SQL LEFT JOIN

Last update on November 09 2019 06:55:13 (UTC/GMT +8 hours)

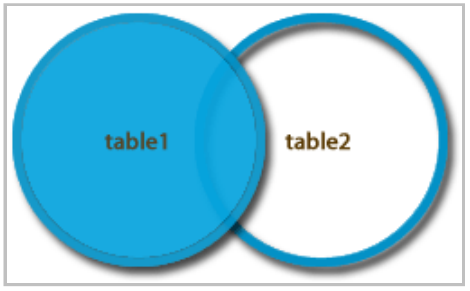
LEFT JOIN

The SQL LEFT JOIN (specified with the keywords LEFT JOIN and ON) joins two tables and fetches all matching rows of two tables for which the SQL-expression is true, plus rows from the frist table that do not match any row in the second table.

Left Join: Syntax

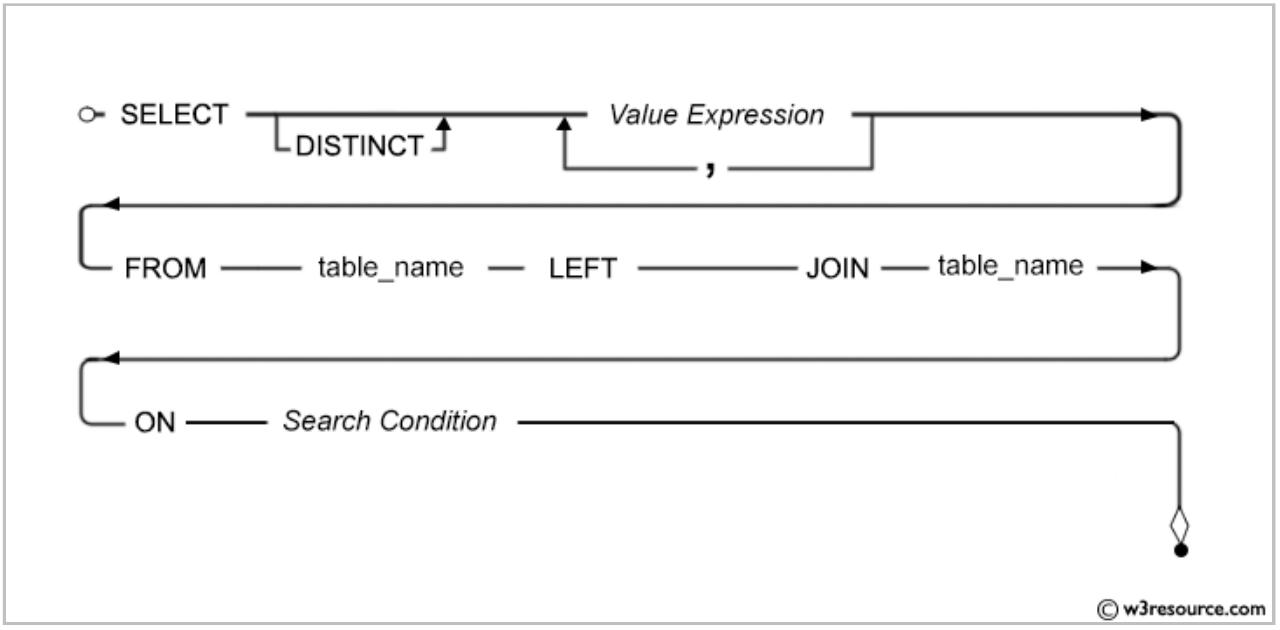
```
SELECT *
FROM table1
LEFT [ OUTER ] JOIN table2
ON table1.column_name=table2.column_name;
```

Pictorial representation:



SQL LEFT join fetches a complete set of records from table1, with the matching records (depending on the availability) in table2. The result is NULL in the right side when no matching will take place.

