

Ścieżkami DB2

POLSKA GRUPA UŻYTKOWNIKÓW DB2

Marzec 2014, NR 1

**DB2 11 Optymalizator:
wykrywanie brakujących
i konfliktujących statystyk**

**Autonomiczne
zbieranie statystyk
w DB2 v10 dla z/OS**

**Top 5 Opportunities
to Reduce Costs and Improve
the Efficiency of DB2
Database Management**

365 dni z PDUG

Sponsor wydania



PDUG
Poland DB2 Users Group

Od Redakcji

Drodzy Czytelnicy!

Do niedawna zastanawialiśmy się wspólnie, z jakim zainteresowaniem będziecie śledzić losy zawiązującego się stowarzyszenia miłośników DB2 i jak chętnie będziecie chcieli je współtworzyć. Ilu członków uda nam się zrzęczyć i czy idea stowarzyszenia przetrwa. Okazało się, że nie tylko istniejemy, nasza społeczność rośnie w siłę, ale też zyskaliśmy rozpoznawalność na forum międzynarodowym. Śmiało możemy powiedzieć, że osiągnęliśmy postawione sobie cele. Jesteśmy z tego dumni i pełni uznania dla Was, Drodzy Czytelnicy. Bez Waszego zaangażowania nie byłoby to możliwe.

Oddajemy w Wasze ręce magazyn, który obok prezentacji wygłaszanych podczas naszych edukacyjnych spotkań, ma na celu dzielenie się wiedzą i doświadczeniem, które tak bardzo cenimy w naszym informatycznym świecie.

Jest to pierwszy numer naszego periodyku. Jego wydanie przypada na jubileusz powstania Polskiej Grupy Użytkowników DB2 i rocznicę pierwszej konferencji zorganizowanej przez PDUG. Pierwszy rok działalności zwińczyło wyróżnienie **The Best New Users Group 2013** przyznane naszemu stowarzyszeniu przez firmę IBM podczas międzynarodowej konferencji **IDUG EMEA** w Barcelonie, a przed nami kolejne ambitne cele do osiągnięcia. Nie ustajemy w wysiłkach, by stowarzyszenie rozwijało się, efektywnie realizowało swoją misję i odpowiadało na potrzeby rozwoju zawodowego jego członków. Od Was - Czytelnicy zależy, czy pismo to wpisze się na stałe w działalność naszej Grupy.

Autorom artykułów powstałych na potrzeby pierwszego numeru periodyku chcielibyśmy złożyć podziękowania i wyrazy uznania za wkład merytoryczny, poświęcony czas oraz chęć dzielenia się wiedzą i doświadczeniem. Dziękujemy naszym koleżankom i kolegom:

Adamowi Peście (Asseco Poland S.A.),

Jackowi Surmie (PKO Bank Polski S.A.),

Jarosławowi Szczepanikowi (CompFort Meridian),

Michałowi Białeckiemu (IBM),

Ani Ogan (CA Technologies).

Narodziny periodyku to bardzo świeży pomysł. Mamy nadzieję rozwijać go i kształtować pismo wedle Waszego uznania. Dlatego liczymy na Wasze nieocenione uwagi, które uwzględnimy przy tworzeniu kolejnych numerów czasopisma PDUG. Zapraszamy również do współtworzenia i aktywnego wpływania na jego zawartość. Piszcie do nas na adres pdug@pdug.pl.

Z zyczeniami miłej lektury

Redakcja

Partner numeru:

CA Technologies was an early pioneer in mainframe software, and since then we have continued to implement new technologies across hundreds of industry-standard products in our mainframe portfolio, that help our customers maximize platform value and performance, and dramatically simplify its management. On the strength of our long-term commitment to the mainframe platform, broad product portfolio and Mainframe 2.0 innovations, CA Technologies is by far the largest mainframe independent software vendor, and the No. 1 or No. 2 vendor in all 10 of the markets in which we compete.

Press and analysts recognize CA for innovation and thought leadership for our Mainframe Modernization products and programs like the Mainframe Software Rationalization program, Mainframe Academy and our internal education program, the Mainframe Associate Software Engineer Program (MASEP).



Anna Ogan, Senior Solution Strategist, CA Technologies, przedstawiciel CA na Polskę.
Anna.Ogan@ca.com

Zapraszam do lektury tekstów moich kolegów z CA Technologies:

How to Maximize the Performance of Your DB2 Databases, Scott Jessee

Top Five Opportunities to Reduce Costs and Improve the Efficiency of DB2 for z/OS Database Management, Osvaldo Ridner

Streaming Operations and Managing Complexity with DB2, Troy Coleman

Przemek Stelmaszewski



Michał Białecki



Małgorzata Delis



Paweł Sękowski



REDAKCJA

Jacek Rafalak



Jarosław Pawelec



Co w numerze?

4

DB2 Plan Stability - Projekt wdrożenia w DB2 V9 dla z/OS

6

Utility RUNSTATS versus Filter Factor

8

365 dni z PDUG

10

Autonomiczne zbieranie statystyk w db2 V10 dla z/OS

13

DB2 11 Optymalizator: wykrywanie brakujących i konfliktujących statystyk

15

Twarze PDUG

DB2 Plan Stability

Projekt wdrożenia w DB2 V9 dla z/OS

Zwiększenie dostępności i wydajności aplikacji

Tekst: Adam Pesta, Asseco Poland S.A.

W wersji 9 DB2 pojawiła się sposobność zarządzania nowymi wersjami aplikacji korzystającymi z DB2 poprzez wykorzystanie cechy zwanej *Plan Stability*. Cecha ta umożliwia zapamiętanie statycznej ścieżki dostępu DB2 dla aplikacji, a następnie utworzenie nowej. Jeśli nowa ścieżka dostępu nie jest wydajna, *Plan Stability* pozwala administratorowi w łatwy sposób przełączyć się do poprzedniej wersji pakietu przez użycie prostej komendy REBIND.

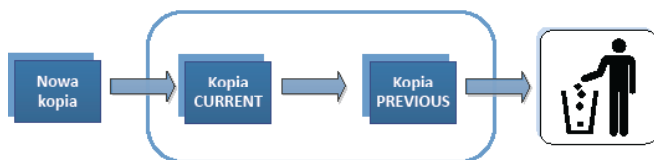
Plan Stability umożliwia zarządzanie dwoma lub trzema kopiami pakietów ze statycznymi SQL'ami. Sterowane jest to przy pomocy parametru konfiguracyjnego PLANMGMT w DB2 ZPARM i przez parametry komendy REBIND. Używając parametru SWITCH komendy REBIND administrator może zdecydować, do której wersji kopii pakietu z dwóch lub trzech wcześniej zapamiętanych chce powrócić.

Cecha *Plan Stability* daje administratorom aplikacji elastyczność w dostępie do tej ścieżki dostępu aplikacji DB2, która jest pożądana. W skomplikowanym systemie dużej instancji DB2 stworzenie procedur, które wykorzystywałyby cechę *Plan Stability*, stwarza szansę na poprawienie dostępności i wydajności aplikacji DB2.

DB2 Plan Stability – DB2 V9

W wersji 9 DB2 pojawił się nowy parametr konfiguracyjny PLANMGMT w makrze DSN6SPRM, który może przyjmować wartości podane poniżej:

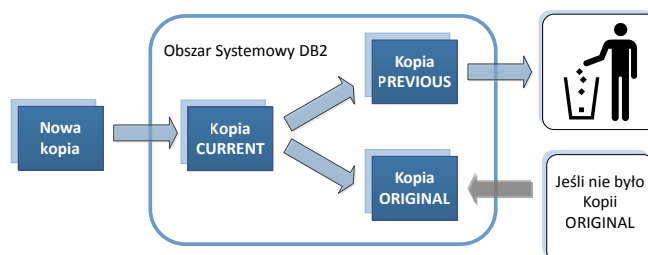
- OFF – default, jedna kopia pakietu nadpisywana w trakcie REBIND,
- BASIC – dwie kopie pakietu, aktualna (*Current*) i poprzednia (*Previous*). W trakcie operacji REBIND tworzona jest nowa ścieżka *Current*, dotychczasowa ścieżka *Current* staje się *Previous* a dotychczasowa *Previous* zostaje usunięta. Schemat opisujący działanie poniżej (Rysunek 1).



Rysunek 1. Kopie pakietu dla PLANMGMT BASIC.

- EXTENDED – trzy kopie pakietu: aktualna (*Current*), poprzednia (*Previous*) i oryginalna (*Original*). Kopia *Current* i *Previous* zachowują się jak w trybie BASIC. Kopia *Original* tworzona jest jako klon kopii *Current* tylko raz podczas pierwszej operacji REBIND z parametrem PLANMGMT(EXTENDED) i nigdy nie jest nadpisywana.

Rysunek 2 przedstawia schemat opisujący działanie parametru.



Rysunek 2. Kopie pakietu dla PLANMGMT EXTENDED.

W komendzie REBIND został wprowadzony nowy parametr PLANMGMT, określający sposób postępowania z dotychczasowymi ścieżkami dostępu dla aplikacji DB2. Może on przyjmować następujące wartości:

- PLANMGMT (OFF) – jeśli któraś z poniższych opcji zostaje zmieniona, to aktualna ścieżka dostępu jest usuwana i zastępowana nową - kopie nie podlegają zmianom.

W przypadku braku zmian poniższych opcji:

- » OWNER
- » QUALIFIER
- » DBPROTOCOL
- » ENABLE
- » DISABLE
- » PATH
- » PATHDEFAULT
- » IMMEDIATEWRITE

wszystkie kopie zostają usunięte;

- Not specified – wartość taka jak w ZPARM dla parametru PLANMGMT;
- PLANMGMT(BASIC) – dwie kopie, jeśli jest kopia *Original*, to nie podlega żadnym zmianom;
- PLANMGMT(EXTENDED) – trzy kopie, jeśli jest *Original* to nie podlega żadnym zmianom, jeśli nie ma *Original*, to zostaje utworzona (klon kopii *Current*).

