

Rafał DUNAL

ESAProjekt sp. z o.o.

## WIELOBAZODANOWE KOMPUTEROWE SYSTEMY UŻYTKOWE

**Streszczenie.** Artykuł przedstawia systemy wykorzystujące równocześnie w swoich strukturach więcej niż jedną relacyjną bazę danych. Przedstawia jedno z podejść do problemu tworzenia i rozwijania tego typu aplikacji oraz krótko prezentuje rzeczywisty system. Referat zawiera również opis zalet i wad takiego rozwiązania oraz przewidywania dotyczące ich przyszłości.

## MULTI-DATABASE APPLICATION SYSTEMS

**Summary.** The article focuses on computer system of multidatabase structure based on relational database engines. It presents an approach towards creating and developing such systems and briefly describes an existing system. It describes advantages and disadvantages of this kind of systems and their future.

### 1. Wprowadzenie

Bazy danych od lat są podstawową częścią większości systemów komputerowych. Od ich parametrów i konfiguracji zależy wydajność każdego systemu przechowującego i przetwarzającego dane. W zdecydowanej większości przypadków systemy komputerowe stworzone są w oparciu o jeden, wybrany typ bazy danych, która odpowiedzialna jest za wydajne i bezpieczne przechowywanie oraz udostępnianie danych w nim zgromadzonych. Rozwiązanie to, choć dominujące, nie jest jedynym występującym w systemach użytkowych. W poniższym artykule postaram się przedstawić alternatywne podejście do problemu przechowywania danych w komputerowych systemach użytkowych.

## 2. Systemy wielobazodanowe

Przez pojęcie systemów wielobazodanowych rozumiem wszystkie systemy komputerowe, które używają równocześnie przynajmniej dwóch różnych baz danych. Nie należą do nich te aplikacje, które podczas swojej pracy przechowują dane w kilku takich samych bazach (np. system wykorzystuje trzy bazy MS SQL Server). Aby jeszcze bardziej zawęzić zakres mojej pracy, ograniczę się do relacyjnych baz danych, których architektura dominuje obecnie na rynku.

Wiele osób może zadać pytanie, po co stosować równocześnie kilka baz do przechowywania często tych samych danych. Odpowiedzi jest tak wiele, jak wiele systemów komputerowych i konfiguracji współpracujących ze sobą baz. W większości przypadków projektantom chodzi o wydzielenie funkcjonalnie różnych części systemu i przyporządkowanie im „niezależnego” źródła danych. Niezależność takiej bazy jest pojęciem względnym, ponieważ najczęściej stosowane jest rozwiązanie, w którym występuje jeden duży serwer bazy danych i kilka mniejszych lokalnych baz synchronizujących z nim zawartość tablic. Wielkość bazy danych też może być w różny sposób definiowana, w moim rozumieniu biorę pod uwagę faktyczną liczbę rekordów zgromadzonych we wszystkich tablicach wraz z innymi obiektami bazy, takimi jak np.: perspektywy, indeksy, sekwencje. Nie brane są natomiast pod uwagę maksymalne parametry silnika określane przez producenta. Przy takim założeniu teoretycznie może istnieć system, w którym rolę centralnej bazy danych pełniłby MS Access, a małe bazy lokalne wykorzystywałyby bardzo wydajne serwery Oracle, tylko po co?

Niektóre systemy wielobazodanowe powstają na skutek rozwinięcia (lub scalania) istniejących aplikacji opartych na jednym typie bazy danych. Głównym czynnikiem mającym wpływ na popularność takiego rozwiązania są względy czysto ekonomiczne. Klienci bardzo często nie widzą potrzeby pozbywania się systemu, który działał przez wiele lat, za który firma kiedyś zapłaciła duże pieniądze. Niestety, takie podejście nie zawsze przynosi oczekiwane korzyści, a często „nowy” hybrydowy system traci np. na szybkości jedynie dlatego, że musi być kompatybilny ze starą bazą danych.

## 3. Na styku dwóch baz danych

Problemu współistnienia w jednym systemie różnych baz danych nie byłoby w ogóle, gdyby nie konieczność wymiany (bezpośrednio lub pośrednio) informacji pomiędzy nimi. Jeżeli żaden z modułów systemu nie synchronizuje danych pomiędzy bazami, to system taki nie zawiera się w ramach tego opracowania.

