Geografická báza údajov 1
Cvičenie 3

Vladimír Pelech

pelech2@uniba.sk, G-23

PRÍRODOVEDECKÁ FAKULTA
Univerzita Komenského
v Bratislave

Cvičenie 3

- Náplň:
  - LIMIT
  - Zoradenie pomocou ORDER BY
  - Klauzula WHERE
  - DISTINCT a DISTINCT ON
  - Vytvorenie pohľadu - CREATE VIEW
  - Agregačné funkcie a zoskupenie pomocou GROUP BY
  - Záloha databázy – Backup
  - Cvičné úlohy

IMPORT údajov

- Zapnite si pgAdmin a zadajte heslo pre vstup do databázy.
- Vytvorte schému *cviko3.*
- Zo stránky predmetu si stiahnite *.csv* súbor s dátami.
- Vo vytvorenej schéme *cviko3* vytvorte tabuľku *obyvatelia* so stĺpcami (okres, obec, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017), prvé dva stĺpce sú reťazce, ostatné sú celé číslo. Pri reťazcoch dajte pozor na dostatočnú dĺžku. Pri neuvedení sa použije prednastavená hodnota 1, čo bude málo. V názvoch je možné použiť číslo, ak bude zadané ako reťazec ("2017").
- Importujte *.csv* súbor do vami vytvorenej tabuľky *obyvatelia*.
- Cesta k súboru znamená celá cesta aj s názvom súboru a jeho koncovkou, vyhnite sa dĺžňom, mäkčeňom a podobne.
- Pozor na právo čítať priečinky! Je možné, že COPY skončí s chybou.

Výber počtu objektov

- Jednoduchý výber bol prezentovaný na poslednom cvičení:
  - *SELECT * FROM názov_schémy.názov_tabuľky;*
- Výber objektov je však možné aj špecifikovať rôznymi podmienkami, čo si ukážeme neskôr, alebo kľúčovým slovom LIMIT a číslom na konci SQL príkazu.
  - *SELECT * FROM názov_schémy.názov_tabuľky LIMIT 10;*
- Uvedenie kľúčového slova LIMIT spolu s číslom v SQL dopyte vráti maximálne limitovaný počet objektov, ak sa ich v danej tabuľke nachádza dostatočný počet.

Výber bez a s obmedzením LIMIT

pokusnaDB on postgres@PostgreSQL 12

```
1  SELECT * FROM cviko3.obyvatelia;
```

Data Output | Explain | Messages | Notifications | Query Histo

	okres character varying	obec character varying	2009 integer
1	Okres Bratislava I	Bratislava - mestská ...	433
2	Okres Bratislava II	Bratislava - m. č. Pod...	270
3	Okres Bratislava II	Bratislava - mestská ...	849
4	Okres Bratislava II	Bratislava - mestská ...	248
5	Okres Bratislava III	Bratislava - mestská ...	490
6	Okres Bratislava III	Bratislava - mestská ...	197
7	Okres Bratislava III	Bratislava - mestská ...	64
8	Okres Bratislava IV	Bratislava - m. č. Dev...	193

pokusnaDB on postgres@PostgreSQL 12

```
1  SELECT * FROM cviko3.obyvatelia LIMIT 5;
```

Data Output | Explain | Messages | Notifications | Query Hist

	okres character varying	obec character varying	2009 integer	20 int
1	Okres Bratislava I	Bratislava - mestská ...	433	
2	Okres Bratislava II	Bratislava - m. č. Pod...	270	
3	Okres Bratislava II	Bratislava - mestská ...	849	
4	Okres Bratislava II	Bratislava - mestská ...	248	
5	Okres Bratislava III	Bratislava - mestská ...	490	

SELECT * FROM cviko3.obyvatelia;

Successfully run. Total query runtime: 172 msec. 2934 rows affected.

SELECT * FROM cviko3.obyvatelia LIMIT 5;

Successfully run. Total query runtime: 114 msec. 5 rows affected.

Zoradenie

- Vykonáva sa pri zadaní kľúčových slov ORDER BY.
- Prednastavené je zoraďovanie od najmenšej po najvyššiu hodnotu.
- Nastavenie zoraďovania je možné pomocou poradia stĺpcov a pomocou kľúčových slov:
  - ASC ->ASCENDING (vzostupne - od najmenšieho po najväčšie)
  - DESC -> DESCENDING (zostupne – od najväčšieho po najmenšie)
- Pri viacerých stĺpcoch dôjde najprv k zoradeniu podľa prvého uvedeného, následne druhého, atď...
- Syntax:
  - SELECT * FROM tabuľka ORDER BY stĺpec1 ASC alebo DESC [, stlpec2 ASC alebo DESC, atď.];

Zoradenie

pokusnaDB on postgres@PostgreSQL 12

```
1  SELECT * FROM cviko3.obyvatelia ORDER BY obec;
```

