

# Biuletyn techniczny

## Strojenie wydajnościowe baz MS SQL dla Comarch ERP Optima

Data ostatniej aktualizacji: 30.07.2015



Comarch ERP  
Optima

**Wersja**  
2016.5.1

## Spis treści

<b>1</b>	<b>Wprowadzenie .....</b>	<b>3</b>
<b>2</b>	<b>Planowanie instalacji.....</b>	<b>4</b>
2.1	Dobór konfiguracji sprzętowej dla serwera bazy danych.....	4
2.1.1	Konfiguracja sprzętowa .....	4
2.1.2	Biura rachunkowe: .....	5
2.2	Edycje serwera SQL .....	5
2.2.1	Architektura 32 bit, a 64 bit. ....	6
<b>3</b>	<b>Konfiguracja serwera SQL .....</b>	<b>7</b>
3.1	Minimalna i Maksymalna ilość wykorzystywanej pamięci RAM.....	7
3.1.1	Minimum Server Memory .....	7
3.1.2	Maximum Server Memory.....	7
3.2	Inne parametry .....	9
3.2.1	Optimize for Ad hoc Workloads .....	9
<b>4</b>	<b>Diagnoza problemów wydajnościowych.....</b>	<b>11</b>
<b>5</b>	<b>Optymalizacja dużych baz danych .....</b>	<b>12</b>
5.1	Wyszukiwanie brakujących indeksów .....	12
5.2	Database Engine Tuning Advisor.....	14
5.2.1	Zapis ruchu za pomocą Microsoft SQL Server Profiler .....	14
5.2.2	Przygotowanie optymalizacji w Database Engine Tuning Advisor .....	14
5.2.3	Zapis rekomendacji.....	16
<b>6</b>	<b>Cykliczne czynności administracyjne .....</b>	<b>17</b>
6.1	Maintenance Plan – zautomatyzowane strojenie dla pełnych wersji MS SQL .....	17
6.1.1	Krok 1 – Uruchomienie kreatora Maintenance Plan'u .....	17
6.1.2	Krok 2 – Wskazanie nazwy planu oraz terminu jego wykonywania .....	18
6.1.3	Krok 3 – wskazanie składników planu serwisowego.....	19
6.1.4	Krok 4 – wskazanie kolejności wykonywania składników planu serwisowego .....	20
6.1.5	Krok 5 – wskazanie baz danych dla każdego elementu planu serwisowego .....	21
6.1.6	Krok 6 – charakterystyczne opcje dla każdego elementu planu serwisowego.....	22
6.1.7	Krok 7 – wskazanie sposobu raportowania wyników wykonania planu serwisowego oraz zakończenie pracy z kreatorem. ....	23
6.2	Plan serwisowy dla bezpłatnych wersji MS SQL .....	24

# 1 Wprowadzenie

Serwer bazy danych jest centralnym miejscem dla całej instalacji **Comarch ERP Optima** i ma bardzo duże znaczenie dla wydajnej jej pracy. Dlatego bardzo ważne jest odpowiednie skonfigurowanie serwera zarówno pod względem sprzętowym, doboru właściwego oprogramowania oraz jego ustawień. Dodatkowo z biegiem czasu i działalnością klienta wpływają nowe dane słownikowe oraz dokumenty, co powoduje przyrost bazy danych. Ważne jest więc monitorowanie stanu serwera SQL i odpowiednie reagowanie w celu zapewnienia jak najlepszej wydajności.

Niniejszy biuletyn zawiera wytyczne dla konfiguracji serwera bazy danych dedykowanego do pracy z **Comarch ERP Optima**. Przedstawiono w nim również podstawowe metody diagnozy problemów wydajnościowych oraz sposoby optymalizacji. W ostatnim rozdziale znajduje się obszerny opis cyklicznych czynności administracyjnych, których celem jest utrzymanie serwera SQL w dobrej kondycji. Do biuletynu dołączone są skrypty, które pozwalają zautomatyzować cykliczne czynności administracyjne dla wersji Express serwera SQL.

## 2 Planowanie instalacji

### 2.1 Dobór konfiguracji sprzętowej dla serwera bazy danych

Serwer bazy danych jest wrażliwym elementem, którego awaria może spowodować przestój całej firmy oraz duże straty finansowe, dlatego warto zainwestować w markowy, sprawdzony sprzęt. Taki serwer w atrakcyjnych cenach można np. nabyć w ramach [Zintegrowanej Oferty Comarch](#).

#### 2.1.1 Konfiguracja sprzętowa

Przy doborze serwera bazy danych należy zwrócić szczególną uwagę to, aby zapewniał on niezawodną pracę i bezpieczeństwo przechowywanych danych. Dlatego warto, aby jak najwięcej komponentów było nadmiarowych usuwając pojedyncze ogniwa awarii. Czyli dobrze jest zastosować podwójne zasilacze oraz zasilanie awaryjne (UPS) pozwalające na prawidłowe zamknięcie systemu w momencie braku prądu.

Jeżeli chodzi o wydajność to przede wszystkim należy zwrócić na ilość pamięci RAM oraz szybkość podsystemu dyskowego. Procesor zwykle odgrywa mniejsze znaczenie. Jednakże powinien zawierać rdzenie nowej generacji (Intel Nehalem lub lepszy) i taktowaniu przynajmniej 2 GHz.

Im więcej dostępnej pamięci tym lepiej, oczywiście ze względu na koszty, jej ilość musi być dobrana do zapotrzebowania, bazując na wielkości instalacji, czyli ilości użytkowników, wielkości baz danych oraz ich ilości. Orientacyjne ilości pamięci RAM dla różnych konfiguracji zostały podane w Podręczniku konfiguracji dla bieżącej wersji [Comarch ERP Optima](#). Później oczywiście należy obserwować liczniki serwera i sprawdzać, czy nie ma potrzeby jej zwiększenia. Informacje na temat przykładowych liczników, na które warto zwrócić uwagę podano w punkcie 4. Diagnoza problemów wydajnościowych.

Podsystem dyskowy powinien być zabezpieczony przed awariami poprzez odpowiednią konfigurację. Zalecana konfiguracja to przynajmniej Raid 1 (mirror) dla wolumenu obsługującego system oraz w miarę możliwości Raid 10 dla wolumenu przechowującego dane. Dodatkowo należy zwrócić uwagę, aby kontroler dyskowy posiadał podtrzymywanie baterijne, które umożliwia zastosowanie opcji write-back przyspieszającego działanie podsystemu dyskowego.

Najczęściej przyjmowaną jednostką wydajności podsystemu dyskowego jest IOPS<sup>1</sup> (ang. Input/Output Operations Per Second, IOPS), czyli ilość operacji wejścia wyjścia na sekundę. Jako operację wejścia/ wyjścia rozumie się odczyt lub zapis fragmentu danych, najczęściej o rozmiarze 4 kB.

---

<sup>1</sup> Wikipedia: [http://pl.wikipedia.org/wiki/Operacja\\_wejścia\\_wyjścia\\_na\\_sekundę](http://pl.wikipedia.org/wiki/Operacja_wejścia_wyjścia_na_sekundę)

Poniżej znajdują się orientacyjne dane dotyczące wydajności różnych podsystemów dyskowych:

Dysk	Orientacyjna wartość IOPS <sup>2</sup>
Dysk SATA 5400	~50-80
Dysk SATA 7200	~75-100
Dysk SAS 10k	~140
Dysk SAS 15k	~175-210
Dysk SSD	~400-1000000 <sup>3</sup>

Wymagania co do podsystemu dyskowego zależą od wielkości instalacji to znaczy wielkości bazy oraz ilości równoczesnych użytkowników.

Jako zgrubną zasadę można przyjąć, że dla mniejszych baz danych (<2 GB) należy zarezerwować około 15-25 IOPS dla użytkownika. Natomiast dla większych baz (> 2 GB) dla jednego użytkownika należy przeznaczyć 25-40 IOPS lub więcej.

