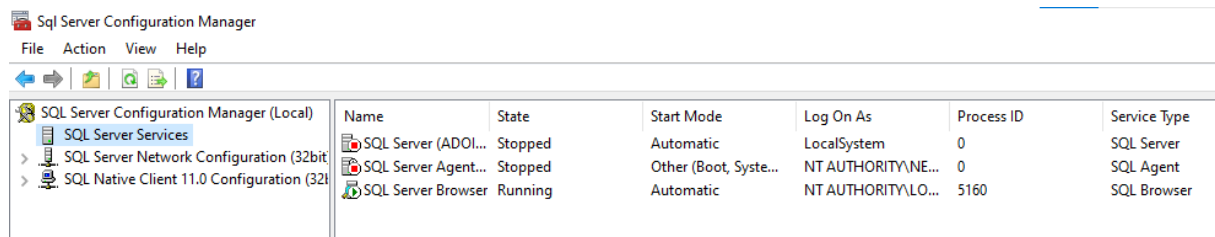


## BLOK 1 – INSTALACJA, PLIKI

1. Przy pomocy Configuration Manager podejrzuj, jakie usługi SQL Servera są uruchomione.

Path: C:\Windows\SysWOW64\SQLServerManager15.msc



SQL Server Configuration Manager

File Action View Help

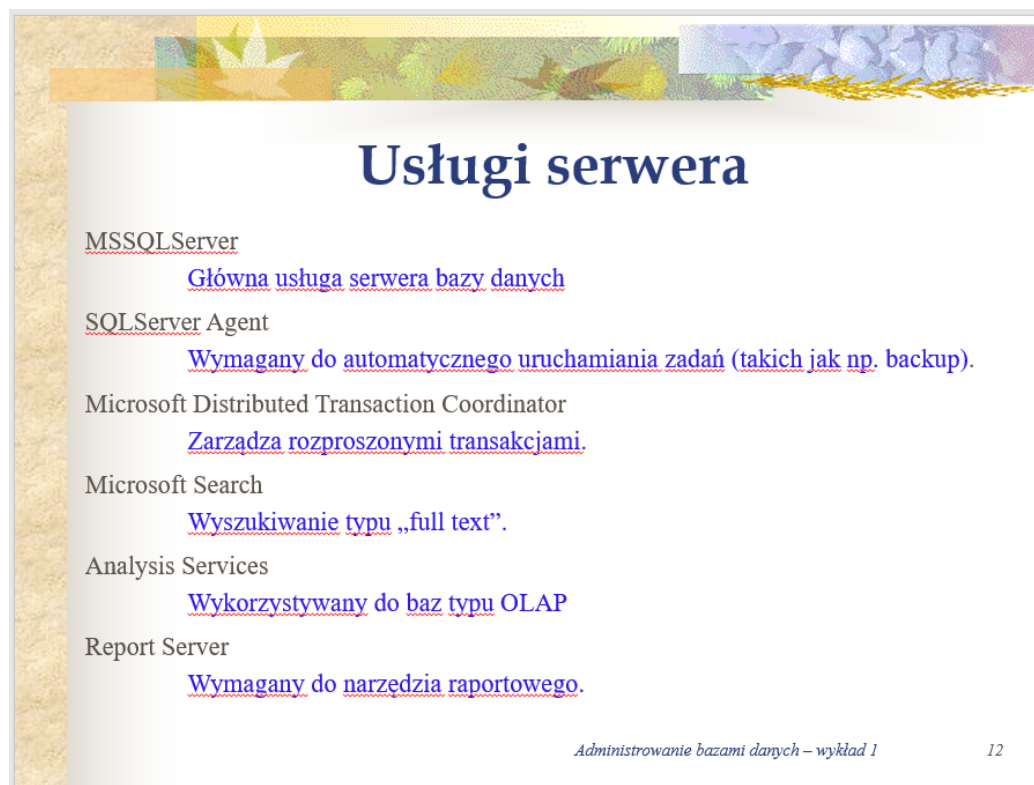
SQL Server Configuration Manager (Local)

SQL Server Services

SQL Server Network Configuration (32bit)

SQL Native Client 11.0 Configuration (32bit)

Name	State	Start Mode	Log On As	Process ID	Service Type
SQL Server (ADOL...	Stopped	Automatic	LocalSystem	0	SQL Server
SQL Server Agent...	Stopped	Other (Boot, Syste...	NT AUTHORITY\NE...	0	SQL Agent
SQL Server Browser	Running	Automatic	NT AUTHORITY\LO...	5160	SQL Browser



# Usługi serwera

MSSQLServer  
[Główna usługa serwera bazy danych](#)

SQLServer Agent  
[Wymagany do automatycznego uruchamiania zadań \(takich jak np. backup\).](#)

Microsoft Distributed Transaction Coordinator  
[Zarządza rozproszonymi transakcjami.](#)

Microsoft Search  
[Wyszukiwanie typu „full text”.](#)

Analysis Services  
[Wykorzystywany do baz typu OLAP](#)

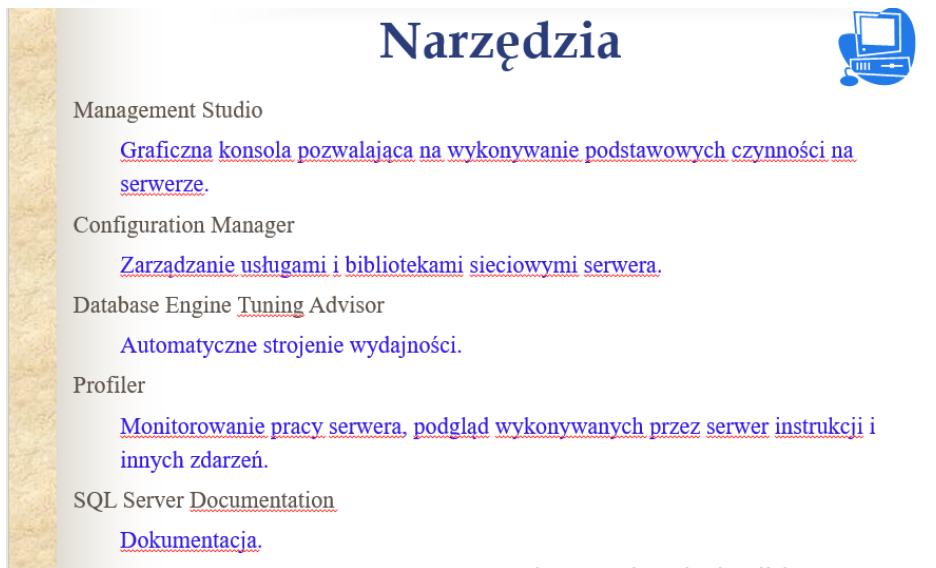
Report Server  
[Wymagany do narzędzia raportowego.](#)

Administrowanie bazami danych – wykład 1

12

2. Zapoznaj się z dostępnymi w zainstalowanym serwerze narzędziami oraz dokumentacją (Management Studio, Books OnLine).

Pisząc skrypt SQL w dowolnym momencie możesz zaznaczyć polecenie i nacisnąć SHIFT+F1. Otworzy się Books OnLine z pomocą dotyczącą wybranego polecenia.



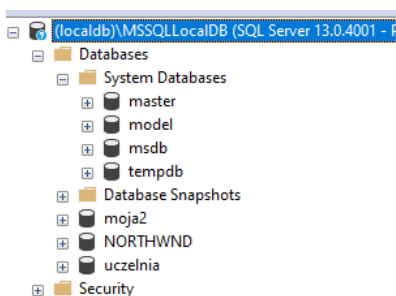
3. Zapoznaj się z bazami danych dostępnymi na serwerze.

Aby sprawdzić zainstalowane instance trzeba wejść w services.msc i poszukać sql server (instancja)

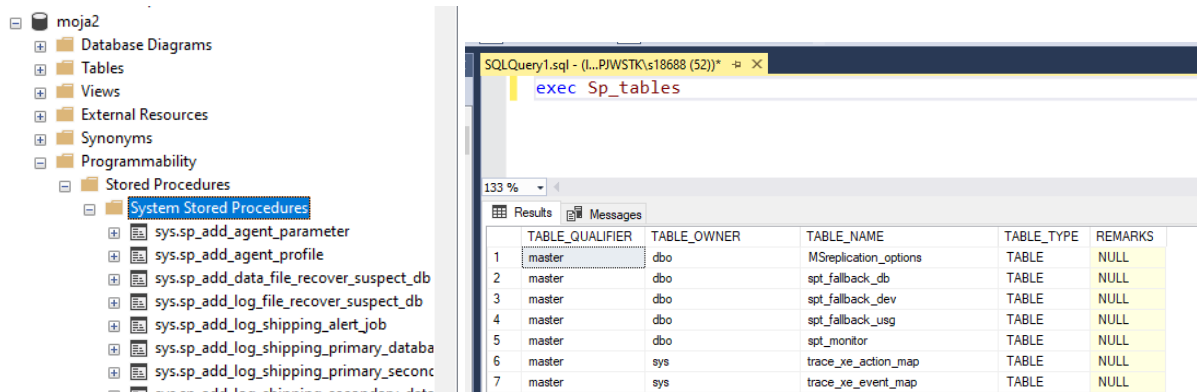
Spot Verifier	Verifies pote...	
SQL Server (ADOITCE)	Provides sto...	
SQL Server Agent (ADOITCE)	Executes jo...	
SQL Server Browser	Provides SQ...	Running
SQL Server VSS Writer	Provides th...	Running
SSDP Discovery	Discovers n...	Running
State Repository Service	Provides re...	Running

