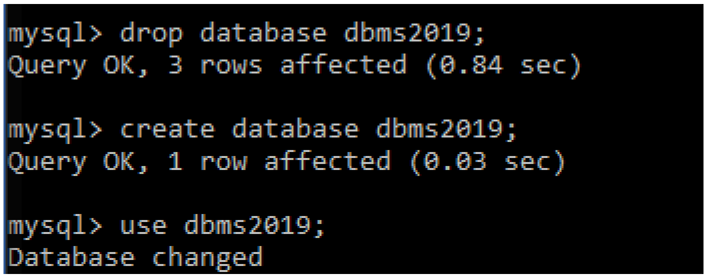
Introduction to Database Systems (CSD202)

Graded Lab#2

Date: 12/04/19
[10 Marks]

You have to upload the screen shot (black screen) of output with the queries one by one, in a word file & file name must be-> (Roll No)_GL#2

1. Remove the DBMS2019, if exist & create new database DBMS2019.

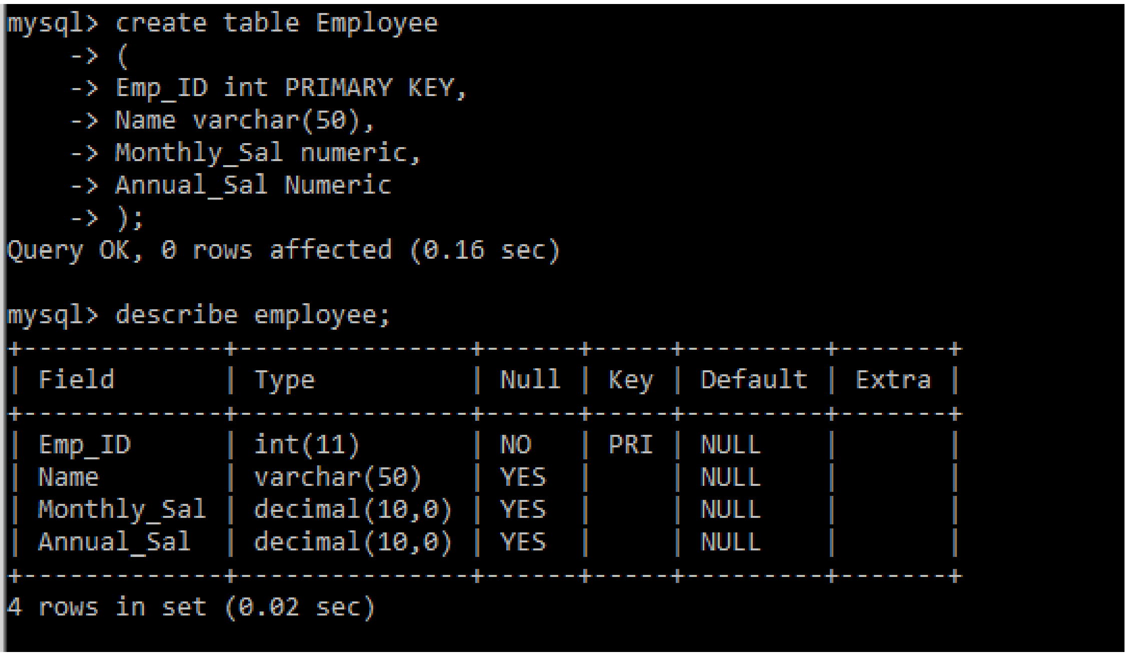


```
mysql> drop database dbms2019;
Query OK, 3 rows affected (0.84 sec)

mysql> create database dbms2019;
Query OK, 1 row affected (0.03 sec)

mysql> use dbms2019;
Database changed
```

2. Create a table employee. **[1.0 Marks]**

Structure for Table: Employee

Column Name	Data Declaration
Emp_ID	Int, Primary Key
Name	Varchar(50)
Monthly_Sal	Numeric
Annual_Sal	Numeric



