UNIT-III

Building a data warehouse – Introduction

- **Gather Requirements:** Aligning the business goals such as the data types to be stored, the frequency of data refresh, the level of data granularity and needs of different departments with the overall data warehouse project.
- **Set Up Environments:** This step is about creating three environments for data warehouse development, testing, and production, each running on separate servers. The most popular database platforms for data warehousing are Oracle, Microsoft SQL Server, and IBM DB2.
- **Data Modeling:** Design the data warehouse schema, including the fact tables and dimension tables, to support the business requirements. The schema design is the blueprint of your data warehouse. It outlines the relationships between the various data entities and attributes that will be stored in the data warehouse.
- **Develop Your ETL Process:** ETL stands for Extract, Transform, and Load. This process is how data gets moved from its source into your warehouse. Popular ETL tools include Microsoft SQL Server Integration Services, Oracle Data Integrator, and Talend.
- **OLAP Cube Design:** Design OLAP cubes to support analysis and reporting requirements. An OLAP cube helps you to analyze the data in your data warehouse or data mart. Since your warehouse will be sorting data from multiple sources, the OLAP cube helps you to organize all that data in a multi-dimensional format that makes it easier to analyze.
- **Reporting & Analysis:** Developing and deploying the reporting and analytics tools that will be used to extract insights and knowledge from the data warehouse.
- **Optimize Queries:** Optimizing queries ensures that the system can handle large amounts of data and respond quickly to queries.
- **Establish a Rollout Plan:** Determine how the data warehouse will be introduced to the organization, which groups or individuals will have access to it, and how the data will be presented to these users.

Critical success factor

† Do not launch the data warehouse unless and until your company is ready for it.

† Find the best executive sponsor. Ensure continued, long-term, and committed support.

† Emphasize the business aspects, not the technological ones, of the project. Choose a business-oriented project manager.

† Take an enterprise-wide view for the requirements.

† Have a pragmatic, staged implementation plan.

† Communicate realistic expectations to the users; deliver on the promises.

† Do not overreach to cost-justify and predict ROI.

† Institute appropriate and effective communication methods.

† Throughout the project life cycle, keep the project as a joint effort between IT and users.

† Adopt proven technologies; avoid bleeding-edge technologies.

† Recognize the paramount importance of data quality.

† Do not ignore the potential of data from external sources.

† Do not underestimate the time and effort for the data extraction, transformation, and loading (ETL) functions.

† Select the architecture that is just right for your environment; data warehousing is not a one-size-fits-all proposition.

† Architecture first, technology next, and only then, tools.

† Determine a clear training strategy.

† Be wary of “analysis paralysis.”

† Begin deployment with a suitable and visible pilot to deliver early benefits.

† Do not neglect the importance of scalability. Plan for growth and evolution.

† Focus on queries, not transactions. The data warehouse is query-centric, not transaction-oriented.

† Emphasize and distinguish between the datawarehousing and analytical environments.

† Clearly define and manage data ownership considerations.

The term "critical success factor" to refer to those things that must go right if an undertaking is to become successful. For a data warehouse project, these critical success factors include

- **Set specific, achievable, and measurable goals**
  A data warehouse project is a very large project and often tends to suffer from "scope creep" in spite of putting in large sums of money. >cope creep generally happens when the users define new requirements as the project progresses, thereby adding to the initial set of requirements. Although every individual request for increasing scope may be valid, the combination of these requests result in late and over budget projects. Thus, tor a big project

of building a data warehouse, the expectations should be managed and the goals should be kept specific and achievable.

- **Involve everyone throughout the project**
  The stakeholders in a data warehouse project are the internal people of the organization. This group comprises of the business analysts, managers, executives, sponsors, and other IT users. All of these people are important for the success of the project and so must be a part of the project right from the beginning They should be kept abreast of the developments as the project proceeds, the project team must keep all these people in the loop well-informed and involved and must take utmost care to keep everyone contented, as in certain Situations, there may be collision in thinking among these people.

- **Keep an eye on the big picture**
  The primary motive of a data warehouse is to provide answers to business problems. So, utmost care should be taken to avoid moving on with other, more local agendas.

- **Pay attention to the details and do not depend on assumptions**
  A data warehouse has an enormous amount of data that is being used by a number of sers, so attention to details must be given. However, these details can be affected by the assumptions made in creating the warehouse. Here are some of the pitfalls that can be encountered:
  - ✓ Assuming the sources of data are clean, consistent, and have an acceptable quality.
  - ✓ Assuming that the summary data that is present in the organization is adequate.
  - ✓ Assuming that the detailed data will not be needed at a later point of time. Assuming that development of the project is on track.
  - ✓ If users have all the skills needed and are capable enough to use the warehouse and its tools.
  - ✓ Assuming that the data warehouse experts will be available on short notice to solve last minute problems.
  - ✓ If the IT department has all the skills that are required to manage the project.

- **Consider long-term strategy**
  A long-term strategy for the data warehouse should be developed which must include a lot of detail but should also be flexible enough so that the warehouse can accommodate changes have that may to be done in due course of time. Tactical decisions like those often given below solve immediate problems but create long-term difficulties.
  - ✓ Building individual data marts without focusing on the warehouse plan. long-term
  - ✓ Establishing individual data marts without considering enterprise-Wide data definitions and standards.

- ✓ Not assigning roles to people who should be responsible for maintaining data quality throughout the organization.
- ✓ Using a platform that does not scale up as the business grows need for and e data increases.
- ✓ Not establishing metadata standards that will be used.

- **Learn from others**
  Data warehousing is a relatively new concept and firms beginning jus to create a data warehouse must struggle with the learning curve The best approach is to learn from those firms that have already built a warehouse and are using it. Links could be made at many data warehousing, conferare conducted every year to bring together vendors, academics, and important people who have practical experience about the subject.

Requirement Analysis

Requirement analysis is the most crucial factor for the success of any project. In he the absence of a clear goal, success rates are low. The steps in the require analysis phase that have been found to work can be listed as below:

- Clearly state the problems that have to be solved.
- Identify all data sources and the formats in which the data is stored in them.
- Identify the users of the data warehouse system.
- Clearly specify the budget in terms of time, money, and personnel.
- Ask the users to specify their expectations from the new system.
- Ask the management to specify the success criteria.
- Filter requirements from their desires. Initially start with designing the system as per the requirements, and then later on in the enhancement phase, address the desires.
- Formulate a prioritized requirements document, listing the requirement, its source, the success criteria, and its priority.
- Get a sign-off of the requirements, resource allocation, and schedule from the top management before the team can proceed with later stages

To begin with requirement analysis, the first and foremost thing that must be done is to identify the sponsors of the data warehouse project and ensure that these sponsors understand and support the project.

The second thing that needs to be done is to thoroughly understand the business before starting any discussions with the users. Then interview the users and work with them to learn their needs and then, turn these requirements into project requirements.

Identify the information that is needed to make the data warehouse successful. For this, do not go by users and do not try to put what data the users think should be in the data warehouse; rather

provide the information that is necessary for the business. And to provide this information in the data Warehouse, the project team must learn about user's objectives and challenges and how they go about making business decisions. Business users and the team should be closely tied to each other during the logical design process, as the users can tell the meaning of existing data much better.

After understanding the business needs, the team must interview the database administrator of the operational systems as well to know what data exists and where it resides. The team should make sure that all the intended 1sers are thoroughly interviewed and that the users should participate equally in the progress of the requirements definition.

Planning for the data warehouse

A data warehouse is planned in terms of business requirements, personal finances, and feasibility.

- **Project Staff**
  - ✓ Technical staff that includes the project leader, a data analyst, a business analyst, a database administrator, and programmers who are familiar with business problems to be solved.
  - ✓ An ad hoc technical staff who will be called to join the project as and when needed for specific project tasks like for technical support technical writing, training, and helpdesk.
  - ✓ An end-user staff that comprises subject matter experts.
  - ✓ Corporate level sponsors such as executives from the end-user and IT community.

- **Project Plan**

  To be successful, a big project like that of a data warehouse calls for good and careful planning.

  - ✓ An overall plan for creating the data warehouse and its infrastructure
  - ✓ Detailed plans for every individual application that would be run in the data warehouse environment.

- **Overall planning :**The overall plan for creating a data warehouse includes two broad aspects:
- **Vision :**Vision of the project states what must be built. Different people in the organization may have different objectives: the sponsor may want an improved EIS application; the business analyst may want a better source for strategic analysis; and the marketing managers may want to work with data mining applications. All the objectives of the various groups involved cannot be satisfied simultaneously, so the conflicting viewpoints of different also must individual be resolved. he output of the vision phase is

a document that describes why warehouse then is being built, what should be included in the data warehouse what should not.

- **Validation and estimation**

Along with defining the vision of the data ware project, the anticipated costs, schedule, and resources that will be required are estimated during the planning phase. The technical feasibility which is also a validation step is completed. Validation involves cross-checking the requirement and identifying the risks associated with the development. Estimation and technical feasibility are interlinked with each other. Many trade-offs will have to be made be made to complete the data warehouse project within reasonable timeframe and a reasonable budget.

- **Detailed Planning**

Detailed planning moves the project from a conceptual entity to a specific one. The main aim here is to define the budget, schedule, and intermediate and final deliverables for the data warehouse project. Project planning tools are used to allow managers to Visualize the time sequence in which the events must occur, the kind of personnel that will have to be assigned, and the hardware and software components that will have to be acquired and integrated.

A well-formulated and a structured plan includes details on every step of the project, from the source of data to how the data is to be cleaned, stored, and used by the end-users, to the end-user training program. The training part of the plan considers teaching end-users the mechanics of how to obtain information from the warehouse, and how to go on with their need to extract strategic information from the data warehouse.

- **Infrastructure planning**

The infrastructure for a data warehouse includes all the hardware and software components that will be needed for the data warehouse to go live. The hardware components include computers, networks, terminals, or PCs; and the software components comprises of database, extraction tools, cleaning tools, and query handling. Proper infrastructure planning is critical for a large project like a data warehouse project that often must be built up from scratch.

