

WPROWADZENIE DO

# DB2 Express-C

Książka dla Społeczności od Społeczności

RAUL CHONG, IAN HAKES, RAV AHUJA

PRZEDMOWA DR. ARVIND KRISHNA



WYDANIE DRUGIE



# **Wprowadzenie do DB2 Express-C**

**Książka dla Społeczności od Społeczności**

**Wydanie drugie (kwiecień 2008)**

**Wydanie to zostało poprawione dla wersji IBM<sup>®</sup> DB2<sup>®</sup> Express-C Version 9.5 dla Linux<sup>®</sup>, UNIX<sup>®</sup> i Windows<sup>®</sup>.**

**© Copyright IBM Corporation, 2007, 2008. Wszelkie prawa zastrzeżone.**

---

# Spis treści

<b>O książce.....</b>	<b>6</b>
Uwarunkowania prawne.....	6
Dla kogo jest tak książka? .....	7
Jak jest zorganizowana ta książka? .....	7
Książka dla Społeczności .....	8
Autorzy i osoby współpracujące .....	9
Podziękowania .....	10
Przedmowa .....	11
<b>CZĘŚĆ I – PRZEGLĄD DB2, INSTALACJA.....</b>	<b>12</b>
<b>Rozdział 1 – Czym jest DB2 Express-C? .....</b>	<b>13</b>
1.1 Bezpłatne rozwijanie, wdrażanie i dystrybucja... bez ograniczeń! .....	14
1.2 Pomoc użytkownika i wsparcie techniczne .....	14
1.3 Serwery DB2 .....	15
1.4 Klient DB2 i sterowniki .....	15
1.5 Swoboda w rozwijaniu aplikacji .....	17
1.6 Wersje DB2, a edycje DB2 .....	17
1.7 Migracja do innej edycji DB2 .....	18
1.8 Utrzymanie DB2 Express-C .....	18
1.9 Pokrewne bezpłatne oprogramowanie .....	18
1.9.1 IBM Data Studio .....	19
1.9.2 DB2 Net Search Extender .....	19
1.9.3 Starter Toolkit dla DB2 on Rails .....	19
1.9.4 Web 2.0 Starter Toolkit dla DB2 .....	19
1.9.5 WebSphere Application Server – Community Edition .....	20
<b>Rozdział 2 – Pokrewne funkcjonalności i produkty .....</b>	<b>21</b>
2.1 Funkcjonalności dostępne z subskrypcją DB2 Express-C .....	25
2.1.1 Pakiety poprawek .....	25
2.1.2 High Availability Disaster Recovery (HADR) .....	25
2.1.3 Replikacja danych.....	26
2.2 Funkcjonalności niedostępne z DB2 Express-C.....	27
2.2.1 Partycjonowanie bazy danych .....	27
2.2.2 Koncentrator Połączeń .....	27
2.2.3 Geodetic Extender .....	27
2.2.4 Workload Management (WLM).....	27
2.3 Płatne produkty powiązane z DB2.....	28
2.3.1 DB2 Connect .....	28
2.3.2 WebSphere Federation Server .....	28
2.3.3 WebSphere Replication Server .....	29
<b>Rozdział 3 – Instalacja DB2.....</b>	<b>31</b>
3.1 Wymagania przedinstalacyjne .....	31
3.2 Instalacja a uprawnienia .....	31
3.3 Kreator instalacji .....	32

---

3.4 Instalacja automatyczna .....	36
Laboratorium #1: Instalacja DB2 i tworzenie przykładowej bazy .....	38
<b>Rozdział 4 – Środowisko DB2.....</b>	<b>41</b>
Laboratorium #2 – Tworzenie nowej bazy danych .....	50
4.1 Konfiguracja DB2 .....	51
4.1.1 Zmienne środowiskowe .....	52
4.1.2 Plik konfiguracyjny menedżera bazy danych (dbm cfg) .....	52
4.1.3 Plik konfiguracyjny bazy danych (db cfg) .....	54
4.1.4 Rejestr profilu DB2.....	55
4.2 Serwer Administracyjny DB2 .....	56
Laboratorium #3 – Praca z instancją, bazą danych i konfiguracja .....	58
<b>Rozdział 5 – Narzędzia DB2 .....</b>	<b>61</b>
5.1 Centrum sterowania.....	63
5.2 Edytor komend.....	66
5.3 Asysta SQL .....	67
5.4 Przycisk Pokaż SQL .....	69
Laboratorium #4: Wypełnianie bazy danych przy użyciu skryptów .....	71
5.5 Skryptowanie .....	73
5.5.1 Skrypty SQL.....	73
5.5.2 Skrypty (powłoki) systemu operacyjnego .....	74
Laboratorium #5: Tworzenie skryptu do instalacji bazy danych .....	76
5.6 Centrum zadań .....	79
5.6.1 Katalog narzędzi bazy danych.....	79
5.7 Kronika.....	81
5.8 Monitor poprawności.....	82
5.8.1 Centrum kontroli poprawności .....	83
<b>CZĘŚĆ II – NAUKA DB2: ADMINISTRACJA BAZĄ DANYCH.....</b>	<b>85</b>
<b>Rozdział 6 – Architektura DB2 .....</b>	<b>87</b>
6.1 Model procesu w DB2.....	87
6.2 Model pamięci w DB2 .....	90
6.3 Model przechowywania danych.....	91
6.3.1 Strony i ekstenty .....	91
6.3.2 Pule buforów .....	91
6.3.3 Przestrzenie tablicowe.....	93
<b>Rozdział 7 – Klient DB2 .....</b>	<b>99</b>
7.1 Katalogi DB2 .....	99
7.2 Asysta podczas konfigurowania .....	100
7.2.1 Konfiguracja serwera .....	101
7.2.2 Konfiguracja klienta .....	103
7.2.3 Profile klienta i serwera.....	107
Laboratorium #6: Użycie narzędzia Asysta podczas konfigurowania .....	110
<b>Rozdział 8 – Praca z obiektami bazy danych .....</b>	<b>113</b>
8.1 Schematy .....	113

---

8.2 Tabele .....	114
8.2.1 Typy danych .....	114
8.2.2 Kolumna identyfikująca .....	117
8.2.3 Sekwencje .....	117
8.2.4 Tabele słownika systemowego .....	118
8.2.5 Tabele tymczasowe .....	119
Laboratorium #7: Tworzenie nowej tabeli .....	120
8.3 Widoki .....	123
8.4 Indeksy .....	123
8.4.1 Doradca projektowania .....	123
8.5 Więzy integralności .....	125
<b>Rozdział 9 – Narzędzia do przenoszenia danych .....</b>	<b>127</b>
9.1 Narzędzie EXPORT .....	128
9.2 Narzędzie IMPORT .....	129
9.3 LOAD .....	130
9.4 Narzędzie db2move .....	131
9.5 Narzędzie db2look .....	131
Laboratorium #8 – Tworzenie pliku DDL dla bazy danych .....	134
<b>Rozdział 10 – Bezpieczeństwo bazy danych .....</b>	<b>139</b>
10.1 Uwierzytelnianie .....	140
10.2 Autoryzacja .....	141
10.3 Uprawnienie DBADM .....	144
10.4 Grupa PUBLIC .....	145
10.5 Instrukcje GRANT i REVOKE .....	145
10.6 Sprawdzanie uprawnień i przywilejów .....	146
10.7 Rozważania o grupowaniu przywilejów .....	147
Laboratorium #9 – Nadawanie i odbieranie uprawnień .....	148
<b>Rozdział 11 – Kopie zapasowe i odzyskiwanie danych .....</b>	<b>151</b>
11.1 Logowanie w bazie danych .....	151
11.2 Typy dzienników .....	152
11.3 Tryby zapisów do dziennika .....	153
11.3.1 Zapis cykliczny .....	153
11.3.2 Zapis archiwalny .....	153
11.4 Logowanie w bazie danych z poziomu Centrum sterowania .....	154
11.5 Parametry zapisu .....	156
11.6 Kopia zapasowa bazy danych .....	156
Laboratorium #10 – Harmonogramowanie kopii zapasowej .....	159
11.7 Odzyskiwanie bazy danych .....	162
11.7.1 Tryby odzyskiwania danych .....	162
11.7.2 Przywrócenie bazy danych .....	162
11.8 Inne operacje z BACKUP I RESTORE .....	163
<b>Rozdział 12 – Zadania konserwacyjne .....</b>	<b>165</b>
12.1 REORG, RUNSTATS, REBIND .....	165
12.1.1 Polecenie REORG .....	166
12.1.2 Polecenie RUNSTATS .....	166

---

12.1.3 BIND / REBIND.....	167
12.1.4 Konserwacja bazy z poziomu Centrum sterowania.....	168
12.2 Sposoby konserwacji.....	169
Laboratorium #11 – Konfigurowanie automatycznej konserwacji .....	172
<b>Rozdział 13 – Współbieżność i blokowanie .....</b>	<b>175</b>
13.1 Transakcje .....	175
13.2 Współbieżność.....	176
13.3 Problemy pojawiające się przy braku kontroli współbieżności ...	177
13.3.1 Utracona modyfikacja .....	177
13.3.2 Odczyt niezatwierdzonych danych .....	178
13.3.3 Odczyt niepowtarzalny.....	179
13.3.4 Odczyt fantomów .....	180
13.4 Poziomy izolacji .....	181
13.4.1 Odczyt niezatwierdzonych danych .....	181
13.4.2 Stabilność kursora .....	181
13.4.3 Stabilność odczytu.....	181
13.4.4 Odczyt powtarzalny .....	182
13.4.5 Porównanie poziomów izolacji.....	182
13.4.6 Ustawianie poziomu izolacji.....	183
13.5 Eskalacja blokad .....	184
13.6 Śledzenie blokad.....	185
13.7 Oczekiwanie na blokadę .....	186
13.8 Przyczyny i wykrywanie zakleszczeń .....	187
13.9 Najlepsze praktyki we współbieżności i blokowaniu.....	188
<b>CZĘŚĆ III – NAUKA DB2: PROGRAMOWANIE APLIKACJI .....</b>	<b>191</b>
<b>Rozdział 14 – Procedury składowane w SQL PL .....</b>	<b>193</b>
14.1 IBM Data Studio.....	194
14.1.2 Tworzenie procedury składowanej w Data Studio.....	195
14.2 Podstawy procedur składowanych w SQL PL .....	199
14.2.1 Struktura procedury składowanej .....	199
14.2.2 Dodatkowe atrybuty procedury składowanej.....	200
14.2.3 Parametry .....	200
14.2.4 Komentarze w procedurach składowanych w SQL PL.....	201
14.2.5 Instrukcje złożone .....	201
14.2.6 Deklaracje zmiennej .....	202
14.2.7 Instrukcje przypisania .....	202
14.3 Kursory.....	203
14.4 Sterowanie przepływem.....	203
14.5 Wywoływanie procedur składowanych .....	204
14.6 Błędy i przechwytywanie warunków .....	206
14.7 Dynamiczny SQL .....	207
<b>Rozdział 15 – Inline SQL PL, wyzwalacze i funkcje użytkownika ....</b>	<b>209</b>
15.1 Inline SQL PL.....	209
15.2 Wyzwalacze .....	210
15.2.1 Rodzaje wyzwalaczy.....	211



---

Laboratorium #12 – Tworzenie wyzwalacza w Centrum sterowania..	214
15.3 Funkcje użytkownika.....	217
15.3.1 Funkcje skalarne.....	217
15.3.2 Funkcje tabelaryczne.....	218
Laboratorium #13 – Tworzenie UDF w IBM Data Studio.....	220
<b>Rozdział 16 – SQL/XML i XQuery.....</b>	<b>223</b>
16.1 Wykorzystanie XML w bazie danych.....	224
16.2 Bazy danych XML.....	224
16.2.1 XML-owe bazy danych.....	224
16.2.2 Natywne bazy danych XML.....	225
16.3 XML w DB2.....	225
16.3.1 Zalety technologii pureXML.....	227
16.3.2 Podstawy XPath.....	229
16.3.3 Definicja XQuery.....	232
16.3.4 Dodawanie dokumentów XML.....	233
16.3.5 Zapytania na danych XML.....	236
16.3.6 Złączenia w SQL/XML.....	241
16.3.7 Złączenia w XQuery.....	242
16.3.8 Operacje aktualizacji i usuwania.....	243
16.3.9 Indeksowanie XML.....	245
Laboratorium #14 - SQL/XML i XQuery.....	247
<b>Rozdział 17 – Programowanie aplikacji Java, PHP i Ruby.....</b>	<b>249</b>
17.1 Programowanie aplikacji w Javie.....	249
17.1.1 Sterownik JDBC typu 2.....	249
17.1.2 Sterownik JDBC typu 4.....	250
17.2 Programowanie aplikacji w PHP.....	251
17.2.1 Opcje połączenia DB2 dla PHP.....	251
17.2.2 Zend Core dla IBM.....	253
17.3 Rozwój aplikacji w Ruby on Rails.....	255
17.3.1 Uruchomieniowy zestaw narzędzi dla DB2 on Rails.....	255
<b>Dodatek A – Rozwiązywanie problemów.....</b>	<b>257</b>
A.1 Uzyskiwanie informacji o kodach błędów.....	258
A.2 SQLCODE i SQLSTATE.....	259
A.3 Dziennik powiadomień.....	259
A.4 db2diag.log.....	259
A.5 Śledzenie CLI.....	260
A.6 DB2 – błędy i poprawki.....	260
<b>Źródła informacji o DB2.....</b>	<b>261</b>
Strony WWW.....	261
Książki.....	262

---

## O książce

### Uwarunkowania prawne

© Copyright IBM Corporation 2007, 2008  
All Rights Reserved.  
IBM Canada  
8200 Warden Avenue  
Markham, ON  
L6G 1C7  
Canada

Wykonywanie kopii czy odtwarzanie niniejszego dokumentu i jakiegokolwiek jego części w dowolnej formie i przy użyciu jakichkolwiek środków oraz tłumaczenie go na inne języki jest zabronione bez wcześniejszej zgody właścicieli praw autorskich wymienionych powyżej.

IBM nie udziela jakichkolwiek gwarancji (w tym także rękojmi) ani nie ustala warunków, a domniemanych szczególności nie udziela domniemanych gwarancji przydatności handlowej i przydatności do określonego celu, w odniesieniu do treści zawartych w niniejszym dokumencie. IBM nie ponosi odpowiedzialności za jakiegokolwiek błędy, które mogą pojawić się w niniejszym dokumencie, a w szczególności błędy w tłumaczeniu. Informacje tu zawarte podlegają zmianom bez wcześniejszego powiadomienia. IBM zastrzega sobie prawo do wprowadzenia takich zmian bez obowiązku powiadamiania o nich. IBM nie zobowiązuje się do aktualizowania informacji zawartych w niniejszym dokumencie.

Informacje dotyczące produktów innych niż produkty IBM pochodzą od dostawców tych produktów. IBM nie testował tych produktów i nie może potwierdzić dokładności pomiarów wydajności, kompatybilności ani żadnych innych danych związanych z tymi produktami. Pytania dotyczące możliwości produktów innych niż produkty IBM należy kierować do dostawców tych produktów.

Wszelkie wzmianki w niniejszym dokumencie na temat produktów lub usług IBM nie oznaczają, że IBM zamierza udostępniać takie produkty bądź usługi we wszystkich krajach, w których prowadzi działalność.

Java oraz pozostałe znaki związane z Java są znakami towarowymi Sun Microsystems, Inc. w Stanach Zjednoczonych i/lub innych krajach.

Microsoft i Windows są znakami towarowymi Microsoft Corporation w Stanach Zjednoczonych i/lub innych krajach.

Linux jest zastrzeżonym znakiem towarowym Linus Torvalds w Stanach Zjednoczonych i/lub innych krajach.

UNIX jest zastrzeżonym znakiem towarowym The Open Group w Stanach Zjednoczonych i/lub innych krajach.

Nazwy innych firm, produktów lub usług mogą być zastrzeżonymi znakami towarowymi lub nazwami.

## Dla kogo jest ta książka?

Książka ta przeznaczona jest dla każdego, kto pracuje lub zamierza pracować z bazami danych – zarówno dla administratorów systemów bazodanowych, programistów, konsultantów, projektantów oprogramowania, zarządzających projektami, jak i wykładowców wykładowców studentów.

## Jak jest zorganizowana ta książka?

Część I: “Przegląd DB, Instalacja” wyjaśnia, czym jest DB2 Express-C, przedstawia produkty rodziny DB2 oraz pokrewne rozszerzenia i funkcjonalności; przeprowadza przez proces instalacji i tworzenia baz danych oraz omawia narzędzia udostępnione wraz z DB2.

Część II: “Nauka DB2: Administracja bazą danych” została zaprojektowana, aby zapoznać Cię ze środowiskiem DB2, architekturą, opcjami zdalnych połączeń, obiektami bazodanowymi, narzędziami do przenoszenia danych (import/export/load), bezpieczeństwem, tworzeniem kopii zapasowych oraz odzyskiwaniem danych, współbieżnością i blokadami, a także innymi popularnymi zadaniami konserwacyjnymi.

Część III: “Nauka DB2: Programowanie aplikacji” omawia procedury składowane, funkcje użytkownika, wyzwalacze, SQL/XML, XQuery, programowanie w Java™, PHP i Ruby.

Dodatek A zawiera przydatne informacje dotyczące rozwiązywania problemów.

Do większości z rozdziałów dołączone są ćwiczenia praktyczne zwane „Laboratorium”; pliki potrzebne do wykonania tych ćwiczeń znajdują się w dołączonym do tej książki archiwum `expressc_book_quicklabs_9.5.zip`. Plik ten dostępny jest także na stronie IBM® DB2 Express-C: [www.ibm.com/db2/express](http://www.ibm.com/db2/express).

Materiały zawarte w tej książce zostały także wykorzystane jako element programu „DB2 on Campus” i stanowią dokładne uzupełnienie e-learningowych prezentacji wideo udostępnionych na stronie [www.channelDB2.com/oncampus](http://www.channelDB2.com/oncampus). Książka ta może pomóc także w przygotowaniach do egzaminu “DB2 on Campus”. Egzamin ten zaświadcza o ukończeniu programu, potwierdzając udział w 16-godzinnym szkoleniu z DB2. Więcej informacji o programie “DB2 On Campus” znajduje się na stronie: [www.ibm.com/db2/express/students.html](http://www.ibm.com/db2/express/students.html).

**Ważne:**

Więcej informacji o programie “DB2 on Campus” uzyskać oglądając film: <http://www.channeldb2.com/video/video/show?id=807741:Video:3902>

## **Książka dla Społeczności**

Książka ta została napisana przez zespół DB2 Express-C i bezpłatnie udostępniona dla Społeczności DB2 Express-C. Liczni członkowie tej Społeczności z całego świata przetłumaczyli tę książkę na różne języki. Jeśli masz jakieś uwagi, chcesz dołączyć nowe treści, uzupełnić istniejące lub pomóc w tłumaczeniu tej książki na inne języki, wyślij e-mail o temacie "DB2 Express-C book changes." z opisem Twoich propozycji na adres [db2x@ca.ibm.com](mailto:db2x@ca.ibm.com).

## **Autorzy i osoby współpracujące**

Następujące osoby opracowały treści tej książki lub wniosły znaczący wkład w ich opracowanie:

**Raul F. Chong** – główny autor książki  
Raul jest kierownikiem programu „DB2 on Campus”, pracuje w Laboratorium IBM w Toronto.

**Ian Hakes** – współautor  
Ian pracuje w Laboratorium IBM w Toronto, zajmuje się wsparciem dla Społeczności DB2 Express-C

**Rav S. Ahuja** – współautor i wydawca  
Rav jest starszym menedżerem produktu (DB2) w Laboratorium IBM w Toronto.

Polska wersja tej książki powstała dzięki zaangażowaniu poniższych osób:

**Marcin Marczewski** – nadzór merytoryczny, przekład, redakcja  
IBM Polska, Laboratorium Oprogramowania w Krakowie

**Katarzyna Bodzioch** – redakcja, korekta językowa  
IBM BTO BCS Polska, Kraków

**Michał Andruszko** – przekład rozdziału 8, 10  
Uniwersytet im. A. Mickiewicza w Poznaniu

**Michał Białecki** – przekład rozdziału 11, 16  
Uniwersytet im. A. Mickiewicza w Poznaniu

**Łukasz Borek** – przekład rozdziału 3, 7, 9  
Uniwersytet im. A. Mickiewicza w Poznaniu

**Małgorzata Domańska** – przekład rozdziału 4, 5  
Uniwersytet im. A. Mickiewicza w Poznaniu

**Justyna Dorsz** – przekład rozdziału 1, 2, dodatek A  
Uniwersytet im. A. Mickiewicza w Poznaniu

**Paweł Rudak** – przekład rozdziału 14, 15  
Uniwersytet im. A. Mickiewicza w Poznaniu

**Tomasz Stańczak** – przekład rozdziału 13  
Uniwersytet im. A. Mickiewicza w Poznaniu

**Jarosław Tempowski** – przekład rozdziału 6, 12, 17  
Uniwersytet im. A. Mickiewicza w Poznaniu

## Podziękowania

Serdecznie dziękujemy następującym osobom za ich wsparcie i opracowanie materiałów, do których odwołuje się ta książka:

- Tedowi Wassermanowi, Clarze Liu i Paulowi Yi z Laboratorium IBM w Toronto, którzy opracowali materiały wykorzystane jako podstawy tej książki,
- Donowi Chamberlinowi i Cindy Saracco za ich artykuły dotyczące XQuery, opublikowane w ramach IBM developerWorks, oraz Matthiasowi Nicola za jego prezentacje o pureXML™,
- Kevinowi Czapowi oraz Grantowi Hutchisonowi za opracowanie materiałów informacyjnych o DB2,
- Katherine Boyachok za zaprojektowanie okładki tej książki,
- Susan Visser za jej wsparcie w publikacji tej książki.

## Przedmowa

Innowacja jest fundamentem postępów w technologii. W IBM innowacja od zawsze była integralną częścią ewolucji naszych serwerów danych. Jako pionierzy technik zarządzania danymi w latach 60 i 70, nieustannie dostarczamy innowacyjne technologie zarządzania informacjami, które znajdują odzwierciedlenie w tysiącach patentów z zakresu zarządzania danymi opracowanych przez technologów IBM. W wyniku tego, wiele największych światowych organizacji polega dzisiaj na produktach IBM takich jak DB2, wzmacniając w ten sposób swoje najbardziej wymagające i krytyczne rozwiązania zarządzania danymi. Jednakże DB2 nie jest już przeznaczone wyłącznie dla wielkich korporacji. Wraz z wypuszczeniem na rynek DB2 Express-C ta zdobywająca liczne nagrody technologia stała się także dostępna, aby zaspokajać potrzeby małych i średnich przedsiębiorstw – bez dodatkowych kosztów. Chociaż istnieje wiele serwerów bazodanowych - darmowych lub udostępnianych jako open source – DB2 Express-C oferuje wyjątkowe, w porównaniu z tymi alternatywami, korzyści.

W DB2 Express-C dostępnych jest wiele technologicznych innowacji. Część z tych innowacji zostało zaprojektowanych by dostarczać nowe, zaawansowane możliwości, część, aby zredukować brzemie administracyjne, część, aby poprawić wydajność, a część, aby zredukować koszty infrastruktury. Nie będziemy tutaj omawiać większości z nich mając nadzieję, że skusisz się i przeczytasz tę książkę – krótko jednak opiszemy jedną jako zachętę.

DB2 Express-C oparte jest na technologii "Viper", będąc pierwszym hybrydowym serwerem baz danych, zarządzającym danymi relacyjnymi i XML przechowywanymi w formacie natywnym. To sprawia, że DB2 jest idealne do tworzenia nowych typów aplikacji SOA i Web 2.0, gdzie wykorzystywane są duże ilości danych XML.

W przeciwieństwie do innych serwerów danych, udostępnianych przez inne firmy, DB2 Express-C nie ogranicza ani ilości danych, które chcesz przechowywać, ani ilości baz danych, które chcesz utworzyć. I co oczywiste – jeśli potrzebujesz wsparcia, bądź pomocy ze strony IBM, pomoc jest na jedno kliknięcie.

Książka ta służy jako przewodnik pomagający rozpocząć pracę i korzystać z DB2 Express-C. Pomoże Ci ona w zrozumieniu pojęć DB2, a także umożliwi rozwinięcie umiejętności administracyjnych oraz programistycznych. Umiejętności oraz wiedza uzyskana z pomocą tej książki będą także bardzo przydatne w pracy z innymi, zaawansowanymi edycjami DB2 dla Linuxa, UNIX-a i Windows.

Mimo, że DB2 Express-C nie jest produktem udostępnianym jako open-source, w IBM bardzo wspieramy i staramy się rozbudzać inicjatywy społecznościowe. Jestem zachwycony, że książka ta jest tworzona przez członków Społeczności DB2 Express-C i jest wszystkim bezpłatnie udostępniona. Bardzo zachęcam Cię do wzbogacania i aktualizacji tej książki o Twoją wiedzę, doświadczenia, a także do pomocy w tłumaczeniu tej książki na inne języki tak, aby inni także mogli czerpać z Twojej wiedzy.



Arvind Krishna –  
Wiceprezydent, Data Servers  
Information Management, IBM Software Group

# **CZEŚĆ I – PRZEGLĄD DB2, INSTALACJA**



# 1

## Rozdział 1 – Czym jest DB2 Express-C?

DB2 Express-C należy do IBM DB2, potężnej rodziny oprogramowania dla serwerów zarządzających zarówno danymi relacyjnymi, jak i XML. DB2 Express-C jest darmową, nieograniczoną i łatwą w użyciu wersją DB2. Znak 'C' w nazwie DB2 Express-C oznacza Społeczność (ang. Community) - Społeczność użytkowników DB2 Express-C, która powstała, aby móc pomagać sobie nawzajem w trybach on-line i off-line. Społeczność DB2 Express-C zrzesza różnych ludzi i firmy, które projektują, rozwijają, wdrażają lub korzystają z rozwiązań bazodanowych. Do Społeczności należą:

- twórcy aplikacji, którzy potrzebują otwartych standardów oprogramowania baz danych do budowy niezależnych, opartych na strukturze klient-serwer, internetowych, wspomagających zarządzanie przedsiębiorstwami aplikacji,
- dostawcy oprogramowania i sprzętu, dostawcy infrastruktury, oraz dostawcy innych rozwiązań, którzy chcą wykorzystać w pełni funkcjonalne serwery danych jako część swoich rozwiązań,
- konsultanci, administratorzy baz danych, architekci IT, którzy potrzebują mocnych serwerów baz danych do nauki, rozwijania umiejętności, oceniania i prototypowania,
- początkujące małe i średnie firmy, które potrzebują niezawodnych serwerów baz danych dla swoich aplikacji i operacji,
- amatorzy baz danych i entuzjaści nowoczesnych technologii, którzy potrzebują łatwych w użyciu serwerów danych do budowy aplikacji nowej generacji i Web 2.0,
- studenci, nauczyciele i inni akademicy użytkownicy, którzy potrzebują uniwersalnych serwerów baz danych do nauki, szkoleń, projektów i badań naukowych.

DB2 Express-C posiada te same podstawowe funkcjonalności co pozostałe, płatne wersje DB2 dla systemów Linux, UNIX i Windows. Serwer DB2 Express-C może zostać uruchomiony na 32-bitowych i 64-bitowych maszynach z zainstalowanym systemem operacyjnym Linux lub Windows. Może być uruchomiony na systemach z dowolną liczbą procesorów i wielkością pamięci i nie posiada żadnych specjalnych wymagań odnośnie miejsca na dysku lub konfiguracji systemu. DB2 Express-C dostarcza także technologię pureXML bez żadnych dodatkowych opłat. pureXML jest unikalną technologią DB2 służącą do natywnego przechowywania i przetwarzania dokumentów XML.

## 1.1 Bezpłatne rozwijanie, wdrażanie i dystrybucja... bez ograniczeń!

To zdanie podsumowuje kluczowe idee DB2 Express-C:

- **Bezpłatne rozwijanie aplikacji:** Jeśli jesteś twórcą programów i potrzebujesz bazy danych dla Twojej aplikacji, możesz użyć DB2 Express-C.
- **Bezpłatne wdrażanie:** Jeśli pracujesz w środowisku produkcyjnym i potrzebujesz systemu do zarządzania danymi, aby móc przechowywać ważne informacje, możesz użyć DB2 Express-C.
- **Bezpłatna dystrybucja:** Jeśli rozwijasz aplikacje lub narzędzia, które potrzebują wbudowanego serwera danych, możesz wykorzystać DB2 Express-C. Nawet jeśli DB2\_Express jest wbudowane w Twoją aplikację i rozprowadzane za każdym razem, gdy ją sprzedajesz, nadal jest ono bezpłatne. Aby móc rozprowadzać DB2 Express-C z Twoim produktem, musisz tylko zarejestrować się w IBM. Rejestracja ta jest również bezpłatna.
- **Bez ograniczeń:** Podczas gdy inne, konkurencyjne oferty baz danych nakładają ograniczenia na rozmiar bazy danych, ich liczbę lub też liczbę użytkowników, DB2 Express-C nie ma żadnych ograniczeń. Twoja baza danych może się ciągle powiększać bez ryzyka złamania umowy licencyjnej. Nie posiada również żadnych ograniczeń na liczbę połączeń lub użytkowników serwera.

### Ważne:

Aby dowiedzieć się więcej o DB2 Express-C i o roli tego produktu w świecie i Web 2.0, obejrzyj prezentację wideo:

<http://www.channeldb2.com/video/video/show?id=807741:Video:3922>

## 1.2 Pomoc użytkownika i wsparcie techniczne

Jeśli masz pytania natury technicznej dotyczące DB2 Express-C, możesz je zadać na forum DB2 Express-C. To bezpłatne forum jest monitorowane przez zespół rozwijający DB2 Express-C, ale to Społeczność bezpłatnie dostarcza większości odpowiedzi.

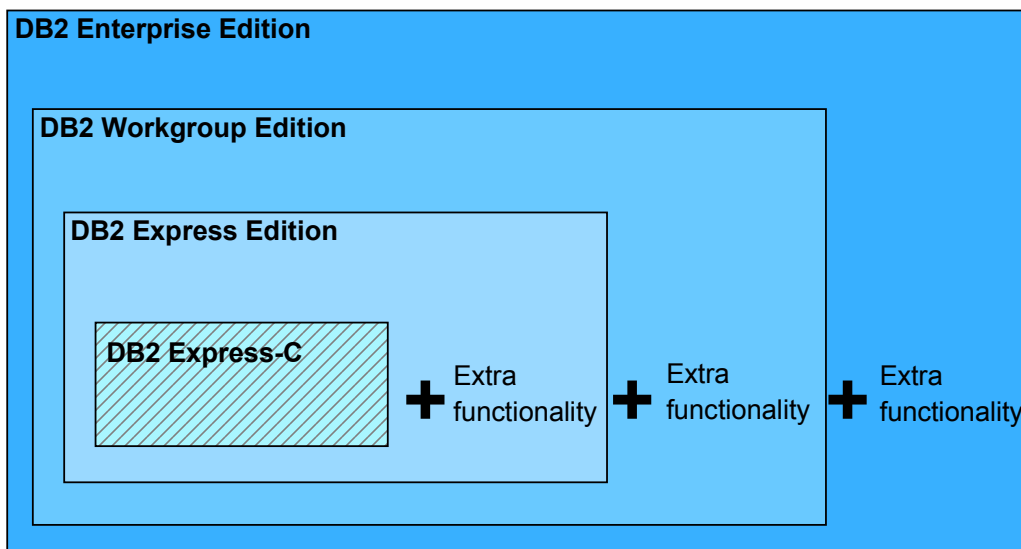
IBM zapewnia użytkownikom także możliwość zakupienia rocznej subskrypcji wsparcia w niskiej cenie (także znanej jako 12-miesięczna Licencja i Subskrypcja) lub Licencja Czasowa. Subskrypcja ta dla DB2 Express-C zawiera całodobowe wsparcie techniczne i aktualizacje oprogramowania. Dodatkowo do wsparcia i utrzymania oprogramowania, za niską opłatą subskrypcyjną (\$2,995 za serwer za rok w Stanach Zjednoczonych – może się różnić w innych krajach) otrzymujesz także dwie dodatkowe, ale kluczowe funkcjonalności: HADR (High Availability and Disaster Recovery) - mechanizm replikacji bazy danych, zapewniający dostępność danych i systemu, oraz replikację SQL (dla replikacji danych z innymi serwerami DB2).

Szczegółowe informacje o subskrypcji DB2 Express-C można znaleźć na stronie:

[www.ibm.com/db2/express/support.html](http://www.ibm.com/db2/express/support.html)

### 1.3 Serwery DB2

Wszystkie edycje serwerów DB2 zawierają takie same podstawowe komponenty; są one udostępnione w pakietach umożliwiającym użytkownikowi wybór funkcjonalności, których on potrzebuje - w odpowiedniej cenie. Rysunek 1.1 przedstawia różne edycje produktu DB2.



Rysunek 1.1 – Serwery DB2

Jak pokazuje Rysunek 1.1, DB2 Express-C jest tym samym, co DB2 Express, ale bez kilku komponentów. DB2 Express-C jest bezpłatne dla Społeczności. Jak wspomnieliśmy wcześniej, wsparcie techniczne jest zapewnione poprzez bezpłatne forum internetowe. Możesz także otrzymać oficjalne całodobowe wsparcie techniczne IBM, jeśli wykupisz dwunastomiesięczną licencję subskrypcyjną.

Rysunek 1.1 wyjaśnia także, dlaczego tak łatwo można zamienić DB2 Express-C na wersję wyższą. Jeśli w przyszłości zechcesz dokonać konwersji na inny serwer DB2, wszystkie serwery DB2 mają takie same podstawowe komponenty. Oznacza to również, że każda aplikacja stworzona na jednej edycji, będzie pracować na innej bez konieczności wprowadzania modyfikacji. Poza tym wszystkie umiejętności, które zdobędziesz pracując z jedną edycją, będą miały zastosowanie w innych.

### 1.4 Klient DB2 i sterowniki

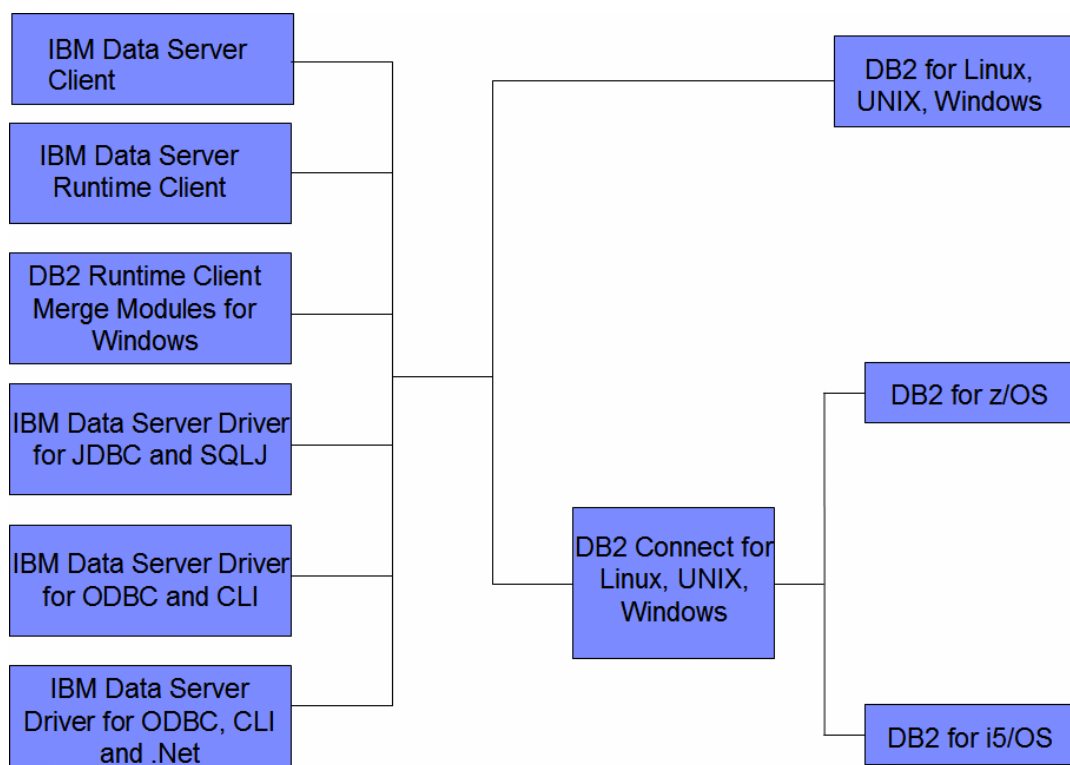
Klient DB2 zawiera wszystkie potrzebne funkcjonalności, aby połączyć się z serwerem DB2, jednak nie zawsze musi być zainstalowany. Na przykład, aplikacja JDBC typu 4 może bezpośrednio łączyć się z serwerem DB2.

Klient DB2 i sterowniki dostępne są w kilku różnych wersjach:

- IBM Data Server Client: najbardziej kompletny, zawiera narzędzia graficzne i sterowniki,

- IBM Data Server Runtime Client: mały klient z podstawowymi funkcjonalnościami, zawiera sterowniki,
- DB2 Runtime Client Merge Modules dla Windows: używany głównie do wbudowywania klienta DB2 jako części aplikacji instalacyjnej dla systemu Windows,
- IBM Data Server Driver dla JDBC i SQLJ: pozwala aplikacjom Java na łączenie się z serwerami DB2 bez potrzeby instalowania pełnego klienta,
- IBM Data Server Driver dla ODBC i CLI: pozwala aplikacjom ODBC i CLI na łączenie się z serwerem DB2 bez potrzeby instalowania klienta
- IBM Data Server Driver dla ODBC, CLI i .NET: sterownik dla Windows, który wspiera środowisko .NET razem z aplikacjami ODBC i CLI

Rysunek 1.2 pokazuje różne dostępne typy klientów i sterowników DB2.



**Rysunek 1.2 – Klienci i sterowniki DB2**

Po lewej stronie Rysunku 1.2 pokazane są wszystkie klienty i sterowniki DB2. Mimo, że wszystkie klienty DB2 zawierają potrzebne sterowniki, począwszy od DB2 9 zapewniamy także indywidualne sterowniki. Wszystkie klienty DB2 i sterowniki są bezpłatne i dostępne do pobrania ze strony DB2 Express-C. Klienty i sterowniki mogą być używane do łączenia się do serwerów DB2 na systemach Linux, UNIX lub Windows. Aby połączyć się do serwerów DB2 na z/OS® lub DB2 dla i5/OS®, musimy przejść przez serwer DB2 Connect™ (na środku Rysunku 1.2). Temat oprogramowania DB2 Connect zostanie omówiony w Rozdziale 2.

## 1.5 Swoboda w rozwijaniu aplikacji

DB2 oferuje środowisko do tworzenia aplikacji, które oparte jest na standardach i jest transparentne dla rodziny DB2. Standaryzacja SQL we wszystkich produktach linii DB2 zapewnia wspólne interfejsy programistyczne dostępu do baz danych. Dodatkowo, każdy produkt DB2 dostarcza prekompilatory SQL, które pozwalają programistom na wbudowywanie w aplikacje statycznego i dynamicznego kodu SQL. DB2 posiada także natywny sterownik dla .NET i integrację z narzędziami Microsoft® Visual Studio.

Języki i standardy, których możemy używać z DB2:

- SQL, XQuery, XPath
- C/C++ (CLI, ODBC i wbudowany SQL)
- Java (JDBC and SQLJ)
- COBOL
- PHP
- Perl
- Python
- Ruby on Rails
- Języki .NET
- OLE-DB
- ADO
- MS Office: Excel, Access, Word
- Usługi sieciowe (ang. Web services)

## 1.6 Wersje DB2, a edycje DB2

Jeśli dopiero rozpoczynasz poznawanie DB2, możesz jeszcze nie dostrzegać różnicy między wersją DB2, a edycją DB2.

Co kilka lat IBM publikuje nową wersję DB2 (ang. Version). Wersja zawiera nowe funkcjonalności i znaczące poprawki dla produktu. Aktualnie, wersje DB2 8 i DB2 9 są oficjalnie wspierane przez IBM. Wersja może mieć kilka odsłon (ang. release), które również mogą zawierać nowe funkcjonalności, ale zazwyczaj niewystarczająco znaczące, aby tworzyły nową wersję. Na przykład, 8.1 i 8.2 są odsłonami wersji 8 DB2. Patrząc wstecz, IBM publikuje nowe odsłony prawie co rok, a nowe wersje są zazwyczaj oddalone w czasie o 2-3 lata. Najnowsza odsłona to V9.5 (nazwa kodowa to DB2 "Viper 2"). Stała się ona Ogólnodostępna (ang. Generally Available) w październiku 2007 roku. Każda odsłona może również zawierać kilka poziomów Zmian (ang. Modification levels), które zazwyczaj zawierają kilka poprawek lub odnoszą się do paczek poprawek i same zawierają nowe funkcjonalności. W czasie pisania tej książki najbardziej aktualną Wersją, Odsłoną, poziomem Zmian (V,R,M) DB2 Express-C jest 9.5.0, która odpowiada poziomowi kodowemu dla wersji 9.5 z paczką poprawek 0, co oznacza, że jest ona na poziomie GA.

Z drugiej strony, edycje to wybrane funkcjonalności lub grupy paczek w każdej wersji. Jak wyjaśniono wcześniej, edycja to zestaw różnych funkcjonalności za konkretną cenę z określoną licencją. Wersja 9.5 DB2 ma kilka edycji - na przykład: DB2 Express-C 9.5, DB2 Express 9.5, DB2 9.5 Workgroup, oraz DB2 9.5 Enterprise (zobacz Rysunek 1.1).

## 1.7 Migracja do innej edycji DB2

Kiedy Twoja baza staje się większa, może się stać konieczna zmiana edycji DB2 na taką, która wspiera większe konfiguracje sprzętowe. Jeśli taka sytuacja wystąpi, zmiana edycji DB2 jest bardzo prosta:

- jeśli zmieniasz na inną edycję DB2 na tym samym systemie komputerowym, zainstaluj nową edycję DB2 z odpowiednią licencją bezpośrednio na DB2 Express-C. Twoja baza danych nie zostanie usunięta (ale stworzenie kopii zapasowej jest zawsze zalecane);
- jeśli zmieniasz DB2 tak, że nowa wersja ma być zainstalowana na innym, mocniejszym komputerze z takim samym systemem operacyjnym, zainstaluj nową edycję na tym komputerze, zrób kopie zapasowe baz na słabszym komputerze, skopiuj obrazy kopii zapasowych na mocniejszy komputer, i odzyskaj na tym komputerze bazy z kopii zapasowych. Możesz też być zmuszony zapisać ustawienia konfiguracji instancji (`dbm cfg`) ze słabszego komputera, i zastosować tę konfigurację na mocniejszym komputerze. Komendy tworzące kopie zapasowe i odzyskujące bazy są szczegółowo omówione w Rozdziale 11 „Kopie zapasowe i odzyskiwanie danych”. Polecenie `dbm cfg` jest omówione bardziej szczegółowo w Rozdziale 5 „Środowisko DB2”;
- w obu przypadkach Twoja aplikacja nie będzie wymagała żadnych modyfikacji.

## 1.8 Utrzymanie DB2 Express-C

Jak omówiliśmy wcześniej, są dwie możliwości otrzymania wsparcia dla DB2 Express-C:

1. Zakup dwunastomiesięcznej subskrypcji. Dostarczy Ci ona pełne, dwudziestoczterogodzinne wsparcie techniczne IBM. Da Ci także możliwość instalowania poprawek DB2 (ang. fixpacks).
2. Internetowe forum społecznościowe DB2 Express-C jest zupełnie darmowe, ale nie uzyskasz w ten sposób oficjalnego wsparcia z IBM. IBM nie zobowiązuje się również dostarczać nowych funkcjonalności i poprawek w wyznaczonym czasie. Dotyczy to również idei poprawek, która jest omówiona w Rozdziale 2. W zamian za to nowe obrazy całego DB2 Express-C udostępniane są od czasu do czasu. Kiedy pojawiają się nowe odsłony, możesz zazwyczaj spodziewać się udostępnienia nowego obrazu DB2 Express-C.

## 1.9 Pokrewne bezpłatne oprogramowanie

Całe oprogramowanie, które jest dostępne do pobrania ze strony DB2 Express-C ([www.ibm.com/db2/express/download.html](http://www.ibm.com/db2/express/download.html)) jest bezpłatne. Oprócz obrazów dla DB2 Express-C (dla Linux, Windows, obu architektur 32- i 64-bitowej), dostępne jest także inne przydatne oprogramowanie, które można pobrać i używać bez opłat:

- IBM Data Studio
- DB2 Net Search Extender
- DB2 Spatial Extender

Istnieje także dodatkowy zestaw narzędzi startowych oparty na DB2 Express-C, dostępny do pobrania ze strony IBM Alphaworks ([www.alphaworks.ibm.com/datamgmt](http://www.alphaworks.ibm.com/datamgmt)), który może okazać się przydatny:

- Starter Toolkit for DB2 on Rails ([www.alphaworks.ibm.com/tech/db2onrails/](http://www.alphaworks.ibm.com/tech/db2onrails/))
- Web 2.0 Starter Toolkit for DB2 ([www.alphaworks.ibm.com/tech/web2db2](http://www.alphaworks.ibm.com/tech/web2db2))

Jeśli poszukujesz małych, darmowych aplikacji serwerowych, IBM oferuje:

- WebSphere® Application Server – Community Edition (WAS CE)

### 1.9.1 IBM Data Studio

IBM Data Studio jest narzędziem opartym na platformie Eclipse, które umożliwia projektowanie, rozwijanie, wdrażanie i zarządzanie Twoimi danymi, bazami danych i aplikacjami bazodanowymi, podczas całego cyklu zarządzania danymi. IBM Data Studio zastępuje wcześniej dostępne rozwiązanie o nazwie DB2 Developer Workbench 9.1.

IBM Data Studio pomaga rozwijać definiowane przez użytkownika funkcje, procedury składowane, zapytania XQuery, polecenia SQL. Zawiera też zintegrowany debugger. Dodatkowo Data Studio pozwala Ci na pracę z diagramami modelującymi fizyczne dane (ang. Physical Data Modeling diagrams), aby zrozumieć relacje między tabelami. Pozwala Ci także rozwijać i publikować dane w serwisach internetowych bez programowania. Omówimy dokładniej IBM Data Studio w Rozdziale 14 „Procedury Składowane SQL PL”.

### 1.9.2 DB2 Net Search Extender

Przy użyciu DB2 Net Search Extender możesz wykonywać szybkie i szczegółowe przeszukiwanie tekstu w dokumentach, także w dokumentach XML przechowywanych w DB2 w sposób natywny.

### 1.9.3 Starter Toolkit dla DB2 on Rails

Starter Toolkit for DB2 on Rails to narzędzia startowe, które są wygodnie spakowanym zbiorem produktów i technologii. Umożliwiają szybkie tworzenie środowiska do budowania aplikacji internetowych DB2 przy użyciu technologii Ruby on Rails. W tych narzędziach startowych zawarte jest całe potrzebne oprogramowanie: DB2 Express-C, sterownik DB2 dla Ruby, adapter DB2 dla Rails; wraz z tutorialami, przykładami i innymi materiałami do nauki. Omówimy Ruby on Rails dokładniej w Rozdziale 17 „Programowanie w Java, PHP i Ruby”.

### 1.9.4 Web 2.0 Starter Toolkit dla DB2

Narzędzia Startowe Web 2.0 dla DB2 są łatwym sposobem by rozpocząć pracę z DB2, PHP i Dojo. Pomogą Ci wdrożyć potrzebne oprogramowanie, zawierają linki do tutoriali i przykładowe aplikacje. Dwie z przykładowych aplikacji to Centrum Sterowania Atom Feed (ang. Atom Feed Control Panel), które generuje Atom feeds z tabel DB2 oraz Centrum Sterowania Serwisem Internetowym (ang. Web Services Control Panel), które tworzy otoczkę serwisu internetowego REST (ang. REST web service wrappers) wokół tabel DB2. Obie opierają się na Dojo z powodu jego znaczących możliwości Ajax i widget.

### **1.9.5 WebSphere Application Server – Community Edition**

Serwer aplikacyjny IBM WebSphere Application Server – Community Edition (Edycja Społecznościowa) jest lekką aplikacją serwerową Java EE 5, dostępną bezpłatnie. Zbudowany w oparciu o technologię Apache Geronimo, wykorzystuje najnowsze rozwiązania społeczności open-source, aby tworzyć zintegrowaną, łatwo dostępną i elastyczną podstawę do rozwoju i wdrażania aplikacji Java. Opcjonalne wsparcie techniczne dla WAS CE jest dostępne poprzez roczną subskrypcję.



# 2

## Rozdział 2 – Pokrewne funkcjonalności i produkty

Rozdział ten opisuje funkcjonalności DB2 dostępne po zakupieniu dwunastomiesięcznej licencji subskrypcyjnej dla DB2 Express-C, a także funkcjonalności dostępne w innych edycjach DB2, w niektórych przypadkach, za dodatkową opłatą.

Możliwości dostarczane w darmowej edycji DB2 Express-C:

- podstawowa funkcjonalność DB2
- Centrum sterowania (ang. Control Center), IBM Data Studio i inne narzędzia zarządzające
- pureXML
- wykorzystanie zasobów do 2 GB i 2 rdzeni procesorów
- dostępność dla Linux, Windows, i Solaris (x86)

Funkcjonalności, które nie są dostępne w darmowej edycji DB2 Express-C, ale są dostępne po zakupieniu dwunastomiesięcznej licencji subskrypcyjnej dla DB2 Express-C:

- pakiety poprawek
- mechanizm replikacji bazy danych, zapewniający dostępność danych i systemu HADR (High Availability and Disaster Recovery)
- replikacja danych (Data Replication, Homogenous SQL)
- wykorzystanie zasobów do 4 GB i 4 rdzeni procesorów (na 2 gniazdach)

Poniższa tabela przedstawia listę funkcjonalności oraz informacje w jakich edycjach DB2 są one dostępne. Funkcjonalności, które można zakupić oddzielnie są przedstawione z nazwą dla danej edycji DB2.

Rozdział 2 – Pokrewne funkcjonalności i produkty 22

<b>Funkcja</b>	<b>Express-C fixed term license</b>	<b>DB2 Express Edition</b>	<b>DB2 Workgroup Server Edition</b>	<b>DB2 Enterprise Server Edition</b>
Homogenous SQL Replication	Tak	Tak	Tak	Tak
Net Search Extender	Tak	Tak	Tak	Tak
Spatial Extender	Tak	Tak	Tak	Tak
pureXML™ technology	Tak	pureXML Feature	pureXML Feature	pureXML Feature
High availability disaster recovery	Tak	High Availability Feature	Tak	Tak
Tivoli® System Automation	Tak	High Availability Feature	Tak	Tak
Advanced Copy Services	Nie	High Availability Feature	Tak	Tak
Online reorganization	Nie	High Availability Feature	Tak	Tak
Homogenous Federation	Nie	Homogeneous Federation Feature	Homogeneous Federation Feature	Homogeneous Federation Feature
MQT	Nie	Nie	Query Optimization Feature	Tak
MDC	Nie	Nie	Query Optimization Feature	Tak
Query parallelism	Nie	Nie	Query Optimization Feature	Tak
Connection concentrator	Nie	Nie	Nie	Tak

Funkcja	Express-C fixed term license	DB2 Express Edition	DB2 Workgroup Server Edition	DB2 Enterprise Server Edition
Table partitioning	Nie	Nie	Nie	Tak
DB2 Governor	Nie	Nie	Nie	Tak
Compression: row level	Nie	Nie	Nie	Storage Optimization Feature
Compression: backup	Nie	Nie	Nie	Storage Optimization Feature
Label-based access control (LBAC)	Nie	Nie	Nie	Advanced Access Control Feature
Geodetic Extender	Nie	Nie	Nie	Geodetic Data Management Feature
Query Patroller	Nie	Nie	Nie	Performance Optimization Feature
DB2 workload management	Nie	Nie	Nie	Performance Optimization Feature
Performance Expert	Nie	Nie	Nie	Performance Optimization Feature
Homogenous Q Replication	Nie	Nie	Nie	Homogeneous Replication Feature
Database partitioning	Nie	Nie	Nie	Nie

Tabela 2.1: Edycje DB2 Wersja 9.5: funkcjonalności i wsparcie

Funkcjonalności dostępne w innych edycjach DB2 to:

Funkcjonalności DB2 Express Edition dostępne za opłatą:

- pureXML
- HADR
- Homogenous Federation Feature – federacja baz danych

Funkcjonalności dostępne za darmo w DB2 Workgroup Edition:

- HADR
- poza dostępnością na Linux i Windows, dostępność na AIX®, Solaris, i HP-UX

Funkcjonalności DB2 Workgroup Edition dostępne za opłatą:

- pureXML
- Query Optimization Feature – optymalizacja zapytań (MQT, MDC, zapytania równoległe)
- Homogenous Federation Feature – federacja baz danych

Funkcjonalności dostępne za darmo w DB2 Enterprise Edition:

- partycjonowanie tabel (ang. Table (Range) Partitioning)
- zmaterializowane widoki (MQT: Materialized Query Tables)
- wielowymiarowe klastrowanie (MDC: Multi-dimensional Clustering)
- HADR
- Tivoli® System Automation
- koncentrator połączeń (Connection Concentrator)

Funkcjonalności DB2 Enterprise Edition dostępne za opłatą:

- pureXML
- Storage Optimization Feature (razem z możliwością kompresji)
- zaawansowana kontrola dostępu (ang. Advanced Access Control) - precyzyjne i zaawansowane zabezpieczenia
- optymalizacja wydajności (narzędzia Workload Management, Performance Expert, Query Patroller)
- analiza lokalizacji geograficznej (Geodetic Data Management)
- mechanizm federacji baz danych (Homogenous Federation Feature) - zdalny dostęp do baz DB2 i Informix®, jak do lokalnych tabel

Płatne produkty związane z DB2:

- DB2 Connect
- DB2 Warehouse Edition
- WebSphere® Federation Server
- WebSphere Replication Server

## 2.1 Funkcjonalności dostępne z subskrypcją DB2 Express-C

Sekcja ta opisuje pakiety poprawek DB2, HADR oraz replikację SQL.

### 2.1.1 Pakiety poprawek

Pakiet poprawek (ang. fixpack) jest zbiorem poprawek kodu przeznaczonym dla zainstalowanego produktu DB2, używanym aby naprawić różne błędy zgłoszone po czasie, kiedy produkt został wydany. Z zainstalowaną licencją subskrypcyjną pakiety poprawek są bezpłatne (do pobrania i zainstalowania). Zazwyczaj są udostępniane co trzy miesiące.

Aby pobrać najnowsze pakiety poprawek, wejdź na stronę wsparcia technicznego dla DB2: [http://www.ibm.com/software/data/db2/support/db2\\_9/](http://www.ibm.com/software/data/db2/support/db2_9/)

### 2.1.2 High Availability Disaster Recovery (HADR)

High Availability Disaster Recovery (HADR) jest niezawodną funkcjonalnością bazy danych, która zapewnia wysoką dostępność i odzyskiwanie danych po częściowych i całkowitych awariach. Środowisko HADR zazwyczaj zawiera dwa serwery: podstawowy (tzw. primary) i rezerwowy (tzw. secondary), które mogą być rozmieszczone w geograficznie różnych miejscach. Podstawowy serwer to ten, na którym jest przechowywana źródłowa baza danych i do której łączą się aplikacje klienckie. Kiedy wykonywane są transakcje na źródłowej bazie, logi bazowe są automatycznie przesyłane do serwera rezerwowego. Serwer rezerwowy posiada sklonowaną kopię źródłowej bazy, zazwyczaj stworzoną jako kopia zapasowa pierwszej bazy i odtworzoną na serwerze rezerwowym. Po przesłaniu logów do serwera rezerwowego są one wykonywane na jego bazie danych. Poprzez ciągłe wykonywanie logów baza rezerwowa pozostaje repliką bazy podstawowej na wypadek, gdyby baza podstawowa zawiodła.

Pełne rozwiązanie HADR, wspierane przez DB2, zapewnia:

- błyskawiczne odzyskiwanie sprawności po awarii (ang. Lightning fast failover capability), z całkowitą przejrzystością dla klientów i aplikacji klienckich,
- zupełną niepodzielność transakcji (ang. full transaction atomicity), aby zapobiec utracie danych,
- możliwość aktualizowania systemu lub aplikacji bez widocznych przerw w usługach
- zdalne odzyskiwanie sprawności (ang. Remote system failover), zapewniające pełne odzyskiwanie sprawności po awariach lokalnych centrów danych,
- łatwe zarządzanie za pomocą narzędzi graficznych DB2,
- wszystko to ze znikomym wpływem na wydajność systemu.

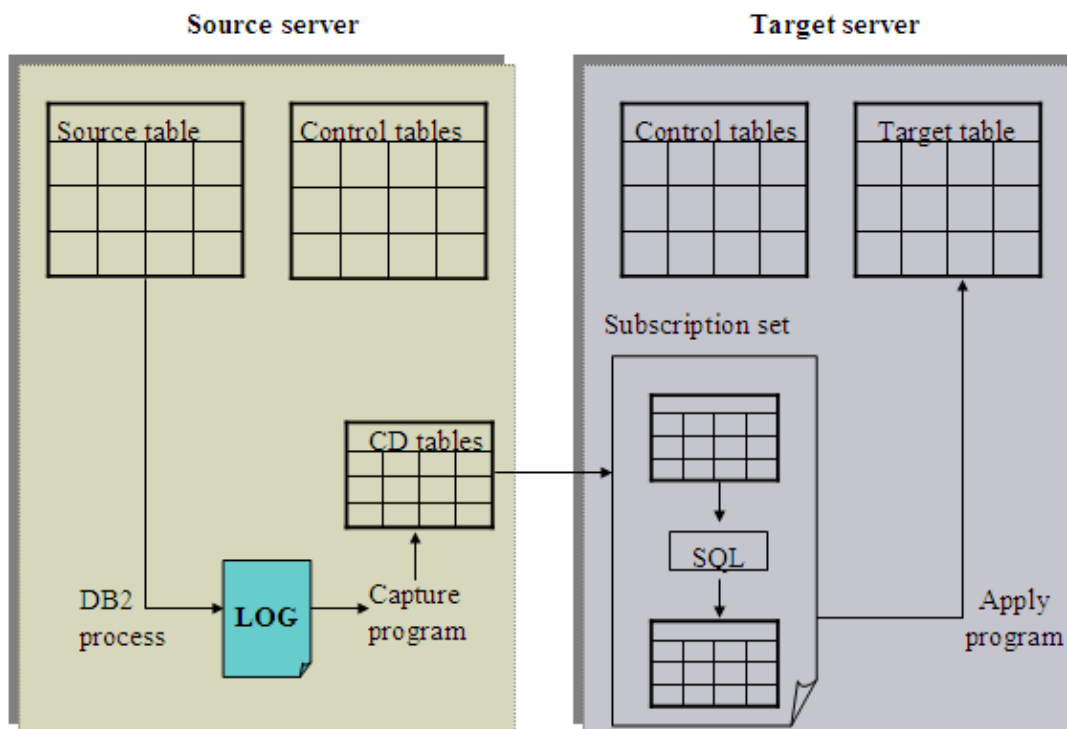
**Ważne:**

Aby obejrzeć prezentację przedstawiającą działanie HADR, odwiedź stronę:

<http://www-306.ibm.com/software/data/db2/express/demo.html>

### 2.1.3 Replikacja danych

Funkcjonalność ta umożliwia replikację danych pomiędzy źródłowym serwerem, gdzie zostały zarejestrowane zmiany danych, a serwerem docelowym, gdzie zmiany danych mają zostać zastosowane. Rysunek 2.1 przedstawia działanie replikacji.



Rysunek 2.1 – Replikacja SQL

Na Rysunku 2.1 przedstawione są dwa serwery: serwer źródłowy i serwer docelowy. Na serwerze źródłowym, program rejestrujący (ang. capture program) rejestruje zmiany wykonywane na bazie. Na serwerze docelowym program do zastosowywania zmian (ang. apply program) wykonuje zmiany na replice bazy danych. Replikacja jest pożyteczna dla realizacji wielu celów, które wymagają zreplikowanych danych, między innymi: przy zmniejszaniu obciążenia bazy, podczas ładowania danych do hurtowni danych, czy w czasie audytu historii zmian. Używając replikacji SQL możemy replikować dane pomiędzy DB2 Express-C i innymi serwerami DB2, także tymi na systemach Linux, UNIX, z/OS i i5/OS.

## **2.2 Funkcjonalności niedostępne z DB2 Express-C**

Sekcja ta opisuje niektóre z funkcjonalności dostępne w innych wersjach DB2, ale nie w DB2 Express-C.

### **2.2.1 Partycjonowanie bazy danych**

Funkcjonalność partycjonowania bazy danych (DPF – Database Partitioning) jest dostępna jedynie z DB2 Warehouse Edition. Pozwala na rozłożenie bazy danych na wiele partycji, które są na różnych komputerach. DPF jest oparty na architekturze „shared-nothing”.

Każdy komputer dodany do grupy partycji, wnosi dodatkową moc procesorową ze swoim własnym CPU i pamięcią. DPF jest szczególnie przydatny w środowiskach dużych serwerów danych, takich jak hurtownie danych, gdzie są wykonywane zapytania wspierające decyzje.

### **2.2.2 Koncentrator Połączeń**

Koncentrator połączeń (ang. Connection Concentrator) jest funkcjonalnością, która pozwala na wspieranie dużej liczby aktualnie podłączonych użytkowników. Wcześniej, każde połączenie do bazy danych wymagało jednego agenta bazy danych. Koncentrator połączeń wprowadza nową koncepcję tzw. logicznego agenta, która pozwala jednemu agentowi na obsługiwaniu kilku połączeń. Agenci zostaną dokładniej omówieni w Rozdziale 6 „Architektura DB2”.

### **2.2.3 Geodetic Extender**

DB2 Geodetic Extender jest dostępny jako płatny dodatek do DB2 Enterprise Edition. Pomaga on rozwijać aplikacje rządowe i biznesowe, które wymagają łatwej analizy lokalizacji geograficznych. DB2 Geodetic Extender może skonstruować wirtualny glob w każdej skali. Większość informacji geograficznych jest zbierana przy użyciu ogólnosiątkowych systemów, takich jak system nawigacji satelitarnej (GPS) i może być reprezentowana za pomocą szerokości i długości geograficznych (współrzędne geograficzne). Dane biznesowe, takie jak adresy, mogą być konwertowane przez DB2 Geodetic Extender do współrzędnych geograficznych, a aplikacje biznesowe pracują lepiej, kiedy przechowują dane w takiej nieprzetworzonej formie, zostawiając rzutowanie map (ziemia do płaskiej mapy) warstwie prezentacji (do pokazywania i drukowania).

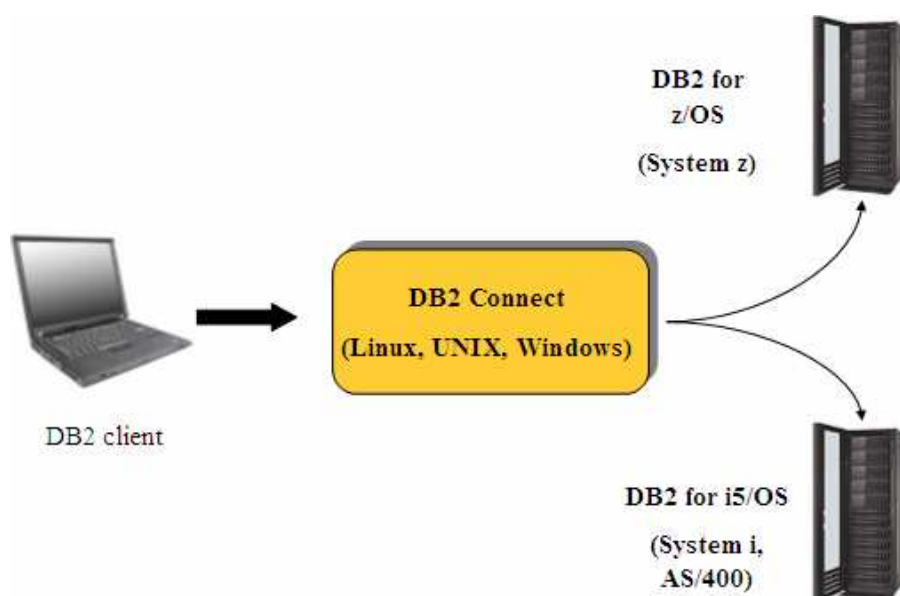
### **2.2.4 Workload Management (WLM)**

Zarządza pracą całej bazy danych pomocą w oparciu o priorytety użytkowników i aplikacji, w połączeniu z dostępnością zasobów i progami obciążenia pracą. Pozwala to na regulowanie obciążenia bazy i zapytań tak, że ważne i o wysokim priorytecie mogą być wykonane natychmiast. Zapobiega także 'zbojeckim' zapytaniom monopolizującym zasoby systemowe zapewniając efektywną pracę systemu. WLM jest nową funkcjonalnością DB2 i posiada większe możliwości niż narzędzia DB2 Query Patroller i DB2 Gubernator dostępne w poprzednich wersjach DB2.

## 2.3 Płatne produkty powiązane z DB2

### 2.3.1 DB2 Connect

DB2 Connect jest płatnym oprogramowaniem, które pozwala klientowi DB2 dla systemów Linux, UNIX i Windows na połączenie się z serwerem DB2 na systemach z/OS lub i5/OS, jak pokazano na Rysunku 2.2. DB2 Connect nie jest wymagany, jeśli połączenie zostało nawiązane w przeciwną stronę: kiedy łączysz się z DB2 dla z/OS lub DB2 dla i5/OS do serwera DB2 na systemach Linux, UNIX lub Windows. DB2 Connect jest dostępny w dwóch głównych edycjach, w zależności od Twoich wymagań co do połączenia: DB2 Connect Personal Edition i DB2 Connect Enterprise Edition.



Rysunek 2.2 – DB2 Connect

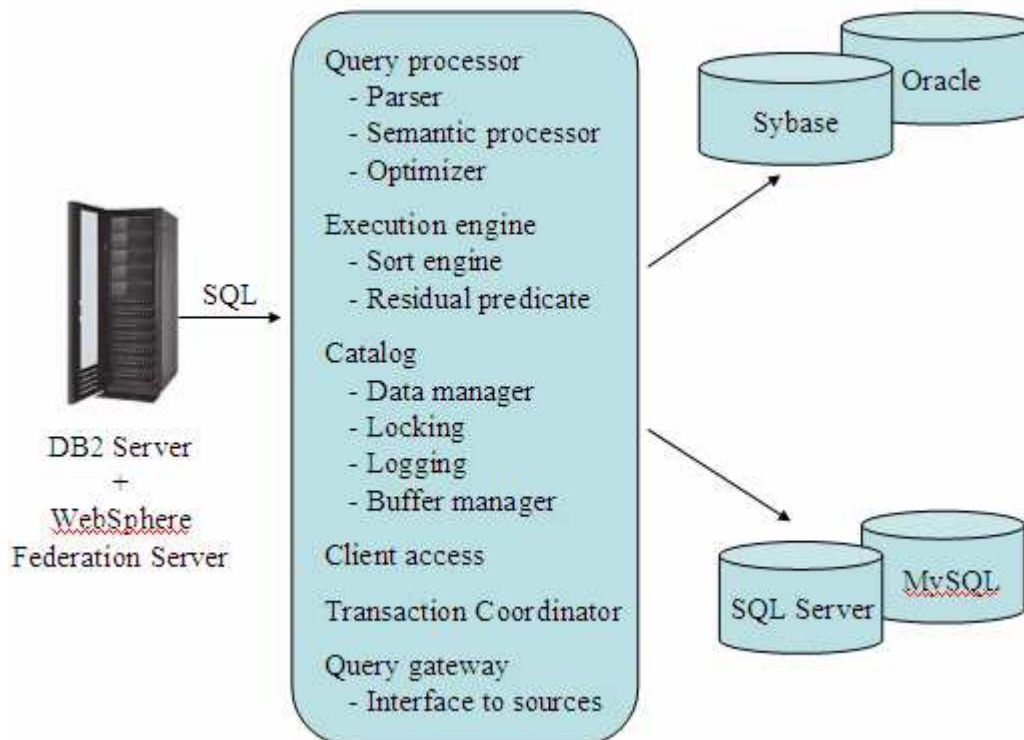
### 2.3.2 WebSphere Federation Server

Poprzednio znany jako WebSphere Information Integrator (dla wsparcia federacji), WebSphere Federation Server pozwala na budowanie sfederowanych baz danych, co oznacza, że możesz uruchomić zapytania, które mogą działać na obiektach z różnych systemów relacyjnych baz danych. Jeśli zakupisz WebSphere Federation Server możesz, na przykład, wykonać następujące zapytanie:

```
SELECT *
FROM   Oracle.Table1 A
       DB2.Table2   B
       SQLServer.Table3 C
WHERE
       A.col1 < 100
and B.col5 = 1000
and C.col2 = 'Test'
```



Rysunek 2.3 ilustruje użycie WebSphere Federation Server.



**Rysunek 2.3 – WebSphere Federation Server**

Dla systemów zarządzających relacyjnymi bazami danych, które są częścią rodziny produktów DB2, wsparcie dla federacji zostało wbudowane w DB2 Express-C. Oznacza to, że WebSphere Federation Server nie jest wymagany, kiedy na przykład, chcesz wykonać zapytanie między dwiema różnymi bazami danych DB2 lub między jedną bazą DB2, a bazą Informix (Informix jest częścią rodziny DB2).

### 2.3.3 WebSphere Replication Server

Poprzednio znany jako WebSphere Information Integrator (dla wsparcia replikacji), WebSphere Replication Server pozwala na replikację SQL rekordów bazy danych dla serwerów innych niż IBM. Zawiera też funkcjonalność znaną jako Q-Replikacja dla replikacji danych przy użyciu kolejek wiadomości (ang. message queues).



# 3

## Rozdział 3 – Instalacja DB2

Przed instalacją DB2 Express-C na Linux lub Windows upewnij się, że Twój system spełnia wszystkie wymagania.

### 3.1 Wymagania przedinstalacyjne

DB2 Express-C dostępne jest dla systemów Linux, Solaris (x64), Windows 2003, 2000, XP, Vista oraz procesorów: 32-, 64-bitowych, PowerPC (Linux). Jeśli chcesz uruchomić DB2 na innych platformach (jak np. UNIX) musisz zakupić jedną z pozostałych edycji, opisanych wcześniej w tej książce. Wymagania dotyczące systemu operacyjnego dla wszystkich edycji DB2 są także opisane na stronie:

<http://www.ibm.com/software/data/db2/udb/sysreqs.html>

Pod względem zasobów sprzętowych DB2 Express-C może zostać zainstalowane na systemie z dowolną liczbą rdzeni i o dowolnym rozmiarze pamięci. Wykorzysta wtedy maksymalnie 2 rdzenie i 2GB pamięci operacyjnej w darmowej, niezarejestrowanej wersji oraz maksymalnie 4 rdzenie i 4GB pamięci w wersji płatnej, zarejestrowanej. Przez system rozumiemy fizyczną lub wirtualną maszynę utworzoną za pomocą sprzętowej lub programowej wirtualizacji. Oczywiście możliwa jest instalacja na mniejszych systemach, np. z jednym procesorem i 1GB pamięci.

Najświeższe informacje o wymaganiach sprzętowych DB2 Express-C znajdują się na stronie: <http://www-306.ibm.com/software/data/db2/express/getstarted.html>

### 3.2 Instalacja a uprawnienia

Do instalacji DB2 Express-C na systemie Linux, Windows niezbędne jest posiadanie użytkownika z odpowiednimi uprawnieniami.

Dla systemu **Linux** konieczny jest użytkownik uprzywilejowany (ang. root). Możliwa jest również instalacja z poziomu konta zwykłego użytkownika, jednak ograniczy to nasze uprawnienia dotyczące produktu. Nie będziemy mogli np. stworzyć innych instancji niż ta utworzona przy instalacji.

Dla systemu **Windows**, użytkownik musi należeć do grupy Administratorów systemu, na którym przeprowadzana jest instalacja. Zwykły użytkownik może również przeprowadzić proces instalacji, ale tylko wtedy, gdy konfiguracja systemu pozwala na instalację oprogramowania przez nie-administratorów.

Dla kont z domeny Windows, aby uwierzytelnić użytkownika na serwerze DB2, użytkownik przeprowadzający instalację musi należeć do grupy Administratorów domeny, w której w której tworzone będą konta. Możesz również użyć wbudowanego konta administratora lokalnego do instalacji wszystkich aplikacji.

Konto użytkownika musi posiadać również uprawnienie „Pozwól na dostęp do tego komputera z sieci”.

**Ważne:**

Obejrzyj prezentację instalacji DB2 Express-C:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4442>

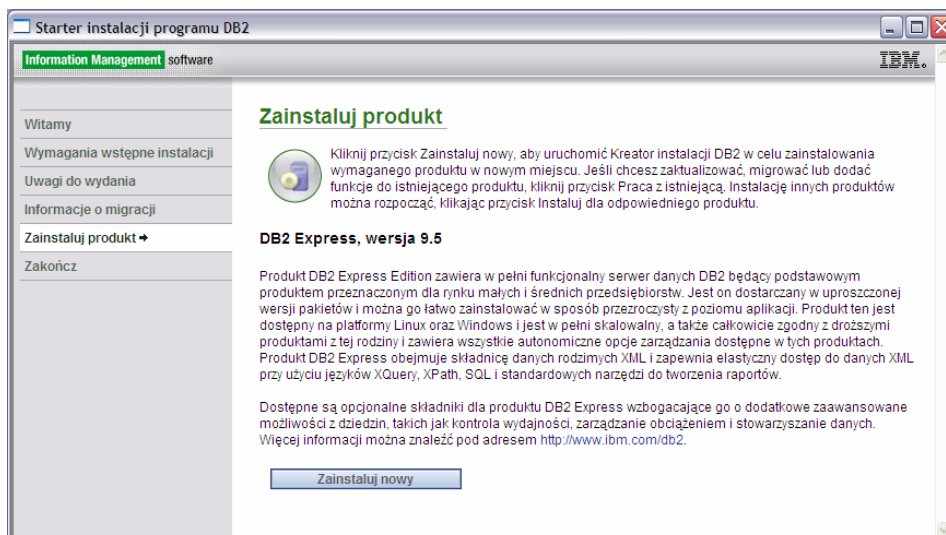
### 3.3 Kreator instalacji

Chociaż istnieje kilka metod instalacji DB2 Express-C, najprostsza jest instalacja za pomocą graficznego kreatora. Po pobraniu i rozpakowaniu plików instalatora uruchamiamy kreator za pomocą:

- Windows: pliku `setup.exe` w katalogu `EXP/image`
- Linux: komendy `db2setup` w katalogu `exp/disk1`

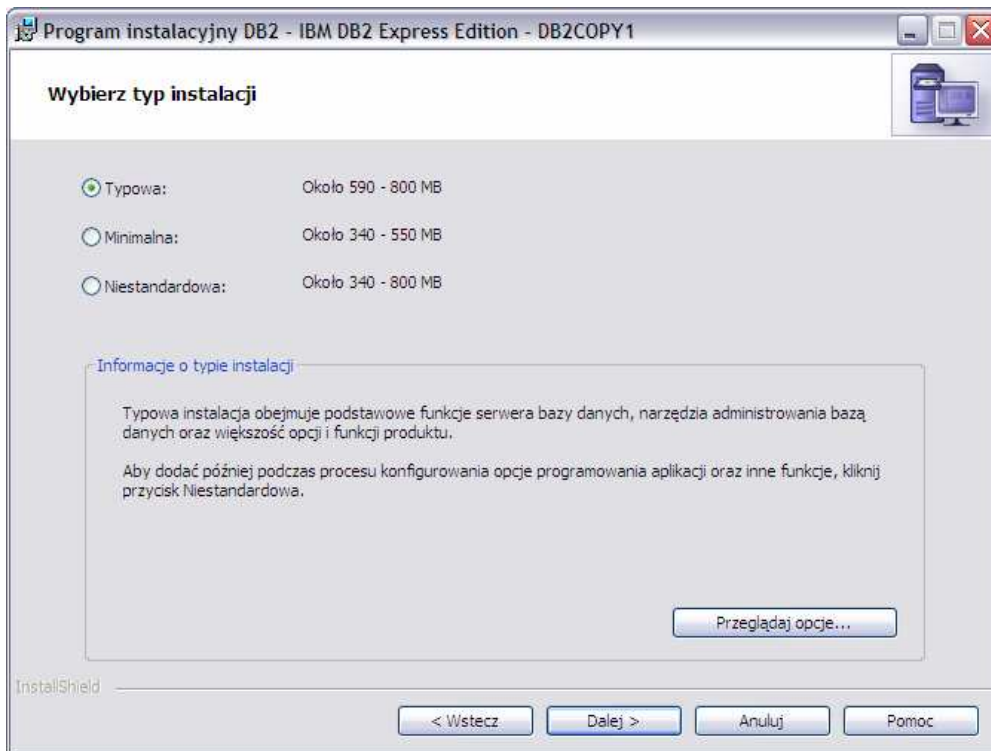
Instalacja DB2 Express-C jest bardzo prosta i opiera się na wykonywaniu poleceń instalatora. W większości przypadków wartości domyślne są wystarczające. Wszystko, co należy zrobić, to zaakceptować warunki licencji, kilka razy kliknąć przycisk *Dalej* oraz na końcu *Zakończ*. Po kilku minutach instalacja będzie zakończona, a DB2 gotowe do użycia!

Rysunek 3.1 przedstawia okno startowe instalatora. Wybierz *Zainstaluj produkt*, a następnie *Zainstaluj Nowy*.



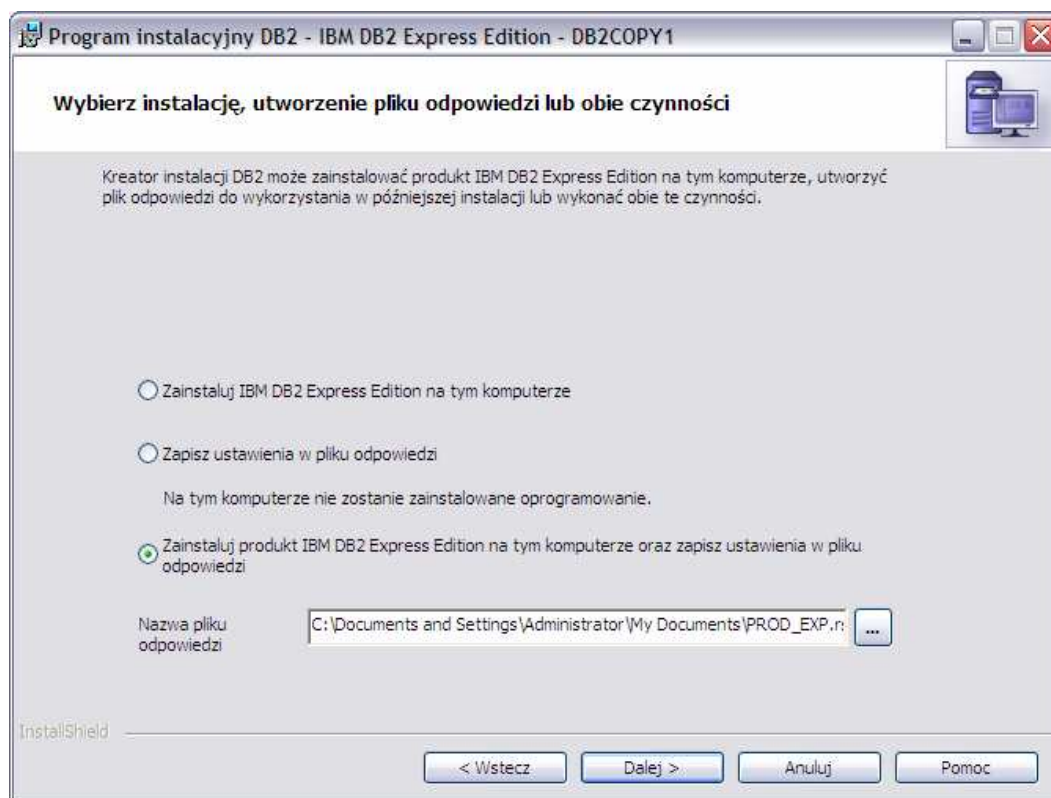
Rysunek 3.1 – Panel startowy

Po zaakceptowaniu licencji, wybieramy rodzaj instalacji. Zwykle najlepiej zaznaczyć opcję *Typowa*, jak pokazano na Rysunku 3.2.



