Department of Computer Science

Gujarat University

Certificate

Roll No: 36 *Seat No:* _____

This is to certify that Mr./Ms. PREKSHA K. SHETH *student of MCA Semester – II has duly completed his/her term work for the semester ending in June 2020, in the subject of* RELATIONAL DATABASE MANAGEMENT SYSTEM - II *towards partial fulfillment of his/her Degree of Masters in Computer Applications.*

Date of Submission *Internal Faculty*

1st - JULY - 2020

Head of Department

Department Of Computer Science
Rollwala Computer Centre
Gujarat University

MCA - II

Subject: - RDBMS - II

Name: - Parkshan K Sheth

Roll No.: - 36 **Exam Seat No.: -** _____

Sr. No.	Contents	Pg. No	Date	Signature
	~~Assignment - 1~~			
1	System Catalog	1-58	12/3/2020	
2	Query processing			
3	optimization			
4	Evaluation & Transaction.			18/3/2020
	Assigament - 2	1-11	1/7/2020	
5	Transaction mangement & Recovery.			

Assignment - 2

Q.1 Discuss 4 properties of transaction to ensure Intigrity of data CACID2.

1) → Consistancy & Isolation:

→ Users are responsible for ensuring transaction consistancy. That is, users who submits a transactions must ensure that, when run to completion by itself against a 'consistance' database instance, the transaction with leave the database in a 'consistant' state.

→ The isolation property is ensured by guaranteeing that, even though actions of several transactions might be interleaved, the net effect is identical to executing all transaction one after other in same serial order

2) Atomacity & Durability:

→ Since, users thinks of transactions is being atomic, a transaction that in interrupted in inconsistance the middle may leave the

database in an inconsistance state.

→ Therefore a DBMS must find a easy to remove the effects of partial transactions from the DB.

→ That is must ensure transaction atomicity : either all of a transaction's actions are carried out or none one.

→ A DBMS ensures transaction atomicity by undoing the actions of incomplete transactions To be able to do this, the DBMS maintains HD record, called the log of all writes the database

→ The log is also used to ensure durability: if the system crashes before the changes by a completed transaction are written to disk the log is used to remember & restore these changes from the system restarts

→ The DBMS Component that ensures atomacity & durability, called recovery manager.

Q.2 Discuss two important assumptions with respect to transactions.

→ 1. Transaction interact with each other only via database read & write operations: for eg. they are not allowed to exchange messages.

2. A database is a filed collection of independent objects when objects are added to or deleted from a database or there are relationship b/w database objects that are used to exploit for performance some additional issues arises.

→ If the first assumption is isolated, DBMS has no way to detect or prevent inconsistence cause by such external interactions b/w transactions. & it is upto the writer of the application to ensure that the program is well behaved.

Q.3 Discuss Thomas - write Rule:

→ If TS(T) < WTS(O), the current write action has, in effect, been made obsolete by the most recent write of O, which follows the current write according to the timestamp ordering.

→ We can think of T's write action as if it had occurred immediately before the most recent write of O & was never read by anyone.

→ If the Thomas write Rule is not used, that is, T is aborted in case (c), the timestamp protocols like 2PL, allows only Conflict Serializable Schedule.

→ If the Thomas write Rule is used, some Schedules are permitted that are not Conflict Serializable.

T_1	T_2
R(A)	
	W(A)
	Commit.
W(A)	
Commit	

A Serializable Schedule That is Not Conflict serializable.

T_1	T_2
R(A)	
	Commit
W(A)	
Commit	

A Conflict Serializable Schedule.

Teacher's Signature :

Q.4x Discuss Three phases of ARIES Recovery Algorithm.

→ ARIES is a recovery algorithm designed to work with a steal, no-force approach.

→ When the recovery manager is invoked after a crash, restart proceeds in 3 phases:

1) Analysis : It Identifies dirty pages in the buffer pool & active transactions at the time of the crash

2) Redo : Repeats all actions, starting from an appropriate point in the log & restores the database state to what it was at the time of the crash.

3) Undo : Undoes the actions of transactions that did not Commit, so that the database reflects only the actions of Commited transactions.

LSN	Log
10	Update: T_1 writes P_5
20	Update: T_2 writes P_3
30	T_2 Commit
40	T_2 end
50	Update: T_3 writes P_1
60	Update: T_3 writes P_3

Teacher's Signature:

→ Consider the Simple Execution history illustrated in fig.

→ When the system is restarted, the Analysis phase identifies T1 & T3 as transactions active at the time of the crash & therefore to be undone; T2 as a commited transaction, and all its actions therefore to be written to disk. & P1, P3 & P5 as potentially dirty pages.

→ All the updates are applied in the order shown during the Redo phase.

→ finally, the actions of T1 & T3 are undone in Reverse order during the Undo phase; That is T3's write of P3's undone. T3's writes of P2 is undone & then T1's write of P5 undone.

Q-5 Short note on CLR.

→ A Comparision log record (CLR) is written just before the changed recorded in an update log record.

→ The CLR C also contains a field called UNDONEXTLSN, which is the LSN of the Next record that is to be undone for the transaction that wrote update record. See this field in C is set the value of PREVLSN in U.

→ Unlike an update log record, a CLR describes an action that will never be undone, that is we never undo an undo action. The reason is simple: An update log record disaibes a change made by a transaction during normal execution & the transaction may subsequently be aborted whereas a CLR describes an action taken to rollback to transaction for which the decision to abort has already been made. Therefore the transaction must be rolled back, and the

Undo action described by the CLR is definitely required.

→ The number of CLRs that can be written during undo is no more than the no. of update log records for active transactions at the time of the crash.

→ A CLR may be written to stable storage, but the undo action it describe may not yet been written to disk when the system crashes again. In this case, the undo action described in the CLR is applied during the Redo phases just like action described in updated log records.

→ For there reason, a CLR contains the information needed to reapply, or redo the changes described but not to reverse it.

Q.6 Define the transaction reference to mysql

→ A Transaction is automatically started as a user executes a statement that access either the database or the catalogs, such as a SELECT query or UPDATE Command.

→ In SQL 1999 two new features are added

→ 1st featured called SAVE point, allows us to identify a point in a transaction & selectively rollback operations carried out after this point.

→ In a log running transactions, we may want to define a series of savepoints. The Savepoint Command allows us to give each savepoint name.

SAVEPOINT (savepoint name)

→ A Subsequent Rollback command can specify the savepoint to rollback to that savepoint.

ROLLBACK TO SAVEPOINT (savepoint name);

Teacher's Signature :

2.7 Discuss the role of DBA in security with reference to database.

→ The DBA plays an important role in enforcing the security related aspects of a database design.

→ The DBA has a special Account which we call the system account, & is responsible for the overall security of the system.

→ In particular, the DBA deals with the following.

i) Creating New Account:

→ Each new Account user or group of users must be assigned an authorization ID & PASSWORD Note that application programs that access the database have the same authorization ID as the user executing the program.

ii) mandatory Control issues:

→ If the DBMS supports mandatory control issues customized System for applications with very high security requirements provide, such

Support the DBP must assign security classes to each database object & assign security clearance to each authorization ID in a accordance with the chosen security policy.

→ The DBA is also responsible for maintaining audit trial, which is essentially the log of updates with authorization ID added each log entry

→ The log is just a mirror extension of the log mechanism used to recover from crashes.

D E P A R T M E N T O F C O M P U T E R S C I E N C E

R O L L W A L A C O M P U T E R C E N T R E

G U J A R A T U N I V E R S I T Y

M.C.A. – II

R O L L NO : 36

N A M E : Preksha Sheth

S U B J E C T : DBMS-II

NO.	TITLE	PAGE NO.	DATE	SIGN
	ASSIGNMENT 1:			
01	Write PL/SQL procedures to delete rows from emp table		01/07/20	
	based on a parameter and display no. of employees			
	deleted.			
02	Write PL/SQL procedures to perform withdraw and		01/07/20	
	deposit operations.			
03	Write Pl/SQL procedures to perform withdraw and		01/07/20	
	transfer operations.			
04	Write Pl/SQL procedures to retrieve all the details of the		01/07/20	
	employees with grade 5 and to apply pattern matching			
	through the procedure.			
05	Write functions named SUMMATION() to add 2 numbers		01/07/20	
	and Other to concate 2 strings.			
06	Write 3 procedures respectively to check if the number is		01/07/20	
	>0, to check if the entered date is > sysdate and to check if			
	the name entered is in uppercase. And write one final to			
	call procedure them as a,b and c respectively.			
07	Write a procedure to display first n fibonacci numbers.		01/07/20	
08	Write a procedure to find simple interest.		01/07/20	

D E P A R T M E N T O F C O M P U T E R S C I E N C E

R O L L W A L A C O M P U T E R C E N T R E

G U J A R A T U N I V E R S I T Y

M.C.A. – II

R O L L NO : 36

N A M E : Preksha Sheth

S U B J E C T : DBMS-II

09	Write a procedure to reverse entered number.		01/07/20	
10	Write a procedure and find its equivalent roman value.		01/07/20	
11	Write a program to enter a number and find addition of		01/07/20	
	each digit of that number using function.			
12	Write a program to input your birthdate and should return		01/07/20	
	your age in year, month and days.			
13	Write a program to reverse the string.		01/07/20	

14	Write a program to convert the string into toggle case.		01/07/20	
15	Write a PL/SQL block to convert given numbers into text		01/07/20	
	Words.			
	ASSIGNMENT 2:			
	:GENERAL PL/SQL BLOCKS:			
01	WAP to input two numbers and find out what is the		01/07/20	
	Output of all arithmetic operations.			
02	WAP to input rollno and three subject marks. Find out		01/07/20	
	total, percentage, result and grade for the student from			
	the entered data.			
03	WAP to print first 10 odd numbers using for loop.		01/07/20	
04	WAP to print prime numbers upto 10 using while loop.		01/07/20	

D E P A R T M E N T O F C O M P U T E R S C I E N C E

R O L L W A L A C O M P U T E R C E N T R E

G U J A R A T U N I V E R S I T Y

M.C.A. – II

R O L L NO : 36

N A M E : Preksha Sheth

S U B J E C T : DBMS-II

05	WAP to input three nos and find out maximum and		01/07/20	
	minimum from it.			
06	WAP to input empno from keybord. Check whether		01/07/20	
	inputed empno exist in emp table or not. If not give error			
	message otherwise display name and salary of that			
	employee.			
07	WAP to insert record in Customer table		01/07/20	
	:FUNCTIONS:			
08	WAF which accepts the name from user and returns the		01/07/20	
	length of that name.			
09	WAF which accepts one number and return TRUE if no		01/07/20	
	is prime and return FALSE if No. is not prime.			
10	Write a function which accepts the department no and		01/07/20	
	returns maximum salary of that Department.			

11	Write a function to display whether the entered (User		01/07/20	
	Input) employee no exists or not.			
12	WAF which accepts one no and returns that no+100.		01/07/20	
13	WAF which accepts the empno. and raises the salary slab		01/07/20	
	Wise.			

D E P A R T M E N T O F C O M P U T E R S C I E N C E

R O L L W A L A C O M P U T E R C E N T R E

G U J A R A T U N I V E R S I T Y

M.C.A. – II

R O L L NO : 36

N A M E : Preksha Sheth

S U B J E C T : DBMS-II

14	WAF which accepts the empno and returns the experience		01/07/20	
	in years.			
	:PROCEDURES:			
15	Write a procedure which accepts the empno and returns		01/07/20	
	the associated empname.			
16	WAP which accepts the student rollno and returns the		01/07/20	
	name,city and marks of all the subjects of that student.			
17	WAP which accepts the name from the user. Return case		01/07/20	
	Of the name i.e. UPPER, LOWER or MIXCASE.			
18	WAP which accepts the student rollno and returns the		01/07/20	
	highest percent and name of that student to the calling			
	block.			
19	WAP which accepts the date of joining for specific		01/07/20	
	employee and returns the years of experience along with			
	its name.			
20	WAP which accepts the student roll no and returns the		01/07/20	
	result (i.e. first class, second class, third class or fail).			

	:CURSORS:			

D E P A R T M E N T O F C O M P U T E R S C I E N C E

R O L L W A L A C O M P U T E R C E N T R E

G U J A R A T U N I V E R S I T Y

M.C.A. – II

R O L L NO : 36

N A M E : Preksha Sheth

S U B J E C T : DBMS-II

21	Create a cursor for the emp table. Produce the output in		01/07/20	
	following format: {empname} employee working in			
	department {deptno} earns Rs. {salary}.			

22	Create a cursor for updating the salary of emp working in		01/07/20	
	deptno 10 by 20%. And display no. of rows affected.			
23	WAP that will display the name, department and salary of		01/07/20	
	the first 10 employees getting the highest salary			
24	WAP using parameterized cursor to display all the		01/07/20	
	information of employee living in specified city.			
25	WAP which display the sum of salary department wise.		01/07/20	
26	Create a cursor to generate different two tables from one		01/07/20	
	master table.			
	ASSIGNMENT 3:			

	:TRIGGERS:			
1	Write a Trigger that stores the old data table of student		01/07/20	
	table in student_backup while updating the student table.			

D E P A R T M E N T O F C O M P U T E R S C I E N C E

R O L L W A L A C O M P U T E R C E N T R E

G U J A R A T U N I V E R S I T Y

M.C.A. – II

R O L L NO : 36

N A M E : Preksha Sheth

S U B J E C T : DBMS-II

2	Write a trigger, that ensures the empno of emp table is in		01/07/20	
	format 'E00001' .If not, than complete empno with this			
	format before inserting into the employee table.			
3	Write a trigger which checks the age of employee while		01/07/20	
	inserting the record in emp table.			
4	Write a trigger which converts the employee name in		01/07/20	
	upper case if it is inserted in any other case.			
5	WAT that stores the data of emp table in emp_backup		01/07/20	
	table for every delete operation and store the old data for			
	every update operation.			
6	WAT which display the message 'Updating','Deleting' or		01/07/20	
	'Inserting' when Update, Delete or Insert operation is			
	performed on the emp table respectively.			
7	WAT which generate an error if any user try to delete from		01/07/20	
	product_master table on weekends.			
8	WAT which inserts the value of client_no in the		01/07/20	
	client_master table whenever user tries to insert data in			
	the emp table			
9	WAT to calculate the Income Tax amount and insert it in		01/07/20	
	emp table. Income Tax calculated slab wise.			
10	Write a trigger which updates the sales value if customer		01/07/20	

D E P A R T M E N T O F C O M P U T E R S C I E N C E

R O L L W A L A C O M P U T E R C E N T R E

G U J A R A T U N I V E R S I T Y

M.C.A. – II

R O L L NO : 36

N A M E : Preksha Sheth

S U B J E C T : DBMS-II

	already exists else create new entry of customer.			
11	WAT to update if the customer is updating the sales value		01/07/20	
	by incrementing the sale_value field.			
12	If the customer is deleting, WAT to update the sales value		01/07/20	
	by decrementing the sale_value field.			
13	Create after update trigger on account table.		01/07/20	
14	Create before update trigger validation on account table.		01/07/20	
15	write a example to create a sales table which provides		01/07/20	
	free shipping on orders above 500.			
	ASSIGNMENT 4:			
	:TRANSACTION PROGRAM:			
01	Create a procedure to commence a transaction using start		01/07/20	
	transaction.			
02	Create procedure to commence a transaction using auto		01/07/20	
	commit.			
03	Create a procedure which display use of savepoint with a.		01/07/20	
	Transaction.			

