المزيد

إنشاء مدونة إلكترونية تسجيل الدخول

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

Joins in sql server - Part 12

Joins in SQL server are used to query (retrieve) data from 2 or more related tables. In general tables are related to each other using foreign key constraints.

Please watch Parts 3 and 5 in this video series, before continuing with this video.
Part 3 - Creating and working with tables
Part 5 - Cascading referential integrity constraint

In SQL server, there are different types of JOINS.
1. CROSS JOIN
2. INNER JOIN
3. OUTER JOIN

Outer Joins are again divided into 3 types
1. Left Join or Left Outer Join
2. Right Join or Right Outer Join
3. Full Join or Full Outer Join

Now let's understand all the JOIN types, with examples and the differences

Complete Tutorials

JavaScript tutorial

Bootstrap tutorial

Angular tutorial for beginners

Angular 5 Tutorial for beginners

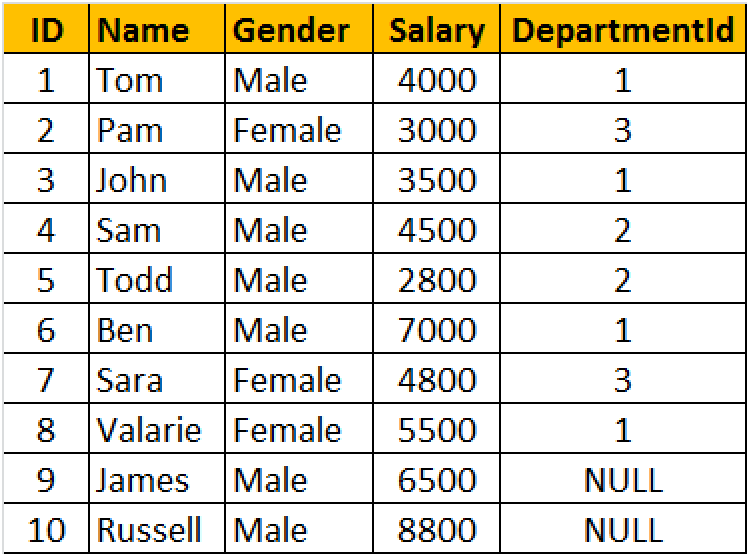
Important Videos

The Gift of Education

Web application for your business

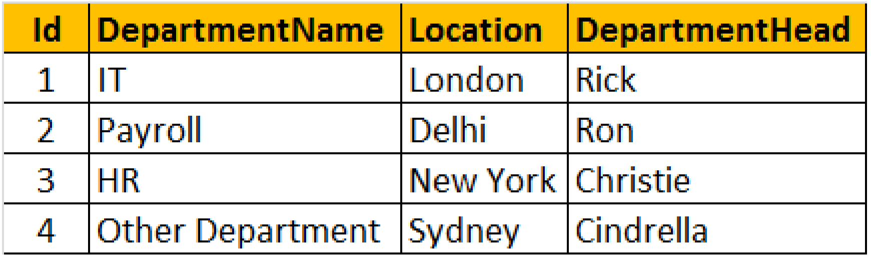
between them.

Employee Table (tblEmployee)



ID	Name	Gender	Salary	DepartmentId
1	Tom	Male	4000	1
2	Pam	Female	3000	3
3	John	Male	3500	1
4	Sam	Male	4500	2
5	Todd	Male	2800	2
6	Ben	Male	7000	1
7	Sara	Female	4800	3
8	Valarie	Female	5500	1
9	James	Male	6500	NULL
10	Russell	Male	8800	NULL

Departments Table (tblDepartment)



Id	DepartmentName	Location	DepartmentHead
1	IT	London	Rick
2	Payroll	Delhi	Ron
3	HR	New York	Christie
4	Other Department	Sydney	Cindrella

SQL Script to create tblEmployee and tblDepartment tables

```
Create table tblDepartment
(
    ID int primary key,
    DepartmentName nvarchar(50),
    Location nvarchar(50),
    DepartmentHead nvarchar(50)
)
Go

Insert into tblDepartment values (1, 'IT', 'London', 'Rick')
```

