VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELGAVI-590018, KARNATAKA

DATA WAREHOUSING

(AS PER CBCS SCHEME 2022)

SUB CODE: BAD515B

PREPARED BY:

INDHUMATHI R (ASST.PROF DEPT OF DS (CSE), KNSIT)

DEPARTMENT OF COMPUTER SCIENCE (DATA SCIENCE) AND ENGINEERING

K.N.S INSTITUTE OF TECHNOLOGY

HEGDE-NAGAR, KOGILU ROAD,
THIRUMENAHALLI, YELAHANKA,
BANGALORE-560064

MODULE 2

CHAPTER 4: PLANNING AND PROJECT MANAGEMENT

PLANNING YOUR DATA WAREHOUSE:

- **Determine the Need**:
  - First, assess whether a data warehouse is truly necessary for the company. This involves evaluating the value it will bring, the kind of data needed, and its use cases.

- **Key Issues**:

- Understanding Business Requirements: Identify the needs of the organization and how the data warehouse will support strategic decision-making
- Value and Expectations: Companies often dive into data warehousing without a clear understanding of its benefits. Assess the data's value and determine whether a warehouse is the right solution for your business goals.

  Example: A retail company like Amazon would consider the value of collecting transaction data from customers across the world and whether the data warehouse would help improve customer segmentation or predict purchasing trends.
- Risk Assessment: Just like any other project, there's a risk of failure in a data warehouse project. You need to assess what could go wrong—whether it's misalignment with business needs, insufficient budget, or technology challenges.

  Example: If Netflix's data warehouse project fails, it could lead to slower recommendations or outages in service, severely impacting user experience and the company's reputation.
- Top-Down or Bottom-Up Approach: You can either plan for an enterprise-wide data warehouse from the top down (for a broader and centralized strategy) or build individual departmental data marts first using the bottom-up approach (focusing on smaller units first). Example: An international bank might start with a top-down approach to ensure all financial operations follow the same system-wide data structure, whereas a retail chain might take a bottom-up approach, focusing on different regions first.

- Build or Buy: Decide whether you want to build the data warehouse in-house or buy pre-built solutions. Building it allows for customization, but buying can be faster. Example: A company like Netflix might build its data warehouse to handle its complex data and large volume, whereas a small business might buy a pre-built solution from a vendor to get started quickly.
- Single Vendor or Best-of-Breed: You can go with a single vendor for simplicity and integration, or you can mix and match tools from different vendors for specialized solutions (best-of-breed). Example: A healthcare provider may choose a single vendor to handle all their data because of tight security and compliance regulations, whereas a larger tech company might choose the best tools from different vendors to create a tailored solution for various departments.

➢ **Business Requirements, Not Technology**:

- Focus on Business Needs: The planning process should prioritize the requirements of the business rather than the technology itself.
- User Involvement: Engage end-users in the planning process to ensure that the data warehouse meets their needs and expectations.
- Focus on the business problems you are solving, not on the specific tools or technology. Understand what your end-users need and base the design on solving their issues, not on using the latest technology.
- Example: If a hotel chain like Marriott is trying to streamline bookings and customer feedback data, they should focus on how to make that information easy for their employees and customers to access, rather than getting too caught up in which technology stack is the most innovative.

➢ **Preliminary Survey**

- Conduct a survey of user needs to get a broad understanding of the business and define the scope of the project.

➢ **Top Management Support**

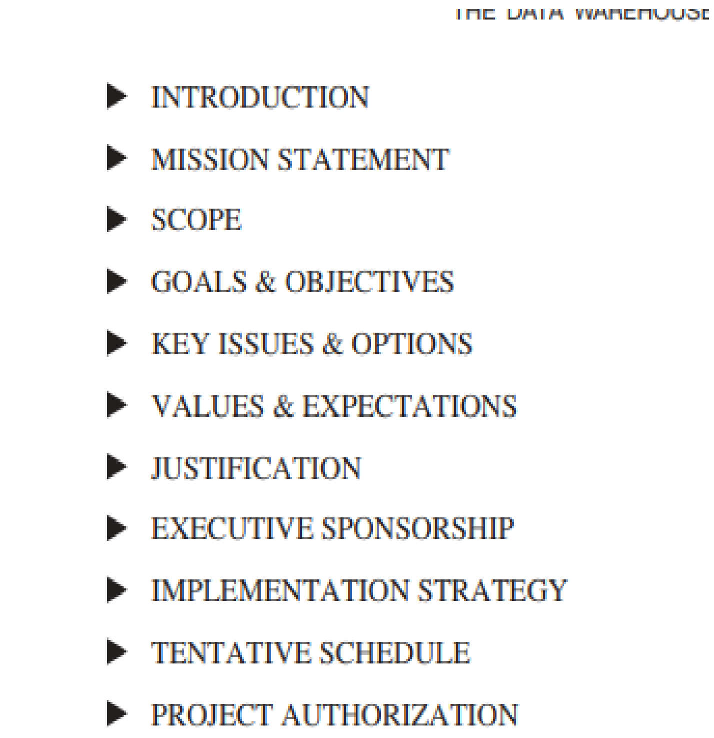
- Secure support from senior management to ensure the project's success. Having a top-level sponsor will help in resolving conflicts and ensuring that the project stays on track.

Justification

- A data warehouse can be a significant investment. The project needs to be justified either through formal analysis (cost-benefit ratio, ROI) or based on intuition and competitive pressures.

The Overall Plan:

- Define Scope: Clearly outline the objectives, deliverables, and boundaries of the data warehouse project.
- Develop a Timeline: Create a project timeline that includes key milestones and deadlines for each phase of development.



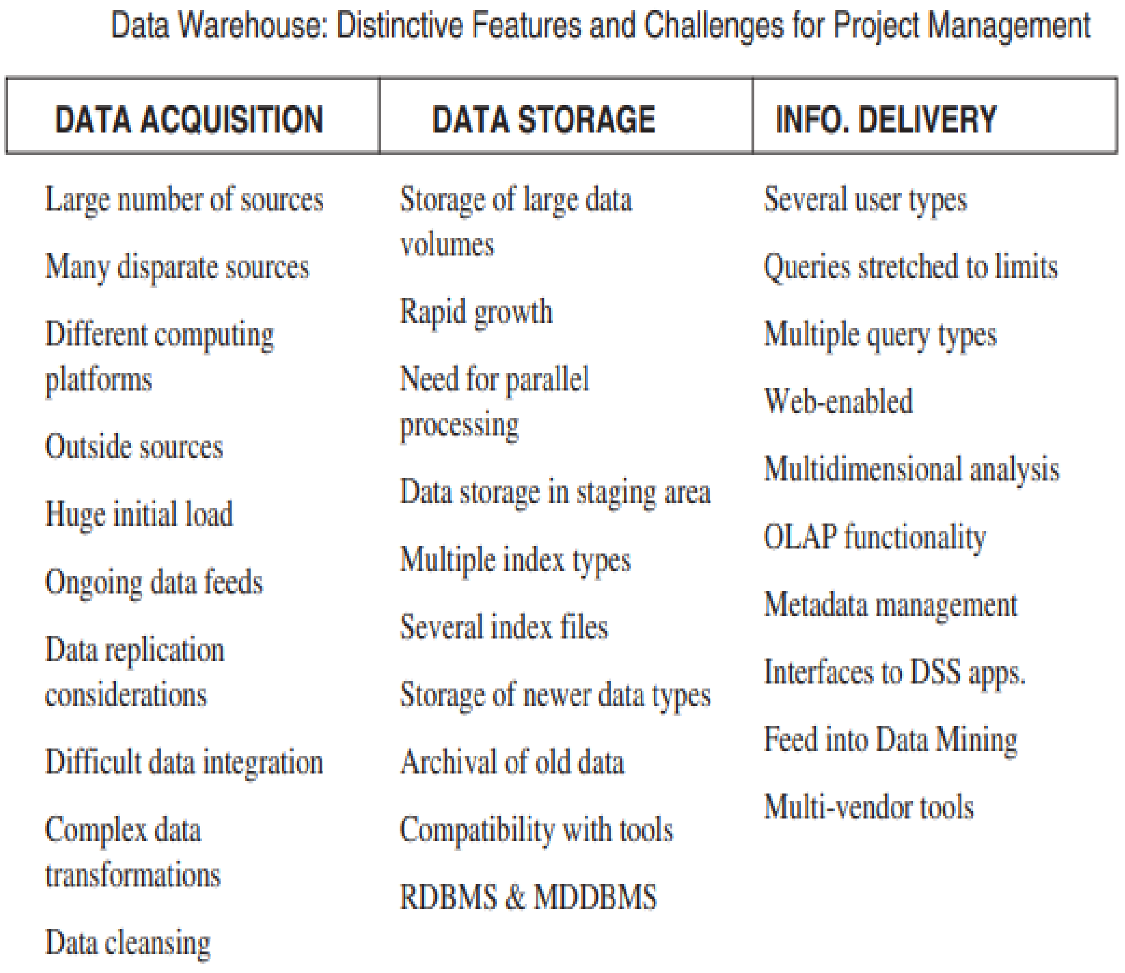
THE DATA WAREHOUSE

- ▶ INTRODUCTION
- ▶ MISSION STATEMENT
- ▶ SCOPE
- ▶ GOALS & OBJECTIVES
- ▶ KEY ISSUES & OPTIONS
- ▶ VALUES & EXPECTATIONS
- ▶ JUSTIFICATION
- ▶ EXECUTIVE SPONSORSHIP
- ▶ IMPLEMENTATION STRATEGY
- ▶ TENTATIVE SCHEDULE
- ▶ PROJECT AUTHORIZATION

Figure 4-1 Overall plan for data warehousing initiative.

THE DATA WAREHOUSE PROJECT

- How is it Different?
  - Unique Characteristics: Data warehouse projects differ from traditional IT projects in terms of scope, complexity, and the need for cross-functional collaboration.
  - Longer Duration: Data warehouse projects typically take longer to implement due to the need for extensive data integration and transformation.



