

Fundamentals of Database Systems

- Fundamentals of Database Systems
- Important
- Introduction
 - Data vs Information
 - Database systems
- Data Classification
- Data Modeling and Abstraction
 - Data Modeling
 - Schema
 - Data Abstraction
 - Data independence
 - Database Management System
- Entity-Relationship (ER) Model
 - Attributes
 - Relationships
 - What is a relationship?
 - Most common cardinality ratios for binary relationships
 - Participation Constraint
 - Recursive Relationship
 - Identifying Relationships
 - ER Diagram notation

Important

Many of the diagrams and information here have been extracted from

- “Database Systems”, by R. ElMasri and S. Navathe, Pearson Higher Education, 6th edition
- Dr. Hossam Sharara’s Slides for the CSCE 2501 : The American University in Cairo

Introduction

Data vs Information

Data usually refers to raw data, while information refers to data after it has been processed, organized and presented. For example, temperature at a given time is “data,” while a graph

representing the change in temperature is information.

Database systems

A database can be regarded as a large collection of related data. It typically models information about a real-world system.

The need for database systems resulted from the following

- Datafication, which can be defined as the technological trend to translate all aspects of life into data.
- Normal data storage methods are insecure, and have poor performance.

Mainly, a database consists of

- Entities (objects)
- Relationships (Relations between entities)
- Constraints (Rules on relationships)

For example, in a university, a student (entity) takes (relationship) a course (entity) that has a maximum capacity (constraint).

Data Classification

Data can be classified into three categories according to its *structure*

- Structured data
 - When data follows a strict, consistent structure; can generally be represented as a table.
- Semi-structured data
 - When data follows some structure, but it is not strict; Different data points might have some data fields and not all of them.
- Unstructured data
 - When data does not follow any form of structure.