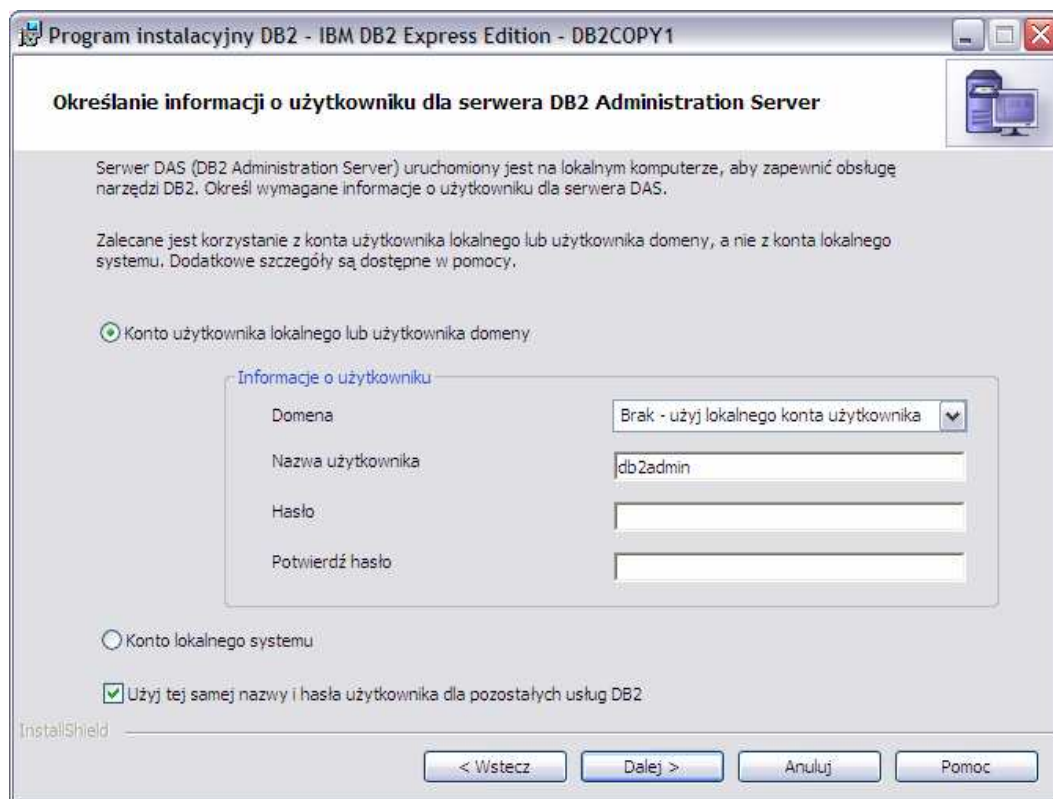
**Rysunek 3.2 – Typy instalacji**

Jak przedstawia Rysunek 3.3, wybieramy pomiędzy samą instalacją, instalacją z utworzeniem pliku odpowiedzi (ang. response file) lub z obiema opcjami jednocześnie. Pliki odpowiedzi będą omówione w sekcji 3.4 „Instalacja automatyczna”. Wybieramy domyślną opcję: *Zainstaluj produkt DB2 Express Edition na tym komputerze oraz zapisz ustawienia w pliku odpowiedzi*.



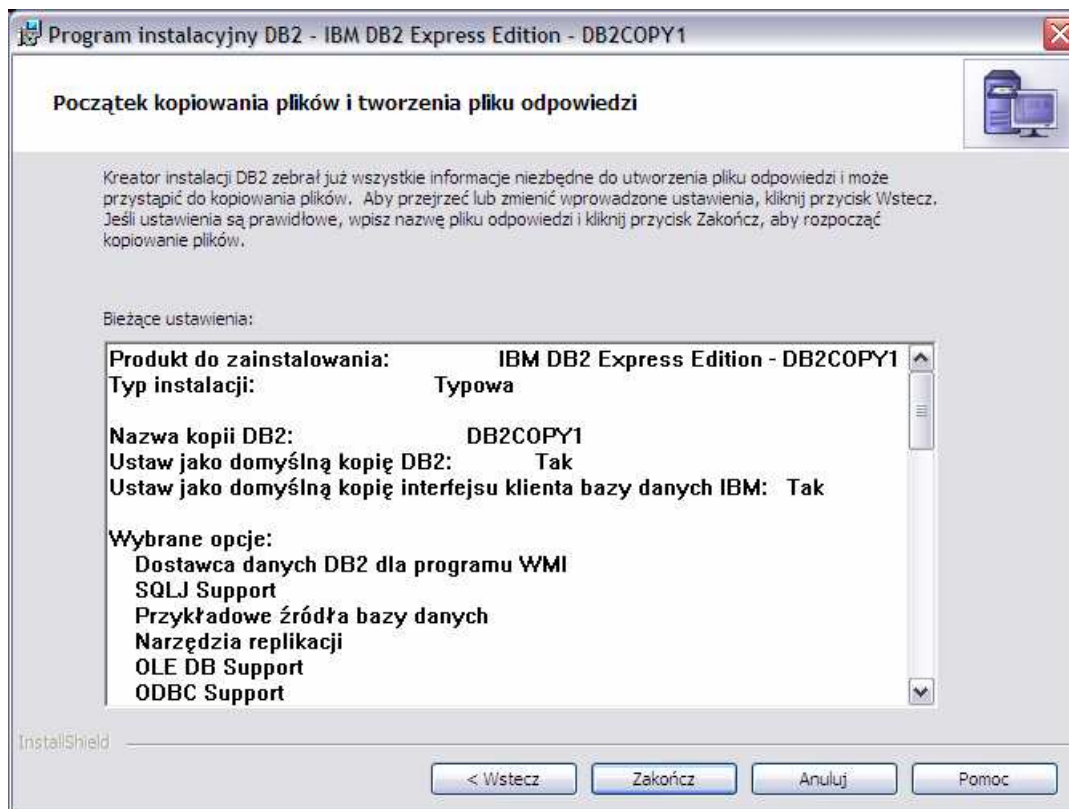
Rysunek 3.3 – Wybór opcji instalacji

Wybieramy wartości domyślne w kilku kolejnych oknach. W oknie z Rysunku 3.4, wprowadzamy nazwę użytkownika, który będzie zarządzał instancją i innymi usługami. Użytkownik musi należeć do grupy administratorów lokalnych systemu Windows. Jeśli podany użytkownik nie istnieje, zostanie stworzony i dodany do tej grupy. Dla użytkowników nie należących do domeny ustawiamy wartość pola *Domena* na *Brak*. Domyślnym użytkownikiem tworzonym w systemie Windows jest db2admin, a w przypadku systemu Linux db2inst1.



Rysunek 3.4 – Użytkownik dla instancji domyślnej.

Rysunek 3.5 pokazuje podsumowanie wcześniej wybranych opcji. Po kliknięciu na przycisku *Zakończ* rozpocznie się instalacja i pliki zostaną umieszczone w systemie.



Rysunek 3.5 – Podsumowanie instalacji.

### 3.4 Instalacja automatyczna

Zdarza się, że musimy zainstalować klienta DB2 na wielu komputerach lub potrzebujemy włączyć serwer DB2 do własnej aplikacji oraz zintegrować proces instalacji. W takich wypadkach instalacja automatyczna (ang. silent installation) będzie najlepszym rozwiązaniem.

DB2 umożliwia instalację automatyczną przy pomocy plików odpowiedzi (ang. response files). Zawierają one informacje dotyczące instalacji. Poniżej widzimy fragment takiego pliku.



Fragment pliku odpowiedzi:

```
PROD=UDB_EXPRESS_EDITION
LIC_AGREEMENT=ACCEPT
FILE=C:\Program Files\IBM\SQLLIB\
INSTALL_TYPE=TYPICAL

LANG=EN

INSTANCE=DB2
DB2.NAME=DB2
DEFAULT_INSTANCE=DB2
DB2.SVCENAME=db2c_DB2
DB2.DB2COMM=TCPIP
...
```

Pliki odpowiedzi możemy tworzyć na kilka sposobów:

- ▶ jedno z pierwszych okien kreatora instalacji pozwala na zapis aktualnych parametrów do pliku odpowiedzi. Po zakończeniu instalacji plik ten znajdzie się we wskazanym katalogu. Jest to plik tekstowy, więc możliwa jest jego późniejsza edycja.
- ▶ edytując plik odpowiedzi dostarczony razem z instalatorem. Przykładowy plik (z rozszerzeniem .rsp) znajduje się w katalogu /db2/platform/samples
- ▶ dla systemu Windows plik odpowiedzi możemy wygenerować za pomocą polecenia:  
db2rspgn -d <output directory>

Instalacja za pomocą pliku odpowiedzi:

- `setup -u <response filename>` na systemie Windows
- `db2setup -r <response filename>` na systemie Linux

## Laboratorium #1: Instalacja DB2 i tworzenie przykładowej bazy

### Cel

Zanim zaczniesz poznawać funkcje i narzędzia dostarczone razem z DB2 Express-C musisz najpierw przejść przez proces instalacji. W Laboratorium tym przeprowadzimy podstawową instalację na systemie Windows. Jeśli chodzi o platformę Linux, dostępny jest identyczny instalator, a poszczególne kroki są analogiczne.

### Kroki

1. **Pliki instalacyjne.** Ściągnij odpowiedni instalator DB2 Express-C lub zamów DiscoveryKit DVD ze strony DB2 Express-C ([ibm.com/db2/express](http://ibm.com/db2/express)) i rozpakuj go do dowolnego folderu.
2. Przejdź do katalogu (lub dysku), gdzie rozpakowałeś pliki instalacyjne.
3. **Uruchomienie panelu startowego.** Uruchom panel startowy przez podwójne kliknięcie ikony pliku `setup.exe` w systemie Windows, lub `db2setup` (jako root) w systemie Linux. Z lewej części panelu startowego wybierz *Zainstaluj produkt..*
4. **Uruchomienie kreatora instalacji.** Kreator instalacji sprawdza, czy spełnione są wymagania systemowe oraz obecność innych instalacji DB2. Kliknij *Dalej*, aby kontynuować instalację.
5. **Umowa licencyjna.** Przeczytaj i zaakceptuj warunki licencji klikając na przycisk *Akceptuję warunki umowy licencyjnej*. Naciśnij *Dalej*, aby kontynuować.
6. **Wybór typu instalacji.** Wybierz opcję *Typowa* (domyślna). Wybór *Minimalna* zainstaluje jedynie podstawowe elementy, a *Niestandardowa* pozwoli samodzielnie dostosować instalowane elementy. Kliknij *Dalej*, aby kontynuować.
7. **Folder docelowy.** Wybieramy dysk i katalog, gdzie zainstalowany zostanie program. Upewnij się, czy ilość wolnego miejsca jest wystarczająca. W tym przykładzie używamy domyślnej ścieżki:

**Dysk:** C:

**Katalog:** C:\Program Files\IBM\SQLLIB

Kliknij *Dalej*, aby kontynuować.

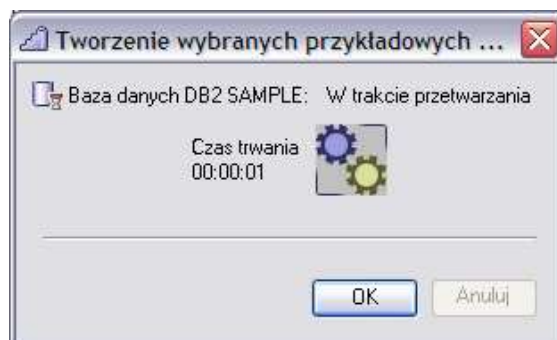
8. **Użytkownik.** Po instalacji każdy proces DB2 działa jako usługa systemowa, która wymaga użytkownika, aby mogła zostać uruchomiona. Na systemie Windows zalecane jest użycie domyślnego użytkownika *db2admin*. Jeśli podany użytkownik nie istnieje, zostanie on utworzony. Możemy również podać nazwę istniejącego użytkownika, ale musi należeć do grupy lokalnych administratorów. Zalecane jest jednak stosowanie domyślnych nazw. Sprawdź również, czy ustawione jest hasło dla użytkowników. W systemie Linux domyślnym właścicielem instancji jest **db2inst1**, **db2fenc1** jest użytkownikiem chronionym, a **dasusr1** zarządza serwerem administracyjnym. Kliknij *Dalej*, aby kontynuować.
9. **Konfiguracja instancji.** Instancję możemy porównać do pojemnika na bazy danych. Instancja musi istnieć przed utworzeniem w niej bazy danych. W systemie Windows domyślnie tworzona jest instancja DB2, w systemie Linux `db2inst1`. Omówimy ten proces w dalszej części książki.

Domyślnie, instancja DB2 jest skonfigurowana, aby nasłuchiwać na porcie TCP/IP 50000. Oba te parametry (port i protokół) możemy zmienić klikając *Konfiguruj* i wybierając zakładkę *TCP/IP*. Zalecamy jednak użycie domyślnych wartości dla tego przykładu. Kliknij *Dalej*, aby kontynuować.

10. **Instalacja.** Sprawdź ponownie wszystkie wybrane opcje, a następnie kliknij przycisk *Zakończ*. Przeprowadzona zostanie także wstępna konfiguracja.
11. **Pierwsze kroki.** Po zakończonej instalacji uruchomione zostanie narzędzie *Pierwsze kroki*. Może być ono również uruchomione później komendą `db2fs`.
12. Baza danych SAMPLE jest używana do celów testowych. Tworzona jest automatycznie razem z instalacją DB2. Sprawdź, czy baza danych istnieje za pomocą narzędzia *Centrum sterowania*. Aby je uruchomić wybierz z menu Start → Programy → IBM DB2 → DB2COPY1(domyślnie) → Narzędzia administracji ogólnej → Centrum sterowania. Z linii komend uruchamiamy `db2cc`.
13. Jeśli baza danych SAMPLE jest widoczna w *Centrum sterowania*, możemy pominąć ten krok i przejść do 16. Jeśli nie, to najpierw spróbujemy odświeżyć dane, przez kliknięcie *Odśwież* z menu *Widok*, aby mieć pewność, że widzimy aktualne dane. Jeśli i to nie dało rezultatu, tworzymy bazę ręcznie za pomocą narzędzia *Pierwsze kroki*. Wybierz zakładkę *Tworzenie bazy danych*, a następnie za pomocą kreatora utwórz bazę danych SAMPLE. Sprawdź, czy opcja *Obiekty XML i SQL oraz dane* jest aktywna. Dzięki temu utworzymy bazę danych z kodowaniem *UNICODE*, co jest

wymagane w wersji 9 do obsługi *puerXML* (wersja DB2 9.5 już tego nie wymaga).

14. Poniższe okno jest wyświetlane podczas procesu tworzenia bazy danych (może to trwać kilka minut). Po zakończeniu kliknij *OK* i zamknij narzędzie *Pierwsze kroki*.

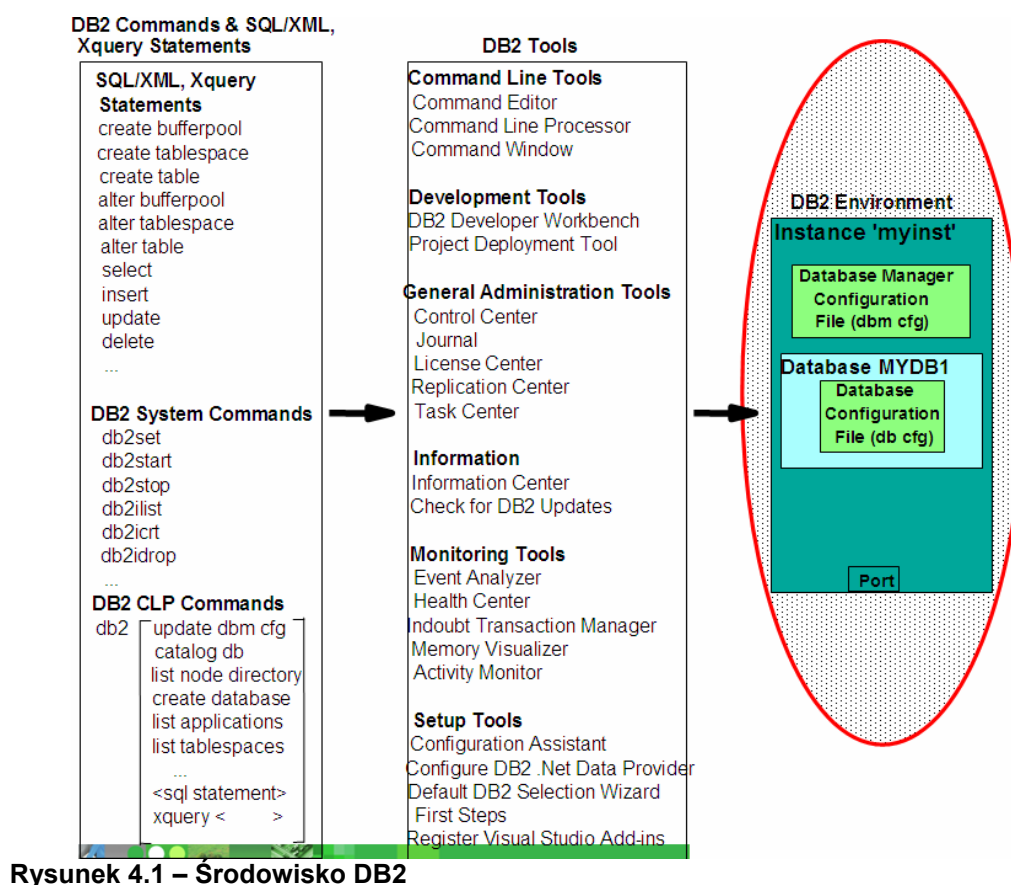


15. Uruchom ponownie *Centrum sterowania* i sprawdź, czy baza danych SAMPLE jest już widoczna w panelu *Widok obiektu*. Ponownie możesz odświeżyć wyświetlane informacje, przez kliknięcie *Odśwież*, z menu *Widok*.
16. **Uruchom ponownie system.** Chociaż krok ten nie jest wspomniany w oficjalnej dokumentacji DB2 sugerujemy, jeśli to możliwe, restart systemu (szczególnie dotyczy to Windows), aby mieć pewność, że procesy zostały wystartowane, a zasoby pamięci wyczyszczone. Ten krok jest opcjonalny!

# 4

## Rozdział 4 – Środowisko DB2

W tym rozdziale omówimy środowisko DB2. Rysunek 4.1 pokazuje wszystkie części składowe DB2, a czerwoną elipsą zaznaczono to, czym zajmiemy się w tym rozdziale. Lewa strona rysunku przedstawia różne polecenia DB2, SQL, SQL/XML oraz wyrażenia XQuery, które mogą być użyte do interakcji z serwerem DB2. Środkowa część rysunku obrazuje nazwy wszelkich narzędzi, które mogą być używane do obsługi serwera DB2. Po prawej stronie umieszczono podstawowe środowisko DB2 składające się z instancji, bazy danych oraz związanych z nimi plików konfiguracyjnych.



Rysunek 4.1 – Środowisko DB2

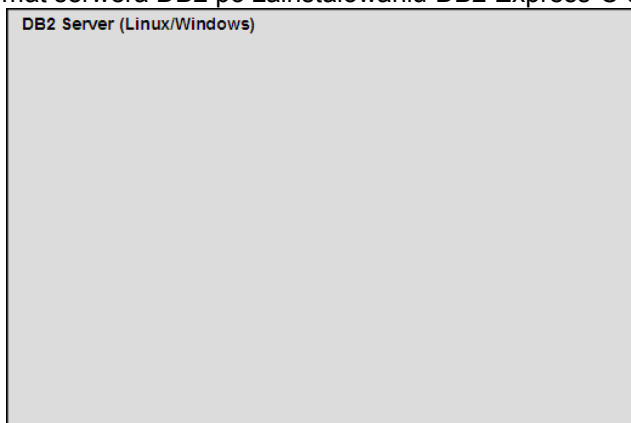
**Uwaga:**

Obejrzyj prezentacje wideo znajdujące się pod poniższymi linkami:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4029>

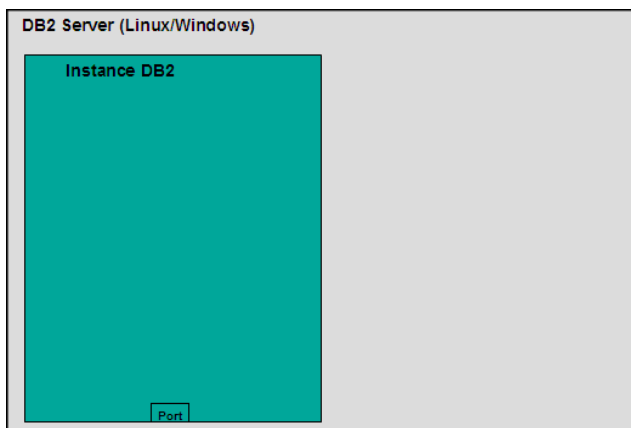
<http://www.channeldb2.com/video/video/show?id=807741:Video:4042>

Aby dobrze opisać środowisko DB2, omówimy każdy komponent krok po kroku. Rysunek 4.2 pokazuje schemat serwera DB2 po zainstalowaniu DB2 Express-C 9.5.



Rysunek 4.2 – Schemat serwera DB2 po instalacji DB2 Express-C 9.5

Podczas instalacji na systemie Windows tworzona jest domyślna instancja o nazwie „DB2” (na Linux „db2inst1”). Na Rysunku 4.3 przedstawiono to jako zielony prostokąt. Instancja jest po prostu niezależnym środowiskiem, w którym mogą być uruchamiane aplikacje oraz mogą być tworzone bazy danych. Na serwerze można tworzyć wiele instancji i używać ich do różnych celów. Na przykład, jedna instancja może być wykorzystywana do przechowywania bazy danych do celów produkcyjnych, inna instancja może pełnić rolę środowiska testowego, a jeszcze inna instancja może być wykorzystywana przez programistów. Wszystkie te instancje są niezależne, co oznacza, że czynności wykonywane na jednej instancji nie wpływają na pozostałe.



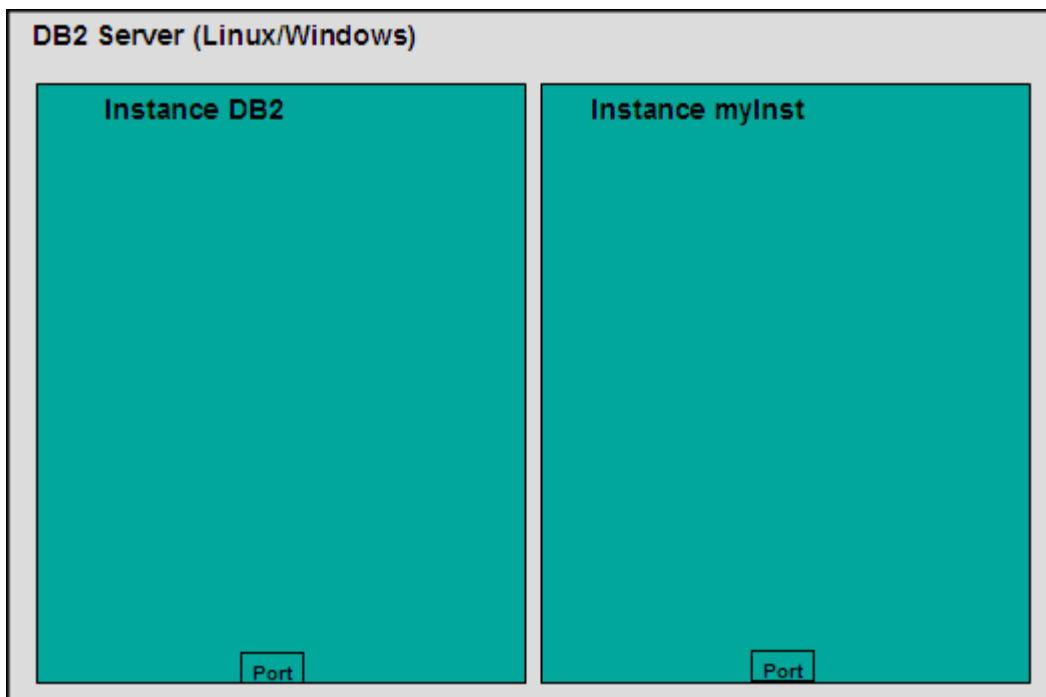
Rysunek 4.3 – Utworzona została domyślna instancja DB2

Aby utworzyć nową instancję DB2, należy użyć polecenia:

```
db2icrt <nazwa instancji>,
```

gdzie <nazwa instancji> może zawierać 8 dowolnych znaków. Na przykład, aby utworzyć instancję `myinst`, należy użyć polecenia: `db2icrt myinst`.

Rysunek 4.4 pokazuje nową instancję `myinst` jako odrębny zielony prostokąt



Rysunek 4.4 – Serwer DB2 z dwoma instancjami

Zauważmy, że każda z instancji ma swój indywidualny numer portu. Pomaga to rozróżnić instancje, kiedy chcemy połączyć się klientem zdalnym z bazą danych znajdującą się na konkretnej instancji. Używając Okna komend DB2 możemy ustawić dowolną instancję DB2 jako bieżącą używając polecenia (dotyczy systemu Windows):

```
set db2instance=myinst
```

W tym przykładzie, jeśli utworzymy teraz bazę danych z Okna komend, zostanie ona utworzona w ramach instancji `myinst`.

Aby wyświetlić wszystkie instancje należy użyć polecenia:

```
db2ilist
```

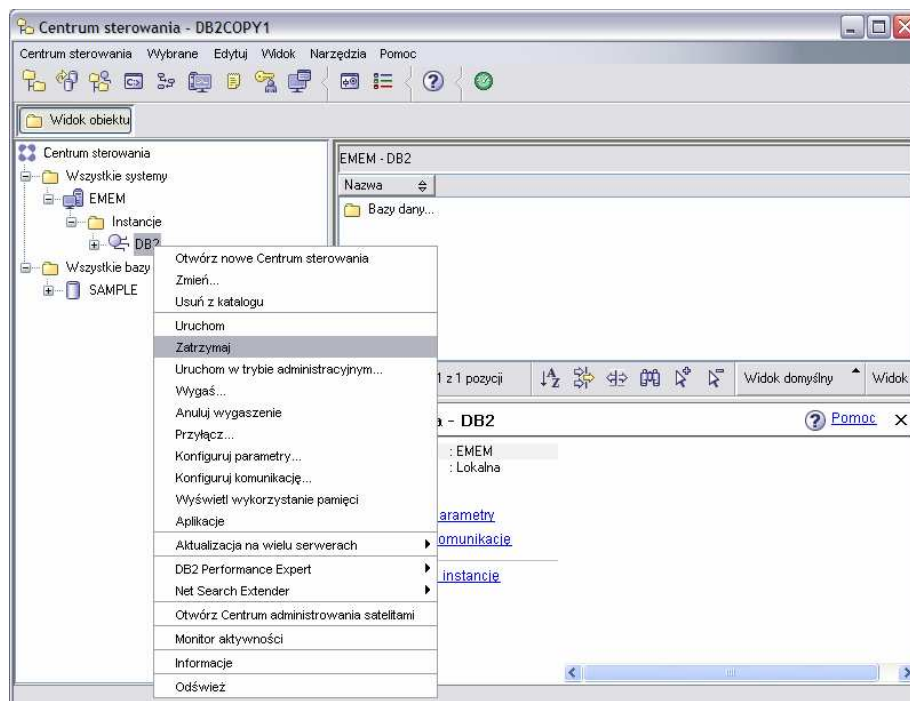
Na systemie Linux nazwa instancji musi pasować do nazw użytkowników systemowych. W związku z tym, aby przełączyć się pomiędzy instancjami należy po prostu przełączyć użytkowników (poleceniem `su`).

Tabela 4.1 zawiera kilka przydatnych poleceń do obsługi instancji.

Komenda	Opis
db2start	Startuje bieżącą instancję
db2stop	Zatrzymuje bieżącą instancję
db2icrt	Tworzy nową instancję
db2idrop	Usuwa instancję
db2iilist	Wyświetla wszystkie dostępne instancje
db2 get instance	Wyświetla bieżącą instancję

**Tabela 4.1 – Użyteczne komendy DB2 dotyczące instancji**

Niektóre z powyższych poleceń mogą być wykonane przy użyciu narzędzia Centrum sterowania. Na przykład, jeśli rozwiniemy w Centrum sterowania folder instancji i klikniemy prawym klawiszem wybraną instancję, możemy ją uruchomić wybierając *Uruchom*, co jest tożsame z wywołaniem polecenia `db2start` z Okna komend. Możemy także zatrzymać instancję wybierając *Zatrzymaj*, co jest tożsame z użyciem polecenia `db2stop`, tak jak to pokazuje Rysunek 4.5.



**Rysunek 4.5 – Polecenia dotyczące instancji, które można wywołać z poziomu Centrum sterowania**



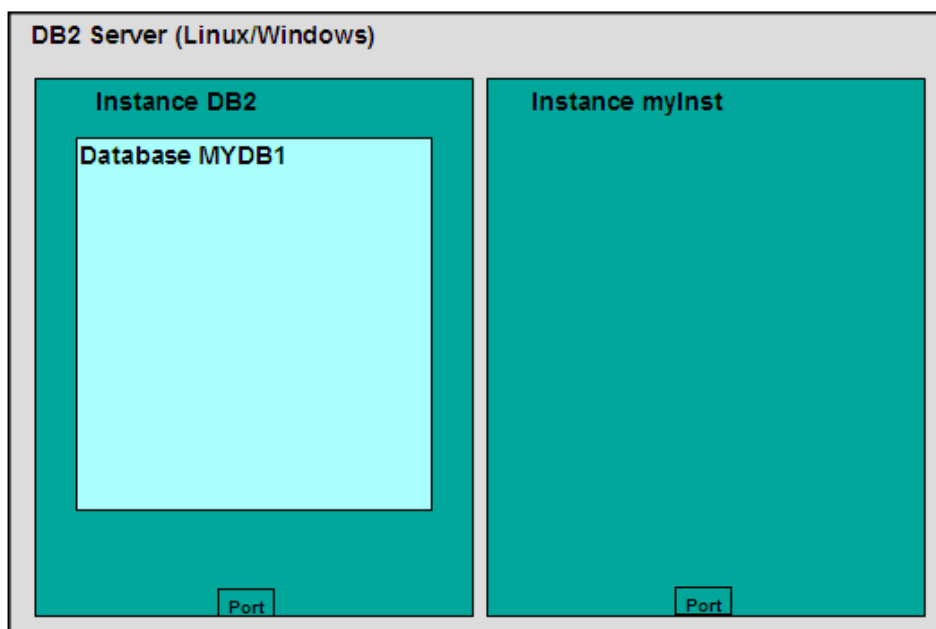
Aby utworzyć bazę danych w ramach bieżącej instancji należy z poziomu Okna komend wywołać polecenie:

```
db2 create database mydb1
```

Aby wyświetlić wszystkie utworzone bazy danych należy użyć polecenia:

```
db2 list db directory
```

W obrębie każdej z dostępnych instancji możemy utworzyć wiele baz danych. Baza danych jest kolekcją obiektów, takich jak tabele, widoki, indeksy, itd. Każda baza jest niezależna i nie współdzieli swoich obiektów z innymi bazami danych. Rysunek 4.6 prezentuje bazę danych „MYDB1” utworzoną w ramach instancji „DB2”.



Rysunek 4.6 – Baza danych “MYDB1” utworzona wewnątrz instancji „DB2”

Tabela 4.2 przedstawia wybrane polecenia do obsługi bazy danych.

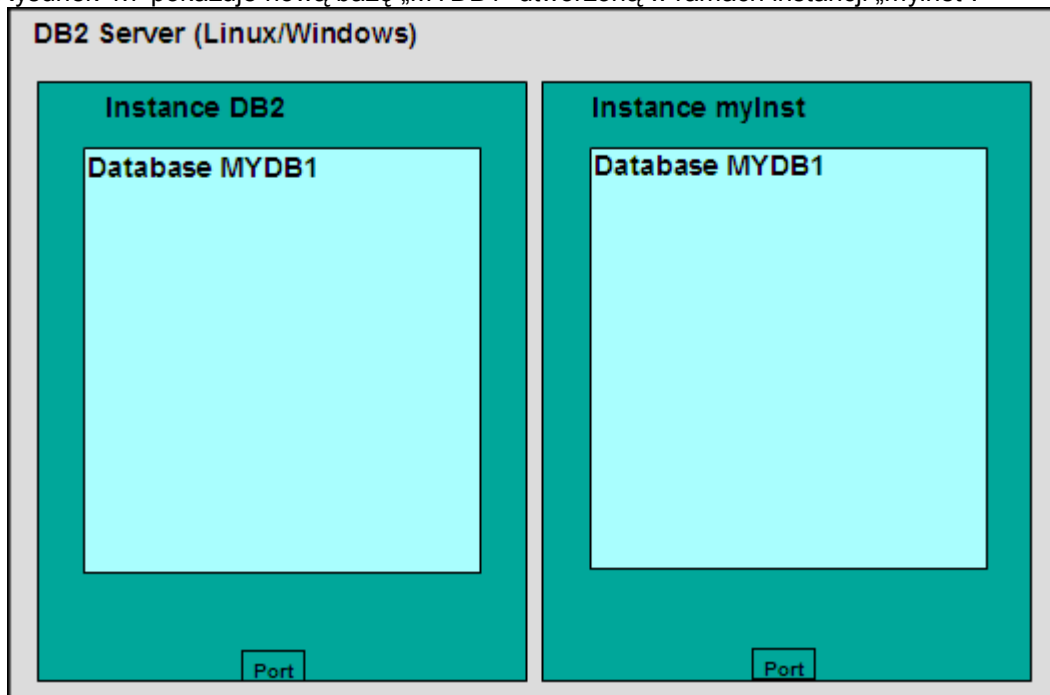
Komenda/Polecenie SQL	Opis
db2 create database	Tworzy nową bazę danych
db2 drop database	Usuwa bazę danych
db2 connect to <database_name>	Łączy z bazą danych
db2 create table/create view/create index	Wyrażenie SQL, które tworzy odpowiednio tabelę, widok i indeks

Tabela 4.2 – Komendy/Polecenia SQL do obsługi bazy danych

Poniższe polecenia pokazują, jak utworzyć nową bazę o tej samej nazwie (MYDB1), ale w ramach instancji „myinst”:

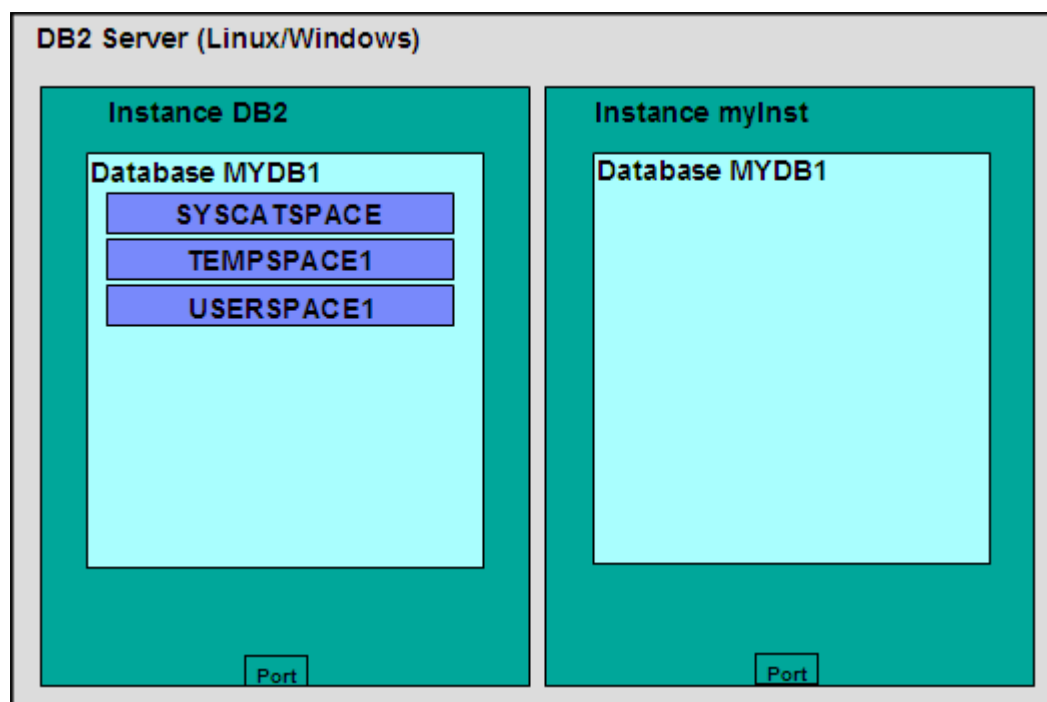
```
db2 list db directory
set db2instance=myinst
db2 create database mydb1
set db2instance=db2
```

Rysunek 4.7 pokazuje nową bazę „MYDB1” utworzoną w ramach instancji „myinst”.



Rysunek 4.7 – Baza danych “MYDB1” utworzona w ramach instancji “myinst”

Podczas tworzenia bazy danych automatycznie tworzone są inne obiekty: przestrzenie tablicowe, tabele, pula buforów i pliki logów. Utworzenie tych wszystkich obiektów zajmuje trochę czasu, dlatego wykonanie polecenia `create database` może potrwać kilka minut. Rysunek 4.8 przedstawia trzy przestrzenie tablicowe, które są domyślnie tworzone. Przestrzenie tablicowe zostaną dokładniej omówione w Rozdziale 6 „Architektura DB2”. W tej chwili wystarczy rozumieć je jako logiczną warstwę pomiędzy logicznymi tabelami, a fizycznymi zasobami, takimi jak np. dysk lub pamięć.



**Rysunek 4.8 – Przestrzenie tablicowe utworzone domyślnie podczas tworzenia bazy danych**

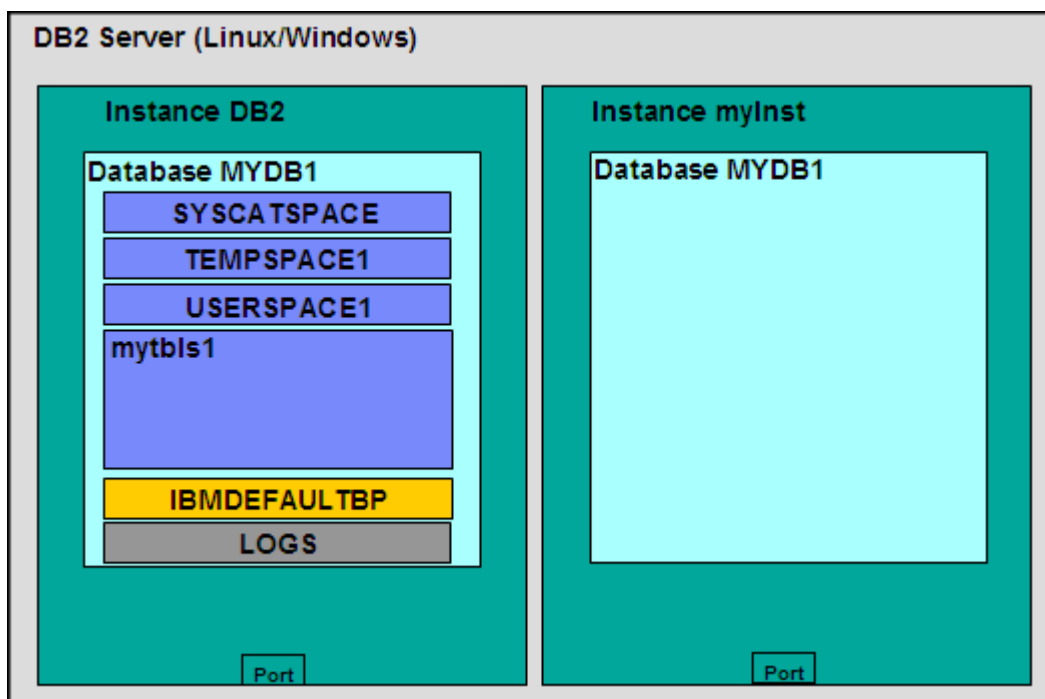
Przestrzeń tablicowa SYSCATSPACE zawiera tabele katalogu. Tabele te są również znane w innych systemach zarządzania relacyjną bazą danych jako słownik danych. Przede wszystkim zawierają informacje o systemie, które nie powinny być modyfikowane lub usuwane. W przeciwnym wypadku baza danych może nie działać poprawnie. Przestrzeń tablicowa TEMPSPACE1 jest używana przez DB2, kiedy zaistnieje potrzeba wykorzystania dodatkowego miejsca, aby wykonać pewne czynności, np. takie jak sortowanie. Przestrzeń tablicowa USERSPACE1 jest zwykle używana do przechowywania tych tabel użytkownika, dla których użytkownik nie określi przestrzeni tablicowej podczas tworzenia tabeli.

Można również utworzyć własną przestrzeń tablicową poleceniem CREATE TABLESPACE. Rysunek 4.9 pokazuje przestrzeń tablicową MYTBLS1 utworzoną wewnątrz bazy danych MYDB1, w ramach instancji DB2. Podczas tworzenia przestrzeni tablicowych należy wybrać dysk oraz pamięć (pulę buforów). Zatem, jeśli mamy tabelę, która jest bardzo często używana, możemy ją umieścić na najszybszym dysku i przydzielić jej najwięcej pamięci poprzez przypisanie przestrzeni tablicowej odpowiednich parametrów.

Na Rysunku 4.9 pokazujemy dwa inne obiekty, które zostały domyślnie utworzone: pula buforów IBMDEFAULTBP oraz pliki logów.

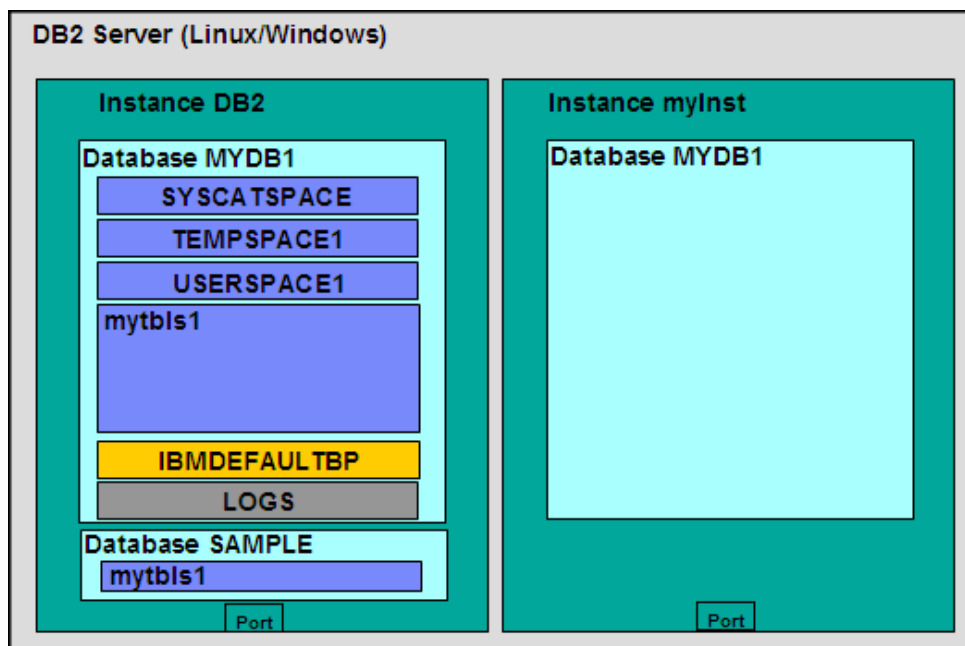
Pula buforów jest po prostu pamięcią używaną przez bazę danych. Można utworzyć kilka pul buforów, ale zawsze musi istnieć przynajmniej jedna pula buforów o wielkości strony, która odpowiada wielkości stron istniejących przestrzeni tablicowych. Strony i wielkość stron będą omawiane bardziej szczegółowo w Rozdziale 6 „Architektura DB2”.

Pliki logów są przydatne przy odzyskiwaniu danych. Podczas pracy z bazą danych, na dysku przechowywane są nie tylko informacje dla bazy danych, ale również pliki logów przechowujące wszystkie operacje wykonywane na danych. Traktujmy logi jako pliki tymczasowe, do których wykonywany jest automatyczny zapis. Logi zostaną omówione bardziej szczegółowo w Rozdziale 11 „Kopie zapasowe i odzyskiwanie danych”.



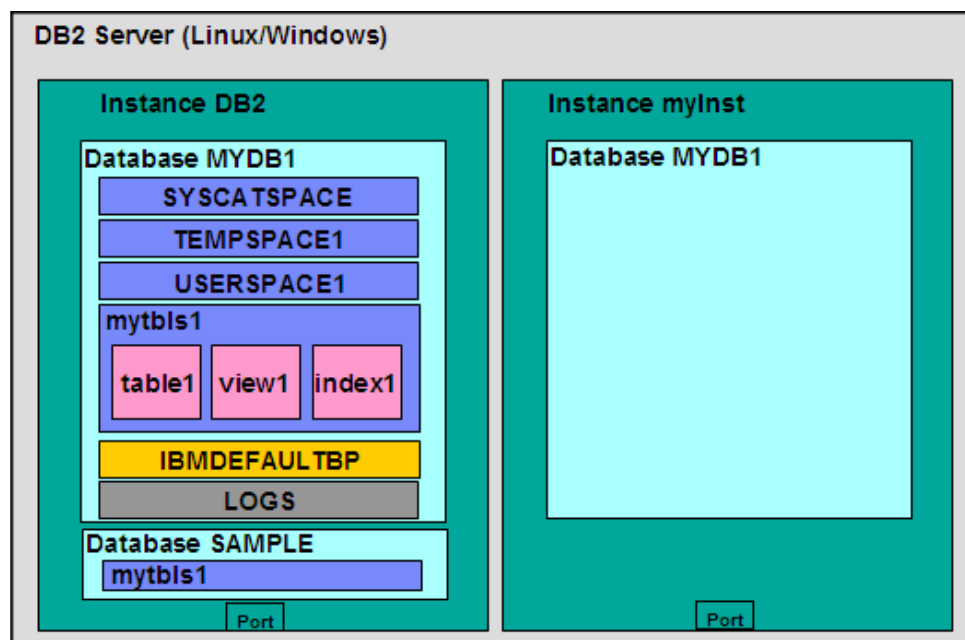
Rysunek 4.9 – Domyślnie utworzona pula buforów i logi

Wcześniej wspomnieliśmy, że instancje są niezależnymi środowiskami i dlatego można tworzyć bazy danych o takiej samej nazwie pod różnymi instancjami. Bazy danych, tak jak instancje, są niezależnymi jednostkami, dlatego obiekty w jednej bazie danych nie mają powiązań z obiektami w innej bazie danych. Rysunek 4.10 pokazuje przestrzeń tablicową „mytbls1” zarówno wewnątrz bazy danych MYDB1, jak i bazy danych SAMPLE wewnątrz instancji DB2. Jest to poprawne, ponieważ bazy danych są niezależnymi jednostkami. Zauważ, że Rysunek 4.10 nie pokazuje domyślnych obiektów bazy danych SAMPLE ze względu na ograniczoną przestrzeń.



Rysunek 4.10 – Przestrzenie tablicowe o takiej samej nazwie w dwóch różnych bazach danych.

Gdy już raz utworzymy przestrzeń tablicową, możemy wewnątrz niej tworzyć obiekty, takie jak tabele, widoki i indeksy. Pokazuje to Rysunek 4.11.



Rysunek 4.11 – Tabele, widoki, indeksy utworzone wewnątrz przestrzeni tablicowej

## Laboratorium #2 – Tworzenie nowej bazy danych

### Cel

Podczas tego Laboratorium stworzysz nową bazę danych przy pomocy odpowiedniego kreatora narzędzia Centrum sterowania.

### Kroki

1. Z drzewa obiektów w Centrum sterowania wybierz prawym przyciskiem myszy folder *Wszystkie bazy danych*, zaznacz *Utwórz bazę danych* i wybierz *Z automatyczną konserwacją*. Uruchomi to Kreatora tworzenia bazy danych.
2. Określ nazwę bazy danych i jej ścieżkę na zakładce *Nazwa* kreatora. Użyj następujących wartości:  

Nazwa bazy danych:	EXPRESS
Domyślny dysk (Windows):	C:
Domyślna ścieżka: (Linux):	/home/db2inst1
Alias:	This will default to EXPRESS if left blank
Komentarz:	This is optional and can be left blank

Kliknij *Dalej*, aby przejść do następnej strony kreatora.
3. W zakładce *Pamięć* nie dokonuj żadnych zmian i kliknij *Dalej*.
4. W zakładce *Konserwacja* pozostaw to, co jest domyślnie wybrane i kliknij *Dalej*.
5. Określ okno czasowe do utrzymywania bazy w trybie off-line w zakładce *Parametry czasowe* kreatora. Wybierz dwie (lub więcej) godziny w tygodniu, kiedy DB2 będzie mogło dokonywać automatycznie zadań konserwujących, aby utrzymać poprawne działanie bazy danych. Teraz skonfiguruj okno tak, aby uruchamiało się o 1 w nocy od poniedziałku do czwartku na 6 godzin. Kliknij *Dalej*.
6. Skonfiguruj powiadomienia mailowe w zakładce *Serwer poczty*. DB2 może wysłać automatycznie e-mail, jeśli zostanie wykryty jakiś problem lub zagrożenie. Jeśli chcesz to skonfigurować wskaż dostępny serwer SMTP, którego DB2 ma używać do wysyłania maili. Nie mamy serwera SMTP w tym Laboratorium, więc pozostaw to pole puste i wybierz *Dalej*.
7. Sprawdź wybrane opcje w zakładce *Podsumowanie*. Kliknij *Zakończ*, aby rozpocząć proces tworzenia bazy danych. Stworzenie bazy danych zazwyczaj zajmuje kilka minut, podczas których wyświetlany jest pasek postępu.

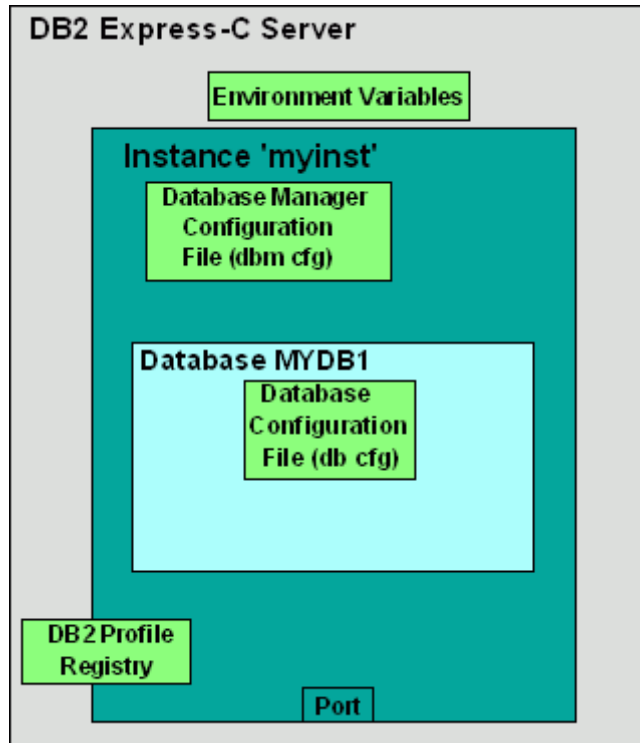
## 4.1 Konfiguracja DB2

Parametry DB2 mogą być konfigurowane przy użyciu narzędzia *Doradca konfigurowania*. Aby dostać się do tego narzędzia poprzez Centrum sterowania kliknij prawym przyciskiem na bazę danych i wybierz *Doradca konfigurowania*. Opierając się na Twoich odpowiedziach na pewne pytania dotyczące Twoich zasobów systemowych i obciążenia, Doradca konfigurowania dostarczy listę parametrów DB2, które powinny być zmienione wraz z proponowanymi wartościami. Jeśli chcesz poznać więcej szczegółów o konfiguracji DB2, powinieneś czytać dalej; w przeciwnym wypadku po użyciu Doradcy konfigurowania możesz rozpocząć pracę z DB2!

Serwer DB2 może być konfigurowany na czterech różnych poziomach:

- zmienne środowiskowe
- plik konfiguracyjny menedżera bazy danych
- plik konfiguracyjny bazy danych
- rejestr profilu DB2

Jest to również pokazane na Rysunku 4.12. Zwróć uwagę, gdzie dokładnie umieszczony jest który prostokąt. Na przykład, zmienne środowiskowe są umieszczone w systemie operacyjnym serwera, podczas gdy parametry pliku konfiguracyjnego menedżera bazy danych są umieszczone na poziomie instancji. Parametry konfiguracji bazy danych są umieszczone na poziomie bazy danych, natomiast rejestr profilu DB2 jest umieszczony albo na poziomie systemu operacyjnego albo na poziomie instancji.



Rysunek 4.12 – Konfiguracja DB2

#### 4.1.1 Zmienne środowiskowe

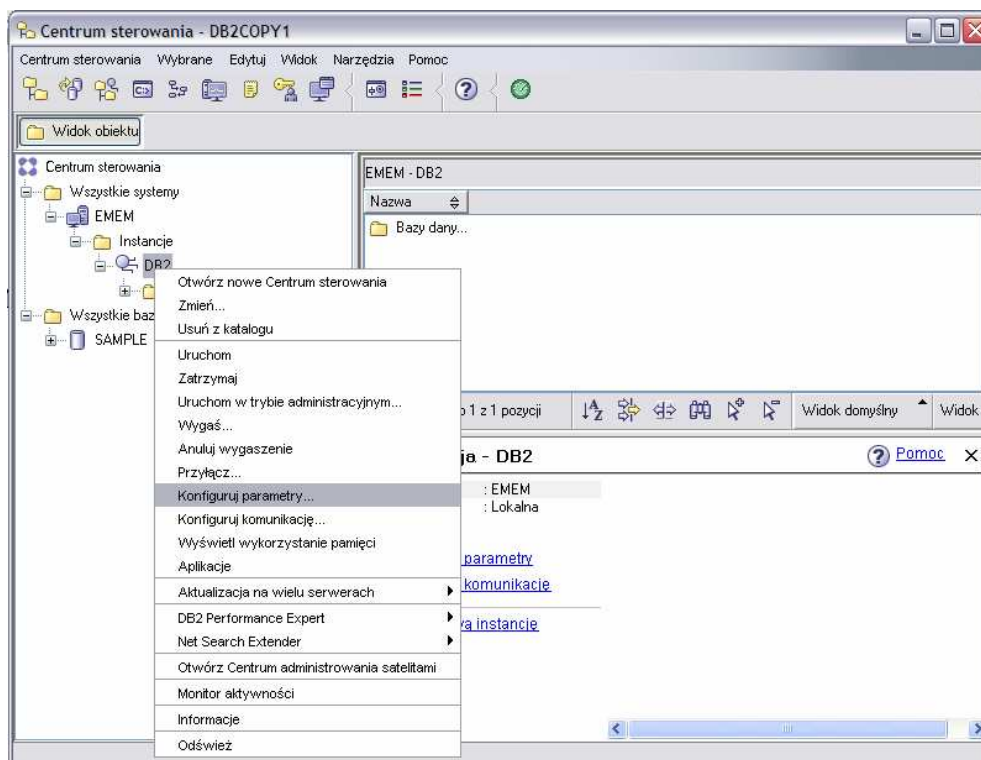
Zmienne środowiskowe są zmiennymi ustawianymi na poziomie systemu operacyjnego. Jedną ze zmiennych środowiskowych jest zmienna DB2INSTANCE. Wskazuje ona na bieżącą instancję, na której pracuje użytkownik i na rzecz której są wykonywane polecenia DB2. Na przykład, aby ustawić „myinst” jako bieżącą instancję na systemie Windows, należy użyć polecenia systemowego:

```
set dbinstance=myinst
```

#### 4.1.2 Plik konfiguracyjny menedżera bazy danych (dbm cfg)

Plik konfiguracyjny menedżera bazy danych zawiera parametry, które wpływają na instancję i wszystkie bazy danych przez nią zarządzane. Plik konfiguracyjny menedżera bazy danych może być przeglądany lub modyfikowany z poziomu wiersza komend lub przy użyciu Centrum sterowania DB2.

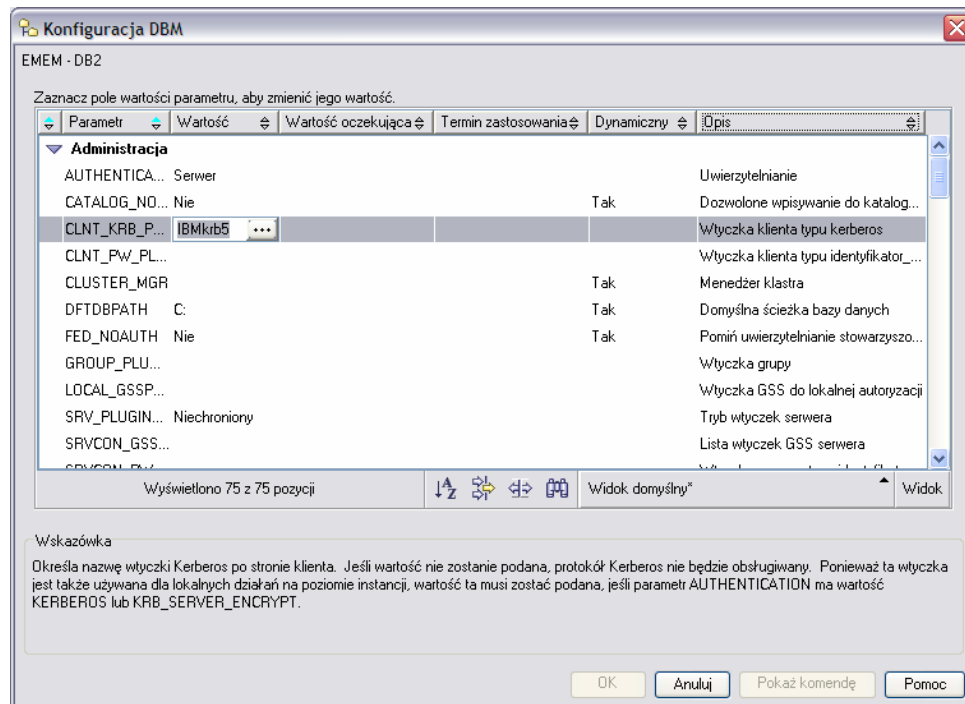
Aby z Centrum sterowania dostać się do pliku konfiguracyjnego menedżera bazy danych, wybierz daną instancję z folderu *Instancje*, a następnie z podręcznego menu wybierz *Konfiguruj parametry*. Pokazuje to Rysunek 4.13.



Rysunek 4.13 – Konfiguracja parametrów menedżera bazy z poziomu Centrum sterowania



Po wybraniu opcji *Konfiguruj parametry*, pokaże się ekran widoczny na Rysunku 4.14, wyświetlający listę parametrów.



**Rysunek 4.14 – Okno dialogowe konfiguracji menedżera bazy danych**

Wiele z parametrów jest dynamicznych, co oznacza, że ich zmiany są widoczne natychmiast, jednakże zmiany niektórych parametrów mogą wymagać zatrzymania i ponownego wystartowania instancji. Z wiersza komend można to wykonać za pomocą poleceń `db2stop` i `db2start`.

Zanim instancja zostanie zatrzymana, wszystkie aplikacje muszą zostać odłączone. Aby wymusić zatrzymanie instancji należy użyć polecenia `db2stop force`.

Instancja może być również zatrzymana z poziomu Centrum sterowania poprzez kliknięcie instancji i wybranie *Zatrzymaj* lub *Uruchom*.

Tabela 4.3 pokazuje wybrane komendy do zarządzania plikiem konfiguracyjnym menedżera bazy danych z poziomu wiersza komend.

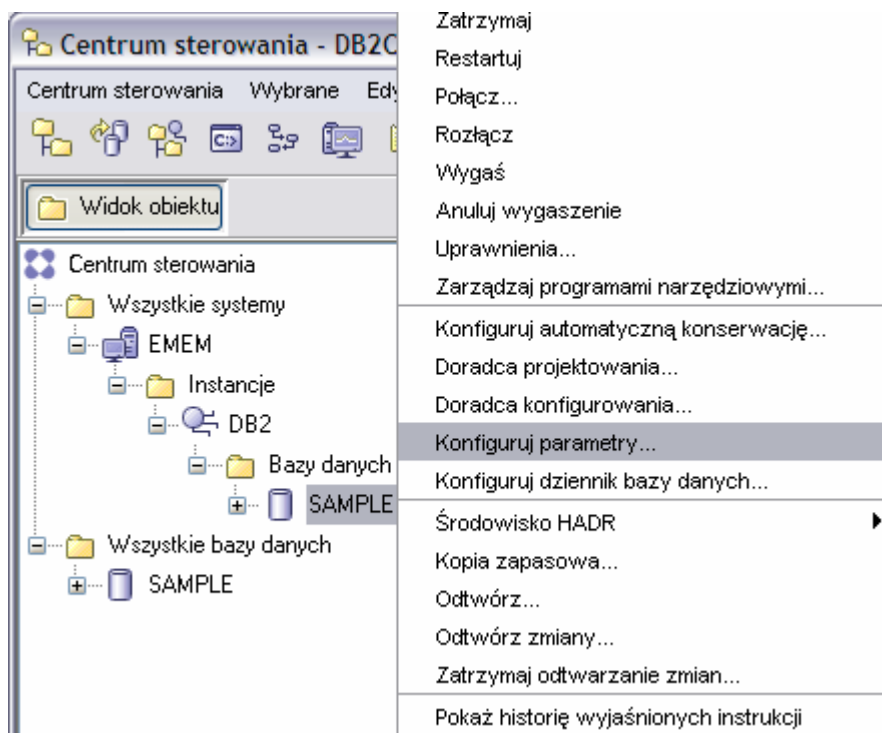
Komenda	Opis
<code>db2 get dbm cfg</code>	Pobiera informacje o konfiguracji menedżera bazy danych
<code>db2 update dbm cfg using &lt;parameter_name&gt; &lt;value&gt;</code>	Modyfikuje wartość parametru

**Tabela 4.3 – Polecenia wykorzystywane w konfiguratorze menedżera bazy danych**

### 4.1.3 Plik konfiguracyjny bazy danych (db cfg)

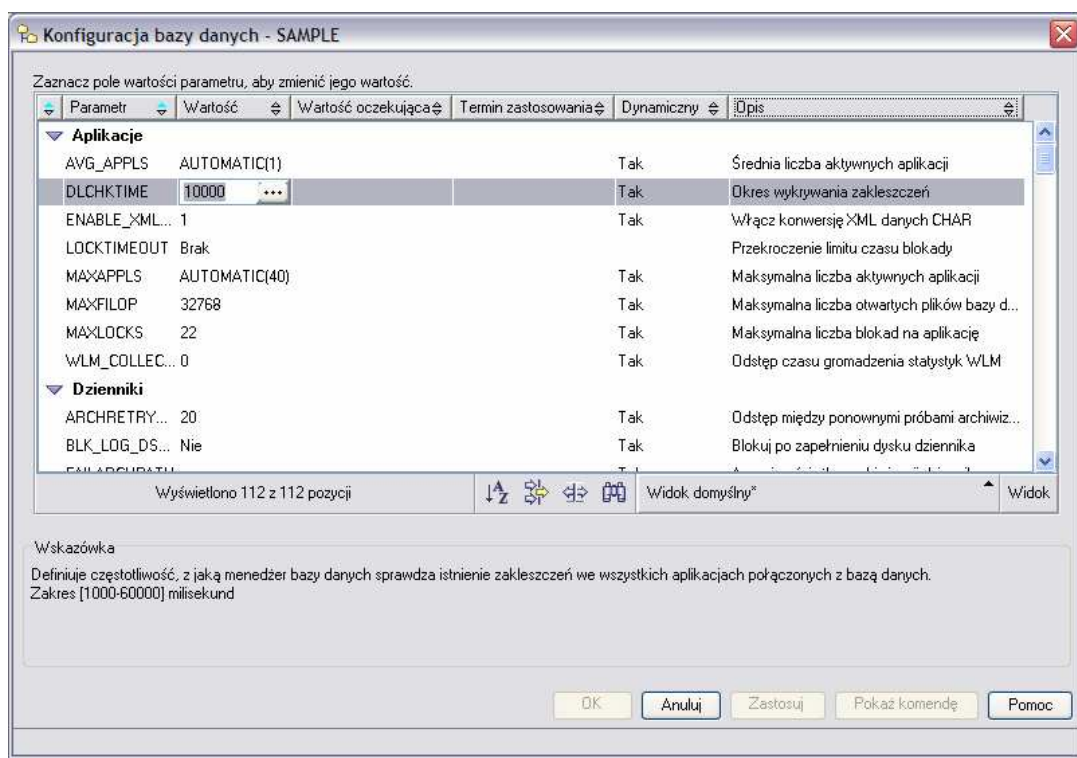
Plik konfiguracyjny bazy danych zawiera parametry, które wpływają bezpośrednio na bazę danych. Plik konfiguracyjny bazy danych może być przeglądany lub modyfikowany z poziomu wiersza komend lub przy pomocy Centrum sterowania.

Aby pracować z plikiem konfiguracyjnym bazy danych przy pomocy Centrum sterowania, należy wybrać bazę danych z folderu *Wszystkie bazy danych*, a następnie z podręcznego menu wybrać *Konfiguruj parametry*. Zostało to pokazane na Rysunku 4.15.



**Rysunek 4.15 – Konfiguracja pliku konfiguracyjnego bazy danych z poziomu Centrum sterowania**

Po wybraniu opcji *Konfiguruj parametry* pokaże się ekran ukazujący listę parametrów bazy danych, taki jak na Rysunku 4.16.



Rysunek 4.16 – Plik konfiguracyjny bazy danych

Tabela 4.4 pokazuje wybrane komendy przydatne do zarządzania plikiem konfiguracyjnym bazy danych z poziomu Wiersza komend.

Komenda	Opis
<code>get db cfg for &lt;database_name&gt;</code>	Pobiera informacje konfiguracyjne danej bazy
<code>update db cfg for &lt;database_name&gt; using &lt;parameter_name&gt; &lt;value&gt;</code>	Modyfikuje wartość parametru

Tabela 4.4 – Polecenia wykorzystywane do konfiguracji bazy danych

#### 4.1.4 Rejestr profilu DB2

Zmienne rejestrowe profilu DB2 zawierają parametry, które mogą być specyficzne dla konkretnej platformy systemowej oraz mogą być ustawiane globalnie (dla wszystkich instancji), lub na poziomie instancji (dotyczą wówczas jednej instancji).

Tabela 4.5 pokazuje wybrane komendy wykorzystywane do konfigurowania parametrów rejestru profilu DB2.

Komenda	Opis
<code>db2set -all</code>	Wyświetla wszystkie zmienne rejestrowe profilu DB2, które są w danym momencie ustawione
<code>db2set -lr</code>	Wyświetla wszystkie dostępne zmienne rejestrowe profilu DB2
<code>db2set &lt;parameter&gt;=&lt;value&gt;</code>	Ustawia wartość danemu parametrowi

**Tabela 4.5 – Polecenia wykorzystywane do ustawiania zmiennych rejestrowych profilu DB2**

Tabela 4.6 pokazuje najczęściej wykorzystywane zmienne rejestru profilu DB2

Zmienna rejestru	Opis
DB2COMM	Określa jaki menedżer komunikacyjny ma być uruchomiony wraz z uruchomieniem menedżera bazy danych
DB2_EXTSECURITY	Na systemie Windows, zapobiega nieautoryzowanemu dostępowi do DB2 poprzez zablokowanie plików systemowych DB2.
DB2_COPY_NAME	Przechowuje nazwę kopii DB2, która jest aktualnie używana Aby przełączyć się na inną zainstalowaną kopię DB2, należy wykorzystać polecenie: <code>install - path\bin\db2envvars.bat</code> . Ta zmienna nie może być wykorzystana do tego celu.

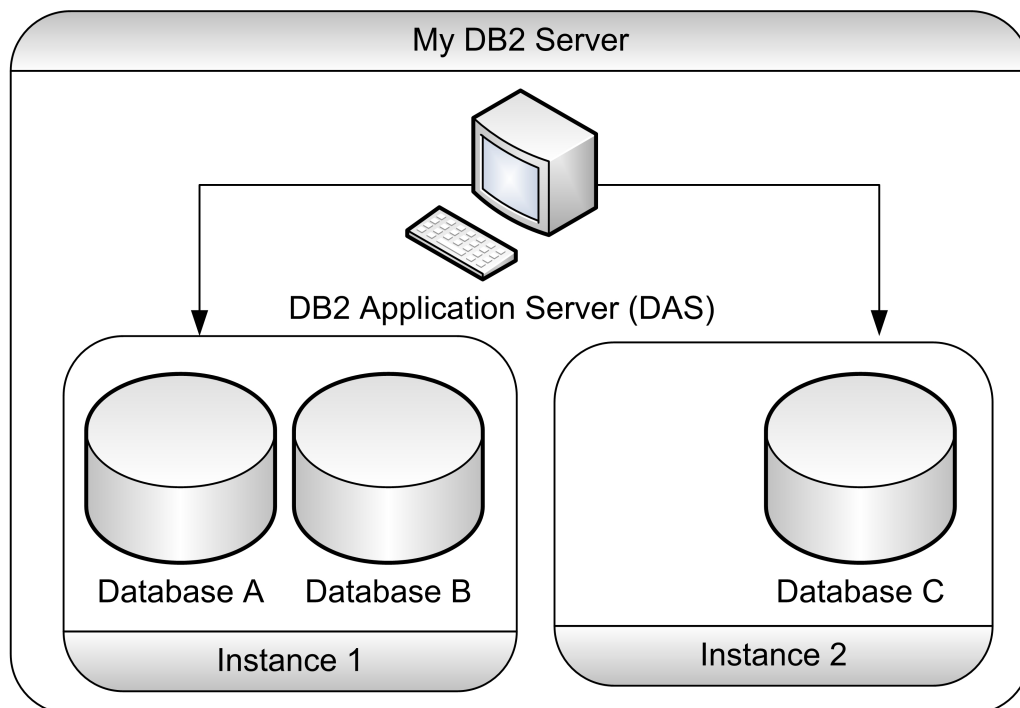
**Tabela 4.6 – Najczęściej wykorzystywane zmienne rejestru profile DB2**

Na przykład, aby umożliwić komunikację wykorzystującą protokół TCP/IP, należy ustawić zmienną rejestru DB2COMM na wartość TCPIP - tak, jak to zostało pokazane poniżej:

```
db2set db2comm=tcPIP
```

## 4.2 Serwer Administracyjny DB2

Serwer Administracyjny DB2 (ang. DB2 Administration Server, DAS) jest procesem, który działa na serwerze DB2, aby umożliwić zdalnym klientom zarządzanie serwerem DB2 za pomocą narzędzi graficznych. Istnieje tylko jeden DAS na każdym fizycznym komputerze, co zostało zobrazowane na Rysunku 4.16.



Rysunek 4.16 – Serwer Administracyjny DB2

## Laboratorium #3 – Praca z instancją, bazą danych i konfiguracja

### Cel

W tym Laboratorium utworzysz nową instancję, bazę danych i zmienisz parametry konfiguracyjne na serwerze DB2 na systemie Windows. Możesz to wykonać zarówno z poziomu Centrum sterowania, jak i z Okna komend. Pokażemy instrukcję wykorzystującą Okno komend.

### Kroki

1. Z Okna komend utwórz nową instancję o nazwie *newinst*.  
`db2icrt newinst`
2. W ramach nowej instancji *newinst* utwórz bazę danych *newdb* z domyślnymi wartościami:  
`set db2instance=newinst`  
`db2start`  
`db2 create database newdb`
3. Wyświetl wszystkie instancje dostępne na Twoim serwerze.  
`db2ilist`
4. Przełącz się na instancję *DB2* i upewnij się, że została rzeczywiście połączona.  
`set db2instance=db2`  
`db2 get instance`
5. Zmień parametr *FEDERATED* w pliku konfiguracyjnym menedżera instancji (`get dbm cfg`) z wartości *NO* na *YES* i sprawdź, czy zmiana została wprowadzona.  
`db2 update dbm cfg using FEDERATED YES`  
`db2 force applications all`  
`db2 terminate`  
`db2stop`  
`db2start`  
`db2 get dbm cfg`
6. Połącz się z bazą danych *SAMPLE* za pomocą użytkownika i hasła, którym jesteś zalogowany do systemu operacyjnego.  
`db2 connect to sample user <userID> using <psw>`
7. Sprawdź, ile aplikacji działa na Twojej bieżącej instancji.  
`db2 list applications show detail`

8. Otwórz kolejne Okno komend DB2 i ponownie połącz się z bazą danych SAMPLE, tym razem nie podając żadnych informacji o użytkowniku. Następnie sprawdź, ile jest nawiązanych połączeń.

```
db2 connect to sample
db2 list applications
```

9. Z Okna komend DB2 wymuś rozłączenie jednej z aplikacji.

```
db2 force application (<identyfikator aplikacji uzyskany za pomocą
komendy db2 list applications dla aplikacji o nazwie "db2bp.exe">)
```

10. Usuń instancję *newinst*.

```
db2idrop newinst
```

11. Usuń i ponownie utwórz DAS, uruchom go.

```
db2admin stop
db2admin drop
db2admin create
db2admin start
```

12. Ustaw zmienną rejestru DB2 DB2COMM na Twojej instancji na wartość *tcPIP* i *npipe*.

```
db2set db2comm=tcPIP,npipe
db2stop
db2start
```

13. Usuń ustawienie zmiennej DB2COMM.

```
db2set db2comm=
db2stop
db2start
```

14. Sprawdź aktualną wartość parametru konfiguracji bazy danych (db cfg) LOGSECOND, a następnie zmień tę wartość na 5 i zweryfikuj to ustawienie.

```
db2 connect to sample
db2 get db cfg
db2 update db cfg using LOGSECOND 5
db2 get db cfg
```

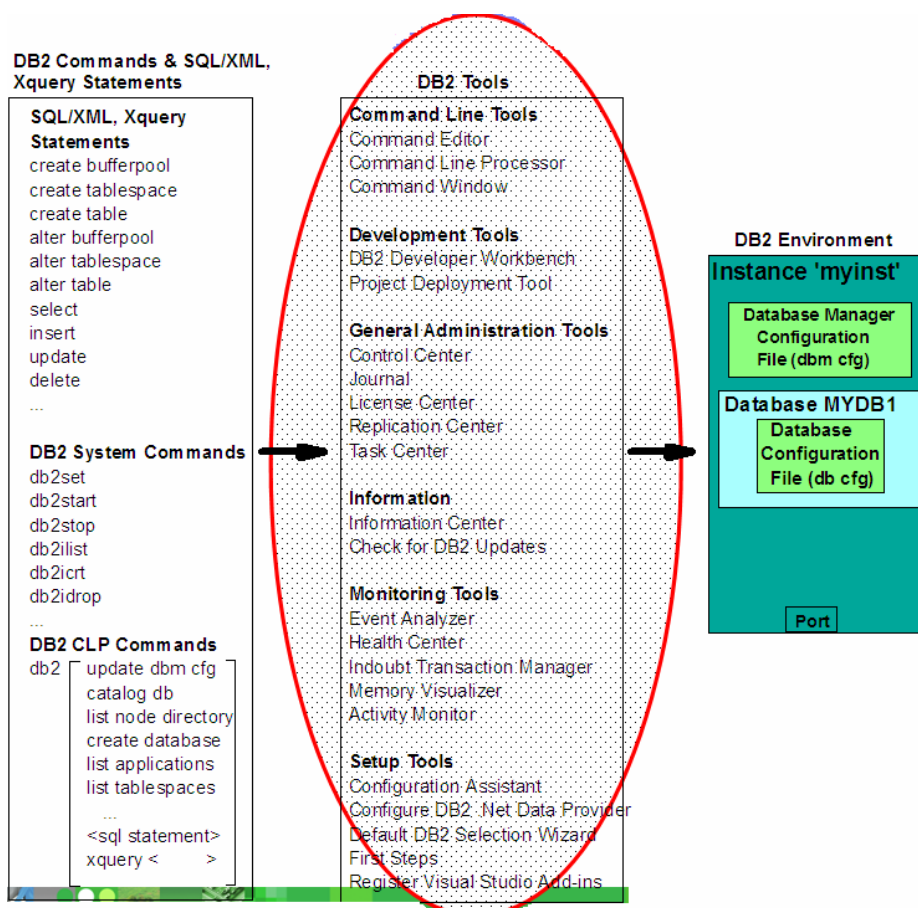




# 5

## Rozdział 5 – Narzędzia DB2

Rozdział ten opisuje wybrane narzędzia, jakie mogą być wykorzystywane w pracy z DB2. Czerwona elipsa na Rysunku 5.1 zaznacza, czym będziemy się zajmować w tym rozdziale.



Rysunek 5.1 – Narzędzia DB2

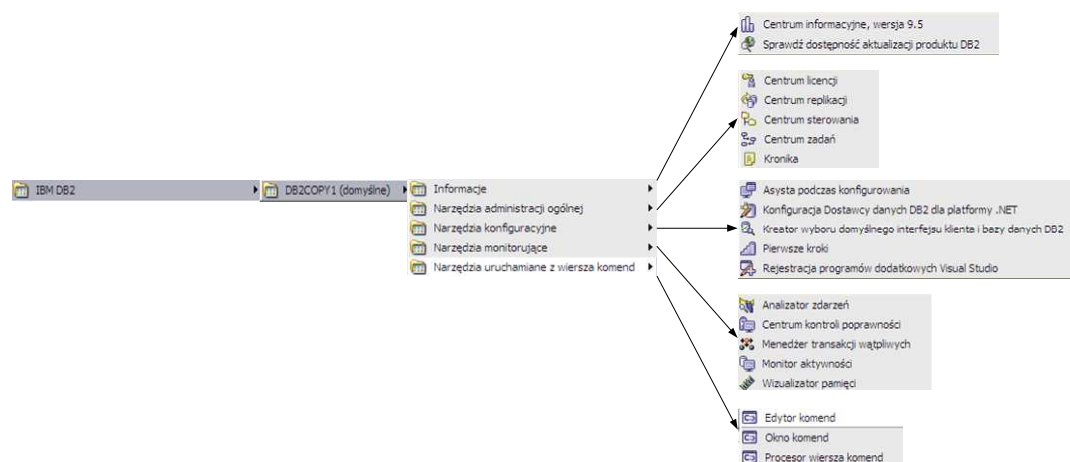
**Ważne:**

Obejrzyj prezentacje wideo dotyczące narzędzi DB2 i skryptowania znajdujące się pod następującymi linkami:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4202>

<http://www.channeldb2.com/video/video/show?id=807741:Video:4182>

Rysunek 5.2 wymienia wszystkie narzędzia DB2 dostępne ze skrótów menu Start - IBM DB2. Większość z nich jest taka sama zarówno na systemie Windows, jak i Linux.



**Rysunek 5.2 – Narzędzia DB2 z menu Start - IBM DB2**

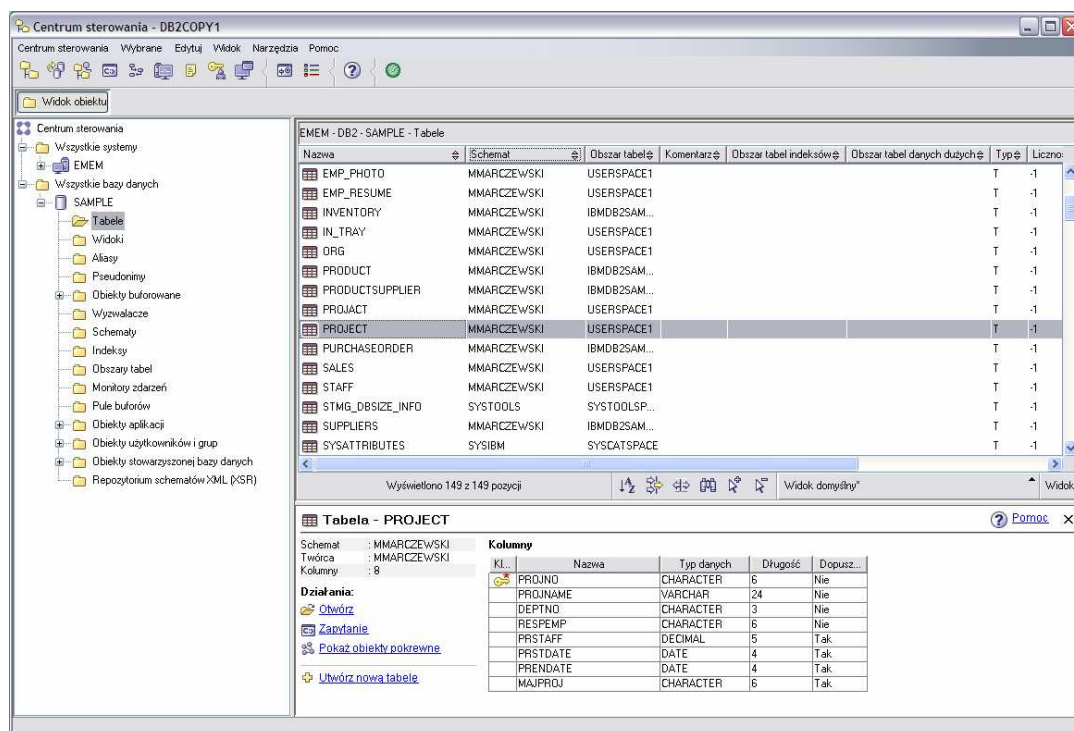
Tabela 5.1 pokazuje listę skrótów poleceń, które mogą być wykorzystane do uruchomienia najbardziej popularnych narzędzi DB2, zarówno na systemie Windows, jak i Linux.

