

Baza danych – definicja

Uporządkowany zbiór danych i oraz narzędzi systemu zarządzania bazą danych (ang. DBSM)

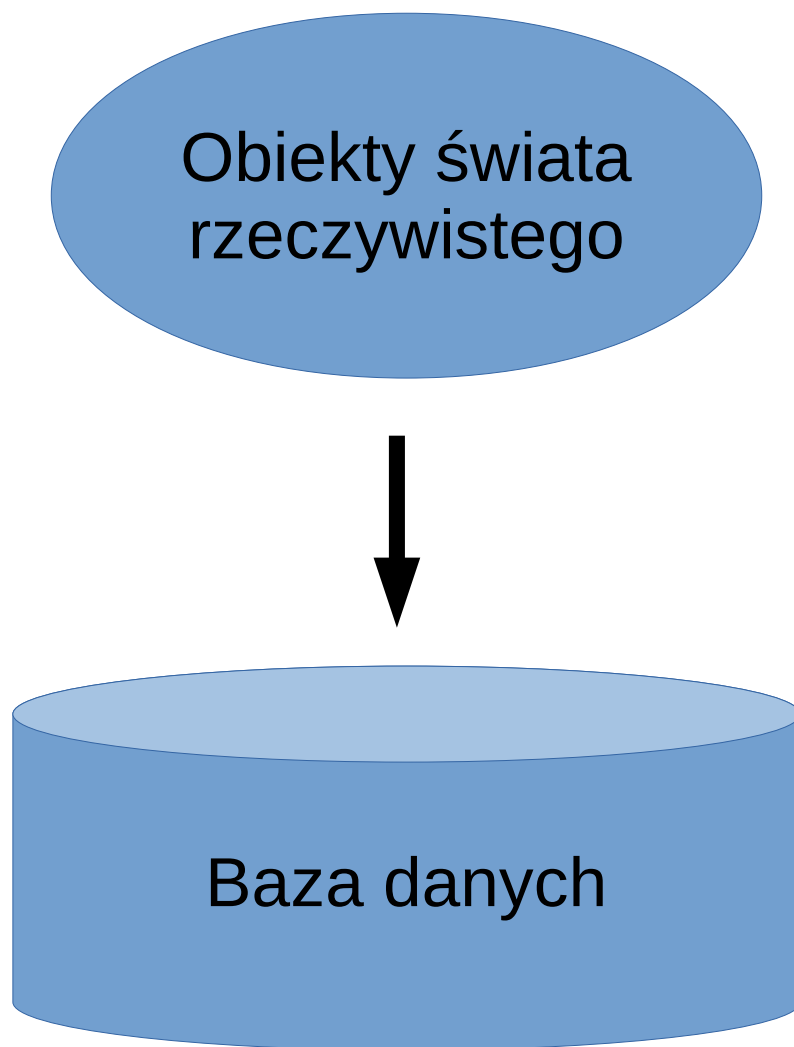


System zarządzania
bazą danych (SZBD)

Zbór danych

Przykłady baz danych

- Dziennik szkolny
- Spis kontaktów w telefonie
- Katalog biblioteczny
- Kartoteka pacjentów
- Rozkład jazdy autobusów
- Rejestr skradzionych pojazdów
- Rachunki bankowe
- ...



Cechy charakterystyczne bazy danych

- Trwałość danych
- Niezależność danych (fizyczny sposób przechowywania)
- Ochrona danych / autoryzacja dostępu
- Integralność / spójność
- Współbieżność dostępu
- Metadane (informacje o strukturze)

Przykłady systemów zarządzania bazą danych (SZBD)

