

Renata BRESIŃSKA

## BAZY DANYCH W ŚRODOWISKU SIECI LOKALNYCH

**Streszczenie.** SQLowe bazy danych zdecydowanie wkraczają w świat komputerów osobistych. Jednym z takich przykładów jest omówiony w niniejszym artykule SQLSystem firmy Gupta Corporation, rozwiązanie szczególnie dopasowane do środowiska sieci lokalnych NetWare. Artykuł przedstawia zalety architektury „klient-serwer”, stanowiącej nową jakość pracy w przetwarzaniu rozproszonym.

## DATABASES MANAGMENT SYSTEMS IN PC LAN ENVIRONMENT

**Summary.** SQL databases enter PC world unquestionably. One of the examples is the SQLSystem from Gupta Corporation, a solution presented in this article, suited especially for NetWare LANs. The article specifies the advantages of the client-server architecture introducing a new work quality in the area of distributed computing.

## DATENBANKEN MIT EINER „CLIENT - SERVER” ARCHITEKTUR IN RECHNERNETZEN

**Zusammenfassung.** SQL-Datenbanken gehören gegenwärtig eindeutig zum Bereich der Personalcomputer. Eine von solchen Lösungen stellt das hier beschriebene SQLSystem der Firma Gupta Corp. dar, die Software, die besonders gut an die Umgebung des NetWare-Netzes angepasst ist. Im Beitrag wurden die Vorteile der „Client - Server” Architektur vorgestellt, von der die neuen Anwendungsmöglichkeiten von Mikrorechnern geschaffen werden.

## 1. SQL - język baz danych

Wraz ze wzrostem mocy komputerów osobistych coraz popularniejszym tematem w środowisku sieci lokalnych stają się bazy oparte na architekturze „klient-serwer”, gdzie językiem dostępu do danych jest SQL. Bazy SQL wywodzą się bezpośrednio z relacyjnego modelu danych. Relacyjne bazy danych mają już za sobą długą historię - pierwsze produkty tworzone były głównie dla dużych maszyn i komputerów mini. Dopiero gwałtowny rozwój technologii spowodował, że bazy te wkraczają w świat komputerów osobistych.

Model danych określony jest poprzez organizację struktur danych oraz stosowaną w DBMS (system zarządzania bazą danych) technikę dostępu do danych. Szerzej znane są trzy rodzaje modeli danych: hierarchiczny, sieciowy i relacyjny. Wady modeli hierarchicznego i sieciowego były przyczyną dużego zainteresowania modelem relacyjnym, którego matematyczny opis przedstawił dr Codd w 1970 r. Model relacyjny był próbą uproszczenia struktury baz danych. Eliminował bezpośrednie powiązania typu ojciec-syn z bazy i wszystkie dane przedstawiał w postaci tablic składających się z wierszy i kolumn. Pierwsze relacyjne bazy nie implementowały wszystkich założeń modelu Codd'a. W 1985 r. Codd przedstawił 12 reguł, które muszą zostać spełnione, aby baza danych mogła być nazwana bazą relacyjną. Żadna z aktualnie dostępnych baz danych na rynku nie implementuje wszystkich założeń Codd'a - stały się one bardziej ideałem, do którego dążą wszyscy producenci relacyjnych baz danych. Stopień, w jakim konkretne implementacje spełniają reguły Codd'a, stanowi doskonałą skalę porównawczą dla różnych produktów. Jedną z zasad Codd'a jest to, aby istniał przynajmniej jeden język, który spełnia wszystkie centralne funkcje bazy danych - tworzenia bazy, wyszukiwania i wprowadzania informacji, zabezpieczeń danych. Takim językiem jest SQL - Structure Query Language, obecnie najbardziej popularny język dostępu do relacyjnych baz danych. SQL nie jest kompletnym językiem programowania, jak Pascal, C czy Fortran. Nie zawiera on typowych instrukcji, jak instrukcje warunkowe, pętli. W programowaniu nie może być stosowany samodzielnie, lecz tylko w połączeniu z innymi językami. Instrukcje SQL są z reguły „włożone” do programu napisanego w innym języku. Pomimo swojej nazwy, SQL nie jest językiem szczególnie strukturalnym. Składa się on z około 30 instrukcji, a każdą instrukcję rozpoczyna proste angielskie słowo. Na początku nie było przesądzone, czy SQL stanie się standardowym językiem relacyjnych baz danych. Wzrost popularności SQL, poparty staraniami IBM, przesądził jednak o jego wyborze. W 1986 standard SQL został zatwierdzony przez ANSI, a w 1987 przez ISO. W Europie SQL dołączony został do standardu X/OPEN tworzącego środowisko dla aplikacji UNIXowych. Różni się on w kilku miejscach od standardu ANSI/OSI. Ponieważ w pracach nad standardem uczestniczyli przedstawiciele wszy-



stkich znaczących producentów baz danych, powstał dość luźny standard zawierający szereg luk. Do najważniejszych jego braków należą:

