Table Naming Dilemma: Singular vs. Plural Names [closed]

Asked 10 years, 8 months ago   Active 1 year, 2 months ago   Viewed 363k times

▲

1355

▼

★

458

Academia has it that table names should be the singular of the entity that they store attributes of.

I dislike any T-SQL that requires square brackets around names, but I have renamed a `Users` table to the singular, forever sentencing those using the table to sometimes have to use brackets.

My gut feel is that it is more correct to stay with the singular, but my gut feel is also that brackets indicate undesirables like column names with spaces in them etc.

Should I stay, or should I go?

| sql | | sql-server | naming-conventions |

edited Nov 5 '17 at 4:11

community wiki
7 revs, 2 users 63%
ProfK

---

closed as primarily opinion-based by Adi Lester, Ry-♦ Oct 9 '13 at 20:24

Many good questions generate some degree of opinion based on expert experience, but answers to this question will tend to be almost entirely based on opinions, rather than facts, references, or specific expertise.

If this question can be reworded to fit the rules in the help center, please edit the question.

Dup of the earlier Database tables naming, plural or singular, but this one got more attention. – outis Mar 2 '12 at 1:37 ✎

---

7   I'm surprised that more people aren't saying: it's determined by what single row represents. In a single database I may have a table who's rows represent single a single widget, and another who's one-to-many relationship to that table means rows represent many widgets. Not doing this looses expressiveness. – andygavin Sep 21 '15 at 22:31

---

1   I just want to add that in all of these discussions, please note that a table in no way shape or form the same as a class. A table is a collection of

by far the highest ranking google answer on this subject, so although the question is closed, the page still has value. – user2654834 Jan 27 '17 at 15:49 ✎

1   I would go for the common practice of the ecosystem you are working in. For example: In Node.js ORMs like Bookshelf.js or Objection.js are mainly based on "Knex.js". And in the "Knex.js" documentation you will find table names in plural. So I would go for plural in that domain. Source: knexjs.org/#Schema-createTable – Benny Neugebauer Sep 7 '17 at 20:58

2   Yes I agree. It makes sense to have a table of users and call it "AppUser" at the same time it also makes sense to have a table of rules applicable to a particular type of user and call it "UserRules" – Arindam Dec 26 '18 at 6:02

41 Answers

1   2   next

Others have given pretty good answers as far as "standards" go, but I just wanted to add this... Is it possible that "User" (or "Users") is not actually a full description of the data held in the table? Not that you should get too crazy with table names and specificity, but perhaps something like "Widget_Users" (where "Widget" is the name of your application or website) would be more appropriate.

233

answered Dec 3 '08 at 19:31   community wiki
Tom H

✓

7   I agree. OrgUsers, AppUsers, anything to avoid using a keyword. – MikeW Mar 12 '09 at 23:11

5   -1. Table Users (and Countries, Languages) can be used in few applications simultaneously. – OZ_ Sep 27 '11 at 10:19 ✎

116   Wouldn't associating schema name would remove all the confusion? AppName1.Users, AppName2.Users ? – Zo Has Oct 25 '11 at 9:49 ✎

5   That's one possibility, but there could be many reasons why someone wouldn't want to use schemata that way. Also, even with a schema you will still run into problems with them being keywords. – Tom H Oct 25 '11 at 13:57

6   I disagree with the table prefix -- that should perhaps be a schema? -- but I agree with a "more descriptive name". Even something as simple as "AppUser" would be sufficient, without entering the entire namespace debate. – user166390 Feb 10 '12 at 16:21 ✎

I had same question, and after reading all answers here I definitely stay with SINGULAR, reasons:

Reason 1 (Concept). You can think of bag containing apples like "AppleBag", it doesn't matter if contains 0, 1 or a million apples, it is always the same bag. Tables are just that, containers, the table name must describe what it contains, not how much data it contains. Additionally, the plural concept is more about a spoken language one (actually to determine whether there is one or more).

Reason 2. (Convenience). it is easier come out with singular names, than with plural ones. Objects can have irregular plurals or not plural at all, but will always have a singular one (with few exceptions like News).