Nazwa narzędzia	Komenda
Edytor komend (ang. Command Editor)	db2ce
Procesor wiersza komend (ang. Command Line processor)	db2
Okno komend (Command Window – tylko na Windows)	db2cmd
Centrum sterowania (ang. Control Center )	db2cc
Centrum zadań (ang. Task Center)	db2tc
Centrum kontroli poprawności (ang. Health Center)	db2hc
Asysta podczas konfigurowania (ang. Configuration Assistant)	db2ca
Pierwsze kroki (ang. First Steps)	db2fs

**Tabela 5.1 – Skróty poleceń uruchamiające niektóre z narzędzi DB2**

## 5.1 Centrum sterowania

Pierwszym narzędziem DB2 do zarządzania bazą danych jest Centrum Sterowania, które ilustruje Rysunek 5.3.



Rysunek 5.3 – Centrum sterowania DB2

Centrum sterowania jest scentralizowanym narzędziem, przy pomocy którego możemy:

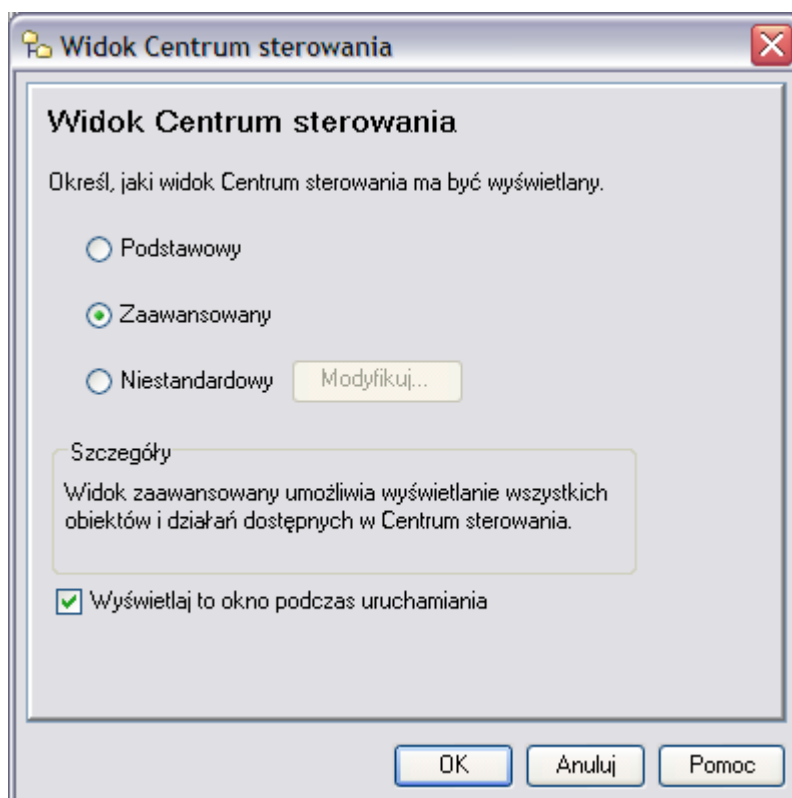
- zobaczyć system, instancje, bazy danych i obiekty bazy;
- tworzyć, edytować i zarządzać bazami danych i obiektami bazy;
- uruchamiać inne narzędzia graficzne DB2

Panel po lewej stronie pokazuje zwizualizowaną hierarchię obiektów baz danych w systemie, prezentując „folder” dla tabel, widoków itd. Po dwukrotnym kliknięciu na folder (na przykład Tabele, jak to zostało pokazane na Rysunku 5.3), górny panel po prawej stronie pokaże wszystkie powiązane obiekty, w tym wypadku, wszystkie tabele z powiązane z bazą danych SAMPLE. Jeśli wybierzemy tabelę w prawym górnym panelu, prawy dolny panel pokaże więcej szczegółowych informacji na jej temat.

Kliknięcie prawym przyciskiem na inny folder/obiekt w drzewie obiektów spowoduje wyświetlenie się menu odpowiedniego dla danego folderu/obiektu. Na przykład, kliknięcie prawym przyciskiem na instancję i wybranie *Konfiguruj parametry*, pozwoli na sprawdzenie i zmodyfikowanie pliku konfiguracyjnego menedżera bazy danych. Podobnie, jeśli klikniemy prawym przyciskiem na bazę danych i wybierzemy *Konfiguruj parametry*, będziemy mogli zobaczyć i zmodyfikować plik konfiguracyjny bazy danych. Środowisko

DB2 i parametry konfiguracji zostały omówione dokładniej w Rozdziale 4 „Środowisko DB2”.

Podczas pierwszego uruchomienia Centrum sterowania użytkownik jest pytany o wybór widoku, z jakiego chce korzystać. Wybór widoku wpływa na to, jakie opcje i obiekty bazy danych zostaną wyświetlone. Rysunek 5.4 pokazuje okno dialogowe do wyboru widoku w Centrum sterowania.



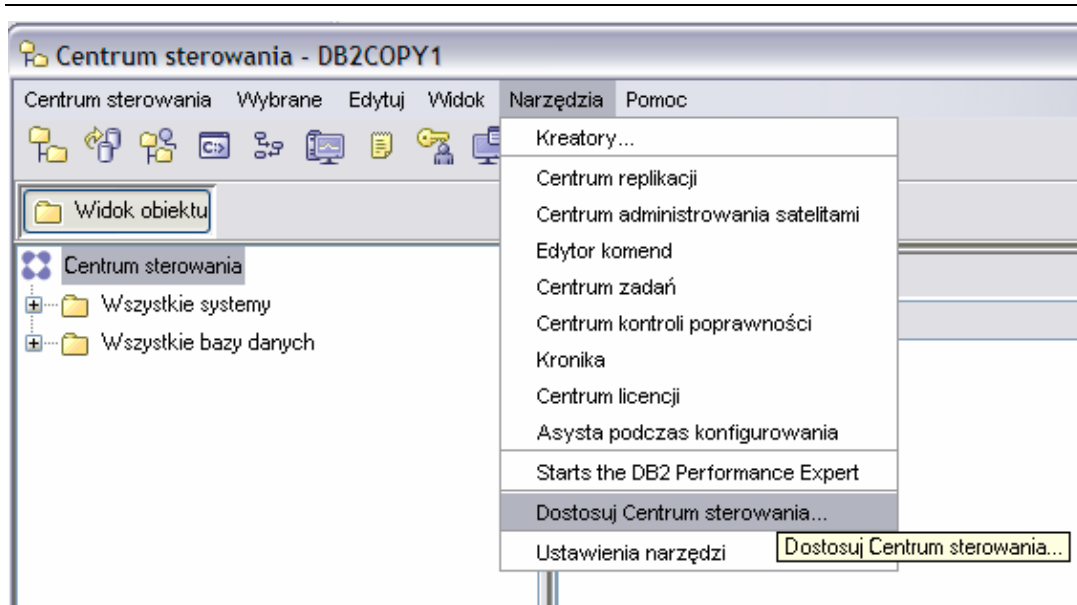
**Rysunek 5.4 – Okno dialogowe wyboru widoku w Centrum sterowania**

Podstawowy widok zapewnia wyświetlenie podstawowych funkcji DB2.

Zaawansowany widok pokazuje więcej opcji i funkcji.

Niestandardowy widok pozwala dostosować do potrzeb użytkownika wyświetlanie funkcji, opcji i obiektów, które można zobaczyć.


Aby ponownie wywołać okno dialogowe widoku w Centrum sterowania należy wybrać opcję *Dostosuj Centrum sterowania* z menu *Narzędzia*, tak jak zostało to pokazane na Rysunku 5.5.

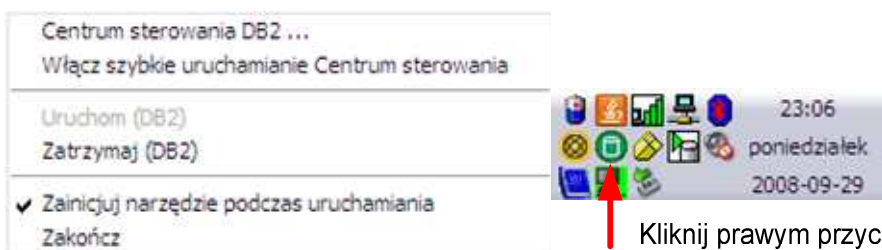


Rysunek 5.5 – Dostosowywanie Centrum sterowania

#### Uruchamianie Centrum sterowania

Istnieje wiele sposobów na uruchomienie Centrum sterowania:

- nawigacja poprzez menu *Start* systemu Windows,
- poprzez wykonanie polecenia `db2cc` z wiersza poleceń,
- poprzez kliknięcie ikony Centrum sterowania  w pasku narzędzi jakiegokolwiek innego graficznego narzędzia DB2,
- poprzez ikonę DB2 w zasobniku systemu Windows, tak jak zostało to pokazane na Rysunku 5.6 (kliknij prawym przyciskiem myszy na zieloną ikonę DB2 i wybierz Centrum sterowania z opcji menu).



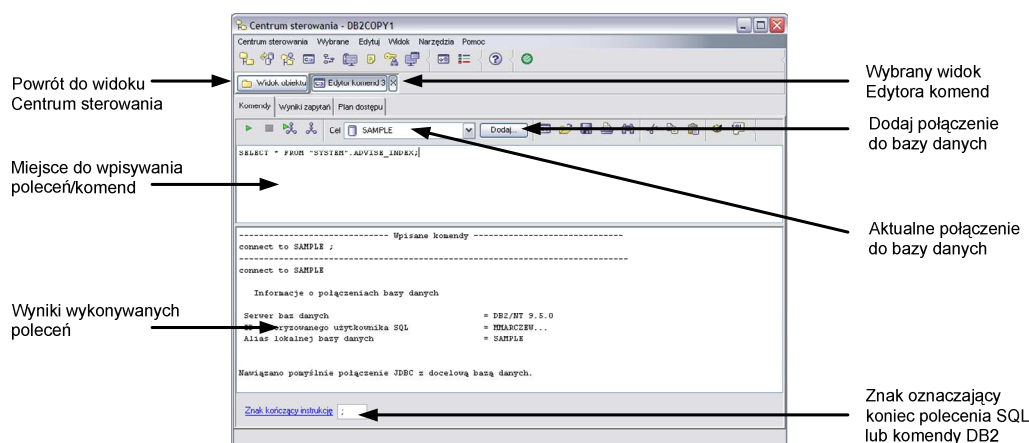
Kliknij prawym przyciskiem myszy na zieloną ikonę DB2 i wybierz z menu Centrum sterowania DB2

Rysunek 5.6 – Uruchamianie Centrum sterowania DB2 z zasobnika systemu Windows

## 5.2 Edytor komend

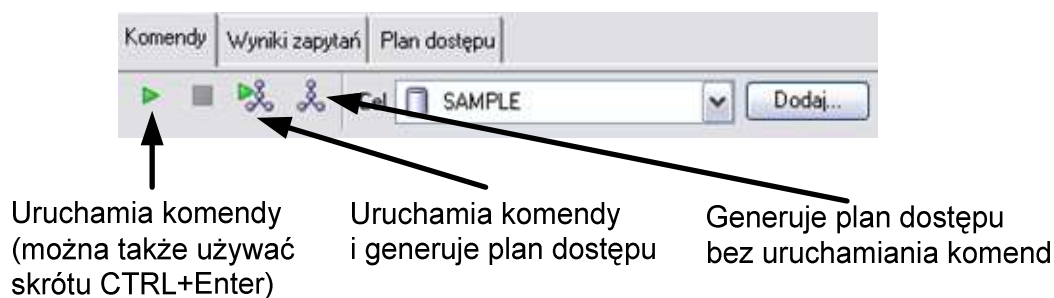
Używając Edytora komend DB2 (ang. Command Editor), można wykonywać polecenia DB2, zapytania SQL, wyrażenia XQuery, analizować plan wykonania zapytań oraz wyświetlać lub aktualizować wyniki zapytań.

Rysunek 5.7 pokazuje Edytor komend razem z opisami jego elementów.



Rysunek 5.7 – Edytor komend DB2

W obszarze wejściowym można wprowadzać wiele wyrażeń, pod warunkiem, że każde z nich będzie zakończone znakiem kończącym. Po naciśnięciu przycisku *Wykonaj* (patrz Rysunek 5.8) wyrażenia zostaną wykonane jedno po drugim. Jeśli zostanie zaznaczone wybrane wyrażenie, tylko ono zostanie wykonane. Aby uruchomić jakiegokolwiek zapytanie SQL, musi istnieć połączenie z bazą danych, jednakże jedno z wyrażeń może zawierać polecenie połączenia (CONNECT).




Rysunek 5.8 – Edytor komend – zakładka komendy

### Uruchamianie Edytora komend

Istnieje wiele sposobów na uruchomienie Edytora komend:

- ▶ poprzez menu Start systemu Windows:  
*Start -> Programy -> IBM DB2 -> DB2COPY1 -> Narzędzia uruchamiane z wiersza komend -> Edytor komend*
- ▶ poprzez uruchomienie polecenia db2ce
- ▶ poprzez menu Narzędzia w Centrum sterowania

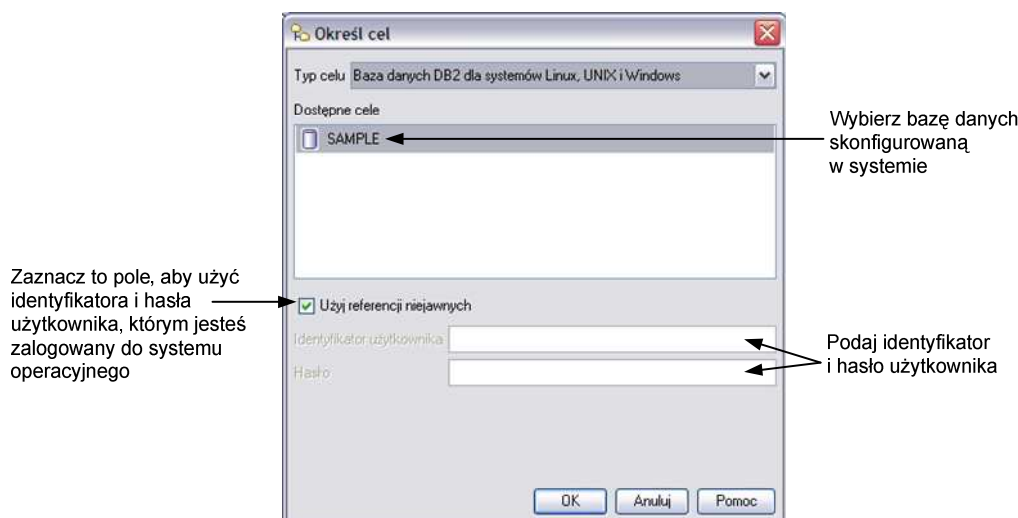
- ▶ poprzez Centrum sterowania
  - prawym przyciskiem myszy należy kliknąć ikonę bazy danych SAMPLE w drzewie obiektów w Centrum sterowania i wybrać z menu opcję *Zapytanie*
  - za każdym razem, kiedy zostanie wybrany obiekt, na którym można wykonać wyrażenia (baza danych, tabela, itd.) można uruchomić Edytor komend poprzez kliknięcie opcji *Zapytanie* w okienku szczegółów w Centrum sterowania.
- ▶ z Centrum sterowania poprzez kliknięcie ikony  w pasku narzędzi, tak jak zostało pokazane na Rysunku 5.9.



Rysunek 5.9 – Edytor komend, ikona w Centrum sterowania

### Dodawanie połączenia do bazy danych

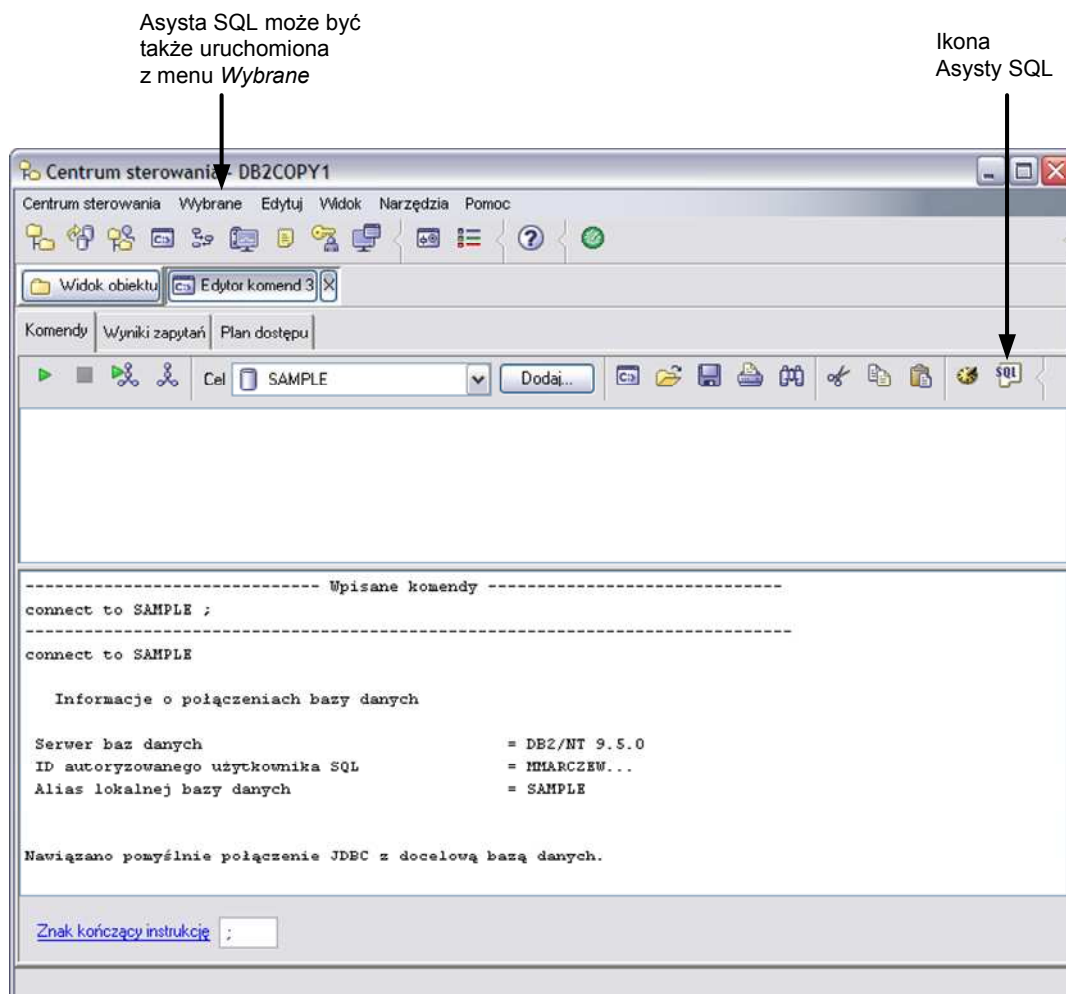
Aby dodać połączenie do bazy danych, należy kliknąć przycisk *Dodaj* – Rysunek 5.7. Wyświetli się okno dialogowe, takie jak zostało pokazane na Rysunku 5.10.



Rysunek 5.10 – Dodawanie połączenia do bazy danych

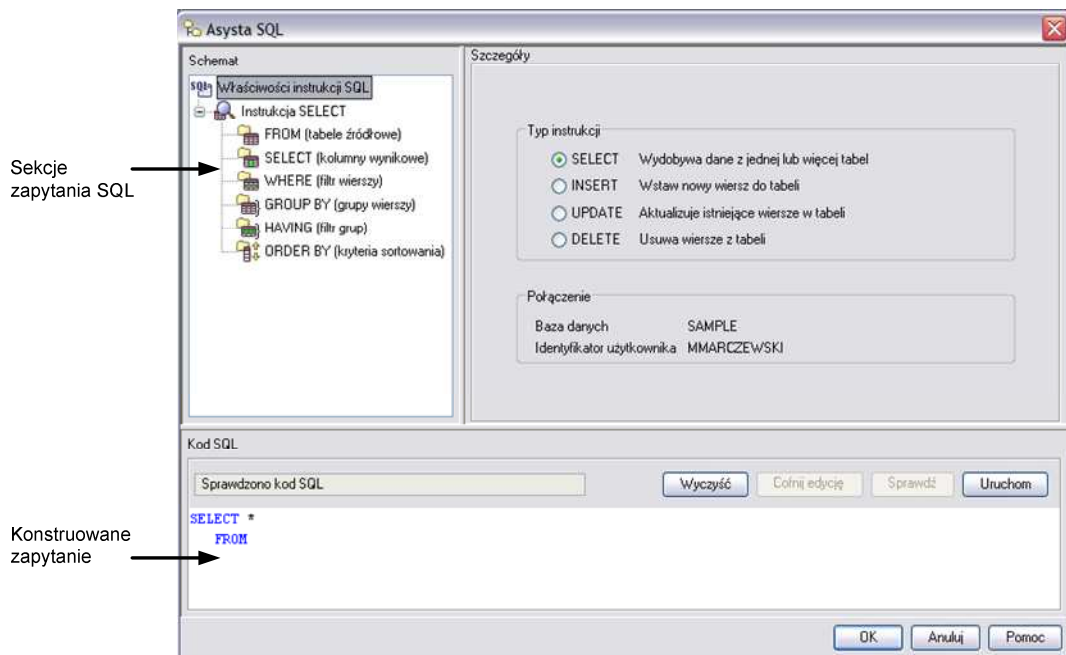
## 5.3 Asysta SQL

Jeśli ktoś nie jest zaznajomiony z językiem SQL i wolałby skorzystać z asystenta lub kreatora do generowania kodu SQL, to narzędzie Asysta SQL (ang. Assist Wizard) jest dostępne z poziomu Edytora komend jako pomoc. Tak, jak zostało to pokazane na Rysunku 5.11, można wywołać Asystę SQL z Edytora komend poprzez kliknięcie na ostatnią ikonę z symbolem SQL.

**Rysunek 5.11 – Uruchomienie Asysty SQL**

Rysunek 5.12 pokazuje asystenta do tworzenia zapytań SQL. Jest on bardzo prosty w obsłudze. Najpierw należy wskazać typ zapytania SQL, jakiego potrzebujemy (SELECT, INSERT, UPDATE, DELETE). W zależności od tego wyświetlą się inne opcje. W dolnej części okna zobaczymy, jak jest konstruowane zapytanie SQL, po wybraniu różnych opcji kreatora.

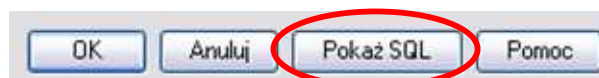




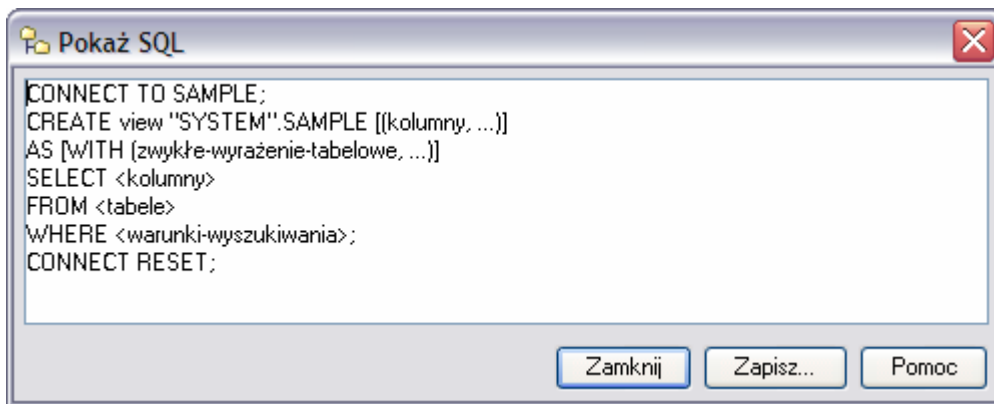
Rysunek 5.12 – Asystent do tworzenia zapytań SQL

#### 5.4 Przycisk Pokaż SQL

Większość narzędzi graficznych i kreatorów w DB2 pozwala na podgląd aktualnie wykonywanego zapytania SQL, które zostało utworzone w wyniku używania tego narzędzia lub kreatora. W celu podejrzenia kodu SQL, należy kliknąć przycisk Pokaż SQL w narzędziu, z którym w danej chwili pracujemy, tak jak zostało to pokazane na Rysunkach 5.13 i 5.14.



Rysunek 5.13 – Przycisk Pokaż SQL



**Rysunek 5.14 – Efekt kliknięcia przycisku Pokaż SQL**

Możliwość podglądu wyrażeń i zapytań SQL jest bardzo przydatna do nauki składni języka SQL, a także do zapisywania poleceń lub wyrażeń do pliku w celu ich późniejszego wykorzystania. Można również budować skrypty poprzez ponowne użycie już wygenerowanych poleceń i wyrażeń.

## Laboratorium #4: Wypełnianie bazy danych przy użyciu skryptów

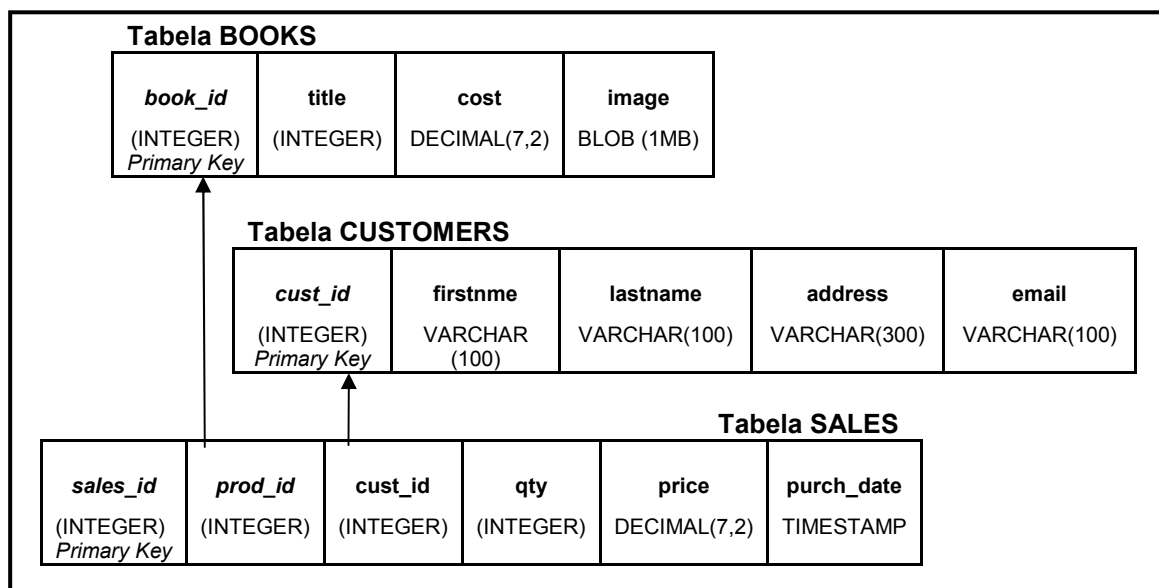
### Cel

W tym Laboratorium dowiesz się, jak wypełnić bazę danych EXPRESS przy użyciu Edytora komend i dwóch dostarczonych skryptów.

### Kroki

1. W Laboratorium #2 została utworzona baza danych EXPRESS, teraz potrzebujemy ją wypełnić kilkoma tabelami i jakimiś danymi. Dla Twojej wygody, dwa skrypty nazwane `quicklab4.db2` i `quicklab4.dat` zostały już utworzone. Skrypt `quicklab4.db2` zawiera polecenia do tworzenia tabel i dlatego musi zostać uruchomiony jako pierwszy. Skrypt `quicklab4.dat` zawiera wyrażenia wstawiające dane do tabel. Oba skrypty można znaleźć w archiwum `quicklabs.zip` dołączonym do tej książki. Aby uruchomić skrypty, uruchom Edytor komend. Upewnij się, że nowoutworzona baza danych została wybrana z rozwijanej listy w pasku narzędzi. Jeśli nowa baza danych nie pojawi się w liście, dodaj połączenie do niej używając przycisku *Dodaj*.
2. Kliknij *Wybrane* → *Otwórz* w menu Edytora komend i przejdź do folderu, w którym zapisane są skrypty. Wybierz plik `quicklab4.db2` i kliknij *OK*. Zawartość pliku powinna się teraz wyświetlić w obszarze wejściowym Edytora komend. Kliknij *Wykonaj*, aby uruchomić skrypt. Sprawdź, czy nie wystąpił żaden błąd podczas wykonywania skryptu.
3. Powtórz krok (2) dla pliku `quicklab4.dat`

Utworzona przez Ciebie baza danych jest przeznaczona dla bardzo prostej księgarni internetowej. Tabela BOOKS zawiera wszystkie informacje o książkach, które posiada księgarnia. Tabela CUSTOMERS zawiera informacje o każdym kliencie sklepu. W końcu, tabela SALES zawiera dane sprzedaży. Za każdym razem, gdy klient kupuje książkę, wszystkie dane zapisywane są w tabeli SALES. Poniższy diagram przedstawia projekt bazy i relacje między tabelami.



## 5.5 Skryptowanie

Zawsze bardzo przydatna jest możliwość tworzenia plików skryptowych, które zawierają wiele poleceń DB2 lub wyrażeń SQL, a które chcemy wielokrotnie wykorzystać. Na przykład, gdy chcemy aby DB2 uruchamiało dany skrypt codziennie, aby sprawdzić ilość rekordów w ważnej tabeli.

Są dwie podstawowe formy pisania skryptów:

1. Skrypty SQL
2. Skrypty systemu operacyjnego (powłoki).

### 5.5.1 Skrypty SQL

Skrypty SQL zawierają wyrażenia, zapytania i polecenia dotyczące bazy danych. Skrypty te są proste do zrozumienia i niezależne od platformy. Zmienne i parametry nie są jednak wspierane.

Na przykład, następujące polecenia zostały zapisane w pliku nazwanym `script1.db2`.

```
CONNECT TO EXPRESS;
CREATE TABLE user1.mytable
    ( col1 INTEGER NOT NULL,
      col2 VARCHAR(40),
      col3 DECIMAL(9,2) );
SELECT * FROM user1.mytable FETCH FIRST 10 ROWS ONLY;
COMMIT;
```

#### Plik `script1.db2`

W powyższym skrypcie wszystkie wyrażenia są wyrażeniami SQL, a każde z nich jest oddzielone separatorem, w tym wypadku średnikiem. Nazwa pliku nie musi korzystać rozszerzenia rozszerzenia „db2”. Dowlone rozszerzenie może zostać użyte.

#### Wykonywanie skryptów SQL

Skrypty SQL mogą być wykonywane zarówno z Edytora komend, jak i z Okna komend DB2 na systemie Windows, czy poprzez powłokę Linux. Aby uruchomić poprzedni skrypt z Okna komend DB2 lub z powłoki Linux, należy wykorzystać następujące polecenie:

```
db2 -t -v -f script1.db2 -z script1.log
lub:
db2 -tvf script1.db2 -z script1.log
```

W tym poleceniu:

- t oznacza, że wyrażenie używa domyślnego znaku końca wyrażenia (średnika)
- v oznacza tzw. tryb mówiący - wykonywane komendy są wyświetlane na ekranie
- f oznacza, że nazwa pliku podana za tą flagą jest nazwą pliku skryptu.

-z oznacza, że komunikaty o błędach powinny być zapisywane do pliku o nazwie podanej po parametrze; powinien być on użyty do późniejszej analizy (jest to opcjonalne, ale zalecane).

Jeżeli flaga -t zostanie użyta, a znak końca wyrażenia nie zostanie określony, zakłada się że takim znakiem jest średnik. Mogą być sytuacje, w których żaden znak końca wyrażenia nie jest wymagany. Na przykład skrypt zawierający kod SQL PL musi używać innego znaku końca wyrażenia niż domyślny (średnik), ponieważ średniki są tam używane w definicjach obiektów w kodzie SQL PL na końcu poleceń wewnątrz procedur.

Na przykład, w poniższym skrypcie `functions.db2`, który zawiera polecenia „Data Definition Language” (DDL) potrzebne do utworzenia funkcji, średnik musi wystąpić na końcu wyrażenia `SELECT`, użytego wewnątrz tworzonej funkcji. Na końcu wyrażenia `CREATE FUNCTION`, jako znak końca wyrażenia, został wykorzystany wykrzyknik (!). Jeśli w tym miejscu użyty zostałby średnik, spowodowałby on konflikt podczas wykonania skryptu, kończący się wystąpieniem błędu zgłoszonego przez DB2.

```
CREATE FUNCTION f1 ()
    SELECT ... ;
...
END!
```

#### **Plik `functions.db2`**

Aby poinformować DB2, że będzie wykorzystywany inny znak końca wyrażen, należy użyć flagi -d, po której wypisany zostanie żądany znak końca wyrażenia, tak jak zostało to pokazane poniżej:

```
db2 -td! -v -f functions.db2 -z functions.log
```

Opis pozostałych flag, z jakich można korzystać, może być wyświetlony z Okna komend lub z powłoki Linux za pomocą polecenia:

```
db2 list command options
```

### **5.5.2 Skrypty (powłoki) systemu operacyjnego**

Skrypty systemu operacyjnego dają większą elastyczność i udogodnienia, takie jak dołączenie dodatkowej logiki programistycznej. Są one zależne od platformy systemowej, ale wspierają parametry i zmienne. Poniżej przedstawiamy przykład prostego skryptu operacyjnego (powłokowego) dla systemu operacyjnego Windows.

```
set DBPATH=c:
set DBNAME=PRODEXPR
set MEMORY=25
```

```
db2 CREATE DATABASE %DBNAME% ON %DBPATH% AUTOCONFIGURE USING
      MEM_PERCENT %MEMORY% APPLY DB AND DBM
db2 CONNECT TO %DBNAME% USER %1 USING %2
del schema.log triggers.log app_objects.log
db2 set schema user1
db2 -t -v -f schema.db2 -z schema.log
db2 -td@ -v -f triggers.db2 -z triggers.log
db2 -td@ -v -f functions.db2 -z functions.log
```

**Plik create\_database.bat**

Aby wykonać ten skrypt systemu operacyjnego z wiersza poleceń, należy użyć następującego polecenia na Windows:

```
create_database.bat db2admin ibmdb2
```

Na platformie Windows rozszerzenie „.bat” mówi systemowi operacyjnemu, że jest to plik wykonywalny.

Na platformie Linux należy zmienić typ pliku aby zaznaczyć, że jest to plik wykonywalny, poprzez polecenie `chmod +x`. Następnie można go uruchomić w taki sam sposób, jak opisano powyżej.

## Laboratorium #5: Tworzenie skryptu do instalacji bazy danych

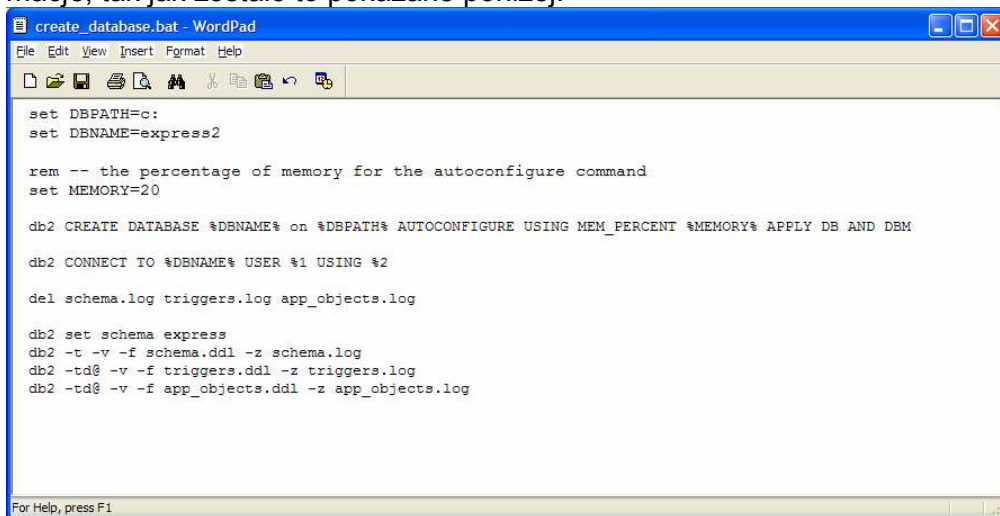
### Cel

Skrypty to potężny mechanizm do wykonywania powtarzających się zadań, takich jak zbieranie statystyk baz danych, tworzenie kopii zapasowych czy tworzenie i odtwarzanie bazy danych. Skrypty systemu operacyjnego mają tę zaletę, że wspierają parametry, co powoduje, że są bardziej elastyczne.

W tym Laboratorium stworzysz skrypt systemu operacyjnego umieszczający bazę EXPRESS jako bazę danych o nazwie EXPRESS2. Skrypt Twój wywoła wcześniej wygenerowany skrypt SQL do tworzenia obiektów bazy danych. W celu zaoszczędzenia miejsca, w Laboratorium pokażemy skrypty i polecenia dotyczące platformy Windows. Jeśli wolisz pracować z systemem Linux upewnij się, że dokonane zostaną odpowiednie zmiany w poniższych poleceniach.

### Kroki

1. Otwórz dowolny edytor tekstu, taki jak Notatnik lub Wordpad i wpisz informacje, tak jak zostało to pokazane poniżej:



```
set DBPATH=c:
set DBNAME=express2

rem -- the percentage of memory for the autoconfigure command
set MEMORY=20

db2 CREATE DATABASE %DBNAME% on %DBPATH% AUTOCONFIGURE USING MEM_PERCENT %MEMORY% APPLY DB AND DBM

db2 CONNECT TO %DBNAME% USER %1 USING %2

del schema.log triggers.log app_objects.log

db2 set schema express
db2 -t -v -f schema.ddl -z schema.log
db2 -td@ -v -f triggers.ddl -z triggers.log
db2 -td@ -v -f app_objects.ddl -z app_objects.log
```

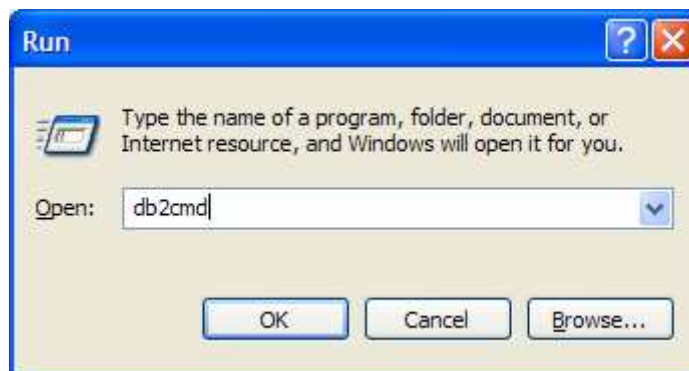
2. Zapisz skrypt w pliku o nazwie `create_database.bat`. W okienku dialogowym *Zapisz jako* upewnij się, że wybrałeś opcje *MS-DOS Format* (w Wordpadzie). Jeśli zapiszesz plik w innym formacie, Wordpad może dodać niewidoczne znaki, które mogą wywołać problemy podczas wykonywania skryptu. Ponadto należy umieścić nazwę pliku w cudzysłowie (aby Windows nie dołączył rozszerzenia `.txt`), tak jak zostało to pokazane na rysunku poniżej.





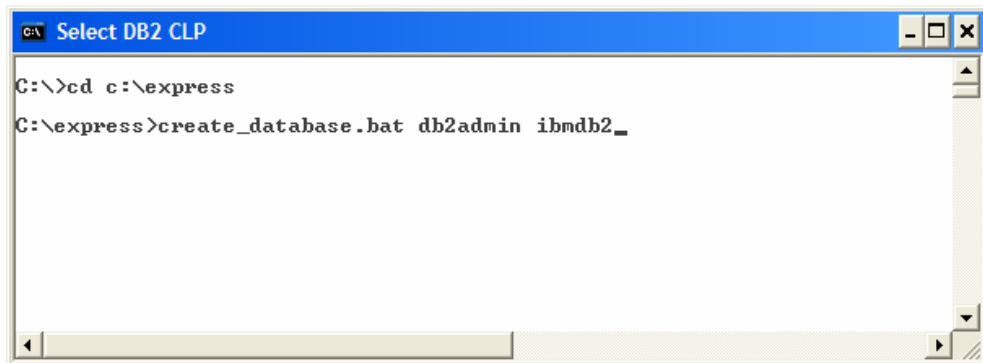
3. Aby uruchomić skrypt, który współpracuje z DB2, należy otworzyć środowisko wiersza komend DB2. Aby otworzyć Okno komend DB2, przejdź do menu *Start > Programy > IBM DB2 > DB2COPY1 (domyślne) > Narzędzie uruchamiane z wiersza komend > Okno komend*.

Alternatywnie, możesz użyć *Start > Uruchom*, wpisując `db2cmd` i naciskając OK, tak jak to pokazano poniżej.



4. Następnie uruchom skrypt wpisując polecenia:

```
cd C:\express  
create_database.bat db2admin ibmdb2
```



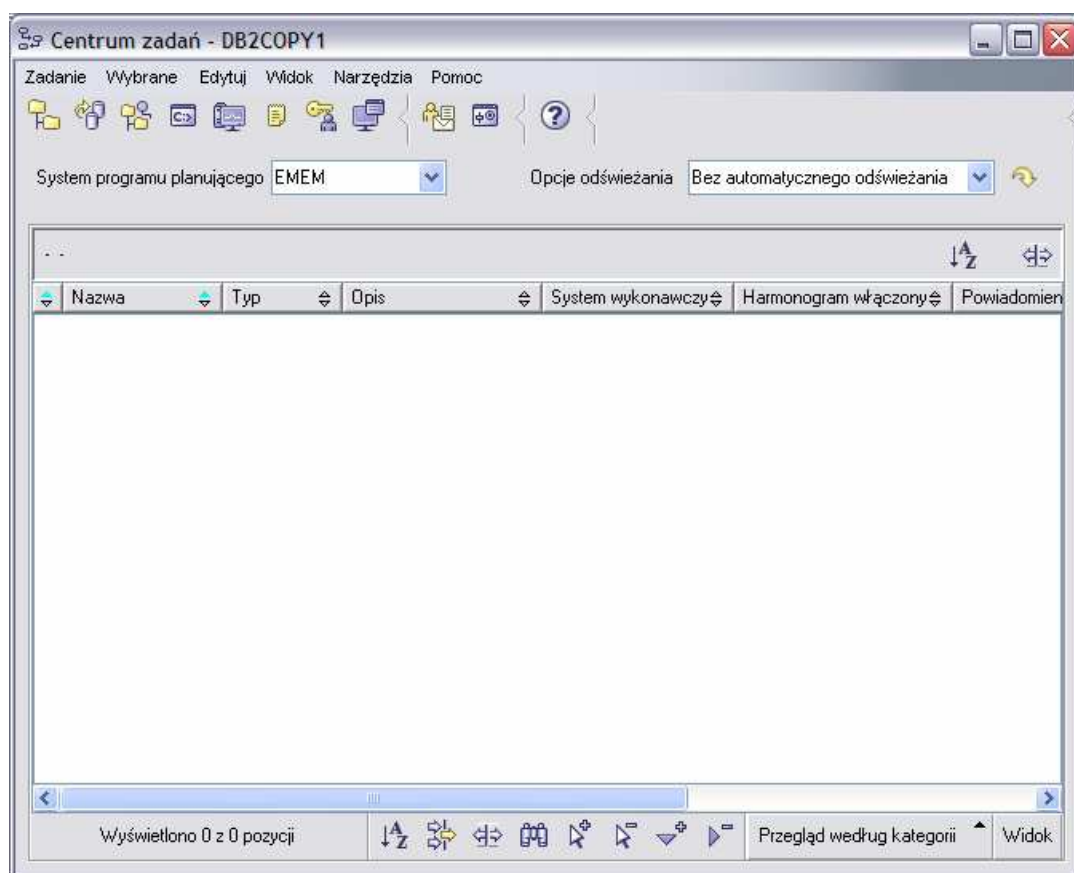
5. Przez moment przyjrzyj się skryptowi, który właśnie został utworzony. Czy rozumiesz co się dzieje w każdej linii?

6. Spróbuj odpowiedzieć na poniższe pytania:

- a. W którym miejscu zostaje utworzone połączenie z bazą danych?
- b. Co oznacza zapis %1 i %2?
- c. Co robi następująca linia kodu? Gdzie jest używana? Po co?  
SET DBPATH=C:
- d. Co robi następująca linia kodu?  
del schema.log, triggers.log, app\_objects.log
- e. Co się wydarzy, jeśli skrypt zostanie wywołany bez parametrów?
- f. Dlaczego wywoływany skrypt nie zawiera wyrażenia CONNECT TO? Jak łączy się z bazą danych?

## 5.6 Centrum zadań

Narzędzie graficzne Centrum zadań (ang. Task Center) pozwala na tworzenie zadań - zestawów operacji, takich jak polecenia DB2, polecenia systemu operacyjnego czy skrypty. Kolejne działania mogą być wykonywane, jeśli zadanie się powiedzie lub nie. Na przykład, jeśli zadanie wykonujące kopię zapasową ważnej bazy danych o 3:00 rano zakończy się sukcesem, do administratora bazy danych może zostać wysłany e-mail, w celu przekazania tej informacji. Z drugiej strony, jeśli wykonanie kopii się nie powiedzie, Centrum zadań może zawiadomić o tym administratora. Rysunek 5.15 pokazuje Centrum zadań.



Rysunek 5.15 – Centrum zadań

### 5.6.1 Katalog narzędzi bazy danych

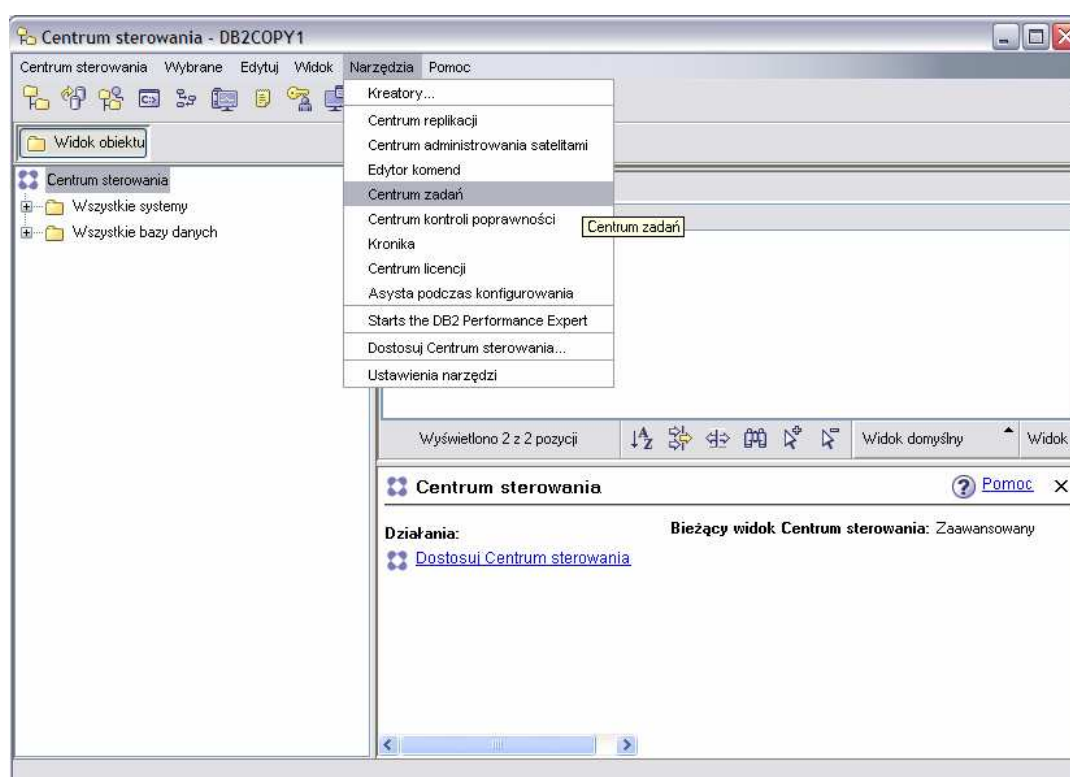
Wszystkie szczegóły dotyczące zadań i harmonogramu zadań są zapisane w oddzielnej bazie danych DB2 nazywanej bazą katalogu narzędzi (ang. Tools Catalog). Baza ta musi istnieć już wcześniej, aby móc harmonogramować zadania. Aby utworzyć bazę danych katalogu narzędzi można wykorzystać następujące polecenie:

```
CREATE TOOLS CATALOG systools CREATE NEW DATABASE toolsdb
```

W powyższym przykładzie `sysools` jest nazwą schematu dla wszystkich tabel w bazie danych, natomiast baza danych nazywa się `toolsdb`. O schematach będziemy mówić więcej w Rozdziale 8 „Praca z obiektami bazy danych”.

### Uruchamianie Centrum zadań

Centrum zadań można uruchomić klikając na menu *Narzędzia > Centrum zadań*, tak jak zostało to pokazane na Rysunku 5.16. Alternatywnie, można uruchomić to narzędzie z menu Start w Windows: *Start > Programy > IBM DB2 > DB2COPY > Narzędzie administracji ogólnej > Centrum zadań*.



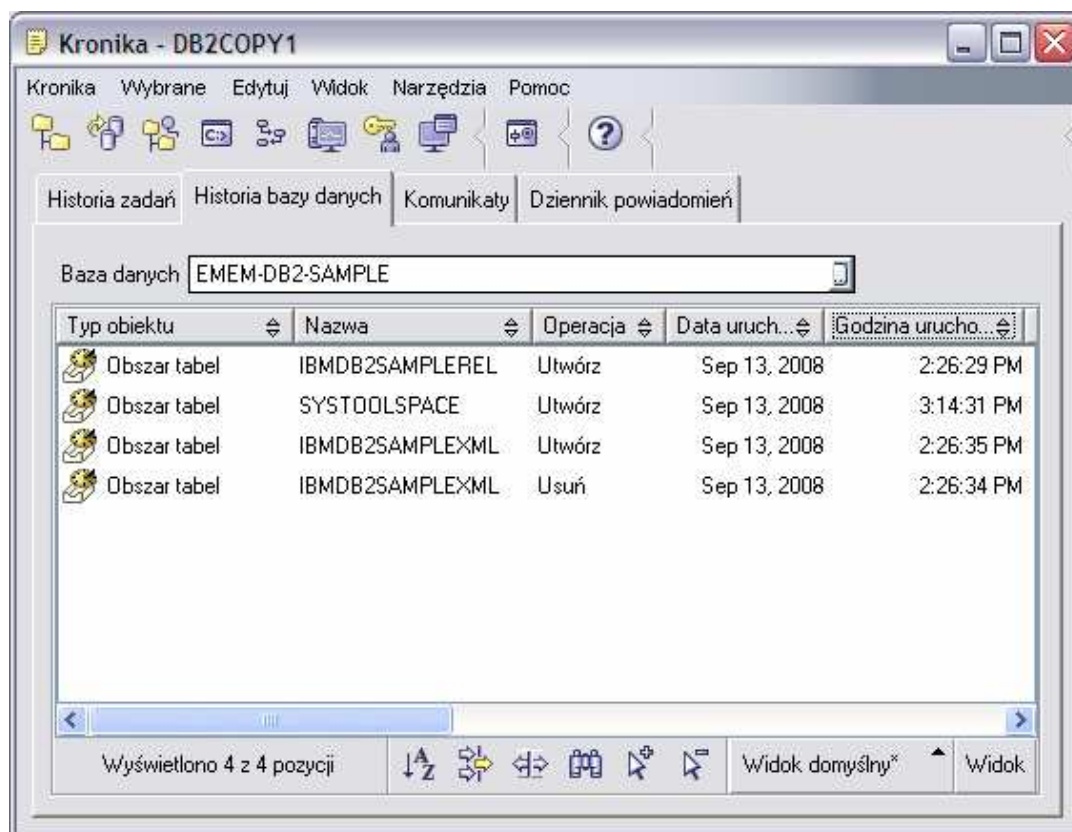
Rysunek 5.16 – Uruchamianie Centrum zadań

### Harmonogramowanie przy użyciu Centrum Zadań (ang. Task Center)

Każdy typ skryptów może być harmonogramowany przy użyciu Centrum zadań, obojętnie czy został utworzony przy pomocy narzędzi graficznych DB2, czy nie. Zadania te są uruchamiane zgodnie z planem, wg czasu systemu operacyjnego, w którym został utworzony katalog narzędzi bazy danych. Zachęcamy Cię do samodzielnego zapoznania się z Centrum zadań. Tworzenie zadań jest bardzo proste.

## 5.7 Kronika

Narzędzie graficzne Kronika (ang. Journal) pozwala administratorowi DB2 na wgląd do dziennika aktywności w trybie on-line. Rysunek 5.17 pokazuje Kronikę. Tabela 5.2 pokazuje informacje, jakie można uzyskać za pomocą Kroniki.



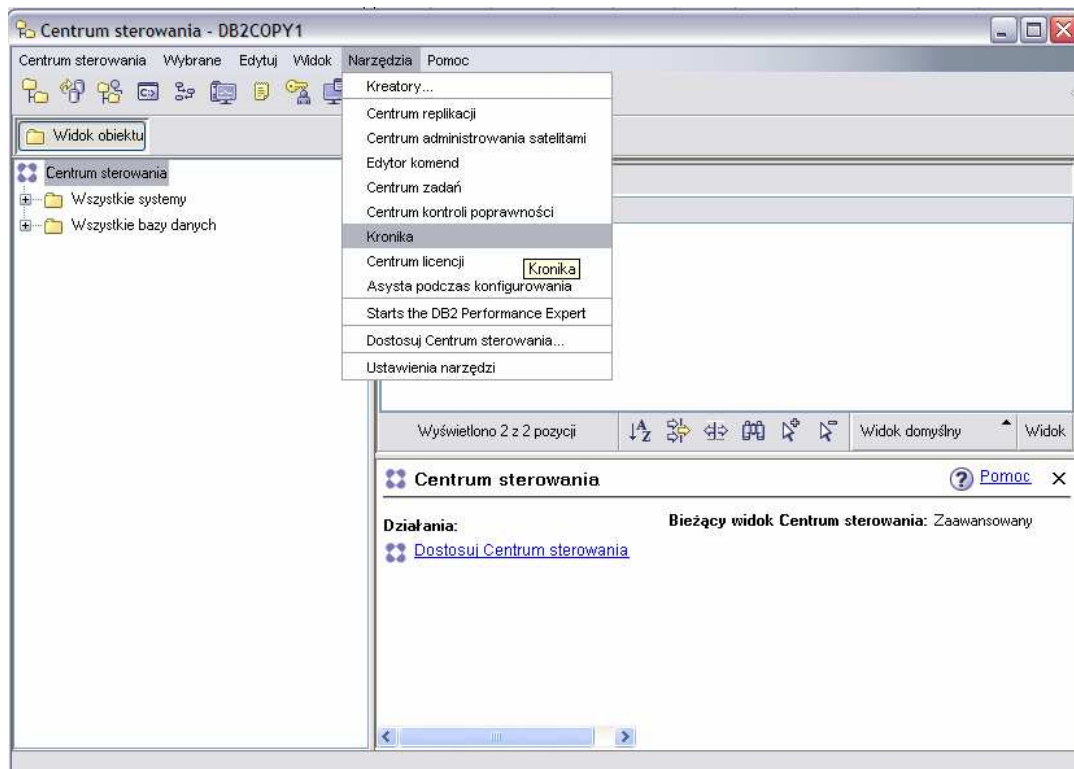
Rysunek 5.17 – Kronika

Typ informacji	Opis
Historia zadań	Wszystkie zaplanowane zadania i ich status.
Historia bazy danych	Rejestr działań wykonanych na bazie danych (tworzenie kopii zapasowej, przywracanie bazy, reorganizacja, itd.).
Komunikaty	Historia wiadomości pochodzących z narzędzi DB2. Jest to bardzo użyteczne, jeśli chcemy przywołać i porównać stare wiadomości o błędach lub jeśli zamkniemy okno dialogowe zbyt szybko, bądź przez przypadek.
Dziennik powiadomień	Zawiera wiadomości z poziomu systemu operacyjnego. To tutaj zapisywane są krytyczne błędy.

Tabela 5.2 – Informacje uzyskane przy pomocy Kroniki

### Uruchamianie Kroniki

Kronikę można uruchomić z Centrum sterowania klikając menu *Narzędzia > Kronika*, tak jak zostało to pokazane na Rysunku 5.18. Alternatywnie można uruchomić to narzędzie z menu Start systemu Windows: *Start > Programy > IBM DB2 > DB2COPY > Narzędzia administracji ogólnej > Kronika*.



Rysunek 5.18 – Uruchamianie Kroniki

## 5.8 Monitor poprawności

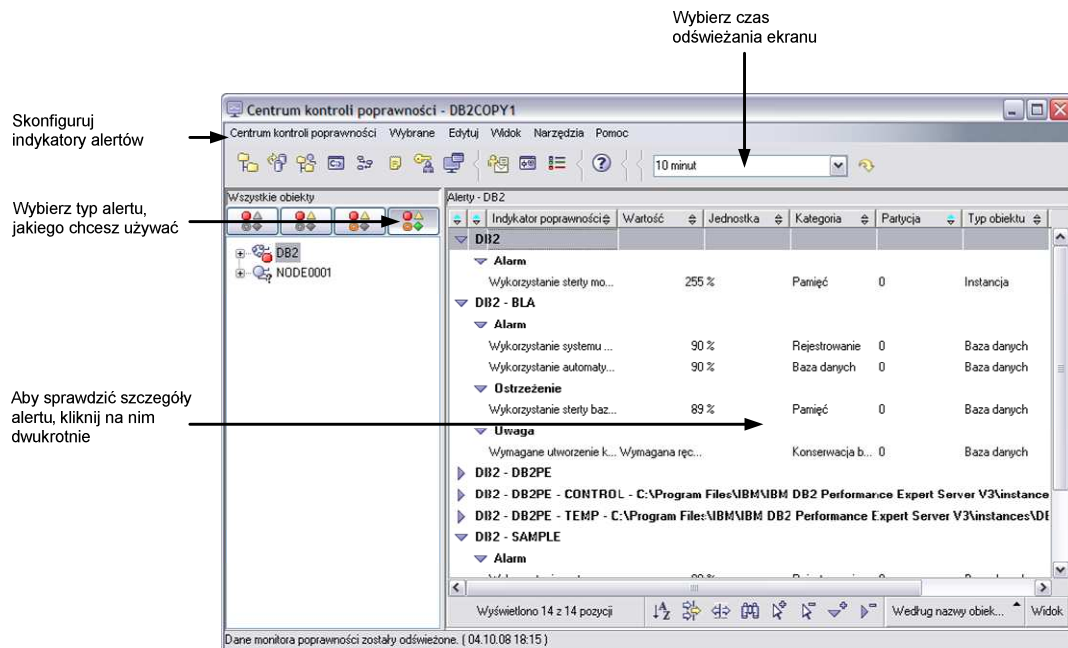
Monitor poprawności (ang. Health Monitor) jest domyślnym agentem, który działa wewnątrz silnika DB2, monitorującym wszystkie aspekty zdrowia bazy danych (pamięć, zarządzanie przestrzenią, automatyczne czynności wcześniej zdefiniowane, itd.). Kiedy któryś z aspektów DB2 działa poza zbiorem parametrów, podniesiony zostaje alarm, aby zwrócić uwagę administratora. Istnieją trzy rodzaje stanów alarmowych:

- ▶ Uwaga: nienormalny stan
- ▶ Ostrzeżenie: niekrytyczny stan, który nie wymaga natychmiastowej uwagi, ale może wskazywać na nieoptymalny system
- ▶ Alarm: krytyczny stan wymagający natychmiastowego działania

Monitor poprawności może być włączany lub wyłączany przy pomocy parametru konfiguracji menedżera bazy danych HEALTH\_MON.

### 5.8.1 Centrum kontroli poprawności

Centrum kontroli poprawności (ang. Health Center) jest graficznym narzędziem służącym do interakcji z Monitorem poprawności. Centrum kontroli poprawności analizuje alarmy w systemie z poziomu instancji, bazy danych i przestrzeni tablicowych. Rysunek 5.19 pokazuje Centrum kontroli poprawności.

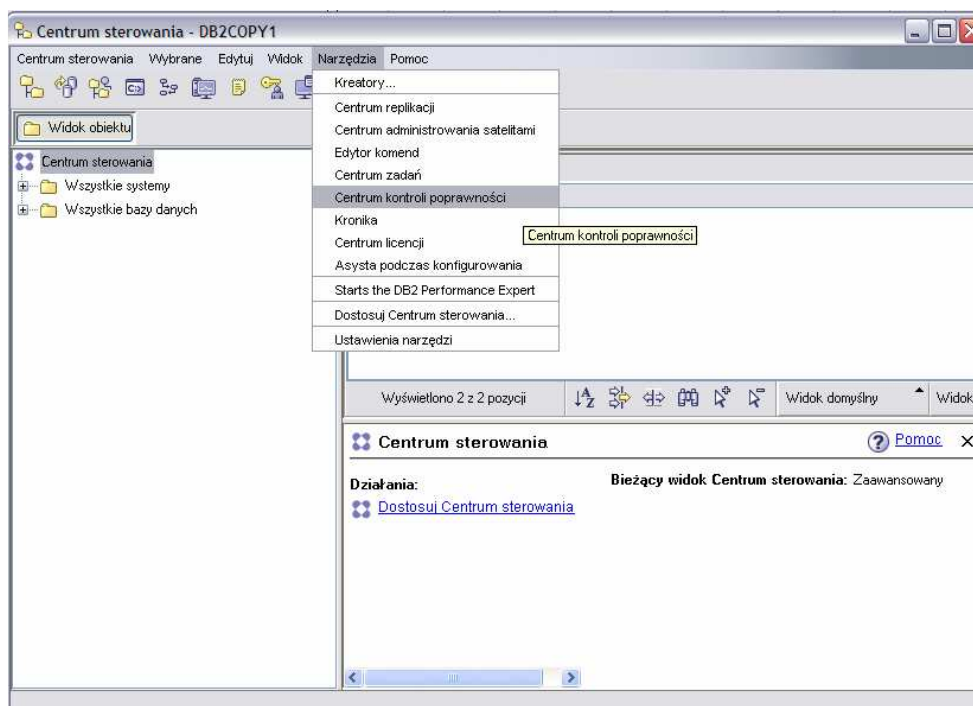


Rysunek 5.19 – Centrum kontroli poprawności

#### Uruchamianie Centrum kontroli poprawności

Centrum kontroli poprawności można uruchomić z Centrum sterowania klikając Menu *Narzędzia* i wybierając Centrum kontroli poprawności. Jest to pokazane na Rysunku 5.20. Narzędzie to można również uruchomić poprzez menu *Start > Programy > IBM DB2 > DB2COPY > Narzędzia Monitorujące > Centrum kontroli poprawności*.

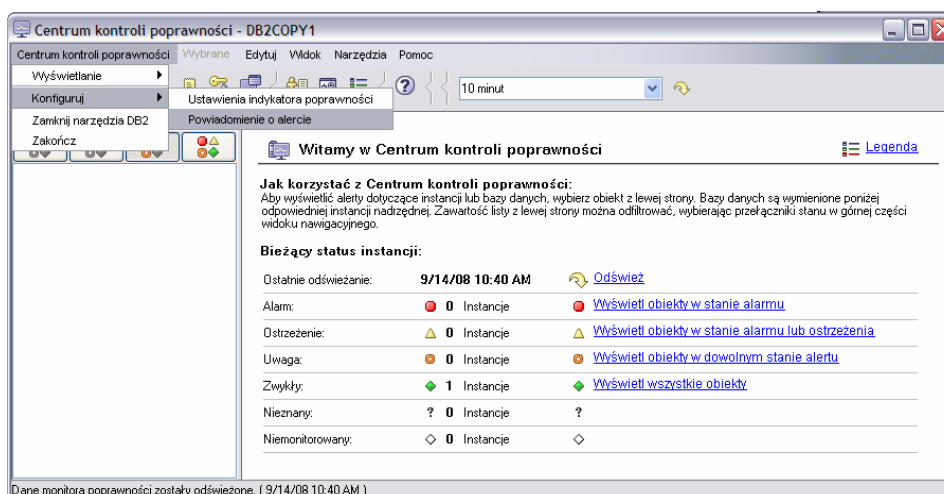




Rysunek 5.20 – Uruchamianie Centrum kontroli poprawności

### Konfigurowanie powiadomień o alertach

Gdy Centrum kontroli poprawności zostanie już uruchomione, można skonfigurować powiadomienia o alertach klikając na menu *Centrum kontroli poprawności* > *Konfiguruj* > *Powiadomienie o alertach*, tak jak zostało to pokazane na Rysunku 5.21. Zawiadomienia o alertach pozwalają na wprowadzenie nazwisk z adresami mailowymi lub numerów pagerów ludzi, z którymi należy się kontaktować, jeśli zostanie podniesiony alarm.



Rysunek 5.21 – Powiadomienia o alarmach



**CZEŚĆ II – NAUKA DB2:  
ADMINISTRACJA BAZĄ DANYCH**



# 6

## Rozdział 6 – Architektura DB2

W tym rozdziale krótko omówimy architekturę DB2, w tym:

- Model procesu w DB2
- Model pamięci w DB2
- Model przechowywania danych w DB2

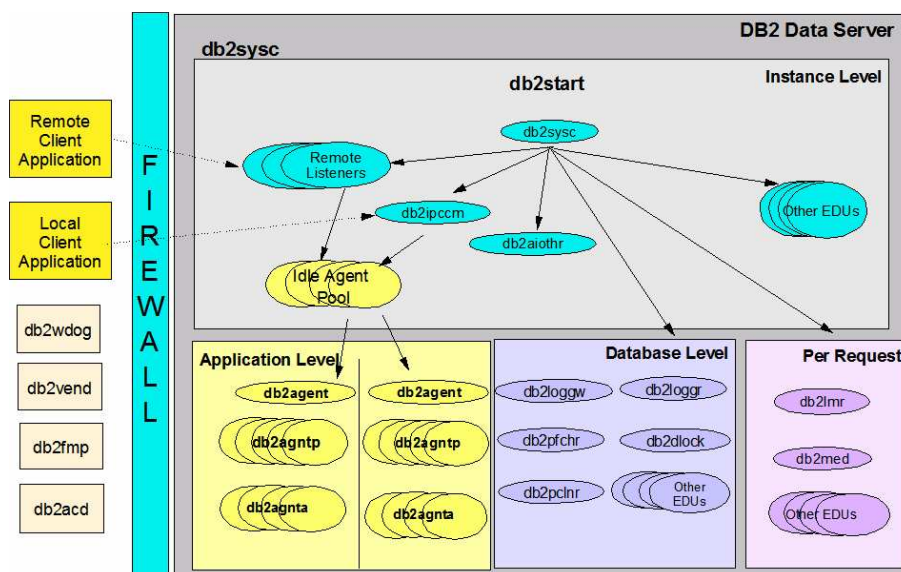
**Ważne:**

Aby uzyskać więcej informacji kliknij na poniższy link:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4482>

### 6.1 Model procesu w DB2

Rysunek 6.1 przedstawia model procesu w DB2. Na rysunku tym prostokąty reprezentują procesy, podczas gdy wątki są oznaczone elipsami. Główny proces DB2 nazywa się `db2sysc`. W jego skład wchodzi kilka wątków, z których główny również nazywa się `db2sysc`. Jest to wątek służący do tworzenia innych wątków. Gdy zdalna aplikacja próbuje połączyć się z serwerem za pomocą polecenia SQL `CONNECT`, wątki nasłuchujące otrzymują żądanie i kontaktują się z agentem koordynacji (`db2agent`). Agent DB2 działa jak robotnik wykonujący operacje w imieniu DB2. Gdy aplikacja działa lokalnie wszystkie operacje wykonywane są na tym samym serwerze co DB2, a poszczególne kroki są zbliżone, z tą różnicą, że agentem obsługującym żądania jest `db2ipccm`, a nie wątek `db2tcpcm`. W niektórych przypadkach, na przykład gdy włączone jest przetwarzanie równoległe, `db2agent` może tworzyć innych agentów, którzy będą pokazywani jako wątki `db2agntp`. Inni agenci pokazani na Rysunku 6.1, tacy jak `db2pfchr`, `db2loggr` czy `db2lock`, mogą zostać użyti do innych celów. Najbardziej powszechne procesy są przedstawione w Tabeli 6.1, a najbardziej powszechne wątki opisane zostały w Tabeli 6.2.



Rysunek 6.1 – Model procesu w DB2

Nazwa procesu	Opis
db2sysc (Linux) db2syscs (Win)	Główny silnik, lub kontroler systemu DB2. W DB2 9.5 istnieje tylko jeden wielowątkowy główny proces-silnik dla całej partycji. Wszystkie inne Jednostki Rozsyłalne Mechanizmu (ang. Engine Dispatchable Units, lub EDU) są wątkami wchodzącymi w skład tego procesu. Bez tego procesu serwer bazodanowy nie może funkcjonować.
db2acd	Autonomiczny demon obliczeniowy. Jest używany do przeprowadzania zautomatyzowanych zadań po stronie klienta, takich jak Monitor poprawności (ang. Health Monitor), automatyczne narzędzia do utrzymywania systemu, czy też harmonogram administratora. Proces ten dawniej nazywany był db2hmon.
db2wdog	Proces obserwatora w DB2. Proces ten jest rodzicem dla głównego silnika, db2sysc. Ma za zadanie czyścić zasoby, gdy proces db2sysc skończy działanie w nietypowy sposób.
db2vend	Otocza procesy dostawcy wprowadzone w DB2 9.5. Każdy kod napisany przez zewnętrznego dostawcę działa poza głównym silnikiem DB2. Jest to kod napisany przez dostawcę innego niż IBM, który współdziała z DB2. Przykładowo archiwizowanie dzienników może być przeprowadzane przez program dostarczony nie przez IBM, poprzez wskazanie kodu jako parametr „wychodzący program użytkownika” (ang. user exit routine)
db2fmp	Otocza procesy, które uruchamiają kod użytkownika na serwerze poza firewallem za pomocą procedur składowanych i funkcji zdefiniowanych przez użytkownika. Ten proces zastępuje zarówno procesy db2udf, jak i db2dari, które były używane we wcześniejszych wersjach DB2.

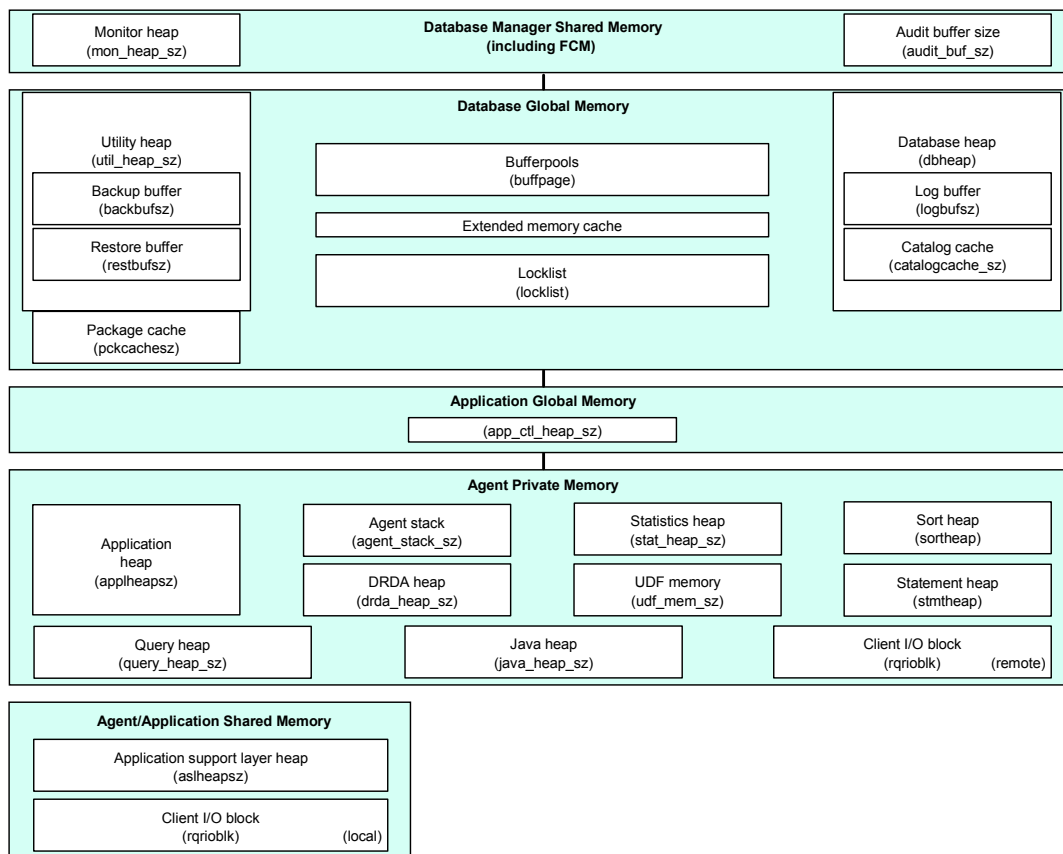
Tabela 6.1 – Najważniejsze procesy w DB2

Nazwa wątku	Opis
db2sysc	Wątek kontrolera systemu. Jest on odpowiedzialny za uruchamianie, zamykanie oraz zarządzanie działającą instancją.
db2tcpm	Wątek nasłuchujący na protokole TCP/IP
db2agent	Agent koordynujący, przeprowadzający operacje w imieniu aplikacji (przynajmniej 1 dla każdego połączenia, w zależności, czy włączony jest koncentrator połączeń).
db2agntp	Aktywny utworzony pod-agent. Wymaga parametru INTRA_PARALLEL ustawionego na YES. Wykonuje operacje bazodanowe dla aplikacji. Wątek db2agent nadzoruje pracę pomiędzy wątkami pod-agentów db2agntp.
db2pfchr	Asynchroniczny wątek służący do wstępnego ładowania danych (NUM_IOSERVERS).
db2pclnr	Asynchroniczny wątek służący do zapisu danych na wejściu/wyjściu (NUM_IOCLEANERS).

**Tabela 6.2 – Najważniejsze wątki w DB2**

## 6.2 Model pamięci w DB2

Model pamięci w DB2 składa się z różnych obszarów w pamięci przydzielanych na poziomie instancji, bazy danych, aplikacji i agenta, tak jak pokazano na Rysunku 6.2. W tej książce nie będziemy szczegółowo opisywać tych obszarów, naszkicujemy natomiast rolę, jaką każdy z nich spełnia.



Rysunek 6.2 – Model pamięci w DB2

W momencie, gdy uruchamiana jest instancja, przydzielona zostaje pamięć współdzielona menadżera bazy danych. Zazwyczaj nie zabiera to dużo miejsca. Kiedy po raz pierwszy podłączasz się do bazy danych alokowana jest globalna pamięć bazy danych (ang. Database Global Memory). Jedną z najważniejszych części tego bloku jest pula buforów (ang. buffer pool), służąca głównie do poprawy wydajności zapytań. Rozmiar puli buforów określa, jaką wielkość będzie miała cała globalna pamięć bazy danych.

Prywatna pamięć agenta to pamięć używana przez każdego agenta DB2. W przypadku wyłączanego koncentratora połączeń, każde połączenie wymaga osobnego agenta. Zazwyczaj agent używa od 3 do 5 MB pamięci. W przypadku koncentrowania połączeń, kilka połączeń może używać jednego agenta, zmniejszając użycie pamięci fizycznej.

## 6.3 Model przechowywania danych

W sekcji tej omówimy poniższe pojęcia:

- strona i ekstent
- pula buforów
- przestrzeń tablicowa

### 6.3.1 Strony i ekstenty

Strona to najmniejsza jednostka przechowująca w DB2. Dopuszczalne rozmiary stron to: 4K, 8K, 16K oraz 32K. Ekstent (obszar) jest grupą stron. Praca na jednej stronie na raz w DB2 byłaby zbyt kosztowna wydajnościowo, dlatego też DB2 pracuje na całych ekstentach. Rozmiary stron i ekstentów są definiowane podczas pracy z pulami buforów i przestrzeniami tablicowymi, o czym przekonamy się w kolejnej sekcji.

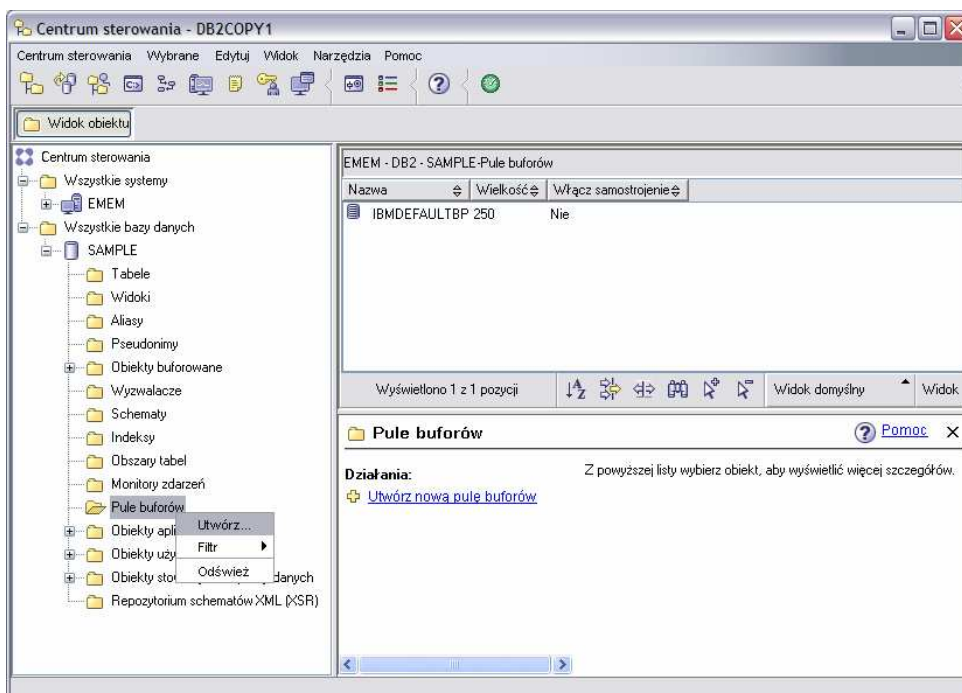
### 6.3.2 Pule buforów

Pula buforów to rzeczywista pamięć podręczna dla danych tablicowych oraz indeksowych. Poprawia wydajność poprzez zmniejszenie bezpośrednich, sekwencyjnych operacji I/O oraz poprzez wykorzystywanie asynchronicznych operacji czytania i zapisywania danych. DB2 przewiduje, które strony będą potrzebne i łąduje je z dysku do puli buforów, skąd są pobierane do użycia.

Pule buforów są to alokowane w pamięci 4K, 8K, 16K lub 32K jednostki stron. Powinna istnieć przynajmniej jedna pula buforów dla każdej bazy danych oraz dla każdej wielkości strony przestrzeni tablicowej.

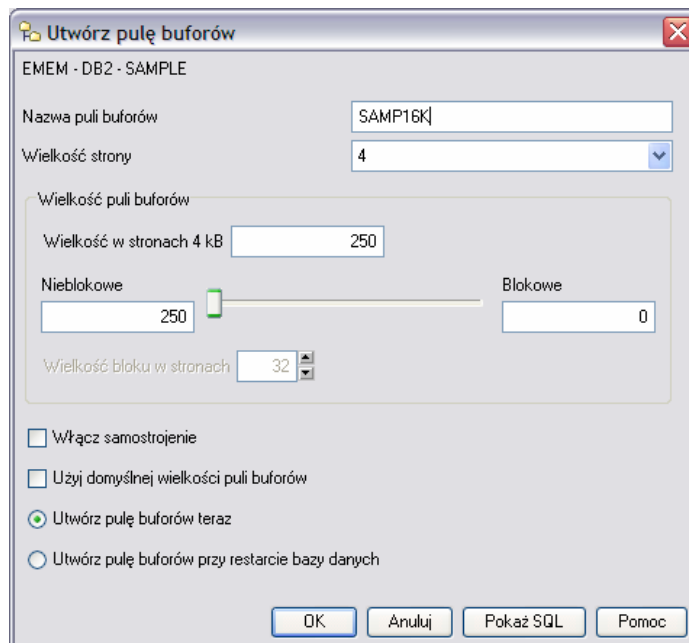
#### Tworzenie puli buforów

Aby stworzyć pulę buforów można użyć polecenia `CREATE BUFFERPOOL`. Można również użyć Centrum sterowania, wystarczy kliknąć prawym przyciskiem myszy na folderze *Pule buforów*, po czym wybrać opcję *Utwórz*, tak jak pokazano na Rysunku 6.3.



Rysunek 6.3 – Tworzenie puli buforów

Po wybraniu opcji *Utwórz*, na ekranie pojawi się okno dialogowe *Utwórz pulę buforów*, tak jak pokazano na rysunku 6.4.

