

SQL: DML, DDL

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.

DML

- **vkładanie záznamov (riadkov):**

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 riadkov:**

UPDATE <table-name> SET <col1>=<val1>, <col2>=<val2> WHERE <condition>

- <http://www.postgresql.org/docs/current/static/sql-update.html>

- **mazanie riadkov:**

DELETE FROM <table> 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', 'espresso-machine', 'The Sharper')
- **INSERT INTO** purchase
(1, 'Joe', 'Fred', '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'

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] <column name> ...
 - zmena atribútov (stĺpcov), podobne sa pridávajú / rušia indexy

CREATE TABLE

```
CREATE TABLE person (  
    personid SERIAL NOT NULL,  
    firstname VARCHAR(50),  
    lastname VARCHAR(50) NOT NULL,  
    date_of_birth DATE,  
    login_count INTEGER NOT NULL DEFAULT 0  
);
```

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

Dátové typy

Text:	
varchar(n)	string, variable-length with limit
text	string, variable unlimited length
Number:	
int/smallint/bigint	celé číslo
real, numeric, double precision	reálne číslo
decimal	fixný počet desatinných miest
Date and time:	
date	calendar date (year, month, day)
time	time of day (no time zone)
timestamp [with timezone]	date and time

Špeciálne dátové typy

- serial / 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, alebo sú variabilné (napr. pre každý riadok niekoľko individuálnych položiek)
- rapid development, dáta prebraté z NoSQL databázy či RESTful API, ...
- <http://www.postgresql.org/docs/current/static/datatype-json.html>

ALTER TABLE

- zmena definície tabuľky bez straty 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, vid' dokumentáciu

Indexy

- umožňujú rýchlejšie vyhľadávanie záznamov
 - rýchlejšie oproti tzv. full table scan (prehľadávaniu všetkých riadkov tabuľky)
- pre zrýchlené vyhľadávanie 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
- dva základné typy indexov:
 - hašovacie tabuľky
 - vyhľadávacie stromy (B/B+)
- <http://www.postgresql.org/docs/current/static/indexes.html>

Indexy

- vytvorenie:

```
CREATE INDEX <name> ON <table> USING btree (<columns>)
```

```
CREATE TABLE <table> (  
    firstname TEXT, lastname TEXT,  
    UNIQUE (lastname, firstname),  
)
```

- index môže byť vytvorený aj na viacerých stĺpcoch a na výraze zostrojenom z hodnôt v jednotlivých stĺpcoch, 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.
- <http://www.postgresql.org/docs/current/static/indexes-types.html>

Kódovanie a collations

- nekompatibilné kódovania reťazcov 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
pozor: utf8 v MySQL nie je utf8, treba použiť utf8mb4...

<https://medium.com/@adamhooper/in-mysql-never-use-utf8-use-utf8mb4-11761243e434>

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 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>