Syntax diagram - LEFT JOIN



Example of SQL Left Join

To get company name and company id columns from company table and company id, item name, item unit columns from foods table, after an OUTER JOINING with these mentioned tables, the following SQL statement can be used :

Sample table: foods

Sample table: company

SQL Code:

```
SELECT company.company_id,company.company_name,
company.company_city,foods.company_id,foods.item_name
FROM   company
LEFT JOIN foods
ON company.company_id = foods.company_id;
```

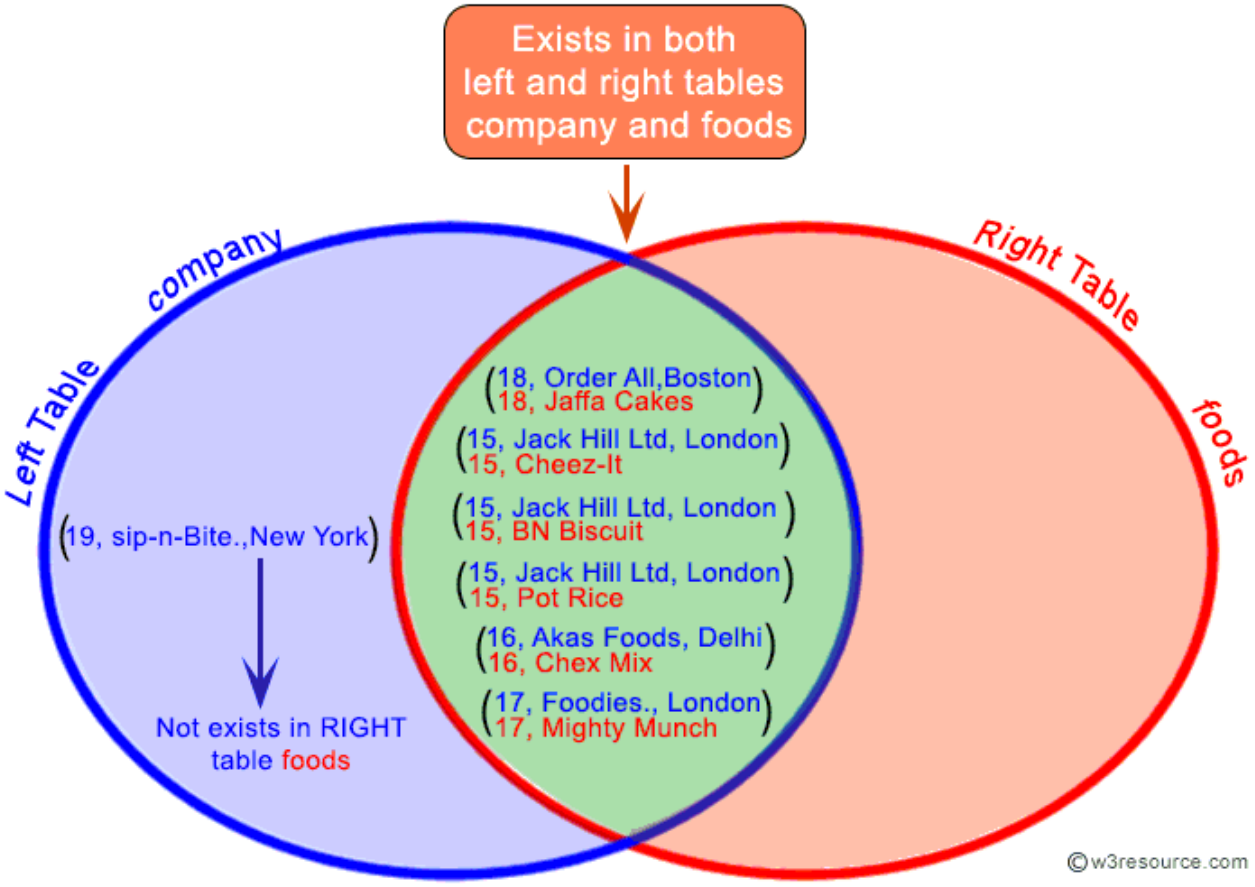
Explanation:

This SQL statement would return all rows from the company table and only those rows from the foods table where the joined fields are equal and if the ON clause matches no records in the 'foods' table, the join will still return rows, but the NULL in each column of the right table.

