Assignment No.: 8

Submission Date: 04/03/2024
Name: Veeransh
Reg No.: 221070063
Batch: SY COMPS

AIM:

Study of Functions and Procedures in SQL

TOOL: MariaDB

PROGRAMMING LANGUAGE: Structured Query language (SQL)

INDEX:

1. Functions and its syntax
2. Stored Procedures

THEORY:

FUNCTIONS IN SQL SERVER :

These are the database objects that contain a set of SQL statements to perform a specific task. A function accepts input parameters, performs actions, and then returns the result. We should note that functions always return either a single value or a table. The main purpose of functions is to replicate the common task easily. We can build functions one time and can use them in multiple locations based on our needs. SQL Server does not allow the use of the functions for inserting, deleting, or updating records in the database tables.

The following are the rules for creating SQL Server functions:

- A function must have a name, and the name cannot begin with a special character such as @, $, #, or other similar characters.
- SELECT statements are the only ones that operate with functions.
- We can use a function anywhere such as AVG, COUNT, SUM, MIN, DATE, and other functions with the SELECT query in SQL.
- Whenever a function is called, it compiles.
- Functions must return a value or result.
- Functions use only input parameters.
- We cannot use TRY and CATCH statements in functions.

User-Defined Functions

Functions that are created by the user in the system database or a user-defined database are known as user-defined functions. The UDF functions accept parameters, perform actions, and return the result. These functions help us to simplify our development by encapsulating complex business logic and making it available for reuse anywhere based on the needs. SQL Server categorizes the user-defined functions mainly into two types:

1. Scalar Functions
2. Table-Valued Functions

Scalar Functions

Scalar function in SQL Server always accepts parameters, either single or multiple and returns a single value. The scalar functions are useful in the simplification of our code. Suppose we might have a complex computation that appears in a number of queries. In such a case, we can build a scalar function that encapsulates the formula and uses it in each query instead of in each query.

STORED PROCEDURES IN SQL SERVER:

A stored procedure is a group of one or more pre-compiled SQL statements into a logical unit. It is stored as an object inside the database server. It is a subroutine or a subprogram in the common computing language that has been created and stored in the database. Each procedure in SQL Server always contains a name, parameter lists, and Transact-SQL statements. The SQL Database Server stores the stored procedures as named objects. We can invoke the procedures by using triggers, other procedures, and applications such as Java, Python, PHP, etc. It can support almost all relational database systems.

SQL Server builds an execution plan when the stored procedure is called the first time and stores them in the cache memory. The plan is reused by SQL Server in subsequent executions of the stored procedure, allowing it to run quickly and efficiently.

User-defined Stored Procedures

Database developers or database administrators build user-defined stored procedures. These procedures provide one or more SQL statements for selecting, updating, or removing data from database tables. A stored procedure specified by the user accepts input parameters and returns output parameters. DDL and DML commands are used together in a user-defined procedure.

We can further divide this procedure into two types:

○ T-SQL Stored Procedures: Transact-SQL procedures are one of the most popular types of SQL Server procedures. It takes parameters and returns them. These procedures handle INSERT, UPDATE, and DELETE statements with or without parameters and output row data.

- CLR Stored Procedures: The SQL Server procedures are a group of SQL commands, and the CLR indicates the common language runtime. CLR stored procedures are made up of the CLR and a stored procedure, which is written in a CLR-based language like VB.NET or C#. CLR procedures are .Net objects that run in the SQL Server database's memory.