Bazy:

- Master - Zawiera tabele i inne obiekty systemowe m.in. konta użytkowników, informacje na temat baz danych użytkowników itp., np. tabele sysdatabases, systypes (metadane).
- Model - Szablon bazy danych. Gdy stworzymy nową bazę, jest ona kopią bazy model.
- Tempdb - Baza w której trzymane są informacje tymczasowe (np. tabele tymczasowe).
- Msdb - Zawiera informacje na temat zaplanowanych zadań.
- Bazy użytkowników



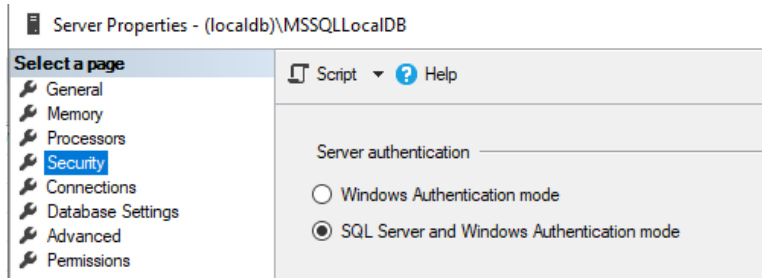
#### 4. Przetestuj procedury *sp\_* omówione w wykładzie.



- Sp\_help obiekt - Wyświetla informacje na temat obiektu np. Tabeli.
- Sp\_helptext obiekt - Wyświetla treść związaną z obiektem np. kod procedury lub wyzwalacza.
- Sp\_helpindex obiekt - Wyświetla informacje na temat indeksów założonych na obiekcie.
- Sp\_tables - Wyświetla tabele bazy danych.

#### 5. Podejrzij ustawienia serwera.

Chodzi o te properties jak się kliknie prawym na instancję bazy, w poszczególnych zakładkach są różne informacje – sposób uwierzytelniania użytkowników ważny:



Na screenie mixed mode

6. Załóż swoją bazę danych, zwracając uwagę na lokalizację plików z danymi i dziennikiem transakcji oraz inne opcje omawiane na wykładzie.

PPM na databases -> new databases

W 1 zakładce są pliki, rodzaje plików:

- Jeden podstawowy plik z danymi (\*.mdf)
- Może zawierać dodatkowe pliki z danymi (\*.ndf)
- Plik z dziennikiem transakcji (transaction log) (\*.ldf)

może być tylko jeden plik mdf, można zmienić startową wielkość plików, sposób ich rozrastania, maksymalną wielkość, lokalizację

W 2 zakładce zmienialiśmy tylko to:

Recovery model

- full
- bulk logged (do logu nie są wpisywane niektóre operacje np. SELECT ... INTO ..., BULK INSERT itp.)
- simple (po zapisaniu w pliku z danymi transakcje są czyszczone z logu, brak możliwości przywrócenia bazy do dowolnego punktu w czasie)

W 3 zakładce są filegroupy, po dodaniu filegroupy dopiero możemy cofnąć się do zakładki 1 i zmienić lokalizację plików (tylko dla nowych,

7. Podejrzyj opcje bazy danych (baza danych -> properties).

Prawdopodobnie interesują nas tylko General, Files, Filegroups i Permissions

8. Używając procedur *sp\_* wymienionych w wykładzie, podejrzuj właściwości swojej bazy danych.

```
exec Sp_helpdb moja2 --baza danych
exec Sp_spaceused --aktualna baza danych
exec Sp_spaceused Osoba --tabelka w aktualnej bazie
```

133 %

Results Messages

	name	db_size	owner	dbid	created	status	compatibility_level
1	moja2	129.00 MB	PJWSTK\18688	6	Mar 10 2021	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	130

	name	fileid	filename	filegroup	size	maxsize	growth	usage
1	moja_dat	1	C:\Users\s18688\Desktop\mojadat.mdf	PRIMARY	8192 KB	Unlimited	5120 KB	data only
2	moja_log	2	C:\Users\s18688\Desktop\mojalog.ldf	NULL	41984 KB	2147483648 KB	5120 KB	log only
3	moja2_dat	3	C:\Users\s18688\moja2dat.ndf	PRIMARY	8192 KB	Unlimited	65536 KB	data only
4	moja3	4	C:\Users\s18688\trzy.ndf	second	73728 KB	Unlimited	65536 KB	data only

	database_name	database_size	unallocated space
1	moja2	129.00 MB	74.69 MB

	reserved	data	index_size	unused
1	13632 KB	8640 KB	4168 KB	824 KB

	name	rows	reserved	data	index_size	unused
1	Osoba	3	72 KB	8 KB	8 KB	56 KB

9. Spróbuj odwołać się do obiektów bazy danych używając pełnej i skróconej nazwy.

*Aby wykonać ćwiczenie, załóż najpierw dowolną tabelę w swojej bazie.*

obiekt

baza\_danych..obiekt

baza\_danych.schemat.obiekt

serwer.baza\_danych.schemat.obiekt –schemat czyli np. dbo

#### 10. Załóż drugą bazę danych z poziomu SQL.

*Aby łatwiej wykonać ćwiczenie, skopiuj polecenie z wykładu zmieniając nazwę bazy i plików oraz ścieżki.*

```
CREATE DATABASE Sales ON  
(NAME = Sales_dat,  
FILENAME = 'c:\program files\microsoft sql server\mssql\data\saledat.mdf',  
SIZE = 2MB, MAXSIZE = UNLIMITED, FILEGROWTH = 5MB)  
LOG ON  
(NAME = Sales_log,  
FILENAME = 'c:\program files\microsoft sql server\mssql\data\salelog.ldf',  
SIZE = 1MB, MAXSIZE = UNLIMITED, FILEGROWTH = 5MB)  
GO
```

#### 11. Dodaj nowy plik z danymi (.ndf) do swojej bazy danych. Plik powinien mieć inną lokalizację niż plik podstawowy.

*Najłatwiej wykonać to ćwiczenie używając interfejsu. Oczywiście można je również wykonać z poziomu SQL, poleceniem ALTER DATABASE. Trudniejsza wersja ćwiczenia (dla chętnych).*

PPM na bazę -> properties -> files

#### 12. Dodaj grupę plików (filegroup) do swojej bazy danych. Utwórz plik w nowej grupie. Przydziel do grupy tabelę bazy danych.

PPM na bazę -> properties -> filegroups –żeby dodać filegroup

PPM na bazę -> properties -> files – żeby dodać plik w nowej grupie

Żeby dodać tabelę w nowej filegroup musimy dodać na końcu polecenia „CREATE TABLE ... ON <filegroup>”

#### 13. Zmień domyślną grupę plików na utworzoną w punkcie 13. Utwórz nową tabelę. Zobacz, do której grupy trafiła.

PPM na bazę -> properties -> filegroups –żeby zmienić domyślną

#### 14. Zapoznaj się z opcją „shrink database” (z poziomu MS i używając DBCC).

*Przy małych rozmiarach bazy nie uda się zapewne zmniejszyć rozmiaru żadnego pliku.*

PPM na bazę -> tasks -> shrink -> database

### 15. Usun swoją bazę danych.

*Baza danych może nie dać się usunąć, jeżeli ktoś jest do niej połączony. Jeżeli tak się dzieje, zamknij wszystkie okna w których pracowałeś na tej bazie.*

