

# SQL Table Aliases - Good or Bad?

## [closed]

Asked 16 years, 4 months ago    Modified 8 years, 1 month ago

Viewed 45k times



47



**Closed.** This question is [opinion-based](#). It is not currently accepting answers.



**Want to improve this question?** Update the question so it can be answered with facts and citations by [editing this post](#).

Closed 11 years ago.

[Improve this question](#)

What are the pros and cons of using table aliases in SQL? I personally try to avoid them, as I think they make the code less readable (especially when reading through large where/and statements), but I'd be interested in hearing any counter-points to this. When is it generally a good idea to use table aliases, and do you have any preferred formats?

sql

alias

Share

Improve this question

Follow

edited Sep 12, 2016 at 20:47



Brian Tompsett - 汤莱恩

5,875 ● 72 ● 61 ● 133

asked Aug 14, 2008 at 13:53



tbreffni

5,132 ● 5 ● 32 ● 30

---

4 [Table aliases are da bomb.](#) – Nick Chammas Dec 2, 2011 at 22:17

---

17 Answers

Sorted by: Highest score (default)



31



Table aliases are a necessary evil when dealing with highly normalized schemas. For example, and I'm not the architect on this DB so bear with me, it can take 7 joins in order to get a clean and complete record back which includes a person's name, address, phone number and company affiliation.

Rather than the somewhat standard single character aliases, I tend to favor short word aliases so the above example's SQL ends up looking like:

```
select person.FirstName
       ,person.LastName
       ,addr.StreetAddress
       ,addr.City
       ,addr.State
       ,addr.Zip
       ,phone.PhoneNumber
       ,company.CompanyName
from tblPeople person
```

```
left outer join tblAffiliations affl on affl.personID
left outer join tblCompany company on company.companyI
```

... etc

Share Improve this answer

edited Aug 14, 2008 at 14:00

Follow

answered Aug 14, 2008 at 13:56



[Rob Allen](#)

17.7k ● 6 ● 53 ● 70

---

21 "Necessary Evil" seems a bit strong. At worst, its not necessary and at best it is necessary to use them. However, most of the time, using aliases adds to the readability of the code. – [ScottCher](#) Oct 14, 2008 at 17:52

---

2 How is this example better than using the literal table names? i.e. select tblPerson.FirstName instead of select person.FirstName, etc – [Chris Schiffhauer](#) Nov 1, 2016 at 19:18

---

4 SELECT p.FirstName is much less cluttered that SELECT tblPerson.FirstName. Forget about a, b, c as aliases - they add to the confusion instead of reducing it. But using p for Person, o for Order etc makes code **much** more readable in a big query. – [Reversed Engineer](#) Apr 26, 2017 at 10:45

---



Well, there are some cases you *must* use them, like when you need to join to the same table twice in one query.

**31**

It also depends on wether you have unique column names across tables. In our legacy database we have 3-



letter prefixes for all columns, stemming from an abbreviated form from the table, simply because one ancient database system we were once compatible with didn't support table aliases all that well.

If you have column names that occur in more than one table, specifying the table name as part of the column reference is a must, and thus a table alias will allow for a shorter syntax.

Share Improve this answer

answered Aug 14, 2008 at 13:55

Follow



[Lasse V. Karlsen](#)

391k ● 106 ● 646 ● 844

---

4 Correlated subquery is another example where you need alias. – [Royi Namir](#) May 26, 2014 at 6:11

---

table variable also requires alias to reference column names  
– [montelof](#) Jun 17, 2015 at 15:46

---



26

Am I the only person here who really hates them?

Generally, I don't use them unless I have to. I just really hate having to read something like



```
select a.id, a.region, a.firstname, a.blah, b.yadda, b
from table toys as a
inner join prices as b on a.blah = b.yadda
inner join customers as c on c.crap = something else
etc
```

When I read SQL, I like to know exactly what I'm selecting when I read it; aliases actually confuse me more because I've got to slog through lines of columns before I actually get to the table name, which generally represents information about the data that the alias doesn't. Perhaps it's okay if you made the aliases, but I commonly read questions on StackOverflow with code that seems to use aliases for no good reason. (Additionally, sometimes, someone will create an alias in a statement and just not use it. Why?)

