STUDIA INFORMATICA 2002
Volume 23 Number 4 (51)

Zygmunt BOK
Zakłady Tworzyw Sztucznych ”Nitron” S.A.

INTEGRACJA RELACYJNYCH BAZ DANYCH W ZASTANYCH PRZEMYSŁOWYCH SYSTEMACH INFORMATYCZNYCH

Streszczenie. W tym artykule - na przykładzie przemysłowego systemu informatycznego pracującego w Zakładach Tworzyw Sztucznych „Nitron” S.A. pod kontrolą sieciowego systemu operacyjnego Novell NetWare 4.x - przedstawiono metodę integracji relacyjnych baz danych typu *dBase*, wykorzystującą technologię ODBC oraz rozszerzony schemat bazy NDS, który użyto do definicji schematu i implementacji tematycznie zorientowanego systemu magazynu danych opartego na modelu relacji uniwersalnej.

INTEGRATION OF RELATIONAL DATABASE SYSTEMS IN INDUSTRIAL LEGACY INFORMATION SYSTEMS

Summary. In this article - on example industrial information system working in ZTS ‘Nitron’ S.A. factory under network Novell NetWare 4.x operating system control, an integration method of relational *dBase* database systems was presented based on ODBC technology and extended schema NDS database, which was used to schema definition and subject-oriented data warehouse system implementation based on universal relation model.

1. Wprowadzenie

W rzeczywistych, obecnie jeszcze eksploatowanych przemysłowych systemach informacyjnych, działających w oparciu o bazy danych typu *dBase*, istnieje potrzeba uzyskania informacji zbiorczych zawartych w bazach danych dotyczących poprzednich zamkniętych okresów obliczeniowych. Przykładowo, w systemie sprzedaży wyrobów gotowych eksploatowanym w ZTS „Nitron” S.A., problemem dla pracowników służb marketingowych jest uzyskanie szybkiej odpowiedzi na analityczne pytania typu: „*Jak kształtowała się realizacja sprzedaży na*

przestrzeni ostatnich 10 lat?" lub „*Jaka była sprzedaż wyrobu X na przestrzeni ostatnich 10 lat?*". Innymi słowy, istnieje problem zadawania analitycznych pytań *ad hoc* dotyczących jednej lub wielu (w zależności od zakresu pytania) oddzielnych baz danych i uzyskania jednoznacznej odpowiedzi. Jednym z możliwych podejść do tak postawionego problemu jest wykorzystanie koncepcji federacyjnych baz danych, zapewniających zintegrowany dostęp do heterogenicznych i niezależnie administrowanych relacyjnych baz danych. W tym miejscu wspomnieć należy, że jeszcze w chwili obecnej istnieje wiele eksperymentalnych projektów badawczych [1] dotyczących federacyjnych baz danych. W tych bazach, przy rozwiązywaniu konkretnych problemów związanych z definicją schematu federacyjnej bazy danych wykorzystano relacyjny [2] oraz obiektowy model danych [3]. Projekty te przyczyniły się do powstania technologii sfederowanych baz danych. Oferują one rozwiązanie problemów integracji strukturalnych i funkcjonalnych elementów modeli danych oraz integracji operacyjnych systemów przetwarzania transakcyjnego typu OLTP (ang. On Line Transaction Processing) [4]. W systemach tego typu operacje wprowadzania, aktualizacji oraz wyszukiwania danych wykonywane są w ramach tzw. transakcji. Jednym z głównych zadań tego typu baz danych jest wykonywanie na bieżąco dużej liczby transakcji [5]. Stanowią one jednak przyczynę niskiej wydajności i efektywności aplikacji analitycznych eksploatowanych w środowisku tego samego systemu bazy danych [4]. Prowadzone w ostatnim czasie prace nad środowiskiem poprawiającym efektywność przetwarzania analitycznego doprowadziły do powstania nowej koncepcji przetwarzania danych typu OLAP (ang. On Line Analytical Processing), wspierającej technologię magazynów danych, tj. wydzielonych systemów baz danych wspomagających przetwarzanie analityczne. Architektura tych systemów [6] zakłada pełną izolację przetwarzania operacyjnego i analitycznego. Informacje powstające w operacyjnych bazach danych są replikowane i fizycznie składowane w magazynie danych do późniejszego przetwarzania analitycznego.

Wspomniany na wstępie zbiór archiwalnych relacyjnych baz danych $\boldsymbol{r} = \{r_{ij}\}$ o schematach $\boldsymbol{R}=\{R_{ij}\}$ potraktować można jako podstawowe repozytorium informacji lub inaczej jako pewny szczególny rodzaj magazynu danych. W tym zbiorze wskaźnikiem $i\in\{1,...,n\}$ oznaczano poszczególne archiwalne bazy danych, natomiast wskaźnikiem $j\in\{1,...,m\}$ kolejne jej relacje. Schemat tego szczególnego magazynu danych definiowany jest jako suma atrybutów schematu abstrakcyjnej relacji uniwersalnej $\boldsymbol{u}$ utworzonej nad zbiorem sumy atrybutów $\boldsymbol{R}=\{\Sigma R_{ij}\}$ [9,10,11,12]. Ponieważ w tym magazynie dla każdego ustalonego j zachodzi $R_{1j}= R_{2j}=...=R_{nj}$, zatem jego schemat definiowany jest jako suma atrybutów $\boldsymbol{R}=\{\Sigma R_j\}$. Innymi słowy, odpowiednie relacje w poszczególnych bazach składowych opisywanego zbioru baz danych mają taki sam schemat. Z tak pojmowanego szczególnego

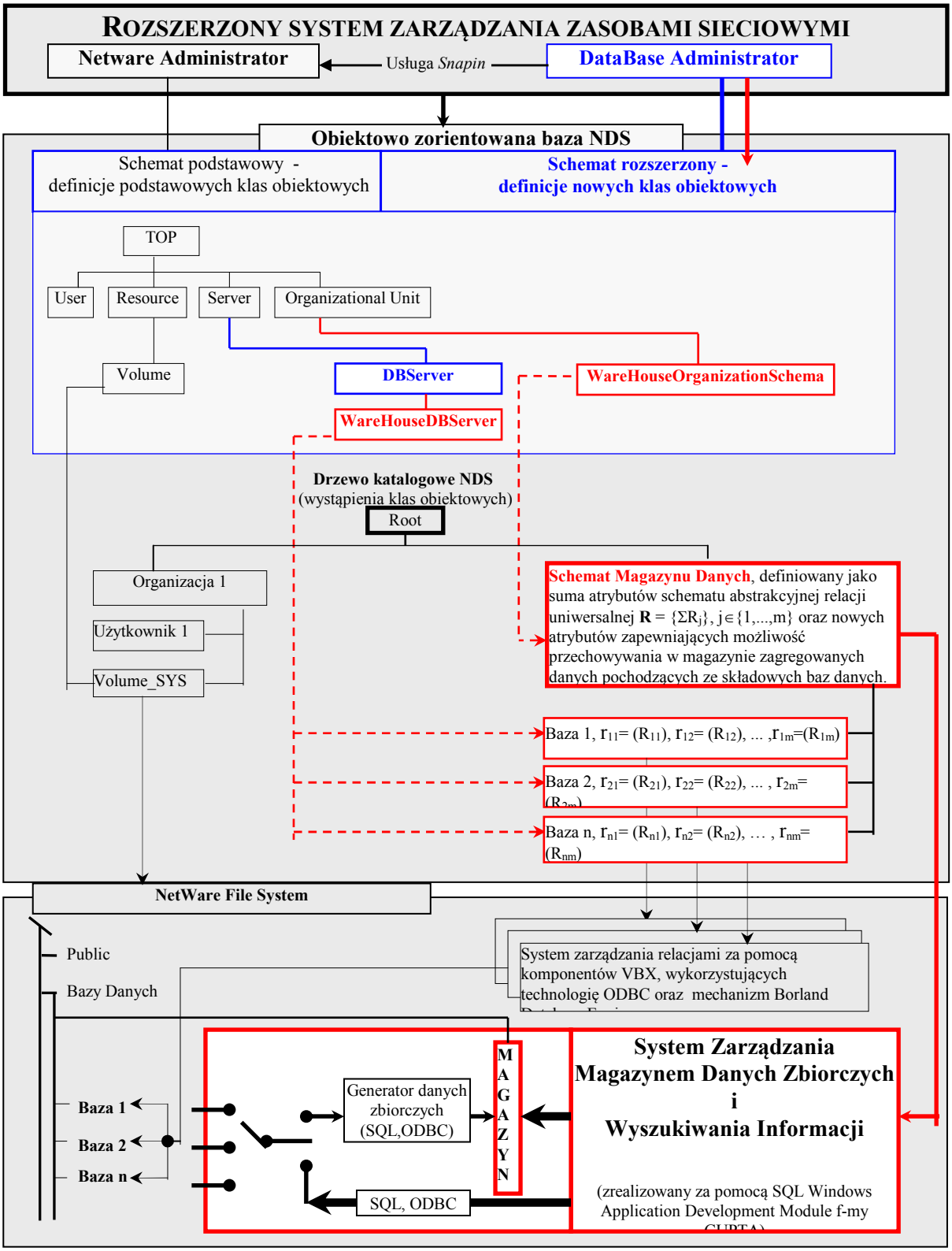
magazynu danych istnieje potrzeba uzyskania informacji zbiorczych, uwzględniających między innymi aspekt czasu.

2. Opis metody

W celu realizacji tak sformułowanego we wprowadzeniu zadania skoncentrowano się na problemie integracji zbioru schematów składowych omawianych archiwalnych relacyjnych baz szczególnego magazynu danych. Wykorzystano w tym celu metodę [13] opierającą się na założeniu, że jeśli każdy zasób sieciowy w sieci Novell NetWare 4.x reprezentowany jest przez pewien obiekt NDS-u[1], tworzony i zarządzany poprzez graficzny interfejs użytkownika znany pod nazwą *NetWare Administrator*, to można również utworzyć inne obiekty w katalogowej bazie NDS, które reprezentować będą elementy składowe szczególnego magazynu danych, zainstalowanego na pewnym serwerze sieciowym. Realizacja tego typu obiektów jest możliwa dzięki rozszerzaniu schematu katalogowej bazy NDS [14]. Właściwości tych obiektów powinny być między innymi takie, aby uprawnionym użytkownikom sieci komputerowej umożliwić uzyskanie bezpośredniego dostępu, za pośrednictwem pewnej warstwy oprogramowania i technologii ODBC [15,16,17], do informacji zawartych w bazach składowych tego magazynu danych. Powinny również umożliwić odwzorowanie schematu szczególnego magazynu danych w katalogowej bazie NDS [18] oraz dodatkowych atrybutów charakteryzujących zagregowane dane, w tym uwzględniających między innymi wymiar czasu. Suma tych atrybutów dodatkowych i atrybutów szczególnego magazynu danych stanowi podstawę do definicji i implementacji tematycznie zorientowanego magazynu danych. Magazyn ten służyć będzie do przechowywania zagregowanych danych, pochodzących ze składowych archiwalnych baz danych. Aby można było udzielić odpowiedzi na sygnalizowane we wprowadzeniu pytania analityczne, schemat tego magazynu powinien zostać odpowiednio zdefiniowany z uwzględnieniem charakteru tych pytań. O pytaniach tych wiadomo tylko tyle, że powinny rozszerzać zbiór standardowych zestawień, predefiniowanych w aplikacji obsługującej bazy archiwalne. Horyzont czasowy tych zestawień sięga jednego roku, tj.: dzień, tydzień, miesiąc, kwartał, rok. Innymi słowy,

[1] NDS (ang. Novell Directory Services) jest obiektowo zorientowaną bazą danych [27,28,29,30,31]. W związku z tym, że w dostępnej autorowi literaturze brak jest odnośników uzasadniających powyższe stwierdzenie w tym sensie, że baza ta spełnia wszystkie wymagania stawiane obiektowym bazom danych [7,8], dlatego też w niniejszym artykule używać się będzie określenia „NDS jest obiektowo zorientowaną implementacją bazy informacyjnej, tj. usług katalogowych" [13,14,18] lub w skrócie „Katalogową bazą NDS".

otrzymywane odpowiedzi na zadane pytania analityczne w wymiarze czasu powinny obejmować zagregowane dane dotyczące kilku lat.



