BHARATIYA VIDYA BHAVAN'S

SARDAR PATEL INSTITUTE OF TECHNOLOGY

Academic Year: 2023– 2024

Class: F.Y.MCA **Course Code: MC502** **Semester: I**

Course: Database Management System

Name of the Student: Abhijit Namdeo Shirke

UCID: 2023510054

Class: FY MCA

Signature of the Student: Abhijit

Class: F.Y.MCA **Course Code: MC502** **Semester: I**

Course: Database Management System

Title: Demonstrate functions, procedures.

Tools Required: MySQL Workbench

Prior Concept: Building blocks of PL/SQL, Conditional statement and Control structures.

New Concept: functions and procedures

Concept:

Function Syntax:

CREATE [OR REPLACE] FUNCTION function_name
[(parameter_name [IN | OUT | IN OUT] type [, ...])] RETURN
return_datatype
{IS | AS} BEGIN
< function_body >
END [function_name];

Procedure Syntax:

CREATE [OR REPLACE] PROCEDURE procedure_name
[(parameter_name [IN | OUT | IN OUT] type [, ...])]
{IS | AS} BEGIN
< procedure_body >
END procedure_name;

Lab Exercise:

Demonstrate all these syntax with the help on the problem statement given by instructor.

BHARATIYA VIDYA BHAVAN'S

SARDAR PATEL INSTITUTE OF TECHNOLOGY

Academic Year: 2023– 2024

Class: F.Y.MCA **Course Code: MC502** **Semester: I**

Course: Database Management System

Lab Assignment on Procedure

1. Use classicmodels . Create a procedure Get_Orders_Status which should accept the status value from user and show the number of orders for each year for that status. Table- **Orders**

CODE:

PROCEDURE

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `Get_Orders_Status`(IN var1 varchar(20))
BEGIN
SELECT YEAR(orderdate) as  Year,  COUNT(status) as Total_order FROM orders
GROUP BY status,Year
HAVING status=var1;
END
```

Calling PROCEDURE:

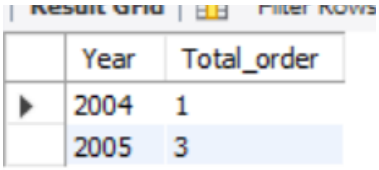
```
use classicmodels;
call Get_Orders_Status('Shipped');
```

	Year	Total_order
▶	2003	108
	2004	145
	2005	50

```
call Get_Orders_Status('Cancelled');
```

	Year	Total_order
▶	2003	2
	2004	4

```
call Get_Orders_Status('on Hold');
```



	Year	Total_order
▶	2004	1
	2005	3

Lab Assignment on Function

1. Create a function to find the cube of a number.

CODE:

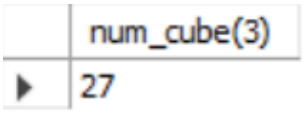
FUNCTION:

```
CREATE DEFINER=`root`@`localhost` FUNCTION `num_cube`(num int)
RETURNS int
    DETERMINISTIC
BEGIN
DECLARE ncube int;
SET ncube = num*num*num;
RETURN ncube;
END
```

CALLING FUNCTION:

```
select num_cube(3);
```

OUTPUT:



	num_cube(3)
▶	27

2. Use Classicmodels. Create a function which will return city of the given officeCode.

CODE:

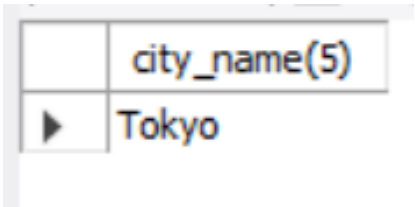
FUNCTION:

```
CREATE DEFINER=`root`@`localhost` FUNCTION `city_name`(of_code INT)
RETURNS varchar(20) CHARSET latin1
DETERMINISTIC
BEGIN
DECLARE city_name VARCHAR(20);
SET city_name = (SELECT city from offices WHERE officeCode = of_code);
RETURN city_name;
END
```

CALLING FUNCTION:

```
select city_name(5);
```

OUTPUT:



	city_name(5)
▶	Tokyo

3. Use Classicmodels. Create a function to show the highest MSRP for each productline using window functions.

CODE:

FUNCTION:

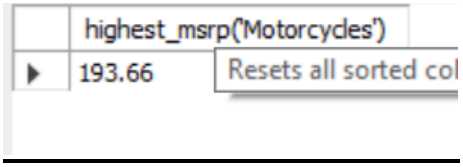
```
CREATE DEFINER=`root`@`localhost` FUNCTION `highest_msrp`(p_name
VARCHAR(20))
RETURNS decimal(10,2)
DETERMINISTIC
BEGIN
DECLARE pname decimal(10,2);
SET pname = (SELECT DISTINCT max(MSRP) OVER( PARTITION BY
productline) AS highest_MSRP FROM products WHERE productline = p_name);
RETURN pname;
END
```

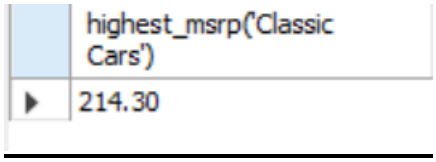
CALLING FUNCTION:

```
select highest_msrp('Motorcycles');

select highest_msrp('Classic Cars');
```

OUTPUT:





4. Use Classicmodels. Create a function to show the customername who has used the highest CreditLimit.

CODE:

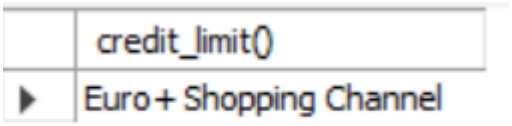
FUNCTION:

```
CREATE DEFINER=`root`@`localhost` FUNCTION `credit_limit`() RETURNS
varchar(25) CHARSET latin1
    DETERMINISTIC
BEGIN
DECLARE cname VARCHAR(25);
SET cname = (SELECT customerName from customers
                        WHERE creditLimit = (SELECT max(creditLimit)
                                                from customers));
RETURN cname;
END
```

CALLING FUNCTION:

```
select credit_limit();
```

OUTPUT:



	credit_limit()
▶	Euro + Shopping Channel

Observation:

1. I have learned how to create a **procedures** in MYSQL and how call it.
   We use ' **call( ) '** for calling the procedures.
   a. There are 3 varient of procedures
      I. A simple procedures.
      II. Procedures with IN parameters.
      III. Procedures with OUT parameters.

2. I have learned how to create a **function** in MYSQL and how call it.
   We use **'select function_name()'** or **' select function_name(parameter)'** for calling the function.
   Before **'BEGIN'** keyword use **DETERMINICSTIC** or **NOT DETERMINCSTIC** keyword. If we don't specify DETERMINISTIC or NOT DETERMINISTIC, by default. MySQL uses the NOT DETERMINISTIC option.

1. **DETERMINICSTIC:** A deterministic function in MySQL always returns the same result for the same input parameters.
2. **NOT DETERMINCSTIC:** A non-deterministic function returns different results for the same input parameters.

The difference between **Procedures** and **Function** is Procedures doesn't return a value and function returns a value.