

A  
Project Report on  
**ATM Management System**

Developed by:  
**HELI VACHHANI - (IT-045)**  
**DIVYASHREE IYER - (IT-026)**

**Guided By:**  
**Internal Guide:**  
**Prof. Archana N. Vyas**  
**Department of Information Technology**  
**Faculty of Technology**  
**DD University**



**Department of Information Technology,**  
**Faculty of Technology,**  
**Dharmsinh Desai University**

**College Road, Nadiad-387001  
DHARMSINH DESAI UNIVERSITY  
NADIAD-387001, GUJARAT**



**~ CERTIFICATE ~**

This is to certify that the project entitled **“ATM Management System”** is a bonafied report of the work carried out by

1) **Ms. HELI VACHHANI,**

Student ID No : **18ITUON113**

2) **Ms. DIVYASHREE IYER,**

Student ID No : **18ECUOS044**

of Department of Information Technology, semester V, under the guidance and supervision for the subject Database Management System. They were involved in Project training during the academic year 2020-2021.

**Prof. Archana N. Vyas**

(Project Guide)

Department of Information Technology,

Faculty of Technology,

Dharmsinh Desai University, Nadiad

Date:

**Prof. Vipul Dabhi**

Head , Department of Information Technology,

Faculty of Technology,

Dharmsinh Desai University, Nadiad

Date:

## ~ ACKNOWLEDGEMENT ~

Apart from the efforts by us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

We would like to show our greatest appreciation to Prof. Sunil Vithlani. We can't say thank you enough for his tremendous support and help. We feel motivated and encouraged every time we attend his labs. Without his encouragement and guidance this project would not have materialized.

The guidance and support received from all the members who contributed and who are contributing to this project, was vital for the success of the project. We are grateful for their constant support and help.

**Yours Sincerely**

Heli Vachhani,  
Divyashree Iyer

## ~ TABLE OF CONTENTS ~

<b>I. Certificate.....</b>	<b>2</b>
<b>II. Acknowledgement .....</b>	<b>3</b>
 <b>SYSTEM OVERVIEW.....</b>	 <b>5</b>
1.1 Current system.....	5
1.2 Objectives of the Proposed System.....	5
1.3 Advantages of the Proposed System (over current).....	5
 <b>E-R DATA.....</b>	 <b>6</b>
2.1 Entities.....	6
2.2 Relationships.....	6
2.3 ER diagram.....	7
 <b>DATA DICTIONARY.....</b>	 <b>8,9,10</b>
 <b>SCHEMA DIAGRAM.....</b>	 <b>11</b>
 <b>DATABASE IMPLEMENTATION.....</b>	 <b>12</b>
5.1 Create Schema.....	12,13,14,15
5.2 Insert Data values.....	16,17,18,19,20,21
5.3 Queries.....	22
5.4 Exceptions (PL/ SQL).....	28
5.5 Functions.....	30
5.6 Cursors.....	32
5.7 Triggers.....	33
 <b>FUTURE ENHANCEMENTS OF THE SYSTEM.....</b>	 <b>35</b>
 <b>BIBLIOGRAPHY.....</b>	 <b>36</b>

# **1. SYSTEM OVERVIEW**

## **1.1 CURRENT SYSTEM**

We developed the “ATM Management System” to provide a system which handles the information of the various cards and atm which performs transactions. Our system stores information of various atm, cards, accounts, banks, branches, customers, transactions, atm operators, atm technicians, atm faults.

## **1.2 OBJECTIVES OF THE PROPOSED SYSTEM**

- ★ It keeps all logs of transactions performed along with precise information of a transaction.
- ★ It keeps data of atm failures.
- ★ It performs pin change functionality.
- ★ It performs credit, debit facilities with the use of atm.
- ★ All the above objectives can be performed by the card user and account holder.
- ★ It also helps the user to get the details of all transactions performed by a user.
- ★ Also we can view the different information of various accounts and customers, etc.

## **1.3 ADVANTAGES OF THE PROPOSED SYSTEM (OVER CURRENT)**

- ★ First and foremost, it offers faster and very efficient access.
- ★ Good documentation and a systematic way to store the data along with exception handling.
- ★ A secured way to hide private details like pin, cvv and not to reveal it to unauthorized people.
- ★ If the transaction gets failed, data is updated accordingly and loss of money is prevented.
- ★ Eliminates the confusion as it automatically does transactions and stores the information just on a few clicks, processing is made simpler and efficient.
- ★ Customers won't have to wait for longtime.