Data Output | Explain | Messages | Notifications | Query History

	okres character varying	obec character varying	2009 integer	2010 integer
1	Okres Lučenec	Ábelová	2	1
2	Okres Rimavská Sob...	Abovce	5	1
3	Okres Galanta	Abrahám	7	1
4	Okres Bardejov	Abrahámovce, okres...	4	
5	Okres Kežmarok	Abrahámovce, okres...	4	
6	Okres Turčianske Te...	Abramová	2	
7	Okres Prešov	Abranovce	11	1
8	Okres Trenčín	Adamovské Kochan...	7	
9	Okres Humenné	Adidovce	4	
10	Okres Nitra	Alekšince	18	1
11	Okres Nové Zámky	Andovce	9	1
12	Okres Bardejov	Andrejová	3	
13	Okres Topoľčany	Ardanovce	1	
14	Okres Rožňava	Ardovo	1	
15	Okres Spišská Nová ...	Arnutovce	16	2
16	Okres Nitra	Báb	12	
17	Okres Vranov nad T...	Babie	1	
18	Okres Námestovo	Babín	18	1

Klauzula WHERE

- Syntax: *SELECT * FROM schéma.tabuľka WHERE výraz;*
- Používa sa na filtrovanie údajov na základe výrazov(podmienok).
- Využíva logické operátory =, >, <, >=, <=, <> (nerovný), != (tiež nerovný), AND, OR, BETWEEN, IN, NOT
- Operátor LIKE sa používa pri textových dátových typoch.
  - Špeciálny symbol _ nahrádza v textovom reťazci práve 1 znak.
  - Špeciálny symbol % nahrádza v textovom reťazci 1 a viac znakov.

Výber obcí okresu Prievidza a ich zoradenie v abecednom poradí.

```
1  SELECT * FROM cviko3.obyvatelia WHERE okres = 'Okres Prievidza' ORDER BY obec;
```

Data Output | Explain | Messages | Notifications | Query History

	okres character varying	obec character varying	2009 integer	2010 integer	2011 integer	2012 integer	2013 integer	2014 integer	2015 integer	2016 integer	2017 integer
1	Okres Prievidza	Bojnice	40	58	39	30	37	30	35	43	29
2	Okres Prievidza	Bystričany	10	16	18	10	10	11	20	12	18
3	Okres Prievidza	Cigeľ	11	17	14	10	11	8	8	14	14
4	Okres Prievidza	Čavoj	4	1	5	7	6	3	2	5	3
5	Okres Prievidza	Čereňany	21	17	10	13	11	12	8	18	25
6	Okres Prievidza	Diviacka Nová Ves	17	18	17	17	18	14	14	14	11
7	Okres Prievidza	Diviaky nad Nitricou	20	17	12	13	19	19	17	17	12
8	Okres Prievidza	Dlžín	2	0	2	2	0	0	1	1	2
9	Okres Prievidza	Dolné Vestenice	22	19	31	22	22	26	20	20	25
10	Okres Prievidza	Handlová	204	165	177	147	150	119	146	145	148
11	Okres Prievidza	Horná Ves, okres Pri...	17	8	20	8	9	15	9	9	13
12	Okres Prievidza	Horné Vestenice	4	5	9	7	3	2	3	4	3
13	Okres Prievidza	Chrenovec-Brusno	15	13	15	14	11	18	5	11	8
14	Okres Prievidza	Chvojnica, okres Prie...	3	4	3	5	0	4	1	2	3

Výber obcí okresu Prievidza, v ktorých sa v roku 2017 narodilo viac ako 20 detí a ich zoradenie od najvyššej hodnoty.

```
1  SELECT * FROM cviko3.obyvatelia WHERE okres = 'Okres Prievidza' AND "2017">20 ORDER BY "2017" DESC;
```

Data Output | Explain | Messages | Notifications | Query History

	okres character varying	obec character varying	2009 integer	2010 integer	2011 integer	2012 integer	2013 integer	2014 integer	2015 integer	2016 integer	2017 integer
1	Okres Prievidza	Prievidza	434	479	446	381	368	390	349	448	403
2	Okres Prievidza	Handlová	204	165	177	147	150	119	146	145	148
3	Okres Prievidza	Kanianka	31	48	28	28	36	36	34	34	40
4	Okres Prievidza	Nitrianske Pravno	40	33	26	24	35	34	24	30	35
5	Okres Prievidza	Bojnice	40	58	39	30	37	30	35	43	29
6	Okres Prievidza	Lehota pod Vtáčnikom	46	37	30	56	37	42	36	42	29
7	Okres Prievidza	Dolné Vestenice	22	19	31	22	22	26	20	20	25
8	Okres Prievidza	Čereňany	21	17	10	13	11	12	8	18	25
9	Okres Prievidza	Nováky	39	46	41	32	32	36	42	36	25

Výber obcí okresov Prievidza a Partizánske a ich zoradenie podľa abecedy.

pokusnaDB on postgres@PostgreSQL 12

```
1  SELECT * FROM cviko3.obyvatelia WHERE okres = 'Okres Prievidza' OR okres = 'Okres Partizánske' ORDER BY obec;
```

