Selected Topics - Part II - Database Transactions

Hui Chen [a]

[a]CUNY Brooklyn College, Brooklyn, NY, USA

May 5, 2022

Outline

Outline

Reminder: Project Demo and Presentation

Are you ready for project demo and presentation?

Outline

Selected Topics

Discussed

- an introduction to database index

To discuss

- ACID and transactions
- Database in programming environment

Outline

Movitation

Often there is a need to group one or more database operations into a single unit of work.

Let's consider that we make a purchase online, and the design may like the following (next slide),

Movitation: Example 1

1. a customer places item 1 in the cart, the system saves the item information in the database
2. the customer places item 2 in the cart, the system saves the item information in the database
3. ...
4. the customer places item n in the cart, the system saves the item information in the database
5. the customer enters shipping address, the system saves the shipping address in the database
6. the customer enters payment card information, the system saves the payment card information
7. the customer confirms the purchase
8. the system then attempts to obtain the payment card authorization from the system operated by a bank
9. the system then recorded the purchase as successful, and changes the status of the purchase in the system.

If any one of these step fail, we may need to undo all of the steps before to ensure the consistency of the data.

These steps should be considered as a single unit of work that should be completely altogether or none at all.

Motivation: Example 2

Let's consider another example – transfer $100 from Zelle account 1234 to account 4321 , which requires three steps with regard to database operations

```
-- Step 1
SELECT balance FROM Account WHERE acct_no='1234';

-- Step 2
UPDATE Account SET balance = balance - 100
WHERE acct_no='1234'

-- Step 3
UPDATE Account SET balance = balance + 100
WHERE acct_no='4321'
```

What if we somehow fail at step 3?

Motivation: Example 2

Let's consider another example – transfer $100 from Zelle account 1234 to account 4321 , which requires three steps with regard to database operations

```
-- Step 1
SELECT balance FROM Account WHERE acct_no='1234';

-- Step 2
UPDATE Account SET balance = balance - 100
WHERE acct_no='1234'

-- Step 3
UPDATE Account SET balance = balance + 100
WHERE acct_no='4321'
```

What if we somehow fail at step 3?

These steps should be considered as a single unit of work that should be completely altogether or none at all.

Motivation: Example 3

Let's consider another example – two persons withdraw $100 from a joint bank account 1234 from two ATMs, which requires to run these database operations from the two ATMs.

```
-- ATM 1 Step 1
SELECT balance FROM Account WHERE acct_no='1234';

-- ATM 1 Step 2
UPDATE Account SET balance = balance - 100 WHERE acct_no='1234'

-- ATM 2 Step 1
SELECT balance FROM Account WHERE acct_no='1234';

-- ATM 2 Step 2
UPDATE Account SET balance = balance - 100 WHERE acct_no='1234'
```

What if the system run these operations in these order ATM 1 Step 1, ATM 2 Step 1, ATM 1 Step 2, ATM 2 Step 2?

Motivation: Example 3

Let's consider another example – two persons withdraw $100 from a joint bank account 1234 from two ATMs, which requires to run these database operations from the two ATMs.

```
-- ATM 1 Step 1
SELECT balance FROM Account WHERE acct_no='1234';

-- ATM 1 Step 2
UPDATE Account SET balance = balance - 100 WHERE acct_no='1234'

-- ATM 2 Step 1
SELECT balance FROM Account WHERE acct_no='1234';

-- ATM 2 Step 2
UPDATE Account SET balance = balance - 100 WHERE acct_no='1234'
```

What if the system run these operations in these order ATM 1 Step 1, ATM 2 Step 1, ATM 1 Step 2, ATM 2 Step 2?

ATM 1 Steps 1 - 2 should be a single unit of work and ATM 2 Steps 1 - 2 should also be a single unit of work.

Transaction

A database transaction is a unit of work that consists of one or more database operations and the unit of work must be executed in isolation, in all or none at all.

ACID

Properly implemented transactions are commonly said to meet the "ACID test"

- Atomicity – the all-or-nothing execution of transactions.
- Isolation – the fact that each transaction must appear to be executed as if no other transaction is executing at the same time.
- Durability – the condition that the effect on the database of a transaction must never be lost, once the transaction has completed.
- Consistency – transactions are expected to preserve the consistency of the database

Declaring Transactions in SQL

SQL allows the programmer to group several statements into a single transaction.

