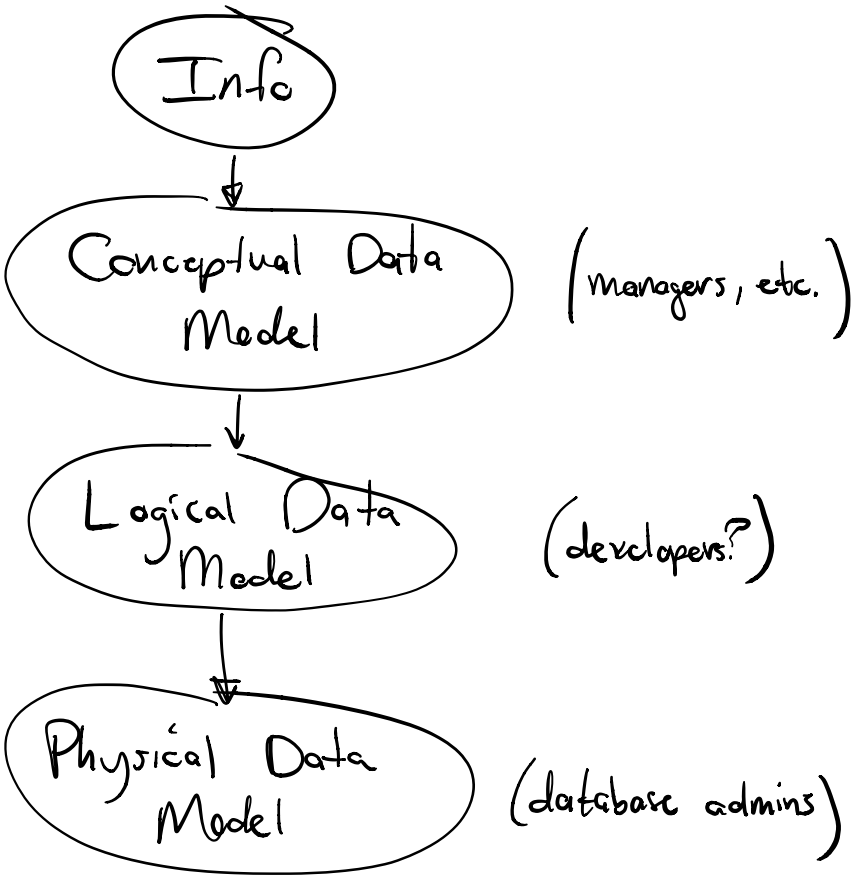
Database Design

- Activity of developing a schema for a database for a given dataset



Conceptual Data Model

- far removed from the actual DB implementation
- just to help think about the data
- identify objects and relationships between them
- the goal of this step is to create a complete description of the database in a informal paper-and-pencil format

Logical Data Model

- start to transform the conceptual design into a more formal schema (logical schema)

Physical Data Model

- start modeling the reality of the DB implementation
- choose a DBMS
- choose data types
- server design / network design
- "choosing access methods"

WHY EACH LEVEL?

Conceptual

- Helping you think / talk about data
- Semantic clarity

Logical

- help in determine what DBMS would be best
- at this stage you can start to structure the schema more abstractly to avoid redundancy and to improve efficiency (normalization)

Physical
- we want a real DB
- performance, optimization (de-normalization)

CONCEPTUAL DESIGN

- The consensus choice for conceptual modelling is the Entity-Relationship (ER) model

Two primitives
Entities: things that exist and are distinguishable
eg. Bob, Alice, Susan's car, DU....
Relationships: two or more entities can participate in a relationship
eg: Bob "borrowed" Susan's car,
Susan "is a patron of" DU

Both Entities and Relationships can have attributes
eg. Bob has an age, Susan's car has color
Susan is a #1000 patron of DU

Take a group of all entities with the same attributes, that forms an "Entity Set"

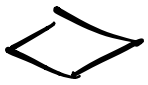
eg: all Persons, all Cars, all Universities

Relationships are also grouped into homogeneous sets called "Relationship Sets". All the pairs of Entities such that the relationship is true for them.

eg: All the (Person, Car) pairs such that the person owns the car.

