

leren. durven. doen.



# *DATABANKEN*

*C# programmeur*

*T-SQL*

*Advanced*

# INHOUD

## Optimalisatie queries

# Sql Server

## Sytem- en database catalogoog

# System-en database catalog

- **system catalog** in de masterdb bevat **metadata** over het **ganse systeem** en over de andere databanken
  - **sysdatabases**: 1 rij per databank
  - **syslogins**: 1 rij per login account
  - ...
- **database catalog** bevat metadata over **1 specifieke db**
  - **sysobjects**: algemene informatie van tabellen, views, stored procedures, ...
  - **syscolumns**: de kolommen in tabellen of views; parameters in stored proc.
  - **sysindexes**: de indexen
  - **syscomments**: de definitie van views, stored procedures, ...
  - **sysdepends**: afhankelijkheden tussen objecten
  - **sysconstraints**: mapping tussen constrainten object waarop gedefinieerd
  - **sysreferences**: mapping van foreign key constraint met refererende kolom
  - **sysusers**: de gebruikers
  - **sysprotects**: de gebruikersbevoegdheden
  - **sysfiles**: de bestanden waarin de database is opgeslagen
  - ...

# Sql Server

## Optimalisatie van Queries

# Space allocation door SQL Server

- SQL Server gebruikt random access file
- Space allocation in *extents* en *pages*
- Page = 8 kB blok aaneensluitende space
- Extent = 8 logisch opeenvolgende pages.
  - uniform extents: voor één db-object
  - mixed extents: kunnen gedeeld worden door 8 db-objecten
- Nieuwe tabel of index: allocatie in mixed extent
- Uitbreiding > 8 pages: in uniform extent

# Table scan

- **heap**: ongeordende verzameling van data-pages zonder clustered index = standaard opslag van een tabel
- toegang via Index Allocation Map (**IAM**)
- **table scan**: als een query alle pages ophaalt.

## **performantie-issues met heap:**

- fragmentatie**: tabel staat verspreid over verschillende, niet-opeenvolgende pages
- forward pointers**: als een rij met variabele lengte (vb. varchar-velden) wordt geüpdatet waardoor ze langer wordt, wordt een forward pointer ingevoegd naar een ander pagina, enz.

**-> table scan wordt nog trager.**



# Veroorzaakt mijn query een table scan?

Bekijk het Execution Plan van de query:

The screenshot displays the SQL Server Enterprise Manager interface. The 'Query' menu is open, showing options like 'Execute', 'Include Actual Execution Plan', and 'Include Live Query Statistics'. The 'Include Actual Execution Plan' option is selected. The query text in the editor is:

```
Script for SelectTopNRows command from SSMS *****/  
(1000) [BusinessEntityID]  
[PersonType]  
[NameStyle]  
[Title]  
[FirstName]  
[MiddleName]  
[LastName]  
[Suffix]  
[EmailPromotion]
```

