

**Bank Account Management System**

The database focuses on customer registration, maintaining different accounts of the customer, perform credit, debit and withdraw transactions on these accounts. The accounts are of two type (Loan, Deposit) which are divided into subtypes. The database also maintains credit and debit card details, nominees and joint account holder details. Techniques like Trigger, procedure, view, event, cursors are used.

**ER Diagram:**

Please refer “DB\_Project\_EER\_Diagram” file to see ER –Diagram properly.

**Trigger, procedure, view, events :**

Please refer “project\_script.sql” file for the code of Event, Trigger, View, and Procedure that are used.

**Event:**

1. reset\_tranId

This Event is used to reset AUTO\_INCREMENT used in transactionid of transaction table on daily basis.

```
CREATE EVENT reset_tranId
ON SCHEDULE
  EVERY 1 DAY
  STARTS (TIMESTAMP(CURRENT_DATE) + INTERVAL 1 DAY + INTERVAL 1 SECOND)
DO
  Alter table mydb.transaction AUTO_INCREMENT=0;
```

### Trigger:

#### 1. tg\_transaction

This trigger is used to take back up transaction table into another database whenever an insert occurs on transaction table.

**Note :** On similar basis we can create triggers on update and insert on all the tables used and thus have a backup on another database.

```
create trigger tg_transaction
after insert on mydb.transaction
for each row
insert into myddb.transaction values (New.transactionid,New.TransactionDate,
New.TransactionType_TransactionType,
New.TransactionStatus_TransactionStatus,
New.CreditAccount_AccountNum,
New.CreditAccount_Branch_RoutingNum,
New.DebitAccount_AccountNum,
New.DebitAccount_Branch_RoutingNum,
New.TransactionAmount,New.Description,New.Balance,New.ProcessDate);
```

### View:

#### 1. v\_customer\_dep\_accounts

It made using customer, depositaccount and transaction tables.

```
create view v_customer_dep_accounts as
select * from customer c left join depositaccount d
on c.CustomerNum = d.Customer_CustomerNum left join transaction t
on ((d.Account_AccountNum = t.creditAccount_accountnum and
d.Account_Branch_RoutingNum = t.creditAccount_Branch_RoutingNum) or
(d.Account_AccountNum = t.debitAccount_accountnum and
d.Account_Branch_RoutingNum = t.debitAccount_Branch_RoutingNum));
```

### Procedure:

#### 1. sp\_depositTransaction

This procedure is used for deposit transactions of accounts.

It internally triggers tg\_transaction when this deposit transaction details are inserted into Trasaction table.

**Please refer “project\_script.sql” for code**

Data before using this procedure

```

Administrator: C:\Windows\system32\cmd.exe - mysql -u root -p
mysql>
mysql> use mydb;
Database changed
mysql> select * from account;
+-----+-----+-----+-----+-----+-----+-----+
| AccountNum | Branch_RoutingNum | Sys | Status | OpenDate | Balance | CloseDate |
+-----+-----+-----+-----+-----+-----+-----+
| 00000000000000000001 | 000000001 | DEP | A | 2017-04-12 | 1000.000 | NULL |
| 00000000000000000002 | 000000002 | DEP | A | 2017-04-12 | 1000.000 | NULL |
| 00000000000000000003 | 000000003 | DEP | A | 2017-04-12 | 1000.000 | NULL |
| 00000000000000000004 | 000000001 | LON | A | 2017-12-12 | -48000.200 | NULL |
| 00000000000000000005 | 000000001 | LON | A | 2017-04-12 | -100500.000 | NULL |
| 00000000000000000006 | 000000002 | DEP | A | 2017-04-12 | 2000.000 | NULL |
+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select * from depositaccount;
+-----+-----+-----+-----+-----+-----+-----+
| Account_AccountNum | Account_Branch_RoutingNum | Customer_CustomerNum | DeposAccountType_AccountType | AvailableBalance | HoldBalance |
+-----+-----+-----+-----+-----+-----+-----+
| 00000000000000000001 | 000000001 | 00000000000000000001 | 0001 | 800.000 | 200.000 |
| 00000000000000000002 | 000000002 | 00000000000000000001 | 0002 | 1000.000 | 0.000 |
| 00000000000000000003 | 000000003 | 00000000000000000002 | 0001 | 1000.000 | 0.000 |
| 00000000000000000006 | 000000002 | 00000000000000000004 | 0002 | 2000.000 | 0.000 |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from branch;
+-----+-----+-----+-----+-----+-----+-----+
| RoutingNum | BranchName | Street | City | State | Country | Zipcode | BranchBalance |
+-----+-----+-----+-----+-----+-----+-----+
| 000000001 | x | Fenway | Boston | MA | United States | 02110 | 100000.000 |
| 000000002 | y | Huntington | Boston | MA | United States | 02114 | 100000.000 |
| 000000003 | z | Parker Hill | Boston | MA | United States | 02113 | 100000.000 |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from transaction;
Empty set (0.00 sec)

mysql> select * from mydb.transaction;
Empty set (0.02 sec)

mysql>

```

Calling sp\_depositTransaction;

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying a SQL script that calls the stored procedure `sp_depositTransaction`. The script includes a `commit` statement, a `select concat` statement for logging, and a `call sp_depositTransaction` statement with parameters `('00000000000000000001', 800, '000000001')`. The 'Result Grid' shows the output of the procedure call, indicating a successful transaction with a balance of 1800.000 and account number 00000000000000000001. The 'Output' tab at the bottom shows the execution log, confirming that the procedure was called successfully at 22:56:41 and 22:56:42, returning 1 row(s) each time.

```

-- Query 1
70 commit ;
71 end if;
72 select concat('Credit Transaction Successful and Balance is :',balance,' Account :',acct,'
73 from account where accountNum = acct and branch_routingNum=rout;
74
75 else
76 select 'invalid crediting account or crediting account is closed' result;
77 end if;
78 end
79
80 delimiter ;
81
82 call sp_depositTransaction('00000000000000000001',800,'000000001');
83

```

Result 38: result  
Credit Transaction Successful and Balance is :1800.000 Account :00000000000000000001 Sys :DEP

Output

#	Time	Action	Message	Duration / Fetch
441	22:56:41	call sp_depositTransaction('00000000000000000001',800,'000000001')	1 row(s) returned	0.000 sec / 0.000 sec
442	22:56:42	call sp_depositTransaction('00000000000000000001',800,'000000001')	1 row(s) returned	- / 0.000 sec

Data after executing sp\_depositTransaction

```
Administrator: C:\Windows\system32\cmd.exe - mysql -u root -p
mysql> select * from account;
+-----+-----+-----+-----+-----+-----+-----+
| AccountNum | Branch_RoutingNum | Sys | Status | OpenDate | Balance | CloseDate |
+-----+-----+-----+-----+-----+-----+-----+
| 0000000000000001 | 000000001 | DEP | A | 2017-04-12 | 1800.000 | NULL |
| 0000000000000002 | 000000002 | DEP | A | 2017-04-12 | 1000.000 | NULL |
| 0000000000000003 | 000000003 | DEP | A | 2017-04-12 | 1000.000 | NULL |
| 0000000000000004 | 000000001 | LON | A | 2017-12-12 | -48000.200 | NULL |
| 0000000000000005 | 000000001 | LON | A | 2017-04-12 | -100500.000 | NULL |
| 0000000000000006 | 000000002 | DEP | A | 2017-04-12 | 2000.000 | NULL |
+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select * from depositaccount;
+-----+-----+-----+-----+-----+-----+
| Account_AccountNum | Account_Branch_RoutingNum | Customer_CustomerNum | DeposAccountType_AccountType | AvailableBalance | HoldBalance |
+-----+-----+-----+-----+-----+-----+
| 0000000000000001 | 000000001 | 0000000000000001 | 0001 | 1600.000 | 200.000 |
| 0000000000000002 | 000000002 | 0000000000000001 | 0002 | 1000.000 | 0.000 |
| 0000000000000003 | 000000003 | 0000000000000002 | 0001 | 1000.000 | 0.000 |
| 0000000000000006 | 000000002 | 0000000000000004 | 0002 | 2000.000 | 0.000 |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from branch;
+-----+-----+-----+-----+-----+-----+-----+
| RoutingNum | BranchName | Street | City | State | Country | Zipcode | BranchBalance |
+-----+-----+-----+-----+-----+-----+-----+
| 000000001 | x | Fenway | Boston | MA | United States | 02110 | 100800.000 |
| 000000002 | y | Huntington | Boston | MA | United States | 02114 | 100000.000 |
| 000000003 | z | Parker Hill | Boston | MA | United States | 02113 | 100000.000 |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from transaction;
+-----+-----+-----+-----+-----+-----+-----+
| TransactionId | TransactionDate | TransactionType_TransactionType | TransactionStatus_TransactionStatus | CreditAccount_AccountNum |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | 2017-12-13 | 0001 | c | 0000000000000001 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Data of transaction table is replicated into myddb.transaction table due to trigger tg\_transaction

