

# Is there a naming convention for MySQL?

Asked 13 years, 2 months ago   Modified 1 year ago   Viewed 199k times



Here's how I do it:

211



1. Table names are lower case, use underscores to separate words, and are singular (e.g. `foo`, `foo_bar`, etc.).
2. I generally (not always) have an auto increment PK. I use the following convention: `tablename_id` (e.g. `foo_id`, `foo_bar_id`, etc.).
3. When a table contains a column that is a foreign key, I just copy the column name of that key from whatever table it came from. For example, say table `foo_bar` has the FK `foo_id` (where `foo_id` is the PK of `foo`).
4. When defining FKs to enforce referential integrity, I use the following: `tablename_fk_columnname` (e.g. furthering example 3, it would be `foo_bar_foo_id`). Since this is a table name/column name combination, it is guaranteed to be unique within the database.
5. I order the columns like this: PKs, FKs, then the rest of columns alphabetically

Is there a better, more standard way to do this?

[mysql](#)[naming-conventions](#)[mysql-workbench](#)[Share](#)[Improve this question](#)[Follow](#)

edited Nov 30, 2023 at 5:33

[Pang](#)

10.1k ● 146 ● 85 ● 124

asked Oct 26, 2011 at 6:08

[StackOverflowNewbie](#)

40.6k ● 113 ● 288 ● 451

---

10 Is it wrong to use for auto increment PK just "id"? Why? The name of the column have meaning only in the context of the table. So I have one "id" in every table, and may have many `id_<table_name>` for FK. – [Zbyszek](#) May 31, 2013 at 11:25

---

3 @Zbyszek I think the simplest reason against that is simply for consistency/simplicity. Rather than having `id_tableB` => *oh no different-named column* `id`, the consistency of `id_tableB` => `id_tableB` just looks neater... or as OP does it: `foo_id` => `foo_id` rather than `foo_id` => `id` – [Don Cheadle](#) Jan 26, 2015 at 22:15 ✎

---

1 This is a great method. I use almost the same, but for table names, I use plural for "root" table and singular for dependencies. Eg: companies and company\_sector. – [leandronn](#) Jul 8, 2021 at 20:40 ✎

---

"Table names (...) are singular" that would be an instant fail in computer engineering db class. Not because of naming conventions, but because "tables" are a group of something. If it's single, it indicates bad design. – [pmiguelpinto90](#) Sep 14, 2021 at 8:32

---

I like the dot 'member' access separator for table names. I'm not sure what it's actually called in C# and stuff. Member access dot? Anyway, in table names it makes things easy to recognise and understand. `Accounts.Users` (or `account.user` ;) – [n00dles](#) Aug 18 at 17:29

---

## 7 Answers

Sorted by:

Highest score (default)



140



I would say that first and foremost: be consistent.

I reckon you are almost there with the conventions that you have outlined in your question. A couple of comments though:

Points 1 and 2 are good I reckon.

Point 3 - sadly this is not always possible. Think about how you would cope with a single table `foo_bar` that has columns `foo_id` and `another_foo_id` both of which reference the `foo` table `foo_id` column. You might want to consider how to deal with this. This is a bit of a corner case though!

Point 4 - Similar to Point 3. You may want to introduce a number at the end of the foreign key name to cater for having more than one referencing column.

Point 5 - I would avoid this. It provides you with little and will become a headache when you want to add or remove columns from a table at a later date.

Some other points are:

## Index Naming Conventions

You may wish to introduce a naming convention for indexes - this will be a great help for any database metadata work that you might want to carry out. For example you might just want to call an index `foo_bar_idx1` or `foo_idx1` - totally up to you but worth considering.

## Singular vs Plural Column Names

It might be a good idea to address the thorny issue of plural vs single in your column names as well as your table name(s). This subject often causes [big debates](#) in the DB community. I would stick with singular forms for both table names and columns. There. I've said it.

