MAA OMWATI DEGREE COLLEGE HASSANPUR

(PALWAL)

NOTES

B.Com 4th SEM

DATA BASE MANAGEMENT SYSTEM-II

Unit-1

🔹 1. Data and Information

👉 What is Data?

Data is raw, unorganized facts that need to be processed to become meaningful. It can be in the form of numbers, characters, symbols, or images. For example:

- A list of numbers: 45, 78, 12
- Names: Alice, Bob, Carol

Characteristics of Data:

- It is unprocessed.
- It lacks context.
- It can be structured or unstructured.

👉 What is Information?

Information is processed data that is meaningful and useful for decision-making.

Example:

- Raw data: Sales numbers – 100, 200, 150
- Information: "Total sales for the month is $450."

Difference Between Data and Information: | Data | Information | |----|----------| | Raw facts | Processed and meaningful | | No context | Has context | | Unorganized | Organized and structured |

🔹 2. Data Processing

Data Processing is the transformation of raw data into meaningful information through a series of operations such as collection, validation, sorting, storing, analyzing, and output.

👉 **Stages of Data Processing:**

1. **Collection**: Gathering data from various sources.
2. **Preparation**: Cleaning and organizing data.
3. **Input**: Entering data into a system.
4. **Processing**: Performing calculations or operations.
5. **Storage**: Saving data for future use.
6. **Output**: Presenting processed data as information.
7. **Feedback**: Revising data and improving systems.

🔹 3. Database Technologies: Introduction

A **database** is a structured collection of data stored in a computer system, managed using a **Database Management System (DBMS)**.

A DBMS allows users to:

- Insert, update, and delete data.
- Query data using languages like SQL.
- Ensure security, backup, and data integrity.

👉 **Types of Database Technologies:**

- Relational Databases (MySQL, Oracle, PostgreSQL)
- NoSQL Databases (MongoDB, CouchDB)
- Cloud Databases (Firebase, AWS RDS)
- In-memory Databases (Redis)

- Graph Databases (Neo4j)

🔹 4. Internet Databases

Internet databases are databases that are connected to web applications and accessible through the internet.

👉 **Features:**

- Web-based interfaces
- Remote access
- Cloud integration
- Real-time querying

Examples: Online shopping platforms, Google Drive, Dropbox.

🔹 5. Web Technology and Web Databases

Web Technology refers to tools and techniques used to communicate via the internet, such as HTML, CSS, JavaScript, and server-side scripting like PHP or Node.js.

Web Databases are databases that power web applications. They allow dynamic content creation, user authentication, and data-driven interfaces.

Example:

- A blog site where content is stored in a MySQL database and displayed using PHP.

🔹 6. Digital Libraries

A **digital library** is a collection of digital resources accessible via electronic means. It includes eBooks, journals, theses, and multimedia content.

👉 **Features:**

- Digitized collections
- Search and indexing
- Remote access
- Metadata management

Examples: Project Gutenberg, IEEE Xplore, JSTOR.

🔹 7. Mobile Databases

Mobile Databases are databases designed for use on mobile devices like smartphones and tablets.

👉 **Characteristics:**

- Lightweight (e.g., SQLite)
- Offline support
- Synchronization with server
- Limited memory usage

Examples: SQLite in Android, Realm DB, Firebase Realtime DB.

🔹 8. Data Independence Concepts

Data Independence refers to the ability to change the schema at one level of a database system without altering the schema at the next higher level.

👉 **Two Types:**

1. **Logical Data Independence**:
    - Ability to change the logical schema (tables, attributes) without changing the application programs.
2. **Physical Data Independence**:
    - Ability to change the physical storage of data without affecting the logical schema.

Importance: Enhances data abstraction, reduces maintenance, and improves flexibility.

🔹 9. Physical Data Organization: Introduction

Physical Data Organization deals with how data is stored on storage devices like hard disks, SSDs, or RAID systems.

It involves:

- File structures
- Indexing
- Storage blocks
- Access paths

🔹 10. Physical Storage Media

👉 Types of Storage Media:

1. **Primary Storage**:
    - RAM, Cache – fast but volatile
2. **Secondary Storage**:
    - Hard drives, SSDs – non-volatile, slower
3. **Tertiary Storage**:

- Optical disks, backup tapes

4. **Cloud Storage**:
    - Online services like Google Drive, AWS S3

🔹 11. RAID Technology (Redundant Array of Independent Disks)

RAID is a storage technology that combines multiple hard drives to improve performance, reliability, and fault tolerance.

