ER Diagrams Pt 1

Data models
- means of describing the structure of data
- a set of operations that manipulate the data
- types: conceptual data model, logical data model - relational, network, hierarchical, inverted list, object-oriented

Conceptual Data models
- shows the structure of the data and how things are related
- communication tool
- independent of commercial DBMSes
- easy to learn and use
- helps show semantics&/meaning of data
- graphical representation
- Entity-Relationship Model common

Logical Data Models - Relational, Network, Hierarchical, Inverted List,
- Relational Data model
  - data is stored in tables. Tables have one value per cell
  - based on mathematical model
- Network
  - data is stored in records (vertices) & association (edges)
  - complex model
- Hierarchical
  - data is stored in a tree structure with parent/child relationships
- Inverted List
  - tabular representation of data using indices to access the tables
  - almost relational, but it allows for non-atomic data values
- Object-Oriented
  - data stored as objects which contain identifier, name, lifetime, structure

Entity-Relationship Model
easy, simple to read

Entities

principle objects about which info is kept
are nouns, person, place, thing, event
shown as rectangles w/ name inside

Relationships

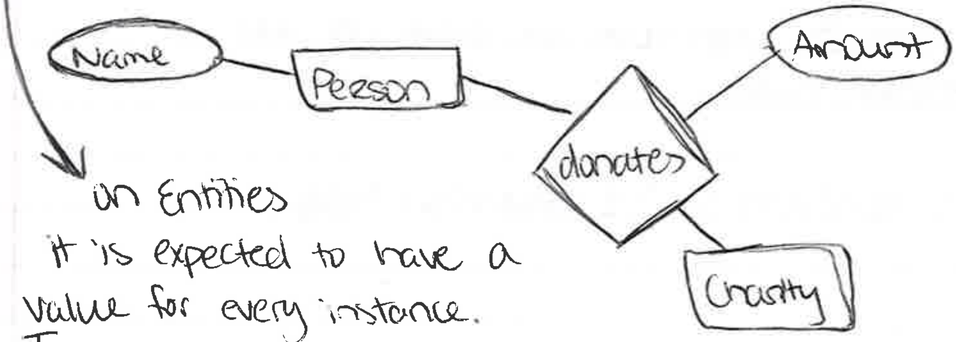
connect one or more entities together to show association
cannot exist without at least one entity
shown as diamond with name inside or just beside

Enrolled In

OR ◇ "Enrolled In"

Attributes

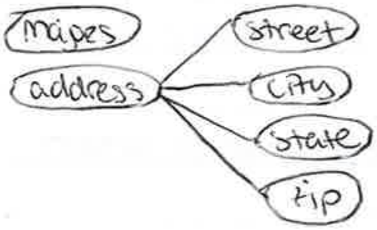
characteristics of entities or relationships
shown as oval



on Relationships
only expected to have a value when the entities come together. Referred to as 'intersection data

on Entities
it is expected to have a value for every instance.

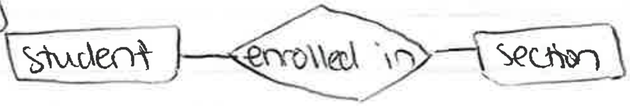
Types of Attributes
identifier (key) — ID
descriptor — Name
multivalued descriptor —
composite attribute —



Degree of a Relationship

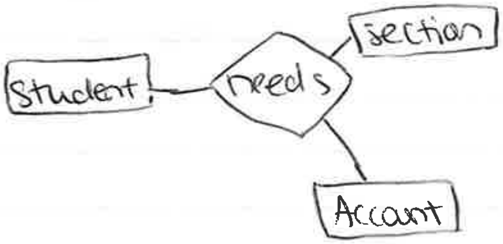
number of entities associated with it

binary



no limit to how many entities there can be in a relationship. we call them n-ary, where n is the degree

ternary



Connectivity of a Relationship

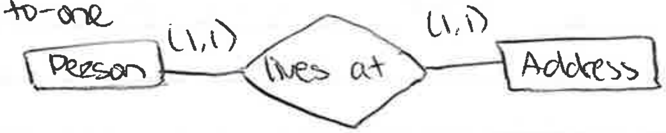
written as (minimum, maximum)

