

Paweł Rajba

[pawel@cs.uni.wroc.pl](mailto:pawel@cs.uni.wroc.pl)

<http://pawel.ii.uni.wroc.pl/>

# SQL Server

# Optymalizacja

# Agenda

- Wprowadzenie
- Optymalizacja struktury tabel
  - W tym postaci normalne
- Wykonywanie zapytań
- Statystyki
- Plan wykonania
- Indeksy
- Pomocne narzędzia

# Wprowadzenie

- Optymalizację można realizować na wielu poziomach
- Bardzo ważne jest poznanie wymagań, żeby wiedzieć co optymalizować.
  - Zwykle nie da zoptymalizować wszystkiego np. SELECT vs. INSERT

# Optymalizacja struktury tabel

- Wielkie tabele, z dużą liczbą kolumn nie sprzyjają wykonywaniu szybkich zapytań
- Uporządkowaniu danych sprzyja wprowadzenie postaci normalnych
- Wyróżniamy następujące postaci normalne:
  - 1NF, 2NF, 3NF, BCNF, 4NF, 5NFchoć w praktyce stosowane głównie postaci 1-3

# Pierwsza postać normalna (1NF)

- Wszystkie wartości w komórkach tabel są atomowe, czyli żadnych list, kolekcji, itp.
- Nie zawiera powtarzających się grup informacji
  - np. nie może być kolumn „Składnik1” „Składnik2”, „Składnik3”
- Kolejność wierszy może być dowolna
  - Znaczenie danych nie zależy od kolejności wierszy np. priorytet incydentów, czy kolejka zdarzeń do obsługi

# Pierwsza postać normalna (1NF)

- Przykład:

Przed:

Płeć	Imię
Męska	Jan, Piotr, Zenon
Żeńska	Anna, Maria, Zofia

Po:

Płeć	Imię
Męska	Jan
Męska	Piotr
Męska	Zenon
Żeńska	Anna
Żeńska	Maria
Żeńska	Zofia

# Pierwsza postać normalna (1NF)

- Mniej oczywisty przykład

	NumerZam	NazwaKlienta	AdresKlienta	DataZamowienia	SzczegolyZamowienia
1	101	Jan Kowalski	ul. Jana Pawła 12, 61-600 Poznań, woj. Wielko...	2012-01-02 00:...	Opony 205 R16 4szt, koszt 1200 PLN
2	102	Anna Dymna	ul. Staszica 1, 30-600 Kraków, Małopolska	2012-03-22 00:...	Alufelgi Silver 4 szt, koszt 2200 PLN
3	103	Piotr Wawrzyniak	al. Niepodległości 1, 30-600 Kraków, woj. Mało...	2012-03-22 00:...	Alufelgi Silver 4 szt, koszt 2200 PLN
4	104	Jan Kowalski	ul. Jana Pawła 12, 61-600 Poznań, woj. Wielko...	2012-10-22 00:...	Komplet żarówek, koszt 80 PLN
5	105	Jan Kowalski	ul. Poznańska 8, 21-120 Wrocław, Dolnośląskie	2012-05-22 00:...	Płyn do spryskiwacza 1szt, Trójkąt ostrzegawczy 1szt, koszt 15 PLN



	NrPozycji	NumerZam...	NazwaKlienta	Adres	KodPocztowy	Miasto	Wojewodztwo	DataZamowienia	ElementZamowienia	Ilosc	CenaJedn	WartZamNetto	Vat	WartZamBrutto
1	1	101	Jan Kowalski	ul. Jana Pawła 12	61-600	Poznań	Wielkopolskie	2012-01-02 00:...	Opony 205 R16	4	300,00	1200,00	23	1476,00
2	2	102	Anna Dymna	ul. Staszica 1	30-600	Kraków	Małopolskie	2012-03-22 00:...	Alufelgi Silver	4	550,00	2200,00	23	2706,00
3	3	103	Piotr Wawrzyniak	al. Niepodległości 1	30-600	Kraków	Małopolskie	2012-03-22 00:...	Alufelgi Silver	4	550,00	2200,00	23	2706,00
4	4	104	Jan Kowalski	ul. Jana Pawła 12	61-600	Poznań	Wielkopolskie	2012-10-22 00:...	Komplet żarówek	1	80,00	80,00	23	98,40
5	5	105	Jan Kowalski	ul. Poznańska 8	21-120	Wrocław	Dolnośląskie	2012-05-22 00:...	Płyn do spryskiwacza	1	10,00	15,00	23	18,45
6	6	105	Jan Kowalski	ul. Poznańska 8	21-120	Wrocław	Dolnośląskie	2012-05-22 00:...	Trójkąt ostrzegawczy	1	5,00	15,00	23	18,45

# Druga postać normalna (2NF)


- Mamy 1NF +
  - Żadna kolumna nie kluczowa nie jest częściowo funkcyjnie zależna od jakiegokolwiek klucza kandydującego (potencjalnego)
    - Lub inaczej: żadna informacja w wierszu nie może zależeć tylko od części klucza podstawowego (głównego)
  - Mniej formalnie:
    - W tabeli powinny być dane dotyczące tylko określonego rodzaju obiektu
  - Cel: eliminacja powtarzających się danych



