

Anna Pamuła*, Marek Zawada**

PRZEGLĄD PROFESJONALNYCH SYSTEMÓW
ZARZĄDZANIA RELACYJNYMI BAZAMI DANYCH

Systemy zarządzania bazami danych, których istotą jest tworzenie i zarządzanie wielkimi zbiorami informacji, wykorzystywane były głównie na dużych komputerach. Obserwowany w ostatnich latach bardzo szybki wzrost popularności mikrokomputerów spowodował pojawienie się wielu mikrokomputerowych systemów zarządzania bazami danych, wśród których główną uwagę zwracają systemy relacyjne. Chociaż na obecnym etapie rozwoju nie realizują one jeszcze wszystkich funkcji relacyjnych baz danych z dużych komputerów, zauważalny jest stały wzrost ich możliwości. Szczególnie interesującym zjawiskiem w tej dziedzinie jest pojawienie się języków czwartej generacji (4GL), a w tym potencjalnego standardu strukturalnego języka zapytań SQL [3]. Za korzystaniem z języków czwartej generacji w chwili obecnej przemawiają dwa powody: po pierwsze możliwość programowania przez użytkowników końcowych (nie będących profesjonalnymi informatykami); po drugie większe możliwości dla profesjonalnych programistów, których zadaniem jest tworzenie programów użytkowych. Języki czwartej generacji muszą umieć obsługiwać zbiory rozproszonej bazy danych, przetwarzać informacje, czuwać nad sprawną komunikacją w takiej bazie, posiadać funkcje obsługi użytkownika końcowego oraz w szybki i łatwy sposób pozwalać na budowę własnych programów aplikacyjnych. Obecnie, kiedy pojawiają się coraz to nowe, bardziej użyteczne narzędzia pomagające szybko tworzyć wysokiej jakości produkty, języki czwartej generacji pomagają użytkownikom końcowym i programi-

* Mgr, asystent w Katedrze Informatyki UŁ.

** Mgr, elektronik Katedry Informatyki UŁ.

stom zrozumieć proces strukturalnego projektowania i programowania oraz przystosować się do niego już na początku pracy.

Według amerykańskiego matematyka E. F. Codd'a, twórcy języka SQL baza danych jest relacyjna, jeśli spełnia:

- podstawową zasadę relacyjności, czyli operowanie na dwuwymiarowych plikach płaskich (relacjach) z użyciem katalogu systemowego,
- co najmniej 6 z tzw. reguł Codd'a.

Reguły Codd'a odnoszą się głównie do struktury relacyjnego modelu, integralności bazy danych, zdolności do manipulowania danymi oraz podstawowego projektu relacyjnej bazy danych. Reguły Codd'a dotyczące bazy danych są następujące:

1. Zasada przepływu informacji (ang. information rule) - każda żądana informacja może być, z punktu widzenia budowy systemu, udostępniona użytkownikowi.
2. Zasada gwarancji dostępu (ang. guaranteed access rule) - wiąże się z tajnością danych oraz ich zabezpieczeniem przed zniszczeniem lub nieupoważnionym dostępem oraz przed wzajemnym nałożeniem się. Dane muszą mieć zapewnioną kontrolę uprawnień (nawet na poziomie pola, gdyż system operacyjny zabezpiecza ochroną na poziomie zbioru, wystarczy jednak określić możliwość dostępu do relacji).
3. Zasada systematycznego stosowania wartości Null (ang. systematic treatment of Null Values rule) - wiąże się z możliwością odtworzenia bazy danych po wystąpieniu błędu. Wartość Null reprezentuje dane utracone lub nieznanne.
4. Zasada katalogu systemowego o bezpośrednim dostępie (ang. dynamic on-line catalog rule) - katalog ten przechowuje informacje w tablicach i jest dostępny przy pomocy języka relacyjnego.
5. Zasada języka manipulacji danymi (comprehensive data sublanguage rule) - kompletność języka relacyjnego.
6. Zasada stałego podtrzymywania aktualnych parametrów otoczenia zawartych w instrukcji VIEW (ang. view updating rule) - możliwość stałej aktualizacji parametrów otoczenia za pomocą języka relacyjnego.
7. Zasada wyższego poziomu poleceń INSERT, UPDATE, DELETE (ang. high level INSERT, UPDATE, DELETE) - umożliwia wykonanie operacji zmiany zawartości pamięci.
8. Zasada fizycznej niezależności danych (ang. physical data independence).

9. Zasada logicznej niezależności danych (ang. logical data independence) - Zasady 8 i 9 powodują uniezależnienie danych od programów. Zapewniają one możliwość zmiany struktury fizycznej globalnego opisu danych i strategii dostępu bez zmian w programach zastosowań oraz możliwość zmian w programach użytkowych bez konieczności zmian w bazie.

10. Zasada zapewnienia integralności bazy (ang. integrity independence) - zapewnia wewnętrzną zgodność bazy danych poprzez utrzymywanie danych w jednakowym stanie aktualności, zapobieganie redundancji i przeprowadzaniu kontroli: poprawności wprowadzanych danych, aktualizacji, równoległego działania i innych.

11. Niezależność od rozproszenia bazy (ang. distribution independence) - w systemach rozproszonych baz danych (rozproszona baza danych to kolekcja danych, które logicznie należą do tego samego systemu, ale są rozmieszczone w różnych punktach sieci komputerowej) programy użytkowe bez dokonywania zmian mogą być przenoszone w różne punkty sieci.

12. Kompatybilność pomiędzy wersjami (ang. non-subversion rule).

Reguły Codd'a stwierdzają, że system, który rości sobie prawo do bycia systemem relacyjnym musi całkowicie zarządzać bazą danych właśnie poprzez swoje relacyjne możliwości. Użytkownik takiego systemu nigdy nie powinien być zmuszony do używania innego narzędzia poza językiem relacyjnym. System zarządzania relacyjną bazą danych powinien posiadać systemowy katalog, który przechowuje informacje w tablicach. Katalog ten musi być dostępny dla języka relacyjnego. Język relacyjny powinien być kompletny i wyposażony we wszystkie operatory relacji oraz posiadać pełny tryb wstawiania, aktualizowania, funkcje poprawiania i zabezpieczania transakcji. W modelu relacyjnym dostęp do danych powinien następować przez podanie nazwy wiersza (rekordu), klucza głównego (identyfikatora rekordu - każde wystąpienie rekordu ma unikalny identyfikator) i nazwy kolumny (pola). Bez klucza głównego dostęp do poszczególnego wystąpienia danych nie może być zapewniony. Musi istnieć wartość Null, która reprezentuje zgubione i nieznanne wartości danych. Język bazy danych powinien mieć możliwość definiowania wymogów integralności i przechowywania tych informacji w katalogu systemowym [4].

Należy zauważyć, że większość baz danych dostępnych dzisiaj na rynku nie spełnia tych wymagań, w związku z tym nie powinny być one uważane za relacyjne bazy danych. Pełna zgodność z relacyjnym mode-

lem Codda znacznie upraszcza modelowanie aplikacji, a także dalszy rozwój tego oprogramowania i jego obsługę.