How to become .NET developer

Resources available to help you

Dot Net Video Tutorials

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

```
Insert into tblDepartment values (2, 'Payroll', 'Delhi', 'Ron')
Insert into tblDepartment values (3, 'HR', 'New York', 'Christie')
Insert into tblDepartment values (4, 'Other Department', 'Sydney', 'Cindrella')
Go

Create table tblEmployee
(
    ID int primary key,
    Name nvarchar(50),
    Gender nvarchar(50),
    Salary int,
    DepartmentId int foreign key references tblDepartment(Id)
)
Go

Insert into tblEmployee values (1, 'Tom', 'Male', 4000, 1)
Insert into tblEmployee values (2, 'Pam', 'Female', 3000, 3)
Insert into tblEmployee values (3, 'John', 'Male', 3500, 1)
Insert into tblEmployee values (4, 'Sam', 'Male', 4500, 2)
Insert into tblEmployee values (5, 'Todd', 'Male', 2800, 2)
Insert into tblEmployee values (6, 'Ben', 'Male', 7000, 1)
Insert into tblEmployee values (7, 'Sara', 'Female', 4800, 3)
Insert into tblEmployee values (8, 'Valarie', 'Female', 5500, 1)
Insert into tblEmployee values (9, 'James', 'Male', 6500, NULL)
Insert into tblEmployee values (10, 'Russell', 'Male', 8800, NULL)
Go
```

General Formula for Joins

```
SELECT      ColumnList
FROM        LeftTableName
JOIN_TYPE   RightTableName
ON          JoinCondition
```

CROSS JOIN

CROSS JOIN, produces the cartesian product of the 2 tables involved in the join. For example, in the Employees table we have 10 rows and in the Departments table we have 4 rows. So, a cross join between these 2 tables produces 40 rows. Cross Join

WCF

ASP.NET Web Services

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

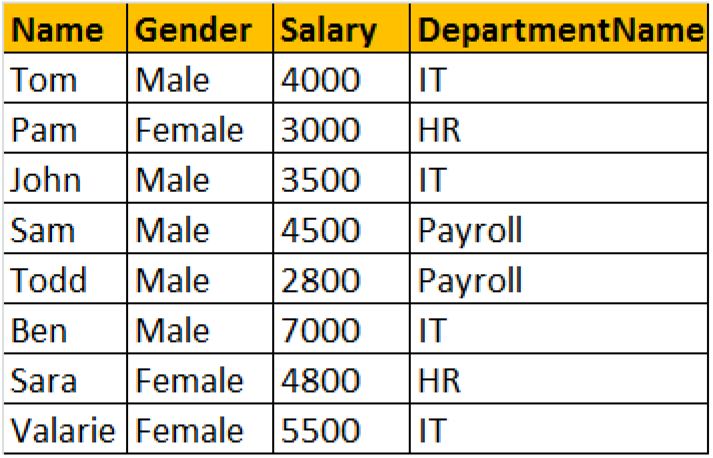
shouldn't have ON clause.

CROSS JOIN Query:

```
SELECT Name, Gender, Salary, DepartmentName
FROM tblEmployee
CROSS JOIN tblDepartment
```

JOIN or INNER JOIN

Write a query, to retrieve Name, Gender, Salary and DepartmentName from Employees and Departments table. The output of the query should be as shown below.



Name	Gender	Salary	DepartmentName
Tom	Male	4000	IT
Pam	Female	3000	HR
John	Male	3500	IT
Sam	Male	4500	Payroll
Todd	Male	2800	Payroll
Ben	Male	7000	IT
Sara	Female	4800	HR
Valarie	Female	5500	IT

```
SELECT Name, Gender, Salary, DepartmentName
FROM tblEmployee
INNER JOIN tblDepartment
ON tblEmployee.DepartmentId = tblDepartment.Id
```

OR

```
SELECT Name, Gender, Salary, DepartmentName
FROM tblEmployee
JOIN tblDepartment
ON tblEmployee.DepartmentId = tblDepartment.Id
```

Note: JOIN or INNER JOIN means the same. It's always better to use INNER JOIN, as this explicitly specifies your intention.

