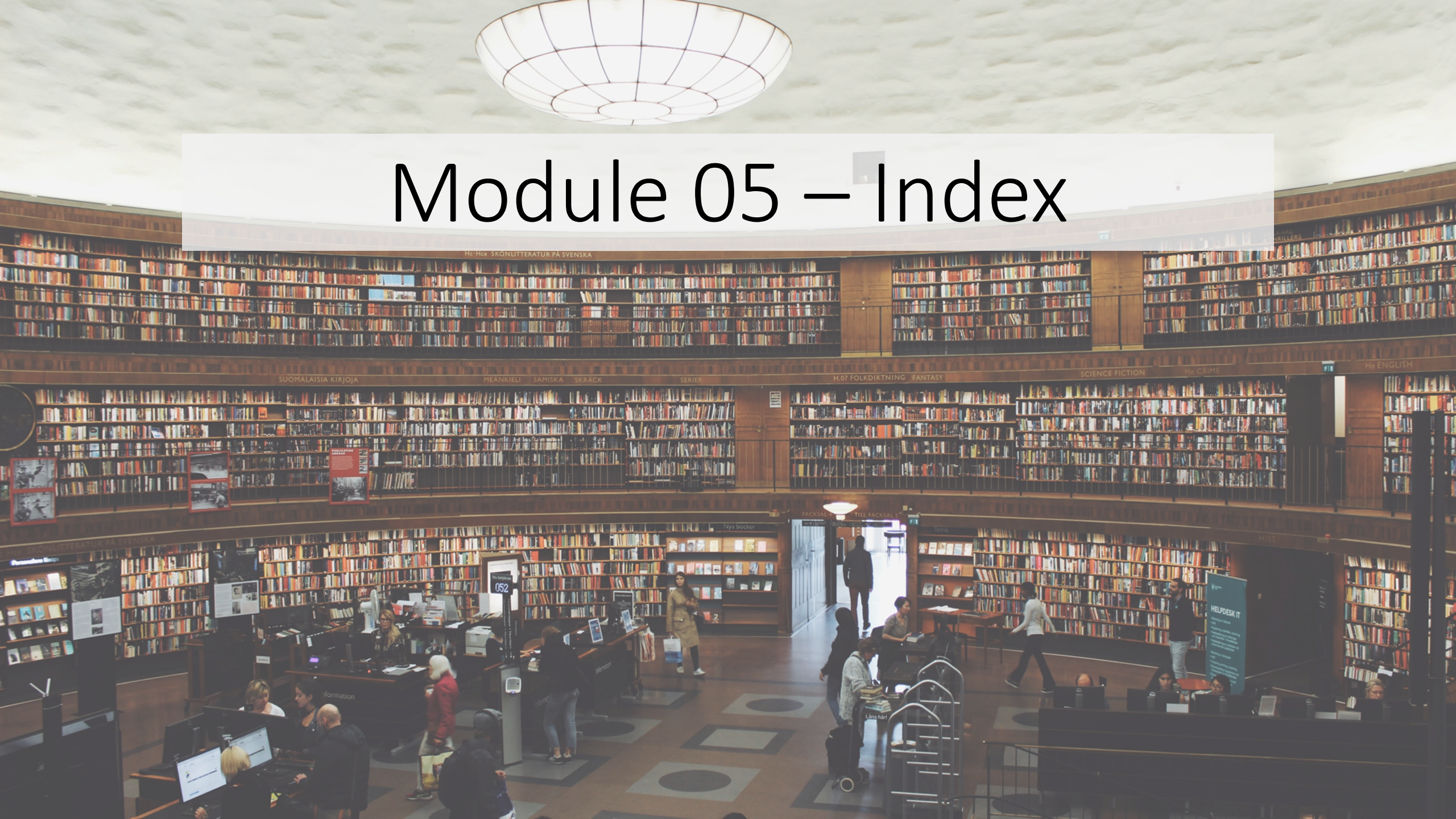


# Module 05 – Index



# Objectifs

- Les index
- Plan d'exécution

# Index

- Un index permet d'accélérer les **recherches** en utilisant une structure de données spécialisée
- L'utilisation d'index a une influence sur toutes les opérations du DML : l'ajout ou la modification de lignes force la mise à jour des index
  - Plus vous avez d'index, plus ces opérations seront ralenties

# Index SQL Server

Type	Description
CLUSTERED INDEX	Un par table. Les lignes de la table sont enregistrées dans l'ordre de l'index. Généralement, on utilise la clef primaire. SQL Server utilise la structure de données B+-Tree.
NON CLUSTERED INDEX	Autant que vous voulez. Séparés de la structure des données. Chaque clef contient un pointeur vers les données.
...	...