PPM na bazę -> delete

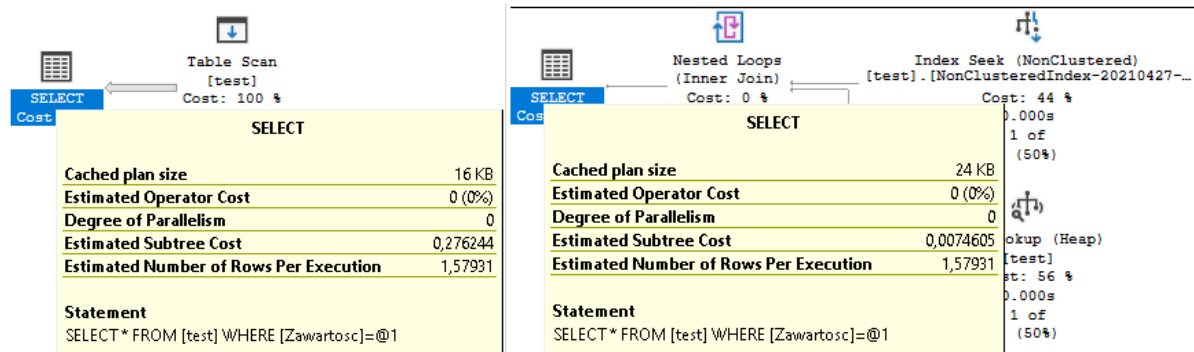
## BLOK 2 - INDEKSY

### 1. Indeks niegrupowany

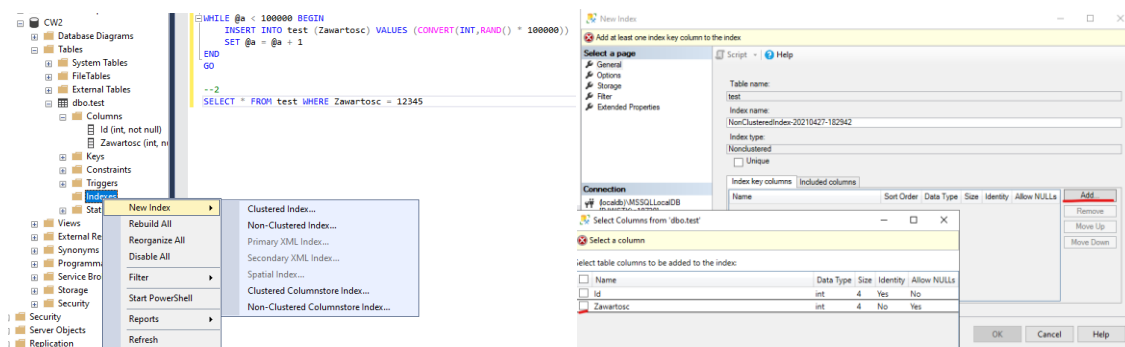
Podgląd planu wykonania zapytania (ctrl + L)



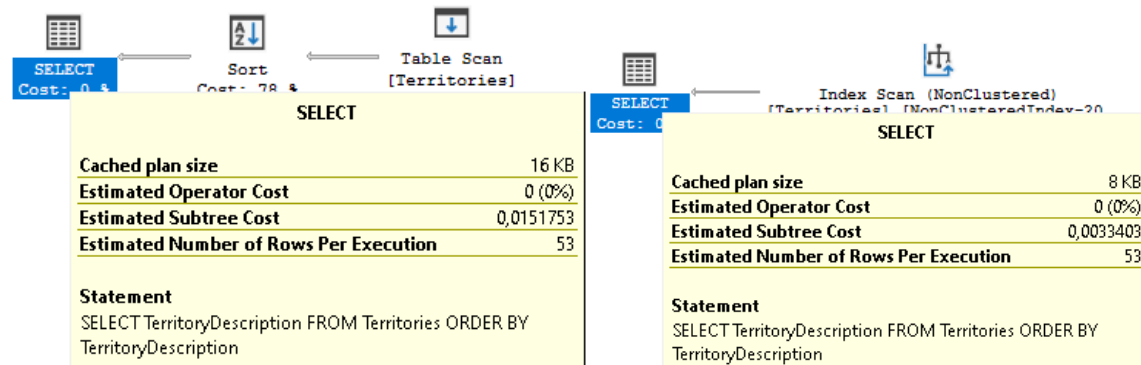
Po wykonaniu polecenia SQL obserwujemy koszt w zakładce „Execution plan” (przed i po indeksie)



Zakładamy indeks niegrupowany (Tabela -> Indexes -> New index)



### 2. Strategia „tylko indeks”



Poprzez założenie niegrupowanego indeksu na wszystkie kolumny z zapytania (jeden indeks na wszystkie) jesteśmy w stanie pominąć odczytywanie stron z danymi.



### 3. Strategia „tylko index” z included columns

<p><b>SELECT</b></p> <p>Cost: 0.00</p> <p>Clustered Index Scan (Clustered) [Orders].[PK Orders]</p> <p><b>SELECT</b></p> <p>Cached plan size 16 KB Estimated Operator Cost 0 (0%) Estimated Subtree Cost 0,0182691 Estimated Number of Rows Per Execution 40</p> <p><b>Statement</b> SELECT CustomerId, ShipCountry FROM Orders WHERE ShipCountry = 'Austria'</p>	<p><b>SELECT</b></p> <p>Cost: 0.00</p> <p>Index Seek (NonClustered) [Orders].[NonClusteredIndex-20210427-185552]</p> <p><b>SELECT</b></p> <p>Cached plan size 16 KB Estimated Operator Cost 0 (0%) Estimated Subtree Cost 0,003326 Estimated Number of Rows Per Execution 40</p> <p><b>Statement</b> SELECT CustomerId, ShipCountry FROM Orders WHERE ShipCountry = 'Austria'</p>
---	---

New Index

Ready

Select a page

- General
- Options
- Storage
- Filter
- Extended Properties

Script Help

Table name: Orders

Index name: NonClusteredIndex-20210427-185552

Index type: Nonclustered

☐ Unique

Index key columns Included columns

Name Data Type Identity Allow NULLs

Add... Remove Move Up Move Down

OK Cancel Help

Select Columns from 'dbo.Orders'

Select a column

Select table columns to be added to the index:

	Name	Data Type	Identity	Allow NULLs
<input type="checkbox"/>	Name			
<input type="checkbox"/>	OrderID	int	Yes	No
<input checked="" type="checkbox"/>	CustomerID	nchar(5)	No	Yes
<input type="checkbox"/>	EmployeeID	int	No	Yes
<input type="checkbox"/>	OrderDate	datetime	No	Yes
<input type="checkbox"/>	RequiredDate	datetime	No	Yes
<input type="checkbox"/>	ShippedDate	datetime	No	Yes

### 4. Indeks pogrupowany

Indeks pogrupowany nadaje się lepiej do zapytań z zakresami niż niepogrupowany.

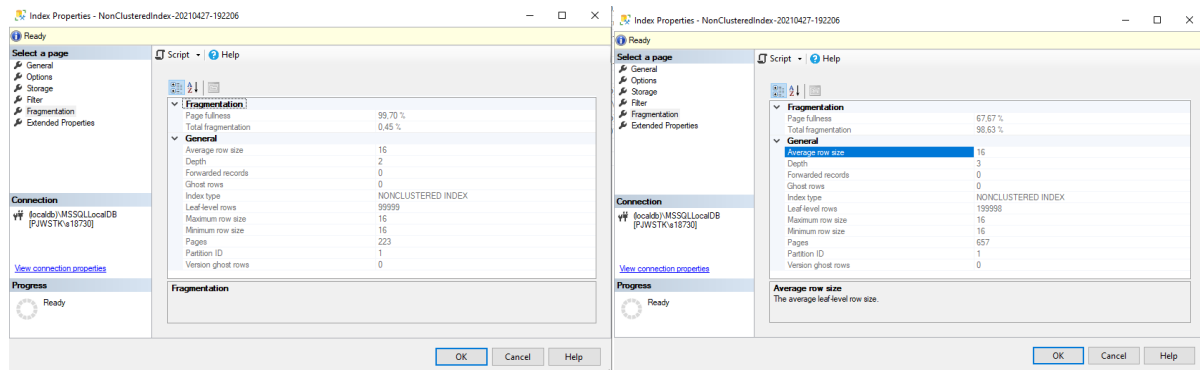
<p><b>SELECT</b></p> <p>Cost: 100.00</p> <p>Table Scan [test]</p> <p><b>SELECT</b></p> <p>Cached plan size 24 KB Estimated Operator Cost 0 (0%) Degree of Parallelism 0 Estimated Subtree Cost 0,295503 Estimated Number of Rows Per Execution 10037,2</p> <p><b>Statement</b> SELECT * FROM [test] WHERE [Zawartosc] &gt;=@1 AND [Zawartosc] &lt;=@2</p>	<p><b>SELECT</b></p> <p>Cost: 100.00</p> <p>Clustered Index Seek (Clustered) [test].[ClusteredIndex-20210427-190...]</p> <p><b>SELECT</b></p> <p>Cached plan size 24 KB Estimated Operator Cost 0 (0%) Degree of Parallelism 0 Estimated Subtree Cost 0,0321006 Estimated Number of Rows Per Execution 10037,2</p> <p><b>Statement</b> SELECT * FROM [test] WHERE [Zawartosc] &gt;=@1 AND [Zawartosc] &lt;=@2</p>
---	---

## 4. Statystyki w Messages

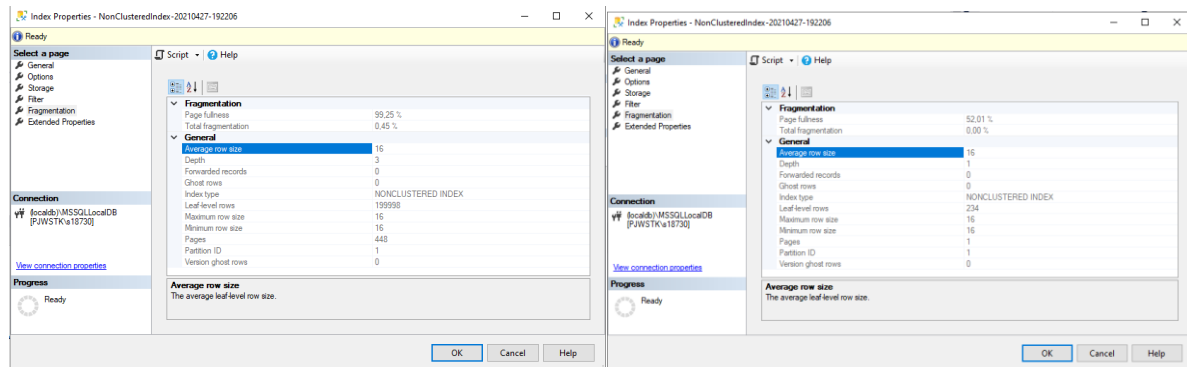
SET STATISTICS IO ON  
SET STATISTICS TIME ON

## 5. Właściwości indeksów

(Przed i po dodaniu wartości)



