or more dimensions.

4. Dice Operation

- The dice operation describes a sub cube by operating a selection on two or more dimension.
- **For example**, Implement the selection (time = day 3 OR time = day 4) AND (temperature = cool OR temperature = hot) to the original cubes we get the following sub cube (still two-dimensional)

Temperature	cool	hot
Day 3	0	1
Day 4	0	0

Consider the following figure 2.19, which shows the dice operations.

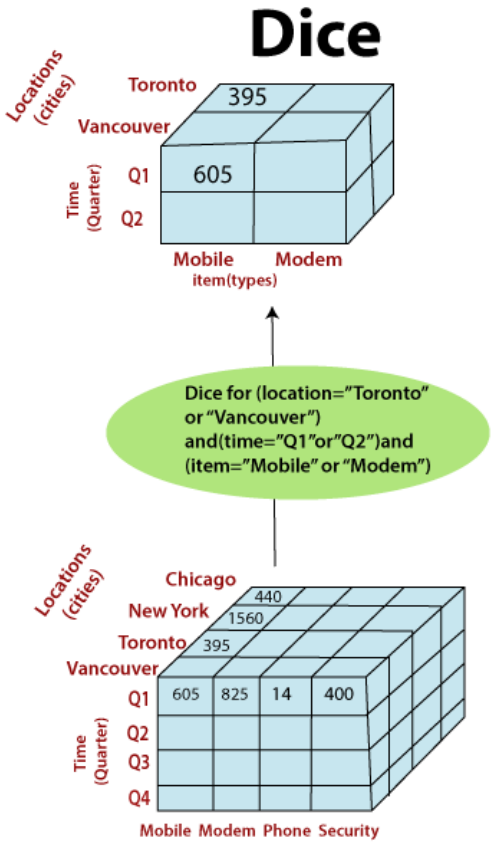


Fig.2.19 Dice Operation

- The dice operation on the cubes based on the following selection criteria involves three dimensions.
- (location = "Toronto" or "Vancouver")
- (time = "Q1" or "Q2")
- (item =" Mobile" or "Modem")

5. Pivot Operation

- The pivot operation is also called a rotation.
- Pivot is a visualization operations which rotates the data axes in view to provide an alternative presentation of the data.
- It may contain swapping the rows and columns or moving one of the row-dimensions into the column dimensions as in figure 2.20.

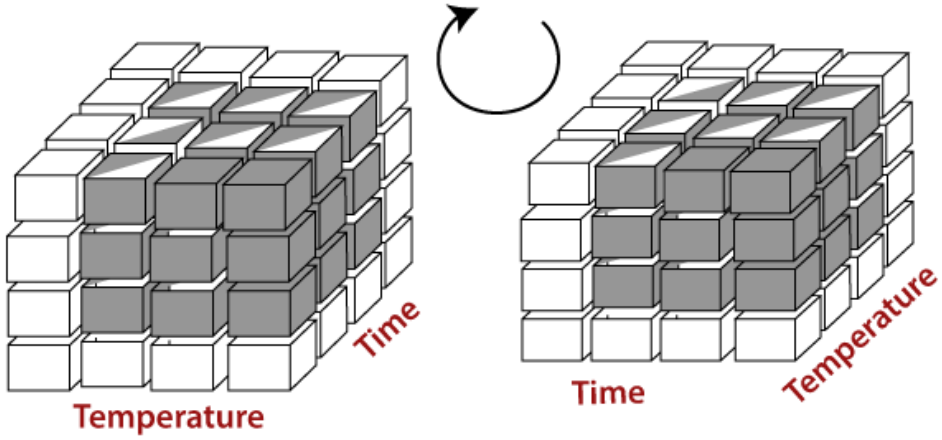


Fig.2.20 Data Cube

Consider the following figure 2.21, which shows the pivot operation.

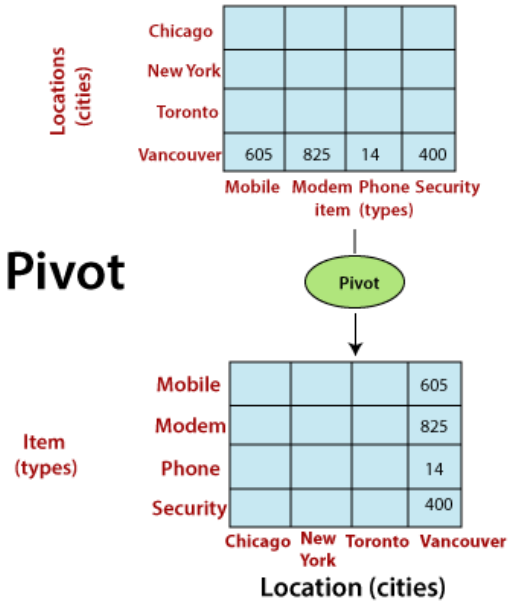


Fig.2.21 Pivot Operation

14. Explain in detail about Types of OLAP. [NOV 2023]

Diagrammatically illustrate and describe the architecture of MOLAP,ROLAP and HOLAP. [Nov 2024]

Types of OLAP – Refer Figure 2.22

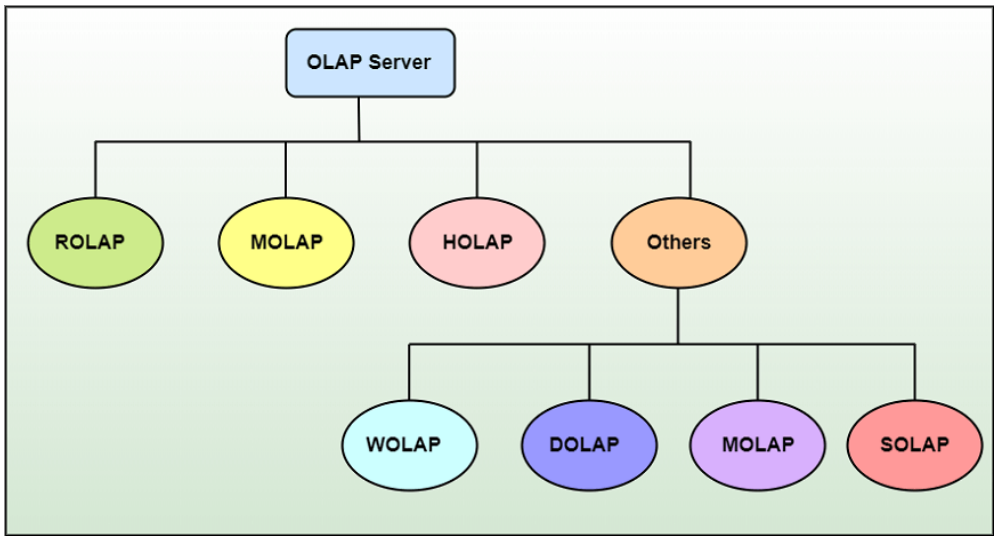


Fig.2.22 Types of OLAP

- **ROLAP** stands for Relational OLAP, an application based on relational DBMSs.
- **MOLAP** stands for Multidimensional OLAP, an application based on multidimensional DBMSs.
- **HOLAP** stands for Hybrid OLAP, an application using both relational and multidimensional techniques.

1. Relational OLAP (ROLAP) Server [NOV/DEC 2023]

- These are intermediate servers which stand in between a relational back-end server and user frontend tools.
- They use a relational or extended-relational DBMS to save and handle warehouse data, and OLAP middleware to provide missing pieces.
- ROLAP servers contain optimization for each DBMS back end, implementation of aggregation navigation logic, and additional tools and services.
- ROLAP technology tends to have higher scalability than MOLAP technology.

- ROLAP systems work primarily from the data that resides in a relational database, where the base data and dimension tables are stored as relational tables. This model permits the multidimensional analysis of data.
- This technique relies on manipulating the data stored in the relational database to give the presence of traditional OLAP's slicing and dicing functionality.
- In essence, each method of slicing and dicing is equivalent to adding a "WHERE" clause in the SQL statement.

Relational OLAP Architecture

ROLAP Architecture in figure 2.23 includes the following components

- Database server.
- ROLAP server.
- Front-end tool.

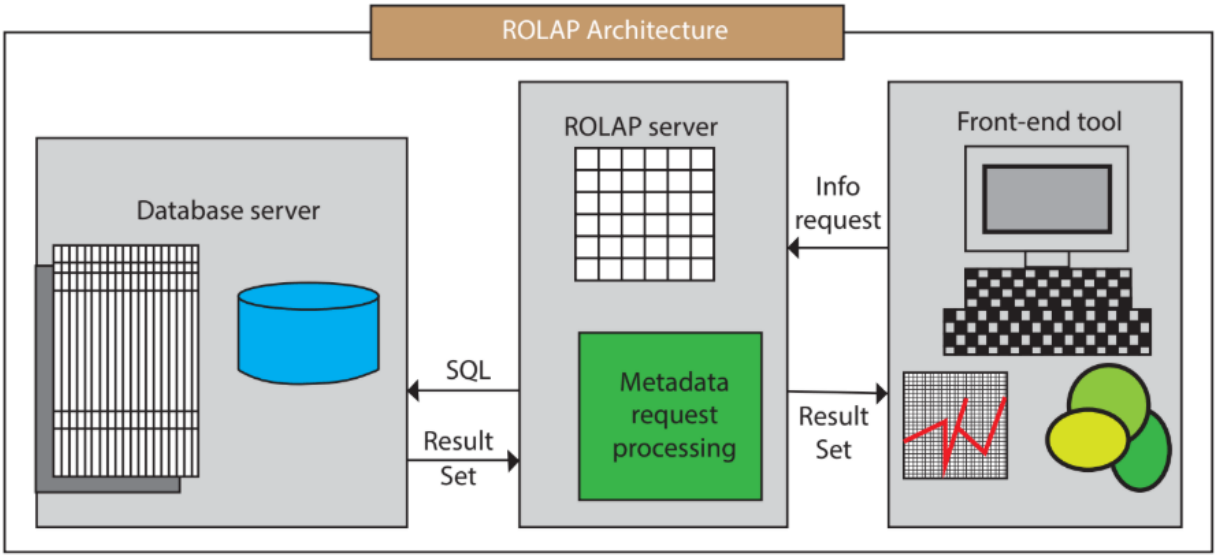


Fig.2.23 ROLAP Architecture

- **Relational OLAP (ROLAP)** is the latest and fastest-growing OLAP technology segment in the market.
- This method allows multiple multidimensional views of two-dimensional relational tables to be created, avoiding structuring record around the desired view.

- Some products in this segment have supported reliable SQL engines to help the complexity of multidimensional analysis.
- This includes creating multiple SQL statements to handle user requests, being 'RDBMS' aware and also being capable of generating the SQL statements based on the optimizer of the DBMS engine.

Advantages

Can handle large amounts of information:

- The data size limitation of ROLAP technology is depends on the data size of the underlying RDBMS. So, ROLAP itself does not restrict the data amount.
- <="" strong="">RDBMS already comes with a lot of features. So ROLAP technologies, (works on top of the RDBMS) can control these functionalities.

Disadvantages

Performance can be slow:

- Each ROLAP report is a SQL query (or multiple SQL queries) in the relational database, the query time can be prolonged if the underlying data size is large.

Limited by SQL functionalities:

- ROLAP technology relies on upon developing SQL statements to query the relational database, and SQL statements do not suit all needs.

2. Multidimensional OLAP (MOLAP) Server

- A MOLAP system is based on a native logical model that directly supports multidimensional data and operations.
- Data are stored physically into multidimensional arrays, and positional techniques are used to access them.
- One of the significant distinctions of **MOLAP** against a **ROLAP** is that data are summarized and are stored in an optimized format in a multidimensional cube, instead of in a relational database.
- In MOLAP model, data are structured into proprietary formats by client's reporting requirements with the calculations pre-generated on the cubes.

MOLAP Architecture [NOV/DEC 2023]

MOLAP Architecture in figure 2.24 includes the following components

- Database server.
- MOLAP server.
- Front-end tool.

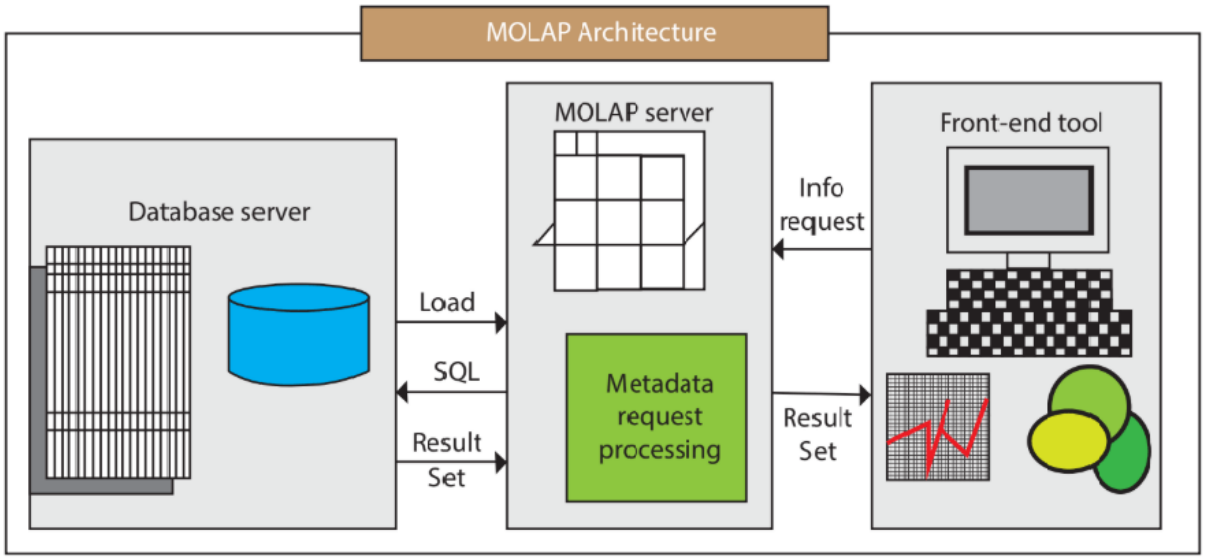


Fig.2.24 MOLAP Architecture

- **MOLAP** structure primarily reads the precompiled data. MOLAP structure has limited capabilities to dynamically create aggregations or to evaluate results which have not been pre-calculated and stored.
- Applications requiring iterative and comprehensive time-series analysis of trends are well suited for MOLAP technology (e.g., financial analysis and budgeting).
- Examples include Arbor Software's Essbase. Oracle's Express Server, Pilot Software's Lightship Server, Sniper's TM/1. Planning Science's Gentium and Kenan Technology's Multiway.
- Some of the problems faced by clients are related to maintaining support to multiple subject areas in an RDBMS.
- Some vendors can solve these problems by continuing access from MOLAP tools to detailed data in and RDBMS.

- This can be very useful for organizations with performance-sensitive multidimensional analysis requirements and that have built or are in the process of building a data warehouse architecture that contains multiple subject areas.
- An example would be the creation of sales data measured by several dimensions (e.g., product and sales region) to be stored and maintained in a persistent structure.
- This structure would be provided to reduce the application overhead of performing calculations and building aggregation during initialization.
- These structures can be automatically refreshed at predetermined intervals established by an administrator.

Advantages

- **Excellent Performance:**
    - A MOLAP cube is built for fast information retrieval, and is optimal for slicing and dicing operations.
- **Can perform complex calculations:**
    - All evaluation have been pre-generated when the cube is created. Hence, complex calculations are not only possible, but they return quickly.

Disadvantages

- **Limited in the amount of information it can handle:**
    - Because all calculations are performed when the cube is built, it is not possible to contain a large amount of data in the cube itself.
- **Requires additional investment:**
    - Cube technology is generally proprietary and does not already exist in the organization. Therefore, to adopt MOLAP technology, chances are other investments in human and capital resources are needed.

3. Hybrid OLAP (HOLAP) Server

- HOLAP incorporates the best features of **MOLAP** and **ROLAP** into a single architecture.

- HOLAP systems save more substantial quantities of detailed data in the relational tables while the aggregations are stored in the pre-calculated cubes.
- HOLAP also can drill through from the cube down to the relational tables for delineated data.
- The **Microsoft SQL Server 2000** provides a hybrid OLAP server as in figure 2.25.

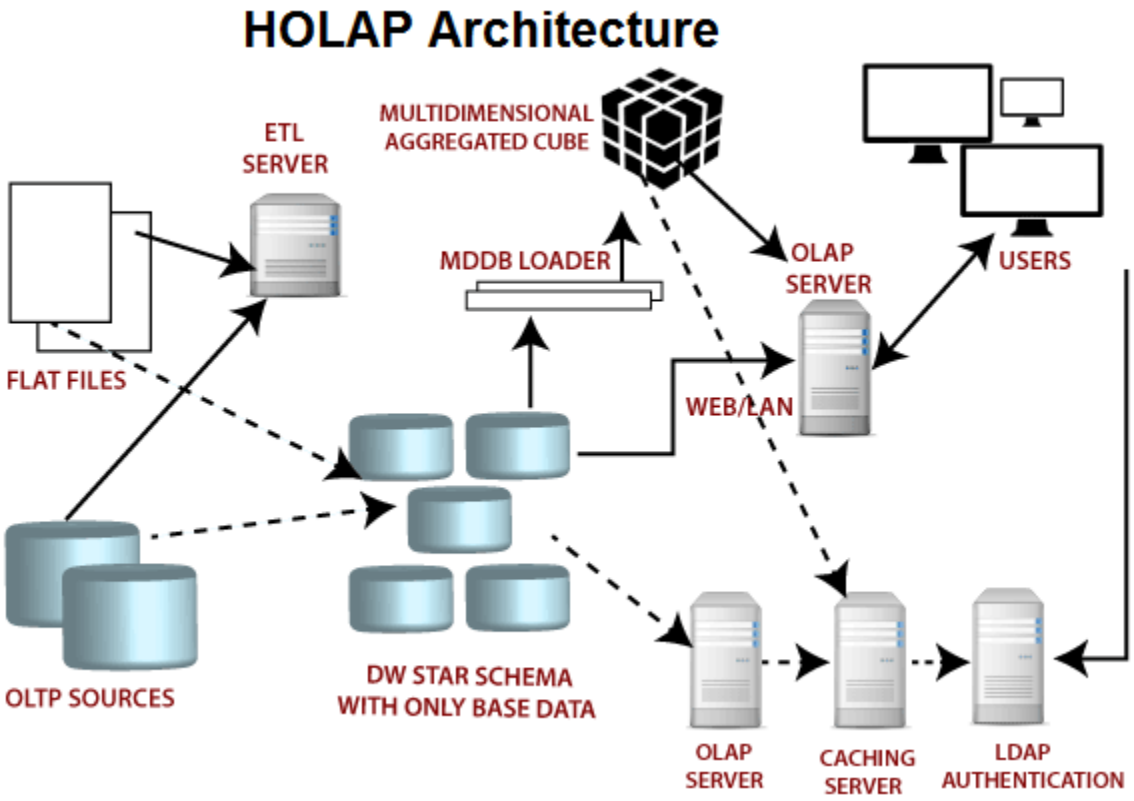


Fig.2.25 HOLAP Architecture

Advantages of HOLAP

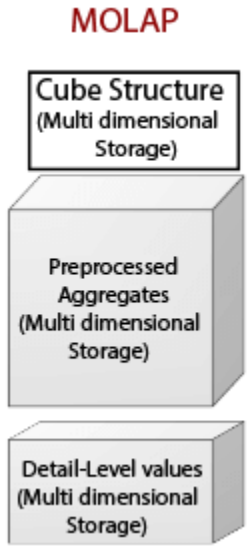
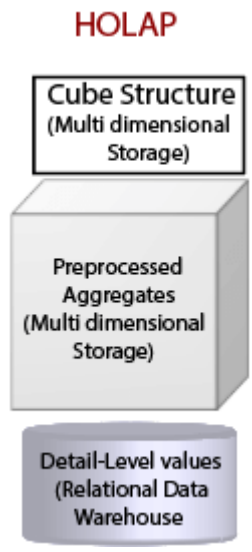
1. HOLAP provide benefits of both MOLAP and ROLAP.
2. It provides fast access at all levels of aggregation.
3. HOLAP balances the disk space requirement, as it only stores the aggregate information on the OLAP server and the detail record remains in the relational database. So no duplicate copy of the detail record is maintained.

Disadvantages of HOLAP

1. HOLAP architecture is very complicated because it supports both MOLAP and ROLAP servers.

15. Explain in detail about ROLAP Vs MOLAP Vs HOLAP. [NOV/DEC 2023]

ROLAP	MOLAP	HOLAP
ROLAP stands for Relational Online Analytical Processing.	MOLAP stands for Multidimensional Online Analytical Processing.	HOLAP stands for Hybrid Online Analytical Processing.
The ROLAP storage mode causes the aggregation of the division to be stored in indexed views in the relational database that was specified in the partition's data source.	The MOLAP storage mode principle the aggregations of the division and a copy of its source information to be saved in a multidimensional operation in analysis services when the separation is processed.	The HOLAP storage mode connects attributes of both MOLAP and ROLAP. Like MOLAP, HOLAP causes the aggregation of the division to be stored in a multidimensional operation in an SQL Server analysis services instance.
ROLAP does not because a copy of the source information to be stored in the Analysis services data folders. Instead, when the outcome cannot be derived from the query cache, the indexed views in the record source are accessed to answer queries.	This MOLAP operation is highly optimize to maximize query performance. The storage area can be on the computer where the partition is described or on another computer running Analysis services. Because a copy of the source information resides in the multidimensional operation, queries can be resolved without accessing the partition's source record.	HOLAP does not causes a copy of the source information to be stored. For queries that access the only summary record in the aggregations of a division, HOLAP is the equivalent of MOLAP.

Query response is frequently slower with ROLAP storage than with the MOLAP or HOLAP storage mode. Processing time is also frequently slower with ROLAP.	Query response times can be reduced substantially by using aggregations. The record in the partition's MOLAP operation is only as current as of the most recent processing of the separation.	Queries that access source record for example, if we want to drill down to an atomic cube cell for which there is no aggregation information must retrieve data from the relational database and will not be as fast as they would be if the source information were stored in the MOLAP architecture.
		

16. Find the major difference between MOLAP and ROLAP. [NOV 2023]

Identify the major difference between MOLAP and ROLAP. [Nov 2024]

S.NO	ROLAP	MOLAP
1.	ROLAP stands for **Relational Online Analytical Processing.**	While MOLAP stands for **Multidimensional Online Analytical Processing.**
2.	ROLAP is used for large data volumes.	While it is used for limited data volumes.
3.	The access of ROLAP is slow.	WHILE the access of MOLAP is fast.
4.	In ROLAP, Data is stored in relation tables.	While in MOLAP, Data is stored in multidimensional array.

5.	In ROLAP, Data is fetched from data-warehouse.	While in MOLAP, Data is fetched from MDDBs database.
6.	In ROLAP, Complicated sql queries are used.	While in MOLAP, Sparse matrix is used.
7.	In ROLAP, Static multidimensional view of data is created.	While in MOLAP, Dynamic multidimensional view of data is created.

17. Compare OLAP with OLTP. **[Nov 2024]**

Difference between OLAP and OLTP.

Category	OLAP (Online Analytical Processing)	OLTP (Online Transaction Processing)
Definition	It is well-known as an online database query management system.	It is well-known as an online database modifying system.
Data source	Consists of historical data from various Databases.	Consists of only operational current data.
Method used	It makes use of a data warehouse.	It makes use of a standard database management system (DBMS).
Application	It is subject-oriented. Used for Data Mining, Analytics, Decisions making, etc.	It is application-oriented. Used for business tasks.
Normalized	In an OLAP database, tables are not normalized.	In an OLTP database, tables are normalized (3NF).
Usage of data	The data is used in planning, problem-solving, and decision-making.	The data is used to perform day-to-day fundamental operations.
Task	It provides a multi-dimensional view of different business tasks.	It reveals a snapshot of present business tasks.

Category	OLAP (Online Analytical Processing)	OLTP (Online Transaction Processing)
Purpose	It serves the purpose to extract information for analysis and decision-making.	It serves the purpose to Insert, Update, and Delete information from the database.
Volume of data	A large amount of data is stored typically in TB, PB	The size of the data is relatively small as the historical data is archived in MB, and GB.
Queries	Relatively slow as the amount of data involved is large. Queries may take hours.	Very Fast as the queries operate on 5% of the data.
Update	The OLAP database is not often updated. As a result, data integrity is unaffected.	The data integrity constraint must be maintained in an OLTP database.
Backup and Recovery	It only needs backup from time to time as compared to OLTP.	The backup and recovery process is maintained rigorously
Processing time	The processing of complex queries can take a lengthy time.	It is comparatively fast in processing because of simple and straightforward queries.
Types of users	This data is generally managed by CEO, MD, and GM.	This data is managed by clerksForex and managers.
Operations	Only read and rarely write operations.	Both read and write operations.
Updates	With lengthy, scheduled batch operations, data is refreshed on a regular basis.	The user initiates data updates, which are brief and quick.

Category	OLAP (Online Analytical Processing)	OLTP (Online Transaction Processing)
Nature of audience	The process is focused on the customer.	The process is focused on the market.
Database Design	Design with a focus on the subject.	Design that is focused on the application.
Productivity	Improves the efficiency of business analysts.	Enhances the user's productivity.

UNIT III META DATA, DATA MART AND PARTITION STRATEGY 7

Meta Data – Categories of Metadata – Role of Metadata – Metadata Repository – Challenges for Meta Management - Data Mart – Need of Data Mart- Cost Effective Data Mart- Designing Data Marts- Cost of Data Marts- Partitioning Strategy – Vertical partition – Normalization – Row Splitting – Horizontal Partition

PART A

1. **Compare Data Mart with Data Warehouse. [Nov 2024]**
   - A data mart is a narrow database, in the sense that it stores data relating to a particular department or aspect of the business.
   - Each department or business area can have its own data mart. A data warehouse, on the other hand, stores data linked to the entire company and to any aspect of the business activity.
2. **Why is data mart considered cost effective compared to a data warehouse?**
   - Data Marts are cost-effective because they are made to store a particular subset, which lowers data storage costs. Because Data Marts require less work than the entire warehouse, they are also more cost-effective in terms of design and maintenance.
   - It stores a smaller, more focused subset of data relevant to a specific department or business function, resulting in lower storage requirements, simpler implementation, and reduced maintenance costs compared to a large, enterprise-wide data warehouse that needs to handle vast amounts of data across multiple departments.
3. **What is Metadata?**
   - Metadata is simply defined as data about data. The data that is used to represent other data is known as metadata.
   - For example, the index of a book serves as a metadata for the contents in the book. In other words, we can say that metadata is the summarized data that leads us to detailed data.

4. What are the Categories of Metadata?

Metadata can be broadly categorized into three categories −

- Business Metadata
- Technical Metadata
- Operational Metadata

5. What are the roles of Meta Data?

The various roles of metadata are explained below.

- Metadata acts as a directory.
- This directory helps the decision support system to locate the contents of the data warehouse.
- Metadata helps in decision support system for mapping of data when data is transformed from operational environment to data warehouse environment.
- Metadata helps in summarization between current detailed data and highly summarized data.
- Metadata also helps in summarization between lightly detailed data and highly summarized data.
- Metadata is used for query tools.
- Metadata is used in extraction and cleansing tools.
- Metadata is used in reporting tools.
- Metadata is used in transformation tools.
- Metadata plays an important role in loading functions.

6. What is meant Metadata Repository?

- Metadata repository is an integral part of a data warehouse system. It has the following metadata
  - Definition of data warehouse
  - Business metadata
  - Operational Metadata
  - Data for mapping from operational environment to data warehouse
  - Algorithms for summarization

7. Define Meta Data in terms of Data Warehouse.

- In terms of data warehouse, we can define metadata as follows.
  - Metadata is the road-map to a data warehouse.
  - Metadata in a data warehouse defines the warehouse objects.
  - Metadata acts as a directory. This directory helps the decision support system to locate the contents of a data warehouse.

8. What are the importance of Meta Data?

- Metadata helps in driving the accuracy of reports, validates data transformation, and ensures the accuracy of calculations.
- Metadata also enforces the definition of business terms to business end-users. With all these uses of metadata, it also has its challenges.

9. What are the challenges of Meta Data?

- Metadata in a big organization is scattered across the organization. This metadata is spread in spreadsheets, databases, and applications.
- Metadata could be present in text files or multimedia files. To use this data for information management solutions, it has to be correctly defined.
- There are no industry-wide accepted standards. Data management solution vendors have narrow focus.
- There are no easy and accepted methods of passing metadata.

10. Why Do We Need a Data Mart?

- To partition data in order to impose access control strategies.
- To speed up the queries by reducing the volume of data to be scanned.
- To segment data into different hardware platforms.
- To structure data in a form suitable for a user access tool.

11. What are the steps to make the data mart is cost effective?

Follow the steps given below to make data marting cost-effective

- Identify the Functional Splits
- Identify User Access Tool Requirements
- Identify Access Control Issues

12. What are the issues to be taken into account while determining the functional split?

- The structure of the department may change.
- The products might switch from one department to other.
- The merchant could query the sales trend of other products to analyze what is happening to the sales.

13. What are the cost measures of Data Mart?

The cost measures for data marting are as follows

- Hardware and Software Cost
- Network Access
- Time Window Constraints

14. Why is the Data Warehouse Necessary to Partition?

Partitioning is important for the following reasons

- For easy management
- To assist backup/recovery
- To enhance performance

15. Differentiate metadata and data mart. [NOV/DEC 2023]

Key	Metadata	Data Mart
Definition	Metadata describes the characteristics, usage, and structure of data.	A data mart is a subset of a data warehouse focused on specific business needs.
Purpose	It provides context and information about data to facilitate understanding, management, and governance.	It stores and manages data tailored for specific departments or projects, optimized for analysis and reporting.

Scope	Metadata covers all data within an organization.	Data mart contains selected data relevant to particular users or business units.
Example	File descriptions, database schemas, data dictionaries.	Sales data mart, finance data mart, marketing data mart.

