Course No: CSE 3110

Course Name: Database Systems Laboratory

Project Name: **Banking System**

Submitted to:

**Jarin Firose Moon Md. Milon Islam**

Lecturer Lecturer

Dept of Computer Science & Engineering Dept of Computer Science & Engineering

Khulna University of Engineering & Technology Khulna University of Engineering & Technology

Submitted by:

Salim Shadman Ankur

Dept of Computer Science & Engineering

Roll: 1507014 Section: A

Khulna University of Engineering & Technology

PROJECT OVERVIEW

* The main aim of this project is it gives information about the Functioning of Databases in Banking System.
* All the Functionalities of Banking Systems are credited in this project.
* Daily Transaction
* Employee Management
* Loan System

DESCRIPTION

**Janata Bank** is a bank in Bangladesh. There are many branches of this bank throughout the country. There are a lot of transactions in a day . To store it , they need a database. This project will be on that . It includes day by day transaction, loan system, credit card system. There will be a few table. Like a **Cutomer table**, **Loan table**, **Employee table**, **Branch table.**

Mainly **Branch Table** defines the branch which we apply our database of this bank. The attributes are branch\_name , city, assets. Branch\_name is the primary key of this table.

Again there is another table for Customer . **Customer Table** includes all the information about customer. The attributes are account\_no, name, branch\_name, date\_of\_birth, gender, balance, phone\_no, account\_type. Account\_no is the primary key of this table. There is M:M relationship between Customer & Branch table. One customer may have more than one account in a branch. On the basis of account\_no , one can easily find a customer.

There is another table of Bank employee. **Employee Table** includes all the information about an employee of a bank . The attributes are employee\_id, name, status, branch\_name, salary, phone\_no, gender. Employee\_id is the primary key of the table. There is M:1 relationship between Employee & Branch table.

Last table is **Loan Table**. All loan transaction are included in this table. The attributes are loan\_no, account\_no, loan\_type, amount, interest, entry\_clerk\_id, access\_date. Loan\_no is the primary key of this table. There is M:1 relationship between Loan & Customer table. Also M:1 relationship happens between Loan & Employee table.

Again transaction system, loan\_repay, tax system have been included.

RULES GOVERNING THE PROJECT

* All the customers of the bank have a **unique** account number
* The account numbers are not nullable i.e., they cannot take **null** values.
* The customers must have a **minimum** account balance of Rs.**500**.
* Any customer is not allowed to withdraw amount from his account, if the withdrawal results in his account balance going below the minimum balance.
* A person is eligible to get a loan from the bank if he has an account in the bank.
* The percentage of interest imposed on the loan depends on the Company’s policies.
* The employees of the bank have **unique** identification numbers.

SCOPE

The scope of the project contains following features:

* It is a database management system aimed at computerizing and automating the working of bank managements.
* It provides a platform for comprehensive data maintenance and information access.
* It is simple and user friendly with an easy to use graphical interface.
* It is networked DBMS hence allows access to large amount of information.

DATABASE STRUCTURE

The database consists of four tables. They are branch, customer, employee and loan tables.

|  |  |
| --- | --- |
|  |  |
| **Database Tables** | **Description** |
| BRANCH | Contains all the branch information. |
| CUSTOMER | Contains information about customer. |
| EMPLOYEE | Contains information about employee. |
| LOAN | Contains details of loan. |

FUNCTIONALITY

The database system allows technicians to complete the following functions:

* Enter new customers and employees.
* Show the details of all customer and employee.
* Modify an existing one

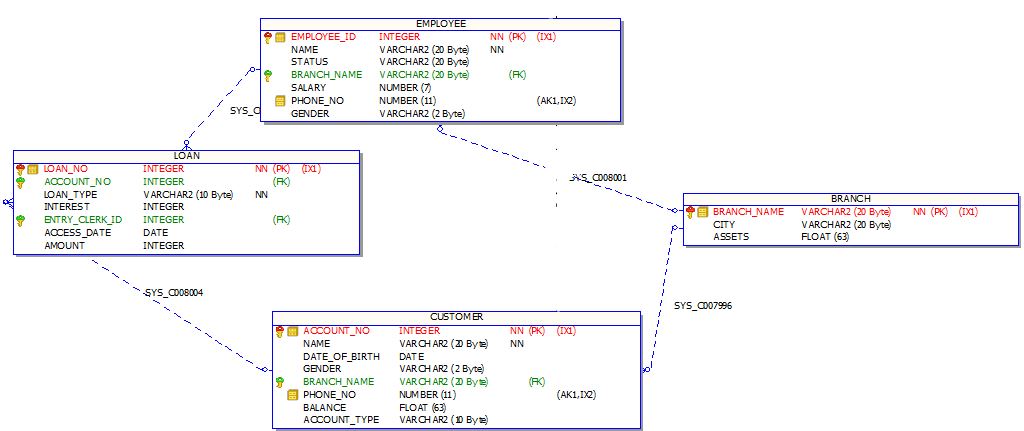
The database is password-protected and requires a login (when we use this database in web application software).