****
****
Name   : Preksha Sheth
Roll No. : 36
Class   : MCA-II
Subject : RDBMS-II(PL/SQL)

****
****

ASSIGNMENT - 1

****
****
Question 1(a) : Develop the following:
a.Create a procedure that deletes rows from the emp table.
It should accept 1 parameter, job; only delete the employee's with that job.
Display how many employees were deleted. Write a PL/SQL block to invoke the
procedure.

****
****

```
MariaDB [preksha]> select * from emp;
+--------+----------+-----------+------------+
| emp_id | emp_name | emp_job   | emp_salary |
+--------+----------+-----------+------------+
|      1 | Preksha  | Developer |      20000 |
|      2 | Prerak   | DBA       |      18000 |
|      3 | Dhruvin  | DBA       |      15000 |
|      4 | Aman     | Developer |      25000 |
|      5 | Mahi     | Designer  |      15000 |
|      6 | Jeet     | Tester    |      12000 |
|      7 | Abhi     | Designer  |      23000 |
+--------+----------+-----------+------------+
7 rows in set (0.00 sec)



MariaDB [preksha]> delimiter $$
MariaDB [preksha]> create procedure delete_emp(in job_name varchar(50))
    -> begin
    -> delete from emp where emp_job = job_name;
    -> select ROW_COUNT() as 'Deleated Rows';
    -> end $$
Query OK, 0 rows affected (0.12 sec)



MariaDB [preksha]> set @job_name = 'Tester';
    -> $$
Query OK, 0 rows affected (0.00 sec)



MariaDB [preksha]> call delete_emp(@job_name) $$
+---------------+
| Deleated Rows |
+---------------+
|             1 |
+---------------+
1 row in set (0.17 sec)

Query OK, 0 rows affected (0.19 sec)
```

```
MariaDB [preksha]> select * from emp $$
+--------+----------+-----------+------------+
| emp_id | emp_name | emp_job   | emp_salary |
+--------+----------+-----------+------------+
|      1 | Preksha  | Developer |      20000 |
|      2 | Prerak   | DBA       |      18000 |
|      3 | Dhruvin  | DBA       |      15000 |
|      4 | Aman     | Developer |      25000 |
|      5 | Mahi     | Designer  |      15000 |
|      7 | Abhi     | Designer  |      23000 |
+--------+----------+-----------+------------+
6 rows in set (0.00 sec)
```

****

****

Question 1 (b) : Change the above procedure so that it returns the number of Employees removed via an OUT parameter. Write a PL/SQL block to invoke the procedure and display how many employees were deleted.

****

****

```
MariaDB [preksha]> select * from emp $$
+--------+----------+-----------+------------+
| emp_id | emp_name | emp_job   | emp_salary |
+--------+----------+-----------+------------+
|      1 | Preksha  | Developer |      20000 |
|      2 | Prerak   | DBA       |      18000 |
|      3 | Dhruvin  | DBA       |      15000 |
|      4 | Aman     | Developer |      25000 |
|      5 | Mahi     | Designer  |      15000 |
|      7 | Abhi     | Designer  |      23000 |
+--------+----------+-----------+------------+
6 rows in set (0.00 sec)

MariaDB [preksha]>  create or replace procedure del_emp1(in des varchar(20) , out
no int)
  ->      begin
  ->      select count(*) into no from emp where job = des ;
  ->      if(no > 1 ) then
  ->          delete from emp where job = des;
  ->               select concat(no,"  Employees are deleted ") as "Message";
  ->      else
  ->               select concat("Invalid job") As "Message";
  ->      end if;
  ->      end $$
Query OK, 0 rows affected (0.17)

MariaDB [preksha]> call del_emp1("accountant",@no)$$
```

```
+----------------------+
| Message              |
+----------------------+
| Invalid job          |
+----------------------+
1 row in set (0.14 sec)
```

****
****

Question 2 : Create a table having the following structure Accounts(Account_id, branch_name, amount_balance)

(a). Write a PL/SQL procedure to perform withdraw operation that only permits a withdrawal
if there sufficient funds in the account. The procedure should take Account_id and withdrawal
amount as input.

****
****

```
MariaDB [preksha]> select * from accounts;
+--------+-------------+-----------+
| acc_id | Branch_Name | Balance   |
+--------+-------------+-----------+
|      1 | Sabarmati   | 200000.00 |
|      2 | Chandkheda  | 330000.00 |
|      3 | Motera      | 123000.00 |
|      4 | sabarmati   | 205000.00 |
|      5 | Motera      | 500000.00 |
|      6 | jantanagar  | 300000.00 |
+--------+-------------+-----------+
6 rows in set (0.00 sec)


MariaDB [preksha]> create procedure withdraw_amt(in id int(10),in w_amt
decimal(8,2))
    ->      begin
    ->      declare bal decimal(8,2);
    ->      select balance into bal from accounts where acc_id = id;
    ->      if(bal > w_amt)
    ->      then
    ->      set bal = bal - w_amt;
    ->      update accounts set balance = bal where acc_id = id;
    ->      else
    ->      select 'Balance is not Sufficient' as 'Warning';
    ->      end if;
    ->      end $$
Query OK, 0 rows affected (0.06 sec)



MariaDB [preksha]> set @id = 5;
    -> set @w_amt = 100000;
    -> $$
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [preksha]> call withdraw_amt(@id,@w_amt)
    -> $$
Query OK, 1 row affected (0.17 sec)



MariaDB [preksha]> select * from accounts $$
+--------+-------------+-----------+
| acc_id | Branch_Name | Balance   |
+--------+-------------+-----------+
|      1 | Sabarmati   | 200000.00 |
|      2 | Chandkheda  | 330000.00 |
|      3 | Motera      | 123000.00 |
|      4 | sabarmati   | 205000.00 |
|      5 | Motera      | 400000.00 |
|      6 | jantanagar  | 300000.00 |
+--------+-------------+-----------+
6 rows in set (0.00 sec)


MariaDB [preksha]> call withdraw_amt(5,100000000)$$
+--------------------------+
| Warning                  |
+--------------------------+
| Balance is not Sufficient |
+--------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected, 1 warning (0.01 sec)
********************************************************************************
********************************************************************************
(b).  Write a procedure to deposit money into someone's account.
      The procedure should accept account_id and deposit amount.

********************************************************************************
********************************************************************************
MariaDB [preksha]> create procedure deposite(in id int , in amt decimal(8,2))
    ->      begin
    ->      declare bal decimal(8,2);
    ->      select balance into bal from accounts where acc_id = id;
    ->      select bal as 'Previous Balance';
    ->      set bal = bal + amt;
    ->      update accounts set balance = bal where acc_id = id;
    ->      select bal 'Updated Balance';
    ->      end $$
Query OK, 0 rows affected (0.13 sec)



MariaDB [preksha]> set @id = 2;
    -> set @amt = 10000;
    -> $$
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)


MariaDB [preksha]> call deposite(@id,@amt) $$
+------------------+
| Previous Balance |
+------------------+
|        340000.00 |
+------------------+
1 row in set (0.00 sec)
```

```
+-----------------+
| Updated Balance |
+-----------------+
|       350000.00 |
+-----------------+
1 row in set (0.13 sec)

Query OK, 0 rows affected (0.15 sec)
```

****
****

Question 3 : Write a procedure to transfer money from one person's account to another . The procedure should table two account_id's one for giver and one for receiver and the amount to be transferred.

****
****

```
MariaDB [preksha]> select * from accounts $$
+--------+-------------+-----------+
| acc_id | Branch_Name | Balance   |
+--------+-------------+-----------+
|      1 | Sabarmati   | 200000.00 |
|      2 | Chandkheda  | 330000.00 |
|      3 | Motera      | 123000.00 |
|      4 | sabarmati   | 205000.00 |
|      5 | Motera      | 400000.00 |
|      6 | jantanagar  | 300000.00 |
+--------+-------------+-----------+
6 rows in set (0.00 sec)
```

```
MariaDB [preksha]> create or replace procedure transfer(in g_account int,in
t_account int , in amount decimal(8,2))      begin
-> declare s_account int;
-> declare  r_account int;
-> Declare s_amount  decimal(8,2);
-> declare r_amount decimal(8,2);
->  select ac_id into s_account from account where ac_id = g_account;

->  select balance into s_amount  from account where ac_id = g_account;

->  select ac_id into r_account from account where ac_id = t_account;

->  select balance into r_amount from account where ac_id = t_account;
->if s_account IS NULL then

->      select 'Account not found !' as 'warning';
->elseif r_account IS NULL then

->      select 'Reciever Account is not found !' as 'warning';
->elseif amount IS NULL then

 ->     select 'You can not transfer the amount' as 'WARNING';

 -> elseif amount > s_amount then

->      select 'Insufficient balance in sender account ' as 'WARNING';

->elseif amount < 0  then

->      select 'amount can not negative' as 'WARNING';

->else

->     set s_amount = s_amount - amount;
```

```
->     update account set balance = s_amount where ac_id = g_account;
->     set r_amount = r_amount + amount;
->     update account set balance = r_amount where ac_id = t_account;
->    select 'Amount transferred successfully' as 'MESSAGE';
->end if;
->end if;
-> end $$
```

Output :

```
MariaDB [preksha]> call transfer(5,4,200000)$$
+-----------------------------------------------------------------------+
|Message                                                                |
+-----------------------------------------------------------------------+
|Amount transferred successfully                                        |
+-----------------------------------------------------------------------+
1 row in set (0.09 sec)


MariaDB [preksha]> call transfer(4,5,400000)$$
+-----------------------------------------------------------------------+
|WARNING                                                                |
+-----------------------------------------------------------------------+
| Insufficient balance in sender account                                |
+-----------------------------------------------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.02 sec)
```

```
********************************************************************************
********************************************************************************
Question 4 b. :  Write a Pl/SQL block to apply pattern matching through the
procedure. (e.g if user supplies Salall the departments containing Sal should be
displayed)`
********************************************************************************
********************************************************************************
MariaDB [preksha]> create or replace procedure pattern(in pttrn varchar(20))
    -> begin
    ->          select * from employee where job like pttrn;
    -> end $$
Query OK, 0 rows affected (0.20 sec)
```

```
MariaDB [preksha]> call pattern("%DB%")$$

+--------+----------+-----------+------------+
| emp_id | emp_name | emp_job   | emp_salary |
+--------+----------+-----------+------------+
|      2 | Prerak   | DBA       |      18000 |
|      3 | Dhruvin  | DBA       |      15000 |
+--------+----------+-----------+------------+
```

****
****

Question 5 : Write a functions named SUMMATION(), develop two labels in functions, The first one will accept accept two number arguments and return the addition of them.

****
****

```
MariaDB [preksha]> create function summation(no1 int ,no2 int)
    -> returns int
    -> begin
    -> declare sum int;
    -> set sum = no1 + no2;
    -> return sum;
    -> end $$
Query OK, 0 rows affected (0.10 sec)

MariaDB [preksha]> select summation(10,20)$$
+--------------------------------------+
| summation(10,20)                     |
+--------------------------------------+
|                30                    |
+--------------------------------------+
1 row in set (0.01 sec)
```

****
****

B. In the second function accept two character type arguments and return concatenated string

****
****

```
MariaDB [preksha]> create function concate(str1 varchar(50), str2 varchar(50))
    -> returns varchar(100)
    ->      begin
    ->      declare rslt varchar(100);
    ->      set rslt = concat (str1 , " " , str2);
    ->      return rslt ;
    ->      end $$
Query OK, 0 rows affected (0.06 sec)
```

```
MariaDB [preksha]> select concate("Preksha" , "Sheth")$$
+--------------------------------------+
| concate("Preksha" , "Sheth")         |
+--------------------------------------+
| Preksha Sheth                        |
+--------------------------------------+
1 row in set (0.00 sec)



********************************************************************************
********************************************************************************

Question 6 :  Create three different procedures and one final procedure to call
              them as follows
         (a). First procedure check for the number is > 0 or not.
********************************************************************************
********************************************************************************
MariaDB [preksha]> create procedure check_no(in num int)
    ->      begin
    ->      if num > 0 then
    ->       select 'Number is Greater Than 0' as 'Check_Number';
    ->      elseif num < 0 then
    ->       select 'Number is Less Than 0' as 'Check_Number';
    ->      else
    ->       select 'Number is 0' as 'Check_Number';
    ->      end if;
    ->      end $$
Query OK, 0 rows affected (0.05 sec)


MariaDB [preksha]> set @num = 10 $$
Query OK, 0 rows affected (0.00 sec)


MariaDB [preksha]> call check_no(@num) $$
+--------------------------+
| Check_Number             |
+--------------------------+
| Number is Greater Than 0 |
+--------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)


MariaDB [preksha]> set @num = -12 $$
Query OK, 0 rows affected (0.00 sec)


MariaDB [preksha]> call check_no(@num) $$
+-----------------------+
| Check_Number          |
+-----------------------+
| Number is Less Than 0 |
+-----------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)


MariaDB [preksha]> set @num = 0 $$
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [preksha]> call check_no(@num) $$
+--------------+
| Check_Number |
+--------------+
| Number is 0  |
+--------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

****

****

(b) .Second procedure accepts one date argument and check that is < sysdate or not.

****

****

```
MariaDB [preksha]> create procedure check_date18(in user_date date)
    ->      begin
    ->      declare date_diff int;
    -> select current_date();
    ->     set date_diff = DATEDIFF(user_date,current_date);
    ->      if date_diff < 0 then
    ->      select 'user date is less then system date' as 'Check_Date';
    ->      elseif date_diff > 0 then
    ->      select 'user date is greater then sysdate' as 'Check_date';
    ->      else
    ->      select 'Both dates are same' as 'Check_Date';
    ->      end if;
    ->      end $$
Query OK, 0 rows affected (0.12 sec)



MariaDB [preksha]> set @user_date = '1998-10-18' $$
Query OK, 0 rows affected (0.00 sec)



MariaDB [preksha]> call check_date18(@user_date) $$
+----------------+
| current_date() |
+----------------+
| 2020-04-25     |
+----------------+
1 row in set (0.00 sec)

+-----------------------------------+
| Check_Date                        |
+-----------------------------------+
| user date is less then system date |
+-----------------------------------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.02 sec)
```

****

****

(c). Third procedure accepts a name and check whether it is in uppercase or not.

****

****

```
MariaDB [preksha]>  create procedure check_name_case(in name varchar(50))
    ->      begin
```

```
    ->      declare u_c boolean default true;
    ->      declare len int;
    ->      declare i int default 1;
    ->      declare temp varchar(50);
    ->      set len = length(name);
    ->      while i <= len do
    ->      set temp = substr(name,i,1);
    ->      if ASCII(temp) >= 97 AND ASCII(temp) <= 122 then
    ->      set u_c = false;
    ->      end if;
    ->      set i = i + 1;
    ->      end while ;
    ->      if u_c then
    ->      select 'Name is Uppercase' as 'Check_case';
    ->      else
    ->      select 'Name is Lowercase' as 'Chack_case';
    ->      end if;
    ->      end $$
Query OK, 0 rows affected (0.09 sec)


MariaDB [preksha]> set @name = 'PREKSHA';
    -> $$
Query OK, 0 rows affected (0.00 sec)

MariaDB [preksha]> call check_name_case(@name) $$
+-------------------+
| Check_case        |
+-------------------+
| Name is Uppercase |
+-------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)


MariaDB [preksha]> set @name = 'Preksha';
    -> $$
Query OK, 0 rows affected (0.00 sec)

MariaDB [preksha]> call check_name_case(@name) $$
+-------------------+
| Chack_case        |
+-------------------+
| Name is Lowercase |
+-------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

********************************************************************************
********************************************************************************
Final Procedure for Question : 6.
********************************************************************************
********************************************************************************
MariaDB [preksha]> create procedure final_3()
    -> begin
    -> call check_no(@num);
    -> call check_date18(@user_date);
    -> call check_name_case(@name);
    -> end $$
Query OK, 0 rows affected (0.14 sec)
```

