**What is Index?**

Indexes are special lookup tables that the database search engine can use to speed up data retrieval. Simply put, an index is a pointer to data in a table. An index in a database is very similar to an index in the back of a book.

**1) Clustered Index**

**2) Non- Clustered Index**

**1) Clustered Index :-**

A Clustered Index in SQL Server defines the order in which data physically stored in a table. It means the SQL cluster index will sort the records first and then store them.

Generally, when you create Primary Key, the Clustered index automatically created by that primary key. However, you can explicitly create a clustered index in SQL Server using the CREATE CLUSTER INDEX statement. Let us see how to create and delete a clustered index in SQL Server using Transact Query, and SQL Server Management Studio

**How to create Clustered Index in SQL Server**

When you create Primary Key Constraint on a column, it will automatically create Clustered Index on that column for you. It only happens if there is no existing Cluster in a table. Remember, You can create only One SQL Server Cluster Index per Table.

In this example, we will show you how to create a Clustered Index in SQL using the Transact SQL statement. Here we will create a [Primary Key](https://www.tutorialgateway.org/sql-primary-key/) at the time of table creation. Refer [Create Table](https://www.tutorialgateway.org/sql-create-table/) article.

-- Create Clustered Index in SQL Server

**USE [SQL Tutorial]**

**GO**

**CREATE TABLE [CustomerRecord]**

**(**

**[CustomerKey] [int] NOT NULL PRIMARY KEY,**

**[FirstName] [varchar](50) NOT NULL ,**

**[LastName] [varchar](50) NULL,**

**[EmailAddress] [nvarchar](50) NULL,**

**[Profession] [nvarchar](100) NULL,**

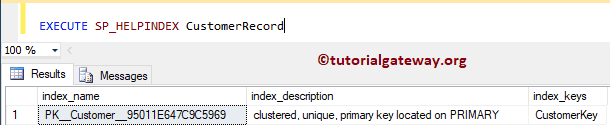
**[Yearly Income] [money] NULL**

**)**

Let me use the sp\_helpindex stored procedure to check the SQL index on CustomerRecord table.

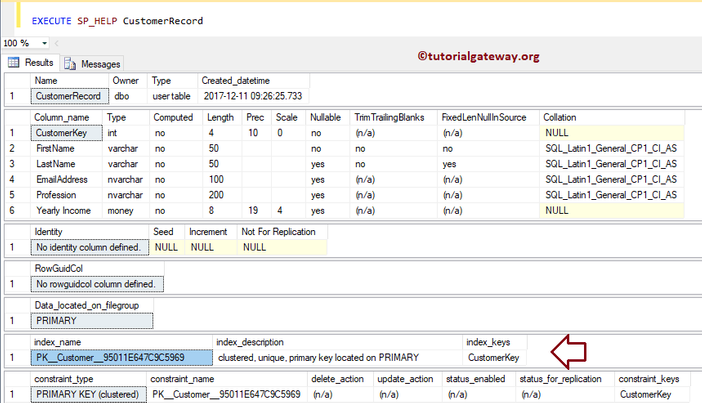
**EXECUTE SP\_HELPINDEX CustomerRecord**

As you can see, the primary key had created a Clustered and Unique index on the Customer Key column.

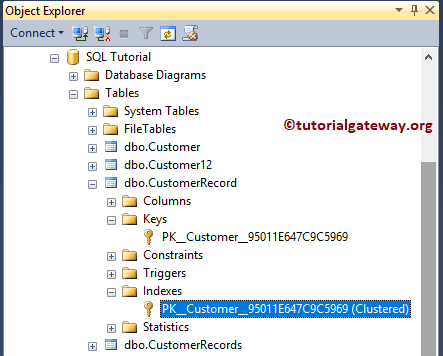


You can also use SP\_HELP stored procedure to check the same.

**EXECUTE SP\_HELP CustomerRecord**



Or you can expand the table on which we created the SQL Server Clustered Index, and expand the Indexes folder



Insert rows into Clustered Index Column

Let me [insert](https://www.tutorialgateway.org/sql-insert-statement/) a few rows to check the SQL cluster Index functionality. As you can see, we are inserting five records into the table, and the Customer Key values are not in any proper order.

USE [SQL Tutorial]

GO

INSERT INTO [dbo].[CustomerRecord]

VALUES (4, 'Imran', 'Khan', 'abc@abc.com', 'Skilled Professional', 15900)

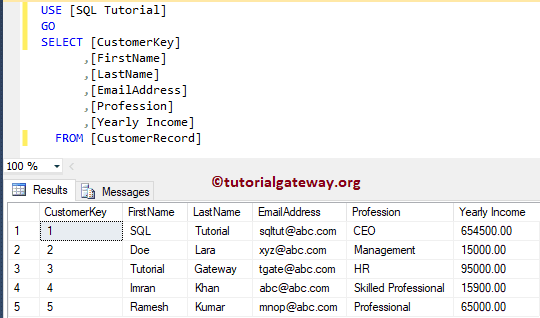
,(2, 'Doe', 'Lara', 'xyz@abc.com', 'Management', 15000)

,(5, 'Ramesh', 'Kumar', 'mnop@abc.com', 'Professional', 65000)

,(1, 'SQL', 'Tutorial', 'sqltut@abc.com', 'CEO', 654500)

,(3, 'Tutorial', 'Gateway', 'tgate@abc.com', 'HR', 95000)

Let me [Select](https://www.tutorialgateway.org/sql-select-statement/) the records that we inserted before. From the below image, see that the records are sorted by Customer Key in [Ascending Order](https://www.tutorialgateway.org/sql-order-by-clause/). This is because, Clustered Index will reorder (rearrange) records.



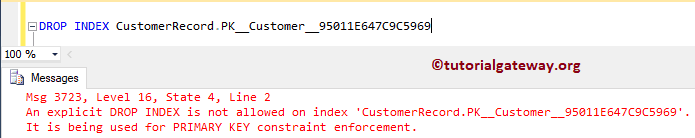
### Delete Clustered Index in SQL Server

Please use the DROP INDEX statement to delete or drop Clustered Index

-- DROP Clustered Index in SQL Server

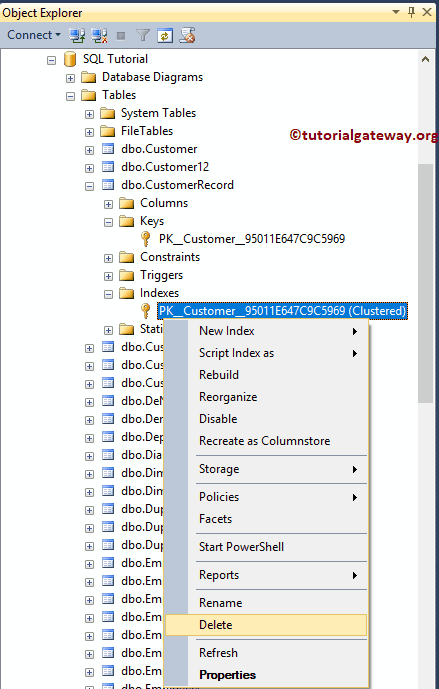
DROP INDEX CustomerRecord.PK\_\_Customer\_\_95011E647C9C5969

Here, our clustered Index created automatically by the primary key, so we can’t simply delete using this statement. But for explicitly created Clustered Index, this will work.