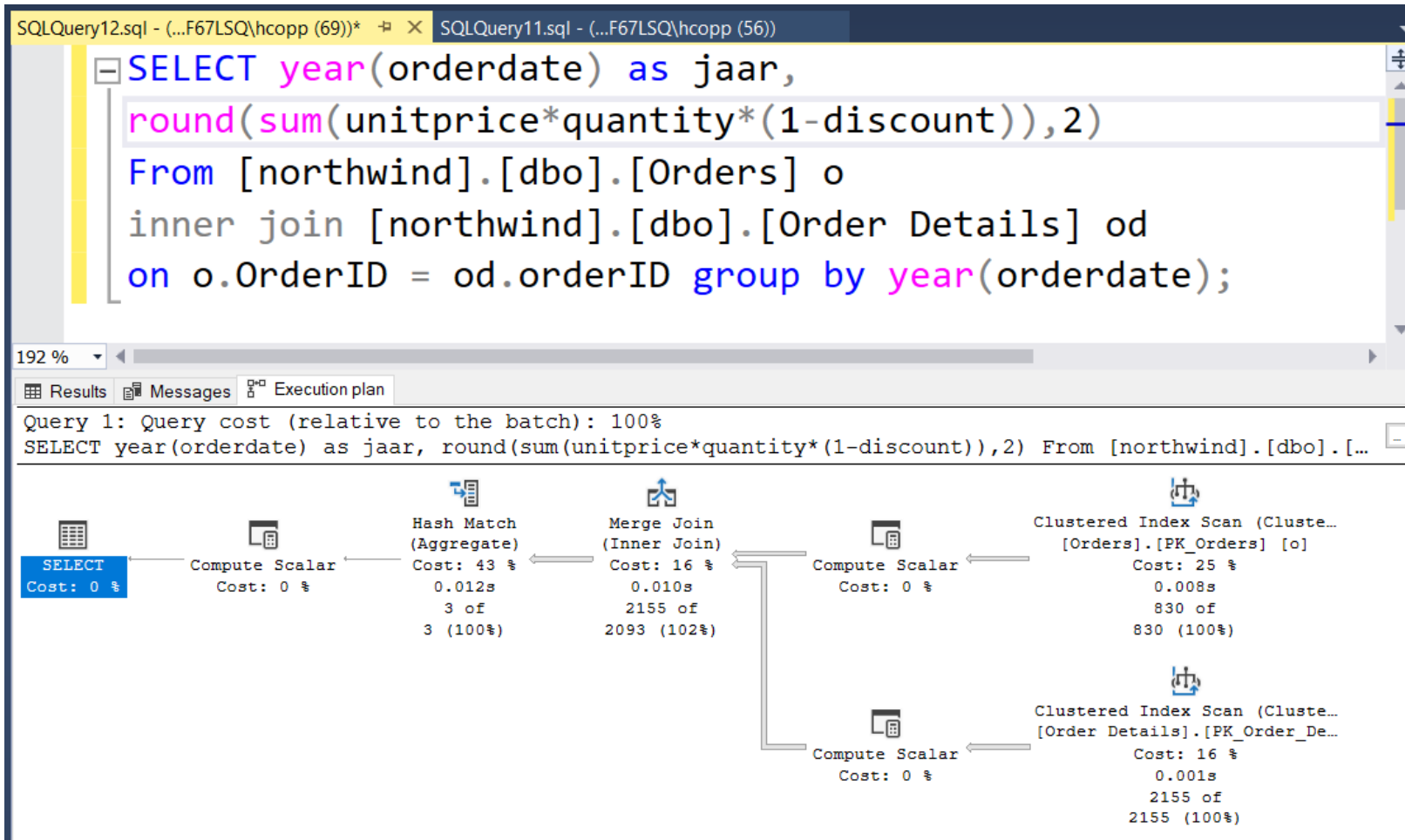
The execution plan for 'Query 1: Query cost (relative to the batch): 100%' is shown below the query text. It consists of three operators:

- Clustered Index Scan (Clustered Index Scan)**: Cost: 100 %, 1000 of 1000 (100%).
- Top**: Cost: 0.034s, 1000 of 1000 (100%).
- SELECT**: Cost: 0 %.

The status bar at the bottom indicates: 'Query executed successfully. (local) (15.0 RTM) DESKTOP-SF67LSQ\hcoppp ... AdventureWorks2017 00:00:00 | 1,000 rows'.

# Query Execution Plan

= Analyse die de server maakt van elke query vóór uitvoering, m.b.t. bijv. te gebruiken indexes.



# Vergelijk 2 query's met elkaar

db AdventureWorks: [https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008/ms124438\(v=sql.100\)](https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008/ms124438(v=sql.100))

- voer 2 query's samen uit.
- tabel Person.Person2 is kopie van Person.Person, maar zonder indexen.

The screenshot displays the SQL Server Enterprise Manager interface. At the top, two query windows are open: 'SQLQuery12.sql' and 'SQLQuery11.sql'. The active window shows the following SQL code:

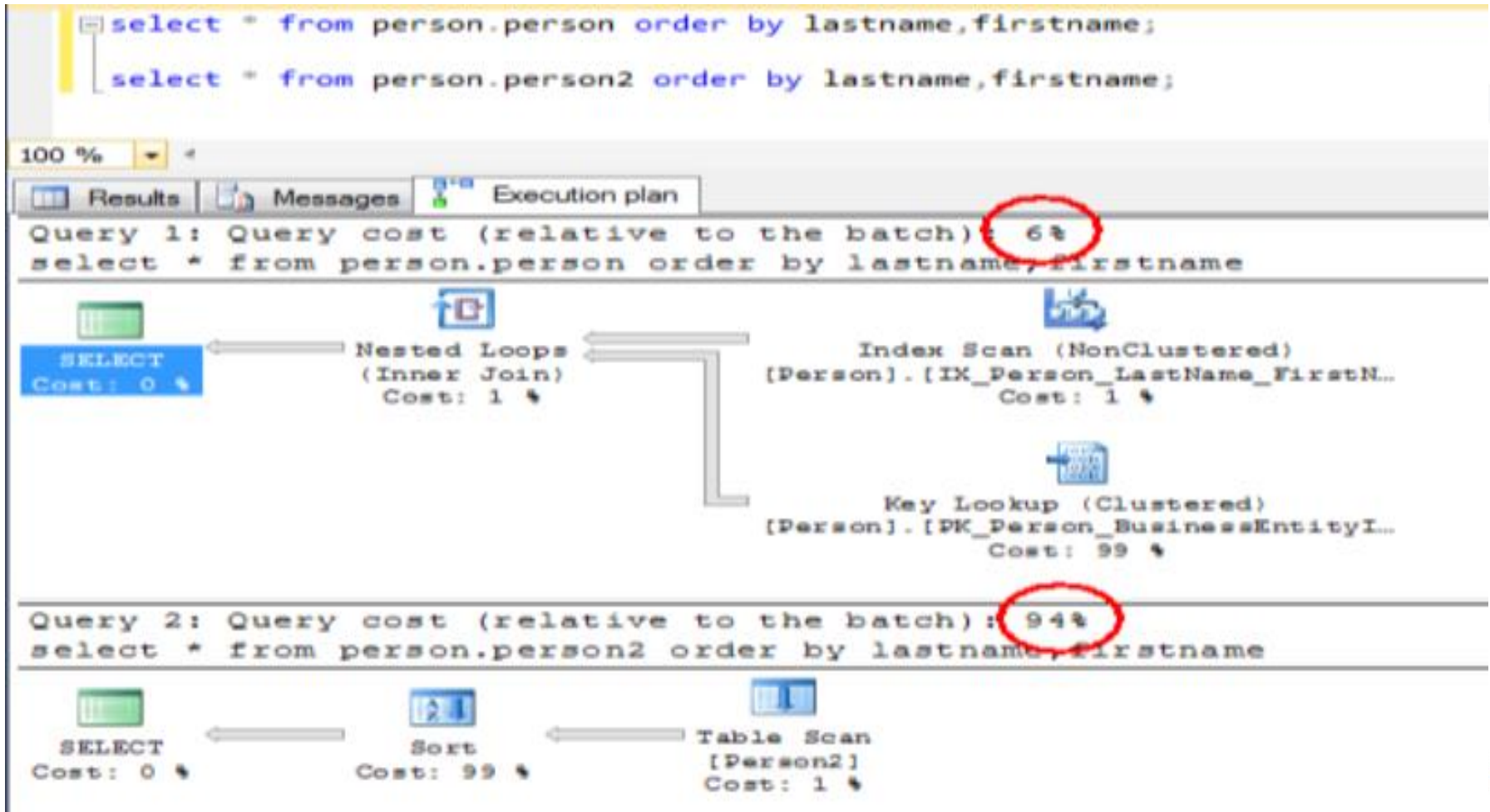
```
--USE [AdventureWorks2017]
--GO
--insert * into Person.Person2 from Person.Person
select * from Person.Person order by lastname, firstname;
select * from Person.Person2 order by lastname, firstname;
```

Below the code, the 'Execution plan' tab is selected. It shows the execution plan for 'Query 1: Query cost (relative to the batch): 50%'. The plan consists of a 'SELECT' operator (Cost: 0 %) connected to a 'Sort' operator (Cost: 32 %, 0.115s, 19972 of 19972 (100%)). The 'Sort' operator is connected to a 'Clustered Index Scan (Clustered Index: [Person].[PK\_Person\_BusinessID])' operator (Cost: 68 %, 0.038s, 19972 of 19972 (100%)).

Below the first execution plan, the 'Execution plan' tab is selected for 'Query 2: Query cost (relative to the batch): 50%'. The plan consists of a 'SELECT' operator (Cost: 0 %) connected to a 'Sort' operator (Cost: 32 %, 0.117s, 19972 of 19972 (100%)). The 'Sort' operator is connected to a 'Table Scan [Person2]' operator (Cost: 68 %, 0.022s, 19972 of 19972 (100%)).