# Druga postać normalna (2NF)

## ■ Przykład 1

Nielegalny związek



Imię	Nazwisko	Płeć	Stanowisko	Stawka za godzinę
Antoni	Anonim	Męska	Stolarz	10 zł
Natalia	Niewiadoma	Żeńska	Sekretarka	20 zł
Alina	Enigma	Żeńska	Sekretarka	20 zł



Klucz



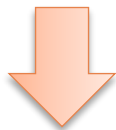
Imię	Nazwisko	Stanowisko	Stawka za godzinę
Antoni	Anonim	Stolarz	10 zł
Natalia	Niewiadoma	Sekretarka	20 zł
Alina	Enigma	Sekretarka	20 zł



Imię	Płeć
Antoni	Męska
Natalia	Żeńska
Alina	Żeńska

# Druga postać normalna (2NF)

## ■ Przykład 2



```
-- tabela w pierwszej postaci normalnej
select * from #Zamowienia_1NF
```

Klient							Detale zamówienia						
NrPozycji	NumerZam...	NazwaKlienta	Adres	KodPocztowy	Miasto	Wojewodztwo	DataZamowienia	ElementZamowienia	Ilosc	CenaJedn	WartZamNetto	Vat	WartZamBrutto
1	101	Jan Kowalski	ul. Jana Pawła 12	61-600	Poznań	Wielkopolskie	2012-01-02 00:...	Opory 205 R16	4	300,00	1200,00	23	1476,00
2	102	Anna Dymna	ul. Staszica 1	30-600	Kraków	Małopolskie	2012-03-22 00:...	Akufelgi Silver	4	550,00	2200,00	23	2706,00
3	103	Piotr Wawrzyniak	al. Niepodległości 1	30-600	Kraków	Małopolskie	2012-03-22 00:...	Akufelgi Silver	4	550,00	2200,00	23	2706,00
4	104	Jan Kowalski	ul. Jana Pawła 12	61-600	Poznań	Wielkopolskie	2012-10-22 00:...	Komplet żarówek	1	80,00	80,00	23	98,40
5	105	Jan Kowalski	ul. Poznańska 8	21-120	Wrocław	Dolnośląskie	2012-05-22 00:...	Płyn do spryskiwacza	1	10,00	15,00	23	18,45
6	105	Jan Kowalski	ul. Poznańska 8	21-120	Wrocław	Dolnośląskie	2012-05-22 00:...	Trójkąt ostrzegawczy	1	5,00	15,00	23	18,45

```
-- tabela w drugiej postaci normalnej
select * from #Zamowienia_2NF
```

	NumerZamowienia	IDKlient	DataZamowienia	WartZamNetto	Vat	WartZamBrutto
1	101	1	2012-01-02 00:00:00	1200,00	23	1476,00
2	102	2	2012-03-22 00:00:00	2200,00	23	2706,00
3	103	3	2012-03-22 00:00:00	2200,00	23	2706,00
4	104	1	2012-10-22 00:00:00	80,00	23	98,40
5	105	4	2012-05-22 00:00:00	15,00	23	18,45

```
-- nowa tabela przechowujące informacje o detalach zamówień
select * from #DetaleZamowien_2NF
```

	NumerZamowienia	KodProduktu	CenaJedn	Ilosc
1	101	1	300,00	4
2	102	2	550,00	4
3	103	2	550,00	4
4	104	3	80,00	1
5	105	4	10,00	1
6	105	5	5,00	1

```
-- nowa tabela przechowujące informacje o obiektach typu Klient
select * from #Klient_2NF
```

	IDKlient	NazwaKlienta	Adres	KodPocztowy	Miasto	Wojewodztwo
1	1	Jan Kowalski	ul. Jana Pawła 12	61-600	Poznań	Wielkopolskie
2	2	Anna Dymna	ul. Staszica 1	30-600	Kraków	Małopolskie
3	3	Piotr Wawrzyniak	al. Niepodległości 1	30-600	Kraków	Małopolskie
4	4	Jan Kowalski	ul. Poznańska 8	21-120	Wrocław	Dolnośląskie

```
-- nowa tabela przechowujące informacje o produktach
select * from #Produkty_2NF
```

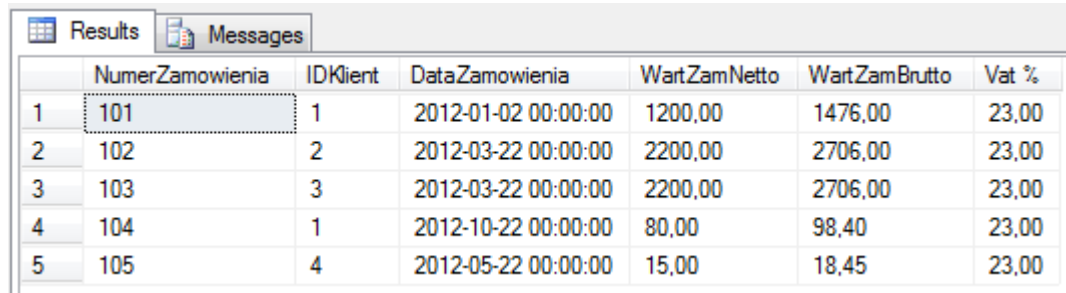
	KodProduktu	Nazwa	Producent	CenaJedn
1	1	Opory 205 R16	Pirelli	300,00
2	2	Akufelgi Silver	ENZO	550,00
3	3	Opory wymiana	NULL	80,00
4	4	Płyn do spryskiwacza	GreenApple	10,00
5	5	Trójkąt ostrzegawczy	GoSafer	5,00