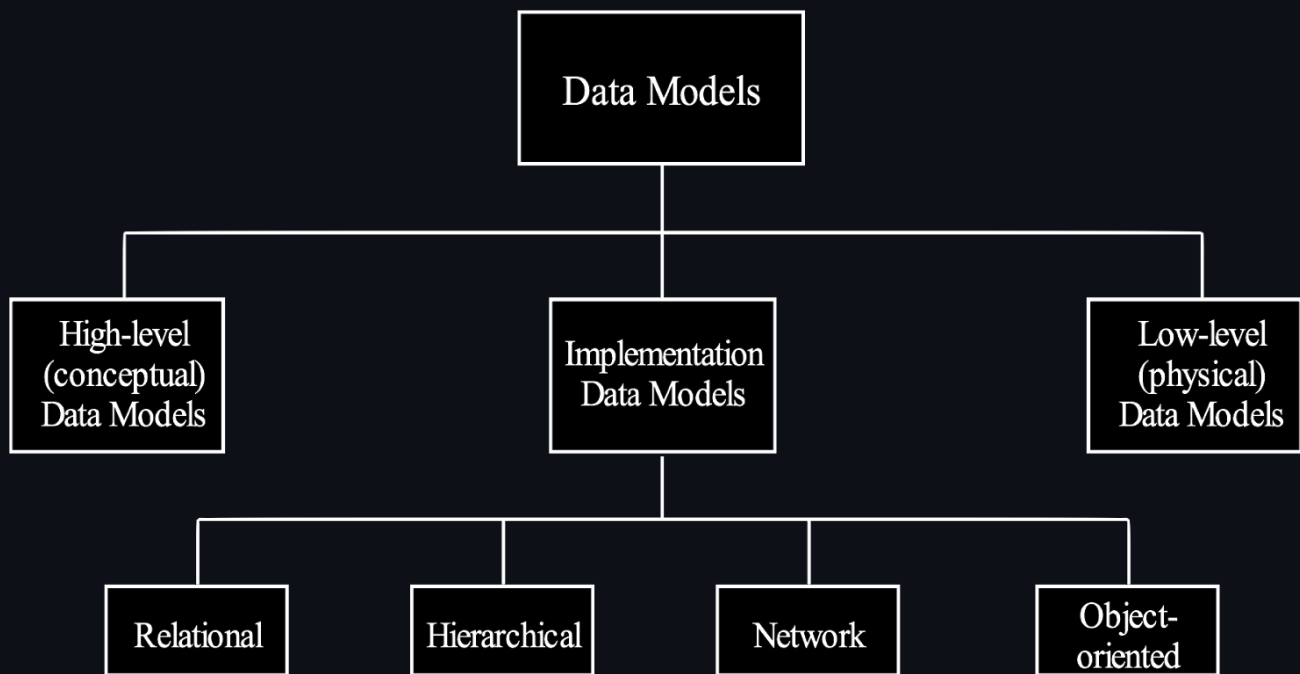
Data Modeling and Abstraction

Structured data is by far the most successful when it comes to storing and operating on as their structure can be exploited for efficiency and ease of use.

Data Modeling

- A collection of concepts that describe how the data is represented/accessed
 - Entity-Relationship model
 - Object-oriented model

- Resource description framework
- Different models vary between descriptive power and ease of use (More descriptive usually means harder to use)
- Allows for reasoning about the data at a high level (isn't this abstraction?)



Schema

- A description of a specific collection of data using a data model.
- A database is an instance of its schema.
