

Relacyjny model danych

Model danych

- Struktury danych



JAN	NOWAK	47
TADEUSZ	KOWALSKI	34
MACIEJ	NOWAK	26
JANINA	RZEPA	19
KUBA	TARZAN	31
JÓZEF	MALINIAK	29
JAN	NOWAK	56

- Ograniczenia integralnościowe

JANINA	RZEPA	19
KUBA	TARZAN	31 ✓

- Operacje

2

Struktury danych

- Baza danych jest zbiorem relacji (tablic)
- Domena D** jest zbiorem wartości atomowych
- Domena jest definiowana przez podanie typu danych, które mogą przyjmować wartości danych danej domeny
- Schemat relacji R** , oznaczony przez $R(A_1, A_2, \dots, A_n)$, składa się z nazwy relacji R oraz listy nazw atrybutów A_1, A_2, \dots, A_n

3

Struktury danych

- Atrybut A_i schematu relacji R opisuje rolę jaką pełni domena D w schemacie R
- D nazywamy domeną atrybutu A_i w schemacie R i oznaczamy przez **$dom(A_i)$**
- Liczba atrybutów składających się na schemat relacji R nazywamy **stopniem relacji**

STUDENT(Pesel, Nazwisko, Adres, Tel, Wiek, GPA)

4

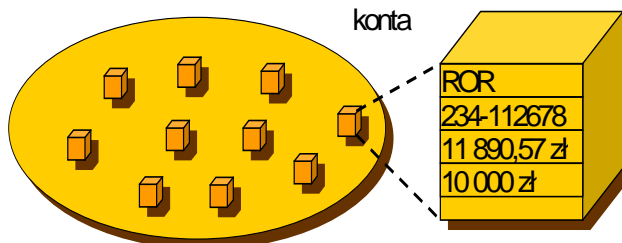
Struktury danych

- Relacją** r o schemacie $R(A_1, A_2, \dots, A_n)$, oznaczoną $r(R)$, nazywamy zbiór n -tek (krotek) postaci $r=\{t_1, t_2, \dots, t_m\}$.
- Pojedyncza krotka t jest uporządkowaną listą n wartości $t=<v_1, v_2, \dots, v_n>$, gdzie $v_i, 1 \leq i \leq n$, jest elementem $dom(A_i)$ lub specjalną wartością pustą (null value)
- i -ta wartość krotki t , odpowiadająca wartości atrybutu A_i , będzie oznaczana przez $t[A_i]$

5

Struktury danych

Relacja jest zbiorem **krotek** (k-wartości), które są listami wartości.



6

Alternatywna definicja relacji

- Wyświetlana relacja ma postać tablicy: krotki są wierszami tej tablicy. Nagłówki kolumn są **atrybutami**
- Relacja** $r(R)$ jest relacją matematyczną stopnia n zdefiniowaną na zbiorze domen $\text{dom}(A_1), \text{dom}(A_2), \dots, \text{dom}(A_n)$ będącą podzbiorem iloczynu kartezjańskiego domen definiujących R :
 $r(R) \subseteq \text{dom}(A_1) \times \text{dom}(A_2) \times \dots \times \text{dom}(A_n)$

7

Baza danych

- Baza danych* = zbiór *relacji*
- Schemat bazy danych* = zbiór *schematów relacji*
- Relacja* = zbiór *krotek*
- Schemat relacji* = zbiór $\{\text{atrybut, typ danej, [ograniczenia integralnościowe]}\}$
- Krotka* = lista atomowych *wartości*

8

Charakterystyka relacji

- Uporządkowanie krotek w relacji – nie jest elementem definicji
- Uporządkowanie wartości w ramach krotki – nie jest istotne
- Wartości atrybutów w ramach krotki (NFNF, puste, atomowe)
- Interpretacja relacji – rodzaj predykatu lub asercji

9

Ograniczenia integralnościowe (1)

- Ograniczenie domeny atrybutu
Zbiór wartości domeny atrybutu może być zawężony przez wyrażenie logiczne do pewnego podzioru: przedziału lub wyliczeniowej listy wartości
- Ograniczenie integralnościowe klucza relacji – unikalność krotek relacji
Relacja jest zbiorem krotek, stąd, z definicji zbioru, wszystkie krotki relacji muszą być unikalne

10

Ograniczenia integralnościowe (2)