- Komercyjne

- Oracle Database
- IBM DB2
- Microsoft SQL Server

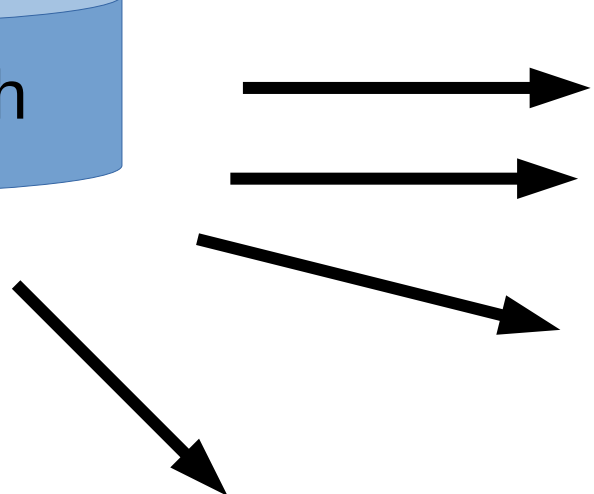


- Niekomercyjne

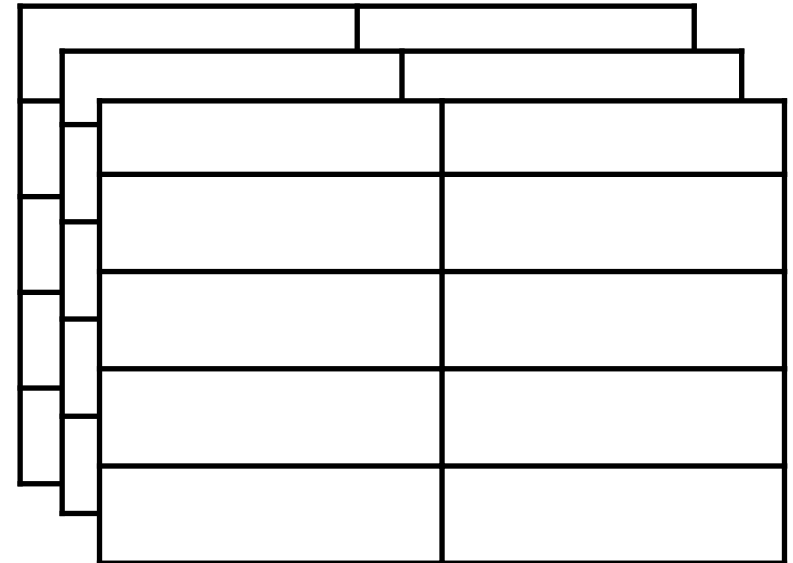
- MySQL
- PostgreSQL
- Firebird



Relacyjne bazy danych



Tabele



Encja = Relacja =
Tabela

Atrybut 1	Atrybut 2	Atrybut 3

Krotka 1 = Wiersz 1 →

Krotka 2 →

Krotka 3 →

Krotka 4 →

Relacyjne bazy danych

podstawowe pojęcia

ENCJA = RELACJA = TABELA = KLASA

Zbiór podobnych obiektów opisanych w jednolity sposób

KROTKA = REKORD = WIERSZ = OBIEKT

Zestaw wartości atrybutów opisujący jeden obiekt
identyfikowany przez wyróżnione atrybuty lub nazwę

ATRYBUT = KOLUMNA = POLE

Pojedyncza dana wchodząca w skład krotki

Przykładowa relacja

Encja (tabela)

Samochody

Krotka (rekord)

Marka	Model	Kolor
Audi	A5	Czerwony
Fiat	Panda	Niebieski
Citroen	C5	Biały
Bmw	535xd	Czarny

Atrybut (pole)

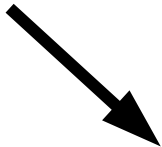
Przykładowa relacja (2)

Samochody

Marka	Model	Kolor
Audi	A5	Czerwony
Fiat	Panda	Niebieski
Fiat	Panda	Niebieski
Fiat	Panda	Niebieski
Citroen	C5	Biały
Bmw	535xd	Czarny

Klucz główny (primary key)

Klucz główny



Samochody

Nr_rej	Marka	Model	Kolor
TST0001	Audi	A5	Czerwony
TST0002	Fiat	Panda	Niebieski
TST9999	Fiat	Panda	Niebieski
TST1234	Fiat	Panda	Niebieski
TST9191	Citroen	C5	Biały
TST0004	Bmw	535xd	Czarny

Klucz główny - definicja

Klucz główny to jedna kolumna (lub zbiór kolumn) która w sposób jednoznaczny (unikatowy) identyfikują wiersz (krotkę) w danej tabeli.

