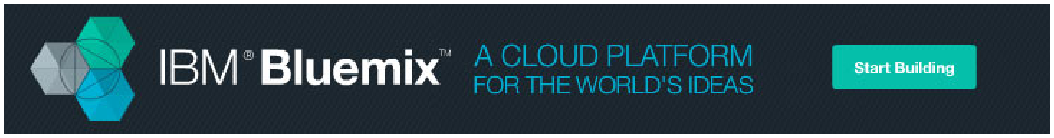
In MySQL queries, why use join instead of where?



It seems like to combine two or more tables, we can either use join or where. What are the advantages of one over the other?

sql mysql database

edited Feb 11 '10 at 16:34
OMG Ponies
157k ● 21 ● 262 ● 344

asked Feb 11 '10 at 3:08
extraeee
940 ● 1 ● 14 ● 23

1 Depending on your usage, it possibly has the same results, but with a JOIN you can do other kinds of associations. – zneak Feb 11 '10 at 3:10

5 Answers

Any query involving more than one table requires some form of association to link the results from table "A" to table "B". The traditional (ANSI-89) means of doing this is to:

1. List the tables involved in a comma separated list in the FROM clause
2. Write the association between the tables in the WHERE clause

```
SELECT *
  FROM TABLE_A a,
       TABLE_B b
 WHERE a.id = b.id
```

Here's the query re-written using ANSI-92 JOIN syntax:

```
SELECT *
  FROM TABLE_A a
  JOIN TABLE_B b ON b.id = a.id
```

From a Performance Perspective:

Where supported (Oracle 9i+, PostgreSQL 7.2+, MySQL 3.23+, SQL Server 2000+), there is no performance benefit to using either syntax over the other. The optimizer sees them as the same query. But more complex queries can benefit from using ANSI-92 syntax:

- Ability to control JOIN order - the order which tables are scanned
- Ability to apply filter criteria on a table prior to joining

From a Maintenance Perspective:

There are numerous reasons to use ANSI-92 JOIN syntax over ANSI-89:

- More readable, as the JOIN criteria is separate from the WHERE clause
- Less likely to miss JOIN criteria
- Consistent syntax support for JOIN types other than INNER, making queries easy to use on other databases
- WHERE clause only serves as filtration of the cartesian product of the tables joined

From a Design Perspective:

ANSI-92 JOIN syntax is pattern, not anti-pattern:

- The purpose of the query is more obvious; the columns used by the application is clear
- It follows the modularity rule about using strict typing whenever possible. Explicit is almost universally better.

Conclusion

Short of familiarity and/or comfort, I don't see any benefit to continuing to use ANSI-89 JOIN syntax instead of ANSI-92 syntax. Some might complain that ANSI-92 syntax is more verbose, but that's what makes it explicit. The more explicit, the easier it is to understand and maintain.

edited Mar 29 '12 at 15:17
RedYeti
441 ● 5 ● 13

answered Feb 11 '10 at 3:14
OMG Ponies
157k ● 21 ● 262 ● 344

11 "...Any query involving more than one table requires some form of association to link the results from table 'A' to table 'B'... Otherwise you'll get a Cartesian Product, and you probably don't want that (those Cartesians make LOUSY products). – Scott Smith Feb 11 '10 at 3:56

1 Excellent and informative answer, +1! – Adam Arold Jan 8 '14 at 14:06

Maximize your data center
availability and capacity
Monitor, model and manage data center assets with CA DCIM.
Learn More
ca
technologies

These are the problems with using the where syntax (other wise known as the implicit join):

First, it is all too easy to get accidental cross joins because the join conditions are not right next to the table names. If you have 6 tables being joined together, it is easy to miss one in the where clause. You will see this fixed all too often by using the distinct keyword. This is ahuge performance hit for the database. You can't get an accidental cross join using the explicit join syntax as it will fail the syntax check.

Right and left joins are problematic (In SQl server you are not guaranteed to get the correct results) in the old syntax in some databases. Further they are deprecated in SQL Server I know.

If you intend to use a cross join, that is not clear from the old syntax. It is clear using the current ANSII standard.

It is much harder for the maintainer to see exactly which fields are part of the join or even which tables join together in what order using the implicit syntax. This means it might take more time to revise the queries. I have known very few people who, once they took the time to feel comfortable with the explicit join syntax, ever went back to the old way.

I've also noticed that some people who use these implicit joins don't actually understand how joins work and thus are getting incorrect results in their queries.

Honestly, would you use any other kind of code that was replaced with a better method 18 years ago?

answered Feb 11 '10 at 15:19
HLGEM
58.1k ● 4 ● 53 ● 105

You can't use WHERE to combine two tables. What you can do though is to write:

```
SELECT * FROM A, B
WHERE ...
```

The comma here is equivalent to writing:

```
SELECT *
FROM A
CROSS JOIN B
WHERE ...
```

Would you write that? No - because it's not what you mean at all. You don't want a cross join, you want an INNER JOIN. But when you write comma, you're saying CROSS JOIN and that's confusing.

answered Feb 11 '10 at 3:13
Mark Byers
349k ● 55 ● 819 ● 1014

Most people tend to find the JOIN syntax a bit clearer as to what is being joined to what. Additionally, it has the benefit of being a standard.

Personally, I "grew up" on WHEREs, but the more I use the JOIN syntax the more I'm starting to see how it's more clear.

answered Feb 11 '10 at 3:11
Tenner
3,627 ● 17 ● 34

but stil i like wheres.. – nawfal May 8 '12 at 17:33

@nawfal Would you mind saying why? – Basic Oct 24 '12 at 9:59

1 @Basic I now like JOINS. Can't explain why even now.. – nawfal Oct 24 '12 at 10:09

@nawfal I understand - I was brought up on the old-style syntax but once I got used to the new one, the benefits became clear eg harder to do a cross join by mistake, more explicit, easier to see what's a join and what's a filter, etc... I became hooked. – Basic Oct 24 '12 at 10:11 ✎

Explicit joins convey intent, leaving the where clause to do the filtering. It is cleaner and it is standard, and you can do things such as left outer or right outer which is harder to do only with where.

answered Feb 11 '10 at 3:10
Otávio Décio
46.4k ● 5 ● 111 ● 168