Database Environment and Development Process

Key Concepts and Entities:

- Databases organize data for structured access.
- Metadata describes data characteristics.
- DBMSs manage databases.
- Data warehouses store data for analysis.

Development Process:

- Analysis: Define data needs and database structure.
- Design: Create logical and physical designs.
- Implementation: Build and load data.
- Administration: Maintain and update.

Components of Database Environment:

- Hardware (computers, storage)
- Software (DBMS, operating systems)
- Data
- Users
- Applications (e.g., CRM)

Applications:

- Personal (small databases)
- Two-Tier (client and server)
- Three-Tier (client, application logic, database server)
- Enterprise (large databases)

Advantages of Database Approach:

- Reduced redundancy and inconsistency
- Improved integrity, accessibility, and security
- Enhanced decision-making

Disadvantages of Database Approach:

- High cost and complexity
- Potential for data loss

Databases organize data for easy access and analysis. Databases are managed by DBMSs, and data characteristics are described by metadata.

Database Development Process:

- Analysis: Determine data requirements and database structure.
- Design: Create logical and physical database layouts.
- Implementation: Build the database and load data.
- Administration: Maintain and update the database.

Database Environment Components:

- Hardware (computers, storage)
- Software (DBMS, operating systems)
- Data
- Users
- Applications (e.g., CRM)

Database Applications:

- Personal (small databases)
- Two-Tier (client-server)
- Three-Tier (client-application logic-database server)
- Enterprise (large databases)

Advantages of Database Approach:

- Reduces data redundancy and inconsistency
- Improves data integrity, accessibility, and security
- Enhances decision-making

Disadvantages of Database Approach:

- High cost and complexity
- Potential for data loss
- Requires specialized expertise

Data and Information:

- Data: Representations of entities with significance.
- Information: Data that has been processed and provides knowledge.

Database Approach Benefits:

- Data Independence: Separates data from programs.
- Targeted Redundancy: Manages redundancy to minimize inconsistencies.
- Data Sharing: Allows authorized users to access data through views.
- Application Development: Simplifies development with existing database and tool
- Data: Representations of entities and events with significance.
- Information: Processed data that enhances knowledge.

Disadvantages of File Processing Systems

- Data reliance on programs
- Data duplication
- Limited sharing
- Time-consuming development
- Extensive maintenance

Database Approach

- Data Models: Describe data structure and relationships (e.g., entity-relationship m

- Relational Databases: Link data through common fields in tables (relations).
- Database Management Systems (DBMS): Software for database management.

Advantages of the Database Approach

- Data Independence: Program-data separation.
- Targeted Redundancy: Manages redundancy to minimize inconsistencies.
- Facilitated Data Sharing: Allows authorized access through user views.
- Enhanced Application Development: Simplifies development with existing databas
- Key entities: Customer, Order, Product
- Relationships: Is Placed By (customer-order), Contains (order-product), Is Contained In (product-orc

Key Concepts

- Relational database: Data represented as tables with common values linking related data.
- DBMS: Software system for managing databases.

Advantages of the Database Approach

- Program-data independence: Data structure changes don't affect programs.
- Planned data redundancy: Data stored in a central location, reducing redundancy and improving cor
- Improved data sharing and collaboration: Data accessible to authorized users.
- Increased application development productivity: DBMS tools simplify development.
- Enforcement of standards: Data standards ensure quality and consistency.
- Improved data quality: Integrity constraints and scrubbing processes ensure accuracy.
- Improved data accessibility and responsiveness: Query languages allow flexible data retrieval.
- Reduced program maintenance: Data changes don't require program updates.
- Improved decision support: Tailored databases provide insights for planning and decision-making.

Costs and Risks of the Database Approach

- Need for specialized personnel: Requires database administrators and data management professior
- Installation and management complexity: DBMS software requires specialized skills.
- Conversion costs: Migrating legacy systems can be time-consuming and expensive.
- Backup and recovery needs: Shared databases require comprehensive backup and recovery measu
- <p>Entities and Relationships: Data is organized into entities (e.g., customers, or
- Relational Database: Data is stored in interconnected tables, enabling efficient da
- Database Management System (DBMS): Software that manages and controls da

Advantages of Database Approach

- Data Independence: Data changes do not impact applications.
- Reduced Redundancy: Centralized data storage eliminates duplication.
- Improved Collaboration: Authorized users can access and share data.
- Enhanced Productivity: DBMS tools streamline application development.
- Data Quality Enforcement: Data integrity is ensured through constraints and scrub
- Increased Accessibility: Query languages allow flexible data retrieval.
- Reduced Maintenance: Data modifications do not require program updates.
- Improved Decision Support: Tailored databases provide insights for planning and

Considerations for Database Implementation

- Personnel Requirements: Database administration expertise is essential.
- Complexity: DBMS installation and management require specialized skills.
- Conversion Costs: Migrating legacy systems can be time-consuming and expensi
- Backup and Recovery: Shared databases need comprehensive backup and recov
- Organizational Conflicts: Data ownership and definitions can lead to conflicts.