Baza danych „Wypożyczalnia”

Klienci

ID	Imię	Nazwisko
1	Jan	Nowak
2	Michał	Kowalski
3	Paweł	Wójcik
4	Sebastian	Majewski

Samochody

Nr_rej	Marka	Model	Kolor
TST0001	Audi	A5	Czerwony
TST0002	Fiat	Panda	Niebieski
TST9999	Fiat	Panda	Niebieski
TST1234	Fiat	Panda	Niebieski
TST9191	Citroen	C5	Biały
TST0004	Bmw	535xd	Czarny

Baza danych „Wypożyczalnia” (2)

ID	Imię	Nazwisko
1	Jan	Nowak
2	Michał	Kowalski
3	Paweł	Wójcik
4	Sebastian	Majewski

Nr_rej	Marka	Model	Kolor
TST0001	Audi	A5	Czerwony
TST0002	Fiat	Panda	Niebieski
TST9999	Fiat	Panda	Niebieski
TST1234	Fiat	Panda	Niebieski
TST9191	Citroen	C5	Biały
TST0004	Bmw	535xd	Czarny

Wypożyczenia

ID	Klient	Samochód
1	1	TST0002
2	4	TST0004

Klucz obcy

Klucz obcy - definicja

Klucz obcy to jedna kolumna (lub zbiór kolumn) która jest jednocześnie kluczem głównym w innej tabeli.

Program XAMPP



X(cross-platform)

APACHE

MySQL

PHP

PERL

Windows, Linux, OSX

Serwer HTTP

SZBD

Interpreter języka PHP

Interpreter języka PERL

XAMPP – panel kontrolny

XAMPP Control Panel v3.2.1 [Compiled: May 7th 2013]

XAMPP Control Panel v3.2.1

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	3648 7796	80, 443	Stop Admin Config Logs
<input type="checkbox"/>	MySQL	6028	3306	Stop Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

1:31:04 PM [main] Checking for prerequisites
1:31:04 PM [main] All prerequisites found
1:31:04 PM [main] Initializing Modules
1:31:04 PM [main] Starting Check-Timer
1:31:04 PM [main] Control Panel Ready
1:35:03 PM [Apache] Attempting to start Apache app...
1:35:04 PM [Apache] Status change detected: running
1:35:10 PM [mysql] Attempting to start MySQL app...
1:35:11 PM [mysql] Status change detected: running

Język SQL

- Język SQL jest językiem deklaratywnym
- Język SQL jest zorientowany na przetwarzanie zbiorów
- Język SQL można podzielić na :
 - DDL (ang. Data Definition Language)
 - **CREATE, ALTER, DROP, TRUNCATE**
 - DML (ang. Data Manipulation Language)
 - **INSERT, UPDATE, DELETE**
 - DQL (ang. Data Query Language)
 - **SELECT**
 - DCL (ang. Data Control Language)
 - **GRANT, DENY, REVOKE**
 - TCL (ang. Transaction Control Language)
 - **COMMIT, ROLLBACK, SAVEPOINT**

Język SQL (2)

- Każde polecenie kończymy średnikiem
- Stałe łańcuchowe zapisujemy w apostrofach
- Język SQL jest niewrażliwy na wielkość liter

Polecenie SELECT

```
SELECT * FROM nazwa_tabeli
```

```
SELECT kolumna1, kolumna2, ..., kolumnaN  
FROM nazwa_tabeli
```

Polecenie SELECT (2)

SELECT wyrażenie1, wyrażenie2, ... wyrażenieN
FROM nazwa_tabeli

Gdzie wyrażenie to:

- W przypadku kolumn **typu liczbowego**:
 - Nazwa kolumny
 - Stała liczbowa
 - Suma, różnica, iloczyn lub iloraz
- W przypadku kolumn **typu łańcuchowego (tekst)**:
 - Nazwa Kolumny
 - Stała łańcuchowa
 - Konkatenacja (łączenie) dwóch łańcuchów

Polecenie SELECT (3)

Operatory arytmetyczne:

- + dodawania
- odejmowania
- * mnożenia
- / dzielenia
- () nawiasy (kolejność wykonywania działań)

Łączenie łańcuchów znaków:

CONCAT(wyrażenie1, wyrażenie2, ..., wyrażenieN)

Gdzie wyrażenie:

- Stała łańcuchowa
- Nazwa kolumny typu łańcuchowego (tekst)

Polecenie SELECT (4)

```
SELECT wyrażenie1 AS alias1,  
        wyrażenie2 AS alias2,  
        ...  
        wyrażenieN AS aliasN  
FROM nazwa_tabeli
```

```
SELECT DISTINCT kolumna1, kolumna2, ..., kolumnaN  
FROM nazwa_tabeli
```

```
SELECT kolumna1, kolumna2, ..., kolumnaN  
FROM nazwa_tabeli  
ORDER BY kolumna2 ASC
```

```
SELECT kolumna1, kolumna2, ..., kolumnaN  
FROM nazwa_tabeli  
ORDER BY kolumna2 DESC (LIMIT)
```