- **Outsourcing vs. Custom Building**

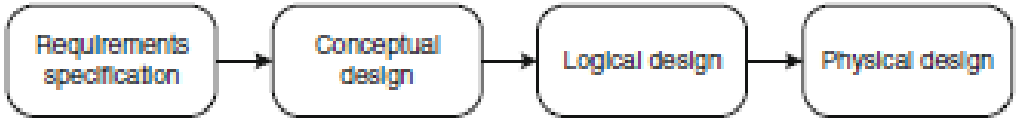
The data warehouse, like all other operational systems may be created in-house outsourced in part or outsourced completely. In other words, the outsourcing decision is a classic make-versus-buy decision which is a fairly complex one Factors like improving the in-house skills, knowledge ot how the organization works, and using full knowledge about what data is available, where it is Stored and how it is stored are considerations that push the decision towards in-house development. On the contrary, considerations that push decisions towards

outsourcing the project include lack of personnel with the appropriate skills, non-availability of data warehouse expertise, and lack of faith that the project can be accomplished in-house Since a data warehouse is usually used to Strategic analysis and is often Considered a decision-support application, the favored approach 1s to keep the warehouse project in-house, seeking the help of outside consultants in the design stage and other areas that are beyond existing expertise within the firm.

Data warehouse design stage

There are two major methods for the design of data warehouses and data marts.

- In the top down approach, the requirements of users at different organizational levels are merged before the design process starts, and one schema for the entire data warehouse is built, from which data marts can be obtained.
- In the bottom-up approach, a schema is built for each data mart, according to the requirements of the users of each business area. The data mart schemas produced are then merged in a global warehouse schema.



- In the analysis-driven approach, key users from different organizational levels provide useful input about the analysis needs. On the other hand, in the source-driven approach, the data warehouse schema is obtained by analyzing the data source systems. In this approach, normally, the participation of users is only required to confirm the correctness of the data structures that are obtained from the source systems or to identify some facts and measures as a starting point for the design of multidimensional schemas.
- Finally, the analysis/source-driven approach is a combination of the analysis- and source-driven approaches, aimed at matching the users' analysis needs with the information that the source systems can provide. This is why this approach is also called top-down/bottom-up analysis.

Conceptual schema is a concise description of the users' data requirements without considering implementation details. Conventional databases are generally designed at the conceptual level using some variation of the well-known entity-relationship (ER) model, although the Unified Modeling Language (UML) is being increasingly used. Conceptual schemas can be easily translated to the relational model by applying a set of mapping rules.

- A **schema** is composed of a set of dimensions and a set of facts.
- A **dimension** is composed of either one level or one or more hierarchies. A hierarchy is in turn composed of a set of levels. There is no graphical element to represent a dimension: it is depicted by means of its constituent elements.
- A **level** is analogous to an entity type in the ER model. It describes a set of real-world concepts that, from the application perspective, have similar characteristics. Instances of a level are called members. A **level** has a set of attributes that describe the characteristics of their members.

- Level has one or several identifiers that uniquely identify the members of a level, each identifier being composed of one or several **attributes**.
- **Instances** of a fact are called **fact members**.
- The **cardinality** of the relationship between facts and levels, indicates the minimum and the maximum number of fact members that can be related to level members.
- A fact may contain attributes commonly called **measures**.

For a **Logical Data Warehouse Design**, there are several approaches for implementing a multidimensional model depending on how the data cube is stored:

- Relational OLAP (ROLAP), which stores data in relational databases and supports extensions to SQL and special access methods to efficiently implement the multidimensional data model and the related operations.
- Multidimensional OLAP (MOLAP), which stores data in specialized multidimensional data structures (e.g., arrays) and implements the OLAP operations over those data structures.
- Hybrid OLAP (HOLAP), which combines both approaches.

In physical warehouse design, a **view** in the relational model is just a query that is stored in the database with an associated name, and which can then be used like a normal table. This query can involve base tables (i.e., tables physically stored in the database) and/or other views.

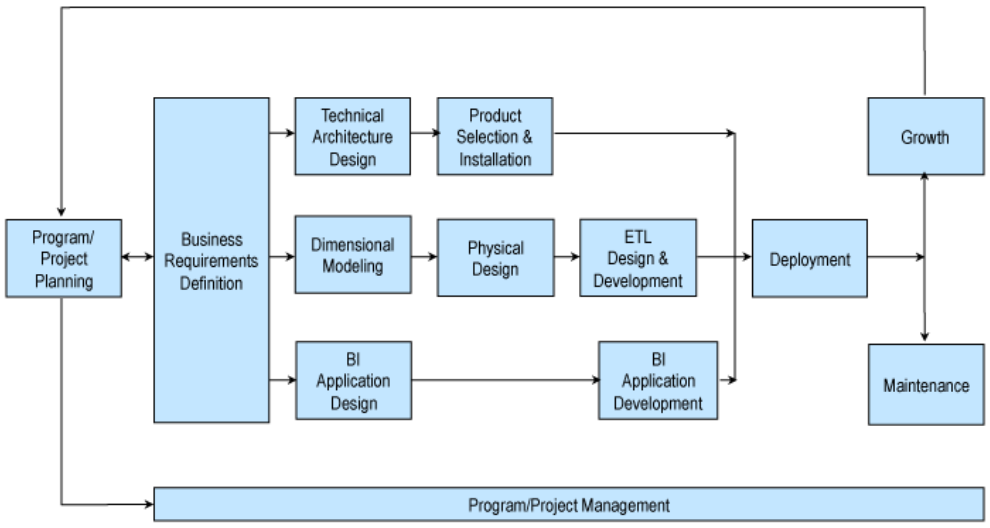
- A **materialized view** is a view that is physically stored in a database. Materialized views enhance query performance by precalculating costly operations such as joins and aggregations and storing the results in the database. In this way, queries that only need to access materialized views will be executed faster. Obviously, the increased query performance is achieved at the expense of storage space. A typical problem of materialized views is updating since all modifications to the underlying base tables must be propagated into the view. Whenever possible, updates to materialized views are performed in an incremental way, avoiding recalculating the whole view from scratch. This implies capturing the modifications to the base tables and determining how they influence the content of the view.

Two common types of indexes for data warehouses are bitmap indexes and join indexes.

- **Bitmap indexes** are a special kind of index, particularly useful for columns with a low number of distinct values (i.e., low cardinality attributes), although several compression techniques eliminate this limitation.
- **Join indexes** materialize a relational join between two tables by keeping pairs of row identifiers that participate in the join. In data warehouses, join indexes relate the values of dimensions to rows in the fact table.
- **Partitioning or fragmentation** is a mechanism frequently used in relational databases to reduce the execution time of queries. It consists in dividing the contents of a relation into several files that can be more efficiently processed in this way.
    - **Vertical partitioning** splits the attributes of a table into groups that can be independently stored. For example, a table can be partitioned such that the most often used attributes are stored in one partition, while other less often used attributes are kept in another partition. Also, column-store database systems make use of this technique.

- **Horizontal partitioning** divides the records of a table into groups according to a particular criterion. A common horizontal partitioning scheme in data warehouses is based on time, where each partition contains data about a particular time period, for instance, a year or a range of months.

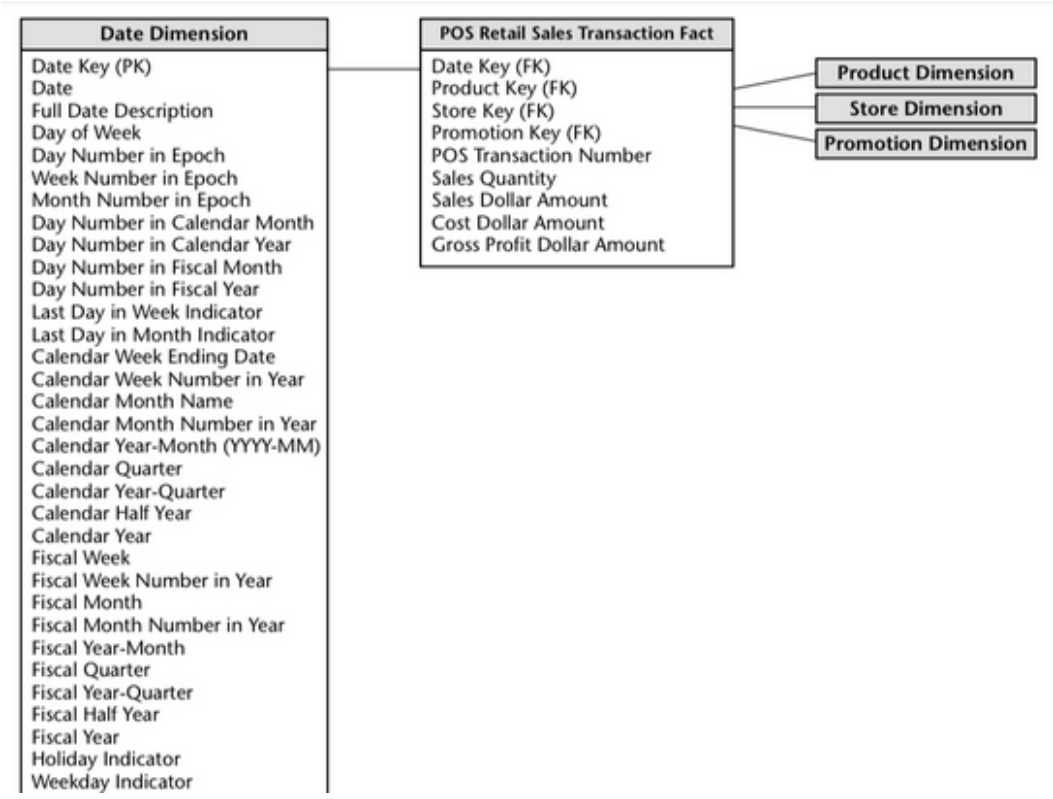
Ralph Kimball identified a nine-step method as follows:
1. Choose the Subject Matter: Identify a specific business process or area of interest for the initial data mart.
2. Decide what the Fact Table Represents: Define the central event or activity being measured in the fact table.
3. Identify and Conform the Dimensions: Determine the descriptive attributes (dimensions) associated with the fact and ensure consistency across other data marts.
4. Choose the Facts: Select the quantitative measures (facts) to be stored in the fact table.
5. Store Pre-calculations in the Fact Table: Include frequently used calculations for faster query performance.
6. Define the Dimensions and Tables: Specify the structure and relationships of dimension tables.
7. Decide the Duration of the Database and Periodicity of Updation: Set data refresh frequency and historical data retention period.
8. Track Slowly Changing Dimensions: Implement strategies for handling changes in dimension attributes over time.
9. Decide the Query Priorities and the Query Modes: Prioritize the types of queries and access methods users will employ.