### 3.1. Typy danych

Jednym z pierwszych elementów różniących większość baz są typy danych, które mogą być przechowywane w ich tablicach. Część typów jest w miarę uniwersalna, podobnie jak wygląda to w językach programowania. Jednak nawet niewielkie różnice mogą powodować poważne błędy podczas przepisywania (lub rzutowania) danych z jednego typu na drugi. Znając tablicę odwzorowań poszczególnych typów, programista może w prosty sposób świadomie konwertować dane i tak dobrać ich format, żeby nie ryzykować podczas konwersji utraty ważnych informacji.

Tabela 1

Przykładowe typy danych

Sybase	MS Access
Test	VARCHAR
Memo	TEXT
Numeric (Byte)	TINYINT
Numeric (Int)	SMALLINT
Numeric (Long Int)	INT
AutoNumber	INT
Single	REAL
Double	DOUBLE
Yes/No	BIT
Decimal	NUMERIC
Currency	NUMERIC
Date/Time	DATETIME
HyperLink	TEXT

Umiejętność manipulowania typami danych jest szczególnie ważna na etapie tworzenia struktury bazy danych, ponieważ wszelkie zmiany dotyczące typów kolumn tablic wypełnionych danymi są procesami dużo bardziej skomplikowanymi, a czasami wręcz niemożliwymi do zrealizowania. W tabeli 1 przedstawione są odpowiadające sobie typy danych baz MS Access i Sybase Adaptive Server.

Na szczególną uwagę zasługuje typ przechowujący datę, który w przypadku większości systemów pełni kluczową rolę, ponieważ informacje przechowywane w tego typu kolumnach są podstawowymi identyfikatorami poszczególnych rekordów. Typ daty i czasu występuje we wszystkich obecnie dostępnych bazach danych, problem konwersji zawartych w nim informacji dotyczy jedynie stosowanego w nim formatu. Format ten może być definiowany globalnie dla całej bazy lub w niektórych przypadkach indywidualnie dla poszczególnych kolumn.

Nie należy również zapominać o wpływie ustawień systemu operacyjnego, pod kontrolą którego baza pracuje, ponieważ systemowy format daty może być domyślnie stosowany w tablicach z danymi (podobna sytuacja dotyczy typu Currency – waluta).

Kłopoty z pozostałymi typami wynikają przede wszystkim z niezgodności rozmiarów (liczby bajtów przechowujących informacje). Różnice te zmuszają projektantów tablic do definiowania większych rozmiarów kolumn niż te, które można by zastosować w systemach korzystających z jednego typu bazy. Skrajnym przypadkiem niekompatybilności typów są typy, które występują jedynie w wybranych (czasami tylko jednym) silnikach baz danych. W takim przypadku konieczne jest stosowanie substytutów w postaci dostępnych typów formatowanych i obrabianych po stronie oprogramowania klienckiego.

### 3.2. Składnia języka

Kolejnym elementem mogącym odróżniać bazy jest składnia języka SQL (Structured Query Language). Chociaż część reguł jest bardzo podobna lub nawet identyczna, to jednak w zależności od „dialektu” użytkownik, a zdecydowanie częściej programista, boryka się z problemami odpowiedniego formułowania zapytań. Przykładem konstrukcji, którą często stosuje się w zapytaniach, a zarazem w bardzo różny sposób jej formę definiują poszczególne bazy danych, jest złączenie zewnętrzne przedstawione w poniższej tabeli.

Tabela 2

Przykład rozbieżności składni języka SQL

Baza danych	Przykład zapytania wykorzystującego złączenie zewnętrzne
MS Access	<i>SELECT Customers.CustomerID, CompanyName, OrderID FROM Customers LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID;</i>
Oracle	<i>SELECT Customers.CustomerID, CompanyName, OrderID FROM Customers, Orders WHERE Customers.CustomerID = Orders.CustomerID(+);</i>

Błędy składni z reguły powodują natychmiastowe zatrzymanie pracy programu (pod warunkiem że nie są tłumione z poziomu aplikacji), przez co ich wykrycie nie jest trudne. Największe niebezpieczeństwo ich popełnienia w zapytaniu występuje podczas tworzenia kwerend w sposób dynamiczny, czyli podczas pracy aplikacji.