- A schema is not expected to change after the design phase is over.

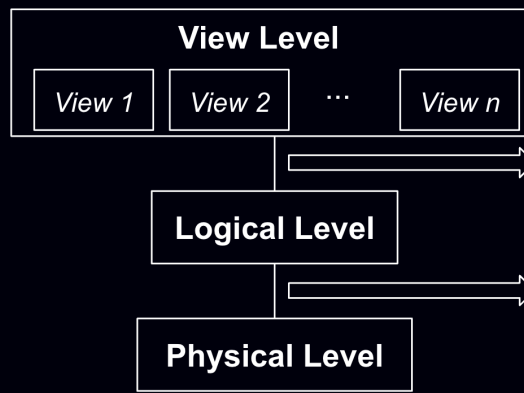
Data Abstraction

- Layering the system into different levels of details
- Helps isolate the users from lower level details

Data independence

- Applications and users are insulated from the structure of the data
- Allows for changing data structures and schema at one level without affecting any higher levels (What does that mean?)

What data is viewable by users / programs?



Logical Data Independence
Protect applications / users from changes in the schema

Physical Data Independence
Protect logical schema from the changes in the physical structure of the data

What data is stored?

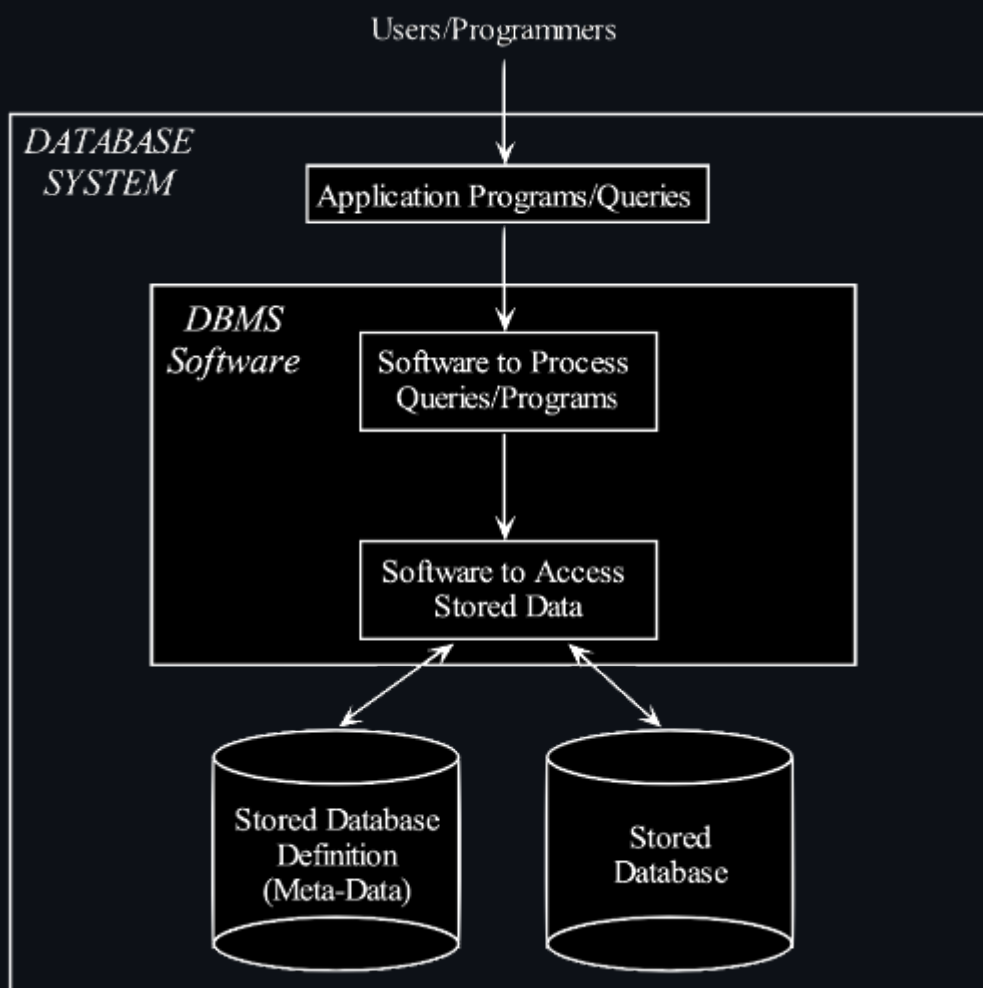
- Semantics, relationships, etc.

