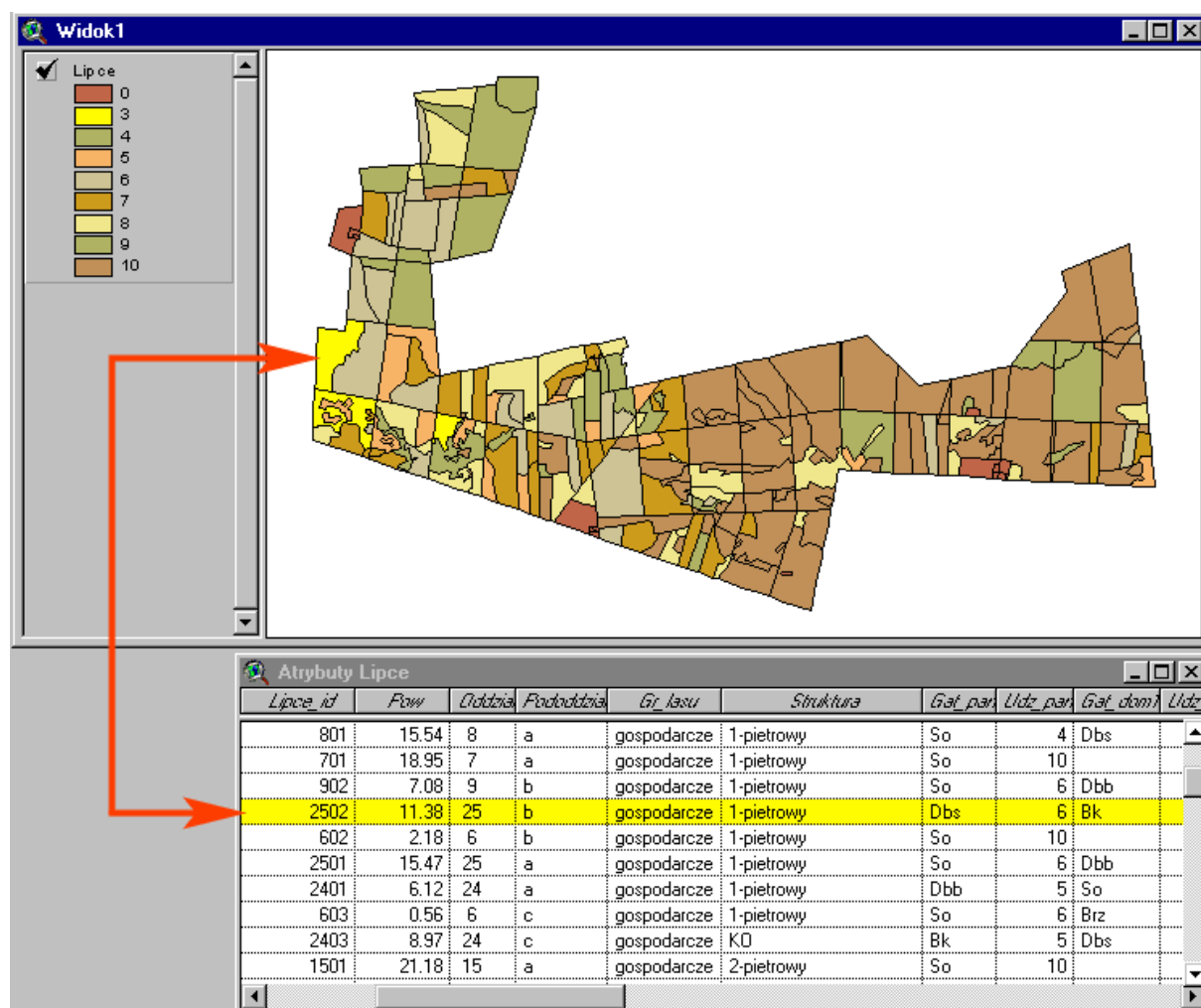


## Bazy danych – podstawy teoretyczne

Baza danych (ang. database) jest zbiorem informacji (danych). W dzisiejszych czasach pojęcie to nie zawsze prawidłowo utożsamiane jest również jako plik zawierający konkretną bazę danych przechowywaną w pamięci komputera. Baza danych to także rodzaj jednego z najbardziej rozpowszechnionego oprogramowania, za którego pomocą możemy tworzyć banki ściśle uszeregowanych tekstów, grafik, dźwięku lub klipów wideo.

