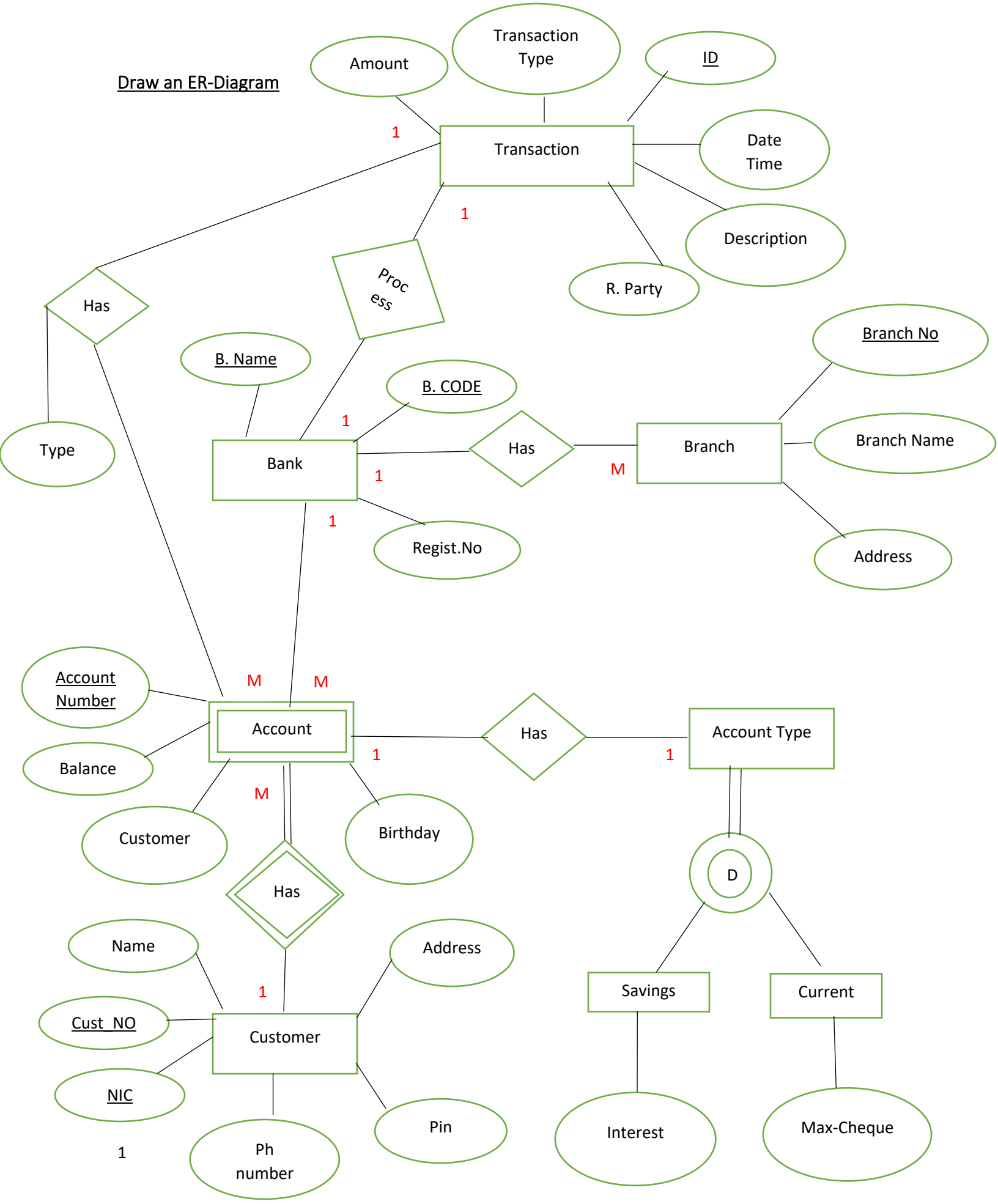
10/23/2020

Advanced Database Management System

Course Work 2

Jadhusan M S
INDEX NO: COHNDNE201F-021

Draw an ER-Diagram



Assumptions.

• All the accounts must have a bank. • All the branches must have a bank.

• All the customers must have an account. • All the accounts must have a customer.

• All the accounts must have an account type. • Al the account type must have an account.

• All the account type are categorized as saving and current.

• All the bank must have a transaction.

• All the transaction must have an accounts.

Questions 1

1. How many entities are there in the ER diagram? What are they?

8 Entities Which Includes:

- Transaction
- Bank
- Branch
- Account
- Account Type
- Savings
- Current
- Customer

2. Are all the entities you identified strong?

- NO

a. If not, what are the weak entities you identified?

- Account Entity

b. How did you identify the weak entities?

- Since, if there is no customer to the bank there will be no account. furthermore, the account has no primary key.

Questions 2

1st Q "When save these records in the oracle database, you have to categorize customers based on the age (If customer age is greater than 18, consider as adult and others consider as Teenagers).Finally you have to save these records into two different tables. Write PL/SQL Stored procedure to categorized and store the customer details in the oracle database."