Po reorganizacji i rebudowie



Rebuild – Total fragmentation > 30%

Reorganize – Total fragmentation 10%-30%

No action – Total fragmentation < 10%

## TEORIA

*Indeksy pogrupowane* – max 1 na tabelę, uporządkowane dane, domyślnie dla PK

*Indeksy niegrupowane* – nieograniczona ilość, niegrupowane

*Indeksy pogrupowane* daje większą wydajność, są do sortowania, zakresów, mniej selektywnych danych

*Indeksy niegrupowane* są często przy kluczach obcych, dobry do warunków z pojedynczymi wierszami (WHERE)

## BLOK 3 - TRANSAKcje

1. Załóż bazę danych. Zaloguj się do tej bazy, używając Management Studio.

Wykonuj kolejno instrukcje:

```
SET IMPLICIT_TRANSACTIONS ON
CREATE TABLE Osoba (Id INT, Nazwisko VARCHAR(50))
COMMIT
INSERT INTO Osoba VALUES (1, 'Lenkiewicz')
INSERT INTO Osoba VALUES (2, 'Kowalski')
SELECT * FROM Osoba -- powinniśmy zobaczyć dwie osoby
ROLLBACK
SELECT * FROM Osoba -- powinniśmy zobaczyć pustą tabelę
INSERT INTO Osoba VALUES (3, 'Iksiński')
COMMIT
SELECT * FROM Osoba -- powinniśmy zobaczyć jedną osobę
```

The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays a SQL query window with the following code:

```
SET IMPLICIT_TRANSACTIONS ON
CREATE TABLE Osoba (Id INT, Nazwisko VARCHAR(50))
COMMIT
INSERT INTO Osoba VALUES (1, 'Lenkiewicz')
INSERT INTO Osoba VALUES (2, 'Kowalski')
SELECT * FROM Osoba -- powinniśmy zobaczyć dwie osoby
ROLLBACK
SELECT * FROM Osoba -- powinniśmy zobaczyć pustą tabelę
INSERT INTO Osoba VALUES (3, 'Iksiński')
COMMIT
SELECT * FROM Osoba -- powinniśmy zobaczyć jedną osobę
```

The bottom pane shows the results of the queries. The first result set, after the first SELECT, shows two rows:

Id	Nazwisko
1	Lenkiewicz
2	Kowalski

The second result set, after the ROLLBACK, shows an empty table.

Id	Nazwisko
----	----------

The third result set, after the second SELECT, shows one row:

Id	Nazwisko
3	Iksiński

SET IMPLICIT\_TRANSACTIONS ON – w tle uruchamia BEGIN TRANSACTION, dlatego, trzeba wyraźnie zaznaczać, że robimy commit czy rollback.

2. Spróbuj wykonać zadanie 1 ponownie, ale z wyłączoną opcją: SET IMPLICIT\_TRANSACTIONS OFF

Przy wykonaniu z opcją OFF wywala błąd, ponieważ commit i rollback są nie mają korespondującego bloku BEGIN TRANSACTION.

*Przed wykonaniem kolejnego zadania ustaw tą opcję z powrotem na ON.*

*W trakcie wykonywania instrukcji COMMIT i ROLLBACK pojawią się błędy mówiące o tym, że nie można zatwierdzić lub wycofać transakcji. Jest to normalne, ponieważ przy wyłączonej opcji IMPLICIT\_TRANSACTIONS, każda transakcja jest zatwierdzana automatycznie. Kolejne zadania wykonuj z włączoną tą opcją.*

3. Otwórz drugie okno MS. Wykonaj:

SET IMPLICIT\_TRANSACTIONS ON

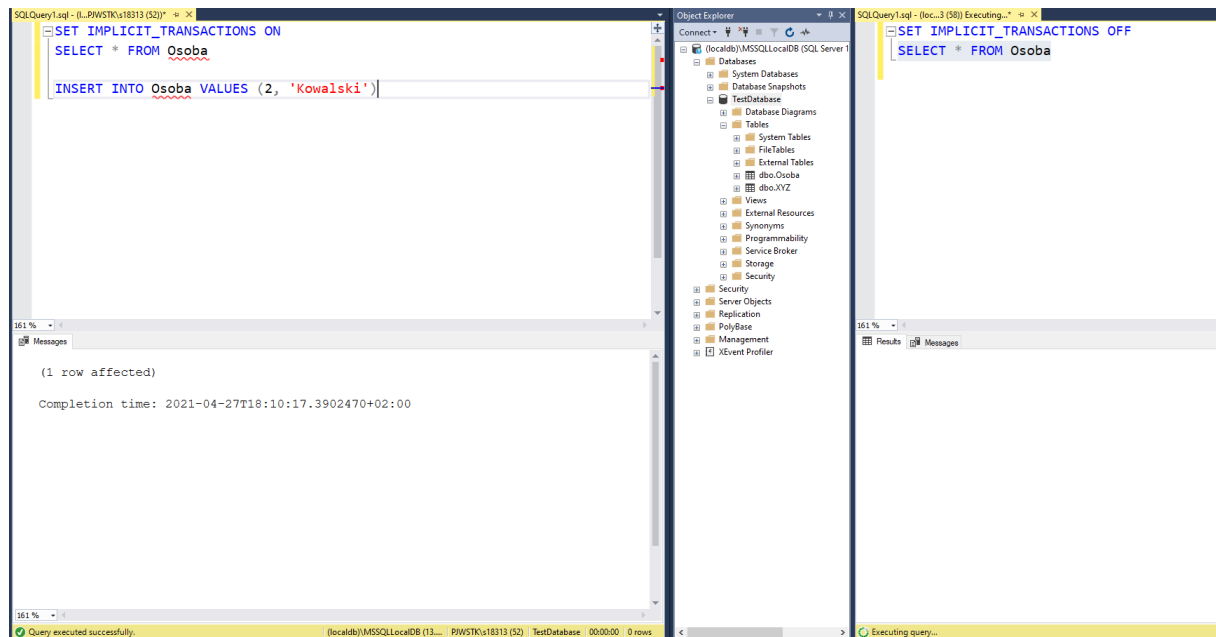
SELECT \* FROM Osoba

*W oknie 1 wstaw nowy rekord:*

INSERT INTO Osoba VALUES (2, 'Kowalski')

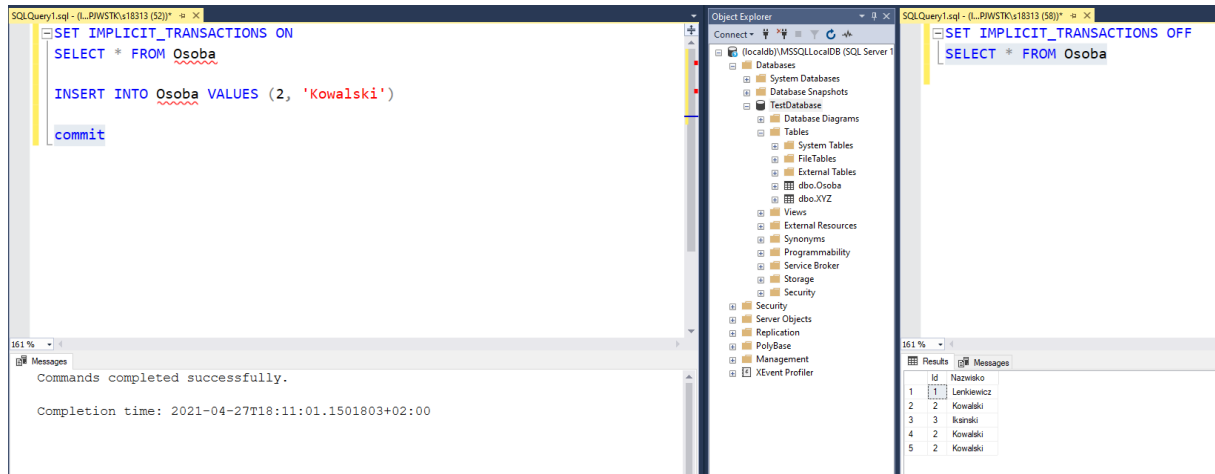
*W oknie 2 spróbuj wykonać jeszcze raz:*

SELECT \* FROM Osoba



Zapytanie będzie czekać na zwolnienie blokady. W oknie 1 wykonaj:

**COMMIT**



Zobacz, czy wykonało się zapytanie z okna 2.

