diffen.com

Inner Join vs Outer Join

3-4 minutes

Inner Join

An inner join focuses on the commonality between two tables. When using an inner join, there must be at least some matching data between two (or more) tables that are being compared. An inner join searches tables for matching or overlapping data. Upon finding it, the inner join combines and returns the information into one new table.

Example of Inner Join

Let's consider a common scenario of two tables: product prices and quantities. The common information in the two tables is product name, so that is the logical column to join the tables ***on***. There are some products that are common in the two tables; others are unique to one of the tables and don't have a match in the other table.

An inner join on *Products* returns information about only those products that are common in both tables.

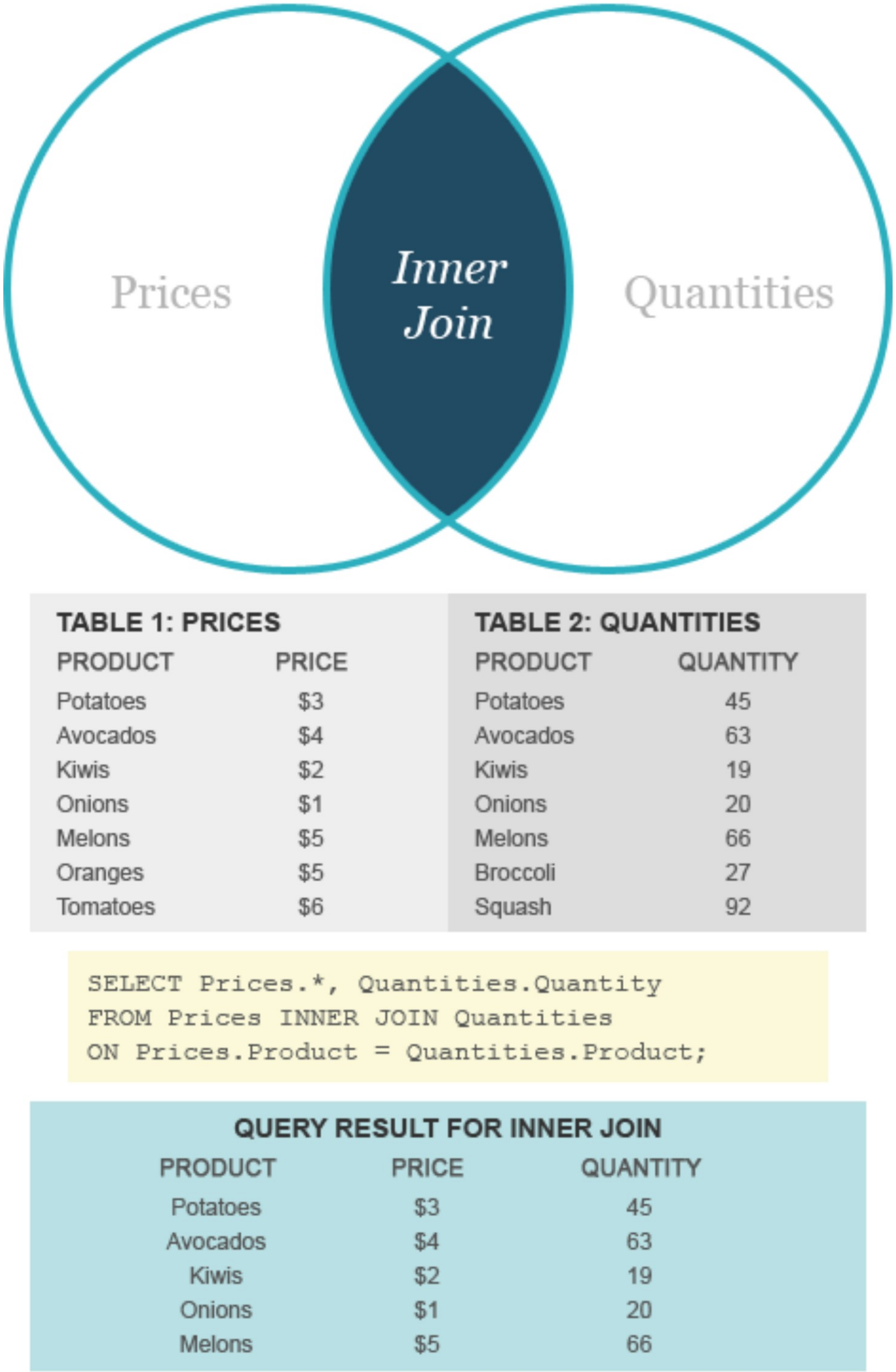


TABLE 1: PRICES

PRODUCT	PRICE
Potatoes	$3
Avocados	$4
Kiwis	$2
Onions	$1
Melons	$5
Oranges	$5
Tomatoes	$6