ENTITTIES AND ATTRIBUTES

* **Branch**
* Branch\_name , City, Assets
* **Customer**
* Account\_no, Name, Branch\_name, Date\_of\_birth, Gender, Balance, Phone\_no, Account\_type
* **Employee**
* Employee\_id, Name, Status, Branch\_name, Salary, Phone\_no, Gender

* **Loan**
* Loan\_no, Account\_no, Loan\_type, Amount, Interest, Entry\_clerk\_id, Access\_date

ER Diagram



SCHEMA DIAGRAM

Branch

|  |
| --- |
| Branch\_name |
| City |
| Assets |

Customer

|  |
| --- |
| Account\_no |
| Name |
| Date\_Of\_Birth |
| Gender |
| Branch\_name |
| Phone\_no |
| Balance |
| Account\_Type |

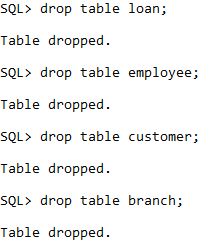
Employee Loan

|  |
| --- |
| Employee\_ID |
| Name |
| Status |
| Branch\_name |
| Salary |
| Phone\_no |
| Gender |

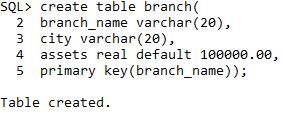
|  |
| --- |
| Loan\_no |
| Account\_no |
| Loan\_Type |
| Interest |
| Entry\_Clerk\_ID |
| Access\_Date |
| Amount |

CREATE TABLE

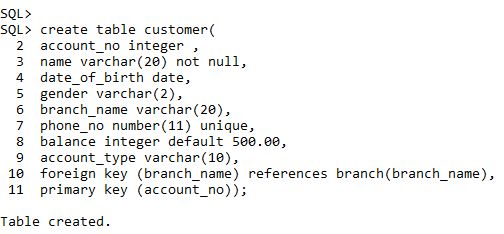
* Firstly, drop those table if they already have been.



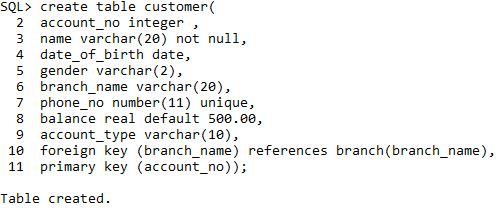
* Create Branch Table

****

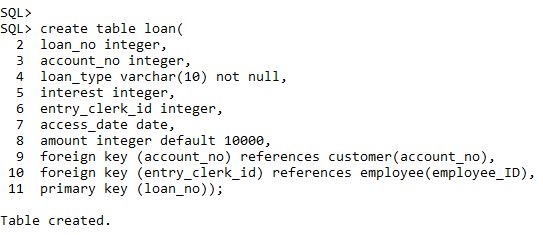
* Create Customer Table



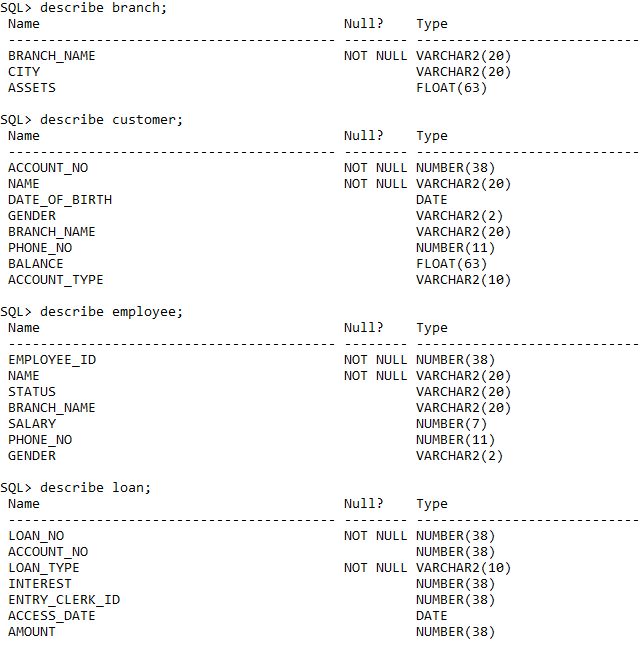
* Create Employee Table



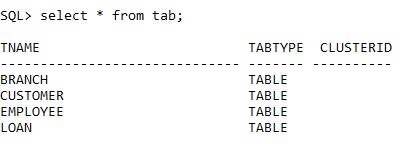
* Create Loan Table



* Describe Tables



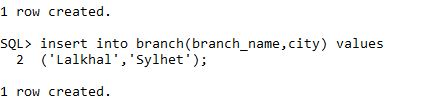
* Select \* from tab;



