



BANK MANAGEMENT SYSTEM

GROUP MEMBERS -

Anshul Dadhwal (102003398), Sanchay Sindhwani (102003496), Divyam Jain (102003501)

Table of Contents

Problem Statement.....	2
ER Diagram	3
ER to Table.....	4
Normalization (Normalized Table).....	5
Unnormalized Table	5
Normalized Table	5
Account.....	5
Transaction	5
PL/SQL code to implement project functionalities	6
Output Screenshots	11
Table Creation.....	11
TABLES.....	12
Bank.....	12
Branch.....	12
Employee	13
Accounts	14
Transaction	15
Customer	16
Loan	17
Customer Avails Loans.....	18
Function Creation.....	19
FUNCTIONS	19
Total Customers	19
Total Loans	20
Total Accounts	21
Output Screen for Function	21
Conclusion	22

Problem Statement

A bank is a financial institution licensed to receive deposits and make loans. Banks may also provide financial services such as wealth management, currency exchange, and safe deposit boxes. There are several different kinds of banks including retail banks, commercial or corporate banks, and investment banks. In most countries, banks are regulated by the national government or central bank.

Commercial banks assume a significant part in the monetary framework and the economy. As a vital part of the monetary framework, banks designate assets from savers to borrowers in an effective way. They offer financial services, which lessen the expense of getting data about the two reserve funds and acquiring potential open doors. These financial services help to make the general economy more effective.

We have created banking management system which provides a general overview of the inner functioning of banks. The database management system created shows some of the main functions of a bank including providing account, loans, and the general hierarchy of banks – bank, branch, employee, customer. It consists of valiant features of taking important and specific details in the form of attributes as represented in the Entity-Relationship model explained in the upcoming pages.

The bank system created works with bank employees providing services to customers by assigning them a bank account. In a bank, one customer can have only one account which is further described by an entity called transaction which maintains multiple records of the customer's deposits and withdrawals. The bank also has multiple employees dealing with a single customer. A single employee can work in only one branch of the bank at a time. A single bank can own multiple branches to increase outreach. Many customers can avail multiple loans from the bank at any given time of the year.

