Inner join vs Where

Asked 16 years, 3 months ago Modified 4 years, 7 months ago Viewed 150k times

307

Is there a difference in performance (in oracle) between

```
Select * from Table1 T1
Inner Join Table2 T2 On T1.ID = T2.ID
```

And

```
Select * from Table1 T1, Table2 T2
Where T1.ID = T2.ID
```

?

sql performance oracle-database

Share

Improve this question

Follow

edited Jun 3, 2009 at 22:00

George Stocker

57.9k ● 29 ● 180 ● 238

asked Sep 23, 2008 at 15:13

juan

81.8k ● 52 ● 164 ● 198

2 just for the record, I have seen queries return different results only by changing from a join to a where clause – BlackTigerX Aug 25, 2010 at 18:55

13 @BlackTigerX: Different results? Was there any outer joins involved? Because I don't see how different results would happen between an `inner join ... on` versus putting equivalent join criteria in the `where` clause. – Shannon Severance May 7, 2011 at 4:12

in old version of oracle database not exists `join` – MajidTaheri May 24, 2012 at 15:58

2 @MajidTaheri: how old of a version of Oracle are you talking about? Any version supported by Oracle supports the JOIN notation. – Jonathan Leffler Jan 1, 2013 at 12:39

Look out for general answers or best practices based on trivial cases. Would like to see a "re-match" with more complex where clauses involving multiple AND'd conditions. At least with SQL Server there's a crossover point. – crokusek Aug 3, 2013 at 9:40

19 Answers

Sorted by: Highest score (default)

No! The same execution plan, look at these two tables:

225

```
CREATE TABLE table1 (
  id INT,
  name VARCHAR(20)
);

CREATE TABLE table2 (
  id INT,
  name VARCHAR(20)
);
```

The execution plan for the query using the inner join:

```
-- with inner join

EXPLAIN PLAN FOR
SELECT * FROM table1 t1
INNER JOIN table2 t2 ON t1.id = t2.id;

SELECT *
FROM TABLE (DBMS_XPLAN.DISPLAY);

-- 0 select statement
-- 1 hash join (access("T1"."ID"="T2"."ID"))
-- 2 table access full table1
-- 3 table access full table2
```

And the execution plan for the query using a WHERE clause.

```
-- with where clause

EXPLAIN PLAN FOR
SELECT * FROM table1 t1, table2 t2
WHERE t1.id = t2.id;

SELECT *
FROM TABLE (DBMS_XPLAN.DISPLAY);

-- 0 select statement
-- 1 hash join (access("T1"."ID"="T2"."ID"))
-- 2 table access full table1
-- 3 table access full table2
```

Share

Improve this answer

Follow

edited Jun 3, 2009 at 22:01

George Stocker

57.9k ● 29 ● 180 ● 238

answered Dec 10, 2008 at 1:25

kiewic

16.4k ● 14 ● 84 ● 104

5 I really do wish to see if there are any official documentation from Oracle saying about this – 4 Leave Cover Sep 6, 2016 at 11:31

If the query optimizer is doing its job right, there should be no difference between those queries. They are just two ways to specify the same desired result.

77

Share Improve this answer Follow

answered Sep 23, 2008 at 15:15

Craig Trader

15.7k ● 6 ● 40 ● 56

25 Yeah, performance should be the same. But the SELECT * FROM Table1, Table2 WHERE ... syntax is EVIL! – Joel Coehoorn Sep 23, 2008 at 15:31

2 I find it much easier to comprehend FOR INNER JOINS than the SQL-92 syntax. Your mileage may vary. – Craig Trader Sep 23, 2008 at 17:39

34 I find the WHERE syntax easier to read than INNER JION - I guess its like Vegemite. Most people in the world probably find it disgusting but kids brought up eating it love it. – ScottCher Oct 27, 2008 at 18:53