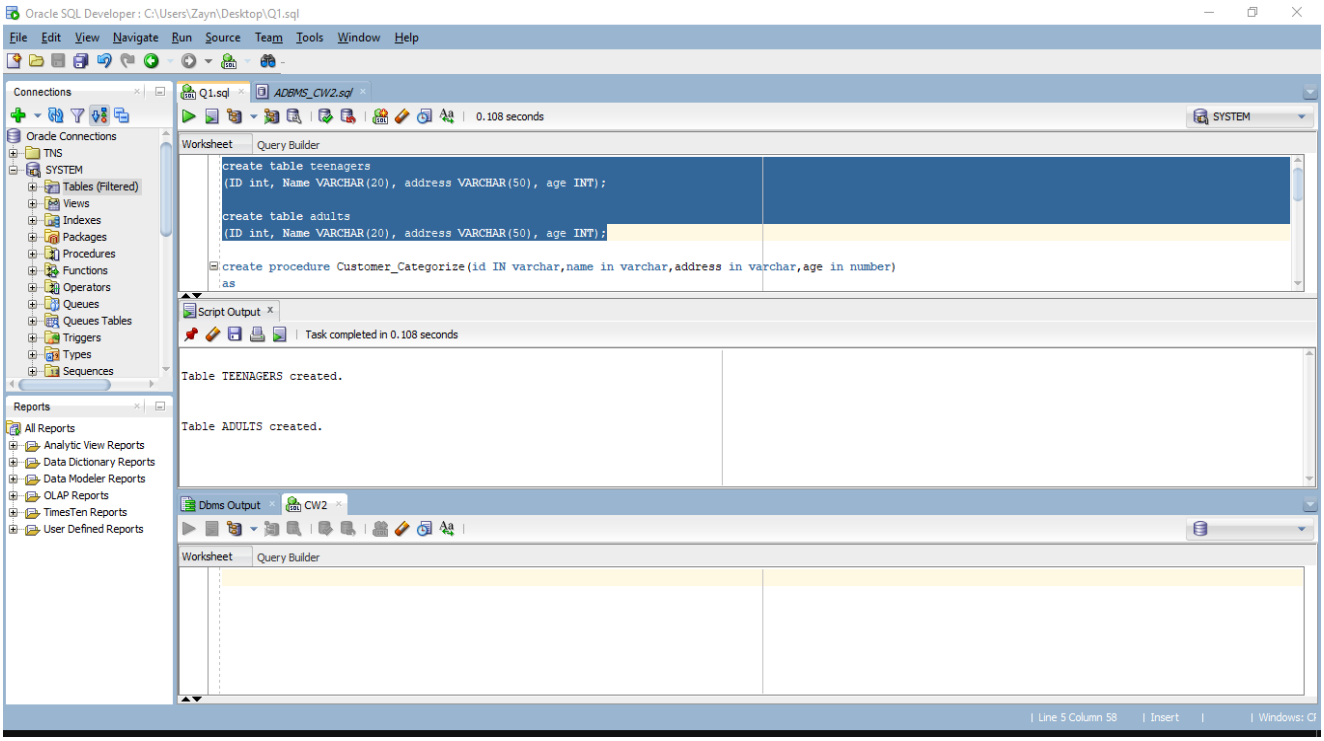
Script

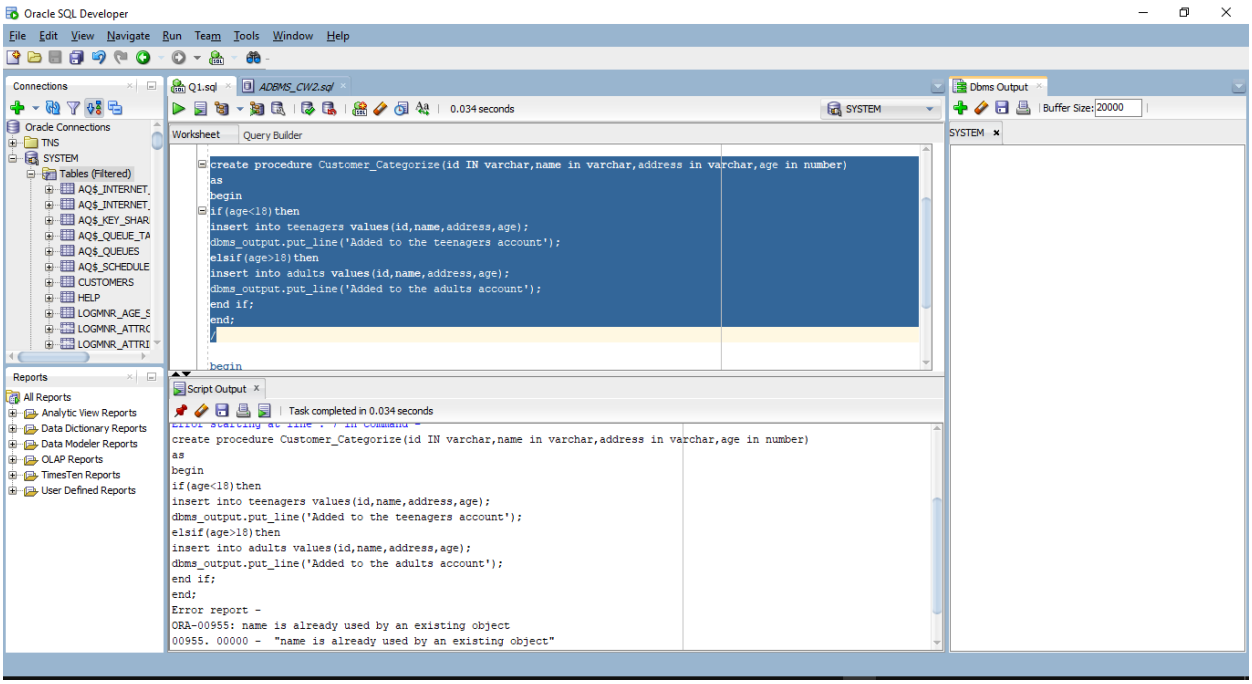
```
create table teenagers

(ID int, Name VARCHAR(20), address VARCHAR(50), age INT);

create table adults

(ID int, Name VARCHAR(20), address VARCHAR(50), age INT);


create procedure Customer_Categorize(id IN varchar,name in varchar,address in varchar,age in number)

as

begin

if(age<18)then

insert into teenagers values(id,name,address,age);

dbms_output.put_line('Added to the teenagers account');

elsif(age>18)then

insert into adults values(id,name,address,age);

dbms_output.put_line('Added to the adults account');

end if;

end;

/

begin

customer_categorize('1','Allan','NewYork',26);

end;


begin
```

