

## Announcements

- Important course dates.
  - Assignment 1 out May 20 due June 9 (at the tutorials)
  - Project out June 23 in July 28 + oral test
- See course web page for details.
- Lecture notes: on [mathlab.utoronto.ca: /courses/courses/csc43s13/koudasni](http://mathlab.utoronto.ca/courses/csc43s13/koudasni)
- Bahar Gadiri office hours: Wednesday 9-10 (400A)
- Mehrnaz Najafi office hours: Thursday 3-4 (400A)
- Tutorials: please attend the section you belong.
- Assignment due on start of corresponding tutorial session.
- Clarification: Assignments conducted alone; for the project you may (if you wish) form teams of two.

CSC43

N.Koudas

1

## Today's Material

- More on ER
- Introduction to the relational model

CSC43

N.Koudas

2

## More Design Issues

1. Subclasses.
2. Keys.
3. Weak entity sets.

CSC43

N.Koudas

3

## Subclasses

Subclass = special case = fewer entities = more properties.

- Example: A cartoon is a kind of movie. In addition to the properties (= attributes and relationships) of movies, there is a *voices* attribute for cartoons.

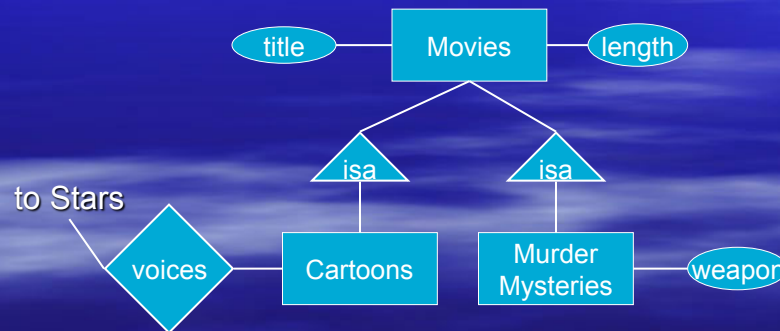
CSC43

N.Koudas

4

## E/R Subclasses

- Subclasses form a tree (no multiple inheritance).
- isa* relationships indicate the subclass relation



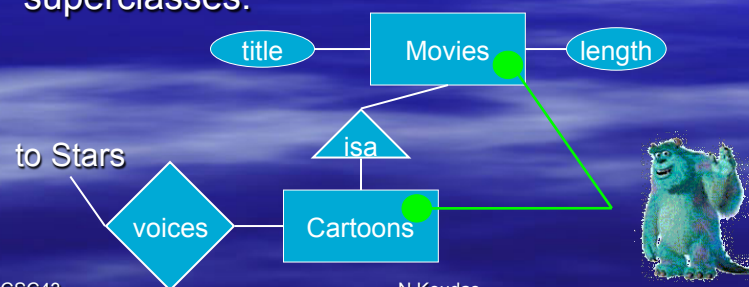
CSC43

N.Koudas

5

## Different Subclass Viewpoints

- *E/R viewpoint*: An entity has a *component* in each entity set to which it logically belongs. Its properties are the union of the properties of these E.S.
- *Object-oriented viewpoint*: An object (entity) belongs to exactly one class. It *inherits* properties of its superclasses.



CSC43

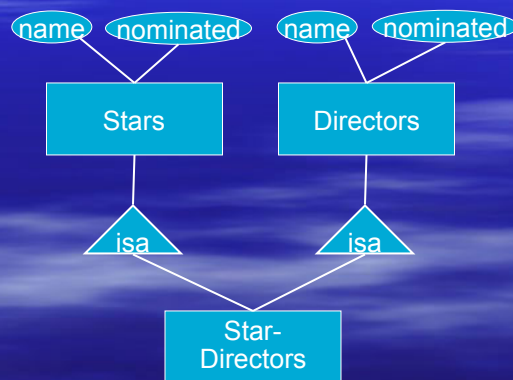
N.Koudas

Courtesy of Pixar

6

## Multiple Inheritance

Theoretically, an E.S. could be a subclass of several other entity sets.