- Customer
- Order
- User
- Status
- News

Reason 3. (Aesthetic and Order). Specially in master-detail scenarios, this reads better, aligns better by name, and have more logical order (Master first, Detail second):

- 1.Order
- 2.OrderDetail

Compared to:

- 1.OrderDetails
- 2.Orders

Reason 4 (Simplicity). Put all together, Table Names, Primary Keys, Relationships, Entity Classes... is better to be aware of only one name (singular) instead of two (singular class, plural table, singular field, singular-plural master-detail...)

- `Customer`
- `Customer.CustomerID`
- `CustomerAddress`
- `public Class Customer {...}`
- `SELECT FROM Customer WHERE CustomerID = 100`

Once you know you are dealing with "Customer", you can be sure you will use the same word for all of your database interaction

Reason 5. (Globalization). The world is getting smaller, you may have a team of different nationalities, not everybody has English as a native language. It would be easier for a non-native English language programmer to think of "Repository" than of "Repositories", or "Statuses" instead of "Status". Having singular names can lead to fewer errors caused by typos, save time by not having to think "is it Child or Children?", hence improving productivity.

Reason 6. (Why not?). It can even save you writing time, save you disk space, and even make your computer keyboard last longer!

- `SELECT Customer.CustomerName FROM Customer WHERE Customer.CustomerID = 100`

- `SELECT Customers.CustomerName FROM Customers WHERE Customers.CustomerID = 100`

You have saved 3 letters, 3 bytes, 3 extra keyboard hits :)

And finally, you can name those ones messing up with reserved names like:

- User > LoginUser, AppUser, SystemUser, CMSUser,...

Or use the infamous square brackets [User]

edited Jun 11 '18 at 8:50	community wiki 7 revs, 7 users 86% Nestor

---

570   I'll bet if you put a label on a sock drawer you'd call it "Socks". – Chris Ward Feb 6 '12 at 8:02

65   Definetelly. It is difficult to get one standard that works for all and for everybody, important is that works for you. This works for me, and here I have explained why, but again, that is for me, and I find it very convenient. – Nestor Feb 14 '12 at 12:10

402   The bigger question Chris, is why would you follow database naming conventions when naming a sock drawer? – Kyle Clegg Jul 25 '12 at 17:29

74   This answer needs more praise. It discusses a bunch of practical reasons that I apply to why I prefer singular names. The alternate discussions about proper language in reference to sets are just philosophical and are obscuring the real point. Singular just works better. – Jason Oct 8 '12 at 21:58

274   I have a "Sock" drawer at home, not a "Socks" drawer. If it were a database, I'd call it the "Sock" table. – jlembke Apr 11 '13 at 21:43

▲   If you use Object Relational Mapping tools or will in the future I suggest **Singular**.

My new rule of thumb is to judge how it will look once it's been converted into an object.

one table I've found that does not fit the new naming I use is UsersInRoles. But there will always be those few exceptions and even in this case it looks fine as UsersInRoles.Username.

answered Dec 3 '08 at 21:09

community wiki
Brian Boatright

---

45   I voted down and I'll tell you why, because I disagree. ORM by it's very nature is about mapping. Every ORM tool that I've ever used supports specifying the table name to be used for an entity when it is different from the entity's name. This is important because the whole reason we map to relational databases is so that we can easily make ad-hoc queries and reports with different shapes than our object model. Otherwise we'd all just be using object/document databases by now. – joshperry Aug 21 '10 at 0:51

22   The ORM should not dictate the names of the objects they map to. The point of ORM is an abstraction of the object, granting this flexibilty. – barrypicker Sep 14 '11 at 23:53

19   In my world I try to use consistent names across the whole project to avoid wasting time wondering if this instance has an s on the end or not. The fact that an ORM *can* rename and be independent doesnt mean that doing so is helping your fellow developers. – John Nicholas May 1 '12 at 15:38

2   Some ORM (like many programming tools) have default behavior that generates implementations without configuration... with the selling point of PRODUCTIVITY. So creating an Employee class, without explicit mapping, would generate an Employee table by default – user919426 Feb 14 '15 at 22:44

1   @barrypicker. Plural names don't just look dumb in ORM code. Plurals look bad in SQL too, especially when referring to a unique attribute. Who's never written select user.id from users? Or perhaps ...from users left join on thingy.user_id = user.id...? – Samuel Danielson Feb 24 '17 at 5:34

I prefer to use the **uninflected** noun, which in English happens to be singular.