- Kody błędów - standard nie definiuje kodów błędów zwracanych przez SQL.
- Typy pól - standard definiuje tylko minimalny zestaw typów pól.
- Tablice systemowe - w standardzie pominięta jest struktura tablic systemowych. Każdy produkt stosuje swoją własną strukturę.
- Interakcyjny SQL - w standardzie zdefiniowany jest tylko programowy SQL (używany w aplikacjach), natomiast pominięty został SQL używany interakcyjnie.
- Interfejs programowy - standard opisuje abstrakcyjną technikę stosowania SQL w aplikacjach. Żaden rynkowy produkt nie stosuje tej techniki i występuje duża różnorodność w rzeczywistości używanych interfejsach programowych.
- Dynamiczny SQL - standard nie zawiera cech języka wymaganych przy pisaniu narzędzi programowych, np. generatora wydruków. Takie cechy, zwane dynamicznym SQLem, występują we wszystkich dostępnych produktach, znacznie je różniąc.
- Różnice semantyczne - standard określa pewne elementy jako definiowalne przez producenta. Przykładowo różnice występują poprzez różne podejścia do wartości pustych (NULL), różne sposoby eliminowania podwójnych wierszy itp.
- Kolejność sortowania - ponieważ jest nie określona, wynik zapytania może być różny w zależności od tego, czy zapytanie wystąpiło na dużej maszynie (kodowanie EBCDIC), czy na komputerze osobistym (ASCII).
- Struktura bazy danych - sposoby uzyskiwania dostępu do bazy (logowanie) oraz nazewnictwo bazy nie są zdefiniowane i bardzo różnią poszczególne implementacje.

Pomimo tych wszystkich niedogodności, SQL staje się coraz szerszym standardem w świecie baz danych. Nawet producenci tradycyjnych baz danych dołączają ten język do obsługi swoich baz obok języka macierzystego.

Zalety SQLowych baz danych to przede wszystkim:

- Architektura klient-serwer. Baza SQL pracuje zwykle na komputerze, na którym fizycznie znajdują się dane. Oznacza to, że dla jednego miejsca przechowywania informacji pracuje jeden program zarządzający bazą danych (DBMS). Taki program określany jest zwykle jako serwer bazy danych. Poszczególni użytkownicy używają na swoich komputerach własnych programów, będących interfejsem do serwera bazy danych. Określane jako programy typu klient są one często produkowane przez niezależnych producentów oprogramowania. Serwer bazy danych zajmuje się m.in. wyszukiwaniem danych. Co więcej, robi to w sposób inteligentny - tak, aby uzyskać najszybszy dostęp do danych. W sieciach NetWare przy tradycyjnym przetwarzaniu, aby uzyskać dostęp do żądanych danych, na stację ściągany jest cały zbiór, niezależnie od tego, co aktualnie jest potrzebne. Tutaj poszczególni klienci wysyłają do bazy żądania w postaci zapytania SQL uzyskując w odpowiedzi tylko te dane, które są po-



trzebne. Powoduje to drastyczne zmniejszenie ilości informacji przesyłanych przez łącza sieci.

