

Geografická báza údajov 1

Cvičenie 2

Vladimír Pelech

pelech2@uniba.sk, G-23



PRÍRODOVEDECKÁ FAKULTA
Univerzita Komenského
v Bratislave



Cvičenie 2

- Náplň:
 - Základy práce s databázou
 - Dátové typy
 - Vytvorenie tabuľky
 - Jednoduché dopytovanie
 - Zmena štruktúry už vytvorenej tabuľky
 - Cvičné úlohy
- Mgr. Vladimír Pelech, Phd., Mgr. Jozef Fábry



Otázky na opakovanie

- Čo je to databáza?
- Aký je vzájomný vzťah medzi PostgreSQL, PostGIS a pgAdmin?
- Čo je to SQL?

- 
- Zapnite si pgAdmin a zadajte heslo pre vstup do databázy.

pgAdmin 4 File Object Tools Help

Browser

Servers (5)

- PostgreSQL 9.5
- PostgreSQL 10
- PostgreSQL 12
 - Databases (2)
 - pokusnaDB
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Schemas (3)
 - cviko3
 - public
 - ukazka
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Procedures
 - Sequences
 - Tables (1)
 - Trigger Functions
 - Types
 - Views

postgres

Login/Group Roles

Tablespaces

kniznica

Dashboard Properties SQL Statistics Dependencies Dependents

Welcome

Feature rich | Maximises PostgreSQL | Open Source

Add New Server

Getting Started

PostgreSQL Documentation

pgAdmin Website

Vytvorené databázy, ktoré môžu byť priestorové alebo nepriestorové, záleží od rozšírenia

Rozšírenia, priestorová databáza musí mať priestorové rozšírenie postgis.

Vytvorené schémy v databáze *pokusnaDB*, umožňujú rozdelenie dát, napr. podľa témy.

Vytvorené tabuľky v databáze *pokusnaDB* a schéme *ukazka*

Vytvorené náhľady v databáze *pokusnaDB* a schéme *ukazka*

pgAdmin 4 File Object Tools Help

Browser

Servers (1)

PostgreSQL 9.5

Databases (2)

- postgis
- postgres
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Schemas (1)
 - public
- Login/Group Roles
- Tablespaces

Dashboard Properties SQL Statistics Dependencies Dependents Query - postgres on postgres@PostgreSQL 9.5 *

postgres on postgres@PostgreSQL 9.5

1

História SQL kódu danej tabuľky/schémy/databázy

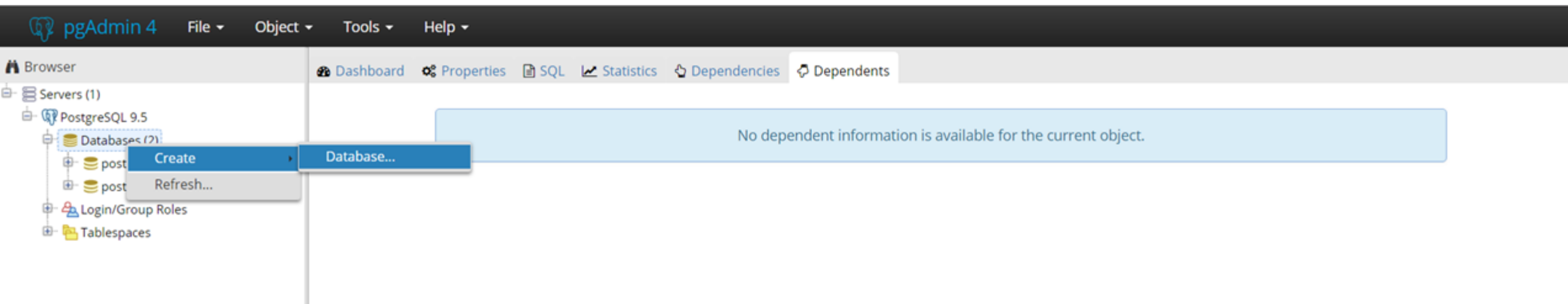
Dopytovací nástroj

Štatistika [veľkosť tabuľky (kb),...]