Data Output | Explain | Messages | Notifications | Query History

	okres character varying	obec character varying	2009 integer	2010 integer	2011 integer	2012 integer	2013 integer	2014 integer	2015 integer	2016 integer	2017 integer
1	Okres Prievidza	Bojnice	40	58	39	30	37	30	35	43	29
2	Okres Partizánske	Bošany	48	41	33	29	43	39	34	31	47
3	Okres Partizánske	Brodzany	6	5	5	5	7	3	7	6	6
4	Okres Prievidza	Bystričany	10	16	18	10	10	11	20	12	18
5	Okres Prievidza	Cigeľ	11	17	14	10	11	8	8	14	14
6	Okres Prievidza	Čavoj	4	1	5	7	6	3	2	5	3
7	Okres Prievidza	Čereňany	21	17	10	13	11	12	8	18	25
8	Okres Prievidza	Diviacka Nová Ves	17	18	17	17	18	14	14	14	11
9	Okres Prievidza	Diviaky nad Nitricou	20	17	12	13	19	19	17	17	12
10	Okres Prievidza	Dlžín	2	0	2	2	0	0	1	1	2
11	Okres Prievidza	Dolné Vestenice	22	19	31	22	22	26	20	20	25
12	Okres Prievidza	Handlová	204	165	177	147	150	119	146	145	148
13	Okres Prievidza	Horná Ves, okres Pri...	17	8	20	8	9	15	9	9	13
14	Okres Prievidza	Horné Vestenice	4	5	9	7	3	2	3	4	3
15	Okres Partizánske	Hradište, okres Parti...	18	17	9	9	14	17	5	10	6
16	Okres Prievidza	Chrenovec-Brusno	15	13	15	14	11	18	5	11	8

Výber obcí okresu Prievidza, ktoré sa začínajú na písmeno Č a ich zoradenie podľa abecedy zostupne.

pokusnaDB on postgres@PostgreSQL 12

```
1  SELECT * FROM cviko3.obyvatelia WHERE okres = 'Okres Prievidza' AND obec LIKE 'Č%' ORDER BY obec DESC;
```

Data Output | Explain | Messages | Notifications | Query History

	okres character varying	obec character varying	2009 integer	2010 integer	2011 integer	2012 integer	2013 integer	2014 integer	2015 integer	2016 integer	2017 integer
1	Okres Prievidza	Čereňany	21	17	10	13	11	12	8	18	25
2	Okres Prievidza	Čavoj	4	1	5	7	6	3	2	5	3

Výber obcí okresu Prievidza, ktoré sa začínajú na písmeno Č alebo v názve obce sa nachádza písmeno Č a ich zoradenie podľa abecedy zostupne.

pokusnaDB on postgres@PostgreSQL 12

```
1 SELECT * FROM cviko3.obyvatelia WHERE okres = 'Okres Prievidza' AND obec LIKE '%Č%' ORDER BY obec DESC;
```

Data Output | Explain | Messages | Notifications | Query History

	okres character varying	obec character varying	2009 integer	2010 integer	2011 integer	2012 integer	2013 integer	2014 integer	2015 integer	2016 integer	2017 integer
1	Okres Prievidza	Veľká Čausa	4	3	10	6	5	2	5	6	7
2	Okres Prievidza	Malá Čausa	4	6	7	5	4	6	8	7	5
3	Okres Prievidza	Čereňany	21	17	10	13	11	12	8	18	25
4	Okres Prievidza	Čavoj	4	1	5	7	6	3	2	5	3

Pokúste sa uvedený dopyt pozmeniť tak, aby boli zobrazené všetky obce, ktoré v názve obsahujú malé č alebo veľké Č. (11 obcí)

Napíšte a použite dopyt, ktorým zistíte počet obcí na Slovensku s trojpísmenovým názvom.

Výber obcí okresu Prievidza, v ktorých sa v roku 2017 narodilo od 25 do 40 detí a ich zoradenie podľa počtu narodených detí zostupne. Využije sa na to operátor BETWEEN, ktorého syntax je:

BETWEEN najnižšia hodnota AND najvyššia hodnota

pokusnaDB on postgres@PostgreSQL 12

```
1  SELECT * FROM cviko3.obyvatelia WHERE okres = 'Okres Prievidza' AND "2017" BETWEEN 25 AND 40 ORDER BY "2017" DESC;
```

Data Output | Explain | Messages | Notifications | Query History

	okres character varying	obec character varying	2009 integer	2010 integer	2011 integer	2012 integer	2013 integer	2014 integer	2015 integer	2016 integer	2017 integer
1	Okres Prievidza	Kanianka	31	48	28	28	36	36	34	34	40
2	Okres Prievidza	Nitrianske Pravno	40	33	26	24	35	34	24	30	35
3	Okres Prievidza	Bojnice	40	58	39	30	37	30	35	43	29
4	Okres Prievidza	Lehota pod Vtáčnikom	46	37	30	56	37	42	36	42	29
5	Okres Prievidza	Čereňany	21	17	10	13	11	12	8	18	25
6	Okres Prievidza	Dolné Vestenice	22	19	31	22	22	26	20	20	25
7	Okres Prievidza	Nováky	39	46	41	32	32	36	42	36	25