TABLE 2: QUANTITIES

PRODUCT	QUANTITY
Potatoes	45
Avocados	63
Kiwis	19
Onions	20
Melons	66
Broccoli	27
Squash	92

```
SELECT Prices.*, Quantities.Quantity
FROM Prices INNER JOIN Quantities
ON Prices.Product = Quantities.Product;
```

QUERY RESULT FOR INNER JOIN

PRODUCT	PRICE	QUANTITY
Potatoes	$3	45
Avocados	$4	63
Kiwis	$2	19
Onions	$1	20
Melons	$5	66

Outer Join

An outer join returns a set of records (or rows) that include what an inner join would return but also includes other rows for which no corresponding match is found in the other table.

There are three types of outer joins:

- Left Outer Join (or Left Join)
- Right Outer Join (or Right Join)
- Full Outer Join (or Full Join)

Each of these outer joins refers to the part of the data that is being compared, combined, and returned. Sometimes nulls will be produced in this process as some data is shared while other data is not.

Left Outer Join

A left outer join will return all the data in Table 1 and all the shared data (so, the inner part of the Venn diagram example), but only corresponding data from Table 2, which is the right join.

Left Join Example

In our example database, there are two products — oranges and tomatoes — on the 'left' (*Prices* table) that do not have a corresponding entry on the 'right' (Quantities table). In a left join, these rows are included in the result set with a NULL in the Quantity column. The other rows in the result are the same as the inner join.

