**Exercise 1**

SHOW PROCEDURE STATUS WHERE Db = 'programm';

1. **Write a program that computes the perimeter and the area of a rectangle. Define**

**your own values for the length and width. (Assuming that *L* and *W* are the length**

**and width of the rectangle, Perimeter = *2\*(L+W)* and Area = *L\*W*.**

**Delimiter //**

**mysql> create procedure abc2() begin declare L int; declare W int; declare P int; set L = 20; set W = 25; set P = 2\*(L+W); insert into tempp values(P, 'Perimeter'); end;**

**Delimiter ;**

**Delimiter //**

**create procedure area() begin declare L int; declare W int; declare A int; set L = 20; set W = 25; set A = L\*W; insert into tempp values(A, 'Area'); end;**

**-> //**

**Delimiter ;**

**2. Write a program that declares an integer variable called *num*, assigns a value to it,**

**and computes and inserts into the tempp table the value of the variable itself, its**

**square, and its cube.**

**Delimiter //**

**create procedure sq() begin declare x int default 7; declare y int; set y = x\*x; insert into tempp values (y,'Squre'); end;//**

**Delimiter ;**

**Delimiter //**

**create procedure cu() begin declare X int default 7; declare C int ; set C = X\*X\*X; insert into tempp values(C, 'cube'); end; //**

**Delimiter ;**

**3. Convert a temperature in Fahrenheit (F) to its equivalent in Celsius (C) and vice**

**versa. The required formulae are:- *C*= (*F*-32)\*5/9**

***F*= 9/5\**C* + 32**

**Delimiter //**

**mysql> create procedure cf() begin declare c int; declare f int;**

**-> declare ctof int; declare ftoc int;**

**-> set c=50; set f=60;**

**-> set ctof=(9/5)\*c+32;**

**-> set ftoc=(f-32)\*5/9;**

**-> insert into temp values(ctof,'in fahr');**

**-> insert into temp values(ftoc,'in cels');**

**-> end; //**

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

**mysql> call cf()//**

**Query OK, 1 row affected (0.01 sec)**

**mysql> select \* from temp;**

**-> //**

**Delimiter ;**

**4. Convert a number of inches into yards, feet, and inches. For example, 124 inches**

**equals 3 yards, 1 foot, and 4 inches.**

**Delimiter //**

**create procedure con() begin declare I int; declare Y int; declare F int; set Y = 124/36; set F = mod(124,36)/12; set I = mod(mod(124,36),12); insert into tempp values (Y,'Yard'); insert into tempp values (F,'foot'); insert into tempp values (I,'Inches'); end;//**

**Delimiter ;**

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

**mysql> call con()//**

**Query OK, 1 row affected (0.01 sec)**

**mysql> select \* from tempp //**

**5. Write a program that enables a user to input an integer. The program should then**

**state whether the integer is evenly divisible by 5.**

**Delimiter //**

**create procedure even(x int ) begin if mod(x,5) = 0 then insert into tempp values(x,'even'); end if; end //**

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

**mysql> call even(20) //**

**Query OK, 1 row affected (0.01 sec)**

**mysql> select \* from tempp //**

**Delimiter ;**

**Delimiter //**

**create procedure even() begin declare x int default 10; if mod(x,5) = 0 then insert into tempp values(x,'even'); end if; end //**

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

**mysql> call even() //**

**Query OK, 1 row affected (0.03 sec)**

**mysql> select \* from tempp //**

**Delimiter ;**

**6. Your block should read in two real numbers and tell whether the product of the two**

**numbers is equal to or greater than 100.**

**Delimiter //**

**create procedure gt(x decimal(10,2),y decimal(10,2)) begin declare pro decimal (10,2); set pro = x\*y; if pro >= 100 then insert into tempp values(pro,'grater'); else insert into tempp values(pro,'less'); end if ; end //**

