DBMS
LAB 8
Functions and Procedures

Hita Juneja
PES1UG20CS645
Section K
Roll No. 27

1. Function

```
DELIMITER $$
CREATE FUNCTION final_func(user varchar(20))
RETURNS varchar(20)
BEGIN
DECLARE res varchar(20);
DECLARE tot_count int;
SET tot_count = (SELECT COUNT(*) FROM ticket NATURAL JOIN ticket_passenger WHERE
ticket.pnr=pnr AND
MONTH(ticket.travel_date)=MONTH(CURRENT_DATE) GROUP BY ticket.pnr);
IF tot_count > 3 THEN
SET res = "NOT ELIGIBLE";
ELSE
SET res = "ELIGIBLE";
END IF;
RETURN res;
END;$$
DELIMITER ;
SELECT final_func(pnr),pnr from ticket;
```

```
MariaDB [pes1ug20cs645_railway]> DELIMITER $$
MariaDB [pes1ug20cs645_railway]> CREATE FUNCTION final_func(user varchar(20))
    -> RETURNS varchar(20)
    -> BEGIN
    -> DECLARE res varchar(20);
    -> DECLARE tot_count int;
    -> SET tot_count = (SELECT COUNT(*) FROM ticket NATURAL JOIN ticket_passenger WHERE ticket.pnr=pnr AND
    -> MONTH(ticket.travel_date)=MONTH(CURRENT_DATE) GROUP BY ticket.pnr);
    -> IF tot_count > 3 THEN
    -> SET res = "NOT ELIGIBLE";
    -> ELSE
    -> SET res = "ELIGIBLE";
    -> END IF;
    -> RETURN res;
    -> END;$$
Query OK, 0 rows affected (0.015 sec)

MariaDB [pes1ug20cs645_railway]> DELIMITER ;
MariaDB [pes1ug20cs645_railway]> SELECT final_func(pnr),pnr from ticket;
+-----------------+--------+
| final_func(pnr) | pnr    |
+-----------------+--------+
| ELIGIBLE        | PNR004 |
| ELIGIBLE        | PNR010 |
| ELIGIBLE        | PNR014 |
| ELIGIBLE        | PNR015 |
| ELIGIBLE        | PNR003 |
| ELIGIBLE        | PNR006 |
| ELIGIBLE        | PNR005 |
| ELIGIBLE        | PNR012 |
| ELIGIBLE        | PNR002 |
| ELIGIBLE        | PNR008 |
| ELIGIBLE        | PNR001 |
| ELIGIBLE        | PNR007 |
| ELIGIBLE        | PNR011 |
| ELIGIBLE        | PNR021 |
+-----------------+--------+
14 rows in set (0.002 sec)
```

2. Procedure

```
DELIMITER $$
CREATE procedure age_updation(
IN UID varchar(30),IN DB date, OUT msg varchar(30))
BEGIN
DECLARE age int; IF DB>sysdate() THEN
set msg= 'Invalid DoB';
ELSE
update train_user
set Age=(datediff(sysdate(),DB))/365 where user_id= UID;
update train_user
set DOB=DB where user_id=UID;
set msg='Age updated Successfully';
END IF;
END;$$
DELIMITER ;
CALL age_updation('USR_001','2002-08-26',@A);
SELECT @A;
select * from train_user where user_id='USR_001';
```

```
MariaDB [pes1ug20cs645_railway]> DELIMITER $$
MariaDB [pes1ug20cs645_railway]> CREATE procedure age_updation(
    -> IN UID varchar(30),IN DB date, OUT msg varchar(30))
    -> BEGIN
    -> DECLARE age int; IF DB>sysdate() THEN
    -> set msg= 'Invalid DoB';
    -> ELSE
    -> update train_user
    -> set Age=(datediff(sysdate(),DB))/365 where user_id= UID;
    -> update train_user
    -> set DOB=DB where user_id=UID;
    -> set msg='Age updated Successfully';
    -> END IF;
    -> END;$$
Query OK, 0 rows affected (0.024 sec)

MariaDB [pes1ug20cs645_railway]> DELIMITER ;
MariaDB [pes1ug20cs645_railway]> CALL age_updation('USR_001','2002-08-26',@A);
Query OK, 2 rows affected (0.007 sec)

MariaDB [pes1ug20cs645_railway]> SELECT @A;
+--------------------------+
| @A                       |
+--------------------------+
| Age updated Successfully |
+--------------------------+
1 row in set (0.000 sec)

MariaDB [pes1ug20cs645_railway]> select * from train_user where user_id='USR_001';
+---------+-----------+-----------+----------+------+------------+---------+---------------+
| user_id | user_type | firstname | lastname | age  | dob        | pincode | street_number |
+---------+-----------+-----------+----------+------+------------+---------+---------------+
| USR_001 | PSGR      |   Ajit    | Ullal    |   20 | 2002-08-26 | 560001  | 10, Janpath   |
+---------+-----------+-----------+----------+------+------------+---------+---------------+
1 row in set (0.000 sec)
```