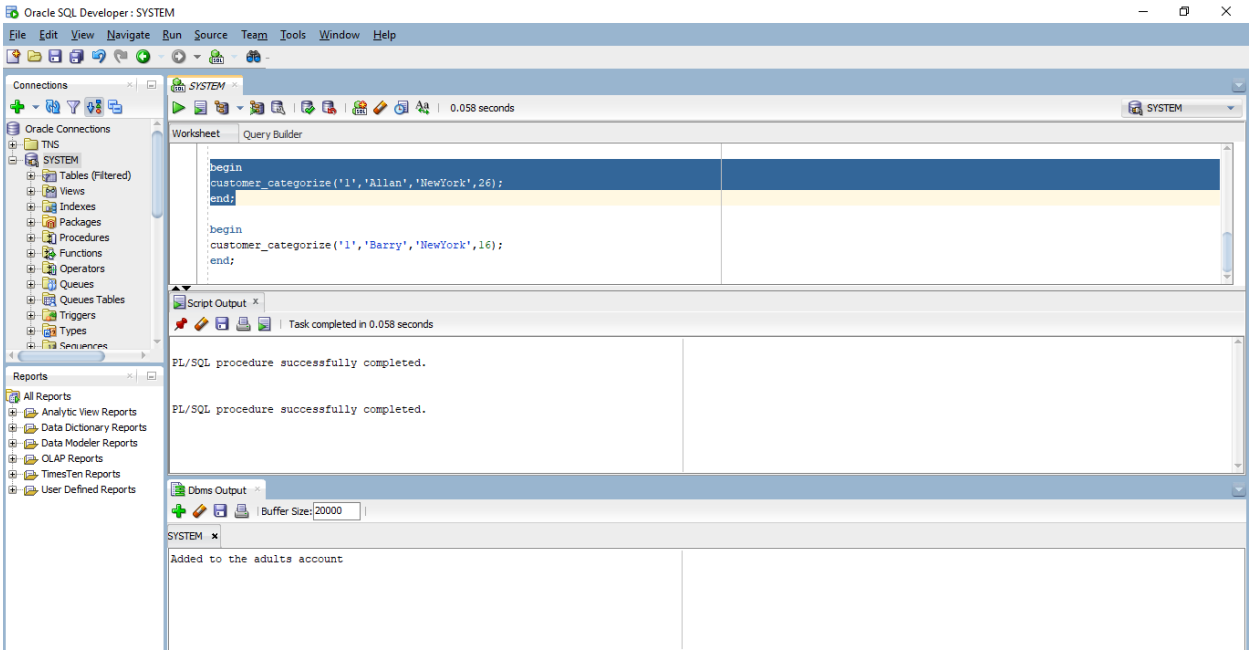
customer_categorize('1','Barry','NewYork',16);

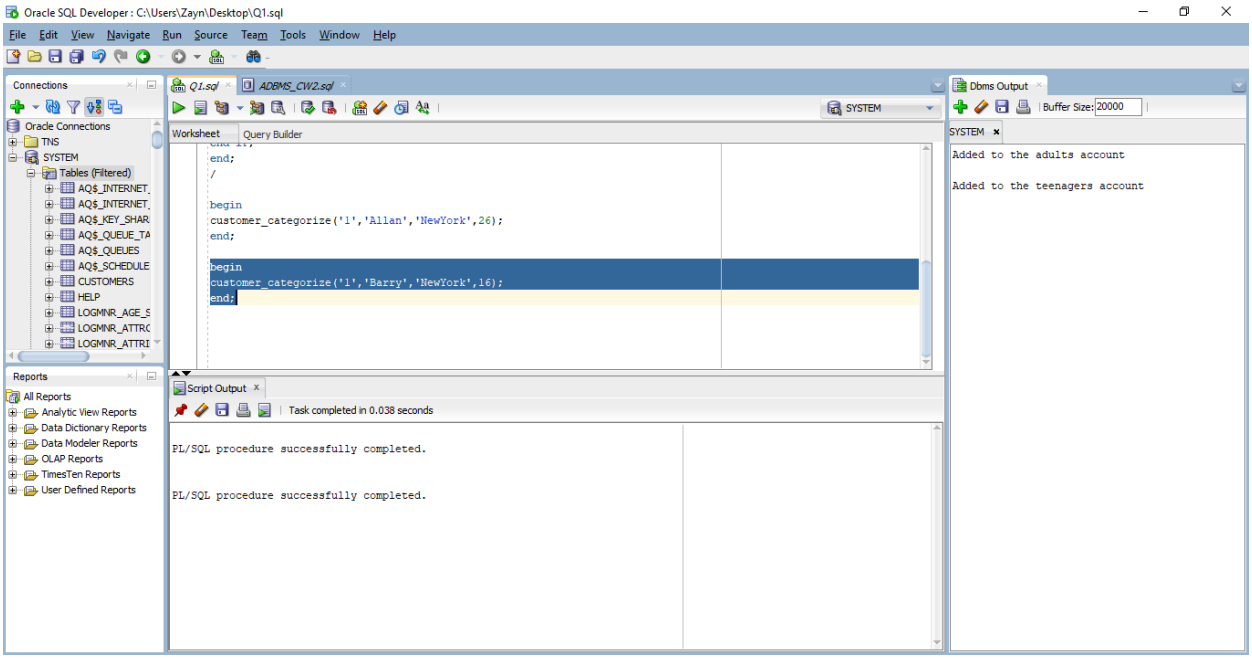
end;

SCREENSHOTs






Oracle SQL Developer : SYSTEM
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
TNS
SYSTEM
Tables (Filtered)
Views
Indexes
Packages
Procedures
Functions
Operators
Queues
Queues Tables
Triggers
Types
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
SYSTEM
0.058 seconds
Worksheet
Query Builder
begin
customer_categorize('1','Allan','NewYork',26);
end;
begin
customer_categorize('1','Barry','NewYork',16);
end;
Script Output
Task completed in 0.058 seconds
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
Dbms Output
Buffer Size: 20000
SYSTEM
Added to the adults account

