IT252
Database Management System

Assignment VIII

NIRAJ NANDISH

191IT234 | IT252 | 28/03/2021

1.

```
create table STUDENT_MARKS(
STUDENT_ID int not null primary key,
NAME varchar(30),
SUB1 int default 0,
SUB2 int default 0,
SUB3 int default 0,
SUB4 int default 0,
SUB5 int default 0,
TOTAL int default 0,
PER_MARKS decimal(5,2) default 0.00,
GRADE varchar(20) default ''
);

insert into STUDENT_MARKS (STUDENT_ID, NAME) values
(1, 'Steven King'),
(2, 'Neena Kochhar'),
(3, 'Lex De Haan'),
(4, 'Alexander Hunold');

delimiter $$

create trigger upd_stud_tmarks
before update
on STUDENT_MARKS for each row
begin
set new.TOTAL = new.SUB1 + new.SUB2 + new.SUB3 + new.SUB4 + new.SUB5;
set new.PER_MARKS = new.TOTAL/5;
if new.PER_MARKS >= 90 then
set new.GRADE = "EXCELLENT";
elseif new.PER_MARKS >= 75 and new.PER_MARKS < 90 then
set new.GRADE = "VERY GOOD";
elseif new.PER_MARKS >= 60 and new.PER_MARKS < 75 then
set new.GRADE = "GOOD";
elseif new.PER_MARKS >= 40 and new.PER_MARKS < 60 then
set new.GRADE = "AVERAGE";
elseif new.PER_MARKS < 40 then
set new.GRADE = "NOT PROMOTED";
end if;
end $$

delimiter ;

update STUDENT_MARKS set SUB1 = 54, SUB2 = 69, SUB3 = 89, SUB4 = 87,
SUB5 = 59 where STUDENT_ID = 1

select * from STUDENT_MARKS;
```

```
mysql> delimiter $$
mysql> create trigger upd_stud_tmarks
    -> before update
    -> on STUDENT_MARKS for each row
    -> begin
    -> set new.TOTAL = new.SUB1 + new.SUB2 + new.SUB3 + new.SUB4 + new.SUB5;
    -> set new.PER_MARKS = new.TOTAL/5;
    -> if new.PER_MARKS >= 90 then
    -> set new.GRADE = "EXCELLENT";
    -> elseif new.PER_MARKS >= 75 and new.PER_MARKS < 90 then
    -> set new.GRADE = "VERY GOOD";
    -> elseif new.PER_MARKS >= 60 and new.PER_MARKS < 75 then
    -> set new.GRADE = "GOOD";
    -> elseif new.PER_MARKS >= 40 and new.PER_MARKS < 60 then
    -> set new.GRADE = "AVERAGE";
    -> elseif new.PER_MARKS < 40 then
    -> set new.GRADE = "NOT PROMOTED";
    -> end if;
    -> end $$
Query OK, 0 rows affected (0.82 sec)

mysql> delimiter ;
mysql> update STUDENT_MARKS set SUB1 = 54, SUB2 = 69, SUB3 = 89, SUB4 = 87, SUB5 = 59 where STUDENT_ID = 1
    -> ;
Query OK, 1 row affected (1.41 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from STUDENT_MARKS;
+------------+-----------------+------+------+------+------+------+-------+-----------+-------+
| STUDENT_ID | NAME            | SUB1 | SUB2 | SUB3 | SUB4 | SUB5 | TOTAL | PER_MARKS | GRADE |
+------------+-----------------+------+------+------+------+------+-------+-----------+-------+
|          1 | Steven King     |   54 |   69 |   89 |   87 |   59 |   358 |     71.60 | GOOD  |
|          2 | Neena Kochhar   |    0 |    0 |    0 |    0 |    0 |     0 |      0.00 |       |
|          3 | Lex De Haan     |    0 |    0 |    0 |    0 |    0 |     0 |      0.00 |       |
|          4 | Alexander Hunold |    0 |    0 |    0 |    0 |    0 |     0 |      0.00 |       |
+------------+-----------------+------+------+------+------+------+-------+-----------+-------+
4 rows in set (0.00 sec)
```

