

SETTING UP TRIGGERS

1. TRANSACTIONS – BALANCE – UPDATE

This trigger updates the balance of the accounts from which amount is credited and debited once a transaction is added.

DELIMITER \$\$

CREATE TRIGGER TRANSACTION_BALANCE_UPDATE_AFTER

AFTER INSERT ON TRANSACTIONS

FOR EACH ROW

BEGIN

 DECLARE EXISTING_BALANCE BIGINT;

 SELECT ACCOUNTS.BALANCE INTO EXISTING_BALANCE

 FROM ACCOUNTS

 WHERE ACCOUNTS.ACCOUNTNO = NEW.DEBITACCOUNTNO

 AND ACCOUNTS.BRANCHIFSC = NEW.DEBITIFSC

 AND ACCOUNTS.ACCSTATUS = 'A';

 IF (EXISTING_BALANCE >= NEW.AMOUNT

 AND NEW.DEBITACCOUNTNO IS NOT NULL

 AND NEW.CREDITACCOUNTNO IS NOT NULL) THEN

 BEGIN

 UPDATE ACCOUNTS

 SET ACCOUNTS.BALANCE = BALANCE - NEW.AMOUNT

 WHERE ACCOUNTS.ACCOUNTNO = NEW.DEBITACCOUNTNO

 AND ACCOUNTS.BRANCHIFSC = NEW.DEBITIFSC

 AND ACCOUNTS.ACCSTATUS = 'A';

 UPDATE ACCOUNTS

 SET ACCOUNTS.BALANCE = BALANCE + NEW.AMOUNT

 WHERE ACCOUNTS.ACCOUNTNO = NEW.CREDITACCOUNTNO

 AND ACCOUNTS.BRANCHIFSC = NEW.CREDITIFSC

 AND ACCOUNTS.ACCSTATUS = 'A';

 END;

 ELSEIF (NEW.DEBITACCOUNTNO IS NULL) THEN

 BEGIN

 UPDATE ACCOUNTS

 SET ACCOUNTS.BALANCE = BALANCE + NEW.AMOUNT

 WHERE ACCOUNTS.ACCOUNTNO = NEW.CREDITACCOUNTNO

 AND ACCOUNTS.BRANCHIFSC = NEW.CREDITIFSC

 AND ACCOUNTS.ACCSTATUS = 'A';

 END;

 ELSEIF (NEW.CREDITACCOUNTNO IS NULL

 EXISTING_BALANCE >= NEW.AMOUNT) THEN

 BEGIN

 UPDATE ACCOUNTS

 SET ACCOUNTS.BALANCE = BALANCE - NEW.AMOUNT

 WHERE ACCOUNTS.ACCOUNTNO = NEW.DEBITACCOUNTNO

```

AND ACCOUNTS.BRANCHIFSC = NEW.DEBITIFSC
AND ACCOUNTS.ACCSTATUS = 'A';

END;
END IF;
END$$
DELIMITER ;

