

Geografická báza údajov 1

Cvičenie 6

Vladimír Pelech

pelech2@uniba.sk, G-23



PRÍRODOVEDECKÁ FAKULTA
Univerzita Komenského
v Bratislave

Cvičenie 6

▪ Náplň:

- opakovanie z minulej hodiny,
- vkladanie do existujúcej tabuľky pomocou INSERT INTO ... SELECT,
- matematické funkcie,
- reťazcové funkcie,
- cvičné úlohy.

Opakovanie z minulej hodiny

- Aké typy obmedzení poznáte?
- Ako je možné vytvárať a mazať obmedzenia?
- Je možné pridať obmedzeniu názov? Ak áno, na čo je to dobré?
- V akom vzťahu sú primárny a cudzí kľúč a čo nám zaisťujú?
- Mení sa pridaním obmedzení tabuľke jej prípadné vymazanie?

Príprava na cvičenie

- Skontrolujte si, či v schéme *cviko3* máte tabuľku *obyvatelia*.
- Ak nie, tak si stiahnite údaje k cvičeniu 3 (cvicenie3Data.zip) a odzipujte ich.
- Vytvorte v schéme *cviko3* tabuľku *obyvatelia*.
- Názvy stĺpcov a ich dátové typy voľte podľa stĺpcov z odzipovaného *.csv* súboru.
- Importujte si csv súbor do príslušnej tabuľky.

INSERT INTO ... SELECT

- Dopyt slúži na vkladanie hodnôt do tabuľky na základe dopytu.
- Doteraz sa na vkladanie nových záznamov do tabuľky používal INSERT INTO alebo príkaz COPY.
- Na vytvorenie novej tabuľky pomocou dopytu SELECT sa využívala syntax:
 - CREATE TABLE názov schémy.názov tabuľky AS SELECT ...
- Dopyt INSERT INTO ... SELECT slúži na vloženie dát z jednej tabuľky do druhej.
- Pomocou príkazu je možné definovať atribúty, ktoré sa majú kopírovať.

INSERT INTO ... SELECT

- Na ukážku sa vytvorí tabuľka „okresy_p“ v schéme „cviko3“, ktorá bude obsahovať všetky záznamy, kde názov okresu začína na písmeno P.
- V ďalšom kroku sa záznamy s okresom začínajúcim na písmeno P z pôvodnej tabuľky vymažú.

```
CREATE TABLE cviko3.okresy_p AS  
SELECT * FROM cviko3.obyvratelia WHERE okres LIKE 'Okres P%';  
  
DELETE FROM cviko3.obyvratelia WHERE okres LIKE 'Okres P%';
```

INSERT INTO ... SELECT

- Ak by sa požadovalo opätovné vloženie údajov za okresy s názvom začínajúcim na P do tabuľky *obyvatelia*, tak je možné použiť práve INSERT INTO ... SELECT.
- Syntax príkazu, ak sa tabuľky zhodujú:
 - *INSERT INTO cviko3.obyvatelia SELECT * FROM cviko3.okresy_p;*

INSERT INTO ... SELECT

- Ak sa štruktúry tabuliek nezhodujú, prípadne je požiadavka na kopírovanie iba niektorých stĺpcov, tak je potrebné syntax pozmeniť:
 - *INSERT INTO cviko3.obyvratelia („2009“, okres) SELECT „2009“, okres FROM cviko3.okresy_p;*
- Platí, že vložené záznamy sú pridávané na koniec tabuľky.

3097	Okres Prešov	Lipníky	1	1	4	2
3098	Okres Prešov	Suchá Dolina	1	2	5	1
3099	Okres Pezinok	[null]	22	[null]	[null]	[null]
3100	Okres Pezinok	[null]	30	[null]	[null]	[null]
3101	Okres Pezinok	[null]	31	[null]	[null]	[null]

Príprava na prácu s funkciami

- V tabuľke *obyvatelia* si vytvorte 2 nové stĺpce:
 - *cislo* (int),
 - *desatinne* (numeric(10,4)).
- Stĺpec *cislo* naplňte hodnotami podľa vzorca: (hodnoty z roku 2009)-500
- Stĺpec *desatinne* naplňte hodnotami podľa vzorca: (hodnoty z roku 2009/7.0) - 100
- Skontrolujte si, či stĺpec *desatinne* obsahuje desatinné čísla.