```
MariaDB [preksha]> call final_3() $$
+--------------+
| Check_Number |
+--------------+
| Number is 0  |
+--------------+
1 row in set (0.00 sec)



+----------------+
| current_date() |
+----------------+
| 2020-04-25     |
+----------------+
1 row in set (0.01 sec)

+-----------------------------------+
| Check_Date                        |
+-----------------------------------+
| user date is less then system date |
+-----------------------------------+
1 row in set (0.02 sec)



+-------------------+
| Chack_case        |
+-------------------+
| Name is Lowercase |
+-------------------+
1 row in set (0.02 sec)

Query OK, 0 rows affected (0.03 sec)


********************************************************************************
********************************************************************************
Question 7 : Write a procedure to display first n fibonacci numbers.
********************************************************************************
********************************************************************************
MariaDB [preksha]->  create or replace procedure fibonacci(in no int)
       ->   begin
       ->   declare first int;
       ->   declare second int;
       ->   declare third int ;
       ->   declare i int default 1 ;
       ->   declare mystr varchar(100) default '' ;
       ->   set first = 0;
       ->   set second = 1;
       ->   while i <= no do
       ->    set third = first + second ;
       ->    set first = second ;
       ->    set second = third ;
       ->    set mystr = concat(mystr,first,',');
       ->    set i = i + 1 ;
       ->   end while ;
       ->   select mystr as 'FIBONACCI SERIES';
       ->   end $$

Query OK, 0 rows affected (0.15 sec)

MariaDB [preksha]> call fibonacci(10)$$
```

```
+--------------------------+
| FIBONACCI SERIES         |
+--------------------------+
| 1,1,2,3,5,8,13,21,34,55, |
+--------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.02 sec)



********************************************************************************
********************************************************************************
Question 8 : Write a procedure to find simple interest.
********************************************************************************
********************************************************************************
MariaDB [preksha]> create procedure simplei(in p int,in r int,in n int,out i int)
    -> begin
    -> set i = (p * r * n) / 100;
    -> select i;
    -> end $$
Query OK, 0 rows affected (0.13 sec)



MariaDB [preksha]> set @p = 10;
    -> set @r = 10;
    -> set @n = 10$$
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)



MariaDB [preksha]> call simplei(@p,@r,@n,@i)$$
+------+
| i    |
+------+
|   10 |
+------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

********************************************************************************
********************************************************************************
Question 9 : Write a procedure to reverse entered number.
********************************************************************************
********************************************************************************
MariaDB [preksha]> create procedure reverse(in no int)
    -> begin
    -> declare rem,rev int;
    -> set rem = 0;
    -> set rev = 0;
    ->  while no != 0 do
    ->  set rem = no % 10;
    ->  set rev = rev * 10 + rem;
    ->  set no = no / 10;
    -> end while;
    -> select rev as 'Reverse No.';
    -> end $$
Query OK, 0 rows affected (0.07 sec)
```

```
MariaDB [preksha]> set @no = 123 $$
Query OK, 0 rows affected (0.00 sec)


MariaDB [preksha]> call reverse(@no) $$
+-------------+
| Reverse No. |
+-------------+
|         321 |
+-------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

****
****

10. Write a procedure and find its equivalent roman value.

****
****

```
MariaDB [preksha]->  create or replace procedure roman(in num int)
    ->           begin
    ->           declare num2 int ;
    ->           declare string varchar(20) default '' ;
    ->           set num2 = num ;
    ->           while num2 != 0
    ->           do
    ->            case
    ->            when num2 >= 1000
    ->            then
    ->            set string = concat(string, 'M');
    ->            set num2 = num2 - 1000 ;
    ->            when num2 >= 900
    ->            then
    ->            set string = concat(string, 'CM');
    ->            set num2 = num2 - 900 ;
    ->      when num2 >= 500
    ->            then
    ->            set string = concat(string, 'D');
    ->            set num2 = num2 - 500 ;
    ->      when num2 >= 400
    ->            then
    ->            set string = concat(string, 'CD');
    ->            set num2 = num2 - 400 ;
    ->      when num2 >= 100
    ->            then
    ->            set string = concat(string, 'C');
    ->            set num2 = num2 - 100 ;
    ->      when num2 >= 90
    ->            then
    ->            set string = concat(string, 'XC');
    ->            set num2 = num2 - 90 ;
    ->      when num2 >= 50
    ->            then
    ->            set string = concat(string, 'L');
    ->            set num2 = num2 - 50 ;
    ->      when num2 >= 40
    ->            then
    ->            set string = concat(string, 'XL');
    ->            set num2 = num2 - 40 ;
    ->      when num2 >= 10
    ->            then
```

```
    ->            set string = concat(string, 'X');
    ->            set num2 = num2 - 10 ;
    ->      when num2 >= 9
    ->            then
    ->            set string = concat(string, 'IX');
    ->            set num2 = num2 - 9 ;
    ->      when num2 >= 5
    ->            then
    ->            set string = concat(string, 'V');
    ->            set num2 = num2 - 5 ;
    ->      when num2 >= 4
    ->            then
    ->            set string = concat(string, 'IV');
    ->            set num2 = num2 - 4 ;
    ->      when num2 >= 1
    ->            then
    ->            set string = concat(string, 'I');
    ->            set num2 = num2 - 1 ;
    ->      end case ;
    ->      end while ;
    ->      select num as 'Number',string as 'ROMAN';
    ->      end $$
Query OK, 0 rows affected (0.15 sec)

MariaDB [preksha]> call roman(1904)$$
+--------+-------+
| Number | ROMAN |
+--------+-------+
|   1904 | MCMIV |
+--------+-------+
1 row in set (0.00 sec)
```

****

****

11. Write a program to enter a number and find addition of each digit of that number using function.

****

****

CREATE FUNCTION ..

```
 MariaDB [preksha]  CREATE OR REPLACE FUNCTION adddigit (no int)
    -> RETURNS int
    -> begin
    -> declare rem INT;
    -> declare res INT;
    -> set res = 0;
    -> set rem = 0;
    ->   while no != 0 do
    ->       set rem = MOD(no,10);
    ->       set res =res +rem;
    ->       set no = no DIV 10;
    ->   end while;
    -> RETURN res;
    -> end $$
Query OK, 0 rows affected (0.06 sec)
```

CREATE PROCEDURE...

```
 MariaDB [preksha]->  create procedure sum (in no int )
    -> begin
    -> select adddigit (no);
    -> end $$
Query OK, 0 rows affected (0.14 sec)
```

```
MariaDB [preksha]->  call sum (12345) $$
+---------------+
| adddigit (no) |
+---------------+
|            15 |
+---------------+
1 row in set (0.00 sec)
```

****
****

Question 13 : Write a program to reverse the string.

****
****

```
MariaDB [preksha]> create procedure revstr(in str varchar(50))
    -> begin
    -> declare str2 varchar(50) default '';
    -> declare len int;
    -> set len = length(str);
    -> while len != 0 do
    -> set str2 = concat(str2,substr(str,len,1));
    -> set len = len - 1;
    -> end while;
    -> select str2 as 'Reverse String';
    -> end $$
Query OK, 0 rows affected (0.13 sec)


MariaDB [preksha]> set @str = 'Preksha' $$
Query OK, 0 rows affected (0.00 sec)


MariaDB [preksha]> call revstr(@str) $$
+----------------+
| Reverse String |
+----------------+
| ahskerP        |
+----------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

****
****

14. Write a program to convert the string into toggle case.

****
****

```
 MariaDB [preksha]->  create or replace procedure togglestr(in str varchar(50))
    ->      begin
    ->      declare start,stop int ;
    ->      declare temp int;
    ->      set start = 0 ;
    ->      set stop = 0;
    ->      set stop = length(str);
    ->      while start <= stop do
    ->      set temp = ASCII(SUBSTR(str,start,1));
    ->       if temp >=65 AND temp <= 90 then
    ->               set temp =temp +32 ;
    ->               set str = INSERT(str,start,1,CHAR(temp));
    ->      elseif temp >=97 AND temp <= 122 THEN
    ->               set temp =temp - 32;
    ->             set str = INSERT(str,start,1,CHAR(temp));
```

```
    ->      end if ;
    ->       set start = start + 1;
    ->      end while ;
    ->      select str as 'Toggled case ';
    ->      end $$
Query OK, 0 rows affected (0.15 sec)

MariaDB [preksha]-> call togglestr('Prekshu')$$
+------------------+
| Toggled case     |
+------------------+
| pREKSHU          |
+------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

****

****

15. Write a PL/SQL block to convert given numbers into text words

****

****

```
MariaDB [preksha]->  create or replace procedure no_conv(in num int)
    ->      begin
    ->      declare str varchar(100) default ' ';
    ->     declare i int default 1;
    ->      declare cnt int;
    ->      declare new_num varchar(1);
    ->      set new_num = '';
    ->      set cnt = length(num);
    ->      while(cnt > 0 ) DO
    ->        set new_num = substr(num,i,1);
    ->              if new_num = 1 then
    ->                      set str = concat(str , ' '  , 'one');
    ->              end if ;
    ->              if new_num = 2 then
    ->                      set str = concat(str ,  ' ' , 'two');
    ->              end if ;
    ->               if new_num = 3 then
    ->                      set str = concat(str ,  ' ' , 'three');
    ->              end if ;
    ->               if new_num = 4 then
    ->                      set str= concat(str,  ' ' , 'four');
    ->              end if ;
    ->               if new_num = 5 then
    ->                      set str= concat(str, ' ', 'five');
    ->              end if ;
    ->               if new_num = 6 then
    ->                      set str = concat(str, ' ', 'six');
    ->              end if ;
    ->               if new_num = 7 then
    ->                      set str = concat(str , ' ' , 'seven');
    ->              end if ;
    ->               if new_num = 8 then
    ->                      set str = concat(str , ' ', 'eight');
    ->              end if ;
    ->               if new_num = 9 then
    ->                      set str = concat(str ,  ' ', 'nine');
    ->              end if ;
    ->               if new_num = 0 then
```

```
    ->                         set str = concat(str ,  ' ' , 'zero');
    ->               end if ;
    ->             set cnt = cnt  - 1 ;
    ->             set i = i + 1 ;
    ->          end while;
    -> select str as 'words', num as 'Number';
    -> end $$
Query OK, 0 rows affected (0.21 sec)
```

****
****

Output:

****
****

```
MariaDB [preksha]-> call no_conv(132)$$
+-----------------+--------+
| words           | Number |
+-----------------+--------+
|   one three two |    132 |
+-----------------+--------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.03 sec)
```

****
****

****
****
Name   : Preksha Sheth
Roll No. : 36
Class   : MCA-II
Subject : RDBMS-II(PL/SQL)

****
****
ASSIGNMENT - 2

****
****
General Procedures

****
****
Q. 1 : WAP to input two numbers and find out what is the output of all arithmetic operations.
(Addition, Subtraction, Multiplication, Division etc.)

****
****
```
MariaDB [preksha]> create procedure calculate(in x int,in y int)
    ->      begin
    ->      declare a int;
    ->      declare s int;
    ->      declare m int;
    ->      declare d int;
    ->      set a = x + y;
    ->      set s = x - y;
    ->      set m = x * y;
    ->      set d = x / y;
    ->      select a as addition, s as subtraction, m as multiplication,d
as division;
    ->      end $$
Query OK, 0 rows affected (0.14 sec)


MariaDB [preksha]> set @x = 100;
    -> set @y = 20 $$
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)


MariaDB [preksha]> call calculate(@x,@y) $$
+----------+-------------+----------------+----------+
| addition | subtraction | multiplication | division |
+----------+-------------+----------------+----------+
|      120 |          80 |           2000 |        5 |
+----------+-------------+----------------+----------+
1 row in set (0.00 sec)
```

```
Query OK, 0 rows affected (0.01 sec)

**************************************************************************
**************************************************************************
Q. 2 : WAP to input rollno and three subject marks. Find out total,
percentage, result and
       grade for the student from the entered data.

**************************************************************************
**************************************************************************
MariaDB [preksha]> create procedure student_data(in rno int,in m1 int,in
m2 int,in m3 int)
    -> begin
    -> declare total int;
    -> declare per decimal(5,2);
    -> declare grade varchar(5);
    -> declare str varchar(500) default ' ';
    -> declare r_no int;
    -> set r_no = rno;
    -> set total = m1 + m2 + m3;
    -> set per = (total * 2) / 3;
    -> if per >= 70 then
    -> set grade = 'A';
    -> elseif per >= 60 && per < 69 then
    -> set grade = 'B';
    -> elseif per >= 50 && per < 59 then
    -> set grade = 'C';
    -> elseif per < 50 then
    -> set grade = 'FAIL';
    -> end if;
    -> set str = concat(str,' Rollno = ',r_no,'  Total = ',total,'
Percentage = ',per,'  Grade = ',grade);
    -> select str;
    -> end $$
Query OK, 0 rows affected (0.21 sec)



MariaDB [preksha]> set @rno = 36;
    ->     set @m1 = 40;
    ->      set @m2 = 46;
    ->     set @m3 = 37 $$
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.02 sec)
```

```
MariaDB [preksha]>  call student_data(@rno,@m1,@m2,@m3) $$
+-------------------------------------------------------------+
| str                                                         |
+-------------------------------------------------------------+
|   Rollno = 36  Total = 123  Percentage = 82.00  Grade = A |
+-------------------------------------------------------------+
1 row in set (0.18 sec)
```

****
****

Q. 3 : WAP to print first 10 odd numbers using for loop.

****
****

```
MariaDB [preksha]> create procedure odd_find()
    -> begin
    -> declare i int default 0;
    -> declare j int default 0;

    -> declare str varchar(500) default '';
    -> odd : LOOP
    -> if i = 10 then
    -> leave odd;
    -> end if;
    -> if mod(j,2) != 0 then
    -> set str = concat(str,' ',j);
    -> set i = i + 1;
    -> end if;
    -> set j = j + 1;
    -> end LOOP odd;
    -> select str;
    -> end $$
Query OK, 0 rows affected (0.14 sec)
```

```
MariaDB [preksha]> call odd_find() $$
+---------------------------+
| str                       |
+---------------------------+
|  1 3 5 7 9 11 13 15 17 19 |
+---------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

****
****

Q. 4 : WAP to print prime numbers upto 10 using while loop.

****
****

```
MariaDB [preksha]> create  procedure prime_no()
    ->      begin
    ->      declare i int default 2;
    ->      declare j int default 2;
    ->      declare cnt int default 0;
    ->      declare str varchar(500) default '';
    ->      while cnt<=10 do
    ->      set i = 2;
    ->      lp1:while i<j do
    ->      if j % i=0 then
    ->      leave lp1;
    ->      end if;
    ->      set i = i + 1;
    ->      end while lp1;
    ->      if i = j then
    ->      set str = concat(str,' ',j);
    ->      set cnt = cnt + 1;
    ->      end if;
    ->      set j = j + 1;
    ->      end while;
    ->      select str;
    ->      end $$
Query OK, 0 rows affected (0.06 sec)



MariaDB [preksha]> call prime_no() $$
+-------------------------------+
| str                           |
+-------------------------------+
|  2 3 5 7 11 13 17 19 23 29 31 |
+-------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
************************************************************************
************************************************************************
Q. 5 : WAP to input three nos and find out maximum and minimum from it.

