

SQL, DDL, DML

Databázové praktikum

ZS 2015/2016

SQL

- Structured query language, v zásade štandard starý 20 rokov
- Zatiaľ sme hovorili iba o príkaze SELECT, ktorý slúži na získavanie dát z databázy
- Súčasťou štandardu SQL sú aj
 - **DDL:** Data definition language (CREATE TABLE, ALTER TABLE, ...)
 - **DML:** Data manipulation language (INSERT / UPDATE / DELETE)
 - **DCL:** Data control language (GRANT / REVOKE)
- Štandard SQL definuje „základnú“ funkcionálnosť, ktorú viac-menej podporuje každý databázový systém
- Databázové systémy však často rozširujú štandard o ďalšie funkcie
 - Napr. INSERT ON DUPLICATE KEY... (MySQL), špecifické typy polí, funkcie a pod...

DDL

- CREATE DATABASE <db-name>
- DROP DATABASE [IF EXISTS] <db-name>
 - Vaše konto na cvika.dcs.fmph.uniba.sk nemá oprávnenie na vytváranie a mazanie databáz
- CREATE TABLE <table-name> (...)
 - Vytvorenie novej tabuľky v databáze
 - V zátvorkách sú uvedené názvy polí a ich typy (prípadne ďalšie parametre)
- DROP TABLE [IF EXISTS] <table-name>;
 - Vymazanie tabuľky z databázy
- ALTER TABLE <tablename> ADD/MODIFY/DROP [column/index]
<colum/index-name> ...
 - Zmena polí tabuľky

CREATE TABLE

```
CREATE TABLE person (  
    personid serial NOT NULL,  
    firstname varchar(50),  
    lastname varchar(50) NOT NULL,  
    date_of_birth date,  
    login_count int NOT NULL DEFAULT 0,  
    PRIMARY KEY (personid),  
    KEY (lastname, firstname)  
);
```

- Veľa rôznych typov polí:
 - <http://www.postgresql.org/docs/current/static/datatype.html>
- Postgres dokumentácia:
 - <http://www.postgresql.org/docs/current/static/sql-createtable.html>

Typy polí

Text:	
varchar(n)	variable-length with limit
text	variable unlimited length
Number:	
int/smallint/bigint	
real, numeric, double precision	
Date and time:	
date	calendar date (year, month, day)
time	time of day (no time zone)
timestamp	date and time (no time zone)

Špeciálne typy polí

- serial / int auto_increment – automatické číslovanie riadkov (vhodné ako primárny kľúč)
- JSON / JSONB – JavaScript Object Notation

- {
 meno: 'Laco',
 priezvisko: 'Novák',
 adresa: {
 'ulica': 'Štúrova',
 'cislo': '2/B',
 ...
 },
 ...
}

- Niekedy dopredu neviete povedať aké dáta / stĺpce budete potrebovať
- Niekedy je škoda pridávať ďalší stĺpec kvôli údaju, ktorý potrebujete evidovať pre 1% riadkov
- rapid development, prebraté NoSQL databázy, ...
- <http://www.postgresql.org/docs/current/static/datatype-json.html>

Indexy

- Umožňujú rýchlejšie vyhľadávanie záznamov
 - Rýchlejšie oproti tzv. full-table-scan – prehľadávaniu všetkých riadkov tabuľky
- Za účelom rýchlejšieho vyhľadávania systém uchováva samostatný index
 - Rýchlejšie vyhľadávanie ($O(1)$ resp. $O(\log n)$) oproti $O(N)$
 - Možno pomalšie vkladanie / úprava ($O(\log n)$ oproti $O(1)$)
- Primárny kľúč
 - Stĺpec resp. množina stĺpcov, ktoré jednoznačne určujú ostatné stĺpce tabuľky
 - Unique index
- Hašované indexy
- Vyhľadávacie stromy (B stromy / B+ stromy / ...)
- <http://www.postgresql.org/docs/current/static/indexes.html>

Indexy

Index definujeme:

- CREATE TABLE <table-name> (
 ...
 KEY `deptno` (`deptno`),
 UNIQUE KEY `name` (`lastname`,`firstname`),
 ...
)
- CREATE INDEX <name> ON <table> USING btree (column)
- Index může být vytvořený aj na viacerých stĺpcoch a na výraze nad stĺpcom napr (lower(firstname)).

