

SQL JOINS

An SQL join clause is used to combine data from different tables, provided that the tables have a related column.

Example:

Consider the product order table of a business

Order ID	Product	Price	Customer ID
1077	USB Charger	5000	899
1078	Air pods	12000	899
1079	Toilet paper	1000	700
1083	Disinfectant	3000	930

Another table records the details of the Customers of the business.

Customer ID	Customer Name
899	Jack Rich
700	Dim Kim
554	Ada Eze
403	Heath Led

The two tables are put in the same database.

To get a customer's name with the product they bought and its price would require getting data from both tables. This is the essence of SQL joins.

Types of SQL joins

INNER JOIN

An Inner Join returns or provides all the records that have matching values in the tables being queried.

Remember a JOIN requires both tables to have a related column. The related column in the examples above is the Order ID column. Using the tables given above, if an INNER JOIN is performed involving both tables, it would yield the following result:

```
SELECT * FROM Sales INNER JOIN `Customer details` ON Sales.`Customer ID` = `Customer details`.`Customer ID`
```

Order ID	Product	Price	Customer ID	Customer ID	Customer Name
1077	USB Charger	5000	899	899	Jack Rich
1078	AirPods	12000	899	899	Jack Rich
1079	Toilet paper	1000	700	700	Dim Kim

From the table above, notice that the last row in the sales table is not included in this joined table. This is because the Customer ID in that column does not match any of the Customer IDs in the Customer table.

Let's assume there is a third table called Loyalty_Bonus that also has the Order ID column.

Order ID	Gift
1077	1000 Airtime
1078	Carton of noodles
1999	500 Airtime
1798	Face towel

To perform a join on the three tables, the syntax would be:

```
SELECT * FROM ((Sales INNER JOIN `Customer details` ON Sales.`Customer ID` = `Customer details`.`Customer ID`) INNER JOIN Loyalty_Bonus ON Sales.`Order ID` = Loyalty_Bonus.`Order ID`)
```

Order ID	Product	Price	Customer ID	Customer ID	Customer Name	Order ID	Gift
1077	USB Charger	5000	899	899	Jack Rich	1077	1000 Airtime
1078	AirPods	12000	899	899	Jack Rich	1078	Carton of noodles

LEFT JOIN

A LEFT JOIN returns or provides all the records that have matching values in the tables being queried, in addition to all the values in the left table. The table that will be considered as the left, is the table mentioned first in the JOIN clause.

Exaple:

```
SELECT * FROM Sales LEFT JOIN `Customer details` ON Sales.`Customer ID` = `Customer details`.`Customer ID`
```

In the above example, the Sales table is the one written first, so it is taken as the left table. The result of the above query would be:

Order ID	Product	Price	Customer ID	Customer ID	Customer Name
1077	USB Charger	5000	899	899	Jack Rich
1078	Air Pods	12000	899	899	Jack Rich
1079	Toilet paper	1000	700	700	Dim Kim
1083	Disinfectant	3000	930	NULL	NULL

Notice from the above result that all the rows from the Sales table, which is the LEFT table, are included.

RIGHT JOIN

A RIGHT JOIN returns or provides all the records that have matching values in the tables being queried, in addition to all the values in the RIGHT table. The table that will be considered as the RIGHT, is the table mentioned last in the JOIN clause.

Example:

```
SELECT * FROM Sales RIGHT JOIN `Customer details` ON Sales.`Customer ID` =  
`Customer details`.`Customer ID`
```

Order ID	Product	Price	Customer ID	Customer ID	Customer Name
1077	USB Charger	5000	899	899	Jack Rich
1078	Air Pods	12000	899	899	Jack Rich
1079	Toilet paper	1000	700	700	Dim Kim
NULL	NULL	NULL	NULL	554	Ada Eze
NULL	NULL	NULL	NULL	403	Heath Led

FULL JOIN aka FULL OUTER JOIN

A RIGHT JOIN returns or provides all the records that have matching values in both the left table and the RIGHT table.

Example:

```
SELECT * FROM sales FULL JOIN Customer_details ON Customer_details.`Customer ID`=  
sales.`Customer ID`
```

Order ID	Product	Price	Customer ID	Customer ID	Customer Name
1077	USB Charger	5000	899	899	Jack Rich
1078	Air Pods	12000	899	899	Jack Rich
1079	Toilet paper	1000	700	700	Dim Kim
1083	Disinfectant	3000	930	NULL	NULL
NULL	NULL	NULL	NULL	554	Ada Eze
NULL	NULL	NULL	NULL	403	Heath Led

N/B: The full JOIN operation is not available on MySQL. A different method can be used to get the same result as a FULL JOIN.

```
SELECT * FROM sales LEFT JOIN Customer_details ON Customer_details.`Customer ID`=
sales.`Customer ID`
```

```
UNION
```

```
SELECT * FROM sales RIGHT JOIN Customer_details ON Customer_details.`Customer ID`=
sales.`Customer ID`
```