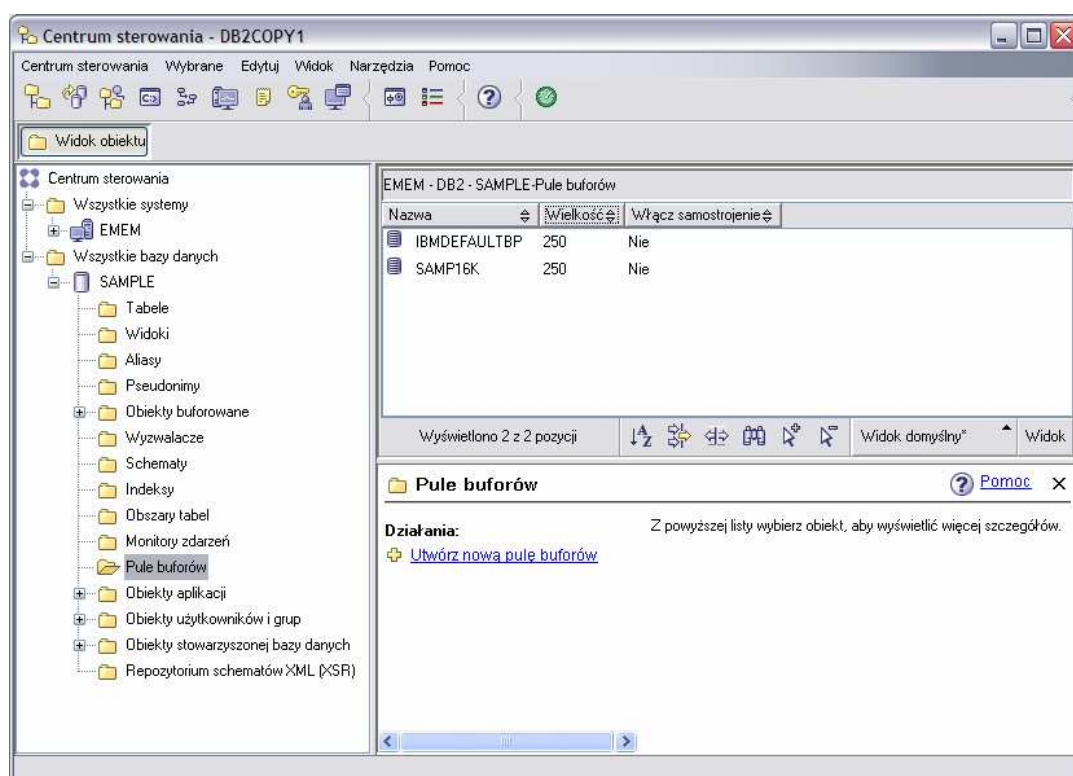


Rysunek 6.4 – Okno dialogowe Utwórz pulę buforów



Większość pól na Rysunku 6.4 nie wymaga wyjaśnień. Pola *Nieblokowane* oraz *Blokowane* odnoszą się do liczby stron, które mają być niezblokowane i zblokowane. Zblokowane pule buforów gwarantują to, że sąsiednie strony na dysku, wewnątrz zblokowanego obszaru będą przeniesione do puli buforów razem. Takie działanie może poprawić wydajność. Liczba stron nie może być większa niż 98 procent całkowitej liczby stron przeznaczonych dla puli buforów. Jeśli podana wartość wynosi 0, oznacza to wyłączenie blokady I/O.

Gdy pula buforów zostanie stworzona, będzie ona wyświetlona w Centrum Sterowania, jak pokazano na Rysunku 6.5.



Rysunek 6.5 – Widok Centrum Sterowania po utworzeniu puli buforów “SAMP16K”

### 6.3.3 Przestrzenie tablicowe

Przestrzenie tablicowe są logicznym interfejsem pomiędzy logicznymi tablicami, a fizyczną pamięcią systemu (pule buforów) oraz kontenerami. Do stworzenia przestrzeni tablicowej używamy polecenia `CREATE TABLESPACE`. Poprzez parametry polecenia możemy określić:

- rozmiar strony dla przestrzeni tablicowej (4KB, 8KB, 16KB, lub 32KB). Pula buforów musi posiadać identyczną wartość rozmiaru strony,
- nazwę puli buforów związanej z tworzoną przestrzenią tablicową,
- rozmiar ekstentu (extent size),
- rozmiar wstępnego ładowania (pre-fetch size).

### Typy przestrzeni tablicowych

Istnieją trzy typy przestrzeni tablicowych:

- Regularne  
Używane są do przechowywania tabeli użytkownika. Na przykład przestrzeń tablicowa USERSPACE1, utworzona domyślnie, jest regularną przestrzenią tablicową.
- Duże  
Używane są opcjonalnie, aby oddzielić duże obiekty, tzw. LOBy (ang. Large Object). Ponadto są używane do przechowywania danych XML w bazach danych stworzonych ze wsparciem dla technologii pureXML – bazy danych muszą być wtedy w standardzie UNICODE, a typy kolumn muszą być danymi XML. Duże przestrzenie tablicowe są ustawieniami domyślnymi.
- Tymczasowe  
Istnieją dwa typy tymczasowych przestrzeni tablicowych
  - ▶ Tymczasowe systemowe  
Używane są w DB2 przez wewnętrzne operacje, takie jak sortowanie. Na przykład: przestrzeń tablicowa TEMPSPACE1, domyślnie utworzona podczas tworzenia bazy danych, to przykład tymczasowej, systemowej przestrzeni tablicowej.
  - ▶ Tymczasowe użytkownika  
Używane do tworzenia przez użytkownika tymczasowych globalnych tabel definiowanych (ang. User Defined Global Temporary). Są to tymczasowe tabele przechowywane w pamięci operacyjnej, często mylone z tymczasowymi, systemowymi przestrzeniami tablicowymi.

### Zarządzanie przestrzenią tablicową

Przeźstrzenie tablicowe mogą być sklasyfikowane na podstawie sposobu zarządzania nimi. Można to określić w ramach polecenia `CREATE TABLESPACE:`

#### Zarządzane przez system

Ten typ przestrzeni tablicowej znany jest jako SMS (ang. System Managed Storage). Oznacza to, że system operacyjny zarządza pamięcią masową. Jest to sposób łatwy do zarządzania, a kontenerami są katalogi systemu plików. Miejsce nie jest pre-alokowane, ale pliki rozrastają się dynamicznie. Kontenery ustala się na stałe w momencie tworzenia przestrzeni tablicowej, nie można dodawać innych kontenerów później, chyba że używane jest preadresowane przywracanie. Podczas używania przestrzeni typu SMS dane tablicowe, indeksowe czy obiekty typu LOB nie mogą być rozciągnięte na różne przestrzenie tablicowe.

#### Zarządzane przez bazę danych

Ten typ przestrzeni tablicowej znany jest jako DMS (ang. Database Managed Storage). Oznacza to, że DB2 zarządza pamięcią masową. Zarządzanie miejscem wymaga więcej ręcznych interwencji ze strony administratora bazy danych. Kontenerami mogą być pre-alokowane pliki, lub urządzenia. W przypadku urządzeń dane są zapisywane bezpośrednio, bez użycia pamięci podręcznej wykorzystywanej przez system operacyjny.

Można dodawać, usuwać lub zmieniać rozmiary kontenerów. Przestrzenie tablicowe typu DMS uzyskują najlepsze wyniki wydajnościowe. Dane tablicowe, indeksowe czy obiekty typu LOB mogą być podzielone na kilka przestrzeni tablicowych, przez co wydajność wzrasta.

#### **Zarządzane przez pamięć automatyczną**

Typ przestrzeni tablicowej zarządzany przez pamięć automatyczną łączy w sobie prostotę obsługi podobną do przestrzeni tablicowych SMS wraz z najlepszymi wynikami pod względem wydajności oraz elastycznością znaną z przestrzeni typu DMS. Dlatego też ten typ przestrzeni tablicowej jest ustawiany domyślnie. Używając tej przestrzeni tablicowej użytkownik najpierw wskazuje logiczną grupę urządzeń pamięci masowej. Nie podaje się wprost żadnej definicji kontenera, ponieważ są one automatycznie tworzone na ścieżkach pamięci masowej. Rozrost istniejących kontenerów oraz dodawanie nowych jest całkowicie zarządzane przez DB2.

Aby użyć tej opcji, najpierw należy stworzyć bazę danych z włączoną opcją zarządzania przez pamięć automatyczną (jest to domyślne ustawienie) oraz powiązać nowoutworzoną bazę danych ze zbiorem ścieżek pamięci masowej. Po utworzeniu, jeśli zachodzi taka potrzeba, można przedefiniować ścieżki, używając na bazie danych operacji RESTORE. Następnie można utworzyć przestrzenie tablicowe w taki sposób, aby używały zarządzania pamięcią automatyczną.

#### **Przykłady używania pamięci automatycznej**

Na początku tworzymy bazę danych z włączoną opcją, tak jak w poniższych przykładach:

Pamięć automatyczna jest włączana domyślnie:  
`CREATE DATABASE DB1`

Pamięć automatyczna jest włączana wprost:  
`CREATE DATABASE DB1 AUTOMATIC STORAGE YES`

Pamięć automatyczna jest włączana domyślnie, dodatkowo podaje się ścieżki pamięci:  
`CREATE DATABASE DB1 ON /data/path1, /data/path2`

Opcja zarządzania przez pamięć automatyczną jest wyłączana:  
`CREATE DATABASE DB1 AUTOMATIC STORAGE NO`

Następnie tworzy się przestrzeń tablicową zarządzaną przez pamięć masową, tak jak w poniższych przykładach:

Pamięć automatyczna dla przestrzeni tablicowej jest włączana domyślnie:  
`CREATE TEMPORARY TABLESPACE TEMPTS`

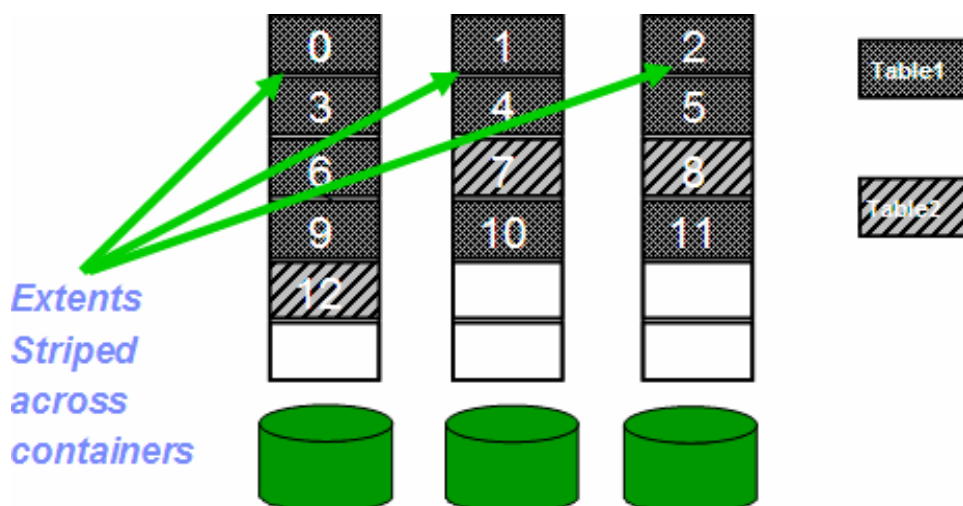
Pamięć automatyczna dla przestrzeni tablicowej jest włączana wprost:  
`CREATE TABLESPACE TS2 MANAGED BY AUTOMATIC STORAGE`

Pamięć automatyczna dla przestrzeni tablicowej jest włączana wprost, ponadto początkowy rozmiar jest alokowany, określone zostają tempo wzrostu oraz rozmiar do jakiego może się rozrosnąć.

```
CREATE TABLESPACE TS1
  INITIALSIZE 500 K
  INCREASESIZE 100 K
  MAXSIZE 100 M
```

### Jak w przestrzeniach tablicowych przechowywane są dane?

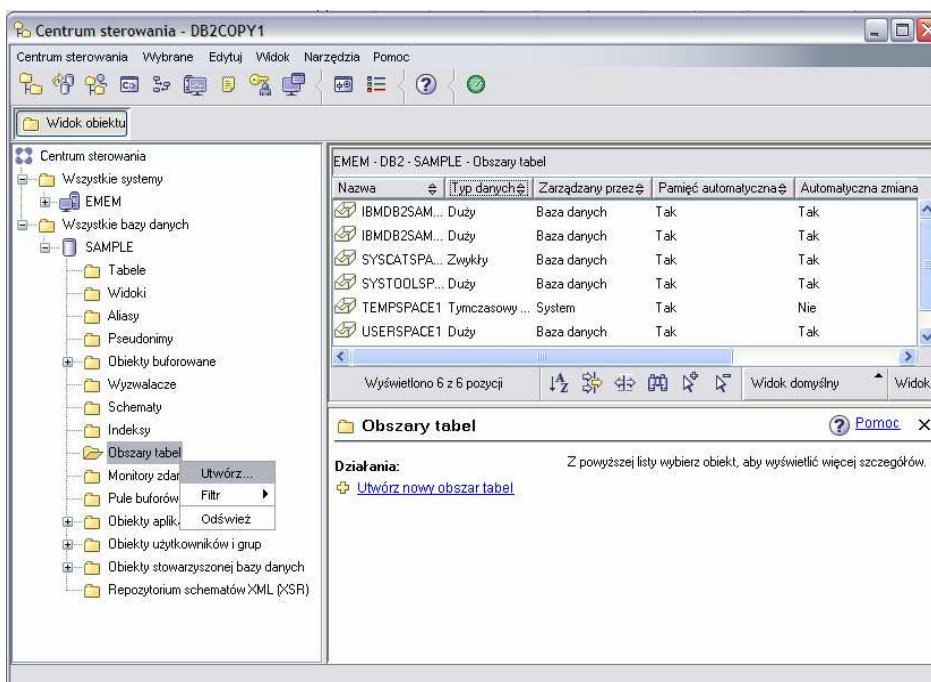
Domyślnie DB2 zapisuje na dysk obszary, rozdzielając je pomiędzy kontenery. Na przykład, jeśli mamy 4K przestrzeń tablicową typu DMS z ekstentami o wielkości 8, wykorzystującą 3 kontenery, oznacza to, że 32K danych (4K x 8 stron na obszar = 32K) zostaną zapisane na jeden dysk, zanim nastąpi proces zapisu na kolejny. Jest to przedstawione na Rysunku 6.6. Zauważ, że tabele nie współdzielą ekstentów.



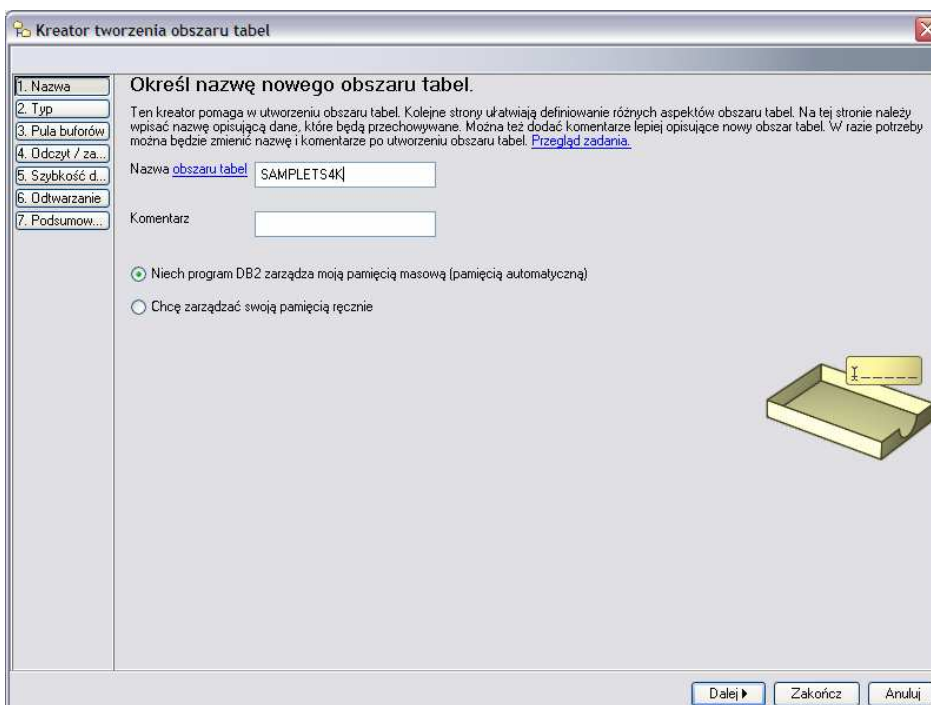
Rysunek 6.6 – Zapisywanie danych do przestrzeni tablicowych

### Tworzenie przestrzeni tablicowych za pomocą Centrum sterowania

Aby stworzyć przestrzeń tablicową z poziomu Centrum sterowania należy kliknąć prawym przyciskiem myszy na folder *Obszary tabel* w wybranej przez nas bazie danych oraz wybrać z menu opcję *Utwórz*, tak jak jest to pokazane na Rysunku 6.7. Uruchomi to kreator tworzenia przestrzeni tablicowych (Rysunek 6.8).



Rysunek 6.7 – Tworzenie przestrzeni tablicowej z poziomu Centrum sterowania



Rysunek 6.8 – Kreator tworzenia przestrzeni tablicowych



# 7

## Rozdział 7 – Klient DB2

W tym rozdziale zajmiemy się konfiguracją niezbędną do połączenia się klientem DB2 do serwera DB2 używając protokołu TCP/IP. Zauważmy, że serwer DB2 jest dostarczany razem z klientem, więc może on również pełnić funkcję klienta i łączyć się z innym serwerem. Jest kilka sposobów na konfigurację połączenia, jednak w tym rozdziale zajmiemy się najprostszą, używając narzędzia Asysta podczas konfigurowania.

### Ważne

Więcej informacji na temat budowy DB2 znajdziesz tutaj:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4222>

### 7.1 Katalogi DB2

Katalogi DB2 są to pliki binarne zawierające informacje o bazach danych, do których możemy się połączyć z naszego systemu. Mamy 4 katalogi:

#### 1. Katalog systemowy

Jest jak spis treści książki. Zawiera wszystkie bazy (lokalne i zdalne), do których możemy się połączyć. Dla baz lokalnych będzie on wskazywał na *Katalog lokalny*, a dla zdalnych na *Katalog węzłów*. Aby wyświetlić zawartość tego katalogu używamy polecenia:

```
list db directory
```

#### 2. Katalog lokalny

Zawiera informacje tylko o bazach, które znajdują się na lokalnym dysku. Aby wyświetlić jego zawartość używamy polecenia:

```
list db directory on <drive/path>
```

#### 3. Katalog węzłów

Katalog ten zawiera informacje, jak połączyć się do danej bazy. Np. jeśli używamy protokołu TCP/IP, wpis w katalogu będzie zawierał adres IP serwera, na którym znajduje się baza oraz port, na którym nasłuchuje instancja. Zawartość katalogu sprawdzamy komendą:

```
list node directory
```

#### 4. Katalog DCS (Database Connection Services)

Pojawi się, jeśli zainstalujemy oprogramowanie DB2 Connect, służące do połączenia się z bazą danych na systemie z/OS (mainframe) lub i5/OS. Zawartość katalogu sprawdzamy komendą:

```
list dcs directory
```

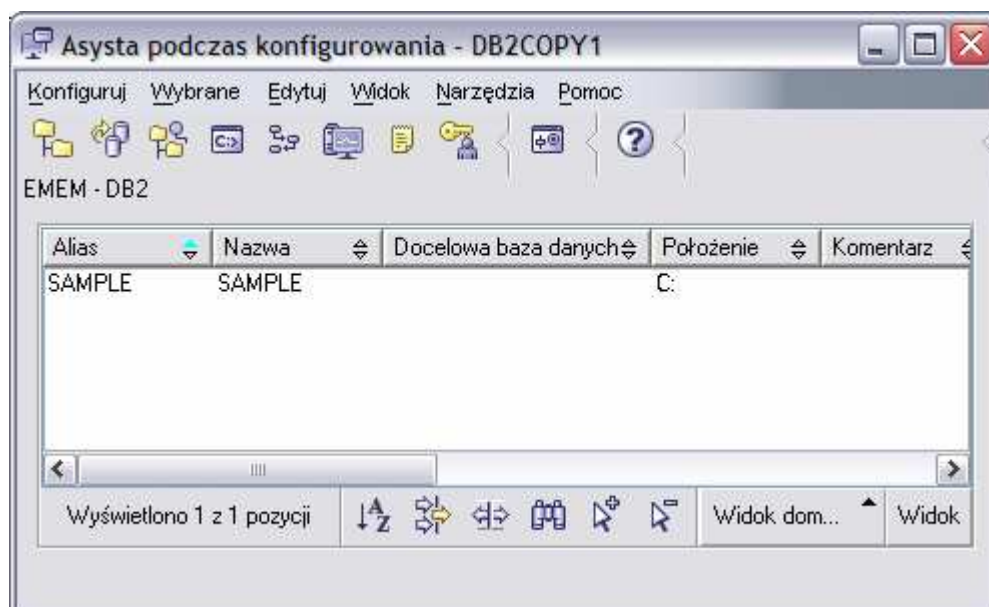
Przeglądanie i aktualizacja katalogów możliwe są również za pomocą narzędzia Asysta podczas konfigurowania.

### 7.2 Asysta podczas konfigurowania

Używając graficznego narzędzia *Asysta podczas konfigurowania*, można łatwo skonfigurować połączenie pomiędzy klientem a serwerem DB2.

Aby uruchomić narzędzie Asysta podczas konfigurowania w systemie Windows wybierz z menu *Start > Programy > IBM DB2 > DB2COPY1 > Narzędzia konfiguracyjne > Asysta podczas konfigurowania*, lub z linii komend *db2ca*.

Rysunek 7.1 pokazuje narzędzie Asysta podczas konfigurowania.



Rysunek 7.1 – Asysta podczas konfigurowania

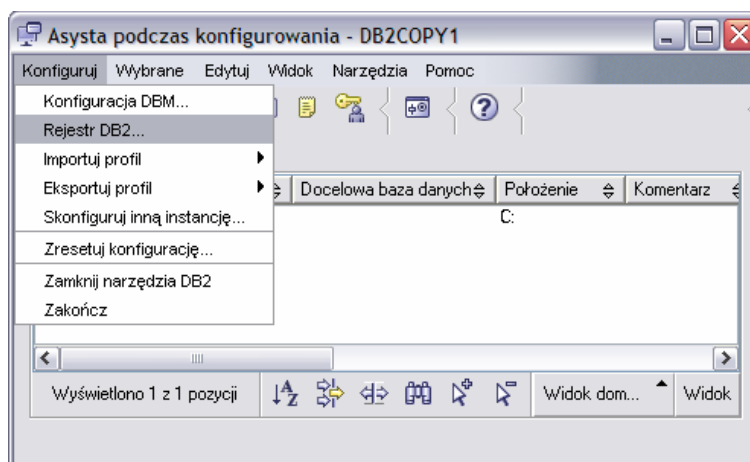


### 7.2.1 Konfiguracja serwera

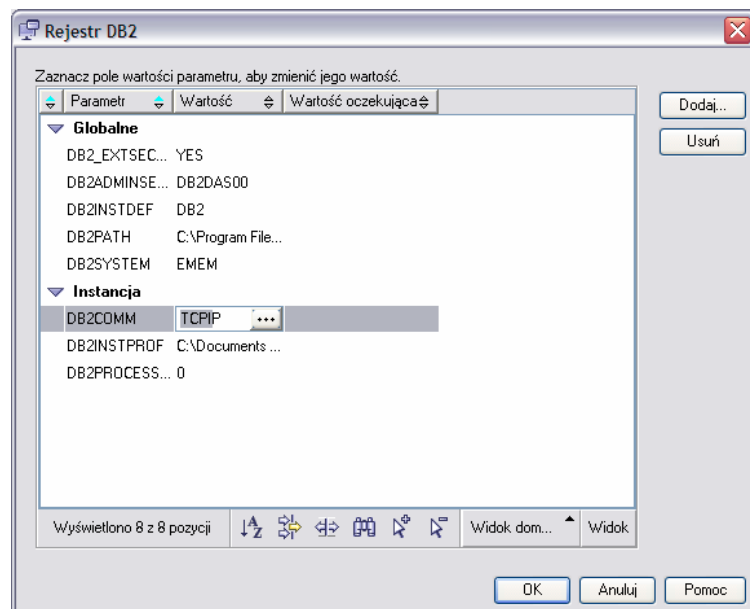
Na serwerze niezbędne jest skonfigurowanie dwóch parametrów:

#### 1) DB2COMM

Zmienna ta definiuje rodzaj protokołu użytego w komunikacji z klientem. Najczęściej jest to TCP/IP. Po zmianie tego parametru należy zrestartować instancję. Aby sprawdzić lub zmienić wartość parametru DB2COMM za pomocą narzędzia Asysta podczas konfigurowania wybierz z menu *Konfiguruj* -> *Rejestr DB*, tak jak jest pokazane na Rysunkach 7.2 i 7.3.



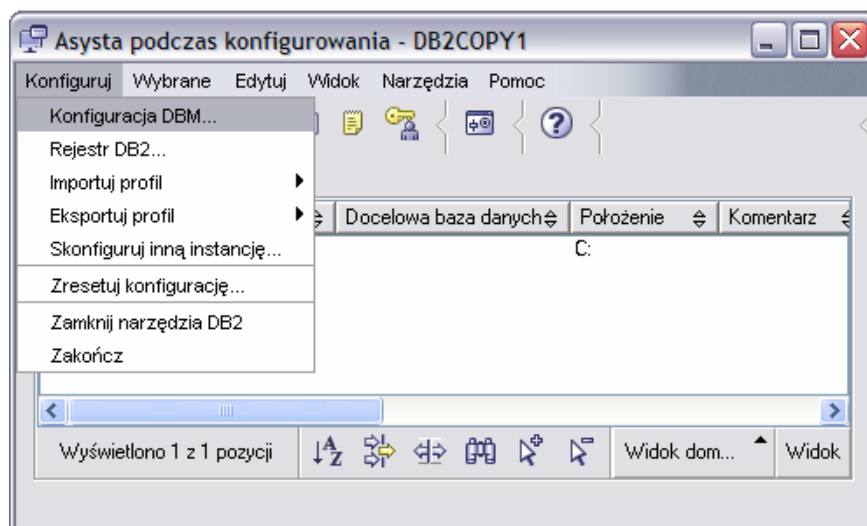
Rysunek 7.2 – Dostęp do rejestru DB2



Rysunek 7.3 –Sprawdzanie zmiennej DB2COMM w rejestrze DB2

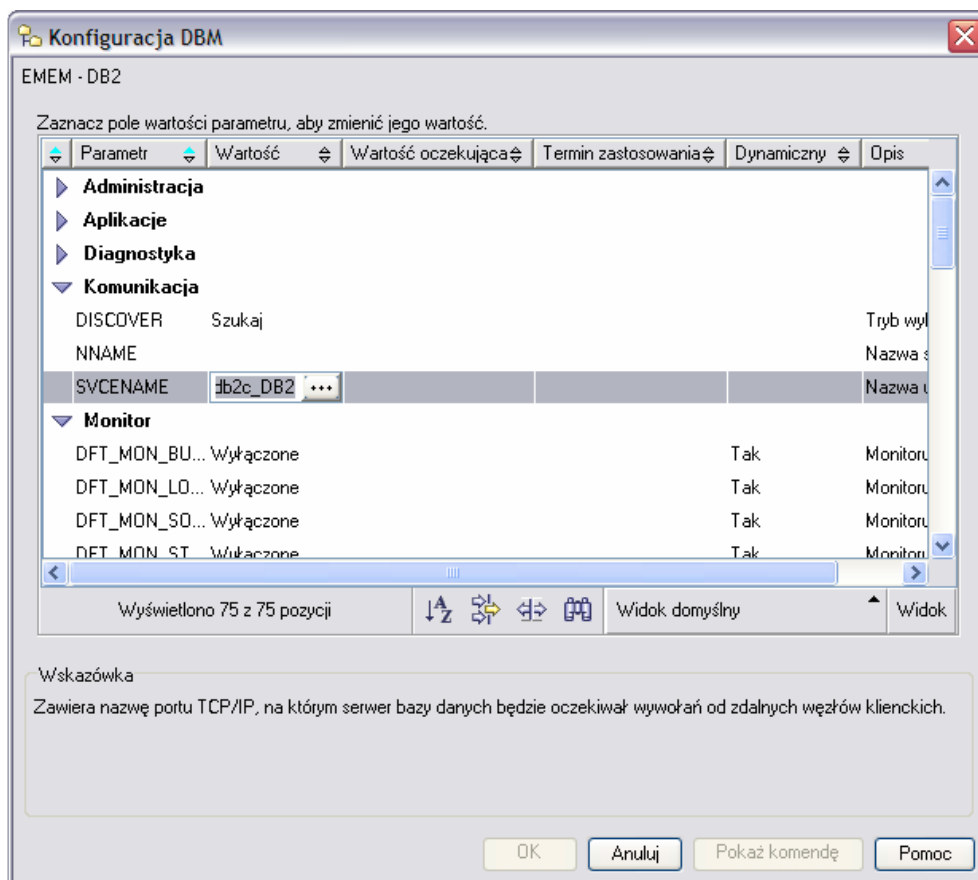
## 2) SVCENAME

Zmienna ta oznacza nazwę usługi, jeśli ta jest zdefiniowana w pliku usług TCP/IP, lub numer użytego portu, jeśli chcemy mieć dostęp do baz danych tej instancji. Z narzędzia Asysta podczas konfigurowania wybierz *Konfiguracja DBM*, jak na Rysunku 7.4.



**Rysunek 7.4 – Konfiguracja dbm cfg z poziomu narzędzia Asysta podczas konfigurowania**

Będąc w oknie *Konfiguracja DBM* przejdź do sekcji *Komunikacja* i poszukaj pola *SVCENAME*. Możesz zmienić tą wartość na łańcuch znaków lub numer portu (Rysunek 7.5).



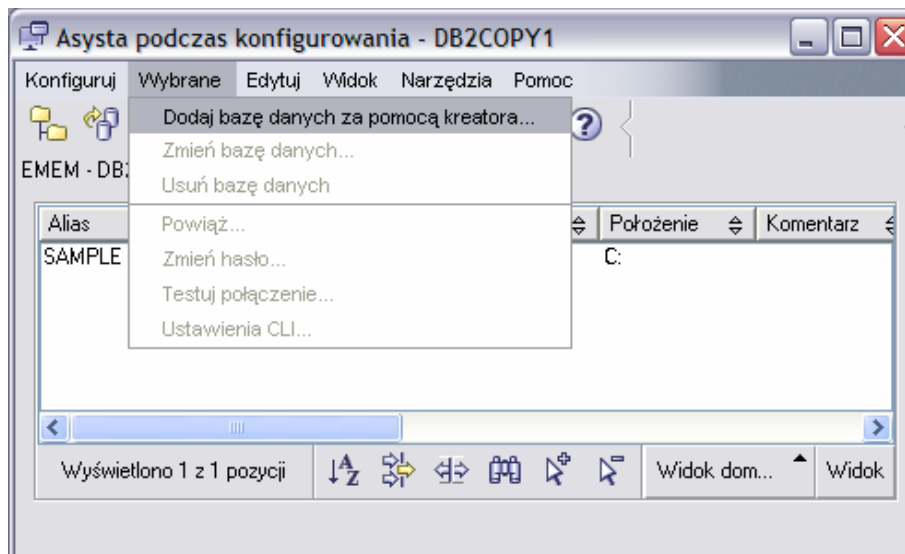
Rysunek 7.5 – Zmiana zmiennej SVCENAME.

### 7.2.2 Konfiguracja klienta

Konfigurując klienta, musimy posiadać informację o:

1. Nazwie bazy danych, do której się łączymy.
2. Numerze portu instancji DB2 na serwerze, gdzie znajduje się baza danych. Możemy również użyć nazwy usługi, jeśli jest ustawiona w pliku usług TCP/IP.
3. Nazwie użytkownika i haśle do połączenia się z bazą danych. Użytkownik musi być wcześniej utworzony na serwerze DB2.

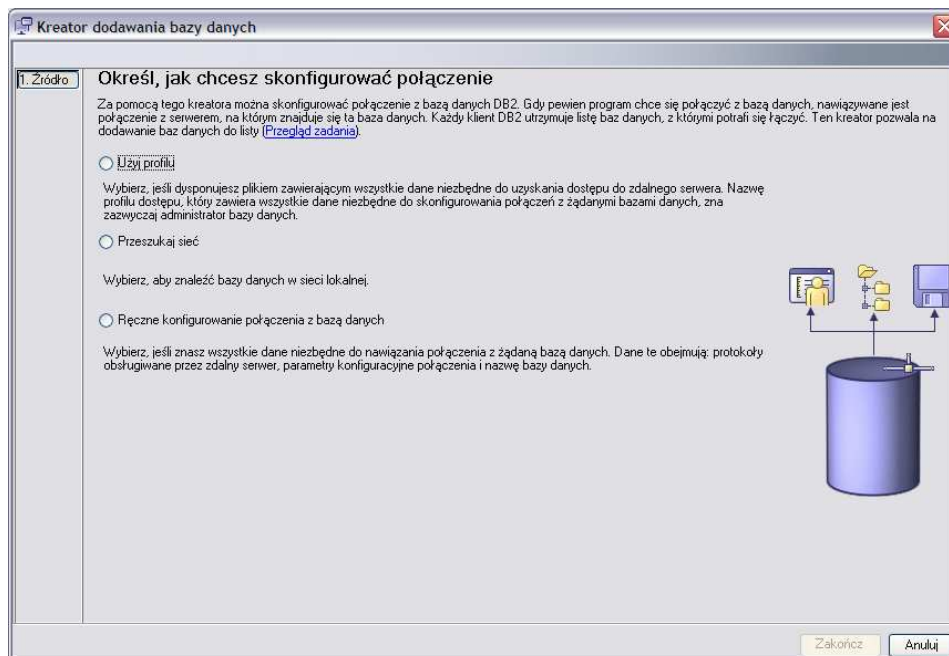
Powyższe parametry mogą być ustawione za pomocą narzędzia Asysta podczas konfiguracji. W tym celu uruchom kreatora dodawania bazy danych z menu *Wybrane -> Dodaj bazę danych za pomocą kreatora* (Rysunek 7.6).



Rysunek 7.6 – Uruchamianie kreatora dodawania bazy danych

Można również uruchomić ten kreator klikając prawym przyciskiem myszy na puste pole w narzędziu Asysta podczas konfigurowania, a następnie wybierając *Dodaj bazę danych za pomocą kreatora*.

Rysunek 7.7 przedstawia Kreator dodawania bazy danych.



Rysunek 7.7 – Kreator dodawania bazy danych

W Kreatorze dodawania bazy danych mamy do wyboru trzy 3 opcje:

### Użyj profilu

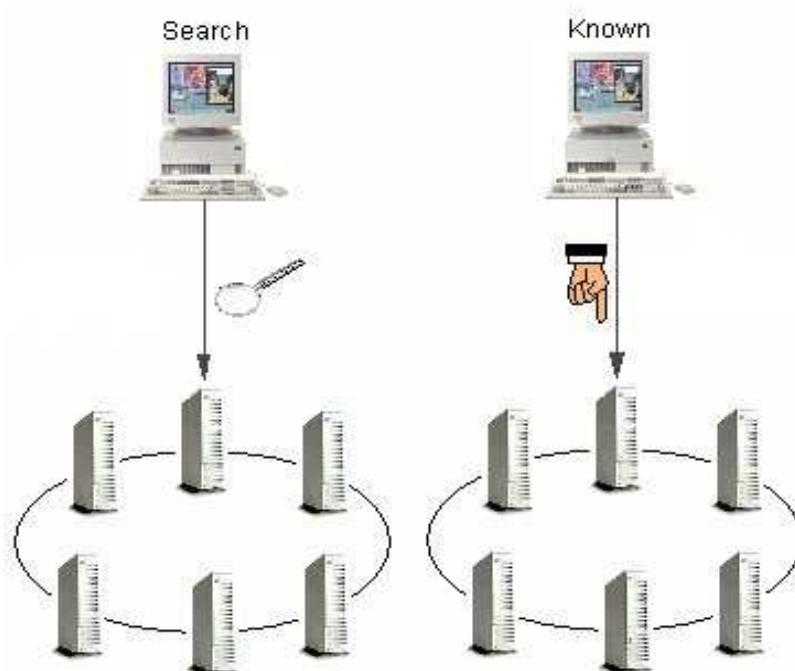
Gdy chcemy skonfigurować połączenie wielu klientów z tym samym serwerem DB2 wygodną metodą jest zachowanie konfiguracji połączenia w pliku. Dzięki temu możemy szybko załadować wszystkie ustawienia u innych użytkowników. Jeśli wybierzemy *Użyj profilu* w następnym kroku zostaniemy poproszeni o załadowanie pliku z profilem. W dalszej części opiszemy, jak tworzyć profile dla klientów i serwerów.

### Przeszukaj sieć

Metoda ta, znana również jako *Discovery*, polega na wyszukaniu w sieci serwera, instancji i bazy danych. Aby opcja wyszukiwania działała, wymagany jest Serwer Administracyjny DB2 (DAS), uruchomiony na każdym przeszukiwanym serwerze. Rozróżniamy dwie metody:

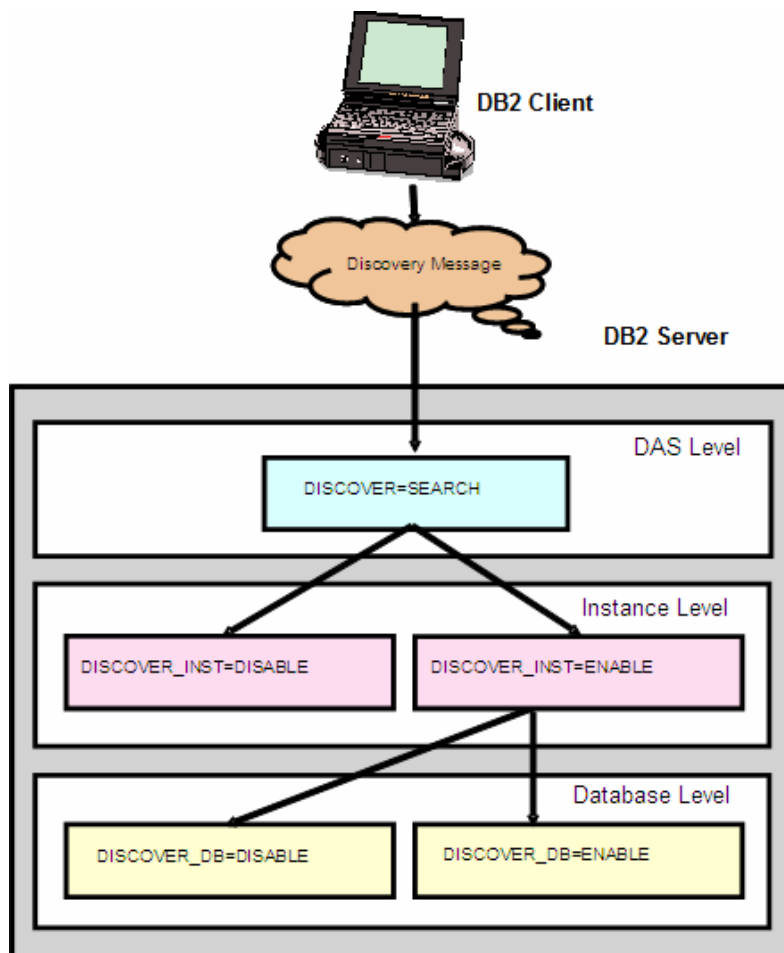
- **Inne systemy (Search).** Przeszukuje całą sieć. Nie jest zalecane przy sieci z dużą ilością hostów. Uzyskanie danych ze wszystkich serwerów może zająć dużo czasu.
- **Znane systemy (Known).** Przeszukuje tylko konkretny serwer o podanym adresie.

Obie metody ilustruje Rysunek 7.8.



Rysunek 7.8 –Metody przeszukiwania sieci: Search i Known.

W pewnych przypadkach administrator może nie chcieć, aby klienci przeszukiwali całą sieć, w której znajdują się poufne bazy. Możemy to uzyskać na poziomie Serwera Administracyjnego, instancji lub konkretnej bazy danych. Szczegóły przedstawia Rysunek 7.9.



**Rysunek 7.9 – Parametry pozwalające na przeszukiwanie baz danych**

Rysunek 7.9 przedstawia poziomy, na których możemy włączyć/wyłączyć przeszukiwanie. Na poziomie Serwera Administracyjnego, możemy ustawić parametr `DISCOVER` na `SEARCH` lub `KNOWN`. Na poziomie instancji, ustawiamy zmienną `DISCOVER_INST` dbm cfg na `ENABLE/DISABLE`. Na poziomie bazy danych parametr `DISCOVER_DB` możemy również ustawić na `ENABLE/DISABLE`. Dzięki konfiguracji tych opcji mamy pełną kontrolę nad procesem przeszukiwania baz.

### Ręczna konfiguracja połączenia do bazy danych

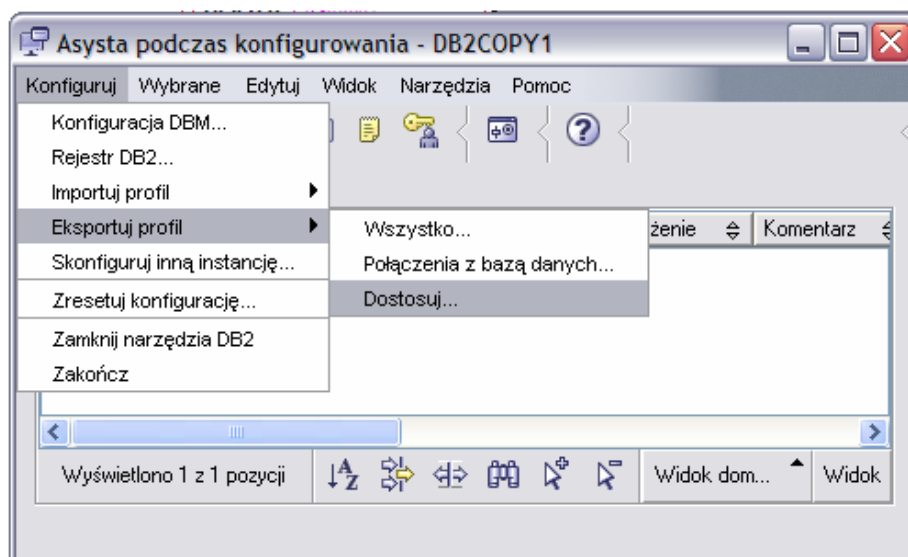
W tym przypadku ręcznie podajemy nazwę hosta, numer portu i parametry bazy danych, przekazując je bezpośrednio do narzędzia Asysta podczas konfigurowania. Wygenerowane zostanie polecenie, które zapewni odpowiednią konfigurację, ale na tym poziomie nie sprawdza wprowadzonych parametrów. Dopiero na etapie nawiązywania połączenia możemy zweryfikować ich poprawność. Należy pamiętać o sprawdzeniu nazwy użytkownika-

ka i hasła. Domyślnie uwierzytelnianie odbywa się na serwerze DB2, do którego się łączymy, zatem tam również weryfikujemy wprowadzane parametry.

### 7.2.3 Profile klienta i serwera

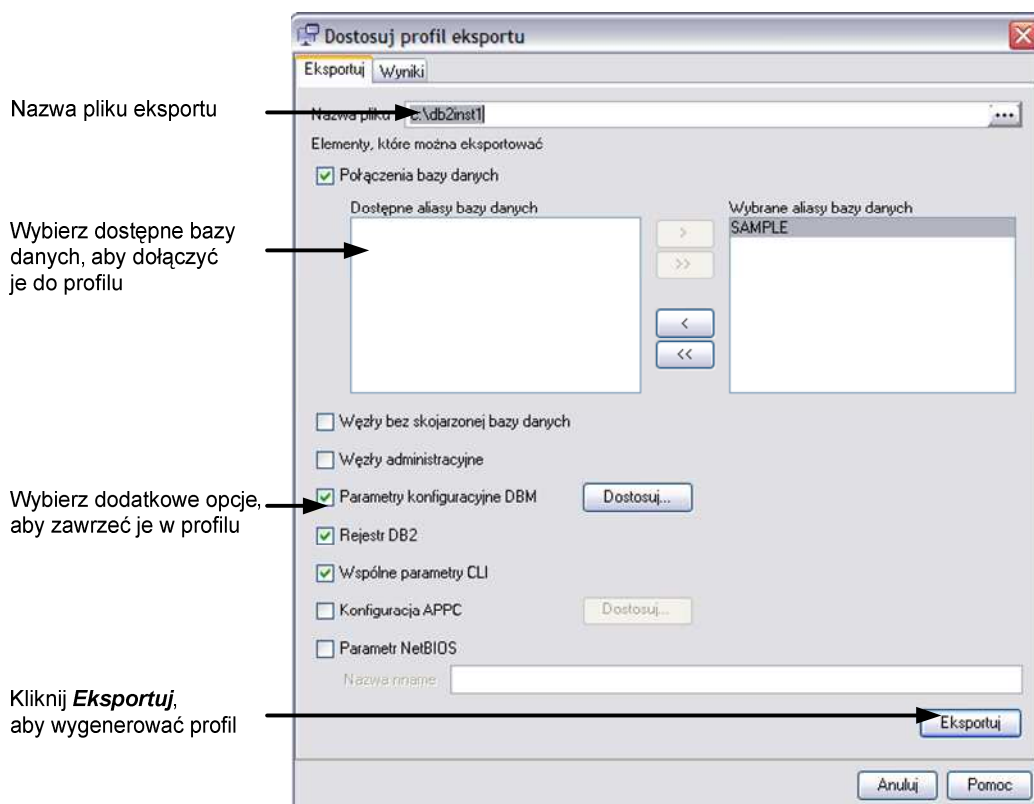
Jeśli konfigurujemy dużą liczbę serwerów lub klientów, nie musimy przeprowadzać konfiguracji na każdej maszynie osobno, ale możemy zrobić to na jednej i wyeksportować profil. Uzyskany w ten sposób plik konfiguracyjny aplikujemy na innych hostach. Oszczędza to wiele czasu administratora, konfigurującego środowisko.

Aby utworzyć profil za pomocą narzędzia Asysta podczas konfigurowania wybierz z menu *Konfiguruj* opcję *Eksportuj profil* → *Dostosuj* (Rysunek 7.10).



Rysunek 7.10 – Eksportowanie profilu

Rysunek 7.11 pokazuje opcje wymagane do eksportu profilu.



Rysunek 7.11 – Okno Dostosuj profil eksportu

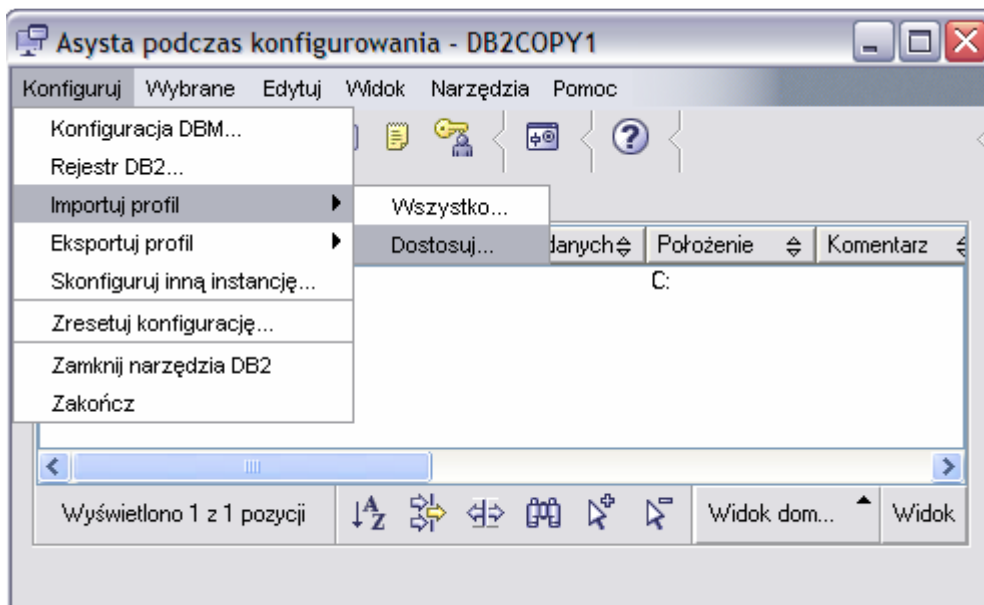
Rysunek 7.12 pokazuje rezultat uruchomienia *Eksportuj* w menu *Dostosuj profil eksportu*.



Rysunek 7.12 – Wynik eksportowania profilu

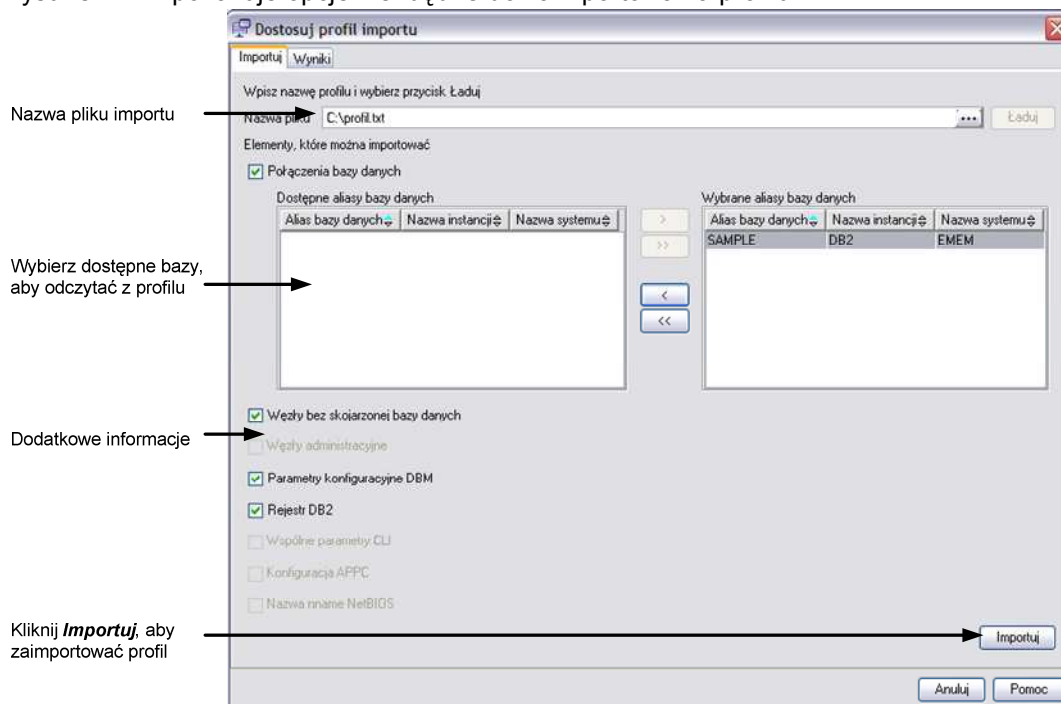


Aby zaimportować utworzony wcześniej profil z poziomu narzędzia Asysta podczas konfigurowania kliknij menu *Konfiguruj* i uruchom *Importuj profil* → *Dostosuj* (Rysunek 7.13).



Rysunek 7.13 – Importowanie profilu

Rysunek 7.14 pokazuje opcje niezbędne do zaimportowania profilu.



**Rysunek 7.14 – Opcja Importuj profil****Laboratorium #6: Użycie narzędzia Asysta podczas konfigurowania**Cel:

Narzędzie Asysta podczas konfigurowania jest używane do szybkiego skonfigurowania połączeń do zdalnych baz. W tym laboratorium skatalogujemy bazę znajdującą się na zdalnym serwerze (jego funkcję będzie pełniła stacja robocza sąsiada), używając funkcji przeszukiwania sieci *Znane system* i *Inne systemy*. Po skatalogowaniu dostęp do bazy danych będzie identyczny, jak w przypadku lokalnych baz. DB2 komunikuje się ze zdalną bazą w tle.

Kroki:

W tym Laboratorium zakładamy, że komputer podłączony jest do sieci. Jeśli nie, możesz użyć swojego komputera jako klienta i serwera jednocześnie, wykonując poniższe instrukcje skonfigurujesz połączenie do własnego systemu.

1. Zapytaj sąsiada (lub instruktora) o następujące informacje:  
Parametry zdalnej bazy danych:

(PR)	Protokół	__TCP/IP__
(IP)	Adres IP lub nazwa hosta	_____
(PN)	Port lub nazwa instancji	_____
(DB)	Nazwa bazy	<u>SAMPLE</u>

Wskazówki

- nazwę hosta w systemie Windows uzyskasz za pomocą polecenia `hostname`
  - adres IP w systemie Windows uzyskasz za pomocą polecenia `ipconfig`
2. Uruchom narzędzie Asysta podczas konfigurowania (dostępny z menu Start).
  3. Wybierz *Wybrane* → *Dodaj bazę za pomocą kreatora*.
  4. W zakładce *Źródło* wybierz *Ręczne konfigurowanie połączenia z bazą danych*. Kliknij *Dalej*, aby przejść do kolejnego kroku.
  5. W zakładce *Protokół* zaznacz opcję TCP/IP. Kliknij *Dalej*, aby

kontynuować.

6. W oknie opcji *TCP/IP* wpisz pełną nazwę hosta lub adres IP oraz numer portu z **Kroku 1**. Kliknij *Dalej*, aby kontynuować.

Uwaga:

Opcja *Nazwa usługi* może zostać wykorzystana, jeśli przypisałeś nazwę usługi do konkretnego portu w pliku usług, na którym nasłuchuje instancja DB2. Jeśli użyjesz tej opcji DB2 szuka usługi w lokalnym pliku usług, nie na serwerze.

7. W opcjach *Baza danych*, wprowadź w pole *Nazwa bazy danych* nazwę bazy danych na zdalnym serwerze z **Kroku 1**. Pole *Alias bazy danych* zostanie automatycznie wypełnione taką samą wartością. Alias bazy danych zostanie wykorzystany przez lokalne aplikacje do łączenia się z nią. Jeśli masz już zdefiniowaną bazę SAMPLE, DB2 nie zezwoli na skatalogowanie bazy o tej samej nazwie. Dlatego musisz wprowadzić inną nazwę. Dla naszego przykładu, zmień *Alias bazy danych* na SAMPLE1. Możemy również dodać opcjonalny komentarz do każdej bazy. Kliknij *Dalej*, aby przejść do kolejnego kroku.
8. W zakładce *Źródło danych* mamy możliwości dodania nowej bazy do źródeł danych ODBC. Dzięki temu DB2 automatycznie rejestruje nowe źródło w systemie Windows. Jeśli nie używasz ODBC odznacz opcję *Zarejestruj bazę danych dla CLI/ODBC*. Kliknij przycisk *Dalej*, aby przenieść się do kolejnego kroku kreatora.
9. W kroku *Opcje węzła* określamy system operacyjny zdalnego serwera, na którym znajduje się baza danych. Z listy wybierz odpowiedni dla serwera, którym którym się łączysz. Nazwa instancji powinna zostać ustawiona na DB2 (lub odpowiednio inna podana przy instalacji). Kliknij *Dalej*, aby kontynuować.
10. Na kolejnej stronie, *Opcje systemu*, kreator umożliwia ponowną weryfikację nazwy hosta, systemu operacyjnego. Kliknij przycisk *Dalej*, aby przejść do kolejnego kroku kreatora.
11. Zakładka *Opcje zabezpieczeń* pozwala na określenie, gdzie i w jaki sposób DB2 będzie przeprowadzało proces uwierzytelniania. Wybierz opcję *Użyj wartości uwierzytelniania z konfiguracji DBM serwera*. Użyta zostanie wtedy metoda określona przez zmienną `AUTHENTICATION` zdalnej instancji. Wciśnij przycisk *Zakończ*, aby skatalogować bazę i zakończyć kreator. Pojawi się monit pytający o potwierdzenie. Kliknij *Testowanie*

---

*połączenia*, aby przetestować połączenie do bazy. Sprawdź poprawność wprowadzanego użytkownika i hasła z tym na zdalnym serwerze (jeśli tylko parametr `AUTHENTICATION=SERVER`). Jeśli wszystko przebiegnie bezbłędnie, katalogowanie zdalnej bazy danych zakończy się sukcesem. Jeśli wystąpią błędy, ponownie prześledź wszystkie kroki i sprawdź wprowadzone parametry (Kliknij przycisk *Zmień*, aby ponownie uruchomić kreator).

12. Uruchom *Centrum sterowania* i spróbuj wyświetlić tabele skatalogowanej bazy.
13. Ponownie uruchom narzędzie *Asysta podczas konfigurowania* i spróbuj skatalogować inną bazę. Tym razem użyj jednak opcji *Przeszukaj sieć*. Dzięki temu nie będziesz musiał ręcznie konfigurować połączenia. W sieciach z dużą liczbą hostów może to być dość czasochłonne.

# 8

## Rozdział 8 – Praca z obiektami bazy danych

W tym rozdziale omówimy obiekty bazy danych, takie jak schematy, tabele, widoki, indeksy, sekwencje i wiele innych. Niektóre zaawansowane obiekty aplikacji, takie jak wyzwalacze, funkcje użytkownika oraz procedury składowane zostaną omówione w Rozdziale 14 „Procedury składowane w SQL PL” oraz w Rozdziale 15 „Inline SQL PL, wyzwalacze i funkcje użytkownika”.

**Ważne:**

W celu uzyskania dodatkowych informacji dotyczących pracy z obiektami bazy danych obejrzyj poniższe nagranie wideo:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4242>

### 8.1 Schematy

Schematy są to *przestrzenie nazw (name spaces)* dla zbioru obiektów baz danych.

Są głównie wykorzystywane do:

- wskazywania właściciela obiektu lub związku z użytkownikiem
- logicznego grupowania spokrewnionych obiektów

Wszystkie nazwy obiektów DB2 składają się z dwóch części; schemat jest pierwszą częścią tej nazwy.

<nazwa\_schematu>.<nazwa\_obiektu>

Pełna nazwa obiektu musi być unikalna. Po połączeniu się z bazą danych, tworzymy lub odwołujemy się do obiektu bez jawnego określenia schematu, DB2 wykorzystuje identyfikator użytkownika, którym się podłączamy do bazy, jako nazwę schematu. Przykładowo, jeżeli połączymy się z bazą SAMPLE jako użytkownik “arfchong” i stworzymy tabelę używając instrukcji CREATE TABLE:

```
CREATE TABLE artists ...
```

pełna nazwa stworzonej tabeli, to w rzeczywistości arfchong.artists.

## 8.2 Tabele

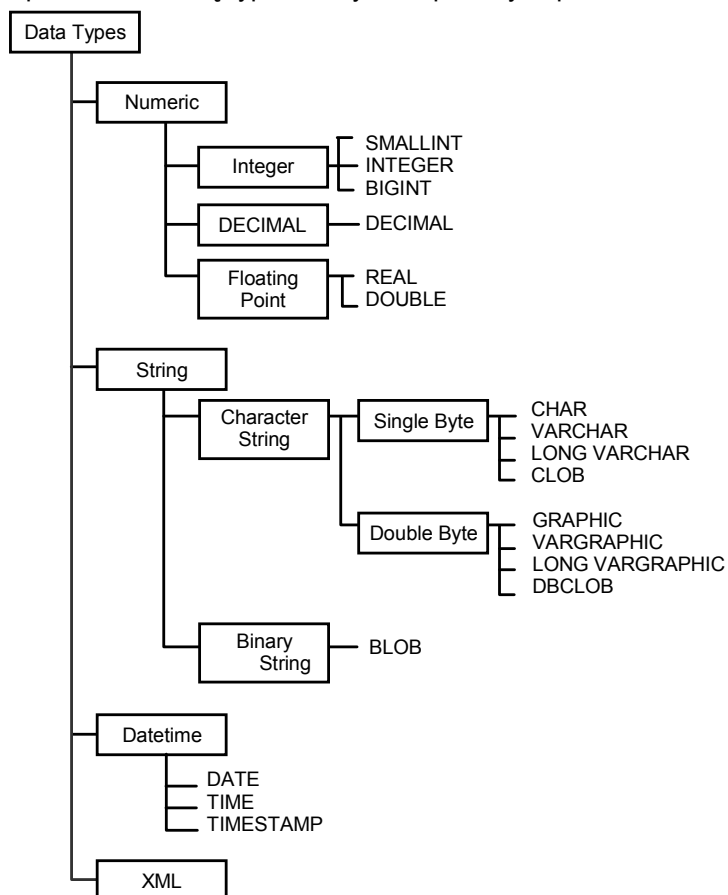
Tabela jest zbiorem spokrewnionych danych, logicznie ustawionych w kolumny i wiersze. Przykładowa instrukcja poniżej pokazuje, jak tworzyć tabelę używając instrukcji `CREATE TABLE`

```
CREATE TABLE artists
(artno          SMALLINT    not null,
 name          VARCHAR(50)  with default 'abc',
 classification CHAR(1)    not null,
 bio           CLOB(100K)  logged,
 picture       BLOB(2M)    not logged compact
)
IN mytbls1
```

W następnej sekcji zostaną opisane główne części składowe instrukcji `CREATE TABLE`

### 8.2.1 Typy danych

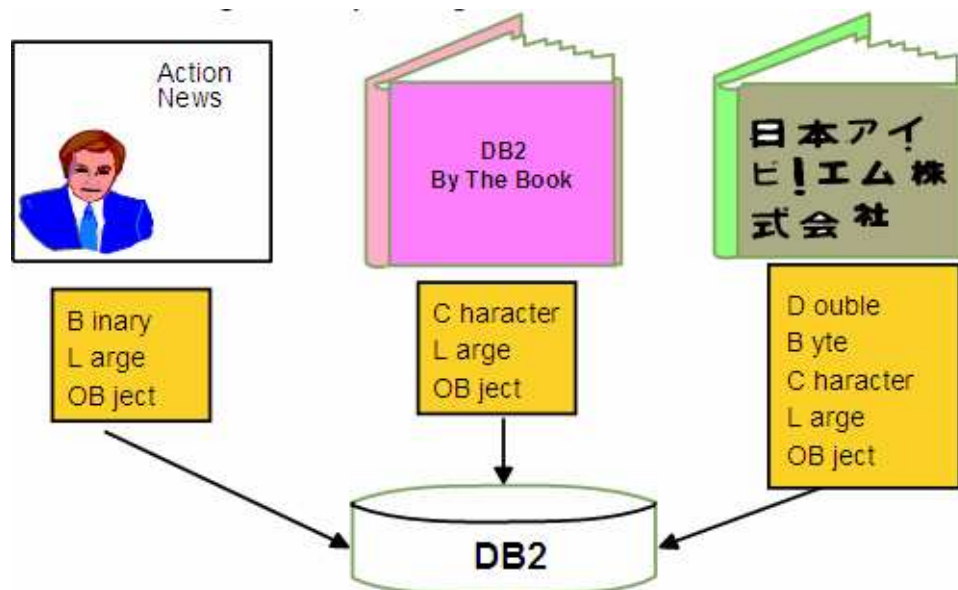
Rysunek 8.1 przedstawia listę typów danych wspieranych przez DB2.



Rysunek 8.1 – Wbudowane typy danych DB2

### Typy danych dużych obiektów

Duże obiekty (ang. Large Objects – LOB) wykorzystywane są do przechowywania dużych łańcuchów znaków, dużych łańcuchów binarnych lub plików, jak zostało pokazane na Rysunku 8.2.



Rysunek 8.2 – Duże typy danych LOB

Nazwy dużych obiektów binarnych są zwykle skrócone dla przejrzystości: binarne duże obiekty to BLOB, znakowe duże obiekty to CLOB, a znakowe (dwubajtowe) znane jako DBCLOB.

### Typy danych zdefiniowane przez użytkownika

DB2 pozwala na zdefiniowanie własnych typów danych, bazujących na wbudowanych typach danych. Takie typy danych nazywamy typami zdefiniowanymi przez użytkownika (ang. User-Defined Types - UDT). UDT są użyteczne, gdy:

- istnieje potrzeba ustanowienia kontekstu wartości
- istnieje potrzeba, aby do DB2 wprowadzono nowy typ danych

Poniższy przykład pokazuje, jak i kiedy używać typów danych UDT.

```
CREATE DISTINCT TYPE POUND AS INTEGER WITH COMPARISONS

CREATE DISTINCT TYPE KILOGRAM AS INTEGER WITH
COMPARISONS

CREATE TABLE person
(f_name    VARCHAR(30),
 weight_p  POUND NOT NULL,
 weight_k  KILOGRAM NOT NULL )
```

W przykładzie tym zostały stworzone dwa typy danych UDT: POUND i KILOGRAM. Obydwa bazują na wbudowanym typie danych INTEGER. Klauzula WITH COMPARISONS jest zdefiniowana jako część składni polecenia. Wskazuje ona, że zostaną stworzone funkcje rzutujące, które będą miały takie same nazwy jak typy danych.

Tabela `person` używa dwóch nowych typów danych użytkownika w kolumnach, odpowiednio `weight_p` i `weight_k`. Jeżeli zostanie wykonana następująca instrukcja:

```
SELECT F_NAME FROM PERSON
WHERE weight_p > weight_k
```

zostanie zwrócony błąd, ponieważ zostały porównane dwie kolumny różnych typów danych. Nawet jeśli `weight_p` i `weight_k` używają typów danych odpowiednio POUND i KILOGRAM i obydwa zostały stworzone bazując na typie danych INTEGER, nie jest możliwe porównanie tych typów. To jest dokładnie to, czego oczekiwaliśmy, ponieważ w rzeczywistości, co oznaczałoby porównanie między funtami a kilogramami? Nie miałyby żadnego sensu.

W następnym przykładzie, będziemy porównywać kolumny `weight_p` z `integer`, jednakże te dwa typy danych są inne i dlatego otrzymamy błąd, dopóki nie użyjemy funkcji rzutujących.

Jak widzimy, w instrukcji poniżej użyta została funkcja rzutująca POUND(), zatem porównanie jest możliwe. Jak pokazaliśmy wcześniej, funkcja rzutująca POUND() została stworzona wraz z UDT, w momencie, gdy wywołaliśmy klauzule WITH COMPARISONS w instrukcji CREATE DISTINCT TYPE.

```
SELECT F_NAME FROM PERSON
WHERE weight_p > POUND(30)
```

### Wartości Null

Wartość Null reprezentuje stan nieznaną. Jednakże w instrukcji CREATE TABLE można definiować kolumny używając klauzuli NOT NULL. Zapewnia ona, że kolumna będzie zawierać znany typ danych. Można również określić wartość domyślną dla pola kolumny, jeżeli NOT NULL jest zadeklarowana. Poniższa instrukcja przedstawia przykład takiego zachowania:

```
CREATE TABLE Staff (
  ID          SMALLINT NOT NULL,
  NAME       VARCHAR(9),
  DEPT       SMALLINT NOT NULL with default 10,
  JOB        CHAR(5),
  YEARS      SMALLINT,
  SALARY     DECIMAL(7,2),
  COMM       DECIMAL(7,2) with default 15
)
```



### 8.2.2 Kolumna identyfikująca

Kolumna identyfikująca jest kolumną numeryczną, która automatycznie generuje unikalne wartości numeryczne dla każdego wstawionego wiersza. W tabeli może być tylko jedna kolumna identyfikująca.

Są dwie metody generowania wartości dla kolumny identyfikującej, w zależności jak została ona zdefiniowana.

- **Generowane zawsze (ang. generated always):** wartości są generowane zawsze przez DB2. Aplikacje nie mogą dostarczyć jawnie wartości.
- **Generowane przez wartość domyślną (ang. generated by default):** wartości mogą być jawnie dostarczone przez aplikacje. Jeśli nie zostaną podane, wtedy zostaną wygenerowane przez DB2. DB2 nie gwarantuje unikalności. Opcja ta przeznaczona jest dla rozprzestrzeniania danych oraz usuwania i ponownego ładowania danych do tabeli.

Przeanalizujmy następujący przykład:

```
CREATE TABLE subscriber(subscriberID INTEGER GENERATED ALWAYS AS
                        IDENTITY (START WITH 100
                        INCREMENT BY 100),
                        firstname VARCHAR(50),
                        lastname  VARCHAR(50) )
```

W przykładzie, kolumna subscriberID jest typu INTEGER i została zdefiniowana jako kolumna identyfikująca, która zawsze generuje wartości. Wartości generowane będą się zaczynać od 100 i będą powiększane o 100.

### 8.2.3 Sekwencje

Sekwencje generują unikalne liczby na poziomie całej bazy danych. W odróżnieniu od kolumn identyfikujących, sekwencje nie są zależne od tabeli. Następująca instrukcja pokazuje przykładowe użycie sekwencji:

```
CREATE TABLE t1 (salary int)

CREATE SEQUENCE myseq
  START WITH 10
  INCREMENT BY 1
  NO CYCLE

INSERT INTO t1 VALUES (nextval for myseq)
INSERT INTO t1 VALUES (nextval for myseq)
INSERT INTO t1 VALUES (nextval for myseq)
```

```
SELECT * FROM t1

SALARY
-----
          10
          11
          12
3 record(s) selected.
```

```
SELECT prevval for myseq FROM sysibm.sysdummy1

1
-----
          12
1 record(s) selected
```

PREVVAL dostarcza aktualnej wartości sekwencji, podczas gdy NEXTVAL dostarcza wartość następną.

Powyższy przykład wykorzystuje również tabelę SYSIBM.SYSDUMMY1. Jest to słownik systemowy, który zawiera jedną kolumnę i jeden wiersz. Może być używany w sytuacji, gdy zapytanie wymaga wyniku, bazując jedynie na jednej wartości zwracanej. Słowniki systemowe są opisane w następnym rozdziale.

#### 8.2.4 Tabele słownika systemowego

Każda baza danych DB2 posiada własne tabele słownika systemowego oraz widoki. Przechowywane są tam *meta dane* o obiektach bazy danej. Możemy wykonywać zapytania do tych tabel, tak samo jak do zwykłych tabel bazodanowych. Do identyfikacji tabel słownika systemowego wykorzystywane są trzy schematy:

- SYSIBM: tabele podstawowe, optymalizowane dla użytku DB2
- SYSCAT: widoki oparte na tabelach SYSIBM, optymalizowana dla łatwości użytkowania
- SYSSTAT: statystyki bazy danych

Poniżej przedstawione są przykładowe widoki słownika:

- SYSCAT.TABLES
- SYSCAT.INDEXES
- SYSCAT.COLUMNS
- SYSCAT.FUNCTIONS
- SYSCAT.PROCEDURES

### 8.2.5 Tabele tymczasowe

Tabele tymczasowe są tabelami tworzonymi w pamięci. Są one wykorzystywane przez daną aplikację i automatycznie usuwane, kiedy aplikacja ta zakończy działanie. Dostęp do tabeli tymczasowej możliwy jest tylko przez aplikację, która ją stworzyła. Żaden wpis nie istnieje w tabelach słownika systemowego DB2. Dostęp do tabel tymczasowych jest bardzo wydajny, ponieważ nie są odczytywane żadne informacje z tabel słownika, nie ma blokowania wierszy, nie ma domyślnego logowania (logowanie jest opcjonalne) oraz sprawdzania autoryzacji. Tabele tymczasowe wspierają także indeksy. Każdy standardowy indeks może być stworzony dla tabel tymczasowych. Można również uruchomić narzędzie RUNSTATS dla takich tabel.

Tabele tymczasowe przechowywane są wewnątrz tymczasowej przestrzeni tablicowej użytkownika (ang. user temporary table space), która musi być zdefiniowana wcześniej, przed utworzeniem jakiegokolwiek tabeli tymczasowej.

Poniższe instrukcje pokazują, jak można utworzyć przykładowe tabele tymczasowe:

```
CREATE USER TEMPORARY TABLESPACE apptemps
  MANAGED BY SYSTEM USING ('apptemps');
```

```
DECLARE GLOBAL TEMPORARY TABLE tempemployees
  LIKE employee NOT LOGGED;
```

```
DECLARE GLOBAL TEMPORARY TABLE tempdept
  (deptid CHAR(6), deptname CHAR(20))
  ON COMMIT DELETE ROWS NOT LOGGED;
```

```
DECLARE GLOBAL TEMPORARY TABLE tempprojects
  AS ( fullselect ) DEFINITION ONLY
  ON COMMIT PRESERVE ROWS NOT LOGGED
  WITH REPLACE IN TABLESPACE apptemps;
```

W momencie, gdy tabele tymczasowe są tworzone, ich schematem jest SESSION i musi być on jawnie określony. Identyfikator użytkownika użyty do tworzenia tabeli tymczasowej, będzie miał do niej wszystkie uprawnienia. Każda aplikacja, która stworzy tabelę tymczasową, będzie miała jej własną niezależną kopię, jak pokazano na Rysunku 8.5.



Rysunek 8.5 – Zakres tabel tymczasowych

## Laboratorium #7: Tworzenie nowej tabeli

### Cel

Do tej pory do ilustrowania omawianych pojęć używaliśmy tabel istniejących w bazie danych *SAMPLE*. W końcu nadejdzie potrzeba, aby stworzyć własne tabele w bazie danych. W Laboratorium tym wykorzystasz *Kreator tworzenia tabel* do stworzenia dwóch nowych tabel w bazie *SAMPLE*.

### Kroki

1. Uruchom *Kreator tworzenia tabel*, jak wcześniej zostało pokazane na prezentacji (*Centrum sterowania* -> *Wszystkie bazy danych* -> *SAMPLE* -> (prawy przycisk myszki) *Tabele* -> *Utwórz*)
2. Określ nazwę tabeli, zdefiniuj kolumny oraz ograniczenia. Tabela będzie wykorzystywana do przechowywania w bazie danych *SAMPLE* informacji o dostawach do biura wykorzystywanych przez projekt. Za każdym razem, kiedy nowa dostawa zostanie zakupiona, dodawany będzie wiersz do tabeli.
  - `product_id`: unikalny identyfikator każdego zakupionego artykułu
  - `description`: opis artykułu
  - `quantity`: ilość artykułów
  - `cost`: cena artykułu
  - `image`: obrazek artykułu (jeśli dostępny)
  - `project_num`: projekt, dla którego został zakupiony artykuł
3. Na pierwszej stronie kreatora w polu określającym nazwę schematu wpisz identyfikator użytkownika, którym jesteś aktualnie zalogowany. Użyj następującej nazwy dla tabeli: *SUPPLIES*. Możesz także opcjonalnie wpisać komentarz. Naciśnij przycisk *Dalej*, aby przejść do następnej strony kreatora.
4. Na tej stronie można dodać kolumny do tabeli. Naciśnij przycisk *Dodaj*, aby dodać kolumny.

Wpisz nazwę kolumny *“product\_id”* i zaznacz typ danych: *INTEGER*. Oznacz *Dopuszczalna wartość pusta* i naciśnij przycisk *Zastosuj*, aby zdefiniować kolumnę.

Powtórz te kroki dla pozostałych kolumn w tabeli, używając opcji przedstawionych w tabeli poniżej. Gdy wszystkie kolumny zostaną dodane (*Zastosuj*) naciśnij przycisk *OK*. Lista kolumn, które zostały zdefiniowane, zostanie wyświetlona w oknie kreatora. Naciśnij przycisk *Dalej*, aby przejść do następnej strony kreatora.

Nazwa kolumny	Atrybuty
product_id (pełna)	INTEGER, NOT NULL
description	VARCHAR, długość 40, NOT NULL
quantity	INTEGER, NOT NULL

---

cost	DECIMAL, precyzja 7, skala 2, NOT NULL
image	BLOB, 1MB, NULLABLE, NOT LOGGED
project_num	CHAR, długość 6, NOT NULL

Ważne: Opcja NOT LOGGED może być określona podczas deklaracji kolumny typu LOB. Jest to obowiązkowe dla kolumn większych niż 1GB. Jest także generalnie zalecane dla pól typu LOB większych niż 10MB, gdyż podczas wprowadzane modyfikacje mogą szybko zapełnić pliki dziennika. Nawet jeśli NOT LOGGED jest użyte, zmiany w plikach LOB wprowadzone podczas transakcji nadal mogą zostać wycofane. Zwróć także uwagę, że kolumna „image” jako jedyna została zdefiniowana jako kolumna „NULLABLE”. Jak myślisz, dlaczego?

5. Wszystkie informacje wymagane do stworzenia tabeli zostały już wprowadzone. Poprzez pominięcie pozostałych stron, zostaną wybrane domyślne wartości dla tych opcji. Można zawsze zdefiniować klucze oraz ograniczenia po tym, jak tabela zostanie stworzona.
6. Dodaj ograniczenia do tabeli dodając wartość ograniczającą dla kolumny *quantity*. Na stronie *Ograniczenia* kreatora, naciśnij przycisk *Dodaj*. W polu *Sprawdź nazwę* wpisz: *valid\_quantities*. W polu *warunek sprawdzania* wpisz: *quantity > 0*

Naciśnij przycisk *OK*. W zakładce *Ograniczenia* powinieneś zobaczyć podsumowanie dotyczące ograniczenia, które właśnie zostało dodane. Naciśnij przycisk *Dalej*, aby przejść do następnej strony kreatora.

7. Korzystając z kreatora możesz kontynuować wprowadzanie zmian ustawień dla reszty parametrów tabeli. Ewentualnie możesz przeskoczyć do strony *Podsumowanie*, albo po prostu nacisnąć przycisk *Zakończ*, aby utworzyć tabelę.
8. W Centrum sterowania kliknij na folder *Tabele* w drzewie obiektów dla bazy danych *SAMPLE*. Tabela, która właśnie została utworzona, powinna pojawić się na liście. Może być konieczne odświeżenie widoku Centrum sterowania, aby zobaczyć wprowadzone zmiany.

## 8.3 Widoki

Widok jest reprezentacją danych z tabel. Dane dla widoku nie są przechowywane osobno, ale uzyskiwane wtedy, kiedy widok jest wywoływany. Zagnieżdżone widoki, czyli widoki, które zostały stworzone w oparciu o inne widoki, są także wspierane przez DB2. Wszystkie informacje dotyczące widoków przechowywane są w następujących tabelach słownika systemowego DB2: SYSCAT.VIEWS, SYSCAT.VIEWDEP i SYSCAT.TABLES. Poniżej zaprezentowany jest przykład pokazujący, jak tworzyć i używać widoków.

```
CONNECT TO MYDB1;

CREATE VIEW MYVIEW1
  AS SELECT ARTNO, NAME, CLASSIFICATION
  FROM ARTISTS;

SELECT * FROM MYVIEW1;
```

Output:

ARTNO	NAME	CLASSIFICATION
10	HUMAN	A
20	MY PLANT	C
30	THE STORE	E
...		

