

HOW TO TRANSFORM AN ENTITY-RELATIONSHIP MODEL INTO A RELATIONAL MODEL?

Two very important models used in database design are the Entity-Relationship (E-R) Model and the Relational Model. Since I have already read two articles about relational model and entity-relationship model, I really wonder to the correlations and transformation between them.

In the E-R model, data is represented using entities, and relationships are defined between these entities. However, with the relational model, the entities and their relationships follow strict guidelines. Usually, an E-R model is first developed, and then it is transformed into a relational model.

The relational model is similar to the E-R model but based on a branch of mathematics called relational algebra, and as such, there are strict definitions and rules regarding the elements of this model. Here are some of these definitions and rules:

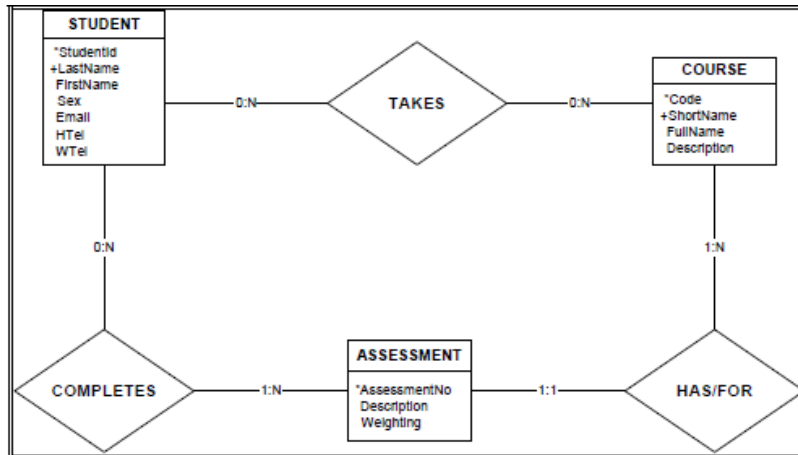
- A relation is defined as a two dimensional table that contains rows and columns. Rows are instances of the entity and columns are attributes of the entity.

- No two rows of the table must be identical – each row must be unique (the contents of the row that is), and the order of the rows is not important.
- Each column must have a unique name, and the order of the columns is not important. They must also contain single values – multiple values and repeating groups are not allowed.
- A key is a group of one or more attributes that uniquely identifies a row – each table must have a key. Since each row in the table is unique, the key is what is used to distinguish one row from another. It can be comprised of a single attribute or it can be composite, comprising of more than one attribute.
- Every column (attribute) in a table must depend solely on the primary key. It should not depend on any other column in the table, or on part of the primary key if it is a composite key.

Now let me show you an example to see how to transform a E-R model to relational model:

The entity-relationship model:

An E-R diagram is used to represent the E-R model. It contains all known entities for the given scenario, their attributes, identifiers, and the relationships that exist among the entities.

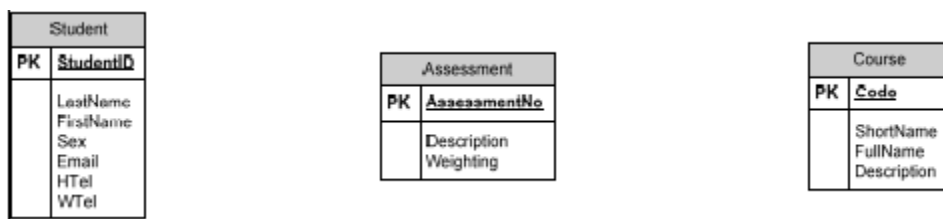


² Unique identifiers are indicated with an asterisk (*), and non-unique identifiers with a plus (+).

Transformation:

Step1:

Clearly identify the primary key and attributes for each entity defined in the E-R model, and ensure that it is in accordance with the rules of the relational model as previously discussed.

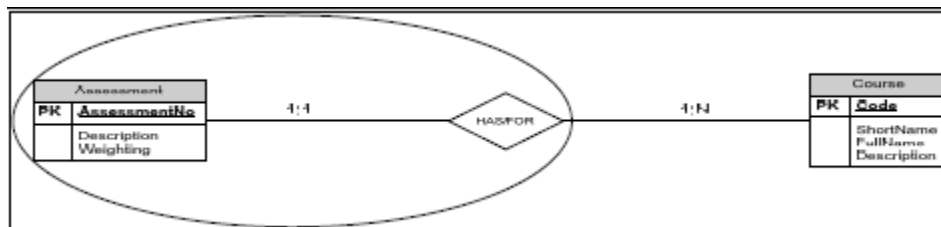


² Each entity, with its clearly identified primary key (indicated by PK), and attributes satisfying the previously discussed rules, becomes a table in the relational model.

