Chapter #2:-

Data Warehouse System LifeCycle

Q. Risk factors Associated with Data Warehouse System Cycle:

⇒ Potential challenges or threats that can hinder the success, implementation, operation or existence of the data warehouse, called risk factors.

⇒ The major risk factors and problem causes for failure of data warehouse projects are as follows:

Risks can be divided into four categories:

1. Risk related to project management

- The project scope may expand over time, leading to increased costs and delays.
- Lack of executive-support, may cause resource constraints and delays.
- Shedule slippage - Delays in project milestones
- Sponsoring
- Goals of data warehousing systems
- Resource issues

2. Risks related to technology:

- Poor scalability of architectures in terms of data volumes and number of users.
- Lack of expandability to implement new technological solutions, components and applications
- Insufficient skills of implementors about data warehouse specific software tools
- Inefficient management of meta-data exchange between every component.

⇒ Because Technological solutions used for designing, implementing, accessing and maintaining data warehouse systems are rapidly changes. The architecture should be able to keep up with new standards.

3. Risk related to data and design:

⇒ These risk factors depend on the quality of the data supplied and project carried out

- Achieving low quality results this happen due to source instability and unreliability. Can also because of user did not properly specify their requirements

- Inability to provide users with high added value when delivering initial prototypes
- ETL process failures or errors
- Reporting or analytical tool limitations
- Data model incompatibility
- Data redundancy
- Inaccurate or incomplete data
- Inefficient data structures or query optimization can impact query performance

4. Risk related to Organization:

⇒ This kind of failure is due to:

- Anability to involve end users, to interest them in the current project or to make them support it
- Lack of training
- Resistance to change
- Might not be clear rules for how to manage the data warehouse
- Users inability to take advantage of the results achieved because of this consolidated organization's practices

Q. Methodological Approaches to Built Data WareHouse:-

⇒ There are Two basic structures to built data warehouses and these structures deeply affect the data warehouse lifecycle.

1. Top-down Approach
2. Bottom-up Approach

Top-down Structure

⇒ The top-down approach starts with a comprehensive data model that defines the entire scope of the data warehouse. Individual data marts are derived from this central model.

⇒ Need to analyze global business needs, plan how to develop a data warehouse, design it, and implement it as a whole.

⇒ **Key Characteristics:**

Consistency:

This approach provides centralized control over data definitions and standards ensuring consistency across the entire.

data warehouse, bcz all data marts are sourced from a common data warehouse.

Enterprise-Wide View:

- Focuses on creating a unified, integrated, enterprise wide view of data.

Reduced data redundancy:

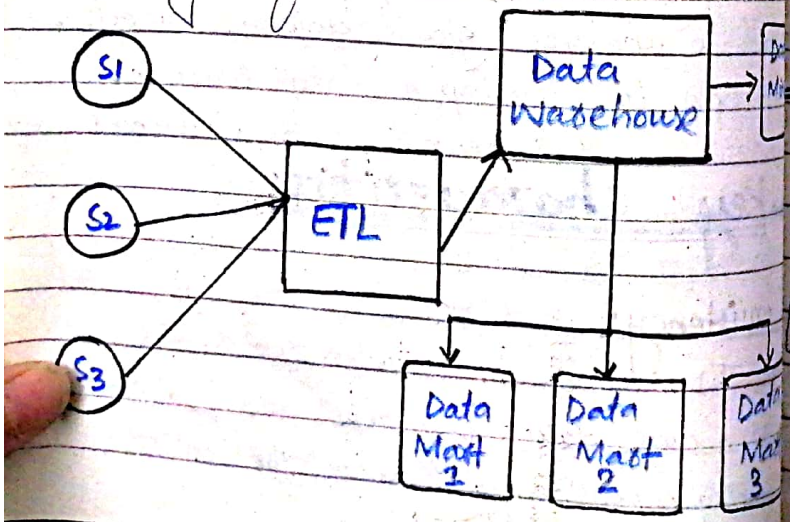
- By defining central model, redundant data can be minimized.

Maintenance is easier:

- Easier to maintain and update data by making changes to the data warehouse and this change will propagate automatically to all the data marts that relay on it.

Scalability:

- Highly scalable, bcz add a new data marts is easy without disturbing the existing infrastructure.



Suitability: suitable for large, complex organizations with diverse data sources and a need for standardized data definitions.

Disadvantages:

⇒ High-cost estimates with long-term implementations

⇒ Analyzing and bringing together all relevant sources is a very difficult task.

⇒ Difficult to forecast the specific needs of every department ↑ limited users involvement

⇒ Since no prototype is going to be delivered in the short-term, users cannot check for this project to be useful, so they lose trust and interest in it.

