

Podstawowe pojęcia

Baza danych

Baza danych jest logicznie spójnym zbiorem danych posiadających określone znaczenie. Precyzyjniej będzie jednak powiedzieć, że baza danych jest informatycznym odwzorowaniem danego fragmentu rzeczywistości, odzwierciedlającym stan tego fragmentu w postaci danych komputerowych. Przykładowo - baza danych, w której są ewidencjonowani pracownicy przedsiębiorstwa nie może zawierać informacji o osobach, które w danym przedsiębiorstwie nie pracują.

Baza danych jest projektowana, budowana i zapełniana danymi dla określonego celu.

Użytkownicy bazy danych:

- użytkownicy (którzy korzystają z danych, tworząc formularze, zestawienia, itp.)
- projektanci (definiują oni strukturę bazy oraz opracowują programy umożliwiające korzystanie z bazy danych)
- procesy (proces to najogólniej wykonywany aktualnie program).

System zarządzania bazą danych SZBD (ang. DBMS -Database Management System)

Jest to oprogramowanie umożliwiające tworzenie, eksploatację bazy danych oraz obsługujące użytkowników bazy.

Definiowanie bazy danych polega na specyfikacji typów przechowywanych w niej danych, określeniu użytkowników i ich praw dostępu do danych. Specyfikacja ta zapamiętywana jest na nośniku kontrolowanym przez SZBD.

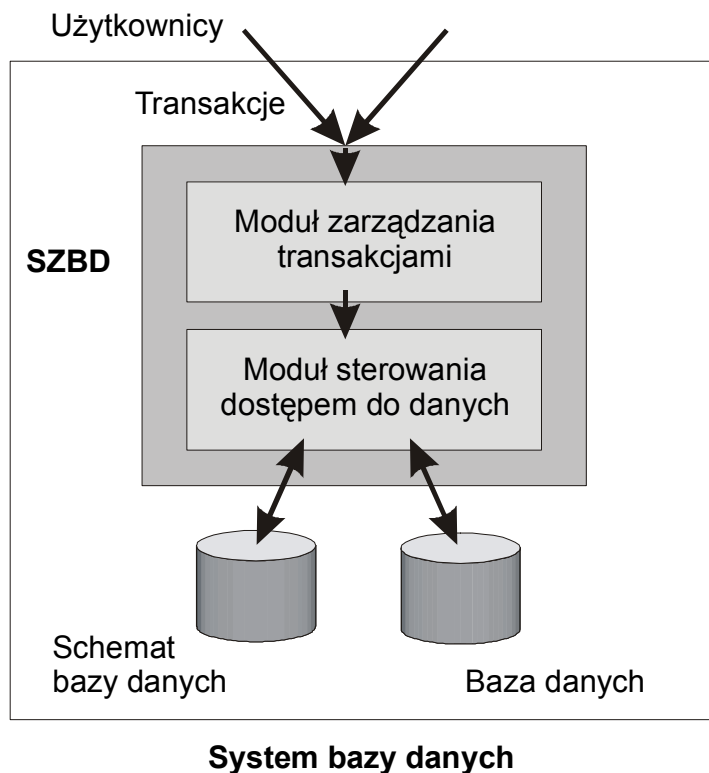
Przetwarzanie bazy danych polega na generowaniu zapytań w celu wyszukania potrzebnych danych, aktualizacji danych zgodnie ze zmianami zachodzącymi w odwzorowanym świecie i wydawaniu raportów o danych interesujących użytkownika.

Podstawowe funkcje jakie musi spełniać SZBD to:

- *optymalizacja zapytań* - takie przekształcanie zapytań kierowanych do bazy przez jej użytkowników aby czas oczekiwania na odpowiedź był możliwie najkrótszy,
- *zapewnienie integralności danych* - uniemożliwienie przejścia bazy do stanu, który nie istnieje w modelowanej rzeczywistości (jeśli baza danych przyjęła stanu, którego nie da się osiągnąć w modelowanej rzeczywistości, to mówimy, że jest ona w stanie niespójnym).
- *zapewnienie wielodostępu do bazy danych* – umożliwienie korzystania z bazy danych wielu użytkownikom w taki sposób, aby każdy użytkownik był przekonany, że jest wyłącznym właścicielem danych (współbieżność),
- *odporność na awarie* (niezawodność bazy danych) - możliwość odtworzenia poprawnego stanu bazy danych przed awarią,
- *ochrona danych* - uniemożliwienie dostępu nieuprawnionych użytkowników do poufnych danych innych użytkowników.