Hodnota NULL

- Hodnota NULL znamená, že údaj pre daný objekt nie je dostupný, je bez hodnoty, bez záznamu.
- Nemýliť si hodnotu NULL s hodnotou 0 pri číselných stĺpcoch.
- Pri dopytovaní sa na objekty bez hodnoty (NULL) sa používa syntax:
  - Názov_stĺpca IS NULL (Pri boolean je možné NULL nahradiť UNKNOWN)
- Opakom je:
  - Názov_stĺpca IS NOT NULL

Výber obcí, ktoré nemajú v roku 2017 záznam o narodení.

pokusnaDB on postgres@PostgreSQL 12

```
1  SELECT * FROM cviko3.obyvatelia WHERE "2017" IS NULL;
```

Data Output | Explain | Messages | Notifications | Query History

	okres character varying	obec character varying	2009 integer	2010 integer	2011 integer	2012 integer	2013 integer	2014 integer	2015 integer	2016 integer	2017 integer
1	Okres Horné	Horná Dolná	150	65	48	23	15	7	1	[null]	[null]
2	Okres Dolné	Dolná Dolná	15	54	7	35	5	0	[null]	[null]	[null]

Dopyt upravte tak, aby zobrazoval obce bez záznamu o narodení v roku 2016, ale so záznamom v roku 2015.

Výber obcí, ktoré nemajú v roku 2017 záznam o narodení alebo sa narodilo 0 detí.

pokusnaDB on postgres@PostgreSQL 12

```
SELECT * FROM cviko3.obyvatelia WHERE "2017" IS NULL OR "2017" = 0;
```

Data Output | Explain | Messages | Notifications | Query History

	okres character varying	obec character varying	2009 integer	2010 integer	2011 integer	2012 integer	2013 integer	2014 integer	2015 integer	2016 integer	2017 integer
1	Okres Senec	Vlky	2	4	4	2	3	5	1	4	0
2	Okres Senec	Nový Svet	0	1	1	1	1	1	1	1	0
3	Okres Dunajská Stre...	Bodíky	3	2	0	2	2	2	0	1	0
4	Okres Hlohovec	Tekoľany	1	0	0	1	0	1	0	1	0
5	Okres Skalica	Koválovec	1	1	2	4	0	2	0	1	0
6	Okres Bánovce nad ...	Omastiná	0	0	1	0	0	0	0	0	0
7	Okres Bánovce nad ...	Slatinka nad Bebravou	0	1	1	5	1	1	2	0	0
8	Okres Bánovce nad ...	Šípkov	1	0	0	0	0	0	1	1	0
9	Okres Bánovce nad ...	Trebichava	0	0	0	1	0	0	1	1	0
10	Okres Bánovce nad ...	Uhrovské Podhradie	0	1	0	0	0	0	0	0	0
11	Okres Bánovce nad ...	Cimenná	3	1	2	1	0	1	0	0	0

Rozdiel medzi DISTINCT a DISTINCT ON

- *DISTINCT* vráti jedinečné hodnoty, DISTINCT ON prvý jedinečný záznam po zoradení
- *SELECT DISTINCT okres FROM cviko3.obyvatelia;*
  - Vyberie jedinečné záznamy zo stĺpca *okres* a iba daného stĺpca.
- *SELECT DISTINCT okres, obec FROM cviko3.obyvatelia;*
  - Vyberie jedinečné záznamy z kombinácie stĺpcov *okres* a *obec* a iba daných stĺpcov.
- *SELECT DISTINCT ON (okres) * FROM cviko3.obyvatelia;*
  - Vyberie všetky stĺpce pre prvý záznam zo stĺpca *okres* po zoradení.
- *SELECT DISTINCT ON (okres) okres, obec FROM cviko3.obyvatelia;*
  - Vyberie stĺpce *okres*, *obec* a NEBUDE vypisovať duplicitné hodnoty v stĺpci okres a BUDE vypisovať IBA prvý záznam zo stĺpca obec

Výber okresov bez toho, aby boli vo výsledku duplicitné hodnoty.

pokusnaDB on postgres@PostgreSQL 12

```
SELECT DISTINCT okres FROM cviko3.obyvatelia ORDER BY okres ASC;
```

Data Output | Explain | Messages | Notifications | Query History

	okres character varying
1	Okres Bánovce nad Bebravou
2	Okres Banská Bystrica
3	Okres Banská Štiavnica
4	Okres Bardejov
5	Okres Bratislava I
6	Okres Bratislava II
7	Okres Bratislava III
8	Okres Bratislava IV
9	Okres Bratislava V
10	Okres Brezno
11	Okres Bytča
12	Okres Čadca
13	Okres Detva
14	Okres Dolné
15	Okres Dolný Kubín
16	Okres Dunajská Streda
17	Okres Galanta