COMMANDS:

```
Microsoft Windows [Version 10.0.19045.4046]
(c) Microsoft Corporation. All rights reserved.

C:\Users\siddhi>mysql -u root -p
Enter password: ****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 5
Server version: 11.2.2-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| db0                |
| db1                |
| db2                |
| db3                |
| db4                |
| db5                |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| vjti               |
+--------------------+
11 rows in set (0.001 sec)

MariaDB [(none)]> use db5;
Database changed
MariaDB [db5]> create function increment(no int)
    -> returns int
    -> return(no+1);
Query OK, 0 rows affected, 1 warning (0.006 sec)

MariaDB [db5]> select increment(5);
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near '(5)' at line 1
MariaDB [db5]> drop function increment;
Query OK, 0 rows affected (0.005 sec)

MariaDB [db5]> create function increment1(no int)
    -> returns int
    -> return(no+1);
Query OK, 0 rows affected (0.005 sec)

MariaDB [db5]> select increment1(5);
+---------------+
| increment1(5) |
+---------------+
|             6 |
+---------------+
1 row in set (0.001 sec)

MariaDB [db5]> create function avg1(n1 int, n2 int)
    -> returns int
    -> return(n1+n2)/2;
Query OK, 0 rows affected (0.006 sec)

MariaDB [db5]> select avg1(2,5);
+-----------+
| avg1(2,5) |
+-----------+
|         4 |
+-----------+
1 row in set (0.001 sec)
```

```
MariaDB [db5]> create function avg2(n1 int, n2 int)
    -> returns int
    -> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near '' at line 2
MariaDB [db5]> delimiter //
MariaDB [db5]> create function avg2(n1 int, n2 int)
    -> returns int
    -> begin
    -> declare a int;
    -> set a = (n1+n2)/2;
    -> return a;
    -> end;
    -> //
Query OK, 0 rows affected (0.008 sec)

MariaDB [db5]> select avg2(4,5)//
+-----------+
| avg2(4,5) |
+-----------+
|         5 |
+-----------+
1 row in set (0.001 sec)

MariaDB [db5]> delimiter ;
MariaDB [db5]> delimiter $$
MariaDB [db5]> create function calculate_income(salary int)
    -> returns int
    -> begin
    -> declare income int;
    -> set income = 0;
    -> label1: while(income<=3000) do set income = income + salary;
    -> end while label1;
    -> return income;
    -> end;
    -> $$
Query OK, 0 rows affected (0.006 sec)

MariaDB [db5]> select calculate_income(2000)$$
+------------------------+
| calculate_income(2000) |
+------------------------+
|                   4000 |
+------------------------+
1 row in set (0.001 sec)

MariaDB [db5]> create function PassID(id int)
    -> returns varchar(20)
    -> begin
    -> declare webid1 varchar(20);
    -> declare webid2 varchar(20);
    -> set webid1 = "www.Google.com";
    -> set webid2 = "www.gmail.com";
    -> if(id<5) then return webid1;
    -> else return webid2;
    -> end if;
    -> end;
    -> $$
Query OK, 0 rows affected (0.006 sec)

MariaDB [db5]> select PassID(10)$$
+---------------+
| PassID(10)    |
+---------------+
| www.gmail.com |
+---------------+
1 row in set (0.001 sec)

MariaDB [db5]> select PassID(4)$$
+----------------+
| PassID(4)      |
+----------------+
| www.Google.com |
+----------------+
1 row in set (0.000 sec)

MariaDB [db5]> create tabel customer$$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near 'tabel customer' at
line 1
MariaDB [db5]> create table customer$$
ERROR 1113 (42000): A table must have at least 1 column
MariaDB [db5]> create table customer(cid int, cname varchar(30), dop date, creditlimit
decimal(5,2)) $$
Query OK, 0 rows affected (0.011 sec)

MariaDB [db5]> insert into customer (1, "AAA", "2020-05-13", 60000.00);
    -> $$
```

```
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near '1, "AAA",
"2020-05-13", 60000.00)' at line 1
MariaDB [db5]> insert into customer (1, "AAA", "2020-05-13", 600.00)$$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near '1, "AAA",
"2020-05-13", 600.00)' at line 1
MariaDB [db5]> insert into customer values(1, "AAA", "2020-05-13", 600.00)$$
Query OK, 1 row affected (0.003 sec)

MariaDB [db5]> select * from customer;
    -> $$
+------+-------+------------+-------------+
| cid  | cname | dop        | creditlimit |
+------+-------+------------+-------------+
|    1 | AAA   | 2020-05-13 |      600.00 |
+------+-------+------------+-------------+
1 row in set (0.001 sec)

MariaDB [db5]> insert into customer values(2, "BBB", "2021-01-12", 70000.00)$$
ERROR 1264 (22003): Out of range value for column 'creditlimit' at row 1
MariaDB [db5]> insert into customer values(2, "BBB", "2021-01-12", 700.00)$$
Query OK, 1 row affected (0.001 sec)

MariaDB [db5]> insert into customer values(3, "CCC", "2020-08-15", 90.00)$$
Query OK, 1 row affected (0.001 sec)

MariaDB [db5]> select * from customer;
    -> $$
+------+-------+------------+-------------+
| cid  | cname | dop        | creditlimit |
+------+-------+------------+-------------+
|    1 | AAA   | 2020-05-13 |      600.00 |
|    2 | BBB   | 2021-01-12 |      700.00 |
|    3 | CCC   | 2020-08-15 |       90.00 |
+------+-------+------------+-------------+
3 rows in set (0.001 sec)

MariaDB [db5]> create function CustomerLevel(credit decimal(5,2))
    -> returns varchar(20)
    -> begin
    -> declare customerlevel varchar(20);
    -> if credit>500 then set customerlevel = "PLATINUM";
    -> elseif (credit<=500 and credit>100) set customerlevel="GOLD";
    -> elseif (credit=100) set customerlevel = "SILVER";
    -> else set customerlevel = "NONE";
    -> $$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near 'set
customerlevel="GOLD";
elseif (credit=100) set customerlevel = "SILVER";
e...' at line 6
MariaDB [db5]> create function CustomerLevel(credit decimal(5,2))
    -> returns varchar(20)
    -> declare customerlevel varchar(20);
    -> if credit>500 then set customerlevel = "PLATINUM";
    -> elseif (credit<=500 and credit>100) then set customerlevel="GOLD";
    -> elseif (credit=100) then set customerlevel = "SILVER";