Output:

COMPANY_ID	COMPANY_NAME	COMPANY_CITY	COMPANY_ID	ITEM
16	Akas Foods	Delhi	16	Chex
15	Jack Hill Ltd	London	15	Chee
15	Jack Hill Ltd	London	15	BN B
17	Foodies.	London	17	Migh
15	Jack Hill Ltd	London	15	Pot
18	Order All	Boston	18	Jaff
19	sip-n-Bite.	New York		

Pictorial Presentation of the above example:



Example of SQL Left Join using multiple columns

To filtered out those bill number, item name and the bill amount for each bill which bill amount exceeds the value 500 and must be available at the food stall, the following SQL statement can be used :

Sample table: foods

Sample table: counter_sale

SQL Code:

```
SELECT a.bill_no, b.item_name, a.bill_amt
FROM counter_sale a
LEFT JOIN foods b
ON a.item_id=b.item_id
WHERE  a.bill_amt>500;
```

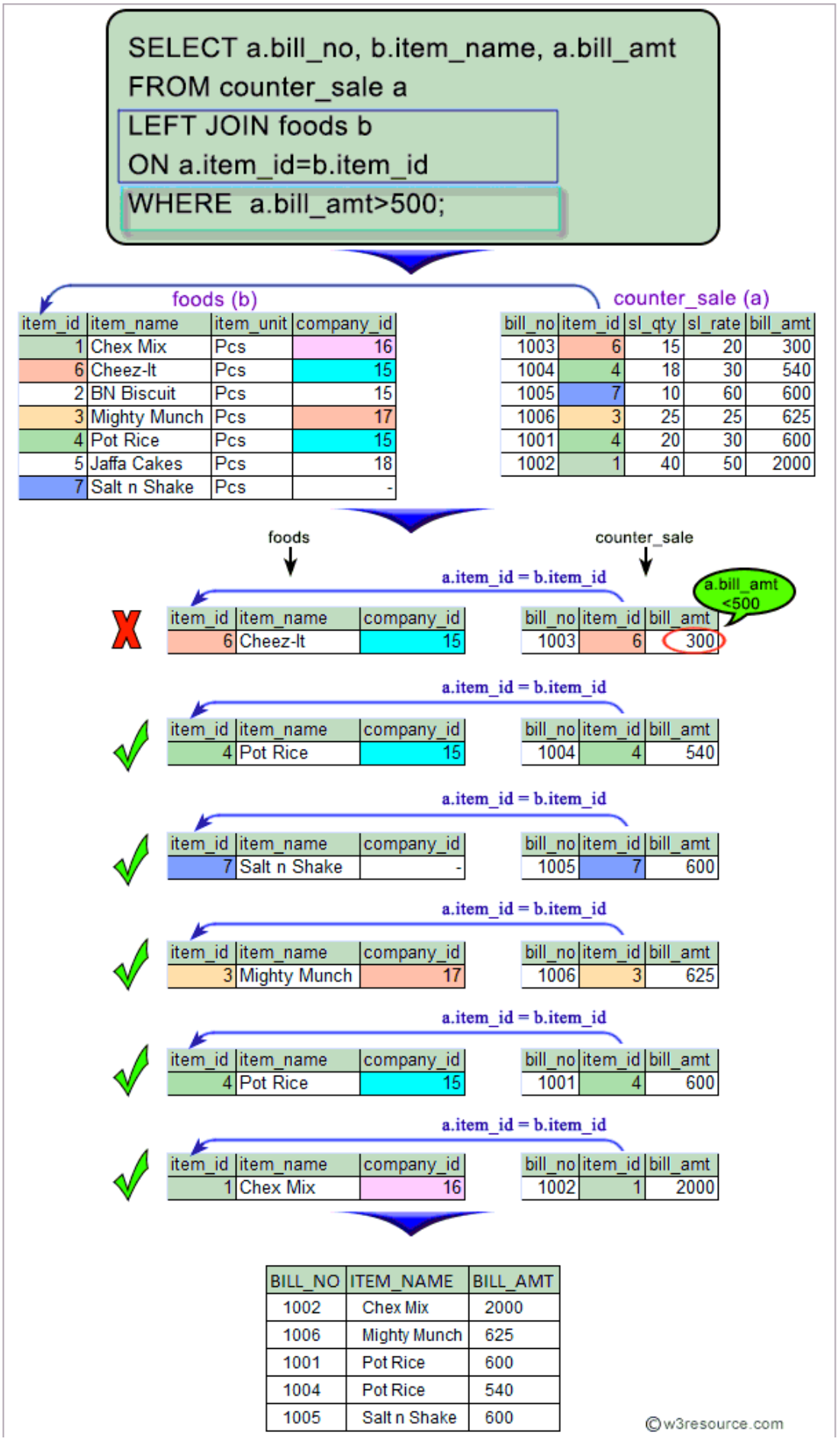
Explanation:

This SQL statement will first join all rows from the counter_sale table and only those rows from the foods table where the joined fields are equal and if the ON clause matches no records in the foods table, the join will still return rows, but the NULL in each column of right table, therefore eliminates those rows which bill amount is less than or equal to 500.

