Chapter 1 Notes: Introduction to Databases

1 Data & Information

- **Data**: Raw, unorganized facts (e.g., sensor readings, survey responses).
- **Information**: Processed, organized data that supports decision-making (e.g., survey reports).

2 Database

- A digital storage system for Organized data.
- **Types**:
  - **Relational**: Data in tables with rows/columns (e.g., MySQL).
  - **Non-relational**: Flexible structures like key-value pairs or documents.

3 Database Management System (DBMS)

- Software for creating and managing databases (e.g., MySQL, PostgreSQL).
- Enables creating, updating, deleting, and securing data efficiently.
- Manages large datasets, defines storage structures, and controls access.

4 Database System

- Combines a database and its DBMS.
- Stores data and provides tools for efficient management and access.

5 Importance of Data

- Supports decision-making, scientific research, and AI.
- Enables planning, forecasting, and spotting trends.

6 File Processing System Issues

- **Data Redundancy**: Duplicate data increases storage costs.
- **Inconsistency**: Same data in multiple places may not match.
- **Access Difficulty**: Hard to retrieve specific data flexibly.
- **Data Isolation**: Scattered data is tough to combine or analyze.
- **Integrity Problems**: Inconsistent data across files (e.g., mismatched IDs).
- **Atomicity Issues**: No rollback if a transaction fails.

- **Security**: Limited control over data access.

7 Data Management Benefits

- **Reliability**: Organized data is trustworthy for decisions.
- **Security**: Backups and access controls protect data.
- **Accessibility**: Easy and quick data retrieval.
- **Scalability**: Simple to add or remove data as needed.

8 Data Models

- **Relational**: Data in linked tables; uses SQL (most common).
- **Entity-Relationship (E-R)**: High-level model with entities and relationships, shown via E-R diagrams.
- **Semi-structured**: Flexible data with varying attributes (e.g., JSON, XML).
- **Object-Based**: Stores data as objects, like in OOP (e.g., classes, inheritance).
- **Hierarchical**: Tree-like structure with one-to-many relationships.
- **Network**: Flexible with many-to-many relationships.

9 DBMS Architecture (Three-Schema)

- **Physical Level**: How data is stored (e.g., B-trees); users dont interact here.
- **Logical Level**: Defines data and relationships (e.g., tables, constraints).
- **View Level**: Shows specific data to users, ensuring security.

10 Data Independence

- **Logical**: Change table structures without affecting user views.
- **Physical**: Change storage methods without altering logical schema.

11 Distributed Databases

- Multiple databases across locations, logically connected.
- **Types**:
  - **Homogeneous**: Same software and setup across nodes.
  - **Heterogeneous**: Different software or systems.
- **Compared to Centralized**:
  - Distributed: Spread across networks, harder to sync, but fault-tolerant.

  - Centralized: Single location, easier to manage, but fails if the system goes down.

12 History of Databases

- 1960s: Network and hierarchical models emerge.
- 1970s: Relational databases (RDBMS) introduced.
- 1980s-1990s: Oracle, Excel, and non-relational databases gain popularity.
- 2000s-2020s: Big data, NoSQL, and cloud databases grow.

Hope this helps you study smarter!