CS 235: Introduction to Databases

Svetlozar Nestorov

Lecture Notes #3

Overview

- Weak entity sets and keys
- Design principles
- Examples

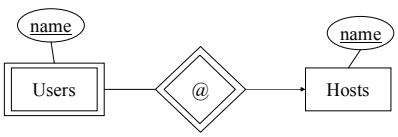
Weak Entity Sets

- Sometimes an E.S. *E*'s key comes not (completely) from its own attributes, but from the keys of one or more E.S.'s to which *E* is linked by a *supporting* many-one relationship.
- Called a *weak* E.S.
- Represented by putting double rectangle around *E* and a double diamond around each supporting relationship.
- Many-one-ness of supporting relationship (includes 1-1) essential.
  - With many-many, we wouldn't know which entity provided the key value.

Example: Email Addresses

- Email address = user name + host name, e.g., *evtimov @ cs.uchicago.edu*.
- Email address corresponds to a user name on a particular host.
- Once on a host, you only need user name, e.g., *evtimov*
- Key for an email = the user name at the host (which is unique for that host only) + the IP address of the host (which is unique globally).

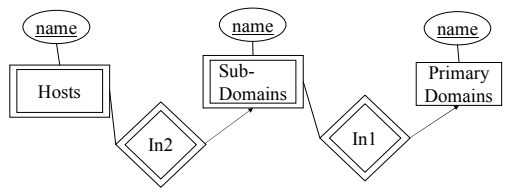
Email Addresses



- Design issue: Under what circumstances could we simply make user-name and host-name be attributes of email, and dispense with the weak E.S.?

Example: Chain of *Weakness*

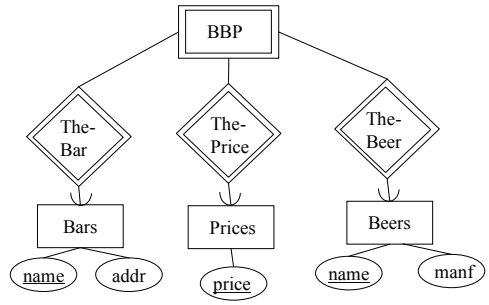
- Consider IP addresses consisting of a primary domain (e.g., *edu*), subdomain (e.g., *uchicago*), and host (e.g., *cs*).



Chain of Keys

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

All *Connecting* Entity Sets Are Weak



All *Connecting* Entity Sets Are Weak

- In this special case, where bar and beer determine a price, we can omit price from the key, and remove the double diamond from ThePrice.
- Better: price is an attribute of BBP.

Constraints

- Part of the schema
- Keys
- Single value constraints
- Referential integrity constraints
- Domain constraints
- General constraints

Single Value Constraints

- Each attribute has a single atomic value
  - No set attributes!
- Many-one, one-one relationships

Referential Integrity

- Exactly one value
  - Compare with at most one for single-value constraints.

Design Principles

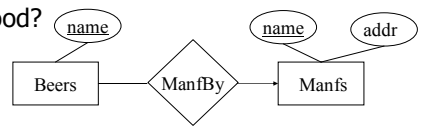
- Setting: client has (possibly vague) idea of what he/she wants. You must design a database that represents these thoughts and only these thoughts.
- Avoid redundancy.
  - Wastes space and encourages inconsistency.
  - Intuition: something is redundant if it could be hidden from view, and you could still figure out what it is from the other data.
- Avoid intermediate concepts.

Design Principles

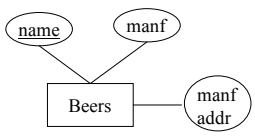
- Faithfulness to requirements.
  - Remember the design *schema* should enforce as many constraints as possible. Don't rely on future data to follow assumptions.
  - Example: If registrar wants to associate only one instructor with a course, don't allow sets of instructors and count on departments to enter only one instructor per course.

Good and Bad Design

Good?

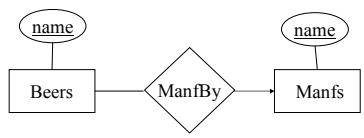


Bad?



Good and Bad Design

Bad?



- Question: Why is it OK to have *Beers* with just its key as attribute? Why not make set of beers an attribute of manufacturers?

Exercise Problem 2

- E/R diagrams

Exercise Problem 3

- Multiway relationships