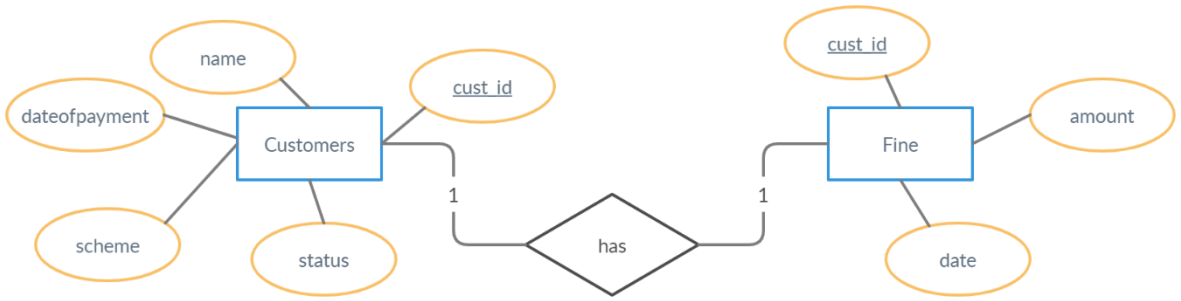
## Roll No-:- 31384

# ER Diagram :-




# Code:-

/* Assignment 5 */

create database plsql;

use plsql;

show tables;

create table Customers (cust_id int,CustName varchar(230),Dateofpayment date,Nameofscheme varchar(230),status varchar(230), Primary key(cust_id) );

create table fine(cust_id int,Dateof date ,amt int , Foreign key(cust_id) REFERENCES Customers(cust_id));

insert into Customers (cust_id,CustName,Dateofpayment,Nameofscheme,status) values(1,'Swapnil Ghule',STR_TO_DATE("August 10 2020", "%M %d %Y"),'ABC','not return') ;

insert into Customers (cust_id,CustName,Dateofpayment,Nameofscheme,status) values(2,'Ganesh Shinde',STR_TO_DATE("August 29 2020", "%M %d %Y"),'PQR','not return') ;

insert into Customers (cust_id,CustName,Dateofpayment,Nameofscheme,status) values(3,'Sourabh Chile',STR_TO_DATE("may 09 2020", "%M %d %Y"),'XYZ','not return') ;

insert into Customers (cust_id,CustName,Dateofpayment,Nameofscheme,status) values(4,'Atul Kulkarni',STR_TO_DATE("August 30 2020", "%M %d %Y"),'KLM','not return') ;

insert into Customers (cust_id,CustName,Dateofpayment,Nameofscheme,status) values(5,'Rushikesh Wanjare',STR_TO_DATE("September 2 2020", "%M %d %Y"),'LOI','not return') ;

select * from Customers;

SET SQL_SAFE_UPDATES = 0;

update Customers set status=" - ";

```
call calcFine(1,"Swapnil Ghule");
call calcFine(2,"Ganesh Shinde");
call calcFine(3,"Sourabh Chile");
call calcFine(4,"Atul Kulkarni");
```

```
call calcFine(5,"Rushikesh Wane");



select * from fine;


truncate table fine;


DELIMITER //
   create procedure calcFine(id int, nameC varchar(15))
     begin
            #Declare variable for storing days
            declare days integer;


            #Exception Handling
            declare continue handler for not found
       begin
                select 'NOT FOUND';
       end;


            # Select difference betwen two dates
            select DATEDIFF(curdate(), Dateofpayment) into days from Customers where cust_id = id AND
Custname=nameC;


            #condition 1
            if( days>15 and days<30 )
             then
               insert into fine values(id, curdate(), (days*5));
            end if;


            #condition2
            if( days>30 )
        then
          insert into fine values(id, curdate(), (days*50));
       end if;
```

```
    update Customers set status='return' where cust_id = id;
   end;
//

DELIMITER ;

drop procedure calcFine;
```

# Output :-

Customer Table before the calling the procedure . all are marked as not return fine status

Result Grid | Filter Rows: | Edit: | Export/Import:

| cust_id | CustName | Dateofpayment | Nameofscheme | status |
|---|---|---|---|---|
| 1 | Swapnil Ghule | 2020-08-10 | ABC | not return |
| 2 | Ganesh Shinde | 2020-08-29 | PQR | not return |
| 3 | Sourabh Chile | 2020-05-09 | XYZ | not return |
| 4 | Atul Kulkarni | 2020-08-30 | KLM | not return |
| 5 | Rushikesh Wanjare | 2020-09-02 | LOI | not return |
| NULL | NULL | NULL | NULL | NULL |

Fine table after calling the procedures for all five customers. since there are only 2 customers whose date is overdue so fine is applicable only to them.

Result Grid | Filter Rows:

| cust_id | Dateof | amt |
|---|---|---|
| 1 | 2020-09-07 | 140 |
| 3 | 2020-09-07 | 6050 |

Customer table after procedure called so all status is marked as return.

Result Grid | Filter Rows: | Edit: | Export/Import:

| cust_id | CustName | Dateofpayment | Nameofscheme | status |
|---|---|---|---|---|
| 1 | Swapnil Ghule | 2020-08-10 | ABC | return |
| 2 | Ganesh Shinde | 2020-08-29 | PQR | return |
| 3 | Sourabh Chile | 2020-05-09 | XYZ | return |
| 4 | Atul Kulkarni | 2020-08-30 | KLM | return |
| 5 | Rushikesh Wanjare | 2020-09-02 | LOI | return |
| NULL | NULL | NULL | NULL | NULL |

# Assignment No. 5

Title: Write PL/SQL block of code for given requirements.

Problem statements:- Write PL/SQL block of code for following requirements.

Create the two table Customer & fine
Calculate fine using PL-SQL code.

objective: • To understand the control structure.

• To understand exception handling in PL/SQL

Sw package: SQL package.

Learning objective

• To understand the control structure.
• To write error free sql-code of PL.

1.) Accept cust-id & name of scheme from user
2.) check the no. of days. if days betn 15 to 30 then fine amt Rs 5 per day.
3.) If no. of days > 30 then will be Rs 50 per day.
4) after payment, change the status.

Theory :

- PL SQL Stands for procedural language / Structured Query language.
- PL SQL offers a set of procedural commands

Blocks of PL-SQL :-

It is defined by DECLARE, BEGIN, EXCEPTION and END

i) Declarative : statement that declares variables, constants & other code elements

ii) Executable : statement that are run when the block is executed.

iii) Exception handling : a specially structured section you can use to catch or trap, any exception that are raised when executable section runs

Hello world : Program example

```
BEGIN
    DBMS.OUTPUT.put-line ("helloworld")
END
```

Exception handling :-

PL-SQL provides feature to handle the exception which occurs in PL-SQL block known as exception handling, using Exception handling we can test the code & avoid it from existing abruptly.

Structure of Exception

```
DECLARE
    declaration section
BEGIN
    Exception section
Exception
    when ex_name then
        error handling statement

END;
```

When an exception raised, Oracle searches for an appropriate exception handler the exception section.

For e.g.
if ex. name raised the error is handled accordingly.

Types of exception :

i) Named system exception :
Automatically raised by Oracle, when program violates the rule.

ii) Unnamed system exception :-
These exception do not occur frequently
These exception have code & message

iii) User-defined exception :-

Apart for system exceptions based on business rule. These are known as user-defined exceptions.

Conclusion :
Hence we implemented PL-SQL with exception handling.