Data Warehouse: Distinctive Features and Challenges for Project Management

DATA ACQUISITION	DATA STORAGE	INFO. DELIVERY
Large number of sources	Storage of large data volumes	Several user types
Many disparate sources	Rapid growth	Queries stretched to limits
Different computing platforms	Need for parallel processing	Multiple query types
Outside sources	Data storage in staging area	Web-enabled
Huge initial load	Multiple index types	Multidimensional analysis
Ongoing data feeds	Several index files	OLAP functionality
Data replication considerations	Storage of newer data types	Metadata management
Difficult data integration	Archival of old data	Interfaces to DSS apps.
Complex data transformations	Compatibility with tools	Feed into Data Mining
Data cleansing	RDBMS & MDDBMS	Multi-vendor tools

Figure 4-2 Differences between a data warehouse project and one on OLTP application.

- Assessment of Readiness:
  - Evaluate Current Infrastructure: Assess the existing IT infrastructure and data sources to determine readiness for a data warehouse implementation.
  - Identify Skill Gaps: Identify any skills or knowledge gaps within the project team that may need to be addressed.
- The Life-Cycle Approach:
  - Phases of Development: Implement a structured approach to development that includes planning, design, implementation, and maintenance.
  - Iterative Development: Consider using an iterative approach to allow for continuous feedback and improvements throughout the project.

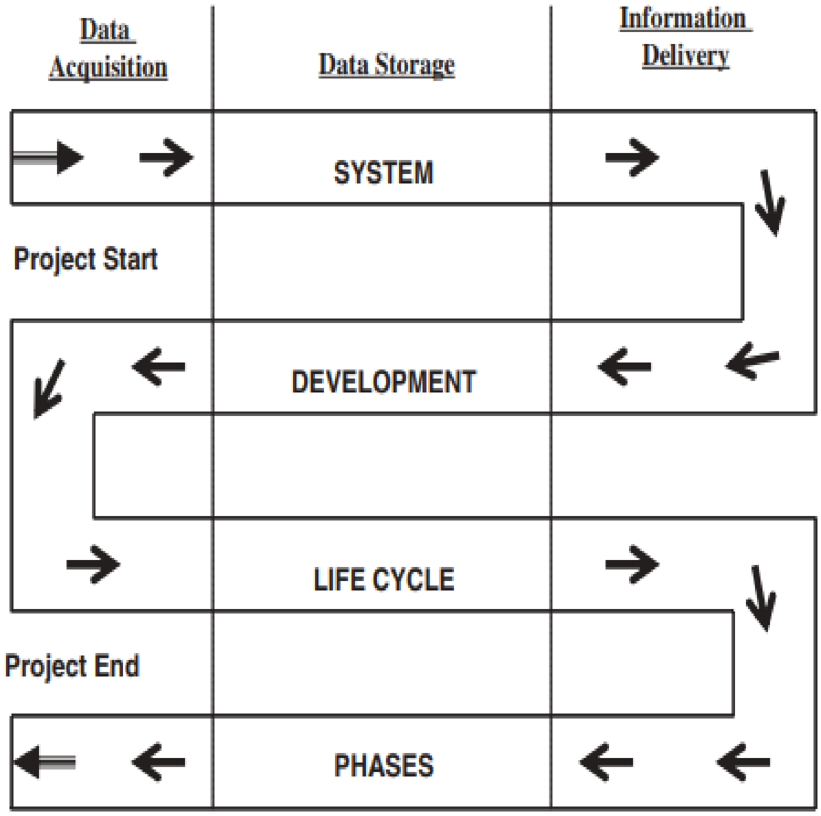


Figure 4-3 Date warehouse functional components and SDLC.

THE DEVELOPMENT PHASES

- Adopting Agile Development:
  - Agile Methodology: Consider using agile development practices to allow for flexibility and adaptability throughout the project.
  - Continuous Improvement: Emphasize continuous improvement and iterative feedback to refine the data warehouse as it is being built.

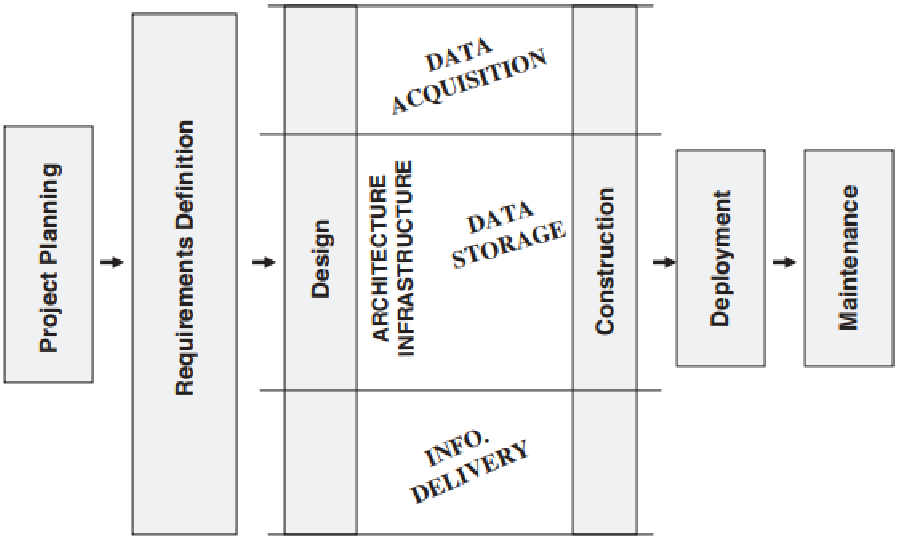


Figure 4-5 Data warehouse development phases.

Data Acquisition:

- This involves collecting and gathering data from various sources.
- The data can come from different platforms or even external sources.
- **Example (Social Media)**: Social media platforms like Facebook or Instagram acquire data from users—posts, likes, comments, and even third-party apps that integrate with these platforms.

Data Storage:

- After the data is collected, it is stored in a structured format, often in databases. Storage must handle huge amounts of data and allow efficient retrieval.
- **Example (Streaming Platforms)**: Streaming platforms like Netflix store huge volumes of user data—what shows users watch, for how long, and on which device. This data needs to be stored and indexed for quick access when needed (e.g., to recommend a show).

Information Delivery:

- This is about how the stored data is accessed and used by end-users or systems. The data may be used for reporting, analytics, or decision-making.
- **Example (Social Media)**: When you see personalized ads or friend suggestions on Facebook, this is an example of information delivery. The platform uses the stored data (from previous user activity) to deliver information that is meaningful to the user.

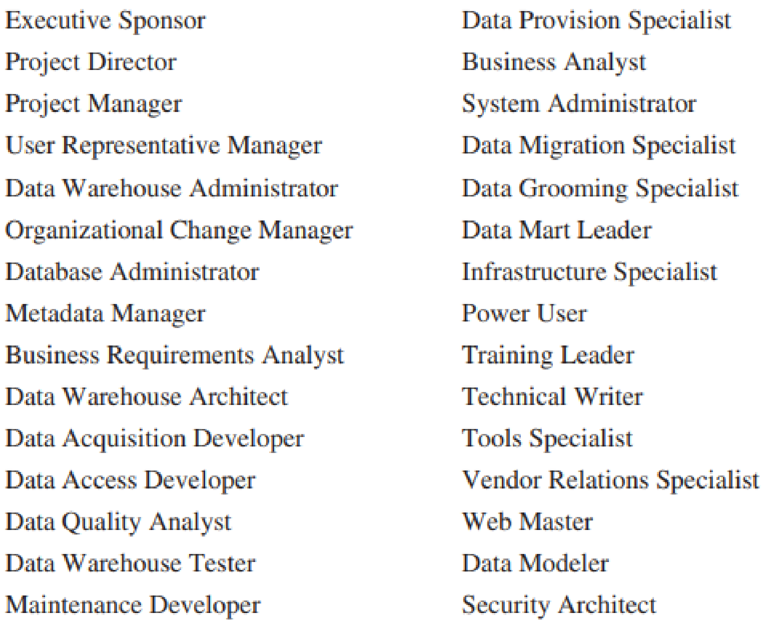
Development Phases in a Data Warehouse Project:

1. **Project Planning**:
   - This is the first phase where the overall plan is created, including setting objectives and timelines.
   - **Example**: When Twitter plans to roll out a new feature, they first outline how they will acquire, store, and deliver data related to this feature.
2. **Requirements Definition**:
   - It involves determining what the system needs to do, based on input from various stakeholders.
   - **Example**: Spotify determining that they need to store user preferences to recommend new songs is part of this phase.
3. **Design**:
   - This phase is about architecting how the data will flow through the system, how it will be stored, and how it will be accessed.
   - **Example**: YouTube designing its data pipeline for handling video uploads and user interaction data.
4. **Construction**:
   - The actual building of the data warehouse—creating databases, setting up the infrastructure, and coding the processes that will move the data.
   - **Example**: Netflix engineers setting up databases and processing pipelines to handle the huge amount of data that comes in from user activity.
5. **Deployment**:
   - The data warehouse is put into operation, and users start interacting with it.
   - **Example**: Instagram launching a new analytics dashboard where influencers can see the engagement data of their posts in real time.

6. **Maintenance**:
   - The ongoing upkeep to ensure that the system runs smoothly, including fixing bugs, scaling up storage, and ensuring data accuracy.
   - **Example**: TikTok continuously maintaining its recommendation algorithms to ensure users are shown the most engaging content.

