Entity/Relationship Model

- Entity sets (usually noun names)
- Attributes (simple values like numbers and strings)
- Relationships (usually verb names)

In E/R diagrams,

- Entity sets are represented by rectangles.
- Attributes are represented by ovals.
- Relationships are represented by diamonds.

[gearhead, bank(customers, branches, accounts, loans)]

Types of E/R Relationships

- Many-many — —
- Many-one — →
- One-one ← →

Relationships can be binary or n-ary (multiway).

An n-ary relationship can be represented by a connecting entity set and n binary relationships.

Features

- Relationships can have attributes.
- Arcs can have labels called roles. Useful when relationship is between entities in the same entity set.
- The isa relationship is a special relationship. It is represented by a triangle with isa inside; top point connected to superclass, bottom side connected to the subclass.

Design Suggestions

- Model reality as accurately as you can, but
- Keep it as simple as you can (don't have entity sets and relationships you don't really need).
- Avoid redundancy. (Especially, don't let an attribute have a value that is the name of an entity in one of the entity sets. Use a relationship instead of an attribute.) Leave out relationships that can be defined in terms of other relationships.
- Use attributes instead of an entity set if only arrows point to the set, no relationship involves the set more than once, and none of the set's attributes functionally depend on any of its other attributes.

Constraints

- The primary key of an entity set is indicated by underlining the names of the attributes in the key.
- In an isa hierarchy, only the root entity set has the primary key attributes.
- A referential integrity constraint is indicated by using a rounded (half circle arc) arrowhead instead of the normal arrowhead.
- If there is a limit on how many times an entity can appear in a relationship, that can be indicated as a label on the arc connecting to the entity set (e.g., <= 5, = 2). (Called a degree constraint.)

Weak Entity Sets

Weak entity sets are entity sets where entities in the set cannot be uniquely determined without checking which entity in another entity set it is related to by some specific supporting relationship.

Useful for representing sets of entities that belong to some other entity, such as line items in a sales receipt.

Represented by double bordered rectangle, with one or more relationships represented by double bordered diamond, with round arrowhead touching another entity set.

The primary key of a weak entity set includes all the primary keys of the supporting entity sets.

[art gallery]

Converting to Schemas

- Generally, entity sets are converted to relation schemas with the same attributes and primary keys.
- Generally, relationships are converted to relation schemas whose attributes are the primary keys of the entity sets they connect to plus any attributes that the relationships have. Sometimes the primary key attribute names have to be changed to keep them distinct (as when an entity set plays several roles). The combined primary keys are the primary key of the new schemas. (The primary key should not include the primary keys for entity sets pointed to by arrows. If all entity sets connected to the relationship are pointed to by arrows, use the primary key of just one of the entity sets as the primary key for the schema.)

Many-to-One Binary Relationships

It is recommended that the schema for the *many* side of such a relationship be extended with the primary key of the entity set on the *one* side of the relationship plus any attributes that the relationship itself has. Then a schema for the relationship itself is not needed (it has been tacked onto the schema for the *many* side entity set).

The primary key for the extended schema is the same as it was before the extension of the schema.

(Non-binary relationships still have to be handled in the general way.)

Weak Entity Sets

- The primary keys for the supporting sets are added to the attributes of the weak entity set and are included as part of the primary key for the weak entity set. (Change attribute names as needed to avoid duplication.) If the supporting relationships have any attributes of their own, they are added to the attributes of the weak entity set too (but not to the primary key for the weak entity set). The supporting relationships do not need to be converted to schemas.
- The above should be done before any relationship pointing to the weak entity set is converted to a schema so that the full primary key is used in their conversions.

Isa Hierarchies

There are three approaches to converting an isa hierarchy into a set of relation schemas; none is optimal.

- The straight E/R approach
- The object-oriented approach
- The nulls approach

Straight E/R Approach

- Note that each entity set in the isa hierarchy that is not the root has no explicit primary key. Add to the schema for the entity set the primary key of the root entity set as its primary key. The isa link itself is not explicitly converted to a schema. This conversion should be made before the conversion of any other relationships connecting to the entity set so that the proper primary key is used.
- Disadvantage – To answer some queries, all the relations representing entity sets in the hierarchy may have to be examined, not just one relation.
- There are more relation schemas than in the nulls approach.

Object-Oriented Approach

- For each subtree of the hierarchy (formed by starting at the root and possibly pruning one or more branches of the tree), make a relation schema by making the primary key of the root entity set be its primary key and adding all the other attributes attached to any entity set in the subtree. This makes a separate schema for each possible pattern of membership in the entity sets in the isa hierarchy.
- Disadvantages – multiple relations have to be examined to answer some queries, makes the most schemas of the three approaches.

The Nulls Approach

- Make one schema which includes all the attributes of all the entity sets in the isa hierarchy. The primary key of the root entity set is the primary key of the schema. Fill in null as the value of any attribute that doesn't apply to a particular entity when populating the table made from the schema.
- Disadvantages – Lots of space taken up by all those nulls, may have to add extra Boolean attributes to explicitly show which entity sets an entity belongs to if membership cannot be reliably determined implicitly (such as all entities that have a non-null value for a certain attribute, or all entities appearing in a table representing some relationship that was connected to the original entity set).