Jak widać Rozwiązaniem godnym polecenia jest zastosowanie dysków SSD. Przy czym należy pamiętać o zabezpieczeniu ich przed awarią np. poprzez konfigurację RAID 1. Bardziej szczegółowe informacje dotyczące określenia wymagań aplikacji do podsystemu dyskowego można znaleźć w poniższym odnośniku: [http://msdn.microsoft.com/en-us/library/ee410782\(v=sql.100\).aspx](http://msdn.microsoft.com/en-us/library/ee410782(v=sql.100).aspx)

Porównanie wydajności pracy Comarch ERP Optima z dyskami SSD i SATA znajduje się w biuletynie technicznym: OPT076 - Porównanie wydajności HDD vs SSD w Comarch ERP Optima.pdf, który jest dostępny na Indywidualnych Stronach Partnerów (<https://www.erp.comarch.pl/partnerzy/default.aspx>).

## 2.1.2 Biura rachunkowe:

Kilka mniejszych baz danych słabiej obciąża serwer niż jedna duża. Dlatego przy skalowaniu serwera SQL dla biur rachunkowych należy przede wszystkim sprawdzić jaka będzie wielkość największej bazy danych i pod nią dobierać konfigurację plus zabezpieczyć dodatkową ilość pamięci RAM współmiernie do ilości baz danych. Raczej nie należy przekraczać ilości 50 baz danych o wielkości do 200 MB na rdzeń procesora.

## 2.2 Edycje serwera SQL

Bardzo duże znaczenie dla wydajnej pracy serwera SQL jest właściwe dobranie edycji serwera SQL. **Comarch ERP Optima** jest dystrybuowana z darmową serwerem SQL o nazwie Express. Edycja ta posiada określone ograniczenia co do możliwości wykorzystania zasobów komputera, na którym jest zainstalowana. Szczegóły znajdują się w poniższej tabeli.

<sup>2</sup> <http://www.symantec.com/connect/articles/getting-hang-iops-v13>

<sup>3</sup> <http://en.wikipedia.org/wiki/IOPS>

Edycja	Maksymalna ilość pamięci RAM (dla puli buforów)	Maksymalna wielkość bazy danych	Ilość obsługiwanych procesorów
SQL 2008 Express	1 GB*	4 GB	1
SQL 2008 R2 Express	1 GB*	10 GB	1
SQL 2012 / 2014 Express	1 GB*	10 GB	1 (maksymalnie 4 rdzenie)
SQL 2008 R2 Workgroup	3 GB	Bez ograniczeń (524 PB)	2
SQL 2008 R2 Standard	64 GB	Bez ograniczeń (524 PB)	4
SQL 2012 Standard / BI	64 GB	Bez ograniczeń (524 PB)	4 procesory (do 16 rdzeni)
SQL 2014 Standard / BI	128 GB	Bez ograniczeń (524 PB)	4 procesory (do 16 rdzeni)

Źródło: <http://www.microsoft.com/sqlserver/en/us/product-info/compare.aspx>

\*Badania niezależne od Producenta pokazują, że wersja Express może wykorzystać maksymalnie 1,4 GB RAM (<http://sqlgeek.pl/2010/08/23/pl-sql-server-limity-w-sql-server-2008-r2-express-edition/>)

### 2.2.1 Architektura 32 bit, a 64 bit.

Obecnie obowiązującą architekturą jest architektura 64 bitowa i w miarę możliwości zalecana jest aktualizacja do niej środowisk 32 bitowych, które posiadają ograniczenia związane z ilością adresowanej pamięci, a także różnymi komplikacjami w jej alokacji. W podstawowej konfiguracji proces 32 bitowy może maksymalnie zaadresować 2 GB pamięci, przy zastosowaniu specjalnego przełącznika można tą wartość zwiększyć do 3 GB, ale dzieje się to kosztem ilości dostępnej przestrzeni adresowej dla systemu operacyjnego dlatego należy robić to ostrożnie. Serwer SQL może dodatkowo wykorzystać mechanizm AWE, który pozwala na systemach 32 bitowych wyjść poza zakres 4 GB pamięci. Szczegóły można znaleźć w archiwalnym biuletynie technicznym OPT041-Wydajność Comarch OPTIMA a procesory wielordzeniowe i 64 bitowe, który jest dostępny na stronach walidowanych.

Na systemach 64 bitowych warto zwrócić uwagę, aby instalować również serwer SQL w wersji 64 bitowej ponieważ jego 32 bitowy odpowiednik będzie w stanie wykorzystać jedynie 4 GB z dostępnej pamięci nawet w edycji Standard.

Więcej informacji na temat możliwości wykorzystania pamięci operacyjnej przez poszczególne wersje systemów operacyjnych Windows można znaleźć tutaj:

[http://msdn.microsoft.com/en-us/library/windows/desktop/aa366778\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa366778(v=vs.85).aspx)

## 3 Konfiguracja serwera SQL

Większość parametrów serwera SQL należy pozostawić bez zmian, ponieważ domyślne wartości są optymalne dla większości warunków pracy. Sugerujemy jednakże zwrócić uwagę na Maksymalną / Minimalną ilość wykorzystywanej pamięci RAM.

### 3.1 Minimalna i Maksymalna ilość wykorzystywanej pamięci RAM

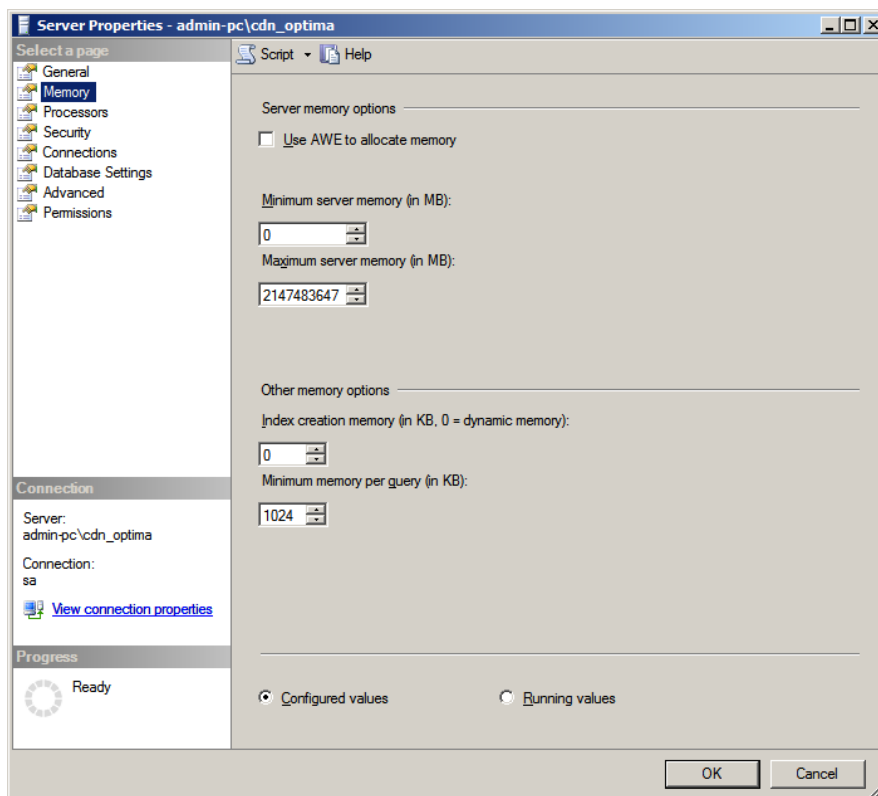
Parametr minimalnej i maksymalnej ilości wykorzystanej pamięci RAM odnosi się wyłącznie do jednego z komponentów serwera SQL, czyli puli buforów. Jest to kluczowy element jednakże, przy rezerwowaniu pamięci dla puli buforów należy wziąć pod uwagę również inne składniki samego serwera SQL jak i samego systemu operacyjnego.