Data Output Explain Messages Notifications Query History

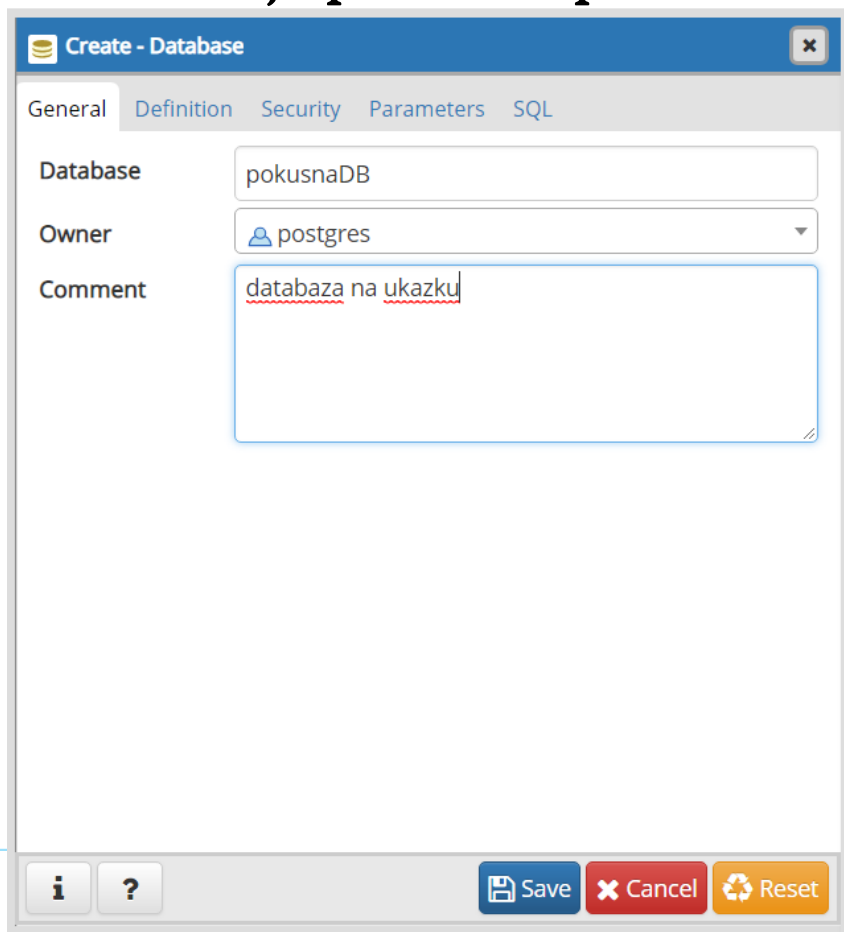
Vytvorenie databázy

- Kliknutie pravým tlačidlom na „Databases“-> Create -> Database...
Alebo je možné využiť aj SQL.



