



BAZY DANYCH



mgr inż. Krzysztof Janiak

e-mail: krzysztofjaniak@zse-zdwola.pl

SQL



Bazy danych - podstawowe pojęcia

Wyjaśnienie pojęcia: baza danych



Czym jest baza danych?

Baza danych - jest to zbiór danych powiązanych w formie tabel. Przykładem bazy danych może być, np.: szkoła, biblioteka, księgarnia itp.

W celu umożliwienia użytkownikom zarządzania bazą danych opracowano oprogramowanie do tego celu przeznaczone **Database Management Systems (DBMS)** - w języku polskim: **System zarządzania bazą danych (SZBD)**.

W/w oprogramowanie umożliwia tworzenie i administrowanie Naszą bazą danych poprzez dodawanie, usuwanie, edytowanie, sprawdzanie poprawności informacji.

Do zarządzania oprogramowaniem do baz danych zastosowano strukturalny język zapytań **SQL** (*Structured Query Language*), z prostym interfejsem użytkownika.

Relacyjny model baz danych



Relacyjny model baz danych

Relacyjna baza danych - to zbiór danych w postaci tabel połączonych relacjami.

Opis relacyjnego modelu baz danych można podzielić na trzy części:

- struktury danych - czyli, w jaki sposób i według jakich zasad organizujemy przechowywanie danych oraz według jakich zasad należy je projektować,
- języki manipulowania danymi - czyli, w jaki sposób zapisywać, modyfikować, usuwać oraz pobierać dane znajdujące się w bazie danych,
- integralność danych - czyli, w jaki sposób zapewnić poprawność przechowywanych danych.

Relacyjny model baz danych

Zasady dla poprawnie działającego relacyjnego modelu baz danych

- każda tabela w bazie danych ma jednoznaczną nazwę,
- każda kolumna tabeli ma jednoznaczną nazwę w obrębie tej tabeli,
- wszystkie wartości w kolumnie są tego samego typu,

Nazwisko	Imie	Pesel	DataUrodzenia	IloscRodzenstwa
Kot	Jan	92091187345	1992-09-11	2
Lisek	125-12	2009-12-12	12 grudzień 1993	brat i siostra
Wojtek	92071287765	Sierpień 1992	Marczak	dwóch braci



Nazwisko	Imie	Pesel	DataUrodzenia	IloscRodzenstwa
Kot	Jan	92091187345	1992-09-11	2
Lisek	Barbara	93121287634	1993-12-12	2
Marczak	Wojciech	92061223871	1992-06-12	2



Relacyjny model baz danych

Zasady dla poprawnie działającego relacyjnego modelu baz danych

- w tabeli nie mogą istnieć dwa identyczne wiersze, każdy wiersz jest różny, tabela może istnieć bez wierszy,

Nazwisko	Imie	DataUrodzenia	IloscRodzenstwa
Kot	Jan	1992-09-11	2
Kot	Jan	1992-09-11	2
Marczak	Wojciech	1992-06-12	2



Pesel	Nazwisko	Imie	DataUrodzenia	IloscRodzenstwa
92091187345	Kot	Jan	1992-09-11	2
93121287634	Kot	Jan	1993-12-12	2
92061223871	Marczak	Wojciech	1992-06-12	2



Relacyjny model baz danych

Zasady dla poprawnie działającego relacyjnego modelu baz danych

- kolejność wierszy i kolejność kolumn w tabeli relacyjnej nie ma żadnego znaczenia, czyli położenie danej w tabeli nie wpływa na jej znaczenie.

Nazwisko	Imie	Pesel	DataUrodzenia
Kotek	Katarzyna	92031275446	1992-03-12
Piesek	Jan	92051587746	1992-05-15
Lisek	Kasia	92022277654	1992-02-22

Imie	Pesel	DataUrodzenia	Nazwisko
Katarzyna	92031275446	1992-03-12	Kotek
Jan	92051587746	1992-05-15	Piesek
Kasia	92022277654	1992-02-22	Lisek

DataUrodzenia	Pesel	Nazwisko	Imie
1992-03-12	92031275446	Kotek	Katarzyna
1992-05-15	92051587746	Piesek	Jan
1992-02-22	92022277654	Lisek	Kasia