Prawidłowo zbudowane systemy informacji przestrzennej bazują na dwóch ściśle powiązanych ze sobą elementach:

1. Informacji przestrzennej zawierającej dane o położeniu, geometrycznych właściwościach i przestrzennych relacjach interesujących nas obiektów
2. Bazie opisowej, w której zgromadzone są różnego typu dane dotyczące obiektów znajdujących się w zasięgu tworzonego przez użytkownika systemu informacji przestrzennej.



Większą część czasu przy tworzeniu systemów informacji przestrzennej zajmuje zbudowanie poprawnej i funkcjonalnej bazy danych. Ich głównym zadaniem będzie bowiem wspomaganie procesów decyzyjnych, dokonywanie różnorodnych analiz i symulacji oraz dawanie użytkownikom odpowiedniej informacji o odpowiedniej jakości

(wiarygodności i spójności) w odpowiednim miejscu i w odpowiednim czasie. Dostęp do danych powinien być łatwy i pewny.

Spełnienie tych wszystkich warunków dla obszarów o skomplikowanej strukturze i dla wielu zagadnień tematycznych pochłania ogromny nakład pracy. Wiąże się również z koniecznością zaangażowania dużych środków finansowych poniesionych na urządzenia udostępniające potrzebne informacje, niezbędne oprogramowanie, czy wreszcie obsługę i ciągłą aktualizację systemu.

Podejmując zatem decyzję o tworzeniu bazy danych trzeba w pierwszej kolejności poznać podstawowe pojęcia z zakresu jej budowy, dokładnie określić swoje oczekiwania względem tworzonego systemu i starannie zaplanować strukturę bazy.

## **Pojęcia podstawowe**

Wszystkie informacje na temat interesujących nas obiektów są przechowywane w tabeli. Z czasem okazało się jednak, że pojedyncza tabela jest niewystarczająca, aby gromadzić coraz większe ilości informacji z różnych dziedzin. Dlatego też zaczęto tworzyć bardziej skomplikowane systemy przechowywujące dane, wśród których jednym z częściej używanych jest relacyjny model danych. W systemie tym, baza danych składa się zwykle z szeregu tabel (relacji) oraz związków pomiędzy tabelami.

Związki umożliwiają nam łączenie tabel ze sobą w celu otrzymania pełnej informacji.

Bazy utworzone zgodnie z powyższym opisem są obecnie powszechnie stosowane w systemach informacji przestrzennej i noszą nazwę relacyjnych baz danych.

Typy danych (ang. data type) - czyli forma zapisu informacji może być bardzo różna – na przykład:

- znakowy (ang.character) – dana może przybierać tylko wartości znaków pisarskich
- liczbowy (ang.number) – dana może przechowywać tylko liczby
- logiczny (ang.logical) – dana może przybierać tylko dwie wartości: prawda, fałsz
- data (ang.date) – dana może przyjmować postać daty i czasu np. rok.miesiąc.dzień godz:min:sek
- alfanumeryczny (ang.alphanumeric) – dana może przybierać wartości znaków ASCII\* oraz cyfry
- numeryczny (ang.numeric) – wartościami danej mogą być tylko cyfry i znaki: + (plus), - (minus).
- walutowy (ang.currency) – dana może przyjmować wartości liczbowe razem z symbolem waluty np.
- notatnikowy (ang.memo) – dana może być oddzielnym zbiorem tekstowym służącym do przechowywania dowolnych opisów.
- binarny (ang.binary) – dana może być np. plikiem dźwiękowym lub filmowym.
- graficzny (ang.graphic) – dana przechowuje grafikę np. rysunki.
- obiektowy (ang.OLE) – dana przechowuje obiekty do których dostęp dokonuje się za pomocą techniki OLE (ang. object linking and embedding), czyli obiektów tworzonych przez inne aplikacje.

\*ASCII (American Standard Code for Information Interchange) – jeden z najbardziej popularnych formatów zapisu plików tekstowych. W ten sposób zakodowanych jest 256 różnych znaków.