```

```

MySQL Shell
MySQL localhost:33060+ ssl thebank SQL > DELIMITER $$
MySQL localhost:33060+ ssl thebank SQL > CREATE TRIGGER TRANSACTION_BALANCE_UPDATE_AFTER
-> AFTER INSERT ON TRANSACTIONS
-> FOR EACH ROW
-> BEGIN
-> DECLARE EXISTING_BALANCE BIGINT;
-> SELECT ACCOUNTS.BALANCE INTO EXISTING_BALANCE
-> FROM ACCOUNTS
-> WHERE ACCOUNTS.ACCOUNTNO = NEW.DEBITACCOUNTNO
-> AND ACCOUNTS.BRANCHIFSC = NEW.DEBITIFSC
-> AND ACCOUNTS.ACCSTATUS = 'A';
-> IF (EXISTING_BALANCE >= NEW.AMOUNT AND NEW.DEBITACCOUNTNO IS NOT NULL AND NEW.CREDITACCOUNTNO IS NOT NULL) THEN
-> BEGIN
-> UPDATE ACCOUNTS
-> SET ACCOUNTS.BALANCE = BALANCE - NEW.AMOUNT
-> WHERE ACCOUNTS.ACCOUNTNO = NEW.DEBITACCOUNTNO
-> AND ACCOUNTS.BRANCHIFSC = NEW.DEBITIFSC
-> AND ACCOUNTS.ACCSTATUS = 'A';
-> UPDATE ACCOUNTS
-> SET ACCOUNTS.BALANCE = BALANCE + NEW.AMOUNT
-> WHERE ACCOUNTS.ACCOUNTNO = NEW.CREDITACCOUNTNO
-> AND ACCOUNTS.BRANCHIFSC = NEW.CREDITIFSC
-> AND ACCOUNTS.ACCSTATUS = 'A';
-> END;
-> ELSEIF (NEW.DEBITACCOUNTNO IS NULL) THEN
-> BEGIN
-> UPDATE ACCOUNTS
-> SET ACCOUNTS.BALANCE = BALANCE + NEW.AMOUNT
-> WHERE ACCOUNTS.ACCOUNTNO = NEW.CREDITACCOUNTNO
-> AND ACCOUNTS.BRANCHIFSC = NEW.CREDITIFSC
-> AND ACCOUNTS.ACCSTATUS = 'A';
-> END;
-> ELSEIF (NEW.CREDITACCOUNTNO IS NULL AND EXISTING_BALANCE >= NEW.AMOUNT) THEN
-> BEGIN
-> UPDATE ACCOUNTS
-> SET ACCOUNTS.BALANCE = BALANCE - NEW.AMOUNT
-> WHERE ACCOUNTS.ACCOUNTNO = NEW.DEBITACCOUNTNO
-> AND ACCOUNTS.BRANCHIFSC = NEW.DEBITIFSC
-> AND ACCOUNTS.ACCSTATUS = 'A';
-> END;
-> END IF;
-> END$$
Query OK, 0 rows affected (1.3103 sec)
MySQL localhost:33060+ ssl thebank SQL > DELIMITER ;
MySQL localhost:33060+ ssl thebank SQL >

```

2. LOANPAYMENT – OUTSTANDING - UPDATE

This trigger updates the outstanding amount on the loan and customer pays an installment and transaction is added to loan payments relation.

```

DELIMITER $$
CREATE TRIGGER LOANPAY_BALANCE_UPDATE_AFTER
AFTER INSERT ON LOANPAYMENTS
FOR EACH ROW
BEGIN
    DECLARE EXISTING_OUTSTANDING BIGINT;
    SELECT LOANS.OUTSTANDING INTO EXISTING_OUTSTANDING
    FROM LOANS
    WHERE LOANS.LOANACCOUNTNO = NEW.LOANACCOUNTNO
    AND LOANS.BRANCHIFSC = NEW.LOANIFSC
    AND LOANS.LOANSTATUS = 'A';
    IF (EXISTING_OUTSTANDING >= NEW.AMOUNT) THEN
        BEGIN
            UPDATE LOANS
            SET LOANS.OUTSTANDING = LOANS.OUTSTANDING - NEW.AMOUNT
            WHERE LOANS.LOANACCOUNTNO = NEW.LOANACCOUNTNO
            AND LOANS.BRANCHIFSC = NEW.LOANIFSC
            AND LOANS.LOANSTATUS = 'A';
        END;
    END IF;
END$$
DELIMITER ;

```

```
MySQL Shell
MySQL localhost:33060+ ssl thebank SQL > DELIMITER $$
MySQL localhost:33060+ ssl thebank SQL > CREATE TRIGGER LOANPAY_BALANCE_UPDATE_AFTER
-> AFTER INSERT ON LOANPAYMENTS
-> FOR EACH ROW
-> BEGIN
-> DECLARE EXISTING_OUTSTANDING BIGINT;
-> SELECT LOANS.OUTSTANDING INTO EXISTING_OUTSTANDING
-> FROM LOANS
-> WHERE LOANS.LOANACCOUNTNO = NEW.LOANACCOUNTNO
-> AND LOANS.BRANCHIFSC = NEW.LOANIFSC
-> AND LOANS.LOANSTATUS = 'A';
->
-> IF (EXISTING_OUTSTANDING >= NEW.AMOUNT) THEN
-> BEGIN
-> UPDATE LOANS
-> SET LOANS.OUTSTANDING = LOANS.OUTSTANDING - NEW.AMOUNT
-> WHERE LOANS.LOANACCOUNTNO = NEW.LOANACCOUNTNO
-> AND LOANS.BRANCHIFSC = NEW.LOANIFSC
-> AND LOANS.LOANSTATUS = 'A';
-> END;
-> END IF;
-> END$$
Query OK, 0 rows affected (1.8659 sec)
MySQL localhost:33060+ ssl thebank SQL > DELIMITER ;
MySQL localhost:33060+ ssl thebank SQL >
```

