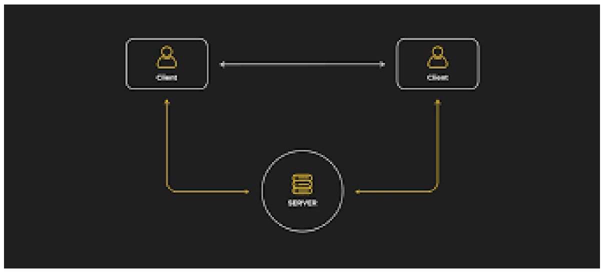
PRO-C232

JOIN STATEMENTS



What is our GOAL for this CLASS?

In this class, we learnt about the JOIN Statements and also explored how different JOINS like LEFT, INNER, FULL and RIGHT Join works.

What did we ACHIEVE in the class TODAY?

- Understanding about the JOIN statements in SQL
- Building queries to join multiple tables
- Learning about RIGHT, LEFT, INNER and FULL OUTER Joins.
- Solving real world problems with JOIN statements and constructing new tables.

Which CONCEPTS/ CODING BLOCKS did we cover today?

- SQL syntax
- SQL queries
- Joining 2 tables by columns in SQL.

WhiteHat Jr + WhiteHat Jr + WhiteHat Jr

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

How did we DO the activities?

Activity:

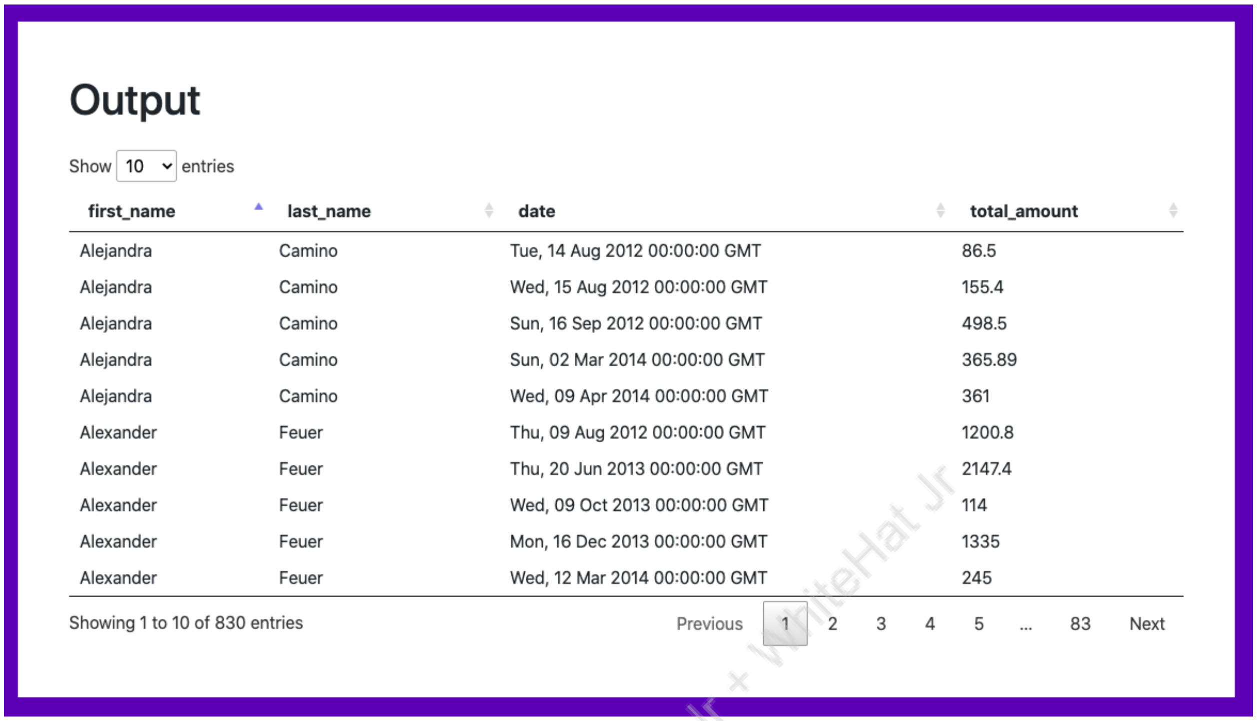
1. JOIN statements are usually used to join 2 or more tables together. It joins the table through columns. This means that it can be used to take a few columns from one table and a few columns from the other table and view it together. It joins based on a relation or a common value both the tables share, just to know which row gets joined with which.
2. There are 4 types of JOIN Statements -
   a. **Inner Join -** The inner join keyword selects records that have matching values in both the tables.
   b. **Full Join -** This keyword returns all the records that match in the left table (table 1) and the right table (table 2). Full Join and Full Outer Join are both the same.
   c. **Left Join -** A left join returns all records from the left table (table 1) and matching records from the right table (table 2). If there is no match, the right side will show 0 records.
   d. **Right Join -** This keyword returns all records from the right table (table 2) and all matching records from the left table (table 1). If there is no match, 0 records will appear from the left side.
3. We understand that our database in the editor here has 5 tables -
   a. Customers
   b. Suppliers
   c. Company products with a relation to Suppliers
   d. Company orders with a relation to Customers
   e. Order items with a relation to both company products and company orders.
4. We try to build a statement that returns a table with the following columns -
   a. Customer's name from the ***customers*** table
   b. Total amount and Date from the ***company_orders*** table
5. We construct the following statement to fetch the result -

