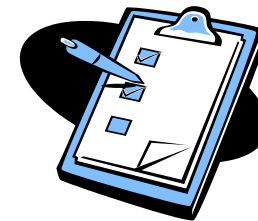


BAZY DANYCH

WYKŁAD 3

Uniwersytet Gdański

Agenda



2

Wprowadzenie do obiektowości

Podstawowe instrukcje języka SQL

Instrukcje tworzące, modyfikujące,
usuwające i wybierające

Język T-SQL i jego rozszerzenia

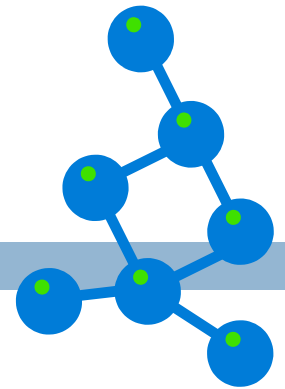
3 Wprowadzenie do obiektowości

Struktura bazy danych



Model danych

4



□ Definiuje:

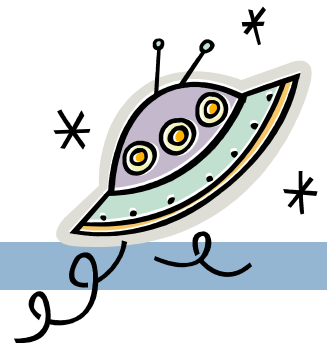
- logiczne reprezentacje struktur danych,
- powiązania pomiędzy elementami struktur.

□ Rodzaje:

- przedrelacyjne – hierarchiczne i sieciowe,
- relacyjne – stosowane obecnie,
- postrelacyjne – obiektowe lub relacyjno-obiektowe.

Obiektość w bazie danych

5



□ Model obiektowy:

- rozpatruje określone zagadnienia jako zespoły obiektów, których budowę i sposobami porozumiewania się rządzą ściśle określone reguły,
- posiada klasy, które są wyznacznikiem dla grupy obiektów (jest część wspólna połączenia dwóch lub więcej tabel).

Obiektość w bazie – praktyczne przykłady



6

- Możliwość obsługi typu obiektowego – np. zdefiniowanie jako obiektu danych: imię, nazwisko, data urodzenia.
- Stosowanie tabel **zagnieżdżonych**.
- Pojęcia:
 - ▣ hermetyzacja,
 - ▣ rozszerzalność,
 - ▣ dziedziczenie,
 - ▣ polimorfizm.