⇒ Lack of flexibility

⇒ Complexity is high

Bottom-UP Structure

⇒ In this approach, starts with the creation of individual data marts, each focused on specific business area or department. These data marts are then integrated to form a larger, enterprise-wide data warehouse.

Key Characteristics

Incremental and iterative:

⇒ Data warehouses are incrementally built and several data marts are iteratively created. Each data mart is based on set of facts that are linked to a specific company department

Reduced Risk:

⇒ By starting with smaller, more manageable data marts, the risk of project failure is reduced. Data marts tested, before incorporate to Data WH.

Kinball approach:

⇒ Given by Kinball, thats why also called Kinball approach.

Users involvement:

⇒ By providing early data marts or prototypes.

Flexibility:

⇒ Offers greater flexibility to adapt to changing business needs and prioritise.

Departmental ownership:

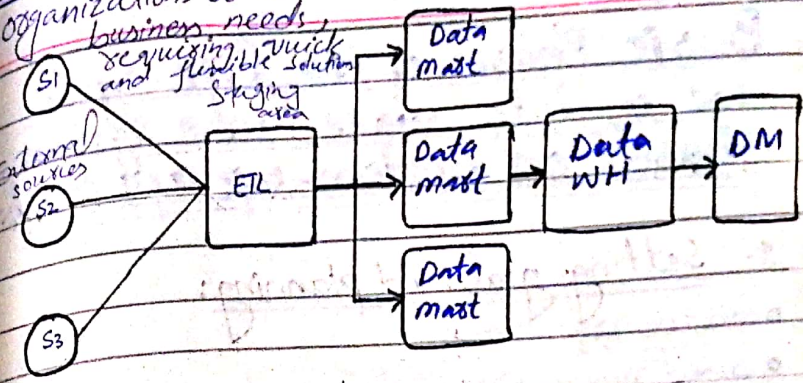
⇒ Each department can take ownership of their respective data mart, fostering buy in and engagement.

Disadvantages:

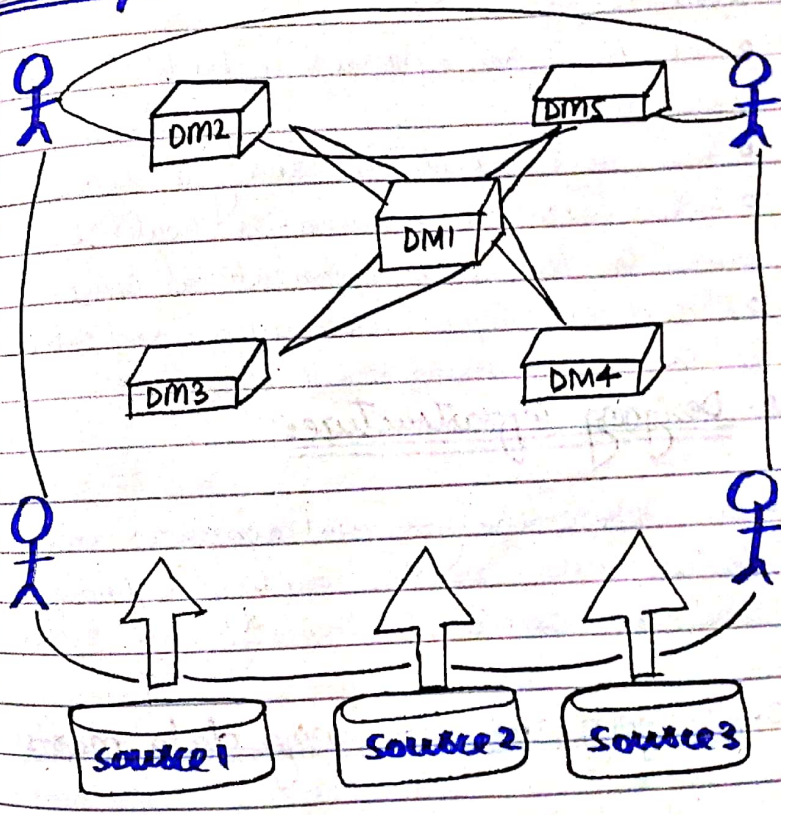
- Lack of enterprise-wide view
- Risk of inconsistency
- Integration challenges

Suitability:

ideal for small organizations or those with rapidly changing business needs, requiring quick and flexible solutions



Example:



Basic Phases of the life-cycle for a data ware-house system based on bottom-up approach