Vo výsledku dostaneme 81 jedinečných záznamov. (Pridané okresy Horné a Dolné)

Upravte dopyt
SELECT DISTINCT ON (okres) okres, obec FROM cviko3.obyvatelia;
aby pre každý okres zobrazoval obec, ktorá je prvá podľa abecedy.

	okres character varying	obec character varying
1	Okres Bánovce nad Bebravou	Bánovce nad Bebrav...
2	Okres Banská Bystrica	Badín
3	Okres Banská Štiavnica	Baďan
4	Okres Bardejov	Abrahámovce, okres...
5	Okres Bratislava I	Bratislava - mestská ...
6	Okres Bratislava II	Bratislava - m. č. Pod...
7	Okres Bratislava III	Bratislava - mestská ...
8	Okres Bratislava IV	Bratislava - m. č. Dev...
9	Okres Bratislava V	Bratislava - mestská ...
10	Okres Brezno	Bacúch
11	Okres Bytča	Bytča
12	Okres Čadca	Čadca
13	Okres Detva	Detva

Časť požadovaného výsledku.

Vytvorenie pohľadu - view

- Funguje ako tabuľka, avšak iba na báze vytvoreného a uloženého dopytu.
- Vzniká ako odkaz na existujúcu tabuľku, resp. tabuľky.
- Aktualizácia zdrojovej tabuľky sa prejaví aktualizáciou pohľadu.
- Je ho možné dopytovať.

Vytvorenie pohľadu - view

- Je ho možné vytvoriť s pomocou výberového dopytu.

pokusnaDB on postgres@PostgreSQL 12

```
1  CREATE VIEW cviko3.nahlad AS
2  SELECT * FROM cviko3.obyvatelia WHERE obec LIKE '%Č%';
3
```

Data Output | Explain | Messages | Notifications | Query History

```
CREATE VIEW

Query returned successfully in 104 msec.
```

Vytvorenie pohľadu - view

- Dopyt na pohľad nám vráti 98 záznamov.
  - *SELECT * FROM cviko3.nahlad;*
- Vložte do tabuľky *obyvatelia* nový záznam o obci Čukotka v okrese Paríž.
- Koľko záznamov je po vložení Čukotky vo view *nahlad*?
- Napriek tomu, že sa pohľad nezmenil, zmenila sa pôvodná tabuľka, čo sa hneď prejavilo v príslušnom pohľade.

Agregačné funkcie

- MIN – vráti najmenšiu hodnotu zo vstupných hodnôt
- MAX – vráti najvyššiu hodnotu zo vstupných hodnôt
- SUM – vráti súčet hodnôt zo vstupných hodnôt
- COUNT – vráti počet riadkov
- AVG – vráti priemernú hodnotu zo vstupných hodnôt
- https://www.postgresql.org/docs/12/functions-aggregate.html

Agregačné funkcie - ukážka

SELECT MIN(“2017“) FROM cviko3.obyvatelia;	0
SELECT MAX(“2017“) FROM cviko3.obyvatelia;	1437
SELECT SUM(“2017“) FROM cviko3.obyvatelia;	58128
SELECT ROUND(AVG(“2017“),2) FROM cviko3.obyvatelia;	19.83
SELECT COUNT(“2017“) FROM cviko3.obyvatelia;	2932

Zoskupenie pomocou GROUP BY

- V tabuľkách sa častokrát vyskytuje viacero záznamov s rovnakou hodnotou atribútu.
- Použitie agregačnej funkcie spolu s klauzulou GROUP BY umožňuje získať hodnoty zoskupené na základe rovnakej hodnoty v určitom atribúte.
- Typický príklad je zoskupenie obcí po okresoch.
- Platí pravidlo o povinnom uvedení stĺpca v časti SELECT pri využití GROUP BY.

Zoskupenie pomocou GROUP BY

- Syntax:
  - SELECT zvolený_stĺpec, [agregačná funkcia aplikovaná na iný stĺpec] FROM schéma.tabuľka GROUP BY zvolený stĺpec;

Využitie dopytu v dopyte

- Syntax jazyka SQL umožňuje v dopyte využiť iný dopyt aj na inú tabuľku:

```
SELECT * FROM (
SELECT * FROM schéma1.tabuľka1 WHERE podmienka)
WHERE
hodnota = (SELECT MIN(zvolený_stĺpec) FROM schéma2.tabuľka2);
```

Záloha Databázy (BACKUP)

Kliknúť pravým tlačidlom myši na databázu.

Zvoliť možnosť Backup.

