

## **Project Description**

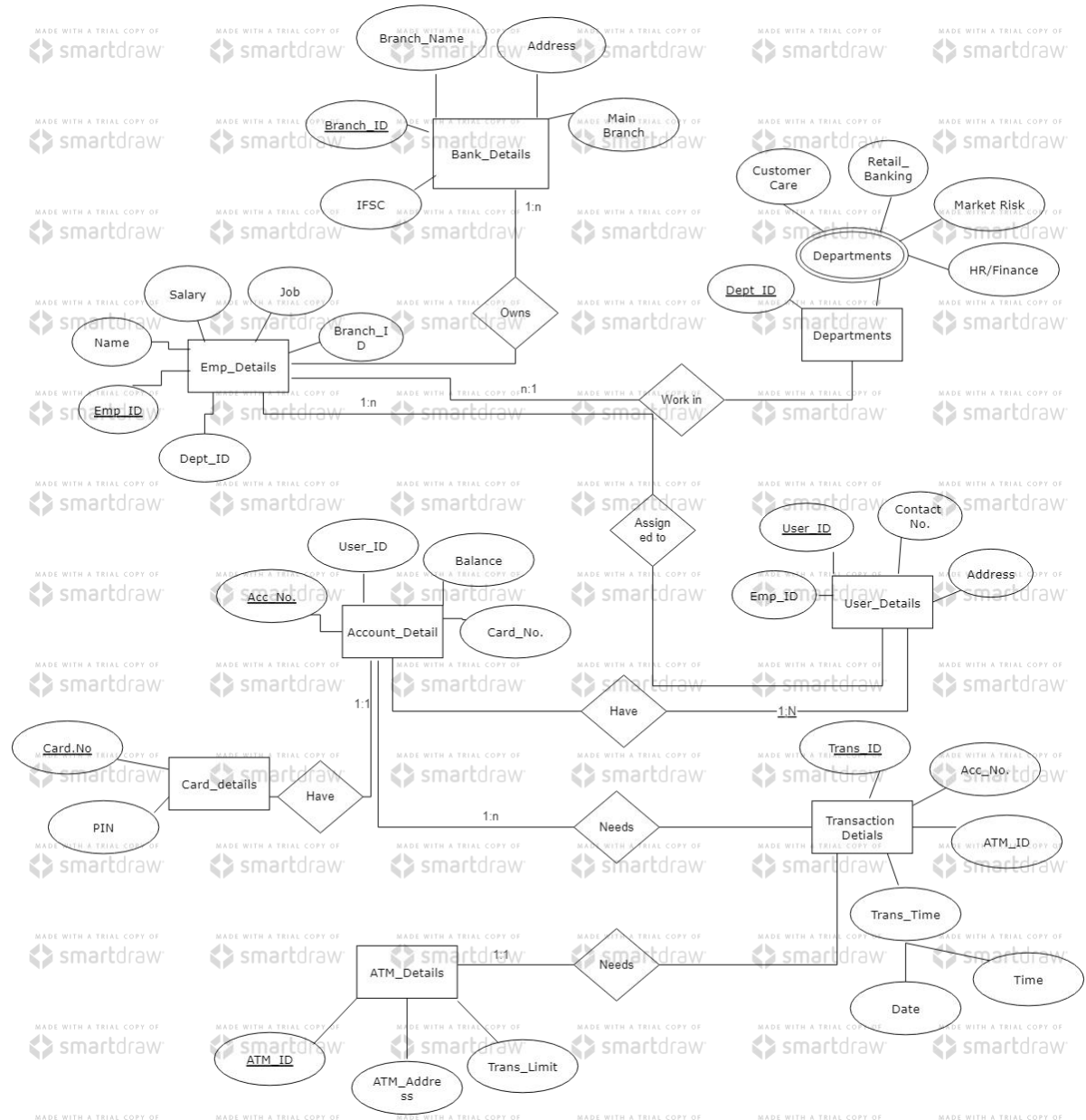
Implemented a Banking System with the appropriate data considering a single Bank and its main Branch. Used SQL to create entity tables and insert data. Furthermore, used PL/SQL to create functionalities like Debit function, Credit function and Viewing of Balance

# TABLE OF CONTENTS

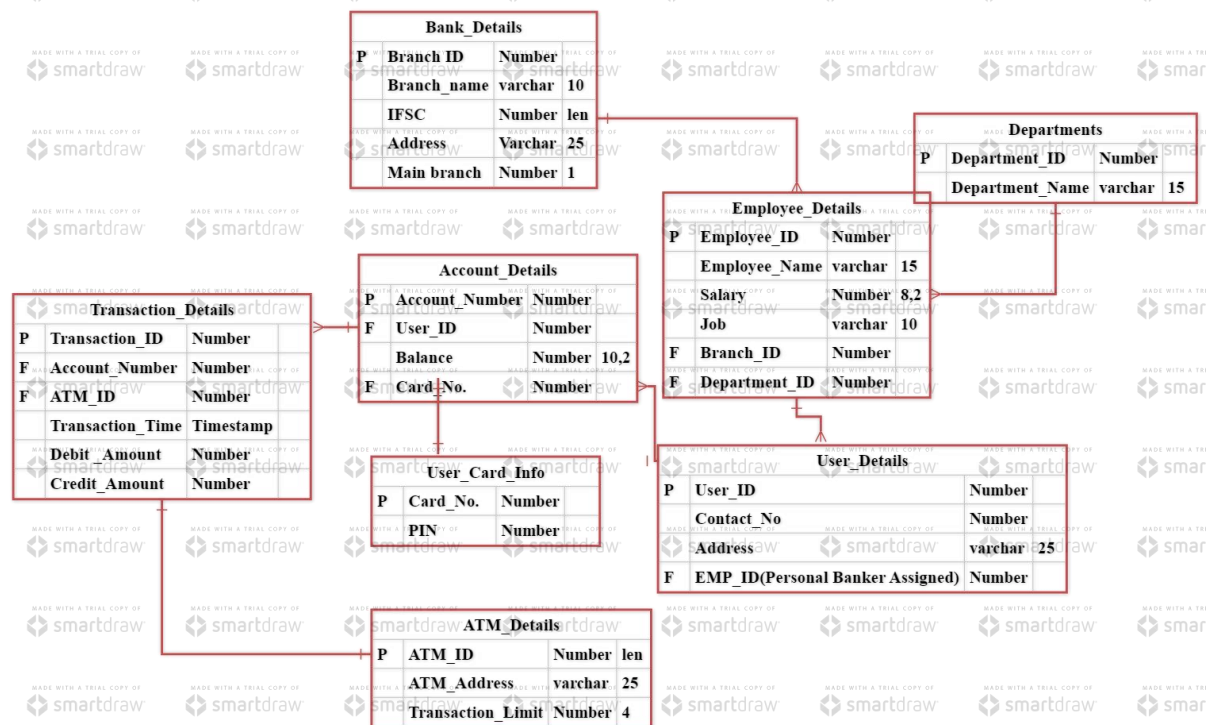
---

<b>Project Description</b>	1
<b>Table of Contents</b>	2
<b>ER DIAGRAM</b>	3
<b>SQL Queries (Input/Output)</b>	5
<b>PL/SQL Queries (Input/Output)</b>	13