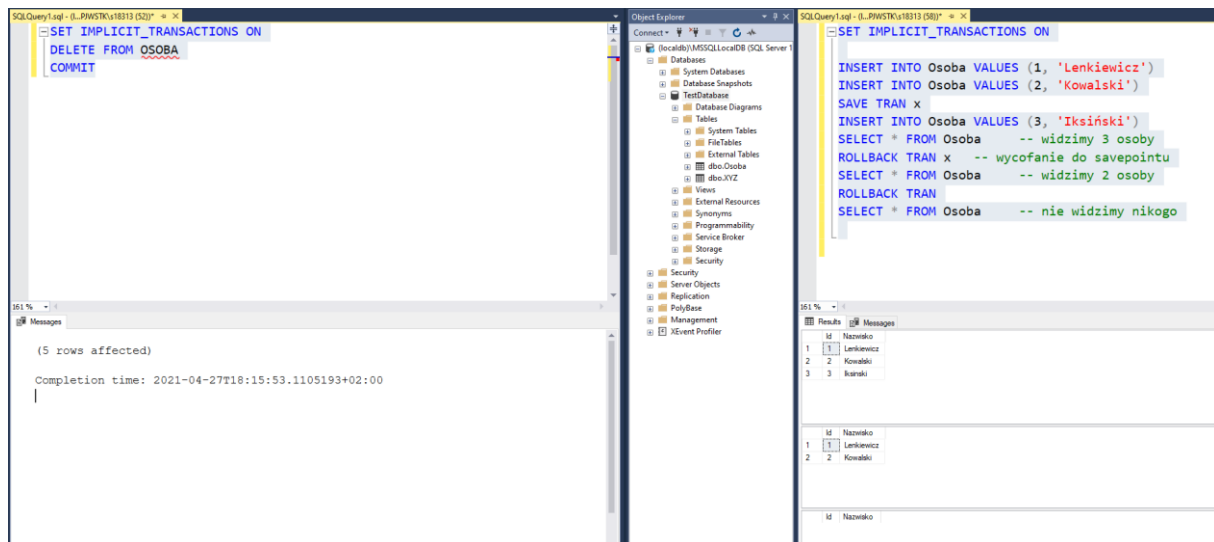
**UWAGA:** Przy tym i kolejnych ćwiczeniach wykonywanych przy `IMPLICIT_TRANSACTIONS` ustawionym na `ON` mogą przeszkadzać nam poprzednie, niezakończone transakcje. Dlatego warto przed każdym ćwiczeniem wykonać `COMMIT` w każdym oknie.

4. W oknie 1 najpierw wyczyść tabelę:

```
DELETE FROM Osoba  
COMMIT
```

Wstawiamy 2 rekordy i robimy SAVEPOINT:

```
INSERT INTO Osoba VALUES (1, 'Lenkiewicz')  
INSERT INTO Osoba VALUES (2, 'Kowalski')  
SAVE TRAN x  
INSERT INTO Osoba VALUES (3, 'Iksiński')  
SELECT * FROM Osoba -- widzimy 3 osoby  
ROLLBACK TRAN x -- wycofanie do savepointu  
SELECT * FROM Osoba -- widzimy 2 osoby  
ROLLBACK TRAN  
SELECT * FROM Osoba -- nie widzimy nikogo
```



## SAVEPOINT

Punkt, do którego możemy wycofać transakcję, przykład:

```
BEGIN TRAN
```

```
INSERT INTO Osoba (Id, Nazwisko) VALUES (1, 'Kowalski')
```

```
INSERT INTO Osoba (Id, Nazwisko) VALUES (2, 'Lenkiewicz')
```

```
SAVE TRAN x
```

```
INSERT INTO Osoba (Id, Nazwisko) VALUES (3, 'Nowak')
```

```
SELECT * FROM Osoba -- widzimy 3 osoby
```

```
ROLLBACK TRAN x
```

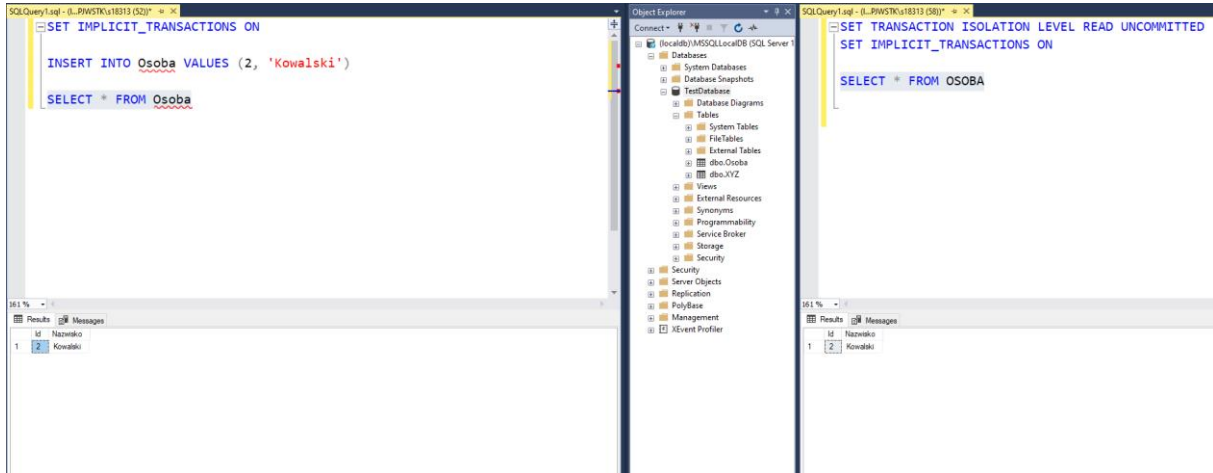
```
SELECT * FROM Osoba -- widzimy 2 osoby
```

```
ROLLBACK TRAN
```

```
SELECT * FROM Osoba -- nie widzimy nikogo
```

5. W drugim oknie MS ustaw poziom izolacji na najniższy:  
**SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED**

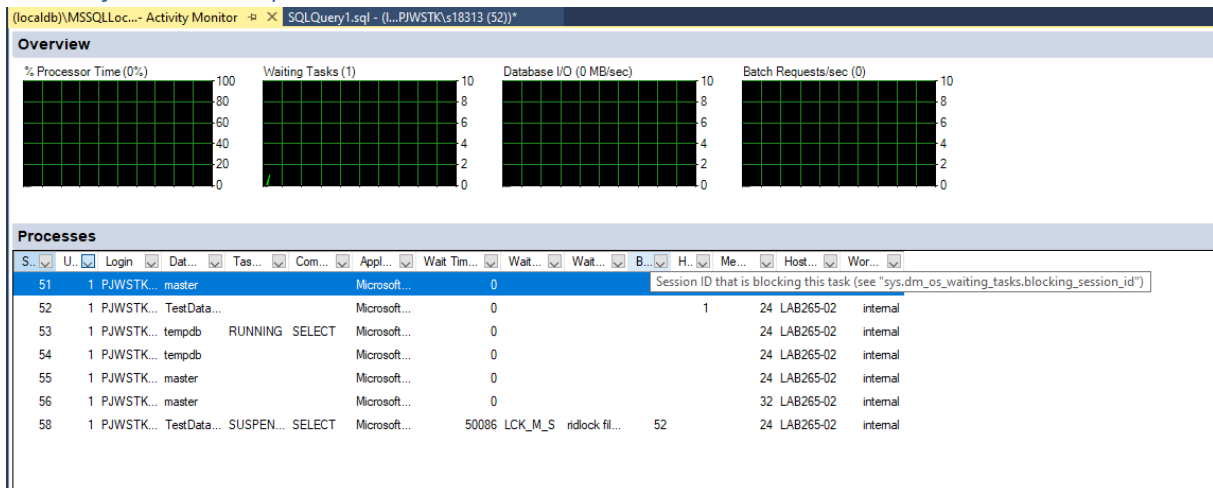
Wykonaj ponownie zadanie 3 (lub podobne). W pierwszym oknie wprowadzaj zmiany, a w drugim zobacz, czy jesteś w stanie odczytać niezatwierdzone dane.



Można odczytać bez robienia commita w pierwszym oknie. Z neta: The uncommitted read isolation level allows an application to access the uncommitted changes of other transactions.

*UWAGA: Ustawienie poziomu izolacji działa dopiero dla kolejnej transakcji i nie dotyczy transakcji, która już się rozpoczęła. Dlatego po instrukcji SET TRANSACTION ISOLATION LEVEL ... warto dla pewności wykonać COMMIT lub ROLLBACK.*

6. Powrót do poziomu izolacji READ COMMITTED. Wykonaj kilka operacji na dwóch oknach (np. INSERT, DELETE) podglądając w narzędziu Activity Monitor, jakie blokady zakłada serwer.



Przykład blokady, by uruchomić activity monitor trzeba prawym klawiszem myszy na bazę. By doprowadzić do tego trzeba zrobić to samo co w zadaniu 3.

