

BAZY DANYCH, WYKŁAD NR 4

Uniwersytet Gdański

Agenda



2

Język definiowania struktur danych

- Typy danych

Wyrażenia SQL

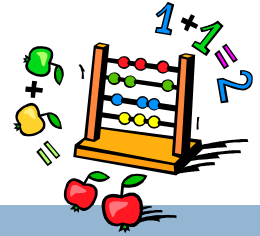
- Instrukcje DISTINCT, GROUP BY, IN, ANY oraz ALL
- Wyrażenia arytmetyczne
- Złączenia tabel

3

Typy danych

- 4

Liczbowe typy danych



5

- Całkowite
 - Tinyint (1 bajt)
 - Smallint (2 bajty)
 - Int (4 bajty)
 - Bigint (8 bajtów)
- Stałoprzecinkowe
 - Numeric(p[,s])
 - Decimal(p[,s])
 - p – precyzja (1-38)
 - s – skala
($0 \leq s \leq p$)
- Zmiennoprzecinkowe
 - Float
 - dla 7 cyfr 4 bajty
 - dla 15 cyfr 8 bajtów
 - Real (4 bajty)
- Waluta
 - Money (8 bajtów)
 - Smallmoney (4 bajty)

Daty



- Typy danych obecnie wspierane:
 - ▣ Date – data
 - ▣ Time – czas
 - ▣ Datetime2 – data i czas
 - ▣ Datetimeoffset – data i czas ze strefą czasową
- Wciąż dostępne, ale wycofywane:
 - ▣ Datetime
 - ▣ Smalldatetime

Znakowe typy danych



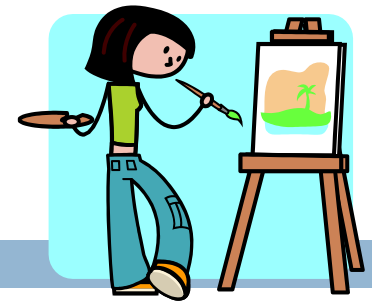
- Char – stałej długości znakowe
- Nchar – stałej długości znakowe Unicode
- Varchar – zmiennej długości znakowe
- Nvarchar – zmiennej długości znakowe Unicode

Binarne



- Przyjmują wartości 0, 1 lub NULL
 - ▣ Binary – stałej długości
 - ▣ Varbinary – zmiennej długości
 - ▣ Bit – tylko jeden znak

Obrazy i tekst



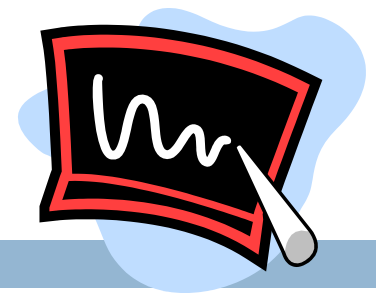
- Image – przechowuje dane binarne, np. obrazy
- Text – maksymalnie $(2^{31}) - 1$ znaków
- Ntext – jak wyżej, ale znaki zapisane w Unicode

Przestrzenne



- Używane do zapisywania danych przestrzennych:
 - Geography
 - Geometry

Uzupełniające



- Typy uzupełniające odnoszą się do specyficznych danych, np. typy tablicowe czy systemowe.
- Wykorzystywane są w automatyzacji zarządzania bazą danych.

Wyrażenia SQL

Stosowanie warunku WHERE



13

- Warunek where stosowany w instrukcji select jest identyczny, jak dla instrukcji delete oraz update.
- Przykłady:
 - ▣ `select * from uzytkownicy where nazwisko like 'K%';`
 - ▣ `select * from uzytkownicy where imie like 'J__';`
 - ▣ `select * from uzytkownicy where imie = 'Jan';`

Instrukcja DISTINCT



14

- Oznacza, że wiersze wielokrotnie występujące w kolumnie nie będą się powtarzać.