Indexy

- Typ indexu určuje aj to, ako vieme vyhľadávať
- HASH indexy (USING hash) – funguje len na “=”
- B-Stromy (USING btree) – funguje aj na “<“, “>”, “<=“, “>=“, “=”
 - vieme vyhľadávať prefixy v reťazcoch (napr. funguje <column> LIKE “prefix%”)
- Špeciálne indexy (GIST v postgres) umožňujú vyhľadávať napr. najbližšieho suseda na mape a pod.
- Podrobnejšie o indexoch na prednáške a ďalších cvikách
- <http://www.postgresql.org/docs/current/static/indexes-types.html>

ALTER TABLE

- Zmena definície tabuľky, bez toho aby došlo k strate jej obsahu
 - <http://www.postgresql.org/docs/current/static/sql-altertable.html>
- Pridávanie stĺpcov:
 - `ALTER TABLE <table-name> ADD COLUMN <column-name> <type>`
- Mazanie stĺpcov
 - `ALTER TABLE <table-name> DROP COLUMN <column-name>`
- Zmena stĺpcov
 - `ALTER TABLE <table-name> ALTER COLUMN <column-name> TYPE <type> ...`
- Veľa ďalších funkcií
 - Pozri dokumentáciu

DML

Vkladanie dát:

- INSERT INTO <table-name> (<col1>,<col2>,...) VALUES (...),(...)
- INSERT INTO <table-name> (<col1>,<col2>,...) SELECT ... FROM ...
 - <http://www.postgresql.org/docs/current/static/sql-insert.html>

Úprava dát:

- UPDATE <table-name> SET <col1>=<val1>, <col2>=<val2> WHERE <condition>
 - <http://www.postgresql.org/docs/current/static/sql-update.html>

Mazanie dát:

- DELETE FROM <table-name> WHERE <condition>
 - <http://www.postgresql.org/docs/current/static/sql-delete.html>

INSERT

purchase(id, buyer, seller, product, store)

- **INSERT INTO** purchase (buyer, seller, product, store)
VALUES ('Joe', 'Fred', 'wakeup-clock-espresso-machine',
 'The Sharper Image')
- **INSERT INTO** purchase
(1, 'Joe', 'Fred', 'wakeup-clock-espresso-machine', 'The Sharper'),
(2, 'Joe', 'John', 'smart-tv', 'unknown');
- **INSERT INTO** purchase (buyer, seller, product, store)
SELECT order_to, order_from, product, store FROM orders
WHERE processed='1'

UPDATE

- UPDATE product
SET price = price/2
WHERE productid>20
- UPDATE product
SET price = price/2
WHERE Product.name IN
(SELECT product
FROM Purchase
WHERE Date ='2015-11-09');

DELETE

- `DELETE FROM` purchase
`WHERE` seller='joe' and product='Brooklyn Bridge'

Kódovanie a collations

- Nekompatibilné kódovania textových polí sú častým zdrojom problémov
- Kódovanie – ako sú jednotlivé znaky textu binárne kódované
 - **UTF8**, WIN1250, LATIN1,...
 - Ak chceme porovnať reťazce v rôznom kódovaní, musíme ich najprv „prekódovať“ na jednotný formát
 - nie vždy sa to dá (rôzne sady znakov)
 - Pozor na rôzne typy kódovania klienta (napr. konzola, .sql súbor) a servera
 - SET CLIENT ENCODING utf8;
 - V súčasnosti je asi vhodné všade používať UTF8

Kódovanie a collations

- Collation
 - Pravidlá pre usporiadavanie reťazcov
 - Klasifikácia znakov
 - čo je písmeno?
 - aký je UPPER-CASE ekvivalent písmena?
 - V niektorých prípadoch (napr. MySQL) vieme povedať, či rozlišujeme malé/veľké písmená pri vyhľadávaní a pod.
 - Modul CITEXT v Postgres

Kódovanie a collations

- Nejednotnosť medzi databázovými systémami, niekedy chaos,...

- Postgres:

- `CREATE DATABASE <db-name>
ENCODING='UTF8'
LC_COLLATE="sk_SK.UTF8"
LC_CTYPE="sk_SK.UTF8"`

- `[CREATE COLLATION sk (locale="sk_SK.UTF8");]`

- `CREATE TABLE <table-name> (
...
description text COLLATE 'sk_SK.UTF8',
...
)`

- <http://www.postgresql.org/docs/current/static/multibyte.html>, <http://www.postgresql.org/docs/current/static/collation.html>