```
Administrator: C:\Windows\system32\cmd.exe - mysql -u root -p
mysql> select * from depositaccount;
+-----+-----+-----+-----+-----+-----+
| Account_AccountNum | Account_Branch_RoutingNum | Customer_CustomerNum | DeposAccountType_AccountType | AvailableBalance | HoldBalance |
+-----+-----+-----+-----+-----+-----+
| 0000000000000001 | 000000001 | 0000000000000001 | 0001 | 1600.000 | 200.000 |
| 0000000000000002 | 000000002 | 0000000000000001 | 0002 | 1000.000 | 0.000 |
| 0000000000000003 | 000000003 | 0000000000000002 | 0001 | 1000.000 | 0.000 |
| 0000000000000006 | 000000002 | 0000000000000004 | 0002 | 2000.000 | 0.000 |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from branch;
+-----+-----+-----+-----+-----+-----+-----+
| RoutingNum | BranchName | Street | City | State | Country | Zipcode | BranchBalance |
+-----+-----+-----+-----+-----+-----+-----+
| 000000001 | x | Fenway | Boston | MA | United States | 02110 | 100800.000 |
| 000000002 | y | Huntington | Boston | MA | United States | 02114 | 100000.000 |
| 000000003 | z | Parker Hill | Boston | MA | United States | 02113 | 100000.000 |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from transaction;
+-----+-----+-----+-----+-----+-----+-----+
| TransactionId | TransactionDate | TransactionType_TransactionType | TransactionStatus_TransactionStatus | CreditAccount_AccountNum |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | 2017-12-13 | 0001 | c | 0000000000000001 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from myddb.transaction;
+-----+-----+-----+-----+-----+-----+-----+
| TransactionId | TransactionDate | TransactionType_TransactionType | TransactionStatus_TransactionStatus | CreditAccount_AccountNum |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | 2017-12-13 | 0001 | c | 0000000000000001 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.02 sec)

mysql>
```

## 2. sp\_WithdrawTransaction

This procedure is used for withdraw transactions of accounts.

It internally triggers tg\_transaction when this withdraw transaction details are inserted into Transaction table.

**Please refer “project\_script.sql” for code**

Calling sp\_WithdrawTransaction

The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'SCHEMAS' tree with 'myddb' selected. The main editor window shows a SQL script with the following code:

```
141
142     select concat('Debit Transaction Successful and Balance is :',balance,' Account : ',acct,
143     from account where accountNum = acct and branch_routingNum=rout;
144     end if;
145
146     else
147     select 'invalid debiting account or debiting account is closed' result;
148     end if;
149
150 --%
151 delimiter ;
152
153 call sp_WithdrawTransaction('0000000000000001',800,'000000001');
154
```

The 'Result Grid' shows the output of the procedure call:

result
Debit Transaction Successful and Balance is :1000.000 Account :0000000000000001 Svs :DEP

The bottom 'Output' pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
445	22:58:01	call sp_WithdrawTransaction('0000000000000001',800,'000000001')	1 row(s) returned	- / 0.000 sec
446	22:58:01	call sp_WithdrawTransaction('0000000000000001',800,'000000001')	1 row(s) returned	- / 0.000 sec

Data after executing sp\_WithdrawTransaction

```
Administrator: C:\Windows\system32\cmd.exe - mysql -u root -p
mysql> select * from account;
```

AccountNum	Branch_RoutingNum	Sys	Status	OpenDate	Balance	CloseDate
0000000000000001	000000001	DEP	A	2017-04-12	1000.000	NULL
0000000000000002	000000002	DEP	A	2017-04-12	1000.000	NULL
0000000000000003	000000003	DEP	A	2017-04-12	1000.000	NULL
0000000000000004	000000001	LON	A	2017-12-12	-48000.200	NULL
0000000000000005	000000001	LON	A	2017-04-12	-100500.000	NULL
0000000000000006	000000002	DEP	A	2017-04-12	2000.000	NULL

6 rows in set (0.00 sec)

```
mysql> select * from depositaccount;
```

Account_AccountNum	Account_Branch_RoutingNum	Customer_CustomerNum	DeposAccountType_AccountType	AvailableBalance	HoldBalance
0000000000000001	000000001	0000000000000001	0001	800.000	200.000
0000000000000002	000000002	0000000000000001	0002	1000.000	0.000
0000000000000003	000000003	0000000000000002	0001	1000.000	0.000
0000000000000006	000000002	0000000000000004	0002	2000.000	0.000

4 rows in set (0.02 sec)

```
mysql> select * from branch;
```

RoutingNum	BranchName	Street	City	State	Country	Zipcode	BranchBalance
000000001	x	Fenway	Boston	MA	United States	02110	100000.000
000000002	y	Huntington	Boston	MA	United States	02114	100000.000
000000003	z	Parker Hill	Boston	MA	United States	02113	100000.000

3 rows in set (0.00 sec)

```
mysql> select * from transaction;
```

TransactionId	TransactionDate	TransactionType_TransactionType	TransactionStatus_TransactionStatus	CreditAccount_AccountNum
1	2017-12-13	0001	C	0000000000000001
2	2017-12-13	0003	C	NULL

Page: 3 of 3 Words: 366

Data of transaction table is replicated into myddb.transaction table due to trigger tg\_transaction

```
Administrator: C:\Windows\system32\cmd.exe - mysql -u root -p
4 rows in set (0.02 sec)
mysql> select * from branch;
```

RoutingNum	BranchName	Street	City	State	Country	Zipcode	BranchBalance
000000001	x	Fenway	Boston	MA	United States	02110	100000.000
000000002	y	Huntington	Boston	MA	United States	02114	100000.000
000000003	z	Parker Hill	Boston	MA	United States	02113	100000.000

3 rows in set (0.00 sec)

```
mysql> select * from transaction;
```

TransactionId	TransactionDate	TransactionType_TransactionType	TransactionStatus_TransactionStatus	CreditAccount_AccountNum
1	2017-12-13	0001	C	0000000000000001
2	2017-12-13	0003	C	NULL

2 rows in set (0.00 sec)

```
mysql> select * from myddb.transaction;
```

TransactionId	TransactionDate	TransactionType_TransactionType	TransactionStatus_TransactionStatus	CreditAccount_AccountNum
1	2017-12-13	0001	C	0000000000000001
2	2017-12-13	0003	C	NULL

2 rows in set (0.00 sec)