CSC43

N.Koudas

7

## Problems

How should conflicts be resolved?

- Example: nominated means Oscar nominated as an actor for *Stars*, and as a director for *Directors*. What does it mean for *Star-Directors*?
- Need ad-hoc notation to resolve meanings.
- In practice, we shall assume a tree of entity sets connected by *isa*, with all “isa’s” pointing from child to parent.

CSC43

N.Koudas

8



## Keys

Consider an entity set  $E$ . A key is a set of attributes  $K$  of  $E$  such that, given two entities  $e_1$  and  $e_2$  of  $E$ ,  $e_1$  and  $e_2$  cannot have identical values in  $K$

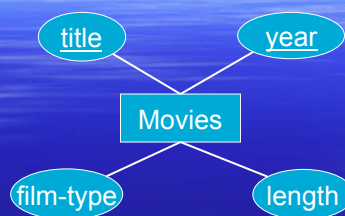
- In E/R model, every E.S. must have a key.
  - It could have more than one key, but one set of attributes is the “designated” key.
  - A key is minimal; otherwise it’s a superkey.
- In E/R diagrams, you should underline all attributes of the designated key.

CSC43

N.Koudas

9

## Example: A Multi-attribute Key



Possibly, the combination of *title* + *length* also forms a key, but we have not designated it as such.

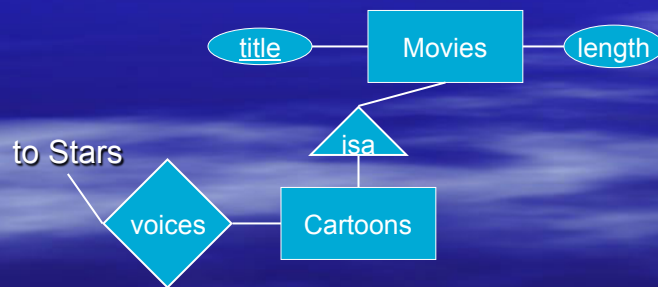
CSC43

N.Koudas

10

## Example

Suppose *title* is the key for *Movies*.  
It makes sense to use *title* as key for *Cartoons* also.  
Thus, we use root key as key for all subclasses.



CSC43

N.Koudas

11

## Weak Entity Sets

Consider an entity set  $E$ . It's possible that  $E$ 's key comes not (completely) from its own attributes, but also from the keys of one or more E.S.'s to which  $E$  is linked (through a *supporting* many-one relationship).

- Entity set  $E$  is called a **weak** E.S.
- It is represented by putting double rectangle around  $E$  and a double diamond around each supporting relationship.
- Supporting relationship should be many-one (includes 1-1)
  - With many-many, we wouldn't know which entity provided the key value.

CSC43

N.Koudas

12

## Example: Logins (Email Addresses)

Login name = user name + host name, e.g. nick@cdf.toronto.edu

- A “login” entity corresponds to a user name on a particular host. The *Login* entity doesn’t record the host, just the user name, e.g., nick.
- Key for a login = the user name at the host (which is unique for that host only) + the IP address of the host (which is unique globally).



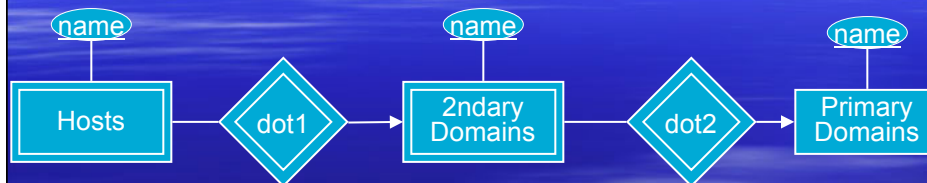
CSC43

N.Koudas

13

## Example: Chain of “Weakness”

Consider IP addresses consisting of a primary domain (e.g., edu), subdomain (e.g., cdf.toronto), and host (e.g., eddie).