System bazy danych

Jest to, w uproszczeniu, baza danych plus system zarządzania bazą danych.



Rys.1. Schemat funkcjonalny systemu bazy danych

Transakcja

Transakcja jest pewną sekwencją instrukcji, po wykonaniu której baza danych z jednego stanu spójnego przechodzi w inny stan spójny. Transakcja jest operacją atomową, co oznacza, że SZBD powinien wykonać wszystkie instrukcje wchodzące w jej skład lub żadną. W praktyce SZBD wykonuje instrukcje transakcji po kolei, a w przypadku niepowodzenia którejs z nich wycofuje poprzednio wykonane instrukcje (jest to tzw. wycofanie transakcji, ang. *rollback*).

Moduł zarządzania transakcjami

Umożliwia jednoczesne korzystanie z bazy danych wielu użytkownikom.

Moduł zarządzania dostępem do danych

Umożliwia właściwą interpretację danych przechowywanych w pamięciach zewnętrznych, wprowadzanie nowych danych oraz modyfikację i aktualizację istniejących danych.

Schemat bazy danych

Dane zawarte w bazie opisują cechy modelowanych obiektów rzeczywistych. *Schemat* jest opisem *struktury* (formatu) przechowywanych danych oraz ich wzajemnych powiązań. Schemat bazy danych pozwala nam „wyciągnąć” z bazy interesujące nas informacje.

Cechy charakterystyczne baz danych

- *Niezależność aplikacji i danych*
Dane mogą być wprowadzane do bazy bez konieczności modyfikacji korzystających z nich programów czy systemów użytkowych, a z drugiej strony aplikacje mogą być modyfikowane niezależnie od stanu baz danych.
- *Abstrakcyjna reprezentacja danych.*
Programy i systemy użytkowe (aplikacje) są tworzone przy użyciu tzw. deklaratywnych języków programowania (w odróżnieniu od języków imperatywnych). Twórca aplikacji nie musi np. interesować się kolejnością danych w bazie, ani sposobem ich reprezentacji i wyszukiwania. Precyzuje jedynie warunki selekcji informacji. Inaczej mówiąc: decyduje „co zrobić”, a nie „jak zrobić”.
- *Różnorodność sposobów widzenia danych.*
Te same dane zawarte w bazie mogą być „widziane” w różny sposób przez różnych użytkowników. Efekt ten uzyskuje się przez stosowanie różnych „filtrów” (perspektyw) nakładanych na te same dane.
- *Fizyczna i logiczna niezależność danych.*
Fizyczna niezależność danych polega na tym, że rozszerzenie systemu komputerowego, na którym pracuje SZBD o nowy sprzęt nie narusza danych w bazie. Logiczna niezależność danych polega na tym, że - po pierwsze wprowadzanie nowych danych do bazy nie deaktualizuje starych, po drugie - dane, które nie są wzajemnie powiązane tzw. więzami integralności mogą być usuwane z bazy niezależnie od siebie.