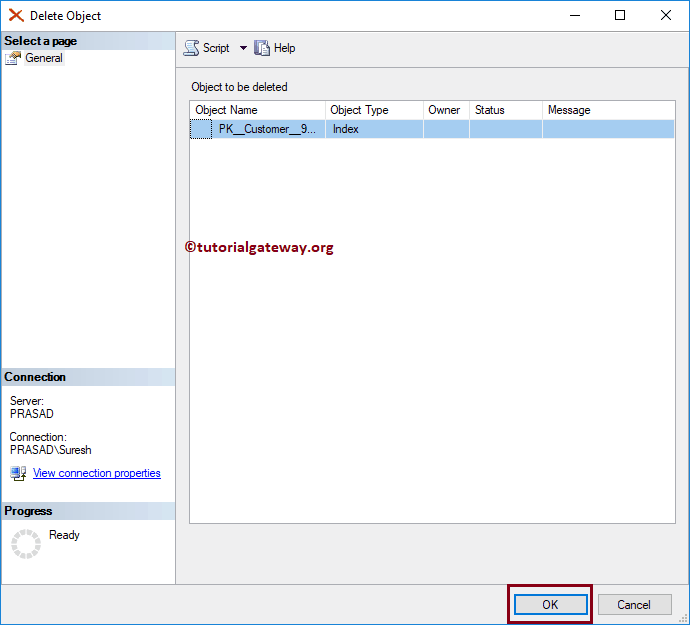


Use SSMS to Delete Clustered Index in SQL Server

Within the object explorer, Expand the Database folder and select the table on which your Clustered Index present. Next, expand the Indexes folder, and right-click on the key name to select the Delete option



Once you choose the Delete option, Delete Object window will open. Click OK to delete the Index.



### Create SQL Clustered Index on Existing table

In this instance, we will show you how to add Clustered Index on the existing table. And the syntax is:

-- Create Clustered Indexes in SQL Server

CREATE CLUSTERED INDEX Index\_Name

ON Table\_Name (Column\_Name(s) ASC/DESC)

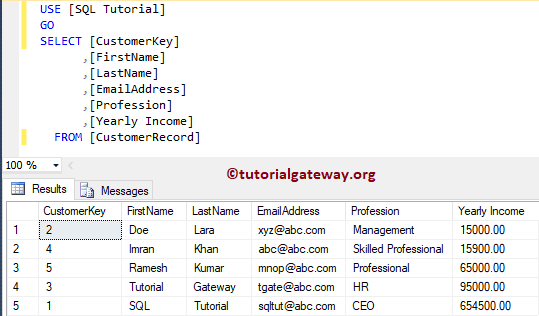
It will create Clustered Index on Yearly Income Column present in Customerrecord table

-- Create Clustered Indexes in SQL Server

CREATE CLUSTERED INDEX IX\_CustomerRecord\_YearlyIncome

ON CustomerRecord ([Yearly Income] ASC)

Now you can see that the Yearly Income column sorts the data in ascending order



This time we created the Clustered Index explicitly. So, let me try the DROP INDEX statement to delete this clustered index.

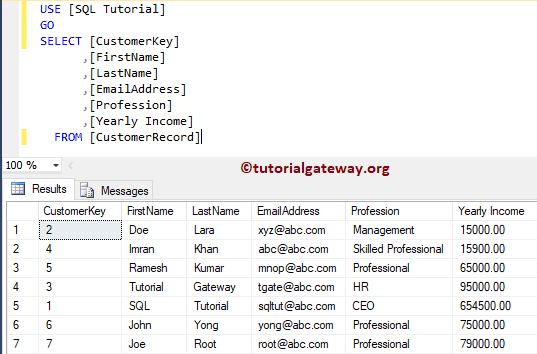
-- Delete Clustered Indexes in SQL Server

DROP INDEX CustomerRecord.IX\_CustomerRecord\_YearlyIncome

We successfully deleted the SQL Server clustered index using the Drop Index statement.

**Create Composite Clustered Index in SQL Server**

The [SQL](https://www.tutorialgateway.org/sql/) allows you to create clustered index on multiple columns (but not multiple clustered indexes). So, you can point the same clustered index to Multiple columns. To demonstrate the SQL composite clustered index, we added two more records to the Customer record table



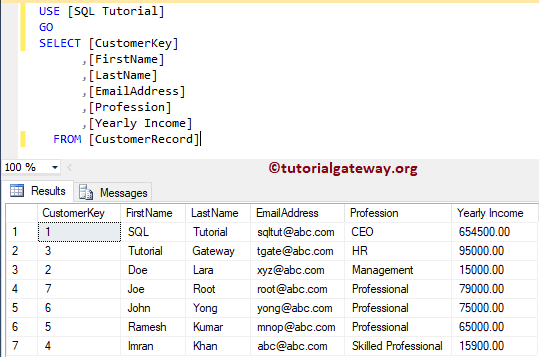
In this example, we are adding a clustered index on Professional column, and Yearly Income column

-- Create Composite Clustered Indexes in SQL Server

CREATE CLUSTERED INDEX IX\_CustomerRecord\_YearlyIncome

ON CustomerRecord ([Profession] ASC, [Yearly Income] DESC)

Now you can see, data is sorted by Professional in Ascending Order, and then by Yearly Income in descending order.



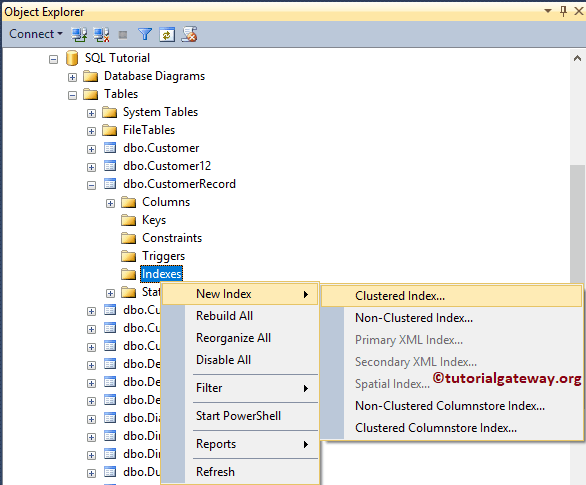
## Create Clustered Index in SQL Server Management Studio

This section covers the creation of clustered index using management studio

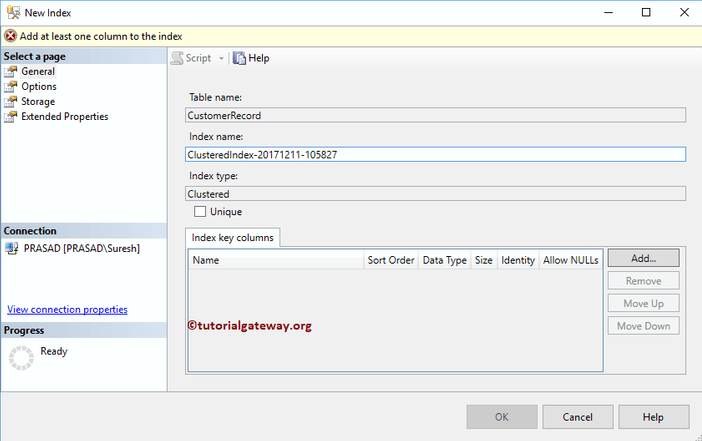
### Use Object Explorer to Create Clustered Index

In this example, we show you how to create a Clustered Index using the [Management Studio](https://www.tutorialgateway.org/sql-server-management-studio/). To do so, please go to the Object Explorer and expand the Database folder in which the table had.