2.

```
delimiter $$

create trigger blog_after_insert
after insert
on blog for each row
begin
if new.deleted then
insert into audit (blog_id, changetype) values (new.id, 'DELETE');
else
insert into audit (blog_id, changetype) values (new.id, 'NEW');
end if;
end $$

create trigger blog_after_update
after update
on blog for each row
begin
if new.deleted then
```

```
insert into audit (blog_id, changetype) values (new.id, 'DELETE');
else
insert into audit (blog_id, changetype) values (new.id, 'EDIT');
end if;
end $$

delimiter ;

insert into blog values (1, 'Article One', 'Initial text', 0);
select * from audit;
update blog set content = 'Edited text' where id = 1;
select * from audit;
```

```
mysql> delimiter $$
mysql> create trigger blog_after_insert
    -> after insert
    -> on blog for each row
    -> begin
    -> if new.deleted then
    -> insert into audit (blog_id, changetype) values (new.id, 'DELETE');
    -> else
    -> insert into audit (blog_id, changetype) values (new.id, 'NEW');
    -> end if;
    -> end $$
Query OK, 0 rows affected (2.84 sec)

mysql> create trigger blog_after_update
    -> after update
    -> on blog for each row
    -> begin
    -> if new.deleted then
    -> insert into audit (blog_id, changetype) values (new.id, 'DELETE');
    -> else
    -> insert into audit (blog_id, changetype) values (new.id, 'EDIT');
    -> end if;
    -> end $$
Query OK, 0 rows affected (1.02 sec)

mysql> delimiter ;
mysql> insert into blog values (1, 'Article One', 'Initial text', 0);
Query OK, 1 row affected (0.79 sec)

mysql> select * fromm audit;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual
mysql> select * from audit;
+----+---------+------------+---------------------+
| id | blog_id | changetype | changetime          |
+----+---------+------------+---------------------+
|  1 |       1 | NEW        | 2021-03-28 19:12:43 |
+----+---------+------------+---------------------+
1 row in set (0.00 sec)

mysql> update blog set content = 'Edited text' where id = 1;
Query OK, 1 row affected (2.38 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from audit;
+----+---------+------------+---------------------+
| id | blog_id | changetype | changetime          |
+----+---------+------------+---------------------+
|  1 |       1 | NEW        | 2021-03-28 19:12:43 |
|  2 |       1 | EDIT       | 2021-03-28 19:13:27 |
+----+---------+------------+---------------------+
2 rows in set (0.00 sec)
```

3.

```
create table Customer_Bank(
Account_No int not null,
Customer_Name varchar(50),
Address varchar(75),
Branch_Code int,
Type_of_Transaction varchar(30),
Balance_Amount decimal(8,2)
);

delimiter $$
create trigger type before update on Customer_Bank for each row
begin
declare amount int default 0;
declare type_type varchar(50);
select new.Balance_Amount into amount;
select new.Type_of_Transaction into type_type;
if type_type = 'Credit' then
set new.Balance_Amount = amount + old.Balance_Amount;
elseif type_type = 'Debit' and amount<=50000 and
amount<old.Balance_Amount then
set new.Balance_Amount = old.Balance_Amount - amount;
end if;
end $$

delimiter ;

insert into Customer_Bank values (1, 'Niraj', 'Mysore', 12345,
'Credit', 50000);

update Customer_Bank set Balance_Amount = 65000 where Account_No = 1;

select * from Customer_Bank;

insert into Customer_Bank values (2, 'Tom', 'UK', 2345, 'Credit',
100000);

update Customer_Bank set Balance_Amount = 30000 where Account_No = 2;

select * from Customer_Bank;
```

```
mysql> delimiter $$
mysql> create trigger type before update on Customer_Bank for each row
    -> begin
    -> declare amount int default 0;
    -> declare type_type varchar(50);
    -> select new.Balance_Amount into amount;
    -> select new.Type_of_Transaction into type_type;
    -> if type_type = 'Credit' then
    -> set new.Balance_Amount = amount + old.Balance_Amount;
    -> elseif type_type = 'Debit' and amount<=50000 and amount<old.Balance_Amount then
    -> set new.Balance_Amount = old.Balance_Amount - amount;
    -> end if;
    -> end $$
Query OK, 0 rows affected (2.51 sec)
```

```
mysql> insert into Customer_Bank values (1, 'Niraj', 'Mysore', 12345, 'Credit', 50000);
Query OK, 1 row affected (2.86 sec)

mysql> select * from Customer_Bank;
+------------+---------------+---------+-------------+--------------------+----------------+
| Account_No | Customer_Name | Address | Branch_Code | Type_of_Transaction | Balance_Amount |
+------------+---------------+---------+-------------+--------------------+----------------+
|          1 | Niraj         | Mysore  |       12345 | Credit             |       50000.00 |
+------------+---------------+---------+-------------+--------------------+----------------+
1 row in set (0.00 sec)

mysql> update Customer_Bank set Balance_Amount = 65000 where Account_No = 1;
Query OK, 1 row affected (2.69 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from Customer_Bank;
+------------+---------------+---------+-------------+--------------------+----------------+
| Account_No | Customer_Name | Address | Branch_Code | Type_of_Transaction | Balance_Amount |
+------------+---------------+---------+-------------+--------------------+----------------+
|          1 | Niraj         | Mysore  |       12345 | Credit             |      115000.00 |
+------------+---------------+---------+-------------+--------------------+----------------+
1 row in set (0.00 sec)
```

```
mysql> insert into Customer_Bank values (2, 'Tom', 'UK', 2345, 'Credit', 100000);
Query OK, 1 row affected (2.63 sec)

mysql> update Customer_Bank set Balance_Amount = 30000 where Account_No = 2;
Query OK, 1 row affected (0.56 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from Customer_Bank;
+------------+---------------+---------+-------------+--------------------+----------------+
| Account_No | Customer_Name | Address | Branch_Code | Type_of_Transaction | Balance_Amount |
+------------+---------------+---------+-------------+--------------------+----------------+
|          1 | Niraj         | Mysore  |       12345 | Credit             |      115000.00 |
|          2 | Tom           | UK      |        2345 | Credit             |      130000.00 |
+------------+---------------+---------+-------------+--------------------+----------------+
2 rows in set (0.00 sec)
```