Polecenie SELECT (5)

SELECT kolumna1, kolumna2, ... **FROM** nazwa_tabeli
WHERE warunek

Operatory porównania:

=	- równe
!= lub <>	- różne
>	- większe
>=	- większe lub równe
<	- mniejsze
<=	- mniejsze lub równe

Operatory logiczne:

AND (logiczna koniunkcja - „i”)

OR (logiczna alternatywa - „lub”)

NOT (logiczne zaprzeczenie, negacja - „nie”)

Polecenie SELECT (6)

SELECT kolumna1, kolumna2, ...,kolumnaN **FROM** tabela
WHERE kolumna? **BETWEEN** wartość **AND** wartość

SELECT kolumna1, kolumna2, ...,kolumnaN **FROM** tabela
WHERE kolumna? **NOT BETWEEN** wartość **AND** wartość

SELECT kolumna1, kolumna2, ...,kolumnaN **FROM** tabela
WHERE kolumna? **IN** (wartość1, wartość2, ..., wartośćN)

SELECT kolumna1, kolumna2, ...,kolumnaN **FROM** tabela
WHERE kolumna? **NOT IN** (wartość1, wartość2, ..., wartośćN)

SELECT (7)

SELECT kolumna1, kolumna2, ...,kolumnaN **FROM** tabela
WHERE kolumna? **LIKE** 'wzorzec'

Wzorzec może zawierać znaki specjalne:

- %
- _

SELECT kolumna1, kolumna2, ...,kolumnaN **FROM** tabela
WHERE kolumna? **NOT LIKE** 'wzorzec'

IS NULL

IS NOT NULL

Polecenie UPDATE

```
UPDATE nazwa_tabeli  
  SET nazwa_kolumny1 = nowa_wartość1,  
       nazwa_kolumny2 = nowa_wartość2,  
       ...  
       nazwa_kolumnyN = nowa_wartośćN  
WHERE warunek
```

Polecenie INSERT

```
INSERT INTO nazwa_tabeli  
  VALUES (kolumna1, kolumna2, ..., kolumnaN)
```

```
INSERT INTO nazwa_tabeli  
  (nazwa_kolumny1, nazwa_kolumny2, ..., nazwa_kolumnyN)  
  VALUES (kolumna1, kolumna2, ..., kolumnaN)
```

```
INSERT INTO nazwa_tabeli  
  (nazwa_kolumny2, nazwa_kolumny1, ..., nazwa_kolumnyN)  
  VALUES (kolumna2, kolumna1, ..., kolumnaN)
```

```
INSERT INTO nazwa_tabeli  
  VALUES (kolumna1, kolumna2, ..., kolumnaN),  
          (kolumna1, kolumna2, ..., kolumnaN),  
  ...
```

Polecenie INSERT (2)

```
INSERT INTO nazwa_tabeli  
  SET kolumna1 = wartość1,  
      kolumna2 = wartość2,  
      ...  
      kolumnaN = wartośćN
```

Tworzenie i usuwanie bazy danych

CREATE DATABASE nazwa_bazy_danych

DROP DATABASE nazwa_bazy_danych

Typy danych SQL – liczby całkowite

INT	Liczba całkowita (4 bajty)
TINYINT	Liczba całkowita (1 bajty)
SMALLINT	Liczba całkowita (2 bajty)
MEDIUMINT	Liczba całkowita (3 bajty)
BIGINT	Liczba całkowita (8 bajty)

Typy danych SQL – liczby stałoprzecinkowe

DECIMAL (precyzja, skala)

lub

NUMERIC (precyzja, skala)

Przykład: **DECIMAL(4,2)**

przechowuje liczby z zakresu od **-9999.99** do **9999.99**

Typy danych SQL – liczby zmiennoprzecinkowe

FLOAT	Mała liczba rzeczywista (4 bajty)
DOUBLE	Duża liczba rzeczywista (8 bajtów)

Typy danych SQL – łańcuchy znaków (napisy)

CHAR (liczba_znaków)

Przechowuje napisy stałej długości

VARCHAR (maksymalna_liczba_znaków)

Przechowuje napisy o zmiennej długości

TEXT

Przechowuje bardzo długie napisy

Napis	Typ	Pamięć
„AB”	CHAR(4)	„AB__”
„AB”	VARCHAR(4)	„AB”

Typy danych SQL – czas i data

DATE	'2019-01-20'
TIME	'12:45:59'
DATETIME	'2019-01-20 12:45:59'
TIMESTAMP	'2038-01-19 03:14:07'
YEAR	2019

Typy danych SQL – BLOB, BOOLEAN

BLOB (duże obiekty binarne)

Przechowuje różne dane binarne, np.: pliki tekstowe, graficzne, dźwiękowe, itp.

