**Created By :**

Lakhani Dinky A. IT062

Modi Aastha N. IT079

Gurvinder Singh P. IT084

**Under the guidance of :**

Prof. Archana N Vyas

Prof Fatema Vhora

**Sem :** 5

**Batch :** H3

**Topic :** OmniLib

**Subject :** Database Management System

**Univ. :** Dharmsinh Desai University

**1) PL/SQL**

**Checks if Payment Amount is from 400,750,1000,300**

**DO $$**

**DECLARE**

**payment\_amount INTEGER :=100;**

**BEGIN**

**if payment\_amount = 400 then**

**elsif payment\_amount = 750 then**

**elsif payment\_amount = 1000 then**

**elsif payment\_amount = 3000 then**

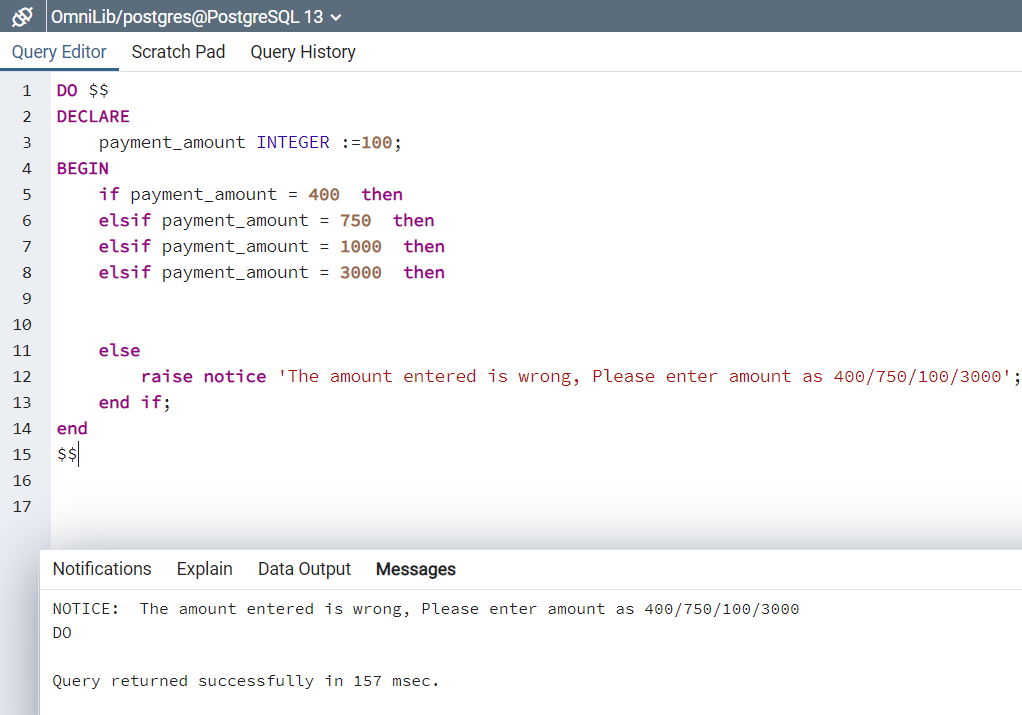
**else**

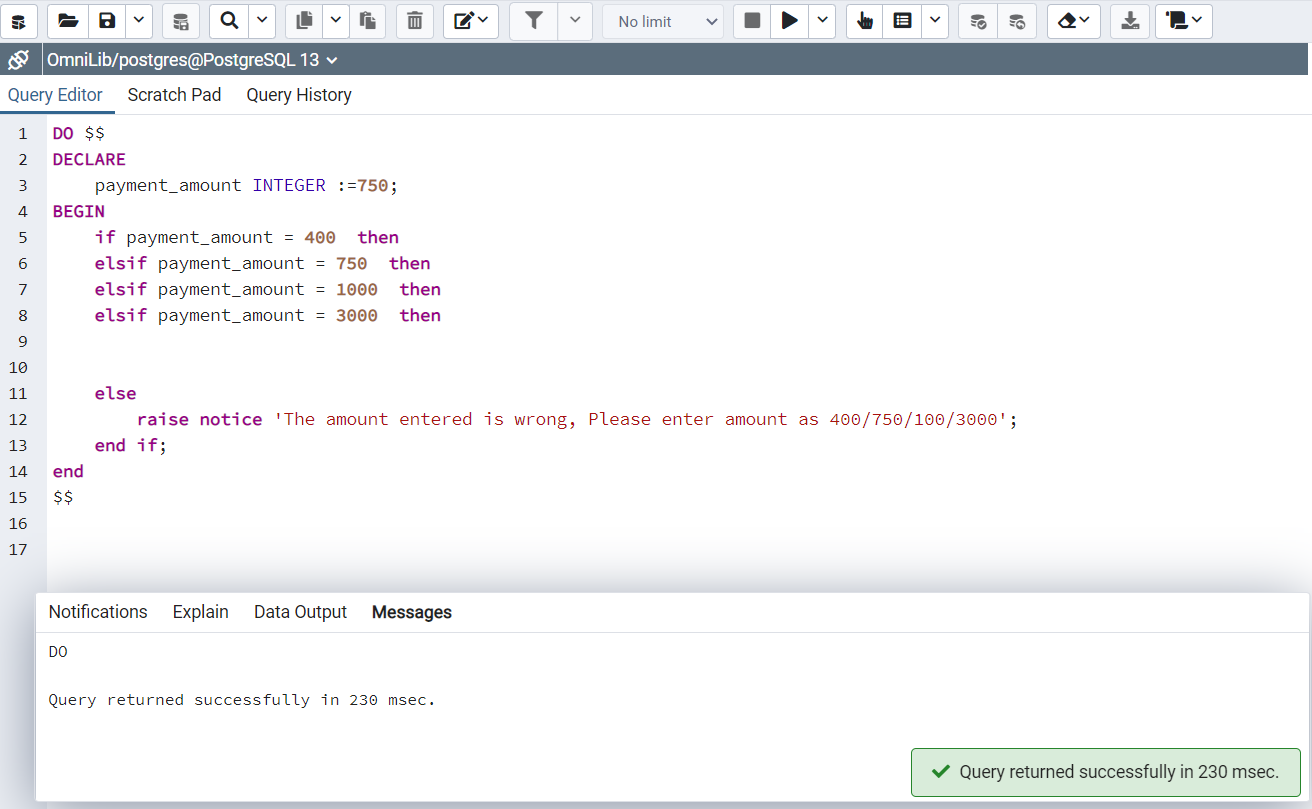
**raise notice 'The amount entered is wrong, Please enter amount as 400/750/100/3000';**