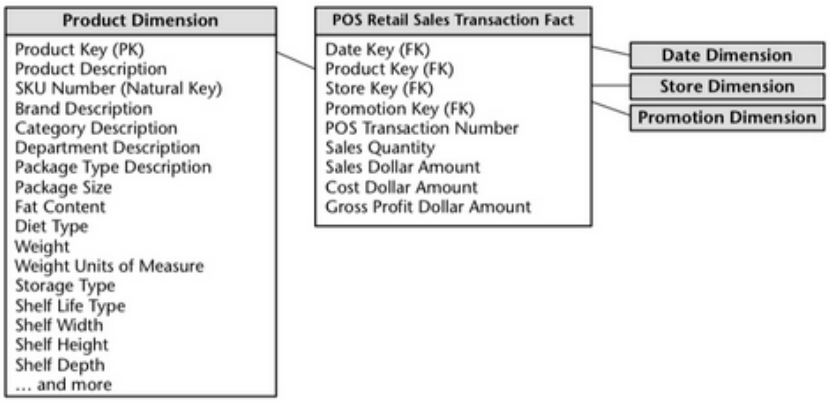


The design stage of a data warehouse comprises of the following sequence:

- **Design the Dimensional Model**
  User requirements guide dimensional modelling which is done as a part of the design phase. The dimensional model must address business needs, grain of detail, the dimensions and the facts to include, the dimensional model must be in tune with the requirements of the end-users so that they easily use it for direct access. The model must also be designed so

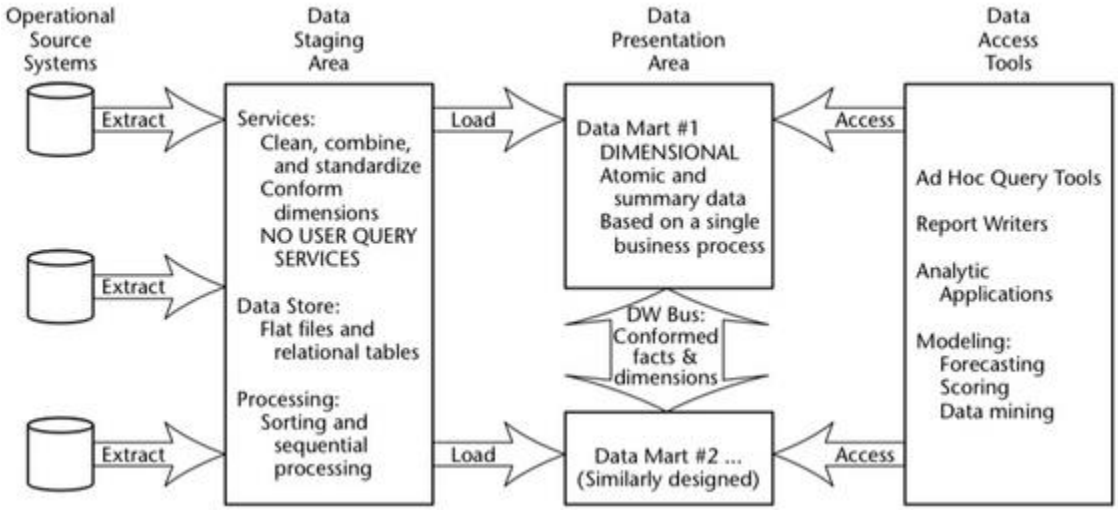
that it is easy to maintain and is flexible enough to adapt to future changes. An operational system is based on a normalized structure to minimize data redundancy, allow validation of input data, and support many transactions that occur repeatedly. In these systems, a transaction may just involve a single business event, such as placing an order. Thus, an operational system model often looks like a spider web of hundreds and thousands of related tables. In contrast, in a data warehouse environment, the dimensional model uses a star schema or a fact constellation schema that is easy to understand and relate to business needs, supports complex business queries, and provides higher query performance by minimizing table joins.





- **Develop the Architecture**

  The data warehouse architecture reflects the dimensional model developed to meet the business requirements. The dimension model specifies the dimension table design and fact definitions, thereby specifying the tact table design and finally the relationship between these fact tables and the dimension tables. Whether to create a star or snowflake schema is not a trivial issue, and it depends more on implementation and maintenance considerations than on business needs. Generally, data warehouse schemas are quite simple and straightforward, in contrast to that of operational database schemas that have hundreds or thousands of tables and relationships. The primary goal of the data warehouse schema is to handle a large quantity of data which can affect the performance and efficiency of the queries. A star schema is good enough for this purpose.



- **Design for Update and Expansion**

  Data warehouse architectures must be designed to absorb the ongoing data updates and cater for future growth with minimum impact on the existing design. The dimensional model and the star schema simplicity these activities with records being entered in the fact table with little effect on most dimensions. For example, a sale made of an existing product, by an existing Salesperson, at an existing store, to an existing customer will not call for any changes to be made on the product, customer, salesperson, or store dimension tables. But in case, the product was to be sold to a new customer, then a new record had to be added to the customer dimension table when the record is added to the fact table. The historical nature of data warehouses implies that records are not deleted from the tables; only updates and additions are done to the existing data in the warehouse. The dimensional model helps for easy expansion, as new dimension attributes and new dimensions can be added without dimensions, its affecting existing attributes, and other historical data stored

in it with due course of previously. time, no doubt, the data warehouse applications will need t be extended, but the Sone existing applications should remain intact. Although applications may be updated to make use of the new functionality information, should remain the same. An entirely new business subject area added including to new a data fat table and dimension tables and thus a new schema can be warehouse without affecting the existing function.

- **Design the Relational Database and OLAP Cubes**

  In this phase, the star schema that is made fact up of a number of dimensions and fact tables is created, surrogate keys are defined, and primary and foreign e relationships are established. Then, any sort of views, indexes, and tact ta partitions are created followed by the design of OLAP cubes to support Strategic needs of the users. Tables are implemented in the relational database after surrogate keys dimension tables have been defined and primary and foreign keys have been established. Views must be created for those end-users who need direct access to the data stored n the warehouse. Appropriate access must be granted to the users so that they can access the underlying data. Generally, the indexed view are designed to improve query performance. OLAP cube design requirements w be designed to support the way the users want to query the data.

- **Decisions in Design**

  Critical decisions have to be taken at various levels of design. Discussed below are the details of the issues which are to be resolved at various stages of design.

  - **Design decisions: Organization of the warehouse**
    - ✓ How much data the data warehouse must handle?
    - ✓ What is the number of tiers that will be present in the architecture of the warehouse?
    - ✓ What structure of the warehouse is desired: a centralized warehouse or a distributed warehouse?

  - **Design decisions: Back-end**
    - ✓ What will be the source of data (external, internal, production, and archived)? Which relational DBMS will be used?
    - ✓ What will be the extraction tools that will be needed for extraction and transformation of data?
    - ✓ Determine the interval between data loads. The more frequently the data is loaded, the more up-to-date it is.

❖ **Design decisions: Data warehouse**

- ✓ The subjects to be included in the warehouse.
- ✓ For decision support system which technology to use: ROLAP or MOLAP.
- ✓ Deciding the level of granularity at which the data must be stored. Granularity involves a trade-off between increased detail and speed of computation.
- ✓ Data Can be summarized in a variety of ways so choose the summaries that must be provided.
- ✓ To minimize the response time of the queries, some warehouses precompute: the answers to frequently asked queries. The trade-off is extra storage for speed of response. So the question is how much precomputation must be done and till what levels.
- ✓ Data in the warehouse is moved from one level of storage medium to another, depending on its age.
- ✓ Decisions to create, update, and maintain the metadata.
- ✓ Decisions involving the computing capability of the data warehouse whether it should have parallel or serial computing capability.

❖ **Design decisions: Front-end**

- ✓ In data warehousing environment, the users want to access the data at various levels of details. So, decisions have to be made regarding the arrangement of data according to various levels of detail.
- ✓ Which users will have access to the data, and which services will be provided to different categories of data warehouse users.
- ✓ Decision regarding the front end of the project as the front-end is the user's prime point of contact with the warehouse. Hence, it needs to be designed carefully.

❖ **Design decisions: Maintaining the system**

- ✓ Provisions for monitoring the warehouse need to be designed so that improvements, updates, and changes can be done easily.
- ✓ Decisions must be made regarding the subject areas. For example, it should be clear much before time that if it is required to incorporate more subject areas in future, how the situation will be handled.
- ✓ Decisions regarding the security of the data warehouse system are crucial as well.

- **Detail Design**

  In this stage, the database schema is developed, the metadata is defined, and the source data is expanded to include all the necessary information needed for the subject area and is validated by the users. In this phase, detailed designing of all

procedures that will be implemented for the data warehouse is completed and documented. These procedures are designed to accomplish the following activities.

- ✓ Data warehouse capacity expansion
- ✓ Purging and archival of historical data.
- ✓ Data extraction, transformation, loading and cleansing functions
- ✓ Configuration management.
- ✓ Security.
- ✓ Testing of every individual module of the data warehouse and the entire system.
- ✓ Data refresh.
- ✓ Data access
- ✓ Data backup and recovery.
- ✓ Disaster recovery.
- ✓ Transition to production.
- ✓ User training and user support.
- ✓ Change management.

Advantages of the Kimball Approach

- The first part of the data warehousing project will be provided fast because it is simple to set up and build.
- The star schema is simple to understand for business users and implement for reporting. Most BI (business intelligence) solutions are compatible with the star schema.
- The data warehousing environment has a small footprint; it takes up less space in the database and makes system maintenance more manageable.
- A small group of developers and architects is all that is required to keep the data warehouse running smoothly.
- Because data marts are designed for departmental or business process-level reporting, it works very well for department-level metrics.
- Drill-across, in which a BI tool generates a report by traversing various star schemas, may be achieved successfully with conformed dimensions.

Disadvantages of the Kimball Approach

- Because data is not fully integrated before addressing reporting purposes, the core of the '**one source of truth**' is lost.
- Over time, redundant data might produce data update abnormalities.
- Adding columns to the fact table can slow things down. This is because tables are built to be very deep. The fact table will get significantly more prominent and perform poorly if more columns are added. As a result, changing the dimensional model as the business requirements vary is difficult.
- Because the model targets business processes rather than the enterprise, it cannot meet all enterprise reporting demands.

- Integrating legacy data into a data warehouse is a time-consuming procedure.