Rys. 1. Zarys metody wykorzystania schematu bazy NDS do definicji i implementacji tematycznie zorientowanego schematu magazynu danych

Fig. 1. The layout of using NDS database method to schema definition and subject-oriented data warehouse system implementation

W opisywanej metodzie - schematycznie przedstawionej na rys. 1 - zdefiniowano i zaimplementowano schemat tematycznie zorientowanego magazynu danych jako tzw. meta dane, które zachowano w postaci obiektów trwałych w katalogowej bazie NDS.

Istota opisywanej metody polega na odwzorowaniu w katalogowej bazie NDS opisu i schematu tematycznego magazynu danych, zwanego w dalszej części artykułu Magazynem Danych Zbiorczych oraz fizycznym jego utworzeniu w strukturze plików pewnego serwera sieciowego. Jak widać z rys. 1, każdą bazę ze zbioru archiwalnych relacyjnych baz danych $\boldsymbol{r}=\{r_{ij}\}$ odwzorowano w kolejnych wystąpieniach klasy obiektowej *WareHouseDBServer*, natomiast schemat Magazynu Danych odwzorowano w wystąpieniu klasy obiektowej *WareHouseOrganizationSchema*. Schemat tego magazynu definiowany jest jako suma atrybutów szczególnego magazynu danych $\mathbf{R}=\{\Sigma R_j\}$ oraz nowych atrybutów, zapewniających możliwość przechowywania w magazynie zagregowanych danych pochodzących ze składowych baz danych.

Magazyn Danych Zbiorczych zasilono następnie zagregowanymi danymi pochodzącymi z poszczególnych składowych archiwalnych baz danych, wykonując z poziomu systemu zarządzania tym magazynem, przy wykorzystaniu technologii ODBC, odpowiednie zapytania SQL. Z drugiej strony do tak zdefiniowanego i zaimplementowanego Magazynu Danych Zbiorczych kierowano za pośrednictwem przykładowego podsystemu Zarządzania i Wyszukiwania Informacji Zbiorczych pytania analityczne, po wykonaniu których za jego pomocą otrzymywano odpowiednie raporty. Pytania analityczne wyrażano w języku SQL, które następnie zachowywano w bazie NDS w celu późniejszego wykorzystania. Za pomocą wspomnianego podsystemu pytania analityczne tworzono również *ad hoc* lub interaktywnie konstruowano, wykorzystując koncepcję modelu relacji uniwersalnej.

2.1. Realizacja metody

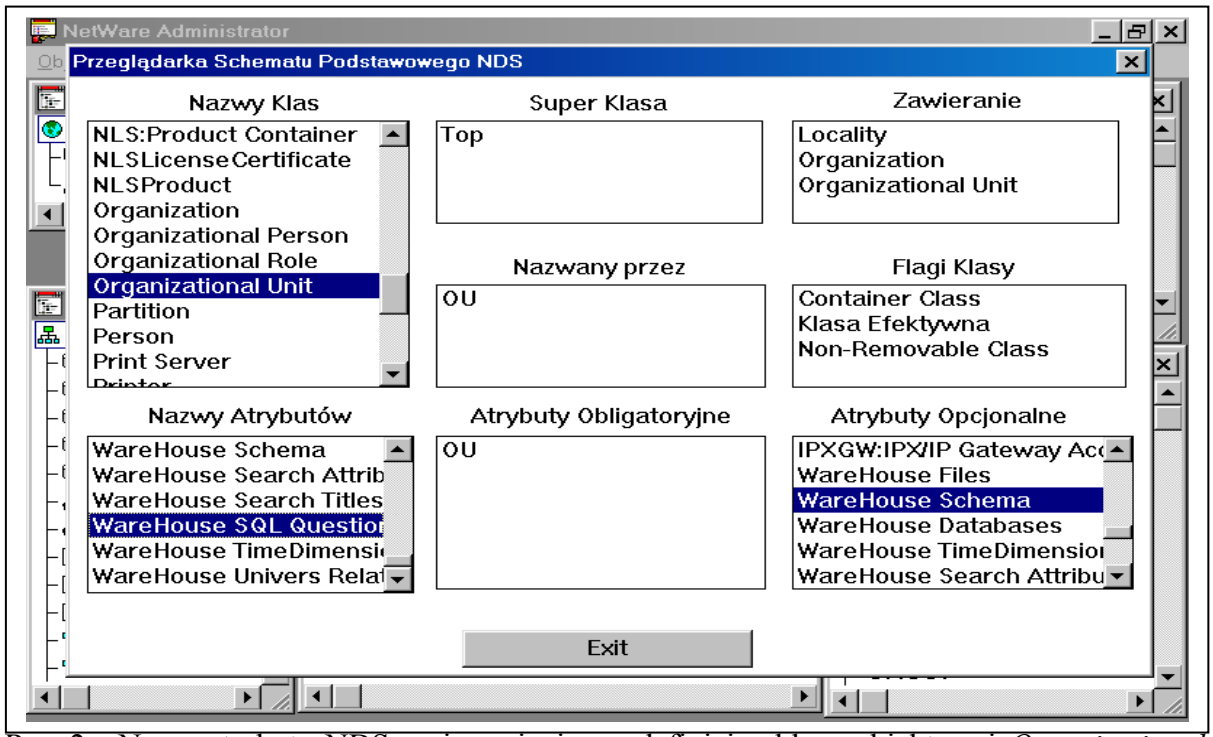
Realizacja opisanej metody polega na:

1) rozszerzeniu standardowego interfejsu użytkownika *NetWare Administrator* za pomocą usługi *Snapin* w taki sposób, aby w ramach odpowiednich opcji menu uruchamianych spod opcji *Tools* możliwe było utworzenie nowych i zmianę istniejących klas obiektowych w schemacie NDS,

2) rozszerzeniu schematu podstawowego NDS o nową klasę obiektową *WareHouseDBServer*, zmianie istniejącej klasy *OrganizationalUnit* i utworzeniu ich wystąpień,

3) wykonaniu przykładowego Systemu Zarządzania Magazynem Danych Zbiorczych i Wyszukiwania Informacji, który będąc sparametryzowaną aplikacją obiektową uruchamianą z poziomu podprogramu obsługującego te wystąpienia zapewni dostęp do informacji zgromadzonych we wskazanych archiwalnych bazach danych i Magazynie Danych Zbiorczych.

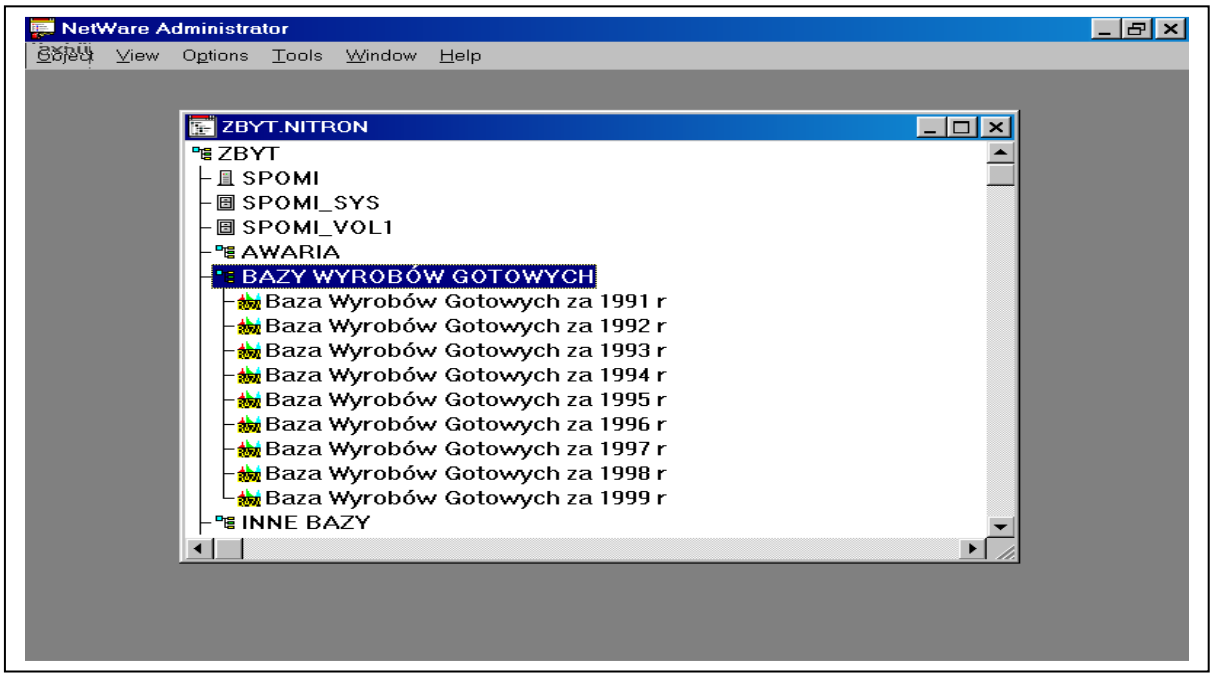
Tak więc w ramach implementacji opisanej metody, rozszerzono schemat NDS-u o nowe atrybuty, pozwalające odwzorować w Informacyjnym Drzewie Katalogowym (DIT) NDS-u schemat Magazynu Danych Zbiorczych. Zaprojektowano i zaimplementowano nową klasę obiektową typu *WareHouseDBServer,* dziedziczącą swoje podstawowe własności z klasy obiektowej typu *DBServer* (typu liść). Następnie zmieniono istniejącą w schemacie podstawowym NDS-u definicję klasy obiektowej *OrganizationalUnit* (typu kontener) poprzez dodanie do niej wcześniej zdefiniowanych - co przedstawiono na rys. 2 - nowych atrybutów, za pomocą których odwzorowano w bazie NDS opis i schemat Magazynu Danych Zbiorczych. Dalej rozszerzono katalogowe drzewo NDS poprzez utworzenie w nim wystąpień nowej klasy obiektowej *WareHouseDBServer* oraz zmienionej klasy *Organizational Unit*. Każde z wystąpień klasy obiektowej *WareHouseDBServer* reprezentuje pewną archiwalną bazę danych zeskładowaną w systemie plików pewnego serwera sieciowego, natomiast wystąpienia zmienionej klasy obiektowej *Organizational Unit* reprezentuje opis pewnego magazynu danych zbiorczych. Na rysunku 3 przedstawiono wystąpienia ww. klas za pomocą odpowiednich ikon. I tak ikona pt. BAZY WYROBÓW GOTOWYCH jest graficznym reprezentantem wystąpienia zmienionej klasy obiektowej *Organizational Unit*, natomiast ikony o nazwach Baza Wyrobów Gotowych za 1991-1999 r. reprezentują wystąpienia nowej klasy obiektowej *WareHouseDBServer*.