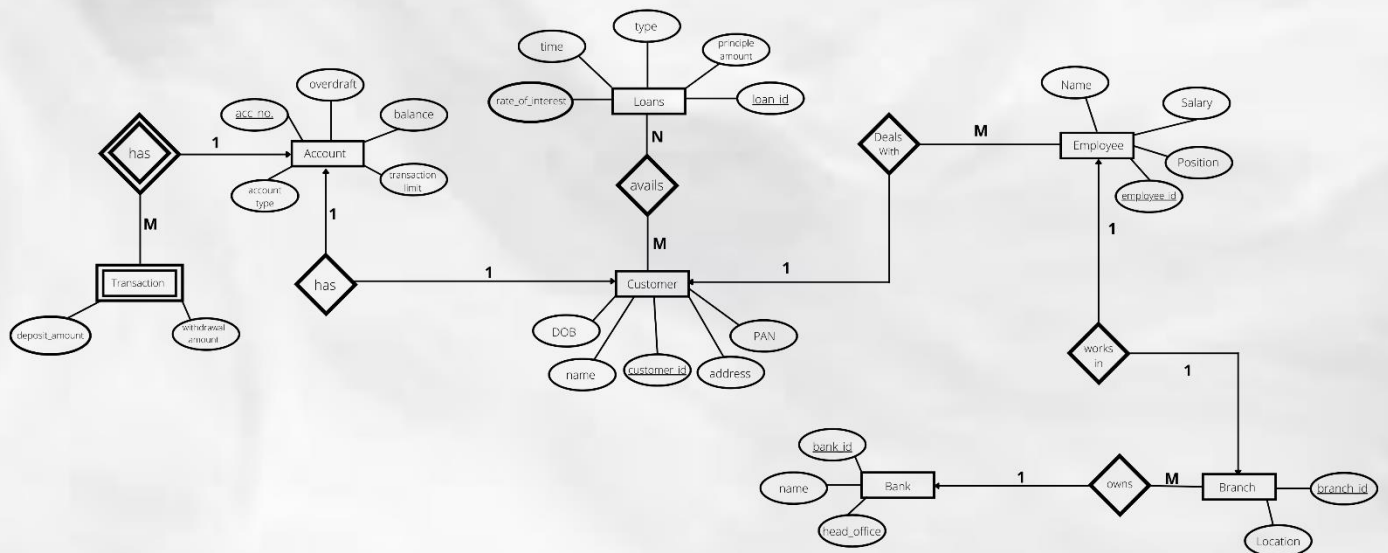
ER Diagram

Banking Management System

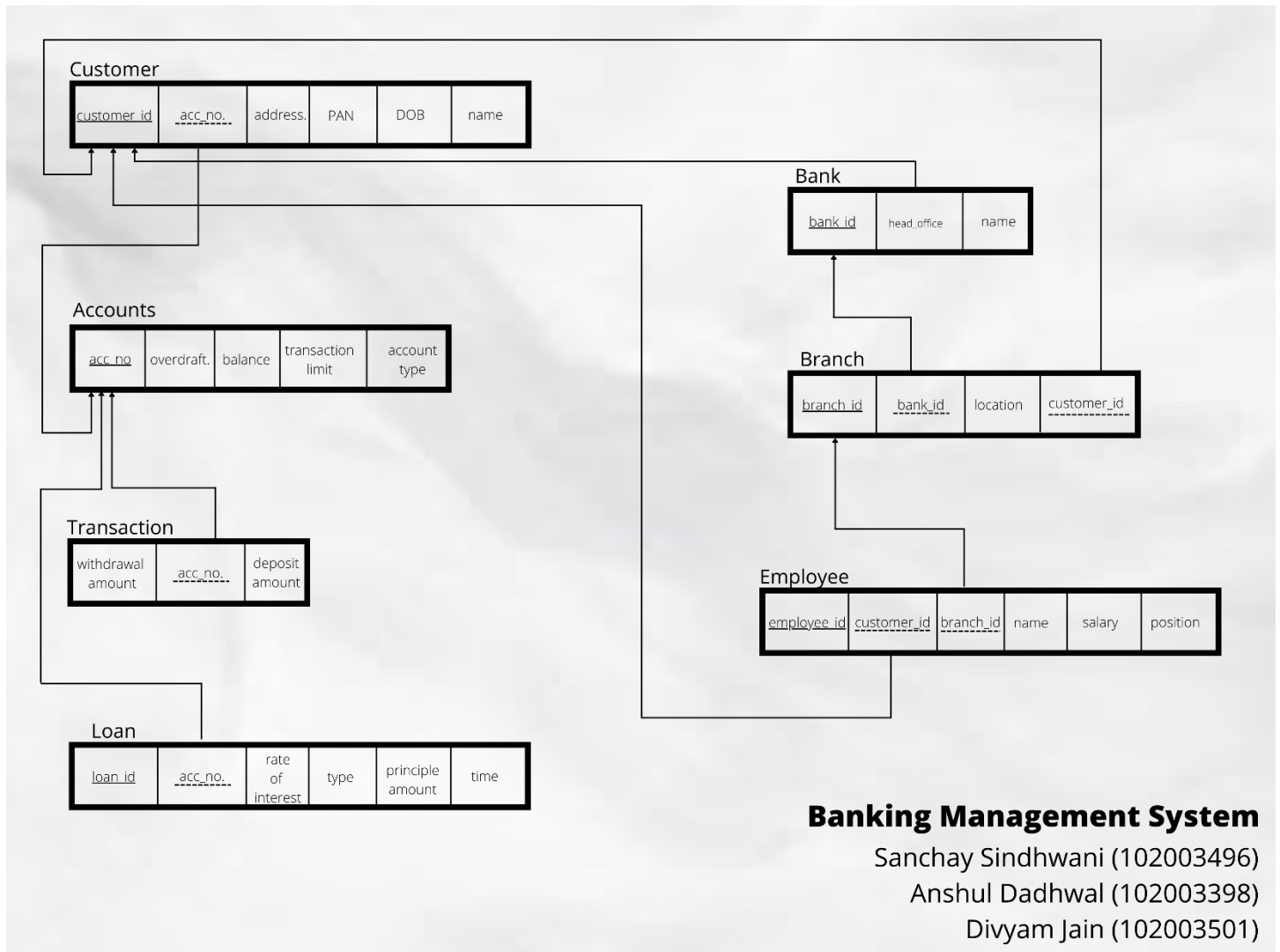
Sanchay Sindhwani (102003496)

Anshul Dadhwal (102003398)

Divyam Jain (102003501)



ER to Table



Normalization (Normalized Table)

Unnormalized Table

acc_no	overdraft	balance	transaction limit	acc_type	withdrawal_amount	deposit_amount
202201	NULL	1350000	10000	saving	NULL	850
					9000	NULL
					950	NULL
					NULL	1670
					NULL	850
202202	NULL	2500000	10000	current	NULL	NULL
202203	NULL	4370000	10000	current	NULL	NULL
202204	NULL	120000	10000	current	NULL	NULL
202205	NULL	10350000	10000	current	6700	NULL
202206	NULL	506000	10000	saving	NULL	200
202207	2300	200000	10000	saving	7500	NULL
					1470	NULL
202208	980	12000	10000	current	NULL	NULL
202209	NULL	1254000	10000	current	NULL	5400

As we can see in this table, there is no possibility of maintaining entity integrity since there are no candidate key that could become primary key. This means that the table is in unnormalized form and needs to be normalized with the help of Rule of Decomposition.