Database Environment Components

- CASE Tools: Automate database design.
- Repository: Stores data definitions and relationships.
- DBMS: Manages database operations and access.
- Database: Organized data for multiple users.
- Applications: Create, maintain, and retrieve data.
- User Interface: Facilitates interaction with the system.
- Database Administrators: Manage data resources and database design.

Types of Database Applications

- Personal Databases: Small, limited sharing.
- Client/Server Databases: Shared databases with client applications.
- Enterprise Applications: Organization-wide databases for decision-making.

Evolution of Database Systems

- Early DBMSs: Complex and hierarchical.
- Relational Model: Commercial success, SQL as standard language.
- Client/Server Computing: Object-oriented databases emerged.
- CASE Tools: Automate database design.
- Repository: Stores data definitions and relationships.
- DBMS: Manages databases and provides data access.
- Database: Organized data for multiple users.
- Application Programs: Create, maintain, and retrieve data.
- User Interface: Facilitates interaction with system components.
- Database Administrators: Manage data resources and physical database design.

Types of Database Applications:

- Personal Databases: Small, limited sharing.
- Two-Tier Client/Server Databases: Shared databases with client applications.
- Multitier Client/Server Databases: Separate layers for user interface, application,
- Enterprise Applications: Organization-wide databases for decision-making.

Evolution of Database Systems:

- Early DBMSs (1960s): Complex projects.
- Hierarchical and Network Models (1970s): More complex data structures.
- Relational Model (1980s): Commercial acceptance, SQL as data retrieval languag
- Client/Server Computing (1990s): Object-oriented databases introduced.

Key Concepts:

- Databases organize data for access and analysis.
- Data warehouses store historical data for analysis.
- DBMSs manage databases.

Development Process:

- Analysis: Define data needs and database structure.
- Design: Create logical and physical database layouts.
- Implementation: Build and load data.
- Administration: Maintain and update the database.

Components:

- Hardware (e.g., computers, storage)
- Software (e.g., DBMS, operating systems)
- Data
- Users
- Applications (e.g., CRM)

Key Advantages:

- Reduced redundancy and inconsistency
- Improved integrity, accessibility, and security
- Enhanced decision-making

Key Disadvantages:

- High cost and complexity
- Potential for data loss
- Need for specialized expertise

Database Concepts

- Data: Representations of entities with significance.
- Information: Data that has been processed and provides knowledge.

Database Approach Benefits:

- Data Independence: Separates data from programs.
- Targeted Redundancy: Manages redundancy to minimize inconsistencies.
- Data Sharing: Allows authorized users to access data through views.
- Application Development: Simplifies development with existing database and tool
- Standardization: Enforces data standards.

Types of Database Applications:

- Personal (small databases)
- Two-Tier (client and server)
- Three-Tier (client, application logic, database server)
- Databases: Integrated data repositories supporting decision-making.

- Data Warehouses: Historical data stores for analysis and reporting.
- DBMS: Software managing database access and control.

Important Examples and Figures

- Table 1-5: Database application uses.
- Figure 1-8: Database technology evolution.
- Figure 1-9: Business function-data entity mapping.

Relationships between Ideas

- Database development involves a systematic process from data modeling to implementation and ma
- Enterprise data modeling defines data requirements, which are refined through logical and physical
- Physical design determines data storage methods, impacting performance and efficiency.
- Database maintenance ensures alignment with business needs and error correction.
- Databases: Store integrated data for decision-making.
- Data Warehouses: Historical data repositories for analysis.
- DBMS: Software that manages database access and control.

Database Development Process:

- Involves systematic steps from data modeling to implementation and maintenance.
- Data modeling involves defining data requirements, logical and physical design, and storage metho
- Physical design impacts performance and efficiency.
- Maintenance ensures alignment with business needs and error correction.

