

**Database Management Systems**  
**UE20CS301 5th Semester,**  
**Academic Year 2022-23**

**Date: 18/10/2022**

**Name: Samyam N**  
**SRN: PES1UG20CS715**  
**Sec: L**

**ASSIGNMENT 8**

```
1)
DELIMITER $$
CREATE FUNCTION count_tickets(ticket int)
RETURNS VARCHAR(50)
DETERMINISTIC
BEGIN
DECLARE VALUE varchar(50);
IF ticket>3 then
set VALUE="Cannot purchase tickets Current limit is over";
ELSE
set VALUE ="Can Purchase ticket";
end if;
return value;

END;
$$
DELIMITER ;

with t as (Select user_id,count(pnr) as count from ticket group by user_id )
select user_id,count_tickets(count) as Validate,count as ticket_purchased from t;
```

```
2)
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
```

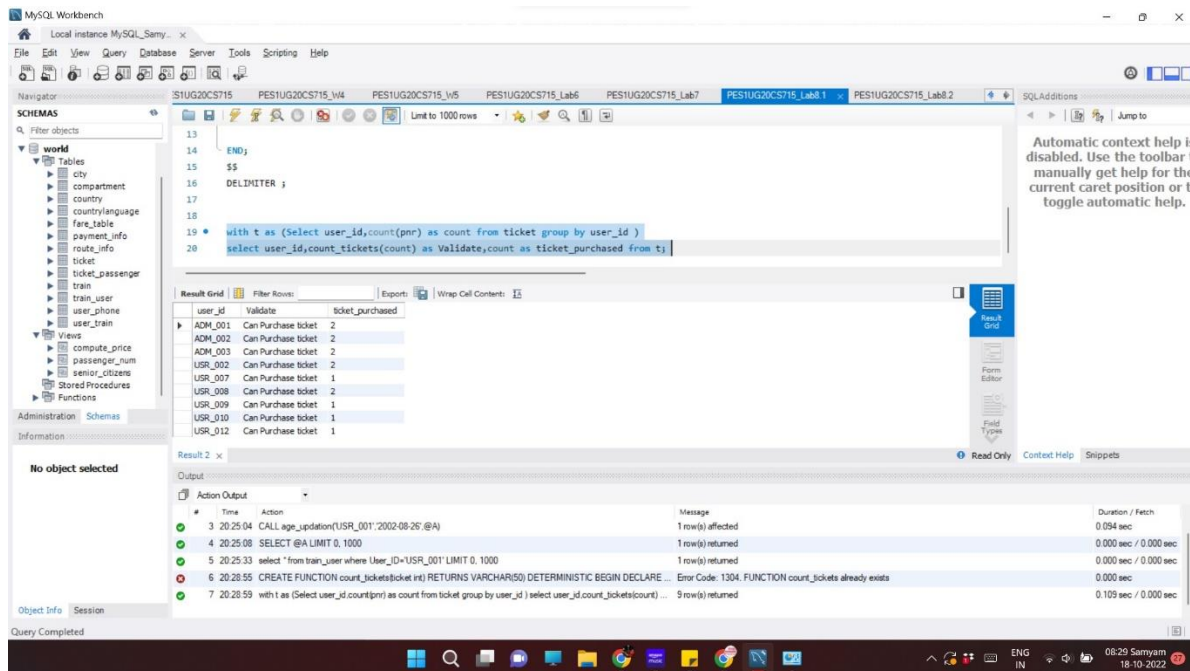
```
where User_ID=UID;
set msg='Age updated Successfully';
```

```
END IF;
END;$$
DELIMITER ;
```

```
CALL age_update('USR_001','2002-08-26',@A);
SELECT @A;
```

```
select * from train_user where User_ID='USR_001';
```

## Task 1)



The screenshot shows the MySQL Workbench interface. The SQL editor contains the following code:

```

13
14 END;
15 $$
16 DELIMITER ;
17
18
19 * with t as (select user_id,count(pnr) as count from ticket group by user_id )
20 select user_id,count_tickets(count) as Validate,count as ticket_purchased from t;

```

The Result Grid shows the following data:

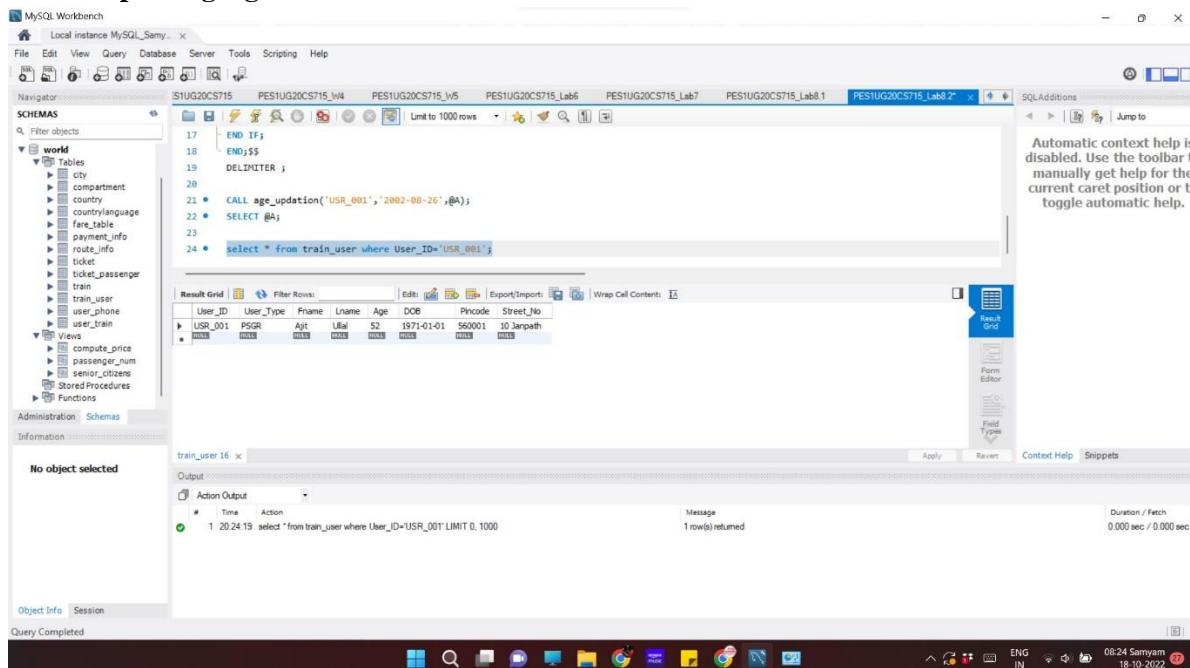
user_id	Validate	ticket_purchased
ADM_001	Can Purchase ticket	2
ADM_002	Can Purchase ticket	2
ADM_003	Can Purchase ticket	2
USR_002	Can Purchase ticket	2
USR_007	Can Purchase ticket	1
USR_008	Can Purchase ticket	2
USR_009	Can Purchase ticket	1
USR_010	Can Purchase ticket	1
USR_012	Can Purchase ticket	1

The Output tab shows the execution log:

#	Time	Action	Message	Duration / Fetch
3	20:25:04	CALL age_update('USR_001','2002-08-26',@A)	1 row(s) affected	0.094 sec
4	20:25:06	SELECT @A LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
5	20:25:33	select * from train_user where User_ID='USR_001' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
6	20:28:55	CREATE FUNCTION count_tickets(pnr) RETURNS VARCHAR(50) DETERMINISTIC BEGIN DECLARE ...	Error Code: 1304. FUNCTION count_tickets already exists	0.000 sec
7	20:28:59	with t as (select user_id,count(pnr) as count from ticket group by user_id ) select user_id,count_tickets(count) ...	9 row(s) returned	0.109 sec / 0.000 sec

## Task 2)

### Before Updating Age and DoB



The screenshot shows the MySQL Workbench interface. The SQL editor contains the following code:

```

17 END IF;
18 END;$$
19 DELIMITER ;
20
21 * call age_update('USR_001','2002-08-26',@A);
22 SELECT @A;
23
24 select * from train_user where User_ID='USR_001';

```

The Result Grid shows the following data:

User_ID	User_Type	Frame	Lname	Age	DOB	Phnoode	Street_No
USR_001	PSGR	Apt	Usal	52	1971-01-01	550001	10 Jangpeth

The Output tab shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	20:24:19	select * from train_user where User_ID='USR_001' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

## Successfully updating Age

The screenshot shows the MySQL Workbench interface with a query window containing the following SQL code:

```
15 set msg='Age updated Successfully';
16
17 END IF;
18 END;$$
19 DELIMITER ;
20
21 CALL age_update('USR_001','2002-08-26',@A);
22 SELECT @A;
```

The output window displays the results of the query execution:

#	Time	Action	Message	Duration / Fetch
1	20:24:19	select * from train_user where User_ID='USR_001' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
2	20:24:56	CREATE procedure age_update(IN UID varchar(30),IN DB date, OUT msg varchar(30)) BEGIN DECLARE ag...	0 row(s) affected	0.000 sec
3	20:25:04	CALL age_update('USR_001','2002-08-26',@A)	1 row(s) affected	0.094 sec
4	20:25:08	SELECT @A LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

The result grid shows the value of @A as 'Age updated Successfully'.

## Updated Age and DoB

The screenshot shows the MySQL Workbench interface with a query window containing the following SQL code:

```
17 END IF;
18 END;$$
19 DELIMITER ;
20
21 CALL age_update('USR_001','2002-08-26',@A);
22 SELECT @A;
23
24 select * from train_user where User_ID='USR_001';
```

The output window displays the results of the query execution:

#	Time	Action	Message	Duration / Fetch
1	20:24:19	select * from train_user where User_ID='USR_001' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
2	20:24:56	CREATE procedure age_update(IN UID varchar(30),IN DB date, OUT msg varchar(30)) BEGIN DECLARE ag...	0 row(s) affected	0.000 sec
3	20:25:04	CALL age_update('USR_001','2002-08-26',@A)	1 row(s) affected	0.094 sec
4	20:25:08	SELECT @A LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
5	20:25:33	select * from train_user where User_ID='USR_001' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

The result grid shows the details of the user with User\_ID 'USR\_001':

User_ID	User_Type	Fname	Lname	Age	DOB	Pncode	Street_No
USR_001	PSGR	Ajit	Ullal	20	2002-08-26	560001	10 Jangpath