

2 Polecenie CREATE

2.1 Uproszczona składnia polecenia CREATE TABLE

CREATE TABLE nazwa_tabeli

(nazwa_kolumny typ_danych [**NULL|NOT NULL**]

[, nazwa_kolumny typ_danych [**NULL|NOT NULL**]]....);

2.2 Rozszerzona składnia CREATE TABLE

CREATE TABLE nazwa_tabeli

(nazwa_pola1 typ_danych1 ograniczenie1,

nazwa_pola2 typ_danych2 ograniczenie2,

nazwa_pola3 typ_danych3

[**NULL|NOT NULL**] [**DEFAULT** wartosc_domyslna],

PRIMARY KEY (nazwa_pola1)

FOREIGN KEY nazwa_klucza_obcego (nazwa_pola_kolumny_obcej)

REFERENCES tabela_klucza_podstawowego (nazwa_pola1)

ON UPDATE czynność_przy_modyfikacji

ON DELETE czynność_przy_kasowaniu;

2.2.1 Ograniczenia tabel

- **PRIMARY KEY** zaznacza kolumnę (nie dopuszczającą wartości null) jako klucz główny tabeli. Wewnętrzna obsługa tego ograniczenia jest różna w różnych systemach, lecz często jest równoważna tworzeniu indeksu
- **UNIQUE** również gwarantuje jednoznaczność każdej wartości w kolumnie, lecz dopuszcza, żeby kolumna mogła być zdefiniowana jako null (zauważ, że kolumnę zadeklarowaną jako **UNIQUE** może przypaść tylko jedna wartość null).
- **DEFAULT** definiuje wartość, jaką system nadaje automatycznie, gdy użytkownik wprowadzając dane nie poda jej w sposób jawny
- **CHECK** określa, jakie dane mogą zostać wprowadzone do określonej kolumny: jest to sposób na zdefiniowanie dziedziny kolumny. Ograniczenie nakładane przez **CHECK** są czasem nazywane regułami poprawności, gdyż pozwalają one systemowi sprawdzić, czy wprowadzana do kolumny wartość jest elementem dziedziny kolumny.
- **REFERENCES** i **FOREIGN KEY** wiąże klucz główny z kluczem obcym.

2.2.2 rozszerzona składnia SQL-92

Rozszerzona składnia SQL dopuszcza:

- wartości domyślne (wartości przypisywane automatycznie) **DEFAULT**
- więzy ograniczające dopuszczalne wartości nadawane poszczególnym elementom kolumny – **CHECK**
- więzy definiujące kolumnę jako klucz główny, klucz jednoznaczny lub jako oba klucze jednocześnie **PRIMARY KEY**, **FOREIGN KEY**, **REFERENCES**, **CONSTRAINT**
- więzy ustawiające sprawdzenie zależności między kluczami głównymi a obcymi

2.3 Wstawianie danych liczbowych – liczby całkowite

- **Całkowitoliczbowe typy danych** służą do przechowywania liczb całkowitych (bez ułamków i liczb dziesiętnych)
 - Number, integer, int, smailint, tinyint ...
- **Porządkowe typy danych** służą do przechowywania liczb rosnących sekwencyjnie. W niektórych przypadkach taka cecha jest traktowana jako oddzielny typ danych
 - Serial, number

Tworzenie tabeli	Wstawianie danych
<pre>CREATE TABLE liczby (id_liczby serial, liczba_4 int4, liczba_8 int8);</pre>	<pre>INSERT INTO liczby VALUES (1, 2147483647, 12345678901234567890);</pre>
<p>NOTICE: CREATE TABLE will create implicit sequence 'liczby_id_liczby_seq' for SERIAL column 'liczby.id_liczby'</p> <p>NOTICE: CREATE TABLE/UNIQUE will create implicit index 'liczby_id_liczby_key' for table 'liczby'</p> <p>CREATE</p>	<pre>INSERT 56166 1</pre>

\d liczby

table = liczby

Field	Type	Length
id_liczby	int4 not null default nextval	4
liczba_4	int4	4
liczba_8	int8	8

Index: liczby_id_liczby_key

Składnia	Efekt
<pre>SELECT * FROM liczby;</pre>	<pre>id_liczby liczba_4 liczba_8 -----+-----+----- 1 2147483647 1234567801234567890 (1 row)</pre>

2.4 Wstawianie danych liczbowych zmiennoprzecinkowych

- **Dziesiętne typy danych** służą do przechowywania liczb z częściami ułamkowymi. Dokładnie liczby dziesiętne są znane pod nazwami decimal lub numeric. Zazwyczaj należy określić ich precyzję (całkowitą liczbę cyfr po obu stronach kropki dziesiętnej) i skalę (liczbę cyfr po kropce dziesiętnej)
 - Real, double, double precision, float, smallfloat

Tworzenie tabeli	Wstawianie danych
CREATE TABLE zmienne (liczba10 decimal(5,2), liczba11 float4, liczba12 float8, liczba13 numeric,);	INSERT INTO liczby VALUES (12.34, 1234567890.1234567890, 1234567890.1234567890, 1234567890.123456);
CREATE	INSERT 56236 1