4.

```
delimiter $$

create procedure updatestuff()
begin
declare suba int;
declare subb int;
declare subc int;
declare subd int;
declare sube int;
declare id_no int;
declare total_marks int;
declare flag int default 0;
declare percentage decimal(19,2);
declare grade_student varchar(50);
declare marks cursor for select STUDENT_ID,SUB1,SUB2,SUB3,SUB4,SUB5
from STUDENT_MARKS;
declare continue handler for not found set flag = 1;
open marks;
getMarks : loop
fetch marks into id_no,suba,subb,subc,subd,sube;
set total_marks = suba + subb + subc + subd + sube ;
if flag = 1 then
leave getMarks;
end if;
set percentage = total_marks/5;
if percentage >= 90 then set grade_student = 'EXCELLENT';
elseif percentage >= 75 then set grade_student = 'VERY GOOD' ;
elseif percentage >=60 then set grade_student = 'GOOD' ;
elseif percentage >= 40 then set grade_student = 'AVERAGE' ;
else set grade_student = 'NOT PROMOTED';
end if;
update STUDENT_MARKS set TOTAL = total_marks,PER_MARKS = percentage,
GRADE = grade_student where STUDENT_ID = id_no;
end loop getMarks;
close marks;
end $$

delimiter ;
```

```
mysql> delimiter $$
mysql> Create procedure updatestuff()
    -> Begin
    -> Declare a int;
    -> Declare b int;
    -> Declare c int;
    -> Declare d int;
    -> Declare e int;
    -> Declare id_no int;
    -> Declare total_marks int;
    -> Declare flag int default 0;
    -> Declare perc decimal(19,2);
    -> Declare grade_student varchar(50);
    -> Declare cursor1 cursor for select STUDENT_ID,SUB1,SUB2,SUB3,SUB4,SUB5 from STUDENT_MARKS;
    -> Declare continue handler for not found set flag = 1;
    -> Open cursor1;
    -> getMarks : loop
    -> Fetch cursor1 into id_no,a,b,c,d,e;
    -> Set total_marks = a+b+c+d+e ;
    -> If flag = 1 then
    -> Leave getMarks;
    -> End if;
    -> Set perc = total_marks/5;
    -> If perc >= 90 then set grade_student = 'EXCELLENT';
    -> elseif perc >= 75 then set grade_student = 'VERY GOOD' ;
    -> elseif perc>=60 then set grade_student = 'GOOD' ;
    -> elseif perc>= 40 then set grade_student = 'AVERAGE' ;
    -> else set grade_student = 'NOT PROMOTED';
    -> end if;
    -> Update STUDENT_MARKS set TOTAL = total_marks,PER_MARKS = perc, GRADE = grade_student where STUDENT_ID = id_no;
    -> end loop getMarks;
    -> close cursor1;
    -> end; $$
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> select * from STUDENT_MARKS;
+------------+-----------------+------+------+------+------+------+-------+-----------+-------+
| STUDENT_ID | NAME            | SUB1 | SUB2 | SUB3 | SUB4 | SUB5 | TOTAL | PER_MARKS | GRADE |
+------------+-----------------+------+------+------+------+------+-------+-----------+-------+
|          1 | Steve King      |   30 |   30 |   30 |   30 |   30 |  NULL |      NULL | NULL  |
|          2 | Neena Kochhar   |   40 |   40 |   40 |   40 |   40 |  NULL |      NULL | NULL  |
|          3 | Lex De Haan     |   60 |   60 |   60 |   60 |   60 |  NULL |      NULL | NULL  |
|          4 | Alexander Hunold |   50 |   50 |   50 |   50 |   50 |  NULL |      NULL | NULL  |
+------------+-----------------+------+------+------+------+------+-------+-----------+-------+
4 rows in set (0.00 sec)

mysql> call updatestuff;
Query OK, 0 rows affected (0.02 sec)

mysql> select * from STUDENT_MARKS;
+------------+-----------------+------+------+------+------+------+-------+-----------+--------------+
| STUDENT_ID | NAME            | SUB1 | SUB2 | SUB3 | SUB4 | SUB5 | TOTAL | PER_MARKS | GRADE        |
+------------+-----------------+------+------+------+------+------+-------+-----------+--------------+
|          1 | Steve King      |   30 |   30 |   30 |   30 |   30 |   150 |   30.0000 | NOT PROMOTED |
|          2 | Neena Kochhar   |   40 |   40 |   40 |   40 |   40 |   200 |   40.0000 | AVERAGE      |
|          3 | Lex De Haan     |   60 |   60 |   60 |   60 |   60 |   300 |   60.0000 | GOOD         |
|          4 | Alexander Hunold |   50 |   50 |   50 |   50 |   50 |   250 |   50.0000 | AVERAGE      |
+------------+-----------------+------+------+------+------+------+-------+-----------+--------------+
4 rows in set (0.00 sec)
```