Przy poszukiwaniu prawdziwie relacyjnej bazy danych dowiedziano, że bazy danych oparte na SQL (jego rozszerzonej wersji IBM SQL) spełniają 7 z 12 reguł Codda, tak więc spełniają teoretyczny warunek relacyjności.

SQL (ang. Structural Query Language) został stworzony na początku lat siedemdziesiątych i od tamtego czasu powstało wiele jego odmian i różnych wersji na różne komputery i systemy operacyjne. Obecnie liczą się praktycznie dwie wersje jego języka:

- ANSI SQL (poziom 1),
- IBM SQL (poziom 2).

IBM SQL jest wersją bogatszą, zawierającą specjalny katalog systemowy oraz dodatkowe komendy służące do:

- manipulacji danymi, czasem, datą, łańcuchami,
- zmiany haseł i przywilejów,
- składowania i zmiany tablic, indeksów i ekranów.

SQL jest językiem nieproceduralnym, zorientowanym na bazy danych. Pozwala to użytkownikowi na precyzowanie CO ma być zrobione, a nie JAK ma być zrobione [5]. Dane są przetwarzane w całych zestawach, a nie jako pojedyncze rekordy, jak ma to miejsce w językach proceduralnych. Języki nieproceduralne są bardziej efektywne i będą one prawdopodobnie dominujące w niedalekiej przyszłości. Główną siłą SQL jest więc fakt, że działa on na zbiorach danych. Ponieważ zdefiniowany jest przez matematyczne reguły relacji, nie potrzebuje żadnych nowych konstrukcji do rozwiązywania problemów zarządzania.

SQL oszczędza czas, jest bardzo przejrzysty i zwięzły. Posiada tylko 12 komend i jest intensywnie eksploatowany od dłuższego czasu na dużych komputerach. Jest to język równie użyteczny dla użytkownika, jak i dla programisty. Jego wielką zaletą jest fakt, że użytkownik i programista mogą pracować w tym samym trybie języka.

Użytkownik końcowy może wykonywać komendy SQL bezpośrednio bądź przez specjalny interfejs programowy (ang. front-end), który generuje komendy SQL. Dzięki takiemu "pośrednikowi" przez interaktywne wprowadzanie, modyfikację, przeglądanie tekstów i danych użytkownik końcowy ma możliwość definiowania i manipulacji na obiektach wyższego poziomu.

Wyróżnia się trzy typy takich interfejsów "front-end":

- interfejs kierowany przez menu,
- interfejs "Query-by-example",
- interfejs w języku angielskim.

Interfejsy w języku angielskim są praktycznie niespotykane i uważane są za pierwszy krok do języków piątej generacji.

SQL składa się z trzech komponentów:

- DML (ang. Data manipulation language) - język manipulacji danymi,
- DDL (ang. data definition language) - język definicji danych,
- DCL (ang. Data control language) - język kontroli danych.

SQL IBM zawiera także katalog systemowy oraz posiada mechanizm optymalizacji przetwarzania. Te dwa ostatnie mechanizmy są właściwie esencją każdego prawdziwego systemu zarządzania relacyjną bazą danych.

DML steruje następującymi funkcjami: wyszukiwaniem, wstawianiem, poprawianiem i kasowaniem rekordów i pól.

DDL tworzy tablice, indeksy i ekrany.

DCL zapewnia bezpieczeństwo zbiorów, kontroluje dostęp do zbiorów i ustala przywileje dostępu do zbiorów.

Katalog systemowy zawiera definicję obiektów stworzonych przez DDL. Definicje te przechowywane są w tablicach i dzięki nim można bardzo szybko i efektywnie poruszać się po bazie danych. Katalog odgrywa rolę administratora bazy danych. Jego znaczenie wzrasta przy rozproszonej bazie danych.

Mechanizm optymalizacji wykorzystywany przez język SQL jest narzędziem pochodzącym z dziedziny sztucznej inteligencji. Jest szczególnie przydatny, jeśli baza danych jest często poprawiana i uzupełniana. W wyniku działania optymalizuje przetwarzanie na jasne i proste dla użytkownika. SQL nie dostarcza użytkownikowi bezpośredniej możliwości odwołania się do indeksu, ponieważ podstawową zasadą relacyjnej bazy danych jest niezależność fizyczna i logiczna danych. Jeśli program optymalizujący zdecyduje, że w danej operacji będzie lepiej używać indeksu, to automatycznie używa go. Jeśli indeks nie jest utworzony, wybiera najlepszy sposób aktualizacji lub poprawiania danych. Program ten może decydować, które indeksy są potrzebne, może również kierować zapytania do katalogu systemowego (np. o liczbę wierszy w tablicy). Powoduje także dostosowywanie się programów do zmian zachodzących w danych.

SQL jest językiem, który rozwiązuje wiele problemów związanych z podziałem czasu i ochroną zasobów w systemach wielodostępnych, sieciowych (typu LAN, WAN i innych) oraz w systemach rozproszonych baz danych, gdyż posiada mechanizmy zabezpieczenia danych i odzyskiwania danych w przypadku awarii systemu. Do tego celu tworzony jest zbiór zawierający historię wszystkich operacji dokonywanych w bazie danych. Innym problemem związanym z integralnością bazy jest tzw. record locking (zablokowanie rekordu). SQL posiada system automatycznego blokowania rekordu, dzięki czemu zapobiega sytuacjom, gdy dwa różne programy (lub więcej) jednocześnie aktualizują ten sam rekord i jeden niszczy zmiany wprowadzane przez drugi. System blokowania w SQL jest kompleksowy, ponieważ pozwala wielu programom pracować na tym samym rekordzie w różnym trybie, w tym samym czasie. Program zarządzający blokowaniem wykrywa też sytuację zakleszczenia (ang. deadlock), gdy dwa programy czekają wzajemnie na przerwaniu (obydwa są w stanie oczekiwania). Po wykryciu takiej sytuacji wybiera jeden program do przerwania, a pozostałym przesyła informację o przyczynie zakleszczenia. Niektóre systemy dostępne na rynku nie zawierają mechanizmów blokowania. Zezwalają one aplikacjom na czytanie rekordu, podczas gdy program zarządzający danymi oczekuje na zmiany z procesu aktualizacji od innego użytkownika; jeśli rekord został zmieniony w czasie gdy rozpoczęło się czytanie, to odpowiednia aktualizacja wraz z wiadomością jest przesyłana tylko do operatora. Takie rozwiązanie prowadzić może jednak do podejmowania decyzji na podstawie nieaktualnych danych.

Ważną cechą języka SQL-a jako standardu jest jego kompatybilność. Programy napisane w SQL dają się łatwo przenosić na coraz nowsze wersje pakietów (także wersje sieciowe), co gwarantuje im długowieczność. Język SQL, jako jedno z narzędzi do tworzenia relacyjnych baz danych, został wybrany do stworzenia systemu nazwanego SAA (ang. System Application Architecture). Podstawowe założenia tej koncepcji (opracowanej przez IBM) można sprowadzić do formuły, że program napisany i opracowany na dowolnej maszynie, spośród PS/2, PC, 370 może być także na dowolnej maszynie wykonany. W ramach tej koncepcji zostały wybrane także inne narzędzia języków 4GL, np. QMF (ang. Query Management Facilities).

