**DBMS - UE20CS301 Lab**

**Lab 8**

**Name**: Vishwa Mehul Mehta

**Sec**: F

**SRN**: PES2UG20CS389

1.

Code:

DELIMITER $$

CREATE FUNCTION number\_of\_tickets\_booked(no\_of\_tickets int)

RETURNS varchar(50)

BEGIN

DECLARE err varchar(50);

IF no\_of\_tickets > 3 THEN

SET err = 'cannot purchase tickets current limit is over';

ELSE

SET err = 'can purchase tickets';

END IF;

RETURN err;

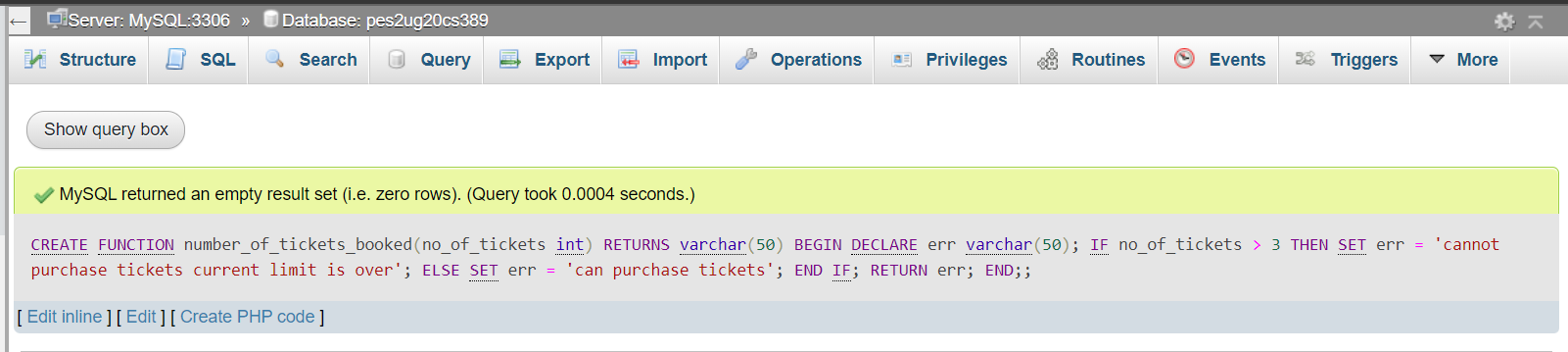
END; $$

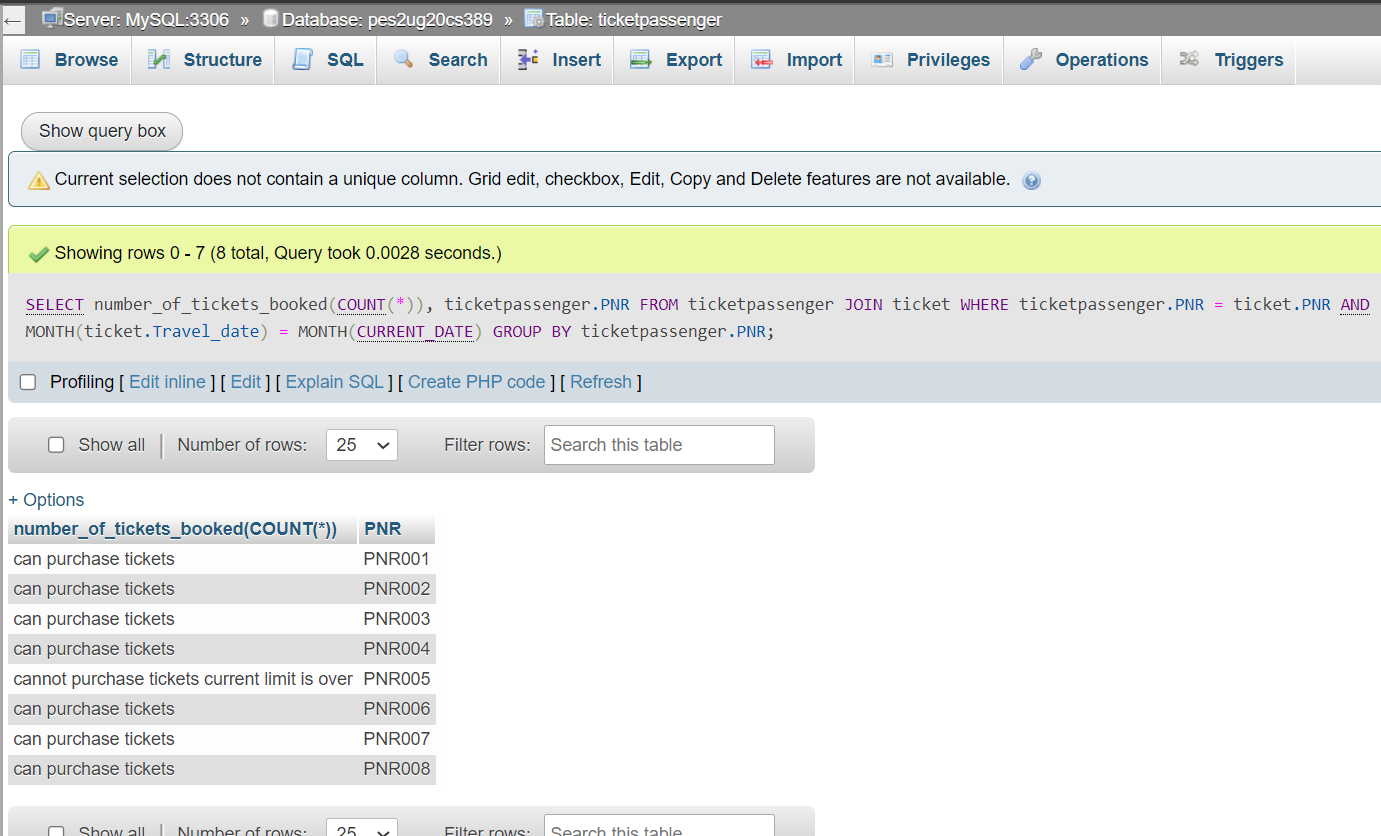
DELIMITER ;

Function call query:

SELECT number\_of\_tickets\_booked(COUNT(\*)), ticketpassenger.PNR FROM ticketpassenger JOIN ticket WHERE ticketpassenger.PNR = ticket.PNR AND MONTH(ticket.Travel\_date) = MONTH(CURRENT\_DATE) GROUP BY ticketpassenger.PNR;

Output Screenshots:





2.

Code:

DELIMITER $$

CREATE procedure age\_updation(IN age date)

BEGIN

update users set users.Age = (year(CURRENT\_DATE) - year(age)) where users.DOB = age;

END; $$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE age()

BEGIN

DECLARE dob DATE;

DECLARE is\_updated INT DEFAULT FALSE;

DECLARE age\_update CURSOR FOR SELECT users.DOB FROM users;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET is\_updated = TRUE;

OPEN age\_update;

iterate\_through\_rows:LOOP

FETCH age\_update INTO dob;

IF is\_updated THEN

LEAVE iterate\_through\_rows;

END IF;

call age\_updation(dob);

END LOOP;

CLOSE age\_update;

END; $$

DELIMITER ;

Procedure call query:

CALL age();

Output Screenshots:

