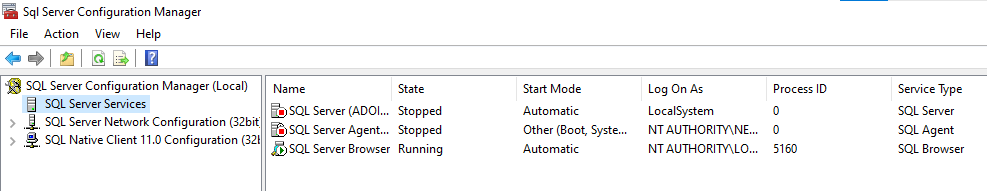
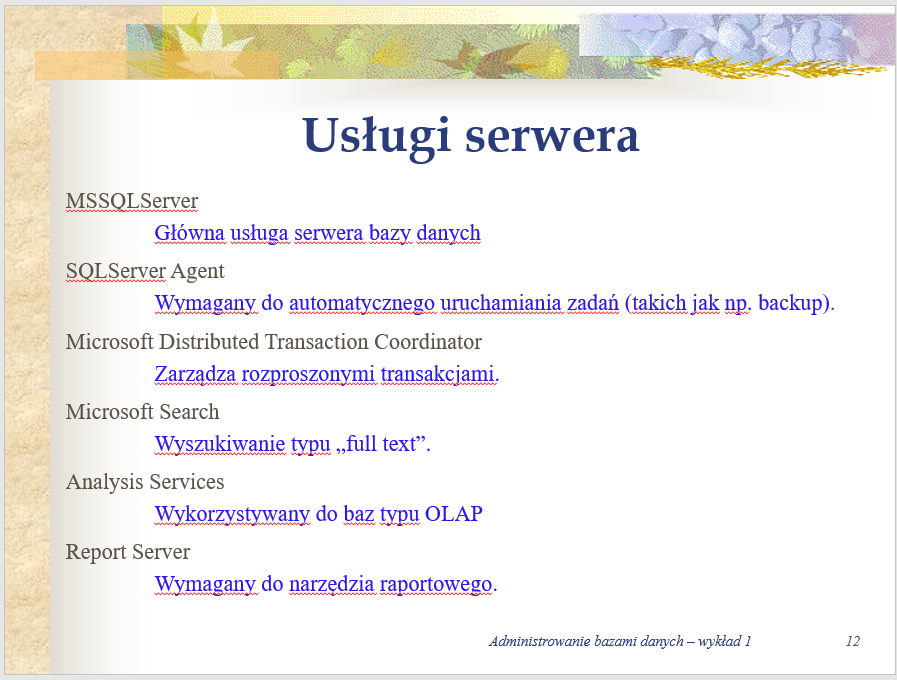
# BLOK 1 – INSTALACJA, PLIKI

## Przy pomocy Configuration Manager podejrzyj, jakie usługi SQL Servera są uruchomione.

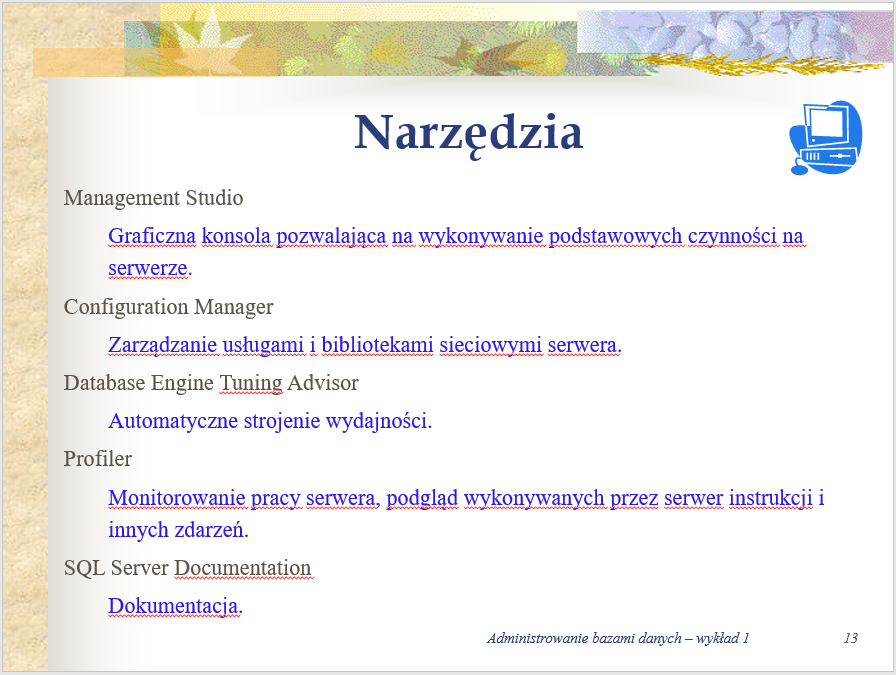
Path: C:\Windows\SysWOW64\SQLServerManager15.msc





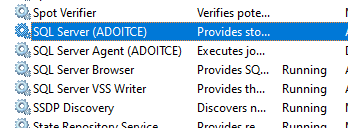
## Zapoznaj się z dostępnymi w zainstalowanym serwerze narzędziami oraz dokumentacją (Management Studio, Books OnLine).

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



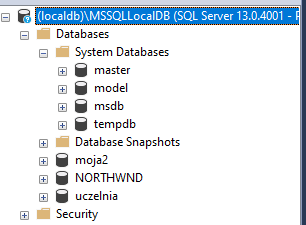
## Zapoznaj się z bazami danych dostępnymi na serwerze.

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

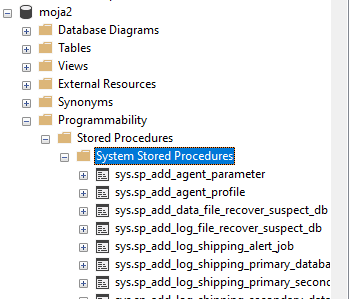
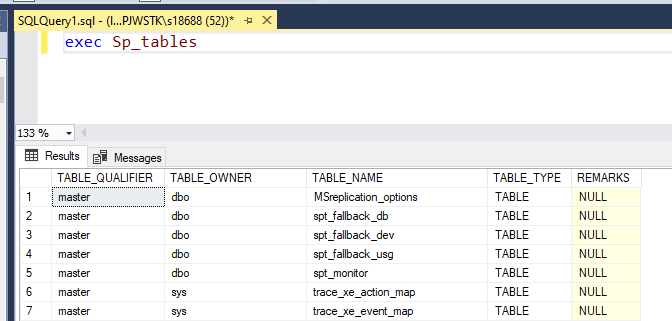


Bazy:

* 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 tworzymy 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



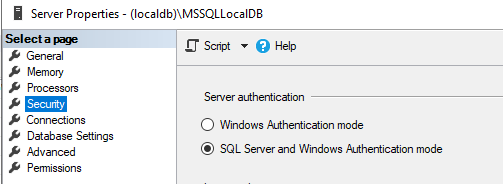
## Przetestuj procedury *sp\_* omówione w wykładzie.

* 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.

## Podejrzyj 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

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

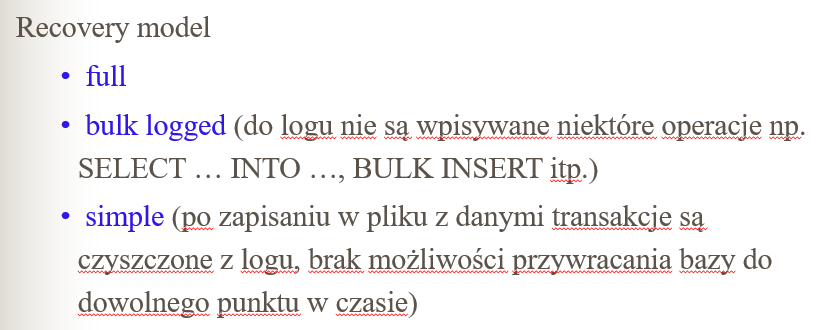
PPM na databases -> new databases

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:

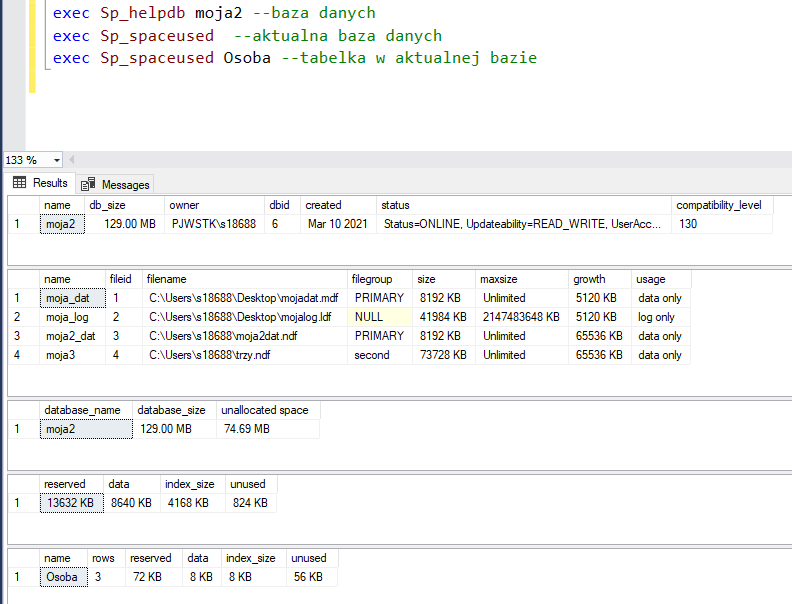


W 3 zakładce są filegroupy, po dodaniu filegroupy dopiero możemy cofnąć się do zakładki 1 i zmienić lokalizację plików (tylko dla nowych,

## Podejrzyj opcje bazy danych (baza danych -> *properties*).

Prawdopodobnie interesują nas tylko General, Files, Filegroups i Permissions

## Używając procedur *sp\_* wymienionych w wykładzie, podejrzyj właściwości swojej bazy danych.



## 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

## 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

## 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

## 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 grupę

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

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

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

## 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

## Usuń 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

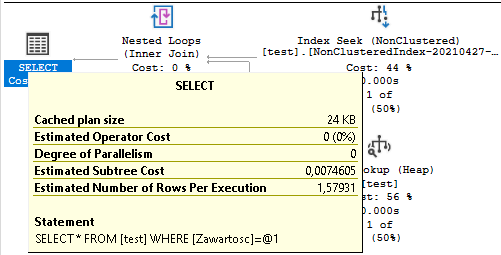
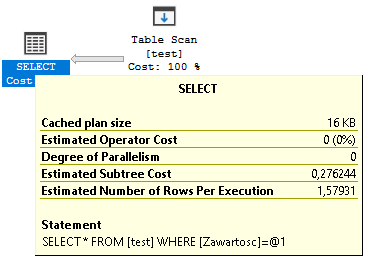
# BLOK 2 - INDEKSY

## 1. Indeks niepogrupowany

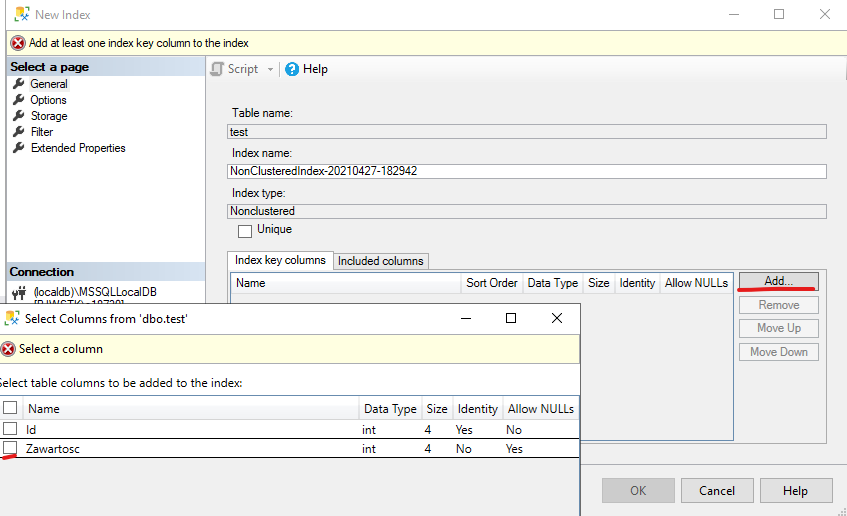
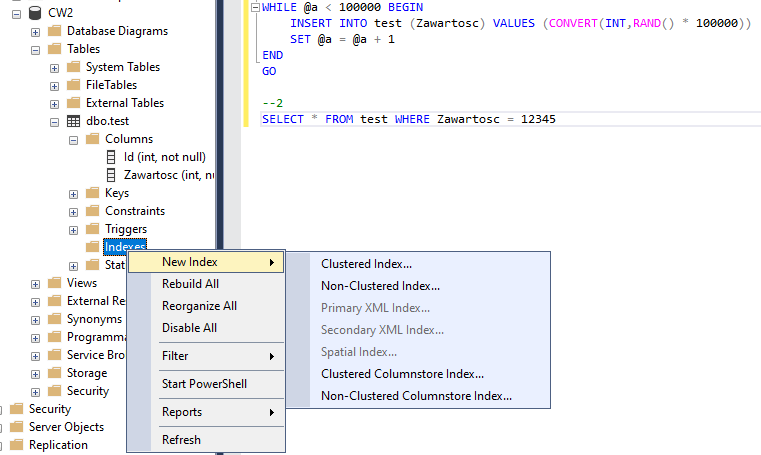
Podgląd planu wykonania zapytania (ctrl + L)



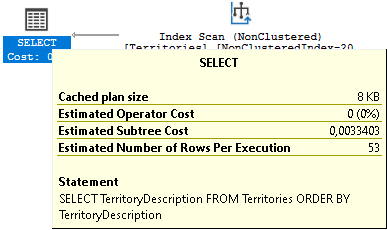
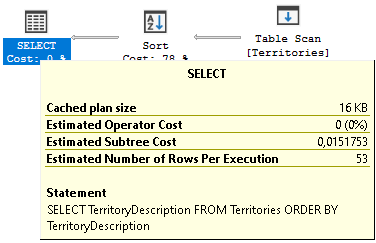
Po wykonaniu polecenia SQL obserwujemy koszt w zakładce „Execution plan” (przed i po indeksie)



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

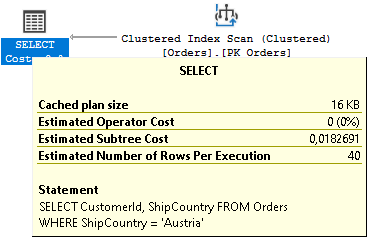
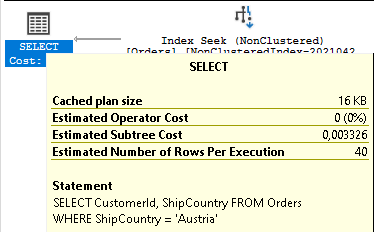


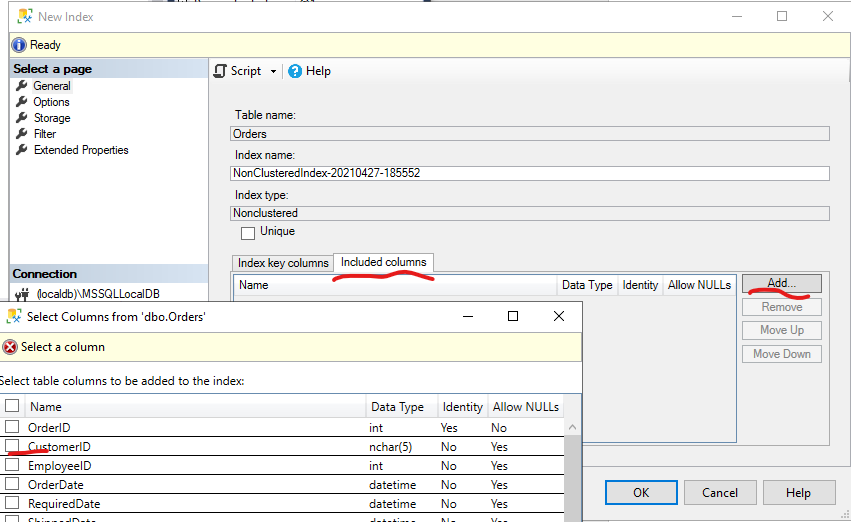
## 2. Strategia „tylko indeks”

****

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

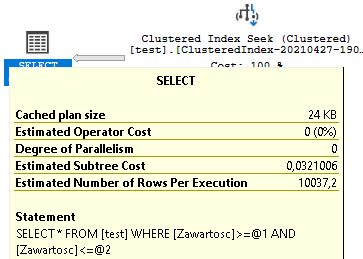
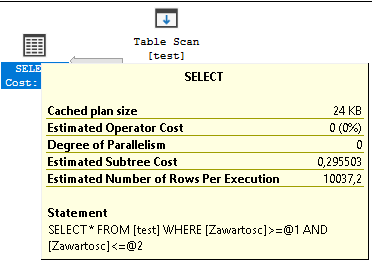
## 3. Strategia „tylko index” z included columns

