Practical – 9 Cursor Handling

1. Create procedure using cursor which calculate bonus 10% of all employee which have salary greater than 2000.

-> -- Create a stored procedure

-- Create a stored procedure

```
DELIMITER //

CREATE PROCEDURE CalculateBonus()
BEGIN
   DECLARE done INT DEFAULT 0;
   DECLARE emp_name VARCHAR(50);
   DECLARE emp_salary DECIMAL(10, 2);
   DECLARE employee_cursor CURSOR FOR
      SELECT Name, Salary
      FROM employee
WHERE salary > 2000;


   DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

   OPEN employee_cursor;

   read_loop: LOOP
      -- Fetch data into variables
      FETCH employee_cursor INTO emp_name, emp_salary;

      IF done THEN
         LEAVE read_loop;
      END IF;

      SET emp_salary= emp_salary * 0.10;

      UPDATE employee
      SET bonus = emp_salary
```

```
      WHERE Name = emp_name;
   END LOOP;


   CLOSE employee_cursor;
END ;

DELIMITER  ;

CALL CalculateBonus();
```

2. Create procddure to calculate total marks of given subject id.
Hint :Use cursor, use stud_sub for calculating total marks

```
DELIMITER //

create procedure procur2(IN sid int)
   -> begin
   ->  declare done int default 0;
   ->  declare total int;
   ->  declare cur cursor for select sum(marks) from stud_sub where subid = sid;
   ->  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
   ->  OPEN cur;
   ->  label:loop
   ->        fetch cur into total;
   ->              if done = 1 then
   ->                    leave label;
   ->              end if;
   ->        select total;
   ->  end loop;
   -> end ;
   -> $$
Query OK, 0 rows affected (0.59 sec)

mysql> delimiter ; mysql>
call procur2(2);
+.....+
```

| total | +-------
+
| 254 | +-------
+

3. Create procedure to update the salary increase salary by 300 of all given department into the procedure parameter
Use cursor

```
DELIMITER //

CREATE PROCEDURE UpdateSalaryByDepartment(IN departmentParam VARCHAR(25))
BEGIN
        DECLARE done INT DEFAULT 0;
    DECLARE empName VARCHAR(50);
  DECLARE currentSalary DECIMAL(10, 2);

  DECLARE employee_cursor CURSOR FOR
    SELECT Name, salary
    FROM employee
    WHERE department = departmentParam;


  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

  OPEN employee_cursor;


  read_loop: LOOP
    FETCH employee_cursor INTO empName, currentSalary;

    IF done THEN
      LEAVE read_loop;
    END IF;

    UPDATE employee
    SET salary = currentSalary + 300
    WHERE Name = empName;
```

```
    END LOOP;
    CLOSE employee_cursor;
END; //

DELIMITER ;
```

4. create procedure which pass the orderid as parameter and find the total quantity order form sales_order_detail, total of quantity order should be stored in OUT parameter. Use cursor

```
DELIMITER //

CREATE PROCEDURE GetTotalQuantityByOrder(
    IN orderIdParam Varchar(6),
    OUT totalQuantityParam INT)
BEGIN
    DECLARE done INT DEFAULT 0;
    DECLARE orderQuantity INT;
    DECLARE order_cursor CURSOR FOR
        SELECT sum(qtyordered)
        FROM sales_order_details
        WHERE orderno = orderIdParam;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
    SET totalQuantityParam = 0;

 OPEN order_cursor;

    read_loop: LOOP

        FETCH order_cursor INTO orderQuantity;

        IF done THEN
            LEAVE read_loop;
        END IF;
```

```
        SET totalQuantityParam = orderQuantity;
    END LOOP;

    CLOSE order_cursor;
END ;
//
DELIMITER ;
CALL GetTotalQuantityByOrder(1, @totalQuantity);
SELECT @totalQuantity AS TotalQuantity;
```

5 Create procedure which calculate total price of order number O19001.

Hint : use any technique of subquery/nested query for creating cursor.

Product_master and sales_order_detail

```
 ->delimiter $$ mysql> create procedure
CalculatePrice()
    -> begin
    ->  declare done int default 0;
    ->  declare total_price int;
    ->
    ->  declare cur cursor for select sum(pm.sellprice*sod.qtyordered) from
    ->  product_master as pm join sales_order_details as sod on
    ->  sod.productno = pm.productno where orderno = 'O19001' group by sod.orderno;
    ->
    ->  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
    ->  OPEN cur;
    ->  label:loop
    ->          fetch cur into total_price;
    ->                  if done = 1 then
    ->                          leave label;
    ->                  end if;
    ->          select total_price;
```

```
   ->  end loop;
   -> end;
   -> $$
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ; mysql>
call CalculatePrice;
+-------------+
| total_price | +-------------
+
|        2100 |
+-------------+
1 row in set (0.03 sec)

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