16. Propose the features of Metadata repository in data warehousing.

[NOV 2023]

- **Integration** of the metadata across the organization
- Build relationship between various **metadata types**
- Build relationship between various **disparate systems**
- **Define business** golden copy of definitions
- **Version** control of the changes at structure level
- Interaction with **Reference data**
- Link view to **master data**

PART B

1. Explain in detail about the Meta Data.

Metadata

- Metadata is simply defined as data about data. The data that is used to represent other data is known as metadata.
- For example, the index of a book serves as a metadata for the contents in the book. In other words, we can say that metadata is the summarized data that leads us to detailed data.
- In terms of data warehouse, we can define metadata as follows.
    - Metadata is the road-map to a data warehouse.
    - Metadata in a data warehouse defines the warehouse objects.
    - Metadata acts as a directory. This directory helps the decision support system to locate the contents of a data warehouse.
    - In a data warehouse, we create metadata for the data names and definitions of a given data warehouse.
    - Along with this metadata, additional metadata is also created for time-stamping any extracted data, the source of extracted data.

Several examples of metadata are:

1. A library catalog may be considered metadata. The directory metadata consists of several predefined components representing specific attributes of a resource, and each item can have one or more values. These components could be the name of the author, the name of the document, the publisher's name, the publication date, and the methods to which it belongs.
2. The table of content and the index in a book may be treated metadata for the book.
3. Suppose we say that a data item about a person is 80. This must be defined by noting that it is the person's weight and the unit is kilograms. Therefore, (weight, kilograms) is the metadata about the data is 80.
4. Another examples of metadata are data about the tables and figures in a report like this book. A table (which is a record) has a name (e.g., table titles), and there

are column names of the tables that may be treated metadata. The figures also have titles or names.

Why is metadata necessary in a data warehouses?

- First, it acts as the glue that links all parts of the data warehouses.
- Next, it provides information about the contents and structures to the developers.
- Finally, it opens the doors to the end-users and makes the contents recognizable in their terms.
- Metadata is Like a Nerve Center. Various processes during the building and administering of the data warehouse generate parts of the data warehouse metadata.
- Another uses parts of metadata generated by one process.
- In the data warehouse, metadata assumes a key position and enables communication among various methods.
- It acts as a nerve center in the data warehouse as in figure 3.1.

Fig.3.1 Location of metadata within the data warehouse

2. Explain in detail about Meta Data. (or) Illustrate the various classification of Meta data with suitable examples and explain the same. [NOV 2023]

- Metadata is data about the data or documentation about the information which is required by the users. In data warehousing, metadata is one of the essential aspects.
- Metadata is used for building, maintaining, managing, and using the data warehouses. Metadata allow users access to help understand the content and find data.

Metadata includes the following:

1. The location and descriptions of warehouse systems and components.
2. Names, definitions, structures, and content of data-warehouse and end-users views.
3. Identification of authoritative data sources.
4. Integration and transformation rules used to populate data.
5. Integration and transformation rules used to deliver information to end-user analytical tools.
6. Subscription information for information delivery to analysis subscribers.
7. Metrics used to analyze warehouses usage and performance.
8. Security authorizations, access control list, etc.

Types of Metadata

Metadata in a data warehouse fall into three major parts:

a) Operational Metadata

b) Extraction and Transformation Metadata

c) End-User Metadata

a) Operational Metadata

- Data for the data warehouse comes from various operational systems of the enterprise. These source systems include different data structures.

- The data elements selected for the data warehouse have various fields lengths and data types.
- In selecting information from the source systems for the data warehouses, we divide records, combine factor of documents from different source files, and deal with multiple coding schemes and field lengths.
- When we deliver information to the end-users, we must be able to tie that back to the source data sets.
- Operational metadata contains all of this information about the operational data sources.

b) Extraction and Transformation Metadata

- Extraction and transformation metadata include data about the removal of data from the source systems, namely, the extraction frequencies, extraction methods, and business rules for the data extraction.
- Category of metadata contains information about all the data transformation that takes place in the data staging area.

c) End-User Metadata

- The end-user metadata is the navigational map of the data warehouses. It enables the end-users to find data from the data warehouses.
- The end-user metadata allows the end-users to use their business terminology and look for the information in those ways in which they usually think of the business.

Metadata Interchange Initiative

- The metadata interchange initiative was proposed to bring industry vendors and user together to address a variety of severe problems and issues concerning exchanging, sharing, and managing metadata.
- The goal of metadata interchange standard is to define an extensible mechanism that will allow the vendor to exchange standard metadata as well as carry along "proprietary" metadata.

- The founding members agreed on the following initial goals:
  1. Creating a vendor-independent, industry-defined, and maintained standard access mechanisms and application programming interfaces (API) for metadata.
  2. Enabling users to control and manage the access and manipulation of metadata in their unique environment through the use of interchange standards-compliant tools.
  3. Users are allowed to build tools that meet their needs and also will enable them to adjust accordingly to those tools configurations.
  4. Allowing individual tools to satisfy their metadata requirements freely and efficiently within the content of an interchange model.
  5. Describing a simple, clean implementation infrastructure which will facilitate compliance and speed up adoption by minimizing the amount of modification.
  6. To create a procedure and process not only for maintaining and establishing the interchange standard specification but also for updating and extending it over time.

Metadata Interchange Standard Framework

- Interchange standard metadata model implementation assumes that the metadata itself may be stored in storage format of any type:
  - ASCII files, relational tables, fixed or customized formats, etc.
- It is a framework that is based on a framework that will translate an access request into the standard interchange index.

Several approaches have been proposed in metadata interchange coalition:

- Procedural Approach
- ASCII Batch Approach
- Hybrid Approach

- In a **procedural approach**, the communication with API is built into the tool. It enables the highest degree of flexibility.

- In **ASCII Batch approach**, instead of relying on ASCII file format which contains information of various metadata items and standardized access requirements that make up the interchange standards metadata model.
- In the **Hybrid approach**, it follows a data-driven model.

Components of Metadata Interchange Standard Frameworks

- Refer figure 3.2

1) Standard Metadata Model**:** It refers to the ASCII file format, which is used to represent metadata that is being exchanged.

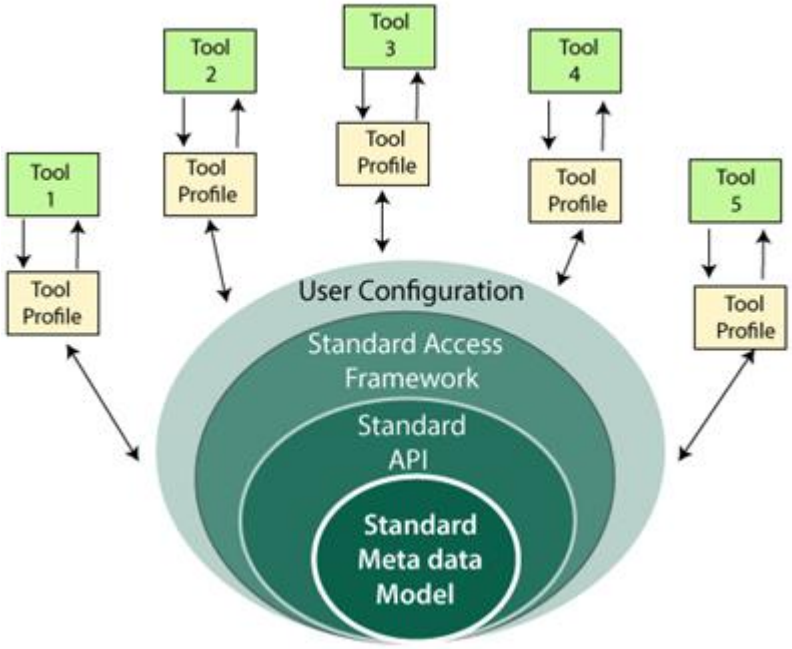


Fig.3.2 MetaData Interchange Standard Framework

2) The standard access framework that describes the minimum number of API functions.

3) Tool profile**,** which is provided by each tool vendor.

4) The user configuration is a file explaining the legal interchange paths for metadata in the user's environment.

Metadata Repository

- The metadata itself is housed in and controlled by the metadata repository.
- The software of metadata repository management can be used to map the source data to the target database, integrate and transform the data, generate code for data transformation, and to move data to the warehouse.

Benefits of Metadata Repository

1. It provides a set of tools for enterprise-wide metadata management.
2. It eliminates and reduces inconsistency, redundancy, and underutilization.
3. It improves organization control, simplifies management, and accounting of information assets.
4. It increases coordination, understanding, identification, and utilization of information assets.
5. It enforces CASE development standards with the ability to share and reuse metadata.
6. It leverages investment in legacy systems and utilizes existing applications.
7. It provides a relational model for heterogeneous RDBMS to share information.
8. It gives useful data administration tool to manage corporate information assets with the data dictionary.
9. It increases reliability, control, and flexibility of the application development process.

3. Explain in detail about Categories of Meta Data

Categories of Metadata

Metadata can be broadly categorized into three categories as in figure 3.3

- **Business Metadata –** It has the data ownership information, business definition, and changing policies.

- **Technical Metadata** – It includes database system names, table and column names and sizes, data types and allowed values. Technical metadata also includes structural information such as primary and foreign key attributes and indices.
- **Operational Metadata** – It includes currency of data and data lineage. Currency of data means whether the data is active, archived, or purged. Lineage of data means the history of data migrated and transformation applied on it.

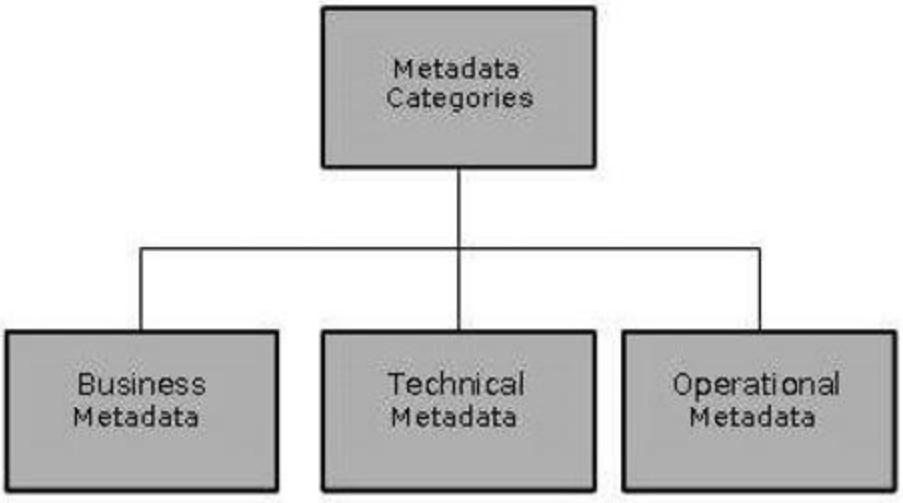


Fig.3.3 Categorization of Meta Data

4. Explain in detail about Role of Meta Data. [Nov 2024]

Role of Metadata

- The role of metadata in a warehouse is different from the warehouse data, yet it plays an important role.
- The various roles of metadata are explained below as in figure 3.4.
  - Metadata acts as a directory.+
  - This directory helps the decision support system to locate the contents of the data warehouse.
  - Metadata helps in decision support system for mapping of data when data is transformed from operational environment to data warehouse environment.

- Metadata helps in summarization between current detailed data and highly summarized data.
- Metadata also helps in summarization between lightly detailed data and highly summarized data.
- Metadata is used for query tools.
- Metadata is used in extraction and cleansing tools.
- Metadata is used in reporting tools.
- Metadata is used in transformation tools.
- Metadata plays an important role in loading functions.

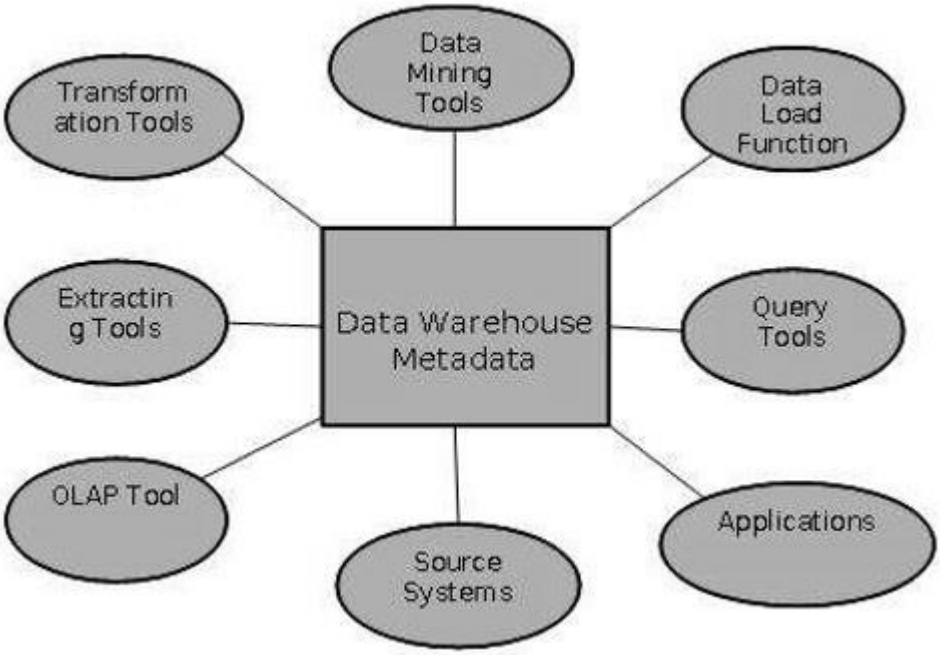


Fig.3.4 Roles of Meta Data

5. Explain in detail about Meta Data Repository. [Nov 2024]

Metadata Repository

- Metadata repository is an integral part of a data warehouse system. It has the following metadata
- **Definition of data warehouse** − It includes the description of structure of data warehouse. The description is defined by schema, view, hierarchies, derived data definitions, and data mart locations and contents.

- **Business metadata** – It contains has the data ownership information, business definition, and changing policies.
- **Operational Metadata** – It includes currency of data and data lineage. Currency of data means whether the data is active, archived, or purged. Lineage of data means the history of data migrated and transformation applied on it.
- **Data for mapping from operational environment to data warehouse** – It includes the source databases and their contents, data extraction, data partition cleaning, transformation rules, data refresh and purging rules.
- **Algorithms for summarization** – It includes dimension algorithms, data on granularity, aggregation, summarizing, etc.

6. Describe the challenges of metadata management. [Nov 2024]

Challenge 1: Increased data volumes.

Challenge 2: New roles for analytics.

Challenge 3: Compliance requirements.

Clearly identify your business goals.

Focus on the quality of data.

Allow the right people to access the data.

Prioritize data security.

Challenges for Metadata Management

- Metadata management can face challenges such as data quality, data security, and data governance.

Data quality

- Poorly structured metadata can make it difficult to understand and use data
- Incorrect metadata can lead to data quality issues

Data security

- Protecting sensitive data is a challenge, especially when working with confidential information
- Organizations must adhere to privacy regulations, encrypt data, and implement access controls

Data governance

- Establishing and enforcing metadata standards and policies can be difficult

- Ensuring that metadata management initiatives are in line with business strategy and goals can be challenging

Data integration

- Ensuring that metadata is consistent and aligned across different sources can be difficult

7. Explain in detail about Data Mart.

Data Mart

- A Data Mart is a subset of a directorial information store, generally oriented to a specific purpose or primary data subject which may be distributed to provide business needs.
- Data Marts are analytical record stores designed to focus on particular business functions for a specific community within an organization.
- Data marts are derived from subsets of data in a data warehouse, though in the bottom-up data warehouse design methodology, the data warehouse is created from the union of organizational data marts as in figure 3.5.
- The fundamental use of a data mart is Business Intelligence (BI) applications. BI is used to gather, store, access, and analyze record.
- It can be used by smaller businesses to utilize the data they have accumulated since it is less expensive than implementing a data warehouse.

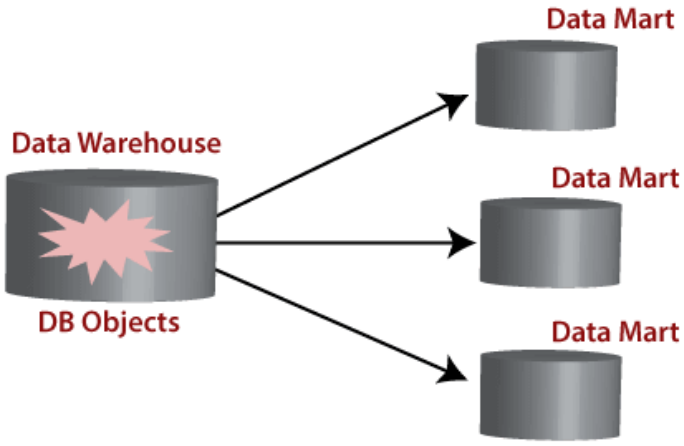


Fig.3.5 Different sets of Data Mart

Reasons for creating a data mart

- Creates collective data by a group of users
- Easy access to frequently needed data
- Ease of creation
- Improves end-user response time
- Lower cost than implementing a complete data warehouses
- Potential clients are more clearly defined than in a comprehensive data warehouse
- It contains only essential business data and is less cluttered.

Types of Data Marts

There are mainly two approaches to designing data marts. These approaches are

- Dependent Data Marts
- Independent Data Marts

Dependent Data Marts

- A dependent data marts is a logical subset of a physical subset of a higher data warehouse. Refer figure 3.6
- According to this technique, the data marts are treated as the subsets of a data warehouse.
- In this technique, firstly a data warehouse is created from which further various data marts can be created.
- This data mart is dependent on the data warehouse and extract the essential record from it. In this technique, as the data warehouse creates the data mart; therefore, there is no need for data mart integration. It is also known as a top-down approach.

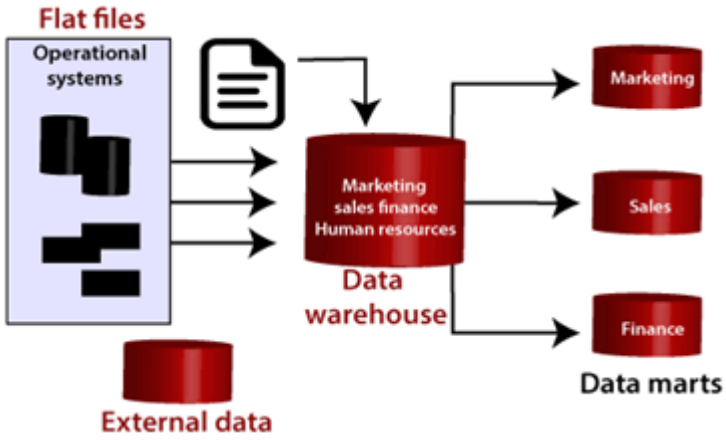


Fig.3.6 Dependent Data Mart

Independent Data Marts

- The second approach is Independent data marts (IDM) Here, firstly independent data marts are created, and then a data warehouse is designed using these independent multiple data marts. Refer figure 3.7
- In this approach, as all the data marts are designed independently; therefore, the integration of data marts is required.
- It is also termed as a bottom-up approach as the data marts are integrated to develop a data warehouse.

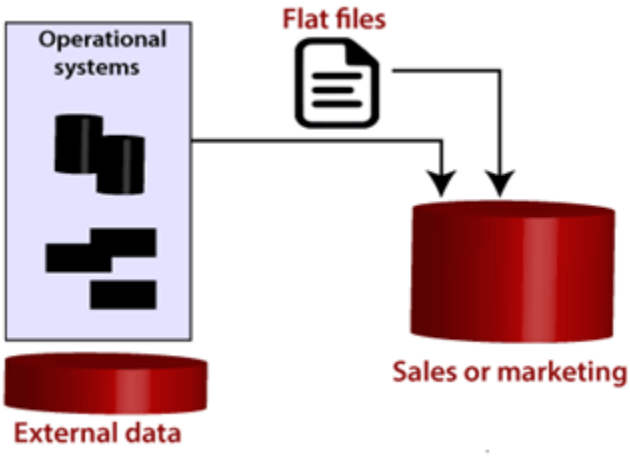


Fig.3.7 Independent Data Mart

Steps in Implementing a Data Mart

- The significant steps in implementing a data mart are to design the schema, construct the physical storage, populate the data mart with data from source systems, access it to make informed decisions and manage it over time. So, the steps are:

a) Designing

- The design step is the first in the data mart process. This phase covers all of the functions from initiating the request for a data mart through gathering data about the requirements and developing the logical and physical design of the data mart.

It involves the following tasks:

1. Gathering the business and technical requirements
2. Identifying data sources
3. Selecting the appropriate subset of data
4. Designing the logical and physical architecture of the data mart.

b) Constructing

- This step contains creating the physical database and logical structures associated with the data mart to provide fast and efficient access to the data.

It involves the following tasks:

1. Creating the physical database and logical structures such as tablespaces associated with the data mart.
2. creating the schema objects such as tables and indexes describe in the design step.
3. Determining how best to set up the tables and access structures.

c) Populating

- This step includes all of the tasks related to the getting data from the source, cleaning it up, modifying it to the right format and level of detail, and moving it into the data mart.

It involves the following tasks:

1. Mapping data sources to target data sources
2. Extracting data
3. Cleansing and transforming the information.
4. Loading data into the data mart
5. Creating and storing metadata

d) Accessing

- This step involves putting the data to use: querying the data, analyzing it, creating reports, charts and graphs and publishing them.

  It involves the following tasks:

  1. Set up and intermediate layer (Meta Layer) for the front-end tool to use. This layer translates database operations and objects names into business conditions so that the end-clients can interact with the data mart using words which relates to the business functions.
  2. Set up and manage database architectures like summarized tables which help queries agree through the front-end tools execute rapidly and efficiently.

e) Managing

- This step contains managing the data mart over its lifetime. In this step, management functions are performed as:

  1. Providing secure access to the data.
  2. Managing the growth of the data.
  3. Optimizing the system for better performance.
  4. Ensuring the availability of data event with system failures.

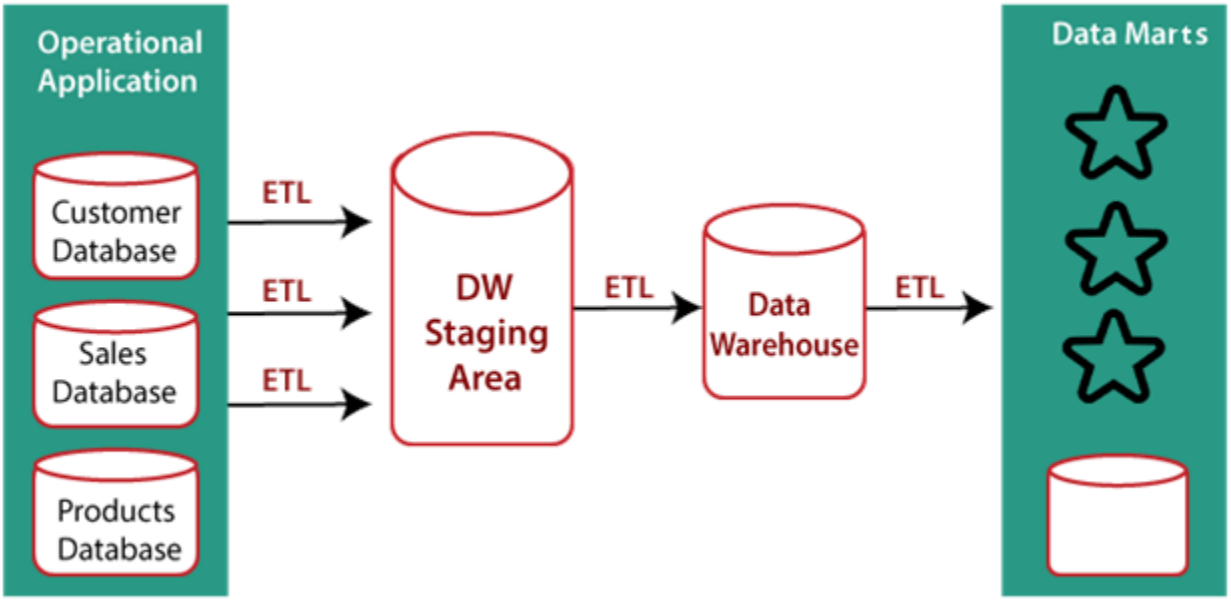


Fig.3.8 Difference between Data Warehouse and Data Mart

8. Explain in detail about Data warehouse and Data Mart.

Data Warehouse	Data Mart
A Data Warehouse is a vast repository of information collected from various organizations or departments within a corporation.	A data mart is an only subtype of a Data Warehouses. It is architecture to meet the requirement of a specific user group.
It may hold multiple subject areas.	It holds only one subject area. For example, Finance or Sales.
It holds very detailed information.	It may hold more summarized data.

Works to integrate all data sources	It concentrates on integrating data from a given subject area or set of source systems.
In data warehousing, Fact constellation is used.	In Data Mart, Star Schema and Snowflake Schema are used.
It is a Centralized System. It is a Decentralized System.	
Data Warehousing is the data-oriented.	Data Marts is a project-oriented.

9. Explain in detail about Need of Data Mart, Cost effective Data Mart, Designing Data Mart, Cost of Data Marts.

Need of Data Mart

- To partition data in order to impose access control strategies.
- To speed up the queries by reducing the volume of data to be scanned.
- To segment data into different hardware platforms.
- To structure data in a form suitable for a user access tool.

Cost-effective Data Marting

- Identify the Functional Splits
- Identify User Access Tool Requirements
- Identify Access Control Issues
- Identify the Functional Splits

Consider a retail organization, where each merchant is accountable for maximizing the sales of a group of products.

For this, the following are the valuable information −

- sales transaction on a daily basis
- sales forecast on a weekly basis

- stock position on a daily basis
- stock movements on a daily basis

As the merchant is not interested in the products they are not dealing with, the data marting is a subset of the data dealing which the product group of interest.

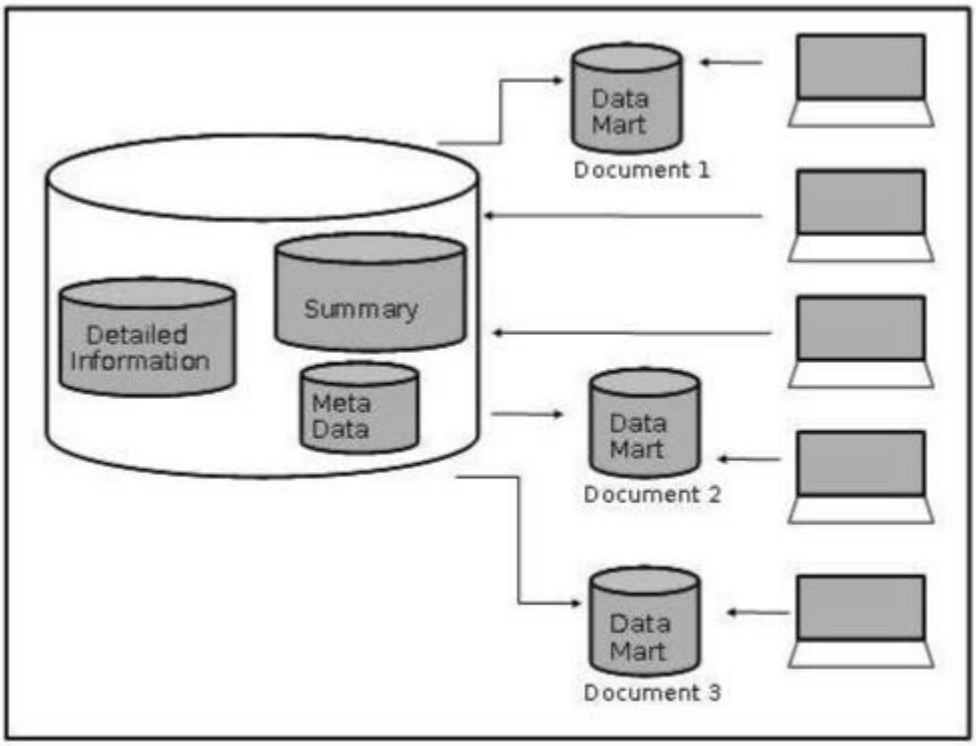


Fig.3.9 Data marting for different users

Issues to be taken into account while determining the functional split −

- The structure of the department may change.
- The products might switch from one department to other.
- The merchant could query the sales trend of other products to analyze what is happening to the sales.

Identify User Access Tool Requirements

- We need data marts to support user access tools that require internal data structures. The data in such structures are outside the control of data warehouse but need to be populated and updated on a regular basis.

- There are some tools that populate directly from the source system but some cannot. Therefore additional requirements outside the scope of the tool are needed to be identified for future.
- In order to ensure consistency of data across all access tools, the data should not be directly populated from the data warehouse, rather each tool must have its own data mart.

Identify Access Control Issues

- There should to be privacy rules to ensure the data is accessed by authorized users only.
- For example a data warehouse for retail banking institution ensures that all the accounts belong to the same legal entity.
- Privacy laws can force you to totally prevent access to information that is not owned by the specific bank.
- Data marts allow us to build a complete wall by physically separating data segments within the data warehouse.
- To avoid possible privacy problems, the detailed data can be removed from the data warehouse.
- We can create data mart for each legal entity and load it via data warehouse, with detailed account data.

Designing Data Marts

- Data marts should be designed as a smaller version of starflake schema within the data warehouse and should match with the database design of the data warehouse in figure 3.10.
- It helps in maintaining control over database instances.