<https://learn.microsoft.com/en-us/sql/relational-databases/indexes/indexes>

[https://en.wikipedia.org/wiki/B%2B\\_tree](https://en.wikipedia.org/wiki/B%2B_tree)



# Création d'index en SQL Server

- Utilisez un nom qui suit la nomenclature suivante :
  - IX\_<NomTable>\_<Champs>
  - FK\_<NomTable>\_<ClefEtrangère>

```
CREATE [ UNIQUE ] [ CLUSTERED | NONCLUSTERED ] INDEX index_name
ON <object> ( column [ ASC | DESC ] [ ,...n ] )
[ INCLUDE ( column_name [ ,...n ] ) ]
[ WHERE <filter_predicate> ]
[ WITH ( <relational_index_option> [ ,...n ] ) ]
[ ON { partition_scheme_name ( column_name )
      | filegroup_name
      | default
    }
]
[ FILESTREAM_ON { filestream_filegroup_name | partition_scheme_name | "NULL"
]
[ ; ]
```

# Création d'index - Exemple

- Exemple de création d'un index sans options:

```
CREATE INDEX idxNomIndex ON NomTable (NomChamp)
```

```
CREATE INDEX idxNomPrenomClient ON Clients (Nom, Prenom)
```

- Exemple de création d'un index UNIQUE

```
CREATE UNIQUE INDEX idxClientEmailUnique ON Clients (Email)
```

# Quand créer des index ?

- Les clefs primaires sont généralement indexées (CLUSTERED INDEX) et les enregistrements sont donc dans l'ordre de la clef
- Les clefs étrangères ne sont pas indexées par défaut => cela peut être une bonne pratique de le faire dans la plupart des cas
- Champs souvent recherchés
  - Recherche sur des nombres, des dates, des débuts de champs, etc.
  - Attention aux champs de type « Text » : il faut utiliser un index de type « Full-text » optimisé pour la recherche de texte
- Éviter d'indexer des champs de tables qui ne sont pas beaucoup utilisés dans des jointures / recherches comme des tables de journaux (beaucoup d'insertions, peu/pas de lectures)

# Plan d'exécution

- Une requête est déclarative : l'ordre d'exécution n'est pas imposé
- On peut forcer cette ordre d'exécution (comme l'ordre des jointures par exemple) en passant des options (Ex. : `OPTION (FORCE ORDER)`) ou en utilisant des sous-requêtes ou des tables temporaires
- Avant d'exécuter une requête, SQL Server va estimer la meilleure façon d'exécuter votre requête en se basant sur les statistiques des tables et des précédentes exécutions
- « Estimated Query Plan » : Plan d'exécution estimé



# Quand créer des index ? - suite

- Pour les petites tables, l'utilisation d'index n'améliore pas les performances
- Un index optimise une requête si la requête retourne moins de 25% des données
  - On ne devrait pas créer d'index sur des champs qui contiennent beaucoup de NULL
- La(les) condition(s) la plus restrictive retourne moins de 10% des enregistrements de la table
- La condition la plus restrictive sera souvent utilisée dans une requête
- Des champs sont souvent référencés dans les clauses ORDER BY et GROUP BY

La condition la plus restrictive signifie une condition dans la clause WHERE qui retourne le moins d'enregistrements.

# Performances

- On doit toujours mettre la condition la plus restrictive à la fin de la requête car les requêtes sont lues à partir de la fin dans la plupart des SGBD
- Pour une requête qui est exécutée régulièrement, utilisez une procédure stockée (sera vu plus tard dans le cours)
- Évitez d'utiliser le OR lorsque possible et les remplacer par IN

# Plan d'exécution

- Une requête est déclarative : l'ordre d'exécution n'est pas imposé
- On peut forcer cet ordre d'exécution (comme l'ordre des jointures par exemple) en passant des options (Ex. : `OPTION (FORCE ORDER)`) ou en utilisant des sous-requêtes ou des tables temporaires
- Avant d'exécuter une requête, SQL Server va estimer la meilleure façon d'exécuter votre requête en se basant sur les statistiques des tables et des précédentes exécutions
- « Estimated Query Plan » : Plan d'exécution estimé