Przykłady:

- `SELECT DISTINCT kolumna FROM tabela;`
- `SELECT DISTINCT (miasto) FROM klient;`

Instrukcja DISTINCT



15

☐ select miasto from klient;

☐ MIASTO

☐ -----

☐ Gdańsk

☐ Sopot

☐ Gdańsk

☐ Gdańsk

☐ Gdańsk

☐ Warszawa

☐ Gdynia

☐ Gdynia

☐ Elbląg

☐ select distinct miasto from klient;

☐ MIASTO

☐ -----

☐ Elbląg

☐ Gdynia

☐ Warszawa

☐ Gdańsk

☐ Sopot

Instrukcja GROUP BY



16

- Celem stosowania instrukcji jest wybranie pogrupowanych rekordów według wskazanych kryteriów.
- Przykłady:
 - ▣ średnia sprzedaż produktów według województw i lat
 - ▣ łączna sprzedaż produktów według nazwy produktu
- Instrukcja grupująca GROUP BY używana jest między innymi w celu wykorzystywania instrukcji arytmetycznych.

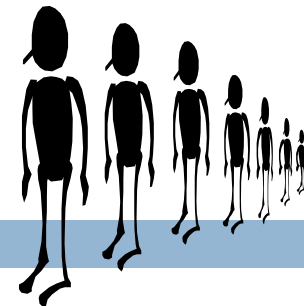
Składnia instrukcji GROUP BY



17

- **SELECT** [instrukcjaArytmetyczna] nazwaKolumny,...
FROM nazwaTabeli
WHERE warunekWhere
GROUP BY nazwaKolumny
- [**HAVING** warunekHaving]
- gdzie
 - ▣ **HAVING** jest warunkiem wyrażenia **GROUP BY**,
definiowanym analogicznie jak warunek **WHERE**.

Przykład instrukcji GROUP BY



18

-- brak GROUP BY

```
select nazwa, cena
from produkt;
```

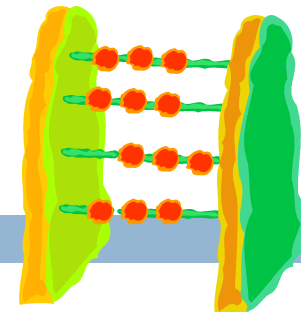
□ NAZWA	CENA
□ Buty	99,9
□ Buty	89,9
□ Buty	129,9
□ Spodnie	169,9
□ Spodnie	79,69
□ Skarpetki	1,89
□ Skarpetki	2,9
□ Skarpetki	1,99

-- jest GROUP BY

```
select nazwa, avg(cena)
from produkt
group by nazwa
having avg(cena)>100;
```

□ NAZWA	AVG(CENA)
□ Buty	106,566.. (skrócony)
□ Spodnie	124,795

Instrukcje arytmetyczne (1 / 2)



19

AVG(wyrażenie)

- Podaje średnią z wyrażenia.

Przykład:

- `SELECT miejscowosc, avg(sredniaPensja) FROM place GROUP BY miejscowosc;`

COUNT(wyrażenie)

- Podaje liczbę wartości w rekordach różnych od null.

Przykład:

- `SELECT miejscowosc, count(*) FROM place GROUP BY miejscowosc;`

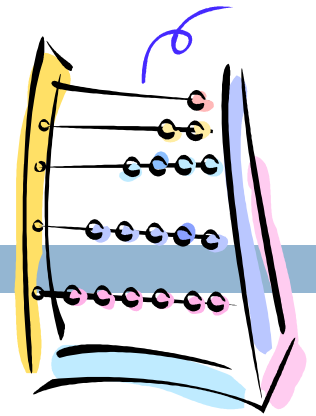
COUNT(DISTINCT wyrażenie, [wyrażenie...])

- Podaje liczbę różnych wartości w pobranych rekordach różnych od null.

Przykład:

- `SELECT count(distinct miasto) FROM place;`

Instrukcje arytmetyczne (2/2)



20

MIN(wyrażenie)

- Podaje minimalną wartość z wyrażenia.

Przykład:

- `SELECT pracownik, min(pensja) FROM place GROUP BY pracownik;`

MAX(wyrażenie)

- Podaje maksymalną wartość z wyrażenia.

Przykład:

- `SELECT pracownik, max(pensja) FROM place GROUP BY pracownik;`

SUM(wyrażenie)

- Podaje sumaryczną wartość z wyrażenia.

