Business Intelligence and Analytics

1. Introduction to Business Intelligence

❖ Definition and history of Business intelligence

Executive Information Systems (EIS) are **computer-based tools** designed specifically to support the **information and decision-making needs of senior executives**. These systems provide **easy access to both internal and external data** relevant to organizational goals, and they typically feature:

- Simple, **user-friendly interfaces**
- **High-level summaries** of organizational data
- **Drill-down capabilities** to explore more detailed information
- **Graphs, dashboards**, and **trend analyses**
- **Alerts** on significant changes or performance issues

How EIS Evolved into Business Intelligence (BI):

- Initially, EIS were standalone tools created to help top-level managers make quick, strategic decisions using summarized data and visuals.
- Over time, the demand for **broader access** to data-driven decision-making grew beyond just executives to include **middle managers and analysts**.
- EIS began integrating more advanced features like **data visualization, performance metrics**, and **real-time alerts**.
- As **enterprise-wide data systems** developed and more organizations adopted **data warehouses**, these EIS tools were extended and rebranded with **more comprehensive analytical capabilities**.
- By around **2006**, the functionalities offered by EIS—combined with **data mining, reporting, online analytical processing (OLAP)**, and **predictive analytics**—began to fall under a unified category known as **Business Intelligence (BI)**.
- BI thus became an **umbrella term** for all tools, systems, and practices that help organizations make better decisions using data—including what was formerly known as EIS.

Executive Information Systems (EIS) were the early forms of decision support tools for executives. As they evolved to include more advanced analytics and broader accessibility, they became part of the broader **Business Intelligence (BI)** landscape.

- Definition of BI :

  Business Intelligence (BI) is an umbrella term that encompasses **architectures, tools, databases, analytical tools, applications, and methodologies**. It is a **content-free expression**, meaning it can vary in interpretation among different people. The primary objective of BI is to **enable interactive (sometimes real-time) access to data**, allow for **data manipulation**, and give **business managers and analysts** the ability to perform appropriate analyses. Through the examination of **historical and current data, situations, and performance**, BI provides **valuable insights** that support more **informed and effective decision-making**. The BI process involves the **transformation of data into information, then into decisions, and finally into actions**.

- History of BI

The term **Business Intelligence (BI)** was coined by the **Gartner Group in the mid-1990s**, but the underlying concept originated much earlier. It evolved from the **Management Information Systems (MIS)** of the **1970s**, which were limited to **static, two-dimensional reporting** with **no analytical capabilities**.

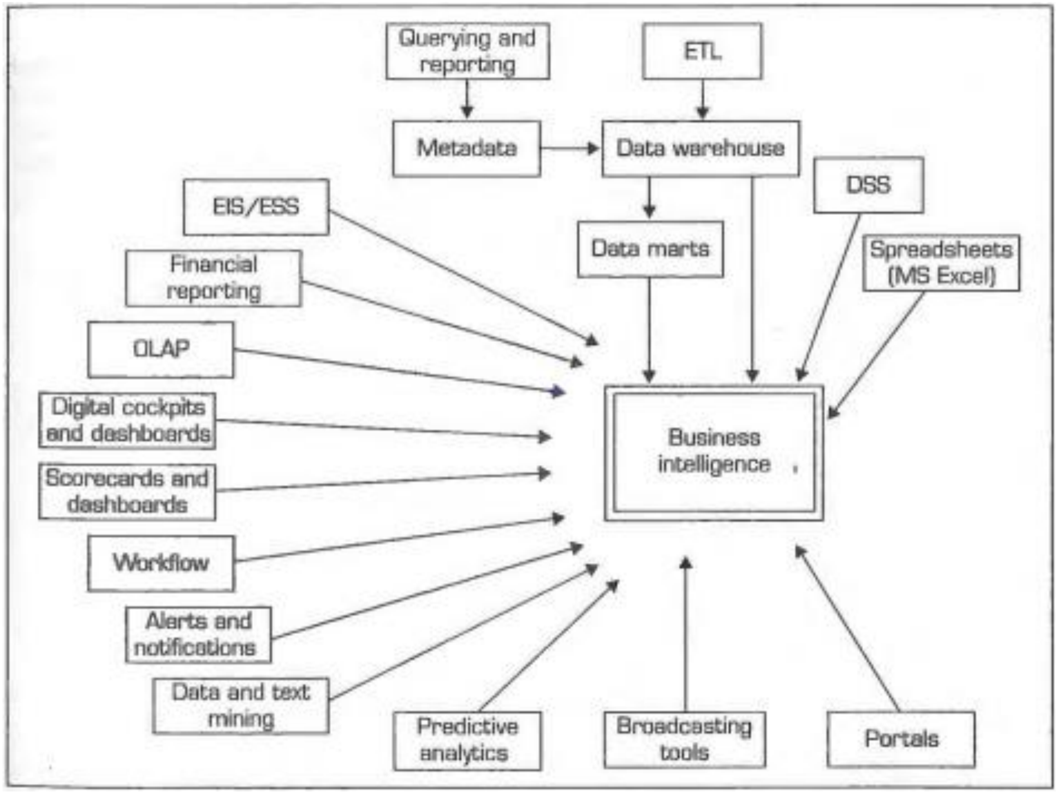
In the **early 1980s**, **Executive Information Systems (EIS)** emerged to support **top-level managers and executives**. These systems introduced advanced features like:

- **Dynamic multidimensional (ad hoc/on-demand) reporting**
- **Forecasting and prediction**
- **Trend analysis**
- **Drill-down capabilities**
- **Status monitoring**
- **Tracking of Critical Success Factors (CSFs)**

From the 1980s to the mid-1990s, such features became common in many commercial products. Eventually, these capabilities, along with new ones, were brought together under the broader term **Business Intelligence (BI)**.

By **2005**, BI systems began incorporating **artificial intelligence (AI)** and **powerful analytical tools**, making them more advanced and comprehensive. Today’s BI systems provide a complete **enterprise information solution**, giving decision-makers all the data and analytical tools they need to make informed decisions.

Thus, **BI evolved from static MIS to dynamic, intelligent systems**, transforming the **original EIS concept** into a powerful decision-support framework.



✓ Business Intelligence (BI) Architecture

A **Business Intelligence (BI) system** consists of **four major components**, which together form its architecture:

1. Data Warehouse

- It is the **cornerstone** of any **medium-to-large BI system**.
- Originally contained only **historical data**, which were **organized and summarized** for easy viewing and manipulation by end users.
- Now, some data warehouses also include **current data**, supporting **real-time decision making**.
- The **data warehousing environment** is primarily managed by **technical staff**.

2. Business Analytics

Business analytics refers to tools and techniques used by **end users** to **manipulate and analyze data** in the data warehouse. It has two major categories:

a. Reports and Queries

- Includes **static and dynamic reporting**, various types of **queries**, **information discovery**, **multidimensional views**, **drill-down to details**, etc.
- These capabilities are also related to **Business Performance Management (BPM)**.

b. Data, Text, and Web Mining

- Involves **searching for unknown patterns or relationships** in large datasets using:
  - **Neural computing**
  - **Predictive analytics**
  - **Advanced statistical methods**
- Can be applied to **structured data**, as well as **textual** and **web data**.

Examples:

- **Epagogix** predicts movie success by analyzing scripts.
- **National Australia Bank** uses mining tools for **predictive marketing** and **competitive analysis** based on historical data.

3. Business Performance Management (BPM)

- Also known as **Corporate Performance Management (CPM)**.
- It is a **portfolio of applications and methodologies** built on BI architecture and tools.
- BPM focuses on:
  - **Monitoring**, **measuring**, and **comparing** performance indicators (e.g., sales, profit, cost).
  - Introducing **management and feedback loops**.
  - Supporting **planning and forecasting** as strategic business activities.
- Unlike traditional BI tools which support **bottom-up** data analysis, BPM emphasizes **top-down enforcement** of **corporate-wide strategy**.
- Often integrated with **balanced scorecards** and **dashboards**.

4. User Interface

- The **user interface** provides access to the BI system through tools like:
  - **Dashboards** – visually present key performance indicators (KPIs), trends, and exceptions.
  - **Corporate portals**, **digital cockpits**, and other **visualization tools**.
- Dashboards integrate data from multiple business areas and give an **at-a-glance** view of organizational health.
- Visualization tools include **multidimensional cubes**, **virtual reality**, and **geographical information systems (GIS)**.
- Any user, including top managers, can connect through the **user interface**, such as a **browser**.

Summary of BI Architecture Components:

Component	Description
Data Warehouse	Stores historical (and now current) data; foundation of BI; managed by tech staff
Business Analytics	Tools for reporting, querying, mining; used by business users
BPM (Performance Mgmt)	Top-down strategy enforcement; includes planning, forecasting, scorecards
User Interface	Dashboards and visual tools for interaction; used via browser/portal

This architecture ensures **interactive access**, **manipulation**, and **analysis** of data to support **informed decision-making**.

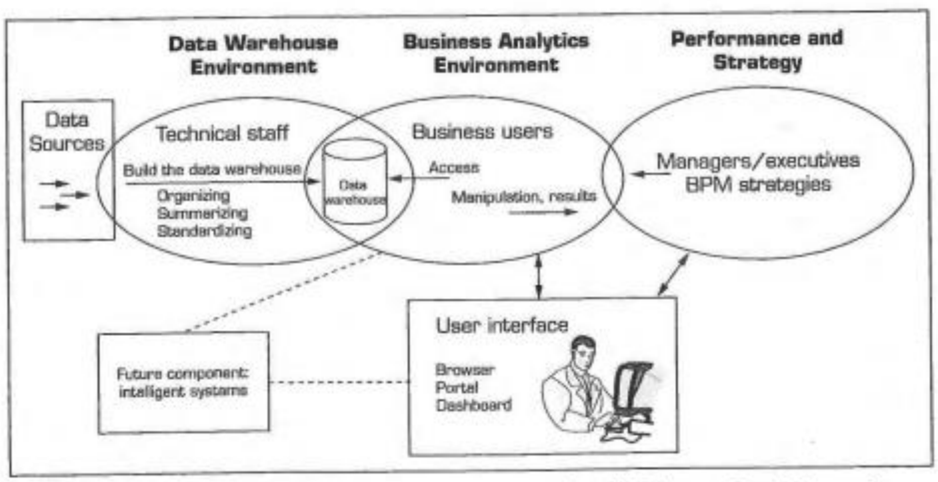


FIGURE 1.3 **A High-Level Architecture of BI.** *Source:* Based on W. Eckerson, *Smart Companies in the 21st Century: The Secrets of Creating Successful Business Intelligent Solutions*. The Data Warehousing Institute, Seattle, WA, 2003, p. 32, Illustration 5.

✓ Styles of Business Intelligence (BI)

The **architecture of a BI system** can vary based on the **type of application** it supports. According to **MicroStrategy Corporation**, there are **five primary styles of BI**, each with dedicated tools and capabilities:

1. **Report Delivery and Alerting**
   - Provides **timely delivery** of reports and **automatic alerts** based on specific conditions or thresholds.
2. **Enterprise Reporting**
   - Involves the use of **dashboards** and **scorecards** for comprehensive, organization-wide **reporting and performance tracking**.
3. **Cube Analysis** (also known as **Slice-and-Dice Analysis**)

   - Allows users to **interactively explore multidimensional data** by slicing, dicing, drilling down, and pivoting to view different data perspectives.
4. **Ad Hoc Queries**
   - Enables **users to create and run their own custom queries** to extract meaningful insights without needing IT support.
5. **Statistics and Data Mining**
   - Focuses on **discovering hidden patterns and relationships** using **statistical models** and **data mining techniques**.

✓ Benefits of Business Intelligence (BI)

The **major benefit of BI** is its ability to **provide accurate and timely information**, offering a **real-time view** of organizational performance and its components. This is essential for:

- **Decision-making**
- **Strategic planning**
- **Operational efficiency**
- Even **business survival**

Common BI Application Areas (as per Thompson, 2004):

- **General Reporting**
- **Sales and Marketing Analysis**
- **Planning and Forecasting**
- **Financial Consolidation**
- **Statutory Reporting**
- **Budgeting**
- **Profitability Analysis**

Why Organizations Rely on BI:

- Organizations are now **compelled** to **capture, understand, and use data** effectively to improve business operations.
- **Regulations** like the **Sarbanes-Oxley Act of 2002** require businesses to:
   - Document processes.
   - Verify and **sign off** on the **accuracy** of reported information.
- Business environments demand:
   - **Faster decision-making**
   - **Smarter operations**
   - **Real-time insight**

This makes **accurate, real-time information** a **competitive necessity**.

The growing need for smart, data-driven operations is why BI has become a **priority for organizations worldwide**.

Business Intelligence (BI) supports **multiple styles** tailored to diverse business needs—ranging from **reporting** and **dashboards** to **data mining** and **custom queries**. The **benefits of BI** lie in its power to deliver the **right information at the right time and place**, enabling better and faster decisions. As a result, BI is essential not just for efficiency, but also for compliance, competitiveness, and strategic advantage.

TABLE 1.2 Business Value of BI Analytical Applications

Analytic Application	Business Question	Business Value
Customer segmentation	What market segments do my customers fall into, and what are their characteristics?	Personalize customer relationships for higher satisfaction and retention.
Propensity to buy	Which customers are most likely to respond to my promotion?	Target customers based on their need to increase their loyalty to your product line. Also, increase campaign profitability by focusing on the most likely to buy.
Customer profitability	What is the lifetime profitability of my customer?	Make individual business interaction decisions based on the overall profitability of customers.
Fraud detection	How can I tell which transactions are likely to be fraudulent?	Quickly determine fraud and take immediate action to minimize cost.
Customer attrition	Which customer is at risk of leaving?	Prevent loss of high-value customers and let go of lower-value customers.
Channel optimization	What is the best channel to reach my customer in each segment?	Interact with customers based on their preference and your need to manage cost.

Source: A. Ziama and J. Kasher (2004), *Data Mining Primer for the Data Warehousing Professional.* Dayton, OH: Teradata.

✓ Automated Decision Systems (ADS)

Definition:

Automated Decision Systems (ADS)—also known as **Decision Automation Systems (DAS)**—are **rule-based systems** designed to **automatically solve specific, repetitive managerial problems** in a **particular functional area** (e.g., finance, manufacturing) and **industry** (e.g., banking, retail, airlines).

These systems **automate decision-making** processes using predefined **business rules** and are especially useful when:

- The problem is **repetitive**

- The decision logic can be **standardized**
- **Speed and consistency** are required

Key Characteristics of ADS:

- **Rule-based**: Decisions are made by applying a set of **predefined conditions or rules**.
- **Industry-specific**: Typically designed for **specific functions within specific industries**.
- **Focused scope**: Handles **repetitive, structured decision problems**, not broad strategic ones.
- **Speed & automation**: Allows for **instantaneous decisions**, reducing human intervention.
- **Frontline usability**: Often used by **frontline employees** for quick decisions based on available data.

