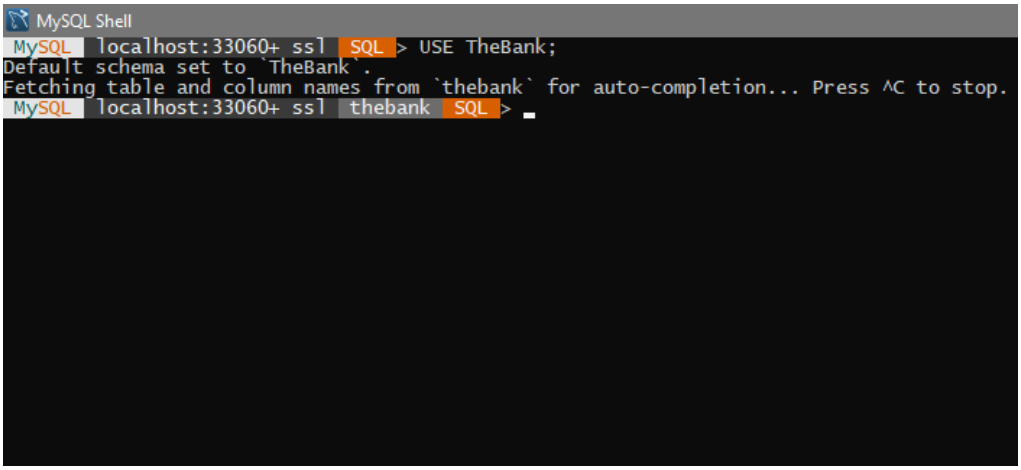
Setting up Relations

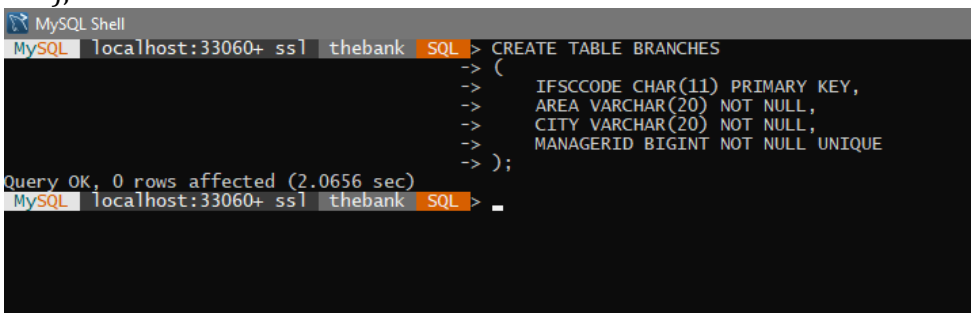
Now, we start creating the relations or tables for our database. After connecting to the database we use " USE TheBank " to make TheBank our current schema.



1. Branches

We create our first table that contains the data on our branches. IFSCCODE is our primary key. ManagerID is unique denoting a person can be a manager to a maximum of one branch. It is also a foreign key to employee table which we will establish later after creation of employee table. We use

```
CREATE TABLE BRANCHES
(
        IFSCCODE CHAR(11) PRIMARY KEY,
        AREA VARCHAR(20) NOT NULL,
        CITY VARCHAR(20) NOT NULL,
        MANAGERID BIGINT NOT NULL UNIQUE
);
```



We add our first branch as IFSCCODE is foreign key for employee relation we just add a MANAGERID value which we will assign to first employee and then apply foreign key constraint.

```
INSERT INTO BRANCHES VALUES
(
    'BANK0001000', 'ANNA UNIVERSITY', 'CHENNAI', 10000
);
```

```
MySQL  localhost:33060+ ssl  thebank  SQL > INSERT INTO BRANCHES VALUES
                                        -> (
                                        ->     'BANK0000001', 'ANNA UNIVERSITY', 'CHENNAI', 10000
                                        -> );
Query OK, 1 row affected (0.0858 sec)
MySQL  localhost:33060+ ssl  thebank  SQL >
```

2. Employees

We setup our employee table with EmployeeID as primary key and it is auto-incremented. So we will add the first record manually. After this we will have to setup the foreign key constraint on IFSCCODE.

```
CREATE TABLE EMPLOYEES
(
        EMPLOYEEID BIGINT PRIMARY KEY AUTO_INCREMENT,
        EMPLOYEENAME VARCHAR(30) NOT NULL,
         SEX CHAR(1) NOT NULL,
         DATEOFBIRTH DATE NOT NULL,
        BRANCHIFSC CHAR(11) NOT NULL,
        DESIGNATION VARCHAR(20) NOT NULL,
        SALARY BIGINT NOT NULL,
        LOGPASSWORD VARCHAR(100) NOT NULL,
        ADDRESSLINE1 VARCHAR(50) NOT NULL,
        ADDRESSLINE2 VARCHAR(50),
        CITY VARCHAR(20) NOT NULL,
        POSTALCODE BIGINT NOT NULL
);
```