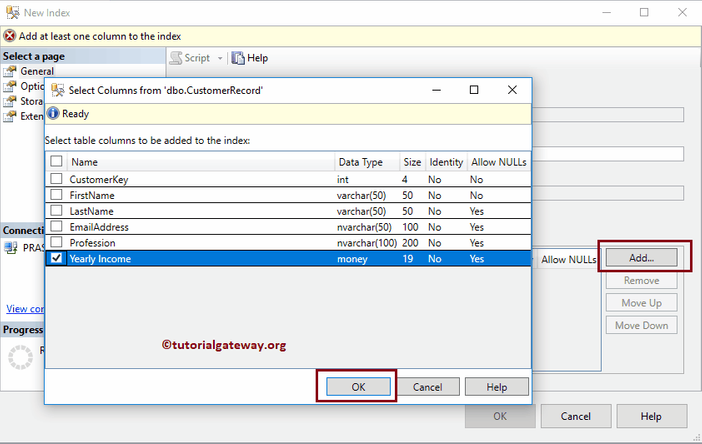
Please expand the table on which you want to create a Clustered Index, and Right-click on the Indexes folder will open the context menu. Please select New Index and then select the Clustered Index.. option



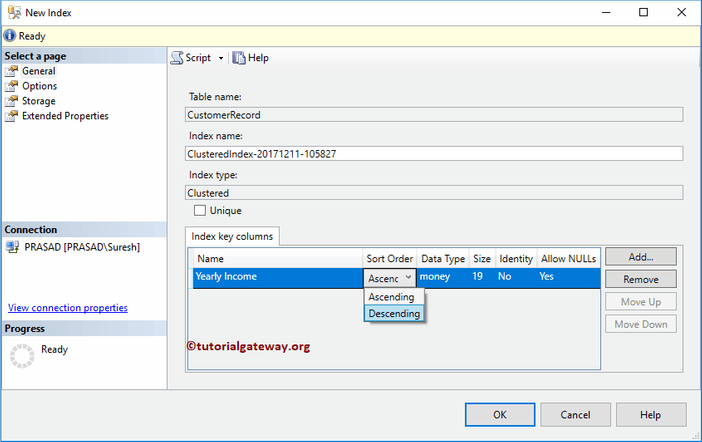
Once you choose the Clustered Index.. option, a new window called New Index will be opened. Please change the Index Name.



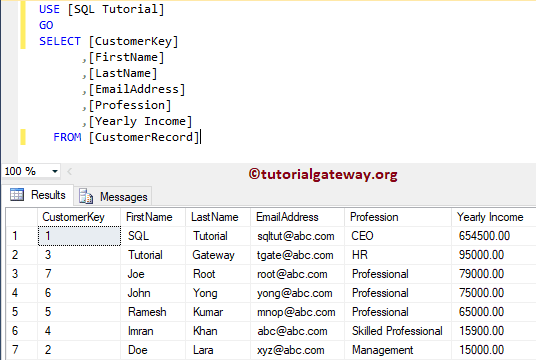
Click on the Add button, and select the column name on which you want to specify the SQL Server Clustered Index. For now, we are choosing the Yearly Income column.



Please change the sort order

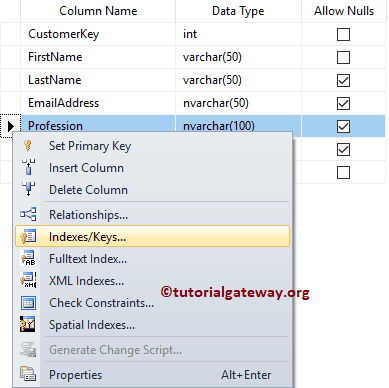


Now you can see that the table is sorted by Yearly Income in descending order

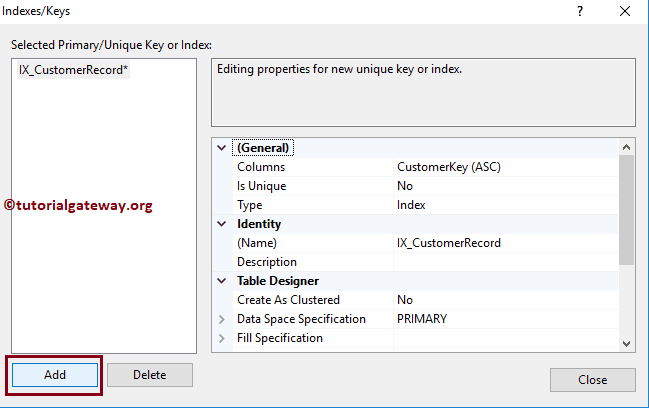


### Create Clustered Index using Table Designer

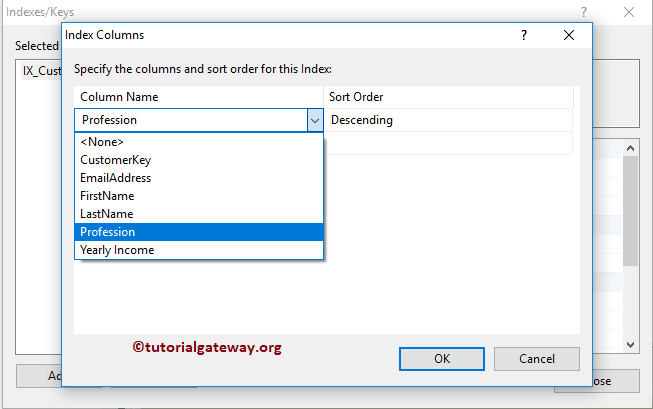
Please expand the table on which you want to create a Clustered Index in SQL Server Object Explorer. Next, Right-click on it and select the Design option to open the table in the Designer window. Next, Right-click on the column and pick the Set Indexes / Keys… option.



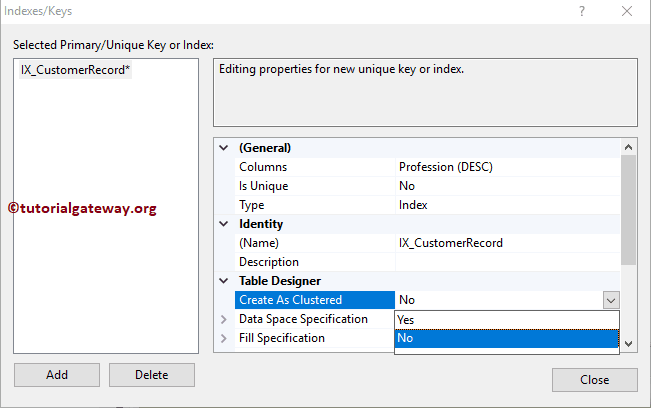
Please click on the Add button to add a Clustered Index. Once you select the Add option, SSMS will create an Index for you.