#### 3.1.1 Minimum Server Memory

Domyślną wartością parametru Minimum Server Memory jest zero, co oznacza, że serwer będzie dynamicznie zarządzał dostępną pamięcią RAM dla puli buforów. Ustawienie tego parametru powyżej zera oznacza, że serwer SQL nie będzie mógł zwolnić tej pamięci w razie potrzeby. Z drugiej strony w środowisku, gdzie pracuje więcej aplikacji prócz samego serwera SQL może być konieczne zarezerwowanie niezbędnego minimum pamięci, ponieważ niemożność zaalokowania niezbędnego minimum spowoduje konieczność korzystania z pliku wymiany na dysku twardym i znaczną degradację wydajności Serwera SQL. Na komputerach dedykowanych do pracy tylko z serwerem SQL zaleca się pozostawienie domyślnej wartości tego parametru.

#### 3.1.2 Maximum Server Memory

Domyślną wartością tego parametru to: 2147483647 MB, co oznacza że Server SQL będzie chciał zająć całą dostępną pamięć na komputerze. Może to niestety prowadzić do spadku wydajności całego środowiska poprzez to, że pula buforów zajmie pamięć potrzebną do działania systemu operacyjnego lub innych komponentów serwera SQL. Dlatego zaleca się ograniczenie tej pamięci rezerwując niezbędną przestrzeń do działania systemu operacyjnego i pozostałych elementów serwera SQL (oraz ewentualnie dla dodatkowych aplikacji pracujących na tym systemie).



Rysunek 1. Ustawienia pamięci dla instancji Microsoft SQL Server

Poniżej znajdują się sugestie ustawień parametru Maximum Server Memory (przy założeniu, że na serwerze nie pracują dodatkowe aplikacje oraz inne moduły serwera SQL takie jak wyszukiwanie pełnotekstowe, analizy czy raporty). Dane te oczywiście dotyczą wersji, które nie posiadają wbudowanych ograniczeń tak jak wersja Express. Dla wersji Express biorąc pod uwagę wcześniej podane informacje można ustawić Maximum Server Memory na 1400 MB.

Pamięć fizyczna	Maximum Server Memory
2 GB	1500 MB
4 GB	3200 MB
6 GB	4800 MB
8 GB	6400 MB
12 GB	10000 MB
16 GB	13500 MB
24 GB	21500 MB

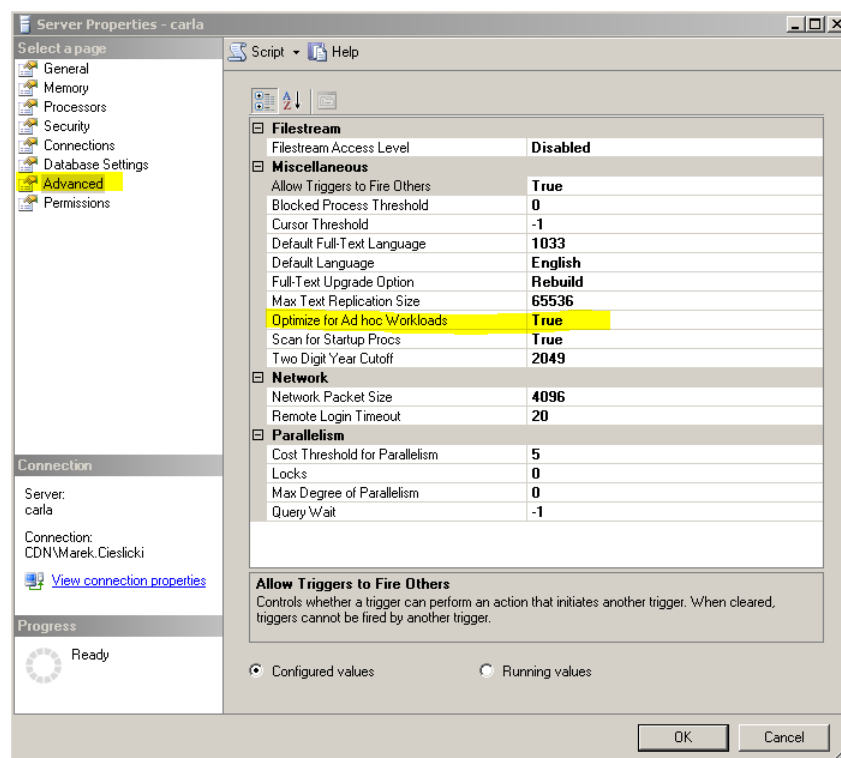
(źródło: [http://www.sqlservercentral.com/blogs/glennberry/2009/10/29/suggested-max-memory-settings-for-sql-server-2005\\_2F00\\_2008/](http://www.sqlservercentral.com/blogs/glennberry/2009/10/29/suggested-max-memory-settings-for-sql-server-2005_2F00_2008/))



## 3.2 Inne parametry

### 3.2.1 Optimize for Ad hoc Workloads

Włączenie parametru „Optimize for Ad hoc Workloads” pozwala na lepsze wykorzystanie dostępnej pamięci RAM w sytuacji, gdy na serwerze generowane jest wiele zapytań, które nigdy więcej lub bardzo rzadko są uruchamiane ponownie. Parametr ten powoduje, że plany dla zapytań „ad hoc” nie są zapisywane w pamięci podręcznej do późniejszego wykorzystania. Zapisywany jest tylko ich mały fragment, cały plan jest zapisywany dopiero przy powtórnym wykonaniu tego samego zapytania. W ten sposób oszczędzana jest pamięć RAM, przez co może być ona wykorzystana przez serwer SQL do innych celów, a co za tym idzie zwiększa się jego wydajność.



Rysunek 2. Zaawansowane właściwości serwera SQL – włączony parametr Optimize for Ad hoc Workloads  
Zaleca się włączenie tego parametru przy pracy z Comarch ERP Optima.  
Opcja ta jest dostępna od wersji SQL 2008.

Zajętość pamięci planów dla zapytań można sprawdzić poniższym zapytaniem:

```
select objtype,  
count(*) as number_of_plans,  
sum(cast(size_in_bytes as bigint))/1024/1024 as size_in_MBs,  
avg(usecounts) as avg_use_count  
from sys.dm_exec_cached_plans  
group by objtype
```

Poniższy wynik wskazuje, że plany typu „Adhoc” zajmują 1439 MB pamięci, co jest stosunkowo dużą wartością więc warto włączyć opcję „Optimize for ad hoc workloads”.

	objtype	number_of_plans	size_in_MBs	avg_use_count
1	UsrTab	224	10	49
2	Prepared	1554	131	20
3	View	340	42	55
4	Adhoc	13989	1439	2
5	Check	130	5	19
6	Trigger	229	161	18
7	Proc	417	152	215

Za pomocą komendy:

```
DBCC FREESYSTEMCACHE ('SQL Plans')
```

można wyczyścić pamięć podręczną planów dla planów typu Adhoc.

## 4 Diagnoza problemów wydajnościowych

Temat diagnozy wydajności serwera SQL jest bardzo obszerny, jednakże poniżej wybrano kilka podstawowych wskaźników, które warto sprawdzić, gdy występują problemy wydajnościowe.

Wszystkie poniższe wskaźniki dostępne są z poziomu systemu operacyjnego.

