

NÁSKOK
DÍKY
ZNALOSTEM

PROFINIT

B0M33BDT

Technologie pro velká data

Supercvičení – SQL, Python, Linux

Sergej Stamenov, Jan Hučín

18. 10. 2017

Osnova cvičení

- › Linux
- › SQL
- › Python

SQL pro uživatele

aneb co potřebuje znát a umět bigdatový uživatel:

- › co to je relační databáze
- › co to je tabulka
- › základní datové typy polí, důležité DDL příkazy
- › důležité DML příkazy
- › trochu víc o SELECTech
- › trochu víc o JOINech
- › agregační funkce a dotazy
- › vnořený SELECT

Relační databáze a tabulky

Relační databáze:

- › kolekce tabulek a informací o jejich vztazích

Tabulka:

- › data ve dvojrozměrné struktuře (záznamy, pole)
- › metadata (datové typy, klíč, indexy atd.)

Každá tabulka patří do nějaké databáze.

Notace:

databáze.tabulka

databáze.tabulka.pole

Pokud je nadřazený objekt jednoznačný, lze jeho název vynechat.

Data v tabulkách – typy

- › celá čísla: int, smallint, bigint, ...
- › desetinná čísla: float, double, decimal, ...
- › textové řetězce: varchar(N), char(N), text, ...
- › datum a čas: datetime, date, time, timestamp, ...
- › a další
- › NULL – neznámá hodnota

Data v tabulkách – DDL příkazy

Poznámka: příkazy VELKÝMI – konvence, lze psát malými

- › CREATE DATABASE *databáze*
- › SHOW DATABASES
- › USE *databáze*
- › DROP DATABASE *databáze*
- › CREATE TABLE *tabulka* (pole1 typ, pole2 typ, ..., poleK typ)
 - **na co ještě musím dát pozor?**
- › DROP TABLE *tabulka*
- › SHOW TABLES
- › a další, např. ALTER TABLE...

Data v tabulkách – DML příkazy

Insert

- › INSERT INTO TABLE *tabulka* VALUES (*řádek*), (*řádek*), ...
- › INSERT INTO TABLE *tabulka* *příkaz_select*

Update

- › UPDATE *tabulka* SET *pole=hodnota*, ... [WHERE *podmínka*]

Delete

- › DELETE FROM *tabulka* [WHERE *podmínka*]

Truncate table *tabulka*

DQL – select

SELECT [DISTINCT] *výraz1, výraz2, ...*

FROM *tabulka* | *spojení tabulek*

[WHERE *podmínka*]

[GROUP BY *sloupec1, sloupec2, ...*]

[ORDER BY *sloupec1, sloupec2, ...*]

[HAVING *podmínka*]

[LIMIT *počet*]

(někdy TOP místo LIMIT)

hvězdička * znamená všechny sloupce:

› SELECT * FROM *tabulka*

DQL – spojení tabulek (join)

- › Join připojí k první tabulce ty řádky z druhé tabulky, kde hodnoty polí vyhovují vazební podmínce.

tabulka1 JOIN tabulka2 ON vazební_podmínka

- › **Inner join** – spojí se jen ty řádky, kde je podmínka splněna
- › **Left join** – pokud k řádku v *tabulka1* není řádek v *tabulka2* splňující podmínku, připojí se místo toho řádek s hodnotami NULL
- › **Right join** – naopak
- › **Full outer join** – sjednocení left join a right join
- › **Cross join** – ke každému řádku *tabulka1* se připojí všechny řádky z *tabulka2*
- › **Natural join** – join bez vazební podmínky, spojení je přes pole se stejným názvem

DQL – vnořený select

- › Výsledek SELECTu je tabulka => lze ji použít jako vstup pro další SELECT.
- › Bývá potřeba pojmenovat tabulku, případně i výrazy:
(SELECT ... AS *název_pole*) AS *název_tabulky*

Příklad:

Vypsát počet unikátních jmen

- › SELECT DISTINCT jmeno FROM lide – unikátní jména
 - › SELECT count(*) FROM ... – počet záznamů
- ⇒ dohromady:
- › SELECT count(*) FROM
(SELECT DISTINCT jmeno FROM lide) AS jmena_unik

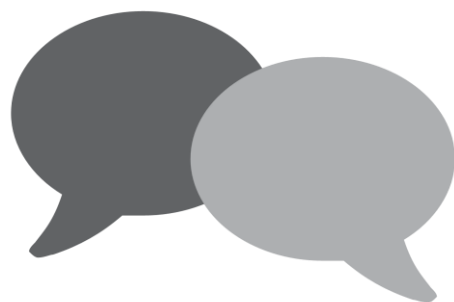
Agregační funkce a dotazy

Agregační dotaz – obsahuje agregační funkci

- › místo všech vyhovujících řádků vypíše nějakou agregovanou hodnotu
- › agregační funkce: např. sum, avg, count
- › agregace všech řádků, nebo po skupinách –
GROUP BY *třídící_sloupec*

Připojení na cluster

- › PuTTY: 212.24.144.105
- › login + heslo



Diskuze

Díky za pozornost

PROFINIT

NÁSKOK DÍKY ZNALOSTEM

Profinit EU, s.r.o.
Tychonova 2, 160 00 Praha 6



Telefon
+ 420 224 316 016



Web
www.profinit.eu



LinkedIn
linkedin.com/company/profinit



Twitter
twitter.com/Profinit_EU