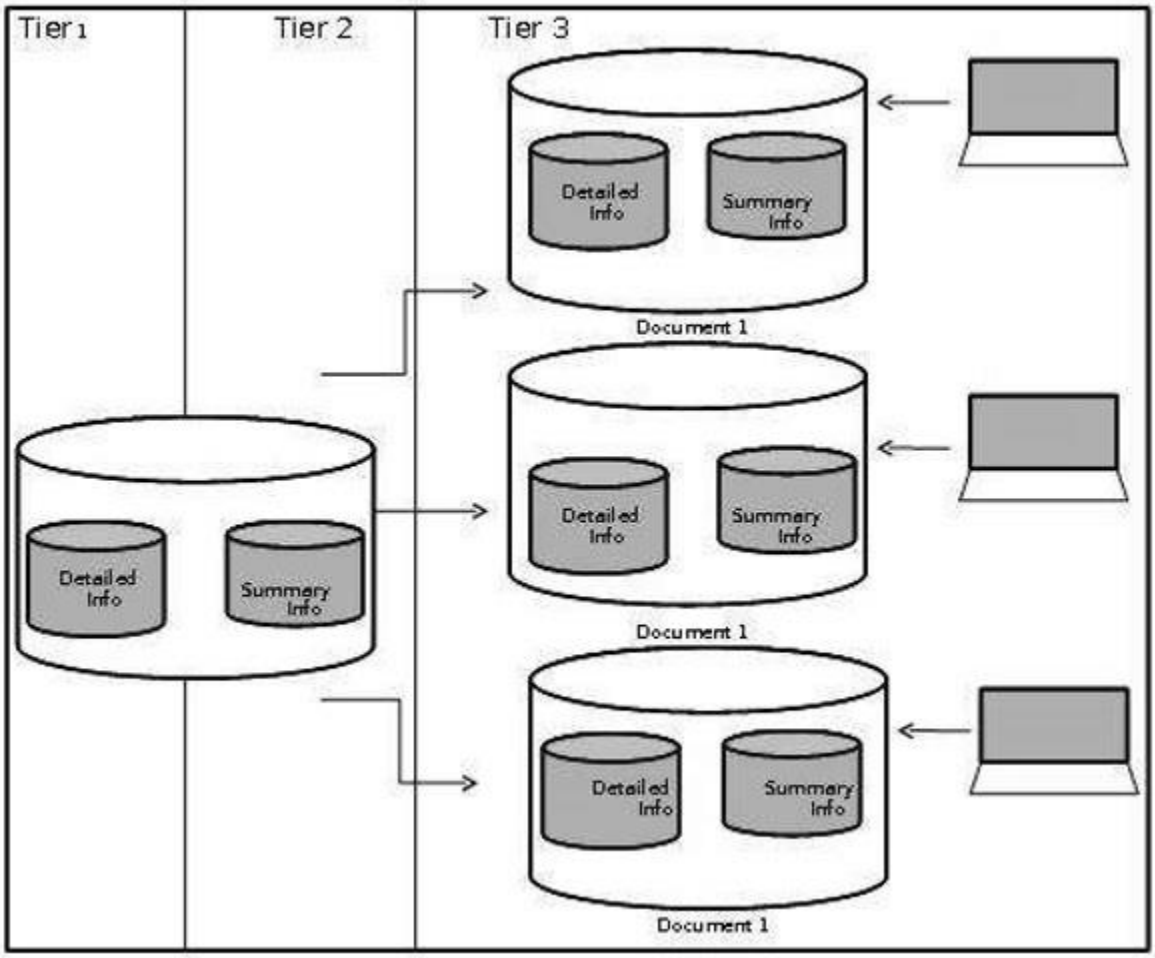


Fig.3.10 Design of Data Mart

- The summaries are data mart in the same way as they would have been designed within the data warehouse.
- Summary tables help to utilize all dimension data in the starflake schema.

Cost of Data Marting

The cost measures for data marting are as follows −

- Hardware and Software Cost
- Network Access
- Time Window Constraints
- Hardware and Software Cost
- Although data marts are created on the same hardware, they require some additional hardware and software.

- To handle user queries, it requires additional processing power and disk storage.
- If detailed data and the data mart exist within the data warehouse, then we would face additional cost to store and manage replicated data.

Note − Data marting is more expensive than aggregations, therefore it should be used as an additional strategy and not as an alternative strategy.

Network Access

- A data mart could be on a different location from the data warehouse, so we should ensure that the LAN or WAN has the capacity to handle the data volumes being transferred within the data mart load process.

Time Window Constraints

- The extent to which a data mart loading process will eat into the available time window depends on the complexity of the transformations and the data volumes being shipped.
- The determination of data marts is possible depends on
    - Network capacity.
    - Time window available
    - Volume of data being transferred
    - Mechanisms being used to insert data into a data mart

10. Explain in detail about Data Warehousing – Partitioning Strategy.

Data Warehousing - Partitioning Strategy

- Partitioning is done to enhance performance and facilitate easy management of data.
- Partitioning also helps in balancing the various requirements of the system.
- It optimizes the hardware performance and simplifies the management of data warehouse by partitioning each fact table into multiple separate partitions.
- In this chapter, we will discuss different partitioning strategies.

Why is it Necessary to Partition?

Partitioning is important for the following reasons −

a) For easy management,

b) To assist backup/recovery,

c) To enhance performance.

a) For Easy Management

- The fact table in a data warehouse can grow up to hundreds of gigabytes in size. This huge size of fact table is very hard to manage as a single entity. Therefore it needs partitioning.

b) To Assist Backup/Recovery

- If we do not partition the fact table, then we have to load the complete fact table with all the data.
- Partitioning allows us to load only as much data as is required on a regular basis. It reduces the time to load and also enhances the performance of the system.

c) To Enhance Performance

- By partitioning the fact table into sets of data, the query procedures can be enhanced.
- Query performance is enhanced because now the query scans only those partitions that are relevant. It does not have to scan the whole data.

Horizontal Partitioning

- There are various ways in which a fact table can be partitioned.
- In horizontal partitioning, we have to keep in mind the requirements for manageability of the data warehouse.

Partitioning by Time into Equal Segments

- In this partitioning strategy, the fact table is partitioned on the basis of time period. Here each time period represents a significant retention period within the business.
- For example, if the user queries for **month to date data** then it is appropriate to partition the data into monthly segments. We can reuse the partitioned tables by removing the data in them.

Partition by Time into Different-sized Segments

- This kind of partition is done where the aged data is accessed infrequently as in figure 3.11.
- It is implemented as a set of small partitions for relatively current data, larger partition for inactive data.

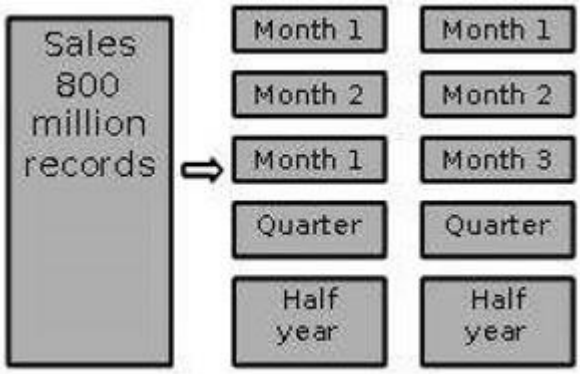


Fig.3.11 Horizontal Partitioning of Data

Partition on a Different Dimension

- The fact table can also be partitioned on the basis of dimensions other than time such as product group, region, supplier, or any other dimension. Let's have an example.
- Suppose a market function has been structured into distinct regional departments like on a **state by state** basis. If each region wants to query on information captured within its region, it would prove to be more effective to partition the fact table into regional partitions. This will cause the queries to speed up because it does not require to scan information that is not relevant.
- The query does not have to scan irrelevant data which speeds up the query process.
- This technique is not appropriate where the dimensions are unlikely to change in future. So, it is worth determining that the dimension does not change in future.
- If the dimension changes, then the entire fact table would have to be repartitioned.

Partition by Size of Table

- When there are no clear basis for partitioning the fact table on any dimension, then we should **partition the fact table on the basis of their size.**
- We can set the predetermined size as a critical point. When the table exceeds the predetermined size, a new table partition is created.
- This partitioning is complex to manage.
- It requires metadata to identify what data is stored in each partition.

Partitioning Dimensions

- If a dimension contains large number of entries, then it is required to partition the dimensions. Here we have to check the size of a dimension.
- Consider a large design that changes over time. If we need to store all the variations in order to apply comparisons, that dimension may be very large. This would definitely affect the response time.

Round Robin Partitions

- In the round robin technique, when a new partition is needed, the old one is archived. It uses metadata to allow user access tool to refer to the correct table partition.
- This technique makes it easy to automate table management facilities within the data warehouse.

Vertical Partition

- Vertical partitioning, splits the data vertically. The following figure 3.12 depicts how vertical partitioning is done.

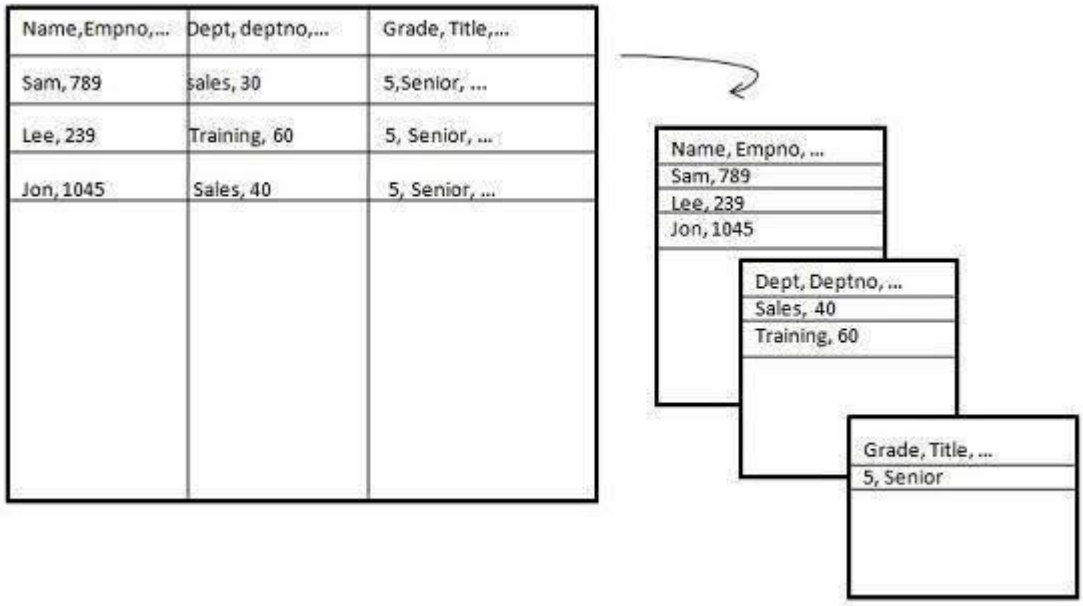


Fig.3.12 Vertical Partitioning

Vertical partitioning can be performed in the following two ways

- Normalization
- Row Splitting

Normalization

- Normalization is the standard relational method of database organization.
- In this method, the rows are collapsed into a single row, hence it reduce space.

Table before Normalization

Product_id	Qty	Value	sales_date	Store_id	Store_name	Location	Region
30	5	3.67	3-Aug-13	16	sunny	Bangalore	S
35	4	5.33	3-Sep-13	16	sunny	Bangalore	S

40	5	2.50	3-Sep-13	64	san	Mumbai	W
45	7	5.66	3-Sep-13	16	sunny	Bangalore	S

Table after Normalization

Store_id	Store_name	Location	Region
16	sunny	Bangalore	W
64	san	Mumbai	S

Product_id	Quantity	Value	sales_date	Store_id
30	5	3.67	3-Aug-13	16
35	4	5.33	3-Sep-13	16
40	5	2.50	3-Sep-13	64
45	7	5.66	3-Sep-13	16

Row Splitting

- Row splitting tends to leave a one-to-one map between partitions. The motive of row splitting is to speed up the access to large table by reducing its size.
- While using vertical partitioning, make sure that there is no requirement to perform a major join operation between two partitions.

Identify Key to Partition

- It is very crucial to choose the right partition key. Choosing a wrong partition key will lead to reorganizing the fact table. Let's have an example. Suppose we want to partition the following table.

  Account_Txn_Table

  transaction_id

  account_id

  transaction_type

  value

  transaction_date

  region

  branch_name

  We can choose to partition on any key.

  The two possible keys could be

  - region
  - transaction_date

- Suppose the business is organized in 30 geographical regions and each region has different number of branches.
- That will give us 30 partitions, which is reasonable.
- This partitioning is good enough because our requirements capture has shown that a vast majority of queries are restricted to the user's own business region.
- If we partition by transaction_date instead of region, then the latest transaction from every region will be in one partition.
- Now the user who wants to look at data within his own region has to query across multiple partitions.
- Hence it is worth determining the right partitioning key.

11. Elaborate in detail about the various issues to be considered when designing and implementing a data warehousing environment. [NOV/DEC 2023]

Here are some of the **difficulties** of **Implementing Data Warehouses**:

1. Implementing a data warehouse is generally a massive effort that must be planned and executed according to established methods.
2. Construction, administration, and quality control are the significant operational issues which arises with data warehousing.
3. Some of the important and challenging consideration while implementing data warehouse are: the design, construction and implementation of the warehouse.
4. The building of an enterprise-wide warehouse in a large organization is a major undertaking.
5. Manual Data Processing can risk the correctness of the data being entered.
6. An intensive enterprise is the administration of a data warehouse, which is proportional to the complexity and size of the warehouse.
7. The complex nature of the administration should be understood by an organization that attempts to administer a data warehouse.
8. There must be a flexibility to accept and integrate analytics to streamline the business intelligence process.
9. To handle the evolutions, acquisition component and the warehouse's schema should be updated.
10. A significant issue in data warehousing is the quality control of data. The major concerns are: quality and consistency of data.

Some best practices for implementing a Data Warehouse:

- The data warehouse must be built incrementally.
- User expectations about he completed projects should be managed.
- It is important to be politically aware.
- There should be a build in adaptability.
- Developing a business/supplier relationship is the best practice.

Implementing a data warehouse can be a complex and challenging process that involves several difficulties, including:

- **Data Integration:** Data warehouses are designed to integrate data from various sources, which can be a complex process. The data may be stored in

different formats, have different levels of granularity, or use different data models. Integrating this data into a cohesive and consistent data warehouse can be challenging.

- **Data Quality:** Data quality is crucial for the success of a data warehouse. Poor data quality can lead to inaccurate or incomplete analyses, which can have significant impacts on business decisions. Ensuring data quality requires careful data cleaning and validation, which can be time-consuming and challenging.
- **Data Volume:** Data warehouses can contain vast amounts of data, which can make it challenging to manage and process. Managing the volume of data requires careful planning, design, and optimization to ensure that the system can handle the required workload.
- **Performance:** Data warehouses must provide fast query response times to support business intelligence and analytics. Achieving high performance can be challenging, as data warehouses require complex data models, indexing strategies, and query optimization techniques.
- **Security:** Data warehouses contain sensitive data, and ensuring data security is crucial. Implementing robust security measures, such as access control, data encryption, and data masking, can be challenging, especially when dealing with large volumes of data.
- **Business Requirements:** Designing and implementing a data warehouse that meets business requirements can be challenging. Business requirements can be complex, and may require specialized data models, analytics, or reporting capabilities. Meeting these requirements requires careful planning, communication, and collaboration between business stakeholders and IT teams.
- implementing a data warehouse can be a complex and challenging process that requires careful planning, design, and execution. Overcoming these difficulties requires a combination of technical expertise, project management skills, and effective communication and collaboration between IT teams and business stakeholders

- **Cost:** Implementing a data warehouse can be expensive, with significant costs associated with hardware, software, and ongoing maintenance. The cost of data warehouse implementation needs to be weighed against the expected benefits of improved decision-making and increased efficiency.
- **Change management:** Data warehouses are designed to support business decision-making, and therefore, changes in business processes and requirements must be reflected in the data warehouse. Effective change management processes must be in place to ensure that the data warehouse can adapt to evolving business needs.

  Advantages:
  - Improved Data Quality
  - Better Decision-Making
  - Historical Analysis
  - Scalability
  - Supports Self-Service Analytics

  Disadvantages:
  - Time and Cost
  - Complexity
  - Data Silos
  - Performance
  - Data Governance

12. Compare and contrast the advantages and disadvantages of vertical partitioning and horizontal partitioning in a data warehousing context.

[Nov 2024]

- Vertical partitioning and horizontal partitioning are two techniques used in database design to optimize the organization and management of large tables. They involve dividing a table into smaller, more manageable pieces to improve query performance and maintenance.

Vertical Partitioning

- Involves dividing a table based on columns. It is useful when you have tables with many columns, and not all columns are frequently accessed together. This

can improve query performance by reducing I/O and allowing for more efficient indexing of relevant columns.

Horizontal Partitioning

- Involves dividing a table based on rows, often using a range or a condition. It is useful when dealing with tables containing a large number of rows, and data can be logically grouped based on certain criteria. This can improve query performance by minimizing the amount of data that needs to be scanned for specific queries and allows for easier data management.

Difference between Vertical and Horizontal Partitioning.

Feature	Vertical Partitioning	Horizontal Partitioning
Definition	Dividing a table into smaller tables based on columns.	Dividing a table into smaller tables based on rows (usually ranges of rows).
Purpose	Reduce the number of columns in a table to improve query performance and reduce I/O.	Divide a table into smaller tables to manage large volumes of data efficiently.
Data distribution	Columns with related data are placed together in the same table.	Rows with related data (typically based on a range or a condition) are placed together in the same table.
Query performance	Improves query performance when queries only involve specific columns that are part of a partition.	Improves query performance when queries primarily access a subset of rows in a large table.
Maintenance and indexing	Easier to manage and index specific columns based on their characteristics and access patterns.	Each partition can be indexed independently, making indexing more efficient.
Joins	May require joins to combine data from multiple partitions when querying.	Joins between partitions are typically not needed, as they contain disjoint sets of data.

Feature	Vertical Partitioning	Horizontal Partitioning
Data integrity	Ensuring data consistency across partitions can be more challenging.	Easier to maintain data integrity, as each partition contains a self-contained subset of data.
Use cases	Commonly used for tables with a wide range of columns, where not all columns are frequently accessed together.	Commonly used for tables with a large number of rows, where data can be grouped based on some criteria (e.g., date ranges).
Examples	Splitting a customer table into one table for personal details and another for transaction history.	Partitioning a large sales order table by date, with each partition containing orders from a specific month or year.

UNIT IV DIMENSIONAL MODELING AND SCHEMA 6

Dimensional Modeling- Multi-Dimensional Data Modeling – Data Cube- Star Schema- Snowflake schema- Star Vs Snowflake schema- Fact constellation Schema- Schema Definition – Process Architecture- Types of Data Base Parallelism – Data warehouse Tools

PART A

1. **Name the types of data warehouse schema. [Nov 2024]**

   The three main types of data warehouse schemas are:

   - Star schema: A basic schema that organizes data into a central fact table and dimension tables
   - Snowflake schema: A more complex schema that builds on the star schema by adding sub-dimension tables
   - Fact Constellation Schema or Galaxy schema: A schema that uses multiple fact tables that share dimension tables

2. **What is the significance of a fact constellation schema in dimensional modelling?**

   - A fact constellation schema, also known as a galaxy schema, is a dimensional modeling design that allows for multiple fact tables to exist within a data warehouse, each representing a different business process, while sharing common dimension tables, providing flexibility to analyze diverse aspects of a business by utilizing shared context across different data sets;
   - It is considered more complex than a standard star schema but offers greater adaptability for intricate business scenarios.

3. **What is a schema dimensional modelling?**

   - Dimensional Data Modelling in a Data Warehouse creates a Schema which is optimized for high performance.
   - It means fewer joins between tables and it also helps with minimized data redundancy. The Dimensional Data Model also helps to boost query performance.

4. **What is dimensional modeling with example?**

   - Dimensional Data Modeling is used for calculating summarized data.

- For example, sales data could be collected on a daily basis and then be aggregated to the week level, the week data could be aggregated to the month level, and so on. The data can then be referred to as aggregate data.

5. What is meant Data Cube?

- A data cube is a data structure that, contrary to tables and spreadsheets, can store data in more than 2 dimensions.
- They are mainly used for fast retrieval of aggregated data. The key elements of a data cube are dimensions, attributes, facts and measures.

6. What is difference between star and snowflake schema?

- In a star schema, relationships between tables are represented by a single join, resulting in a simple data structure for fast query performance and easy data analysis.
- The snowflake schema has a complex data structure with multiple levels of relationships between tables, represented by multiple joins.

7. What is Star Schema? [NOV/DEC 2023]

- A star schema is a multi-dimensional data model used to organize data in a database so that it is easy to understand and analyze.
- Star schemas can be applied to data warehouses, databases, data marts, and other tools. The star schema design is optimized for querying large data sets.

8. What is meant Snowflake Schema? [NOV/DEC 2023]

- A snowflake schema is a multi-dimensional data model that is an extension of a star schema, where dimension tables are broken down into sub dimensions.
- Snowflake schemas are commonly used for business intelligence and reporting in OLAP data warehouses, data marts, and relational databases.

9. What is the fact constellation schema?

- Fact Constellation is a schema for representing multidimensional model.
- It is a collection of multiple fact tables having some common dimension tables.
- It can be viewed as a collection of several star schemas and hence, also known as Galaxy schema.

10. What is the difference between snowflake schema and fact constellation schema?

- Snowflake schema is a normalized form of star schema. While fact constellation schema is a normalized form of snowflake schema and star schema.

- Snowflake schema is easy to operate as compared to fact constellation schema as it has less number of joins between the tables.

11. How is fact constellation different from a star schema?

- Unlike the Star schema, fact constellation schema uses heavily complex queries to access data from the database.
- A star schema depicts each dimension with only one-dimension table. While in this, dimension tables are shared by many fact tables.

12. Why is it called a snowflake schema?

- It's called a snowflake schema because its entity-relationship diagram (ERD) looks like a snowflake

13. What is the difference between schema and dimension?

- A dimensional schema physically separates the measures that quantify the business from the descriptive elements (also called dimensions) that describe and categorize the business.

14. Is star schema normalized or denormalized?

- Star schemas denormalize the data, which means adding redundant columns to some dimension tables to make querying and working with the data faster and easier.

15. What is schema model?

- A schema is a collection of database objects, including tables, views, indexes, and synonyms.
- There is a variety of ways of arranging schema objects in the schema models designed for data warehousing. One data warehouse schema model is a star schema.

16. Why is dimensional Modelling important?

- Dimensional data modeling is a helpful activity because it helps organizations think in modular terms about how they operate their business, what are the connective dimensions between business processes, and understand how things relate to one another.

17. Why is it called schema?

- The word schema comes from the Greek word “σχήμα” (skhēma), which means shape, or more generally, plan. The plural is “σχήματα” (skhēmata).

18. What is the difference between model and schema?

- In the context of databases, a model is a high-level representation of the structure of a database, while a schema is a low-level representation of the same structure.
- A database model provides an abstract view of the entities, attributes, and relationships between entities in a database.

19. What is Multi-Dimensional Data Model in Data Warehouse?

- Multidimensional data model in data warehouse is a model which represents data in the form of data cubes.
- It allows to model and view the data in multiple dimensions and it is defined by dimensions and facts. Multidimensional data model is generally categorized around a central theme and represented by a fact table.

20. What are the types of parallel database?

- Shared Memory Architecture.
- Shared Disk Architecture.
- Shared Nothing Architecture.

21. What are the different types of query parallelism?

- There are two types of query parallelism: interquery parallelism and intraquery parallelism.
- Interquery parallelism refers to the ability of the database to accept queries from multiple applications at the same time.
- Intra-query parallelism is the processing of several parts of a single query simultaneously using either intra-partition parallelism or inter-partition parallelism. Not all queries are suitable for parallel processing.
- Fast-running queries do not need parallel processing, for example.

22. What are the various Data warehouse tools?

- Amazon Redshift
- Snowflake
- Big Query
- Microsoft Azure
- Teradata
- PostgreSQL
- Apache Hive
- Oracle Database
- IBM Db2

PART B

1. Explain in detail about Dimensional Modeling, Objectives and its pros and cons.

Dimensional Modeling

- Dimensional modeling represents data with a cube operation, making more suitable logical data representation with OLAP data management. The perception of Dimensional Modeling was developed by Ralph Kimball and is consist of "fact" and "dimension" tables.
- In dimensional modeling, the transaction record is divided into either "facts," which are frequently numerical transaction data, or "dimensions," which are the reference information that gives context to the facts.
- For example, a sale transaction can be damage into facts such as the number of products ordered and the price paid for the products, and into dimensions such as order date, user name, product number, order ship-to, and bill-to locations, and salesman responsible for receiving the order.

Objectives of Dimensional Modeling

The purposes of dimensional modeling are:

- To produce database architecture that is easy for end-clients to understand and write queries.
- To maximize the efficiency of queries. It achieves these goals by minimizing the number of tables and relationships between them.

Advantages of Dimensional Modeling

Following are the benefits of dimensional modeling are:

- Dimensional modeling is simple: Dimensional modeling methods make it possible for warehouse designers to create database schemas that business customers can easily hold and comprehend.
- There is no need for vast training on how to read diagrams, and there is no complicated relationship between different data elements.

Dimensional modeling promotes data quality:

- The star schema enable warehouse administrators to enforce referential integrity checks on the data warehouse.

- Since the fact information key is a concatenation of the essentials of its associated dimensions, a factual record is actively loaded if the corresponding dimensions records are duly described and also exist in the database.
- By enforcing foreign key constraints as a form of referential integrity check, data warehouse DBAs add a line of defense against corrupted warehouses data.
- Performance optimization is possible through aggregates: As the size of the data warehouse increases, performance optimization develops into a pressing concern.
- Customers who have to wait for hours to get a response to a query will quickly become discouraged with the warehouses. Aggregates are one of the easiest methods by which query performance can be optimized.

Disadvantages of Dimensional Modeling

- To maintain the integrity of fact and dimensions, loading the data warehouses with a record from various operational systems is complicated.
- It is severe to modify the data warehouse operation if the organization adopting the dimensional technique changes the method in which it does business.

Elements of Dimensional Modeling

Fact

- It is a collection of associated data items, consisting of measures and context data. It typically represents business items or business transactions.

Dimensions

- It is a collection of data which describe one business dimension. Dimensions decide the contextual background for the facts, and they are the framework over which OLAP is performed.

Measure

- It is a numeric attribute of a fact, representing the performance or behavior of the business relative to the dimensions.
- Considering the relational context, there are two basic models which are used in dimensional modeling:
    - Star Model
    - Snowflake Model

- The star model is the underlying structure for a dimensional model. It has one broad central table (fact table) and a set of smaller tables (dimensions) arranged in a radial design around the primary table.
- The snowflake model is the conclusion of decomposing one or more of the dimensions.

Fact Table

- Fact tables are used to data facts or measures in the business. Facts are the numeric data elements that are of interest to the company.

Characteristics of the Fact table

- The fact table includes numerical values of what we measure. For example, a fact value of 20 might means that 20 widgets have been sold.
- Each fact table includes the keys to associated dimension tables. These are known as foreign keys in the fact table.
- Fact tables typically include a small number of columns.
- When it is compared to dimension tables, fact tables have a large number of rows.

Example:

- A city and state can view a store summary in a fact table in figure 4.1. Item summary can be viewed by brand, color, etc. Customer information can be viewed by name and address.

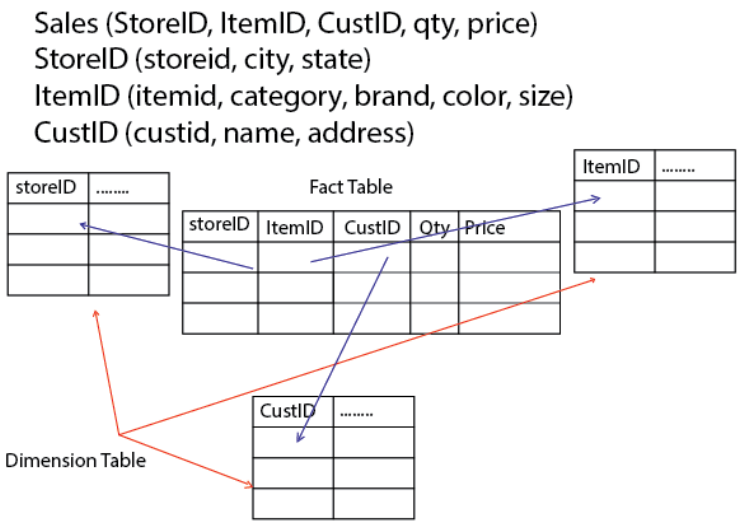


Fig 4.1 Fact Table

Time ID	Product ID	Customer ID	Unit Sold
4	17	2	1
8	21	3	2
8	4	1	1

- In this example, Customer ID column in the facts table is the foreign keys that join with the dimension table. By following the links, we can see that row 2 of the fact table records the fact that customer 3, Gaurav, bought two items on day 8.

Dimension Tables

Customer ID	Name	Gender	Income	Education	Region
1	Rohan	Male	2	3	4
2	Sandeep	Male	3	5	1
3	Gaurav	Male	1	7	3

Hierarchy

- A hierarchy is a directed tree whose nodes are dimensional attributes and whose arcs model many to one association between dimensional attributes team.
- It contains a dimension, positioned at the tree's root, and all of the dimensional attributes that define it.

2. Explain in detail about Multidimensional Data Modeling with neat diagram explanation.

Multi-Dimensional Data Modeling

- A multidimensional model views data in the form of a data-cube. A data cube enables data to be modeled and viewed in multiple dimensions. It is defined by dimensions and facts.
- The dimensions are the perspectives or entities concerning which an organization keeps records.
- For example, a shop may create a sales data warehouse to keep records of the store's sales for the dimension time, item, and location as in figure 4.2.
- These dimensions allow the save to keep track of things, for example, monthly sales of items and the locations at which the items were sold.
- Each dimension has a table related to it, called a dimensional table, which describes the dimension further. For example, a dimensional table for an item may contain the attributes item_name, brand, and type.
- A multidimensional data model is organized around a central theme, for example, sales. This theme is represented by a fact table.
- Facts are numerical measures. The fact table contains the names of the facts or measures of the related dimensional tables.

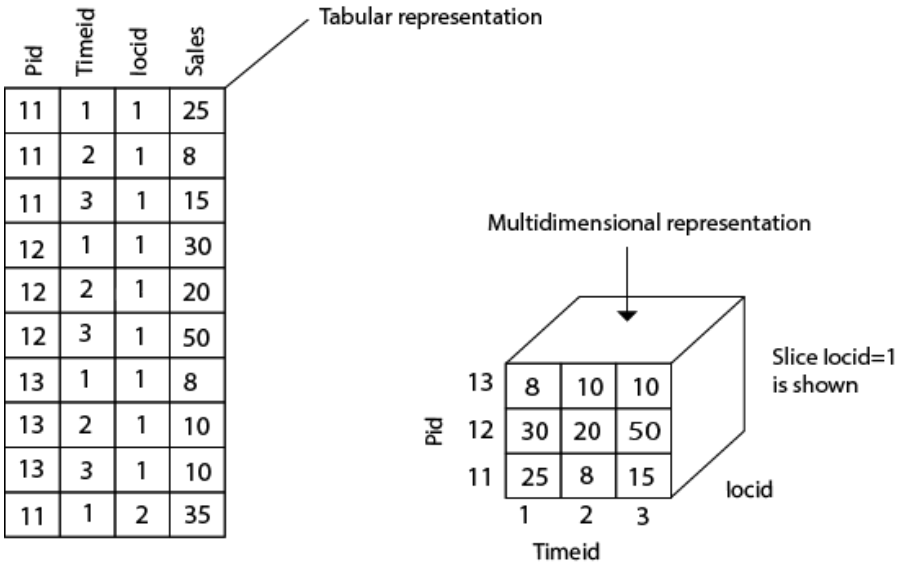


Fig.4.2 Sales Database - Relational Table to Multidimensional Cube

- Consider the data of a shop for items sold per quarter in the city of Delhi. The data is shown in the table.
- In this 2D representation, the sales for Delhi are shown for the time dimension (organized in quarters) and the item dimension (classified according to the types of an item sold).
- The fact or measure displayed in rupee_sold (in thousands).