The main thing here is of course consistency!

Share Improve this answer

Follow

edited May 23, 2017 at 12:34



Community Bot

1 • 1

answered Oct 26, 2011 at 7:09



Tom Mac

9,843 • 3 • 30 • 39


---

What would be better than point 5? Why it could become a headache? – [Rasshu](#) Mar 28, 2014 at 0:58


- 
- 9 To follow up, i found this very helpful for those who will come here later: [launchbylunch.com/posts/2014/Feb/16/sql-naming-conventions/...](http://launchbylunch.com/posts/2014/Feb/16/sql-naming-conventions/) – [Enissay](#) Feb 2, 2016 at 14:59
-

1 I'm having trouble finding a good "scheme" for naming my tables which hold objects of models consisting of two names (DocumentChapter, DocumentVersion, DocumentType etc.). For example for DocumentType i could name it `documenttype` or `document_type` . I would prefer the latter one, but most of the time i have a many to many relationship and i need a table looking like `document_document_type` . Any suggestions how to handle this? – [malifa](#) Apr 7, 2016 at 15:28

---

@rsb2097 Re: point 5 - After adding one column (to the end) the order could become invalid. You shouldn't add any constraint that requires re-ordering the columns as that's an unnecessary overhead. – [Will Sheppard](#) Aug 21, 2018 at 10:26 

---

I would add to that, to make a small note of your naming convention for any possible future db fondlers who follow on. It takes 5 minutes and mitigates confusion quite effectively. (It also helps you if you come back after a break ;)) – [n00dles](#) Aug 18 at 17:16 



Consistency is the key to any naming standard. As long as it's logical and consistent, you're 99% there.

**28**



The standard itself is very much personal preference - so if you like your standard, then run with it.



To answer your question outright - no, MySQL doesn't have a preferred naming convention/standard, so rolling your own is fine (and yours seems logical).

Share Improve this answer

answered Oct 26, 2011 at 6:16

Follow



mal-wan

4,476 ● 4 ● 27 ● 38



23



MySQL has a short description of their more or less strict rules:

<https://dev.mysql.com/doc/internals/en/coding-style.html>

The most common coding style for MySQL by Simon Holywell:

<https://www.sqlstyle.guide/>

See also this question:

[Are there any published coding style guidelines for SQL?](#)

Share Improve this answer

edited Jul 12, 2023 at 21:51

Follow

answered Oct 19, 2016 at 14:40



Daniel W.

32.2k ● 15 ● 98 ● 154

3 This is a good reference. sqlstyle.guide has a very comprehensive set of guidelines that cover just about any case. – [SolStack](#) Jul 19, 2021 at 22:41



Thankfully, PHP developers aren't "Camel case bigots" like some development communities I know.

5

Your conventions sound fine.



Just so long as they're a) simple, and b) consistent - I don't see any problems :)



PS: Personally, I think 5) is overkill...

Share Improve this answer

edited Jan 19, 2013 at 4:44

Follow



Camilo Martin

37.8k ● 21 ● 118 ● 156

answered Oct 26, 2011 at 6:11



paulsm4

121k ● 21 ● 171 ● 238

---

2 Far from being bigots, camel case is actually frowned upon by many of the DB community because some of the RDBMS' are case insensitive to the point where they will strip cases (or change everything to upper case) so things get really ugly really quickly. – [mal-wan](#) Oct 26, 2011 at 6:21

---

11 I like the irony of MySQL users disliking CamelCase and the name of the product we use: MySQL, written in CamelCase – [DBX12](#) Dec 5, 2016 at 19:43

---

3 @DBX12 To be pedantic, that's PascalCase, not camelCase, but your point still stands. – [MarredCheese](#) Feb 24, 2019 at 0:07

---

I love camelCase, not PascalCase but I think databases should do away with capitalisation and stick with `u_n_d_e_r_s_c_o_r_e_s`. It just feels right. Also, I thought it was `mySQL`. Like `eMail`. Or `iPhone`. – [n00dles](#) Aug 18 at 17:18

