



Questions

Tags

Tour

Users

Ask Question

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required.

Take the 2-minute tour



## What is the difference between Left, Right, Outer and Inner Joins?



177



121

I am wondering how to differentiate all these different joins ...

sql

database

join

share | improve this question

edited Mar 24 '09 at 20:45



mwigdahl

8,030 ●2 ●20 ●41

asked Jan 15 '09 at 19:07



MrM

3,649 ●12 ●51 ●90

11 from Coding Horror [A Visual Explanation of SQL Joins](#) – [SQLMenace](#) Jan 15 '09 at 19:12

Also depends on your level of understanding, for someone like you that article does nothing but for someone who maybe doesn't fully understand joins it is pretty clear – [SQLMenace](#) Jan 15 '09 at 19:27

9 This is a classic question. – [John Nolan](#) Jan 15 '09 at 20:38

2 This is a duplicate of [stackoverflow.com/questions/419375/sql-join-differences](#) and no doubt others... – [cletus](#) Jan 15 '09 at 21:00

does anyone know a site which explains multiple nested joins and how they work further in detail? – [phill](#)

asked 5 years ago  
viewed 78956 times  
active 7 months ago

### CAREERS 2.0

[Web Developer](#)  
**Clinton Foundation**  
New York, NY

[Sr Software Engineer- .Net C# MVC](#)  
**TMP Worldwide**  
New York, NY

[Python Mentor: Help Our Students Learn Test Driven Web...](#)  
**Thinkful**  
New York, NY / remote

[Product Engineer at Online Lending startup](#)  
**Bond Street**  
New York, NY

[Senior Mobile Developer](#)

show 3 more comments

## 7 Answers

active

oldest

votes



256

**Simple Example:** Lets say you have a Students table, and a Lockers table.

Each student can be assigned to a locker, so there is a "LockerNumber" column in the student table. More than one student could potentially be in a single locker, but especially at the beginning of the school year, you may have some incoming students without lockers and some lockers that have no students assigned.

For the sake of this example, lets say you have **100 students**, 70 of which have lockers. You have a total of **50 lockers**, 40 of which have at least 1 student.

**INNER JOIN** is equivalent to *"show me all students with lockers"*.

Any students without lockers, or any lockers without students are missing.

**Returns 70 rows**

**LEFT OUTER JOIN** would be *"show me all students, with their corresponding locker if they have one"*.

This might be a general student list, or could be used to identify students with no locker.

**Returns 100 rows**

**RIGHT OUTER JOIN** would be *"show me all lockers, and the students assigned to them if there are any"*.

This could be used to identify lockers that have no students assigned, or lockers that have too many students.

**Returns 80 rows** (list of 70 students in the 40 lockers, plus the 10 lockers with no student)

**FULL OUTER JOIN** would be silly and probably not much use.

Something like *"show me all students and all lockers, and match them up where you can"*

**Returns 110 rows** (all 100 students, including those without lockers. Plus the 10 lockers with no student)

**CROSS JOIN** is also fairly silly in this scenario.

It doesn't use the linked "lockernumber" field in the students table, so you basically end up with a big giant list of every possible student-to-locker pairing, whether or not it actually exists.

**Returns 5000 rows** (100 students x 50 lockers). Could be useful (with filtering) as a starting point to match up the new students with the empty lockers.

EveryMove  
Seattle, WA

Senior Rails Engineer  
Imprint  
New York, NY

[More jobs near Newark...](#)

## Linked

- 203 [Difference between JOIN and INNER JOIN](#)
- 26 [difference between natural join and inner join](#)
- 2 [What is the difference between left joins and right joins in mysql](#)
- 1324 [Difference between inner and outer join](#)
- 247 [Questions every good .NET developer should be able to answer?](#)
- 28 [SQL Join Differences](#)
- 17 [The best tutorial on Joins in MySQL?](#)
- 10 [SQL recursive query on self referencing table \(Oracle\)](#)
- 0 [MySQL - Join two tables, different columns in table 1 on the same column in table 2](#)
- 0 [Table Merge Without](#)



BradC

17.5k ● 5 ● 40 ● 64

Dupes

[see more linked questions...](#)