SELECT customers.first_name, customers.last_name, company_orders.date, company_orders.total_amount **FROM** customers **INNER JOIN** company_orders **ON** customers.id=company_orders.customer_id;

Output -

WhiteHat Jr - WhiteHat Jr - WhiteHat Jr

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.



Output

Show 10 entries

first_name	last_name	date	total_amount
Alejandra	Camino	Tue, 14 Aug 2012 00:00:00 GMT	86.5
Alejandra	Camino	Wed, 15 Aug 2012 00:00:00 GMT	155.4
Alejandra	Camino	Sun, 16 Sep 2012 00:00:00 GMT	498.5
Alejandra	Camino	Sun, 02 Mar 2014 00:00:00 GMT	365.89
Alejandra	Camino	Wed, 09 Apr 2014 00:00:00 GMT	361
Alexander	Feuer	Thu, 09 Aug 2012 00:00:00 GMT	1200.8
Alexander	Feuer	Thu, 20 Jun 2013 00:00:00 GMT	2147.4
Alexander	Feuer	Wed, 09 Oct 2013 00:00:00 GMT	114
Alexander	Feuer	Mon, 16 Dec 2013 00:00:00 GMT	1335
Alexander	Feuer	Wed, 12 Mar 2014 00:00:00 GMT	245

Showing 1 to 10 of 830 entries

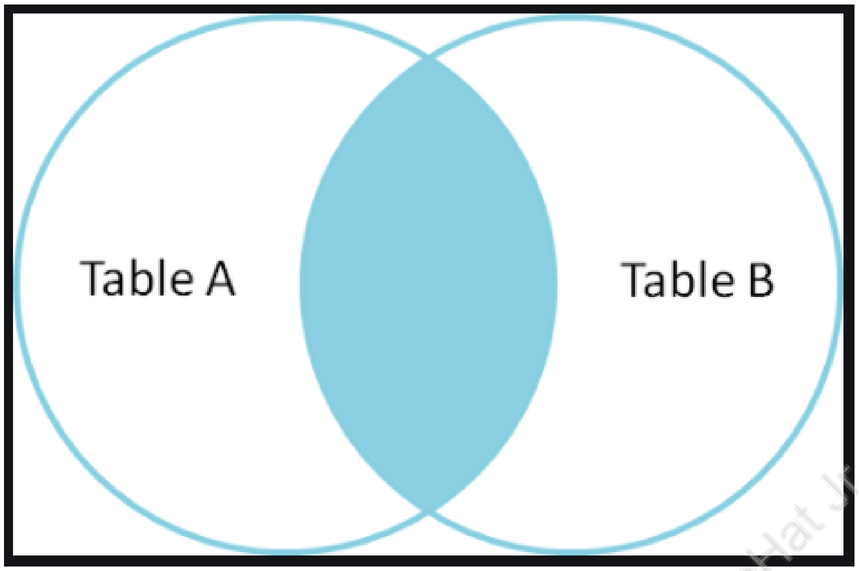
6. We understand the the structure of the query is as follows -