Location="Delhi"				
Time (quarter)	item (type)			
	Egg	Milk	Bread	Biscuit
Q1	260	508	15	60
Q2	390	256	20	90
Q3	436	396	50	40
Q4	528	483	35	50

Table. 4.1 – 3D data

- Now, if we want to view the sales data with a third dimension,
- For example, suppose the data according to time and item, as well as the location is considered for the cities Chennai, Kolkata, Mumbai, and Delhi.
- These 3D data are shown in the table.
- The 3D data of the table are represented as a series of 2D tables.

	Location="Chennai"				Location="Kolkata"				Location="Mumbai"				Location="Delhi"			
	item				item				item				item			
Time	Egg	Milk	Bread	Biscuit	Egg	Milk	Bread	Biscuit	Egg	Milk	Bread	Biscuit	Egg	Milk	Bread	Biscuit
Q1	340	360	20	10	435	460	20	15	390	385	20	39	260	508	15	60
Q2	490	490	16	50	389	385	45	35	463	366	25	48	390	256	20	90
Q3	680	583	46	43	684	490	39	48	568	594	36	39	436	396	50	40
Q4	535	694	39	38	335	365	83	35	338	484	48	80	528	483	35	50

Table. 4.2 - 3-D data to 2-D data

- Conceptually, it may also be represented by the same data in the form of a 3D data cube, as shown in fig 4.3:

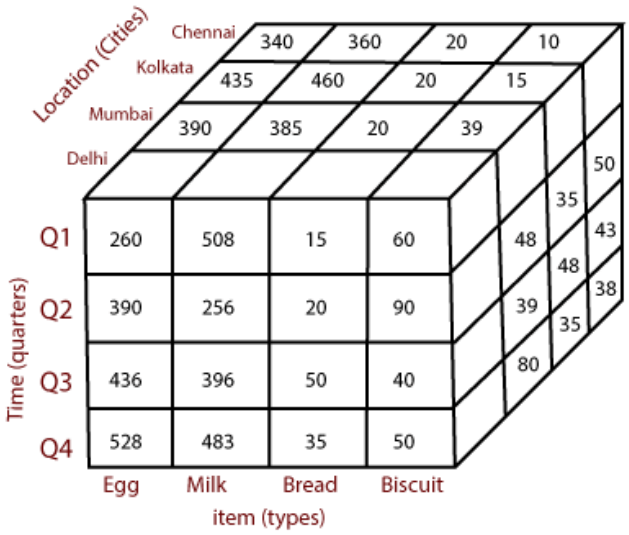


Fig.4.3 Table 4.2 in 3D Data Cube

3. Explain in detail about Data Cube with an example.

Data Cube

- When data is grouped or combined in multidimensional matrices called Data Cubes.
- The data cube method has a few alternative names or a few variants, such as "Multidimensional databases," "materialized views," and "OLAP (On-Line Analytical Processing)."
- The general idea of this approach is to materialize certain expensive computations that are frequently inquired.
- **For example,** a relation with the schema sales (part, supplier, customer, and sale-price) can be materialized into a set of eight views as shown in fig 4.4, where psc indicates a view consisting of aggregate function value (such as total-sales) computed by grouping three attributes part, supplier, and customer, p indicates a view composed of the corresponding aggregate function values calculated by grouping part alone, etc.

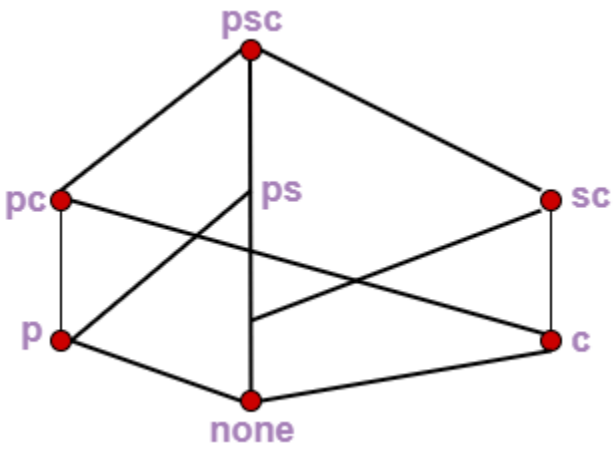


Fig 4.4 Eight Views of Data Cubes for Sales Information

- A data cube is created from a subset of attributes in the database. Specific attributes are chosen to be measure attributes, i.e., the attributes whose values are of interest.
- Another attributes are selected as dimensions or functional attributes. The measure attributes are aggregated according to the dimensions.

- For example, XYZ may create a sales data warehouse to keep records of the store's sales for the dimensions time, item, branch, and location.
- These dimensions enable the store to keep track of things like monthly sales of items, and the branches and locations at which the items were sold.
- Each dimension may have a table identify with it, known as a dimensional table, which describes the dimensions.
- For example, a dimension table for items may contain the attributes item_name, brand, and type.
- Data cube method is an interesting technique with many applications. Data cubes could be sparse in many cases because not every cell in each dimension may have corresponding data in the database.
- Techniques should be developed to handle sparse cubes efficiently.
- If a query contains constants at even lower levels than those provided in a data cube, it is not clear how to make the best use of the precomputed results stored in the data cube.
- The model view data in the form of a data cube. OLAP tools are based on the multidimensional data model. Data cubes usually model n-dimensional data.
- A data cube enables data to be modeled and viewed in multiple dimensions. A multidimensional data model is organized around a central theme, like sales and transactions. A fact table represents this theme. Facts are numerical measures. Thus, the fact table contains measure (such as Rs_sold) and keys to each of the related dimensional tables.
- Dimensions are a fact that defines a data cube. Facts are generally quantities, which are used for analyzing the relationship between dimensions as in figure 4.5 .

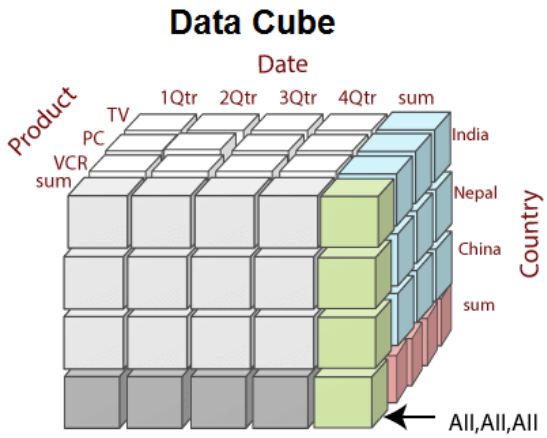


Fig.4.5 Electronics items in Data Cube

Example:

- In the 2-D representation, the All Electronics sales data for items sold per quarter in the city of Vancouver. The measured display in dollars sold (in thousands).

2-D view of Sales Data

location ="Vancouver"				
	item (type)			
time (quarter)	home entertainment	computer	phone	security
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q3	927	1038	38	580

Table 4.3 – 2D Views of Sales Data

3-Dimensional Cuboids

- Let suppose we would like to view the sales data with a third dimension.
- For example, suppose we would like to view the data according to time, item as well as the location for the cities Chicago, New York, Toronto, and Vancouver. The measured display in dollars sold (in thousands).

- These 3-D data are shown in the table. The 3-D data of the table are represented as a series of 2-D tables.

3-D view of Sales Data

location ="Chicago"					location ="New York"				location ="Toronto"			
item					item				item			
time	home ent.	comp.	phone	sec.	time	comp.	phone	sec.	home ent.	comp.	phone	sec.
Q1	854	882	89	623	1087	968	38	872	818	746	43	591
Q2	943	890	64	698	1130	1024	41	925	894	769	52	682
Q3	1032	924	59	789	1034	1048	45	1002	940	795	58	728
Q4	1129	992	63	870	1142	1091	54	984	978	864	59	784

Table 4.4 3-D View of Sales Data in Relational Table

- Conceptually, we may represent the same data in the form of 3-D data cubes, as shown in fig 4.6:

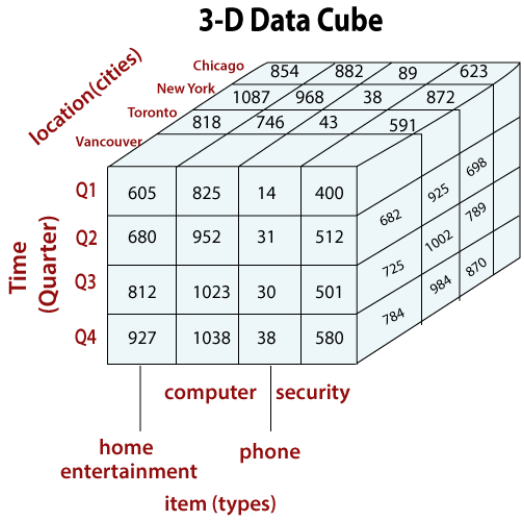


Fig. 4.6 3-D View of Sales Data in Multidimensional Cube

- Let us suppose that we would like to view our sales data with an additional fourth dimension, such as a supplier.
- In data warehousing, the data cubes are n-dimensional. The cuboid which holds the lowest level of summarization is called a base cuboid.

- For example, the 4-D cuboid in the figure 4.7 is the base cuboid for the given time, item, location, and supplier dimensions.

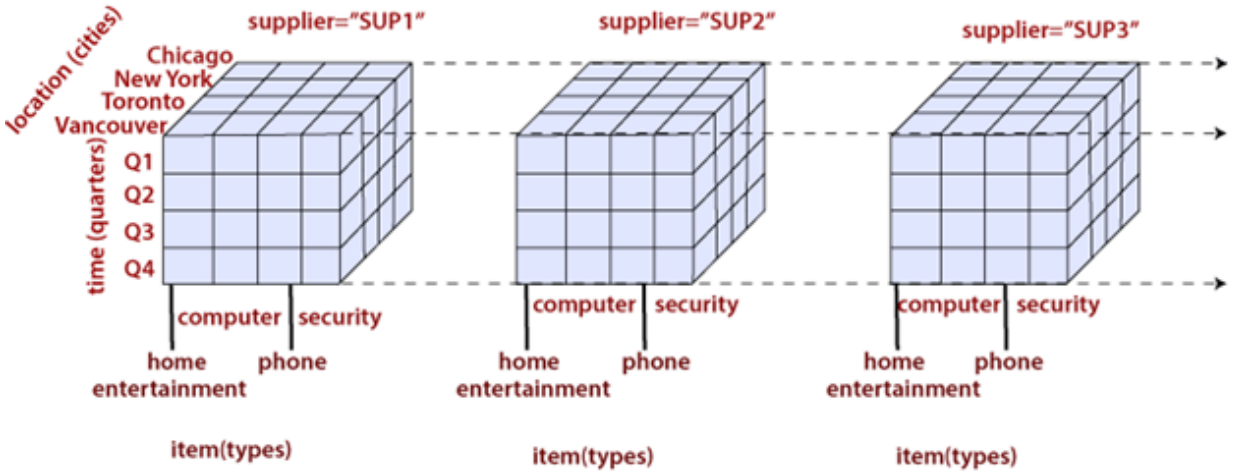


Fig. 4.7 4-D Cuboid

- Figure 4.7 is shown a 4-D data cube representation of sales data, according to the dimensions time, item, location, and supplier. The measure displayed is dollars sold (in thousands).
- The topmost 0-D cuboid, which holds the highest level of summarization, is known as the apex cuboid. In this example, this is the total sales, or dollars sold, summarized over all four dimensions.
- The lattice of cuboid forms a data cube. The figure 4.8 shows the lattice of cuboids creating 4-D data cubes for the dimension time, item, location, and supplier. Each cuboid represents a different degree of summarization.

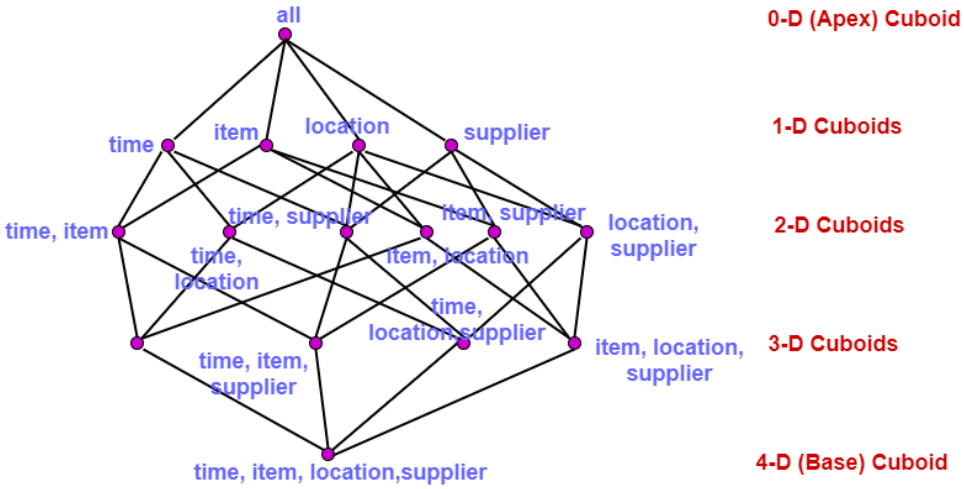


Fig.4.8 Lattice of Cuboid

4. Explain in detail about Star Schema with its advantages & disadvantages.

Describe the process architecture involved in designing and implementing a star schema. [Nov 2024]

Star Schema

- A star schema is the elementary form of a dimensional model, in which data are organized into facts and dimensions.
- A fact is an event that is counted or measured, such as a sale or log in. A dimension includes reference data about the fact, such as date, item, or customer.
- A star schema is a relational schema where a relational schema whose design represents a multidimensional data model.
- The star schema is the explicit data warehouse schema. It is known as star schema because the entity-relationship diagram of this schemas simulates a star, with points, diverge from a central table.
- The center of the schema consists of a large fact table, and the points of the star are the dimension tables as in figure 4.9.

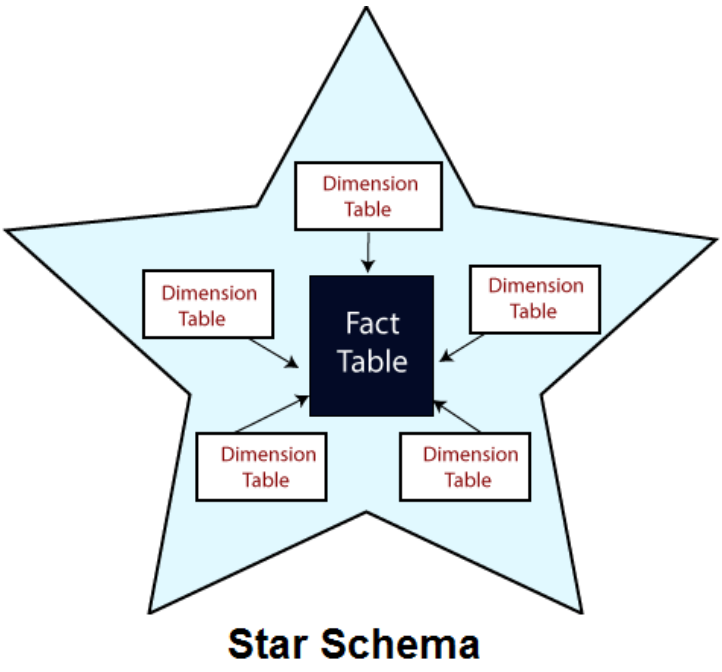


Fig.4.9 Star Schema

Fact Tables

- A table in a star schema which contains facts and connected to dimensions.
- A fact table has two types of columns: those that include fact and those that are foreign keys to the dimension table. The primary key of the fact tables is generally a composite key that is made up of all of its foreign keys.
- A fact table might involve either detail level fact or fact that have been aggregated (fact tables that include aggregated fact are often instead called summary tables). A fact table generally contains facts with the same level of aggregation.

Dimension Tables

- A dimension is an architecture usually composed of one or more hierarchies that categorize data.
- If a dimension has not got hierarchies and levels, it is called a flat dimension or list. The primary keys of each of the dimensions table are part of the composite primary keys of the fact table.
- Dimensional attributes help to define the dimensional value. They are generally descriptive, textual values. Dimensional tables are usually small in size than fact table.

- Fact tables store data about sales while dimension tables data about the geographic region (markets, cities), clients, products, times, channels.

Characteristics of Star Schema

- The star schema is intensely suitable for data warehouse database design because of the following features:
- It creates a DE-normalized database that can quickly provide query responses.
- It provides a flexible design that can be changed easily or added to throughout the development cycle, and as the database grows.
- It provides a parallel in design to how end-users typically think of and use the data.
- It reduces the complexity of metadata for both developers and end-users.

Advantages of Star Schema

- Star Schemas are easy for end-users and application to understand and navigate. With a well-designed schema, the customer can instantly analyze large, multidimensional data sets.

The main advantage of star schemas in a decision-support environment are:

Fig.4.10 Advantages of Star Schema

Query Performance

- A star schema database has a limited number of table and clear join paths, the query run faster than they do against OLTP systems. Small single-table queries, frequently of a dimension table, are almost instantaneous. Large join queries that contain multiple tables takes only seconds or minutes to run.

- In a star schema database design, the dimension is connected only through the central fact table. When the two-dimension table is used in a query, only one join path, intersecting the fact tables, exist between those two tables. This design feature enforces authentic and consistent query results.

Load performance and administration

- Structural simplicity also decreases the time required to load large batches of record into a star schema database. By describing facts and dimensions and separating them into the various table, the impact of a load structure is reduced. Dimension table can be populated once and occasionally refreshed. We can add new facts regularly and selectively by appending records to a fact table.

Built-in referential integrity

- A star schema has referential integrity built-in when information is loaded. Referential integrity is enforced because each data in dimensional tables has a unique primary key, and all keys in the fact table are legitimate foreign keys drawn from the dimension table. A record in the fact table which is not related correctly to a dimension cannot be given the correct key value to be retrieved.

Easily Understood

- A star schema is simple to understand and navigate, with dimensions joined only through the fact table. These joins are more significant to the end-user because they represent the fundamental relationship between parts of the underlying business. Customer can also browse dimension table attributes before constructing a query.

Disadvantage of Star Schema

- There is some condition which cannot be meet by star schemas like the relationship between the user, and bank account cannot describe as star schema as the relationship between them is many to many.

Example: Suppose a star schema is composed of a fact table, SALES, and several dimension tables connected to it for time, branch, item, and geographic locations as in figure 4.11.

- The TIME table has a column for each day, month, quarter, and year.
- The ITEM table has columns for each item_Key, item_name, brand, type, supplier_type. The BRANCH table has columns for each branch_key, branch_name, branch_type.
- The LOCATION table has columns of geographic data, including street, city, state, and country.

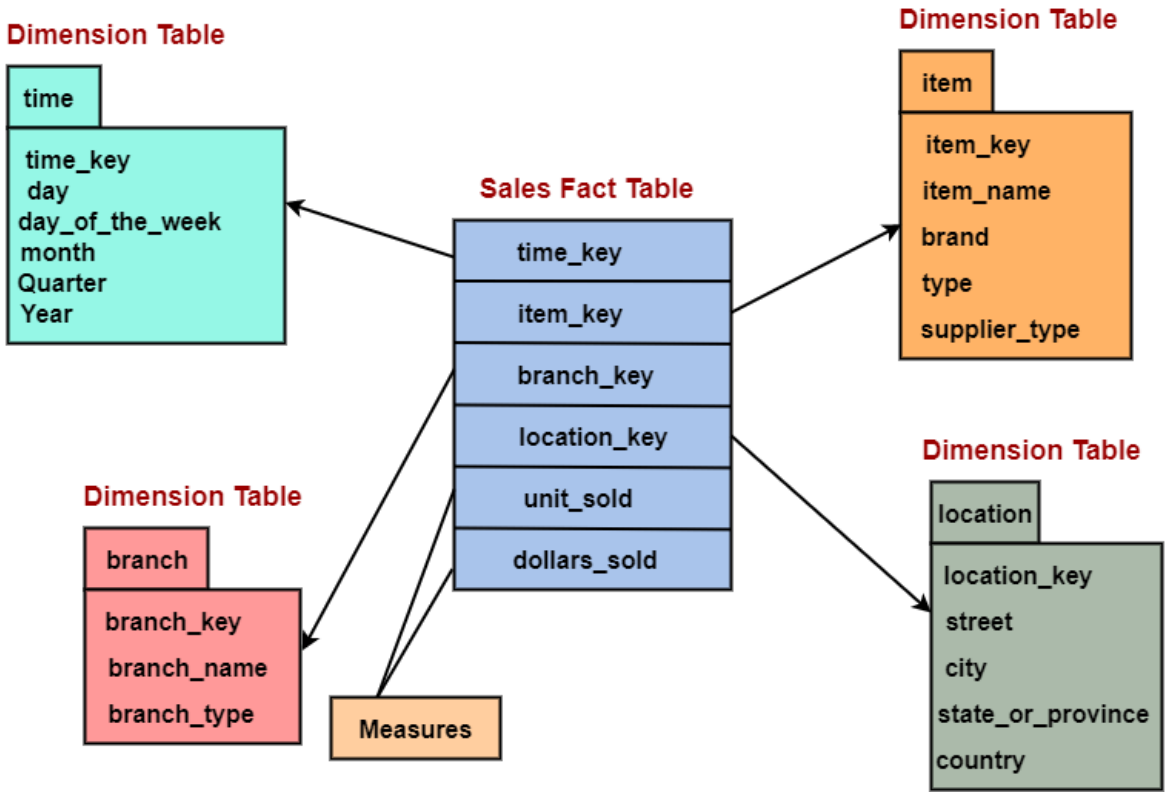


Fig.4.11 Example - Star Schema

- In this scenario, the SALES table contains only four columns with IDs from the dimension tables, TIME, ITEM, BRANCH, and LOCATION, instead of four columns for time data, four columns for ITEM data, three columns for BRANCH data, and four columns for LOCATION data.
- Thus, the size of the fact table is significantly reduced.
- When we need to change an item, we need only make a single change in the dimension table, instead of making many changes in the fact table.
- We can create even more complex star schemas by normalizing a dimension table into several tables. The normalized dimension table is called a Snowflake.

5. Explain in detail about Snowflake Schema with an example.

Snowflake Schema

- "A schema is known as a snowflake if one or more dimension tables do not connect directly to the fact table but must join through other dimension tables."
- The snowflake schema is an expansion of the star schema where each point of the star explodes into more points. It is called snowflake schema because the diagram of snowflake schema resembles a snowflake.
- Snowflaking is a method of normalizing the dimension tables in a STAR schemas. When we normalize all the dimension tables entirely, the resultant structure resembles a snowflake with the fact table in the middle.
- Snowflaking is used to develop the performance of specific queries. The schema is diagramed with each fact surrounded by its associated dimensions, and those dimensions are related to other dimensions, branching out into a snowflake pattern.
- The snowflake schema consists of one fact table which is linked to many dimension tables, which can be linked to other dimension tables through a many-to-one relationship.
- Tables in a snowflake schema are generally normalized to the third normal form. Each dimension table performs exactly one level in a hierarchy.
- The following diagram shows a snowflake schema with two dimensions, each having three levels.
- A snowflake schemas can have any number of dimension, and each dimension can have any number of levels.
- Refer 4.12

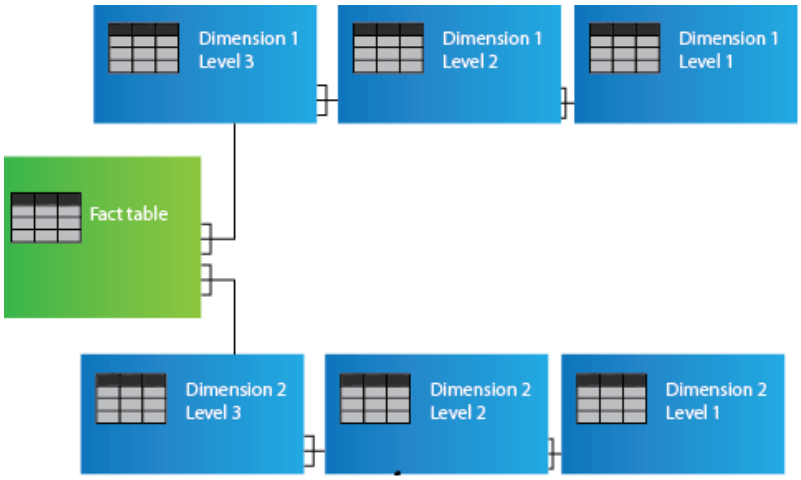


Fig.4.12 Snowflake Schema

- **Example:** Figure shows a snowflake schema with a Sales fact table, with Store, Location, Time, Product, Line, and Family dimension tables.
- The Market dimension has two dimension tables with Store as the primary dimension table, and Location as the outrigger dimension table.
- The product dimension has three dimension tables with Product as the primary dimension table, and the Line and Family table are the outrigger dimension tables.

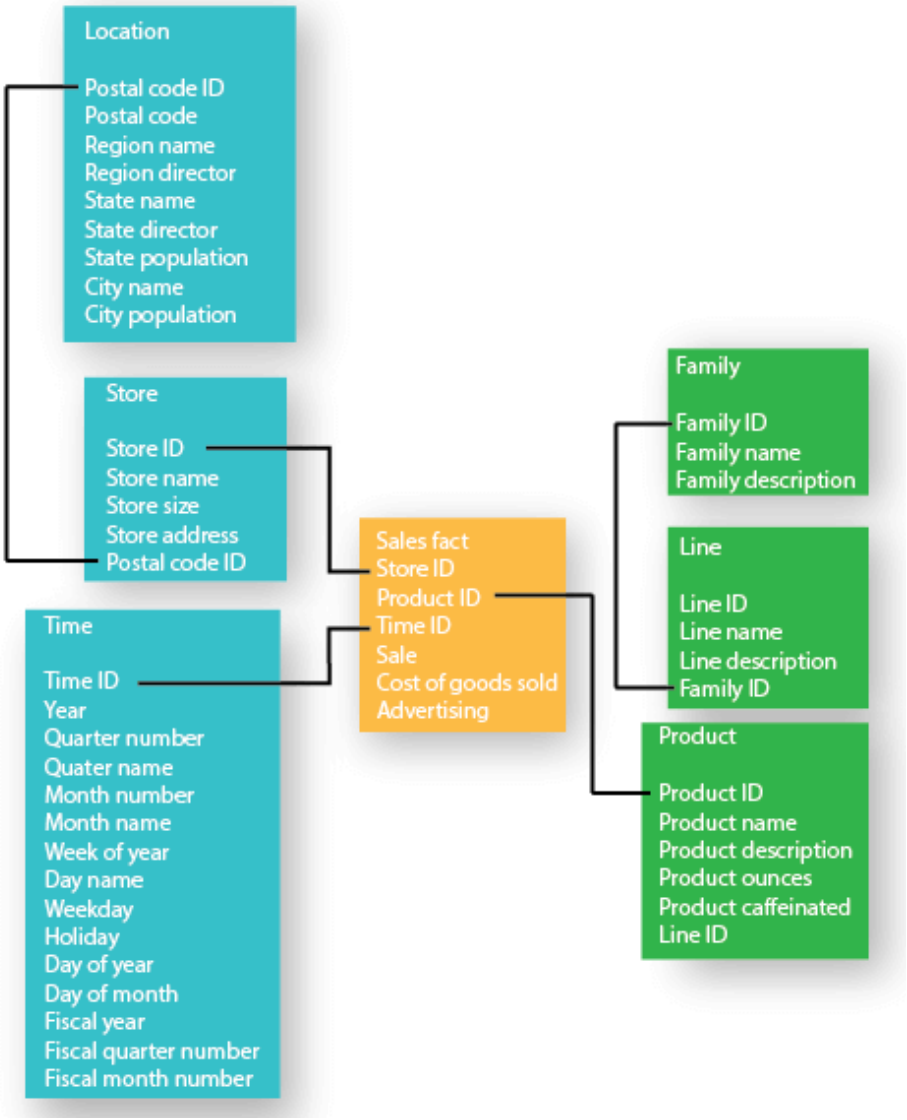


Fig. 4.13 Sales Example – Snowflake schema

- A star schema store all attributes for a dimension into one denormalized table. This needed more disk space than a more normalized snowflake schema.
- Snowflaking normalizes the dimension by moving attributes with low cardinality into separate dimension tables that relate to the core dimension table by using foreign keys.
- Snowflaking for the sole purpose of minimizing disk space is not recommended, because it can adversely impact query performance.

- In snowflake, schema tables are normalized to delete redundancy. In snowflake dimension tables are damaged into multiple dimension tables.
- Figure shows a simple STAR schema for sales in a manufacturing company. The sales fact table include quantity, price, and other relevant metrics. SALESREP, CUSTOMER, PRODUCT, and TIME are the dimension tables.

6. Explain in detail about Fact Constellation Schema with an example.

Fact Constellation Schema

- A Fact constellation means two or more fact tables sharing one or more dimensions. It is also called **Galaxy schema**. Refer figure 4.14
- Fact Constellation Schema describes a logical structure of data warehouse or data mart.
- Fact Constellation Schema can design with a collection of de-normalized FACT, Shared, and Conformed Dimension tables.