- 3 Using your example, CROSS join would be useful as a starting point for creating locker assignments: start with all possible combinations and then use other criteria to filter results from the list. — [Joel Coehoorn](#) Jan 15 '09 at 20:15
- 10 I like the illustration — [MrM](#) Jan 15 '09 at 21:15
- 1 Nice answer. I believe Cross Join is most often used to generate testing data from a few rows when you need a large number of records. — [Eli](#) Jan 15 '09 at 21:24
- 1 FULL OUTER JOINS can be useful when searching for orphaned data, or when comparing different versions of the same data set. — [Lara Dougan](#) Jan 21 '09 at 7:17
- 2 Very good post. +1 — [Camilo Díaz Repka](#) Feb 26 '09 at 3:37

show 8 more comments



63

There are three basic types of join:

- **INNER** join compares two tables and only returns results where a match exists. Records from the 1st table are duplicated when they match multiple results in the 2nd. INNER joins tend to make result sets smaller, but because records can be duplicated this isn't guaranteed.
- **CROSS** join compares two tables and return every possible combination of rows from both tables. You can get a lot of results from this kind of join that might not even be meaningful, so use with caution.
- **OUTER** join compares two tables and returns data when a match is available or NULL values otherwise. Like with INNER join, this will duplicate rows in the one table when it matches multiple records in the other table. OUTER joins tend to make result sets larger, because they won't by themselves remove any records from the set. You must also qualify an OUTER join to determine when and where to add the NULL values:
  - **LEFT** means keep all records from the 1st table no matter what and insert NULL values when

## Related

- 1324 [Difference between inner and outer join](#)
- 279 [Left join and Left outer join in SQL Server](#)
- 203 [Difference between JOIN and INNER JOIN](#)
- 179 [How to join data frames in R \(inner, outer, left, right\)?](#)
- 5 [What is the difference between LEFT, RIGHT, INNER and OUTER JOINS?](#)
- 141 [What's the difference between INNER JOIN, LEFT JOIN, RIGHT JOIN and FULL JOIN](#)
- 0 [Problems with INNER JOIN and LEFT/RIGHT OUTER JOIN](#)
- 0 [what is the difference between inner and outer joins sql server](#)
- 1 [What is difference between inner join and outer join](#)
- 1 [Are INNER JOIN and](#)

the 2nd table doesn't match.

- **RIGHT** means the opposite: keep all records from the 2nd table no matter what and insert NULL values when the 1st table doesn't match.
- **FULL** means keep all records from both tables, and insert a NULL value in either table if there is no match.

Often you see will the **OUTER** keyword omitted from the syntax. Instead it will just be "LEFT JOIN", "RIGHT JOIN", or "FULL JOIN". This is done because INNER and CROSS joins have no meaning with respect to LEFT, RIGHT, or FULL, and so these are sufficient by themselves to unambiguously indicate an OUTER join.

Here is an example of when you might want to use each type:

- **INNER**: You want to return all records from the "Invoice" table, along with their corresponding "InvoiceLines". This assumes that every valid Invoice will have at least one line.
- **OUTER**: You want to return all "InvoiceLines" records for a particular Invoice, along with their corresponding "InventoryItem" records. This is a business that also sells service, such that not all InvoiceLines will have an InventoryItem.
- **CROSS**: You have a digits table with 10 rows, each holding values '0' through '9'. You want to create a date range table to join against, so that you end up with one record for each day within the range. By CROSS-joining this table with itself repeatedly you can create as many consecutive integers as you need (given you start at 10 to 1st power, each join adds 1 to the exponent). Then use the DATEADD() function to add those values to your base date for the range.

share | improve this answer

edited May 26 '09 at 20:48

community wiki

9 revs

Joel Coehoorn

---

1 excellently explained!! – [Shachi](#) Apr 10 '13 at 9:26

---


[add comment](#)



Here is a graphic view of all joins that gives clear visual explanations.