# Trzecia postać normalna (3NF)

- Mamy 2NF
  - + każda kolumna nie będąca częścią klucza, zależy od niego bezpośrednio (a nie przechodnio)
    - Lub inaczej: żadna informacja w kolumnie, która nie jest kluczem podstawowym, nie może zależeć od niczego innego, jak tylko od klucza podstawowego.
  - Cel: Eliminowanie danych, które nie zależą od klucza

# Trzecia postać normalna (3NF)

- Przykład



	NumerZamowienia	IDKlient	DataZamowienia	WartZamNetto	WartZamBrutto	Vat %
1	101	1	2012-01-02 00:00:00	1200,00	1476,00	23,00
2	102	2	2012-03-22 00:00:00	2200,00	2706,00	23,00
3	103	3	2012-03-22 00:00:00	2200,00	2706,00	23,00
4	104	1	2012-10-22 00:00:00	80,00	98,40	23,00
5	105	4	2012-05-22 00:00:00	15,00	18,45	23,00

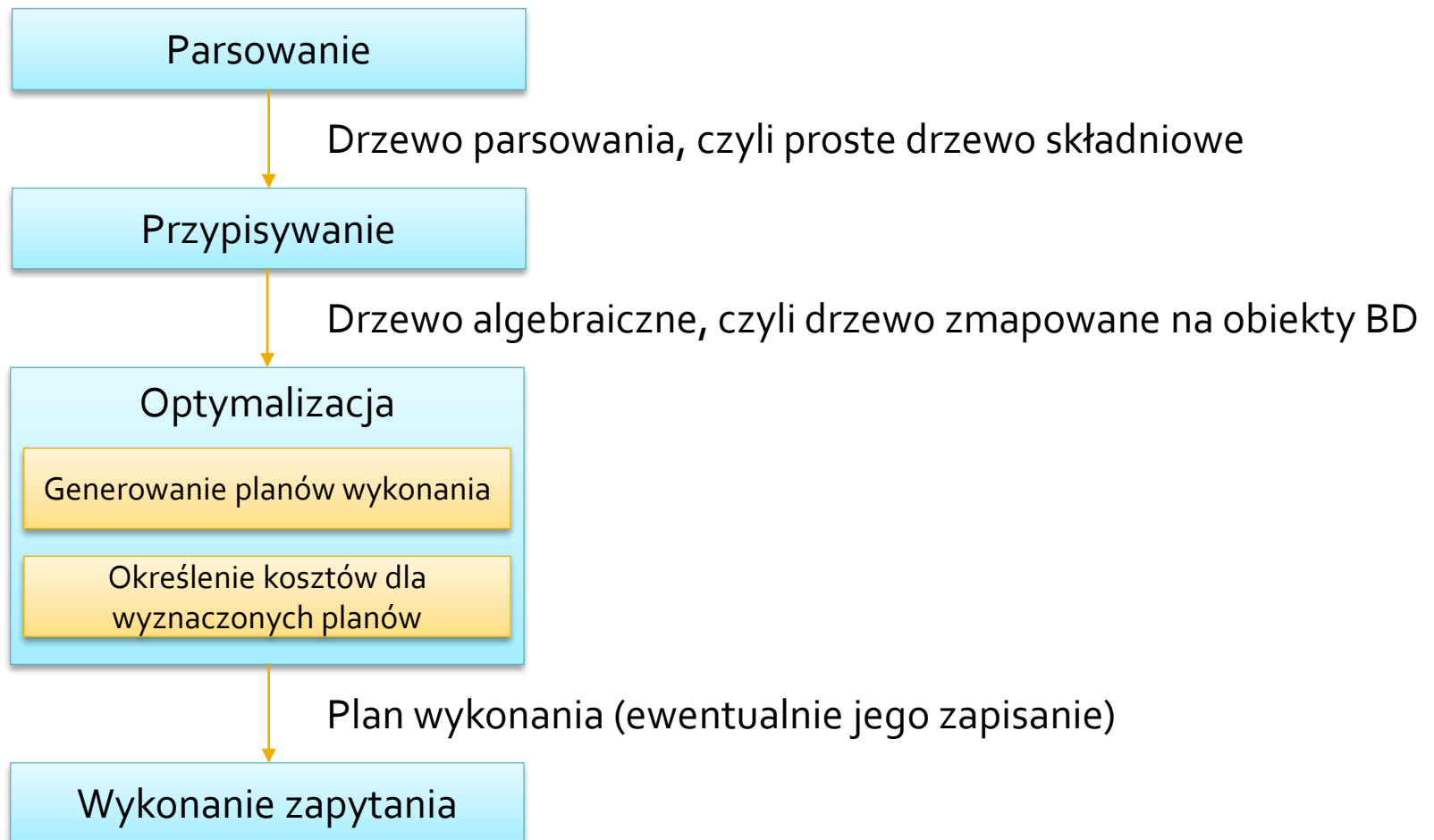
# Co daje dojście do 3NF?