3. DESIGNATION UPDATION TO MANAGER

This trigger updates the designation of employee to manger once his/her EmployeeID is recorded as ManagerID of a branch in branches relations. This also updates the IFSCcode of the manager.

```
DELIMITER $$
CREATE TRIGGER MANAGER_UPDATE_AFTER
AFTER INSERT ON BRANCHES
FOR EACH ROW
BEGIN
    UPDATE EMPLOYEES
    SET EMPLOYEES.BRANCHIFSC = NEW.IFSCCODE,
        EMPLOYEES.DESIGNATION = 'MANAGER'
    WHERE EMPLOYEES.EMPLOYEEID = NEW.MANAGERID;
END$$
DELIMITER ;
```

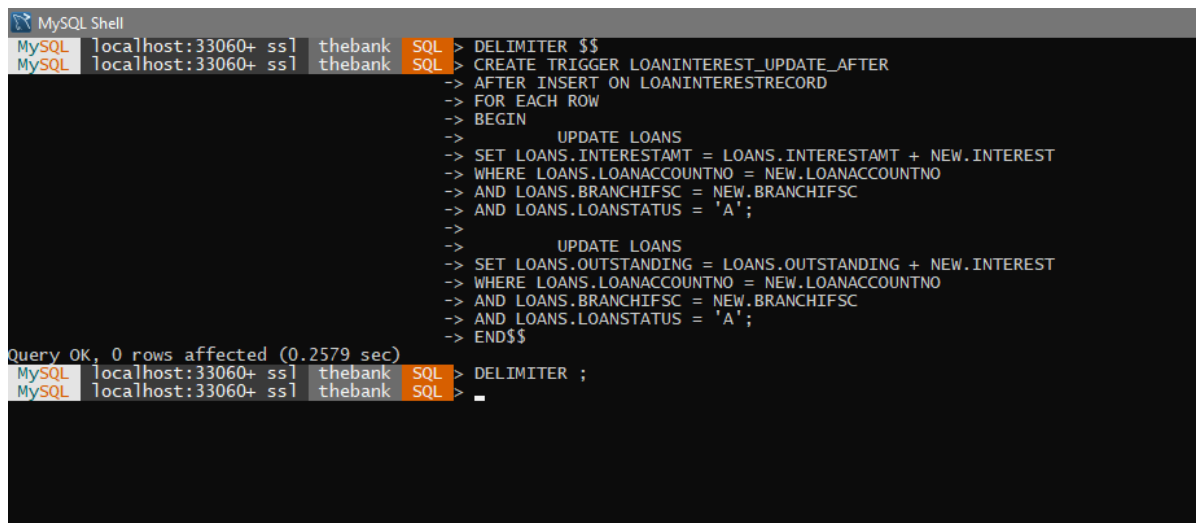
```
MySQL Shell
MySQL localhost:33060+ ssl thebank SQL > DELIMITER $$
MySQL localhost:33060+ ssl thebank SQL > CREATE TRIGGER MANAGER_UPDATE_AFTER
-> AFTER INSERT ON BRANCHES
-> FOR EACH ROW
-> BEGIN
-> UPDATE EMPLOYEES
-> SET EMPLOYEES.BRANCHIFSC = NEW.IFSCCODE, EMPLOYEES.DESIGNATION = 'MANAGER'
-> WHERE EMPLOYEES.EMPLOYEEID = NEW.MANAGERID;
-> END$$
Query OK, 0 rows affected (0.2394 sec)
MySQL localhost:33060+ ssl thebank SQL > DELIMITER ;
MySQL localhost:33060+ ssl thebank SQL >
```