THE PROJECT TEAM

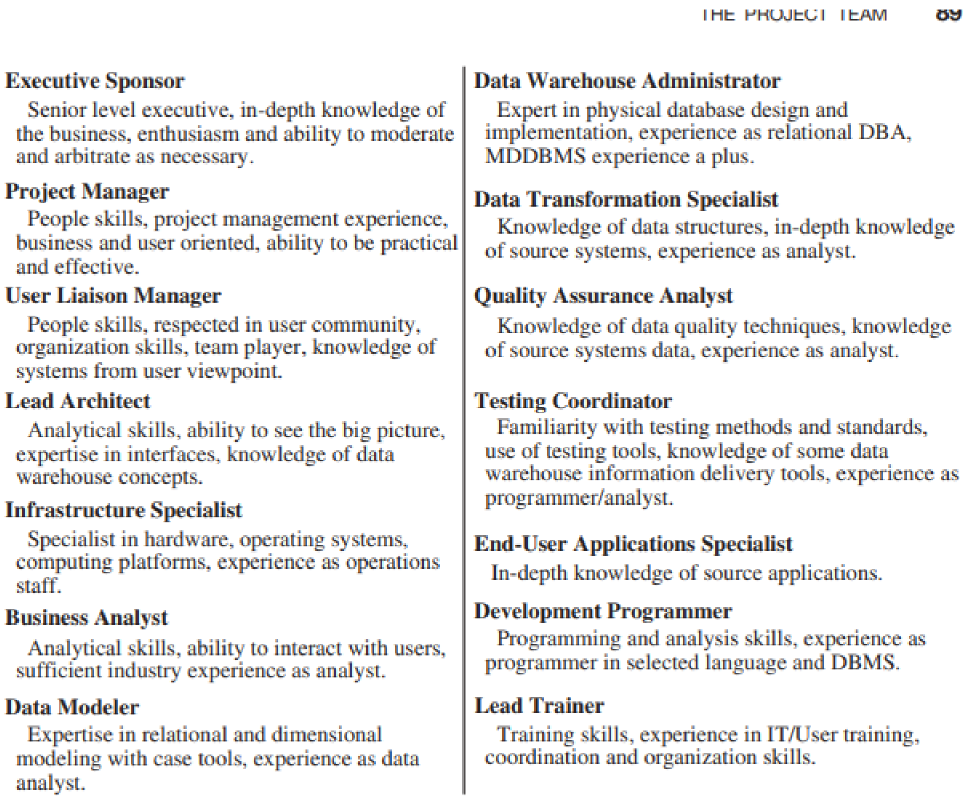
- **Organizing the Project Team**:
  - Cross-Functional Teams: Assemble a project team that includes members from different departments, such as IT, business units, and data governance.
  - Roles and Responsibilities: Clearly define roles and responsibilities for each team member to ensure accountability and effective collaboration.



Executive Sponsor
Project Director
Project Manager
User Representative Manager
Data Warehouse Administrator
Organizational Change Manager
Database Administrator
Metadata Manager
Business Requirements Analyst
Data Warehouse Architect
Data Acquisition Developer
Data Access Developer
Data Quality Analyst
Data Warehouse Tester
Maintenance Developer
Data Provision Specialist
Business Analyst
System Administrator
Data Migration Specialist
Data Grooming Specialist
Data Mart Leader
Infrastructure Specialist
Power User
Training Leader
Technical Writer
Tools Specialist
Vendor Relations Specialist
Web Master
Data Modeler
Security Architect

Figure 4-6 Job titles in a data warehouse project.

- Skills and Experience Levels:
  - Required Skills: Identify the skills and experience needed for the project, including data modeling, ETL (Extract, Transform, Load) processes, and business analysis.
  - Training and Development: Provide training and development opportunities for team members to enhance their skills and knowledge.



THE PROJECT TEAM 89

Executive Sponsor
Senior level executive, in-depth knowledge of the business, enthusiasm and ability to moderate and arbitrate as necessary.

Project Manager
People skills, project management experience, business and user oriented, ability to be practical and effective.

User Liaison Manager
People skills, respected in user community, organization skills, team player, knowledge of systems from user viewpoint.

Lead Architect
Analytical skills, ability to see the big picture, expertise in interfaces, knowledge of data warehouse concepts.

Infrastructure Specialist
Specialist in hardware, operating systems, computing platforms, experience as operations staff.

Business Analyst
Analytical skills, ability to interact with users, sufficient industry experience as analyst.

Data Modeler
Expertise in relational and dimensional modeling with case tools, experience as data analyst.

Data Warehouse Administrator
Expert in physical database design and implementation, experience as relational DBA, MDDBMS experience a plus.

Data Transformation Specialist
Knowledge of data structures, in-depth knowledge of source systems, experience as analyst.

Quality Assurance Analyst
Knowledge of data quality techniques, knowledge of source systems data, experience as analyst.

Testing Coordinator
Familiarity with testing methods and standards, use of testing tools, knowledge of some data warehouse information delivery tools, experience as programmer/analyst.

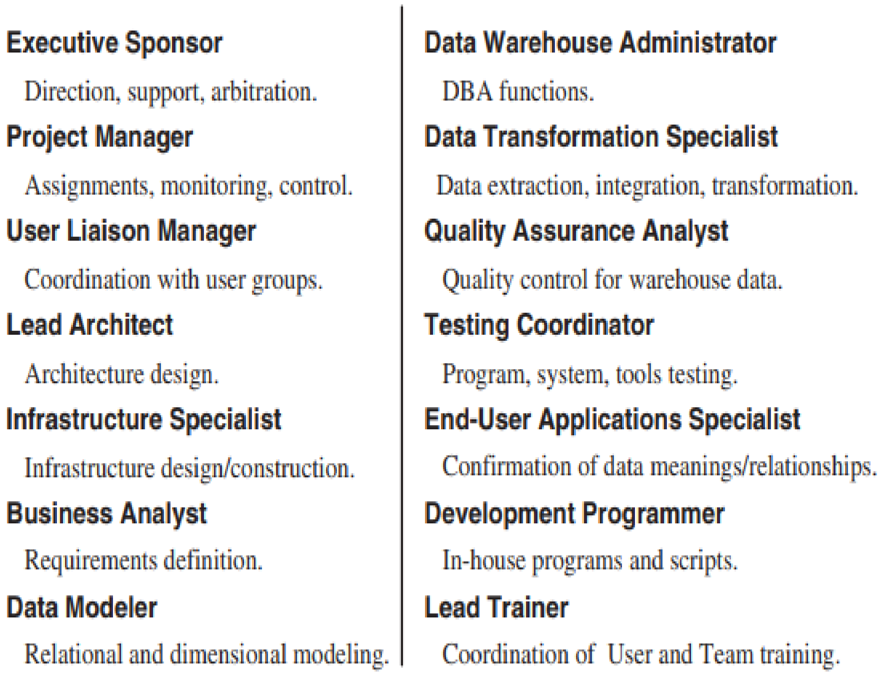
End-User Applications Specialist
In-depth knowledge of source applications.

Development Programmer
Programming and analysis skills, experience as programmer in selected language and DBMS.

Lead Trainer
Training skills, experience in IT/User training, coordination and organization skills.

Figure 4-8 Skills and experience levels required for a data warehouse project team.

- **User Participation**:
  - Engagement of End-Users: Involve end-users throughout the project to ensure that the data warehouse meets their needs and expectations.
  - Feedback Mechanisms: Establish feedback mechanisms to gather input from users during the development process.



Executive Sponsor

Direction, support, arbitration.

Project Manager

Assignments, monitoring, control.

User Liaison Manager

Coordination with user groups.

Lead Architect

Architecture design.

Infrastructure Specialist

Infrastructure design/construction.

Business Analyst

Requirements definition.

Data Modeler

Relational and dimensional modeling.

Data Warehouse Administrator

DBA functions.

Data Transformation Specialist

Data extraction, integration, transformation.

Quality Assurance Analyst

Quality control for warehouse data.

Testing Coordinator

Program, system, tools testing.

End-User Applications Specialist

Confirmation of data meanings/relationships.

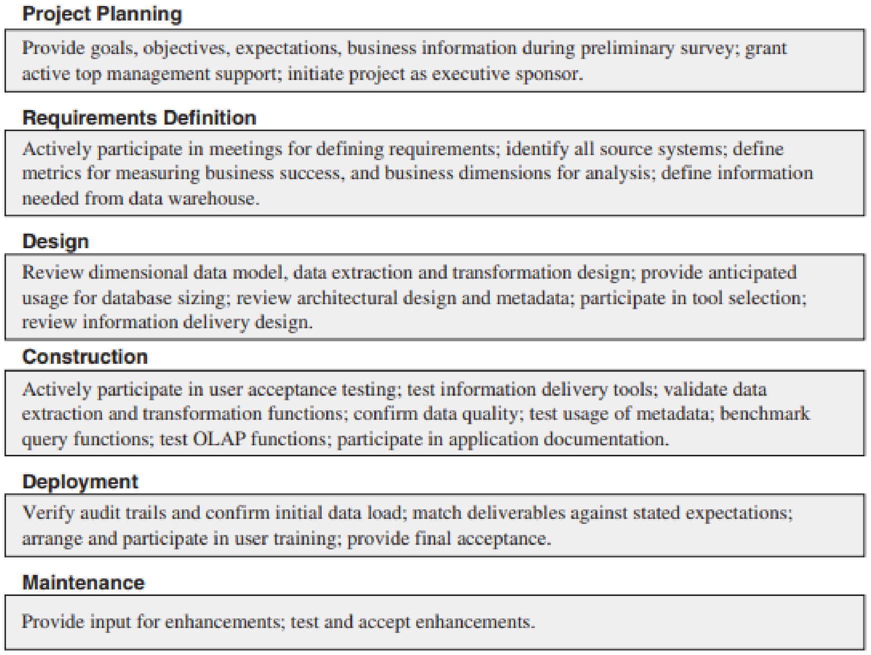
Development Programmer

In-house programs and scripts.

Lead Trainer

Coordination of User and Team training.

Figure 4-7 Roles and responsibilities of a data warehouse project team.



Project Planning

Provide goals, objectives, expectations, business information during preliminary survey; grant active top management support; initiate project as executive sponsor.

Requirements Definition

Actively participate in meetings for defining requirements; identify all source systems; define metrics for measuring business success, and business dimensions for analysis; define information needed from data warehouse.

Design

Review dimensional data model, data extraction and transformation design; provide anticipated usage for database sizing; review architectural design and metadata; participate in tool selection; review information delivery design.

Construction

Actively participate in user acceptance testing; test information delivery tools; validate data extraction and transformation functions; confirm data quality; test usage of metadata; benchmark query functions; test OLAP functions; participate in application documentation.