\d zmienne

Table = zmienne

Field	Type	Length
liczba10	Numeric	5.2
liczba11	float4	4
liczba12	float8	8
liczba	numeric	30.2

Składnia	efekt
SELECT * FROM zmiene;	Liczba10 liczba11 liczba12 liczba -----+-----+-----+----- 12.34 1.23457e+09 1234567890.12346 1234567890.123456 (1 row)

2.5 Wstawianie danych logicznych

Tworzenie tabeli	Wstawianie danych
CREATE TABLE logika (Tak Boolean);	INSERT INTO logika VALUES ('T'); INSERT INTO logika VALUES ('Y');
CREATE	INSERT 56337 1 INSERT 56338 1

\d logika

Table = logika

Field	Type	length
tak	bool	1

Składnia	efekt
SELECT * FROM logika;	Tak ----- t t (2 rows)

2.6 Wstawianie walut - kwotowe

Walutowy typ danych służy do przechowywania wartości walutowych. Jeśli twój system nie ma specjalnego typu walutowego, używaj typu dokładnych liczb dziesiętnych

Tworzenie tabeli	Wstawianie danych
<pre>CREATE TABLE place (placa_zasadnicza money);</pre>	<pre>INSERT INTO place VALUES ('12345678,12'</pre>
CREATE	INSERT 56265 1

\d place

Table = place

Field	Type	Length
placa_zasadnicza	Money	4

Składnia	efekt
<pre>SELECT * FROM place;</pre>	<pre>Placa_zasadnicza ----- Z12.345.678,12 (1 row)</pre>

2.7 Wstawianie danych znakowych

- **Znakowe typy danych** służą do przechowywania liter, liczb i znaków specjalnych
 - Char (character), varchar (variable character)
 Mogą występować też inne
 - Text, long, nchar

Tworzenie tabeli	Wstawianie danych
<pre>CREATE TABLE tekst (jeden_zn char, wiele_znakow char(5), zmienny_zn varchar(5), opis text);</pre>	<pre>INSERT INTO place VALUES ('T', 'Marek', 'Marco', 'Opis tekstowy'</pre>
CREATE	INSERT 56314

\d tekst

Table = tekst

Field	Type	Length
jeden_zn	char()	1
wiele_zn	char()	5
zmienny_zn	varchar()	5
opis	text	var

Składnia	efekt
<pre>SELECT * FROM place;</pre>	<pre>jeden_zn wiele_zn zmienny_zn opis -----+-----+-----+----- T Marek Marco Opis techniczny (1 row)</pre>

2.8 Wstawianie danych typu Data

Datowy i czasowy typ danych służy do przechowywania daty, czasu oraz kombinacji daty i czasu

Tworzenie tabeli	Wstawianie danych
<pre>CREATE TABLE daty (data_ur date, data_sys date DEFAULT 'today');</pre>	<pre>INSERT INTO daty (data_ur) VALUES ('10/03/1961');</pre>
CREATE	INSERT 56326 1

\d daty

Table = daty

Field	Type	Length
data_ur	date	4
data_sys	date default 'today'	4

Składnia	efekt
<pre>SELECT * FROM daty;</pre>	<pre>Data_ur data_sys -----+----- 10-03-1961 12-12-2001 (1 row)</pre>

CREATE TABLE mozliwosci

```
(
  id_mozliw      int NOT NULL,
  data_wprow     date DEFAULT 'today',
  Nazwisko       char(10) NOT NULL,
  Imie           varchar(8) NOT NULL,
  Miasto         varchar(10) DEFAULT 'Ostroleka',
  Data_ur        date,
  dzieci         char CHECK (dzieci = 'T' OR dzieci = 'N'),
  zarobki        money,
  zonaty         boolean NOT NULL,
PRIMARY KEY (id_mozliwosci));
```

CREATE TABLE mozliwosci

```
(  
  id_mozliw      int NOT NULL,  
  data_wprow    date DEFAULT 'today',  
  Nazwisko      char(10) NOT NULL,  
  Imie          varchar(8) NOT NULL,  
  Miasto        varchar(10) DEFAULT 'Ostroleka',  
  Data_ur       date,  
  dzieci        char CHECK (dzieci = 'T' OR dzieci = 'N'),  
  zarobki       money,  
  zonaty        boolean NOT NULL,  
  PRIMARY KEY (id_mozliwosci)  
);
```

CREATE TABLE mozliwosci

```
(  
  nr_mozl       int CONSTRAINT ID_m PRIMARY KEY,  
  data_wprow    date  DEFAULT 'today',  
  Nazwisko      char(10)  NOT NULL,  
  Imie          varchar(8) NOT NULL,  
  Miasto        varchar(10) DEFAULT 'Ostroleka',  
  Data_ur       date,  
  dzieci        char  CHECK (dzieci = 'T' OR dzieci = 'N'),  
  zarobki       money,  
  zonaty        boolean  NOT NULL  
);
```

INSERT INTO mozliwosci

(zonaty, dzieci, imie, nazwisko, nr_mozl)

VALUES

```
(  
  'T', 'N', 'Marek', 'RADECKI', 1  
);
```