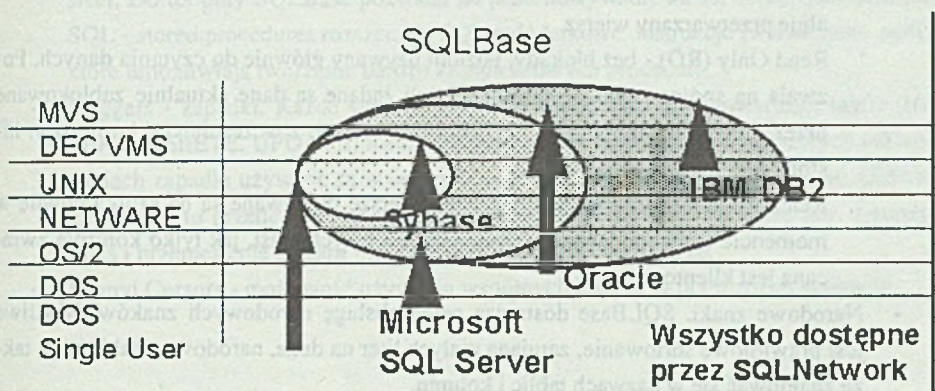
- **Niezależność od producenta.** Zarówno bazy danych, jak i programy typu klient mogą być przenoszone do baz danych innych producentów. Działania takie nie wymagają dużych nakładów pracy poświęconych konwersji danych. Najczęściej programy typu klient i serwery baz danych różnych producentów pasują nawzajem do siebie.
- **Przenośność oprogramowania.** Producenci SQLowych baz danych produkują swoje bazy na różnych platformach - od mikrokomputerów poprzez mini aż do dużych maszyn. Pozwala to na zastosowanie odpowiednio mocnego serwera bazy danych w zależności od potrzeb użytkownika - stworzony początkowo system dla pojedynczego użytkownika może zostać przeniesiony na platformę mini lub dużą maszynę bez konieczności zmian kodu. Naturalny podział oprogramowania na serwer bazy danych i program typu klient pozwala na stosowanie preferowanych systemów po każdej ze stron.
- **Wielodostępność i kontrola dostępu.** Ponieważ bazy SQLowe od początku stworzone były dla systemów wielodostępnych, dlatego są one uzbrojone w odpowiednie mechanizmy współużywania danych. Baza sama używa odpowiednich blokad, strzegąc użytkownika przed niezamierzoną interferencją i uwalniając go od stosowania funkcji synchronizujących. Osobną sprawą jest kontrola dostępu do danych. Bazy SQLowe mają rozbudowany mechanizm kontroli dostępu poprzez system użytkowników, haseł i praw. System zabezpieczeń pozwala nadawać użytkownikom tylko prawa do wybranych tablic bazy danych, a nawet do wybranych kolumn czy też wierszy.
- **Spójność danych.** Spójność danych można podzielić na fizyczną i logiczną. Spójność fizyczna, to zgodna z zdefiniowaną strukturą reprezentacja danych w fizycznym zbiorze. Spójność logiczna, to zgodność danych z projektem programu. Bazy SQLowe chronią swoją spójność poprzez mechanizmy zwane transakcjami - każda instrukcja w języku SQL - to początek transakcji, koniec oznajmiany jest poprzez polecenie COMMIT. W razie awarii nie dokończone transakcje są odwijane (ROLLBACK) i przywracany jest stan bazy przed rozpoczęciem transakcji. Jednym z elementów logicznej spójności jest spójność powiązań (referential integrity).

## 2. SQLBase - idealny serwer bazy danych dla sieci NetWare

SQLowe bazy danych pozwoliły stworzyć w sieciach lokalnych bazy danych zastępujące tradycyjne instalacje na dużych komputerach lub minikomputerach. Na rynku jest obecnie parę komercyjnych implementacji SQLowych baz danych - jednym z takich produktów jest



SQLSystem firmy Gupta Corporation. Produkt ten obejmuje serwer bazy danych oraz całą grupę narzędzi 4GL służących programiście. Swoje zainteresowania Gupta skupiła głównie na platformie sieci lokalnej komputerów PC - SQLBase, serwer bazy danych produkowany jest dla następujących środowisk: DOS, OS/2, UNIX Sun oraz NetWare. W grudniu tego roku spodziewane są wersje dla UnixWare'a oraz Windows NT.



Rys.1. Miejsce SQLBase

Fig.1. Position of SQLBase

SQLBase posiada następujące cechy:

- Zgodność z tzw. industry standard SQL (OS/2 EE Database Manager, DB2).
- Implementacja spójności powiązań poprzez klucze pierwszy i obcy. Klucz pierwszy (primary key) jest to kolumna lub zbiór kolumn w tabeli, które używa się w celu jednoznacznego zidentyfikowania wiersza. Klucz obcy (foreign key) odwołuje się do klucza pierwszego w innej lub tej samej tabeli. Klucze obcy i pierwszy tworzą powiązania pomiędzy tabelą ojciec (posiadająca klucz pierwszy) i tabelą zależną (posiadająca klucz obcy). Poprzez definiowanie powiązania można określić zasady usuwania wierszy z tabeli. Zasada ON DELETE RESTRICT nie pozwala na usunięcie wiersza z tabeli „ojciec”, jeżeli wiersz ten posiada jakikolwiek wiersz w tabeli zależnej. ON DELETE CASCADE usuwa automatycznie wszystkie zależne wiersze, natomiast ON DELETE SET NULL ustawia wszystkie klucze obce na NULL.
- Skrołowanie kursora (Scrollable cursor). Wiersze które są wynikiem zapytania (SELECT), znajdują się w specjalnie utworzonej tabeli odpowiedzi (result set). Skrołowanie kursora umożliwi przeglądanie wierszy (browse), znajdujących się w tabeli odpowiedzi, wykonywanie na zbiorze poleceń INSERT, UPDATE, DELETE wg aktualnej tabeli odpowiedzi.
- Zachowanie kontekstu kursora (Cursor context preservation). Dzięki temu, po wykonaniu polecenia COMMIT, nie jest zburzony result set i w dalszym ciągu można poruszać się po zbiorze odpowiedzi.

- Poziomy izolacji (Isolation Levels). Różne poziomy izolacji pozwalają na kontrolę współużywania danych. Dane blokowane są na poziomie strony. Możliwe są 4 poziomy izolacji:
  - Read Repeatability (RR) - blokowane są wszystkie strony, na których znajdują się żądane wiersze, dopóki nie zostanie wydane polecenie COMMIT.
  - Cursor Stability (CS) - blokowana jest tylko ta strona, na której znajduje się aktualnie przetwarzany wiersz.
  - Read Only (RO) - bez blokady, poziom używany głównie do czytania danych. Pozwala na spójne czytania danych - jeżeli żądane są dane aktualnie zablokowane przez innych użytkowników, SQLBase czyta dane z wcześniejszej kopii, tzw. history-file.
  - Release Lock (RL) - na tym poziomie wiersze blokowane są na krótką chwilę w momencie czytania. Blokada zwalniana jest natychmiast, jak tylko kontrola zwracana jest klientowi.
- Narodowe znaki. SQLBase dostarcza pełną obsługę narodowych znaków. Możliwe jest prawidłowe sortowanie, zamiana małych liter na duże, narodowe znaki mogą także znajdować się w nazwach tablic i kolumn.
- Typy pól. W celu kompatybilności z DB2, SQLBase posiada kilkanaście różnych słów kluczowych w celu zadeklarowania tego samego typu pola. Oprócz standardowych typów pól SQLBase posiada typ pola LONG VARCHAR - jest to nieograniczony ciąg znaków. Służy do przechowywania obrazków lub dźwięków (BLOBS).
- Obsługiwane protokoły. SQLBase obsługuje następujące protokoły:
  - SPX/IPX
  - NetBIOS
  - TCP/IP
  - Named Pipes
  - APPC
- Składowanie. Składowanie możliwe jest zarówno on-line, jak i off-line, całościowe, jak i przyrostowe.
- Szybkość pracy. W celu zwiększenia szybkości pracy SQLBase stosuje następujące metody:
  - buforowanie pamięci,
  - indeksy (B-Tree, Hashing),
  - optymalizacja - SQLBase podczas wyszukiwania danych stosuje optymalizację kosztową.

Charakterystyczną cechą SQLBase'a jest to, że powstał on specjalnie dla środowiska sieci lokalnych PC. Dla sieci NetWare pracuje jako NLM i jest specjalnie optymalizowany - obecnie uważany jest za najszybszy serwer dla tego środowiska. Wiele baz danych powstawało na ogół dla platform wyższych i dopiero obecnie są dostosowywane dla środowisk PC. Bardzo często odbywa to się kosztem tracenja pewnych istotnych cech oraz szybkości. W



obecnej chwili SQLBase nie posiada mechanizmów służących do obsługi baz rozproszonych (two-phase commit), niemniej w zapowiadanej wersji na początku przyszłego roku spodziewana jest implementacja nie tylko tej cechy, ale i wielu innych, jak np.:

- Stored Procedures - możliwość przechowywania na serwerze skompilowanych poleceń SQL. Używanie Stored Procedures znacznie ogranicza przesyłanie informacji w sieci. Do tej pory SQLBase pozwalał na przechowywanie na serwerze tylko komend SQL - stored procedures rozszerzają SQL o dodatkowe instrukcje (warunkowe, pętle), które umożliwiają tworzenie bardzo kompleksowych procedur.
- Triggers - zapadki. Każde wydarzenie powodujące zmianę w zawartości tablic (INSERT, DELETE, UPDATE) można związać z określoną akcją. W niektórych rozwiązaniach zapadki używane są w celu zaimplementowania spójności powiązań. Główna ich zaleta, to przede wszystkim możliwość przechowywania w bazie tzw. business rules i przeniesienie ciężaru programowania z aplikacji na bazę.
- Shared Cursors - możliwość używania wspólnych kursorów przez różne aplikacje.

