

### Question for ODD numbers

Modify the bank experiment(exp.No: 7) to implement following.

1. Implement a procedure for fund transfer

The parameters are `source_accountNo`, `destination_accountNo`, and `amount`.

It should check all possibilities of errors including the validity of account number.

2. Create a new table `transactionLog` with following fields to record the entries of all debit and credit operations.

`trans_id, acc_no, tran_date, tran_type, tran_amount, remarks`

`trans_id` is the primary key and it must incremented automatically (use `AUTO_INCREMENT`).

`tran_date` should be data type date and is same as the system date [use function `curdate()`].

`tran_type` possible values are `credit` & `debit`.

`acc_no` is a NOT a foreign key to `cust_details` table.

All credit/debit procedure should have an appropriate entry in the transaction table. a fund transfer should also have a debit and credit operations.

3. Write a function to get the number of credits operation performed on an account.

`cust_details`

acc_no	acc_type	name	address	balance
101	saving	hari	KSD	3000
102	saving	raj	PKD	5000
201	current	sree	EKM	6000
202	current	john	KSD	7000

`credit(101,2000); SUCCESS`

`credit(102,1000); SUCCESS`

`credit(103,2000); ERROR`

`cust_details`

acc_no	acc_type	name	address	balance
101	saving	hari	KSD	5000
102	saving	raj	PKD	6000
201	current	sree	EKM	6000
202	current	john	KSD	7000

transactionLog

trans_id	acc_no	tran_date	tran_type	tran_amount	remarks
1	101	13-11-2019	credit	1000	Success
2	102	13-11-2019	credit	1000	Success
3	103	13-11-2019	credit	2000	ERROR

debit(101,4500);    ERROR  
debit(102,1000);    SUCCESS  
debit(201,6000);    SUCCESS  
debit(203,6000);    ERROR

cust\_details

acc_no	acc_type	name	address	balance
101	saving	hari	KSD	5000
102	saving	raj	PKD	5000
201	current	sree	EKM	0
202	current	john	KSD	7000

transactionLog

trans_id	acc_no	tran_date	tran_type	tran_amount	remarks
1	101	13-11-2019	credit	1000	Success
2	102	13-11-2019	credit	1000	Success
3	103	13-11-2019	credit	2000	ERROR
4	101	13-11-2019	debit	4500	ERROR
5	102	13-11-2019	debit	1000	Success
6	201	13-11-2019	debit	6000	Success
7	203	13-11-2019	debit	6000	ERROR

fundTransfer(202,102,2000);

cust\_details

acc_no	acc_type	name	address	balance
101	saving	hari	KSD	5000
102	saving	raj	PKD	7000
201	current	sree	EKM	0
202	current	john	KSD	5000

transactionLog

trans_id	acc_no	tran_date	tran_type	tran_amount	remarks
1	101	13-11-2019	credit	1000	Success
2	102	13-11-2019	credit	1000	Success
3	103	13-11-2019	credit	2000	ERROR
4	101	13-11-2019	debit	4500	ERROR
5	102	13-11-2019	debit	1000	Success
6	201	13-11-2019	debit	6000	Success
7	203	13-11-2019	debit	6000	ERROR
8	202	13-11-2019	debit	2000	Success
9	102	13-11-2019	credit	2000	Success

fundTransfer(102,101,1000);

cust\_details

acc_no	acc_type	name	address	balance
101	saving	hari	KSD	6000
102	saving	raj	PKD	6000
201	current	sree	EKM	0
202	current	john	KSD	5000

transactionLog

trans_id	acc_no	tran_date	tran_type	tran_amount	remarks
1	101	13-11-2019	credit	1000	Success
2	102	13-11-2019	credit	1000	Success
3	103	13-11-2019	credit	2000	ERROR
4	101	13-11-2019	debit	4500	ERROR
5	102	13-11-2019	debit	1000	Success
6	201	13-11-2019	debit	6000	Success
7	203	13-11-2019	debit	6000	ERROR
8	202	13-11-2019	debit	2000	Success
9	102	13-11-2019	credit	2000	Success
10	102	13-11-2019	debit	1000	Success
11	101	13-11-2019	credit	1000	Success

fundTransfer(102,103,1000);    ERROR

fundTransfer(103,101,1000);    ERROR

countCredit(102)

2

countCredit(103)

1

### Question for EVEN numbers

Modify the bank experiment (exp.No: 7) to implement following.

Modify the `customer_details` table to include an additional value `loan` under `acc_type` field.(inaddtion to saving/current account)

Create a table `loan` with following Fields.