## 2. E-R DATA

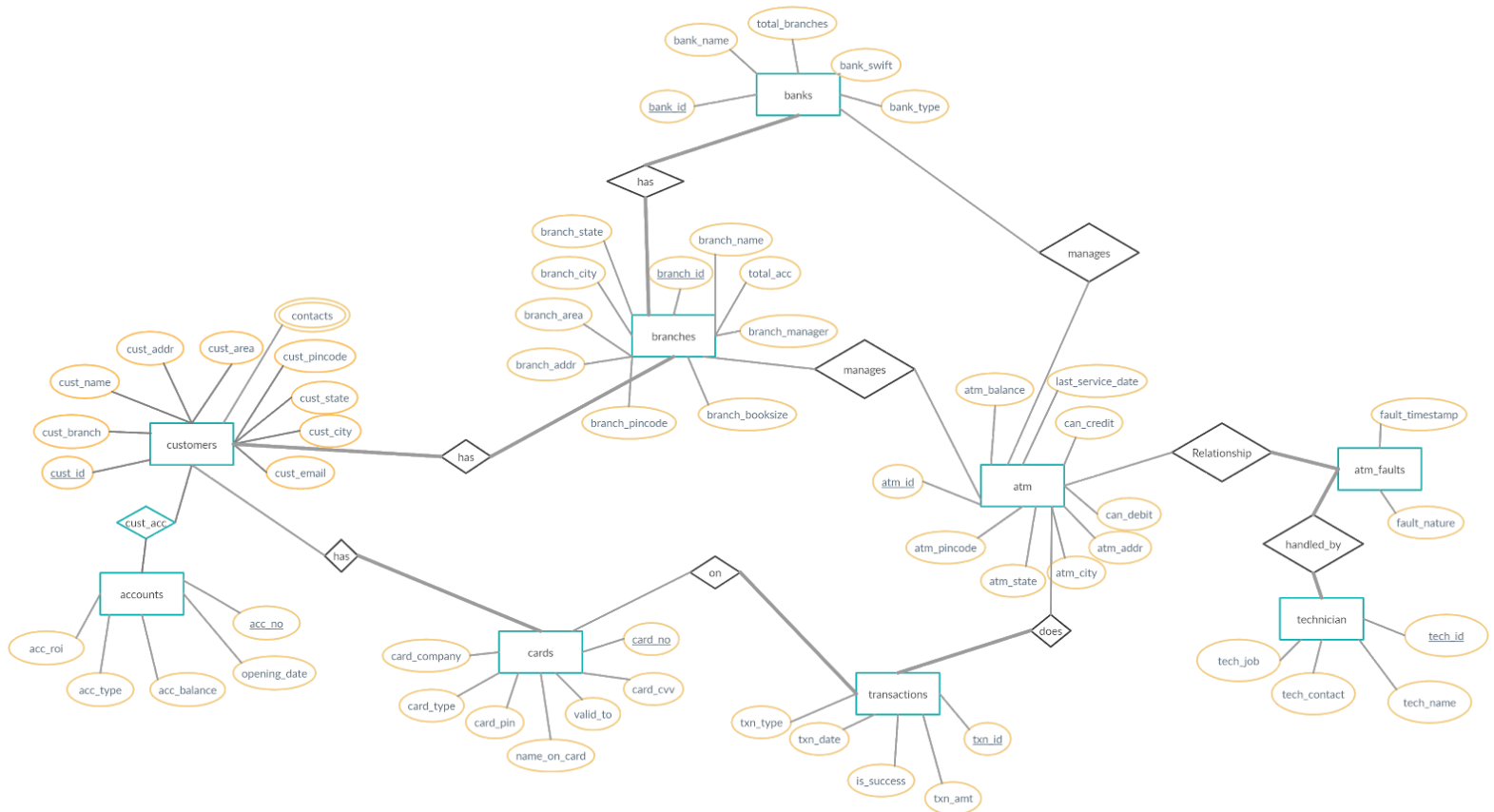
### 2.1 ENTITIES (10)

- ★ Cards
- ★ Transactions
- ★ Atm
- ★ Atm\_fault
- ★ Customers
- ★ Accounts
- ★ Banks
- ★ Branches
- ★ Technician

### 2.2 RELATIONSHIPS

- ★ Customers to Branches have Many to One relationship..
- ★ Accounts to Customers have Many to Many relationships.
- ★ Accounts to Cust\_acc has Many to One relationship.
- ★ Cust\_acc to Customers has One to Many relationships.
- ★ Customers to Contacts has One to Many relationships.
- ★ Banks to Branches has One to Many relationships.
- ★ Cards to Customers has Many to One relationship.
- ★ Banks to Atm has One to Many relationships.
- ★ Atm\_fault to Technicians has Many to One relationship.
- ★ Transactions to Cards has Many to One relationship.
- ★ Atm to Atm\_fault has One to Many relationships.
- ★ Atm to Transaction has One to Many relationships.

## 2.3 ER DIAGRAM



### 3. DATA DICTIONARY

Figure 3.1 Table Cards

```
SQL> desc cards;
```

Name	Null?	Type
CARD_NO	NOT NULL	NUMBER(38)
CARD_CVV	NOT NULL	NUMBER(38)
NAME_ON_CARD	NOT NULL	VARCHAR2(35)
VALID_TO	NOT NULL	DATE
CARD_PIN	NOT NULL	NUMBER(38)
CARD_TYPE	NOT NULL	VARCHAR2(12)
CARD_COMPANY	NOT NULL	VARCHAR2(20)
CARD_CUST	NOT NULL	VARCHAR2(14)

Figure 3.2 Table Transactions

```
SQL> desc transactions;
```

Name	Null?	Type
TXN_ID	NOT NULL	VARCHAR2(20)
TXN_AMT	NOT NULL	NUMBER(38)
TXN_DATE	NOT NULL	DATE
TXN_TYPE	NOT NULL	VARCHAR2(20)
IS_SUCCESS		CHAR(1)
TXN_CARD		NUMBER(38)
TXN_ATM		VARCHAR2(11)

Figure 3.3 Table Atm

Name	Null?	Type
ATM_ID	NOT NULL	VARCHAR2(11)
ATM_BALANCE	NOT NULL	NUMBER(38)
LAST_SERVICE_DATE		DATE
CAN_DEBIT		CHAR(1)
CAN_CREDIT		CHAR(1)
ATM_ADDR		VARCHAR2(50)
ATM_AREA		VARCHAR2(35)
ATM_PINCODE		NUMBER(38)
ATM_BRANCH		VARCHAR2(12)
ATM_BANK		VARCHAR2(5)
CITY		VARCHAR2(20)
STATE		VARCHAR2(20)



Figure 3.4 Table Atm\_Fault

```
SQL> desc atm_fault;
```

Name	Null?	Type
ATM_ID	NOT NULL	VARCHAR2(11)
FAULT_TIMESTAMP	NOT NULL	TIMESTAMP(6)
FAULT_NATURE	NOT NULL	VARCHAR2(50)
ASSIGNED_TO	NOT NULL	VARCHAR2(14)

Figure 3.5 Table Customers

```
SQL> desc customers;
```

Name	Null?	Type
CUST_ID	NOT NULL	VARCHAR2(14)
CUST_NAME	NOT NULL	VARCHAR2(20)
CUST_ADDR		VARCHAR2(35)
CUST_AREA		VARCHAR2(20)
CUST_CITY		VARCHAR2(20)
CUST_STATE		VARCHAR2(9)
CUST_PINCODE		NUMBER(38)
CUST_EMAIL		VARCHAR2(23)
CUST_BRANCH		VARCHAR2(12)