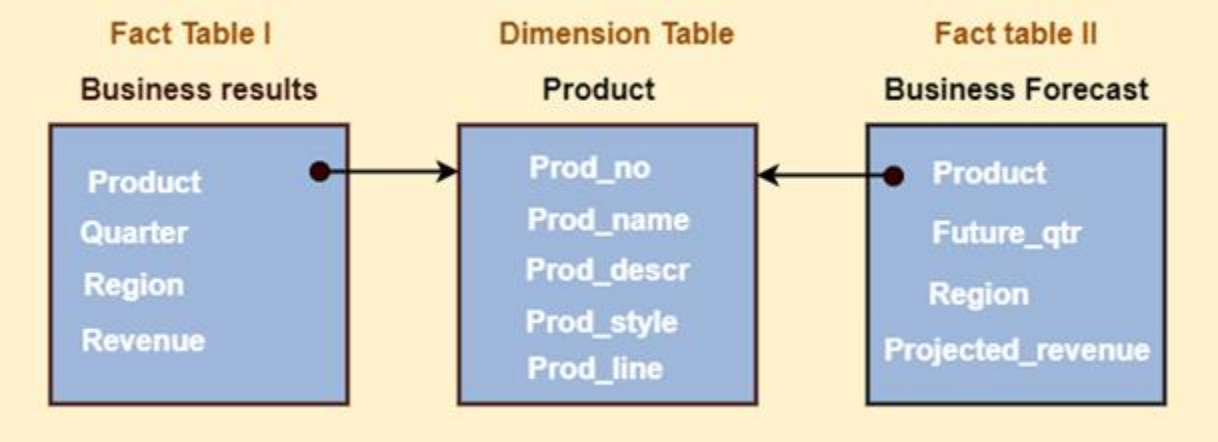


Fig.4.14 Fact Constellation Schema

- Fact Constellation Schema is a sophisticated database design that is difficult to summarize information.
- Fact Constellation Schema can implement between aggregate Fact tables or decompose a complex Fact table into independent simplex Fact tables as in figure 4.15.

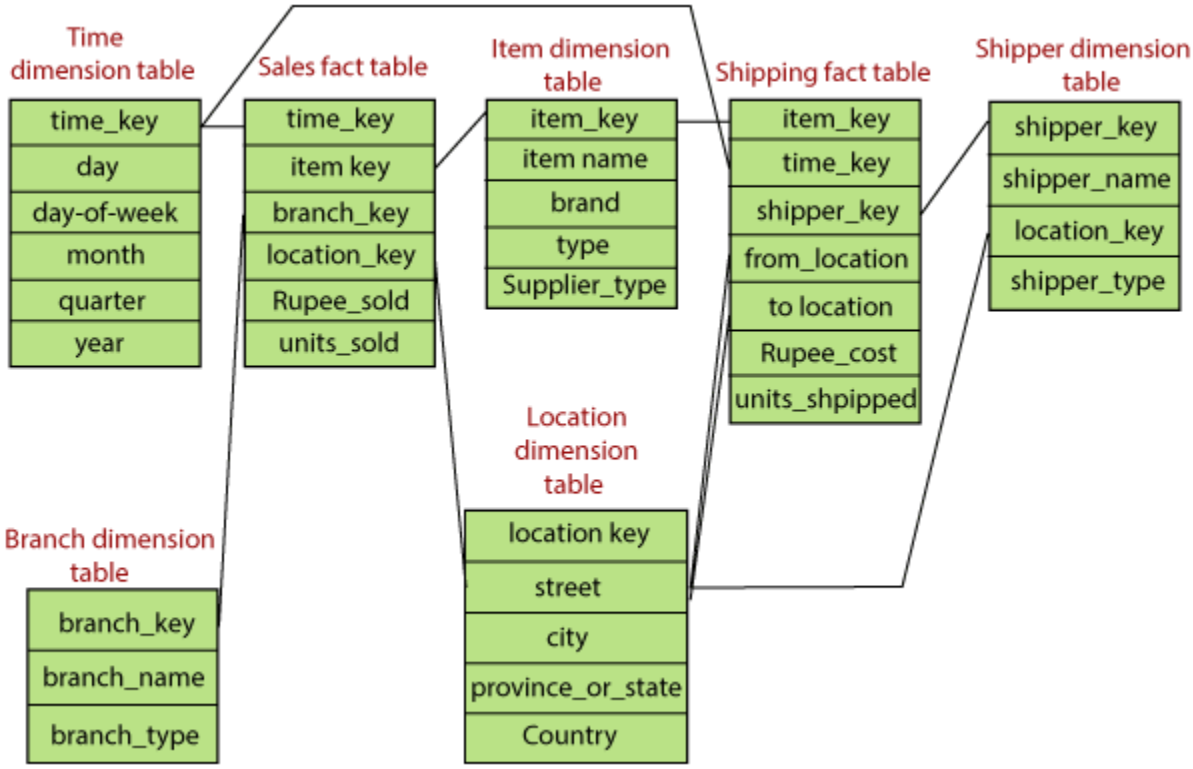


Fig.4.15 Sales Example: A fact constellation schema is shown in the figure.

- This schema defines two fact tables, sales, and shipping. Sales are treated along four dimensions, namely, time, item, branch, and location.
- The schema contains a fact table for sales that includes keys to each of the four dimensions, along with two measures: Rupee_sold and units_sold.
- The shipping table has five dimensions, or keys: item_key, time_key, shipper_key, from_location, and to_location, and two measures: Rupee_cost and units_shipped.
- The primary disadvantage of the fact constellation schema is that it is a more challenging design because many variants for specific kinds of aggregation must be considered and selected.

7. Explain in detail about Data Warehouse Applications. (or) Describe in brief about various schemas in multidimensional data model. [NOV 2023]

Data Warehouse Applications

The application areas of the data warehouse are:

1. **Information Processing**
   - It deals with querying, statistical analysis, and reporting via tables, charts, or graphs. Nowadays, information processing of data warehouse is to construct a low cost, web-based accessing tools typically integrated with web browsers.
2. **Analytical Processing**
   - It supports various online analytical processing such as drill-down, roll-up, and pivoting. The historical data is being processed in both summarized and detailed format.
   - OLAP is implemented on data warehouses or data marts. The primary objective of OLAP is to support ad-hoc querying needed for support DSS.
   - The multidimensional view of data is fundamental to the OLAP application. OLAP is an operational view, not a data structure or schema. The complex nature of OLAP applications requires a multidimensional view of the data.
3. **Data Mining**
   - It helps in the analysis of hidden design and association, constructing scientific models, operating classification and prediction, and performing the mining results using visualization tools.
   - Data mining is the technique of designing essential new correlations, patterns, and trends by changing through high amounts of a record save in repositories, using pattern recognition technologies as well as statistical and mathematical techniques.
   - It is the phase of selection, exploration, and modeling of huge quantities of information to determine regularities or relations that are at first unknown to access precise and useful results for the owner of the database.
   - It is the process of inspection and analysis, by automatic or semi-automatic means, of large quantities of records to discover meaningful patterns and rules.

- Schema is a logical description of the entire database. It includes the name and description of records of all record types including all associated data-items and aggregates. Much like a database, a data warehouse also requires to maintain a schema.
- A database uses relational model, while a data warehouse uses Star, Snowflake, and Fact Constellation schema. In this chapter, we will discuss the schemas used in a data warehouse.

Star Schema

- Each dimension in a star schema is represented with only one-dimension table.
- This dimension table contains the set of attributes.
- The following figure 4.16 shows the sales data of a company with respect to the four dimensions, namely time, item, branch, and location.

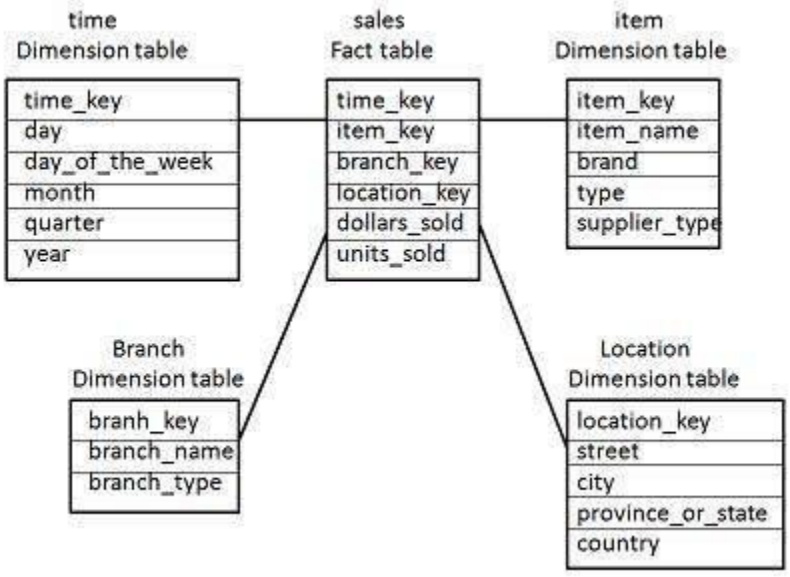


Fig.4.16 Star Schema – Sales Database

- There is a fact table at the center. It contains the keys to each of four dimensions.
- The fact table also contains the attributes, namely dollars sold and units sold.

Note –

- Each dimension has only one dimension table and each table holds a set of attributes. For example, the location dimension table contains the attribute set {location_key, street, city, province_or_state,country}.
- This constraint may cause data redundancy. For example, "Vancouver" and "Victoria" both the cities are in the Canadian province of British Columbia. The entries for such cities may cause data redundancy along the attributes province_or_state and country.

Snowflake Schema

- Some dimension tables in the Snowflake schema are normalized.
- The normalization splits up the data into additional tables.
- Unlike Star schema, the dimensions table in a snowflake schema are normalized. For example the figure 4.17, the item dimension table in star schema is normalized and split into two dimension tables, namely item and supplier table.

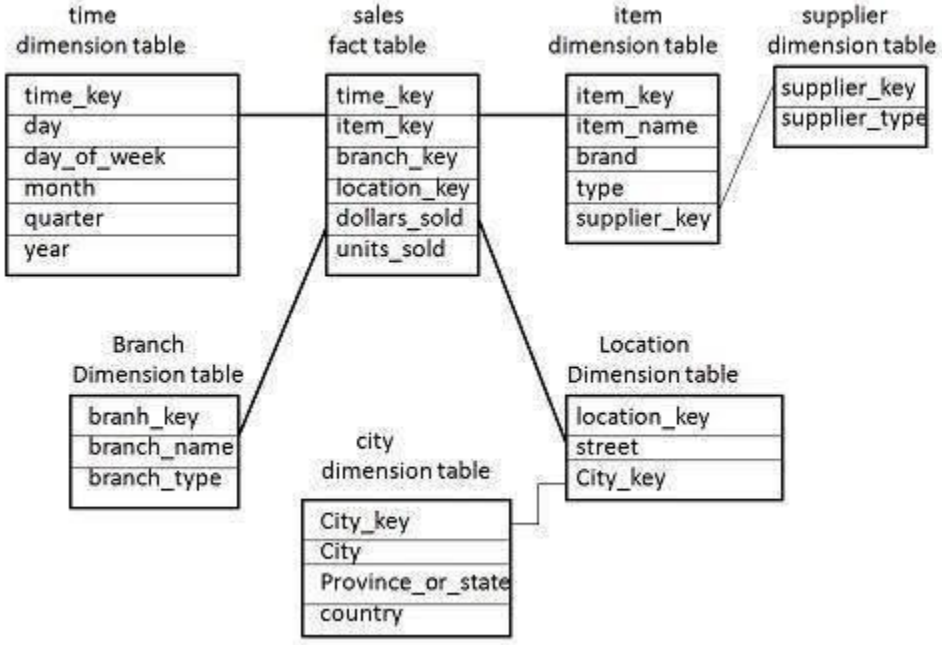


Fig 4.17 Sales Database - Snowflake Schema

- Now the item dimension table contains the attributes item_key, item_name, type, brand, and supplier-key.

- The supplier key is linked to the supplier dimension table. The supplier dimension table contains the attributes supplier_key and supplier_type.

Note – Due to normalization in the Snowflake schema, the redundancy is reduced and therefore, it becomes easy to maintain and the save storage space.

Fact Constellation Schema

- A fact constellation has multiple fact tables. It is also known as galaxy schema.
- The following figure 4.18 shows two fact tables, namely sales and shipping.

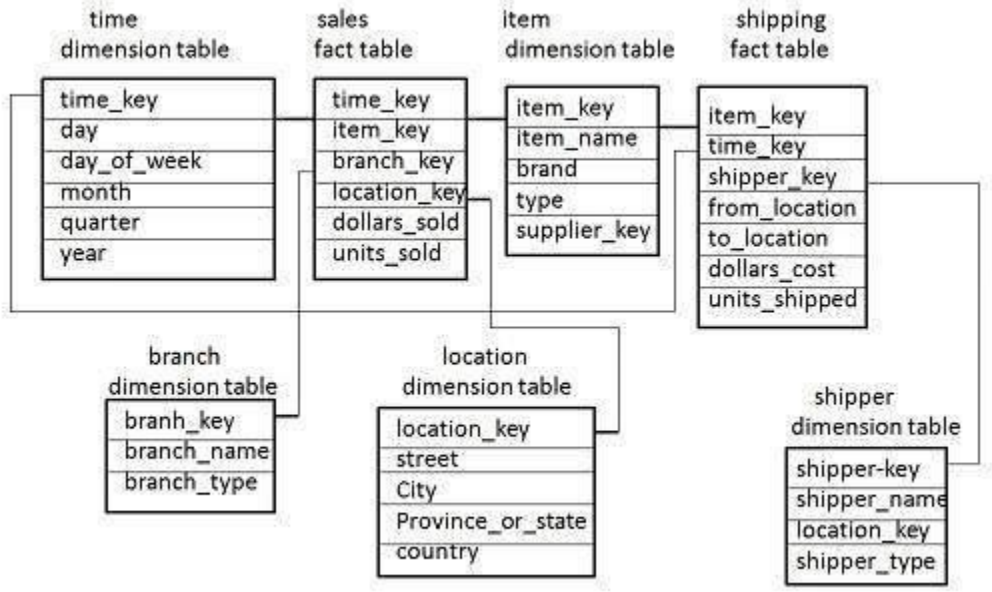


Fig. 4.18 Sales Database - Fact Constellation Schema

- The sales fact table is same as that in the star schema.
- The shipping fact table has the five dimensions, namely item_key, time_key, shipper_key, from_location, to_location.
- The shipping fact table also contains two measures, namely dollars sold and units sold.
- It is also possible to share dimension tables between fact tables. For example, time, item, and location dimension tables are shared between the sales and shipping fact table.

Schema Definition

- Multidimensional schema is defined using Data Mining Query Language (DMQL).
- The two primitives, cube definition and dimension definition, can be used for defining the data warehouses and data marts.

Syntax for Cube Definition

define cube < cube_name > [< dimension-list > }: < measure_list >

Syntax for Dimension Definition

define dimension < dimension_name > as (< attribute_or_dimension_list >)

Star Schema Definition

The star schema that we have discussed can be defined using Data Mining Query Language (DMQL) as follows −

define cube sales star [time, item, branch, location]:

dollars sold = sum(sales in dollars), units sold = count(*)

define dimension time as (time key, day, day of week, month, quarter, year)

define dimension item as (item key, item name, brand, type, supplier type)

define dimension branch as (branch key, branch name, branch type)

define dimension location as (location key, street, city, province or state, country)

Snowflake Schema Definition

Snowflake schema can be defined using DMQL as follows −

define cube sales snowflake [time, item, branch, location]:

dollars sold = sum(sales in dollars), units sold = count(*)

define dimension time as (time key, day, day of week, month, quarter, year)

define dimension item as (item key, item name, brand, type, supplier (supplier key, supplier type))

define dimension branch as (branch key, branch name, branch type)

define dimension location as (location key, street, city (city key, city, province or state, country))

Fact Constellation Schema Definition

Fact constellation schema can be defined using DMQL as follows −

define cube sales [time, item, branch, location]:

dollars sold = sum(sales in dollars), units sold = count(*)

define dimension time as (time key, day, day of week, month, quarter, year)

define dimension item as (item key, item name, brand, type, supplier type)

define dimension branch as (branch key, branch name, branch type)

define dimension location as (location key, street, city, province or state,country)

define cube shipping [time, item, shipper, from location, to location]:

dollars cost = sum(cost in dollars), units shipped = count(*)

define dimension time as time in cube sales

define dimension item as item in cube sales

define dimension shipper as (shipper key, shipper name, location as location in cube sales, shipper type)

define dimension from location as location in cube sales

define dimension to location as location in cube sales

8. Explain in detail about process Architecture or Three Schema Architecture and its objectives.

Process Architecture or three schema architecture

- The three schema architecture is also called ANSI/SPARC architecture or three-level architecture.
- This framework is used to describe the structure of a specific database system.

- The three schema architecture is also used to separate the user applications and physical database.
- The three schema architecture contains three-levels. It breaks the database down into three different categories.

The three-schema architecture is as follows:

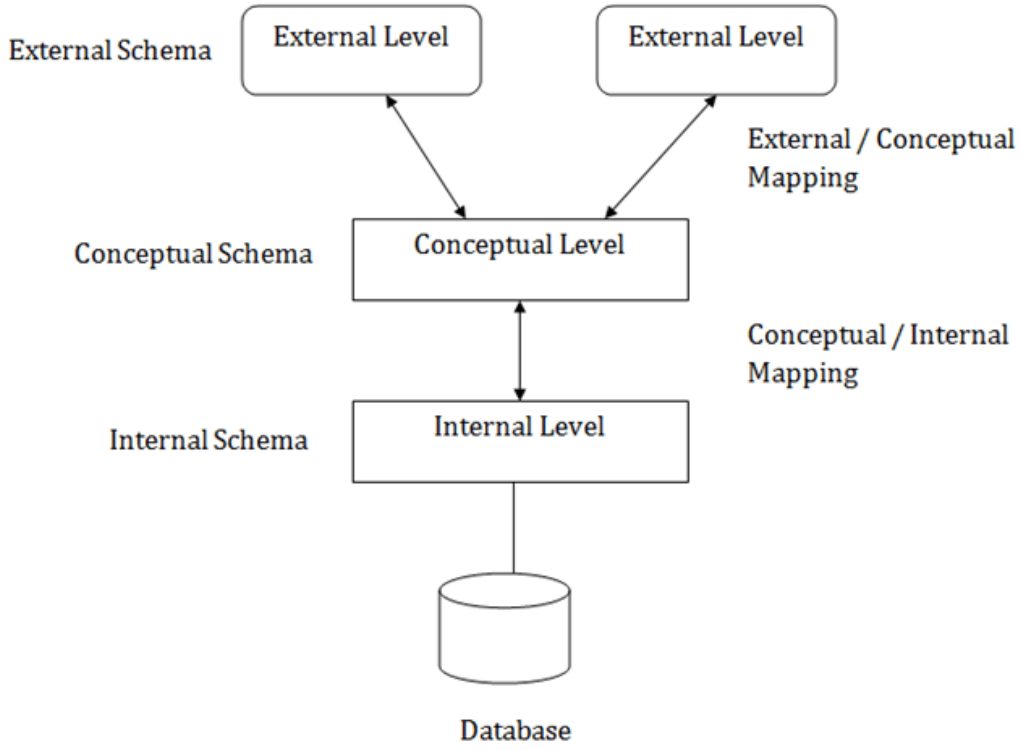


Fig.4.19 Three Schema Architecture

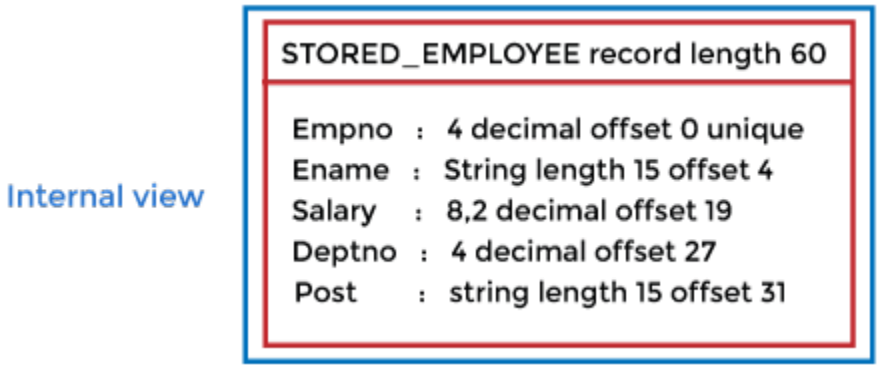
In the above figure 4.19:

- It shows the DBMS architecture.
- Mapping is used to transform the request and response between various database levels of architecture.
- Mapping is not good for small DBMS because it takes more time.
- In External / Conceptual mapping, it is necessary to transform the request from external level to conceptual schema.
- In Conceptual / Internal mapping, DBMS transform the request from the conceptual to internal level.

Objectives of Three schema Architecture

- The main objective of three level architecture is to enable multiple users to access the same data with a personalized view while storing the underlying data only once.
- Thus it separates the user's view from the physical structure of the database. This separation is desirable for the following reasons:
- Different users need different views of the same data.
- The approach in which a particular user needs to see the data may change over time.
- The users of the database should not worry about the physical implementation and internal workings of the database such as data compression and encryption techniques, hashing, optimization of the internal structures etc.
- All users should be able to access the same data according to their requirements.
- DBA should be able to change the conceptual structure of the database without affecting the user's
- Internal structure of the database should be unaffected by changes to physical aspects of the storage.

1. Internal Level



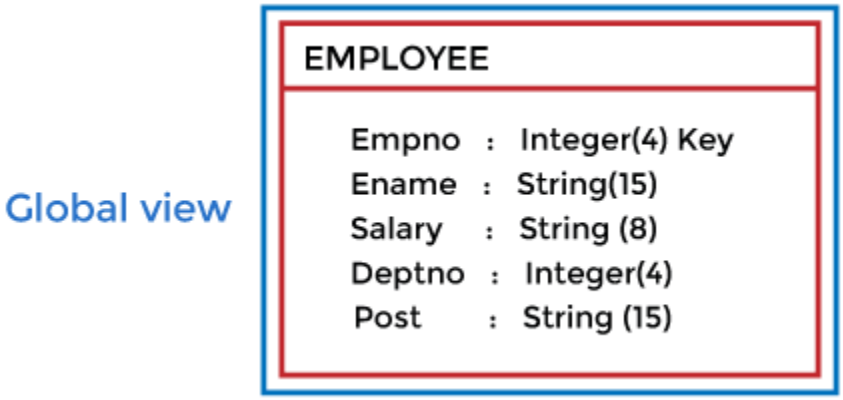
- The internal level has an internal schema which describes the physical storage structure of the database.
- The internal schema is also known as a physical schema.

- It uses the physical data model. It is used to define that how the data will be stored in a block.
- The physical level is used to describe complex low-level data structures in detail.

The internal level is generally is concerned with the following activities:

- Storage space allocations. - **For Example:** B-Trees, Hashing etc.
- Access paths. - **For Example:** Specification of primary and secondary keys, indexes, pointers and sequencing.
- Data compression and encryption techniques.
- Optimization of internal structures.
- Representation of stored fields.

2. Conceptual Level



- The conceptual schema describes the design of a database at the conceptual level. Conceptual level is also known as logical level.
- The conceptual schema describes the structure of the whole database.
- The conceptual level describes what data are to be stored in the database and also describes what relationship exists among those data.
- In the conceptual level, internal details such as an implementation of the data structure are hidden.
- Programmers and database administrators work at this level.

3. External Level

- At the external level, a database contains several schemas that sometimes called as subschema. The subschema is used to describe the different view of the database.
- An external schema is also known as view schema.
- Each view schema describes the database part that a particular user group is interested and hides the remaining database from that user group.
- The view schema describes the end user interaction with database systems.

9. Explain in detail about Mapping between Views.

Mapping between Views

- The three levels of DBMS architecture don't exist independently of each other. There must be correspondence between the three levels i.e. how they actually correspond with each other.
- DBMS is responsible for correspondence between the three types of schema. This correspondence is called Mapping.

There are basically two types of mapping in the database architecture:

- Conceptual/ Internal Mapping
- External / Conceptual Mapping

Conceptual/ Internal Mapping

- The Conceptual/ Internal Mapping lies between the conceptual level and the internal level. Its role is to define the correspondence between the records and fields of the conceptual level and files and data structures of the internal level.

External/ Conceptual Mapping

- The external/Conceptual Mapping lies between the external level and the Conceptual level. Its role is to define the correspondence between a particular external and the conceptual view.

Types of Database parallelism [NOV/DEC 2023]

- Parallelism is used to support speedup, where queries are executed faster because more resources, such as processors and disks, are provided.
- Parallelism is also used to provide scale-up, where increasing workloads are managed without increase response-time, via an increase in the degree of parallelism.
- Different architectures for parallel database systems are shared-memory, shared-disk, shared-nothing, and hierarchical structures.

(a)Horizontal Parallelism:

- It means that the database is partitioned across multiple disks, and parallel processing occurs within a specific task (i.e., table scan) that is performed concurrently on different processors against different sets of data.

(b)Vertical Parallelism:

- It occurs among various tasks. All component query operations (i.e., scan, join, and sort) are executed in parallel in a pipelined fashion. In other words, an output from one function (e.g., join) as soon as records become available.

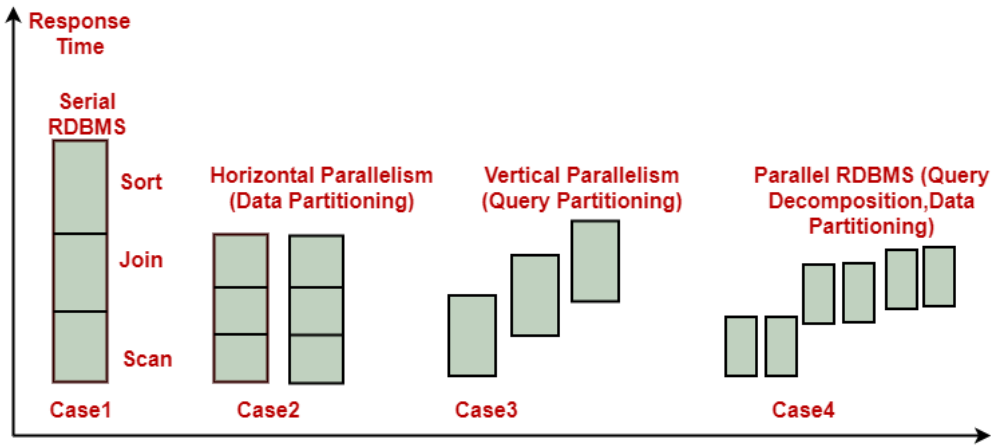


Fig.4.19 Horizontal & Vertical Parallelism

Intraquery Parallelism

- Intraquery parallelism defines the execution of a single query in parallel on multiple processors and disks. Using intraquery parallelism is essential for speeding up long-running queries.
- Interquery parallelism does not help in this function since each query is run sequentially.
- To improve the situation, many DBMS vendors developed versions of their products that utilized intraquery parallelism.
- This application of parallelism decomposes the serial SQL, query into lower-level operations such as scan, join, sort, and aggregation.
- These lower-level operations are executed concurrently, in parallel.

Interquery Parallelism

- In interquery parallelism, different queries or transaction execute in parallel with one another.
- This form of parallelism can increase transactions throughput. The response times of individual transactions are not faster than they would be if the transactions were run in isolation.
- Thus, the primary use of interquery parallelism is to scale up a transaction processing system to support a more significant number of transactions per second.
- Database vendors started to take advantage of parallel hardware architectures by implementing multiserver and multithreaded systems designed to handle a large number of client requests efficiently.
- This approach naturally resulted in interquery parallelism, in which different server threads (or processes) handle multiple requests at the same time.
- Interquery parallelism has been successfully implemented on SMP systems, where it increased the throughput and allowed the support of more concurrent users.

Shared Disk Architecture

- Shared-disk architecture implements a concept of shared ownership of the entire database between RDBMS servers, each of which is running on a node of a distributed memory system. Refer figure 4.20
- Each RDBMS server can read, write, update, and delete information from the same shared database, which would need the system to implement a form of a distributed lock manager (DLM).
- DLM components can be found in hardware, the operating system, and separate software layer, all depending on the system vendor.
- On the positive side, shared-disk architectures can reduce performance bottlenecks resulting from data skew (uneven distribution of data), and can significantly increase system availability.
- The shared-disk distributed memory design eliminates the memory access bottleneck typically of large SMP systems and helps reduce DBMS dependency on data partitioning.

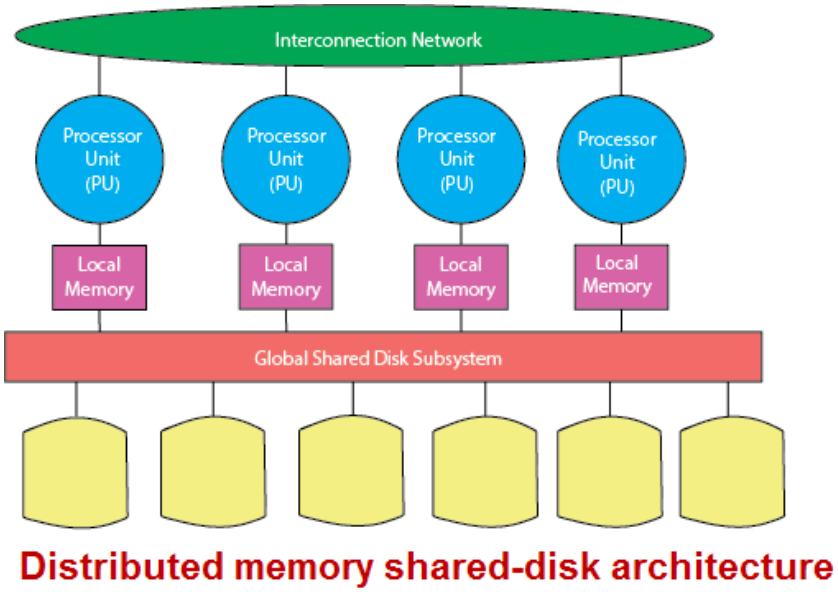


Fig.4.20 Shared Disk Architecture

Shared-Memory Architecture

- Shared-memory or shared-everything style is the traditional approach of implementing an RDBMS on SMP hardware. Refer figure 4.21
- It is relatively simple to implement and has been very successful up to the point where it runs into the scalability limitations of the shared-everything architecture.
- The key point of this technique is that a single RDBMS server can probably apply all processors, access all memory, and access the entire database, thus providing the client with a consistent single system image.