maximum → commonly 1 or a letter denoting many

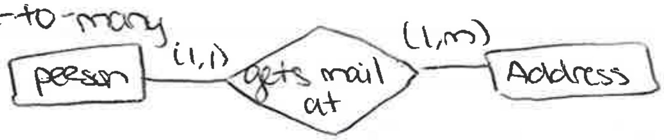
minimum → usually 0 or 1

Actual number is the cardinality

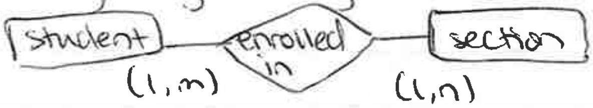
one-to-one



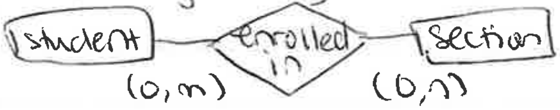
one-to-many



mandatory many-to-many

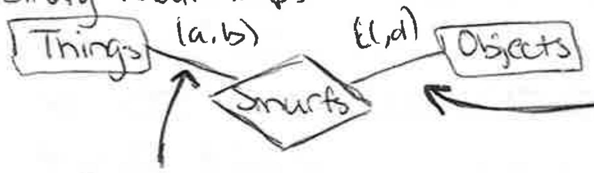


optional many-to-many



Reading Cardinalities

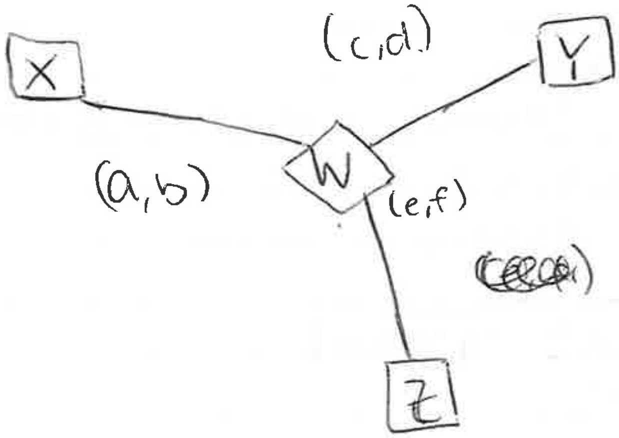
for binary relationships



for each Object that smurfs, there is a min of a and a max of c

for each Thing that smurfs, there are a min of c, and a max of d objects

for higher degree relationships

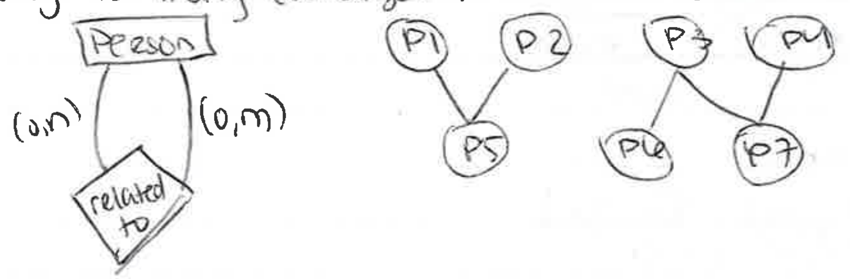


X+Y in relationship W, (min of e, max of f Z)
X+Z in relationship W, (min of c, max of d Y)
Y+Z in relationship W (min of a, max of b X)

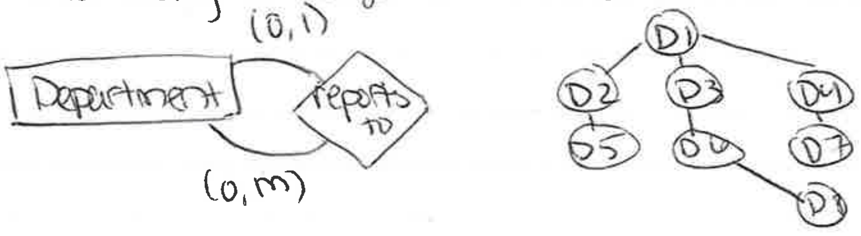
Recursive Relationships

entity that has a relationship with itself

Many-to-many (arranged in network structure



one-to-many (arranged in tree structure)



ER Diagrams Pt 2

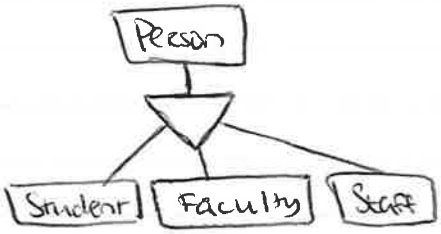
Inheritance

"IS a" - shows subtype IS a member of the supertype

"is part of" - shows that the supertype contains, or is made up of members of the subtypes

IS A Inheritance

happens when you have a supertype and one or more subtypes that are members of the supertype



Defining IS-A Inheritance

mutually exclusive. pick 1 — Generalization vs Specialization: supertype union of all subtypes and cannot exist without belonging to at least one subtype VS subtype entities specialize the supertype and can exist without being related to the subtypes

mutually exclusive. Pick 1 — Overlapped vs disjoint subtypes - instance of the supertype to be related to more than one subtype VS subtype entities are mutually exclusive and it is not possible ~~to~~ for an instance of the supertype to be related to more than one subtype