Which database table Schema is more efficient?

Asked 16 years, 3 months ago Modified 16 years, 3 months ago

Viewed 3k times

4

Which Database table Schema is more efficient and why?

```
"Users (UserID, UserName, CompamyId)"
"Companies (CompamyId, CompanyName)"
```

OR

```
"Users (UserID, UserName)"
"Companies (CompamyId, CompanyName)"
"UserCompanies (UserID, CompamyId)"
```

Given the fact that user and company have one-to-one relation.

database-design

Share

Improve this question

Follow

asked Sep 2, 2008 at 4:01

Ramesh Soni

16.1k ● 28 ● 96 ● 117

3 Really, one to one? Every company has only one user and every user belongs to only one company? Then you don't need a companies table at all, it's just an attribute of user. Or vice versa. – Stephanie Page May 18, 2010 at 22:35

6 Answers

Sorted by: Highest score (default)

7

well that's a bit of an open ended question and depends on your business rules. The first option you have only allows one company to be mapped to one user. you're defining a many-to-one relationship.

The second schema defines a many-to-many relationship which allows multiple users to be mapped to multiple companies.

They solve different problems and depending on what you're trying to solve will determine what schema you should use.

Strictly speaking from a "transactions" point of view, the first schema will be quicker because you only have to commit one row for a user object to be associated to a company and to retrieve the company that your user works for requires only one join, however the second solution will scale better if your business requirements change and require you to have multiple companies assigend to a user.

Share Improve this answer edited Sep 2, 2008 at 4:11

Follow

answered Sep 2, 2008 at 4:06

lomaxx

116k ● 58 ● 147 ● 180

6

For sure, the earlier one is more efficient given that constraint. For getting the same information, you will have less number of joins in your queries.

Share Improve this answer

Follow

answered Sep 2, 2008 at 4:05

Vaibhav

11.4k ● 11 ● 53 ● 71

1

As always it depends. I would personally go with answer number one since it would have less joins and would be easier to maintain. Less joins should mean that it requires less table and index scans.

```
SELECT userid, username, companyid, companyname
FROM companies c, users u
WHERE userid = companyid
```

Is much better than...

```
SELECT userid, username, companyid, companyname
FROM companies c, users u, usercompanies uc
```

```
WHERE u.userid = uc.userid
AND c.companyid = uc.companyid
```

Share Improve this answer

Follow

answered Sep 2, 2008 at 4:06

Kevin Lamb

717 ● 1 ● 9 ● 19

1

The two schemas cannot be compared, as they have different relationships, you should proablly look at what the spec is for the tables and then work out which one fits the relationship needed.

The first one implies that a User can only be a member of *one* company (a belongs_to relationship). Whereas the second schema implies that a User can be a member of many companies (a has_many relationship)

If you are looking for a schema that can (or will later) support a has_many relationship then you want to go with the second one. For the reason compare:

```
//select all users in company x with schema 1
select username, companyname from companies
inner join users on users.companyid =
companies.companyid
where companies.companyid = __some_id__;
```

and

```
//select all users in company x with schema 2
select username, companyname from companies
inner join usercompanies on
usercompanies.companyid = companies.companyid
```

```
inner join users on usercompanies.userid =
users.userid
where companies.companyid = __some_id__;
```

You have an extra join on the select table. If you only want the belongs_to relationship then the second query does more work than it should - and so makes it less efficient.

Share Improve this answer

Follow

answered Sep 2, 2008 at 4:09

roo

7,196 ● 9 ● 42 ● 45

0

I think you mean "many to one" when it comes to users and companies - unless you plan on having a unique company for each user.

To answer your question, go with the first approach. One less table to store reduces space and will make your queries use less JOIN commands. Also, and more importantly, it correctly matches your desired input. The database schema should describe the format for all valid data - if it fits the format it should be considered valid. Since a user can only have one company it's possible to have incorrect data in your database if you use the second schema.

Share Improve this answer

Follow

answered Sep 2, 2008 at 4:08

Kyle Cronin

79k ● 45 ● 151 ● 167

0

If User and Company really have a **one-to-one** relationship, then you only need one table:

```
(ID, UserName, CompanyName)
```

But I suspect you really meant that there is a *one-to-many* relationship between user and company - one or more users pr company but only one company pr user. In that case the two-table solution is correct.

If there is a *many-to-many* relationship (a company can have several users and a user can be attached to several companies), then the three-table solution is correct.

Note that *efficiency* is not really the issue here. Its the nature of the data that dictates which solution you should use.

Share Improve this answer

Follow

answered Sep 17, 2008 at 13:44

JacquesB
42.6k ● 13 ● 75 ● 88