- Key for primary domain = its name.
- Key for secondary domain = its name + name of primary domain.
- Key for host = its name + key of secondary domain = its name + name of secondary domain + name of primary domain.

CSC43

N.Koudas

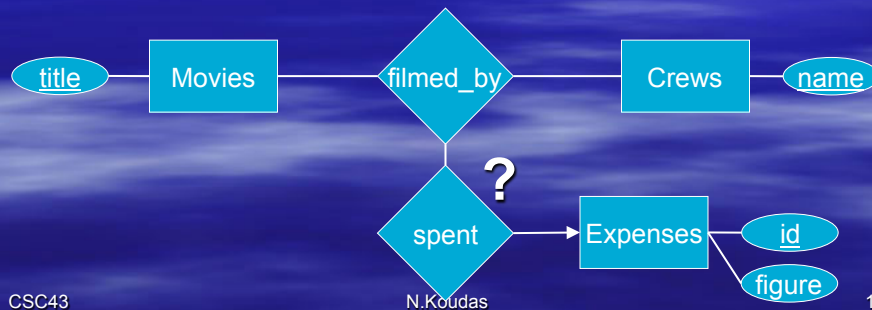
14

## Additional example

Consider a relationship, *filmed\_by*, between two entity sets, *Movies* and *Crews*



How can we add *Expenses* to the mix?



CSC43

N.Koudas

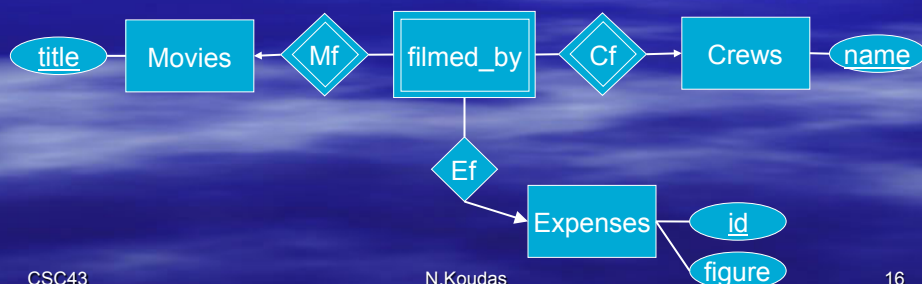
15

## Additional Example (cont' )

Solution: convert *filmed\_by* into a weak entity set.



And then add *Expenses* to it...



CSC43

N.Koudas

16



## Design Principles

Setting: client has (possibly vague) idea of what is required.  
You must design a database that represents these thoughts  
and only these thoughts.

Important note:

**Avoid redundancy**, i.e., saying the same thing more than once.  
It wastes space and encourages inconsistency.

Example

Good:



CSC43

N.Koudas

17

## Example

This is a bad design...



Yeap! That's also a bad design...



CSC43

N.Koudas

18

## Use Schema to Enforce Constraints

The design *schema* should enforce as many constraints as possible.

- Don't assume future data will follow assumptions.

Example:

- If a user must associate only one director with a movie, don't allow sets of directors and count on users to enter only one director per movie.

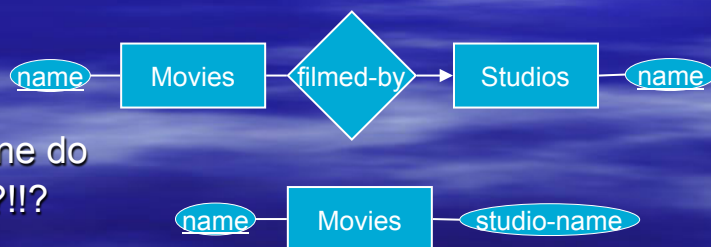
CSC43

N.Koudas

19

## Entity Sets Vs. Attributes

- You may be unsure which concepts are worthy of being entity sets, and which are handled more simply as attributes.