Deployment

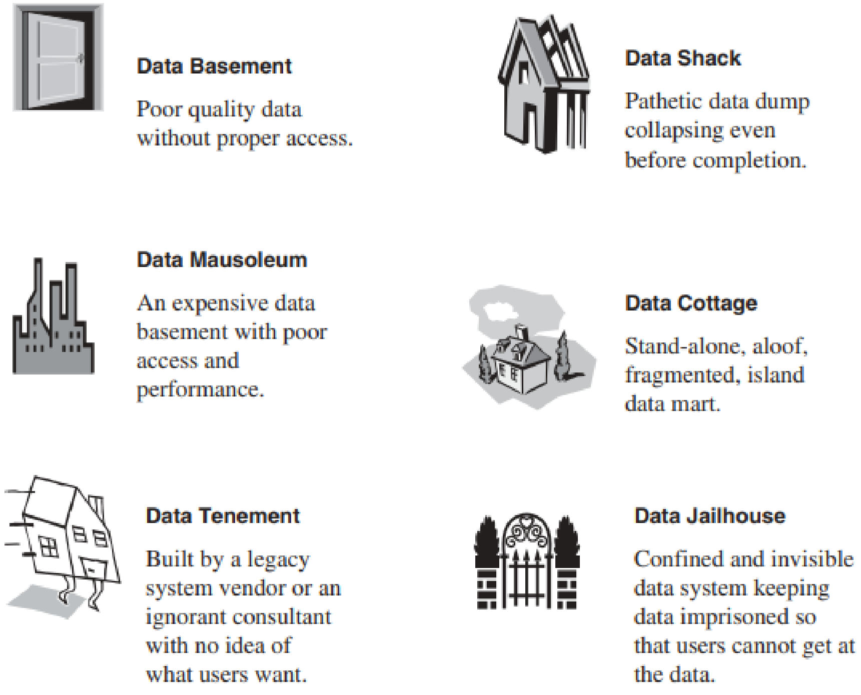
Verify audit trails and confirm initial data load; match deliverables against stated expectations; arrange and participate in user training; provide final acceptance.

Maintenance

Provide input for enhancements; test and accept enhancements.

Figure 4-9 User participation in data warehouse development.

PROJECT MANAGEMENT CONSIDERATIONS



Data Basement

Poor quality data without proper access.

Data Shack

Pathetic data dump collapsing even before completion.

Data Mausoleum

An expensive data basement with poor access and performance.

Data Cottage

Stand-alone, aloof, fragmented, island data mart.

Data Tenement

Built by a legacy system vendor or an ignorant consultant with no idea of what users want.

Data Jailhouse

Confined and invisible data system keeping data imprisoned so that users cannot get at the data.

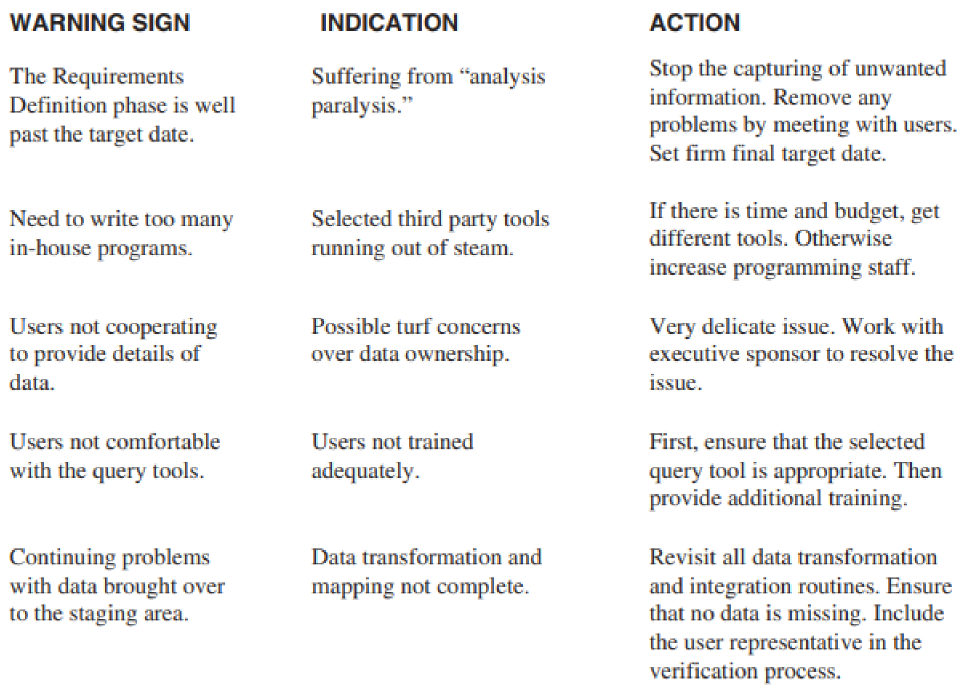
Figure 4-10 Possible scenarios of failure.

- **Guiding Principles**:

- Clear Objectives: Set clear objectives and success criteria for the project to guide decision-making and project direction.
- Effective Communication: Maintain open lines of communication among project team members and stakeholders to ensure alignment and transparency.

1. **Sponsorship:** A data warehouse project needs strong executive support to succeed.
2. **Project Manager Orientation:** A project manager should focus on user and business needs, not just technology.
3. **Data Quality:** The quality of data is crucial, focusing on accuracy, consistency, and reliability.
4. **Building for Growth:** The data warehouse should be built with future growth in mind.
5. **Dimensional Data Modeling:** A data model that supports easy querying and reporting is essential.
6. **Training Users:** Users should know how to query and use the data warehouse tools effectively

- **Warning Signs**:
- Identify Risks Early: Monitor the project for potential risks and issues that may arise, and address them proactively.
- Recognize Scope Creep: Be vigilant against scope creep, where additional features or requirements are added without proper evaluation.



WARNING SIGN	INDICATION	ACTION
The Requirements Definition phase is well past the target date.	Suffering from "analysis paralysis."	Stop the capturing of unwanted information. Remove any problems by meeting with users. Set firm final target date.
Need to write too many in-house programs.	Selected third party tools running out of steam.	If there is time and budget, get different tools. Otherwise increase programming staff.
Users not cooperating to provide details of data.	Possible turf concerns over data ownership.	Very delicate issue. Work with executive sponsor to resolve the issue.
Users not comfortable with the query tools.	Users not trained adequately.	First, ensure that the selected query tool is appropriate. Then provide additional training.
Continuing problems with data brought over to the staging area.	Data transformation and mapping not complete.	Revisit all data transformation and integration routines. Ensure that no data is missing. Include the user representative in the verification process.

Figure 4-11 Warning signs for a data warehouse project.

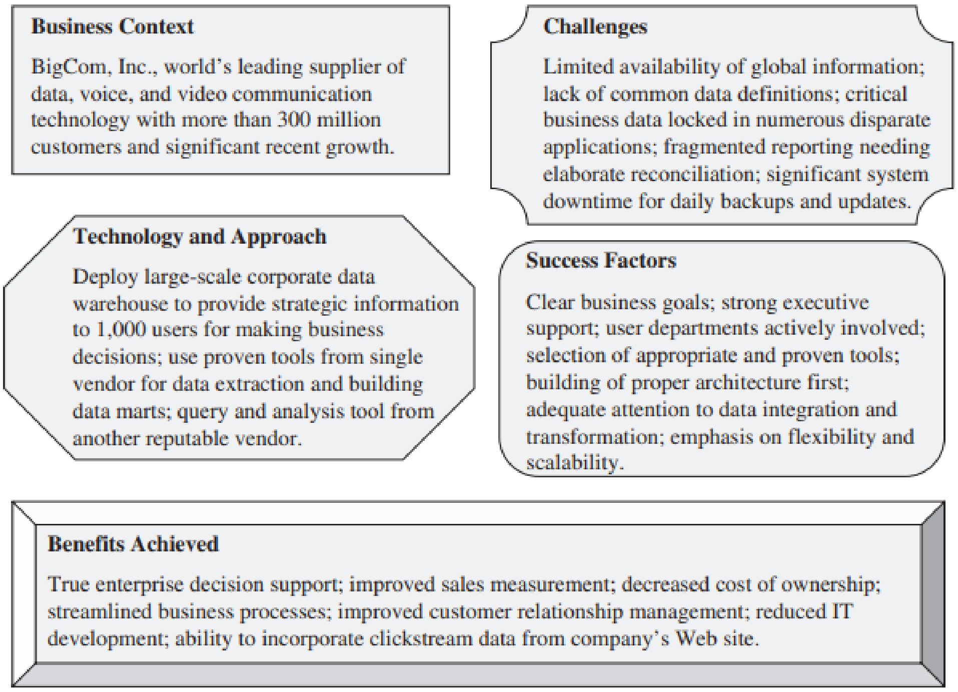
- **Success Factors**:
- Strong Leadership: Ensure that the project has strong leadership to guide the team and make critical decisions.
- Stakeholder Buy-In: Secure buy-in from stakeholders to foster support and commitment to the project.
- **Anatomy of a Successful Project**:
- Best Practices: Implement best practices for project management, including regular status updates, risk assessments, and stakeholder engagement.
- Iterative Reviews: Conduct iterative reviews to assess progress and make necessary adjustments to the project plan.
- Ensure continued, long-term, committed support from the executive sponsors. Up front, establish well-defined, real, and agreed business value from your data warehouse.
- Manage user expectations realistically. Get the users enthusiastically involved throughout the project.

- The data extraction, transformation, and loading (ETL) function is the most timeconsuming, labor-intensive activity. Do not under-estimate the time and effort for this activity. Remember architecture first, then technology, then tools.
- Select an architecture that is right for your environment. The right query and information tools for the users are extremely critical.
- Select the most useful and easy-to-use ones, not the glamorous. Avoid bleedingedge technology.
- Plan for growth and evolution. Be mindful of performance considerations. Assign a user-oriented project manager.
- Focus the design on queries, not transactions. Define proper data sources. Only load the data that is needed.