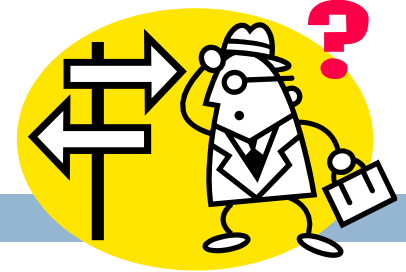
Język SQL – historia

8

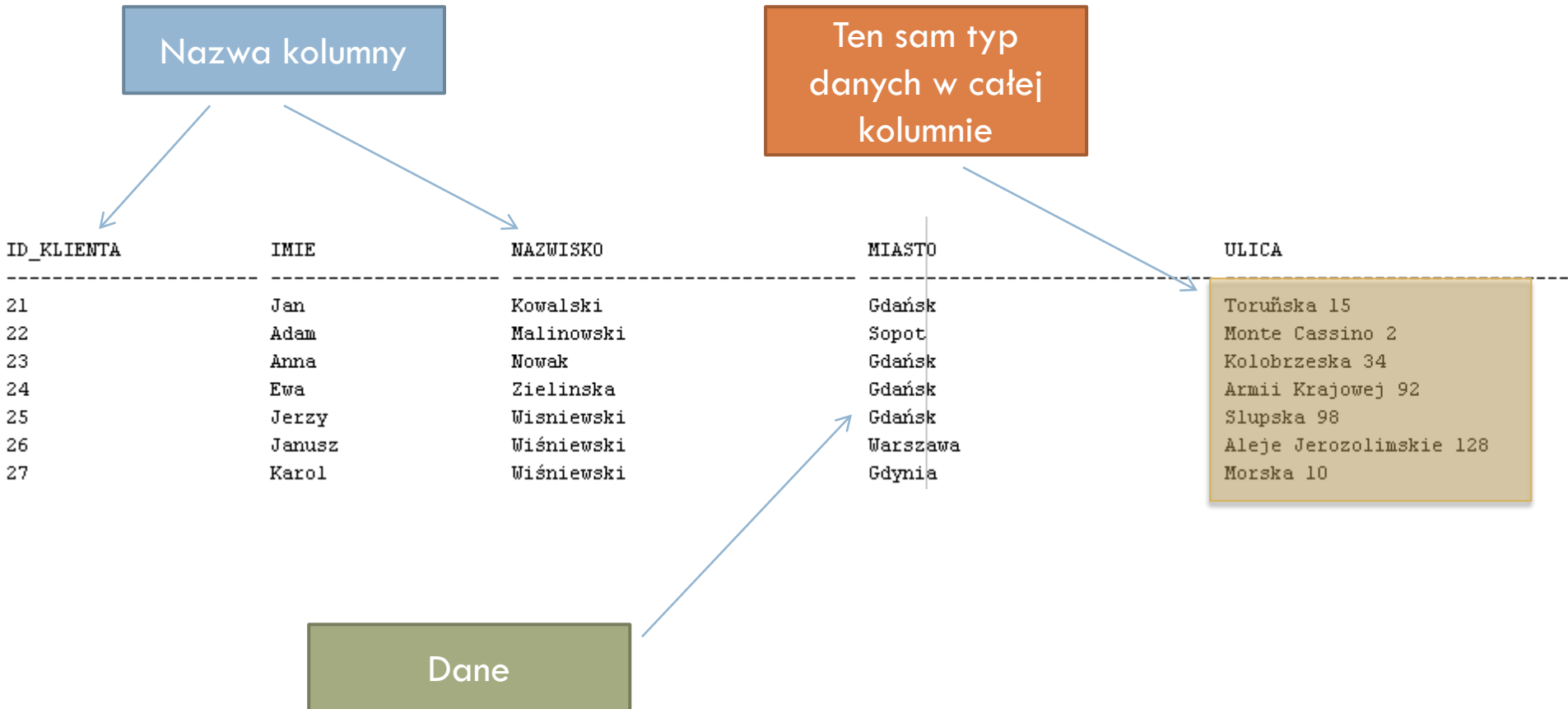


- Structure Query Language
- Język deklaratywny
- W latach 70tych opracowany w firmie IBM
- W roku 1986 stał się standardem (ISO, ANSI)

Tabela w bazie danych



9



Przykład struktury tabel

10



Nazwa kolumny

Ten sam typ
danych w całej
kolumnie

Name	Null	Type
-----	-----	-----
ID_KLIENTA	NOT NULL	NUMBER(5)
IMIE	NOT NULL	VARCHAR2(20)
NAZWISKO	NOT NULL	VARCHAR2(30)
MIASTO	NOT NULL	VARCHAR2(30)
ULICA	NOT NULL	VARCHAR2(30)

Czy dozwolone
wartości NULL?

Język SQL – komentarze

11



- -- komentarz jednolinijkowy
- /* komentarz
wielolinijkowy */

Ograniczenia i klucze



12

- Klucz główny
 - ▣ nazwa_kolumny typ_danych PRIMARY KEY
- Klucz obcy
 - ▣ FOREIGN KEY (nazwa_kolumny_klucza_obcego)
REFERENCES nazwa_tabeli_referencyjnej
(nazwa_kolumny_referencyjnej)

Polecenie CREATE

13



- **CREATE TABLE** nazwa_tabeli
- (nazwa_kolumny typ_danych ograniczenia,
- ...)

- Podstawowe typy danych to:
 - ▣ NUMERIC
 - ▣ VARCHAR

Polecenie CREATE



14

- create table klient (
 - id_klienta numeric(5) PRIMARY KEY,
 - imie varchar (20) not null,
 - nazwisko varchar(30) not null,
 - Miejscowosc varchar(30) not null,
 - ulica varchar(30) null
-);

Usuwanie tabel

15



- DROP TABLE nazwa_tabeli;

- Przykład:
 - ▣ DROP TABLE Student;

Składnia polecenia wstawiającego i usuwającego wiersze



16

- ❑ INSERT INTO nazwa_tabeli [(lista kolumn)]
VALUES (lista wartości);
- ❑ DELETE FROM nazwa_tabeli [WHERE warunek];
- ❑ UWAGA! Polecenie „DELETE FROM nazwa_tabeli;”
usunie wszystkie wiersze
z tabeli.