How data is stored?

- Disks, Filesystems, etc.

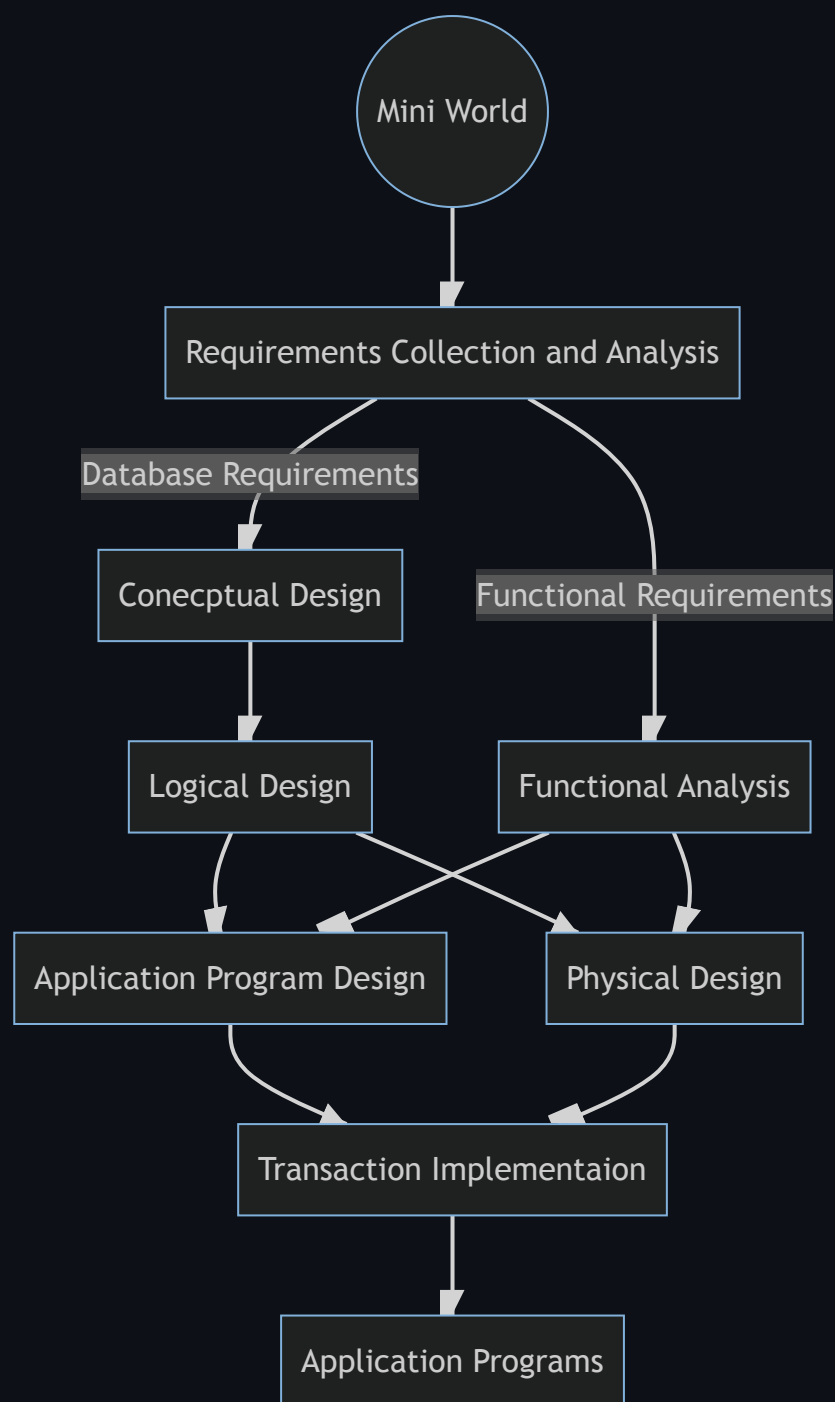
Database Management System

A database management system is a software designed to store, manage, and facilitate access to databases.



It facilitates the process of defining, constructing, and manipulating databases for various applications and guarantees correctness in the presence of failures, concurrency in data access, and data semantics.

- Data Definition Language (DDL)
 - Defining and modifying database schema
- Data Manipulation Language (DML)
 - Retrieving, modifying, and analyzing the data itself.



Entity-Relationship (ER) Model

The ER data model is a set of techniques used for higher level (conceptual) design and representation of

- Different entities
- Relationships between the entities

Attributes

A set of properties that describe an entity

- Simple vs. Composite

- Simple (atomic) : cannot be broken down further
- Composite: Can be broken down into smaller sub-parts
- Single-valued vs. Multi-valued
 - Single: Each entity can have at most one value
 - Multi: Each entity can have more than one value
- Stored vs. Derived
 - Stored: Must be stored in the data (cannot be inferred from other information)
 - Derived: Can be computed from other attributes (e.g. age from DOB)
- Null-valued
 - The value of an attribute is not stored nor can be derived
 - A null value will not be used in any computation
- Key Attribute
 - Used to uniquely identify entities of some type; accordingly, its values are distinct.

If a given entity type does not have a key attribute, it is called a weak type.

Avoid creating new IDs for types that have natural IDs: For example, use National ID instead of Student ID, or Bank Account Number.

Relationships

• What is a relationship?

- Defines an association between two or more entities
- Refers to an instance between objects
- Might have one or more attributes
- Might include more than two entities
- **Degree** of a relationship is the number of participating entity types

• Most common *cardinality* ratios for binary relationships

- 1:1 (one to one) One person can only have one spouse
- 1:N (one to many) One person can have more than one child
- M:N (many to many) many people can be relatives of many people

• Participation Constraint

- Partial (Does not have to participate)
- Total (All entities have to participate)

Participation constraints can be associated with a structural constraint, specifying min/max number of entities that can exist in the relationship.

• Recursive Relationship

Between two different entities of the same type.








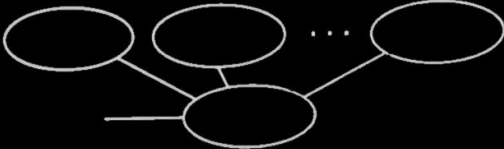



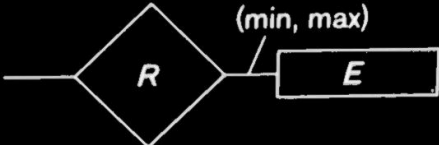
Roles should be added to each participating entity so signify the roles each entity plays. For example, a supervisor employee has a supervision relationship with a supervised employee.

- **Identifying Relationships**

Weak entity types are identifies by being related to a strong entity in combination with some of their attributes.

ER Diagram notation

Total relations are represented by two adjacent lines

Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Identifying Relationship
	Attribute
	Key Attribute
	Multivalued Attribute
	Composite Attribute
	Derived Attribute
	Total Participation of E_2 in R
	Cardinality Ratio 1: N for $E_1:E_2$ in R
	Structural Constraint (min, max) on Participation of E in R