👉 Types of RAID:

RAID Level	Description	Features
RAID 0	Striping	High speed, no redundancy
RAID 1	Mirroring	Redundancy, no performance gain
RAID 5	Striping with parity	Balanced performance and redundancy
RAID 10	Combination of 1 and 0	High performance and fault tolerance

✅ Advantages:

- Improved data access speed
- Fault tolerance
- Data redundancy
- Scalability

❌ Disadvantages:

- Costly (multiple disks required)
- Complex management

- Risk of data loss in lower RAID levels

🔹 12. E-R (Entity-Relationship) Model

The **ER Model** is a high-level conceptual data model used to design database schemas by identifying entities, relationships, and attributes.

👉 **Components:**

1. **Entity**: A real-world object (e.g., Student)
2. **Attribute**: Property of an entity (e.g., Name, ID)
3. **Relationship**: Association between entities (e.g., Enrolls)

👉 **Types of Attributes:**

- Simple vs Composite
- Single-valued vs Multivalued
- Derived attributes

👉 **Types of Relationships:**

- One-to-One
- One-to-Many
- Many-to-Many

🔹 13. Conversion of ER Model into Relations

To implement an ER model in a relational database, we convert entities and relationships into **tables**.

👉 **Steps:**

1. **Entity to Table**:
    - Create a table for each entity with a primary key.

2. **Attributes**:
    - Simple attributes become columns.
    - Composite attributes are split.
    - Multivalued attributes require a new table.
3. **Relationship to Table**:
    - One-to-One: Add foreign key to one table.
    - One-to-Many: Add foreign key in "many" side.
    - Many-to-Many: Create a junction table with foreign keys from both entities.

🧩 **Example:**

ER Diagram:

- Entity: Student(StudentID, Name)
- Entity: Course(CourseID, Title)
- Relationship: Enrolls(StudentID, CourseID, Date)

Converted Tables:

sql

CopyEdit

Student(StudentID PRIMARY KEY, Name)

Course(CourseID PRIMARY KEY, Title)

Enrolls(StudentID, CourseID, Date, PRIMARY KEY(StudentID, CourseID)).

Unit-2

🔹 1. Introduction to Decision Support Systems (DSS)

A **Decision Support System (DSS)** is an interactive, computer-based system that supports decision-making activities. It helps decision-makers compile useful information from raw data, documents, personal knowledge, or business models to identify and solve problems and make decisions.

🔹 2. History of DSS

The concept of DSS emerged in the **1960s** and evolved significantly over the decades:

- **1960s:** Initial systems focused on **Management Information Systems (MIS)**, which provided regular reports and summaries but were not interactive.
- **1970s:** The term "DSS" was formally introduced. DSS systems began supporting semi-structured decision-making.
- **1980s:** Personal computing and spreadsheet tools like Lotus 1-2-3 and Excel made DSS more accessible to end-users.
- **1990s–2000s:** Emergence of **data warehouses**, **OLAP**, **data mining**, and **web-based DSS**.
- **Present Day:** Integration with AI, machine learning, big data, and cloud-based platforms has made DSS more intelligent and predictive.

🔹 3. Characteristics of DSS

- **Interactive Interface:** Allows users to communicate directly with the system.
- **Support for Semi-structured/Unstructured Decisions:** Unlike MIS which supports structured decisions, DSS helps in complex and ill-defined problems.
- **Model-Oriented:** Incorporates analytical and simulation models to support decision-making.
- **Adaptability:** Can be tailored to specific organizational needs.
- **Data-Driven:** Utilizes internal and external data for insights.
- **Support for Multiple Users:** Often supports collaborative decision-making.
- **User-Friendly:** Designed for non-technical users.

🔹 4. Benefits of DSS

- **Improved Decision Quality:** By analyzing large volumes of data and presenting relevant insights.
- **Faster Decision Making:** By automating data retrieval and analysis.
- **Enhanced Productivity:** Helps employees and managers make better use of time and data.
- **Supports Innovation:** Encourages experimentation with "what-if" scenarios.
- **Facilitates Communication:** Especially in group DSS (GDSS).
- **Reduces Costs:** By optimizing resource allocation and reducing decision errors.
- **Competitive Advantage:** Through informed, data-driven strategic planning.