GridView

ASP.NET MVC

Visual Studio Tips and Tricks

Java Video Tutorials

Part 1 : Video | Text | Slides

Part 2 : Video | Text | Slides

Part 3 : Video | Text | Slides

Interview Questions

C#

SQL Server

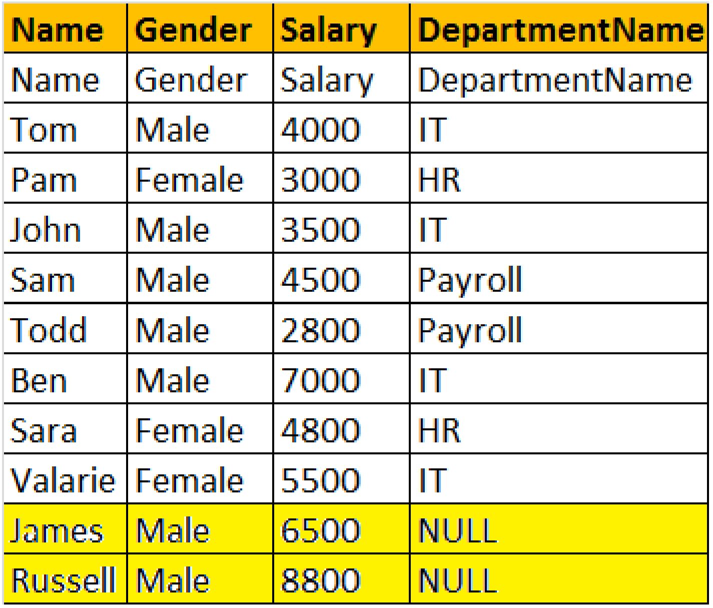
Written Test

If you look at the output, we got only 8 rows, but in the Employees table, we have 10 rows. We didn't get JAMES and RUSSELL records. This is because the DEPARTMENTID, in Employees table is NULL for these two employees and doesn't match with ID column in Departments table.

So, in summary, INNER JOIN, returns only the matching rows between both the tables. Non matching rows are eliminated.

LEFT JOIN or LEFT OUTER JOIN

Now, let's say, I want all the rows from the Employees table, including JAMES and RUSSELL records. I want the output, as shown below.



Name	Gender	Salary	DepartmentName
Name	Gender	Salary	DepartmentName
Tom	Male	4000	IT
Pam	Female	3000	HR
John	Male	3500	IT
Sam	Male	4500	Payroll
Todd	Male	2800	Payroll
Ben	Male	7000	IT
Sara	Female	4800	HR
Valarie	Female	5500	IT
James	Male	6500	NULL
Russell	Male	8800	NULL

```
SELECT Name, Gender, Salary, DepartmentName
FROM tblEmployee
LEFT OUTER JOIN tblDepartment
ON tblEmployee.DepartmentId = tblDepartment.Id
```

OR

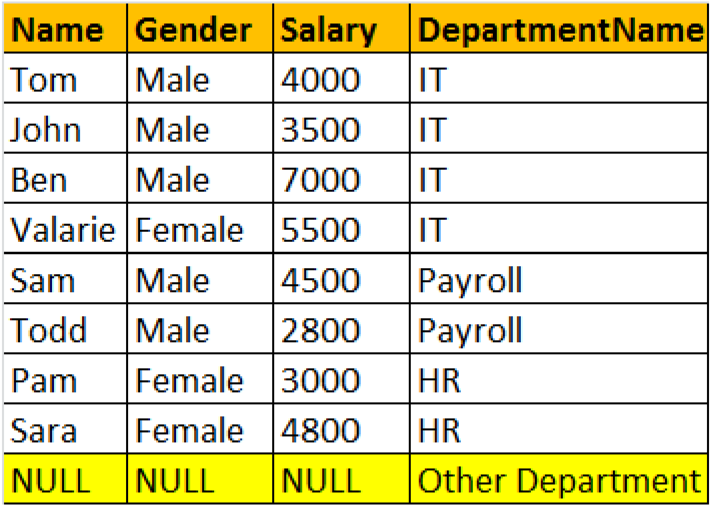
```
SELECT Name, Gender, Salary, DepartmentName
FROM tblEmployee
LEFT JOIN tblDepartment
ON tblEmployee.DepartmentId = tblDepartment.Id
```