### 3. SQLWindows - nowoczesne narzędzie pracy dla programisty

Oprócz bazy danych Gupta dostarcza całej grupy narzędzi dla programistów. Przede wszystkim jest to SQLWindows, nowoczesny język programowania czwartej generacji do tworzenia aplikacji baz danych w środowisku MS Windows. Główny atut SQLWindows, to obiektowe tworzenie programów. Programista posiada zestaw wstępnie zdefiniowanych klas obiektów. Może z nich od razu budować aplikację lub tworzyć swoje własne klasy obiektów z wszystkimi korzyściami programowania obiektowego (dziedziczenie, polimorfizm). Klasy mogą być funkcjonalne (bez reprezentacji graficznej, tylko dane i funkcje) oraz okienkowe, wykorzystujące funkcje dostarczone przez SQLWindows dla danego obiektu. Tworzenie aplikacji polega na wykorzystaniu własnych i wbudowanych w SQLWindows klas obiektów i dopisywaniu kodu programu. Programowanie jest sterowane zdarzeniami. Określone zdarzenia (np. naciśnięcie klawisza myszy, edycja pola) powodują wysłanie komunikatu do aplikacji. Zdarzenie może zostać oprogramowane - oznacza to, że ze zdarzeniem może zostać związana akcja, czyli kod programu. Pisanie kodu programu odbywa się w strukturalnym języku SQLWindows Application Language (SAL), posiadającym kilkadziesiąt wbudowanych funkcji. Funkcje wewnętrzne pozwalają na:

- dostęp do danych z bazy SQL,
- wygodną prezentację danych,
- zarządzanie oknami,

- komunikację z innymi procesami (DDE i OLE),
- odczyt i zapis do zbiorów DOSowskich oraz wiele innych.

SQLWindows posiada szereg udogodnień dla programisty. Kontekstowa pomoc pozwala na pisanie programów bez sięgania do dokumentacji. Okno wyboru opcji prowadzi piszącego za rękę, pozwalając wpisać tylko właściwe w danym miejscu elementy języka oraz redukując używanie klawiatury. Okno narzędzi daje możliwość wyboru obiektów, które programista rozmieszcza odpowiednio w oknach swojego programu. W trakcie uruchamiania programista może zakładać punkty kontrolne i sprawdzać wartości zmiennych. SQLWindows posiada pewne wbudowane mechanizmy służące do zespołowej pracy programistów. Cecha ta została rozwinięta w nowym produkcie - TeamWindows.

## 4. Quest

Następny program to Quest - graficzne narzędzie dostępu do baz danych SQL. Jest doskonałym uzupełnieniem pozostałych produktów, pozwalającym na obsługę bazy danych bez konieczności poznawania komend języka SQL. Dzięki temu Quest jest niezwykle użyteczny nie tylko dla profesjonalistów, ale przede wszystkim dla przeciętnego użytkownika bazy danych. Wszechstronne możliwości tego produktu pozwalają na dostęp do danych znajdujących się zarówno na dużych maszynach i minikomputerach, jak również w sieciach typu LAN. Quest umożliwia:

- tworzenie, modyfikowanie i przeglądanie tablic baz danych bez używania poleceń SQL,
- uzyskiwanie potrzebnych informacji z bazy za pomocą zapytań stworzonych w sposób prosty i intuicyjny,
- tworzenie formatki do wprowadzania danych,
- tworzenie w prosty sposób różnorodnych raportów,
- używanie danych z Questa w innych aplikacjach MS Windows ( technika OLE ).