SQL i inne narzędzia języków 4GL są szeroko stosowane w profesjonalnych systemach relacyjnych baz danych (RDBS). Bazy danych,

które nie stosują tych narzędzi nie liczą się na rynku i nie są w pełni relacyjnymi bazami danych, gdyż najczęściej nie spełniają one nawet połowy reguł Codd'a dotyczących modelu relacyjnej bazy danych. Poniżej przedstawiono krótką analizę niektórych, dostępnych obecnie na rynku i najpopularniejszych systemów baz danych.

ORACLE 5.1. System ten wywodzi się z programu ORACLE stosowanego na dużych komputerach (DEC VM/CMS, MVS). Choć zawiera wiele nowych możliwości, nie rezygnuje z rozwiązań wykorzystywanych przez bazę danych z dużych komputerów. ORACLE 5.1 używa języka SQL i innych narzędzi języków 4GL, dzięki czemu ma o wiele więcej możliwości niż przeciętna baza danych [6]. ORACLE 5.1 działa na komputerach IBM PC AT, PS/2 i na komputerach opartych na procesorze 80386. Program ten wymaga pamięci rozszerzonej powyżej 1MB oraz twardego dysku. Może być uruchamiany na 100% klonach IBM (Compaq), ale mogą wystąpić kłopoty z innymi komputerami z powodu wrażliwości ORACLE na działanie BIOS-u.

ORACLE 5.1 zawiera wiele mechanizmów do tworzenia programów aplikacyjnych: SQL Plus, SQL Forms, SQL Report, SQL Calc, Proc C.

SQL Plus jest prostym, szybkim interpretacyjnym językiem używanym do formułowania i wykonywania zapytań języka SQL, nie zawiera niestety mechanizmu przesuwania tekstu na ekranie.

SQL Forms jest narzędziem nieproceduralnym. Zawiera moduły (ang. screen painter) ułatwiające projektowanie plansz ekranowych oraz zbiory komend języka SQL i innych instrukcji służących do sterowania, wprowadzania i aktualizowania danych (tzw. "triggers"). Powszechnie zbiory komend i instrukcji mogą dotyczyć takich obiektów jak pola, grupy pól lub cała plansza. Mogą być uaktywniane przy wejściu do lub wyjściu z powiązanego z nimi obiektu, przy zmianie tego obiektu lub inicjacji zapytania dotyczącego obiektu. Ponieważ komendy zawarte w tych zbiorach mogą dotyczyć innych obiektów, jeden zbiór może uruchamiać następne zbiory komend i instrukcji.

Stosowanie takich zbiorów komend i instrukcji jest interesującą cechą charakterystyczną ORACLE 5.1 i stanowi rozwiązanie sterowania logiką programu, alternatywne w stosunku do metody łączenia SQL z komendami typu IF... THEN... ELSE lub stosowania pełnego języka strukturalnego.

SQL Report jest bardzo pomocnym mechanizmem do projektowania raportów wyjściowych, ale ma pewne ograniczenia uniemożliwiające two-

rzenie skomplikowanych raportów. Twórcy ORACLE 5.1 zapowiadają opracowanie bardziej funkcjonalnej wersji tego narzędzia.

SQL Calc jest zintegrowanym arkuszem kalkulacyjnym, który może mieć dostęp do bazy danych. Firma ORACLE opracowała także interfejs dla użytkowników, którzy chcą wykorzystać zbiory pakietu LOTUS 1-2-3 w ORACLE 5.1, [2].

Proc C jest narzędziem umożliwiającym umieszczenie instrukcji języka SQL bezpośrednio w kodzie źródłowym języka C.

ORACLE 5.1 jest także wyposażony w mechanizm wykrywania błędów. Rozpoznaje on dwie grupy błędów występujących w programach aplikacyjnych: pojawiające się bez odniesienia do systemu operacyjnego i te, które są związane z odwołaniem do systemu operacyjnego. Błędy są ponumerowane i sklasyfikowane w 40 grupach. Do każdego błędu jest podawany komunikat mówiący o przyczynie powstania błędu oraz zalecane działanie.

Na szczególne podkreślenie zasługuje system zabezpieczenia danych. Jest on bardzo rozbudowany. Poza zabezpieczeniem za pomocą haseł istnieje pełen zapis dostępu i prób dostępu do bazy danych przez różnych użytkowników.

Przy pracy z ORACLE w dużej sieci wersja sieciowa Networkstation ORACLE zezwala na połączenie pomiędzy aplikacjami na PC z bazą danych ORACLE rezydującą na minikomputerze. Bardzo ciekawym produktem firmy ORACLE jest SQL Star, baza danych, która działa przy pewnych ograniczeniach zarówno na dużym komputerze, jak i na mikrokomputerze PC [6].

W ostatnim czasie firma ORACLE wypuściła na rynek specjalną wersję servera relacyjnej bazy danych ORACLE na komputery PC/AT i 80386 pod systemem XENIX.

Podsumowując można stwierdzić, że ORACLE 5.1 jest doskonałą bazą danych dla programistów mających doświadczenie w pracy na dużych komputerach i chcących implementować systemy zarządzania bazą danych na komputery PC w sieciach LAN. Efektywne wykorzystanie profesjonalnych możliwości ORACLE 5.1 wymaga dużego doświadczenia programowego.

INGRES 5.0. Podobnie jak system zarządzania bazą danych ORACLE, INGRES 5.0 jest jednym z najczęściej używanych na dużych komputerach. Oba programy ostatnio zaimplementowano na komputery klasy PC.

Korzystanie z INGRES 5.0 związane jest z dość dużymi wymagania-

mi sprzętowymi: ok. 10 MB pamięci dysku twardego, co najmniej 640 KB pamięci RAM, komputera klasy AT lub 80386. Z tego względu polecany jest dla dużych systemów informatycznych. INGRES spełnia w całości 6 (1, 4, 7, 8, 11 i 12) a częściowo 3 (2, 5 i 9) z warunków Coda.

Liczba zbiorów w bazie danych i rekordów w zbiorze jest ograniczona tylko ilością wolnego miejsca na dysku. Każdy rekord może zawierać do 127 pól o długości do 2000 bajtów. INGRES korzysta z 8 typów danych (znakowego, pięciu typów numerycznych, typu daty - DATE i typu pieniężnego - MONEY). Nie posiada wartości Null dla oznaczenia zgubionych lub nieznanymi wartości danych.

Katalog systemowy pracuje znakomicie wewnątrz systemu, ale nie jest zbyt komunikatywny dla użytkownika.