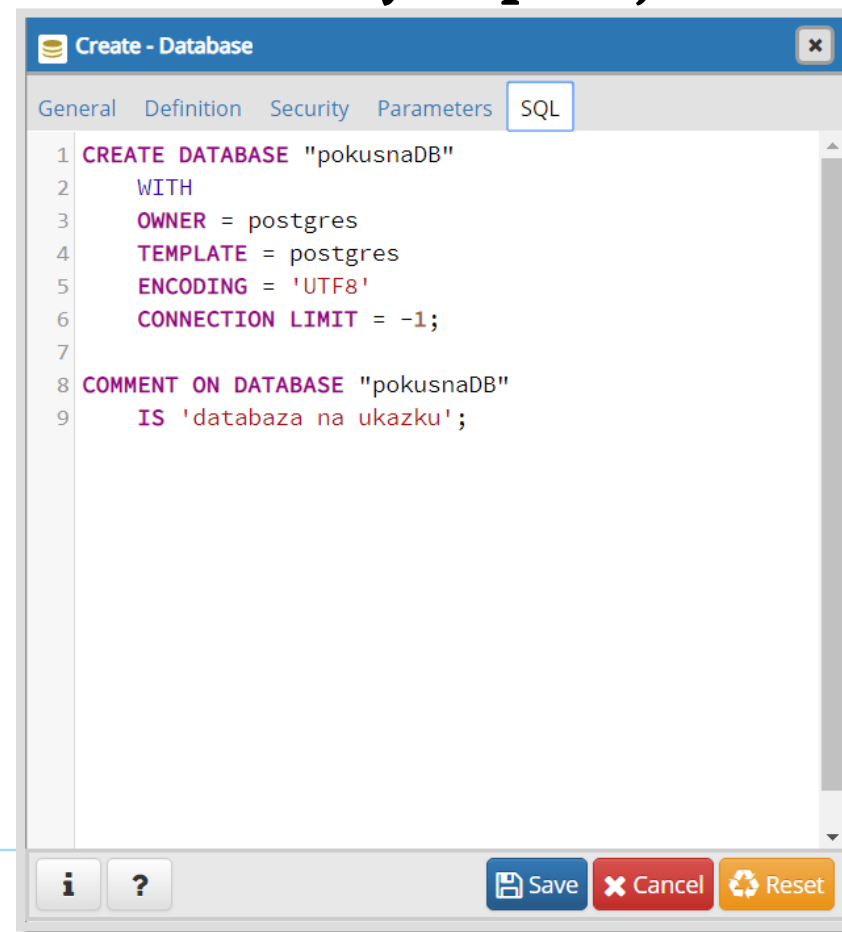
Vytvorenie databázy

Databázu je potrebné pomenovať



The screenshot shows the 'Create - Database' dialog box with the 'General' tab selected. The 'Database' field contains 'pokusnaDB'. The 'Owner' dropdown menu shows 'postgres'. The 'Comment' text area contains 'databaza na ukazku'. At the bottom, there are buttons for 'Save', 'Cancel', and 'Reset'.

Zápis v záložke GENERAL sa automaticky zapisuje ako SQL



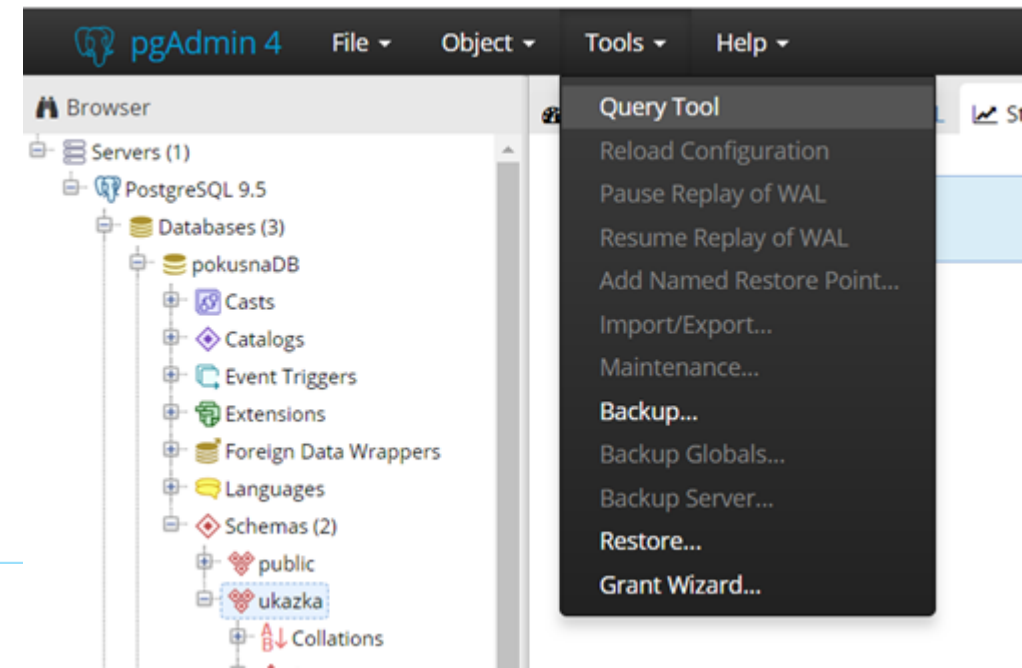
The screenshot shows the 'Create - Database' dialog box with the 'SQL' tab selected. The SQL text area contains the following SQL code:

```
1 CREATE DATABASE "pokusnaDB"
2 WITH
3 OWNER = postgres
4 TEMPLATE = postgres
5 ENCODING = 'UTF8'
6 CONNECTION LIMIT = -1;
7
8 COMMENT ON DATABASE "pokusnaDB"
9 IS 'databaza na ukazku';
```

At the bottom, there are buttons for 'Save', 'Cancel', and 'Reset'.