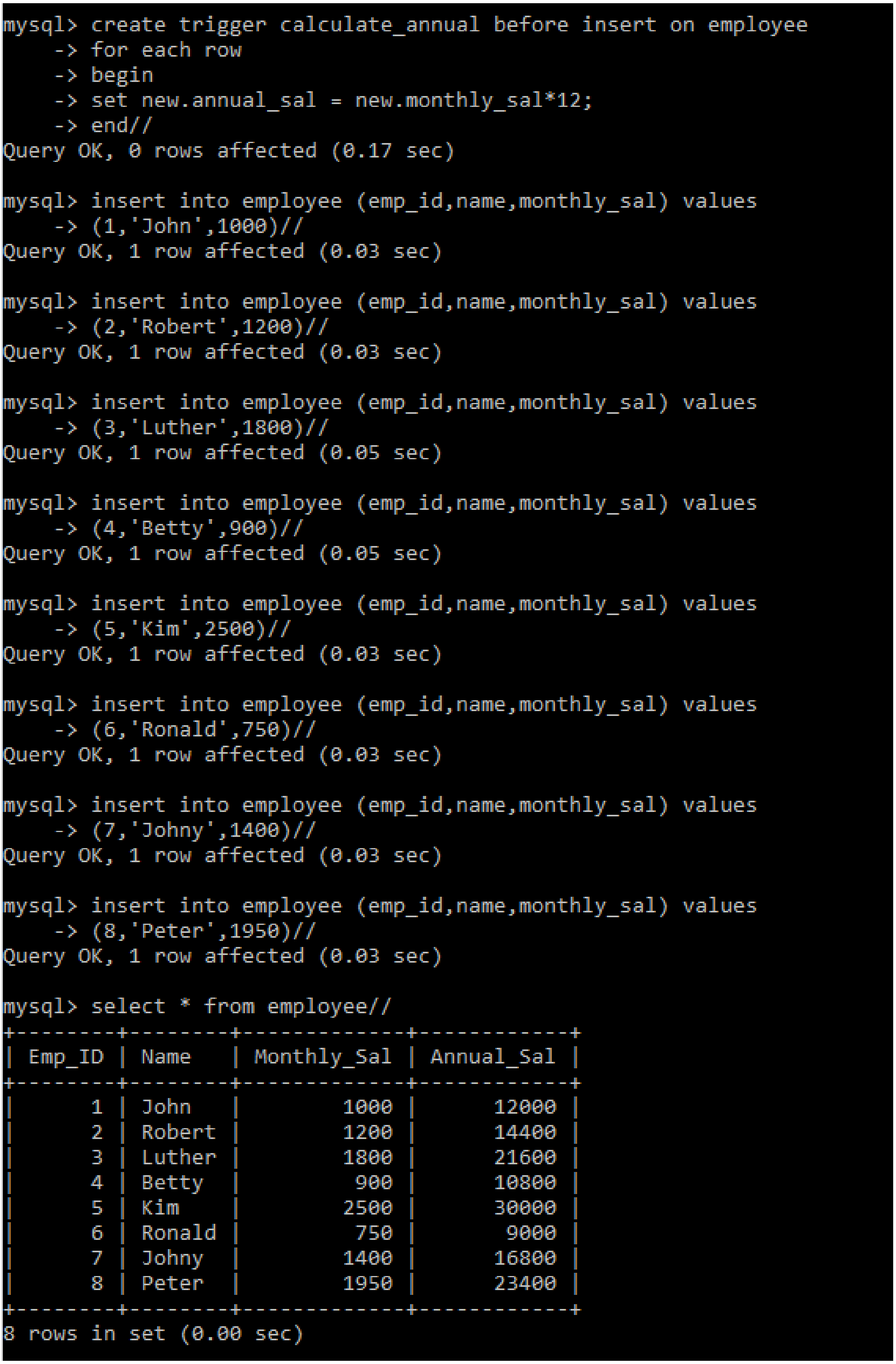
```
mysql> create table Employee
    -> (
    -> Emp_ID int PRIMARY KEY,
    -> Name varchar(50),
    -> Monthly_Sal numeric,
    -> Annual_Sal Numeric
    -> );
Query OK, 0 rows affected (0.16 sec)

mysql> describe employee;
+-------------+---------------+------+-----+---------+-------+
| Field       | Type          | Null | Key | Default | Extra |
+-------------+---------------+------+-----+---------+-------+
| Emp_ID      | int(11)       | NO   | PRI | NULL    |       |
| Name        | varchar(50)   | YES  |     | NULL    |       |
| Monthly_Sal | decimal(10,0) | YES  |     | NULL    |       |
| Annual_Sal  | decimal(10,0) | YES  |     | NULL    |       |
+-------------+---------------+------+-----+---------+-------+
4 rows in set (0.02 sec)
```

3. Create a trigger **Calculate_Annual** (before insert) for table employee, at the time of data insertion you will only insert monthly salary & annual salary will be automatically calculated by the calculate_Annual trigger.