# ER DIAGRAM



# Banking\_System



# SQL QUERIES

## Queries to Create Tables:

```
create table Bank(Branch_ID Number Primary Key, B_Name varchar(10),  
IFSC Number, Addr varchar(25), Main_B Number(1));
```

```
create table Department(Dept_ID Number Primary Key, Dept_Name  
varchar(10));
```

```
create table Emp(Emp_ID Number Primary Key, Emp_Name varchar(15),  
Salary Number(8,2), Job_Desc varchar(10), Branch_ID Number references  
Bank(Branch_ID), Dept_ID Number references Department(Dept_ID));
```

```
create table User_Det(Name varchar(20), ID Number Primary Key,  
Contact_No Number check (Contact_No > 999999999 and Contact_No <  
9999999999), Addr varchar(25), Emp_ID references Emp(Emp_ID));
```

```
create table Passwords(Card_No Number Primary Key, PIN Number);
```

```
create table Account( Acc_No Number Primary Key, ID Number references  
User_Det(ID), Balance Number(10, 2), Card_No Number references  
Passwords(Card_No));
```

```
create table ATM_Det(ATM_ID Number Primary Key, ATM_addr  
varchar(25), Trans_Limit number(4));
```

```
create table Transactions(Trans_ID Number Primary Key, Acc_No Number  
references Account(Acc_No), ATM_ID Number references  
ATM_Det(ATM_ID), Trans_Time Timestamp, Debit_Ammount  
Number(4), Credit_Ammount Number);
```

### Queries to Insert Data in Tables:

```
insert into bank values(1001,'PNB-Shimla','PUNB0155920','MallRoad,  
Shimla',0);
```

```
insert into bank values(1002,'PNB-SRE','PUNB0168990','Sharda Nagar,  
Saharanpur',1);
```

```
insert into bank values(1001,'PNB-Delhi','PUNB0233450','Karol Bhag,  
Delhi',0);
```

```
insert into department values(1,'Retail Banking');
```

```
insert into department values(2,'Market Risk');
```

```
insert into department values(3,'HR/Finance');
```

```
insert into department values(4,'Customer care');
```

```
insert into emp values(1,'Tommy',223413.22,'Branch Manager',1002,1);
```

```
insert into emp values(2,'Ada',56313.12,'Teller',1002,1);
```

```

insert into emp values(3,'Arthur',57313.32,'Teller',1002,1);
insert into emp values(4,'John',56843.42,'Teller',1002,1);
insert into emp values(5,'Ben',25113.41,'Security Guard',1002,1);
insert into emp values(6,'Snowy',33113.44,'Receptionist',1002,1);
insert into emp values(7,'Maya',31114.46,'Receptionist',1002,1);
insert into emp values(8,'Polly',42113.12,'Buisness Analyst',1002,2);
insert into emp values(9,'Jiren',33113.12,'Jr.Buisness Analyst',1002,2);
insert into emp values(10,'Alfie',36134.12,'Credit Analyst',1002,2);
insert into emp values(11,'Amy',35213.12,'Jr.Credit Analyst',1002,2);
insert into emp values(12,'Sheldon',764213.12,'Internal Auditor',1002,2);
insert into emp values(13,'Raj',55323.12,'Financial Risk Manager',1002,3);
insert into emp values(14,'Sami',57313.12,'Human resource
Manager',1002,3);
insert into emp values(15,'Jimmy',33113.12,'Complaint Filer',1002,3);
insert into emp values(16,'Harold',77113.12,'Lawyer',1002,3);
insert into emp values(17,'Jessi',24113.12,'Customer Care Executive',1002,4);
insert into emp values(18,'Meena',22113.12,'Customer Care
Executive',1002,4);
insert into emp values(19,'Howard',22513.74,'Customer care
Executive',1002,4);
insert into emp values(20,'Carli',33521.42,'Personal Banker',1002,1);
insert into emp values(21,'Simmi',34413.12,'Personal Banker',1002,1);
insert into emp values(22,'Toni',33883.12,'Personal Banker',1002,1);

```

```
insert into User_Det values('Sahil', 101, 9760539672, 'Saharanpur', 20);  
insert into User_Det values('Arnav', 102, 9760554672, 'Shimla', 21);  
insert into User_Det values('Ayush', 103, 9760523672, 'Bhopal', 22);  
insert into User_Det values('Sehaj', 104, 9760551234, 'Goa', 21);  
insert into User_Det values('Sailish', 105, 9723454672, 'Noida', 20);
```

```
insert into Passwords values(79805, 8930);  
insert into Passwords values(79520, 1390);  
insert into Passwords values(85805, 8293);  
insert into Passwords values(78295, 1389);  
insert into Passwords values(79201, 1937);  
insert into Passwords values(10945, 1472);  
insert into Passwords values(71895, 2345);
```

```
insert into Account values(123456, 101, 2840281.24, 85805);  
insert into Account values(123457, 101, 840281.20, 78295);  
insert into Account values(123458, 102, 1840271.54, 79201);  
insert into Account values(123459, 102, 3840271.24, 10945);  
insert into Account values(123460, 103, 284451.74, 71895);  
insert into Account values(123461, 104, 540281.26, 79805);  
insert into Account values(123462, 105, 440281.94, 79520);
```

```
insert into ATM_Det values (56, 'Saharanpur', 9000);
```



```
insert into ATM_Det values (57, 'Shimla', 5000);  
insert into ATM_Det values (58, 'Bhopal', 9900);  
insert into ATM_Det values (59, 'Goa', 8000);  
insert into ATM_Det values (60, 'Noida', 9000);
```

## Outputs of SQL:

The screenshot displays a SQL query execution environment. The top toolbar includes icons for file operations and a status bar indicating the task completion time of 0.48100001 seconds. The main workspace is divided into two panels: 'Worksheet' and 'Query Builder'. The 'Query Builder' panel contains a list of SQL queries, with the last query, 'select \* from ATM\_Det;', highlighted in yellow. Below the queries, the 'Query Result' panel shows the output of the selected query. The output is presented in a table format with columns: BRANCH\_ID, B\_NAME, IFSC, ADDR, and MAIN\_B. The table contains three rows of data for PNB branches in Shimla, Saharanpur, and Delhi. Below this, another table header 'DEPT\_ID DEPT\_NAME' is shown, followed by a list of four departments: Retail Banking, Market Risk, HR/Finance, and Customer care.

```
select * from Bank;
select * from Department;
select * from Emp;
select * from User_Det;
select * from Passwords;
select * from Account;
select * from ATM_Det;
```