****



## 4. Indeks pogrupowany

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

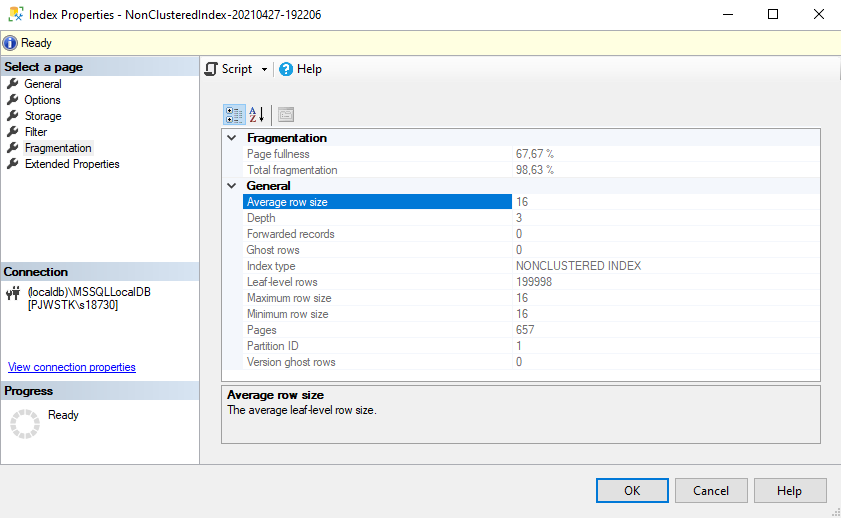
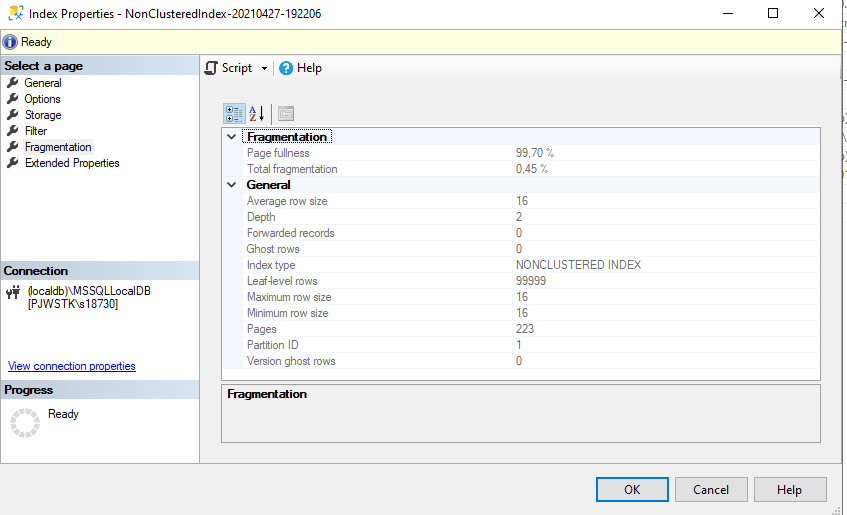


## 4. Statystyki w Messages

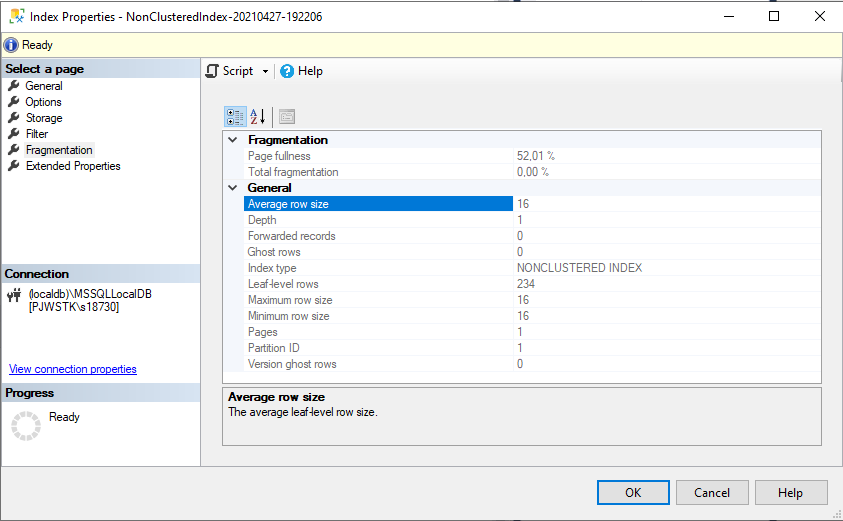
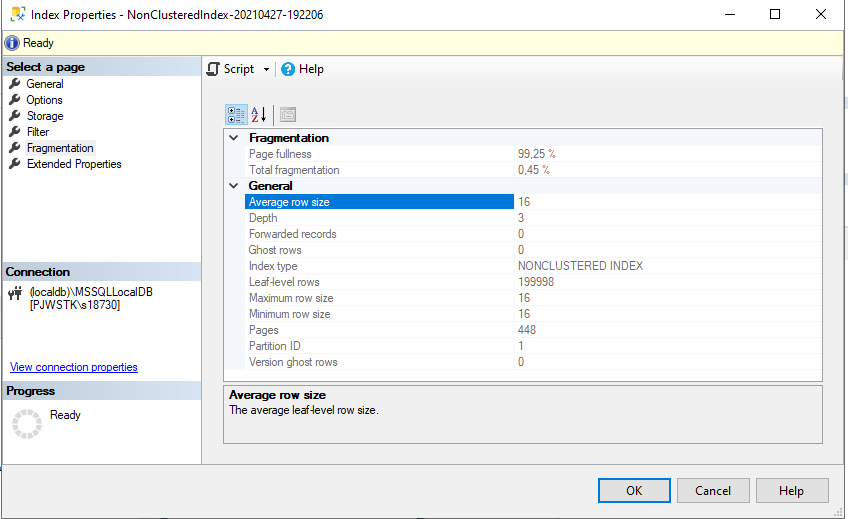
SET STATISTICS IO ON  
SET STATISTICS TIME ON

## 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 niepogrupowane* – nieograniczona ilość, niepogrupowane

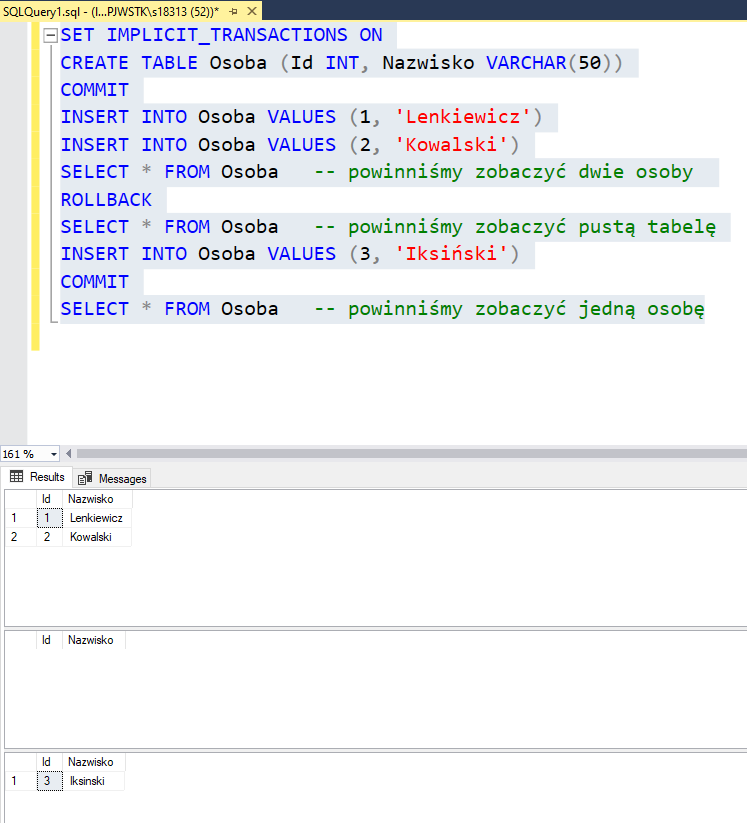
*Indeksy pogrupowane* daje większą wydajność, są do sortowania, zakresów, mniej selektywnych danych