# Vergelijk 2 query's met elkaar

- Query op Person2 duurt 16x langer! (wanneer tabel véél records bevat en dus niet volledig in geheugen kan worden bijgehouden)



# Indexen

- **wat?**

- geordende structuur die op de records uit een tabel wordt gelegd

- **waarom?**

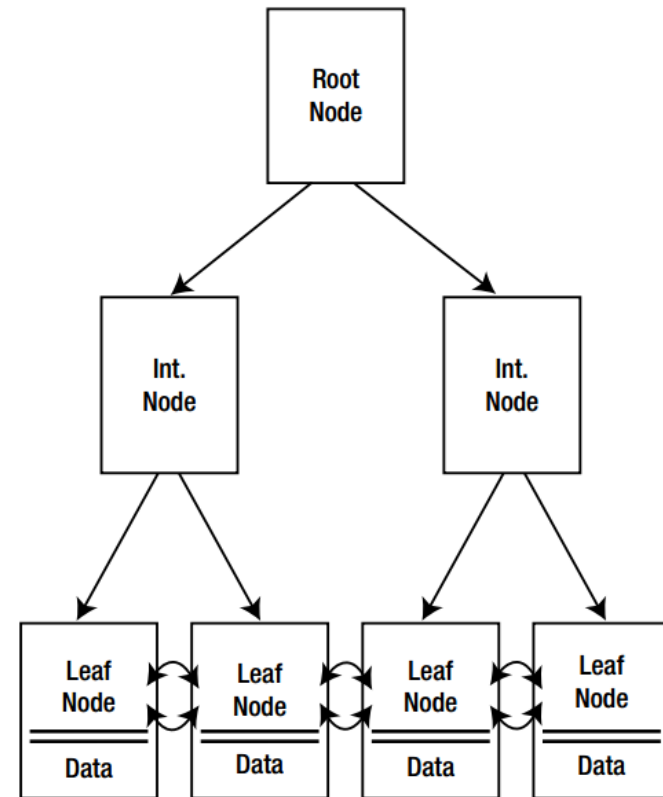
- kan toegang tot data versnellen
  - kan uniciteit van rijen afdwingen

- **waarom niet?**

- indexen nemen opslagruimte in beslag (overhead)
  - kunnen performantie ook doen dalen bv. vertragen van select en updates

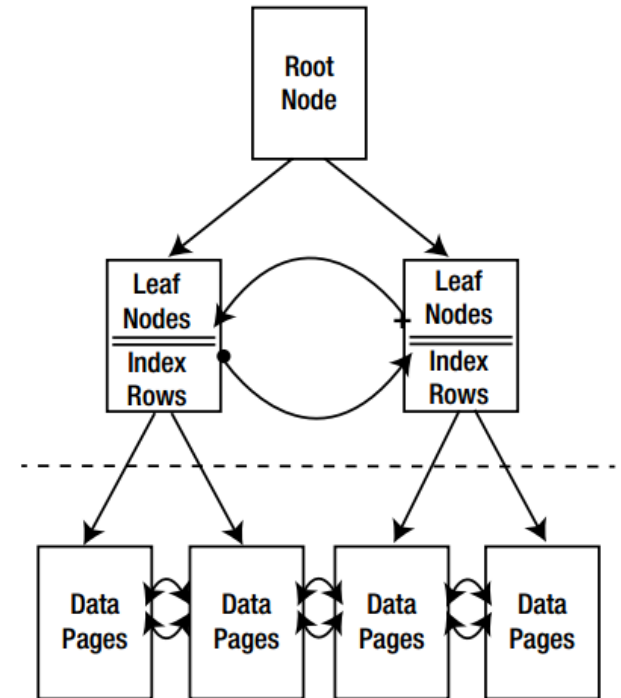
# Clustered index

- de **fysische** volgorde van de rijen in een tabel is deze van de clustered index
- elke tabel kan maar 1 clustered index hebben
- legt **unieke** waarden op
- primary key constraint
- voordelen t.o.v. table scan:
  - dubbel gelinkte lijst zorgt voor volgorde bij lezen van sequentiële records
  - geen forward pointers