BRANCH_ID	B_NAME	IFSC	ADDR	MAIN_B
1001	PNB-Shimla	PUNB0155920	MallRoad, Shimla	0
1002	PNB-SRE	PUNB0168990	Sharda Nagar, Saharanpur	1
1003	PNB-Delhi	PUNB0233450	Karol Bhag, Delhi	0

DEPT_ID	DEPT_NAME
1	Retail Banking
2	Market Risk
3	HR/Finance
4	Customer care

```

select * from Bank;
select * from Department;
select * from Emp;
select * from User_Det;
select * from Passwords;
select * from Account;
select * from ATM_Det;

```

Query Result x

Script Output x

Task completed in 0.481 seconds

EMP_ID	EMP_NAME	SALARY	JOB_DESC	BRANCH_ID	DEPT_ID
2	Ada	56313.12	Teller	1002	1
3	Arthur	57313.32	Teller	1002	1
4	John	56843.42	Teller	1002	1
16	Harold	77113.12	Lawyer	1002	3
1	Tommy	223413.22	Branch Manager	1002	1
5	Ben	25113.41	Security Guard	1002	1
6	Snowy	33113.44	Receptionist	1002	1
7	Maya	31114.46	Receptionist	1002	1
8	Polly	42113.12	Buisness Analyst	1002	2
9	Jiren	33113.12	Jr.Buisness Analyst	1002	2
10	Alfie	36134.12	Credit Analyst	1002	2

EMP_ID	EMP_NAME	SALARY	JOB_DESC	BRANCH_ID	DEPT_ID
11	Amy	35213.12	Jr.Credit Analyst	1002	2
12	Sheldon	764213.12	Internal Auditor	1002	2
13	Raj	55323.12	Financial Risk Manager	1002	3
14	Sami	57313.12	Human resource Manager	1002	3
15	Jimmy	33113.12	Complaint Filer	1002	3
17	Jessi	24113.12	Customer Care Executive	1002	4
18	Meena	22113.12	Customer Care Executive	1002	4
19	Howard	22513.74	Customer care Executive	1002	4
20	Carli	33521.42	Personal Banker	1002	1

0.48100001 seconds

Worksheet    Query Builder

```

select * from Bank;
select * from Department;
select * from Emp;
select * from User_Det;
select * from Passwords;
select * from Account;
select * from ATM_Det;

```

Query Result x    Script Output x

Task completed in 0.481 seconds

NAME	ID	CONTACT_NO	ADDR	EMP_ID
Sahil	101	9760539672	Saharanpur	20
Sailish	105	9723454672	Noida	20
Arnav	102	9760554672	Shimla	21
Ayush	103	9760523672	Bhopal	22
Sehaj	104	9760551234	Goa	21

CARD_NO	PIN
85805	8293
78295	1389
79201	1937
10945	1472
71895	2345
79805	8930
79520	1390

7 rows selected.

Worksheet

Query Builder

```

select * from Bank;
select * from Department;
select * from Emp;
select * from User_Det;
select * from Passwords;
select * from Account;
select * from ATM_Det;

```

Query Result x

Script Output x

Task completed in 0.481 seconds

7 rows selected.

ACC_NO	ID	BALANCE	CARD_NO
123456	101	2840281.24	85805
123457	101	840281.2	78295
123458	102	1840271.54	79201
123459	102	3840271.24	10945
123460	103	284451.74	71895
123461	104	540281.26	79805
123462	105	440281.94	79520

7 rows selected.

ATM_ID	ATM_ADDR	TRANS_LIMIT
56	Saharanpur	9000
57	Shimla	5000
58	Bhopal	9900
59	Goa	8000
60	Noida	9000

## **PL/SQL QUERIES**

DECLARE

id\_input number;

pin\_input number;

pin\_stored number;

card\_input number;

operation number;

account\_no number;

no\_account\_found EXCEPTION;

amount Number(6, 2);

actual\_amount Number(10,2);

low\_balance EXCEPTION;

trans\_id number;

time\_of\_trans varchar;

atm\_id\_generated number;

debit\_transaction\_complete EXCEPTION;

credit\_transaction\_complete EXCEPTION;

BEGIN

operation := &operation;

id\_input := &id\_input;

card\_input := &card\_input;

```

pin_input := &pin_input;

amount := &amount;

IF operation = 1 THEN

    select PIN into pin_stored from User_Det, Account, Passwords where
    User_Det.ID = Account.ID and Account.Card_No = Passwords.Card_No
    and User_Det.ID = id_input and Account.Card_No = card_input;

    if pin_stored = pin_input then

        dbms_output.put_line('Welcome User No.: '||id_input);

        select Acc_No into account_no from Account where Card_No =
        card_input;

        select Balance into actual_amount from User_Det, Account where
        User_Det.ID = Account.ID and User_Det.ID = id_input and
        Account.Acc_No = account_no;

        IF amount > actual_amount then

            raise low_balance;

        ELSE

            trans_id := floor(dbms_random.value*100);

            select to_char(sysdate, 'HH:MM:SS') into time_of_trans from dual;

            select ATM_ID into atm_id_generated from User_Det, ATM_Det
            where User_Det.Addr = ATM_Det.ATM_addr and User_Det.ID = id_input;

            INSERT INTO Transactions values (trans_id, account_no,
            atm_id_generated, time_of_trans, amount, NULL);

            UPDATE Account SET Balance = actual_amount-amount where
            Card_No = card_input;

        END IF;

```

```

else

    raise no_account_found;

end if;

raise debit_transaction_complete;

ELSIF operation = 2 THEN

    select PIN into pin_stored from User_Det, Account, Passwords where
    User_Det.ID = Account.ID and Account.Card_No = Passwords.Card_No
    and User_Det.ID = id_input and Account.Card_No = card_input;

    dbms_output.put_line('Doing Credit Transfer');

    if pin_stored = pin_input then

        dbms_output.put_line('Welcome User No.: '||id_input);

        select Acc_No into account_no from Account where Card_No =
        card_input;

        select Balance into actual_amount from User_Det, Account where
        User_Det.ID = Account.ID and User_Det.ID = id_input and
        Account.Acc_No = account_no;

        trans_id := floor(dbms_random.value*100);

        select to_char(sysdate, 'HH:MM:SS') into time_of_trans from dual;

        select ATM_ID into atm_id_generated from User_Det, ATM_Det
        where User_Det.Addr = ATM_Det.ATM_addr and User_Det.ID = id_input;

        INSERT INTO Transactions values (trans_id, account_no,
        atm_id_generated, time_of_trans, NULL, amount);

        UPDATE Account SET Balance = actual_amount+amount where
        Card_No = card_input;

    else