Difference from Traditional Decision Models:

ADS (Automated Decision Systems)	Management Science (Traditional)
Rule-based solutions	Model-based solutions
Industry- and function-specific	General structured problem-solving
Applied to specific repetitive tasks	Applied to broader structured problems
Example: Approving a loan, setting product price	Example: Resource allocation, inventory control

Examples of ADS in Use:

1. Airline Industry (Origin of ADS)

- Known as **Revenue/Yield Management Systems**
- Automatically adjust **ticket prices** based on **real-time demand**
- **Example rule**:
  "If only 70% of the seats are sold 3 days before departure, offer a discount to non-business travelers."

2. Banking and Credit Industry

- Used to **automatically approve/reject loan or credit requests**
- **Example rule**:
  "If the applicant owns a house and earns over $100,000/year, offer a $10,000 credit line."

3. Retail and Procurement

- Used to **approve purchases** or determine **pricing**

- **Example rule**:
  "If an item costs more than $2,000 and is purchased only once a year, no special approval is needed."

4. Higher Education Admissions

- Used for **automated eligibility decisions**
- **Example rule**:
  "Based on the provided information and subject to verification, you will be admitted to our university."

Integration with Data Mining and Mathematical Models:

- Business rules in ADS can be derived from:
  - **Expert knowledge and experience**
  - **Patterns discovered via data mining**
- These rules may also be combined with **mathematical models** to improve accuracy and applicability.
- Final decisions may either be:
  - **Fully automated**, or
  - **Forwarded to a human** for confirmation.

Purpose and Suitability:

- Justified in cases where:
  - **Decisions are frequent and repetitive**
  - **Speed and consistency** are crucial
  - **Frontline employees** need to act instantly with **real-time data**
- **Example scenario**:
  A customer service representative instantly offers a loan limit or discount based on automated rules while speaking to a client.

Conclusion:

Automated Decision Systems (ADS) are a modern, efficient way to **automate structured decisions** in business using **predefined rules**. First used in airlines, ADS are now common in finance, retail, marketing, and education. They are ideal for **high-volume, quick-turnaround decisions**, and help organizations **reduce costs**, **improve response time**, and **ensure consistency** in decisions.

✓ Event-Driven Alerts

Definition:

An **event-driven alert** is a **warning or automatic action** triggered when a **predefined event** or an **unusual activity** occurs in a system. These alerts are a **form of Automated Decision System (ADS)**, designed to notify users or systems in real time so that **immediate actions or decisions** can be taken.

Purpose:

- To **monitor systems continuously**
- To **detect anomalies or predefined conditions**
- To **automatically respond** or **notify relevant parties**
- To improve **decision-making speed**, **accuracy**, and **customer service**

Key Characteristics:

- **Triggered by specific conditions or deviations**
- Often based on **historical patterns or predictive models**
- Can be **automated actions** (like offers) or **human alerts** (like fraud verification)
- Integrated with **dashboards** in Business Performance Management (BPM) systems

Examples of Event-Driven Alerts:

1. Credit Card Fraud Detection

- **Trigger**: Unusual or suspicious transaction
- **Example**: A large purchase in a foreign location where the customer has no history of travel.
- **Action**: The system **automatically sends an alert** to the customer for **transaction verification**.

2. Banking – Promotional Offers

- **Trigger**: A customer makes a **large deposit**
- **Example**: If someone deposits 5 lakhs in their savings account.
- **Action**: The system **automatically offers** a **high-interest Certificate of Deposit (CD)** or another **investment option**.

3. Retail Promotions

- **Trigger**: A customer completes a purchase
- **Example**: After buying a smartphone, the system sends an alert offering discounts on accessories like headphones or covers.
- **Action**: A **promotion alert** is automatically sent to the customer.

4. Business Dashboards (BPM)

- **Trigger**: A **Key Performance Indicator (KPI)** deviates from the expected value

- **Example**: Sales fall 20% below the forecast in a specific region.
- **Action**: The dashboard **generates an alert** for the responsible **manager** to investigate and respond.

Conclusion:

Event-driven alerts are a practical application of **Automated Decision Systems (ADS)** that allow businesses to **respond instantly** to important events or anomalies. These alerts are used in various domains like **finance, banking, retail, and performance monitoring**, helping organizations **improve responsiveness, efficiency, and customer engagement**.

✓ Intelligence Creation And Use And Bl Governance

Steps of Intelligence Creation and Use:

The **intelligence creation and use process** in Business Intelligence (BI) is **cyclical** and follows a model similar to military or national security intelligence:

- **Step 1: Identify and prioritize BI projects**
  Based on business needs and objectives.
- **Step 2: Estimate costs and benefits**
  Using **ROI (Return on Investment)** and **TCO (Total Cost of Ownership)** to assess each project.
- **Step 3: Examine decision-making impacts**
  Measure potential benefits such as **cash flow acceleration** and other decision support improvements.
- **Step 4: Perform proper analysis**
  Reliable analysis is only possible if prior steps are well addressed.
- **Step 5: Maintain and monitor the BI application**
  Ensure ongoing support and effectiveness.

What is BI Governance?

BI Governance refers to the **structured process** of **prioritizing, selecting, and managing BI projects** within an organization.

Key Elements:

- Involves **cost-benefit analysis**, **risk management**, and **project interdependency planning**.
- Requires collaboration between:
  - **Middles**: Functional or product area heads who ensure enterprise-wide alignment.
  - **Customers and Providers**: Business-side users and IT professionals.

Typical Responsibilities of BI Governance:

1. Categorize projects (e.g., strategic, mandatory, investment-based).
2. Define selection criteria.
3. Set risk management frameworks.
4. Leverage project interdependencies.
5. Monitor and adjust the project portfolio over time.

What is Intelligence Gathering?

Intelligence gathering in BI refers to the **ethical and legal collection of data** from multiple sources such as:

- Customers
- Stakeholders
- Business processes
- Competitors
- External environment

Key Activities:

- Collecting, cataloging, tagging, filtering, and analyzing large volumes of raw data.
- Transforming data into **usable, actionable information** that improves decision-making and organizational performance.

Modern Intelligence Sources and Techniques:

- **Sensor data (e.g., RFID)**
- **Text mining**
- **Web mining**

Though BI borrows concepts from national security and espionage, it focuses on **legal, transparent, and beneficial** data practices for business improvement.

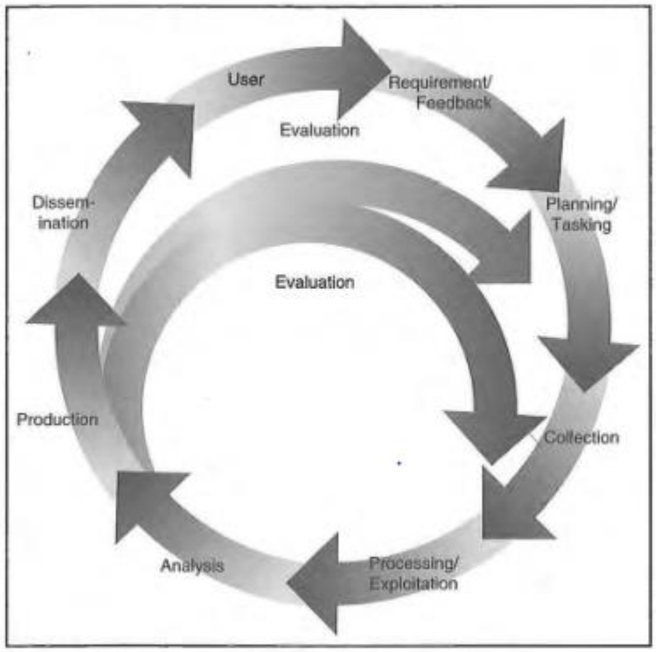


FIGURE 1.5 **Process of Intelligence Creation and Use.** *Source:* L. Krizan,

---

❖ Transaction processing versus analytical processing

Online Transaction Processing (OLTP):

- Supports **routine, operational tasks** such as:
  - ATM withdrawals
  - Bank deposits
  - Cash register transactions
- These systems update **operational databases** in real time.
- Every request is a **transaction** (discrete event like inventory receipt or customer order).
- Operates on a **"real-time, all-or-nothing" basis**—ensures data integrity.

Limitations of OLTP Systems for BI:

- Designed for **fast, efficient processing** of individual transactions—not for reporting or analysis.
- Poor at:
  - Ad hoc reporting
  - Data exploration
  - Decision support
- Business users in the 1980s viewed mainframes as "**black holes**":
  - Data went in, but was hard to retrieve.
  - Reports required **IT staff** to program them.
  - Only **predefined ("pre-canned") reports** were available.

- Even 1990s **ERP systems** (e.g., SAP) improved things only marginally for **non-technical users**.

The Need for BI & Data Warehousing (DW)

Why BI Was Needed:

- To allow **non-technical users** to:
  - Perform **operational reporting**
  - Conduct **interactive analysis**
  - Extract insights from enterprise data
- BI was introduced to solve the **usability and accessibility issues** of OLTP systems.

Role of Data Warehouse (DW) in BI

What is a Data Warehouse?

- A **separate system** from OLTP.
- Stores a **wide variety of integrated data** from multiple sources (including OLTP).
- Intended for:
  - **Querying**
  - **Analysis**
  - **Decision support**

Key Characteristics of DW:

- Contains **historical and current data**, structured for **efficient querying**.
- Presents a **coherent snapshot** of business conditions **at a single point in time**.
- Designed for **Online Analytical Processing (OLAP)**, not transactions.
- Optimized for:
  - Complex queries
  - Trend analysis
  - Business forecasting

OLTP vs. OLAP (BI)

Feature	OLTP (Transaction Processing)	OLAP (BI/Decision Support)
Purpose	Run daily operations	Analyze data for insights
Users	Frontline employees, clerks	Managers, analysts
Operations	Insert, update, delete	Read, aggregate, filter
Speed	Fast for transactions	Fast for complex queries
Data	Current operational data	Historical + current analytical data
System	ERP, CRM, SCM	DW, BI dashboards

Advantages of Separating OLTP and BI (DW):

Improved Performance:

- OLTP is not burdened with heavy analytical queries.
- DW is optimized for **speed and complexity in analysis**.

Better Decision Support:

- Supports both **tactical and strategic decisions**.
- Enables **interactive dashboards**, **reports**, and **real-time insights**.

User Empowerment:

- **Non-technical users** can explore and analyze data without relying on IT.

Competitive Advantage:

- BI systems help in:
    - Detecting market trends
    - Improving operations
    - Gaining **business insights**
    - Making **faster, data-driven decisions**

Conclusion:

- **BI is not OLTP.** It's a separate system focused on **insight and intelligence**, not transactions.
- BI emerged to give organizations the ability to **extract value** from data through **analysis**, not just store it.
- Through **Data Warehouses** and **OLAP**, BI systems transform raw operational data into **actionable business intelligence**, giving organizations a **strategic edge**

OLTP (Online Transaction Processing):

Definition:
Transaction system that is primarily responsible for capturing and storing data related to clay-to-clay business functions.

Key Features:

- Handles **short, atomic transactions**
- Focuses on **data integrity and speed**
- Supports **thousands of concurrent users**
- Data is **current** and frequently updated

- Examples:
    - ATM withdrawals
    - Point-of-sale (POS) systems
    - Online booking systems
    - Order entry systems

OLAP (Online Analytical Processing):

Definition:
An information system that enables the user, while at a PC, to query the system, conduct an analysis, and so on. The result is generated in seconds.

Key Features:

- Used for **querying, reporting, data mining, and forecasting**
- Supports **multidimensional analysis** (slice, dice, drill-down, roll-up)
- Data is mostly **read-only**, structured for quick access
- Focuses on **business trends, performance metrics, and decision-making**
- Examples:
    - Sales trend analysis
    - Profitability reports
    - Customer segmentation
    - Inventory forecasting

In Summary:

Feature	OLTP	OLAP
Purpose	Manage daily transactions	Analyze data for insights
Users	Clerks, cashiers, front-office	Managers, analysts
Data	Current, transactional	Historical, aggregated
Operations	Insert, update, delete	Read, query, summarize
Speed	Fast for write operations	Fast for read/complex queries
System Examples	ERP, CRM, POS systems	Data warehouse, dashboards

---

❖ BI Implementation

Implementing a Business Intelligence (BI) **initiative can be** time-consuming, costly, and prone to failure **if not handled correctly. A critical factor in success is recognizing the** diverse and widespread user community **within an organization.**

Key Points:

- **BI must benefit the entire enterprise**, not just a single department.
- Organizations have **multiple types of users**, ranging from **strategic-level managers** to **tactical-level personnel**.
- These user groups should be **involved early** in the Data Warehouse (DW) investment and planning process.
- The **structure of the DW** and the choice of **BI tools** should reflect the **needs of different user classes**.
- Understanding user diversity helps in **accurate cost-benefit analysis** of BI projects.
- Companies successful in BI recognize and support the **varied roles and requirements** of different BI users.

✓ Appropriate Planning and Alignment with Business Strategy in BI Initiatives

Implementing **Business Intelligence (BI)** successfully requires more than just deploying tools—it must be **closely aligned with a company's overall business strategy**. Below is a structured explanation based on the provided content:

1. Strategic Alignment is Essential

- **BI is not just a technical IT project.**
- It must:
  - Support **strategic and operational business objectives**
  - **Improve business processes**
  - **Transform decision-making** to be more **data-driven**
- BI should change the **way the business operates**, not just how data is stored or accessed.

2. Framework for Planning BI

According to a framework developed by **Gartner (2004)**, successful BI implementation involves coordinated planning in **four key areas**:

1. **Business**
   - Align BI with **business goals and strategy**
   - Ensure BI delivers value to business performance
2. **Organization**
   - Consider **organizational structure and skills**
   - Understand if the **culture is ready for change**
3. **Functionality**
   - Define **functional requirements** for BI tools
   - Identify the **needs of different user groups**
4. **Infrastructure**
   - Assess current **IS (Information Systems) capabilities**
   - Plan integration with **existing IT systems and partner systems**

3. Assessing Readiness and Planning for Change

- Conduct an early **assessment** of:
  - IS team skills
  - End-user skills and diversity
  - Organizational culture and **willingness to adapt**
- Based on this, develop a **detailed action plan**
- Address **integration** of various BI projects and how they will align with other IT/business systems

4. Establishing a BI Competency Center (BICC)

If conditions are right (alignment, readiness, user base), the company should establish a **BI Competency Center (BICC)**. This center serves as the **hub for BI excellence and strategy execution**.

Functions of a BICC (Gartner, 2004):

- Show how **BI supports strategic execution**
- Act as a **bridge** between business users and IS teams
- Store and **share best practices**
- Advocate **BI standards and excellence**
- Help IS teams **understand business needs** and tool requirements
- Ensure **data warehouse flexibility** to meet evolving business needs
- Help **executives see the strategic value** of BI

Conclusion

BI planning must be **intentional, strategic, and organization-wide**. When properly aligned with business goals and supported by a well-prepared organization, BI can lead to **transformational improvements** in decision-making and business performance. The **BI Competency Center** plays a vital role in maintaining this alignment and driving success.