201

Inflecting the number of the table name causes orthographic problems (as many of the other answers show), but choosing to do so because tables usually contain multiple rows is also semantically full of holes. This is more obvious if we consider a language that inflects nouns based on case (as most do):

Since we're usually doing something with the rows, why not put the name in the accusative case? If we have a table that we write to more than we read, why not put the name in dative? It's a table *of* something, why not use the genitive? We wouldn't do this, because the table is defined as an abstract container that exists regardless of its state or usage. Inflecting the noun without a precise and absolute semantic reason is babbling.

edited Apr 3 '17 at 17:48

community wiki
2 revs, 2 users 80%
Ian Mackinnon

35   Probably the most logical argument on this subject I've ever seen, and makes me glad I spent that time on Latin. +1 for sure. – user788472 Apr 22 '13 at 20:57

40   Well I clearly need to beef up my vocabulary. – TJ Biddle May 21 '13 at 4:54

17   +1 See, these are the kinds of answers the Internet needs more of. Impeccable proofs using rich vocabulary to execute perfect logic. – OCDev Oct 26 '13 at 23:54 🖉

8   I'll take note of this next time I'm programming in Latin. In the meantime users will go into the users table and customers into the customers table. – Caltor Aug 3 '15 at 23:18

8   Convinced – uninflected it is. Interesting to see that after all this time, the popular choices of "singular" and "plural" are **both** wrong! – Stuart Aug 21 '15 at 15:51 🖉

What convention requires that tables have singular names? I always thought it was plural names.

125

A user is added to the Users table.

This site agrees:
http://vyaskn.tripod.com/object_naming.htm#Tables

This site disagrees (but I disagree with it):
http://justinsomnia.org/writings/naming_conventions.html

As others have mentioned: these are just guidelines. Pick a convention that works for you and your company/project and stick with it. Switching between singular and plural or sometimes abbreviating words and sometimes not is much more aggravating.

edited Dec 3 '08 at 19:28

community wiki
2 revs
Michael Haren

44   When applying set theory to tables, any instance in the set is representative of the set, as Apple is an Apple set, it is agnostic of how many apples are

84   What if you have a bag with 5 apples inside? Do you call it a bag of apple? or a bag of apples? – Christopher Mahan Jan 3 '09 at 7:30

25   I think the theory would be that the set is named Apples. A singular apple is still "a set of Apples" - albeit, a set with a single instance. Multiple apples are also a "set of Apples". – Mark Brackett Apr 4 '09 at 2:16

25   @Christopher, if the raison d'être of the bag is to hold apples and only apples, then it is an "apple bag" , regardless of whether it contains 1 apple, 100 apples or no apples. – Ian Mackinnon Oct 7 '10 at 19:58 ✎

13   @ Ian: That's because a table is generic, and can be compared to a shipping container (can contain nearly anything, from apple crates to crates of Harley Davidson motorcycles). You say: a cargo container of oranges, not an orange cargo container. You say: a cargo container of car parts, not a car parts cargo container. If you made a custom data structure meant to hold only a specific type of data, such as names of apples, and you named it "kungabogo", then you could have an apple kungaboko. I know what your thinking, but think first of a sac of balls, and understand the difference in meaning. – Christopher Mahan Oct 8 '10 at 4:20

How about this as a simple example:

▲

70
```
SELECT Customer.Name, Customer.Address FROM Customer WHERE Customer.Name > "def"
```

▼
vs.

```
SELECT Customers.Name, Customers.Address FROM Customers WHERE Customers.Name > "def"
```

The SQL in the latter is stranger sounding than the former.

I vote for **singular**.

edited Apr 18 '15 at 2:16	community wiki 3 revs, 3 users 83% james

---

45   In that example, yes, but in practical sql it would never be written that way. You'd have a table alias so it'd be more like `SELECT C.Name, C.Address FROM Customers WHERE Customers.Name > 'def'` – Michael Haren Aug 21 '09 at 22:37

+1, (a year later) You cited a TERRIFIC example on how the singular makes sense. This is somewhat a religious debate. I was turned onto the singular several years ago by a data architect many years my senior and it's felt right to me (after requiring much convincing for me to switch). – Chris Adragna Oct 4 '10 at 13:52

> 6   How about using an alias/AS to get around that one issue? SELECT Customer.Name, Customer.Address FROM Customers AS Customer WHERE Customer.Name > "def" – James Hughes Jun 15 '11 at 16:45

> 1   The second one sounds much better, singular sounds like someone who can't speak English. – HLGEM Aug 30 '13 at 15:00

60

I am of the firm belief that in an Entity Relation Diagram, the entity should be reflected with a singular name, similar to a class name being singular. Once instantiated, the name reflects its instance. So with databases, the entity when made into a table (a collection of entities or records) is plural. Entity, User is made into table Users. I would agree with others who suggested maybe the name User could be improved to Employee or something more applicable to your scenario.

