Introduction to Databases and DBMSs

1. Database Characteristics:

Motivation for Learning About Databases:

- **Vital Support for Operations:** Databases are crucial for everyday business activities like processing orders, making reservations, and managing employee payroll.
- **Decision Making:** Databases help with decisions like choosing suppliers, resolving complaints, or opening new locations.
- **Database Management Systems (DBMS):** These systems enable businesses to create, manage, and utilize databases. They have evolved over 40 years and are essential for managing long-term memory in organizations, such as data about customers, employees, products, and orders.

Definition of Data:

- **Raw Facts (Data):** Data comes in two forms:
    1. **Conventional Data:** Names, addresses, birthdates, interest rates, sales amounts, etc. These are basic facts in numbers and text forms that are encountered in everyday interactions.
    2. **Unconventional Data:** Images, videos, maps, and fingerprints. Due to advances in technology, unconventional data, like media captured on phones or product videos, are now managed as data by databases.

Data to Information:

- **Transformation Process:** Data needs to be processed, interpreted, and formatted before it can become useful information for decision-making. A simple example of this transformation would be adding a currency symbol to a raw number to make it understandable as a monetary value.

Defining a Database:

- A database, in a business context, refers to a structured collection of data that meets specific organizational needs.
- **Key Characteristics of a Database:**
    1. **Persistency:** Data in a database is stored permanently, even after the computer or system shuts down.

2. **Inter-related:** Data in the database is connected. For example, the relationship between customers and their orders must be stored.
3. **Shared:** Databases allow multiple users and applications to access and use the data simultaneously. For example, an airline reservation system allows several customers to make bookings at the same time.

University Database Example:

- A university database manages data on students, faculty, courses, and enrollments.
- **Persistency:** This data is stored over a long period.
- **Inter-related:** The database links data like which student is enrolled in which course or which faculty member is teaching which course.
- **Shared:** Many users (students, faculty, administrators) and applications (course registration, grading) interact with the database at the same time.

Entity-Relationship Diagram (ERD):

- **ERD Overview:** This diagram shows the entities (like students, courses) and their relationships (like student enrollment in courses). It is used to visualize how data is connected within a database.
- **Real-world Example:** A real university ERD would be large and complex, possibly covering an entire wall, with hundreds of entities and relationships represented.

Organizational Roles:

Organizational Roles in Database Usage: In organizations, we can engage with databases in two primary ways: as functional users or as information technology (IT) professionals. Functional users typically work in business areas like accounting, marketing, or finance, using databases to perform their day-to-day tasks. IT professionals are responsible for the technical aspects, including the development, design, and maintenance of databases.

Functional Roles vs. Information Technology Roles: Functional users are found in business areas like accounting, marketing, or finance, where they interact with databases to carry out essential business functions. On the other hand, IT professionals handle the technical side of things, ensuring that databases are designed and maintained properly. We should consider whether we prefer working with databases in a business capacity to support decisions, or in a more technical capacity focused on the database infrastructure.

Organization Size—Large vs. Small: In larger organizations, roles tend to be more specialized. For instance, we may find teams or individuals dedicated to specific areas such as database design, maintenance, or usage. This allows for a deeper level of expertise in one area. In smaller organizations, roles are more diverse, meaning we may need to handle both business-related and technical tasks. While this diversity can broaden our skill set, it can also require balancing multiple responsibilities. We should think about whether we are drawn to the specialization that a large organization offers or the role diversity of a smaller one.

Focus on People Skills vs. Technical Skills: We can tailor our career focus based on our personal strengths and preferences. If we are more people-oriented, we may prefer roles that involve collaboration, communication, and project management. On the other hand, if we enjoy working with technology, we might lean towards roles involving database development, data analysis, or software engineering. Alternatively, we may seek a balance between people-oriented tasks and technical responsibilities, where we perform technical work while also interacting with teams and managing projects.

Water Utility Database Example: The instructor's example of a water utility database demonstrates the practical application of databases. This database stored customer details, bills, payments, and meter readings. It also maintained relationships such as linking payments to customers and meter readings to employees. This illustrates how databases are crucial in managing both everyday operations (like billing and meter readings) and long-term relationships (such as customer histories).

Functional Users vs. IT Professionals: Functional users interact with databases in different ways depending on their roles:

- **Indirect users** passively interact with databases by receiving reports or extracted data without directly manipulating the database.
- **Parametric users** actively interact by inputting parameters like date ranges or department names to retrieve information.
- **Power users** are the most skilled, able to create custom reports and access data directly based on the organization's needs.

On the other hand, IT professionals are responsible for building, maintaining, and troubleshooting databases. They play a critical role in ensuring that data is stored and managed correctly.

