From E-R Models to Relational Databases

E-R Modeling Framework vs. Relational Database Model

Entity-Relationship Models

- **Entities**: objects
- **Attributes**: individual features of objects
- **Keys**: means of unique identification of objects
- **Relationships**: associations between objects
- **Entity Sets**: collections of objects
- **Relationship Sets**: collections of associations
- **Constraints**: restrictions on objects and associations
- **Weak Entity Sets**: entity sets w/o a key

Relational Database Model

- **Relational Table**: 2D representation of data
- **Table row**: data about a single object
- **Table column**: one attribute
- **Table schema**: list of columns making a table
- **Database schema**: list of relational table schemas
- **Primary key**: unique identification of rows in tables
- **Foreign key**: primary key of one table included in another table

From E-R Models to Database Schemas

Basic Translation

**Entity Sets** are translated into relational tables with the same attributes and the same primary keys.

**Note**: This applies only to strong entity sets.

**Relationship Sets** are translated into relational tables which consist of the following attributes:

- **Primary keys** of all tables representing entity sets which participate in the relationship set. Must be declared as foreign keys.
- **Any identifying attributes** of the relationship set.

**Note**: The primary key of the new relational table is application-dependent.

The following is possible:
The primary key of the new table is the union of all foreign key attributes in the table.

- The primary key of the new table is the union of all foreign key attributes in the table plus one or more descriptive attributes.
- The primary key of the new table is a subset of the foreign key attributes.

Representing Different Types of Relationship Sets

The basic translation mechanism works for strong entity sets and many-to-many relationship sets.

Constrained (many-to-one, one-to-one) relationship sets may be translated in several different ways.

Many-to-one relationship sets:

Suppose we have entity sets $E$ and $F$ and a relationship set $R$ which is many-to-one from $F$ to $E$. This can be translated into relational model in one of the following ways.

1. Basic translation. Use basic translation to create relational tables $E$, $F$ and $R$ representing the two entity sets and the relationship set respectively. Do not include attributes from the primary key of $E$ into the primary key of $R$.

2. Special translation. Use basic translation to create relational table $E$ representing the entity set $E$. Create one relational table $F$ to represent both $F$ and $R$ as follows:
   - $F$ contains the following attributes:
     - All attributes of $F$.
     - Primary key of $E$ (declare a foreign key).
     - Any identifying attributes of $R$.
   - Primary key of $F$ is the primary key of $F$.

One-to-one relationship sets:

Suppose we have entity sets $E$ and $F$ and a relationship set $R$ which is one-to-one between $E$ and $F$. This can be translated into relational model in one of the following ways.

1. Basic translation. Use basic translation to create relational tables $E$, $F$ and $R$ representing the two entity sets and the relationship set respectively. Select the primary key of $R$ to be EITHER the primary key of $E$ OR the primary key of $F$.

2. Special translation 1. Treat $R$ as a many-to-one relationship set from $E$ to $F$ and translate using the special translation for many-to-one relationships.

3. Special translation 2. Treat $R$ as a many-to-one relationship set from $F$ to $E$ and translate using the special translation for many-to-one relationships.

4. Special translation 3. Combination of Special translation 1 and Special translation 2. In this case, table $E$ will contain the primary key of $F$ and table $F$ will contain the primary key of $E$.
Representing Weak Entity Sets

A weak entity set comes with a many-to-one identifying relationship. Let $F$ be a weak entity set, $E$ be its identifying owner and $R$ be the identifying relationship. Then, the relational table for $F$ is constructed as follows:

- $F$ contains the following attributes:
  - All attributes of $F$.
  - Primary key of $E$ (declare a foreign key).
  - Any identifying attributes of $R$.
- The primary key of $F$ consists of the primary key of $E$ and the discriminator attributes of $F$.

Note: To create a relational table for a weak entity set, we basically are using the special translation for a many-to-one relationship set, with the only change being the designation of the primary key in the new table.

Representing Aggregation

Let $R$ be a relationship set and $G$ be an entity set, and let $S$ be a relationship set between $G$ and $R$, where $R$ is treated as an aggregate.

To translate $G$ into relational data model:

1. Determine the primary key for $R$.
2. Follow the basic translation or special translation rules outlined above.
   Use the primary key for $R$ in the translation.

Representing Class Hierarchies

Class Hierarchies can be converted into relational database schemas in three different ways: E-R-syle, Object-Oriented-style and via a universal table.

E-R-style Conversion

- Each entity set in the class hierarchy is represented as a separate relational table.
- The attributes for each table are: the attributes of the respective entity set and the primary key attributes from the root entity set.
- All relational tables have the same primary key.

Object-Oriented Conversion

- Each terminal path in the class hierarchy is represented as a separate relational table.
- The attributes for each table are the union of attributes in all entity sets located on the terminal path represented by the table.
- The primary key is the primary key of the root entity set.
Universal Table Conversion

- All entity sets are represented as a single universal table
- The attributes of the universal table are the union of all attributes in all entity sets.
- The primary key is the primary key of the root entity set.