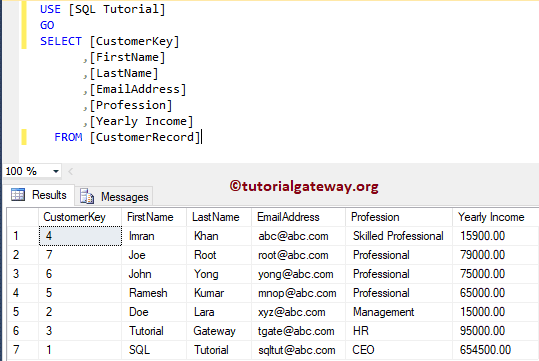
Next, click on the Browse button beside the Columns option to change the column. Next, we selected the Profession as the Column Name, and Descending is the Sort Order.



Under the Table Designer, Please change the Create As Clustered option from default No to Yes



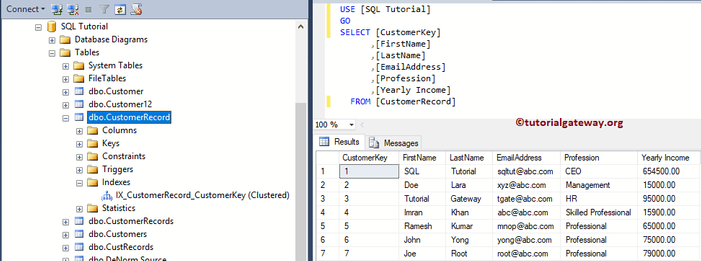
That’s it! We created the clustered index. Now you can see that the data is sorted by Income in descending order.



**2) Non- Clustered Index**

A non-clustered index doesn't sort the physical data inside the table. In fact, a non-clustered index is stored at one place and table data is stored in another place. This is similar to a textbook where the book content is located in one place and the index is located in another.

In this chapter, we will show you how to create and delete a non clustered index in SQL Server using Transact SQL Query, and SQL Server Management Studio



How many Non Clustered Indexes in SQL Server?

You can create 999 Non Clustered Indexes per Table.

**Create Non Clustered Index in SQL Server**

This example explains to you how to add Non Clustered Index to an existing table. And the syntax is:

-- Create Non Clustered Indexes in SQL Server

CREATE NONCLUSTERED INDEX Index\_Name

ON Table\_Name (Column\_Name(s) ASC/DESC)

It will create Non Clustered Index on Yearly Income Column present in the Customer record table

-- Create Non Clustered Indexes in SQL Server

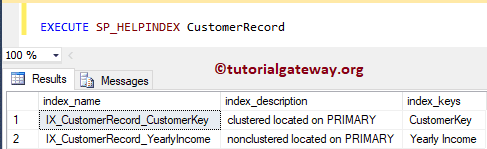
CREATE NONCLUSTERED INDEX IX\_CustomerRecord\_YearlyIncome

ON CustomerRecord ([Yearly Income] ASC)

Let me use the sp\_helpindex stored procedure to check the indexes on the Customer Record table.

EXECUTE SP\_HELPINDEX CustomerRecord

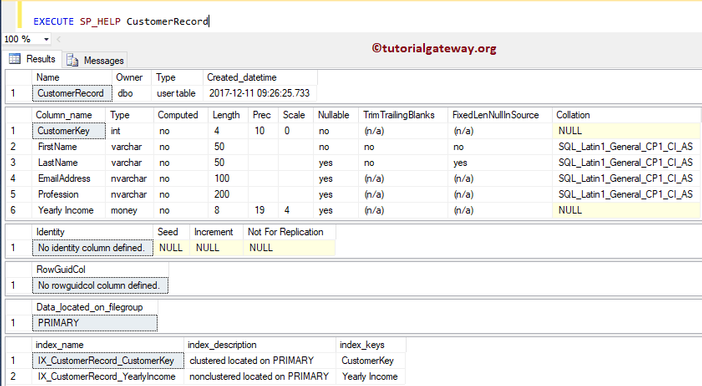
As you can see, the Yearly Income column has Non Clustered index.



You can also use [SP\_HELP](https://www.tutorialgateway.org/useful-system-stored-procedures-in-sql/) stored procedure in [SQL Server](https://www.tutorialgateway.org/sql/) to check the same.

EXECUTE SP\_HELP CustomerRecord

It displays all the information about the table.



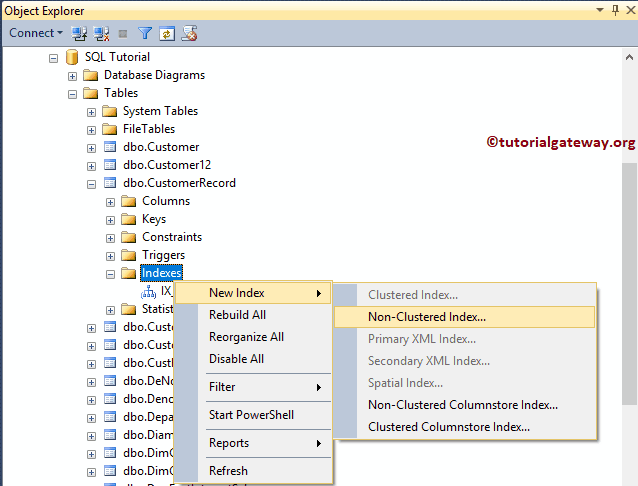
Create Non Clustered Index in SQL Server Management Studio

Let me show you, how to create a non clustered index, how to delete it using Management Studio

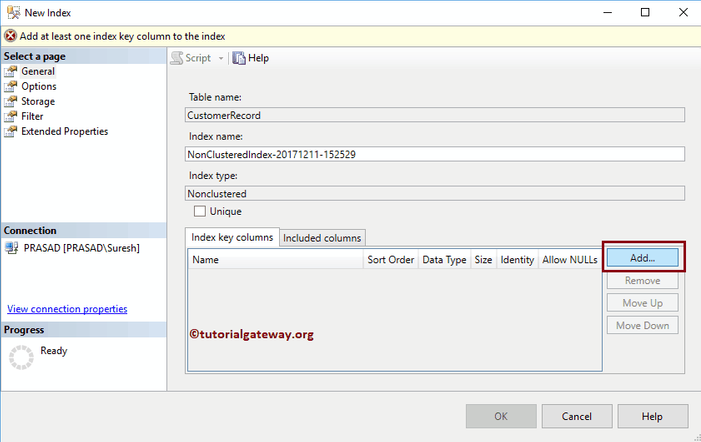
Use Object Explorer to Create Non Clustered Index

In this example, we explain how to create a Non Clustered Index using the [Management Studio](https://www.tutorialgateway.org/sql-server-management-studio/).

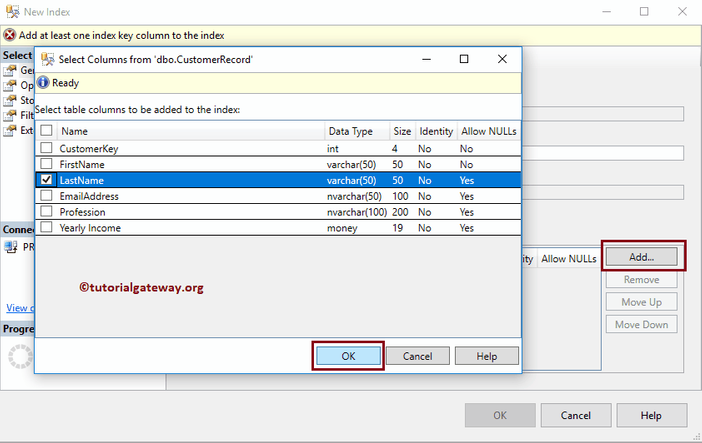
To do so, please go to the Object Explorer and expand the Database folder. Please expand the table to create a Non Clustered Index, and Right-click on the Indexes folder will open the context menu. Please select New Index and then select the Non-Clustered Index..



