SETTING UP PROCEDURES (CURSORS INCLUDED)

1. DEPOSIT INTEREST UPDATE

This procedure make use of a cursor to fetch all fixed and recurring deposit details from customer accounts view and create records in the deposit interest records. From there triggers update the balance of these accounts.

```
DELIMITER $$
CREATE PROCEDURE DEPOSITINTERESTUPDATEPROC()
BEGIN

DECLARE ACCNO BIGINT;
DECLARE BRANCHIFSC CHAR(11);
DECLARE BALANCE FLOAT;
DECLARE INTERESTRATE FLOAT;
DECLARE FLAG INT DEFAULT 0;

DECLARE DEPOSITCURSOR CURSOR FOR
        SELECT DEPOSITINTERESTUPDATE.ACCOUNTNO,
        DEPOSITINTERESTUPDATE.BRANCHIFSC,
        DEPOSITINTERESTUPDATE.BALANCE
        FROM DEPOSITINTERESTUPDATE;

DECLARE CONTINUE HANDLER
        FOR NOT FOUND SET FLAG = 1;

BEGIN
        SELECT INTERESTINFO.INTERESTRATE INTO INTERESTRATE
        FROM INTERESTINFO
        WHERE LOANTYPE='FIX';

        OPEN DEPOSITCURSOR;

        SELECT INTERESTINFO.INTERESTRATE FROM INTERESTINFO WHERE
        LOANTYPE='FIX';

        LABEL1 :LOOP

                FETCH DEPOSITCURSOR INTO ACCNO, BRANCHIFSC, BALANCE;

                IF FLAG=1 THEN
                        LEAVE LABEL1;
                END IF;

                INSERT INTO DEPOSITINTERESTINFO VALUES
                (
                        ACCNO, BRANCHIFSC, CURDATE(),
                        (BALANCE*(INTERESTRATE/12))/100
                );
```

```
        END LOOP;
        CLOSE DEPOSITCURSOR;
END;
END$$
DELIMITER ;
```


2. LOAN INTEREST UPDATE

This procedure make use of a cursor to fetch all loan details from customer loans view and create records in the loan interest records. From there triggers update the outstanding amount on these loans.

```
DELIMITER $$
CREATE PROCEDURE LOANINTERESTUPDATEPROC()
BEGIN

DECLARE LOANACCNO BIGINT;
DECLARE BRANCHIFSC CHAR(11);
DECLARE OUTSTANDING FLOAT;
DECLARE INTERESTRATE FLOAT;
DECLARE LOANSTATUS CHAR(1);
DECLARE INTEREST FLOAT;
DECLARE FLAG INT DEFAULT 0;

DECLARE LOANCURSOR CURSOR FOR
        SELECT LOANINTERESTUPDATE.LOANACCOUNTNO,
        LOANINTERESTUPDATE.BRANCHIFSC,
        LOANINTERESTUPDATE.OUTSTANDING,
        LOANINTERESTUPDATE.INTERESTRATE,
        LOANINTERESTUPDATE.LOANSTATUS
        FROM LOANINTERESTUPDATE;

DECLARE CONTINUE HANDLER
        FOR NOT FOUND SET FLAG = 1;

BEGIN

        OPEN LOANCURSOR;
```

```
LABEL1 :LOOP

        FETCH LOANCURSOR INTO LOANACCNO, BRANCHIFSC, OUTSTANDING,
        INTERESTRATE, LOANSTATUS;

        IF FLAG=1 THEN
                LEAVE LABEL1;
        END IF;

        IF LOANSTATUS='A' THEN
                INSERT INTO LOANINTERESTRECORD VALUES
                (
                        LOANACCNO, BRANCHIFSC, CURDATE(),
                        (OUTSTANDING*(INTERESTRATE/12))/100
                );
        END IF;
    END LOOP;

    CLOSE LOANCURSOR;
END;
END$$
DELIMITER ;
```



```
DELIMITER $$
CREATE PROCEDURE LOANINTERESTUPDATEPROC()
-> BEGIN
->      DECLARE LOANACCNO BIGINT;
->      DECLARE BRANCHIFSC CHAR(11);
-> DECLARE OUTSTANDING FLOAT;
-> DECLARE INTERESTRATE FLOAT;
-> DECLARE LOANSTATUS CHAR(1);
->      DECLARE INTEREST FLOAT;
->      DECLARE FLAG INT DEFAULT 0;
-> DECLARE LOANCURSOR CURSOR FOR
-> SELECT LOANINTERESTUPDATE.LOANACCOUNTNO,
->          LOANINTERESTUPDATE.BRANCHIFSC,
->          LOANINTERESTUPDATE.OUTSTANDING,
->          LOANINTERESTUPDATE.INTERESTRATE,
->          LOANINTERESTUPDATE.LOANSTATUS
->          FROM LOANINTERESTUPDATE;
-> DECLARE CONTINUE HANDLER
->      FOR NOT FOUND SET FLAG = 1;
-> BEGIN
-> OPEN LOANCURSOR;
->          LABEL1 :LOOP
-> FETCH LOANCURSOR INTO LOANACCNO, BRANCHIFSC, OUTSTANDING, INTERESTRATE, LOANSTATUS;
->              IF FLAG=1 THEN
-> LEAVE LABEL1;
-> END IF;
->              IF LOANSTATUS='A' THEN
-> INSERT INTO LOANINTERESTRECORD VALUES(LOANACCNO, BRANCHIFSC, CURDATE(), (OUTSTANDING*(INTERESTRATE/12))/100);
->              END IF;
-> END LOOP;
->          CLOSE LOANCURSOR;
-> END;
-> END$$
Query OK, 0 rows affected (2.0847 sec)
DELIMITER ;
```