✓ Real-Time, On-Demand BI Is Attainable

With growing demand for **immediate access to information**, especially to bridge the gap between **operational data and strategic goals**, **real-time, on-demand Business Intelligence (BI)** has become not just possible but increasingly necessary.

Why Real-Time BI?

- **Business environments** now demand **faster decision-making**.

- Traditional BI relied on **static, historical data** loaded in batches into a **data warehouse (DW)**.
- However, today's users need:
  - **Live business monitoring**
  - **Instant performance analysis**
  - **Alerts on changes and trends**
  - **Actionable insights in real time**

Key Drivers of Real-Time BI

1. **Technological Advancements:**
   - Tools like **RFID (Radio Frequency Identification)** generate continuous streams of data.
   - These technologies accelerate the **need for real-time data processing and analysis**.
2. **User Expectations:**
   - Users now expect:
     - **Alerts** via e-mail, web, or instant messaging
     - **Instant access** to relevant reports
     - **Event notifications** as they occur
3. **Automated Responses:**
   - Business applications are being **programmed to act automatically** based on insights from real-time BI.
   - **Examples:**
     - **SCM (Supply Chain Management)** systems placing orders automatically when inventory falls below a threshold.
     - **CRM (Customer Relationship Management)** systems triggering action when a large order is placed (e.g., over $10,000).

Two Approaches to Real-Time BI

1. Enhanced Traditional DW-Based Real-Time BI:

- Uses the same architecture as traditional BI but with **faster, near-real-time ETL tools**.
- **Vendors:** Ascential, Informatica
- **Features:**
  - Accelerated ETL (Extract, Transform, Load) processes
  - Updates the **data warehouse more frequently** (not just nightly)
  - Offers **up-to-date reports and analytics**

2. Business Activity Monitoring (BAM):

- Bypasses the data warehouse entirely.
- **Vendors:** Savvion, Iteration Software, Vitria, webMethods, Tibco, Vineyard Software
- **How it works:**
  - Uses **Web services** or **intelligent agents**

    - **Monitors business events** and operations directly from:
        - **Transactional databases**
        - **Separate monitoring servers**
    - Relies on **event- and process-based tracking** to give real-time updates
    - Provides **proactive and intelligent measurement** of operational processes

Conclusion

Real-time, on-demand BI is no longer theoretical—it is **attainable and increasingly essential** for organizations to stay competitive. By either enhancing traditional data warehouse systems with **faster ETL** or adopting **BAM technologies**, companies can monitor, analyze, and act upon data **as events happen**—automating decisions and improving response times.

---

1. Developing or Acquiring BI Systems

- Organizations can **develop**, **purchase**, or **lease** BI systems.
- Most BI applications use **vendor-provided tools or shells** (pre-programmed templates).
- These shells require only **custom data input** and are available through **purchase or lease**.
- Vendors may deliver **custom solutions** or collaborate with **outsourcing partners**.
- The key decision revolves around which option is best for the organization: **build, buy, or lease**.
- **Key criterion** for decision-making: **cost-benefit analysis** and justification.

---

2. Justification and Cost-Benefit Analysis

- As **BI applications multiply**, organizations must **justify and prioritize** each one.
- The challenge: Many benefits of BI are **intangible** (e.g., better decisions, faster response).
- It's necessary to:
    - Identify **both tangible (direct)** and **intangible (indirect)** benefits.
    - Use **case studies** and industry examples to guide benefit estimation.
- **Resources like TDWI (tdwi.org)** provide case studies, product insights, and help in estimating ROI (Return on Investment).

---

3. Security and Protection of Privacy

- BI systems handle **sensitive and strategic business data**.
- Therefore, ensuring **data security** and **privacy protection** is **critical**.
- Special care must be taken to:

    - Protect **employee and customer data**
    - Prevent **unauthorized access or breaches**

---

4. Integration of Systems and Applications

- BI systems must integrate with various existing and external systems:
    - Internal systems: **Databases, legacy systems, ERP, CRM**
    - External systems: **E-commerce platforms, partner systems**
    - Online systems: **Web-based platforms**
- BI tools often need to be **interconnected**, creating **synergy and unified analytics**.
- Integration challenges have prompted vendors to:
    - Expand the **capabilities** of their BI tools
    - Offer **"all-in-one" packages** to reduce compatibility issues

Trade-off:

- **All-in-one solutions**:
    - One vendor = easier support and integration
    - May lack flexibility
- **Best-of-breed solutions**:
    - Combine top tools from multiple vendors
    - More customized, but harder to integrate

Conclusion

Successful BI implementation requires:

- Choosing the right **development/acquisition approach**
- Performing thorough **cost-benefit analysis**
- Ensuring **data security and privacy**
- Achieving smooth **integration** across systems

All of this ensures that BI not only supports business operations but also enhances **strategic decision-making** effectively.

❖ Major tools and Techniques of BI

1. Tools and Techniques in BI

- The way **Decision Support Systems (DSS)** or **Business Intelligence (BI)** are implemented depends heavily on the **tools and techniques** chosen.

- Over the years, a **variety of computerized tools** have been developed to assist with **managerial decision-making**.
- Many tools fall into **major categories** (e.g., reporting, querying, data mining), and some are known by **different names or definitions**.
- These categories and their full details are covered in later chapters of the book (as outlined in Table 1.3).

2. Selected BI Vendors

- There has been a **significant rise** in the number of **BI software providers**.
- Major BI vendors include:
  - **Teradata**
  - **MicroStrategy**
  - **Microsoft**
  - **IBM** (acquired **Cognos** and **SPSS**)
  - **SAP** (acquired **Business Objects**)
  - **Oracle** (acquired **Hyperion**)
  - **SAS**
- The market has seen **many consolidations**, where **larger vendors acquire others** to offer more **comprehensive BI solutions**.
- New companies are also emerging in specialized areas like:
  - **Text mining**
  - **Web analytics**
  - **Advanced data analysis**
- **Collaborations** among companies are becoming more common:
  - Example: **SAS and Teradata** partnered to offer combined **data warehousing and predictive analytics** capabilities.

BI involves a wide **range of tools and vendors** that support decision-making processes. The **choice of tools** and understanding **vendor capabilities** are crucial in implementing effective BI systems tailored to business needs.

TABLE 1.3 Computerized Tools for Decision Support

Tool Category	Tools and Their Acronyms
Data management	Databases and database management system (DBMS)
	Extraction, transformation, and load (ETL) systems
	Data warehouses (DW), real-time DW, and data marts
Reporting status tracking	Online analytical processing (OLAP)
	Executive information systems (EIS)
Visualization	Geographical information systems (GIS)
	Dashboards
	Multidimensional presentations
Strategy and performance management	Business performance management(Corporate performance managemen (CPM)
	Dashboards and scorecards
Business analytics	Data mining
	Web mining, and text mining
	Web analytics
Social networking	Web 2.0
New tools for massive data mining	Reality Mining

Key Terms:-

- **analytics** : The science of analysis.
- **automated decision systems (ADS)**: A business-rulebasecl system that uses intelligence to recommend solutions to repetitive decisions (such as pricing).
- **automated decision support** : A rule-based system that provides a solution to a repetitive managerial problem. Also known as ente1prise decision management (EDM).
- **BI governance** : The process of prioritizing BI projects.
- **business analytics** : The application of models directly to business data . Business analytics involve using DSS tools, especially models, in assisting decision makers. It is essentially OLAP /DSS. See business intelligence (BI).
- **business intelligence (BI)** : A conceptual framework for decision support. It combines architecture, databases (or data warehouses), analytical tools, and applications.
- **business performance management (BPM)** : An advanced performance measurement and analysis approach that embraces planning and strategy. See corporate performance management (CPM).
- **complexity** : A measure of how difficult a problem is in terms of its formulation for optimization, its required optimization effort, or its stochastic nature.
- **corporate portal** : A gateway for entering a corporate Web site. A corporate portal enables communication, collaboration, and access to company information.
- **data** : Raw facts that are meaningless by themselves (e.g. , names, numbers).
- **database** : A collection of files that is viewed as a single storage concept. The data are then available to a wide range of users.
- **data mining** : A process that uses statistical, mathematical, artificial intelligence, and machine-learning techniques to extract and identify useful information and subsequent knowledge from large databases.
- **decision making** : The action of selecting among alternatives.
- **geographical information system (GIS)** : An information system capable of integrating, editing, analyzing, sharing, and displaying geographically referenced information.
- **global positioning system (GPS)** : Wireless devices that use satellites to enable users to detect the position on earth of items (e.g. , cars or people) the devices are attached to, with reasonable precision.
- **information** : Data organized in a meaningful way.
- **information overload** : An excessive amount of information being provided, making processing and absorbing tasks ve1y difficult for the individual.
- **intelligence** : A degree of reasoning and learned behavior, usually task or problem solving oriented.
- **intelligent agent** : An expert or knowledge-based system embedded in computer-based information systems (or their components) to make them smarter.
- **knowledge** : Understanding, awareness, or familiarity acquired through education or experience; anything that has been learned, perceived, discovered, inferred, o r understood; the ability to use information. In a knowledge management system, knowledge is information in action.

- **management science** : The application of a scientific approach and mathematical models to the analysis and solution of managerial decision situations (e.g., problems, opportunities). Also known as operations research (OR).
- **online analytical processing (OLAP)** : An information system that enables the user, while at a PC, to query the system, conduct an analysis, and so on. The result is generated in seconds.
- **Online Transaction Processing (OLTP)** : Transaction system that is primarily responsible for capturing and storing data related to clay-to-clay business functions.
- **predictive analysis** : Use of tools t11at help determine the probable future outcome for an event or the likelihood of a situation occurring. These tools also identify relationships and patterns.
- **predictive analytics** : A business analytical approach toward forecasting (e.g., demand, problems, opportunities) that is used instead of simply reporting data as they occur.
- **user interface** : The component of a computer system that allows bidirectional communication between the system and its user.
- **Web service** : An architecture that enables assembly of distributed applications from software services and ties them together.

Review Questions

1. Define BI (Business Intelligence)

Business Intelligence (BI) is an umbrella term that includes **architectures, tools, databases, analytical tools, applications, and methodologies** aimed at enabling interactive access (often real-time) to data, data manipulation, and effective analysis. Its goal is to help business managers and analysts make more informed and better decisions by transforming **raw data → information → decisions → actions**.

2. Major Components of BI

BI systems have **four major components**:

1. **Data Warehouse**
    - A central repository for current and historical data, structured for analysis and decision-making.
2. **Business Analytics**
    - Tools and techniques (e.g., reporting, data mining, statistical tools) to analyze and discover patterns or insights from data.
3. **Business Performance Management (BPM)**
    - Also called Corporate Performance Management (CPM), it includes planning, forecasting, monitoring, and managing performance based on strategic goals.
4. **User Interface**
    - Dashboards, scorecards, and visualization tools that provide easy access to BI outputs for users across different organizational levels.

3. Typical Applications of BI

Some **common application areas of BI** include:

- General reporting
- Sales and marketing analysis
- Planning and forecasting
- Financial consolidation
- Budgeting
- Profitability analysis
- Statutory reporting
- Real-time performance monitoring
- Predictive analytics (e.g., movie success prediction, market trends)

4. Examples of ADS (Automated Decision Systems)

ADS are **rule-based systems** used to automate repetitive managerial decisions. Examples include:

- **Airlines' yield management** systems to dynamically price tickets based on demand.
- **Loan approval systems**: e.g., "If applicant owns a house and earns over ₹10,00,000/year, approve ₹10 lakhs credit line."
- **Retail pricing systems**: e.g., "If only 70% of seats are sold 3 days before a flight, offer a discount to non-business travelers."
- **University admission systems**: auto-approve based on predefined criteria subject to verification.

5. Examples of Event-Driven Alerts

Event-driven alerts are automated warnings or actions triggered by predefined events. Examples include:

- **Credit card fraud detection**: Alerting customers when a transaction is made in a foreign or unusual location.
- **Banking systems**: Offering high-interest CD (Certificate of Deposit) after a large deposit is made.
- **Retail promotions**: Triggering promotional offers after a certain purchase threshold is crossed.
- **BPM dashboards**: Alerting managers when a key performance indicator (KPI) deviates from its expected range.

6. Steps of Intelligence Creation and Use

The process of intelligence creation and use in BI is **cyclical** and involves several interrelated steps, inspired by military intelligence models:

- **Identification and prioritization** of specific BI projects.
- **Estimation of ROI** and total cost of ownership for each project.
- **Cost-benefit analysis**, including cost of implementation and maintenance.
- **Assessment of decision-making impact**, such as acceleration of cash flow.
- **Execution of analysis**, which converts raw data into decision-supporting information.
- **Continuous monitoring and adjustment** of the BI project portfolio.

Accurate analysis depends on proper execution of all preceding steps in the cycle.

7. What is BI Governance?

BI Governance refers to the structured process of **managing, prioritizing, and overseeing BI projects** to ensure they align with organizational goals. Key points include:

- Involves partnerships between:
  - **Functional/product area leaders** (called Middles)
  - **Customers and providers** (Business and IT sides)
- Addresses:
  - Categorizing projects (strategic, investment, mandatory, etc.)
  - Defining selection criteria
  - Managing risk and project interdependencies
  - Continually updating the project portfolio
- Ensures BI initiatives reflect the **entire organization's needs** rather than sub-optimizing for a single area.

8. What is Intelligence Gathering?

Intelligence Gathering is the ethical and legal process by which companies:

- Collect data from:
  - Customers
  - Business processes
  - Stakeholders
  - Competitors
  - External environments
- Organize and transform that data into **analyzed, actionable information**.
- Use tools like **sensor data (e.g., RFID), text mining, and web mining**.
- Aim to convert massive raw data into meaningful **knowledge** that supports better business decisions.

Though similar in structure to military espionage processes, corporate intelligence gathering is **legitimate, data-driven, and focused on improving business performance**.

9. Define OLTP (Online Transaction Processing):

OLTP refers to systems that support **routine, ongoing business transactions** like ATM withdrawals, bank deposits, or point-of-sale purchases. These systems:

- **Continuously update operational databases** in real time.
- Are optimized for **efficiency in processing many short online transactions**.
- Examples: Updating bank balances, inventory deduction during purchase.

10. Define OLAP (Online Analytical Processing):

OLAP refers to systems that support **decision-making through analysis**. Unlike OLTP:

- OLAP systems are used with **data warehouses**.
- Designed for **ad hoc querying, reporting, and analysis**.
- Focus on **historical and current data**, structured for analysis, not transaction speed.

11. Describe the Major Types of BI Users:

BI user community is diverse, including:

- **Strategic-level users**: Executives and senior managers using BI for long-term planning.
- **Tactical-level users**: Operational or departmental managers using BI for day-to-day decisions.
- Proper BI success requires understanding and involving **all user types** during planning.