3 Vegemite is nasty indeed, but then again I love scrapple. Go figure. – StingyJack Aug 27, 2009 at 12:05

2 @Darryl I think the `JOIN`s are easier to read because the condition for joining the tables is defined right there immediately instead of "somewhere" in the `WHERE` clause. I prefer to reserve the `WHERE` clause for limiting the dataset (e.g. `WHERE DATE > (SYSDATE - 1)`) instead of also defining how the tables relate to each other (e.g. `WHERE T1.ID = T2.ID`). For a small table like the example in question makes little difference, but for large queries involving several tables and several conditions, I think it makes the query much easier to understand. – ImaginaryHuman072889 May 24, 2019 at 17:33

They should be exactly the same. However, as a coding practice, I would rather see the Join. It clearly articulates your intent,

76

Share Improve this answer Follow

answered Sep 23, 2008 at 15:23

Nescio

28.4k ● 10 ● 55 ● 76

12 I agree. Especially if you're joining to multiple tables, it's a lot easier to parse a select statement if you're doing explicit joins. – Paul Morie May 19, 2009 at 16:34

17 Indeed. Joins represent a semantic relationship between two sets of data but a where suggests a filtered set. +1 – EightyOne Unite Jul 19, 2011 at 9:03

35

Using `JOIN` makes the code easier to read, since it's self-explanatory.

There's no difference in speed(*I have just tested it*) and the execution plan is the same.

Share

Improve this answer

Follow

edited May 16, 2014 at 15:46

Malachi

3,221 ● 4 ● 30 ● 47

answered Sep 23, 2008 at 17:18

user21241

391 ● 2 ● 3

1 Thank you. i was looking for speed compression between this two methods.
– Farhad Navayazdan Dec 6, 2016 at 18:21

20

[For a bonus point...]

Using the JOIN syntax allows you to more easily comment out the join as its all included on one line. This *can* be useful if you are debugging a complex query

As everyone else says, they are functionally the same, however the JOIN is more clear of a statement of intent. It therefore *may* help the query optimiser either in current oracle versions in certain cases (I have no idea if it does), it *may* help the query optimiser in future versions of Oracle (no-one has any idea), or it *may* help if you change database supplier.

Share Improve this answer Follow

answered Aug 27, 2009 at 12:01

Chris Gill

2,918 ● 5 ● 26 ● 31

3 Or... easily change the INNER JOINS to LEFT JOINS, so you get see which join is causing you to miss expected rows. I do this because I do the whole query at once. If you comment out INNER JOINS, you sort of have to do a process of elimination. It takes longer. But +1 for you because this is one of my favorite reasons for INNER JOINS aside from readability!
– Matthew McPeak Aug 27, 2015 at 12:44

15

I don't know about Oracle but I know that the old syntax is being deprecated in SQL Server and will disappear eventually. Before I used that old syntax in a new query I would check what Oracle plans to do with it.

I prefer the newer syntax rather than the mixing of the join criteria with other needed where conditions. In the newer syntax it is much clearer what creates the join and what other conditions are being applied. Not really a big problem in a short query like this, but it gets much more confusing when you have a more complex query. Since

people learn on the basic queries, I would tend to prefer people learn to use the join syntax before they need it in a complex query.

And again I don't know Oracle specifically, but I know the SQL Server version of the old style left join is flawed even in SQL Server 2000 and gives inconsistent results (sometimes a left join sometimes a cross join), so it should never be used. Hopefully Oracle doesn't suffer the same issue, but certainly left and right joins can be mcuh harder to properly express in the old syntax.

Plus it has been my experience (and of course this is strictly a personal opinion, you may have differnt experience) that developers who use the ANSII standard joins tend to have a better understanding of what a join is and what it means in terms of getting data out of the database. I belive that is becasue most of the people with good database understanding tend to write more complex queries and those seem to me to be far easier to maintain using the ANSII Standard than the old style.

