

Sql server, .net and c# video tutorial

Free C#, .Net and Sql server video tutorial for beginners and intermediate programmers.

[Support us](#) [.Net Basics](#) [C#](#) [SQL](#) [ASP.NET](#) [ADO.NET](#) [MVC](#) [Slides](#) [C# Programs](#) [Subscribe](#) [Buy DVD](#)

Cross apply and outer apply in sql server

Suggested Videos

Part 88 - Difference between except and not in sql server

Part 89 - Intersect operator in sql server

Part 90 - Difference between union intersect and except in sql server

In this video we will discuss **cross apply and outer apply in sql server** with examples.

We will use the following 2 tables for examples in this demo



Best software training and placements in marathahalli, bangalore. For further details please call 09945699393.

Complete Tutorials

JavaScript tutorial

Bootstrap tutorial

Angular tutorial for beginners

Angular 5 Tutorial for beginners

Important Videos

The Gift of Education

Web application for your business

How to become .NET developer

Resources available to help you

Dot Net Video Tutorials

Department Table		Employee Table				
Id	DepartmentName	Id	Name	Gender	Salary	DepartmentId
1	IT	1	Mark	Male	50000	1
2	HR	2	Mary	Female	60000	3
3	Payroll	3	Steve	Male	45000	2
4	Administration	4	John	Male	56000	1
5	Sales	5	Sara	Female	39000	2

SQL Script to create the tables and populate with test data

Create table Department

```
(
    Id int primary key,
    DepartmentName nvarchar(50)
)
Go
```

Insert into Department values (1, 'IT')

Insert into Department values (2, 'HR')

Insert into Department values (3, 'Payroll')

Insert into Department values (4, 'Administration')

Insert into Department values (5, 'Sales')

Go

Create table Employee

```
(
    Id int primary key,
    Name nvarchar(50),
    Gender nvarchar(10),
    Salary int,
    DepartmentId int foreign key references Department(Id)
)
Go
```

Insert into Employee values (1, 'Mark', 'Male', 50000, 1)

[ASP.NET Core Tutorial](#)

[Angular 6 Tutorial](#)

[Angular CRUD Tutorial](#)

[Angular CLI Tutorial](#)

[Angular 2 Tutorial](#)

[Design Patterns](#)

[SOLID Principles](#)

[ASP.NET Web API](#)

[Bootstrap](#)

[AngularJS Tutorial](#)

[jQuery Tutorial](#)

[JavaScript with ASP.NET Tutorial](#)

[JavaScript Tutorial](#)

[Charts Tutorial](#)

[LINQ](#)

[LINQ to SQL](#)

[LINQ to XML](#)

[Entity Framework](#)

[WCF](#)

[ASP.NET Web Services](#)

[Insert into Employee values](#) (2, 'Mary', 'Female', 60000, 3)
[Insert into Employee values](#) (3, 'Steve', 'Male', 45000, 2)
[Insert into Employee values](#) (4, 'John', 'Male', 56000, 1)
[Insert into Employee values](#) (5, 'Sara', 'Female', 39000, 2)
[Go](#)

We want to retrieve all the matching rows between **Department** and **Employee** tables.

DepartmentName	Name	Gender	Salary
IT	Mark	Male	50000
Payroll	Mary	Female	60000
HR	Steve	Male	45000
IT	John	Male	56000
HR	Sara	Female	39000

This can be very easily achieved using an Inner Join as shown below.

[Select](#) D.DepartmentName, E.Name, E.Gender, E.Salary
[from](#) Department D
 Inner Join Employee E
[On](#) D.Id = E.DepartmentId

Now if we want to retrieve all the matching rows between **Department** and **Employee** tables + the non-matching rows from the LEFT table (**Department**)

DepartmentName	Name	Gender	Salary
IT	Mark	Male	50000
IT	John	Male	56000
HR	Steve	Male	45000
HR	Sara	Female	39000
Payroll	Mary	Female	60000
Administration	NULL	NULL	NULL
Sales	NULL	NULL	NULL

This can be very easily achieved using a Left Join as shown below.