BOOLEAN

Przechowuje wartość logiczną **TRUE** lub **FALSE** (0)

Tworzenie i usuwanie tabel

```
CREATE TABLE nazwa_tabeli (  
    nazwa_kolumny1    typ_danych,  
    nazwa_kolumny2    typ_danych,  
    ...  
    nazwa_kolumnyN    typ_danych  
);
```

```
DROP TABLE nazwa_tabeli;
```

Więzy integralności - klucz główny

```
CREATE TABLE nazwa_tabeli (  
    nazwa_kolumny1    typ_danych PRIMARY KEY,  
    nazwa_kolumny2    typ_danych  
);
```

```
CREATE TABLE nazwa_tabeli (  
    nazwa_kolumny1    typ_danych,  
    nazwa_kolumny2    typ_danych,  
    PRIMARY KEY (nazwa_kolumny1)  
);
```

Więzy integralności - klucz obcy

```
CREATE TABLE nazwa_tabeli (  
    nazwa_kolumny1    typ_danych,  
    nazwa_kolumny2    typ_danych,  
    nazwa_kolumny3    typ_danych,  
  
    FOREIGN KEY (nazwa_kolumny2)  
    REFERENCES tabela(nazwa_kolumny2)  
);
```

Więzy integralności – not null, unique, default, check

```
CREATE TABLE nazwa_tabeli (  
  nazwa_kolumny1      typ_danych NOT NULL,  
  nazwa_kolumny2      typ_danych UNIQUE KEY,  
  nazwa_kolumny3      typ_danych DEFAULT wartość,  
  nazwa_kolumny4      typ_danych ,  
  nazwa_kolumny5      typ_danych CHECK (warunek),  
  UNIQUE KEY (nazwa_kolumny4)  
);
```

Autoinkrementacja

```
CREATE TABLE nazwa_tabeli (  
    nazwa_kolumny1    typ_danych AUTO_INCREMENT,  
    nazwa_kolumny2    typ_danych,  
    PRIMARY KEY (nazwa_kolumny1)  
);
```

Zmiana nazwy tabeli

```
ALTER TABLE nazwa_tabeli RENAME TO  
nowa_nazwa_tabeli;
```


Dodawanie nowej kolumny

```
ALTER TABLE nazwa_tabeli ADD COLUMN  
nowa_kolumna typ;
```

```
ALTER TABLE nazwa_tabeli ADD COLUMN  
nowa_kolumna typ  
FIRST;
```

```
ALTER TABLE nazwa_tabeli ADD COLUMN  
nowa_kolumna typ  
AFTER nazwa_kolumny;
```

Usuwanie kolumny

```
ALTER TABLE nazwa_tabeli DROP COLUMN  
nowa_kolumna;
```

Zmiany definicji istniejącej kolumny

```
ALTER TABLE nazwa_tabeli ADD PRIMARY KEY  
    (nowa_kolumna);
```

```
ALTER TABLE nazwa_tabeli DROP PRIMARY KEY;
```

```
ALTER TABLE nazwa_tabeli ADD FOREIGN KEY  
    (nowa_kolumna)  
    REFERENCES tabela(nazwa_kolumny);
```

```
ALTER TABLE nazwa_tabeli DROP FOREIGN KEY id_key
```

Zmiany definicji istniejącej kolumny

```
ALTER TABLE nazwa_tabeli MODIFY COLUMN  
nazwa_kolumna typ;
```

```
ALTER TABLE nazwa_tabeli MODIFY COLUMN  
nazwa_kolumna typ FIRST
```

```
ALTER TABLE nazwa_tabeli MODIFY COLUMN  
nazwa_kolumna typ  
AFTER nazwa_kolumny
```

```
ALTER TABLE nazwa_tabeli CHANGE COLUMN  
nazwa_kolumna nowa_nazwa_kolumny typ;
```