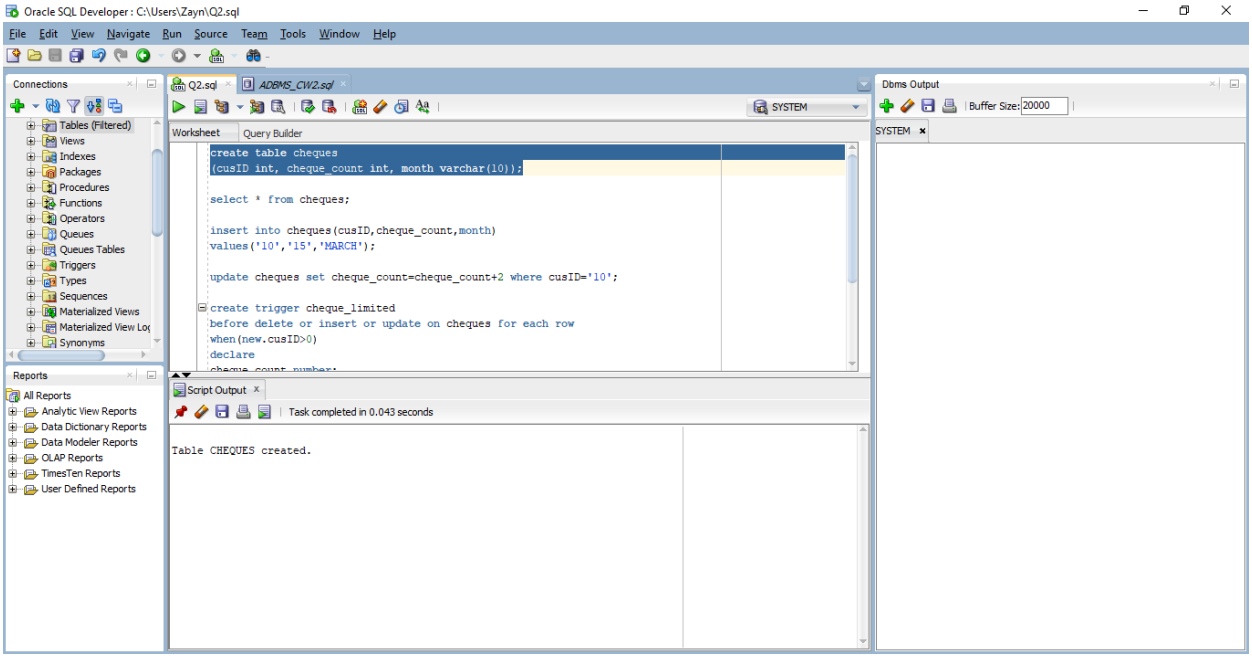

Oracle SQL Developer : C:\Users\Zayn\Desktop\Q1.sql
File Edit View Navigate Run Source Team Tools Window Help
Q1.sql
ADBMS_CW2.sql
Connections
Oracle Connections
TNS
SYSTEM
Tables (Filtered)
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
SYSTEM
Worksheet
Query Builder
end;
/
begin
customer_categorize('1','Allan','NewYork',26);
end;
begin
customer_categorize('1','Barry','NewYork',16);
end;
Script Output
Task completed in 0.038 seconds
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
Dbms Output
Buffer Size: 20000
SYSTEM
Added to the adults account
Added to the teenagers account

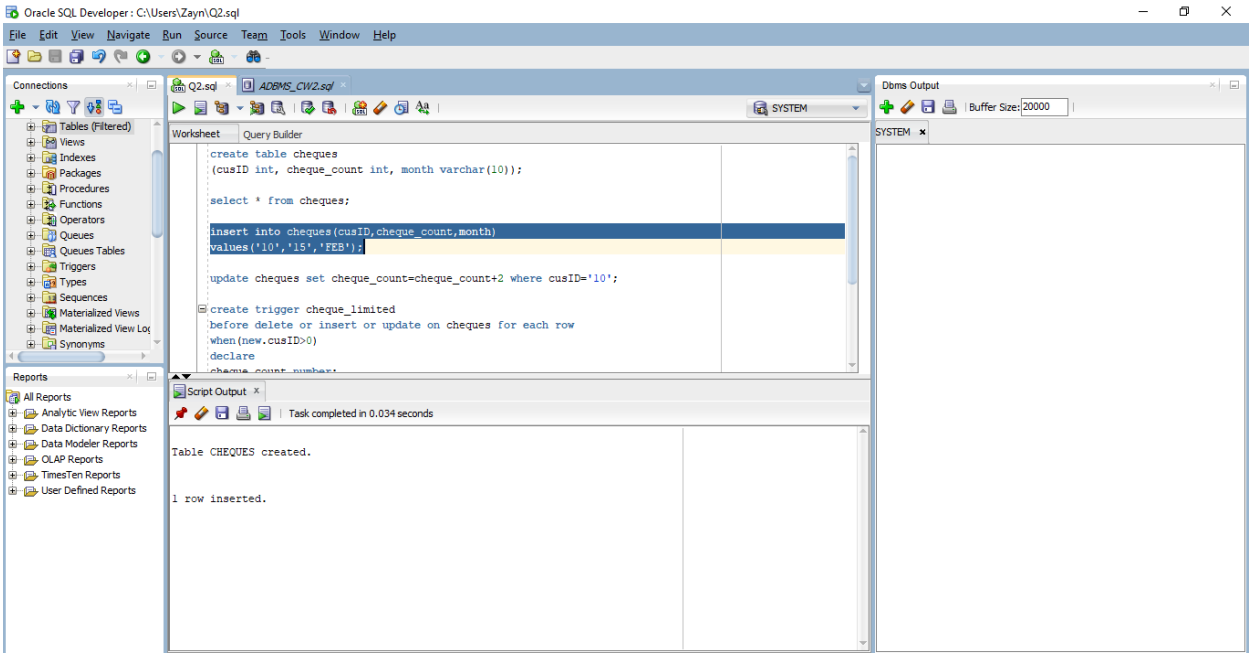
2nd Q "There is a limitation to writing cheques for current accounts holders per month which is 20. If customer writes more than 20 cheques per month write PL/SQL triggers to notify that operation."

