

Práce s daty – shrnutí předmětu

4. května 2023

Životní cyklus dat

1. Plánování správy dat
2. Organizování a dokumentace (návrh schématu SQL databáze)
3. Zpracování dat (RE, SQL)
4. Uložení dat (SQL databáze)
5. Ochrana dat (uživatelské účty a oprávnění v SQL databázi)
6. Archivování a zveřejnění dat
7. Objevování dat

Regulární výrazy

Řetězec reprezentující vyhledávací masku používáme hlavně pro

1. Validaci vstupních dat
2. Hromadné změny v textech

RE nabízí konstrukci masky s pomocí:

- zástupných znaků (např. tečka, `\d`, `\w` ...)
- opakování (`*` + `?`, `{}`)
- vyznačení začátků a konců řádků či slov (`^`, `$`, `\b`)
- negace výčtu (`[^...]`, `\D`, `\W`)
- alterace (s pomocí `|` ve smyslu logické spojky „nebo“)

Relační databáze

- Způsob ukládání propojených dat
- Teoretický základ v relačním modelu (kalkulu)
 - selekce, projekce, rozdíl, sjednocení, kartézský součin, spojení
- Data jsou uložena v tabulkách
- SQL je jazyk pro práci s relačními databázemi
 - Do značné míry univerzální napříč systémy různých výrobců (Oracle, MSSQL, MySQL, MariaDB, PostgreSQL, SQLite, ...)
 - Připomíná běžné anglické věty

Relační model I

- Projekce – vybíráme sloupce
 - `SELECT call_no, barcode FROM knihy;`
- Selekce – vybíráme řádky
 - `SELECT title FROM knihy WHERE call_no LIKE 'FIN%';`
 - `SELECT title FROM knihy GROUP BY title HAVING sum(price)>5000;`
- Kartézský součin = každý s každým
 - `SELECT * FROM knihy JOIN vyp;`
- Spojení – specifikujeme podmínku spojení, obvykle rovnost klíčů
 - `SELECT * FROM knihy JOIN vyp ON knihy.id=vyp.reckey;`

Relační model II

- Sjednocení
 - `SELECT title FROM knihy WHERE call_no LIKE 'POJ%'`
UNION `SELECT author FROM knihy WHERE price>2000;`
- Rozdíl
 - `SELECT title FROM knihy WHERE call_no LIKE 'M%'`
EXCEPT `SELECT title FROM knihy WHERE call_no LIKE 'MAR%';`
- Průnik
 - `SELECT title FROM knihy WHERE call_no LIKE 'M%'`
INTERSECT `SELECT title FROM knihy WHERE price>1000;`
- V MariaDB se u těchto operátorů nevypisují duplicity
 - Pokud je chceme, zadáme `UNION ALL`, `EXCEPT ALL`, `INTERSECT ALL`

Pokročilé možnosti SELECT

- DISTINCT – odstranění duplicitních řádků ve výpisu
- LIMIT – omezení počtu vypsaných řádků
- GROUP BY – seskupování řádků dle zvolených sloupců
- ORDER BY ASC | DESC – řazení výstupu
- LEFT | RIGHT JOIN – jednostranné spojení
- Vnořené dotazy
 - A s nimi spojené operátory IN, ANY, ALL, EXISTS
- Spojování tabulek samých se sebou – s pomocí aliasů

Funkce v jazyce SQL

- Agregáční
- Pro práci s čísly
- Pro práci s řetězci
 - S využitím regulárních výrazů
- Pro práci s časovými údaji
- Okenní (ROW_NUMBER, RANK, LAG, LEAD, MEDIAN, ...)
- Řídící (IF, CASE, IFNULL, NULLIF)
- Konverzní (CAST, CONVERT)

Manipulace s daty

- CREATE TABLE, CREATE DATABASE
- INSERT INTO tabulka
- REPLACE tabulka
- UPDATE tabulka
- DELETE FROM tabulka
- DROP TABLE, DROP DATABASE
- LOAD DATA [LOCAL] INFILE 'soubor' INTO tabulka ...
- SELECT ... INTO OUTFILE 'soubor'

Optimalizace databáze

- Indexy = klíče – rychlejší přístup k datům
 - PRIMARY KEY, UNIQUE KEY, INDEX, FULLTEXT
- Omezení toho, co může být v tabulce uloženo – CONSTRAINT
 - Dle podmínky pro daný sloupec – CHECK
 - Dle vazby na primární klíč jiné tabulky – FOREIGN KEY
- Spouštěče – TRIGGER
 - Při manipulaci s daty spustí požadovanou akci
- Zamykání tabulek – LOCK TABLE
 - Aby např. do tabulky nemohlo najednou zapisovat více uživatelů

Administrace databáze

- Správa uživatelů a rolí
 - CREATE | DROP USER, SET PASSWORD
 - CREATE | DROP ROLE, SET [DEFAULT] ROLE
- Nastavení oprávnění
 - GRANT | REVOKE *právo* ON *objekt* TO *uživatel*
- Práce s proměnnými databázového systému
 - SHOW VARIABLES, SET *proměnná* = *hodnota*
- Národní abecedy
 - SHOW CHARACTER SET – výběr znakových sad
 - SHOW COLLATION – výběr metod řazení dat

Ukázka vytvoření a zavedení vlastní funkce

Definice funkce v souboru vyhovuje.sql:

```
DELIMITER //
CREATE FUNCTION vyhovuje(obor_p VARCHAR(30), obor_s VARCHAR(30), prumer
decimal(10,2))
  RETURNS VARCHAR(3)
  BEGIN
    DECLARE vysledek VARCHAR(3);
    IF (prumer > 0) and (prumer<=2.65) and (obor_p=obor_s)
      THEN SET vysledek = 'ano'; ELSE SET vysledek = 'ne'; END IF;
    RETURN vysledek;
  END //
DELIMITER ;
```

Zavedení této funkce do systému: SOURCE vyhovuje.sql