

Créer un index SQL Server

Catégorie : Transact-SQL



Créer un index dans SQL Server pour améliorer les performances des requêtes sur les tables et aussi dans les vues cluster. Il existe deux types d'index, l'index cluster et l'index non cluster. Pour ajouter un index non cluster à une table, utiliser cet exemple et adapter-le aux besoins du projet. Les index sont la première étape recommandée pour l'optimisation des performances dans SQL Server.

Expliquons le contexte de l'exemple. Tout d'abord, c'est un exemple très basique pour montrer la commande T-SQL CREATE INDEX et comment il est facile d'optimiser les performances. Deuxièmement, les deux colonnes indexées sont les colonnes typiques utilisées dans l'exécution de grandes requêtes, c'est-à-dire l'année et le mois. Une autre colonne très courante est la date.

Comment créer un index SQL Server sur une table ?

Tout d'abord, la première section explique comment créer un index sur une table des ventes.

Pourquoi les très grandes tables (comme les ventes) ont besoin de colonnes indexées liées à la date ?

Prenons l'exemple d'une entreprise, les responsables ont besoin de voir les chiffres dans un contexte pour savoir si les ventes sont bonnes. Pour cela, le temps est la meilleure dimension, parce que les comparaisons temporelles sont les plus courantes et parce que les responsables financiers des entreprises utilisent l'année fiscale comme référence.

Les vues les plus populaires sont les vues annuelles, trimestrielles, mensuelles et quotidiennes. Et comme les données évoluent le long de l'année fiscale, les [index](#) évoluent et suivent la date.



Ajouter un index MS SQL sur une table existante

Etape 1 : Créer l'exemple de la table des ventes

Par exemple, les tables de ventes ont cette structure initiale. Utilisez le code SQL suivant pour créer la table des ventes comme exemple pour la création de l'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)
);
```

