Database Management System (DBMS)

→ Basic Introduction
  - ↳ definitions
  - → 2 tier, 3 tier
  - → 3 schema architecture / 3 levels of abstraction.

  Data independence.

  → various data models
    - → Network data model
    - → heirarichal data model
    - → Relational data model
    - → ER model
    - → object Oriented data model.

→ E-R Model.
  - ↳ conceptual schema.

  ↓

  Entity-Relationship model.
  - → Types of attributes
  - → Relationships
  - → Types of Relationships (one to one, one to many etc.)

→ Basics of keys
  - ↳ Primary key
  - → candidate key
  - → super key
  - → Foreign key

→ Normalization

- ↳ Closure method
  - ↳ used to find candidate keys
- → Functional dependencies
- → 1NF, 2NF, 3NF, BCNF

→ Transaction, Control and Concurrency

- ↳ ACID Properties
- → Read - Write problem
- → Write - Read problem
- → Write - Write problem
- → Conflict serializability.
- → Recoverability
- → 2 phase Locking } concurrency.
- → timestamp locking protocol. } concurrency.

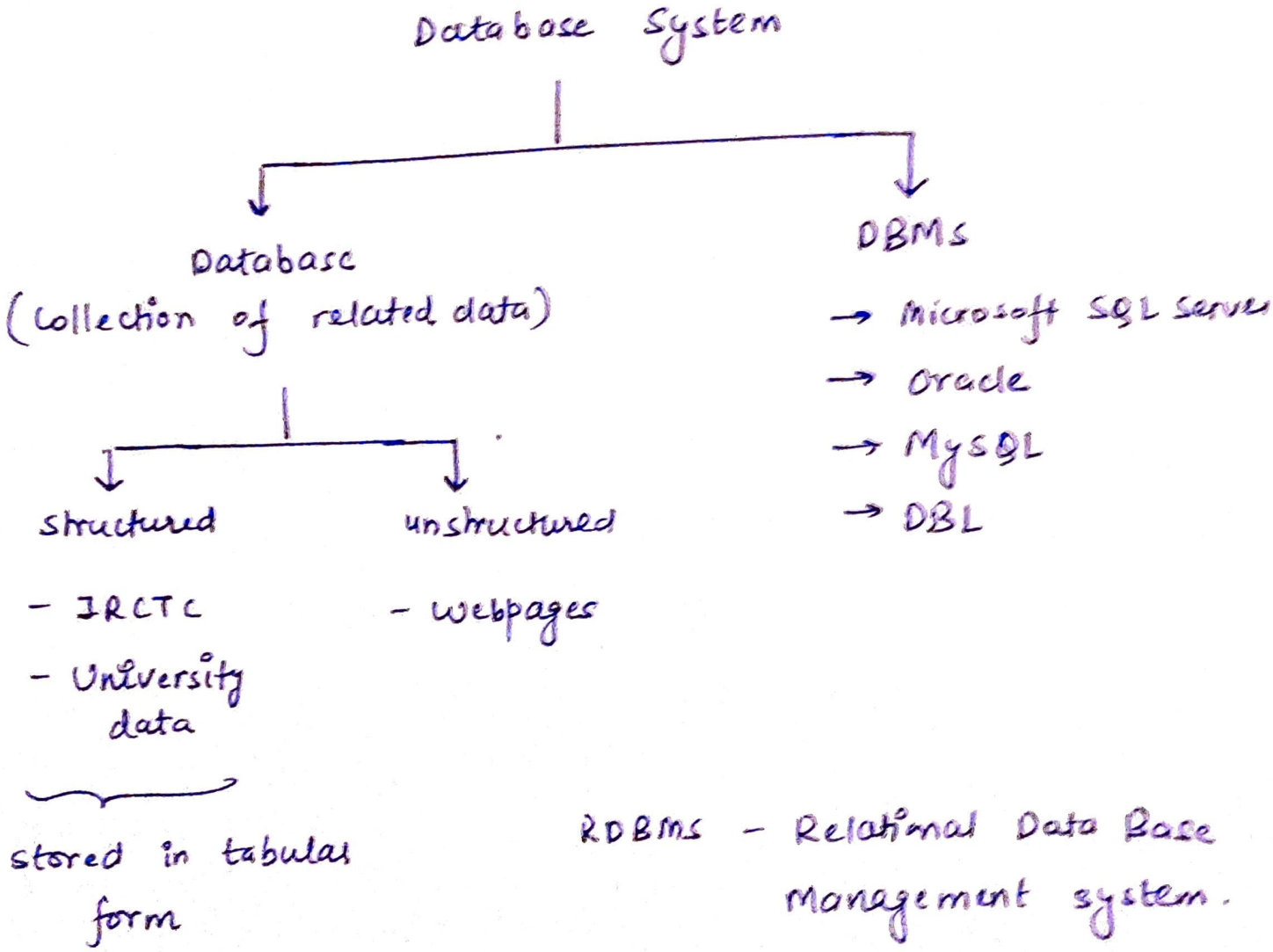
→ SQL and Relationship Algebra

- → DDL, DML, DCL commands
- → constraints
- → Aggregate Functions
- → Joins
- → Nested Query

→ Indexing

- → Primary indexing
- → Cluster indexing
- → Secondary indexing
- → B tree, B+ tree

Introduction to DBMS



RDBMs - Relational Data Base Management system.