Role of Developers and Information System Professionals: Developers gather requirements, design applications, and implement information systems. They create the

forms, reports, and interfaces that allow users to interact with databases. Information system professionals ensure that these databases and systems function smoothly, creating views (filtered subsets of data) that help users retrieve the right information efficiently.

Role of Database Administrators (DBAs): Database Administrators (DBAs) support both technical users (IT professionals) and functional users (business users). Their responsibilities include:

- **Technical tasks:** Managing the database to ensure it runs efficiently, backing it up, and troubleshooting issues.
- **Non-technical tasks:** Communicating with users, understanding business requirements, and ensuring that the database meets the organization's needs.

Role of Data Administrators: Data Administrators have broader responsibilities than Database Administrators. They plan how data is used across the entire organization, set policies for data access, and manage both structured (traditional business data) and unstructured data (like images and social media content). While Database Administrators handle day-to-day operations, Data Administrators focus on policy setting and the strategic use of data within the organization.

DBMS overview and database definition feature:

Introduction to Databases and DBMS

- DBMS differs from standard desktop software (e.g., Word Processor, Spreadsheet) in terms of complexity and functionality.
- Databases require planning, while desktop software can evolve dynamically without much initial setup.

What is a DBMS?

- **Definition:** A DBMS (Database Management System) is a collection of components that supports the creation, use, and maintenance of databases.
- Initially focused on efficient storage and retrieval of data, DBMS has evolved to include data acquisition, dissemination, formatting, and more.

Evolution of DBMS

- **Complexity:** DBMS has become increasingly complex due to evolving marketplace demands and product innovation.

- **Continuous Learning:** Mastery of a DBMS can take years and requires constant learning as new features are introduced.

Enterprise DBMS vs. Desktop DBMS

- **Enterprise DBMS:**
  - Supports large databases, thousands of users, and high performance requirements.
  - Common environments include Linux, IBM MVS, and Microsoft Windows Server.
  - Leading providers: Oracle, IBM, Microsoft, SAP, Teradata, PostgreSQL (Open Source).
- **Desktop DBMS:**
  - Supports small databases with fewer users and less stringent performance needs.
  - Dominated by Microsoft Access in small workgroup settings.

Embedded DBMS

- **Definition:** Embedded DBMS are integrated into applications like accounting software and are hidden from end-users.
- **Usage:** These require minimal maintenance and are used in mobile and personal computing devices.

Importance of Database Definition

- **Key Differentiator:** The primary distinction between a DBMS and standard desktop software is the need for defining and planning databases before implementation.
- **Planning:** A diagram showing entities and relationships should be prepared to support the database's intended usage.

Tools for Database Definition

- **SQL (Structured Query Language):** Provides the CREATE TABLE statement to define database tables and their relationships.
- **Graphical Tools:** DBMSs offer graphical tools to augment SQL and help in creating visual diagrams of the database structure.

Database Diagrams

- **Oracle Database Diagram:** Provides a visual representation of interrelated entities (e.g., courses, faculty, students).
- **PostgreSQL Database Diagram (pgAdmin):** Similar to Oracle's, with minor differences in symbols and tools.

Importance of Database Technology

- **Daily Operations:** Databases are vital for supporting e-commerce, billing, and batch operations.
- **Decision Making:** Databases aid in supplier selection, customer complaint resolution, and site location planning.

Need for Planning in DBMS

- **Pre-Deployment Planning:** Essential for successful database deployment, as poor planning can lead to issues later.
- **Refinement:** Although refinement is possible post-deployment, initial planning is crucial.

Non-procedural Access:

Non-Procedural Access

- **Definition of a Query:** A request for data to answer specific questions (e.g., identifying customers with large balances or products with strong sales in a region).
- **Ease of Use:** Non-procedural access enables users with limited computing skills to submit queries without needing to write complex procedures.
- **Focus on Data Retrieval:** Users specify only the parts of the database they want to retrieve, not the implementation details.

Advantages of Non-Procedural Access

- **Simplified Code:** Non-procedural database languages do not include looping statements (like for or while), leading to a significant reduction in code length—up to 100 times less than procedural access.
- **Increased Productivity:** Given that much of business software involves data access, non-procedural access can dramatically enhance software development productivity.

Standard Tool for Non-Procedural Access

- **SQL Select Statement:** The primary tool for non-procedural access in databases.
- **Client Tools:** Most DBMSs provide an SQL client for entering and executing SQL statements, including the select statement.

Client Tools for Database Retrieval

- **Oracle SQL Developer:**
  - Used to connect to Oracle databases and execute SQL statements.
  - Features include:
    - Worksheet tab for executing SQL statements.
    - History of executed statements.
    - Functions for making changes permanent, undoing changes, and erasing window content.
- **pgAdmin for PostgreSQL:**
  - An open-source client for connecting to PostgreSQL databases and executing SQL statements.
  - Includes a query tool for executing SQL and managing statements.

