**Assignments 5 Cursor**

**SetA**

**1.Movie - Actor Database: Movie (m\_name char (25), release\_year integer, budget money) Actor (a\_name varchar (20), role char (20), charges money, a\_address varchar (20)) Producer (producer\_id integer, p\_name char (30), p\_address varchar (20)) Each actor has acted in one or more movies. Each producer has produced many movies and each movie can be produced by more than one producer. Each movie has one or more actors acting in it, in different roles. Constraints: Primary Key, role should be ‘Main’,’Supportive’,’Villan’,’Comedy’ p\_name should not be null. budget,charges > 0**

create table actor(a\_name varchar(20) prima

ry key, role char(10),charges money,addr varchar(2

0));

CREATE TABLE

mydb=# create table movie(m\_name varchar(25)primar

y key,release\_year int,budget money,a\_name varchar

(20)references actor);

CREATE TABLE

mydb=# create table producer(p\_id int primary key,

p\_name char(30),p\_add varchar(20));

CREATE TABLE

mydb=# create table movie\_producer(m\_name varchar(

25)references movie,p\_id int references producer);

CREATE TABLE

mydb=# insert into actor values('amitabh','main',5

0000,'mumbai');

INSERT 0 1

mydb=# insert into actor values('shreyas','comedy'

,70000,'pune');

INSERT 0 1

mydb=# insert into actor values('jacky','vilan',40

000,'pune');

INSERT 0 1

mydb=# insert into movie values('bagwan',1995,1000

00,'amitabh');

INSERT 0 1

mydb=# insert into movie values('golmal',2001,1050

00,'shreyas');

INSERT 0 1

mydb=# insert into movie values('dhamal',2005,1000

0,'jacky');

INSERT 0 1

mydb=# insert into producer values(11,'ravi mahaja

n','mumbai');

INSERT 0 1

mydb=# insert into producer values(12,'karan','pun

e');

INSERT 0 1

mydb=# insert into producer values(13,'kapil','pun

e');

INSERT 0 1

insert into movie\_producer values('bagwan',11);

INSERT 0 1

mydb=# insert into movie\_producer values('golmal',12);

INSERT 0 1

mydb=# insert into movie\_producer values('dhamal',11);

INSERT 0 1

mydb=# select \* from actor;

a\_name | role | charges | addr

---------+------------+------------+--------

amitabh | main | $50,000.00 | mumbai

shreyas | comedy | $70,000.00 | pune

jacky | vilan | $40,000.00 | pune

(3 rows)

mydb=# select \* from movie;

m\_name | release\_year | budget | a\_name

--------+--------------+-------------+---------

bagwan | 1995 | $100,000.00 | amitabh

golmal | 2001 | $105,000.00 | shreyas

dhamal | 2005 | $10,000.00 | jacky

(3 rows)

mydb=# select \* from producer;

p\_id | p\_name | p\_add

------+--------------------------------+--------

11 | ravi mahajan | mumbai

12 | karan | pune

13 | kapil | pune

(3 rows)

mydb=# select \* from movie\_producer;

m\_name | p\_id

--------+------

bagwan | 11

golmal | 12

dhamal | 11

(3 rows)

**1.Write a cursor to pass a\_name as a parameter to a function and return total number of movies in which given actor is acting.**

mydb=# CREATE OR REPLACE FUNCTION demo(v\_name VARCHAR)

mydb-# RETURNS VOID AS $$

mydb$# DECLARE

mydb$# actor\_cursor CURSOR FOR

mydb$# SELECT m.m\_name FROM movie m, actor a

mydb$# WHERE a.a\_name = m.a\_name AND a.a\_name = v\_name;

mydb$# rec RECORD;

mydb$# BEGIN

mydb$# OPEN actor\_cursor;

mydb$# LOOP

mydb$# FETCH actor\_cursor INTO rec;

mydb$# EXIT WHEN NOT FOUND;

mydb$# RAISE NOTICE '%', rec.m\_name;

mydb$# END LOOP;

mydb$# CLOSE actor\_cursor;

mydb$# END;

mydb$# $$ LANGUAGE plpgsql;