Figure 3.6 Table Cust\_acc

```
SQL> desc cust_acc;
```

Name	Null?	Type
ACC_NO	NOT NULL	NUMBER(38)
CUST_ID	NOT NULL	VARCHAR2(14)

Figure 3.7 Table Contacts

```
SQL> desc contacts;
```

Name	Null?	Type
CUST_ID		VARCHAR2(14)
PHONE_NUM	NOT NULL	VARCHAR2(10)

Figure 3.8 Table Accounts

```
SQL> desc accounts;
```

Name	Null?	Type
ACC_NO	NOT NULL	NUMBER(38)
ACC_ROI		NUMBER(5,2)
ACC_TYPE	NOT NULL	VARCHAR2(10)
OPENING_DATE	NOT NULL	DATE
ACC_BALANCE		NUMBER(12,2)

Figure 3.9 Table Banks

```
SQL> desc banks;
```

Name	Null?	Type
BANK_ID	NOT NULL	VARCHAR2(5)
BANK_NAME	NOT NULL	VARCHAR2(35)
BANK_SWIFT	NOT NULL	VARCHAR2(12)
BANK_TYPE	NOT NULL	VARCHAR2(10)
TOTAL_BRANCHES	NOT NULL	NUMBER(38)

Figure 3.10 Table Branches

```
SQL> desc branches;
```

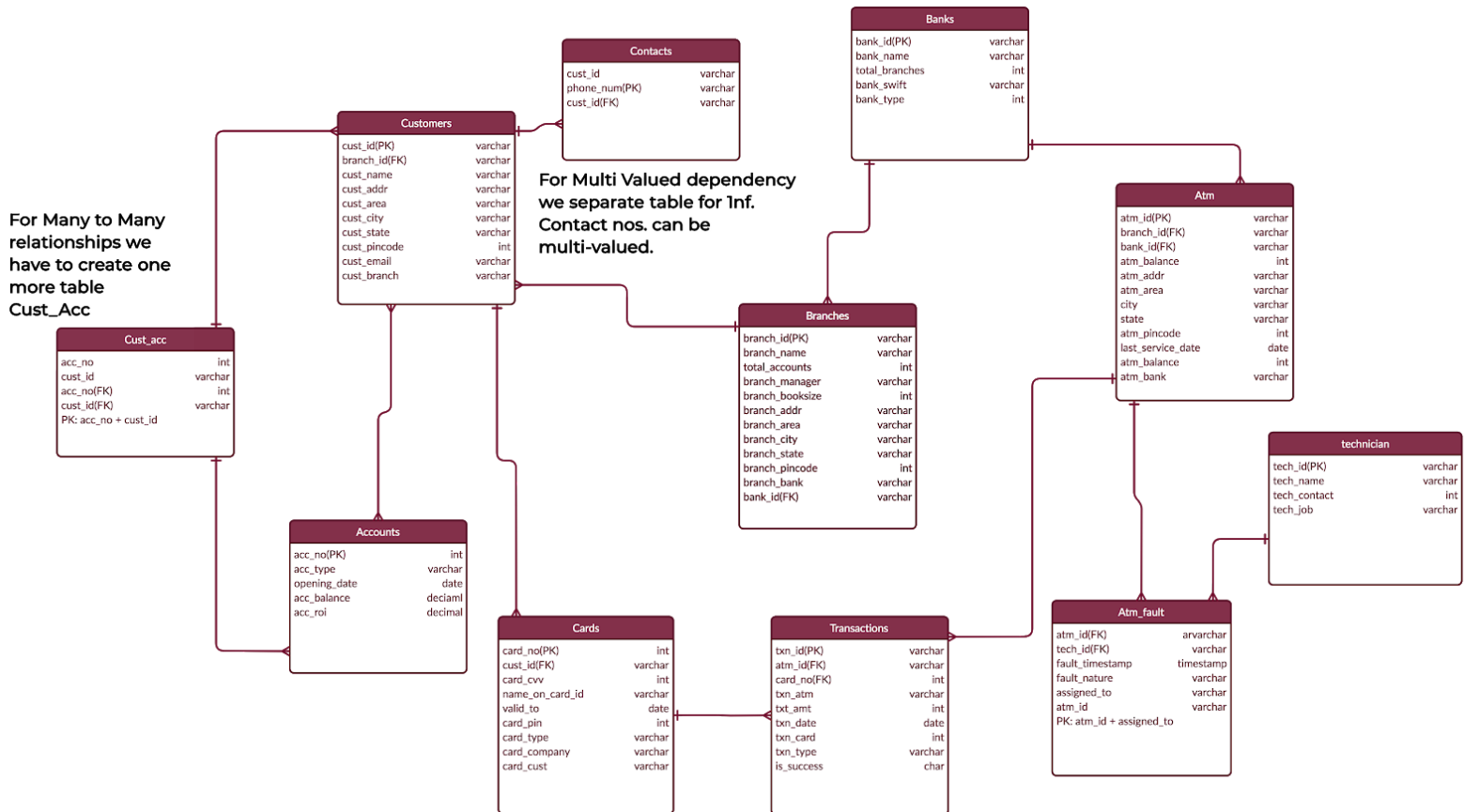
Name	Null?	Type
BRANCH_ID	NOT NULL	VARCHAR2(12)
BRANCH_NAME	NOT NULL	VARCHAR2(12)
TOTAL_ACCOUNTS	NOT NULL	NUMBER(38)
BRANCH_BOOKSIZE	NOT NULL	NUMBER(38)
BRANCH_MANAGER	NOT NULL	VARCHAR2(20)
BRANCH_ADDR		VARCHAR2(22)
BRANCH_AREA		VARCHAR2(12)
BRANCH_CITY		VARCHAR2(12)
BRANCH_STATE		VARCHAR2(12)
BRANCH_PINCODE		NUMBER(38)
BRANCH_BANK		VARCHAR2(5)

