Different Types of Tables in Relational Databases: an Ontological Perspective. Part 1.

Tom Johnston
September, 2005

First a word about types. Types depend on a context. For example, from the point of view of relationship dependencies among relational tables, all tables belong to one of only *three* types, those being:

- **Kernel tables**, ones whose unique identifier does not contain any foreign keys. Policy Holder and Invoice for example. Rows of kernel tables do not depend on any relationships in order to distinguish themselves one from the other.

- **Dependent tables**, ones whose unique identifier contains a foreign key back to a kernel table, together with a unique identifier for each instance of the dependent. A reference to a row in another table is a necessary component of the set of attributes which distinguish these rows from one another. Covered Family Member (of Policy Holder) and Line Item (of Invoice) for example.

- **Associative tables**, ones whose primary key contains two or more foreign keys, and no non-foreign key components. References to rows in the tables related by the associative table are sufficient to distinguish rows of the associative table from one another.[1] Customer / Salesperson and Invoice / Payment for example.

The context in which kernel, dependent and associative tables are types has to do with relationships among tables, and whether they are necessary and sufficient (associative tables), necessary but not sufficient (dependent tables), or neither necessary nor sufficient (kernel tables).[2]

So within what context do I distinguish different types of tables? That context is an *ontology*. "Ontology", of course, is a *big* word, one that I will discuss in depth elsewhere. For now, think of it as a set of basic categories of what there is. The formal study of ontology goes back to Aristotle, in particular to his works *Categories* and *de Interpretatione*. The set of basic categories which my work in information technology has led me to formulate are the subject of the rest of this essay.

---

[1] The definitions would have to be extended a little to be truly exhaustive of all cases. What about, for example, a table with three foreign keys, or perhaps two foreign keys plus a date, as the primary key?

[2] The fourth case – that in which relationships are sufficient but not necessary – is ruled out because, for reasons I will not go into here, every component of a relational table's unique identifier must be necessary.

Things.

Most types have instances.[3] Human being is a type, and you and I are instances. MasterCard cardholder is a type, and I and a few million other people are instances. Invoice – for the Acme Company – is a type, and every invoice Acme has issued is an instance.[4,5]

Secondly, there are two types of types of things. One type is the type of thing whose instances are individuals, i.e. whose instances exist in their own right. Human beings, cardholders and invoices are all types of this type.

A second type is a type of thing whose instances do not exist in their own right. Instances of types of this type exist as properties of things or relationships among things. Color is an example of a type whose instances are properties of individuals. Being taller than is an example of a type whose instances are relationships among individuals.

Grammar is a good guide to this distinction. If you can use "the", "a", "some" or "all" in front of the term, you are referring to individuals; otherwise not. For example, we can say "A customer who ….. ", or "Some customers …..", but not "A blue which ….." or "Some blues ..…".

Traditionally, types of the first type have been called "particulars", while types of the second type have been called "universals".[6] I have already said that I will refer to

---

[3] The ones that don't are null sets, sets that have no members. Examples include the set of odd numbers divisible by two, the set of all men who do and do not shave themselves, and the set of all men born after 1900 who were or are over twenty feet tall.

[4] I have used the fictitious Acme Company as an example in many of my articles, and will do so here also.

[5] Logicians often refer to this as the distinction between types and "tokens", but I think the word "instance" is a better one to use. I would note also that, in object theory, classes are types and objects are their instances. In set theory, sets are types, and the members of those sets are their instances.

[6] The claim, made here, that particulars exist in their own right, while universals have only derived existence as properties or relationships of particulars, is both ancient and controversial. Those who make it are called nominalists; those who deny it are called realists. For the purposes of this article, however, it doesn't matter which position you take. It is worthy of note, however, that the realism vs. nominalism debate has an exact parallel in recent discussions of types of object-oriented systems. Those systems which treat classes "as first-class objects" are precisely realist, in the traditional philosophical sense. Those which do not are precisely nominalist. If you think that the difference between these two kinds of object-oriented systems is important, then you think that this ancient philosophical debate is important, too.

instances of the first type of thing as “individuals”. I will refer to instances of the second type of thing as “concepts”.[7]

Particulars.

Particulars are types whose instances exist in their own right. “Existence in its own right” is a concept that philosophers have discussed for centuries – indeed, for millenia. But the use to which I will put the term does not, I believe, require any reference to those discussions. Let’s just note that what exists in its own right is what does not exist merely as a feature of other things. You and I are human beings, and not merely properties of other things. By contrast, color is not a particular, nor are red, blue and green individuals. Some individuals which are material objects *have* color, such as cars and flowers. So instances of color exist, not in their own right, but as properties of things. Instances of particulars exist in their own right. Instances of universals exist only derivatively, as instances of the concepts which are their subtypes.