# Plan d'exécution sans index

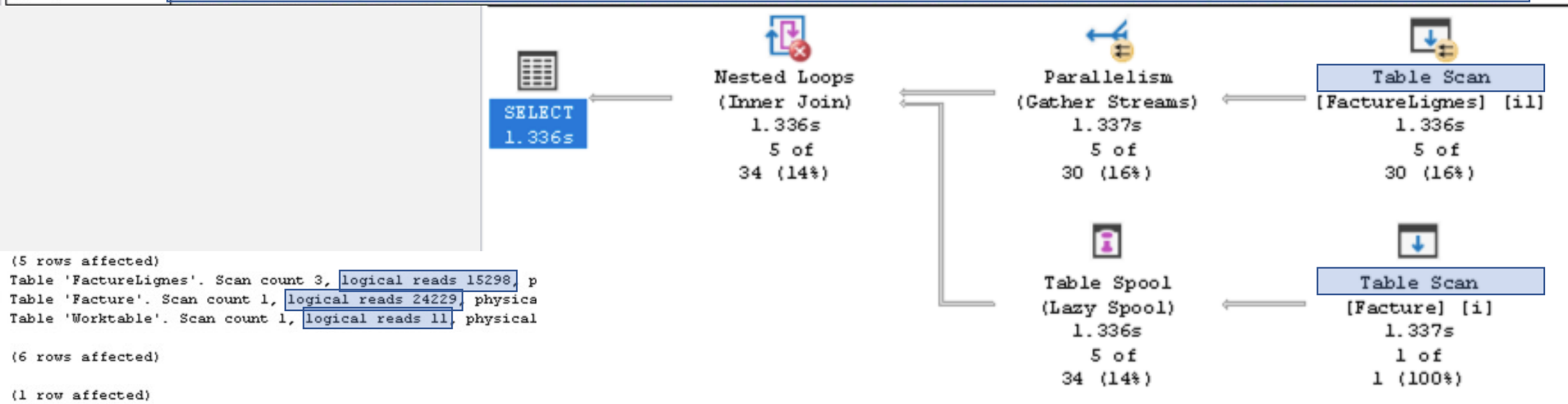
```
SELECT *  
INTO Facture  
FROM Sales.Invoices
```

```
SELECT *  
INTO FactureLignes  
FROM Sales.InvoiceLines
```

```
SET STATISTICS IO ON  
SET STATISTICS TIME ON
```

```
SELECT *  
FROM Facture i  
INNER JOIN FactureLignes il ON il.InvoiceID = i.InvoiceID  
WHERE i.InvoiceID = 126  
ORDER BY i.InvoiceID
```

Estimated query progress: 100%	Query 1: Query cost (relative to the batch): 100% SELECT * FROM Facture i INNER JOIN dbo.FactureLignes il ON il.InvoiceID = i.InvoiceID WHERE i.InvoiceID = 126 ORDER BY i.InvoiceID Missing Index (Impact 60.5032): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[Facture] ([InvoiceID])
--------------------------------	---



(5 rows affected)  
Table 'FactureLignes'. Scan count 3, logical reads 15298, p  
Table 'Facture'. Scan count 1, logical reads 24229, physical reads 1, logical reads 11, physical reads 11  
Table 'Worktable'. Scan count 1, logical reads 11, physical reads 11

(6 rows affected)

(1 row affected)

SQL Server Execution Times:  
CPU time = 249 ms, elapsed time = 485 ms.  
SQL Server parse and compile time:  
CPU time = 0 ms, elapsed time = 0 ms.

# Plan d'exécution avec index

```
SELECT *  
FROM Sales.Invoices i  
INNER JOIN Sales.InvoiceLines il ON il.InvoiceID = i.InvoiceID  
WHERE i.InvoiceID = 126  
ORDER BY i.InvoiceID
```

Estimated query progress: 100% Query 1: Query cost (relative to the batch): 100%  
SELECT \* FROM Sales.Invoices i INNER JOIN Sales.InvoiceLines il ON il.InvoiceID = i.InvoiceID WHERE i.InvoiceID = 126 ORDER BY i.InvoiceID