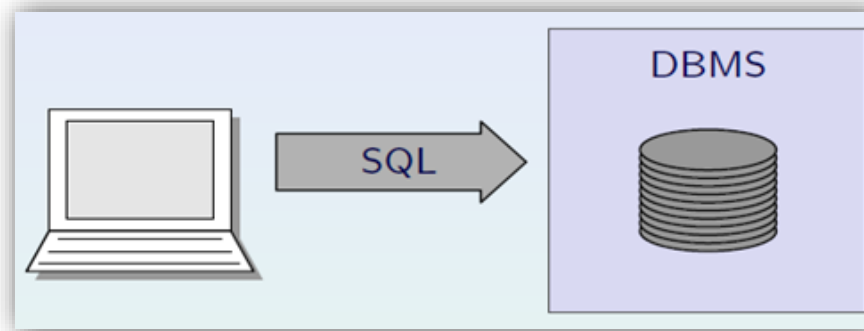
**STRUKTURALNY
JĘZYK ZAPYTAŃ
SQL**

(z ang. Structured Query Language)

Structured Query Language



Czym jest język SQL?



SQL — *Structured Query Language*

- Interfejs pomiędzy programem klienckim a bazą danych.
- Język dostępu do danych.
- Oparty na rachunku relacyjnym.
- Standardowy język stosowany we współczesnych systemach zarządzania bazami danych.
- Język *nieproceduralny* (inaczej *deklaratywny*).

Structured Query Language



Programowanie proceduralne, a deklaratywne

C/C++, Java, Fortran itp.

```
int main(int argc, char **argv){
    if (argc < 3) {
        printf("Za mało arg.!");
        return 1;
    }
    for (int i=0; i<argc; i++)
        printf("Argument %d=%s", i, argv[i]);
    return 0;
}
```

- Elementem składni są instrukcje sterujące wykonaniem (pętle, warunki).
- Kolejne kroki programu definiują sposób uzyskania określonego wyniku.

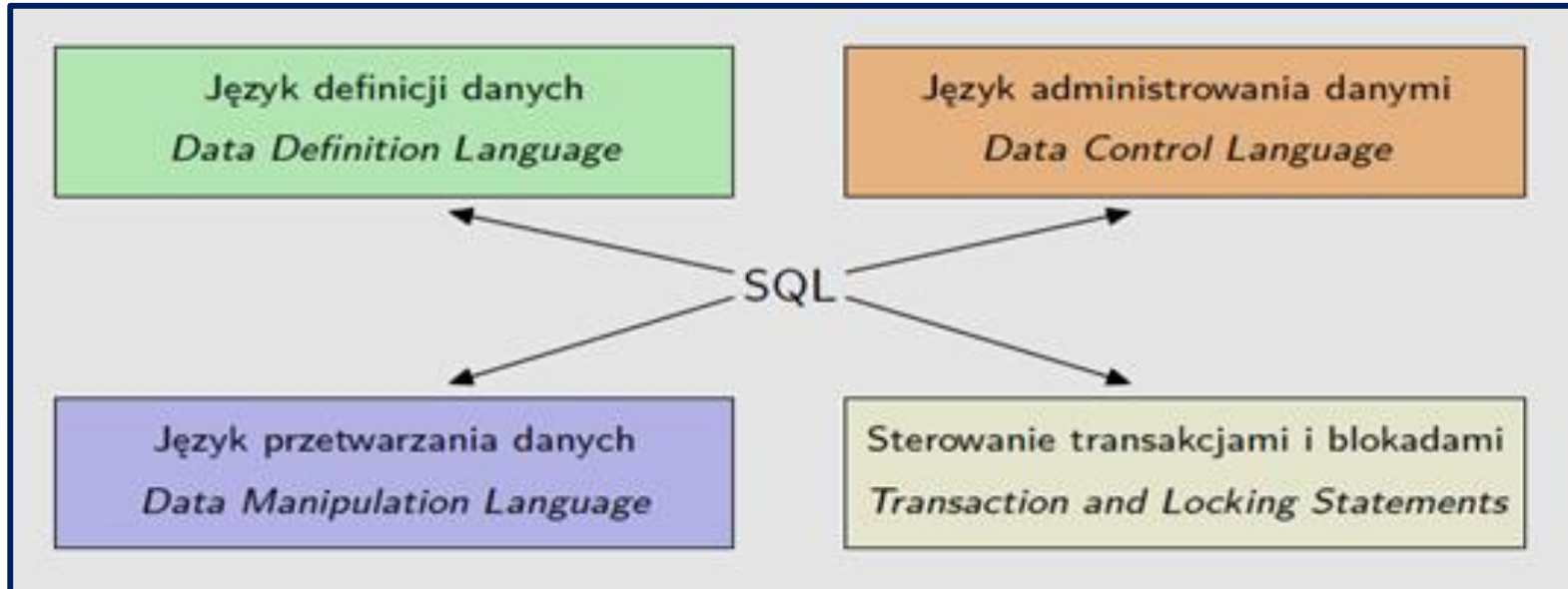
- Składnia zbliżona do języka naturalnego.
- Instrukcje deklarują *co* uzyskać z bazy danych a nie *jak* to zrobić.