Figure 3.11 Table Technician

```
SQL> desc technician;
```

Name	Null?	Type
TECH_ID	NOT NULL	VARCHAR2(14)
TECH_NAME	NOT NULL	VARCHAR2(20)
TECH_CONTACT	NOT NULL	NUMBER(38)
TECH_JOB	NOT NULL	VARCHAR2(35)

## 4. SCHEMA DIAGRAM



## 5. DATABASE IMPLEMENTATION

### 5.1 CREATE SCHEMA

```
create table accounts (  
    acc_no int not null primary key,  
    acc_roi decimal(5,2) default 0.00,  
    acc_type varchar(10) not null,  
    opening_date date not null,  
    acc_balance decimal(12,2) default 0.00  
);
```

```
create table atm(  
    atm_id varchar(11) not null primary key,  
    atm_balance int not null,  
    last_service_date date,  
    can_debit char(1) default 'Y',  
    can_credit char(1) default 'N',  
    atm_addr varchar(50),  
    atm_area varchar(35),  
    atm_city varchar(35);  
    atm_state varchar(35);  
    atm_pincode int,  
    atm_branch varchar(12),  
    atm_bank varchar(5),  
    constraint fk_atm_bank foreign key (atm_bank) references banks(bank_id),  
    constraint fk_atm_branch foreign key (atm_branch) references  
branches(branch_id)  
);
```

```
create table technician(  
    tech_id varchar(14) not null primary key,  
    tech_name varchar(20) not null,  
    tech_contact int not null,  
    tech_job varchar(35) not null  
);
```

```
create table banks(  
    bank_id varchar(5) not null primary key,  
    bank_name varchar(35) not null,  
    bank_swift varchar(12) not null,  
    bank_type varchar(10) not null,  
    total_branches smallint not null  
);
```

```
create table branches(  
    branch_id varchar(8) not null primary key,  
    branch_name varchar(12) not null,  
    total_accounts int not null,  
    branch_booksize int not null,  
    branch_manager varchar(15) not null,  
    branch_addr varchar(20),  
    branch_area varchar(12),  
    branch_city varchar(12),  
    branch_state varchar(7),  
    branch_pincode int,  
    branch_bank varchar(5),  
    constraint fk_bank foreign key (branch_bank) references banks(bank_id)  
);
```

```
create table atm_fault(  
    atm_id varchar(11) not null,  
    fault_timestamp timestamp not null,  
    fault_nature varchar(50) not null,  
    assigned_to varchar(14) not null,  
    constraint pk primary key (atm_id,assigned_to),  
    constraint fk_atm foreign key (atm_id) references atm(atm_id),  
    constraint fk_tech foreign key (assigned_to) references technician(tech_id)  
);
```

```
create table customers(  
    cust_id varchar(14) not null primary key,  
    cust_name varchar(20) not null,  
    cust_addr varchar(35),  
    cust_area varchar(20),  
    cust_city varchar(20),  
    cust_state varchar(35),  
    cust_pincode int,  
    cust_email varchar(23),  
    cust_branch varchar(12),  
    constraint fk_cust_branch foreign key (cust_branch) references  
branches(branch_id)  
);
```

```
create table cust_acc(  
    acc_no int not null,  
    cust_id varchar(14) not null,  
    constraint pk_cust_acc primary key (acc_no,cust_id),  
    constraint fk_cust foreign key (cust_id) references customers (cust_id),  
    constraint fk_acc foreign key (acc_no) references accounts (acc_no)  
);
```

```
create table contacts(  
    cust_id varchar(14),  
    phone_num varchar(10) not null primary key,  
    constraint fk_cont_cust foreign key (cust_id) references customers (cust_id)  
);
```

```
create table cards(  
    card_no int not null primary key,  
    card_cvv smallint not null,  
    name_on_card varchar(35) not null,  
    valid_to date not null,  
    card_pin smallint not null,  
    card_type varchar(12) not null,  
    card_company varchar(20) not null,
```

```
card_cust varchar(14) not null,  
constraint fk_card_cust foreign key (card_cust) references  
customers(cust_id)  
);
```

```
create table transactions(  
    txn_id varchar(20) not null primary key,  
    txn_amt int not null,  
    txn_date date not null,  
    txn_type varchar(20) not null,  
    is_success char(1) default 'Y',  
    txn_card int,  
    txn_atm varchar(11),  
    constraint fk_card foreign key (txn_card) references cards(card_no),  
    constraint fk_txn_atm foreign key (txn_atm) references atm(atm_id)  
);
```

## 5.2 INSERTed DATA VALUES

### ACCOUNTS TABLE

```
SQL> select * from accounts;
```

ACC_NO	ACC_ROI	ACC_TYPE	OPENING_D	ACC_BALANCE
1.2323E+11	4.2	Savings	23-NOV-12	1200000
2.1323E+11	4.2	Savings	09-NOV-12	1100000
2.4523E+11	0	Current	09-OCT-11	6500000
2.4723E+11	0	Current	09-OCT-19	6100000
2.4723E+11	4.2	Savings	19-DEC-19	450000
2.4723E+11	4.5	Savings	10-DEC-10	230000

```
6 rows selected.
```

### BANKS TABLE

```
SQL> select * from banks;
```

BANK_	BANK_NAME	BANK_SWIFT	BANK_TYPE	TOTAL_BRANCHES
AXIS	Axis Bank	AXISINBBXXX	Private	5
CITI	CITI Bank	CITIINBXXX	Private	5
HDFC	HDFC Bank	HDFCINBXXX	Private	5
ICICI	ICICI Bank	ICICIINBXXX	Private	5
PUNB	Punjab National Bank	PUNBINBXXX	Government	5
SBI	State Bank of India	SBIINBXXX	Government	5
YESB	YES Bank	YESBINBXXX	Private	5

```
7 rows selected.
```



