

LAB - 10

DBMS ASSIGNMENT

Examples:

STORED PROCEDURES AND FUNCTIONS:-

1Q)

```
mysql> delimiter //
mysql> create procedure display_book()
  -> begin
  -> select *from book;
  -> end //
Query OK, 0 rows affected (0.01 sec)

mysql> call display_book();//
+-----+-----+-----+-----+-----+-----+-----+
| BookId | ISBN | book_name                | author | ed_num | price | pages |
+-----+-----+-----+-----+-----+-----+-----+
| 1      | 1    | Glimpses of the past     | 1      | 1      | 650   | 396   |
| 2      | 2    | Beyond The Horizons of Venus | 1      | 1      | 650   | 396   |
| 3      | 3    | Ultrasonic Aquaculture   | 2      | 1      | 799   | 500   |
| 4      | 4    | Cryogenic Engines        | 2      | 1      | 499   | 330   |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

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

2Q)

```
mysql> delimiter //
mysql> create procedure update_price (IN temp_ISBN varchar(10), IN new_price
  -> integer) begin
  -> update book set price=new_price where ISBN=temp_ISBN;
  -> end //
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> call update_price(1,2216)//
Query OK, 1 row affected (0.01 sec)

mysql> call display_book()//
+-----+-----+-----+-----+-----+-----+
| BookId | ISBN | book_name          | author | ed_num | price | pages |
+-----+-----+-----+-----+-----+-----+
| 1      | 1    | Glimpses of the past | 1      | 1      | 2216  | 396   |
| 2      | 2    | Beyond The Horizons of Venus | 1      | 1      | 650   | 396   |
| 3      | 3    | Ultrasonic Aquaculture | 2      | 1      | 799   | 500   |
| 4      | 4    | Cyrogenic Engines    | 2      | 1      | 499   | 330   |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

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

3Q)

```
mysql> delimiter //
mysql> create procedure disp_max(OUT highestprice integer)
-> begin
-> select max(price) into highestprice from book;
-> end //
Query OK, 0 rows affected (0.00 sec)

mysql> call disp_max(@N)//
Query OK, 1 row affected (0.01 sec)

mysql> select @N//
+-----+
| @N    |
+-----+
| 2216  |
+-----+
1 row in set (0.00 sec)
```

4Q)

```
mysql> delimiter //
mysql> create procedure disp_gender(INOUT mfgender integer, IN emp_gender varchar(6))
    -> begin
    -> select count(gender) into mfgender from author where gender= emp_gender;
    -> end //
Query OK, 0 rows affected (0.01 sec)

mysql> call disp_gender(@MFGENDER,'Female')//
Query OK, 1 row affected (0.00 sec)

mysql> Select @MFGENDER//
+-----+
| @MFGENDER |
+-----+
|          1 |
+-----+
1 row in set (0.00 sec)
```

TRIGGERS:-

1Q)

```
mysql> DELIMITER $$
mysql> CREATE TRIGGER `updateItemPrice`
    -> BEFORE UPDATE ON `items`
    -> FOR EACH ROW
    -> BEGIN
    -> IF NEW.cost <> OLD.cost
    ->
    -> THEN
    -> SET NEW.price = NEW.cost * 1.40;
    -> END IF ;
    -> END$$
Query OK, 0 rows affected (0.02 sec)

mysql> DELIMITER ;
mysql> Update items Set cost = 10.00 Where id = 1 ;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> Select * from items;
+-----+-----+-----+-----+
| id | name          | cost | price |
+-----+-----+-----+-----+
|  1 | Basic Widget |   10 |   14 |
|  2 | Micro
Widget |  0.95 |  1.35 |
|  3 | Mega Widget | 99.95 |  140 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

ERROR HANDLING:-

1Q)

```
mysql> DELIMITER //
```

```
mysql> CREATE PROCEDURE emp_details
```

```
  -> (
```

```
  -> InputID INTEGER
```

```
  -> ,InputName VARCHAR(50)
```

```
  -> ,InputDept VARCHAR(50)
```

```
  -> )
```

```
  -> BEGIN
```

```
  -> DECLARE EXIT HANDLER FOR SQLEXCEPTION
```

```
  -> SELECT 'Error occured';
```

```
  -> INSERT INTO employees VALUES(InputID, InputName, InputDept);
```

```
  -> SELECT *FROM employees;
```

```
  -> END
```

```
  -> // DELIMITER ;
```

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

```
mysql> call emp_details(200,'Johny','HR');
```

ID	name	department
200	Johny	HR

```
1 row in set (0.02 sec)
```

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

```
mysql> call emp_details(200,'Johny','HR');call emp_details(200,'Johny','HR');
```

Error occured
Error occured

MYSQL CURSOR:-

1Q)

```
mysql> DELIMITER $$
mysql> CREATE PROCEDURE emp_curs()
-> BEGIN
-> DECLARE finished INTEGER DEFAULT 0;
-> DECLARE ename varchar(100);
-> DECLARE eplace varchar(100);
-> -- declare cursor for employee name and place
->
-> DECLARE curname CURSOR FOR SELECT emp_name, place FROM
-> employee; -- declare NOT FOUND handler
-> DECLARE CONTINUE HANDLER FOR NOT FOUND SET finished = 1;
-> OPEN curname;
-> getname: LOOP
-> FETCH curname INTO ename, eplace;
-> IF finished = 1 THEN
-> LEAVE getname;
-> END IF;
-> -- build employee names
-> SELECT ename,eplace;
-> END LOOP getname;
-> CLOSE curname;
-> END$$
```

Query OK, 0 rows affected (0.01 sec)