Funkcie

- Doteraz sú známe iba agregačné funkcie a funkcia ROUND().
- Existuje ich však oveľa viac, ich počet sa medzi verziami môže líšiť.
- Slúžia na spracovanie a úpravu atribútov v tabuľke, podľa ich použitia v dopyte SELECT alebo UPDATE.
- Funkcie preberajú argumenty, ktoré sú uvedené v zátvorkách, pričom niektoré z argumentov môžu byť voliteľné.
- Prehľad dostupných funkcií závisí od verzie. Pre verziu 12 je ich možné nájsť na:
 - <https://www.postgresql.org/docs/12/functions.html>



Funkcie

- Na tomto cvičení sa kvôli nedostatku času nebudú preberať všetky funkcie.
- V prípade záujmu sa môžete kedykoľvek venovať manuálom.
- Dávať si pozor na dátové typy, s ktorými funkcia pracuje.

Matematické funkcie

dp – double precision

- `abs(x)` – absolútna hodnota čísla `x` a v rovnakom dátovom type, ako je číslo `x`;
- `ceil(dp alebo numeric)` – najbližšie celé číslo väčšie alebo rovné (`ceil(5.4) => 6`);
- `floor(dp alebo numeric)` – najbližšie celé číslo menšie alebo rovné (`floor(5.4) => 5`);
- `exp(dp alebo numeric)` – umocnené číslo `e` na uvedený argument.

Matematické funkcie - Ukážka

*SELECT cislo, abs(cislo), desatinne, ceil(desatinne), floor(desatinne), exp(1)
FROM cviko3.obyvratelia;*

cislo integer	abs integer	desatinne numeric (10,4)	ceil numeric	floor numeric	exp double precision
-478	478	-96.8571	-96	-97	2.71828182845905
-482	482	-97.4286	-97	-98	2.71828182845905
[null]	[null]	[null]	[null]	[null]	2.71828182845905
-490	490	-98.5714	-98	-99	2.71828182845905
-499	499	-99.8571	-99	-100	2.71828182845905

Matematické funkcie

- `ln (dp alebo numeric)` – prirodzený logaritmus argumentu,
- `log (dp alebo numeric)` – logaritmus argumentu pri základe 10,
- `log (numeric, numeric)` – logaritmus druhého argumentu pri základe prvého argumentu,
- `pi()` – konštanta π ,
- `power ( dp alebo numeric, dp alebo numeric)` – prvý argument umocní na druhý argument,
- `degrees (dp)` – prevedie uhol v radiánoch na stupne,
- `radians (dp)` – uhol v stupňoch prevedie na radiány.

Matematické funkcie - Ukážka

*SELECT cislo, ln(abs(cislo)), log(1000), log(5,25),
pi(), power(3,2), degrees(2*pi()), radians(180)
FROM cviko3.obyvratelia;*

- Pozor na použitie funkcie ln, obsahuje vo svojom argumente inú funkciu.

cislo integer	ln double precision	log double precision	log numeric	pi double precision	power double precision	degrees double precision	radians double precision
-478	6.16961073249146	3	2.0000000000000000	3.14159265358979	9	360	3.14159265358979
-482	6.1779441140506	3	2.0000000000000000	3.14159265358979	9	360	3.14159265358979
[null]	[null]	3	2.0000000000000000	3.14159265358979	9	360	3.14159265358979
-490	6.19440539110467	3	2.0000000000000000	3.14159265358979	9	360	3.14159265358979
-499	6.21260609575152	3	2.0000000000000000	3.14159265358979	9	360	3.14159265358979

Matematické funkcie

- `round (dp alebo numeric)` – zaokrúhli na celé číslo,
- `round (numeric, int)` – zaokrúhli prvý argument na počet desatinných miest v druhom argumente,
- `scale (numeric)` – počet číslic za desatinnou čiarkou,
- `sqrt (numeric)` – odmocnina z argumentu,
- `trunc (dp alebo numeric)` – odstráni desatinnú časť (`trunc (15.72) => 15`),
- `trunc (numeric, int )` – ponechá prvému argumentu počet číslic za desatinnou čiarkou uvedený v druhom argumente (`trunc (15.7218,2) => 15,72`),

Matematické funkcie - Ukážka

```
SELECT desatinne, round(desatinne),round(desatinne,2),  
scale(desatinne), scale(cislo),  
sqrt(abs(desatinne)),  
trunc (desatinne), trunc (desatinne,2)  
FROM cviko3.obyvratelia;
```

desatinne numeric (10,4) 	round numeric 	round numeric 	scale integer 	scale integer 	sqrt numeric 	trunc numeric 	trunc numeric 
-96.8571	-97	-96.86	4	0	600479596802	-96	-96.85
-97.4286	-97	-97.43	4	0	592687371919	-97	-97.42
[null]	[null]	[null]	[null]	[null]	[null]	[null]	[null]
-98.5714	-99	-98.57	4	0	313049053198	-98	-98.57
-99.8571	-100	-99.86	4	0	852445623322	-99	-99.85