`loan_acc_no, cust_acc_no, loan_amount, status`  
`cust_acc_no`, is the primary key and is also a foreign key to `acc_no` field in `cust_details` table.  
`loan_amount` is field to indicate the amount of loan (it's a constant field)  
`status` field has only two values `active`, `closed`.  
`cust_acc_no` is a foreign key to `acc_no` field in `cust_details` table

1. Implement a procedure `loan_payment` for crediting the amount ONLY for the loan account. (It should check the validity of account also the type account)

**Restrict the procedure credit and debit only for saving or current account.**

2. Develop a trigger to change the `status` of the loan to `closed` if the `balance` amount in the loan account is greater than or equal to `loan_amount` in the `loan` table.
3. Create a new table `transactionLog` with following fields to record the entries of all `debit`, `credit` and `loan_payment` operations

`trans_id, acc_no, tran_date, tran_type, tran_amount, remarks`

`trans_id` is the primary key and it must incremented automatically (use `AUTO_INCREMENT`).  
`tran_date` should be data type date and is same as the system date [use function `curdate()`].  
`tran_type` possible values are `credit`, `debit` and `loan_payment`.  
`acc_no` is a NOT a foreign key to `cust_details` table.

4. Write a function to get the number of loans under one an account.

`cust_details`

<code>acc_no</code>	<code>acc_type</code>	<code>name</code>	<code>address</code>	<code>balance</code>
101	saving	hari	KSD	3000
201	current	sree	EKM	6000
301	loan	hari	KSD	100000
302	loan	hari	KSD	200000
303	loan	sree	KSD	300000

loan			
loan_acc_no	cust_acc_no	loan_amount	status
301	101	0	Active
302	101	0	Active
303	201	0	Active

credit(101,2000); SUCCESS  
credit(103,2000); ERROR  
credit(301,1000); ERROR

cust\_details

acc_no	acc_type	name	address	balance
101	saving	hari	KSD	5000
201	current	sree	EKM	6000
301	loan	hari	KSD	100000
302	loan	hari	KSD	200000
303	loan	sree	KSD	300000

transactionLog

trans_id	acc_no	tran_date	tran_type	tran_amount	remarks
1	101	13-11-2019	credit	1000	Success
2	103	13-11-2019	credit	2000	ERROR
3	301	13-11-2019	credit	1000	ERROR

debit(101,4500); ERROR  
debit(101,1000); SUCCESS  
debit(201,6000); SUCCESS  
debit(301,6000); ERROR  
debit(202,6000); ERROR

cust\_details

acc_no	acc_type	name	address	balance
101	saving	hari	KSD	4000
201	current	sree	EKM	0
301	loan	hari	KSD	100000
302	loan	hari	KSD	200000
303	loan	sree	KSD	300000

transactionLog

trans_id	acc_no	tran_date	tran_type	tran_amount	remarks
1	101	13-11-2019	credit	1000	Success
2	103	13-11-2019	credit	2000	ERROR
3	301	13-11-2019	credit	1000	ERROR
4	101	13-11-2019	debit	4500	ERROR
5	101	13-11-2019	debit	1000	Success
6	201	13-11-2019	debit	6000	Success
7	301	13-11-2019	debit	6000	ERROR
8	202	13-11-2019	debit	6000	ERROR

loanpayment(101,2000);           Error  
loanpayment(201,2000);           Error  
loanpayment(202,2000);           Error  
loanpayment(301,10000);          success  
loanpayment(302,20000);          success  
**cust\_details           NO changes**

loan			
loan_acc_no	cust_acc_no	loan_amount	status
301	101	10000	Active
302	101	20000	Active
303	201	0	Active

transactionLog

trans_id	acc_no	tran_date	tran_type	tran_amount	remarks
1	101	13-11-2019	credit	1000	Success
2	103	13-11-2019	credit	2000	ERROR
3	301	13-11-2019	credit	1000	ERROR
4	101	13-11-2019	debit	4500	ERROR
5	101	13-11-2019	debit	1000	Success
6	201	13-11-2019	debit	6000	Success
7	301	13-11-2019	debit	6000	ERROR
8	202	13-11-2019	debit	6000	ERROR
9	101	13-11-2019	loan_payment	2000	ERROR
10	201	13-11-2019	loan_payment	2000	ERROR
11	202	13-11-2019	loan_payment	2000	ERROR
12	301	13-11-2019	loan_payment	10000	Success
13	302	13-11-2019	loan_payment	20000	Success

loanpayment(301,90000);  
loanpayment(301,180000);

loan

loan_acc_no	cust_acc_no	loan_amount	status	
301	101	100000	Closed	
302	101	200000	Closed	
303	201	0	Active	

transactionLog

trans_id	acc_no	tran_date	tran_type	tran_amount	remarks
1	101	13-11-2019	credit	1000	Success
2	103	13-11-2019	credit	2000	ERROR
3	301	13-11-2019	credit	1000	ERROR
4	101	13-11-2019	debit	4500	ERROR
5	101	13-11-2019	debit	1000	Success
6	201	13-11-2019	debit	6000	Success
7	301	13-11-2019	debit	6000	ERROR
8	202	13-11-2019	debit	6000	ERROR
9	101	13-11-2019	loan_payment	2000	ERROR
10	201	13-11-2019	loan_payment	2000	ERROR
11	202	13-11-2019	loan_payment	2000	ERROR
12	301	13-11-2019	loan_payment	10000	Success
13	302	13-11-2019	loan_payment	20000	Success
14	301	13-11-2019	loan_payment	90000	Success
15	302	13-11-2019	loan_payment	180000	Success

countloan(101)

2

countloan(201)

1