S20230010225 DBMS LAB 9

Code and screenshots

Q1

```
1   delimiter //
2   create function q1(balance decimal) returns varchar(10)
3   deterministic
4   begin
5   declare level varchar(10);
6   if balance>= 900 then set level = "platinum";
7   elseif balance >=700 and balance <900 then set level = "gold";
8   else set level= "silver";
9   end if;
10  return level;
11  end//
12  delimiter ;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

accountnumber	Level
A-101	silver
A-102	silver
A-201	platinum
A-215	gold
A-217	gold
A-222	gold
A-305	silver

esult 1

Q2

```
16      delimiter //
17
18  •   create function q2(doj date) returns int
19      deterministic
20      begin
21      declare e int;
22      set e = timestampdiff(year, doj, curdate());
23      return e;
24      end//
25      delimiter ;
26
27  •   select *, q2(doj) as experience from employee;
28
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

emp_id	emp_name	emp_dept	emp_age	place	income	doj	experience
2505	peter	Finance	32	Newyork	100000	2002-08-25	22
2506	Mark	HR	32	California	120000	1980-03-25	44
2507	Donald	Finance	28	Arizona	100000	1995-12-26	28
2508	Obama	Management	35	Florida	500000	1990-10-30	33
2509	Linklon	HR	25	Georgia	25000	2008-08-08	16
2510	Kane	Sales	29	Alaska	30000	2000-01-01	24

Q3

```
31  create function q3() returns int
32  deterministic
33  begin
34      declare c int;
35      select count(*) INTO c
36      from borrower_relation b
37      join loan_relation l on b.loan_number = l.loan_number
38      WHERE l.amount > 1000;
39      RETURN c;
40  end //
41  delimiter ;
42
43  select q3() as countofpeople;
44
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

countofpeople
4

Result 3

Q4

```
47  create function q4() returns varchar(50)
48  deterministic
49  begin
50      declare richguy varchar(50);
51
52      select d.customername into richguy
53      from account_relation a
54      join depositor_relation d on a.accountnumber = d.account_number
55      order by a.balance desc
56      limit 1;
57
58      return richguy;
59  end //
60  delimiter ;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

topcustomer
Johnson

Error handling Q1

Creating a table

```
64  create table test_employee (
65      emp_id int not null,
66      emp_name varchar(100) not null,
67      primary key (emp_id)
68  );
69
70
71
```

Output

Action Output

#	Time	Action	Message
17	16:10:55	create function q4() returns varchar(50) deterministic begin declare richguy var...	Error Code: 1304. FUNCTION q4 already exists
18	16:34:46	create table test_employee (emp_id int not null, emp_name varchar(100) no...	0 row(s) affected

Creating procedure

```
74  create procedure q5(
75      pid int,
76      pname varchar(100)
77  )
78  begin
79  declare exit handler for sqlexception
80  begin
81  select 'Trying to populate a non-null column with null value';
82  end;
83  insert into test_employee(emp_id, emp_name)
84  values (pid, pname);
85  end //
86
87  delimiter ;
88
```

Output

Action Output

#	Time	Action	Message
18	16:34:46	create table test_employee (emp_id int not null, emp_name varchar(100) no...	0 row(s) affected
19	16:40:56	create procedure q5(pid int, pname varchar(100)) begin declare exit handl...	0 row(s) affected

Calling the procedure with null val

```
88
89  call q5(null, 'Shriansh');
90
91
92
93
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

Trying to populate a non-null column with null value
Trying to populate a non-null column with null value

esult 5

utput

Action Output

#	Time	Action	Message
19	16:40:56	create procedure q5(pid int, pname varchar(100)) begin declare exit handl...	0 row(s) affected
20	16:43:25	call q5(null, 'Shriansh')	1 row(s) returned

Code:

```
-- delimiter //
-- create function q1(balance decimal) returns varchar(10)
-- deterministic
-- begin
-- declare level varchar(10);
-- if balance>= 900 then set level = "platinum";
-- elseif balance >=700 and balance <900 then set level = "gold";
-- else set level= "silver";
-- end if;
-- return level;
-- end//
-- delimiter ;
-- select accountnumber, q1(balance) AS `Level`
-- from account_relation;

-- delimiter //

-- create function q2(doj date) returns int
-- deterministic
-- begin
-- declare e int;
-- set e = timestampdiff(year, doj, curdate());
-- return e;
-- end//
-- delimiter ;
```

```
-- select *, q2(doj) as experience from employee;



-- delimiter //
-- create function q3() returns int
-- deterministic
-- begin
--    declare c int;
--    select count(*) INTO c
--    from borrower_relation b
--    join loan_relation l on b.loan_number = l.loan_number
--    WHERE l.amount > 1000;
--    RETURN c;
-- end //
-- delimiter ;

-- select q3() as countofpeople;



-- delimiter //
-- create function q4() returns varchar(50)
-- deterministic
-- begin
--    declare richguy varchar(50);

--    select d.customername into richguy
--    from account_relation a
--    join depositor_relation d on a.accountnumber = d.account_number
```

```
--    order by a.balance desc
--    limit 1;
--    return richguy;
-- end //
-- delimiter ;

-- select q4() as topcustomer;



-- create table test_employee (
--    emp_id int not null,
--    emp_name varchar(100) not null,
--    primary key (emp_id)
-- );




-- delimiter //

-- create procedure q5(
--    pid int,
--    pname varchar(100)
-- )
-- begin
-- declare exit handler for sqlexception
-- begin
-- select 'Trying to populate a non-null column with null value';
-- end;
```

```
-- insert into test_employee(emp_id, emp_name)
-- values (pid, pname);
-- end //

-- delimiter ;

-- call q5(null, 'Shriansh');
```