1. Setting goals and planning:

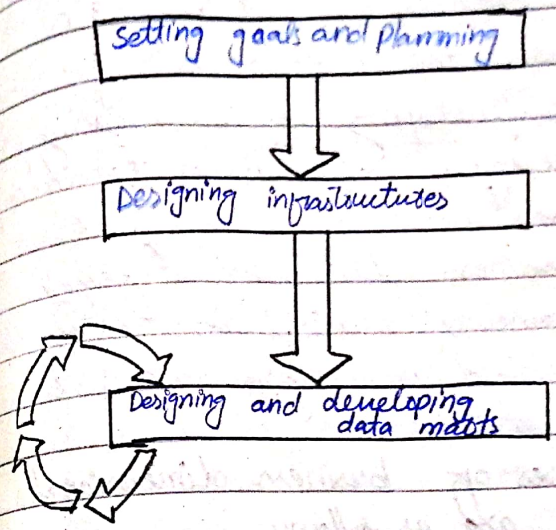
- Preliminary phase.
- Based on a feasibility study
- Setting system goals, properties and size estimates
- Selecting an approach to build the data warehouse
- Risk and expectation analysis done
- Team competence examination should be done to deal with organizational issues.
- Also define implementation plan and submit to the top-management.

2. Designing infrastructure:

- This Phase analyze and compares architecture and assesses existing technology and tools to create a preliminary plan of the whole system.

3. Designing and developing data marts:

Every iteration causes a new data mart and new applications to be created and progressively added to the data warehouse system.



⇒ Two methodology based on this framework are:

- Business Dimensional Lifecycle
- Rapid warehousing methodology

⇒ Both comes under bottom-up approach.

Business Dimensional Lifecycle

⇒ Business dimensional lifecycle stands for the time needed for designing, developing and implementing data warehouse systems as reported by Kimball.

⇒ The business dimensional lifecycle is a framework that outlines the key stages involved in managing data within a data warehouse.

The Phases of business dimensional lifecycle are as follows:

1. Project Planning:

⇒ It includes:

- Definition of system goals and properties.
- An assessment of the impact on organizational practice.
- Allocation of required resources
- Estimation of cost and benefits
- Preliminary plan for the project that will be carried out.

2. Business Requirement definition:

⇒ It helps to ensure that designers properly and fully understand users needs. To maximize the benefits and the profitability of the system under-construction.

⇒ Also, should detect key factors for making decisions and turn them into design specification.

⇒ The definition of requirements consists of the following three parallel tracks, that also consists of different phases:

- Data
- Technology
- Application.

⇒ The Phases of data track are as follows:

Dimensional modeling:

⇒ The first phase of data track.

- User requirements and analysis of operational sources lead to the definition of data structures in a data warehouse.
- Specific type of data model that organizes data into facts and dimensions.
- Dimensions provide context, while facts represent measureable quantities.
- final result is a set of logical

schemata and a set of relationships with source schemata.

Physical design:

⇒ Translate the dimensional model into detailed representation that can be implemented in a database system.

⇒ Logical schemata is optimized and implemented into the selected DBMS, such as indexing and partitioning.

Data staging design and implementation:

⇒ It includes:

- All the issues linked with data extraction, transformation, loading and quality.

⇒ The Technology track Phases are as follows:

Architecture design:

⇒ Based on:

- Performance requirements that users wants
- Current technical specifications for business information systems.

Product selection and installation:

⇒ It involves:

- Study and assess usable hardware platforms, DBMS, extraction, transformation and loading tools and data analysis tools available in market.

⇒ The application track composed of the following:

Users application specification

⇒ Collect the specification for the applications that will provide end users with data access

- Assessment the needs for reports to be created
- interactive navigation of data
- Automatic knowledge extraction.

Users development Phase:

⇒ The tools selected in product selection phase should be set up and configured.

3- Deployment Phase:

⇒ it involves all the tracks and lead to system setup.

⇒ Handover to the users.

4- Maintenance:

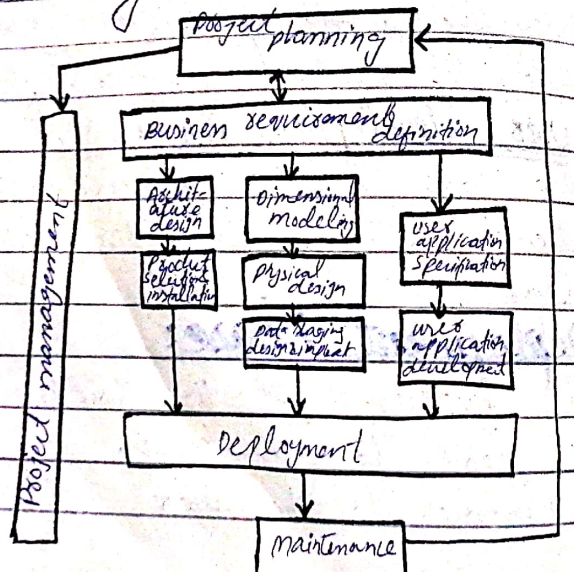
- System need continuous maintenance to provide users with support and training. Monitor usage - identify areas of improvement.
- Performing regular updates, refresh data and expire the data to meets the changing business needs.