I think that table aliases are used so much because a lot of people are averse to typing. I don't think that's a good excuse, though. That excuse is the reason we end up with terrible variable naming, terrible function acronyms, bad code...I would take the time to type out the full name. I'm a quick typer, though, so maybe that has something to do with it. (Maybe in the future, when I've got carpal tunnel, I'll reconsider my opinion on aliases. :P ) I especially hate running across table aliases in PHP code, where I believe there's absolutely no reason to have to do that - you've only got to type it once!

I always use column qualifiers in my statements, but I'm not averse to typing a lot, so I will gladly type the full name multiple times. (Granted, I do abuse MySQL's tab completion.) Unless it's a situation where I have to use an alias (like some described in other answers), I find the extra layer of abstraction cumbersome and unnecessary.

**Edit:** (Over a year later) I'm dealing with some stored procedures that use aliases (I did not write them and I'm new to this project), and they're kind of painful. I realize that the reason I don't like aliases is because of how they're defined. You know how it's generally good practice to declare variables at the top of your scope? (And usually at the beginning of a line?) Aliases in SQL don't follow this convention, which makes me grind my teeth. Thus, I have to search the entire code for a single alias to find out where it is (and what's frustrating is, I have to read through the logic before I find the alias declaration). If it weren't for that, I honestly might like the system better.

If I ever write a stored procedure that someone else will have to deal with, I'm putting my alias definitions in a comment block at the beginning of the file, as a reference. I honestly can't understand how you guys don't go crazy without it.

[Share](#) [Improve this answer](#)

[edited Nov 7, 2012 at 20:12](#)


[Follow](#)

answered Jun 9, 2011 at 21:40



user677526

- 
- 6 That's certainly a bad usage of aliases. Ideally aliases will at least hint as to the table they are from. In simple cases, single characters will do, but in some cases I find it is really helpful to identify the kind of data I am getting. For instance, if I am joining tblAddress for the mailing and shipping

addresses, I may alias it as mailAdd, and shipAdd so that it is crystal clear not only that the alias is an address, but exactly which address type it is. using aliases like a, b, c is simply bad practice and I think even most who like using aliases would agree that's c.crap – [CStroliaDavis](#) Nov 3, 2011 at 17:52 

---

Table aliases are necessary in SQL to name subselect tables & to join tables with common column names. Alias scoping is straightforward. The problem is that the SELECT clause belongs not at the start but after FROM-WHERE-GROUP-HAVING and before ORDER BY. – [philipxy](#) Jun 30, 2017 at 5:19

---

I do a lot of joining c.crap (customers crap)! – [dustbuster](#) May 19, 2023 at 15:31

---

Absolutely on board with this; aliases make SQL less readable as you have to track down their definition in order to see what they refer to. – [Matt Arnold](#) Jul 31, 2023 at 9:16

---



## Good

9



As it has been mentioned multiple times before, it is a good practice to prefix *all* column names to easily see which column belongs to which table - and aliases are shorter than full table names so the query is easier to read and thus understand. If you use a good aliasing scheme of course.

And if you create or read the code of an application, which uses externally stored or dynamically generated table names, then without aliases it is really hard to tell at the first glance what all those "%s"es or other

placeholders stand for. It is not an extreme case, for example many web apps allow to customize the table name prefix at installation time.

Share Improve this answer

edited Nov 5, 2008 at 17:45

Follow

answered Nov 5, 2008 at 17:34



user30932



8



Microsoft SQL's query optimiser benefits from using either fully qualified names or aliases.