Niestety, nawet dobre zapytanie może powodować błędy. Sytuacja taka może mieć miejsce na przykład w momencie komunikowania się aplikacji z bazą poprzez mechanizm ODBC (Open Database Connectivity, w wolnym tłumaczeniu: otwarte łącze baz danych), a wykorzystywany przez programistę sterownik nie obsługuje zawartej w zapytaniu funkcji. Technologia ODBC umożliwia dostęp praktycznie do każdego typu bazy (zgodnej

ze standardem ANSI SQL) w oparciu o wspólny protokół komunikacyjny. Zadaniem sterownika ODBC jest m.in. przetłumaczenie zapytania na dialekt języka SQL rozumiany przez bazę. Niestety, szybkość takiego połączenia nie zawsze jest wystarczająca i z tego powodu programiści częściej wykorzystują rozwiązania z reguły mniej uniwersalne, jednak szybsze.

### 3.3. Bezpieczeństwo

Zarządzanie dostępem do danych zgromadzonych w tablicach bazy danych jest kluczowym elementem każdego systemu. W większości przypadków każde połączenie do bazy danych wymaga podania przez użytkownika jego identyfikatora i skojarzonego z nim hasła. Uprawniony użytkownik może otworzyć sesję, w ramach której wykonuje wszelkie operacje na bazie danych. W tym samym czasie użytkownik może mieć aktywnych więcej niż jedną sesję, pod warunkiem że konfiguracja bazy tego nie zabrania.

Mechanizm autoryzacji dostępu wydaje się być uniwersalny, jednak podobieństwo występuje głównie tylko po stronie klienta. Właściwe zabezpieczenia znajdują się natomiast w samej bazie danych, programiści i użytkownicy korzystają jedynie z udostępnianych do nich przez producentów interfejsów. Jeżeli tylko nie jest to system wymagający szczególnych zabezpieczeń, to dla użytkownika większość zabezpieczeń powinna być niewidoczna.

W zależności od konfiguracji systemu przesyłanie nazwy użytkownika i jego hasła może odbywać się w sposób jawny lub szyfrowany (zalecany). W celu uniknięcia konieczności pamiętania kilku haseł i poprawienia komfortu pracy niektóre bazy danych (np. Oracle, MS SQL Server) oferują możliwość wykorzystywania mechanizmów weryfikacji osób poprzez system operacyjny. W takim wypadku użytkownicy systemu operacyjnego (lub domeny) muszą posiadać prawo do dostępu do określonych obiektów bazy danych. Możliwe jest również stosowanie wyspecjalizowanego sprzętu poprawiającego bezpieczeństwo (np. karty chipowe, identyfikacja biometryczna).

Współpraca dwóch różnych baz danych z reguły wpływa na obniżenie ogólnego bezpieczeństwa. Jest to powodowane faktem, iż dublowanie konfiguracji wszystkich zabezpieczeń występujących w poszczególnych bazach jest dużo bardziej czasochłonne i uciążliwe. Efektem tej niedogodności może być rozsynchronizowanie listy kont użytkowników każdej z baz lub ich uprawnień. W takiej sytuacji zbawienne może okazać się wykorzystanie wspomnianej autoryzacji poprzez system operacyjny. Jeżeli bazy, z którymi ma pracować system, nie oferują satysfakcjonujących zabezpieczeń, programiści często decydują się na tworzenie własnych, programowych rozwiązań.

### 3.4. Aktualizacja danych

Bardzo ważną cechą danych przechowywanych w bazie jest ich aktualność. W dzisiejszych czasach informacja bardzo szybko ulega zmianie (dezaktualizuje się), tak więc konieczne jest odpowiednie zarządzanie tymi zmianami ze szczególnym zwróceniem uwagi na synchronizację równoczesnego dostępu do tablic i rekordów kilku użytkowników.

Zastosowanie więcej niż jednej bazy danych w systemie powoduje decentralizację informacji, i co za tym idzie, zwiększenie ryzyka wystąpienia problemów z ich aktualizacją. W tego typu systemach część danych jest powielana przez projektantów w celu udostępnienia ich kopii w bazach lokalnych. Niektóre bazy danych np. MS Access posiadają wbudowane narzędzia do replikacji danych, nie zawsze jednak spełniają oczekiwania projektantów i użytkowników, dlatego też bywają zastępowane rozwiązaniami programowymi.

W systemach wielobazodanowych nie można uniknąć rozszynchronizowania się poszczególnych tabel przechowujących „te same” dane. Ponieważ jest to praktycznie nieuniknione, nie należy przede wszystkim mieć tego świadomość i w sytuacjach kiedy wykonywane są kluczowe operacje systemu, wymuszać synchronizację baz. W celu optymalizacji takiej operacji powinno się wydzielić grupy tablic, które przechowują dane o podobnym prawdopodobieństwie ich aktualizacji przez użytkownika (lub system zewnętrzny). Dla każdej zdefiniowanej grupy należy określić politykę aktualizacji i synchronizacji informacji.