CREATE FUNCTION

mydb=#

mydb=#

mydb=# select demo('amitabh');

NOTICE: bagwan

demo

------

(1 row)

**2.Write a stored function using cursor to display role wise the names of actors.**

mydb=# CREATE OR REPLACE FUNCTION demo1()

mydb-# RETURNS VOID AS $$

mydb$# DECLARE

mydb$# role\_cursor CURSOR FOR

mydb$# SELECT role, a\_name FROM actor ORDER BY role;

mydb$# rec RECORD;

mydb$# BEGIN

mydb$# OPEN role\_cursor;

mydb$# LOOP

mydb$# FETCH role\_cursor INTO rec;

mydb$# EXIT WHEN NOT FOUND;

mydb$# RAISE NOTICE '%', rec;

mydb$# END LOOP;

mydb$# CLOSE role\_cursor;

mydb$# END;

mydb$# $$ LANGUAGE plpgsql;

CREATE FUNCTION

mydb=# select demo1();

NOTICE: ("comedy ",shreyas)

NOTICE: ("main ",amitabh)

NOTICE: ("vilan ",jacky)

demo1

(1 row)

**3) Write a cursor to display producer name that produces more than 2 movies in which ‘Amitabh’ is acted.**

mydb=# CREATE OR REPLACE FUNCTION demo(v\_name VARCHAR(20))

mydb-# RETURNS VOID AS $$

mydb$# DECLARE

mydb$# pro\_cursor CURSOR FOR

mydb$# SELECT p.p\_id, p.p\_name

mydb$# FROM producer p, movie m, movie\_producer mp

mydb$# WHERE m.m\_name = mp.m\_name

mydb$# AND p.p\_id = mp.p\_id

mydb$# AND m.a\_name = v\_name

mydb$# GROUP BY p.p\_id,p.p\_name

mydb$# HAVING COUNT(mp.m\_name) > 0;

mydb$# rec RECORD;

mydb$# BEGIN

mydb$# OPEN pro\_cursor;

mydb$# LOOP

mydb$# FETCH pro\_cursor INTO rec;

mydb$# EXIT WHEN NOT FOUND;

mydb$# RAISE NOTICE '% %', rec.p\_id, rec.p\_name;

mydb$# END LOOP;

mydb$# CLOSE pro\_cursor;

mydb$# END;

mydb$# $$ LANGUAGE plpgsql;

CREATE FUNCTION

mydb=# select demo('amitabh');

NOTICE: 11 ravi mahajan

demo

------

(1 row)

**2.Consider the following Entities and their Relationships Railway -Reservation Database:**

**Train (tno int, tname varchar (20), depart\_time time, arrival\_time time, source\_stn char (10),**

**dest\_stn char (10), no\_of\_res\_bogies int ,bogie\_capacity int)**

**Passenger (passenger\_id int, passenger\_name varchar (20), address varchar (30), age int, gender**

**char)**

**Relationship between Train and Passenger is many to many with descriptive attribute ticket.**

**Ticket (train\_no int, passenger\_id int, ticket\_no int,bogie\_no int, no\_of\_berths int, tdate date, ticket\_amt**

**decimal (7,2),status char)**

**Constraints: Primary Key,**

**Status of a berth can be 'W' (waiting) or 'C' (confirmed).**

create table Train(t\_no int primary key,t\_name varchar(20),dept\_time time,arrival\_time time,source\_stn char(10),dest\_stn char(10),no\_of\_res\_bogies int,bogie\_capacity int);

CREATE TABLE

mydb=# create table passenger(passenger\_id int primary key,passenger\_name varchar(20),address varchar(30),age int,gender char);

CREATE TABLE

create table Ticket(t\_no int references Train,passenger\_id int references passenger,ticket\_no int primary key,bogie\_no int,no\_of\_berths int,t\_date date,ticket\_amt Decimal(7,2),status char(1)check(status in('w','c')));

