Subject: Recommendations when setting up databases
Posted by David Lee on Sat, 13 Jan 2007 07:51:37 GMT
View Forum Message <> Reply to Message

There are various features in Radicore which are most easily used if the database structure is set up in a particular manner. I am not aware of any single document that collects any such recommendations together, and suspect that Tony, as the author of Radicore, is so used to always setting up databases to use all the Radicore features, that it is difficult for him to list them.

Others, from different backgrounds, may become more aware of them, so hopefully this thread will be a place to record them, and so reduce the occurrences of changing the database structure during development of the user interface!

I am sure Tony will correct any errors, and may find a better place for this list, especially if it grows.

Enough introduction, here is my initial list:

On Data Types:

A.1) Boolean fields. These are stored as single characters, but declared as boolean in the dictionary

A.2) Enum data types. Allowed, but only possible with MySQL. See FAQ64. A drop-down list using a numeric index field may be a better universal solution.

A.3) MySQL SET data type. supported, and mapped to the PostgreSQL ARRAY data type. See FAQ59

On field names:

B.1) start_date and end_date fields. If both of these are present in a database, an extra field, curr_or_hist, with values of "current","historic", or "future" is available. See FAQ53

B.2) pop-up index fields. The index field is displayed by default on some standard screens. This should therefore, be a string which is a short-form representation of the selected option.

B.3) Preferably all field names should be unique within a database, except for related primary and foreign keys. This needs further comment.

On Indexes and relationships:

C.1) Foreign index fields. The examples generally use the same same for the primary key of a parent, and the related foreign key in the child table. Although this is not essential, following this convention makes it easier to copy the examples.

C.2) Primary index. Using a field name such as "ID" for all tables is likely to cause problems.

The FAQ noted in the above can be found at
http://www.tonymarston.net/php-mysql/infrastructure-faq.html

David Lee