   SELECT table1.column1, table1.column2, table2.column1, table2.column2… **FROM** table1 **INNER JOIN** table2 **ON** table1.matching_column=table2.matching_column;

7. We can understand the query is constructed in the following way -
   a. We are using the ***select*** statement in a different way here, with ***table.column*** to syntax to define the columns that we need.
   b. We have the ***from*** keyword to tell the query which table to run the query on.
   c. We then use the ***INNER JOIN*** to specify the second table we want to join the first table with. Do note that the first table here is the table on the left and the second table here is the table on the right.
   d. We could have also used ***LEFT JOIN, RIGHT JOIN or FULL JOIN*** too here.
   e. We then use the **ON** keyword which means **"ON** what relation do you want to join the 2 tables?**"**
   f. We know that the ***customer_id*** field in the ***company_orders*** would be the same as the ***id*** column of the customer since it's a relation.

8. The ***INNER JOIN*** in context with the query we have executed above means to only join those rows which satisfy ***customer.id=company_orders.customer_id*** statement.
9. The INNER JOIN Looks something like this -

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

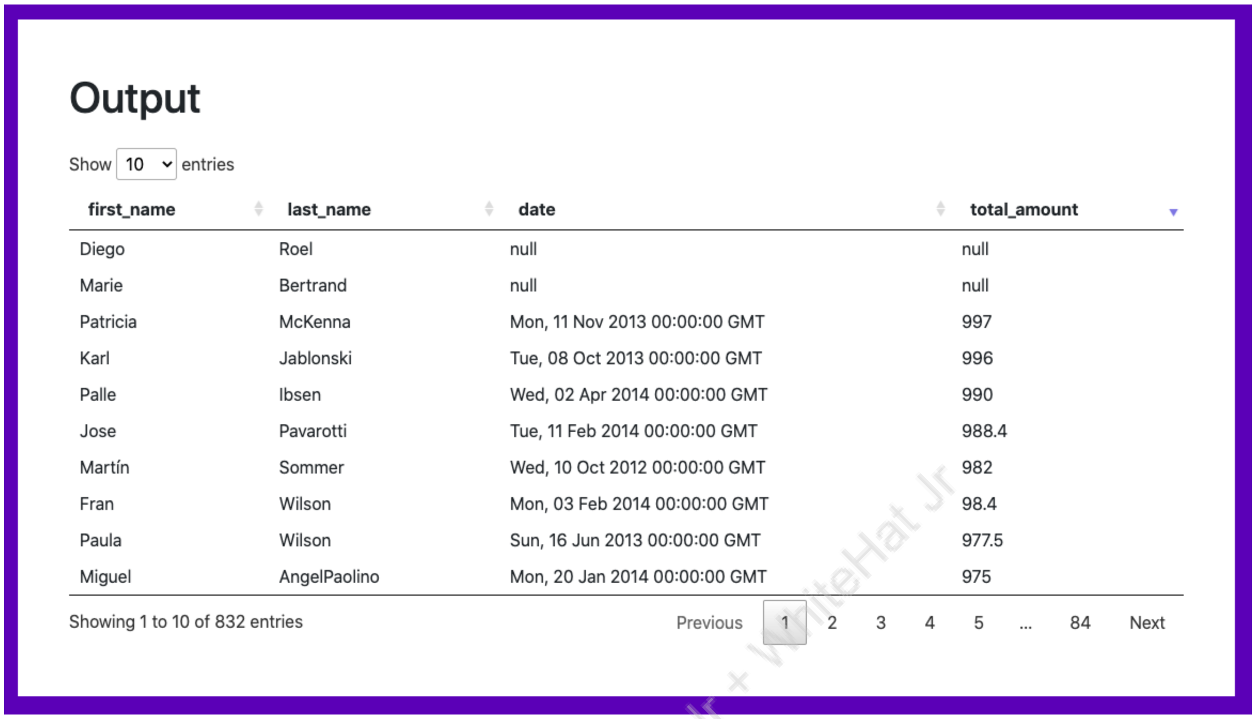