Normalized Table

In this we split the table into two tables where one table named Account contains all the non-repeating attributes and another table named Transactions contains all the repeating attributes with one non-repeating attributes.

Account

acc_no	overdraft	balance	transaction limit	acc_type
202201	NULL	1350000	10000	saving
202202	NULL	2500000	10000	current
202203	NULL	4370000	10000	current
202204	NULL	120000	10000	current
202205	NULL	10350000	10000	current
202206	NULL	506000	10000	saving
202207	2300	200000	10000	saving
202208	980	12000	10000	current
202209	NULL	1254000	10000	current

Transaction

acc_no	withdrawal_amount	deposit_amount
202201	NULL	850
202201	9000	NULL
202201	950	NULL
202201	NULL	1670
202205	6700	NULL
202206	NULL	200
202207	7500	NULL
202207	1470	NULL
202209	NULL	5400

PL/SQL code to implement project functionalities

Declare

Begin

```
Execute Immediate 'create table accounts(  
    acc_no int,  
    overdraft float,  
    balance float,  
    transaction_limit float not null,  
    acc_type varchar(255) not null,  
    constraint pk_acc primary key(acc_no));
```

```
Execute Immediate 'create table customer(  
    cust_id int,  
    acc_no int,  
    address varchar(255),  
    pan varchar(20) unique not null,  
    dob date not null,  
    cust_name varchar(50) not null,  
    constraint pk_cust primary key(cust_id),  
    constraint fk_cust foreign key(acc_no) references accounts(acc_no));
```

```
Execute Immediate 'create table transactions(  
    acc_no int,  
    deposit_amt float,  
    withdrawal_amt float,  
    constraint fk_trans foreign key(acc_no) references accounts(acc_no));
```

```
Execute Immediate 'create table loan(  
    loan_id int,  
    acc_no int,  
    loan_type varchar(255) not null,  
    principle_amt float not null,  
    rate float not null,  
    time_months int not null,  
    constraint pk_loan primary key(loan_id),  
    constraint fk_loan foreign key(acc_no) references accounts(acc_no));
```

```
Execute Immediate 'create table bank(  
    bank_id int,  
    bank_name varchar(255) unique not null,  
    head_office varchar(100) not null,  
    constraint pk_bank primary key(bank_id));
```

```
Execute Immediate 'create table branch(  
    branch_id int,  
    bank_id int,  
    branch_loc varchar(255) not null,  
    cust_id int,  
    constraint pk_branch primary key(branch_id),  
    constraint fk_branch foreign key(bank_id) references bank(bank_id),  
    constraint fk2_branch foreign  
key(cust_id) references customer(cust_id));
```

```

Execute Immediate 'create table employee(
    emp_id int,
    emp_name varchar(50) not null,
    salary float not null,
    pos varchar(100) not null,
    cust_id int,
    branch_id int,
    constraint pk_emp primary key(emp_id),
    constraint fk1_emp foreign key(cust_id) references customer(cust_id),constraint fk2_emp foreign
key(branch_id) references branch(branch_id));
Execute Immediate 'create table customer_avails_loans(
    cust_id int,
    loan_id int,
    constraint fk1_avails foreign key(cust_id) references customer(cust_id),constraint fk2_avails foreign
key(loan_id) references loan(loan_id));
End;

-- drop table accounts;
-- drop table customer;
-- drop table transactions;
-- drop table loan;
-- drop table bank;
-- drop table branch;
-- drop table employee;
-- drop table customer_avails_loans;

Create Or Replace Procedure Addacc(Acc_No In Int,Overdraft In Float,Balance In Float,Trans_Limit In
Float,Acc_Type In Varchar)
Is
Begin
Insert Into Accounts Values(Acc_No,Overdraft,Balance,Trans_Limit,Acc_Type);
End Addacc;
/

Declare
    Acc_No Accounts.Acc_No%Type;
    Overdraft Accounts.Overdraft%Type;
    Balance Accounts.Balance%Type;
    Trans_Limit Accounts.Transaction_Limit%Type;
    Acc_Type Accounts.Acc_Type%Type;
Begin
    Addacc(&acc_No,&overdraft,&balance,&trans_Limit,'&acc_type');
End;

Desc Accounts;
Select * From Accounts;