INSERT DATA INTO TABLES

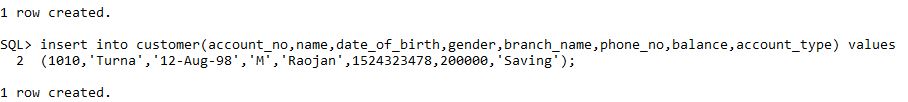
* Insert data into Branch Table





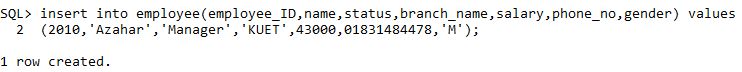
* Insert data into Customer Table



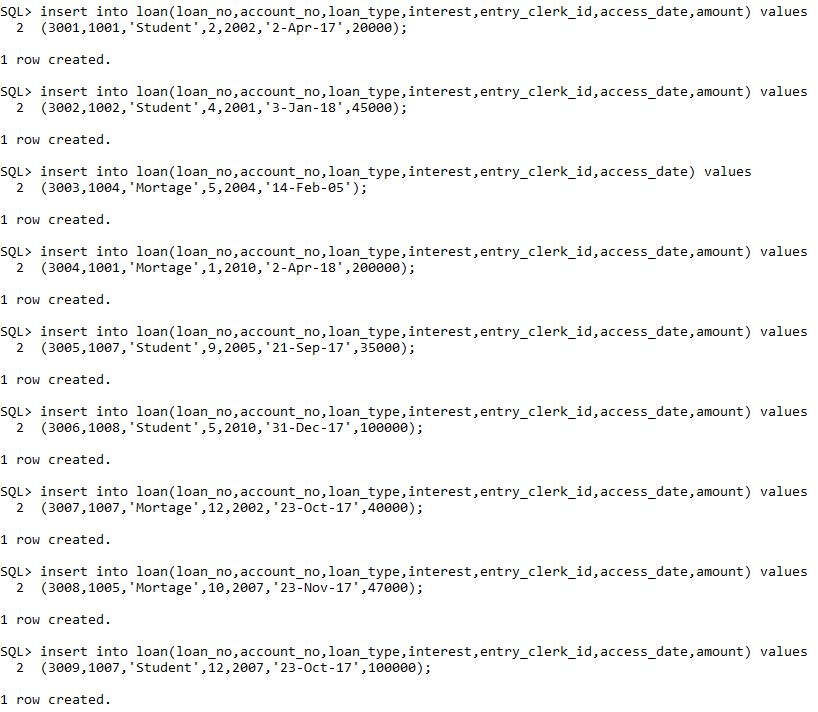


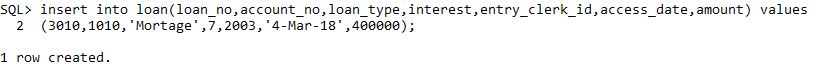
* Insert data into Employee Table





* Insert Data into Loan Table

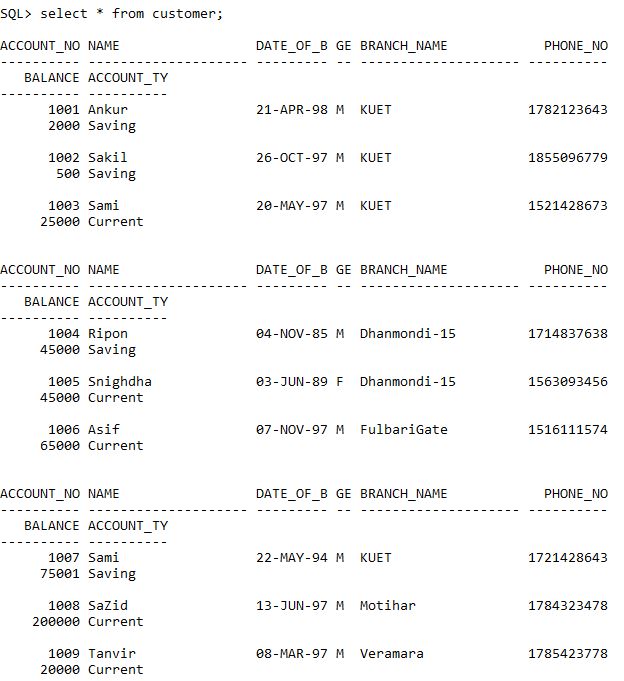


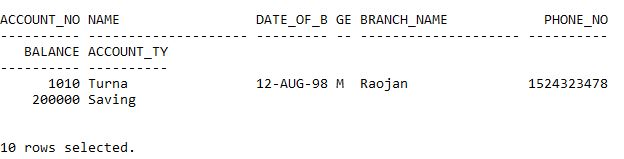


* Select \* from branch;

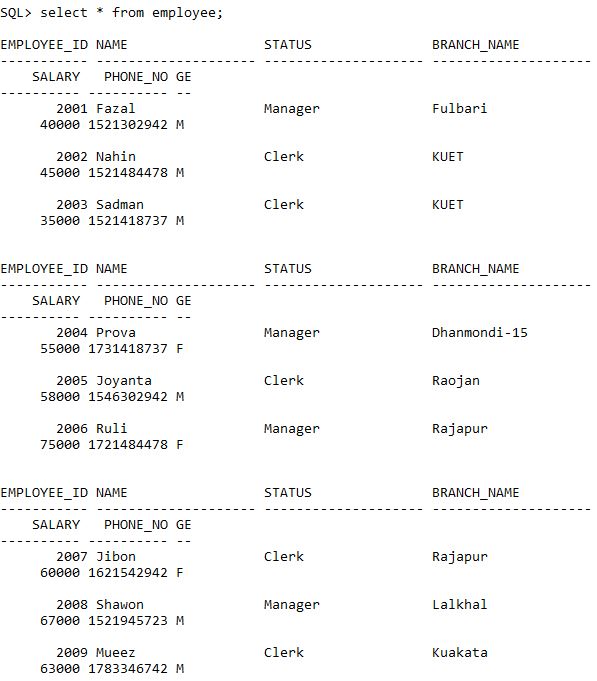


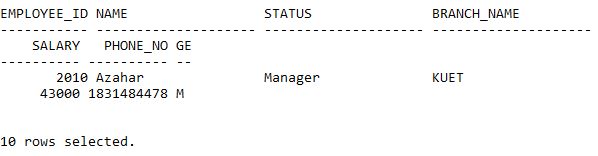
* Select \* from Customer;



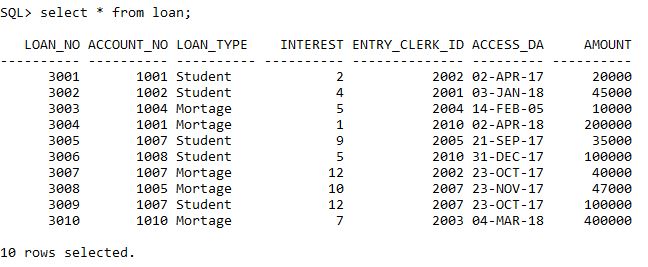


* Select \* from Employee;