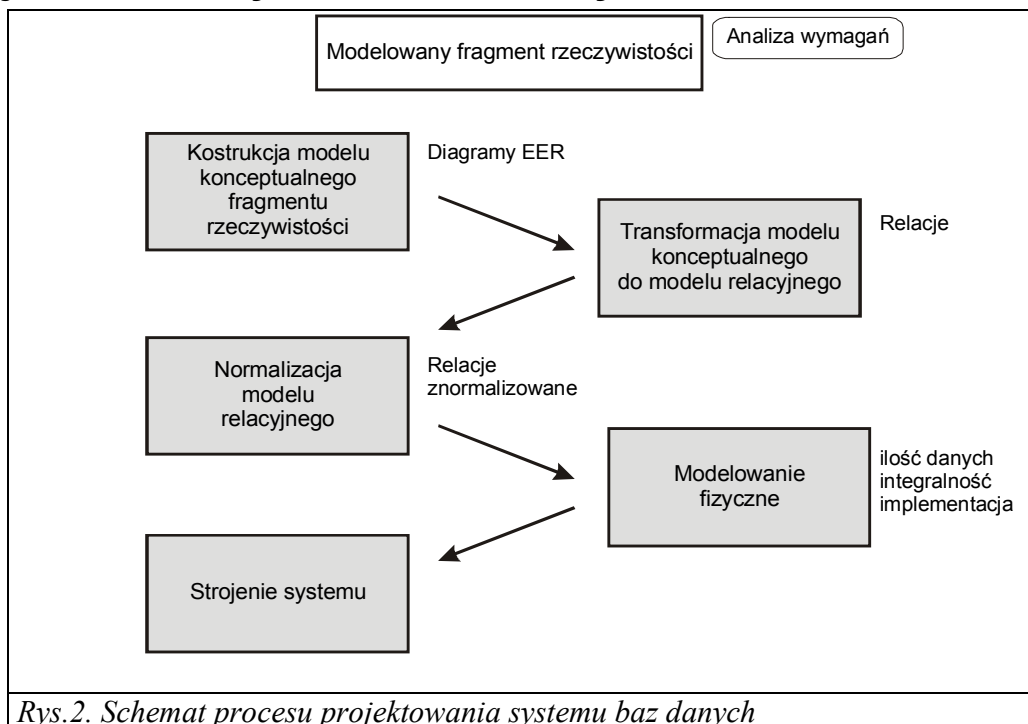
Języki baz danych

Do operowania na bazach danych służą następujące języki:

- język definiowania danych (ang. DDL - Data Definition Language) umożliwiający definiowanie struktury danych przechowywanych w bazie, czyli schematu bazy danych
- język manipulowania danymi (ang. DML - Data Manipulation Language) umożliwiający dodawanie, modyfikowanie i usuwanie informacji w bazie danych
- język sterowania danymi (ang. DCL - Data Control Language) umożliwiający sterowanie transakcjami
- język zapytań (ang. QL - Query Language) - umożliwiający pobieranie informacji z bazy danych.

W praktyce te cztery języki są ze sobą zintegrowane. Takim zintegrowanym językiem jest m.in. SQL (ang. *Structured Query Language*). Do listy języków można tu jeszcze dodać rozszerzenia proceduralne stosowane przez różne firmy produkujące SZBD: pl/pgsql w PostgreSQL, PL/SQL w Oracle i inne.

Projektowanie systemów baz danych



Rys.2. Schemat procesu projektowania systemu baz danych

Etapy projektowania:

1. Wyróżnienie fragmentu rzeczywistości, który chcemy odwzorować w bazie danych.
2. Analizy tego fragmentu i budowa jego modelu konceptualnego reprezentowany przez tzw. diagramy związków i encji. Podstawowymi elementami diagramów tzw. *modelu rozszerzonego* (EER - *Extended Entity Relationship*) są encja, atrybut i związek.
3. Transformacji modelu konceptualnego do modelu relacyjnego, którego trzy podstawowe składowe to:
 - relacyjne struktury danych,
 - operatory algebry relacyjnej, które umożliwiają tworzenie, przeszukiwanie i modyfikowanie danych
 - ograniczenia (więzy integralności), które jawnie lub niejawnie określają możliwe/dopuszczalne wartości danych.
4. Normalizacja modelu relacyjnego (wiążąca się z zapewnieniem optymalności i poprawności)
5. Modelowanie fizyczne, które obejmuje szczegóły związane z konkretną implementacją systemu baz danych

Elementy modelu konceptualnego

Encja

Jest to pewna rzecz (obiekt), o której chcemy przechowywać informacje. Ogólnie można wyróżnić kilka rodzajów takich obiektów:

- obiekty materialne - osoba (imię, nazwisko, adres, numer PESEL), samochód (model, numer fabryczny, kolor, zużycie paliwa),
- obiekty koncepcyjne – rachunek bankowy, stanowisko służbowe,
- zdarzenia - wysłanie towaru (data wysłania, nazwa towaru, symbol, nazwa i adres odbiorcy),
- fakty - znajomość języka (nazwa języka, czas nauki, stopień znajomości).