```



```

Create Or Replace Procedure Addtrans(Acc_No In Int,Deposit In Float,Withdrawal In Float)
Is
Begin
Insert Into Transactions Values(Acc_No,Deposit,Withdrawal);
End Addtrans;
/

```

```

Declare
    Acc_No Transactions.Acc_No%Type;
    Deposit Transactions.Deposit_Amt %Type;
    Withdrawal Transactions.Withdrawal_Amt%Type;
Begin
    Addtrans( &acc_No,&deposit,&withdrawal);
End;

```

```

Desc Transactions;
Select * From Transactions;

```

```

Create Or Replace Procedure Addcust(Cust_Id In Int,Acc_No In Int,Addr In Varchar,Pan In Varchar,Dob In
Varchar,Cust_Name In Varchar)
Is
Begin
Insert Into Customer Values(Cust_Id,Acc_No,Addr,Pan,Dob,Cust_Name);
End Addcust;
/

```

```

Declare
    Cust_Id Customer.Cust_Id%Type;
    Acc_No Customer.Acc_No%Type;
    Addr Customer.Address%Type;
    Pan Customer.Pan%Type;
    Dob Customer.Dob%Type;
    Cust_Name Customer.Cust_Name%Type;
Begin
    Addcust( &cust_Id,&acc_No,'&addr','&pan','&dob','&cust_name');
End;

```

```

Desc Customer;
Select * From Customer;

```

```

Create Or Replace Procedure Addloan(Loan_Id In Int,Acc_No In Int,Loan_Type In Varchar,Principle In
Float,Rate In Float,Tm_Mnths In Int)
Is
Begin
Insert Into Loan Values(Loan_Id,Acc_No,Loan_Type,Principle,Rate,Tm_Mnth);
End Addloan;
/

```

Declare

```
Loan_Id Loan.Loan_Id%Type;  
Acc_No Loan.Acc_No%Type;  
Loan_Type Loan.Loan_Type%Type;  
Principle Loan.Principle_Amt%Type;  
Rate Loan.Rate%Type;  
Tm_Mnth Loan.Time_Months%Type;
```

Begin

```
Addloan(&loan_Id,&acc_No,'&loan_type',&principle,&rate,&tm_Mnth);
```

End;

Desc Loan;

*Select * From Loan;*

Create Or Replace Procedure Addavails(Cust_Id In Int,Loan_Id In Int)

Is

Begin

```
Insert Into Customer_Avails_Loans Values(Cust_Id,Loan_Id);
```

End Addavails;

/

Declare

```
Cust_Id Customer_Avails_Loans.Cust_Id%Type;  
Loan_Id Customer_Avails_Loans.Loan_Id%Type;
```

Begin

```
Addavails(&cust_Id,&loan_Id);
```

End;

Desc Customer_Avails_Loans;

*Select * From Customer_Avails_Loans;*

Create Or Replace Procedure Addbank(Bank_Id In Int,Bank_Name In Varchar,Head_Off In Varchar)

Is

Begin

```
Insert Into Bank Values(Bank_Id,Bank_Name,Head_Off);
```

End Addbank;

/

Declare

```
Bank_Id Bank.Bank_Id%Type;  
Bank_Name Bank.Bank_Name%Type;  
Head_Off Bank.Head_Office%Type;
```

Begin

```
Addbank(&bank_Id,'&BANK_NAME','&head_off');
```

End;

Desc Bank;
*Select * From Bank;*

Create Or Replace Procedure Addbranch(Branch_Id In Int,Bank_Id In Int,Branch_Loc In Varchar,Cust_Id In Int)
Is
Begin
Insert Into Branch Values(Branch_Id,Bank_Id,Branch_Loc,Cust_Id);
End Addbranch;
/

Declare
Branch_Id Branch.Branch_Id%Type;
Bank_Id Branch.Branch_Id%Type;
Branch_Loc Branch.Branch_Loc%Type;
Cust_Id Branch.Cust_Id%Type;
Begin
Addbranch(&branch_Id,&bank_Id,'&branch_loc',&cust_Id);
End;

Desc Branch;
*Select * From Branch;*

Create Or Replace Procedure Addemp(Emp_Id In Int,Emp_Name In Varchar,Sal In Float,Pos In Varchar,Cust_Id In Int,Branch_Id In Int)
Is
Begin
Insert Into Employee Values(Emp_Id,Emp_Name,Sal,Pos,Cust_Id,Branch_Id);
End Addemp;
/

Declare
Emp_Id Employee.Emp_Id%Type;
Emp_Name Employee.Emp_Name%Type;
Salary Employee.Salary%Type;
Pos Employee.Pos%Type;
Cust_Id Employee.Cust_Id%Type;
Branch_Id Employee.Branch_Id%Type;
Begin
Addemp(&emp_Id,'&Emp_Name',&salary,'&Pos',&cust_Id,&branch_Id);
End;

