**In this video we will discuss**  
1. Intersect operator in sql server  
2. Difference between intersect and inner join   
  
   
  
**Intersect operator retrieves the common records from both the left and the right query of the Intersect operator.**

* Introduced in SQL Server 2005
* The number and the order of the columns must be same in both the queries
* The data types must be same or at least compatible

Let us understand INTERSECT operator with an example.   
  
We will use the following 2 tables for this example.

|  |  |
| --- | --- |
| http://4.bp.blogspot.com/-Mc7Xw7RolFE/VexzNsl9n-I/AAAAAAAAd7A/IvZRCYKICDs/s1600/Table%2BA.png | http://4.bp.blogspot.com/-rF2mc6VNMRM/VexzXRo989I/AAAAAAAAd7I/rFs99jziFg0/s1600/Table%2BB.png |

SQL Script to create the tables and populate with test data

Create Table TableA

(

    Id int,

    Name nvarchar(50),

    Gender nvarchar(10)

)

Go

Insert into TableA values (1, 'Mark', 'Male')

Insert into TableA values (2, 'Mary', 'Female')

Insert into TableA values (3, 'Steve', 'Male')

Go

Create Table TableB

(

    Id int,

    Name nvarchar(50),

    Gender nvarchar(10)

)

Go

Insert into TableB values (2, 'Mary', 'Female')

Insert into TableB values (3, 'Steve', 'Male')

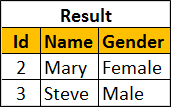
Go

The following query retrieves the common records from both the left and the right query of the Intersect operator. 

Select Id, Name, Gender from TableA

Intersect

Select Id, Name, Gender from TableB

**Result :**   
   
  
We can also achieve the same thinkg using INNER join. The following INNER join query would produce the exact same result. 

Select TableA.Id, TableA.Name, TableA.Gender

From TableA Inner Join TableB

On TableA.Id = TableB.Id

**What is the difference between INTERSECT and INNER JOIN**  
1. INTERSECT filters duplicates and returns only DISTINCT rows that are common between the LEFT and Right Query, where as INNER JOIN does not filter the duplicates.   
  
To understand this difference, insert the following row into TableA

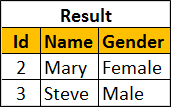
Insert into TableA values (2, 'Mary', 'Female')

Now execute the following INTERSECT query. Notice that we get only the DISTINCT rows 

Select Id, Name, Gender from TableA

Intersect

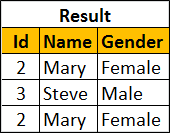
Select Id, Name, Gender from TableB

**Result :**  
   
  
Now execute the following INNER JOIN query. Notice that the duplicate rows are not filtered. 

Select TableA.Id, TableA.Name, TableA.Gender

From TableA Inner Join TableB

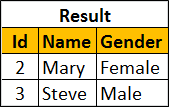
On TableA.Id = TableB.Id

**Result :**    
   
  
You can make the INNER JOIN behave like INTERSECT operator by using the DISTINCT operator 

Select DISTINCT TableA.Id, TableA.Name, TableA.Gender

From TableA Inner Join TableB

On TableA.Id = TableB.Id

**Result :**   
   
  
**2. INNER JOIN treats two NULLS as two different values**. So if you are joining two tables based on a nullable column and if both tables have NULLs in that joining column then, INNER JOIN will not include those rows in the result-set, where as INTERSECT treats two NULLs as a same value and it returns all matching rows.  
  
To understand this difference, execute the following 2 insert statements

Insert into TableA values(NULL, 'Pam', 'Female')

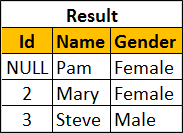
Insert into TableB values(NULL, 'Pam', 'Female')

**INTERSECT query**

Select Id, Name, Gender from TableA

Intersect

Select Id, Name, Gender from TableB

**Result :**   
   
  
**INNER JOIN query**

Select TableA.Id, TableA.Name, TableA.Gender

From TableA Inner Join TableB

On TableA.Id = TableB.Id

**Result :**   