```



```

        raise no_account_found;

    end if;

    raise credit_transaction_complete;

ELSIF operation = 3 THEN

    select PIN into pin_stored from User_Det, Account, Passwords where
    User_Det.ID = Account.ID and Account.Card_No = Passwords.Card_No
    and User_Det.ID = id_input and Account.Card_No = card_input;

    if pin_stored = pin_input then

        dbms_output.put_line('Welcome User No.: '||id_input);

        select Acc_No into account_no from Account where Card_No =
        card_input;

        select Balance into actual_amount from User_Det, Account where
        User_Det.ID = Account.ID and User_Det.ID = id_input and
        Account.Acc_No = account_no;

        dbms_output.put_line(actual_amount);

    else

        raise no_account_found;

    end if;

END IF;

EXCEPTION

when no_account_found then dbms_output.put_line('No Record for Account
found.');
```

```

when no_data_found then dbms_output.put_line('No account exists');
```

```

when too_many_rows then dbms_output.put_line('Too many records found');
```

```

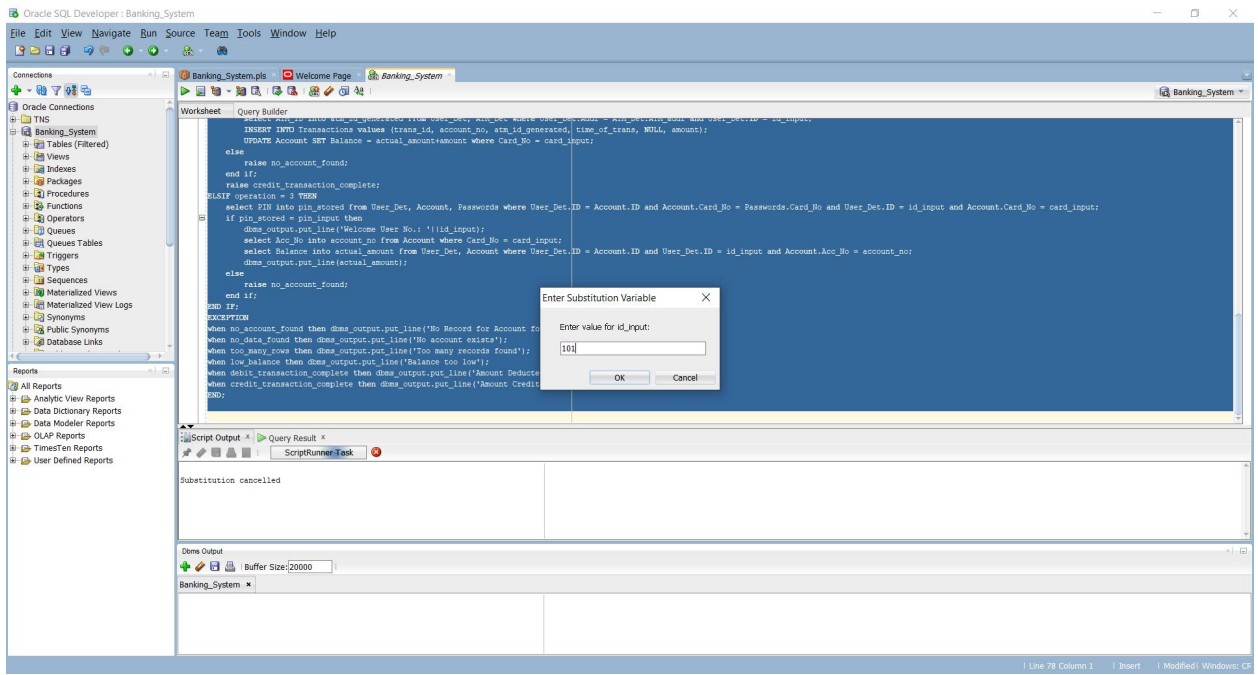
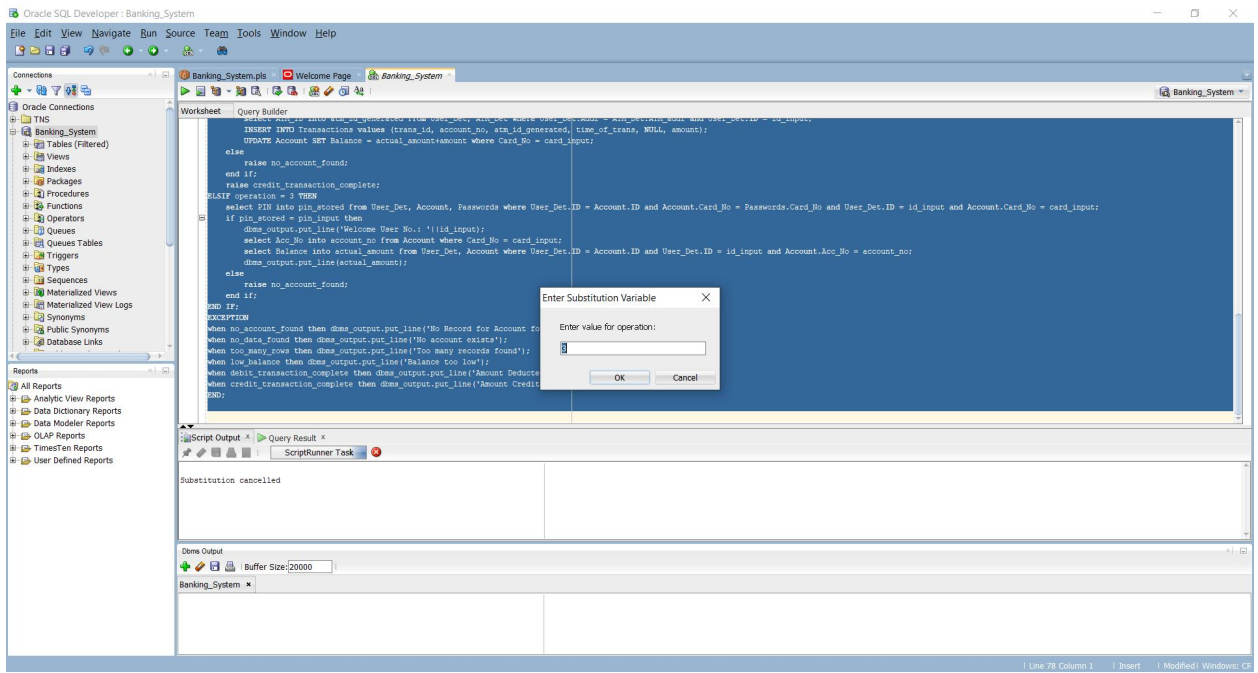
when low_balance then dbms_output.put_line('Balance too low');
```