    -> else set customerlevel = "NONE";
    -> end;
    -> $$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near 'declare
customerlevel varchar(20);
if credit>500 then set customerlevel = "PL...' at line 3
MariaDB [db5]>  create function CustomerLevel(credit decimal(5,2))
    ->     -> returns varchar(20)
    ->     -> declare customerlevel varchar(20);
    ->     -> if credit>500 then set customerlevel = "PLATINUM";
    ->     -> elseif (credit<=500 and credit>100) then set customerlevel="GOLD";
    ->     -> elseif (credit=100) then set customerlevel = "SILVER";
    ->     -> else set customerlevel = "NONE";
    -> end if
    -> end;
    -> $$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near '-> returns
varchar(20)
    -> declare customerlevel varchar(20);
    -> if cr...' at line 2
MariaDB [db5]> CREATE FUNCTION CustomerLevel(credit DECIMAL(5, 2))
    -> RETURNS VARCHAR(20)
    -> BEGIN
    ->     DECLARE customerlevel VARCHAR(20);
    ->
    ->     IF credit > 500 THEN
    ->         SET customerlevel = "PLATINUM";
    ->     ELSEIF credit <= 500 AND credit > 100 THEN
    ->         SET customerlevel = "GOLD";
    ->     ELSEIF credit = 100 THEN
    ->         SET customerlevel = "SILVER";
    ->     ELSE
    ->         SET customerlevel = "NONE";
    ->     END IF;
    ->
    ->     RETURN customerlevel;
    -> END;
    -> $$
Query OK, 0 rows affected (0.007 sec)

MariaDB [db5]> select CustomerLevel(600)$$
+--------------------+
| CustomerLevel(600) |
+--------------------+
| PLATINUM           |
+--------------------+
1 row in set (0.002 sec)

MariaDB [db5]> select CustomerLevel(400)$$
+--------------------+
| CustomerLevel(400) |
+--------------------+
| GOLD               |
+--------------------+
1 row in set (0.001 sec)

MariaDB [db5]> select CustomerLevel(100)$$
+--------------------+
| CustomerLevel(100) |
+--------------------+
| SILVER             |
+--------------------+
1 row in set (0.000 sec)

MariaDB [db5]> select CustomerLevel(90)$$
+-------------------+
| CustomerLevel(90) |
+-------------------+
| NONE              |
+-------------------+
1 row in set (0.000 sec)
```

```
MariaDB [db5]> --return(customerlevel) is also correct
MariaDB [db5]> select cname, CustomerLevel(creditlimit) as level from customer$$
+-------+----------+
| cname | level    |
+-------+----------+
| AAA   | PLATINUM |
| BBB   | PLATINUM |
| CCC   | NONE     |
+-------+----------+
3 rows in set (0.001 sec)

MariaDB [db5]> select cname, CustomerLevel(creditlimit) as level from customer order by
cname desc$$
+-------+----------+
| cname | level    |
+-------+----------+
| CCC   | NONE     |
| BBB   | PLATINUM |
| AAA   | PLATINUM |
+-------+----------+
3 rows in set (0.001 sec)

MariaDB [db5]> Bye
Ctrl-C -- exit!

C:\Users\siddhi>


Microsoft Windows [Version 10.0.19045.4046]
(c) Microsoft Corporation. All rights reserved.

C:\Users\siddhi>mysql -u root -p
Enter password: ****
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)

C:\Users\siddhi>mysql -u root -p
Enter password: ****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 11.2.2-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
MariaDB [(none)]> show databases
    -> ;
+--------------------+
| Database           |
+--------------------+
| db0                |
| db1                |
| db2                |
| db3                |
| db4                |
| db5                |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| vjti               |
+--------------------+
11 rows in set (0.017 sec)

MariaDB [(none)]> use db5;
Database changed
MariaDB [db5]> show tables;
+---------------+
| Tables_in_db5 |
+---------------+
| customer      |
+---------------+
1 row in set (0.001 sec)

MariaDB [db5]> desc customer;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| cid         | int(11)      | YES  |     | NULL    |       |
| cname       | varchar(30)  | YES  |     | NULL    |       |
| dop         | date         | YES  |     | NULL    |       |
| creditlimit | decimal(5,2) | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
4 rows in set (0.049 sec)

MariaDB [db5]> create procedure proc1(in var1 int, out var2 int)
    -> begin
    -> set var2 = var1 + 1;
```

```
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near '' at line 3
MariaDB [db5]> delimiter $$
MariaDB [db5]> create procedure proc1(in var1 int, out var2 int)
    -> begin
    -> set var2 = var1 + 1;
    -> end;
    -> $$
Query OK, 0 rows affected (0.012 sec)

MariaDB [db5]> call proc1(1)$$
ERROR 1318 (42000): Incorrect number of arguments for PROCEDURE db5.proc1; expected
2, got 1
MariaDB [db5]> call proc1(1,@a)$$
Query OK, 0 rows affected (0.001 sec)

MariaDB [db5]> select @a$$
+------+
| @a   |
+------+
|    2 |
+------+
1 row in set (0.000 sec)

MariaDB [db5]> create procedure proc2(out var2 char(50))
    -> begin
    -> set var2 = "Hello Siddhi"
    -> end;
    -> $$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near 'end' at line 4
MariaDB [db5]> create procedure proc2(out var2 char(50))
    -> begin
    -> set var2 = "Hello Siddhi";
    -> end;
    -> $$
Query OK, 0 rows affected (0.006 sec)

MariaDB [db5]> call proc2(@b)$$
Query OK, 0 rows affected (0.001 sec)

MariaDB [db5]> select @b $$
+--------------+
| @b           |
+--------------+
| Hello Siddhi |
+--------------+
1 row in set (0.000 sec)

MariaDB [db5]> create procedure proc3(inout var1 int)
    -> begin
    -> set var1 = var1 + 1;
    -> end;
    -> $$
Query OK, 0 rows affected (0.005 sec)

MariaDB [db5]> call proc3(@c) $$
Query OK, 0 rows affected (0.001 sec)

MariaDB [db5]> select @c $$
+------+
| @c   |
+------+
| NULL |
+------+
1 row in set (0.000 sec)

MariaDB [db5]> set @c = 6 $$
Query OK, 0 rows affected (0.000 sec)

MariaDB [db5]> select @c $$
+------+
| @c   |
+------+
|    6 |
+------+
1 row in set (0.000 sec)

MariaDB [db5]> call proc3(@c) $$
Query OK, 0 rows affected (0.000 sec)

MariaDB [db5]> select @c $$
+------+
| @c   |
+------+
|    7 |
+------+
1 row in set (0.000 sec)
```