**CUSTOMERS TABLE**

```
SQL> select * from customers;
```

CUST_ID	CUST_NAME	CUST_ADDR	CUST_AREA	CUST_CITY
abcd0001000001	Arjun Rampal	118,Shanta Society	Sabarmati	Ahmedabad
abcd0002000001	Shatrugna Sinha	10, Nanik niwas	Sabarmati	Ahmedabad
abcd0003000001	Tara Sutaria	13, Riddhi Siddhi	Kalupur	Ahmedabad
abcd0004000001	Arya Dixit	43, Janardan	Satellite	Ahmedabad
abcd0005000001	Kiara Adwani	43, Ashok Nagar	Prernatirth	Ahmedabad
ocde0001000001	Naira Shekh	A/2, Dattatray	Lal Darwaja	Ahmedabad
ocde0001000002	Amaan Shekh	A/2, Dattatray	Lal Darwaja	Ahmedabad

```
7 rows selected.
```

**CUSTOMER TABLE(conti. )**

CUST_STAT	CUST_PINCODE	CUST_EMAIL	CUST_BRANCH
Gujarat	380001	arjunrampal@gmail.com	AXIS0001
Gujarat	380001	shatrugna@gmail.com	AXIS0002
Gujarat	380079	stars@gmail.com	AXIS0003
Gujarat	380079	arya@gmail.com	AXIS0004
Gujarat	380015	kiara@gmail.com	AXIS0005
Gujarat	380015	naira@gmail.com	CITI0001
Gujarat	380015	naira@gmail.com	CITI0001

**TRANSACTIONS TABLE**

```
SQL> select * from transactions;
```

TXN_ID	TXN_AMT	TXN_DATE	TXN_TYPE	I	TXN_CARD	TXN_ATM
poi0987	500	16-OCT-20	Cash Withdrawal	y	1.2346E+15	abc123765
poi0988	500	16-OCT-20	Cash Withdrawal	y	1.2346E+15	abc123765
oiy876	500	16-OCT-20	Cash Deposit	y	1.2346E+15	abc123098
oiy877	1500	16-OCT-20	Cash Deposit	y	1.2346E+15	abc123098
oiy878	0	16-OCT-20	No Change	y	4.3346E+15	abc123098

**CUSTOMER\_ACCOUNT TABLE**

```
SQL> select * from cust_acc;
```

ACC_NO	CUST_ID
1.2323E+11	abcd0001000001
2.1323E+11	abcd0002000001
2.4523E+11	abcd0003000001
2.4723E+11	abcd0004000001
2.4723E+11	abcd0005000001
2.4723E+11	bcde0001000001
2.4723E+11	bcde0001000002

**BRANCHES TABLE**

```
SQL> select * from branches;
```

BRANCH_ID	BRANCH_NAME	TOTAL_ACCOUNTS	BRANCH_BOOKSIZE	BRANCH_MANAGER	BRANCH_ADDR
AXIS0001	Sabarmati	1	1200000	Amit Shah	Near Sabarmati River
AXIS0002	Ashram	1	1100000	Amit Patel	Near Sabarmati Ashram
AXIS0003	Kalupur	1	6500000	Rajdeep Sardesai	Near Kalupur Station
AXIS0004	Satellite	1	6100000	Raj desai	Near S.G. Road
AXIS0005	Prernatirth	1	450000	Raj Choksi	Near Shiv Mandir
CITI0001	Sabarmati	1	230000	Pankil Shah	Opp. Sabarmati Ashram
CITI0002	Adrakuri	1	2054000	Parth Patel	Opp. Asha School
CITI0003	Alkusa	1	8239000	Raju Bawa	Opp. Applewoods app.
CITI0004	Balidih	1	23239000	Vishwas Bawa	Near Sun Studio
CITI0005	Chama	1	2334209	Visnu Shiva	Near Sun Photo
HDFC0001	Hindgir	1	348700	Sushant Singh	Opp. Medkart
HDFC0002	Khalari	1	8948700	Shashank Singh	Opp. Medical hospital
HDFC0003	Akri	1	4448700	Shashank Rajput	Opp. Prerna hospital
HDFC0004	Balab	1	9348700	Arjun Pujo	Opp. Prerna school
HDFC0005	Alakode	1	34348700	Arjun Cheru	Opp. Raju Drinks

```
15 rows selected.
```

**BRANCHES TABLE(Conti.)**

BRANCH_AREA	BRANCH_CITY	BRANCH_STATE	BRANCH_PINCODE	BRANC
Lal Darwaja	Ahmedabad	Gujarat	380001	AXIS
Lal Darwaja	Ahmedabad	Gujarat	380001	AXIS
Kalupur	Ahmedabad	Gujarat	380079	AXIS
Satellite	Ahmedabad	Gujarat	380015	AXIS
Prernatirth	Ahmedabad	Gujarat	380015	AXIS
Lal Darwaja	Ahmedabad	Gujarat	380001	CITI
Adrakuri	Bokaro	Jharkhand	827013	CITI
Alkusa	Bokaro	Jharkhand	827013	CITI
Balidih	Bokaro	Jharkhand	827014	CITI
Chama	Ranchi	Jharkhand	829205	CITI
Hindgir	Ranchi	Jharkhand	829209	HDFC
khalari	Ranchi	Jharkhand	829205	HDFC
Akri	Ranchi	Jharkhand	835225	HDFC
Akri	Ranchi	Jharkhand	835225	HDFC
Alakode	Kannur	Kerala	670571	HDFC



**TECHNICIAN TABLE**

```
SQL> select * from technician;
```

TECH_ID	TECH_NAME	TECH_CONTACT	TECH_JOB
qwer1234	Arun sobti	9876543210	work verifier
qwer1254	Arun kashyap	9876543299	display Maintainer
qwer1354	Aruna kashyap	9876543289	payment slip entry
qwer1384	puja kashyap	9876543280	electricity maintainer