Przechodząc do wspomnianego tematu synchronizacji dostępu, to w bardzo dużym stopniu zapewniają ją mechanizmy zasyte w silniku bazy danych. Po stronie programistów leży obowiązek odpowiedniego ich stosowania oraz właściwej konfiguracji parametrów bazy (np. włączenie/wyłączenie trybu „autocommit” – automatyczne zatwierdzanie wykonanych zmian). Jednym z przykładów takiego typu mechanizmów są transakcje, zapewniają one między innymi spójność danych. Baza danych zawsze przeprowadzana jest za ich pomocą z jednego do drugiego stanu spójnego. Użytkownicy mają wtedy pewność, że albo wszystkie zmiany przez nich wykonane na danych zostaną zatwierdzone, albo operacje zostaną wycofane do ostatniego stanu spójnego.

Problemy związane z dezaktualizacją danych mogą wynikać z braku aktywnego połączenia z jedną z baz. Sytuacja taka ma często miejsce, kiedy system połączony jest ze zdalnym serwerem bazy za pomocą sieci lub łącza modemowego. W takim wypadku konieczne jest zastosowanie różnych technik odtwarzających połączenie i przywracających zawartość każdej z baz do stanu sprzed wystąpienia awarii lub równoległa aktualizacja danych po obydwu stronach połączenia.

### 3.5. Bilans zysków i strat

Zetknięcie się ze sobą różnych podejść producentów baz danych do problemu składowania danych w systemach komputerowych, jak już wiemy, powoduje często kłopoty. Opisane powyżej pokrótce sytuacje obejmują zaledwie wycinek problemów, z którymi może spotkać się programista tworzący (lub rozwijający) system wielobazodanowy. Poniższa tabela zawiera próbę bilansu tego, co tracimy i zyskujemy decydując się na tego typu system.

Tabela 3

Wady i zalety systemów opartych na co najmniej dwóch typach baz danych

Zalety	Wady
<ul style="list-style-type: none"> <li>• odciążenie głównego serwera bazy danych,</li> <li>• zminimalizowanie ruchu w sieci poprzez przechowywanie lokalnie rzadko zmienianych danych,</li> <li>• rozbieżność użytkowników na podgrupy przypisane poszczególnym bazom,</li> <li>• zwiększenie pojemności i możliwości systemu.</li> </ul>	<ul style="list-style-type: none"> <li>• problemy z aktualizacją danych w poszczególnych bazach,</li> <li>• rozbieżność składni języka SQL,</li> <li>• większy nakład pracy wymagany od twórców systemu,</li> <li>• problemy dotyczące kompatybilności poszczególnych baz z systemami operacyjnymi.</li> </ul>

Oczywiście, nie wszystkie wady i zalety znalazły miejsce w tabeli 3, a te, które tam są w zależności od systemu mogą być inaczej zaklasyfikowane przez projektantów i użytkowników, niż ja to uczyniłem. Trudno się jednak oprzeć subiektywności wyboru, na który w moim przypadku ogromny wpływ miała praca nad systemem przedstawionym w dalszej części tego referatu.

## 4. EcoSystem – przykład rzeczywistego systemu

System, który chcę przedstawić w tym punkcie, skłonił mnie do zajęcia się problemem wielobazodanowości. EcoSystem, bo o nim będzie mowa, jest produktem włoskiej firmy Project Automation S.p.A i jest przeznaczony do monitorowania wybranych parametrów środowiska naturalnego. System ten jest klasycznym przykładem systemu wielobazodanowego. Jest on w obecnej formie oparty na dwóch typach relacyjnych baz danych:

- Sybase Adaptive Server 12,
- Microsoft Access 2000.

Wykorzystane bazy różnią się przede wszystkim możliwościami oraz podstawowymi parametrami pracy. Systemy tak odmienne na pewno nie były tworzone z myślą o wykorzystywaniu ich równocześnie w rozbudowanym systemie komputerowym, w którym gromadzone są ogromne ilości danych każdego dnia.