Przykład:

- `SELECT miejscowosc, sum(pensja) FROM place GROUP BY miejscowosc;`

Wartość NULL



21

- Wiersze z wartościami NULL są pomijane podczas grupowania, jeżeli grupowanie następuje według kolumny zawierającej wartość NULL.
- Analogicznie licząc średnią z kolumny w której występują wartości NULL, te wiersze zostaną pominięte.
- Tym samym średnia dla listy wartości: 4, 3, NULL, 5, NULL, NULL wynosi $12 / 3 = 4$, a nie $12 / 6 = 2$

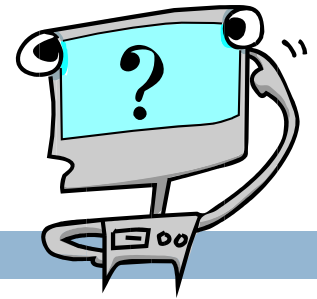
Wartości NULL – puste pola



22

□ NAZWA	CENA
□ -----	
□ Buty	99,9
□ Buty	89,9
□ Buty	129,9
□ Spodnie	169,9
□ Spodnie	79,69
□ Skarpetki	1,89
□ Skarpetki	2,9
□ Skarpetki	1,99
□ Buty	

Podzapytania



23

- Podzapytania są tworzone w celu zawarcia jednej instrukcji `select` w warunku innej instrukcji takiej jak `delete`, czy też `select`.
- Ma to między innymi zastosowanie w przypadku, gdy należy wybrać wiersze z jednej tabeli, dla warunków zdefiniowanych w ramach innej tabeli.

Składnia instrukcji z podzapytaniem



24

Ogólna składnia tego polecenia jest następująca:

- instrukcjaSQL where kolumna1 >(select kolumna2 from tabela2);
- Uwaga! Zapytanie wewnętrzne musi określać wartość w sposób jednoznaczny – nie można zastosować znaku równości, gdy zwracanych jest wiele wierszy.



Przykłady tabel

25

□ T1

□ K1	K2	NAZWA
□ 1	1	jeden
□ 2	2	dwa
□ 3	3	trzy

□ T2

□ K1	K2	NAZWA
□ 1	1	jeden
□ 2	2	dwa
□ 3	3	trzy

Przykłady podzapytań



26

Wyświetlane są wszystkie wiersze z tabeli t1, dla których kolumna k1 jest większa od maksymalnej wartości w kolumnie k2.

□ `select k1 from t1 where k1 >=`
`(select max(k2) from t2);`

□ K1	K2	NAZWA
□ 3	3	trzy

Składnia podzapytań z operatorami IN, ANY oraz SOME



27

- Składnia podzapytań ze słowami kluczowymi any, in oraz some jest następująca:

operand operatorPorównania ANY (podzapytanie)

operand IN (podzapytanie)

operand operatorPorównania SOME (podzapytanie)

Podzapytania a słowo kluczowe ANY

28

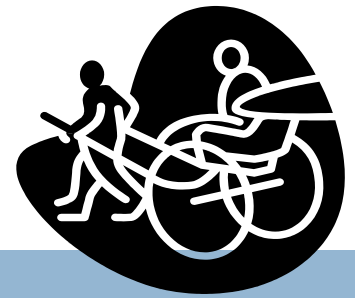


Słowo kluczowe ANY, które musi znajdować się po operatorze porównania, oznacza zwrócenie wartości true, jeżeli warunek będzie spełniony dla dowolnego z wierszy zwracanych przez podzapytanie.

Przykład:

- `select k1 from t1 where k1 > any (select k1 from t2);`
 - K1
 - -----
 - 3
 - 2

Podzapytania a słowo kluczowe IN



29

- Słowo IN jest synonimem dla ANY.

Przykład:

- `select k1 from t1 where k1=any(select k1 from t2);`
- `select k1 from t1 where k1 in (select k1 from t2);`

- Wynik:

- K1

- -----

- 1

- 2

- 3

Podzapytania a słowo kluczowe SOME



30

Słowo kluczowe SOME jest synonimem dla ANY.