INGRES 5.0 jest pakietem bardzo łatwym w użyciu dzięki sterowaniu za pomocą menu i możliwości korzystania z bazy programów wspomagających. Użytkownicy końcowi nie muszą konieczności używać komend języka SQL. Zapewnia to moduł QBF (ang. Query by Forms) [10], który generuje odpowiednie plansze na ekranie, pozwalające użytkownikom dodawać, usuwać, zmieniać, przeglądać dane. Moduł QBF wykorzystuje metodę QBE (ang. Query by Example), która polega na tym, że użytkownik podaje w odpowiednich polach planszy tzw. wzorce (ang. examples), wskazując w ten sposób pola, których zawartości go interesują. Jednocześnie podaje w wybranych polach kryteria wyszukiwania (odnoszące się do tych pól). Uzupełnieniem opcji QBF jest ViFRED (ang. Visual Form Editor), który umożliwia dostosowywanie generowanych plansz do konkretnych zastosowań, np. nadawanie polom wartości początkowych, wykonywanie procedur kontroli poprawności wprowadzonych danych, czy określenia ścieżki kursora [4].

Do generowania raportów wyjściowych dla konkretnych zastosowań służy opcja Report Writer. Dysponuje ona własnym językiem do definiowania postaci raportów. Dla utworzenia skomplikowanych zastosowań programiści mogą także wykorzystywać język czwartej generacji (INGRES 4GL). Używając tego niestukturalnego języka posiadającego wiele funkcji, programiści precyzują jedynie cele, które chcą osiągnąć, nie precyzując poszczególnych kroków prowadzących do osiągnięcia celu. INGRES 5.0 zapewnia również automatyczne odzyskiwanie danych w wypadku awarii systemowej. Jest systemem znacznie silniejszym od dBASE, R:base i podobnych. Oprócz języka SQL posiada też

inny niestandardowy język QUEL (ang. QUERy Language), posiada też dobry program optymalizujący.

Duże możliwości tej bazy danych potwierdzają także bardzo dobre wyniki testów szybkościowych. Pomimo tak olbrzymich możliwości zapowiadana jest ulepszona wersja INGRES 6.0, która będzie mogła współpracować także z systemem operacyjnym OS/2.

INFORMIX SQL i 4GL. INFORMIX SQL 2.1 składa się z trzech podstawowych komponentów: języka SQL, bloku tworzenia plansz ekranowych (Perform) i generatora raportów (Ace). Program wymaga co najmniej komputera IBM PC/AT lub jego kompatybilnej wersji, ok. 2,5 MB pamięci dysku twardego i co najmniej 512 KB RAM.

Za pomocą języka SQL można przetwarzać bazę danych według podanych kryteriów w celu otrzymania na nie odpowiedzi oraz formułować zapytania. Zapytania mogą być zapamiętywane, aby można je było wielokrotnie wykorzystać.

Blok Perform składa się z komend, które definiują plansze ekranowe do konkretnych zastosowań, określają kryteria edycji dla pól i pozwalają na pewne podstawowe przypisanie i arytmetyczne manipulacje na danych wyświetlanych na planszy.

Blok Ace posiada podobną architekturę opartą na komendach, które pozwalają generować raporty wyjściowe.

Wadą tych modułów jest niemożność korzystania w nich z języka SQL. Aby skompensować ten brak, firma Informix Software opracowała język Informix 4GL. Jest to specjalny język programowania dla tworzenia programów aplikacyjnych. Jego głównym celem jest umożliwienie programistom pełnego dostępu do bazy danych poprzez język SQL. Informix SQL zawiera bardzo szeroki zestaw funkcji statystycznych, instrukcji manipulacji na zmiennych łańcuchowych i możliwość działania na tablicach. Posiada on także instrukcję przypisania i konstrukcję pętlową. Informix 4GL w wersji 1.1 nie zawiera, niestety, modułu umożliwiającego łatwe projektowanie formatów na ekranie, co w znacznym stopniu zaoszczędziłoby czas, jaki pochłania ta czynność [6].

Informix 4GL może być uruchamiany na wielu różnych typach komputerów pracujących pod różnymi systemami operacyjnymi (Xenix, Unix, Dos, w przyszłości OS/2) [7].

Informix SQL i 4GL może działać na dwóch typach sieci:

- na sieciach typu LAN takich, jak Novell, PC Network, Token-Ring,

- na sieciach typu Starlan pod kontrolą UNIX-a.

Niedawno firma Informix wprowadziła specjalną bazę danych Turbo obsługującą serwera. Ta relacyjna baza danych odznacza się wysoką przetwarzalnością obsługi serwera przy dużych zbiorach danych. Turbo posiada także lepsze mechanizmy kontroli i odzyskiwania zbiorów. Informix oferuje także kilka innych pomocniczych narzędzi, tj.:

- C-ISAM, bibliotekę programów do języka C służących do obsługi zbiorów sekwencyjno-indeksowych,

- Datasheet Add-in umożliwiający łączenie zbiorów z pakietu Lotus 1-2-3 z bazą danych Informix SQL,

- ESQL/C, specjalny interfejs do języka C dla komputerów PC.

W Informix-ie dla systemu Unix są automatyczne interfejsy do języków C, Ada i Cobol.

Informix SQL i Informix 4GL są bazami danych o dużych możliwościach i pozwalają na tworzenie profesjonalnych systemów informatycznych. Szczególnie popularne są implementacje Informix-a pod wielodostępnym systemem Xenix.

SQLBase. SQLBase firmy Gupta Technologies była pierwszą relacyjną bazą danych, która zaimplementowała architekturę serwera w sieci LAN na komputerach PC/AT. W tym czasie kilka firm oferuje bazy danych SQL obsługujące serwera na AT, ale SQLBase jest jedyną, której serwer SQL może pracować także pod DOSem (możliwa jest praca w trybie wielozadaniowym). Program wymaga komputera PC AT lub kompatybilnego, 640 KB RAM, twardego dysku i wersji DOS-u co najmniej 3.1.

Serwer bazy danych SQLBase centralnie kontroluje mechanizmy:

- zamykania, otwierania zbiorów i rekordów,
- zabezpieczania zasobów,
- odzyskiwania straconych zbiorów i danych.

Wszystkie te mechanizmy są kontrolowane automatycznie i programista nie musi o nich pamiętać pisząc programy.

W środowisku serwera całe przetwarzanie bazy danych odbywa się przez serwera. Tylko najbardziej potrzebne informacje wysyłane są do stacji sieciowych, co redukuje zajętość i zmniejsza możliwość kolizji w sieci, a znacznie zwiększa jej funkcjonalność [6].

SQLBase w wersji 3.3 może zarządzać kilkoma serwerami w sieci, a więc program na określonym mikrokomputerze w sieci może połączyć się ze wszystkimi dostępnymi aktualnie bazami danych na różnych

serwerach. SQLBase prowadzi katalog, gdzie zachowywane są aktualne informacje o tym, który serwer zawiera jakie bazy danych.

Firma Gupta Technologies oferuje także super-bazę danych SQLNet (\$ 20 000 za wersję na duży komputer i \$ 2 000 na PC), która umożliwia współpracę programów na PC z bazą danych typu DB2 na dużym komputerze. SQLNet zawiera APPC (ang. advanced program-to-program communication), które umożliwia połączenie z relacyjną bazą danych typu DB2 na dużym komputerze. Dzięki APPC można wysyłać w języku SQL zapytania z lokalnego komputera do relacyjnej bazy danych DB2 i jednocześnie otrzymywać poprzez APPC odpowiedzi na pytania.