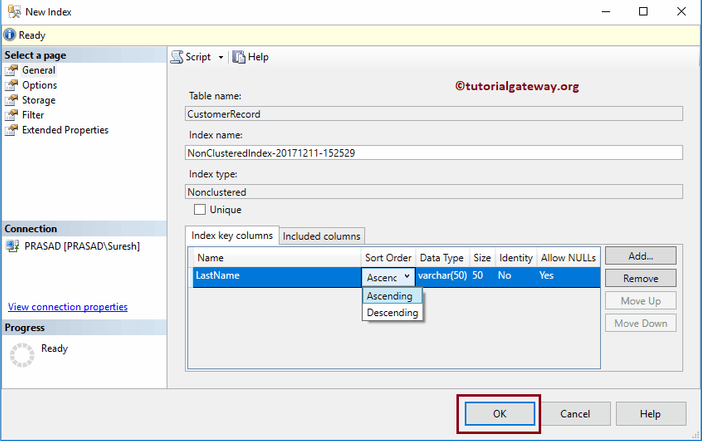
Once you choose the Non-Clustered Index.. option, a new window called New Index will be opened as we showed below. Please change the non clustered Index Name as per your requirements



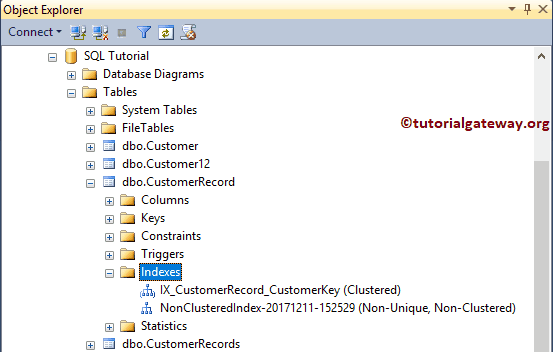
Click on the Add button, and select the column name on which you want to specify the Non Clustered Index. For now, we are selecting the Last Name column.



Please change the sort order.



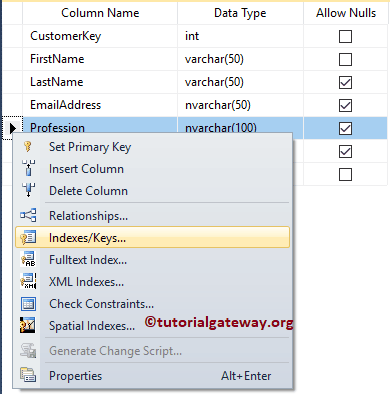
Now you can see the non clustered index



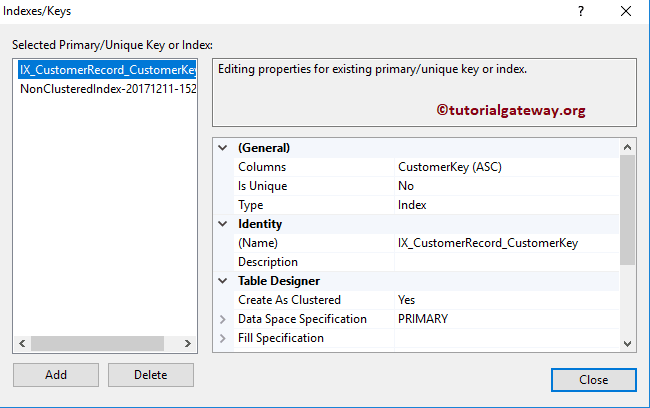
Create Non Clustered Index using Table Designer

Please extend the table on which you require to create a Non Clustered Index, and Right-click on it will open the context menu. Please choose the Design option to open the table in the Designer window.

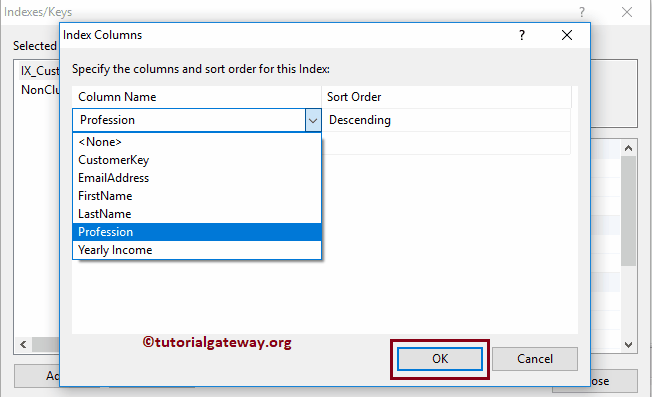
Next, Right-click on the column and select the Set Indexes / Keys… option.



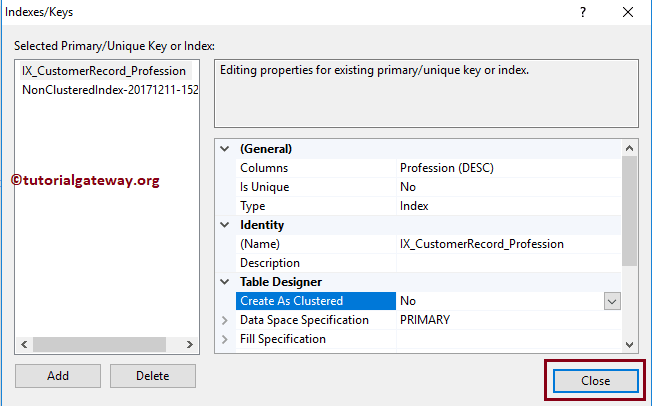
Please click on the Add button to add a Non Clustered Index. Once you select the Add option, SSMS will create an Index for you.



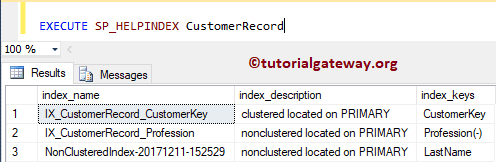
Next, click on the Browse button beside the Columns option to change the column. Next, we selected the Profession as the Column Name, and Descending is the Sort Order.