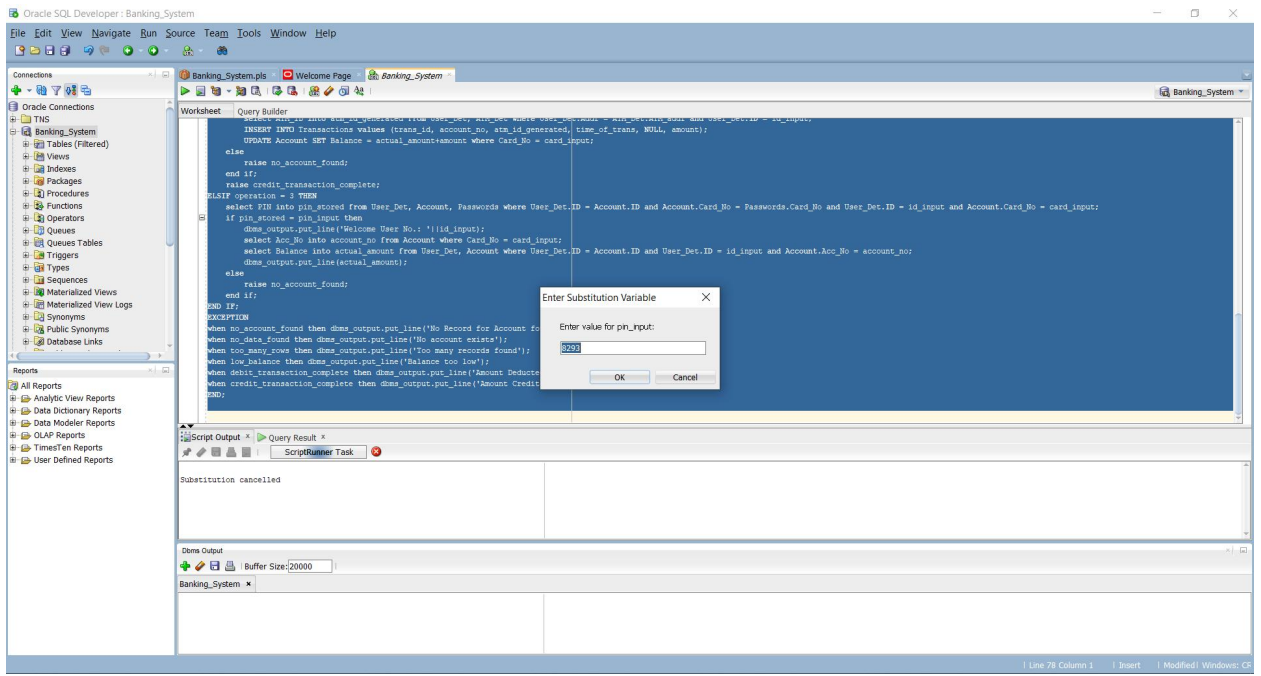
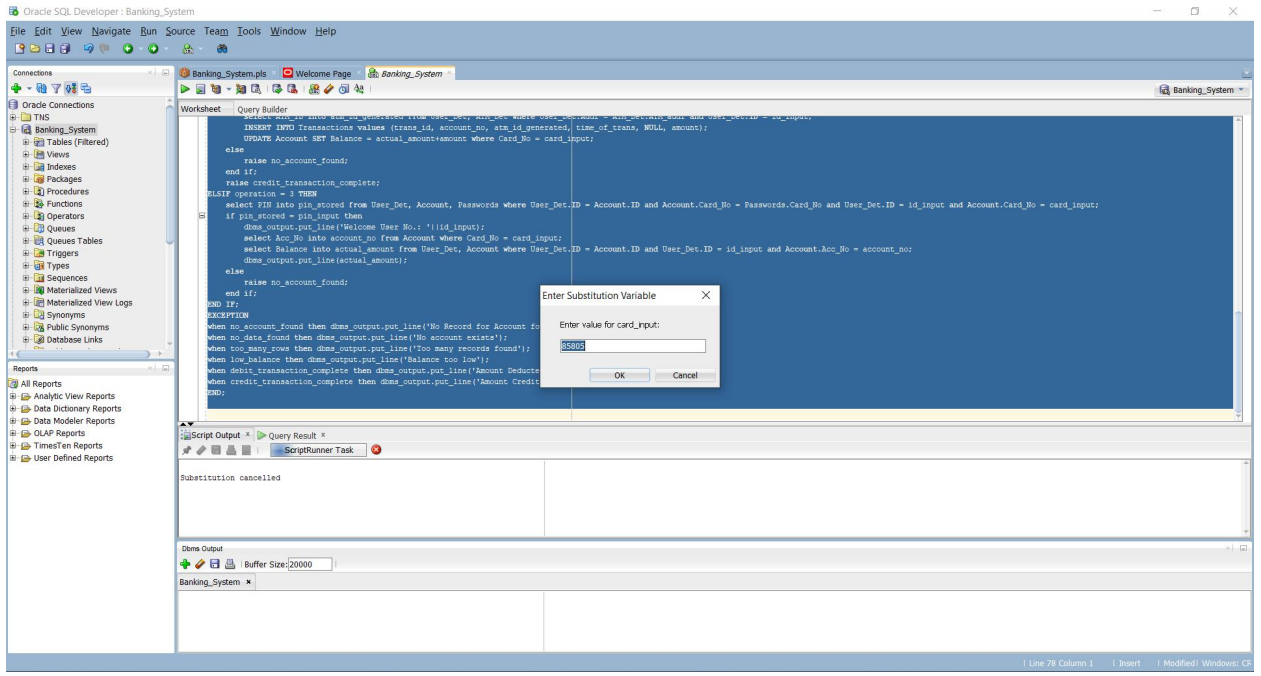
```
when debit_transaction_complete then dbms_output.put_line('Amount  
Deducted');
```

```
when credit_transaction_complete then dbms_output.put_line('Amount  
Credited');
```

```
END;
```

Output of PL/SQL(Checking Balance):





Script Output x

Query Result x

Task completed in 99.831 seconds

```
when debit_transaction_complete then dbms_output.put_line('Amount Deducted');
when credit_transaction_complete then dbms_output.put_line('Amount Credited');
END;

PL/SQL procedure successfully completed.
```