Firma Gupta oferuje także programy wspomagające SQLBase, m. in. SQLWindows i interfejsy do języka C. Zapowiedziana jest także wersja SQLBase zawierająca pełne możliwości języka 4GL.

Btrieve i XQL. Btrieve 4.1 firmy Novell jest pakietem zawierającym bazę danych i pomocniczy system do zarządzania plikami. Odznacza się on dużą szybkością przetwarzania. Przy jego pomocy programy użytkowników pisane w C mogą używać prostych procedur do poprawiania, przeszukiwania, sortowania i grupowania rekordów. Btrieve implementuje algorytm b-tree z automatycznym balansem do szybkiego i efektywnego indeksowania zbiorów. Posiada ponad 20 interfejsów do wszystkich popularnych języków programowania, takich jak: C, Basic, Pascal, Cobol itp. Posiada wbudowany system zabezpieczeń danych z użyciem haseł, kodów i mechanizmu odczytu dopiero po weryfikacji operacji zapisu, co jest szczególnie istotne w operacjach wielodostępnych. Btrieve zabezpiecza bazę danych przed nagłymi wypadkami i katastrofami, gdyż posiada dwa poziomy "fault tollerance" gwarantujące integralność danych nawet w wypadku nagłego wyłączenia komputera oraz możliwość odzyskiwania straconych zbiorów (dzięki poziomowi kontroli transakcji) [6].

Warto zaznaczyć, że Btrieve jest rozpowszechniany w wersji jedno- i wieloużytkowej dla sieci typu LAN, systemu wielodostępnego Xenix i systemu wielozadaniowego OS/2.

Na potrzeby użytkownika końcowego sprzedaje się dodatkowy pakiet Xtrieve, ściśle współpracujący z Btrieve, zawierający m. in. edytor raportów prowadzony przez menu i liniowy system zapytań. Do pakietu Btrieve dołączony jest program XQL 1.0, który umożliwia relacyjne zarządzanie danymi. XQL wymaga komputera IBM PC, PC AT lub kompatybilnego, dysku twardego, 512 KB RAM, wersji DOS-u 2.1 lub

wyżej. Do uruchomienia XQL 1.0 konieczna jest wersja 4.1 programu Btrieve. XQL posiada własną unikalną syntaktykę, która nie jest zgodna ze standardem języka SQL. Poza tym w XQL brakuje wielu ważnych komend języka SQL, m. in. możliwości definiowania subzapytań. Język zapytań XQL może być łączony poprzez Btrieve z programami napisanymi w Basicu, Pascalu i C. XQL posiada wystarczające mechanizmy zabezpieczenia i odtwarzania danych oraz niewygodny program optymalizujący.

Firma Novell zdając sobie sprawę z ograniczeń XQL zapowiada pełną implementację SQL.

XDB II. XDB II firmy Software Systems Technology jest relacyjną bazą danych przeznaczoną na komputery PC/AT i kompatybilne. Wyróżnia się spośród podobnej klasy baz danych łatwym sposobem komunikacji z użytkownikiem. Zaraz po załadowaniu programu na ekranie użytkownika wyświetla się menu zawierające listę dostępnych opcji. Zamiast wybierania opcji można używać języka SQL. Zapytania tworzone przy pomocy tego języka mogą być zapamiętywane do wielokrotnego wykorzystywania. W celu ułatwienia wyboru odpowiedniego zapytania można również zapamiętać odpowiedni komentarz. Generator raportów ułatwiający projektowanie wydruków jest bezpośrednio dostępny z poziomu języka SQL. Pozwala on na dowolne rozmieszczenie pól na wydruku, określenie tytułu wydruku, a także ustalenie wielkości marginesów.

Opcjonalnie pakiet XDB II może zawierać generator plansz ekranowych (ang. Forms generator package), który umożliwia m. in. definiowanie logiki edycji i korzystanie z komend języka SQL do wprowadzania, aktualizowania i usuwania rekordów. Generator ten jest bardzo łatwy w użyciu dzięki wykorzystaniu techniki okien [6].

XDB II zawiera także łatwy w użyciu generator menu. Dodatkowo można dokupić pakiet obsługujący grafikę, a także pakiet pozwalający na korzystanie z bazy danych poprzez programy aplikacyjne napisane w języku C lub Cobol. XDB II charakteryzuje się dużą szybkością działania, ale wydaje się być najsłabszą z baz danych SQL dostępnych na mikrokomputerach. Nie posiada jeszcze specjalnej wersji do obsługi serwera w sieci typu LAN ani też nie jest wykorzystywana w systemie XENIX.

SCO INTEGRA. SCO INTEGRA firmy Santa Cruz Operation jest relacyjną bazą danych specjalnie zaprojektowaną pod system operacyjny SCO XENIX SYSTEM V.

SCO INTEGRA używa języka SQL w standardzie ANSI (słabszy od standardu IBM). Zawiera pełen zestaw funkcji i komend potrzebnych w pracy wieloużytkowej. Poprzez interfejs kierowany przez menu INTEGRA jest łatwa do nauki, łatwa w użytkowaniu. Charakteryzuje się także szybkim dostępem do danych. SCO INTEGRA automatycznie tworzy standardowe formaty i raporty, które nawet początkujący użytkownik może łatwo modyfikować. Do języka SQL dodane jest pomocnicze narzędzie ADL (ang. Application Development Library) - biblioteka programów służąca do manipulacji bazą danych, przetwarzania zapytań, przetwarzania transakcji i tworzenia raportów z języka C. INTEGRA dołącza także pakiet SCO ISAM, który zawiera ulepszone metody korzystania ze zbiorów sekwencyjno-indeksowych.

Ta relacyjna baza danych posiada także mechanizmy zabezpieczenia zasobów przed niepożądanym dostępem i odzyskiwania straconych danych w razie nieprzewidzianego wypadku.

INTEGRA akceptuje zbiory z większości znanych baz danych i pakietów zintegrowanych, szczególnie łatwo można przenieść programy i dane utworzone przy pomocy INFORMIX 3.3.

Na rynku relacyjnych baz danych INTEGRA jest produktem dość nowym, zapowiadającym się bardzo interesująco, w Polsce jeszcze mało znanym.

dBASE IV. Długo oczekiwana, znacznie poprawiona wersja dBASE pojawiła się na rynku w sprzedaży w lipcu 1988 r. dBASE IV jest jakościowo zupełnie nową bazą danych, ale akceptuje wszystkie aplikacje i formaty zbiorów dBASE III. Głównym wyróżnikiem dBASE IV jest język SQL. Wersja ta nie ma jednakże jeszcze obsługi serwera SQL sieci LAN, ale taka wersja jest przewidziana w najbliższym czasie.

dBASE IV zawiera znacznie poprawiony interfejs komunikujący się z użytkownikiem (bardzo czytelny i przejrzysty), udoskonalony generator aplikacji, możliwości wprowadzania procedur macro z klawiatury i mechanizm QBE (ang. Query-by-Example) podobny do tego, w jaki wyposażony jest Paradox [1].