Zarówno Microsoft, jak i Sybase kładą bardzo duży nacisk na rozwój aplikacji ściśle współpracujących z bazami danych, wydaje się jednak, że będąc konkurentami nie są zainteresowane standaryzacją mechanizmów wykorzystywanych w swoich silnikach. W przeszłości Microsoft zakupił od firmy Sybase silnik bazy danych do swojego obecnie sztandarowej bazy – MS SQL Server i produkty te były rozwijane razem aż do wersji 4.2. Obecnie właśnie MS SQL Server jest głównym konkurentem Adaptive Servera Sybase'a, natomiast MS Access wykorzystywany jest raczej do tworzenia interfejsu graficznego użytkownika i zarządzania stosunkowo (w porównaniu do dużych serwerów) niewielkimi bazami danych.

#### 4.1. Ogólna struktura systemu

EcoSystem jest rozbudowanym systemem monitorującym, na określonym terenie, wybrane parametry związane z ochroną środowiska. Zasięg monitorowanego obszaru może być praktycznie niczym nie ograniczony. W rzeczywistości są to tereny określone przez granice administracyjne (np. regiony, prowincje) państwa, ponieważ w zdecydowanej większości EcoSystem wykorzystywany jest przez instytucje państwowe. Na badanym terenie umiejscowione są stacje pomiarowe wyposażone w instrumenty badające określone czynniki (np. CO, NO<sub>2</sub>, temperaturę). Stacje pomiarowe wyposażone są w autorskie (nie oferowane na rynku) bazy danych oparte na cyklicznych plikach binarnych, na których można dokonywać prostych operacji odczytu/zapisu w formach zbliżonych do zapytań SQL. Każdej grupie stacji pomiarowych przypisana jest przynajmniej jedna stacja centralna wyposażona w wydajny serwer bazy danych, do której dane pobierane są poprzez sieć lub połączenie modemowe, gdzie następnie poddawane są przeliczeniom i analizie. W skład systemu wchodzi następujące elementy (bazy danych i aplikacje):

- **serwer bazy danych Sybase Adaptive Server 12,**
- EcoManager Server – serce systemu, zestaw procesów wykonujących m.in. wszelkie przeliczenia potrzebne do analiz oraz operacje pobierania danych ze stacji pomiarowych,
- EcoManager Client – narzędzie, wyposażone w rozbudowany interfejs graficzny, służące do przeglądania danych pomiarowych oraz konfigurowania zadań związanych z obróbką zebranych danych,



- EcoRemote – aplikacja działająca w stacjach pomiarowych, odpowiedzialna za bezpośrednią akwizycję danych z instrumentów pomiarowych,
- EcoEdit – narzędzie do konfiguracji podstawowych parametrów całego systemu,
- EcoAnalyzer – moduł raportujący, wykorzystywany do analizy danych zgromadzonych w systemie,
- EcoExchange – uniwersalny moduł do importu i eksportu danych do/z bazy danych systemu, wyposażony w możliwość konfiguracji zadań wykonywanych automatycznie bez udziału użytkownika,
- Data Validation – moduł wykonujący automatyczną walidację (na podstawie określonych testów) danych pobranych ze stacji.

Na powyższej liście wyróżnione zostały pozycje, które są rzeczywistymi bazami danych. Aplikacje EcoEdit i EcoAnalyzer zostały stworzone z wykorzystaniem formularzy, raportów i struktur dostępu do danych oferowanych przez bazę MS Access (pliki .mdb). Posiadają one lokalne tablice, w których znajdują się kopie części danych znajdujących się w bazie centralnej.

Ze względu na temat tego opracowania, w dalszej części będę się jedynie zajmował mechanizmami i problemami związanymi z wymianą danych pomiędzy bazami Access oraz Sybase. Pozostałe aplikacje komunikują się bezpośrednio jedynie z bazą danych Sybase i mają przede wszystkim wpływ na jej wydajność. Programy te w większości przypadków łączą się z serwerem bazy danych za pomocą bibliotek oferowanych przez producenta. Podejście takie gwarantuje najszybszą pracę oraz najszerszy wachlarz funkcji oferowanych przez serwer, jednak jest praktycznie przypisane do jednego typu bazy.

Podstawowym założeniem dla EcoSystemu była centralizacja danych na poziomie lokalnego centrum. Założenie to miało bardzo duży wpływ na wybór wydajnego serwera bazy danych. Produkt Sybase'a spełniał wymagania stawiane zarówno przez potencjalnych użytkowników systemu, jak i jego projektantów. Jego niewątpliwą zaletą był fakt, że od lat serwer bazy danych Sybase dostępny był na różne platformy systemowe. Serwer bazy danych Sybase'a, poza gromadzeniem danych pobieranych z instrumentów pomiarowych znajdujących się w odległych stacjach, odpowiedzialny jest za zapewnienie danym wymaganego poziomu bezpieczeństwa. Poza autoryzacją użytkowników, po stronie serwera spoczywa również obowiązek m.in. tworzenia kopii bezpieczeństwa i zarządzanie mechanizmami odpowiedzialnymi za odtwarzanie bazy danych po awarii.