Desc Employee;
*Select * From Employee;*

Output Screenshots

Table Creation

```
1  Declare
2  Begin
3      Execute Immediate 'create table accounts(
4          acc_no int,
5          overdraft float,
6          balance float,
7          transaction_limit float not null,
8          acc_type varchar(255) not null,
9          constraint pk_acc primary key(acc_no));'
10     Execute Immediate 'create table customer(
11         cust_id int,
12         acc_no int,
13         address varchar(255),
14         pan varchar(20) unique not null,
15         dob date not null,
16         cust_name varchar(50) not null,
17         constraint pk_cust primary key(cust_id),
18         constraint fk_cust foreign key(acc_no) references accounts(acc_no));'
19     Execute Immediate 'create table transactions(
20         acc_no int,
21         deposit_amt float,
22         withdrawal_amt float,
23         constraint fk_trans foreign key(acc_no) references accounts(acc_no));'
24     Execute Immediate 'create table loan(
25         loan_id int,
26         acc_no int,
27         loan_type varchar(255) not null,
28         principle_amt float not null,
29         rate float not null,
30         time_months int not null,
31         constraint pk_loan primary key(loan_id),
32         constraint fk_loan foreign key(acc_no) references accounts(acc_no));'
33     Execute Immediate 'create table bank(
34         bank_id int,
35         bank_name varchar(255) unique not null,
36         head_office varchar(100) not null,
37         constraint pk_bank primary key(bank_id));'
38     Execute Immediate 'create table branch(
39         branch_id int,
40         bank_id int,
41         branch_loc varchar(255) not null,
42         cust_id int,
43         constraint pk_branch primary key(branch_id),
44         constraint fk_branch foreign key(bank_id) references bank(bank_id),constraint fk2_branch foreign key(cust_id) references customer(cust_id));'
45     Execute Immediate 'create table employee(
46         emp_id int,
47         emp_name varchar(50) not null,
48         salary float not null,
49         pos varchar(100) not null,
50         cust_id int,
51         branch_id int,
52         constraint pk_emp primary key(emp_id),
53         constraint fk1_emp foreign key(cust_id) references customer(cust_id),constraint fk2_emp foreign key(branch_id) references branch(branch_id));'
54     Execute Immediate 'create table customer_avails_loans(
55         cust_id int,
56         loan_id int,
57         constraint fk1_avails foreign key(cust_id) references customer(cust_id),constraint fk2_avails foreign key(loan_id) references loan(loan_id));'
58 End;
59
60 -- drop table accounts;
61 -- drop table customer;
62 -- drop table transactions;
63 -- drop table loan;
64 -- drop table bank;
65 -- drop table branch;
66 -- drop table employee;
67 -- drop table customer_avails_loans;
```

TABLES

Bank



```
1 Create Or Replace Procedure Addbank(Bank_Id In Int,Bank_Name In Varchar,Head_Off In Varchar)
2 Is
3 Begin
4 Insert Into Bank Values(Bank_Id,Bank_Name,Head_Off);
5 End Addbank;
6 /
7
8 Declare
9     Bank_Id Bank.Bank_Id%Type;
10    Bank_Name Bank.Bank_Name%Type;
11    Head_Off Bank.Head_Office%Type;
12 Begin
13     Addbank(&bank_Id, '&BANK_NAME', '&head_off');
14 End;
15
16 Select * From Bank;
```

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	BANK_ID	NUMBER(38,0)	No	(null)	1 (null)	
2	BANK_NAME	VARCHAR2(255 BYTE)	No	(null)	2 (null)	
3	HEAD_OFFICE	VARCHAR2(100 BYTE)	No	(null)	3 (null)	

❖ BANK_ID	❖ BANK_NAME	❖ HEAD_OFFICE
12	Sate Bank Of India	Mumbai

Branch



```
1 Create Or Replace Procedure Addbranch(Branch_Id In Int,Bank_Id In Int,Branch_Loc In Varchar,Cust_Id In Int)
2 Is
3 Begin
4 Insert Into Branch Values(Branch_Id,Bank_Id,Branch_Loc,Cust_Id);
5 End Addbranch;
6 /
7
8 Declare
9     Branch_Id Branch.Branch_Id%Type;
10    Bank_Id Branch.Bank_Id%Type;
11    Branch_Loc Branch.Branch_Loc%Type;
12    Cust_Id Branch.Cust_Id%Type;
13 Begin
14     Addbranch(&branch_Id,&bank_Id, '&branch_loc', &cust_Id);
15 End;
16
17 Select * From Branch;
```