CREATE TABLE

insert into Train values(1,'Express','10:00:00','18:00:00','pune','delhi',10,50);

INSERT 0 1

mydb=# insert into Train values(2,'Rajdhani','13:30:00','21:00:00','goa','pune',8,70);

INSERT 0 1

mydb=# insert into Train values(3,'vande mataram','15:45:00','20:00:00','banglore','mumbai',9,60);

INSERT 0 1

mydb=# insert into passenger values(101,'Manu','pune',30,'F');

INSERT 0 1

mydb=# insert into passenger values(102,'Ram','goa',40,'M');

INSERT 0 1

mydb=# insert into passenger values(103,'nilam','delhi',32,'F');

INSERT 0 1

mydb=# insert into Ticket values(1,101,1001,1,2,'2024-11-16',1000.00,'c');

INSERT 0 1

mydb=# insert into Ticket values(2,102,1002,3,4,'2024-11-17',1500.00,'w');

INSERT 0 1

mydb=# insert into Ticket values(3,103,1003,3,5,'2024-11-18',2500.00,'w');

INSERT 0 1

mydb=# select \*from Train;

t\_no | t\_name | dept\_time | arrival\_time | source\_stn | dest\_stn | no\_of\_res\_bogies | bogie\_capacity

------+---------------+-----------+--------------+------------+------------+------------------+----------------

1 | Express | 10:00:00 | 18:00:00 | pune | delhi | 10 | 50

2 | Rajdhani | 13:30:00 | 21:00:00 | goa | pune | 8 | 70

3 | vande mataram | 15:45:00 | 20:00:00 | banglore | mumbai | 9 | 60

(3 rows)

mydb=# select \*from passenger;

passenger\_id | passenger\_name | address | age | gender

--------------+----------------+---------+-----+--------

101 | Manu | pune | 30 | F

102 | Ram | goa | 40 | M

103 | nilam | delhi | 32 | F

(3 rows)

mydb=# select \*from Ticket;

t\_no | passenger\_id | ticket\_no | bogie\_no | no\_of\_berths | t\_date | ticket\_amt | status

------+--------------+-----------+----------+--------------+------------+------------+--------

1 | 101 | 1001 | 1 | 2 | 2024-11-16 | 1000.00 | c

2 | 102 | 1002 | 3 | 4 | 2024-11-17 | 1500.00 | w

3 | 103 | 1003 | 3 | 5 | 2024-11-18 | 2500.00 | w

(3 rows)

**1) Write a stored function using cursor to accept date and passenger name and display no. of berths**

mydb=# Create or Replace function berths\_ticket\_amount(p\_date date,p\_name varchar)

mydb-# Returns void as $$

mydb$# Declare

mydb$# ticket\_cursor Cursor for

mydb$# select t.ticket\_no,t.no\_of\_berths,t.ticket\_amt from ticket t join passenger p on p.passenger\_id=t.passenger\_id where p.passenger\_name=p\_name and t\_date=p\_date;

mydb$# rec Record;

mydb$# total\_berths int:=0;

mydb$# total\_amount decimal(7,2):= 0.00;

mydb$# Begin

mydb$# open ticket\_cursor;

mydb$# Loop

mydb$# Fetch ticket\_cursor into rec;

mydb$# Exit when not found; --exit the loop if no more rows

mydb$# total\_berths:= total\_berths+rec.no\_of\_berths;

mydb$# total\_amount:=total\_amount+rec.ticket\_amt;

mydb$# end loop;

mydb$# close ticket\_cursor;

mydb$# Raise Notice'Total berths reserved:%,Total ticket amount:%',total\_berths,total\_amount;

mydb$# End;

mydb$# $$ Language plpgsql;

CREATE FUNCTION

mydb=#

mydb=# select berths\_ticket\_amount('2024-11-16','Manu');

NOTICE: Total berths reserved:2,Total ticket amount:1000.00

berths\_ticket\_amount

----------------------

(1 row)