Chcąc odciążać serwer Sybase'a, a równocześnie zaprezentować w trybie graficznym użytkownikowi wybrane funkcje systemu i udostępnić informacje w postaci raportów i wykresów, dwa moduły EcoSystemu zostały w pełni zrealizowane w oparciu o bazy MS Access. Choć z użytkowego punktu widzenia programy te (EcoEdit i EcoAnalyzer) pełnią całkiem odmienne role, to mechanizmy w nich wykorzystane związane z zarządzaniem

danymi systemu są takie same. Właśnie one i ich współpraca z serwerem Sybase'a zostaną omówione w dalszej części niniejszego opracowania.

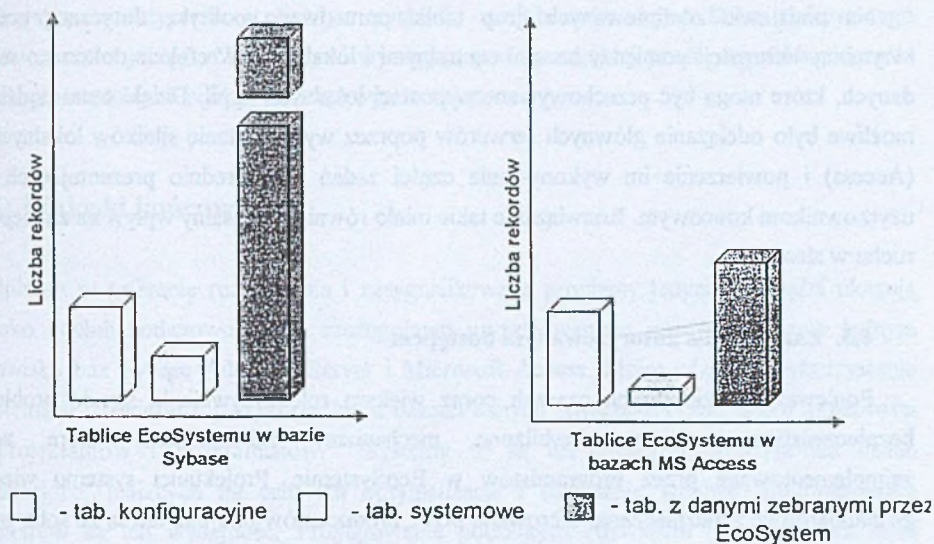
#### 4.2. Tablice systemu

System monitorujący wybrane parametry środowiska naturalnego wymagał odpowiednio zaprojektowanej struktury bazy danych. Wszystkie tablice znajdujące się w bazie Sybase'a zostały podzielone na grupy w zależności od rodzaju danych w nich przechowywanych, są to:

- tablice konfiguracyjne – przechowują konfigurację poszczególnych elementów systemu (sieci, stacje itp.),
- tablice systemowe – wykorzystywane przez główne procesy systemu,
- tablice z danymi – przechowują dane otrzymane ze stacji pomiarowych lub otrzymane w wyniku dokonanych przeliczeń.

Powyższy podział nie tylko definiuje pełnione przez tablice funkcje, ale również częstotliwość, z jaką zmieniana jest ich zawartość.

W przypadku tablic konfiguracyjnych modyfikacje dokonywane są sporadycznie. Zawartość tych tablic zmieniana jest głównie w początkowej fazie określania parametrów systemu, kiedy to administrator określa ustawienia każdej stacji pomiarowej. Na tym etapie definiowane są zgodnie z wymaganiami użytkowników ich podstawowe parametry. Po wykonaniu właściwej konfiguracji jedynie użytkownicy, będący administratorami poszczególnych sieci, mogą dokonywać modyfikacji jej parametrów, jednak zdarza się to wyjątkowo rzadko. Jest to spowodowane tym, iż konfiguracja jest ściśle związana ze sprzętem, pracującym w zdalnych stacjach pomiarowych, którego zmiany podczas normalnej pracy systemu nie mają miejsca. Rekordy tablic konfiguracyjnych są sporadycznie modyfikowane, jednak bardzo często wykonywane są na nich operacje odczytu. Zawierają one podstawowe informacje konieczne podczas prawie każdej operacji systemu, w której wykonywane są zapytania dotyczące danych pomiarowych. Są to np. listy przyrządów pomiarowych zainstalowanych na danej stacji czy jednostki, w jakich dokonywane zostają odczyty.



