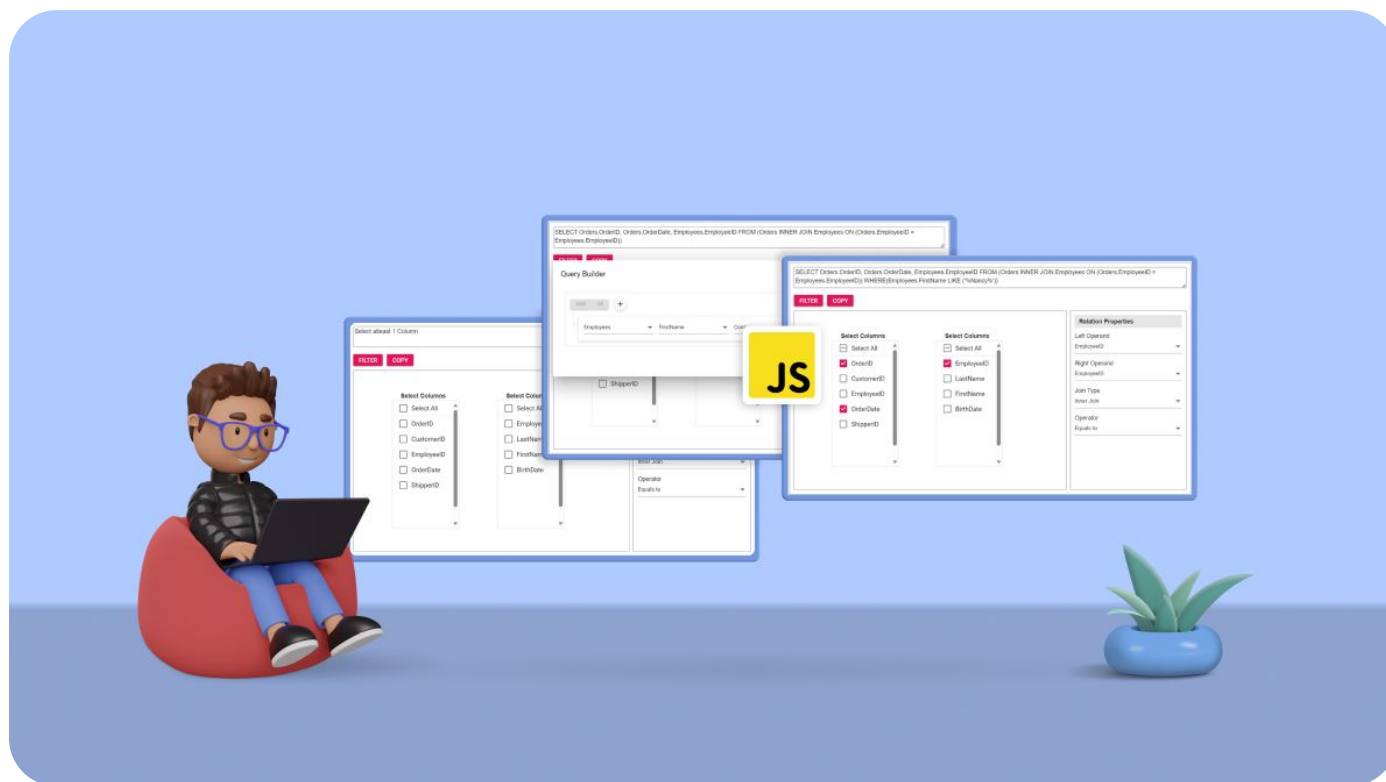


# How to Join Two Tables Using JavaScript Query Builder?

 **Satheeskumar S** •  6 min read •  Nov 18, 2024 • **Updated**





Build stunning  
real world apps using

## JavaScript Controls

 **Try it free**

**TL;DR:** *Let's see how to use the Syncfusion JavaScript Query Builder to join two tables. This blog guides you through creating a custom JoinComponent and configuring WHERE, SELECT, and JOIN clauses using list boxes and dropdowns. The steps ensure efficient query generation, making it easy to connect and manage complex data sources. Check out the Stackblitz demo for complete code examples.*

Syncfusion [JavaScript Query Builder](#) is an interactive UI element designed to create queries. Its rich features include complex data binding, templating, importing, and exporting queries in JSON and SQL formats. Additionally, it can convert queries into predicates for use with a data manager.

This blog explains how to join two tables using the JavaScript Query Builder component. Here, we'll integrate the Query Builder component with complex data binding support to connect two distinct tables. We'll create the query for the SQL **WHERE** clause, embed a list box for crafting the **SELECT** clause, and a dropdown list to streamline the construction of join queries.