CSC43

N.Koudas

20

## Intuitive Rule for E.S. Vs. Attribute

Make an entity set only if either:

1. Is more than the 'name' of something; *i.e.*, it has non-key attributes or it has relationships with a number of different entity sets, or
2. Is the "many" in a many-one relationship.

CSC43

N.Koudas

21

## Example

The following design illustrates both points:



- *Studios* deserves to be an E.S. because we record *addr*, a non-key attribute.
- *Movies* deserves to be an E.S. because it is at the "many" end.

CSC43

N.Koudas

22

## Don't Overuse Weak E.S.

- There is a tendency to feel that no E.S. has its entities uniquely determined without following some relationships.
- However, in practice, we almost always create unique ID's to compensate: social-security numbers, OHIP's, etc.
- The only times weak E.S.'s seem necessary are when:
  - a) We can't easily create such ID's; e.g., no one is going to accept a "species ID" as part of the standard nomenclature (species is a weak E.S. supported by membership in a genus).
  - b) There is no global authority to create them, e.g., crews and studios.

## The Relational Model



## Relational Model

- Table = relation.
- Column headers = attributes.
- Attribute domains (atomic)
- Row = tuple
- Relation schema = name(attributes) + other structure info., e.g., keys, other constraints.  
Example: *stars-in(movie-name, star-name)*
  - Order of attributes is arbitrary, but in practice we need to assume the order given in the relation schema.
- Relation instance is current set of rows for a relation schema.
- Database schema is a collection of relation schemas.

Relation stars-in	
Movies	Stars
Basic Instinct	Sharon Stone
Total Recall	Arnold Schwarzenegger
Total Recall	Sharon Stone
...	...

CSC43

N.Koudas

25

## Why Relations?

- Very simple model.
- *Often* a good match for the way we think about our data.
- Data base query languages (e.g., SQL) manipulate such relational tables..

CSC43

N.Koudas

26

## Relational Design

We start with an E/R design. Then, the simplest approach (but not always best) is :

- convert each E.S. to a relation and
- convert each relationship to a relation.

Example:



*Movies(title, year, length, film-type)*

Note: E.S. attributes become relational attributes

## Keys in Relations

An attribute (or set of attributes)  $K$  is a *key* for a relation  $R$  if we expect that in no instance of  $R$  will two different tuples agree on all the attributes of  $K$ . We indicate a key by underlining the key attributes. (pretty much what we did in E/R)

Example:

If *title* is a key for *Movies*, then  
*Movies(title, year, length, film-type)*

## From E/R Relationships to Relations

The constructed relation has the following attributes:

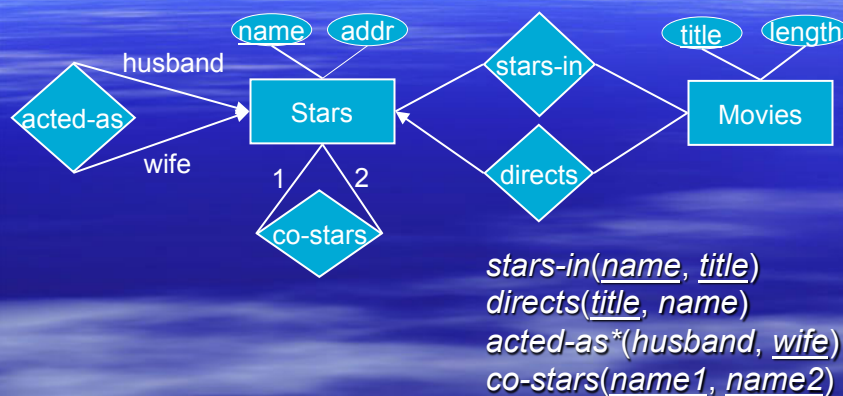
- The *key* attributes of each E.S. that participates in the relationship.
- Any attributes that belong to the relationship itself.
- Renaming attributes is necessary if an E.S. has multiple roles in the relationship