Rys. 1. Szacunkowe porównanie liczby rekordów w poszczególnych tablicach systemu

Fig. 1. Estimation of number of records in particular tables

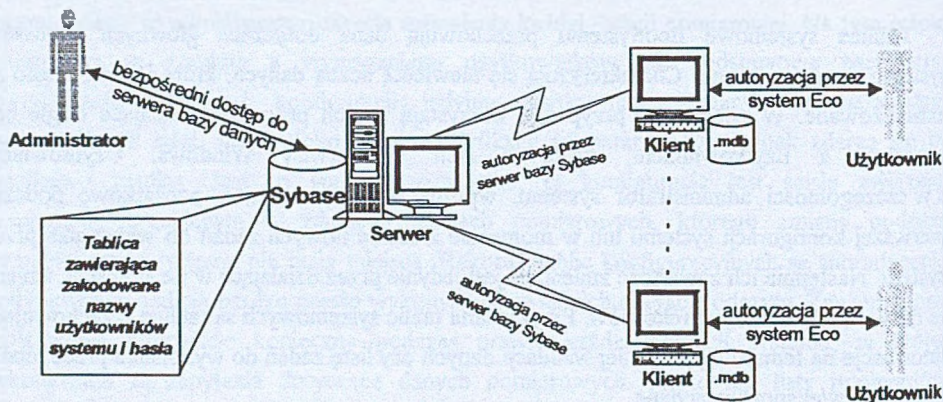
Tablice systemowe EcoSystemu przechowują dane dotyczące głównych procesów systemu monitorującego. Charakteryzują się niewielką liczbą danych, które jednak często są aktualizowane. W większości przypadku korzystają z nich programy działające ciągle bez interakcji z użytkownikiem, część z nich to serwisy Windows. Użytkownicy, a w szczególności administrator systemu, wpisują do nich wartości początkowe podczas pierwszej konfiguracji systemu lub w momencie zlecenia nowych zadań do wykonania przez system. Następnie ich zawartość zmieniana jest jedynie przez działające w tle aplikacje, zmiany te dotyczą pojedynczych rekordów. Przykładami tablic systemowych są tablice przechowujące informacje na temat daty ostatniej walidacji danych czy listę zadań do wykonania przez moduł importujący/eksportujący dane.

Każdy z typów tablic jest równie ważny dla poprawnej pracy systemu, jednak z punktu widzenia serwera bazy danych Sybase, tablice przechowujące dane otrzymane z instrumentów pomiarowych odgrywają kluczową rolę. Swoją pozycję zawdzięczają głównie ogromnej liczbie rekordów w nich przechowywanych oraz złożoności i częstotliwości wykonywanych na nich zapytań. Operacje na nich wykonywane mają kolosalny wpływ na szybkość, a co za tym idzie wydajność zarówno silnika bazy Sybase'a, jak i lokalnych baz opartych na plikach .mdb. Dane w nich zawarte nie tylko pochodzą bezpośrednio z instrumentów pomiarowych, ale również są wynikiem przeliczeń, które mają za zadanie ich walidację i obliczanie wartości średnich, np. dziennych czy miesięcznych.

Na podstawie zdefiniowanych grup tablic opracowano politykę dotyczącą podziału i wymiany informacji pomiędzy bazami centralnymi i lokalnymi. W efekcie dokonano selekcji danych, które mogą być przechowywane w postaci lokalnych kopii. Dzięki temu podziałowi możliwe było odciążanie głównych serwerów poprzez wykorzystanie silników lokalnych baz (Access) i powierzenie im wykonywania części zadań bezpośrednio prezentujących dane użytkownikom końcowym. Rozwiązanie takie miało również kolosalny wpływ na zmniejszenie ruchu w sieci.

### 4.3. Zarządzanie autoryzowanym dostępem

Ponieważ w dzisiejszych czasach coraz większą rolę przywiązuje się do problemów bezpieczeństwa, zostaną przybliżone mechanizmy zabezpieczeń, które zostały zaimplementowane przez programistów w EcoSystemie. Projektanci systemu wiedzieli, że standardowe zabezpieczenie oferowane przez producentów obu baz nie są ze sobą w pełni kompatybilne. Postanowiono więc, że w serwerze Sybase'a w jednej z tablic konfiguracyjnych przechowywane będą nazwy wszystkich użytkowników, ich zaszyfrowane hasła i lista uprawnień do opcji poszczególnych aplikacji wchodzących w skład systemu.