🔹 5. Components of DSS

A typical DSS consists of the following components:

1. Database Management Subsystem

- Stores both current and historical data.
- Supports queries and data manipulation.

2. Model Management Subsystem

- Provides access to decision models (statistical, financial, optimization, simulation).
- Allows users to experiment with different scenarios.

3. User Interface (UI)

- Allows interaction between the user and the system.
- Includes dashboards, menus, forms, and visualization tools.

4. Knowledge-Based Subsystem (optional)

- Integrates AI to mimic expert knowledge.
- Can provide intelligent recommendations.

🔹 6. Operational Data vs DSS Data

Feature	Operational Data	DSS Data
Purpose	Day-to-day operations	Decision making and analysis
Type	Current, real-time	Historical, aggregated
Update Frequency	Continuous	Periodic

Feature	Operational Data	DSS Data
Users	Operational staff	Analysts, managers
Detail Level	High granularity	Summarized, analytical
Source	ERP, CRM, POS systems	Data warehouses, external sources

---

🔹 7. Relationship Types in Databases

➤ One-to-One (1:1)

- One record in Table A corresponds to only one record in Table B.
- Example: One employee has one unique company car.

➤ One-to-Many (1:N)

- One record in Table A corresponds to multiple records in Table B.
- Example: One teacher teaches many students.

➤ Many-to-Many (M:N)

- Multiple records in Table A relate to multiple records in Table B.
- Example: Many students enroll in many courses.

M:N relationships require an **intermediate (junction) table** in a relational database to resolve.

🔹 8. Data Warehousing

👉 Introduction to Data Warehousing

A **Data Warehouse** is a centralized repository that stores large volumes of structured data from multiple sources. It is designed to facilitate query and analysis rather than transaction processing.

Coined by **Bill Inmon**, the “father of data warehousing,” a data warehouse is subject-oriented, integrated, time-variant, and non-volatile.

👉 Main Components of Data Warehouses

1. **Data Source Layer**: Operational databases, flat files, web logs, etc.
2. **Data Staging Layer**: ETL (Extract, Transform, Load) processes.
3. **Data Storage Layer**: Centralized warehouse storing historical data.
4. **Metadata Layer**: Data about the data — definitions, mappings, data lineage.
5. **Presentation Layer**: OLAP cubes, reporting tools, dashboards.
6. **Data Marts**: Subsets of the data warehouse, focused on specific business areas.

✅ Benefits of Data Warehousing

- Consolidates data from multiple sources.
- Provides historical insights.
- Improves query performance.
- Supports better business intelligence (BI).
- Enhances data quality and consistency.

❌ Limitations of Data Warehousing

- High initial costs for setup and maintenance.

- Complex implementation and integration.
- Data latency — may not reflect real-time updates.
- Requires skilled professionals.

🔹 9. Data Mining

👉 Introduction to Data Mining

Data Mining is the process of discovering patterns, correlations, trends, or useful information from large datasets using statistical, machine learning, and AI techniques.

It transforms raw data into useful knowledge.

👉 Data Mining Tools

1. **WEKA** – Open-source tool for data analysis and predictive modeling.
2. **RapidMiner** – Offers data prep, ML, deep learning, text mining.
3. **KNIME** – Drag-and-drop visual workflow editor.
4. **Orange** – Interactive, component-based machine learning tool.
5. **SAS Enterprise Miner** – Advanced analytics and data mining.
6. **R & Python Libraries** – scikit-learn, pandas, caret, etc.

👉 Applications of Data Mining

Domain	Application
Retail	Market basket analysis, customer segmentation
Finance	Fraud detection, credit scoring
Healthcare	Disease prediction, treatment effectiveness
Marketing	Customer churn prediction, targeted campaigns
E-commerce	Recommendation systems, sentiment analysis
Banking	Risk analysis, investment strategy

Unit-3

🔷 I. Advanced Database Models

As databases evolved from simple file-based systems to complex relational models, the need for more **semantically rich, flexible, and specialized models** arose. These advanced models were developed to handle complex data types, relationships, and application-specific requirements.

