

Bazy danych – przegląd zagadnień

Z zagadnieniem konieczności przechowywania informacji spotykamy się często w życiu codziennym. Listy spotkań, terminarze, numery telefonów, adresy i telefony przyjaciół, listy personalne to tylko niektóre zestawy danych, jakie chcemy lub musimy przechowywać i z nich korzystać. Naturalne więc wydaje się pojawienie się urządzeń wspomagających przechowywanie i udostępnianie danych. Na przykład telefony komórkowe, a nawet niektóre modele zegarków naręcznych umożliwiają obsługiwanie listy telefonów. Oczywiście na polu przechowywania i obróbki danych nie mogło zabraknąć wkładu komputerów – wymarzonego urządzenia do tego celu.

Za *bazę danych* w informatyce uznaje się uporządkowany zbiór informacji. Oznacza to, że dane, zawarte w bazie charakteryzują się pewną strukturą. Utrzymanie tej struktury jest podstawowym zadaniem urządzenia przechowującego informacje. Jeśli urządzeniem jest komputer, jego działanie w znacznym stopniu wyznaczone jest przez wykonywany przez niego program. Stąd wniosek, że musi istnieć specjalny program, służący do obsługiwaniania baz danych.

Bazy danych są składnikiem każdego większego systemu informatycznego – począwszy od systemu księgowego w firmie przechowującego dane kilku pracowników, na wyszukiwarkach internetowych, przechowujących dane o milionach witryn internetowych skończywszy. Zasyte przeważnie w głębi systemów informatycznych bazy danych wykonują ogromną pracę, nie zawsze nawet zauważaną (któż zastanawia się, skąd biorą się wyniki wyszukiwań np. w wyszukiwarce Google?). Sytuacja ta bierze się stąd, że system obsługi bazy danych stanowi jedynie źródło danych, które są następnie prezentowane np. jako strona HTML wyświetlana w przeglądarce internetowej, po “opakowaniu” w estetyczne ramki, grafikę itp. przez inne komponenty systemu informatycznego.

Najpopularniejsza metoda tworzenia struktury baz danych oparta jest na tzw. *modelu relacyjnym*. Podstawowym założeniem modelu relacyjnego jest uznanie bazy danych za *zbiór powiązanych ze sobą tabel*. Tabele składają się z wierszy o takiej samej strukturze, które są w ramach danej tabeli unikalne. Struktura wierszy tabeli zdefiniowana jest poprzez specyfikację kolumn tabeli, dokonywaną przez podanie dla każdej z kolumn, unikalnej w ramach tabeli, nazwy oraz typu przechowywanych danych. Zatem tabela składa się z “komórek” o właściwościach określonych przez specyfikację kolumn tabeli, zgrupowanych w wiersze. Jako przykład rozważmy tabelę o nazwie “*personalia*”:

<i>ID</i>	<i>Imie</i>	<i>Nazwisko</i>
1	Adam	Drzewiecki
2	Jan	Kowalski
3	Anna	Nowak

Powiązania pomiędzy tabelami związane są z pojęciami *klucza głównego* oraz *klucza obcego*. Klucz główny jest to kolumna (lub zbiór kolumn), której wartości pozwalają na jednoznaczne zidentyfikowanie wiersza w tabeli. Przykładowo, w powyższej tabeli kolumna *ID* może być kluczem głównym. Podczas łączenia tabel pojawia się pojęcie *klucza obcego*: jest to kolumna (lub zbiór kolumn), w której wartość odpowiada wartości klucza głównego w innej tabeli. Dla przykładu w tabeli o nazwie *telefony*:

<i>IDT</i>	<i>IDN</i>	<i>kierunkowy</i>	<i>numer</i>
1	2	68	3280001
2	1	68	3280002
3	3	68	3280003
4	1	68	3282907

kolumna *IDN* stanowi klucz obcy: pozwala w połączeniu w kluczem głównym tabeli *personalia* (czyli kolumną *ID*) powiązać numer telefonu z daną osobą. Powiązanie to zrealizowane jest poprzez

wpisanie wartości klucza obcego w wierszu tabeli, równej wartości w kluczu głównym powiązanego wiersza w drugiej z tabel. Na przykład wartości w tabelach informują, że Adam Drzewiecki ma dwa numery telefonów: 3280002 oraz 3282907, zaś np. Anna Nowak tylko jeden, 3280003. Tabela *telefony* ma również klucz główny, którym jest kolumna IDT.