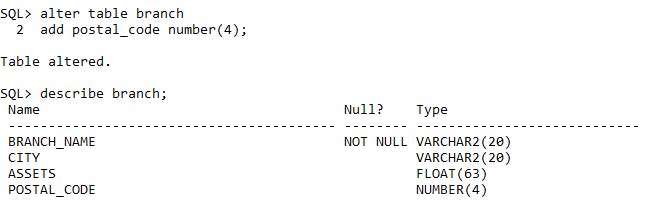


* Select \* from Loan;

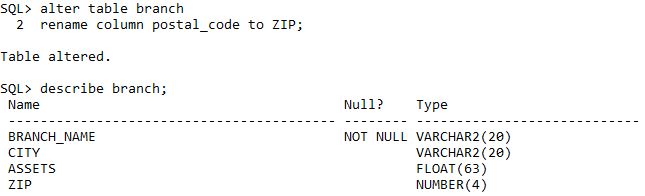


ALTER OPERATION

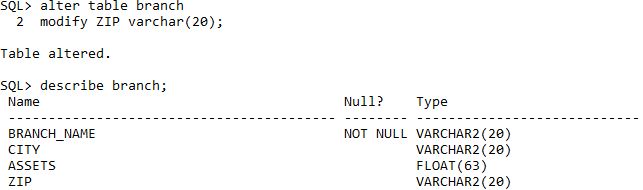
* Adding Column Postal\_Code in Branch Table



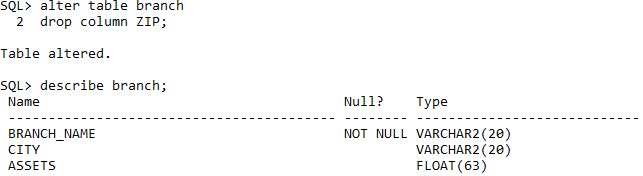
* Renaming Column Postal\_Code to ZIP



* Modifying ZIP column of Branch Table

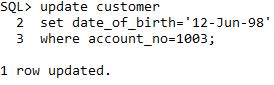


* Dropping ZIP column from Branch Table



UPDATE OPERATION

* Updating Customer’s Date of Birth



1. Before Update



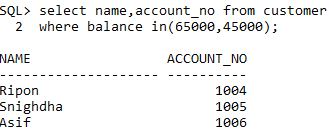
1. After Update



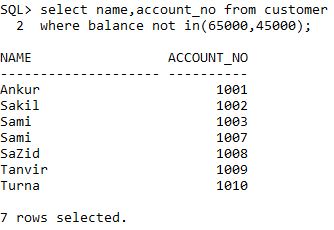
SET MEMBERSHIP

-checks the condition which exactly matches

* Checks customers’ whose balance are 65000 & 45000



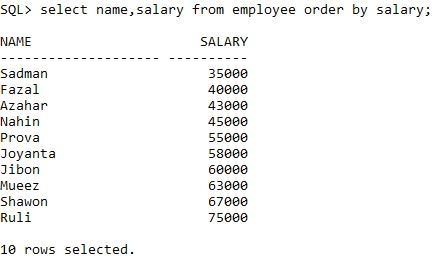
* Checks customers’ whose balance are not 65000 & 45000



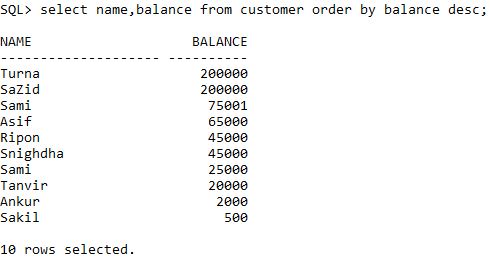
SINGLE COLUMN ORDERING

-Sorts the table based on attributes

* Sort the Employee Table based on their salary in ascending order



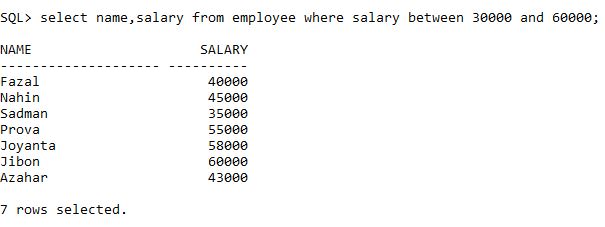
* Sort the Customer Table based on their balance in descending order



RANGE SEARCH CONDITION

-Used to concatenate two condition in sql

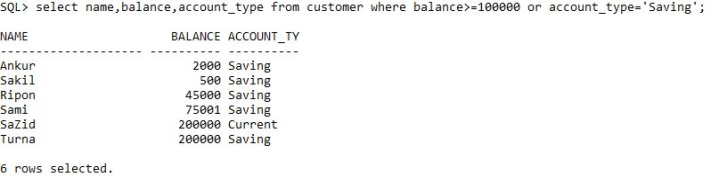
* Find out the Employee whose salary range is 30000 to 60000



COMPOUND COMPARISON SEARCH CONDITION

-Works when one or more condition from multiple condition is true

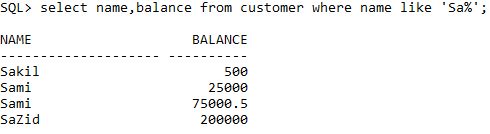
* Find out the customer whose account type is saving or have balance > 100000