Building and implementing data marts

Although a data mart is smaller in size as compared to a data warehouse and hence easier to install, some issues like data loading, vendor performance, and software integration can make this task a bit more difficult. To minimize such problems, some vendors offer "data mart in a box" that provides all the required components through one-stop shopping. However, such data marts may not be able to solve all the problems.

- **Building data warehouse**
  After completing the work of building and implementing various data marts at the department level, the next stage is to build a complete data warehouse from these data marts following a bottom-up approach. In case you want to follow a top-down approach of building a data warehouse, you first have to build the complete enterprise-wide warehouse and then make individual data marts from it. Thus, the design decision varies from organization to organization. The required hardware, software, and middleware components are acquired and installed, establishment of the development and test environment is done, and the configuration management processes are implemented. Software programs to extract, clean, transform, load the source data, and periodically refresh the existing data in the data warehouse are written These programs must be unit tested against a test database with a sample e source data. Technical and business metadata are loaded in the metadata repository. Canned production reports are produced, and sample ad hoc queries are run against the test database. to measure the validity of the output. Users are provided access to the data warehouse. Once all the components are in place, system functionality and user acceptance testing is conducted for the complete integrated data warehouse system. All system support processes like that of database security, backup and recovery, disaster recovery, data purging, and archival are implemented and tested as the system is prepared for deployment.

- **Test and Deploy the System**
  The end-users must be a part of the testing process. After the initial testing is conducted by the development and test teams, users should use the system to execute their queries, form their reports, and do any kind of analysis they wish to do. User involvement in the testing phase provides a significant number of benefits, such as:
  - ✓ Errors and discrepancies can be found and corrected.
  - ✓ Users can familiarize themselves with the system.
  - ✓ Tuning of indexes can be done.

✓ When the users exercise the system during the test phase, with all kinds of queries that they will be generally executing, a considerable amount of empirical index tuning can be done even before the system comes online. Although additional tuning will have to be done after

the deployment, the system starts with a satisfactory performance. Generally, testing and reviews of the data warehouse will be done throughout the development of the system. It is crucial to appoint a single individual in charge of the testing and review process. The person in charge should be empowered with the appropriate authority and must identify most of the problems before they become too big to handle. We will study more about testing the data warehouse in later chapters. After testing the data warehouse, training sessions are scheduled concurrently with the installation to make extinctive use of time. The best way to ensure success is to effectively train the users so that they will use the system and possibly sing its praises. These training sessions are however ongoing sessions because new employees or employees being moved or promoted will need to be trained. Also, whenever there are some enhancements made to the system, new training sessions must be scheduled.

- **Transition to Production**

  This phase transfers the data warehouse project into the production environment. The production database is created, the extraction, and transformation routines are run on the operations system source data. The project team now works with the operational system staff to perform the initial loading of the data warehouse and execute the first refresh cycle. The staff 1s given appropriate training and the programs and applications are moved into the production libraries and catalogs. Good explanatory presentations and tool demonstrations are given to the end-users. The help desk is established and made operational. A Service Level Agreement is signed by the organization. Finally, a Change Management Board is established for ongoing maintenance and the implementation of change control procedures for future development cycles.

- **User Training and Support**

  The importance of training and orientation for the users cannot be neglected From an IT department's perspective, the end-users must be trained mainly in three areas: data, applications, and tools. But, on the other hand, the end users see the training sessions as one. It is very important that the training program be designed from the user's point of view. The data warehouse otters many more capabilities and has a lot more potential than that of an operational system. In a data warehouse environment, users are not aware of how much they can really do with the tools in the data warehouse.

  - ✓ Database and data storage concepts.
  - ✓ Predefined queries and reports.
  - ✓ Analysis that can be performed on the stored data.
  - ✓ Features of the data warehouse.
  - ✓ Contents of the data warehouse.
  - ✓ Scheduling and delivery of data reports
  - ✓ Browsing through the warehouse Contents of data and metadata.

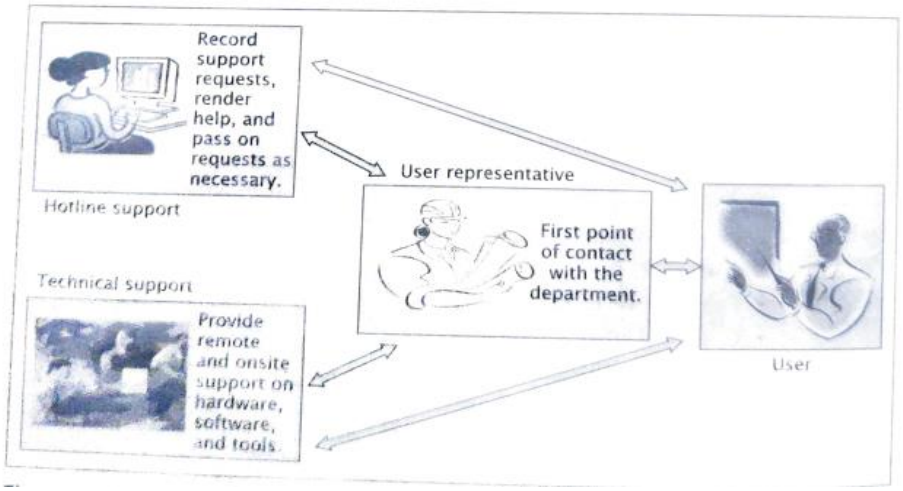
- ✓ Usage of query templates.
- ✓ Data loading schedules and the Currency of data stored in the warehouse.
- ✓ Making efficient use of data access and retrieval tools.
- ✓ Usage of the Web for information delivery.

The first step in formulating a schedule lor user training is to determine the areas where the users need to be trained. Try to match the content to the training with anticipated usage of the warehouse. If users use only predefined queries and reports, then the job of training is easier, but this cannot be true case of a data warehouse. The data warehouse end-users will formulate their own ad hoc queries and perform complex analysis. Thus, the content of the training program needs to be more intense. The training program must be both deep and wide to cater to the needs of all User groups. The other consideration for preparing the contents of the training program is that the different users of the data warehouse have different levels of skills and knowledge. Hence, the training program must cater to the needs of every individual user. Among other things, there are three basic components that must be included as a part of the training program:

(i) The users must be aware of what data is available for them in the data warehouse. They must be able to navigate through the contents to find the data that they need.

(ii) The users must be aware of the different applications that they can perform. They must be informed about the predefined queries and reports and pre-constructed applications; and

(iii) Train the users on the tools that they would be using to access the information. In the early days of post-deployment of the data warehouse, every member of the support staff is busy with solving user's wide range of questions, from basic sign-on to performing complex analysis to other hardware issues. The users need a lot of spoon-feeding, at least during the initial stages.



AS seen in the figure, the user representative is the first point Within to contact the department, this person must be trained well enough to most of the answer questions on applications, data content of the warehouse, and the er tools. The hotline support comes into picture when the user representative 1S unable to solve user problems. Apart from user training, emphasis should also be given to user because user frustration tends to support increase in the absence of a sound structure.

- **Success Factors of a Training Program**
  Once you have decided the topics on which the users have to be trained, then the next step is to determine who should be given the responsibility of the relevant content/course material. preparing material, It is only after you have the course ready with you that you can proceed with training good course material calls the users. Preparing for a lot of effort and this underestimated. activity cannot be underestimated. Let us now look at the various tasks needed to prepare a program. The good training program will of course vary based on the of the requirements organization. Given below are a few general points that must be considered to form a solid user training program.

✓ The success of a training program depends on the joint participation of user representatives and the IT. User representatives on the project team and the subject area experts in the user departments must work with IT. Let both jointly prepare the contents of the course material.

✓ Do not forget to include all the topics of data content, applications, and tools in the training program.

✓ Divide the users who need to be trained into different groups based on their level of skills and knowledge.

✓ Determine the topics on which each group needs to be trained so that the training program can be tailor made to match the requirements of the organization.

✓ Determine how many different training courses would be needed to train the entire group of users. The set of courses will include an introductory course, an in-depth course, and a specialized course on tool usage. Make sure that the introductory course runs to one day and every user of the data warehouse attends it.

✓ Design the in-depth course in several tracks in such a way that each track Caters to a specific group of users and covers one or two subject areas.

✓ Make the course documentation simple so that it is easy for the users to understand it. Include enough graphics and pictures. For example, the course covers dimensional modelling, then makes a sample star schema so that the users can visualize how the relationships are termed between the tables.

✓ The introductory course material can just be a demo or a theory lecture, but the in-depth course and the end-user tool usage course must provide hands-on experience to the users.

It is always better to complete the training session even before the deployment of the first version of the data warehouse. What the users la during the initial training sessions will remain fresh in

their minds when they start working with the data warehouse. However, ongoing training sessions will continue for additional sets of users. With the implementation of the new versions of the data warehouse, modification of the training materials must be done when they are done. There are always some users who would ask for refresher courses, The users who could not be trained during the initial training session also have to be trained as a part of the ongoing training. Users have their own responsibilities to run the business, and they need to find time to fit into the training slots. In a data warehouse environment, special consideration needs to be given to training the executives and sponsors, as they would also be using the data warehouse to run queries and produce the desired reports. Some of them may not even know how to look for the information they are interested in. Employees at a higher level of the organization may not be interested in attending the training sessions along with user staff. So, separate training sessions need to be arranged for them. It is not uncommon to find that even after training sessions are completed, there may still be some users who have not been trained, as some of them may be too busy to be able to get away from their business responsibilities or some of them may think that they need not attend any formal course and that they can learn by themselves. The organization must have a strict well-defined policy to cater to such a situation. When the users access the data warehouse without any formal training, then two things may happen. Either they will disrupt the support structure by asking for too much attention or if they are unable to perform any function, they will blame the system without attributing it to the lack of training. However, the best policy to be followed in this scenario is "No training means no data warehouse access.

- **Issues in User Support**
  Let us review some of the general issues that are important in context of the data warehouse.

✓ Let every user be clear about the support path to be taken. Every user must know whom to contact first it they have any problem regarding software, hardware, applications, or the tool.

✓ In a multi-tiered support architecture, the roles and responsibilities of every tier must be clearly specified.

✓ While using the data warehouse, many users will try to match the results obtained from the data warehouse with results obtained from the operational systems, so the support structure must also be able to address such data reconciliation issues.

✓ Include support on how to find and execute predefined queries and preformatted reports and finally how to navigate through the contents of the warehouse.