The overall structure of a DB can be expressed graphically as an E-R diagram, typically with the following components:

Entity Sets → 

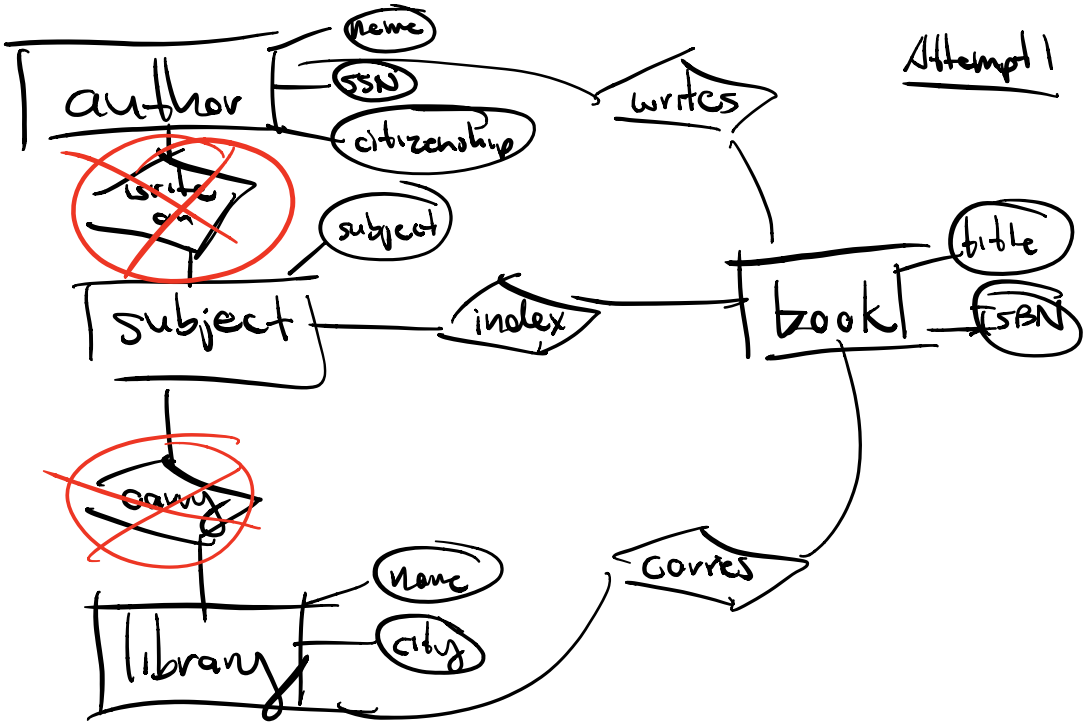
Relationship Sets → 

Attributes → 

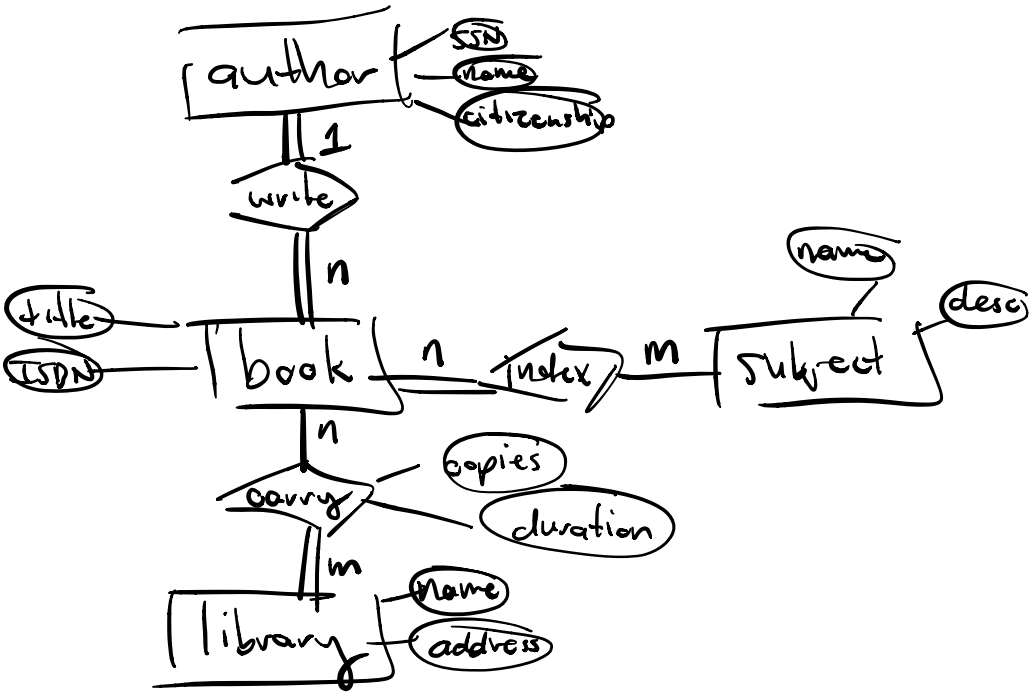
Links between → 

Ex: We want a library DB where authors have written books about various subjects (one author per book). It also has info about libraries that carry books on the various subjects.

Attempt 1



Q: I have an author "Isaac Asimov", which books has he written?



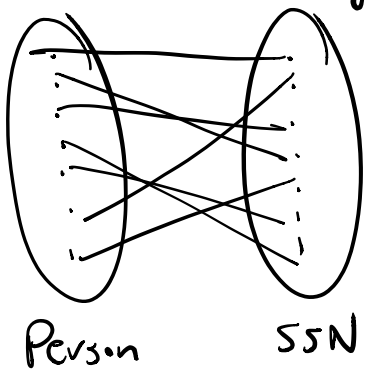
Relationship is an ordered set of Entities
Relationship Set is a set of ordered sets of Entities

'borrowed' → { (Bob, Susan's car),
(Alice, Bob's book),
(Joe, John's nailclippers) }

A relationship might involve:
2 entities: binary relationship
3 entities: ternary relationship
n entities: N-ary relationship

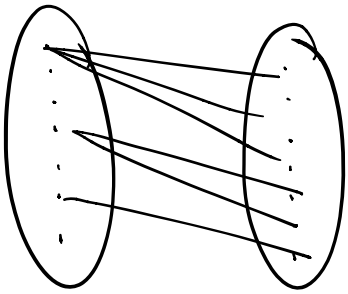
A binary relationship between two sets of entities may be: ("arity")

1 : 1 relationship
eg. a Person having a SSN



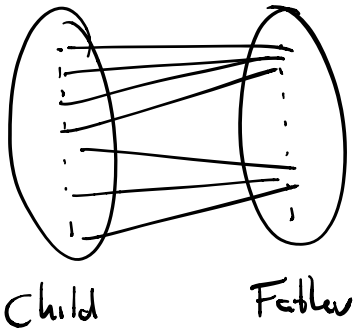
1 : N Relationship

e.g. Father having children



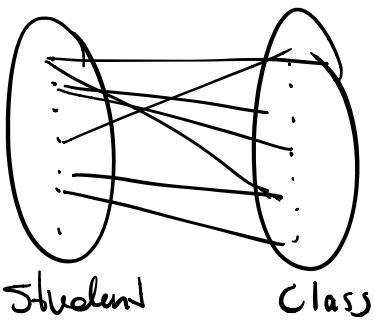
N : 1 Relationship

e.g. Children having fathers



