
CS246: Database Management Systems Lab

Lab # 14 (1 Questions, 84 Points)

Held on 24-Apr-2023

Lab Timings: 14:00 to 18:00 Hours Pages: 3

Submission: 18:00 Hrs, 24-Apr-2023

Instructors Dr. V. Vijaya saradhi & Prof. Jatindra Kumar Deka

Head TAs Adithya Moorthy & Laxita Agrawal

Department of CSE, IIT Guwahati

- a. This lab assignment is based on the concepts covered on **SQL transaction management** in the CS245 theory class.
- b. You can refer to the text book for SQL syntax.

Question 1: (84 points)

SQL transaction management Implement the following three MySQL transaction problems

Task 01 (1 mark) Create a database named **week14a**

Task 02 (3 marks) Create the following tables under **week14a**:

- a. (1 mark) Create a table **account** whose description is given below:

1 st column	account_number	string of characters of fixed size 5
2 nd column	balance	integer
3 rd column	original_balance	integer

- b. (1 mark) Create a **move_funds** table whose description is given below:

1 st column	from_acc	string of characters of fixed size 5
2 nd column	to_acc	string of characters of fixed size 5
3 rd column	transfer_amount	integer

- c. (1 mark) Create a **move_funds_log** table whose description is given below:

1 st column	account_number	string of characters of fixed size 5
2 nd column	move_fund_type	string of characters of fixed size 10
3 rd column	amount	integer
4 th column	timestamp	date time

- **account_number** is a foreign key referring to table **account**.
- **move_fund_type** takes values from the set {'deposit', 'withdraw'}

Task 03 (2 marks) **populate data**

- a. (1 mark) Populate data from file **account.csv** into table **account**
- b. (1 mark) Populate data from file **trnx.csv** into table **move_funds**

Task 04 (2 marks) As a **root** user, create the following users

- a. **saradhi** with a chosen password

- b. `pbhaduri` with a chosen password

Task 05 (7 marks) As a `root` user, provide the following privileges to the above two users

- a. `Read, write, update and delete` privileges to
 - `account` table
 - `move_funds` table
 - `move_funds_log_1` table
- b. `Lock tables` privilege for the table `account`
- c. `Execute` privilege to the stored procedure `transfer_funds_1`¹
- d. `Execute` privilege to the stored procedure `main_transfer_2`²
- e. `Execute` privilege to the stored procedure `transfer_funds_2`³

Task 06 (15 marks) Perform the following sub-tasks. All the three sub-tasks 6 (a), 6 (b) and 6(c) should be running simultaneously.

- a. (5 marks) Run a `MySQL` client instance as `root` user and write a stored procedure `transfer_funds_1` whose specifications are
 - i. Input
 - i. `from_acc`
 - ii. `to_acc`
 - iii. `transfer_amount`
 - ii. output: none
 - iii. Perform funds moving from one account to another account the specified amount. You must update the `balance` column of `account` table.Note: `Execute` privilege to the stored procedure `transfer_funds_1` to users `saradhi` and `pbhaduri` at this stage.

- b. (5 marks) Run a `MySQL` client instance as `saradhi` user
 - i. `lock` the `account` table
 - ii. Invoke `transfer_funds_1()` using the first record of `move_funds` table
 - iii. `unlock` the table
- c. (5 marks) Run a `MySQL` client instance as `pbhaduri` user
 - i. `lock` the `account` table
 - ii. Invoke `transfer_funds_1()` using the first record of `move_funds` table
 - iii. `unlock` the table

Task 07 (20 marks) Create a stored procedure `transfer_funds_2` with the following specification:

¹Provide this privilege after creating this stored procedure at appropriate place

²Provide this privilege after creating this stored procedure at appropriate place

³Provide this privilege after creating this stored procedure at appropriate place

- a. Input arguments
 - `from_acc` string of characters of fixed size 5 representing from account number
 - `to_acc` string of characters of fixed size 5 representing to account number
 - `transfer_amount` integer representing money being transferred
- b. Return output: none
- c. (20 marks) Perform the following steps:
 - i. Start a transaction
 - ii. Verify that the `from_acc` has a `balance` of at least 100. If not, `rollback`
 - iii. Deduct the `transfer_amount` from `balance` of `from_acc`
 - iv. Add `transfer_amount` to `balance` of `to_acc`.
 - v. Insert a record into the `move_funds_log` table about the transfer.

Task 08 (15 marks) Create a stored procedure `main_transfer_2` with the following specification

- a. Input arguments: None
- b. Output arguments: None
- c. (10 marks) Read one row at-a-time from table `move_funds` using a MySQL `CURSOR`. Get the input arguments.
- d. (5 marks) Call the stored procedure `transfer_funds_2` using input arguments from **task 08** (c)

Task 09 (10 marks) Perform the following sub-tasks. These two sub-tasks 8 (a) and 8 (b) should be running simultaneously.

- a. (5 marks) Run a MySQL client instance as `saradhi` user. Invoke `transfer_funds_2()`
- b. (5 marks) Run a MySQL client instance as `pbhaduri` user. Invoke `transfer_funds_2()`

Task 10 (9 marks) Integrity check:

- a. (3 marks) Compute total `withdraw` amount on each `account_number`. Name this column as `total_withdraw`
- b. (3 marks) Compute total `deposit` amount on each `account_number`. Name this column as `total_deposit`
- c. (3 marks) `original_balance` should be equal to `balance + total_deposit - total_withdraw`

SQL statements Write the SQL statements corresponding to each task in a text.

File naming text file name should be [Your roll number].sql

Independent efforts You should make an honest and independent effort in obtaining the solution to the above problem.

Mobile phones are not allowed inside the lab