This then makes more sense in a SQL statement because you are selecting from a group of records and if the table name is singular, it doesn't read well.

edited Jan 3 '09 at 17:01	community wiki 2 revs Adam Carr

> 3   I especially like the SQL statement comment. Using singular here does not feel intuitive to the reader. – hochl Oct 5 '11 at 11:18

> 2   Excellent point about the ERD. I suspect this is why, to someone who sees the world through DBA eyes, singular naming makes sense. I suspect they don't get, as you point out, the difference between an entity and a collection of them. – William T. Mallard May 9 '14 at 1:02 ✏

> 1   A table is not a collections of records; a table is a definition of what a record looks like. That's the disconnect all the plural/singular folk seem to have. – rich remer Sep 21 '16 at 20:05

> I disagree with the SQL statement comment. What if you want to join against the Users.Id – hashtable Nov 30 '18 at 16:31

> One old_lady has many cat OR old_ladies have cats. I think ERDs read nicer as plural. And table of entities, tables have many entities so again I think plural sounds nice. – user3121518 Jul 8 at 2:05

37

I stick with **singular** for table names and any programming entity.

The reason? The fact that there are irregular plurals in English like *mouse* ⇒ *mice* and *sheep* ⇒ *sheep*. Then, if I need a *collection*, i just use *mouses* or *sheeps*, and move on.

It really helps the plurality stand out, and I can easily and programatically determine what the collection of things would look like.

7   what about a word ending with a 's'? If you have a table called 'News' (just as an example), what would you call the collection of news? Newss? Or would you call the table 'New'? – Anthony Aug 28 '09 at 11:35

18   I would call the table NewsItem and a collection NewsItems. – Ash Machine Aug 28 '09 at 20:46

4   What if you have to spell-check all code or else it will not compile ;) ? – Hamish Grubijan Jun 8 '10 at 15:33

2   The ORM should not dictate the names of the objects they map to. The point of ORM is an abstraction of the object, granting this flexibilty. – barrypicker Sep 14 '11 at 23:55

13   @HamishGrubijan then stop using Word to write your code! ;) – Valentino Vranken Feb 8 '12 at 12:22

31

Singular. I don't buy any argument involving which is most logical - every person thinks his own preference is most logical. No matter what you do it is a mess, just pick a convention and stick to it. We are trying to map a language with highly irregular grammar and semantics (normal spoken and written language) to a highly regular (SQL) grammar with very specific semantics.

My main argument is that I don't think of the tables as a set but as relations.

So, the `AppUser` relation tells which entities are `AppUsers` .

The `AppUserGroup` relation tells me which entities are `AppUserGroups`

The `AppUser_AppUserGroup` relation tells me how the `AppUsers` and `AppUserGroups` are related.

The `AppUserGroup_AppUserGroup` relation tells me how `AppUserGroups` and `AppUserGroups` are related (i.e. groups member of groups).

In other words, when I think about entities and how they are related I think of relations in singular, but of course, when I think of the entities in collections or sets, the collections or sets are plural.

In my code, then, and in the database schema, I use singular. In textual descriptions, I end up using plural for increased readability - then use fonts etc. to distinguish the table/relation name from the plural s.

I like to think of it as messy, but systematic - and this way there is always a systematically generated name for the relation I wish to express, which to me is very important.

1   exactly. the main thing many people are not aware here is what they are naming... you are giving name to a relation (a single record in the table), not the set of records in the table. – Alexandre Martini Dec 3 '14 at 16:55

2   Couldn't disagree more. 'Select * from Users where Name like 'J%'' because I am selecting all users where the name starts with 'J'. If your argument is that you want to write '...where User.Name like...' then simply use an alias. Same reason I say 'Give me a pair from all available socks.' – MarquelV Jul 9 '15 at 4:38 ✏

     If I was that particular my table name would be sock_pair – Manuel Hernandez May 3 '17 at 14:49

▲

30

▼

IMHO, table names should be **plural** like *Customers*.

