

How to create a SQL Server index?

August 17, 2020 [SQL Server Index](#)

Original article from <https://expert-only.net/sql-server/index/create-sql-server-index/>



Create a SQL Server index to improve the performance of queries on tables and clustered views. To create an index use this example and adapt it to your needs. Indexes are the first recommended step to performance optimization in SQL Server.

How to create a simple SQL Server index on a table?

Finally, let's consider that the Analytics system the main columns to filter the data are the year and the month name: [Year] and [MonthName].

The How-to below creates an index on the [sales table created here](#).

Time Needed : 03 minutes

How to create a SQL Server index on existing table?

1. Create the sales table example

As a reminder, the sales tables has this initial structure. Use the following SQL code to create the sales table as an example for the index.

```
CREATE TABLE [dbo].[SALES] (  
[Year] TINYINT,  
[MonthName] NVARCHAR(50),  
[MonthCurrent] BIT,  
[NumberMonth] TINYINT,  
[EmployeeNumber] SMALLINT,  
[NumberOfClients] INTEGER,  
[NumberOfSales] BIGINT,  
[Amount_ET] NUMERIC(15,5),  
[Amount_IT] DECIMAL(15,5)  
);
```

<https://expert-only.net/sql-server/index/create-sql-server-index/>

Article from [Expert-Only.net](https://expert-only.net)

```
SQLQuery1.sql - 1...ESTLE\YEssabb (70))* -> X
1 CREATE TABLE [dbo].[SALES] (
2     [Year] TINYINT,
3     [MonthName] NVARCHAR(50),
4     [MonthCurrent] BIT,
5     [NumberMonth] TINYINT,
6     [EmployeeNumber] SMALLINT,
7     [NumberOfClients] INTEGER,
8     [NumberOfSales] BIGINT,
9     [Amount_ET] NUMERIC(15,5),
10    [Amount_IT] DECIMAL(15,5)
11 );
```

142 %

Messages

Commands completed successfully.

Completion time: 2020-08-27T20:55:32.7447183+02:00

2. Create the index on the sales table

Finally, let's consider that on our Analytics system the main columns to filter the data are the year and the month name: [Year] and [MonthName].

```
CREATE INDEX indexYearMonth
ON [dbo].[SALES] ([Year], [MonthName]);
```

```
SQLQuery1.sql - 1...ESTLE\YEssabb (70))* -> X
1 CREATE INDEX indexYearMonth
2 ON [dbo].[SALES] ([Year], [MonthName]);
```

142 %

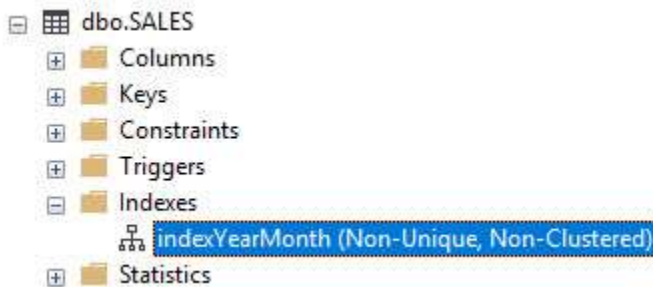
Messages

Commands completed successfully.

Completion time: 2020-08-27T20:57:57.7914367+02:00

3. Check the index under the table in SSMS

Expand the sales table properties and check the index presence. Note that per default the index is a non-unique and non-clustered index.



Tools

- Use the SQL Server Management Studio (SSMS) free Microsoft software to perform the steps and execute the queries.

Below both T-SQL examples are formatted and ready to copy and paste, adjust and execute in SSMS.

```
CREATE TABLE [dbo].[SALES]
(
[Year] TINYINT,
[MonthName] NVARCHAR(50),
[MonthCurrent] BIT,
[NumberMonth] TINYINT,
[EmployeeNumber] SMALLINT,
[NumberOfClients] INTEGER,
[NumberOfSales] BIGINT,
[Amount_ET] NUMERIC(15,5),
[Amount_IT] DECIMAL(15,5)
);
```

```
CREATE INDEX indexYearMonth
ON [dbo].[SALES] ([Year], [MonthName]);
```

The result is in any query using the Year and the Month, the newly created index called indexYearMonth will be used and it improve the performance of the query.

The next step is to analyze and **rebuild the index** in order to have organized values inside the index and statistics.