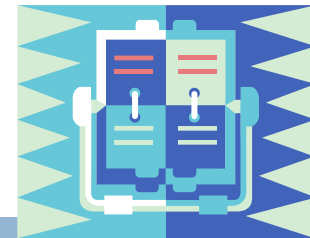
Przykłady wstawiania danych

17

- `insert into klient values(1,'Jan','Kowalski',
'Gdańsk','Toruńska 15');`

- `insert into transakcje (idt, idk, idn, cena, data)
values (1, 4, 4, 220000, '2016-10-15');`

Stosowanie synonimów



18

- `SELECT nazwaKolumny AS nazwaSynonimu FROM nazwaTabeli;`
- `lub`
- `SELECT nazwaKolumny nazwaSynonimu FROM nazwaTabeli;`
- Przykłady:
 - ▣ `select nazwisko as NazwiskoKlienta from klient;`
 - ▣ `select nazwisko + imie nazwa from klient;`

Iloczyn kartezjański



19

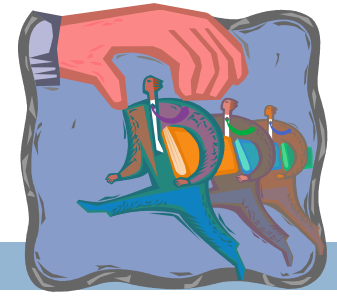
- `SELECT s1.nazwaKolumny, s2.nazwaKolumny FROM nazwaTabeli1 s1, nazwaTabeli2 s2 where s1.idTabeli1 = s2.idTabeli1;`
- Gdzie `s1.idTabeli1` to klucz główny w tabeli 1, a `s2.idTabeli1` to klucz obcy w tabeli 2

Wyrażenie ORDER BY



20

- Wyrażenie order by używane jest używane jest w celu sortowania zbioru danych. Sortowanie może być w porządku rosnącym lub malejącym.
 - ▣ `SELECT nazwaKolumny FROM nazwaTabeli ORDER BY { nazwaKolumny | wyrażenie | pozycja } [ASC | DESC];`
- gdzie
 - ▣ ORDER BY – definiuje kolumnę według której nastąpi sortowanie.
 - ▣ ASC – oznacza rosnące sortowanie.
 - ▣ DESC – oznacza sortowanie malejące.
- Przykłady:
 - ▣ `select imie, nazwisko from uzytkownicy order by imie, nazwisko;`
 - ▣ `select imie, nazwisko from uzytkownicy order by nazwisko desc;`
 - ▣ `select imie, nazwisko, miasto from uzytkownicy order by 2, 3 asc;`



Wybieranie danych

21

1. `select * from dom;`
2. `select imie, nazwisko, miasto from klient2 order by miasto asc;`
3. `select ulica, miasto, metraz from dom order by metraz desc;`
4. `select * from dom where metraz < 200;`
5. `select k.imie, k.nazwisko, d.miasto, d.ulica,
d.cena_proponowana, t.cena_uzyskana,
t.data_transakcji
from klient2 k, dom d, transakcje t
where k.id_klienta = t.id_klienta and
d.id_domu = t.id_domu;`

Wybieranie danych



22

1. `select * from uzytkownicy where nazwisko like 'K%';`
2. `select * from uzytkownicy where imie like 'J__';`
3. `select * from uzytkownicy where imie = 'Jan';`
4. `select imie, nazwisko from uzytkownicy order by imie, nazwisko;`
5. `select imie, nazwisko from uzytkownicy order by nazwisko desc;`
6. `select imie, nazwisko, miasto from uzytkownicy order by 2, 3 asc;`

Przykłady instrukcji



Język T-SQL

24



- Transact – Structured Query Language
- Rozwijany w bazie danych MS SQL Server
- Niewielkie różnice w podstawowej składni
- Pozwala na automatyzację pracy z bazami danych
- Rozszerza możliwości języka SQL

Typy składni języka T-SQL



- DDL (Data Definition Language) – np. instrukcje CREATE
- DCL (Data Control Language) – np. instrukcje GRANT, DENY, REVOKE
- DML (Data Manipulation Language) – np. INSERT

