--LAB CA2

--Banking Application

--Relational Schemas

CREATE TABLE Customer

(

Sno int,

Cno varchar(10),

Customer\_name varchar(30),

Address varchar(100),

Phone\_Number number(10, 0)

);

DESC Customer;

CREATE TABLE Account

(

Ano varchar(10),

Amount number(10,2),

Account\_Type varchar(10)

);

DESC Account;

CREATE TABLE Cus\_Acc

(

Cno varchar(10),

Ano varchar(10)

);

DESC Cus\_Acc;

--Sequence

CREATE SEQUENCE SNO

INCREMENT BY 1

MINVALUE 1

MAXVALUE 35;

--View

CREATE VIEW Customer\_Account AS

SELECT CA.Cno, CA.Ano, A.amount

FROM Cus\_Acc CA, Account A

WHERE CA.Ano = A.Ano;

DESC Customer\_Account;

SELECT \* FROM Customer\_Account;

--Add PK And FK

ALTER TABLE Customer ADD CONSTRAINT pk\_Customer PRIMARY KEY(Cno);

ALTER TABLE Account ADD CONSTRAINT pk\_Account PRIMARY KEY(Ano);

ALTER TABLE Cus\_Acc

ADD CONSTRAINT fk\_Cus\_Acc1

FOREIGN KEY(Cno)

REFERENCES Customer(Cno);

ALTER TABLE Cus\_Acc

ADD CONSTRAINT fk\_Cus\_Acc2

FOREIGN KEY(Ano)

REFERENCES Account(Ano);

ALTER TABLE Cus\_Acc

ADD CONSTRAINT pk\_Cus\_Acc

PRIMARY KEY(Cno, Ano);

DESC Customer;

DESC Account;

DESC Cus\_Acc;

--Penalty Schema

CREATE TABLE Penalty

(

Sno int,

Cno varchar(10),

Ano varchar(10),

Amount int

);

DESC Penalty;

--Inserting Records

INSERT INTO Customer VALUES(SNO.nextval, 'C1', 'AB', 'ABCDE', 1234567897);

INSERT INTO Customer VALUES(SNO.nextval, 'C2', 'ABC', 'ABCD', 1234567890);

INSERT INTO Customer VALUES(SNO.nextval, 'C3', 'DEF', 'ABCE', 1234567899);

INSERT INTO Customer VALUES(SNO.nextval, 'C4', 'GHI', 'ABCF', 1234567898);

INSERT INTO Customer VALUES(SNO.nextval, 'C5', 'JKL', 'ABCG', 1234567896);

SELECT \* FROM Customer;

INSERT INTO Account VALUES('A1', 5000, 'AB');

INSERT INTO Account VALUES('A2', 1000, 'CD');

INSERT INTO Account VALUES('A3', 500, 'EF');

INSERT INTO Account VALUES('A4', 350, 'GH');

INSERT INTO Account VALUES('A5', 300, 'IJ');

SELECT \* FROM Account;

INSERT INTO Cus\_Acc VALUES ('C1', 'A1');

INSERT INTO Cus\_Acc VALUES ('C2', 'A2');

INSERT INTO Cus\_Acc VALUES ('C3', 'A1');

INSERT INTO Cus\_Acc VALUES ('C4', 'A3');

INSERT INTO Cus\_Acc VALUES ('C1', 'A5');

INSERT INTO Cus\_Acc VALUES ('C5', 'A4');

SELECT \* FROM Cus\_Acc;

--PL/SQL

--Function

CREATE OR REPLACE FUNCTION CustCheck(Cno IN varchar, Ano IN varchar, Withdrawal\_Amt IN number, Balance IN number)

RETURN varchar

AS

PenaltyFee number(10, 2):=50;

BEGIN

IF(Cno IS NULL) THEN

RETURN 'Customer Number is Not Found';

ELSIF(Ano IS NULL) THEN

RETURN 'Account Number is Not Found';

ELSE

IF Balance < Withdrawal\_Amt THEN

RETURN 'Withdrawal Amount is larger than the current Amount';

ELSE

IF Balance - Withdrawal\_Amt < 500 THEN

UPDATE Customer\_Account C

SET C.Amount = Balance - Withdrawal\_Amt - PenaltyFee

WHERE C.Cno = Cno AND C.Ano = Ano;

RETURN 'Successfully completed with Penalty';

ELSE

UPDATE Customer\_Account C

SET C.Amount = C.Amount - Withdrawal\_Amt

WHERE C.Cno = Cno AND C.Ano = Ano;

RETURN 'Successfully completed';

END IF;

END IF;

END IF;

END;

--Select Statements

SELECT \* FROM Customer;

SELECT \* FROM Account;

SELECT \* FROM Cus\_Acc;

SELECT \* FROM Customer\_Account;

SELECT \* FROM Penalty;

--Trigger

CREATE OR REPLACE TRIGGER PenaltyTrig

INSTEAD OF UPDATE

ON Customer\_Account

FOR EACH ROW

DECLARE

Cno varchar(20);

Ano varchar(20);

Amt number(10,2);

BEGIN

SELECT Amount A INTO Amt FROM Customer\_Account C

WHERE :old.Ano = C.Ano AND :old.Cno = C.Cno;

IF Amt < 500 THEN

SELECT C.Cno INTO Cno

FROM Customer\_Account C

WHERE C.Cno = :old.Cno AND C.Ano = :old.Ano;

SELECT C.Ano INTO Ano

FROM Customer\_Account C

WHERE C.Cno = :old.Cno AND C.Ano = :old.Ano;

INSERT INTO Penalty VALUES(SNO.nextval, Cno, Ano, 50);

END IF;

END;

--PL/SQL

DECLARE

Cnum varchar(20):= 'C2';

Anum varchar(20):= 'A2';

Cno varchar(20);

Ano varchar(20);

Withdrawal\_Amt number(10,2);

Balance number(10, 2);

BEGIN

--Case1

SELECT C.Cno INTO Cno

FROM Customer\_Account C

WHERE C.Cno = Cnum AND C.Ano = Anum;

SELECT C.Ano INTO Ano

FROM Customer\_Account C

WHERE C.Cno = Cnum AND C.Ano = Anum;

SELECT C.Amount INTO Balance

FROM Customer\_Account C

WHERE C.Cno = Cnum AND C.Ano = Anum;

Withdrawal\_Amt := 700;

dbms\_output.put\_line(CustCheck(Cno, Ano, Withdrawal\_Amt, Balance));

END;