Output:

```
   BILL_NO ITEM_NAME                  BILL_AMT
---------- ------------------------- ----------
      1002 Chex Mix                        2000
      1006 Mighty Munch                     625
      1001 Pot Rice                         600
      1004 Pot Rice                         540
      1005 Salt n Shake                     600
```

Pictorial Presentation:



item_id	item_name	item_unit	company_id
1	Chex Mix	Pcs	16
6	Cheez-It	Pcs	15
2	BN Biscuit	Pcs	15
3	Mighty Munch	Pcs	17
4	Pot Rice	Pcs	15
5	Jaffa Cakes	Pcs	18
7	Salt n Shake	Pcs	-

bill_no	item_id	sl_qty	sl_rate	bill_amt
1003	6	15	20	300
1004	4	18	30	540
1005	7	10	60	600
1006	3	25	25	625
1001	4	20	30	600
1002	1	40	50	2000

BILL_NO	ITEM_NAME	BILL_AMT
1002	Chex Mix	2000
1006	Mighty Munch	625
1001	Pot Rice	600
1004	Pot Rice	540
1005	Salt n Shake	600

Example of SQL Left Join using multiple tables

To filtered out those bill number, item name, company name and city and the bill amount for each bill, which items are available in foods table, and their manufacturer must have enlisted to supply that item, and no NULL value for manufacturer are not allowed, the following SQL statement can be used:

Sample table: foods

Sample table: company

Sample table: counter_sale

SQL Code:

```
SELECT a.bill_no, b.item_name,c.company_name,
c.company_city, a.bill_amt
FROM counter_sale a
LEFT JOIN foods b ON a.item_id=b.item_id
LEFT JOIN company c ON b.company_id=c.company_id
WHERE c.company_name IS NOT NULL
ORDER BY a.bill_no;
```