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

# BLOK 3 - TRANSAKCJE

## 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



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

## 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.

## Otwórz drugie okno MS. Wykonaj:

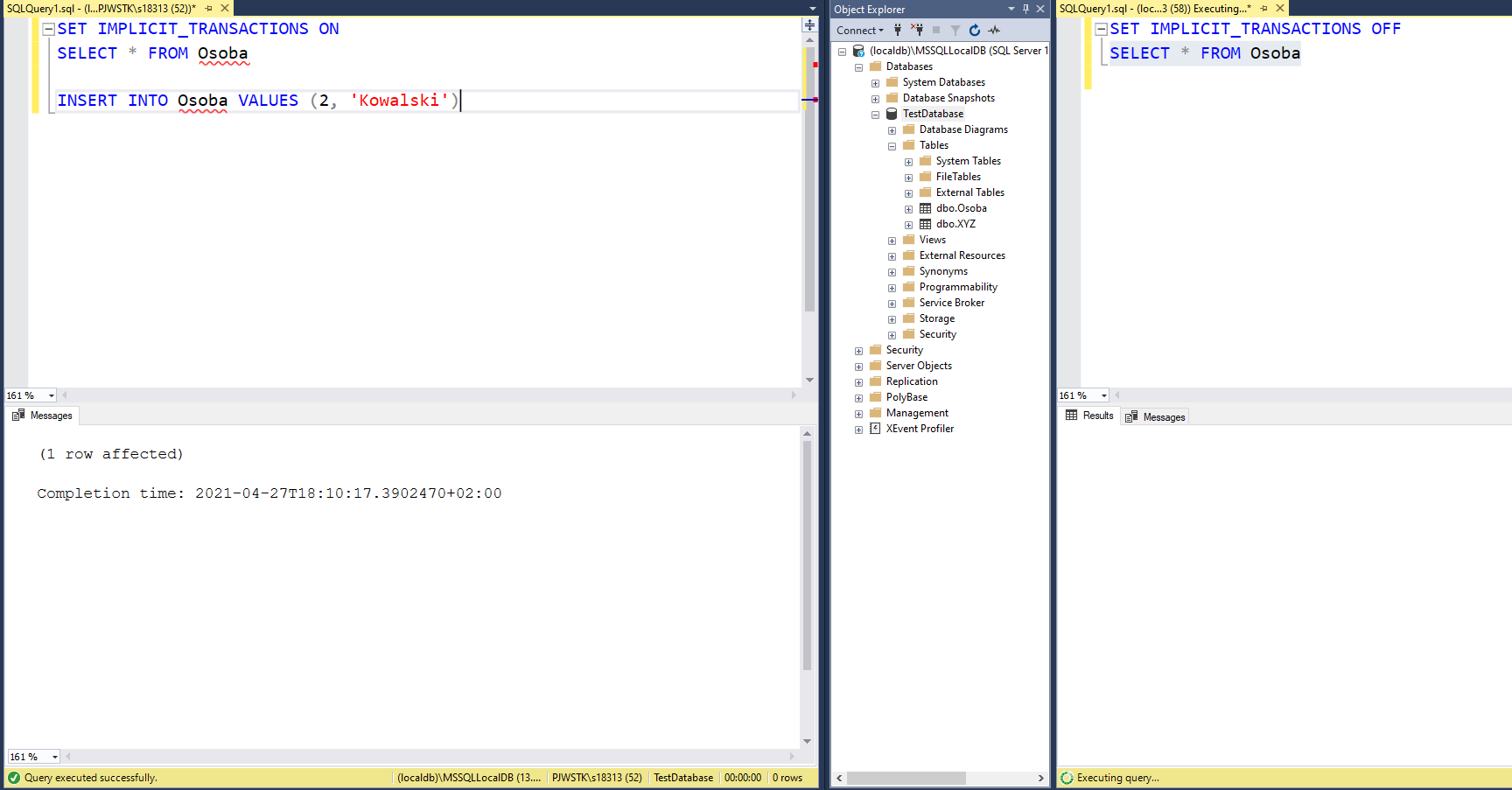
SET IMPLICIT\_TRANSACTIONS ON  
SELECT \* FROM Osoba

W oknie 1 wstaw nowy rekord:

INSERT INTO Osoba VALUES (2, ‘Kowalski’)

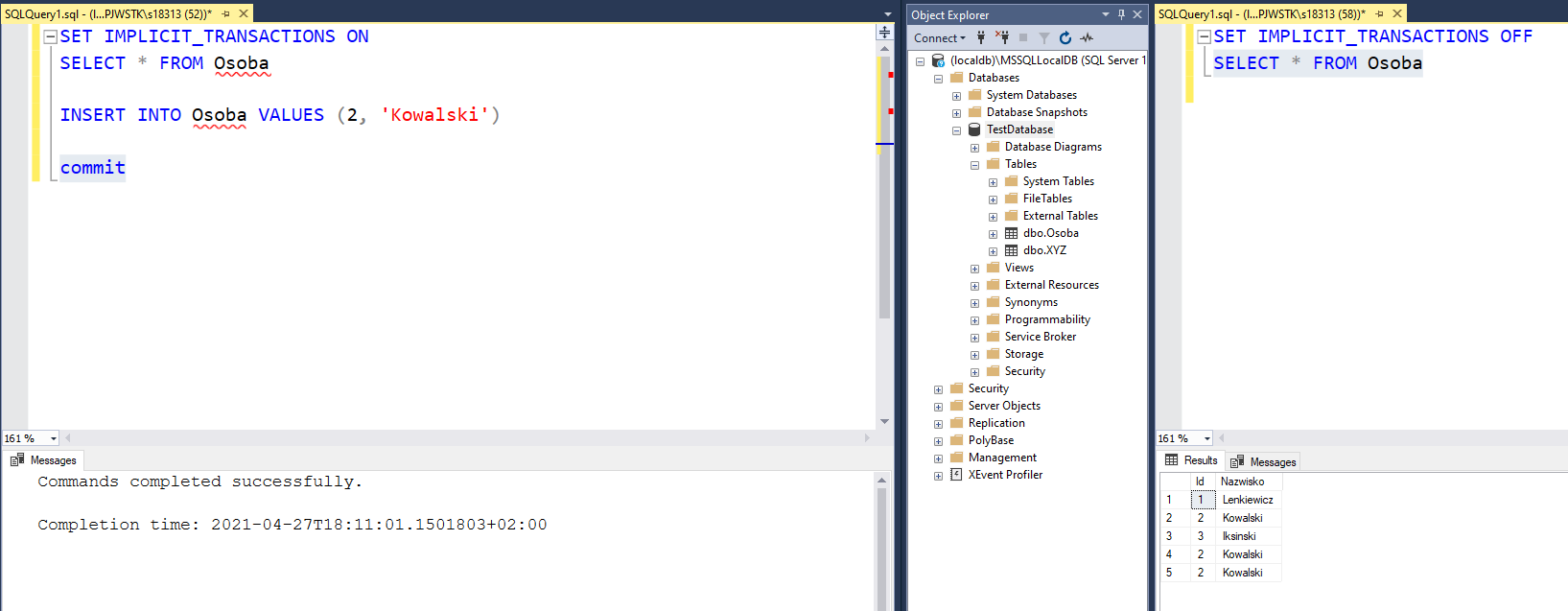
W oknie 2 spróbuj wykonać jeszcze raz:

SELECT \* FROM Osoba



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

COMMIT

  
  
Zobacz, czy wykonało się zapytanie z okna 2.

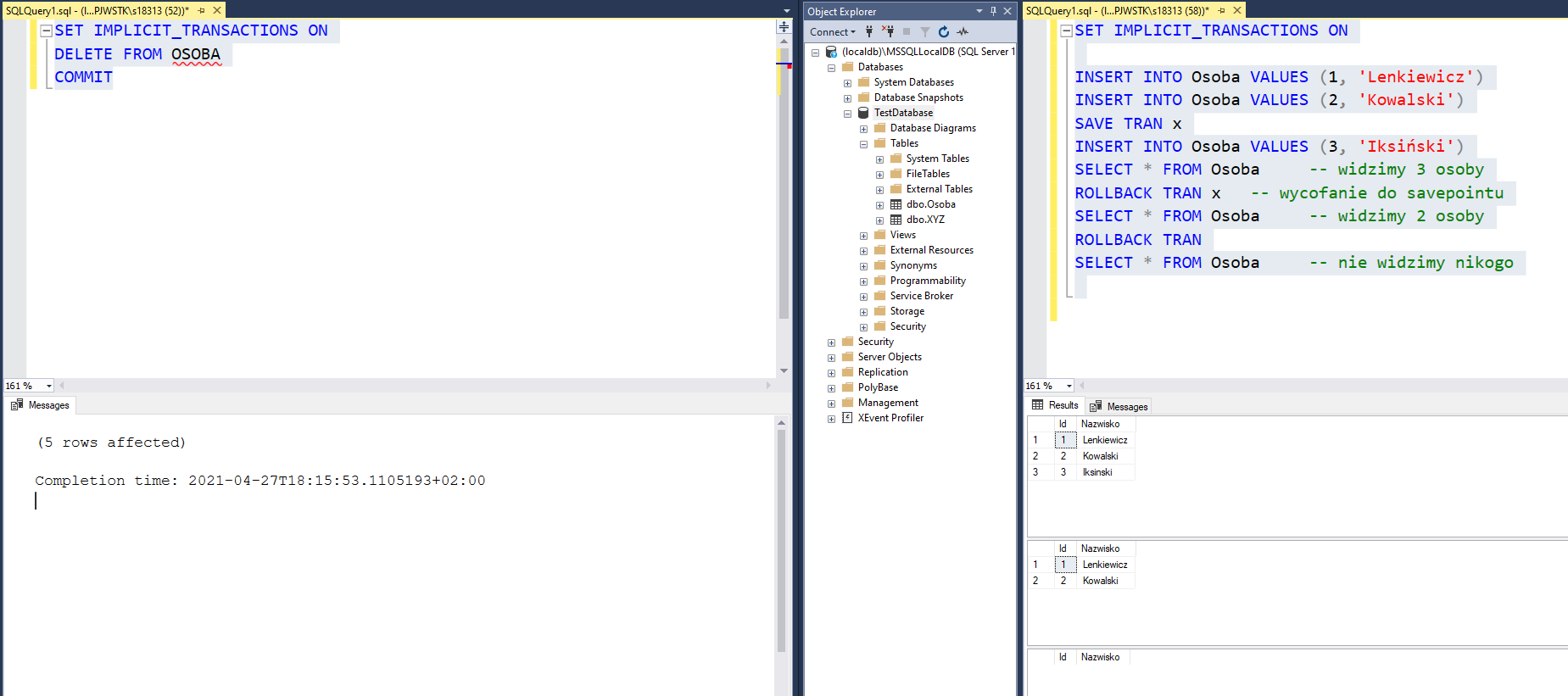
UWAGA: Przy tym i kolejnych ćwiczeniach wykonywanych przy IMPLICIT\_TRANSACTIONS ustawionym na ON mogą przeszkadzać nam poprzednie, niezakończone transakcje. Dlatego warto przed każdym ćwiczeniem wykonać COMMIT w każdym oknie.

