

Name: Varun H Ramurs  
SRN: PES1UG20CS487  
Section: H  
Roll-No: 12

## **DBMS LAB-8**

Q1)

### **Code:**

```
DELIMITER $$  
CREATE FUNCTION number_of_tickets(P_ID varchar(255))  
RETURNS varchar(255)  
DETERMINISTIC  
BEGIN  
DECLARE msg varchar(255);  
DECLARE tickets int;  
SELECT count(pnr) into tickets  
FROM ticket  
WHERE passenger_id = P_ID AND MONTH(travel_date) = MONTH(NOW()) AND  
YEAR(travel_date) = YEAR(NOW());  
IF tickets > 3 THEN  
SET msg = 'cannot purchase tickets current limit is over';  
ELSE  
SET msg = 'can purchase tickets';  
END IF;  
RETURN msg;  
END; $$  
DELIMITER ;  
  
SELECT number_of_tickets('ADM_001') from ticket;
```

### **Output Screenshot:**

```
mysql> CREATE FUNCTION number_of_tickets(P_ID varchar(255))
-> RETURNS VARCHAR(255)
-> DETERMINISTIC
-> BEGIN
-> DECLARE msg varchar(255);
-> DECLARE tickets int;
-> SELECT count(pnr) into tickets
-> FROM ticket
-> WHERE passenger_id = P_ID AND MONTH(travel_date) = MONTH(NOW()) AND YEAR(travel_date) = YEAR(NOW());
-> IF tickets > 3 THEN
-> SET msg = 'cannot purchase tickets current limit is over'';
-> ELSE
-> SET msg = 'can purchase tickets';
-> END IF;
-> RETURN msg;
-> END; $$
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql>
mysql> SELECT number_of_tickets('ADM_001') from ticket;
+-----+
| number_of_tickets('ADM_001') |
+-----+
| can purchase tickets         |
| can purchase tickets         |
| can purchase tickets         |
| can purchase tickets         |
| can purchase tickets         |
| can purchase tickets         |
| can purchase tickets         |
| can purchase tickets         |
| can purchase tickets         |
| can purchase tickets         |
| can purchase tickets         |
| can purchase tickets         |
| can purchase tickets         |
| can purchase tickets         |
+-----+
14 rows in set (0.01 sec)
```

Q2)

## **Code:**

```
DELIMITER $$
CREATE procedure update_age(IN U_ID varchar(30), IN DOB DATE)
BEGIN
DECLARE new_age int;
SET new_age = FLOOR(DATEDIFF(NOW(),DOB)/365);
UPDATE train_user set age = new_age where user_id = U_ID;
END;$$
DELIMITER ;

CALL update_age('ADM_001','1989-04-14');
```

## Output Screenshot:

```
mysql> DELIMITER $$
mysql> CREATE procedure update_age(IN U_ID varchar(30), IN DOB DATE)
-> BEGIN
-> DECLARE new_age int;
-> SET new_age = FLOOR(DATEDIFF(NOW(),DOB)/365);
-> UPDATE train_user set age = new_age where user_id = U_ID;
-> END;$$
```

Query OK, 0 rows affected (0.01 sec)

```
mysql> DELIMITER ;
```

```
mysql>
```

```
mysql> CALL update_age('ADM_001','1989-04-14');
```

Query OK, 1 row affected (0.01 sec)

```
mysql> select * from train_user;
```

user_id	user_type	firstname	lastname	age	dob	pincode	street_number
ADM_001	ADMIN	Narendra	Modi	33	1989-04-14	600047	Old Jail Road
ADM_002	ADMIN	Droupadi	Murmu	NULL	1990-04-14	640012	High Street