SQL

```
CREATE VIEW PracownicyDr(SELECT *
    FROM Pracownicy WHERE Stopien="dr");
UPDATE PracownicyDr SET
    Pensja=Pensja+100 WHERE
    Pensja < 1000;
```

Structured Query Language

Polecenia SQL



- DDL — tworzenie, usuwanie i zmiana schematów (baz danych), tabel itp.
- DML — wprowadzanie, usuwanie, modyfikacja i wydobywanie danych.
- DCL — tworzenie i zarządzanie kontami użytkowników, zapisywanie danych.
- TLS — zatwierdzenie i anulowanie transakcji, blokowanie tabel.

Structured Query Language



Instrukcja tworzenia nowych struktur

```
CREATE structure
```

```
structure:
```

```
    DATABASE
```

```
    |  
    TABLE
```

```
    |  
    INDEX
```

```
    |  
    VIEW
```

```
    |  
    TRIGGER
```

```
    |  
    PROCEDURE
```

```
    . . .
```

Instrukcja tworzenia nowej tabeli - przykład

Przykład

```
mysql> CREATE TABLE example_tbl(  
  -> id INT NOT NULL AUTO_INCREMENT,  
  -> title VARCHAR(100) NOT NULL,  
  -> author VARCHAR(40) NOT NULL,  
  -> submission_date DATE,  
  -> PRIMARY KEY (id)  
  -> );
```

Instrukcja wprowadzania danych do tabeli

Składnia

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
      [INTO] tbl_name [(col_name ,...)]
      {VALUES | VALUE} ({expr | DEFAULT} ,...), (...), ...
```

Przykłady

— wprowadzenie wiersza wartości domyślnych

```
mysql> INSERT INTO tbl_name () VALUES ();
```

— wyrażenie w drugiej kolumnie

```
mysql> INSERT INTO tbl_name (col1 , col2) VALUES (15 , col1 * 2);
```

— kilka rekordów

```
mysql> INSERT INTO tbl_name (a , b , c)
      -> VALUES (1 , 2 , 3) , (4 , 5 , 6) , (7 , 8 , 9);
```


Structured Query Language



Aktualizacja danych w tabeli

Składnia

```
UPDATE [IGNORE] tbl_name SET
  col_name1={expr1 | DEFAULT } [,
  col_name2={expr2 | DEFAULT } ] ...
[WHERE where_condition]
[ORDER BY ...]
[LIMIT row_count]
```

- Instrukcja UPDATE zmienia wartości istniejących rzędów w tabeli.
- SET wskazuje kolumny, które ulegną modyfikacji.
- WHERE identyfikuje rzędy, które ulegną modyfikacji (brak powoduje zmianę wszystkich rzędów tabeli).
- ORDER BY powoduje, że aktualizowanie rzędów odbywać się będzie według określonego porządku.
- LIMIT ogranicza liczbę modyfikowanych rzędów.

Przykłady

```
mysql> UPDATE persondata SET age=age+1;
mysql> UPDATE t SET id = id + 1 ORDER BY id DESC;
mysql> UPDATE items,month SET items.price=month.price
-> WHERE items.id=month.id;
```

Structured Query Language

Instrukcja SELECT

Składnia