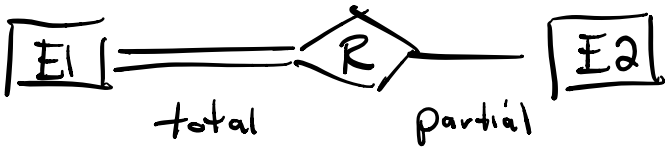
M : N Relationship

e.g. Students enrolled in classes



Participation Constraints

- addresses whether or not Entities can or must participate in some relationship



total participation constraint
- every entity in E1 must participate in the relationship R

partial participation constraint
- each entity in E2 may or may not participate in the relationship R

Keys

Keys are a way of distinguishing Entities or Relationships, and there are various kinds of keys with varying specificity

'Superkey' - a set of one or more attributes of an entity or relationship whose values uniquely identify that entity or relationship

eg: the name "Isaac Asimov" and a particular SSN and a citizenship is a superkey for a specific entity

'Candidate key' - a minimal superkey, ie. one where no subset of its attributes is a superkey

eg. just the SSN

'Primary Key' - one specific candidate key chosen to serve as the identifier for a relationship or entity set

'Foreign Key' - a set of one or more attributes of an entity or relationship that serves as a primary key for another entity or relationship

eg. for students enrolling in a class, we might use student ID as a foreign key in the enroll relationship