Conceptual Data Modeling:

- Represents data requirements using entities and relationships.
- Identifies organizational data and ensures consistency.
- Creates a detailed, technology-independent data specification (conceptual schema).

Logical Database Design:

- Represents data using implementation technology.
- Minimizes redundancy and improves integrity through normalization.
- Includes tables, columns, rows, and constraints.

Physical Database Design:

- Specifies data storage in secondary memory.
- Optimizes storage and retrieval using file organizations.
- Includes concepts like record organization, file organizations, and indexes.

Database Implementation:

- Involves programming and data loading.
- Uses languages like Java, SQL, and nonprocedural languages.

Database Maintenance:

- Maintains and optimizes the database.
- Includes tasks like adding, deleting, and rebuilding elements.

Other Concepts:

- Different IS development approaches include SDLC, RAD, Prototyping, and Agile Development.
- Three-Schema Architecture provides different views of the database.
- Entities and relationships form a high-level representation of data requirements (Figure 1-3a).
- Databases: Store integrated data for decision-making.
- Data Warehouses: Historical data repositories for analysis.
- DBMS: Software managing database access and control.

Database Development Process:

- Involves data modeling, implementation, and maintenance.
- Data modeling includes conceptual, logical, and physical design.
- Physical design impacts performance and efficiency.
- Maintenance ensures alignment with business needs and error correction.

Data Modeling:

- Conceptual: Entities and relationships represent data requirements.
- Logical: Data represented in implementation technology, minimizing redundancy.
- Physical: Data storage in secondary memory, optimizing storage and retrieval.

Alternative Development Approaches:

- SDLC, RAD, Prototyping, and Agile Development.

Database Architecture:

- Three-Schema Architecture: Physical, Logical, and View Schemas.

Project Management:

- Team includes analysts, architects, users, and project managers.

Key Entities and Relationships:

- Entities (e.g., Customer, Product) and relationships (e.g., Places, Contains, Has).

Data Warehousing:

- Stores historical data for analysis using OLAP.

Case Study: Pine Valley Furniture Company:

- Data model developed for marketing support system using a prototyping approach.

Database Design:

- Relational database tables created using SQL.
- Entities represented by tables, attributes by columns.
- Physical, Logical, and View Schemas

Project Management

- Project: Undertaking with a defined scope, timeline, and leadership.
- Team Members: Business analysts, database architects, users, project managers

Database Development

- Example: Pine Valley Furniture Company's transition to a database approach.
- Entities: CUSTOMER, PRODUCT, EMPLOYEE, etc.
- Relationships: Places, Contains, Has

Relational Database

- Tables represent entities, with columns as attributes and rows as instances.
- Primary Keys: Ensure unique identification.

Data Warehousing

- Data Warehouses: Store historical and summarized information for decision supp
- Online Analytical Processing (OLAP): Facilitates data analysis.

Pine Valley Furniture Company Case Study

- Helen Jarvis: Requested sales data for analysis.
- Chris Martin: Systems analyst assigned to develop a marketing support system.
- Prototyping and Life-Cycle Approach: Extracted data and built a preliminary data

Data Model Analysis

- Chris presented the data model to Helen, who provided additional requirements.

Database Design

- Chris translated the data model into relational database tables using SQL.

Concepts and Architecture:

- Three-Schema Architecture (Physical, Logical, View Schemas)
- Relational Database (Tables, Columns, Rows, Primary Keys)

Project Management:

- Project scope, timeline, and leadership
- Team members (analysts, architects, users, managers)

Data Modeling:

- Entities (e.g., Customer, Product)
- Relationships (e.g., Places, Contains, Has)

Data Warehousing:

- Stores historical and summarized data for analysis
- Supports Online Analytical Processing (OLAP)

Case Study: Pine Valley Furniture Company

- Data model developed for marketing support system
- Data extracted and analyzed using a prototyping approach

Database Design:

- Data model translated into relational database tables
- SQL used to define tables and relationships

Key Entities and Relationships:

- Entities: Customer, Order, Invoice, Product, Product Line
- Relationships established through foreign keys

Process:

- Data requirements identified
- Data model developed and analyzed
- Database designed and implemented
- Entities: Customer, Order, Invoice, Payment, Product, Product Line
- Examples: Home Office product line marketing support system

Important Examples and Figures

- SQL definitions of ProductLine and Product tables (Figures 1-16, 1-17)
- Data model for Home Office system (Figures 1-15, 1-18)
- Sales-to-goal comparison query (Figure 1-19)
- Product line sales comparison (Figure 1-20)

Relationships between Ideas

- Entities are represented by tables, attributes by columns.
- Relationships between entities are established through foreign keys.
- Database development includes planning, analysis, design, implementation, and maintenance.