## 8.4 Indeksy

Indeks jest uporządkowanym zbiorem kluczy, z których każdy wskazuje na wiersz w tabeli. Indeks może pozwalać na niepowtarzalność, oraz zwiększać wydajność. Oto kilka cech, jakie można zdefiniować dla indeksu:

- indeks może być rosnący albo malejący
- klucze indeksu mogą być unikalne lub nie
- do stworzenia indeksu może być użytych kilka kolumn (taki indeks nazywamy złożonym)
- jeżeli indeks i dane fizyczne są sklastrowane w podobnych sekwencjach, to wtedy jest to indeks sklastrowany. (cluster index)

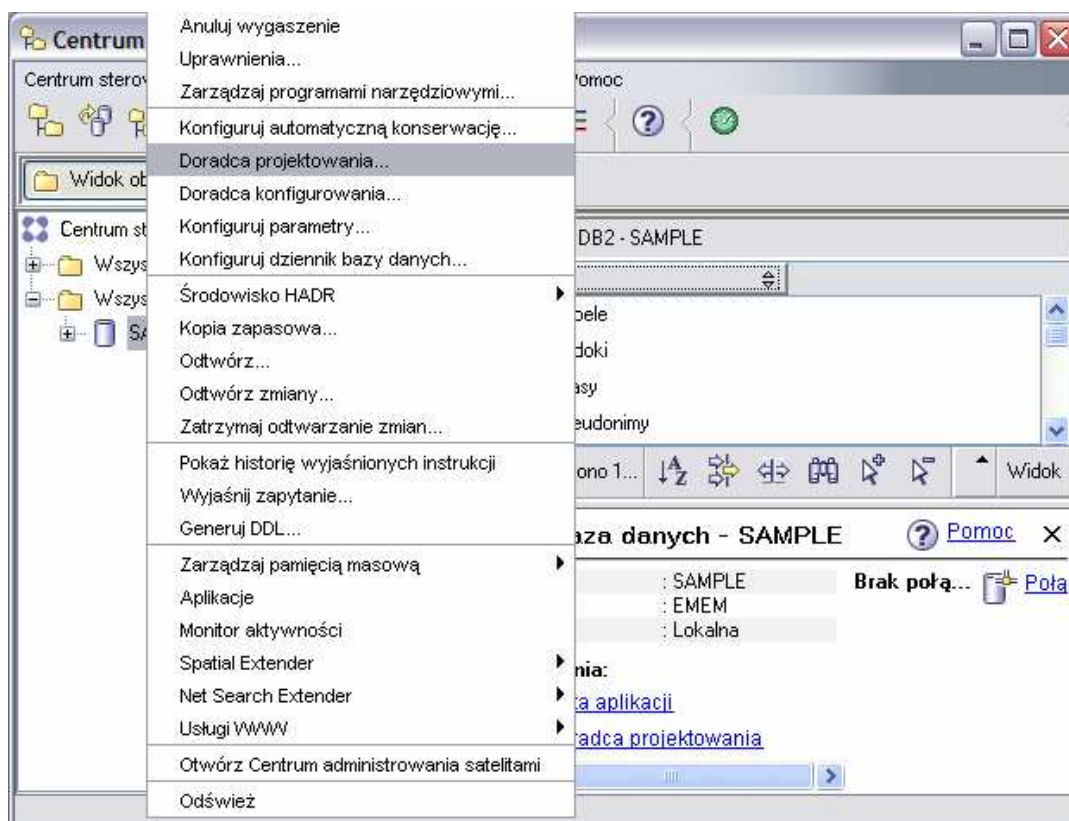
Przykład:

```
CREATE UNIQUE INDEX artno_ix ON artists (artno)
```

### 8.4.1 Doradca projektowania

Doradca projektowania (ang. Design Advisor) jest to znakomite narzędzie, które potrafi doradzić w kwestiach optymalnego projektu bazy danych w oparciu o analizę obciążenia bazy danych (ang. SQL workload). Doradca projektowania może pomóc w projektowaniu indeksów, zmaterializowanych widoków (ang. Materialized Query Tables - MQT), wielowymiarowych tabel klastrowych (ang. Multi-dimension clustering - MDC) oraz partycji bazy

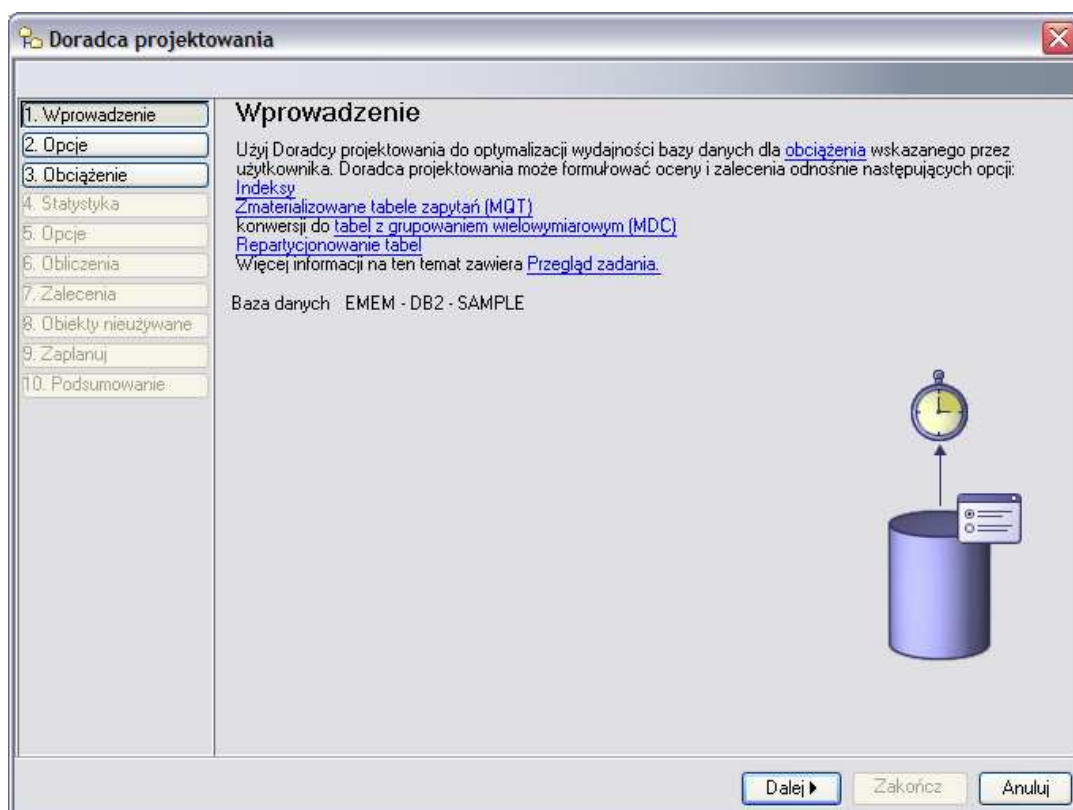
danych. Doradcę projektowania uruchamiamy za pomocą Centrum sterownia, klikając prawym przyciskiem na bazę danych i wybierając *Doradca projektowania*, tak jak pokazano na Rysunku 8.6.



**Rysunek 8.6 – Uruchamianie Doradcy projektowania z Centrum sterownia**

Rysunek 8.7 przedstawia Doradcę projektowania. Przejdź przez kroki kreatora, aby otrzymać rekomendacje projektowe.

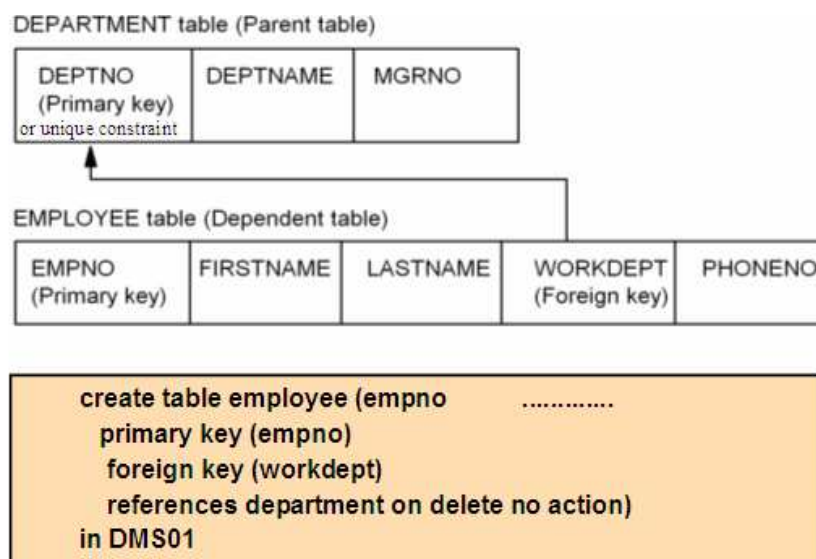




Rysunek 8.7 – Doradca projektowania

## 8.5 Więzy integralności

Więzy integralności pozwalają bazie danych zarządzać relacjami pomiędzy tabelami. Możemy ustanowić relacje rodzic-dziecko pomiędzy tabelami, tak jak pokazano na Rysunku 8.8. Na rysunku tym dwie tabele, DEPARTMENT oraz EMPLOYEE, są połączone poprzez numer departamentu. Kolumna WORKDEPT w tabeli EMPLOYEE może jedynie zawierać numery departamentu, które istnieją w tabeli DEPARTMENT. Jest tak, ponieważ tabela DEPARTMENT w tym przykładzie jest rodzicem (tabelą nadrzędną), a tabela EMPLOYEE jest dzieckiem (albo tabelą zależną). Na rysunku widoczna jest również niezbędna instrukcja CREATE TABLE dla tabeli EMPLOYEE, potrzebna do ustanowienia relacji.



Rysunek 8.8 – Przykład więzów integralności pomiędzy tabelami

Następujące pojęcia są wykorzystywane podczas definiowania więzów integralności:

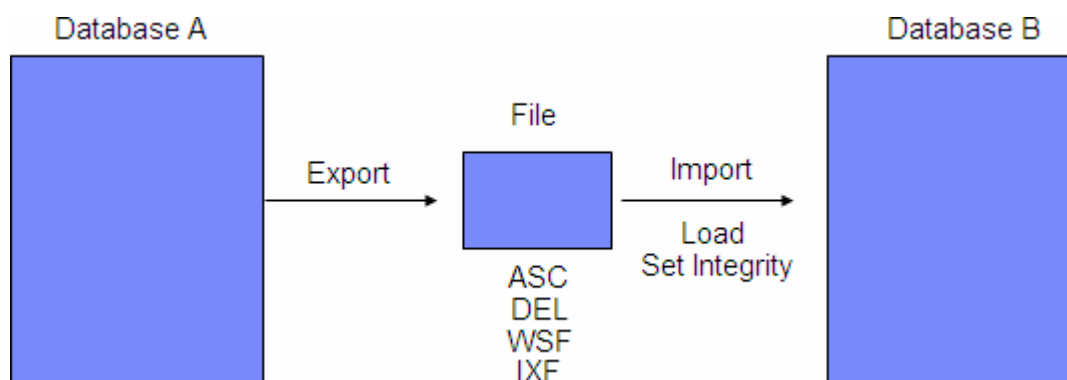
Pojęcie	Opis
Tabela rodzica (tabela nadrzędna)	Główna tabela z danymi, ze zdefiniowanymi kluczami głównymi.
Tabela zależna	Tabela zależna od danych w tabeli rodzica. Zawiera także klucze obce. Do krotki istniejącej w tabeli zależnej musi pasować krotka, która istnieje w tabeli rodzica.
Klucz główny	Definiuje klucz nadrzędny tabeli rodzica. Nie może posiadać wartości null oraz musi być unikalny. Klucz główny może składać się z jednej lub więcej kolumn tabeli.
Klucz obcy	Referencja do klucza głównego w tabeli nadrzędnej.

Dane w tabelach mogą być powiązane z danymi z jednej lub z wielu tabel poprzez więzy integralności. Ograniczenia mogą być narzucone na wartości danych, co oznacza, że mogą być dopasowywane do pewnych właściwości albo reguł biznesowych. Przykładowo, jeżeli kolumna w tabeli przechowuje informację o płci osoby, ograniczenie może być narzucone w taki sposób, aby wymuszać wartość „M” dla mężczyzn i „K” dla kobiet.

# 9

## Rozdział 9 – Narzędzia do przenoszenia danych

Polecenia i narzędzia opisane w tym rozdziale służą do przenoszenia danych pomiędzy tymi samymi bazami, różnymi bazami, na tej samej lub innej platformie sprzętowej. Rysunek 9.1 przedstawia przegląd narzędzi do przenoszenia danych.



Rysunek 9.1 – Narzędzia do przenoszenia danych

Na Rysunku 9.1 widzimy dwie bazy danych: A i B. Używając narzędzia EXPORT możemy wyeksportować tabelę do pliku, który ma jeden z poniższych formatów:

ASC = ASCII  
 DEL = Delimited ASCII  
 WSF = Worksheet format  
 IXF = Integrated Exchange Format

ASC i DEL są to pliki tekstowe, które mogą być otwarte i przeglądane w dowolnym edytorze tekstowym. Formatu WSF możemy używać przy przenoszeniu danych do arkusza kalkulacyjnego, takiego jak Excel lub Lotus® 1-2-3. Format IXF przenosi nie tylko dane, ale również definicje obiektów w postaci instrukcji DDL (Data Definition Language). Jest to wygodne, gdy potrzebujemy odtworzyć tabelę bezpośrednio z pliku IXF, co nie jest możliwe w przypadku innych formatów.

Po wyeksportowaniu danych do pliku, narzędzie IMPORT służy do ponownego zaimportowania tych danych do tabeli. Tabela musi istnieć w przypadku formatów: ASC, DEL i WSF (nie jest to konieczne dla IXF). Innym sposobem na załadowanie danych jest narzędzie LOAD, które jest szybsze i ładuje dane bezpośrednio do stron bazy, bez interakcji

z silnikiem DB2 (nie zostanie sprawdzona poprawność i integralność danych, wyzwalacze zostaną pominięte). Aby zapewnić spójność bazy danych po użyciu polecenia LOAD wykonujemy komendę SET INTEGRITY.

Dalej dokładniej przedstawimy narzędzia EXPORT, IMPORT i LOAD.

### Ważne:

Więcej informacji o narzędziach do przenoszenia danych znajdziesz tutaj:

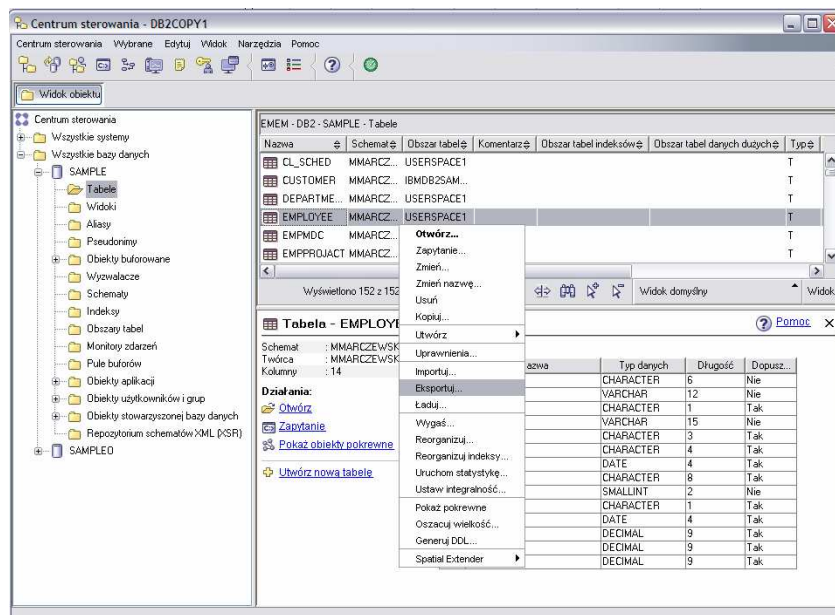
<http://www.channeldb2.com/video/video/show?id=807741:Video:4262>

## 9.1 Narzędzie EXPORT

Narzędzie EXPORT jest wykorzystywane do eksportowania danych z tabeli do wcześniej omawianych typów plików. Faktycznie przetwarzane jest tylko zapytanie SQL SELECT. Poniższy przykład eksportuje do pliku *employee.ixf*, w formacie IXF, 10 wierszy z tabeli *employee*.

```
EXPORT TO employee.ixf OF IXF
  SELECT * FROM employee
  FETCH FIRST 10 ROWS ONLY
```

Zachęcamy do wypróbowania powyższego przykładu. Tabela *employee* jest częścią bazy danych *SAMPLE*, więc musisz najpierw połączyć się do bazy utworzonej w poprzednim rozdziale. Jeśli wolisz pracować z narzędziami graficznymi, możesz wykonać to również za pomocą Centrum sterowania, jak pokazano na Rysunku 9.2.



Rysunek 9.2 – Uruchomienie kreatora eksportowania danych.

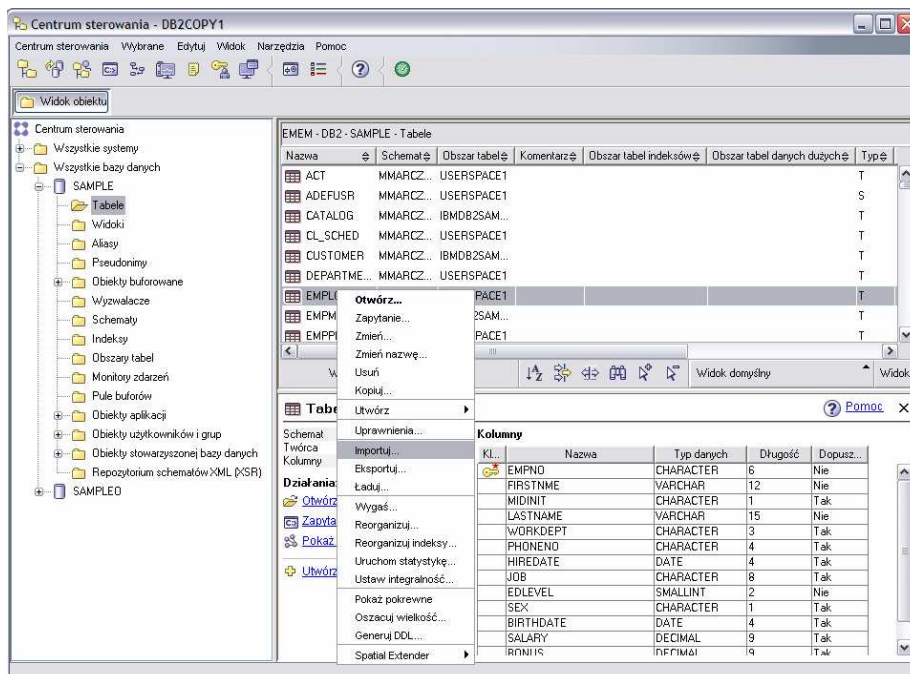
Jak pokazano na Rysunku 9.2, najpierw zaznaczamy tabelę `employee`, a następnie klikamy prawym przyciskiem, aby z menu kontekstowego wybrać opcję *Eksportuj*. Uruchomimy w ten sposób kreatora eksportu tabeli, który poprowadzi nas przez całą operację.

## 9.2 Narzędzie IMPORT

Narzędzie IMPORT używane jest do importowania danych z pliku do tabel. W rzeczywistości podczas operacji wykonywane jest polecenie SQL INSERT. Podczas operacji IMPORT aktywowane są wszystkie wyzwalacze, sprawdzane są więzy integralności oraz używane są pule buforów. Poniższy przykład importuje dane z pliku `employee.ixf` o formacie IXF do tabeli `employee_copy`. Opcja REPLACE\_CREATE jest jedną z wielu dostępnych dla polecenia IMPORT. Powoduje ona, że zawartość tabeli `employee_copy` jest nadpisywana, jeśli tabela ta istniała przed wykonaniem polecenia IMPORT. W przeciwnym wypadku opcja ta utworzy nową tabelę i zaimportuje dane. Zachęcamy do wypróbowania poniższego polecenia (potrzebny będzie wynik polecenia EXPORT z poprzedniego punktu).

```
IMPORT FROM employee.ixf OF IXF
REPLACE_CREATE
INTO employee_copy
```

Jeśli wolisz pracować w środowisku graficznym, możesz uruchomić polecenie IMPORT z poziomu Centrum sterowania przez zaznaczenie tabeli, kliknięcie na niej prawym przyciskiem, a następnie wybranie opcji *Importuj*, tak jak na Rysunku 9.3.



Rysunek 9.3 – Uruchamianie narzędzia IMPORT

### 9.3 LOAD

Narzędzie LOAD jest szybszą metodą na załadowanie danych z pliku do tabeli. Jak wcześniej wspominaliśmy, LOAD omija interakcję z silnikiem DB2, więc wyzwalacze nie są aktywowane, nie są używane pule buforów, a więzy integralności są sprawdzane w osobnym kroku. LOAD jest szybszy niż IMPORT, gdyż działa na niższym poziomie, operując bezpośrednio na stronach danych. Wyróżniamy 3 fazy: LOAD, BUILD i DELETE.

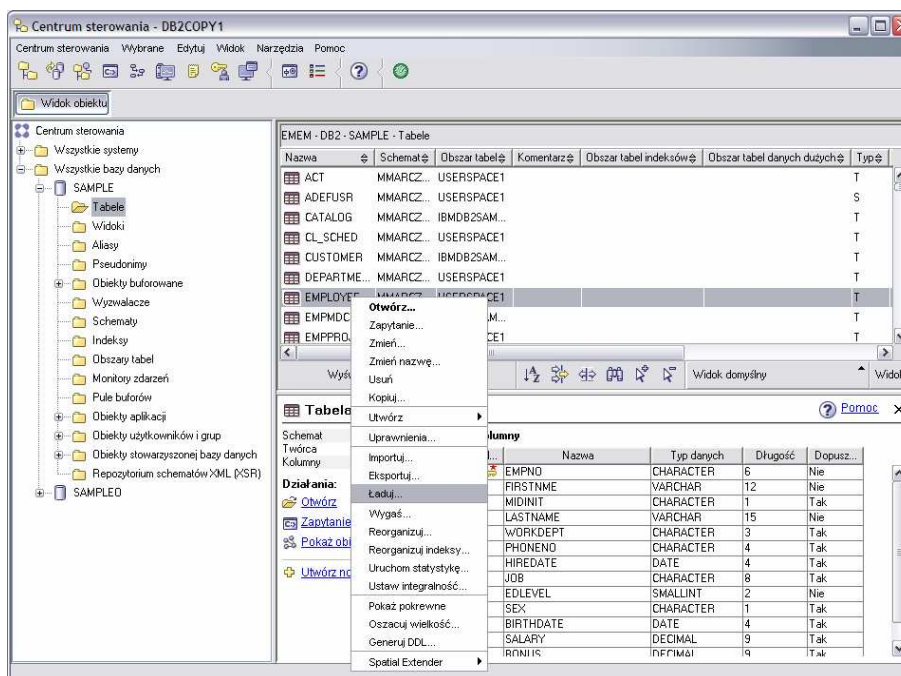
Następujący przykład ładuje dane z pliku IXF *employee.ixf* do tabeli *employee\_copy*. Opcja REPLACE jest jedną z wielu dostępnych dla narzędzia LOAD i powoduje zastąpienie wszystkich rekordów w tabeli *employee\_cop* rekordami wgrywanymi.

```
LOAD FROM employee.ixf OF IXF
  REPLACE INTO employee_copy
```

Po wykonaniu powyższego polecenia przestrzeń tablicowa, w której znajduje się tabela, zmienia status na CHECK PENDING. Następnie musimy wykonać polecenie SET INTEGRITY, aby sprawdzić integralność danych. Pokazuje to poniższy przykład:

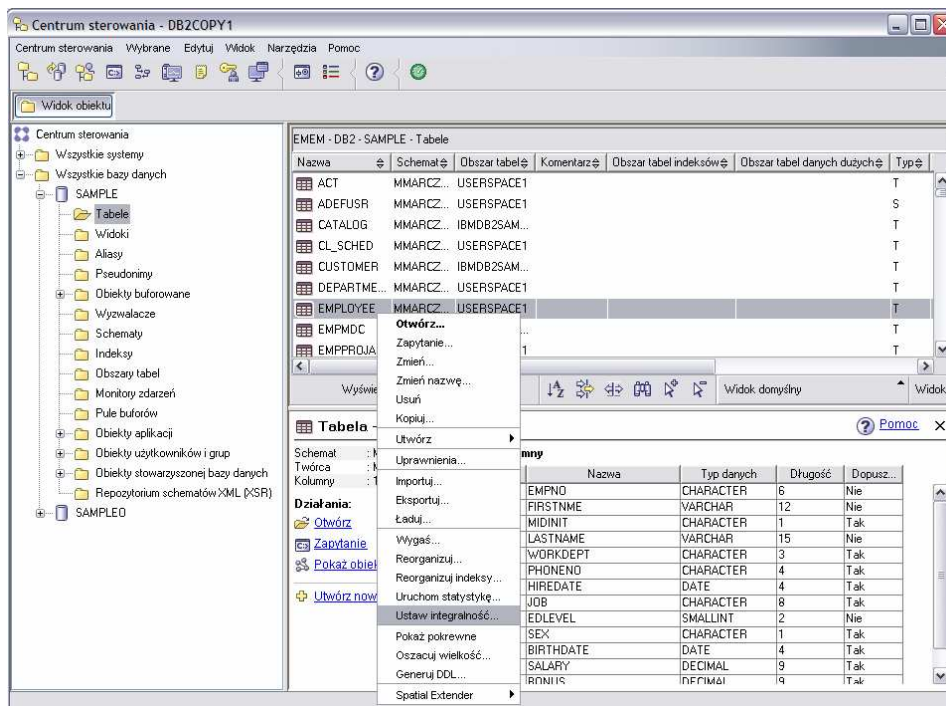
```
SET INTEGRITY FOR employee_copy
  ALL IMMEDIATE UNCHECKED
```

Możemy oczywiście zrobić to z poziomu Centrum sterowania. Uruchamiamy narzędzie LOAD i SET INTEGRITY odpowiednio, jak na Rysunkach 9.4 i 9.5.



Rysunek 9.4 – Uruchamianie kreatora narzędzia LOAD





Rysunek 9.5 – Uruchamianie kreatora SET INTEGRITY (Ustaw integralność)

## 9.4 Narzędzie db2move

Narzędzia EXPORT, IMPORT i LOAD mogą operować w danym momencie tylko na jednej tabeli. Zamiast przygotowywać dla każdej tabeli osobny skrypt, możemy wykorzystać narzędzie db2move, które zrobi to za nas. Wykorzystuje ono wyłączenie pliki typu IXF, a ich nazwy plików są generowane automatycznie. Poniższe przykłady pokazują, jak wykorzystać db2move do eksportu i importu danych znajdujących się w bazie danych SAMPLE.

```
db2move sample export
db2move sample import
```

Centrum sterowania nie ma zaimplementowanej obsługi narzędzia db2move.

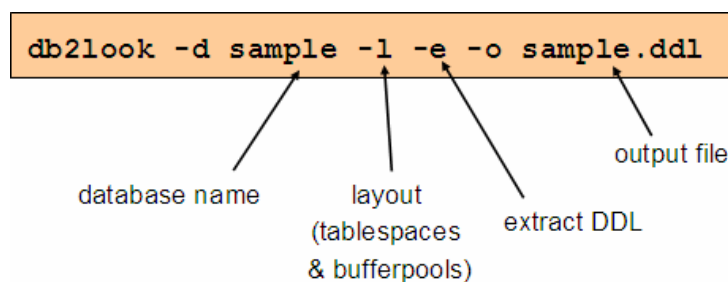
## 9.5 Narzędzie db2look

Podczas, gdy narzędzia EXPORT, IMPORT, LOAD i db2move pozwalają na przenoszenie danych z jednej tabeli do drugiej, w tej samej bazie danych lub pomiędzy różnymi bazami danych, narzędzia db2look możemy użyć do uzyskania pliku skryptowego, zawierającego strukturę DDL, statystyki bazy danych i opcje przestrzeni tablicowych. Plik ten możemy później wykorzystać na innym systemie. Na przykład, jeśli chcemy sklonować bazę danych z serwera DB2 działającego na systemie Linux na serwer działający na systemie Windows, uruchamiamy narzędzie db2look na serwerze Linux, otrzymując strukturę bazy przechowywaną w pliku skryptowym. Uruchamiamy ten skrypt na systemie Windows, otrzymując strukturę bazy danych. Teraz na systemie Linux uruchamiamy narzędzie db2move z opcją export. Następnie kopiujemy otrzymane pliki na serwer DB2 działający na systemie

Windows i uruchamiamy narzędzie `db2move` z opcją `import` lub `load`. Po wykonaniu tych czynności otrzymamy pełną kopię bazy danych na nowej platformie.

Powyższy scenariusz jest używany, gdy pracujemy z bazami danych na różnych platformach systemowych, takich jak Windows czy Linux. Jeśli oba serwery działają pod kontrolą tych samych systemów lepiej użyć poleceń `backup` i `restore`, które upraszczają i przyspieszają cały proces. Będą one omawiane w dalszych rozdziałach tej książki.

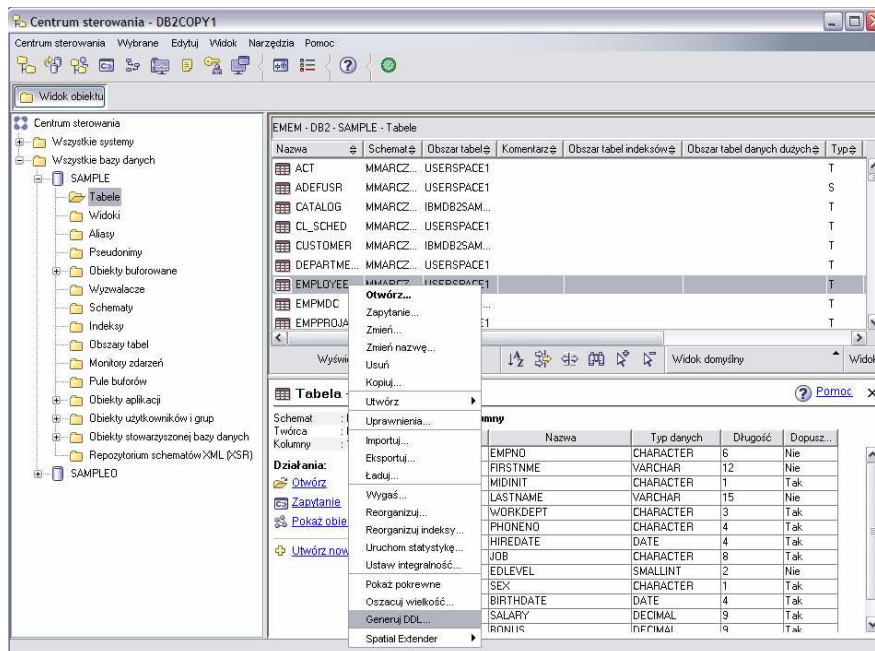
Następny przykład ekstrahuje opis przestrzeni tablicowych i puli buforów razem ze strukturą DDL i przechowuje je w pliku `sample.ddl`. Zalecamy wykonanie tego polecenia i przyjrzenie się temu plikowi.



Polecenie `db2look` ma zbyt wiele opcji, aby zawrzeć je w tej książce. Aby otrzymać pełną listę opcji użyj komendy:

```
db2look -h
```

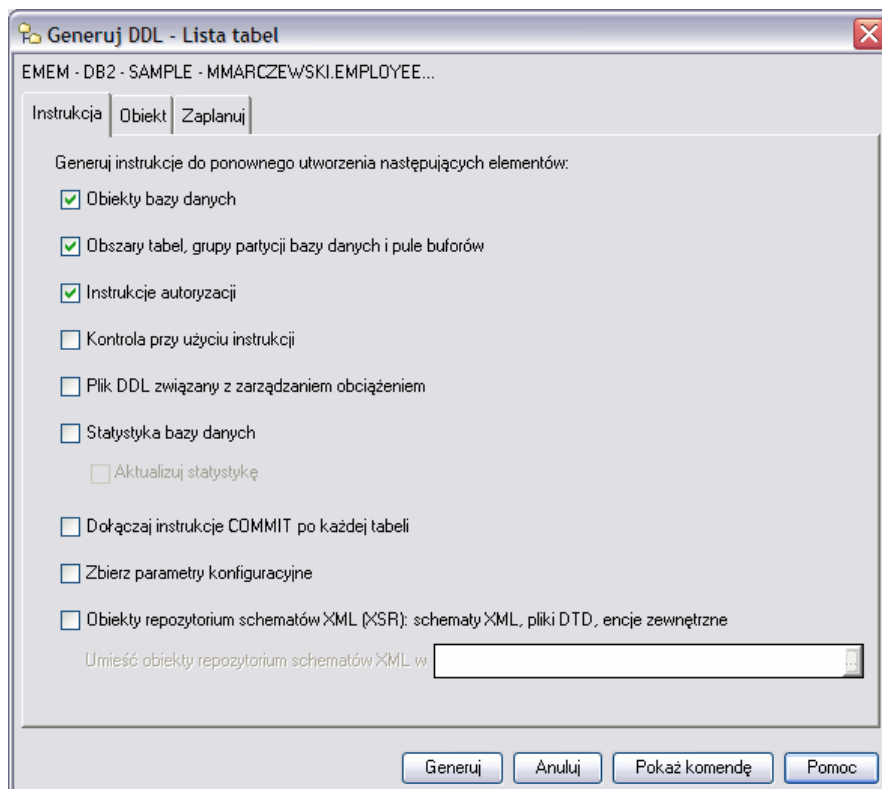
Możemy również użyć Centrum sterowania, jak pokazano na Rysunku 9.6



Rysunek 9.6 – Eksportowanie struktury DDL z Centrum sterowania



Jak przedstawiono na Rysunku 9.6, zaznacz bazę danych, którą chcesz wyeksportować i otrzymać plik DDL. Kliknij na niej prawym przyciskiem myszy i wybierz Generuj DDL. Pojawi się okno, w którym możesz wybrać kilka opcji (Rysunek 9.7).



Rysunek 9.7 – Eksportowanie DDL z poziomu Centrum sterowania

---

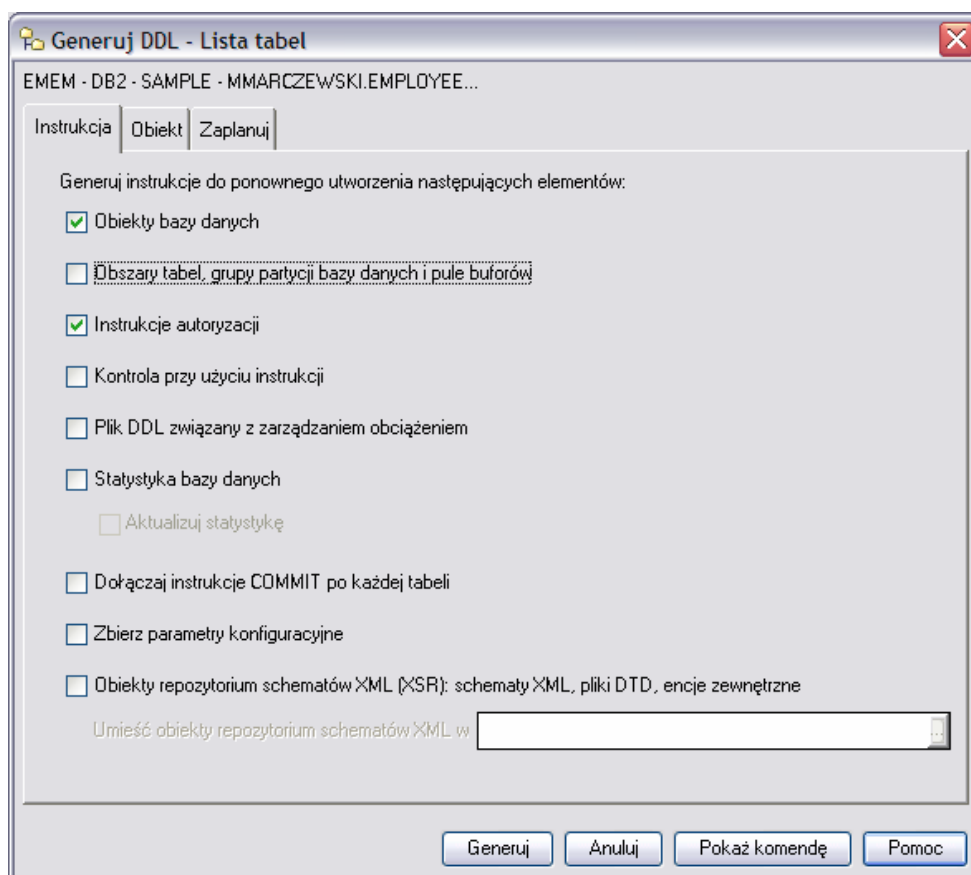
## Laboratorium #8 – Tworzenie pliku DDL dla bazy danych

### Cel

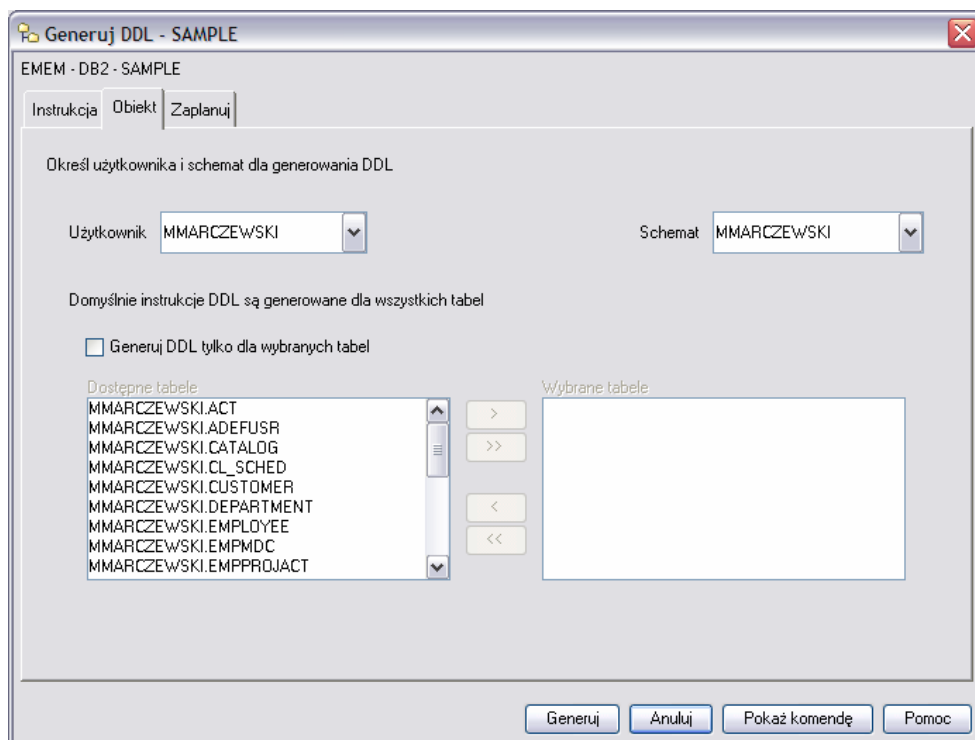
Podczas kopiowania bazy danych naszym celem powinno być maksymalne uproszczenie i powtarzalność całego procesu. Zwykle używamy do tego skryptów SQL, które mogą zostać wykonane zaraz po instalacji DB2. W tym Laboratorium wyeksportujemy definicje obiektów bazy danych EXPRESS (utworzonej w Laboratorium #2) używając Centrum sterowania.

### Kroki

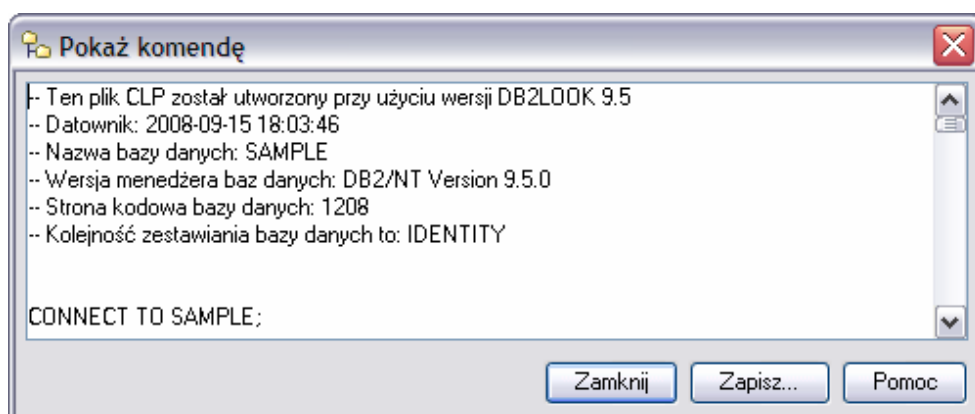
1. Uruchom Centrum sterowania.
2. Kliknij prawym przyciskiem myszy w drzewie obiektów na bazę danych EXPRESS i wybierz Generuj DDL. Otworzy się okno dialogowe Generuj DDL.
3. W oknie *Generuj DDL* ustaw opcje dotyczące eksportu. Jeśli utworzyłeś dodatkowe obiekty (przestrzenie tablicowe, pule buforów) wybierz je tutaj. Jako, że takich obiektów nie tworzyłeś, odznacz to pole. Statystyki bazy danych nie zostały dołączone, ponieważ środowisko produkcyjne prawdopodobnie zawierać będzie inny zestaw statystyk, niż środowisko testowe. Również parametry konfiguracji będą różne. Jeśli jesteśmy przekonani, co do identyczności konfiguracji obu instancji, możemy włączyć dodatkowe pozycje.



- Przejdź do zakładki Obiekt. Tutaj możemy zdefiniować obiekty, z których chcemy wygenerować DDL. W naszym przypadku wybierzmy użytkownika i schemat, w którym utworzyliśmy wszystkie obiekty. Domyślnie tworzona jest struktura DDL dla wszystkich tabel. Kliknij Generuj, aby rozpocząć generowanie struktury.



5. Przyjrzyj się otrzymanemu plikowi DDL. Rezultatem poprzedniego kroku jest skrypt zawierający definicje wybranych obiektów w języku SQL. Uporządkujemy teraz ten skrypt.
6. Utwórz katalog `C:\express` i umieść tam wygenerowany plik DDL nazywając go `schema.ddl` (kliknij przycisk Zapisz).



7. Otwórz zapisany plik w Edytorze komend. (Wskazówka: W oknie Edytora komend wybierz: *Plik => Otwórz*)

8. Pomimo, że naszym celem było jedynie wygenerowanie pliku DDL dla tabel, zauważ, że dołączone zostały również definicje innych obiektów. Przenieś wszystkie deklaracje CREATE TRIGGER do osobnego pliku nazwanego triggers.ddl. Wprawdzie utworzyliśmy tylko jeden wyzwalacz, ale umieszczanie definicji różnych obiektów w osobnych plikach jest dobrą praktyką.
9. Sugerujemy usunięcie również deklaracji:
  - CONNECT TO
  - DISCONNECT

Na tym etapie powinniśmy mieć dwa skrypty:

Plik DDL dla tabel, widoków, indeksów, więzów integralności:

```
C:\express\schema.ddl
```

DDL dla wyzwalaczy:

```
C:\express\triggers.ddl
```

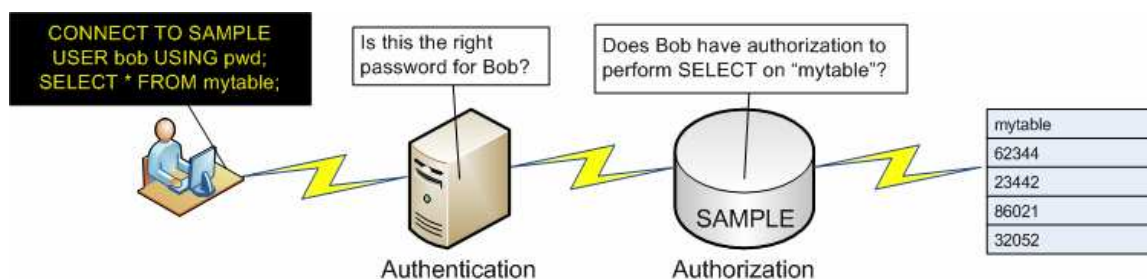
10. Uporządkujmy ostatecznie nasz skrypt:
  - usuń niepotrzebne komentarze (np. -- CONNECT TO...)
  - oddziel funkcje i procedury przenosząc je do osobnych plików (użyteczne, gdy używamy dużej ilości funkcji i procedur). Możemy również pogrupować je według funkcji lub używanej aplikacji (np. billing.ddl, math.ddl, stringfunc.ddl itd.)
11. Zauważ również, że DB2 używa specjalnego znaku (@), aby oznaczyć koniec wyzwalacza, funkcji czy procedury. Jest to niezbędne, aby oddzielić definicję CREATE <obiekt> od innych definicji obiektów proceduralnych.



# 10

## Rozdział 10 – Bezpieczeństwo bazy danych

W tym rozdziale omówimy, jak w DB2 zapewniane jest bezpieczeństwo. Rysunek 10.1 dostarcza podstawowego opisu.



Rysunek 10.1 - Zarys bezpieczeństwa w DB2

Jak pokazano na Rysunku 10.1, bezpieczeństwo w DB2 składa się z dwóch elementów:

### Uwierzytelnianie

Jest to proces, w którym potwierdzana jest tożsamość użytkownika. Uwierzytelnianie jest wykonywane przez obiekt bezpieczeństwa poza DB2 (zazwyczaj przez system operacyjny, usługi autoryzacji sieciowej, albo zwyczajnie wbudowane wtyczki uwierzytelniające). Domyślną opcją jest uwierzytelnienie oparte na systemie operacyjnym.

Kiedy używane jest uwierzytelnienie oparte na systemie operacyjnym, identyfikator użytkownika i hasło przesyłane są do serwera bazy danych (np. jako część instrukcji połączenia). Serwer bazodanowy wywołuje wtedy uwierzytelnienie przez system operacyjny w celu zweryfikowania identyfikatora i hasła użytkownika.

### Autoryzacja

W tym etapie DB2 sprawdza, czy uwierzytelniony użytkownik może wykonywać żądane operacje. Informacje dotyczące autoryzacji są przechowywane w katalogu DB2 i w pliku konfiguracyjnym DBM.

Przykładowo, na Rysunku 10.1 użytkownik "bob" podłącza się do bazy SAMPLE instrukcją:

```
CONNECT TO sample USER bob USING pwd
```

Zarówno "bob" i "pwd" są przesyłane do systemu operacyjnego albo zewnętrznego obiektu uwierzytelniającego, żeby dokonać uwierzytelnienia oraz weryfikacji, czy użytkownik „bob”

jest zdefiniowany oraz czy dostarczone hasło pasuje do tego użytkownika. Jeżeli ta część zakończy się sukcesem, system operacyjny zwróci kontrolę bezpieczeństwa do DB2. Kiedy użytkownik „bob” wykona polecenie takie jak:

```
SELECT * FROM mytable
```

DB2 przejmuje kontrolę bezpieczeństwa, aby autoryzować użytkownika i potwierdzić, że „bob” posiada przywilej SELECT do tabeli „mytable”. Jeżeli proces autoryzacji nie powiedzie się, DB2 zwróci komunikat błędu. W przeciwnym razie instrukcja zostanie wykonana na tabeli „mytable”.

**Uwaga:**

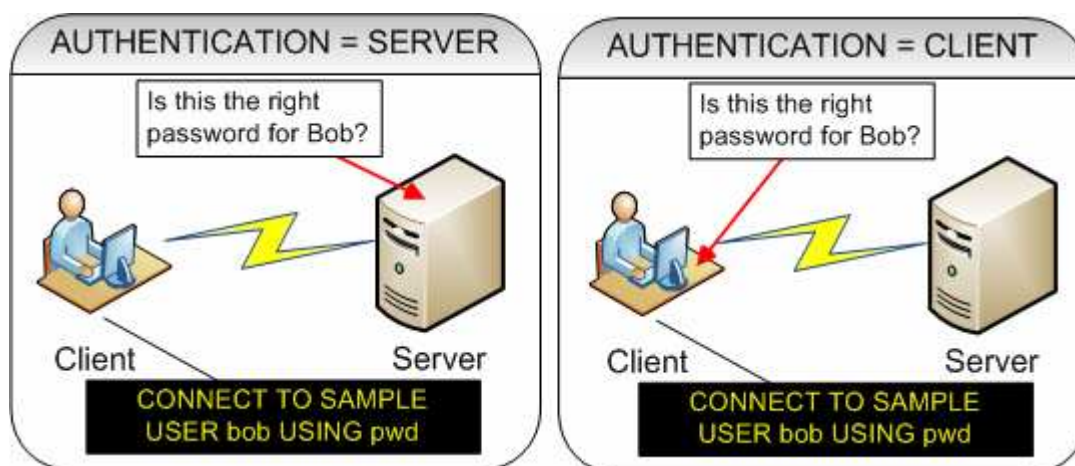
W celu uzyskania dodatkowych informacji dotyczących pracy z bezpieczeństwem DB2, obejrzyj poniższe wideo:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4267>

## 10.1 Uwierzytelnianie

Mimo że w rzeczywistości uwierzytelnianie jest wykonywane przez system operacyjny (albo inny zewnętrzny obiekt bezpieczeństwa), DB2 decyduje na jakim poziomie proces ten zostanie zrealizowany.

Parametr AUTHENTICATION w DBM CFG, ustawiony w serwerze DB2, ma duży zakres możliwych wartości. Przykładowo, gdy parametr ten ustawiony jest jako SERVER (domyślnie), uwierzytelnianie jest wykonywane przez system operacyjny lub zewnętrzny obiekt bezpieczeństwa na serwerze. Jednakże, jeżeli parametr AUTHENTICATION jest ustawiony na CLIENT, uwierzytelnianie jest wykonywane przez system operacyjny (zewnętrzny obiekt bezpieczeństwa) po stronie klienta. Zostało to pokazane na Rysunku 10.2.



Rysunek 10.2 – Uwierzytelnienie w DB2



Parametr AUTHENTICATION może być ustawiony jako każda wartość z Tabeli 10.1.

Polecenie	Opis
SERVER (domyślnie)	Uwierzytelnienie odbywa się po stronie serwera
KLIENT	Uwierzytelnienie odbywa się po stronie klienta
SERVER_ENCRYPT	Tak jak SERVER z wyjątkiem, że ID użytkownika oraz hasło są szyfrowane
KERBEROS	Uwierzytelnienie odbywa się poprzez użycie mechanizmu bezpieczeństwa Kerberos
SQL_AUTHENTICATION_DATAENC	Uwierzytelnienie serwera oraz połączenie musi używać szyfrowania danych
SQL_AUTHENTICATION_DATAENC_CMP	Podobnie jak wyżej, z wyjątkiem szyfrowania danych używanych tylko wtedy, kiedy jest to możliwe
GSSPLUGIN	Uwierzytelnienie używa GSS API, które opiera się na wtyczce mechanizmu bezpieczeństwa

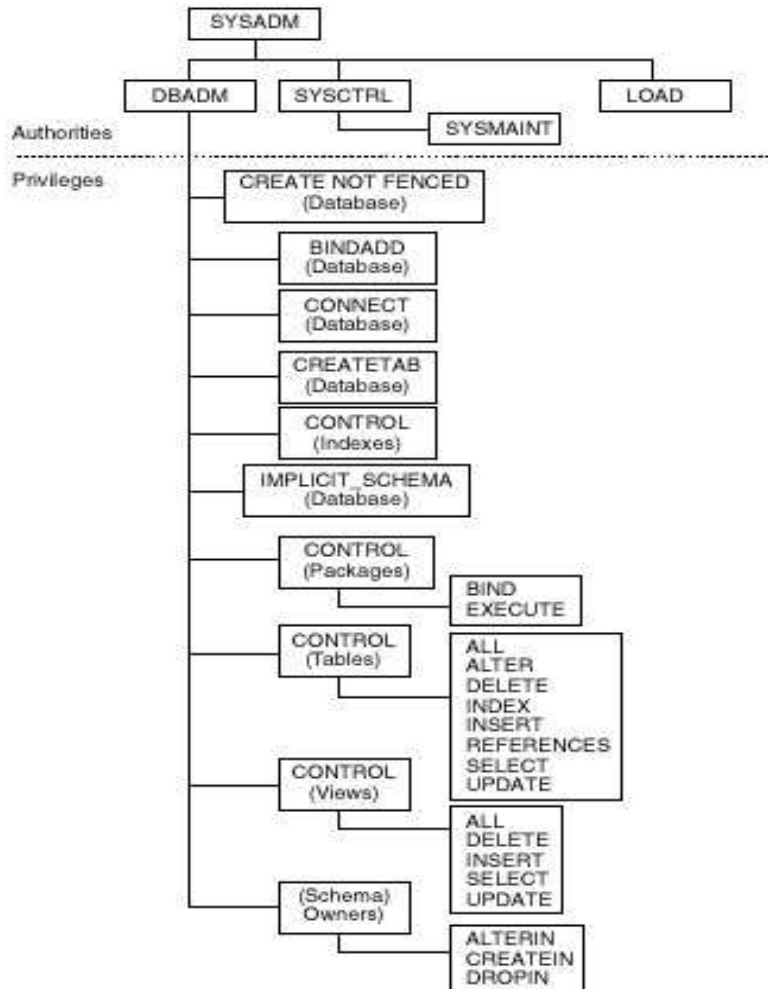
**Tabela 10.1 – Poprawne wartości parametru AUTHENTICATION**

## 10.2 Autoryzacja

Autoryzacja składa się z przywilejów i uprawnień, które są przechowywane w tabelach systemowych DB2 oraz są zarządzane przez DB2.

Przywileje pozwalają użytkownikowi na wykonywanie pojedynczych operacji na bazie danych, takich jak CREATE, UPDATE, DELETE, INSERT, itp.

Upewnienie jest predefiniowaną rolą składającą się z kilku przywilejów. Rysunek 10.3 pokazuje różne przywileje i uprawnienia w DB2.



Rysunek 10.3 – Uprawnienia i przywileje

Tabela 10.2 przedstawia funkcje, które mogą być wykonywane przy różnych uprawnieniach. Jak widać, SYSADM ma największy zakres dostępnych funkcji, a SYSMON ma najmniejszy.

<i>Operacja</i>	<i>SYSADM</i>	<i>SYSCTRL</i>	<i>SYSMAINT</i>	<i>SYSMON</i>	<i>DBADM</i>	<i>LOAD</i>
Aktualizacja DBM	T					
CFG	T					
Grant/revoke DBADM	T					
Ustanowić/zmienić SYSCTRL	T					
Ustanowić/zmienić SYSMAINT	T					
Ustanowić/zmienić SYSMON	T					
Wymuszone odłączenie użytkownika od bazy	T	T				
Tworzenie/usuwanie baz danych	T	T				
Przywracanie do nowej bazy	T	T				
Aktualizacja DB CFG	T	T	T			
Kopia zapasowa bazy danych / przestrzeni tablicowych	T	T	T			
Przywracanie do istniejącej bazy	T	T	T			
Wykonanie odzyskiwania roll-forward	T	T	T			
Uruchamianie/zatrzymanie instancji	T	T	T			
Przywrócenie przestrzeni tablicowych	T	T	T			
Uruchamianie śladu	T	T	T	T		
Uzyskiwanie obrazów stanu monitora	T	T	T			
Odpytywanie stanu przestrzeni tablicowych	T	T	T			
Skrócenie plików logów historii	T	T	T			
Tryb konserwacyjny przestrzeni tablicowych	T	T	T		T	T

Operacja	SYSADM	SYSCTRL	SYSMAINT	SYSMON	DBADM	LOAD
LOAD tabel	T				T	T
Zaznaczenie/odznaczenie sprawdzania stanu oczekującego	T				T	
Tworzenie/porzucanie zdarzeń monitora	T				T	

**Tabela 10.2 – Przywileje i uprawnienia w DB2**

Aby nadać grupie uprawnienie SYSADM, SYSCTRL lub SYSMAINT, parametry SYSADM\_GROUP, SYSCTRL\_GROUP i SYSMAINT\_GROUP pliku konfiguracyjnego menedżera instancji (DBM) mogą być przypisane do grupy użytkowników systemu operacyjnego.

Przykładowo, aby nadać grupie użytkowników systemu operacyjnego „db2admns” uprawnienie SYSADM, możemy użyć polecenia:

```
update dbm cfg using SYSADM_GROUP db2admns
```

Każda instancja DB2 ma własne definicje grup uprawnień.

W systemie Windows parametry te domyślnie są puste, co oznacza, że lokalna grupa administratorów Windows będzie posiadała uprawnienie SYSADM. W systemie Linux grupa właściciela instancji jest domyślną grupą SYSADM.

### 10.3 Uprawnienie DBADM

DBADM (DataBase Administrator) jest to uprawnienie superużytkownika dla bazy danych. Nie jest to uprawnienie na poziomie instancji, dlatego nie zostało umieszczone w poprzedniej sekcji. Aby nadać uprawnienie DBADM, używamy instrukcji GRANT, tak jak pokazuje poniższy przykład:

```
connect to sample
grant DBADM on database to user <userid>
```

W powyższym przykładzie w pierwszej kolejności musimy podłączyć się do bazy danych - w tym przypadku bazy danych “sample” - a następnie możemy nadać użytkownikowi uprawnienie DBADM. Aby nadać uprawnienie DBADM musimy być SYSADM.

Należy pamiętać, że DBADM nie może tworzyć przestrzeni tablicowych, nawet jeśli są to obiekty wewnątrz bazy danych, ponieważ przestrzenie tablicowe są powiązane z kontenerami (dyskiem) i pulą buforów (pamięcią), które są zasobami fizycznymi systemu.

## 10.4 Grupa PUBLIC

DB2 definiuje wewnętrzną grupę nazwaną PUBLIC. Każdy użytkownik rozpoznany przez system operacyjny albo sieciową usługę uwierzytelniającą zostaje niejawnie członkiem grupy PUBLIC. Podczas gdy tworzona jest baza danych, pewne przywileje nadawane są automatycznie dla grupy PUBLIC:

- CONNECT,
- CREATETAB,
- IMPLICIT\_SCHEMA,
- BINDADD

Dla podwyższonego bezpieczeństwa zalecane jest odebranie wszystkich przywilejów grupie PUBLIC, jak pokazano poniżej:

```
REVOKE CONNECT           ON DATABASE FROM PUBLIC
REVOKE CREATETAB        ON DATABASE FROM PUBLIC
REVOKE IMPLICIT_SCHEMA  ON DATABASE FROM PUBLIC
REVOKE BINDADD          ON DATABASE FROM PUBLIC
```

## 10.5 Instrukcje GRANT i REVOKE

Instrukcje GRANT i REVOKE są częścią standardowego SQL i są używane, aby nadawać albo odbierać przywileje użytkownikowi lub grupie. Poniżej kilka przykładów tych instrukcji:

Aby nadać przywilej SELECT do tabeli T1 użytkownikowi USER1:

```
GRANT SELECT ON TABLE T1 TO USER user1
```

Aby nadać wszystkie przywileje do tabeli T1 dla grupy GROUP1:

```
GRANT ALL ON TABLE T1 TO GROUP group1
```

Aby odebrać wszystkie przywileje do tabeli T1 grupie GROUP1:

```
REVOKE ALL ON TABLE T1 FROM GROUP group1
```

Aby nadać przywilej EXECUTE do procedury p1 użytkownikowi USER1:

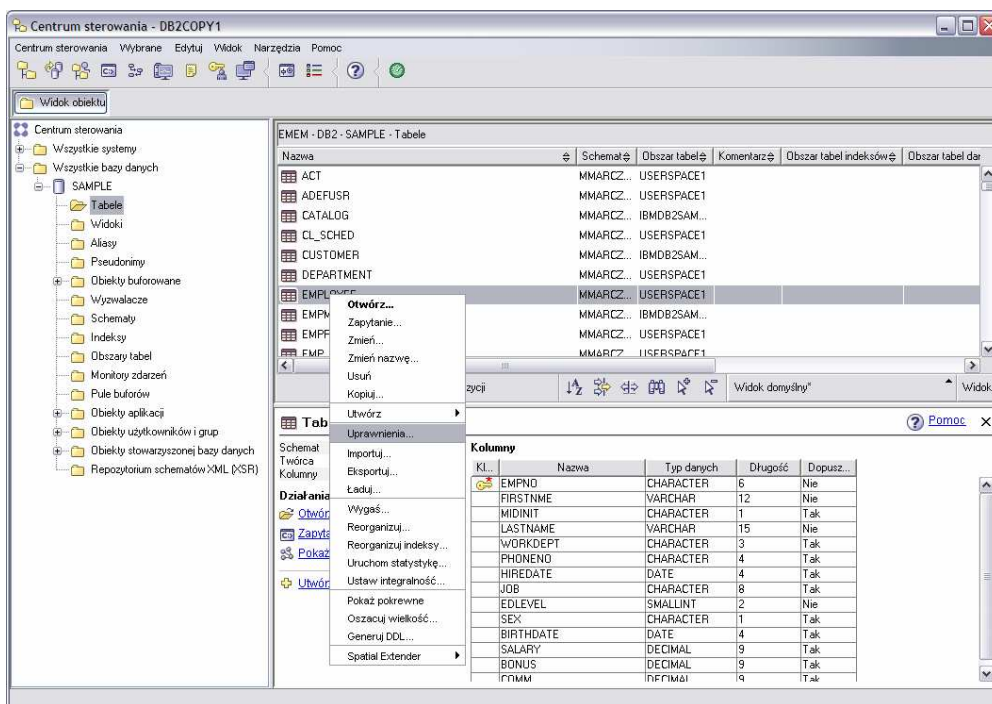
```
GRANT EXECUTE ON PROCEDURE p1 TO USER user1
```

Aby odebrać przywilej EXECUTE do procedury p1 użytkownikowi USER1:

```
REVOKE EXECUTE ON PROCEDURE p1 FROM USER user1
```

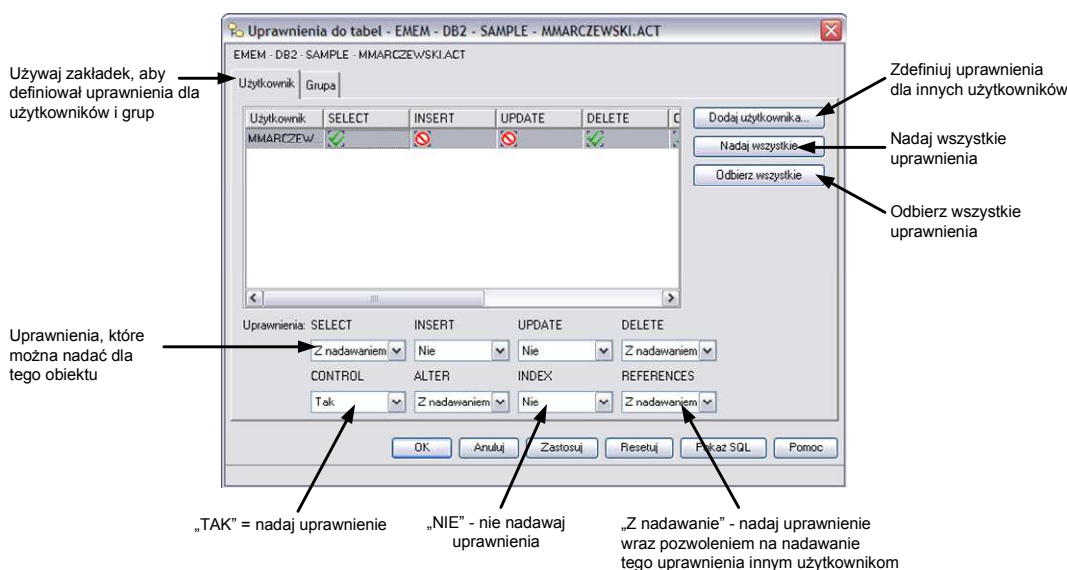
## 10.6 Sprawdzanie uprawnień i przywilejów

Najprostszym sposobem, aby sprawdzić uprawnienia oraz przywileje jest użycie Centrum sterowania. Rysunek 10.4 pokazuje, jak uruchomić okno z uprawnieniami tabeli EMPLOYEE z poziomu Centrum sterowania.



Rysunek 10.4 – Sprawdzanie uprawnień

Jak przedstawiono na Rysunku 10.4, zaznaczamy wybraną tabelę, klikamy na nią prawym przyciskiem i wybieramy z menu opcję Uprawnienia. Po wybraniu, pojawi się okno dialogowe *Uprawnienia do tabeli*, tak jak przedstawiono na Rysunku 10.5. Rysunek ten wyjaśnia również znaczenie poszczególnych pól i elementów okna dialogowego.



**Rysunek 10.5 – Okno dialogowe Uprawnienia do tabel**

Alternatywnie, możemy wykonać zapytanie do tabel słownika systemowego DB2 (SYSCAT), który zawiera informacje o uprawnieniach. Przykładowo, jeżeli chcielibyśmy wiedzieć, czy użytkownik DB2ADMIN posiada przywilej SELECT do tabeli T2 i chcielibyśmy wiedzieć, kto przydzielił mu to uprawnienie, możemy wykonać następujące zapytanie:

```
SELECT grantor, grantee, selectauth
FROM syscat.tabauth
WHERE tablename = 'T2'
```

GRANTOR	GRANTEE	SELECTAUTH
ARFCHONG	DB2ADMIN	Y

W powyższym przykładzie użytkownik ARFCHONG nadał uprawnienie SELECT użytkownikowi DB2ADMIN.

## 10.7 Rozważania o grupowaniu przywilejów

W celu uproszczenia administracji DB2 możemy połączyć użytkowników w grupy i wtedy przydzielić tym grupom wymagane uprawnienia.

Kiedy grupie przyznane jest uprawnienie, członkom tej grupy przyznane jest ono niejawnie, poprzez dziedziczenie wynikające z członkostwa w tej grupie.

W chwili, gdy użytkownik zostaje usunięty z grupy, traci on niejawnie przypisane przywileje grupy, ale wciąż zachowuje każde poprzednie uprawnienie, które zostało mu jawnie przyznane. Przywileje, które zostały jawnie przypisane użytkownikowi, muszą być zabrane mu jawnie.

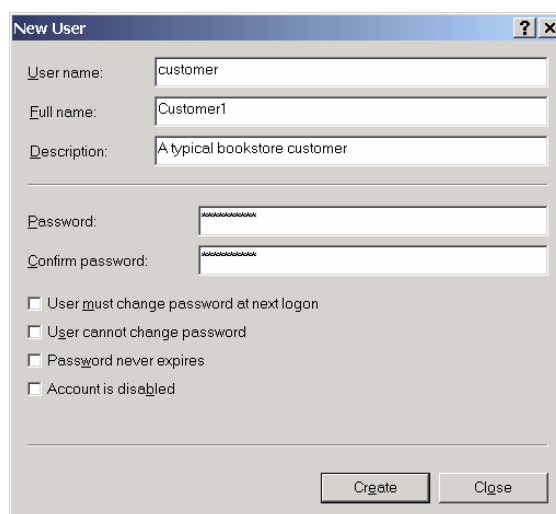
## Laboratorium #9 – Nadawanie i odbieranie uprawnień

### Cel

Do tej pory używaliśmy konta administratora (SYSADM) do wydawania wszystkich bazodanowych poleceń. Konto to ma pełen dostęp to wszystkich narzędzi, danych i obiektów bazy danych. Dlatego bardzo ważne jest, aby zabezpieczyć to konto w celu uniknięcia przypadkowej lub zamierzonej utraty danych. W większości przypadków będziemy chcieli stworzyć inne konta lub grupy, z ograniczonymi uprawnieniami. W tym Laboratorium stworzymy nowe konto użytkownika i nadamy mu wybrane uprawnienia.

### Kroki

1. Otwórz panel *Zarządzanie komputerem*, klikając prawym przyciskiem na ikonę *Mój Komputer* na Pulpicie i wybierając opcję *Zarządzaj*.
2. Rozwiń opcję *Narzędzia systemowe* w drzewie w lewym panelu okna i rozwiń folder *Użytkownicy i grupy lokalne*. Naciśnij prawym przyciskiem na folder *Użytkownicy* i wybierz *Nowy użytkownik*.
3. W oknie dialogowym *Nowy użytkownik* wpisz następujące dane: w polu *Nazwa użytkownika* wpisz „klient” i w polu *Pełna nazwa* wpisz „Klient1”. W polu *Opis* wpisz „Zwyczajny klient księgarni”. W polu *Hasło* i *Potwierdź hasło* wpisz „ibmdb2”. Odznacz pole *Użytkownik musi zmienić hasło przy następnym logowaniu* i naciśnij przycisk *Utwórz*, żeby stworzyć nowego użytkownika.



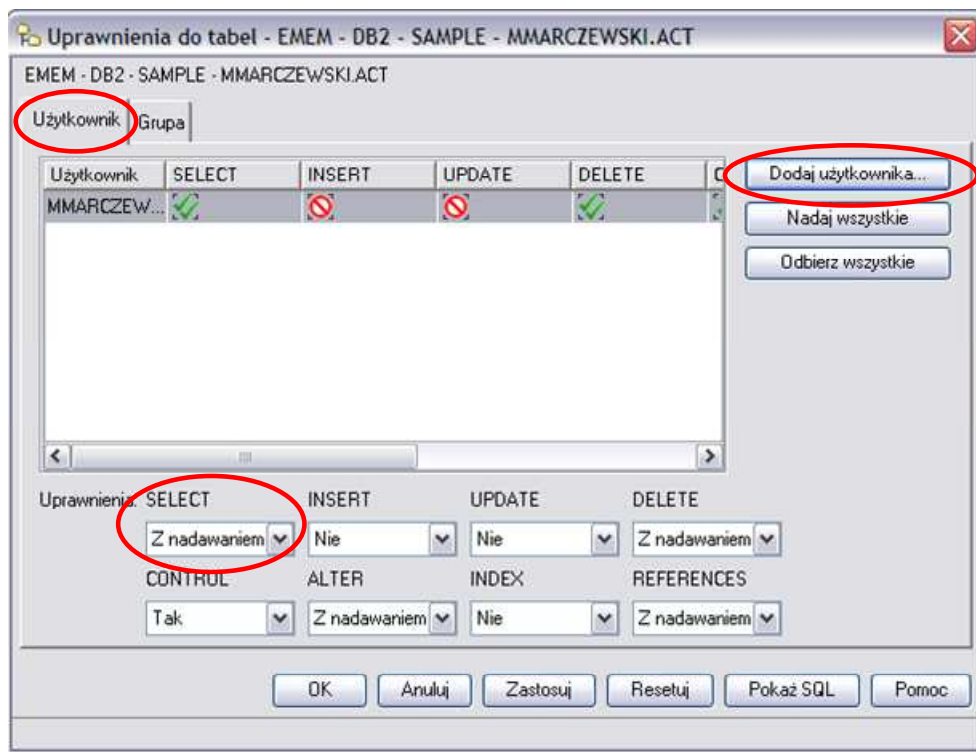
The screenshot shows the 'New User' dialog box with the following fields and options:

- User name: customer
- Full name: Customer1
- Description: A typical bookstore customer
- Password: [masked]
- Confirm password: [masked]
- User must change password at next logon
- User cannot change password
- Password never expires
- Account is disabled

Buttons: Create, Close



4. Sprawdź, czy w Centrum sterowania DB2 używany jest widok zaawansowany. Aby przejść do widoku zaawansowanego, wybierz element menu *Dostosuj Centrum sterowania* z menu *Narzędzia* w Centrum Sterowania. Wybierz opcje *Zaawansowany* i naciśnij przycisk *OK*.
5. Rozwiń drzewo obiektów w Centrum sterowania w lewym panelu obiektów: *Wszystkie bazy danych > EXPRESS > Tabele*.
6. Przyznaj niezbędne przywileje nowostworzonemu użytkownikowi. Z listy tabel bazy danych EXPRESS, naciskając prawy przycisk na tabeli CUSTOMERS, wybierz tabelę, a następnie wybierz opcję *Uprawnienia*, aby wyświetlić okno dialogowe *Uprawnienia do tabel*.
7. Naciśnij przycisk *Dodaj użytkownika* i wybierz użytkownika *klient*, który właśnie został stworzony. Naciśnij przycisk *OK*, żeby zamknąć okno dialogowe *Dodaj użytkownika*.
8. Zostaniesz powiadomiony o tym, że użytkownik *klient* został dodany do listy użytkowników, ale nie ma nadanych żadnych przywilejów. W celu nadania przywilejów SELECT, INSERT, UPDATE i DELETE temu użytkownikowi, zmień każdą rozwijaną listę na *TAK*. Internetowy klient powinien być w stanie przeglądać/dodawać/aktualizować/usuwać swoje dane. Nie przyznajemy użytkownikowi innych uprawnień, ponieważ ich nie potrzebuje. Naciśnij przycisk *OK*, aby zamknąć okno dialogowe *Uprawnienia do tabeli* i zaakceptować zamiany, których dokonaliśmy.



9. Powtórz kroki 7-9 dla tabel BOOKS i SALES. Dla tabeli BOOKS nadaj tylko przywilej SELECT, ponieważ klient nie powinien mieć możliwości do modyfikowania żadnych danych magazynu sklepu. Dla tabeli SALES nadaj tylko przywilej SELECT i INSERT. Klient NIE powinien posiadać przywileju DELETE lub UPDATE, ponieważ tylko pracownicy sklepu powinni mieć możliwość modyfikowania informacji o transakcjach sprzedaży.
10. Podłącz się do bazy danych używając identyfikatora użytkownika *klient* stworzonego powyżej. Spróbuj wykonać zapytanie SELECT do tabeli CUSTOMERS. Co się stało? Spróbuj wykonać polecenie DELETE albo UPDATE do tabeli SALES. Co się stało?

*W tym Laboratorium stworzyliśmy tylko jednego użytkownika, chociaż w Twojej aplikacji możesz potrzebować wiele różnych typów użytkowników. Poeksperymentuj z tworzeniem innych użytkowników i przyznawaniem im uprawnień. Możesz również stworzyć grupy użytkowników i nadać tym grupom uprawnienia, zamiast przydzielać je każdemu użytkownikowi osobno.*

---

# 11

## Rozdział 11 – Kopie zapasowe i odzyskiwanie danych

W rozdziale tym omówimy kwestie dotyczące logowania; pokażemy, jak wykonać pełną i częściową kopię zapasową bazy danych z użyciem narzędzia BACKUP oraz jak odzyskać dane wykorzystując narzędzie RESTORE.

**Ważne:**

Aby dowiedzieć się więcej o logowaniu, kopiach zapasowych, odzyskiwaniu danych obejrzyj wideo:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4282>

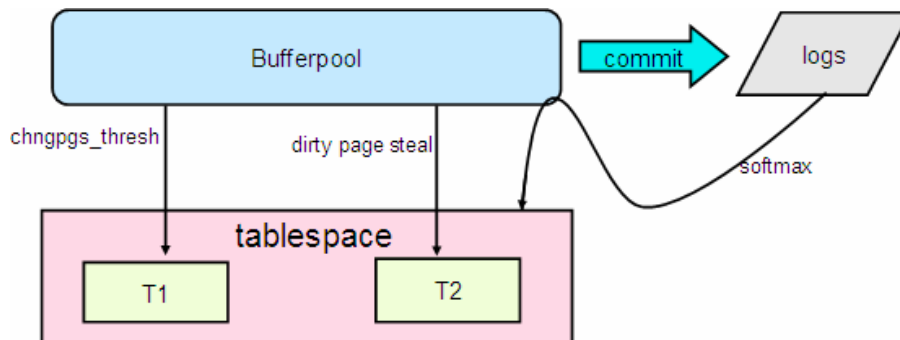
### 11.1 Logowanie w bazie danych

Pracując z edytorem tekstu, za każdym razem, gdy chcemy mieć pewność, że nasz dokument będzie zapisany, klikamy "Zapisz". W świecie baz danych odpowiada temu operacja COMMIT. Za każdym razem kiedy wykonujemy polecenie COMMIT, mamy gwarancję, że jeżeli jakiegokolwiek zmiany zostały dokonane, to zostaną one zachowane.

Analogicznie jeżeli pracujemy z edytorem tekstu, to czasami zauważamy w prawym dolnym rogu ekranu krótki komunikat: „automatyczne zapisywanie”. W świecie baz danych dzieje się podobnie, ponieważ każda operacja, jaką wykonamy na danych, jak UPDATE, INSERT i DELETE, będzie zapisana.

Miejsce, w którym zachowywane są te zmiany to dzienniki bazy danych. Dzienniki bazy danych są przechowywane na dysku i są wykorzystywane do odtwarzania lub wycofywania transakcji podczas procesu odzyskiwania danych (ang. recovery).

Rysunek 11.1 przedstawia graficzną interpretację tego, co dzieje się z dziennikami, kiedy pracujemy z bazą danych.



**Rysunek 11.1 Logowanie w bazie danych DB2**

Na Rysunku 11.1 widzimy przestrzeń tablicową oraz dzienniki. Oba obiekty znajdują się na dyskach twardej, ale nie jest zalecane trzymanie ich na tym samym dysku. Kiedy wykonywana jest operacja UPDATE, strony dla danych wierszy w zapytaniu zostaną załadowane do bufora, a stare i nowe wartości będą przechowywane w dziennikach. Czasami nastąpi to od razu, a czasami dopiero wtedy, kiedy miejsce w buforze się wyczerpie. Jeżeli zostanie wykonany COMMIT po operacji UPDATE, stare wartości zostaną natychmiast zachowane w dziennikach. Proces ten występuje w wielu różnych operacjach SQL wykonywanych na bazie danych. Jedynie w przypadku, gdy spełnione są pewne warunki, takie jak zmiany spełniające próg określony w parametrze CHNGPGS\_THRES, strony w buforze będą zwrócone lub zapisane do przestrzeni tabel na dysk. Parametr CHNGPGS\_THRES wskazuje procent bufora z „wadliwymi” stronami, czyli stronami zawierającymi zmiany.

Pod względem wydajności nie ma sensu, aby dokonywać dwóch zapisów dla każdej operacji COMMIT: jeden zapis do dzienników, a drugi do przestrzeni tabel na dysku. Dlatego zwracanie danych do przestrzeni tablicowych na dysku występuje tylko wtedy, gdy osiągnięte zostaną parametry, takie jak np. próg określony przez "chngpgs\_thres".

## 11.2 Typy dzienników

Możemy wyróżnić dwa typy dzienników:

### Dzienniki podstawowe (ang. primary logs)

Dzienniki te są pre-alokowane, a liczba dostępnych dzienników podstawowych jest określana przez parametr db cfg: LOGPRIMARY.

### Dzienniki dodatkowe (ang. secondary logs)

Są one dynamicznie przydzielane przez DB2, gdy jest to wymagane. Maksymalna liczba dodatkowych dzienników jest ustawiana poprzez parametr db cfg LOGSECOND. Dynamiczne przydzielanie dziennika jest kosztowne, należy się upewnić, że dla codziennych czynności używane są tylko dzienniki podstawowe. Dzienniki dodatkowe są usuwane, kiedy wszystkie połączenia z bazą danych się zakończą.

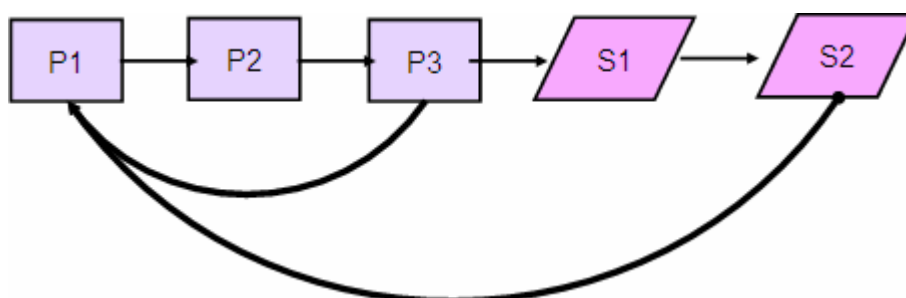
Możliwe jest ustawienie nieskończonego zapisywania do dziennika, jeżeli parametr LOGSECOND będzie ustawiony na -1. Nie jest to jednak zalecane, ponieważ może dojść do zajęcia całej pamięci systemowej.

### 11.3 Tryby zapisów do dziennika

Rozróżniamy dwa tryby zapisów do dziennika: zapis cykliczny (domyślny) oraz zapis archiwalny.

#### 11.3.1 Zapis cykliczny

Rysunek 11.2 pokazuje, jak działa zapis cykliczny.



Rysunek 11.2 – Praca z podstawowymi i dodatkowymi dziennikami

Na Rysunku 11.2 widzimy 3 podstawowe dzienniki. Możemy więc sądzić, że parametr LOGPRIMARY jest ustawiony na 3. Dla uproszczenia przyjmijmy, że w tym przykładzie wykonywana jest tylko jedna transakcja. Podczas, gdy transakcja ta jest wykonywana, zmiany zapisywane są do dziennika P1, a później do P2. Jeśli pojawi się komenda zatwierdzenia (COMMIT), a informacje zostaną zwrócone do przestrzeni tabel na dysku, to pliki dzienników P1 i P2 mogą zostać nadpisane, ponieważ informacje te nie są już potrzebne przy odzyskiwaniu (zostanie to omówione bardziej szczegółowo w dalszej części tego rozdziału). Z drugiej strony, jeżeli transakcja trwa tak długo, że używa P1, P2, P3 i potrzebuje kolejnych dzienników, ponieważ nie była ani zatwierdzona ani cofnięta, wtedy zostanie dynamicznie przydzielony dodatkowy plik dziennika (S1 na rysunku). Jeżeli transakcja wciąż się nie zakończyła, to zostaną przydzielone kolejne dodatkowe dzienniki, aż do osiągnięcia wartości określonej przez parametr LOGSECOND. Jeżeli wciąż będą potrzebne dzienniki, użytkownikowi zostanie zgłoszony błąd zappełnienia dzienników, a transakcja zostanie cofnięta.

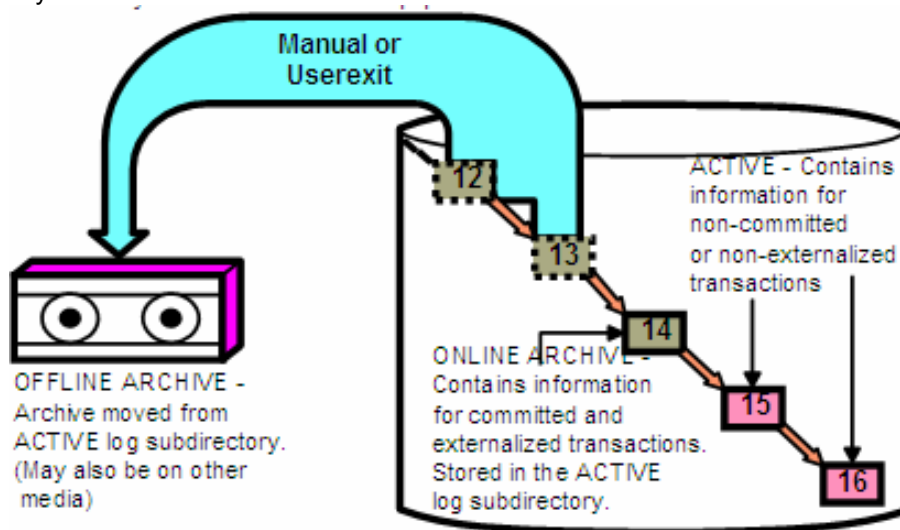
#### 11.3.2 Zapis archiwalny

W zapisie archiwalnym (zapisie zachowującym) dzienniki nie są nadpisywane, ale są przechowywane w trybie on-line lub off-line. Pliki dzienników on-line przechowywane są wraz z aktywnymi logami potrzebnymi do odtworzenia danych na wypadek awarii. Zapis w trybie off-line charakteryzuje się tym, że dane zapisywane są na innych nośnikach, takich jak taśmy, a można tego dokonać za pomocą polecenia USEREXIT. Aby włączyć zapis archiwalny należy ustawić parametr db cfg LOGRETAIN na YES.

Zapis archiwalny jest używany w systemach produkcyjnych i dlatego dzienniki są zachowywane. Pozwala to w większości przypadków na odzyskanie bazy danych aż do najstar-

szego dziennika. Przy zapisie archiwalnym administrator bazy danych może odzyskać dane (do pewnego stopnia) po błędach wprowadzonych przez użytkownika. Na przykład, jeżeli użytkownik uruchomił przypadkowo złą transakcję, a błąd ten zostanie wykryty dopiero po kilku dniach, administrator może odtworzyć stan, w jakim znajdował się system przed wystąpieniem tego błędu. Jednakże niekiedy może być potrzebna ręczna interwencja, aby poprawnie przeprowadzić transakcję.

