

Nadzor sočasnosti v MySQL/MariaDB



Kaj si bomo pogledali?

- Transakcije
- Transakcijski ukazi
 - Sintaksa
 - Semantika

Definicija transakcije

- Transakcija je operacija ali **niz operacij**, ki berejo ali pišejo v podatkovno bazo in so izvedene s strani enega uporabnika oziroma uporabniškega programa.
- Razvijalec določi, katere operacije tvorijo transakcijo (primer: bančni prenos sredstev med računi)
- Transakcija je logična enota dela – lahko je cel program ali samostojen ukaz (npr. INSERT ali UPDATE)
- Izvedba uporabniškega programa je s stališča podatkovne baze vidna kot ena ali več transakcij.

ACID lastnosti transakcij

- Atomarnost (atomicity): transakcija predstavlja atomaren sklop operacij. Ali se izvede vse ali nič. Atomarnost mora zagotavljati SUPB.
- Konsistentnost (consistency): transakcija je sklop operacij, ki podatkovno bazo privede iz enega konsistentnega stanja v drugo. Zagotavljanje konsistentnosti je naloga SUPB (zagotavlja, da omejitve nad podatki niso kršene...) in programerjev (preprečuje vsebina neskladnosti).
- Izolacija (isolation): transakcije se izvajajo neodvisno ena od druge → delni rezultati transakcije ne smejo biti vidni drugim transakcijam. Za izolacijo skrbi SUPB.
- Trajnost (durability): učinek potrjene transakcije je trajen – če želimo njen učinek razveljaviti, moramo to narediti z novo transakcijo, ki z obratnimi operacijami podatkovno bazo privede v prvotno stanje. Zagotavljanje trajnosti je naloga SUPB.

Zaklepanje v MySQL

- Primer: MySQL nudi več različnih domorodnih (native) tipov datotečne organizacije (podobno MariaDB)

MySQL:

- MyISAM
- MERGE
- FEDERATED
- ARCHIVE
- **InnoDB** (privzeto)
- ...

MariaDB:

- ARIA
- MERGE
- **FEDERATEDX**
- ARCHIVE
- **XTRADB** (privzeto)
- **TOKUDB**

- Kriteriji **izbire**: podpora transakcijam, zrnatost zaklepanja (vrstice/tabele/zapisi), hitrost, varnost
 - <http://www.developer.com/db/article.php/2235521/Pros-and-Cons-of-MySQL-Table-Types.htm>
 - <https://mariadb.com/kb/en/mariadb/show-engines/>
 - <http://dev.mysql.com/doc/refman/5.7/en/show-engines.html>

Transakcije v SQL

- SQL vsebuje mehanizme za uporabo in (delno) nadzor upravljanja s transakcijami
- **Kako** uporabljamo SQL mehanizme za podporo transakcijam
 - Začetek in konec transakcije
 - Stopnje izolacije
 - Dodatki ukazom
 - Preverjanje omejitev

Transakcije v SQL

- SQL definira transakcijski model z ukazoma COMMIT in ROLLBACK.
- Razširitve:
 - Microsoft Transact-SQL (BEGIN TRANSACTION),
 - MySQL: START TRANSACTION / BEGIN
 - Ponekod klici internih funkcij
- Spremembe, ki so narejene znotraj poteka transakcije, načeloma niso vidne navzven drugim transakcijam, dokler transakcija ni končana.

Način autocommit (MySQL, MariaDB)

- Privzet način delovanja v seji (session) MySQL ali MariaDB
- Princip: en ukaz = ena transakcija
- Vkllop/izklop autocommit:
 - SQL:
SET autocommit = 0 ali 1
 - pyODBC:
cnxn = pyodbc.connect(conn_str, **autocommit=False**)
 - Transakcijski ukazi:
START TRANSACTION # implicitno izklopi autocommit
...
COMMIT ali ROLLBACK # ponovno vklopi autocommit

Transakcije v SQL

- Transakcija se lahko zaključi na enega od štirih načinov:
 - Transakcija se uspešno zaključi s COMMIT; spremembe so permanentne.
 - Transakcija se prekine z ROLLBACK; spremembe, narejene s transakcijo, se razveljavijo.
 - Program, znotraj katerega se izvaja transakcija, se uspešno konča. Transakcija je potrjena implicitno (brez COMMITa).
 - Program, znotraj katerega se izvaja transakcija, se ne konča uspešno. Transakcija se implicitno razveljavi (brez ROLLBACKa).

Transakcije v SQL

- Nova transakcija se začne z novim SQL stavkom, ki transakcijo začne (prvi stavek, za BEGIN/START TRANSACTION, za COMMIT ali ROLLBACK).
- SQL transakcij ne moremo gnezditi.
- Transakcijske nastavitve upravljamo s pomočjo ukaza SET TRANSACTION

SET TRANSACTION

[READ ONLY | READ WRITE] |

[ISOLATION LEVEL

READ UNCOMMITTED | READ COMMITTED |
REPEATABLE READ | SERIALIZABLE]

Transakcije v SQL

- READ ONLY – pove, da transakcija vključuje samo operacije, ki iz baze berejo.
 - SUPB bo dovolil INSERT, UPDATE in DELETE samo nad začasnimi tabelami.
- ISOLATION LEVEL – pove stopnjo interakcije, ki jo SUPB dovoli med to in drugimi transakcijami.
- MySQL:
SET [GLOBAL | SESSION] TRANSACTION

GLOBAL: globalno

SESSION: znotraj iste povezave

Brez: le za naslednjo transakcijo

Transakcije v SQL

- Učinek SET TRANSACTION ISOLATION LEVEL

	Branje neobstoječega podatka	Nekonsistentna analiza	Fantomsko branje	Izgubljeno ažuriranje
Read Uncommitted	D	D	D	D
Read Committed	N	D	D	D
Repeatable Read	N	N	D	N
Serializable	N	N	N	N

(fantomsko branje je sorodno nekonsistentni analizi: ob kasnejši izvedbi neke poizvedbe znotraj transakcije se pojavijo **nove** vrstice, ki jih ob prejšnjih izvedbah iste poizvedbe znotraj iste transakcije ni bilo)

Takojšnje in zapoznele omejitve...

- Včasih želimo, da se omejitve ne bi upoštevale takoj, po vsakem SQL stavku, temveč ob zaključku transakcije.
- Omejitve lahko definiramo kot
 - INITIALLY IMMEDIATE – ob začetku transakcije;
 - INITIALLY DEFERRED – ob zaključku transakcije.
- Če izberemo INITIALLY IMMEDIATE (privzeta možnost), lahko določimo tudi, ali je zakasnitev moč določiti kasneje. Uporabimo [NOT] DEFERRABLE.

Takojšnje in zapoznele omejitve

- Način upoštevanja omejitev za trenutno transakcijo nastavimo z ukazom SET CONSTRAINTS.

SET CONSTRAINTS

{ALL | constraintName [, . . .]}

{DEFERRED | IMMEDIATE}

Transakcijski dodatki k SELECT stavku

- Pomagamo upravljalcu transakcij da pisalno ali bralno zaklene prebrani podatek, ne glede na nivo izolacije
- `SELECT ... FOR UPDATE;` -- na koncu SELECT stavka vse prebrane vrstice zaklene pisalno (ekskluzivno)
- `SELECT ... LOCK IN SHARE MODE;` -- na koncu SELECT vse prebrane vrstice zaklene bralno (deljeno)
- tovrstno zaklepanje **ni** odvisno od ISOLATION LEVEL, upoštevanje teh zaklepanj pa **je**