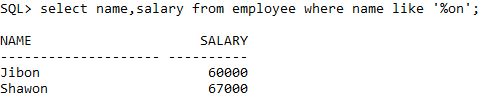
PATTERN MATCHING

- Identifies using some pattern

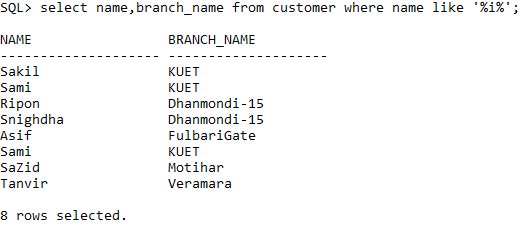
* Left Pattern



* Right Pattern

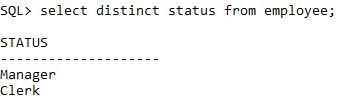


* Middle Pattern



USE OF DISTINCT

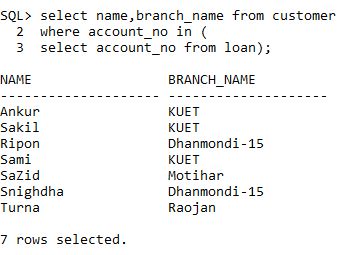
* Shows the categories of status of employee



SUB-QUERY

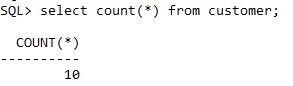
-Used for complex transaction

* Find out the customers who have loans

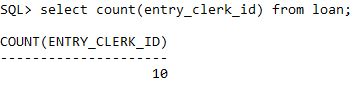


AGGREGATE FUNCTION

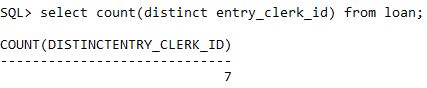
* **COUNT()**
* Counts the number of customer



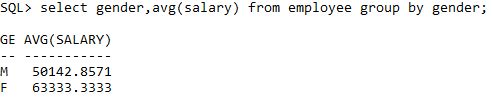
* Counts the number of clerks who entries loan . (It won’t remove duplication)



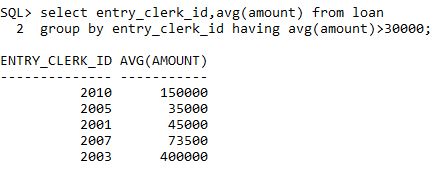
* Counts the unique number of clerks who entries loan . (Remove duplication)



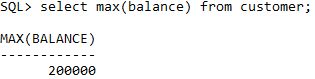
* **AVG()**
* Calculates average salary based on gender of employee

****

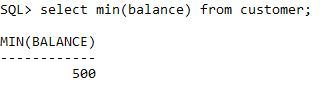
* Use of having

****

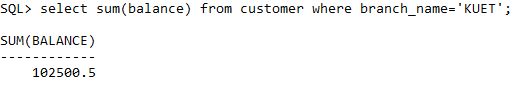
* **Max()**
* Return the maximum value



* **Min()**
* Return the minimum value

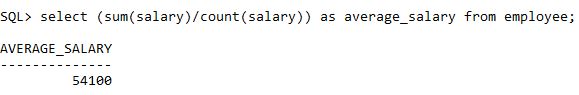


* **Sum()**
* Returns the summation



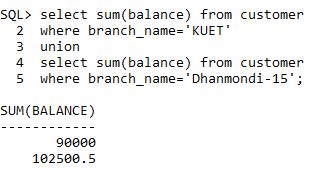
CALCULATED FIELD GIVING DIFFERENT COLUMN NAME

* Calculating average salary giving the name average\_salary

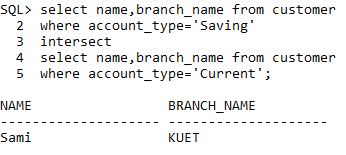


SET ORERATION

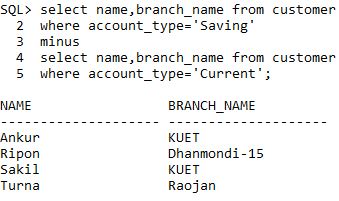
* **Union**
* Merges two branches’ sum of customer balance



* **Intersection**
* Find out the customers who have both account of current & saving