## Budowa tabeli

Rekord (patrz rys. kolor niebieski) - to pojedynczy wiersz tabeli danych; zwykle opisuje jeden obiekt, lub fragment obiektu

Pole (patrz rys. pole szare) - to część rekordu; przechowuje pewną niepodzielną informację

Klucz (patrz rys. kolor żółty) - to kolumna lub kilka kolumn; pozwalających na jednoznaczne zidentyfikowanie danego rekordu

Klucz obcy – kolumna lub kolumny, której odpowiada klucz w innej tabeli

Kolumna (atrybut) (patrz rys. kolor zielony) - to zbiór pól rekordów, które posiadają identyczne charakterystyki:

- nazwę,
- typ przechowywanych danych,
- szerokość

Tabela OPIS

LIPCE-ID	ODDZIAŁ	PODODDZIAŁ	UDZ_PAN	GAT_DOM	GAT_PAN	KL_WIEKU
101	1	a	4	60	42	IIb
102	1	b	9	1	1	IIIa
201	2	a	8	42	1	IIb
204	2	d	6	1	60	IIb
202	2	b	4	1	1	IIIa
203	2	c	8	60	1	IIb

Tabela KODY GATUNKÓW

KOD_GAT	SKRÓT	NAZWA
1	So	Sosna pospolita
.....		
42	Dbb	Dąb bezszypułkowy
.....		
60	Brz	Brzoza brodawkowata
.....		

**ZWIĄZEK**



## Projektowanie baz danych

Projekt liczby i struktury tabel oraz definicje związków następują w procesie projektowania bazy danych. W fazie projektowania ustalane są rodzaje danych, które użytkownicy chcą przechowywać w bazie. Sposób podziału na tabele i związki jest ustalany w procesie normalizacji bazy danych

Normalizacja danych to takie uporządkowanie tablic, aby były one maksymalnie wydajne i zwarte. Unikamy dzięki temu błędów i nieporozumień podczas pracy z relacyjną bazą danych. Istnieje pięć zasad normalizacji danych, ale najważniejsze są dwie pierwsze:

### 1. Usuwanie powtarzania się informacji

Nazwa lub nazwisko klienta	Adres	Telefon	Data złożenia zamówienia	Wartość zamówienia
Sun Microsystems	201 W. 44 <sup>th</sup> St.	(317) 555-2394	2/5/98	\$195.90
Sun Microsystems	201 W. 44 <sup>th</sup> St.	(317) 555-2394	5/14/98	\$90.24
Sun Microsystems	201 W. 44 <sup>th</sup> St.	(317) 555-2394	8/8/98	\$1200.01
ESRI Inc.	1155 Redlands	(812) 456-9725	5/16/98	\$900.83
ESRI Inc.	1155 Redlands	(812) 456-9725	9/9/98	\$1800.00
Mike's Pizza	108 Corner Ave	(672) 243-4789	9/10/98	\$3500.00

Zakładamy, że chcemy mieć pod dostęp do informacji na temat wszystkich klientów wraz z wykazem transakcji kupna-sprzedaży. Przechowywanie tych informacji w tylko jednej tabeli, zmuszałoby do wpisywania danych klienta przy każdorazowym wprowadzaniu nowej transakcji. W sytuacji zmiany adresu któregoś z klientów trzeba by było wprowadzać konieczne zmiany do każdej transakcji z osobna.

Lepszym rozwiązaniem jest stworzenie dwóch niezależnych tabel grupujących oddzielnie adresy wszystkich klientów oraz zawartych z nimi transakcji. Każdemu z klientów nadajemy unikalny identyfikator wprowadzając jego dane tylko raz do tabeli KLIENCI. Aktualizując tabelę ZAMÓWIENIA wprowadzamy jedynie jego identyfikator. Uzyskanie zaś informacji na temat wszystkich klientów wraz z wykazem transakcji kupna-sprzedaży uzyskujemy po połączeniu ze sobą obu tabel za pomocą związku.

Tabela KLIENCI

Identyfikator Klienta	Nazwa lub nazwisko klienta	Adres	Telefon
1	Sun Microsystems	201 W. 44 <sup>th</sup> St.	(317) 555-2394
2	ESRI Inc.	1155 Redlands	(812) 456-9725
3	Mike's Pizza	108 Corner Ave	(672) 243-4789

Tabela ZAMÓWIENIA

Identyfikator Klienta	Data złożenia zamówienia	Wartość zamówienia
1	2/5/98	\$195.90
1	5/14/98	\$90.24
1	8/8/98	\$1200.01
2	5/16/98	\$900.83
2	9/9/98	\$1800.00
3	9/10/98	\$3500.00