Dodatkowo zawiera nowy generator formatów ekranowych i raportów, które można projektować na ekranie, przeliczalne pola, kontrolę aktualności danych i inne nowe mechanizmy do obsługi ekranu. Istnieją także możliwości przedstawienia wyników działania programu w postaci graficznej, można to uczynić uruchamiając oddzielnie pakiet

graficzny CHART-MASTER, dBASE IV zawiera także nowy typ danych numerycznych (64 bity długości), posiada mechanizm automatycznego indeksowania zbiorów i dodatkowe biblioteki funkcji finansowych i matematycznych. Możliwe jest tworzenie okien z menu rozwijanym w górę i w dół. Nowy dBASE zawiera własny kompilator (wykonanie programów jest dziewięciokrotnie szybsze niż pod dBASE III).

dBASE IV posiada wiele mechanizmów potrzebnych do pracy wielo-użytkowej: ochronę zasobów, automatyczny locking zbiorów i rekordów, przetwarzanie transakcji i automatyczne wykrywanie błędów.

W lipcu 1988 r. dBASE IV był sprzedawany w dwóch zestawach: standard dBASE IV i w wersji rozszerzonej Developers Edition dBASE IV.

Wersja rozszerzona zawiera dodatkową dokumentację dla zaawansowanych programistów, pomocnicze narzędzie do łączenia kodów źródłowych, nieograniczony kompilator (Runtime support) i klucze LAN do testowania wielo-użytkowych aplikacji. Tak więc dopiero rozszerzona wersja dBASE IV zawiera naprawdę aktywna narzędzia i mechanizmy służące do tworzenia profesjonalnych systemów informatycznych [8].

PARADOX 2.0. PARADOX jest relacyjną bazą danych przeznaczoną dla komputera IBM PC. Można ją wykorzystywać dokonując wyboru odpowiednich opcji z menu lub pisząc programy w języku PAL (ang. Paradox Application Language).

Tworząc nowy zbiór przy użyciu Paradoxu należy określić strukturę rekordu, przy czym do dyspozycji jest 5 typów pól. Z wyborem konkretnego typu wiążą się automatyczne kontrole podczas wypełniania pola. Można też definiować dodatkowe kontrole (np. minimalna wartość), jak również ustalić wartości domyślne itd.

Poszczególne pliki mogą być przeglądane, redagowane lub sortowane. Sortowanie następuje automatycznie, gdy zdefiniowane są pola kluczowe. Można sortować dane wewnątrz starego zbioru lub tworzyć oddzielny posortowany zbiór.

Najbardziej cenną cechą Paradoxu wydaje się być dobrze zorganizowana obsługa zapytań oparta na metodzie "Query-by-Example". Korzystając z tej metody użytkownik otrzymuje na ekranie obraz struktury danego zbioru w postaci tabeli, której nagłówkami kolumn są nazwy pól. W odpowiednich kolumnach (polach) tej tabeli użytkownik wpisuje tzw. wzorce (ang. example), wskazując w ten sposób pola, których zawartość go interesuje. Użytkownik wprowadza również kry-

teria wyszukiwania (każde kryterium dotyczy tej kolumny-pola, w której została wpisana wartość). Można tworzyć złożone warunki używając operatorów logicznych, jak również używać komendy "podobny do..." dla poszukiwania łańcuchów znaków podobnych do zadanych. Istnieje też możliwość określenia (za pomocą odpowiednio podanych wzorców) połączeń między plikami, z których pochodzić będą dane potrzebne do uzyskiwania odpowiedzi na zapytanie. Po sprecyzowaniu wszystkich kryteriów określa się, jakie operacje mają być wykonane na danych. Możliwe jest wyszukiwanie rekordów i pól spełniających podane warunki, usuwanie wybranych rekordów (usunięte rekordy zachowywane są w specjalnym zbiorze, więc możliwe jest ich odzyskanie), jak również przepisanie zawartości rekordów i pól do ich odpowiedników w innym zbiorze. Po podaniu rodzaju operacji, jakie mają być wykonane, PARADOX generuje optymalny, z punktu widzenia czasu wykonania, program w języku PAL, a następnie go wykonuje. Program ten może być wykorzystywany wielokrotnie [1].

-Standardowo zbiory wyświetlone są na ekranie w formie tabelarycznej (jeden rekord w jednej linii). Można też zaprojektować do dziewięciu własnych plansz wejściowych (każda z nich może zawierać do 15 ekranów). Przy wyświetlaniu można dokonywać obliczania wartości pewnych pól, gdyż dostępnych jest wiele funkcji arytmetycznych. Istnieje możliwość określania formatu poszczególnych pól z wyjątkiem pól wyliczanych oraz uzyskanie pojedynczych i podwójnych ramek. Za pomocą odpowiednio zdefiniowanych plansz wejściowych można wprowadzić dane jednocześnie do kilku zbiorów. Niestety, PARADOX nie posiada mechanizmów obsługi słowników. Raporty wyjściowe mogą być standardowe lub zaprojektowane przez użytkownika (szerokość do 2000 znaków). Generując raport można definiować nagłówki i stopki, określać sposób uporządkowania i grupowania rekordów, wskazywać sposób sumowania poszczególnych kolumn, jak również używać pól o wartości wyliczalnej lub pól o zmiennej długości. Niestety, nie ma możliwości tworzenia raportów jednocześnie z kilku plików. Trzeba najpierw użyć opcji 'Query', stworzyć nowy zbiór zawierający żądane informacje z różnych plików. Generator raportów jest na tyle uniwersalny, że może służyć do obsługi korespondencji.

PARADOX jest również wyposażony w interpreter własnego języka PAL, służącego do pisania programów aplikacyjnych, a także edytor tekstu, wykorzystywany przy pisaniu tych programów. Syntaktyka jest

zbliżona do języka dBASE. Język PAL zawiera bogaty zestaw funkcji matematycznych i statystycznych, ma możliwość generowania menu, zabezpieczania hasłem, definiowania własnych procedur i łączenia ich w biblioteki, możliwość używania tablic, w których mogą być tymczasowo przechowywane całe rekordy, jak również wspomaganie przy uruchamianiu programów (debugger). Dodatkowo do PARADOXu dołączony jest specjalny program napisany w języku PAL. Program ten nosi nazwę ApGen i służy do łatwego generowania programów aplikacyjnych, wolnych od błędów, o optymalnej strukturze. Program ten oparty jest na wyborze opcji z odpowiednich menu. Można używać do 15 zbiorów i tworzyć menu na maksymalnie 10 poziomach. Na rynku dostępny jest także bardzo tani program Runtime, który pozwala na uruchomienie gotowych programów napisanych w PAL bez konieczności korzystania z PARADOXu. Niestety nie istnieje jeszcze kompilator języka PAL.

PARADOX może łatwo i szybko importować lub eksportować zbiory z, lub do formatów: ASCII, dBASE, LOTUS 1-2-3 i innych.