Figure 4-12 Key success factors for a data warehouse project.

➢ **Adopt a Practical Approach:**

- Realistic Planning: Develop a realistic project plan that considers available resources, timelines, and potential challenges.
- Flexibility: Be willing to adapt the project plan as needed based on changing business requirements or unforeseen obstacles



Business Context

BigCom, Inc., world's leading supplier of data, voice, and video communication technology with more than 300 million customers and significant recent growth.

Challenges

Limited availability of global information; lack of common data definitions; critical business data locked in numerous disparate applications; fragmented reporting needing elaborate reconciliation; significant system downtime for daily backups and updates.

Technology and Approach

Deploy large-scale corporate data warehouse to provide strategic information to 1,000 users for making business decisions; use proven tools from single vendor for data extraction and building data marts; query and analysis tool from another reputable vendor.

Success Factors

Clear business goals; strong executive support; user departments actively involved; selection of appropriate and proven tools; building of proper architecture first; adequate attention to data integration and transformation; emphasis on flexibility and scalability.

Benefits Achieved

True enterprise decision support; improved sales measurement; decreased cost of ownership; streamlined business processes; improved customer relationship management; reduced IT development; ability to incorporate clickstream data from company's Web site.

Figure 4-13 Analysis of a successful data warehouse.

CHAPTER 5: DEFINING THE BUSINESS REQUIREMENTS

DIMENSIONAL ANALYSIS

In several ways, building a data warehouse is very different from building an operational system. This becomes notable especially in the requirements gathering phase. Because of this difference, the traditional methods of collecting requirements that work well for operational systems cannot be directly applied to data warehouses.

- **Usage of Information Unpredictable:**
  - Business requirements can be unpredictable, and users may not always know what information they need.
  - The nature of business data is often dimensional, requiring a structured approach to analysis.
- **Dimensional Nature of Business Data**:
  - Business data is inherently multidimensional, meaning it can be analyzed from various perspectives.
  - Examples include sales data analyzed by product, region, time, and customer demographics.

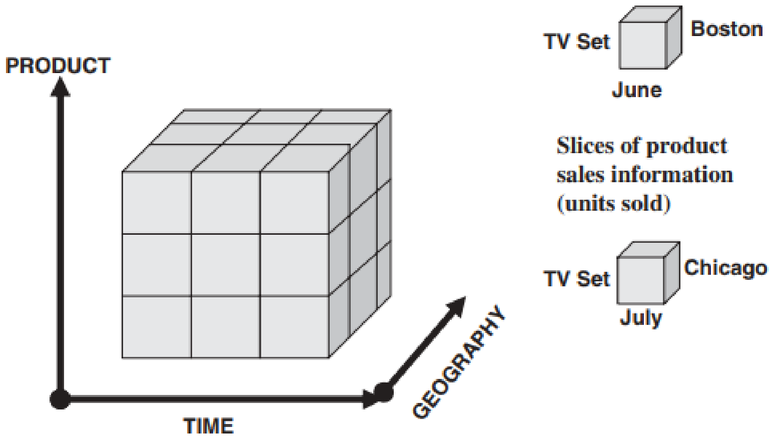


Figure 5-2 Dimensional nature of business data.

- **Examples of Business Dimensions**:

Time: Year, quarter, month, day.

Product: Category, brand, SKU.

Customer: Age, location, purchasing history.

Geography: Region, city, store.

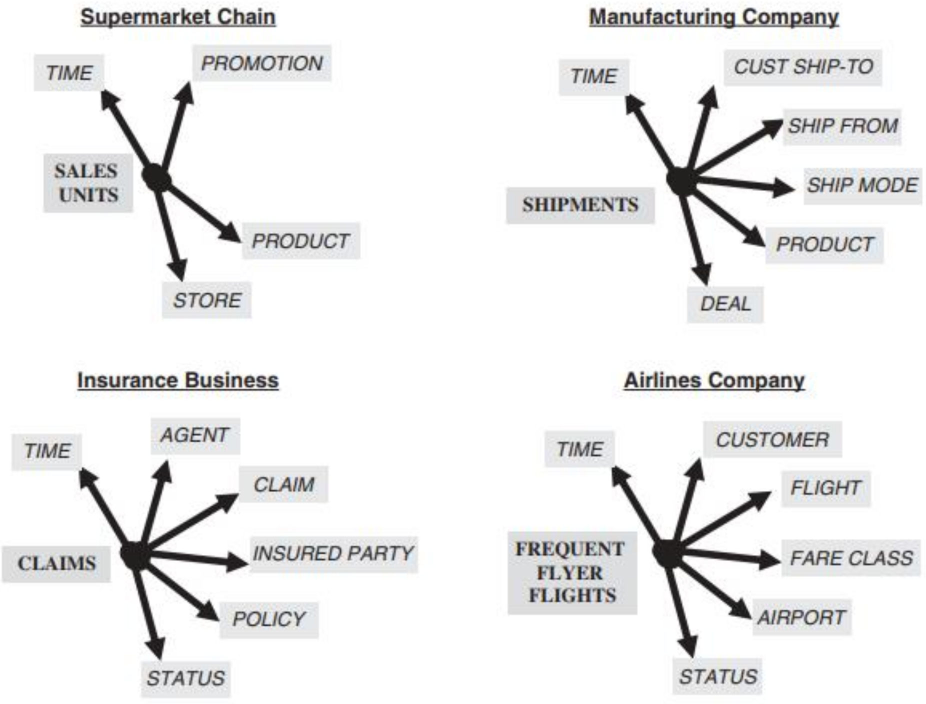
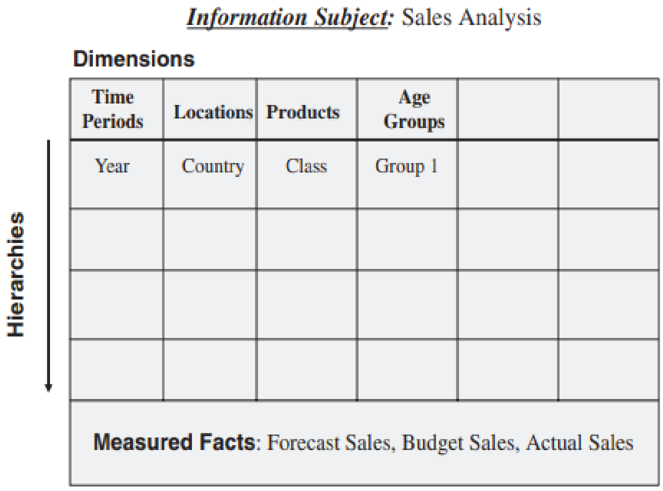


Figure 5-3 Examples of business dimensions.

INFORMATION PACKAGES—A USEFUL CONCEPT

- **Requirements Not Fully Determinate**:

- Business requirements can evolve, and initial needs may change as users interact with data.
- Flexibility in requirement definitions is crucial to accommodate changes.



Information Subject: Sales Analysis

Dimensions

Hierarchies

Time Periods	Locations	Products	Age Groups		
Year	Country	Class	Group 1		

Measured Facts: Forecast Sales, Budget Sales, Actual Sales

Figure 5-4 An information package.

- **Business Dimensions:**

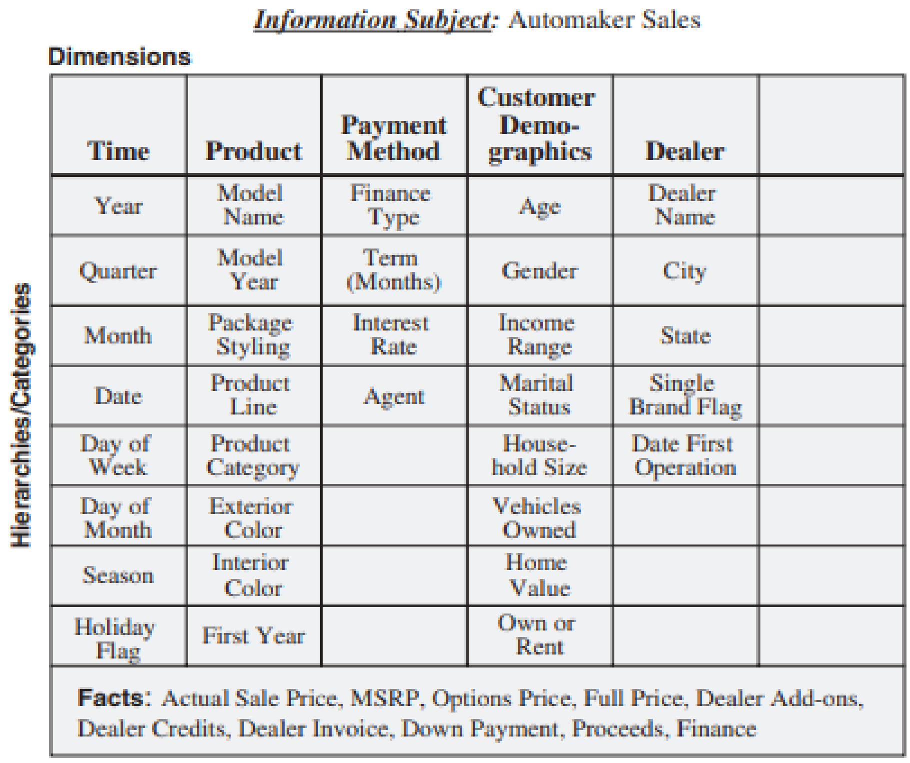
- Clearly defined business dimensions help in structuring data and ensuring relevant information is captured.
- Dimensions provide context for analyzing key business metrics.

- **Dimension Hierarchies and Categories**:

- Dimensions can have hierarchies (e.g., year → quarter → month) that allow for drill-down analysis.
- Categories within dimensions help to segment data for more focused analysis.

- **Key Business Metrics or Facts**:

- Identify key performance indicators (KPIs) and metrics that are critical for decision-making.
- Examples include sales revenue, customer acquisition cost, and inventory turnover.