10. The above statement gave us the result of only those customers that had ordered something since we applied an inner join. If we wanted to fetch the data of those users as well, who have not ordered anything yet, then we could simply use the LEFT JOIN or the RIGHT JOIN.
    - **a.** **LEFT JOIN** when the **customers** table is Table 1 on the left side since it will then take all the rows of the left table.
    - **b.** **RIGHT JOIN** when the **customers** table is Table 2 on the right side since it will then take all the rows from the right table.
    - c. The statement with Left Join to fetch all the customers and order data for those who have ordered something would be as follows -

SELECT customers.first_name, customers.last_name, company_orders.date, company_orders.total_amount **FROM** customers **LEFT JOIN** company_orders **ON** customers.id=company_orders.customer_id;

Output -

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.



Output

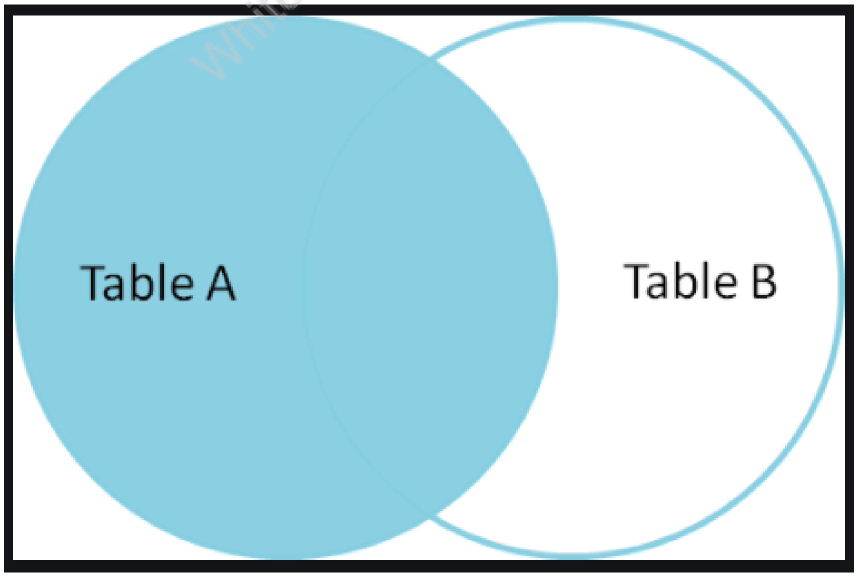
Show 10 entries

first_name	last_name	date	total_amount
Diego	Roel	null	null
Marie	Bertrand	null	null
Patricia	McKenna	Mon, 11 Nov 2013 00:00:00 GMT	997
Karl	Jablonski	Tue, 08 Oct 2013 00:00:00 GMT	996
Palle	Ibsen	Wed, 02 Apr 2014 00:00:00 GMT	990
Jose	Pavarotti	Tue, 11 Feb 2014 00:00:00 GMT	988.4
Martín	Sommer	Wed, 10 Oct 2012 00:00:00 GMT	982
Fran	Wilson	Mon, 03 Feb 2014 00:00:00 GMT	98.4
Paula	Wilson	Sun, 16 Jun 2013 00:00:00 GMT	977.5
Miguel	AngelPaolino	Mon, 20 Jan 2014 00:00:00 GMT	975