- Usunięcie redundancji danych
- Łatwiej dane utrzymywać
- ... ale aby wyciągnąć komplet danych, trzeba łączyć tabele

# Kilka uwag na temat struktury tabel

- Jeżeli kolumna nie będzie używana, czy na pewno powinna być w tabeli?
- Jeśli mamy kolumny rzadko używane, może warto je wyciągnąć do osobnej tabeli?
- Czasami warto też trzymać przeliczone wartości, które są redundantne
- W pewnych sytuacjach również warto naruszyć 3NF, aby uniknąć zbyt wielu złączeń
- Tworząc strukturę trzeba jednak pamiętać, jakie operacje będą na tej strukturze wykonywane
  - Wrócimy do tego przy tworzeniu agregatów w DDD

# Wykonywanie zapytań



# Wykonywanie zapytań

- Istotnym i najtrudniejszym etapem jest optymalizacja wykonania zapytania
- Przy optymalizacji SQL Server rozpatruje m.in.
  - Statystyki tabel
  - Indeksy, które można wykorzystać
    - Czasami nawet opłaca się przejrzeć tabelę zamiast indeksu
  - Różne kolejności i sposoby złączeń tabel
- Rozpatrywany koszt to głównie
  - Operacje WE/WY
  - Czas procesora



# Statystyki

- Są to obiekty przechowujące informacje statystyczne tabel
  - Np. liczba wierszy, gęstości, rozkład wartości
- Do obejrzenia statystyk można
  - Użyć polecenia DBCC SHOW\_STATISTICS
    - Np. dbcc show\_statistics ( "SalesLT.SalesOrderHeader", PK\_SalesOrderHeader\_SalesOrderID)
  - Mgmt Studio → Tabela → Statistics
- Ważnym elementem związanym z optymalizacją jest częstość odświeżania statystyk
  - Domyślnie odświeżanie jest automatyczne i zarządzane przez SQL Server
  - Można też zarządzać statystykami „ręcznie”
- Więcej:
  - <https://msdn.microsoft.com/en-us/library/ms190397.aspx>

# DEMO

- Odpalamy
  - dbcc show\_statistics  
("SalesLT.SalesOrderHeader",  
PK\_SalesOrderHeader\_SalesOrderID)
- Oglądamy statystyki w Mgmt Studio

# Plan wykonania

- Plany wykonania dzielimy na:
  - Szacowane (estimated)
  - Faktyczne (actual)
- Plany można zobaczyć w formie
  - Graficznej
  - Tekstowej
  - XML

# Plan wykonania

- Plany wykonania są przechowywane w buforze
- Przed wykonaniem zapytania:
  - Jeśli plan już jest, zostaje wykorzystany
  - Jeśli go nie ma, zostaje utworzony i zachowany
- SQL Server zarządza tym buforem
- Kilka wybranych przyczyn rekompilacji planu
  - zmiana struktury tabeli lub widoku (ALTER)
  - odświeżenie statystyk (ręcznie lub automatycznie)
  - usunięcie indeksu wykorzystywanego w planie
  - wywołanie procedury sp\_recompile
  - duże zmiany w zawartości tabeli (INSERT, DELETE)

# Plan wykonania

- W planie wykonania mamy
  - Elementy języka (zielone)
  - Operatory fizyczne/logiczne (niebieskie)
  - Operacje związane z kursorem (żółte)
- Plan wykonania jest w postaci drzewa
  - Drzewa, które powstaje po sparsowaniu zapytania
- Po najechaniu na węzeł dostajemy sporo dodatkowych szczegółów
- Może się zdarzyć, że SQL Server zasugeruje w planie faktycznym utworzenie indeksu

# Plan wykonania

- Pobranie informacji o planie wykonania

Szacowany	Faktyczny
SET SHOWPLANTEXT {ON OFF}	SET STATISTICS PROFILE {ON OFF}
SET SHOWPLAN_XML {ON OFF}	SET STATISTICS XML {ON OFF}
Display Estimated Execution Plan (Management Studio)	Include Actual Execution Plan (Management Studio)

- Dodatkowo możemy uzyskać statystyki
  - Czasu: SET STATISTICS TIME {ON|OFF}
  - Operacji WE/WY: SET STATISTICS IO {ON|OFF}
- Bardzo ciekawą opcją są Live Query Statistics
  - Więcej:
    - <https://msdn.microsoft.com/en-us/library/dn831878.aspx>
    - <https://blogs.technet.microsoft.com/cansql/2017/02/28/live-query-statistics-in-sql-server-2016/>