W komendzie REBIND został wprowadzony nowy parametr SWITCH pozwalający administratorowi aplikacji zdecydować, do której kopii pakietu „przełączyć” aplikację DB2. Może on przyjmować następujące wartości:

- SWITCH(PREVIOUS) – dotychczasowa kopia *Previous* staje się *Current*, dotychczasowa kopia *Current* staje się *Previous*,
- SWITCH(ORIGINAL) – dotychczasowa kopia *Original* staje się *Current*, dotychczasowa kopia *Current* staje się *Previous*, dotychczasowa kopia *Previous* jest usuwana.

W komendzie FREE został wprowadzony nowy parametr PLANMGMTSCOPE, umożliwiający usuwanie kopii ścieżki dostępu aplikacji DB2. Może on przyjmować następujące

wartości:

- PLANMGMTSCOPE(ALL) - usuwa pakiet łącznie z kopiami,
- PLANMGMTSCOPE(INACTIVE) - usuwa wszystkie *Previous* i *Original* kopie pakietu. Po tej operacji można wykonać nową kopię *Original*.

DB2 Plan Stability – DB2 V10

W wersji DB2 V10 cecha *Plan Stability* została rozwinięta. Do katalogu DB2 zostały dodane nowe tablice, co ułatwia zarządzanie kopiami pakietów.

Nowa tablica SYSIBM.SYSPACKCOPY – zawiera dane dla kopii pakietów, analogiczne do tablica SYSIBM.SYSPACKAGE, plus informacje w kolumnach:

- COPYID, której zawartość jednoznacznie przypisuje dane z wierszy odpowiednio do:
 - » 1 – kopii PREVIOUS
 - » 2 – kopii ORIGINAL
- PLANMGMT, która przechowuje informacje o wartości parametru PLANMGMT w trakcie BIND/REBIND:
 - » B – PLANMGMT BASIC
 - » E – PLANMGMT EXTENDED – default w V10
 - » F – PLANMGMT OFF
 - » O – PLANMGMT ON

Poniżej przykład zapytania SQL, które z tablic SYSPACKAGE i SYSPACKCOPY wydobywa informacje o kopiach ścieżek dla pakietów w kolekcji ADMDCOLL:

```
SELECT
SUBSTR(COLLID,1,10) AS COLLID ,SUBSTR(NAME,1,10) AS NAME ,LASTUSED ,VALID
,OPERATIVE ,COPYID ,PLANMGMT ,APRETAINDUP
FROM SYSIBM.SYSPACKAGE
WHERE COLLID = 'ADMDCOLL' AND NAME = 'DSNBCLTC'
UNION ALL
SELECT
SUBSTR(COLLID,1,10) AS COLLID ,SUBSTR(NAME,1,10) AS NAME ,LASTUSED ,VALID
,OPERATIVE ,COPYID ,PLANMGMT ,APRETAINDUP
FROM SYSIBM.SYSPACKCOPY
WHERE COLLID = 'ADMDCOLL' AND NAME = 'DSNBCLTC'
WITH UR;
```

Efekt zapytania to:

COLLID	NAME	LASTUSED	VALID	OPERATIVE	COPYID	PLANMGMT	APRETAINDUP
ADMDCOLL	DSNBCLTC	2013-12-28	Y	Y	0	E	Y
ADMDCOLL	DSNBCLTC	2013-07-27	Y	Y	1	E	Y
ADMDCOLL	DSNBCLTC	2013-03-01	Y	Y	2	E	Y

DSNE610I NUMBER OF ROWS DISPLAYED IS 3

W komendzie REBIND pojawił się parametr APRETAINDUP, określający zachowanie DB2 w przypadku, gdy nowa ścieżka jest identyczna z dotychczasową. Ma to na celu zaoszczędzenie miejsca w SPT01. Parametr może przyjmować następujące wartości:

- APRETAINDUP(YES) – default, motor zachowuje nową i starą kopię pakietu (jak w DB2 V9),
- APRETAINDUP(NO) – zachowuje tylko nowszą kopię pakietu i dzięki temu oszczędza miejsce.

Projekt wdrożenia Plan Stability w wersji 9 DB2

Pojawienie się cechy *Plan Stability* w wersji 9 DB2 wywołało spore zainteresowanie wśród administratorów aplikacji w dużym ośrodku przetwarzania w Polsce. Poniżej kilka wartości, które pozwolą zobrazować skalę środowiska produkcyjnego w tym ośrodku:

- liczba pakietów: > 10000,
- czas trwania operacji REBIND: > 2500 sekund,

- wielkość SPT01: 222525 ścieżek (5 zbiorów po 44505 ścieżek).

Administratorzy aplikacji wiązali pewne nadzieje z cechą *Plan Stability*. Oczekiwano:

- skrócenia czasu operacji REBIND poprzez eliminację operacji dla pakietów, dla których ścieżka dostępu się nie zmieni lub się pogorszy,
- umożliwienia powrotu do „dobrej” ścieżki w przypadku pogorszenia ścieżki dostępu.

Wielkość środowiska produkcyjnego, jak i poziom skomplikowania przetwarzania, stawiały przed próbą wdrożenia cechy *Plan Stability* poważne wyzwania związane z:

- przyrostem SPT01 - przechowywanie dwóch lub, dla niektórych krytycznych pakietów, trzech ścieżek dostępu wiąże się ze wzrostem wykorzystania i potencjalnym wzrostem wielkości obszarów SPT01,
- obsługą informacji o kopiach pakietów - w DB2 V9 brak jest informacji, ile i jakie kopie ścieżek dostępu są przechowywane w zasobach systemowych DB2.

Kwestia przyrostu wielkości SPT01 została oszacowana na poziomie 600 MB, która to wielkość dla kolejnych operacji REBIND PLANMGMT nie zwiększała się. Poważniejszym wyzwaniem było zapewnienie dostępu do aktualnych i wiarygodnych informacji o kopiach ścieżek dostępu zapamiętanych w DB2. Stworzono środowisko do obsługi tych informacji:

- utworzono dodatkowe tablice administracyjne do przechowywania informacji o kopiach pakietów,
- przygotowano zmiany w procedurach eksploatacyjnych, które w trakcie operacji REBIND uaktualniają ww. tablice informacjami o kopiach pakietów,
- w oparciu o tablice EXPLAIN zbudowano mechanizm oceny, czy nowa kopia pakietu będzie lepsza od dotychczasowej.

Opisane powyżej środowisko było dodatkowym kosztem. Przestrzeń potrzebna dla tablic z informacjami o kopiach i tablic EXPLAIN w środowisku produkcyjnym oszacowano na 800 MB.

W wersji 9 DB2 nie udało się zapewnić spójności między tablicą SYSIBM.SYSPACKAGE a tablicami administracyjnymi, przechowującymi informacje o kopiach pakietów. W środowisku produkcyjnym zbyt dużym ryzykiem było podejmowanie decyzji o ewentualnych przełączeniach ścieżek dostępu w oparciu o dane z tablic administracyjnych. Oczekujemy, że w wersji 10 DB2 uda się w pełni wykorzystać cechę *Plan Stability*.

Podsumowanie

Cecha *Plan Stability* wydaje się warta zainteresowania w ramach zwiększania dostępności i wydajności aplikacji. Wszystko wskazuje na to, że producent będzie ją nadal rozwijał. Cecha ta powinna też znaleźć szersze zastosowanie w procesie migracji motoru bazy danych DB2. W dużym środowisku możliwość powrotu do wcześniejszych ścieżek dostępu aplikacji DB2 musi wzbudzać zainteresowanie administratorów aplikacji.

Utility RUNSTATS versus Filter Factor

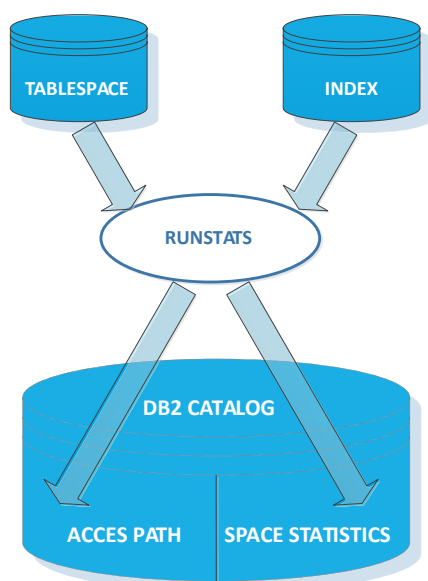
Tekst: Jacek Surma, PKO Bank Polski S.A.

Jednym z wielu zadań, z którymi każdy DBA musi radzić sobie w codziennej pracy, jest dbanie o wydajność aplikacji komunikujących się z DB2. Głównym parametrem determinującym wybór optymalnej ścieżki dostępu do danych przechowywanych w DB2, a tym samym mającym wpływ na wydajność, jest *Filter Factor (FF)*, który odpowiada na pytanie, ile wierszy zostanie zwróconych przez dany predykat ze zdania WHERE. W tej sytuacji zadaniem administratora bazy danych jest dostarczenie optymalizatorowi DB2 (*DB2 z/OS optimizer*) odpowiednich danych potrzebnych do obliczenia *FF* dla każdego predykatu ze zdania WHERE. Niniejszy artykuł ma na celu pomoc w wyborze prawidłowych wartości zdań kontrolnych, czyli parametrów narzędzia (utility) RUNSTATS pod kątem wydajności w wersji 9 oraz 10 DB2.

Utility RUNSTATS

Głównym zadaniem RUNSTATS jest zbieranie statystyk typu:

- ACCESSPATH – konieczne, by optymalizator mógł ustalić optymalną ścieżkę dostępu do danych,
- SPACE – konieczne, by administrator mógł monitorować obiekty DB2 i podjąć akcje związane z utrzymaniem danych w tabelach w „dobrej kondycji”, czyli np. uruchomienie utility REORG itp. Jeśli dodatkowo zbieramy dane historyczne do tabel SYSIBM.*_HIST poprzez uruchomienie narzędzia RUNSTATS z parametrem HISTORY, to mamy wgląd w trend zmiany danych i jesteśmy w stanie wykonać tzw. *capacity planning*. Warto również zaznaczyć, że od wersji 7 DB2 możemy i powinniśmy korzystać z RTS-ów, które w czasie rzeczywistym (domyślnie co 30 minut) zbierają statystyki typu SPACE do dwóch tabel katalogu DB2: SYSIBM.SYSTABLESPACESTATS i SYSIBM.SYSINDEXSPACESTATS.