**2) Write a stored function using cursors to find the confirmed bookings of all the trains on a particular**

create or Replace function confirmed\_booking(p\_date Date)

mydb-# Returns void as $$

mydb$# Declare

mydb$# booking\_cursor Cursor for

mydb$# select t.t\_no,t.t\_name,t.dept\_time,t.arrival\_time,t.source\_stn,t.dest\_stn,t.no\_of\_res\_bogies,t.bogie\_capacity,tk.ticket\_no,tk.bogie\_no,tk.no\_of\_berths,tk.ticket\_amt from ticket tk join train t on tk.ticket\_no=t.t\_no where tk.t\_date=p\_date and tk.status='c'; --filter for confirmed ticketsr

mydb$# rec Record;

mydb$# Begin

mydb$# open booking\_cursor;

mydb$# Loop

mydb$# Fetch booking\_cursor into rec;

mydb$# Exit when not found; --exit the loop if no more rows

mydb$# Raise Notice'Train:%,Departure:%,Arrival:%,Source:%,Destination:%,Bogie no:%,Berths:%,Ticket amount:%',rec.t\_name,rec.dept\_time,rec.arrival\_time,rec.source\_stn,rec.dest\_stn,rec.bogie\_no,rec.no\_of\_berths,rec.ticket\_amt;

mydb$# end loop;

mydb$# close booking\_cursor;

mydb$# End;

mydb$# $$ Language plpgsql;

CREATE FUNCTION

mydb=#

mydb=# select confirmed\_booking('2024-11-17');

confirmed\_booking

-------------------

(1 row)

**3) Write a stored function using cursors to accept a date and find the total number of berths**

Create or Replace function berths\_reserved(p\_date Date)

mydb-# Returns void as $$

mydb$# Declare

mydb$# berths\_cursor Cursor for

mydb$# select no\_of\_berths from ticket where t\_date=p\_date;

mydb$# rec Record;

mydb$# total\_berths int:=0;

mydb$#

mydb$# Begin

mydb$# open berths\_cursor;

mydb$# Loop

mydb$# Fetch berths\_cursor into rec;

mydb$# Exit when not found; --exit the loop if no more rows

mydb$# total\_berths:= total\_berths+rec.no\_of\_berths;

mydb$#

mydb$# end loop;

mydb$# close berths\_cursor;

mydb$# Raise Notice'Total Number of berths reserved:%',total\_berths;

mydb$# End;

mydb$# $$ Language plpgsql;

CREATE FUNCTION

mydb=#

mydb=# select berths\_reserved('2024-11-18');

NOTICE: Total Number of berths reserved:5

berths\_reserved

-----------------

(1 rows)

**SET B**

**1.Bus Driver Database BUS (bus\_no int , capacity int , depot\_name varchar(20)) ROUTE (route\_no int, source char(20), destination char(20),no\_of\_stations int) DRIVER (driver\_no int , driver\_name char(20), license\_no int, address char(20), d\_age int, salary float) The relationships are as follows: BUS\_ROUTE: M-1 BUS\_DRIVER: M-M with descriptive attributes Date of duty allotted and Shift – it can be 1 (Morning) or 2 ( Evening ). Constraints: 1. License\_no is unique. 2. Bus capacity is not null**

mydb=# create table route(route\_no int primary key,source char(20),destination char,no\_of\_stations int);

CREATE TABLE

mydb=# create table Bus(bus\_no int primary key,capacity int not null,depot\_name varchar(20),route\_no int referen

ces route);

CREATE TABLE

mydb=# create table Driver(driver\_no int primary key,driver\_name char(20),license\_no int,address char(20),d\_age

int,salary float);

CREATE TABLE

mydb=# create table Bus\_Driver(bus\_no int references Bus,driver\_no int references Driver,date\_of\_duty Date,shift

int check(shift IN(1,2)));

CREATE TABLE

mydb=# insert into route values(101,'X','Y',6);

INSERT 0 1

mydb=# insert into route values(102,'O','P',8);

INSERT 0 1

mydb=# insert into route values(103,'S','T',5);

INSERT 0 1

mydb=# insert into Bus values(1,35,'depotA',101);