We add the foreign key constraint using

```
ALTER TABLE EMPLOYEES
ADD FOREIGN KEY (BRANCHIFSC)
REFERENCES BRANCHES (IFSCCODE);
```

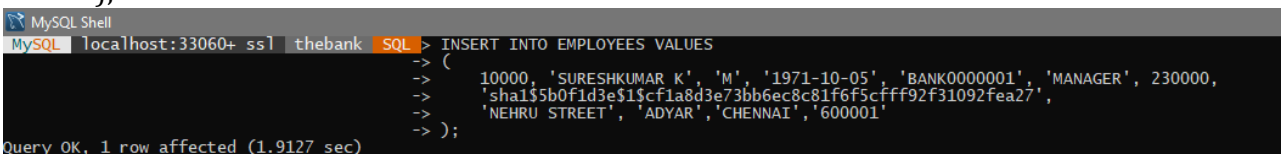
Finally we add the constraint to ensure that sex is always ‘M’ for male or ‘F’ for female.

```
ALTER TABLE EMPLOYEES
ADD CONSTRAINT CHEECK_SEX CHECK (SEX IN ('M','F','O'));
```

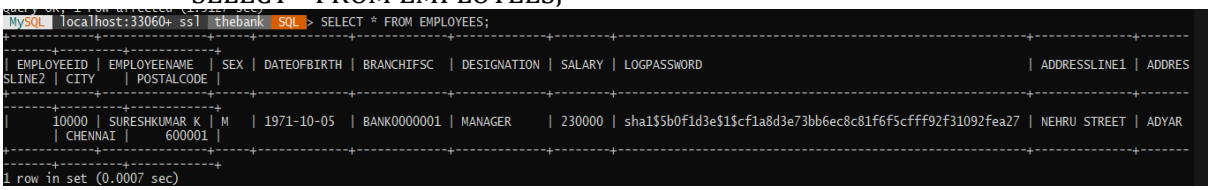
```
MySQL Shell
MySQL  localhost:33060+ ssl  thebank  SQL > CREATE TABLE EMPLOYEES
                                        -> (
                                        -> EMPLOYEEID BIGINT PRIMARY KEY AUTO_INCREMENT,
                                        ->     EMPLOYEENAME VARCHAR(30) NOT NULL,
                                        ->     SEX CHAR(1) NOT NULL,
                                        ->     DATEOFBIRTH DATE NOT NULL,
                                        ->     BRANCHIFSC CHAR(11) NOT NULL,
                                        ->     DESIGNATION VARCHAR(20) NOT NULL,
                                        ->     SALARY BIGINT NOT NULL,
                                        ->     LOGPASSWORD VARCHAR(100) NOT NULL,
                                        ->     ADDRESSLINE1 VARCHAR(50) NOT NULL,
                                        ->     ADDRESSLINE2 VARCHAR(50),
                                        ->     CITY VARCHAR(20) NOT NULL,
                                        ->     POSTALCODE BIGINT NOT NULL
                                        -> );
Query OK, 0 rows affected (2.4355 sec)
MySQL  localhost:33060+ ssl  thebank  SQL > ALTER TABLE EMPLOYEES
                                        -> ADD FOREIGN KEY (BRANCHIFSC)
                                        -> REFERENCES BRANCHES (IFSCCODE);
Query OK, 0 rows affected (2.8491 sec)

Records: 0  Duplicates: 0  Warnings: 0
MySQL  localhost:33060+ ssl  thebank  SQL > ALTER TABLE EMPLOYEES
                                        -> ADD CONSTRAINT CHEECK_SEX CHECK (SEX IN ('M','F','O'));
Query OK, 0 rows affected (1.9480 sec)

Records: 0  Duplicates: 0  Warnings: 0
```