# DEMO

- Oglądamy plany dla zapytania

DBCC FREEPROCCACHE -- wyczyszczenie bufora planów wykonania

```
USE Northwind;
```

```
GO
```

```
SELECT DISTINCT(ShipCity) FROM Orders;
```

```
GO
```

```
DECLARE c CURSOR FOR SELECT TOP 2 ShipCity FROM Orders;
```

```
OPEN c;
```

```
FETCH NEXT FROM c;
```

```
WHILE @@FETCH_STATUS = 0
```

```
BEGIN
```

```
    FETCH NEXT FROM c;
```

```
END
```

```
CLOSE c;
```

```
DEALLOCATE c;
```

# DEMO

- Oglądamy plany dla zapytań

```
DBCC FREEPROCCACHE
```

```
SELECT DISTINCT(a.City) FROM SalesLT.SalesOrderHeader o JOIN  
SalesLT.Address a ON o.ShipToAddressID = a.AddressID WHERE  
o.SubTotal>30000  
GO
```

```
DECLARE c CURSOR FOR SELECT TOP 3 a.City FROM  
SalesLT.SalesOrderHeader o JOIN SalesLT.Address a ON  
o.ShipToAddressID = a.AddressID;  
OPEN c;  
FETCH NEXT FROM c;  
WHILE @@FETCH_STATUS = 0  
BEGIN  
    FETCH NEXT FROM c;  
END  
CLOSE c;  
DEALLOCATE c;
```

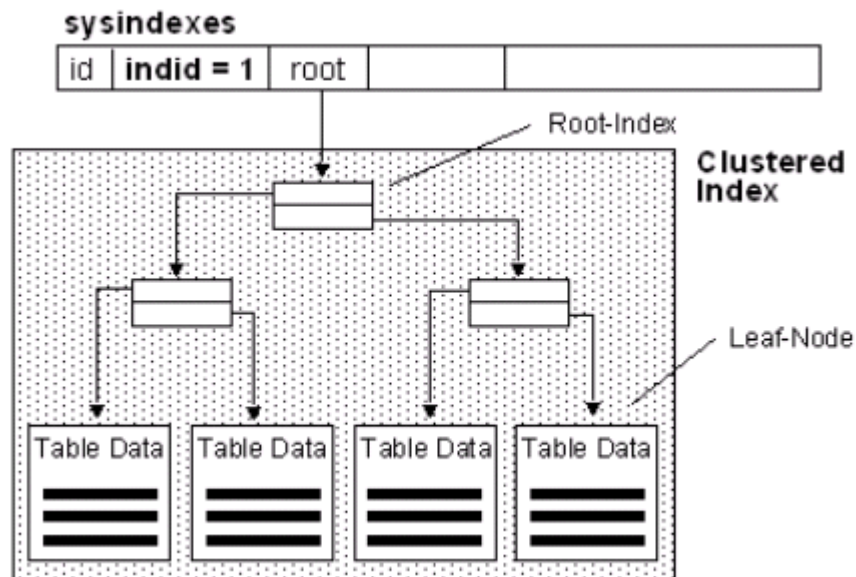


# Indeksy

- Indeks to struktura danych mająca na celu przyspieszenie pobierania danych
  - Indeksy są przechowywane w strukturze B-drzewa
- Jest związana z tabelą lub widokiem oraz jej/jego wybranymi kolumnami
  - Indeksować można prawie wszystko oprócz głównie LOB (np. images, text, varchar(max))
- Dobór odpowiednich indeksów jest balansem pomiędzy
  - Szybkością pobierania danych
  - Kosztem związanym z jego utrzymaniem (CRUD)

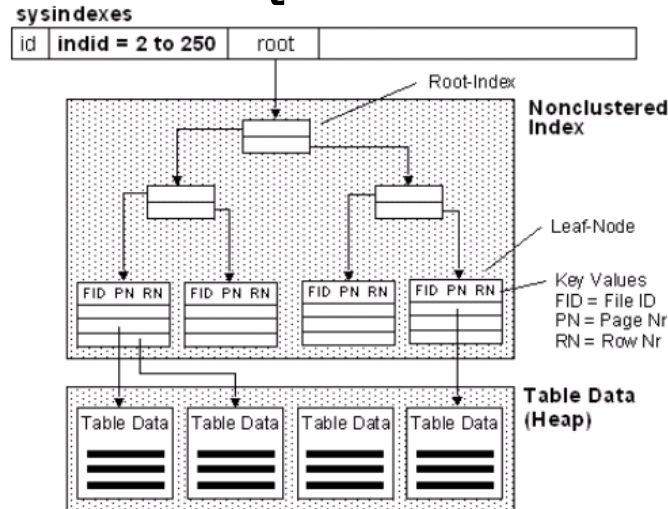
# Indeksy

- Indeks zgrupowany (clustered index)
  - Może być tylko jeden w tabeli
  - Wyznacza on porządek danych w samej tabeli



# Indeksy

- Indeks niezgrupowany (nonclustered index)
  - Może być ich wiele w tabeli
  - Nie zawiera danych tylko referencje do nich
  - Na poziomie liści można dołączyć tzw. included columns, które nie są indeksowane