Przykład:

- `select k1 from t1 where k1 <> any(select k1 from t2);`
- `select k1 from t1 where k1 <> some(select k1 from t2);`

□ Wynik:

- K1

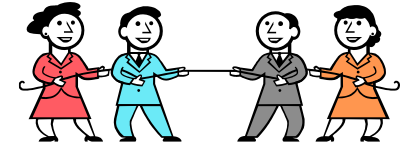
- -----

- 1

- 2

- 3

Podzapytania a słowo kluczowe ALL



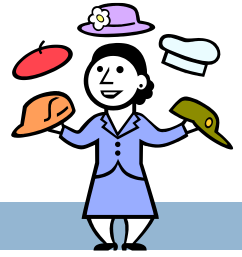
31

Składnia podzapytań ze słowem kluczowym all jest następująca:

□ operand operator Porównania ALL (podzapytanie);

Słowo kluczowe ALL oznacza, że ma być zwrócony wynik zapytania, jeżeli warunek jest prawdziwy dla wszystkich wierszy zwracanych przez podzapytanie.

Przykłady podzapytań ze słowem kluczowym ALL



32

- `select k1 from t1 where k1 <= all(select k1 from t2);`
- `select * from t1 where k1 <= all(select k1 from t2);`
- `select * from t1 where k1 < all(select max(k1) from t2);`

Wyrażenie EXISTS



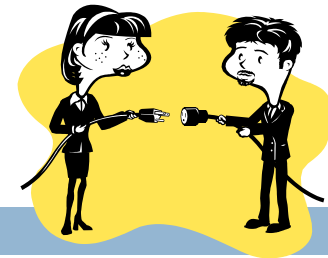
33

- Jeżeli podzapytanie nie zwraca żadnych wartości, to wyrażenie EXISTS podzapytanie ma wartość FALSE, a NOT EXISTS podzapytanie ma wartość TRUE.

Przykład:

- `select k1 from t1 where exists (select * from t2);`
- Wynik:
 - K1
 - -----
 - 1
 - 2
 - 3

Złączenia - JOIN

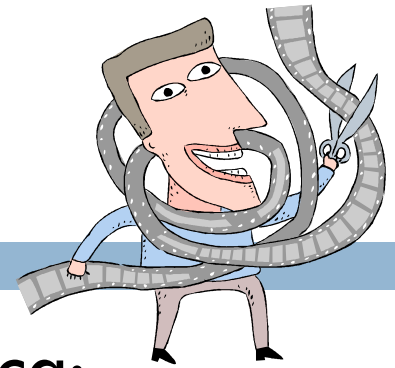


34

- Składnia polecenia JOIN pozwala tworzyć złączenia.
- Złączenia są używane, gdy mają być wyświetlone wyniki z wielu tabel, również z pustymi wartościami pochodzącymi z jednej z tych tabel.
- Przykładem może być sytuacja, kiedy mają zostać wyświetlone wszystkie pozycje z jednej tabeli, w przypadku, gdy występują pasujące do warunku pozycje z drugiej tabeli powinny one zostać wyświetlone. W innym przypadku powinny pozostać puste.

Składnia instrukcji JOIN

35



Ogólna postać instrukcji join jest następująca:

- `odwołanieDoTabeliLewej LEFT [OUTER] JOIN
odwołanieDoTabeliPrawej [warunekZłączenia]`
- `odwołanieDoTabeliLewej RIGHT [OUTER] JOIN
odwołanieDoTabeliPrawej [warunekZłączenia]`

Warunki złączenia JOIN



36

- Warunek złączenia, definiowany jest przez słowo kluczowe ON i jest identyczny jak składnia warunku WHERE.
- Warunkiem złączenia może być również klauzula USING, która pozwala na złączenie poprzez kolumnę, która znajduje się w obu tabelach oraz posiada w nich identyczną nazwę.

Przykłady zapytań JOIN z pomocą USING i ON



37

Poniższe dwa przykłady dają identyczny wynik:

- ❑ `SELECT t1.NAZWA, t2.NAZWA FROM t1 LEFT JOIN t2 USING (k1);`
- ❑ `SELECT t1.NAZWA, t2.NAZWA FROM t1 LEFT JOIN t2 ON t1.k1=t2.k1;`

Zapytania JOIN

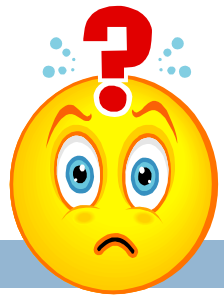


38

- ❑ INSERT INTO **T1** VALUES (4, 4, 'cztery');
- ❑ INSERT INTO **T1** VALUES (5, 5, 'pięć');
- ❑ INSERT INTO **T2** VALUES (6, 6, 'sześć');

- ❑ **T1** posiada wartości 1,2,3,4,5 a **T2**:1,2,3,6.