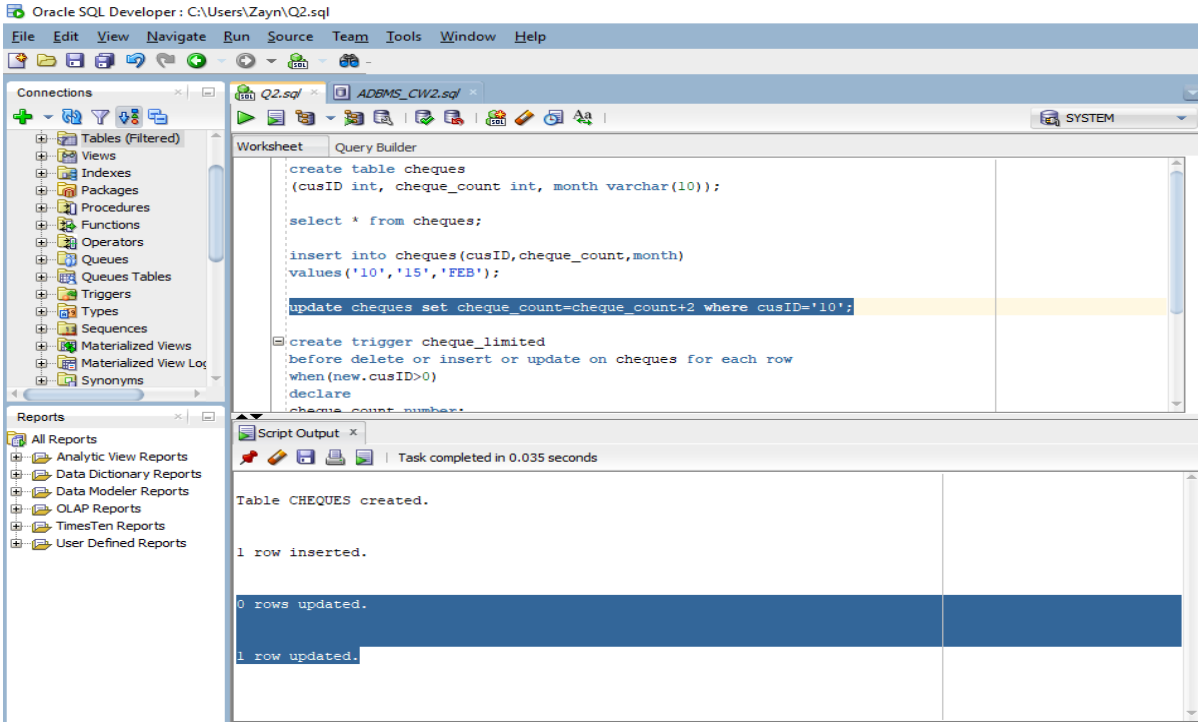
Script

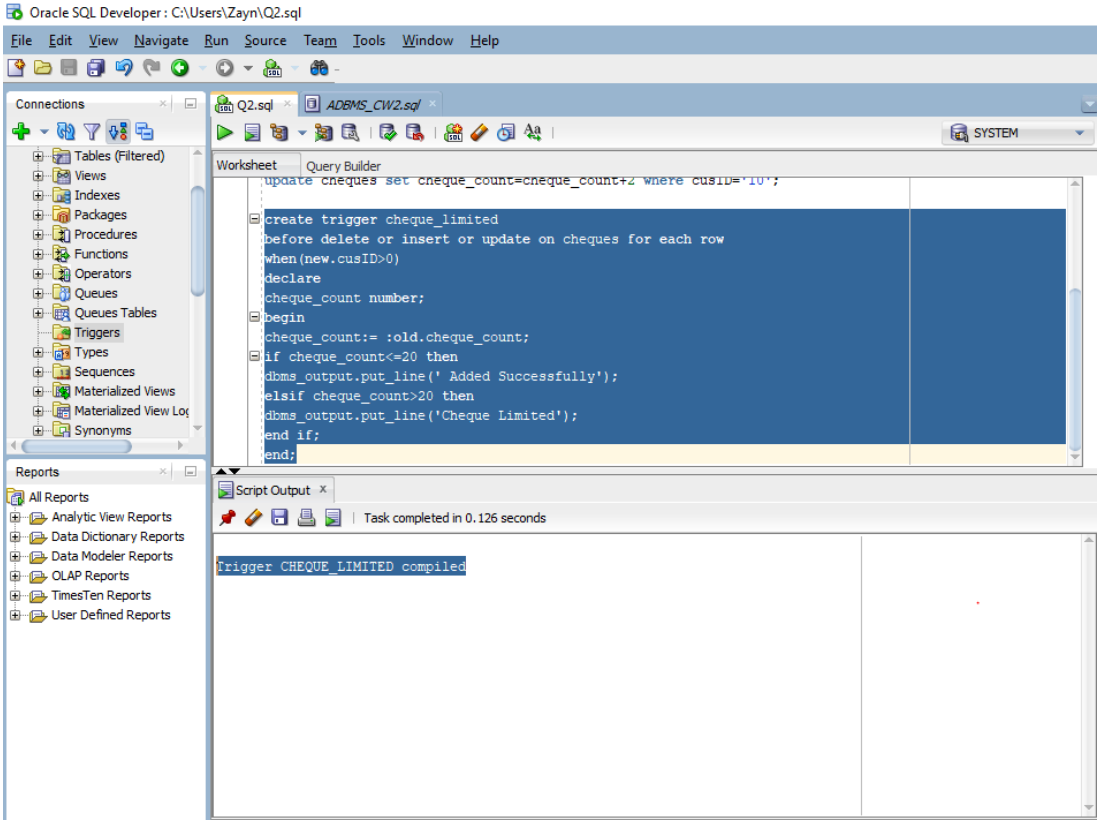
```
create table cheques
(cusID int, cheque_count int, month varchar(10));

select * from cheques;

insert into cheques(cusID,cheque_count,month)
values('10','15','FEB');

update cheques set cheque_count=cheque_count+2 where cusID='10';

create trigger cheque_limited
before delete or insert or update on cheques for each row
when(new.cusID>0)
declare
cheque_count number;
begin
cheque_count:= :old.cheque_count;
if cheque_count<=20 then
dbms_output.put_line(' Added Successfully');
elsif cheque_count>20 then
dbms_output.put_line('Cheque Limited');
end if;
end;
```

SCREENSHOTs






Oracle SQL Developer : C:\Users\Zayn\Q2.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Tables (Filtered)
Views
Indexes
Packages
Procedures
Functions
Operators
Queues
Queues Tables
Triggers
Types
Sequences
Materialized Views
Materialized View Log
Synonyms
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Q2.sql
ADBMS_CW2.sql
SYSTEM
Worksheet
Query Builder
create table cheques
(cusID int, cheque_count int, month varchar(10));
select * from cheques;
insert into cheques(cusID,cheque_count,month)
values('10','15','FEB');
update cheques set cheque_count=cheque_count+2 where cusID='10';
create trigger cheque_limited
before delete or insert or update on cheques for each row
when(new.cusID>0)
declare
Script Output
Task completed in 0.035 seconds
Table CHEQUES created.
1 row inserted.
0 rows updated.
1 row updated.