**Note:** Before proceeding, refer to the [getting started with JavaScript Query Builder documentation](#).

## Create a custom component using JavaScript Query Builder

Let's create a custom component known as **JoinComponent** to facilitate the creation of join queries and offer flexibility through a set of parameters. With this component, users can specify the element ID, data sources for the tables, table names, and left and right operands, all essential for constructing join queries.

Within this **JoinComponent**, we'll integrate the JavaScript Query Builder within a [Dialog](#) component. We'll also incorporate [ListBox](#) and [Dropdown List](#) components to enhance the user's experience and streamline the process of configuring and executing join operations. The result is a versatile and user-friendly component that simplifies the creation of join queries.

You can refer to the code example for creating the custom **JoinComponent** in this [Stackblitz repository](#).

# Joining two tables using JavaScript Query Builder

Once the custom component is created, follow these steps to join two tables.

## Step 1: Create a WHERE clause

The SQL **WHERE** clause filters records in a database according to the specified conditions.

In this context, our JavaScript Query Builder component plays a crucial role in obtaining the value for the **WHERE** clause. It supports [complex data binding](#), enabling the generation of rules and SQL queries by combining information from two tables. This functionality is achieved by using a **column directive** to specify complex tables and including a **separator** property within the component.

By configuring these properties, the Query Builder will be rendered with two tables, producing a resultant join query resembling the code snippet given below.

```
Employees.FirstName LIKE ("%Nancy%")
```



## Step 2: Create a **SELECT** clause

The **SELECT** clause in SQL designates the columns or expressions we wish to retrieve from one or more database tables. To facilitate this, we'll render a listbox component to select the required columns from the left and right table.

## Step 3: Create a **JOIN** clause

Joining tables involves combining rows from two or more tables based on the related column or columns. It retrieves data distributed across multiple tables and creates a result set that combines relevant information from those tables.

Here are the key aspects of joining tables:

- **Related columns:** Table joins rely on columns that establish relationships between tables. Typically, these columns represent primary and foreign keys. A primary key identifies each row in a table, and a foreign key creates a link between two tables by referring to the primary key of another table.
- **Join types:** There are different types of joins, including inner, left, right, and full outer joins.

- **Join conditions:** Join conditions specify the criteria for combining rows from different tables. They typically involve comparing the related columns using operators such as =, <>, <, >, etc. Join conditions can also involve multiple columns or complex expressions.

To perform a join operation, we need relational columns, a join type, and a join condition. To facilitate this, we'll render a dropdown list component to select the **Left** and **Right Operands**. The **Join Type** dropdown list provides options for different types of joins, such as INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN. Lastly, the **Operator** dropdown list allows you to specify the conditions for connecting the two operands.

Refer to the following image.

Join component user interface

## Step 4: Integrating the custom component into the app

To incorporate the custom **JoinComponent** into your app, import it and place it within a **div** element during rendering. You can provide essential properties to tailor the component to your needs, streamlining its integration into your app's user interface.

Upon clicking the **Filter** button, the Query Builder component will be displayed, allowing users to construct a query. Subsequently, clicking the **Copy** button will

copy the generated query to the clipboard.

Refer to the following code example to render the custom component on the HTML page.

```
<div id="join"></div>
```