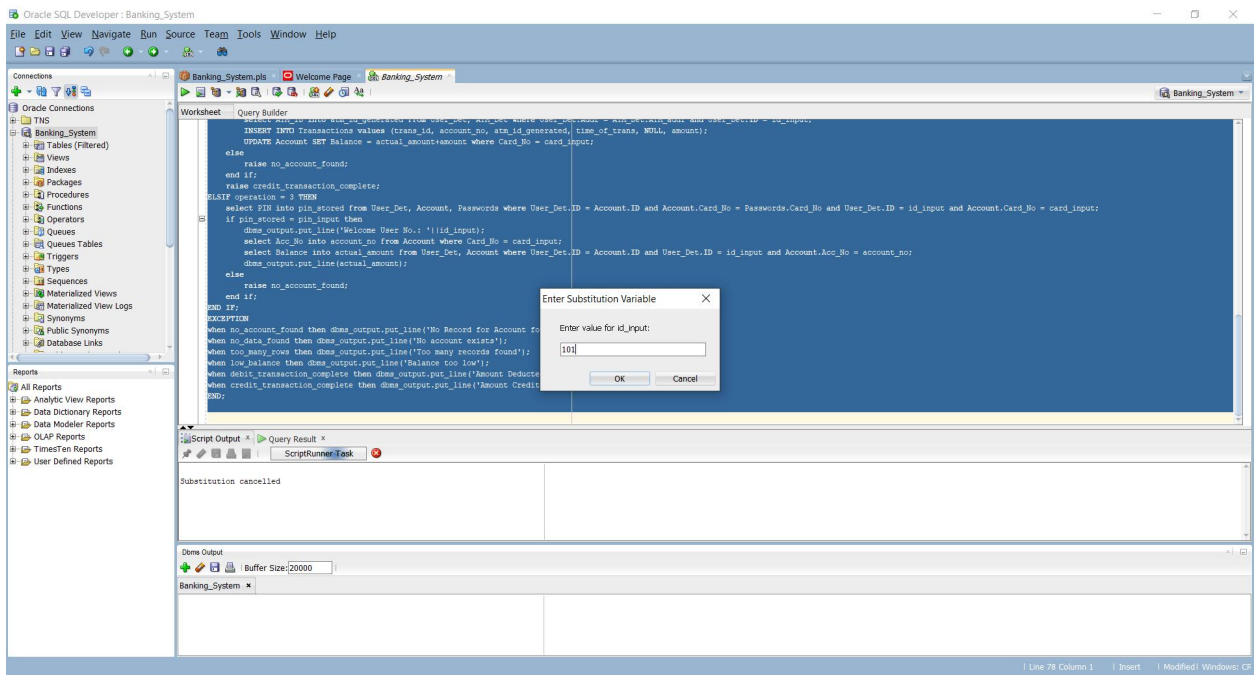
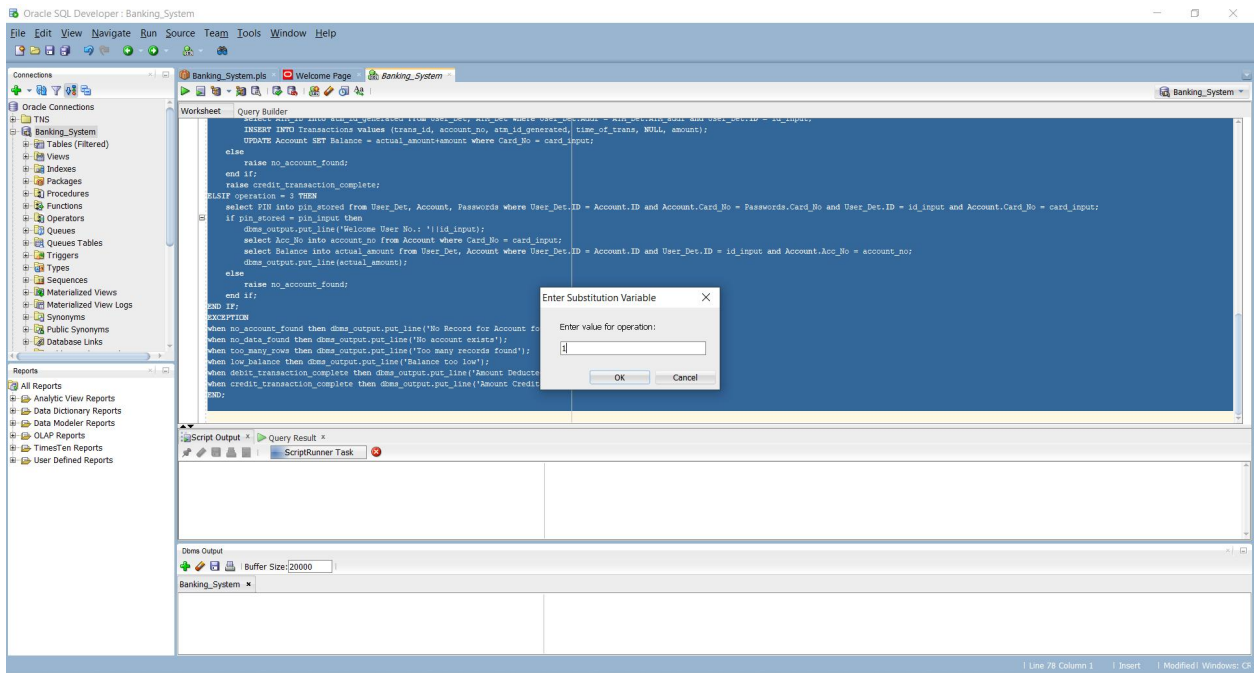
Dbms Output