**CARDS TABLE**

```
SQL> select * from cards;
```

CARD_NO	CARD_CVV	NAME_ON_CARD
1.2346E+15	123	Arjun Rampal
1.2346E+15	883	Shatrugna Sinha
1.2346E+15	173	Tara Sutaria
1.2346E+15	567	Arya Dixit
4.3346E+15	517	Kiara Adwani
4.3346E+15	117	Naira Shekh
4.3346E+15	199	Amaan Shekh

```
7 rows selected.
```

VALID_TO	CARD_PIN	CARD_TYPE	CARD_COMPANY	CARD_CUST
01-NOV-22	987	Debit	mastercard	abcd0001000001
01-JAN-22	3456	Debit	mastercard	abcd0002000001
01-NOV-24	9867	Debit	VISA	abcd0003000001
01-NOV-23	9800	Debit	VISA	abcd0004000001
01-SEP-23	9899	Credit	VISA	abcd0005000001
01-JAN-23	348	Debit	VISA	bcde0001000001
01-JAN-23	1448	Debit	VISA	bcde0001000002

**ATM TABLE**

ATM_ID	ATM_BALANCE	LAST_SERV	C	C	ATM_ADDR	ATM_AREA	ATM_PINCODE	ATM_BRANCH	ATM_B
abc123765	3445000	27-SEP-20	y	n	near. hindustan petrol pump	Lal darwaja	380001	AXIS0001	
abc123098	2345000	30-SEP-20	y	y	near. sharda school	Lal darwaja	380001	AXIS0001	
abc123076	9345000	30-SEP-20	y	y	opp.haran crossing	Lal darwaja	380001	AXIS0002	
abc123086	1345000	14-OCT-20	y	y	opp.haran crossing	Lal darwaja	380001	CITI0001	
abc123089	1945000	14-SEP-20	y	n	opp. satyam mandir	Satellite	380015		CITI
abc123459	1945000	14-SEP-20	y	n	opp. satyam mandir	Satellite	380015		AXIS

**CITY****STATE**

Ahmedabad	Gujarat
Ahmedabad	Gujarat
Ahmedabad	Gujarat
Ahmedabad	Gujarat
Ahmedabad	Gujarat
Ahmedabad	Gujarat

**ATM\_FAULT TABLE**

ATM_ID	FAULT_TIMESTAMP
abc123765	23-APR-20 08.18.07.000000 PM
abc123098	20-OCT-20 08.18.07.000000 PM
abc123098	20-SEP-20 12.18.07.000000 PM
abc123459	23-SEP-20 12.18.07.000000 PM

**FAULT\_NATURE****ASSIGNED\_TO**

display not working	qwer1254
display not working	qwer1254
no electric connection	qwer1384
no electric connection	qwer1384

## 5.3 QUERIES

### 1. Find out the address of those ATMs which can be credited.

Select ATM\_ADDR, ATM\_AREA, ATM\_PINCODE from atm  
where CAN\_CREDIT ='y';

ATM_ADDR	ATM_AREA	ATM_PINCODE
near. sharda school	Lal darwaja	380001
opp.haran crossing	Lal darwaja	380001
opp.haran crossing	Lal darwaja	380001

### 2. Give the Card no. and CVV no. of those cards by Visa.

Select CARD\_NO, CARD\_CVV from cards where  
CARD\_COMPANY='VISA';

CARD_NO	CARD_CVV
1.2346E+15	173
1.2346E+15	567
4.3346E+15	517
4.3346E+15	117
4.3346E+15	199

### 3. Give the name of the branches and maximum no. of accounts present out of group of same branch names.

Select BRANCH\_NAME, max(total\_accounts) from branches  
group by BRANCH\_NAME;

BRANCH_NAME	MAX(TOTAL_ACCOUNTS)
-----	-----
Sabarmati	1
Balab	1
Balidih	1
Chama	1
Akri	1
Kalupur	1
Prernatirth	1
Alkusa	1
Khalari	1
Alakode	1
Ashram	1
Adrakuri	1
Hindgir	1
Satellite	1

### 4. Give information of all accounts of type Current.

Select ACC\_no || ' ' from Accounts where ACC\_TYPE like 'C\_%';

ACC_NO   ' '
-----
245234345459
247234345458

### 5. Give details of all the ATMs arranged in decreasing order by the current balance.

Select \* from atm order by ATM\_balance desc;

ATM_ID	ATM_BALANCE	LAST_SERV	C	C	ATM_ADDR	ATM_AREA	ATM_PINCODE	ATM_BRANCH	ATM_B
abc123076	9345000	30-SEP-20	y	y	opp.haran crossing	Lal darwaja	380001	AXIS0002	
abc123765	3445000	27-SEP-20	y	n	near. hindustan petrol pump	Lal darwaja	380001	AXIS0001	
abc123098	2345000	30-SEP-20	y	y	near. sharda school	Lal darwaja	380001	AXIS0001	
abc123459	1945000	14-SEP-20	y	n	opp. satyam mandir	Satellite	380015		AXIS
abc123089	1945000	14-SEP-20	y	n	opp. satyam mandir	Satellite	380015		CITI
abc123086	1345000	14-OCT-20	y	y	opp.haran crossing	Lal darwaja	380001	CITI0001	

### 6. List all the transactions performed by a given customer.

```
SELECT TXN_TYPE, TXN_DATE FROM transactions t WHERE
EXISTS
(SELECT * FROM cards c, atm a
WHERE t.TXN_CARD = c.CARD_NO and
t.TXN_Atm = a.ATM_ID and
c.CARD_Cust = 'abcd0001000001');
```

TXN_TYPE	TXN_DATE
Cash Withdrawal	16-OCT-20



**7. List all the banks which have an ATM at a given location.**

```
SELECT BANK_NAME FROM banks WHERE BANK_ID IN
(SELECT
ATM_BANK as BANK_ID FROM atm WHERE
pincode=380015;
```

```
BANK_NAME
-----
Axis Bank
CITI Bank
```

