

SQL a správa databáze

Práce s daty, 27. dubna 2023

Indexy

= uspořádaná data s odkazem na originální záznam. Rozlišujeme:

- PRIMARY KEY – primární klíč tabulky, může být jen jeden, nesmí obsahovat prázdné záznamy (tj. hodnoty NULL)
- UNIQUE KEY – může obsahovat NULL, jinak jen unikátní hodnoty
- INDEX / KEY – prostý index bez unikátních hodnot
- FULLTEXT – index pro hledání textů v textových sloupcích
 - `SELECT * FROM t1 WHERE MATCH(sl1) AGAINST('slovo');`

Vytvoření indexu při definici / změně tabulky nebo pomocí

- `CREATE [typ] INDEX název ON tabulka (sloupec1, ...);`

Omezení sloupců

- Kromě omezení na unikátní hodnoty sloupců u primárního a unikátního klíče (indexu) lze dále nastavit CONSTRAINT:
- CHECK – kontroluje se, zda uložená data splňují podmínku
 - `CREATE TABLE t1 (a INT CHECK(a>100), b INT, CONSTRAINT omezeni CHECK (a+b=0));`
- FOREIGN KEY – kontroluje se vazba na primární klíč v rodičovské tabulce
 - `CREATE TABLE t2 (m INT NOT NULL, n VARCHAR(5), FOREIGN KEY (m) REFERENCES t1 (a) ON UPDATE cascade ON DELETE restrict);`
 - Další možností automatické vazby cizího klíče je SET NULL

Spouštěče

Ke každé tabulce lze k operacím INSERT, UPDATE a DELETE přiřadit spouštěč (TRIGGER), který spustí definovanou akci, například:

- `CREATE TRIGGER vypujcka AFTER INSERT ON knihy2 FOR EACH ROW INSERT INTO vyp2 (reckey, bortype, renewals, ld) VALUES (new.id, 'SP', 0, now());`
 - Vytvoří se spouštěč *vypujcka*, který se spustí po každém vložení nového řádku do tabulky *knihy2*
 - Automaticky se vloží nový řádek do tabulky *vyp2* se zadanými parametry
 - *ld* z *knihy2* se pomocí *new.id* zkopíruje do sloupce *reckey* v tabulce *vyp2*
- `SHOW TRIGGERS` – ukáže spouštěče; `DROP TRIGGER spouštěč` – smaže

Optimalizace tabulek a dotazů

- OPTIMIZE TABLE *tabulka* – defragmentace dat v tabulce
- REPAIR TABLE *tabulka* – pokus o opravu dat (jen některé enginy)
- RENAME TABLE *tabulka* TO *tabulka2* – přejmenování tabulky
- LOCK TABLES *tabulka typ_zámku* – zamkne tabulku dle parametru:
 - READ, READ LOCAL, WRITE, WRITE CONCURRENT
- SELECT ... LIMIT *odkud, počet řádků* – omezení počtu řádků ve výpisu
- SELECT ... LIMIT ROWS EXAMINED *počet* – limit zpracovaných řádků
- EXPLAIN SELECT ... | UPDATE ... | DELETE ... – vysvětlí dotaz
- ANALYZE *dotaz* – spustí a analyzuje dotaz

Cvičení I

1. Vytvořte fulltextový index ke sloupci *title* v tabulce *knihy* a s pomocí MATCH – AGAINST najděte signatury knih, v jejichž názvu se vyskytuje slovo „marketing“.
2. Vytvořte tabulku *cisla* s číselnými sloupci X a Y s podmínkou, že v Y nesmí být menší číslo jak v X (v daném řádku).
3. Vytvořte tabulku *cisla2* s číselným sloupcem Z a spouštěč, který zajistí, že při každém vložení dat do tabulky *cisla* se sem do sloupce Z uloží součet čísel vkládaných do sloupců X a Y.
4. Tabulku *cisla* uzamkněte pro zápis a otestujte funkčnost zámku.

Správa uživatelů databáze

Komplexní databázové systémy umožňují správu uživatelů databáze včetně nastavení oprávnění pro práci s databázovými objekty.