We insert our first record. THE PASSWORD IS HASHED. The value we use is the hash value for ‘ADMIN’. Hashing of text password is done by password-hash module in NodeJS.

```
INSERT INTO EMPLOYEES VALUES
(
        10000, 'SURESHKUMAR K', 'M', '1971-10-05', 'BANK0001000', 'MANAGER',
        230000,
         ' sha1$5b0f1d3e$1$cf1a8d3e73bb6ec8c81f6f5cfff92f31092fea27',
        'NEHRU STREET', 'ADYAR','CHENNAI','600001'
);
```




```
SELECT * FROM EMPLOYEES;
```



We now fix foreign key constraint on BRANCHES.

```
ALTER TABLE BRANCHES
ADD FOREIGN KEY (MANAGERID)
REFERENCES EMPLOYEES (EMPLOYEEID);
```

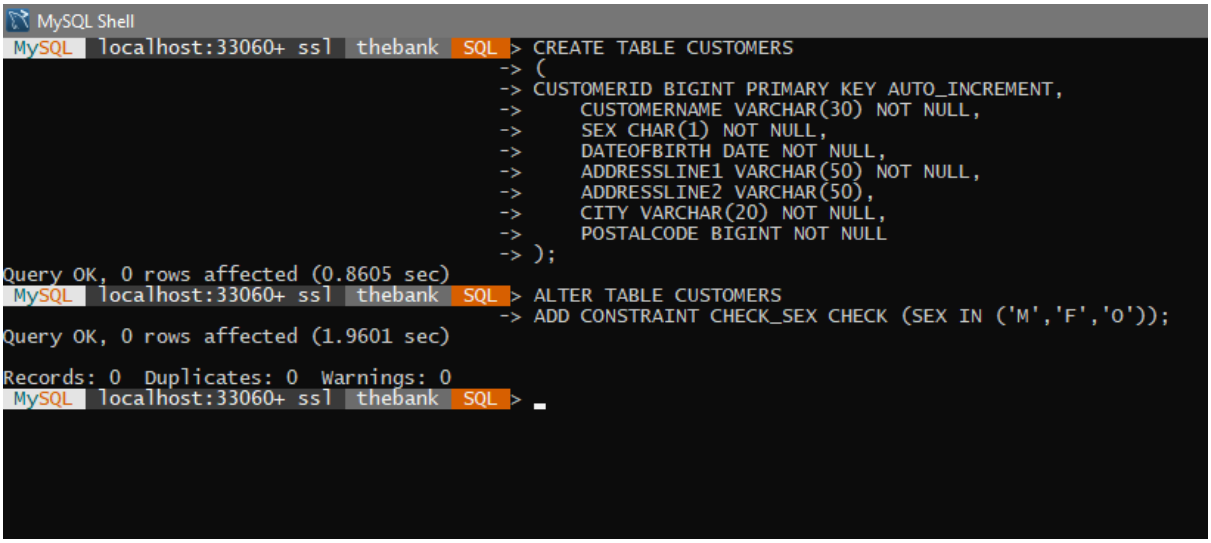
3. Customers

We add the customers relations.

```
CREATE TABLE CUSTOMERS
(
        CUSTOMERID BIGINT PRIMARY KEY AUTO_INCREMENT,
        CUSTOMERNAME VARCHAR(30) NOT NULL,
        SEX CHAR(1) NOT NULL,
        DATEOFBIRTH DATE NOT NULL,
        ADDRESSLINE1 VARCHAR(50) NOT NULL,
        ADDRESSLINE2 VARCHAR(50),
        CITY VARCHAR(20) NOT NULL,
        POSTALCODE BIGINT NOT NULL
);
```

Adding constraint to check gender

```
ALTER TABLE CUSTOMERS
ADD CONSTRAINT CHECK_SEX CHECK (SEX IN ('M','F','O'));
```



We insert our first record manually to give the initial value of CUSTOMERID for auto_increment to function properly.

```
INSERT INTO CUSTOMERS VALUES
(
        10000, 'SURESHKUMAR K', 'M', '1971-10-05', 'NEHRU STREET', 'ADYAR', 'CHENNAI',
        600001
);
```