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

**mysql> call gt(100,50) //**

**Query OK, 1 row affected (0.01 sec)**

**mysql> select \* from tempp //**

**Delimiter ;**

**Delimiter //**

**Delimiter ;**

**MySQL**

**Exercise 2**

**1. Select from any table a number and determine whether it is within a given range (for**

**example, between 1 and 10).**

**Delimiter //**

**create procedure btw(y int) begin declare x int ;select deptno into x from emp where empno = y ; if (x >1 and x<10) then insert into tempp values(x,'range'); end if; end //**

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

**mysql> call btw(5)//**

**Query OK, 1 row affected (0.00 sec)**

**mysql> select \* from tempp//**

**Delimiter ;**

**2. Select from any table three positive integers representing the sides of a triangle, and**

**determine whether they form a valid triangle. Hint: In a triangle, the sum of any two**

**sides must always be greater than the third side.**

**Delimiter //**

**create procedure grt(a int,b int,c int) begin declare x int; declare y int; declare z int; select deptno into x from emp where empno =a ; select deptno into y from emp where empno =b ; select deptno into z from emp where empno =c ; if (a+b>c and b+c>a and c+a>b) then insert into tempp values(x+y+z,'valid'); end if; end //**

**Query OK, 0 rows affected (0.26 sec)**

**mysql> call grt(5,4,3)//**

**Query OK, 1 row affected (0.04 sec)**

**mysql> select \* from tempp//**

**Delimiter ;**

**3. Check if a given a year is a leap year. The condition is:- year should be (divisible by 4**

**and not divisible by 100) or (divisible by 4 and divisible by 400.). The year should be**

**Selected from some table.**

**Delimiter //**

**create procedure leap(a int) begin if (mod(a,4)=0 and mod(a,100) <> 0) or (mod(a,4)=0 and mod(a,400)=0) then insert into tempp values(a,'leap year'); end if; end //**