Matematické funkcie

- `random()` - číslo v intervale $<0,1>$,
- `sin(x)`, `cos(x)`, `tan(x)`, `cot(x)`,
- `asin(x)`, `acos(x)`, `atan(x)`, `atan2(y,x)`,
- Ukážka vytvorenia určitého rozsahu náhodných čísel od najnižšieho po najvyššie:
 - `SELECT floor(random() * (najvyššie-najnižšie+1) + najnižšie);`
- <https://www.postgresql.org/docs/12/functions-math.html>

Matematické funkcie- Ukážka

```
1 SELECT random ();
2
3 /*nahodné číslo v rozsahu od 3 po 9*/
4 SELECT floor((random()*(9-3+1))+3);
5 SELECT floor((random()*(7))+3);
```

Data Output Explain Messages Notifications

	floor double precision	
1	4	

Reťazcové funkcie

- `char_length(reťazec)` alebo `len length(reťazec)` – počet znakov v reťazci,
- `lower (reťazec)` – vstupný reťazec iba s malými písmenami,
- `upper (reťazec)` – vstupný reťazec iba s veľkými písmenami,
- `concat (reťazec [, reťazec, ...])` – reťazec ako spojenie textovej reprezentácie všetkých argumentov, čiže aj čísel, dátumov, boolean hodnôt, hodnoty NULL sú ignorované,
- `concat_ws (oddeľovač typu reťazec, reťazec [, reťazec, ...])` – ako predchádzajúca funkcia, ale vstupné parametre budú oddelené znakom/mi oddeľovača z prvého argumentu.

Reťazcové funkcie - Ukážka

```
SELECT okres, "2009", char_length (okres),  
lower (okres), upper (okres),  
concat (lower(okres), "2009",upper(okres)),  
concat_ws ('%',lower(okres), "2009",upper(okres))  
FROM cviko3.obyvratelia;
```

okres character varying	2009 integer	char_length integer	lower text	upper text	concat text	concat_ws text
Okres Ilava	22	11	okres ilava	OKRES ILAVA	okres ilava22OKRES ILAVA	okres ilava%22%OKRES ILAVA
Okres Levice	18	12	okres levi...	OKRES LEVICE	okres levice18OKRES LE...	okres levice%18%OKRES LEVICE
Paríž	[null]	5	paríž	PARÍŽ	parížPARÍŽ	paríž%PARÍŽ
Okres Nitra	10	11	okres nitra	OKRES NITRA	okres nitra10OKRES NITRA	okres nitra%10%OKRES NITRA
Okres Nitra	1	11	okres nitra	OKRES NITRA	okres nitra1OKRES NITRA	okres nitra%1%OKRES NITRA

Reťazcové funkcie

- `initcap` (reťazec) – reťazec s veľkým písmenom na začiatku každého slova,
- `ltrim` (reťazec [, reťazec]) – z ľavej strany reťazca odstráni znaky nájdené v 2. argumente, bez uvedenia 2. argumentu odstráni medzery,
- `rtrim` (reťazec [, reťazec]) – to čo predchádzajúci, iba sprava,
- `lpad` (reťazec, int, doplňujúci sa reťazec)- doplní pred vstupný reťazec 1. argumentu taký počet znakov doplňujúceho sa reťazca 3. argumentu, aký je uvedený v 2. argumente. Ak je 2. argument kratší ako dĺžka vstupného reťazca, tak vstupný argument skráti na dĺžku 2. argumentu,
- `rpadd` (reťazec, int, doplňujúci sa reťazec)- ako `lpad`, iba znaky dopĺňa za vstupný reťazec.


```
SELECT okres, "2009",
lower(okres), initcap(lower(okres)), ltrim(' medzery'), ltrim(okres, 'ikOr'),
lpad(okres, 20, '*'), char_length(lpad(okres, 20, '*')), /*Dunajská Streda*/
rpad(okres, 20, 'ab')
FROM cviko3.obyvratelia;
```

okres character varying	2009 integer	lower text	initcap text	ltrim text	ltrim text	lpad text	char_length integer	rpad text
Okres Ilava	22	okres ilava	Okres Ilava	medzery	es Ilava	*****Okres Ilava	20	Okres Ilavaababababa
Okres Levice	18	okres levi...	Okres Levi...	medzery	es Levice	*****Okres Levice	20	Okres Leviceabababab
Paríž	[null]	paríž	Paríž	medzery	Paríž	*****Paríž	20	Parížabababababababa
Okres Nitra	10	okres nitra	Okres Nitra	medzery	es Nitra	*****Okres Nitra	20	Okres Nitraababababa
Okres Nitra	1	okres nitra	Okres Nitra	medzery	es Nitra	*****Okres Nitra	20	Okres Nitraababababa

Reťazcové funkcie

- `left (reťazec, int)` – počet znakov od začiatku reťazca uvedených v 2. argumente, ak je číslo záporné, tak odstráni z konca reťazca príslušný počet znakov,
- `right (reťazec, int)` – ako `left`, ale opačne,
- `replace (reťazec, vyhľadávaný reťazec, náhrada za vyhľadávaný reťazec)` – vysvetľujú argumenty, dôjde k nahradeniu každého výskytu,
- `split_part (reťazec, deliaci reťazec, int)` – vráti reťazec podľa poradia reťazcov v 3. argumente, pričom pôvodný reťazec rozdelí podľa znaku/znakov v 2. argumente.