Oracle SQL Developer : C:\Users\Zayn\Q2.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Tables (Filtered)
Views
Indexes
Packages
Procedures
Functions
Operators
Queues
Queues Tables
Triggers
Types
Sequences
Materialized Views
Materialized View Log
Synonyms
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Q2.sql
ADBMS_CW2.sql
SYSTEM
Worksheet
Query Builder
update cheques set cheque_count=cheque_count+2 where cusID='10';
create trigger cheque_limited
before delete or insert or update on cheques for each row
when(new.cusID>0)
declare
cheque_count number;
begin
cheque_count:= :old.cheque_count;
if cheque_count<=20 then
dbms_output.put_line(' Added Successfully');
elsif cheque_count>20 then
dbms_output.put_line('Cheque Limited');
end if;
end;
Script Output
Task completed in 0.126 seconds
Trigger CHEQUE_LIMITED compiled

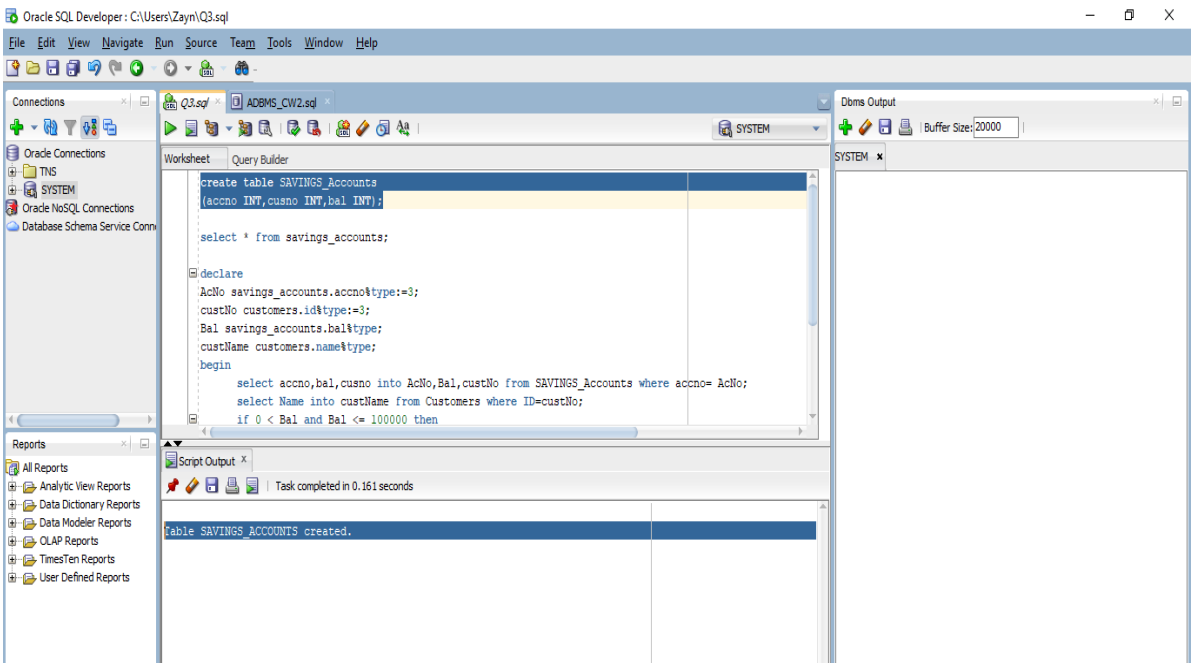
3rd Q "Saving account holders can be divide into low, medium and high groups based on their saving balance. Write a PL/SQL Code to identify these customers and give a separate flag for each customer."