Rys. 2. Autoryzacja użytkowników systemu  
Fig. 2. System users authorization

Wszystkie aplikacje, czyli także te oparte na plikach .mdb weryfikują swoich użytkowników bezpośrednio na podstawie danych zawartych w tablicach serwera. Połączenie z bazą Sybase'a realizowane jest poprzez jednego uniwersalnego użytkownika, który posiada uprawnienia do wszystkich funkcji bazy danych wymaganych do jego poprawnej pracy. Rozwiązanie takie umożliwia dostęp do danych wszystkich użytkowników systemu z każdego miejsca, w którym zainstalowane jest oprogramowanie klienta. Ponieważ nie bazuje na żadnych mechanizmach specyficznych dla jednego typu bazy danych, może zostać

wykorzystane praktycznie w każdym systemie wielobazodanowym. Dane opisywanego systemu monitorującego nie są danymi wymagającymi szczególnej poufności, zabezpieczenia tak zaimplementowane wydają się więc wystarczające.

## 5. Wnioski końcowe

Opisane w referacie rozwiązania i zasygnalizowane problemy jedynie w części ukazują zjawisko wielobazodanowości, ze szczególnym uwzględnieniem współdziałania w jednym środowisku baz Sybase Adaptive Server i Microsoft Access. Moim zdaniem wykorzystanie w systemach rozwiązań opartych na kilku bazach danych stwarza zupełnie nowe możliwości dla projektantów i programistów. Systemy te są też idealnym tematem do badań akademickich, mających na celu ich optymalizację i określanie wpływu poszczególnych parametrów na ich wydajność. Propagowanie podobnych rozwiązań może również mieć wpływ na rozwój prac mających na celu standaryzację interfejsów różnych producentów silników baz danych, a szczególnie do stworzenia uniwersalnych mechanizmów zabezpieczeń.

## LITERATURA

1. Muraszkiwicz M., Rybiński H.: Bazy danych. Akademicka Oficyna Wydawnicza, Warszawa 1993.
2. Delobel C., Adiba M.: Relacyjne bazy danych. WNT, Warszawa 1989.
3. Strona firmy Project Automation – <http://www.p-a.it> .
4. Strona firmy Microsoft – <http://www.microsoft.com> .
5. Strona firmy Sybase– <http://www.sybase.com> .

Recenzent: Prof. dr hab. inż. Stanisław Kozielski

Wpłynęło do Redakcji 11 kwietnia 2002 r.

## Abstract

The article describes computer systems that use in their structures more than one engine type of relational database. Such systems, within this study, were defined as multi-database

systems. Opinions presented here result, to a large extent, from observations made during my work consisting in creation and modification of Eco series system in the Italian company Project Automation S.p.A.. My experiences connected with this subject had been already presented at the doctorate seminars at the Faculty of Automatic Control, Electronics and Computer Science at The Silesian University of Technology.

Multi-database systems do not play the dominant role on the market. Such solutions are used mainly in huge systems that consist of several applications requiring access to various types of data. In such systems, as a rule, there is one efficient database server, which houses centrally all (or almost all) data, and a group of smaller engines of local databases. There is a connection between the bases, via net or modem link, used mainly for data synchronization and exchange of information applied by protection mechanisms.

Multi-database systems create new possibilities for system designers and programmers. Architecture based on direct / indirect data exchange between different database engines can also cause problems related first of all to discrepancies in data types, syntax of SQL language (Structured Query Language), as well as in data protection mechanisms. Depending on their approach, system designers use either solutions dedicated to chosen bases types, which ensure the fastest access or more universal solutions, which are generally much slower. Italian EcoSystem, which is the system monitoring chosen parameters of natural environment, is an example of multi-database solution based on Sybase Adaptive Server 12 and Microsoft Access 2000.

Presented solutions and problems resulting from them partially show the multi-database phenomenon. Such systems seem to be an ideal subject for academic research, aimed at their optimization and defining the influence of single parameters on their efficiency. Supporting similar solutions can also influence the development of work focused on standardization of interfaces coming from various producers of database engines, and particularly on creating universal protection mechanisms.