✓ The user support structure will provide you with an open channel tor communicating with the users and getting their feedback.

Backup and Recovery

A data warehouse stores huge amounts of data that may have taken years to collect. The historical data in the data warehouse may be as old as 10 or even 20 vears. Before storing the data in the data

warehouse, the data goes through a rigorous process of cleansing and transformation. So, the users cannot afford to lose this data and it is highly desirable that you are able to recreate the data as and when required. Within a short time after the deployment, the number of users, the complexity of their queries, and the duration of their analysis sessions reach great heights. In this situation, backing up the data content and the ability to recover quickly from malfunctions becomes even more sophisticated. In operational systems, the data is backed up regularly. Whenever a disaster occurs, the recovery process proceeds from the last backup and recovers to the point where the system stopped working. Some people argue that data backup is not required in a data warehouse environment, as the data warehouse does not represent an accumulation of data directly through data entry. It has been taken from the source systems, whenever so Convenient any disaster occurs, data will be recreated from the source systems. it may sound, but it is rather impractical as recreation would consume a lot of time and the data warehouse users cannot tolerate such periods of downtime. long Therefore, you need to develop a clear and well-defined backup and recovery strategy.

A sound backup strategy comprises several crucial factors. Let us go over some of them.
Here is a collection of useful tips on what to include in your backup strategy:

- † Determine what you need to back up. Make a list of the user databases, system databases, and database logs.
- † The enormous size of the datawarehouse stands out as a dominant factor. Let the factor of size govern all decisions in backup and recovery. The need for good performance plays a key role.
- † Strive for a simple administrative setup.
- † Be able to separate the current from the historical data and have separate procedures for each segment. The current segment of live data grows with the feeds from the source operational systems. The historical or static data is the content from the past years. You may decide to back up historical data less frequently.
- † Apart from full backups, also think of doing log file backups and differential backups. As you know, a log file backup stores the transactions from the last full backup or picks up from the previous log file backup. A variation of this is a full differential backup. A differential backup contains all the changes since the last full backup.
- † Do not overlook backing up system databases.
- † Choosing the medium for backing up is critical. Here, size of the data warehouse dictates the proper choice.
- † Commercial RDBMSs adopt a “container” concept to hold individual files. A container is a larger storage area that holds many physical files. The containers are known as table spaces, file groups, and the like. RDBMSs have special methods to back up the entire container more efficiently. Make use of such RDBMS features.
- † Although the backup functions of the RDBMSs serve the OLTP systems, data warehouse backups need higher speeds. Look into backup and recovery tools from thirdparty vendors.
- † Plan for periodic archiving of very old data from the data warehouse. A good archival plan pays off by reducing the time for backup and restore and also contributes to improvement in query performance.

Data warehouse will be used by many users for a constant flow of information. But the huge size of the data warehouse is a serious factor that affects all decisions about backup and recovery. At the time of a disaster, re-extracting data from the source systems and reloading the data warehouse is not viable. So, we need to set up a practical schedule for performing backups. However, there are several issues affecting this practical schedule. First, we must think of scheduling the backups during the night because in a data warehouse environment, the night slots are usually allocated for the daily incremental loads. Second, in case the data warehouse end-users are in different time zones, then finding a time slot becomes even more difficult. Thus, finally setting up a backup schedule comes down to a question like "How much downtime the users can tolerate before the recover process gets completed?

A practical backup schedule for your data warehouse certainly depends on the conditions and circumstances in your organization. Generally, a practical approach includes the following elements:

- † Division of the data warehouse into active and static data
- † Establishing different schedules for active and static data
- † Having more frequent periodic backups for active data in addition to less frequent backups
- for static data
- † Inclusion of differential backups and log file backups as part the backup scheme
- † Synchronization of the backups with the daily incremental loads
- † Saving of the incremental load files to be included as part of recovery if applicable

Backup terminologies

There are two major backup types:

1. Physical Backup: This is a copy of physical database files such as data, control files, log files, and archived redo logs. It is a copy of the files that store database information in another location and forms the foundation of the database recovery mechanism.
2. Logical Backup: It contains the logical data that is extracted from a database, and it consists of tables, procedures, views, functions, etc. However, logical backups alone are not recommended or useful since it only provides structural information..

Others include,

- **Complete backup** − It backs up the entire database at the same time. This backup includes all the database files, control files, and journal files.
- **Partial backup** − As the name suggests, it does not create a complete backup of the database. Partial backup is very useful in large databases because they allow a strategy whereby various parts of the database are backed up in a round-robin fashion on a day-to-day basis, so that the whole database is backed up effectively once a week.
- **Cold backup** − Cold backup is taken while the database is completely shut down. In multi-instance environment, all the instances should be shut down.

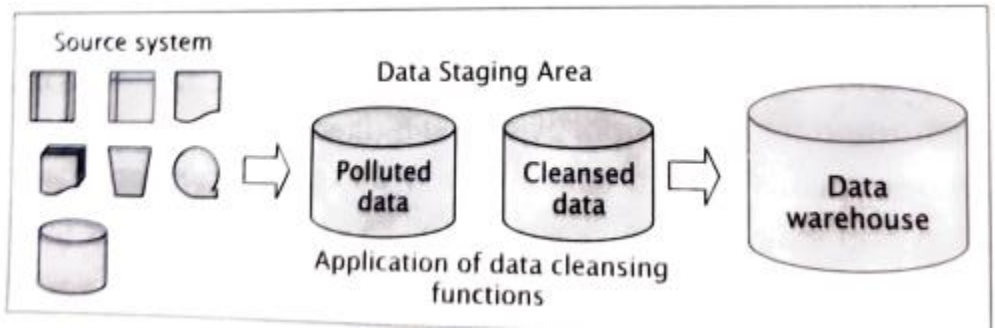
- **Hot backup** − Hot backup is taken when the database engine is up and running. The requirements of hot backup varies from RDBMS to RDBMS.
- **Online backup** − It is quite similar to hot backup.
- **Full Backups:** Entire data warehouse is copied during the backup process to create a complete snapshot of the data at a specific point in time.
- **Incremental Backups:** Only changes made since the last backup are stored, reducing backup time and storage requirements.

Establish the data recovery quality framework.

Data Warehouse Recovery is important for several reasons:

- **Data Integrity:** Data warehouses often contain critical business data. Ensuring the integrity and availability of this data is crucial for business operations and decision-making.
- **Business Continuity:** System failures or data loss can disrupt business operations. Having a robust recovery plan in place ensures that operations can be resumed quickly, minimizing downtime and potential financial losses.
- **Compliance Requirements:** Many industries have regulatory requirements for data retention and availability. Data Warehouse Recovery helps organizations meet these compliance obligations.
- **Protecting Investments:** Data warehouses require significant investments in terms of infrastructure, resources, and implementation efforts. Data Warehouse Recovery protects these investments by minimizing the impact of data loss or system failures.

Data Purification To proceed with the purification process, divide the data elements into priorities with the help of users. You may simply categorize all the data elements into three levels of priority: high, medium, and low. Achieving 100% data quality is critical for the high-priority data elements. The medium-priority data requires as much cleansing as possible and some errors may be ignored to make a balance between the cost of correction and the potential effect of bad data. The low priority data may be cleansed if you have any time and resources left. Begin the data cleansing efforts with the high-priority data and then move on to the medium-priority data, and so on.



A universal data corruption problem relates to duplicate records, so make sure that the overall data purification process includes correcting techniques TO the duplication problem. However, pollution of data can ais introduced in the data warehouse from the erroneous external data. If the

company has paid for the external data, then you have the right to demand a warranty on data quality. But in case you have utilized the external data that is freely available on some public domain, then you need to have some kind or data quality check.

† Test the recovery procedure carefully. Conduct regular recovery drills.

† Considering the conditions in your organization and the established recovery procedure, estimate an average downtime to be expected for recovery. Get a general agreement from the users about the downtime. Do not surprise the users when the first disaster strikes. Let them know that this is part of the whole scheme and that they need to be prepared if it should ever happen.

† In the case of each outage, determine how long it will take to recover. Keep the users.

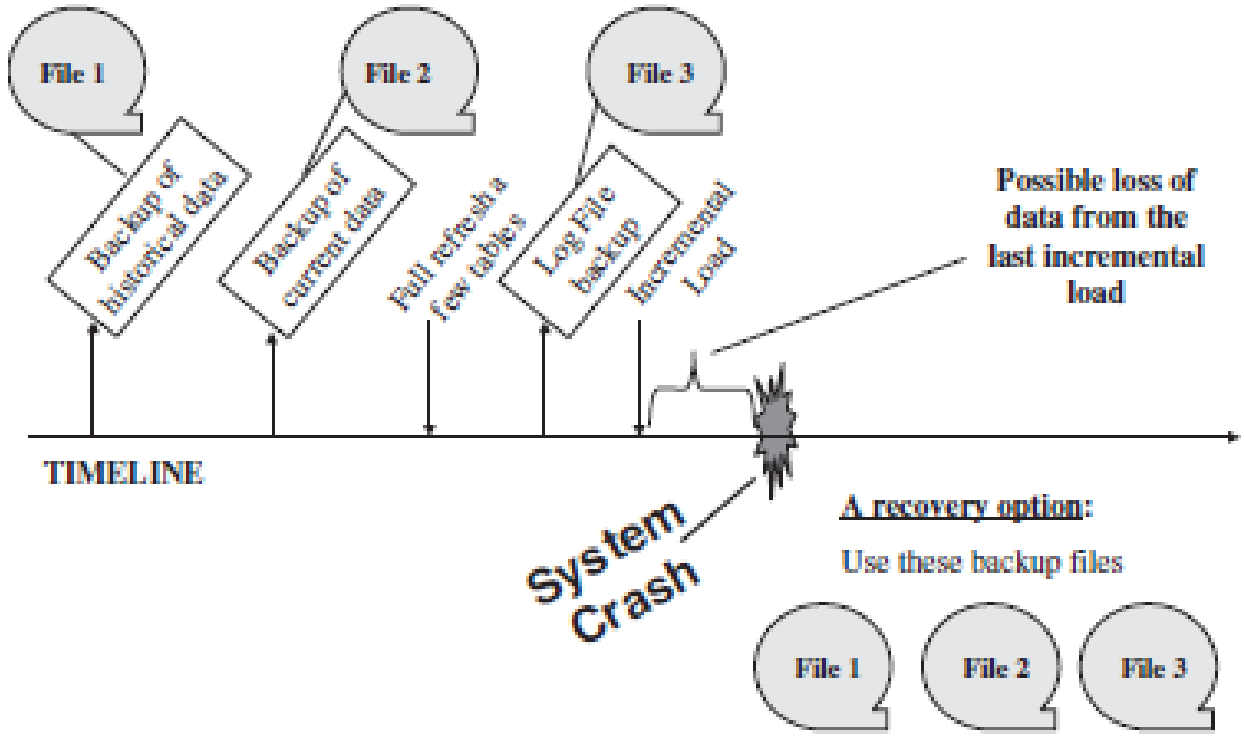
properly and promptly informed.