- Ograniczenie na unikalność krotek relacji (cd)
Każdy podzbiór SK atrybutów relacji R , taki że dla każdych dwóch krotek ze zbioru $r(R)$ zachodzi $t1[SK] \neq t2[SK]$, jest nazywany **nadkluczem** (ang. super key) relacji R . Każda relacja musi mieć co najmniej jeden klucz – cały schemat relacji
 $SK = \{STUDENT.NrIndeksu\}$

11

Ograniczenia integralnościowe (3)

- Ograniczenie na unikalność krotek relacji (cd)
Superklucz może posiadać nadmiarowe atrybuty.
Kluczem K schematu relacji R nazywamy superklucz schematu R o takiej własności, że usunięcie dowolnego atrybutu A z K powoduje, że $K' = K - A$ nie jest już superkluczem.
Klucz jest minimalnym superkluczem zachowującym własność unikalności krotek relacji
Schemat relacji może posiadać więcej niż jeden klucz

12

Ograniczenia integralnościowe (4)

- Ograniczenie integralności klucza podstawowego
Jeden z kluczy relacji może być wyróżniony jako **klucz podstawowy** jednoznacznie identyfikujący krotki relacji. W związku z tym klucz podstawowy nie może przyjmować wartości pustych
Pozostałe klucze schematu relacji nazywamy kluczami **wtórnymi** lub **kandydującymi**

13

Ograniczenia referencyjne

Dane są relacje $R1$ i $R2$. Podzbiór FK atrybutów relacji $R1$ nazywany jest **kluczem obcym** $R1$ (ang. foreign key), jeżeli:

- Atrybuty w FK mają taką samą domenę jak atrybuty klucza podstawowego PK relacji $R2$;
- Dla każdej krotki $t1$ relacji $R1$ istnieje krotka $t2$ relacji $R2$, taka że: $t1[FK] = t2[PK]$, lub $t1[FK] = null$

14

Przykład

Dane dwie relacje z ograniczeniami referencyjnymi

Pracownicy

IdPrac	Nazwisko	Etat	Placa	Szef	IdZesp
120	Kowalski	Analitik	850	100	10
100	Tarzan	Dyrektor	1700		10
130	Nowak	Referent	600	100	10
110	Józik	Kierownik	1200	100	20
140	Nowacki	Analitik	800	110	20
150	Bunio	Se kretarka	700	100	10

Zespoły

IdZesp	Nazwa	Lokalizacja	Budżet
10	Sprzedaż	Krucza 10	10 000
20	Badania	Piwna 30	10 000

15

Przykład (cd)

- Wstaw do relacji *Pracownicy* krotkę:
 $t = \langle 120, 'Malinowski', 'Akwizytor', 500, 100, 10 \rangle$
- Zmodyfikuj krotkę relacji *Zespoły*:
 $t[Zespoły.IdZesp] = 30$ dla $t[Zespoły.IdZesp]=20$
- Usuń krotkę relacji *Pracownicy*, taką że:
 $t[Nazwisko] = 'Tarzan'$

16

Relacyjne języki dostępu do danych

Algebra relacji, rachunek krotek



SQL (Structured Query Language) - IBM, Oracle
Quel - Ingres
QBE (Query by Example) - IBM

17

Standard SQL

- 1965 - Początek prac nad standardem języka bazy danych (DBTG – CODASYL)
- 1981 - Początek prac nad standardem języka relacyjnej bazy danych (SQL)
- 1986 - NDL (Network-structured Data Language)
- 1986 - SQL86 ANSI
- 1987 - SQL86 ISO
- 1989 - SQL89 (ISO/ANSI)
- 1991 - początek prac nad SQL3
- 1992 - SQL92 (ISO/ANSI)
- 1999/2000 – SQL 3
- SQL 4

18

Podstawowe operacje algebry relacji

- Operacje na zbiorach
 - UNION, INTERSECTION, SET DIFFERENCE
 - CARTESIAN PRODUCT, DIVISION
- Operacje 'relacyjne' (opracowane dla relacyjnego modelu danych)
 - SELECT, PROJECT, JOIN
 - OUTER JOIN, GROUP BY

