**Triggers**

create table passenger(

id serial primary key,

name varchar,

age int,

seatnumber int,

price float

);

create table price\_hike(

description varchar

);

CREATE OR REPLACE function display\_price\_changes()

returns trigger as

$$

DECLARE

price\_diff int;

BEGIN

price\_diff := NEW.price - OLD.price;

insert into price\_hike values('old price: ' || OLD.price || ' updated price: ' || NEW.price || ' and the price difference is : ' || price\_diff || ' for the passenger ' || NEW.name);

return new;

END;

$$

Language 'plpgsql'

insert into passenger values(1,'himanshu',19,10,1000);

insert into passenger values(2,'anurag',19,20,2000);

create trigger price\_hike\_trigger

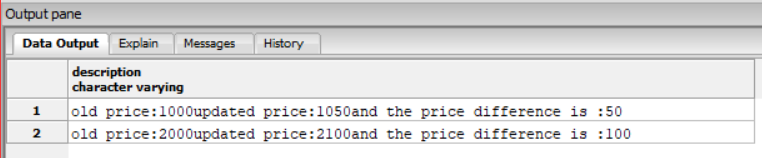
before update

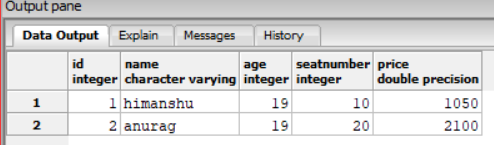
on passenger

for each row

execute procedure display\_price\_changes();

update passenger set price = price + price\*0.05;





create table passenger\_data(

id serial primary key,

name varchar,

seatnumber int,

price float,

from\_city varchar,

to\_city varchar

);

create table booking\_details(

description varchar

);

create or replace function booking\_insert()

returns trigger as

$$

begin

insert into booking\_details values(new.name || ' has booked a ticket from :' || new.from\_city || ' to :' || new.to\_city || ' at date/time : ' || now());

return new;

end;

$$

language 'plpgsql';

create trigger passenger\_detail

after insert

on passenger\_data

for each row

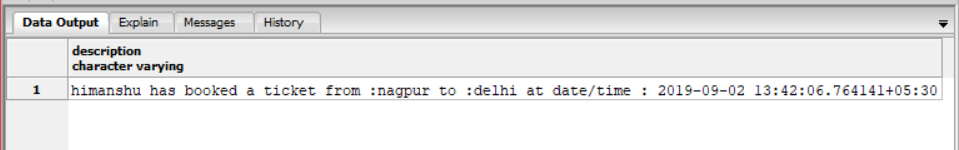
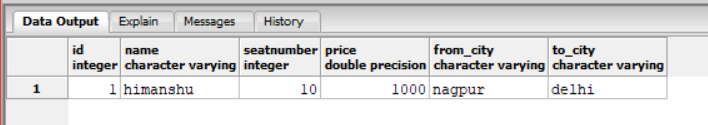
execute procedure booking\_insert();

insert into passenger\_data values(1,'himanshu',10,1000,'nagpur','delhi');

delete from passenger\_data

select \* from passenger\_data

select \* from booking\_details

**Procedures**

create table iiii(n integer);

delete from iiii;

CREATE OR REPLACE FUNCTION fibonacci (n INTEGER)

RETURNS INTEGER AS $$

DECLARE

counter INTEGER := 0 ;

i INTEGER := 0 ;

j INTEGER := 1 ;

BEGIN

IF (n < 1) THEN

RETURN 0 ;

END IF;

LOOP

EXIT WHEN counter = n ;

counter := counter + 1 ;

SELECT j, i + j INTO i, j ;

insert into iiii values(j);

END LOOP ;

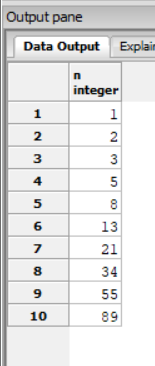
RETURN i ;

END ;

$$ LANGUAGE plpgsql;

select fibonacci(10)

select \* from iiii



create function sum\_of\_two\_numbers(m integer,n integer)

returns integer as $$

begin

return m+n;

end;

$$ language 'plpgsql'

select sum\_of\_two\_numbers(10,20)