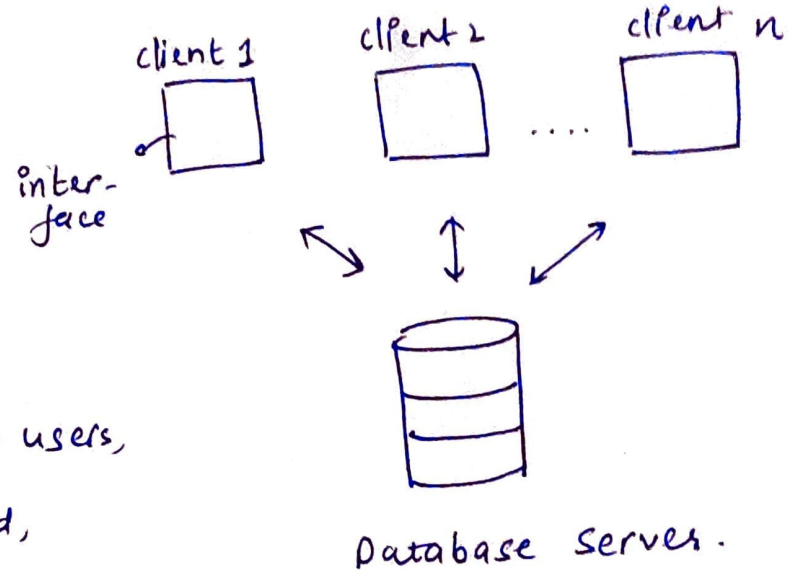
File System v/s DBMS

- File system is good for individual users but DBMS is required for client-server architecture.
- User does not need all data and using DBMS, he can search.
- Attributes are not required to search data in DBMS
- Concurrency - Multiple people accessing data at same time. DBMS has protocols for this
  - concurrency is a problem for file system.
- Security - (Role based security)
  - ↳ heirarchal security
    ex. students, faculty, dean.
  - Role based access control.
- Redundancy - Redundancy is controlled in DBMS.

2 tier and 3 tier architecture

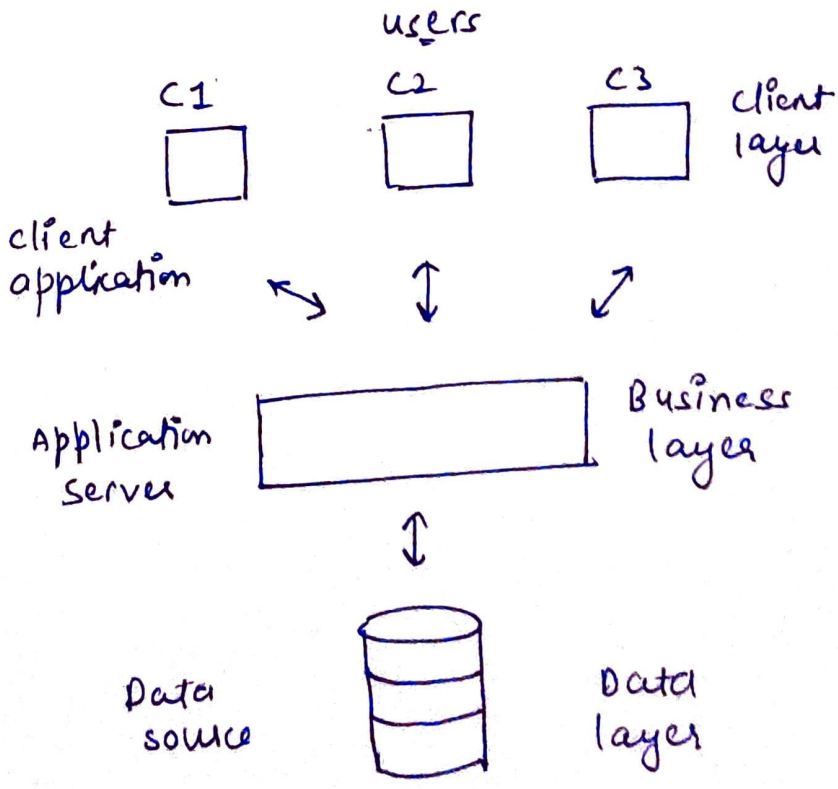
2 tier → 2 layers

client-server architecture



- This architecture fails when there are many users, much data is called, at any time etc.
  thus, scalability is difficult.
- security is also a problem. (because client directly interacts with database)
- However, maintenance is easy

3 Tier



- Business layer processes the queries
- Load on database server is low
- This is also useful because data is mostly distributed across multiple servers, scalability is possible