12. Implementation Topics Addressed by Gartner's Report:

Gartner's BI implementation framework includes:

- **Business and organizational planning** (define strategic/operational goals).
- **Functionality and infrastructure** alignment.
- Assessment of:
    - **IS (Information Systems) capabilities**.
    - **User skillsets**.
    - **Organizational culture** and **readiness for change**.
- Establishing a **BI Competency Center (BICC)**.

13. Other Success Factors of BI:

- **Clear linkage to business strategy and objectives**.
- **Interaction between business users and IT departments**.
- **Sharing best practices across departments**.
- **Data warehouse flexibility** for changing needs.
- **Executive support and stakeholder involvement**.

14. Why is it Difficult to Justify BI Applications?

- BI applications often deliver **intangible benefits** like improved insights or quicker decisions.
- Hard to **quantify benefits** in monetary terms.
- Requires **knowledge of similar case studies and industry practices** for benchmarking.
- Both **direct and indirect benefits** must be considered.

15. List the Six Major Categories of Decision Support Tools:

As summarized in Table 1.3 (from your source), the main categories are:

1. **Data Warehousing**
2. **OLAP (Online Analytical Processing)**
3. **Data Mining**
4. **Business Performance Management (BPM)**
5. **Dashboards and Scorecards**
6. **Business Activity Monitoring (BAM)**

16. Major Vendors in BI:

Some of the key BI vendors include:

- **Teradata**
- **MicroStrategy**
- **Microsoft**
- **IBM + Cognos + SPSS**
- **SAP + Business Objects**
- **Oracle + Hyperion**
- **SAS**

Also, emerging vendors focus on **text mining**, **Web mining**, and **predictive analytics**, with examples like **SAS-Teradata partnerships**.

2. Data Warehousing

❖ Definition and Concepts

Using **real-time data warehousing (RDW)** with **DSS and BI tools** enhances business decision-making by providing fast, reliable, and integrated insights from large, varied data sources. For example, **DirecTV** uses RDW to monitor operations in real time, quickly detect issues, and offer customers up-to-date service information—creating a competitive edge. RDW ensures decision makers access **accurate, timely, and unified data**, overcoming the challenge of fragmented systems.

<u>Data Warehousing</u> : A physical repository where relational data are specially organized to provide enterprise-wide, cleansed data in a standardized format.

In simple terms, **data warehousing** is the process of collecting, storing, and organizing **current and historical data** from various sources into a central **repository** to support **decision making**.

A **data warehouse** is:

- **Subject-oriented**: Focuses on specific business areas (e.g., sales, finance).
- **Integrated**: Combines data from different sources into a consistent format.
- **Time-variant**: Stores historical data over time for trend analysis.
- **Nonvolatile**: Data is stable and not frequently changed once entered.

It enables **analytical processing** like:

- **OLAP**
- **Data mining**
- **Querying**
- **Reporting**

The ultimate goal is to provide **managers and decision makers** with data that is **organized, reliable, and ready for analysis**.

✓ Characteristics of Data Warehousing (Based on Inmon, 2005):

1. Subject-Oriented

- Data is organized by specific business subjects like **sales**, **products**, or **customers**.
- Only information relevant for **decision support** is included.

- This organization helps users understand **how** and **why** the business is performing in a certain way.
- Unlike operational databases, which are **product-oriented** and focus on transactions, subject orientation gives a **comprehensive view** of the organization.

2. Integrated

- Data from different sources is brought into a **consistent format**.
- The system handles **naming conflicts** and **unit discrepancies** to ensure consistency.
- A data warehouse is assumed to be **fully integrated**.

3. Time-Variant

- Data warehouses store **historical data** and support **time-series analysis**.
- Though some may include current data, the focus is on **long-term trends**, **deviations**, and **comparisons** over time.
- Time is a key dimension, with data collected at various time intervals like **daily**, **weekly**, or **monthly**.

4. Nonvolatile

- Once data is entered, it **cannot be changed or updated** by users.
- Instead, **new data entries** are made, and **obsolete data** is discarded.
- This ensures the system is optimized for **data access** rather than frequent updates.

Additional Characteristics:

- **Web-Based**: Designed for efficient use with web-based applications.
- **Relational/Multidimensional**: May use relational or multidimensional structures for data organization.
- **Client/Server Architecture**: Supports easy access for end-users via client/server systems.
- **Real-Time**: Some warehouses offer **real-time or active data access**.
- **Metadata**: Includes metadata—**data about data**—which explains how data is organized and how to use it effectively.

While a **data warehouse** is the repository, **data warehousing** is the **entire process** that enables decision support, easy access to business information, and business insights.

✓ Data Marts:

A **data mart** is a **smaller, more focused version** of a data warehouse. It is designed to serve the **needs of a specific department or business function**, such as **marketing**, **sales**, or **operations**.

Key Features:

- **Subset of a data warehouse**
- Focuses on **one subject area**
- Provides **department-level access** to data
- Enables faster and simpler analysis compared to a full data warehouse

Types of Data Marts:

1. **Dependent Data Mart:**
    - Extracted directly from an existing **enterprise data warehouse (EDW)**
    - Ensures **data consistency and quality**
    - Supports a **unified enterprise-wide data model**
    - All users view the **same version of the data**
2. **Independent Data Mart:**
    - **Standalone** system, not built from a central data warehouse
    - Created for a specific **strategic business unit or department**
    - Often used by **smaller organizations** due to lower cost
    - May have **inconsistent data** if not well integrated

- **Data marts** offer a more **cost-effective** and **targeted solution** for business analysis.
- **Large companies** typically use **dependent data marts** as part of a unified BI strategy.
- **Smaller companies** or **departments** may rely on **independent data marts** for flexibility and quicker deployment.

✓ Operational Data Stores (ODS):

An **Operational Data Store (ODS)** is a type of **database** that holds **current or near-real-time operational data**, typically used for **short-term, mission-critical decisions**.

Key Features of ODS:

- **Updated Continuously**: Unlike a data warehouse (DW), which is updated periodically, an ODS is **frequently updated** as business operations occur.
- **Short-Term Decision Support**: Used for **daily or real-time decisions**, especially in **mission-critical applications**.
- **Recent Data Only**: Acts like **short-term memory**, storing **very recent, volatile information**.
- **Integrated View**: Consolidates data from **multiple operational systems** into a **single, consistent format**.
- **ETL Processes**: Uses the same **Extract, Transform, Load (ETL)** processes as a data warehouse to gather and prepare data.

ODS vs. Data Warehouse:

Feature	Operational Data Store (ODS)	Data Warehouse (DW)
Data Freshness	Near real-time	Periodically updated
Usage	Short-term, operational decisions	Long-term, strategic decisions
Data Type	Current, volatile data	Historical, stable data
Function	Like short-term memory	Like long-term memory

Related Concept: Oper Mart

- An **oper mart** is a **mini data mart** created from ODS data when there's a need for **multidimensional analysis** of operational data.

An ODS serves as a **bridge** between operational systems and the data warehouse, offering **up-to-date, integrated data** to support **real-time decision making**.

✓ Enterprise Data Warehouses (EDWs):

An **Enterprise Data Warehouse (EDW)** is a **large-scale, centralized data repository** used across the **entire organization** to support **decision-making** and **business intelligence (BI)** efforts.

Key Characteristics of EDW:

- **Enterprise-Wide Scope**: Integrates data from **many departments and systems** across the entire organization.
- **Standardized Format**: Converts diverse source data into a **consistent structure**, enabling unified reporting and analysis.
- **Supports Decision Support Systems (DSS)**: Powers various decision support applications and advanced analytics tools.

Applications of EDW:

EDWs provide data for several strategic systems, such as:

- **Customer Relationship Management (CRM)**
- **Supply Chain Management (SCM)**
- **Business Performance Management (BPM)**
- **Business Activity Monitoring (BAM)**
- **Product Lifecycle Management (PLM)**
- **Revenue Management**
- **Knowledge Management Systems (KMS)**

Benefits of EDW:

- Provides a **"single version of the truth"** for the entire enterprise.
- Enhances **data consistency**, **accuracy**, and **decision quality**.
- Enables **cross-functional insights** and **enterprise-level planning**.
- Supports **real-time** and **historical analysis**.

Example Use Case:

As mentioned, companies like **DirecTV** use EDW to integrate and analyze data from multiple business functions to **gain competitive advantage**, **improve customer service**, and **streamline operations**.

An **EDW** is the **backbone of enterprise-wide BI**, enabling **scalable**, **strategic**, and **data-driven decision making** by integrating all business data into one comprehensive and usable system.

✓ Metadata in Data Warehousing and Business Intelligence

Metadata are essentially **"data about data"**, providing context, structure, and meaning to other data to enable more **effective use** and **understanding**.

Types and Patterns of Metadata:

- **Technical Metadata**: Related to system-level information like data types, structure.
- **Business Metadata**: Provides business context to structured data (e.g., definitions, ownership).
- **Syntactic Metadata**: Describes data format and syntax.
- **Structural Metadata**: Describes data structure (tables, columns, relationships).
- **Semantic Metadata**: Explains the meaning of data in a domain.

Importance & Purpose:

- Enhances **context**, **understanding**, and **knowledge creation** from data.
- Supports **data integration**, **ETL**, **EII**, and **SOA** strategies.
- Enables **interoperability**, **performance**, **versioning**, and **low maintenance** in data systems.

Metadata Strategy:

- Must be **enterprise-wide** and **holistic**, involving:
  - **Ontology & metadata registries**
  - **Enterprise Information Integration (EII)**
  - **ETL processes**
  - **Service-Oriented Architecture (SOA)**

Maturity Levels of Metadata Management (Zhao, 2005):

1. **Ad hoc** – Unstructured, inconsistent metadata usage
2. **Discovered** – Metadata is partially known and explored
3. **Managed** – Metadata is documented and used systematically
4. **Optimized** – Integrated and monitored for improvement
5. **Automated** – Fully embedded and self-maintaining system

Ethical Considerations:

- Involves issues of **privacy**, **intellectual property**, and **data ownership**, especially during design, collection, and dissemination stages.

Metadata are crucial for transforming raw data into usable information and knowledge. A well-defined metadata strategy increases business value, system flexibility, and decision-making effectiveness.

❖ Data Warehousing architecture

1. Basic Information System Architectures for Data Warehousing

➢Client/Server and N-Tier Architectures

- These are architectural models used for designing information systems.
- They separate data processing and presentation across multiple tiers (layers).
- The most common models are:
    - **One-tier** (all functions on a single system),
    - **Two-tier**, and
    - **Three-tier** architectures.

2. Three Main Components of a Data Warehouse System

a) Data Warehouse

- The core repository that stores **cleaned, transformed, and integrated data** from various sources.
- Uses **RDBMS** (Relational Database Management Systems) like Oracle, SQL Server, or DB2.
- Optimized for query processing and analysis, **not transactional updates**.

b) Data Acquisition (Back-End) Software

- Responsible for:
  - **Extracting** data from operational (legacy) systems and external sources.
  - **Cleaning**, **transforming**, and **consolidating** data.
  - **Loading** it into the data warehouse.
- This is often achieved using **ETL (Extract, Transform, Load)** tools.

c) Client (Front-End) Software

- This is the user-facing component.
- Users use **BI/DSS/BA tools** (like Tableau, Power BI, QlikView) to:
  - Analyze,
  - Query,
  - Generate reports,
  - And visualize data.

3. Architectural Models

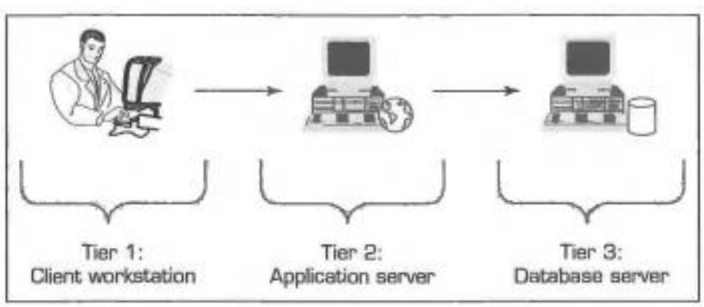
a) Three-Tier Architecture

Tiers:

1. **Operational Systems + Data Acquisition** – Includes source databases and ETL tools.
2. **Data Warehouse** – Central storage of processed data.
3. **Application Server + Client Tools** – Tools for analytics and reporting.

Advantages:

- Clear separation of data processing tasks.
- **Better scalability** and **resource management**.
- Easier creation and use of **data marts** for departmental analytics.



b) Two-Tier Architecture

Structure:

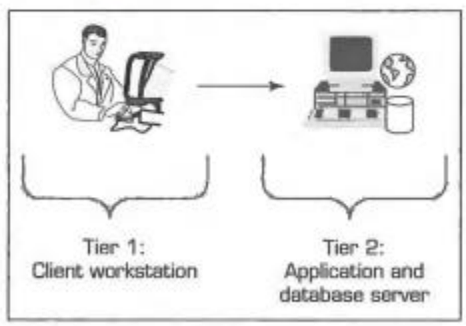
- The **data warehouse and BI tools** operate on the same hardware platform.

Advantages:

- **Cost-effective** and simpler.

Limitations:

- May suffer from **performance issues** for large datasets or complex queries due to resource contention.



Architecture of a Two-Tier Data Warehouse.

4. Web-Based Data Warehousing

➢Three-Tier Web Architecture:

1. **PC Client** – A user with a browser (Java-enabled preferred).
2. **Web Server** – Handles HTTP requests/responses between client and application.
3. **Application Server + Data Warehouse** – Executes queries, runs BI tools, and retrieves data.

➢Communication Layer:

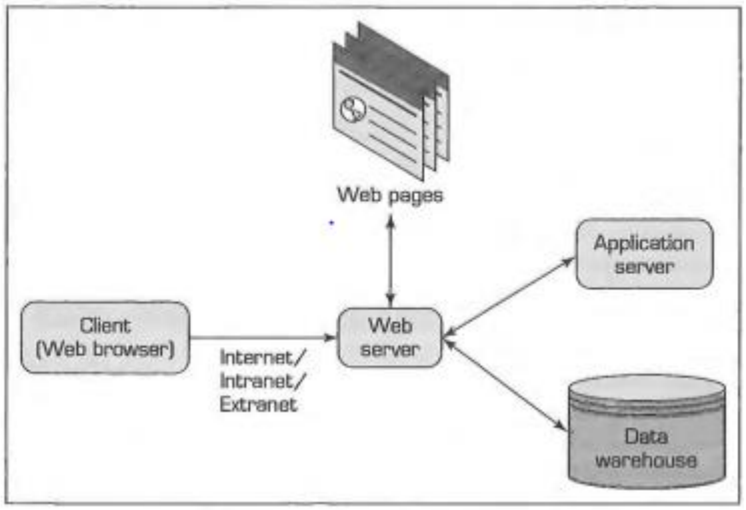
- **Internet/Intranet/Extranet** acts as the medium between clients and servers.