Złączenie lewostronne



39

BRAK JOIN

☐ `select t1.nazwa, t2.nazwa
from t1, t2
where t1.k1=t2.k1;`

- | | |
|--------------------------------|-------|
| <input type="checkbox"/> NAZWA | NAZWA |
| <input type="checkbox"/> jeden | jeden |
| <input type="checkbox"/> dwa | dwa |
| <input type="checkbox"/> trzy | trzy |

WYSTĘPUJE LEFT JOIN

☐ `select t1.nazwa, t2.nazwa
from t1 left join t2 on
t1.k1=t2.k1;`

- | | |
|---------------------------------|-------|
| <input type="checkbox"/> NAZWA | NAZWA |
| <input type="checkbox"/> jeden | jeden |
| <input type="checkbox"/> dwa | dwa |
| <input type="checkbox"/> trzy | trzy |
| <input type="checkbox"/> pięć | |
| <input type="checkbox"/> cztery | |

Złączenie lewostronne a prawostronne



40

WYSTĘPUJE LEFT JOIN

□ `select t1.nazwa, t2.nazwa
from t1 left join t2 on
t1.k1=t2.k1;`

- NAZWA NAZWA
- jeden jeden
- dwa dwa
- trzy trzy
- pięć
- cztery

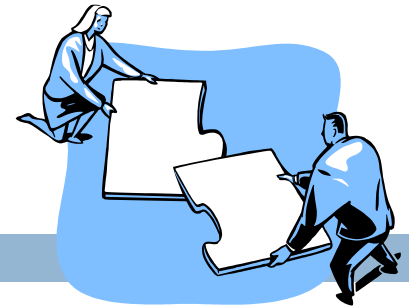
WYSTĘPUJE RIGHT JOIN

□ `select t1.nazwa, t2.nazwa
from t1 right join t2 on
t1.k1=t2.k1;`

- NAZWA NAZWA
- jeden jeden
- dwa dwa
- trzy trzy
- sześć

Inne przykłady zapytań JOIN

41



- ❑ `select * from tabela1 left join tabela2 on tabela1.id=tabela2.id;`
- ❑ `select * from tabela1 left join tabela2 on tabela1.id=tabela2.id left join tabela3 on tabela2.id=tabela3.id;`

Instrukcja UNION

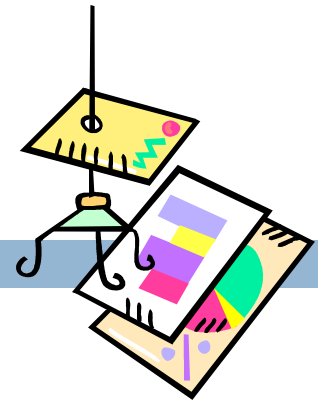


42

- Składnia UNION jest wykorzystywana do połączenia wyników z wielu instrukcji SELECT w jeden zbiór wynikowy.
- Kolumny wymienione na odpowiadających sobie pozycjach muszą mieć ten sam typ.

Składnia instrukcji UNION

43



Ogólna postać tej instrukcji jest następująca:

- SELECT ...
- UNION [ALL | DISTINCT]
- SELECT ...
- [UNION [ALL | DISTINCT]
- SELECT ...]

Przykład zapytania UNION



44

□ `select * from t1 union select * from t2;`

□ K1 K2 NAZWA

□ 1 1 jeden

□ 2 2 dwa

□ 3 3 trzy

□ 4 4 cztery

□ 5 5 pięć

□ 6 6 sześć



Przykład zapytania UNION ALL

45

- ❑ `select * from t1 union all select * from t2;`

- ❑ K1 K2 NAZWA
- ❑ 1 1 jeden
- ❑ 2 2 dwa
- ❑ 3 3 trzy
- ❑ 4 4 cztery
- ❑ 5 5 pięć
- ❑ 1 1 jeden
- ❑ 2 2 dwa
- ❑ 3 3 trzy
- ❑ 6 6 sześć

Pytanie...



- ❑ Która instrukcja wypisuje wszystkie wartości z kolumny, nie powtarzając duplikatów:
- ❑ UNIQUE
- ❑ DISTINCT
- ❑ UNION
- ❑ JOIN