```
mysql>
```

Page: 6 of 3 Words: 366

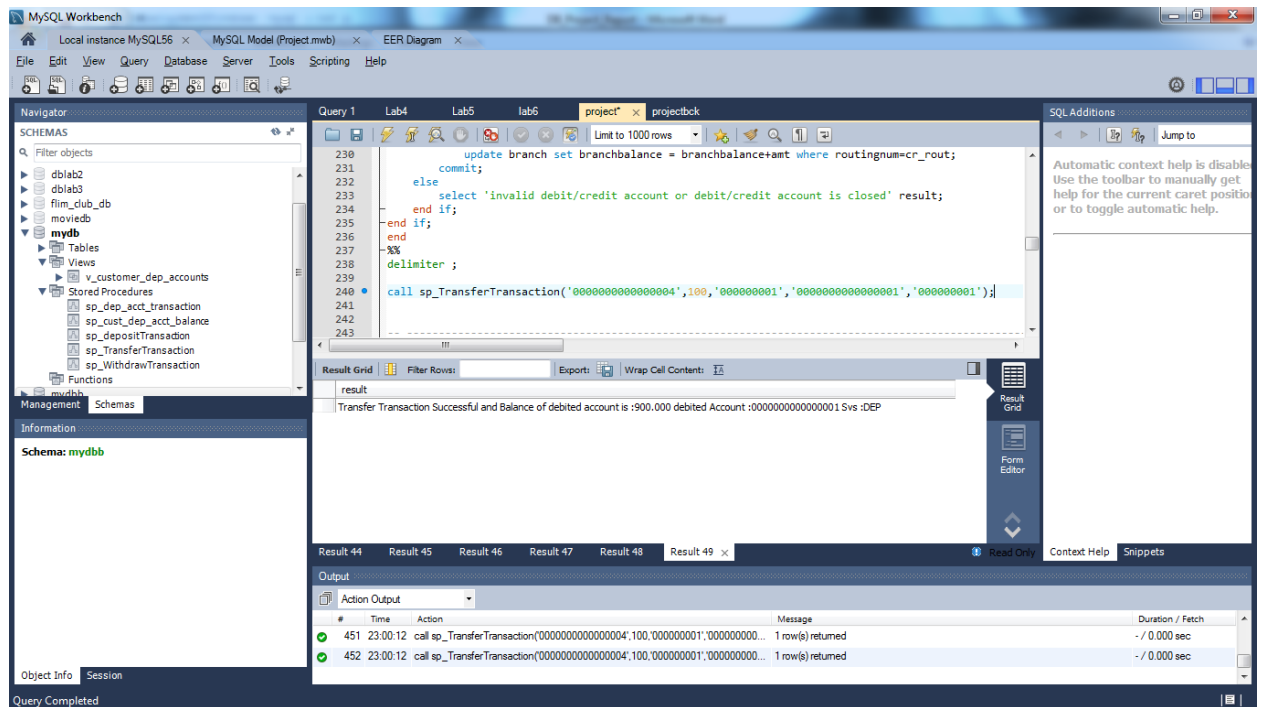
### 3. sp\_TransferTransaction

This procedure is used for transfer transactions between accounts.

It internally triggers tg\_transaction when this transfer transaction details are inserted into Trasaction table.

**Please refer “project\_script.sql” for code**

### Calling sp\_TransferTransaction



### Data after executing sp\_TransferTransaction

```
Administrator: C:\Windows\system32\cmd.exe - mysql -u root -p

mysql> select * from account;
+-----+-----+-----+-----+-----+-----+-----+
| AccountNum | Branch_RoutingNum | Sys | Status | OpenDate | Balance | CloseDate |
+-----+-----+-----+-----+-----+-----+-----+
| 0000000000000001 | 000000001 | DEP | A | 2017-04-12 | 900.000 | NULL |
| 0000000000000002 | 000000002 | DEP | A | 2017-04-12 | 1000.000 | NULL |
| 0000000000000003 | 000000003 | DEP | A | 2017-04-12 | 1000.000 | NULL |
| 0000000000000004 | 000000001 | LON | A | 2017-12-12 | -47900.200 | NULL |
| 0000000000000005 | 000000001 | LON | A | 2017-04-12 | -100500.000 | NULL |
| 0000000000000006 | 000000002 | DEP | A | 2017-04-12 | 2000.000 | NULL |
+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.02 sec)

mysql> select * from depositaccount;
+-----+-----+-----+-----+-----+-----+
| Account_AccountNum | Account_Branch_RoutingNum | Customer_CustomerNum | DeposAccountType_AccountType | AvailableBalance | HoldBalance |
+-----+-----+-----+-----+-----+-----+
| 0000000000000001 | 000000001 | 0000000000000001 | 0001 | 700.000 | 200.000 |
| 0000000000000002 | 000000002 | 0000000000000001 | 0002 | 1000.000 | 0.000 |
| 0000000000000003 | 000000003 | 0000000000000002 | 0001 | 1000.000 | 0.000 |
| 0000000000000006 | 000000002 | 0000000000000004 | 0002 | 2000.000 | 0.000 |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from branch;
+-----+-----+-----+-----+-----+-----+-----+
| RoutingNum | BranchName | Street | City | State | Country | Zipcode | BranchBalance |
+-----+-----+-----+-----+-----+-----+-----+
| 000000001 | x | Fenway | Boston | MA | United States | 02110 | 100000.000 |
| 000000002 | y | Huntington | Boston | MA | United States | 02114 | 100000.000 |
| 000000003 | z | Parker Hill | Boston | MA | United States | 02113 | 100000.000 |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from transaction;
+-----+-----+-----+-----+-----+-----+-----+
| TransactionId | TransactionDate | TransactionType_TransactionType | TransactionStatus_TransactionStatus | CreditAccount_AccountNum |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | 2017-12-13 | 0001 | | c | 0000000000000001 |
+-----+-----+-----+-----+-----+-----+-----+
1 rows in set (0.00 sec)
```

Data of transaction table is replicated into myddb.transaction table due to trigger tg\_transaction

```
Administrator: C:\Windows\system32\cmd.exe - mysql -u root -p

mysql> select * from branch;
+-----+-----+-----+-----+-----+-----+-----+
| RoutingNum | BranchName | Street | City | State | Country | Zipcode | BranchBalance |
+-----+-----+-----+-----+-----+-----+-----+
| 000000001 | x | Fenway | Boston | MA | United States | 02110 | 100000.000 |
| 000000002 | y | Huntington | Boston | MA | United States | 02114 | 100000.000 |
| 000000003 | z | Parker Hill | Boston | MA | United States | 02113 | 100000.000 |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from transaction;
+-----+-----+-----+-----+-----+-----+-----+
| TransactionId | TransactionDate | TransactionType_TransactionType | TransactionStatus_TransactionStatus | CreditAccount_AccountNum |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | 2017-12-13 | 0001 | | c | 0000000000000001 |
| 2 | 2017-12-13 | 0003 | | c | NULL |
| 3 | 2017-12-13 | 0005 | | c | 0000000000000004 |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)

mysql> select * from myddb.transaction;
+-----+-----+-----+-----+-----+-----+-----+
| TransactionId | TransactionDate | TransactionType_TransactionType | TransactionStatus_TransactionStatus | CreditAccount_AccountNum |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | 2017-12-13 | 0001 | | c | 0000000000000001 |
| 2 | 2017-12-13 | 0003 | | c | NULL |
| 3 | 2017-12-13 | 0005 | | c | 0000000000000004 |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql>
```



#### 4. sp\_cust\_dep\_acct\_balance

This procedure is used for get all deposit accounts and their balances of a given customer .  
It uses **v\_customer\_dep\_accounts** view.

```
delimiter %%
create procedure sp_cust_dep_acct_balance(IN cust char(16))
begin
select distinct Account_AccountNum,AvailableBalance+HoldBalance+UnclearedBalance Acct_Balance
from v_customer_dep_accounts where CustomerNum = cust;
end
%%
delimiter ;

call sp_cust_dep_acct_balance('0000000000000001');
```

Calling sp\_cust\_dep\_acct\_balance

The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'SCHEMAS' tree with 'myddb' selected. The main editor window shows the SQL code for the procedure and its execution. The 'Result Grid' shows the output of the procedure call, displaying two rows of account data. The 'Output' pane at the bottom shows the execution log, including the time and duration of the procedure call.

Account_AccountNum	Acct_Balance
0000000000000001	900.000
0000000000000002	1000.000

#	Time	Action	Message	Duration / Fetch
452	23:00:12	call sp_TransferTransaction('0000000000000004';100;'0000000001';'0000000000...)	1 row(s) returned	- / 0.000 sec
453	23:02:45	call sp_cust_dep_acct_balance('0000000000000001')	2 row(s) returned	0.000 sec / 0.000 sec

## 5. sp\_dep\_acct\_transaction

This procedure is used for get the transaction details of a given account .  
It also uses **v\_customer\_dep\_accounts** view.