---



## Simple Answer: NO

5



Well, at least a naming convention as such encouraged by Oracle or community, no, however, basically you have to be aware of following the rules and limits for identifiers, such as indicated in MySQL documentation:



<https://dev.mysql.com/doc/refman/8.0/en/identifiers.html>



About the naming convention you follow, I think it is ok, just the number 5 is a little bit unnecessary, I think most visual tools for managing databases offer a option for sorting column names (I use DBeaver, and it have it), so if the purpouse is having a nice visual presentation of your table you can use this option I mention.

By personal experience, I would recommed this:

- **Use lower case.** This almost ensures interoperability when you migrate your databases from one server to another. Sometimes the `lower_case_table_names` is not correctly configured and your server start throwing errors just by simply unrecognized your camelCase or PascalCase standard (case sensitivity problem).
- **Short names.** Simple and clear. The most easy and fast is identify your table or columns, the better. Trust me, when you make a lot of different queries in a short amount of time is better having all simple to write (and read).



- **Avoid prefixes.** Unless you are using the same database for tables of different applications, don't use prefixes. This only add more verbosity to your queries. There are situations when this could be useful, for example, when you want to indentify primary keys and foreign keys, that usually table names are used as prefix for id columns.
- **Use underscores for separating words.** If you still want to use more than one word for naming a table, column, etc., so use underscores for *separating\_the\_words*, this helps for legibility (your eyes and your stressed brain are going to thank you).
- **Be consistent.** Once you have your own standard, follow it. Don't be the person that create the rules and is the first who breaking them, that is shameful.

And what about the "Plural vs Singular" naming? Well, this is most a situation of personal preferences. In my case I try to use plural names for tables because I think a table as a collection of elements or a package containig elements, so a plural name make sense for me; and singular names for columns because I see columns as attributes that describe singularly to those table elements.

Share Improve this answer

Follow

answered Jul 2, 2020 at 22:45



**Fabrizio Valencia**

22k ● 4 ● 18 ● 19



Consistency is what everyone strongly suggest, the rest is upto you as long as it works.

1



For beginners its easy to get carried away and we name whatever we want at that time. This make sense at that point but a headache later.



`foo` `foobar` or `foo_bar` is great. We name our table straight forward as much as possible and only use underscore if they are two different words.

`studentregistration` to `student_registration`

like @Zbyszek says, having a simple `id` is more than enough for the auto-increment. The simplier the better. Why do you need `foo_id` ? We had the same problem early on, we named all our columns with the table prefix. like `foo_id` , `foo_name` , `foo_age` . We dropped the tablename now and kept only the col as short as possible.

Since we are using just an id for PK we will be using `foo_bar_fk` (tablename is unique, folowed by the unique PK, followed by the `_fk` ) as foreign key. We don't add `id` to the col name because it is said that the name 'id' is always the PK of the given table. So we have just the tablename and the `_fk` at the end.

For constrains we remove all underscores and join with camelCase (tablename + Colname + Fk)

`foobarUsernameFk` (for username\_fk col). It's just a way

we are following. We keep a documentation for every names structures.

When keeping the col name short, we should also keep an eye on the RESTRICTED names.

```
+-----+
|          foobar          |
+-----+
| id (PK for the current table) |
| username_fk (PK of username table) |
| location (other column)      |
| tel (other column)           |
+-----+
```

Share Improve this answer

answered Nov 16, 2020 at 6:20

Follow



Dexter

9,254 ● 4 ● 46 ● 46



0



as @fabrizio-valencia said use lower case. in windows if you export mysql database (phpmyadmin) the tables name will converted to lower case and this lead to all sort of problems. see [Are table names in MySQL case sensitive?](#)



Share Improve this answer

answered Nov 10, 2020 at 6:47



Follow



stren-12

57 ● 2