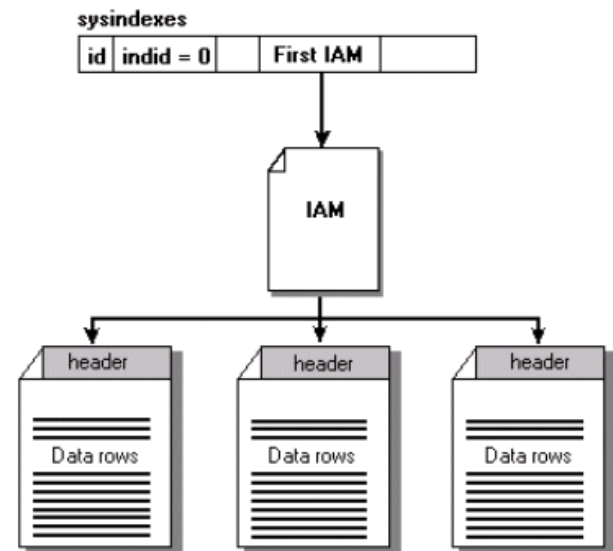


# Indeksy

- Kolejne rodzaje indeksów
  - Złożony (composite)
    - kiedy zawiera więcej niż jedną kolumnę
  - Unikalny (unique)
    - wszystkie wartości muszą być różne)
  - Kryjący (covering)
    - Kiedy zawiera wszystkie kolumny dla wybranego zapytania
  - Filtrowany (filtered)
    - Tylko dla non-clustered
    - Indeksowane są tylko wybrane dane

# Indeksy

- Zwykle każda tabela zawiera jakiś indeks
  - Przykładowo: przy tworzeniu primary key lub definiując ograniczenie „unique” tworzony jest indeks
- Jeśli jednak tabela nie ma indeksu, wtedy ma strukturę sterty (heap)



# Indeksy

- Tworzenie indeksu

```
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 ] ) ]
```

- Usuwanie indeksu

```
DROP INDEX index_name
```

# Indeksy

- Indeksy niezgrupowane można utworzyć tak, żeby przy zapytaniu nie było potrzeby odwoływać się do stron z danymi
- Zasady dotyczące takich indeksów
  - indeks musi zawierać wszystkie kolumny wyniku
    - Można zastosować `included columns`
  - zastosowanie indeksowanych widoków może prowadzić do indeksowania częściowych sum itp., co z kolei może dać duży wzrost wydajności

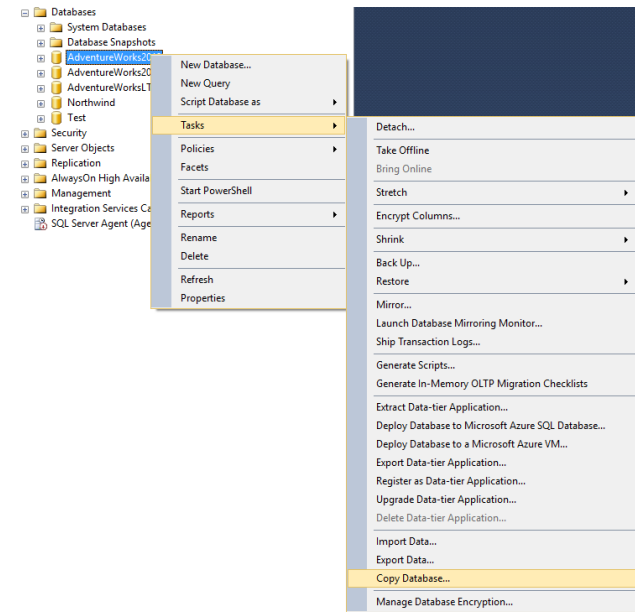
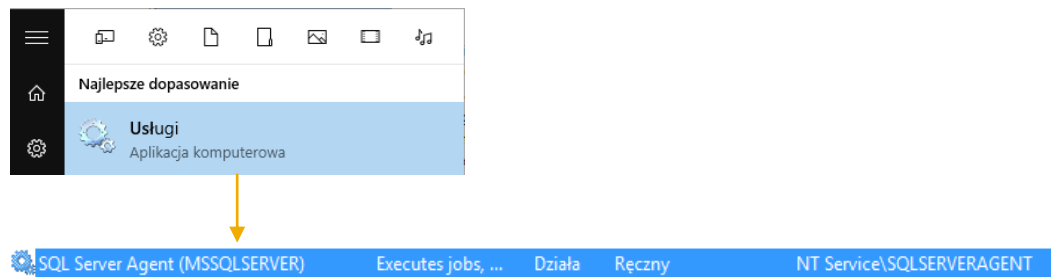
# Jak tworzyć indeksy?