5- Project management:

⇒ Should be accurate in every data warehouse lifecycle system.

⇒ Project management allows:

- To keep tasks in sync
- To monitor current project work
- Check that design team is closely collaborating with users.



Rapid Warehousing Methodology

⇒ Iterative and evolutionary approach to managing data warehousing projects.

⇒ This approach created by SAS institute, A leader in the statistical analysis industry, divides potentially large projects into smaller, much less risky projects called builds.

⇒ Each build takes advantage of the data warehouse environments developed during previous builds.

⇒ it expands them to add new features and evolves them.

⇒ Each build can makes previous data warehouse environments keep on meeting user's ever-changing needs.

⇒ This approach ensures that users will still be interested and involved in projects. and it lays the foundation for successful long-term projects.

⇒ Phases of this approach are as follows:

1. **Assessment:**
- Corresponds to Kimball planning Phase
- Purpose is at ascertaining whether an enterprise is ready to undertake a data warehousing project
- Setting goals
- Risks and benefits

2. **Requirements:**
- includes end users application specification
- Enables analysis, project and architecture specifications for the system.

3 **Design:**
- This phase focuses on one project build at a time.
- Analysis specifications are refined to generate logical and physical data design and data-staging design.
- implementation tools also selected at this stage.
- Detailed architecture, data models & schema is developed.
- Define data sources, transformation & loading processes.

4 **Construction:**
- Data warehouse infrastructure is built or implemented and populated with the data extracted from data sources.
- Front-end applications are developed.

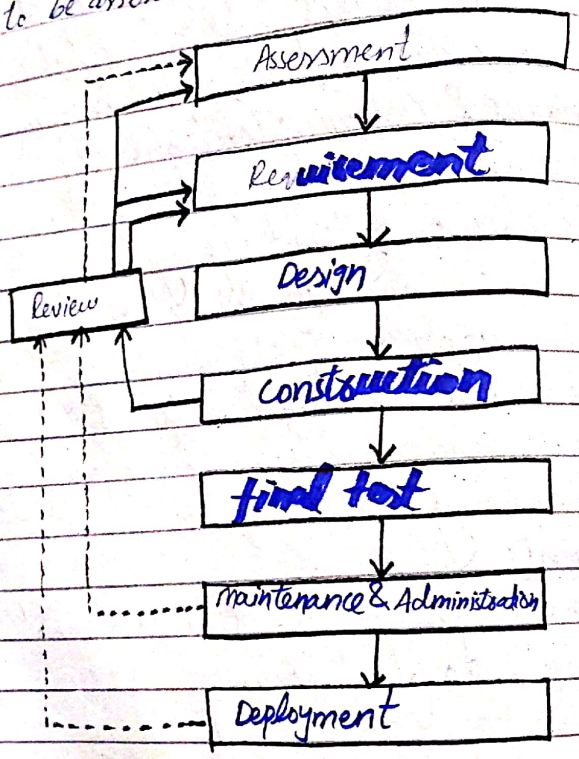
5 **Final Test:**
- At this phase, unit testing, integration testing, data quality checks and user acceptance testing is performed.

7 **Maintenance & Administration:**
- This phase is as long as the whole data warehouse system lifecycle. maintenance and administration can implement additional features, upgrade data warehouse architecture to meet new needs and check for data quality. Refine existing data.

6 **Deployment:**
- The system is delivered and started up after training its end users.
- Move the data warehouse to production

8 **Review:**
Each build includes three review Processes:

1. Implementation check
2. Post-deployment check (make sure that organization is properly prepared to take advantage of data warehouse)
3. Final check for costs and benefits to be assessed.



Q. Data Mart Design Phases:

⇒ These are seven Phases to design the data marts:

⇒ Order of these Phases depends on design approach.

1: Analysis and Reconciliation of data Sources:

⇒ Reconciled sheema is define and document in this phase.

Reconciled Schema:

⇒ The schema of the reconciled data layer used to feed data mart.

⇒ This phase ensures that the data being integrated into data mart is accurate, consistent and relevant to the business needs.