## 5. TeamWindows

Gupta dostarcza także TeamWindows - oprogramowanie służące do zarządzania i nadzorowania dużych projektów informatycznych. Narzędzie to umożliwia pełną kontrolę procesu tworzenia aplikacji, wprowadzanie standardów w projekcie, specjalizację w programowaniu. Raz napisane części programu mogą zostać ponownie użyte w innych aplikacjach lub przez



innych programistów. W specjalnym obszarze bazy danych SQLBase, zwanym repozytorium, przechowywane są wszystkie informacje o aplikacji - od formatów i raportów do danych o zespole tworzącym system. Dzięki temu możliwe jest:

- generowanie aplikacji na podstawie struktury bazy - TeamWindows utrzymuje informacje o strukturze baz danych, powiązaniach pomiędzy tablicami, kryteriach kontroli i innych parametrach,
- dokonywanie analizy, w jaki sposób zmiany struktury bazy wpływają na aplikację oraz automatyczne modyfikowanie tej aplikacji,
- kontrolę kodu źródłowego i wersji oprogramowania - gotowa aplikacja przechowywana jest w repozytorium, gdzie specjalne mechanizmy kontroli dostępu nie pozwalają nieuprawnionemu użytkownikowi na dostęp do aplikacji; podczas zmian w aplikacji automatycznie generowana jest kolejna wersja programu,
- tworzenie i utrzymywanie biblioteki wzorców; TeamWindows dostarcza wstępnie zdefiniowane wzorce do szybkiego tworzenia ekranów i aplikacji z danych zawartych w repozytorium.

Niektóre z cech oprogramowania TeamWindows są charakterystyczne dla narzędzi typu "lower CASE", a połączone z oprogramowaniem "upper CASE" pozwalają stworzyć w pełni profesjonalne narzędzie do projektowania dużych systemów informatycznych.

## 6. Otwartość

Aplikacja stworzona za pomocą SQLWindows może współpracować nie tylko z serwerem SQLBase. Ponieważ SQLBase obsługuje tylko najniższe platformy, Gupta dostarcza także produkty z grupy SQLNetwork (routery) pozwalające na dostęp do baz danych innych producentów jak: Oracle, DB2, SQL/400 (AS/400), Informix, Sybase i Ingres. Dzięki temu w zależności od potrzeb można wykorzystywać odpowiednio mocny serwer bazy danych. Przewiduje się, że w latach 1995 - 2000 około 80% sieci lokalnych będzie posiadało serwery baz danych. Nadchodzące lata będą zdecydowaną ofensywą baz SQLowych w środowisku sieci lokalnych. Bazy te stworzą nowe możliwości przezroczystego łączenia różnych środowisk w jedną całość. Wszystko to odzwierciedla zmiany i oczekiwania informatyki na świecie.

## LITERATURA

- [1] James R. Groff & Paul N. Weinberg : Using SQL, Osborne McGraw-Hill, 1990.

- [2] Alope Nath, *The Guide to SQLServer*, Addison-Wesley Publishing Company, Inc., 1990.
- [3] *SQLBase, Database Administrator's Guide*, Gupta Corporation, 1991.
- [4] *SQLBase, Language Reference Manual*, Gupta Corporation, 1991.

Recenzent: Dr inż. Andrzej Kwiecień

Wpłynęło do Redakcji 24 września 1993 r.

## Abstract

The first discussed topic refers to the history of SQL and specifies the pros and cons of using it as a language of databases.

Next chapter is devoted to the detailed description of SQLBase and its specific features. It also mentions next versions of SQLBase and its new characteristics: two-phase commit, stored procedures and triggers. SQLBase is an ideal server for NetWare, DOS, OS/2 and SUN UNIX platforms, as shown on the picture enclosed. New versions for UnixWare and Windows NT are to be launched in the next months.

Besides SQLBase database server Gupta Corporation provides a complete range of tools for developers. The most important is SQLWindows, full-featured 4GL, object-oriented development tool, described in the third chapter. It provides programmers with visual programming environment for designing application screens, forms and objects.

Quest, another graphical tool, is useful for both regular end-users and professional developers due to its simplicity. It brings a new level of personal productivity, business reporting power, and management control to enterprise-wide SQL environments. Quest users, as described in the fourth chapter, can query databases, create sophisticated business reports and perform data management functions.

The latest achievement of Gupta Corporation - TeamWindows - puts control of team programming back in the hands of the project manager. It enables sharing and reuse of application components, enforcement of standards across application projects and specialization of programming tasks. TeamWindows automates administrative tasks such as staffing, security, standards enforcement, and quality control of the development, test and production environments of an application.

Gupta SQLSystem is known for its open architecture. SQLNetwork connectivity software enables users to simultaneously access data residing on a wide variety of corporate and departmental databases: Sybase, Oracle, DB2, Informix, Ingres.