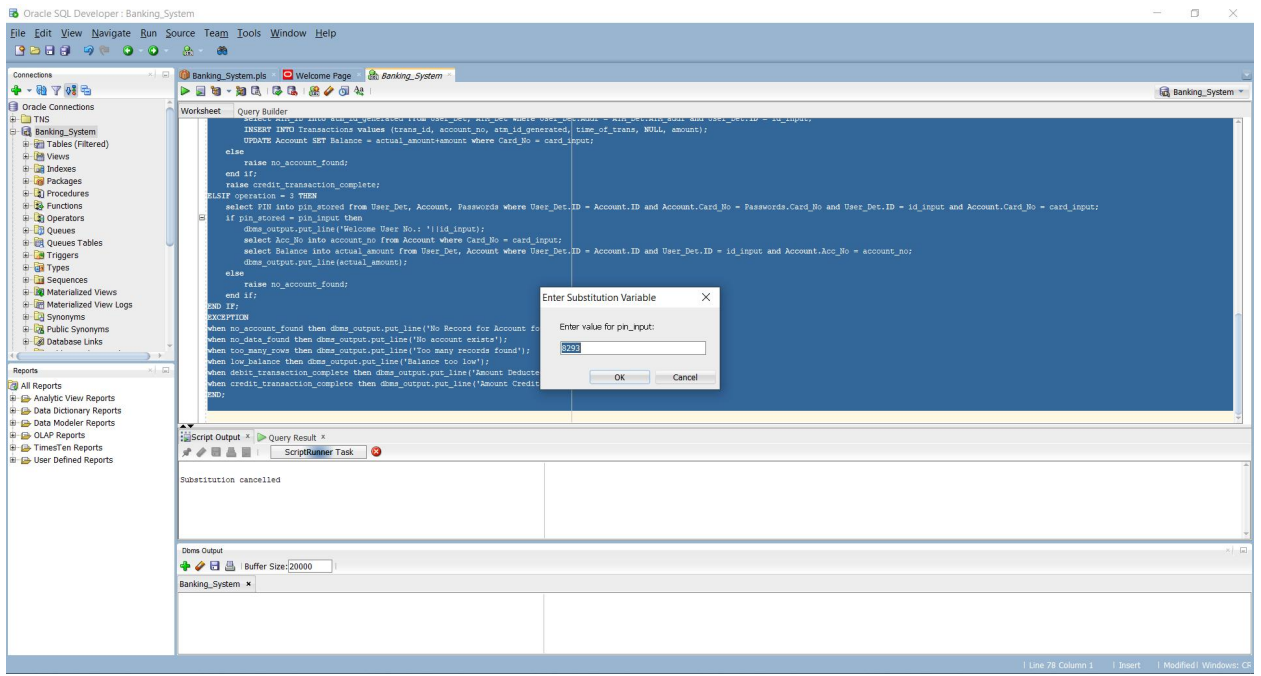
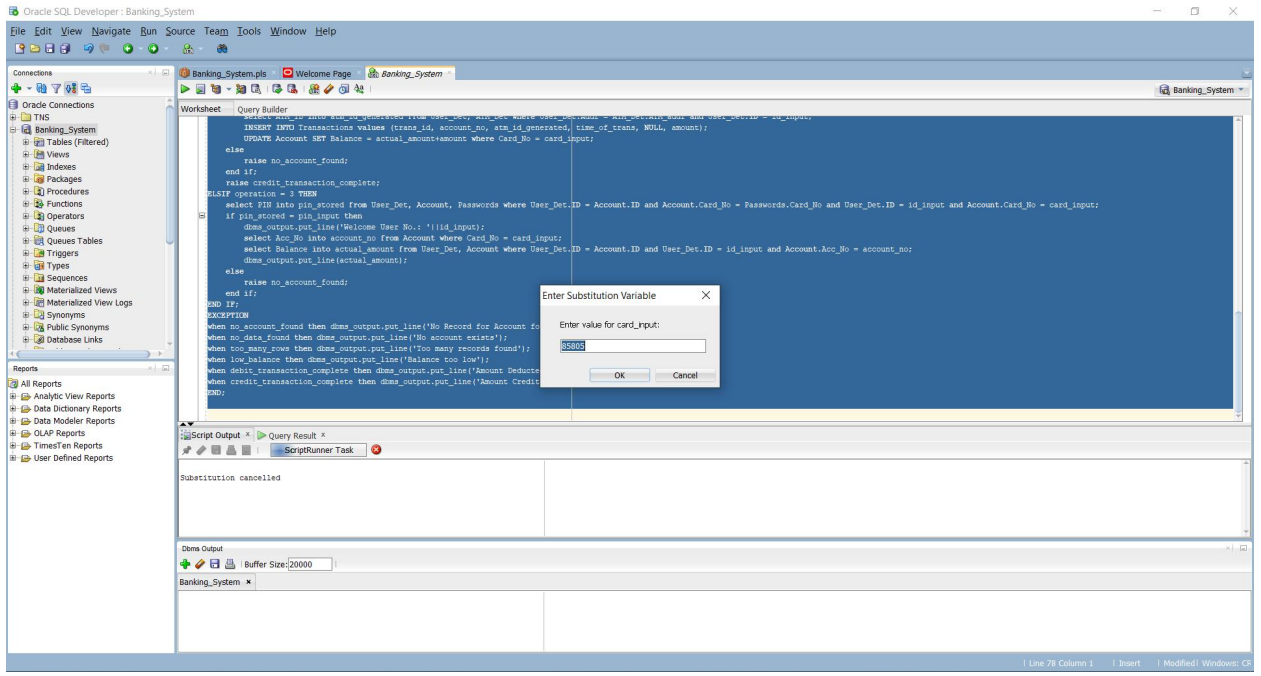
Buffer Size: 20000