Rys. 2. Nowe atrybuty NDS-u i zmieniona definicja klasy obiektowej *Organizational Unit*

Fig. 2. The new NDS attributes and changed *Organizational Unit* object class definition

Następnie, wykorzystując opracowany przykładowy System Zarządzania Magazynem Danych Zbiorczych i Wyszukiwania Informacji, fizycznie utworzono Magazyn Danych Zbiorczych w strukturze plików pewnego serwera sieciowego. Wykorzystano w tym celu pakiet SQLWindows Application Development Module firmy CENTURA (dawniej GUPTA). Końcowa postać schematu tego magazynu o strukturze gwiazdy [6] jest taka, że można udzielić między innymi odpowiedzi na postawione we wprowadzeniu pytania analityczne. Utworzony Magazyn Danych Zbiorczych zasilono zagregowanymi danymi pochodzącymi z poszczególnych archiwalnych baz danych, wykorzystując w tym celu odpowiednio przygotowane pytania SQL oraz technologię ODBC. Pytania te zadawano z poziomu Systemu Zarządzania Magazynem Danych Zbiorczych do wszystkich archiwalnych baz danych, w których znajdywały się istotne – z punktu widzenia przyszłych pytań analitycznych – informacje. Po uzyskaniu odpowiedzi zapisywano ją w Magazynie Danych Zbiorczych.



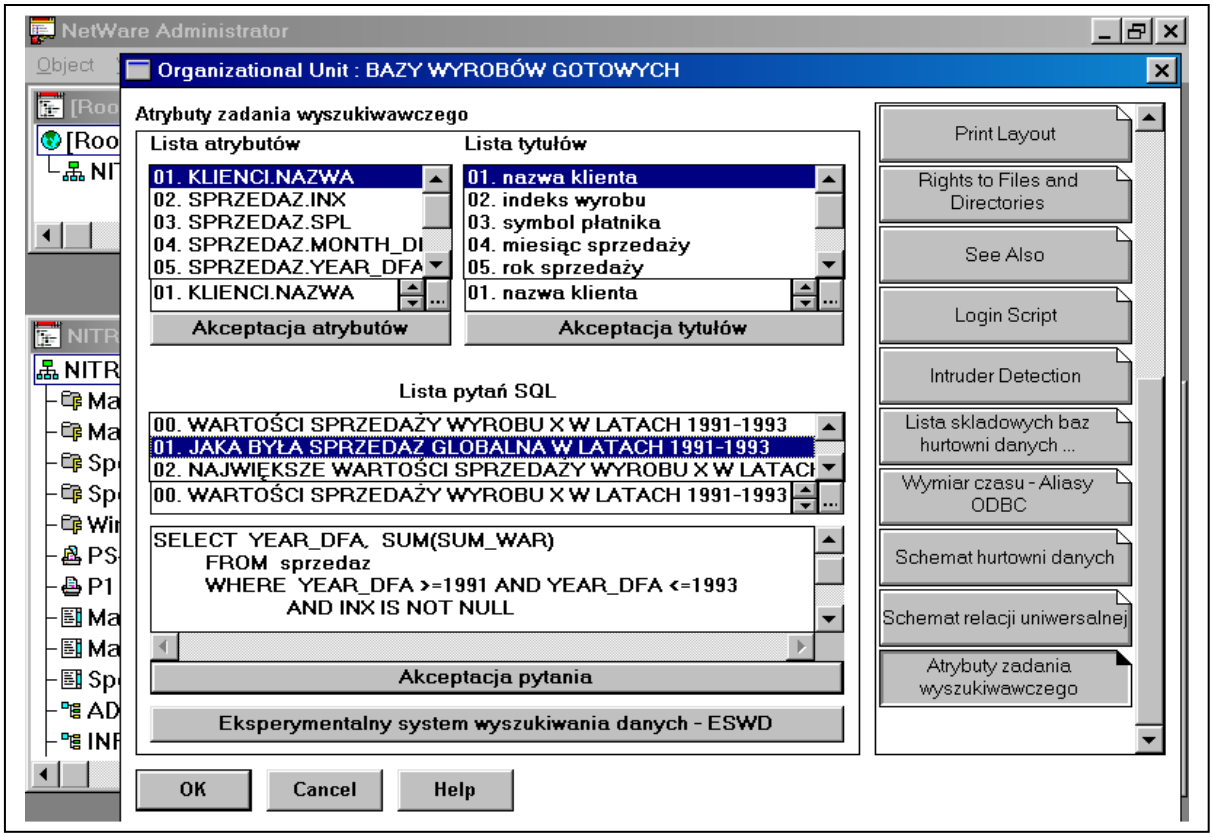
Rys. 3. Wystąpienia zmienionej (*Organizational Unit*) oraz nowych (*WareHouseDBServer*) klas obiektowych

Fig. 3. The changed (*Organizational Unit*) and new (*WareHouseDBServer*) object class instances

Realizację według zaproponowanej metody sygnalizowanych we wprowadzeniu pytań analitycznych dokonywano za pomocą prostego podsystemu wyszukiwania informacji. Umożliwia on sformułowanie lub wygenerowanie względnie odczytanie z pliku tekstowego lub pomocniczej tablicy Magazynu Danych Zbiorczych pytań analitycznych. Pytania te następnie kierowano za jego pomocą do Magazynu Danych Zbiorczych, otrzymując w odpowiedzi odpowiednie zestawienia w postaci tabelarycznej i graficznej.

2.1.1. Odwzorowanie schematu Magazynu Danych Zbiorczych w bazie NDS

Jak już wspomniano, opis archiwalnych baz danych oraz schematu Magazynu Danych Zbiorczych odwzorowano w odpowiednich, wcześniej zdefiniowanych, atrybutach wystąpienia zmienionej klasy obiektowej *Organizational Unit*, zarządzanej poprzez rozszerzony standardowy interfejs użytkownika *NetWare Administrator*. Dostęp do odziedziczonych oraz nowych atrybutów wystąpienia ww. klasy obiektowej uzyskuje się za pomocą tego interfejsu przez wskazanie odpowiedniej zakładki. Stanowi ona pewien fragment kodu, za pośrednictwem którego użytkownik ma możliwość zmiany wartości tych atrybutów i zapisania ich w katalogowej bazie NDS. Jedna zakładka może obsługiwać jeden lub więcej wielowartościowych atrybutów NDS. Tak więc, korzystając z usługi *Snapin* umożliwiającej rozszerzenie standardowego interfejsu użytkownika oraz ww. mechanizmu, utworzono i zapisano w katalogowej bazie NDS niezbędny opis archiwalnych baz danych. W podobny sposób odwzorowano w niej schemat Magazynu Danych Zbiorczych. Przykładowo, na rysunku 4 przedstawiono zakładkę "*Atrybuty zadania wyszukiwawczego*". Służy ona do zapisania w atrybucie NDS-u o nazwie *WareHouse Search Attributes* zmienionej klasy obiektowej *Organizational Unit* listy nazw atrybutów ze schematu Magazynu Danych Zbiorczych, natomiast w atrybucie o nazwie *WareHouse Search Titles* listy ich tytułów.



Rys. 4. Zakładka "*Atrybuty zadania wyszukiwawczego*" rozszerzonego interfejsu użytkownika *NetWare Administrator*

Fig. 4. The extended *NetWare Administrator* user interface "*Search task attributes*" tab

Listy te wykorzystywane są przez sparametryzowany System Zarządzania Magazynem Danych Zbiorczych i Wyszukiwania Informacji do budowy pytań analitycznych. Opcjonalnie z poziomu tej zakładki można zapisać w bazie NDS predefiniowaną dla użytkowników końcowych listę gotowych pytań SQL, kierowanych do Magazynu Danych Zbiorczych za pośrednictwem Systemu Zarządzania Magazynem Danych Zbiorczych i Wyszukiwania Informacji. Z poziomu tej zakładki rozszerzonego interfejsu użytkownika *NetWare Administrator* uruchamiany jest zaimplementowany (zgodnie z rys. 1) przykładowy System Zarządzania Magazynem Danych Zbiorczych i Wyszukiwania Informacji (SZMDZ), którego opis podano w rozdziale 2.3.

2.2. Dynamiczne tworzenie/rozszerzanie schematu Magazynu Danych Zbiorczych

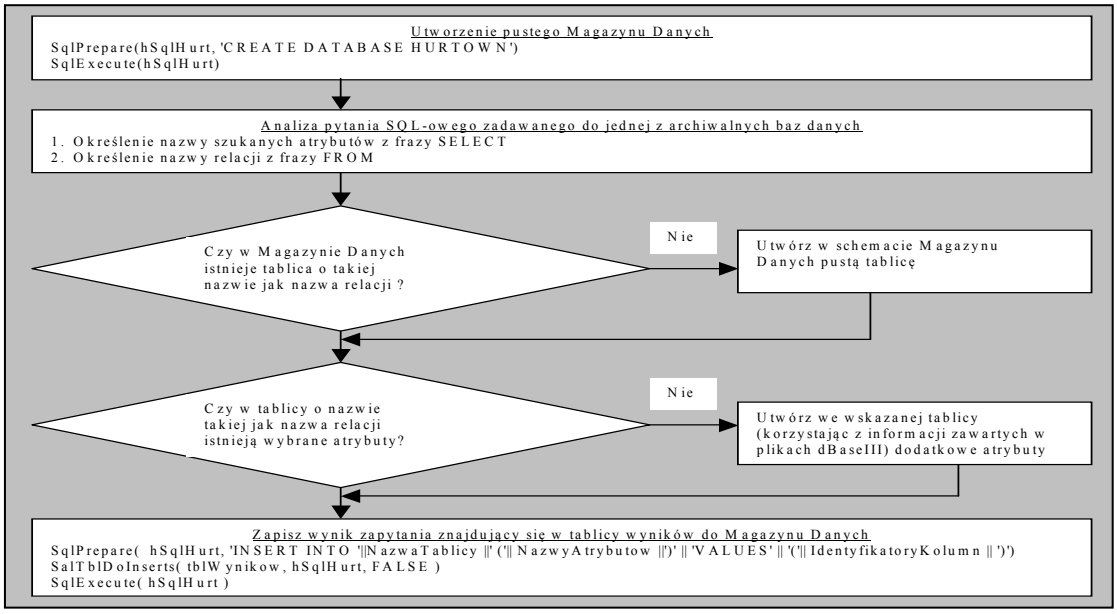
Budowanie hurtowni danych wymaga korzystania z wielu narzędzi. Istniejący w komercyjnych magazynach danych problem dotyczący magazynowania danych, obejmujący proces pozyskiwania, konwersji i integracji danych źródłowych, zwany również ekstrakcją danych, wykonywany jest cyklicznie przez specjalnie opracowane programy. Ekstrakcja danych może być również ręcznie realizowana przez użytkowników systemu [4], na których spoczywa pełna odpowiedzialność za poprawność procesu magazynowania danych i co za tym idzie – za spójność magazynu danych. Jednocześnie, istniejące komercyjne systemy zarządzania bazami danych, wspierające technologię hurtowni danych [6], umożliwiają integrację danych pochodzących z systemów relacyjnych. W procesie integracji danych systemy te wykorzystują wypracowane przez technologię sfederowanych systemów baz danych standardy ODBC, TUXEDO, CORBA, DCE i ODP. Dostarczane przez największych producentów systemów relacyjnych specjalistyczne oprogramowanie ułatwia rozwiązanie ww. problemu. Za pomocą tego oprogramowania dokonywane jest pobranie danych z operacyjnych baz danych, oczyszczenie ich z niejednorodności i niespójności, przekształcenie do postaci wymaganej przez schemat magazynu danych oraz załadowanie ich do hurtowni. Przykładowo, oprogramowanie to w przypadku firmy Oracle (dostarczacej wraz z sieciowym systemem operacyjnym NetWare 4.2 swoją bazę Oracle8) nosi nazwę *Oracle Loader*, *Oracle Transparent Gateway* oraz program *Oracle Open Gateway*. Zwykle przeniesienie danych do hurtowni następuje za pośrednictwem pliku płaskiego zapisywanego przez program pobierania danych. Następnie zawartość tego pliku za pomocą specjalnego programu ładowana jest do hurtowni. W przypadku programu *Oracle Open Gateway* proces pobierania, przekształcania i ładowania danych następuje w jednym kroku [19]. Realizacja tych procesów przez ten program jest możliwa dzięki wcześniej opracowanemu dla hurtowni modelowi danych oraz odwzorowaniu danych operacyjnych na ten model. Opracowanie dla hurtowni modelu danych i odwzorowanie danych operacyjnych dokonuje się za pomocą zaawansowanych narzędzi