INSERT 0 1

mydb=# insert into Bus values(2,45,'depotB',102);

INSERT 0 1

mydb=# insert into Bus values(3,50,'depotC',103);

INSERT 0 1

mydb=# insert into Driver values(11,'xyz',1234,'pune',35,40000);

INSERT 0 1

mydb=# insert into Driver values(12,'abc',1235,'nashik',28,50000);

INSERT 0 1

mydb=# insert into Driver values(13,'pqr',1236,'mumbai',45,60000);

INSERT 0 1

mydb=# insert into Bus\_Driver values(1,11,'2023-11-16',1);

INSERT 0 1

mydb=# insert into Bus\_Driver values(2,12,'2023-11-18',2);

INSERT 0 1

mydb=# insert into Bus\_Driver values(3,13,'2023-11-17',1);

INSERT 0 1

mydb=# select\* from route;

route\_no | source | destination | no\_of\_stations

----------+----------------------+-------------+----------------

101 | X | Y | 6

102 | O | P | 8

103 | S | T | 5

(3 rows)

mydb=# select\* from Bus;

bus\_no | capacity | depot\_name | route\_no

--------+----------+------------+----------

1 | 35 | depotA | 101

2 | 45 | depotB | 102

3 | 50 | depotC | 103

(3 rows)

mydb=# select\* from Driver;

driver\_no | driver\_name | license\_no | address | d\_age | salary

-----------+----------------------+------------+----------------------+-------+--------

11 | xyz | 1234 | pune | 35 | 40000

12 | abc | 1235 | nashik | 28 | 50000

13 | pqr | 1236 | mumbai | 45 | 60000

(3 rows)

mydb=# select\* from Bus\_Driver;

bus\_no | driver\_no | date\_of\_duty | shift

--------+-----------+--------------+-------

1 | 11 | 2023-11-16 | 1

2 | 12 | 2023-11-18 | 2

3 | 13 | 2023-11-17 | 1

(3 rows)

**1) Write a stored function using cursor to display the details of a driver,(Accept driver name as input prameter).**

create or replace function driver\_detail(d\_name varchar(20)) returns void as $$

mydb$# declare

mydb$# d\_cursor cursor for

mydb$# select driver\_no,driver\_name,license\_no,address,d\_age,salary

mydb$# from Driver

mydb$# where driver\_name = d\_name;

mydb$# rec record;

mydb$# begin

mydb$# open d\_cursor;

mydb$# loop

mydb$# fetch d\_cursor into rec;

mydb$# exit when not found;

mydb$# raise notice'% % % % % %',rec.driver\_no, rec.driver\_name, rec.license\_no, rec.address,rec.d\_age,rec.salar

y;

mydb$# END loop;

mydb$# close d\_cursor;

mydb$# END;

mydb$# $$ language plpgsql;

CREATE FUNCTION

mydb=#

mydb=# Select driver\_detail('xyz');

NOTICE: 11 xyz 1234 pune 35 40000

driver\_detail

---------------

(1 row)

**2) Write a cursor to display the details of the buses that run on route\_no = 1 and route\_no= 2**

create or replace function route\_detail() returns void as $$

mydb$# declare

mydb$# R\_cursor cursor for

mydb$# SELECT b.bus\_no, b.capacity, b.depot\_name, r.route\_no, r.source, r.destination, r.no\_of\_stations

mydb$# FROM bus b, route r where b.route\_no = r.route\_no

mydb$# and r.route\_no = 101 ;

mydb$# rec record;

mydb$# begin

mydb$# open R\_cursor;

mydb$# loop

mydb$# fetch R\_cursor into rec;

mydb$# exit when not found;

mydb$# raise notice'% % % % % % %', rec.bus\_no, rec.capacity, rec.depot\_name, rec.route\_no, rec.source, rec.dest

ination, rec.no\_of\_stations

mydb$# ;

mydb$# END loop;

mydb$# close R\_cursor;

mydb$# END;

mydb$# $$ language plpgsql;