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	BRANCH_ID	NUMBER(38,0)	No	(null)	1	(null)
2	BANK_ID	NUMBER(38,0)	Yes	(null)	2	(null)
3	BRANCH_LOC	VARCHAR2(255 BYTE)	No	(null)	3	(null)
4	CUST_ID	NUMBER(38,0)	Yes	(null)	4	(null)

❖ BRANCH_ID	❖ BANK_ID	❖ BRANCH_LOC	❖ CUST_ID
111	12	New Delhi	1001
112	12	Gurgaon	1002
113	12	Noida	1003
114	12	Mumbai	1004
115	12	Bengaluru	1005
116	12	Shimla	1006
117	12	Goa	1007
118	12	Manesar	1008
119	12	Ambala	1009

Employee



```

1 Create Or Replace Procedure Addemp(Emp_Id In Int,Emp_Name In Varchar,Sal In Float,Pos In Varchar,Cust_Id In Int,Branch_Id In Int)
2 Is
3 Begin
4 Insert Into Employee Values(Emp_Id,Emp_Name,Sal,Pos,Cust_Id,Branch_Id);
5 End Addemp;
6 /
7
8 Declare
9     Emp_Id Employee.Emp_Id%Type;
10    Emp_Name Employee.Emp_Name%Type;
11    Salary Employee.Salary%Type;
12    Pos Employee.Pos%Type;
13    Cust_Id Employee.Cust_Id%Type;
14    Branch_Id Employee.Branch_Id%Type;
15 Begin
16     Addemp(&emp_Id, '&Emp_Name', &salary, '&Pos', &cust_Id, &branch_Id);
17 End;
18
19 Select * From Employee;

```

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	EMP_ID	NUMBER(38,0)	No	(null)	1	(null)
2	EMP_NAME	VARCHAR2(50 BYTE)	No	(null)	2	(null)
3	SALARY	FLOAT	No	(null)	3	(null)
4	POS	VARCHAR2(100 BYTE)	No	(null)	4	(null)
5	CUST_ID	NUMBER(38,0)	Yes	(null)	5	(null)
6	BRANCH_ID	NUMBER(38,0)	Yes	(null)	6	(null)

EMP_ID	EMP_NAME	SALARY	POS	CUST_ID	BRANCH_ID
1201	Rakesh Kalia	50000	Sales Manager	1001	111
1202	Arpit Taneja	65000	Mortgage Consultant	1002	112
1203	Ansh Chaudhary	10000	Receptionist	(null)	112
1204	Mira Mishra	7000	Housekeeping	(null)	116
1205	Sher Malhotra	100000	Branch Head	1005	117
1206	Dolma Garg	250000	Assistant Vice President	1006	117
1207	Ishika Kapoor	60000	Financial Manager	1007	117
1208	Manvider Singh	89000	Branch Head	1008	118
1209	Aditi Panday	45000	Sales Manager	1009	119

Accounts



```

1 Create Or Replace Procedure Addacc(Acc_No In Int,Overdraft In Float,Balance In Float,Trans_Limit In Float,Acc_Type In Varchar)
2 Is
3 Begin
4 Insert Into Accounts Values(Acc_No,Overdraft,Balance,Trans_Limit,Acc_Type);
5 End Addacc;
6 /
7
8 Declare
9     Acc_No Accounts.Acc_No%Type;
10    Overdraft Accounts.Overdraft%Type;
11    Balance Accounts.Balance%Type;
12    Trans_Limit Accounts.Transaction_Limit%Type;
13    Acc_Type Accounts.Acc_Type%Type;
14 Begin
15     Addacc(&acc_No,&overdraft,&balance,&trans_Limit,'&acc_type');
16 End;
17
18 Select * From Accounts;

```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ACC_NO	NUMBER(38,0)	No	(null)	1	(null)
2	OVERDRAFT	FLOAT	Yes	(null)	2	(null)
3	BALANCE	FLOAT	Yes	(null)	3	(null)
4	TRANSACTION_LIMIT	FLOAT	No	(null)	4	(null)
5	ACC_TYPE	VARCHAR2(255 BYTE)	No	(null)	5	(null)

ACC_NO	OVERDRAFT	BALANCE	TRANSACTION_LIMIT	ACC_TYPE
202201	(null)	1350000	10000	savings
202202	(null)	2500000	10000	current
202203	(null)	4370000	10000	current
202204	(null)	120000	10000	current
202205	(null)	10350000	10000	current
202206	(null)	506000	10000	savings
202207	2300	200000	10000	savings
202208	980	12000	10000	current
202209	(null)	1254000	10000	current