************************************************************************
************************************************************************
MariaDB [preksha]> create procedure find_min_max(in no1 int,in no2 int,in
no3 int)
    -> begin
    -> declare min int;
    -> declare max int;
    -> if no1 > no2 && no1 > no3 then
    -> set max = no1;
    -> elseif no2 > no1 && no2 > no3 then
    -> set max = no2;
    -> elseif no3 > no1 && no3 > no2 then
    -> set max = no3;
    -> end if;
    -> if no1 < no2 && no1 < no3 then
```

```
    -> set min = no1;
    -> elseif no2 < no1 && no2 < no3 then
    -> set min = no2;
    -> elseif no3 < no1 && no3 < no2 then
    -> set min = no3;
    -> end if;
    -> select max as 'Maximum',min as 'Minimum';
    -> end $$
Query OK, 0 rows affected (0.13 sec)



MariaDB [preksha]> set @no1 = 10;
    -> set @no2 = 45;
    -> set @no3 = 20 $$
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.01 sec)



MariaDB [preksha]> call find_min_max(@no1,@no2,@no3) $$
+---------+---------+
| Maximum | Minimum |
+---------+---------+
|      45 |      10 |
+---------+---------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
**************************************************************************
**************************************************************************
Q. 6 : WAP to input empno from keybord. Check whether inputed empno exist
in emp table
       or not. If not give error message otherwise display name and
salary of that employee.

**************************************************************************
**************************************************************************
MariaDB [preksha]> select * from emp $$
+--------+----------+-----------+------------+------------+
| emp_id | emp_name | emp_job   | emp_salary | DOJ        |
+--------+----------+-----------+------------+------------+
|      1 | Preksha  | Developer |      20000 | 2005-10-25 |
|      2 | Prerak   | DBA       |      18000 | 1998-12-18 |
|      3 | Dhruvin  | DBA       |      15000 | 2010-05-15 |
|      4 | Aman     | Developer |      25000 | 2002-09-10 |
|      5 | Mahi     | Designer  |      15000 | 2015-10-20 |
|      7 | Abhi     | Designer  |      23000 | 2010-12-12 |
+--------+----------+-----------+------------+------------+
6 rows in set (0.00 sec)
```

```
MariaDB [preksha]> create procedure emp_exists(in id int)
    -> begin
    -> declare cnt int default 0;
    -> select count(*) into cnt from emp where emp_id = id;
    -> if cnt >= 1 then
    ->   select concat(cnt,' Employee exists..') as 'count Message';
    ->   select emp_name,emp_salary from emp where emp_id = id;
    -> else
    ->   select 'Employee does not Exists..' as 'Message';
    -> end if;
    -> end $$
Query OK, 0 rows affected (0.06 sec)



MariaDB [preksha]> set @id = 2 $$
Query OK, 0 rows affected (0.00 sec)



MariaDB [preksha]> call emp_exists(@id) $$
+---------------------+
| count Message       |
+---------------------+
| 1 Employee exists.. |
+---------------------+
1 row in set (0.03 sec)

+----------+------------+
| emp_name | emp_salary |
+----------+------------+
| Prerak   |      18000 |
+----------+------------+
1 row in set (0.04 sec)

Query OK, 0 rows affected (0.06 sec)



MariaDB [preksha]> call emp_exists(10) $$
+---------------------------+
| Message                   |
+---------------------------+
| Employee does not Exists.. |
+---------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

********************************************************************************
********************************************************************************
```

Q. 7 : WAP to insert record in Customer table.
Customer(cust_id,cust_name,address,city);

****
****

```
MariaDB [preksha]> create procedure ins_data_cust(in id int,in nm
varchar(50),in addr varchar(100),in cty varchar(50))
    -> begin
    -> insert into customer_master(cust_id,cust_name,address,city)
values(id,nm,addr,cty);
    -> select 'Record Inserted..' as 'Message';
    -> end $$
Query OK, 0 rows affected (0.08 sec)



MariaDB [preksha]> call ins_data_cust(1,'preksha','motera','ahmedabad')
$$
+-------------------+
| Message           |
+-------------------+
| Record Inserted.. |
+-------------------+
1 row in set (0.16 sec)

Query OK, 0 rows affected (0.18 sec)


MariaDB [preksha]> call ins_data_cust(2,'prerak','motera','ahmedabad') $$
+-------------------+
| Message           |
+-------------------+
| Record Inserted.. |
+-------------------+
1 row in set (0.13 sec)

Query OK, 0 rows affected (0.15 sec)


MariaDB [preksha]> call ins_data_cust(3,'virati','odhav','ahmedabad') $$
+-------------------+
| Message           |
+-------------------+
| Record Inserted.. |
+-------------------+
1 row in set (0.40 sec)

Query OK, 0 rows affected (0.42 sec)
```

```
MariaDB [preksha]> call ins_data_cust(4,'abhi','mahavirnagar','surat') $$
+-------------------+
| Message           |
+-------------------+
| Record Inserted.. |
+-------------------+
1 row in set (0.13 sec)

Query OK, 0 rows affected (0.14 sec)


MariaDB [preksha]> call ins_data_cust(5,'dhruvin','gorwa','vadodara') $$
+-------------------+
| Message           |
+-------------------+
| Record Inserted.. |
+-------------------+
1 row in set (0.03 sec)

Query OK, 0 rows affected (0.05 sec)



MariaDB [preksha]> select * from customer_master $$
+---------+-----------+--------------+-----------+
| cust_id | cust_name | address      | city      |
+---------+-----------+--------------+-----------+
|       1 | preksha   | motera       | ahmedabad |
|       2 | prerak    | motera       | ahmedabad |
|       3 | virati    | odhav        | ahmedabad |
|       4 | abhi      | mahavirnagar | surat     |
|       5 | dhruvin   | gorwa        | vadodara  |
+---------+-----------+--------------+-----------+
5 rows in set (0.00 sec)
**************************************************************************
**************************************************************************
```

Function

```
**************************************************************************
**************************************************************************
Q. 1 : WAF which accepts the name from user and returns the length of
that name.
**************************************************************************
**************************************************************************
MariaDB [preksha]> create function name_len(name varchar(50))
    -> returns int
    -> begin
    -> declare len int;
    -> set len = length(name);
    -> return (len);
    -> end $$
Query OK, 0 rows affected (0.16 sec)
```

```
MariaDB [preksha]> set @name = 'preksha' $$
Query OK, 0 rows affected (0.00 sec)



MariaDB [preksha]> select name_len(@name) $$
+-----------------+
| name_len(@name) |
+-----------------+
|               7 |
+-----------------+
1 row in set (0.00 sec)

*************************************************************************
*************************************************************************
Q.2 : WAF which accepts one number and return TRUE if no is prime and
return FALSE if No. is not prime.
*************************************************************************
*************************************************************************
MariaDB [preksha]> create function prime_check(num int(10))
    -> returns varchar(20)
    -> begin
    -> declare val,lim int;
    -> declare flag int default 0;
    -> set val=2;
    -> set lim=num-1;
    -> myloop: LOOP
    -> if val=lim then
    -> leave myloop;
    -> else
    -> if mod(num,val)=0 then
    -> return "Number is not Prime.";
    -> set flag=1;
    -> leave myloop;
    -> else
    -> set val=val+1;
    -> end if;
    -> end if;
    -> end loop;
    -> if flag=0 then
    ->  return "Number is Prime";
    -> end if;
    -> end $$
Query OK, 0 rows affected (0.12 sec)
```

```
MariaDB [preksha]> select prime_check(10) $$
+----------------------+
| prime_check(10)      |
+----------------------+
| Number is not Prime. |
+----------------------+
1 row in set (0.00 sec)



MariaDB [preksha]> select prime_check(5) $$
+-----------------+
| prime_check(5)  |
+-----------------+
| Number is Prime |
+-----------------+
1 row in set (0.00 sec)
```

****
****

Q.3 : Write a function which accepts the department no and returns maximum salary of that Department.

Handle the error if deptno does not exist or select statement return more than one row.

****
****

```
MariaDB [preksha]> select * from employee $$
+------+---------+-----------+---------+
| e_id | e_name  | e_salary  | dept_id |
+------+---------+-----------+---------+
|    1 | Preksha | 303000.00 |       3 |
|    2 | Prerak  | 352000.00 |       2 |
|    3 | Aman    | 254000.00 |       4 |
|    4 | Abhi    | 204000.00 |       4 |
|    5 | Pushti  | 281000.00 |       1 |
|    6 | Mahi    | 321000.00 |       1 |
|    7 | Dhruvin | 275000.00 |       5 |
+------+---------+-----------+---------+
7 rows in set (0.00 sec)



MariaDB [preksha]> create function max_Salary(d_id int)
    -> returns varchar(500)
    -> begin
    -> declare str varchar(500) default '';
    -> declare sal decimal(8,2);
    -> select max(e_salary) into sal from employee where dept_id = d_id;
    -> if sal > 0 then
    ->  set str = concat(str,' ',sal);
    -> else
    ->  set str = concat(str,' ','Department doesnot exists');
    -> end if;
```

```
    -> return (str);
    -> end $$
Query OK, 0 rows affected (0.15 sec)



MariaDB [preksha]> select max_salary(8) $$
+----------------------------+
| max_salary(8)              |
+----------------------------+
|  Department doesnot exists |
+----------------------------+
1 row in set (0.00 sec)



MariaDB [preksha]> select max_salary(1) $$
+---------------+
| max_salary(1) |
+---------------+
|  321000.00    |
+---------------+
1 row in set (0.00 sec)
```

****
****

Q.4 : Write a function to display whether the entered (User Input) employee no exists or not.

****
****

```
MariaDB [preksha]> create function chk_emp_exist(id int)
    -> returns varchar(500)
    -> begin
    -> declare str varchar(500) default '';
    -> declare cnt int default 0;
    -> select count(*) into cnt from employee where e_id = id;
    -> if cnt > 0 then
    ->  set str = concat(str,' ','Employee Exists..');
    -> else
    ->  set str = concat(str,' ','Employee does not Exists..');
    -> end if;
    -> return(str);
    -> end $$
Query OK, 0 rows affected (0.04 sec)


MariaDB [preksha]> select chk_emp_exist(1) $$
+--------------------+
| chk_emp_exist(1)   |
+--------------------+
|  Employee Exists.. |
+--------------------+
1 row in set (0.00 sec)
```

```
MariaDB [preksha]> select chk_emp_exist(8) $$
+----------------------------+
| chk_emp_exist(8)           |
+----------------------------+
|  Employee does not Exists.. |
+----------------------------+
1 row in set (0.00 sec)
```

****
****

Q.5 : WAF which accepts one no and returns that no+100. Use INOUT mode.

****
****

```
MariaDB [preksha]> create function add_number(no int)
    -> returns int
    -> begin
    -> set no = no + 100;
    -> return(no);
    -> end $$
Query OK, 0 rows affected (0.15 sec)


MariaDB [preksha]> select add_number(50) $$
+----------------+
| add_number(50) |
+----------------+
|            150 |
+----------------+
1 row in set (0.00 sec)
```

****
****

Q.6 : WAF which accepts the empno.
If salary<10000 than give raise by 30%.
If salary<20000 and salary>=10000 than give raise by 20%.
If salary>20000 than give raise by 10%. Handle the error if any.

****
****

```
MariaDB [preksha]> select * from emp $$
+--------+----------+-----------+------------+
| emp_id | emp_name | emp_job   | emp_salary |
+--------+----------+-----------+------------+
|      1 | Preksha  | Developer |      20000 |
|      2 | Prerak   | DBA       |      18000 |
|      3 | Dhruvin  | DBA       |      15000 |
|      4 | Aman     | Developer |      25000 |
|      5 | Mahi     | Designer  |      15000 |
|      7 | Abhi     | Designer  |      23000 |
+--------+----------+-----------+------------+
6 rows in set (0.00 sec)
```

```
MariaDB [preksha]> create function inc_sal(e_id int)
    ->      returns varchar(500)
    ->      begin
    ->      declare str varchar(500) default '';
    ->      declare sal int;
    ->      declare cnt int;
    ->      select count(*) into cnt from emp where emp_id = e_id;
    ->      if cnt != 0 then
    ->      select emp_salary into sal from emp where emp_id = e_id;
    ->      set str = concat(str,' ',sal);
    ->      if sal < 10000 then
    ->      set sal = sal + ((sal * 30) / 100);
    ->      elseif sal > 10000 && sal < 20000 then
    ->      set sal = sal + ((sal * 20) / 100);
    ->      elseif sal > 20000 then
    ->      set sal = sal + ((sal * 10) / 100);
    ->      end if;
    ->      set str = concat('salary = ',str,'  After Increase Salary =
',sal);
    ->      else
    ->      set str = concat(str,' ','Employee Does not Exists..');
    ->      end if;
    ->      return (str);
    ->      end $$
Query OK, 0 rows affected (0.06 sec)



MariaDB [preksha]> select inc_sal(2) $$
+-----------------------------------------------+
| inc_sal(2)                                    |
+-----------------------------------------------+
| salary =  18000  After Increase Salary = 21600 |
+-----------------------------------------------+
1 row in set (0.00 sec)



MariaDB [preksha]> select inc_sal(10) $$
+-----------------------------+
| inc_sal(10)                 |
+-----------------------------+
|  Employee Does not Exists.. |
+-----------------------------+
1 row in set (0.00 sec)

**************************************************************************
**************************************************************************
```

Q.7 : WAF which accepts the empno and returns the experience in years.
Handle the error if
empno does not exist.
EMP(Empno, Empname, DOJ);

****

****

```
MariaDB [preksha]> select * from emp $$
+--------+----------+-----------+------------+------------+
| emp_id | emp_name | emp_job   | emp_salary | DOJ        |
+--------+----------+-----------+------------+------------+
|      1 | Preksha  | Developer |      20000 | 2005-10-25 |
|      2 | Prerak   | DBA       |      18000 | 1998-12-18 |
|      3 | Dhruvin  | DBA       |      15000 | 2010-05-15 |
|      4 | Aman     | Developer |      25000 | 2002-09-10 |
|      5 | Mahi     | Designer  |      15000 | 2015-10-20 |
|      7 | Abhi     | Designer  |      23000 | 2010-12-12 |
+--------+----------+-----------+------------+------------+
6 rows in set (0.00 sec)



MariaDB [preksha]> create function experience_emp(e_id int)
    -> returns varchar(500)
    -> begin
    -> declare c_y int;
    -> declare j_y int;
    -> declare exp int;
    -> declare j_dt date;
    -> declare cnt int default 0;
    -> declare str varchar(500) default '';
    -> select count(*) into cnt from emp where emp_id = e_id;
    -> if cnt != 0 then
    -> set str = concat(str,' ','Experience is ');
    -> select DOJ into j_dt from emp where emp_id = e_id;
    -> set c_y = YEAR(current_date());
    -> set j_y = YEAR(j_dt);
    -> set exp = c_y - j_y;
    -> set str = concat(str,' ',exp);
    -> set str = concat(str,' ','Years..');
    -> else
    -> set str = concat(str,' ','Employee Does not Exists..');
    -> end if;
    -> return (str);
    -> end $$
Query OK, 0 rows affected (0.07 sec)


MariaDB [preksha]> select experience_emp(1) $$
+----------------------------+
| experience_emp(1)          |
+----------------------------+
|  Experience is  15 Years.. |
+----------------------------+
1 row in set (0.00 sec)
```

```
MariaDB [preksha]> select experience_emp(10) $$
+-----------------------------+
| experience_emp(10)          |
+-----------------------------+
|  Employee Does not Exists.. |
+-----------------------------+
1 row in set (0.00 sec)
```

****
****

Procedures

****
****

Q. 1 : Write a procedure which accepts the empno and returns the associated empname. If
empno does not exist than give proper error message.
EMP(Empno, Empname).

****
****