➢Advantages:

- **Ease of access**: Can access through any browser.
- **Platform independence**: Works across operating systems.
- **Lower cost**: No need for proprietary software installation.

➢Real-World Use Cases:

- **Vanguard Group**: Unified internal and customer-facing data access via a web-based architecture.
- **Hilton Hotels**:
  - Migrated to a Web-based 3-tier architecture.
  - Spent $3.8 million, gained **6x speed**, and expected **$4.5–$5 million** annual savings.
  - Used **Dell clustering** for scalability and performance.



5. Key Architectural Design Considerations

a) Choice of DBMS

- Popular options: **Oracle**, **SQL Server**, **DB2**.
- Factors for selection:
  - Scalability,
  - Performance,
  - Integration capability,
  - Vendor support.

b) Parallel Processing and Partitioning

- **Parallel Processing**:
  - Multiple processors handle different parts of the workload simultaneously.
  - Improves speed and scalability.
- **Partitioning**:
  - Dividing large tables into **smaller, manageable segments** for efficient querying.
  - Helps with load balancing and query performance.

c) Data Migration Tools

- Moving data from operational systems to the warehouse can be:

    - **Simple** (if sources are similar), or
    - **Complex** (if sources are diverse and unstructured).
- Decisions:
    - Use **manual methods**,
    - Buy **ETL tools** (e.g., Informatica, Talend, Pentaho),
    - Use **built-in tools** provided by data warehouse vendors.

d) Data Retrieval and Analysis Tools

- These tools assist in:
    - **Querying**,
    - **Visualizing**,
    - **Analyzing** data.
- Options:

1. **In-house development** – Full control, but time-consuming.
2. **Third-party tools** – Specialized and feature-rich.
3. **Built-in tools** – Integrated with warehouse DBMS but may lack advanced features.

For **complex and real-time** analysis, third-party ETL and BI tools are recommended.

Component/Aspect	Description
Architecture Types	One-tier, Two-tier, Three-tier
Three-Tier Benefits	Separation of concerns, scalability, resource optimization
Two-Tier Benefits/Issues	Economical, but may lack performance for large-scale systems
Web-Based Warehouse	Internet-enabled access to data warehouse via browser
DBMS Choices	Oracle, SQL Server, DB2
Parallel Processing	Speeds up data querying by distributing tasks
Partitioning	Splits large tables into chunks for faster access
Data Migration	Tools and processes to move data from source to warehouse
Analysis Tools	Tools for visualization, reporting, and data insights (e.g., BI tools)

✓ Alternative Data Warehousing Architectures - **High-Level Design Categories**

At the top level, **data warehouse architectures** fall into two primary categories:

1. Enterprise Data Warehouse (EDW) Design

- A centralized system for storing and analyzing data across the entire organization.
- Unified and consistent data representation.

2. Data Mart (DM) Design

- Smaller, department-specific systems.
- Tailored to meet specific business unit needs.

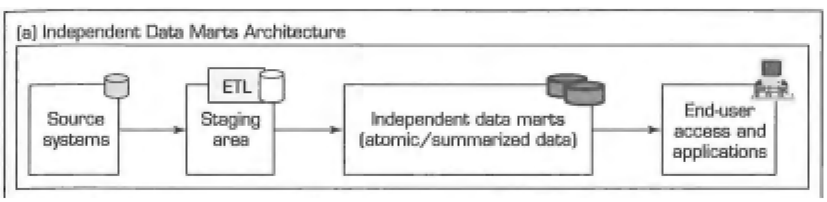
However, many real-world systems are hybrids that fall **between EDW and DM**, leading to new architecture types.

* Alternative Data Warehousing Architectures

(as proposed by **Ariyachandra and Watson**)

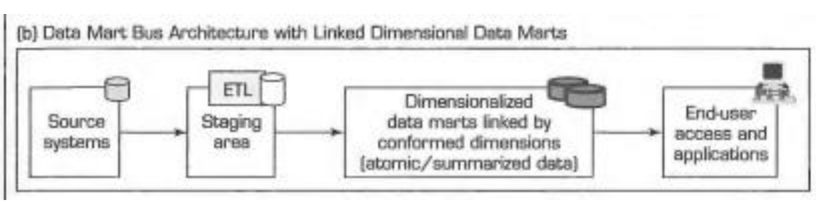
a. Independent Data Marts

- **Description**: Standalone data marts built for individual departments (e.g., sales, HR).
- **Advantages**:
  - Simple and low-cost.
  - Quick to implement.
- **Disadvantages**:
  - **No central control** over data definitions.
  - **Data inconsistency** across marts.
  - Difficult to achieve a unified view ("single version of the truth").



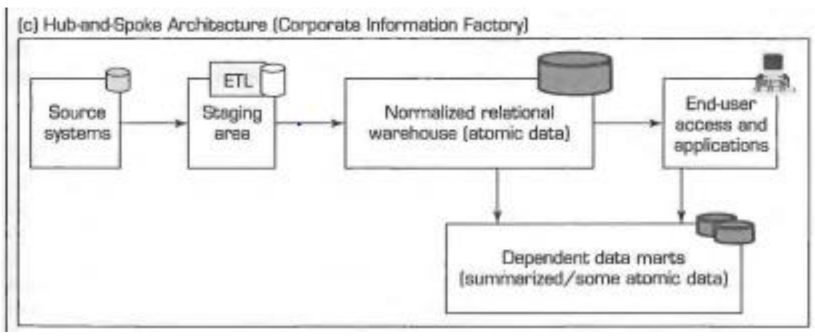
b. Data Mart Bus Architecture

- **Description**: Individual data marts are **linked via middleware** (a bus architecture) that ensures some level of integration, often through **shared dimensions** (conformed dimensions).
- **Advantages**:
  - More **consistent** than independent marts.
  - Supports cross-departmental queries.
- **Disadvantages**:
  - **Performance** may suffer due to query complexity.
  - Middleware adds complexity and maintenance overhead.



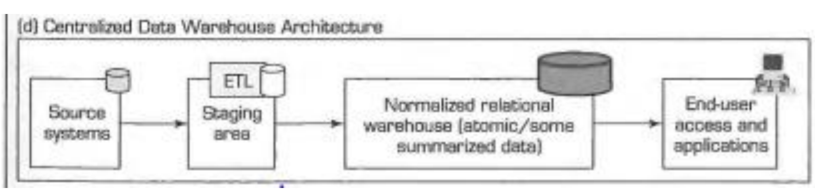
c. Hub-and-Spoke Architecture

- **Description**:
  - **Centralized data warehouse (hub)**.
  - **Dependent data marts (spokes)** for individual departments.
- **Advantages**:
  - Good **scalability** and **modular development** (can be done in phases).
  - Custom interfaces for specific departments.
- **Disadvantages**:
  - May lead to **data latency**.
  - Not a true **enterprise-wide view**.
  - Risk of **data duplication** across marts.



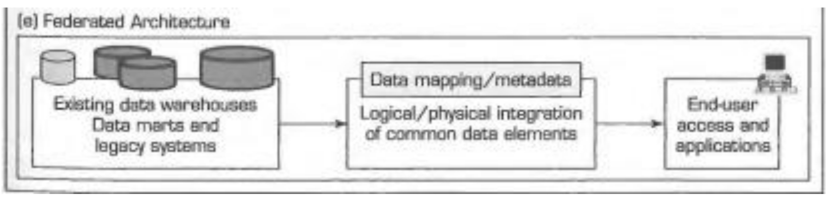
d. Centralized Data Warehouse Architecture

- **Description**: A **single, large data warehouse** used by the entire organization, with **no dependent marts**.
- **Advantages**:
  - **Unified view** of enterprise data.
  - Reduces duplication and simplifies maintenance.
  - Easier **data governance and administration**.
- **Disadvantages**:
  - Complex to design.
  - High **initial cost and effort**.
  - May become a bottleneck if not scaled properly.



e. Federated Data Warehouse

- **Description**:
  - Integrates **existing, diverse systems** without changing them.
  - Uses **middleware and distributed query tools** to access and combine data.
- **Advantages**:
  - **Quick integration** of legacy and new systems.
  - Useful when **data migration isn't feasible**.
  - Offers a **virtual unified view** across sources.
- **Disadvantages**:
  - Often **supplemental**, not a replacement for core warehouses.
  - Suffers from **performance** and **data quality issues**.
  - Complex to maintain real-time consistency.



* Architecture Selection Factors

(**Ariyachandra and Watson, 2005** identified **10 key factors** that influence which architecture to choose):

Factor	Explanation
1. Information Interdependence	How much different departments rely on shared data. High interdependence favors EDW or centralized structures.
2. Upper Management's Information Needs	Architectures must support the reporting and analytics needed by executives.
3. Urgency of Need	If quick results are needed, independent data marts might be preferred due to faster deployment.
4. Nature of End-User Tasks	Whether users need detailed analytics, reporting, or dashboards; more complex tasks may need centralized or federated systems.
5. Resource Constraints	Budget, personnel, and infrastructure limitations can shape which architecture is realistic.
6. Strategic View Before Implementation	Long-term goals (enterprise-wide vs. departmental solutions) heavily affect architecture design.
7. Compatibility with Existing Systems	Some architectures integrate more easily with legacy systems.
8. IT Staff Capabilities	In-house expertise may limit choices (e.g., can't build or maintain a complex federated system).
9. Technical Issues	Scalability, performance, data latency, and tool compatibility are critical.

10. Social/Political Factors	User resistance, internal politics, and management support affect success more than technical aspects alone.

Behavioral and organizational factors (like user involvement and management support) are often more important than technical ones in determining success.

* Effectiveness of Architectures

Architecture	Effectiveness in Practice
Centralized Data Warehouse	Highly effective for enterprise-wide access and consistency.
Hub-and-Spoke	Popular and flexible, but with some redundancy.
Data Mart Bus	Better than independent marts, but may still struggle with performance.
Independent Data Marts	Least effective – lacks consistency and integration.
Federated Architecture	Good as a temporary or supplementary approach.

Each architecture has **specific contexts** where it works best. No one-size-fits-all solution exists. A **hybrid approach** is often adopted depending on organizational structure, budget, urgency, and technical capability.

✓ Which Data Warehouse Architecture Is the Best?

The question of the "best" data warehouse architecture has long been debated in the data warehousing community, especially between the two most well-known proponents:

Two Competing Philosophies:

Expert	Advocated Architecture	Key Concept
Bill Inmon	**Hub-and-Spoke Architecture**	Central warehouse with dependent data marts (Corporate Information Factory).
Ralph Kimball	**Bus Architecture with Conformed Dimensions**	Data marts connected via shared dimensions and facts.

Empirical Study by Ariyachandra & Watson (2006b)

To objectively assess which architecture performs best, the researchers conducted a **Web-based survey** involving:

- **454 respondents** across organizations of various sizes and industries.
- Participants answered questions on architecture type, platform, data warehouse size, and implementation success.

Top Industries Represented:

- **Financial services** was the most represented industry (15%).
- Majority of companies were based in the **U.S. (60%)**.

Most Common Platforms:

Platform	Usage %
Oracle	41%
Microsoft	19%
IBM	18%

Architecture Usage by Popularity

Architecture	% of Respondents Using
Hub-and-Spoke	39%
Bus Architecture	26%
Centralized (No DMs)	17%
Independent Data Marts	12%
Federated	4%

Performance Evaluation Criteria

The study measured architectural success using **four key metrics**, each on a 7-point scale:

Measure	Definition
Information Quality	Accuracy, consistency, and usefulness of the data.
System Quality	Performance, reliability, and ease of use.
Individual Impacts	Effectiveness of the system for individual users.
Organizational Impacts	Contribution to overall organizational performance.

Average Scores by Architecture (Out of 7)

Architecture	Info Quality	System Quality	Individual Impacts	Org Impacts
Bus Architecture	5.16	**5.60**	**5.80**	**5.34**
Hub-and-Spoke	**5.35**	5.56	5.62	5.24
Centralized	5.23	5.41	5.64	5.30
Federated	4.73	4.69	5.15	4.77
Independent DMs	**4.42**	**4.59**	**5.08**	**4.66**

- **Independent Data Marts** perform **worst** across all metrics.
- **Federated Architecture** scores slightly better but still **below average**, likely due to performance and integration issues.

- **Bus, Hub-and-Spoke, and Centralized** architectures perform **similarly well**, with **Bus scoring slightly higher overall**.

Other Findings

Aspect	Observation
Enterprise Scope	**Hub-and-Spoke** most commonly used in **enterprise-wide**, large-scale implementations.
Implementation Cost	**Hub-and-Spoke** is **most expensive and time-consuming** to implement.
Warehouse Size	Larger data volumes tend to favor **Hub-and-Spoke** or **Centralized** models.

Final Verdict: Which Architecture Is Best?

Architecture	Best Suited For	Strengths	Weaknesses
Bus Architecture	Organizations needing **cross-departmental integration** and fast deployment	High user impact, good system quality	May face performance issues for large data sets
Hub-and-Spoke	**Enterprise-wide systems** with scalable, modular design requirements	Central control, scalable, customizable	Expensive, slower to implement
Centralized	Organizations needing **a single source of truth** and **simplified management**	Unified access, consistent data, good performance	Risk of becoming a bottleneck if not optimized
Federated	**Temporary integration** of legacy or distributed systems	Rapid deployment without data migration	Poor performance, complex data management
Independent DMs	**Small-scale, isolated units** needing quick, low-cost solutions	Simple, low-cost	Inconsistent, fragmented, least effective

- **No one architecture is universally "best"** — it depends on:
  - Organization size,
  - Scope of deployment,
  - Budget,
  - Integration needs,
  - Time constraints.
- **Bus and Hub-and-Spoke** architectures are generally most successful across industries.
- **Avoid Independent Data Marts** for long-term strategy.

➢ ETL process, data warehouse development, Comparison of OLTP and OLAP

Due to global competition, ROI demands, investor inquiries, and government regulations, business managers are re-evaluating how they manage and integrate business data. Decision-makers often require access to data from multiple sources, making integration essential.

Before the advent of data warehouses, data marts, and BI tools, accessing and integrating data was a labor-intensive process. Even with modern web-based tools, identifying and delivering the right data remains complex and requires database experts. As data warehouses expand, integration challenges also increase.

Business needs are evolving with factors like mergers, regulatory demands, and new channels driving changes in BI requirements. Users now demand access not just to historical, cleansed, and consolidated data, but also to **real-time**, **unstructured**, and **remote** data. This data must integrate seamlessly with existing data warehouses.

Moreover, new access methods such as personal digital assistants and speech recognition further complicate integration. Many data integration projects span the enterprise. Orovic (2003) noted key practices for successful integration, while Nash (2002) warned that improper integration can lead to failure in large systems like **CRM**, **ERP**, and **supply chains**.