[3.0 Marks]

Now insert the data as mentioned below:

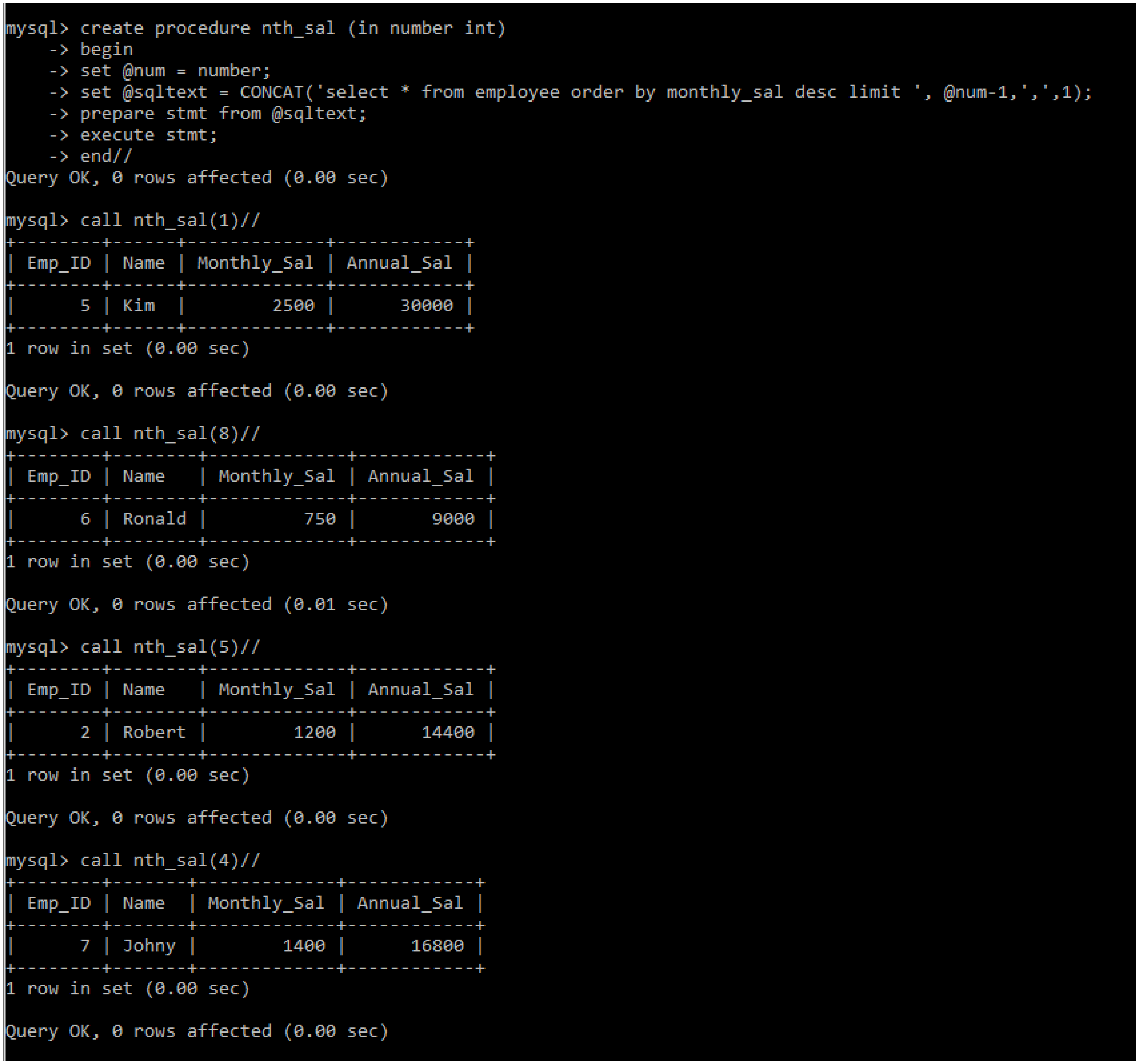
Emp_ID	**Name**	**Monthly_Sal**	**Annual_Sal**
1	John	1000	
2	Robert	1200	
3	Luther	1800	
4	Betty	900	
5	Kim	2500	
6	Ronald	750	
7	Johny	1400	
8	Peter	1950	



```
mysql> create trigger calculate_annual before insert on employee
    -> for each row
    -> begin
    -> set new.annual_sal = new.monthly_sal*12;
    -> end//
Query OK, 0 rows affected (0.17 sec)

mysql> insert into employee (emp_id,name,monthly_sal) values
    -> (1,'John',1000)//
Query OK, 1 row affected (0.03 sec)

mysql> insert into employee (emp_id,name,monthly_sal) values
    -> (2,'Robert',1200)//
Query OK, 1 row affected (0.03 sec)

mysql> insert into employee (emp_id,name,monthly_sal) values
    -> (3,'Luther',1800)//
Query OK, 1 row affected (0.05 sec)

mysql> insert into employee (emp_id,name,monthly_sal) values
    -> (4,'Betty',900)//
Query OK, 1 row affected (0.05 sec)

mysql> insert into employee (emp_id,name,monthly_sal) values
    -> (5,'Kim',2500)//
Query OK, 1 row affected (0.03 sec)

mysql> insert into employee (emp_id,name,monthly_sal) values
    -> (6,'Ronald',750)//
Query OK, 1 row affected (0.03 sec)

mysql> insert into employee (emp_id,name,monthly_sal) values
    -> (7,'Johny',1400)//
Query OK, 1 row affected (0.03 sec)

mysql> insert into employee (emp_id,name,monthly_sal) values
    -> (8,'Peter',1950)//
Query OK, 1 row affected (0.03 sec)

mysql> select * from employee//
+--------+--------+-------------+------------+
| Emp_ID | Name   | Monthly_Sal | Annual_Sal |
+--------+--------+-------------+------------+
|      1 | John   |        1000 |      12000 |
|      2 | Robert |        1200 |      14400 |
|      3 | Luther |        1800 |      21600 |
|      4 | Betty  |         900 |      10800 |
|      5 | Kim    |        2500 |      30000 |
|      6 | Ronald |         750 |       9000 |
|      7 | Johny  |        1400 |      16800 |
|      8 | Peter  |        1950 |      23400 |
+--------+--------+-------------+------------+
8 rows in set (0.00 sec)
```