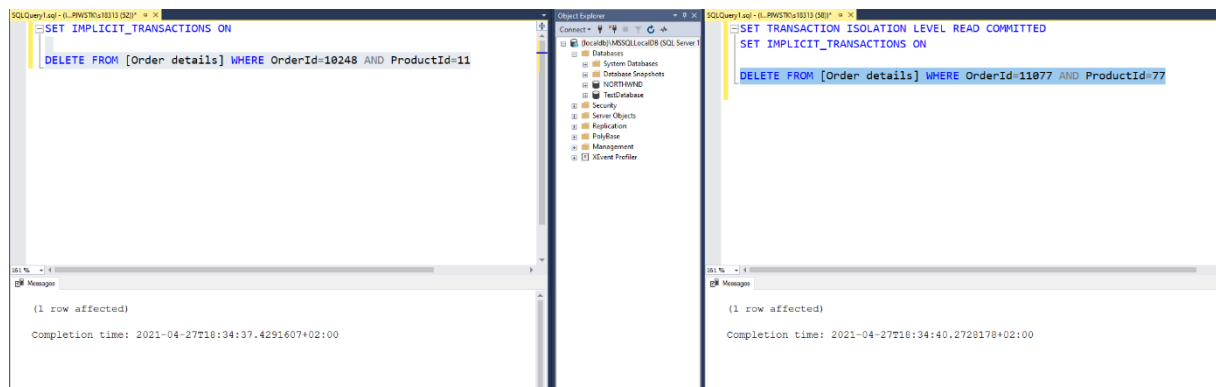
7. Do kolejnego zadania potrzebujemy tabeli z większą ilością danych. Przełącz się na bazę *Northwind* w obu oknach. Opcja *IMPLICIT\_TRANSACTIONS* powinna być włączona. Obejrzyj jak wygląda tabela *Order details*. Następnie wykonaj w oknie 1:

*DELETE FROM [Order details] WHERE OrderId=10248 AND ProductId=11*

a w oknie 2:

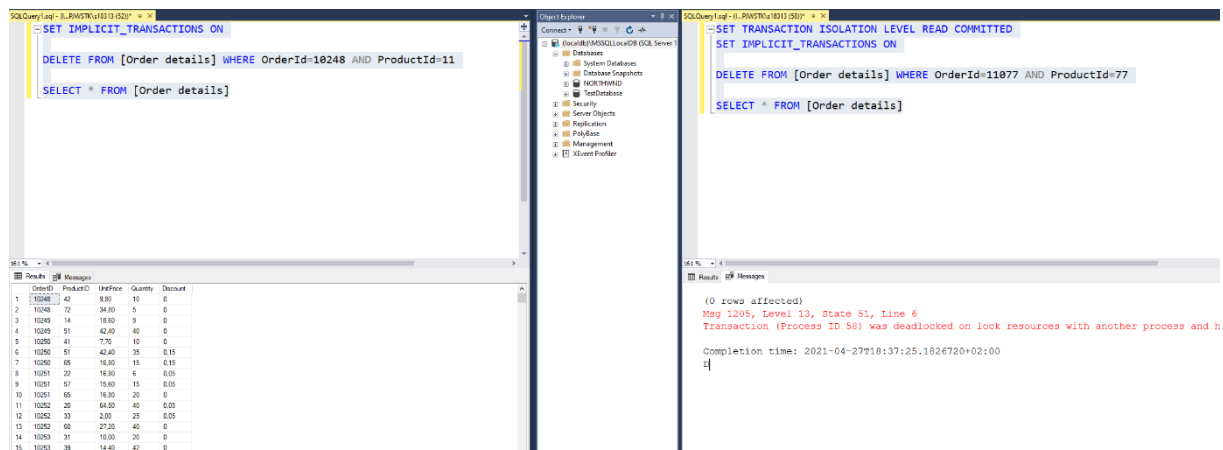
*DELETE FROM [Order details] WHERE OrderId=11077 AND ProductId=77*

Zauważ, że udało się wykonać równocześnie dwie operacje na tej samej tabeli, ponieważ serwer zakłada blokady tylko na wybrane wiersze lub strony. Nie powinno się jednak zapytanie:



*SELECT \* FROM [Order details]*

do momentu wykonania *COMMIT*, ponieważ wymaga dostępu do całej tabeli.



Doprowadziło to do deadlocka, ponieważ delete from blokuje wiersz w jednej tabeli, a select \* from tabela potrzebuje całej tabeli. Jeżeli przykładowo mamy pierwszą transakcję, która blokuje dane w tabeli A i próbuje zmodyfikować dane w tabeli B oraz drugą transakcję, która blokuje dane w tabeli B i próbuje zmodyfikować dane w tabeli A to pojawia się deadlock



KOD:

OKNO1

```
SET IMPLICIT_TRANSACTIONS ON
```

```
DELETE FROM [Order details] WHERE OrderId=10248 AND ProductId=11
```

```
SELECT * FROM [Order details]
```

OKNO2

```
SET TRANSACTION ISOLATION LEVEL READ COMMITTED
```

```
SET IMPLICIT_TRANSACTIONS ON
```

```
DELETE FROM [Order details] WHERE OrderId=11077 AND ProductId=77
```

```
SELECT * FROM [Order details]
```

8. Wróć do swojej bazy danych w obu oknach. Sprawdź, czy istnieje tabela „Osoba” i czy zawiera jakieś rekordy. Jeżeli nie, utwórz ją przy pomocy instrukcji z punktu 1. Nie zapomnij wykonać COMMIT po zakończeniu. Następnie ustaw poziom izolacji SERIALIZABLE w obu oknach.

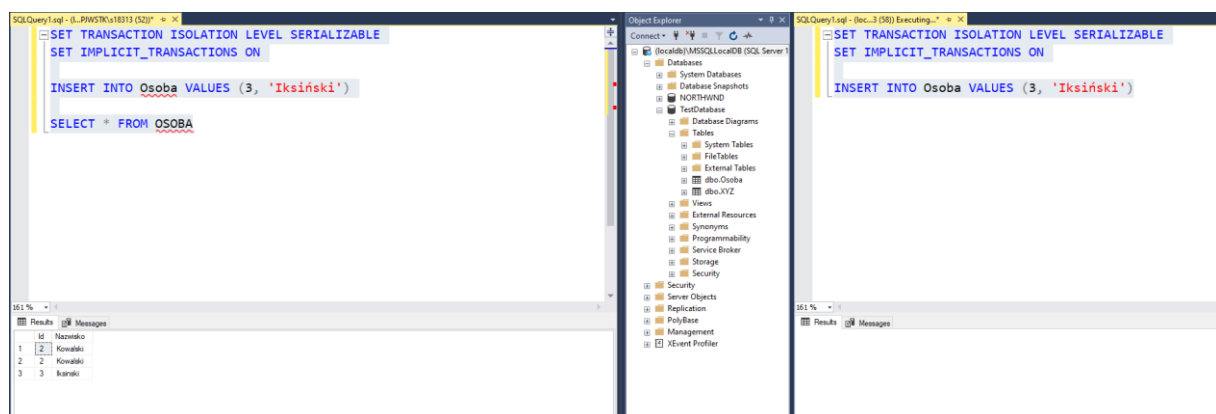
```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
```

Następnie wykonaj w obu oknach:

```
SELECT * FROM Osoba
```

A następnie w pierwszym oknie dowolny INSERT:

```
INSERT INTO Osoba VALUES (3, 'Iksiński')
```



Zauważ, że tym razem, już wykonanie instrukcji SELECT spowodowało założenie blokady na całą tabelę „Osoba” i nie jesteśmy w stanie wykonać INSERT do momentu, gdy w oknie 2 wykonamy COMMIT. W przypadku READ COMMITTED instrukcja INSERT wykonałaby się od razu (fantom).

9. Doprowadź do sytuacji, w której wystąpi problem niepowtarzalnego odczytu.

First query:

```
set transaction isolation level read committed
begin transaction
select * from _tmp
```

Second

```
update _tmp set id = 2
```

Create and populate table

```
create table _tmp ( id int)
insert _tmp (id )values(1)
```