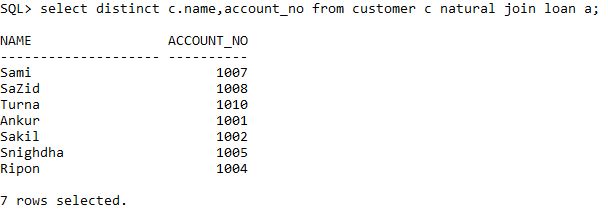


* **Minus**
* Find out the customers who have only an account

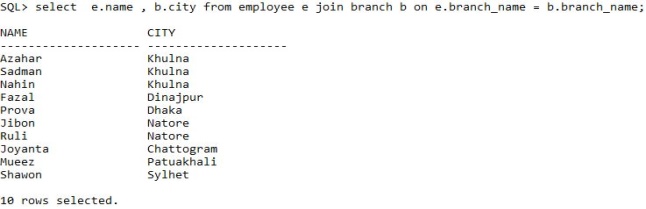
****

JOIN OPERATION

* Find out the customers names who have loans

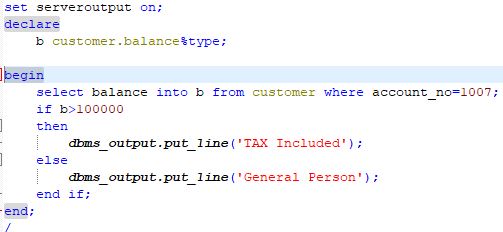


* Find out employees city



CONDITION

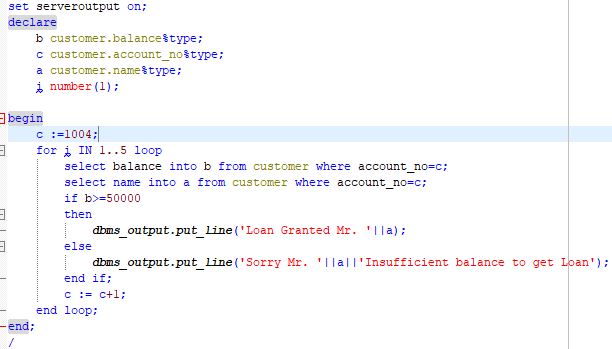
* Taxation System

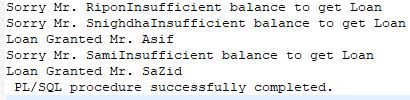




LOOP

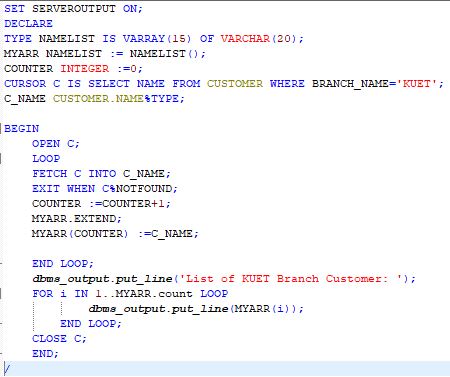
* Loan Application

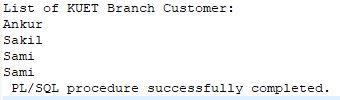




ARRAY

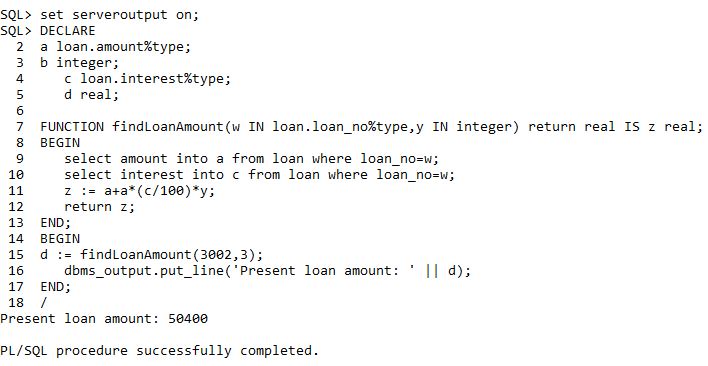
* Branch Customers List



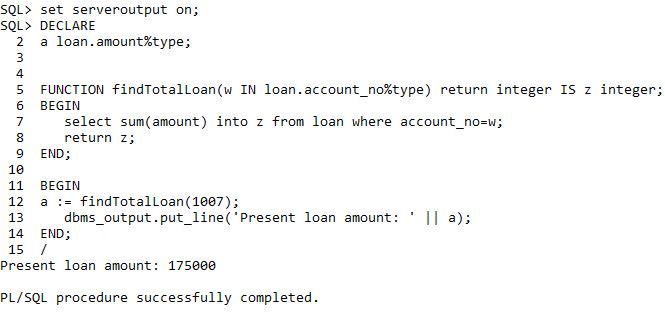


PROCEDURE/FUNCTION

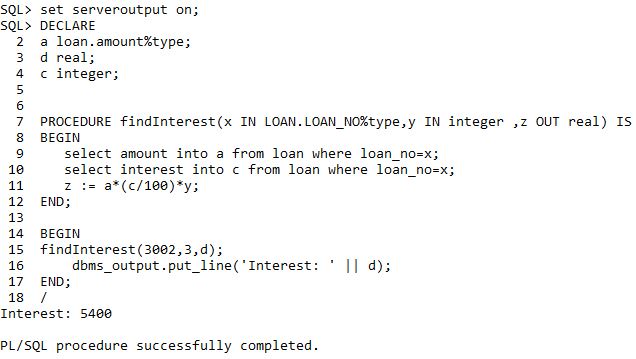
* Find out the present amount of loan after some years



