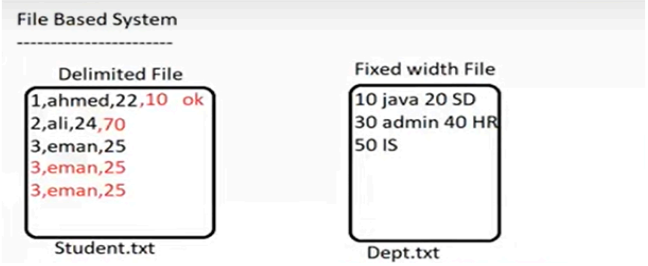
File-Based System VS Database System

File-Based System:

1. **Definition:**
   A file-based system stores data in separate files (text, CSV, Excel, binary) managed by the operating system. Applications directly read/write data from files without a central management system.
2. **Characteristics:**
   - Data stored in flat files or application-specific files.
   - Each application manages its own data.
   - No standard query language (you write custom programs to access data).
   - Files may be text or binary.
3. **Advantages:**
   - Simple to implement – small systems can quickly store data in files.
   - Low cost – no need for special DBMS software.
   - Good for small-scale applications – simple, single-user apps like notepad lists.
4. **Disadvantages:**
   - Data Redundancy(Data duplication) & Inconsistency – same data stored in multiple files (customer name in order file and payment file).
   - Data Isolation – difficult to retrieve data across multiple files.
   - No Data Integrity & Security – enforcing constraints is hard.
   - Poor Scalability – difficult to manage as data grows.
   - Concurrency Issues – hard to handle multi-user access safely.
   - No Standard Query Support – custom programs must be written to process files.
   - low performance.
   - No relationship
   - Manual backup & restore
   - No standard
   - Long development time
   - No constraints & rules
   - Difficult search
   - Separated copies



Database System:

1. **Definition:**
   database system uses a Database Management System (DBMS) to store, manage, and access data. Data is organized into structured formats (tables in relational databases), and applications interact with the DBMS rather than files directly.
2. **Characteristics:**
   - Centralized management of data.
   - Provides data abstraction (logical view vs physical storage).
   - Supports SQL (Structured Query Language).
   - Enforces constraints, relationships, and security.
3. **Advantages:**
   - Reduced Redundancy & Inconsistency – data is stored once and shared.
   - Data Integrity & Accuracy – constraints (primary key, foreign key) ensure correctness.
   - Data Security – role-based access control.
   - Concurrency Control – multiple users can safely access data.
   - Backup & Recovery – DBMS supports data recovery after failures.
   - Scalability & Efficiency – optimized for handling large data and complex queries.
   - Standard Query Language (SQL) – easy data retrieval and updates.
4. **Disadvantages:**
   - Cost – commercial DBMS licenses can be expensive.
   - Complexity – requires skilled personnel for design and management.
   - Overhead – for very small applications, DBMS may be “too heavy.”