DBMS Lab 11
2019

General Instruction

1. [P] marked questions are for your practice, need not be submitted. There are also some practice problems at the end. [B] marked questions are bonus. You will get 1 mark extra for bonus.
2. Make a single pdf file using screen shots. Work out the questions in the order given and also arrange them in the same order in the submitted pdf.
3. Error messages are also output, we need to check if you are getting correct error messages or not.
4. blue part for your knowledge. No need to write them in report.
5. Use MariaDB and university database.

---

1. Transaction management
    a. Study system variable 'autocommit' and 'in_transaction' [what value they can have]
    b. Report current values of these two system variables. What does it means.
    c. If autocommit is ON then set it OFF and verify the change. Use transactions (create, insert, update) on student table to demonstrate. For demonstration, show outcome before and after autocommit is OFF.
    d. Write a transaction containing following tasks, and commit after that
        i. Create a Table *avg_sal with* one column *asalary (numeric)*
        ii. Update avg_sal.asalary = average(instructors.salary)
    e. Redo 'd.' (drop Table *avg_sal*)but before it commit, check the system variable *in_transaction* value. What does this variable store?
    f. Write a transaction which is rolled back before commit
        i. Execute the following statements
            1. start transaction
            2. create a Table *avg_sal_rollback with* one column *asalary*
            3. insert Avg_sal_rollback.asalary = average(instructors.salary)
            4. rollback
        ii. Report if '*avg_sal_rollback'* has been created and what value *asalary* has. Does it match with your expectation? If not read about implicit commit.
        iii. Which kinds of statements have implicit commit even when autocommit mode is off?
2. Transaction isolation
    a. State the different kind of isolations in MariaDB.

   b. What is current isolation method in use (report all three types of transaction isolation levels, i.e, global, session and current)
   c. Execute the following statements and report effect of each of them
      1. *`start transaction;`*
      2. *`set global transaction isolation level serializable;`*
      3. *`set transaction isolation level serializable;`*
      4. *`commit;`*
   d. Validate transaction isolation
      i. Create user ‘<yourname>_dummy’ with all privileges in your database.
      ii. Check if table avg_sal exists. If not, create it *with* one column *asalary (numeric)*
      iii. *Confirm that the current and global transaction isolation level is* SERIALIZABLE.
      iv. You as your old *username* start the following transaction
         1. `start transaction`
         2. `insert into avg_sal values (2);`
      v. Open another session of MariaDB and login as user ‘<yourname>_dummy’ and execute following statement
         1. `alter table avg_sal add column b int;`
      vi. Report your observation.
      vii. Find out how long ‘<yourname>_dummy’ needs to wait if other user forgot to complete the current transaction?
3. There is a table in MariaDB which shows active metadata locks. The table will be empty if there are no active metadata locks.
   a. Table can be read by using following plugins
      1. `install  SONAME 'metadata_lock_info';`
      2. `Select * from information_schema.metadata_lock_info;`
   b. Write an SQL statement to show who is locking whom.[hints: Need to know the schema of both table PROCESSLIST P and METADATA_LOCK_INFO]
4. Locking a table.
   a. Create 2 users A and B with a privilege for “lock table” also study various kind of privilege one user may have.
   b. Login as A
      i. Lock table student with option ‘wait’. [lock table is a command to put a lock in table]
   c. Login as B
      i. Lock table instructor
      ii. Insert a new data in student . What happens and why ?
   d. As ‘A’ insert a new data in instructor. What happens and why?
   e. As ‘B’ insert a new data in student. What happens and why ?
5. DeadLock

f. Execute the following statement and report the outcome and also the causes

   i. Open session 1: login as ‘A’

      ----

      1. set global innodb_deadlock_detect=OFF . Study what is “innodb_deadlock_detect”
      2. set global innodb_print_all_deadlocks=ON. Study the variable.
      3. set innodb_lock_wait_timeout =120; [Study it]
      4. create table dl1(pk int primary key, data varchar(100));
      5. create table dl2(pk int primary key, pk1 int not null, constraint dl2_fk foreign key(pk1) references dl1(pk), data varchar(100));
      6. set autocommit=off;
      7. insert into dl1 values(1, 'a');

   ii. Open session 2: login as ‘B’

      ----

      1. set autocommit=off;
      2. insert into dl2(pk, pk1, data) values(10, 1, 'a0'); [What happens and why?]

   iii. Go back to session 2:

      ----

      1. insert into dl2(pk, pk1, data) values(10, 1, 'a0'); [What happens and why?]

   iv. Study how to see current deadlocks