✓ Data Integration: Key Processes

Data integration includes three major processes that make data accessible to ETL tools, analysis tools, and the data warehousing environment:

1. **Data Access**
    - Ability to access and extract data from any data source.
2. **Data Federation**
    - Integration of business views across multiple data stores.
3. **Change Capture**
    - Identifying, capturing, and delivering changes made to enterprise data sources.

– Vendor Tools for Data Integration

- **SAS Institute, Inc.**
    - Provides strong data integration tools.
    - SAS Enterprise Data Integration Server includes customer data integration features to improve data quality.
- **Oracle Business Intelligence Suite**
    - Assists in data integration.

– Purpose of Data Warehousing

- One main purpose is to **integrate data from multiple systems** using various technologies.

– **Technologies Enabling Data Integration**

1. **Enterprise Application Integration (EAI)**
   - Pushes data from source systems into the data warehouse.
   - Focuses on sharing application functionality (not just data).
   - Enables reuse via **application programming interfaces (APIs)**.
   - Often implemented using **Service-Oriented Architecture (SOA)**.
   - SOA involves **coarse-grained services** (well-defined business functions).
   - **Web services** are a specialized form of SOA.
   - EAI supports near real-time data acquisition and delivering decisions to OLTP systems.
2. **Enterprise Information Integration (EII)**
   - Provides **real-time integration** from various sources: relational databases, web services, and multidimensional databases.
   - Pulls data from sources to fulfill information requests.
   - Uses **predefined metadata** to create views that appear relational to users.
   - **XML** plays a key role by allowing flexible tagging of data at creation or afterward.

– **Integration Approaches**

- **Physical Data Integration**
  - Traditional approach used in data warehouses and marts.
- **Virtual Data Integration (enabled by EII tools)**
  - Expands on physical methods by offering flexible, comprehensive data views.
  - Discussed by **Manglik and Mehra (2005)**.

✓ Extraction, Transformation, and Load (ETL): Detailed Explanation

What is ETL?

ETL stands for:

- **Extraction** – Reading data from various source systems.
- **Transformation** – Converting the extracted data into a suitable format for analysis or storage.
- **Load** – Inserting the transformed data into a target system, typically a **data warehouse**.

ETL is at the **core of the technical process** of data warehousing and is **crucial** in any data-centric project.

Time & Effort

- **ETL consumes ~70% of the time** in most data-centric projects.
- It's one of the **most resource-intensive** aspects of data warehousing.

Breakdown of the ETL Process

1. Extraction

- Data is pulled from multiple **heterogeneous sources**, including:
  - OLTP systems
  - ERP/CRM applications
  - Flat files
  - Excel spreadsheets
  - Message queues
  - External databases (e.g., Access)

2. Transformation

- Data is **cleaned, standardized, enriched**, and converted.
- May involve:
  - **Rules application**
  - **Lookup tables**
  - **Combining multiple data sources**
- Handles:
  - Business rules
  - Summarization
  - Standardization (e.g., codes and formats)
  - Calculations
- Can be done through:
  - **Graphical tools** (GUI-based)
  - **Custom code** (PL/SQL, C++, .NET languages)

3. Load

- Data is inserted into **staging tables** first to prepare for loading into the warehouse.
- Then, it's loaded into the **data warehouse** or **consolidated databases**.
- Ensures:
  - **Cleansed data**
  - **Integrated formats**
  - **Consistent metadata usage**

Key Features of ETL Tools

ETL tools perform more than just moving data. They also:

- Transport data between **source and target systems**
- Track and document **metadata** transformations
- Support **process automation** (e.g., scheduling, error handling, logs)
- **Administer runtime operations**

Tool Selection Criteria (Brown, 2004)

A good ETL tool should:

- Read/write from **any number of data sources**
- **Automatically capture and deliver metadata**
- Support **open standards**
- Have an **easy interface** for both developers and functional users

Build vs. Buy Decision

Building In-house

- Flexible and customizable
- Time-consuming and hard to maintain

Buying ETL Tools

- Faster to implement
- Tools may be **expensive** and have a **steep learning curve**
- May **simplify maintenance** and improve **data quality**
- Better support for **scrubbing (cleaning) data**, which is vital for **OLAP** and **data mining**

Example: Motorola Inc.

- Uses ETL to consolidate data from **30 procurement systems** into a **global supply chain data warehouse** for **spending analysis**.

ETL Tool Classification (Solomon, 2005)

ETL technologies fall into four categories:

1. **Sophisticated** – Highly documented and well-managed tools.
2. **Enabler**
3. **Simple**
4. **Rudimentary**

Sophisticated tools result in **better-managed** and **more accurate** data warehouse projects.

ETL and Data Quality

- Extensive ETL may indicate **poor data management**.
- Karacsony (2006): Redundant data increases ETL complexity.
- **Good enterprise data management** reduces ETL needs and improves:
    - **Efficiency**
    - **Maintenance**
    - **Data quality**

Leading ETL Software Providers

- **SAS** (offers integrated ETL + data quality)
- **Microsoft**
- **Oracle**
- **IBM**
- **Informatica**
- **Embarcadero**
- **Tibco**

Conclusion

ETL is a **critical**, resource-heavy process that ensures data is:

- **Accurately extracted**
- **Properly transformed**
- **Cleanly loaded**
  into the data warehouse. Tool selection, process design, and maintenance are vital for a **successful data warehousing strategy**.

➢ Data Warehouse Development

Implementing a **data warehouse** is a **major project** that impacts many departments, systems, and strategic operations (e.g., CRM). It is far more complex than a simple mainframe selection or implementation because of the **multiple input/output interfaces** and **cross-departmental influence**.

Benefits of a Data Warehouse

Direct Benefits

These are **immediate and measurable** advantages of implementing a data warehouse:

1. **Extensive End-User Analysis**
    - Users can perform diverse and complex analyses independently.
2. **Single Version of the Truth**
    - A unified, consolidated view of organizational data eliminates discrepancies across departments.
3. **Improved Information Quality & Timeliness**
    - Processing is shifted from expensive operational systems to cost-effective servers, allowing faster response to queries.
4. **Enhanced System Performance**
    - By moving reporting tasks to the DSS (Decision Support System), operational systems are relieved, boosting overall efficiency.
5. **Simplified Data Access**
    - Data is centralized and more easily accessible to business users.

Indirect Benefits

These are **derived through the use of direct benefits** and have a broader impact on business operations:

- **Enhanced business knowledge**
- **Improved customer service & satisfaction**
- **Better decision-making**
- **Competitive advantage**
- **Support for business process reengineering**

These indirect benefits are considered the **strongest contributors** to **long-term competitive advantage**.

Return on Investment (ROI) Considerations

Given the **substantial cost** and **time investment**, it's critical to **structure the project to maximize success** and carefully weigh costs versus benefits.

Kelly's ROI Approach (2001)

Kelly proposes a categorization of benefits and calculation of Net Present Value (NPV) over the warehouse's lifecycle:

Benefits Categories

1. **Keepers (20%)**
    - Cost savings from improved traditional decision support functions.
2. **Gatherers (30%)**
    - Savings from automation in data collection and distribution.
3. **Users (50%)**
    - Financial gains/savings from better decisions made using the data warehouse.

Costs to Consider

- Hardware
- Software
- Network bandwidth
- Internal development & support
- Training
- External consulting

Key Insight

- Since **users account for 50%** of benefits, they should be **actively involved in the development process**.
- This aligns with the principle that **user involvement is critical** in systems that cause organizational change.

Case Example: Hokuriku Coca-Cola Bottling Company (HCCBC)

- Developed a highly successful data warehouse system.
- Created **intense competitive advantage**.
- Future plans involve expanding the system to cover **over 1 million vending machines** in Japan.

Success Factors for Data Warehousing Projects

To ensure success, an organization must:

1. **Clearly define business objectives**
2. **Gain support from management and end-users**
3. **Set realistic timelines and budgets**
4. **Manage expectations throughout the project**

Developing a Data Warehousing Strategy

A strategy acts as a **blueprint** for successful implementation and must answer:

- **Where** the organization wants to go
- **Why** it wants to go there
- **What** it plans to do once it arrives

The strategy should align with the:

- Organization's **vision**
- **Structure**
- **Culture**

Refer to **Matney (2003)** for steps to create a flexible and efficient support strategy.

Vendor Selection

Once a plan and strategy are in place, organizations should:

- Evaluate **data warehouse vendors**
- Explore **software demos** provided by vendors
- Refer to resources like:
    - **The Data Warehousing Institute (tdwi.com)**
    - **DM Review (dmreview.com)**

Vendor	Product Offerings
Computer Associates (**cai.com**)	Comprehensive set of data warehouse (DW) tools and products
DataMirror (**datamirror.com**)	DW administration, management, and performance products
Data Advantage Group (**dataadvantagegroup.com**)	Metadata software
Dell (**dell.com**)	DW servers
Embarcadero Technologies (**embarcadero.com**)	DW administration, management, and performance products
Business Objects (**businessobjects.com**)	Data cleansing software
Harte-Hanks (**harte-hanks.com**)	CRM products and services
HP (**hp.com**)	DW servers
Hummingbird Ltd. (**hummingbird.com**)	DW engines and exploration warehouses
Hyperion Solutions (**hyperion.com**)	Comprehensive set of DW tools, products, and applications
IBM (**ibm.com**)	DW tools, products, and applications
Informatica (**informatica.com**)	DW administration, management, and performance products
Microsoft (**microsoft.com**)	DW tools and products
Oracle (including PeopleSoft and Siebel) (**oracle.com**)	DW, ERP, and CRM tools, products, and applications
SAS Institute (**sas.com**)	DW tools, products, and applications
Siemens (**siemens.com**)	DW servers
Sybase (**sybase.com**)	Comprehensive set of DW tools and applications
Teradata (**teradata.com**)	DW tools, products, and applications

When selecting data warehouse vendors, **McCloskey (2002)** suggests considering six key factors:

1. **Financial strength**
2. **ERP linkages**
3. **Qualified consultants**
4. **Market share**
5. **Industry experience**

6. **Established partnerships**

Information about vendors can be gathered from **trade shows**, **corporate websites**, or by **requesting product details**.

Van den Hoven (1998) categorized data warehousing products into **three types**:

1. **ETL Tools** – For locating, extracting, transforming, cleansing, and loading data.
2. **Data Management Tools** – Database engines that manage the warehouse and metadata.
3. **Data Access Tools** – Enable end-user analysis, including querying, visualization, OLAP, EIS, and data mining.

These categories together support the **complete lifecycle** of data warehousing.

➢ Data Warehouse Development Approaches

When an organization decides to build a data warehouse for decision support, it typically adopts one of **two major development approaches**, each backed by a renowned expert in the field:

1. The Inmon Approach – *Enterprise Data Warehouse (EDW)*

- **Proposed by**: *Bill Inmon*, often called the "Father of Data Warehousing."
- **Approach**: *Top-down.*
- **Description**:
  - Starts by building a centralized **enterprise-wide data warehouse (EDW)**.
  - Uses traditional **relational database development tools**, such as **Entity-Relationship Diagrams (ERDs)**.
  - Follows a **spiral development approach**, which allows for iterative improvements.
- **Key Points**:
  - Does **not exclude** data marts, but sees them as **offshoots** of the EDW.
  - Focuses on creating a **consistent and comprehensive enterprise view** of data.
  - Ideal for organizations aiming for **data consistency**, high integration, and long-term scalability.
  - More **complex** and **IT-driven**.

2. The Kimball Approach – *Data Mart Strategy*

- **Proposed by**: *Ralph Kimball*.
- **Approach**: *Bottom-up*.
- **Description**:
  - Starts with the creation of **data marts**, which are **department-specific** or **subject-oriented** mini data warehouses (e.g., for sales, marketing).

    - Uses **dimensional data modeling**, beginning with **tables** and focusing on **business processes**.
    - Builds toward enterprise integration via **data bus architecture** and **conformed dimensions**.
- **Key Points**:
    - A “**plan big, build small**” strategy.
    - Easier to implement initially; helps quickly **show business value**.
    - **End-user friendly** and built to deliver faster results.
    - Less complex in design and execution than the Inmon model.

Which Model Is Best?

There is no universal best model.

- The ideal strategy depends on:
    - **User demands**
    - **Business requirements**
    - **Organizational maturity**
- Many organizations:
    - Start with **data marts** to gain experience and deliver value quickly.
    - Evolve into full **enterprise data warehouses** (EDWs) later.
- Data marts are seen as a **practical first step**, but the **EDW** remains the **ultimate goal** for a unified, enterprise-wide data view.

Comparison Table: Inmon vs. Kimball

Characteristic	Inmon (EDW Approach)	Kimball (Data Mart Approach)
Methodology	Top-down	Bottom-up
Architecture	Enterprise-wide atomic data warehouse feeds departmental databases	Data marts model single business processes, integrated via data bus & conformed dimensions
Complexity	Quite complex	Fairly simple
Methodological Base	Spiral methodology; follows traditional RDBMS methods	Four-step process; deviates from RDBMS methods
Physical Design	Fairly thorough	Fairly light
Data Orientation	Subject- or data-driven	Process-oriented
Modeling Tools	Traditional (ERD, data flow diagrams)	Dimensional modeling (departure from relational modeling)
User Accessibility	Low	High
Philosophy	Technical soundness based on proven DBMS methods	Easy querying and fast access for users
Audience	IT professionals	End users
Organizational Role	Integral to Corporate Information Factory	Transformer & retainer of operational data
Goal	Deliver a sound technical	Deliver a user-friendly and efficient

	solution	querying solution

- **Inmon = Top-down, IT-centric, centralized enterprise focus.**
- **Kimball = Bottom-up, user-centric, business unit focus.**
- Both approaches have strengths, and the **best choice depends on the organization's needs**, goals, and readiness. Often, organizations **start with Kimball** and **grow toward Inmon**.

✓ Benefits of Data Marts Toward Enterprise Data Warehouse (EDW)

- **Data marts** offer a **practical starting point** for many organizations, especially when:
  - The organization is **unable or unwilling** to invest in a **large-scale EDW** project.
  - A full EDW seems too complex or expensive initially.
- They can:
  - Show **early feasibility** and **quick wins**.
  - Provide **measurable benefits**, which can help justify future investment in a full **EDW**.
- Once business users recognize the **value of data marts**, they may support expansion into an EDW.

Additional Data Warehouse Development Considerations

Some organizations prefer **not to manage** data warehousing systems internally. They may choose to:

Use Hosted Data Warehouses

- A **third-party vendor** with expertise develops and maintains the data warehouse.
- Benefits:
  - No need to buy **hardware or software**.
  - No burden of managing complex systems internally.
- **Risks/Concerns**:
  - **Security and privacy** issues are a major concern.
  - Sensitive organizational data is hosted by another company.

Representation of Data in a Data Warehouse

Regardless of architecture, the **design of data representation** is crucial for performance and usability.