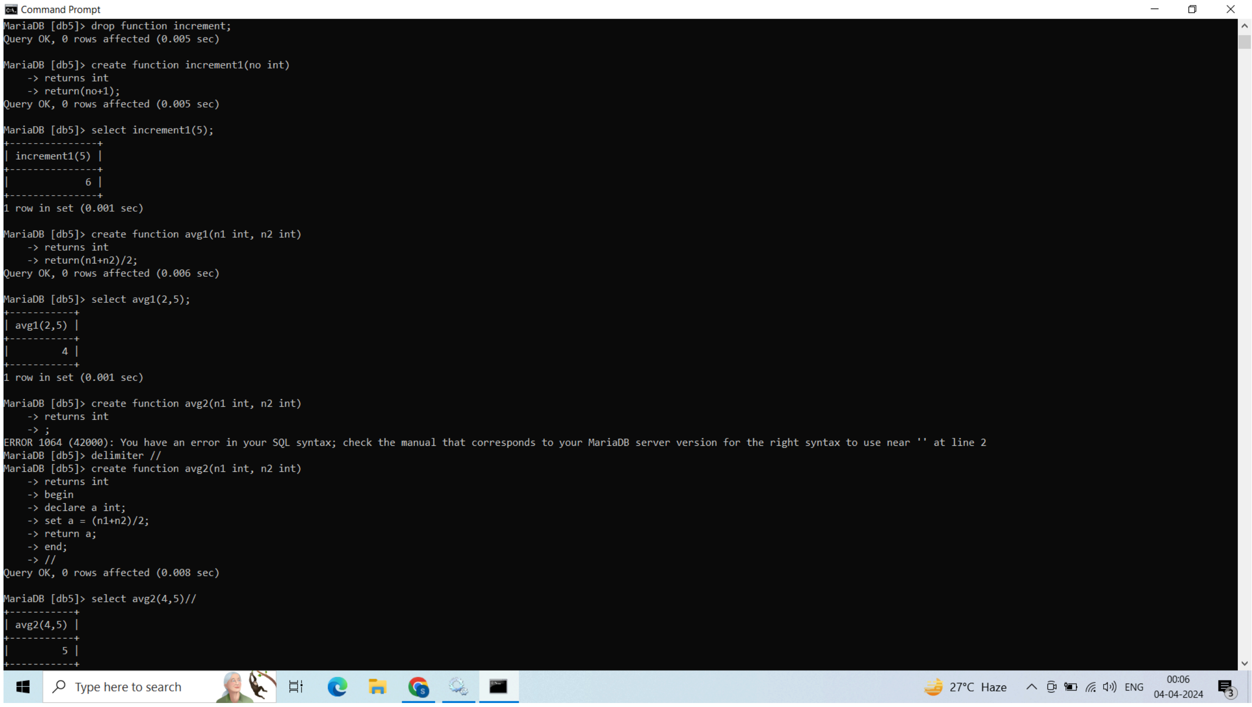
```
MariaDB [db5]> create procedure proc4()
    -> begin
    -> select * from customer;
    -> end;
    -> $$
Query OK, 0 rows affected (0.006 sec)

MariaDB [db5]> call proc4()$$
+------+-------+------------+-------------+
| cid  | cname | dop        | creditlimit |
+------+-------+------------+-------------+
|    1 | AAA   | 2020-05-13 |      600.00 |
|    2 | BBB   | 2021-01-12 |      700.00 |
|    3 | CCC   | 2020-08-15 |       90.00 |
+------+-------+------------+-------------+
3 rows in set (0.010 sec)

Query OK, 0 rows affected (0.027 sec)

MariaDB [db5]> create procedure age_limit(in age int out drive varchar(100))
    -> begin
    -> if(age>18) then set drive = "Driving Allowed";
    -> else set drive = "Not allowed";
    -> end if;
    -> end;
    -> $$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near 'out drive
varchar(100))
begin
if(age>18) then set drive = "Driving Allowed";
...' at line 1
MariaDB [db5]> create procedure age_limit(in age int, out drive varchar(100))
    -> begin
    -> if(age>=18) then set drive = "Driving Allowed";
    -> else set drive = "Not Allowed";
    -> end if;
    -> end;
    -> $$
Query OK, 0 rows affected (0.007 sec)

MariaDB [db5]> call age_limit (22, @d) $$
Query OK, 0 rows affected (0.002 sec)
```

```
MariaDB [db5]> select @d $$
+-----------------+
| @d              |
+-----------------+
| Driving Allowed |
+-----------------+
1 row in set (0.000 sec)

MariaDB [db5]> call age_limit(2, @d) $$
Query OK, 0 rows affected (0.000 sec)

MariaDB [db5]> select @d $$
+-------------+
| @d          |
+-------------+
| Not Allowed |
+-------------+
1 row in set (0.000 sec)

MariaDB [db5]> create procedure double(in N1 int, out N2 int)
    -> begin
    -> while(N1<5) do set N2 = N1 + 1;
    -> set N1 = N1 + 1;
    -> end while;
    -> end;
    -> $$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near 'double(in N1 int,
out N2 int)
begin
while(N1<5) do set N2 = N1 + 1;
set N1 = ...' at line 1
MariaDB [db5]> create procedure double1(in N1 int, out N2 int)
    -> begin
    -> while(N1<5) do set N2 = N1 + 1;
    -> set N1 = N1 + 1;
    -> end while;
    -> end;
    -> $$
Query OK, 0 rows affected (0.007 sec)

MariaDB [db5]> call double1(1, @e) $$
Query OK, 0 rows affected (0.001 sec)
```