Spustenie dopytovacieho nástroja

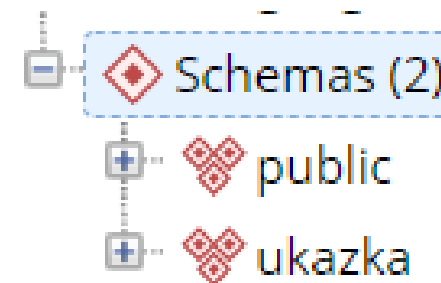
- Dostupné až po vstupe do databázy, inak nedostupné.
- Pre spustenie je potrebné si zvoliť požadovanú databázu.
- Spustenie cez: Tools->Query Tool



Vytvorenie schémy

- Podobne ako vytvorenie databázy.

The screenshot shows a PostgreSQL client window titled "pokusnaDB on postgres@PostgreSQL 9.5". The query editor contains the command `create schema ukazka;`. Below the editor, the "Messages" tab is active, displaying the output: `CREATE SCHEMA` and `Query returned successfully in 193 msec.`. On the right, the "Browser" pane shows the database hierarchy. Under "Databases (3)", "pokusnaDB" is expanded, showing various database objects. The "Schemas (1)" folder is highlighted, and a context menu is open with options "Create" and "Refresh...".



Dátové typy

- Je ich možné rozdeliť do niekoľkých skupín.
- Podrobný prehľad dátových typov pre verziu 12 je uvedený na stránke:
<https://www.postgresql.org/docs/12/datatype.html>
- Na hodinách budeme používať hlavne číselné, znakové, dátumovo/časové typy, boolean a typ geometria v ďalšom semestri.

Dátové typy - Číselné

- Smallint, integer (alias int), bigint – celočíselné, líšia sa rozsahom
- Numeric (decimal) – desatinné čísla, kde používateľ môže definovať presnosť pomocou syntaxe *numeric ([precision], [scale])*.
 - Precision znamená celkový počet znakov a scale počet znakov napravo od desatinnej čiarky. Zápis numeric (4,2) umožní do databázy zapísať číslo 27,51, nie však 100,715.

Dátové typy - Číselné

- Real (float4), double precision (float8)- desatinné čísla, kde používateľ nemôže definovať presnosť, pri požiadavke presných výpočtov sa odporúčajú predchádzajúce číselné typy.
- Smallserial, serial, bigserial- slúži na automatické generovanie čísel, najmä pri vytváraní univerzálnych id.

Dátové typy - Znakové

- Character (char) – slúži na uchovávanie reťazcov s presne určeným počtom, preto sa vždy zadáva spolu s dĺžkou reťazca.
 - Napr. char (10) – bude mať vždy 10 znakov. V prípade, že uchováva kratší reťazec, tak do počtu 10 budú doplnené medzery. Pri pokuse vložiť dlhšie reťazce zobrazí chybovú hlášku, s výnimkou, že nejde o medzery. Pri nezadaní dĺžky je prednastavená hodnota 1.
- Character varying (varchar) – to isté, čo predchádzajúci s výnimkou nedopĺňania medzier, aby sa naplnil počet znakov.
- Text – slúži na uchovávanie veľmi dlhých reťazcov (až do 1GB).

