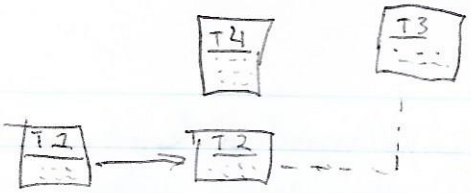
WWT1

Class notes - Ch. 5

ER Diagram:



- if this is developed, it'll generate the db through DDL
- OR db scripts could generate the ER Diagram

* - a subquery approach usually has a corresponding left join approach

* Learn CTE's (inner query is 1st for subqueries)

Database Ch. 5 - notes

Data Modeling w/ the Entity-Relationship Model

E-R Diagram: Entity-Relationship (* See top of page for graphic)

* Data modeling occurs in the requirements analysis step of the systems development life cycle in the systems analysis and design process

I. The Purpose of a Data Model

Data Model: a blueprint for a db design; generalized & non-DBMS specific
- Ex planning stage before building a house

II. The Entity-Relationship Model
- publ. by Peter Chen in 1976.
- Submodels were added later, & called the extended ER model, but is now essentially the standard

Entities
- something users want to track (CUSTOMER entity = data that's important to the customer)
  - Entities of a given type are grouped into an entity class (i.e. CUSTOMER, EMPLOYEE)

Entity Class: collection of entities to descr. by the structure of the entities in that class

Entity Instance: occurrence of a particular entity (i.e. CUSTOMER 123)
- entity class has many instances of an entity
- it's like 1 record of data in a table

Attributes

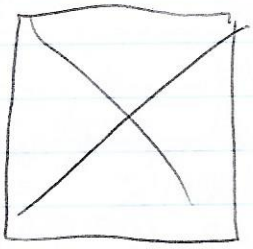
defn:
- describe characteristics (data columns)
- ER model assumes all instances of a given entity class have the same attributes
- Attributes in Ellipses vs Attributes in Rectangle (common)

Identifiers

defn:
- attributes that name, or id, entity instances
  - Ex. EmployeeNumber id's EMPLOYEE instances

Composite identifier: identifiers that consist of 2 or more attributes
- Ex. (ProjectName, TaskName)

- entities are displayed in 3 layers of detail today:



EMPLOYEE ← entity

EmployeeNumber	← identifier
EmployeeName Phone E-mail HireDate	← attributes

Relationships

defn:
- how entities can associate w/ one another
- ER Model contains both relationship classes & instances

Relationship classes: associations among ~~entities~~ entity classes

Relationship instances: " " among entity instances

Binary Rel: Entity 1 — Rel. Name — Entity 2

- Rel. names are given to describe the nature of the relationship

Degree: number of ~~relationship~~ entity classes in a relationship
- degree 3 = Ternary Rel.

- relationships in ER model are classified by their cardinality
  - Max Cardinality is max # instances than can be a part of a relationship instance
  - Min Cardinality is the opposite

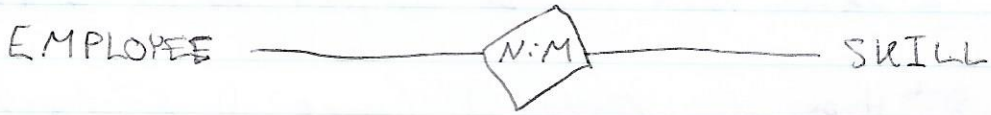
Max Cardinality *(1 = parent / N: child)*

- 3 Types:

1.) One-to-One relationship

2.) One-to-Many relationship

3.) Many-to-Many relationship



1a.) 1 EMPLOYEE instance is associated w/ 1 BADGE entity instance ("at most 1")

1b.) 1 EMPL. entity instance ~~can~~ ≠ can be associated w/ many COMPUTER entity instances, but only 1 COMPUTER instance is associated w/ only 1 EMPL. instance

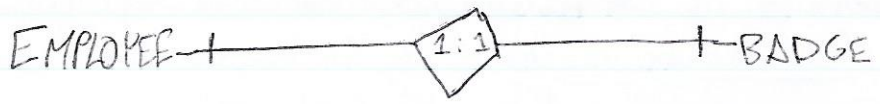
1c.) 1 EMPL. instance can have many SKILL instances and vise versa
- N:M highlights possibility of different ~~instances~~ cardinalities

* Ex. EMPL 1 has 9 skills, but that skill is associated w/ 3 EMPL instances

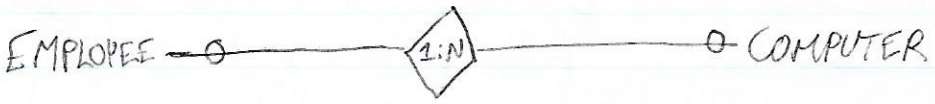
Min Cardinality : min # entities that must participate in a relationship
- 0 to 1 : (0 = optional / 1 = mandatory) ☆

- 3 Types:

1.) Mandatory - to- Mandatory Relationship



2.) Optional - to - Optional Relationship



3.) Optional - to - Mandatory Relationship

1a) entities are required on both sides

1b) an EMPL need not have an COMPUTER & vise versa

1c.) an EMPL must be assigned to at least 1 SKILL, but a SKILL need not be assigned to an EMPL.