Odpowiednikiem obiektu w modelu EER jest encja (ang. *entity*). Encja posiada pewne charakterystyczne dla siebie cechy, które występują w modelu jako atrybuty.

Encje tego samego typu można grupować w zbiory encji. Każda z encji jest jednoznacznie określona za pomocą nazwy. Encje pochodzące z różnych zbiorów różnią się nazwami, a przede wszystkim własnościami. Encje niosą ze sobą niepełną informację o modelowanym wycinku rzeczywistości.

Niech np.

Pracownicy to zbiór zawierający m.in. encje **Kazimierz** i **Anna**

Stanowiska to zbiór zawierający encje **kierownik**, **pracownik**, **zaopatrzeniowiec**.

Sam fakt przynależności encji do określonych zbiorów nie opisuje wystarczająco dokładnie sytuacji w modelowanym przedsiębiorstwie. W systemie należałoby zapisać m.in. informację o stanowiskach, jakie zajmują poszczególni pracownicy. Jeżeli **Kazimierz** jest zatrudniony na etacie **kierownik**, a **Anna** - na etacie **zaopatrzeniowca**, to do umieszczenia tej informacji w bazie danych należy wprowadzić związek o nazwie `pracuje_na_etacie` pomiędzy zbiorami encji **Pracownicy** a **Stanowiska**.

Procedura konstrukcji modelu konceptualnego fragmentu rzeczywistości składa się zwykle z trzech kroków:

- Określenie wymagań dla systemu informatycznego z punktu widzenia zamawiającego.
- Zdefiniowanie informacji o obiektach i ich wzajemnych związkach. Rezultatem jest kompletny diagram związków i encji.
- Określenie hierarchii funkcji realizowanych w systemie informatycznym.

Po opracowaniu zbioru diagramów następuje przekształcenie modelu konceptualnego w model implementacyjny - najczęściej jest to model relacyjny. Wynikiem tego przekształcenia są tzw. schematy relacji.

Projektowanie logicznej struktury relacyjnej bazy danych polega na zdefiniowaniu schematu logicznego bazy danych, czyli na zdefiniowaniu schematów relacji tworzących tą bazę w taki sposób, aby posiadały one pożądane własności ułatwiające eksploatację bazy danych.

Relacyjny model danych

Relacyjny model danych posiada trzy podstawowe składowe:

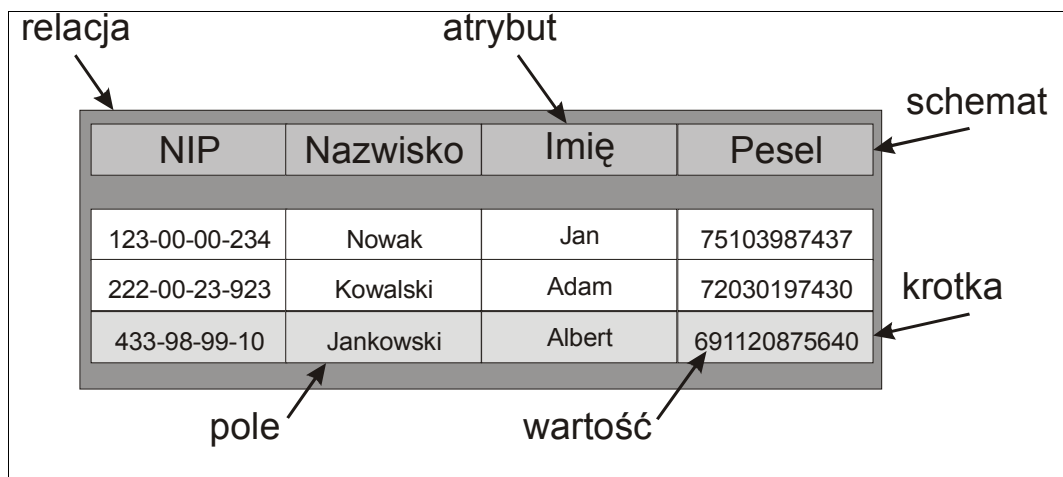
- relacyjne struktury danych
- operatory algebry relacyjnej, które umożliwiają tworzenie, przeszukiwanie i modyfikowanie danych
- ograniczenia (więzy) integralnościowe jawnie lub niejawnie określającymi możliwe/dopuszczalne wartości danych.