[OUTER JOIN necessary?](#)


## Hot Network Questions


 [How to parse these crazy sentences](#)


 [Are people sick of logins?](#)


 [What is Armllet toggling?](#)


 [How do I say in german "car parking fine lawyer"?](#)

 [How can you send the message that you deserve respect as the new guy in a department of experienced coworkers?](#)


 [Was the mitochondrion or chloroplast first?](#)

 [Down-to-earth expositions of Hodge theory](#)

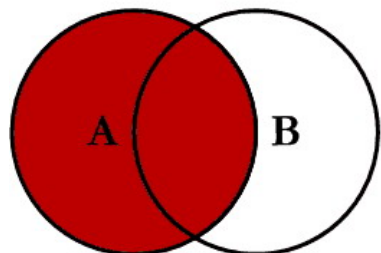
 [What does "advanced simple template" mean in the context of a creature?](#)

 [Remove a column from a nested list in Python](#)

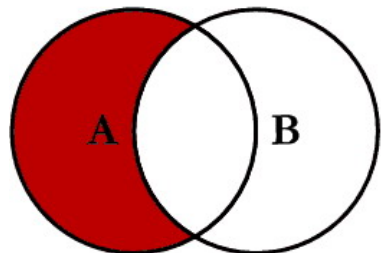
 [How can gold be traded to other players Diablo 3?](#)

 [How strong of an obligation does an advisor have to attend](#)

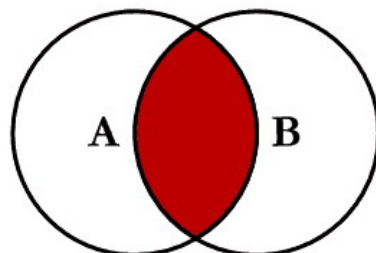
# SQL JOINS



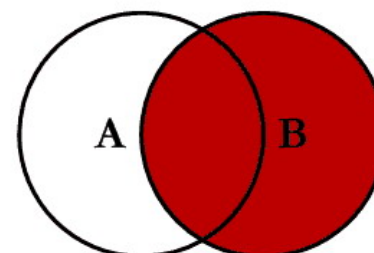
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



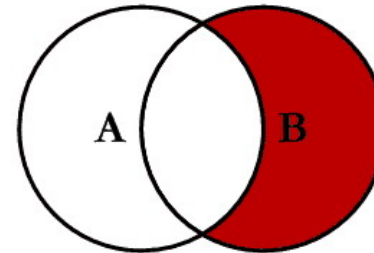
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



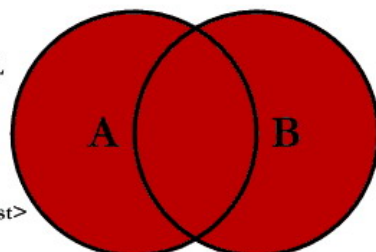
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



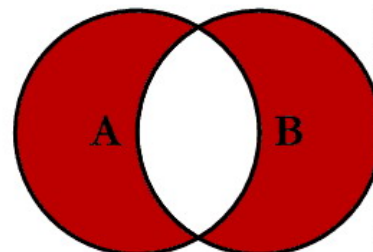
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

© C.L. Moffatt, 2008

nis/ner Ph.D. students' graduation?

ux Is red an acceptable colour for a submit button on a form for a purchase or booking?

Is it legal to modify an object created with new through a const pointer?

ask Is there any way for Ubuntu to speed up Unicoi mining?

Why are the cockpit controls of airplanes so complicated?

GD How to balance dignity with a calculated risk of offensiveness / tastelessness?

ELL "I'm a large." (Seinfeld)

Do we need waves for fields?

Can you actually "fly under the radar"?

U How do I copy multiple files by wildcard?

Is a load balancer used once per new client or all the time?

PCG Convert short month names to their longer counterparts

Which beamer theme is this? Or how to create the one like this

Why have private static methods?

and here [Visual-Representation-of-SQL-Joins](#) explained in detail by [C.L. Moffatt](#)

share | improve this answer

edited Jul 25 '13 at 17:29

answered Jul 17 '13 at 6:01



vels4j

4,425 ●1 ●9 ●18

4 Please include a link to the source of this image, for attribution to its author. – [Michael Berkowski](#) Jul 25 '13 at 15:57

1 @MichaelBerkowski Actually I got this pic from google image. Now I just googled and got the creator. Very



add comment



18



There are only 4 kinds:

1. **Inner join:** The most common type. An output row is produced for every pair of input rows that match on the join conditions.
2. **Left outer join:** The same as an inner join, except that if there is any row for which no matching row in the table on the right can be found, a row is output containing the values from the table on the left, with `NULL` for each value in the table on the right. This means that every row from the table on the left will appear at least once in the output.
3. **Right outer join:** The same as a left outer join, except with the roles of the tables reversed.
4. **Full outer join:** A combination of left and right outer joins. Every row from both tables will appear in the output at least once.