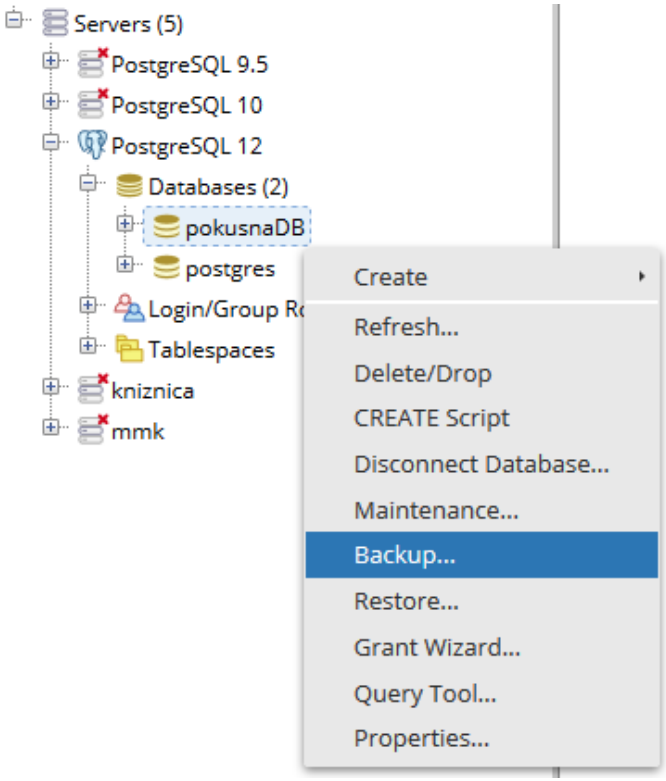
Vo *FILENAME* kliknúť na 3 bodky.

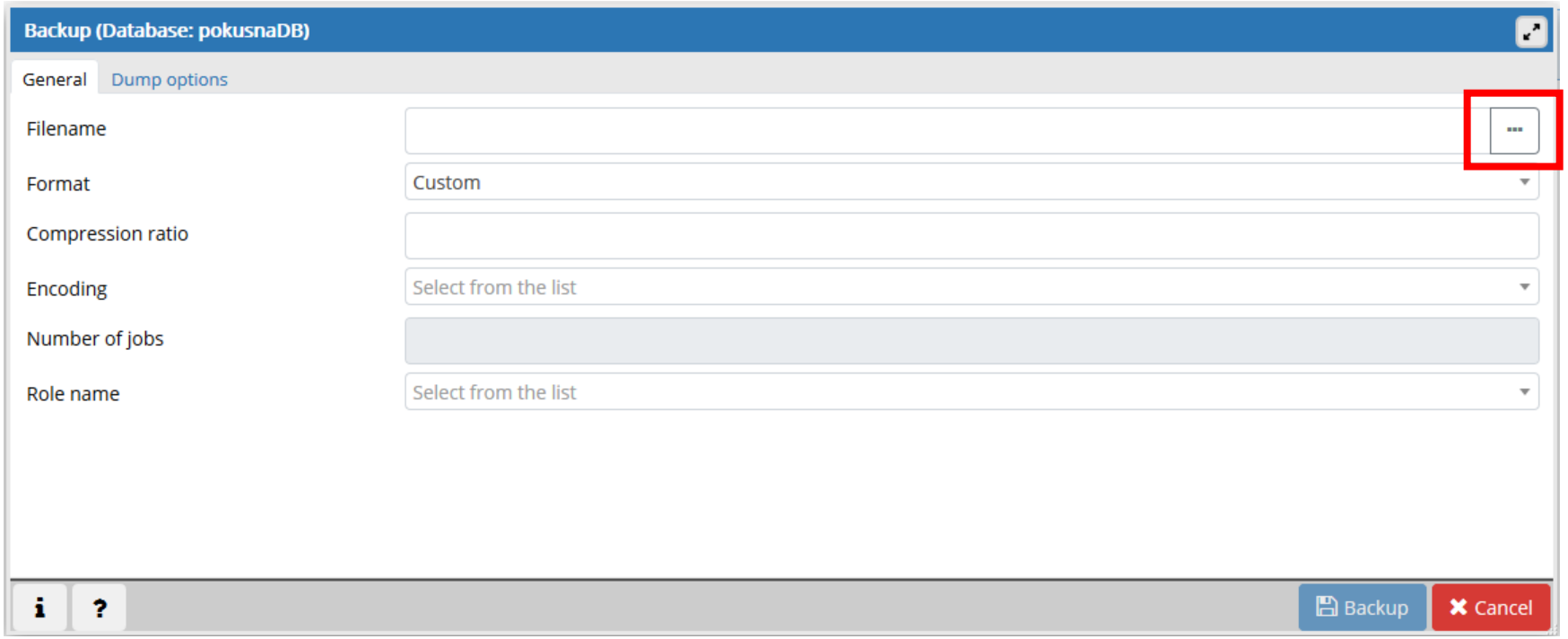
Zvoliť cestu a názov zálohy a jej formát vpravo dole.

Kliknúť na *Select*.