projektowych typu Oracle CASE, takimi jak *Oracle Designer/2000*. Narzędzie to przeznaczone jest do projektowania aplikacji, baz oraz hurtowni danych. Umożliwia ono również zapisanie i przetwarzanie metadanych, opisujących strukturę baz i hurtowni danych.

Z drugiej strony istnieją inne znane narzędzia, jak np. wymieniany przez ANSI/ISO standard języka programowania SQL dla relacyjnych baz, który posiada tę własność, że jednoczy język zapytań z językiem zarządzania schematem bazy danych i tym samym ma wpływ na fizyczny projekt bazy [20]. W chwili obecnej większość popularnych obecnie RSZBD implementuje język SQL na wyjściowym poziomie zgodności ze standardem SQL-92. Postanowiono zatem - mimo pewnych ograniczeń - wykorzystać ww. własność do realizacji koncepcji dynamicznego definiowania schematu Magazynu Danych Zbiorczych w trakcie procesu ekstrakcji danych. Do wspomnianych ograniczeń zaliczyć można między innymi ograniczenia i wady języka SQL, w szczególności zaś jego ograniczenia implementacyjne w pakiecie SQLWindows Application Development Module (zgodnego ze specyfikacją Microsoft ODBC ver. 2.0). Tak więc pomimo tych ograniczeń, biorąc jednak pod uwagę wspomnianą klasę pytań analitycznych, możliwe było pobranie z archiwalnych baz niezbędnych danych i załadowanie ich do Magazynu Danych Zbiorczych w dwóch krokach. W kroku pierwszym pobrane dane poddawano wstępnej filtracji i agregowaniu. W kroku drugim, w trakcie procesu zapisywania pobranych danych do przykładowego Magazynu Danych Zbiorczych, dynamicznie tworzono/rozszerzano jego schemat. Podczas dynamicznego tworzenia/rozszerzania tego schematu korzystano ze składni pytań SQL zadawanych do archiwalnych baz danych, jak również z informacji o strukturze plików składowych tych baz [21]. Tak otrzymane dane, będące wynikiem zadanych pytań SQL, poddawano konwersji niektórych typów danych ze schematu archiwalnych baz danych do postaci akceptowanej przez ww. schemat. Następnie, korzystając z poleceń SQL, wstępnie zagregowane i przetworzone dane ładowano do Magazynu Danych Zbiorczych. Takie podejście do problemu projektowania/tworzenia schematu Magazynu Danych Zbiorczych powoduje, że użytkownik końcowy koncentruje się tylko – z punktu widzenia przyszłych pytań analitycznych – na niezbędnych informacjach/danych i ich agregatach, które musi zaimportować z archiwalnych baz danych, aby w rezultacie otrzymać poprawną odpowiedź na zadane później do Magazynu Danych Zbiorczych pytanie analityczne. Tak więc w wyniku zastosowania ww. podejścia możliwe było uzyskanie prostego schematu Magazynu Danych Zbiorczych o strukturze gwiazdy bez potrzeby stosowania do jego projektowania diagramów związków encji ER (ang. Entity Relationship), których użycie w przypadku zastosowań OLAP nie jest już tak naturalne [22] jak dla zastosowań OLTP.

2.2.1. Koncepcja dynamicznego tworzenia schematu Magazynu Danych Zbiorczych

Jak już wspomniano w opisie metody, schemat Magazynu Danych Zbiorczych powinien wynikać z pytań analitycznych. Wspomniano również, że wiedza na temat zbioru pytań analitycznych jest ograniczona. Ponadto nie wiadomo, kiedy pojawią się nowe pytania analityczne sformułowane przez kierownictwo przedsiębiorstwa. Biorąc to pod uwagę zdecydowano, aby w dynamiczny sposób definiować schemat Magazynu Danych Zbiorczych w trakcie procesu ekstrakcji danych z archiwalnych baz danych. Implementację tej koncepcji, opartej na przykładowym algorytmie przedstawionym na rys. 5., zmaterializowano w procedurach mechanizmu dynamicznego tworzenia/rozszerzania schematu, będącego częścią składową przykładowego Systemu Zarządzania Magazynem Danych Zbiorczych.



Rys. 5. Algorytm mechanizmu dynamicznego rozszerzania schematu magazynu danych

Działanie tego prostego algorytmu opiera się na analizie składni pytań SQL zadawanych do archiwalnych baz danych w trakcie ekstrakcji danych, jak również na informacjach dotyczących struktury plików składowych tych baz. W wyniku analizy pytania SQL znana staje się dla systemu nazwa relacji (znajdującej się za frazą FROM) oraz lista szukanych atrybutów (znajdujących się za frazą SELECT). W zależności od wyników analizy System Zarządzania Magazynem Danych Zbiorczych podejmował decyzję o rozszerzeniu jego schematu o nową tablicę lub nowe atrybuty w ramach tej tablicy. Typ nowo tworzonych atrybutów system określa wstępnie na podstawie opracowanej funkcji zwracającej listę atrybutów i ich typów z przeszukiwanego pliku *.DBF [21]. Następnie, jeśli zachodzi konieczność, system automatycznie zmienia ich typ do postaci akceptowanej przez *SQLWindows Application Development Module*. Tak więc, za każdym razem, kiedy w składni kolejnego zadanego pytania SQL pojawia się atrybut, którego jeszcze nie zdefiniowano w

schemacie Magazynu Danych, za pomocą tego mechanizmu w sposób dynamiczny rozszerzany jest schemat Magazynu Danych Zbiorczych w procesie ładowania zagregowanych danych z archiwalnych baz danych.

2.2.2. Realizacja schematu Magazynu Danych Zbiorczych

W celu utworzenia według opisanego podejścia prostego schematu Magazynu Danych Zbiorczych wykorzystano przykładowy System Zarządzania Magazynem Danych Zbiorczych, w którym zaimplementowano mechanizm dynamicznego tworzenia/rozszerzania schematu. Z poziomu tego systemu wykonano kilka SQL-owych pytań, które kierowano do niektórych relacji w składowych archiwalnych bazach danych (krok 1), opisanego w rozdziale 1 szczególnego magazynu danych. Pytania do tych baz kierowano za pośrednictwem zaimplementowanej w pakiecie *SQLWindows Application Development Module* technologii ODBC, zrealizowanej w postaci odpowiednich sterowników ODBC, zarządzanych za pomocą programu zarządzającego *Qest*. Fragmenty schematów tych relacji były postaci:

```
KLIENCI{ SPL        (symbol płatnika),
         NAZWA      (nazwa klienta),
         ... },
WYROBY{ INX         (indeks wyrobu),
        NAZ         (nazwa wyrobu),
        ZD          (zakład produkcyjny),
        CENS        (cena sprzedaży),
        ...},
SPRZEDAŻ{ DFA       (data wystawienia faktury),
          SPL       (symbol płatnika),
          WAR       (wartość faktury),
          ...}.
```

Zadawane pytania SQL podczas ekstrakcji danych były następujące:

1. *Wybierz poprawne dane o klientach*

```
SELECT SPL, NAZWA
    FROM KLIENCI
    WHERE SPL <> "  AND NAZWA <> "
```

2. *Wybierz poprawne dane o wyrobach*

```
SELECT INX, NAZ,  ZD,  CENS
        FROM WYROBY
    WHERE  INX <> " AND NAZ <> "
```

3. *Wartości sprzedaży dla poszczególnych klientów w poszczególnych miesiącach roku ...*

```
SELECT  YEAR(DFA),  MONTH(DFA), SPL, SUM(WAR)
    FROM  SPRZEDAZ
    WHERE  YEAR(DFA)=1991
```

```
        GROUP BY  YEAR(DFA), MONTH(DFA), SPL
```

4. *Wartości sprzedaży dla poszczególnych klientów w roku ...*

```
SELECT YEAR(DFA),  SPL, SUM(WAR)
        FROM  SPRZEDAZ
        WHERE  YEAR(DFA)=1991
        GROUP BY  YEAR(DFA), SPL
```

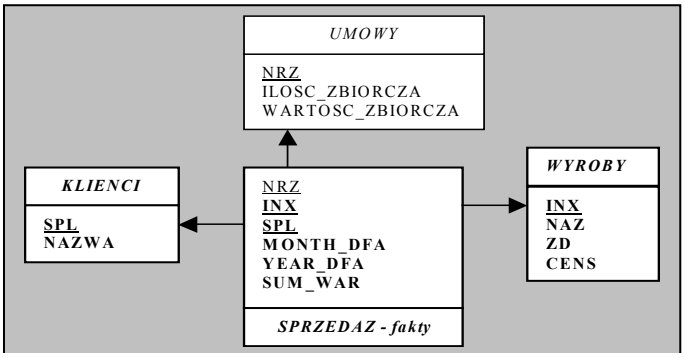
5. *Wartości sprzedaży dla poszczególnych wyrobów w poszczególnych miesiącach roku ...*

```
SELECT  YEAR(DFA),  MONTH(DFA),  INX, SUM(WAR)
        FROM  SPRZEDAZ
        WHERE  YEAR(DFA)=1991
        GROUP BY  YEAR(DFA), MONTH(DFA),  INX
```

6. *Wartości sprzedaży dla poszczególnych wyrobów w roku ...*

```
SELECT  YEAR(DFA),   INX, SUM(WAR)
        FROM  SPRZEDAZ
        WHERE  YEAR(DFA)=1991
        GROUP BY  YEAR(DFA),  INX
```

Następnie (krok 2) wyniki poszczególnych zapytań zapisywano do Magazynu Danych Zbiorczych. W trakcie wykonywania procedury zapisywania wyników rozszerzano schemat Magazynu Danych Zbiorczych, wykorzystując zaimplementowany mechanizm dynamicznego rozszerzania schematu. W końcu możliwe było uzyskanie, co przedstawiono na rys. 6, schematu Magazynu Danych Zbiorczych o strukturze gwiazdy, w którym poziomy wymiaru czasu (lata, rok, miesiące) przechowywane są w relacji faktów, tj. w relacji SPRZEDAZ(INX, SPL, MONTH_DFA, YEAR_DFA, SUM_WAR).