## 2. Unikanie nadmiarowych danych

Nazwisko pracownika	Adres	Telefon	Data szkolenia	Nazwa kursu	Liczba godzin	Zaliczenie
Phil Sharp	29 W. 67 <sup>th</sup> St.	4569725	6/15/98	Intermediate	3	Tak
Becky Rowan	19 E. 52 <sup>th</sup> St.	6534712	2/5/98	First Certificate	2	Tak
Nick Gianti	1 N. 4 <sup>th</sup> St.	5929103	2/5/98	Advanced	9	Tak
Martha Red	34 W. 2 <sup>nd</sup> St.	9929581	6/15/98	Advanced	9	Nie
Cynthia Welsh	21 S. 49 <sup>th</sup> St.	1895910	6/15/98	First Certificate	2	Tak
Bill Gates	28 W. 22 <sup>th</sup> St.	2957948	2/5/98	Intermediate	3	Tak

Tabela PRACOWNICY

IDENTYF. pracownika	Nazwisko pracownika	Adres	Telefon
1	Phil Sharp	29 W. 67 <sup>th</sup> St.	4569725
2	Becky Rowan	19 E. 52 <sup>th</sup> St.	6534712
3	Nick Gianti	1 N. 4 <sup>th</sup> St.	5929103
4	Martha Red	34 W. 2 <sup>nd</sup> St.	9929581
5	Cynthia Welsh	21 S. 49 <sup>th</sup> St.	1895910
6	Bill Gates	28 W. 22 <sup>th</sup> St.	2957948

Tabela SZKOLENIA

IDENTYF. Kursu	Data szkolenia	Nazwa kursu	Liczba godzin
C1	6/15/98	Intermediate	3
C2	2/5/98	First Certificate	2
C3	2/5/98	Advanced	9

Tabela UCZESTNICTWO W SZKOLENIU

IDENTYF. pracownika	Nazwa kursu	Zaliczenie
1	C1	Tak
2	C2	Tak
3	C3	Tak
4	C3	Nie
5	C2	Tak
6	C1	Tak

Założmy, że w firmie prowadzone są różnego typu szkolenia językowe dla pracowników. Wszystkie informacje na ten temat można oczywiście umieścić w jednej tabeli. Co się jednak dzieje, gdy któryś z pracowników został zwolniony z pracy? Musimy usunąć jeden rekord z naszej tabeli tracąc tym samym informację o liczbie godzin, które firma przeznaczyła na szkolenia. Liczbę kolumn w tabelach należy zatem ograniczyć do niezbędnego minimum, zaś wszelkie analizy dokonywać na bazie stworzonej z wielu różnych tabel połączonych za pomocą unikalnego identyfikatora.

Prawidłowe zaprojektowanie bazy danych wpływa na jej efektywne użytkowanie, aktualizowanie i poszerzanie o nowe informacje. Kluczem do sukcesu jest właściwa organizacja tabel tworzonych według następujących zasad:

- każda tabela powinna mieć jakiś temat przewodni i powinna mieć jedną nazwę

- należy unikać powtarzania się danych w tabelach poprzez odpowiednie ich podzielenie
- metodą ułatwiającą łączenie ze sobą poszczególnych tablic jest stosowanie unikalnych identyfikatorów redukujących konieczność wielokrotnego wprowadzania długich ciągów znaków

## ***Podstawowe operacje na relacyjnych bazach danych***

Po podzieleniu danych na tabele i zdefiniowaniu pól kluczy podstawowych trzeba wprowadzić do systemu bazy danych informacje na temat sposobu poprawnego łączenia powiązanych danych w logiczną całość. W tym celu definiuje się związki między tabelami.

### **Typy związków**

#### **1. Związek jeden-do-jednego**

W związku jeden-do-jednego każdy rekord w tabeli A może mieć tylko jeden dopasowany rekord z tabeli B, i tak samo każdy rekord w tabeli B może mieć tylko jeden dopasowany rekord z tabeli A. Ten typ relacji spotyka się rzadko, ponieważ większość informacji powiązanych w ten sposób byłoby zawartych w jednej tabeli. Związek jeden-do-jednego można używać do podziału tabeli z wieloma polami, do odizolowania części tabeli ze względów bezpieczeństwa, albo do przechowania informacji odnoszącej się tylko do podzbioru tabeli głównej.

#### **2. Związek jeden-do-wielu**

Związek jeden-do-wielu jest najbardziej powszechnym typem związków.