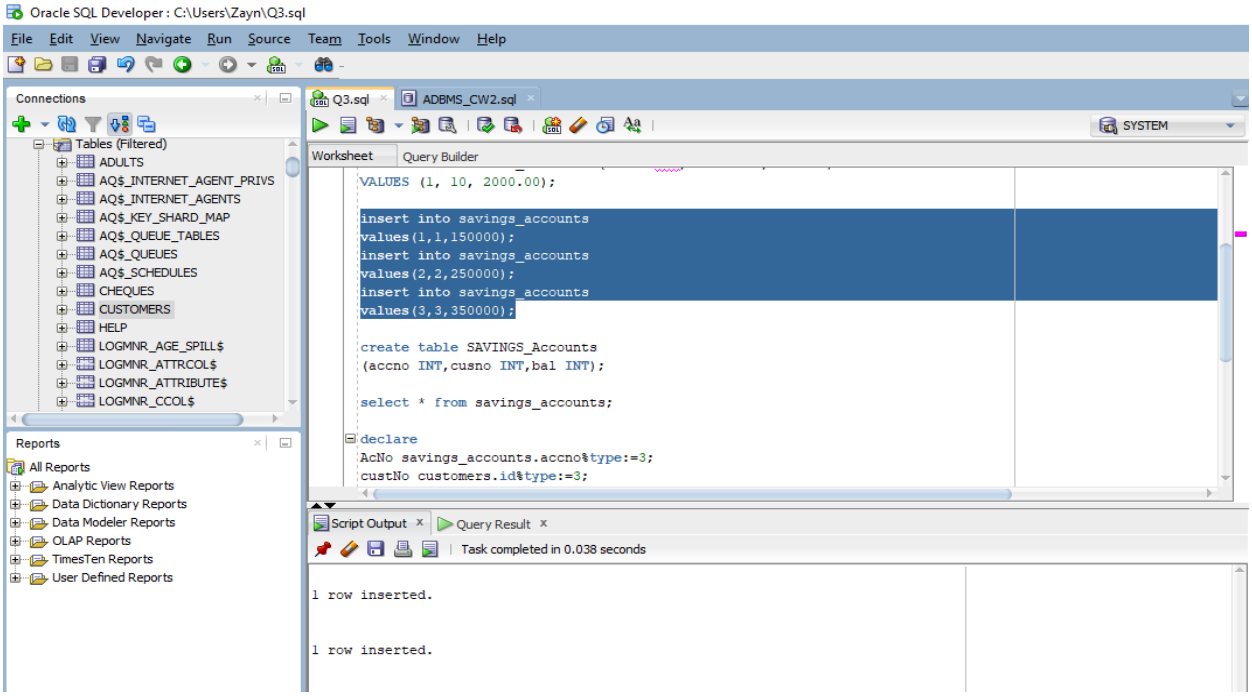
Script

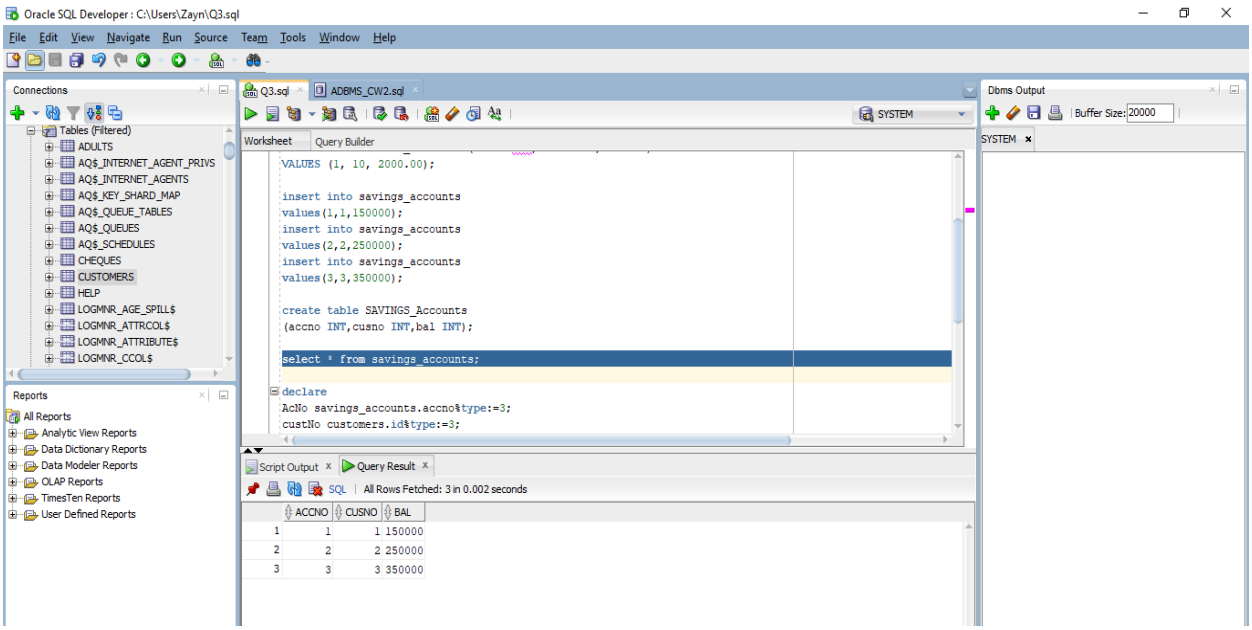
```
create table SAVINGS_Accounts
(accno INT,cusno INT,bal INT);

insert into savings_accounts
values(1,1,150000);
insert into savings_accounts
values(2,2,250000);
insert into savings_accounts
values(3,3,350000);
insert into savings_accounts
values(4,4,8500000);

select * from savings_accounts;

declare
AcNo savings_accounts.accno%type:=3;
custNo customers.id%type:=3;
Bal savings_accounts.bal%type;
custName customers.name%type;
begin
    select accno,bal,cusno into AcNo,Bal,custNo from SAVINGS_Accounts where accno= AcNo;
    select Name into custName from Customers where ID=custNo;
    if 0 < Bal and Bal <= 100000 then
```

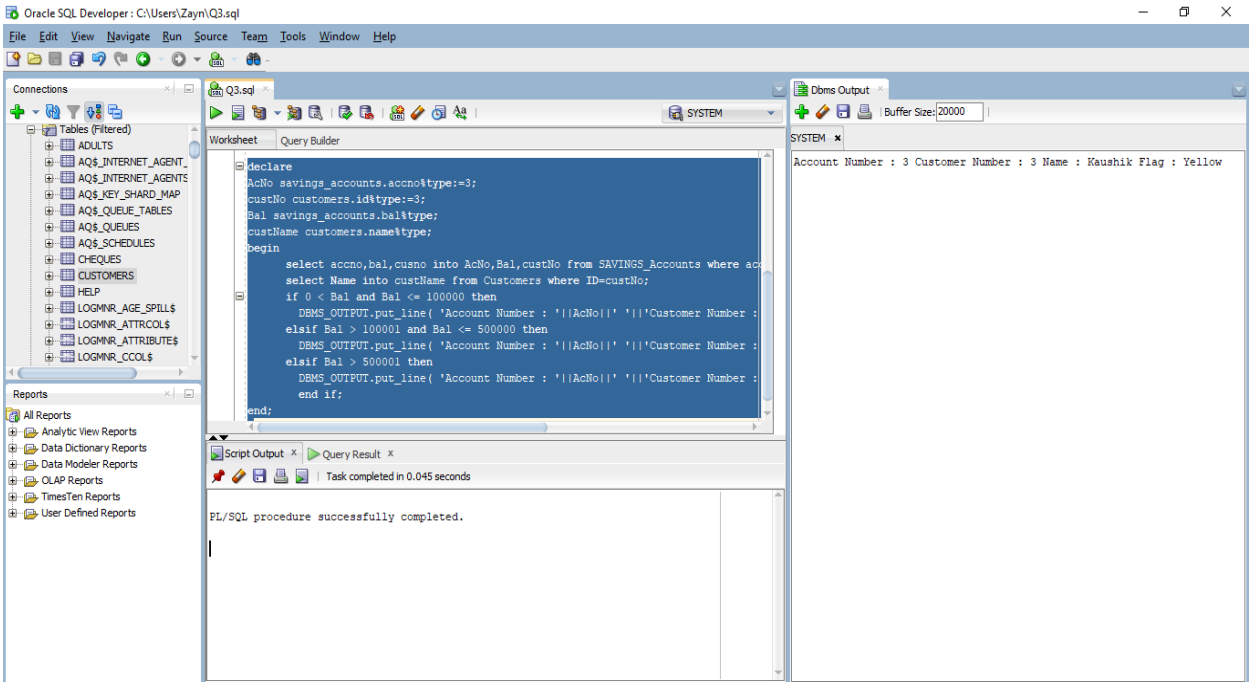
```
      DBMS_OUTPUT.put_line( 'Account Number : '||AcNo||' '||'Customer Number : '||custNo||' '||'Name : '||custName||' Flag : Red');

    elsif Bal > 100001 and Bal <= 500000 then

      DBMS_OUTPUT.put_line( 'Account Number : '||AcNo||' '||'Customer Number : '||custNo||' '||'Name : '||custName||' Flag : Yellow');

    elsif Bal > 500001 then

      DBMS_OUTPUT.put_line( 'Account Number : '||AcNo||' '||'Customer Number : '||custNo||' '||'Name : '||custName||' Flag : Green');

      end if;

end;

/
```