```
delimiter %%
create procedure sp_dep_acct_transaction(IN acct char(16),IN rout char(9))
begin
select transactiondate, coalesce(creditaccount_accountNum,' ') CreditAccount,
coalesce(creditaccount_branch_routingnum,' ') CreditAcctRountingNum,
coalesce(debitaccount_accountNum,' ') DebitAccount,
coalesce(debitaccount_branch_routingnum,' ') DebitAcctRountingNum , transactionAmount,
tt.Description Transaction_type, ts.Description Transaction_status
from v_customer_dep_accounts inner join transactiontype tt
on transactiontype_transactiontype = tt.transactiontype
inner join transactionstatus ts on transactionstatus_transactionstatus = ts.transactionstatus
where ((creditaccount_accountNum=acct && creditaccount_branch_routingnum=rout) OR
(debitaccount_accountNum=acct && debitaccount_branch_routingnum=rout))
order by TransactionDate desc, TransactionId desc;
end
%%
delimiter ;

call sp_dep_acct_transaction('0000000000000001','000000001');
```

Calling sp\_dep\_acct\_transaction

The screenshot displays the MySQL Workbench interface. The Navigator pane on the left shows the database schema, including the **v\_customer\_dep\_accounts** view and the **sp\_dep\_acct\_transaction** stored procedure. The central Query Editor shows the procedure code, which is being executed. The Result Grid displays the output of the procedure, showing transaction details for three transactions. The Output pane at the bottom shows the Action Output table, which includes the time, action, message, and duration for each step.

TransactionDate	CreditAccount	CreditAcctRountingNum	DebitAccount	DebitAcctRountingNum	TransactionAmount	TransactionType	TransactionStatus
17-12-13	0000000000000004	000000001	0000000000000001	000000001	100.000	Transfer	
17-12-13	0000000000000001	000000001	0000000000000001	000000001	800.000	Deposit	
17-12-13	0000000000000001	000000001	0000000000000001	000000001	800.000	Deposit	

#	Time	Action	Message	Duration / Fetch
453	23:02:45	call sp_cust_dep_acct_balance('0000000000000001')	2 row(s) returned	0.000 sec / 0.000 sec
454	23:03:15	call sp_dep_acct_transaction('0000000000000001','000000001')	3 row(s) returned	0.031 sec / 0.000 sec

## 6. sp\_deeduct\_min\_bal\_charge

This procedure uses **Cursor** to deduct minimum balance charge on applicable accounts and park pending charges on account if there are insufficient funds.

Please refer “project\_script.sql” for code

Data before executing sp\_deeduct\_min\_bal\_charge

```
Administrator: C:\Windows\system32\cmd.exe - mysql -u root -p
mysql> select * from deposaccounttype;
+-----+-----+-----+-----+-----+-----+
| AccountType | Status | Description | InterestRate | MinimumBalance | ChargeAmt |
+-----+-----+-----+-----+-----+-----+
| 0001        | A      | Saving      | 2.00         | 100.000        | 10.000    |
| 0002        | A      | Current     | 0.00         | 0.000          | 0.000     |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from account;
+-----+-----+-----+-----+-----+-----+-----+
| AccountNum | Branch_RoutingNum | Sys | Status | OpenDate | Balance | CloseDate |
+-----+-----+-----+-----+-----+-----+-----+
| 00000000000000000001 | 0000000001 | DEP | A      | 2017-04-12 | 900.000 | NULL      |
| 00000000000000000002 | 0000000002 | DEP | A      | 2017-04-12 | 1000.000 | NULL      |
| 00000000000000000003 | 0000000003 | DEP | A      | 2017-04-12 | 1000.000 | NULL      |
| 00000000000000000004 | 0000000001 | LON | A      | 2017-12-12 | -47900.200 | NULL      |
| 00000000000000000005 | 0000000001 | LON | A      | 2017-04-12 | -100500.000 | NULL      |
| 00000000000000000006 | 0000000002 | DEP | A      | 2017-04-12 | 2000.000 | NULL      |
+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select * from depositaccount;
+-----+-----+-----+-----+-----+-----+-----+
| Account_AccountNum | Account_Branch_RoutingNum | Customer_CustomerNum | DeposAccountType_AccountType | AvailableBalance | HoldBalance |
+-----+-----+-----+-----+-----+-----+-----+
| 00000000000000000001 | 0000000001 | 00000000000000000001 | 0001 | 700.000 | 200.000 |
| 00000000000000000002 | 0000000002 | 00000000000000000001 | 0002 | 1000.000 | 0.000 |
| 00000000000000000003 | 0000000003 | 00000000000000000002 | 0001 | 1000.000 | 0.000 |
| 00000000000000000006 | 0000000002 | 00000000000000000004 | 0002 | 2000.000 | 0.000 |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from branch;
+-----+-----+-----+-----+-----+-----+-----+
| RoutingNum | BranchName | Street | City | State | Country | Zipcode | BranchBalance |
+-----+-----+-----+-----+-----+-----+-----+
| 0000000001 | x          | Fenway | Boston | MA | United States | 02110 | 100000.000 |
| 0000000002 | y          | Huntington | Boston | MA | United States | 02114 | 100000.000 |
| 0000000003 | z          | Parker Hill | Boston | MA | United States | 02113 | 100000.000 |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from transaction;
```

```

Administrator: C:\Windows\system32\cmd.exe - mysql -u root -p

+-----+-----+-----+-----+-----+-----+-----+-----+
| RoutingNum | BranchName | Street      | City   | State | Country   | Zipcode | BranchBalance |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 000000001 | x          | Fenway      | Boston | MA    | United States | 02110    | 100000.000    |
| 000000002 | y          | Huntington  | Boston | MA    | United States | 02114    | 100000.000    |
| 000000003 | z          | Parker Hill | Boston | MA    | United States | 02113    | 100000.000    |
+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from transaction;
+-----+-----+-----+-----+-----+-----+-----+-----+
| TransactionId | TransactionDate | TransactionType_TransactionType | TransactionStatus_TransactionStatus | CreditAccount_AccountNum |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | 2017-12-13 | 0001 | C | 0000000000000000001 |
| 2 | 2017-12-13 | 0003 | C | NULL |
| 3 | 2017-12-13 | 0005 | C | 0000000000000000004 |
+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from myddb.transaction;
+-----+-----+-----+-----+-----+-----+-----+-----+
| TransactionId | TransactionDate | TransactionType_TransactionType | TransactionStatus_TransactionStatus | CreditAccount_AccountNum |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | 2017-12-13 | 0001 | C | 0000000000000000001 |
| 2 | 2017-12-13 | 0003 | C | NULL |
| 3 | 2017-12-13 | 0005 | C | 0000000000000000004 |
+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>

```

Withdraw from account '0000000000000001' so that available balance is < Minimum Balance and available balance > 0

Calling sp\_WithdrawTransaction

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying a SQL script that calls the stored procedure `sp_WithdrawTransaction` with parameters `'0000000000000001'`, `610`, and `'000000001'`. The 'Result Grid' shows a single row with the message 'Debit Transaction Successful and Balance is :290.000 Account :0000000000000001 Svs :DEP'. The 'Output' tab at the bottom shows the execution log, indicating that the procedure was called successfully at 23:08:05 and returned 1 row(s).