19

Operacja selekcji - SELECT

- przeznaczenie: wyodrębnienie podzbioru krotek relacji, które spełniają warunek selekcji
- notacja: $\sigma_{\langle \text{warunek selekcji} \rangle} (\langle \text{Nazwa relacji} \rangle)$, gdzie warunek selekcji jest zbiorem predykatów postaci:
 $\langle \text{nazwa atrybutu} \rangle \langle \text{operator relacyjny} \rangle \langle \text{literal} \rangle$
lub
 $\langle \text{nazwa atrybutu} \rangle \langle \text{operator relacyjny} \rangle \langle \text{nazwa atrybutu} \rangle$
połączonych operatorami logicznymi: AND lub OR. Zbiór operatorów relacyjnych zawiera następujące elementy: $\{=, \neq, <, \leq, >, \geq\}$

20

Operacja selekcji - Przykłady

- $\sigma_{\text{IdPrac} = 100} (\text{Pracownicy})$
- $\sigma_{\text{Płaca} > 1000} (\text{Pracownicy})$
- $\sigma_{(\text{IdZesp}=10 \text{ AND } \text{Płaca} > 1000) \text{ OR } (\text{IdZesp}=20) \text{ AND } \text{Płaca} > 800)} (\text{Pracownicy})$
- $\sigma_{\text{Etat}='Analityk' \text{ AND } (\text{Płaca} \geq 700 \text{ AND } \text{Płaca} < 1200)} (\text{Pracownicy})$
- własności: operacja selekcji jest komutatywna:
 $\sigma_{\langle \text{war1} \rangle} (\sigma_{\langle \text{war2} \rangle} (R)) = \sigma_{\langle \text{war2} \rangle} (\sigma_{\langle \text{war1} \rangle} (R))$

21

Operacja selekcji w języku SQL

```
SELECT * FROM studenci  
WHERE rok_studiów = 4  
AND średnia_ocen > 4.5
```

```
SELECT * FROM studenci  
WHERE UPPER(nazwisko) = 'MORZY'  
OR stypendium < zasiłek
```

```
SELECT * FROM studenci  
WHERE (rok_studiów, typ_studiów, średnia) = (3,  
'podyplomowe', 5)
```

22

Operacja projekcji - Project

przeznaczenie: wyodrębnienie wybranych atrybutów relacji

notacja: $\pi_{\langle \text{lista atrybutów} \rangle} (\langle \text{Nazwa relacji} \rangle)$, gdzie lista atrybutów jest podzbiorem atrybutów ze schematu relacji

własności: operacja projekcji nie jest komutatywna. Składanie operacji projekcji jest możliwe przy spełnieniu pewnych dodatkowych warunków:

$\pi_{\langle \text{lista1} \rangle} (\pi_{\langle \text{lista2} \rangle} (R)) = \pi_{\langle \text{lista1} \rangle} (R)$
jeżeli *lista2* zawiera wszystkie atrybuty *listy1*

23

Operacja projekcji - Przykłady

- $\pi_{\text{Nazwisko, Etat}} (\text{Pracownicy})$
- $\pi_{\text{Etat}} (\text{Pracownicy})$
- $\pi_{\text{IdPrac, Nazwisko}} (\text{Pracownicy})$

24

Operacja projekcji w języku SQL (1)

Rozszerzenia języka SQL

Operacje na danych: +, -, *, /, ||

*1,22 * cena, cena - rabat, dochód / 12, imię || nazwisko*

Funkcje: arytmetyczne, znakowe, na datach, konwersji, ogólne

UPPER(nazwisko), SIN(kąt), LN(podstawa),

ROUND(składka, 1),

TO_CHAR(data, 'Day DD-MM-YYYY')

25

Operacja projekcji w języku SQL (2)

*SELECT upper(nazwisko), 12*placa, nullif(premia, 0)
FROM pracownicy*

Aliaszy wyrażień: wyrażenie AS alias

TO_CHAR(data, 'Day DD-MM-YYYY') AS data

ROUND(składka, 1) AS składka_zaokrąglona

26

Usuwanie duplikatów

SELECT [DISTINCT | ALL] ...

Przykłady:

SELECT adres_miasto

FROM studenci

SELECT DISTINCT adres_miasto

FROM studenci

27

Porządkowanie relacji wynikowej

ORDER BY wyrażenie1 [ASC | DESC], ...