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

**mysql> call leap(2004)//**

**Query OK, 1 row affected (0.01 sec)**

**mysql> select \* from tempp//**

**Delimiter ;**

**4. Write a program that Selects from any table two character strings. Your program should**

**then determine if one character string exists inside another character string.**

**Delimiter //**

**Delimiter ;**

**MySQL**

**Exercise 3**

**1. Write a program containing a loop that iterates from 1 to 1000 using a variable *I*,**

**which is incremented each time around the loop. The program should output the**

**value of *I* every hundred iterations (i.e., the output should be 100, 200, etc.).**

**Delimiter //**

**Delimiter ;**

**2. Write a program that examines all the numbers from 1 to 999, displaying all those**

**for which the sum of the cubes of the digits equal the number itself.**

**Delimiter //**

**Delimiter ;**

**3. Write a program that Selects from any table a minimum and maximum value for a**

**radius, along with an increment factor, and generates a series of radii by repeatedly**

**adding the increment to the minimum until the maximum is reached. For each value**

**of the radius, compute and display the circumference, area, and volume of the**

**sphere. (Be sure to include both the maximum and the minimum values.).**

**Delimiter //**

**Delimiter ;**

**4. A *palindrome* is a word that is spelled the same forward and backward, such as**

***level*, *radar*, etc. Write a program to Selects from any table a five letter word and**

**determine whether it is a palindrome.**

**Delimiter //**

**Delimiter ;**

**5. Modify the above program to Select from any table a variable length word. This**

**requires determining how many characters are read in.**

**Delimiter //**

**Delimiter ;**

**MySQL**

**Exercise 4**

**1. The CUSTOMER table of a state electricity board consists of the following fields:-**

**Meter Number Varchar 4**

**Meter Type Char 1**

**Previous Reading Int 5**

**Current Reading Int 5**

**Customer Type Char 1**

**Last Bill payment Char 1 (values could be ‘Y’ or ‘N’)**

**There are two types of meters viz. 3- phase or 1-phase coded as ‘T’ or ‘S’**

**respectively. There are 4 types of customers viz. Agricultural Industrial,**

**Commercial and Residential with codes ‘A’ , ‘I’, ‘C’ and ‘R’ respectively.**

**Formulae:-**

**Units used = Current Reading – Previous Reading Rate**

**=Rs.1/ 1.25/ 1.50/ 1.30 for A/I/C/R respectively.**

**Amount = rate\*units used**

**Surcharge = 5% for single phase**

**10% for 3 phase**

**Excise = 30% of (amount +Surcharge)**

**Net = Amount +Surcharge + Excise**

**Write a block to calculate the bill for each customer. The program should insert the**

**Meter no., Units used, Rate, Amount, Surcharge, Excise duty and Net for each**

**customer into some other suitable table. Also, at the end, it should insert the total**

**Amount, Surcharge, Excise and Net into some other table.**

**MySQL**

**Exercise 5**

**1. Write a stored function to take three parameters, the sides of a triangle. The sides of**

**the triangle should be accepted from the user. The function should return a Boolean**

**value:- *true* if the triangle is valid, *false* otherwise. A triangle is valid if the length**

**of each side is less than the sum of the lengths of the other two sides. Check if the**

**dimensions entered can form a valid triangle.**

**Delimiter //**

**Delimiter ;**

**2. Write a function that generates a random number between 1 and 10. Use any logic**

**of your choice to achieve this.**

**Delimiter //**

**Delimiter ;**

**3. Create a function that accepts a string of *n* characters and exchanges the first**

**character with the last, the second with the next – to – last, and so forth until *n***

**exchanges have been made. What will the final string look like? Write the function**

**to verify your conclusion. *Sameer Dehadrai***

**Delimiter //**

**Delimiter ;**

**MySQL**

**Exercise 6**

**1. Write a stored procedure by the name of Comp\_intr to calculate the amount of**

**interest on a bank account that compounds interest yearly. The formula is:- I**

**= p (1+ r) y – p**

**where:-**

***I* is the total interest earned.**

***p* is the principal.**

***r* is the rate of interest as a decimal less than 1, and**

***y* is the number of years the money is earning interest.**

**Your stored procedure should accept the values of *p*, *r* and *y* as parameters and insert**

**the Interest and Total amount into tempp table.**

**Delimiter //**

**Delimiter ;**

**2. Create a stored function by the name of Age\_calc. Your stored function should**

**accept the date of birth of a person as a parameter. The stored function should**

**calculate the age of the person in years. The stored function should return the age**

**in years.**

**Delimiter //**

**Delimiter ;**

**MySQL**

**Exercise 7**

**Create the following 3 tables and insert sample data as shown:-**

**Ord\_mst**

**Ord\_no Cust\_cd Status**

**1 C1 p**

**Ord\_dtl**

**Ord\_no Prod\_cd Qty**

**1 P1 100**

**1 P2 200**

**Prod\_mst**

**Prod\_cd Prod\_name Qty\_in\_stock Booked\_qty**

**P1 Floppies 10000 1000**

**P2 Printers 5000 600**

**P3 Modems 3000 200**

**1. Write a Before Insert trigger on Ord\_dtl. Anytime a row is inserted in Ord\_dtl, the**

**Booked\_qty in Prod\_mst should be increased accordingly.**

**Delimiter //**

**Delimiter ;**

1. **Write a Before Delete trigger on Ord\_dtl. Anytime a row is deleted from Ord\_dtl, the Booked\_qty in Prod\_mst should be decreased accordingly.**

**Delimiter //**

**Delimiter ;**

**3. Write a Before Update of Prod\_cd, Qty trigger on Ord\_dtl. Anytime the Prod\_cd or**

**Qty is updated, the Booked\_qty in Prod\_mst should be increased/decreased**

**accordingly.**

**Delimiter //**

**Delimiter ;**

**4. Write a Before Update of Status trigger on Ord\_mst. If the Status is updated from P**

**(Pending) to D (Delivered), the Booked\_qty and Qty\_in\_stock from Prod\_mst should**

**be decreased accordingly. If the Status is updated from P (Pending) to C (Cancelled),**

**the details of the order should be deleted from Ord\_dtl and corresponding Booked\_qty**

**from Prod\_mst should be decreased accordingly. (The Before delete trigger on Ord\_dtl**

**would automatically decrease the Booked\_qty from Prod\_mst).**

**Delimiter //**

**Delimiter ;**