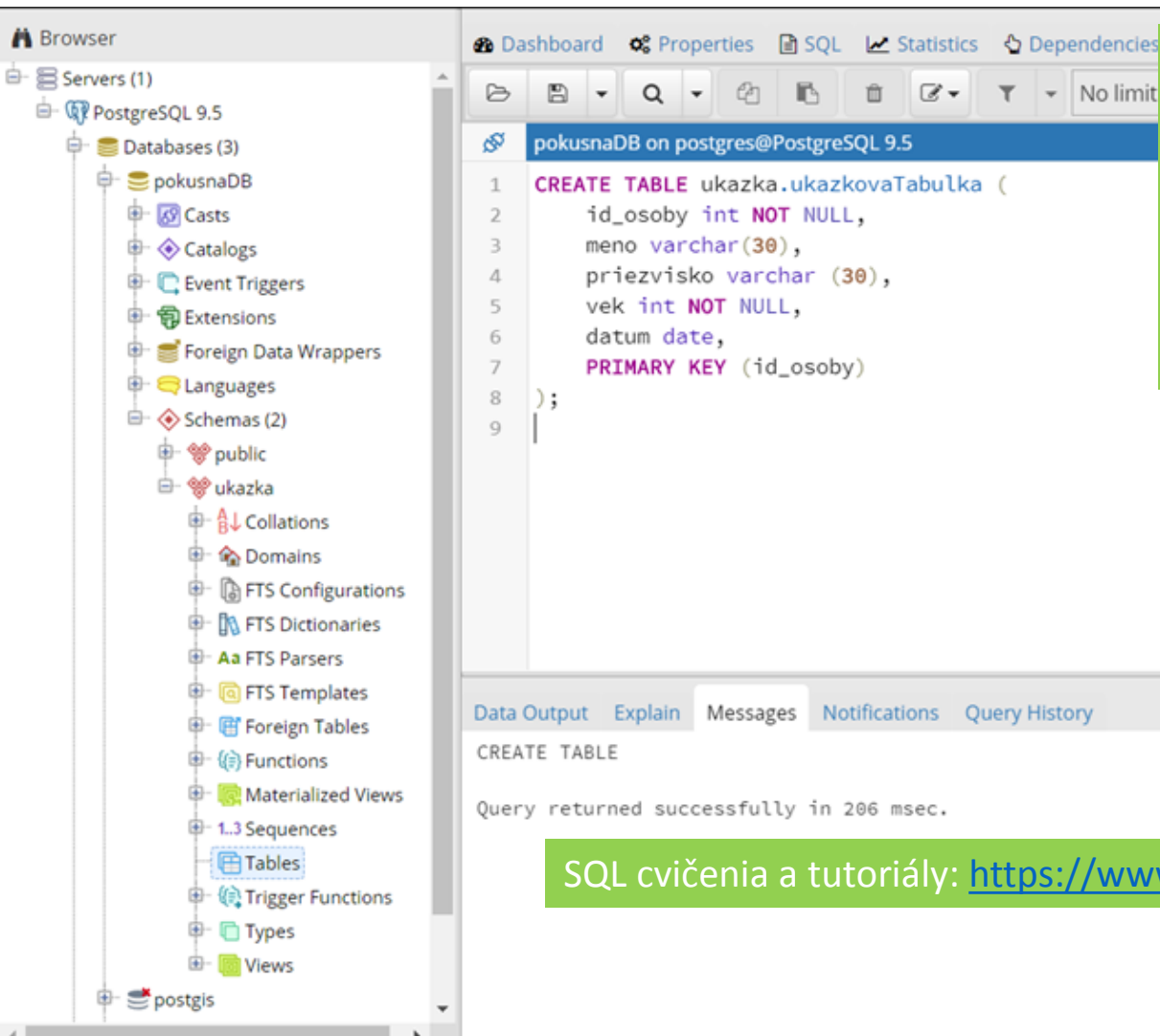
Dátumovo časové typy

- Timestamp [without time zone] – dátum aj čas bez uvedenia zóny
- Timestamp with time zone (timestamp tz)- dátum aj čas s uvedením zóny
- Date – iba dátum
- Time [without time zone] – iba čas bez časovej zóny
- Time with time zone (time tz)- iba čas s časovou zónou
- Interval

Typ Boolean (bool)