Class names should be singular like *Customer* if it maps to a row in the *Customers* table.

edited Apr 18 '15 at 2:10   community wiki
                                             4 revs, 2 users 64%
                                             Gulzar Nazim

▲

28

▼

I also would go with **plurals**, and with the aforementioned *Users* dilemma, we do take the square bracketing approach.

We do this to provide uniformity between both database architecture and application architecture, with the underlying understanding that the *Users* table is a collection of *User* values as much as a *Users* collection in a code artifact is a collection of *User* objects.

Having our data team and our developers speaking the same conceptual language (although not always the same object names) makes it easier to convey ideas between them.

edited Apr 18 '15 at 2:23   community wiki
                                             2 revs, 2 users 71%
                                             Paulo Freitas

6   I agree.. why the inconsistency between code and storage? I would never name a collection of user objects "User" in code... so why would I call a table that? It makes no sense. When I read the arguments above about it, they are focusing on the entity, not the table... there is a distinction between whats in the table than the table itself in my mind. – Jason Mar 12 '12 at 20:39

1   By remembering that the singular of `companies` is `company` , and that this id is a reference to a singular item. It shouldn't bother us in code any more than it bothers us in English. – David Jan 4 '18 at 19:47

▲
20
▼

I personaly prefer to use plural names to represent a set, it just "sounds" better to my relational mind.

At this exact moment i am using singular names to define a data model for my company, because most of the people at work feel more confortable with it. Sometimes you just have to make life easier to everyone instead of imposing your personal preferences. (that's how i ended up in this thread, to get a confirmation on what should be the "best practice" for naming tables)

After reading all the arguing in this thread, i reached one conclusion:

I like my pancakes with honey, no matter what everybody's favorite flavour is. But if i am cooking for other people, i will try to serve them something they like.

answered Feb 26 '13 at 18:39   community wiki
                                       someone

It is not wise to use such convention in relational model world, especially when you describe relation between objects, e.g. "Each Team may have only one Main Coach and many secondary Coaches" , which is described: Team->MainCoach , Team->>SecondaryCoach – noonex Aug 2 '15 at 13:04 ✎

▲
16
▼

Singular. I'd call an array containing a bunch of user row representation objects 'users', but the table is 'the user table'. Thinking of the table as being nothing but the set of the rows it contains is wrong, IMO; the table is the metadata, and the set of rows is hierarchically attached to the table, it is not the table itself.

I use ORMs all the time, of course, and it helps that ORM code written with plural table names looks stupid.

answered Jan 2 '09 at 23:48   community wiki
                                       chaos

To each his own, I guess. A relational database table is by definition a heading (i.e. metadata naming the attributes) and a set of tuples matching the heading. You can focus on the metadata, whereas other folks focus on the tuples. :-) – Bill Karwin Jan 3 '09 at 19:43

2   The ORM should not dictate the names of the objects they map to. The point of ORM is an abstraction of the object, granting this flexibilty. – barrypicker Sep 14 '11 at 23:56

1   I look at it this way.. if you create an array/list/dictionary of anything in code, my bet is you name it with the plural name of whatever it holds. If you are using an ORM to abstract your database, tables are represented with some sort of collection, so why would you treat them any different? To use singular names may sound good, but you are always fighting your instinct that a table holds many of the same thing, just like a collection in code does. Why the inconsistency? – Jason Mar 12 '12 at 20:19

@Jason: Please compare and contrast the way these things read: 1) `$db->user->row(27)` , `$db->product->rows->where(something)` 2) `$db->users->row(27)` , `$db->products->rows->where(something)` . – chaos Mar 12 '12 at 20:35

---

15 ▲ ▼

I don't like plural table names because some nouns in English are not countable (water, soup, cash) or the meaning changes when you make it countable (chicken vs a chicken; meat vs bird). I also dislike using abbreviations for table name or column name because doing so adds extra slope to the already steep learning curve.

Ironically, I might make `User` an exception and call it `Users` because of USER (Transac-SQL), because I too don't like using brackets around tables if I don't have to.

I also like to name all the ID columns as `Id` , not `ChickenId` or `ChickensId` (what do plural guys do about this?).