SELECT nazwisko

FROM pracownicy

ORDER BY zespół ASC, etat ASC, placa DESC

28

Składanie operacji

Sekwencja wielu operacji, w której kolejne operacje są wykonywane na pośrednich wynikach operacji poprzednich, może być zastąpiona pojedynczą operacją złożoną, powstałą przez zagnieżdżenie operacji elementarnych:

Przykłady:

PracownicyZesp10 ← $\sigma_{IdZesp = 10}$ (Pracownicy)

Wynik ← $\pi_{IdPrac, Nazwisko}$ (PracownicyZesp10)



$\pi_{IdPrac, Nazwisko} (\sigma_{IdZesp = 10} (Pracownicy))$

29

Operacje na zbiorach

Kompatybilność relacji

Dwie relacje: $R(A_1, \dots, A_n)$ i $S(B_1, \dots, B_n)$ są kompatybilne, jeżeli mają ten sam stopień i jeżeli $\text{dom}(A_i) = \text{dom}(B_i)$ dla $1 \leq i \leq n$

Operacje na zbiorach

Dane są dwie kompatybilne relacje: $R(A_1, \dots, A_n)$ i $S(B_1, \dots, B_n)$. Operacje sumy, iloczynu i różnicy relacji R i S są zdefiniowane następująco:

30

Operacje na zbiorach

- suma:** Wynikiem tej operacji, oznaczanej przez $R \cup S$, jest relacja zawierająca wszystkie krotki, które występują w R i wszystkie krotki, które występują w S, z wyłączeniem duplikatów krotek. Operacja sumy jest operacją komutatywną: $R \cup S = S \cup R$
- iloczyn:** Wynikiem tej operacji, oznaczanej przez $R \cap S$, jest relacja zawierająca wszystkie krotki, które występują zarówno w R i S. Operacja iloczynu jest operacją komutatywną: $R \cap S = S \cap R$
- różnica:** Wynikiem tej operacji, oznaczanej przez $R - S$, jest relacja zawierająca wszystkie krotki, które występują w R i nie występują w S. Operacja różnicy nie jest operacją komutatywną: $R - S \neq S - R$

31

Operacje na zbiorach - Przykłady

Uczniowie

Imię	Nazwisko
Ala	Kusiak
Edek	Musiak
Adam	Zajac
Olek	Strus
Ola	Buba

Instruktorzy

Imię	Nazwisko
Jan	Kuc
Edek	Musiak
Wacek	Misiek

Uczniowie \cap Instruktorzy

Imię	Nazwisko
Edek	Musiak

Uczniowie \cup Instruktorzy

Imię	Nazwisko
Ala	Kusiak
Edek	Musiak
Adam	Zajac
Olek	Strus
Ola	Buba
Jan	Kuc
Wacek	Misiek

Uczniowie - Instruktorzy

Imię	Nazwisko
Ala	Kusiak
Adam	Zajac
Olek	Strus
Ola	Buba

Instruktorzy - Uczniowie

Imię	Nazwisko
Jan	Kuc
Wacek	Misiek

32

Operacje na zbiorach - SQL

```
SELECT nr_konta, saldo FROM filia_A
WHERE saldo > 10000
UNION
```

```
SELECT konto_nr, SUM(operacje) FROM filia_B
GROUP BY konto_nr
```

```
SELECT imię, nazwisko, data_urodzenia FROM filia_A
INTERSECT
SELECT imię, nazwisko, data_urodzenia FROM filia_B
```

```
SELECT imię, nazwisko, data_urodzenia FROM filia_A
EXCEPT
SELECT imię, nazwisko, data_urodzenia FROM filia_B
```

33

Iloczyn Kartezjański

- Dane są dwie relacje: $R(A_1, \dots, A_n)$ i $S(B_1, \dots, B_m)$. Wynikiem iloczynu kartezjańskiego relacji R i S, oznaczonym przez $R \times S$, jest relacja Q stopnia $n+m$ i schemacie: $Q(A_1, \dots, A_n, B_1, \dots, B_m)$.
- Krotkom w relacji Q odpowiadają wszystkie kombinacje krotek z relacji R i S. Jeżeli relacja R ma n_r krotek, a relacja S ma n_s krotek, to relacja Q będzie miała $n_r \cdot n_s$ krotek