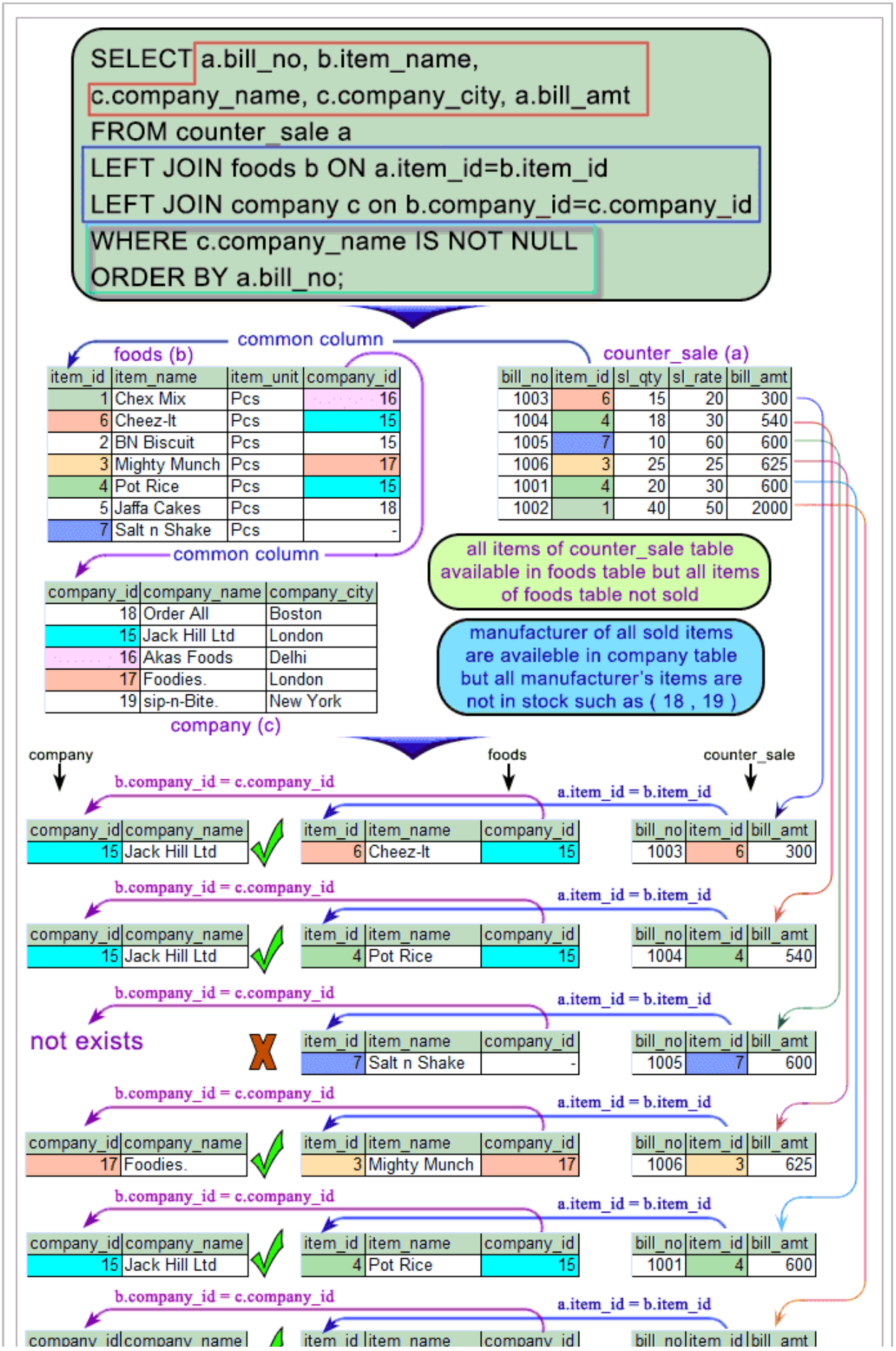
Explanation:

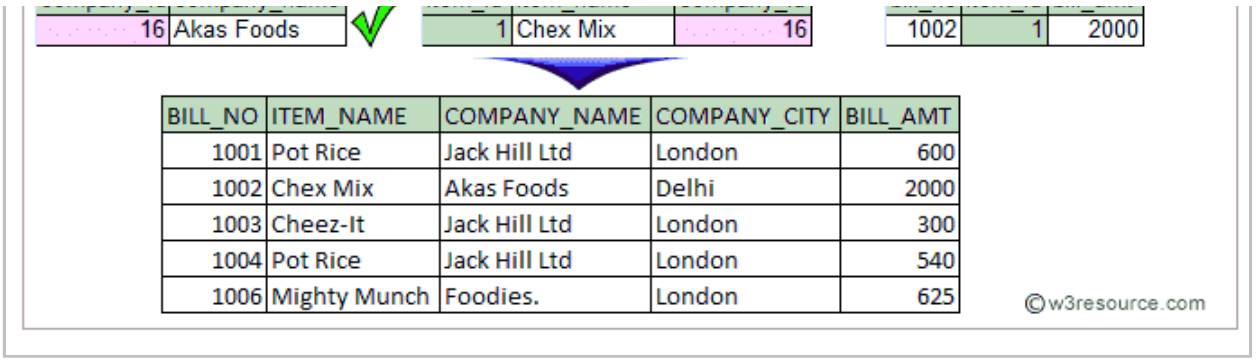
This SQL statement will first join all rows from the counter_sale table and only those rows from the foods table where the joined fields are matching and if the ON clause matches no records in the foods table, the join will still return rows, but the NULL in each column of the right table. Therefore this result will join with company table and all rows from result table and matched and unmatched rows from company table will also come, but for the unmatched rows of company table, the column value will be NULL. Therefore the WHERE clause will eliminate those rows which company name column value is NULL and after that, the ORDER BY clause will arrange the rows in ascending order according to the bill number.