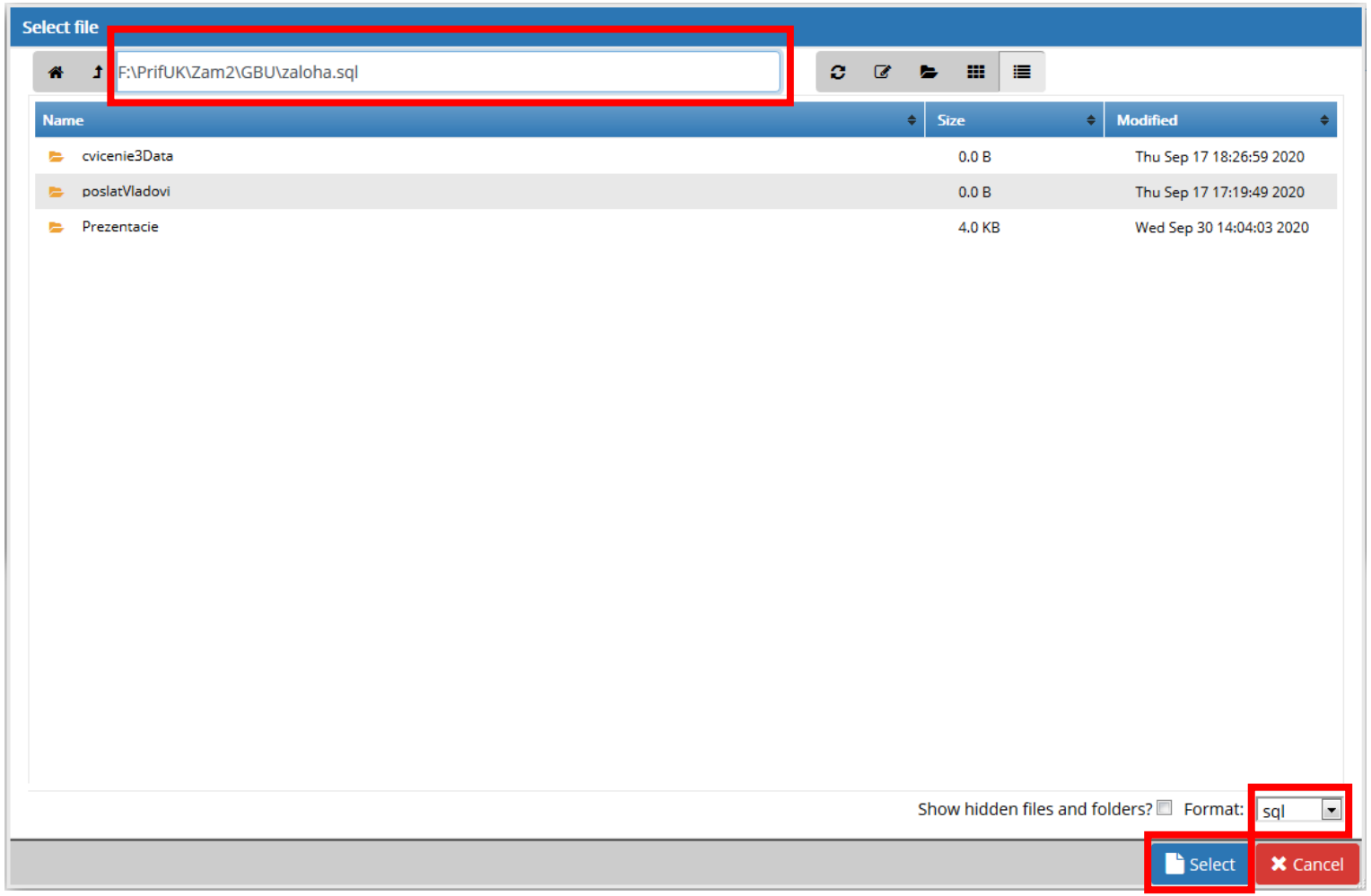
Kliknúť na *Backup.*

Záloha Databázy (BACKUP)