- Jest cała masa artykułów i opracowań
- Kilka wybranych wskazówek
  - Poziom modyfikacji
    - Dużo → indeksy trzeba tworzyć ostrożnie
    - Mało → można utworzyć więcej indeksów do usprawnienia pobierania danych
  - Dla indeksu zgrupowanego kolumna powinna być mała i najlepiej liczbowa
    - Dlatego też zwykle PK to indeks zgrupowany, chociaż nie zawsze będzie to optymalne
  - Unikalność wartości indeksu wpływa na wydajność
  - Dla indeksów złożonych kolejność kolumn ma znaczenie
    - Częściej używane (np. te w WHERE powinny być wcześniej)
    - Im bardziej unikalne wartości, tym wcześniej na liście
  - Indeksować: wyszukiwane kolumny, biorące udział w złączeniach



# Zrobienie kopii bazy danych

- Bardzo pomocne do testów z indeksami
- Główne etapy
  - Uruchamiamy usługę SQL Server Agent



- Odpalamy kreatora kopiowania bazy danych
  - I klikamy dalej

# DEMO

```
DBCC FREEPROCCACHE
```

```
-- 1 --
```

```
USE AdventureWorks2012_new;  
CREATE NONCLUSTERED INDEX IX_PurchaseOrderDetail_RejectedQty ON AdventureWorks2012_new.Purchasing.PurchaseOrderDetail(RejectedQty  
DESC, ProductID ASC, DueDate, OrderQty);
```

```
USE AdventureWorks2012;
```

```
GO  
SELECT RejectedQty, ((RejectedQty/OrderQty)*100) AS RejectionRate, ProductID, DueDate FROM Purchasing.PurchaseOrderDetail ORDER BY  
RejectedQty DESC, ProductID ASC;
```

```
GO  
USE AdventureWorks2012_new;  
GO  
SELECT RejectedQty, ((RejectedQty/OrderQty)*100) AS RejectionRate, ProductID, DueDate FROM Purchasing.PurchaseOrderDetail ORDER BY  
RejectedQty DESC, ProductID ASC;
```

```
GO
```

```
-- 2 --
```

```
USE AdventureWorks2012;  
GO  
SELECT SalesOrderDetailID, UnitPrice, UnitPriceDiscount FROM Sales.SalesOrderDetail s WHERE UnitPrice>1000
```

```
GO  
USE AdventureWorks2012_new;  
SELECT SalesOrderDetailID, UnitPrice, UnitPriceDiscount FROM Sales.SalesOrderDetail s WHERE UnitPrice>1000
```

```
GO
```

```
USE AdventureWorks2012_new;  
DROP INDEX IF EXISTS Sales.SalesOrderDetail.IX_SalesOrderDetails_UnitPrice;  
CREATE INDEX IX_SalesOrderDetails_UnitPrice ON AdventureWorks2012_new.Sales.SalesOrderDetail (UnitPrice) INCLUDE (UnitPriceDiscount)  
WHERE UnitPrice>1000
```

# Pomocne narzędzia

## SQL Server Profiler

EventClass	TextData	ApplicationName	NTUserName	LoginName	CPU	Reads	Writes	Duration	ClientIP
SQL:BatchCompleted	DECLARE @edition sysname; SET @edit...	Microsoft SQ...	Paweł	MUFASA...	0	0	0	0	0
SQL:BatchStarting	DECLARE @edition sysname; SET @edit...	Microsoft SQ...	Paweł	MUFASA...					
SQL:BatchCompleted	DECLARE @edition sysname; SET @edit...	Microsoft SQ...	Paweł	MUFASA...	0	0	0	0	0
SQL:BatchStarting	USE Adventureworks2012;	Microsoft SQ...	Paweł	MUFASA...					
SQL:BatchCompleted	USE Adventureworks2012;	Microsoft SQ...	Paweł	MUFASA...	0	0	0	0	0
SQL:BatchStarting	SELECT SalesOrderDetailID, UnitPric...	Microsoft SQ...	Paweł	MUFASA...					
SQL:BatchCompleted	SELECT SalesOrderDetailID, UnitPric...	Microsoft SQ...	Paweł	MUFASA...	31	1462	0	121	
SQL:BatchStarting	USE Adventureworks2012_new; SELECT...	Microsoft SQ...	Paweł	MUFASA...					
SQL:BatchCompleted	USE Adventureworks2012_new; SELECT...	Microsoft SQ...	Paweł	MUFASA...	0	462	0	153	
SQL:BatchStarting	DECLARE @edition sysname; SET @edit...	Microsoft SQ...	Paweł	MUFASA...					
SQL:BatchCompleted	DECLARE @edition sysname; SET @edit...	Microsoft SQ...	Paweł	MUFASA...	0	0	0	0	
RPC:Completed	exec msdb..sp_DTA_help_session @Ses...	Database Eng...	Paweł	MUFASA...	0	911	0	17	
SQL:BatchStarting	DECLARE @edition sysname; SET @edit...	Microsoft SQ...	Paweł	MUFASA...					
SQL:BatchCompleted	DECLARE @edition sysname; SET @edit...	Microsoft SQ...	Paweł	MUFASA...	0	0	0	0	
SQL:BatchStarting	DECLARE @edition sysname; SET @edit...	Microsoft SQ...	Paweł	MUFASA...					
SQL:BatchCompleted	DECLARE @edition sysname; SET @edit...	Microsoft SQ...	Paweł	MUFASA...	0	0	0	0	
RPC:Completed	exec msdb..sp_DTA_help_session @Ses...	Database Eng...	Paweł	MUFASA...	16	672	0	9	
RPC:Completed	exec msdb..sp_DTA_help_session @Ses...	Database Eng...	Paweł	MUFASA...	0	10	0	0	
RPC:Completed	exec msdb..sp_DTA_help_session @Ses...	Database Eng...	Paweł	MUFASA...	0	12	0	0	
Audit Login	-- network protocol: LPC set quote...	Database Eng...	Paweł	MUFASA...					
RPC:Completed	exec msdb..sp_DTA_help_session @Ses...	Database Eng...	Paweł	MUFASA...	0	10	0	0	