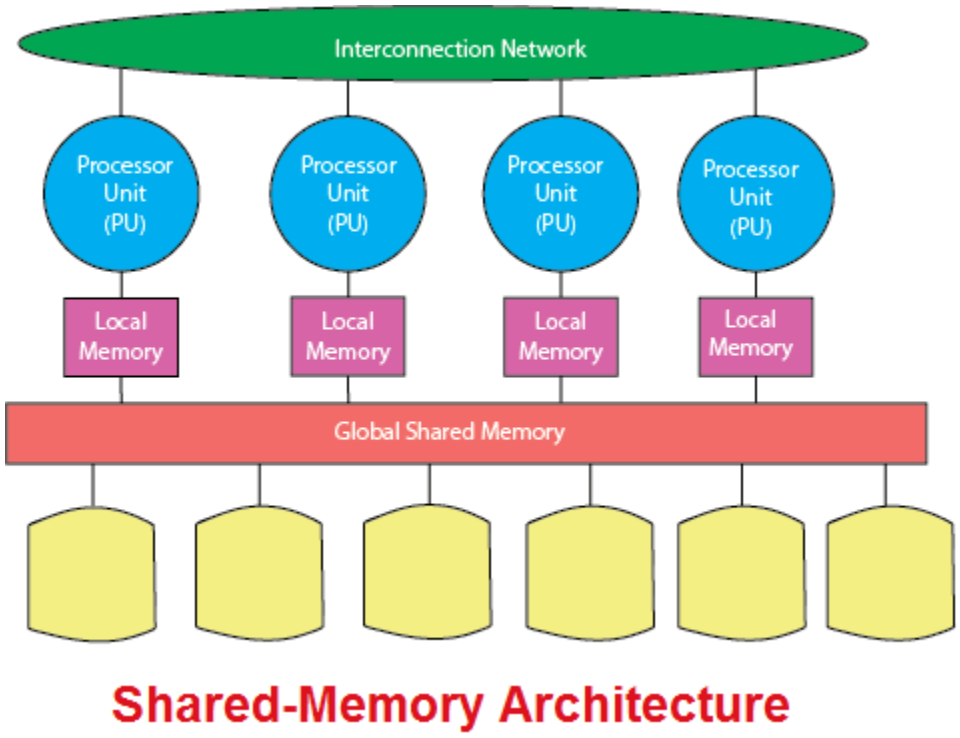


Fig.4.21 Shared Memory Architecture

- In shared-memory SMP systems, the DBMS considers that the multiple database components executing SQL statements communicate with each other by exchanging messages and information via the shared memory.
- All processors have access to all data, which is partitioned across local disks.

Shared-Nothing Architecture

- In a shared-nothing distributed memory environment, the data is partitioned across all disks, and the DBMS is "partitioned" across multiple co-servers, each of which resides on individual nodes of the parallel system and has an ownership of its disk and thus its database partition. Refer figure 4.22
- A shared-nothing RDBMS parallelizes the execution of a SQL query across multiple processing nodes.
- Each processor has its memory and disk and communicates with other processors by exchanging messages and data over the interconnection network.
- This architecture is optimized specifically for the MPP and cluster systems.
- The shared-nothing architectures offer near-linear scalability. The number of processor nodes is limited only by the hardware platform limitations (and budgetary constraints), and each node itself can be a powerful SMP system.

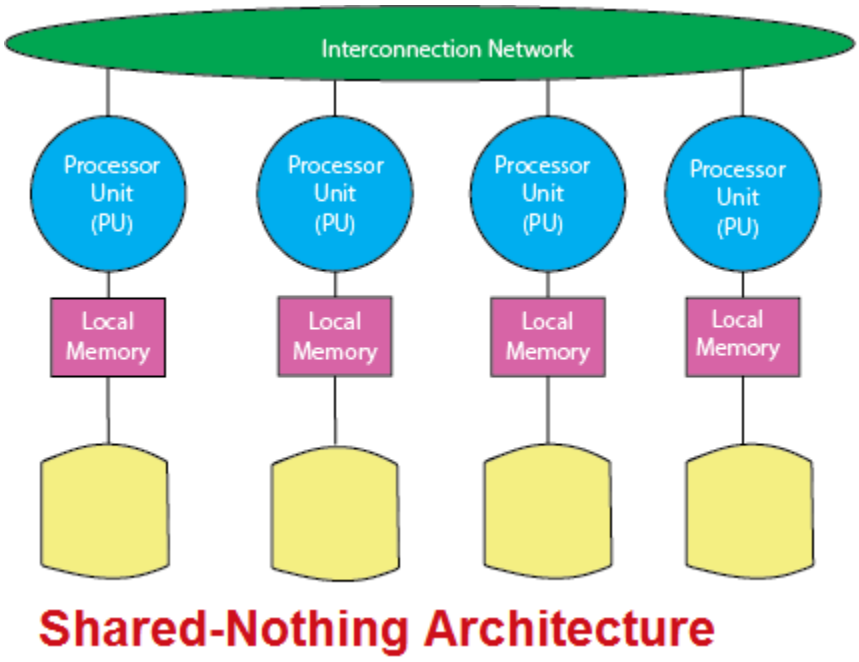


Fig. 4.22 Shared Nothing Architecture

10. Explain in detail between Data Warehouse Tools.

Data Warehouse Tools

The tools that allow sourcing of data contents and formats accurately and external data stores into the data warehouse have to perform several essential tasks that contain:

- Data consolidation and integration.
- Data transformation from one form to another form.
- Data transformation and calculation based on the function of business rules that force transformation.
- Metadata synchronization and management, which includes storing or updating metadata about source files, transformation actions, loading formats, and events.

There are several selection criteria which should be considered while implementing a data warehouse:

1. The ability to identify the data in the data source environment that can be read by the tool is necessary.
2. Support for flat files, indexed files, and legacy DBMSs is critical.
3. The capability to merge records from multiple data stores is required in many installations.
4. The specification interface to indicate the information to be extracted and conversation are essential.
5. The ability to read information from repository products or data dictionaries is desired.
6. The code develops by the tool should be completely maintainable.
7. Selective data extraction of both data items and records enables users to extract only the required data.
8. A field-level data examination for the transformation of data into information is needed.

9. The ability to perform data type and the character-set translation is a requirement when moving data between incompatible systems.
10. The ability to create aggregation, summarization and derivation fields and records are necessary.
11. Vendor stability and support for the products are components that must be evaluated carefully.

11. Explain in detail about Data Warehouse Software Components.

Data Warehouse Software Components

- A warehousing team will require different types of tools during a warehouse project. These software products usually fall into one or more of the categories illustrated, as shown in the figure 4.23.

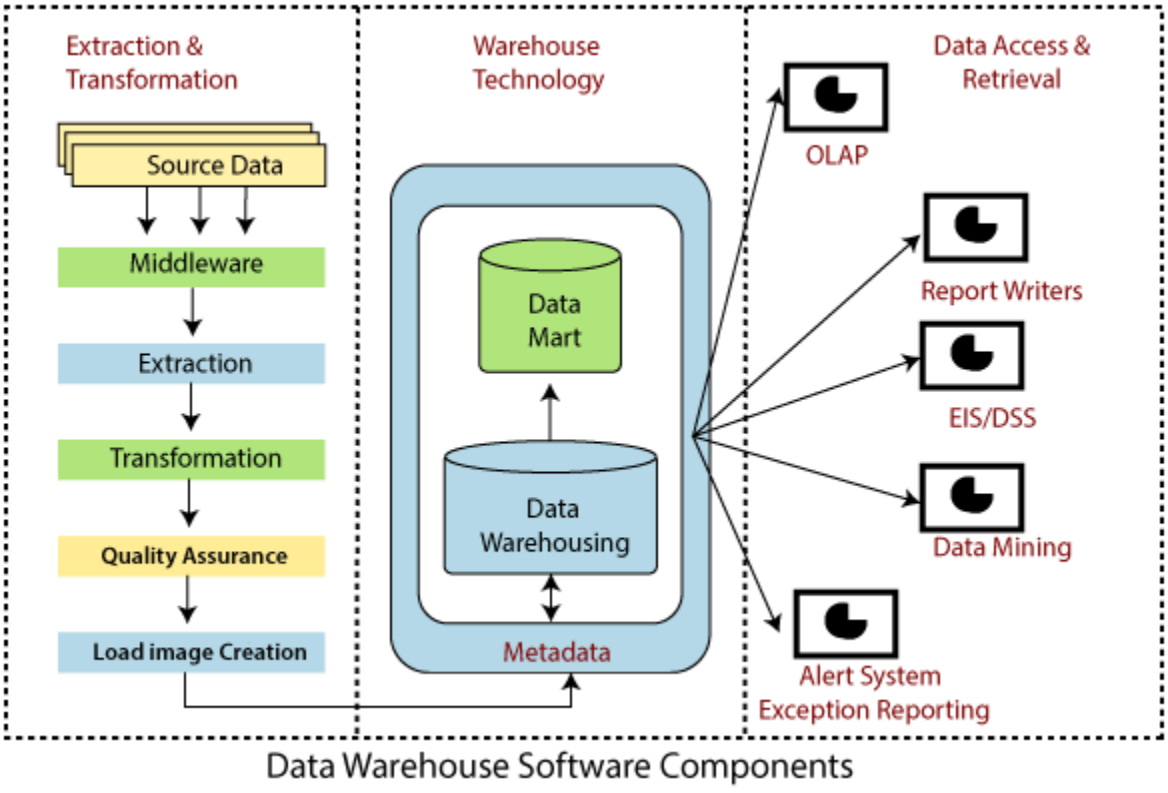


Fig. 4.23 Data Warehouse Software Components

Extraction and Transformation

- The warehouse team needs tools that can extract, transform, integrate, clean, and load information from a source system into one or more data warehouse databases. Middleware and gateway products may be needed for warehouses that extract a record from a host-based source system.

Warehouse Storage

- Software products are also needed to store warehouse data and their accompanying metadata. Relational database management systems are well suited to large and growing warehouses.

Data access and retrieval

- Different types of software are needed to access, retrieve, distribute, and present warehouse data to its end-clients.

12. Explain in detail about Steps of Dimensional Modelling.

Steps of Dimensional Modelling

The accuracy in creating your Dimensional modeling determines the success of your data warehouse implementation. Here are the steps to create Dimension Model as in figure 4.24

- Identify Business Process
- Identify Grain (level of detail)
- Identify Dimensions
- Identify Facts
- Build Star

The model should describe the Why, How much, When/Where/Who and What of your business process

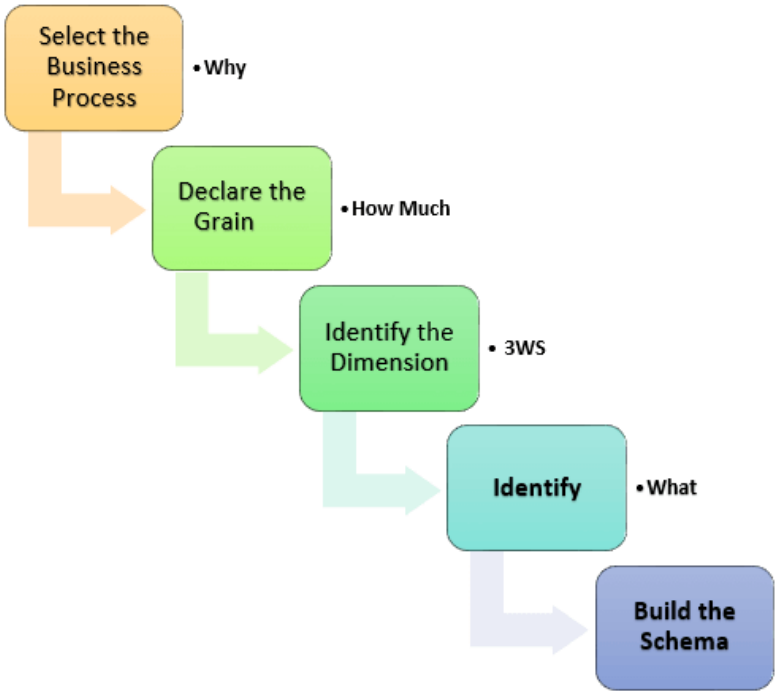


Fig. 4.24 Steps of Dimensional Modelling

Step 1) Identify the Business Process

- Identifying the actual business process a data warehouse should cover. This could be Marketing, Sales, HR, etc. as per the data analysis needs of the organization.
- The selection of the Business process also depends on the quality of data available for that process. It is the most important step of the Data Modelling process, and a failure here would have cascading and irreparable defects.
- To describe the business process, you can use plain text or use basic Business Process Modelling Notation (BPMN) or Unified Modelling Language (UML).

Step 2) Identify the Grain

- The Grain describes the level of detail for the business problem/solution. It is the process of identifying the lowest level of information for any table in your data warehouse.
- If a table contains sales data for every day, then it should be daily granularity. If a table contains total sales data for each month, then it has monthly granularity.

Step 3) Identify the Dimensions

- Dimensions are nouns like date, store, inventory, etc. These dimensions are where all the data should be stored.
- For example, the date dimension may contain data like a year, month and weekday.

Example of Dimensions:

- The CEO at an MNC wants to find the sales for specific products in different locations on a daily basis.
- Dimensions: Product, Location and Time
- Attributes: For Product: Product key (Foreign Key), Name, Type, Specifications
- Hierarchies: For Location: Country, State, City, Street Address, Name

Step 4) Identify the Fact

- This step is co-associated with the business users of the system because this is where they get access to data stored in the data warehouse.
- Most of the fact table rows are numerical values like price or cost per unit, etc.

Example of Facts:

- The CEO at an MNC wants to find the sales for specific products in different locations on a daily basis.
- The fact here is Sum of Sales by product by location by time.

Step 5) Build Schema

- In this step, you implement the Dimension Model. A schema is nothing but the database structure (arrangement of tables). There are two popular schemas

Star Schema

- The star schema architecture is easy to design. It is called a star schema because diagram resembles a star, with points radiating from a center. The center of the star consists of the fact table, and the points of the star is dimension tables.
- The fact tables in a star schema which is third normal form whereas dimensional tables are de-normalized.

Snowflake Schema

- The snowflake schema is an extension of the star schema. In a snowflake schema, each dimension are normalized and connected to more dimension tables.

13. Propose a fact constellation schema for a healthcare data warehouse to support complex analytical queries. [Nov 2024]

- A healthcare fact constellation schema could include separate fact tables for "Patient Admissions", "Medical Procedures", "Medication Dispensing", and "Lab Results", all sharing common dimension tables like "Patient", "Doctor", "Location", "Date", and "Diagnosis", enabling complex analysis across various aspects of patient care while reusing shared dimensional information.

Key elements of the schema:

Fact Tables:

Patient Admissions:

Patient ID (FK to Patient dimension)

Admission Date (FK to Date dimension)

Admitting Doctor ID (FK to Doctor dimension)

Admission Type (e.g., Emergency, Elective)

Length of Stay

Medical Procedures:

Patient ID (FK to Patient dimension)

Procedure Date (FK to Date dimension)

Procedure Code (FK to Procedure Code dimension)

Performing Doctor ID (FK to Doctor dimension)

Procedure Cost

Medication Dispensing:

Patient ID (FK to Patient dimension)

Medication Date (FK to Date dimension)

Medication Code (FK to Medication Code dimension)

Prescribing Doctor ID (FK to Doctor dimension)

Dosage

Lab Results:

Patient ID (FK to Patient dimension)

Lab Test Date (FK to Date dimension)

Lab Test Code (FK to Lab Test Code dimension)

Lab Result Value

Dimension Tables:

Patient:

Patient ID (PK)

Patient Name

Date of Birth

Gender

Insurance Information

Contact Details

Doctor:

Doctor ID (PK)

Doctor Name

Specialty

Location:

Location ID (PK)

Location Name

Facility Type (Hospital, Clinic)

Date:

Date Key (PK)

Date

Day of Week

Month

Year

Diagnosis:

Diagnosis Code (PK)

Diagnosis Description

Benefits of this schema:

- Flexible Analysis: Enables complex queries across different aspects of patient care by combining data from multiple fact tables using shared dimensions.
- Efficient Data Storage: Reduces redundancy by sharing common dimension tables across various fact tables.
- Scalability: Allows for easy addition of new fact tables as healthcare needs evolve.

Important Considerations:

- Data Quality: Ensure accurate and consistent data from source systems before loading into the data warehouse.
- Granularity: Determine the appropriate level of detail for each fact table based on analysis requirements.
- Security and Privacy: Implement robust access controls to protect sensitive patient data.

14. Explain how does a snowflake schema differ from a star schema in terms of normalization? [Nov 2024]

1. Normalization of dimension tables

- The snowflake schema is a fully normalized data structure. Dimensional hierarchies (such as city > country > region) are stored in separate dimensional tables.

- On the other hand, star schema dimensions are denormalized. Denormalization refers to the repeating of the same values within a table.

2. Data redundancy

- Star schema stores redundant data in dimension tables, while snowflake schema fully normalizes dimension tables and avoids data redundancy.
- For example, a star schema would repeat the values in field customer_address_country for each order from the same country.
- The redundancy, or duplicated entries, occurs because of the denormalization vs normalization schema design.

3. Query complexity

- A simple star schema leads to simple query writing. Because the fact table is joined to only one level of dimensional tables, analysts do not need to write multiple joins.
- On the other hand, snowflake schemas require a more complex query design. Because of complex relationships between the fact table and its dimensional tables, more joins are needed to link the additional tables. This causes an additional overhead when writing analytical queries.

4. Query performance

- The query execution time is faster in star schemas. Because they require a single join between a fact and its set of attributes in dimensional tables, a star schema acts almost as a single table for query lookups.
- In contrast, snowflake schemas require complex joins of dimensional tables with their own sub-dimensional or supra-dimensional tables. This slows down query processing and can affect other OLAP products such as cube processing.

5. Disk space

- Star schemas might run queries faster, but they require more storage space than snowflake schemas because of their data redundancy.

6. Data integrity

- Data integrity is more at risk in star schemas than snowflake schemas. Because data is stored redundantly, multiple copies of the same data exist in the star

schema's dimensional tables. This means new inserts, updates, or deletes can compromise the integrity of data.

- In contrast, the snowflake schema is less prone to data integrity issues, because it fully normalizes dimensional tables, storing dimension data only once in the appropriate table.

7. Set up and maintenance

- Star schemas are easier to design and set up. Because they are represented by simple relationships, it is easy for a data engineer or data architect to set up an appropriate star schema.
- On the other hand, star schemas are harder to maintain than snowflake schemas. As new data is ingested into the data warehouse, star schemas become harder to maintain and check against data integrity violations.

15. Explain the relationship between the Load manager and the warehouse manager in a data warehousing process. [Nov 2024]

Load Manager

- This component performs the operations required to extract and load process.
- The size and complexity of the load manager varies between specific solutions from one data warehouse to other.

Load Manager Architecture

The load manager as in figure 4.25 performs the following functions:

- Extract the data from source system.
- Fast Load the extracted data into temporary data store.
- Perform simple transformations into structure similar to the one in the data warehouse.

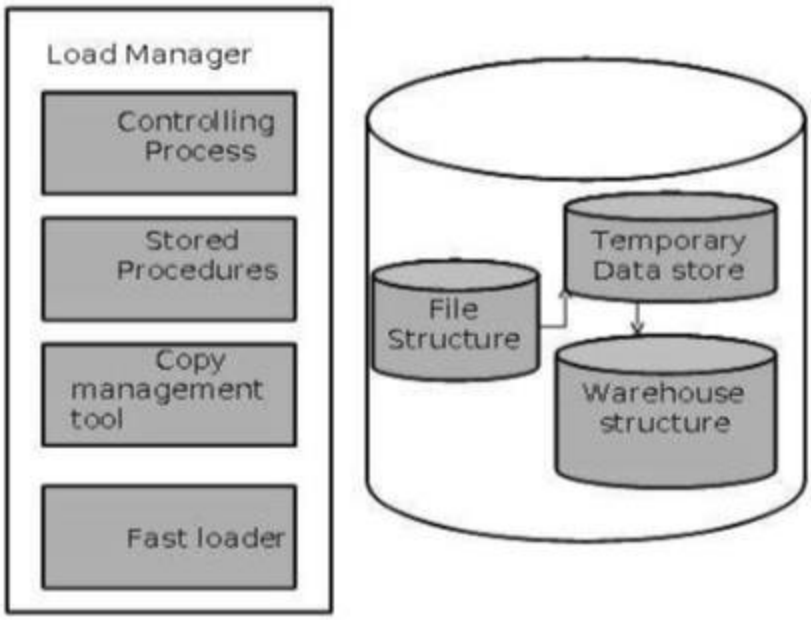


Fig. 4.25 Load Manager

Extract Data from Source

- The data is extracted from the operational databases or the external information providers. Gateways is the application programs that are used to extract data. It is supported by underlying DBMS and allows client program to generate SQL to be executed at a server.
- Open Database Connection (ODBC), Java Database Connection (JDBC), are examples of gateway.

Fast Load

- In order to minimize the total load window the data need to be loaded into the warehouse in the fastest possible time.
- The transformations affects the speed of data processing.
- It is more effective to load the data into relational database prior to applying transformations and checks.
- Gateway technology proves to be not suitable, since they tend not be performant when large data volumes are involved.

Simple Transformations

While loading it may be required to perform simple transformations. After this has been completed we are in position to do the complex checks. Suppose we are loading the EPOS sales transaction we need to perform the following checks:

- Strip out all the columns that are not required within the warehouse.
- Convert all the values to required data types.

Warehouse Manager

- A warehouse manager is responsible for the warehouse management process. It consists of third-party system software, C programs, and shell scripts.
- The size and complexity of warehouse managers varies between specific solutions.

Warehouse Manager Architecture

A warehouse manager includes the following as in figure 4.26:

- The controlling process
- Stored procedures or C with SQL
- Backup/Recovery tool
- SQL Scripts

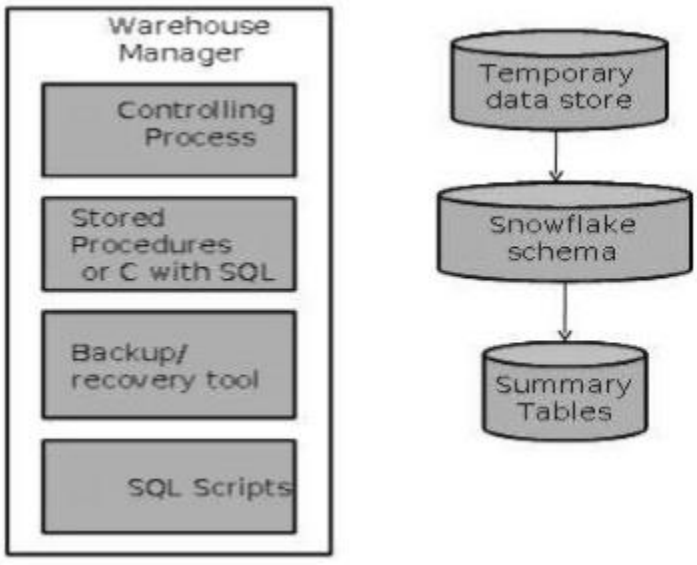


Fig.4.26 Warehouse Manager Architecture

Operations Performed by Warehouse Manager

- A warehouse manager analyzes the data to perform consistency and referential integrity checks.
- Creates indexes, business views, partition views against the base data.
- Generates new aggregations and updates existing aggregations. Generates normalizations.
- Transforms and merges the source data into the published data warehouse.
- Backup the data in the data warehouse.
- Archives the data that has reached the end of its captured life.

Note: A warehouse Manager also analyzes query profiles to determine index and aggregations are appropriate.

Query Manager

- Query manager is responsible for directing the queries to the suitable tables.
- By directing the queries to appropriate tables, the speed of querying and response generation can be increased.
- Query manager is responsible for scheduling the execution of the queries posed by the user.

Query Manager Architecture

The following figure 4.27 shows the architecture of a query manager.

It includes the following:

- Query redirection via C tool or RDBMS
- Stored procedures
- Query management tool
- Query scheduling via C tool or RDBMS
- Query scheduling via third-party software

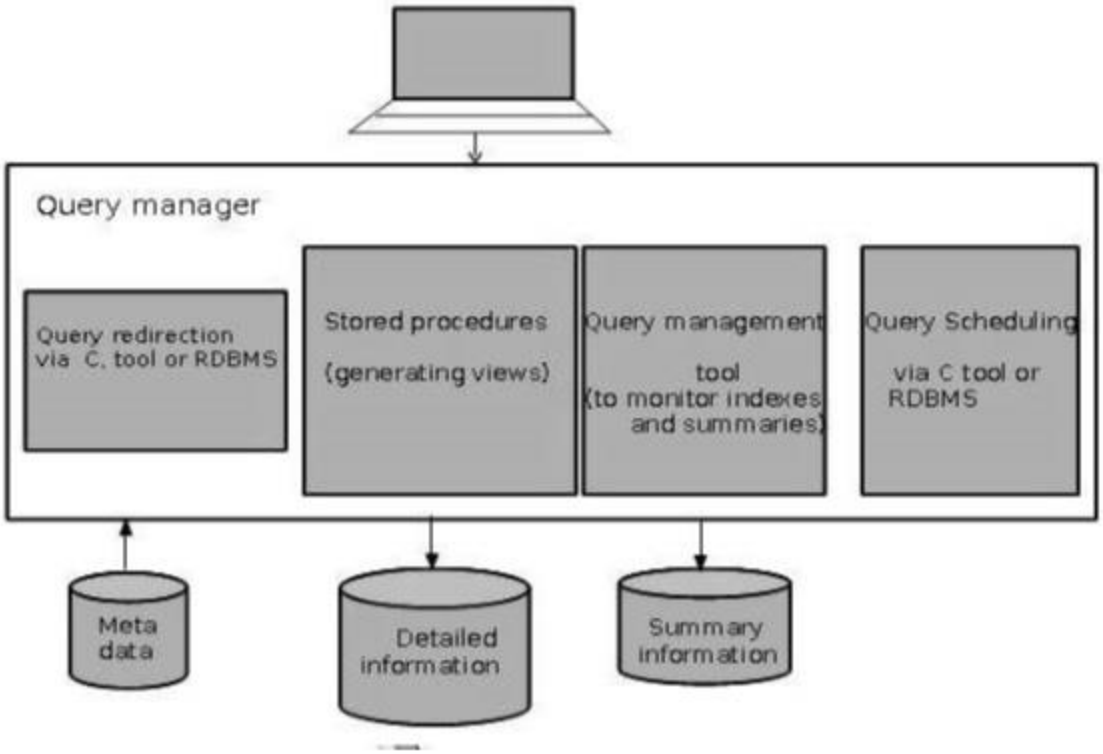


Fig.4.27 Query Manager Architecture

UNIT V SYSTEM & PROCESS MANAGERS 6

Data Warehousing System Managers: System Configuration Manager- System Scheduling Manager - System Event Manager - System Database Manager - System Backup Recovery Manager - Data Warehousing Process Managers: Load Manager – Warehouse Manager- Query Manager – Tuning – Testing

PART A

1. **Recall the responsibilities of a Data warehousing system configuration manager. [Nov 2024]**
    - A Data Warehousing System Configuration Manager is primarily responsible for designing, implementing, and maintaining the architecture of a data warehouse system, ensuring optimal data flow, storage, and access for analysis,
    - By managing data source connections, data transformations, and database configurations, while also monitoring performance and addressing technical issues to maintain data integrity and security.

2. **Outline the primary task of a data warehousing system Backup Recovery Manager. [Nov 2024]**
    - The primary task of a data warehousing system Backup Recovery Manager is to ensure the integrity and availability of data stored within the data warehouse by creating and maintaining backups, allowing for the restoration of data to a consistent state in the event of system failures, data corruption, or accidental deletion, effectively protecting against data loss and enabling business continuity;
    - This includes managing backup schedules, performing data recovery operations, and monitoring the backup process to identify and resolve issues.

3. **Define load manager.**
    - A load manager performs the operations required to extract and load the process.
    - The size and complexity of load manager varies between specific solutions from data warehouse to data warehouse.

4. Define the functions of a load manager.

- A load manager extracts data from the source system. Fast load the extracted data into temporary data store.
- Perform simple transformations into structure similar to the one in the data warehouse.

5. Define a warehouse manager.

- Warehouse manager is responsible for the warehouse management process.
- The warehouse manager consist of third party system software, C programs and shell scripts.
- The size and complexity of warehouse manager varies between specific solutions.

6. Define the functions of a warehouse manager.

- The warehouse manager performs consistency and referential integrity checks, creates the indexes, business views, partition views against the base data, transforms and merge the source data into the temporary store into the published data warehouse, backs up the data in the data warehouse, and archives the data that has reached the end of its captured life.

7. What is the responsibility for System Configuration Manager?

- The system configuration manager is responsible for the management of the setup and configuration of data warehouse.

8. What is the responsibility for System Scheduling Manager?

- System Scheduling Manager is responsible for the successful implementation of the data warehouse. Its purpose is to schedule ad hoc queries.

9. Write short notes on Event Manager.

- The event manager is a kind of a software. The event manager manages the events that are defined on the data warehouse system.
- We cannot manage the data warehouse manually because the structure of data warehouse is very complex.
- Therefore we need a tool that automatically handles all the events without any intervention of the user.

10. Write short notes on System Backup Recovery Manager.

- The backup and recovery tool makes it easy for operations and management staff to back-up the data.

- The system backup manager must be integrated with the schedule manager software being used.

11. What are the features of System Backup Recovery Manager?

- The important features that are required for the management of backups are as follows –
  - Scheduling
  - Backup data tracking
  - Database awareness

12. What is meant Process Manager and its types?

- Process managers are responsible for maintaining the flow of data both into and out of the data warehouse.
- There are three different types of process managers –
  - Load manager
  - Warehouse manager
  - Query manager

13. What is meant Data Warehouse Load Manager?

- Load manager performs the operations required to extract and load the data into the database.
- The size and complexity of a load manager varies between specific solutions from one data warehouse to another.

14. What is the responsibility for Warehouse Manager?

- The warehouse manager is responsible for the warehouse management process.
- It consists of a third-party system software, C programs, and shell scripts.
- The size and complexity of a warehouse manager varies between specific solutions.

15. What is the responsibility for Query Manager?

- The query manager is responsible for directing the queries to suitable tables.
- By directing the queries to appropriate tables, it speeds up the query request and response process.

In addition, the query manager is responsible for scheduling the execution of the queries posted by the user

16. What is meant Data Warehouse Testing and its types?

- Testing is very important for data warehouse systems to make them work correctly and efficiently.
- There are three basic levels of testing performed on a data warehouse –
  - Unit testing
  - Integration testing
  - System testing

17. What are the difficulties in Data Warehouse Tuning?

- Tuning a data warehouse is a difficult procedure due to following reasons –
  - Data warehouse is dynamic; it never remains constant.
  - It is very difficult to predict what query the user is going to post in the future.
  - Business requirements change with time.
  - Users and their profiles keep changing.
  - The user can switch from one group to another.
  - The data load on the warehouse also changes with time.