Information Subject: Automaker Sales

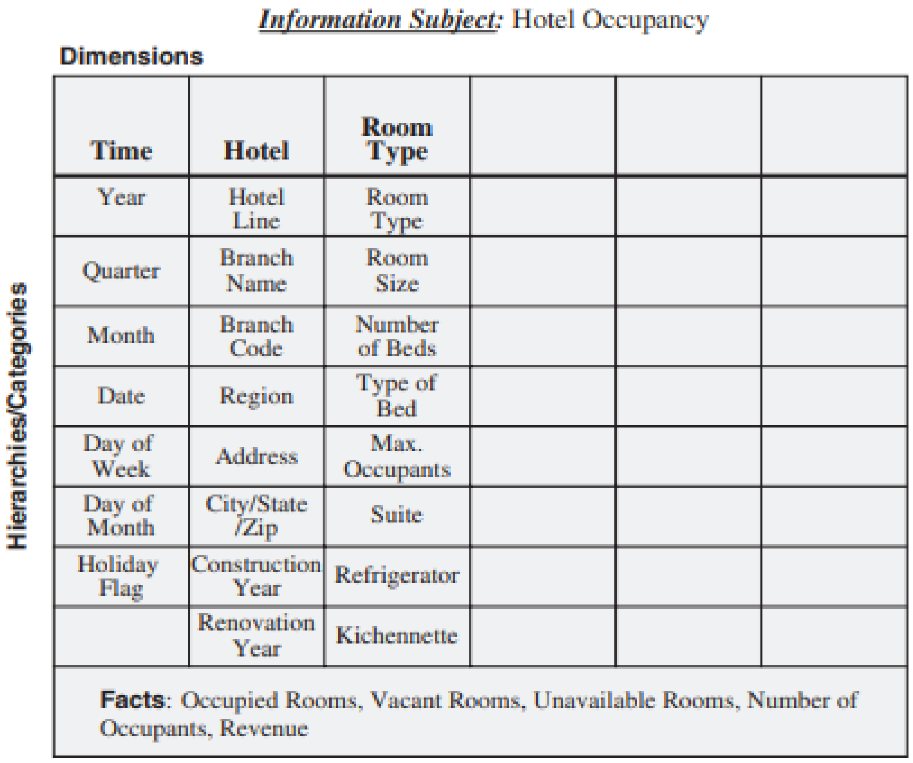
Dimensions

Hierarchies/Categories

Time	Product	Payment Method	Customer Demographics	Dealer	
Year	Model Name	Finance Type	Age	Dealer Name	
Quarter	Model Year	Term (Months)	Gender	City	
Month	Package Styling	Interest Rate	Income Range	State	
Date	Product Line	Agent	Marital Status	Single Brand Flag	
Day of Week	Product Category		House-hold Size	Date First Operation	
Day of Month	Exterior Color		Vehicles Owned		
Season	Interior Color		Home Value		
Holiday Flag	First Year		Own or Rent		

Facts: Actual Sale Price, MSRP, Options Price, Full Price, Dealer Add-ons, Dealer Credits, Dealer Invoice, Down Payment, Proceeds, Finance

Figure 5-5 Information package: automaker sales.



Information Subject: Hotel Occupancy

Dimensions

Hierarchies/Categories

Time	Hotel	Room Type			
Year	Hotel Line	Room Type			
Quarter	Branch Name	Room Size			
Month	Branch Code	Number of Beds			
Date	Region	Type of Bed			
Day of Week	Address	Max. Occupants			
Day of Month	City/State /Zip	Suite			
Holiday Flag	Construction Year	Refrigerator			
	Renovation Year	Kichennette			

Facts: Occupied Rooms, Vacant Rooms, Unavailable Rooms, Number of Occupants, Revenue

Figure 5-6 Information package: hotel occupancy.

REQUIREMENTS GATHERING METHODS

 Requirements Gathering: The process of collecting the needs and expectations of users for a data warehouse system.

 Types of Users:

- **Senior Executives**: Provide strategic direction and high-level scope.
- **Key Departmental Managers**: Define operational needs and focus areas.
- **Business Analysts**: Handle reports and analyses based on the system.
- **IT Staff and DBAs**: Provide insights into existing operational systems and data sources.

 Types of Requirements:

- **Data Elements**: Key metrics and dimensions (e.g., sales figures, customer segments).
- **Business Rules**: Conditions or rules under which the system operates.
- **Data Sources**: Extracting data from existing systems.

 Methods for Gathering Requirements:

- **Interviews**: One-on-one or small group discussions with users to gather detailed information.
- **Joint Application Development (JAD) Sessions**: Workshops with multiple users to confirm and elaborate on requirements.
- **Questionnaires**: Forms to collect information quickly from a large or dispersed group.

 Types of Questions:

- **Open-Ended Questions**: Encourage detailed responses (e.g., "What challenges do you face with current data reports?").
- **Closed Questions**: Seek specific, focused answers (e.g., "Do you need data updated daily?").

 Interview Structures:

- **Pyramid Structure**: Start with specific questions, then expand.

- **Funnel Structure**: Begin with broad questions and narrow down.
- **Diamond Structure**: Mix of both, often preferred for balance.

 Joint Application Development (JAD): A methodology to involve users and IT professionals in structured sessions to gather, refine, and confirm requirements.

- ➢ **Adapting the JAD Methodology**:
    - Joint Application Development (JAD) involves collaborative sessions with users to define requirements.
    - This approach fosters communication and ensures that user needs are accurately captured.
- ➢ **Phases of JAD:**

1. **Project Definition:**
    - o High-level interviews with management are conducted to outline the scope and objectives.
    - o These initial interviews help identify the key stakeholders and the direction of the project.
    - o A **management definition guide** is prepared to communicate this understanding.
2. **Research:**
    - o The project team gathers detailed information about the business area and current systems.
    - o This includes identifying user information needs, understanding business processes, and preparing for the next phases.
    - o Preliminary data gathering is conducted to lay the groundwork for the JAD sessions.
3. **Preparation:**
    - o A working document based on the research is created.
    - o The project team conducts training for scribes and prepares visual aids and other necessary tools.
    - o Pre-session meetings are held to set expectations and establish a checklist of objectives.
4. **JAD Sessions:**

  - These sessions typically open with a review of the agenda and the purpose.
  - Assumptions are reviewed, and data requirements, business metrics, dimensions, and hierarchies are discussed.
  - The group works together to resolve open issues and finalize decisions about the data warehouse design.
  - The sessions end with a list of action items, outlining what steps need to be taken next.

5. **Final Document:**
   - The working document is finalized, mapping all the gathered information, including data sources, business metrics, dimensions, and hierarchies.
   - Review sessions are conducted to ensure the accuracy of the document, and final approvals are obtained.
   - A change procedure is established to manage any future adjustments to the requirements.

Participants in JAD:

1. **Executive Sponsor:**
   - The person controlling the project’s funding, providing overall direction, and empowering the team to make decisions.
2. **Facilitator:**
   - The guide who leads the team through the JAD process, ensuring that sessions are productive and objectives are met.
3. **Scribe:**
   - The person responsible for documenting decisions and discussions during the JAD sessions.
4. **Full-Time Participants:**
   - Individuals who are involved in making decisions throughout the entire project.
5. **On-Call Participants:**
   - Experts or stakeholders who are brought in when specific areas of the project need their input.
6. **Observers:**
   - Those who sit in on sessions to observe but do not participate in the decision-making process.

➢ Using Questionnaires in JAD Sessions:

Questionnaires are a useful tool for gathering data, especially when direct interaction is not possible. Here are a few points to keep in mind when using questionnaires:

- **Type and Choice of Questions:** A mix of open-ended and closed questions helps in gaining both detailed responses and straightforward answers. For instance, in a real-world data warehouse project for an e-commerce business, closed questions could ask how often users access sales reports, while open-ended questions could explore what additional data insights they would like to have.
- **Application of Scales:** Using nominal scales to categorize responses (e.g., user roles like 'Manager', 'Analyst') and interval scales to measure frequency or importance (e.g., rating the importance of various reports on a scale of 1-5) ensures that data is quantifiable for analysis.
- **Questionnaire Design:** Just as in surveys for customer satisfaction, JAD questionnaires must be user-friendly and non-intrusive. For example, start with simple questions like “What types of reports do you frequently use?” before diving into more complex questions about data analysis preferences.
- **Administering Questionnaires:** Questionnaires can be distributed during JAD sessions or through email or online forms to collect responses in advance, allowing participants to focus on more critical issues during the session itself. An example could be sending pre-session questionnaires to department heads to gather initial requirements, which are then discussed in depth during the session.

➢ Review of Existing Documentation:

Reviewing documentation is vital to understand the current operational systems and business processes without burdening the business users too much. The process involves:

- **Documentation from User Departments:** This involves looking at existing reports and operational systems to understand how users interact with data. For instance, in a retail organization, reviewing sales reports and inventory systems would help understand the data flows that are critical for the data warehouse.
- **Documentation from IT:** IT’s role here is crucial, as they provide the technical details about the source systems (e.g., database structures, data fields, relationships). For

example, in a finance data warehouse project, IT would provide data dictionaries for various financial systems (e.g., Oracle, SAP) that feed data into the warehouse.

REQUIREMENTS DEFINITION: SCOPE AND CONTENT:

This is the formal documentation created after the JAD sessions and other requirement-gathering activities. It acts as a foundation for subsequent phases of the project. Let’s look at key elements:

1. **Data Sources:** Listing all data sources (e.g., CRM systems, ERP systems) ensures that the project team knows where to extract the data from. For example, in a telecom company, you may list data sources like customer usage databases, billing systems, and customer support records.
2. **Data Transformation:** Data from operational systems often needs to be cleaned and transformed before being loaded into the data warehouse. For instance, sales data from a point-of-sale system may need to be aggregated by date or product category before being stored.
3. **Data Storage:** Understanding the level of detailed and aggregated data is critical. For example, a retail chain might need detailed transactional data to analyze daily sales and summary data for weekly or monthly reporting.
4. **Information Delivery:** The requirements definition should specify how users expect to access and analyze the data, whether through dashboards, ad hoc reports, or more advanced tools like OLAP (Online Analytical Processing). A marketing department may want to slice and dice customer data by demographics, product categories, and regions.