Typowa składnia języka



- **USE NazwaBazy;**
 - ▣ przełącza na inną bazę danych
- **GO;**
 - ▣ wykonuje poprzednie polecenie
- **SELECT * FROM Tabela;**
 - ▣ typowe zapytanie SQL

Identyfikatory zmiennych, tabel i obiektów



- Pierwszy znak musi być literą z zakresu od A do Z lub a do z, bez polskich znaków
- Pozostałe znaki mogą zawierać cyfry oraz znaki takie jak #, @ lub _
- Nie są dozwolone spacje oraz specjalne znaki takie jak ?, * czy podobne (choć dopuszczalne w przypadku nazw tabel)
- Nie mogą być słowami kluczowymi języka T-SQL np. if, case itp.

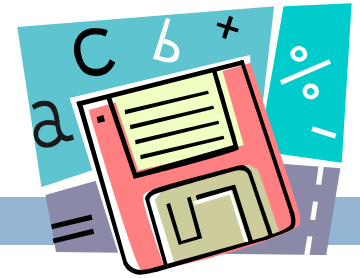
Odwołania do nazw



- SerwerBazodanowy.BazaDanych.NazwaSchematu.NazwaObiektu
 - ▣ UG.Uczelnia.Student.Tabela

- Możliwe są również odwołania częściowe np.
 - ▣ Tabela
 - ▣ Student.Tabela

Dozwolone operatory



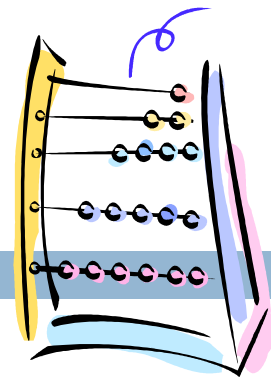
- Arytmetyczne
- Porównania
- Logiczne
- Konkatenacji
- Złożone

Operatory arytmetyczne



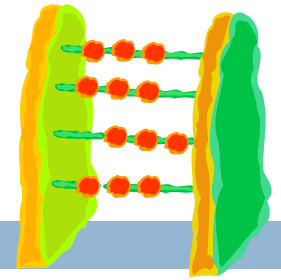
- + dodawanie
- - odejmowanie
- * mnożenie
- / dzielenie
- % reszta z dzielenia (modulo)

Operatory porównania



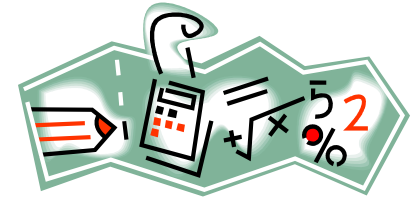
- = równy
- > większy niż
- < mniejszy niż
- >= większy bądź równy
- <= mniejszy bądź równy
- <> różny

Operatory logiczne



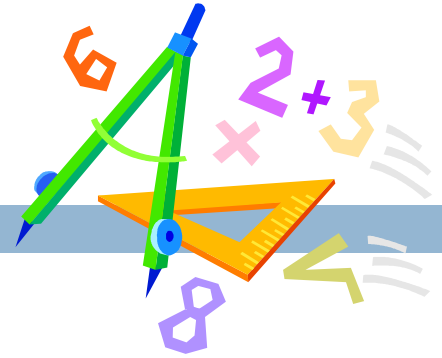
- ALL – zwraca True jeżeli wszystkie wyniki zapytania się zgadzają
- AND – zwraca True jeżeli oba operandy są prawdziwe
- ANY – zwraca True jeżeli jakikolwiek wynik zapytania jest prawdziwy
- BETWEEN – określa dolną i górną granicę w warunku
- EXISTS – jeżeli podzapytanie posiada co najmniej jeden wiersz to zwróci wartość True
- IN – zwraca True jeżeli operand zgadza się z jakąkolwiek wartością określoną przez IN
- LIKE – zwraca True jeżeli operand jest zgodny z wzorcem
- NOT – negacja wartości logicznej
- OR – zwraca True jeżeli jakikolwiek z operandów jest True
- SOME – zwraca True jeżeli jakikolwiek wynik zapytania jest prawdziwy