4. LOAN INTEREST UPDATE

This trigger updates the outstanding and interest on a loan after records are inserted in loan interest record.

```
DELIMITER $$
CREATE TRIGGER LOANINTEREST_UPDATE_AFTER
AFTER INSERT ON LOANINTERESTRECORD
FOR EACH ROW
BEGIN
    UPDATE LOANS
    SET LOANS.INTERESTAMT = LOANS.INTERESTAMT + NEW.INTEREST
    WHERE LOANS.LOANACCOUNTNO = NEW.LOANACCOUNTNO
    AND LOANS.BRANCHIFSC = NEW.BRANCHIFSC
    AND LOANS.LOANSTATUS = 'A';

    UPDATE LOANS
    SET LOANS.OUTSTANDING = LOANS.OUTSTANDING + NEW.INTEREST
    WHERE LOANS.LOANACCOUNTNO = NEW.LOANACCOUNTNO
    AND LOANS.BRANCHIFSC = NEW.BRANCHIFSC
    AND LOANS.LOANSTATUS = 'A';
END$$
DELIMITER ;
```



```
MySQL Shell
MySQL localhost:33060+ ssl thebank SQL > DELIMITER $$
MySQL localhost:33060+ ssl thebank SQL > CREATE TRIGGER LOANINTEREST_UPDATE_AFTER
-> AFTER INSERT ON LOANINTERESTRECORD
-> FOR EACH ROW
-> BEGIN
->     UPDATE LOANS
->     SET LOANS.INTERESTAMT = LOANS.INTERESTAMT + NEW.INTEREST
->     WHERE LOANS.LOANACCOUNTNO = NEW.LOANACCOUNTNO
->     AND LOANS.BRANCHIFSC = NEW.BRANCHIFSC
->     AND LOANS.LOANSTATUS = 'A';
->
->     UPDATE LOANS
->     SET LOANS.OUTSTANDING = LOANS.OUTSTANDING + NEW.INTEREST
->     WHERE LOANS.LOANACCOUNTNO = NEW.LOANACCOUNTNO
->     AND LOANS.BRANCHIFSC = NEW.BRANCHIFSC
->     AND LOANS.LOANSTATUS = 'A';
-> END$$
Query OK, 0 rows affected (0.2579 sec)
MySQL localhost:33060+ ssl thebank SQL > DELIMITER ;
MySQL localhost:33060+ ssl thebank SQL >
```

5. DEPOSIT INTEREST UPDATE

This trigger updates balance on deposits after records are inserted in deposit interest record.

```
DELIMITER $$
CREATE TRIGGER DEPOSITINTEREST_UPDATE_AFTER
AFTER INSERT ON DEPOSITINTERESTINFO
FOR EACH ROW
BEGIN
    UPDATE ACCOUNTS
    SET ACCOUNTS.BALANCE = ACCOUNTS.BALANCE + NEW.INTEREST
    WHERE ACCOUNTS.ACCOUNTNO = NEW.ACCOUNTNO
    AND ACCOUNTS.BRANCHIFSC = NEW.BRANCHIFSC
```

```
        AND ACCOUNTS.ACCSTATUS = 'A';
END$$
DELIMITER ;
```

```
MySQL Shell
MySQL localhost:33060+ ssl thebank SQL > DELIMITER $$
MySQL localhost:33060+ ssl thebank SQL > CREATE TRIGGER DEPOSITINTEREST_UPDATE_AFTER
-> AFTER INSERT ON DEPOSITINTERESTINFO
-> FOR EACH ROW
-> BEGIN
->     UPDATE ACCOUNTS
->     SET ACCOUNTS.BALANCE = ACCOUNTS.BALANCE + NEW.INTEREST
->     WHERE ACCOUNTS.ACCOUNTNO = NEW.ACCOUNTNO
->     AND ACCOUNTS.BRANCHIFSC = NEW.BRANCHIFSC
->     AND ACCOUNTS.ACCSTATUS = 'A';
-> END$$
Query OK, 0 rows affected (0.2650 sec)
MySQL localhost:33060+ ssl thebank SQL > DELIMITER ;
MySQL localhost:33060+ ssl thebank SQL >
```