4. Accounts

Creating relation

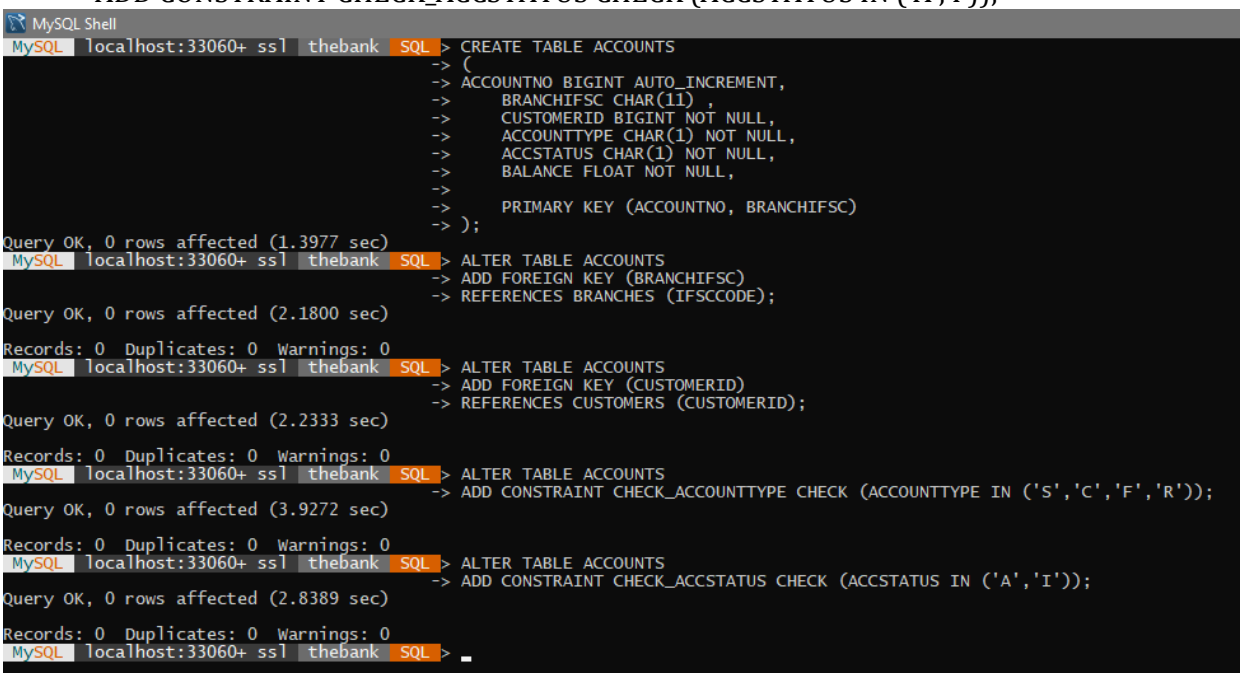
```
        CREATE TABLE ACCOUNTS
        (
                ACCOUNTNO BIGINT AUTO_INCREMENT,
                BRANCHIFSC CHAR(11) ,
                CUSTOMERID BIGINT NOT NULL,
                ACCOUNTTYPE CHAR(1) NOT NULL,
                ACCSTATUS CHAR(1) NOT NULL,
                BALANCE FLOAT NOT NULL,

                PRIMARY KEY (ACCOUNTNO, BRANCHIFSC)
        );
```

Adding foreign key constraints:

```
        ALTER TABLE ACCOUNTS
        ADD FOREIGN KEY (BRANCHIFSC)
        REFERENCES BRANCHES (IFSCCODE);


        ALTER TABLE ACCOUNTS
        ADD FOREIGN KEY (CUSTOMERID)
        REFERENCES CUSTOMERS (CUSTOMERID);
```

Adding constraints on account type which is S – Savings account, C – Current Account, F – Fixed Deposit Account, R – Recurring Deposit Account and on Account status which is A – Active and I for Inactive

```
ALTER TABLE ACCOUNTS
ADD CONSTRAINT CHECK_ACCOUNTTYPE CHECK (ACCOUNTTYPE IN ('S','C','F','R'));


ALTER TABLE ACCOUNTS
ADD CONSTRAINT CHECK_ACCSTATUS CHECK (ACCSTATUS IN ('A','I'));
```



Inserting first record

```
INSERT INTO ACCOUNTS VALUES
(
10000, 'BANK0000001', 10000, 'S', 'A', 0
);
```