✅ 1. Object-Oriented Database Model (OODB)

- Combines object-oriented programming concepts with database technology.
- Stores data in the form of **objects** (like classes in OOP), with **attributes** and **methods**.
- Supports features like **inheritance**, **encapsulation**, and **polymorphism**.

Advantages:

- Best suited for applications like **CAD**, **multimedia**, and **AI**.
- Handles **complex data** like images, audio, and video efficiently.
- Provides **tight integration** with programming languages like Java and C++.

✅ 2. Object-Relational Database Model (ORDB)

- A hybrid of relational and object-oriented models.
- Extends the relational model by allowing objects, user-defined types, and methods.

- Used in systems like **PostgreSQL** and **Oracle**.

Example:

sql

CopyEdit

CREATE TYPE complex AS (real FLOAT, imag FLOAT);

Advantages:

- Maintains **data integrity** and supports **complex data types**.
- Allows better **scalability** and **reusability** of code and data.

✅ 3. Hierarchical Model

- Organizes data in a **tree-like structure** using parent-child relationships.
- Each child has only **one parent**, but a parent can have multiple children.

Example Use: IBM's IMS (Information Management System).

Limitations:

- Rigid structure.
- Difficult to reorganize.

✅ 4. Network Database Model

- Similar to hierarchical, but allows **many-to-many relationships** via **pointers**.
- Uses **records** and **sets** to define relationships.

Advantage: More flexible than hierarchical models.

Example Use: Integrated Data Store (IDS).

✅ 5. NoSQL Databases (Non-relational)

- Designed for **unstructured**, **semi-structured**, and **large-scale** data.
- Categories:
  - **Document Stores** (MongoDB)
  - **Key-Value Stores** (Redis)
  - **Columnar Stores** (Cassandra)
  - **Graph Databases** (Neo4j)

Advantages:

- High scalability.
- Schema flexibility.
- Excellent performance for **Big Data** applications.

🔷 II. Database Security

Database security involves protecting the database against **unauthorized access**, **data breaches**, **corruption**, and **other threats**.

✅ 1. Types of Database Failures

🔸 a) System Crash

- Sudden hardware/software failure causes system to stop.
- Recovery needed to prevent data loss.

🔸 b) Transaction Failure

- A single transaction fails due to logical error or violation of constraints.

🔸 **c) Media Failure**

- Physical failure of storage device, such as a hard disk crash.

🔸 **d) Application Software Failure**

- Bugs in the application interacting with the database.

🔸 **e) Natural Disasters**

- Earthquakes, floods, or fires can destroy physical infrastructure.

✅ 2. Types of Database Recovery

Database recovery ensures that the database is restored to a consistent state after a failure.

🔸 **a) Manual Recovery**

- Admin manually restores backup files.

🔸 **b) Automated Recovery**

- DBMS automatically restores to last consistent state using logs.

🔸 **c) Immediate Update and Deferred Update Recovery**

- **Deferred**: Changes are made to log first, then to the database after commit.
- **Immediate**: Changes made to database immediately but recorded in logs for rollback.

🔸 **d) Checkpoints**

- Periodic snapshots of the database state used to speed up recovery.

🔸 **e) Shadow Paging**

- Maintains two page tables: current and shadow. After crash, use the shadow table to restore.

🔸 **f) Log-Based Recovery**

- Keeps a **transaction log** of changes. Uses undo/redo operations.

✅ **3. Types of Database Security Issues**

🔹 **a) Unauthorized Access**

- Accessing confidential data without permission.

🔹 **b) SQL Injection**

- Malicious code injected through SQL queries.

🔹 **c) Privilege Abuse**

- Users misusing their access rights.

🔹 **d) Data Leaks**

- Sensitive data exposed through software bugs or system misconfigurations.

🔹 **e) Insider Threats**

- Employees or partners leaking or destroying data.

🔹 **f) Backup Theft**

- Stolen backup media resulting in complete data exposure.

✅ **4. Authorization and Authentication**

🔹 **Authentication**

- **Verifies identity** of the user.
- Methods include:

- Username/password
- OTP
- Biometric
- Smart cards

🔹 **Authorization**

- Determines what actions a verified user is **allowed** to perform.
- Handled via **roles**, **permissions**, and **access control lists (ACLs)**.

Example:

- **User A** can read table but not update.
- **User B** can read and write.