Rysunek 1. Schemat pracy narzędzia RUNSTATS.

W celu prawidłowego przygotowania narzędzia RUNSTATS pod kątem zbierania statystyk typu ACCESSPATH, musimy wiedzieć, jakich danych potrzebuje optymalizator w celu oszacowania parametru *FF*. Odpowiedź na powyższe znajdziemy we wzorach przedstawionych w poniższej tabeli.

Tabela 1. Wartość parametru Filter Factor w zależności od typu predykatu i jego wartości

L.p.	Typ predykatu	Wartość Filter Factor
1.	COL1 = value	$\frac{\text{Ilość wystąpień wartości: value}}{\text{CARDF}}$
2.	COL1 IS NULL	$\frac{\text{Ilość wystąpień wartości: NULL}}{\text{CARDF}}$
3.	COL1 IN(list)	$\frac{\text{Ilość wystąpień każdej wartości z list}}{\text{CARDF}} * \text{ilość elementów z list}$
4.	COL1 BETWEEN value1 AND value2	$\frac{\text{value2} - \text{value1}}{\text{HIGH2KEY} - \text{LOW2KEY}}$
5.	COL1 LIKE 'char%'	$\frac{\text{char!! FF} - \text{char!! 00}}{\text{HIGH2KEY} - \text{LOW2KEY}}$
6.	COL1 = :hv	$\frac{1}{\text{COLCARDF}}$
7.	COL1 BETWEEN :hv1 AND :hv2	Wartości domyślne w zależności od wielkości COLCARDF Np. jeśli COLCARDF >= 1000 to FF = 1/100

Z powyższego zestawienia wynika, że informacje niezbędne do ustalenia przez optymalizator parametru *FF* możemy podzielić na dwie kategorie. Pierwsza z nich dotyczy predykatów, których wartość jest znana w momencie wykonania SQL-a, czyli odnosi się do pięciu pierwszych pozycji (1-5) z Tabeli 1 (np. package statyczne zbindowane z opcją REOPT(ALWAYS)).

Ze wzorów zebranych w kolumnie *Wartość Filter Factor* Tabeli 1 wynika, że optymalizator potrzebuje wtedy wartości: **CARDF** – całkowita liczba wierszy w tabeli (kolumna z tabeli SYSIBM.SYSTABLES),

HIGH2KEY – druga co do wielkości wartość w kolumnie (kolumna z tabeli SYSIBM.SYSCOLUMNS),

LOW2KEY – przedostatnia co do wielkości wartość w kolumnie (kolumna z tabeli SYSIBM.SYSCOLUMNS),

FREQUENCY – częstotliwość występowania danej wartości w kolumnie,

HISTOGRAM – rozkład danych w kolumnie.

Druga kategoria informacji zbieranych przez utility RUNSTATS dotyczy predykatów zawierających zmienne (ang.: *host variables*), czyli wartości nieznanne podczas polecenia BIND/REBIND – dwie ostatnie pozycje z tabeli powyżej (Tabela 1).

Ze wzorów wynika, że optymalizator potrzebuje wtedy jedynie wartości:

COLCARDF – licznosc kolumny, a więc liczba niepowtarzalnych wartości w kolumnie (kolumna z tabeli SYSIBM.SYSCOLUMNS).

Zdania kontrolne utility RUNSTATS

Uzbrojeni w wiedzę zawartą w Tabeli 1 możemy przygotować poprawnie zdania kontrolne RUNSTATS pod kątem optymalizacji zapytań SQL i w tym celu posłużyć się poniższymi tabelami (Tabela 2 oraz Tabela 3). Zebrane w nich składnie uwzględniają rozróżnienie sytuacji, w których znamy wartości predykatów zanim wywołana zostanie procedura SQL oraz takie, kiedy tych wartości nie znamy.

Tabela 2. Konstrukcja zdań kontrolnych w sytuacji ustalonych wartości predykatów podczas polecenia BIND.

Wartości predykatów są znane w momencie uruchomienia SQL	Zdania kontrolne utility RUNSTATS
Założone są odpowiednie index-y na predykatkach oraz predykaty są typu range (>, <, >=, <=, BETWEEN, LIKE)	TABLE(ALL) COLUMN(ALL) INDEX(ix1) HISTOGRAM KEYCARD FREQUVAL NUMCOLS n COUNT m
Brak index-ów na kolumnach typu range (>, <, >=, <=, BETWEEN, LIKE)	TABLE(ALL) COLUMN(ALL) COLGROUP(col1,col2..) HISTOGRAM FREQUVAL NUMCOLS n COUNT m
Założone są odpowiednie index-y na predykatkach oraz predykaty są typu equal (=, IS NULL, IN(list))	TABLE(ALL) COLUMN(ALL) INDEX(ix1) KEYCARD FREQUVAL NUMCOLS n COUNT m
Brak index-ów na kolumnach typu equal (=, IS NULL, IN(list))	TABLE(ALL) COLUMN(ALL) COLGROUP(col1,col2..) FREQUVAL NUMCOLS n COUNT m

Tabela 3 przedstawia listę zdań kontrolnych dla predykatów przyjmujących wartości zmienne (ang.: *host variables*).

Tabela 3. Konstrukcja zdań kontrolnych RUNSTATS w sytuacji braku wiedzy o wartościach predykatów podczas polecenia BIND.

Wartości predykatów są nie znane (zmienne hostowe) w momencie uruchomienia SQL	Zdania kontrolne utility RUNSTATS
All	TABLE(ALL) COLUMN(ALL)

Podsumowanie

Powyższe zestawienia wskazują, że w sytuacji, gdy mamy do czynienia np. ze statycznymi programami zawierającymi zmienne „hostowe”, to zdanie kontrolne narzędzia RUNSTATS powinno ograniczać się jedynie do: TABLE(ALL) COLUMN(ALL). Optymalizator pominięte pozostałe parametry, a ich uwzględnienie w zdaniach kontrolnych RUNSTATS skutkować będzie jedynie wydłużeniem czasu pracy tego narzędzia i wygenerowaniem dodatkowych kosztów związanych ze zużyciem CPU. W pozostałych przypadkach, czyli np. dla programów dynamicznych, powinniśmy stosować bardziej złożone zdania kontrolne pokazane w Tabeli 2.

Podziękowania za dotychczasową współpracę

Dziękujemy serdecznie firmom Asseco Poland S.A., PKO Bank Polski S.A. i IBM, które zaufały Polskiej Grupie Użytkowników DB2 i dotychczas wspierały nas w organizacji konferencji. Cieszymy się, że kolejne firmy postanowiły przyłączyć się do naszej inicjatywy. CA Technologies oraz Arrow Ecs sp. z o.o. zgodziły się być naszymi partnerami podczas wiosennej konferencji 2014r.

Dziękujemy wszystkim angażującym się w organizację, obsługę logistyczną, dokumentowanie i dbałość o sprawny przebieg naszych konferencji.

Dziękujemy naszym wspieranym autorytetom z IBM LAB za uświetnienie konferencji Polskiej Grupy Użytkowników DB2 swoimi wystąpieniami, a także firmom Ecosoft i Compuware za ich wkład merytoryczny na rzecz stowarzyszenia. Na liście naszych prelegentów znaleźli się: Christian Michel (IBM), Thomas Beavin (IBM), śp. Adam Dutkowski (Ecosoft), Bart Steegmans (IBM), Nigel Slinger (IBM), John Lloyd (Compuware).

Szczególne gratulacje składamy na ręce członków Polskiej Grupy Użytkowników DB2: Jacka Surmy (PKO Bank Polski), Jacka Banasiaka (Accenture), Pawła Hryba (PKO Bank Polski) - naszym dotychczasowym prelegentom na konferencjach PDUG oraz wszystkich autorów tekstów o tematyce technicznej zawartych w tym periodyku za zabranie głosu w dyskusji w zakresie dobrych praktyk i ich wkład w realizację misji stowarzyszenia.

Zarząd PDUG

Poland DB2 Users Group

365 dni z PDUG

Tekst: Jacek Rafalak, *Asseco Poland S.A.*

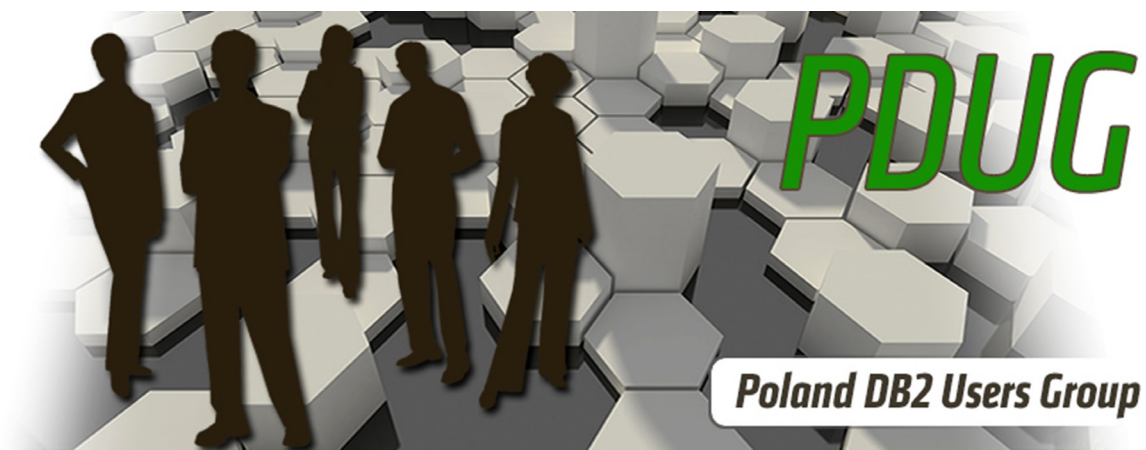
Stowarzyszenie *Polska Grupa Użytkowników DB2* z początkiem 2014 roku świętuje rocznicę swego powstania. Naszym członkom, utrzymującym czołowe instalacje DB2 w Polsce, udało się zbudować społeczność skupioną wokół technologii DB2, stawiając sobie za cel działalność na rzecz wspierania i wzmacniania wspólnoty usług informatycznych.