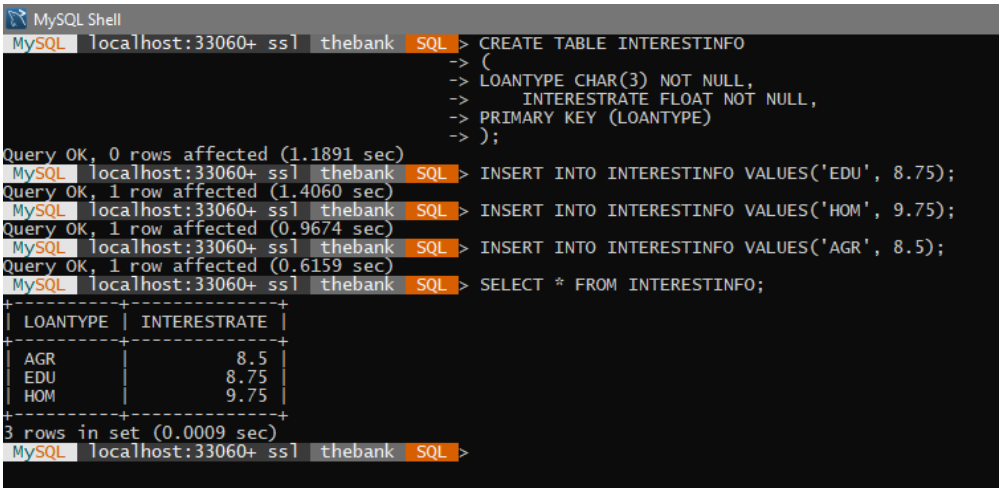
5. Interest Information

Create relation

```
CREATE TABLE INTERESTINFO
(
        LOANTYPE CHAR(3) NOT NULL,
        INTERESTRATE FLOAT NOT NULL,
        PRIMARY KEY (LOANTYPE)
);
```

Adding values manually as these values can be altered only by admin

```
INSERT INTO INTERESTINFO VALUES('EDU', 8.75);
INSERT INTO INTERESTINFO VALUES('HOM', 9.75);
INSERT INTO INTERESTINFO VALUES('AGR', 8.5);
```



6. Loans

Create relation

```
CREATE TABLE LOANS
(
	LOANACCOUNTNO BIGINT NOT NULL AUTO_INCREMENT,
	 BRANCHIFSC CHAR(11) NOT NULL,
	CUSTOMERID BIGINT NOT NULL,
	PRINCIPLE FLOAT NOT NULL,
	INTERESTAMT FLOAT,
	OUTSTANDING FLOAT NOT NULL,
	LOANSTATUS CHAR(1) NOT NULL,
	LOANSECURITY VARCHAR(20) NOT NULL,
	LOANTYPE CHAR(3) NOT NULL,
	APPROVEDBY BIGINT NOT NULL,
	PRIMARY KEY (LOANACCOUNTNO, BRANCHIFSC)
);
```