Output:

```
   BILL_NO ITEM_NAME                 COMPANY_NAME               COMPANY_CITY
---------- ------------------------- -------------------------- ---------------
      1001 Pot Rice                  Jack Hill Ltd              London
      1002 Chex Mix                  Akas Foods                 Delhi
      1003 Cheez-It                  Jack Hill Ltd              London
      1004 Pot Rice                  Jack Hill Ltd              London
      1006 Mighty Munch              Foodies.                   London
```

Pictorial Presentation:


SELECT a.bill_no, b.item_name,
c.company_name, c.company_city, a.bill_amt
FROM counter_sale a
LEFT JOIN foods b ON a.item_id=b.item_id
LEFT JOIN company c on b.company_id=c.company_id
WHERE c.company_name IS NOT NULL
ORDER BY a.bill_no;
common column
foods (b)
item_id item_name item_unit company_id
1 Chex Mix Pcs 16
6 Cheez-It Pcs 15
2 BN Biscuit Pcs 15
3 Mighty Munch Pcs 17
4 Pot Rice Pcs 15
5 Jaffa Cakes Pcs 18
7 Salt n Shake Pcs -
counter_sale (a)
bill_no item_id sl_qty sl_rate bill_amt
1003 6 15 20 300
1004 4 18 30 540
1005 7 10 60 600
1006 3 25 25 625
1001 4 20 30 600
1002 1 40 50 2000
common column
company_id company_name company_city
18 Order All Boston
15 Jack Hill Ltd London
16 Akas Foods Delhi
17 Foodies. London
19 sip-n-Bite. New York
company (c)
all items of counter_sale table available in foods table but all items of foods table not sold
manufacturer of all sold items are availeble in company table but all manufacturer's items are not in stock such as (18 , 19)
company
foods
counter_sale
b.company_id = c.company_id
a.item_id = b.item_id
not exists



What is the difference between Left Join and Left Outer Join in SQL?

There is actually no difference between a left join and a left outer join – both of them refer to the similar operation in SQL.

Sample table: company

```
COMPANY_ID COMPANY_NAME              COMPANY_CITY
---------- ------------------------- -------------
18         Order All                 Boston
15         Jack Hill Ltd             London
16         Akas Foods                Delhi
17         Foodies.                  London
19         sip-n-Bite.               New York
```

Sample table: foods

```
ITEM_ID  ITEM_NAME                 ITEM_UNIT  COMPANY_ID
-------- ------------------------- ---------- ----------
1        Chex Mix                  Pcs        16
6        Cheez-It                  Pcs        15
2        BN Biscuit                Pcs        15
3        Mighty Munch              Pcs        17
4        Pot Rice                  Pcs        15
5        Jaffa Cakes               Pcs        18
7        Salt n Shake              Pcs
```

The important point to be noted that the very last row in the company table, the company ID does not exist in the foods table. Also, the very last row in the foods table the value of company ID is NULL and does not exist in the company table. These facts will prove to be significant of the left join.

Here the SQL statement without using "outer" with "left join".

SQL Code:

```
SELECT company.company_id,company.company_name,
foods.item_id, foods.item_name, foods.company_id
FROM company
LEFT JOIN foods
ON company.company_id = foods.company_id;
```

Running the SQL with the "outer" keyword, would give us the exact same results as running the SQL without the “outer”. Here the SQL statement with "outer" with "left join".

SQL Code:

```
SELECT company.company_id,company.company_name,
foods.item_id, foods.item_name, foods.company_id
FROM company
LEFT OUTER JOIN foods
ON company.company_id = foods.company_id;
```

A left outer join or left join retains all of the rows of the left table company, regardless of whether there is a row that matches on the right table foods. Here is the output below for both of the above statement.

Output:

```
COMPANY_ID COMPANY_NAME              ITEM_ID  ITEM_NAME                 COMPAN
---------- ------------------------- -------- ------------------------- ------
16         Akas Foods                1        Chex Mix                  16
15         Jack Hill Ltd             6        Cheez-It                  15
15         Jack Hill Ltd             2        BN Biscuit                15
17         Foodies.                  3        Mighty Munch              17
15         Jack Hill Ltd             4        Pot Rice                  15
18         Order All                 5        Jaffa Cakes               18
19         sip-n-Bite.               NULL     NULL                      NULL
```

Outputs of the said SQL statement shown here is taken by using Oracle Database 10g Express Edition.