✅ **5. Audit Trails**

- A **log** that records all database access and operations.
- Helps track:
  - Who accessed data?
  - What was modified?
  - When and how?

Uses:

- Detecting unauthorized access.
- Ensuring compliance with legal standards (e.g., GDPR, HIPAA).
- Forensic investigations after a breach.

✅ **6. Firewalls**

- A **barrier** between the internal database system and external networks.
- Controls incoming and outgoing traffic.
- Types include:
  - **Network Firewalls** (packet filtering)
  - **Web Application Firewalls (WAF)** – for HTTP traffic

Role in DB Security:

- Prevents attackers from gaining network-level access to the database.

✅ 7. Data Encryption and Decryption (Cryptography)

Cryptography protects data by converting it into a form that is **unreadable** without a special key.

🔹 a) Encryption

- Converts **plaintext** into **ciphertext** using an encryption algorithm.
- Types:
  - **Symmetric encryption** (same key for encryption and decryption)
    - Example: AES, DES
  - **Asymmetric encryption** (public key for encryption, private key for decryption)
    - Example: RSA

🔹 b) Decryption

- Converts **ciphertext back to plaintext** using a key.

🔹 c) Hashing

- One-way encryption; used for storing passwords.
- Example: SHA-256, bcrypt

Use Cases in DB:

- **Encrypted connections** (SSL/TLS)
- **Encrypted columns** (e.g., credit card data)
- **Encrypted backups**

✅ 8. Best Practices in Database Security

- Use **strong authentication** mechanisms.
- **Encrypt** sensitive data at rest and in transit.
- Regularly **patch** DBMS and applications.
- **Limit privileges** using the principle of least privilege (PoLP).
- Use **audit logs** and monitor anomalies.
- Implement **intrusion detection** systems.
- Secure **physical access** to servers and backups.

Unit-4

🔷 Part 1: Database Operations in Microsoft Access

📌 Introduction to Microsoft Access

Microsoft Access is a powerful **Relational Database Management System (RDBMS)** developed by Microsoft. It combines the **relational Microsoft Jet Database Engine** with a **graphical user interface** and **software-development tools**. It's widely used for creating desktop-based database applications, especially in small- to medium-sized businesses and for educational purposes.

Access allows users to **store data in tables**, **query** data, build **forms for data input**, and create **reports** for printing and data analysis.

✅ 1. Creating a Table

A **table** is the fundamental building block in MS Access where data is stored in rows and columns.

✳️ Steps to Create a Table:

1. Open Microsoft Access and create a **new database**.
2. Go to the **"Create"** tab on the ribbon.
3. Click **“Table Design”** to define fields and data types manually.
4. For each field, enter:
    - **Field Name** (e.g., ID, Name, Age)
    - **Data Type** (e.g., Text, Number, Date/Time, AutoNumber)
    - **Description** (optional notes for each field)
5. Choose a **Primary Key** to uniquely identify each record.

6. Save the table with a meaningful name (e.g., Students, Employees).

🛠️ Data Types in Access:

- **Short Text**: up to 255 characters
- **Long Text**: larger text fields (e.g., notes)
- **Number**: numeric data
- **Currency**: for financial data
- **Date/Time**: for time-related data
- **AutoNumber**: auto-incrementing numbers (often used as Primary Key)
- **Yes/No**: Boolean values
- **Attachment**: files or images

✅ 2. Creating Forms

A **form** is a user-friendly interface for entering and editing data.

✳️ Steps to Create a Form:

1. Go to the **“Create”** tab.
2. Select **“Form”** if you want to generate a form automatically based on a selected table.
3. Use **“Form Design”** to create a form from scratch.
4. Add form controls (labels, text boxes, combo boxes) from the **“Design”** tab.
5. Link the form to a specific table or query.
6. Use **"Property Sheet"** to adjust layout, formatting, and control properties.

7. Save the form.

🔹 **Benefits of Forms:**

- Simplified data entry
- Better validation
- Custom layout and formatting
- Controlled user access

✅ 3. Creating a Simple Query and Modifying a Query

A **query** is used to search, filter, and retrieve specific data from one or more tables.

✳️ Creating a Simple Query:

1. Go to the **"Create"** tab.
2. Click on **"Query Design"**.
3. Add one or more tables you want to query.
4. Double-click on the fields you want to include.
5. In the **Criteria row**, enter conditions (e.g., > 18 for Age).
6. Run the query using the **"Run"** (red exclamation mark) button.