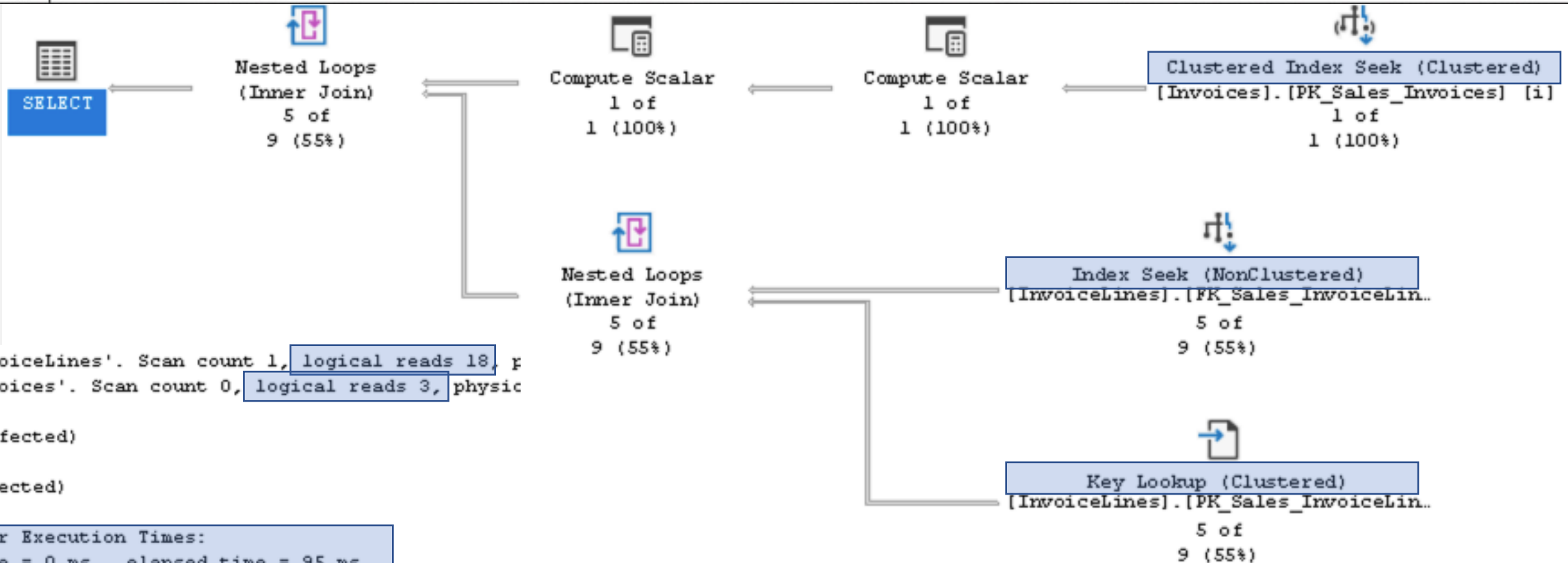


Table 'InvoiceLines'. Scan count 1, logical reads 18, physical reads 0  
Table 'Invoices'. Scan count 0, logical reads 3, physical reads 0