**end if;**

**end**

**$$**

****

****

**2) Function**

**Function get\_record displays the payment amount as well as name and payment\_type of a user**

**Function**

**create or replace function get\_record (**

**pid varchar(20)**

**)**

**returns table (**

**amt numeric(4,0),**

**pay\_ty varchar(20),**

**user\_f varchar(50),**

**user\_l varchar(50)**

**)**

**language plpgsql**

**as $$**

**begin**

**return query**

**select**

**amount,**

**pay\_type::varchar(20),**

**first\_name,**

**last\_name**

**from**

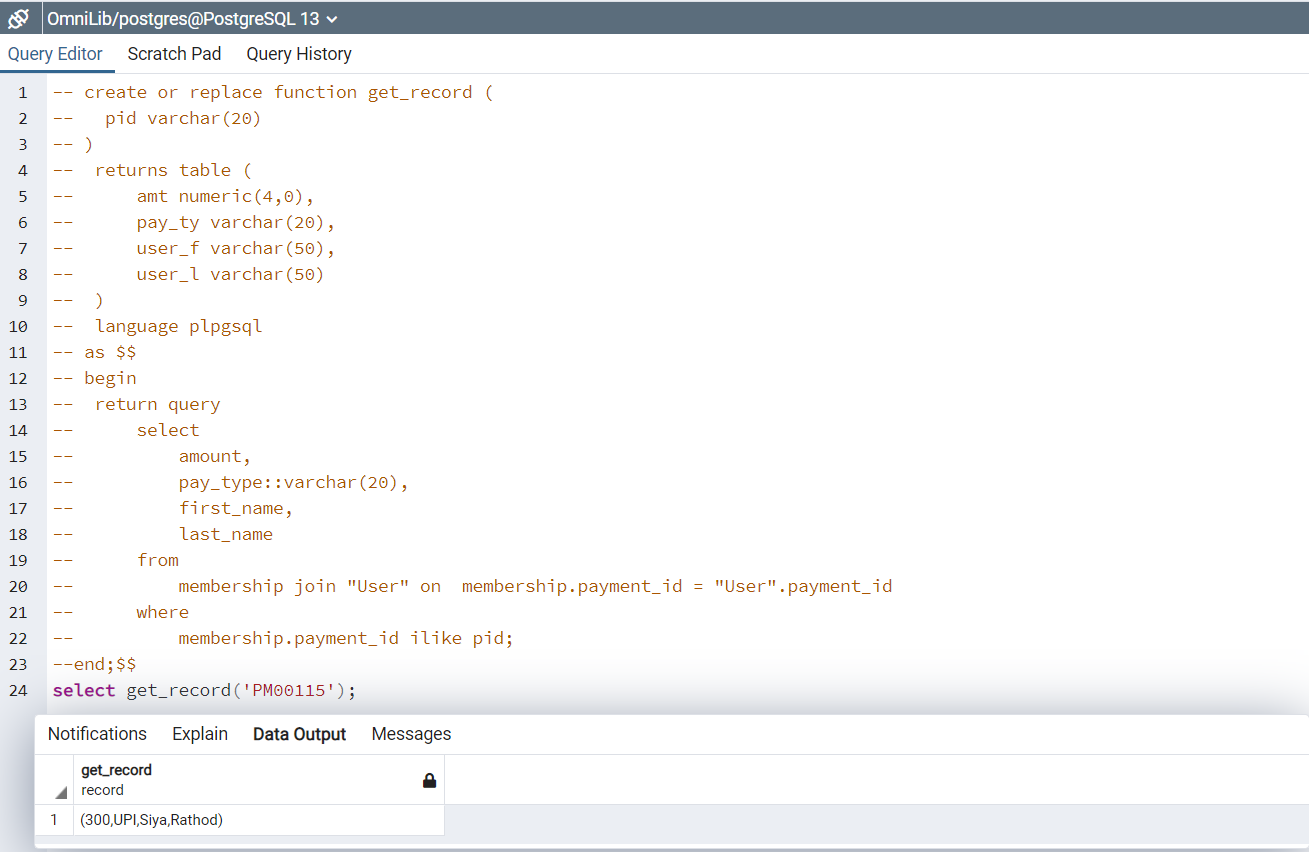
**membership join "User" on membership.payment\_id = "User".payment\_id**