- 4th type is M-to-O (swap of 3rd type)

ER Diagrams & Their Versions

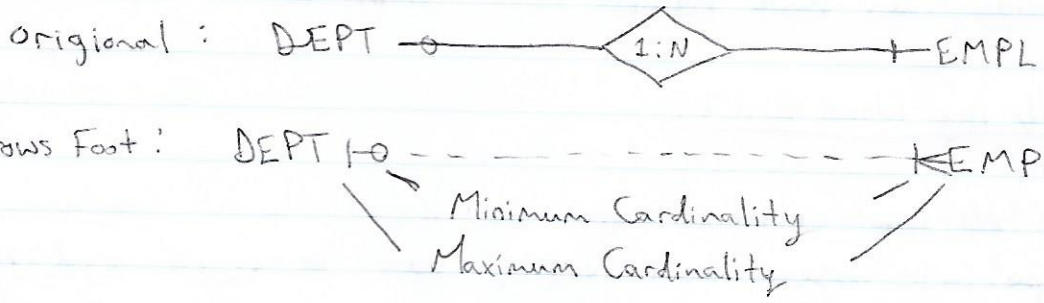
Variations of ER Model

1.) Information Engineering model: uses crow's feet to show the "Many"; called the IE Crow's Foot model

2) Integrated Definition 1, Extended (IDEF1X): incorp. basic ideas of ER model but uses different symbols
- national standard
- difficult to use (Appendix C)

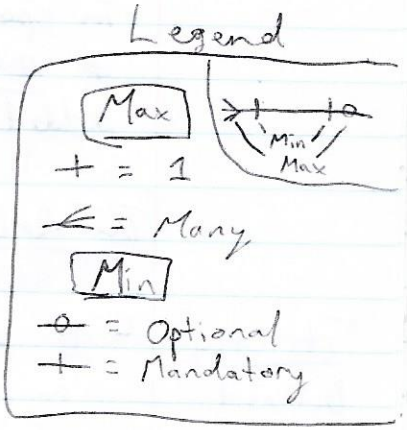
3) Unified Modeling Language: OOD spin on the model (Appendix D)

ER Diagrams Using IE Crow's Foot



Notation

a) —||—: Mandatory - One; exactly 1
b) —|<—: Mandatory - Many; one or more
c) —O|—: Optional - One; zero or one
d) —O<—: Optional - Many; zero or more

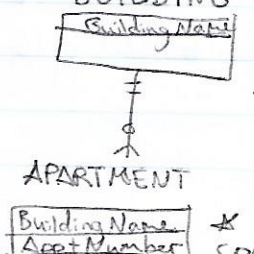


Strong Entities & Weak Entities
- Strong: represents something that can stand alone (i.e. PERSON)
- Weak: entity whose existence depends on presence of another entity

ID-Dependent Entities

ID-Dep.: an entity whose identifier includes the identifier of another entity

Example:



- APPT needs both ~~Building Number~~ Building Name and Appt Number to determine APPT, making APPT ID-Dependend on BUILDING

identifying relationship:
-  rounded corners represent the ID-Dep. entity
- solid line b/w ID-Dep. entity & it's parent

nonidentifying relationship: ~~neither~~ neither entity is ID-Dependent
- dashed line b/w 2 strong entities
- no ID-Dep entities in that particular relationship

* - 1 entity can have multiple relationships, all w/ different properties

Non-ID Dep Weak Entities

- All ID-Dep. entities are weak
- in special cases, an entity can be weak and non-ID-Dep. For this, we use a nonidentifying rel. w/ a note added to the data model, saying the entity is weak

The Ambiguity of a Weak Entity

Strictly Defined
- If a weak entity = any entity whose db presence depends on another entity, then any entity w/ a minCard. of 1 to a second entity = weak
  - Ex. if STUDENT must have an ADVISOR, than ~~Stude~~ STUDENT is weak b/c it cant be stored w/o an ADVISOR

Narrowly Defined
- weak = an entity must logically depend on another entity
  - Ex. in this case, aside from business rules, a STUDENT can logically exist w/o ~~an~~ an ADVISOR

Ex. 2: an APPT cannot exist w/o a BUILDING

Subtype Entities (extended ER)

- a subtype is a special case of another entity (supertype)
  & STUDENT ← supertype

  → GRADUATE
  → UNDERGRADUATE ⇐ Subtypes

Notation:

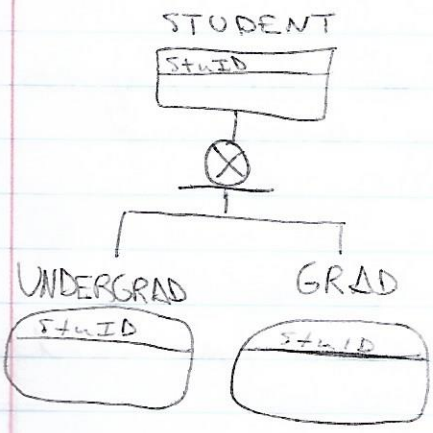
Subtype = O
exclusive sub = ⊗

- Discriminator: an attribute of a supertype that determines which subtype is appropriate for a given instance
  - not all supertypes have a discriminator

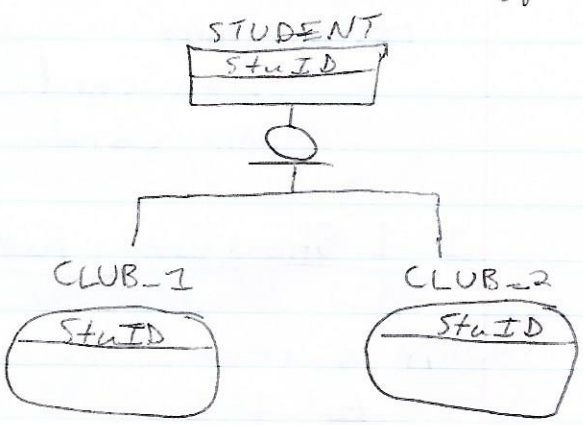
exclusive Subtype: supertype instance is related to at most 1 subtype

inclusive Subtype: relates to 1 or more subtypes
  - for this reason ↑, it does not need a discriminator

Ex. Exclusive Subtype w/ Discriminator



Ex. Inclusive Subtypes



* - most important reason for subtypes in a data model is to avoid value-inappropriate nulls

* refer to pg210, Figure 5-14 for everything in this section

III. Patterns in Forms, Reports, and ER Models

- to speak to users where they understand, we must infer data indirectly from user docs, convos, & behaviors

Ways:

1.) Study the user's forms & reports
   - From here, we can begin to comprise a data model (or entities/rel at least)
   - also can use to validate an already-created data model

* Data-first design: the structure of forms & reports determine the structure of the data model

Strong Entity Patterns

- 3 relationships possible: 1:1, 1:N, N:M
- max/min cardinality must be found when modeling relationships
  - max card. can be found on forms/reports
  - min card: will need to be determined

1:1 Strong Entity Rels

- if a relationship is 1:1, we can set max card. to 1:1

Ex:

Relationship is:
- 1:1
- Optional: Optional
- Not ID-Dep
- Optional: One
- nonidentifying rel (- -)

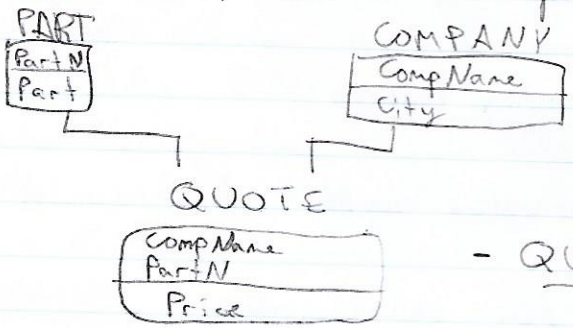
- When trying to determine relationships, you can:
  - think logically
  - ask someone
  - look at the business setting

ID-Dependent Relationships

- 3 principle patterns: multivalued attribute, version/instance, association

Association Pattern to Entity

- similar to N:M strong relationship
- ~~Price~~ attr. is ~~not~~ an attribute to the combo of 2 entities, not 1 in particular



- this can occur b/w 2 or more entities
- QUOTE = associative entity

- QUOTE is ~~HDEP~~ ID-Dep on both entities

Multivalued Attribute Pattern

- Today: attr. must have a single value/row

Ex. 1 user w/ >1 phone number

- solution: phone is not an attribute of USER, but an ~~HDEP~~ ID-Dep entity w/ a 1:N relationship
  - new entity, PHONE, would include a composite identifier (user ID, Phone Number)