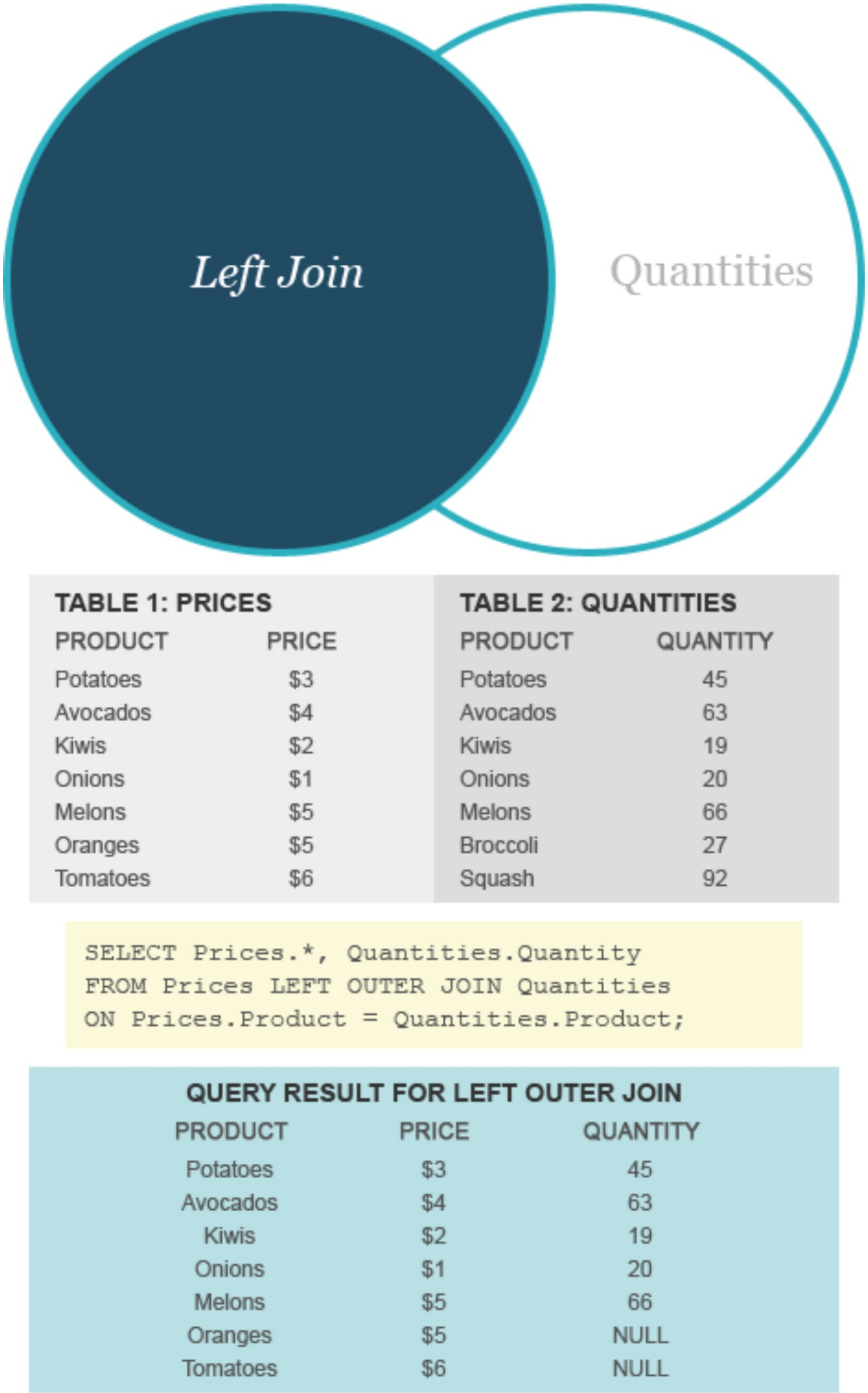


TABLE 1: PRICES

PRODUCT	PRICE
Potatoes	$3
Avocados	$4
Kiwis	$2
Onions	$1
Melons	$5
Oranges	$5
Tomatoes	$6

TABLE 2: QUANTITIES

PRODUCT	QUANTITY
Potatoes	45
Avocados	63
Kiwis	19
Onions	20
Melons	66
Broccoli	27
Squash	92

```
SELECT Prices.*, Quantities.Quantity
FROM Prices LEFT OUTER JOIN Quantities
ON Prices.Product = Quantities.Product;
```

QUERY RESULT FOR LEFT OUTER JOIN

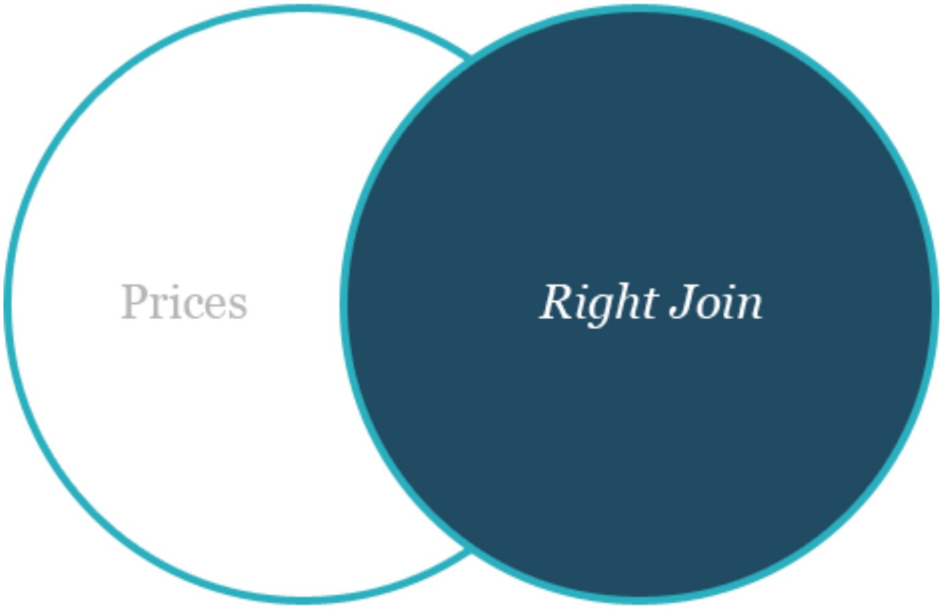
PRODUCT	PRICE	QUANTITY
Potatoes	$3	45
Avocados	$4	63
Kiwis	$2	19
Onions	$1	20
Melons	$5	66
Oranges	$5	NULL
Tomatoes	$6	NULL

Right Outer Join

A right outer join returns Table 2's data and all the shared data, but only corresponding data from Table 1, which is the left join.

Right Join Example

Similar to the left join example, the output of a right outer join includes all rows of the inner join and two rows — broccoli and squash — from the 'right' (*Quantities* table) that do not have matching entries on the left.