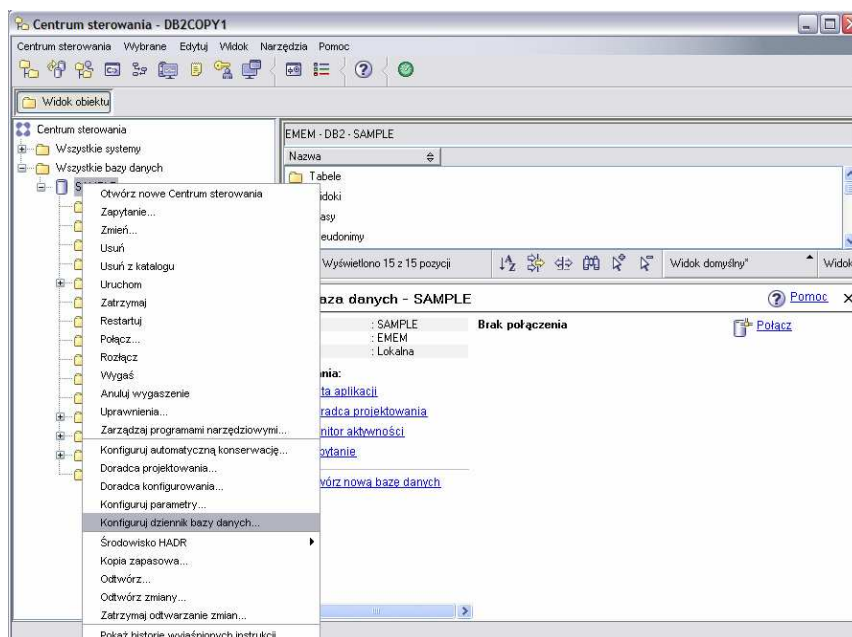
Zapis archiwalny jest wymagany w trakcie odtwarzania bazy danych w trybie „roll forward” oraz do wykonywania kopii zapasowych w trybie on-line. Rysunek 11.3 obrazuje zapis archiwalny.



Rysunek 11.3 – Zapis archiwalny

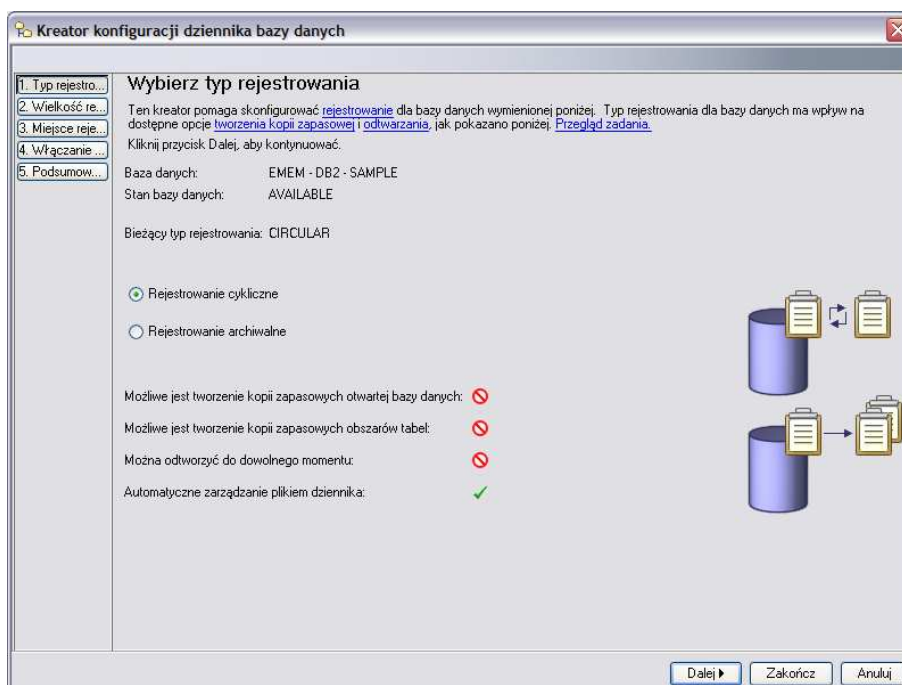
#### 11.4 Logowanie w bazie danych z poziomu Centrum sterowania

Możemy skonfigurować zapis dzienników bazy danych z poziomu Centrum sterowania, klikając prawym przyciskiem myszy na bazie danych i wybierając „Konfiguruj dziennik bazy danych”. Jest to pokazane na Rysunku 11.4.



Rysunek 11.4 – Konfiguracja zapisu dzienników bazy danych z poziomu Centrum sterowania

Rysunek 11.5 pokazuje Kreator konfiguracji dziennika bazy danych, gdzie możemy wybrać rejestrowanie danych cykliczne lub archiwalne.



Rysunek 11.5 – Kreator konfiguracji dziennika bazy danych

## 11.5 Parametry zapisu

Zdefiniowanych jest wiele parametrów DB CFG związanych z zapisem dzienników. Tabela 11.1 przedstawia ważniejsze z nich.

Parametr	Opis
logbufsz	Ilość pamięci do użycia przez bufor dzienników przed zapisem danych na dysk.
logfilsz	Rozmiar każdego skonfigurowanego dziennika w liczbie stron (po 4KB każda).
logprimary	Liczba dzienników podstawowych o rozmiarze logfilsz.
logsecond	Liczba dzienników dodatkowych, które są tworzone w razie potrzeby dla procesu odzyskiwania.
logpath/newlogpath	Miejsce przechowywania dzienników zapisanych archiwalnie i miejsce, gdzie będą tworzone kolejne.
mirrorlogpath	Lokalizacja, która pozwala zabezpieczyć dzienniki podstawowe przed awarią lub przypadkowym usunięciem. Możesz zdefiniować, aby druga kopia dzienników znajdowała się w tej lokalizacji.
loghead	Nazwa obecnie wykorzystywanego dziennika.
userexit	Pozwala programowi userexit na kopiowanie dzienników w trybie off-line.
softmax	Ograniczenie kosztów odzyskiwania.
logretain	Włącza tryb archiwalny zapisu.
overflowlogpath	Podobnie jak opcja OVERFLOW LOG PATH w poleceniu ROLLFORWARD. Jednak zamiast ustawiać opcję ROLLFORWARD komendy OVERFLOW LOG PATH za każdym razem, można to zrobić raz.
blk_log_dsk_ful	Zabezpiecza przed wystąpieniem błędów o braku miejsca na dysku w przypadku, kiedy DB2 nie może stworzyć dziennika w lokalizacja dzienników w zapisie archiwalnym. Zamiast tego DB2 będzie próbowało stworzyć nowy dziennik co 5 minut aż do pomyślnego rezultatu. Tylko odblokowane polecenia czytające SQL mogą być kontynuowane.
max_log	Procent maksymalnego wykorzystania miejsca na zapis archiwalny przez transakcję.
num_log_span	Ilość aktywnych dzienników dla jednego aktywnego UOW.

**Tabela 11.1 – Parametry zapisu**

## 11.6 Kopia zapasowa bazy danych

Polecenie DB2 backup pozwala nam na stworzenie kopii zapasowej bazy danych w momencie, kiedy polecenie zostanie wywołane. Najprostsza konstrukcja tego polecenia wygląda następująco:

```
BACKUP DATABASE <dbname> [ TO <path> ]
```

Większość poleceń i narzędzi może być używana w trybie on-line i off-line. W trybie on-line inni użytkownicy mogą być podłączeni do bazy i wykonywać na niej operacje w czasie tworzenia kopii zapasowej. W trybie off-line żaden inny użytkownik nie może być podłą-



czony do bazy danych. Aby włączyć pracę w trybie on-line, dodaj słowo ONLINE w składni polecenia. Domyślnie kopia zapasowa bazy danych jest tworzona w trybie off-line.

Na przykład, jeżeli chcemy stworzyć kopię zapasową bazy *sample* w katalogu C:\BACKUPS możemy wykonać następujące polecenie z poziomu Okna komend DB2:

```
db2 BACKUP DB sample TO C:\BACKUPS
```

Zauważmy, że katalog C:\BACKUPS musi wcześniej istnieć. Upewnijmy się także, że nie ma żadnych innych połączeń do bazy danych, ponieważ w przeciwnym wypadku otrzymamy komunikat o błędzie, mówiący że nie jest możliwe wykonanie kopii zapasowej typu off-line, kiedy otwarte są inne połączenia.

Aby dowiedzieć się, jakie istnieją połączenia do bazy danych wprowadzamy w Oknie komend DB2 albo powłoce Linux następującą komendę:

```
db2 list applications
```

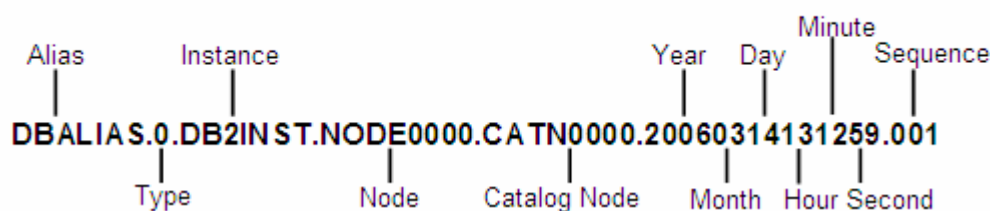
Aby zakończyć wszystkie połączenia do wszystkich baz danych w instancji, wprowadzamy w Oknie komend DB2 albo powłoce Linux komendę:

```
db2 force applications all
```

Być może nie powinniśmy wykonywać ostatniego polecenia w środowisku produkcyjnym, w którym pracuje wielu użytkowników - możemy otrzymać wiele zażaleń od zdenerwowanych współpracowników! Zauważmy również, że ostatnie polecenie jest wykonywane asynchronicznie. Oznacza to, że jeżeli będziemy chcieli wykonać kopię zapasową od razu, to polecenie backup może nie zadziałać. Musimy poczekać kilka sekund i powtórzyć polecenie, jeżeli za pierwszym razem otrzymaliśmy komunikat o błędzie.

Po udanej operacji tworzenia kopii zapasowej, tworzony jest nowy plik zawierający kopię bazy. Jego nazwa jest nadawana według metody pokazanej na Rysunku 11.6.

### Linux/UNIX/Windows



Rysunek 11.6 – Wzór nadawania nazwy pliku kopii bazy danych

Typ „0” oznacza, że kopia jest pełną kopią zapasową bazy danych. Typ „3” oznaczałby, że jest to tylko kopia zapasowa przestrzeni tablicowej. Dla bazy składającej się z jednej partycji nazwa węzła będzie następująca: NODE0000, co dotyczy wszystkich wersji DB2

oprócz DB2 Enterprise Edition z opcją DPF. Nazwa węzła katalogowego jest ustalona na CATN0000. Aby dowiedzieć się więcej zajrzyj do dokumentacji DB2.

Kiedy tworzonych jest wiele kopii zapasowych i są one przechowywane w jednym miejscu, można je rozróżnić po parametrze „timestamp” na końcu nazwy pliku. Jak zobaczysz w następnej sekcji, polecenie RESTORE może wykorzystać ten parametr, aby odtworzyć wskazaną bazę danych.

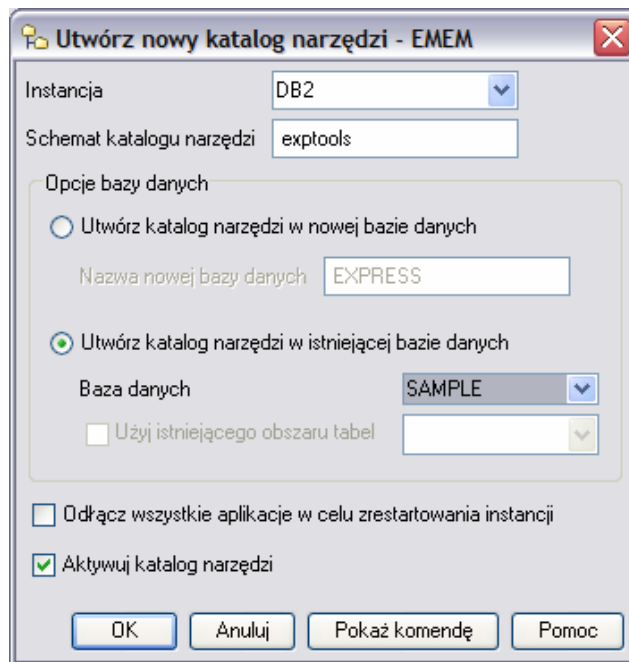
## Laboratorium #10 – Harmonogramowanie kopii zapasowej

### Cel

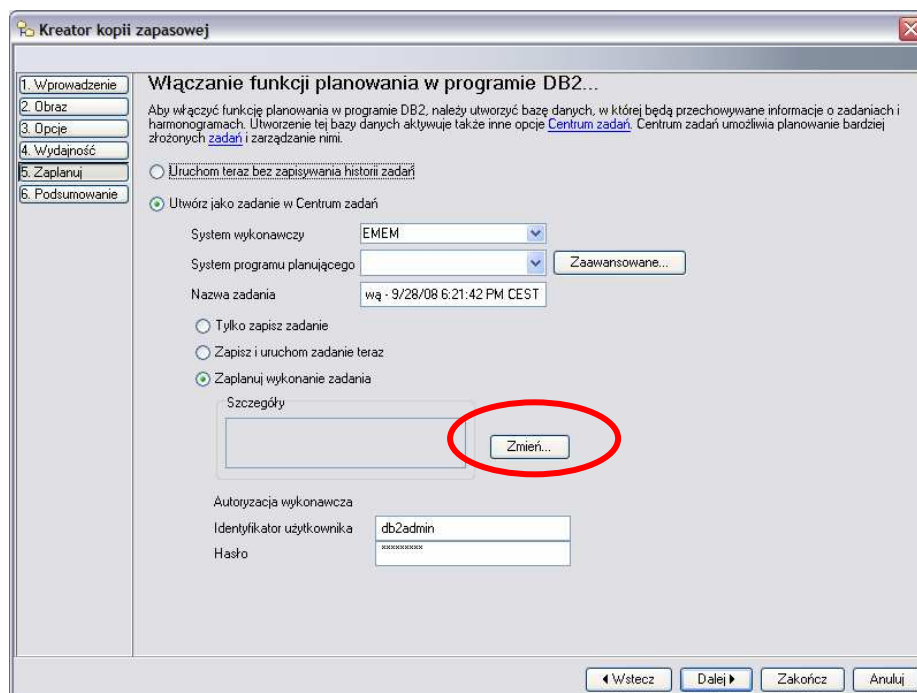
Chociaż DB2 jest w stanie zautomatyzować wiele czynności utrzymania bazy danych, czasami będziesz chciał dostosować je do pewnych działań. W tym Laboratorium utworzysz niestandardowy, nocny harmonogram kopii zapasowej bazy danych EXPRESS.

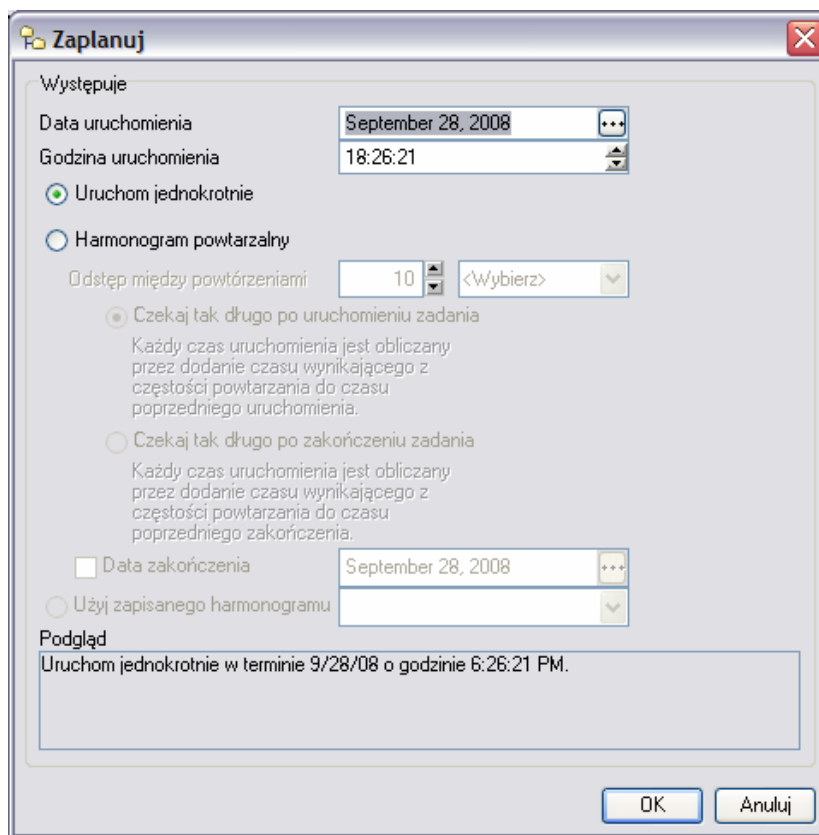
### Kroki

1. Z Centrum sterowania przejdź do Centrum sterowania => Bazy danych. Kliknij prawym przyciskiem myszy na bazę *EXPRESS* i wybierz *Kopia zapasowa*. Uruchomi się *Kreator kopii zapasowej*.
2. Strona wprowadzająca podsumowuje aktualny stan bazy danych, przedstawiając m.in. czas stworzenia ostatniej kopii zapasowej. Naciśnij *Dalej*, aby przejść do następnego kroku.
3. W karcie *Obraz* wybierz lokalizację obrazu bazy danych. Będziesz zazwyczaj wybierał inny dysk twardy, a później ścieżkę do istniejącej już bazy. Na razie utwórz folder `C:\db2backup` i wybierz go jako lokalizację obrazu. W kreatorze wybierz obiekt *System plików* z listy wyboru *Typ Nośnika*. Naciśnij przycisk *Dodaj*, wybierz folder, który przed chwilą stworzyłeś. Naciśnij *Dalej*, aby przejść do następnego kroku.
4. Możesz rozwinąć zakładki *Opcje* i *Wydajność*, ale w większości przypadków domyślne ustawienia są najlepsze, ponieważ DB2 samo dostosowuje ustawienia w najbardziej optymalny sposób. Kiedy skończysz przejdź do zakładki *Zaplanuj*.
5. W zakładce *Zaplanuj*, jeżeli program planujący nie został jeszcze włączony, uaktywnij go. Włącz tworzenie Katalogu narzędzi i pozwól systemowi na stworzenie nowego Katalogu narzędzi. Wybierz schemat dla Katalogu narzędzi i pozwól, aby został stworzony w bazie EXPRESS. Katalog narzędzi przechowuje metadane o wszystkich zadaniach. Kliknij na przycisk *OK*, aby kontynuować. Przejdź do następnej strony klikając przycisk *Dalej*.



6. W zakładce *Zaplanuj* wybierz tworzenie harmonogramu dla wykonywania zadań. Upewnij się, że tworzenie kopii zapasowej będzie tworzone raz dziennie o 1:00. Naciśnij przycisk *Dalej*, aby przejść do następnej strony.





7. W zakładce *Podsumowanie* możesz zobaczyć jakie zaplanowane zadania zostaną wykonane. Kiedy sprawdzisz ustawienia naciśnij *Zakończ*, aby sfinalizować zadania.
8. Uruchom Centrum zadań, aby sprawdzić nowoutworzone zadanie tworzenia kopii zapasowej.

## 11.7 Odzyskiwanie bazy danych

Odzyskiwanie bazy danych zakłada odtworzenie bazy danych z kopii zapasowej i/lub dzienników. Jeżeli odtwarzamy bazę danych tylko z kopii zapasowej, uzyskamy dokładnie taką bazę, jaka istniała w momencie tworzenia kopii.

Jeżeli zostało włączone zapisywanie archiwalne przed wykonaniem kopii zapasowej, możemy odzyskać dane nie tylko z obrazu bazy danych, ale także z dzienników. Jak zobaczymy w kolejnej sekcji, odzyskiwanie typu „roll-forward” pozwala nam na odzyskanie danych z kopii zapasowej, a później na zastosowanie dzienników, aby wprowadzić zmiany aż do końca dziennika lub do wybranego momentu w czasie.

Zauważmy, że pojęcie „odzyskanie” jest często wykorzystywane w tym rozdziale, ale polecenie do jego wykonania to RESTORE.

### 11.7.1 Tryby odzyskiwania danych

Istnieją trzy tryby odzyskiwania danych:

- **Odtwarzanie bazy po awarii (crash lub restart)**

Załóżmy, że pracujemy na komputerze, na którym wykonywane są ważne operacje na bazie DB2. Nagle następuje przerwanie zasilania lub ktoś niechcący odłączy przewód zasilający: co się stanie?

Przy następnym uruchomieniu komputera i DB2, odzyskiwanie bazy w trybie po awarii zostanie automatycznie uruchomione. W odzyskiwaniu tego typu DB2 automatycznie wykona komendę RESTART DATABASE i odczyta przeprowadzone transakcje na podstawie dzienników. Kiedy polecenie się wykona, będziemy mieli gwarancję, że baza będzie w stanie spójnym. Cokolwiek zostało potwierdzone zostanie zapisane, a wszystko, co nie zostało potwierdzone, zostanie wycofane.

- **Odtwarzanie pełnego obrazu bazy**

Ten tryb odzyskiwania oznacza, że odtwarzanie ma miejsce tylko z pełnej kopii zapasowej. Po odzyskaniu, nasza baza będzie w stanie, w jakim znajdowała się podczas tworzenia kopii zapasowej. Jakiegokolwiek operacje, które miały miejsce po utworzeniu kopii zostaną utracone.

- **Odtwarzanie zmian (roll-forward recovery)**

W tym trybie odtwarzamy nie tylko pełen obraz bazy danych za pomocą polecenia RESTORE, ale także wykorzystujemy polecenie ROLLFORWARD, aby uzupełnić odtworzony obraz o operacje zapisane w dziennikach do określonego punktu w czasie. Ten tryb odtwarzania danych minimalizuje poniesione straty danych.

### 11.7.2 Przywrócenie bazy danych

Aby przywrócić bazę danych z jej kopii zapasowej używamy polecenia RESTORE. Najprościej wykonać to polecenie używając poniższej składni:

```
RESTORE DATABASE <dbname> [from <path>] [taken at <timestamp>]
```

Na przykład, jeżeli mamy kopię zapasową bazy *sample* o takiej nazwie:

```

Alias      Instance      Year  Day  Minute  Sequence
|         |         |    |   |      |
SAMPLE.0.DB2INST.NODE0000.CATN0000.20060314131259.001
          |         |         |         |   |   |   |
          Type     Node     Catalog Node  Month Hour Second

```

przywrócenie bazy danych można wykonać następująco:

```
RESTORE DB sample FROM <path> TAKEN AT 20060314131259
```

## 11.8 Inne operacje z BACKUP I RESTORE

Poniższa lista przedstawia kilka operacji, które także możemy wykonać z poleceniami BACKUP i RESTORE. Aby dowiedzieć się więcej, zajrzyj do pomocy.

- Stworzenie kopii zapasowej bazy danych na instancji 32-bitowej i odtworzenie jej na instancji 64-bitowej;
- Odzyskanie bazy zastępując już istniejącą;
- Przekierowanie odtwarzania w komputerze, gdzie istnieje inna liczba dysków, niż podano przy tworzeniu kopii zapasowej;
- Tworzenie kopii zapasowej lub odtwarzanie jedynie przestrzeni tablicowych zamiast całej bazy;
- Tworzenie kopii zapasowej przyrostowej i delta. Kopia typu delta zawiera zmiany tylko od ostatniej zachowanej zmiany. Kopia przyrostowa zapisuje wszystko od ostatniej pełnej kopii zapasowej;
- Zapisywanie kopii zapasowej na nośniku typu flash (wymagane dodatkowe oprogramowanie);
- Przywracanie usuniętych tabel (jeżeli ta opcja była włączona dla danej tabeli)
- Tworzenie kopii zapasowej na jednym systemie (np. Windows) i odtwarzanie na innym (np. Linux) nie jest możliwe. Można natomiast użyć db2look i db2move, aby zrealizować to zadanie.





---

# 12

## Rozdział 12 – Zadania konserwacyjne

Rozdział ten omawia niektóre z zadań, potrzebne aby baza danych była dobrze utrzymana. Głównym kierunkiem w DB2 jest automatyzacja większości z tych zadań. DB2 Express-C, jak i wszystkie inne aktualne wydania DB2 zawierają możliwości automatyzujące te czynności. Jest to wielka zaleta dla małych i średnich firm, które nie mogą sobie pozwolić na zatrudnienie administratora DB2 do zarządzania serwerem bazodanowym. Z drugiej strony, jeśli administrator jest już zatrudniony, będzie miał więcej czasu na przeprowadzenie bardziej zaawansowanych zadań, mogących zwiększyć przychód firmy.

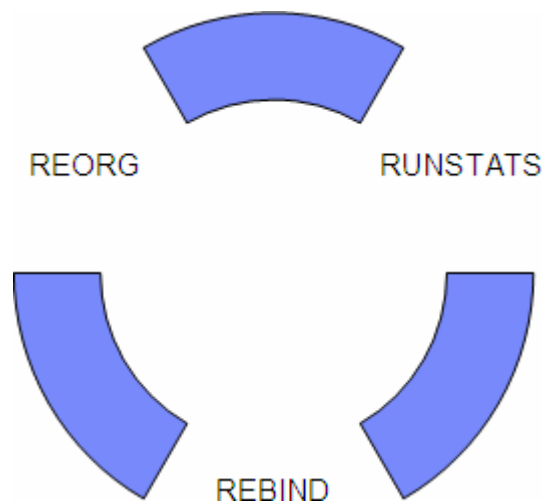
**Ważne:**

Aby uzyskać więcej informacji kliknij na poniższy link:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4302>

### 12.1 REORG, RUNSTATS, REBIND

Tak jak pokazano na Rysunku 12.1, REORG, RUNSTATS oraz REBIND to trzy główne zadania konserwacyjne DB2,



Rysunek 12.1 – Zadania konserwacyjne: REORG, RUNSTATS, REBIND

Rysunek 12.1 pokazuje, że konserwacja bazy danych jest wykonywana w sposób cykliczny. Jeżeli wykonana zostanie komenda REORG, zalecane jest, aby również wywołać RUNSTATS, a potem REBIND.

Wraz z upływem czasu tabele w bazie danych ulegają modyfikacji (w wyniku działania komend UPDATE, INSERT, DELETE). Należy wtedy wykonać cały cykl konserwacyjny rozpoczynając od REORG.

### 12.1.1 Polecenie REORG

Wykonywanie operacji INSERT, UPDATE bądź DELETE sprawia, że wraz z upływem czasu dane są porzucane po różnych stronach bazy danych. Komenda REORG odzyskuje niewykorzystane miejsca w pamięci oraz reorganizuje dane w taki sposób, aby zwracanie wyników było bardziej efektywne. Największy zysk płynie ze stosowania komendy REORG na często modyfikowanych tabelach. REORG można również stosować do reorganizacji indeksów. Reorganizacja może być przeprowadzana zarówno w trybie on-line, jak i off-line.

Stosowanie REORG w trybie off-line jest bardziej efektywne, ale nie pozwala na dostęp do bazy danych. Tryb on-line pozwala na dostęp do bazy, ale wymaga większej ilości zasobów systemowych; sprawdza się najlepiej w przypadku mniejszych tabel.

#### Składnia:

```
REORG TABLE <tablename>
```

#### Przykład:

```
REORG TABLE employee
```

Komenda REORGCHK może być użyta przed komendą REORG w celu sprawdzenia, czy tabela bądź indeks wymaga reorganizacji.

### 12.1.2 Polecenie RUNSTATS

„Mózgiem” DB2 jest Optymalizator DB2. Znajduje on najbardziej efektywne ścieżki dostępu, aby zlokalizować i wydobyć dane. Optymalizator DB2 to zorientowany na koszty system, który poprzez użycie statystyk obiektów bazodanowych, przechowywanych w tabelach słownika systemowego, maksymalizuje wydajność bazy danych. Na przykład, tabele te zawierają statystyki o ilości kolumn i wierszy w tabelach oraz ile i jakiego typu indeksy są dostępne dla określonej tabeli.

Statystyki nie są aktualizowane dynamicznie. Takie ustawienie jest domyślne, ponieważ ciągła aktualizacja statystyk wraz z każdą operacją przeprowadzaną na bazie danych może negatywnie wpłynąć na jej wydajność. Zamiast tego do aktualizacji statystyk w DB2 dostarczona została komenda RUNSTATS. Jest rzeczą bardzo ważną, aby statystyki bazy danych były jak najbardziej aktualne. Optymalizator DB2 może całkowicie zmienić ścieżkę dostępu do danych jeśli uważa, że tabela zawiera 1 rekord, zamiast 1 miliona. Jeśli statystyki bazy danych są aktualne, DB2 wybierze najlepszą drogę dostępu. Częstotliwość

przeprowadzania aktualizacji statystyk jest zależna od tego, jak często modyfikowane są dane w tabeli.

#### Składnia:

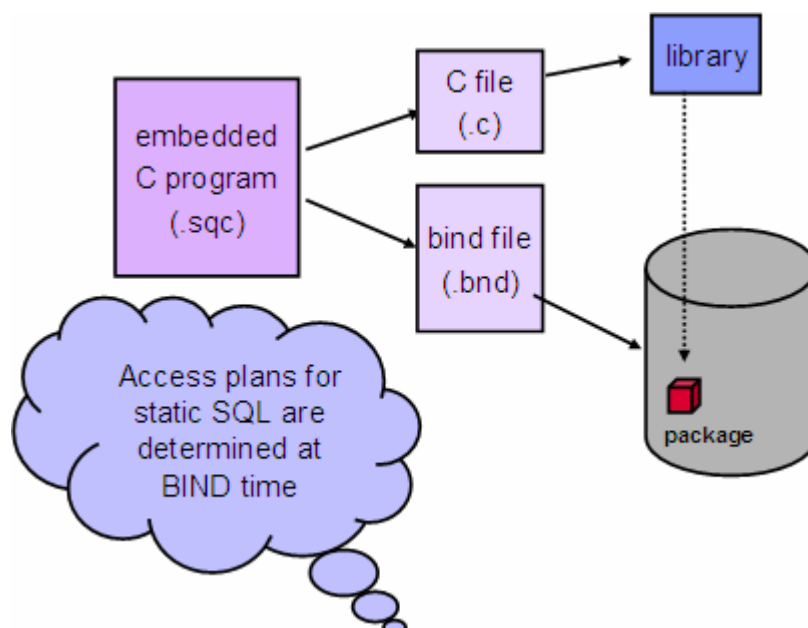
```
RUNSTATS ON TABLE <schema.tablename>
```

#### Przykład:

```
RUNSTATS ON TABLE myschema.employee
```

### 12.1.3 BIND / REBIND

Po pomyślnym wykonaniu komendy RUNSTATS, nie wszystkie zapytania użyją najnowszych statystyk. Plany dostępu statycznego SQL są określane w momencie wykonania komendy BIND, więc nie zawsze statystyki używane w danym momencie są tymi najbardziej aktualnymi. Pokazuje to Rysunek 12.2.



**Rysunek 12.2 – Proces wiązania statycznego SQL**

Na Rysunku 12.2 wbudowany program napisany w C (zapisany jako plik z rozszerzeniem „.sqc”) jest prekompilowany. W wyniku tego procesu zostają stworzone dwa pliki: „.c” zawierający kod w języku C wraz z wycommentowanymi wszystkimi liniami języka SQL oraz plik „.bnd”, który zawiera wszystkie instrukcje SQL. Plik C (ten z rozszerzeniem „.c”) jest kompilowany przez kompilator języka C w zwykły sposób. Tworzona jest biblioteka, tak jak pokazano na Rysunku 12.2. Plik „.bnd” jest w podobny sposób łączony i wygenerowany zostaje pakiet (ang. package), który jest przechowywany w bazie danych. Proces łączenia jest równoważny procesowi kompilacji instrukcji SQL w momencie, w którym dzięki

statystykom dostępnym w danym czasie, określony zostaje najlepszy plan dostępu do danych. Następnie instrukcje są zapisywane w pakiecie.

Co się stanie, gdy dodamy milion wierszy do tabeli użytej przez nasz wbudowany program języka C? Po ich dodaniu, jeśli została wykonana operacja RUNSTATS, statystyki będą zaktualizowane, jednakże pakiet nie będzie automatycznie zaktualizowany, a ścieżki dostępu nie zostaną przeliczone na podstawie zaktualizowanych statystyk. Polecenie db2rbind może być użyte do ponownego wiązania wszystkich istniejących pakietów z najbardziej aktualnymi statystykami.

#### Składnia:

```
db2rbind database_alias -l <logfile>
```

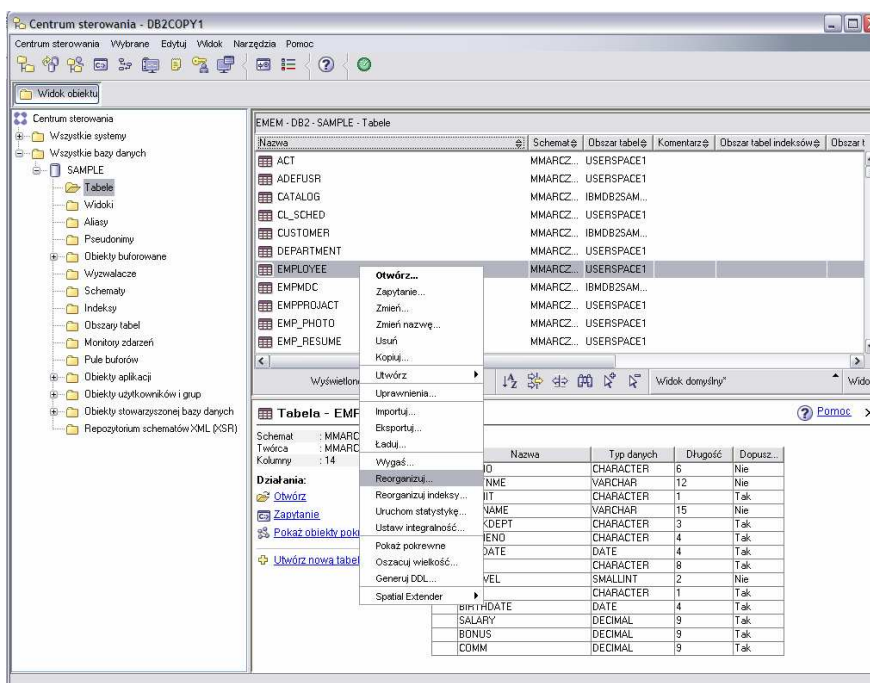
#### Przykład:

Aby ponownie związać paczki bazy danych *sample* i zapisać dziennik do pliku *mylog.txt* należy użyć polecenia:

```
db2rbind sample -l mylog.txt
```

### 12.1.4 Konserwacja bazy z poziomu Centrum sterowania

Z poziomu Centrum sterowania możemy wykonać polecenia REORG lub RUNSTATS. Rysunek 12.3 pokazuje, w jaki sposób można to zrealizować.

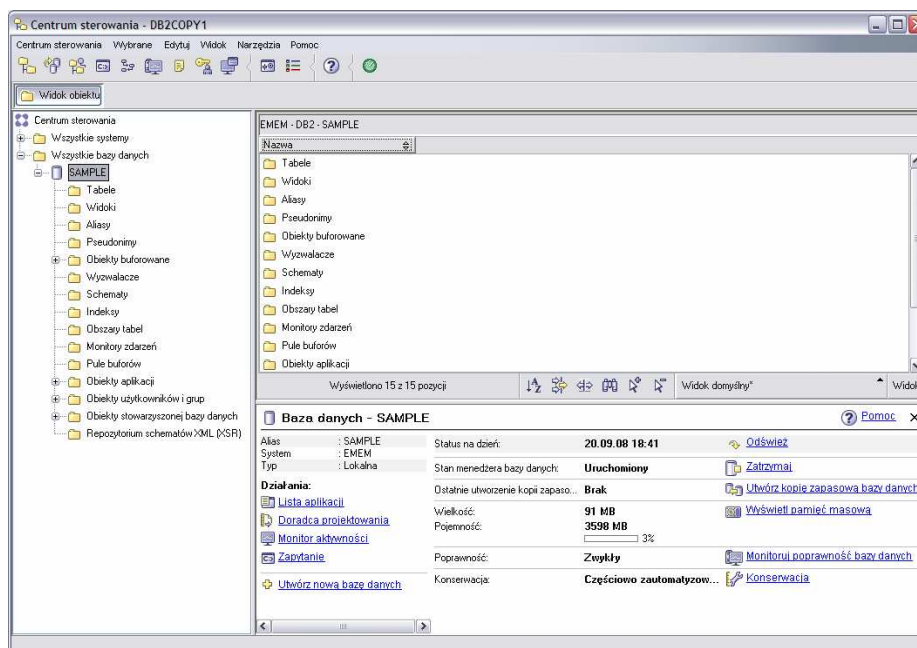


Rysunek 12.3 – REORG i RUNSTATS widok Centrum sterowania

Po wybraniu tabeli, na której chcemy pracować, należy kliknąć prawym przyciskiem myszy, a następnie wybrać opcję Reorganizuj (dla komendy REORG), bądź Uruchom statystykę (dla RUNSTATS).

### Widok operacyjny bazy danych

Po wybraniu bazy danych w prawym dolnym rogu Centrum sterowania pojawi się widok operacyjny, który wyświetli kilka informacji na temat bazy (rozmiar, data ostatniej kopii bezpieczeństwa, czy też ustawienia konserwacji bazy danych). Za pomocą tego widoku można w bardzo łatwy sposób określić czynności, które należy wykonać w zakresie utrzymania bazy. Rysunek 12.4 pokazuje te informacje.



Rysunek 12.4 – Widok operacyjny bazy danych w Centrum sterowania

## 12.2 Sposoby konserwacji

Istnieją trzy sposoby, aby wykonać zadania konserwacyjne:

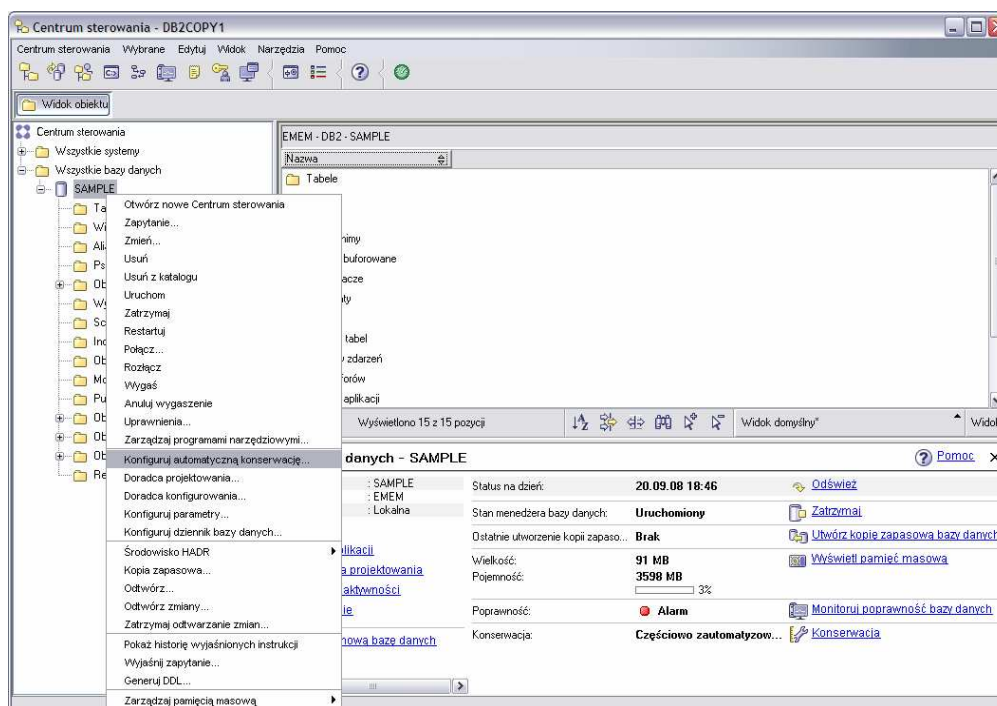
1. Konserwacja ręczna  
Ty przeprowadzasz wszystkie czynności konserwacyjne ręcznie wtedy, gdy istnieje taka potrzeba.
2. Stworzenie skryptów do przeprowadzania konserwacji  
Możesz stworzyć skrypty zawierające komendy konserwacyjne oraz zaplanować ich regularne uruchamianie.
3. Konserwacja automatyczna  
DB2 może automatycznie dbać o konserwację bazy danych za Ciebie (REORG, RUNSTATS, BACKUP).

W tej sekcji skupimy się na konserwacji automatycznej.

Konserwacja automatyczna składa się z następujących elementów:

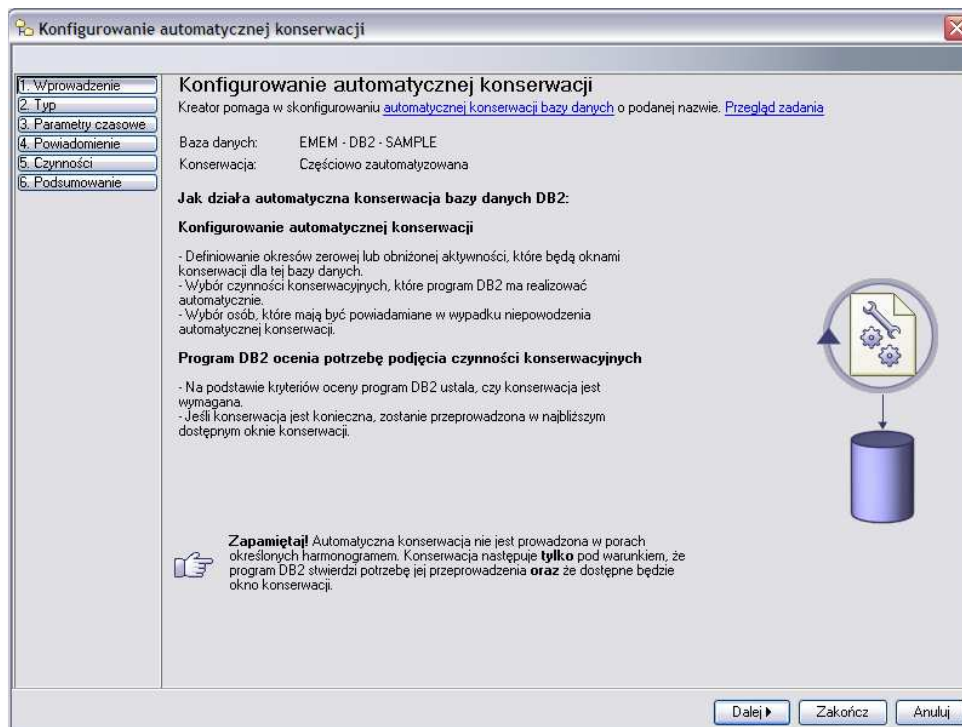
- Użytkownik definiuje *okno konserwacyjne* (ang. maintenance window), czyli okres, w którym wykonywanie zadań nie zakłóci funkcjonowania systemu w dużym stopniu. Na przykład, gdy wykorzystanie systemu w niedzielę między godziną 2:00 a 4:00 jest najmniejsze, to ten okres będzie dobrym *oknem konserwacyjnym*.
- Istnieją dwa okna konserwacyjne: jedno dla operacji w trybie on-line oraz drugie dla trybu off-line.
- DB2 automatycznie przeprowadzi operacje konserwacyjne tylko wtedy, gdy są one potrzebne i tylko w trakcie okna konserwacyjnego.

Z poziomu Centrum sterowania możemy uruchomić kreatora Konfigurowania automatycznej konserwacji, tak jak pokazano to na Rysunku 12.5.



Rysunek 12.5 – Uruchamianie kreatora Konfigurowania automatycznej konserwacji

Rysunek 12.6 pokazuje kreator Konfigurowania automatycznej konserwacji.



Rysunek 12.6 – Konfigurowanie automatycznej konserwacji

---

## Laboratorium #11 – Konfigurowanie automatycznej konserwacji

### Cel

W tym Laboratorium za pomocą kilku prostych kroków skonfigurujesz automatyczną konserwację dla bazy danych SAMPLE.

### Kroki

1. Kliknij prawym przyciskiem myszy na bazę danych *SAMPLE* w drzewie obiektów w Centrum sterowania. Następnie wybierz z menu opcję *Konfigurowanie automatycznej konserwacji*. Uruchomi to kreatora *Konfigurowania automatycznej konserwacji*.
2. Na zakładce *Wprowadzenie* kreator wyświetla aktualne ustawienia automatycznej konserwacji. Jeśli przy tworzeniu bazy danych była włączona opcja automatycznej konserwacji, to jest ona już skonfigurowana. Możesz użyć tego kreatora do ponownej konfiguracji opcji automatycznej konserwacji. Kliknij przycisk *Dalej*, aby przejść do następnego kroku kreatora.
3. Na zakładce *Typ* można wyłączyć automatyczną konserwację lub też zmienić jej ustawienia. Wybierz opcje, aby zmienić aktualne ustawienia. Kliknij przycisk *Dalej*.
4. Na stronie *Parametry czasowe* można ustawić okno konserwacyjne. Ustaw okno konserwacji zamkniętej bazy danych (*off-line*) na każdą sobotę i niedzielę od północy do 6 rano, tak jak pokazano poniżej. Kliknij na przycisk *Zmień* obok podglądu okna konserwacyjnego i ustaw czas. Po wprowadzeniu wymaganych informacji kliknij przycisk *OK*, aby powrócić do kreatora. Okno konserwacji otwartej bazy (*on-line*) pozostaw bez zmian (konserwacja on-line nie zakłóca pracy DB2). Kliknij przycisk *Dalej*.



Zmień specyfikację okna konserwacji - działania na zamkniętej bazie danych

Określ, kiedy może być prowadzona automatyczna konserwacja.

We wskazanym okresie.  
 Poza wskazanym okresem.

Określ godzinę rozpoczęcia i przedział czasu okna konserwacji. Godzinę rozpoczęcia należy podać w formacie 24-godzinnym.

Godzina uruchomienia

Przedział czasu  godziny

Określ, jak często to okno konserwacji ma występować. Poprawne okno konserwacji musi jednocześnie spełniać warunki określone na kartach Dni tygodnia i Dni miesiąca.

Dni tygodnia | Dni miesiąca

Wszystkie  
 Tylko w wybrane dni

poniedziałek       piątek  
 wtorek                 sobota  
 środa                     niedziela  
 czwartek

Podgląd

Automatyczna konserwacja zamkniętej bazy danych może być prowadzona w ramach następującego okna.

Godzina	00:00 - 06:00 (6 godziny)
Dni tygodnia	sobota, niedziela
Dni miesiąca	ALL
Czynności korzystające z tego okna	Brak

OK    Anuluj    Pomoc

- Na stronie *Powiadomienie* można ustawić dane kontaktowe, w razie gdyby automatyczna konserwacja nie powiodła się. Pomiń ten krok. Kliknij na przycisk *Dalej*.
- Na stronie *Czynności* można indywidualnie zautomatyzować albo wyłączyć automatyzację czynności, takich jak powiadomienia o określonych działaniach. W tym przypadku zaznacz wszystkie pola wyboru dotyczące *automatyzacji* i odznacz wszystkie dotyczące *powiadomień*. Kliknij przycisk *Dalej*.
- Zanim przejdziesz do następnej strony kreatora powinieneś określić lokalizację kopii zapasowej bazy danych. Najlepszym rozwiązaniem jest przechowywanie kopii zapasowych na innych dyskach fizycznych, w razie problemów ze sprzętem. Ze strony *Czynności* wybierz opcję *Utwórz kopię*

---

*zapasową bazy danych (BACKUP)*, a następnie kliknij na przycisk *Konfiguruj ustawienia*.

8. Na zakładce *Kryteria wykonywania kopii zapasowej* w oknie dialogowym *Konfiguracja ustawień* wybierz opcję *Kompromis między możliwością odtworzenia a wydajnością bazy danych*. Na zakładce *Położenie kopii zapasowej* zaznacz istniejącą lokalizację kopii zapasowej i kliknij przycisk *Zmień*. Wybierz inną lokalizację, aby wykonać kopię zapasową (upewnij się, że jest wystarczająco dużo miejsca na dysku). Na zakładce *Tryb wykonywania kopii zapasowej* upewnij się, że *Kopia zapasowa zamkniętej bazy danych (off-line)* jest zaznaczona. Kliknij na przycisk *OK*, aby zamknąć zakładkę *Kryteria wykonywania kopii zapasowej*. Kliknij na przycisk *Dalej*
9. Na stronie *Podsumowanie* kreatora można zobaczyć wszystkie wybrane opcje. Kliknij przycisk *Zakończ*, aby zaakceptować wprowadzone zmiany.

---

# 13

## Rozdział 13 – Współbieżność i blokowanie

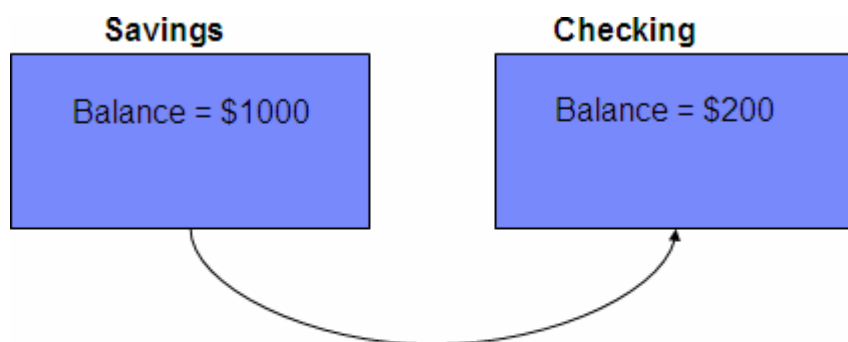
Rozdział ten omawia w jaki sposób wielu użytkowników może korzystać z tej samej bazy danych nie przeszkadzając sobie i zachowując spójność wykonywanych operacji. Przedstawimy zagadnienia transakcji, współbieżności i blokowania.

**Ważne:**

W celu uzyskania dodatkowych informacji o współbieżności i blokowaniu, obejrzyj wideo: <http://www.channeldb2.com/video/video/show?id=807741:Video:4322>

### 13.1 Transakcje

Transakcja, bądź jednostka pracy (ang. unit of work), składa się z jednego lub wielu poleceń języka SQL, które podczas wykonywania powinny być zawsze traktowane jako całość. Oznacza to, że jeżeli jedno z poleceń składających się na transakcję nie powiedzie się, cała transakcja zostanie uznana za nieudaną i wszystkie polecenia, które zostały już wykonane, zostaną wycofane. Transakcja kończy się poleceniem COMMIT, które określa także początek nowej transakcji. Rysunek 13.1 pokazuje przykład transakcji.



Transfer \$100 from Savings to Checking:

- Debit \$100 from Savings account
- Credit \$100 to Checking account

Rysunek 13.1 – Przykład transakcji

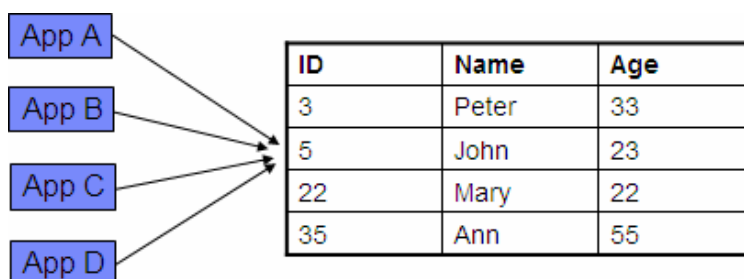
Rysunek 13.1 przedstawia przykładową sytuację, w której próbujemy wykonać przelew z rachunku oszczędnościowego na rachunek bieżący. Aby przelać pieniądze musimy:

Odjąć 100\$ z rachunku oszczędnościowego  
Dodać 100\$ na rachunku bieżącym

Jeżeli powyższe dwie operacje nie zostaną obsłużone jako transakcja, to możemy sobie wyobrazić, co się stanie gdy dojdzie do awarii zasilania po tym, jak uaktualniony zostanie stan rachunku oszczędnościowego, ale jeszcze przed uaktualnieniem rachunku bieżącego. Stracilibyśmy 100\$!

## 13.2 Współbieżność

Współbieżność pozwala, aby kilku użytkowników mogło pracować w jednym momencie na tych samych obiektach bazy danych. DB2 zostało zaprojektowane jako wieloużytkownikowa (ang. multi-user) baza danych. Dostęp do danych musi być zarządzany poprawnie i przejrzysto z wykorzystaniem mechanizmów zapewniających spójność i integralność danych. Przyjrzyjmy się przykładowi na Rysunku 13.2.



**Rysunek 13.2 – Przykład współbieżności i potrzeba wprowadzenia kontroli współbieżności**

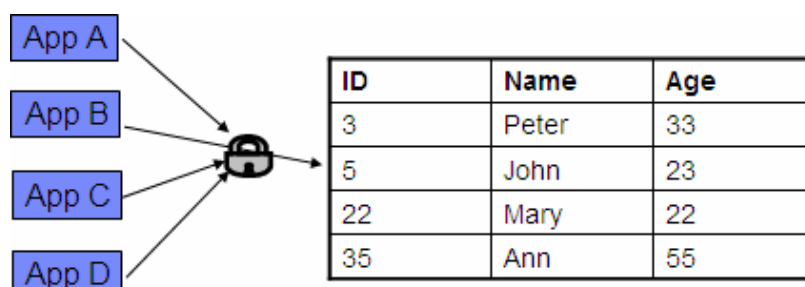
Na Rysunku 13.2 widzimy cztery aplikacje, App A, App B, App C, oraz App D, próbujące uzyskać dostęp do tego samego rekordu tabeli (o numerze 2). Bez kontroli współbieżności wszystkie aplikacje mogłyby wykonać swoje operacje na rekordzie. Zakładając, że każda z nich zapisuje w kolumnie Age inną wartość, aplikacja dokonująca aktualizacji najpóźniej byłaby „zwycięzcą” w tej sytuacji. Z tego przykładu jasno wynika, że pewien zakres kontroli współbieżności jest niezbędny, aby zapewnić spójne wyniki operacji. Taka kontrola opiera się na blokadach.

Blokowanie i współbieżność są ze sobą ściśle związane. Blokowanie zatrzymuje wykonywanie operacji przez aplikację do czasu, aż inna operacja zostanie wykonana w całości. Im więcej blokad jest w systemie, tym mniejsza jest współbieżność. Z drugiej strony, im mniej blokad w systemie, tym większa współbieżność.

Blokady są zakładane automatycznie, gdy potrzebne jest wsparcie dla transakcji, oraz zdejmowane, gdy transakcja dobiega końca (w wyniku polecenia COMMIT bądź ROLLBACK). Blokady mogą być zakładane na rekordy bądź tabele. Występują dwa podstawowe typy blokad:

- Blokady współdzielone (ang. share locks, blokady S) – zakładane, gdy aplikacja chce przeczytać rekord i zapobiec dokonywaniu modyfikacji na tym rekordzie przez inne aplikacje.
- Blokady na wyłączność (ang. exclusive locks, blokady X) – zakładane, gdy aplikacja wprowadza, aktualizuje lub usuwa dane.

Teraz przyjrzyjmy się Rysunkowi 13.3, który pokazuje sytuację z Rysunku 13.2 z zastosowaniem blokady.



**Rysunek 13.3 – Przykład współbieżności i potrzeba używania blokad**

Założmy, że w sytuacji z Rysunku 13.3 App B jako pierwsza uzyskuje dostęp do rekordu 2 i wykonuje polecenie UPDATE - wtedy App B trzyma blokadę X na tym rekordzie. Gdy App A, App C oraz App D chcą dostać się do tego rekordu, nie będą w stanie wykonać polecenia UPDATE z powodu blokady X. Takie zachowanie zapewnia spójność i integralność danych.

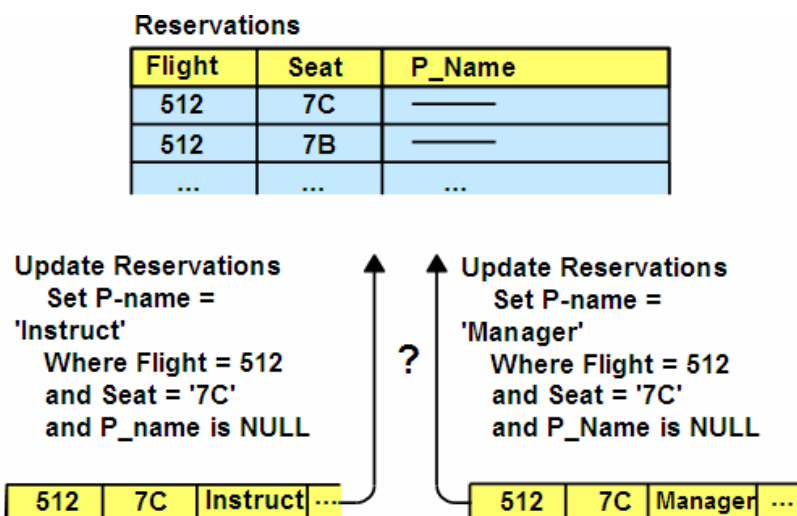
### 13.3 Problemy pojawiające się przy braku kontroli współbieżności

Przy braku kontroli współbieżności mogą pojawiać się następujące problemy:

- ▶ utracona modyfikacja (ang. lost update)
- ▶ odczyt niezatwierdzonych danych (ang. uncommitted read)
- ▶ odczyt niepowtarzalny (ang. non-repeatable read)
- ▶ odczyt fantomów (ang. phantom read)

#### 13.3.1 Utracona modyfikacja

Utracona modyfikacja jest problemem podobnym do przytoczonej wcześniej w tym rozdziale sytuacji, w której aplikacja wykonująca polecenie UPDATE jako ostatnia, zostaje „zwycięzcą”.



Rysunek 13.4 – Utracona modyfikacja

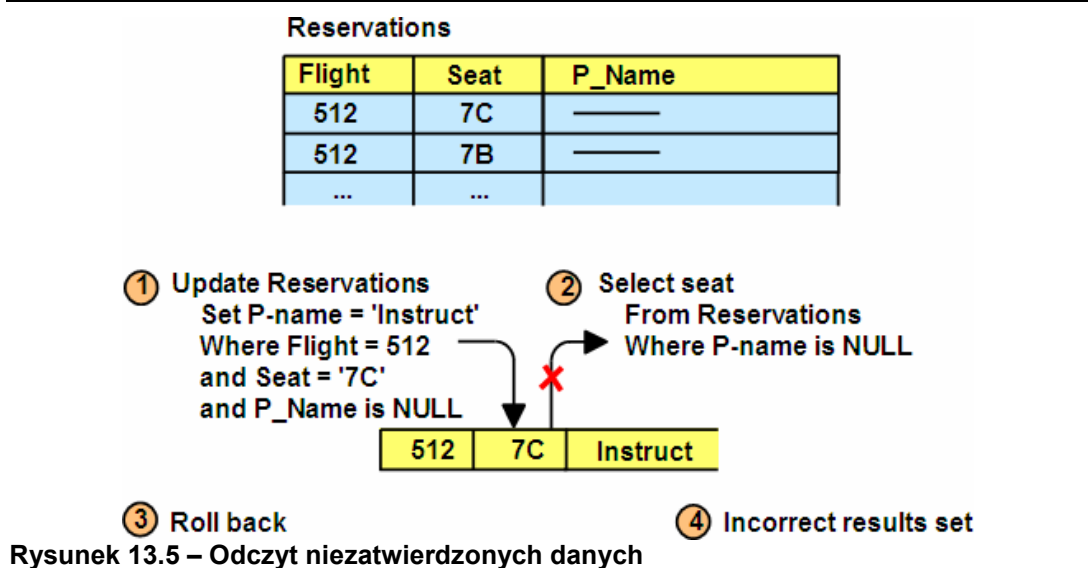
Na Rysunku 13.4 dwie aplikacje próbują aktualizować ten sam rekord. Nazwijmy tę po lewej stronie: App 1 i tę po prawej: App2. Kolejność zdarzeń jest wtedy następująca:

1. App1 aktualizuje rekord
2. App2 aktualizuje ten sam rekord
3. Aktualizacja App1 zostaje zatwierdzona
4. Aktualizacja App2 zostaje zatwierdzona

Aktualizacja dokonana przez App 1 zostaje nadpisana, gdy App 2 dokonuje modyfikacji – stąd nazwa “utracona modyfikacja”.

### 13.3.2 Odczyt niezatwierdzonych danych

Odczyt niezatwierdzonych danych, czy też „odczyt na brudno” (ang. „dirty read”) występuje, gdy aplikacja odczytuje dane, które nie zostały jeszcze zatwierdzone i w związku z tym mogą być nieprawidłowe.



Rysunek 13.5 – Odczyt niezatwierdzonych danych

Rysunek 13.5 pokazuje następującą kolejność zdarzeń:

1. App1 aktualizuje rekord
2. App2 odczytuje dane z tego rekordu
3. App1 wycofuje zmiany wykonane na tym rekordzie

App 2 czyta zatem dane niezatwierdzone i przez to nieprawidłowe – stąd nazwa “odczyt niezatwierdzonych danych”.

### 13.3.3 Odczyt niepowtarzalny

Odczyt niepowtarzalny występuje, gdy wykonując dwa razy to samo zapytanie uzyskuje się różne dane.

FLIGHT	SEAT	NAME	DESTINATION	ORIGIN
512	7B	_____	DENVER	DALLAS
....				
....				
814	8A	_____	SAN JOSE	DENVER
....				
134	1C	_____	HONOLULU	SAN JOSE
....				....

Rysunek 13.6 – Odczyt niepowtarzalny

Rozpatrzmy sytuację z Rysunku 13.6, w której próbujemy dokonać rezerwacji na lot z Dallas do Honolulu. Kolejność zdarzeń jest następująca:

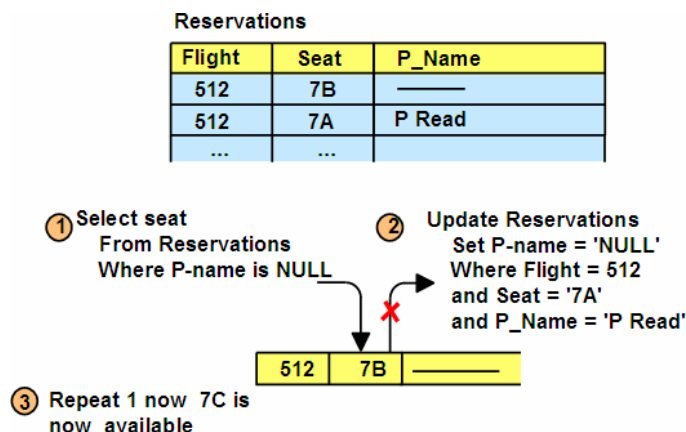
1. App1 otwiera kursor (nazywany także zbiorem wyników) uzyskując dane, jak na Rysunku 13.6
2. App2 kasuje jeden z zaznaczonych rekordów (na przykład rekord z lotem do San Jose)
3. App2 zatwierdza zmiany
4. App1 zamyka i ponownie otwiera znacznik

W takiej sytuacji App 1 otrzyma inny zbiór danych po ponownym wykonaniu tego samego zapytania; dlatego problem ten nazywany jest “odczytem niepowtarzalnym”.

### 13.3.4 Odczyt fantomów

Problem odczytu fantomów jest podobny do odczytu niepowtarzalnego, z tą różnicą, że po ponownym pobraniu danych zbiór zawiera rekordy, których wcześniej nie było (rekordy nie “znikają” jak w odczycie niepowtarzalnym).

Rysunek 13.7 przedstawia przykład takiej sytuacji.



**Rysunek 13.7 – Odczyt fantomów**

Rysunek 13.7 przedstawia następującą kolejność zdarzeń:

1. App1 otwiera kursor
2. App2 dodaje rekord, który znalazłby się w zbiorze wyników
3. App2 zatwierdza zmiany
4. App1 zamyka i ponownie otwiera kursor

W tym przypadku, App1 nie otrzyma tych samych danych po ponownym wykonaniu zapytania. App1 otrzyma teraz zbiór zawierający dodatkowy rekord – “fantom”. Dlatego problem ten nazywamy “odczytem fantomów”.



## 13.4 Poziomy izolacji

Poziomy izolacji są to reguły blokowania, gdzie zależnie od poziomu izolacji, możemy otrzymać różne zachowanie bazy danych przy zakładaniu blokad dla aplikacji.

DB2 dostarcza kilku poziomów izolacji:

- ▶ odczyt niezatwierdzonych danych (ang. Uncommitted Read - UR)
- ▶ stabilność kursora (ang. Cursor Stability - CS)
- ▶ stabilność odczytu (ang. Read Stability - RS)
- ▶ odczyt powtarzalny (ang. Repeatable Read - RR)

### 13.4.1 Odczyt niezatwierdzonych danych

Odczyt niezatwierdzonych danych jest nazywany także “odczytem na brudno”. Jest to najniższy spośród poziomów izolacji i zapewnia najwyższy poziom współbieżności. Podczas operacji odczytu, blokady w ogóle nie są zakładane, o ile inna aplikacja nie próbuje usunąć lub zmienić całej tabeli; operacje aktualizacji zachowują się tak samo, jak na poziomie izolacji “stabilność kursora”.

Problemy, które wciąż mogą pojawiać się na tym poziomie izolacji:

- ▶ odczyt niezatwierdzonych danych
- ▶ odczyt niepowtarzalny
- ▶ odczyt fantomów

Problemy, przed którymi chroni ten poziom izolacji:

- ▶ utracona modyfikacja

### 13.4.2 Stabilność kursora

Stabilność kursora jest domyślnym poziomem izolacji. Zapewnia minimalny stopień blokowania. Na tym poziomie izolacji blokada zakładana jest na bieżący rekord kursora. Jeżeli rekord jest jedynie odczytywany, blokada jest utrzymywana do czasu pobrania następnego rekordu lub do zakończenia transakcji. Gdy rekord jest aktualizowany, blokada jest utrzymywana do czasu zakończenia transakcji.

Problemy, które wciąż mogą pojawiać się na tym poziomie izolacji:

- ▶ odczyt niepowtarzalny
- ▶ odczyt fantomów

Problemy, przed którymi chroni ten poziom izolacji:

- ▶ utracona modyfikacja
- ▶ odczyt niezatwierdzonych danych

### 13.4.3 Stabilność odczytu

Przy poziomie stabilności odczytu blokada zakładana jest na wszystkie wiersze, do których aplikacja uzyskuje dostęp w trakcie transakcji. Dla danego kursora blokada zakładana jest na wszystkie wiersze, które znajdują się w zbiorze wyników. Na przykład, jeśli zapytanie

zwraca 10 wierszy z tabeli zawierającej 10000 wierszy, to blokada zostanie założona tylko na te 10 wierszy. Stabilność odczytu wykorzystuje umiarkowany poziom blokowania.

Problemy, które wciąż mogą pojawiać się na tym poziomie izolacji:

- ▶ odczyt fantomów

Problemy, przed którymi chroni ten poziom izolacji:

- ▶ utracona modyfikacja
- ▶ odczyt niezatwierdzonych danych
- ▶ odczyt niepowtarzalny

#### 13.4.4 Odczyt powtarzalny

Odczyt powtarzalny to najwyższy dostępny poziom izolacji. Zapewnia najwyższy poziom blokowania. Blokady są zakładane na wszystkie rekordy brane pod uwagę w trakcie budowania zbioru wyników. Oznacza to, że blokady mogą zostać założone także na rekordach, które nie znajdują się w ostatecznym zbiorze wyników. Żadna inna aplikacja nie może w tym czasie aktualizować, usuwać lub wstawiać nowych rekordów, które mogłyby wpłynąć na zbiór wyników, do czasu aż bieżąca transakcja się zakończy. Odczyt powtarzalny gwarantuje, że to samo zapytanie wykonane przez aplikację więcej niż raz podczas jednej transakcji, zwróci za każdym razem ten sam zbiór wyników.

Problemy, które wciąż mogą pojawiać się na tym poziomie izolacji:

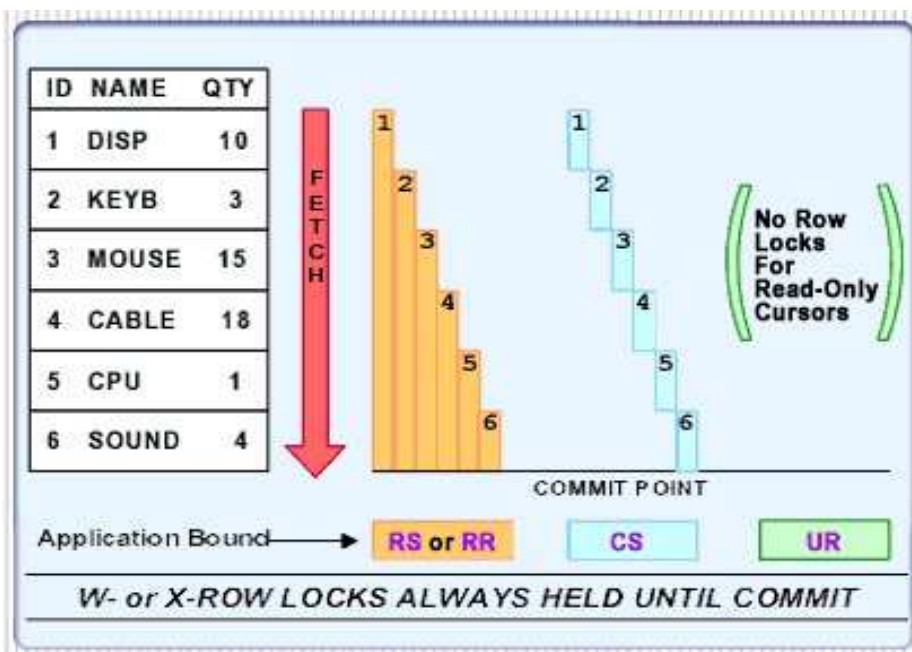
- ▶ brak

Problemy, przed którymi chroni ten poziom izolacji:

- ▶ utracona modyfikacja
- ▶ odczyt niezatwierdzonych danych
- ▶ odczyt niepowtarzalny
- ▶ odczyt fantomów

#### 13.4.5 Porównanie poziomów izolacji

Rysunek 13.8 porównuje poszczególne poziomy izolacji przy pobieraniu danych. Na rysunku widać, że przy poziomie odczytu niezatwierdzonych danych (UR) blokady nie są zakładane. Przy poziomie stabilności kursora (CS) blokada jest zakładana na pierwszy rekord podczas pobierania i zwalniana, gdy tylko rozpocznie się pobieranie drugiego rekordu itd. Przy poziomie stabilności odczytu oraz odczytu powtarzalnego blokada zakładana jest na wszystkie pobierane wiersze i utrzymywana aż do zakończenia transakcji (momentu zatwierdzenia).



Rysunek 13.8 – Porównanie poziomów izolacji przy pobieraniu danych

#### 13.4.6 Ustawianie poziomu izolacji

Poziomy izolacji mogą być określone na wielu poziomach:

- ▶ sesji (aplikacji)
- ▶ połączenia
- ▶ polecenia

Poziom izolacji jest definiowany zazwyczaj na poziomie sesji (aplikacji). Jeżeli w aplikacji nie określono poziomu izolacji to ustawia się on domyślnie na stabilność kursora. Dla przykładu, Tabela 13.1 przedstawia dostępne poziomy izolacji dla aplikacji .NET oraz JDBC oraz ich odwzorowanie na poziomy izolacji w DB2.

DB2	.NET	JDBC
Odczyt niezatwierdzonych danych (UR)	ReadUncommitted	TRANSACTION_READ_UNCOMMITTED
Stabilność kursora (CS)	ReadCommitted	TRANSACTION_READ_COMMITTED
Stabilność odczytu (RS)	RepeatableRead	TRANSACTION_REPEATABLE_READ
Odczyt powtarzalny (RR)	Serializable	TRANSACTION_SERIALIZABLE

Tabela 13.1 – Porównanie poziomów izolacji w różnych technologiach

Poziom izolacji dla polecenia może zostać określony za pomocą klauzuli WITH. Na przykład:

```
SELECT ... WITH {UR | CS | RS | RR}
```

Przykładowy scenariusz:

Aplikacja potrzebuje uzyskać przybliżoną liczbę wierszy w tabeli. Najważniejszy jest czas wykonania zapytania. Poza jedną instrukcją wymagane jest zastosowanie poziomu stabilności kursora:

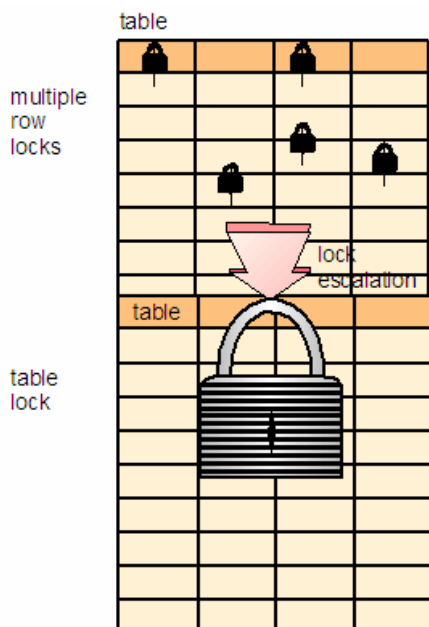
```
SELECT COUNT(*) FROM tabl WITH UR
```

Dla wbudowanego SQL (ang. embedded SQL), poziom izolacji jest ustawiany w momencie wiązania, dla dynamicznego SQL poziom izolacji ustawiany jest w czasie wykonania.

Wybór odpowiedniego poziomu izolacji jest zależny od potrzeb aplikacji. Jeśli aplikacja nie wymaga dokładnych danych, jak w powyższym przykładzie, należy wybrać poziom odczytu niezatwierdzonych danych. Jeżeli aplikacja musi dokładnie kontrolować dane, na których pracuje, należy zdecydować się na poziom odczytu powtarzalnego.

### 13.5 Eskalacja blokad

Każda blokada zakładana przez DB2 zużywa trochę pamięci. Jeżeli Optymalizator DB2 stwierdzi, że lepiej jest założyć jedną blokadę na całą tabelę niż wiele blokad na pojedynczych rekordach, dochodzi do eskalacji blokad. Rysunek 13.9 przedstawia taką sytuację.



Rysunek 13.9 – Eskalacja blokad

Istnieją dwa podstawowe parametry konfiguracji bazy danych związane z eskalacją blokad:

- ▶ **LOCKLIST** – ilość pamięci (liczona w stronach po 4K) zarezerwowana dla blokad dla wszystkich podłączonych aplikacji. Wartość domyślna to pięćdziesiąt stron (200K) na systemie Windows.
- ▶ **MAXLOCKS** – maksymalny procent całej zarezerwowanej pamięci, jaki może być przydzielony pojedynczej aplikacji. Wartość domyślna to 22%.

Zatem, jeśli używane są wartości domyślne, eskalacja blokad zachodzi, gdy pojedyncza aplikacja potrzebuje więcej niż 44K pamięci dla zakładanych przez siebie blokad ( $200K * 22\% = 44K$ ). Jeżeli eskalacja blokad uruchamia się często przy takich ustawieniach, należy zwiększyć wartości LOCKLIST i MAXLOCKS. Eskalacja blokad nie jest dobra dla wydajności, ponieważ zmniejsza współbieżność. Plik z logami diagnostycznymi DB2 (db2diag.log, znajduje się zazwyczaj w katalogu C:\Program Files\IBM\SQLLIB\DB2) może zostać użyty w celu sprawdzenia, czy eskalacja blokad jest uruchamiana.

### 13.6 Śledzenie blokad

Możemy śledzić wykorzystanie blokad za pomocą monitora obrazów stanu blokad aplikacji. Aby włączyć monitor obrazu stanu należy wykonać następujące polecenie:

```
UPDATE MONITOR SWITCHES USING LOCK ON
```

Po włączeniu śledzenia będą zbierane dane na temat blokad. Aby uzyskać raport (obraz stanu) dotyczący blokad w danym czasie, należy wykonać polecenie:

```
GET SNAPSHOT FOR LOCKS FOR APPLICATION AGENTID <handle>
```

Rysunek 13.9 pokazuje wynik przykładowego obrazu stanu blokad aplikacji.

```

Application Lock Snapshot

Snapshot timestamp                = 11-05-2002
00:09:08.672586

Application handle                = 9
Application ID                   =
*LOCAL.DB2.00B9C5050843
Sequence number                  = 0001
Application name                  = db2bp.exe
Authorization ID                  = ADMINISTRATOR
Application status                = UOW Waiting
Status change time               = Not Collected
Application code page             = 1252
Locks held                       = 4
Total wait time (ms)             = 0

List Of Locks
Lock Name                        = 0x050007000480010000000000052
Lock Attributes                  = 0x00000000
Release Flags                    = 0x40000000
Lock Count                       = 255
Hold Count                       = 0
Lock Object Name                 = 98308
Object Type                      = Row
Tablespace Name                  = TEST4K
Table Schema                     = ADMINISTRATOR
Table Name                       = T2
Mode                             = X

```

Rysunek 13.9 – Obraz stanu blokad aplikacji

### 13.7 Oczekiwanie na blokadę

Gdy dwie (lub więcej) aplikacje chcą wykonać operacje na tym samym obiekcie, jedna z nich musi czekać, aby otrzymać blokadę. Domyślnie aplikacja będzie czekać aż do skutku. Czas przez jaki aplikacja oczekuje na uzyskanie blokady jest określony przez parametr konfiguracyjny bazy danych LOCKTIMEOUT. Wartością domyślną jest -1 (brak ograniczenia).

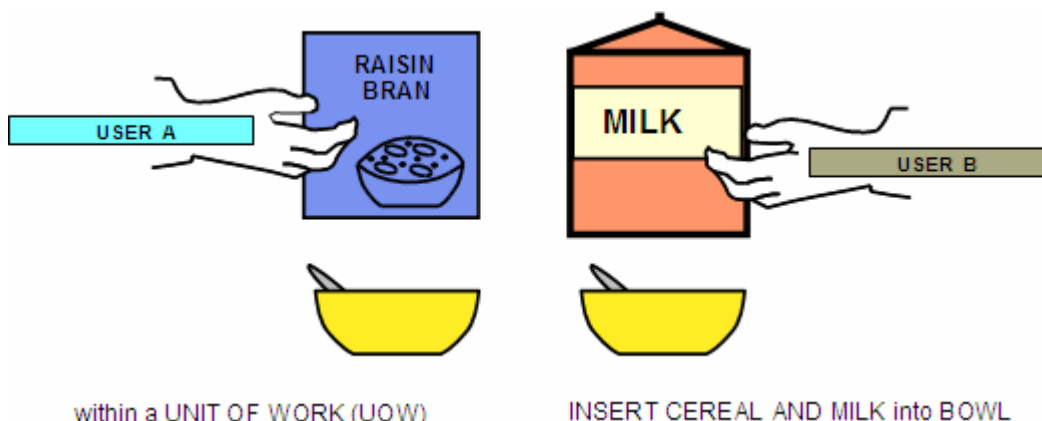
Rejestr CURRENT LOCK TIMEOUT może zostać użyty w celu ustawienia czasu oczekiwania na blokadę dla danego połączenia. Domyślnie, rejestr ma taką samą wartość jak LOCKTIMEOUT. Aby zmienić tę wartość należy wykonać polecenie SET LOCK TIMEOUT. Gdy wartość tego rejestru zostanie ustawiona dla połączenia, będzie stała między transakcjami.

Przykład:

```
SET LOCK TIMEOUT=WAIT n
```

### 13.8 Przyczyny i wykrywanie zakleszczeń

Zakleszczenie występuje, gdy dwie (lub więcej) aplikacje podłączone do tej samej bazy danych oczekują w nieskończoność na dostęp do obiektu. To oczekiwanie nie może zostać przerwane, gdyż każda z aplikacji trzyma blokadę na obiekcie, którego potrzebuje druga. Zakleszczenie jest przeważnie wynikiem złego zaprojektowania aplikacji. Rysunek 13.10 ilustruje zakleszczenie.



**Rysunek 13.10 – Przykład zakleszczenia**

Na Rysunku 13.10, użytkownik A trzyma płatki i nie odda ich, dopóki nie otrzyma mleka. Jednocześnie użytkownik B jest w posiadaniu mleka i nie odda go, dopóki nie otrzyma płatków. W tej sytuacji mamy do czynienia z zakleszczeniem.

Aby zasymulować zakleszczenie, wykonajmy następujące kroki:

1. Otwórzmy dwa Okna komend DB2 (które będziemy nazywali odpowiednio „CLP1” oraz „CLP2”) reprezentujące dwie aplikacje łączące się do bazy danych.
2. W CLP1 wykonujemy następujące polecenia:

```
db2 connect to sample
db2 +c update employee set firstnme = 'Mary' where empno =
'000050'
```

Najpierw łączymy się do bazy danych SAMPLE, a później aktualizujemy rekord z „empno=50000” w tabeli employee. Opcja „+c” w poleceniu oznacza, że nie chcemy, aby Okno komend DB2 automatycznie zatwierdziło transakcję. Ustawiamy ją w celu utrzymania blokad.