W związku jeden-do-wielu rekord w tabeli A może mieć wiele dopasowanych do niego rekordów z tabeli B, ale rekord w tabeli B ma tylko jeden dopasowany rekord w tabeli A.

### **Sortowanie**

Sortowaniem rekordów nazywamy ich porządkowanie według jakiegoś kryterium.

Kryterium to nazwa lub nazwy pól według których odbywa się sortowanie.

Przykładowo może być to sortowanie rosnące (sortuje wartości w porządku rosnącym (od A do Z, od 0 do 9) lub sortowanie malejące (sortuje wartości w porządku malejącym od Z do A, od 9 do 0).

### **Zapytanie (ang. query)**

Zapytanie, to taka konstrukcja językowa, która pozwala na wyszukiwanie danych z bazy danych za pomocą zadawania pytań. Może to być specjalna konstrukcja języka programowania lub okno graficzne w którym należy podać parametry poszukiwanych danych.

## **Filtr**

Filtr pozwala na wyświetlanie rekordów spełniające pojedyncze kryterium lub wiele kryteriów albo sortować rekordy w porządku rosnącym lub malejącym.

## **System zarządzania relacyjnymi bazami (RDBMS)**

Jest to program zarządzający (system bazy danych) bazami danych oraz służący do wykonywania tabel, formularzy, zapytań, raportów, procedur, podprogramów oraz produkcji gotowych aplikacji.

Najpopularniejsze systemy zarządzania relacyjnymi bazami danych dla komputerów klasy IBM PC to: Oracle, Informix, Sybase, Progress, Visual FoxPro, Access.

## Structured Query Language – SQL

System zarządzania relacyjną bazą danych umożliwia jednej lub wielu osobom korzystanie z relacyjnej bazy danych. W większości systemów istnieje zaimplementowany język SQL, który jest standardowym językiem manipulacji i zapytań w relacyjnych bazach danych.

### Operatory używane w języku zapytań SQL

=	Równy
<> lub !=	nie równy
<	mniejszy od
>	większy od
<=	mniejszy lub równy
>=	większy lub równy

### Podstawowe zapytania SQL (FoxPro 2.5)

**select <kolumna> from <tabela> where <warunek>**

*Przykłady zapytań w języku SQL odnoszą się do dwóch tabel zamieszczonych poniżej.*

*Wybranie wydzieleń gdzie gatunkiem panującym jest dąb bezszypułkowy*

```
select * from opis where gat_pan = „Dbb”
```

*Wybranie wydzieleń o powierzchni większej niż 3ha, gdzie gatunkiem panującym jest dąb bezszypułkowy*

```
select * from opis where gat_pan = „Dbb” and pow > 3
```

*Zsumowanie powierzchni wszystkich wydzieleń, gdzie gatunkiem panującym jest dąb bezszypułkowy*

```
select sum(pow) from opis where gat_pan = „Dbb”
```

*Zestawienie powierzchni w rozbiciu na gatunek panujący*

```
select sum(pow), gat_pan from opis group by gat_pan
```

*Zestawienie powierzchni w rozbiciu na gatunek panujący i klasę wieku*

```
select sum(pow), gat_pan, kl_wieku from opis group by  
gat_pan, kl_wieku
```



Połączenie dwóch tabel w jedną według identyfikatora LIPCE\_ID i zapisanie wyniku w nowej tabeli LIPCE\_ALL

```
select * from opis, zabiegi where opis.lipce_id =
zabiegi.lipce_id into table lipce_all
```

*Tabela OPIS*

<i>Lipce_ID</i>	<i>Gat_pan</i>	<i>Gat_dom</i>	<i>Kl_wieku</i>	<i>Pow</i>
101	So	Brz	IVa	9.62
102	So	Sw	IIa	2.03
103	Sw	Brz	Ia	0.99
104	So	Sw	IIa	1.01
105	Ol	So	IIIb	2.9
106	So	Sw	VI	5.26
107	Dbb	Bk	IVb	6.15
108	Dbb	Bk	IIa	2.67
109	So	Sw	Ia	0.61
110	Dbb	Bk	VII	3.1

*Tabela ZABIEGI*

<i>Lipce_ID</i>	<i>Zabieg</i>
101	TP
102	TW
103	CW
104	TW
105	TP
106	TP
107	TP
108	TW
109	CW
110	TP