

Funkce v SQL

Práce s daty, 13. dubna 2023

Přehled kategorií funkcí

Databázové systémy disponují funkcemi pro:

- práci s čísly
- práci s řetězci
- práci s časovými hodnotami
- rozhodování
- seskupená data

Některé systémy mají také další kategorie funkcí:

- geometrické, geografické, kryptografické, kompresní, ...

Společné základy

Co je dobré vědět pro začátek:

- **TRUE** a **FALSE** – konstanty odpovídající hodnotám 1 a 0
- (...) – prioritu vyhodnocování operátorů lze ovlivnit závorkami
- **CAST**, **CONVERT** – s pomocí těchto funkcí lze změnit datových typ
 - Např. `CAST('148' AS INTEGER)`, `CAST(148 AS CHAR)`
 - Do jisté míry se nicméně přetypování děje [automaticky](#)
- Výsledky funkcí lze použít jako vstupní parametry jiných funkcí
- Výsledky některých funkcí mohou záviset na tom, jaká je aktuální používaná znaková sada
- A výsledkem funkce také často může být hodnota NULL

Numerické funkce

- + - * / % (zbytek po dělení)
- DIV – celočíselné dělení
- RAND – generování náhodného čísla
- ROUND, CEIL, FLOOR – zaokrouhlování (přirozené, nahoru, dolů)
- SIN, COS, TAN – sinus, kosinus, tangens
- LOG, LOG2, LOG10, LN – logaritmy
- POW, SQRT – mocnina, druhá odmocnina
- LEAST, GREATEST – nejmenší či největší hodnota z čísel na vstupu
- a další, viz např. [numerické funkce v MariaDB](#)

Řetězcové funkce

- CONCAT, CONCAT_WS – spojování řetězců
- LENGTH, CHAR_LENGTH – délka řetězce (v bajtech, ve znacích)
- TRIM, LPAD, RPAD – odstranění / přidání přepon a/nebo přípon
- LOCATE – vrací pozici hledaného podřetězce v daném řetězci
- SUBSTR – vrací podřetězec z řetězce od dané pozice a dané délky
- REPLACE – nahradí všechny výskyty podřetězce jiným textem
- INSERT – od dané pozice přepíše řetězec jiným řetězcem
- LOWER, UPPER – mění velikost písmen v daném řetězci
- a další, viz např. [řetězcové funkce v MariaDB](#)

Řetězcové funkce pro regulární výrazy

- REGEX, RLIKE – regulární alternativa k LIKE
- NOT REGEX, NOT RLIKE – negace srovnávání dle RE
- REGEXP_INSTR – vrací první pozici řetězce, který vyhovuje masce RE
- REGEXP_REPLACE – v řetězci nahradí všechny vyhovující výskyty
- REGEXP_SUBSTR – vrátí podřetězec vyhovující masce RE

MariaDB využívá PCRE a umí většinu rozšíření včetně Unicode, líných kvantifikátorů, vnořených modifikátorů či *lookarounds*. Více [zde](#).

Cvičení I

Pracujte s tabulkou rc:

1. Vypište, kolik osob se narodilo sedmý den v měsíci.
2. Vypište ta rodná čísla, která nejsou validní.
 - Česká rodná čísla by měla být dělitelná 11.
3. Vypište hodnoty písmen ze sloupců A a B (volitelně i C) v abecedním pořadí jako spojený řetězec.
4. Ke každému rodnému číslu dekodujte pohlaví – vypište 0 pro muže a 1 pro ženy.
 - Ženy mají k hodnotě měsíce přičteno číslo 50.

Funkce pro práci s časem

- CURTIME, CURDATE, NOW – vrací aktuální *čas, datum, datum a čas*
- YEAR, MONTH, DAY, HOUR, MINUTE – číselná hodnota dané části času
- DAYOFWEEK, DAYOFMONTH, DAYOFYEAR – číslo dne v daném období
- DAYNAME, MONTHNAME – vypíše jméno dne nebo měsíce
- MAKETIME, MAKEDATE – vytvoří čas či datum ze zadaných hodnot
- STR_TO_DATE, DATE_FORMAT – řetězec na datum a zpět
- DATEDIFF, PERIOD_DIFF – počet dní nebo měsíců mezi dvěma daty
- ADDDATE, SUBDATE, ADDTIME, SUBTIME – přičítání data a času
- a mnoho dalších, viz např. [funkce pro práci s časem v MariaDB](#)

Cvičení II

1. Do sloupce *dn* vložte datum narození ve tvaru 'YYYYMMDD' .
 - *Všechny osoby ve vzorku se narodily před rokem 2000*
2. Vypočítejte aktuální věk osob.
 - a) Volitelně jako počet let plus počet měsíců.
3. Naformátujte datum narození ve tvaru „5. září 2005“.
4. Ke každé osobě vypište název dne,
 - a) kdy se dotyčná osoba narodila;
 - b) kdy má v letošním roce dotyčná osoba narozeniny.

Názvy dnů a měsíců se vypisují dle aktuálního národního nastavení:

- `set lc_time_names='cs_CZ';`

Řídící funkce (rozhodování)

Tj. výsledek funkce závisí na podmínce

- *IF(podmínka, výsledek při splnění, výsledek při nesplnění)*
 - `IF(10>5, 'ano', 'ne');`
- *CASE testovaná hodnota WHEN hd1 THEN výsl1 WHEN hd2 THEN výsl2 ... ELSE výsledek jinak END*
 - `CASE country WHEN 'cze' THEN 'Česko' WHEN 'ger' THEN 'Německo' WHEN 'pol' THEN 'Polsko' ELSE 'jiná' END;`
- *IFNULL(NULL?, náhrada)*
- *NULLIF(h1, h2)* – vrátí NULL, pokud $h1=h2$, jinak vrátí $h1$
- *x IS NULL* – prosté testování, zda x je NULL

Cvičení III

1. Ke každému rodnému číslu vypište pohlaví – 'muž' nebo 'žena'.
2. Vypište hodnoty písmen ze sloupců A, B a C v abecedním pořadí jako spojený řetězec s využitím funkce IF.
 - *domácí úkol*