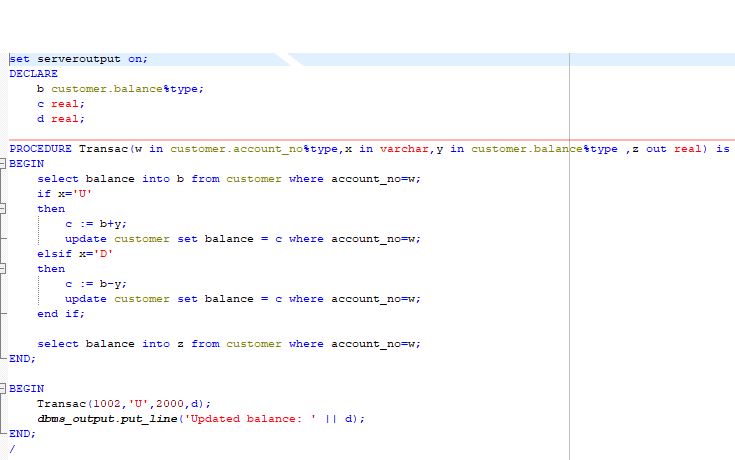
* Find out the total of a customer



* Find interest

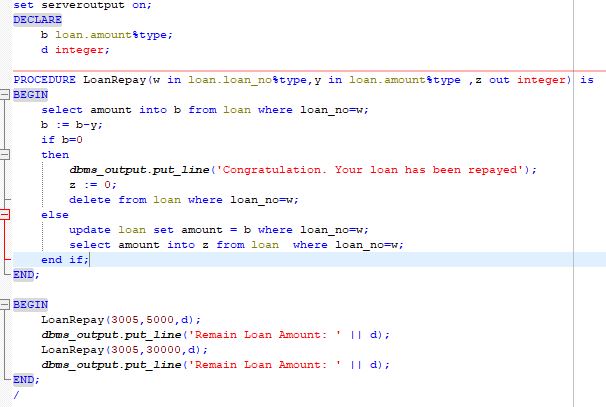


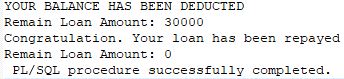
* Transaction System



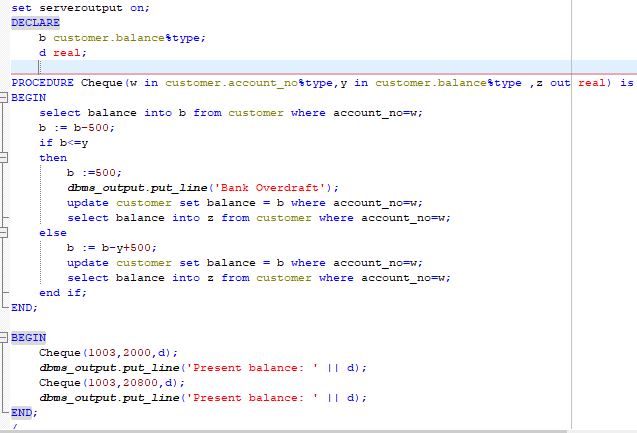


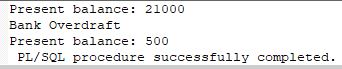
* Loan Repay System





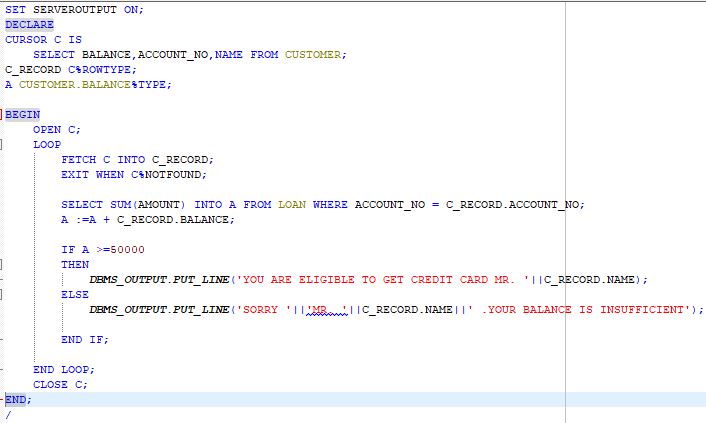
* Cheque Book

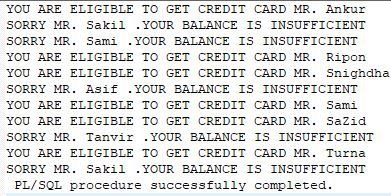




CURSOR

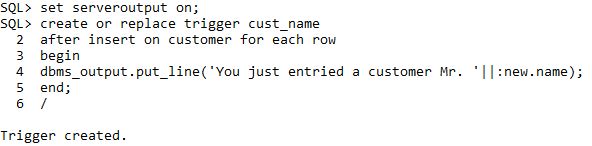
* Credit card system



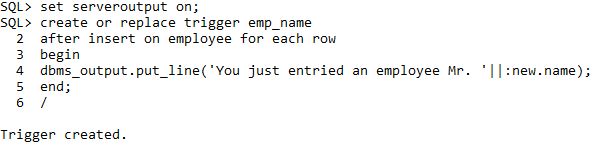


TRIGGER

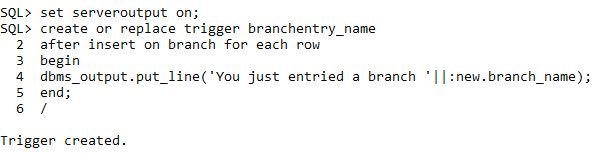
* Automatically Runs when insert or delete or update
* To entry a customer