**8. List customer name and contact of Axis bank account holder.**

```
select cust_name, phone_num from contacts full join customers on
customers.cust_id=contacts.cust_id where cust_branch like 'A_%';
```

```
CUST_NAME          PHONE_NUM
-----
Arjun Rampal       9876543210
Shatrugna Sinha    9876543287
Tara Sutaria        9876873287
Tara Sutaria        9876873221
Arya Dixit          8876873221
Kiara Adwani
```

**9. Give names of all those customers whose transactions have been successful.**

Select CUST\_name from customers, transactions, cards where  
txn\_card=card\_no and IS\_SUCCESS='Y';

```
CUST_NAME
-----
Arjun Rampal
Shatrugna Sinha
Tara Sutaria
Arya Dixit
Kiara Adwani
Naira Shekh
Amaan Shekh
Arjun Rampal
Shatrugna Sinha
Tara Sutaria
Arya Dixit
Kiara Adwani
Naira Shekh
Amaan Shekh

14 rows selected.
```

**10. List all the transactions done by all cards.**

```
select txn_date, txn_card from transactions left join cards on  
cards.card_no=transactions.txn_card;
```

```
TXN_DATE      TXN_CARD  
-----  
16-OCT-20 1.2346E+15  
16-OCT-20 1.2346E+15  
16-OCT-20 1.2346E+15  
16-OCT-20 1.2346E+15  
16-OCT-20 4.3346E+15  
29-OCT-20 4.3346E+15  
29-OCT-20 4.3346E+15  
  
7 rows selected.
```

## 5.4 EXCEPTIONS (PL/SQL)

**1. Write a PL/SQL block that lets you withdraw amount from your account and raises an exception if there is low balance or if entered account number is wrong. In case if transaction takes place without exception calling, insert data into transaction table.**

```

DECLARE cardno cards.CARD_NO%TYPE:=6222917841407202;
wdamt accounts.acc_balance%TYPE:=2500;
prevbal accounts.acc_balance%TYPE;
rembal accounts.acc_balance%TYPE;
aid accounts.ACC_NO%TYPE;
Low_bal EXCEPTION;
BEGIN
SELECT accounts.ACC_balance INTO prevbal FROM accounts join cust_acc on
cust_acc.acc_no=accounts.acc_no join customers on
customers.cust_id=cust_acc.cust_id join cards on
cards.card_cust=customers.cust_id
where cards.card_no=cardno;
if prevbal<wdamt then
Raise Low_bal;
else
SELECT accounts.ACC_NO INTO aid FROM accounts join cust_acc on
cust_acc.acc_no=accounts.acc_no join customers on
customers.cust_id=cust_acc.cust_id join cards on
cards.card_cust=customers.cust_id
where cards.card_no=cardno;
rembal:=prevbal-wdamt;
UPDATE accounts SET accounts.acc_balance=rembal WHERE
accounts.ACC_NO=aid;
DBMS_OUTPUT.PUT_LINE ('Amount '|| wdamt || ' Withdrawn from account
' || aid);
select accounts.acc_balance into rembal from accounts where
accounts.acc_no=aid;
DBMS_OUTPUT.PUT_LINE ('New account balance is ' || rembal );

```

```

INSERT INTO transactions(TXN_ID, TXN_Atm, TXN_CARD,
TXN_AMT, TXN_DATE, TXN_TYPE, IS_SUCCESS)
VALUES((CONCAT('sdf' , '0045')), 'abc123089', cardno, wdamt,
TO_CHAR(Sysdate, 'DD-MON-YYYY'), 'Cash Withdraw', 'Y');
END if;
exception
when Low_Bal Then
DBMS_OUTPUT.PUT_LINE ('You dont have enough balance!!');
when no_data_found then
DBMS_OUTPUT.PUT_LINE ('Entered account number is wrong please
Retry');
END;

```

Result #1 (1×1)

pString

Entered card number is wrong please Retry

Result #1 (1×1)

pString

You dont have enough balance!!

Result #1 (1×1)

Result #2 (1×1)

pString

Amount 250.00 Withdrwan from account 87190000325204

pString

New account balance is 205038.77

## 5.5 FUNCTIONS

**1. Write a function that deposits amount to the user's account and display appropriate message and inserts data into transaction Table.**

```
CREATE OR REPLACE FUNCTION deposit(accno int, amt decimal, card_no int)
RETURN decimal
IS
    upbal decimal(12,2);
BEGIN
    UPDATE accounts SET accounts.acc_balance=accounts.acc_balance+amt
    WHERE
        accounts.ACC_NO=accno;
    INSERT INTO transactions(TXN_ID, TXN_atm, TXN_CARD,
        TXN_AMT, TXN_DATE, TXN_TYPE, IS_SUCCESS)
    VALUES(CONCAT('wfd' , '0032'),'abc123089', card_no, amt,
        TO_CHAR(Sysdate, 'DD-MON-YYYY'), 'Cash Deposit', 'Y');
    SELECT accounts.ACC_balance INTO upbal FROM accounts WHERE
        accounts.ACC_NO=accno;
    RETURN upbal;
END;
DECLARE
    acc_no INT:=247234345499;
    dep_amt decimal(12,2):=1500;
    card_no int:=4334567809871659;
    newbal decimal(12,2);
BEGIN
    newbal:=deposit(acc_no, dep_amt, card_no);
    dbms_output.put_line('Updated Balance is ' || newbal);
END;
/
```



A screenshot of a Java IDE showing a variable named `pString` with the value `Updated Balance is 160116`. The text is displayed in a green font on a light background.

## 2. Write a function that returns the remaining or available balance in an ATM and display appropriate messages.

```

CREATE or replace FUNCTION atm_balance(atm_id varchar)
RETURN decimal IS
atm_balance decimal(12,2);
BEGIN
SELECT atm.ATM_balance INTO atm_balance FROM atm WHERE
atm.ATM_ID=atm_id;
RETURN atm_balance;
END;
/
DECLARE
aid atm.ATM_ID%TYPE:="GAO585714TZ";
newbal decimal(12,2);
BEGIN
newbal:=atm_balance(aid);
dbms_output.put_line('Available Balance in ATM ' || aid || ' is ' || newbal);
END;
/

```



The screenshot shows a window titled 'pString' with a green text output: 'Available Balance in ATM GAO585714TZ is 1278900'.