3. W CLP2 wykonujemy następujące polecenia:

```
db2 connect to sample
```

```
db2 +c update employee set firstnme = 'Tom' where empno =
'000030'
```

W oknie CLP2, które reprezentuje drugą aplikację, także łączymy się do bazy SAMPLE, ale aktualizujemy inny wiersz w tabeli employee.

4. W CLP1 wykonujemy:

```
db2 +c select firstnme from employee where empno = '000030'
```

Po naciśnięciu klawisza ENTER w celu wykonania powyższego polecenia, może wydawać się, że aplikacja się zawiesiła. W istocie CLP1 oczekuje na zwolnienie blokady na wyłączność założonej przez CLP2 w kroku 3. W takiej sytuacji, jeżeli nie zmieniono wartości LOCKTIMEOUT (domyślnie -1), CLP1 będzie czekać w nieskończoność.

5. W CLP2 wykonujemy:

```
db2 +c select firstnme from employee where empno = '000050'
```

Wykonanie powyższego polecenia spowoduje zakleszczenie. To polecenie SELECT nie wykona się, ponieważ będzie oczekiwało na zwolnienie blokady na wyłączność przydzielonej CLP1 w kroku 2.

W powyższym scenariuszu DB2 sprawdzi parametr konfiguracyjny DLCHKTIME. Parametr ten określa, jak często DB2 szuka zakleszczeń. Na przykład, jeśli parametr ustawiony jest na 10 sekund, DB2 będzie co dziesięć sekund sprawdzało czy wystąpiło zakleszczenie. Jeśli tak się stało, DB2 wykona wewnętrzny algorytm w celu określenia, która z dwóch transakcji powinna zostać wycofana, a która może kontynuować wykonywanie poleceń.

Jeśli często dochodzi do zakleszczeń, należy przyjrzeć się dokładnie wykorzystywanym transakcjom i zastanowić się, czy nie ma możliwości ich przeprojektowania.

### 13.9 Najlepsze praktyki we współbieżności i blokowaniu

Poniżej znajduje się kilka dobrych wskazówek, jak zapewnić możliwie największą współbieżność:

1. Utrzymuj transakcje tak krótko, jak się da. Można to osiągnąć poprzez wywoływanie co pewien czas polecenia COMMIT (nawet dla transakcji, które jedynie odczytują dane), gdy tylko pozwala na to logika aplikacji.
2. Loguj transakcje tylko wtedy, gdy jest to niezbędne.
3. Usuwać (czyścić) dane szybko używając polecenia:

```
ALTER TABLE ACTIVATE NOT LOGGED INITIALLY WITH EMPTY TABLE
```



4. Grupuj wykonywane modyfikacje. Na przykład:

```
DELETE FROM (  
    SELECT * FROM tedwas.t1 WHERE c1 = ... FETCH FIRST 3000 ROWS  
    ONLY)
```

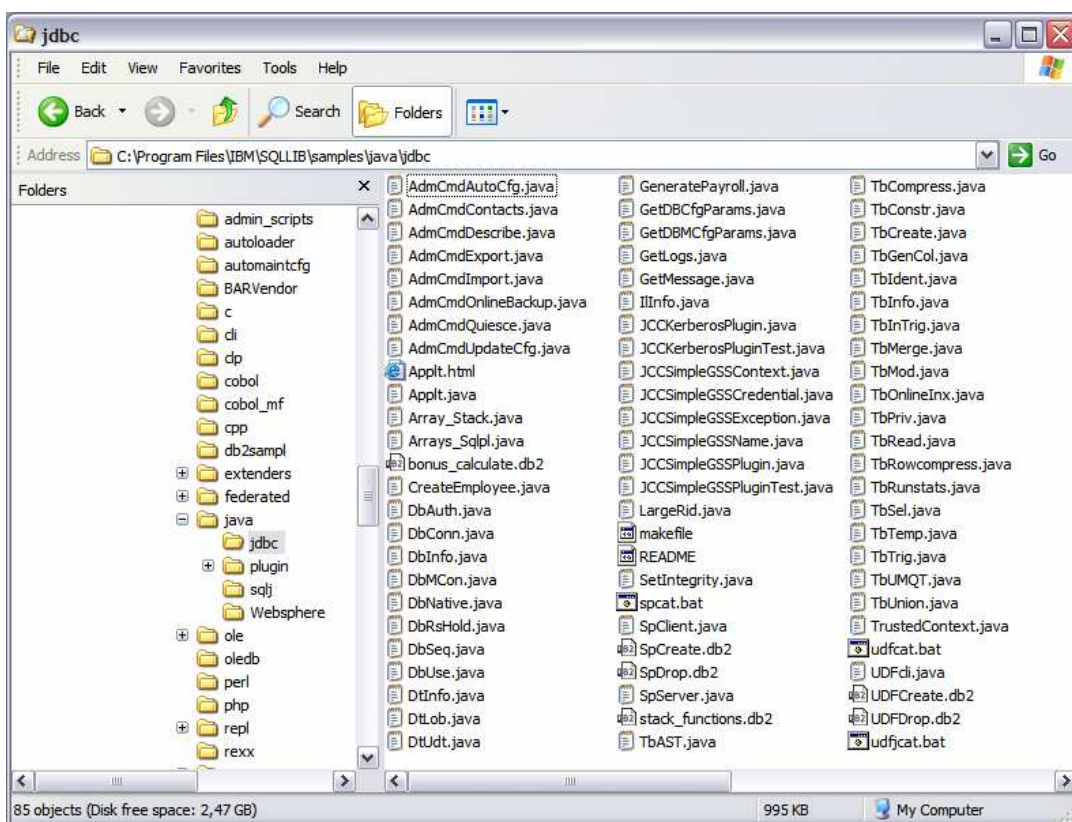
5. Wykorzystuj funkcje wspomagające współbieżność dostarczane w narzędziach DB2 do przenoszenia danych.
6. Ustaw parametr LOCKTIMEOUT (sugerowany czas to między 30, a 120 sekund). Nie zostawiaj domyślnej wartości -1. Możesz też używać ustawień długości oczekiwania dla sesji.
7. Nie pobieraj więcej danych niż potrzeba. Możesz, na przykład używać klauzuli FETCH FIRST n ROWS ONLY w poleceniach SELECT.



---

**CZEŚĆ III – NAUKA DB2:  
PROGRAMOWANIE APLIKACJI**

W części III tej książki szczegółowo omówione zostały takie obiekty bazodanowe, jak procedury składowane, funkcje użytkownika (UDF) czy wyzwalacze. Zwracamy także uwagę, że możesz się uczyć programować aplikacje w różnych językach, wykorzystując DB2 jako serwer baz danych, także poprzez analizowanie przykładów, które są dostarczane wraz z DB2. Znajdują się one w katalogu SQLLIB\samples. Poniższy rysunek przedstawia przykładowe programy Java dostarczane wraz z DB2 dla systemu Windows.



**Przykładowe programy Java dostarczane wraz z DB2**

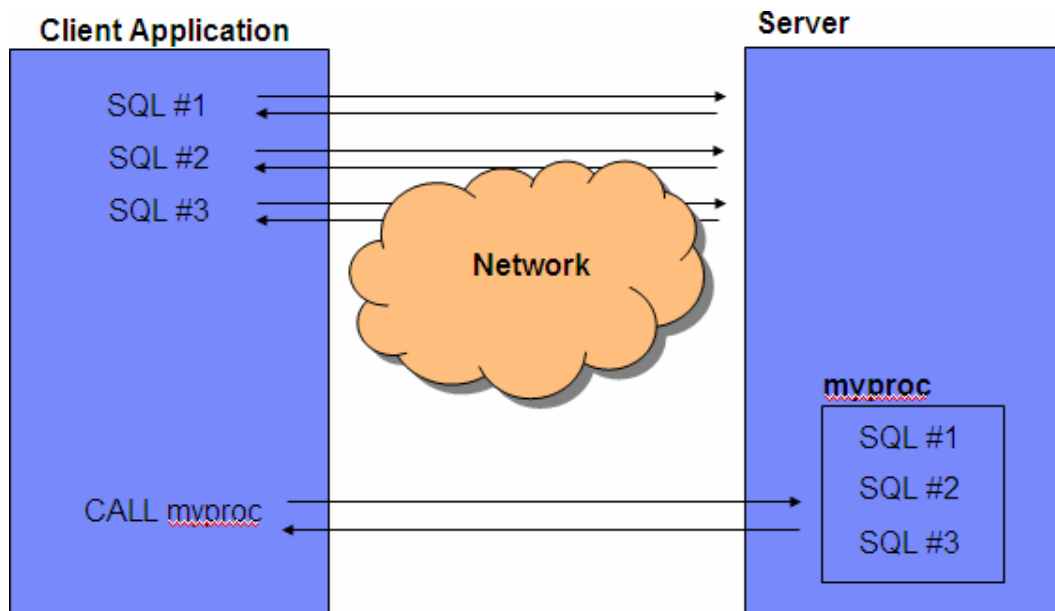
# 14

## Rozdział 14 – Procedury składowane w SQL PL

W tym rozdziale omówimy procedury składowane (ang. stored procedures). Procedura składowana jest obiektem bazy danych, która zawiera instrukcję języka SQL oraz logikę biznesową. Utrzymanie logiki w bazie danych umożliwia polepszenie wydajności, co znacznie redukuje ruch sieciowy pomiędzy aplikacją a bazą danych. Co więcej, procedury składowane umożliwiają scentralizowane składowanie kodu, dzięki czemu inne aplikacje mogą używać tych samych procedur.

Procedury składowane w DB2 mogą być napisane w SQL PL, C/C++, Java, Cobol, CLR (Common Language Runtime) i OLE. W tym rozdziale skupimy się na procedurach składowanych napisanych w SQL PL, ze względu na ich popularność i prostotę.

Rysunek 14.1 ilustruje działanie procedur składowanych.



Rysunek 14.1 – Zmniejszenie ruchu sieciowego dzięki procedurom składowanym

W lewym górnym rogu Rysunku 14.1 znajduje się kilka instrukcji SQL wykonywanych jedna po drugiej. Każda z nich przesyłana jest od klienta do serwera, po czym serwer zwraca

wynik do klienta. Przy uruchomieniu większej ilości instrukcji dochodzi do zwiększenia ruchu w sieci. Z drugiej strony, w prawym dolnym rogu rysunku, znajduje się procedura składowana "myproc" trzymana na serwerze, która zawiera te same instrukcje SQL; klient (po lewej stronie), wywołuje procedurę używając wyrażenia CALL. Druga metoda jest bardziej efektywna, ponieważ poprzez sieć przesyłane jest tylko jedno wywołanie procedury oraz wynik zwracany jest tylko jeden raz.

Procedury składowane mogą być również pomocne w celach związanych z bezpieczeństwem bazy danych. Przykładowo, można zezwolić użytkownikom na dostęp do tabel lub widoków tylko poprzez procedury składowane, dzięki czemu odciążony jest serwer a użytkownicy nie mają dostępu do informacji, do których nie zostali upoważnieni. Jest to możliwe, ponieważ nie są wymagane od użytkowników jawne uprawnienia do tabel czy widoków, do których mają dostęp poprzez procedury składowane. Użytkownicy potrzebują tylko przyznania wystarczających uprawnień do wywoływania procedur składowanych.

**Ważne:**

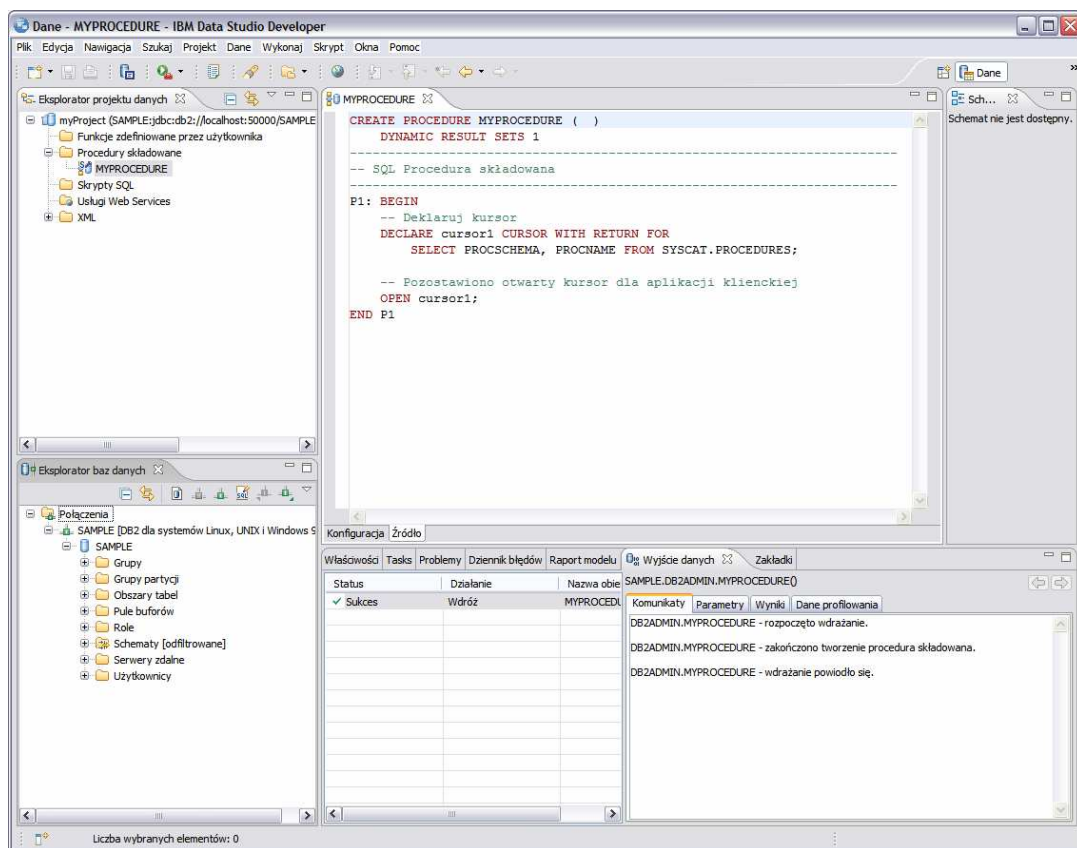
Więcej informacji o procedurach składowanych w SQL PL dowiesz się z wideo na:  
<http://www.channeldb2.com/video/video/show?id=807741:Video:4343>

## 14.1 IBM Data Studio

IBM Data Studio jest kompleksowym rozwiązaniem, które pomaga opracowywać i zarządzać aplikacjami bazodanowymi przez cały cykl życia treści. Oto kilka czynności, które mogą być wykonane w Data Studio:

- tworzenie, zmienianie i usuwanie obiektów DB2 (wraz z analizowaniem wpływów)
- badanie i edytowanie danych - relacyjnych i XML
- wizualne budowanie wyrażen SQL i XQuery
- optymalizowanie zapytań w Visual Explain
- opracowywanie, debugowanie i wdrażanie procedur składowanych w SQL i Java™
- opracowywanie funkcji zdefiniowanych przez użytkownika
- opracowywanie aplikacji SQLJ
- opracowywanie zapytań i funkcji dla aplikacji pureXML
- wykonywanie zadań przemieszczających dane
- współpraca i dzielenie się projektami z członkami zespołu
- szybkie budowanie usług sieciowych Web Services (SOAP i REST)
- wykrywanie związków pomiędzy obiektami baz danych a fizycznymi modelami (diagram)
- wizualizowanie danych rozdzielonych pomiędzy tabele

IBM Data Studio oparte jest na platformie Eclipse. Jest to oddzielny obraz (nie jest on częścią obrazu instalacyjnego DB2), który jest również bezpłatny. Obraz Data Studio można pobrać za pośrednictwem zakładki "Download" na stronie [ibm.com/db2/express](http://ibm.com/db2/express). Rysunek 14.2 przedstawia IBM Data Studio.



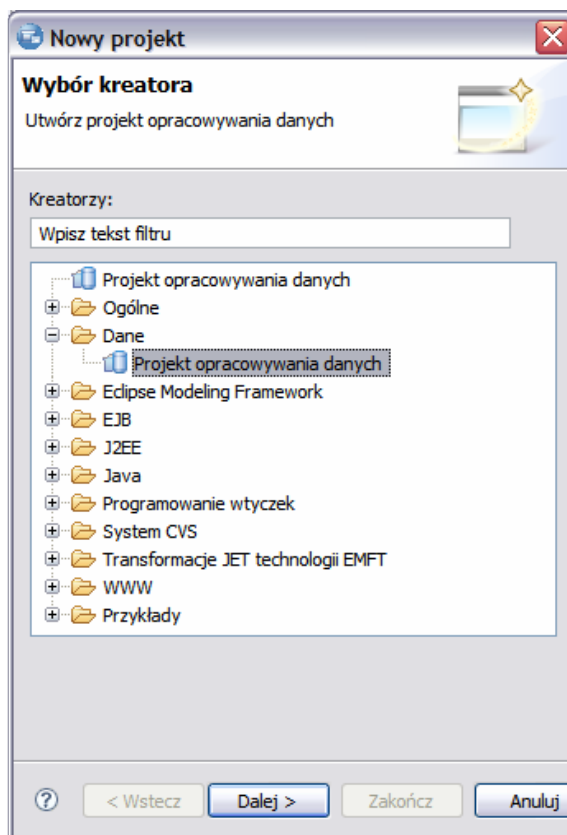
Rysunek 14.2 – IBM Data Studio

### 14.1.2 Tworzenie procedury składowanej w Data Studio

W celu utworzenia procedury składowanej w SQL PL lub w Java wykonujemy poniższe kroki. Zauważmy, że w IBM Data Studio nie można utworzyć procedur składowanych w innych językach.

#### Krok 1: Tworzenie projektu w Data Studio

Z Data Studio menu, wybieramy *Plik* -> *Nowy* -> *Projekt* a następnie *Projekt opracowywania danych*. Tak, jak ilustruje Rysunek 14.3.



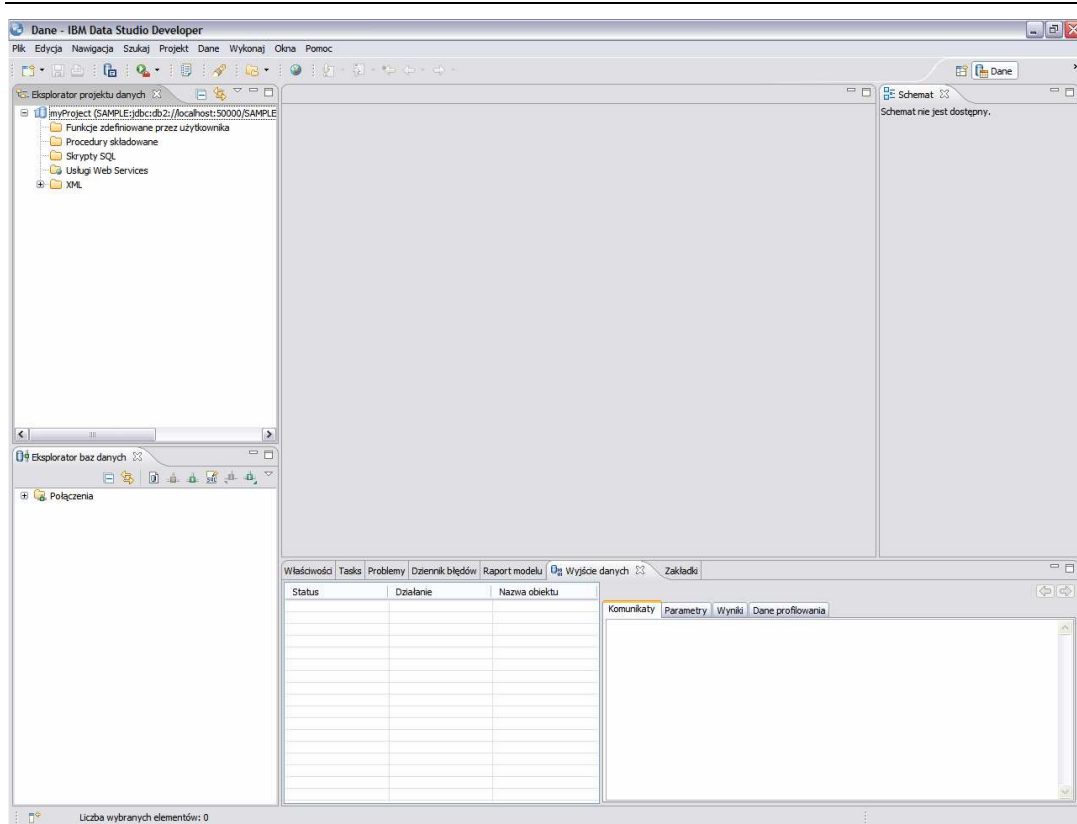
**Rysunek 14.3 – Projekt opracowywania danych**

Zgodnie z poleceniami kreatora wprowadź nazwę projektu, wskaż bazę danych, do której chcesz się podłączyć oraz określ katalog JDK (ten, który jest ustawiony domyślnie, zazwyczaj jest poprawny).

**Krok 2: Tworzenie procedury składowanej**

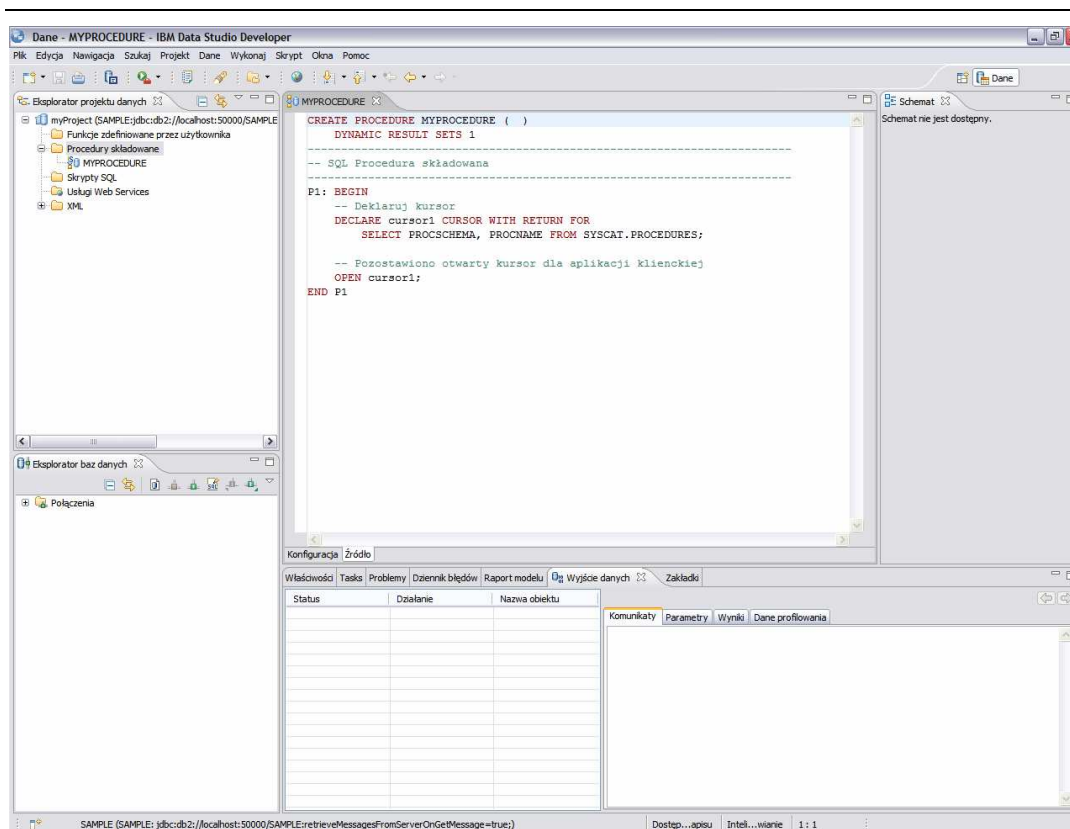
Utworzony projekt będzie widoczny w Eksploratorze projektu danych. Na Rysunku 14.4 pokazano rozszerzony widok projektu "myProject".





**Rysunek 14.4 – Projekt “myProject”**

Rysunek 14.4 ilustruje różne foldery, które należą do projektu. W celu utworzenia procedury składowanej, kliknij prawym przyciskiem myszy na folderze *Procedury składowane* i wybierz *Nowy* -> *Procedura składowana*. Uzupełnij informacje takie jak: projekt z którym ma być powiązana procedura składowana, nazwę oraz język procedury (zauważ, że IBM Data Studio wspiera tylko SQL PL i Java), oraz instrukcję SQL używane przez procedurę. Data Studio domyślnie wygeneruje przykładową instrukcję SQL. W tym momencie, po kliknięciu *Zakończ* zostanie utworzona procedura składowana. Tak, jak ilustruje Rysunek 14.5.

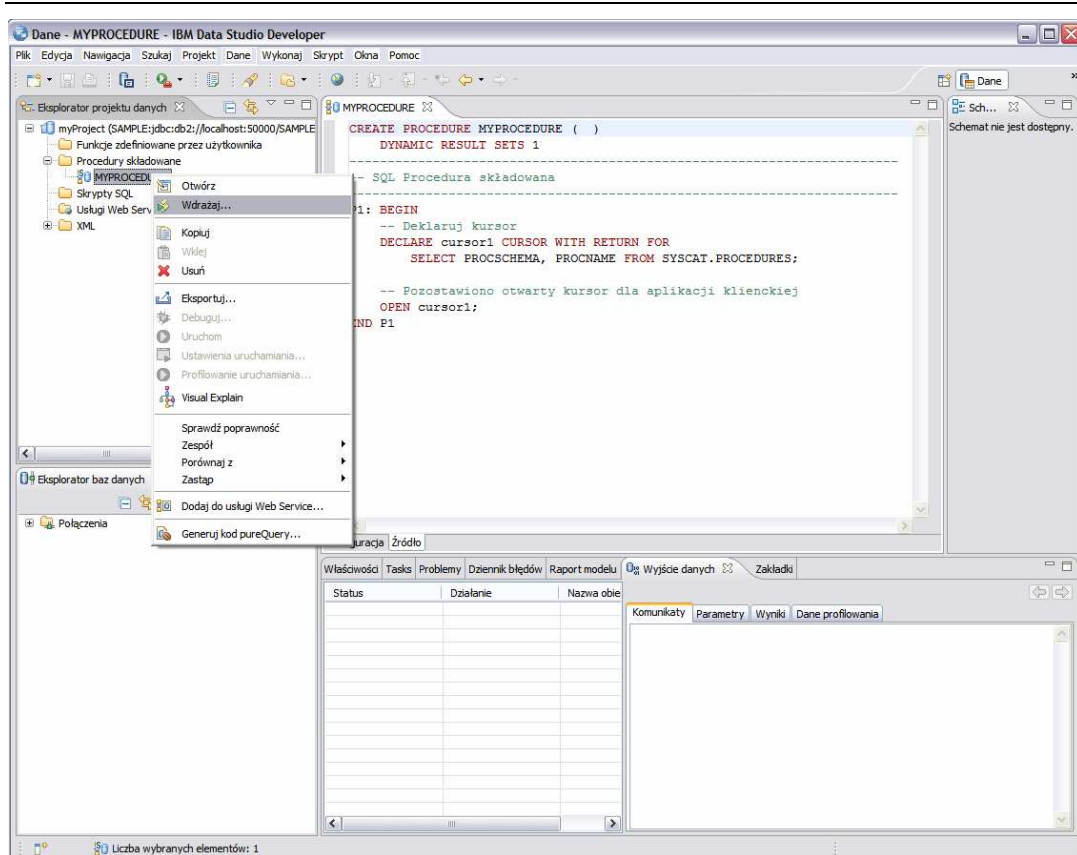


**Rysunek 14.5 – Przykładowa procedura składowana**

Rysunek 14.5 przedstawia wygenerowany kod przykładowej procedury składowanej “PROCEDURA1”. Powyższa przykładowa procedura składowana będzie używana w dalszej części książki jako domyślna.

### **Krok 3: Kompilowanie (wdrażanie) procedury składowanej**

Skompiluj i wdróż utworzoną procedurę składowaną klikając na niej prawym przyciskiem myszy w lewym panelu i wybierając “Wdrażaj...”. Krok ten przedstawia Rysunek 14.6.



Rysunek 14.6 – Wdrażanie procedury składowanej

#### Krok 4: Uruchamianie procedury składowanej

Uruchom wdrożoną procedurę składowaną klikając na niej prawym przyciskiem myszy i wybierając "Uruchom". Wyniki pojawią się w zakładce "Wyniki" w prawym dolnym rogu okna.

Do uruchomienia procedury składowanej z Okna Komend lub Edytora Komend, użyj składni CALL <procedure name>.

## 14.2 Podstawy procedur składowanych w SQL PL

Procedury składowane w proceduralnym języku SQL PL (ang. SQL Procedural Language) są łatwe, zarówno do tworzenia, jak i do nauki. Najlepszą wydajność osiągają w DB2. W tym rozdziale skupimy się na procedurach składowanych w SQL PL (lub prościej: „procedurach składowanych SQL”).

### 14.2.1 Struktura procedury składowanej

Oto podstawowa składnia procedury składowanej.

```
CREATE PROCEDURE proc_name [( {optional parameters} )]  
    [optional procedure attributes]    <statement>
```

gdzie <statement> jest pojedynczym wyrażeniem lub zbiorem wyrażień pogrupowanym przez BEGIN [ATOMIC] ... END

### 14.2.2 Dodatkowe atrybuty procedury składowanej

Poniżej opisano kilka dodatkowych atrybutów procedury składowanej:

- LANGUAGE SQL  
Wskazuje na język używany przez procedurę składowaną. Domyślna wartość to LANGUAGE SQL. Dla innych języków, takich jak Java lub C atrybut przyjmuje odpowiednio wartości LANGUAGE JAVA lub LANGUAGE C.
- RESULT SETS <n>  
Wymagany, jeśli procedura składowana będzie zwracać n zbiorów wyników.
- SPECIFIC my\_unique\_name  
Unikalna nazwa dla procedury. Procedura składowana może być przeciążana, tzn. kilka procedur składowanych może mieć taką samą nazwę, ale musi mieć inną liczbę parametrów. Słowo kluczowe SPECIFIC dostarcza unikalną nazwę dla każdej z tych procedur, co ułatwia zarządzanie nimi. Przykładowo, aby usunąć procedurę zawierającą atrybut SPECIFIC, wystarczy użyć: DROP SPECIFIC PROCEDURE. Gdyby jednak nie był wyszczególniony atrybut SPECIFIC, wówczas należałoby użyć: DROP PROCEDURE oraz podać nazwę i liczbę parametrów procedury; DB2 musi wiedzieć, którą z przeciążonych procedur należy usunąć.

### 14.2.3 Parametry

Wyróżniamy trzy typy parametrów w procedurach składowanych w SQL PL:

- IN – parametr wejściowy
- OUT – parametr wyjściowy
- INOUT – parametr wejściowy i wyjściowy

Na przykład:

```
CREATE PROCEDURE proc(IN p1 INT, OUT p2 INT, INOUT p3 INT)
```

Podczas wywoływania procedury w instrukcji CALL, muszą być uwzględnione wszystkie parametry. Przykładowo, aby wywołać powyższą procedurę składowaną należy użyć:

```
CALL proc (10,?,4)
```

Znak zapytania (?) w instrukcji CALL jest używany dla parametru OUT.

Oto inny przykład procedury składowanej z parametrami:

```
CREATE PROCEDURE P2 ( IN    v_p1 INT,
                    INOUT v_p2 INT,
                    OUT   v_p3 INT)

LANGUAGE SQL
SPECIFIC myP2
BEGIN
    -- my second SQL procedure
    SET v_p2 = v_p2 + v_p1;
    SET v_p3 = v_p1;
END
```

Aby wywołać procedurę z Edytora Komend wpisz:

```
call P2 (3, 4, ?)
```

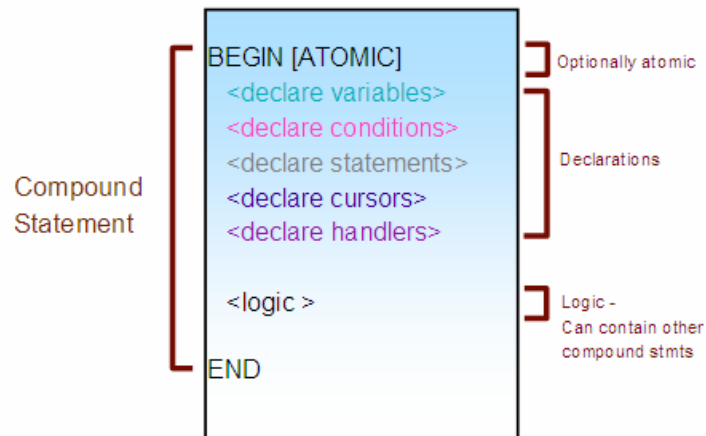
#### 14.2.4 Komentarze w procedurach składowanych w SQL PL

Są dwie możliwości umieszczania komentarzy w procedurach składowanych w SQL PL:

- Używając dwóch kresek. Na przykład:  
`-- This is an SQL-style comment`
- Używając formatu podobnego jak w języku C. Na przykład:  
`/* This is a C-style coment */`

#### 14.2.5 Instrukcje złożone

Instrukcja złożona w procedurze składowanej jest to instrukcja składająca się z kilku proceduralnych instrukcji oraz instrukcji SQL zawartych pomiędzy słowami kluczowymi BEGIN i END. Gdy po słowie kluczowym BEGIN występuje słowo kluczowe ATOMIC, instrukcja złożona traktowana jest jako jedna jednostka, tzn. wszystkie z instrukcji znajdujące się w instrukcji złożonej muszą się zakończyć powodzeniem, aby cała instrukcja złożona zakończyła się sukcesem. Jeśli jakkolwiek z instrukcji zakończy się niepowodzeniem, wówczas wszystkie instrukcje są wycofywane. Rysunek 14.7 przedstawia strukturę instrukcji złożonej.



Rysunek 14.7 – Instrukcje złożone

### 14.2.6 Deklaracje zmiennej

Do zadeklarowania zmiennej służy instrukcja DECLARE:

```
DECLARE var_name <data type> [DEFAULT value];
```

Oto kilka przykładów:

```

DECLARE temp1 SMALLINT DEFAULT 0;
DECLARE temp2 INTEGER DEFAULT 10;
DECLARE temp3 DECIMAL(10,2) DEFAULT 100.10;
DECLARE temp4 REAL DEFAULT 10.1;
DECLARE temp5 DOUBLE DEFAULT 10000.1001;
DECLARE temp6 BIGINT DEFAULT 10000;
DECLARE temp7 CHAR(10) DEFAULT 'yes';
DECLARE temp8 VARCHAR(10) DEFAULT 'hello';
DECLARE temp9 DATE DEFAULT '1998-12-25';
DECLARE temp10 TIME DEFAULT '1:50 PM';
DECLARE temp11 TIMESTAMP DEFAULT '2001-01-05-12.00.00';
DECLARE temp12 CLOB(2G);
DECLARE temp13 BLOB(2G);
  
```

### 14.2.7 Instrukcje przypisania

Do przypisania wartości do zmiennej służy instrukcja SET. Na przykład:

```
SET total = 100;
```

Powyższa instrukcja jest równoważna z:

```
VALUES(100) INTO total;
```

Dodatkowo, każda zmienna może być ustalona jako NULL:

```
SET total = NULL;
```

Wywołanie warunku; wybieranie tylko pierwszej wiersza z tabeli:

```
SET total = (select sum(c1) from T1);  
SET first_val = (select c1 from T1 fetch first 1 row only)
```

Można również ustalać zmienne w zależności od zewnętrznych właściwości bazy danych:

```
SET sch = CURRENT SCHEMA;
```

### 14.3 Kursory

Kursor jest zbiorem wynikowym, który przetrzymuje wyniki z instrukcji SELECT. Poniżej przedstawiona jest składnia deklaracji, otwierania, przechodzenia i zamykania kursora:

```
DECLARE <cursor name> CURSOR [WITH RETURN <return target>]  
    <SELECT statement>;  
OPEN <cursor name>;  
FETCH <cursor name> INTO <variables>;  
CLOSE <cursor name>;
```

Zadeklarowany kursor z parametrem WITH RETURN może być użyty z następującymi wartościami:

- CLIENT: zbiór wynikowy zostanie zwrócony do aplikacji klienta
- CALLER: zbiór wynikowy zwrócony jest do klienta lub do procedury składowanej, która dokonała wywołania

Oto przykład procedury składowanej używającej kursora:

```
CREATE PROCEDURE set()  
DYNAMIC RESULT SETS 1  
LANGUAGE SQL  
BEGIN  
    DECLARE cur CURSOR WITH RETURN TO CLIENT  
        FOR SELECT name, dept, job  
            FROM staff  
            WHERE salary > 20000;  
    OPEN cur;  
END
```

### 14.4 Sterowanie przepływem

Tak jak wiele innych języków, SQL PL posiada kilka instrukcji do logicznego sterowania przepływem. Poniżej znajduje się kilka wspieranych instrukcji sterowania przepływem:

CASE (wybiera ścieżkę wykonania (łatwe wyszukiwanie))

---

IF	
FOR	(przechodzi przez każdy wiersz z tabeli)
WHILE	
ITERATE	(wymusza kolejną iterację. Podobna do CONTINUE w C)
LEAVE	(opuszcza blok lub pętlę. "Strukturalne Goto")
LOOP	(pętla nieskończona)
REPEAT	
GOTO	
RETURN	
CALL	(wywoływanie procedury)

## 14.5 Wywoływanie procedur składowanych

Poniższe fragmenty kodów pokazują jak wywołać (CALL) procedury składowane napisane w różnych językach programowania.

### Przykład wywołania procedury składowanej w CLI/ODBC

```
SQLCHAR *stmt = (SQLCHAR *)
"CALL MEDIAN_RESULT_SET( ? )" ;
SQLDOUBLE sal = 20000.0; /* Bound to parameter marker in
stmt */
SQLINTEGER salind = 0; /* Indicator variable for sal */

sqlrc = SQLPrepare(hstmt, stmt, SQL_NTS);
sqlrc = SQLBindParameter(hstmt, 1, SQL_PARAM_OUTPUT,
SQL_C_DOUBLE, SQL_DOUBLE, 0, 0, &sal, 0, &salind);
SQLExecute(hstmt);

if (salind == SQL_NULL_DATA)
printf("Median Salary = NULL\n");
else
printf("Median Salary = %.2f\n\n", sal );

/* Get first result set */
sqlrc = StmtResultPrint(hstmt);
/* Check for another result set */
sqlrc = SQLMoreResults(hstmt);
if (sqlrc == SQL_SUCCESS) {
/* There is another result set */
sqlrc = StmtResultPrint(hstmt);
}
}
```

Więcej szczegółów w przykładowym pliku DB2: `sqllib/samples/sqlproc/resultset.c`.

### Przykład wywołania procedury składowanej w VB.NET

```
Try
    \ Create a DB2Command to run the stored procedure
    Dim procName As String = "TRUNC_DEMO"
```



```
Dim cmd As DB2Command = conn.CreateCommand()
Dim parm As DB2Parameter

cmd.CommandType = CommandType.StoredProcedure
cmd.CommandText = procName

` Register the output parameters for the DB2Command
parm          = cmd.Parameters.Add("v_lastname",
DB2Type.VarChar)
parm.Direction = ParameterDirection.Output
parm          = cmd.Parameters.Add("v_msg",
DB2Type.VarChar)
parm.Direction = ParameterDirection.Output

` Call the stored procedure
Dim reader As DB2DataReader = cmd.ExecuteReader

Catch myException As DB2Exception
  DB2ExceptionHandler(myException)
Catch
  UnhandledExceptionHandler()
End Try
```

#### Przykład wywołania procedury składowanej w Java

```
try
{
  // Connect to sample database
  String url = "jdbc:db2:sample";
  con = DriverManager.getConnection(url);

  CallableStatement cs = con.prepareCall("CALL
trunc_demo(?, ?)");

  // register the output parameters
  callStmt.registerOutParameter(1, Types.VARCHAR);
  callStmt.registerOutParameter(2, Types.VARCHAR);

  cs.execute();
  con.close();
}
catch (Exception e)
{
  /* exception handling logic goes here */
}
```

## 14.6 Błędy i przechwytywanie warunków

W DB2, do określenia pomyślnego lub niepomyślnego wykonania instrukcji SELECT używa się słów kluczowych SQLCODE i SQLSTATE. Słowa te muszą być jawnie zadeklarowane w zewnętrznym obszarze procedury:

```
DECLARE SQLSTATE CHAR(5);
DECLARE SQLCODE INT;
```

Po każdej operacji SQL, DB2 automatycznie ustawia wartości dla powyższych słów kluczowych. Dla SQLCODE, wartości mogą być następujące:

- = 0, pomyślnie
- > 0, pomyślnie z ostrzeżeniem
- < 0, niepomyślnie
- = 100, nie znaleziono danych. (to znaczy: instrukcja FETCH nie zwróciła danych)

Dla SQLSTATE, wartości mogą być następujące:

- sukces: SQLSTATE '00000'
- nie znaleziono: SQLSTATE '02000'
- ostrzeżenie: SQLSTATE '01XXX'
- wyjątek: wszystkie inne wartości

SQLCODE jest charakterystyczny dla systemów zarządzania relacyjnymi bazami danych (ang. Relational Database Management System - RDBMS) i bardziej szczegółowy niż SQLSTATE. SQLSTATE jest standardowy wśród RDBMS, ale z natury jest bardzo ogólny. Kilka SQLCODE może dorównać jednemu SQLSTATE. SQLCODE i SQLSTATE omówiono szczegółowo w Dodatku A „Rozwiązywanie problemów”.

Warunek może zostać przechwycony przez dowolną instrukcję SQL. Przykładowo, warunek SQLSTATE '01004' zostaje wywołany, gdy podczas operacji SQL dochodzi do ucięcia wartości. Zamiast używać SQLSTATE '01004' do warunku można przypisać nazwę. W poniższym przykładzie do warunku SQLSTATE '01004' została przypisana nazwa "trunc".

```
DECLARE trunc CONDITION FOR SQLSTATE '01004'
```

Oto inne predefiniowane warunki:

- SQLWARNING
- SQLEXCEPTION
- NOT FOUND

### Obsługa warunku

Aby obsłużyć warunek należy utworzyć warunek przechwytywania (ang. condition handler), który musi zawierać informacje:

- który warunek przechwycić
- gdzie wznowić wykonywanie (w oparciu o typ przechwytywania: CONTINUE, EXIT lub UNDO)

- akcje wykonywane w celu przechwycenia warunku. Akcją może być dowolna instrukcja, włączając strukturę sterowania.

W przypadku, gdy nie ma utworzonego warunku przechwytywania i wywołany zostaje warunek SQLEXCEPTION, procedura zostaje przerwana a klient otrzymuje komunikat o błędzie.

### Typy przechwytywań

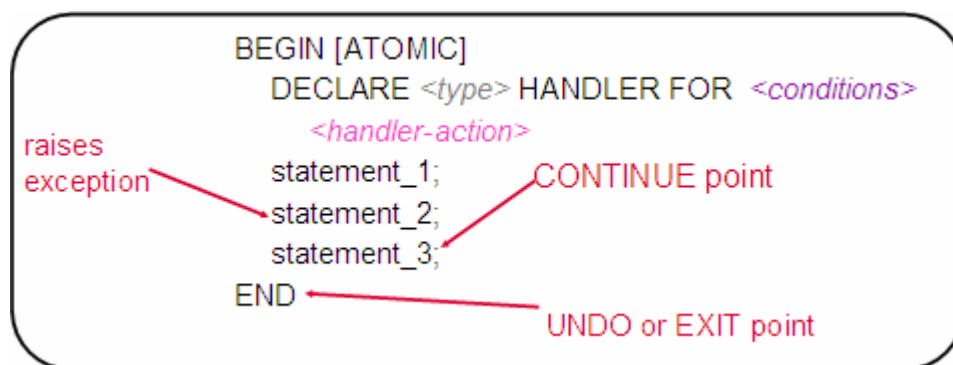
Wyróżniamy trzy typy przechwytywań:

**CONTINUE** – przepływ zostanie kontynuowany od instrukcji następującej po instrukcji, która wywołała warunek.

**EXIT** – przepływ zostanie skierowany na koniec procedury.

**UNDO** – przepływ zostanie skierowany na koniec procedury oraz wszystkie wykonane instrukcje zostaną wycofane.

Rysunek 14.8 ilustruje różne warunki przechwytywania i ich zachowania.



Rysunek 14.8 – Typy przechwytywań warunków

## 14.7 Dynamiczny SQL

W dynamicznym SQL, w przeciwieństwie do statycznego, podczas uruchamiania nie jest znana cała instrukcja SQL. Przykładowo, jeśli `col1` i `tablename` są zmiennymi w tej instrukcji, wówczas mamy do czynienia z dynamicznym SQL:

```
'SELECT ' || col1 || ' FROM ' || tablename;
```

Dynamiczny SQL należy stosować przy tworzeniu DDL, dzięki czemu można uniknąć problemów z zależnościami i unieważnieniem pakietów (ang. package). Jest on również wymagany przy implementacji rekursji.

Dynamiczny SQL może być wywołany na dwa sposoby:

- Instrukcją `EXECUTE IMMEDIATE` – idealna dla pojedynczych wywołań SQL
- Instrukcją `PREPARE` wraz z instrukcją `EXECUTE` – idealna dla wielokrotnych wywołań SQL

Poniższy fragment kodu przedstawia powyższe podejścia wywołania dynamicznego SQL. W przykładzie przyjęto następującą definicję tabeli T2:

```
CREATE TABLE T2 (c1 INT, c2 INT)

CREATE PROCEDURE dyn1 (IN value1 INT, IN value2 INT)
SPECIFIC dyn1
BEGIN
DECLARE stmt varchar(255);
DECLARE st STATEMENT;

SET stmt = 'INSERT INTO T2 VALUES (?, ?)';

PREPARE st FROM stmt;

EXECUTE st USING value1, value1;
EXECUTE st USING value2, value2;

SET stmt = 'INSERT INTO T2 VALUES (9,9)';
EXECUTE IMMEDIATE stmt;
END
```

---

# 15

## Rozdział 15 – Inline SQL PL, wyzwalacze i funkcje użytkownika

W tym rozdziale zajmiemy się Inline SQL PL, funkcjami użytkownika (ang. user-defined functions - UDF) oraz wyzwalaczami.

**Ważne:**

Więcej informacji o wyzwalaczach i funkcjach użytkownika uzyskasz oglądając wideo:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4367>

<http://www.channeldb2.com/video/video/show?id=807741:Video:4362>

### 15.1 Inline SQL PL

W Rozdziale 14 omówiliśmy tworzenie procedur składowanych w SQL PL. Język ten może być również użyty jako „inline”, tzn. istnieć samodzielnie bez potrzeby tworzenia procedury składowanej. SQL PL używany w funkcjach użytkownika i wyzwalaczach jest również typu inline; co więcej, jest to dynamiczny SQL. Inline SQL PL wspiera tylko pewien podzbiór instrukcji SQL PL. Oto słowa kluczowe wspierane przez inline SQL PL:

```
DECLARE <variable>
SET
CASE
FOR
GET DIAGNOSTICS
GOTO
IF
RETURN
SIGNAL
WHILE
ITERATE
LEAVE
```

Oto słowa kluczowe nie wspierane przez inline SQL PL:

```
ALLOCATE CURSOR
ASSOCIATE LOCATORS
```

```

DECLARE <cursor>
DECLARE ...HANDLER
PREPARE
EXECUTE
EXECUTE IMMEDIATE
LOOP
REPEAT
RESIGNAL
CALL
COMMIT/ROLLBACK

```

Poniżej znajduje się przykład złożonego dynamicznego SQL z użyciem inline SQL PL. Do poprawnego działania należy utworzyć poniższe tabele:

```

CREATE TABLE T1 (c1 INT)
CREATE TABLE T3 (c1 INT)

BEGIN ATOMIC
  DECLARE cnt          INT DEFAULT 0;
  DECLARE sumevens    INT DEFAULT 0;
  DECLARE err_msg     VARCHAR(1000) DEFAULT '';
  WHILE (cnt < 100) DO
    IF mod(cnt,2) = 0 THEN
      SET sumevens = sumevens + cnt;
    END IF;
    SET cnt=cnt+1;
  END WHILE;
  INSERT INTO T3 values (sumevens);
  SET cnt = (SELECT 0 FROM SYSIBM.SYSDUMMY1);
  FOR cur1 AS SELECT * FROM T1 DO
    IF cur1.c1 > 100 THEN
      SET cnt = cnt + 1;
    END IF;
  END FOR;

  SET err_msg = 'Rows with values > 100 is:' ||
char(cnt);
  SIGNAL SQLSTATE '80000' SET MESSAGE_TEXT = err_msg;
END!

```

Jeśli zapisaliśmy powyższy kod do pliku o nazwie “myScript.txt”, możemy wykonać go w następujący sposób:

```
db2 -td! -vf myScript.txt
```

## 15.2 Wyzwalacze

Wyzwalacze (ang. triggers) są obiektami bazy danych, które są powiązane z tabelą, dla której zdefiniowane są operacje realizowane, gdy na tabeli została wykonana operacja

INSERT, UPDATE lub DELETE. Wyzwalacze są aktywowane automatycznie. Operacje, które spowodowały uruchomienie wyzwalacza nazywamy wyzwalającymi instrukcjami SQL.

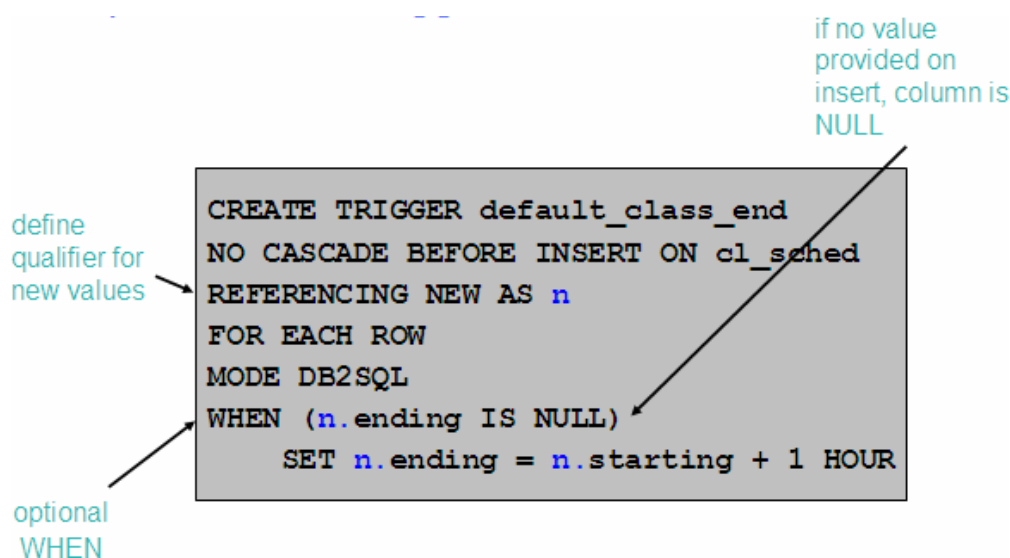
### 15.2.1 Rodzaje wyzwalaczy

Wyróżniamy trzy typy wyzwalaczy: „before”, „after” i „instead of”.

#### Wyzwalacze BEFORE

Wyzwalacze typu “before” są uruchamiane zanim wiersz zostanie wstawiony, zaktualizowany lub usunięty. Operacje wykonywane przez powyższy typ wyzwalacza nie mogą uruchamiać innych wyzwalaczy (dlatego operacje INSERT, UPDATE i DELETE są niedozwolone).

Prosty przykład wyzwalacza typu “before” przedstawia Rysunek 15.1.



Rysunek 15.1 – Przykład wyzwalacza typu „before”

Rysunek 15.1 ilustruje wyzwalacz „default\_class\_end”, który zostanie uruchomiony zanim wykona się instrukcja SQL INSERT na tabeli cl\_sched. Tabela ta jest częścią bazy danych SAMPLE. Po podłączeniu się do tej bazy możemy utworzyć i przetestować powyższy wyzwalacz. Zmienna „n” znajdująca się w definicji wyzwalacza będzie reprezentować nową wartość instrukcji INSERT, która jest wartością wstawianą. Wyzwalacz będzie sprawdzał ważność tego, co jest wstawiane do tabeli. Jeśli kolumna „ending” podczas dodawania rekordu nie ma przypisanej wartości, wyzwalacz wstawi wartość z kolumny „starting” dodając 1 godzinę.

Poniższe instrukcje pokazują jak przetestować wyzwalacz.

```

C:\Program Files\IBM\SQLLIB\BIN>db2 insert into cl_sched
(class_code, day, starting) values ('abc',1,current time)
DB20000I The SQL command completed successfully.
  
```

```
C:\Program Files\IBM\SQLLIB\BIN>db2 select * from cl_sched
```

CLASS_CODE	DAY	STARTING	ENDING
042:BF	4	12:10:00	14:00:00
553:MJA	1	10:30:00	11:00:00
543:CWM	3	09:10:00	10:30:00
778:RES	2	12:10:00	14:00:00
044:HD	3	17:12:30	18:00:00
abc	1	11:06:53	12:06:53

```
6 record(s) selected.
```

Poniżej przedstawiono wyzwalacz „validate\_sched”, który rozszerza funkcjonalność wcześniej opisanego wyzwalacza „default\_class\_end”. Tak jak poprzednio, możemy go utworzyć i przetestować wykorzystując bazę danych SAMPLE.

```
CREATE TRIGGER validate_sched
NO CASCADE BEFORE INSERT ON cl_sched
REFERENCING NEW AS n
FOR EACH ROW
MODE DB2SQL
BEGIN ATOMIC
-- supply default value for ending time if null
IF (n.ending IS NULL) THEN
    SET n.ending = n.starting + 1 HOUR;
END IF;

-- ensure that class does not end beyond 9pm
IF (n.ending > '21:00') THEN
    SIGNAL SQLSTATE '80000'
    SET MESSAGE_TEXT='class ending time is beyond 9pm';
ELSEIF (n.DAY=1 or n.DAY=7) THEN
    SIGNAL SQLSTATE '80001'
    SET MESSAGE_TEXT='class cannot be scheduled on a
weekend';
END IF;
END
```

### Wyzwalacze AFTER

Wyzwalacze typu “after” są uruchamiane, gdy wyzwalana instrukcja SQL zakończyła się powodzeniem. Operacje wykonywane przez powyższy wyzwalacz mogą uruchamiać inne wyzwalacze (dozwolone jest kaskadowanie do 16 poziomów). Wyzwalacze typu “after” wspierają operacje INSERT, UPDATE i DELETE. Poniżej znajduje się przykład wyzwalacza typu “after”.

```
CREATE TRIGGER audit_emp_sal
AFTER UPDATE OF salary ON employee
REFERENCING OLD AS o NEW AS n
```



```
FOR EACH ROW
MODE DB2SQL
  INSERT INTO audit VALUES (
    CURRENT TIMESTAMP, ' Employee ' || o.empno || ' sa-
lary changed from ' || CHAR(o.salary) || ' to ' ||
CHAR(n.salary) || ' by ' || USER)
```

W tym przykładzie wyzwalacz `audit_emp_sal` jest używany do sprawdzania kolumny „salary” w tabeli „employee”. Gdy ktoś dokona zmiany w tej kolumnie, uruchomiony zostanie wyzwalacz, który zapisze do tabeli „audit” informacje o dokonanych zmianach. Linia „OLD as o NEW as n” wskazuje, że przedrostek „o” będzie reprezentował starą lub istniejącą wartość w tabeli a przedrostek „n” będzie reprezentował nową wartość pochodzącą z instrukcji UPDATE. Zatem, „o.salary” reprezentuje starą lub istniejącą wartość kolumny salary, a „n.salary” wartość zaktualizowaną.

#### **Wyzwalacze „INSTEAD OF”**

Wyzwalacze typu „instead of” są definiowane na widokach. Logika zdefiniowana w wyzwalaczu jest wykonywana zamiast wyzwalanej instrukcji SQL. Przykładowo, jeśli dokonamy aktualizacji na widoku, powyższy wyzwalacz zostanie w rzeczywistości uruchomiony do dokonania aktualizacji tabel bazowych, na których opiera się widok.

Wyzwalacze nie mogą być tworzone w IBM Data Studio. Można je utworzyć z Centrum sterowania lub za pomocą Narzędzi uruchamianych z wiersza komend (Okno komend, Procesor wiersza komend lub Edytor komend).

## Laboratorium #12 – Tworzenie wyzwalacza w Centrum sterowania

### Cel

Wyzwalacze są obiektami bazy danych, które służą do wykonywania logiki biznesowej, gdy dochodzi do operacji modyfikującej dane w tabeli. W tym Laboratorium za pomocą Centrum sterowania utworzysz wyzwalacz, który będzie przetrzymywał zapis zmian dokonanych w tabeli *SALES*. Zarejestrujesz czas oraz identyfikator użytkownika, który dokonał tych zmian.

### Kroki

1. Uruchom Centrum sterowania.
2. W tym Laboratorium będziesz musiał utworzyć dodatkową tabelę, która będzie przechowywała logi. Utwórz tabelę o następujących własnościach:

Nazwa tabeli: `saleslog`

Pierwsza kolumna:

Nazwa: `userid`

Typ danych: `VARCHAR(128)`

Inne atrybuty: `NOT NULL`

Druga kolumna

Nazwa: `daytime`

Typ danych: `TIMESTAMP`

Inne atrybuty: `NOT NULL`

Wskazówka: Utwórz tabelę za pomocą Edytora komend instrukcją `CREATE TABLE` lub użyj *Kreatora tworzenia tabel* w Centrum sterowania.

3. Kliknij prawym przyciskiem myszy na folderze *Wyzwalacze* i wybierz opcję *Utwórz....* Pojawi się okno *Utwórz wyzwalacz*.
4. Wprowadź poniższe informacje:

Schemat wyzwalacza: Identyfikator użytkownika, którym jesteś zalogowany (powinno być ustawione domyślnie)

Nazwa wyzwalacza: `audit_sales`

Schemat tabeli lub widoku: Identyfikator użytkownika, którym jesteś zalogowany (powinno być ustawione domyślnie)

Nazwa tabeli lub widoku: *SALES*

Czas wyzwolenia działania: *Później niż*

Operacja powodująca uruchomienie wyzwalacza: *Aktualizacja kolumn* (nie określaj żadnej z kolumn, ponieważ chcemy odpalić wyzwalacz, gdy jakkolwiek z kolumn zostanie zaktualizowana).

Komentarz: *Logi wszystkich aktualizacji na tabeli Sales.*

5. W zakładce *Wyzwalane działanie* wybierz opcję *Dla każdego Instrukcja*. Użyj poniższego kodu dla wyzwalanego działania:

```
WHEN ( 1=1 )
BEGIN ATOMIC
  INSERT INTO saleslog (userid, daytime) VALUES (CURRENT
USER, CURRENT TIMESTAMP);
END
```

(*Ważne: Instrukcja: odpala się jednorazowo, gdy zakończy się instrukcja uruchamiająca wyzwalacz. Wiersz: powoduje, że wyzwalane działanie zostanie uruchomione za każdym razem, gdy instrukcja SQL wpłynie na wiersz.*)

**Utwórz wyzwalacz**

EMEM - DB2 - SAMPLE

Wyzwalacz | Wyzwalane działanie

Nazwa korelacji dla starych wierszy

Nazwa korelacji dla nowych wierszy

Tabela tymczasowa dla starych wierszy

Tabela tymczasowa dla nowych wierszy

Dla każdego  Wiersz  Instrukcja

Wyzwalane działanie

```
WHEN ( 1=1 )
BEGIN ATOMIC
  INSERT INTO saleslog (userid, daytime) VALUES (CURRENT USER, CURRENT
TIMESTAMP);
END
```

Cofnij

Wyczyść

Asysta SQL

OK Anuluj Pokaż SQL Pomoc

Aby utworzyć wyzwalacz kliknij *OK*.

6. Wyzwalacz powinien znajdować się w folderze *Wyzwalacze*.
7. Usuń wszystkie wiersze z tabeli *saleslog*. (`DELETE FROM saleslog`).
8. Zaktualizuj rekord w tabeli *sales*. (Wskazówka: użyj Edytora komend lub Asysty SQL).
9. Sprawdź zawartość tabeli *saleslog*. Ile zawiera wierszy?

### 15.3 Funkcje użytkownika

Funkcja użytkownika (ang user-defined function - UDF) jest obiektem bazy danych, który mapuje zbiór wartości wejściowych w zbiór wartości wyjściowych. Przykładowo, funkcja może pobierać miarę w calach i zwracać wynik w centymetrach.

DB2 wspiera tworzenie funkcji w SQL PL, C/C++, Java, CLR (Common Language Runtime) i OLE (Object Linking and Embedding). Ze względu na prostotę, popularność i wydajność zajmiemy się funkcjami napisanymi w języku SQL PL.

Wyróżniamy cztery typy funkcji: skalarne, tabelaryczne, wierszowe i kolumnowe. W tym rozdziale skupimy się tylko na funkcjach skalarnych i tabelarycznych.

#### 15.3.1 Funkcje skalarne

Funkcje skalarne zwracają pojedynczą wartość. Funkcje skalarne nie mogą zawierać instrukcji SQL, które zmieniają stan bazy. Oznacza to, że instrukcje INSERT, UPDATE i DELETE są niedozwolone. Oto kilka wbudowanych funkcji skalarnych: SUM(), AVG(), DIGITS(), COALESCE() i SUBSTR().

DB2 umożliwia tworzenie funkcji dostosowanych do potrzeb użytkownika, w których można zawrzeć często używane zapytania. Przykładowo, gdy przeprowadza się migrację aplikacji z bazy Oracle do DB2, często wywołuje się funkcję Oracle o nazwie NVL(). Odpowiednikiem tej funkcji w DB2 jest wbudowana funkcja o nazwie COALESCE. Zamiast zmieniać nazwy wszystkich funkcji NVL, możemy utworzyć funkcję zdefiniowaną przez użytkownika w DB2 o nazwie NVL i wywołać funkcję COALESCE tak, jak pokazano poniżej.

```
CREATE FUNCTION NVL (p_var1 VARCHAR(30),
                    p_var2 VARCHAR(30))
SPECIFIC nvlvarchar30
RETURNS VARCHAR(30)
RETURN COALESCE(p_var1, p_var2)
```

Funkcja COALESCE zwróci pierwszy niezerowy argument.

Poniżej przedstawiono kolejny przykład funkcji skalarnej. Funkcja o nazwie "deptname" bazująca na id pracownika zwróci numer działu pracownika:

```
CREATE FUNCTION deptname(p_empid VARCHAR(6))
RETURNS VARCHAR(30)
SPECIFIC deptname
BEGIN ATOMIC
  DECLARE v_department_name VARCHAR(30);
  DECLARE v_err VARCHAR(70);
  SET v_department_name = (
    SELECT d.deptname FROM department d, employee e
    WHERE e.workdept=d.deptno AND e.empno= p_empid);
  SET v_err = 'Error: employee ' || p_empid || ' was not
found';
  IF v_department_name IS NULL THEN
    SIGNAL SQLSTATE '80000' SET MESSAGE_TEXT=v_err;
```

```

        END IF;
    RETURN v_department_name;
END

```

Aby przetestować funkcję wpisujemy poniższe polecenie w Oknie komend lub powłocie Linux/UNIX:

```
db2 "values (deptname ('000300'))"
```

### Wywoływanie skalarnych UDF

Skalarnie UDF mogą być wywoływane w instrukcjach SQL, gdy oczekiwana jest wartość skalarna, lub w klauzuli VALUES. Oto dwa przykłady przedstawiające wywołanie skalarnej funkcji COALESCE:

```

SELECT DEPTNAME, COALESCE(MGRNO, 'ABSENT') FROM DEPARTMENT

VALUES COALESCE('A', 'B')

```

### 15.3.2 Funkcje tabelaryczne

Funkcje tabelaryczne zwracają tablicę wierszy. Możemy je wywołać za pomocą klauzuli FROM. W przeciwieństwie do funkcji skalarnych, funkcje tabelaryczne mogą zmieniać stan bazy danych, dlatego też instrukcje INSERT, UPDATE i DELETE są dozwolone. Oto kilka wbudowanych funkcji tabelarycznych: SNAPSHOT\_DYN\_SQL() i MQREADALL(). Funkcje tabelaryczne są podobne do widoków, z tą różnicą, że pozwalają na modyfikację danych (INSERT, UPDATE i DELETE). Zazwyczaj są używane do zwracania tablicy i przechowywania rekordu kontrolnego.

Poniżej przedstawiono przykład funkcji tabelarycznej, która wylicza zbiór pracowników z danego działu:

```

CREATE FUNCTION getEnumEmployee(p_dept VARCHAR(3))
RETURNS TABLE
    (empno CHAR(6),
     lastname VARCHAR(15),
     firstnme VARCHAR(12))
SPECIFIC getEnumEmployee
RETURN
    SELECT e.empno, e.lastname, e.firstnme
    FROM employee e
    WHERE e.workdept=p_dept

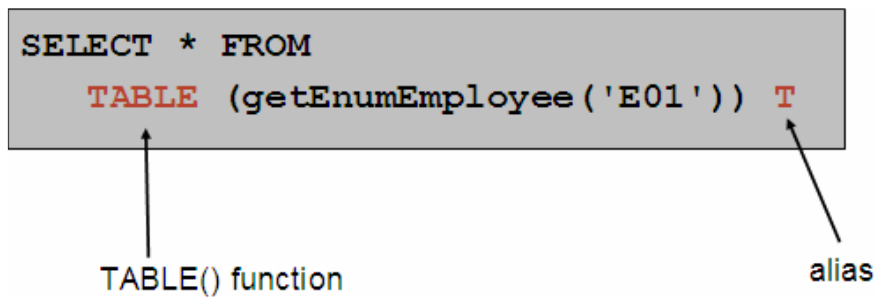
```

Aby przetestować powyższą funkcję wpisz:

```
db2 "SELECT * FROM table(getEnumEmployee('D11')) AS t"
```

**Wywoływanie tabelarycznych UDF**

Tabelaryczny UDF wywołuje się za pomocą klauzuli FROM w instrukcji SQL. Funkcja TABLE() musi posiadać alias. Rysunek 15.2 ilustruje jak wywołać funkcję "getEnumEmployee".



Rysunek 15.2 – Wywoływanie funkcji tabelarycznej.

## Laboratorium #13 – Tworzenie UDF w IBM Data Studio

### Cel

W tym Laboratorium utworzysz w IBM Data Studio skalarną funkcję użytkownika. Dzięki temu zdobędziesz dodatkowe doświadczenie w pracy z Data Studio, a także poszerzysz swoją wiedzę dotyczącą SQL PL dla funkcji użytkownika.

### Kroki

1. Uruchom IBM Data Studio (Wskazówka: jest dostępne z menu Start).
2. Z okna *Eksplorator projektu danych* wybierz projekt, który utworzyłeś w poprzednim Laboratorium i wybierz *Otwórz projekt*.
3. Prawym przyciskiem myszy kliknij na folder *Funkcje zdefiniowane przez użytkownika*. Z menu wybierz *Nowy*, a następnie *Funkcja zdefiniowana przez użytkownika*.
4. W widoku edycji powinien pojawić się szkielet funkcji. Zmodyfikuj kod, tak jak poniżej:

```
CREATE FUNCTION booktitle(p_bid INTEGER)
RETURNS VARCHAR(300)
-----
SQL UDF (Scalar)
-----
SPECIFIC booktitle
F1: BEGIN ATOMIC
DECLARE v_book_title VARCHAR(300);
DECLARE v_err VARCHAR(70);
SET v_book_title = (SELECT title FROM books WHERE p_bid = bo-
ok_id);
SET v_err = 'Error: The book with ID ' || CHAR(p_bid) || '
was not found.';
IF v_book_title IS NULL THEN SIGNAL SQLSTATE '80000' SET
MESSAGE_TEXT=v_err;
END IF;
RETURN v_book_title;
END
```

5. Zbuduj funkcję klikając na niej prawym przyciskiem myszy i wybierając opcję *Wdrażaj*.
6. Uruchom stworzoną funkcję klikając *Wykonaj* na pasku narzędzi.



7. Jako, że stworzona funkcja przyjmuje jeden parametr, wyświetlone zostanie okno dialogowe, w które należy wpisać wartość parametru.

Wprowadź wartość: 80002

Jaki jest wynik?

Wprowadź wartość: 1002

Co stało się tym razem? (Wskazówka: Spójrz na zakładkę *Komunikaty* w sekcji *Wyjście danych*).

8. Po zakończeniu zamknij IBM Data Studio.

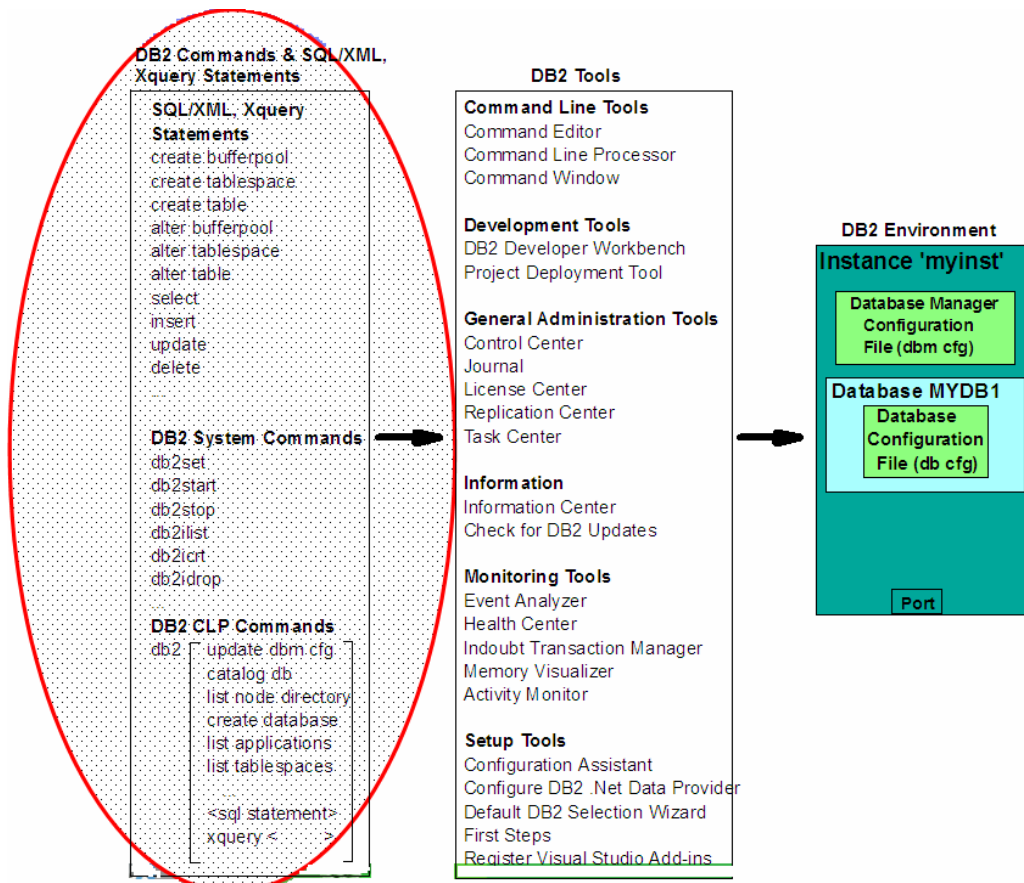


# 16

## Rozdział 16 – SQL/XML i XQuery

W rozdziale tym omówimy pureXML - technologię dostarczaną wraz z DB2, wpierającą natywne przechowywanie danych XML. Wiele z przykładów i wniosków zaprezentowanych w tym rozdziale zostało zaczerpniętych z IBM Redbook: *DB2 9: pureXML overview and fast start*. Aby dowiedzieć się więcej, zajrzyj do sekcji Źródła informacji o DB2.

Rysunek 16.1 pokazuje obszar zagadnień DB2, który będziemy omawiać w tym rozdziale.



Rysunek 16.1 – DB2: polecenia DB2, SQL/XML oraz XQuery

**Ważne**

Aby dowiedzieć się więcej o pureXML obejrzyj wideo:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4382>

## 16.1 Wykorzystanie XML w bazie danych

Dokumenty XML mogą być przechowywane w plikach tekstowych, repozytoriach XML oraz bazach danych.

Są dwa główne powody, dla których tak wiele firm decyduje się na przechowywanie plików XML w bazach danych:

- Mapowanie wielkich plików XML podpada pod zadania baz danych. Pliki XML to takie same dane jak każde inne, tylko mają trochę inny format. Z tych samych powodów relacyjne bazy danych pasują do plików XML: Bazy danych udostępniają wydajne przechowywanie i odczytywanie, solidne wsparcie dla trwałości danych, mechanizmy tworzenia kopii zapasowych i odzyskiwania danych, wsparcie dla transakcji, wydajności i skalowalności.
- Spójność: poprzez przechowywanie zależnych od siebie dokumentów XML razem, możemy dołączyć nowe dane XML do istniejących już zależności oraz łączyć SQL z XPath i XQuery w jednym zapytaniu. Ponadto, dane relacyjne mogą być publikowane jako XML i vice versa. Dzięki integracji z bazami danych można lepiej wspierać aplikacje Webowe SOA i usługi sieciowe (Web Services).

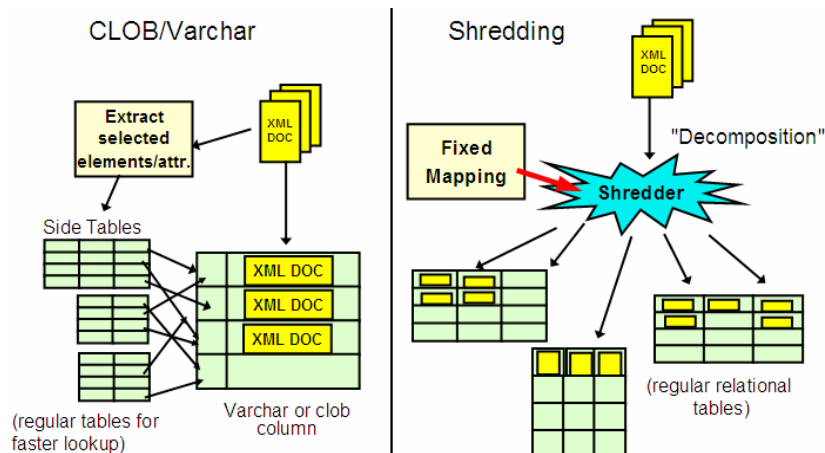
## 16.2 Bazy danych XML

Istnieją dwa typy baz danych do przechowywania danych XML:

- XML-owe bazy danych
- natywne bazy danych XML

### 16.2.1 XML-owe bazy danych

XML-owe bazy danych wykorzystują model relacyjny do przechowywania podstawowych danych modelu. Wymaga to mapowania pomiędzy modelem danych XML (hierarchicznym), a relacyjnym, albo przechowywania danych XML jako duże pliki tekstowe. Choć można to uznać za przestarzałą technologię, jest ona wciąż używana przez wielu dostawców baz danych. Rysunek 16.2 wyjaśnia dokładnie obie opcje przechowywania dla XML-owych baz danych.



**Rysunek 16.2 – Dwie możliwości przechowywania danych w XML-owych bazach danych.**

Po lewej stronie Rysunku 16.2 widzimy metodę "CLOB/Varchar" służącą do przechowywania danych XML. Gdy używamy tej metody, dokument XML jest przechowywany jako ciąg znaków lub CLOB bez jakiegokolwiek przetworzenia. Jeśli chcemy odczytać część dokumentu XML, nasz program musi przetworzyć cały ciąg znaków i znaleźć to, czego szukamy. Nie jest to metoda bardzo elastyczna.

Inną możliwością przechowywania danych dla XML-owych baz danych jest ich rozdrabnianie (ang. shredding). Jest ona przedstawiona na Rysunku 16.2 po prawej stronie. Przy wykorzystaniu tej metody cały dokument XML jest cięty na mniejsze części, które są przechowywane w tabelach. Hierarchiczny model dokumentu XML jest przekształcany na model relacyjny. Nie sprzyja to elastyczności: wprowadzanie zmian w dokumencie XML (do tabel odpowiadających danemu dokumentowi) nie jest już takie łatwe. Konieczne może być także utworzenie wielu dodatkowych tabel. Nie jest to też najlepsza metoda pod względem wydajności. Jeśli chcemy otrzymać oryginalne dane musimy wykonać kosztowne operacje SQL, które mogą być tym kosztowniejsze, im więcej tabel jest połączonych.

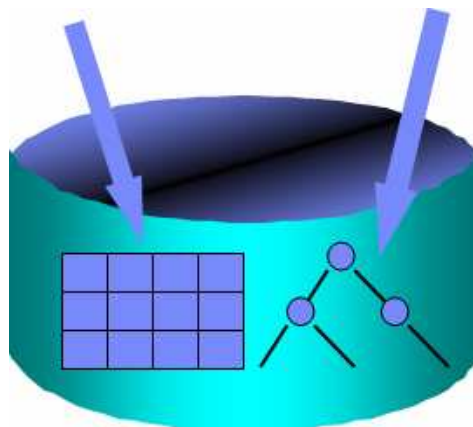
### 16.2.2 Natywne bazy danych XML

Natywne bazy XML wykorzystują hierarchiczny model danych XML do przechowywania i obróbki danych. Format przechowywanych danych jest taki sam, jak danych wejściowych, nie następuje żadne mapowanie na model relacyjny, i dokumenty XML nie są przechowywane jako duże obiekty. Podczas użycia zapytań XPath lub XQuery są one wykonywane natywnie przez silnik, a nie przerabiane na zapytania SQL. Dlatego takie bazy danych nazywamy natywnymi. DB2 od wersji 9 jest jedynym komercyjnym serwerem danych wykorzystującym i posiadającym tę opcję.

### 16.3 XML w DB2

Rysunek 16.3 przedstawia w zarysie, jak dane relacyjne i hierarchiczne XML są przechowywane w DB2, zaczynając od wersji 9. Zakładamy, że tabela `dept` jest określona w następujący sposób:

```
CREATE TABLE dept (deptID CHAR(8),..., deptdoc XML);
```

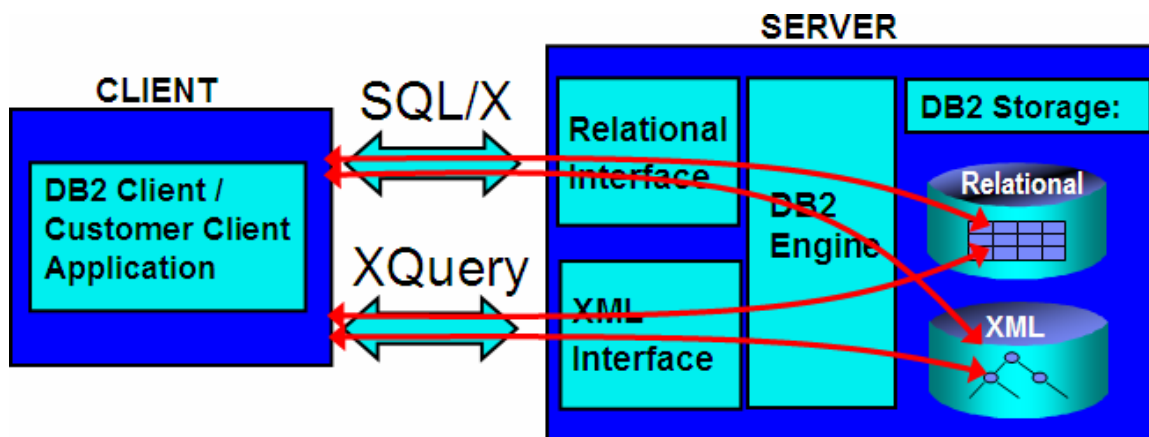


Rysunek 16.3 – XML w DB2

Zauważmy, że definicja tabeli używa nowego typu danych XML dla kolumny `deptdoc`. Lewa strzałka na rysunku wskazuje relacyjną kolumnę `deptID` przechowującą dane w formacie relacyjnym, a kolumna XML `deptdoc` w formacie hierarchicznym.

Rysunek 16.4 pokazuje, że począwszy od DB2 9, istnieją obecnie cztery sposoby dostępu do danych:

- przy użyciu SQL do danych relacyjnych
- przy użyciu SQL z rozszerzeniami XML do danych XML
- przy użyciu XQuery do danych XML
- przy użyciu XQuery do danych relacyjnych



Rysunek 16.4 – Cztery sposoby dostępu do danych w DB2

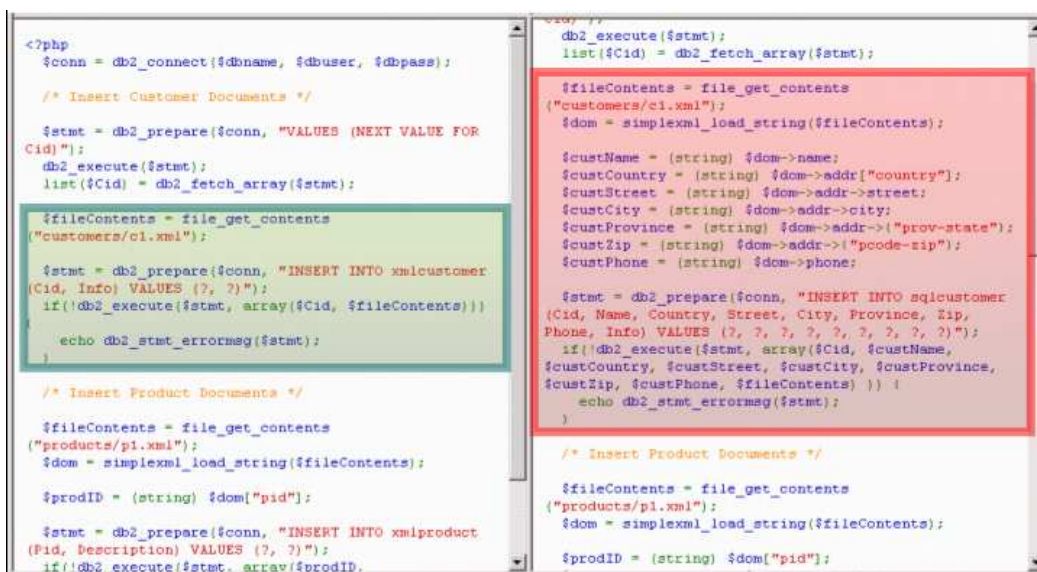
Podczas gdy osoba pracująca z SQL może widzieć DB2 jako świat klasy RDBMS, która także wspiera XML, osoba zajmująca się XML-em może widzieć DB2 jako świat klasy repozytorium, która też wspiera XML.