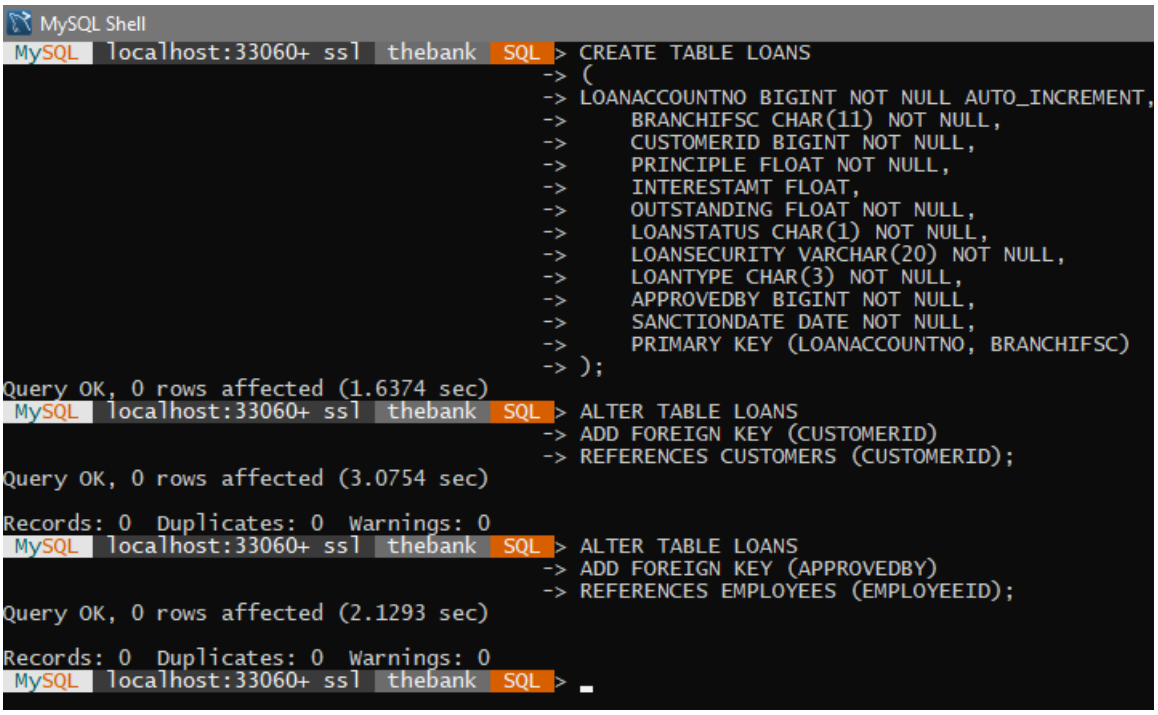
Foreign key constraints

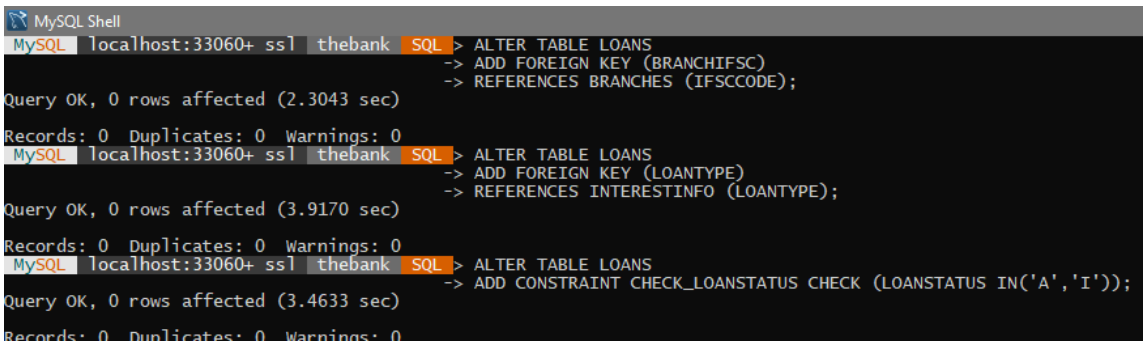
```
ALTER TABLE LOANS
ADD FOREIGN KEY (CUSTOMERID)
REFERENCES CUSTOMERS (CUSTOMERID);

ALTER TABLE LOANS
ADD FOREIGN KEY (APPROVEDBY)
REFERENCES EMPLOYEES (EMPLOYEEID);

ALTER TABLE LOANS
ADD FOREIGN KEY (BRANCHIFSC)
REFERENCES BRANCHES (IFSCCODE);

ALTER TABLE LOANS
ADD FOREIGN KEY (LOANTYPE)
REFERENCES INTERESTINFO (LOANTYPE);

ALTER TABLE LOANS
```

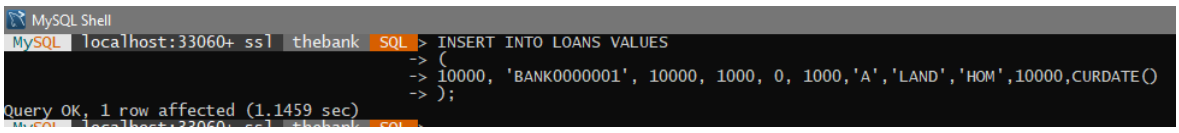
ADD CONSTRAINT CHECK_LOANSTATUS CHECK (LOANSTATUS IN('A','I'));

Inserting first record

```
INSERT INTO LOANS VALUES
(
10000, 'BANK0001000', 10000, 1000, 0, 1000,'A','LAND','HOM',10000,CURDATE()
);
```