Krótka powtórka z historii dla tych z Was, których nie było z nami od początku. Inicjatywa rodziła się powoli. W kularowych rozmowach między specjalistami w dziedzinie technologii DB2 z różnych ośrodków przetwarzania w Polsce, podczas rozmaitych konferencji organizowanych w kraju i za granicą, powtarzały się narzekania

Przyglądając się podobnym inicjatywom w Europie i na świecie, postanowiliśmy sami decydować o podnoszeniu własnych kompetencji. Na świecie od lat użytkownicy DB2 zrzeszają się i tworzą grupy regionalne (Regional Users Groups - RUGs), stawiające sobie za cel wymianę doświadczeń, dzielenie się sprawdzonymi sposobami postę-

z/OS i Linux, Unix, Windows (LUW).

W Krajowym Rejestrze Sądowym istniejemy od marca 2013r. i jako inicjatywa edukacyjna możemy już pochwalić się zorganizowaniem dwóch konferencji technicznych. Wystąpienia prowadzili sami członkowie stowarzyszenia, przedstawiciele firm konsultingowych, dostawców oprogramowania



na marketingowy, mało edukacyjny, charakter wielu dostępnych wystąpień. Wysokie koszty szkoleń technicznych, często wiążące się z delegacjami zagranicznymi, powodowały, że nie były one dostępne dla wszystkich. W świetle wciąż rosnących wymagań rynku, wyznaczających potrzeby ciągłej optymalizacji procesów informatycznych działających na rzecz biznesu, wymóg ustawicznego podnoszenia kwalifikacji technologów jest sprawą oczywistą. Sytuacja, w której zapotrzebowanie na poszerzenie wiedzy i umiejętności technicznych, by sprostać oczekiwaniom biznesu jest stałe, a gotowość pracodawców do ponoszenia dodatkowych wydatków na edukację pracowników niska, wydawała się patowa. Na rozwiązanie wpadliśmy my - sami użytkownicy technologii DB2.

powania i promowanie wypracowanych rozwiązań. Na stronie organizacji *International DB2 Users Group (IDUG)* można doliczyć się kilkudziesięciu takich inicjatyw w samej Europie, a niektóre z nich działają z powodzeniem od 25 lat. Zamiast słuchać specjalistów od marketingu zainteresowanych sprzedażą produktów dedykowanych platformie bazodanowej DB2 - stworzyliśmy PDUG – Poland DB2 Users Group, a we wrześniu 2012r. w Jachrance podczas konferencji z/Day namawialiśmy Was do wsparcia zawiązującej się koncepcji. *Polska Grupa Użytkowników DB2*, dla uproszczenia zwana PDUG, jest stowarzyszeniem typu non-profit, skupionym wokół technologii IBM DB2. Naszą misją jest propagowanie wiedzy w zakresie efektywnego wykorzystania motoru bazy danych DB2 na platformach

oraz czołowi eksperci dziedzinowi IBM z Doliny Krzemowej. W przeciągu niespełna roku *Polskiej Grupy Użytkowników DB2* udało się zrzeszyć większość użytkowników technologii DB2 na platformę z/OS z kluczowych ośrodków w Polsce oraz partnerów gotowych wesprzeć stowarzyszenie. W ramach celów na kolejny rok działalności stowarzyszenia, poza kontynuacją działań informacyjnych i edukacyjnych w zakresie technologii IBM DB2 na platformę z/OS, podjęliśmy starania o rozszerzenie inicjatywy o użytkowników platform Linux, Unix, Windows (LUW). Cieszymy się niezmiernie, że pierwsza tegoroczna konferencja inicjuje otwarcie stowarzyszenia na użytkowników LUW. Jesteśmy też dumni, że udało nam się poszerzyć dotychczasową ofertę konferencji o całonocne seminarium adresowane do

specjalistów o profilu technicznym. Dla zachowania transparentności działania stowarzyszenia wyłoniliście zarząd i komisję rewizyjną, a w statucie stowarzyszenia określiliśmy zasady, którymi grupa kieruje się w swej działalności. Pomimo krótkiego okresu istnienia, stowarzyszenie z sukcesem pozyskało partnerów do organizacji konferencji, takich jak Asseco Poland S.A., PKO Bank Polski S.A., CA Technologies czy ARROW Ecs Sp. z o.o. oraz entuzjastów chętnych do wspierania swoich inicjatyw w kolejnych miesiącach. Członkowie stowarzyszenia rekrutują się z takich firm i organizacji jak: Accenture, Asseco Poland S.A., BRE Bank, CA Technologies, CompFort Meridian, IBM, Ministerstwo Sprawiedli-

wości, PKO Bank Polski S.A., PKO S.A. czy Zakład Ubezpieczeń Społecznych. Wśród prelegentów, oprócz użytkowników DB2, PDUG gościł również przedstawicieli firm Ecosoft czy Compuware. PDUG, jako platforma wymiany doświadczeń, sprzyja wymianie i budowaniu wiedzy, wypracowaniu dobrych praktyk oraz tworzeniu rozwiązań optymalnego wykorzystania bazy danych DB2. W takiej atmosferze rodzą się innowacyjne pomysły na optymalizację procesów zarządzania danymi zmierzające do obniżenia kosztów eksploatacji baz danych i zwiększenia produktywności pracy administratorów i

programistów. Motorem działań *Polskiej Grupy Użytkowników DB2* są zgromadzeni wokół niej wolontariusze – specjaliści gotowi zaangażować swój prywatny czas na budowanie organizacji, pozyskiwanie prelegentów zza granicy oraz budowanie i utrzymanie zasobów stowarzyszenia. To właśnie dzięki ich wysiłkom, by inicjatywa PDUG była widoczna na rynku polskim, rozpoznawalna lokalnie oraz wśród międzynarodowej społeczności i żyła, *Polska Grupa Użytkowników DB2* 16 października 2013r. otrzymała wyróżnienie *The Best New Users*



Group 2013 na gali IBM zorganizowanej z okazji 30-lecia technologii DB2. Gala ta towarzyszyła odbywającej się w dniach 13-18.10.2013r. konferencji IDUG EMEA w Barcelonie.

Konferencje organizowane przez PDUG to nie tylko szansa na uczestnictwo w bezpłatnych szkoleniach organizowanych w odpowiedzi na badanie potrzeb uczestników, dostęp do czołowych ekspertów pracujących w laboratoriach IBM DB2, okazja do spotkań i dyskusji w gronie specjalistów na tematy technologiczne - to również możliwość certyfikacji w obszarze zarządzania informacją (*Information Management*)

na bardzo preferencyjnych warunkach dzięki współpracy, jaką stowarzyszenie nawiązało z firmą IBM.

Wystąpienia podczas konferencji organizowanych przez PDUG stanowią również sposobność dla firm na zaznaczenie swojej obecności wśród największych użytkowników technologii DB2 i specjalistów w tym zakresie. Jest to znakomita okazja, by na forum partnerów biznesowych zaprezentować swoje osiągnięcia w obszarze świadczonych usług czy budowy własnych narzędzi. Stowarzyszenie to platforma ludzi dzielących tę samą pasję zawodową, tak z sektora publicznego, jak i prywatnego. Wciąż przyciągamy nowych członków i wolontariuszy chętnych do podejmowania działań na rzecz aktywnie rozwijającej się społeczności i mamy jasno sprecyzowaną

wizję dalszego rozwoju. Stwarzamy naszym członkom możliwość poszerzania i uaktualniania wiedzy, jak również rozwijania kompetencji osobistych, takich jak warsztat prezentera, również w języku angielskim, komunikacja interpersonalna, czy praca w zespole. Dla firm to miejsce, w którym pracownicy nie tylko poszerzają swoje doświadczenia zawodowe, ale również wzmacniają wizerunek firmy wśród innych liczących się partnerów biznesowych, dając świadectwo profesjonalizmu, rzetelności i ciągłego rozwoju kompetencji firm.



Autonomiczne zbieranie statystyk w db2 V10 dla z/OS

Automatyczna konserwacja bazy danych

Tekst: Jarosław Szczepanik, *CompFort Meridian*

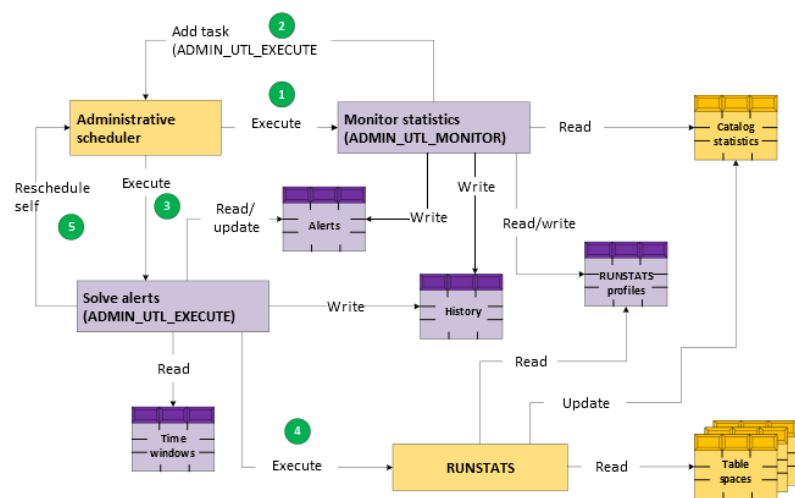
Zadania związane z konserwacją bazy DB2 dla z/OS, takie jak zbieranie statystyk, reorganizacje czy operacje REBIND należą do rutynowych i pracochłonnych czynności, zazwyczaj nielubianych przez administratorów z powodu ich powtarzalności. Z drugiej strony, bez poprawnie realizowanych procedur konserwacyjnych wydajność bazy i aplikacji z niej korzystających szybko ulega degradacji. Dlatego w większości poważnych instalacji wykorzystujących DB2 dla z/OS działają własne lub komercyjne rozwiązania do automatyzacji tego typu zadań. Głównymi celami stawianymi tego typu rozwiązaniom są automatyczna identyfikacja obiektów wymagających określonej czynności konserwacyjnej (np. zebrania statystyk) oraz automatyzacja procesu realizacji danej czynności naprawczej (w naszym przykładzie uruchomienia UTILITY RUNSTATS) w przeznaczonym dla konserwacji oknie czasowym.