```
MariaDB [preksha]> select * from emp $$
+--------+----------+-----------+------------+------------+
| emp_id | emp_name | emp_job   | emp_salary | DOJ        |
+--------+----------+-----------+------------+------------+
|      1 | Preksha  | Developer |      20000 | 2005-10-25 |
|      2 | Prerak   | DBA       |      18000 | 1998-12-18 |
|      3 | Dhruvin  | DBA       |      15000 | 2010-05-15 |
|      4 | Aman     | Developer |      25000 | 2002-09-10 |
|      5 | Mahi     | Designer  |      15000 | 2015-10-20 |
|      7 | Abhi     | Designer  |      23000 | 2010-12-12 |
+--------+----------+-----------+------------+------------+
6 rows in set (0.00 sec)
```

```
MariaDB [preksha]> create procedure emp_data(in id int)
    -> begin
    -> declare cnt int default 0;
    -> declare nm varchar(50);
    ->
    -> select count(*) into cnt from emp where emp_id = id;
    -> if cnt > 0 then
    -> select emp_name into nm from emp where emp_id = id;
    -> select concat(nm) as 'Employee Name';
    -> else
    -> select concat('Employee Does not Exist..') as 'Message';
    -> end if;
    -> end $$
Query OK, 0 rows affected (0.13 sec)
```

```
MariaDB [preksha]> call emp_data(1) $$
+---------------+
| Employee Name |
+---------------+
| Preksha       |
+---------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)



MariaDB [preksha]> call emp_data(8) $$
+--------------------------+
| Message                  |
+--------------------------+
| Employee Does not Exist.. |
+--------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

**************************************************************************
**************************************************************************
Q. 2 : WAP which accepts the student rollno and returns the name,city and
marks of all the
       subjects of that student.
       STUDENT (Stud_ID, Stud_name, m1, m2, m3).
**************************************************************************
**************************************************************************
MariaDB [preksha]> select * from student $$
+---------+-----------+------+------+------+-------------+
| stud_id | stud_name | m1   | m2   | m3   | city        |
+---------+-----------+------+------+------+-------------+
|       1 | preksha   |   46 |   42 |   38 | ahmedabad   |
|       2 | prerak    |   39 |   32 |   45 | ahmedabad   |
|       3 | abhi      |   35 |   43 |   40 | himmatnagar |
|       4 | het       |   45 |   33 |   46 | vadodara    |
|       5 | kriya     |   46 |   43 |   34 | surat       |
+---------+-----------+------+------+------+-------------+
5 rows in set (0.00 sec)



MariaDB [preksha]> create procedure stud_data(in rno int)
    -> begin
    -> declare cnt int default 0;
    -> select count(*) into cnt from student where stud_id = rno;
    -> if cnt > 0 then
    -> select * from student where stud_id = rno;
    -> else
```

```
    -> select concat('Student does not exist..') as 'message';
    -> end if;
    -> end $$
Query OK, 0 rows affected (0.05 sec)



MariaDB [preksha]> set @rno = 2 $$
Query OK, 0 rows affected (0.00 sec)



MariaDB [preksha]> call stud_data(@rno) $$
+---------+-----------+------+------+------+-----------+
| stud_id | stud_name | m1   | m2   | m3   | city      |
+---------+-----------+------+------+------+-----------+
|       2 | prerak    |   39 |   32 |   45 | ahmedabad |
+---------+-----------+------+------+------+-----------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)



MariaDB [preksha]> call stud_data(8) $$
+--------------------------+
| message                  |
+--------------------------+
| Student does not exist.. |
+--------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

**************************************************************************
**************************************************************************
Q. 3 : WAP which accepts the name from the user. Return UPPER if name is
in uppercase,
       LOWER if name is in lowercase, MIXCASE if name is entered using
both the case.

**************************************************************************
**************************************************************************
MariaDB [preksha]> create procedure chk_case(in name varchar(20))
    -> begin
    -> declare val int default 1;
    -> declare temp varchar(5);
    -> declare len int;
    -> declare upper boolean default FALSE;
    -> declare lower boolean default FALSE;
    ->
    -> set len = length(name);
    -> while val < len do
```

```
    -> set temp = substr(name,val,1);
    -> if ascii(temp) >= 97 AND ascii(temp) <= 122 then
    -> set lower = TRUE;
    -> else
    -> set upper = TRUE;
    -> end if;
    -> set val = val + 1;
    -> end while;
    -> if upper = TRUE AND lower = TRUE then
    -> select 'NAME IS IN Mixcase..';
    -> elseif lower = true then
    -> select 'name is in lowercase..';
    -> elseif upper = TRUE then
    -> select 'Name is in UPPERCASE..';
    -> end if;
    -> end $$
Query OK, 0 rows affected (0.03 sec)



MariaDB [preksha]> call chk_case('Preksha') $$
+----------------------+
| NAME IS IN Mixcase.. |
+----------------------+
| NAME IS IN Mixcase.. |
+----------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)



MariaDB [preksha]> call chk_case('PREKSHA') $$
+------------------------+
| Name is in UPPERCASE.. |
+------------------------+
| Name is in UPPERCASE.. |
+------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)



MariaDB [preksha]> call chk_case('preksha') $$
+------------------------+
| name is in lowercase.. |
+------------------------+
| name is in lowercase.. |
+------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

```
**************************************************************************
**************************************************************************
Q. 4 : WAP which accepts the student rollno and returns the highest
percent and name of that
       student to the calling block.
       STUDENT(Stud_ID,Stud_name,percent);

**************************************************************************
**************************************************************************
MariaDB [preksha]> select * from stud1$$
+------+---------+---------+
| s_id | s_name  | percent |
+------+---------+---------+
|    1 | preksha |   85.40 |
|    2 | prerak  |   86.77 |
|    3 | het     |   76.67 |
|    4 | mahi    |   79.60 |
|    5 | dhruvin |   77.60 |
+------+---------+---------+
5 rows in set (0.00 sec)



MariaDB [preksha]> create procedure h_per(in rn int)
    -> begin
    -> select s_name,percent from stud1 where s_id = rn;
    -> end $$
Query OK, 0 rows affected (0.11 sec)



MariaDB [preksha]> call h_per(2)$$
+--------+---------+
| s_name | percent |
+--------+---------+
| prerak |   86.77 |
+--------+---------+
1 row in set (0.00 sec)

**************************************************************************
**************************************************************************
Q. 5 : WAP which accepts the date of joining for specific employee and
returns the years of
       experience along with its name. Accept the Employee no from user.
       EMP (empno, empname, DOJ);

**************************************************************************
**************************************************************************
```

```
MariaDB [preksha]> select * from emp$$
+--------+----------+-----------+------------+------------+
| emp_id | emp_name | emp_job   | emp_salary | DOJ        |
+--------+----------+-----------+------------+------------+
|      1 | Preksha  | Developer |      20000 | 2005-10-25 |
|      2 | Prerak   | DBA       |      18000 | 1998-12-18 |
|      3 | Dhruvin  | DBA       |      15000 | 2010-05-15 |
|      4 | Aman     | Developer |      25000 | 2002-09-10 |
|      5 | Mahi     | Designer  |      15000 | 2015-10-20 |
|      7 | Abhi     | Designer  |      23000 | 2010-12-12 |
|      8 | rishi    | tester    |       1000 | 2000-10-11 |
+--------+----------+-----------+------------+------------+
7 rows in set (0.04 sec)

============================
Function :
============================

MariaDB [preksha]> create function experience_emp(e_id int)
    -> returns varchar(500)
    -> begin
    -> declare c_y int;
    -> declare j_y int;
    -> declare exp int;
    -> declare j_dt date;
    -> declare cnt int default 0;
    -> declare str varchar(500) default '';
    -> select count(*) into cnt from emp where emp_id = e_id;
    -> if cnt != 0 then
    -> set str = concat(str,' ','Experience is ');
    -> select DOJ into j_dt from emp where emp_id = e_id;
    -> set c_y = YEAR(current_date());
    -> set j_y = YEAR(j_dt);
    -> set exp = c_y - j_y;
    -> set str = concat(str,' ',exp);
    -> set str = concat(str,' ','Years..');
    -> else
    -> set str = concat(str,' ','Employee Does not Exists..');
    -> end if;
    -> return (str);
    -> end $$
Query OK, 0 rows affected (0.07 sec)


============================
Trigger :
============================

MariaDB [preksha]> create procedure exper_emp(in en int)
    -> begin
    -> declare msg varchar(500);
    -> set msg = experience_emp(en);
    -> select msg as 'MESSAGE';
    -> end $$
```

```
Query OK, 0 rows affected (0.05 sec)



MariaDB [preksha]> call exper_emp(1)$$
+----------------------------+
| MESSAGE                    |
+----------------------------+
|  Experience is  15 Years.. |
+----------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)



MariaDB [preksha]> call exper_emp(10)$$
+-----------------------------+
| MESSAGE                     |
+-----------------------------+
|  Employee Does not Exists.. |
+-----------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

*****************************************************************************
*****************************************************************************
Q. 6 : WAP which accepts the student roll no and returns the result (in
the form of class: first
       class, second class, third class or fail).
       STUDENT (Stud_ID, Stud_name,m1, m2, m3).

*****************************************************************************
*****************************************************************************
MariaDB [preksha]> select * from student1$$
+------+---------+------+------+------+
| s_id | s_name  | m1   | m2   | m3   |
+------+---------+------+------+------+
|    1 | preksha |   87 |   90 |   88 |
|    2 | prerak  |   77 |   80 |   90 |
|    3 | abhi    |   73 |   86 |   80 |
|    4 | jeet    |   83 |   79 |   90 |
|    5 | rahi    |   58 |   65 |   55 |
|    6 | krishna |   45 |   22 |   33 |
+------+---------+------+------+------+
6 rows in set (0.00 sec)



MariaDB [preksha]> create procedure grade_stu(in rn int)
    -> begin
    -> declare rno int;
    -> declare total int;
```

```
    -> declare ma1 int;
    -> declare ma2 int;
    -> declare ma3 int;
    -> declare per decimal(5,2);
    -> select m1 into ma1 from student1 where s_id = rn;
    -> select m2 into ma2 from student1 where s_id = rn;
    -> select m3 into ma3 from student1 where s_id = rn;
    -> set total = ma1 + ma2 + ma3;
    -> select total;
    -> if total > 99 then
    -> select 'PASS' as 'RESULT';
    -> else
    -> select 'FAIL' as 'RESULT';
    -> end if;
    -> set per = (total / 300) * 100;
    -> select per as 'Percentage';
    -> if per <= 50 then
    -> select 'FAIL' as 'Grade';
    -> elseif per > 50 AND per < 70 then
    -> select 'SECOND CLASS' as 'Grade';
    -> elseif per > 70 AND per < 80 then
    -> select 'FIRST CLASS' as 'Grade';
    -> else
    -> select 'DISTINCTION' as 'Grade';
    -> end if;
    -> end $$
Query OK, 0 rows affected (0.06 sec)



MariaDB [preksha]> call grade_stu(1)$$
+-------+
| total |
+-------+
|   265 |
+-------+
1 row in set (0.00 sec)

+--------+
| RESULT |
+--------+
| PASS   |
+--------+
1 row in set (0.01 sec)

+------------+
| Percentage |
+------------+
|      88.33 |
+------------+
1 row in set (0.02 sec)
```

```
+-------------+
| Grade       |
+-------------+
| DISTINCTION |
+-------------+
1 row in set (0.03 sec)

Query OK, 0 rows affected, 1 warning (0.04 sec)
```

****
****

Cursor

****
****

Q. 1 : Create a cursor for the emp table. Produce the output in following format:

{empname} employee working in department {deptno} earns Rs. {salary}.

EMP(empno, empname, salary, deptno);

****
****

```
MariaDB [preksha]> select * from emp$$
+-------+---------+----------+--------+
| empno | empname | salary   | deptno |
+-------+---------+----------+--------+
|     1 | preksha | 23000.00 | D01    |
|     2 | prerak  | 25000.00 | D01    |
|     3 | dhruvin |  7000.00 | D02    |
+-------+---------+----------+--------+
3 rows in set (0.00 sec)
```

```
MariaDB [preksha]> create procedure emp_cur()
    -> begin
    -> declare name varchar(30);
    -> declare dno varchar(10);
    -> declare sal decimal(8,2);
    -> declare flag int default 0;
    -> declare result varchar(100);
    ->
    -> declare c cursor for select empname,salary,deptno from emp;
    -> declare continue handler for not found set flag=1;
    ->
    -> open c;
    -> get_emp:loop
    -> fetch c into name,sal,dno;
    -> if flag=1 then
    -> leave  get_emp;
    -> end if;
```

```
    -> set result=concat("{",name,"} employee working in department
{",dno,"} earns Rs. {",sal,"}");
    -> select result;
    -> end loop;
    -> close c;
    -> end $$
Query OK, 0 rows affected (0.06 sec)



MariaDB [preksha]> call emp_cur()$$
+-----------------------------------------------------------------------+
| result                                                                |
+-----------------------------------------------------------------------+
| {preksha} employee working in department {D01} earns Rs. {23000.00} |
+-----------------------------------------------------------------------+
1 row in set (0.00 sec)

+-----------------------------------------------------------------------+
| result                                                                |
+-----------------------------------------------------------------------+
| {prerak} employee working in department {D01} earns Rs. {25000.00} |
+-----------------------------------------------------------------------+
1 row in set (0.01 sec)

+-----------------------------------------------------------------------+
| result                                                                |
+-----------------------------------------------------------------------+
| {dhruvin} employee working in department {D02} earns Rs. {7000.00} |
+-----------------------------------------------------------------------+
1 row in set (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

************************************************************************
************************************************************************
Q. 2 :  Create a cursor for updating the salary of emp working in deptno
10 by 20%.
      If any rows are affected than display the no of rows affected. Use
implicit cursor.

************************************************************************
************************************************************************
MariaDB [preksha]> select * from emp$$
+-------+---------+----------+--------+
| empno | empname | salary   | deptno |
+-------+---------+----------+--------+
|     1 | preksha | 23000.00 |     21 |
|     2 | prerak  | 25000.00 |     10 |
|     3 | dhruvin |  7000.00 |     10 |
+-------+---------+----------+--------+
3 rows in set (0.00 sec)
```

```
MariaDB [preksha]> create procedure emp_updt_impl()
    -> begin
    -> declare cnt int;
    -> update emp set salary=salary+(salary*0.2) where deptno=10;
    -> set cnt=row_count();
    -> if cnt>0 then
    -> select concat("Number of rows affected: ",cnt);
    -> else
    -> select "No rows are Affected!!";
    -> end if;
    ->
    -> end $$
Query OK, 0 rows affected (0.06 sec)



MariaDB [preksha]> call emp_updt_impl()$$
+----------------------------------------+
| concat("Number of rows affected: ",cnt) |
+----------------------------------------+
| Number of rows affected: 2             |
+----------------------------------------+
1 row in set (0.14 sec)

Query OK, 0 rows affected (0.16 sec)

**************************************************************************
**************************************************************************
Q. 3 :  Create a cursor for updating the salary of emp working in deptno
10 by 20%.
      Use explicit cursor.
      EMP(empno, empname, salary, deptno);

**************************************************************************
**************************************************************************
MariaDB [preksha]> select * from emp$$
+-------+---------+----------+--------+
| empno | empname | salary   | deptno |
+-------+---------+----------+--------+
|     1 | preksha | 23000.00 |     21 |
|     2 | prerak  | 24000.00 |     10 |
|     3 | dhruvin |  8000.00 |     10 |
|     4 | mahi    | 20000.00 |     11 |
+-------+---------+----------+--------+
4 rows in set (0.00 sec)


MariaDB [preksha]> create procedure emp_updt_expl()
    -> begin
    -> declare flag int default 0;
    -> declare flag2 int default 0;
    -> declare dno int;
    -> declare eid varchar(10);
    ->
```

```
    -> declare c cursor for select empno,deptno from emp;
    -> declare continue handler for not found set flag=1;
    -> open c;
    -> updt_sal:loop
    -> fetch c into eid,dno;
    -> if flag=1 then
    -> leave updt_sal;
    -> end if;
    -> if dno=10 then
    -> set flag2=1;
    -> update emp set salary=salary+(salary*0.2) where empno=eid ;
    -> end if;
    -> end loop;
    ->
    -> close c;
    ->
    -> if flag2=1 then
    -> select "Salary Updated";
    -> else
    -> select "dept name not found!!";
    -> end if;
    ->
    -> end $$
Query OK, 0 rows affected (0.06 sec)



MariaDB [preksha]> call emp_updt_expl() $$
+-----------------+
| Salary Updated |
+-----------------+
| Salary Updated |
+-----------------+
1 row in set (0.08 sec)

Query OK, 0 rows affected (0.10 sec)



MariaDB [preksha]> select * from emp$$
+-------+---------+----------+--------+
| empno | empname | salary   | deptno |
+-------+---------+----------+--------+
|     1 | preksha | 23000.00 |     21 |
|     2 | prerak  | 28800.00 |     10 |
|     3 | dhruvin |  9600.00 |     10 |
|     4 | mahi    | 20000.00 |     11 |
+-------+---------+----------+--------+
4 rows in set (0.00 sec)

********************************************************************************
********************************************************************************
```