 Copy

Refer to the following Typescript code to render the custom component.

```
import { JoinComponent } from './JoinComponent';

let ordersData = [
  { "OrderID": 10248, "CustomerID": 9, "EmployeeID": 5, "OrderDate": "7/4/1996", "
  { "OrderID": 10249, "CustomerID": 81, "EmployeeID": 6, "OrderDate": "7/5/1996",
];

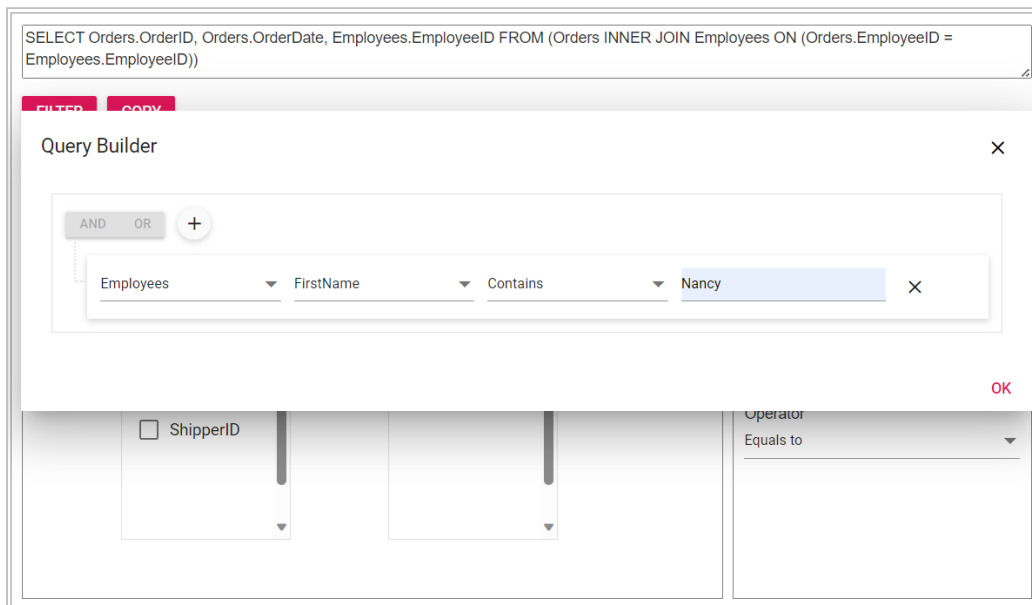
let employeesData = [
  { "EmployeeID": 1, "LastName": "Davolio", "FirstName": "Nancy", "BirthDate": "
  { "EmployeeID": 2, "LastName": "Fuller", "FirstName": "Andrew", "BirthDate": "
  { "EmployeeID": 3, "LastName": "Leverling", "FirstName": "Janet", "BirthDate":
  { "EmployeeID": 4, "LastName": "Peacock", "FirstName": "Margaret", "BirthDate"
  { "EmployeeID": 5, "LastName": "Buchanan", "FirstName": "Steven", "BirthDate":
  { "EmployeeID": 6, "LastName": "Suyama", "FirstName": "Michael", "BirthDate":
];
```

 Copy



```
let comp: JoinComponent = new JoinComponent(  
    'join', // component ID  
    ordersData, // left table  
    employeesData, // right table  
    'Orders', // left table name  
    'Employees', // right table name  
    'EmployeeID', // left operand  
    'EmployeeID' // right operand  
);
```

Refer to the following images displaying the Query Builder and the join component user interfaces.



JavaScript Query Builder user interface

SELECT Orders.OrderID, Orders.OrderDate, Employees.EmployeeID FROM (Orders INNER JOIN Employees ON (Orders.EmployeeID = Employees.EmployeeID)) WHERE(Employees.FirstName LIKE ('%Nancy%'))

FILTER COPY

Select Columns

☐ Select All
 

☒ OrderID
 ☐ CustomerID
 ☐ EmployeeID
 ☒ OrderDate
 ☐ ShipperID

Select Columns

☐ Select All
 

☒ EmployeeID
 ☐ LastName
 ☐ FirstName
 ☐ BirthDate

Relation Properties

Left Operand  
EmployeeID
 Right Operand  
EmployeeID
 Join Type  
Inner Join
 Operator  
Equals to

Joining two tables using the JavaScript Query Builder

The sample join query is as follows, and you can directly validate this query using [this link](#).

 Copy

```
SELECT Orders.OrderID, Orders.OrderDate, Employees.EmployeeID FROM (Orders INNER
```

## Reference

For more details, refer to the entire code example for [joining two tables using the JavaScript Query Builder on Stackblitz](#).



Explore the endless possibilities with Syncfusion's outstanding React UI components.

Try It Free

## Conclusion

Thanks for reading! In this blog, we've explored how to join two tables using Syncfusion [JavaScript Query Builder](#). Follow these steps to achieve similar results, and feel free to share your thoughts or questions in the comments below.

If you're an existing customer, you can download the latest version of Essential Studio® from the [License and Downloads](#) page. For those new to Syncfusion, try our 30-day [free trial](#) to explore all our features.

You can contact us through our [support forum](#), [support portal](#), or [feedback portal](#). We are here to help you succeed!

## Related blogs

- [Top 5 Techniques to Protect Web Apps from Unauthorized JavaScript Execution](#)
- [Easily Render Flat JSON Data in JavaScript File Manager](#)
- [Effortlessly Synchronize JavaScript Controls Using DataManager](#)
- [Optimizing Productivity: Integrate Salesforce with JavaScript Scheduler](#)



MEET THE AUTHOR

**Satheeskumar S**

Satheeskumar works as a product manager at Syncfusion, where he specializes in the development of web components with cutting-edge technologies. He is interested in the latest web technologies and provides solutions for great products.