⇒ During this Phase:

- Data from various sources is examined, Profiled by understanding the structure, content and quality of the data from each source.
- map data elements from various source system to the target structure in the data mart.
- Address and resolve discrepancies blw data sources.

- Identify and manage duplicates data that might exist across sources.
- Ensure that data conforms to the business rules and logic defined during requirement analysis phase.
- Select which group of data can be useful for the decision making processes in the specific business area to which the data mart is devoted.
- Analyze and understand available source schemata
- if multiple sources are to be used they should homogenize and integrate their schemata to determine common features and remove every inconsistency

⇒ This phase involves:

- Data designers
- Data processing center staff.

Are the only ones who can assign meaning to schemata and records that are often hardly to understandable.

→ Their familiarity with the fields of application is essential for schema normalization

Importance:

- Ensures data integrity
- Facilitates trust in data
- Optimize performance

2: Requirement Analysis:

- Designers collect, filter and document end user requirements in the requirement analysis phase to select relevant information to be represented in compliance with the strategic goals.
- After this stage, specifications for the facts are obtained that should be modeled, as well as preliminary guidelines for the work-load.

Fact:

- Concepts of primary importance in decision making process. Facts are quantitative data.
- Phenomenon that that dynamically occurs in an enterprise:
- Occurrence of each fact is called an event.
- End users are responsible for the selection of facts.
- Designers help end users to do this based on the documentation collected in analysis and source reconciliation phase.
- Each fact requires historical interval

- Preliminary workload is expressed as "Pseudo-natural language.
- Workload should specify the measures and aggregations for every fact.
- Through this designer have opportunity to define dimensions and measures for the conceptual design.
- Granularity decide the level of detail for each fact.
- Granularity defines how flexible data mart queries are.
- Granularity:

is the result of compromise between the system feedback speed and the maximum level of query detail.

3: Conceptual design:

⇒ At this phase users requirements are exploited to create conceptual schema for a data mart on the basis of reconciled schema.

⇒ Also known as fact dimensional model

⇒ High level architecture and structure of the data mart is define.

⇒ This phase translates the business requirements and preliminary analysis from the requirements analysis phase into a conceptual model that outlines how the data mart will be organized and how the data will flow through it.

- A fact schema created for every fact that can graphically represent all multidimensional model concepts such as facts, measures, dimensions and hierarchies.
- Key Activities at this Phase:

Define the fact tables:

- identify central fact tables that will store the quantitative data.
- Define the grain of the fact table, which determines the level of details.
- Core of the data mart, where primary data for analysis is reside.

Design the dimension Tables:

- Identify the dimensions that will be used to analyze the facts.
- Define the attribute within each dimension table
- Ensure that dimensions provide the necessary context for the facts & support the required business queries.

Model Relationships between fact and dimensions:

- Establish relationship using star or snowflake schema between fact and dimensional table.
- Main aim is to create a model that allows users to easily drill down into data and perform complex analysis.

High-level architecture design:

- Define how data will be moved, ETL processes, storage and access layers.

DFD:

- Define visualization of how data will move from source systems through ETL process to the data mart.

4: Workload Refinement and validation of conceptual schemata:

- At this phase, workload that was expressed in preliminary Phase should be refine.
- For doing this, queries should be clear represent in the conceptual schema.
- This allows to check for all the expected queries to be actually feasible and leads to the validation of conceptual obtained from previous phase.

5: Logical Design:

- This phase focuses on creating logical model that outlines how data will be structured, processed, and stored, aligning with the requirements and technical constraints.
- Basically, the selection of a logical model to act as a reference framework. opt for ROLAP or MOLAP.
- Conceptual model is transformed into a detailed technical specification that defines the structure and relationships of the data mart entities and attributes.
- it involves the following key activities:
  - Define data mart entities and attributes.
  - Normalize the data model.
- The technique that mostly affects the performance called view materialization.

6: Physical Design:

- The logical model is translated into a concrete, technical implementation. This Phase involves making decisions about how the data will be stored, accessed and managed within the database system.