```
expert-only.com
SQLQuery4.sql - loc...ginExpertOnly (67)*
1 CREATE TABLE [dbo].[SALES]
2 (
3 [Year] TINYINT,
4 [MonthName] NVARCHAR(50),
5 [MonthCurrent] BIT,
6 [NumberMonth] TINYINT,
7 [EmployeeNumber] SMALLINT,
8 [NumberOfClients] INTEGER,
9 [NumberOfSales] BIGINT,
10 [Amount_ET] NUMERIC(15,5),
11 [Amount_IT] DECIMAL(15,5)
12;
13;
14;
180 %
Messages expert-only.com
Commands completed successfully.
Completion time: 2022-05-13T10:42:00.9165368+02:00
```

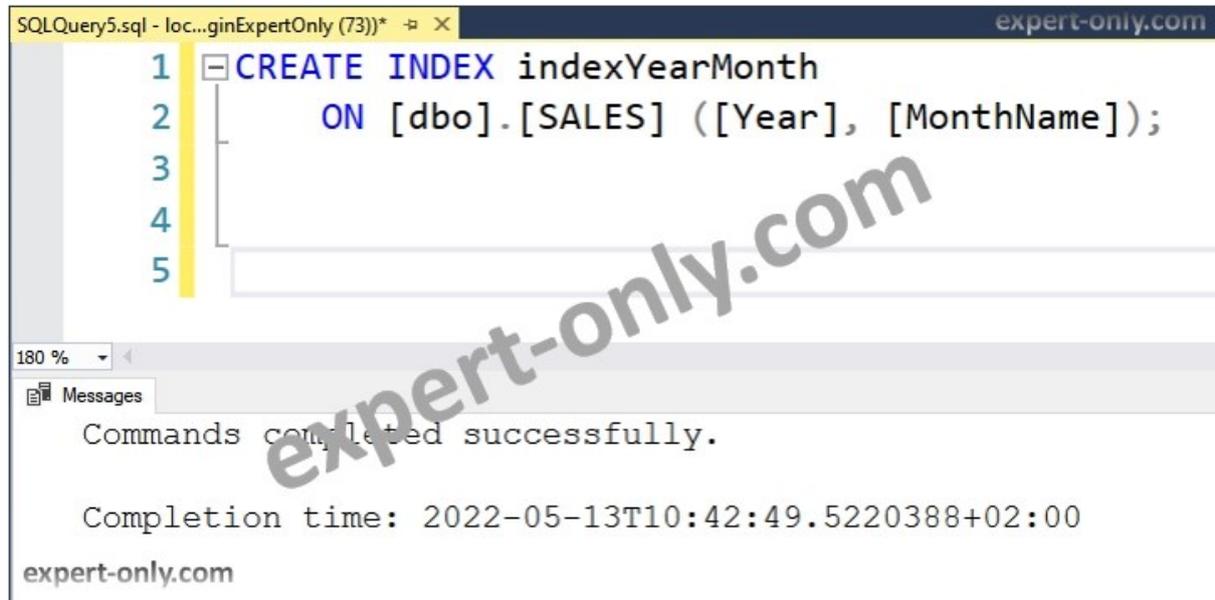
Créer une table des ventes avec SQL Server



Etape 2 : Ajouter l'index sur la table des ventes "Sales"

Enfin, considérons sur notre système d'analyse les colonnes principales utilisées pour filtrer les données. Il s'agit de l'année et du nom du mois : [Year] et [MonthName].

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



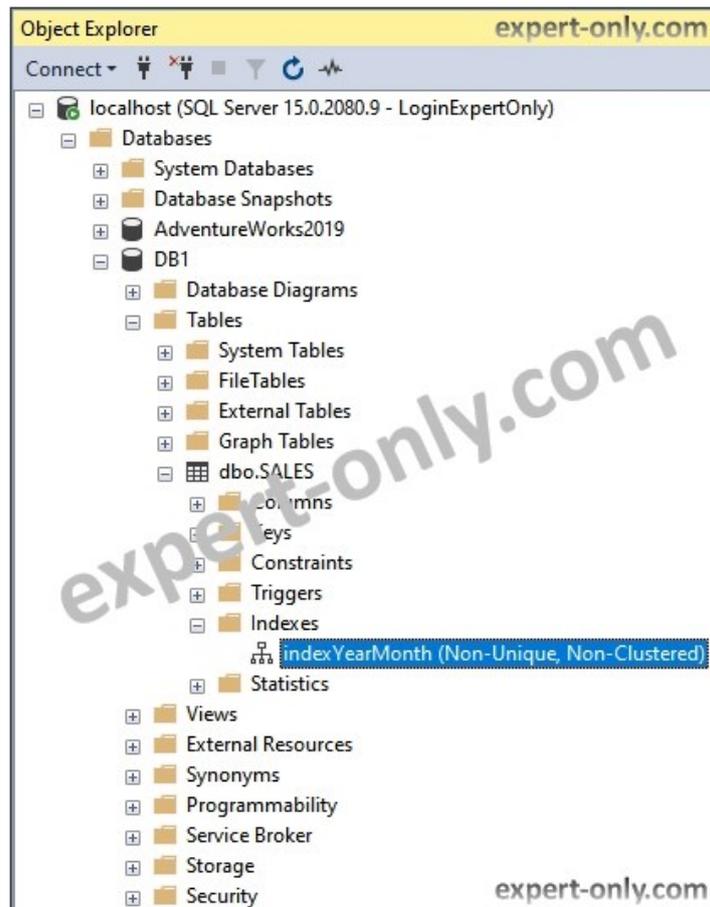
```
SQLQuery5.sql - loc...ginExpertOnly (73)* x expert-only.com  
1 CREATE INDEX indexYearMonth  
2   ON [dbo].[SALES] ([Year], [MonthName]);  
3  
4  
5  
180 %  
Messages  
Commands completed successfully.  
Completion time: 2022-05-13T10:42:49.5220388+02:00  
expert-only.com
```

Créer un index SQL Server



Étape 3 : Vérifier l'index sous la table dans SSMS

Développez les propriétés de la table des ventes et vérifiez la présence de l'index. Noter que **par défaut l'index est un index non-unique et non-cluster**.



Vérifier un index SQL Server avec SSMS

Notes :

- Utilisez une table existante pour créer l'index ou créer une nouvelle table qui doit être indexée.
- Utilisez le logiciel gratuit de Microsoft SQL Server Management Studio (SSMS) pour effectuer les différentes comme exécuter les requêtes.

En outre, réutiliser les exemples T-SQL ci-dessus. Le code est prêt à être copié et collé, il suffit donc de l'adapter et de l'exécuter dans [SQL Server Management Studio](#) (SSMS).

En effet, toute requête utilisant **l'année et le mois** dans la clause where utilise l'index nouvellement créé. En d'autres termes, l'index **indexYearMonth** améliore les performances de la requête. La maintenance des index, comme les commandes **reorganize** et **rebuild**, permet de les maintenir efficaces.

Pour aller plus loin, l'étape suivante consiste à analyser et **reconstruire l'index** afin d'organiser les valeurs dans l'index et les statistiques.