CREATE FUNCTION

mydb=#

mydb=# Select route\_detail();

NOTICE: 1 35 depotA 101 X Y 6

route\_detail

--------------

(1 row)

**2. Consider the following entities and their relationships.**

**Library(Lno, Lname, Location, Librarian, no\_of\_books)**

**Book(Bid, Bname, Author\_Name, Price, publication)**

**Relation between Library and Book is one to many.**

**Constraint: Primary key, Price should not be null. Create a RDB in 3NF and write PL/SQL**

**procedure for the following:**

create table library(l\_id int primary key,l\_name varchar(20),location varchar(20),librarian varchar(20)

,no\_of\_books int);

CREATE TABLE

mydb=# create table Book(b\_id int primary key,b\_name varchar(20),author\_name varchar(20),price int,publication

varchar(20),l\_id int references library);

CREATE TABLE

mydb=# insert into library values(1,'lib A','pune','Mayuri',20);

INSERT 0 1

mydb=# insert into library values(2,'lib B','nashik','Manu',30);

INSERT 0 1

mydb=# insert into library values(3,'lib C','moshi','Parth',90);

INSERT 0 1

mydb=# insert into Book values(11,'sambhaji','xyz',250,'nirali',1);

INSERT 0 1

mydb=# insert into Book values(12,'mrugjal','abc',300,'nvneet',2);

INSERT 0 1

mydb=# insert into Book values(13,'success','mno',200,'target',3);

INSERT 0 1

mydb=# select \*from library;

l\_id | l\_name | location | librarian | no\_of\_books

------+--------+----------+-----------+-------------

1 | lib A | pune | Mayuri | 20

2 | lib B | nashik | Manu | 30

3 | lib C | moshi | Parth | 90

(3 rows)

mydb=# select \*from Book;

b\_id | b\_name | author\_name | price | publication | l\_id

------+----------+-------------+-------+-------------+------

11 | sambhaji | xyz | 250 | nirali | 1

12 | mrugjal | abc | 300 | nvneet | 2

13 | success | mno | 200 | target | 3

(3 rows)

**1) Write a cursor which will display library wise book details. (Use Parameterized Cursor)**

CREATE OR REPLACE FUNCTION lib(lib\_no int)

mydb-# RETURNS VOID AS $$

mydb$# DECLARE

mydb$# Lib\_cursor cursor for

mydb$# SELECT b.b\_id, b.b\_name

mydb$# FROM book b, library l

mydb$# WHERE b.l\_id = l.l\_id AND l.l\_id = lib\_no

mydb$# ORDER BY b.b\_name;

mydb$# rec record;

mydb$# BEGIN

mydb$# Open Lib\_cursor;

mydb$# loop

mydb$# fetch Lib\_cursor into rec;

mydb$# exit when not found;

mydb$# RAISE NOTICE '% %', rec.b\_id, rec.b\_name;

mydb$# END LOOP;

mydb$# close Lib\_cursor;

mydb$# END;

mydb$# $$ LANGUAGE plpgsql;

CREATE FUNCTION

mydb=# select lib(3);

NOTICE: 13 success

lib

-----

(1 row)

**2) Write a cursor which will display publication wise book details.**

CREATE OR REPLACE FUNCTION lab(publicationn varchar(20))

mydb-# RETURNS VOID AS $$

mydb$# DECLARE

mydb$# pub\_cursor cursor for

mydb$# SELECT b.b\_id, b.b\_name

mydb$# FROM book b, library l

mydb$# WHERE b.l\_id = l.l\_id AND publication = publicationn

mydb$# ORDER BY b.publication;

mydb$# rec RECORD;

mydb$# BEGIN

mydb$# Open pub\_cursor;

mydb$# LOOP

mydb$# Fetch pub\_cursor into rec;

mydb$# Exit when not found;

mydb$# RAISE NOTICE '% %', rec.b\_id, rec.b\_name;

mydb$# END LOOP;

mydb$# close pub\_cursor;

mydb$# END;

mydb$# $$ LANGUAGE plpgsql;