Select Specific DBMS on which data

mast will be implemented.
⇒ Workload and data volume also play role at physical design phase.
⇒ Key activities involves at this stage.
- Database selection
- storage & indexing
- Partitioning
- Denormalization
- Data types and constraints
- Performance tuning
- Security & Access control
- Backup and recovery

7: Data-Staging design:

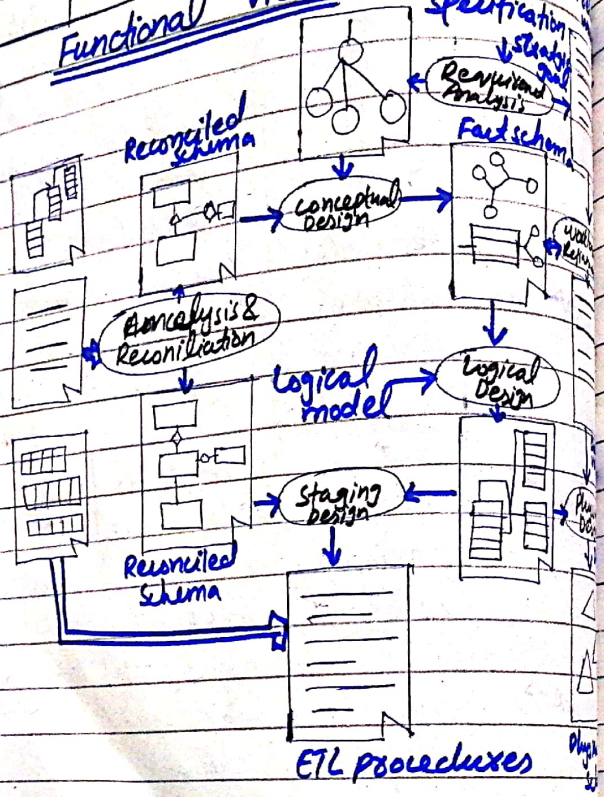
- Designers, end-users and database administrators collaborate and make all the significant decisions on the population process of the reconciled layer.
- Planning and preparing the infrastructure to receive, transform and store incoming data before it's loaded into the data warehouse. Ensures data quality & consistency.
- Key Activities
  - Data source identification - Data loading strategy
  - Extraction strategy - Error handling
  - Data transformation - Performance optimization

Overview of Phases:

Phases	i/p	o/p	People involved
Phase-1	Operational Source Schemata	Reconciled schema	Designers, data processing centre staff
Phase-2	Strategic goals	Requirement specifications, workload	Designers, end users
Phase-3	Reconciled schema, Requirement Specification	Fact schemata	Designers, end-users
Phase-4	Fact schemata, Preliminary workload	work load, data volume, validated fact schemata	designers, end-users
Phase-5	Fact schemata, target logical model, workload.	Logical data mart schema	Designers
Phase-6	Source Schemata, logical data mart schema	ETL Procedures	Designers, DB administrators

Phase-1	Logical data mart schema, target DBMS workload	Physical data mart schema	Designed

Functional View Requirement Specification



Methodological Framework

⇒ A methodological framework is a structured approach that outlines the steps, techniques, and procedures to be followed in conducting research or analysis.

⇒ The approaches to designing data marts divided into Two categories:

- Data-driven approach / supply-driven
- Requirement driven / demand-driven

Data-driven / Supply-Driven Approach:

⇒ Design data marts on the basis of close operational data source analysis.

⇒ Users-requirements show designess which groups of data, relevant for decision-making processes, should be selected and how to define data structures on the basis of multidimensional model.

⇒ In this approach:

Data sources are significant for modeling data marts.

Due to this, Data source analysis and reconciliation phase is the first Phase

⇒ Requirements are analyzed and a conceptual schema is created for data mart without losing the actual features of the original data. By doing so, a set of specific data obtained for data

⇒ In data driven approach, users requirements phase comes before the conceptual design phase.

⇒ Activities performed in the requirement analysis phase, such as specifying the dimensions are completed in the conceptual design phase when they triggers operations that specify the structures of conceptual schemata.

⇒ In this methodology, it is possible and suitable to do requirement analysis at the same time as conceptual design is opting. Initial conceptual schema is extracted from reconciled schema.

⇒ At the end of the conceptual design phase is the workload refinement Phase followed by logical design Phase.

⇒ Data staging design & physical design may possible occurs at a time.

Prerequisites for successful adoption of the data-driven approach:

- An in-depth knowledge of data sources to populate data marts should be available and it should be achieved at a reasonable price and in the short term.
- Data source schemata should show good level of normalization.
- Data source schemata should not be too complex.

⇒ if architecture has a reconciled layer, Normalization & in-depth knowledge obtained at the end of the reconciliation process.

⇒ if reduced data sources into single one, small well designed database, same results can be obtained after ingestion phase.

Advantages:

- ETL design is extremely streamlined (efficient ↑ Less complicated) bcz every single information piece is directly stored in data marts and associated with more than one attributes.

- Time required for data-driven conceptual design is proportional to the complexity of data sources bcz initial conceptual schema is derived from reconciled layer that strictly depends on the data source structures.
- Reach to the project goal within short time.
- Data marts are quite stable in time because they are rooted from source schemata that less frequently change rather than requirements expressed by end users.