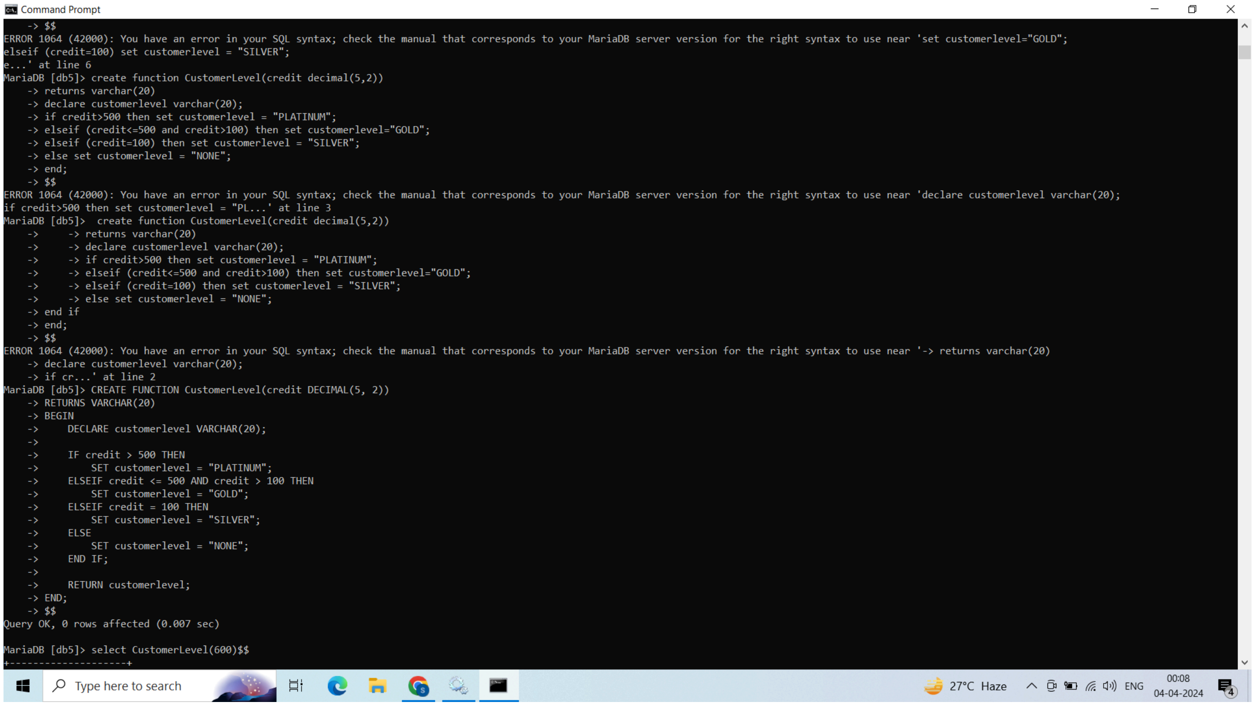
```
MariaDB [db5]> select @e $$
+------+
| @e   |
+------+
|    5 |
+------+
1 row in set (0.000 sec)

MariaDB [db5]> create procedure calc_income(in salary int, out income int)
    -> begin
    -> set income = 0;
    -> label1: while(income<3000)
    -> do set income = income + salary;
    -> end while label1;
    -> end;
    -> $$
Query OK, 0 rows affected (0.006 sec)

MariaDB [db5]> call calc_income(2000, @f) $$
Query OK, 0 rows affected (0.001 sec)

MariaDB [db5]> select @f;
    -> $$
+------+
| @f   |
+------+
| 4000 |
+------+
1 row in set (0.000 sec)

MariaDB [db5]> create procedure GetCustomerLevel(in cno int, out clevel varchar(30))
    -> begin
    -> declare credit dec(10,2) default = 0;
    -> select creditlimit into credit from customer where cid = cno;
    -> set clevel = clevel(credit)
    -> end;
    -> $$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near '= 0;
select creditlimit into credit from customer where cid = cno;
set clevel...' at line 3
MariaDB [db5]> create procedure GetCustomerLevel(in cno int, out clevel varchar(30))
    -> begin
    -> declare credit dec(10,2) default = 0;
    -> select creditlimit into credit from customer where cid = cno;
    -> set clevel = clevel(credit);
    -> end;
    -> $$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near '= 0;
select creditlimit into credit from customer where cid = cno;
set clevel...' at line 3
MariaDB [db5]> create procedure GetCustomerLevel(in cno int, out clevel varchar(30))
    -> begin
    -> declare credit dec(10,2) default 0;
    -> select creditlimit into credit from customer where cid = cno;
    -> set clevel = clevel(credit);
    -> end;
    -> $$
Query OK, 0 rows affected (0.008 sec)

MariaDB [db5]> call  GetCustomerLevel(1, @g)$$)
ERROR 1305 (42000): FUNCTION db5.clevel does not exist
    -> $$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near ')' at line 1
MariaDB [db5]> --how to use functions inside procedures:
MariaDB [db5]> --create procedure GetCustomerLevel(in cno int, out clevel varchar(30))
MariaDB [db5]> --begin
MariaDB [db5]> --declare credit dec(5,2) default 0;
MariaDB [db5]> --select creditlimit into credit from customer where cid = cno;
MariaDB [db5]> --set clevel = CustomerLevel(credit);
MariaDB [db5]> --end;
MariaDB [db5]> --$$
MariaDB [db5]> create procedure GetCustomerLevel(in cno int, out clevel varchar(30))
    ->     -> begin
    ->     -> declare credit dec(10,2) default 0;
    ->     -> select creditlimit into credit from customer where cid = cno;
    ->     -> set clevel = clevel(credit);
    ->     -> end;
    -> $$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near '-> begin
    -> declare credit dec(10,2) default 0;
    -> select creditlimit...' at line 2
MariaDB [db5]> create procedure GetCustomerLevel(in cno int, out clevel varchar(30))
    ->     -> begin
```

```
   ->     -> declare credit dec(10,2) default 0;
   ->     -> select creditlimit into credit from customer where cid = cno;
   ->     -> set clevel = clevel(credit);
   ->     end;
   -> $$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near '-> begin
   -> declare credit dec(10,2) default 0;
   -> select creditlimit...' at line 2
MariaDB [db5]> select * from products;
   -> ;
   -> $$
ERROR 1146 (42S02): Table 'db5.products' doesn't exist
MariaDB [db5]> create table products$$
ERROR 1113 (42000): A table must have at least 1 column
MariaDB [db5]> create table products(prod_id int, prod_name char(30), prod_cost dec(4,2),
prod_price dec(5,2));
   -> $$
Query OK, 0 rows affected (0.028 sec)

MariaDB [db5]> insert into products((1, "ParleG", 5.95, 8.35))$$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near '1, "ParleG", 5.95,
8.35))' at line 1
MariaDB [db5]> insert into products(1, "ParleG", 5.95, 8.35)$$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near '1, "ParleG", 5.95,
8.35)' at line 1
MariaDB [db5]> insert into products values(1, "ParleG", 5.95, 8.35)$$
Query OK, 1 row affected (0.006 sec)

