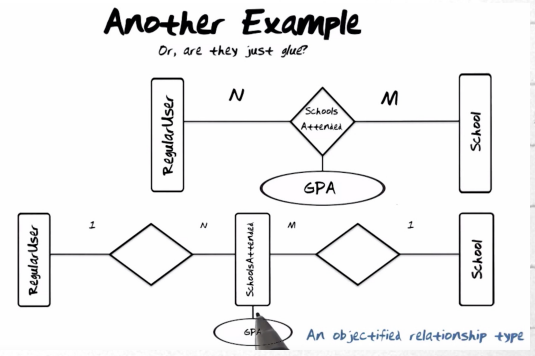
Lesson 4 – EEMR

2018-08-30

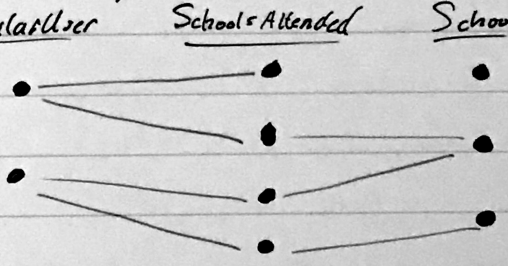
Another Relationship Entity Example:



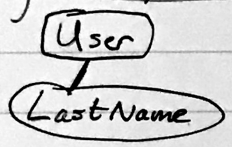
=> Example: SchoolsAttended need to be an entity type, not a relationship type

This is an objectified relationship type

=> We need two relationship types to bridge the gap; two 1-N types



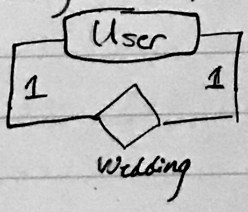
Messing with your brain – Part 1: Property or Entity Type?



→ Generally "User" is an Entity type, and LastName is a Property Type

→ However, this depends: "Andersen", "Anderson", "Carpenter", "Kim" are all last names. But, in a genealogy database, it also could be a central entity type

Messing with your brain – Part 1: Relationship type or Entity Type?



=> In the "Wedding Planner", the person who serves as the relationship to the wedding is also the person getting married.

=> Context is very important; nothing about reality is automatic when it comes to mapping databases

Summary: Objectified relationship types are used for when a relationship type needs to be an entity type. When mapping databases, never assume that things are properties, entities, relationships, or whatever.

Lesson 4 - EERM 2018-08-30

What can the EER do?: Classification, Aggregation, Generalization

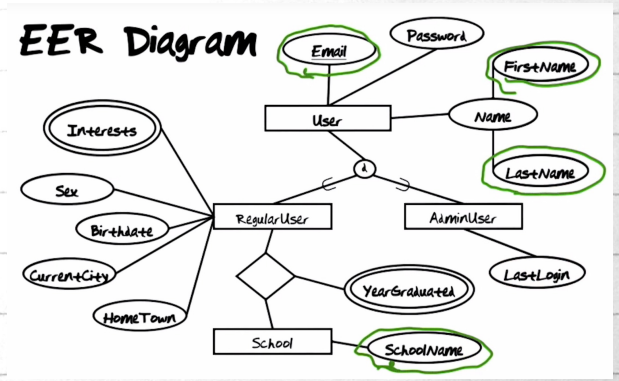
→ Classification means being able to define entity types

→ Generalization is supported by super/sub types

→ Aggregation = ? The ERM does not explicitly support aggregation!

↳ You can always fudge it, but it's not a good idea

What's the result type of a query?



→ You want a list of information about regular users, and the schools they attended

→ The list of information has no type!

→ Therefore, the result cannot be operated on with a query languages.

→ There are no EER DB products, and DBMS don't use EERM!

Relational Model - Theoretical Foundation v2

→ Need to look at three things in a data model: data structures, constraints, and operations (relational algebra and calculus)

→ Tuple calculus (SQL) and domain calculus (QBE)

Data structures:

→ A domain, D, is a set of atomic values (has no meaning inside)

→ In relational model, the only structure-type is relations

→ A relation, R, is a subset of the set of ordered n-tuples

$$R \subseteq \{\langle d_1, d_2, \ldots, d_n \rangle \mid d_i \in D_i, i = 1, \ldots, n\}$$

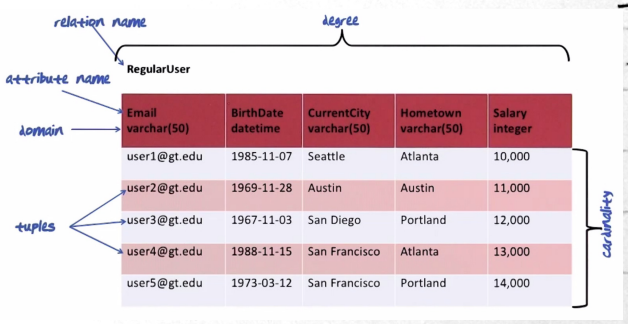
→ Each element "d_i" is pulled from ($\in$) the domain "D_i"

→ An attribute, A, is a unique name given to a domain in a relation helping us interpret domain values

Lesson 4 - EEMR

2018-08-30

Big Deal!



Email varchar(50)	BirthDate datetime	CurrentCity varchar(50)	Hometown varchar(50)	Salary integer
user1@gt.edu	1985-11-07	Seattle	Atlanta	10,000
user2@gt.edu	1969-11-28	Austin	Austin	11,000
user3@gt.edu	1967-11-03	San Diego	Portland	12,000
user4@gt.edu	1988-11-15	San Francisco	Atlanta	13,000
user5@gt.edu	1973-03-12	San Francisco	Portland	14,000

→ We illustrate relations by tables
→ Tuples are data, or rows
→ Domains are data types
→ Degree is number of columns
→ Column names are attribute names
→ Number of tuples is cardinality

Big deal: The value of a relation is independent of attribute order and tuple order!

* Quiz Knowledge Check

Constraints:

→ Major types: Keys and primary keys
→ Integrity: Entity and referential
→ Primary keys cannot be NULL!
→ Other implication: when primary keys are used as colums in another table, the set of emails in the other table must be a subset of the original