CREATE FUNCTION

mydb=# select lab('nirali');

NOTICE: 11 sambhaji

lab

-----

(1 row)

**SET C**

**1. Consider the following entities and their relationships.**

**Employee (emp\_id, emp\_name, address)**

**Investment (inv\_no, inv\_name, inv\_date, inv\_amount)**

**Relation between Employee and Investment is One to Many.**

**Constraint: Primary key, inv\_amount should be > 0.**

**Create a RDB in 3NF and write PL/SQL procedure for the following for the following:**

create table Emp(e\_id int primary key, e\_name varchar(20),addr varchar(25));

CREATE TABLE

mydb=# create table inv(inv\_no int primary key, inv\_name varchar(20),inv\_date date,inv\_amt int check(inv\_amt>0

),e\_id int references Emp);

CREATE TABLE

mydb=# insert into Emp values(1,'Nilam','pune');

INSERT 0 1

mydb=# insert into Emp values(2,'Sham','Bhosari');

INSERT 0 1

mydb=# insert into Emp values(3,'Ram','nashik');

INSERT 0 1

mydb=# insert into inv values(101,'mutual fund','2024-10-12',50000,1);

INSERT 0 1

mydb=# insert into inv values(102,'sip','2024-11-18',80000,2);

INSERT 0 1

mydb=# insert into inv values(103,'Lic','2024-11-16',75000,3);

INSERT 0 1

mydb=# select \* from Emp;

e\_id | e\_name | addr

------+--------+---------

1 | Nilam | pune

2 | Sham | Bhosari

3 | Ram | nashik

(3 rows)

mydb=# select \* from inv;

inv\_no | inv\_name | inv\_date | inv\_amt | e\_id

--------+-------------+------------+---------+------

101 | mutual fund | 2024-10-12 | 50000 | 1

102 | sip | 2024-11-18 | 80000 | 2

103 | Lic | 2024-11-16 | 75000 | 3

(3 rows)

**1.Write a cursor which will display details of employees invested amount in "Mutual Fund"**

mydb=# create or replace function invest() returns void as $$

mydb$# declare

mydb$# m\_cursor cursor for

mydb$# SELECT e.e\_id, e.e\_name, e.addr

mydb$# FROM Emp e, inv i

mydb$# where e.e\_id = i.e\_id

mydb$# and i.inv\_name = 'mutual fund';

mydb$# rec record;

mydb$# begin

mydb$# open m\_cursor;

mydb$# loop

mydb$# fetch m\_cursor into rec;

mydb$# exit when not found;

mydb$# raise notice'% % %',rec.e\_id, rec.e\_name, rec.addr;

mydb$# end loop;

mydb$# close m\_cursor;

mydb$# end;

mydb$# $$ language plpgsql;

CREATE FUNCTION

mydb=#

mydb=# select invest();

NOTICE: 1 Nilam pune

invest

--------

(1 row)

**2.Write a cursor which will display date wise investment details.**

create or replace function date\_Details() r

eturns void as $$

mydb$# declare

mydb$# date\_cursor cursor for

mydb$# select inv\_no,inv\_name,inv\_date,inv\_amt

mydb$# from inv order by inv\_date;

mydb$# rec record;

mydb$# begin

mydb$# open date\_cursor;

mydb$# loop

mydb$# fetch date\_cursor into rec;

mydb$# exit when not found;

mydb$# raise notice'% % % %',rec.inv\_no, rec.inv\_n

ame, rec.inv\_date, rec.inv\_amt;

mydb$# end loop;

mydb$# close date\_cursor;

mydb$# end;

mydb$# $$ language plpgsql;

CREATE FUNCTION

mydb=#

mydb=# select date\_Details();

NOTICE: 101 mutual fund 2024-10-12 50000

NOTICE: 103 Lic 2024-11-16 75000

NOTICE: 102 sip 2024-11-18 80000

date\_details

--------------

(1 row)