† Generally, your backup strategy determines how recovery will be done. If you plan to

include the possibility of recovering from the daily incremental load files, keep the

backups of these files handy.

† If you must go to the source systems to complete the recovery process, ensure that the sources will still be available.



Data Warehouse Recovery has several important use cases:

- **Disaster Recovery:** In the event of major disasters like natural calamities, cyberattacks, or hardware failures, Data Warehouse Recovery ensures that data can be recovered and business operations can be restored.
- **Data Corruption:** Data corruption can occur due to software bugs, human errors, or system malfunctions. Data Warehouse Recovery helps in detecting and recovering from such corruption.
- **Data Loss:** Accidental deletion, hardware failures, or software bugs can lead to data loss in data warehouses. Data Warehouse Recovery helps in recovering lost data and minimizing the impact.
- **Data Migration:** When migrating from one data warehouse environment to another, Data Warehouse Recovery ensures a smooth transition by allowing the data to be recovered and restored in the new environment.

Operating the warehouse

Until now, we have concentrated on the design aspects of the data warehouse. However, when the warehouse is built and the initial loading 1S completed, it is time to make the data warehouse operational.

Day-to-day Operations

The main use of the data warehouse during the daytime is to service the user's queries. Other operations that can occur during the day are.

- Query management.
- Backup and recovery management.
- Performance management.
- Running of housekeeping scripts.

Query management is a vital part of daily operations. Some of the monitoring and control can be automated, but a DBA is still needed to solve any problem. If there is any job that is exceeding its quota of resources, then that job needs to be killed immediately. Also, there are certain housekeeping scripts that need to be run repeatedly to log off idle connections. Generally, backup activities are performed during the night to avoid contentions with the user queries during the day.

Warehouse Administration

Continuous monitoring and administering activities in the data warehouse must be done simultaneously. The basic principles here are:

- The data warehouse will change over time, so hardware and software require maintenance and updating.
- Automated procedures for routine operations should be present.

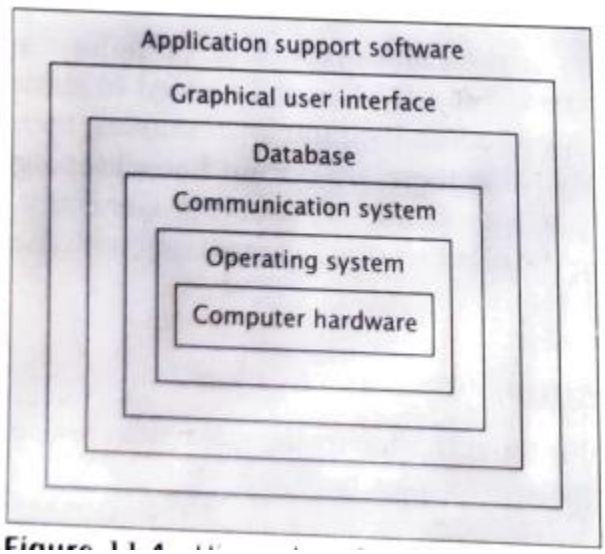
The responsibility for administering the data warehouse and data marts lies in the hands of the IT department. One approach, proposed by Inmon and Hackathorn in their book Using the Data Warehouse (1994), is to form a data architecture group which would intertace with all the concerned parties (management, IT, and end-user) and be responsible for the data warehouse, Its duties include: Monitoring the warehouse operations

Monitoring the Warehouse Operations

Monitoring the data warehouse involves activities that track the growth and usage of the data warehouse. It is not uncommon for the growth pattern to be different from what was predicted and designed. Reviewing the end-user activity gives an indication of which data is useful and which is not. The warehouse is also continuously monitored to see that the user queries perform well and do not take more time to execute.

Platform Upgrades

The data warehouse platform comprises of the data transport component, end user information delivery, data storage, metadata, the database and the OLAP component, components. Thus, a data warehouse is basically a cross-plat- form environment. Over time, upgrades to these components are brought into the market by its vendors. After deploying the first version of the data warehouse, make a proper plan for the new applying releases of the Technically platform components. speaking, upgrades must be properly managed, lest they cause serious potentially work. Good interruption to the normal planning minimizes this. Therefore, disruption it is the always better to schedule upgrades at some convenience based on Operating system Computer hardware when users can tolerate interruptions.



Addition and Deletion of Subject Areas

As the business of a firm changes, its interests also warehouse change. Since the data is organized around subject areas, major changes will typically involve the subject areas covered by the warehouse.

Managing Data Growth

In the data warehouse environment, managing data growth calls for attention. Until and unless special you are vigilant about data growth, it could get out of hand very soon. Data warehouses already contain large volumes of data and even a small increase can result in substantial additional data. In the first place, a data warehouse contains too much historical data which may even go beyond 10 years. End-users also tend to keep the data at the detailed level and want some aggregate or summary tables in addition the detailed data which adds to the data stored in the data warehouse. In t course of usage of the data warehouse, this data will always become more in volume. So, follow some practical approach to manage data growth, e-g you may archive the old data promptly.

ETL Management

ETL management is an ongoing function, so it must be automated as much as possible. Given below are some useful suggestions on data extraction, transformation, and loading management.

- Run daily extraction jobs on schedule and if for some reasons, source systems are not available, then reschedule the extraction jobs.
- Make sure that all rules for data transformation and cleansing are strictly followed.
- Resolve exceptions detected by the transformation and cleansing functions.
- Verify the data before loading it in the warehouse. Cross-check if the referential integrity constraint is properly followed.
- Ensure completion of daily incremental loads as per the schedule.

Supporting End-users

The end-users need a formal training before they could actually start working with the warehouse. Once they become familiar with the warehouse, their re quests for help become more sophisticated. Whenever the data warehouse system produces unexpected results or no results at all, the users need a helpdesk to get their problems solved. Users also need help for executing complex queries. In case the users want to execute the same set of queries repeatedly, then the IT people can build pre-formatted reports to make that query simple. In order to assist the users, the warehouse design team builds a metadata so that the users can use it to know exactly what data is present in the warehouse. Finally, the warehouse design team will also support changes in user's requirements by addition and deletion of the subject areas in the warehouse.

Maintaining the Metadata

Since the contents of the warehouse keep changing with time, the metadata will also change to reflect the current status of information in the warehouse. Even though, tools for automatic maintenance of metadata are available, manual support for editing the metadata is still needed/

Upgrading the Warehouse

With time, it is often necessary to upgrade the data warehouse contents or performance. Successful data warehouse systems will cause increasing demands from the users and thus, additional data may be more, if needed. Further the system is successful, its usage and maybe the users will increase thereby making the response time longer. At this stage, the need to data warehouse upgrade hardware and software arises. To make such upgrades possible, the system may have to be shut down for that duration. But finding the time window when this can be done is not a trivial task, especially when the warehouse is being heavily used by its users. Finally, changes in the warehouse may give a way for errors to creep in the warehouse environment, so it is always recommended to test the upgrades on a separate test system before applying them to the actual warehouse. Generally, overnights and weekends are the most preferred times to upgrade the data warehouse.

Storage Management

With the increase in the volume of data, the utilization of storage area also increases. Since data warehouses are designed to store a large amount of data, the storage costs tend to be very high. It is not misleading to say that the storage costs are almost four or five times the software costs. So, you cannot overlook the issues involved in storage management. Given below are some tips that are used as guidelines for managing storage space.

- Plan for the increase in storage amount needed soon.
- Ensure that the storage configuration is easily scalable so that more storage can be added with minimum problems.
- Try to use modular storage systems.
- If the data warehouse is deployed in a distributed environment with multiple servers having their own storage pools, consider connecting the servers to a single storage pool that can be intelligently accessed.
- Ensure that it is easy to shift data from bad storage sectors.

Capacity Planning

With the growth of data warehouses, the amount of data stored in it as well as subject areas that the warehouses cover also grows. With demand of data increasing posed in the data warehouses, the data warehouse seems to be that big enough at the time of implementation now seems too small to handle the user's demands. Capacity planning is needed so that additional hardware, software,

and other resources are added, and the contents or warehouse are modified to fit within the existing capabilities.

Maintaining Security

Data warehouse contains an enormous amount of data. The utmost care should be taken t see that only authorized people are able to access and modify content stored in the warehouse. The warehouse is more sensitive to the security point front of View than any other operational database because it contains all the data that will be highly desirable by the organization's competitor. Log-in control is the most common routine security operation. Because people are given access privileges when they log in, care must be taken to make certain that individuals are not given too much or too little access to their needs and responsibilities. Care must be taken that users with special privileges do not cause inadvertent changes to the data in the warehouse. Only limited people should be given special privileges. The next security operation is to maintain logging information in a separate file. Although logging and tracing consume a large amount of memory, they are necessary to be maintained for the security of the data warehouse. Hence, the optimal solution to this problem is to move all the data related to logging of users to a separate location. Security of data also refers to maintaining data backups and provisions for recovery if the data is lost. Backups are generally taken overnight. The backup of the data will help in case of data losses that occur because of computer crashes and natural disasters like fires and earthquakes.