## 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

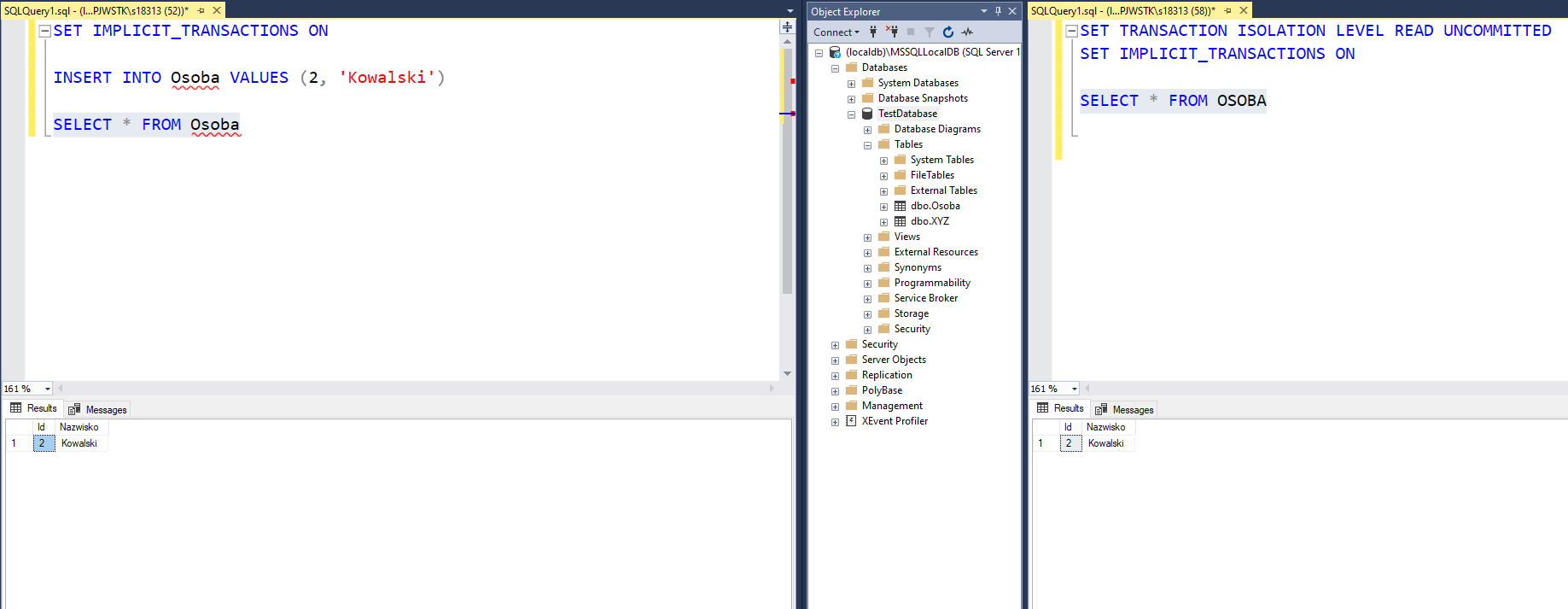




## W drugim oknie MS ustaw poziom izolacji na najniższy:

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED

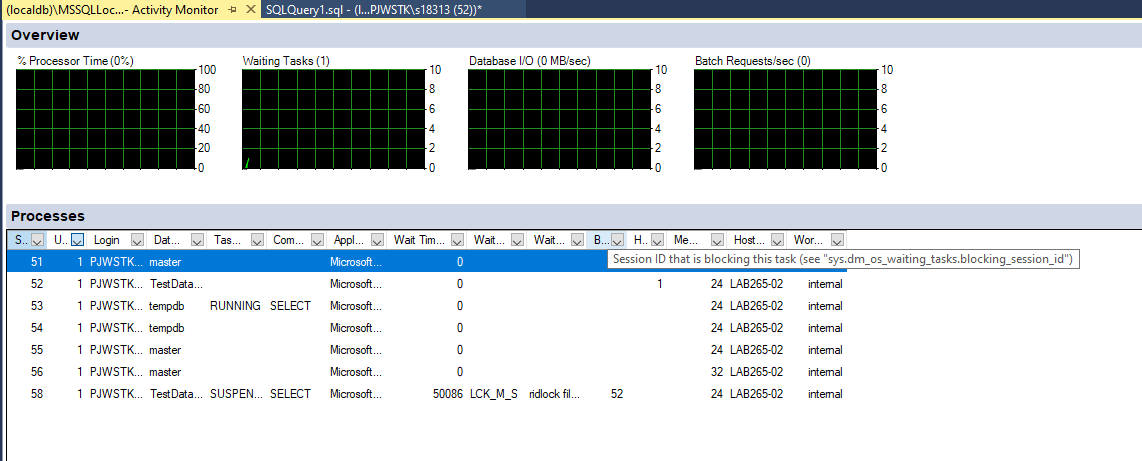
Wykonaj ponownie zadanie 3 (lub podobne). W pierwszym oknie wprowadzaj zmiany, a w drugim zobacz, czy jesteś w stanie odczytać niezatwierdzone dane.



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.

## Powróć 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.

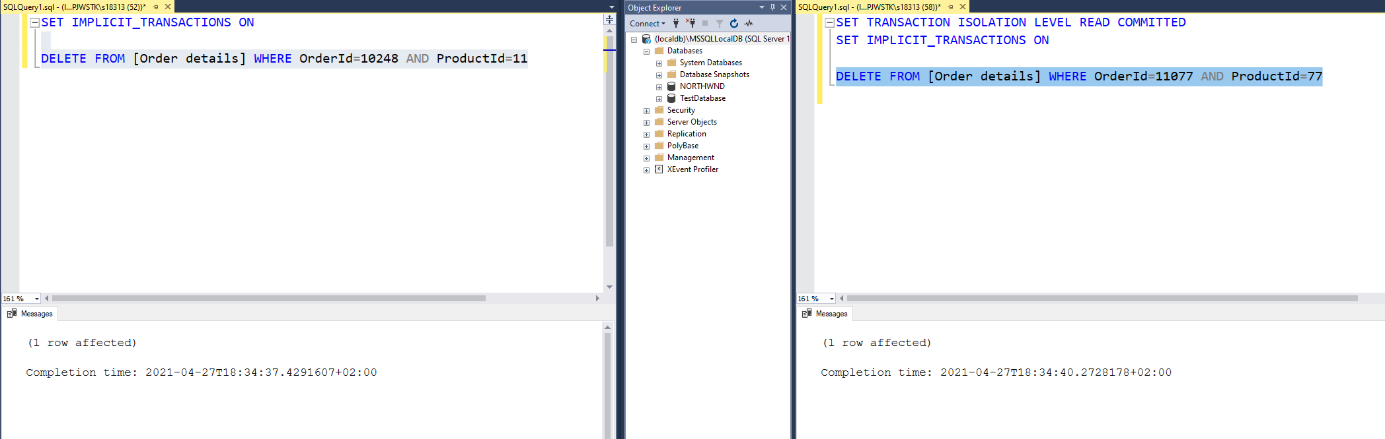
## Do kolejnego zadania potrzebujemy tabeli z większą ilością danych. Przełącz się na bazę *Northwind* w obu oknach. Opcja IMPLICIT\_TRANSACTIONS powinna być włączona. Obejrzyj jak wygląda tabela *Order details*. Następnie wykonaj w oknie 1:

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

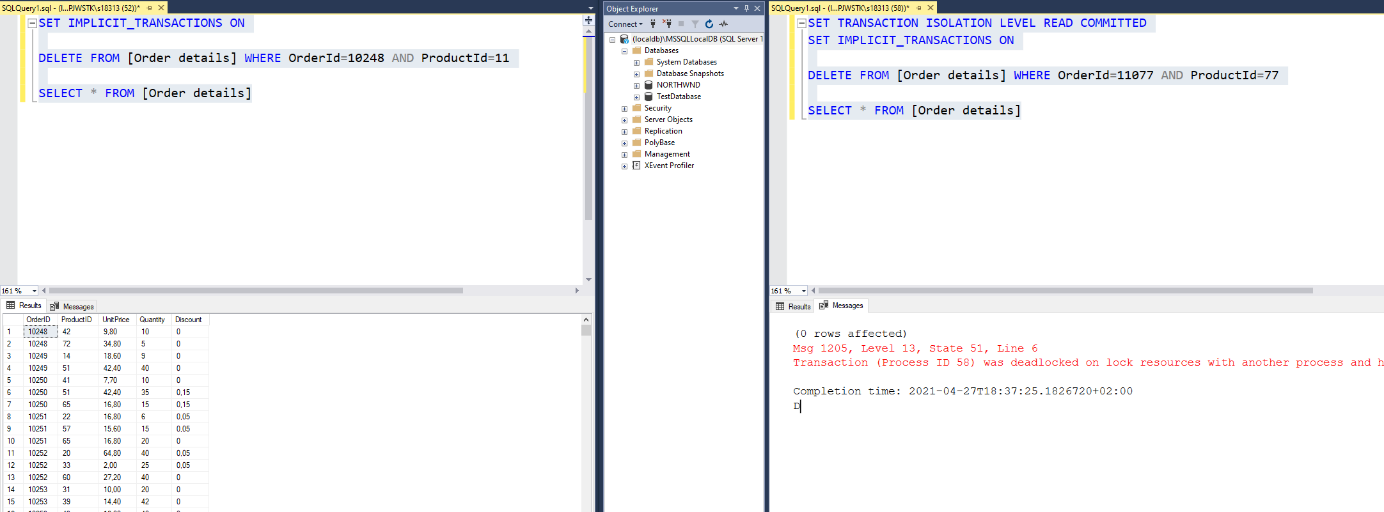
a w oknie 2:

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 udać zapytanie:

  
  
SELECT \* FROM [Order details]

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



Doprowadziło to do deadlocka, ponieważ delete from blokuje wiesz 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]

## 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.

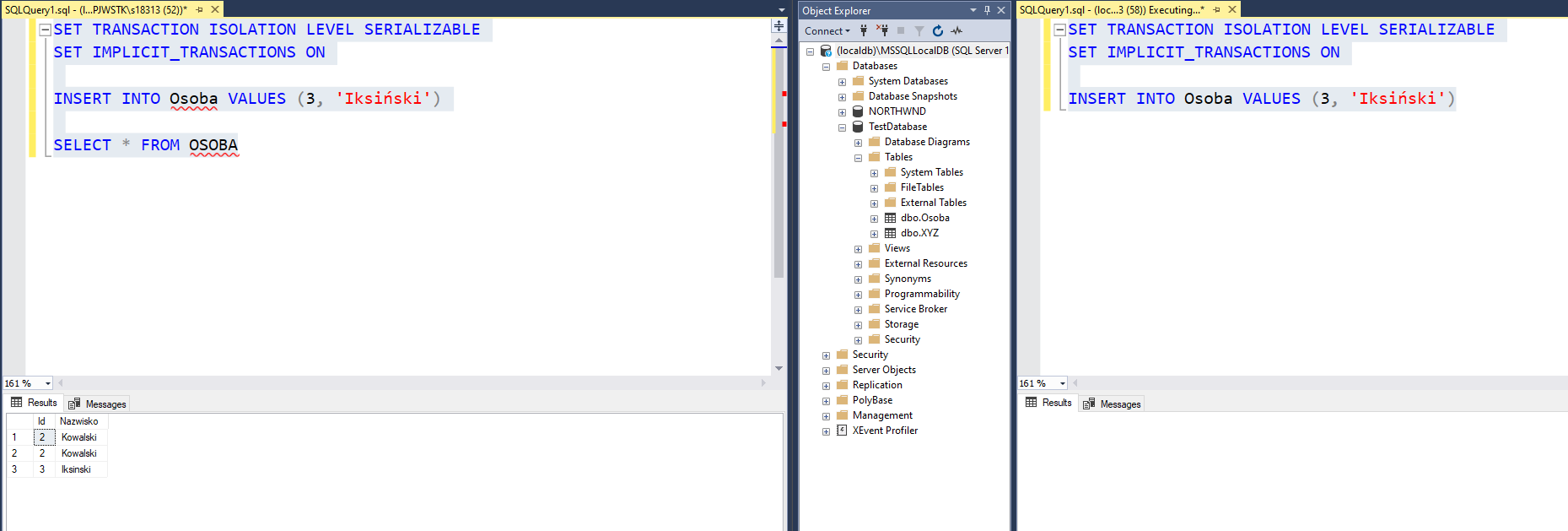
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE

Następnie wykonaj w obu oknach:

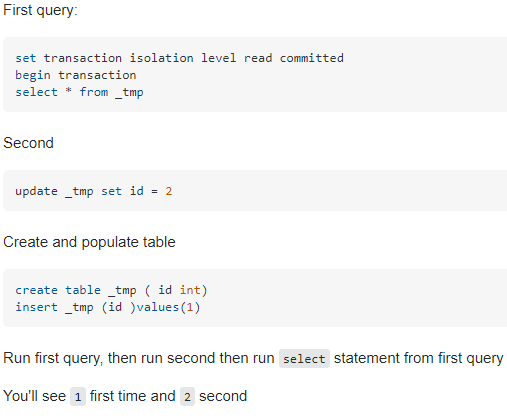
SELECT \* FROM Osoba

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).

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



set transaction isolation level read committed

begin transaction

select \* from \_tmp

update \_tmp set id = 2

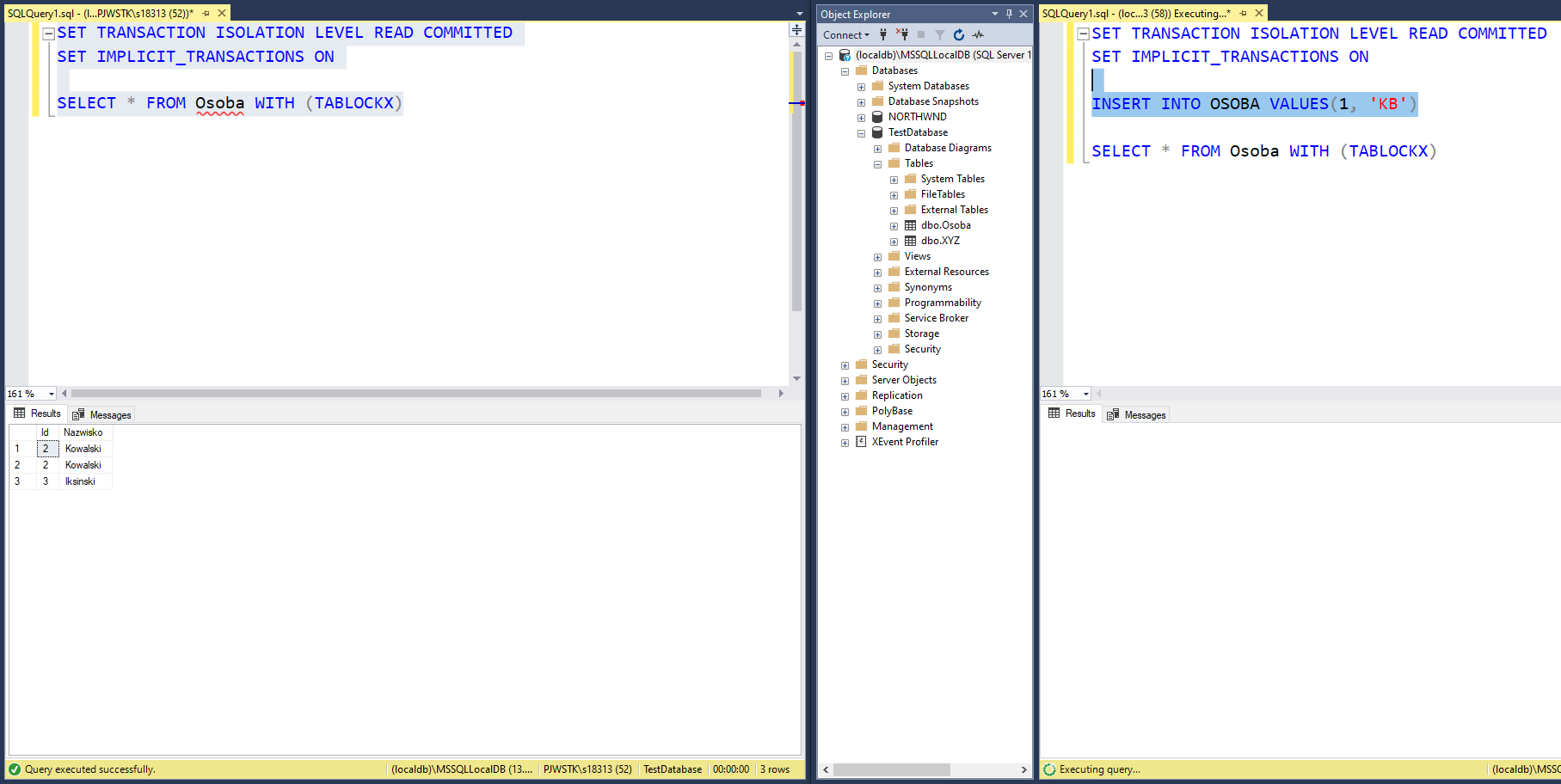
create table \_tmp ( id int)

insert \_tmp (id )values(1)

