

2018-08-29

Lesson 4 - Extended Entity Relationship Model (EER Model)

Entity type and surrogates.

Rectangle *

User → Time invariant representation of a set of users

→ The users can be thought of as surrogates

→ Within entity relationship diagram, all entity type names must be unique

Ellipses *

Single-valued Properties: Property types are represented by ellipses



→ Property values are lexical, visible, audible

→ they are things that name other things

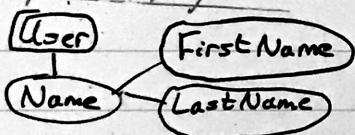
Underline *

Identifying properties: Identifying property types are underlined



→ Implication is that identifying properties must be unique; at most one instance

Composite properties: → Name is composed from FirstName and LastName



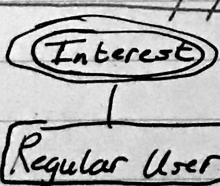
FirstName

LastNames

→ Is this one database column?

Double *
ellipses

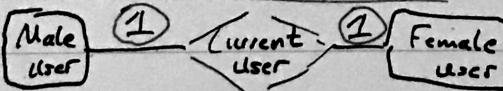
Multi-valued properties: → Modeled by double-ellipses



→ Can be just one, or multiple values

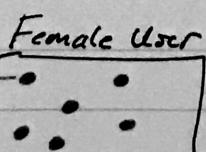
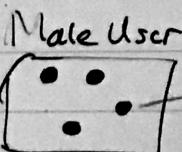
Relationship types
are diamonds

1-to-1 Relationship types → Represented by diamonds



→ Notation of 1-1 is the cardinality of the relationship

Partial *
Functions



<- Partial Function ->

→ Since some males and females are unmarried, this mapping is a partial function

Summary: EERM diagrams represent rectangles as surrogates, ovals as single-valued properties, underlined ovals as identifying properties, and double-ovals as multivalued properties. Finally, diamonds represent 1-to-1 relationship types.

Lesson 4: EERM

1- To - Many Types of Relation:



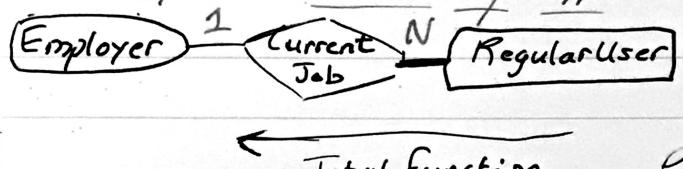
→ Partial function is only one way

Partial Function

→ Employer can be connected to multiple regular users

- * → Partial function because some users do not have employers and because each regular user is mapped to at most 1

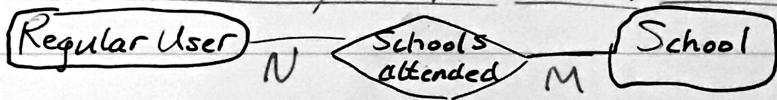
Mandatory 1-N Relationship Types



→ This is mandatory because current regular users must

participate in the current job relationship ; they cannot be unlinked

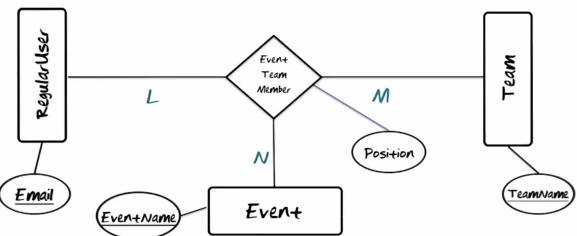
N-M Relationship Types (Many-to-many)



- Both sides can have many relationships to entities on the other side

N-ary relationship types

N-ary relationship types



→ All previous types have been binary

→ Event team member is a ternary relationship type

→ RegularUser , Team, and Event can all be many relationship types

→ Rare to see relationship degrees higher than 2

Summary : One to many relationships are partial functions towards the "1" entity whereas mandatory 1-N relationships are total functions . N-M relationships are more common, and N-ary relationships greater than 2 or 3 are rare.