SCREENSHOTs




Oracle SQL Developer : C:\Users\Zayn\Q3.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Tables (Filtered)
ADULTS
AQ$_INTERNET_AGENT_PRIVS
AQ$_INTERNET_AGENTS
AQ$_KEY_SHARD_MAP
AQ$_QUEUE_TABLES
AQ$_QUEUES
AQ$_SCHEDULES
CHEQUES
CUSTOMERS
HELP
LOGMNR_AGE_SPILL$
LOGMNR_ATTRCOL$
LOGMNR_ATTRIBUTE$
LOGMNR_CCOL$
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Q3.sql
ADBMS_CW2.sql
SYSTEM
Worksheet
Query Builder
VALUES (1, 10, 2000.00);
insert into savings_accounts
values(1,1,150000);
insert into savings_accounts
values(2,2,250000);
insert into savings_accounts
values(3,3,350000);
create table SAVINGS_Accounts
(accno INT,cusno INT,bal INT);
select * from savings_accounts;
declare
AcNo savings_accounts.accno%type:=3;
custNo customers.id%type:=3;
Script Output
Query Result
Task completed in 0.038 seconds
1 row inserted.
1 row inserted.


Oracle SQL Developer : C:\Users\Zayn\Q3.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Tables (Filtered)
ADULTS
AQ$_INTERNET_AGENT_PRIVS
AQ$_INTERNET_AGENTS
AQ$_KEY_SHARD_MAP
AQ$_QUEUE_TABLES
AQ$_QUEUES
AQ$_SCHEDULES
CHEQUES
CUSTOMERS
HELP
LOGMNR_AGE_SPILL$
LOGMNR_ATTRCOL$
LOGMNR_ATTRIBUTE$
LOGMNR_CCOL$
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Q3.sql
ADBMS_CW2.sql
SYSTEM
Worksheet
Query Builder
VALUES (1, 10, 2000.00);
insert into savings_accounts
values(1,1,150000);
insert into savings_accounts
values(2,2,250000);
insert into savings_accounts
values(3,3,350000);
create table SAVINGS_Accounts
(accno INT,cusno INT,bal INT);
select * from savings_accounts;
declare
AcNo savings_accounts.accno%type:=3;
custNo customers.id%type:=3;
Dbms Output
Buffer Size: 20000
SYSTEM
Script Output
Query Result
SQL | All Rows Fetched: 3 in 0.002 seconds
ACCNO CUSNO BAL
1 1 1 150000
2 2 2 250000
3 3 3 350000


Oracle SQL Developer : C:\Users\Zayn\Q3.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Tables (Filtered)
ADULTS
AQ$_INTERNET_AGENT_
AQ$_INTERNET_AGENTS
AQ$_KEY_SHARD_MAP
AQ$_QUEUE_TABLES
AQ$_QUEUES
AQ$_SCHEDULES
CHEQUES
CUSTOMERS
HELP
LOGMNR_AGE_SPILL$
LOGMNR_ATTRCOL$
LOGMNR_ATTRIBUTE$
LOGMNR_CCOL$
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Q3.sql
SYSTEM
Worksheet
Query Builder
declare
AcNo savings_accounts.accno%type:=3;
custNo customers.id%type:=3;
Bal savings_accounts.bal%type;
custName customers.name%type;
begin
select accno,bal,cusno into AcNo,Bal,custNo from SAVINGS_Accounts where ac
select Name into custName from Customers where ID=custNo;
if 0 < Bal and Bal <= 100000 then
DBMS_OUTPUT.put_line('Account Number : '||AcNo||' '||'Customer Number :
elsif Bal > 100001 and Bal <= 500000 then
DBMS_OUTPUT.put_line('Account Number : '||AcNo||' '||'Customer Number :
elsif Bal > 500001 then
DBMS_OUTPUT.put_line('Account Number : '||AcNo||' '||'Customer Number :
end if;
end;
Script Output
Query Result
Task completed in 0.045 seconds
PL/SQL procedure successfully completed.
Dbms Output
Buffer Size: 20000
SYSTEM
Account Number : 3 Customer Number : 3 Name : Kaushik Flag : Yellow

4th Q "When retrieving data from the Customer table, we are applying the filter for the customer_id field. If customer_id is not available, then it will give an error. To overcome this issue, we can apply exception handling. Write a PL/SQL block for this scenario.**"**

Script

```
declare
c_id cheques.cusID%type:=10;
c_count cheques.cheque_count%type;
c_mo cheques.month%type;
begin
select cheque_count,month into c_mo,c_count from cheques where cusID=c_id;
dbms_output.put_line('Customer id : '||c_id||' '||'cheque_count : '||c_count||' '||'Month : '||c_mo);
exception
when no_data_found then
dbms_output.put_line ('Not Available!');
when others then
dbms_output.put_line('Error!');
end;
/
```

SCREENSHOTs