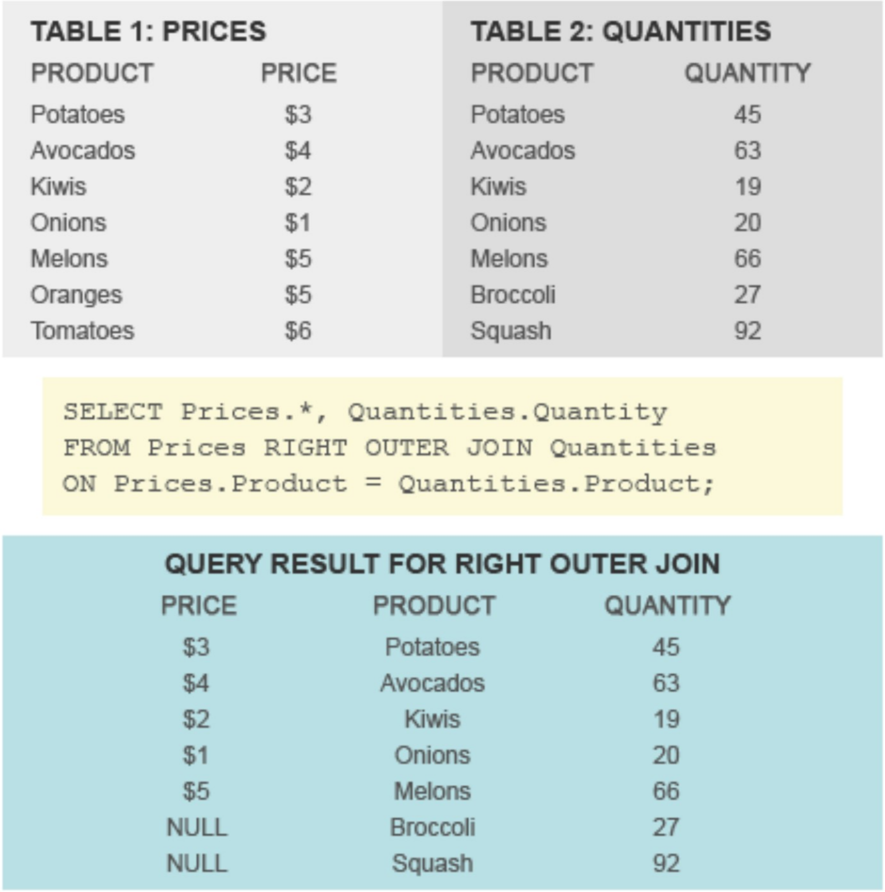


TABLE 1: PRICES	
PRODUCT	PRICE
Potatoes	$3
Avocados	$4
Kiwis	$2
Onions	$1
Melons	$5
Oranges	$5
Tomatoes	$6

TABLE 2: QUANTITIES	
PRODUCT	QUANTITY
Potatoes	45
Avocados	63
Kiwis	19
Onions	20
Melons	66
Broccoli	27
Squash	92

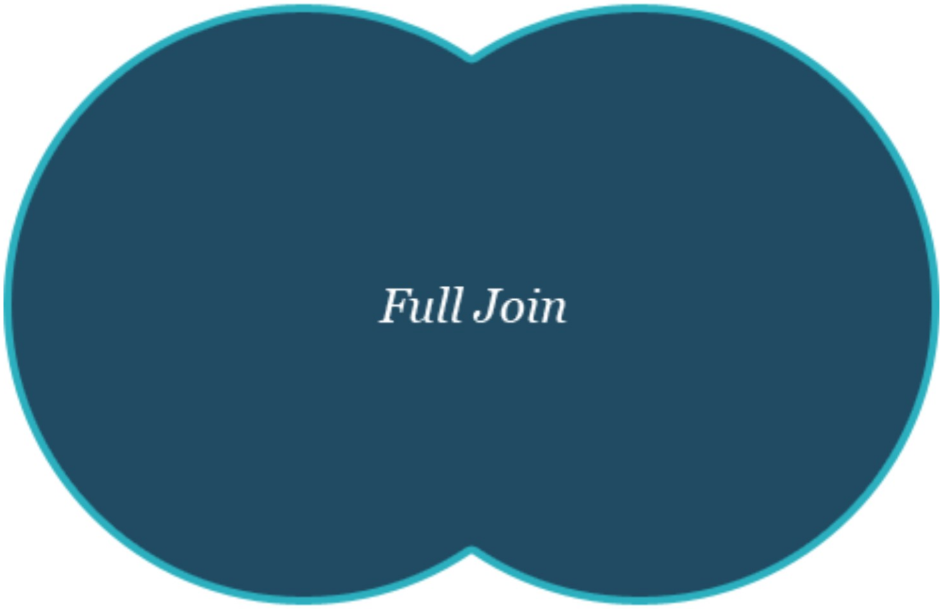
```
SELECT Prices.*, Quantities.Quantity
FROM Prices RIGHT OUTER JOIN Quantities
ON Prices.Product = Quantities.Product;
```

QUERY RESULT FOR RIGHT OUTER JOIN

PRICE	PRODUCT	QUANTITY
$3	Potatoes	45
$4	Avocados	63
$2	Kiwis	19
$1	Onions	20
$5	Melons	66
NULL	Broccoli	27
NULL	Squash	92

Full Outer Join

A full outer join, or full join, which is *not* supported by the popular MySQL database management system, combines and returns *all* data from two or more tables, regardless of whether there is shared information. Think of a full join as simply duplicating all the specified information, but in one table, rather than multiple tables. Where matching data is missing, nulls will be produced.