All this is because I don't have proper respect for the database systems, I am just reapplying one-trick-pony knowledge from OO naming conventions like Java's out of habit and laziness. I wish there were better IDE support for complicated SQL.

answered Dec 3 '08 at 19:28   community wiki
Eugene Yokota

8   Us plural guys either name the 'id' column 'id' like you do, or 'singular_id'. I believe tables should be plural (think of them like arrays), but column names should be singular (attributes of a single element). – mpen Apr 4 '09 at 2:29

plu_ral/PluRal for table names, singular_id/singularId for primary keys. – hochl Oct 5 '11 at 11:26

---

14 ▲

We run similar standards, when scripting we demand [ ] around names, and where appropriate schema qualifiers - primarily it hedges your bets against future name grabs by the SQL syntax.

This has saved our souls in the past - some of our database systems have run 10+ years from SQL 6.0 through SQL 2005 - way past their intended lifespans.

answered Dec 3 '08 at 19:24

community wiki
stephbu

Seems like ritualistic self-flagellation. Has any such name grabs happened yet? – Kjetil S. Feb 20 at 22:02

I've actually always thought it was popular convention to use plural table names. Up until this point I've always used plural.

13

I can understand the argument for singular table names, but to me **plural** makes more sense. A table name usually describes what the table contains. In a normalized database, each table contains specific sets of data. Each row is an entity and the table contains many entities. Thus the plural form for the table name.

A table of cars would have the name *cars* and each row is a car. I'll admit that specifying the table along with the field in a `table.field` manner is the best practice and that having singular table names is more readable. However in the following two examples, the former makes more sense:

```
SELECT * FROM cars WHERE color='blue'
SELECT * FROM car WHERE color='blue'
```

Honestly, I will be rethinking my position on the matter, and I would rely on the actual conventions used by the organization I'm developing for. However, I think for my personal conventions, I'll stick with plural table names. To me it makes more sense.

edited Apr 18 '15 at 2:15

community wiki
2 revs, 2 users 50%
Randy

6   Isn't this the convention in RoR too? Plural names for tables and Singular for ORM classes? Makes a lot of sense to me. Table is called "cars" because it has many instances of "car" and class is called "Car" because it will hold one instance of a car!! – Sap Jul 14 '11 at 4:38

@Sap A minor correction of the latter part of your sentence - The class "Car" is an Abstract DataType representing a real-life Car. Whether it'll hold one instance or multiples depends on how it's used. – asgs Oct 10 '16 at 6:15

2   I really like this answer! Let me explain. If the collection is `car` and you want everything from the `car` that is `blue` the result should be something like `tire, mirror, engine`. And then it is getting confusing because all results are `parts` from a `car`. So the table name should be `carparts` (or `car_parts`, `CarParts` whatever you like) – arnoudhgz Jan 6 '17 at 13:18 ✏

Any database designer who enforces singular table names is basically declaring war against any Ruby on Rails app developers who may come into touch with that database in the future. Rail's strict insistence on singular words for classes, and pluralized names for tables, enables a lot of powerful behaviour within many gems inside Ruby's ecosystem. So even if you think singular sounds better, for the sake of compatibility you should stick to plural. I imagine this holds true for many other Object Relational Mappers too. – Kelsey Hannan Jan 26 '18 at 10:39 ✏

▲

12

▼

I am a fan of singular table names as they make my ER diagrams using CASE syntax easier to read, but by reading these responses I'm getting the feeling it never caught on very well? I personally love it. There is a good overview with examples of how readable your models can be when you use singular table names, add action verbs to your relationships and form good sentences for every relationships. It's all a bit of overkill for a 20 table database but if you have a DB with hundreds of tables and a complex design how will your developers ever understand it without a good readable diagram?

http://www.aisintl.com/case/method.html

As for prefixing tables and views I absolutely hate that practice. Give a person no information at all before giving them possibly bad information. Anyone browsing a db for objects can quite easily tell a table from a view, but if I have a table named tblUsers that for some reason I decide to restructure in the future into two tables, with a view unifying them to keep from breaking old code I now have a view named tblUsers. At this point I am left with two unappealing options, leave a view named with a tbl prefix which may confuse some developers, or force another layer, either middle tier or application to be rewritten to reference my new structure or name viewUsers. That negates a large part of the value of views IMHO.

answered Jan 2 '09 at 23:50   community wiki
                                     Shane Delmore

---

1   Good example of a pitfall of prefixing object names with a 'type' qualifier! – Kenny Evitt Nov 8 '10 at 16:14

▲

12

Tables: plural

Multiple users are listed in the users table.