Another thing to notice about individuals is that because they exist through time, they are subject to change. So particulars are things whose instances can change. These changes we understand and express as changes in either their properties or their relationships. Possibly, coming into existence and dropping out of existence are additional properties (which we usually record as start dates and end dates in database tables) – although philosophers have also discussed whether or not existence (let alone existence in its own right) is a property.

Things, Once Again.

Another point about individuals, already alluded to, is that individuals have properties and relationships. For example, physical objects have mass and density; some of them have color. Non-physical objects have their own properties and relationships. Prime numbers, for example, have the property that they are divisible only by themselves and 1.

However, this does not distinguish individuals from concepts, for concepts may also have properties and relationships. For example, the concept of an even number has a relationship to the concept of number, namely that the former is a subtype of the latter or, equivalently, that all instances of the former are instances of the latter.

---

[7] This is not entirely standard usage among philosophers. Concepts are usually thought of as mental entities, but all I mean by the term is “instance of a universal”. More precisely, I mean “a universal which is a subtype of another universal, and whose instances are properties of or relationships among, individuals”. In object theory, the first kind of universal (for which I reserve the term “universal”), is a non-instantiable class, while the second kind of universal (which I call a “concept”) is an instantiable class.

Another important point about both particulars and universals is related to Aristotle's distinction between essential properties and accidental properties, and to Leibniz's principle of the identity of indiscernables. As to Aristotle's distinction, essential properties are those which, if they change, make the individual a different individual. As to Leibniz's principle, if two things cannot be distinguished by any of their properties or relationships, they are not two things, but one and the same thing.[8]

In a table which is a relation, i.e. a set, no two rows can be identical. A relational DBMS will not allow the insertion of two rows that have the same values in each of their columns. But usually, relational tables have primary keys. A primary key is a set of one or more relationships (implemented as foreign keys) and/or properties (non-foreign keys) that distinguish every row in the table from every other row. This key usually consists of fewer than all the columns of the table in question.A relational DBMS will also not allow the insert of two rows that have the same values in their primary key columns.

In the first case, we have the relational DBMS implementation of Leibniz's principle. In the second case, primary key columns are Aristotelian essential properties, while the other columns are accidental properties. Values in the non-primary key columns can change without forcing us to delete the row and insert a new row. Values of Aristotelian accidental properties can change without making the instance a different instance. As for primary keys, values in primary key columns cannot change. Looked at another way, if they do change, that means we have a different row. Aristotelian essential properties cannot change or, if they do, that means we have a different individual.

A Terminology of Things.

So we have the following terminology.

- **Things** are what there is.
- **Particulars** are types of things that exist in their own right.
- **Individuals** are instances of particulars.
- **Universals** are types of things whose instances exist as properties or relationships of things (either particulars, or other universals).
- **Concepts** are instances of universals.
- (And so it follows that) **Properties and relationships** are instances of concepts.

[8] These and other allusions in this essay to philosophical discussions of ontology make a short essay on the history of ontology desirable. I will provide that essay later on.

Events.

If things are what *exist*, events are what *happens*. Like things, events have both types and instances. But unlike things, we don't have convenient one-word terms to designate event types and event instances.

An event type is distinguished from all other event types by any or all of the following:

- the particulars that take part in the event;
- the properties and relationships of those individuals that are affected by the event; and
- properties and relationships of the event itself.

For example, the receipt of a shipment is an event. Particulars involved in the shipment might include the supplier, the carrier, and the department and company receiving the shipment.[9] are One relevant property of the shipment is the items it includes, along with their product types and quantities. A relevant relationship of the shipment might be a relationship to the one or more purchase orders which authorized it. One property of the event itself includes when it took place.[10]

So we have the following terminology.

- **Events** are what happens.
- **Types of event** are types of what happens.
- **Instances of events** (which the unqualified term "event" will more commonly be used to refer to) are points in time, or delimited stretches of time, at which an event of a designated type occurs.

In Part 2, we will see how these ontological categories are reflected in different types of database tables.

---

[9] "Might include", rather than "does include", because the particulars and universals involved in events depends on what the company interested in those event wants to keep track of.

[10] "Includes", rather than "may include", because although what is included or not is always a matter of what we are interested in, we almost never fail to be interested in when an event occurred.