Transaction



```

1 Create Or Replace Procedure Addtrans(Acc_No In Int,Deposit In Float,Withdrawal In Float)
2 Is
3 Begin
4 Insert Into Transactions Values(Acc_No,Deposit,Withdrawal);
5 End Addtrans;
6 /
7
8 Declare
9     Acc_No Transactions.Acc_No%Type;
10    Deposit Transactions.Deposit_Amt %Type;
11    Withdrawal Transactions.Withdrawal_Amt%Type;
12 Begin
13     Addtrans(&acc_No,&deposit,&withdrawal);
14 End;
15
16 Select * From Transactions;

```

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	ACC_NO	NUMBER(38,0)	Yes	(null)	1 (null)	
2	DEPOSIT_AMT	FLOAT	Yes	(null)	2 (null)	
3	WITHDRAWAL_AMT	FLOAT	Yes	(null)	3 (null)	

❖ ACC_NO	❖ DEPOSIT_AMT	❖ WITHDRAWAL_AMT
202201	(null)	850
202201	5000	(null)
202201	950	(null)
202201	(null)	1670
202205	6700	(null)
202206	(null)	200
202207	7500	(null)
202207	1470	(null)
202209	(null)	5400

Customer

```

1 Create Or Replace Procedure Addcust(Cust_Id In Int,Acc_No In Int,Addr In Varchar,Pan In Varchar,Dob In Varchar,Cust_Name In Varchar)
2 Is
3 Begin
4 Insert Into Customer Values(Cust_Id,Acc_No,Addr,Pan,Dob,Cust_Name);
5 End Addcust;
6 /
7
8 Declare
9 Cust_Id Customer.Cust_Id%Type;
10 Acc_No Customer.Acc_No%Type;
11 Addr Customer.Address%Type;
12 Pan Customer.Pan%Type;
13 Dob Customer.Dob%Type;
14 Cust_Name Customer.Cust_Name%Type;
15 Begin
16 Addcust(&cust_Id,&acc_No,'&addr','&pan','&dob','&cust_name');
17 End;
18
19 Select * From Customer;

```

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	CUST_ID	NUMBER(38,0)	No	(null)	1 (null)	
2	ACC_NO	NUMBER(38,0)	Yes	(null)	2 (null)	
3	ADDRESS	VARCHAR2(255 BYTE)	Yes	(null)	3 (null)	
4	PAN	VARCHAR2(20 BYTE)	No	(null)	4 (null)	
5	DOB	DATE	No	(null)	5 (null)	
6	CUST_NAME	VARCHAR2(50 BYTE)	No	(null)	6 (null)	

❖ CUST_ID	❖ ACC_NO	❖ ADDRESS	❖ PAN	❖ DOB	❖ CUST_NAME
1001	202201	New Delhi	JOPL8720L	22-NOV-01	Divyam Harishankar
1002	202202	Gurgaon	UIXN8902K	04-JAN-00	Sanchay Kumar
1003	202203	Patiala	JLKJ9786L	09-DEC-90	Anshul Dadhwal
1004	202204	Agra	DBMS6969K	15-OCT-80	Shreedhar Patel
1005	202205	Gwalior	CONE7895M	26-JUN-87	Aryan Rastogi
1006	202206	New Delhi	VGHI6459N	07-JUN-82	Samreedhi Tiwari
1007	202207	Gurgaon	GGNC8970X	19-DEC-95	Aarushi Rudra
1008	202208	Goa	DAAA8965P	22-NOV-01	Sarisha Rathore
1009	202209	Jaipur	CSBS8765A	06-FEB-99	Akshat Wangrur

Loan

```

1 Create Or Replace Procedure Addloan(Loan_Id In Int,Acc_No In Int,Loan_Type In Varchar,Principle In Float,Rate In Float,Tm_Mnths In Int)
2 Is
3 Begin
4 Insert Into Loan Values(Loan_Id,Acc_No,Loan_Type,Principle,Rate,Tm_Mnth);
5 End Addloan;
6 /
7
8 Declare
9     Loan_Id Loan.Loan_Id%Type;
10    Acc_No Loan.Acc_No%Type;
11    Loan_Type Loan.Loan_Type%Type;
12    Principle Loan.Principle_Amt%Type;
13    Rate Loan.Rate%Type;
14    Tm_Mnths Loan.Time_Months%Type;
15 Begin
16     Addloan(&loan_Id,&acc_No,'&loan_type',&principle,&rate,&tm_Mnth);
17 End;
18
19 Select * From Loan;