> A singular user can be selected from the users table.

Controllers: plural

> http://myapp.com/users would list multiple users.

That's my take on it anyway.

answered Nov 25 '09 at 21:23   community wiki
Andrew

---

1   Closer to my take, but mine is that the table's storage of multiple users is actually incidental, and that any singular user is represented by the table, or rather the relation that is a set of tuples representing a User entity. – ProfK Nov 26 '09 at 8:54

▲

12

▼

The system `tables/views` of the server itself (`SYSCAT.TABLES` , `dbo.sysindexes` , `ALL_TABLES` , `information_schema.columns` , etc.) are almost always plural. I guess for the sake of consistency I'd follow their lead.

edited Apr 28 '18 at 14:26   community wiki
2 revs, 2 users 67%
Michel

Microsoft are what they are for business reasons first (and often unethical reasons at that), logical reasons last. My only reason for following them would be that they are the big gorilla and everyone else goes that way. When I have a choice, I choose the other way. – Bruce Patin Nov 30 '12 at 16:38

1   It should be noted that the `information_schema` is part of ISO/IEC 9075-11, the SQL standard. And yes, it does use plural tables/views names. – Paulo Freitas May 29 '18 at 0:53

▲

9

This may be a bit redundant, but I would suggest being cautious. Not necessarily that it's a bad thing to rename tables, but standardization is just that; a standard -- this database may already be "standardized", however badly :) -- I would suggest consistency to be a better goal given that this database already exists and presumably it consists of more than just 2 tables.

Practical consistency sometimes is the best standard... :)

my2cents ---

community wiki
Borzio

Possible alternatives:

- Rename the table SystemUser
- Use brackets
- Keep the plural table names.

IMO using brackets is technically the safest approach, though it is a bit cumbersome. IMO it's 6 of one, half-a-dozen of the other, and your solution really just boils down to personal/team preference.

9

edited Jan 3 '09 at 7:20

community wiki
2 revs
Dave Markle

2   I like your 'prefix' idea, but would call it SystemUser. – ProfK   Dec 5 '08 at 14:50

My take is in semantics depending on how you define your container. For example, A "bag of apples" or simply "apples" or an "apple bag" or "apple".

9

Example: a "college" table can contain 0 or more colleges a table of "colleges" can contain 0 or more collegues

```
a "student" table can contain 0 or more students
a table of "students" can contain 0 or more students.
```

My conclusion is that either is fine but you have to define how you (or people interacting with it) are going to approach when referring to

8

As others have mentioned here, conventions should be a tool for adding to the ease of use and readability. Not as a shackle or a club to torture developers.

That said, my personal preference is to use singular names for both tables and columns. This probably comes from my programming background. Class names are generally singular unless they are some sort of collection. In my mind I am storing or reading individual records in the table in question, so singular makes sense to me.

This practice also allows me to reserve plural table names for those that store many-to-many relationships between my objects.

I try to avoid reserved words in my table and column names, as well. In the case in question here it makes more sense to go counter to the singular convention for Users to avoid the need to encapsulate a table that uses the reserved word of User.

I like using prefixes in a limited manner (tbl for table names, sp_ for proc names, etc), though many believe this adds clutter. I also prefer CamelBack names to underscores because I always end up hitting the + instead of _ when typing the name. Many others disagree.

Here is another good link for naming convention guidelines: http://www.xaprb.com/blog/2008/10/26/the-power-of-a-good-sql-naming-convention/

Remember that the most important factor in your convention is that it make sense to the people interacting with the database in question. There is no "One Ring to Rule Them All" when it comes to naming conventions.

answered Dec 3 '08 at 19:11 community wiki
Chris

---

18 Ignoring the horrors of hungarian notation. Never, never, never use sp_ in front of stored procedures because MS-SQL uses that for system stored procedures and treats them special. Since sp_ are stored in the master table, MS-SQL always looks thier first even if you qualify the location. –
Will Dieterich Dec 8 '08 at 7:59

I think using the singular is what we were taught in university. But at the same time you *could* argue that unlike in object oriented programming, a table is not an instance of its records.

in Chinese languages there are no plural forms anyway.

answered Mar 10 '10 at 11:08

community wiki
helloworlder

8

I've always used singular simply because that's what I was taught. However, while creating a new schema recently, for the first time in a long time, I actively decided to maintain this convention simply because... it's shorter. Adding an 's' to the end of every table name seems as useless to me as adding 'tbl_' in front of every one.

edited Oct 10 '16 at 6:11

community wiki
2 revs, 2 users 67%
Just Some Guy

8

If we look at `MS SQL Server's` system tables, their names as assigned by Microsoft are in `plural` .