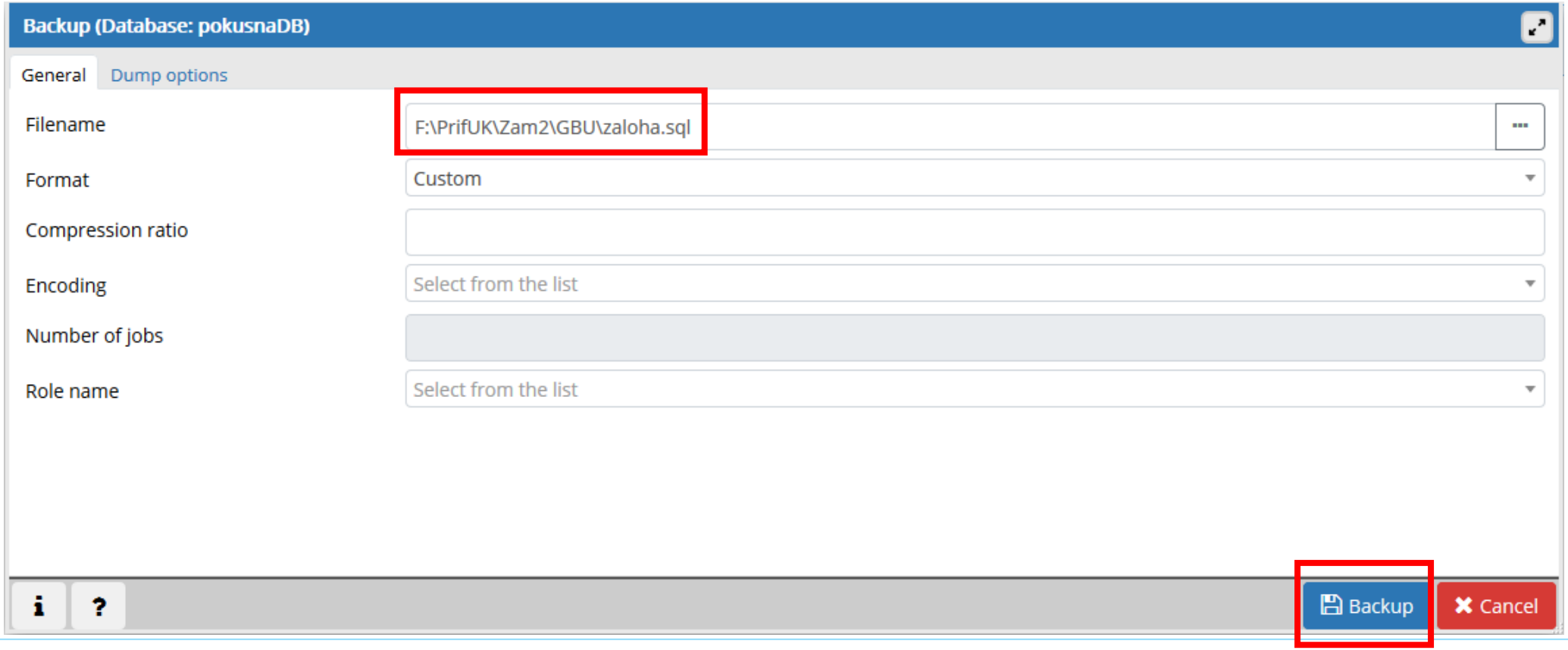




Záloha Databázy (BACKUP)



Záloha Databázy (BACKUP)



Vaše otázky?

Otázky

- Načo slúži kľúčové slovo LIMIT?
- Ako by ste vykonali zoradenie vo výbere?
- Načo slúži klauzula WHERE?
- Načo slúžia logické spojky AND a OR? A aký je medzi nimi rozdiel?
- Čo robí DISTINCT ON naviac v porovnaní s DISTINCT?
- Aké agregačné funkcie poznáte?
- Vlastnými slovami popíšte princíp GROUP BY.

Úlohy

- Doplňte chýbajúci údaj, aby ste získali všetky stĺpce z tabuľky “Customers“ zoradení podľa stĺpca name zostupne:
  - _____ * ___ Customers ___ ___ ___ ___;
- Doplňte chýbajúce údaje, aby ste vybrali stĺpec “City“ z tabuľky “Customers“, pričom chceme iba také záznamy, kde názov stĺpca začína na písmeno A a končí na pímeno O :
  - SELECT city FROM customers ___ ___ _____ ___;
- Určte jedinečné hodnoty zo stĺpca “okres“ v tabuľke “obce“.
  - _____ ___ _____ _____ _____;
- Určte maximum v stĺpci “pocet“ v tabuľke “obce“ :
  - ___ ____ ____ _____ ____;
- Určte počet záznamov po okresoch v tabuľke “obce“:
  - ___ ____ ____ _____ ____ ____ ____ ____ ____;

- Majme tabuľku obce (id, nazov, okres, pocet_obyv) v schéme public.
- Čo vykonajú nasledujúce dopyty?
  - SELECT DISTINCT okres FROM obce;
  - SELECT * FROM obce WHERE nazov IN ('*Piešťany*' , '*Michalovce*');
  - SELECT názov, obyv FROM obce WHERE nazov LIKE '*Lúčky*' OR nazov LIKE '*Dúbravka*';
  - SELECT AVG(pocet_obyv) FROM obce GROUP BY (okres);

Praktické úlohy

- Z tabuľky *obyvatelia* si vyžiadajte:
  - prvých 10 záznamov s najvyšším počtom narodených v roku 2014,
  - stĺpec obec vo všetkých záznamoch, ktoré majú v názve spojovník,
  - všetky stĺpce pre záznamy v okresoch Banská Štiavnica alebo Banská Bystrica,
  - určte počty obcí, minimum stĺpca 2015 a maximum stĺpca 2015 po okresoch,
  - vytvorte pohľad *Ilava15* , v ktorom budú stĺpce obec a 2015 pre záznamy z okresu Ilava.