**2. Consider the following entities and their relationships.**

**Plan (plan no, plan name, no of free calls, freecalltime, fix amt)**

**Customer (cust no cust name, mobile no)**

**Relation between Plan and Customer is One to Many,**

**Constraint, Primary key, fiz amt should be greater than 0.**

**Create a RDB in 3NF and write procedure for the following**

create table plan(p\_no int primary key,p\_na

me varchar(20),no\_of\_free\_calls int,free\_call\_time

int,fix\_amt int check(fix\_amt>0));

CREATE TABLE

mydb=# create table customer(cus\_no int primary ke

y,cus\_name varchar(30),mobile\_no int,p\_no int refe

rences plan);

CREATE TABLE

mydb=# insert into plan values(1,'premium',100,60,

999);

INSERT 0 1

mydb=# insert into plan values(2,'basic',50,30,199

);

INSERT 0 1

mydb=# insert into plan values(3,'unlimited',300,1

00,499);

INSERT 0 1

mydb=# create table customer(cus\_no int primary key, cus\_name varchar(30),mobile\_no varchar(15)unique,p\_no int references plan);

CREATE TABLE

mydb=# insert into customer values(101,'amit',9988776655,1);

INSERT 0 1

mydb=# insert into customer values(102,'sahil',9834567821,2);

INSERT 0 1

mydb=# insert into customer values(103,'gaurav',8834563821,3);

INSERT 0 1

mydb=# select \* from plan;

p\_no | p\_name | no\_of\_free\_calls | free\_call\_t

ime | fix\_amt

------+-----------+------------------+------------

----+---------

1 | premium | 100 |

60 | 999

2 | basic | 50 |

30 | 199

3 | unlimited | 300 |

100 | 499

(3 rows)

mydb=# select \* from customer;

cus\_no | cus\_name | mobile\_no | p\_no

--------+----------+------------+------

101 | amit | 9988776655 | 1

102 | sahil | 9834567821 | 2

103 | gaurav | 8834563821 | 3

(3 rows)

**1)Write a cursor which will accept plan number from user and display all the details of the selected plan.**

create or replace function details(plan\_num int) returns void as $$

mydb$# declare

mydb$# p\_cursor cursor for

mydb$# select p\_no,p\_name,no\_of\_free\_calls,free\_call\_time,fix\_amt

mydb$# from plan where p\_no = plan\_num ;

mydb$# rec record;

mydb$# begin

mydb$# open p\_cursor;

mydb$# loop

mydb$# fetch p\_cursor into rec;

mydb$# exit when not found;

mydb$# raise notice'% % % % %',rec.p\_no,rec.p\_name

,rec.no\_of\_free\_calls,rec.free\_call\_time,rec.fix\_a

mt;

mydb$# end loop;

mydb$# close p\_cursor;

mydb$# end;

mydb$# $$ language plpgsql;

CREATE FUNCTION

mydb=#

mydb=# select details(2);

NOTICE: 2 basic 50 30 199

details

---------

(1 row)

**2)Write a cursor which will display customer wise plan details (Use Parameterized Cursor)**

create or replace function customer\_details

(cust\_num int) returns void as $$

mydb$# declare

mydb$# name\_cursor cursor for

mydb$# select c.cus\_no,c.cus\_name,c.mobile\_no,p.p\_

no,p.p\_name

mydb$# from customer c, plan p where c.cus\_no = cu

st\_num ;

mydb$# rec record;

mydb$# begin

mydb$# open name\_cursor;

mydb$# loop

mydb$# fetch name\_cursor into rec;

mydb$# exit when not found;

mydb$# raise notice'% % % % %',rec.cus\_no,rec.cus\_

name,rec.mobile\_no,rec.p\_no,rec.p\_name;

mydb$# end loop;

mydb$# close name\_cursor;

mydb$# end;

mydb$# $$ language plpgsql;

CREATE FUNCTION

mydb=#

mydb=# select customer\_details(102);

NOTICE: 102 sahil 9834567821 1 premium

NOTICE: 102 sahil 9834567821 2 basic

NOTICE: 102 sahil 9834567821 3 unlimited

customer\_details

------------------

(1 row)