Podstawową strukturą danych modelu relacyjnego jest relacja, będąca podzbiorem iloczynu kartezyjskiego wybranych domen i przedstawiana w postaci dwuwymiarowej tabeli. W tym ujęciu pojęcia relacji (ang. *relation*) i tabeli (ang. *table*) są synonimami. Pojęcie relacji używane jest na poziomie teorii baz relacyjnych i modelu konceptualnego, a pojęcie tabeli na poziomie realizacji fizycznej.

Relację, nieformalnie rzecz biorąc, definiuje się za pomocą predykatu jako pewien zbiór krotek (ang. *tuples*) (rekordów) posiadających taką samą strukturę (schemat) i różne wartości. Zbiór ten jest przedstawiany w postaci wierszy tablicy, tzn. na poziomie fizycznym krotka jest przedstawiana jako wiersz tabeli, zawierający wartość co najmniej jednego atrybutu o określonej dziedzinie (kolumny tablicy). Inaczej mówiąc: każda krotka danej relacji ma dokładnie taką samą strukturę, zgodną ze schematem relacji. Każda krotka posiada przynajmniej jeden atrybut (ang. *attribute*), który jest przedstawiany w postaci kolumny.

Relacja posiada następujące właściwości:

- wszystkie jej krotki są różne
- wszystkie jej atrybuty są różne
- kolejność krotek nie ma znaczenia i w ogólności nie jest ona znana
- kolejność atrybutów nie ma znaczenia
- wartości atrybutów są niepodzielne (atomowe), tj. nie mogą być zbiorem wartości.



Z punktu widzenia semantyki (znaczenia) rozróżnia się dwa pojęcia - pojęcie *intensji relacji* oraz *ekstensji relacji*.

Intensja relacji odpowiada znaczeniu relacji - predykat związany z relacją stanowi element jej intensji. Na intensję relacji mogą poza tym składać się inne jej własności.

Ekstensja relacji jest zbiorem krotek spełniających własności, które stanowią intensję relacji. Własności te nazywa się *warunkami integralności*.

Dla relacji $R(\text{PRZEDMIOT, DZIEŃ, GODZINA, SALA, WYKŁADOWCA})$:

- *intensję relacji* wyrażają następujące warunki integralności:

1. predykat związany z relacją R mówi o tym, że „wykładowca w prowadzi zajęcia z przedmiotu p w dniu d , o godzinie g , w sali s ”;
2. dziedzina atrybutu $GODZINA$ jest zbiorem liczb całkowitych z przedziału $\langle 7, 24 \rangle$;
3. dany wykładowca może znajdować się o określonej godzinie tylko w jednej sali.

- *ekstensję relacji* R stanowi zbiór krotek spełniających własności (1), (2), (3).

Schematy relacji definiuje projektant bazy danych, natomiast relacje składają się na bieżącą realizację bazy danych. Szczególnie istotnymi warunkami integralności są zależności funkcyjne. Pojęcie zależności funkcyjnej wiąże się z pojęciem *klucza relacji*. Klucz relacji stanowi najmniejszy zbiór atrybutów relacji, których wartości pozwalają wskazać jednoznacznie każdą z krotek relacji.