CSC43

N.Koudas

29

## From E/R Relationships to Relations (cont.)



(\*) For *acted-as*, we can choose either *husband* or *wife* as key

CSC43

N.Koudas

30

## Combining Relations

Common case: Combine relation for an E.S.  $E$  with the relation for some many-one relationship from  $E$  to another E.S.  $F$

Example: Combine *Movies*(title, length) with *directs*(title, name) to get *Movies-new*(title, length, director-name).

However danger in pushing this idea too far: redundancy

Example: Combine *Movies*(title, length) with *stars-in*(name, title) to get *Movies-stars*(title, name, length, )

title	name	length
Star Wars II	Ewan McGregor	142
Star Wars II	Natalie Portman	142

CSC43

N.Koudas

31

## Weak Entity Sets & Relationships to Relations

- Relation for a weak E.S. must include its full key (*i.e.*, attributes of related entity sets) as well as its own attributes.
- A supporting (double-diamond) relationship yields a relation that is actually redundant and should be deleted from the database schema.

CSC43

N.Koudas

32



## Weak Entity Sets & Relationships to Relations (cond.)



- In *At*, *hname* and *hname2* must be the same host, so delete one of them.
- Then, *Logins* and *At* become the same relation; delete one of them.
- In this case, *Hosts*' schema is a subset of *Logins*' schema. Delete *Hosts*?

CSC43

N.Koudas

33

## Subclasses to Relations

Three approaches:

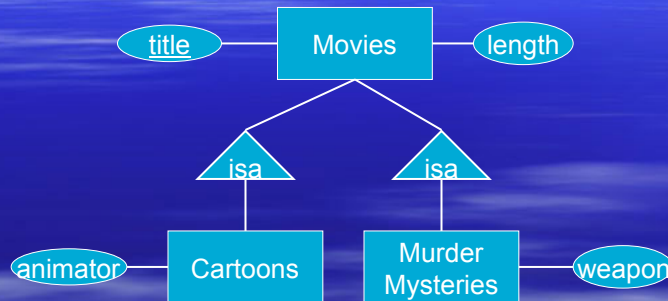
1. Object-oriented: each entity is in one class. Create a relation for each class, with all the attributes for that class.
  - Don't forget inherited attributes.
2. E/R style: an entity is in a network of classes related by *isa*. Create one relation for each E.S.
  - An entity is represented in the relation for each subclass to which it belongs.
  - Relation has only the attributes attached to that E.S. + key.
3. Use nulls. Create one relation for the root class or root E.S., with all attributes found anywhere in its network of subclasses.
  - Put NULL in attributes not relevant to a given entity.

CSC43

N.Koudas

34

## Example



CSC43

N.Koudas

35

## Subclasses to Relations: An Example

OO Style:

<u>Title</u>	<u>Length</u>	<u>Weapon</u>	<u>Title</u>	<u>Length</u>	<u>Animator</u>	<u>Title</u>	<u>Length</u>
Episode I	133	Lightsaber	Monsters	92	John	Amadeus	92
...	...		...	...		...	...

E/R Style:

<u>Title</u>	<u>Length</u>	<u>Title</u>	<u>Weapon</u>	<u>Title</u>	<u>Animator</u>
Episode I	133	Episode I	Lightsaber	Monsters	John
Monsters	92	...	...	...	...
...	...				

Using Nulls:

<u>Title</u>	<u>Length</u>	<u>Weapon</u>	<u>Animator</u>
Episode I	133	Lightsaber	Null
Monsters	92	Null	John

CSC43

N.Koudas

36



That's it for today...

CSC43

N.Koudas

37