```
mysql> DELIMITER ;
mysql> call emp_curs();
```

```
+-----+-----+
| ename | eplace |
+-----+-----+
| peter | Newyork |
+-----+-----+
1 row in set (0.00 sec)
```

```
+-----+-----+
| ename | eplace |
+-----+-----+
| Mark  | California |
+-----+-----+
1 row in set (0.00 sec)
```

```
+-----+-----+
| ename | eplace |
+-----+-----+
| Donald | Arizona |
+-----+-----+
1 row in set (0.00 sec)
```

```
+-----+-----+
|  ename | eplace |
+-----+-----+
|  Obama | Florida |
+-----+-----+
1 row in set (0.01 sec)
```

```
+-----+-----+
|  ename | eplace |
+-----+-----+
| Linklon | Georgia |
+-----+-----+
1 row in set (0.01 sec)
```

```
+-----+-----+
|  ename | eplace |
+-----+-----+
|  Kane  | Alaska |
+-----+-----+
1 row in set (0.02 sec)
```

```
+-----+-----+
|  ename | eplace |
+-----+-----+
|  Adam  | California |
+-----+-----+
1 row in set (0.02 sec)
```

```
+-----+-----+
|  ename | eplace |
+-----+-----+
|  Mac   | Florida |
+-----+-----+
1 row in set (0.04 sec)
```

```
+-----+-----+
|  ename | eplace |
+-----+-----+
|  Manas | India |
+-----+-----+
1 row in set (0.05 sec)
```

```
+-----+-----+
|  ename | eplace |
+-----+-----+
|  Vasin | India |
+-----+-----+
1 row in set (0.05 sec)
```

```
+-----+-----+
|  ename | eplace |
+-----+-----+
| peter | Newyork |
+-----+-----+
1 row in set (0.07 sec)
```

```
+-----+-----+
|  ename | eplace |
+-----+-----+
|  Mark  | California |
+-----+-----+
```

```
+-----+-----+
|  ename  | eplace  |
+-----+-----+
| Donald  | Arizona |
+-----+-----+
1 row in set (0.08 sec)
```

```
+-----+-----+
|  ename  | eplace  |
+-----+-----+
| Obama   | Florida |
+-----+-----+
1 row in set (0.10 sec)
```

```
+-----+-----+
|  ename  | eplace  |
+-----+-----+
| Linklon | Georgia |
+-----+-----+
1 row in set (0.11 sec)
```

```
+-----+-----+
|  ename  | eplace  |
+-----+-----+
| Kane    | Alaska  |
+-----+-----+
1 row in set (0.11 sec)
```

```
+-----+-----+
|  ename  | eplace  |
+-----+-----+
| Adam    | California |
+-----+-----+
1 row in set (0.13 sec)
```

```
+-----+-----+
|  ename  | eplace  |
+-----+-----+
| Mac     | Florida |
+-----+-----+
1 row in set (0.14 sec)
```

```
+-----+-----+
|  ename  | eplace  |
+-----+-----+
| Manas   | India   |
+-----+-----+
1 row in set (0.14 sec)
```

```
+-----+-----+
|  ename  | eplace  |
+-----+-----+
| Vasin   | India   |
+-----+-----+
1 row in set (0.15 sec)
```

Query OK, 0 rows affected (0.16 sec)

2Q)

```
mysql> DELIMITER //
mysql> DROP PROCEDURE IF EXISTS product_count;
mysql> CREATE PROCEDURE product_count()
  -> BEGIN
  -> DECLARE p_type VARCHAR(255);
  -> DECLARE p_count INT;
  -> DECLARE done INT DEFAULT 0;
  -> DECLARE product_curs CURSOR FOR
  -> SELECT type,COUNT(*)FROM product GROUP BY type;
  -> DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET done = 1;
  -> TRUNCATE product_type_count;
  -> OPEN product_curs;
  -> REPEAT
  -> FETCH product_curs
  -> INTO p_type, p_count;
  -> IF NOT done
  -> THEN
  -> INSERT INTO product_type_count
  -> SET
  -> type = p_type,
  -> count = p_count;
  -> END IF;
  -> UNTIL done
  ->
  -> END REPEAT;
  -> CLOSE product_curs;
  -> END //
```

Query OK, 0 rows affected, 1 warning (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

```
mysql> DELIMITER ;
mysql> call product_count();
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> select * from product_type_count;
+-----+-----+
| type  | count |
+-----+-----+
| dress |      2 |
| food  |      3 |
+-----+-----+
2 rows in set (0.00 sec)
```


LAB QUESTIONS:

3Q)

```
1 • use lab10;
2
3   delimiter //
4 • create procedure BOOK_AVG(in auth int)
5   begin
6     declare AVG_Price int;
7     declare COUNT_Author int;
8     select COUNT(book_name) into COUNT_Author from book WHERE author = auth;
9     Select AVG(price) into AVG_Price from book WHERE author = auth;
10    Select authorName,COUNT_Author as Number_of_Books,AVG_Price from author WHERE author_id = auth;
11
12  end //
13
14 • call BOOK_AVG(1)//
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	authorName	Number_of_Books	AVG_Price
1	Kraig Muller	2	1433