Rozbudowane rozwiązania udostępniają szereg dodatkowych funkcjonalności, takich jak np. możliwość nadawania priorytetów, gdy okno na prace utrzymaniowe jest zbyt krótkie dla obsługi wszystkich obiektów, czy kontrola zasobów zużywanych przez zadania konserwacyjne, szczególnie te realizowane w trybie ONLINE, tzn. równoległe z aplikacjami korzystającymi z bazy danych. Do wersji V10 dla DB2 dla z/OS producent nie dostarczał rozwiązania do automatyzacji czynności konserwacyjnych ani z motorem bazy danych, ani z zestawem narzędzi konserwujących (DB2 UTILITIES). Co ciekawe, baza DB2 UDB posiadała takie mechanizmy już od wersji 8.2.

W niniejszym artykule przedstawiono podstawowe informacje na temat rozwiązania automatycznego zbierania statystyk udostępnionego przez firmę IBM wraz z bazą DB2 dla z/OS w wersji 10.1.

Komponenty rozwiązania

Główne komponenty rozwiązania do automatycznego zbierania statystyk przedstawia Rysunek 1.



Rysunek 1. Schemat rozwiązania

Pierwszy komponent, który należy omówić, to **Administrative Task Scheduler (ADMT)**. Jest to dodatkowy DB2 STARTED TASK, który może być uruchamiany i zamykany razem z DB2 automatycznie lub ręcznie. Decydujemy o tym w trakcie instalacji wpisując na panelu DSNTIPX (ZPARM ADMTPROC) nazwę procedury startowej, lub wpisując spację, gdy chcemy startować ADMT ręcznie. Oprócz automatyzacji procesu zbierania statystyk ADMT może być używany do wykonywania automatycznych akcji tuż po starcie DB2 i tuż przed zamknięciem DB2 (np. wyświetlić status UTILITY lub listę obiektów w stanie RESTRICTED tuż przed zamknięciem podsystemu) – oczywiście wtedy praca ADMT musi być kontrolowana przez DB2, a nie ręcznie.

W zadanym momencie czasowym ADMT uruchamia procedurę składowa-

ną **SYSPROC.ADMIN_UTL_MONITOR**, której zadaniem jest analiza statystyk dla zadanego zestawu obiektów (krok 1, Rysunek 1).

Zadanie (TASK) ADMIN_UTL_MONITOR należy wcześniej zdefiniować dla ADMT

przy pomocy procedury składowanej **SYSPROC.ADMIN_TASK_ADD**. Można do tego wykorzystać tradycyjnie SPUFI, ale wygodniej to zrobić z IBM Data Studio, które przejrzysto prezentuje parametry procedury jak to pokazano w Tabeli 1 na stronie 11.

Większość parametrów procedury ADMIN_TASK_ADD jest oczywista. Na uwagę zasługują cztery z nich. Po pierwsze dodajemy zadanie ADMIN_UTL_MONITOR, a więc parametry PROCEDURE_SCHEMA i PROCEDURE_NAME mają odpowiednio wartości SYSPROC i ADMIN_UTL_MONITOR.

Po drugie parametr POINT_IN_TIME definiuje moment automatycznego uruchamiania definiowanego zadania. W przykładzie przedstawionym na Rysunku 2, zapis '3-59/15 * * * *' oznacza, że chcemy wykonywać analizę co kwadrans, zaczynając od 3 minuty każdej godziny.

Tabela 1. Parametry procedury ADMIN_TASK_ADD w IBM Data Studio.

Name	Type	Value	
USERID	VARCHAR(128)		
PASSWORD	VARCHAR(24)		
BEGIN_TIMESTAMP	TIMESTAMP		
END_TIMESTAMP	TIMESTAMP		
MAX_INVOCATIONS	INTEGER		
INTERVAL	INTEGER		
POINT_IN_TIME	VARCHAR(400)	3-59/15****	
TREGER_TASK_NAME	VARCHAR(128)		
TREGER_TASK_COND	CHAR(2)		
TREGER_TASK_CODE	INTEGER		
DB2_SSID	VARCHAR(4)	D0A1	
PROCEDURE_SCHEMA	VARCHAR(128)	SYSPROC	
PROCEDURE_NAME	VARCHAR(128)	ADMIN_UTL_MONIT...	
PROCEDURE_INPUT	VARCHAR(4096)	SELECT `statistics-sc...	
JCL_LIBRARY	VARCHAR(44)		
JCL_MEMBER	VARCHAR(8)		
JOB_WAIT	VARCHAR(8)		
TASK_NAME	VARCHAR(128)	STATMON1	
DESCRIPTION	VARCHAR(128)	Statistics Monitoring...	

Format zmiennej POIN_IN_TIME jest na pierwszy rzut oka nieczytelny, ale tak naprawdę jest to sposób definiowania czasu wykonania zadań znany z UNIX CRON – pierwsza pozycja minuty, dalej odpowiednio godziny, dni miesiąca, miesiące i dni tygodnia. Za bardziej praktyczną ilustrację posłużyć może sytuacja, w której chcielibyśmy wykonywać analizę statystyk raz dziennie o północy. Wtedy POINT_IN_TIME przyjęłoby wartość '0 0 * * *'. Po trzecie parametr **PROCEDURE_INPUT** pozwala na wskazanie, które obiekty mają być analizowane. W naszym przykładzie chcemy analizować tylko obiekty znajdujące się w bazie JSZDB.

```
SELECT 'STATISTICS-SCOPE=PROFILE,
RESTRICT-TS="DBNAME='JSZDB'','',
0, 0, '' FROM SYSIBM.SYSDUMMY1
```

Dokładniejszego opisu parametrów dla procedury ADMIN_UTL_MONITOR trzeba poszukać w dokumentacji. Opis parametryzacji tej procedury jest rozbudowany, ponieważ algorytmy automatycznego wybierania obiektów do konserwacji w praktycznych rozwiązaniach mogą być dosyć złożone. W skrócie, parametr **STATISTICS-SCOPE=PROFILE** oznacza, że sprawdzana jest aktualność statystyk, ich kompletność, a także czy wszystkie statystyki opisane w profilu obiektu zostały zebrane.

Przy okazji należy zwrócić uwagę na tabelę RUNSTATS PROFILES (SYSTABLES_PROFILES) na Rysunku 1. Profil

RUNSTATS jest to zapisany zestaw opcji dla narzędzia RUNSTATS do stosowania dla danej tabeli. Jak widać na Rysunku 1, procedura monitorująca statystyki zarówno czyta, jak i zapisuje profile RUNSTATS.

Mechanizmem profili RUNSTATS, pomyslanym wstępnie do wykorzystania przy automatyzacji z ADMT, można posługiwać się również przy budowaniu innych rozwiązań. Jego główną zaletą jest to, że możemy pracować z jednym, prostym wzorcem zadania RUNSTATS i jednocześnie uruchamiać zbieranie statystyk z wcześniej zdefiniowanym zestawem opcji, innym dla każdej tabeli. Narzędzie RUNSTATS w V10 uzupełniono o odpowiednie funkcjonalności do pracy z profilami. Jeśli wiemy, jaki zestaw opcji (*runstat-options*) chcemy standardowo stosować dla danej tabeli, zapisujemy go przy pomocy frazy SET PROFILE (patrz komenda -1- poniżej).

Jeśli nie wiemy, jakie statystyki zbierano dotychczas dla danej tabeli (szczególnie COLGROUP, FREQVAL, HISTOGRAM, itd.), możemy użyć frazy FROM EXISTING STATS (komenda -2-). Wtedy profil RUNSTATS zostanie zbudowany automatycznie na podstawie istniejących statystyk. Zapisany profil wykorzystujemy pisząc po prostu USE PROFILE (komenda -3-).

```
-1- RUNSTATS TABLESPACE ts-name TABLE table-
name runstat-options SET PROFILE
-2- RUNSTATS TABLESPACE ts-name TABLE table-
name FROM EXISTING STATS
-3- RUNSTATS TABLESPACE ts-name TABLE table-
name USE PROFILE
```

Wracając do definicji zadania ADMIN_UTL_MONITOR, warto zweryfikować, czy dodanie zadania przebiegało poprawnie. Można to zrobić m. in. przy pomocy funkcji ADMIN_TASK_LIST() [DB2 UDF]. Również w tym przypadku w praktyce wygodniejsze będzie użycie IBM DATA STUDIO, ale dla odmiany podano dalej przykład wywołania tej funkcji z poziomu SPUFI.

```
SELECT
SUBSTR(PROCEDURE_SCHEMA,1,8)AS SCHEMA
,SUBSTR(PROCEDURE_NAME,1,20) AS NAME
,SUBSTR(POINT_IN_TIME,1,16) AS PIT
,SUBSTR(PROCEDURE_INPUT,1,24) AS FARMS
,SUBSTR(CREATOR,1,8) AS CREATOR
, LAST MODIFIED
,SUBSTR(DESCRIPTION,1,24) AS DESC
FROM TABLE(DSNADM.ADMIN_TASK_LIST()) AS
TASKLIST
WHERE PROCEDURE_SCHEMA IS NOT NULL
;
```

Kiedy ADMIN_UTL_MONITOR wykrywa obiekt, dla którego trzeba wykonać zebranie statystyk, to rejestruje ten fakt w tablicy ALERTS (SYSIBM.SYSAUTOALERTS) i wystawia wezwanie do procedury ADMIN_TASK_ADD, aby uruchomić zadanie ADMIN_UTL_EXECUTE.