**where**

**membership.payment\_id ilike pid;**

**end;$$**

**select get\_record('PM00115');**

****

**3)Trigger**

**-- create or replace function check\_Qty() returns trigger as $$**

**-- BEGIN**

**-- if NEW."Qty" <=0 then**

**-- raise exception 'Book Not available at the moment';**

**-- end if;**

**-- return NEW;**

**-- END;**

**-- $$**

**-- LANGUAGE plpgsql;**

**-- create trigger Qty\_check**

**-- BEFORE INSERT OR UPDATE**

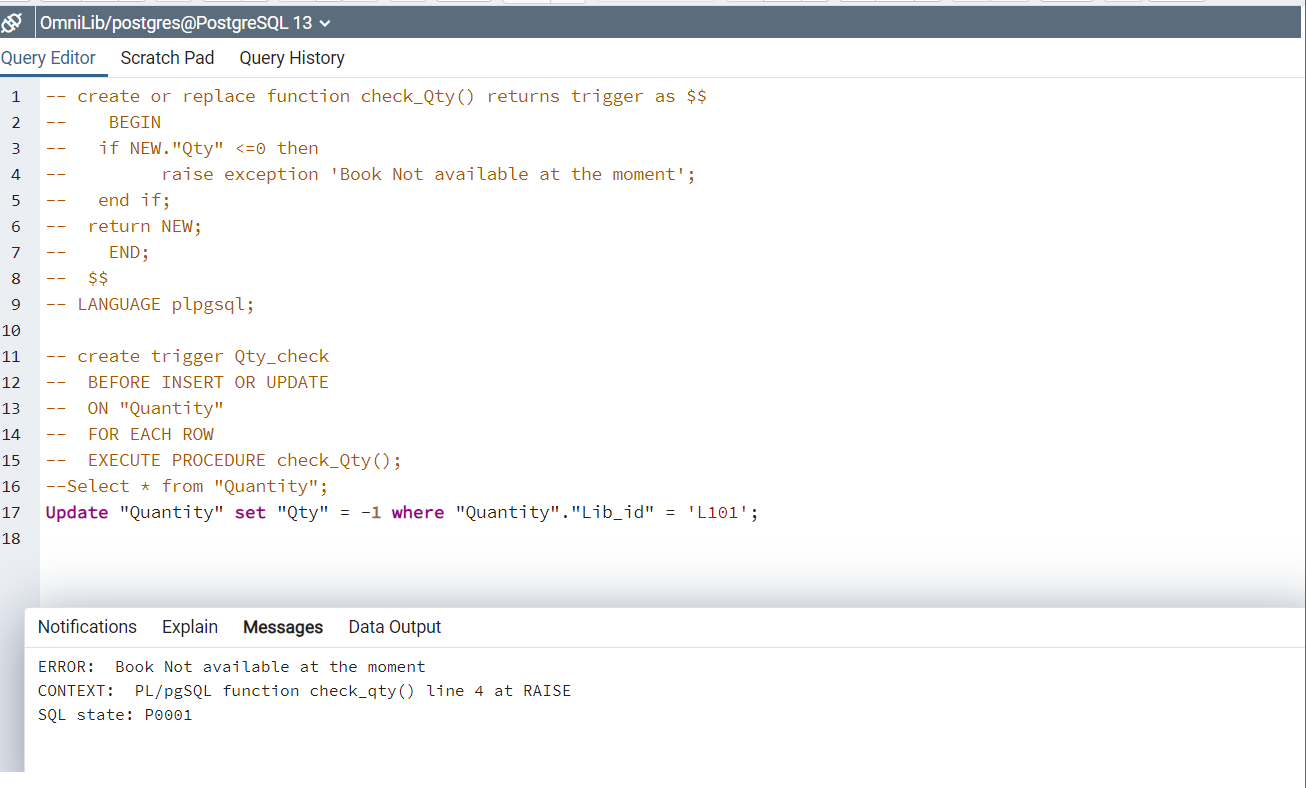
**-- ON "Quantity"**

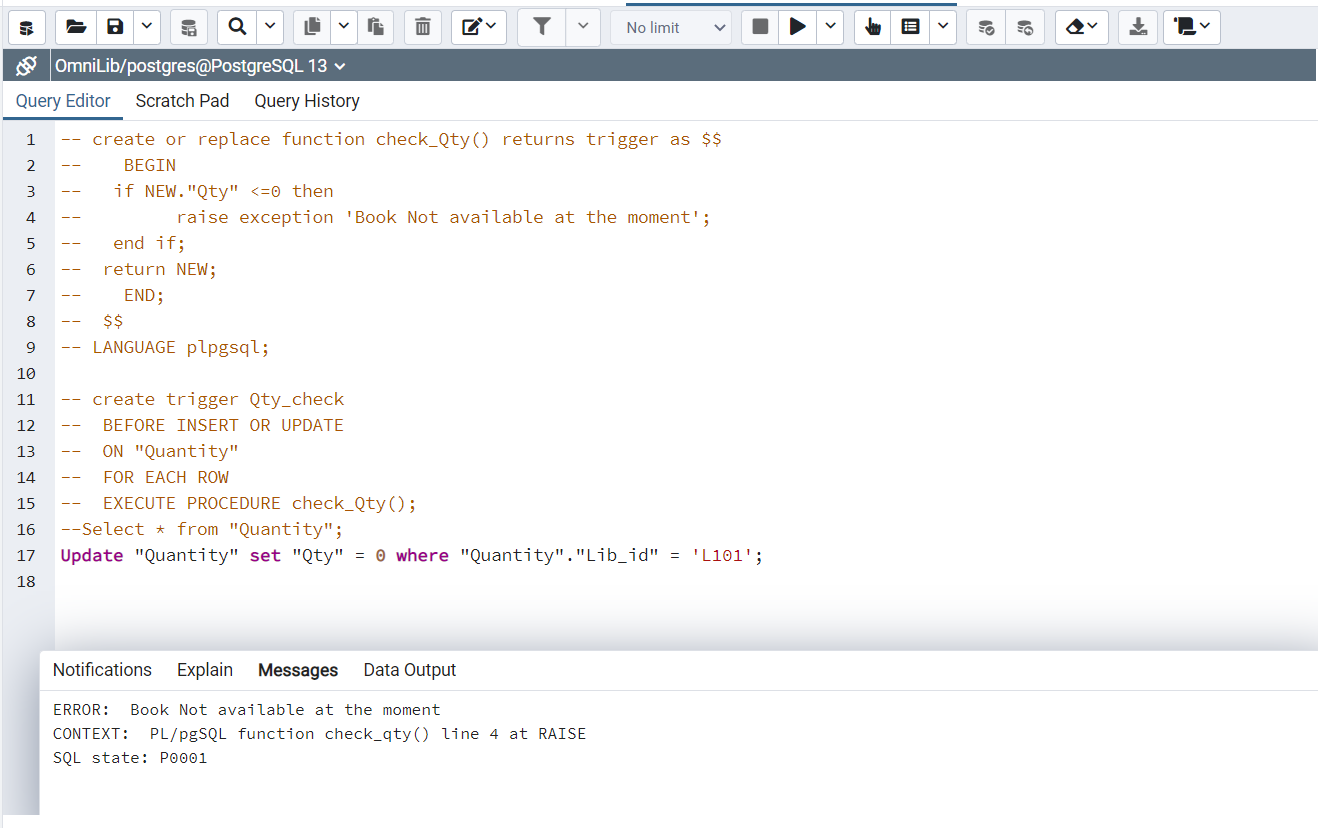
**-- FOR EACH ROW**

**-- EXECUTE PROCEDURE check\_Qty();**

**--Select \* from "Quantity";**

**Update "Quantity" set "Qty" = 0 where "Quantity"."Lib\_id" = 'L101';**

****

****

**4)Trigger**

**CREATE OR REPLACE FUNCTION emp\_record() RETURNS