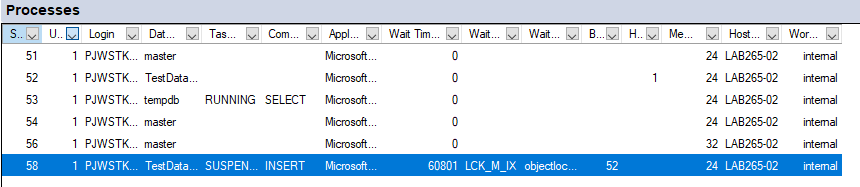
## Powróć do poziomu izolacji READ COMMITTED w obu oknach. Wykonaj instrukcję SELECT z użyciem locking hint wymuszającym założenie wyłącznej blokady na tabeli:

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.



## 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:

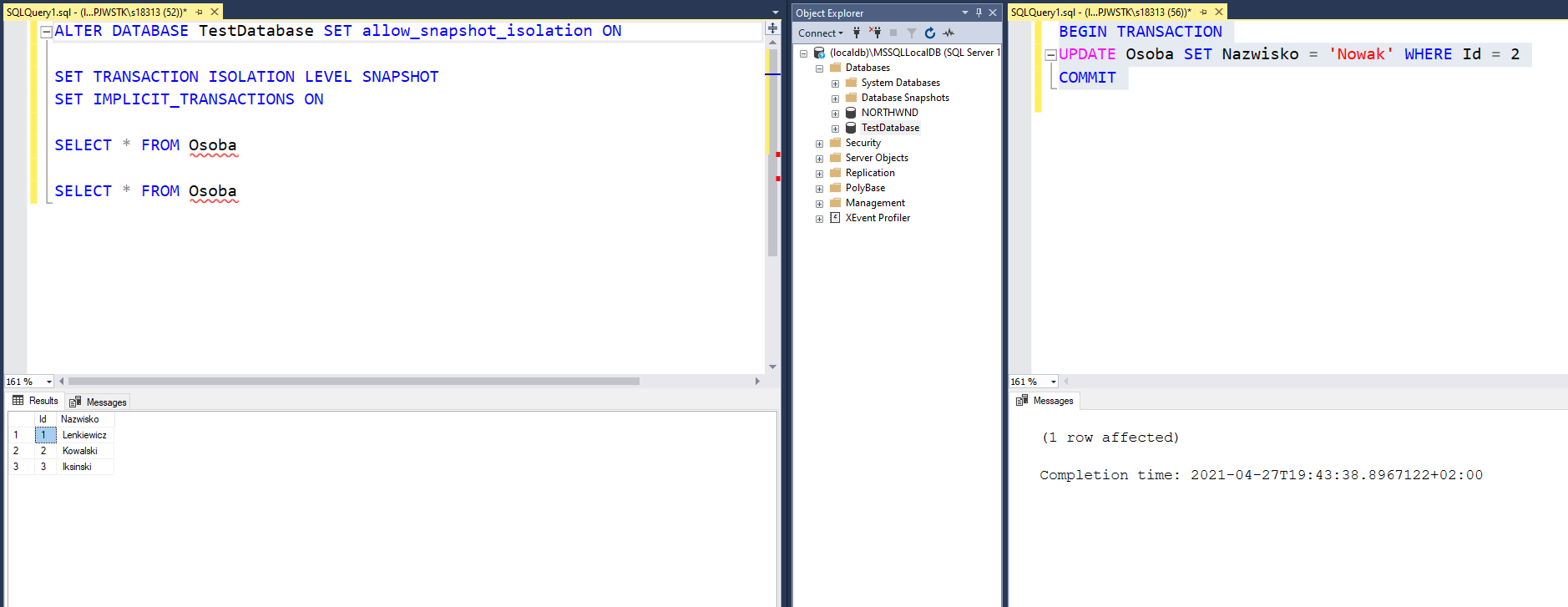
SELECT \* FROM Osoba

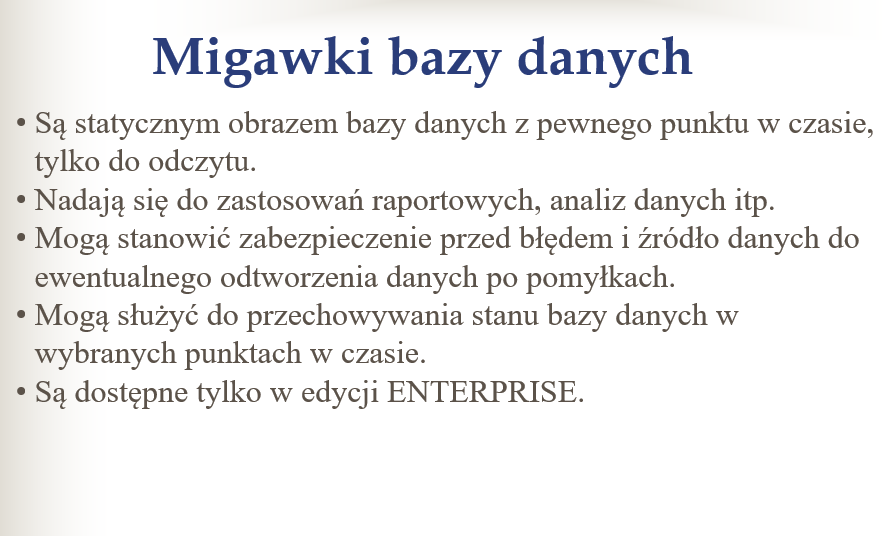
W oknie 2 wykonaj:

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

W oknie 1:

SELECT \* FROM Osoba

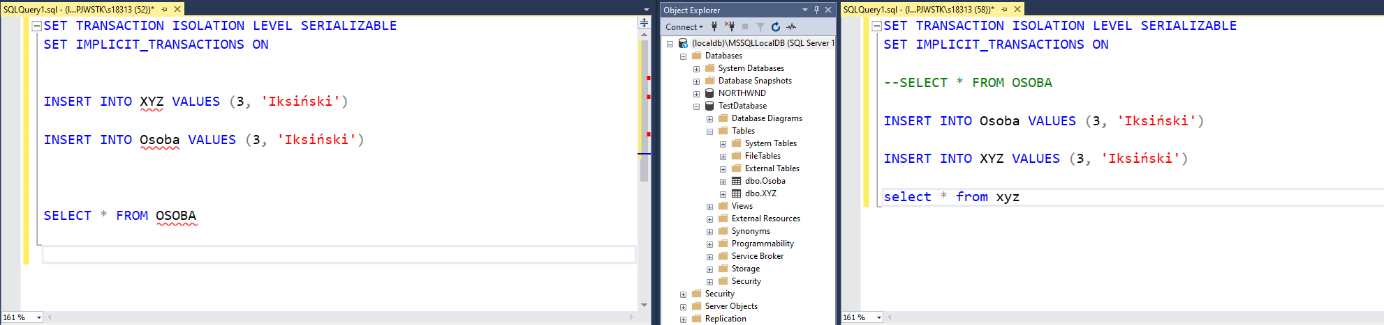


  
  
Zobaczysz starą, nieaktualną zawartość tabeli, gdyż transakcja pracuje na migawce. Spróbuj zaktualizować ten sam rekord:

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

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

## 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



# BLOK 4 - UPRAWNIENIA

## 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

1. 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”.