## 5.6 CURSORS

**1. Write a PL/SQL block that warns the customer to keep a minimum of rupees 10000 balance if they have a Savings account else display the account balance.**

```

DECLARE CURSOR min_bal IS SELECT acc.ACC_BAL FROM acc
WHERE acc.ACC_TYPE = 'Savings';
sa acc.ACC_BAL%TYPE;
BEGIN OPEN min_bal;
  LOOP FETCH min_bal INTO sa;
  SELECT sa;
  if (sa < 10000) then DBMS_OUTPUT.PUT_LINE('You need to keep a
  minimum balance of 10000!!');
  else DBMS_OUTPUT.PUT_LINE('Your account balance is:' || sa); end if;
  exit when min_bal%notfound; END LOOP;
CLOSE min_bal;
END;
/

```

sa	pString
329,156.12	Your account balance is:329156.12
sa	pString
0.00	You need to keep a minimum balance of 10000!!
sa	pString
541.00	You need to keep a minimum balance of 10000!!
sa	pString
160,115.62	Your account balance is:160115.62
sa	pString
983,038.50	Your account balance is:983038.50
sa	pString
983,038.50	Your account balance is:983038.50



## 5.7 TRIGGERS

**1. All ATMs are under maintainance from 2AM to 3AM, Write a trigger to make all transaction unprocessed and mark them as unseccessful.**

```
CREATE OR REPLACE TRIGGER TXN_TIME
BEFORE INSERT ON txn
FOR EACH ROW
DECLARE
  ttime txn.TXN_TIME%TYPE;
  tt1 txn.TXN_TIME%TYPE;
  tt2 txn.TXN_TIME%TYPE;
BEGIN
  tt1:= '02:00:00';
  tt2:= '03:00:00';
  IF NEW.TXN_TIME BETWEEN tt1 AND tt2 THEN
    SET NEW.TXN_IS_SUCCESS='N';
  END IF;
END;
INSERT INTO txn(TXN_ID, TXN_AID, TXN_CARD, TXN_AMT,
TXN_DATE, TXN_TIME, TXN_TYPE, TXN_IS_SUCCESS)
VALUES('TD163467293885808974', 'MQZ533565KY',
'6224216213309791', 500, '2018-01-16', '02:07:56', 'TRIGGER CALLED',
'Y');
```

[Data Entered in txn table is:](#)

TD163467293885808974	MQZ533565KY	6,224,216,213,309,791	500	2018-01-16	02:07:56	TRIGGER CALLED	N
----------------------	-------------	-----------------------	-----	------------	----------	----------------	---

**2. If any of the atm is having less balance than 1000, then disable debit facility from that atm and mark all transaction unsuccessful, write the data to atm fault table mentioning atm is running out of cash and assign a technician for solving that.**

```
CREATE OR REPLACE TRIGGER atmlimit
BEFORE INSERT ON txn
FOR EACH ROW
DECLARE
aid txn.TXN_AID%TYPE;
abal atm.ATM_BAL%TYPE;
BEGIN SELECT atm.ATM_BAL INTO abal FROM atm WHERE
atm.ATM_ID=NEW.TXN_AID;
IF abal <1000 then SET NEW.TXN_IS_SUCCESS='N';
INSERT INTO atm_system.atm_fault (atm_id, date_of_fault, time_of_fault,
nature_of_fault, assigned_t)
VALUES (NEW.TXN_AID, (select SYSDATE()), (select CURTIME()), 'Out of
cash', 'IWSP5646562751');
end if;
END;
INSERT INTO txn(TXN_ID, TXN_AID, TXN_CARD, TXN_AMT,
TXN_DATE, TXN_TIME, TXN_TYPE, TXN_IS_SUCCESS) VALUES((select
CONCAT('TRIG', LOCALTIME()+1)), 'QFO978562ZG', '2310376183112023',
500, (SELECT SYSDATE()), (SELECT CURTIME()), 'Out of Cash: Trigger
Called', 'Y');
```

[Data Entered in txn table is:](#)

TXN_ID	TXN_AID	TXN_CARD	TXN_AMT	TXN_DATE	TXN_TIME	TXN_TYPE	TXN_IS_SUCCESS
TRIG20191013230926	QFO978562ZG	2,310,376,183,112,023	500	2019-10-13	23:09:25	Out of Cash: Trigger	N

[Data Entered in Atm fault table is:](#)

QFO978562ZG	2019-10-13	23:09:25	Out of cash	IWSP5646562751
-------------	------------	----------	-------------	----------------

## **6. FUTURE ENHANCEMENTS OF THE SYSTEM**

This project has been done very systematically and all the efforts have been taken so that the project would be efficient and a success. Though, there still exists a scope of improvement for our project. The system has been developed in such a way that it can accept modifications and further changes.

The following could be the enhancements that can be incorporated in our project:

- ★ The relevant information can only be accessed by the authenticate users (multiple) who are assigned for it.
- ★ Adding more features that are currently used by the various ATMs such as utility bill payments and pin change among other things.

## 7. BIBLIOGRAPHY

After brainstorming along with our mentor, our current project was a success, various sources have been referred. We have referenced a few books and websites as stated below. We have used **ORACLE SQLplus (upto 5.3)** then onwards due to the limitations of SQLplus we had to shift to **MARIADB and HeidiSQL (5.4 onwards)**.

Website References and Websites Used:

1. [www.creately.com](http://www.creately.com)
2. [www.stackoverflow.com](http://www.stackoverflow.com)
3. [www.mariadb.org](http://www.mariadb.org)
4. [www.w3school.com](http://www.w3school.com)
5. [www.oracle.com](http://www.oracle.com)

Reference Book:

1. Database System Concepts Henry F. Korth & A. Silberschatz. 2nd Ed. McGraw-Hill 1991
2. PL/SQL By Evan Bayross.