- Slúži na uchovávanie pravdivostných hodnôt.
- Môže obsahovať trojicu hodnôt: true, false, unkown (NULL)
- Za TRUE sú považované nasledovné reťazce (pozor na slovo reťazce):
 - True, yes, on, 1, t
- Za FALSE sú považované nasledovné reťazce:
 - False, no, off, 0, f

Obmedzenia v tabuľke



Constraints – obmedzenia

- NOT NULL – stĺpec nesmie obsahovať hodnotu NULL
- UNIQUE – všetky hodnoty v stĺpci budú unikátne
- PRIMARY KEY – jedinečný identifikátor každého riadku v tabuľke
- FOREIGN KEY – jedinečne identifikuje riadok v inej tabuľke
- CHECK – všetky hodnoty v stĺpci spĺňajú nejakú podmienku
- DEFAULT – nastaví predvolenú hodnotu pre stĺpec
- INDEX – rýchle vytváranie a načítavanie údajov z databázy

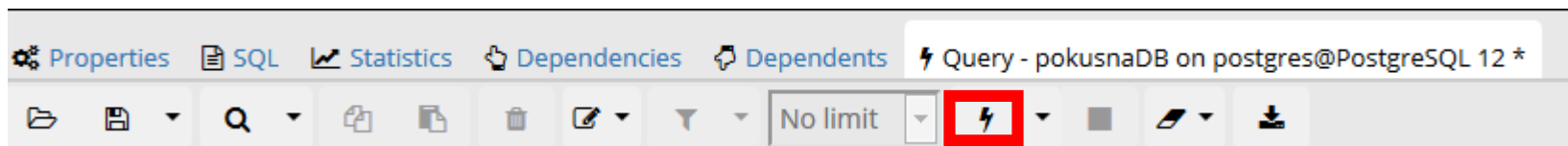
SQL cvičenia a tutoriály: <https://www.w3schools.com/sql/default.asp>



Query returned successfully in 206 msec.

Vytvorenie tabuľky s N stĺpcami

- Využíva sa SQL dopyt
- *CREATE TABLE názov_tabuľky (názov_stĺpca1 dátový typ, názov_stĺpca2 dátový typ, názov_stĺpcaN dátový typ, obmedzenia_tabuľky);*
- Každý SQL dopyt by mal byť ukončený bodkočiarkou ;
- Obsah okna v dopytovacom nástroji sa spúšťa klávesom F5 alebo ikonkou blesku na lište.



Názvy stĺpcov v tabuľke

- Názov stĺpca musí vždy začínať písmenom (v novších verziách je možné použiť dĺžne, mäkčene) alebo podčiarkovníkom _.
- Môžu obsahovať číslce.
- Názov môže obsahovať iba číslce, ak je zadaný ako text v úvodzovkách.
- Nesmie obsahovať interpunkciu.
- Dĺžka názvu stĺpca je obmedzená na 63 znakov, pričom môže byť navýšená zmenou konštanty v zdrojovom kóde.
- Pozor na kľúčové slová.

Ukážka SQL kódu Vytvorenia tabuľky

```
CREATE TABLE ukazka.ukazkovaTabulka (  
    id_osoby int,  
    meno varchar(30),  
    priezvisko varchar(30),  
    vek int,  
    datum date);
```

Dopytovanie tabuľky

- Základná syntax SQL dopytu na dva stĺpce:
- *SELECT stĺpec_tabuľky1, stĺpec_tabuľky2 FROM názov_tabuľky;*
- V prípade dopytu na všetky stĺpce sa miesto názvov stĺpcov využije znak *.
- Pozor na dopytovanie tabuliek z iných schém ako public. V takom prípade je potrebné v dopyte určiť aj schému, v ktorej sa daná tabuľka nachádza:
 - *SELECT stĺpec_tabuľky1, stĺpec_tabuľky2 FROM názov_schémy.názov_tabuľky*



Dopytovanie tabuľky

- Poradie stĺpcov
- Veľké a malé písmená.
- Možnosť spúšťania sekvencie SQL dopytov.
- V pravom dolnom rohu info o úspešnom/ neúspešnom vykonaní SQL dopytov a jeho trvaní.