Reťazcové funkcie - Ukážka

*SELECT okres, "2009", left (okres,7), left (okres,-7),
right (okres,5), right (okres,-4),
replace (okres, 'Okres',''), split_part (okres, ' ',3)
FROM cviko3.obyvratelia;*

okres character varying	2009 integer	left text	left text	right text	right text	replace text	split_part text
Okres Ilava	22	Okres I	Okre	Ilava	s Ilava	Ilava	
Okres Levice	18	Okres L	Okres	evice	s Levice	Levice	
Paríž	[null]	Paríž		Paríž	ž	Paríž	
Okres Nitra	10	Okres N	Okre	Nitra	s Nitra	Nitra	
Okres Nitra	1	Okres N	Okre	Nitra	s Nitra	Nitra	

Reťazcové funkcie

- `substring (reťazec, int [, int])` – reťazec od poradového znaku v 2. argumente, ak bude uvedený 3. argument, tak vrátený reťazec bude mať dĺžku uvedenú v treťom argumente, ak 3. argument nebude uvedený, vráti zvyšok reťazca,
- `strpos(reťazec, vyhľadávaný reťazec)` – poradové číslo znaku v reťazci 1. argumentu, kde začína 2. argument. Niekedy je výhodné použiť túto funkciu v spojení s inými funkciami.
- <https://www.postgresql.org/docs/12/functions-string.html>

Reťazcové funkcie - Ukážka

```
SELECT okres, "2009",  
substring (okres, 7),  
strpos (okres, 'a'), substring (okres, strpos(okres, 'a'))  
FROM cviko3.obyvratelia;
```

okres character varying	2009 integer	substring text	strpos integer	substring text
Okres Ilava	22	Ilava	9	ava
Okres Levice	18	Levice	0	Okres Levice
Paríž	[null]		2	aríž
Okres Nitra	10	Nitra	11	a
Okres Nitra	1	Nitra	11	a

Opakovanie z tohto cvičenia

- Import do tabuliek pomocou `INSERT INTO ... SELECT`
 - Pozor na kompatibilitu stĺpcov medzi tabuľkami.
- Funkcie:
 - majú povinné a voliteľné argumenty,
 - výstup z funkcie môže slúžiť ako argument inej funkcie,
 - matematické,
 - Reťazcové.
- Neprešli sme všetky, v prípade záujmu pozrieť manuál.



V budúcnosti

- Časové a dátumové typy a funkcie nad nimi.
- Návrh databázy, to sme vôbec neriešili a je to základ fungovania. Ak je štruktúra databázy dobre navrhnutá a popísaná, tak to ušetrí veľké množstvo času a prostriedkov pri jej správe.
- Do databázy zahrnieme priestorovú zložku, čiže budeme pracovať s geometriou objektov.



Vaše otázky?



Otázky

- Aký je rozdiel medzi CREATE TABLE ... AS SELECT a INSERT INTO ... AS SELECT?
- Čo je to argument funkcie?
- Môže vystupovať funkcia v argumente inej funkcie?
- Má vplyv použitie funkcie na dáta?



- Majme tabuľku obce (id, nazov, okres, pocet_obyv) v schéme public, ktorej 1 zo záznamov vyzerá takto:

- 1, 'Horný Koniec', 'Konce', 500

- Aký výsledok vrátia nasledujúce funkcie:

- LENGTH(okres)

- CEIL (ABS(-7.1521))

- DEGREES (2*PI())

- LEFT(UPPER (okres),2)

- REPLACE (nazov, 'on', 'ep')

Praktické úlohy

- V schéme *ulohy* si vytvorte tabuľku *obyvatelia_vychod*, kde budú všetky záznamy z tabuľky *cviko3.obyvratelia*, kde je v názve okresu Prešov alebo Košice.
- Pridajte do tabuľky *obyvatelia_vychod* záznamy za okresy Poprad a Trebišov.
- Vytvorenej tabuľke:
 - určte počet znakov najdlhšieho názvu obce,
 - odstráňte reťazec „Okres „ z názvu okresu,
 - určte priemerné počty po okresoch za rok 2015 a zaokrúhlite ich na celé číslo nadol,
 - vyžiadaajte si stĺpce obec a 2015 do jedného stĺpca, oddeľovačom nech je znak %,
 - určte pozíciu písmena „v“ v reťazcoch stĺpca *obec*.