Rys. 6. Przykład dynamicznie wygenerowanego gwiaździstego schematu magazynu danych
Fig. 6. The dynamic generated data warehouse star schema example

Przykładowo, wykonanie poniższego pytania SQL-owego zadanego za pośrednictwem technologii ODBC do jednej z archiwalnych baz danych:

Wartości sprzedaży dla poszczególnych klientów w roku ...

```
SELECT YEAR(DFA),  SPL, SUM(WAR)
        FROM  SPRZEDAZ
        WHERE  YEAR(DFA)=1991
        GROUP BY  YEAR(DFA), SPL
```

a następnie zapisanie otrzymanego rezultatu do Magazynu Danych Zbiorczych, spowoduje rozszerzenie jego schematu o nową tablicę SPRZEDAZ(SPL, YEAR_DFA, SUM_WAR). Wykonanie natomiast i zapisanie wyników do Magazynu Danych Zbiorczych pytania postaci:

Wartości sprzedaży dla poszczególnych wyrobów w poszczególnych miesiącach roku ...

```
SELECT  YEAR(DFA),  MONTH(DFA),  INX, SUM(WAR)
        FROM  SPRZEDAZ
        WHERE  YEAR(DFA)=1991
        GROUP BY  YEAR(DFA), MONTH(DFA),  INX
```

spowoduje dalsze jego rozszerzenie poprzez rozszerzenie wcześniej utworzonej tablicy SPRZEDAZ o nowe atrybuty, tj. MONTH_DFA oraz INX, dając w rezultacie tablicę SPRZEDAZ(INX, SPL, MONTH_DFA, YEAR_DFA, SUM_WAR).

W tym miejscu należy zaznaczyć, że implementacja tego mechanizmu działającego według przykładowego algorytmu z rys. 5, kopiującego strukturę archiwalnej bazy danych i dodającego agregaty, nie gwarantuje utworzenia schematu Magazynu Danych Zbiorczych o poprawnej strukturze gwiazdy w każdym przypadku. W sytuacji, gdy dane podlegające agregowaniu znajdują się w innej niż SPRZEDAŻ relacji, wówczas ich agregaty nie będą zapisane w relacji faktów. Przedstawioną wadę można wyeliminować poprzez implementację nieznacznie zmienionego przedstawionego przykładowego algorytmu. Zmiana ta polega na wprowadzeniu do tego algorytmu nowego bloku decydującego o tym, w jakiej tablicy Magazynu Danych Zbiorczych należy zachowywać zagregowane dane pochodzące z archiwalnych baz danych. Za pomocą tak zmodyfikowanego algorytmu możliwe staje się utworzenie poprawnej struktury Magazynu Danych Zbiorczych typu gwiazdy lub płatka śniegu.

Tak więc korzyści wynikające z proponowanego rozwiązania w przypadku tworzenia prostego Magazynu Danych Zbiorczych są następujące:

1) prostota w obsłudze interfejsu użytkownika, tj.:
   - pobranie zagregowanych danych za pomocą odpowiedniego pytania SQL,
   - zapis danych do magazynu danych,
2) wyeliminowanie procesu projektowania schematu magazynu danych,
3) wyeliminowanie procesu tworzenia plików-deskryptorów [23] przechowujących opis operacyjnej (archiwalnej) bazy danych, tj. połączenia z bazą, tabele, atrybuty i ich typy oraz mapowania tych informacji na schemat magazynu danych.

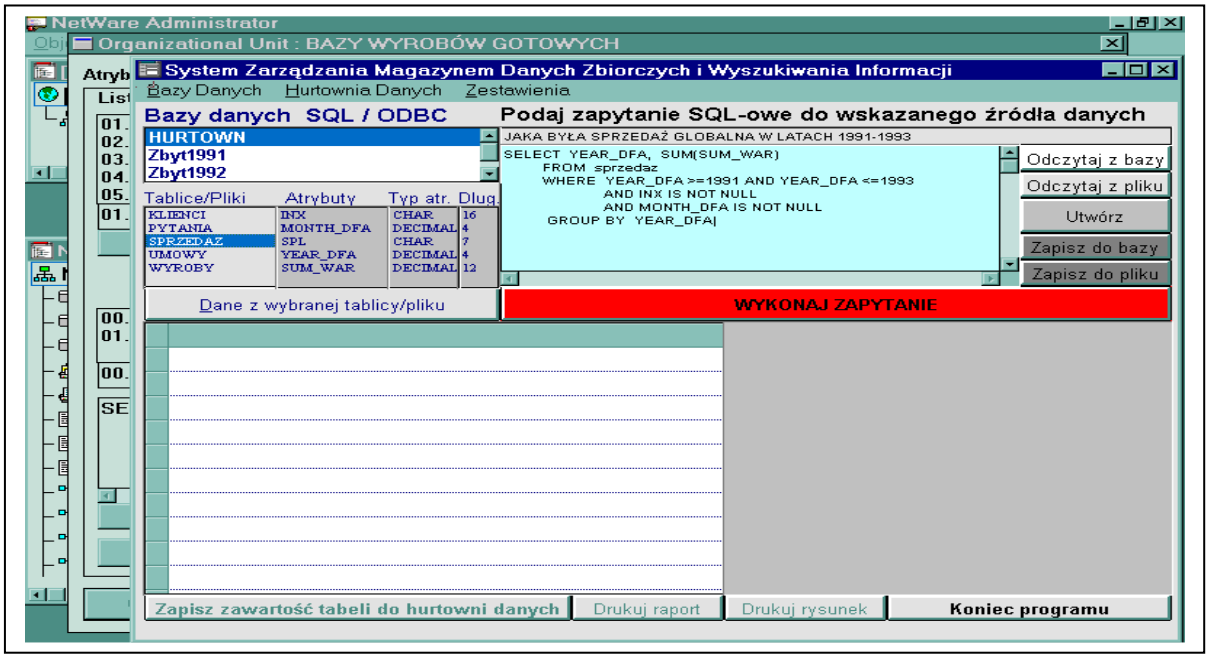
Wady tego rozwiązania, działającego na podstawie przykładowego algorytmu z rys. 5, są zaś następujące:

1. Zaproponowany mechanizm nie gwarantuje utworzenia poprawnego schematu magazynu danych o strukturze gwiazdy w każdym przypadku.

2. Niemożliwe jest wygenerowanie za pomocą zaproponowanego i zaimplementowanego mechanizmu złożonego schematu magazynu danych o strukturze płatka śniegu - w tym z utworzoną hierarchią wymiarów w postaci osobnych tablic.
3. Wygenerowana postać schematu magazynu danych jest całkowicie zależna od schematu archiwalnych baz danych.

2.3. Opis Systemu Zarządzania Magazynem Danych Zbiorczych

Przykładowy System Zarządzania Magazynem Danych Zbiorczych i Wyszukiwania Informacji, który pokazano na rys. 7, zrealizowano wykorzystując pakiet SQLWindows Application Development Module, stworzony przez amerykańską firmę komputerową CENTURA (dawniej GUPTA). SQLWindows Application Development Module jest w pełni obiektowym narzędziem (4GL), którego głównym atutem jest możliwość tworzenia obiektowo zorientowanych programów. Narzędzie to wykorzystuje wstępnie zdefiniowane klasy obiektowe z wszystkimi korzyściami wynikającymi z programowania obiektowego, między innymi dziedziczenie i polimorfizm [7,8,20,24,25]. Pisanie kodu programu odbywa się za pomocą strukturalnego języka SAL (ang. SQLWindows Application Language) [26].



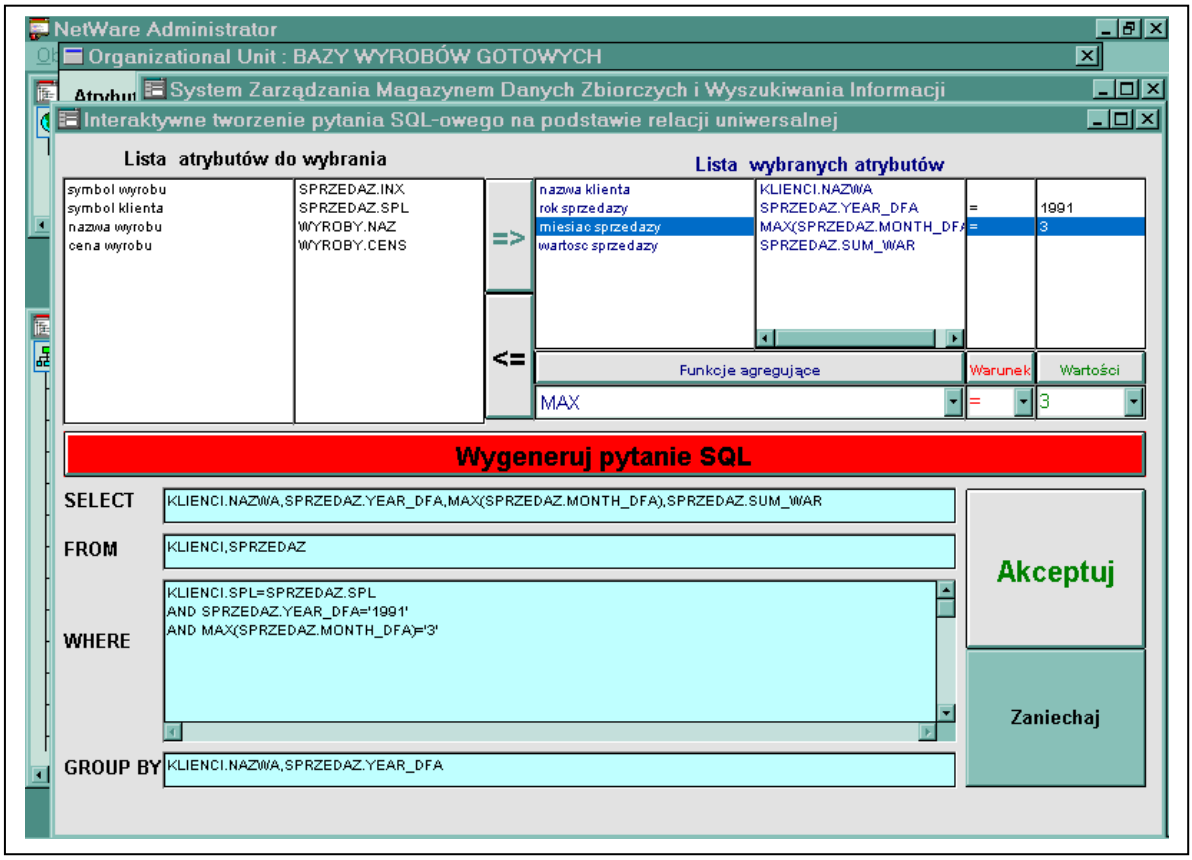
Rys. 7. Interfejs Systemu Zarządzania Magazynem Danych i Wyszukiwania Informacji
Fig. 7. The System Managament Data Warehouse and Information Retrieval Interface

Przedstawiony System Zarządzania Magazynem Danych Zbiorczych i Wyszukiwania Informacji wykonywany jest z poziomu procedur napisanych w języku Borland C++, obsługujących nowe i zmienione wystąpienia klas obiektowych NDS'u. Oprócz niezbędnej funkcjonalności, tj.:

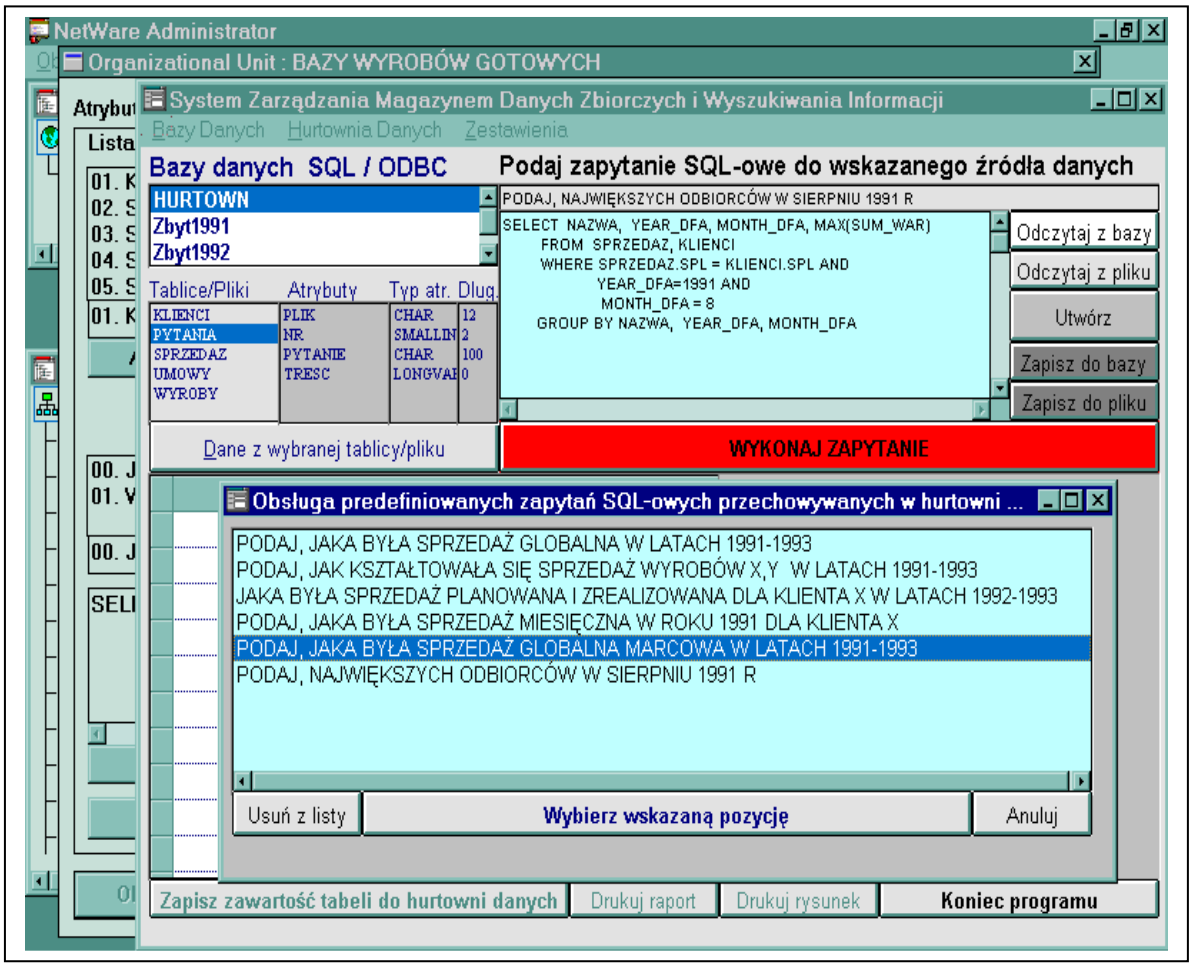
- możliwości importowania - za pośrednictwem języka zapytań SQL i technologii ODBC - z archiwalnych baz danych niezbędnych zagregowanych informacji i zapisywania ich w Magazynie Danych Zbiorczych,
- wykonywania zapytań *ad hoc* i uzyskania jednoznacznej odpowiedzi w postaci tabelarycznej lub graficznej za pomocą zmodyfikowanej klasy obiektowej typu *cQuickGraph*, dostępnej w tym narzędziu programistycznym,

posiada również dodatkowe możliwości ułatwiające wykonanie następujących zadań:

- tworzenie schematu Magazynu Danych Zbiorczych – poprzez dynamiczne jego rozszerzanie (dodawanie tablic lub dodatkowych atrybutów) na podstawie analizy składni zapytania SQL-owego zadawanego do archiwalnych baz danych,
- tworzenie pytań SQL-owych – generowanych za pomocą przykładowego generatora zapytań SQL-owych, kierowanych następnie do Magazynu Danych Zbiorczych (rys. 8),
- odczytywanie, wykonywanie i zachowywanie predefiniowanych pytań SQL-owych zapisanych w pomocniczej tablicy o nazwie 'PYTANIA' Magazynu Danych Zbiorczych (rys. 9) lub w plikach tekstowych,
- prezentowanie informacji ze źródłowych plików danych *.DBF tworzących poszczególne archiwalne bazy danych,
- zarządzanie schematem Magazynu Danych Zbiorczych.



Rys. 8. Interfejs generatora analitycznych pytań SQL
Fig. 8. The analitycal SQL questions generator



Rys. 9. Lista predefiniowanych analitycznych pytań SQL
Fig. 9. The predefinied analitycal SQL questions list

3. Utworzenie oraz eksploatacja Tematycznego Magazynu Danych

3.1. Opis systemu informatycznego w ZTS "Nitron" S.A.

Obecnie eksploatowany system informatyczny w ZTS „Nitron" S.A., działający na osnowie sieci światłowodowej FDDI, pracuje pod kontrolą sieciowego systemu operacyjnego Novell NetWare 4.2. W jego skład wchodzą trzy serwery sieciowe, na których zainstalowano typowe systemy informacyjne służące do zarządzania firmą, w tym podsystem „*Wyroby Gotowe*". Podsystem ten jest aplikacją użytkową, współpracującą z plikami relacyjnej bazy danych typu *dBase* wraz z jego archiwalnymi kopiami, dotyczącymi zamkniętych okresów obliczeniowych za lata 1990-2001, zeskładowanych w oddzielnych katalogach systemu plików pewnego serwera sieciowego.

3.2. Instalacja i konfiguracja utworzonych bibliotek *.DLL typu Snapin

Utworzone według proponowanej metody biblioteki *.DLL typu *snapin*, zawierające niezbędny kod do obsługi nowych i zmienionych klas obiektowych (*dbserver.dll*) oraz kod do obsługi ich wystąpień (*warehous.dll*), wywoływane są oraz inicjalizowane przez rozszerzony program *NetWare Administrator* z tzw. katalogu domowego (*home directory*) użytkownika, integrującego w ten sposób zasoby sieciowe oraz systemy informacyjne działające na podstawie relacyjnych baz danych. Po skonfigurowaniu pozostałych elementów środowiska sieciowego, tj. sterowników ODBC za pomocą programu zarządzającego *Microsoft ODBC Administrator*, jak również po załadowaniu do tematycznego Magazynu Danych Zbiorczych pochodzących z archiwalnych nieaktywnych baz zagregowanych danych, środowisko analityka jest gotowe do pracy, tj. do zadawania pytań analitycznych.

3.3. Realizacja niektórych przykładowych pytań analitycznych

W opisywanej implementacji przykładowego Magazynu Danych Zbiorczych jego schemat wygenerowano w automatyczny sposób. Dokonano tego podczas ładowania Magazynu Danych Zbiorczych zagregowanymi danymi pochodzącymi z archiwalnych baz danych. Pytania SQL-owe kierowano tylko do kilku archiwalnych baz danych (z roku 1991, 1992 oraz 1993). Na rysunkach 10,11,12,13 przedstawiono wyniki pytań analitycznych skierowanych do Magazynu Danych Zbiorczych, uwzględniających również wymiar czasu, tj. lata, rok, miesiąc:

1. Pytanie: *„Jaka była sprzedaż globalna w latach 1991-1993?”* (rys. 10)

```
SELECT  YEAR_DFA,  SUM(SUM_WAR)
      FROM  SPRZEDAZ
      WHERE  YEAR_DFA >= 1991 AND YEAR_DFA <= 1993
                AND INX IS NOT NULL
                AND MONTH_DFA IS NOT NULL
       GROUP BY  YEAR_DFA
```

2. Pytanie: *„Jaka była sprzedaż miesięczna w 1991 roku?”* (rys. 11)

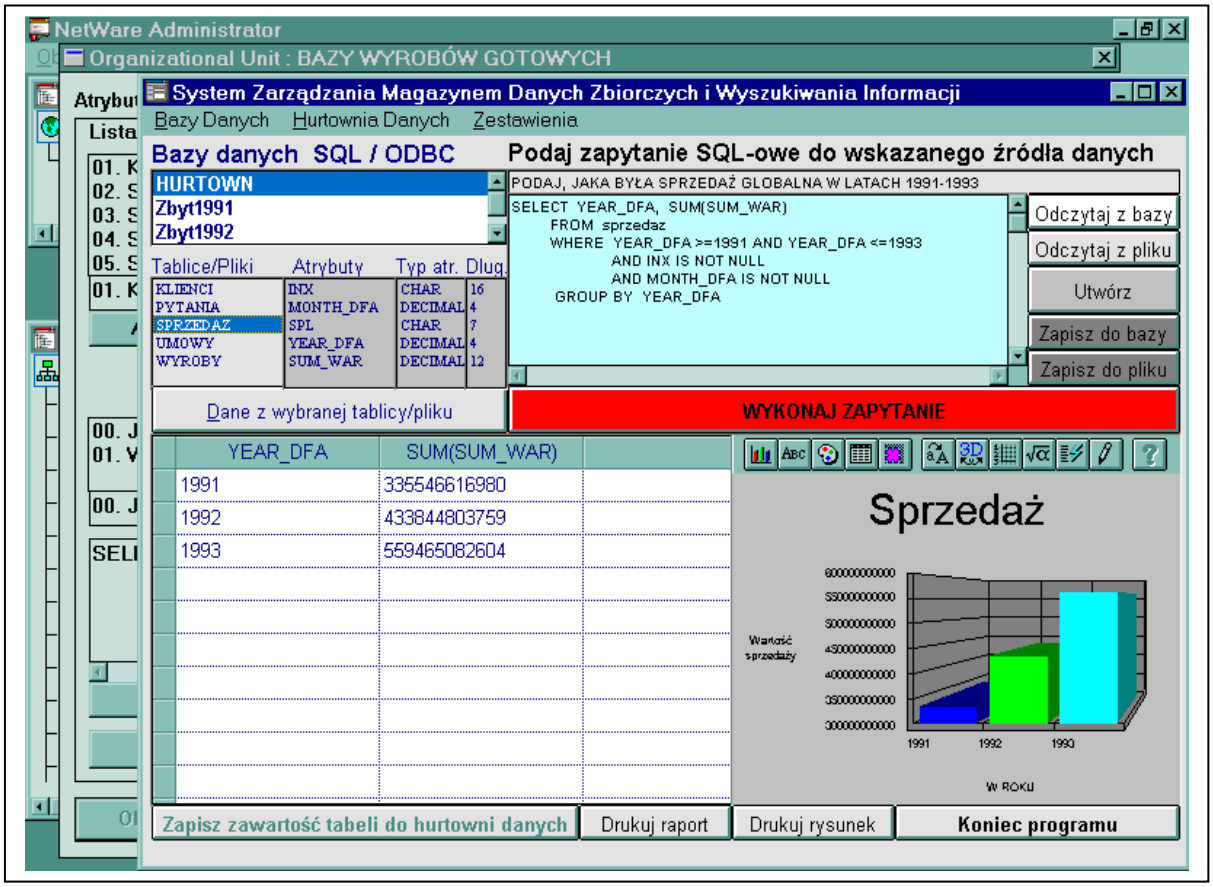
```
SELECT  YEAR_DFA,  MONTH_DFA,  SUM(SUM_WAR)
      FROM  SPRZEDAZ
      WHERE  YEAR_DFA = 1991
                AND INX IS NOT NULL
                AND MONTH_DFA IS NOT NULL
       GROUP BY  YEAR_DFA, MONTH_DFA
```