Note: You can use, LEFT JOIN or LEFT OUTER JOIN. OUTER keyowrd is optional

LEFT JOIN, returns all the matching rows + non matching rows from the left table. In reality, INNER JOIN and LEFT JOIN are extensively used.

RIGHT JOIN or RIGHT OUTER JOIN

I want, all the rows from the right table. The query output should be, as shown below.



Name	Gender	Salary	DepartmentName
Tom	Male	4000	IT
John	Male	3500	IT
Ben	Male	7000	IT
Valarie	Female	5500	IT
Sam	Male	4500	Payroll
Todd	Male	2800	Payroll
Pam	Female	3000	HR
Sara	Female	4800	HR
NULL	NULL	NULL	Other Department

```
SELECT Name, Gender, Salary, DepartmentName
FROM tblEmployee
RIGHT OUTER JOIN tblDepartment
ON tblEmployee.DepartmentId = tblDepartment.Id
```

OR

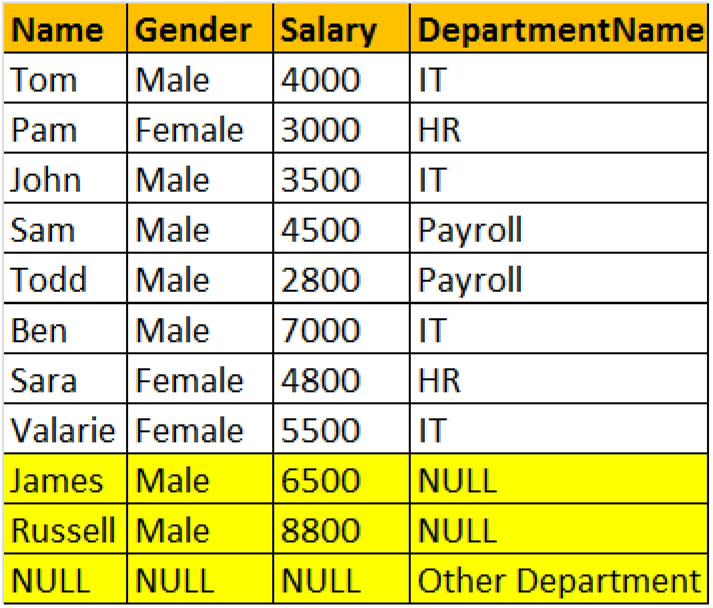
```
SELECT Name, Gender, Salary, DepartmentName
FROM tblEmployee
RIGHT JOIN tblDepartment
```

```
ON tblEmployee.DepartmentId = tblDepartment.Id
```

Note: You can use, RIGHT JOIN or RIGHT OUTER JOIN. OUTER keyowrd is optional

RIGHT JOIN, returns all the matching rows + non matching rows from the right table.

FULL JOIN or FULL OUTER JOIN
I want all the rows from both the tables involved in the join. The query output should be, as shown below.



Name	Gender	Salary	DepartmentName
Tom	Male	4000	IT
Pam	Female	3000	HR
John	Male	3500	IT
Sam	Male	4500	Payroll
Todd	Male	2800	Payroll
Ben	Male	7000	IT
Sara	Female	4800	HR
Valarie	Female	5500	IT
James	Male	6500	NULL
Russell	Male	8800	NULL
NULL	NULL	NULL	Other Department

```
SELECT Name, Gender, Salary, DepartmentName
FROM tblEmployee
FULL OUTER JOIN tblDepartment
ON tblEmployee.DepartmentId = tblDepartment.Id
```