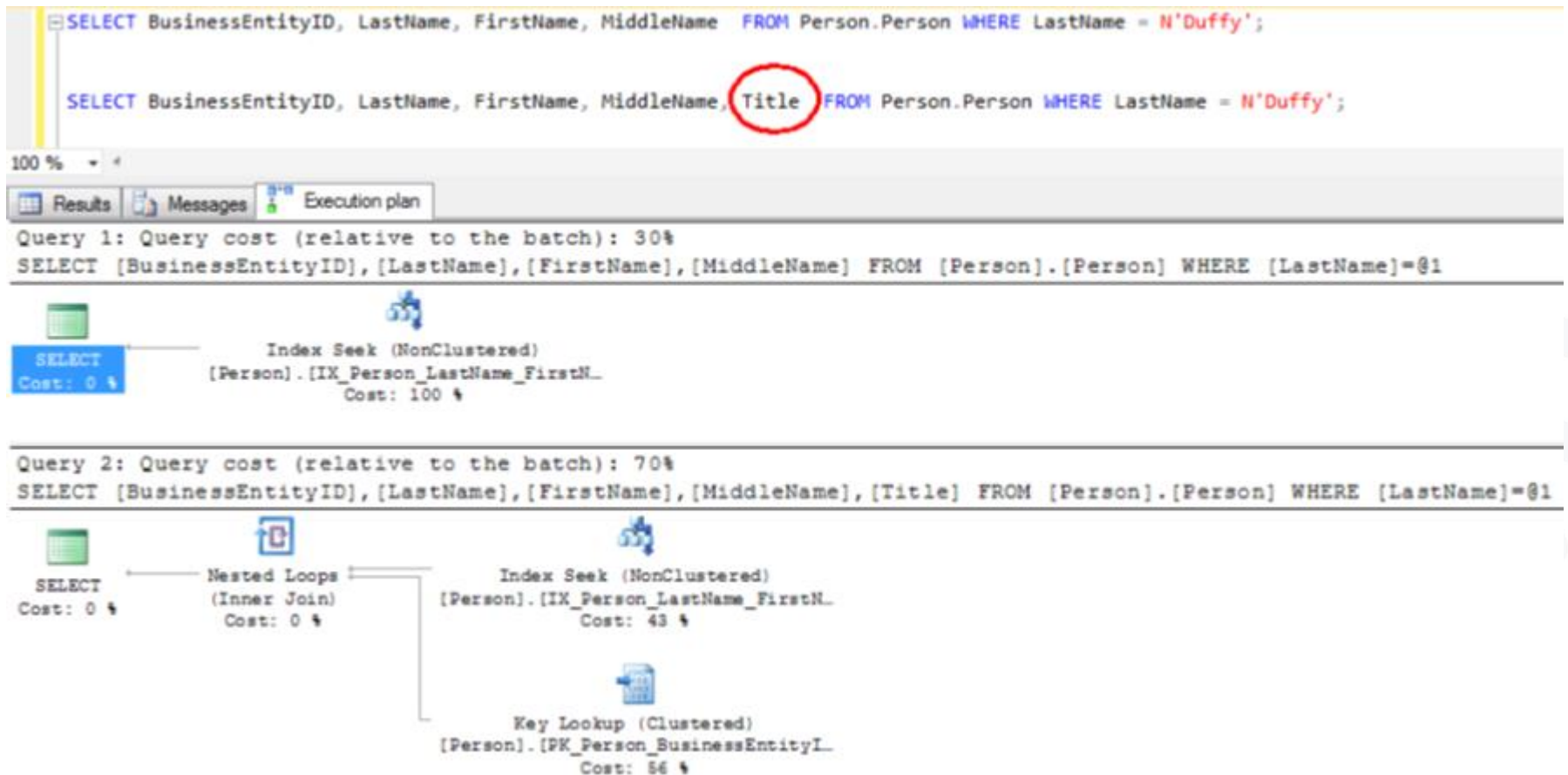


# Non-clustered index

- is de **default** type van index
- werkt **trager** dan **clustered index**
- meerdere per tabel mogelijk
- elke *leaf* bevat sleutelwaarde en *row locator*
  - naar positie in clustered index als die bestaat
  - naar heap in het andere geval