- model can be extended for multivalued attributes

* - if multivalues needed, make a new entity w/ 1:N relationship

The Archetype (version) / Instance Pattern

- occurs when 1 entity represents an instance of another entity
  - Ex: x-model is instance of x archetype
    - a yacht manuf. has various yacht-designs
      - each yacht is an instance of a particular yacht-design · archetype
- the child entity (yacht) is alway an ID-Dep entity to its archetype
- if yacht can be identified by more characteristics than specified in the ~~id~~ entity, then its added, it will end up being weak, but no longer ID-Dep

Mixed Identifying & Nonidentifying Patterns

- some patterns involve both id & nonid rels

The Line-Item Pattern (Ex. sales order)

- the identifier of a line is a composite of the identifier of a particular line & the identifier of a particular order
  - entries for line items are always ID-Dependent on the order in which they appear
- line items are existence dependent on orders. If an order is deleted, the line item corresponding to it cannot exist

* Figure 5-33 pg 223

Other Mixed Patterns

- look for mixed pattern when a strong entity has a multiv. composite group
- Figure 5-35

The For-Use-By Pattern

- if I ever see a "For Use by ... only" (on a form), this indicates the need for subtype entities
  - Ex. A Fishing-License entity can either have a Commercial-License or a Sport-License
- Figure 5-37 (pg225) * relationships can be different

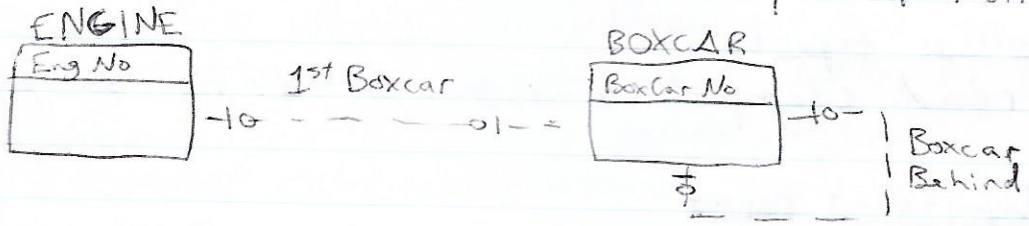
Recursive Patterns (unary relationship)

- occurs when an entity type has a relationship to itself (w/ all 3 types of rels)

1:1 Recursive

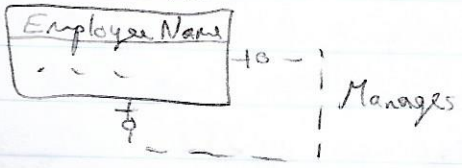
Ex. ~~RAT~~ db for railroad
entity: BOXCAR

- each boxcar has 1 in front of it, except for the 1st
- each has 1 behind, except for the last
  - 1:1 b/w boxcars, w/ an optional rel. for 1st & last



1:N Recursive

- organizational charts (1 manager manages several others)
* - breadcrumbs, rabbit trails ....



N:M Recursive

- manufacturing apps, where they can represent bills of materials

Ex: car : engine, tranny, interior, exterior
engine: nuts, bolts, cylinders, spark plugs
tranny: gears, clutch, clutch pads
interior: seats, stereo, speakers, steering wheel
exterior: paint, body, sunroof, wheels, tires

it keeps going like a hierarchy

- family tree is another example

IV. The Data Modeling Process

- analyzation of user requirements to constructs a data model from data sources, forms, reports, and user interviews
- iterative process → model can change as more forms are analyzed
- model also needs user validation

Examples:

The College Report

- read example to Figure 5-44/45 on pg 229

The Department Report

- this report is representative of having to add additional requirements as more are analyzed
- see example to Figure 5-46/47 on pg 230

The DepartMent/Major Report

- see example to Figure 5-48 on pg 232

The Student Acceptance Letter

- includes data about student, major, department, and advisee
- problems w/ parent/child relationships in data model to produce the letter perfectly in >1 case
  - see example & Figure 5-50/51 on pg 233

- it's typical to revise the data model many times throughout the process

Review (key terms)

1) ID-Dep Entity: identifier includes the identifier of another entity
   - identifying relationship (parent required, child isn't)
2) Weak Entity: existence depends on the presence of another entity
   - all ID-Dep entities
3) Subtype Entity: special case of supertype; exclusive or inclusive
   - avoid value-inappropriate nulls
4) relationship b/w an entity and itself is a recursive relationship (can be 1:1, 1:N, or N:M)
5) relationships among nonsubtype entities are HAS-A rels. relationships among supertype/subtype entities are IS-A rels.
6) 1:1, 1:N, N:M strong entity patterns
7) ID-Dependent relationship patterns: association, multivalued attribute, version/instance
   - mixed Id & nonId patterns as well
8) For-Use-By pattern indicates the need for subtypes
9) Subtypes- differ b/c of different attributes or relationships
10) Data modeling process is iterative (evidence gathering)
11) Will not always find minimum cardinality from a form/report