Zadanie naprawcze dodane do ADMT przez ADMIN_UTL_MONITOR można rozpoznać m.in. na podstawie komentarza (kolumna DESCRIPTION) – resolve alerts.

Następnie ADMT uruchamia ADMIN_UTL_EXECUTE (krok 3 na Rysunku 1). Procedura ADMIN_UTL_EXECUTE sprawdza w tablicy nazwanej na Rysunku 1 *Time Windows* (SYSIBM.SYSAUTOTIMEWINDOWS), czy może uruchomić UTILITY RUNSTATS w danym momencie. Jeśli tak to przystępuje do zbierania statystyki (krok 4, Rysunek 1). Jeśli nie, to ADMIN_UTL_EXECUTE sam dodaje się poprzez ADMIN_TASK_ADD do następnego okna czasowego przeznaczonego na konserwację bazy (krok 5, Rysunek 1).

Użytkowanie rozwiązania

W dużym uproszczeniu użytkowanie rozwiązania sprowadza się do:

- przygotowania profili statystyk dla poszczególnych obiektów,
 - zaplanowania okien czasowych na analizę i samo wykonanie operacji RUNSTATS,
 - okresowego weryfikowania działania rozwiązania w razie potrzeby.
- O przygotowaniu profili statystyk dla poszczególnych obiektów wspomniano w poprzednim akapicie. Ogólnie rzecz ujmując, przygotowanie właściwej syntaktyki RUNSTATS dla różnych obiektów jest zadaniem złożonym i kwalifikuje się, jako temat na oddzielny artykuł lub nawet całą serię artykułów. Jeśli uruchomimy rozwiązanie bez profili,

to automatycznie zostaną stworzone profile domyślne. W samym projekcie automatyzacji najatrakcyjniej wygląda opcja tworzenia profili FROM EXISTING STATS lub wykorzystanie istniejących zadań JCL RUNSTATS i dodanie do nich frazy SET PROFILE – zadania wykonują się szybko, bo dodanie SET PROFILE powoduje, że statystyki nie są zbierane, tylko tworzone są profile.

Oczywiście istnieją stosowne opcje do modyfikacji i usuwania profili – odpowiedni UPDATE PROFILE i DELETE PROFILE. Warto zauważyć, że fraza UPDATE PROFILE zmienia tylko te opcje, które są wskazane, pozostawiając pozostałe elementy zapisane w kolumnie PROFILE_TEXT tablicy SYSTABLES_PROFILES bez zmian.

Jeśli chodzi o planowanie okien czasowych, zgodnie z zapisami powyższego akapitu dla ADMIN_UTL_MONITOR wykorzystujemy parametr POINT_IN_TIME. W tym miejscu pozostaje tylko wyjaśnienie sposobu harmonogramowania okien czasowych przeznaczonych na operacje RUNSTATS. Wiemy już, że okna opisane są w tablicy systemowej SYSAUTOTIMEWINDOWS. Tablicę tę możemy modyfikować stosownie do naszych potrzeb przy pomocy SQL DML. Każdy jej wiersz opisuje pojedyncze okno czasowe dla danego podsystemu DB2. Wiersze, dla których wartość kolumny DB2_SSID jest NULL, odnoszą się do wszystkich podsystemów w grupie DATA SHARING.

Dla przykładu, jeśli w podsystemie DSN1 chcemy zdefiniować okno RUNSTATS na każdą niedzielę od północy do godziny 12 w południe, pozwalając na maksymalnie 5 równoległych zadań RUNSTATS, to trzeba dodać wiersz jak w przykładzie:

```
INSERT INTO SYSIBM.SYSAUTOTIMEWINDOWS
(DB2_SSID, MONTH_WEEK, MONTH, DAY, FROM_TIME,
TO_TIME, ACTION, MAX_TASKS)
VALUES ('DSN1', 'W', NULL, 7, '00:00', '12:00:00', NU
LL, 5);
```

Nazwy większości kolumn są samo opisujące się. Komentarza wymagać może jedynie kolumna MONTH_WEEK. War-

tość 'W' w tej kolumnie oznacza WEEK i łącznie z resztą zapisów, jak w przykładzie powyżej, oznacza niedzielę w każdym tygodniu w roku. Więcej przykładów można znaleźć w nowym podręczniku *DB2 V10 Managing Performance*.

Ostatnim ważnym do omówienia punktem są metody weryfikacji funkcjonowania rozwiązania. Zazwyczaj w pierwszej kolejności sprawdza się działanie procedury monitorującej statystyki ADMIN_UTL_MONITOR oraz procedury wykonawczej ADMIN_UTL_EXECUTE. Służy do tego funkcja DSNADM.ADMIN_TASK_STATUS(). Występują dwie wersje tej funkcji – bez parametrów i z parametrem MAX_HISTORY. Szczególnie polecam tę drugą, ponieważ pokazuje określoną przez parametr ilość ostatnich uruchomień. Podczas analizy zwracamy uwagę głównie na kolumny TASK_NAME, STATUS NUM_INVOCATIONS oraz START i END TIMESTAMP. Zadanie (TASK) ADMIN_UTL_MONITOR zdefiniowane wg przykładu z akapitu powyżej miało nazwę STATMON1, natomiast dodane automatyczne zadania mają nazwę DB2 AUTO PROCEDURE EXECUTE. Status poprawnie zakończonych zadań powinien być COMPLETED. Kolejnym ważnym miejscem dla weryfikacji poprawnego działania rozwiązania jest tablica SYSIBM.SYSAUTORUNS_HIST. Kolumny OUTPUT i ERROR_MESSAGE zawierają wartościowe informacje przy diagnozowaniu problemów z procedurami.

Jak już wspomniano, wykryte przez ADMIN_UTL_MONITOR rejestrowane są w tablicy SYSIBM.SYSAUTOALERTS. Warto upewnić się, czy nie zostają w niej nieobsłużone obiekty.

W tablicy tej (kolumna OUTPUT) znajdziemy też raporty z zadań RUNSTATS. Korzystanie z nich będzie jednak wymagać napisania prostej procedury w SQL-PL, bo SPUFI niezbyt dobrze radzi sobie z kolumnami CLOB 2M.

Warto dodać do ADMT procedurę ADMIN_UTL_MODIFY, służącą do usuwania wpisów z obydwu wymienionych

wyżej tablic. Procedurę należy zdefiniować z parametrem *history-days=nn*, gdzie nn to wymagana przez nas długość przechowywania informacji w tych tablicach. Oczywiście można uruchamiać ADMIN_UTL_MODIFY również ręcznie (np. wykorzystując IBM Data Studio), ale jest to wewnętrznie sprzeczne z ideą automatyzacji.

Ostatnim mechanizmem diagnostycznym, o którym warto wspomnieć, jest możliwość włączania śledzenia (TRACE) na poziomie ADMT. Wykonujemy to komendami jak w ramce poniżej.

```
/* start ADMT */
S DOA1ADMT
/* stop ADMT */
F DOA1ADMT,appl=shutdown
/* TRACE ON */
F DOA1ADMT,appl=trace-on
/* TRACE OFF */
F DOA1ADMT,appl=trace=off
```

Należy również pamiętać o logach STARTED TASKÓW uruchamianych przez WLM ENVIRONMENTS, w których wykonywane są procedury składowane wchodzące w skład rozwiązania.

PODSUMOWANIE

Rozwiązania do automatycznego zbierania statystyk dostarczane w ramach DB2 dla z/OS V10, wydaje się dobrym kierunkiem, w którym powinien podążać producent. Dodatkowo profile RUNSTATS mogą być łatwo wykorzystane przez każde rozwiązanie do zbierania statystyk, nie tylko oparte o ADMT. Uruchomienie samego rozwiązania i operowanie nim wydaje się dosyć kłopotliwe. Pracę z funkcjami i procedurami znacząco ułatwia wykorzystanie IBM Data Studio. Jednak poważne produkcyjne rozwiązanie do automatyzacji zbierania statystyki obiektów DB2 oparte o ADMT, moim zdaniem, będzie wymagało napisania prostej aplikacji ukrywającej przed użytkownikami intrygujący świat procedur składowanych, funkcji UDF i tablic z katalogu DB2 opisany w tym artykule.

DB2 11 Optymalizator: wykrywanie brakujących i konfliktujących statystyk

Tekst: Michał Białecki, IBM

Najnowsza wersja bazy danych DB2 11 na platformę z/OS oferuje zarówno nowe rozwiązania, jak również dostarcza ulepszeń w zakresie dotychczasowych funkcjonalności. Zainteresowanych szczegółami odsyłam do prezentacji Johna Campbella dostępnych na stronie <http://bit.ly/DB211webcast>. W artykule tym omówię jedno z najbardziej przełomowych – w mojej ocenie – rozwiązanie dotyczące wykrywania brakujących i konfliktujących statystyk dostępne w nowej wersji.

Dokładne, kompletne i aktualne statystyki są elementem niezbędnym do podjęcia przez optymalizator (*optimizer*) decyzji o najmniej kosztownej ścieżce dostępu (*the lowest cost access path*). Niektóre statystyki łatwo zidentyfikować (np. te podstawowe dla tabeli i indeksów). Ale „im dalej w las, tym więcej drzew” – statystyki bardziej szczegółowe, np. *multi-column statistics*, *non-uniform distributed statistics*, trudno wybrać bez dogłębnej i czasochłonnej analizy, a zatem wyważenia, które z nich są kluczowe, a które zupełnie nieistotne dla wyboru ścieżki dostępu dla danego SQLa.