MariaDB [db5]> insert into products values( (2, "MangoBite", 0.95, 1.35), (3,"Sugar", 99.95, 140)
)$$
ERROR 1241 (21000): Operand should contain 1 column(s)
MariaDB [db5]> insert into products values(2, "MangoBite", 0.95, 1.35), (3,"Sugar", 99.95, 140)
$$
Query OK, 2 rows affected (0.002 sec)
Records: 2  Duplicates: 0  Warnings: 0

MariaDB [db5]> select * from products$$
+---------+-----------+-----------+------------+
| prod_id | prod_name | prod_cost | prod_price |
+---------+-----------+-----------+------------+
|       1 | ParleG    |      5.95 |       8.35 |
|       2 | MangoBite |      0.95 |       1.35 |
|       3 | Sugar     |     99.95 |     140.00 |
+---------+-----------+-----------+------------+
3 rows in set (0.001 sec)

MariaDB [db5]> create function calcProfit( cost dec(4,2), price dec(5,2))
    -> returns decimal (9,2)
    -> begin
    -> declare profit decimal(9,2);
    -> set profir = price - cost;
    -> return profit;
    -> end;
    -> $$
ERROR 1193 (HY000): Unknown system variable 'profir'
MariaDB [db5]> create function calcProfit( cost dec(4,2), price dec(5,2))
    -> returns decimal (9,2)
    -> declare profit decimal(9,2);
    -> set profit = price - cost;
    -> returb profit;
    -> end;
    -> $$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near 'declare profit
decimal(9,2);
set profit = price - cost;
returb profit;
end' at line 3
MariaDB [db5]> create function calcProfit( cost dec(4,2), price dec(5,2))
    -> returns decimal (9,2)
    -> declare profit decimal(9,2);
    -> set profit = price - cost;
    -> return profit;
    -> end;
    -> $$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near 'declare profit
decimal(9,2);
set profit = price - cost;
return profit;
end' at line 3
MariaDB [db5]> create function calcProfit( cost dec(4,2), price dec(5,2))
    -> returns decimal (9,2)
    -> begin
    -> declare profit decimal(9,2);
    -> set profit = price - cost;
    -> return profit;
    -> end;
    -> $$
Query OK, 0 rows affected (0.005 sec)

MariaDB [db5]> select *, calcProfit(prod_cost, prod_price) as PROFIT from products $$
+---------+-----------+-----------+------------+--------+
| prod_id | prod_name | prod_cost | prod_price | PROFIT |
+---------+-----------+-----------+------------+--------+
|       1 | ParleG    |      5.95 |       8.35 |   2.40 |
|       2 | MangoBite |      0.95 |       1.35 |   0.40 |
|       3 | Sugar     |     99.95 |     140.00 |  40.05 |
+---------+-----------+-----------+------------+--------+
3 rows in set (0.004 sec)

MariaDB [db5]> create procedure calcProfit(in prod_cost dec(4,2), in prod_price dec(5,2), out
profit dec(9,2))
    -> begin
    -> declare profit = prod_price - prod_cost;
    -> end;
    -> $$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near '= prod_price -
prod_cost;
end' at line 3
MariaDB [db5]> create procedure calcProfit(in prod_cost dec(4,2), in prod_price dec(5,2), out
profit dec(9,2))
    -> begin
    -> set profit = prod_price - prod_cost;
    -> end;
    -> $$
Query OK, 0 rows affected (0.006 sec)

MariaDB [db5]> call calcProfit(c, p, @g)$$
ERROR 1054 (42S22): Unknown column 'c' in 'field list'
MariaDB [db5]> call calcProfit( prod_cost,  prod_price, @g)$$
ERROR 1054 (42S22): Unknown column 'prod_cost' in 'field list'
MariaDB [db5]> select * from customer;
    -> call calcProfit( prod_cost,  prod_price, @g) as PROFIT from products$$
+------+-------+------------+-------------+
| cid  | cname | dop        | creditlimit |
+------+-------+------------+-------------+
|    1 | AAA   | 2020-05-13 |      600.00 |
|    2 | BBB   | 2021-01-12 |      700.00 |
|    3 | CCC   | 2020-08-15 |       90.00 |
+------+-------+------------+-------------+
3 rows in set (0.001 sec)

ERROR 1054 (42S22): Unknown column 'prod_cost' in 'field list'
MariaDB [db5]> call calcProfit(5.95, 8.35, @g)$$
Query OK, 0 rows affected (0.001 sec)

MariaDB [db5]> select @g$$
+------+
| @g   |
+------+
| 2.40 |
+------+
1 row in set (0.000 sec)

MariaDB [db5]>
```

