ER to Relational Mapping

Georgia Tech

ER to Relational Mapping

- Step 1: Strong Entities
- Step 2: Weak Entities
- Step 3: Binary 1:1 Relationships
- Step 4: Binary 1:N Relationships
- Step 5: Binary M:N Relationships
- Step 6: Multivalued Attributes
- Step 7: N-ary Relationships (not covered)
- Step 8: Class Hierarchies

Georgia Tech

Step 1: Mapping Regular (Strong) Entity Types

A strong entity type is modeled as a relation schema.

- Each simple attribute of the entity type becomes an attribute of the relation schema
  - For composite attributes, only the simple component attributes are present in the relation schema
- Choose a key of the entity type to be the primary key of the relation schema

A tuple of a relation is an entity instance.

Georgia Tech

Step 2: Mapping Weak Entity Types

Map the attributes of the weak entity type to a relation schema as you wold do for a strong entity type.
Add the primary key attribute(s) of the identifying relationship to the attributes of the weak entity type's relation schema. These attributes should be foreign keys to the identifying relation's primary key.
The primary key of the weak entity types relation schema is a combination of the foreign keys to the identifying relation schema and the weak entity's partial key. If no partial key, then the whole attribute list is the primary key.

Georgia Tech

Step 3: Mapping Binary 1:1 Relationship Types

Three approaches:

1. Foreign keys
2. Merged relation
3. Cross-reference or relationship relation

Favor the foreign key approach

Georgia Tech

Binary 1:1 Relationships - Foreign Key Approach

Include the primary key of one relation schema as a foreign key of the other relation schema, as well as all the simple attributes of the relationship type. If there is a total participation constraint on only one entity type, that entity type's relation schema should be the one with the foreign key (otherwise relation states with the foreign key could have many NULLs).

Georgia Tech

Binary 1:1 Relationships - Merged Relation Schema Approach

If the relationship type has a total participation constraint with both entity types in the relationship, then both entity types and the relationsip type can be merged into a single table.

Georgia Tech

Binary 1:1 Relationships - Cross-reference or Relationship Relation Schema Approach

Set up a third relation schema to represent the relationship, with foreign keys to both participating relation schemas' primary keys. One foreign key becomes the primary key of the relationship schema and the other is a unique key. This approach isn't necessary in 1:1 relationships but is required for M:N relationships.

Georgia Tech

Step 4: Mapping Binary 1:N Relationship Types

Two approaches:

1. Foriegn keys
2. Relationship relation schema

Georgia Tech

Binary 1:N Relationships - Foreign Key Approach

Give the relation schema on the N side of the relationship a foreign key to the primary key of the relation schema on the 1 side of the relationship. The relation schema on the N side of the relationship should also include all the simple attributes of the relationship.

Georgia Tech

Binary 1:N Relationships - Relationship Relation Schema Approach

Create a relationship relation schema whose attributes are the primary keys of the relation schemas representing the participating entity types, which are also foreign keys to the related relation schemas.
This option avoids excessive NULL values if few of the tuples on the N side of the relationship participate in the relationship.

Georgia Tech

Step 5: Mapping Binary M:N Relationship Types

Create a relationship relation schema with foreign keys that are the primary keys of the participating entity types. The combination of these foreign keys is the primary key of the relationship relation. Also include any simple attributes of the relationship.

Georgia Tech

Step 6: Mapping Multivalued Attributes

Create a relation schema for each multivalued attribute which includes the multivalued attribute, A, and a foreign key, K, which is the primary key of the relation schema which represents the entity type from which the multivalued attribute comes. The primary key of the relation schema is the combination of A and K.

Georgia Tech

Step 8: Mapping Superclasses and Subclasses

Two options:

- Multiple relation schemas
  - Subclasses determined by relation schemas
- Single relation schema
  - Subclasses determined by type attributes

Georgia Tech

Step 8A: Multiple relation schemas – all classes

Map the superclass and all subclasses to their own relation schemas. Each relation schema includes all the attributes that are part of their entity type, plus the same primary key, which comes from the superclass. In the subclass relation schemas the primary key is also a foreign key to the superclass's relation schema.
This approach that works for any class hierarchy but will result in single-attribute relation schemas for subclasses with no specialized attributes.

Georgia Tech

Step 8B: Multiple relation schemas – subclasses

Create relation schemas for each subclass with all the attributes of the subclass plus all the attributes of the superclass, and a primary key chosen from the superclass. This approach only works for superclasses that are totally specialized and is only recommended for specializations that are disjoint (overlapping specializations would lead to duplicate entities in subclass relations).

Georgia Tech

Step 8C: Single relation schema – one type attribute

Create a single relation schema with all the attributes of the superclass and all subclasses, plus a type attribute whose value indicates which class each tuple belongs to.
This approach only works for subclasses that are disjoint and may result in many NULL values if subclasses have many attributes.

Georgia Tech

Step 8D: Single relation schema – multiple type attributes

Create a single table with all the attributes of the superclass and all subclasses, plus boolean type attributes for each subclass whose values indicate whether a tuple belongs to that subclass.
This approach works for overlapping subclasses and disjoint subclasses. Like any single-table approach, it may result in many NULL values if subclasses have many attributes.

Georgia Tech

Class Hierarchy Modeling Summary

Use the multiple relation schema approach for classes that have specialized attributes.
Use single table approach for subclasses that don't have specialized attributes, unless subclass participates in a relationship on the N side that its superclass does not.
Can use a combination of multiple- and single-schema mapping approaches.

Georgia Tech