34

Iloczyn Kartezjański - Przykład

Pracownicy

Imię	Nazwisko
Ala	Kusiak
Edek	Musiak
Adam	Zajac

Zespoły

Nazwa	Lokalizacja
Reklama	Krucza 10
Badania	Piotrowo 3A

Pracownicy x Zespoły

Imię	Nazwisko	Nazwa	Lokalizacja
Ala	Kusiak	Reklama	Krucza 10
Edek	Musiak	Reklama	Krucza 10
Adam	Zajac	Reklama	Krucza 10
Ala	Kusiak	Badania	Piotrowo 3A
Edek	Musiak	Badania	Piotrowo 3A
Adam	Zajac	Badania	Piotrowo 3A

35

Operacja połączenia - JOIN

- przeznaczenie:** łączenie na podstawie warunku połączeniowego wybranych krotek z dwóch relacji w pojedynczą krotkę
- notacja:** operacja połączenia relacji $R(A_1, \dots, A_n)$ i $S(B_1, \dots, B_m)$, jest oznaczona jako:

$R \underset{\langle \text{warunek połączeniowy} \rangle}{\Join} S$

gdzie **warunek połączeniowy** jest zbiorem predykatów połączonych operatorami logicznymi AND. Predykaty są postaci: $A_i \theta B_j$, gdzie A_i i B_j są **atrybutami połączeniowymi**, a A_i jest atrybutem R, a B_j jest atrybutem S, $\text{dom}(A_i) = \text{dom}(B_j)$, a θ jest operatorem relacyjnym ze zbioru $\{=, \neq, <, \leq, >, \geq\}$.

36

Operacja połączenia - JOIN

- Ogólna postać operacji połączenia, gdzie jest dowolnym operatorem relacyjnym jest nazywana połączeniem typu THETA (ang. THETA JOIN)
- Operacja połączenia, dla której θ jest operatorem $=$, nazywana jest połączeniem równościowym (ang. EQUI JOIN)
- Operacja połączenia, dla której θ jest operatorem różnym od $=$, nazywana jest połączeniem nierównościowym (ang. NON-EQUI JOIN)
- Operacja połączenia równościowego, w której jeden z atrybutów połączeniowych jest usunięty ze schematu relacji wynikowej, jest nazywana połączeniem naturalnym (ang. NATURAL JOIN). Połączenie naturalne jest oznaczane jako: $R * S$, przy czym wymagane jest, by atrybuty połączeniowe w obu relacjach miały taką samą nazwę

37

Operacja połączenia - Przykłady

Pracownicy					Zespoły	
IdPrac	Imię	Nazwisko	Szef	IdZesp	IdZesp	Nazwa
100	Jan	Miś		10	10	Reklama
110	Piotr	Wilk	100	10	20	Badania
120	Roman	Lis	100	20		

Pracownicy					Pracownicy				
IdPrac	Imię	Nazwisko	Szef	IdZesp	IdPrac	Imię	Nazwisko	Szef	IdZesp
110	Piotr	Wilk	100	10	100	Jan	Miś		10
120	Roman	Lis	100	20	100	Jan	Miś		10

Pracownicy * Zespoły

IdPrac	Imię	Nazwisko	Szef	IdZesp	Nazwa
100	Jan	Miś		10	Reklama
110	Piotr	Wilk	100	10	Reklama
120	Roman	Lis	100	20	Badania

38

Operacja połączenia zewnętrznego – OUTER JOIN

przeznaczenie: łączenie na podstawie warunku połączeniowego wybranych krotek z dwóch relacji w pojedynczą krotkę; w przypadku gdy danej krotce relacji nie odpowiada żadna krotka drugiej relacji, krotka ta jest łączona z krotką pustą

- operacja połączenia zewnętrznego lewostronnego** relacji $R(A_1, \dots, A_n)$ i $S(B_1, \dots, B_m)$, jest oznaczona jako:

$$R \bowtie_{<\text{warunek połączeniowy}>} S$$

oznacza, że w relacji wynikowej znajdują się krotki odpowiadające wszystkim krotkom relacji R (lewostronnej).