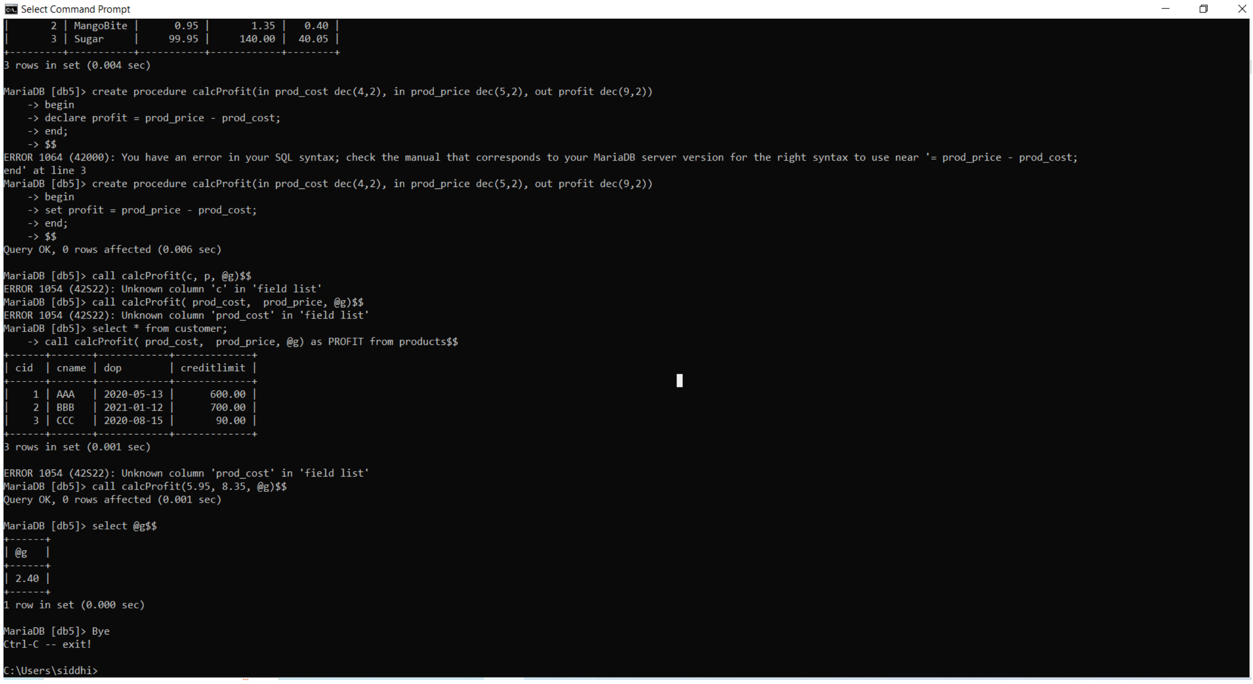
SS OF THE OUTPUT:



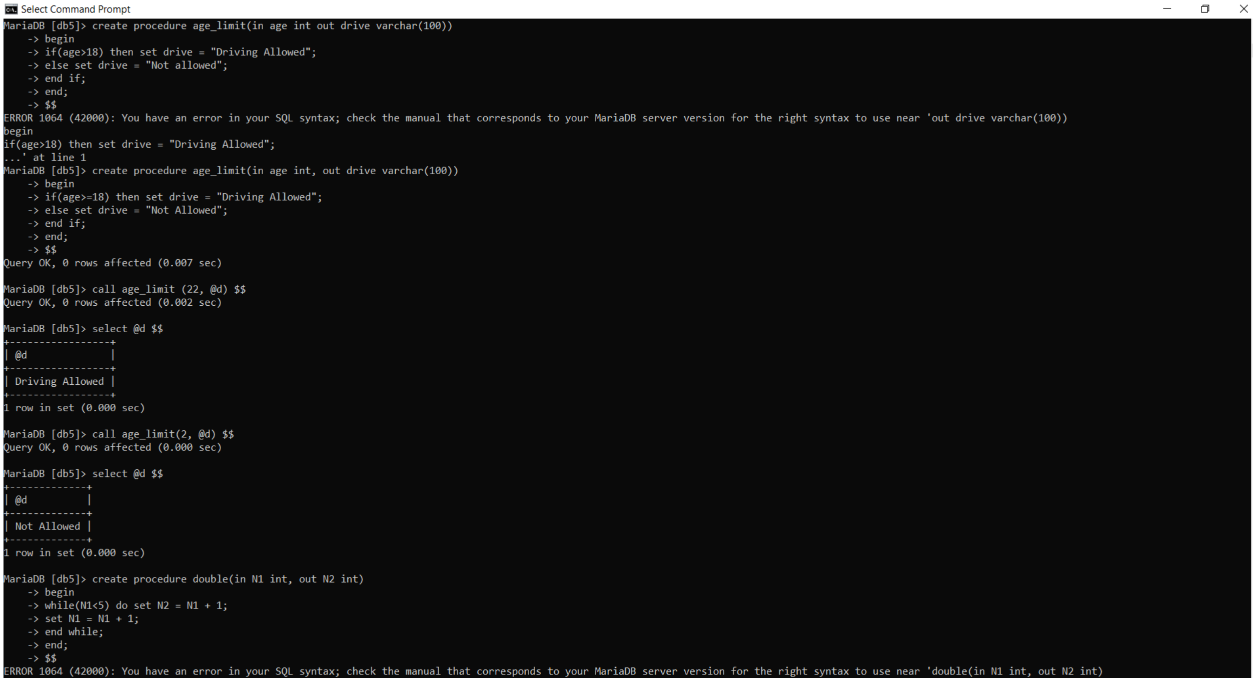
```
MariaDB [db5]> drop function increment;
Query OK, 0 rows affected (0.005 sec)

MariaDB [db5]> create function increment1(no int)
    -> returns int
    -> return(no+1);
Query OK, 0 rows affected (0.005 sec)

MariaDB [db5]> select increment1(5);
+---------------+
| increment1(5) |
+---------------+
|             6 |
+---------------+
1 row in set (0.001 sec)

MariaDB [db5]> create function avg1(n1 int, n2 int)
    -> returns int
    -> return(n1+n2)/2;
Query OK, 0 rows affected (0.006 sec)

MariaDB [db5]> select avg1(2,5);
+-----------+
| avg1(2,5) |
+-----------+
|         4 |
+-----------+
1 row in set (0.001 sec)

MariaDB [db5]> create function avg2(n1 int, n2 int)
    -> returns int
    -> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '' at line 2
MariaDB [db5]> delimiter //
MariaDB [db5]> create function avg2(n1 int, n2 int)
    -> returns int
    -> begin
    -> declare a int;
    -> set a = (n1+n2)/2;
    -> return a;
    -> end;
    -> //
Query OK, 0 rows affected (0.008 sec)

MariaDB [db5]> select avg2(4,5)//
+-----------+
| avg2(4,5) |
+-----------+
|         5 |
+-----------+
```



```
    -> $$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'set customerlevel="GOLD";
elseif (credit=100) set customerlevel = "SILVER";
e...' at line 6
MariaDB [db5]> create function CustomerLevel(credit decimal(5,2))
    -> returns varchar(20)
    -> declare customerlevel varchar(20);
    -> if credit>500 then set customerlevel = "PLATINUM";
    -> elseif (credit<=500 and credit>100) then set customerlevel="GOLD";
    -> elseif (credit=100) then set customerlevel = "SILVER";
    -> else set customerlevel = "NONE";
    -> end;
    -> $$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'declare customerlevel varchar(20);
if credit>500 then set customerlevel = "PL...' at line 3
MariaDB [db5]>  create function CustomerLevel(credit decimal(5,2))
    ->     -> returns varchar(20)
    ->     -> declare customerlevel varchar(20);
    ->     -> if credit>500 then set customerlevel = "PLATINUM";
    ->     -> elseif (credit<=500 and credit>100) then set customerlevel="GOLD";
    ->     -> elseif (credit=100) then set customerlevel = "SILVER";
    ->     -> else set customerlevel = "NONE";
    -> end if
    -> end;
    -> $$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '-> returns varchar(20)
    -> declare customerlevel varchar(20);
    -> if cr...' at line 2
MariaDB [db5]> CREATE FUNCTION CustomerLevel(credit DECIMAL(5, 2))
    -> RETURNS VARCHAR(20)
    -> BEGIN
    ->     DECLARE customerlevel VARCHAR(20);
    ->
    ->     IF credit > 500 THEN
    ->         SET customerlevel = "PLATINUM";
    ->     ELSEIF credit <= 500 AND credit > 100 THEN
    ->         SET customerlevel = "GOLD";
    ->     ELSEIF credit = 100 THEN
    ->         SET customerlevel = "SILVER";
    ->     ELSE
    ->         SET customerlevel = "NONE";
    ->     END IF;
    ->
    ->     RETURN customerlevel;
    -> END;
    -> $$
Query OK, 0 rows affected (0.007 sec)

MariaDB [db5]> select CustomerLevel(600)$$
+--------------------+
```



```
|       2 | MangoBite |      0.95 |       1.35 |   0.40 |
|       3 | Sugar     |     99.95 |     140.00 |  40.05 |
+---------+-----------+-----------+------------+--------+
3 rows in set (0.004 sec)

MariaDB [db5]> create procedure calcProfit(in prod_cost dec(4,2), in prod_price dec(5,2), out profit dec(9,2))
    -> begin
    -> declare profit = prod_price - prod_cost;
    -> end;
    -> $$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '= prod_price - prod_cost;
end' at line 3
MariaDB [db5]> create procedure calcProfit(in prod_cost dec(4,2), in prod_price dec(5,2), out profit dec(9,2))
    -> begin
    -> set profit = prod_price - prod_cost;
    -> end;
    -> $$
Query OK, 0 rows affected (0.006 sec)

MariaDB [db5]> call calcProfit(c, p, @g)$$
ERROR 1054 (42S22): Unknown column 'c' in 'field list'
MariaDB [db5]> call calcProfit( prod_cost,  prod_price, @g)$$
ERROR 1054 (42S22): Unknown column 'prod_cost' in 'field list'
MariaDB [db5]> select * from customer;
    -> call calcProfit( prod_cost,  prod_price, @g) as PROFIT from products$$
+------+-------+------------+-------------+
| cid  | cname | dop        | creditlimit |
+------+-------+------------+-------------+
|    1 | AAA   | 2020-05-13 |      600.00 |
|    2 | BBB   | 2021-01-12 |      700.00 |
|    3 | CCC   | 2020-08-15 |       90.00 |
+------+-------+------------+-------------+
3 rows in set (0.001 sec)

ERROR 1054 (42S22): Unknown column 'prod_cost' in 'field list'
MariaDB [db5]> call calcProfit(5.95, 8.35, @g)$$
Query OK, 0 rows affected (0.001 sec)

MariaDB [db5]> select @g$$
+------+
| @g   |
+------+
| 2.40 |
+------+
1 row in set (0.000 sec)

MariaDB [db5]> Bye
Ctrl-C -- exit!

C:\Users\siddhi>
```



```
MariaDB [db5]> create procedure age_limit(in age int out drive varchar(100))
    -> begin
    -> if(age>18) then set drive = "Driving Allowed";
    -> else set drive = "Not allowed";
    -> end if;
    -> end;
    -> $$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'out drive varchar(100))
begin
if(age>18) then set drive = "Driving Allowed";
...' at line 1
MariaDB [db5]> create procedure age_limit(in age int, out drive varchar(100))
    -> begin
    -> if(age>=18) then set drive = "Driving Allowed";
    -> else set drive = "Not Allowed";
    -> end if;
    -> end;
    -> $$
Query OK, 0 rows affected (0.007 sec)

MariaDB [db5]> call age_limit (22, @d) $$
Query OK, 0 rows affected (0.002 sec)

MariaDB [db5]> select @d $$
+-----------------+
| @d              |
+-----------------+
| Driving Allowed |
+-----------------+
1 row in set (0.000 sec)

MariaDB [db5]> call age_limit(2, @d) $$
Query OK, 0 rows affected (0.000 sec)

MariaDB [db5]> select @d $$
+-------------+
| @d          |
+-------------+
| Not Allowed |
+-------------+
1 row in set (0.000 sec)

MariaDB [db5]> create procedure double(in N1 int, out N2 int)
    -> begin
    -> while(N1<5) do set N2 = N1 + 1;
    -> set N1 = N1 + 1;
    -> end while;
    -> end;
    -> $$
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'double(in N1 int, out N2 int)
```

CONCLUSION:

Thus, we learnt about functions and stored procedures in sql, their different types and how to use them.