Cons:

- Provides limited support to designer when facts, dimensions and measures need to be determined.
- if not rely on in-depth knowledge of operational sources in the conceptual design phase, then data mart are difficult to implement.
- Minor role of user requirements

+ Diagram same like on previous page.

2: Requirement-Driven Approach:

⇒ A requirement-driven approach to data mart design is a strategic methodology that ensures the data mart aligns with the specific needs of its intended users.

⇒ This approach involves a series of steps to gather, analyze and prioritize requirements, ultimately leading to the creation of a data mart that delivers value.

⇒ The requirements expressed by end-users are the driving force of the design phase in requirement-driven approach.

⇒ Formalism (structure, formal or technical aspects are emphasize) used for requirement specifications.

⇒ Then the collected requirements should be turned into a conceptual schema.

⇒ Conceptual design phase is critical. because cannot take advantages of the detailed knowledge of operational data to extract structural relationships among the bits of information to be represented in data mart.

⇒ After completing the conceptual design phase, the subsequent phase of same as data-driven approach.

⇒ However, data-staging design causes a heavy burden because of poor source analysis and need to manually extract the relationships among data and operational data sources.

⇒ Also known as goal-oriented approach based on the decision-information model.

⇒ Requirements driven approach will best if requirement be iteratively and incrementally collected on the basis of use cases.

Disadvantages:

- Time-expensive bcz users do not have a clear and shared understanding of business goals and processes.
- Strong leadership need for designers and meeting also needs for grab the different point-of-views.

To sum up:

- This approach begin with the definition of information requirements of data marts users.
- Problem is how to map the requirements into existing data sources is addressed at a later stage when appropriate ETL procedures are implemented.

3: Mixed approach:

- In this approach, requirement and source analysis are conducted at the same time.
- Requirement analysis specifies project requirements.
- Source analysis results in a reconciled layer drawing.
- Conceptual design Phase is carried out semi-automatically and requirements basically reduced complexity.
- Subsequent phases remains un-changed.

Recommend approach if the reconciled layer extension and complexity are remarkable.

- Low-level of complexity for data staging design.

Testing data marts within a data warehouse system is important for ensuring accuracy, integrity and reliability of data. It involves several steps.

Testing Data Marts:

⇒ The test phase purpose is checking for data marts are properly created and for user requirements to be fulfilled.

⇒ Testing data mart involves checking:
- Functionality
- Performance
- Robustness
- Usability

⇒ The test phase is part of the data warehouse lifecycle.

⇒ At beginning of project the test phase should be plan and arrange and determine which type of tests should perform, which data set needs to be tested.

⇒ Data ware house systems are particularly suitable for tests to be carried out in a modular fashion.

⇒ Specifically, a test process do separately for back end components:
- mainly ETL processes

& front end components such as:
- OLAP
- Reporting
- Distributed environments

⇒ The test phase can take place before the project end to avoid delays and deadlines that can force testers to perform their tasks within time.

⇒ Thoroughly check the system is very difficult and requires time and cost.

⇒ Sample tests are very suitable for testing the data warehouse system.

alpha test
- conducted by system developers.

beta test
- conducted by end users.

⇒ To test the data marts, test phase should include different types of tests:

Unit test:
- Test individual components of data marts such as ETL processes, data cleansing rules and data quality checks.
- Verify that each component is working correctly and producing expected results.

Integration test:
- Checks the general and integrated operation of all the system components.
- Test how the data mart interacts with other components of the data warehouse

including the source systems, data warehouse and reporting tools.
- Ensure that data flows smoothly between these components and that there are no conflicts or inconsistencies.

Architecture test:

- Checks for the system implementation.
- it involves evaluating the overall design and structure of the data mart to ensure it meets the specification of the architectural schema.
- it involves focuses on:
  - Data flow & ETL processes
  - Validate data model & schema
  - Assess the suitability of the storage infrastructure
- This helps to find any problems before the data mart is built and used.

Usability test:

- To evaluate how easy it is for users to interact (understand & learn) with the data mart's interface & retrieve the desired information.
- Users observations, surveys are done & focus on user friendliness, intuitiveness & and clarity of the interface.

Safety test:

- Checks for the proper operation of hardware and software machines used to implement a safe system.
- To ensure that the data mart complies with relevant safety regulations and standards especially if handles the critical data.

Fault tolerance testing:

- Checks how robust the system actually is.
- The test simulates error in one or more components and evaluate the system response to errors, understand and overload operating conditions.