A "cross join" or "cartesian join" is simply an inner join for which no join conditions have been specified, resulting in all pairs of rows being output.

Thanks to RusselH for pointing out FULL joins, which I'd omitted.

share | improve this answer

edited Jan 16 '09 at 19:15

answered Jan 15 '09 at 19:13



[j\\_random\\_hacker](#)

26.9k ● 3 ● 39 ● 91

Also FULL and CROSS – [Joel Coehoorn](#) Jan 15 '09 at 19:15

1 what about full outer join and cross join (Cartesian product)? – [SQLMenace](#) Jan 15 '09 at 19:15

full is really the equivalent of two outer joins – [RussellH](#) Jan 15 '09 at 19:16

12 FULL is what you get when you screw up your inner join, and then you ask a question here "why am I getting  $N^2$  rows instead of  $N$ "? Then everybody gets CROSS at you. – [Paul Tomblin](#) Jan 15 '09 at 19:19

lol @ Paul... :) – [j\\_random\\_hacker](#) Jan 16 '09 at 15:05

add comment



Have a look at the tutorials at [www.w3schools.com](http://www.w3schools.com). They have a lot of useful tutorials about SQL (and

12

other technologies as well), including tutorials and examples about the different types of JOINS.

share | improve this answer

answered Jan 15 '09 at 19:10



M4N

46.4k ● 15 ● 118 ● 184

How is Martin's post against the rules? – [Paul Tomblin](#) Jan 15 '09 at 19:16

@M4N www.w3schools.com is not a detailed tutorial. It only help students for exam – [HYDER ALI](#) Nov 9 '13 at 11:58

add comment



3



Check out [Join \(SQL\) on Wikipedia](#)

- Inner join - Given two tables an inner join returns all rows that exist in both tables
- left / right (outer) join - Given two tables returns all rows that exist in either the left or right table of your join, plus the rows from the other side will be returned when the join clause is a match or null will be returned for those columns
- Full Outer - Given two tables returns all rows, and will return nulls when either the left or right column is not there
- Cross Joins - Cartesian join and can be dangerous if not used carefully

share | improve this answer

edited Jan 15 '09 at 19:19

answered Jan 15 '09 at 19:15



rob

8,680 ● 6 ● 46 ● 86



JoshBerke

32.5k ● 7 ● 71 ● 125

add comment



2



LEFT and RIGHT are both types of outer joins. Inner join is the default -- rows from both tables must match the join condition.

share | improve this answer

answered Jan 15 '09 at 19:14



RussellH

696 ● 5 ● 12

add comment

protected by [hjpotter92](#) Oct 18 '13 at 15:04

This question is protected to prevent "thanks!", "me too!", or spam answers by new users. To answer it, you must have earned at least 10 [reputation](#) on this site.

Not the answer you're looking for? Browse other questions tagged [sql](#) [database](#) [join](#) or [ask your own question](#).

 question feed

[about](#) [help](#) [badges](#) [blog](#) [chat](#) [data](#) [legal](#) [privacy policy](#) [jobs](#) [advertising info](#) [mobile](#) [contact us](#) [feedback](#)

TECHNOLOGY

Stack Overflow  
Server Fault  
Super User  
Web Applications  
Ask Ubuntu  
Webmasters  
Game Development  
TeX - LaTeX  
Programmers  
Unix & Linux  
Ask Different (Apple)  
WordPress Development  
Geographic Information Systems  
Electrical Engineering  
Android Enthusiasts  
Information Security

Database Administrators  
Drupal Answers  
SharePoint  
User Experience  
Mathematica  
**more (14)**

LIFE / ARTS

Photography  
Science Fiction & Fantasy  
Seasoned Advice (cooking)  
Home Improvement  
Personal Finance & Money  
**more (12)**

CULTURE / RECREATION

English Language & Usage  
Skeptics  
Mi Yodeya (Judaism)  
Travel  
Christianity  
Arcade (gaming)  
Bicycles  
Role-playing Games  
**more (21)**

SCIENCE

Mathematics  
Cross Validated (stats)  
Theoretical Computer Science  
Physics  
MathOverflow  
**more (7)**

OTHER

Stack Apps  
Meta Stack Overflow  
Area 51  
Stack Overflow Careers