Under the Table Designer, Please change the Create As Clustered option to No



See the newly created non clustered index



Drop Non Clustered Index in SQL Server Example

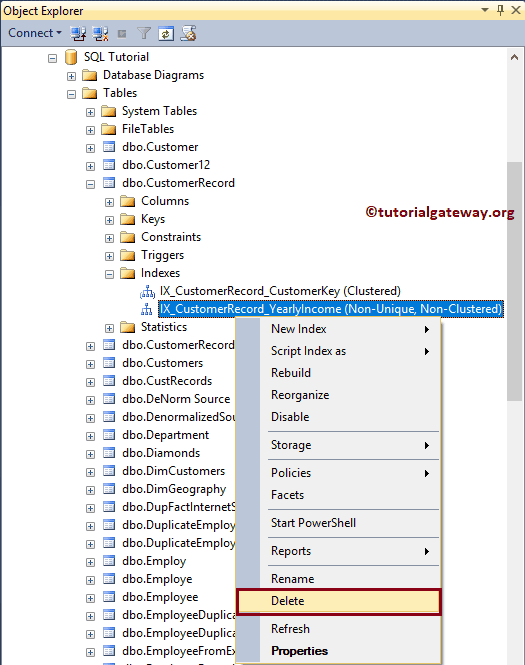
Please use the DROP INDEX statement to delete or drop Non Clustered Index in SQL Server

-- DROP Non Clustered Index in SQL Server

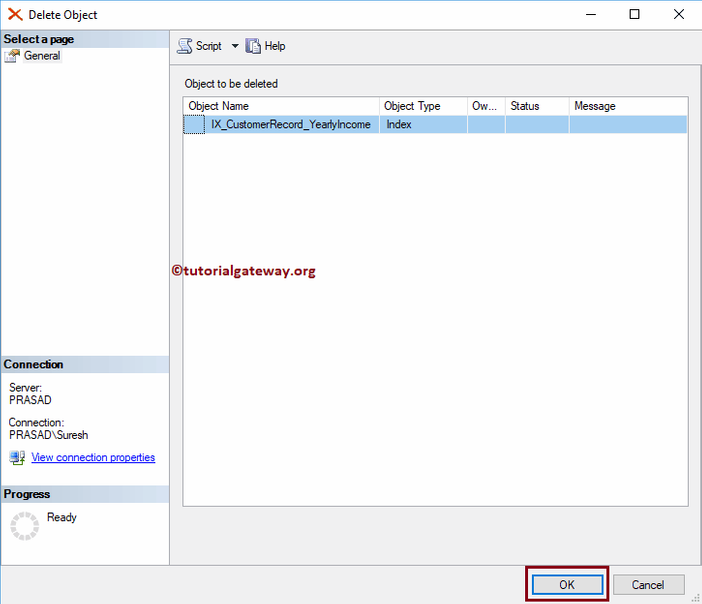
DROP INDEX CustomerRecord.IX\_CustomerRecord\_Profession

Use SSMS to Drop Non Clustered Index Example

Within the object explorer, Expand the Database folder and choose the table on which your Non Clustered Index present. Next, expand the Indexes folder, and right-click on the key name will open the context menu. Please select the Delete option



Once you select the Delete option, Delete Object window opened. Click OK to delete the Non clustered Index in SQL Server.



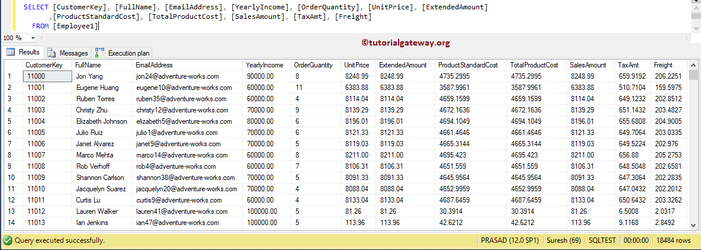
**Filtered Indexes :-**

Filtered Index is a new feature introduced in SQL Server 2008. A Filtered Index in SQL Server is nothing but a non clustered Indexes along with Where Clause. Because of the where clause, indexing will perform on a portion of records (records match where condition).

Before we go into examples of creating and deleting Filtered Indexes in SQL Server, please remember these points

* SQL Filtered Indexes will significantly improve the query performance because it is smaller than the [Non-Clustered Index](https://www.tutorialgateway.org/non-clustered-index-in-sql-server/) (works on Full table)
* The maintenance cost of the Filtered Indexes is very less because it is small.
* SQL Filtered Index will reduce the Disk storage space. So, whenever possible try Filtered Indexes

We created three tables Employee1, Employee2, and Employee3 tables with below-shown [SQL Server](https://www.tutorialgateway.org/sql/) data:



If you want to use the same data that we used in this filtered index example, then use below query

USE [AdventureWorksDW2014]

GO

SELECT Cust.CustomerKey,

Cust.[FirstName] + ' ' + Cust.[LastName] AS [FullName]

,Cust.[EmailAddress]

,Cust.[YearlyIncome]

,SUM([OrderQuantity]) AS [OrderQuantity]

,SUM([UnitPrice]) AS [UnitPrice]

,SUM([ExtendedAmount]) AS [ExtendedAmount]

,SUM([ProductStandardCost]) AS [ProductStandardCost]

,SUM([TotalProductCost]) AS [TotalProductCost]

,SUM([SalesAmount]) AS [SalesAmount]

,SUM([TaxAmt]) AS [TaxAmt]

,SUM([Freight]) AS [Freight]

FROM DimCustomer AS Cust

JOIN [FactInternetSales] on

Cust.CustomerKey = FactInternetSales.CustomerKey

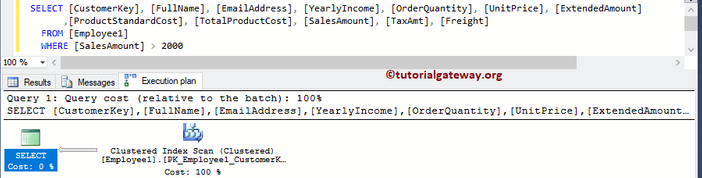
GROUP BY Cust.CustomerKey, Cust.[FirstName], Cust.[LastName]

,Cust.[EmailAddress], Cust.[YearlyIncome]

ORDER BY Cust.CustomerKey

## Create Filtered Indexes in SQL Server

Before we start creating a SQL Server Filtered Index, let me show you the Execution plan of the Query. Below statement will select all the records from Employee1 table



### Create Non Clustered Index in SQL Server

Before we start creating the Filtered Index, let me create a basic Non Clustered Index on the Sales Amount column. We already explained the creation of the [Non Clustered Index](https://www.tutorialgateway.org/non-clustered-index-in-sql-server/) in our previous article. So, please refer to the same.

The below statement will create a SQL Server Non Clustered Index on Sales Amount Column present in Employee2 table

-- Create Non Clustered Indexes in SQL Server

CREATE NONCLUSTERED INDEX IX\_Employee2\_SalesAmount

ON Employee2 (SalesAmount)

Now, we will compare the performance of a query without Non Clustered Index (Employee 1) and With Non Clustered Index (Employee 2). The below [Where clause](https://www.tutorialgateway.org/sql-where-clause/) statement selects Customer Key and Sales Amount from Employee1 and Employee2 table

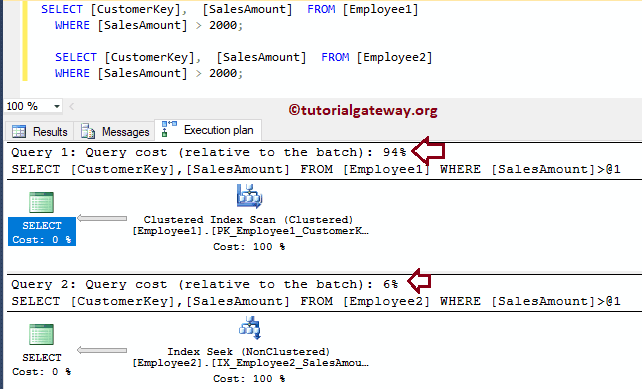
SELECT [CustomerKey], [SalesAmount] FROM [Employee1]