Q. 4 : WAP that will display the name, department and salary of the first 10 employees
getting the highest salary.

```
**************************************************************************
**************************************************************************
MariaDB [preksha]> select * from emp$$
+------+---------+-----------+--------+
| id   | name    | d_name    | salary |
+------+---------+-----------+--------+
|    1 | preksha | HR        |  21000 |
|    2 | prerak  | Accounts  |   6000 |
|    3 | mahi    | Marketing |  23000 |
|    4 | dhruvin | Accounts  |  31000 |
|    5 | kashish | HR        |  25000 |
|    6 | jash    | HR        |  32000 |
|    7 | jay     | Accounts  |  18000 |
|    8 | virati  | Marketing |   9000 |
|    9 | abhi    | Accounts  |  17000 |
|   10 | jeet    | HR        |  30000 |
|   11 | yash    | HR        |   5000 |
|   12 | milind  | IT        |   6000 |
|   13 | pushti  | Marketing |  20000 |
|   14 | pratik  | Accounts  |   1000 |
|   15 | neel    | HR        |   4000 |
|   16 | hemang  | Accounts  |   4000 |
|   17 | nisarg  | IT        |  20000 |
+------+---------+-----------+--------+
17 rows in set (0.00 sec)


MariaDB [preksha]> create procedure cur_emp10()
    -> begin
    -> declare nm varchar(30);
    -> declare dp varchar(30);
    -> declare sal int;
    -> declare flag int default 0;
    ->
    -> declare c cursor for select name,d_name,salary from emp order by
salary desc limit 10;
    -> declare continue handler for not found set flag=1;
    ->
    -> open c;
    ->
    -> emp_sal: loop
    -> fetch c into nm,dp,sal;
    -> if flag=1 then
    -> leave emp_sal;
    -> end if;
    -> select nm as Name,dp as "Department name",sal as Salary;
    -> end loop;
    ->
    -> close c;
    -> end $$
Query OK, 0 rows affected (0.13 sec)
```

```
MariaDB [preksha]> call cur_emp10()$$
+------+-----------------+--------+
| Name | Department name | Salary |
+------+-----------------+--------+
| jash | HR              |  32000 |
+------+-----------------+--------+
1 row in set (0.00 sec)

+---------+-----------------+--------+
| Name    | Department name | Salary |
+---------+-----------------+--------+
| dhruvin | Accounts        |  31000 |
+---------+-----------------+--------+
1 row in set (0.01 sec)

+------+-----------------+--------+
| Name | Department name | Salary |
+------+-----------------+--------+
| jeet | HR              |  30000 |
+------+-----------------+--------+
1 row in set (0.01 sec)

+---------+-----------------+--------+
| Name    | Department name | Salary |
+---------+-----------------+--------+
| kashish | HR              |  25000 |
+---------+-----------------+--------+
1 row in set (0.01 sec)

+------+-----------------+--------+
| Name | Department name | Salary |
+------+-----------------+--------+
| mahi | Marketing       |  23000 |
+------+-----------------+--------+
1 row in set (0.02 sec)

+---------+-----------------+--------+
| Name    | Department name | Salary |
+---------+-----------------+--------+
| preksha | HR              |  21000 |
+---------+-----------------+--------+
1 row in set (0.02 sec)

+--------+-----------------+--------+
| Name   | Department name | Salary |
+--------+-----------------+--------+
| nisarg | IT              |  20000 |
+--------+-----------------+--------+
1 row in set (0.03 sec)
```

```
+--------+-----------------+--------+
| Name   | Department name | Salary |
+--------+-----------------+--------+
| pushti | Marketing       |  20000 |
+--------+-----------------+--------+
1 row in set (0.03 sec)

+------+-----------------+--------+
| Name | Department name | Salary |
+------+-----------------+--------+
| jay  | Accounts        |  18000 |
+------+-----------------+--------+
1 row in set (0.04 sec)

+------+-----------------+--------+
| Name | Department name | Salary |
+------+-----------------+--------+
| abhi | Accounts        |  17000 |
+------+-----------------+--------+
1 row in set (0.04 sec)

Query OK, 0 rows affected (0.05 sec)
```

****
****

Q. 5 : WAP using parameterized cursor to display all the information of employee living in
specified city. Ask the city from user.

****
****

```
MariaDB [preksha]> select * from emp10$$
+-------+---------+----------+-----------+
| empno | empname | salary   | city      |
+-------+---------+----------+-----------+
|     1 | preksha | 23000.00 | ahmedabad |
|     2 | prerak  | 25000.00 | Ahmedabad |
|     3 | jeet    |  7000.00 | vadodara  |
|     4 | dhruvin | 23000.00 | Rajkot    |
|     5 | kriya   | 15000.00 | surat     |
+-------+---------+----------+-----------+
5 rows in set (0.00 sec)



MariaDB [preksha]> create procedure emp_slt_city(in_city varchar(30))
    -> begin
    -> declare flag int default 0;
    -> declare cnt int default 0;
    -> declare veno varchar(10);
    -> declare vename,vdeptnm,vcity varchar(30);
    -> declare vsal decimal(8,2) default 0;
    -> declare cur CURSOR for select * from emp10 where city=in_city;
    -> declare continue handler for not found set flag=1;
    -> select count(*) into cnt from emp10 where city=in_city;
```

```
    ->
    -> if cnt=0 then
    -> select "Employee not Found" as "Error!";
    -> else
    -> open cur;
    -> lp: loop
    -> fetch cur into veno,vename,vsal,vcity;
    -> if flag=1 then
    -> leave lp;
    -> end if;
    -> select veno as "NO",vename as "Name",vsal as "Salary",vcity as
"city";
    -> end loop lp;
    -> close cur;
    -> end if;
    -> end $$
Query OK, 0 rows affected (0.06 sec)



MariaDB [preksha]> call emp_slt_city('Ahmedabad')$$
+------+---------+----------+-----------+
| NO   | Name    | Salary   | city      |
+------+---------+----------+-----------+
| 1    | preksha | 23000.00 | ahmedabad |
+------+---------+----------+-----------+
1 row in set (0.00 sec)

+------+--------+----------+-----------+
| NO   | Name   | Salary   | city      |
+------+--------+----------+-----------+
| 2    | prerak | 25000.00 | Ahmedabad |
+------+--------+----------+-----------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)



MariaDB [preksha]> call emp_slt_city('himatnagar')$$
+--------------------+
| Error!             |
+--------------------+
| Employee not Found |
+--------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

********************************************************************************
********************************************************************************
```

Q. 6 : WAP which display the sum of salary department wise.

```
**************************************************************************
**************************************************************************
MariaDB [preksha]> select * from emp11$$
+-------+---------+----------+--------+
| empno | empname | salary   | deptno |
+-------+---------+----------+--------+
|     1 | preksha | 20000.00 |     21 |
|     2 | Mahi    | 21000.00 |     10 |
|     3 | prerak  | 27000.00 |     10 |
|     4 | Dhruvin |  8000.00 |     21 |
|     5 | jeet    | 22000.00 |     11 |
+-------+---------+----------+--------+
5 rows in set (0.00 sec)



MariaDB [preksha]> create procedure emp_sal_dept()
    -> begin
    -> declare sal decimal(8,2);
    -> declare dno,flag int;
    ->
    -> declare c cursor for select deptno,sum(salary) from emp11 group by
deptno;
    -> declare continue handler for not found set flag=1;
    ->
    -> open c;
    ->
    -> emp_loop: loop
    -> fetch c into dno,sal;
    -> if flag=1 then
    -> leave emp_loop;
    -> end if;
    -> select dno,sal;
    -> end loop;
    -> close c;
    -> end $$
Query OK, 0 rows affected (0.13 sec)



MariaDB [preksha]> call emp_sal_dept()$$
+------+----------+
| dno  | sal      |
+------+----------+
|   10 | 48000.00 |
+------+----------+
1 row in set (0.00 sec)
```

```
+------+----------+
| dno  | sal      |
+------+----------+
|   11 | 22000.00 |
+------+----------+
1 row in set (0.01 sec)

+------+----------+
| dno  | sal      |
+------+----------+
|   21 | 28000.00 |
+------+----------+
1 row in set (0.02 sec)

Query OK, 0 rows affected (0.03 sec)
```

****

****

Q. 7 : Create a cursor to generate defferent two tables from one master table.

Student(Rno, Name, Std, B_date, Sex);
Girl_Table(Rno, Name, Std, B_date);
Boy_Table(Rno, Name, Std, B_date);

First fetch the row from Student table. If sex is 'M' then insert that row in Boy_Table
and if 'F' then insert that row in Girl_Table.
In both table Rollno entry must be in Sequence(Using create sequence command).

****

****

```
MariaDB [preksha]> select * from student$$
+------+---------+------------+------+
| Rno  | Name    | B_date     | Sex  |
+------+---------+------------+------+
|    1 | preksha | 1998-10-18 | F    |
|    2 | prerak  | 2000-01-25 | M    |
|    3 | viru    | 1999-02-06 | F    |
|    4 | jeet    | 1998-01-05 | M    |
|    5 | dhruvin | 1998-10-23 | M    |
+------+---------+------------+------+
5 rows in set (0.00 sec)



MariaDB [preksha]> desc boy$$
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| Rno    | int(11)     | YES  |     | NULL    |       |
| Name   | varchar(30) | YES  |     | NULL    |       |
| B_date | date        | YES  |     | NULL    |       |
| Sex    | varchar(1)  | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
4 rows in set (0.03 sec)
```

```
MariaDB [preksha]> desc girl$$
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| Rno    | int(11)     | YES  |     | NULL    |       |
| Name   | varchar(30) | YES  |     | NULL    |       |
| B_date | date        | YES  |     | NULL    |       |
| Sex    | varchar(1)  | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
4 rows in set (0.10 sec)
```

```
MariaDB [preksha]> create procedure student_divide()
    -> begin
    -> declare rn,flag int;
    -> declare nm varchar(30);
    -> declare bd date;
    -> declare s varchar(1);
    ->
    -> declare c cursor for select * from student;
    -> declare continue handler for not found set flag=1;
    -> create or replace table Boy(Rno int,Name varchar(30),B_date
date,Sex varchar(1));
    -> create or replace table Girl(Rno int,Name varchar(30),B_date
date,Sex varchar(1));
    -> open c;
    -> divide_loop:loop
    -> fetch c into rn,nm,bd,s;
    -> if flag=1 then
    -> leave divide_loop;
    -> end if;
    -> if s="M" then
    -> insert into Boy values(rn,nm,bd,s);
    -> else
    -> insert into Girl values(rn,nm,bd,s);
    -> end if;
    -> end loop;
    -> close c;
    -> end $$
Query OK, 0 rows affected (0.14 sec)
```

```
MariaDB [preksha]> call student_divide()$$
Query OK, 0 rows affected (1.12 sec)
```

```
MariaDB [preksha]> select * from girl$$
+------+---------+------------+------+
| Rno  | Name    | B_date     | Sex  |
+------+---------+------------+------+
|    1 | preksha | 1998-10-18 | F    |
|    3 | viru    | 1999-02-06 | F    |
+------+---------+------------+------+
2 rows in set (0.00 sec)


MariaDB [preksha]> select * from boy$$
+------+---------+------------+------+
| Rno  | Name    | B_date     | Sex  |
+------+---------+------------+------+
|    2 | prerak  | 2000-01-25 | M    |
|    4 | jeet    | 1998-01-05 | M    |
|    5 | dhruvin | 1998-10-23 | M    |
+------+---------+------------+------+
3 rows in set (0.00 sec)
```

****
****

ASSIGNMENT - 3

****
****

Triggers

****
****

Q. 1 : Write a Trigger that stores the old data table of student table in student_backup while updating
the student table.