Narzędzia administracyjne \ Monitor wydajności

Nazwa parametru	Zalecane wartości	Zalecane działania
Procesor: Czas procesora [%]	< 80%	Jeżeli wartość tego parametru przynajmniej kilka razy dziennie na dłuższy czas przekracza zalecaną wartość należy zaplanować dołożenie drugiego procesora lub jego wymianę na wydajniejszy
System: Processor Queue Length	< 2 (dla rdzenia)	Jeżeli wartość tego parametru przynajmniej kilka razy dziennie na dłuższy czas przekracza zalecaną wartość należy zaplanować dołożenie drugiego procesora lub jego wymianę na wydajniejszy
Pamięć: Strony/s	< 20	Zbyt mała ilość dostępnej pamięci RAM
Pamięć: Dostępne bajty	> 300	Zbyt mała ilość dostępnej pamięci RAM
Dysk fizyczny: Czas dysku [%] (poddzielone przez ilość dysków)	< 55%	Wysoka wartość tego parametru może wskazywać na nie wystarczająco szybki podsystem dyskowy lub zbyt małą ilość dostępnej pamięci RAM.
Dysk fizyczny: Średnia długość kolejki dysku (podzielone przez ilość dysków)	< 2 (najlepiej w okolicach zera)	Wysoka wartość tego parametru może wskazywać na nie wystarczająco szybki podsystem dyskowy lub zbyt małą ilość dostępnej pamięci RAM.
SQL Server Buffer: Buffer Cache Hit Ratio	> 90% (najlepiej w granicach 99%)	Zbyt mała ilość dostępnej pamięci RAM dla serwera SQL
Page life expectancy	> 300	Zbyt mała ilość dostępnej pamięci RAM dla serwera SQL

Zbyt mała ilość dostępnej pamięci dla serwera SQL może wynikać z poniższych czynników:

- Za mało pamięci w serwerze
- Ograniczenia wersji Express
- Nieprawidłowo skonfigurowany parametr Max Server Memory
- Ograniczenia architektury 32 bitowej

Importy dużych dokumentów poprzez pracę rozproszoną lub inne mechanizmy mogą na dłuższy czas blokować dostęp do tabel, a przez co powodować wydłużenie czasu operacji dla pozostałych użytkowników. Dlatego zaleca się, aby szczególnie duże importy były wykonywane poza godzinami pracy innych użytkowników.

Jeżeli jakiś scenariusz działania okazuje się szczególnie wolny prosimy o opisanie go krok po kroku i zgłoszenie go przez System Obsługi Zgłoszeń (SOZ).

## 5 Optymalizacja dużych baz danych

Program jest dostosowany do pracy w większości warunków, jednakże niektóre szczególnie duże bazy ze względu na specyficzny rozkład danych w tabelach mogą wymagać dodatkowej optymalizacji. Za duże bazy danych uważamy te, które mają rozmiar rzędu kilku gigabajtów lub więcej. Przed przystąpieniem do poniższych czynności dobrze upewnić się, czy serwer jest prawidłowo skonfigurowany i posiada odpowiednią ilość zasobów.

### 5.1 Wyszukiwanie brakujących indeksów

Microsoft SQL Server posiada wbudowane mechanizmy, które pozwalają określić orientacyjnie jakich indeksów może brakować. W tym celu można uruchomić na serwerze następujące zapytanie:

```
select d.*
    , s.avg_total_user_cost
    , s.avg_user_impact
    , s.last_user_seek
    ,s.unique_compiles
from sys.dm_db_missing_index_group_stats s
    ,sys.dm_db_missing_index_groups g
    ,sys.dm_db_missing_index_details d
where s.group_handle = g.index_group_handle
and d.index_handle = g.index_handle
order by s.avg_user_impact desc
go
```

(źródło:

[http://www.google.pl/url?sa=t&rct=j&q=performance\\_tuning\\_waits\\_queues.doc&source=web&cd=1&ved=0CFkQFjAA&url=http%3A%2F%2Fdownload.microsoft.com%2Fdownload%2F4%2F7%2Fa%2F47a548b9-249e-484c-abd7-29f31282b04d%2FPerformance\\_Tuning\\_Waits\\_Queue.doc&ei=IryYT\\_\\_PacXk4QSOxJzRaw&usg=AFQjCNFRle bSjMLnry8gH99CQklhdmoKJw](http://www.google.pl/url?sa=t&rct=j&q=performance_tuning_waits_queues.doc&source=web&cd=1&ved=0CFkQFjAA&url=http%3A%2F%2Fdownload.microsoft.com%2Fdownload%2F4%2F7%2Fa%2F47a548b9-249e-484c-abd7-29f31282b04d%2FPerformance_Tuning_Waits_Queue.doc&ei=IryYT__PacXk4QSOxJzRaw&usg=AFQjCNFRle bSjMLnry8gH99CQklhdmoKJw))

Zapytanie to najlepiej uruchomić po dłuższej pracy użytkowników wykonujących swoje zadania z **Comarch ERP Optima** w szczególności w obszarach, gdzie zauważają problemy wydajnościowe. Należy pamiętać, że dane o brakujących indeksach są usuwane po restarcie serwera SQL.

Przykładowy wynik powyższego zapytania (pominięto kilka kolumn, aby zachować czytelność).

database_id	equality_columns	inequality_columns	included_columns	statement	avg_total_user_cost	avg_user_impact
15	[TrN_TypDokumentu], [TrN_Bufor], [TrN_DataWys]	NULL	NULL	[CDN_GKop].[CDN].[TraNag]	0,128988319	97
15	[TrN_TypDokumentu], [TrN_Bufor]	[TrN_DataWys]	[TrN_RazemNetto], [TrN_ZwroconoCalalosc]	[CDN_GKop].[CDN].[TraNag]	0,129070432	96,94
15	[TrN_TypDokumentu], [TrN_Anulowany]	NULL	[TrN_RazemNetto]	[CDN_GKop].[CDN].[TraNag]	0,126708136	96,83
15	[TrN_Bufor], [TrN_DataOpe]	[TrN_TypDokumentu]	NULL	[CDN_GKop].[CDN].[TraNag]	0,135401029	92,01
15	[TrN_TypDokumentu], [TrN_Bufor], [TrN_DataOpe]	NULL	NULL	[CDN_GKop].[CDN].[TraNag]	0,134590229	91,96
15	[TrN_TypDokumentu], [TrN_Anulowany]	NULL	[TrN_TrNID], [TrN_RazemNetto]	[CDN_GKop].[CDN].[TraNag]	0,141105667	86,95
15	[TrN_TypDokumentu], [TrN_Anulowany]	NULL	[TrN_TrNID], [TrN_RazemNetto], [TrN_RazemBrutto]	[CDN_GKop].[CDN].[TraNag]	0,141538904	86,68
15	[Bzd_Kierunek], [Bzd_Rozliczono]	[Bzd_DataReal]	[Bzd_BRaID], [Bzd_PodmiotTyp], [Bzd_KwotaSys]	[CDN_GKop].[CDN].[BnkZdarzenia]	0,04641686	66,81

Po pierwsze patrzymy na kolumny avg\_user\_impact oraz avg\_total\_user\_cost, podają one zysk jaki można uzyskać przy zastosowaniu danego indeksu. Pierwszy z nich avg\_user\_impact wskazuje na procentową poprawę w zmniejszeniu kosztu wykonania zapytań użytkownika. Drugi natomiast avg\_total\_user\_cost podaje całkowity zysk w koszcie wykonywanych przez użytkownika zapytań.

Należy mieć również świadomość, że podejście to ma również swoje ograniczenia. Najbardziej istotne z nich jest takie, że podane dane należy traktować jako sugestię, a nie jako konieczność. Pozostałe ograniczenia podane są tutaj: [http://msdn.microsoft.com/en-us/library/ms345485\(v=sql.105\).aspx](http://msdn.microsoft.com/en-us/library/ms345485(v=sql.105).aspx)

Poniżej znajduje się skrypt używany przez Dział Wsparcia Microsoft:

(<http://msdn.microsoft.com/en-us/library/ms345421.aspx>)

```
PRINT 'Missing Indexes: '
PRINT 'The "improvement measure" column is an indicator of the (estimated) improvement that
might '
PRINT 'be seen if the index was created. This is a unitless number, and has meaning only
relative '
PRINT 'the same number for other indexes. The measure is a combination of the
avg_total_user_cost, '
PRINT 'avg user impact, user seeks, and user scans columns in
sys.dm_db_missing_index_group_stats.'
PRINT ''
PRINT '-- Missing Indexes --'
SELECT CONVERT (varchar, getdate(), 126) AS runtime,
    mig.index_group_handle, mid.index_handle,
    CONVERT (decimal (28,1), migs.avg total user cost * migs.avg user impact * (migs.user seeks
+ migs.user_scans)) AS improvement_measure,
    'CREATE INDEX missing_index_' + CONVERT (varchar, mig.index_group_handle) + '_' + CONVERT
(vvarchar, mid.index_handle)
    + ' ON ' + mid.statement
    + ' (' + ISNULL (mid.equality_columns, '')
    + CASE WHEN mid.equality_columns IS NOT NULL AND mid.inequality_columns IS NOT NULL THEN
', ' ELSE '' END + ISNULL (mid.inequality_columns, '')
    + ')'
    + ISNULL (' INCLUDE (' + mid.included_columns + ')', '') AS create_index_statement,
    migs.*, mid.database_id, mid.[object_id]
FROM sys.dm_db_missing_index_groups mig
INNER JOIN sys.dm_db_missing_index_group_stats migs ON migs.group_handle =
mig.index_group_handle
INNER JOIN sys.dm_db_missing_index_details mid ON mig.index_handle = mid.index_handle
WHERE CONVERT (decimal (28,1), migs.avg_total_user_cost * migs.avg_user_impact *
(migs.user_seeks + migs.user_scans)) > 10
ORDER BY migs.avg total user cost * migs.avg user impact * (migs.user seeks + migs.user scans)
DESC
PRINT ''
GO
```

Przygotowuje on od razu definicję indeksu wraz z przykładową nazwą sortując indeksy po polu „miara poprawy”, które jest wyliczane biorąc pod uwagę wspomniane wcześniej kolumny: avg\_total\_user\_cost, avg\_user\_impact, oraz dodatkowo user\_seeks i user\_scans z sys.dm\_db\_missing\_index\_group\_stats.



**Uwaga:** Należy przygotować skrypt dodający oraz usuwający dodawane indeksy, ponieważ dodatkowe indeksy uniemożliwią wykonanie konwersji do nowej wersji programu. Po skonwertowaniu bazy można na nowo dodać przygotowane indeksy



**Uwaga:** Dodane indeksy mogą mieć negatywny wpływ na operacje dodawania, aktualizowania i usuwania rekordów w bazie, dlatego należy przetestować ich wpływ na całościowe funkcjonowanie programu.

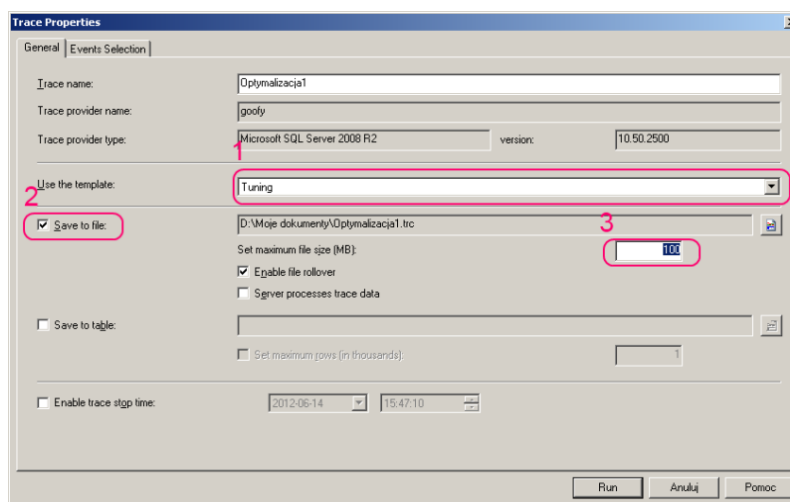
## 5.2 Database Engine Tuning Advisor

Pełne wersje Microsoft SQL Server posiadają dodatkowe narzędzie o nazwie Database Engine Tuning Advisor. Pozwala ono również na dodanie brakujących indeksów na podstawie zapisanego wcześniej ruchu SQL. Ruch ten można zapisać za pomocą innego narzędzia Microsoft SQL Server Profiler, które również jest dostępne w pełnej wersji serwera SQL.

W skrócie proces optymalizacji można przedstawić w poniższych krokach:

- Zapis ruchu za pomocą Microsoft SQL Server Profiler
- Przygotowanie optymalizacji w Database Engine Tuning Advisor
- Zapis rekomendacji
- Przygotować plik usuwający dodatkowe indeksy i statystyki
- Utworzenie dodatkowych indeksów i statystyk

### 5.2.1 Zapis ruchu za pomocą Microsoft SQL Server Profiler



Po wyborze nowego trace'a należy wskazać szablon Tuning, zaleca się zapisać plik od razu na dysk i ograniczyć jego rozmiar np. do 100 MB. Zaznaczony domyślnie parametr Enable file rollover powoduje utworzenie nowego pliku po osiągnięciu zadanego limitu. Zbyt duże pliki znacząco zwiększają czas analizy przez Database Engine Tuning Advisor.

### 5.2.2 Przygotowanie optymalizacji w Database Engine Tuning Advisor

Po zakończeniu zapisu ruchu przechodzimy do Database Engine Tuning Advisor.

Na pierwszej zakładce „General” nowej sesji wskazujemy plik (File) jako źródło ruchu do optymalizacji oraz podajemy bazę do analizy. Na drugiej zakładce „Tuning Options” pozostawiamy domyślne parametry.

**General** | **Tuning Options**

Session name: 2012-06-14 15:01:45

Workload: ☒ File ☐ Table

D:\Moje dokumenty\Optymalizacja1.trc

Database for workload analysis: CDN\_Prezentacja\_KH\_6

Select databases and tables to tune:

Name	Selected Tables
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_Prezentacja_KH_6	309 of 309
CDN_Prezentacja_KH_6/...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>
CDN_...	<a href="#">Click to select individual tables</a>

☒ Save tuning log

Description: Provide a new session name. In the Workload section, select a database to which Database Engine Tuning Advisor will also change the database context while analyzing the

---

**General** | **Tuning Options**

☒ Limit tuning time

Stop at: 14 czerwca 2012 16:01

Physical Design Structures (PDS) to use in database

☐ Indexes and indexed views ☐ Indexed views ☐ Include filtered indexes

☒ Indexes ☐ Nonclustered indexes

☐ Evaluate utilization of existing PDS only

Partitioning strategy to employ

☒ No partitioning ☐ Full partitioning

☐ Aligned partitioning

Physical Design Structures (PDS) to keep in database

☐ Do not keep any existing PDS ☐ Keep indexes only

☒ Keep all existing PDS ☐ Keep clustered indexes only

☐ Keep aligned partitioning

W celu uruchomienia analizy klikamy ikonę Start Analysis (menu Actions \ Start Analysis)

General   Tuning Options   Progress   Recommendations   Reports						
Estimated improvement: 15%						
Partition Recommendations						
Index Recommendations						
Database Name	Object Name	Recommendation	Target of Recommendation	Details	Partition Scheme	Size (KB)
[CDN] [Kontrahenci]	create	create	_dla_stat_1586104691_3_86			
	create	create	_dla_stat_1586104691_3_79_86			
	create	create	_dla_stat_1586104691_79_86			
	create	create	_dla_stat_1586104691_79_1			
	create	create	_dla_stat_1586104691_118_86			
	create	create	_dla_stat_1586104691_1_3_86_79			
	create	create	_dla_stat_1586104691_118_79_86			
	create	create	_dla_stat_1586104691_1_86			
	create	create	_dla_stat_91147370_4_65_3			
	create	create	_dla_stat_91147370_37_1			
[CDN] [Towary]	create	create	_dla_stat_91147370_57_1			
	create	create	_dla_stat_91147370_65_3_2			
	create	create	_dla_stat_91147370_65_1_4_3			
	create	create	_dla_stat_91147370_3_2_1			
	create	create	_dla_stat_91147370_1_26_38_4			
	create	create	_dla_index_Towary_12_91147370_K4_K65_K2_K3_1			56
	create	create	_dla_stat_91147370_1_3			
	create	create	_dla_stat_91147370_1_67			
	create	create	_dla_stat_91147370_1_65_3_2_4			
	create	create	_dla_stat_123147484_20_19_2_36			
[CDN] [Trosk]	create	create	_dla_stat_123147484_20_19_36			
	create	create	_dla_stat_123147484_20_2_36			
	create	create	_dla_stat_123147484_2_9_1_7			
	create	create	_dla_stat_123147484_20_1			
	create	create	_dla_stat_123147484_20_19_10_36_2			
	create	create	_dla_stat_123147484_20_36_1			
	create	create	_dla_stat_123147484_7_1_2_3			
	create	create				
	create	create				
	create	create				

### 5.2.3 Zapis rekomendacji

Po zakończonej analizie otrzymujemy szacowany wzrost wydajności oraz rekomendacje co do założenia dodatkowych indeksów i statystyk. Przygotowane rekomendacje należy zapisać w pliku poprzez przejście do menu Action \ Save Recommendations.