Trace is running. Ln 50, Col 1 | Rows: 50 | Connections: 1

## Database Engine Tuning Advisor

Database Engine Tuning Advisor

Session Monitor: MUFASA - Paweł 24.04.2016 21:39:14

General | Tuning Options | Progress

**Tuning Progress**

4 Remaining | 5 Total | 0 Error | 1 Success | 0 Warning

Action	Status	Message
Submitting Configuration Information	Success	
Consuming Workload	0% complete	
Performing Analysis	Estimated improvement...	
Generating Reports		
Generating Recommendations		

**Tuning Log**

CategoryID	Event	Statement	Frequency	Reason
------------	-------	-----------	-----------	--------

Analyzing workload. | Connections: 3

# Literatura

- Postacie normalne
  - <https://pg.edu.pl/documents/1403427/1bacfc1e-012d-4800-b6b2-4foc4cce3c99>
  - <https://devszczepaniak.pl/postaci-normalne/>
  - <https://mst.mimuw.edu.pl/lecture.php?lecture=bad&part=Ch6>
  - <https://home.agh.edu.pl/~horzyk/lectures/db/BazyDanychAccess-Normalizacja.pdf>
  - <https://edu.pjwstk.edu.pl/wyklady/rbd/scb/rW5.htm>
- Podstawy optymalizacji
  - <http://pdf.ebookpoint.pl/sql140/sql140.pdf>
  - <https://msdn.microsoft.com/pl-pl/library/encyklopedia-sql--t-sql-optymalizacja-zapytan.aspx>
  - <http://wss.geekclub.pl/baza-wiedzy/kurs-transact-sql-czesc-4-optymalizacja-zapytan,773>
  - <https://www.mssqltips.com/sqlservertutorial/2250/graphical-query-plan-tutorial/>
  - [http://sqlmag.com/site-files/sqlmag.com/files/archive/sqlmag.com/content/content/144603/top\\_10\\_tips\\_for\\_optimizing-final.pdf.pdf](http://sqlmag.com/site-files/sqlmag.com/files/archive/sqlmag.com/content/content/144603/top_10_tips_for_optimizing-final.pdf.pdf)

# Literatura

- Statystyki
  - <https://msdn.microsoft.com/en-us/library/ms190397.aspx>
  - <https://msdn.microsoft.com/en-us/library/ms174384.aspx>
  - [https://msdn.microsoft.com/pl-pl/library/ms187348\(v=sql.110\).aspx](https://msdn.microsoft.com/pl-pl/library/ms187348(v=sql.110).aspx)
  - <https://msdn.microsoft.com/en-us/library/dn831878.aspx>
- Plany wykonania
  - [https://technet.microsoft.com/en-us/library/ms181055\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms181055(v=sql.105).aspx)
  - <https://www.simple-talk.com/sql/performance/execution-plan-basics/>
  - <https://www.simple-talk.com/sql/performance/graphical-execution-plans-for-simple-sql-queries/>
  - <https://www.simple-talk.com/sql/performance/understanding-more-complex-query-plans/>
- Indeksy
  - [https://technet.microsoft.com/en-us/library/aa964133\(v=sql.90\).aspx](https://technet.microsoft.com/en-us/library/aa964133(v=sql.90).aspx)
  - [https://technet.microsoft.com/pl-pl/library/jj835095\(v=sql.110\).aspx](https://technet.microsoft.com/pl-pl/library/jj835095(v=sql.110).aspx)
  - <https://msdn.microsoft.com/en-us/library/ms175049.aspx>
  - <https://www.simple-talk.com/sql/learn-sql-server/sql-server-index-basics/>
  - [https://msdn.microsoft.com/pl-pl/library/ms190806\(v=sql.110\).aspx](https://msdn.microsoft.com/pl-pl/library/ms190806(v=sql.110).aspx)
  - [https://msdn.microsoft.com/pl-pl/library/cc280372\(v=sql.110\).aspx](https://msdn.microsoft.com/pl-pl/library/cc280372(v=sql.110).aspx)
  - <https://www.simple-talk.com/sql/performance/introduction-to-sql-server-filtered-indexes/>
  - <http://www.sqlskills.com/blogs/kimberly/indexes-just-because-you-can-doesnt-mean-you-should/>