Share Improve this answer Follow

answered Sep 23, 2008 at 17:16

HLGEM

96.4k ● 15 ● 119 ● 189

1 Amen brother. Down with the JOINERS!! – ScottCher Oct 27, 2008 at 18:55

13

They're logically identical, but in the earlier versions of Oracle that adopted ANSI syntax there were often bugs with it in more complex cases, so you'll sometimes encounter resistance from Oracle developers when using it.

Share

Improve this answer

Follow

edited May 16, 2014 at 15:48

Malachi

3,221 ● 4 ● 30 ● 47

answered Sep 23, 2008 at 15:40

David Aldridge

52.3k ● 8 ● 72 ● 99

Earlier versions of Oracle had bugs with this? How early? What version(s)? – ScottCher Oct 27, 2008 at 18:53

Metalink has details ... they pop-up all over the place. – David Aldridge Oct 28, 2008 at 18:32

7

The performance should be identical, but I would suggest using the join-version due to improved clarity when it comes to outer joins.

Also unintentional cartesian products can be avoided using the join-version.

A third effect is an easier to read SQL with a simpler WHERE-condition.

Share Improve this answer Follow

answered Dec 4, 2008 at 20:04

stili

674 ● 6 ● 8

I really think the key is unintentional effects where the criteria is ambiguous. If you specify the type of join you know exactly what you are getting. I have found that different databases and even different versions of the same database platform will handle null values differently in an implied join. When you specify left/right inner/outer you spend the time to think about which is correct. When you use the ambiguous method, you assume that it works the way you hope/intend it to. – Steve Kallestad Jan 15, 2014 at 6:28

6

Don't forget that in Oracle, provided the join key attributes are named the same in both tables, you can also write this as:

```
select *
from Table1 inner join Table2 using (ID);
```

This also has the same query plan, of course.

Share

Improve this answer

Follow

edited Jun 3, 2009 at 23:36

juan

81.8k ● 52 ● 164 ● 198

answered Jun 3, 2009 at 21:55

cheduardo

6,619 ● 1 ● 20 ● 11

I rollbacked the edition because the previous revision changed the meaning of the answer – juan Jun 3, 2009 at 23:37

6

In a scenario where tables are in 3rd normal form, joins between tables shouldn't change. I.e. join CUSTOMERS and PAYMENTS should always remain the same.

However, we should distinguish **joins** from **filters**. Joins are about relationships and filters are about partitioning a whole.

Some authors, referring to the standard (i.e. Jim Melton; Alan R. Simon (1993). Understanding The New SQL: A Complete Guide. Morgan Kaufmann. pp. 11–12. ISBN 978-1-55860-245-8.), wrote about benefits to adopt JOIN syntax over comma-separated tables in FROM clause.

I totally agree with this point of view.

There are several ways to write SQL and achieve the same results but for many of those who do teamwork, source code legibility is an important aspect, and certainly separate how tables relate to each other from specific filters was a big leap in sense of clarifying source code.

Share

Improve this answer

Follow

edited May 14, 2020 at 22:28

answered Jul 1, 2016 at 2:12

abrittaf

537 ● 5 ● 12

Inner join on means cross join where. Comma is cross join with lower precedence than keyword joins. On vs "filter" is irrelevant for inner join. NFs are irrelevant to querying. Nothing in the standard promotes keyword joins over comma. Trivial optimizations treat on & where alike. This answer is a bunch of misconceptions. – philipxy May 13, 2020 at 3:19

1) I guess @philipxy is trying to say "In FROM clause tables separated by comma have the same meaning as tables CROSS JOINed." I agree with that. Of course, you can stick to old-syntax and that can coexist with JOIN clauses. RDBMS optimizers will perfectly understand both cases as the same (in case they are equivalent of course) and will elaborate the same plan. – abrittaf May 14, 2020 at 19:39

4

In PostgreSQL, there's definitely no difference - they both equate to the same query plan. I'm 99% sure that's also the case for Oracle.