```

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	LOAN_ID	NUMBER(38,0)	No	(null)	1	(null)
2	ACC_NO	NUMBER(38,0)	Yes	(null)	2	(null)
3	LOAN_TYPE	VARCHAR2(255 BYTE)	No	(null)	3	(null)
4	PRINCIPLE_AMT	FLOAT	No	(null)	4	(null)
5	RATE	FLOAT	No	(null)	5	(null)
6	TIME_MONTHS	NUMBER(38,0)	No	(null)	6	(null)

❖ LOAN_ID	❖ ACC_NO	❖ LOAN_TYPE	❖ PRINCIPLE_AMT	❖ RATE	❖ TIME_MONTHS
95	202201	Education	20000	1.2	12
32	202201	Travel	30000	12.6	14
56	202203	Government	12000	0.8	6
45	202203	Government	1000	0.8	6
23	202205	Personal	12500	12.6	3
21	202206	Personal	2300	6.8	2
54	202208	Personal	4000	10	3
11	202208	Education	65000	1.5	12
12	202209	Government	7000	0.8	6



```

1  Create Or Replace Procedure Addavails(Cust_Id In Int,Loan_Id In Int)
2  Is
3  Begin
4  Insert Into Customer_Avails_Loans Values(Cust_Id,Loan_Id);
5  End Addavails;
6  /
7
8  Declare
9      Cust_Id Customer_Avails_Loans.Cust_Id%Type;
10     Loan_Id Customer_Avails_Loans.Loan_Id%Type;
11 Begin
12     Addavails(&cust_Id,&loan_Id);
13 End;
14
15 Select * From Customer_Avails_Loans;

```

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	CUST_ID	NUMBER(38,0)	Yes	(null)	1	(null)
2	LOAN_ID	NUMBER(38,0)	Yes	(null)	2	(null)

❖ CUST_ID	❖ LOAN_ID
1001	95
1001	32
1003	56
1003	45
1005	23
1006	21
1008	54
1008	11
1009	12



```
1 CREATE OR REPLACE FUNCTION totalCustomers
2 RETURN number IS
3     total number(20) := 0;
4 BEGIN
5     SELECT count(*) into total
6     FROM customers;
7     RETURN total;
8 END;
```



```
1 DECLARE
2     total_cust number(20);
3 BEGIN
4     total_cust:=totalCustomers();
5     dbms_output.put_line('Total no. of customers registered in the bank: '||total_cust);
6 END;
```



```
1 CREATE OR REPLACE FUNCTION totalLoans
2 RETURN number IS
3     total_loan number(20) := 0;
4 BEGIN
5     SELECT count(*) into total_loan
6     FROM loan;
7     RETURN total_loan;
8 END;
```



```
1 DECLARE
2     total_loan number(20);
3 BEGIN
4     total_loan:=totalLoans();
5     dbms_output.put_line('Total no. of loans registered in the bank: '||total_loan);
6 END;
```

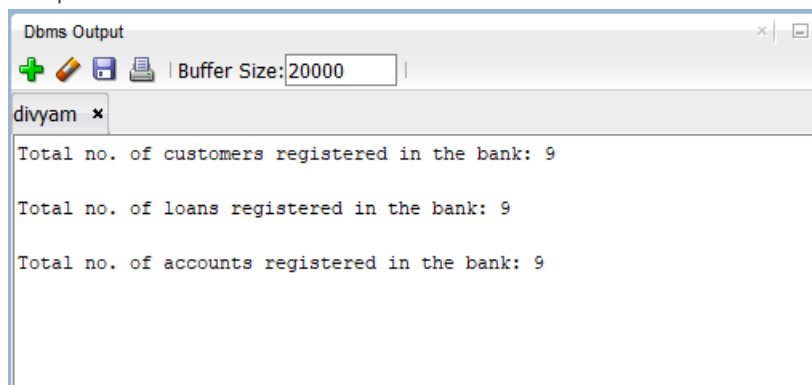


```
1 CREATE OR REPLACE FUNCTION totalAccounts
2 RETURN number IS
3     total_account number(20) := 0;
4 BEGIN
5     SELECT count(*) into total_account
6     FROM accounts;
7     RETURN total_account;
8 END;
```



```
1 DECLARE
2     total_account number(20);
3 BEGIN
4     total_account:=totalAccounts();
5     dbms_output.put_line('Total no. of accounts registered in the bank: '||total_account);
6 END;
```

Output Screen for Function



Conclusion

Database management systems are efficient systems that allow us to access data with ease and increases data integrity. The salient features of database management systems are responsible for consistent and reliable data and increases productivity and decision-making.

As more and more organizations adopt the new standard for storing data, the demand for effective implementation of systems is ever so significant. Our aim with this project was to create a simple yet efficient banking database system that provide a general overview of the inner working of a commercial bank.