trigger AS $emp\_record$**

**Declare**

**fl int := 0;**

**BEGIN**

**-- Check that empname and salary are given**

**IF NEW.first\_name IS NULL THEN**

**RAISE EXCEPTION 'First name cannot be null';**

**fl:=1;**

**END IF;**

**IF NEW.salary IS NULL THEN**

**RAISE EXCEPTION 'Cannot have null salary';**

**fl:=1;**

**END IF;**

**IF NEW.desig NOT IN ('librarian','watchman','staff') THEN**

**RAISE EXCEPTION 'not allowed for %', NEW.desig;**

**fl:=1;**

**END IF;**

**IF NEW.salary < 15000 THEN**

**RAISE EXCEPTION 'Cannot have a salary less than 15000 of %', NEW.first\_name;**

**fl:=1;**

**END IF;**

**-- IF fl=0 then**

**-- INSERT INTO employee VALUES(NEW.desig,NEW.first\_name,NEW.last\_name,NEW.salary,NEW.emp\_id,NEW.lib\_id);**

**-- End if;**

**RETURN NEW;**

**END;**

**$emp\_record$ LANGUAGE plpgsql;**

**-- Who**

**-- CREATE TRIGGER emp\_rec**

**-- BEFORE**

**-- INSERT OR UPDATE ON employee**

**-- FOR EACH ROW EXECUTE FUNCTION emp\_record();**

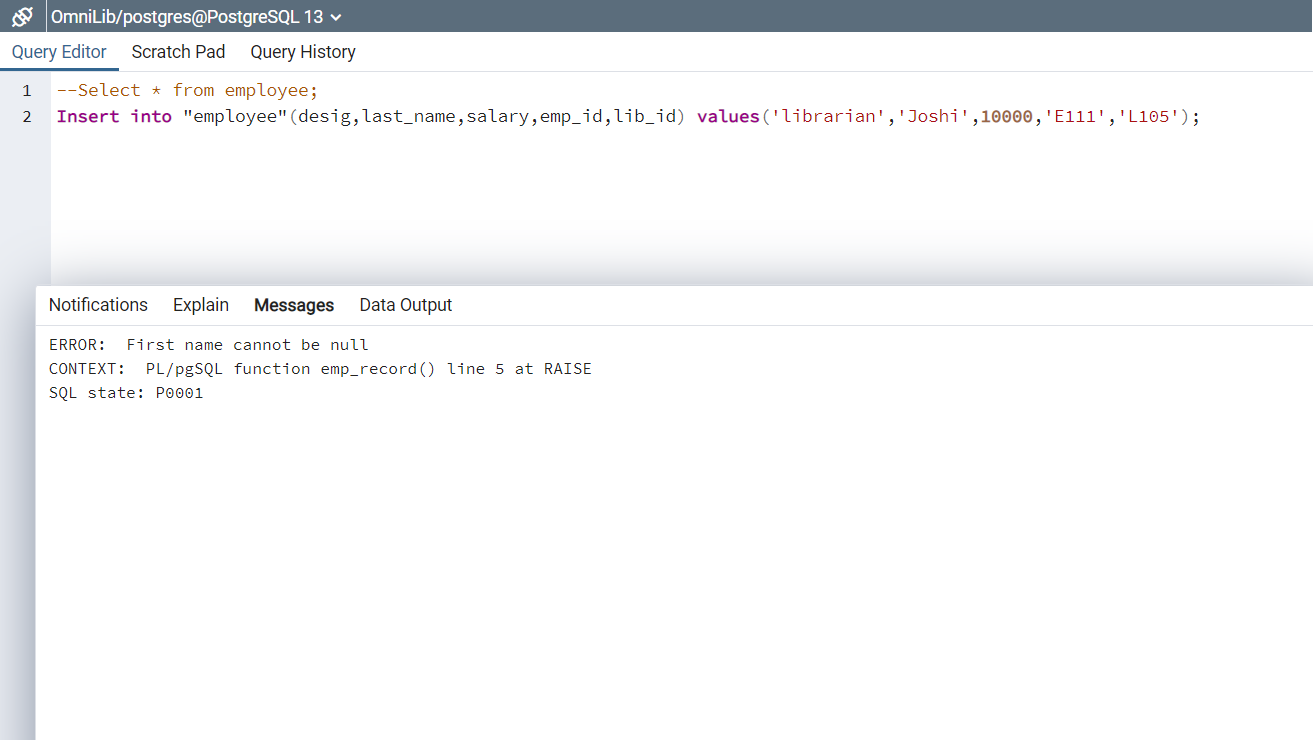
**--Select \* from employee;**

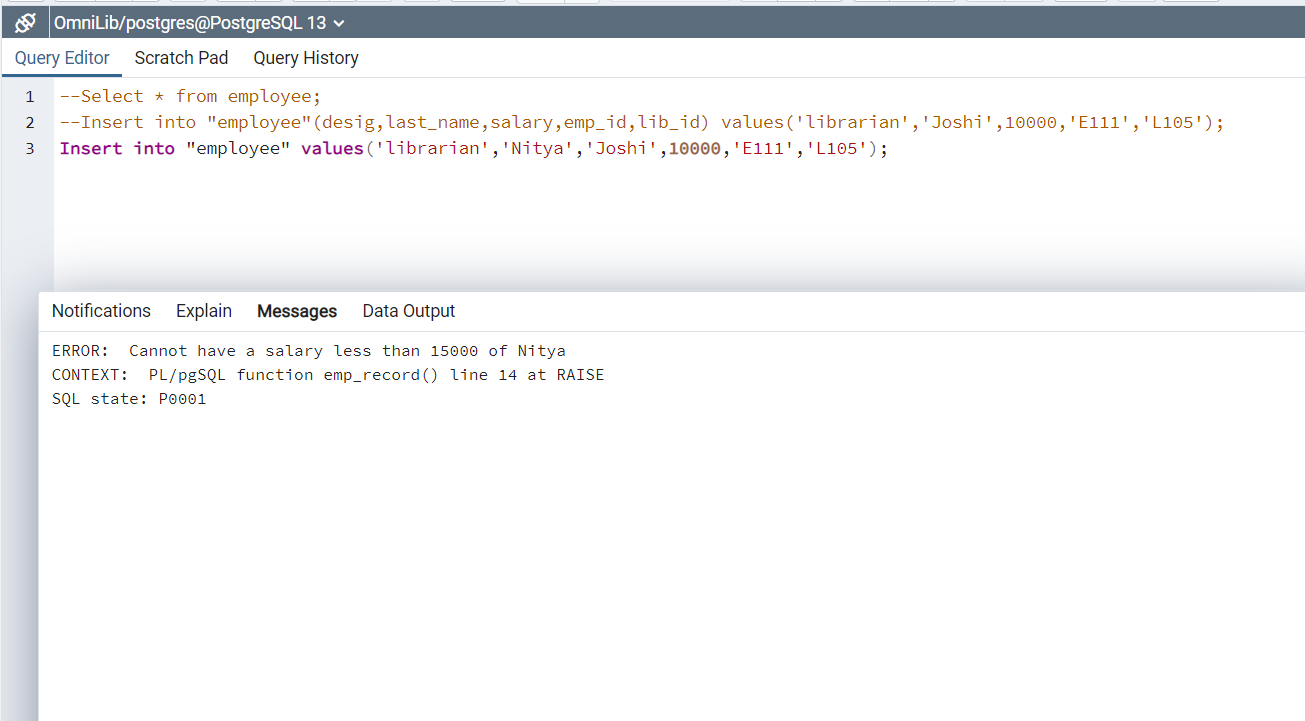
**--Insert into "employee" values('teacher','Nitya','Joshi',10000,'E111','L105');**

**--Insert into "employee"(desig,last\_name,salary,emp\_id,lib\_id) values('librarian','Joshi',10000,'E111','L105');**

**--Insert into "employee" values('librarian','Nitya','Joshi',10000,'E111','L105');**

**--Insert into "employee" values('librarian','Nitya','Joshi',20000,'E111','L105');**

****

****

**5) Cursor**

**-- Cursor**

**-- ::::::To display which book is issued by a user at this time**

**create or replace function book\_issued(uid varchar(50)) returns text as $$**

**declare**

**name text default '';**

**emprec record;**

**booki cursor(uid varchar(50))**

**for select first\_name,last\_name,"User"."User\_id/email\_id",isbn**

**from "User" join "Return/Issue" on**

**"User"."User\_id/email\_id" = "Return/Issue"."User\_id/email\_id" and "User"."User\_id/email\_id" = uid;**

**begin**

**open booki(uid);**

**loop**

**fetch booki into emprec;**

**exit when not found;**

**name := emprec.first\_name || ' , ' || emprec.last\_name || ' , ' || emprec."User\_id/email\_id" || ' , ' || emprec.isbn;**

**end loop;**

**close booki;**

**return name;**

**end; $$**

**language plpgsql;**

**select book\_issued('sharvarigad14@gmail.com');**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**