****
****

```
MariaDB [preksha]> create table student_backup(
    -> stud_id int,
    -> stud_name varchar(20),
    -> m1 int,
    -> m2 int,
    -> m3 int,
    -> city varchar(20),
    -> percent decimal(5,2))$$
Query OK, 0 rows affected (0.27 sec)
```

```
MariaDB [preksha]> select * from student$$
+---------+-----------+------+------+------+-------------+---------+
| stud_id | stud_name | m1   | m2   | m3   | city        | percent |
+---------+-----------+------+------+------+-------------+---------+
|       1 | preksha   |   46 |   42 |   38 | ahmedabad   |   75.70 |
|       2 | prerak    |   39 |   32 |   45 | ahmedabad   |   73.00 |
|       3 | abhi      |   35 |   43 |   40 | himmatnagar |   70.50 |
|       4 | het       |   45 |   33 |   46 | vadodara    |   72.50 |
|       5 | kriya     |   46 |   43 |   34 | surat       |   77.90 |
+---------+-----------+------+------+------+-------------+---------+
5 rows in set (0.13 sec)
```

```
MariaDB [preksha]> create trigger student_backup before update on student
    -> for each row
    -> begin
    -> insert into student_backup
values(old.stud_id,old.stud_name,old.m1,old.m2,old.m3,old.city,old.percen
t);
    -> end $$
Query OK, 0 rows affected (0.13 sec)
```

```
MariaDB [preksha]> update student set m1 = 45,m2 = 39,m3 = 40 where
stud_id = 2$$
Query OK, 1 row affected (0.10 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
MariaDB [preksha]> select * from student $$
+---------+-----------+------+------+------+-------------+---------+
| stud_id | stud_name | m1   | m2   | m3   | city        | percent |
+---------+-----------+------+------+------+-------------+---------+
|       1 | preksha   |   46 |   42 |   38 | ahmedabad   |   75.70 |
|       2 | prerak    |   45 |   39 |   40 | ahmedabad   |   73.00 |
|       3 | abhi      |   35 |   43 |   40 | himmatnagar |   70.50 |
|       4 | het       |   45 |   33 |   46 | vadodara    |   72.50 |
|       5 | kriya     |   46 |   43 |   34 | surat       |   77.90 |
+---------+-----------+------+------+------+-------------+---------+
5 rows in set (0.00 sec)
```

```
MariaDB [preksha]> select * from student_backup $$
+---------+-----------+------+------+------+-----------+---------+
| stud_id | stud_name | m1   | m2   | m3   | city      | percent |
+---------+-----------+------+------+------+-----------+---------+
|       2 | prerak    |   39 |   32 |   45 | ahmedabad |   73.00 |
+---------+-----------+------+------+------+-----------+---------+
1 row in set (0.00 sec)
```

```
**************************************************************************
**************************************************************************
Q. 2 : Write a trigger, that ensures the empno of emp table is in a
format 'E00001' (empno must start
       with 'E' and must be 6 characters long). If not, than complete
empno with this format before
       inserting into the employee table.
**************************************************************************
**************************************************************************
MariaDB [preksha]> create or replace trigger emp_no before insert on
emp12
    -> for each row
    -> begin
    -> declare len int;
    -> declare temp varchar(6);
    -> declare chr varchar(1);
    -> set len=length(new.empno);
    -> if len>6 then
    -> signal sqlstate "45000"
    -> set message_text="Maximum 6 charactors!!";
    -> else
    -> set chr=substr(new.empno,1,1);
    -> if chr="E" then
    -> set temp=substr(new.empno,2);
    -> set new.empno="E";
    -> if len=1 then
    -> set new.empno=concat(new.empno,"00000");
    -> elseif len=2 then
    -> set new.empno=concat(new.empno,"0000",temp);
    -> elseif len=3 then
    -> set new.empno=concat(new.empno,"000",temp);
    -> elseif len=4 then
    -> set new.empno=concat(new.empno,"00",temp);
    -> elseif len=5 then
    -> set new.empno=concat(new.empno,"0",temp);
    -> elseif len=6 then
    -> set new.empno=concat(new.empno,temp);
    -> end if;
    ->
    -> else
    -> set temp=new.empno;
    -> set new.empno="E";
    ->
    -> if len=1 then
    -> set new.empno=concat(new.empno,"0000",temp);
    -> elseif len=2 then
    -> set new.empno=concat(new.empno,"000",temp);
    -> elseif len=3 then
    -> set new.empno=concat(new.empno,"00",temp);
    -> elseif len=4 then
    -> set new.empno=concat(new.empno,"0",temp);
    -> elseif len=5 then
    -> set new.empno=concat(new.empno,temp);
```

```
    -> elseif len=6 then
    -> signal sqlstate "45000"
    -> set message_text="Invalid Input!!";
    ->
    -> end if;
    -> end if;
    -> end if;
    -> end $$
Query OK, 0 rows affected (0.09 sec)

MariaDB [preksha]> insert into emp12
values('2','prerak',24000,'manager')$$
Query OK, 1 row affected (0.13 sec)

MariaDB [preksha]> select * from emp12$$
+--------+---------+--------+-------------+
| empno  | name    | salary | designation |
+--------+---------+--------+-------------+
| 1      | preksha |  23000 | Manager     |
| E00002 | prerak  |  24000 | manager     |
+--------+---------+--------+-------------+
2 rows in set (0.00 sec)

MariaDB [preksha]> insert into emp12
values('12345678','prerak',24000,'manager')$$
ERROR 1644 (45000): Maximum 6 charactors!!
```

****
****

Q.3 : Write a trigger which checks the age of employee while inserting the record in emp table. If it is negative than generate the error and display proper message.

****
****

```
MariaDB [preksha]> select * from person$$
+------+---------+------+
| p_id | name    | age  |
+------+---------+------+
|    1 | preksha |   21 |
|    2 | prerak  |   20 |
|    3 | kashish |   15 |
+------+---------+------+
3 rows in set (0.00 sec)


MariaDB [preksha]> create trigger person_trg before update on person
    -> for each row
    -> begin
    -> if new.age < 0 then
    -> signal sqlstate '80000'
    -> set message_text = 'age must be greater than 0';
```

```
    -> end if;
    -> end$$
Query OK, 0 rows affected (0.16 sec)


MariaDB [preksha]> update person set age = -2 where p_id = 1$$
ERROR 1644 (80000): age must be greater than 0
```

****
****

Q.4 : Write a trigger which converts the employee name in upper case if it is inserted in any other case.
Change should be done before the insertion only.

****
****

```
MariaDB [preksha]> select * from emp2$$
+------+---------+
| e_id | e_name  |
+------+---------+
|    1 | preksha |
|    2 | prerak  |
|    3 | Mahi    |
+------+---------+
3 rows in set (0.00 sec)


MariaDB [preksha]> create trigger chng_case before insert on emp2
    -> for each row
    -> begin
    -> set new.e_name = upper(new.e_name);
    -> end$$
Query OK, 0 rows affected (0.19 sec)


MariaDB [preksha]> insert into emp2 values(4,'dhruvin')$$
Query OK, 1 row affected (0.00 sec)


MariaDB [preksha]> select * from emp2$$
+------+---------+
| e_id | e_name  |
+------+---------+
|    1 | preksha |
|    2 | prerak  |
|    3 | Mahi    |
|    4 | DHRUVIN |
+------+---------+
4 rows in set (0.00 sec)
```

```
**************************************************************************
**************************************************************************
Q.5 :  WAT that stores the data of emp table in emp_backup table for
every delete operation and
       store the old data for every update operation.
       EMP(Empno, Empname, salary);
       Emp_Backup(Empno,Empname,Date_of_operation,Type_of_operation
(i.e.update or delete));

**************************************************************************
**************************************************************************
MariaDB [preksha]> select * from emp3$$
+--------+----------+----------+
| emp_id | emp_name | salary   |
+--------+----------+----------+
|      1 | preksha  | 20000.00 |
|      2 | riya     | 10000.00 |
|      3 | kriya    | 14000.00 |
+--------+----------+----------+
3 rows in set (0.00 sec)


MariaDB [preksha]> create trigger emp_backup before update on emp3
    -> for each row
    -> begin
    -> insert into emp_backup
values(old.emp_id,old.emp_name,current_date(),'UPDATED');
    -> end $$
Query OK, 0 rows affected (0.24 sec)



MariaDB [preksha]> update emp3 set salary = 22000 where emp_id = 1$$
Query OK, 1 row affected (0.11 sec)
Rows matched: 1  Changed: 1  Warnings: 0



MariaDB [preksha]> select * from emp3$$
+--------+----------+----------+
| emp_id | emp_name | salary   |
+--------+----------+----------+
|      1 | preksha  | 22000.00 |
|      2 | riya     | 10000.00 |
|      3 | kriya    | 14000.00 |
+--------+----------+----------+
3 rows in set (0.00 sec)
```

```
MariaDB [preksha]> select * from emp_backup$$
+--------+----------+------------+----------+
| emp_id | emp_name | d_o_oper   | t_o_oper |
+--------+----------+------------+----------+
|      1 | preksha  | 2020-05-06 | UPDATED  |
+--------+----------+------------+----------+
1 row in set (0.00 sec)
```

****
****

Q. 6 : WAT which display the message 'Updating','Deleting' or 'Inserting' when Update, Delete or
Insert operation is performed on the emp table respectively.

****
****

```
MariaDB [preksha]> select * from emp13;
+-------+---------+--------+
| empno | empname | salary |
+-------+---------+--------+
|     1 | preksha |  21000 |
|     2 | prerak  |  26000 |
|     3 | mahi    |  23000 |
|     4 | dhruvin |  31000 |
|     5 | jeet    |  15000 |
+-------+---------+--------+
5 rows in set (0.00 sec)
```

```
MariaDB [preksha]> create or replace trigger emp_update after update on
emp13
    -> for each row
    -> begin
    -> signal sqlstate "45000"
    -> set message_text="Updating";
    -> end $$
Query OK, 0 rows affected (0.15 sec)

MariaDB [preksha]> create or replace trigger emp_insert after insert on
emp13
    -> for each row
    -> begin
    -> signal sqlstate "45000"
    -> set message_text="Inserting";
    -> end $$
Query OK, 0 rows affected (0.09 sec)


MariaDB [preksha]> create or replace trigger emp_delete before delete on
emp13
    -> for each row
    -> begin
    -> signal sqlstate "45000"
```

```
    -> set message_text="Deleting";
    -> end $$
Query OK, 0 rows affected (0.09 sec)

MariaDB [preksha]> update emp13 set salary = 8000 where empno = 1$$
ERROR 1644 (45000): Updating


MariaDB [preksha]> insert into emp13 values(6,'virati',30000)$$
ERROR 1644 (45000): Inserting


MariaDB [preksha]> delete from emp13 where empname = 'preksha'$$
ERROR 1644 (45000): Deleting


**************************************************************************
**************************************************************************
Q. 7 : WAT which generate an error if any user try to delete from
product_master table on weekends
       (i.e. Saturday and Sunday).
**************************************************************************
**************************************************************************
MariaDB [preksha]> create trigger pro_del before delete on product_master
    -> for each row
    -> begin
    -> declare day int;
    -> set day=dayofweek(curdate());
    -> if day=1 or day=7 then
    -> signal sqlstate "45000"
    -> set message_text="You can not delete on weekends!!(i.e. Saturday
and Sunday).";
    -> end if;
    -> end $$
Query OK, 0 rows affected (0.14 sec)



MariaDB [preksha]> delete from product_master where p_id=1;
ERROR 1644 (45000): You can not delete on weekends!!(i.e. Saturday and
Sunday).

**************************************************************************
**************************************************************************
Q. 8 : We have two tables student_mast and stu_log. student_mast have
three columns
      STUDENT_ID, NAME, ST_CLASS. stu_log table has two columns user_id
and description.
      WAT which inserts the student details in stu_log table as soon as
we promote the students in
      student master table( e.g. when a student is promoted from sem 2 to
3, auto entry in log table)
```

```
*****************************************************************************
*****************************************************************************
MariaDB [preksha]> select * from stud$$
+------+---------+-------+------+
| s_id | s_name  | class | sem  |
+------+---------+-------+------+
|    1 | preksha | MCA   |    2 |
|    2 | prerak  | BCA   |    4 |
|    3 | virati  | BCA   |    2 |
+------+---------+-------+------+
3 rows in set (0.00 sec)


MariaDB [preksha]> create table stu_log(
    -> s_id int,
    -> descr varchar(200))$$
Query OK, 0 rows affected (0.31 sec)


MariaDB [preksha]> create trigger stu_backup before update on stud
    -> for each row
    -> begin
    -> declare descri varchar(200);
    -> set descri = concat('student promoted from semester ',old.sem,'to
sem ',new.sem);
    -> insert into stu_log values(new.s_id,descri);
    -> end$$
Query OK, 0 rows affected (0.17 sec)


MariaDB [preksha]> update stud set sem = 3 where s_id = 1$$
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0



MariaDB [preksha]> select * from stud$$
+------+---------+-------+------+
| s_id | s_name  | class | sem  |
+------+---------+-------+------+
|    1 | preksha | MCA   |    3 |
|    2 | prerak  | BCA   |    4 |
|    3 | virati  | BCA   |    2 |
+------+---------+-------+------+
3 rows in set (0.00 sec)


MariaDB [preksha]> select * from stu_log$$
+------+-----------------------------------------+
| s_id | descr                                   |
+------+-----------------------------------------+
|    1 | student promoted from semester 2to sem 3 |
+------+-----------------------------------------+
1 row in set (0.00 sec)
```

```
**************************************************************************
**************************************************************************
Q. 9 : WAT to calculate the Income Tax amount and insert it in emp table.
EMP(emp_no,emp_name,
      emp_income, income_tax);
      If emp_income <100000 and >=50000 then incometax = 10%
      If emp_income <200000 and >=100000 then incometax = 15%
      If emp_income <300000 and >=200000 then incometax = 20%
**************************************************************************
**************************************************************************
MariaDB [preksha]> create table emp4(
    -> e_no int,
    -> e_name varchar(50),
    -> e_inc decimal(8,2),
    -> e_inc_tax decimal(5,2))$$
Query OK, 0 rows affected (0.31 sec)


MariaDB [preksha]> create trigger emp_inc before insert on emp4
    -> for each row
    -> begin
    -> if new.e_inc < 100000 AND new.e_inc >= 50000 then
    -> set new.e_inc_tax = (new.e_inc) * (10/100);
    -> elseif new.e_inc < 200000 AND  new.e_inc >= 100000 then
    -> set new.e_inc_tax = (new.e_inc) * (15/100);
    -> elseif new.e_inc < 300000 AND  new.e_inc >= 200000 then
    -> set new.e_inc_tax = (new.e_inc) * (20/100);
    -> end if;
    -> end $$
Query OK, 0 rows affected (0.13 sec)


MariaDB [preksha]> insert into emp4(e_no,e_name,e_inc) values
(1,'Preksha','300000')$$
Query OK, 1 row affected (0.00 sec)


MariaDB [preksha]> select * from emp4$$
+------+---------+----------+-----------+
| e_no | e_name  | e_inc    | e_inc_tax |
+------+---------+----------+-----------+
|    1 | Preksha | 300000.00 |      NULL |
+------+---------+----------+-----------+
1 row in set (0.00 sec)


**************************************************************************
**************************************************************************
Q. 10 : This example is divided in three categories : Insert, Update and
Delete
```

a. Write a trigger which updates the sale value if customer already exists else create
new entry of customer.
b. Update : If the customer is updating , WAT to update the sales value by
incrementing the Sale_vale field.
c. Delete : If the customer is deleting , WAT to update the sales value by
decrementing the Sale_vale field.