W najnowszej wersji 2.0 PARADOX posiada już mechanizm automatycznego uaktualniania danych na ekranach użytkowników w trybie "Co-Edit". Funkcja "Screen-Refresh" pozwala wybrać czas, w jakim dane będą uaktualniane (od 1 s do 1 h). PARADOX 2.0 został specjalnie zaprojektowany dla aplikacji pracujących w wielodostępie i sieciach - posiada rozbudowany mechanizm blokowania zbiorów i rekordów oraz zabezpieczania zasobów.

Oprócz wersji sieciowej PARADOX Network Pack dostępna jest też implementacja tego pakietu na komputer 32-bitowy. PARADOX 386 pozwala użytkownikom wykorzystać duże możliwości procesora 80386, a także umożliwia zakładanie olbrzymich baz danych dzięki możliwości działania na rozszerzonej pamięci RAM do 16 MB.

Twórcy pakietu PARADOX zapowiadają wersję SQL - PARADOX SQL oraz specjalną implementację PARADOXu pod OS/2 - PARADOX OS/2, oraz wersję dla systemu UNIX - PARADOX UNIX. Wszystkie wersje PARADOXu będą w pełni kompatybilne ze sobą pod względem aplikacji i menu.

W klasie baz danych pracujących pod DOS-em PARADOX 2.0 uważany jest za najmocniejszą i jednocześnie łatwą do nauki bazę danych.

FoxBASE. Firma Fox Software w roku 1987 wypuściła na rynek dwa bardzo interesujące produkty zgodne ze standardem dBASE: FoxBASE (odpowiednik dBASE II) i FoxBASE+ (odpowiednik dBASE III+). Główne

zalety tych pakietów to pełna kompatybilność z odpowiednią wersją pakietu dBASE, akceptacja programów aplikacyjnych napisanych pod dBASE-m, znacznie szybsze wykonywanie programów aplikacyjnych niż w dBASE (5 do 7 razy większa prędkość przetwarzania), pełny zestaw komend sieciowych w wersji wieloużytkowej, szybkość działania programów uruchamianych z poziomu interpretera, porównywalna z szybkością działania programów skompilowanych w innych językach, rozszerzone właściwości m. in.: możliwość definiowania tablic, zmiennych pamięciowych, wielokrotne relacje w bazie danych itp. Dla pakietów FoxBASE dostępny jest również pełen ich kompilator Runtime System.

FoxBASE zyskał sobie dużą popularność nie tylko dzięki dużej prędkości przetwarzania, ale też z tego powodu, że prawie równoległe pojawiły się jego wersje wieloużytkowe dla systemu DOS i XENIX. Obecnie firma Santa Cruz (twórca systemu XENIX System V) firmuje specjalne wersje tej bazy danych: SCO FoxBASE+ i SCO FoxBASE. Pojawiła się również wersja FoxBASE na komputer 32-bitowy - FoxBASE+386. Wersja ta dzięki wykorzystaniu możliwości procesora 80386 jest ponad dwukrotnie szybsza od FoxBASE+ 2.0.

Porównanie przedstawionych baz danych

T a b e l a 1

Zestawienie ogólne, wymagania sprzętowe

Baza danych	Firma	Produkt	Wymagania sprzętowe
1	2	3	4
Ingres 5.0	Relational Technology	Ingres 5.0 (baza) Aplikacje (4GL) C/ESQL IngresNet Ingres PC Link	IBM PC AT lub SUPER AT 640 KB RAM dysk twardy min 10 MB

Tabela 1 (cd.)

1	2	3	4
Oracle. 5.1	ORACLE	Professional Oracle Oracle (baza) Oracle Network station	IBM PC AT, PS/2 lub SUPER AT 1 MB RAM dysk twardy
Informix SQL 4GL	Informix Softwa- re	Informix SQL Informix 4GL C-ISAM ESQL/C	IBM PC AT 512 KB RAM dysk twardy min 2,5 MB
SQLBase 3.2	Gupta Technolo- gies	SQLBase SQLBase (multiuser) SQLNet	IBM PC AT 640 KB RAM dysk twardy DOS ver 3.1
Btrieve 4.1 XQL 1.0	Novell	Btrieve Btrieve/N XQL Xtrieve	IBM PC AT 512 KB RAM dysk twardy DOS ver 2.1
XDB II	Software System Tech.	XDB II Forms generator interfejs do C	IBM PC AT 512 KB RAM dysk twardy DOS ver 2.0
SCO Integra	Santa Cruz Opera- tion	SCO Integra	IBM PC AT lub SUPER AT 1,5 MB RAM dysk twardy
dBASE IV	Ashton-Tate	dBASE IV (stan- dard) Developers Edition dBASE III+ dBASE LAN PACK	IBM PC XT, AT lub PS/2 512 KB RAM dysk twardy

Tabela 1 (cd.)

1	2	3	4
FoxBASE	For Software Santa Cruz Operation	FoxBASE+2.0 FoxBASE+ (multiuser) FoxBASE+/386 SCO FoxBASE+ Runtime System	IBM PC XT, AT 512 KB RAM dysk twardy
Paradox	Borland/Ansi	Paradox 2.0 Paradox Network Pack	IBM PC XT, AT 512 KB RAM dysk twardy

Tabela 2

Zestawienie testów prędkości przetwarzania baz zawierających język SQL

Nr testu	Informix	Ingres	Oracle	SQLBase	XDB II	XQL
1	23	29	76	35	64	316
2	43	30	21	46	16	xx
3	78	22	24	48	18	23
4	39	6	5	2	1	4
5	20	13	5	23	9	29
6	10	21	24	22	7	6
7	26	9	8	17	9	44
8	124	41	15	9	30	37
9	19	33	24	46	40	12

U w a g a: Wszystkie czasy podane są w sekundach.

Oznaczenie testów:

1. Załaduj 1000 rekordów;
2. Utwórz indeks według SEQNO;
3. Utwórz indeks według ZIP;
4. Wykonaj SELECT * FROM PERSONS WHERE ZIP = '60606' AND SEQNO < '999' (testuje zdolność optymalizacji wyboru właściwego indeksu);

5. Wykonaj `SELECT * FROM PERSONS WHERE ZIP = '60606' OR SEQNO = '999'` (testuje zdolność optymalizacji używania indeksów);

6. Wykonaj `SELECT * FROM PERSONS WHERE SEQNO > '980' ORDER BY ZIP` (testuje zdolność optymalizacji używania indeksów według ZIP; zapytanie nie wymaga wtedy sortowania zewnętrznego);

7. Wykonaj `SELECT SUM(SALARY) FROM PERSONS` (testuje matematyczną funkcję agregowania);

8. Wykonaj `SELECT ZIP FROM PERSONS GROUP BY ZIP HAVING COUNT (*) > 5` (grupowanie wymagające sortowania z dodatkowymi funkcjami);

9. Wykonaj `SELECT A.SEQNO, B. SEQNO AND A.ZIP LIKE '606%'` (łączenie z opcją LIKE).

Opis relacji PERSONS używanej przez powyższe testy:

SEQNO CHAR(4)

NAME CHAR(30)

TITLE CHAR(30)

COMPANY CHAR(30)

DEPARTAMENT CHAR(30)

ADRESS CHAR(30)

CITY CHAR(20)

STATE CHAR(2)

ZIP CHAR(1)

SALARY MONEY

Większość testów sprawdzała efektywność przetwarzania przy prostych, a raczej podstawowych funkcjach baz danych. Testy 4, 5, 6 sprawdzały przy okazji zdolność danej bazy do optymalizacji. Test 9 dawał także wybór drogi do odpowiedzi na zapytanie. Testy oraz wyniki podano na podstawie artykułu "SQL Database w Management System", [6].