Run first query, then run second then run `select` statement from first query

You'll see `1` first time and `2` second

*set transaction isolation level read committed*

*begin transaction*

*select \* from \_tmp*

*update \_tmp set id = 2*

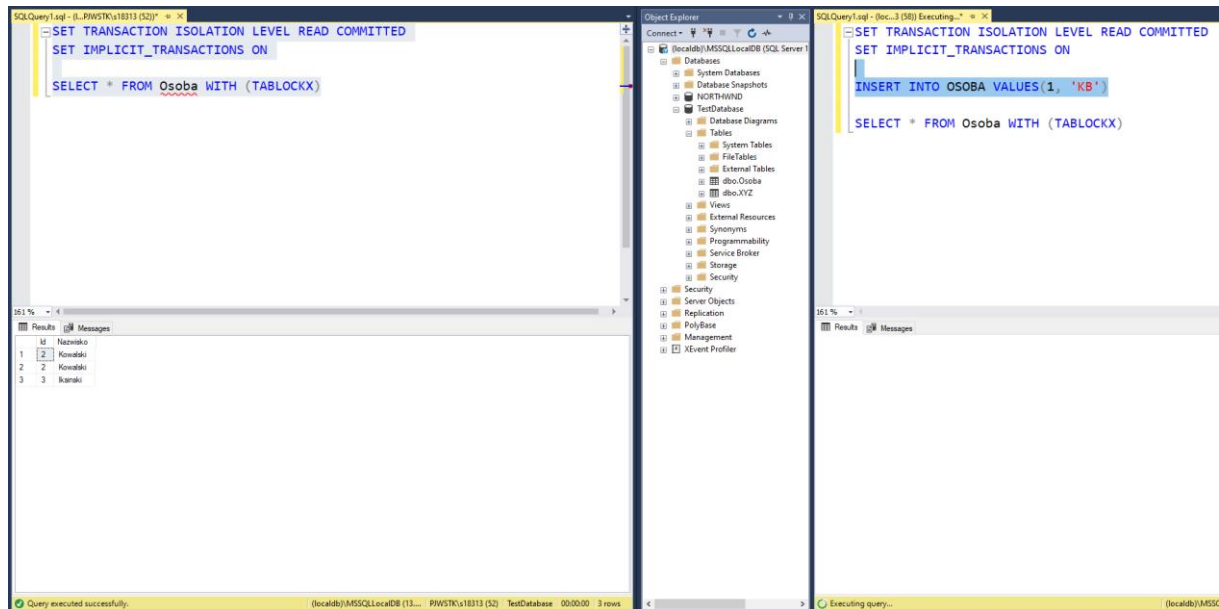
*create table \_tmp ( id int)*

*insert \_tmp (id )values(1)*

10. Powrót do poziomu izolacji READ COMMITTED w obu oknach. Wykonaj instrukcję SELECT z użyciem locking hint wymuszającym założenie wyłącznej blokady na tabeli:

**SELECT \* FROM Osoba WITH (TABLOCKX)**

Zobacz, czy w drugim oknie jesteś w stanie wykonać jakąkolwiek operację na tabeli (np. SELECT). Możesz również podejrzeć założone blokady w Activity Monitor.



Nie jestem w stanie wykonać żadnych operacji, jest blok na tabeli.

Processes											
S...	U...	Login	Dat...	Tas...	Com...	App...	Wait Tim...	Wait...	Wait...	B...	H...
51	1	PJPSTK...	master			Microsoft...	0				24
52	1	PJPSTK...	TestData...			Microsoft...	0			1	24
53	1	PJPSTK...	tempdb	RUNNING	SELECT	Microsoft...	0				24
54	1	PJPSTK...	master			Microsoft...	0				24
56	1	PJPSTK...	master			Microsoft...	0				32
58	1	PJPSTK...	TestData...	SUSPEN...	INSERT	Microsoft...	60801	LCK_M_IX	objectloc...	52	24

11. Testujemy migawkowy poziom izolacji. Aby przygotować tabelę do ćwiczenia, wykonaj:

```
DELETE FROM Osoba
INSERT INTO Osoba VALUES (1, 'Lenkiewicz')
INSERT INTO Osoba VALUES (2, 'Kowalski')
INSERT INTO Osoba VALUES (3, 'Iksinski')
COMMIT
```

Następnie włącz możliwość ustawiania poziomu izolacji SNAPSHOT:

```
ALTER DATABASE nazwa_bazy SET allow_snapshot_isolation ON
```

Ustaw ten poziom izolacji:

```
SET TRANSACTION ISOLATION LEVEL SNAPSHOT
SET IMPLICIT_TRANSACTIONS ON
```

a następnie wypisz wszystkie rekordy:

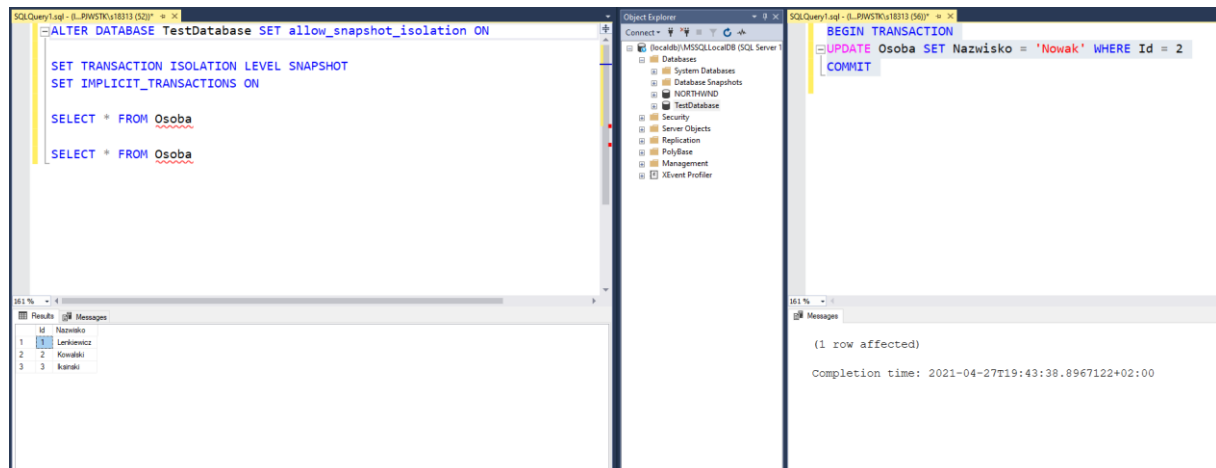
```
SELECT * FROM Osoba
```

W oknie 2 wykonaj:

```
UPDATE Osoba SET Nazwisko = 'Nowak' WHERE Id = 2
COMMIT
```

W oknie 1:

```
SELECT * FROM Osoba
```



# Migawki bazy danych

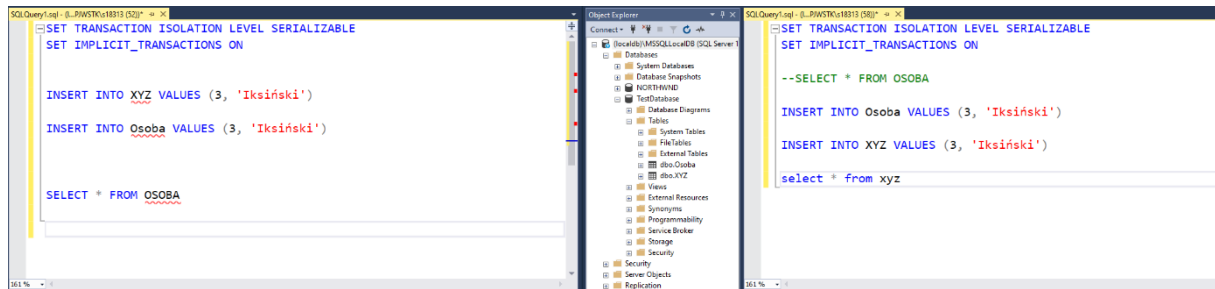
- Są statycznym obrazem bazy danych z pewnego punktu w czasie, tylko do odczytu.
- Nadają się do zastosowań raportowych, analiz danych itp.
- Mogą stanowić zabezpieczenie przed błędem i źródło danych do ewentualnego odtworzenia danych po pomyłkach.
- Mogą służyć do przechowywania stanu bazy danych w wybranych punktach w czasie.
- Są dostępne tylko w edycji ENTERPRISE.

*Zobaczysz starą, nieaktualną zawartość tabeli, gdyż transakcja pracuje na migawce. Spróbuj zaktualizować ten sam rekord:*

*UPDATE Osoba SET Nazwisko = 'Iksinski' WHERE Id = 2*

*Powinieneś zauważyć błąd: „Snapshot isolation transaction aborted due to update conflict...”. Nie da się zatwierdzić transakcji, gdyż w trakcie jej działania dane zostały zmienione.*

## 12. Czy potrafisz doprowadzić do zakleszczenia (deadlock)?



Okno1:

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
```

```
SET IMPLICIT_TRANSACTIONS ON
```

```
INSERT INTO XYZ VALUES (3, 'Iksiński')
```

```
INSERT INTO Osoba VALUES (3, 'Iksiński')
```

```
SELECT * FROM OSOBA
```

Okno2:

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
```

```
SET IMPLICIT_TRANSACTIONS ON
```

```
INSERT INTO Osoba VALUES (3, 'Iksiński')
```

```
INSERT INTO XYZ VALUES (3, 'Iksiński')
```

```
select * from xyz
```

Poziomy Izolacji

## Poziomy izolacji

Standard ANSI/ISO definiuje cztery *poziomy izolacji* określające, czy transakcje widzą zmiany dokonywane przez inne współbieżnie działające transakcje.

Wszystkie cztery poziomy są implementowane w MS SQL Server.

Poziom izolacji	Niezatwierdzony odczyt	Niepowtarzalny odczyt	Fantomy
Read Uncommitted	TAK	TAK	TAK
Read Committed	NIE	TAK	TAK
Repeatable Reads	NIE	NIE	TAK
Serializable	NIE	NIE	NIE

## BLOK 4 - UPRAWNIENIA

1. Załóż konto logowania (login) w trybie SQL Server Authentication i na jego podstawie utwórz użytkownika (user) w swojej bazie danych. Spróbuj zalogować się tym użytkownikiem do Management Studio i wybrać swoją bazę. Sprawdź, czy możesz wykonywać instrukcje SQL na tabelach bazy danych.