[Dot Net Basics](#)

[C#](#)

[SQL Server](#)

[ADO.NET](#)

[ASP.NET](#)

[GridView](#)

[ASP.NET MVC](#)

[Visual Studio Tips and Tricks](#)

[Dot Net Interview Questions](#)

Slides

[Entity Framework](#)

[WCF](#)

[ASP.NET Web Services](#)

[Dot Net Basics](#)

[C#](#)

[SQL Server](#)

[ADO.NET](#)

[ASP.NET](#)

[GridView](#)

[ASP.NET MVC](#)

[Visual Studio Tips and Tricks](#)

```
Select D.DepartmentName, E.Name, E.Gender, E.Salary
from Department D
Left Join Employee E
On D.Id = E.DepartmentId
```

Now let's assume we do not have access to the Employee table. Instead we have access to the following Table Valued function, that returns all employees belonging to a department by Department Id.

```
Create function fn_GetEmployeesByDepartmentId(@DepartmentId int)
Returns Table
as
Return
(
    Select Id, Name, Gender, Salary, DepartmentId
    from Employee where DepartmentId = @DepartmentId
)
Go
```

The following query returns the employees of the department with Id =1.

```
Select * from fn_GetEmployeesByDepartmentId(1)
```

Now if you try to perform an Inner or Left join between **Department** table and **fn_GetEmployeesByDepartmentId()** function you will get an error.

```
Select D.DepartmentName, E.Name, E.Gender, E.Salary
from Department D
Inner Join fn_GetEmployeesByDepartmentId(D.Id) E
On D.Id = E.DepartmentId
```

If you execute the above query you will get the following error

Msg 4104, Level 16, State 1, Line 3
The multi-part identifier "D.Id" could not be bound.

This is where we use **Cross Apply** and **Outer Apply** operators. **Cross Apply** is semantically equivalent to **Inner Join** and **Outer Apply** is semantically equivalent to **Left Outer Join**.

Java Video Tutorials

Part 1 : [Video](#) | [Text](#) | [Slides](#)

Part 2 : [Video](#) | [Text](#) | [Slides](#)

Part 3 : [Video](#) | [Text](#) | [Slides](#)

Interview Questions

C#

SQL Server

Written Test

Just like Inner Join, Cross Apply retrieves only the matching rows from the Department table and fn_GetEmployeesByDepartmentId() table valued function.

```
Select D.DepartmentName, E.Name, E.Gender, E.Salary  
from Department D  
Cross Apply fn_GetEmployeesByDepartmentId(D.Id) E
```

Just like Left Outer Join, Outer Apply retrieves all matching rows from the Department table and fn_GetEmployeesByDepartmentId() table valued function + non-matching rows from the left table (Department)

```
Select D.DepartmentName, E.Name, E.Gender, E.Salary  
from Department D  
Outer Apply fn_GetEmployeesByDepartmentId(D.Id) E
```

How does Cross Apply and Outer Apply work

- The APPLY operator introduced in SQL Server 2005, is used to join a table to a table-valued function.
- The Table Valued Function on the right hand side of the APPLY operator gets called for each row from the left (also called outer table) table.
- Cross Apply returns only matching rows (semantically equivalent to Inner Join)
- Outer Apply returns matching + non-matching rows (semantically equivalent to Left Outer Join). The unmatched columns of the table valued function will be set to NULL.

WWW.PRAGIMTECH.COM

**CLICK HERE SQL SERVER YOUTUBE
PLAYLIST**

TRAINING + PLACEMENTS = OUR SUCCESS

WWW.FACEBOOK.COM/PRAGIMTECH.COM

6 comments:

Anonymous September 8, 2015 at 1:37 PM

Please discuss Sequence in SqlServer

[Reply](#)

Nawaraj Subedi September 9, 2015 at 12:39 AM

Thank you very much for great video tutorial. This is very helpful to me and others also. My request is also upload the videos of OVER () CLAUSE and Partition also Thank You...!!!!

[Reply](#)



Bhargava November 24, 2015 at 3:48 PM

Hi Venkat,

In the above Outer Apply you have mentioned it is same as left outer join. Any specific

reason why you have mentioned left "OUTER" join. Left join and left outer join are the same. Please correct me.

Reply

Replies



ipsita pani December 17, 2015 at 8:58 AM

OUTER JOIN IS 3 THREE TYPES.
LEFT OUTER JOIN
RIGHT OUTER JOIN

FULL OUTER JOIN.

THERE IS NO DIFFERENCE BETWEEN LEFT JOIN AND LEFT OUTER JOIN.
I HOPE IT CLEARS YOUR DOUBT.

Reply



Unknown December 14, 2018 at 2:04 AM

You say outer apply is semantically equivalent to left outer join can you draw the same parallel with right outer join?

Reply



Aqib April 9, 2019 at 2:13 PM

```
SELECT D.DepartmentName, E.Name,E.Gender,E.Salary  
FROM Employee E  
RIGHT JOIN Department D  
ON E.DepartmentId = D.Id
```

HOW to Apply OUTER APPLY JOIN on this

Reply

Enter your comment...



Comment as:

toilati123vn@g ▼

Publish

Preview

If you like this website, please share with your friends on facebook and Google+ and recommend us on google using the g+1 button on the top right hand corner.

Links to this post

[Create a Link](#)

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)

Powered by Blogger.