Porównanie możliwości przedstawionych baz danych

Z przedstawionych 10 systemów zarządzania relacyjnymi bazami danych zdecydowanie pozytywnie wyróżniają się: Ingres, Oracle i Informix. Wszystkie te pakiety są zaimplementowane z dużych komputerów. Dostyc wcześnie zastosowały język SQL i inne narzędzia 4GL (QBE, QBF, QMF) w swoich wersjach. Bazy te spełniają warunki Codda na relacyjność baz danych, a ich kolejne wersje są znacznie ulepszone i

pozbawiane wykrytych błędów. Twórcy tych programów rozwiązali już także problem serwera SQL w sieci. Następnymi krokami w rozwoju tych produktów będzie pełne zastosowanie języków 4GL, opracowanie wersji języków baz danych wykorzystujących moc procesora 32-bitowego, połączenia PC z różnymi komputerami (poprzez tzw. PCLink) o dużej mocy przetwarzania, podłączenie zasobów bazy danych do innych systemów informatycznych, tj. CAD/CAM (istnieje już pierwszy program Postgres firmy Ingres) czy systemy ekspertowe. W rozwoju systemów zarządzania relacyjnymi bazami danych dają się zauważyć pewne tendencje dążenia do: umożliwienia dostępu do plików przygotowanych przez inną bazę danych, zwiększenia stopnia współbieżności dostępu do bazy oraz wyposażenia pakietów w nowe języki wysokiego poziomu pozwalające na pracę użytkownikom nie będącym profesjonalnymi informatykami. Ze względu na rosnącą w Polsce popularność systemu SCO XENIX należy podkreślić fakt, że wyżej wymienione systemy posiadają już swoje implementacje pod tym systemem. Jediną większą wadą tych systemów RDB są wymagania sprzętowe, duża zajętość pamięci dysku twardego (5-10MB), efektywne działanie dopiero przy rozszerzonej pamięci RAM (ok. 2MB), jednostka centralna co najmniej AT, a najlepiej 80386. Systemy te oferują za to profesjonalne mechanizmy ochrony zasobów, prawdziwą pracę w wielodostępie, profesjonalną kontrolę danych wejściowych, proste możliwości rozbudowy istniejących systemów aplikacyjnych i wiele innych narzędzi i mechanizmów potrzebnych do tworzenia profesjonalnych systemów informatycznych.

Pozostałe bazy danych są bardzo dobre w swojej klasie (PARADOX, FoxBASE), ale nie posiadają jeszcze wszystkich mechanizmów do tworzenia profesjonalnego oprogramowania.

Podsumowanie

Dziedzina relacyjnych baz danych na mikrokomputerach jest bardzo młoda i rozwija się bardzo dynamicznie. Trzeba jednoznacznie stwierdzić, że nie ma bezkonkurencyjnego lidera wśród systemów zarządzania relacyjnymi bazami danych, chociaż zwraca uwagę fakt, że cały czas obecne są na rynku takie firmy jak: RTI, Oracle, Informix, Ashton-Tate.

Powszechne jest stosowanie w dziedzinie systemów zarządzania relacyjnymi bazami danych języka SQL, narzędzi 4GL oraz trend do łą-

czenia różnych komputerów w sieć poprzez pracę z tą samą bazą danych. W przyszłości bazy danych powinny posiadać standardowy język typu SQL, który pozwalałby całemu oprogramowaniu: systemom ekspertowym, systemom CAD/CAM czy interaktywnym interfejsom zapytań na dostęp do centralnego źródła informacji. Coraz realniejsza wydaje się być także perspektywa używania naturalnego języka angielskiego w oprogramowaniu. Z analizy szeregu relacyjnych baz danych wynika, że znacznie lepiej jest rozwiązany problem pracy olbrzymiej bazy danych na dużej jednostce centralnej w trybie wielodostępu, niż rozproszonej bazy danych w sieci (zwłaszcza problem kontynuowania pracy pozostałych komputerów w sieci przy awarii jednego komputera). Z tego powodu należy szczególnie uważnie obserwować implementacje RDBS na komputery 32-bitowe (89386), ponieważ pełne wykorzystanie możliwości tego procesora może rozwiązać wiele problemów związanych z zastosowaniem relacyjnych baz danych na mikrokomputerach.

Literatura

- [1] B a r a n N., dBASE IV a PARADOX Killer?, "Byte", April 1988.
- [2] B r y a n M., 1988 The Year of The Database, "Personal Computing", January 1988.
- [3] C h o r a f a s D. N., Fourth and Fifth Generation Programming Languages, McGraw-Hill International Editions, N. Y. 1987.
- [4] D a t a C. J., Relational Database, Addison-Wesley Publishing Company, Massachusetts 1986.
- [5] F i n k e l s t e i n R., Lingua Franca for Databases, PC Tech. Journal, December 1987.
- [6] F i n k e l s t e i n R., P a s c a l F., SQL Database Management System, "Byte", January 1988.
- [7] F r a n k l i n C. Jr, SQL - based Database Managers, "Byte", January 1988.
- [8] L i s k i n M., And now dBASE IV, "Personal Computer", May 1988.
- [9] P a s c a l F., Relational Power PC Ease, PC Tech. Journal, December 1987.
- [10] W i n s t o n A., What is Different about The Many Fourth-Generation Languages on The Market Today? And How Do You Choose Among Them, "Unix World", September 1988.

Anna Pamuła, Marek Zawada

REVIEW OF PROFESSIONAL SYSTEMS OF MANAGING
RELATIONAL DATA BASES

Relational data bases are a novel but very dynamically developing field of knowledge. The article contains a review of the available and most popular systems of management of data bases. Its introductory part describes the rules concerning the relational model of data base elaborated by an American mathematician E. F. Codd and a short description of the SQL language being a standard of the structural language of questions increasingly more commonly used in the systems of management of relational data bases today. The core of the article are short presentations of the following data bases: Oracle 5.1, Ingres 5.0, Informix SQL and 4GL, SQLBase, Btrieve 4.1. and XQL 1.0, XDB II, SCO Integra, dBASE IV, Paradox 2.0, FoxBASE. In the final part of the article, the author presents a list of base processing speed tests containing the SQL language and a general list of the described data bases containing names of producers and minimal hardware requirements. The article ends with presentations of the present trends in development of the systems of management of data bases.