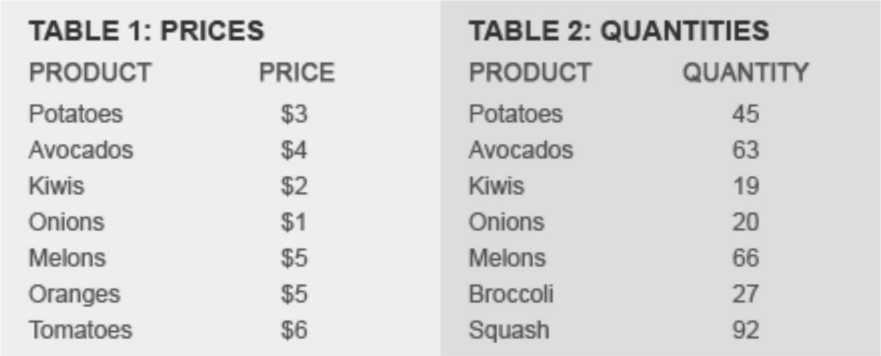
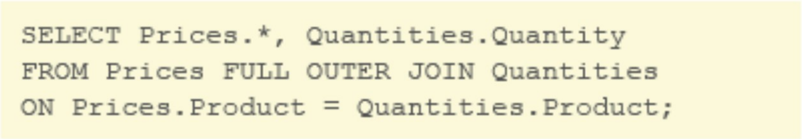
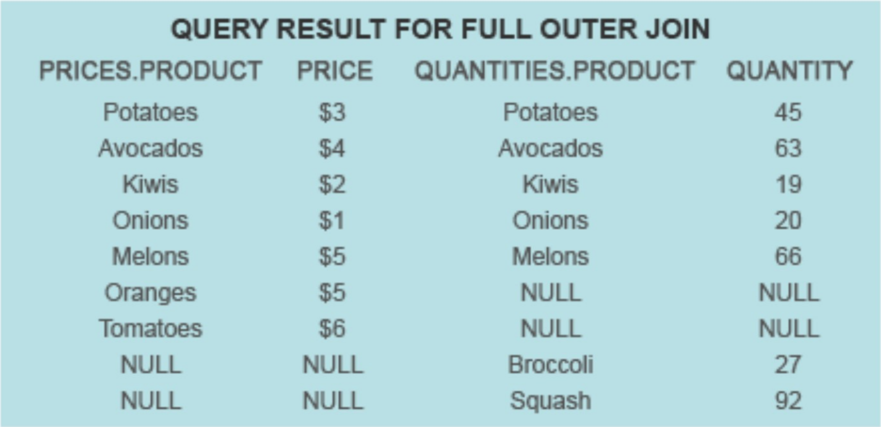


TABLE 1: PRICES		TABLE 2: QUANTITIES	
PRODUCT	PRICE	PRODUCT	QUANTITY
Potatoes	$3	Potatoes	45
Avocados	$4	Avocados	63
Kiwis	$2	Kiwis	19
Onions	$1	Onions	20
Melons	$5	Melons	66
Oranges	$5	Broccoli	27
Tomatoes	$6	Squash	92



```
SELECT Prices.*, Quantities.Quantity
FROM Prices FULL OUTER JOIN Quantities
ON Prices.Product = Quantities.Product;
```



QUERY RESULT FOR FULL OUTER JOIN

PRICES.PRODUCT	PRICE	QUANTITIES.PRODUCT	QUANTITY
Potatoes	$3	Potatoes	45
Avocados	$4	Avocados	63
Kiwis	$2	Kiwis	19
Onions	$1	Onions	20
Melons	$5	Melons	66
Oranges	$5	NULL	NULL
Tomatoes	$6	NULL	NULL
NULL	NULL	Broccoli	27
NULL	NULL	Squash	92

These are just the basics, but many things can be done with joins. There are even joins that can exclude other joins!

Video Explaining Inner vs Outer Joins

This video explains the difference between various types of joins. It is cued up to begin at the point where the discussion about joins begins.

References

- Difference between inner and outer join - *Stack Overflow*
- SQL Inner Join - *Quackit*
- SQL Outer Join - *Quackit*
- Using Inner Joins - *Microsoft SQL Server*
- Using Outer Joins - *Microsoft SQL Server*
- Visual Representation of SQL Joins - *CodeProject*
- Wikipedia: Join (SQL)