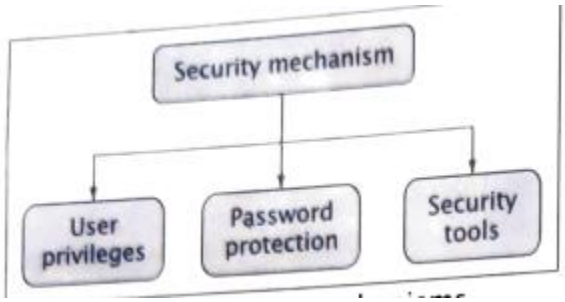
Security issues in a warehouse

A warehouse is like a gold mine of information, as all the organization's critical information is readily available in a format that makes it easy to retrieve and use. Besides, the security provisions must cover all the information that is extracted from the data warehouse and stored in applications such as OLAP. Access may be granted to the individual tables. But in a data warehouse environment, access restrictions are difficult to set up. For example, an analyst may start an analysis session by getting information from one or two tables and then fetch information from other tables. The entire query process is query centric and ad hoc in nature. So, which tables are to be restricted and which one should be opened for the analyst is not a trivial question to answer. It is critical for the data warehouse project team to establish a sound security policy for the data warehouse. This policy must first recognize the immense value of the data stored in the data warehouse and then design a few guidelines for granting privileges and instituting user roles. Given below are the usual provisions found in the security policy of a data warehouse.

- Physical security of the data stored in the data warehouse.
- Security at different levels of summarizations.
- Security of the OLAP tools.

- Web access security.
- Security of the workstation
- Security over network connections.
- Tables access privileges.
- User roles and privileges.
- Resolution of security breaches and security violations.

In a data warehouse environment, security 1s provided using three mechanisms: managing user privileges, through passwords, and using specific security tools. In this section, we will learn all three of them. User privileges for a data warehouse, the project team prefers a role based security. This feature is supported by most of the RDBMS today. A role is nothing but a grouping of users with common requirements for accessing the database. After creating the roles, users can be assigned appropriate roles. Access privileges may be granted at the level of a role or at the level of an individual. When this is done, all the n. Users assigned to that role will receive the same access privileges that are granted at the level of that role. For example, let us say the user Paul is an end-user. You have granted certain: privileges to the users under this role. All the privileges granted to the end-User will be availed by Paul too. However, if some extra privilege to access a dimension table is also given to Paul, then only he can access that one extra dimension table and rest of the end-users cannot. Given below Is a list of roles, responsibilities, and privileges that are applied in a data warehouse environment.



- End-users who execute queries and formulate reports against the data warehouse database are given access privileges only for certain tables.
- System analysts and power users who run ad hoc complex queries and themselves formulate their reports are given access privileges for all the tables.
- Support staff that help the users to run their queries and reports and resolve the problems encountered by the users are given access privileges for all the tables.
- Query tool specialists who are responsible for installing and troubleshooting end-user tools and OLAP tools are given access privileges for all the tables.
- Security administrators who are responsible for granting and revoking privileges and monitoring the data warehouse usage are given access rights for the entire data warehouse system including the database administration rights and access to all the tables stored in the data warehouse.

- System and network administrators who are responsible for installing and looking after the maintenance of operating systems and network are given access rights for the entire data warehouse system but not for database administration rights. No rights are given to access the stored in the data warehouse.
- Data warehouse administrators who install and maintain DBMD provide backup and recovery features are given access rights for the entire data warehouse system including the database administration rights and access to all the tables stored in the data warehouse.

Password protection

Security in a data warehouse using passwords follows the same old way; we do it in operational systems. Since the data ware se is updated only during the loading process, user passwords are relevant to the load jobs. Delete operation also is generally not used in a data warehouse environment. So, the main issue with passwords is to authorize the users for read only data access. Users need passwords to get into the data warehouse. It is the duty of the security administrator to set up acceptable patterns and the expiry period for the passwords. The security system of the data warehouse automatically expires the password on its expiry date. A user may change his/her initial password after receiving it from the security administrator. The users must make sure that the passwords that they use are arbitrary and not easily recognizable by anyone. Users should not use their name, their spouse's name, or their pet names as their passwords. Every user must follow the pattern set by the administrator. Generally, passwords include text and numeric data. The data warehouse security mechanism must make a record unauthorized attempt to gain access using invalid passwords so that after, a prescribed number of such attempts the user must be suspended temporarily from the data warehouse, that is, until the data warehouse administrator reinstates the user. Following a successful log-in, the number of attempts to deliberately access the system must be reported as it could mean that someone Is trying to work at a user workstation in the absence of the authorized user.

Security tools

In the data warehouse environment, the security provided by the chosen DBMS provides the primary security tool. One way of providing security by DBMS is role-based security that we have already discussed. Some organizations also have third-party security systems installed to govern the security of the data warehouse system. Using third-party tools brings the data warehouse under the larger security umbrella. However, some of the end-user tools like the OLAP tools come with their own security system. But the drawback here is that the tool-based security systems do not offer as much flexibility as the DBMS offers. If you have provided the security systems in the tool set, there is absolutely no need to repeat it at the DBMS level, but some data warehouse teams go for double protection by invoking the security features of the DBMS as well. Similarly, if you are planning to use the DBMS itself for security protection, then tool-based security may be considered redundant. To implement the security features, you may use security mechanisms at any of the three levels: third party tools, end-user tools, and DBMS.

Information Delivery Enhancements

With the passage of time, the users become more dependent on the data warehouse to get the desired information. They will become more efficient in finding the data themselves and using the data warehouse without any assistance from the IT. They will start posting even more complex queries. Even in the market, you will find that new information delivery tools keep coming, so the users must be provided with the latest tools available. But to do this, keep the following points in mind.

- Ensure compatibility of the new toolset with other components of the data warehouse.
- Ensure integration of the new tool set with the end-user metadata.
- Training on the new toolset must be scheduled.

If there are any data stores attached to the previous toolset used, then plan for migration of that data to the new toolset. Other Activities The other operational issues that can affect the operation of the system that need to be considered are as follows: Startup and shutdown of data warehouse applications, database, and data warehouse server. Problem management. All these operations may have a passive effect on the system and thus require proper management. All these tasks and services must be designed and developed by the data warehouse design team. Starting up and shutting down functions are executed for the data warehouse server, its database, and the applications. They are, however, critical tasks because shutting down the machine or database incorrectly can cause problems on restart. The data warehouse applications should be shut down gracefully, allowing them to complete whatever work they are currently doing, not just aborted. Similar is the case with the data warehouse database. Unless necessary, it must not be forced down because this will cause it to perform recovery operations on startup to clean up any jobs that were running when the database was shut down. The other area of concern in shutting down a database is getting all the users connections logged off as it is not uncommon for users to leave connections running and forget about them. Scripts would be needed to find these connecting processes and kill them. Problem management is an important area that needs to be clearly defined and documented. The administration staff must clearly know whom to approach, where to approach and when to approach if there are any problem areas in the system. It is not uncommon to have several groups inside an organization responsible for different parts of the data warehouse where one such go may be a central helpdesk for applications running in the system.

Overnight Processing: It is a key challenge that any data warehouse designer must face. There are certain major issues that if not addressed, will become a stumbling block to the success of the data warehouse. 1iedo in bottleneck is keeping not eat into to the time window so that the overnight process he the next business day. The serial nature of the tasks that must be performed and its sheer volume make this job even more difficult. The tasks that must be completed overnight are:

- Data rollup operations.
- Gathering the data.

- Data transformation functions.
- Data cleansing functions.
- Daily loading and data refreshing.
- Creation of indexes.
- Building of aggregate and summarized tables.
- Taking backups.
- Archival of old data.

In data roll-up, older data is rolled up into aggregated form to reduce the space needed to store the entire data. However, the first step in the overnight processing is that of collecting the data from the source systems. Any delay in the data transfer process from the source systems to the staging area causes a serious risk to the data warehouse. If a key piece of daily data does not make it to the staging area and finally to the data warehouse until the next day, then handling such an instance becomes a difficult task. Another critical issue that needs to be resolved is that if data is missing should the available data be made visible to the users? These questions are important as well as difficult to answer. One single answer cannot be applied to every situation, as the effect of missing data will vary from business to business and depend on the purpose of the data warehouse. Once the data has been acquired, the next step is data transformation and data cleanup. We already know that the data must be transformed and cleaned before being loaded in the data warehouse. All the tasks involved in data transformation and data cleanup must be completed during the overnight processing. The next steps, index build, and aggregation creation and maintenance will be accomplished once the data has arrived, and has been transformed, loaded, and cleaned. The last overnight operation that must be done at the end is the backup process. If data must be archived, then it would be archived as a part of the backup process. However, this process can become an overhead if the frequency of the data archiving is very high. For example, its data is loaded daily, but is archived monthly, then the whole month's data would have to be archived; this extra overhead may not be completed within the overnight window. It is therefore recommended to schedule archiving activities to run outside the overnight window and during periods of low user activity, maybe at the weekends. If data marts also exist in the organization, then even they may have to be refreshed on a regular basis and if the archiving and data mart refresh are part of the overnight process, they need to be very tightly controlled, and their effect on the capacity of the server and the network needs to be well understood. If allowed to take unlimited time, these tasks will rapidly become a resource bottleneck and will drive overnight processing into the business day.

Recipe for a successful data warehouse

The following guidelines are to be adhered to build and maintain a successful warehouse.

- Make a list of the sponsors for the data warehouse. It must contain many sponsors so that even if a single sponsor leaves, the system will not get orphaned.
- From the very first day itself, consider data warehousing to be a user/builder project.

- Clarify that maintaining data quality will be an ongoing joint user/ builder responsibility.
- Train the users step by step.
- Try to build a high-level data model in not more than three weeks.
- Monitor the accuracy of the data extracting, cleaning, and loading tools.
- Design the metadata in such a way that it is easy and efficient for the warehouse who will use it.
- Formulate a well-designed plan to test the integrity of the data in the data warehouse.
- Encourage the end-users to test the Before complex queries themselves.
- Before initiating the construction of the warehouse, learn from the experiences of the companies who have already gone for data warehousing.
- Be on the lookout for small but strategic projects.

Data warehouse pitfalls

The following list summarizes the Much time limitations of a warehouse.

- Much time is wasted in data extracting, cleaning, and loading of data.
- One thing that should always be taken as default is that the scope of the data warehouse will always go beyond expectations.
- There will be a number of problems that the team will have to face because of the disparate source systems that feed the data in the data warehouse.
- Often there will be a need to store data not being captured by the existing systems.
- In continuation with the previous problem, there would also be a ne validate the data that is currently not being validated by the transaction processing systems.
- Some operational systems that act as a source of data for the warehouse may not be capturing the data at the lowest level of detail.
- Despite the best efforts made by the project team to train the users, n users will never apply their training to solve their problems.
- Many a times, the time needed for data loading will over-run the amount of time in the available window.
- A data warehouse is a high-maintenance system. The data warehouse project team will fail if it concentrates more on resource and optimization neglect, data and customer management issues and an understanding of what adds value to the customer.