W środowiskach, gdzie wykonywany jest dynamiczny SQL, trudnym jest zebranie kompletnej charakterystyki przetwarzania (*workload*) na potrzeby takiej analizy. Dzieje się tak choćby z uwagi na charakter Dynamic Statement Cache, który zawiera tylko część wykonywanych zapytań (usuwanych zgodnie z algorytmem LRU). Ze względu na tę złożoność analizy, w wielu instalacjach DB2 kończy się na zebraniu tylko statystyk zbyt ogólnych, opartych na domyślnych ustawieniach statystyk, np. `RUNSTATS TABLESPACE DB.TS TABLE(ALL) SAMPLE 25 INDEX(ALL)`, lub przyjęciem podejścia zgoła odwrotnego - zbieraniem statystyk zbyt szczegółowych, bez wglądu w zapytania (ponieważ zapytań jest zbyt dużo i są zbyt skomplikowane, nie wiadomo kiedy dane ulegają istotnym zmianom, itp.).

Zużycie procesora (CPU) jest podstawowym wyznacznikiem „kosztu” pracy w środowisku mainframe. Nieopty-

malne ścieżki dostępu dla zapytań w DB2 generują zwiększony koszt CPU oraz I/O. Z drugiej strony także zbieranie nadmiarowych statystyk generuje niepotrzebne i niemałe koszty zadań serwisowych (mimo tego że stałym trendem w coraz większym stopniu jest przekierowanie - *offload* - zadań serwisowych bazy danych na procesory typu zIIP).

Podczas procesu wyboru ścieżki dostępu w DB2 11 (BIND lub PREPARE), optymalizator rozpoznaje, kiedy zostają zastosowane wartości domyślne statystyk, jak i przybliżenia, a następnie uzupełnia statystyki, które nie zostały zebrane lub konfliktują ze sobą. Poprzez zapis rekomendacji do tablic *DB2 catalog* lub *explain*, nowo wprowadzona funkcjonalność w DB2 dostarcza administratorowi bazy danych informacje do optymalnego wyboru zestawu statystyk (zadań `RUNSTATS`) dopasowanych do danego przetwarzania i do wykonywanych zapytań. Wszystko odbywa się niejako „przy okazji” normalnej pracy optymalizatora bazy danych i jest mechanizmem wbudowanym.

Podobną funkcjonalność dla pojedynczych zapytań w DB2 9 i 10 można było uzyskać za pomocą:

- bezpłatnego narzędzia *Query Tuner: Statistics Advisor*, stosowanego zwykle post factum, po tym jak wystąpił problem z nieoptymalną ścieżką dostępu;

- *Autonomic statistics*, umożliwiającego automatyczne zbieranie statystyk. Jest to mechanizm oparty o ADMT (ADMIN SCHEDULER) i procedury składowane (*stored procedures*) - w opiniach użytkowników nie do końca wygodny w implementacji i użytkowaniu.

Od wersji 11 podejście producenta do wykrywania brakujących i konfliktujących statystyk jest zdecydowanie bardziej proaktywne i zaimplementowane w samym „motorze” bazy danych.

W DB2 11 wprowadzono:

- nową tablicę DB2 *catalog*: `SYSIBM.SYSSTATFEEDBACK` wypełnianą asynchronicznie rekomendacjami w odstęпах określonych przez ZPARAM

`STATSINT`,

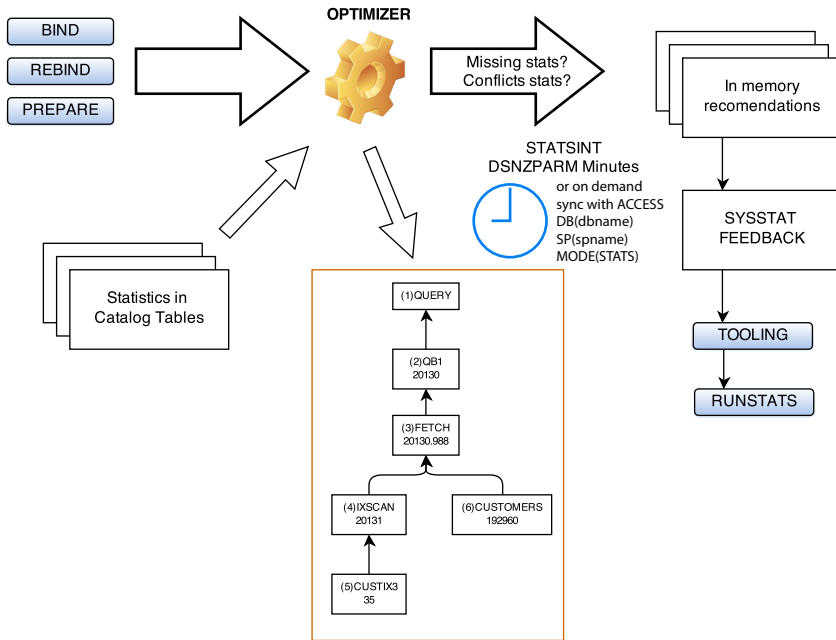
- nowy ZPARAM `STATFDBK_SCOPE` (NONE, STATIC, DYNAMIC, ALL) kontrolujący zakres wydawanych rekomendacji w `SYSIBM.SYSSTATFEEDBACK`, dla statycznego, dynamicznego SQLa, (domyślnie zbierane są rekomendacje dla obu typów),

- nową tablicę typu `explain`: `DSN_STAT_FEEDBACK` wypełnianą synchronicznie podczas `EXPLAIN`, `BIND` / `REBIND` z opcją `EXPLAIN(YES|ONLY)`,

- nową komendę `-ACCESS DATABASE (DB2)MODE(STATS)` powodującą synchroniczne wypełnienie `SYSSTATFEEDBACK` na żądanie administratora (też dla *RealTimeStats RTS*),

- nową kolumnę z możliwością *update* - `STATS_FEEDBACK` w `SYSIBM.SYSTABLES`, która steruje, czy rekomendacje są zapisywane dla konkretnej tabeli, czy też nie.

Wykrywanie brakujących i konfliktujących statystyk jest dostępne od DB2 11 New Function Mode (NFM), a dla `EXPLAIN`, także w Conversion Mode (CM) jeśli tylko tablica `DSN_STAT_FEEDBACK` została utworzona. Rysunek 1 strona 14 przedstawia schemat działania tej funkcjonalności na przykładzie `BIND` (rekomendacje zapisywane asynchronicznie w `SYSSTATFEEDBACK`). Rekomendacje dla tabel, indeksów, czy też kolumn zapisane są w tabelach `SYSSTATFEEDBACK` lub `DSN_STAT_FEEDBACK` i mogą zostać następnie użyte przez administratora bazy danych w celu konstrukcji lub automatyzacji odpowiednich zadań `RUNSTATS`, jak również mogą zostać wykorzystane przez oprogramowanie zewnętrzne (np. IBM Optim Query Tuner, czy innych dostawców oprogramowania dla DB2 z/OS). Analogicznie, lecz już synchronicznie, odbywa się to dla `EXPLAIN` (zapis natychmiastowy do tabeli `explain DSN_STAT_FEEDBACK`).



Rysunek 1. Schemat wykrywania konfliktujących statystyk na przykładzie BIND.

Poniżej definicje nowych tablic na przykładzie SYSIBM.SYSSTATFEEDBACK (dla DSN_STAT_FEEDBACK dochodzą kolumny występujące dla tablic typu explain, np. queryno, applname, itp.). I tak w kolumnie TYPE zapisany jest typ rekomendowanych statystyk, a w REASON uzasadnienie wydanej rekomendacji:

Otrzymany wynik w kolumnie REASON należy interpretować według następującego klucza:

- REASON='BASIC' oznacza brak statystyk dla danego obiektu;
- REASON='CONFLICT' oznacza, że wykryto statystyki, których wartości są niezgodne. Przykładem konfliktu jest brak synchronizacji danych

ze statystykami (*mis-timed statistics*), czyli sytuacja, w której np. dla jednego indeksu statystyki zostały zebrane w dniu A, potem dane istotnie się zmieniły, a statystyki dla tabeli zostały zebrane po zmianie danych w dniu B - w rezultacie powstają różne statystyki liczności rekordów. Z punktu widzenia optymalizatora to konfliktujące i niezrzetelne dane, które w procesie wyboru ścieżki zostają zignorowane;

- REASON='LOWCARD', 'NULLABLE', 'DEFAULT' oznacza, że nie ma statystyk typu *frequency* dla kolumn, które prawdopodobnie zawierają np. dane z niską liczebnością (*cardinality*), czy z wartościami NULL (*skewed data*).

Poniżej w Tabeli 2 znajdują się instrukcja wg dokumentacji DB2 11, dotycząca interpretacji rekomendacji TYPE i ich przełożenia na składnie dla RUNSTATS utility.

Po wykonaniu wymaganych RUNSTATS, rekomendacje są usuwane z tablicy SYSSTATFEEDBACK.

Cały mechanizm wydaje się prosty intuicyjny. Ta nowa funkcjonalność w DB 11 daje dość poręczny i „tani” sposób na zbieranie odpowiednich statystyk. Przekłada się ona na obniżenie kosztu użytkowania i utrzymywania bazy danych - TCO (*Total cost of ownership*) - głównie poprzez zmniejszenia nakładu pracy administratorów odpowiedzialnych za obsługę DB2 i wykonywane analizy, ale również poprzez zmniejszenie kosztu działania DB2 dzięki optymalizacji ścieżek dostępu i wykonywania jedynie niezbędnych zadań RUNSTATS.

Tabela 1. Definicja nowej tablicy catalog SYSSTATFEEDBACK w DB2 11.