*Efekt ćwiczenia powinien być taki, że zalogujemy się do serwera, ale nie będziemy w stanie wykonać żadnej operacji.*

Security <całej instancji> -> logins -> ppm -> new login

<baza danych> -> security -> users -> ppm -> new user

(jak robimy konto sql server authentications to łącząc się musimy zmienić sposób logowania)

Mamy użytkownika, ale nie mamy żadnych uprawnień

2. Załóż użytkownika w systemie Windows. Na jego podstawie utwórz konto logowania (login). Wyloguj się z systemu Windows i zaloguj używając nowego użytkownika. Uruchom Management Studio i użyj Windows authentication. Sprawdź, czy możesz wykonywać instrukcje SQL na tabelach bazy danych.

*Efekt powinien być podobny do punktu pierwszego. Aby nie było konieczne wylogowanie / zalogowanie w systemie operacyjnym, można użyć metody „run as”.*

Najszybciej użytkownika założymy poprzez zarządzanie komputerem (na labach nie zrobimy, na wirtualkach zrobimy)

Security <całej instancji> -> logins -> ppm -> new login

3. Zaloguj się z powrotem na konto administratora. Przy pomocy Management Studio nadaj uprawnienia do wykonywania np. instrukcji SELECT użytkownikowi z punktu 1. Zaloguj się w drugim oknie Management Studio jako ten użytkownik. Sprawdź, czy możesz wykonywać instrukcje SQL.

Aby dodać uprawnienia Userowi:

User PPM -> properties -> securables -> search (ja wziąłem all obj in schema dbo XD) -> nadajemy uprawnienia na dole

Uprawnienia możemy nadawać na różnych poziomach:

Dla konta logowania (*login*):

- Na poziomie serwera

Dla użytkownika (*user*):

- Na poziomie bazy danych
- Do obiektów (tabel, procedur, perspektyw...)
- Dla roli
- Do schematu

## Uprawnienia na poziomie serwera



Dotyczą konta logowania (*login*). Niektóre z nich to:

- **Alter any database** – pozwala na modyfikowanie ustawień dowolnej bazy danych. Umożliwia również zakładanie i usuwanie baz danych.
- **Alter any login** – pozwala modyfikować i zakładać konta logowania.
- **Connect SQL** – pozwala na połączenie się z serwerem (domyślnie włączone dla każdego konta).
- **Create any database** – pozwala zakładać bazy danych.

Przykład nadania tego typu uprawnień z poziomu SQL:

```
GRANT ALTER ANY DATABASE TO login
```

## Uprawnienia na poziomie bazy danych



Między innymi:

- **CREATE TABLE** – pozwala tworzyć tabele
- **CREATE VIEW** – tworzyć perspektywy
- **ALTER ANY USER** – pozwala tworzyć i modyfikować użytkowników
- **ALTER** – pozwala tworzyć i modyfikować dowolne obiekty
- **BACKUP DATABASE** – pozwala wykonywać kopię zapasową
- **BACKUP LOG** – pozwala wykonywać kopię zapasową dziennika transakcji
- **EXECUTE** – pozwala wykonywać wszystkie procedury składowane
- **SELECT** – pozwala wykonywać SELECT na wszystkich obiektach
- **INSERT** – pozwala wykonywać INSERT na wszystkich obiektach
- **UPDATE** – pozwala wykonywać UPDATE na wszystkich obiektach
- **DELETE** – pozwala wykonywać DELETE na wszystkich obiektach
- **SHOW PLAN** – pozwala oglądać plan wykonania zapytania

Dla tabel, perspektyw, procedur:

- **SELECT** (tabela, kolumna)
- **INSERT** (tabela)
- **UPDATE** (tabela, kolumna)
- **DELETE** (tabela)
- **EXECUTE** (procedura)
- **ALTER** (tabela, procedura)
- **CONTROL** (tabela, procedura),
- **REFERENCES** (tabela)
- **TAKE OWNERSHIP** (tabela, procedura)
- **VIEW DEFINITION** (tabela, procedura)



4. Załóż jeszcze jedno konto logowania i użytkownika oraz nadaj dowolne uprawnienia, lecz tym razem nie używaj graficznego interfejsu, lecz wykonaj wszystko przy pomocy odpowiednich instrukcji SQL i/lub procedur sp\_.

○ Konto typu Windows authentication:

`sp_grantlogin` lub

`CREATE LOGIN nazwa FROM WINDOWS`

○ Konto typu SQL Server authentication:

`sp_addlogin` lub

`CREATE LOGIN nazwa WITH PASSWORD = 'hasło'`

Użytkowników (user) możemy tworzyć:

- używając Management Studio,
- z poziomu SQL

`sp_grantdbaccess` lub

`CREATE USER nazwa FOR LOGIN login`

5. Nadaj uprawnienia do wykonywania instrukcji UPDATE na wybranej kolumnie. Przetestuj te uprawnienia robiąc UPDATE na tej i innych kolumnach.

Jak w punkcie 3 komendy

6. Nadaj użytkownikowi prawo SELECT na poziomie bazy danych. Przetestuj to prawo.

*Przed kolejnymi zadaniami warto usunąć to prawo, aby użytkownik nie miał nadmiernych uprawnień.*

`GRANT SELECT TO Użytkownik`

7. Utwórz rolę w swojej bazie danych. Dodaj użytkownika z punktu 1 do tej roli. Nadaj roli uprawnienia inne niż w poprzednich punktach. Zobacz, czy użytkownik będzie mógł wykonywać operacje wynikające z roli i z jego własnych uprawnień.

Role można założyć tylko dla baz danych

`sp_addrolemember <rola>, <user> -- albo graficznie w properties usera`

8. Zablokuj (DENY) wykonywanie jakiegś operacji na poziomie roli, lecz pozostaw to prawo (GRANT) na poziomie użytkownika. Sprawdź, czy użytkownik będzie mógł wykonać tę operację. Możesz wykonać to ćwiczenie w dowolnej konfiguracji np. 2 role, 2 role + uprawnienia użytkownika, DENY u użytkownika i GRANT w roli itp.

Deny na poziomie roli jest mocniejsze niż grant na poziomie usera

Rola -> properties -> securables -> search -> dodajemy to czego szukamy i nadajemy uprawnienia

User -> properties -> securables -> search -> dodajemy to czego szukamy i nadajemy uprawnienia

9. Dodaj użytkownika do roli db\_ddladmin. Sprawdź, czy może wykonać dowolną instrukcję DDL.

*sp\_addrolemember db\_ddladmin <user> (add role uruchomi się na bazie, nie na serwerze)*

Może XD

10. Utwórz schemat w swojej bazie danych. Przenieś jakiś obiekt bazy danych do nowego schematu. Spróbuj odwołać się do tego obiektu np. przy pomocy instrukcji SELECT.

*CREATE SCHEMA nazwa*

*ALTER SCHEMA nowy\_schemat TRANSFER dotychczasowy\_schemat.obiekt*

*SELECT \* from schemat.tabela*

11. Utwórz nowego użytkownika i nadaj mu uprawnienia do schematu, lecz nie nadawaj żadnych uprawnień na poziomie obiektów. Zaloguj się przy pomocy tego użytkownika i spróbuj wykonać instrukcję do której nadałeś uprawnienia.

User -> properties -> securables -> search -> all object of the types -> schemas -> wybieramy nasz schemat i na dole uprawnienia

Tak, ma uprawnienia do obiektów w schemacie

12. Zmień domyślny schemat użytkownika na utworzony w punkcie 10. Zaobserwuj różnicę w sposobie odwoływania się do obiektów.

User -> properties -> general -> default schema (nie trzeba szukać można wpisać na choma nazwę)

Można odwoływać się bez podawania schematu

## BLOK 5 – BACKUPY

### 1. Backup podstawowy

Prawy na bazę -> Tasks -> Back up

### 2. Wczytanie podstawowe

Wczytywanie Databases -> Restore -> Podaj ścieżkę pliku .bak

### 3. Wykonywanie backupu z kodu

Używając backup device:

BACKUP DATABASE nazwa\_bazy TO device

Bez backup device:

BACKUP DATABASE nazwa\_bazy TO DISK = „C:\backup\plik.bak”

Różnicowy:

BACKUP DATABASE nazwa\_bazy TO DISK = „C:\backup\plik.bak” WITH DIFFERENTIAL

Logu:

BACKUP LOG nazwa\_bazy TO DISK = „C:\backup\plik.bak”

### 4. Utworzenie backup device i zrobienie pierwszego backupu

Server objects -> prawym na backup devices -> New backup device

Podajemy nazwę i ścieżkę do pliku, klikamy okej i wykonujemy backup do backup device zamiast do pliku.

### 5. Teoria

Opcja NORECOVERY

Używana wtedy, gdy chcemy odtworzyć po kolei wiele backupów. Przez nią zmiany nie pojawią się od razu na bazie.

Procent zajętości oraz wielkość poszczególnych plików

Sprawdzamy za pomocą taska shrink