Personally I prefer aliases, and unless I have a lot of tables they tend to be single letter ones.



```
--seems pretty readable to me ;-)  
select a.Text  
from Question q  
      inner join Answer a  
      on a.QuestionId = q.QuestionId
```

There's also a practical limit on how long a Sql string can be executed - aliases make this limit easier to avoid.

Share Improve this answer

answered Aug 14, 2008 at 14:00

Follow



Keith

155k ● 82 ● 306 ● 446



---

how does the query optimizer benefit from using aliases??

– [Tao](#) Oct 20, 2011 at 21:59

---

@Tao - in my example above I use `a.Text` , but `Text` would work too (so long as only `Answer` has a field called `Text` ). You save the query planner from needing to figure that out by using either `Answer.Text` or `a.Text` . It doesn't matter whether you use the table name or the alias, but qualifying the field helps. – [Keith](#) Oct 23, 2011 at 21:43

---

What you're saying sounds reasonable from a human point of view, but I have never heard anything to that effect about SQL Server's query optimizer. When you say "qualifying the field helps", is that just your instinct talking, or do you have any data or references that suggest this? I'm not trying to cause trouble here, just very interested if this is true! (it would reflect very poorly on the optimizer if so...) – [Tao](#) Oct 24, 2011 at 10:44

---

@Tao - I've not looked at this recently, but in SQL2000 not fully qualifying field names would result in ad-hoc query plans being discarded. Also `a.Text` just looks up the fields and statistics in the one table, while `Text` needs to check all the tables in the query (bear in mind that can be thousands of fields across hundreds of joins). I haven't compared them directly in terms of query compilation time, but it would be interesting to do so. – [Keith](#) Oct 24, 2011 at 14:41

---

Do you have documentation showing that SQL2000 will discard ad-hoc query plans because of unqualified field names? That seems really weird, since it implies the canonicalization of the columns is not a quality of the parser, but is somehow exposed to the optimizer. – [Robert Fischer](#) May 14, 2012 at 19:59

---



5

If I write a query myself (by typing into the editor and not using a designer) I always use aliases for the table name just so I only have to type the full table name once.



I really hate reading queries generated by a designer with the full table name as a prefix to every column name.



Share Improve this answer

answered Aug 14, 2008 at 14:02

Follow



BTB

2,146 ● 3 ● 21 ● 22



5

I suppose the only thing that really speaks against them is excessive abstraction. If you will have a good idea what the alias refers to (good naming helps; 'a', 'b', 'c' can be quite problematic especially when you're reading the statement months or years later), I see nothing wrong with aliasing.



As others have said, joins *require* them if you're using the same table (or view) multiple times, but even outside that situation, an alias can serve to clarify a data source's purpose in a particular context. In the alias's name, try to answer *why* you are accessing particular data, not *what* the data is.

Share Improve this answer

answered Aug 18, 2008 at 10:00

Follow



Sören Kuklau

19.9k ● 8 ● 55 ● 90



4



I LOVE aliases!!!! I have done some tests using them vs. not and have seen some processing gains. My guess is the processing gains would be higher when you're dealing with larger datasets and complex nested queries than without. If I'm able to test this, I'll let you know.



Share Improve this answer



Follow

edited Nov 8, 2012 at 18:20



Chris Gerken

16.4k ● 6 ● 47 ● 60

answered Nov 8, 2012 at 17:50



Kirk Harrington

49 ● 1



3



You need them if you're going to join a table to itself, or if you use the column again in a subquery...

Share Improve this answer

Follow

answered Aug 14, 2008 at 13:57



Shawn

19.8k ● 20 ● 100 ● 153



2



Aliases are great if you consider that my organization has table names like:

SchemaName.DataPointName\_SubPoint\_Sub-SubPoint\_Sub-Sub-SubPoint... My team uses a pretty standard set of abbreviations, so the guesswork is



minimized. We'll have say ProgramInformationDataPoint shortened to pidp, and submissions to just sub.