WHERE [SalesAmount] > 2000;

SELECT [CustomerKey], [SalesAmount] FROM [Employee2]

WHERE [SalesAmount] > 2000;

If you observe the execution plan, there is a significant difference in the performance after we created a Non Clustered Index



### Create a Filtered Index

In this example, we will show you how to add Filtered Indexes to an existing table. And the syntax is:

-- Create Non Clustered Indexes in SQL Server

CREATE NONCLUSTERED INDEX Index\_Name

ON Table\_Name (Column\_Name(s) ASC/DESC)

WHERE Some\_Condition

The below statement will create a Filtered Index in Sql Server on Sales Amount Column present in Employee 3 table

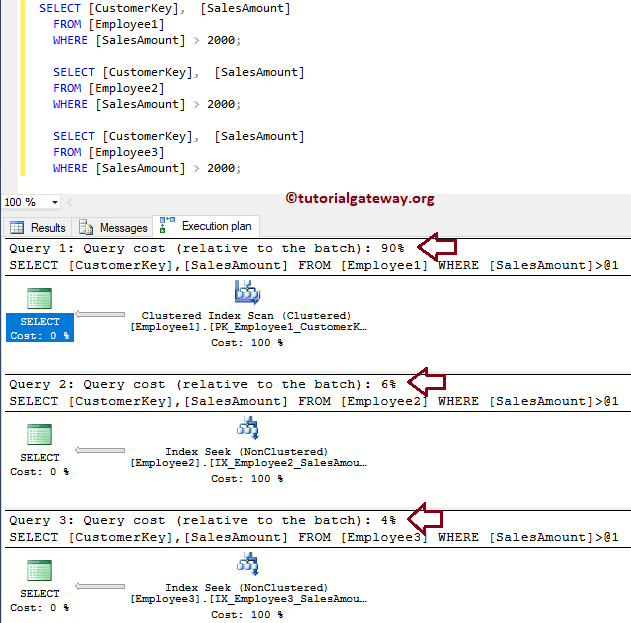
-- Create Non Clustered Indexes in SQL Server

CREATE NONCLUSTERED INDEX IX\_Employee3\_SalesAmount

ON Employee3 (SalesAmount)

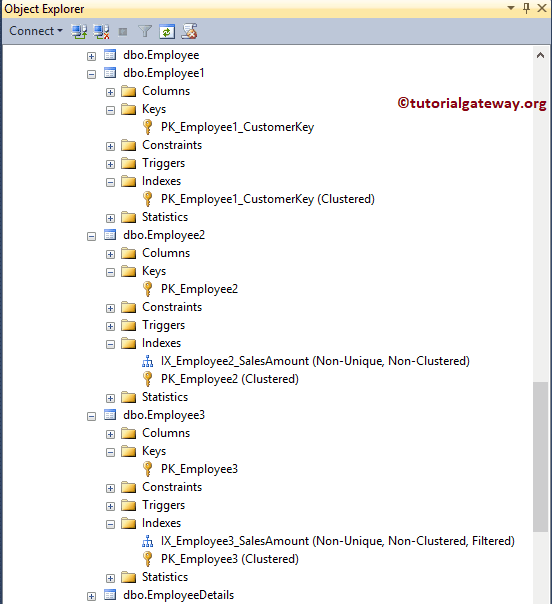
WHERE [SalesAmount] > 1000

This time, we will compare the performance of a query without Non Clustered Index (Employee 1), With Non Clustered Index (Employee 2), and With Filtered Index. Below statement will select Customer Key, and Sales Amount from Employee1, Employee2, and Employee3 table



If you observe the execution plan in the above screenshot, there is a significant difference in the performance after we created a Filtered Index in SQL.

The following screenshot will show you all the indexes we created until now.



### Create Composite Filtered Index in SQL Server

[SQL](https://www.tutorialgateway.org/sql/) allows you to create a Filtered index on multiple columns called composite indexes. So, you can point the same index to Multiple columns. In this example, we are adding Filtered indexes on the Sales Amount column, and Yearly Income column

CREATE NONCLUSTERED INDEX IX\_Employee4\_SalesAmount

ON Employee4 ([YearlyIncome], SalesAmount)

WHERE [SalesAmount] > 1000

Messages

-------

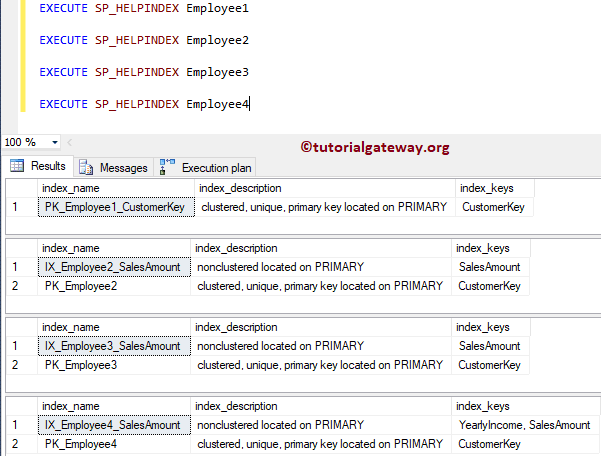
Command(s) completed successfully.

### How to find Information about Filtered Index?

You can use the sp\_helpindex stored procedure to get the information about all the indexes present in a table.

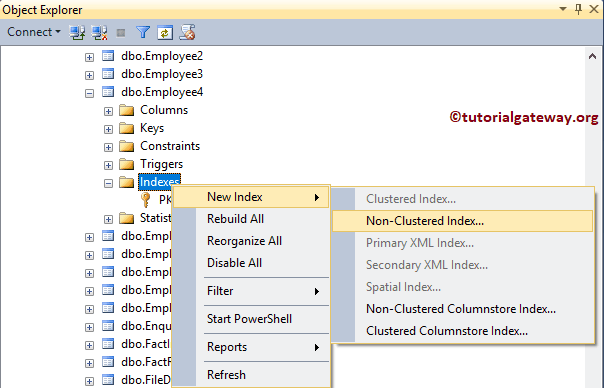
EXECUTE SP\_HELPINDEX Table\_Name

Let me use the sp\_helpindex stored procedure to check the indexes on Employee1, Employee2, Employee3, and Employee4 tables.