Showing 1 to 10 of 832 entries

Previous 1 2 3 4 5 ... 84 Next

11. Here, the ***null*** values mean that the user has not ordered something, but since we are using the LEFT JOIN statement with customers as Table 1 on the left side, it is displaying all the customers regardless of if they have ordered something or not whereas earlier it was only displaying those customers who have ordered something. The diagram for the ***LEFT JOIN*** looks something like this -



© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

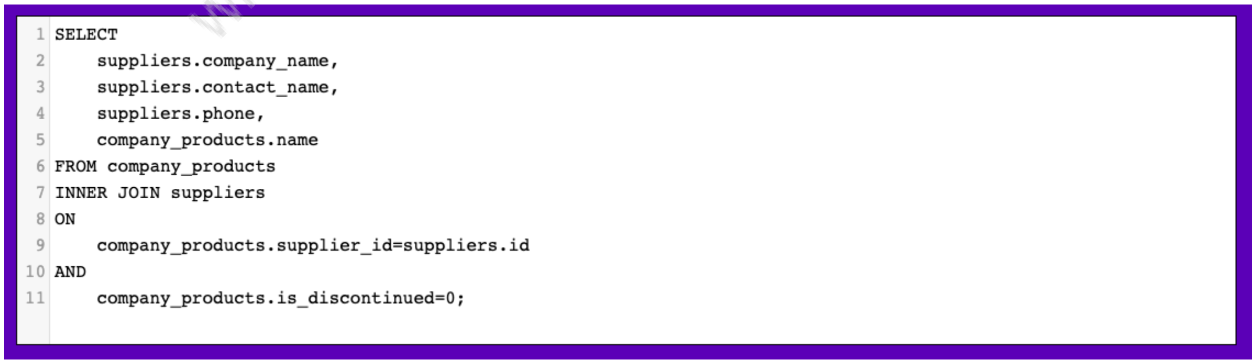
12. We try to construct a query where we can get -
    a. Name of the supplier company
    b. Their contact's name and phone number
    c. The name of the product
    d. Also check if the product is discontinued or not and only display those that are not discontinued

13. From the ***suppliers*** table, we can get the following values -
    a. Company Name
    b. Contact Name
    c. Phone Number

14. From the ***company_products*** table, we can get the following values -
    a. Name of the Product
    b. If it's discontinued or not
    c. The relation ***supplied_id*** which is common in both ***suppliers and company_products.***
15. The following query gets us the desired result -

SELECT suppliers.company_name, suppliers.contact_name, suppliers.phone , company_products.name **FROM** company_products **INNER JOIN** suppliers **ON** company_products.supplier_id=suppliers.id **AND** company_products.is_discontinued=0;