✳️ Modifying a Query:

- Use **Design View** to:
  - Add/remove fields.
  - Change sort order.
  - Modify filter criteria.
- Use **SQL View** to directly write or edit SQL queries.

Example SQL:

sql

CopyEdit

SELECT FirstName, LastName FROM Students WHERE Age > 18;

✅ 4. Subqueries

A **subquery** is a query nested inside another query.

- 🔹 **Uses of Subqueries:**
  - Filter results based on related data.
  - Compare values across tables.

Example:

sql

CopyEdit

SELECT Name FROM Employees

WHERE DepartmentID IN

(SELECT DepartmentID FROM Departments WHERE DepartmentName = 'HR');

Subqueries allow for **complex filtering** and **data comparisons** within Access queries.

✅ 5. Data Retrieval and Deletion

- 🔹 **Retrieval:**
  - Use SELECT queries to **view data** without changing it.
  - Example:

sql

CopyEdit

SELECT * FROM Orders WHERE OrderDate > #01/01/2024#;

🔹 **Deletion:**

- Use **DELETE queries** to remove records that meet specific criteria.
- Be cautious—deleted data is **not recoverable** without backup.

sql

CopyEdit

DELETE FROM Students WHERE Age < 10;

Use **confirmation prompts** and **backups** before executing delete queries.

🔷 Part 2: Microsoft PowerPoint

📌 Introduction

Microsoft PowerPoint is a **presentation software** developed by Microsoft. It is used for creating **slideshows**, **educational materials**, **business presentations**, and **visual storytelling**. PowerPoint presentations consist of a sequence of **slides** that can include text, images, charts, audio, video, and animations.

✅ 1. Tools and Menus in PowerPoint

PowerPoint's UI is organized into **tabs**, each containing a **ribbon** with grouped tools.

✳️ Main Menus:

- **Home**: Basic tools for text formatting, paragraph, slides.

- **Insert**: Add new elements like tables, images, charts, audio, video, shapes.
- **Design**: Apply themes, slide size, and formatting.
- **Transitions**: Visual effects between slides.
- **Animations**: Motion effects within slides.
- **Slide Show**: Tools for presentation setup and running.
- **Review**: Spell check, comments.
- **View**: Switch between slide view, grid, notes.

✅ 2. Creating Slides

✳️ Steps:

1. Open PowerPoint.
2. Choose a **template** or blank presentation.
3. Click **New Slide** (Home > New Slide).
4. Select layout (Title Slide, Title and Content, Two Content, etc.).
5. Add content using text boxes, images, or multimedia.

🛠️ Common Elements:

- **Title** and **subtitle**
- **Bulleted or numbered lists**
- **Charts and graphs**
- **SmartArt graphics**
- **Tables**
- **Embedded videos and sound clips**

✅ 3. Using Animations

Animations bring dynamic effects to slide content.

🔹 Types of Animations:

- **Entrance** (Fade In, Fly In)
- **Emphasis** (Pulse, Spin)
- **Exit** (Fade Out, Fly Out)
- **Motion Paths** (move objects along a path)

✳️ How to Add Animation:

1. Select the object (text/image).
2. Go to the **Animations** tab.
3. Click **Add Animation**.
4. Set **timing** and **effect options**.
5. Use the **Animation Pane** to view and reorder animations.

✅ 4. Slide Transitions

Slide transitions are visual effects between slides during a presentation.

✳️ To Add Transitions:

1. Go to the **Transitions** tab.
2. Choose a transition style (Fade, Wipe, Push, etc.).
3. Set **duration** and add **sound** if needed.
4. Apply to all slides (optional).

✅ 5. Presentation Modes

PowerPoint provides various **views** and **modes**:

- **Normal View**: Main editing window.
- **Slide Sorter View**: Manage slide sequence.
- **Reading View**: Play slideshow within the window.
- **Slide Show View**: Full-screen presentation.

✅ 6. Tips for Effective Presentations

- Keep slides **simple and readable**.
- Use **images and graphics** for visual appeal.
- Limit the amount of text per slide.
- Use **consistent fonts and colors**.
- Rehearse using **Presenter View** for notes and timing.