Import údajov do tabuľky

- Syntax:

- `INSERT INTO názov_schémy.názov_tabuľky VALUES` (hodnoty pre prvý stĺpec, pre druhý stĺpec atď.), (hodnoty pre prvý stĺpec, pre druhý stĺpec atď.) Hodnoty musia byť oddelené čiarkou, rovnako aj atribúty objektov.

```
INSERT INTO ukazka.ukazkovaTabulka VALUES  
(2, 'Filoména', 'Hravá', 97, '23.5.2020'),  
(3, 'Arnold', 'Dravý', 101, '24.5.1989');
```

Pozor na dátum, možno bude potrebný anglický zápis.

	id_osoby integer	meno character varying (30)	priezvisko character varying (30)	vek integer	datum date
1	1	Lojzo	Dokrútený	97	1980-07-15
2	2	Filoména	Hravá	97	2020-05-23
3	3	Arnold	Dravý	101	1989-05-24

Import údajov do tabuľky

- Syntax pre import iba do špecifických stĺpcov:
- INSERT INTO názov_schémy.názov_tabuľky (špecifický_stĺpec1, špecifický_stĺpec2) VALUES (hodnota pre špecifický_stĺpec1, hodnota pre špecifický_stĺpec2)

```
INSERT INTO ukazka.ukazkovaTabulka (id_osoby, priezvisko)
VALUES
(4, 'Zbojník'),
(5, 'Veselý');
```

1	1	Lojzo	Dokrútený	97	1980-07-15
2	2	Filoména	Hravá	97	2020-05-23
3	3	Arnold	Dravý	101	1989-05-24
4	4	[null]	Zbojník	[null]	[null]
5	5	[null]	Veselý	[null]	[null]

Zmeny v už vytvorenej tabuľke

- Používa sa príkaz ALTER TABLE.
- Slúži na pridanie, zmazanie stĺpcov z tabuľky, zmenu dátového typu v stĺpci tabuľky.
- ALTER TABLE sa používa aj pre zmenu obmedzení.
- Ukážka syntaxe pre pridanie stĺpca:

ALTER TABLE ukazka.ukazkovaTabulka

ADD nový_stlpec dátový_typ [,

ADD nový_stlpec dátový_typ ...];

Komentáre v SQL

- Pre jednoriadkový komentár používame dvojicu znakov --
- Pre viacriadkový komentár používame na začiatku dvojicu znakov /* a na konci dvojicu znakov */.
- Po zakomentovaní určitej časti kódu dôjde aj k farebnej zmene komentovaného textu.

```
--SELECT * FROM ukazka.ukazkovaTabulka;  
  
/*INSERT INTO ukazka.ukazkovaTabulka (id_osoby, priezvisko)  
VALUES  
(4, 'Zbojník'),  
(5, 'Veselý');*/
```

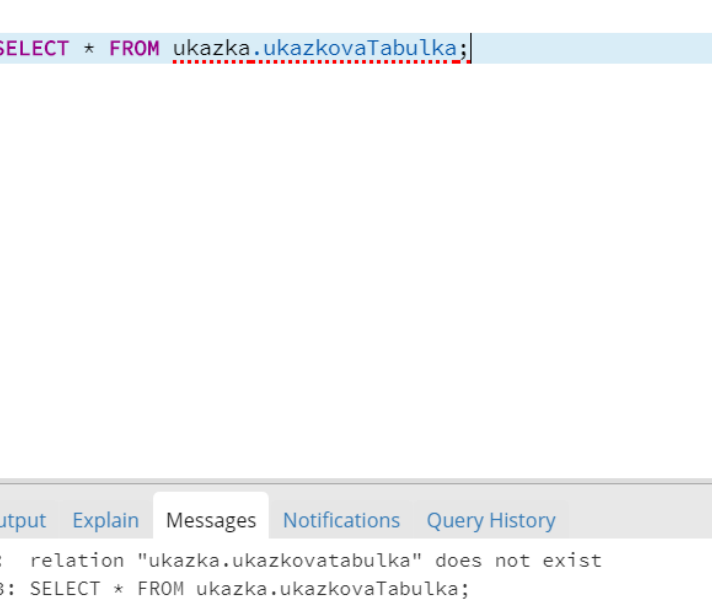
Vymazanie údajov z tabuľky a vymazanie tabuľky