```

-- end if;
-- end
-- %
delimiter ;
call sp_WithdrawTransaction('0000000000000001',610,'000000001');
-- Procedures
delimiter %

```

#	Time	Action	Message	Duration / Fetch
457	23:08:05	call sp_WithdrawTransaction('0000000000000001',610,'000000001')	1 row(s) returned	- / 0.000 sec
458	23:08:05	call sp_WithdrawTransaction('0000000000000001',610,'000000001')	1 row(s) returned	- / 0.000 sec

Withdraw from account '0000000000000003' so that available balance is < Minimum Balance and available balance = 0

Calling sp\_WithdrawTransaction

The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'SCHEMAS' tree with 'myddb' selected. The main editor shows a SQL query with a call to the stored procedure `sp_WithdrawTransaction`. The query is as follows:

```
144     end if;  
145  
146     else select 'invalid debiting account or debiting account is closed' result;  
147     end if;  
148  
149     --  
150     delimiter ;  
151  
152     call sp_WithdrawTransaction('0000000000000003',1000,'00000003');  
153  
154  
155  
156  
157 -- Procedures
```

The 'Result Grid' shows the output of the query:

result
Debit Transaction Successful and Balance is :0.000 Account :0000000000000003 Sys :DEP

The 'Output' tab at the bottom shows the execution log:

#	Time	Action	Message	Duration / Fetch
461	23:09:15	call sp_WithdrawTransaction('0000000000000003',1000,'00000003')	1 row(s) returned	- / 0.000 sec
462	23:09:15	call sp_WithdrawTransaction('0000000000000003',1000,'00000003')	1 row(s) returned	- / 0.000 sec

The status bar at the bottom indicates 'Query Completed'.

## Data after executing sp\_WithdrawTransaction

```
Administrator: C:\Windows\system32\cmd.exe - mysql -u root -p

mysql> select * from account;
+-----+-----+-----+-----+-----+-----+-----+
| AccountNum | Branch_RoutingNum | Sys | Status | OpenDate | Balance | CloseDate |
+-----+-----+-----+-----+-----+-----+-----+
| 0000000000000001 | 000000001 | DEP | A | 2017-04-12 | 290.000 | NULL |
| 0000000000000002 | 000000002 | DEP | A | 2017-04-12 | 1000.000 | NULL |
| 0000000000000003 | 000000003 | DEP | A | 2017-04-12 | 0.000 | NULL |
| 0000000000000004 | 000000001 | LON | A | 2017-12-12 | -47900.200 | NULL |
| 0000000000000005 | 000000001 | LON | A | 2017-04-12 | -100500.000 | NULL |
| 0000000000000006 | 000000002 | DEP | A | 2017-04-12 | 2000.000 | NULL |
+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select * from depositaccount;
+-----+-----+-----+-----+-----+-----+-----+
| Account_AccountNum | Account_Branch_RoutingNum | Customer_CustomerNum | DeposAccountType_AccountType | AvailableBalance | HoldBalance |
+-----+-----+-----+-----+-----+-----+-----+
| 0000000000000001 | 000000001 | 0000000000000001 | 0001 | 90.000 | 200.000 |
| 0000000000000002 | 000000002 | 0000000000000001 | 0002 | 1000.000 | 0.000 |
| 0000000000000003 | 000000003 | 0000000000000002 | 0001 | 0.000 | 0.000 |
| 0000000000000006 | 000000002 | 0000000000000004 | 0002 | 2000.000 | 0.000 |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from branch;
+-----+-----+-----+-----+-----+-----+-----+
| RoutingNum | BranchName | Street | City | State | Country | Zipcode | BranchBalance |
+-----+-----+-----+-----+-----+-----+-----+
| 000000001 | x | Fenway | Boston | MA | United States | 02110 | 99390.000 |
| 000000002 | y | Huntington | Boston | MA | United States | 02114 | 100000.000 |
| 000000003 | z | Parker Hill | Boston | MA | United States | 02113 | 99000.000 |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)

mysql> select * from transaction;
+-----+-----+-----+-----+-----+-----+-----+
| TransactionId | TransactionDate | TransactionType_TransactionType | TransactionStatus_TransactionStatus | CreditAccount_AccountNum |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | 2017-12-13 | 0001 | C | 0000000000000001 |
+-----+-----+-----+-----+-----+-----+-----+
1 rows in set (0.00 sec)
```

Data of transaction table is replicated into myddb.transaction table due to trigger tg\_transaction

```
Administrator: C:\Windows\system32\cmd.exe - mysql -u root -p

mysql> select * from transaction;
+-----+-----+-----+-----+-----+-----+-----+
| TransactionId | TransactionDate | TransactionType_TransactionType | TransactionStatus_TransactionStatus | CreditAccount_AccountNum |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | 2017-12-13 | 0001 | C | 0000000000000001 |
| 2 | 2017-12-13 | 0003 | C | NULL |
| 3 | 2017-12-13 | 0005 | C | 0000000000000004 |
| 4 | 2017-12-13 | 0003 | C | NULL |
| 5 | 2017-12-13 | 0003 | C | NULL |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from myddb.transaction;
+-----+-----+-----+-----+-----+-----+-----+
| TransactionId | TransactionDate | TransactionType_TransactionType | TransactionStatus_TransactionStatus | CreditAccount_AccountNum |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | 2017-12-13 | 0001 | C | 0000000000000001 |
| 2 | 2017-12-13 | 0003 | C | NULL |
| 3 | 2017-12-13 | 0005 | C | 0000000000000004 |
| 4 | 2017-12-13 | 0003 | C | NULL |
| 5 | 2017-12-13 | 0003 | C | NULL |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

## Calling sp\_deeduct\_min\_bal\_charge

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'myddb' selected. The main editor shows a SQL script with the following key parts:

```
-- update account a set Balance=Balance-v_chramt
-- where a.accountNum = v_acct and a.branch_routingNum=v_rout;
commit;
end if;
end loop get_acct;
close acct_cursor;
end
--%
delimiter ;
call sp_deeduct_min_bal_charge();
-- Procedures
drop procedure sp_recover_pending_chr;
delimiter %
create procedure sp_recover_pending_chr()
begin
  declare v_finished INTEGER DEFAULT 0;
  declare v_acct char(16) DEFAULT "";
  declare v_rout char(9) DEFAULT "";
  declare v_availbal decimal(23,3) default 0;
  declare v_holdbal decimal(23,3) default 0;
  declare v_unclbal decimal(23,3) default 0;
  -- declare v_penchr decimal(23,3) default 0;
  declare v_penchr decimal(23,3) default 0;
```

The 'Output' pane at the bottom shows the execution results:

#	Time	Action	Message	Duration / Fetch
462	23:09:15	call sp_WithdrawTransaction('0000000000000003',1000,'0000000003')	1 row(s) returned	- / 0.000 sec
463	23:11:35	call sp_deeduct_min_bal_charge()	0 row(s) affected	0.141 sec

## Data after executing sp\_deeduct\_min\_bal\_charge

The screenshot shows a MySQL command prompt window with the following queries and results:

Query 1: `select * from account;`

AccountNum	Branch_RoutingNum	Sys	Status	OpenDate	Balance	CloseDate
0000000000000001	000000001	DEP	A	2017-04-12	280.000	NULL
0000000000000002	000000002	DEP	A	2017-04-12	1000.000	NULL
0000000000000003	000000003	DEP	A	2017-04-12	0.000	NULL
0000000000000004	000000001	LON	A	2017-12-12	-47900.200	NULL
0000000000000005	000000001	LON	A	2017-04-12	-100500.000	NULL
0000000000000006	000000002	DEP	A	2017-04-12	2000.000	NULL

Query 2: `select * from depositaccount;`

Account_AccountNum	Account_Branch_RoutingNum	Customer_CustomerNum	DeposAccountType_AccountType	AvailableBalance	HoldBalance
0000000000000001	000000001	0000000000000001	0001	80.000	200.000
0000000000000002	000000002	0000000000000001	0002	1000.000	0.000
0000000000000003	000000003	0000000000000002	0001	0.000	0.000
0000000000000006	000000002	0000000000000004	0002	2000.000	0.000