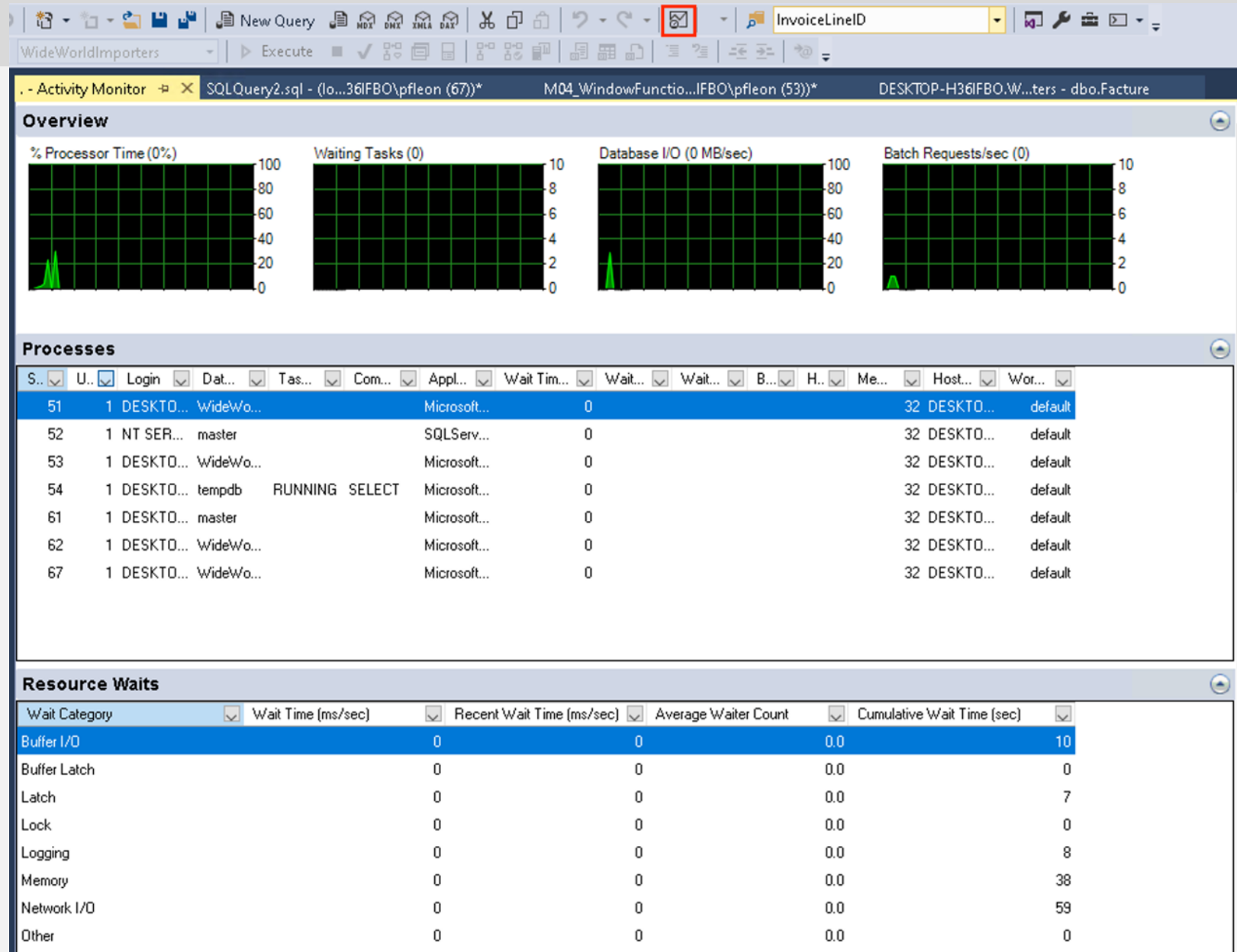
(8 rows affected)

(1 row affected)

SQL Server Execution Times:  
CPU time = 0 ms, elapsed time = 95 ms.

SQL Server parse and compile time:  
CPU time = 0 ms, elapsed time = 0 ms.