The Oracle's system tables are named in `singular` . Although a few of them are plural. Oracle recommends plural for user-defined table names. That doesn't make much sense that they recommend one thing and follow another. That the architects at these two software giants have named their tables using different conventions, doesn't make much sense either... After all, what are these guys ... PhD's?

I do remember in academia, the recommendation was singular.

For example, when we say:

```
select OrderHeader.ID FROM OrderHeader WHERE OrderHeader.Reference = 'ABC123'
```

maybe b/c each `ID` is selected from a particular single row ...?

edited Apr 28 '18 at 13:34

community wiki
2 revs, 2 users 85%
Richard

Microsoft are what they are for business reasons first (and often unethical reasons at that), logical reasons last. My only reason for following them would

you are 'Selecting all IDs from OrderHeaders where something is true' but if you had to use table names because of a join or whatever, you would use an alias like so... 'select OrderHeader.ID FROM OrderHeaders as OrderHeader WHERE OrderHeader.Reference = 'ABC123' – MarquelV Jul 9 '15 at 4:48 ✎

▲

7

▼

I always thought that was a dumb convention. I use plural table names.

(I believe the rational behind that policy is that it make it easier for ORM code generators to produce object & collection classes, since it is easier to produce a plural name from a singular name than vice-versa)

answered Dec 3 '08 at 18:17　　　　community wiki
James Curran

11　This convention has been part of relational theory long, long, before ORM ever existed. – ProfK　Dec 5 '08 at 14:51

1　The ORM should not dictate the names of the objects they map to. The point of ORM is an abstraction of the object, granting this flexibilty. – barrypicker Sep 15 '11 at 0:07

▲

7

▼

I only use nouns for my table names that are spelled the same, whether singular or plural:

moose fish deer aircraft you pants shorts eyeglasses scissors species offspring

answered May 27 '12 at 18:51　　　　community wiki
bobby

▲

7

▼

I once used "Dude" for the User table - same short number of characters, no conflict with keywords, still a reference to a generic human. If I weren't concerned about the stuffy heads that might see the code, I would have kept it that way.

answered Nov 30 '12 at 16:28　　　　community wiki
Bruce Patin

Guidelines are really there as just that. It's not "set in stone" that's why you have the option of being able to ignore them.

5   I would say that it's more logically intuitive to have pluralized table names. A table is a collection of entity after all. In addition to other alternatives mentioned I commonly see prefixes on table names...

- tblUser

- tblThis

- tblThat

- tblTheOther

I'm not suggesting this is the way to go, I also see spaces used a LOT in table names which I abhor. I've even come across field names with idiotic characters like ? as if to say this field answers a question.

answered Dec 3 '08 at 18:19   community wiki
                                      BenAlabaster

---

4   Agreed. Spaces and prefixes are from the Devil. – Dave Markle Dec 3 '08 at 18:24

2   MSAccess encourages table names with spaces. I suspect many MSSQL tables in spaces were imported from there. – James Curran Dec 3 '08 at 18:26

4   +1 again. Tables and Fields named with spaces are the mark of the whizz kid Office Junior making a real kool access application for Beryl in account :-). – Cruachan Dec 3 '08 at 20:30

1   Hey, I like Beryl in accounts... but it wouldn't ever cause me to put prefixes or spaces in my table names... nor would it cause me to put question marks or exclamation marks in my field names. I don't care how cute she is. :P – BenAlabaster Dec 3 '08 at 20:53

3   tbl as a prefix is the son of devilspawn and smelly carrion, but semantic prefixes are a little better. In our main application, we, Chase Software, prefix all our tables with 'ca' or 'cs', being chase application, or chase system. – ProfK Dec 5 '08 at 14:52

---

1 2 next