CHAPTER 6: REQUIREMENTS AS THE DRIVING FORCE FOR DATA WAREHOUSING

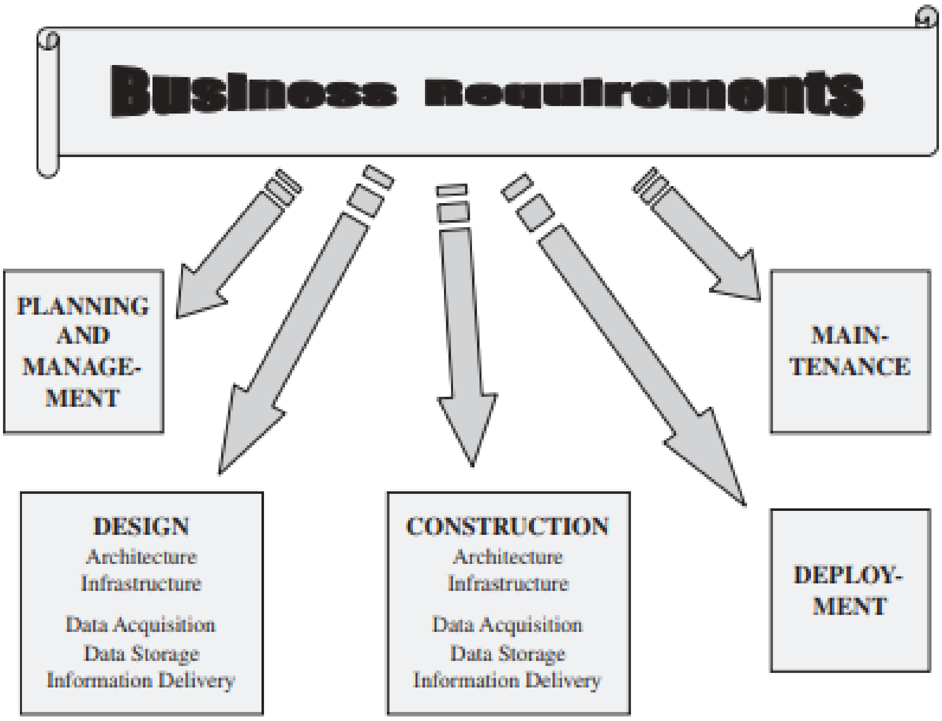


Figure 6-1 Business requirements as the driving force.

Requirements as the Driving Force for Data Warehousing

The first diagram (Figure 6-1) shows how **business requirements** are the key driving force behind all phases of a data warehouse project:

- **Planning and Management**: This is where the project's scope is defined based on the business needs. In the case of Netflix, this would involve determining what kind of data (e.g., viewing habits, user preferences) the warehouse needs to handle.
- **Design**: This phase includes the architecture of the data warehouse, such as what data will be stored, where it will come from, and how it will be structured.
    - For Netflix, the **design** would involve creating data models to track user interactions (e.g., which shows users watch, for how long, etc.) and organizing this information into the warehouse for easy retrieval.
- **Construction**: Once designed, the warehouse is built, including data extraction (from operational databases), storage, and how users will access the data.

  - For example, Netflix would collect data from user devices, store it in a central repository, and create methods for analysts to access and query this data.
- **Deployment**: After construction, the warehouse is deployed for use, and users begin interacting with the data.
  - Netflix could deploy the warehouse to give their business intelligence teams tools for real-time analytics on show popularity, churn rate, and more.
- **Maintenance**: Regular updates are made to ensure the warehouse remains functional and relevant.
  - For example, Netflix might add new data sources as the platform expands into new regions or develops new features.

Data Design

In the second diagram (Figure 6-2), the process of designing the data warehouse is shown, with requirements driving both **dimensional modeling** (for reporting) and **relational modeling** (for structured data).

- **Relational Model**: This is used for the **Enterprise Data Warehouse**, where structured data is stored and retrieved efficiently.
  - For Netflix, this could be the backend database storing information like user profiles, subscriptions, and payment information.
- **Dimensional Model**: This is used to build **Data Marts**, which are subsets of data tailored for specific analysis.
  - For Netflix, there might be a data mart dedicated to analyzing content preferences, showing metrics like viewing time, preferred genres, and so on.

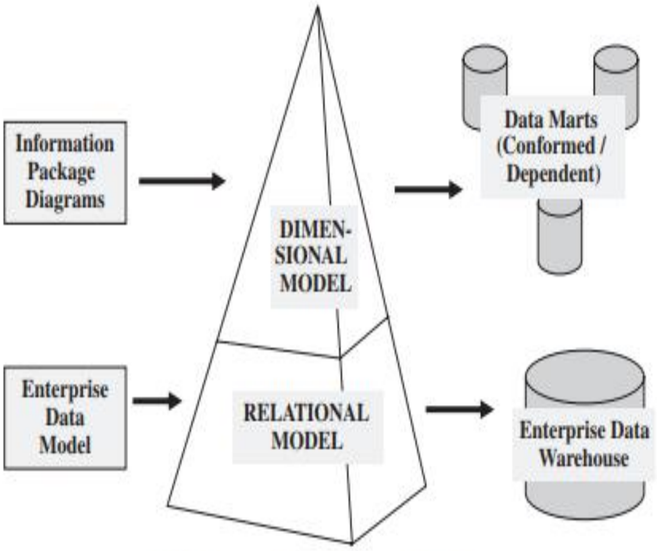
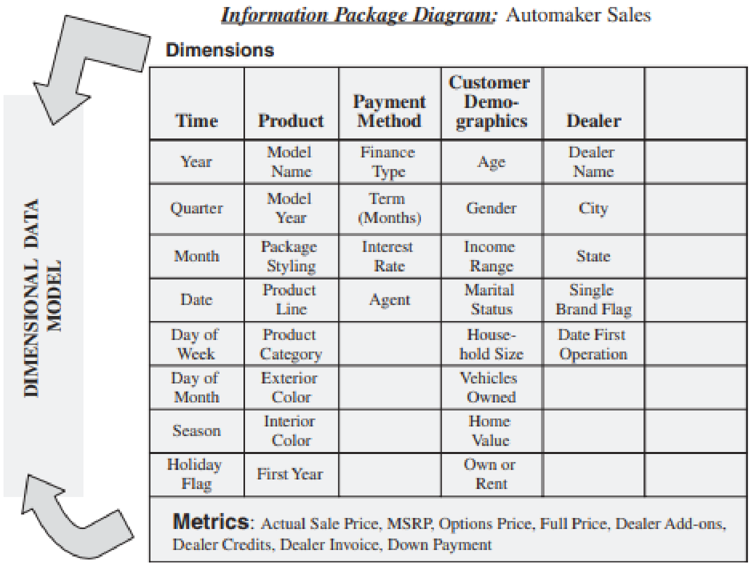

Information Package Diagrams
DIMEN-SIONAL MODEL
Data Marts (Conformed / Dependent)
Enterprise Data Model
RELATIONAL MODEL
Enterprise Data Warehouse

Figure 6-2 Requirements driving the data model.

➢ Structure for Business Dimensions:

- Defining Dimensions: Clearly define business dimensions that will be used in the data warehouse.
- Hierarchical Organization: Organize dimensions hierarchically to facilitate drill-down analysis (e.g., Year → Quarter → Month).
- Conforming Dimensions: Ensure that dimensions are consistent across different data marts to maintain data integrity.


DIMENSIONAL DATA MODEL

Information Package Diagram: Automaker Sales

Dimensions

Time	Product	Payment Method	Customer Demo-graphics	Dealer	
Year	Model Name	Finance Type	Age	Dealer Name	
Quarter	Model Year	Term (Months)	Gender	City	
Month	Package Styling	Interest Rate	Income Range	State	
Date	Product Line	Agent	Marital Status	Single Brand Flag	
Day of Week	Product Category		House-hold Size	Date First Operation	
Day of Month	Exterior Color		Vehicles Owned		
Season	Interior Color		Home Value		
Holiday Flag	First Year		Own or Rent		

Metrics: Actual Sale Price, MSRP, Options Price, Full Price, Dealer Add-ons, Dealer Credits, Dealer Invoice, Down Payment

Figure 6-3 Business dimensions in the data model.

- **Structure for Key Measurements:**
- Key Performance Indicators (KPIs): Identify and define KPIs that are critical for measuring business performance.
- Fact Tables: Design fact tables to store quantitative data (e.g., sales, revenue) related to the defined dimensions.
- Granularity: Determine the level of detail (granularity) for the data stored in fact tables, balancing detail with performance.

- **Levels of Detail**:
  - Different Levels: Consider multiple levels of detail for data storage to allow for both summary and detailed analysis.
  - Aggregation: Implement aggregation strategies to summarize data for higher-level reporting while maintaining detailed records for analysis.

Architectural Plan

The **data warehouse architecture** is a blueprint for organizing the components of a data warehouse in a way that meets business requirements. Every data warehouse has similar architectural components, but the **size, scope, and integration** of these components vary depending on the business needs. The architecture includes multiple layers:

- **Composition of the Components**:
  - Data Sources: Identify all data sources that will feed into the data warehouse, including operational databases and external data.
    - Production data
    - Internal data
    - Archived data
    - External data
  - Data Staging Area: Design a staging area for data extraction, transformation, and loading (ETL) processes.

- Data extraction
- Data transformation
- Data loading

- Data Warehouse: Define the architecture for the data warehouse, including storage and processing components.
  - Data storage
  - Information delivery
  - Metadata Management and control

➢ **Special Considerations**:

- Performance Requirements: Ensure that the architecture supports performance requirements for querying and reporting.
- Scalability: Design the architecture to be scalable to accommodate future data growth and increased user demand.