Z tak przygotowanych rekomendacji można wybrać kilka lub wszystkie indeksy i statystyki, jednakże należy pamiętać, że każdy dodatkowy indeks będzie spowalniał operacje dodawania, aktualizowania i usuwania rekordów w tabeli, której on dotyczy. Następnie wybrane indeksy i statystyki należy zapisać w skrypcie nadając im nazwy, które będą łatwe do identyfikacji. Na koniec należy również przygotować skrypt, który będzie usuwał niestandardowe indeksy i statystyki, ponieważ trzeba je usuwać przed przystąpieniem do konwersji bazy danych. Po zakończeniu konwersji można je ponownie dodać.



**Uwaga:** Należy przygotować skrypt dodający oraz usuwający dodawane indeksy, ponieważ dodatkowe indeksy uniemożliwią wykonanie konwersji do nowej wersji programu. Po skonwertowaniu bazy można na nowo dodać przygotowane indeksy.



**Uwaga:** Dodane indeksy mogą mieć negatywny wpływ na operacje dodawania, aktualizowania i usuwania rekordów w bazie, dlatego należy przetestować ich wpływ na całokształt funkcjonowania programu.



## 6 Cykliczne czynności administracyjne

Cykliczne czynności administracyjne są bardzo istotne z punktu widzenia bezpieczeństwa jak i wydajności serwera SQL i należy traktować je jako obowiązkowe, a nie opcjonalne.

Do najważniejszych czynności administracyjnych można zaliczyć:

- Kopia bezpieczeństwa
- Cykliczne odtwarzanie kopii bezpieczeństwa w celu weryfikacji poprawności backupu.
- Optymalizacja indeksów
- Kontrola poprawności baz danych DBCC CHECKDB

Dla wydajności szczególnie dla większych baz duże znaczenie ma cykliczna optymalizacja indeksów w bazie, które ze względu na swoją defragmentację będą spowalniać pracę Comarch ERP Optima.

W kolejnych podrozdziałach opisano w jaki sposób można zautomatyzować niektóre z wymienionych czynności administracyjnych.

### 6.1 Maintenance Plan – zautomatyzowane strojenie dla pełnych wersji MS SQL

Każda pełna wersja MS SQL posiada mechanizm automatyzacji procesów, których zadaniem jest optymalizowanie bazy danych oraz kreowanie kopii bezpieczeństwa. Raz skonfigurowany zestaw operacji nazywany Maintenance Plan'em, czyli planem serwisowym, może być wielokrotnie uruchamiany w zadanym czasie. Tworzony jest zatem pewien „automat”, który o konkretnej porze wykona za nas operacje, które można wykonać z interfejsu programu Comarch ERP Optima.

W niniejszym podrozdziale przedstawiony został przykład planu serwisowego obejmującego:

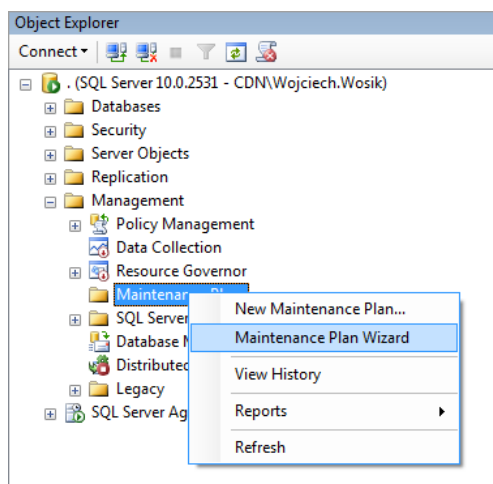
- Sprawdzenie spójności i ciągłości bazy danych (element testów integralności)
- Odbudowę indeksów (ikona pioruna na oknie listy baz danych w konfiguracji)
- Aktualizację statystyk
- Wykonanie kopii bezpieczeństwa bazy konfiguracyjnej oraz firmowej
- Usunięcie plików powstałych na potrzeby wykonywania planu

Materiał został sporządzony przy pomocy MS SQL 2008 za pomocą kreatora. Dla wcześniejszych wersji silnika bazy danych postępuje się podobnie.

#### 6.1.1 Krok 1 – Uruchomienie kreatora Maintenance Plan'u

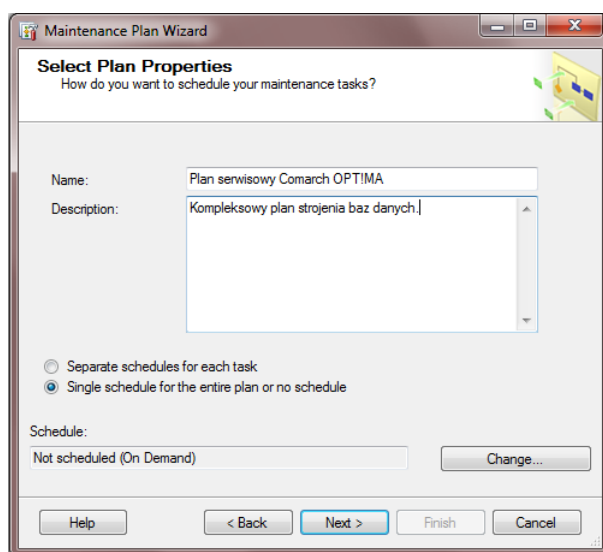
Kreator uruchamiany jest z poziomu programu Management Studio (pełna wersja).

SQL Server /Management/Maintenance Plan Wizard



### 6.1.2 Krok 2 – Wskazanie nazwy planu oraz terminu jego wykonywania

Kolejnym oknem kreatora, na które natrafiamy, jest krótki opis możliwości samego kreatora. Możemy zaznaczyć opcję, aby nie pokazywało się ono następnym razem. Interesuje nas następne okno.



Wypełniamy w nim pola odpowiadające za nazwę oraz opis planu. Deklarujemy następnie jego ramy czasowe. Mamy dwie możliwości:

„Separate schedule for each task” – opcja pozwala na ustalenie terminu wykonania dla każdej operacji planu osobno.

„Single schedule for entire plan or no schedule” – opcja pozwala na ustalenie terminu wykonania całego planu lub wykonania go na życzenie (on demand).

Naciskając przycisk „Change” przechodzimy do okna konfiguracji ram czasowych.

Poniższy obraz przedstawia wybór ram czasowych dla opisywanego przykładu.

Istnieją cztery kombinacje ram czasowych dla planu. Mamy zatem:

„Schedule type” – typy terminarza:

- „Start automatically (...)” – plan uruchomi się tuż po uruchomieniu serwera oraz usługi SQL Agent
- „Start whenever when CPU (...)” – plan uruchomi się, kiedy procesor nie będzie obciążony
- „Recurring” – plan ustala się według interwałów rok / miesiąc / dzień
- „On time” – plan wykona się tylko raz w zadanym dniu i godzinie

W przykładzie użyjemy opcji terminarza według interwału co tydzień. Na obrazku widać, iż ma wykonywać się w każdy piątek o godzinie 18:30. Plan ma rozpocząć się od 15 grudnia 2009 roku i ma nie mieć końca.

Plan ma symulować koniec tygodnia roboczego, kiedy po godzinie 18:00 wszyscy pracownicy skończyli pracę, a serwer SQL nie jest już niczym obciążony.

