1 ER-Modeling

The Entity-Relationship Model is a data model like the Relational Model. An ER analysis of a problem results in an ER schema.

2 ER Model Components

The ER Model is based on **entity sets** and **relationship sets**. I will start by looking at entity sets.

2.1 Entity Sets

An **entity** is a tuple of values that represents some object. A possible value is **NULL**, signifying that no value has been set.

An **attribute** represents a property of an entity (or a relationship). Each attribute has a **domain** of permitted values. **Composite attributes** are sets of simple attributes, e.g. name: { first-name, last-name }. In **multivalued attributes**, each value is a set, e.g. phone-numbers = {98211909, 29617479}. **Derived attributes** can be computed from the values of other attribues, e.g. phone-number-count.

Entity sets correspond to classes in OOD. They are defined as a set of attributes, e.g. person = {name, phone-numbers}. Entity sets can be **generalizations** or **specializations** of other sets, similar to class inheritance. Entitiy sets correspond to relations (tables) in the relational model.

2.2 Keys

A **key** is a set of attributes. A **superkey** is a key, such that for all pairs of entities, when the values of the key attributes are equal, the entities are equal. A **candidate key** is a minimal superkey, i.e. no proper subset of the candidate key is a superkey. For each entity set that has one or more candidate keys, one is selected as the **primary key**, which can be used to uniquely identify all entities within the entity set. Entity sets with primary keys are denoted **strong entity sets**.

An entity set is **weak** if it has no **primary key**.

2.3 Relationships

A **relationship** represents an association between two or more entities. A **relationship set** is a relation (a subset of the Cartesian product) between

entity sets, i.e. a set of entity tuples. Relationship sets may also have **attributes**. The entity sets are said to **participate** in the relationship.

Relationship sets are used to model **actions**, such as owning.

Weak entity sets must be part of an **identifying relationship** with an **identifying** or **owner** entity set. All entities in a weak entity set must be associated with an identifying entity. A set of attributes of the weak entity set, the **discriminator**, and the primary key of the owner entity set together uniquely identifies each entity of the weak entity set, e.g. loan = {loan-number, amount, date} - loan-number is the discriminator, which together with a primary key of a customer, identifies the loan.

2.4 Constraints

Relationships can be **one-to-one**, **many-to-one**, or **many-to-many**, thus constraining the participation from the entity sets.

3 Reduction from ER Schema to Tables

As the ER Model and the Relational Model are equivalent, there exists an algorithm for converting from tne former to the latter.

For each **strong entity set**, create a table with one column for each simple attribute, and a column for each component of each composite attribute. Derived and multi-valued are omitted for now.

For each **weak entity set**, create a table like for the strong entity set, and add a column for each attribute of the primary key of the owner entity set. The primary key for such a table are the (foreign) owner primary key attributes and the discriminating attributes of the weak entity set.

For each **one-one relationship set**, extend the table of one of the participating entity sets with the primary key of the other entity sets (foreign primary key). Also add columns for the attributes of the relationship set.

For each **many-one relationship set**, extend the table of the 'many' entity set with participating entity sets with the primary key of the other entity sets (foreign primary key). Add columns for the attributes of the relationship set.

For each **multivalued attribute**, create a new table with 2 columns, one for the value of the attribute and one for the primary key of the entity with the given attribute.