Obsługa systemu baz danych, opartych o model relacyjny, realizowana jest przez specjalistyczny program, nazywany *Systemem Zarządzania Relacyjnymi Bazami Danych*, w skrócie RDBMS (ang. Relational DataBase Management System). Przykładem RDBMS jest program *MySQL*. Zadaniem RDBMS jest między innymi: przechowywanie danych, zawartych w bazach, umożliwianie wstawiania, usuwania i modyfikacji danych oraz udostępnianie danych na żądanie. Ponieważ do danej bazy danych na ogół powinno mieć jednoczesny dostęp wiele osób lub programów, systemy RDBMS są bardzo często realizowane w tzw. *architekturze klient-serwer*. Pokróćce mówiąc oznacza to, że specjalistyczny program lub wręcz komputer zwany *serwerem* realizuje zadania, których żądają *klienci*, czyli programy komputerowe, których naczelnym zadaniem jest nawiązanie połączenia z serwerem i przekazanie mu żądań użytkownika, a następnie odebrania wyników, zwróconych przez serwer, który realizuje żądania, sprawdzając możliwość realizacji danych żądań, uprawnienia do wykonywania pewnych operacji (np. sprzątaczką nie powinna mieć możliwości obniżenia pensji dyrektorowi) i tym podobne. W zagadnieniu baz danych serwer realizuje prowadzenie baz danych oraz udostępnia wyniki, czyli żądane informacje z baz danych (np. “podaj kwotę zarobków pracownika o nazwisku Kowalski”), natomiast klient pozwala połączyć się z serwerem, pobrać od użytkownika komendy, skierować je do serwera oraz wyświetlić odpowiedź serwera.

Do komunikacji z serwerami baz danych w przeważającej liczbie systemów RDBMS stosowany jest język *SQL* (ang. *Structured Query Language*) – strukturalny język zapytań. Każdy z systemów *RDBMS* oferuje składnię języka *SQL*, zbliżoną do jednej z kilku standardowych wersji tego języka. Różnice wynikają z występowania rozszerzeń standardu języka i/lub z braku implementacji niektórych jego elementów. Praca z bazą danych sprowadza się wtedy do wydawania poleceń, kierowanych do systemu *RDBMS*, wyrażonych w języku *SQL*, w szczególności *zapytań* (ang. *Query*, z racji brzmienia angielskiej nazwy zapytania są nazywane czasem *kwerendami*), czyli instrukcji służących do uzyskiwania danych z bazy, oraz oczekiwaniu na ich wyniki.

Kilka prostych przykładów zapytań języka *SQL*, które można zastosować do przykładowych tabel:

SELECT * FROM personalia

wybierz zawartość wszystkich kolumn z tabeli personalia – wyświetli wszystkie wiersze tabeli personalia

SELECT imie, nazwisko FROM personalia

wybierz kolumny imie oraz nazwisko z tabeli personalia – wyświetli listę imion i nazwisk z tabeli personalia

SELECT imie, nazwisko, numer FROM personalia, telefony WHERE ID=IDN

wybierz z połączenia tabel personalia oraz telefony dane z kolumn imie, nazwisko i numer, dla których wartości w kolumnie ID są równe IDN – wyświetli listę imion, nazwisk i numerów telefonów, realizując powiązanie klucza obcego IDN tabeli telefony z kluczem głównym tabeli personalia