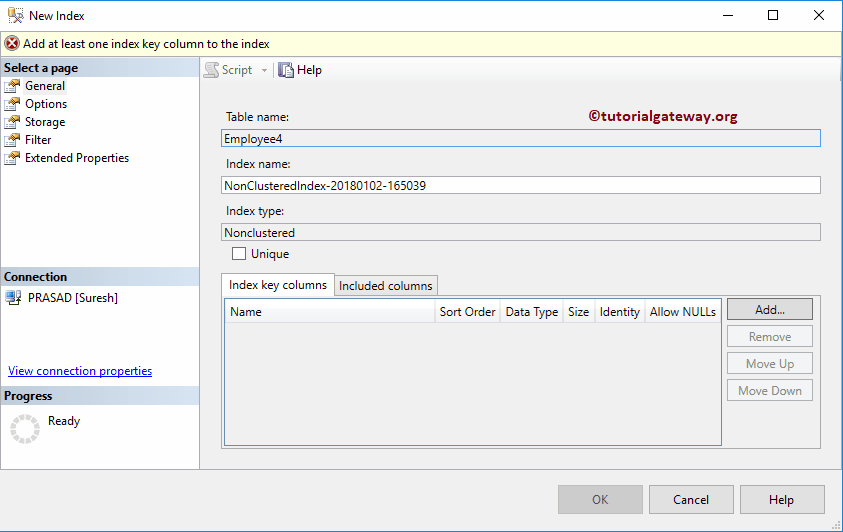


## Create Filtered Indexes using SSMS

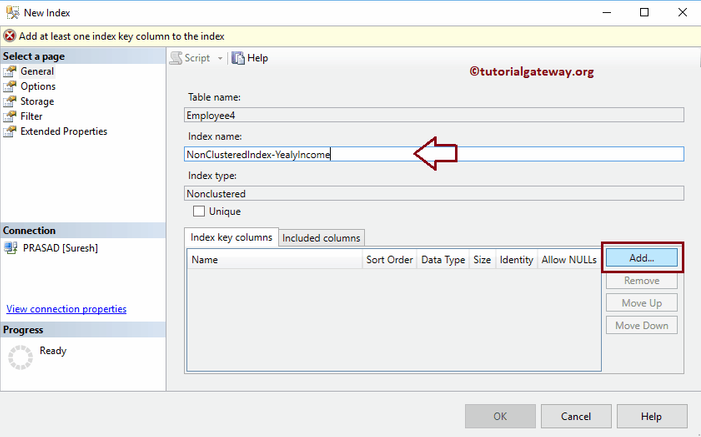
In this sample, we show you how to create SQL Filtered Indexes using the [Management Studio](https://www.tutorialgateway.org/sql-server-management-studio/). To do so, Please expand the table on which you want to create a Filtered Index, and Right-click on the Indexes folder will open the context menu. Next, please select New Index and then select the Non-Clustered Index.. option



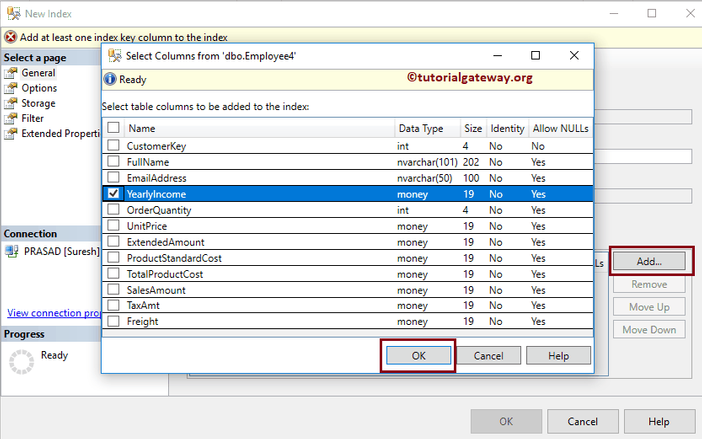
Once you choose the Non-Clustered Index.. option, a new window called New Index will open.



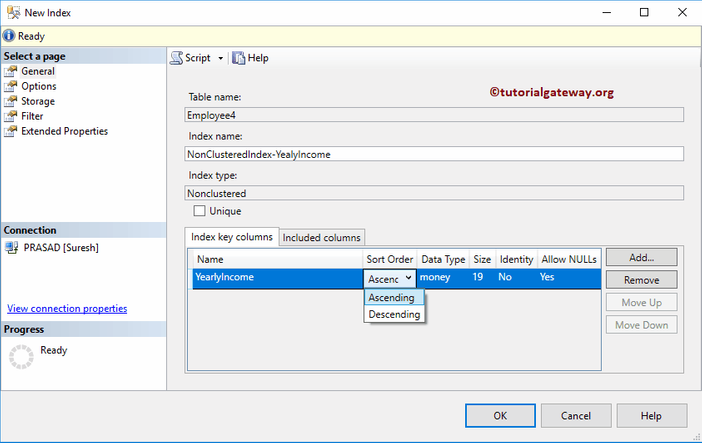
Please change the Index Name as per your requirements. Click on the Add button to select columns



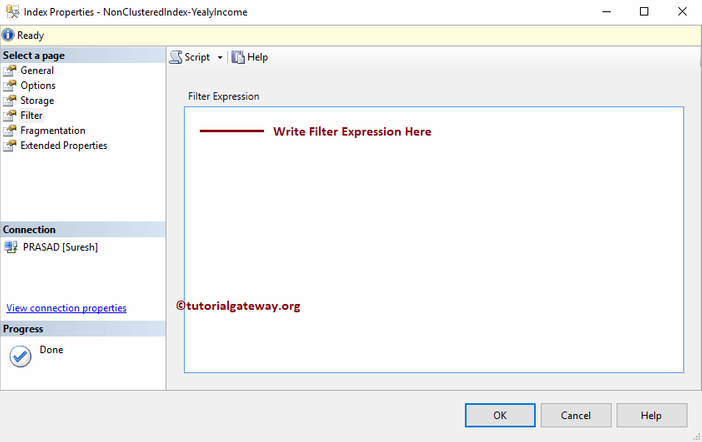
Next, select the column name on which you want to specify the Filtered Index. For now, we are choosing the Yearly Income column.



Please change the sort order.



Next, go to the Filter page and write your filter condition



## Delete Filtered Indexes in SQL Server

Please use the DROP INDEX statement to delete or drop Non Clustered Filtered Indexes in SQL Server

-- DROP Filtered Index in SQL Server

DROP INDEX Employee3.IX\_Employee3\_SalesAmount

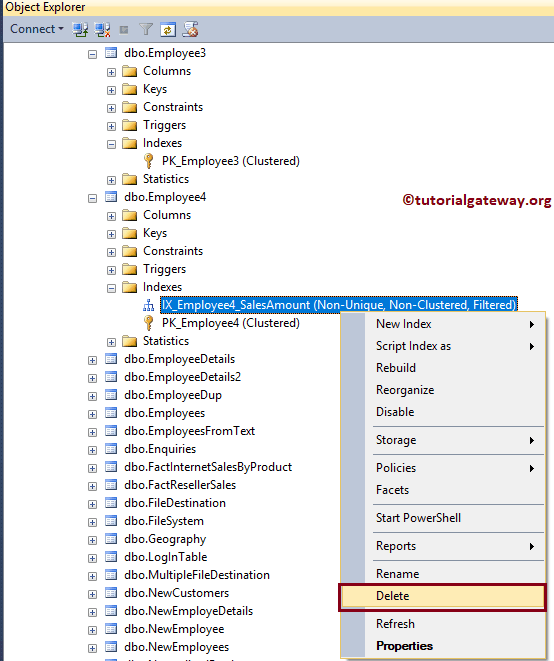
Messages

--------

Command(s) completed successfully.

### Use SSMS to Delete Filtered Index

To delete filtered Index in SQL Server, select the table on which your Filtered Index present. Next, expand the Indexes folder, and right-click on the key name, and choose the Delete option



Once you choose the Delete option, Delete Object window will open. Click OK to delete the Index.