Dimensional Modeling

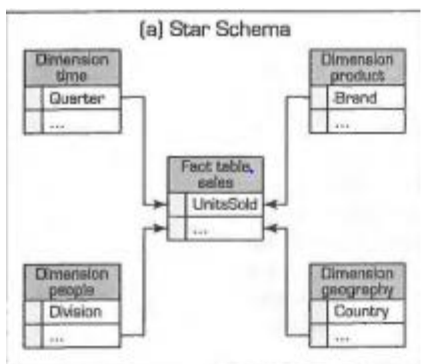
- A **retrieval-based system** optimized for **high-volume query access**.
- It enables complex **multidimensional queries**.

Dimensional modeling is typically implemented using:

- **Star Schema**
- **Snowflake Schema**

Star Schema (Star Join Schema)

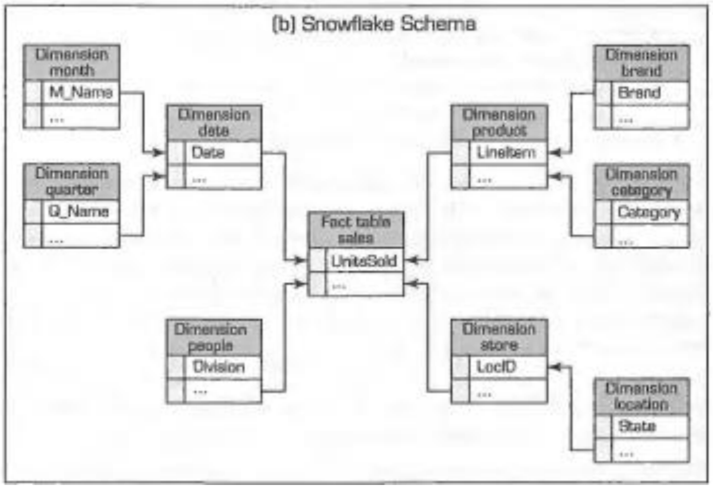
- The **most popular and simplest** form of dimensional modeling.
- Consists of:
  - A **central fact table**:
    - Contains **observed facts**, **metrics**, and **performance measures** (e.g., sales volume, profit margins).
    - Includes **foreign keys** that link to dimension tables.
    - Answers **“what”** is happening in the organization.
  - Several **dimension tables**:
    - Contain **descriptive attributes** to explain and categorize facts (e.g., time, product, location).
    - Have a **one-to-many** relationship with the fact table.
    - Used for **slicing and dicing** the data during querying.
- Benefits:
  - **Fast** query response times.
  - **Simple** structure and easy maintenance.
  - Ideal for **read-only** analytical databases.



Snowflake Schema

- A **variation of the star schema**.
- Also features:
  - A **central fact table**.
  - Multiple **dimension tables**.
- **Difference** from Star Schema:
  - Dimension tables are **normalized** into multiple related tables.
    - In contrast, star schema dimension tables are **denormalized** (flat).
- The snowflake shape results from the **branching structure** of normalized dimension tables.

- More **complex** but can improve **data integrity** and save storage.



➢ Analysis of Data in Data Warehouse

Once data is **properly stored** in a data warehouse, it becomes a **powerful asset** for supporting **organizational decision-making**.

What Is OLAP?

- **OLAP (Online Analytical Processing)** is the most commonly used data analysis technique in data warehouses.
- It enables **fast, ad hoc queries** on large volumes of **multidimensional data**.
- OLAP has gained popularity due to:
  - Rapid **growth in data volumes**.
  - Increased **awareness of the business value** in data-driven decision-making.
- It operates on data stored in **data warehouses** and **data marts**.

OLAP vs OLTP

Aspect	OLTP (Online Transaction Processing)	OLAP (Online Analytical Processing)
Purpose	Handles daily business transactions (ERP, CRM, SCM, POS)	Supports **management decisions** and **business queries**
Data Source	**Normalized** transaction databases focused on **efficiency**	**Non-normalized** data warehouses/marts focused on **accuracy & completeness**
Reporting Type	Routine, **periodic**, narrow-focused reports	**Ad hoc**, **multidimensional**, wide-focused analysis
Resource	Basic relational databases	Requires **large-capacity**, specialized,

Requirements		often multiprocessor systems
Execution Speed	Fast for transaction recording and routine reports	Slower due to **complex, large-scale** queries

- **OLAP and OLTP are complementary**:
  - OLTP **collects and stores** business data.
  - OLAP **analyzes** that data for strategic insights.

OLAP Operations and Concepts

Cube

- The **core structure** in OLAP.
- A **multidimensional array** (real or virtual) for **fast, multi-angle analysis**.
- Designed to **overcome relational database limitations** in performing complex analytical queries quickly.

Common OLAP Operations

1. **Slice**:
   - Extracts a **2D view** from a multidimensional cube.
   - E.g., fixing the "Time" dimension to Q1 2025 to analyze sales by region and product.
2. **Dice**:
   - Performs slicing on **multiple dimensions**.
   - E.g., selecting "Region = South", "Time = Q1", and "Product = Electronics".
3. **Drill Down / Drill Up**:
   - **Drill Down**: Navigate from **summary to detailed** data.
   - **Drill Up**: Move from **detailed to summarized** data.
   - E.g., from yearly sales → quarterly → monthly.
4. **Roll Up**:
   - Computes **aggregations** (e.g., totals, averages) across dimensions.
   - E.g., compute total annual revenue from quarterly data.
5. **Pivot**:
   - Rotates the cube to **reorient dimensions** for different perspectives in reports.
   - E.g., switch rows from "Region" to "Product Category".

Types of OLAP Technologies

1. ROLAP (Relational OLAP)

- Uses **relational databases** as the backend.
- **No pre-computation**: generates **SQL queries** dynamically to retrieve data.
- Can use **summary tables** to optimize performance.
- **Requires careful database design**; OLTP-style databases won't perform well.

Pros:

- Works with large data volumes.
- Scales well with existing relational DBMS.

Cons:

- Slower performance compared to MOLAP.
- Heavily depends on database optimization.

2. MOLAP (Multidimensional OLAP)

- Stores **pre-computed**, aggregated data in **multidimensional cubes**.
- Data is stored in **optimized array-based structures**, not in traditional relational databases.

Pros:

- **Faster query performance** due to pre-processing.
- Ideal for frequent, repetitive analytical queries.

Cons:

- More **storage space** needed.
- **ETL preprocessing** is time-consuming.
- Less flexible with dynamic query structures.

3. HOLAP (Hybrid OLAP)

- **Combines MOLAP and ROLAP**:
  - Part of the data is **pre-computed** and stored (MOLAP).
  - Other parts are stored in **relational form** and queried on-demand (ROLAP).
- Cube designers can sometimes control how the data is partitioned between MOLAP and ROLAP.

Benefits:

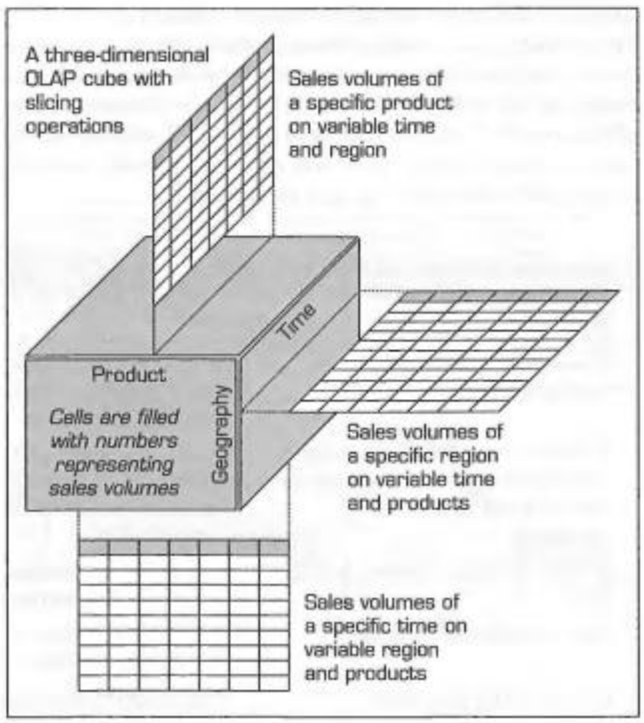
- Optimizes the **trade-offs** of MOLAP and ROLAP.
- Offers **flexibility**, **scalability**, and **performance**.

Summary OLAP Types

Type	Backend Storage	Query Speed	Data Preparation	Flexibility
ROLAP	Relational DB (SQL-based)	Medium/Slow	Low (no pre-computation)	High

MOLAP	Multidimensional arrays	Fast	High (pre-computed cubes)	Low
HOLAP	Hybrid of both	Balanced	Moderate	Moderate to High

- OLAP is central to **analyzing large datasets** in data warehouses.
- It offers **ad hoc analysis**, **flexibility**, and **multidimensional insights**.
- OLAP operations like **slicing, dicing, drilling, and pivoting** enhance exploration of data.
- Choosing between **ROLAP, MOLAP, and HOLAP** depends on the organization's **data volume, query complexity, and infrastructure**.



!.9 Slicing Operations on a Simple Three-Dimensional Data Cube.

- **Differentiation** between the **Data Mart Approach** (Kimball) and the **Enterprise Data Warehouse (EDW) Approach** (Inmon):

Criteria	Data Mart Approach (Kimball - Bottom-up)	EDW Approach (Inmon - Top-down)
Effort	One subject area	Several subject areas
Scope	Limited to a specific department or function	Enterprise-wide coverage
Development Time	Months	Years
Development Cost	$10,000 to $100,000+	$1,000,000+
Development Difficulty	Low to medium	High
Data Prerequisite for Sharing	Common within a business area	Common across the entire enterprise
Sources	Few operational and some external systems	Many operational and external systems
Size	Megabytes to several gigabytes	Gigabytes to petabytes
Time Horizon	Near-current and historical data	Primarily historical data
Data Transformations	Low to medium complexity	High complexity
Update Frequency	Hourly, daily, or weekly	Weekly or monthly
Technology	Workstations and departmental servers	Enterprise servers and mainframe computers
Hardware	Windows and Linux	Unix, Z/OS, OS/390
Operating System	Workgroup or standard DB servers	Enterprise-level DB servers
Databases	Typically departmental DBMS	Enterprise DBMS like DB2, Oracle, etc.
Usage	Used within a department	Used across the organization
Number of Users	Tens	Hundreds to thousands
User Types	Business area analysts and managers	Enterprise analysts and senior executives
Business Spotlight	Focus on optimizing activities within a business area	Focus on cross-functional optimization and decision-making

- **Data Mart (Kimball)**:
  - Quick, cost-effective, focused on individual departments.
  - Suitable for organizations just starting with analytics.
- **EDW (Inmon)**:
  - Holistic, integrated, complex, and high-investment.
  - Ideal for large enterprises aiming for a **single source of truth** across functions.

✓ Data Warehouse Implementation Overview:

Implementing a data warehouse is a complex, enterprise-level initiative that requires careful planning, collaboration, and adherence to best practices.

Key Parallel Tasks for Implementation (Reeves, Solomon):

1. Set service-level agreements and refresh requirements
2. Identify data sources and governance policies
3. Plan for data quality
4. Design the data model
5. Select ETL tools
6. Choose relational DBMS and platform
7. Plan data transport
8. Handle data conversion
9. Define reconciliation processes
10. Plan for purging and archiving
11. Provide end-user support

Success Factors (Hwang & Xu, Ariyachandra & Watson):

- Success is **multidimensional** and should focus on **enhancing user productivity**.
- Key benefits: **faster data access** and **better information quality**.
- Benchmarking is important to evaluate progress compared to other organizations.

TDWI Maturity Model Stages (Eckerson):

1. Prenatal
2. Infant
3. Child
4. Teenager
5. Adult
6. Sage

   ➡ Each stage reflects improvements in scope, analytics, executive support, and value.

Best Practices for Implementation (Saunders, Weir):

- Align the project with **business strategy**.
- Ensure **executive and user buy-in**.
- **Manage user expectations**.
- **Build incrementally**, ensuring **scalability** and **adaptability**.
- Involve both **IT and business teams**.
- Load only **relevant, cleansed, and trusted data**.

- Don’t neglect **user training**.
- Use **proven tools** that align with existing systems.
- Be mindful of **organizational politics and resistance**.

➢ Key Issues & Risks in Data Warehouse Projects

Data warehouse projects are large-scale, resource-intensive, and complex. Because of this, the **risks involved are more severe** than typical IT projects. These issues must be carefully **identified and managed from the beginning** of the project to ensure success.

1. Wrong Sponsorship Chain

- You need **strong executive sponsors** with the power to allocate required resources.
- A **project driver** should be someone respected by peers, decisive, flexible, and realistic about technology.
- A qualified **IT/IS manager** is necessary to lead the project.

2. Unrealistic Expectations

- Phase 1: You promote the data warehouse internally (selling the idea).
- Phase 2: You **struggle to meet the expectations** created in Phase 1.
- Setting **overambitious goals** can lead to frustration and loss of credibility.

3. Political Naivety

- Avoid suggesting that managers aren’t making good decisions—this may offend them.
- Instead, **position the warehouse as a tool to improve access to information**, not as a judgment of current decision-making.

4. Overloading the Warehouse

- Don’t load **all available data**—this creates a “data landfill.”
- Focus on loading **only relevant, quality, and timely data** to avoid inefficiency and system slowdowns.

5. Incorrect Assumptions about Database Design

- **Transactional databases** (OLTP) are normalized and optimized for insert/update operations.
- **Data warehouses** are denormalized, **multidimensional**, and optimized for analysis and querying.
- Treating both the same can lead to performance and design issues.

6. Choosing a Tech-Oriented Manager

- The project leader should be **user-oriented**, focused on delivering business value—not just technology.
- **Understanding user needs** is critical for data warehouse success.

7. Ignoring External or Unstructured Data

- A lot of business-critical data exists outside traditional databases (e.g., **text, images, audio, video**).
- Ignoring this data limits the scope and utility of the warehouse.
- Proper **metadata cataloging** is also essential.

8. Confusing or Overlapping Data Definitions

- Lack of **data standardization and cleansing** leads to conflicting reports and confusion.
- Requires **executive-level involvement** for reconciliation and agreement on definitions.

9. Blind Trust in Vendor Promises

- Vendors often **overpromise** on speed, capacity, and scalability.
- Actual needs often exceed early estimates—**plan for scalability and future growth**.

10. Assuming the Job Is Done Once the DW Is Live

- A data warehouse is **not a one-time deployment**.
- It requires **ongoing iteration**, integration of new data sets, and addition of analytic tools.
- **Annual budgets and continuous efforts** must be planned.

11. Overemphasis on Ad Hoc Reports

- Many organizations start with ad hoc reporting and then create scheduled reports.
- However, **alert systems** (that notify users of critical events in real time) are often **more valuable** and mission-critical.

Success Factors for Data Warehouse Projects

Strong Management Support

- Senior-level commitment is crucial for **budget approval, resource allocation, and conflict resolution**.
- A **high-level champion** can drive momentum and remove organizational barriers.