# Covering index:voorbeeld (db AdventureWorks)



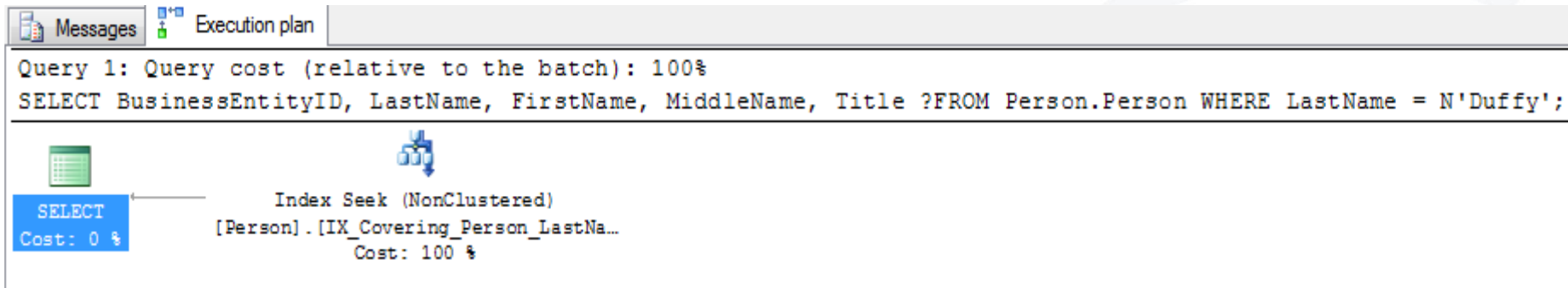
- index seek via nonclustered index om lastname=N'Duffy' op te zoeken
- key\_loop up in clustered index (= data) om Title op te halen want dit zit niet in de index



# 1 index met meerdere kolommen vs. meerdere indexen met 1 kolom

M.a.w.

```
CREATE NONCLUSTERED INDEX [IX_Covering_Person_LastName_FirstName_MiddleName]
ON
[Person].[Person]
(
[LastName]ASC,
[FirstName]ASC,
[MiddleName]ASC
)INCLUDE (Title);
```



# Covering index: oplossing

**M.A.W.**

```
CREATE NONCLUSTERED INDEX IX_Person_LastName_FirstName ON  
Person.Person (LastName, FirstName)
```

**OF ?**

```
CREATE NONCLUSTERED INDEX IX_Person_LastName ON  
Person.Person (LastName)  
en  
CREATE NONCLUSTERED INDEX IX_Person_FirstName ON  
Person.Person (FirstName)  
DB
```

# Covering index: oplossing

Regel in SQL Server:

## **Regel in SQL Server:**

Bij query (bijv. where-clause) op enkel 2<sup>de</sup> en/of 3<sup>de</sup>, ...veld van de index, wordt de index niet gebruikt.

**Dus bij:**

```
SELECT LASTNAME, FIRSTNAME  
FROM PERSON.PERSON  
WHERE FIRSTNAME = 'Chris';
```

wordt de dubbele index niet gebruikt!

Conclusie: kijk naar de meest gebruikte query's en stem daar je indexen op af.

# Covering index: oplossing

Proef op de som: met alleen gecombineerde index op  
Lastname\_Firstname\_Middlename  
DB :

SQL Server Enterprise Manager interface showing query execution plans for two queries.

**Query 1:** Query cost (relative to the batch): 3%  
SELECT [LASTNAME], [FIRSTNAME] FROM [PERSON].[PERSON] WHERE [LASTNAME]=@1

Execution plan for Query 1: Index Seek (NonClustered) on (Person). (IX\_Person\_LastName\_FirstM... Cost: 100 %

**Query 2:** Query cost (relative to the batch): 97%  
SELECT [LASTNAME], [FIRSTNAME] FROM [PERSON].[PERSON] WHERE [FIRSTNAME]=@1

Execution plan for Query 2: Index Scan (NonClustered) on (Person). (IX\_Person\_LastName\_FirstM... Cost: 100 %

Message: Missing Index (Impact 96.5897): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON...

# VRAGEN?

# REFERENTIES

<https://www.w3schools.com/sql>

<https://www.sqlservertutorial.net/sql-server-triggers/sql-server-create-trigger/>

<https://docs.microsoft.com/en-us/sql/t-sql/statements/create-trigger-transact-sql?view=sql-server-2017>