****
****
Q10.(a) :insert
****
****

```
MariaDB [preksha]> create table cutomer_t(
    -> cus_id int primary key,
    -> value int) $$
Query OK, 0 rows affected (0.21 sec)



MariaDB [preksha]> insert into cutomer_t values(1,10) $$
Query OK, 1 row affected (0.14 sec)

MariaDB [preksha]> insert into cutomer_t values(2,30) $$
Query OK, 1 row affected (0.05 sec)

MariaDB [preksha]> insert into cutomer_t values(3,20) $$
Query OK, 1 row affected (0.05 sec)

MariaDB [preksha]> insert into cutomer_t values(4,50) $$
Query OK, 1 row affected (0.20 sec)



MariaDB [preksha]> select * from cutomer_t $$
+--------+-------+
| cus_id | value |
+--------+-------+
|      1 |    10 |
|      2 |    30 |
|      3 |    20 |
|      4 |    50 |
+--------+-------+
4 rows in set (0.00 sec)


MariaDB [preksha]> create table sales_t(
    -> s_id int primary key,
    -> cus_id int REFERENCES cutomer_t,
    -> amt int) $$
Query OK, 0 rows affected (0.33 sec)
```

```
MariaDB [preksha]> create trigger sales_tr before insert on sales_t
    -> for each row
    -> begin
    -> declare cnt int default 0;
    -> select count(*) into cnt from cutomer_t where cus_id = new.cus_id;
    -> if cnt > 0 then
    -> update cutomer_t set value = value + new.amt where cus_id =
new.cus_id;
    -> else
    -> insert into cutomer_t values(new.cus_id,new.amt);
    -> end if;
    -> end $$
Query OK, 0 rows affected (0.10 sec)



MariaDB [preksha]> insert into sales_t values(1,1,400) $$
Query OK, 1 row affected (0.08 sec)

MariaDB [preksha]> insert into sales_t values(2,2,500) $$
Query OK, 1 row affected (0.13 sec)

MariaDB [preksha]> insert into sales_t values(3,3,700) $$
Query OK, 1 row affected (0.06 sec)

MariaDB [preksha]> insert into sales_t values(4,5,700) $$
Query OK, 1 row affected (0.14 sec)


ariaDB [preksha]> select * from sales_t $$
+------+--------+------+
| s_id | cus_id | amt  |
+------+--------+------+
|    1 |      1 |  400 |
|    2 |      2 |  500 |
|    3 |      3 |  700 |
|    4 |      5 |  700 |
+------+--------+------+
4 rows in set (0.00 sec)



MariaDB [preksha]> select * from cutomer_t $$
+--------+-------+
| cus_id | value |
+--------+-------+
|      1 |   410 |
|      2 |   530 |
|      3 |   720 |
|      4 |    50 |
|      5 |   700 |
+--------+-------+
5 rows in set (0.00 sec)
```

```
**************************************************************************
**************************************************************************
Q10.(b) : update
**************************************************************************
**************************************************************************
MariaDB [preksha]> create trigger updt_trg before update on sales_t
    -> for each row
    -> begin
    -> update cutomer_t set value = value +(new.amt-old.amt) where cus_id
= new.cus_id;
    -> end $$
Query OK, 0 rows affected (0.18 sec)



MariaDB [preksha]> update sales_t set amt = 200 where cus_id = 3 $$
Query OK, 1 row affected (0.08 sec)
Rows matched: 1  Changed: 1  Warnings: 0




MariaDB [preksha]> select * from sales_t $$
+------+--------+------+
| s_id | cus_id | amt  |
+------+--------+------+
|    1 |      1 |  400 |
|    2 |      2 |  500 |
|    3 |      3 |  200 |
|    4 |      5 |  700 |
+------+--------+------+
4 rows in set (0.00 sec)


MariaDB [preksha]> select * from cutomer_t $$
+--------+-------+
| cus_id | value |
+--------+-------+
|      1 |   410 |
|      2 |   530 |
|      3 |   220 |
|      4 |    50 |
|      5 |   700 |
+--------+-------+
5 rows in set (0.00 sec)

**************************************************************************
**************************************************************************
Q.10 (c) : delete
**************************************************************************
**************************************************************************
```

```
MariaDB [preksha]> create trigger del_trg before delete on sales_t
    -> for each row
    -> begin
    -> update cutomer_t set value = value - old.amt where cus_id =
old.cus_id;
    -> end $$
Query OK, 0 rows affected (0.08 sec)



MariaDB [preksha]> select * from sales_t $$
+------+--------+------+
| s_id | cus_id | amt  |
+------+--------+------+
|    1 |      1 |  400 |
|    3 |      3 |  200 |
|    4 |      5 |  700 |
+------+--------+------+
3 rows in set (0.00 sec)



MariaDB [preksha]> select * from cutomer_t $$
+--------+-------+
| cus_id | value |
+--------+-------+
|      1 |   410 |
|      2 |    30 |
|      3 |   220 |
|      4 |    50 |
|      5 |   700 |
+--------+-------+
5 rows in set (0.00 sec)
**************************************************************************
**************************************************************************
Q. 11 : Wirte a program to create trigger signal to restrict entering
negative value
       in balance.
**************************************************************************
**************************************************************************
MariaDB [preksha]> select * from emp $$
+--------+----------+-----------+------------+------------+
| emp_id | emp_name | emp_job   | emp_salary | DOJ        |
+--------+----------+-----------+------------+------------+
|      1 | Preksha  | Developer |      20000 | 2005-10-25 |
|      2 | Prerak   | DBA       |      18000 | 1998-12-18 |
|      3 | Dhruvin  | DBA       |      15000 | 2010-05-15 |
|      4 | Aman     | Developer |      25000 | 2002-09-10 |
|      5 | Mahi     | Designer  |      15000 | 2015-10-20 |
|      7 | Abhi     | Designer  |      23000 | 2010-12-12 |
|      8 | rishi    | tester    |         -1 | 2000-10-11 |
+--------+----------+-----------+------------+------------+
7 rows in set (0.00 sec)
```

```
MariaDB [preksha]> create trigger neg_inc_trg before update on emp
    -> for each row
    -> begin
    -> if new.emp_salary < 0 then
    -> SIGNAL SQLSTATE '80000'
    -> set MESSAGE_TEXT = 'Your Account Balance cannot be Less than 0';
    -> end if;
    -> end $$
Query OK, 0 rows affected (0.17 sec)


MariaDB [preksha]> update emp set emp_salary = -2 where emp_id = 7 $$
ERROR 1644 (80000): Your Account Balance cannot be Less than 0


MariaDB [preksha]> update emp set emp_salary = 1000 where emp_id = 8 $$
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0



MariaDB [preksha]> select * from emp $$
+--------+----------+-----------+------------+------------+
| emp_id | emp_name | emp_job   | emp_salary | DOJ        |
+--------+----------+-----------+------------+------------+
|      1 | Preksha  | Developer |      20000 | 2005-10-25 |
|      2 | Prerak   | DBA       |      18000 | 1998-12-18 |
|      3 | Dhruvin  | DBA       |      15000 | 2010-05-15 |
|      4 | Aman     | Developer |      25000 | 2002-09-10 |
|      5 | Mahi     | Designer  |      15000 | 2015-10-20 |
|      7 | Abhi     | Designer  |      23000 | 2010-12-12 |
|      8 | rishi    | tester    |       1000 | 2000-10-11 |
+--------+----------+-----------+------------+------------+
7 rows in set (0.00 sec)
**************************************************************************
**************************************************************************
Q. 12 : Write a trigger to perform data validation using select
statement.

**************************************************************************
**************************************************************************
MariaDB [preksha]> create table emp_age(
    -> id int,
    -> age int) $$
Query OK, 0 rows affected (0.28 sec)


MariaDB [preksha]> insert into emp_age values (1,33) $$
Query OK, 1 row affected (0.13 sec)
```

```
MariaDB [preksha]> insert into emp_age values (1,23) $$
Query OK, 1 row affected (0.13 sec)


MariaDB [preksha]> insert into emp_age values (1,20) $$
Query OK, 1 row affected (0.11 sec)


MariaDB [preksha]> select * from emp_age $$
+------+------+
| id   | age  |
+------+------+
|    1 |   33 |
|    1 |   23 |
|    1 |   20 |
+------+------+
3 rows in set (0.00 sec)



MariaDB [preksha]> create trigger age_trg before update on emp_age
    -> for each row
    -> begin
    -> declare dummy int;
    -> if new.age < 0 then
    -> select 'Age is not Less than 0';
    -> end if;
    -> end $$
ERROR 1415 (0A000): Not allowed to return a result set from a trigger




MariaDB [preksha]>  create or replace trigger age_trg before update on
emp_age
    ->  for each row
    ->  begin
    ->
    ->  if new.age < 18 then
    ->  signal sqlstate '80000'
    ->  set message_text = 'Age must be greater than 18';
    ->  end if;
    ->  end $$
Query OK, 0 rows affected (0.21 sec)


MariaDB [preksha]> update emp_age set age = 20 where id = 1$$
Query OK, 3 rows affected (0.07 sec)
Rows matched: 3  Changed: 3  Warnings: 0


MariaDB [preksha]> update emp_age set age = 10 where id = 1$$
ERROR 1644 (80000): Age must be greater than 18
```

```
MariaDB [preksha]> select * from emp_age$$
+------+------+
| id   | age  |
+------+------+
|    1 |   20 |
|    1 |   20 |
|    1 |   20 |
|    2 |   15 |
+------+------+
4 rows in set (0.00 sec)
```

****
****

Q. 13 : write a example to create a sales table which provides free shipping on orders
above 500.

****
****

```
MariaDB [preksha]> create table sales_t2(
    -> s_id int,
    -> s_val int,
    -> free_sale varchar(1),
    -> disc decimal(5,2)) $$
Query OK, 0 rows affected (0.70 sec)
```

```
MariaDB [preksha]> create trigger free_shp_trg before insert on sales_t2
    -> for each row
    -> begin
    -> if new.s_val > 500 then
    -> set new.free_sale = 'y';
    -> else
    -> set new.free_sale = 'n';
    -> end if;
    -> if new.s_val > 1000 then
    -> set new.disc = new.s_val * .15;
    -> else
    -> set new.disc = 0;
    -> end if;
    -> end $$
Query OK, 0 rows affected (0.22 sec)
```

```
MariaDB [preksha]> insert into sales_t2(s_id,s_val) values (1,500) $$
Query OK, 1 row affected (0.07 sec)

MariaDB [preksha]> insert into sales_t2(s_id,s_val) values (2,1500) $$
Query OK, 1 row affected (0.07 sec)
```

```
MariaDB [preksha]> insert into sales_t2(s_id,s_val) values (3,100) $$
Query OK, 1 row affected (0.13 sec)

MariaDB [preksha]> insert into sales_t2(s_id,s_val) values (4,700) $$
Query OK, 1 row affected (0.13 sec)



MariaDB [preksha]> select * from sales_t2 $$
+------+-------+-----------+--------+
| s_id | s_val | free_sale | disc   |
+------+-------+-----------+--------+
|    1 |   500 | n         |   0.00 |
|    2 |  1500 | y         | 225.00 |
|    3 |   100 | n         |   0.00 |
|    4 |   700 | y         |   0.00 |
+------+-------+-----------+--------+
4 rows in set (0.00 sec)
```

****
****

ASSIGNMENT - 4

****
****

TRANSACTION

****
****

Q. 1 : Create a procedure to commence a transaction using auto commit.

****
****

```
MariaDB [preksha]> select * from employee$$
+------+---------+-----------+---------+
| e_id | e_name  | e_salary  | dept_id |
+------+---------+-----------+---------+
|    1 | Preksha | 303000.00 |       3 |
|    2 | Prerak  | 352000.00 |       2 |
|    3 | Aman    | 254000.00 |       4 |
|    4 | Abhi    | 204000.00 |       4 |
|    5 | Pushti  | 281000.00 |       1 |
|    6 | Mahi    | 321000.00 |       1 |
|    7 | Dhruvin | 275000.00 |       5 |
+------+---------+-----------+---------+
7 rows in set (0.12 sec)



MariaDB [preksha]> create procedure trans1(in ad_id int, in su_id int,in
amt int)
    -> begin
    -> set autocommit = 0;
```

```
    -> update employee set e_salary = e_salary + amt where e_id = ad_id;
    -> update employee set e_salary = e_salary - amt where e_id = su_id;
    -> commit;
    -> end $$
Query OK, 0 rows affected (0.15 sec)



MariaDB [preksha]> call trans1(2,1,3000) $$
Query OK, 0 rows affected (0.14 sec)



MariaDB [preksha]> select * from employee $$
+------+---------+----------+---------+
| e_id | e_name  | e_salary  | dept_id |
+------+---------+----------+---------+
|    1 | Preksha | 300000.00 |       3 |
|    2 | Prerak  | 355000.00 |       2 |
|    3 | Aman    | 254000.00 |       4 |
|    4 | Abhi    | 204000.00 |       4 |
|    5 | Pushti  | 281000.00 |       1 |
|    6 | Mahi    | 321000.00 |       1 |
|    7 | Dhruvin | 275000.00 |       5 |
+------+---------+----------+---------+
7 rows in set (0.00 sec)

**************************************************************************
**************************************************************************
Q. 2 : Create a procedure to commence a transaction using start
transaction.
**************************************************************************
**************************************************************************
MariaDB [preksha]> select * from accounts$$
+--------+-------------+-----------+
| acc_id | Branch_Name | Balance   |
+--------+-------------+-----------+
|      1 | Sabarmati   | 200000.00 |
|      2 | Chandkheda  | 350000.00 |
|      3 | Motera      | 123000.00 |
|      4 | sabarmati   | 205000.00 |
|      5 | Motera      | 400000.00 |
|      6 | jantanagar  | 300000.00 |
+--------+-------------+-----------+
6 rows in set (0.00 sec)



MariaDB [preksha]> create procedure transfer_start_trans(from_acid
int,to_acid int,amt decimal(8,2))
    -> begin
    -> start transaction;
    -> update accounts set balance = balance - amt where acc_id =
from_acid;
```

```
    -> update accounts set balance = balance + amt where acc_id =
to_acid;
    -> commit;
    -> end $$
Query OK, 0 rows affected (0.13 sec)



MariaDB [preksha]> call transfer_start_trans(3,2,3000)$$
Query OK, 0 rows affected (0.14 sec)



MariaDB [preksha]> select * from accounts$$
+--------+-------------+-----------+
| acc_id | Branch_Name | Balance   |
+--------+-------------+-----------+
|      1 | Sabarmati   | 200000.00 |
|      2 | Chandkheda  | 353000.00 |
|      3 | Motera      | 120000.00 |
|      4 | sabarmati   | 205000.00 |
|      5 | Motera      | 400000.00 |
|      6 | jantanagar  | 300000.00 |
+--------+-------------+-----------+
6 rows in set (0.00 sec)

**************************************************************************
**************************************************************************
Q. 3 : create a procedure which displays use of Savepoint with a
transaction.
**************************************************************************
**************************************************************************
MariaDB [preksha]> create table audit_log(audit_message varchar(500)) $$
Query OK, 0 rows affected (0.36 sec)


MariaDB [preksha]> desc audit_log $$
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| audit_message | varchar(500) | YES  |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
1 row in set (0.10 sec)


MariaDB [preksha]> create table location(location varchar(50),address
varchar(50),zip int(6))$$
Query OK, 0 rows affected (0.34 sec)
```

```
MariaDB [preksha]> desc location $$
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| location | varchar(50) | YES  |     | NULL    |       |
| address  | varchar(50) | YES  |     | NULL    |       |
| zip      | int(6)      | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
3 rows in set (0.02 sec)


MariaDB [preksha]> create table dept (dept_name varchar(50),location
varchar(50),m_id int)$$
Query OK, 0 rows affected (0.31 sec)


MariaDB [preksha]> desc dept $$
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| dept_name | varchar(50) | YES  |     | NULL    |       |
| location  | varchar(50) | YES  |     | NULL    |       |
| m_id      | int(11)     | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
3 rows in set (0.02 sec)


MariaDB [preksha]> create procedure savepoint_trans(in d_name
varchar(50),in lcn varchar(50),in addr varchar(50),in zip_code int(6),in
mngr_id int(6))
    -> begin
    -> declare lc_exist int DEFAULT 0;
    -> declare duplicate_dept int DEFAULT 0;
    -> START TRANSACTION;
    -> select count(*) into lc_exist from location where location = lcn ;
    -> if lc_exist = 0 then
    -> insert audit_log values (concat('Creating new location ',lcn));
    -> insert into location values (lcn,addr,zip_code);
    -> else
    -> update location set address = addr,zip = zip_code where location =
lcn;
    -> end if;
    -> SAVEPOINT savepoint_location_exists;
    -> BEGIN
    -> DECLARE DUPLICATE_KEY CONDITION FOR 1062;
    -> DECLARE CONTINUE HANDLER FOR DUPLICATE_KEY
    -> BEGIN
    -> SET duplicate_dept = 1;
    -> ROLLBACK TO SAVEPOINT savepoint_location_exists;
    -> END;
    -> insert into audit_log values (concat('creating new department
',d_name));
```

```
    -> insert into dept values(d_name,lcn,mngr_id);
    -> IF duplicate_dept = 1 then
    -> update dept set location = lcn,m_id = mngr_id WHERE dept_name =
d_name;
    -> END IF;
    -> END;
    -> COMMIT;
    -> end $$
Query OK, 0 rows affected (0.08 sec)



MariaDB [preksha]> select * from location $$
+-----------+---------+--------+
| location  | address | zip    |
+-----------+---------+--------+
| ahmedabad | Motera  | 380005 |
+-----------+---------+--------+
1 row in set (0.00 sec)



MariaDB [preksha]> select * from audit_log $$
+---------------------------------+
| audit_message                   |
+---------------------------------+
| Creating new location ahmedabad |
| creating new department Account |
+---------------------------------+
2 rows in set (0.00 sec)



MariaDB [preksha]> select * from dept $$
+-----------+-----------+------+
| dept_name | location  | m_id |
+-----------+-----------+------+
| Account   | ahmedabad |    1 |
+-----------+-----------+------+
1 row in set (0.00 sec)

**************************************************************************
**************************************************************************
```