Meta Data – Introduction

Metadata defines the contents and location of the data (or data model) in the data warehouse, relationships between the operational database and the data warehouse and the business views of the data in the warehouse as accessible to the end-user tools. Metadata is searched by users to find the subject areas and the definitions of the data. For decision support, the pointers required to data warehouse are provided by the metadata. Therefore, it acts as logical link between the decision support system application and the data warehouse. Thus, any data warehouse design should assure

that there is a mechanism that populates and maintains the metadata repository and that all access paths to data warehouse have metadata as an entry point. In other words, there should be no direct access permitted to the data-warehouse data (especially updates) if it does not use metadata definitions to gain access. Metadata definition can be done by the user in any given data warehousing environment. The software environment as decided by the software tools used will provide a facility for metadata definition in a metadata repository.

Types of Metadata

† Operational Metadata

As you know, data for the data warehouse comes from several operational systems of the enterprise. These source systems contain different data structures. The data elements selected for the data warehouse have various field lengths and data types. In selecting data from the source systems for the data warehouse, you split records, combine parts of records from different source files, and deal with multiple coding schemes and field lengths. When you deliver information to the end-users, you must be able to tie that back to the original source data sets. Operational metadata contains all of this information about the operational data sources.

† Extraction and Transformation Metadata

Extraction and transformation metadata contains data about the extraction of data from the source systems, namely, the extraction frequencies, extraction methods, and business rules for the data extraction. Also, this category of metadata contains information about all the data transformations that take place in the data staging area.

† End-User Metadata

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

Meta Data - Data Management

Metadata management is the administration of data that describes other data. It involves establishing policies and processes that ensure information can be integrated, accessed, shared, linked, analyzed, and maintained to best effect across the organization. Metadata management is important because you can leverage metadata in understanding, aggregating, grouping, and sorting data for use. You can also trace back many data quality problems to metadata. Metadata is generated whenever data is created, acquired, added to, deleted from, or updated. For example, document metadata in Microsoft Word includes the file size, date of document creation, the name(s) of the author and most recent modifier, the dates of any changes and the total edit time. Further metadata can be added, including title, tags and comments. The goal of metadata management is to make it easier for a person or program to locate a specific data asset. This

requires designing a metadata repository, populating the repository, and making it easy to use information in the repository. Benefits of metadata management include:

- Consistency of definitions of metadata so that terminology variations don't cause data retrieval problems.
- Less redundancy of effort and greater consistency across multiple instances of data because data can be reused appropriately.
- Maintenance of information across the organization that is not dependent on a particular employee's knowledge.
- Greater efficiency, leading to faster product and project delivery.
- Increasing need for data governance, regulatory and compliance requirements, and data enablement
- Increasing importance of higher data quality and trusted analytics driving business value from data
- Growing complexity of data, with new sources augmenting the traditional sources
- More business users actively interact with data.
- Increasing need to accelerate transformation efforts, such as digitization, omnichannel enablement and data modernization.

When an organization is establishing policies to manage metadata, it is important for managers to gather together and agree upon a common data vocabulary and taxonomy. Intra-department variations should be addressed, and custom usages eliminated or replaced. In some cases, the organization may choose to use a metadata repository that comes with a toolset already in use. For instance, ETL vendors offer metadata management applications for cataloging and managing ETL metadata, as well as metadata associated with source and target applications.

Implementing metadata management

Implementing metadata management can be easy or complex depending on how you approach your use cases. Data governance and data analysis are identified as the most important use cases for metadata management solutions. Considering that metadata users and sources are very diverse, you will need to align metadata management with your data governance and data analysis strategies. Data governance is a critical enabler for creating and managing metadata. Data governance goes hand in hand with metadata management to ensure access to trusted data that is correctly understood throughout the lifecycle and used in the right context. Automation and self-service can work only when high quality trusted data is available with a shared understanding of metadata. The complexity of metadata management implementation varies according to the size and diversity of sources, use cases, users and their roles. It is also impacted by technology, which generates new sources and use cases and also creates opportunities to manage them better.

Metadata management works on three levels; it is important to focus on the linkage between all three.

- **Terms:** Common business language and definitions. Sources are industry standards, policy manuals, contracts, reference guides and handbooks.

- **Attributes:** Business resource-specific, such as system or reports. Sources include data dictionaries, system documentation, data models – enterprise, conceptual and logical.
- **Elements:** Data resource-specific, such as database tables or reports. Sources include database catalogs, spreadsheets and data models – physical.

Implementing metadata management requires that metadata is captured, stored and governed consistently at all three levels. Metadata linkage between levels and also with the top-level domains (such as customers, vendors or products) supports search, navigation and drill-down. Processes govern all the levels to change, review, validate and certify terms. You also need enterprise-level policy management to ensure quality metadata at all levels. Metadata management drives business value improves innovation and collaboration and helps mitigate risk. It also enables data citizens to access high-quality and trusted data, thus ensuring that they work with the right data to deliver accurate insights.

Meta Data - Query Generation

The most used query language for relational database is SQL, which allows retrieval and manipulation of the data stored in the tables, as well as the calculation of aggregate functions such as average, sum, min, max and count. For instance, an SQL query to select the videos grouped by category would be: SELECT count (*) FROM Items WHERE type=video GROUP BY category. The metadata repository should contain information such as that listed below:

- Description of the data model.
- Description of the layouts used in the database design.
- Definition of the primary system managing the data items.
- A map of the data from the system of record to the other locations in the data warehouse, including the descriptions of transformations and aggregations.
- Specific database design definitions.
- Data element definitions, including rules for derivations and summaries.

It is through metadata that a data warehouse becomes an effective tool for an overall enterprise. This repository of information will tell the story of the data: where it originated, how it has been transformed, where it went and how often – that is, its genealogy or artefacts. Technically, the metadata will also improve the maintainability and manageability of a warehouse by making impact analysis information and entity life histories available to the support staff. Equally important, metadata provides interactive access to users to help understand content and find data. Thus, there is a need to create a metadata interface for users. One important functional component of the metadata repository is the information directory. The content of the information directory is the metadata that helps users exploit the power of data warehousing. This directory helps integrate, maintain, and view the contents of the data warehousing system. From a technical requirements point of view, the information directory and the entire metadata repository should:

- Be a gateway to the data warehouse environment, and therefore, should be accessible from any platform via transparent and seamless connections.

- Support an easy distribution and replication of its content for high performance and availability. Be searchable by business-oriented keywords.
- Act as a launch platform for end-user data access and analysis tools.
- Support the sharing of information objects such as queries, reports, data collections and subscriptions between users.
- Support a variety of scheduling options for requests against the data warehouse, including on-demand, one-time, repetitive, event-driven and conditional delivery (in conjunction with the information delivery system).
- Support the distribution of query results to one or more destinations in any of the user-specified formats (in conjunction with the information delivery system).
- Support and provide interfaces to other applications such as e-mail, spreadsheet and schedules.
- Support end-user monitoring of the status of the data warehouse environment.

At a minimum, the information directory components should be accessible by any Web browser, and should run on all major platforms, including MS Windows, Windows NT and UNIX. Also, the data structures of the metadata repository should be supported on all major Relational database platforms. These requirements define a very sophisticated repository of metadata information. However, existing products often come up short when implementing these requirements.

Meta Data and Tools

Access tools

The principal purpose of data warehousing is to provide information to business users for strategic decision-making. These users interact with the data warehouse using front-end tools. Although ad hoc requests, regular reports and custom applications are the primary delivery vehicles for the analysis done in most data warehouses, many development efforts of data warehousing projects are focusing on exceptional reporting also known as alerts, which alert a user when a certain event has occurred. For example, if a data warehouse is designed to access the risk of currency trading, an alert can be activated when a certain currency rate drops below a predefined threshold. When an alert is well synchronised with the key objectives of the business, it can provide warehouse users with a tremendous advantage. The front-end user tools can be divided into five major groups:

Query and reporting tools

This category can be further divided into two groups: reporting tools and managed query tools. Reporting tools can be divided into production reporting tools and desktop report writers. Production reporting tools let companies generate regular operational reports or support high-volume batch jobs, such as calculating and printing pay cheques. Report writers, on the other hand, are affordable desktop tools designed for end-users. Managed query tools shield end-users from the complexities of SQL and database structures by inserting a metalayer between users and the database. The meta layer is software that provides subject-oriented views of a database and

supports point-and-click creation of SQL. Some of these tools proceed to format the retrieved data into easy-to-read reports, while others concentrate on on-screen presentations. These tools are the preferred choice of the users of business applications such as segment identification, demographic analysis, territory management and customer mailing lists. As the complexity of the questions grows, these tools may rapidly become inefficient.

Application development tools

Often, the analytical needs of the data warehouse user community exceed the built-in capabilities of query and reporting tools. Organisations will often rely on a true and proven approach of in-house application development, using graphical data access environments designed primarily for client-server environments. Some of these application development platforms integrate well with popular OLAP tools, and can access all major database systems, including Oracle and IBM Informix.

Executive information systems (EIS) tools

The target users of EIS tools are senior management of a company. The tools are used to transform information and present that information to users in a meaningful and usable manner. They support advanced analytical techniques and free-form data exploration, allowing users to easily transform data into information. EIS tools tend to give their users a high-level summarisation of key performance measures to support decision-making.

Online analytical processing (OLAP) tools.

These tools are based on concepts of multidimensional database and allow a sophisticated user to analyse the data using elaborate, multidimensional and complex views. Typical business applications for these tools include product performance and profitability, effectiveness of a sales program or a marketing campaign, sales forecasting and capacity planning. These tools assume that the data is organised in a multidimensional model, which is supported by a special multidimensional database or by a Relational database designed to enable multidimensional properties.

Data mining tools

Data mining can be defined as the process of discovering meaningful new correlation, patterns and trends by digging (mining) large amounts of data stored in a warehouse, using artificial intelligence (AI) and/or statistical/mathematical techniques. The major attraction of data mining is its ability to build predictive rather than retrospective models. Using data mining to build predictive models for decision-making has several benefits. First, the model should be able to explain why a particular decision was made. Second, adjusting a model on the basis of feedback from future decisions will lead to experience accumulation and true organisational learning. Finally, a predictive model can be used to automate a decision step in a larger process. For example, using a

model to instantly predict whether a customer will default on credit card payments will allow automatic adjustment of credit limits rather than depending on expensive staff making inconsistent decisions.