The good thing is that once you get going in this manner and people agree with it, it makes those HAYUGE files just a little smaller and easier to manage. At least for me, fewer characters to convey the same info seems to go a little easier on my brain.

Share Improve this answer  
Follow

answered Aug 25, 2008 at 18:40



Pulsehead

5,170 ● 9 ● 36 ● 37



1

I like long explicit table names (it's not uncommon to be more than 100 characters) because I use many tables and if the names aren't explicit, I might get confused as to what each table stores.



So when I write a query, I tend to use shorter aliases that make sense within the scope of the query and that makes the code much more readable.

Share Improve this answer  
Follow

answered Aug 14, 2008 at 13:57



mbillard

38.8k ● 18 ● 75 ● 98



1

I always use aliases in my queries and it is part of the code guidebook in my company. First of all you need aliases or table names when there are columns with identical names in the joining tables. In my opinion the



aliases improve readability in complex queries and allow me to see quickly the location of each columns. We even use aliases with single table queries, because experience has shown that single table queries don't stay single table for long.

Share Improve this answer

answered Aug 18, 2008 at 20:07

Follow



Bjorgvin



1



IMHO, it doesn't really matter with short table names that make sense, I have on occasion worked on databases where the table name could be something like VWRECOFLY or some other random string (dictated by company policy) that really represents users, so in that case I find aliases really help to make the code FAR more readable. (users.username makes a lot more sense than VWRECOFLY.username)

Share Improve this answer

edited Jul 26, 2013 at 16:00

Follow



Daniel Daranas

22.6k ● 9 ● 65 ● 121

answered Aug 14, 2008 at 13:56



Adam Lerman

3,399 ● 9 ● 43 ● 53



0

I always use aliases, since to get proper performance on MSSQL you need to prefix with schema at all times. So you'll see a lot of



Select Person.Name From  
dbo.Person As Person



Share Improve this answer

answered Aug 14, 2008 at 14:33

Follow



Stu

15.8k ● 4 ● 45 ● 74



0

I always use aliases when writing queries. Generally I try and abbreviate the table name to 1 or 2 representative letters. So Users becomes u and debtor\_transactions becomes dt etc...



It saves on typing and still carries some meaning.



The shorter names makes it more readable to me as well.



Share Improve this answer

answered Aug 18, 2008 at 9:47

Follow



Craig

1,337 ● 1 ● 13 ● 15



0

If you do not use an alias, it's a bug in your code just waiting to happen.



```
SELECT Description -- actually in a
FROM
table_a a,
table_b b
WHERE
a.ID = b.ID
```



What happens when you do a little thing like add a column called Description to Table\_B. That's right, you'll get an error. Adding a column doesn't need to break anything. I never see writing good code, bug free code, as a necessary evil.

Share Improve this answer

answered Oct 14, 2008 at 17:57

Follow



Mark Brady

---

2 Agree, but the alias is not the point, it's the table name qualification. So without an alias this could have been written: `SELECT table_a.Description` – [WW](#). Oct 29, 2008 at 6:59

---

3 Agreed that there is a difference between table aliases and qualifying a column... but they go hand in glove. It would be silly to alias a table and still use the table name in select list. it's equally silly to use the table name ad nauseum in the select list when SQL allows for meaningful aliases  
– Mark Brady Oct 30, 2008 at 19:20

---



Aliases are required when joining tables with columns that have identical names.

**-1**



Share Improve this answer

edited Nov 1, 2016 at 19:19

Follow



[Chris Schiffhauer](#)

17.3k ● 15 ● 81 ● 89

answered Aug 15, 2008 at 15:45



[Juha Syrjälä](#)

34.2k ● 33 ● 136 ● 189



---

2 This is incorrect. You can use the table's actual name rather than an alias. – [Chris Schiffhauer](#) Nov 1, 2016 at 19:20

---

Aliases are required when you join a table to itself, or when you join to the same table multiple times. – [Ineffable21](#) Nov 25, 2022 at 14:22

---