# Moniteur d'activités





# SQL Server Profiler

EventClass	TextData	ApplicationName	NTUserName	LoginName	CPU	Reads	Writes	Duration	ClientProcessID	SPID	StartTime	EndTime	
Audit Login	-- network protocol: LPC set quoted...	Microsoft SQ...	pfleon	DESKTO...						9784	66	2022-11-15 17:30:59...	
RPC:Completed	declare @p1 int set @p1=1 exec sp_...	Microsoft SQ...	pfleon	DESKTO...	0	334	0	3		9784	66	2022-11-15 17:30:59...	2022
SQL:BatchStarting	SELECT * FROM Sales.Invoices i INN...	Microsoft SQ...	pfleon	DESKTO...						9784	67	2022-11-15 17:30:58...	
SQL:BatchCompleted	SELECT * FROM Sales.Invoices i INN...	Microsoft SQ...	pfleon	DESKTO...	15	486	0	106		9784	67	2022-11-15 17:30:58...	2022
RPC:Completed	declare @p1 int set @p1=2 exec sp_...	Microsoft SQ...	pfleon	DESKTO...	0	334	0	2		9784	66	2022-11-15 17:30:59...	2022
Audit Logout		Microsoft SQ...	pfleon	DESKTO...	0	764	0	1043		9784	66	2022-11-15 17:30:59...	2022
SQL:BatchStarting	SET STATISTICS XML OFF SET STATISTIC...	Microsoft SQ...	pfleon	DESKTO...						9784	67	2022-11-15 17:31:00...	
SQL:BatchCompleted	SET STATISTICS XML OFF SET STATISTIC...	Microsoft SQ...	pfleon	DESKTO...	0	0	0	0		9784	67	2022-11-15 17:31:00...	2022
SQL:BatchStarting	SELECT @@SPID;	Microsoft SQ...	pfleon	DESKTO...						9784	67	2022-11-15 17:31:39...	
SQL:BatchCompleted	SELECT @@SPID;	Microsoft SQ...	pfleon	DESKTO...	0	0	0	0		9784	67	2022-11-15 17:31:39...	2022
SQL:BatchStarting	SET STATISTICS XML ON SET STATISTICS...	Microsoft SQ...	pfleon	DESKTO...						9784	67	2022-11-15 17:31:39...	
SQL:BatchCompleted	SET STATISTICS XML ON SET STATISTICS...	Microsoft SQ...	pfleon	DESKTO...	0	0	0	0		9784	67	2022-11-15 17:31:39...	2022
SQL:BatchStarting	SELECT @@SPID	Microsoft SQ...	pfleon	DESKTO...						9784	67	2022-11-15 17:31:39...	
SQL:BatchCompleted	SELECT @@SPID	Microsoft SQ...	pfleon	DESKTO...	0	0	0	0		9784	67	2022-11-15 17:31:39...	2022
Audit Login	-- network protocol: LPC set quoted...	Microsoft SQ...	pfleon	DESKTO...						9784	66	2022-11-15 17:31:39...	
RPC:Completed	declare @p1 int set @p1=1 exec sp_...	Microsoft SQ...	pfleon	DESKTO...	0	259	0	2		9784	66	2022-11-15 17:31:39...	2022
SQL:BatchStarting	SELECT * FROM Facture i INNER JOIN...	Microsoft SQ...	pfleon	DESKTO...						9784	67	2022-11-15 17:31:39...	
RPC:Completed	declare @p1 int set @p1=2 exec sp_...	Microsoft SQ...	pfleon	DESKTO...	0	334	0	2		9784	66	2022-11-15 17:31:40...	2022
RPC:Completed	declare @p1 int set @p1=3 exec sp_...	Microsoft SQ...	pfleon	DESKTO...	0	166	0	37		9784	66	2022-11-15 17:31:40...	2022
RPC:Completed	declare @p1 int set @p1=4 exec sp_...	Microsoft SQ...	pfleon	DESKTO...	0	334	0	1		9784	66	2022-11-15 17:31:40...	2022
SQL:BatchStarting	select ms_ticks from sys.dm_os_sys_info	Microsoft SQ...	pfleon	DESKTO...						9784	66	2022-11-15 17:31:40...	
SQL:BatchCompleted	select ms_ticks from sys.dm_os_sys_info	Microsoft SQ...	pfleon	DESKTO...	15	260	0	50		9784	66	2022-11-15 17:31:40...	2022
RPC:Completed	declare @p1 int set @p1=5 exec sp_...	Microsoft SQ...	pfleon	DESKTO...	0	268	0	13		9784	66	2022-11-15 17:31:40...	2022
RPC:Completed	declare @p1 int set @p1=6 exec sp_...	Microsoft SQ...	pfleon	DESKTO...	0	334	0	1		9784	66	2022-11-15 17:31:40...	2022
SQL:BatchStarting	select ms_ticks from sys.dm_os_sys_info	Microsoft SQ...	pfleon	DESKTO...						9784	66	2022-11-15 17:31:40...	
SQL:BatchCompleted	select ms_ticks from sys.dm_os_sys_info	Microsoft SQ...	pfleon	DESKTO...	0	183	0	1		9784	66	2022-11-15 17:31:40...	2022
RPC:Completed	declare @p1 int set @p1=7 exec sp_...	Microsoft SQ...	pfleon	DESKTO...	0	191	0	1		9784	66	2022-11-15 17:31:40...	2022
RPC:Completed	declare @p1 int set @p1=8 exec sp_...	Microsoft SQ...	pfleon	DESKTO...	0	334	0	2		9784	66	2022-11-15 17:31:40...	2022
SQL:BatchStarting	select ms_ticks from sys.dm_os_sys_info	Microsoft SQ...	pfleon	DESKTO...						9784	66	2022-11-15 17:31:40...	
SQL:BatchCompleted	select ms_ticks from sys.dm_os_sys_info	Microsoft SQ...	pfleon	DESKTO...	0	183	0	1		9784	66	2022-11-15 17:31:40...	2022
RPC:Completed	declare @p1 int set @p1=9 exec sp_...	Microsoft SQ...	pfleon	DESKTO...	0	191	0	1		9784	66	2022-11-15 17:31:40...	2022
SQL:BatchCompleted	SELECT * FROM Facture i INNER JOIN...	Microsoft SQ...	pfleon	DESKTO...	109	39858	1	1401		9784	67	2022-11-15 17:31:39...	2022
SQL:BatchCompleted	SELECT * FROM Sales.Invoices i INNER JOIN Sales.InvoiceLines il ON il.InvoiceID = i.InvoiceID WHERE i.InvoiceID = 126 ORDER BY i.InvoiceID go SELECT *	Microsoft SQ...	pfleon	DESKTO...	0	334	0	2		9784	66	2022-11-15 17:31:41...	2022