5.

```
delimiter $$

create procedure add_level()
begin
declare flag int default 0;
declare acc_no int;
declare amount double;
declare cal_level varchar(50);
declare curno cursor for select Account_no from Customer_Bank;
declare curamt cursor for select Balance_Amount from Customer_Bank;
declare continue handler for not found set flag = 1;
open curno; open curamt;
changelevel: loop
fetch curamt into amount;
fetch curno into acc_no;
if flag=1 then
leave changelevel;
end if;
if amount>=100000 then set cal_level = 'PLATINUM';
elseif amount>=50000 and amount<100000 then
set cal_level='GOLD';
elseif amount<50000 then set cal_level='SILVER';
end if;
update Customer_Bank set Customer_Level=cal_level
where Account_no = acc_no;
end loop changelevel;
close curno;
close curamt;
end $$

delimiter ;
```

```
mysql> delimiter $$
mysql> create procedure add_level()
    -> begin
    -> declare flag int default 0;
    -> declare acc_no int;
    -> declare amount double;
    -> declare cal_level varchar(50);
    -> declare curno cursor for select Account_no from Customer_Bank;
    -> declare curamt cursor for select Balance_Amount from Customer_Bank;
    -> declare continue handler for not found set flag = 1;
    -> open curno; open curamt;
    -> changelevel: loop
    -> fetch curamt into amount;
    -> fetch curno into acc_no;
    -> if flag=1 then
    -> leave changelevel;
    -> end if;
    -> if amount>=100000 then set cal_level = 'PLATINUM';
    -> elseif amount>=50000 and amount<100000 then
    -> set cal_level='GOLD';
    -> elseif amount<50000 then set cal_level='SILVER';
    -> end if;
    -> update Customer_Bank set Customer_Level=cal_level
    -> where Account_no = acc_no;
    -> end loop changelevel;
    -> close curno;
    -> close curamt;
    -> end $$
Query OK, 0 rows affected (3.02 sec)

mysql> delimiter ;
mysql> select * from Customer_Bank;
+------------+---------------+---------+-------------+---------------------+----------------+
| Account_No | Customer_Name | Address | Branch_Code | Type_of_Transaction | Balance_Amount |
+------------+---------------+---------+-------------+---------------------+----------------+
|          1 | Niraj         | Mysore  |       12345 | Credit              |      115000.00 |
|          2 | Tom           | UK      |        2345 | Credit              |      130000.00 |
+------------+---------------+---------+-------------+---------------------+----------------+
2 rows in set (0.00 sec)

mysql> call add_level();
ERROR 1054 (42S22): Unknown column 'Customer_Level' in 'field list'
mysql> alter table Customer_Bank add Customer_Level varchar(50);
Query OK, 0 rows affected (4.81 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> call add_level();
Query OK, 0 rows affected (1.09 sec)

mysql> select * from Customer_Bank;
+------------+---------------+---------+-------------+---------------------+----------------+----------------+
| Account_No | Customer_Name | Address | Branch_Code | Type_of_Transaction | Balance_Amount | Customer_Level |
+------------+---------------+---------+-------------+---------------------+----------------+----------------+
|          1 | Niraj         | Mysore  |       12345 | Credit              |      230000.00 | PLATINUM       |
|          2 | Tom           | UK      |        2345 | Credit              |      260000.00 | PLATINUM       |
+------------+---------------+---------+-------------+---------------------+----------------+----------------+
2 rows in set (0.00 sec)
```