Input -

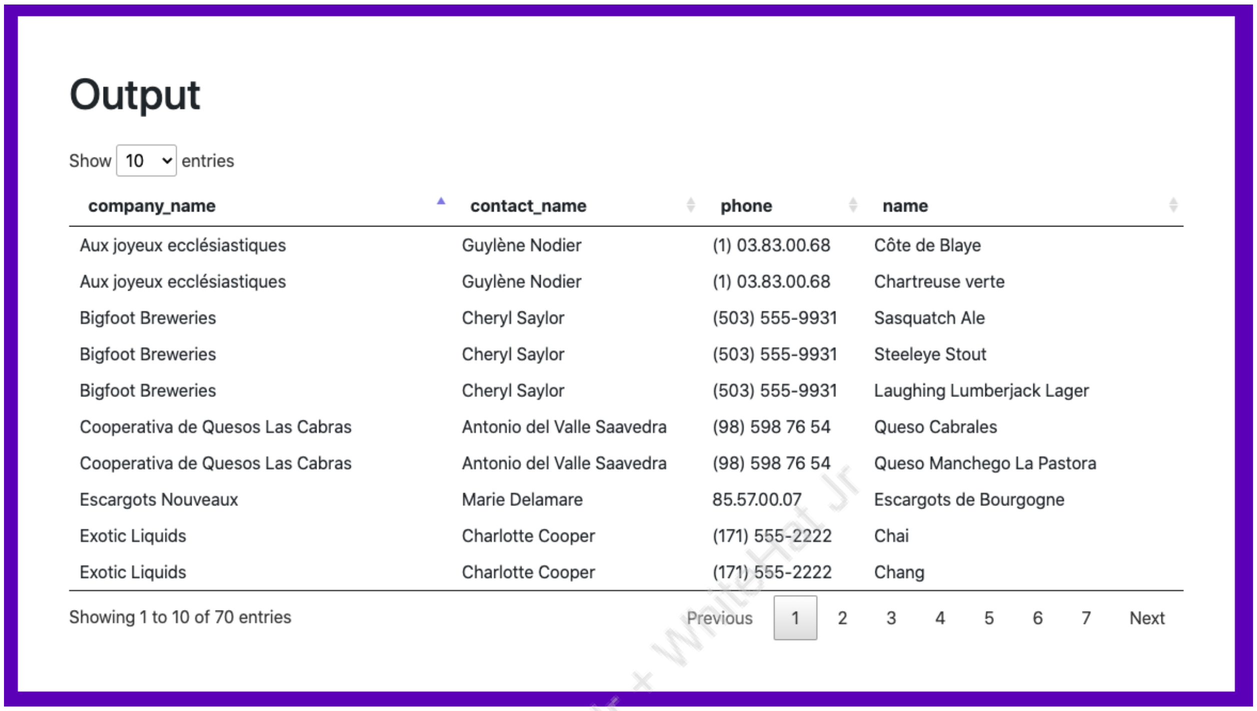


```
SELECT
    suppliers.company_name,
    suppliers.contact_name,
    suppliers.phone,
    company_products.name
FROM company_products
INNER JOIN suppliers
ON
    company_products.supplier_id=suppliers.id
AND
    company_products.is_discontinued=0;
```

Output -

WhiteHat Jr + WhiteHat Jr + WhiteHat Jr

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.



Output

Show 10 entries

company_name	contact_name	phone	name
Aux joyeux ecclésiastiques	Guylène Nodier	(1) 03.83.00.68	Côte de Blaye
Aux joyeux ecclésiastiques	Guylène Nodier	(1) 03.83.00.68	Chartreuse verte
Bigfoot Breweries	Cheryl Saylor	(503) 555-9931	Sasquatch Ale
Bigfoot Breweries	Cheryl Saylor	(503) 555-9931	Steeleye Stout
Bigfoot Breweries	Cheryl Saylor	(503) 555-9931	Laughing Lumberjack Lager
Cooperativa de Quesos Las Cabras	Antonio del Valle Saavedra	(98) 598 76 54	Queso Cabrales
Cooperativa de Quesos Las Cabras	Antonio del Valle Saavedra	(98) 598 76 54	Queso Manchego La Pastora
Escargots Nouveaux	Marie Delamare	85.57.00.07	Escargots de Bourgogne
Exotic Liquids	Charlotte Cooper	(171) 555-2222	Chai
Exotic Liquids	Charlotte Cooper	(171) 555-2222	Chang

Showing 1 to 10 of 70 entries

16. We tried the following query with all of ***INNER, OUTER, LEFT and RIGHT JOIN*** -
    a. We got 70 rows for INNER JOIN
    b. We got 78 rows for LEFT JOIN
    c. We got 71 rows for RIGHT JOIN
    d. We got 79 rows for FULL JOIN
17. The cases were as follows -

    a. INNER JOIN - In this case, it only displays those products that have a supplier and are not discontinued.
    b. LEFT JOIN - In this case, it displays all the data from the Inner Join, plus some extra rows for the products that either do not have a supplier or are discontinued.
    c. RIGHT JOIN - In this case, it displays all the data from the Inner Join, plus some extra rows from the suppliers that do not supply any products anymore.
    d. FULL JOIN - In this case, it displays all the rows from both the tables.

What's NEXT?

In the next class, we will learn about the SET Operators in SQL.

Expand Your Knowledge:

Explore more about SQL v/s NoSQL databases here.

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.