Share Improve this answer Follow

answered Sep 23, 2008 at 15:22

Nick Johnson

101k ● 17 ● 130 ● 198

2

They're both inner joins that do the same thing, one simply uses the newer ANSI syntax.

Share Improve this answer Follow

answered Sep 23, 2008 at 15:26

Bob Gettys

1,202 ● 1 ● 9 ● 12

2

Functionally they are the same as has been said. I agree though that doing the join is better for describing exactly what you want to do. Plenty of times I've thought I knew how I wanted to query something until I started doing the joins and realized I wanted to do a different query than the original one in my head.

Share Improve this answer Follow

answered Sep 23, 2008 at 15:29

MattC
12.3k ● 10 ● 56 ● 78

1

It is true that, functionally, both queries should be processed the same way. However, experience has shown that if you are selecting from views that use the new join syntax, it is important to structure your queries using it as well. Oracle's optimizer can get confused if a view uses a "join" statement, but a query accessing the view uses the traditional method of joining in the "where" clause.

Share Improve this answer Follow

answered Sep 27, 2008 at 2:10

JoshL
11k ● 11 ● 58 ● 62

That's more a problem with views than with joins at all. – unexist Sep 29, 2008 at 15:04

0

Although the identity of two queries seems obvious sometimes some strange things happens. I have come accros the query wich has different execution plans when moving join predicate from JOIN to WHERE in Oracle 10g (for WHERE plan is better), but I can't reproduce this issue in simplified tables and data. I think it depends on my data and statistics. Optimizer is quite complex module and sometimes it behaves magically.

Thats why we can't answer to this question in general because it depends on DB internals. But we should know that answer has to be '*no differences*'.

Share Improve this answer Follow

answered Jun 27, 2014 at 8:59

greatvovan
3,107 ● 1 ● 30 ● 50

0

i had this conundrum today when inspecting one of our sp's timing out in production, changed an inner join on a table built from an xml feed to a 'where' clause instead....average exec time is now 80ms over 1000 executions, whereas before average exec was 2.2 seconds...major difference in the execution plan is the dissapearance of a key lookup... The message being you wont know until youve tested using both methods.

cheers.

Share Improve this answer Follow

answered Oct 28, 2014 at 14:15

trbullet81

1

Just a note, many years later - what you observed is more likely be the product of rebuilding the stored procedure than changing the syntax. – Leonardo Herrera Aug 27, 2020 at 14:21

0

They're both joins and where that do the same thing.

Give a look at In MySQL queries, why use join instead of where?

Share

Improve this answer

Follow

edited May 23, 2017 at 10:31

Community Bot

1 • 1

answered Mar 3, 2017 at 5:56

rxpande

13 • 6

0

As kiewik said, the execution plan is the same.

The JOIN statement is only more easy to read, making it easier not to forget the ON condition and getting a cartesian product. These errors can be quite hard to detect in long queries using multiple joins of type : SELECT * FROM t1, t2 WHERE t1.id=t2.some_field.

If you forget only one join condition, you get a very long to execute query returning too many records... really too many. Some poeple use a DISTINCT to patch the query, but it's still very long to execute.

That's accurately why, using JOIN statement is surely the best practice : a better maintainability, and a better readability.

Further more, if I well remember, JOIN is optimized concerning memory usage.

Share Improve this answer Follow

answered Aug 8, 2017 at 12:45

ROQUEFORT François
71 ● 1 ● 5

0

I have an addition to that good answer:

That's what is defined as SQL92 and SQL89 respectively, there is no performance difference between them although you can omit the word INNER (using just JOIN is clear enough and in the simplest query you save 5 keyboard strokes now imagine how many strokes there are in big ones).

Share Improve this answer Follow

answered Dec 19, 2019 at 15:18

user2188550

Highly active question. Earn 10 reputation (not counting the association bonus) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.