Mapping Approach

The mapping approach for our database is fairly simple. We have tried to keep our SQL code concise by minimizing the use of cross-reference table i.e. only for many to many relationship. All other forms of relationship has been added to an existing entity table.

Relation between entities:

Part_of:

Variable to Equation is a one to many relation (1:n) as a variable may be a part of more than one equation

Equation to Variable is also a one to many relation (1:n) since an equation contains more than one variable.

Calculated_by:

Even though more than one physicist may be associated with a constant/equation, our database is modeled to show the physicist with major contribution for the discovery.
So, constant has a one to one relation (1:1) with Physicist.

Since our database consists of physicists contribution to equations and constants, there might be a case where a physicist, in our database, has only discovered an equation but not a constant. Hence, physicist entity is related to constant by a 0:n relation.

Given_by:

Similar to Calculated_by relation between Constant and Physicist, Equation to Physicist has a 1:1 and Physicist to Equation has a 0:n relation.

Appears_in:

Since every equation is association with a field, Equation and Field are related by 1:1 relationship.

A field has numerous equations that appear in it. So, Field and Equation has a 1:n relation.

IS_A Hierarchies:

Quantity, Variable and Constant

The entity Variable is a child class of Quantity and, further, the entity Constant of Variable. The child entities have their own primary key even though they inherit a primary key from parent class. This is because it makes relating Variable and Constant with other entities easier and more logical in the actual code. The attributes Vid and Cid then can be used as a reference and form a one to something relation with other entities.

We could have treated the entire set as a one big relation but it doesn't make sense to have a quantity related with equations and physicist as quantity is rather a very general component of our database. Also, it could be said that our project is entirely based upon Variable and Constant.

Equation, Differential Equation and Polynomial Equation

An equation can be divided into differential and polynomial equation. This property makes using ISA relationship relevant. We treat the entire set as a one big relation because the differential and polynomial entities are just an additional property of the equation that we want to show the user. So, the entity sets with instances do not have any relations.