Zauważmy, że IBM używa terminu *pureXML* zamiast „natywny XML” do opisu tej technologii. Inni producenci używają starych technologii używających typu CLOB lub łańcucha znaków albo rozdrabniania dokumentu XML i nazywają je „natywnym XML”. Aby uniknąć nieporozumień, IBM zdecydował się użyć nowego terminu - *pureXML*, jako znaku towarowego, także aby zapobiec wykorzystywaniu tego określenia przez innych producentów do oznaczania nim innych technologii. W wersji 9.1 DB2, technologia *pureXML* mogła zostać użyta tylko wtedy, gdy baza danych była stworzona ze stroną kodową Unicode. W wersji 9.5 zniesiono to ograniczenie i *pureXML* można wykorzystywać w bazach danych także ze stroną kodową inną niż Unicode.

### 16.3.1 Zalety technologii *pureXML*

Ze stosowania technologii *pureXML* wynika wiele korzyści:

- Możemy bezproblemowo i swobodnie wykorzystywać relacyjność, biorąc pod uwagę, że dokumenty XML są przechowywane w kolumnach tabel przy użyciu nowego typu danych XML.
- Możemy zmniejszyć złożoność kodu. Na Rysunku 16.5 przedstawiony jest przykładowy skrypt PHP, z użyciem i bez użycia *pureXML*. Korzystanie z *pureXML* (zielona ramka) pozwala zmniejszyć liczbę linii kodu. Oznacza to nie tylko, że kod jest mniej skomplikowany, ale ogólne wyniki wydajnościowe mogą się poprawić, ponieważ jest mniej kodu do analizy i utrzymania.

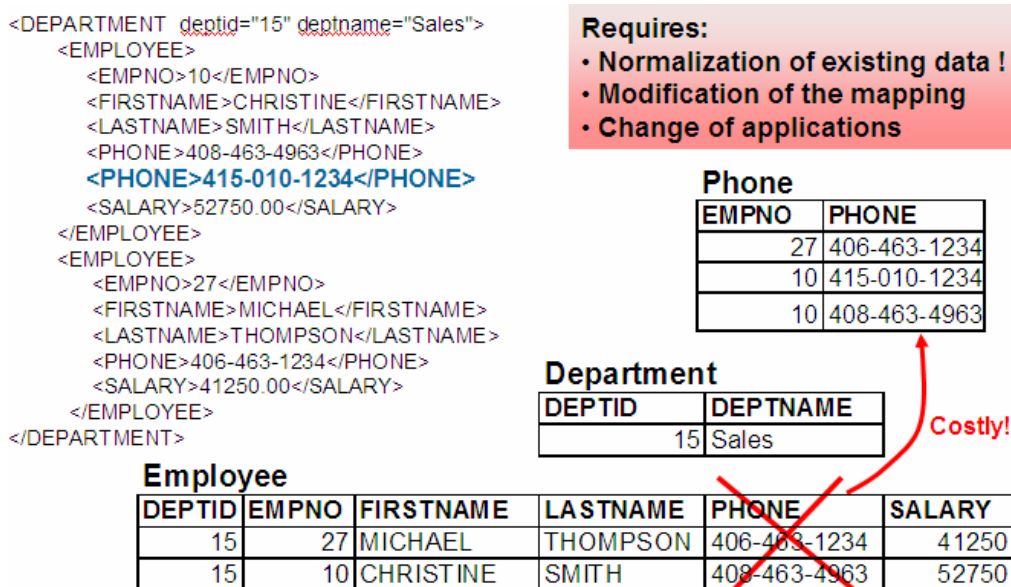


Rysunek 16.5 – Złożoność kodu z i bez użycia *pureXML*

- Zmiany w schemacie są łatwiejsze przy użyciu XML i technologii *pureXML*. Rysunek 16.6 pokazuje przykład zwiększonej elastyczności. Załóżmy, że mamy do dyspozycji bazę danych składającą się z tabel *Employee* i *Department*. Pracując z nie-XML-ową bazą danych, gdy przełożony poleci nam zapamiętać, oprócz numeru telefonu do domu pracownika, także jego numer telefonu komórkowego, zwykle aby

wprowadzić takie zmiany dodalibyśmy nową kolumnę do tabeli Employee i w niej zapisalibyśmy numer telefonu komórkowego. Metoda ta jednak byłaby sprzeczna relacyjnych zasadami normalizacji relacyjnych baz danych. Jeśli chcemy zachować te zasady, należałoby utworzyć nową tabelę Phone i przenieść do niej wszystkie informacje o numerach telefonów. Do tej tabeli można by także dodawać informacje o numerach telefonów komórkowych. Tworzenie nowej tabeli "Phone" jest kosztowne, nie tylko dlatego, że duża ilość wcześniej istniejących danych musi zostać przeniesiona, ale również dlatego, że wszystkie zapytania SQL w aplikacji będą musiały zostać zmienione, aby wskazywały na nową tabelę.

Zamiast tego, po lewej stronie Rysunku 16.6, pokazujemy, jak można to zrobić za pomocą języka XML. Jeżeli pracownik "Christine" posiada również numer telefonu komórkowego, aby wprowadzić tę informację, wystarczy dodać nowy tag XML. Jeżeli pracownik "Michael" nie ma numeru telefonu komórkowego, może zostać bez zmian.



Rysunek 16.6 – Większa elastyczność danych przy użyciu XML

- Możemy poprawić wydajność aplikacji XML. Badania wykonane z użyciem technologii pureXML wykazały ogromną poprawę wydajności dla wielu aplikacji XML. Rysunek 16.7 pokazuje wyniki testu dla firmy, która zrezygnowała ze starszych technologii na rzecz pureXML. W środkowej kolumnie przedstawione są wyniki uzyskane przy pomocy starej metody pracy z XML, trzecia kolumna pokazuje wyniki otrzymana przy użyciu pureXML.



Cel	Inne relacyjne bazy danych	DB2 z pureXML
Wyszukiwanie i pobieranie w procesach biznesowych	CLOB: 8 hrs Shred: 2 hrs	30 min.
Zależne polecenia wejścia/wyjścia	100	35 (zysk 65%)
Dodanie pola do schematu	1 tydzień	5 min.
Zapytania	24 - 36 godzin	20 sekund - 10 min

Rysunek 16.7 – Zwiększona wydajność przy użyciu technologii pureXML

### 16.3.2 Podstawy XPath

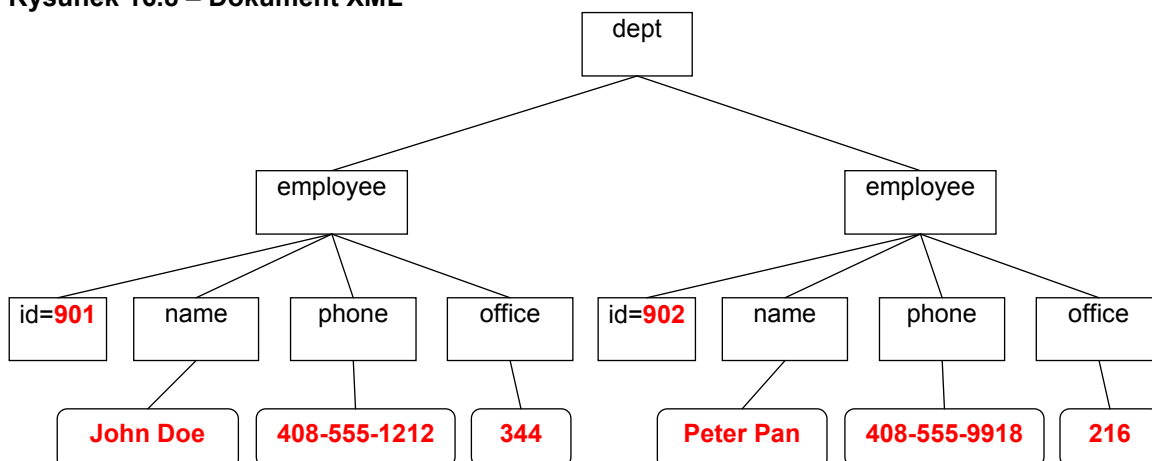
XPath jest językiem, który może być używany do odpytywania dokumentów XML. Rysunek 16.8 pokazuje przykładowy dokument XML, a Rysunek 16.9 przedstawia ten sam dokument reprezentowany w postaci struktury drzewiastej. Będziemy korzystać z postaci struktury drzewiastej, aby wyjaśnić w jaki sposób działa XPath.

```

<dept bldg="101">
  <employee id="901">
    <name>John Doe</name>
    <phone>408 555 1212</phone>
    <office>344</office>
  </employee>
  <employee id="902">
    <name>Peter Pan</name>
    <phone>408 555 9918</phone>
    <office>216</office>
  </employee>
</dept>

```

Rysunek 16.8 – Dokument XML



Rysunek 16.9 – Reprezentacja drzewiasta dokumentu XML z Rysunku 16.8

Szybkim sposobem, aby nauczyć się XPath, jest porównanie go do polecenia `change directory (cd)` w MS-DOS albo Linux/Unix. Przy użyciu polecenia `cd` poruszasz się po drzewie w taki sposób:

```
cd /directory1/directory2/...
```

Podobnie w XPath używasz symbolu „slash”, aby przejść od jednego elementu do drugiego w dokumencie XML. Na przykład w dokumencie z Rysunku 16.9 możesz odczytać nazwiska wszystkich pracowników przy użyciu następującego zapytania XPath:

```
/dept/employee/name
```

### Wyrażenia XPath

Wyrażenia XPath używają w pełni kwalifikowanych ścieżek do określenia elementów i/lub atrybutów. Aby pobrać tylko wartość (węzeł tekstowy) elementu, używamy funkcji „text()”. Tabela 16.1 pokazuje zapytania XPath i odpowiadające im wyniki, przy użyciu dokumentu XML z Rysunku 16.9.

XPath	Wynik
/dept/@bldg	101
/dept/employee/@id	901 902
/dept/employee/name	<name>Peter Pan</name> <name>John Doe</name>
/dept/employee/name/text()	Peter Pan John Doe

Tabela 16.1 – Przykłady wyrażen XPath

### Znaki specjalne XPath

Wyróżniamy dwa główne znaki specjalne w XPath:

- “\*” - odpowiada dowolnemu tagowi
- “//” - znak „jestem-swoim-przodkiem”

Tabela 16.2 zawiera więcej przykładów z wykorzystaniem dokumentu XML z Rysunku 16.9

XPath	Wynik
/dept/employee/*/text()	John Doe 408 555 1212 344 Peter Pan 408 555 9918 216

/dept*/@id	901 902
//name/text()	Peter Pan John Doe
/dept//phone	<phone>408 555 1212</phone> <phone>408 555 9918</phone>

Tabela 16.2 – Przykłady znaków specjalnych w XPath

**Preedykаты w XPath**

Preedykаты są zawarte w nawiasach kwadratowych [ ]. Możemy je porównać do wyrażenia WHERE w SQL. Na przykład [`@id="902"`] może być przeczytane jako: „gdzie atrybut id jest równy 902”. Wiele preedykatów może być zdefiniowanych w jednym zapytaniu XPath. Aby wskazać pozycję szukanego elementu używamy preedykatu [`n`], co oznacza, że będzie wybrany n-ty element. Na przykład, `pracownik[2]` oznacza, że zostaną zwrócone dane o drugim pracowniku. Tabela 16.3 zawiera więcej przykładów:

XPath	Wynik
/dept/employee[@id="902"]/name	<name>Peter Pan</name>
/dept[@bldg="101"]/employee[office > "300"]/name	<name>John Doe</name>
//employee[office="344" OR office="216"]/@id	901 902
/dept/employee[2]/@id	902

Table 16.3 – Przykłady preedykatów w XPath

**XPath: oznaczenie rodzica**

Podobnie do MS-DOS lub Linux/UNIX, możemy użyć znaku „.” (kropka), aby wyrazić, że odnosimy się do bieżącego kontekstu, a „...” (kropka kropka) aby wyrazić, że odnosimy się w kontekście rodzica.

Tabela 16.4 zawiera więcej przykładów:

XPath	Wynik
/dept/employee/name[../@id="902"]	<name>Peter Pan</name>
/dept/employee/office[.>"300"]	<office>344</office>
/dept/employee[office > "300"]/office	<office>344</office>
/dept/employee[name="John Doe"]/..@bldg	101
/dept/employee/name[.="John Doe"]/...@bldg	101

Tabela 16.4 – Oznaczenie rodzica w XPath

### 16.3.3 Definicja XQuery

XQuery jest językiem stworzonym dla XML. Obsługuje on ścieżki wyrażeń pozwalające przechodzić po hierarchicznej strukturze XML. W rzeczywistości, XPath jest podzbiorem XQuery, dlatego wszystko, czego nauczyłeś się wcześniej na temat XPath dotyczy także XQuery. XQuery obsługuje zarówno dane znanych typów, jak i pozostałe. XQuery nie dostarcza wartości null, ponieważ dokumenty XML pomijają wartości brakujące i nieznanne. XQuery zwraca sekwencje danych w postaci XML.

Ważne jest, aby pamiętać, że w wyrażeniach XPath i XQuery uwzględniana jest wielkość liter.

XQuery wspiera wyrażenia FLWOR. W kontekście języka SQL, odpowiadałoby ono wyrażeniu SELECT-FROM-WHERE. W następnym sekcji opisano wyrażenia FLWOR bardziej szczegółowo.

### XQuery: Wyrażenia FLWOR

FLWOR oznacza:

- FOR - przechodzenie po kolekcji, przypisywanie elementu do zmiennej
- LET - przypisywanie elementy do kolekcji
- WHERE - usuwanie elementów z kolekcji
- ORDER - zmiana kolejności elementów w kolekcji
- RETURN - zwracanie wyniku

Jest to wyrażenie, które umożliwia manipulowanie dokumentami XML, pozwalające zwracać nowe wyrażenie. Na przykład: Załóżmy, że mamy tabelę o następującej strukturze:

```
CREATE TABLE dept (deptID CHAR(8), deptdoc XML);
```

i poniższy dokument XML znajduje się w kolumnie deptdoc:

```
<dept bldg="101">
  <employee id="901">
    <name>John Doe</name>
    <phone>408 555 1212</phone>
    <office>344</office>
  </employee>
  <employee id="902">
    <name>Peter Pan</name>
    <phone>408 555 9918</phone>
    <office>216</office>
  </employee>
</dept>
```

Wyrażenie XQuery razem z wyrażeniem FLWOR może wyglądać następująco:

```
xquery
for $d in db2-fn:xmlcolumn('dept.deptdoc')/dept
let $emp := $d//employee/name
```

```

where $d/@bldg > 95
order by $d/@bldg
return
  <EmpList>
  {$d/@bldg, $emp}
  </EmpList>

```

Zwróci ono następujący rezultat:

```

<EmpList bldg="101">
  <name>
    John Doe
  </name>
  <name>
    Peter Pan
  </name>
</EmpList>

```

#### 16.3.4 Dodawanie dokumentów XML

Dodawanie dokumentów XML do bazy danych DB2 może zostać wykonane za pomocą polecenia INSERT języka SQL, albo polecenia IMPORT. Nie można tutaj użyć XQuery, ponieważ nie zostało to jeszcze ujęte w standardzie.

Przeanalizujmy poniższy skrypt, który można wykonać z Okna komend DB2, jak i powłoki Linux:

```
db2 -tvf table_creation.txt
```

##### table\_creation.txt

```

-- (1)
drop database mydb
;

-- (2)
create database mydb using codeset UTF-8 territory US
;

-- (3)
connect to mydb
;

-- (4)
create table items (
  id          int primary key not null,
  brandname   varchar(30),
  itemname    varchar(30),
  sku         int,
  srp         decimal(7,2),
  comments    xml
);

```

```
-- (5)
create table clients(
  id          int primary key not null,
  name        varchar(50),
  status      varchar(10),
  contact     xml
);

-- (6)
insert into clients values (77, 'John Smith', 'Gold',
  '<addr>111 Main St., Dallas, TX, 00112</addr>')
;

-- (7)
IMPORT FROM "D:\Raul\clients.del" of del xml from "D:\Raul" INSERT
INTO CLIENTS (ID, NAME, STATUS, CONTACT)
;

-- (8)
IMPORT FROM "D:\Raul\items.del" of del xml from "D:\Raul" INSERT
INTO ITEMS (ID, BRANDNAME, ITEMNAME, SKU, SRP, COMMENTS)
;
```

Zauważmy, że skrypt ten, jak i pliki z nim powiązane możemy znaleźć w archiwum **expressc\_book\_quicklabs.zip**, które jest dołączone do tej książki. Przeanalizujmy skrypt linia po linii:

1. Usuwanie bazy "mydb". Jest to standardowa czynność przy czyszczeniu. Jeżeli baza "mydb" jeszcze nie istnieje, otrzymamy błąd.
2. Tworzymy bazę danych "mydb" ze stroną kodową UTF-8. Baza UNICODE jest wymagana przy technologii pureXML, dlatego krok ten też jest wymagany.
3. Podłączamy się do nowoutworzonej bazy "mydb". Jest to niezbędne do utworzenia obiektów w tej bazie.
4. Tworzymy tabelę "items". Zauważmy, że ostatnia kolumna w tabeli (kolumna "comments") jest typu XML (nowy wbudowany typ danych).
5. Tworzymy tabelę "clients". Zauważmy, że ostatnia kolumna w tabeli (kolumna "contact") również jest kolumną typu XML.
6. Przy użyciu polecenia SQL INSERT, możemy dodać dokument XML do kolumny XML. W poleceniu INSERT przekazujemy dokument XML jako ciąg znaków zawarty w cudzysłowie.
7. Przy użyciu komendy IMPORT możemy dodać albo zaimportować wiele dokumentów XML. W (7) importujemy dane z pliku clients.del (plik separowany ASCII), dodatkowo wskazujemy lokalizację dokumentu XML, do którego odnosi się clients.del (na przykład do D:\Raul).

Przyjrzymy się bliżej plikowi clients.del, ale najpierw obejrzymy zawartość katalogu D:\Raul (Rysunek 16.10).

Name	Size	Type
Client3227.xml	1 KB	XML Docum
Client4309.xml	1 KB	XML Docum
Client5681.xml	1 KB	XML Docum
Client8877.xml	1 KB	XML Docum
Client9077.xml	1 KB	XML Docum
Client9177.xml	1 KB	XML Docum
ClientInfo.xsd	2 KB	XML Schem
clients.del	1 KB	DEL File
Comment3926.xml	1 KB	XML Docum
Comment4023.xml	1 KB	XML Docum
Comment4272.xml	1 KB	XML Docum
items.del	1 KB	DEL File

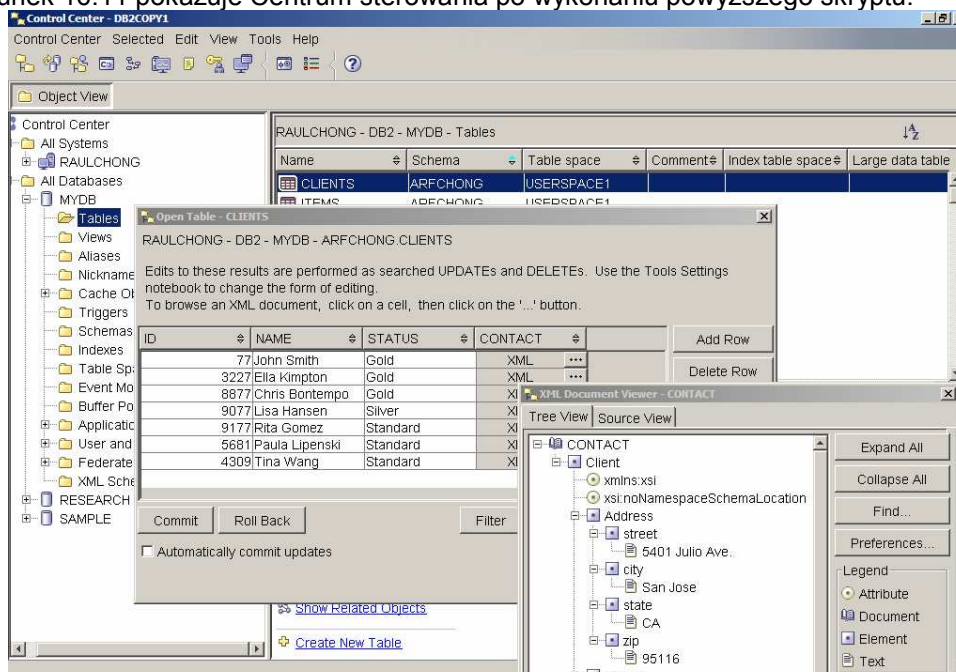
Rysunek 16.10 - Zawartość katalogu D:\Raul

Zawartość pliku clients.del:  
clients.del

```
3227,Ella Kimpton,Gold,<XDS FIL='Client3227.xml' />,
8877,Chris Bontempo,Gold,<XDS FIL='Client8877.xml' />,
9077,Lisa Hansen,Silver,<XDS FIL='Client9077.xml' />,
9177,Rita Gomez,Standard,<XDS FIL='Client9177.xml' />,
5681,Paula Lipenski,Standard,<XDS FIL='Client5681.xml' />,
4309,Tina Wang,Standard,<XDS FIL='Client4309.xml' />
```

W pliku clients.del, "XDS FIL=" służy do wskazania dokumentu XML.

Rysunek 16.11 pokazuje Centrum sterowania po wykonaniu powyższego skryptu.



Rysunek 16.11 – Centrum Sterowania po wykonaniu table\_creation.txt

Zauważmy, że na Rysunku tym przedstawiona została zawartość tabeli CLIENTS. Ostatnia kolumna „Contact” jest kolumną XML. Kiedy klikniemy na przycisk z trzema kropkami, wyświetli się nowe okno z zawartością pliku XML. Jest to pokazane w prawym dolnym rogu Rysunku 16.11.

### 16.3.5 Zapytania na danych XML

Istnieją dwa rodzaje zapytań XML w DB2:

- przy użyciu SQL z rozszerzeniami XML (SQL/XML)
- przy użyciu XQuery

W obu przypadkach DB2 trzyma się międzynarodowych standardów XML.

#### Zapytania na danych XML przy użyciu SQL/XML

Używanie zwykłego SQL umożliwia pracę z wierszami i kolumnami. Zapytanie SQL może być wykorzystane do pracy z pełnymi dokumentami XML, nie pozwoli nam więc na dostęp jedynie do fragmentu dokumentu. W takich przypadkach, należy użyć rozszerzenia SQL: SQL/XML.

Tabela 16.5 opisuje niektóre funkcje SQL/XML dostępne w standardzie z SQL 2006:

Nazwa funkcji	Opis
XMLPARSE	Analizuje ciąg znaków lub obiekt binarny i zwraca dokument XML.
XMLSERIALIZE	Tłumaczy dokument XML na ciąg znaków lub obiekt binarny.
XMLVALIDATE	Waliduje dokument XML według schematu i określa jego typ.
XMLEXISTS	Określa, czy polecenie XQuery zwraca wyniki (np. jeden lub więcej elementów).
XMLQUERY	Wykonuje polecenie XQuery i zwraca wynik.
XMLTABLE	Wykonuje polecenie XQuery, zwraca wynik w postaci tabeli relacyjnej (jeżeli jest to możliwe).
XMLCAST	Rzuca z lub na dokument XML.

**Tabela 16.5 – Funkcje SQL/XML**

Poniższe przykłady możemy przetestować na bazie „mydb”, którą stworzyliśmy wcześniej.

#### Przykład 1

Jest to przykład typowego problemu zapytania. Wyobraźmy sobie, że potrzebujemy znaleźć nazwiska wszystkich klientów, którzy mają dane kody pocztowe. Tabela „clients” przechowująca adresy klientów, zawiera kody pocztowe w kolumnie XML. Używając XMLEXISTS, możemy przeszukać kolumnę XML, aby odnaleźć kody pocztowe i w ten sposób ograniczyć zestaw wyników.

```
SELECT name FROM clients
WHERE xmlexists(
    '$c/Client/Address[zip="95116"]'
    passing clients.contact as "c"
)
```



Pierwsza linia jest klauzulą SQL wskazującą, że chcemy pobrać informacje z kolumny „name” z tabeli „clients”.

Klauzula WHERE powołuje się na funkcję XMLEXISTS, określającą wyrażenie XPath, które każe DB2 przejść do elementu „zip” i sprawdzić czy wartość wynosi 95116.

Klauzula „\$c/Client/Address” wskazuje ścieżkę wewnątrz hierarchii dokumentu XML, gdzie DB2 może odnaleźć element „zip”. Znak dolara (\$) jest używany do określenia zmiennej, dlatego „c” jest zmienną. Zmienna jest wówczas definiowana tą linią: `passing clients.contact as „c”`. Tutaj, „clients” jest nazwą tabeli a „contact” jest nazwą kolumny z typem danych XML. Innymi słowy, parsujemy dokument XML do zmiennej „c”.

DB2 kontroluje dane zawarte w kolumnie „contact” w XML, przechodząc z korzenia „Client” w dół do węzła „Address”, a następnie do węzła „zip”, i w końcu określa, czy klient ma przypisany dany kod pocztowy. Funkcja XMLEXISTS określa „prawdę” i DB2 zwraca nazwisko klienta powiązanego z tym wierszem.

W DB2 9.5, powyższe zapytanie może być uproszczone, tak jak poniżej:

```
SELECT name FROM clients
  WHERE xmlexists(
    '$CONTACT/Client/Address[zip="95116"]'
  )
```

Zmienna o tej samej nazwie co kolumna w pliku XML jest tworzona automatycznie przez DB2. W powyższym przykładzie będzie to zmienna CONTACT. Jej nazwa pasuje do nazwy kolumny CONTACT w pliku XML.

#### Przykład 2

Rozważmy, jak rozwiązać problem tworzenia raportu wyświetlającego adresy e-mail klientów, którzy mają status „Gold”. Można uruchomić następujące zapytanie:

```
SELECT xmlquery('$c/Client/email' passing contact as "c")
  FROM clients
  WHERE status = 'Gold'
```

Pierwsza linia wskazuje, że chcemy zwrócić adresy e-mail, które są elementem dokumentu XML (a nie kolumną relacyjną). Tak jak w poprzednim przykładzie, „\$c” jest zmienną, która zawiera dokument XML. W tym przykładzie wykorzystujemy funkcję XMLQUERY, która może być użyta po klauzuli SELECT, podczas gdy funkcja XMLEXISTS może być użyta po klauzuli WHERE.

#### Przykład 3

Mogą być sytuacje, w których będziemy chcieli prezentować dane XML jako tabele. Jest to możliwe z funkcją XMLTABLE, tak jak jest to pokazane poniżej.

```
SELECT t.comment#, i.itemname, t.customerID, Message
  FROM items i,
  xmltable('$c/Comments/Comment' passing i.comments as "c"
    columns Comment# integer path 'CommentID',
```

```
CustomerID integer path 'CustomerID',
Message varchar(100) path 'Message') AS t
```

Pierwsza linia określa kolumny, które będą zawarte w zbiorze wyników. Kolumny z prefiksem zmiennej „t” są oparte na wartości elementu XML.

Druga linia wywołuje funkcję XMLTABLE, aby określić kolumnę XML, zawierającą dane docelowe („i comments”), oraz ścieżki w kolumnach dokumentów XML, gdzie znajduje się interesujący nas element.

Klauzula „columns”, obejmująca linie od 4 do 6, określa konkretne elementy XML, które zostaną przyporządkowane wyjściowym kolumnom SQL w zestawie wyników określonym w linii 1. Część tego mapowania obejmuje określenie typów danych XML, do którego elementy zostaną przekonwertowane. W tym przykładzie wszystkie dane XML są konwertowane na tradycyjne typy danych SQL.

#### Przykład 4

Zbadajmy teraz prosty przykład, w którym wykorzystamy wyrażenie XQuery FLWOR wewnątrz funkcji SQL/XML XMLQUERY.

```
SELECT name, xmlquery(
  'for $e in $c/Client/email[1] return $e'
  passing contact as "c"
)
FROM clients
WHERE status = 'Gold'
```

Pierwsza linia określa, że nazwiska klientów i wyjście funkcji XMLQUERY będzie zawarte w zestawie wyników. Druga linia wskazuje, że zostanie zwrócony pierwszy podelement „email”, elementu „Client”. Trzecia linia określa źródło danych XML (kolumnę „contact”). Czwarta linia mówi nam, że kolumny są brane z tabeli „clients”, a piąta linia wskazuje, że interesują nas jedynie klienci ze statusem „Gold”.

#### Przykład 5

Przykład ten pokazuje funkcję XMLQUERY, która używa wyrażenia XQuery FLWOR. Zauważmy, że tym razem zwracamy nie tylko kod XML, ale również HTML.

```
SELECT xmlquery('for $e in $c/Client/email[1]/text()
  return <p>{$e}</p>'
  passing contact as "c")
FROM clients
WHERE status = 'Gold'
```

Klauzula „return” wyrażenia XQuery pozwala na przekształcanie wyjścia XML, tak jak to potrzebne. Używając funkcji text() w pierwszej linii wskazujemy, że interesuje nas jedynie reprezentacja tekstowa pierwszego adresu e-mailowego kwalifikujących się klientów. Druga linia określa, że informacja ta jest otoczona tagami HTML.

#### Przykład 6

Poniższy przykład wykorzystuje funkcję XMLEMENT do tworzenia serii elementów, z których każdy zawiera podelement z ID, nazwą marki, SKU, odpowiadającą wartościom uzyskanym z tabeli „items”. Zasadniczo, można użyć funkcji XMLEMENT wtedy, kiedy chcemy skonwertować relacyjne dane do postaci kodu XML.

```
SELECT
  xmlelement (name "item", itemname),
  xmlelement (name "id", id),
  xmlelement (name "brand", brandname),
  xmlelement (name "sku", sku)
FROM items
WHERE srp < 100
```

Rezultatem powyższego wyrażenia może być na przykład:

```
<item>
  <id>4272</id>
  <brand>Classy</brand>
  <sku>981140</sku>
</item>
...
<item>
  <id>1193</id>
  <brand>Natural</brand>
  <sku>557813</sku>
</item>
```

## Odpytywanie danych XML za pomocą XQuery

W poprzedniej sekcji zobaczyliśmy, jak odpytywać dane XML za pomocą zapytań SQL rozszerzonymi rozszerzeniami XML. SQL był zawsze podstawowym językiem zapytań, a XPath został osadzony wewnątrz SQL. W tej sekcji omówimy, w jaki sposób odpytywać dane za pomocą SQL w XQuery. Tym razem to XQuery będzie podstawową metodą zapytań, a w niektórych przypadkach będziemy używać SQL osadzonego wewnątrz XQuery (za pomocą funkcji "db2-fn:sqlquery"). Przy użyciu XQuery, będziemy wywoływać kilka funkcji, a także używać wyrażeń FLWOR.

### Przykład 1

Proste zapytanie XQuery zwracające kontakt do klienta:

```
xquery db2-fn:xmlcolumn('CLIENTS.CONTACT')
```

Wyrażenia XQuery zawsze poprzedzamy prefiksem „xquery”, aby DB2 wiedziało, że należy użyć parsera XQuery. Jeżeli tak nie postąpimy, DB2 będzie próbowało wykonać podane polecenie jako wyrażenie SQL. Funkcja **db2-fn:xmlcolumn** to funkcja, która zwraca dokumenty XML z kolumny podanej w parametrze. Odpowiada to następującemu wyrażeniu SQL:

```
SELECT contact FROM clients
```

**Przykład 2**

W tym przykładzie używamy wyrażenia FLWOR, aby odczytać numery faksu klientów.

```
xquery
  for $y in db2-fn:xmlcolumn('CLIENTS.CONTACT')/Client/fax
  return $y
```

Pierwsza linia uruchamia parser XQuery. W drugim wierszu instruujemy DB2 do iterowania po numerze faksu w podelementach zawartych w kolumnie CLIENTS.CONTACT. Każdy element *fax* przypisywany jest do zmiennej \$y. Trzecia linia oznacza, że w każdej iteracji zwracana jest wartość zmiennej "\$y".

Wynik tego zapytania jest zbliżony do poniższego (może on zawierać domyślnie „namespace”, ale nie przedstawiamy tego poniżej, gdyż wynik byłby trudniejszy do odczytania, ponieważ mógłby obejmować kilka linii):

```
<fax>4081112222</fax>
<fax>5559998888</fax>
```

**Przykład 3**

Ten przykład odpytuje dane XML i zwraca wynik zapytania jako HTML

```
xquery
  <ul> {
    for $y in db2-fn:xmlcolumn('CLIENTS.CONTACT')/Client/Address
    order by $y/zip
    return <li>{$y}</li>
  }
  </ul>
```

Przykładowy kod HTML wygląda następująco:

```
<ul>
<li>
<address>
  <street>9407 Los Gatos Blvd.</street>
  <city>Los Gatos</city>
  <state>ca</state>
  <zip>95302</zip>
</address>
</li>
<address>
<street>4209 El Camino Real</street>
  <city>Mountain View</city>
  <state>CA</state>
  <zip>95302</zip>
</address>
</li>
...
</ul>
```

**Przykład 4**

Poniższy przykład pokazuje, w jaki sposób należy wbudować kod SQL w kod XQuery za pomocą funkcji `db2-fn:sqlquery`. Funkcja `db2-fn:sqlquery` wykonuje zapytanie SQL i zwraca tylko wybrane dane XML. Zapytania SQL przekazane do `db2-fn:sqlquery` muszą zwracać tylko dane XML. Te dane XML mogą być następnie przetwarzane przez XQuery.

```
xquery
  for $y in
    db2-fn:sqlquery(
      'select comments from items where srp > 100'
    )/Comments/Comment
  where $y/ResponseRequested='Yes'
  return (
    <action>
      {$y/ProductID
       $y/CustomerID
       $y/Message}
    </action>
  )
```

W przykładzie tym, zapytanie SQL filtruje wiersze w oparciu o warunek, że kolumna "srp" ma wartość większą niż 100. Z tych przefiltrowanych wierszy wybierana jest kolumna „comments”, która jest kolumną XML. Następnie stosowany jest XQuery (lub XPath), aby przejść do podelementów.

**Ważne:** W DB2 nie jest ważna wielkość liter. DB2 traktuje wszystkie nazwy kolumn tabeli jako złożone z wielkich liter, podczas gdy w XQuery wielkość liter jest uwzględniana. Powyższe funkcje XQuery są funkcjami interfejsu, tak więc wszystkie nazwy tabel i kolumn powinny być przekazywane do tych funkcji jako złożone z wielkich liter. Przekazywanie nazw obiektów w małych literach może skutkować błędem niezidentyfikowanego obiektu.

**16.3.6 Złączenia w SQL/XML**

W sekcji tej opisano, jak wykonywać operację złączenia (ang. JOIN) między dwoma kolumnami XML z różnych tabel, lub między jedną kolumną XML i jedną kolumną relacyjną. Zakładamy, że tabele zostały stworzone za pomocą poniższego zapytania:

```
CREATE TABLE dept (unitID CHAR(8), deptdoc XML)

CREATE TABLE unit (unitID CHAR(8) primary key not null,
                   name CHAR(20),
                   manager VARCHAR(20),
                   ...
                   )
```

Możemy wykonać operację złączenia jedną z dwóch poniższych metod:

**Metoda 1:**

```
SELECT u.unitID
  FROM dept d, unit u
 WHERE XMLEXISTS (
   '$e//employee[name = $m]'
   passing d.deptdoc as "e", u.manager as "m")
```

W linii 3 powyższego zapytania widzimy, że operacja łączenia jest wykonywana pomiędzy elementem „name”, który jest podelementem „deptdoc” kolumny XML w tabeli „dept”, a relacyjną kolumną „manager” tabeli „unit”.

**Metoda 2:**

```
SELECT u.unitID
  FROM dept d, unit u
 WHERE u.manager = XMLCAST(
   XMLQUERY('$e//employee/name '
   passing d.deptdoc as "e")
   AS char(20))
```

W tej alternatywnej metodzie, kolumna relacyjna jest po lewej stronie operacji łączenia. Kiedy kolumna relacyjna jest po lewej stronie łączenia, zamiast indeksu XML może zostać użyty indeks relacyjny.

**16.3.7 Złączenia w XQuery**

Zakładamy, że tabele zostały utworzone w następujący sposób:

```
CREATE TABLE dept(unitID CHAR(8), deptdoc XML)
CREATE TABLE project(projectDoc XML)
```

Jeżeli użyjemy zapytania SQL/XML, będzie ono wyglądało następująco:

```
SELECT XMLQUERY (
 '$d/dept/employee' passing d.deptdoc as "d")
  FROM dept d, project p
 WHERE XMLEXISTS (
   '$e/dept[@deptID=$p/project/deptID]'
   passing d.deptdoc as "e", p.projectDoc as "p")
```

Odpowiadające mu zapytanie XQuery będzie wyglądało w ten sposób:

```
xquery
  for $dept in db2-fn:xmlcolumn("DEPT.DEPTDOC")/dept
  for $proj in db2-fn:xmlcolumn("PROJECT.PROJECTDOC")/project
  where $dept/@deptID = $proj/deptID
  return $dept/employee
```

Druga metoda jest łatwiejsza do zinterpretowania - zmienna „\$dept” zawiera dokument XML na kolumnie XML „deptdoc” w tabeli „dept”. Zmienna „\$proj” zawiera dokument XML

projectdoc kolumny „projectdoc” w tabeli „projekt”. Następnie w linii 4 wykonywane jest łączenie pomiędzy atrybutem pierwszego dokumentu XML i elementem drugiego dokumentu XML.

### 16.3.8 Operacje aktualizacji i usuwania

Operacje aktualizacji (UPDATE) i usuwania (DELETE) w dokumentach XML mogą być wykonane na dwa sposoby:

- ▶ Użycie operacji SQL: UPDATE, DELETE
- ▶ Użycie wyrażenia TRANSFORM

Korzystając z pierwszego sposobu, czyli SQL UPDATE i DELETE, aktualizacja lub usunięcie dotyczy całego dokumentu. Oznacza to, że cały dokument XML zostaje usunięty lub zaktualizowany. Jeżeli w poniższym przykładzie chcielibyśmy zmodyfikować element <state>, to cały dokument XML zostanie zastąpiony.

```
UPDATE clients SET contact=(
  xmlparse(document
    '<Client>
      <address>
        <street>5401 Julio ave.</street>
        <city>San Jose</city>
        <state>CA</state>
        <zip>95116</zip>
      </address>
      <phone>
        <work>4084633000</work>
        <home>4081111111</home>
        <cell>4082222222</cell>
      </phone>
      <fax>4087776666</fax>
      <email>newemail@someplace.com</email>
    </Client>')
  )
WHERE id = 3227
```

Przy użyciu drugiego sposobu, można wykonywać częściową aktualizację za pomocą wyrażenia TRANSFORM, które jest o wiele bardziej efektywne. Dzięki temu można zastąpić, wstawić, usunąć lub zmienić nazwy węzłów w dokumencie XML. Możemy także zmienić wartość węzła bez zastępowania całego węzła. Zazwyczaj przy zmianach wartości lub nazwy węzła zamieniany jest cały węzeł. Jest to nowość w DB2 9.5.

Wyrażenie TRANSFORM jest częścią języka XQuery. Możemy go używać wszędzie tam, gdzie normalnie używamy XQuery, na przykład w wyrażeniach FLWOR lub w funkcjach XMLQUERY w zapytaniach SQL/XML. Najczęściej wykorzystywane jest w SQL UPDATE, aby zmodyfikować dokument XML w kolumnie XML.

Poniżej znajduje się składnia wyrażenia TRANSFORM:

---

```
>>-transform--| copy clause |--| modify clause |--| return clause
|-><
```

```
copy clause
```

```
      .-,'-----'
      v |
|--copy---$VariableName--:=---CopySourceExpression+-----|
```

```
modify clause
```

```
|--modify--ModifyExpression-----|
```

```
return clause
```

```
|--return--ReturnExpression-----|
```

---

Wyrażenie `copy` jest używane, aby przypisać do zmiennej dokument XML, z którym chcemy pracować. W wyrażeniu `modify`, możemy wykonywać wyrażenia `insert`, `delete`, `rename` lub `replace`. Wyrażenia te pozwalają nam na dokonywanie modyfikacji w naszych dokumentach XML. Na przykład, jeżeli chcemy dodać nowy węzeł do dokumentu XML, moglibyśmy użyć polecenia `insert`, natomiast aby usunąć węzły z dokumentu XML moglibyśmy użyć polecenia `delete`. Podobnie, aby zmienić nazwę elementu lub atrybutu w dokumencie XML moglibyśmy użyć wyrażenia `rename`, a żeby zastąpić element nowym elementem, lub ciągiem kilku nowych elementów moglibyśmy użyć wyrażenia `replace`. Wyrażenie `replace` może być użyte tylko do zmiany elementu lub atrybutu. Wyrażenie `return` zwraca wynik transformacji.

Poniżej znajduje się przykład operacji UPDATE z wykorzystaniem wyrażenia TRANSFORM.

---

```
(1)-- UPDATE customers
(2)-- SET contactinfo = xmlquery( 'declare default element namespace
(3)--                               "http://posample.org";
(4)--   transform
(5)--   copy $newinfo := $c
(6)--       modify do insert <email2>my2email.gm.com</email2>
(7)--           as last into $newinfo/customerinfo
(8)--   return $newinfo' passing contactinfo as "c")
(9)-- WHERE id = 100
```

---

W powyższym przykładzie, wiersze (1), (2) i (9) są częścią składni SQL UPDATE. W linii (2) jest wywoływana funkcja XMLQUERY, która wywołuje wyrażenie transformacji w linii (4). Blok wyrażenia transformacji znajduje się między liniami (4) i (8), jest używany, aby wstawić nowy węzeł do dokumentu XML, zawierający element `email2`. Należy pamiętać, że uaktualnianie elementów dokumentu XML za pomocą widoku nie jest obsługiwane.



Usuwanie całego dokumentu XML z tabeli jest tak proste, jak użycie wyrażenia SELECT w SQL/XML. Użyj polecenia SQL DELETE i określ dodatkowe ograniczenia w klauzuli WHERE, jeżeli będzie to wymagane.

### 16.3.9 Indeksowanie XML

W dokumencie XML indeksy mogą być stworzone dla elementów, atrybutów lub wartości (węzły tekstowe). Poniżej znajduje się kilka przykładów. Zakładamy, że tabela poniżej została utworzona w następujący sposób:

```
CREATE TABLE customer(info XML)
```

Zakładamy także, że poniższy dokument XML znajduje się w tej tabeli:

```
<customerinfo Cid="1004">
  <name>Matt Foreman</name>
  <addr country="Canada">
    <street>1596 Baseline</street>
    <city>Toronto</city>
    <state>Ontario</state>
    <pcode>M3Z-5H9</pcode>
  </addr>
  <phone type="work">905-555-4789</phone>
  <phone type="home">416-555-3376</phone>
  <assistant>
    <name>Peter Smith</name>
    <phone type="home">416-555-3426</phone>
  </assistant>
</customerinfo>
```

1) Wyrażenia tworzące indeks dla atrybutu „Cid”:

```
CREATE UNIQUE INDEX idx1 ON customer(info)
  GENERATE KEY USING
  xmlpattern '/customerinfo/@Cid'
  AS sql DOUBLE
```

2) Wyrażenie, które tworzy indeks dla elementu „name”:

```
CREATE INDEX idx2 ON customer(info)
  GENERATE KEY USING
  xmlpattern '/customerinfo/name'
  AS sql VARCHAR(40)
```

3) Wyrażenie, które tworzy indeks na wszystkich elementach „name”:

```
CREATE INDEX idx3 ON customer(info)
```

```
GENERATE KEY USING  
xmlpattern '//name'  
AS sql VARCHAR(40);
```

4) Wyrażenie, które tworzy indeks na wszystkich węzłach tekstowych (dla wszystkich wartości). Nie jest to zalecane, ponieważ operacja update, delete lub insert mogłaby być zbyt kosztowna i indeks mógłby być zbyt duży.

```
CREATE INDEX idx4 ON customer(info)  
GENERATE KEY USING  
xmlpattern '//text()'  
AS sql VARCHAR(40);
```

## Laboratorium #14 - SQL/XML i XQuery

### Cel

Zobaczyłeś wiele przykładów wyrażeń SQL/XML i XQuery i znasz już Edytor komend DB2 oraz IBM Data Studio. W tym Laboratorium przetestujesz swoją wiedzę z SQL/XML i XQuery oraz zdobędziesz doświadczenia związane z używaniem tych narzędzi. Będziemy używali bazy „mydb” stworzonej za pomocą skryptu `table_creation.txt`, który opisywaliśmy wcześniej w tym rozdziale.

### Kroki

1. Stwórz bazę „mydb” i załaduj dane XML, tak jak to zostało omówione w tym rozdziale.
2. Przy użyciu poleceń Edytora komend DB2 lub IBM Data Studio:
  - a) Odczytaj wszystkie komentarze z dokumentów XML z tabeli `ITEMS` na dwa sposoby, ale tylko przy wykorzystaniu XQuery.
  - b) Dlaczego przy użyciu poniższego zapytania SQL wynik nie byłby identyczny?  

```
SELECT comments FROM items
```
  - c) Zwróć ID i BRANDNAME dla wierszy, których element Response-Requested ma wartość „No”.

---

### ROZWIĄZANIA:

2a)

```
xquery db2-fn:xmlcolumn('ITEMS.COMMENTS')
xquery db2-fn:sqlquery("select comments from
items")
```

2b)

Wynik jest różny ponieważ zapytanie SQL zwraca wartość NULL, jeżeli element nie ma wartości, a XQuery nie zwraca nic.

2c)

```
SELECT id, brandname FROM items WHERE
XMLEXISTS('$c/Comments/Comment[ResponseRequested="No"]'
passing ITEMS.COMMENTS as "c
```



---

# 17

## Rozdział 17 – Programowanie aplikacji Java, PHP i Ruby

W rozdziale tym omówimy podstawy programowania aplikacji w Javie, PHP i Ruby on Rails, używając do tego serwera DB2. Celem tego rozdziału nie jest nauka tych języków, ale pokazanie, jak one współgrają z DB2.

**Ważne:**

Aby uzyskać więcej informacji obejrzyj poniższe wideo:

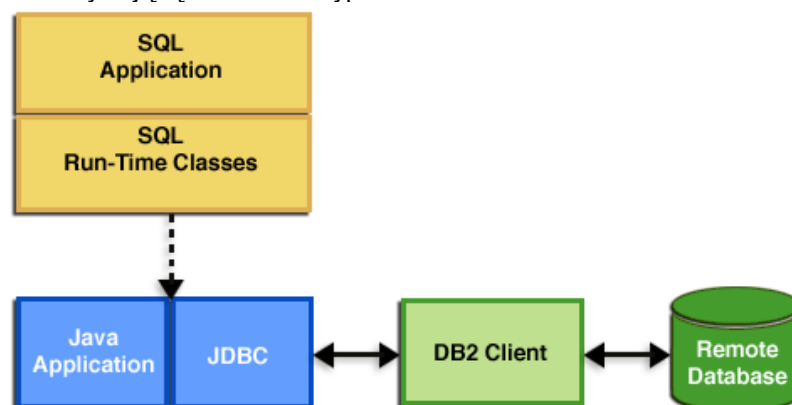
<http://www.channeldb2.com/video/video/show?id=807741:Video:4402>

### 17.1 Programowanie aplikacji w Javie

Sterownik IBM DB2 dla JDBC (znany również jako sterownik JCC) został zoptymalizowany dla wszystkich serwerów DB2 na wszystkich platformach. Plik db2jcc.jar (com.ibm.db2.jcc) zawiera sterowniki typu 2 oraz 4. Plik db2jcc.jar jest częścią każdego klienta DB2, lub może być pobrany ze strony DB2 Express-C (ibm.com/db2/express, plik: sterownik IBM DB2 dla JDBC oraz SQLJ).

#### 17.1.1 Sterownik JDBC typu 2

Sterownik JDBC typu 2 wymaga zainstalowanego klienta DB2. Rysunek 17.1 ilustruje aplikację JDBC używającą sterownika typu 2.



Rysunek 17.1 – Sterownik JDBC typu 2

Rysunek 17.2 przedstawia listing, na którym pokazano, jak ustanowić połączenie używając sterownika JDBC typu 2. Zauważ, że URL nie zawiera ani nazwy hosta, ani numeru portu. Te informacje są pobierane z klienta DB2.

```

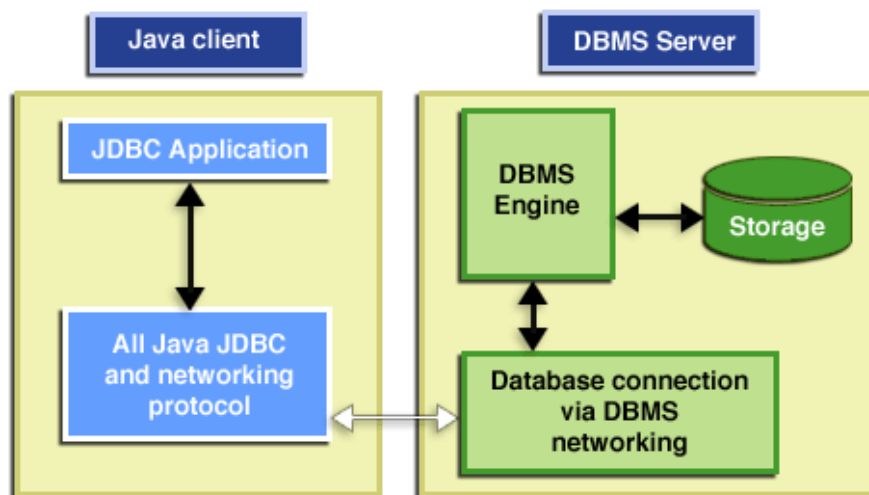
...
public static final String DB_URL = "jdbc:db2:sample";
Properties connectProperties = new Properties();
connectProperties.put("user", "db2admin");
connectProperties.put("password", "ibmdb2");
Connection connection = null
try
{
    Class.forName("com.ibm.db2.jcc.DB2Driver").newInstance();
    connection = DriverManager.getConnection(url, connectProperties)
}
catch (Exception e)
throw e;
}
...

```

Rysunek 17.2 – Nawiązywanie połączenia przy użyciu sterownika JDBC typu 2

#### 17.1.2 Sterownik JDBC typu 4

Sterownik JDBC typu 4 nie wymaga działającego klienta DB2 do połączenia z serwerem DB2. Rysunek 17.3 ilustruje aplikację JDBC używającą sterownika typu 4.



Rysunek 17.3 – Sterownik JDBC typu 4

Rysunek 17.4 przedstawia listing pokazujący, jak nawiązać połączenie używając sterownika typu 4. Zauważ, że URL wymaga podania nazwy hosta oraz numeru portu.

```
...
public static final String DB_URL = "jdbc:db2://server1:50000/sample";
Properties connectProperties = new Properties();
connectProperties.put("user", "db2admin");
connectProperties.put("password", "ibmdb2");
Connection connection = null
try
{
    Class.forName("com.ibm.db2.jcc.DB2Driver").newInstance();
    connection = DriverManager.getConnection(url, connectProperties)
}
catch (Exception e)
    throw e;
}
...
```

Rysunek 17.4 – Nawiązywanie połączenia używając sterownika typu 4.

## 17.2 Programowanie aplikacji w PHP

PHP (PHP Hypertext Preprocessor) to otwarty (darmowy), niezależny od platformy język skryptowy przeznaczony do programowania aplikacji internetowych. Jest jednym z najszerzej stosowanych języków w dzisiejszym świecie Internetu. Popularność PHP wynika z jego cech:

- gwałtowne, interaktywne cykle rozwoju z niską krzywą nauczania
- solidny, skalowany oraz wysoce wydajny
- stabilny i bezpieczny
- alternatywa dla J2EE™ oraz .NET
- łatwa integralność z heterogenicznymi środowiskami/systemami
- jakość potwierdzona poprzez szerokie użycie
- żywa i dobrze zorganizowana społeczność

PHP jest częścią LAMP (Linux, Serwer HTTP Apache, MySQL, PHP/Perl/Python). Jest to otwarta technologia internetowa, często oferowana w ramach usług za rozsądną cenę.

### 17.2.1 Opcje połączenia DB2 dla PHP

IBM wspiera dostęp do baz danych DB2 z poziomu aplikacji PHP poprzez dwa rozszerzenia.

#### **ibm\_db2:**

Rozszerzenie `ibm_db2` to interfejs programistyczny do tworzenia, czytania, aktualizowania i zapisywania operacji bazodanowych. Zapewnia on również szeroki dostęp do meta danych bazy. Może zostać skompilowany do pracy z PHP 4 lub PHP 5. Rozszerzenie jest dostępne z repozytorium PECL objętym licencją Apache 2.0. Zostało ono rozwinięte i jest wspierane przez IBM. W pełni wspiera ono procedury składowane oraz obiekty typu LOB, ponadto optymalizacja pod względem DB2 gwarantuje jego szybkość.

#### **PDO\_ODBC**

PDO\_ODBC jest sterownikiem dla rozszerzenia obiektów danych PHP (ang. PHP Data Objects - PDO). Oferuje dostęp do DB2 poprzez zorientowany obiektowo interfejs bazy danych wprowadzony w PHP 5.1. Może zostać skompilowany bezpośrednio z bibliotek DB2. Dostarcza standardowy interfejs dostępu do danych dla PHP. Jest szybki, lekki oraz zorientowany obiektowo. Rozszerzenie PDO\_ODBC używa bibliotek DB2 do dostępu natywnego oraz jest wbudowane w PHP 5.1. Więcej informacji można uzyskać na poniższych stronach:

- <http://pecl.php.net/package/pdo>
- [http://pecl.php.net/package/PDO\\_ODBC](http://pecl.php.net/package/PDO_ODBC)

## Podłączanie do nieskatalogowanej bazy danych DB2

Listing 17.1 pokazuje jak można połączyć się z bazą danych DB2 używając jednego z dwóch rozszerzeń opisanych powyżej.

```
$host = 'localhost';
$port = 50000;
$DSN = "DRIVER={IBM DB2 ODBC DRIVER}; PORT=$port;
        HOSTNAME=$host; DATABASE=$database; PROTOCOL=TCPIP;
        USER=$user; PWD=$password";

-- If using the ibm_db2 extension --
$uconn = db2_connect($DSN, null, null);

-- If using the PDO_ODBC extension --
try {
    $uconn = new PDO("odbc:$DSN", null, null);
}
catch (PDOException $e) { print $e->errmsg(); }
```

**Listing 17.1 – Podłączanie do nieskatalogowanej bazy danych DB2**

Listing 17.2 przedstawia przykład prostej aplikacji PHP używającej rozszerzenia `ibm_db2`.

```
<?php
$sql = "SELECT name, breed FROM ANIMALS WHERE weight < ?";
$conn = db2_connect($database, $user, $password);
$stmt = db2_prepare($conn, $sql);
$res = db2_execute($stmt, array(10));
while ($row = db2_fetch_assoc($stmt)) {
    print "{$row['NAME']} is a {$row['BREED']}. \n";
}
?>
```

**Listing 17.2 – Prosta aplikacja PHP używająca rozszerzenia `ibm_db2`**

## Konfiguracja PHP dla `ibm_db2`

Systemy Linux bądź UNIX mogą wymagać następującej zmiany w pliku `php.ini`:

```
extension=ibm_db2.so
ibm_db2.instance_name=<instance name>
```



W systemie Windows należy zmodyfikować plik `php.ini` w następujący sposób:

```
extension=php_ibm_db2.dll
```

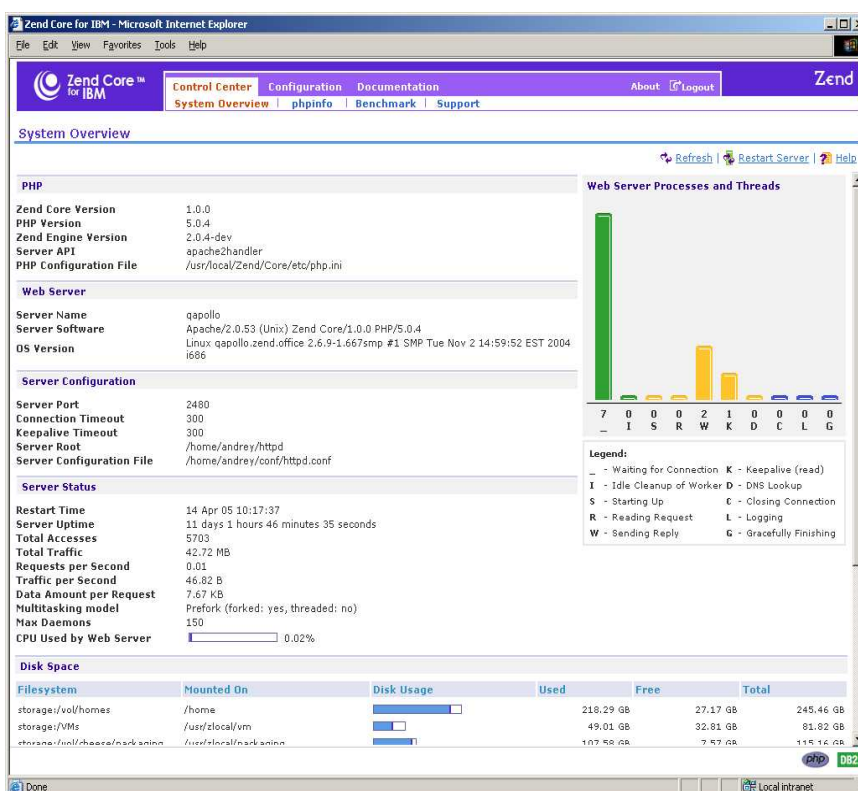
Alternatywnie możemy pobrać i zainstalować pakiet **Zend Core dla IBM**, tak jak opisano w następnej sekcji, i nie martwić się o problemy z konfiguracją.

### 17.2.2 Zend Core dla IBM

Zend Core to jednolite, gotowe od razu do użycia środowisko PHP, służące do rozwijania i produkcji biznesowych aplikacji internetowych. Zapewnia solidność, efektywność oraz uniwersalność potrzebną do uruchamiania aplikacji PHP. Może być pobrane za darmo ze strony:

<http://ibm.com/software/data/info/zendcore>

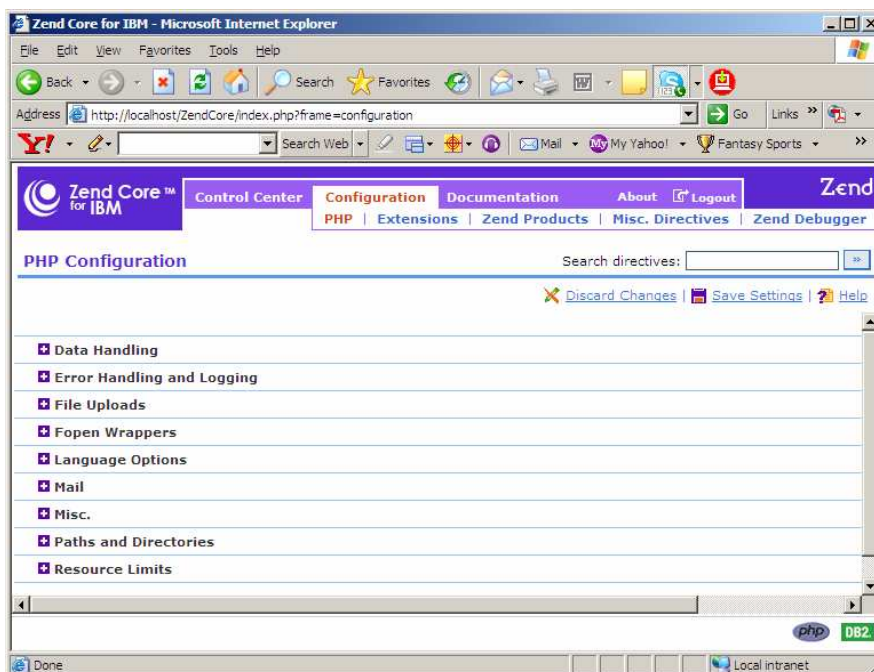
Zend Core dla IBM instaluje klientów DB2 oraz IDS, dodatkowy serwer HTTP Apache, PHP 5 oraz popularne rozszerzenia PHP m.in. `ibm_db2` oraz `PDO_INFORMIX`. Zend Core dla IBM może opcjonalnie zainstalować serwery DB2 Express-C oraz IBM Cloudspace™, pełen podręcznik użytkownika oraz przykładowe aplikacje DB2. Jest częścią łatwego w obsłudze i konfiguracji środowiska PHP, tak jak pokazano na Rysunkach 17.5, 17.6 i 17.7.



Rysunek 17.5 – Interfejs zarządzania i kontroli Zend Core



Rysunek 17.6 – Interfejs konfiguracji PHP Zend Core



Rysunek 17.7 - Interfejs konfiguracji PHP Zend Core

### 17.3 Rozwój aplikacji w Ruby on Rails

Ruby to zorientowany obiektowo, dynamiczny oraz międzyplatformowy język skryptowy. Ułatwia szybki rozwój oprogramowania i zawiera bogatą bibliotekę. Ruby to prosty i zorientowany na zabawę język programowania wymyślony przez Yukihiro Matsumoto („Matz”) w 1995 roku.

Rails to framework dla aplikacji internetowych opartych na bazach danych i napisanych w Ruby. Implementuje architekturę Model-Widok-Kontroler (ang. Model-View-Controller). Jest bardzo efektywny i łatwy w obsłudze. Rails jest jednym z najszybciej rozwijających się frameworków internetowych od roku 2004. Został stworzony przez Davida Heinemeiera Hanssona.

#### 17.3.1 Uruchomieniowy zestaw narzędzi dla DB2 on Rails

IBM dostrzega znaczenie Ruby on Rails w społeczności programistów. Dlatego też stworzył **uruchomieniowy zestaw narzędzi dla DB2 on Rails** (ang. Startup Toolkit for DB2 on Rails). Jest to zintegrowany instalator, który tworzy pełne środowisko programistyczne DB2 Ruby on Rails. Może być bezpłatnie pobrany i używany ze strony IBM AlphaWorks:

<http://www.alphaworks.ibm.com/tech/db2onrails>.

Uruchomieniowy zestaw narzędzi dla DB2 on Rails:

- ▶ zawiera zintegrowany instalator
- ▶ pomaga zainstalować i skonfigurować Ruby on Rails
- ▶ instaluje DB2 Express - C 9 wraz z narzędziami
- ▶ zawiera napisany przez IBM sterownik DB2 Ruby oraz adapter DB2 Rails
- ▶ zawiera różne demo i tutoriale

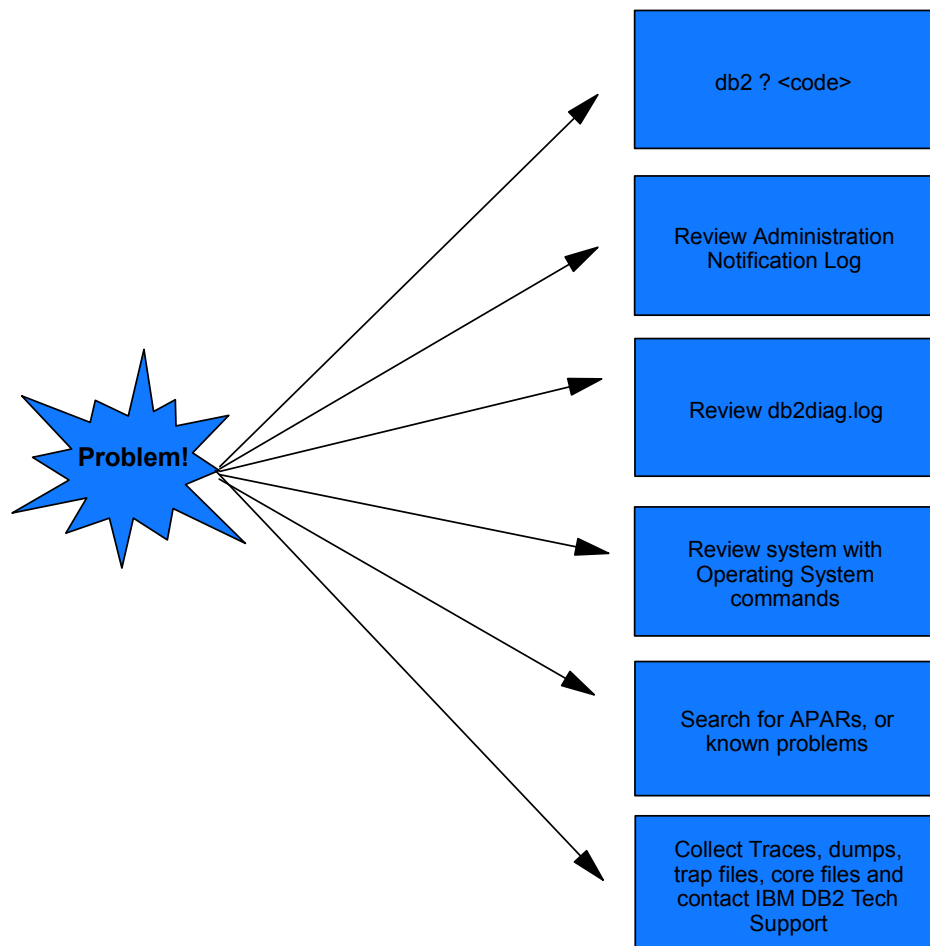


---

# A

## Dodatek A – Rozwiązywanie problemów

Dodatek ten omawia sposoby rozwiązywania problemów, które możemy napotkać podczas pracy z DB2. Rysunek A.1 pokazuje krótki przegląd działań, jakie należy podjąć, kiedy powstanie jakiś problem.



Rysunek A.1 – Rozwiązywanie problemów

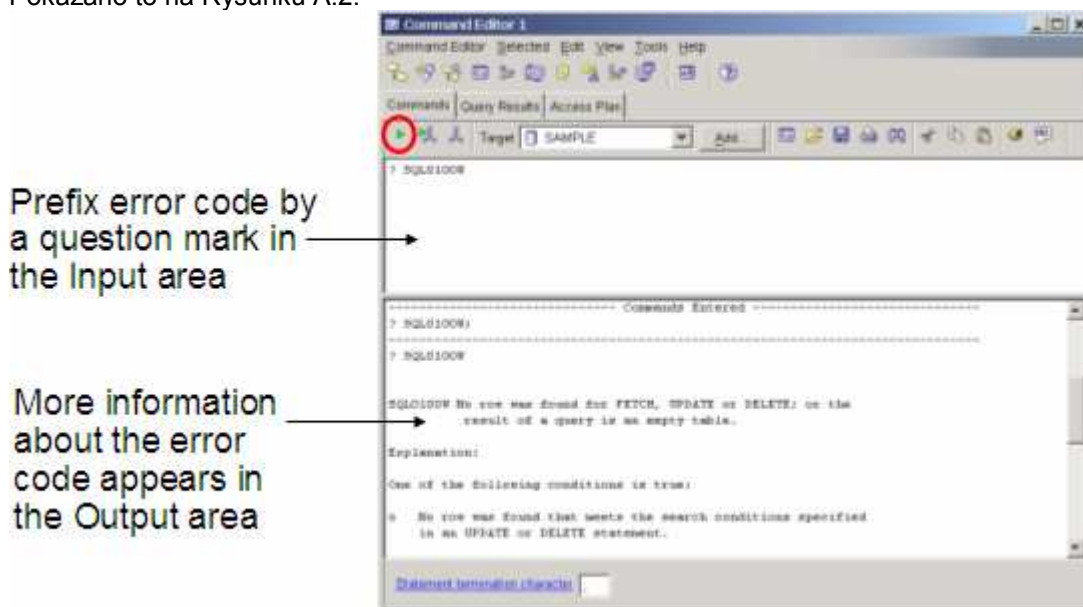
**Ważne:**

W celu uzyskania dodatkowych informacji dotyczących rozwiązywania problemów obejrzyj poniższe nagranie wideo:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4462>

**A.1 Uzyskiwanie informacji o kodach błędów**

Aby uzyskać więcej informacji o otrzymywanych kodach błędów, wprowadź kod błędu poprzedzony znakiem zapytania w oknie Edytora komend i naciśnij przycisk *Wykonaj*. Pokazano to na Rysunku A.2.



**Rysunek A.2 – Uzyskiwanie informacji o kodach błędów DB2**

Znak zapytania (?) wywołuje komendę pomocy. Poniżej znajduje się kilka przykładów, jak wywołać tę komendę, jeśli otrzymamy, na przykład, kod błędu „-104”. Wszystkie przykłady są równoważne.

```

db2 ? SQL0104N
db2 ? SQL104N
db2 ? SQL-0104
db2 ? SQL-104
db2 ? SQL-104N

```

## A.2 SQLCODE i SQLSTATE

SQLCODE jest kodem otrzymywanym po każdym wywołaniu zapytania SQL. Poniżej pokazano, co oznaczają zwracane wartości:

SQLCODE = 0 - komenda zakończyła się powodzeniem  
SQLCODE > 0 - komenda zakończyła się powodzeniem ale, zwróciła ostrzeżeniem  
SQLCODE < 0 - komenda się nie powiodła i zwróciła błąd

SQLSTATE jest ciągiem pięciu znaków, które są zgodne ze standardem ISO/ANSI SQL92. Pierwsze dwa znaki są znane jako klasa kodu SQLSTATE:

Klasa kodu 00 oznacza, że komenda zakończyła się powodzeniem.  
Klasa kodu 01 oznacza ostrzeżenie.  
Klasa kodu 02 oznacza nieodnaleziony warunek.  
Wszystkie pozostałe klasy kodów oznaczają błędy.

## A.3 Dziennik powiadomień

Dziennik powiadomień DB2 (ang. DB2 administration notification log) dostarcza diagnostycznych informacji o błędach i miejscach awarii. Na platformach Linux/UNIX dziennik powiadomień jest plikiem tekstowym nazwanym <nazwa instancji>.nfy (np. „db2inst.nfy”). Na systemie Windows wszystkie wiadomości dziennika powiadomień są zapisywane do Dziennika Zdarzeń Systemu Windows (ang. Windows Event Log).

Parametr konfiguracyjny DB2 `notifylevel` pozwala administratorom DB2 na określenie poziomu informacji, które mają być zapisywane:

- 0 – żadne wiadomości dziennika powiadomień nie są zapisywane (nie zalecane)
- 1 – fatalne i nieodwracalne błędy
- 2 – wymaga natychmiastowej reakcji
- 3 – ważna wiadomość, nie wymaga natychmiastowej reakcji (ustawienie domyślne)
- 4 – wiadomości informacyjne

## A.4 db2diag.log

Plik logu db2diag.log zapewnia bardziej szczegółowe informacje niż dziennik powiadomień DB2. Jest on zazwyczaj używany tylko przez wsparcie techniczne IBM DB2 lub doświadczonych DBA. Informacje w db2diag.log zawierają:

- kod lokacji DB2 raportującej błąd.
- identyfikatory aplikacji pozwalające na powiązanie zapisów w db2diag.log na serwerach i klientach z aplikacjami.
- wiadomość diagnostyczna (zaczynająca się od “DIA”) wyjaśniająca powód błędu.
- inne dostępne dane, jak struktury danych SQLCA i wskaźniki do miejsc, gdzie zapisywane są inne informacje o błędach.

W systemie Windows plik db2diag jest ulokowany domyślnie w folderze:

C:\Program Files\IBM\sqlllib\<instance name>\db2diag.log

W systemach Linux/UNIX, db2diag.log jest ulokowany domyślnie w folderze:

```
/home/<instance_owner>/sqlllib/db2dump/db2diag.log
```

Złożoność diagnostycznego tekstu zależy od parametru konfiguracyjnego dbm cfg DIAGLEVEL. Zakres wynosi od 0 do 4, gdzie 0 jest najmniej złożone, a 4 najbardziej. Domyślnym poziomem jest 3.

## A.5 Śledzenie CLI

Dla aplikacji CLI i Java, możemy włączyć śledzenie obiektu (ang. trace facility) do rozwiązania problemu aplikacji. Możemy to zrobić zmieniając plik db2cli.ini na serwerze, gdzie aplikacja jest uruchomiona. Typowe zapisy w pliku db2cli są pokazane poniżej.

```
[common]
trace=0
tracerefreshinterval=300
tracepathname=/path/to/writeable/directory
traceflush=1
```

Jest także możliwe niskopoziomowe śledzenie (db2trc), ale jest zazwyczaj użyteczne tylko dla wsparcia technicznego DB2.

## A.6 DB2 – błędy i poprawki

Czasem problem, który napotkamy, może być spowodowany przez wady, błędy DB2. IBM regularnie wydaje pakiety poprawek, które zawierają poprawki kodu (tzw. APARy). Dokumentacja pakietu poprawek zawiera listę poprawek zawartych w pakiecie. Rozwijając nowe aplikacje, zawsze zalecamy używanie najnowszych pakietów poprawek. Aby zobaczyć wersję i poziom pakietu poprawek, z którą pracujemy: z Centrum sterowania, wybieramy opcję *Informacje* z menu *Pomoc*; w Edytorze komend, wprowadzamy polecenie "db2level". Zauważmy, że pakiet poprawek i oficjalne wsparcie techniczne IBM DB2 jest dostępne w DB2 Express-C, tylko jeśli zakupimy 12 miesięczną licencję subskrypcyjną.



---

## Źródła informacji o DB2

### Strony WWW

1. Strona DB2 Express-C:  
[www.ibm.com/db2/express](http://www.ibm.com/db2/express)  
Ze strony tej można ściągnąć obraz serwera DB2 Express-C, klienta DB2, sterowniki DB2, dokumentację, a także dostać się do bloga zespołu DB2 Express-C i list dyskusyjnych.
2. Forum DB2 Express:  
[www.ibm.com/developerworks/forums/dw\\_forum.jsp?forum=805&cat=19](http://www.ibm.com/developerworks/forums/dw_forum.jsp?forum=805&cat=19)  
Korzystaj z tej strony, aby zapytać o kwestie techniczne, gdy nie możesz znaleźć odpowiedzi na pytania w dokumentacji.
3. DB2 Information Center  
<http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp>  
DB2 Information Center zapewnia dostęp do dokumentacji DB2 w trybie on-line. Jest to najczęściej aktualizowane źródło informacji o DB2.
4. developerWorks  
<http://www-128.ibm.com/developerworks/db2>  
Strona ta jest doskonałym źródłem informacji dla programistów oraz administratorów DB2. Udostępnia bezpłatnie najnowsze publikacje i tutoriale.
5. alphaWorks  
<http://www.alphaworks.ibm.com/>  
Strona ta zapewnia bezpośredni dostęp do najnowszych technologii IBM. Jest to miejsce, gdzie można znaleźć informacje o nowinkach technologicznych pochodzących z IBM Research.
6. planetDB2  
[www.planetDB2.com](http://www.planetDB2.com)  
Jest to zbiór blogów o tematyce dotyczącej DB2.
7. Wsparcie techniczne DB2  
[http://www.ibm.com/software/data/db2/support/db2\\_9/](http://www.ibm.com/software/data/db2/support/db2_9/)  
Jeśli zakupiłeś 12 miesięczną licencję subskrypcyjną, z tej strony możesz ściągnąć pakiety poprawek.
8. ChannelDB2  
<http://www.ChannelDB2.com/>  
ChannelDB2 jest stroną Społeczności DB2. Dostarcza informacje, takie jak nagrania wideo dotyczące DB2, dema, prezentacje, blogi, dyskusje i inne źródła informacji dla systemów Linux, UNIX, Windows, z/OS, i i5/OS.

## Książki

1. Bezpłatny Redbook: DB2 Express-C: The Developer Handbook for XML, PHP, C/C++, Java, and .NET  
Whei-Jen Chen, John Chun, Naomi Ngan, Rakesh Ranjan, Manoj K. Sardana,  
Sierpień 2006 - SG24-7301-00  
<http://www.redbooks.ibm.com/abstracts/sg247301.html?Open>
2. Understanding DB2 – Learning Visually with Examples V9.5  
Raul F. Chong, et all. Styczeń 2008  
ISBN-10: 0131580183
3. DB2 9: pureXML overview and fast start by Cynthia M. Saracco, Don Chamberlin,  
Rav Ahuja Czerwiec 2006 SG24-7298  
<http://www.redbooks.ibm.com/abstracts/sg247298.html?Open>
4. DB2® SQL PL: Essential Guide for DB2® UDB on Linux™, UNIX®, Windows™,  
i5/OS™, and z/OS®, 2nd Edition  
Zamil Janmohamed, Clara Liu, Drew Bradstock, Raul Chong, Michael Gao, Fraser  
McArthur, Paul Yip  
ISBN: 0-13-100772-6
5. Free Redbook: DB2 pureXML Guide  
Whei-Jen Chen, Art Sammartino, Dobromir Goutev, Felicity Hendricks, Ippei Komi,  
Ming-Pang Wei, Rav Ahuja, Matthias Nicola. August 2007  
<http://www.redbooks.ibm.com/abstracts/sg247315.html?Open>
6. Information on Demand - Introduction to DB2 9 New Features  
Paul Zikopoulos, George Baklarz, Chris Eaton, Leon Katsnelson  
ISBN-10: 0071487832  
ISBN-13: 978-0071487832
7. Redbook: Developing PHP Applications for IBM Data Servers.  
Whei-Jen Chen, Holger Kirstein, Daniel Krook, Kiran H Nair, Piotr Pietrzak  
May 2006 - SG24-7218-00  
<http://www.redbooks.ibm.com/abstracts/sg247218.html?Open>

## Adresy kontaktowe

DB2 Express-C: [db2x@ca.ibm.com](mailto:db2x@ca.ibm.com)

Program DB2 on Campus: [db2univ@ca.ibm.com](mailto:db2univ@ca.ibm.com)



**Pierwsze kroki z DB2 nie mogą być prostsze.**

**Przeczytaj tę książkę, aby:**

- **dowiedzieć się, czym jest DB2 Express-C**
- **poznać architekturę DB2, narzędzia, zasady bezpieczeństwa**
- **nauczyć się administrowania bazami danych DB2**
- **nauczyć się budować zapytania SQL, XQuery, procedury składowane**
- **tworzyć aplikacje bazodanowe dla DB2**
- **sprawdzić swoją wiedzę o DB2 w praktyce**
- **przygotować się do egzaminu DB2 on Campus**

Łatwe i szybkie zastosowanie XML w procesie integracji aplikacji, Web 2.0 oraz SOA powoduje, że istnieje duże zapotrzebowanie na innowacyjne, hybrydowe serwery danych. DB2 Express-C jest bezpłatnym, Nielimitowanym, hybrydowym serwerem baz danych IBM, umożliwiającym łatwe zarządzanie zarówno danymi XML, jak i tradycyjnymi danymi relacyjnymi. DB2 Express-C można za darmo ściągnąć, bez ponoszenia dodatkowych kosztów wykorzystywać do tworzenia aplikacji bazodanowych, wdrażać do środowisk produkcyjnych, a także rozpowszechniać w ramach własnych rozwiązań. DB2 nie nakłada żadnych sztucznych limitów na wielkość bazy danych, ich liczbę, czy liczbę użytkowników.

DB2 Express-C działa na systemach Windows i Linux i dostarcza sterowniki umożliwiające pisanie aplikacji w takich językach jak C/C++, Java, .NET, PHP, Perl czy Ruby. Dostępna jest także opcjonalna, tania licencja subskrypcyjna i wsparcie techniczne wraz z dodatkami zwiększającymi możliwości serwera. Jeśli będziesz potrzebować lepszej skalowalności lub dodatkowych, zaawansowanych funkcjonalności, możesz bez problemu wdrażać aplikacje przygotowane dla DB2 Express-C na innych edycjach DB2, takich jak na przykład DB2 Enterprise.

Ta bezpłatna edycja DB2 jest idealna dla programistów, konsultantów, ISV, administratorów baz danych, studentów oraz wszystkich, którzy zamierzają rozwijać, testować, wdrażać i rozpowszechniać aplikacje bazodanowe.

Dołącz już dzisiaj do wciąż powiększającej się Społeczności DB2 Express-C i przetestuj serwer baz danych DB2 Express-C. Zaczynaj odkrywać, jak można tworzyć aplikacje nowej generacji i dostarczać innowacyjne rozwiązania.