- Pre vymazanie všetkých údajov z tabuľky sa používa príkaz so syntaxou:
 - `DELETE FROM názov_schémy.názov_tabuľky;`
- Pre vymazanie tabuľky sa používa príkaz so syntaxou:
 - `DROP TABLE názov_schémy.názov_tabuľky;`

Import údajov do/z tabuľky z/do csv súboru

- Syntax exportu tabuľky do csv súboru:
 - COPY schéma.tabuľka TO 'cesta k súboru na disku' DELIMITER ',' CSV HEADER;
 - Ukážka:
 - COPY ukazka.ukazkovaTabulka TO 'F:\ukazka.csv' DELIMITER ',' CSV HEADER;
 - Znak ' sa zadáva na SVK klávesnici ako pravý Alt + P, na ENG §
- Syntax importu do tabuľky z csv súboru:
 - COPY schéma.tabuľka FROM 'cesta k súboru na disku' DELIMITER ',' CSV HEADER;

- # hlášky s porozumením.



```
pokusnaDB on postgres@PostgreSQL 9.5
1 DROP TABLE ukazka.ukazkovaTabulka;
2
3 SELECT * FROM ukazka.ukazkovaTabulka;
```

Data Output Explain Messages Notifications Query History

ERROR: relation "ukazka.ukazkovatabulka" does not exist
LINE 3: SELECT * FROM ukazka.ukazkovaTabulka;
^

SQL state: 42P01
Character: 51



Vaše otázky?

Otázky

- Čo sú to kľúčové slová jazyka SQL? Vymenujte aspoň 2 príklady.
- Aké dátové typy poznáte a kde by ste hľadali ďalšie?
- Aká je syntax pre vytvorenie tabuľky?
- Aký typ dopytu použijete pre vloženie nového záznamu do tabuľky?
- Aký je rozdiel medzi dopytom *DELETE* a *DROP*?
- Na čo môže slúžiť príkaz *COPY*.
- Je rozdiel medzi zápisom *2022* a *'2022'*?

Úlohy

- Doplníte chýbajúci údaj, aby ste získali všetky stĺpce z tabuľky “Customers”:
 - _____ * _____ Customers;
- Doplníte chýbajúce údaje, aby ste vybrali stĺpec “City” z tabuľky “Customers”:
 - _____ _____ _____ _____;
- Vymažte všetky údaje z tabuľky “studenti”, ktorá sa nachádza v schéme “zik”:
 - _____ _____ _____ _____;
- Následne vymažte tabuľku “studenti”:
 - _____ _____ _____ _____;

- Majme tabuľku obce (id, nazov, okres, pocet_obyv) v schéme public.
- Čo vykonajú nasledujúce dopyty?
- `SELECT id, názov FROM obce;`
- `SELECT *, nazov FROM obce;`
- `SELECT 5+4 FROM obce;`
- `InSeRt intO obce VaLueS (4, 'Kokšov Bakša', 'Košice Okolie' , 457);`
- `DELETE obce;`

Praktické úlohy

- Vo zvolenej databáze vytvorte schému *ulohy*.
- V schéme *ulohy* vytvorte tabuľku *spoluziaci*, ktorá bude mať 4 stĺpce (id, meno, priezvisko, vyska), dátové typy vyberte zmysluplne podľa názvu.
- Vložte do tabuľky 3 nové, rôzne záznamy s hodnotami pre všetky stĺpce.
- Vložte do tabuľky 1 nový záznam s id a menom. Ostatné atribúty budú bez hodnoty.
- Vytvorte dopyt, ktorým vo výsledku získate priezvisko a výšku vydelenú číslom 3.
- Vytvorte na disku csv súbor, ktorý bude slúžiť ako záloha vašej tabuľky.
- Odstráňte obsah vašej tabuľky a skontrolujte, či je tabuľka naozaj bez záznamu.
- Načítajte obsah súboru do tabuľky a skontrolujte, či obsahuje 4 záznamy.
- Vymažte tabuľku.