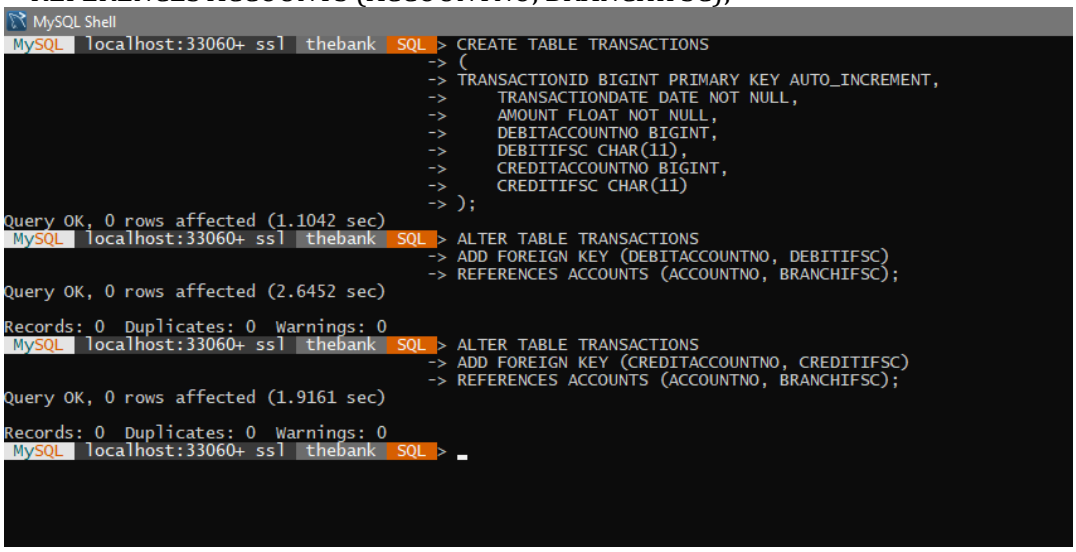
7. Transactions

Creating table

```
CREATE TABLE TRANSACTIONS
(
        TRANSACTIONID BIGINT PRIMARY KEY AUTO_INCREMENT,
        TRANSACTIONDATE DATE NOT NULL,
        AMOUNT FLOAT NOT NULL,
        DEBITACCOUNTNO BIGINT,
        DEBITIFSC CHAR(11),
        CREDITACCOUNTNO BIGINT,
        CREDITIFSC CHAR(11)
);
```

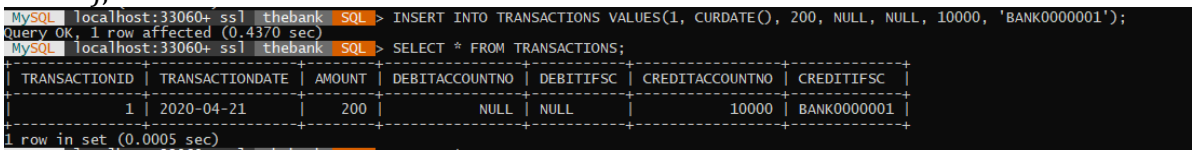
Adding foreign key constraint

```
ALTER TABLE TRANSACTIONS
ADD FOREIGN KEY (DEBITACCOUNTNO, DEBITIFSC)
REFERENCES ACCOUNTS (ACCOUNTNO, BRANCHIFSC);

ALTER TABLE TRANSACTIONS
ADD FOREIGN KEY (CREDITACCOUNTNO, CREDITIFSC)
REFERENCES ACCOUNTS (ACCOUNTNO, BRANCHIFSC);
```



Inserting the first record. Do this after setup the TRANSACTION_BALANCE_UPDATE Trigger to update balance.

```
INSERT INTO TRANSACTIONS VALUES
(
        10000 , CURDATE(),     200, NULL, NULL, 10000, 'BANK0000001'
);
```



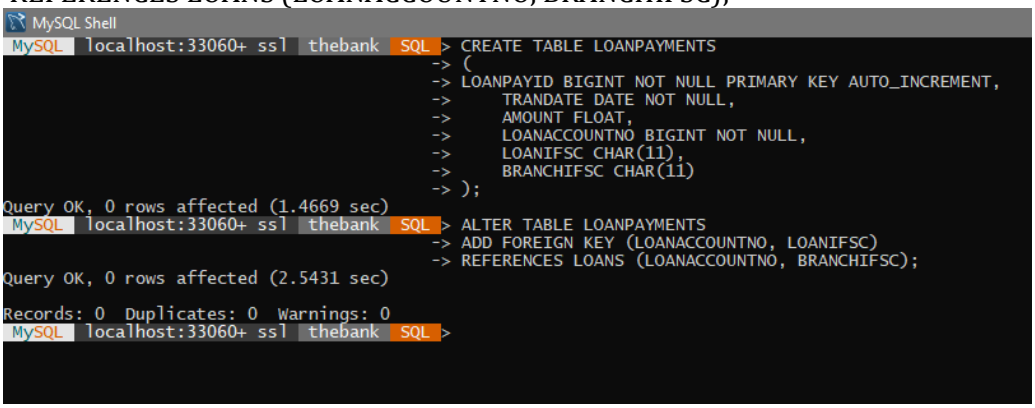
8. Loan Payments

Creating table

```
CREATE TABLE LOANPAYMENTS
(
        LOANPAYID BIGINT NOT NULL PRIMARY KEY AUTO_INCREMENT,
        TRANDATE DATE NOT NULL,
        AMOUNT FLOAT,
        LOANACCOUNTNO BIGINT NOT NULL,
        LOANIFSC CHAR(11),
        BRANCHIFSC CHAR(11)
);
```