Graphical Tools for Database Retrieval

- **Oracle's Query Builder:**
  - A graphical tool for specifying database retrieval requirements.
  - Displays tables, connections among tables, columns, and criteria.
- **pgAdmin's Limitations:**
  - Does not include a graphical tool for database retrieval, though other clients may offer this feature for PostgreSQL databases.

Limitations of Non-Procedural Access

- **Not Comprehensive for All Applications:** While non-procedural access is powerful, it may not fully support data-intensive applications (e.g., shopping cart web pages, data mining algorithms).
- **Need for Programming Capabilities:** For complex applications requiring database access, DBMSs also provide full programming language capabilities along with embedded non-procedural access.

Database Programming Languages

- **Purpose:** Enable detailed database interactions, such as:

    - Establishing database connections.
    - Executing SQL statements.
    - Using results from executed SQL statements.
    - Associating programming language variables with database columns.
    - Handling exceptions in SQL executions.
- **Examples:**
    - **Oracle:** PL/SQL language.
    - **PostgreSQL:** PL/pgSQL language.

Importance of Database Technology

- **Vital Role in Organizations:** Databases and database technology support daily operations and decision-making processes in modern organizations.
- **Non-Procedural Access Benefits:** Reduces coding requirements related to data retrieval by a factor of 100, significantly enhancing productivity in software development.
- **Skills Development:** Query formulation is a fundamental skill for database work, essential for power users and information systems professionals.

Transaction Processing Overview:

Transaction Processing

- **Analogy with Production Management:**
    - **Production Management:** Involves control of physical goods.
    - **Transaction Management:** Involves control of information goods.
- **Importance of Transaction Management:**
    - Essential for modern organizations like banks, airlines, and universities.
    - Enables reliable and efficient processing of transactions.
    - Large organizations can conduct thousands of transactions per minute.
    - The rise of electronic commerce and mobile computing increases the importance of transaction processing.

Definition of Database Transactions

- **Common Definition:** Interaction among two or more parties for business conduct (e.g., buying a car).
- **Precise Database Definition:**
    - A database transaction is a collection of operations processed as a single unit of work.

    - Transactions are **all or nothing:** all operations must succeed or the entire transaction is canceled.

Key Characteristics of Database Transactions

- **Reliability:**
    - Transactions must be processed reliably to avoid data loss.
    - Prevents interference among concurrent users and system failures.
- **Overhead:**
    - Ensuring no loss of data adds overhead but is crucial for organizational success.

Example: Airline Reservation Transaction

- **Transaction Involves:**
    - User requests flight preferences.
    - Multiple database operations (two updates and one insert).
    - Example pseudocode demonstrates how departure and return are treated as a single transaction.
- **Error Handling:**
    - Utilizes ON ERROR statements to handle failures.
    - Implements a ROLLBACK to revert any partial actions if errors occur.
    - COMMIT ensures that all actions are recorded without data loss.

Example: ATM Transaction

- **Structure Similar to Airline Transactions:**
    - Involves user data (account number, transaction type).
    - Multiple database actions to debit and credit accounts.
- **Error Handling:**
    - Similar to the airline example, utilizes ON ERROR to manage failures.

DBMS Services for Reliable Transaction Processing

- **Concurrency Control:**
    - Ensures concurrent users do not overwrite or cancel each other's actions.
- **Recovery Manager:**
    - Deals with failures to prevent data loss after transactions.

Importance of Transparency in DBMS

- **Definition of Transparency:**
    - In common usage, means invisibility of inner details.
    - In DBMS, it means transaction services' inner workings are invisible to developers.
- **Benefits of Transparency:**
    - Simplifies the developer's job as they don't need to write code for these services.
    - However, these services come with overhead, requiring additional resources (servers, disks, memory).

Data Warehouse Processing Overview:

Levels of Decision-Making

1. Lower-Level Management

- **Focus**: Short-term problems and daily operations.
- **Typical Decisions**:
    - Resolving shipment delays.
    - Scheduling employees.
    - Restocking products.

2. Middle Management

- **Focus**: Implementation of organizational strategies, usually on an annual basis.
- **Typical Decisions**:
    - Forecasting annual sales.
    - Choosing suppliers and contract terms.
    - Determining annual staffing levels.

3. Top-Level Management

- **Focus**: Long-term strategic decisions guiding organizational direction.
- **Typical Decisions**:
    - Identifying new markets.
    - Determining pricing strategies.
    - Choosing locations for new plants and stores.