Poprawne wskazanie ram czasowych dla planu jest bardzo istotne. Uruchomienie go podczas szczytu aktywności pracowników może nawet uniemożliwić im pracę. Podczas optymalizacji serwer pobiera dużo zasobów sprzętowych oraz blokuje elementy struktury bazy danych.

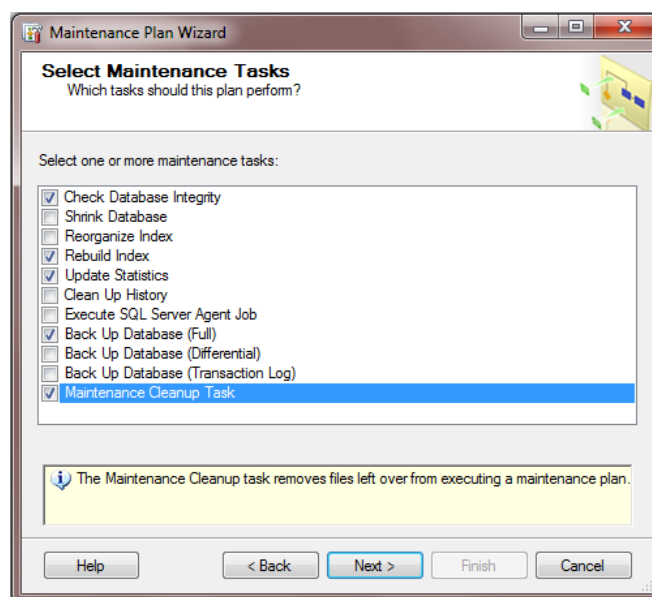
### 6.1.3 Krok 3 – wskazanie składników planu serwisowego

Kolejne okno kreatora planu przedstawia zakres czynności, jakie można wykonać. Są to:

- „Check Database Integrity” – kontrola poprawności struktury bazy danych. Nie można mylić jej z testami integralności, które walidują poprawność samych danych.
- „Shrink Database” – pomniejszenie wielkości bazy danych poprzez usunięcie już niepotrzebnej rezerwy. Serwer SQL podczas pracy alokuje nowe zasoby na potrzeby przyszłych danych oraz operacji. Nie usuwa natomiast powstałym w ten sposób nadwyżki, kiedy dane zostaną usunięte a operacja się zakończy. Shrink pozwala na uwolnienie tych danych. Należy jednak pamiętać o tym, iż usunięcie nadmiarowych danych skutkuje fragmentacją dysku i może negatywnie wpływać na wydajność SQL. Nie zaleca się shrinkowania bazy danych tuż po wykonaniu odbudowy indeksów. Może to przynieść efekt odwrotny od zamierzonego.

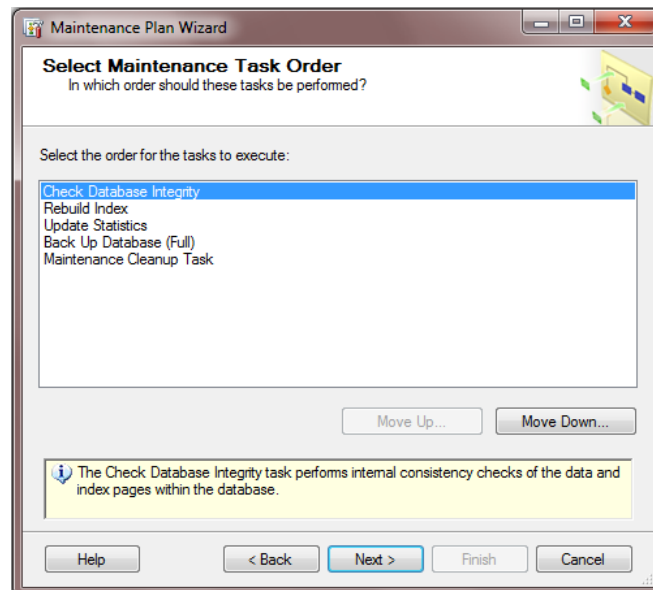
- „Reorganize Index” – defragmentacja indeksów.
- „Rebuild Index” – odbudowa indeksów na nowo. Operacja ta jest bardziej długotrwała aniżeli defragmentacja, lecz daje lepsze efekty.
- „Update Statistics” – aktualizacja „query optimizer’a” dzięki której serwer będzie potrafił wydajniej wykonywać polecenia.
- „Clean Up History” – usuwanie historii wykonywania i odtwarzania kopii baz danych.
- „Execute SQL Server Agent Job” – wykonanie „Job’a”.
- „Back Up Database” – wykonanie kopii bezpieczeństwa bazy danych.
- „Maintenance Cleanup Task” – usunięcie plików powstałych podczas wykonywania planu.

Poniższy obraz przedstawia wybór opcji, które realizowane są w opisywanym przykładzie.



#### 6.1.4 Krok 4 – wskazanie kolejności wykonywania składników planu serwisowego

Kolejne okno kreatora planu pozwala na zadecydowanie, w jakiej kolejności powinny wykonać się poszczególne składniki planu serwisowego.



### 6.1.5 Krok 5 – wskazanie baz danych dla każdego elementu planu serwisowego

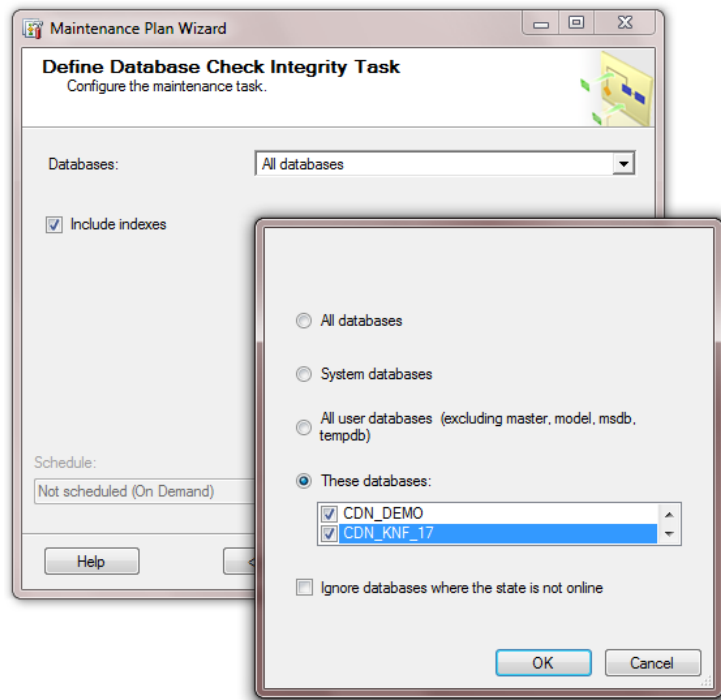
Każdy element planu serwisowego można wykonać dla dowolnego zestawu baz danych. Naszym celem będzie baza konfiguracyjna i baza firmowa.

Krok ten pozwala na elastyczność podczas tworzenia planu dla serwera przechowującego dużą ilość baz danych. Każdej z nich można przypisać oddzielne punkty planu.

Ogólna zasada wyboru baz danych jest wspólna dla wszystkich elementów planu serwisowego. Możemy wybrać:

- „All databases” – operacja zostanie wykonana na wszystkich bazach danych znajdujących się na serwerze.
- „System databases” – operacja zostanie wykonana tylko dla systemowych baz danych (przykład nie zakłada żadnych optymalizacji na tym typie baz danych).
- „All user databases” – operacja zostanie wykonana tylko dla baz danych użytkownika, czyli niesystemowych. Dzięki tej opcji mamy pewność, iż zoptymalizujemy wszystkie bazy danych i nie narazimy na niebezpieczeństwo bazy systemowe.
- „These databases” – operacja zostanie wykonana tylko dla zaznaczonych baz danych. Opisywany przykład wykorzystuje tę opcję.
- „Ignore databases (...)” – wszystkie te bazy danych, których stan nie jest „Online” nie będą podlegać optymalizacji.