OR

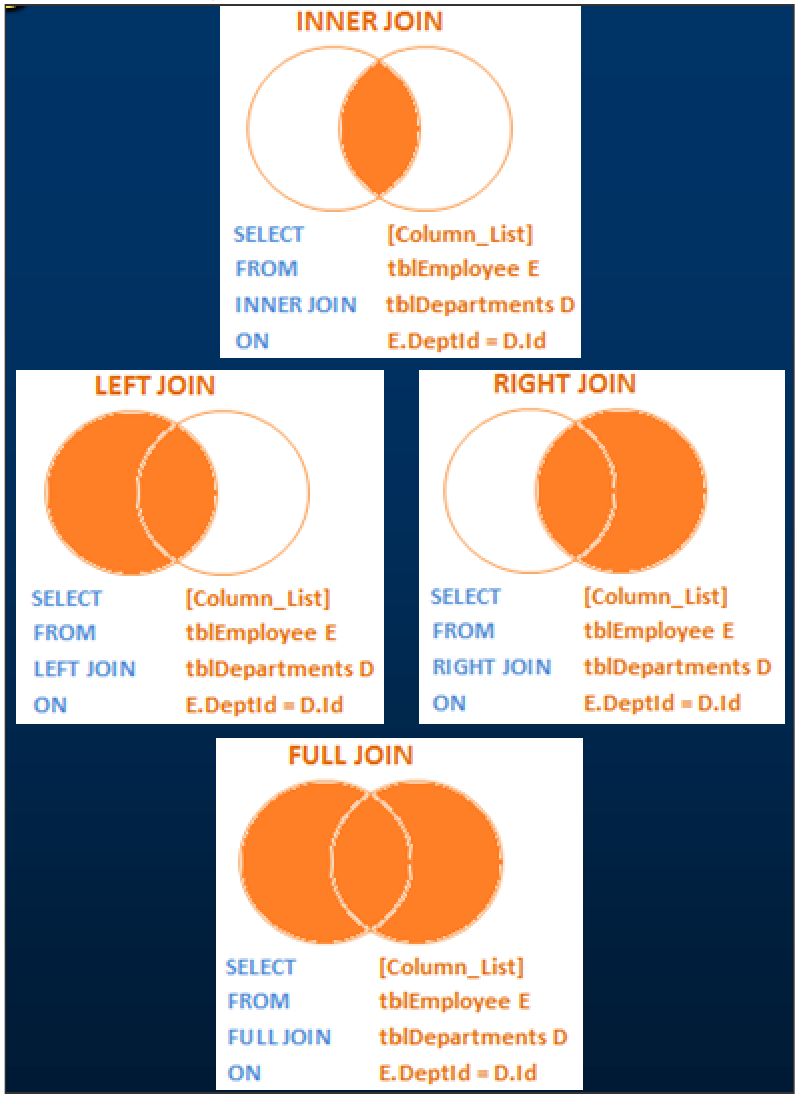
```
SELECT Name, Gender, Salary, DepartmentName
FROM tblEmployee
FULL JOIN tblDepartment
```

ON tblEmployee.DepartmentId = tblDepartment.Id

Note: You can use, FULLJOIN or FULL OUTER JOIN. OUTER keyowrd is optional

FULL JOIN, returns all rows from both the left and right tables, including the non matching rows.

Joins Summary



INNER JOIN

```
SELECT      [Column_List]
FROM        tblEmployee E
INNER JOIN  tblDepartments D
ON          E.DeptId = D.Id
```