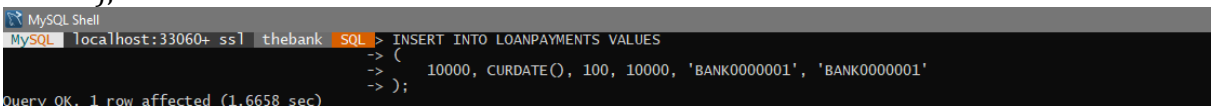
Adding Foreign key constraint

```
ALTER TABLE LOANPAYMENTS
ADD FOREIGN KEY (LOANACCOUNTNO, LOANIFSC)
REFERENCES LOANS (LOANACCOUNTNO, BRANCHIFSC);
```



Adding first record manually. . Do this after setup the TRANSACTION_BALANCE_UPDATE Trigger to update balance.

```
INSERT INTO LOANPAYMENTS VALUES
(
        10000, CURDATE(),  100, 10000, 'BANK0000001', 'BANK0000001'
);
```



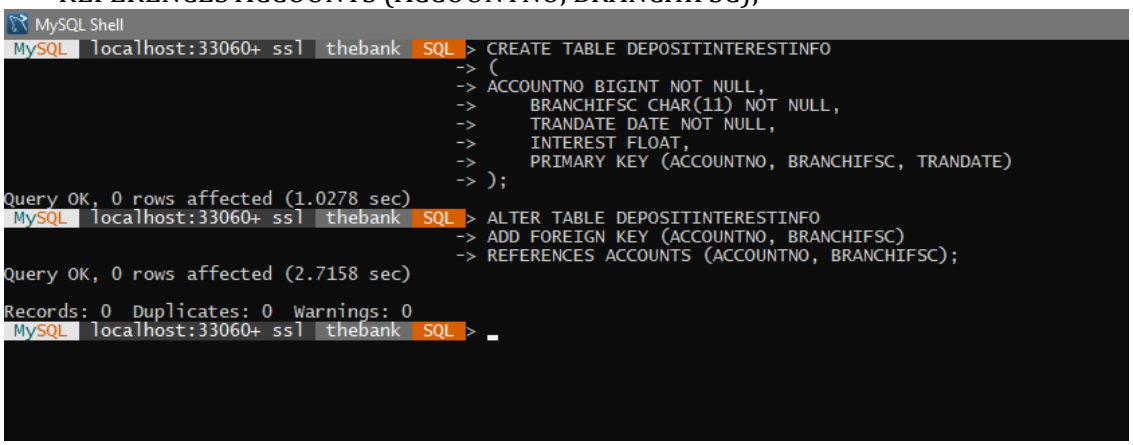
9. Deposit Interest Record (RECORDS FOR THIS TABLES ARE ADDED ONLY BY PROCEDURES WHEN INTERESTS ON DEPOSITS ARE CALCULATED PERIODICALLY)

```
CREATE TABLE DEPOSITINTERESTINFO
(
        ACCOUNTNO BIGINT NOT NULL,
        BRANCHIFSC CHAR(11) NOT NULL,
        TRANDATE DATE NOT NULL,
        INTEREST FLOAT,
        PRIMARY KEY (ACCOUNTNO, BRANCHIFSC, TRANDATE)
```

);

Adding foreign key constraint

```
ALTER TABLE DEPOSITINTERESTINFO
ADD FOREIGN KEY (ACCOUNTNO, BRANCHIFSC)
REFERENCES ACCOUNTS (ACCOUNTNO, BRANCHIFSC);
```



10. Loan Interest Record (RECORDS FOR THIS TABLES ARE ADDED ONLY BY PROCEDURES WHEN INTERESTS ON LOANS ARE CALCULATED PERIODICALLY)

Creating table

```
CREATE TABLE LOANINTERESTRECORD
(
        LOANACCOUNTNO BIGINT NOT NULL,
        BRANCHIFSC CHAR(11) NOT NULL,
        TRANDATE DATE NOT NULL,
        INTEREST FLOAT,
        PRIMARY KEY (LOANACCOUNTNO, BRANCHIFSC, TRANDATE)
);
```

Adding foreign key constraints

```
ALTER TABLE LOANINTERESTRECORD
ADD FOREIGN KEY (LOANACCOUNTNO, BRANCHIFSC)
REFERENCES LOANS (LOANACCOUNTNO, BRANCHIFSC);
```