```
SELECT [ALL | DISTINCT | DISTINCTROW ]  
      select_expr , ...  
      [FROM table_references  
        [WHERE where_condition]  
        [GROUP BY {col_name | expr | position }  
          [ASC | DESC], ... [WITH ROLLUP]]  
        [HAVING where_condition]  
        [ORDER BY {col_name | expr | position }  
          [ASC | DESC], ...]
```

- *select_expr* — co ma być wybrane
- FROM — z której tabeli mają być wybrane rekordy
- WHERE — kryterium selekcji
- GROUP BY — grupowanie
- HAVING — ograniczenie wybranych rekordów w grupowaniu

Structured Query Language



Instrukcja SELECT, a operacje relacyjne

Operacje relacyjne

- Selekcja — wybór rekordów na podstawie kryterium
- Projekcja — ukrycie niektórych pól rekordów
- Złączenie — połączenie rekordów dwóch tabel na podstawie równości ustalonych pól

SELECT	<i>które pola wybrać</i>	Projekcja
FROM	<i>z których tabel wybrać dane</i>	Argumenty operacji
WHERE	<i>jakie kryterium selekcji</i>	Selekcja lub złączenie

Przykłady zapytań SQL

```
mysql> SELECT * FROM Pracownicy;  
  
mysql> SELECT Nazwisko, Imię FROM Pracownicy;  
  
mysql> SELECT Nazwisko, Imię FROM Pracownicy  
-> WHERE pensja > 3000;  
  
mysql> SELECT Nazwisko, Imię, Pensja  
-> FROM Pracownicy WHERE Pensja > 3000  
-> ORDER BY Nazwisko, Imię;
```

Synonim

Synonim (*alias*) używany jest w celu zastąpienia nazwy pola...

```
mysql> SELECT Nazwisko , Imię , Pensja AS Pobory  
  -> FROM Pracownicy  
  -> WHERE Pobory > 3000;
```

...lub tabeli