LEFT JOIN

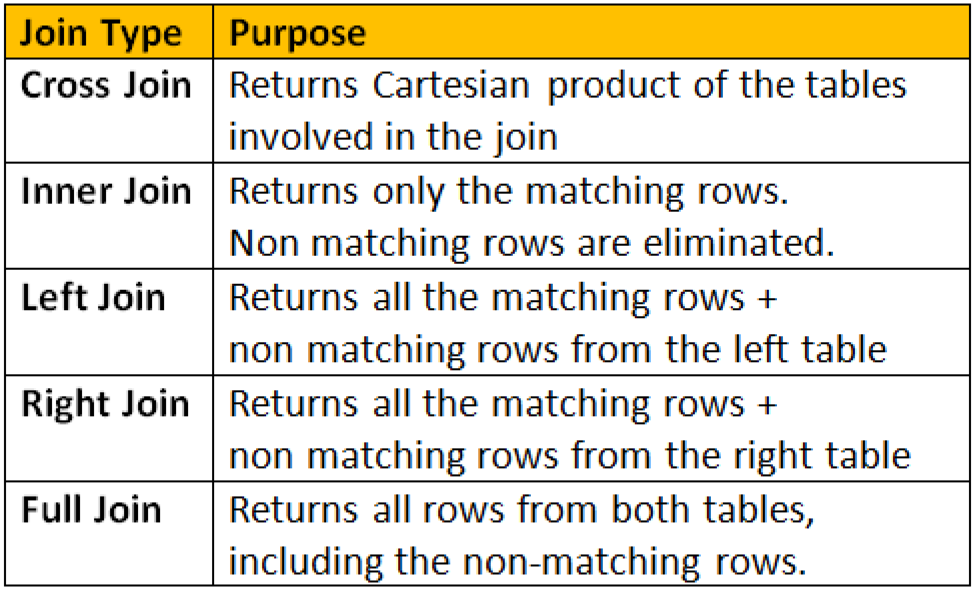
```
SELECT      [Column_List]
FROM        tblEmployee E
LEFT JOIN   tblDepartments D
ON          E.DeptId = D.Id
```

RIGHT JOIN

```
SELECT      [Column_List]
FROM        tblEmployee E
RIGHT JOIN  tblDepartments D
ON          E.DeptId = D.Id
```

FULL JOIN

```
SELECT      [Column_List]
FROM        tblEmployee E
FULL JOIN   tblDepartments D
ON          E.DeptId = D.Id
```



Join Type	Purpose
Cross Join	Returns Cartesian product of the tables involved in the join
Inner Join	Returns only the matching rows. Non matching rows are eliminated.
Left Join	Returns all the matching rows + non matching rows from the left table
Right Join	Returns all the matching rows + non matching rows from the right table
Full Join	Returns all rows from both tables, including the non-matching rows.

7 comments:

vikul October 4, 2013 at 4:07 AM

I have a doubt.... In which case we use cross join??

Reply

Replies

Anonymous April 7, 2015 at 1:10 PM

Consider you have Car table which holds model information Car(Make, Model) and AvaliableColorOption(Color)
All available car options can be achieved by cross join..
Car table:
1. Benz C-Class
2. Benz S-Class

AvaliableColorOption:
1. Red
2. Green

Cartesian Product of the tables will yield:
1. Benz C-Class Red
2. Benz S-Class Red
3. Benz C-Class Green
4. Benz S-Class Green

This may not be the best example, but it just came to my mind on fly just wanted to throw it out there to check if it helps.

Anonymous December 16, 2015 at 3:22 PM

@Vikul, To add a one more point. I read somewhere, If you want to have a huge test data . you can use cross join.

Reply

Kaya Kayıkçı December 12, 2013 at 8:46 AM

Hello,
Thanks for videos.

Could you please show how can we join three tables (e.g. table1 (Name, CityID), table2 (CityID, CityName, TerritoryID), table3 (TerritoryID, TerritoryName) we will show Name and TerritoryName)
Thank you in advance.

Reply

sandy sammy January 9, 2014 at 1:49 PM

HI,
Try this :-

```
Select dbo.Table1.Name,dbo.Table3.TerritoryName
from dbo.Table1
join Table2
on dbo.Table2.CityID = dbo.Table1.CityId
join Table3
on dbo.Table2.TerritoryId = dbo.Table3.TerritoryId
```

Reply

parag January 27, 2014 at 8:00 PM

Could you please explain how to use Cross apply and outer apply , In which scenario we can use these two. Thank you.

Reply

Unknown March 28, 2019 at 5:58 AM

Hi sir Can u make power bi and DAX videos

Reply

Enter your comment...

Comment as: ahm7dkhalifa@ ▼ Sign out

Publish Preview □ Notify me

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.

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)