18. What are the objective measures of performance?

- Average query response time
- Scan rates
- Time used per day query
- Memory usage per process
- I/O throughput rates

19. What is meant Integrity Checks?

- Integrity checking highly affects the performance of the load.
- Integrity checks need to be limited because they require heavy processing power.
- Integrity checks should be applied on the source system to avoid performance degrade of data load.

20. What are the types of Tuning Queries in Data Warehouse?

Two kinds of queries in data warehouse

- Fixed queries
- Ad hoc queries

- Fixed queries are well defined. Following are the examples of fixed queries −
  - regular reports

- Canned queries
- Common aggregations

- To understand ad hoc queries, it is important to know the ad hoc users of the data warehouse.

21. Write about Data Warehouse Management Tools.

- **Management Tools.** − It is required to test all the management tools during system testing. Here is the list of tools that need to be tested.
  - Event manager
  - System manager
  - Database manager
  - Configuration manager
  - Backup recovery manager

22. What are the three ways of Testing the Database in Data Warehouse?

- Testing the database manager and monitoring tools
- Testing database features
- Testing database performance

21. Define Query Manager. [NOV/DEC 2023]

The query manager is responsible for directing the queries to suitable tables. By directing the queries to appropriate tables, it speeds up the query request and response process. In addition, the query manager is responsible for scheduling the execution of the queries posted by the user.

22. What are the various sources for data warehouses? [NOV/DEC 2023]

- Operational systems
- External data sources
- Flat files
- Legacy systems
- Cloud-based applications
- IoT devices
- Web and social media data

PART B

1. Explain in detail about Data Warehousing System Managers.

- System management is mandatory for the successful implementation of a data warehouse. The most important system managers are –
  - System configuration manager
  - System scheduling manager
  - System event manager
  - System database manager
  - System backup recovery manager

System Configuration Manager

- The system configuration manager is responsible for the management of the setup and configuration of data warehouse.
- The structure of configuration manager varies from one operating system to another.
- In Unix structure of configuration, the manager varies from vendor to vendor.
- Configuration managers have single user interface.
- The interface of configuration manager allows us to control all aspects of the system.
- **Note** − The most important configuration tool is the I/O manager.

System Scheduling Manager

- System Scheduling Manager is responsible for the successful implementation of the data warehouse. Its purpose is to schedule ad hoc queries.
- Every operating system has its own scheduler with some form of batch control mechanism.
- The list of features a system scheduling manager must have is as follows –
  - Work across cluster or MPP boundaries
  - Deal with international time differences
  - Handle job failure
  - Handle multiple queries
  - Support job priorities
  - Restart or re-queue the failed jobs

  - Notify the user or a process when job is completed
  - Maintain the job schedules across system outages
  - Re-queue jobs to other queues
  - Support the stopping and starting of queues
  - Log Queued jobs
  - Deal with inter-queue processing
- **Note –** The above list can be used as evaluation parameters for the evaluation of a good scheduler.
- Some important jobs that a scheduler must be able to handle are as follows –
  - Daily and ad hoc query scheduling
  - Execution of regular report requirements
  - Data load
  - Data processing
  - Index creation
  - Backup
  - Aggregation creation
  - Data transformation
- **Note** – If the data warehouse is running on a cluster or MPP architecture, then the system scheduling manager must be capable of running across the architecture.

System Event Manager

- The event manager is a kind of a software. The event manager manages the events that are defined on the data warehouse system.
- We cannot manage the data warehouse manually because the structure of data warehouse is very complex.
- Therefore we need a tool that automatically handles all the events without any intervention of the user.
- **Note** – The Event manager monitors the events occurrences and deals with them. The event manager also tracks the myriad of things that can go wrong on this complex data warehouse system.

Events

- Events are the actions that are generated by the user or the system itself. It may be noted that the event is a measurable, observable, occurrence of a defined action.
- Given below is a list of common events that are required to be tracked.
  - Hardware failure
  - Running out of space on certain key disks
  - A process dying
  - A process returning an error
  - CPU usage exceeding an 805 threshold
  - Internal contention on database serialization points
  - Buffer cache hit ratios exceeding or failure below threshold
  - A table reaching to maximum of its size
  - Excessive memory swapping
  - A table failing to extend due to lack of space
  - Disk exhibiting I/O bottlenecks
  - Usage of temporary or sort area reaching a certain thresholds
  - Any other database shared memory usage
- The most important thing about events is that they should be capable of executing on their own.
- Event packages define the procedures for the predefined events. The code associated with each event is known as event handler.
- This code is executed whenever an event occurs.

System and Database Manager

- System and database manager may be two separate pieces of software, but they do the same job.
- The objective of these tools is to automate certain processes and to simplify the execution of others.
- The criteria for choosing a system and the database manager are as follows
  - increase user's quota.
  - assign and de-assign roles to the users
  - assign and de-assign the profiles to the users

- perform database space management
- monitor and report on space usage
- tidy up fragmented and unused space
- add and expand the space
- add and remove users
- manage user password
- manage summary or temporary tables
- assign or deassign temporary space to and from the user
- reclaim the space form old or out-of-date temporary tables
- manage error and trace logs
- to browse log and trace files
- redirect error or trace information
- switch on and off error and trace logging
- perform system space management
- monitor and report on space usage
- clean up old and unused file directories
- add or expand space.

System Backup Recovery Manager

- The backup and recovery tool makes it easy for operations and management staff to back-up the data.
- Note that the system backup manager must be integrated with the schedule manager software being used.
- The important features that are required for the management of backups are as follows –
  - Scheduling
  - Backup data tracking
  - Database awareness
- Backups are taken only to protect against data loss. Following are the important points to remember –
  - The backup software will keep some form of database of where and when the piece of data was backed up.
  - The backup recovery manager must have a good front-end to that database.
  - The backup recovery software should be database aware.

- Being aware of the database, the software then can be addressed in database terms, and will not perform backups that would not be viable.

2. Explain in detail about Data Warehousing - Process Managers.

Explain about various data warehousing process managers, its collaboration and interaction to ensure operations and performance of the entire data warehousing system. [Nov 2024]

- Process managers are responsible for maintaining the flow of data both into and out of the data warehouse.
- There are three different types of process managers –
  - Load manager
  - Warehouse manager
  - Query manager

Data Warehouse Load Manager

- Load manager performs the operations required to extract and load the data into the database.
- The size and complexity of a load manager varies between specific solutions from one data warehouse to another.

Load Manager Architecture

The load manager in figure 5.1 does performs the following functions –

- Extract data from the source system.
- Fast load the extracted data into temporary data store.
- Perform simple transformations into structure similar to the one in the data warehouse.

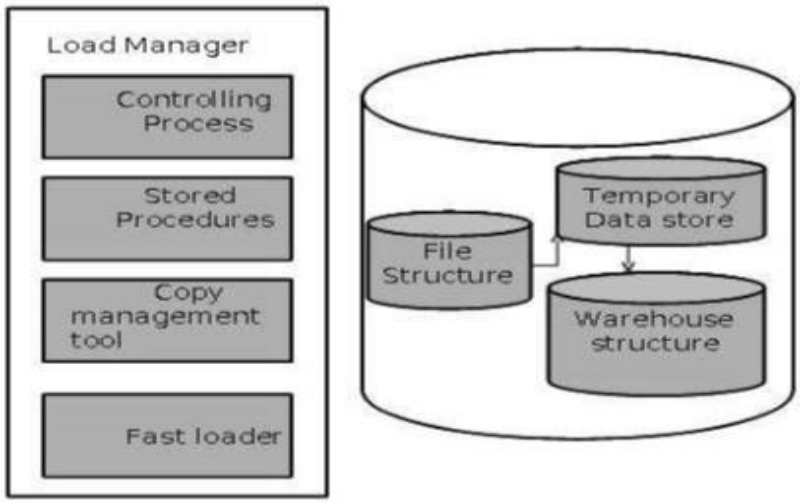


Fig.5.1 Load Manager

Extract Data from Source

- The data is extracted from the operational databases or the external information providers.
- Gateways are the application programs that are used to extract data.
- It is supported by underlying DBMS and allows the client program to generate SQL to be executed at a server.
- Open Database Connection (ODBC) and Java Database Connection (JDBC) are examples of gateway.

Fast Load

- In order to minimize the total load window, the data needs to be loaded into the warehouse in the fastest possible time.
- Transformations affect the speed of data processing.
- It is more effective to load the data into a relational database prior to applying transformations and checks.
- Gateway technology is not suitable, since they are inefficient when large data volumes are involved.

Simple Transformations

- While loading, it may be required to perform simple transformations. After completing simple transformations, we can do complex checks.

- Suppose we are loading the EPOS sales transaction, we need to perform the following checks –
    - Strip out all the columns that are not required within the warehouse.
    - Convert all the values to required data types.

Warehouse Manager

- The warehouse manager is responsible for the warehouse management process.
- It consists of a third-party system software, C programs, and shell scripts.
- The size and complexity of a warehouse manager varies between specific solutions.

Warehouse Manager Architecture

A warehouse manager in figure 5.2 includes the following –

- The controlling process
- Stored procedures or C with SQL
- Backup/Recovery tool
- SQL scripts

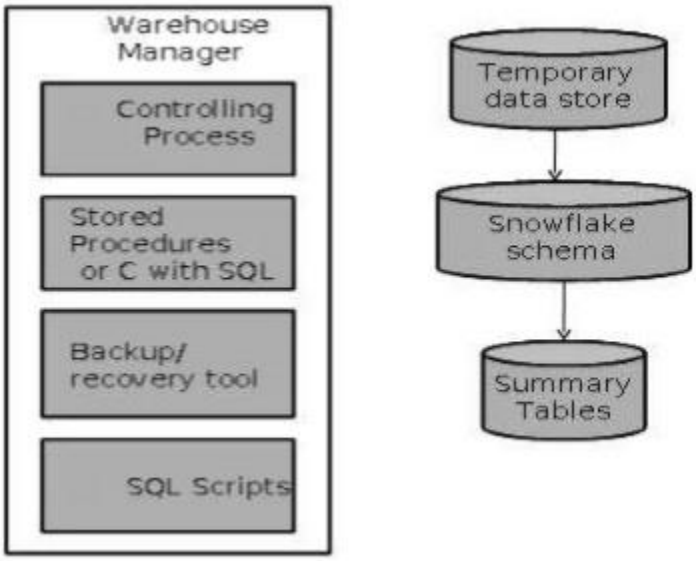


Fig.5.2 Warehouse Manager Architecture

Functions of Warehouse Manager

- A warehouse manager performs the following functions −
  - Analyzes the data to perform consistency and referential integrity checks.
  - Creates indexes, business views, partition views against the base data.
  - Generates new aggregations and updates the existing aggregations.
  - Generates normalizations.
  - Transforms and merges the source data of the temporary store into the published data warehouse.
  - Backs up the data in the data warehouse.
  - Archives the data that has reached the end of its captured life.
- Note − A Warehouse Manager analyzes query profiles to determine whether the index and aggregations are appropriate.

Query Manager

- The query manager is responsible for directing the queries to suitable tables.
- By directing the queries to appropriate tables, it speeds up the query request and response process.
- In addition, the query manager is responsible for scheduling the execution of the queries posted by the user.

Query Manager Architecture

A query manager in figure 5.3 includes the following components –

- Query redirection via C tool or RDBMS
- Stored procedures
- Query management tool
- Query scheduling via C tool or RDBMS
- Query scheduling via third-party software

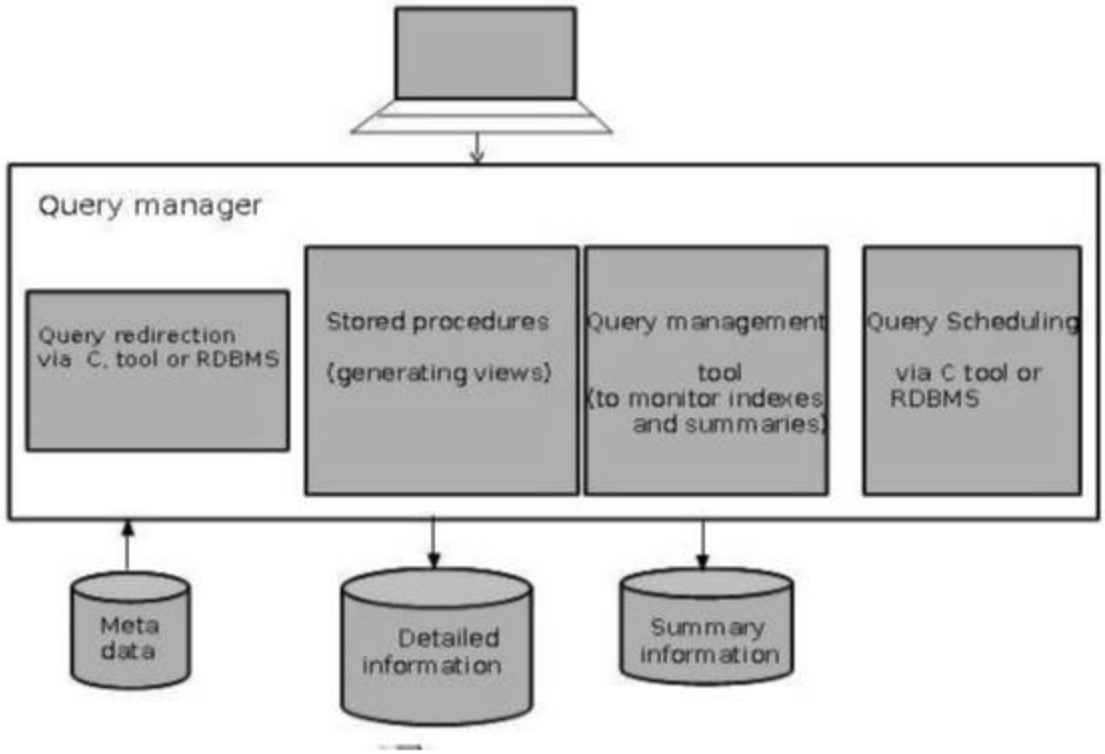


Fig. 5.3 Query Manager Architecture

Functions of Query Manager

- It presents the data to the user in a form they understand.
- It schedules the execution of the queries posted by the end-user.
- It stores query profiles to allow the warehouse manager to determine which indexes and aggregations are appropriate.

3. Explain in detail about Data Warehousing – Tuning.

- A data warehouse keeps evolving and it is unpredictable what query the user is going to post in the future.
- Therefore it becomes more difficult to tune a data warehouse system.

Difficulties in Data Warehouse Tuning

- Tuning a data warehouse is a difficult procedure due to following reasons –
  - Data warehouse is dynamic; it never remains constant.
  - It is very difficult to predict what query the user is going to post in the future.
  - Business requirements change with time.

  - Users and their profiles keep changing.
  - The user can switch from one group to another.
  - The data load on the warehouse also changes with time.
- Note − It is very important to have a complete knowledge of data warehouse.

Performance Assessment

Here is a list of objective measures of performance –

- Average query response time
- Scan rates
- Time used per day query
- Memory usage per process
- I/O throughput rates

Following are the points to remember.

- It is necessary to specify the measures in service level agreement (SLA).
- It is of no use trying to tune response time, if they are already better than those required.
- It is essential to have realistic expectations while making performance assessment.
- It is also essential that the users have feasible expectations.
- To hide the complexity of the system from the user, aggregations and views should be used.
- It is also possible that the user can write a query you had not tuned for.

Data Load Tuning

- Data load is a critical part of overnight processing. Nothing else can run until data load is complete. This is the entry point into the system.
- **Note –** If there is a delay in transferring the data, or in arrival of data then the entire system is affected badly. Therefore it is very important to tune the data load first.
- There are various approaches of tuning data load that are discussed below –
  - The very common approach is to insert data using the SQL Layer. In this approach, normal checks and constraints need to be performed.

- When the data is inserted into the table, the code will run to check for enough space to insert the data.
- If sufficient space is not available, then more space may have to be allocated to these tables. These checks take time to perform and are costly to CPU.
- The second approach is to bypass all these checks and constraints and place the data directly into the preformatted blocks.
- These blocks are later written to the database. It is faster than the first approach, but it can work only with whole blocks of data. This can lead to some space wastage.
- The third approach is that while loading the data into the table that already contains the table, we can maintain indexes.
- The fourth approach says that to load the data in tables that already contain data, drop the indexes & recreate them when the data load is complete.
- The choice between the third and the fourth approach depends on how much data is already loaded and how many indexes need to be rebuilt.

Integrity Checks

Integrity checking highly affects the performance of the load. Following are the points to remember −

- Integrity checks need to be limited because they require heavy processing power.
- Integrity checks should be applied on the source system to avoid performance degrade of data load.

Tuning Queries

We have two kinds of queries in data warehouse −

- Fixed queries
- Ad hoc queries

Fixed Queries

- Fixed queries are well defined. Following are the examples of fixed queries −
  - regular reports
  - Canned queries

- Common aggregations

- Tuning the fixed queries in a data warehouse is same as in a relational database system. The only difference is that the amount of data to be queried may be different. It is good to store the most successful execution plan while testing fixed queries. Storing these executing plan will allow us to spot changing data size and data skew, as it will cause the execution plan to change.
- **Note –** We cannot do more on fact table but while dealing with dimension tables or the aggregations, the usual collection of SQL tweaking, storage mechanism, and access methods can be used to tune these queries.

Ad hoc Queries

- To understand ad hoc queries, it is important to know the ad hoc users of the data warehouse. For each user or group of users, you need to know the following
  - The number of users in the group
  - Whether they use ad hoc queries at regular intervals of time
  - Whether they use ad hoc queries frequently
  - Whether they use ad hoc queries occasionally at unknown intervals.
  - The maximum size of query they tend to run
  - The average size of query they tend to run
  - Whether they require drill-down access to the base data
  - The elapsed login time per day
  - The peak time of daily usage
  - The number of queries they run per peak hour
- It is important to track the user's profiles and identify the queries that are run on a regular basis.
- It is also important that the tuning performed does not affect the performance.
- Identify similar and ad hoc queries that are frequently run.
- If these queries are identified, then the database will change and new indexes can be added for those queries.
- If these queries are identified, then new aggregations can be created specifically for those queries that would result in their efficient execution.

4. Explain in detail about Data warehousing – Testing.

Justify the statement "Testing process contribute to the overall quality a data warehousing system", and describe the types of tests conducted on the system.

[Nov 2024]

- Data warehouse testing is a crucial aspect of data management, as it guarantees the reliability and accuracy of data for decision-making.
- It is essential to implement a comprehensive testing strategy encompassing data integration, quality, performance, and security to maintain the credibility of the data warehouse.

- Testing is very important for data warehouse systems to make them work correctly and efficiently.
- There are three basic levels of testing performed on a data warehouse –
  - Unit testing
  - Integration testing
  - System testing

Unit Testing

- In unit testing, each component is separately tested.
- Each module, i.e., procedure, program, SQL Script, Unix shell is tested.
- This test is performed by the developer.

Integration Testing

- In integration testing, the various modules of the application are brought together and then tested against the number of inputs.
- It is performed to test whether the various components do well after integration.

System Testing

- In system testing, the whole data warehouse application is tested together.
- The purpose of system testing is to check whether the entire system works correctly together or not.
- System testing is performed by the testing team.
- Since the size of the whole data warehouse is very large, it is usually possible to perform minimal system testing before the test plan can be enacted.

Test Schedule

- First of all, the test schedule is created in the process of developing the test plan. In this schedule, we predict the estimated time required for the testing of the entire data warehouse system.
- There are different methodologies available to create a test schedule, but none of them are perfect because the data warehouse is very complex and large. Also the data warehouse system is evolving in nature.
- One may face the following issues while creating a test schedule –
    - A simple problem may have a large size of query that can take a day or more to complete, i.e., the query does not complete in a desired time scale.
    - There may be hardware failures such as losing a disk or human errors such as accidentally deleting a table or overwriting a large table.
- Note − Due to the above-mentioned difficulties, it is recommended to always double the amount of time you would normally allow for testing.

Testing Backup Recovery

Testing the backup recovery strategy is extremely important. Here is the list of scenarios for which this testing is needed –

- Media failure
- Loss or damage of table space or data file
- Loss or damage of redo log file
- Loss or damage of control file
- Instance failure
- Loss or damage of archive file
- Loss or damage of table
- Failure during data failure

Testing Operational Environment

- There are a number of aspects that need to be tested. These aspects are listed below.
- **Security** − A separate security document is required for security testing. This document contains a list of disallowed operations and devising tests for each.

- **Scheduler** − Scheduling software is required to control the daily operations of a data warehouse. It needs to be tested during system testing. The scheduling software requires an interface with the data warehouse, which will need the scheduler to control overnight processing and the management of aggregations.
- **Disk Configuration.** − Disk configuration also needs to be tested to identify I/O bottlenecks. The test should be performed with multiple times with different settings.
- **Management Tools.** − It is required to test all the management tools during system testing. Here is the list of tools that need to be tested.
    - Event manager
    - System manager
    - Database manager
    - Configuration manager
    - Backup recovery manager

Testing the Database

The database is tested in the following three ways –

- **Testing the database manager and monitoring tools** − To test the database manager and the monitoring tools, they should be used in the creation, running, and management of test database.
- **Testing database features −** Here is the list of features that we have to test −
    - Querying in parallel
    - Create index in parallel
    - Data load in parallel
- **Testing database performance –**
    - Query execution plays a very important role in data warehouse performance measures. There are sets of fixed queries that need to be run regularly and they should be tested.
    - To test ad hoc queries, one should go through the user requirement document and understand the business completely.
    - Take time to test the most awkward queries that the business is likely to ask against different index and aggregation strategies.

Testing the Application

- All the managers should be integrated correctly and work in order to ensure that the end-to-end load, index, aggregate and queries work as per the expectations.
- Each function of each manager should work correctly
- It is also necessary to test the application over a period of time.
- Week end and month-end tasks should also be tested.

Logistic of the Test

- The aim of system test is to test all of the following areas –
  - Scheduling software
  - Day-to-day operational procedures
  - Backup recovery strategy
  - Management and scheduling tools
  - Overnight processing
  - Query performance
- Note − The most important point is to test the scalability. Failure to do so will leave us a system design that does not work when the system grows.

5. Write short notes on Data Warehousing - Future Aspects.

Following are the future aspects of data warehousing.

- As we have seen that the size of the open database has grown approximately double its magnitude in the last few years, it shows the significant value that it contains.
- As the size of the databases grow, the estimates of what constitutes a very large database continues to grow.
- The hardware and software that are available today do not allow to keep a large amount of data online. For example, a Telco call record requires 10TB of data to be kept online, which is just a size of one month's record. If it requires to keep records of sales, marketing customer, employees, etc., then the size will be more than 100 TB.
- The record contains textual information and some multimedia data. Multimedia data cannot be easily manipulated as text data. Searching the multimedia data

is not an easy task, whereas textual information can be retrieved by the relational software available today.

- Apart from size planning, it is complex to build and run data warehouse systems that are ever increasing in size. As the number of users increases, the size of the data warehouse also increases. These users will also require to access the system.
- With the growth of the Internet, there is a requirement of users to access data online.

6. Explain in detail about Data Warehousing – Security

- The objective of a data warehouse is to make large amounts of data easily accessible to the users, hence allowing the users to extract information about the business as a whole.
- But we know that there could be some security restrictions applied on the data that can be an obstacle for accessing the information.
- If the analyst has a restricted view of data, then it is impossible to capture a complete picture of the trends within the business.
- The data from each analyst can be summarized and passed on to management where the different summaries can be aggregated.
- As the aggregations of summaries cannot be the same as that of the aggregation as a whole, it is possible to miss some information trends in the data unless someone is analyzing the data as a whole.

Security Requirements

- Adding security features affect the performance of the data warehouse, therefore it is important to determine the security requirements as early as possible. It is difficult to add security features after the data warehouse has gone live.
- During the design phase of the data warehouse, we should keep in mind what data sources may be added later and what would be the impact of adding those data sources.
- We should consider the following possibilities during the design phase.
    - Whether the new data sources will require new security and/or audit restrictions to be implemented?

  - Whether the new users added who have restricted access to data that is already generally available?
- This situation arises when the future users and the data sources are not well known. In such a situation, we need to use the knowledge of business and the objective of data warehouse to know likely requirements.
- The following activities get affected by security measures –
  - User access
  - Data load
  - Data movement
  - Query generation

User Access

- We need to first classify the data and then classify the users on the basis of the data they can access. In other words, the users are classified according to the data they can access.

Data Classification

The following two approaches can be used to classify the data −

- Data can be classified according to its sensitivity. Highly-sensitive data is classified as highly restricted and less-sensitive data is classified as less restrictive.
- Data can also be classified according to the job function. This restriction allows only specific users to view particular data. Here we restrict the users to view only that part of the data in which they are interested and are responsible for.
- There are some issues in the second approach. To understand, let's have an example.
- Suppose you are building the data warehouse for a bank. Consider that the data being stored in the data warehouse is the transaction data for all the accounts.
- The question here is, who is allowed to see the transaction data. The solution lies in classifying the data according to the function.

User classification

The following approaches can be used to classify the users –

- Users can be classified as per the hierarchy of users in an organization, i.e., users can be classified by departments, sections, groups, and so on.
- Users can also be classified according to their role, with people grouped across departments based on their role.

Classification on basis of Department

- Refer figure 5.4 for the Classification on basis of Department
- Let's have an example of a data warehouse where the users are from sales and marketing department.
- We can have security by top-to-down company view, with access centered on the different departments.
- But there could be some restrictions on users at different levels. This structure is shown in the following diagram.

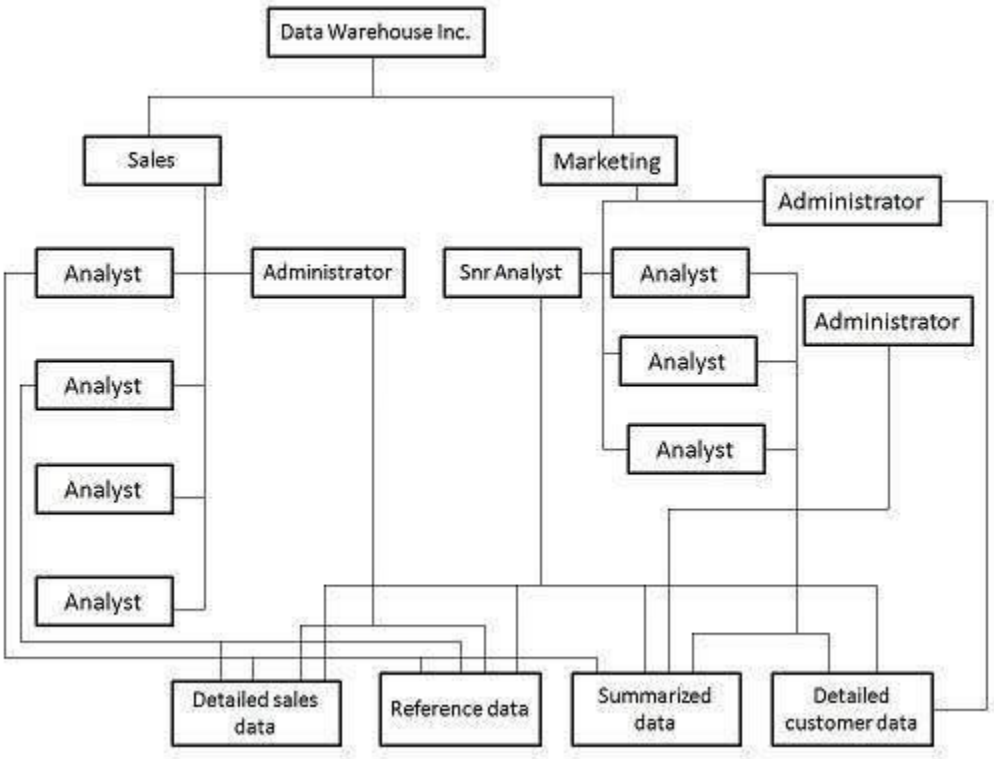


Fig.5.4 Classification on basis of department

- But if each department accesses different data, then we should design the security access for each department separately.

- This can be achieved by departmental data marts. Since these data marts are separated from the data warehouse, we can enforce separate security restrictions on each data mart. This approach is shown in the following figure 5.5.

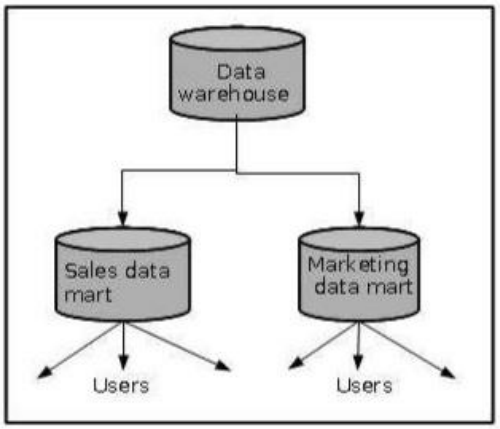


Fig.5.5 Classification

Classification Based on Role

- If the data is generally available to all the departments, then it is useful to follow the role access hierarchy.
- In other words, if the data is generally accessed by all the departments, then apply security restrictions as per the role of the user.
- The role access hierarchy is shown in the following figure 5.6.