Poniższy obraz przedstawia wybór baz danych poddawanych sprawdzaniu poprawności. Każde kolejne okno opiera się na podobnym wyborze.



### 6.1.6 Krok 6 – charakterystyczne opcje dla każdego elementu planu serwisowego

Podczas wskazywania, które bazy danych mają zostać zoptymalizowane, możemy ustalić pewne opcje charakterystyczne dla elementu planu serwisowego.

Dla elementów wykorzystanych w przykładzie mamy:

- Check Database Integrity
  - Include indexes – po odznaczeniu proces kontroli nie obejmie indeksów, zajmie więc mniej czasu.
- Rebuild Index
  - Reorganize pages with the default amount of free space – odbudowa indeksów z domyślnym “fill factor”. Oznacza to, iż indeksy zostaną odbudowane zgodnie z ustawieniami zadanymi podczas kreowania bazy danych, czyli przewidziane przez Comarch.
  - Change free space per page percentage to ... – wskazanie wartości “fill factor” samodzielnie.
  - Sort results in tempdb – zastosowanie SORT\_IN\_TEMPDBoption.
  - Keep index online while reindexing – podczas odbudowy indeksy będą możliwe do odczytu (ta opcja dostępna jest dla edycji Enterprise)
- Update Statistics
  - Update: – wybór elementów poddanych aktualizacji.
  - Scan type: – zakres elementów poddanych aktualizacji.
- Back Up Database
  - Sekcja „Create a backup file for every database” – opcje związane z katalogiem, w którym zostaną umieszczone pliki kopii zapasowych.

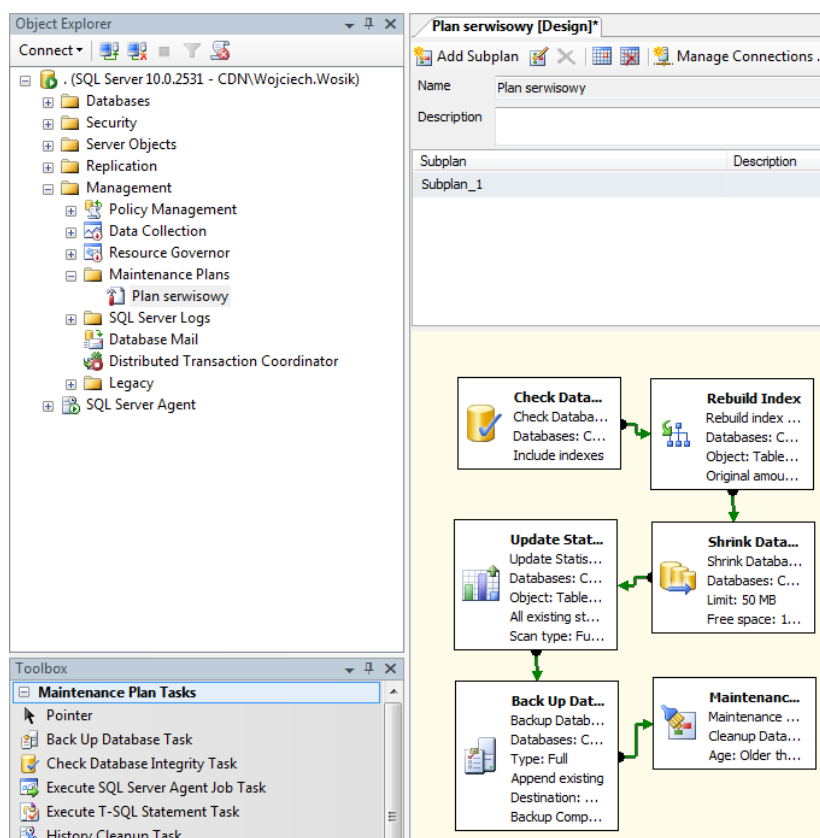
- Maintenance Cleanup Task

- Dostępne opcje pozwalają na zadecydowanie jaki typ danych historycznych ma podlegać kasowaniu. Wskazuje się albo pewien plik, albo folder. Określa się również wymagany wiek pliku.

### 6.1.7 Krok 7 – wskazanie sposobu raportowania wyników wykonania planu serwisowego oraz zakończenie pracy z kreatorem.

Ostatnim etapem tworzenia planu serwisowego jest wskazanie ścieżki dostępu dla pliku raportu. Zawiera on zestawienie podjętych czynności oraz wyniki ich działania.

Po zakończeniu kreatora, nowy plan znajduje się na liście Maintenance Plans w programie Management Studio.



Na tym etapie kończy się konfiguracja planu serwisowego. Wykona się w następny piątek o godzinie 18:30.

Szerszy opis wszystkich elementów Maintenance Plan'u w języku angielskim można uzyskać na stronie:

<http://msdn.microsoft.com/en-us/library/ms188981.aspx>

## 6.2 Plan serwisowy dla bezpłatnych wersji MS SQL

Opisany wcześniej plan serwisowy zbudować można tylko dla płatnych wersji silnika bazy danych. Nic nie stoi jednak na przeszkodzie, aby samemu sporządzić podobną funkcjonalność i zaproponować ją klientowi. Podobny efekt uzyskamy poprzez sporządzenie:

- Skryptu SQL, który zawierał będzie wszystkie potrzebne dla optymalizacji zapytania.
- Skryptu JS/VBS lub pliku .bat, który będzie zdalnie uruchamiał skrypt SQL z poziomu konsoli.
- Harmonogramu systemu Windows, w którym zawrze się ramy czasowe planu.

W niniejszym biuletynie technicznym załączone zostały trzy pliki:

- StrojnieBazy.sql
- StrojnieBazCDN.sql
- StrojnieBazy.vbs

Plik StrojnieBazy.sql to skrypt zawierający wszystkie niezbędne zapytania, aby przeprowadzić plan tożsamy z wcześniej opracowanym Maintenance Plan'em. Przed użyciem należy zmodyfikować w nim dwa parametry:

**SET @Nazwa\_bazy = 'CDN\_DEMO'** -- Tu wpisz nazwę bazy danych do optymalizacji

**SET @Sciezka = 'N:C:\BACKUP'** -- Tu wpisz ścieżkę dostępu do katalogu dla kopii baz

Skrypt optymalizuje tylko jedną, wybraną bazę danych. Został tak sporządzony dla łatwiejszego zrozumienia jego działania. Warto prześledzić jego strukturę.

Drugi skrypt o nazwie StrojnieBazCDN.sql wymaga wskazania ścieżki dostępu do katalogu dla kopii baz. Zastosowano w nim kursor wyszukujący wszystkie bazy danych, których nazwa zaczyna się od 'CDN'. Dla każdej z nich przeprowadzona zostanie optymalizacja.

Plik StrojnieBazy.vbs to skrypt VBS, który uruchamia skrypt SQL. Skrypt tworzy dodatkowo dynamiczną nazwę pliku raportu dla planu serwisowego.

Przed uruchomieniem skryptu należy zmodyfikować w nim następujące elementy:

**sqlcmd.exe -S SERWER -E -i D:\StrojnieBazCDN.sql -o C:\Backup**

- SERWER – nazwa serwera, na którym znajdują się bazy danych.
- D:\StrojnieBazCDN.sql – ścieżka dostępu oraz nazwa skryptu SQL.
- C:\Backup – ścieżka docelowego miejsca składowania kopii baz danych (katalog musi istnieć).

Aby uruchomić automatyczne wykonywanie planu serwisowego należy w harmonogramie systemu Windows utworzyć nową regułę, która będzie w zadanym czasie uruchamiać odpowiedni skrypt VBS. Tym sposobem uzyskamy ten sam efekt, jaki przynieść może Maintenance Plan.



**COMARCH ERP**

Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie na nośniku filmowym, magnetycznym lub innym, powoduje naruszenie praw autorskich niniejszej publikacji.

Copyright © 2016 COMARCH  
Wszelkie prawa zastrzeżone.