⇒ for ex:

- Cut off power supply when ETL process is in progress.
- Set a database offline while an OLAP session is in progress

⇒ To evaluate the data mart's ability to continue functioning in the event of one or more failure.

Error simulation test:

- Checks for that main application error conditions are properly managed
- ETL flow test are particularly important because they must check for procedures to be able to manage incoming and incomplete data.
- This testing need white box - testing techniques.
- Technique involves injecting errors into the system, monitoring responses and assessing the recovery mechanisms.

Performance (workload test):

- Checks that software meets the efficiency requirements.
- To measure the data mart's performance under various load conditions including peak usage and stress scenarios.
- it focuses on response times, query execution times, scalability etc.
- ETL components checks for operational data processing time
- for OLAP components, submits a group of concurrent queries to a system & checks for the time needed to process the queries

Regression test:

- Critical quality assurance process that ensures that changes made to the data mart do not negatively impact existing functionality.
- it involves re-executing test cases to verify that previously working features continue to function as expected

Chapter #3

Analysis & Reconciliation of Data Sources

Local source schema:

⇒ Local source schema refers to the conceptual schema that describes the data stored in a source without regard to the models and technological solutions implemented.

⇒ focus on the representation of the application domain.

Local source schemata result from the analysis of relations and constraints in relational database.

Definition:

- Refers to the processes used to evaluate, compare and ensure consistency, accuracy, and integration of data from multiple sources before consolidating them in a unified system, such as data warehouse.

(i) Data source analysis:

Data sources can show strong relationships or can be completely independ

Data mart designer's primary goal is to assume his or her knowledge of data sources the source analysis phase.

key question involves:

- What are the data types, formats and relationships in the data source?
- Are there inconsistencies or missing data that need to be addressed?
- Are there duplicates records or conflicting information across sources?
- How does data from one source relate to data from another source?

Reconciliation process:

⇒ Reconciliation process is needed in order to obtain consistent, error free information.

⇒ Reconciliation process involves:

- integrating
- cleansing
- transforming

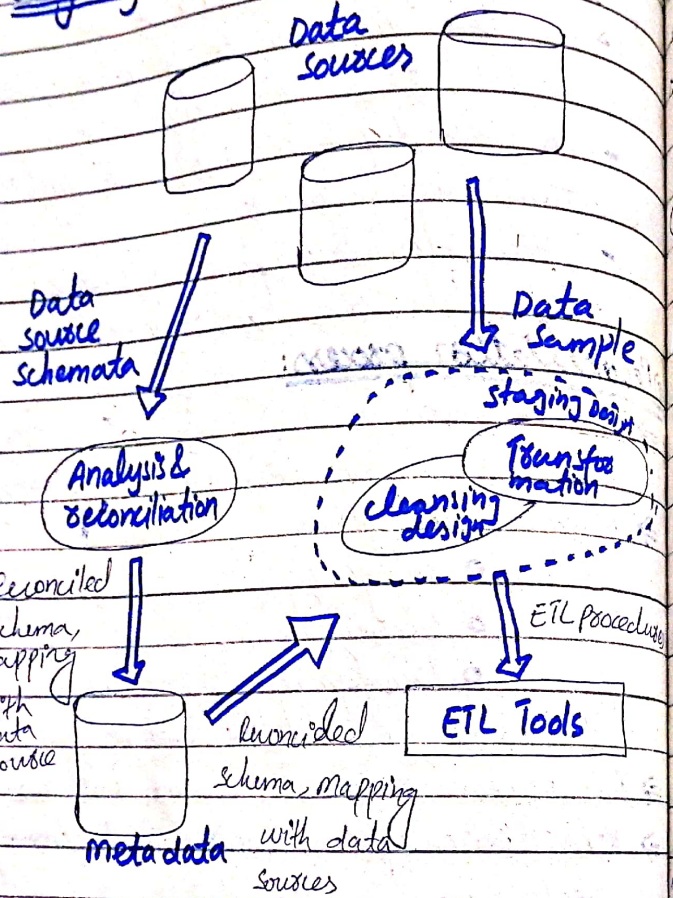
data to create a reconciled database.

⇒ This process requires time and resources.

⇒ Reconciliation process ensures that ensuring that data from different sources matches and correctly integrated.

⇒ Integration phase is based on the intensional features of data sources.

Three-Phase Reconciled layer design from data source layer :



Short,

Reconciliation and analysis phase do the following:

- Inspect and normalize all the local schemata to generate a set of comprehensive, locally consistent, conceptual schemata.
- Then do integration, resulting in a globally consistent conceptual schema.

Design steps forming the analysis & reconciliation phase of two data sources