

Forum:[Hack Forums](#)

Thread: [Best way to store lists in MySQL database?](#)

SIM-964740 - July 13, 2020, 11:10 pm

I am theorising a project and trying to find an efficient way to store lists which may have variable headers.

For example List 1 may have the headers of [Name, Phone Number, Address, Postcode] and List 2 [First Name, Last Name, Job Title, Dogs Name, Postcode]

Each list header would warrant a different data type and number of columns.

Is it acceptable and scalable to create multiple tables for each list on an application?

saltycracker - July 14, 2020, 12:08 am

List 'generated from a program' don't necessarily translate well to database tables. You should design the database first and then make the app's list or data fit the database.

I'm old school, so newer database models may accommodate a more flexible arrangement but you did mention MySQL specifically.

SIM-964740 - July 14, 2020, 12:26 am

(Jul 14, 2020 12:08 AM)saltycracker Wrote: [\[CITE\]](#) List 'generated from a program' don't necessarily translate well to database tables. You should design the database first and then make the app's list or data fit the database.

I'm old school, so newer database models may accommodate a more flexible arrangement but you did mention MySQL specifically.

I'm more thinking, for example, a CRM where a user uploads their own data. So each account on the application can upload any list. If we even forget about data types and columns, would a new table be necessary for each list?

saltycracker - July 14, 2020, 12:58 am

Did you look at this data type?

<https://www.mysqltutorial.org/mysql-json/>

SIM-964740 - July 14, 2020, 2:06 am

(Jul 14, 2020 12:58 AM)saltycracker Wrote: [\[CITE\]](#) Did you look at this data type?

<https://www.mysqltutorial.org/mysql-json/>

That's amazing and a never knew about it thank you.

Are you suggesting that it's okay to store lists in a single row with a json datatype column (4gb capacity presuming is based on LongText) or is this for the properties in each contact you are referencing?

Maxie - July 14, 2020, 2:14 am

If I were you I would look into NoSQL databases for this type of data. It's not at all beneficial to store

all of the data in blobs and if you want to do so, you might as well serialize the list object into the database.

A NoSQL database will also allow you to query the data, which would be hard with MySQL even with the JSON-storage.

Another way to do it, which I've done personally with a MySQL database is to store "value, label, id corresponding to the data, id of the entry", which will allow you to store lists. The downside with this is that it's harder to store different datatypes. I suppose you could store each entry as a varchar and have another column which would be the type. It would still be a pain to query that database though.

pointandlaugh - July 20, 2020, 4:52 pm

I'd make a schema then beat my data into it with a translation app. Your alternative is to use something like the [JSON type](#) which gives you a queryable column that uses JSON data.

The biggest questions you'll have to answer are: Are these different lists distinct objects or just different ways to represent the same data?

Were you to translate it into a generic schema, would you be losing anything important?

How complex would it be to translate the objects into your schema?

I'd avoid being seduced by the NoSQL beast, things like Mongo or Dynamo have broader implications on your application than it first appears and I've only really found much use in using them for deduplication myself.