3. Pytanie: *„Jaka była sprzedaż wyrobów X,Y w latach 1991-1993?”* (rys. 12)

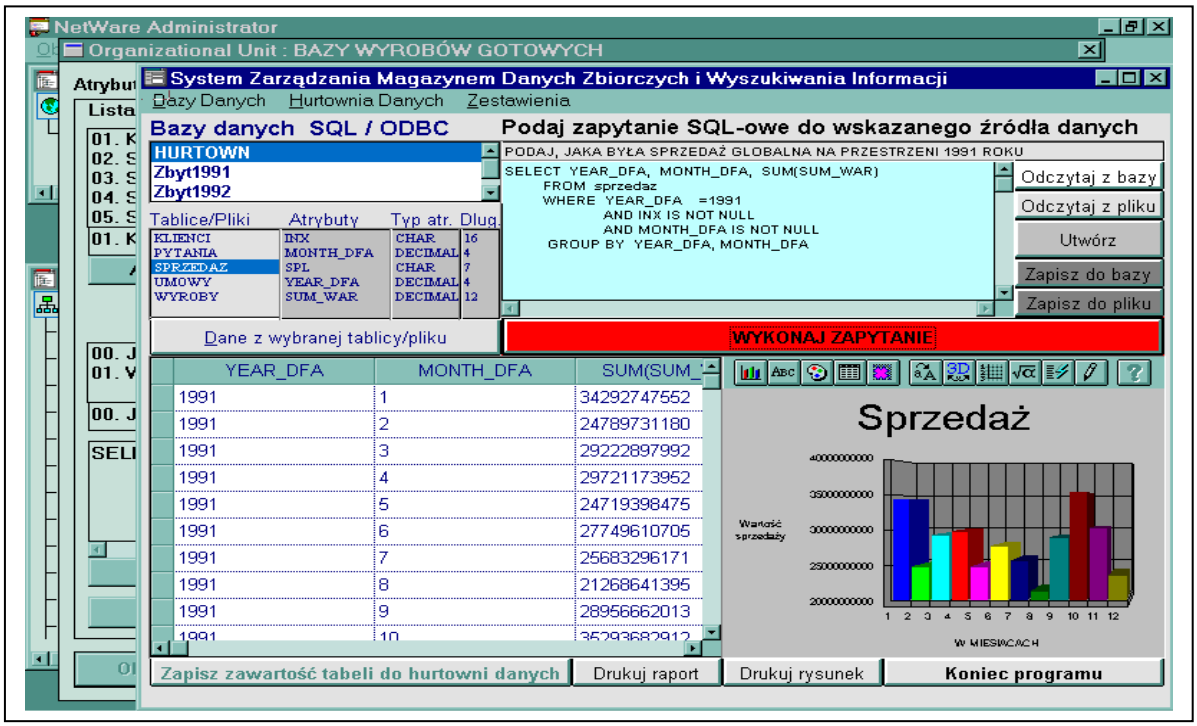
```
SELECT  NAZ, YEAR_DFA, SUM(SUM_WAR)
      FROM  SPRZEDAZ, WYROBY
      WHERE  SPRZEDAZ.INX = WYROBY.INX
            AND SPRZEDAZ.INX IS NOT NULL
            AND (SPRZEDAZ.INX = '1361-281-121-100'  OR
                  SPRZEDAZ.INX = '1361-281-121-000')
            AND YEAR_DFA  >= 1991 AND YEAR_DFA  <= 1993
            AND MONTH_DFA IS NOT NULL
       GROUP BY  NAZ, YEAR_DFA
```

4. Pytanie - *„Jaka była sprzedaż dla klienta X w latach 1991-1993?”* (rys. 13)

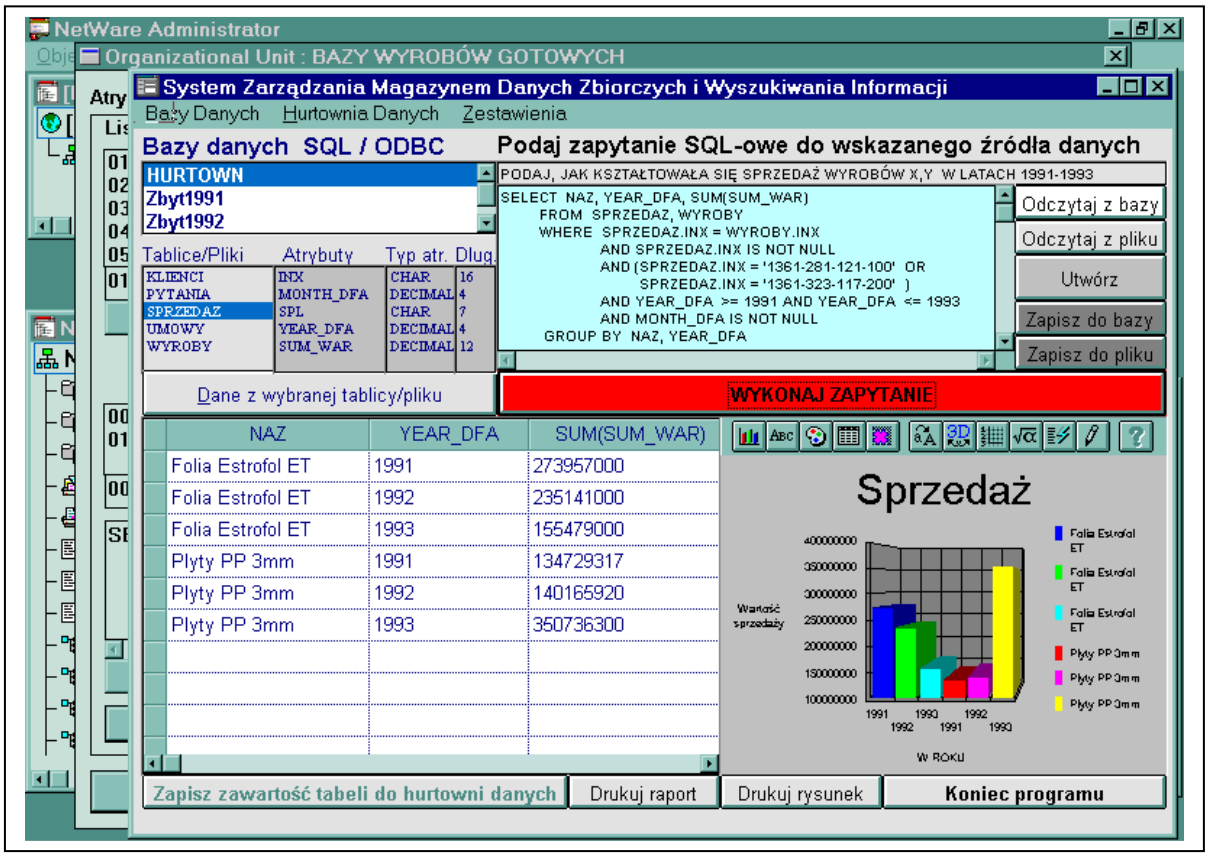
```
SELECT  YEAR_DFA, NAZWA, SUM(SUM_WAR)
      FROM  SPRZEDAZ, KLIENCI
      WHERE  SPRZEDAZ.SPL = KLIENCI.SPL
            AND SPRZEDAZ.SPL IS NOT NULL
            AND SPRZEDAZ.SPL = '2601425'
            AND YEAR_DFA  >= 1991 AND YEAR_DFA  <= 1993
            AND MONTH_DFA IS NOT NULL
       GROUP BY  YEAR_DFA, NAZWA
```



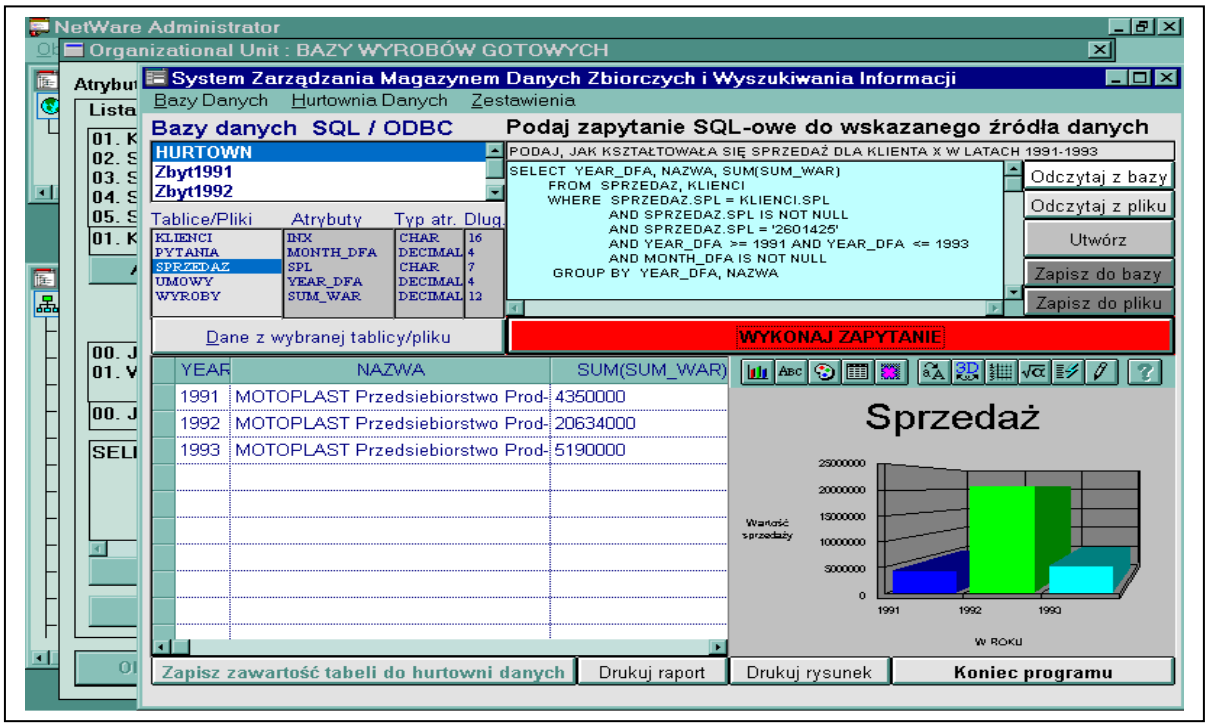
Rys. 10. Pytanie: *Jaka była sprzedaż globalna w latach 1991-1993?*
Fig. 10. Question: *What was the global sales in 1991-1993 years?*



Rys. 11. Pytanie: *Jaka była sprzedaż miesięczna w 1991 roku?*
Fig. 11. Question: *What was the monthly sales in 1991 year?*