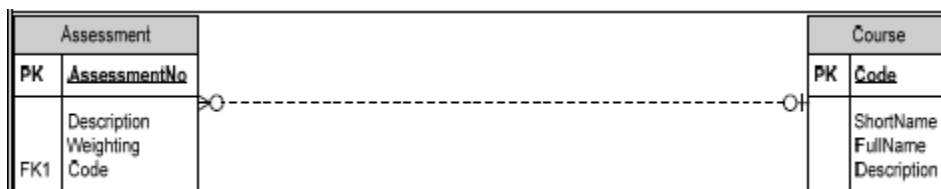
Step 2:

Group together tables (formerly entities) and their relationships that have a cardinality of one relationships with 0:1 or 1:1 for their opt:card. (Optionality says what can and must happen in a relationship, and cardinality indicates the number of entity occurrences in a relationship) That is, absorb relationships where the cardinality is one into the corresponding tables.

- Maintain the initial structure for the absorbing table – do not change its primary key or any of its attributes.



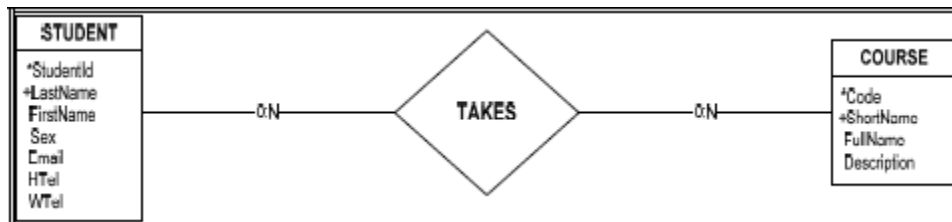
- The primary key of the other table in the relationship becomes a foreign key in the absorbing table. This is indicated by an FK in the absorbing table.
- If the cardinality between the absorbed relationship and the other table in the relationship was N (0:N or 1:N), then the modified table becomes the many part of a one-to-many relationship in the new relational model.



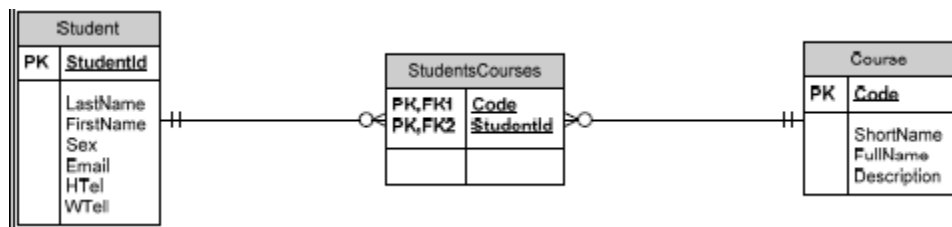
- Otherwise, if the cardinality between the absorbed relationship and the other table in the relationship was 1 (0:1 or 1:1), then the new relationship becomes a one-to-one relationship in the new relational model (not shown).

Step 3:

The remaining relationships whose cardinalities are N (1:N or 0:N) on both sides become new tables in the new relational model.

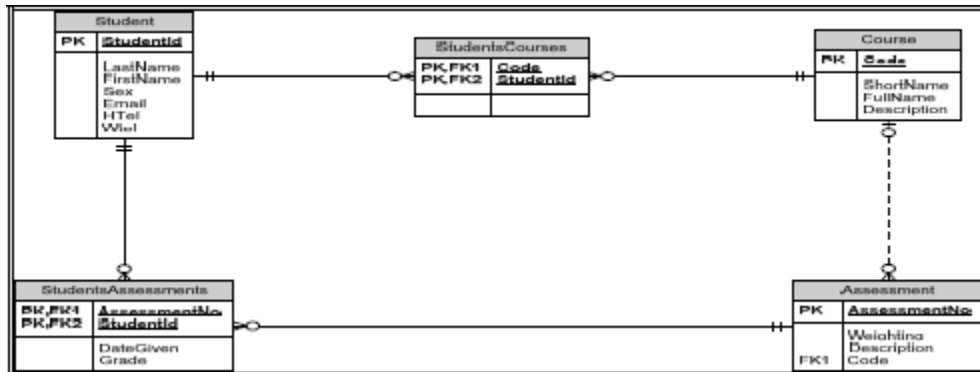


- The primary keys from the two tables involved in the relationship become a composite primary key in the new table, and the new table usually has a name that is a combined form of the two original table names.



- The newly created table becomes the many part of the relationship between both tables, and thus creates a many-to-many relationship between the two pre-existing tables. In some instances, the newly created may have its own attributes, but this is rare.

The resulting diagram is given below.



Conclusion:

While an E-R diagram displays the logical nature of the data that exists in the user's domain, the relational model shows how this data will be represented in a Relational Database Management System (RDBMS). This model is important because the most common and popular database management systems in use today are relational, and relational model diagrams can be directly implemented on any RDBMS.

Source: <http://toyhouse.cc/profiles/blogs/how-to-transform-an-entity-relationship-model-into-a-relational>