One database or many?

Asked 16 years, 4 months ago Modified 14 years, 8 months ago

Viewed 2k times

▲

9

▼

I am developing a website that will manage data for multiple entities. No data is shared between entities, but they may be owned by the same customer. A customer may want to manage all their entities from a single "dashboard". So should I have one database for everything, or keep the data seperated into individual databases? Is there a best-practice? What are the positives/negatives for having a:

- database for the entire site (entity has a "customerID", data has "entityID")
- database for each customer (data has "entityID")
- database for each entity (relation of database to customer is outside of database)

Multiple databases seems like it would have better performance (fewer rows and joins) but may eventually become a maintenance nightmare.

performance database-design multi-tenant maintainability

Share

Improve this question

edited Sep 3, 2009 at 11:10

Follow

hlovdal
28k ● 11 ● 100 ● 175

asked Aug 21, 2008 at 15:53

dsims
1,322 ● 1 ● 15 ● 24

11 Answers

Sorted by: Highest score (default)

5

Personally, I prefer separate databases, specifically a database for each entity. I like this approach for the following reasons:

1. Smaller = faster regarding the queries.
2. Queries are simpler.
3. No risk of ever accidentally displaying one customer's data to another.
4. One database could pose a performance bottleneck as it gets large (# of entities increase). You get a sort of build in horizontal scalability with 1 per entity.
5. Easy data clean up as customers or entities are removed.

Sure it'll take more time to upgrade the schema, but in my experience modifications are fairly uncommon once you deploy and additions are trivial.

Share Improve this answer Follow

answered Aug 28, 2008 at 15:26

Vinnie
12.7k ● 15 ● 61 ● 80

2

I think this is hard to answer without more information.

I lean on the side of one database. Properly coded business objects should prevent you from forgetting clientId in your queries.

The type of database you are using and how it scales might help you make your decision.

For schema changes down the road, it seems one database would be easier from a maintenance perspective - you have one place to make them.

Share Improve this answer

Follow

answered Aug 21, 2008 at 17:47

JasonS

23.9k ● 9 ● 42 ● 47

1

What about backup and restore? Could you experience a customer wanting to restore a backup for one of their entities?

Share Improve this answer

Follow

answered Aug 21, 2008 at 15:54

Lasse V. Karlsen

391k ● 106 ● 646 ● 844

1

This is a fairly normal scenario in multi-tenant SAAS applications. Both approaches have their pros and cons. Search on best practices for multi-tenant SAAS (software as a service) and you will find tons of stuff to ponder upon.

Share Improve this answer

Follow

answered Aug 21, 2008 at 15:56

Vaibhav

11.4k ● 11 ● 53 ● 71

1

Check out this article on Microsoft's site. I think it does a nice job of laying out the different costs and benefits associated with Multi-Tenant designs. Also look at the Multi tenancy article on wikipedeia. There are many trade offs and your best match greatly depends on what type of product you are developing.

Share Improve this answer Follow

answered Feb 2, 2009 at 17:50

Aaron Fischer
21.2k ● 18 ● 78 ● 117

0

One good argument for keeping them in separate databases is that its easier to scale (you can simply have multiple installations of the server, with the client databases distributed across the servers).

Another argument is that once you are logged in, you don't need to add an extra where check (for client ID) in each of your queries.

So, a master DB backed by multiple DBs for each client may be a better approach,

Share Improve this answer Follow

answered Aug 21, 2008 at 15:58

Vaibhav
11.4k ● 11 ● 53 ● 71

If the client would ever need to restore only a single entity from a backup and leave the others in their current state,

0

then the maintenance will be much easier if each entity is in a separate database. if they can be backed up and restored together, then it may be easier to maintain the entities as a single database.

Share Improve this answer Follow

answered Aug 21, 2008 at 16:00

Chris Miller
4,899 ● 4 ● 34 ● 50

0

I think you have to go with the most realistic scenario and not necessarily what a customer "may" want to do in the future. If you are going to market that feature (i.e. seeing all your entities in one dashboard), then you have to either find a solution (maybe have the dashboard pull from multiple databases) or use a single database for the whole app.

IMHO, having the data for multiple clients in the same database just seems like a bad idea to me. You'll have to remember to always filter your queries by clientID.

Share Improve this answer Follow

answered Aug 21, 2008 at 16:40

Tundey
2,965 ● 1 ● 24 ● 28

0

It also depends on your RDBMS e.g.

With SQL server databases are cheep

With Oracle it is easy to partition tables by customer "customerID", so a single large database can run as fast as a small database for each customer.

However witch every you choose, try to hide it as a low level in your data access code

Share Improve this answer Follow

answered Sep 3, 2009 at 12:02

Ian Ringrose

51.9k ● 57 ● 217 ● 321

0

Do you plan to have your code deployed to multiple environments?

If so, then try to keep it within one database and have all table references prefixed with a namespace from a configuration file.

Share Improve this answer Follow

answered Sep 20, 2009 at 12:33

MathGladiator

1,211 ● 1 ● 10 ● 24

▲

0

▼

The single database option would make the maintenance much easier.

Share Improve this answer

Follow

answered Apr 26, 2010 at 15:42

iDevlop

25.3k ● 12 ● 94 ● 157