User Involvement in Data Modeling

- Users help determine:
  - What data is needed
  - What business rules apply
  - What aggregations and calculations are required
- Their involvement in **access modeling** helps define:
  - Data retrieval methods
  - Indexing needs
  - Whether **dependent data marts** are necessary

Web-based Data Warehouses (Webhousing)

- These increase accessibility but come with:
  - **Security concerns**
  - **Access control** challenges for authorized users only
  - Difficulty in quantifying **"hard" (monetary) benefits**

Skilled Implementation Team

- A capable team must understand:
  - **Database technology**
  - **ETL tools**
  - **Source systems**
  - **Development platforms**
  - **Maintenance processes**

✓ Scalability in Data Warehousing

Scalability is crucial for data warehouses due to:

- Rapid data growth (both current and historical)
- Increasing numbers of users
- Rising complexity of queries
- Expansion to support new business functions

A scalable warehouse must scale:

- **Horizontally** (adding more machines)
- **Vertically** (enhancing the capacity of existing systems)

Examples of Massive Data Warehouses

- **Wal-Mart**: Uses a warehouse with **hundreds of terabytes** to analyze sales and inventory.
- **IBM**: Demonstrated a **50-terabyte** warehouse benchmark.
- **U.S. Department of Defense**: Manages a **5-petabyte** warehouse for 9 million military medical records.
- **CNN**: Uses petabyte-level storage for archived news footage.
- **AT&T**: Operates a **26-terabyte** warehouse to detect fraud and support investigations.

Techniques for Improving Scalability

- **Parallel processing**
- **Clever indexing/search schemes**
- **Data distribution across physical stores**
- **Query optimization techniques** (Rosenberg, 2006)

Offshore Development Considerations

- **Offshore outsourcing** for BI and data warehousing is growing (20–25% annually).
- Must address **security** and **cultural differences** when offshoring (Jukic & Lang, 2004).

In essence, as data warehouses grow into the terabyte and petabyte ranges, **scalability** becomes a critical success factor—requiring robust infrastructure, smart data management strategies, and, at times, global development support.

➢ Real - time Data Warehousing

Real-time data warehousing (RTDW)—also known as **Active Data Warehousing (ADW)**—is a data architecture approach where data is continuously or frequently updated in the warehouse **as events happen**, allowing **immediate access** to fresh data for **tactical and operational decision-making**.

This contrasts with traditional data warehouses that are typically updated in **nightly or weekly batches**, making them suitable only for **strategic decisions** based on historical trends.

Why the Need for Real-Time DW?

Due to:

- Massive data volumes
- High update frequency
- Operational decisions requiring **immediate insights**

Traditional systems were too **slow**, as they processed snapshots of past data.

Real-time BI evolved to serve:

- Call centers
- Supply chain managers
- Customer-facing staff
- Logistics operations

How It Works

Instead of the traditional **ETL → ODS → DW** process:

- **Real-time DW directly loads** data from OLTP systems **as it is generated**
- This **bypasses nightly batches** and intermediate systems
- The result: **Up-to-the-minute data availability**

Evolution of DW Systems

Stage	Capability
Level 1	Basic Reporting: “What happened?”
Level 2	Analysis: “Why did it happen?”
Level 3	Prediction: “What will happen?”
Level 4	Action: “Let’s trigger events (campaigns, alerts) as needed”

At the **highest level**, RTDW enables **automatic action** based on real-time events.

Business Applications of RTDW

- **Continental Airlines**: Monitors flight issues in real time
- **UPS**: Used RTDW to reduce delivery miles and fuel usage
- **Egg Bank**: Near real-time refresh of customer data
- **Overstock.com**: Connected users to a real-time data warehouse

Comparison: Traditional vs. Real-Time DW

Feature	Traditional DW	Active (Real-Time) DW
Usage	Strategic only	Strategic + Tactical
Data freshness	Daily/weekly/monthly	Minutes or seconds
Query type	Predefined, periodic reports	Ad hoc, flexible, real-time alerts
User types	Analysts, execs (internal)	Call centers, ops teams, external users
Concurrency	Moderate	1,000+ simultaneous users
Result measurement	Often unclear	Operational results visible

Challenges in Real-Time DW

1. **Architecture Complexity**: Real-time systems need new architectural patterns (not just batch ETL).
2. **Data Modeling**: Models must accommodate fast-changing and detailed data.
3. **Performance Issues**: Concurrent queries + frequent writes = potential slowdowns.
4. **"Multiple Truths" Problem**: Real-time queries run milliseconds apart may return different results.
5. **Scalability and Storage**: High processing power and large storage capacity are essential.
6. **Not All Data Needs Real-Time Updates**: Historical or static data (e.g., old grades) doesn't require real-time refresh.
7. **Versioning Conflicts**: When users view reports generated at slightly different times, results may differ unexpectedly.
8. **Security & Synchronization**: Web-based systems must protect sensitive real-time data and sync metadata accurately.

Key Technologies for RTDW

- **Enterprise Application Integration (EAI)**: Links data flows across systems in real time.
- **Service-Oriented Architecture (SOA)**: Especially using **XML and Web services** to blend strategic and tactical data.
- **Event-driven architectures**: Push events to trigger updates or actions.
- **Parallel processing, real-time ETL**, and **stream processing** engines.

Best Practices for Implementing RTDW

- Plan thoroughly; it's not just "faster ETL"
- Identify which data truly needs real-time updates
- Build scalable, fault-tolerant infrastructure
- Ensure stakeholder buy-in (especially operations & business teams)
- Train users on interpreting fast-changing reports
- Implement proper **data validation**, **alerting**, and **auditing** mechanisms

Real-Time Data Warehousing represents the **next step** in data-driven decision making. Instead of reacting **after the fact**, businesses can now act **in the moment**, boosting:

- **Customer satisfaction**
- **Operational efficiency**
- **Revenue opportunities**

However, success depends on:

- Smart design
- Scalable technology
- Clear understanding of when real-time data is truly needed

➢ Slicing, dicing and cross applications reporting and complex data analysis

1. Slicing and Dicing in Data Analysis

What is "Slicing"?

- **Slicing** refers to **selecting a single layer** (or slice) from a **multidimensional data cube**.
- It produces a **subset of data** for one particular value of a **dimension**.
- Typically results in a **2D table** from a 3D cube.

Example: In a sales data cube with dimensions *Time*, *Product*, and *Region*, slicing by *Time = Q1* gives a 2D table of *Product vs Region* for Q1 only.

What is "Dicing"?

- **Dicing** selects data based on **multiple dimensions** or ranges of values.
- It creates a **smaller cube** by specifying a **sub-cube** with constraints on more than one dimension.
- This is a **more refined filter** than slicing.

Example: In the same cube, dicing for *Time = Q1 & Q2*, *Region = East*, *Product = Electronics* gives a small subcube with only relevant data.

Key Characteristics:

Operation	Dimensions Filtered	Output Shape	Use
Slice	One dimension	2D Table	View single value of a dimension
Dice	2+ dimensions	Sub-cube	Explore multiple dimension combinations

2. Cross-Application Reporting

What Is Cross-Application Reporting?

Cross-application reporting refers to the ability to **gather, combine, and report** data from **multiple systems or applications** into a **unified report**.

Examples of Applications:

- ERP (SAP, Oracle)
- CRM (Salesforce)
- SCM
- HR Systems

- Financial Systems

Scenario: Combine *sales data* from a CRM with *inventory data* from an ERP for a comprehensive supply chain report.

Why It's Important:

- Organizations use **multiple software systems**; data is **siloed**.
- **Decision-makers need a 360-degree view** that involves data across systems.
- Cross-application reporting provides **real-time visibility** across departments.

Technologies Used:

- **ETL Tools** (e.g., Talend, Informatica)
- **Data Warehouses**
- **Middleware or APIs**
- **BI Tools** (e.g., Power BI, Tableau, SAP BO)

Benefits:

- Unified view of business metrics
- Reduces manual data reconciliation
- Increases decision-making speed and accuracy
- Supports **enterprise-wide KPIs and dashboards**

3. Complex Data Analysis

Definition:

Complex data analysis refers to **deep analytical processing** involving **large volumes of structured, semi-structured, and unstructured data**, often involving **advanced techniques**.

Types of Complex Analyses:

Type	Description
Multidimensional Analysis	Analysis across multiple dimensions (e.g., time, geography, customer segments).
Predictive Analysis	Using statistical models/machine learning to forecast future outcomes.
Prescriptive Analysis	Suggests actions based on predictive models.
Sentiment/Text Analysis	Analyzes textual data (e.g., customer feedback, social media).
Pattern Discovery	Identifying trends, clusters, anomalies (e.g., fraud detection).

Tools Involved:

- **OLAP Cubes** for drill-down, roll-up
- **SQL/NoSQL** databases
- **Python/R** for statistical computing
- **AI/ML libraries** (e.g., Scikit-learn, TensorFlow)
- **Data mining platforms** (e.g., RapidMiner, KNIME)
- **Big Data platforms** (e.g., Hadoop, Spark)

Key Techniques:

Technique	Use Case
Drill Down/Up	Move between summary and detailed levels
Pivoting	Rotate data view to analyze from different perspectives
Regression/Classification	Predict outcomes based on inputs
Clustering	Group similar items (e.g., customer segmentation)
Association Rules	Discover relationships (e.g., market basket analysis)

Challenges:

- Data variety (structured vs unstructured)
- Volume and speed (Big Data)
- Need for specialized skills (data science, statistics)
- Integration from multiple sources

Quick Recap:

Topic	Key Idea
Slicing	Selects a single dimension value (e.g., only Q1 data)
Dicing	Filters across multiple dimensions (e.g., Q1 + East + Electronics)
Cross-Application Reporting	Combines data from multiple systems into one report
Complex Data Analysis	Uses advanced tools/techniques to derive deep insights

---

Key Terms:-

– **active data warehousing (ADW)** : The process of loading and providing data via a data warehouse as they become available.
– **ad hoc query** : A query that cannot be determined prior to the moment the query is issued.
– **best practices** : The best methods for solving problems in an organization. These are often stored in the knowledge reposito1y of a knowledge management system.
– **cloud computing** : Information technology infrastructure (hardware, software, applications, and platform) that is available as a service, usually as virtualized resources.
– **cube** : A subset of highly intenelated data that is organized to allow users to combine any attributes in a cube (e.g., stores, products, customers, suppliers) with any metrics in the cube

(e.g., sales, profit, units, age) to create various two-dimensional views, or slices, that can be displayed on a computer screen.

- **data cube** : A two-dimensional, three-dimensional, or higherdimensional object in which each dimension of the data represents a measure of interest.
- **data integration** : In Integration that comprises three major processes: data access, data federation, and change capture. When these three processes are correctly implemented, data can be accessed and made accessible to an array of ETL, analysis tools, and data warehousing environments.
- **data mart** : A departmental data warehouse that stores only relevant data .
- **data quality** : The holistic quality of data, including their accuracy, precision, completeness, and relevance.
- **data warehouse (DW)** : A physical reposito1y where relational data are specially organized to provide enterprisewide, cleansed data in a standardized format.
- **data warehouse administrator (DWA)** : A person responsible for the administration and management of a data warehouse.
- **database management system (DBMS)** : Software for establishing, updating, and querying (e.g., managing) a database.
- **decision support systems (DSS)** : A conceptual framework for a process of supporting managerial decision making, usually by modeling problems and employing quantitative models for solution analysis.
- **dependent data mart** : A subset that is created directly from a data warehouse.
- **dimensional modeling** : A retrieval-based system that supports high-volume query access.
- **dimension tables** : A table that addresses how data will be analyzed.
- **drill down** : The investigation of information in detail (e.g., finding not only total sales but also sales by region, by product, or by salesperson). Finding the detailed sources.
- **enterprise application integration (EAI)** : A technology that provides a vehicle for pushing data from source systems into a data warehouse.
- **enterprise data warehouse (EDW)** : An organizational-level data warehouse developed for analytical purposes.
- **enterprise decision management** : A rule-based system that provides a solution to a repetitive managerial problem. Also known as ente1prise decision management (EDM).
- **enterprise information integration (Ell)** : An evolving tool space that promises real-time data integration from a variety of sources, such as relational databases, \Xfeb services, and multidimensional databases.
- **expert** : A human being who has developed a high level of proficiency in making judgments in a specific, usually narrow, domain
- **extraction** : The process of capturing data from several sources, synthesizing them, summarizing them, determining which of them are relevant, and organizing them, resulting in their effective integration
- **extraction, transformation, and load (ETI)** : A data warehousing process that consists of extraction (i.e., reading data from a database), transformation (i.e., converting the extracted data from its previous form into the form in which it needs to be so that it can be placed into a data warehouse or simply another database), and load (i.e., putting the data into the data warehouse).
- **grain** : A definition of the highest level of detail that is supported in a data warehouse.

- **graphical user interface (GUI)** : An interactive, user-friendly interface in which, by using icons and sinlilar objects, the user can control communication with a computer.
- **independent data mart** : A small data warehouse designed for a strategic business unit or a department.
- **metadata** : a Data about data. In a data warehouse, metadata describe the contents of a data warehouse and the manner of its use.
- **multidimensional analysis** : A modeling method that involves data analysis in several dimensions.
- **multidimensional database** : A database in which the data are organized specifically to support easy and quick multidimensional analysis.
- **multidimensional OLAP (MOLAP)** : OLAP implemented via a specialized multidimensional database (or data store) that summarizes transactions into multidimensional views ahead of time.
- **oper marts** : An operational data mart. An oper mart is a small-scale data mart typically used by a single department or functional area in an organization.
- **operational data store (ODS)** : A type of database often used as an interim area for a data warehouse, especially for customer information files.
- **parallel processing** : An advanced computer processing technique that allows a computer to perform multiple processes at once, in parallel
- **prototyping** : In system development, a strategy in which a scaled-clown system or portion of a system is constructed in a short time, tested, and improved in several iterations.
- **real-time data warehousing (RDW)** : The process of loading and providing data via a data warehouse as they become available.
- **relational database** : A database whose records are organized into tables that can be processed by either relational algebra or relational calculus.
- **Relational Online Analytical Processing (ROLAP)** : The implementation of an OLAP database on top of an existing relational database.
- **risk** : A probabilistic or stochastic decision situation.
- **scenario** : A statement of assumptions and configurations concerning the operating environment of a particular system at a particular time.
- **software agent** : A piece of autonomous software that persists to accomplish the task it is designed for (by its owner).
- **speech recognition** : An area of artificial intelligence research that attempts to allow computers to recognize words or phrases of human speech.
- **SQL (Structured Query Language)** : A data definition and management language for relational databases. SQL frontends most relational DBMS.
- **snowflake schema** : A logical arrangement of tables in a multidimensional database in such a way that the entity relationship diagram resembles a snowflake in shape.
- **star schema** : Most commonly used and simplest style of dimensional modeling