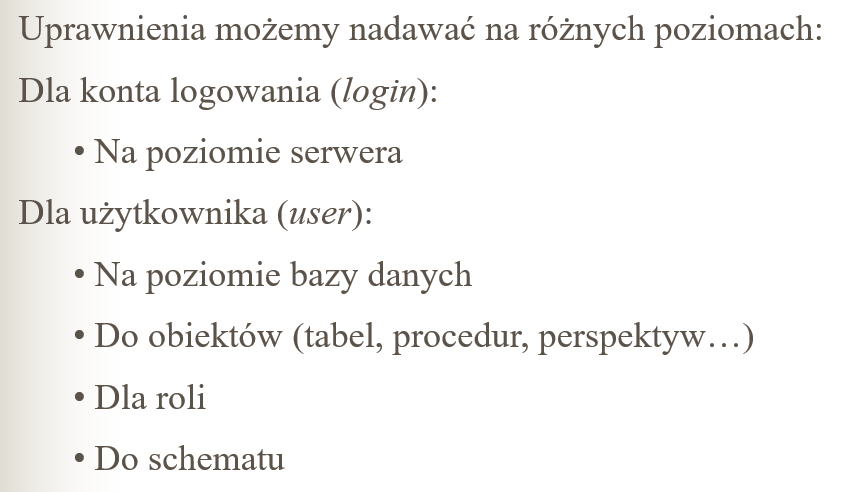
Najszybciej użytkownika założymy poprzez zarządzanie komputerem (na labach nie zrobimy, na wirtualkach zrobimy)

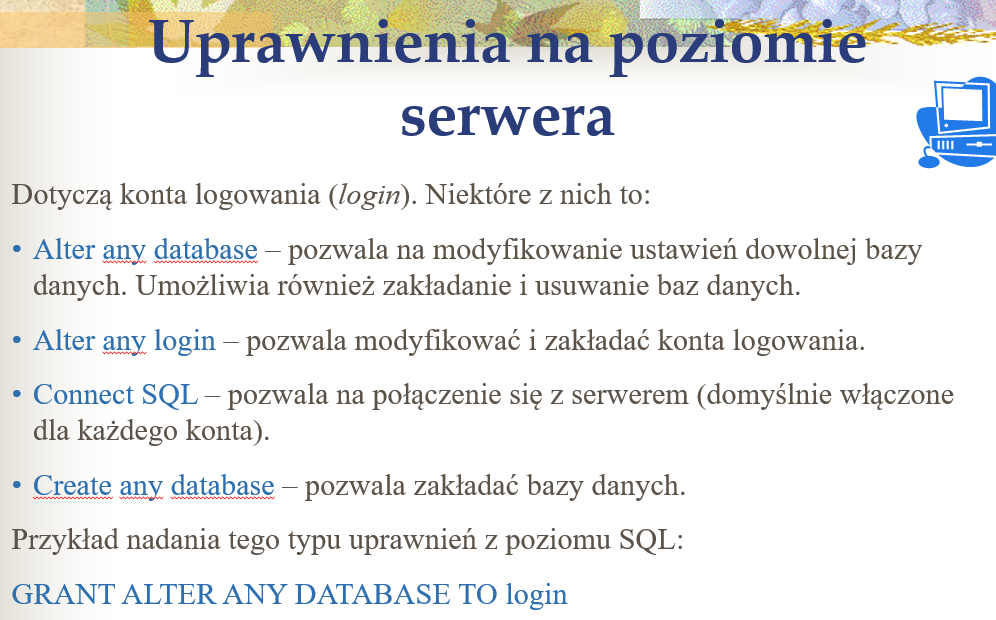
Security <całej instancji> -> logins -> ppm -> new login

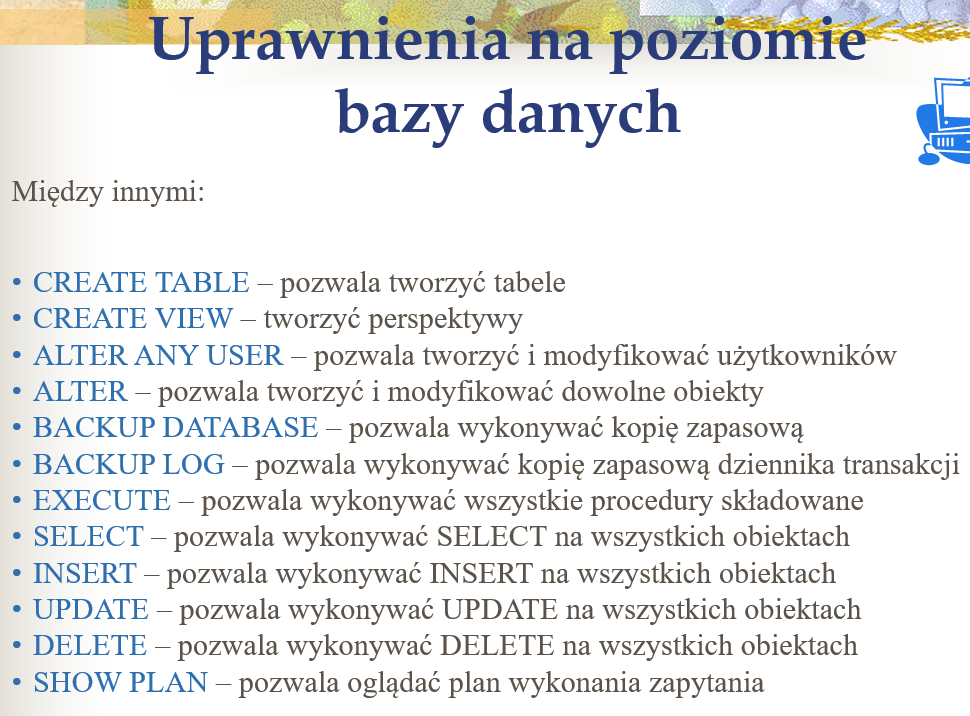
## Zaloguj się z powrotem na konto administratora. Przy pomocy Management Studio nadaj uprawnienia do wykonywania np. instrukcji SELECT użytkownikowi z punktu 1. Zaloguj się w drugim oknie Management Studio jako ten użytkownik. Sprawdź, czy możesz wykonywać instrukcje SQL.

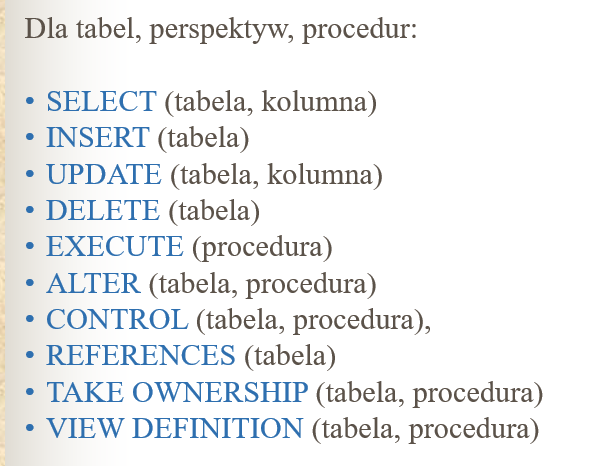
Aby dodać uprawnienia Userowi:

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

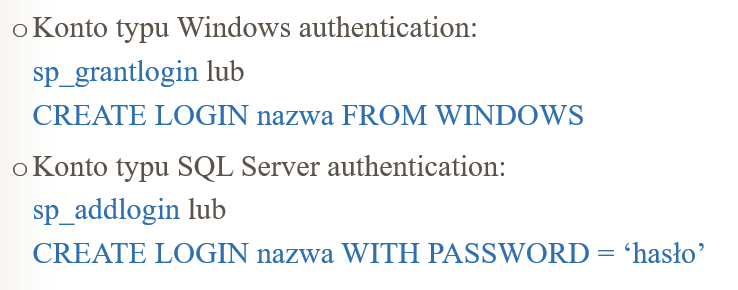


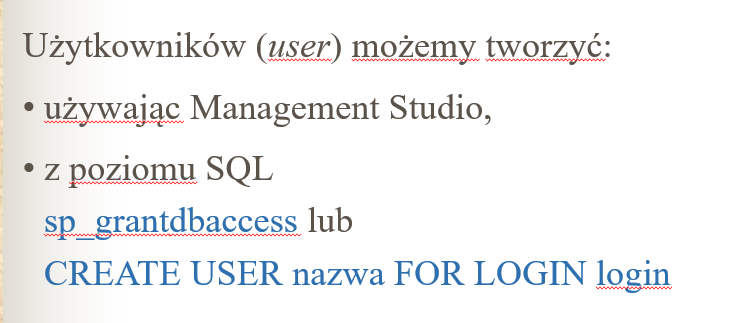






## Załóż jeszcze jedno konto logowania i użytkownika oraz nadaj dowolne uprawnienia, lecz tym razem nie używaj graficznego interfejsu, lecz wykonaj wszystko przy pomocy odpowiednich instrukcji SQL i/lub procedur sp\_.





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

Jak w punkcie 3 komendy

## 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

## 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

## Zabroń (DENY) wykonywania jakiejś operacji na poziomie roli, lecz pozostaw to prawo (GRANT) na poziomie użytkownika. Sprawdź, czy użytkownik będzie mógł wykonać tą 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

## 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

## 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.tabelka

## 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

## 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 chama nazwę)

Można odwoływać się bez podawania schematu

# BLOK 5 – BACKUPY

## Backup podstawowy

Prawy na bazę -> Tasks -> Back up

## Wczytanie podstawowe

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

## 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”

## 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.

## 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