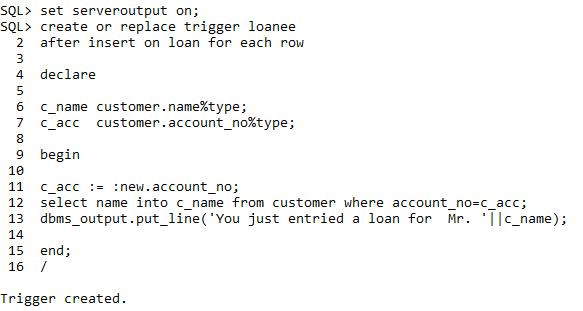
* To entry an employee



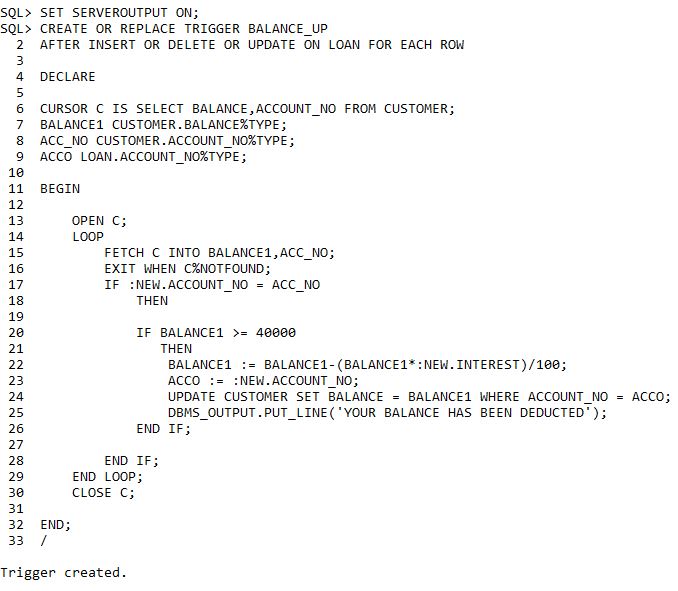
* To entry a branch



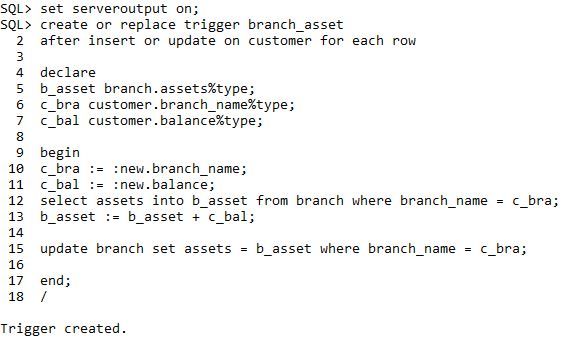
* To entry a loan



* First year interest deduction
* Will deduct the interest of first year while loan transaction occurs

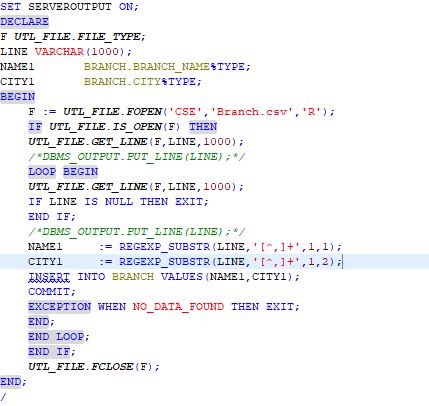


* Branch Asset Calculation
* Will update the bank asset while transaction occurs



FILE SYSTEM (READ)

* Read Branch Table



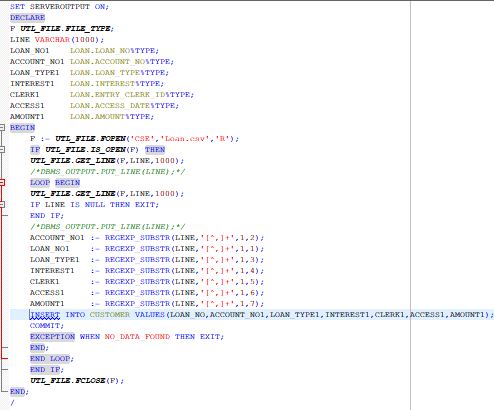
* Read Customer Table



* Read Employee table

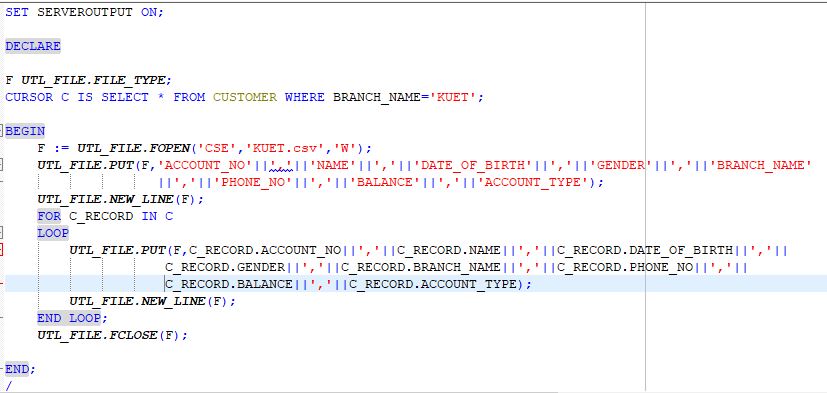


* Read Loan table



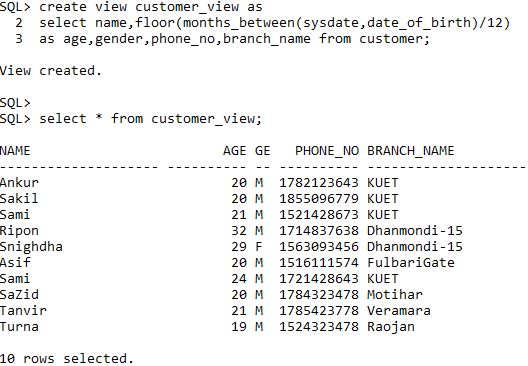
FILE SYSTEM (WRITE)

* Write individual branch information



VIEW

* Customer View



SUMMARY

The project was a learning experience for all people who have their transaction. It will help the bank employee & authorities to the transaction.