39

Operacja połączenia zewnętrznego – OUTER JOIN

- operacja połączenia zewnętrznego prawostronnego** relacji $R(A_1, \dots, A_n)$ i $S(B_1, \dots, B_m)$, jest oznaczona jako:

$$R \bowtie_{<\text{warunek połączeniowy}>} S$$

oznacza, że w relacji wynikowej znajdują się krotki odpowiadające wszystkim krotkom relacji S (prawostronnej)

40

Operacja połączenia zewnętrznego – OUTER JOIN

- operacja połączenia zewnętrznego dwustronnego** relacji $R(A_1, \dots, A_n)$ i $S(B_1, \dots, B_m)$, jest oznaczona jako:

$$R \bowtie_{<\text{warunek połączeniowy}>} S$$

oznacza, że w relacji wynikowej znajdują się krotki odpowiadające wszystkim krotkom relacji R i wszystkim krotkom relacji S

41

Operacja połączenia zewnętrznego – Przykład

Pracownicy

IdPrac	Imię	Nazwisko	Szef	IdZesp
100	Jan	Miś		10
110	Piotr	Wilk	100	10
120	Roman	Lis	100	20

Pracownicy

$<\text{Szef=IdPrac}>$ Pracownicy

IdPrac	Imię	Nazwisko	Szef	IdZesp	IdPrac	Imię	Nazwisko	Szef	IdZesp
100	Jan	Miś		10					
110	Piotr	Wilk	100	10	100	Jan	Miś		10
120	Roman	Lis	100	20	100	Jan	Miś		10

42

Operacja połączenia - SQL

Stary styl:

```
SELECT a.*, b.*
FROM konta AS a, operacje AS b
WHERE a.id_konta = b.id_konta AND
      a.data_założenia > '1-01-1995'
```

```
SELECT a.*, b.*
FROM konta AS a, operacje AS b
WHERE a.id_konta = b.id_konta (+)
```

43

Operacja połączenia - SQL

Nowy styl:

```
SELECT *
FROM konta NATURAL JOIN operacje
```

```
SELECT a.*, b.*
FROM konta AS a LEFT OUTER JOIN operacje AS b
ON a.id_konta = b.id_konta
```

44

Wartości puste

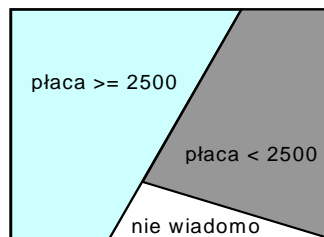
- Zbiory wartości wszystkich typów danych zawierają dodatkowo wyróżnioną wartość pustą – NULL, np. $INTEGER(3) - \langle -999, 999 \rangle \cup NULL$
- Możliwość występowania wartości pustych wymaga specyficznej obsługi wartości pustych.
 - $10 + NULL = NULL$
 - $1024 * 118 + NULL * 119 = NULL$
 - $SUM(płaca) = NULL$, gdy choć jedna z sumowanych wartości jest wartością pustą
- Funkcja $NULLIF(wyrażenie, wartość\ jeśli\ NULL)$

```
SELECT SUM(NULLIF(dochód, 1500))
FROM ...
```

45

Wartości puste

Również wyrażenia logiczne oprócz wartości *prawda* i *fałsz* mogą przyjmować wartości nieokreślone



WHERE $płaca \geq 2000$ OR $płaca < 2000$ OR $płaca$ IS NULL

46

Wartości puste

Logika trójwartościowa

OR	T	F	Null
T	T	T	T
F	T	F	Null
Null	T	Null	Null

AND	T	F	Null
T	T	F	Null
F	F	F	F
Null	Null	F	Null

NOT	T	F	Null
	F	T	Null

47

Definiowanie struktur danych SQL - Typy danych

Typ	Przykład użycia
CHARACTER/CHAR	Tytuł_filmu CHAR(25)
CHAR VARYING/VARCHAR	Nazwisko VARCHAR (30)
NATIONAL CHAR/NCHAR	Miasto NCHAR (20)
NCHAR VARYING	Miasto NCHAR VARYING (20)
INTEGER/INT	Nr_konta INT
NUMERIC (s, p)	Saldo NUMERIC (8, 2)
DECIMAL (s, p)	Cena DECIMAL (7, 2)
FLOAT	Odległość FLOAT
REAL	Szerokość REAL
DOUBLE PRECISION	Wielkość DOUBLE PRECISION
BIT	Flagi BIT
BIT VARYING	Pole_flagi BIT VARYING
DATE	Data_urodzenia DATE
TIME	Początek_zajęć TIME
TIME WITH ZONE	Czas_odlotu TIME WITH ZONE
TIMESTAMP	Spotkanie TIMESTAMP
TIMESTAMP WITH ZONE	Spotkanie TIMESTAMP WITH ZONE
INTERVAL	Czas_pracy INTERVAL