Column name	Data type	Description
TBCREATOR	VARCHAR(128)	The creator of the table.
TBNAME	VARCHAR(128)	The name of the table.
IXCREATOR	VARCHAR(128)	The creator of the index.
IXNAME	VARCHAR(128)	The name of the index.
COLNAME	VARCHAR(128)	The name of the column.
NUMCOLUMNS	SMALLINT	The number of columns in the column group.
COLGROUPCOLNO	VARCHAR(254) FOR BIT DATA	A hex representation that identifies the set of columns associated with the statistics. If the statistics are only associated with a single column, the field contains a zero length. Otherwise, the field is an array of SMALLINT column numbers with a dimension equal to the value in NUMCOLUMNS.
TYPE	CHAR(1)	The type of statistic to collect: 'C' Cardinality. 'F' Frequency. 'H' Histogram. 'I' Index. 'T' Table.
DBNAME	VARCHAR(24)	The name of the database.
TSNAME	VARCHAR(24)	The name of the table space.
REASON	CHAR(8)	The reason that the statistic was recommend: 'BASIC' A basic statistical value for a column table or index is missing. 'KEYCARD' The cardinalities of index key columns are missing. 'LOWCARD' The cardinality of the column is a low value, which indicates that data skew is likely. 'NULLABLE' Distribution statistics are not available for a nullable column. 'DEFAULT' A predicate references a value that is probably a default value. 'RANGEPD' Histogram statistics are not available for a range predicate. 'PARALLEL' Parallelism could be improved by uniform partitioning of key ranges. 'CONFLICT' Another statistic conflicts with this statistic. 'COMPFIX' Multi-column cardinality statistics are needed for an index compound filter factor.
BLOCK_RUNSTATS	CHAR(1)	Whether the row is used when optimization tools collect statistics based on the recommendations. DB2 inserts a blank value in this column for all new rows. DB2 does not refer to or change the value of this column. This is an updatable column.
REMARKS	VARCHAR(254)	Free form text for extensibility.
LASTDATE	DATE	The last date that this statistics recommendation was updated by DB2.

Tabela 2. Opcje dla RUNSTATS wynikające z rekomendacji w tabeli SYSSTATFEEDBACK / DSN_STAT_FEEDBACK.

TYPE	Statistics to collect	RUNSTATS utility options to use
'T'	Basic table statistics	Use the TABLE option: RUNSTATS TABLESPACE ... TABLE(table-name)
'I'	Basic index statistics	Use the INDEX option: RUNSTATS INDEX
'C'	Cardinality statistics	For a single column cardinality, use the COLUMN option: RUNSTATS TABLESPACE ... TABLE(table-name) COLUMN(column-name) For a multi-column cardinality, use the COLGROUP option: RUNSTATS TABLESPACE ... TABLE(table-name) COLGROUP(column-name1,column-name2)
'F'	Frequency statistics	Use the FREQUAL option: RUNSTATS. For example, RUNSTATS TABLESPACE ... TABLE(table-name) COLGROUP(column-name) FREQUAL COUNT integer
'H'	Histogram statistics	Use the HISTOGRAM option: RUNSTATS TABLESPACE ... TABLE(table-name) COLGROUP(column-name) HISTOGRAM



Michał Białycki - Pracuje w IBM Lab jako DB2 Level 2 support & development. W ramach działalności na rzecz stowarzyszenia Polska Grupa Użytkowników DB2 pomaga w pozyskaniu prelegentów i sponsorów. Wzmacnia kontakty członków grupy z ekspertami z IBM Lab. Chętnie występuje z kolegami z PDUG w roli prelegenta na konferencjach IDUG. W wolnych chwilach uprawia wake, kite, snowboard i splitboard.

Jacek Rafalak - Od 16 lat pracuje w Asseco Poland S.A. jako specjalista DB2 dla z/OS. Znamy go jako Współzałożyciela, aktywnego wolontariusza, pierwszego prezesa stowarzyszenia "Polska Grupa Użytkowników DB2" oraz prelegenta na konferencjach technicznych DB2 z/OS (IDUG 2012, PDUG 2014). Jest faszynatem technologii Mainframe i orędownikiem DB2 z/OS. Za swój wkład w propagowanie wiedzy na jej temat został wyróżniony tytułem IBM Champion for Information Management na rok 2013. Poza pracą fascynuje się historią Polski. Żywo kibicuje drużynom siatkarskim.



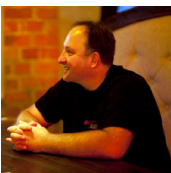
Emil Tarabasz - Pracuje w Zakładzie Ubezpieczeń Społecznych jako Administrator DB2 dla z/OS od 7 lat. Jest członkiem Zarządu stowarzyszenia "Polska Grupa Użytkowników DB2". W wolnej chwili trenuje boks. Interesuje się motoryzacją.



Paweł Sękowski - Pracuje w Asseco Poland S.A. Jako Starszy specjalista zajmuje się testami na platformie przedprodukcyjnej w projekcie KSI ZUS. Jest współzałożycielem, aktywnym wolontariuszem i PDUG blogerem. Wspiera prace zarządu PDUG w zakresie organizacji eventów. Chętnie prowadzi dokumentację fotograficzną przebiegu naszych konferencji. W wolnej chwili lubi jazdę na rowerze.



Roman Głodowski - Od 8 lat pracuje w PKO Banku Polskim jako systemowy administrator DB2, zajmuje się wszystkimi aspektami związanymi z utrzymaniem baz danych, w tym technologią i narzędziami replikacji danych. PDUG wspiera będąc członkiem zarządu oraz prowadząc nasze forum. Ze względu na brak wolnych chwil hobby odłożył na emeryturę.



Krzysztof Smolarek - Pracuje w Zakładzie Ubezpieczeń Społecznych od 15 lat, czyli od początku projektu KSI, zaś jako Administrator DB2 dla z/OS od 12 lat. W stowarzyszeniu "Polska Grupa Użytkowników DB2" jeste przewodniczącym Komisji Rewizyjnej. Prywatnie uwielbia górskie wycieczki i sporty motorowe.



Marek Górka - Pracuje w Ministerstwie Sprawiedliwości. Jego przygoda z platformą Mainframe zaczęła się niespełna 5 lat temu, a od dwóch lat pełni stanowisko Administratora DB2. Interesuje się nowymi technologiami. W stowarzyszeniu Polska Grupa Użytkowników DB2 pełni funkcję członka Komisji Rewizyjnej. Wolny czas spędza z książką w dłoni.



Małgorzata Delis - Od ponad 4 lat związana zawodowo z Asseco Poland S.A. W PDUG chętnie pomaga m.in. przy organizacji i obsłudze konferencji, przygotowaniu prezentacji i prelegerów, również na konferencji międzynarodowej (IDUG), a także wspiera kolegów przy pisaniu tekstów i tłumaczeniu ich na język angielski. Baterie ładuje tańcząc w rytm muzyki latynoskiej, ucząc się włoskiego i podróżując.



Przemek Stelmaszewski - Od momentu powstania stowarzyszenia Polska Grupa Użytkowników DB2 aktywnie uczestniczy w jego działaniach. Jest jego współzałożycielem. Entuzjasta technologii i informatyzacji, zawodowo związany z platformą Mainframe oraz rozwiązaniami opartymi o scheduler Tivoli. Swoje zainteresowania rozwija również tworząc rozwiązania webowe wspierające działalność stowarzyszenia oraz kreuje jego spójny graficzny wizerunek w internecie oraz materiałach do druku. Pracuje w Centrali Zakładu Ubezpieczeń Społecznych.



Jarosław Pawelec - Pochodzi ze Szczecina, miasta portowego, z którym czuje się wciąż bardzo związany. Swoją przygodę z DB2 rozpoczął wraz z przeprowadzką do stolicy. Pracuje w Asseco Poland S.A., na stanowisku Starszy specjalista ds. technologii, przy realizacji projektu KSI ZUS. W stowarzyszeniu PDUG jest współzałożycielem, aktywnym wolontariuszem i wsparciem przy organizacji eventów. Oprócz DB2 jego pasje to astronomia, odkrywanie ciekawych zjawisk astronomicznych i ich fotografowanie. Uwielbia podróżować po świecie - do jego ostatnich udanych wypraw należą: Machu Picchu (Peru), Irkuck (Syberia, Rosja), Wielki Mur Chiński (Chiny).



Paweł Hryb - Pracuje na stanowisku Administratora baz danych w PKO Banku Polskim. Jego doświadczenia z DB2 dla z/OS to 16 lat wykonywania zadań związanych z instalacją, serwisowaniem, migracją oraz wszelkimi aspektami administrowania baz i strojenia aplikacji. W stowarzyszeniu Polska Grupa Użytkowników DB2 pełni funkcję członka Komisji Rewizyjnej.

Tech Events 2014

**24
MAR** DB2 Update Day 2014, DNB, Oslo, Norway
<http://www.eventbrite.co.uk/e/db2-update-day-2014-oslo-tickets-10015513671>, free entrance

**26
MAR** DB2 Update Day 2014, Sweden HSB Stockholm
free entrance

**27
MAR** DB2 Update Day 2014, Silkeborg, Denmark
<http://bit.ly/1jnvk6X>, free entrance

**9-14
NOV**

IDUG DB2 Tech Conference
Prague, Czech Republic
<http://www.idug.org/p/cm/ld/fid=376>

**27
MAR** BeLux DB2 Working Group, NRB Herstal, Belgium
<http://www.gsebelux.com>

**28
MAR** DB2 Update Day 2014, Kobenhavn, Ballerup
www.eventbrite.co.uk/e/db2-update-day-2014-kbenhavn-tickets-10015617983, free entrance

**09
APR** SQLAdria, Ljubljana, Slovenia
<http://www.sqladria.net>

**16
APR** IDUG DB2 Seminar, London IBM South Bank
<http://www.idug.org/p/cm/ld/fid=462> , free entrance

**09
MAY** German DB2 LUW User Group
http://www.ars.de/web/contact/subscription/default?eventId=deDUG03_140509

**25-28
MAY** SQLAdria, Dubrovnik, Croatia
<http://www.sqladria.net>

**26
SEP** SQLAadria, Zagreb, Croatia
<http://www.sqladria.net>

**06
NOV** csDUG conference, Prague, Czech Republic
<http://db2forz.blogspot.co.uk/>

**27-28
NOV** SQLAdria, Opatia, Croatia
<http://www.sqladria.net>

PDUG TECH EVENT
SEPTEMBER, POLAND, WARSAW
www.pdug.pl