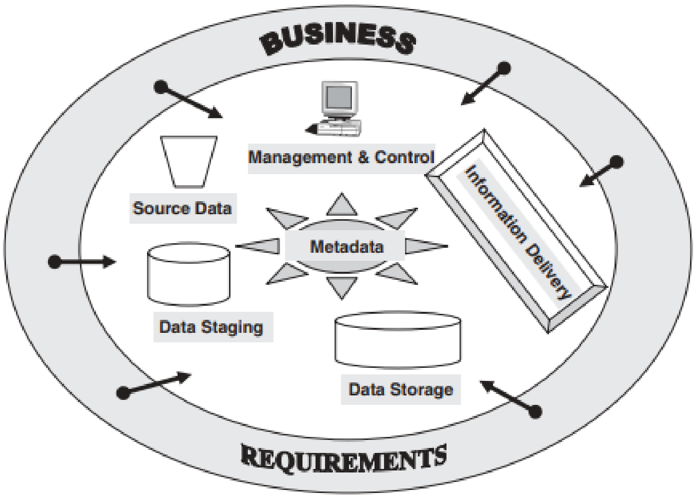


Figure 6-4 Impact of requirements on architecture.

- Data Extraction: Clearly identify all the internal data sources. Specify all the computing platforms and source files from which the data is to be extracted. If you are going to include external data sources, determine the compatibility of your data structures with those of the outside sources. Also indicate the methods for data extraction.
- Data Transformation: Many types of transformation functions are needed before data can be mapped and prepared for loading into the data warehouse repository. These functions include input selection, separation of input structures, normalization and de normalization of source structures, aggregation, conversion, resolving of missing

values, and conversions of names and addresses. In practice, this turns out to be a long and complex list of functions. Examine each data element planned to be stored in the data warehouse against the source data elements and ascertain the mappings and transformations.

- Data Loading: Define the initial load. Determine how often each major group of data must be kept up-to-date in the data warehouse. How much of the updates will be nightly updates? Does your environment warrant more than one update cycle in a day? How are the changes going to be captured in the source systems? Define how the daily, weekly, and monthly updates will be initiated and carried out. If your plan includes real time data warehousing, specify the method for real time updates.
- Data Quality: Bad data leads to bad decisions. No matter how well you tune your data warehouse, and no matter how adeptly you provide for queries and analysis functions to the users, if the data quality of your data warehouse is suspect, the users will quickly lose confidence and flee the data warehouse. Even simple discrepancies can result in serious repercussions while making strategic decisions with far-reaching consequences.
- Data quality in a data warehouse is sacrosanct. Therefore, right in the early phase of requirements definition, identify potential sources of data pollution in the source systems. Also, be aware of all the possible types of data quality problems likely to be encountered in your operational systems. Note the following tips.
    - Data Pollution Sources System conversions and migrations
    - Heterogeneous systems integration
    - Inadequate database design of source systems
    - Data aging Incomplete information from customers
    - Input errors Internationalization/localization of systems
    - Lack of data management policies/procedures
    - Types of Data Quality Problems Dummy values in source system fields
    - Absence of data in source system fields Multipurpose fields
    - Cryptic data Contradicting data Improper use of name and address lines
    - Violation of business rules Reused primary keys Non unique identifier

- **Tools and Products:**
  - ETL Tools: Select appropriate ETL tools for data extraction, transformation, and loading processes.

Database Management Systems (DBMS): Choose a suitable DBMS that supports the architectural requirements of the data warehouse.

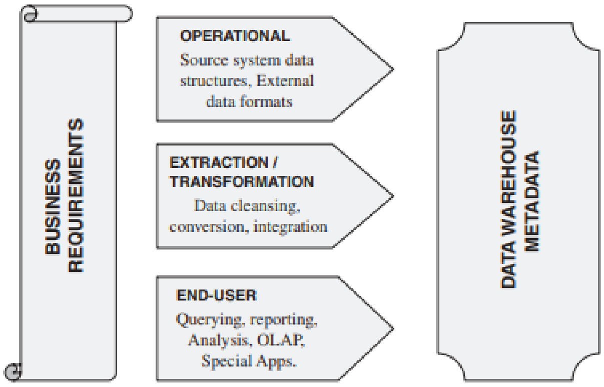

BUSINESS REQUIREMENTS
OPERATIONAL
Source system data structures, External data formats
EXTRACTION / TRANSFORMATION
Data cleansing, conversion, integration
END-USER
Querying, reporting, Analysis, OLAP, Special Apps.
DATA WAREHOUSE METADATA

Figure 6-5 Impact of requirements on metadata.

DATA STORAGE SPECIFICATIONS

- **DBMS Selection**:
  - Criteria for Selection: Consider factors such as performance, scalability, support for analytical queries, and cost when selecting a DBMS.
  - Compatibility: Ensure compatibility with existing systems and data sources.
    - Level of User Experience: If the users are totally inexperienced with database systems, the DBMS must have features to monitor and control runaway queries. On the other hand, if many of your users are power users, then they will be formulating their own queries. In this case, the DBMS must support an easy SQL-type language interface.
    - Types of Queries: The DBMS must have a powerful optimizer if most of the queries are complex and produce large result sets. Alternatively, if there is an even mix of simple and complex queries, there must be some sort of query management in the database software to balance the query execution.

- Need for Openness: The degree of openness depends on the back-end and front-end architectural components and those, in turn, depend on the business requirements.
- Data Loads: The data volumes and load frequencies determine the strengths in the areas of data loading, recovery, and restart.
- Metadata Management: If your metadata component does not have to be elaborate, then a DBMS with an active data dictionary may be sufficient. Let your requirements definition reflect the type and extent of the metadata framework.
- Data Repository Locations: Is your data warehouse going to reside in one central location, or is it going to be distributed? The answer to this question will establish whether the selected DBMS must support distributed databases.
- Data Warehouse Growth: Your business requirements definition must contain information on the estimated growth in the number of users, and in the number and complexity of queries. The growth estimates will have a direct relation to how the selected DBMS supports scalability

- **Storage Sizing**:
  - Capacity Planning: Estimate the storage requirements based on the volume of data to be stored and the expected growth over time.
  - Data Retention Policies: Define data retention policies to determine how long data will be stored in the warehouse.

- Data Staging Area: Calculate storage estimates for the data staging area of the overall corporate data warehouse from the sizes of the source system data structures for each business subject. Figure the data transformations and mapping into your calculation. For the data marts, initially estimate the staging area storage based on the business dimensions and metrics for the first data mart.
- Overall Corporate Data Warehouse: Estimate the storage size based on the data structures for each business subject. You know that data in the data warehouse is stored by business subjects. For each business subject, list the various attributes, estimate their field lengths, and arrive at the calculation for the storage needed for that subject.
- Data Marts—Conformed, Independent, Dependent, or Federated: While defining requirements, you create information diagrams. A set of these diagrams constitutes a data mart. Each information diagram contains business dimensions and their attributes. The information diagram also holds the metrics or business measurements that are meant for analysis. Use the details of the business dimensions and business measures found in the information diagrams to estimate the storage size for the data marts. Begin with your first data mart.
- Multidimensional Databases: These databases support OLAP or multidimensional analysis. How much online analytical processing (OLAP) is necessary for your users? The corporate data warehouse or the individual conformed or dependent data mart supplies the data for the multidimensional databases. Work out the details of OLAP planned for your users and then use those details to estimate storage for these multidimensional databases

INFORMATION DELIVERY STRATEGY

- **Queries and Reports**:
  - User Requirements: Gather requirements for types of queries and reports that users will need.
  - Ad Hoc Reporting: Plan for ad hoc reporting capabilities to allow users to generate their own reports as needed.

- **Types of Analysis**:
  - Descriptive Analysis: Provide tools for users to perform descriptive analysis to understand historical data.
  - Predictive Analysis: Implement capabilities for predictive analysis to forecast future trends based on historical data.

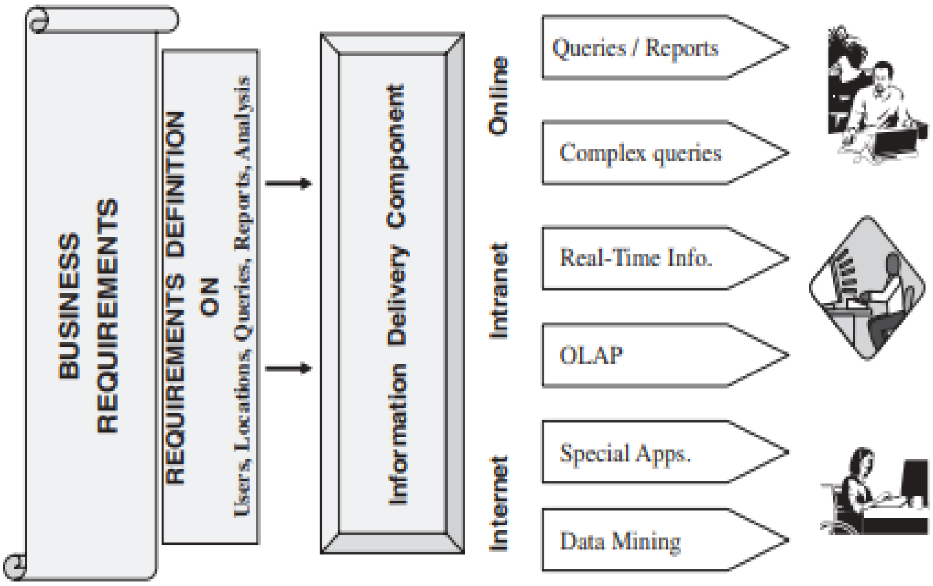


Figure 6-6 Impact of business requirements on information delivery.

- **Information Distribution**:
    - Delivery Mechanisms: Define how information will be delivered to users, including dashboards, reports, and alerts.
    - User Interfaces: Design user-friendly interfaces that facilitate easy access to the data warehouse.

- **Real-Time Information Delivery:**
    - Timeliness: Ensure that users have access to timely information for decision-making.
    - Streaming Data: Consider incorporating streaming data capabilities for real-time analysis.

- **Decision Support Applications**:
    - Integration with BI Tools: Integrate the data warehouse with business intelligence (BI) tools to enhance decision support capabilities.
    - User Training: Provide training for users on how to effectively use the tools and access the data they need.