* Partial function one way

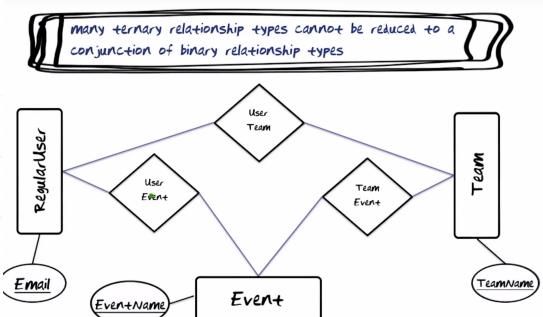
* Notice the bold single line

* Multiple connections

*

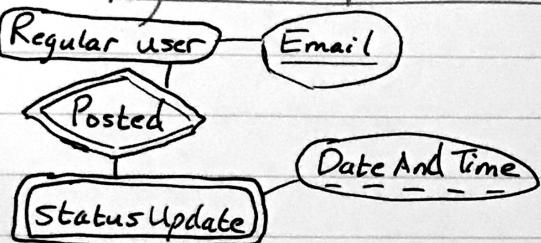
Lesson 4 - EERM

Many-relationship types



- Many ternary relationship types cannot be reduced to a conjunction of binary relationship types
- Relationships are much more narrow:
"A particular user on a particular team participates in a particular event."

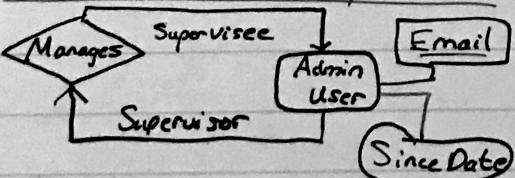
Identifying relationships / weak entity types:



→ User posts multiple updates during the day, but DateAndTime could have multiple users posting at that same time

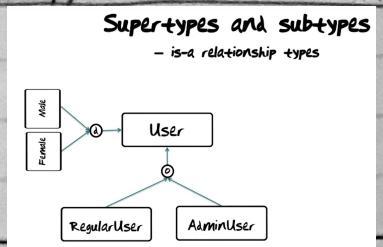
- * → Need to model: each status update is identified by user who posted it, and DateAndTime is unique for that status update by that user
- StatusUpdate cannot: (1) exist without RegularUser, and (2) cannot exist without RegularUser
- Email + Date Time identifies StatusUpdate
- Entity type is weak because they cannot exist without others
- Date Time by itself is a partial identifier

Recursive relationship types:



- Recursive because it relates an entity type to itself
- Relationship sides have roles
- "D" means disjointness: user cannot be both male and female
- "O" means overlap: a user can be a regular user and admin user

Super-types and sub-types

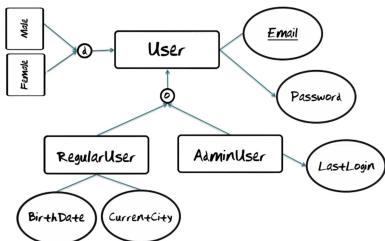


Summary: Many relationship types don't necessarily have nice binary constituents. Weak entity types are weak because they cannot exist without other types. Partial identifiers are dotted underlines. Some relationships are recursive, and have super and sub-types.

Lesson 4: EERM

Super/Sub-types and inheritance:

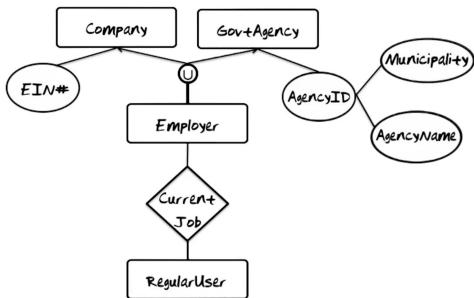
Supertypes and subtypes - inheritance



- Regular User and Admin User will inherit Email and Password
- User by itself will not be able to see BirthDate, CurrentCity or last login

Union Entity Type:

Union entity types



- Union entity types are like an "if-else" statement. Ex: Employer can either be "Company" or "GovtAgency"
- $\text{Employer} \subseteq \text{Company} \sqcup \text{GovtAgency}$
- "Employer is a subset of Company union Govt Agency" i.e. employer must be either a company or GovtAgency

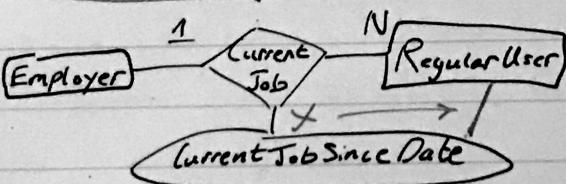
*First rule
*See rule

→ $\text{Company} \cap \text{Govt Agency} = \emptyset$ Intersection of "Company" and "Govt Agency" is empty. Employer cannot be both Company and Govt Agency

Thing, Relationship, Property v2 → Things to consider

- Is something a thing, relationship, or property?
- Does EERM support fundamental types of abstraction?
- On the EERM, what would a query be?

* Are relationships entities?



- Relationships may have attributes
 - For 1-N and 1-1, attributes may be moved to the entity on the "many" side (either)
- ↳ Important

Summary: Super and sub types can have inheritance just like objects. Union types define "if-and-or-xor" relationships. For relationship properties, rule is to move properties to the "many" side.