Rys. 12. Pytanie: *Jaka była sprzedaż wyrobów X,Y w latach 1991-1993?*
Fig. 12. Question: *What was the X,Y products sales in 1991-1993 years?*



Rys. 13. Pytanie: *Jaka była sprzedaż dla klienta X w latach 1991-1993?*
Fig. 13. Question: *What was the sales for X client in 1991-1993 years?*

4. Podsumowanie

NDS będąca obiektowo zorientowaną implementacją usług katalogowych, umożliwiającą hierarchiczne przedstawienie sieci komputerowej i jej zasobów, jako część systemu operacyjnego NetWare 4.x zyskała sobie od czasu swojej promocji w 1993 r. uznanie i dużą popularność. Jedną z najważniejszych cech katalogowej bazy NDS, jaką jest możliwość rozszerzenia jej schematu, wykorzystano do odwzorowania w drzewie katalogowym NDS opisu relacyjnych baz danych oraz schematu Magazynu Danych Zbiorczych, integrując je wzajemnie za pomocą przedstawionej powyżej metody. Implementacja opisanej metody umożliwia rozwiązanie przedstawionego we wprowadzeniu problemu, jakim jest zadawanie *ad hoc* pytań analitycznych. Wdrożenie tej metody do budowy tematycznych magazynów danych daje następujące korzyści:

- umożliwia ujednolicenie problemu zarządzania dotychczasowymi zasobami sieciowymi oraz informacyjnymi bazami danych traktowanymi jako zasoby sieciowe,
- ułatwia uprawnionym użytkownikom uzyskanie dostępu do informacji zawartych w Magazynie Danych Zbiorczych jak również w relacyjnych archiwalnych bazach danych za pomocą wspólnego interfejsu dostępu do danych,

- ułatwia zarządzanie zasobami sieciowymi oraz dostęp do informacji zawartych w bazach danych z jednego miejsca w sieci komputerowej za pomocą rozszerzonego programu *NetWare Administrator*,
- zapewnia właściwą ochronę informacji zawartych w Magazynie Danych Zbiorczych oraz w plikach składowych archiwalnych baz danych.

W pracy poruszono problem projektowania/tworzenia schematu hurtowni danych. Ogólnie rzecz biorąc, aby można było udzielić odpowiedzi na sygnalizowane we wprowadzeniu pytania analityczne, schemat Magazynu Danych Zbiorczych powinien wynikać z tych pytań. Zaproponowane podejście do problemu projektowania/tworzenia schematu Magazynu Danych Zbiorczych w postaci metody dynamicznego tworzenia/rozszerzania jego schematu powoduje, że użytkownik końcowy koncentruje się tylko – z punktu widzenia przyszłych pytań analitycznych – na niezbędnych informacjach/danych i ich agregatach, które musi zaimportować z archiwalnych baz danych, aby w rezultacie otrzymać poprawną odpowiedź na zadane później do Magazynu Danych Zbiorczych pytanie analityczne. Za pomocą zaimplementowanego w tej metodzie zmodyfikowanego przykładowego algorytmu dynamicznego tworzenia/rozszerzania schematu możliwe jest utworzenie poprawnej struktury Magazynu Danych Zbiorczych typu gwiazdy lub płatka śniegu.

Tak więc korzyści wynikające z proponowanego rozwiązania w przypadku tworzenia prostego Tematycznego Magazynu Danych Zbiorczych są następujące:

1) prostota w obsłudze interfejsu użytkownika,
2) wyeliminowanie procesu projektowania schematu magazynu danych,
3) wyeliminowanie procesu tworzenia plików-deskryptorów,
4) możliwe jest utworzenie poprawnej struktury magazynu danych typu gwiazdy lub płatka śniegu, w przypadku zaimplementowania zmodyfikowanego przykładowego algorytmu dynamicznego tworzenia/rozszerzania schematu.

Wady tego rozwiązania są zaś następujące:

1) wygenerowana postać schematu magazynu danych jest całkowicie zależna od schematu archiwalnych baz danych,
2) wygenerowany schemat magazynu danych pozbawiony jest tego, co człowiek wnosi do procesu projektowania struktury hurtowni, czyli optymalizacji struktury danych,
3) przedstawiona metoda obejmuje tylko przypadki jednoprzebiegowe, tj. gdy pytanie SQL-owe zadane do archiwalnych baz danych generuje oczekiwane wyniki (dane zagregowane); nie zaimplementowano mechanizmu umożliwiającego pozyskanie wyników oczekiwanych po kilku przebiegach,
4) przedstawiona metoda nie gwarantuje wygenerowania schematu magazynu danych wystarczającego dla każdego pytania analitycznego,

5) przedstawiona metoda nie gwarantuje, że schemat magazynu danych nie będzie wymagał zmiany, gdy pojawią się nowe pytania analityczne.

LITERATURA

1. Eaglestone B., Ridley M.: Object databases: An introduction. McGraw-Hill Book Company, England, 1998, ISBN 0-07-709354-2.
2. Igras E.: A framework for query processing in a federated database system: A case Study. URISA (1994), p167-178, copyright Urban and Regional Information Association, http://www.odyssey.maine.edu/gisweb/spatdb/urisa/ur94016.html.
3. Bonjour M., Falquet G.: Concept Bases: A support to Information System Integration. Proceedings of CAiSE*94 Conference, Utrecht, 1994, http://cuiwww.unige.ch/dbresearch/Members/mb/papers/caise/CAISE94_1.html.
4. Koszlajda T.: Technologia Magazynów Danych. III Konferencja użytkowników i developerów, Zakopane, 21-25 października 1997.
5. Grzywak A, Kozielski S., Irek P., Kmonk J., Pierzchała D.: Hurtownie Danych. Projekt Celowy nr 8T11c 009 96C/2995, Politechnika Śląska, Gliwice, wrzesień 1998.
6. Wrembel R.: Dane Hurtowo. Informatyka nr 10, 1998.
7. Atkinson M., Bancilhon F., DeWitt D. Dittrich K., Maier D., Zdonik S.: The Object Oriented Database System Manifesto. In Proceedings of the First International Conference on Deductive and Object-Oriented Databases. Kyoto, Japan, December 1989, http://www.cs.cmu.edu/People/clamen/OODBMS/Manifesto/htManifesto/Manifesto.html.
8. Augustyn D., Stąpor K.: Obiektowo Zorientowany System Zarządzania Bazą Danych O_2. Zeszyty Naukowe Politechniki Śląskiej, s. Informatyka, z. 31, Gliwice 1996.
9. Grzywocz J.: Własności języka zapytań a opis bazy danych. Zeszyty Naukowe Politechniki Śląskiej, s. Informatyka, z. 25, Gliwice 1994.
10. Boruta A., Grzywocz J., Kozielski S.: Wykorzystanie elementów języka naturalnego w systemie wyszukiwania opartym na modelu relacji uniwersalnej. Zeszyty Naukowe Politechniki Śląskiej, s. Informatyka, z. 27, Gliwice 1994.
11. Kozielski S.: Języki zapytań relacyjnych baz danych a rozpraszanie obliczeń w sieci komputerowej. Zeszyty Naukowe Politechniki Śląskiej, s. Informatyka, z. 24, Gliwice 1994.
12. Grzywocz J.: Metody opisu baz danych jako podstawa automatyzacji procesu wyszukiwania. Zeszyty Naukowe Politechniki Śląskiej, s. Informatyka, z. 29, Gliwice 1995.

13. Bok Z.: Integracja procesów zarządzania w zastanych przemysłowych systemach informatycznych. Zeszyty Naukowe Politechniki Śląskiej, s. Informatyka, z. 35, Gliwice 1998.
14. Bok Z.: Analiza modelu logicznego obiektowo zorientowanej bazy NDS w sieciowym systemie operacyjnym Novell NetWare 4.1. Software nr 3, marzec 1998.
15. plik internetowy: http://www.microsoft.com/odbc/default.htm.
16. plik internetowy: DataDirect ODBC Drivers Reference. Intersolv Inc., 9420 Key West Avenue, Rockville, Maryland 20850, printed in the U.S.A. 1996, http://www.intersolve.com.
17. Geiger, Kyle: Inside ODBC. Microsoft Press, 1995.
18. Understanding NetWare Directory Services. NetWare 4.1 Manual Set Chapter 2.
19. Przekształcić dane w decyzje, materiały informacyjne firmy Oracle Polska, Sp. z o.o.
20. Muller R.J.: Bazy danych, język UML w modelowaniu danych. ISBN 83-7279-000-0, wydawnictwo MIKOM, Warszawa, luty 2000.
21. Skowronek M.: Struktury plików danych w relacyjnych bazach danych. Zeszyty Naukowe Politechniki Śląskiej, s. Informatyka, z. 22, Gliwice 1993.
22. Małysiak M.: Technologia OLAP. III Konferencja PLOUG, Zakopane 1997.
23. Grzywak A., Kozielski S., Irek P., Kmonk J., Pierzchała D.: Hurtownie danych. Projekt Celowy nr 8T11C 009 96C/2995, Gliwice 1998.
24. Bok Z.: Analiza modelu wymiany danych pomiędzy bazami i aplikacjami zorientowanymi obiektowo w środowisku Turbo Pascal 7.0 (cz. I). Software nr 11, 1997.
25. Bok Z.: Analiza modelu wymiany danych pomiędzy bazami i aplikacjami zorientowanymi obiektowo w środowisku Turbo Pascal 7.0 (cz. II). Software nr 12, 1997.
26. Bieniek Z.: Programowanie w SQL Windows. Wydawnictwo INFOPLAN, Szczecin-Warszawa 1998, ISBN 83-7240-002-4.
27. plik internetowy: NDS Technical Overview, NetWare 5.0, NDS Version 709, January 1999, http://developer.novell.com/nds/nds_over.pdf.
28. Bucklel J.: NDS Application Development Using C++. Developer Notes, October 1995, Volume 4, Number1, http://developer.novell.com/research/devnotes/1995/january/02/index.htm.
29. Bunnel K.: Accessing and Modifying Information in the NDS Database. Developer Notes, October 1997, Volume 4, Number 10, http://developer.novell.com/research/devnotes/1997/october/04/index.htm.
30. Crossen N., Williams J., Herrin S.: Overview of Novell Directory Services. Developer Notes, March 1997, Volume 4, Number 3, http://developr.novell.com/research/devnotes/1997/march/04/index.htm.

31. Wilson J.: NDS Schema Overview. Developer Notes, October 1998, Vol. 5, Number 10, http://developer.novell.com/research/devnotes/1998/october/06/index.htm.

Recenzent: Dr inż. Marcin Gorawski

Wpłynęło do Redakcji 25 października 2001 r.

Abstract

In this article - on example industrial information system working in ZTS 'Nitron' S.A. factory under network Novell NetWare 4.x operating system control, an integration method of relational *dBase* database systems was presented. This method based on ODBC technology [15,16,17] and extended schema object-oriented NDS database (Novell Directory Services) [13,14,18,27,28,29,30,31] was used to schema definition and subject-oriented data warehouse system implementation, based on universal relation model [9,10,11,12]. In proposed method – schematically presented on fig. 1 - schema data warehouse was defined, imitated in NDS directory tree and implemented as meta data for certain relational database systems, which was saved in NDS object-oriented database from Novell NetWare 4.x operating system as persistent objects. This ROLAP subject-oriented data warehouse schema definition based on universal relation model was used to implement an OLAP simple information retrieval system.