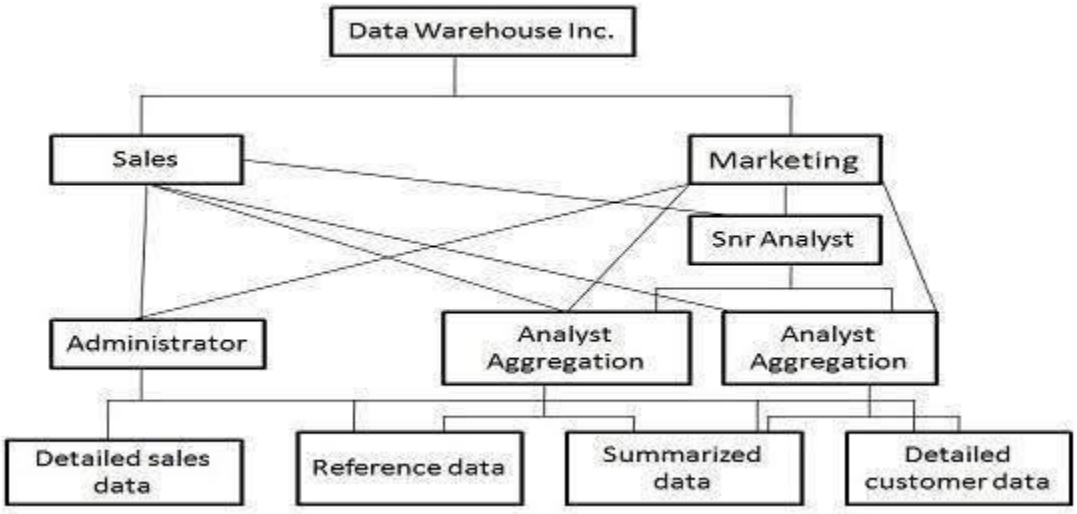


Fig.5.6 Classification based on Role

Audit Requirements

- Auditing is a subset of security, a costly activity. Auditing can cause heavy overheads on the system.
- To complete an audit in time, we require more hardware and therefore, it is recommended that wherever possible, auditing should be switched off.
- Audit requirements can be categorized as follows –
    - Connections
    - Disconnections
    - Data access
    - Data change
- Note − For each of the above-mentioned categories, it is necessary to audit success, failure, or both. From the perspective of security reasons, the auditing of failures are very important. Auditing of failure is important because they can highlight unauthorized or fraudulent access.

Network Requirements

- Network security is as important as other securities. We cannot ignore the network security requirement. We need to consider the following issues –
    - Is it necessary to encrypt data before transferring it to the data warehouse?
    - Are there restrictions on which network routes the data can take?
- These restrictions need to be considered carefully. Following are the points to remember –
    - The process of encryption and decryption will increase overheads. It would require more processing power and processing time.
    - The cost of encryption can be high if the system is already a loaded system because the encryption is borne by the source system.

Data Movement

- There exist potential security implications while moving the data. Suppose we need to transfer some restricted data as a flat file to be loaded.
- When the data is loaded into the data warehouse, the following questions are raised –
    - Where is the flat file stored?

  - Who has access to that disk space?
- If we talk about the backup of these flat files, the following questions are raised –
  - Do you backup encrypted or decrypted versions?
  - Do these backups need to be made to special tapes that are stored separately?
  - Who has access to these tapes?
- Some other forms of data movement like query result sets also need to be considered. The questions raised while creating the temporary table are as follows –
  - Where is that temporary table to be held?
  - How do you make such table visible?
- We should avoid the accidental flouting of security restrictions. If a user with access to the restricted data can generate accessible temporary tables, data can be visible to non-authorized users. We can overcome this problem by having a separate temporary area for users with access to restricted data.

Documentation

The audit and security requirements need to be properly documented. This will be treated as a part of justification. This document can contain all the information gathered from –

- Data classification
- User classification
- Network requirements
- Data movement and storage requirements
- All auditable actions

Impact of Security on Design

- Security affects the application code and the development timescales. Security affects the following area –
  - Application development
  - Database design
  - Testing

- Application Development
  - Security affects the overall application development and it also affects the design of the important components of the data warehouse such as load manager, warehouse manager, and query manager.
  - The load manager may require checking code to filter record and place them in different locations. More transformation rules may also be required to hide certain data. Also there may be requirements of extra metadata to handle any extra objects.
  - To create and maintain extra views, the warehouse manager may require extra codes to enforce security. Extra checks may have to be coded into the data warehouse to prevent it from being fooled into moving data into a location where it should not be available.
  - The query manager requires the changes to handle any access restrictions. The query manager will need to be aware of all extra views and aggregations.

Database design

- The database layout is also affected because when security measures are implemented, there is an increase in the number of views and tables.
- Adding security increases the size of the database and hence increases the complexity of the database design and management.
- It will also add complexity to the backup management and recovery plan.

Testing

- Testing the data warehouse is a complex and lengthy process. Adding security to the data warehouse also affects the testing time complexity.
- It affects the testing in the following two ways –
  - It will increase the time required for integration and system testing.
  - There is added functionality to be tested which will increase the size of the testing suite.

7. Discuss the various access types to the data stored in a data warehouse and consider the data types of online shopping. [NOV/DEC 2023]

Accessing data in a data warehouse involves various types of access methods depending on the needs of the users and the structure of the data. Let's discuss the access types and consider how they relate to the data types typically found in online shopping scenarios.

Access Types to Data in a Data Warehouse

1. **SQL Queries**: Structured Query Language (SQL) is commonly used to retrieve data from data warehouses. Users write SQL queries to specify the data they need, using SELECT statements to fetch specific columns or aggregating functions for summaries.
2. **OLAP (Online Analytical Processing)**: OLAP tools allow users to analyze multidimensional data interactively. OLAP operations include slicing (viewing data from one viewpoint), dicing (viewing data from multiple viewpoints), drilling down (navigating from summary to detailed data), and rolling up (aggregating data).
3. **Data Mining**: Advanced users and analysts use data mining techniques to discover patterns and trends in large datasets. This involves applying statistical algorithms, machine learning models, or other analytical techniques to extract actionable insights.
4. **Reporting Tools**: Business intelligence (BI) and reporting tools provide formatted reports and dashboards based on predefined queries or visualizations. These tools often offer a user-friendly interface for non-technical users to explore data.
5. **ETL (Extract, Transform, Load)**: Data engineers and administrators use ETL processes to extract data from various sources, transform it into a unified format, and load it into the data warehouse. This ensures data consistency and readiness for analysis.

Data Types in Online Shopping

In the context of online shopping, the data stored and analyzed in a data warehouse typically includes:

1. **Transactional Data**: Records of individual transactions such as purchases, returns, and payments. This data includes details like customer IDs, product IDs, quantities, prices, timestamps, and payment methods.
2. **Customer Data**: Information about customers such as demographics, preferences, browsing history, and purchase patterns. This data helps in understanding customer behavior and targeting marketing efforts.
3. **Product Data**: Details about the products sold, including categories, descriptions, prices, inventory levels, and attributes. Product data is crucial for analyzing sales trends and optimizing inventory management.
4. **Marketing Data**: Data related to marketing campaigns, promotions, discounts, and customer responses. This includes data from email campaigns, social media interactions, and ad impressions.

Access Types in Relation to Online Shopping Data

- **SQL Queries**: Retail analysts can use SQL to query transactional data to analyze sales performance, customer behavior, and inventory turnover.
- **OLAP Tools**: Managers can use OLAP tools to slice and dice sales data by different dimensions such as product categories, customer segments, or geographic regions to identify trends and patterns.
- **Data Mining**: Data scientists can apply data mining techniques to predict customer churn, recommend products based on purchase history, or detect fraud patterns in transactions.
- **Reporting Tools**: Marketing teams can use BI tools to create reports and dashboards that visualize key performance indicators (KPIs) such as sales revenue, conversion rates, and customer acquisition costs.
- **ETL Processes**: Data engineers ensure that all relevant data from online shopping platforms, CRM systems, and marketing channels is integrated into the data warehouse through efficient ETL processes.

- In conclusion, accessing data in a data warehouse involves using SQL queries, OLAP tools, data mining techniques, reporting tools, and ETL processes.
- Each access type serves a specific purpose in analyzing diverse data types such as transactional, customer, product, and marketing data in online shopping scenarios.
- These tools and methods collectively enable businesses to derive valuable insights for strategic decision-making and operational optimization.

8. Explain how to use a familiar approach help reduce training costs associated with building a query and reporting environment with suitable example.

[NOV/DEC 2023]

- Reducing training costs associated with building a query and reporting environment can be effectively achieved by leveraging a familiar approach that simplifies the learning curve for users.
- Let's explore how adopting a self-service BI (Business Intelligence) approach can help achieve this goal, along with a suitable example.

Using Self-Service BI to Reduce Training Costs

- **Self-service BI** empowers users to generate reports and queries without extensive technical expertise.
- This approach typically involves intuitive tools and interfaces that allow users to access and analyze data independently, reducing the need for specialized training and support.

Example Scenario:

- Imagine a retail company, "E-Shop Express," which wants to implement a self-service BI environment to reduce training costs associated with querying and reporting on their sales data.

Steps to Implement Self-Service BI:

1. **Choose an Intuitive BI Tool**: Select a user-friendly BI tool that offers drag-and-drop interfaces, pre-built templates, and natural language querying capabilities. Examples include Microsoft Power BI, Tableau, or Google Data Studio.
2. **Data Modeling and Integration**: Ensure that the BI tool can integrate with E-Shop Express's data sources, such as transactional databases and customer data platforms. Create standardized data models or data sets that simplify querying for common business questions.
3. **Training on BI Tool Basics**: Conduct introductory training sessions focused on navigating the BI tool, connecting to data sources, and using basic functionalities like filtering, sorting, and creating simple visualizations. This training should be accessible to all employees, regardless of technical background.
4. **Template Creation**: Develop standardized report templates and dashboards that address typical business queries, such as sales trends by product category, customer demographics, or geographic regions. These templates serve as starting points for users to explore data without starting from scratch.
5. **Encourage Exploration**: Promote a culture of exploration and experimentation with data. Encourage users to use self-service BI tools to ask and answer their own business questions rather than relying on IT or data analysts for every query.

Benefits of Using Self-Service BI:

- **Reduced Dependency on IT**: Users can independently access and analyze data, reducing the burden on IT departments for ad-hoc queries and reports.
- **Faster Insights**: Self-service BI allows users to quickly generate insights, enabling faster decision-making across the organization.
- **Cost Efficiency**: By minimizing the need for extensive training and specialized skills, self-service BI lowers overall training costs associated with building and maintaining a query and reporting environment.

Conclusion:

- Implementing a self-service BI approach like the one outlined for E-Shop Express can significantly reduce training costs while empowering users to leverage data for better decision-making.
- By selecting intuitive BI tools, providing basic training, creating standardized templates, and promoting user exploration, organizations can foster a data-driven culture without incurring substantial training expenses traditionally associated with complex query and reporting environments.

9. Design a Multidimensional Cube with an Example. [Nov 2024]

Multidimensional Cube Design for Retail Sales Data Mart

- A multidimensional cube (OLAP cube) is used for efficient querying and analysis of data across multiple dimensions. It enables slicing, dicing, drilling down, and rolling up data for insightful decision-making.

Cube Dimensions and Measures

- We design a cube that focuses on sales performance, customer behavior, and inventory management.

Dimensions:

Time Dimension (Year, Quarter, Month, Week, Day)

Product Dimension (Category, Subcategory, Brand, Product Name)

Store Dimension (Region, City, Store Type)

Customer Dimension (Age Group, Gender, Income Level, Membership Tier)

Sales Channel Dimension (Online, Physical Store)

Measures (Aggregated Data):

Total Sales Revenue (SUM of revenue)

Total Quantity Sold (SUM of quantity)

Total Profit Margin (SUM of revenue - cost)

Inventory Levels (Stock at a given time)

Example Multidimensional Cube

Consider the following example:

Time	Product Category	Region	Sales Channel	Total Sales ($)	Total Quantity Sold
Q1 2024	Electronics	North	Online	1,200,000	3,500
Q1 2024	Clothing	North	Physical Store	750,000	8,200
Q1 2024	Electronics	South	Online	950,000	2,900
Q1 2024	Clothing	South	Physical Store	500,000	6,500

1. Time Dimension (Dim_Date)

Column Name	Data Type	Description
Date_ID	INT (PK)	Unique Date Identifier
Date	DATE	Calendar Date
Month	VARCHAR	Month Name
Quarter	VARCHAR	Quarter (Q1, Q2, Q3, Q4)
Year	INT	Year
Weekday	VARCHAR	Day Name (Monday, Tuesday, etc.)

2. Product Dimension (Dim_Product)

Column Name	Data Type	Description
Product_ID	INT (PK)	Unique Product Identifier
Product_Name	VARCHAR	Name of the Product
Category	VARCHAR	Product Category (Electronics, Clothing, etc.)
Brand	VARCHAR	Product Brand

Column Name	Data Type	Description
Price	DECIMAL	Selling Price

3. Store Dimension (Dim_Store)

Column Name	Data Type	Description
Store_ID	INT (PK)	Unique Store Identifier
Store_Name	VARCHAR	Name of the Store
Region	VARCHAR	Geographical Region (North, South, etc.)
City	VARCHAR	City of Operation
Store_Type	VARCHAR	'Physical Store' or 'Online'

4. Customer Dimension (Dim_Customer)

Column Name	Data Type	Description
Customer_ID	INT (PK)	Unique Customer Identifier
Age_Group	VARCHAR	Age Bracket (18-25, 26-40, etc.)
Gender	VARCHAR	Male/Female/Other
Income_Level	VARCHAR	Low, Medium, High
Membership_Tier	VARCHAR	Loyalty Tier (Bronze, Silver, Gold)

5. Sales Channel Dimension (Dim_SalesChannel)

Column Name	Data Type	Description

Time Product Category Region Sales Channel Total Sales ($) Total Quantity Sold

Q1 2024 Electronics North Online 1,200,000 3,500

Q1 2024 Clothing North Physical Store 750,000 8,200

Q1 2024 Electronics South Online 950,000 2,900

Q1 2024 Clothing South Physical Store 500,000 6,500

OLAP Operations on the Cube

1. Slice

- Select a single dimension value while keeping others constant.
- Example: Analyze only "Electronics" category across all regions and sales channels.

2. Dice

- Select specific values for multiple dimensions.
- Example: View sales for Q1 2024 and Electronics in the North and South regions.

3. Drill Down

- Increase the level of detail.
- Example: From Q1 2024, drill down into January, February, and March sales.

4. Roll Up

- Aggregate data to a higher level.
- Example: Roll up from monthly sales to quarterly or yearly sales.

5. Pivot (Rotate Cube)

- Rearrange dimensions for different perspectives.
- Example: Swap Region with Sales Channel to analyze performance by channel instead of geography.

Benefits of the Cube Design

- Fast retrieval of sales and inventory data
- Supports trend analysis and forecasting
- Helps in decision-making for pricing, promotions, and stock management

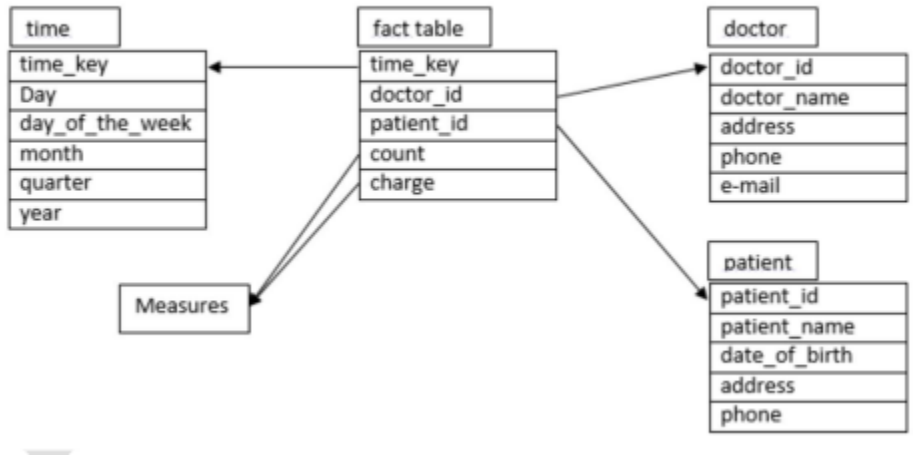
This OLAP cube structure will enable deeper insights into sales performance, customer trends, and inventory management.

10. Suppose that a data warehouse consists of the four dimensions date, spectator, location and game, and the two measures count and charge, where charge is the fare that a spectator pays when watching a game on a given date spectators may be students, adults, or seniors, with each category having its own charge rate.

(i) Draw a star schema diagram for the data warehouse. (4)

(ii) Starting with a base cuboid [date, spectator, location, game], what specific OLAP operations should one perform in order to list the total charge paid by student spectators at GM_Place in 2000? (6) [Nov 2024]

a. Ans:



b. Ans:

First, we should use roll-up operation to get the year 2004(rolling-up from day then month to year). After getting that, we need to use slice operation to select (2004). Second, we should use roll-up operation again to get all patients. Then, we need to use slice operation to select (all). Finally, we get list the total fee collected by each doctor in 2004.

So,

1. roll up from day to month to year

2. slice for year = “2004”

3. roll up on patient from individual patient to all

4. slice for patient = “all”

4. get the list of total fee collected by each doctor in 2004

c. Ans

Select doctor, Sum(charge) From fee Where year = 2004 Group by doctor

11. Consider you are working for a retail chain that operates in multiple regions and sells a variety of products both in physical stores and online. The company wants to improve its decision making process by analyzing sales data more effectively. They have specific business requirements to be met through the design of a data mart.

Requirements:

(i) Analyze sales performance across different regions, product categories and sales channels (physical stores vs. online).

(ii) Identify trends in customer purchasing behavior, including popular products, seasonal trends and customer demographics.

(iii) Track inventory levels and monitor stock movement to optimize inventory management and prevent stock outs.

(iv) Integrate data from various sources, including sales transactions, inventory databases and customer demographics, to provide a comprehensive view of the business.

Design a data mart to meet the above mentioned requirements. [Nov 2024]

1. Data Mart Schema - Star Schema

The data mart will follow a star schema structure, consisting of a central Fact Table surrounded by multiple Dimension Tables.

Fact Table: Fact_Sales

This table stores transactional sales data, enabling sales analysis by different dimensions.

This table stores transactional sales data, enabling sales analysis by different dimensions.

Column Name	Data Type	Description
Sales_ID	INT (PK)	Unique identifier for each sales transaction
Date_ID	INT (FK)	Foreign key referencing Dim_Date
Product_ID	INT (FK)	Foreign key referencing Dim_Product
Store_ID	INT (FK)	Foreign key referencing Dim_Store
Customer_ID	INT (FK)	Foreign key referencing Dim_Customer
Sales_Channel	VARCHAR	'Online' or 'Physical Store'
Quantity_Sold	INT	Number of products sold
Revenue	DECIMAL	Total revenue from the sale
Discount_Amount	DECIMAL	Discount applied to the sale
Cost_Price	DECIMAL	Cost of the product
Profit_Margin	DECIMAL	Profit after subtracting cost from revenue

Dimension Tables:

Each dimension table provides additional context for analyzing sales.

1. Dim_Date

Stores date-related information for time-based analysis.

Column Name	Data Type	Description
Date_ID	INT (PK)	Unique ID for each date
Date	DATE	Actual date
Day	INT	Day of the month
Month	INT	Month of the year
Year	INT	Year
Quarter	INT	Quarter (1,2,3,4)
Weekday	VARCHAR	Day name (Monday, Tuesday, etc.)
Holiday_Flag	BOOLEAN	Indicates whether the date is a holiday

2. Dim_Product

Stores product details to analyze sales by category and brand.

Column Name	Data Type	Description
Product_ID	INT (PK)	Unique product identifier
Product_Name	VARCHAR	Name of the product
Category	VARCHAR	Product category (e.g., Electronics, Clothing)
Subcategory	VARCHAR	More specific classification
Brand	VARCHAR	Brand of the product
Unit_Cost	DECIMAL	Cost price of the product
Unit_Price	DECIMAL	Selling price of the product

3. Dim_Store

Stores store details to analyze performance by region.

Column Name	Data Type	Description
Store_ID	INT (PK)	Unique store identifier
Store_Name	VARCHAR	Name of the store
Region	VARCHAR	Geographical region (e.g., East, West)
City	VARCHAR	City where the store is located
Country	VARCHAR	Country of operation
Store_Type	VARCHAR	'Physical Store' or 'Online'

4. Dim_Customer

Stores customer demographics for behavioral analysis.

Column Name	Data Type	Description
Customer_ID	INT (PK)	Unique customer identifier
First_Name	VARCHAR	Customer first name
Last_Name	VARCHAR	Customer last name
Age	INT	Age of the customer
Gender	VARCHAR	Male/Female/Other

Column Name	Data Type	Description
Income_Level	VARCHAR	Income category (Low, Medium, High)
Membership_Tier	VARCHAR	Loyalty program tier (e.g., Bronze, Silver, Gold)

5. Fact_Inventory

Tracks stock movement to prevent stockouts.

Column Name	Data Type	Description
Inventory_ID	INT (PK)	Unique inventory tracking ID
Product_ID	INT (FK)	References Dim_Product
Store_ID	INT (FK)	References Dim_Store
Date_ID	INT (FK)	References Dim_Date
Stock_Beginning	INT	Stock level at the start of the period
Stock_Received	INT	New stock received
Stock_Sold	INT	Stock sold
Stock_Ending	INT	Stock level at the end of the period

---

2. ETL (Extract, Transform, Load) Process

To integrate data from different sources:

1. **Extract** data from:
    - Sales transactions database
    - Inventory system
    - Customer demographics database
2. **Transform**:
    - Clean and format data (e.g., standardize dates, remove duplicates)
    - Aggregate sales by region, product, and store
    - Compute profit margins and seasonal trends
3. **Load** data into the **Retail Sales Data Mart** for analysis.

3. Business Insights & Reports

With this data mart, the company can generate reports like:

1. **Sales Performance Dashboard**:
    - Revenue by product category, region, and sales channel
    - Best-selling and worst-selling products
2. **Customer Insights**:
    - Purchasing behavior by demographics
    - Customer segmentation (age, income level, region)
3. **Inventory Management Reports**:
    - Stock levels across stores
    - Fast-moving vs. slow-moving products
    - Seasonal inventory trends
4. **Trend Analysis**:
    - Monthly and yearly sales trends
    - Impact of promotions and discounts on sales

This data mart enables **data-driven decision-making**, improving **sales, inventory management, and customer engagement** across regions and sales channels.

Reg. No. : | E | N | G | G | T | R | E | E | . | C | O | M |

Question Paper Code : 20404

B.E./B.Tech. DEGREE EXAMINATIONS, NOVEMBER/DECEMBER 2023.

Fifth Semester

Computer Science and Design

CCS 341 — DATA WAREHOUSING

(Common to : Computer Science and Engineering/Computer Science and Engineering (Artificial Intelligence and Machine Learning/Computer Science and Engineering (Cyber Security)/Computer and Communication Engineering/Artificial Intelligence and Data Science/Computer Science and Business Systems and Information Technology)

(Regulations 2021)

Time : Three hours | Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. What is data warehouse? List out the benefits of Data warehousing.
2. How is data warehouse different from a database? Identify the similarity.
3. Compare OLTP and OLAP Systems.
4. List out the views in the design of a data warehouse.
5. Differentiate metadata and data mart.
6. Propose the features of Metadata repository in data warehousing.
7. Define star schema.
8. What is snowflake schema?
9. Define Query Manager.
10. What are the various sources for data warehouse?

PART B — (5 × 13 = 65 marks)

11. (a) Explain with three-tier architecture diagram and give the steps for design and construction of Data warehouses.

Or

(b) Describe the technologies used to improve the performance in data warehouse environment. Mention a few alternate technologies also.

12. (a) Explain the different types of OLAP tools. What type of OLAP tool is best suited for datasets that require both detailed and summarized analysis?

Or

(b) Diagrammatically illustrate and describe the architecture of MOLAP and ROLAP. Find the major difference between MOLAP and ROLAP.

13. (a) Illustrate the various classification of Meta data with suitable examples and explain the same.

Or

(b) Elaborate in detail about the various issues to be considered when designing and implementing a data-warehousing environment.

14. (a) Describe in brief about various schemas in multidimensional data model.

Or

(b) Discuss the various types of database parallelism with suitable examples.

15. (a) Discuss the various access types to the data stored in a data warehouse and consider the data types of online shopping.

Or

(b) Explain how to use a familiar approach help reduce training costs associated with building a query and reporting environment with suitable example.

PART C — (1 × 15 = 15 marks)

16. (a) Discuss in detail about the case study of Data Warehouse storage and accessibility for the Government of Tamilnadu for any two schemes in practice.

Or

(b) Assume you are working as a data analyst for a retail company. The company has a data warehouse that stores information about customers, products, salespersons, and sales time. The data warehouse has three measures: sales amount (in rupees) VAT (in rupees), and payment type (in rupees). You have been ask to analyze the sales data to identify the top-selling products in each region. Which schema would you use to model the data warehouse? Explain your reasoning.

———————

Reg. No. : E N G G T R E E . C O M

Question Paper Code : 50424

B.E./B.Tech. DEGREE EXAMINATIONS, APRIL/MAY 2024.

Fifth/Sixth Semester

Computer Science and Engineering

CCS 341– DATA WAREHOUSING

(Common to : Computer Science and Design/Computer Science and Engineering (Artificial Intelligence and Machine Learning)/Computer Science and Engineering (Cyber Security)/Computer and Communication Engineering/Artificial Intelligence and Data Science/Computer Science and Business Systems/Information Technology)

(Regulations 2021)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Define a data warehouse. What are its key components?
2. Compare and contrast an operational database with a data warehouse.
3. What does ETL stand for, and how does it differ from ELT?
4. List the differences between OLAP and OLTP.
5. How to choose data partitioning strategy for Data warehouse?
6. What is data mart?
7. Compare and Contrast star schema and a snowflake schema.
8. What is Data Cube?
9. Name two types of Data Warehousing System Managers and their roles.
10. What is the function of the Load Manager in data warehousing?

PART B — (5 × 13 = 65 marks)

11. (a) Explain the three-tier data warehouse architecture and its significance.

Or

(b) Compare and contrast Autonomous Data Warehouses with Snowflake data warehousing solutions.

12. (a) Explain the characteristics of OLAP and its operations.

Or

(b) Discuss the ETL process in data warehousing.

13. (a) Describe the process of designing a cost-effective data mart.

Or

(b) Explain the role of meta data in a data warehouse. Discuss the challenges associated with metadata management.

14. (a) Detail the process of dimensional modeling and its importance in data warehousing. Discuss the advantages of using a multi-dimensional data model.

Or

(b) Write notes on the following

(i) Database parallelism and (7)

(ii) Data warehouse Tools. (6)

15. (a) Explain the concept of data warehouse tuning and testing.

Or

(b) Discuss the roles of system and process managers in a data warehousing environment.

PART C — (1 × 15 = 15 marks)

16. (a) Outline a key strategy for ensuring a data warehousing solution remains scalable as a tech startup grows. Highlight one specific technology or approach that could be utilized.

Or

(b) Describe how modernizing the data warehouse architecture could improve data analytics and decision-making for a global retail company. Focus on one major limitation of the old system and how a modern solution addresses it.

Reg. No. : 4 2 1 6 2 2 2 0 5 0 7 8

Question Paper Code : 40442

B.E./B.Tech. DEGREE EXAMINATIONS, NOVEMBER/DECEMBER 2024.

Fifth/Sixth Semester

Computer Science and Engineering

CCS 341 – DATA WAREHOUSING

(Common to : Computer Science and Design/ Computer Science and Engineering (Artificial Intelligence and Machine Learning)/ Computer Science and Engineering (Cyber Security)/ Computer and Communication Engineering/ Artificial Intelligence and Data Science/ Computer Science and Business Systems/ Information Technology)

(Regulations 2021)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Write the different steps in knowledge discovery in databases.
2. How is data warehouse different from a database? Identify the similarity.
3. What is ETL?
4. Outline the characteristics of OLAP.
5. Compare DataMart with Data Warehouse.
6. Why is a data mart considered cost-effective compared to a data warehouse?
7. Name the types of data warehouse schema.
8. What is the significance of a fact constellation schema in dimensional modelling?
9. Recall the responsibilities of a Data Warehousing System Configuration Manager.
10. Outline the primary task of a Data Warehousing System Backup Recovery Manager.

15. (a) Explain about various data warehousing process managers, its collaboration and interaction to ensure operations and performance of the entire data warehousing system. (13)

Or

(b) Justify the statement "Testing process contribute to the overall quality a data warehousing system", and describe the types of tests conducted on the system. (13)

PART C — (1 × 15 = 15 marks)

16. (a) (i) Design a multidimensional cube with an example. (5)

(ii) Suppose that a data warehouse consists of the four dimensions date, spectator, location and game, and the two measures count and charge, where charge is the fare that a spectator pays when watching a game on a given date spectators may be students, adults, or seniors, with each category having its own charge rate.

(1) Draw a star schema diagram for the data warehouse. (4)

(2) Starting with a base cuboid [date, spectator, location, game], what specific OLAP operations should one perform in order to list the total charge paid by student spectators at GM_Place in 2000? (6)

Or

(b) Consider you are working for a retail chain that operates in multiple regions and sells a variety of products both in physical stores and online. The company wants to improve its decision-making process by analyzing sales data more effectively. They have specific business requirements to be met through the design of a data mart.

Requirements:

(i) Analyze sales performance across different regions, product categories and sales channels (physical stores vs. online).

(ii) Identify trends in customer purchasing behavior, including popular products, seasonal trends and customer demographics.

(iii) Track inventory levels and monitor stock movement to optimize inventory management and prevent stockouts.

(iv) Integrate data from various sources, including sales transactions, inventory databases and customer demographics, to provide a comprehensive view of the business.

Design a data mart to meet the above mentioned requirements.

PART B — (5 × 13 = 65 marks)

11. (a) (i) With a neat sketch, explain the Steps for design and construction of Data warehouses and explain with three tier architecture. (6)

(ii) Describe the three layers of Snowflake's architecture. (7)

Or

(b) (i) Suppose that a data warehouse consists of four dimensions customer, product, salesperson, and sales time and the three measure sales Amount (in rupees), VAT (in rupees) and payment type (in rupees). Draw the different classes of schemas that are popularly used for modeling data warehouses and explain it. (8)

(ii) Discuss on autonomous data warehouse. (5)

12. (a) (i) Compare OLAP with OLTP system. (6)

(ii) Discuss the typical OLAP operations with an example (KDD). (7)

Or

(b) (i) Diagrammatically illustrate and describe the architecture of MOLAP, ROLAP, HOLAP. (7)

(ii) Identify the Major differences between MOLAP and ROLAP. (6)

13. (a) Explain the following

(i) Metadata repository. (7)

(ii) Role of Metadata. (6)

Or

(b) (i) Compare and contrast the advantages and disadvantages of vertical partitioning and horizontal partitioning in a data warehousing context. (7)

(ii) Describe the challenges of metadata management. (6)

14. (a) (i) Propose a fact constellation schema for a healthcare data warehouse to support complex analytical queries. (7)

(ii) Describe the process architecture involved in designing and implementing a star schema. (6)

Or

(b) (i) Explain how does a snowflake schema differ from a star schema in terms of normalization? (7)

(ii) Explain the relationship between the Load Manager and the Warehouse Manager in a data warehousing process. (6)