Založení nového uživatelského účtu:

- `CREATE USER 'joe';`
- `CREATE USER 'joe' IDENTIFIED BY 'heslo';`
- `CREATE USER 'joe'@'localhost' IDENTIFIED BY 'heslo';`
 - Není-li počítač zadán, chápe se implicitní „cokoliv“ = '%'
- Účty je možné měnit, uzamknout či omezit pomocí `ALTER USER`
- `SET PASSWORD` – změna hesla; `DROP USER` – zrušení účtu

Oprávnění k databázi

Zvolené právo k danému objektu pro určitého uživatele přistupujícího z uvedené adresy zadáváme pomocí příkazu GRANT:

- GRANT *právo* ON *objekt* TO *uživatel*;

Příklady:

- GRANT SELECT ON db3518 TO '3518'@'svi-polacek.econ.muni.cz';
- GRANT SELECT, INSERT ON mysql.user TO student;
- GRANT ALL PRIVILEGES ON *.* TO 'admin'@'localhost'
IDENTIFIED BY PASSWORD '*2470C0C06D...' WITH GRANT OPTION;

Oprávnění lze odebrat pomocí příkazu REVOKE.

Uživatelské role

Uživatelské role umožňují lépe spravovat oprávnění pro skupinu uživatelů:

- `CREATE ROLE student;` – vytvoření role
- `GRANT student TO '3518'@'%';` – přiřazení uživatele k roli
- `SET ROLE student;` – „zapnutí“ zvolené role
- `SET ROLE none;` – „vypnutí“ uživatelské role

Automatické přiřazení zvolené role k uživateli při jeho připojení k databázi lze zajistit příkazem `SET DEFAULT ROLE`.

Roli lze smazat podobně jako uživatele: `DROP ROLE název;`

Cvičení II

1. Zjistěte informace o svém uživatelském účtu a o svých oprávněních v systému pomocí příkazů `SHOW CREATE USER`, `SHOW GRANTS` a `SELECT CURRENT_ROLE`.
2. Změňte si heslo na nové a pak zpět na původní pomocí `SET PASSWORD = PASSWORD(' noveheslo ');`
3. Zapněte si roli `student`, vytvořte nového uživatele a dejte mu oprávnění pro aktualizaci řádků tabulek ve vaší databázi.
4. Vytvořte novou roli a oprávnění používat ji vybranými spolužáky.
 - Všechny uživatele zjistíte pomocí `SELECT User FROM mysql.user;`
 - Všechna možná oprávnění lze vypsát pomocí `SHOW PRIVILEGES;`

Příkazy SHOW a SET

- SHOW vypisuje mnohé užitečné informace
 - SHOW CREATE TABLE / DATABASE / VIEW / USER
 - SHOW TABLES / DATABASES
 - SHOW STATUS
 - SHOW WARNINGS
 - SHOW VARIABLES
- SET nastavuje vybrané prvky databáze, zejména proměnné
 - SET *proměnná* = *hodnota*
 - SET ROLE
 - SET PASSWORD

Národní abecedy v SQL

Většina databázových systémů podporuje volbu vlastní znakové sady a metody řazení položek (collation):

- `SHOW CHARACTER SET; SHOW COLLATION;`
- `SHOW variables LIKE 'c%'; SET NAMES 'utf8mb4';`

Sadu a metodu lze definovat na úrovni databáze, tabulky či sloupce:

- `CREATE TABLE t1 (...) CHARACTER SET = 'ucs2' COLLATE = 'ucs2_czech_ci';`
- `CREATE TABLE t2 (s11 TEXT CHARACTER SET 'cp1250', s12 VARCHAR(9) COLLATE 'cp1250_czech_cs', s13 MEDIUMTEXT COLLATE 'latin1_swedish_nopad_ci');`

Možnosti „přetypování“ přímo v dotazech:

- `SELECT CAST(jmeno AS CHAR CHARACTER SET 'latin2') FROM ...`
- `SELECT jmeno FROM lide ORDER BY prijmeni COLLATE 'utf8mb4_czech_ci';`

Další národní funkce

- Názvy dnů a měsíců – proměnná *lc_time_names*:
 - `SET lc_time_names = 'cs_CZ' ('sk_SK')`
- Chybová hlášení v jiném jazyce – proměnná *lc_messages*:
 - `SET lc_messages='cs_CZ';`
- Formátování čísel (desetinná čárka, oddělovač tisíců) – `FORMAT`:
 - `FORMAT(123456789.0154, 2, 'cs_CZ');`
- Přirozené řazení – funkce `NATURAL_SORT_KEY`:
 - `SELECT call_no FROM knihy ORDER BY NATURAL_SORT_KEY(call_no);`
 - *Tato funkce je v MariaDB dostupná až od verze 10.7.0 :-)*

Cvičení III

1. Zapněte si jazyk chybového hlášení do svého rodného jazyka.
2. Vypište tituly z tabulky knihy začínající na C nebo Ā seřazené dle české abecedy (bez duplicit).