Subject: Re: Tutorial
Posted by DannyBoyPoker on Sun, 21 Apr 2013 07:57:17 GMT
View Forum Message <> Reply to Message

AJM wrote on Sat, 20 April 2013 06:33You are just nitpicking over trivialities.
Some of putative anomalies that I listed were pretty trivial.

AJM 4) I dislike case sensitive software immensely, which is why I wrote Case Sensitive Software is Evil. Consequently I consider "TEST" to be the same as "test".

The status quo is that some things are case sensitive. In that sense, I don't have any problems with casing. I mean, I might sympathize--case sensitivity, shall we say, has got to go. But, the world happens to be case sensitive, and this just happens to be one of the many ways that programmers can generate run-time errors. Requiring exact case matching is much easier to support. One might range widely, in considering this as a practical issue--for example, case insensitive comparisons are definitely much harder to do when you are dealing with more than just the standard English character set. Casing is an arbitrary distinction, but we are dealing with machines.

AJM5) It should not be difficult to understand that the "subsystem with the id of 'test'" is exactly the same as the "subsystem with the description 'test subsystem'".

Well, my response is going to resemble my previous response--what is easy to understand, is a rather subjective category, even if we were not also dealing with machines.

AJM6) The fact that some of the navigation buttons on the screens has been updated since the screenshots were made does not invalidate the functionality which is being described.

This is good to know. And also I'm sorry if I came off abrasive, chalk it up to enthusiasm.

AJM7) You do not understand the difference between a "foreign key" and a "foreign key constraint". A one-to-many relationship between two tables requires that the "many" table contains a foreign key which links to the primary key on the "one" table. It is possible to define a foreign key without any constraints, which is exactly what I do. A great deal of my database development has been with databases that did not have foreign key constraints (such as the MyISAM engine in MySQL)y I don't use them but instead have the relevant functionality built into my framework.

I do not understand, that it is possible to define a foreign key without any constraints? I understand that MySQL tables do not enforce referential integrity, because some MySQL engines (MyISAM, NDB) do not enforce referential integrity. I guess that I might attempt a list of some of the advantages of foreign key enforcement, but on the other hand, you have the relevant functionality built into your framework. So, you are leveraging some types of constraints, and the key advantage of this, is to increase the integrity of the data.