Operatory konkatencji



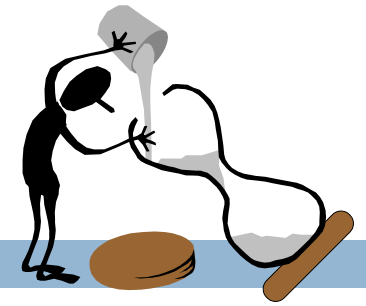
- + - operator przeciężony
- Należy uważać podczas dodawania dwóch kolumn typu Varchar
 - ▣ np. wartosc1 Varchar(10), wartosc2 Numeric(10,0)
 - wartosc1='00001'
 - wartosc2=2
 - działanie wartosc1+wartosc2 zwróci 3

Operatory złożone



- $+=$ dodawanie z przypisaniem
- $-=$ odejmowanie z przypisaniem
- $*=$ mnożenie z przypisaniem
- $/=$ dzielenie z przypisaniem
- $\%=$ reszta z dzielenia z przypisaniem

Tworzenie tabel tymczasowych



- Nazwa tabeli powinna być poprzedzona hashem:
 - ▣ #NazwaTabeli – tabela tymczasowa
 - ▣ ##NazwaTabeli – globalna tabela tymczasowa
- Tabela tymczasowa jest usuwana w momencie rozłączenia klienta, który utworzył tę tabelę.
- Tabela tymczasowa globalna jest usuwana, gdy wszyscy użytkownicy z niej korzystający się rozłączą.

Instrukcja CASE



- CASE
 - WHEN warunek THEN wartość
 - ELSE wartość_domyślna
- END AS 'Nazwa kolumny'

Przykład instrukcji CASE



- SELECT '2008' as 'Rok',
- w as 'Wojewodztwo',
- **CASE**
- **WHEN** symbol<500 **THEN** 1
- **WHEN** symbol<=1000 **THEN** 2
- **ELSE** 1
- **END** as 'Rodzaj'
- FROM Tabela
- WHERE symbol <= 1000;

Zapytania złożone



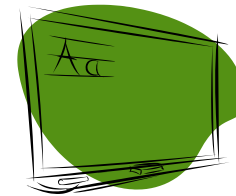
- WITH nazwa_zapytania AS (
 - ▣ **podzapytanie (wewnętrzne)**
-)
 - ▣ **zapytanie zewnętrzne**
- Wszystkie kolumny oraz aliasy kolumn dostępne w podzapytaniu będą do wykorzystania w zapytaniu zewnętrznym.

Przykład zapytania wewnętrznego i zewnętrznego



- **WITH Zapytanie AS (**
- **SELECT '2012' as 'Rok', woj as 'Wojewodztwo',**
- **CASE**
- **WHEN symbol<500 THEN 1**
- **WHEN symbol<=800 THEN 2**
- **else 1**
- **END as 'Forma',**
- **'3' as 'Rodzaj', liczba as 'Wartosc' FROM Tabela**
- **)**
- **SELECT rok, wojewodztwo, forma, rodzaj, sum(wartosc) as Wartosc**
- **FROM Zapytanie**
- **GROUP BY rok, wojewodztwo, forma, rodzaj**

Kopiowanie danych do nowej tabeli



- `SELECT kolumna1, kolumna2 INTO nowa_tabela FROM tabela`

- Przykład:
 - ▣ `SELECT a.wojewodztwo, a.nazwa, b.wartosc`
 - ▣ **`INTO DaneFirm`**
 - ▣ `FROM opis a, dane b where a.id_firmy=b.id_firmy;`

Kopiowanie danych do istniejącej tabeli



- INSERT INTO tabela SELECT kolumna FROM tabela2;

- Przykład:
 - ▣ **INSERT INTO DaneFirm**
 - ▣ SELECT a.wojewodztwo, a.nazwa, b.wartosc
 - ▣ FROM opis a, dane b where a.id_firmy=b.id_firmy;

Pytanie?

42

Które z zapytań spowodują wypisanie wszystkich wierszy zawierających w kolumnie imie wartość „Jan”:

- ❑ `select * from uzytkownicy where imie like 'J_';`
- ❑ `select * from uzytkownicy where imie = 'J%';`
- ❑ `select * from uzytkownicy where imie = 'Jan';`
- ❑ `select * from uzytkownicy where imie like 'J*';`