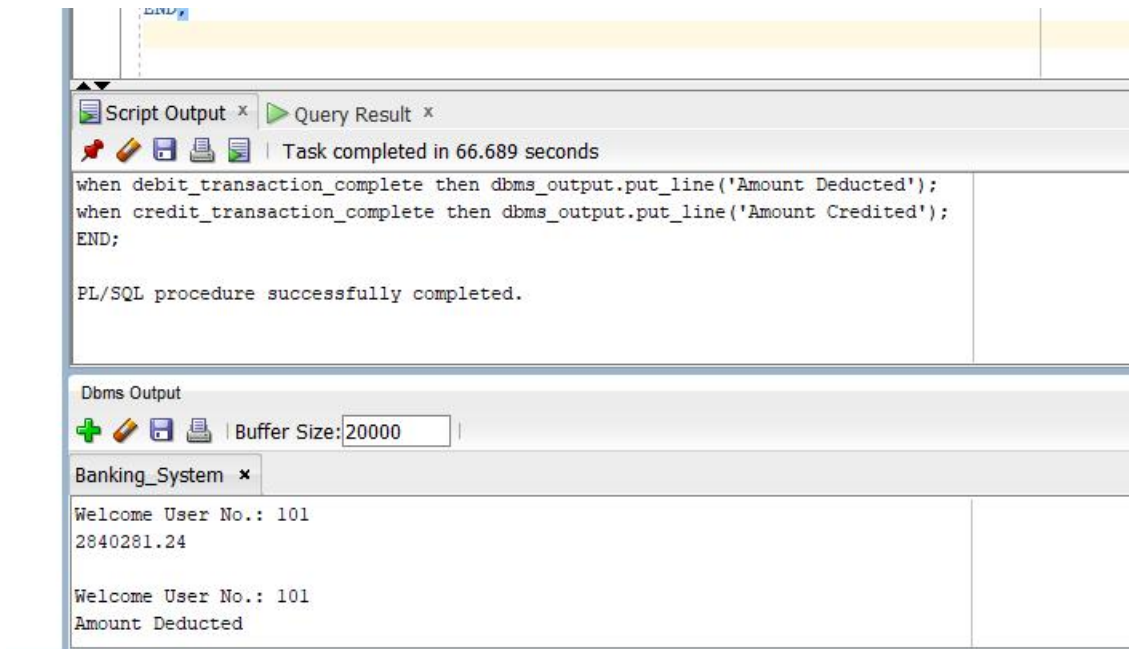
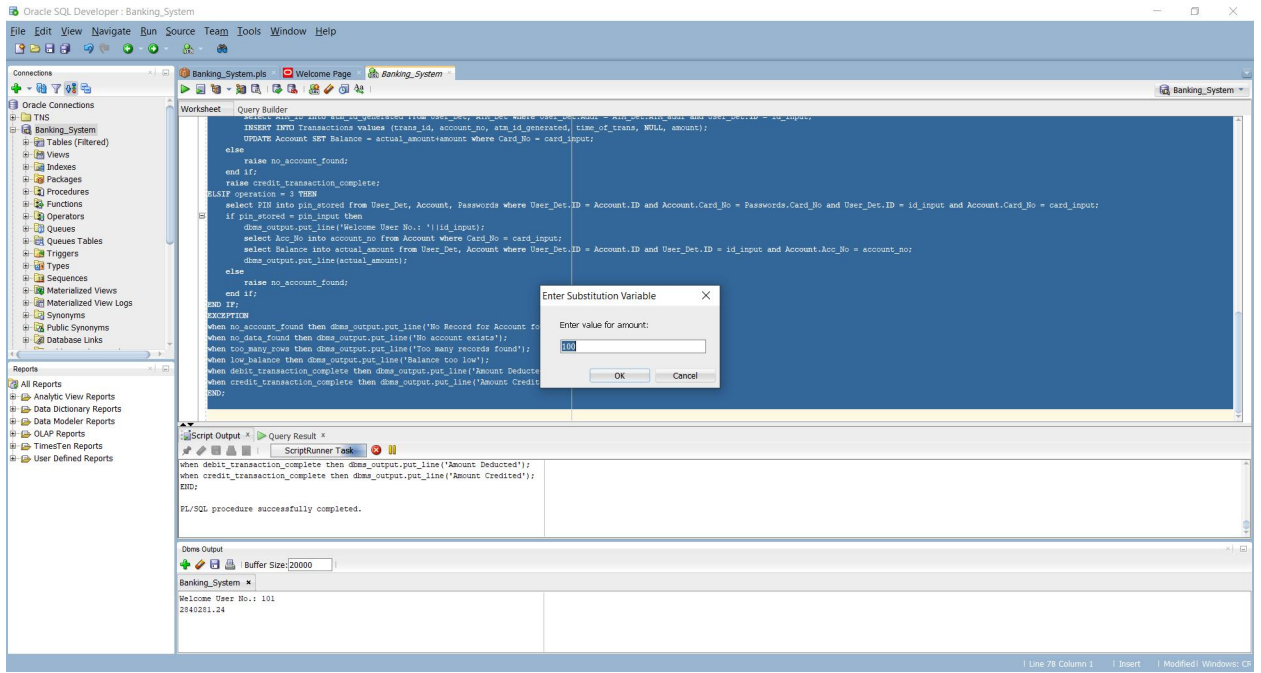
Banking\_System x

Welcome User No.: 101  
2840281.24

### Output of PL/SQL(Debiting):









```
when credit_transaction_complete then dbms_output.put_line('Amount Credited');  
END;
```

Script Output x

Query Result x

Task completed in 12.219 seconds

```
when debit_transaction_complete then dbms_output.put_line('Amount Deducted');  
when credit_transaction_complete then dbms_output.put_line('Amount Credited');  
END;
```

PL/SQL procedure successfully completed.

Dbms Output

Buffer Size: 20000

Banking\_System x

Amount Deducted

Welcome User No.: 101  
2840181.24

## Output of PL/SQL(Crediting):

Oracle SQL Developer: Banking\_System

Connections: Banking\_System.pls, Banking\_System.pls, Welcome Page, Banking\_System

Worksheet: Query Builder

```

DECLARE
  id_input number;
  pin_input number;
  pin_stored number;
  card_input number;
  operation number;
  account_no number;
  no_account_found EXCEPTION;
  amount Number(4, 2);
  actual_amount Number(10,2);
  low_balance EXCEPTION;
  trans_id number;
  time_of_trans varchar2;
  atm_id_generated number;
  debit_transaction_complete EXCEPTION;
  credit_transaction_complete EXCEPTION;
  BEGIN
    operation := &operation;
    id_input := &id_input;
    card_input := &card_input;
    pin_input := &pin_input;
    amount := &amount;
    IF operation = 1 THEN

```

Enter Substitution Variable

Enter value for operation:

OK Cancel

Query Result x1 Script Output x1

ScriptRunner Task

7 rows selected.

ACC_NO	ID	BALANCE	CARD_NO
123454	101	2040281.24	85005
123457	101	1440201.2	70296
123458	102	1040271.54	79201
123459	102	3040271.24	10945
123460	103	2044451.74	71895

Demo Output

Buffer Size:20000

Line 1 Column 1 | Insert | Modified | Windows: CF

Oracle SQL Developer: Banking\_System

Connections: Banking\_System.pls, Welcome Page, Banking\_System

Worksheet: Query Builder

```

INSERT INTO Transactions values (trans_id, account_no, atm_id_generated, time_of_trans, NULL, amount);
UPDATE Account SET Balance = actual_amount-amount where Card_No = card_input;
else
  raise no_account_found;
end if;
raise credit_transaction_complete;
--IF operation = 1 THEN
select PIN into pin_stored from User_Det, Account, Passwords where User_Det.ID = Account.ID and Account.Card_No = Passwords.Card_No and User_Det.ID = id_input and Account.Card_No = card_input;
if pin_stored = pin_input then
  dms_output.put_line('Welcome User No: '||id_input);
  select Acc_No into account_no from Account where Card_No = card_input;
  select Balance into actual_amount from User_Det, Account where User_Det.ID = Account.ID and User_Det.ID = id_input and Account.Acc_No = account_no;
  dms_output.put_line(actual_amount);
else
  raise no_account_found;
end if;
EXCEPTION
when no_account_found then dms_output.put_line('No Record for Account No');
when no_data_found then dms_output.put_line('No account exists!');
when too_many_rows then dms_output.put_line('Too many records found!');
when low_balance then dms_output.put_line('Balance too low!');
when debit_transaction_complete then dms_output.put_line('Amount Deducted');
when credit_transaction_complete then dms_output.put_line('Amount Credit');
END;

```

Enter Substitution Variable

Enter value for id\_input:

101

OK Cancel

Script Output x1 Query Result x1

ScriptRunner Task

Substitution cancelled

Demo Output

Buffer Size:20000

Banking\_System x1

Line 78 Column 1 | Insert | Modified | Windows: CF