Query 3: `select * from branch;`

RoutingNum	BranchName	Street	City	State	Country	Zipcode	BranchBalance
000000001	x	Fenway	Boston	MA	United States	02110	99400.000
000000002	y	Huntington	Boston	MA	United States	02114	100000.000
000000003	z	Parker Hill	Boston	MA	United States	02113	99000.000

Query 4: `select * from transaction;`

TransactionId	TransactionDate	TransactionType_TransactionType	TransactionStatus_TransactionStatus	CreditAccount_AccountNum
1	2017-12-13	0001	C	0000000000000001
2	2017-12-13	0003	C	NULL

Data of transaction table is replicated into mydbb.transaction table due to trigger tg\_transaction

```
Administrator: C:\Windows\system32\cmd.exe - mysql -u root -p
mysql> select * from transaction;
+-----+
| TransactionId | TransactionDate | TransactionType_TransactionType | TransactionStatus_TransactionStatus | CreditAccount_AccountNum |
+-----+
| 1 | 2017-12-13 | 0001 | C | 0000000000000001 |
| 2 | 2017-12-13 | 0003 | C | NULL |
| 3 | 2017-12-13 | 0005 | C | 0000000000000004 |
| 4 | 2017-12-13 | 0003 | C | NULL |
| 5 | 2017-12-13 | 0003 | C | NULL |
| 6 | 2017-12-13 | 0006 | C | NULL |
+-----+
6 rows in set (0.01 sec)

mysql> select * from mydbb.transaction;
+-----+
| TransactionId | TransactionDate | TransactionType_TransactionType | TransactionStatus_TransactionStatus | CreditAccount_AccountNum |
+-----+
| 1 | 2017-12-13 | 0001 | C | 0000000000000001 |
| 2 | 2017-12-13 | 0003 | C | NULL |
| 3 | 2017-12-13 | 0005 | C | 0000000000000004 |
| 4 | 2017-12-13 | 0003 | C | NULL |
| 5 | 2017-12-13 | 0003 | C | NULL |
| 6 | 2017-12-13 | 0006 | C | NULL |
+-----+
```

Charge amount parked into pending charges field of the account with insufficient funds

```
Administrator: C:\Windows\system32\cmd.exe - mysql -u root -p
mysql> select * from mydbb.transaction;
+-----+
| TransactionId | TransactionDate | TransactionType_TransactionType | TransactionStatus_TransactionStatus | CreditAccount_AccountNum |
+-----+
| 1 | 2017-12-13 | 0001 | C | 0000000000000001 |
| 2 | 2017-12-13 | 0003 | C | NULL |
| 3 | 2017-12-13 | 0005 | C | 0000000000000004 |
| 4 | 2017-12-13 | 0003 | C | NULL |
| 5 | 2017-12-13 | 0003 | C | NULL |
| 6 | 2017-12-13 | 0006 | C | NULL |
+-----+
6 rows in set (0.01 sec)

mysql> select Account_AccountNum,Account_Branch_RoutingNum,DeposAccoutType_AccountType,AvailableBalance,PendingChargeAmt
-> from depositaccount;
+-----+
| Account_AccountNum | Account_Branch_RoutingNum | DeposAccoutType_AccountType | AvailableBalance | PendingChargeAmt |
+-----+
| 0000000000000001 | 000000001 | 0001 | 80.000 | 0.000 |
| 0000000000000002 | 000000002 | 0002 | 1000.000 | 0.000 |
| 0000000000000003 | 000000003 | 0001 | 0.000 | 10.000 |
| 0000000000000006 | 000000002 | 0002 | 2000.000 | 0.000 |
+-----+
4 rows in set (0.00 sec)

mysql>
```



## 7. sp\_recover\_pending\_chr

This procedure uses **Cursor** to recover pending charges of accounts .

Please refer “project\_script.sql” for code

Deposit transaction to so that we can recover the pending charge amount

Calling sp\_depositTransaction

The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'SCHEMAS' panel with a tree view of databases including 'myddb'. The main editor window shows a SQL query with the following code:

```
71 end if;
72 select concat('Credit Transaction Successful and Balance is :',balance,' Account :',acct,' :');
73 from account where accountNum = acct and branch_routingNum=rout;
74
75 else
76 select 'invalid crediting account or crediting account is closed' result;
77 end if;
78 end
79 %%
80 delimiter ;
81
82 call sp_depositTransaction('0000000000000003',10,'00000003');
83
84
```

The 'Result Grid' shows the output of the query:

result
Credit Transaction Successful and Balance is :10.000 Account :0000000000000003 Sys :DEP

The 'Output' panel at the bottom shows the execution log:

#	Time	Action	Message	Duration / Fetch
464	23:19:52	call sp_depositTransaction('0000000000000003',10,'00000003')	1 row(s) returned	0.000 sec / 0.000 sec
465	23:19:52	call sp_depositTransaction('0000000000000003',10,'00000003')	1 row(s) returned	- / 0.000 sec

The status bar at the bottom indicates 'Query Completed'.

Data after executing sp\_depositTransaction

```
Administrator: C:\Windows\system32\cmd.exe - mysql -u root -p
mysql> select * from account;
+-----+-----+-----+-----+-----+-----+-----+
| AccountNum | Branch_RoutingNum | Sys | Status | OpenDate | Balance | CloseDate |
+-----+-----+-----+-----+-----+-----+-----+
| 0000000000000001 | 000000001 | DEP | A | 2017-04-12 | 280.000 | NULL |
| 0000000000000002 | 000000002 | DEP | A | 2017-04-12 | 1000.000 | NULL |
| 0000000000000003 | 000000003 | DEP | A | 2017-04-12 | 10.000 | NULL |
| 0000000000000004 | 000000001 | LON | A | 2017-12-12 | -47900.200 | NULL |
| 0000000000000005 | 000000001 | LON | A | 2017-04-12 | -100500.000 | NULL |
| 0000000000000006 | 000000002 | DEP | A | 2017-04-12 | 2000.000 | NULL |
+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select * from depositaccount;
+-----+-----+-----+-----+-----+-----+-----+
| Account_AccountNum | Account_Branch_RoutingNum | Customer_CustomerNum | DeposAccountType_AccountType | AvailableBalance | HoldBalance |
+-----+-----+-----+-----+-----+-----+-----+
| 0000000000000001 | 000000001 | 0000000000000001 | 0001 | 80.000 | 200.000 |
| 0000000000000002 | 000000002 | 0000000000000001 | 0002 | 1000.000 | 0.000 |
| 0000000000000003 | 000000003 | 0000000000000002 | 0001 | 10.000 | 0.000 |
| 0000000000000006 | 000000002 | 0000000000000004 | 0002 | 2000.000 | 0.000 |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from branch;
+-----+-----+-----+-----+-----+-----+-----+
| RoutingNum | BranchName | Street | City | State | Country | Zipcode | BranchBalance |
+-----+-----+-----+-----+-----+-----+-----+
| 000000001 | x | Fenway | Boston | MA | United States | 02110 | 99400.000 |
| 000000002 | y | Huntington | Boston | MA | United States | 02114 | 100000.000 |
| 000000003 | z | Parker Hill | Boston | MA | United States | 02113 | 99010.000 |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from transaction;
+-----+-----+-----+-----+-----+-----+-----+
| TransactionId | TransactionDate | TransactionType_TransactionType | TransactionStatus_TransactionStatus | CreditAccount_AccountNum |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | 2017-12-13 | 0001 | C | 0000000000000001 |
+-----+-----+-----+-----+-----+-----+-----+
-13 |
```