- `START TRANSACTION` marks the beginning of a transaction.

There are two ways to end a transaction:

- `COMMIT` causes the transaction to end successfully
  - All changes to the database were caused by the SQL statement or statements since the current transaction began are installed permanently in the database.
  - Before this, changes are tentative and may or may not be visible to other transactions.
- `ROLLBACK` causes the transaction to abort, or terminate unsuccessfully.
  - Any changes made in response to the SQL statements of the transaction are undone.

Autocommit

When using the generic SQL interface (e.g., mysql), each statement is a transaction by itself (so called the autocommit mode); however, this may or may not be true with other SQL interface.

For MariaDB, we can query the autocommit status

```
SELECT @@autocommit;
```

To turn off autocommit, use the SET command,

```
SET autocommit=0
```

To turn on autocommit, use the SET command,

```
SET autocommit=1
```

When autocommit is off, we must issue COMMIT to make changes permanent.

Declaring Transaction: Example 1

```
DELIMITER //

CREATE PROCEDURE WithdrawFrom(IN acctNo CHAR(5), IN amount INTEGER)
BEGIN
  DECLARE currentBalance INTEGER DEFAULT 0;

  START TRANSACTION;

  SELECT balance INTO currentBalance FROM Account WHERE acct_no=acctNo;

  IF currentBalance >= amount THEN
      UPDATE Account SET balance = balance - amount WHERE acct_no=acctNo;
  ELSE
      ROLLBACK;
      SELECT
        255 AS retcode,
        'Insufficient funds' AS retmsg,
        'Insufficient funds' AS retval;
  END IF;

  COMMIT;

END//

DELIMITER ;
```

Declaring Transaction: Example 2 (to be continued)

```
DELIMITER //
CREATE PROCEDURE TransferTo (IN fromAcctNo CHAR(5), IN toAcctNo CHAR(5), IN amount INTEGER)
BEGIN
  DECLARE currentBalance INTEGER DEFAULT 0;
  DECLARE errorCode INTEGER DEFAULT 0;
  DECLARE errorMsg VARCHAR(255) DEFAULT 'Query␣OK';

  START TRANSACTION;

  transfer_blk: BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
      BEGIN
        ROLLBACK;
        SELECT  errorCode AS retcode, errorMsg AS retmsg, errorMsg AS retval;
        RESIGNAL;
      END;

    SELECT balance INTO currentBalance FROM Account WHERE acct_no=fromAcctNo;
```

Continue to next slide

Declaring Transaction: Example 2 (continued)

Continued from previous slide

```
    IF currentBalance >= amount THEN
      UPDATE Account SET balance = balance - 100 WHERE acct_no=fromAcctNo;
      IF ROW_COUNT() = 0 THEN
        SET errorCode = 251, errorMsg = 'From Account not found';
        SIGNAL SQLSTATE '45000' SET MYSQL_ERRNO = errorCode, MESSAGE_TEXT = errorMsg;
      END IF;
      UPDATE Account SET balance = balance + 100 WHERE acct_no=toAcctNo;
      IF ROW_COUNT() = 0 THEN
        SET errorCode = 252, errorMsg = 'To Account not found';
        SIGNAL SQLSTATE '45000' SET MYSQL_ERRNO = errorCode, MESSAGE_TEXT = errorMsg;
      END IF;
    ELSE
        SET errorCode = 253, errorMsg = 'Insufficient funds';
        SIGNAL SQLSTATE '45000' SET MYSQL_ERRNO = errorCode, MESSAGE_TEXT = errorMsg;
    END IF;

    COMMIT;
  END;

END//
DELIMITER ;
```

For the standard SQLSTATE code, take a look at here.

Discussion Questions

Should we make transactions as big as possible?

Discussion Questions

Should we make transactions as big as possible?

- It is important to note transactions has a cost on throughput.
- DBMS often has tuning options – continue to explore this on your own or in a future database class.

Outline

Questions and Summary

- A brief introduction to database transactions

Outline

Assignment

Let's work on an assignment using paper and pencil/pen ...