Summary

Analyzing Requirements

- Data entities and attributes for Home Office system were identified.
- Data model was translated into SQL tables.

Designing the Database

- Attributes were formatted, properties specified.
- Attributes were indexed to enhance performance.

Using the Database

- Ad hoc queries and reports were utilized.
- Prewritten routines simplified data access.

Administering the Database

- Weekly data downloads from operational databases were scheduled.

Database Concepts

- Data: Representations of entities with significance.
- Information: Processed data that provides knowledge.

Data Models

- Entities and Relationships: Data is represented as objects and the relationships b
- Data Models: Describe data structure and relationships (e.g., entity-relationship m
- Relational Databases: Link data through common fields in tables (relations).

Advantages of Database Approach:

- Data Independence: Separates data from programs.
- Targeted Redundancy: Manages redundancy to minimize inconsistencies.
- Facilitated Data Sharing: Allows authorized access through user views.
- Enhanced Application Development: Simplifies development with existing databas

Databases are vital for managing organized data and metadata. They offer advantages over file systems, including data independence, sharing, reduced redundancy, and improved productivity.

Database Development:

Database development involves creating schemas (conceptual, internal, and external) to provide different perspectives on the data. Enterprise data modeling and the systems development process guide this process.

Data Models:

Data models represent real-world entities, attributes, and relationships. Key concepts include entities (e.g., Patient), attributes (e.g., Patient's name), and relationships (e.g., Physician treats Patient).

Data Model Types:

Key Figures:

- Figure 1-17: Example of a product table structure
- Figure 1-18: Identifying important customers using entities

Key Concepts:

- Entity: Real-world object (e.g., Patient)
- Attribute: Property of an entity (e.g., Patient's name)
- Relationship: Association between entities (e.g., Physician treats Patient)

Entities:

- Patient, Physician, Order, Service/Drug, Medical/Surgical Item, Supply Item, Vendor, Facility, Ward,

Relationships:

- Physician treats Patient
- Physician diagnoses Patient
- Patient assigned to Ward
- Patient uses Medical/Surgical Items
- Vendor supplies Medical/Surgical and Supply Items
- Physician writes Orders for Patient
- Facility maintains Diagnostic Units and Wards
- Ward staffed by Staff members
- Facility staffs Physicians

Data Model Types:

- Enterprise model: High-level overview of organization's entities and relationships
- Conceptual model: Logical structure of data
- Logical model: Physical structure of data (data types, tables)
- Physical model: Actual storage details (page sizes, storage devices)

Key Figures:

- Figure 1-17: Product table structure with ProductID field
- Figure 1-18: Entities for identifying most important customers

Databases are essential for managing data and metadata efficiently. Compared to file systems, they offer data independence, enhanced sharing, reduced redundancy, and increased productivity.

Database Development and Schemas:

Database development involves creating schemas (conceptual, internal, and external) to provide different perspectives on the data. Enterprise data modeling and the systems development process guide this process.

Data Models and Types:

Data models represent real-world entities, attributes, and relationships. Different data models exist, including enterprise, conceptual, logical, and physical models.

Key Concepts and Entities:

Key concepts include entities (real-world objects), attributes (properties of entities), and relationships (associations between entities). Entity types commonly used in hospital systems include facility, physician, patient, service/drug, and supply item.

Relationships between Ideas:

- Cost-saving factors (CSFs): Reduce expenses and increase efficiency.
- Business functions: Includes patient care, clinical services, and financial manager
- Entity types: Facility, physician, patient, service/drug, and supply item.

Important Examples and Figures

- Figure 1-2: Illustrates the breakdown of business functions.
- Figure 1-3: Shows the preliminary enterprise data model.
- Figure 1-4: Provides examples of relational database tables.
- Figure 1-5: Displays a sample patient bill.

Relationships between Ideas

- CSFs guide the selection of business functions.
- Business functions are organized hierarchically, with high-level functions subdivided into specific tas
- Entity types capture data needed for different business functions.
- Business rules establish relationships between entities.
- Relational database technology underlies many hospital computer applications.
- Enterprise data modeling offers a comprehensive view of organizational data.