Data of transaction table is replicated into myddb.transaction table due to trigger tg\_transaction

```
Administrator: C:\Windows\system32\cmd.exe - mysql -u root -p
mysql> select * from transaction;
+-----+-----+-----+-----+-----+-----+-----+
| TransactionId | TransactionDate | TransactionType_TransactionType | TransactionStatus_TransactionStatus | CreditAccount_AccountNum |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | 2017-12-13 | 0001 | C | 0000000000000001 |
| 2 | 2017-12-13 | 0003 | C | NULL |
| 3 | 2017-12-13 | 0005 | C | 0000000000000004 |
| 4 | 2017-12-13 | 0003 | C | NULL |
| 5 | 2017-12-13 | 0003 | C | NULL |
| 6 | 2017-12-13 | 0006 | C | NULL |
| 7 | 2017-12-13 | 0001 | C | 0000000000000003 |
+-----+-----+-----+-----+-----+-----+-----+
7 rows in set (0.01 sec)

mysql> select * from myddb.transaction;
+-----+-----+-----+-----+-----+-----+-----+
| TransactionId | TransactionDate | TransactionType_TransactionType | TransactionStatus_TransactionStatus | CreditAccount_AccountNum |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | 2017-12-13 | 0001 | C | 0000000000000001 |
| 2 | 2017-12-13 | 0003 | C | NULL |
| 3 | 2017-12-13 | 0005 | C | 0000000000000004 |
| 4 | 2017-12-13 | 0003 | C | NULL |
| 5 | 2017-12-13 | 0003 | C | NULL |
| 6 | 2017-12-13 | 0006 | C | NULL |
+-----+-----+-----+-----+-----+-----+-----+
-13 |
```

```
Administrator: C:\Windows\system32\cmd.exe - mysql -u root -p
mysql> select * from mydbb.transaction;
+-----+-----+-----+-----+-----+-----+
| TransactionId | TransactionDate | TransactionType_TransactionType | TransactionStatus_TransactionStatus | CreditAccount_AccountNum |
+-----+-----+-----+-----+-----+-----+
| 1 | 2017-12-13 | 0001 | C | 0000000000000000001 |
| 2 | 2017-12-13 | 0003 | C | NULL |
| 3 | 2017-12-13 | 0005 | C | 0000000000000000004 |
| 4 | 2017-12-13 | 0003 | C | NULL |
| 5 | 2017-12-13 | 0003 | C | NULL |
| 6 | 2017-12-13 | 0006 | C | NULL |
| 7 | 2017-12-13 | 0001 | C | 0000000000000000003 |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.01 sec)

mysql> select Account_AccountNum,Account_Branch_RoutingNum,DeposAccoutType_AccountType,AvailableBalance,PendingChargeAmt
-> from depositacctnum;
+-----+-----+-----+-----+-----+
| Account_AccountNum | Account_Branch_RoutingNum | DeposAccoutType_AccountType | AvailableBalance | PendingChargeAmt |
+-----+-----+-----+-----+-----+
| 0000000000000000001 | 0000000001 | 0001 | 80.000 | 0.000 |
| 0000000000000000002 | 0000000002 | 0002 | 1000.000 | 0.000 |
| 0000000000000000003 | 0000000003 | 0001 | 10.000 | 10.000 |
| 0000000000000000006 | 0000000002 | 0002 | 2000.000 | 0.000 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

[illegible]

## Data after executing sp\_recover\_pending\_chr

```
Administrator: C:\Windows\system32\cmd.exe - mysql -u root -p
mysql> select * from account;
```

AccountNum	Branch_RoutingNum	Sys	Status	OpenDate	Balance	CloseDate
00000000000000000001	000000001	DEP	A	2017-04-12	280.000	NULL
00000000000000000002	000000002	DEP	A	2017-04-12	1000.000	NULL
00000000000000000003	000000003	DEP	A	2017-04-12	0.000	NULL
00000000000000000004	000000001	LON	A	2017-12-12	-47900.200	NULL
00000000000000000005	000000001	LON	A	2017-04-12	-100500.000	NULL
00000000000000000006	000000002	DEP	A	2017-04-12	2000.000	NULL

```
6 rows in set (0.00 sec)

mysql> select * from depositaccount;
```

Account_AccountNum	Account_Branch_RoutingNum	Customer_CustomerNum	DeposAccountType_AccountType	AvailableBalance	HoldBalance
00000000000000000001	000000001	00000000000000000001	0001	80.000	200.000
00000000000000000002	000000002	00000000000000000001	0002	1000.000	0.000
00000000000000000003	000000003	00000000000000000002	0001	0.000	0.000
00000000000000000006	000000002	00000000000000000004	0002	2000.000	0.000

```
4 rows in set (0.00 sec)

mysql> select * from branch;
```

RoutingNum	BranchName	Street	City	State	Country	Zipcode	BranchBalance
000000001	x	Fenway	Boston	MA	United States	02110	99400.000
000000002	y	Huntington	Boston	MA	United States	02114	100000.000
000000003	z	Parker Hill	Boston	MA	United States	02113	99020.000

```
3 rows in set (0.00 sec)

mysql> select * from transaction;
```

TransactionId	TransactionDate	TransactionType_TransactionType	TransactionStatus_TransactionStatus	CreditAccount_AccountNum
1	2017-12-13	0001	c	00000000000000000001

```
1 row in set (0.00 sec)

Query Completed
```

Data of transaction table is replicated into mydbb.transaction table due to trigger tg\_transaction

```
Administrator: C:\Windows\system32\cmd.exe - mysql -u root -p
mysql> select * from transaction;
+-----+
| TransactionId | TransactionDate | TransactionType_TransactionType | TransactionStatus_TransactionStatus | CreditAccount_AccountNum |
+-----+
| 1 | 2017-12-13 | 0001 | C | 000000000000000001 |
| 2 | 2017-12-13 | 0003 | C | NULL |
| 3 | 2017-12-13 | 0005 | C | 000000000000000004 |
| 4 | 2017-12-13 | 0003 | C | NULL |
| 5 | 2017-12-13 | 0003 | C | NULL |
| 6 | 2017-12-13 | 0006 | C | NULL |
| 7 | 2017-12-13 | 0001 | C | 000000000000000003 |
| 8 | 2017-12-13 | 0006 | C | NULL |
+-----+
8 rows in set (0.00 sec)

mysql> select * from mydbb.transaction;
+-----+
| TransactionId | TransactionDate | TransactionType_TransactionType | TransactionStatus_TransactionStatus | CreditAccount_AccountNum |
+-----+
| 1 | 2017-12-13 | 0001 | C | 000000000000000001 |
| 2 | 2017-12-13 | 0003 | C | NULL |
| 3 | 2017-12-13 | 0005 | C | 000000000000000004 |
| 4 | 2017-12-13 | 0003 | C | NULL |
| 5 | 2017-12-13 | 0003 | C | NULL |
+-----+
```

Pending charges recovered.

```
Administrator: C:\Windows\system32\cmd.exe - mysql -u root -p
8 rows in set (0.00 sec)

mysql> select * from mydbb.transaction;
+-----+
| TransactionId | TransactionDate | TransactionType_TransactionType | TransactionStatus_TransactionStatus | CreditAccount_AccountNum |
+-----+
| 1 | 2017-12-13 | 0001 | C | 000000000000000001 |
| 2 | 2017-12-13 | 0003 | C | NULL |
| 3 | 2017-12-13 | 0005 | C | 000000000000000004 |
| 4 | 2017-12-13 | 0003 | C | NULL |
| 5 | 2017-12-13 | 0003 | C | NULL |
| 6 | 2017-12-13 | 0006 | C | NULL |
| 7 | 2017-12-13 | 0001 | C | 000000000000000003 |
| 8 | 2017-12-13 | 0006 | C | NULL |
+-----+
8 rows in set (0.00 sec)

mysql> select Account_AccountNum,Account_Branch_RoutingNum,DeposAccoutType_AccountType,AvailableBalance,PendingChargeAmt
-> from depositaccount;
+-----+
| Account_AccountNum | Account_Branch_RoutingNum | DeposAccoutType_AccountType | AvailableBalance | PendingChargeAmt |
+-----+
| 000000000000000001 | 0000000001 | 0001 | 80.000 | 0.000 |
| 000000000000000002 | 0000000002 | 0002 | 1000.000 | 0.000 |
| 000000000000000003 | 0000000003 | 0001 | 0.000 | 0.000 |
| 000000000000000006 | 0000000002 | 0002 | 2000.000 | 0.000 |
+-----+
4 rows in set (0.00 sec)