Operational Databases

- **Purpose**: Support major functions like order processing, manufacturing, accounts payable, and product distribution.
- **Investment Reasons**:
    - Faster processing.
    - Larger business volumes.
    - Reduced personnel costs.

Relationship Between Operational Databases and Decision-Making

- **Data Source**: Operational databases provide raw materials for management decision-making.
- **Value Addition**: Lower-level management uses operational databases for problem-solving; however, middle and upper management require summarized and integrated data.

Importance of Integration

- **Need for Integration**: Operational databases often develop in isolation, failing to meet the information needs for tactical and strategic decision-making.
- **External Data**: Strategic decision-making requires data from outside the organization, such as industry and government datasets.

Data Warehouse Definition

- **Definition**: A data warehouse is a logically centralized data repository where data from operational databases and other sources are integrated, cleaned, and standardized to support business intelligence.
- **Origin**: The term was coined by William Inmon in 1990.

Benefits of Data Warehouses

- **Transformational Activities**: Cleaning, integrating, and standardizing data are essential for deriving benefits.
- **Tangible Benefits**:
    - Increased revenue.
    - Reduced expenses.
    - Improved fraud detection.
    - Enhanced customer retention and targeted marketing.
    - Lower inventory carrying costs through improved demand forecasting.

Processing Types

Transaction Processing

- **Characteristics**:
    - Relies on operational databases with current data at the individual level.
    - Focuses on updating a few records.
    - Primarily concerned with resolving operational problems.

Business Intelligence Processing

- **Characteristics**:
    - Utilizes data warehouses with historical data at both individual and summarized levels.
    - Provides flexibility for various business intelligence needs.
    - Involves querying thousands to millions of records.

Process Orientation vs. Subject Orientation

- **Operational Databases**: Process-oriented, focusing on data relevant to specific business processes (e.g., order entry).
- **Data Warehouses**: Subject-oriented, aggregating data for broader analysis (e.g., all customer data).

Daily Processing and Reporting

- **Business Intelligence**: Requires substantial daily processing, typically during non-peak hours, for data transformations and integrations.
- **Reporting Needs**: Business intelligence applications may query large volumes of data, contrasting with the smaller data volumes typical of transaction processing.

DBMS Technology Evoluation:

Evolution of Database Management Systems (DBMS)

First Generation DBMS

- **Description:** Supported sequential and random searching of files.
- **Characteristics:**
    - Users required to write detailed computer programs to access data.
    - Minimal to no industry standards.

Second Generation DBMS

- **Description:** First true database management systems capable of managing multiple entity types and relationships.
- **Characteristics:**
    - Required computer programs for data access (navigational systems).
    - Some progress on standards, but major companies like IBM ignored them.

Third Generation DBMS

- **Description:** Known as relational DBMSs.
- **Characteristics:**
    - Based on mathematical relations and associative operators.
    - Non-procedural languages made data access more efficient.
    - IBM's research in the 1970s led to the commercialization of relational databases in the 1980s.

Fourth Generation DBMS

- **Description:** Extended the boundaries of database technology.
- **Characteristics:**
    - Supports unconventional data types (images, videos, sounds).
    - Incorporates distributed processing and data warehouse processing.
    - Often referred to as object-relational or object-oriented systems.

Recent Advancements in Database Technology

- **Business Intelligence Processing:** New features for data integration and usage of summary data.
- **Cloud Computing:**
    - Supports on-demand, pay-per-use access.
    - Eliminates fixed costs of software ownership.
    - Established vendors and new startups have developed cloud-based DBMS products.

Big Data Challenges

- **Growth of Data Sources:** Sensors, social media, RFID tags, and digitized content are creating massive data growth.
- **NoSQL Technology:** Developed to handle the challenges of big data.

Market Landscape

- **Enterprise DBMS Market:**
    - Oracle leads with over 40% market share.
    - Competition from Microsoft SQL Server, IBM DB2, Teradata, SAP Sybase, and open-source products.
- **Desktop DBMS Market:**
    - Microsoft Access dominates due to the popularity of Microsoft Office.
    - Open-source DBMS products are increasingly competing at the low end of the enterprise market.

Open-Source DBMS Products

- **MySQL:** The leader in the open-source DBMS market.
- **Competition:** From MongoDB, PostgreSQL, and MariaDB.
- **Cost:** Many organizations report lower costs using open-source DBMS despite needing support contracts.

Popularity Ranking

- **DB Engines Ranking (June 2021):**
    - Top 10 SQL-supporting DBMSs: Oracle, MySQL, Microsoft SQL Server, PostgreSQL, IBM DB2, SQLite, Microsoft Access, MariaDB, Hive, Microsoft Azure SQL Database.