```
mysql> SELECT Nazwisko  
  -> FROM Pracownicy AS Pr  
  -> WHERE Pr.Pensja > 3000;
```

Funkcje agregujące

Wybrane funkcje agregujące

Nazwa	Opis
SUM()	Suma
COUNT()	Liczba elementów w grupie
AVG()	Średnia arytmetyczna
MAX()	Wartość maksymalna
MIN()	Wartość minimalna
FIRST()	Pierwszy z grupy
LAST()	Ostatni z grupy

Structured Query Language

Przykład grupowania

Tabela *Operacje*

Miesiac	Wplyw
Styczen	1000
Styczen	2300
Styczen	4000
Luty	1000
Luty	1300
Marzec	1450
Marzec	2300
Marzec	100

Wynik selekcji z grupowaniem

Miesiac	SUM(Wplyw)
Styczen	7300
Luty	2300
Marzec	3850

Zapytanie

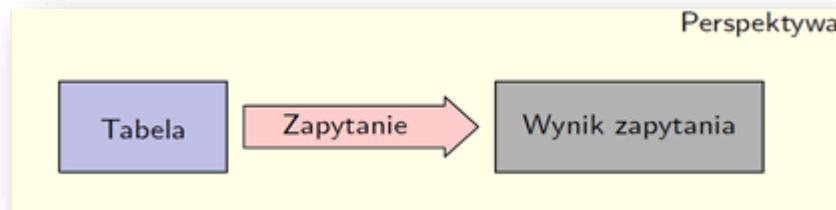
```
mysql> SELECT Miesiac , SUM(Wplyw) FROM Operacje  
→ GROUP BY Miesiac ;
```

Structured Query Language

Widok (perspektywa)

Wirtualna tabela, którą określa zapytanie SQL.

Można z niej pobierać dane **SELECT** tak samo, jak ze zwykłej tabeli. Dzięki niej możemy zaoszczędzić sobie tworzenia w aplikacji długich zapytań, a także w niektórych przypadkach uniezależnić aplikacje od właściwej struktury podstawowych tabel.



- zapamiętanie definicji zapytania
- ukrycie szczegółów zapytania
- ukrycie dostępu do części pól tabeli

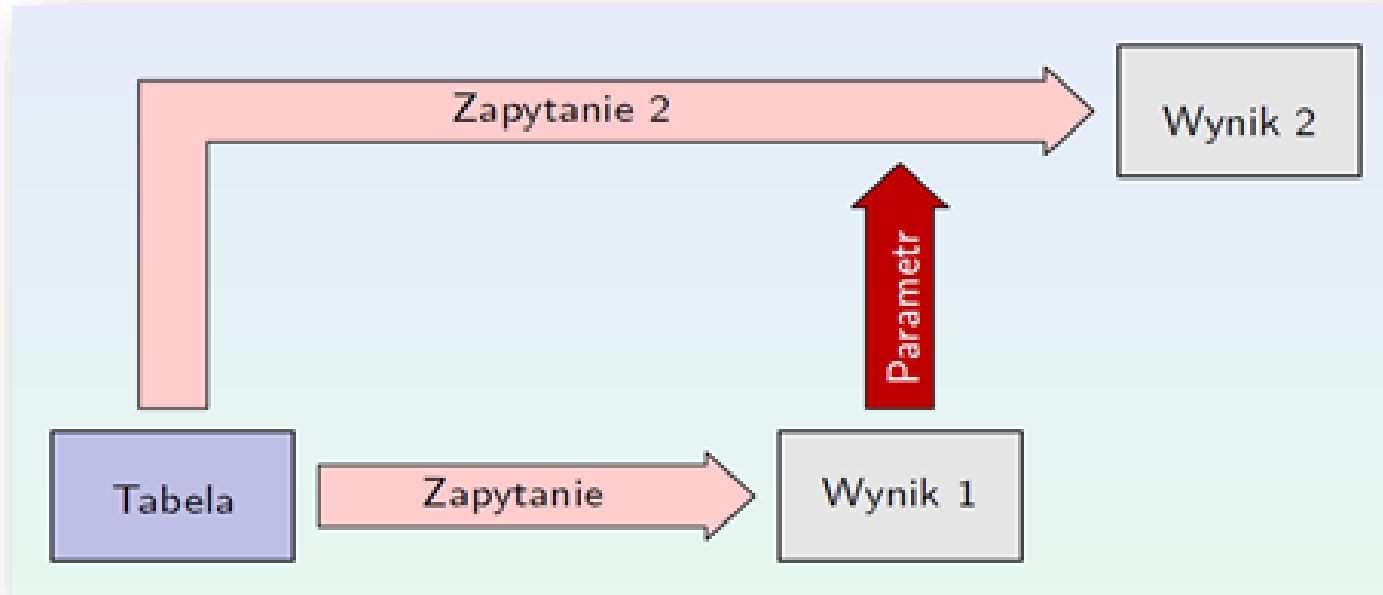
```
CREATE VIEW WplywyMiesieczne AS SELECT  
Miesiac, SUM(Wplyw) AS SumaW FROM Operacje GROUP BY Miesiac;
```

- perspektywa traktowana jak nazwa tabeli
- wykonanie zapytania w momencie odwołania do perspektywy
- możliwość realizacji łańcucha zapytań

```
SELECT * FROM WplywyMiesieczne WHERE Miesiac < 12;
```

Zagadnienia z zakresu systemów baz danych

Zapytanie zagnieżdżone



```
SELECT Miesiac, Wplyw FROM Operacje  
WHERE Wplyw > (SELECT AVG(Wplyw) FROM Operacje);
```


Structured Query Language



Podsumowanie

- Język SQL jest językiem czwartej generacji — wyraża to, czego się oczekuje, a nie jak to wykonać (deklaratywność).
- SQL pozwala na wykonanie dowolnych operacji zdefiniowanych w relacyjnej teorii baz danych.
- W języku SQL występują trudności w wyrażaniu rekurencji oraz złożonych algorytmów.
- Implementacja języka SQL w każdym systemie DBMS jest częściowo zgodna ze standardem SQL, a częściowo go rozszerza.
- Dokumentacja systemu MySQL dostępna jest pod adresem: <http://dev.mysql.com/doc/>.



Dziękuję za uwagę



mgr inż. Krzysztof Janiak

e-mail: krzysztofjaniak@zse-zdwola.pl

SQL

