How to create a SQL Server index?

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 the project requirements. Indexes are the first recommended step to performance optimization in SQL Server.

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

First, the How-to section below creates an index on the **sales table created here**.

Let's explain the context of the example. Firstly, it's a very basic example to show the CREATE INDEX T-SQL command and how easy it is to optimize performance. Secondly the 2 indexed columns are the typical ones used in large queries execution, i.e. the Year and the month. Another very common one is the date.

Why the large sales tables need to use some date related columns as indexed columns ?

Let's consider any business, the managers need to see the figures in a context to know how good the sales are. To achieve this the time is the best dimension, because the time comparisons are the most common and because the business financial managers uses fiscal year as references.

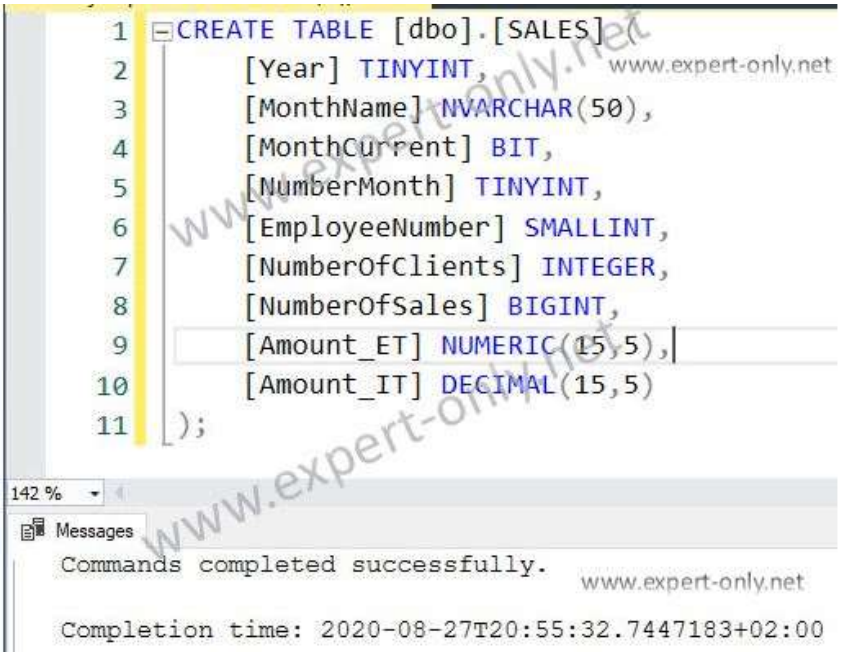
Popular views are the yearly, quarterly, monthly and daily ones. and as the data evolves along the fiscal year, **indexes** evolves and follow the date.

How to create a SQL Server index on existing table?

Step 1 : Create the sales table example

For instance, the sales tables have this initial structure. Use the following SQL code to create the sales table as an example for the index creation.

```
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)
);
```



Step 2 : Create the index on the sales table

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

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

Step 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.

Notes:

- Use an existing table to create the index or create a new one that needs to be indexed.
- Use the SQL Server Management Studio (SSMS) free Microsoft software to perform the steps and execute the queries.

Moreover, check the below T-SQL examples. Hence, the code is ready to copy and paste, so simply adjust it and execute in SQL Server Management Studio.

As a result, any query using the Year and the Month in the where clause uses the newly created index. In other words, the indexYearMonth index improves the performance of the query. The index maintenance like **reorganize** and rebuild commands keeps them effective.

To go further, the next step is to analyze and **rebuild the index** in order to organize values inside the index and statistics.