mysql>
```

## 8. sp\_arch\_transaction

This procedure uses **Cursor** to archive transaction table records to transactionarch table.

Please refer “project\_script.sql” for code

Data before executing sp\_arch\_transaction

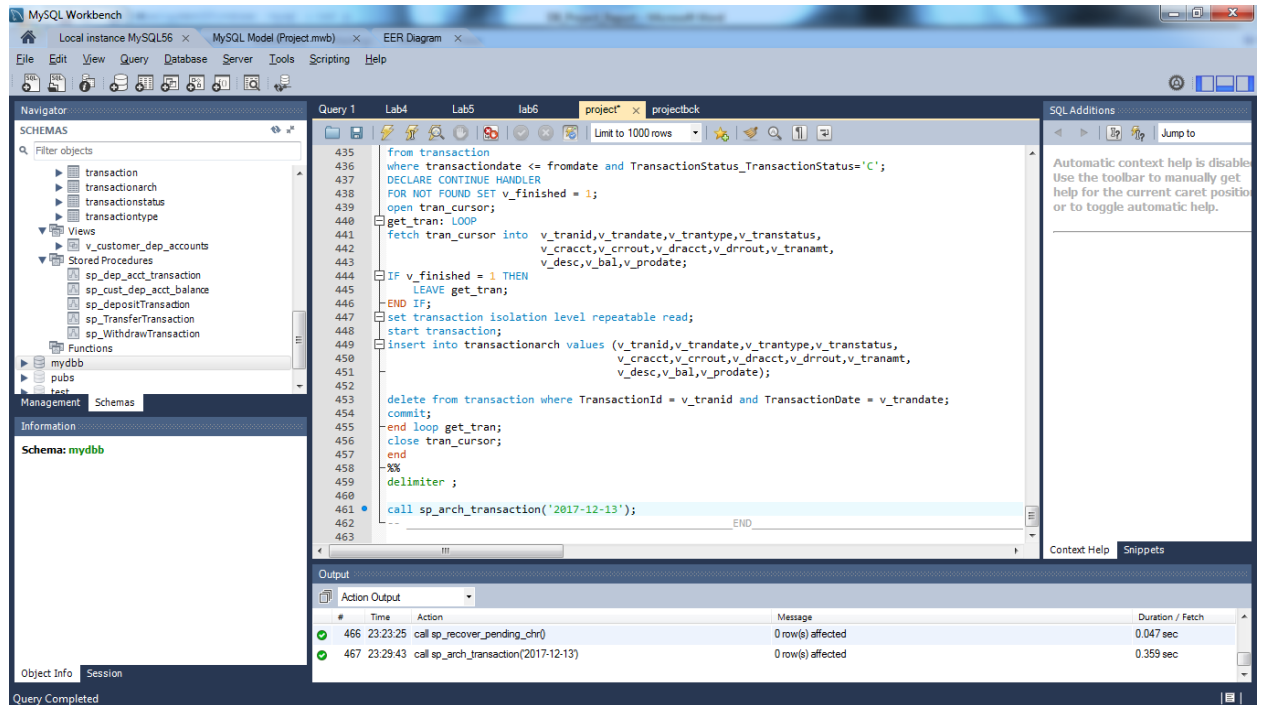
```
Administrator: C:\Windows\system32\cmd.exe - mysql -u root -p
mysql> select Account_AccountNum,Account_Branch_RoutingNum,DeposAccoutType_AccountType,AvailableBalance,PendingChargeAmt
-> from depositaccount;
+-----+-----+-----+-----+-----+
| Account_AccountNum | Account_Branch_RoutingNum | DeposAccoutType_AccountType | AvailableBalance | PendingChargeAmt |
+-----+-----+-----+-----+-----+
| 00000000000000000001 | 0000000001 | 0001 | 80.000 | 0.000 |
| 00000000000000000002 | 0000000002 | 0002 | 1000.000 | 0.000 |
| 00000000000000000003 | 0000000003 | 0001 | 0.000 | 0.000 |
| 00000000000000000006 | 0000000002 | 0002 | 2000.000 | 0.000 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from transaction;
+-----+-----+-----+-----+-----+-----+
| TransactionId | TransactionDate | TransactionType_TransactionType | TransactionStatus_TransactionStatus | CreditAccount_AccountNum |
+-----+-----+-----+-----+-----+-----+
| 1 | 2017-12-13 | 0001 | C | 00000000000000000001 |
| 2 | 2017-12-13 | 0003 | C | NULL |
| 3 | 2017-12-13 | 0005 | C | 00000000000000000004 |
| 4 | 2017-12-13 | 0003 | C | NULL |
| 5 | 2017-12-13 | 0003 | C | NULL |
| 6 | 2017-12-13 | 0006 | C | NULL |
| 7 | 2017-12-13 | 0001 | C | 00000000000000000003 |
| 8 | 2017-12-13 | 0006 | C | NULL |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

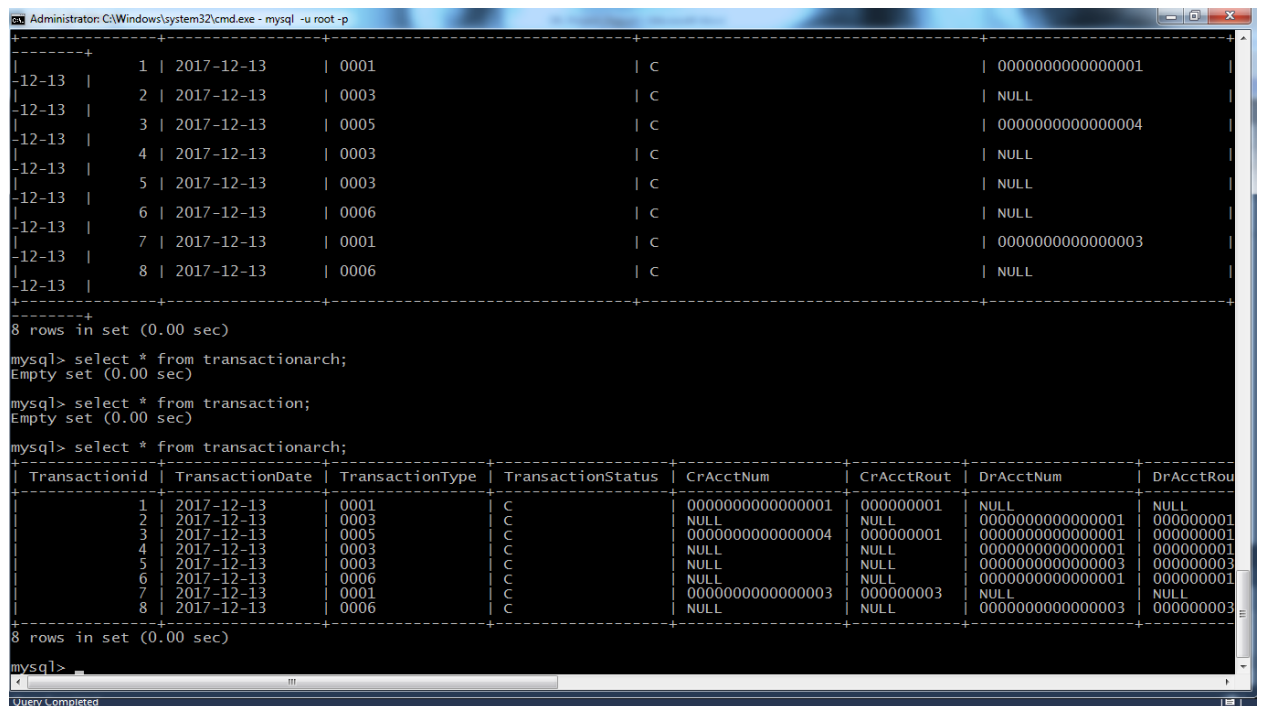
mysql> select * from transactionarch;
Empty set (0.00 sec)

mysql>
```

## Calling sp\_arch\_transaction



Data of transaction table archived from the specified transaction date



END