4. Create a procedure **nth_Sal** to find out the details of the employee who is getting nth highest salary. You will only pass the number means 6th highest, 7th hieghest,8th highest etc… at the time of procedure call. For example, find out the details of the employee who is getting 4th highest salary in procedure call.

[3.0 Marks]



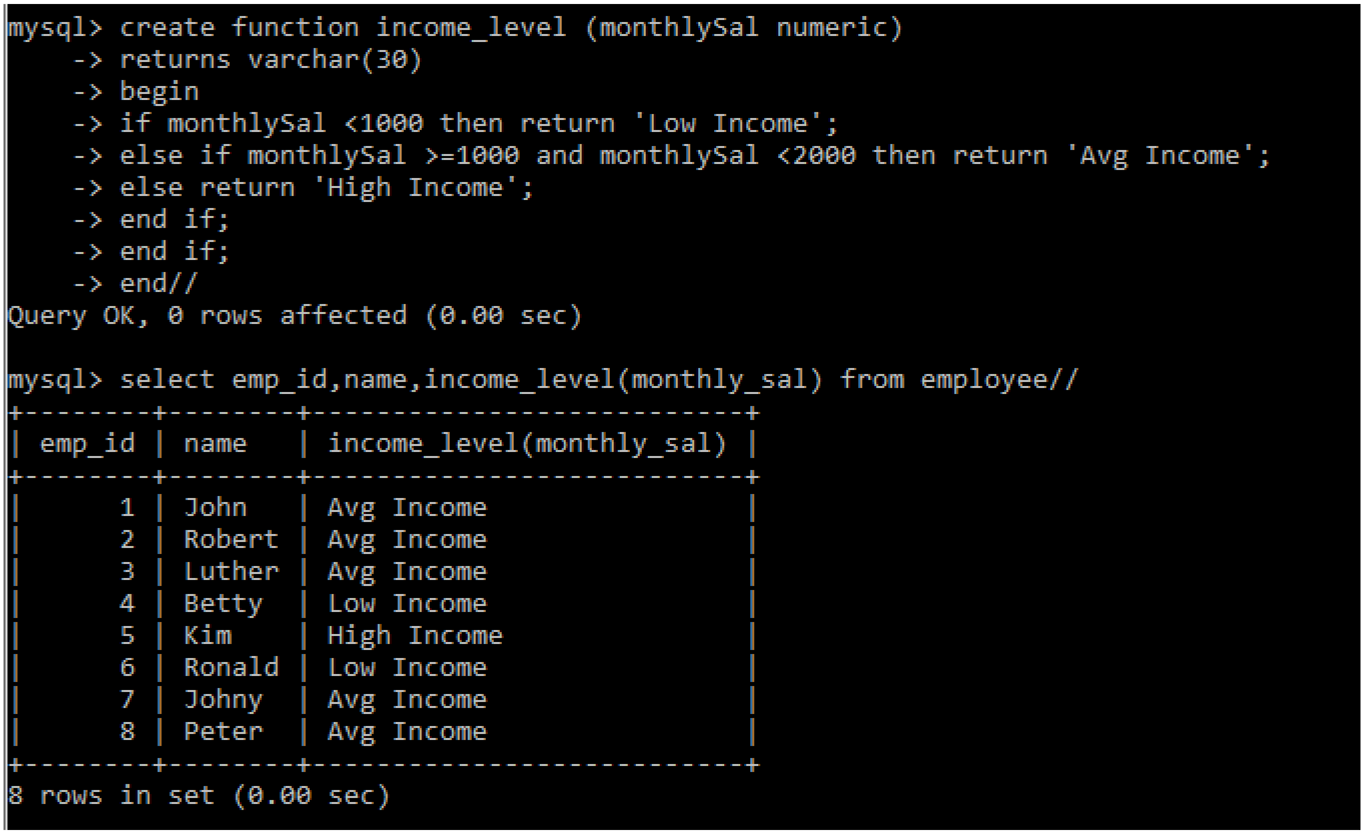
```
mysql> create procedure nth_sal (in number int)
    -> begin
    -> set @num = number;
    -> set @sqltext = CONCAT('select * from employee order by monthly_sal desc limit ', @num-1,',',1);
    -> prepare stmt from @sqltext;
    -> execute stmt;
    -> end//
Query OK, 0 rows affected (0.00 sec)

mysql> call nth_sal(1)//
+--------+------+-------------+------------+
| Emp_ID | Name | Monthly_Sal | Annual_Sal |
+--------+------+-------------+------------+
|      5 | Kim  |        2500 |      30000 |
+--------+------+-------------+------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> call nth_sal(8)//
+--------+--------+-------------+------------+
| Emp_ID | Name   | Monthly_Sal | Annual_Sal |
+--------+--------+-------------+------------+
|      6 | Ronald |         750 |       9000 |
+--------+--------+-------------+------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> call nth_sal(5)//
+--------+--------+-------------+------------+
| Emp_ID | Name   | Monthly_Sal | Annual_Sal |
+--------+--------+-------------+------------+
|      2 | Robert |        1200 |      14400 |
+--------+--------+-------------+------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> call nth_sal(4)//
+--------+-------+-------------+------------+
| Emp_ID | Name  | Monthly_Sal | Annual_Sal |
+--------+-------+-------------+------------+
|      7 | Johny |        1400 |      16800 |
+--------+-------+-------------+------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

5. create a function **income_level** to check the income level of the employee's as per given below conditions:

[3.0 Marks]

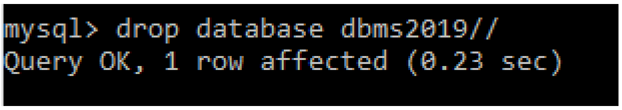
If Monthly_Sal < 1000 then income level is “Low Income”
If Monthly_Sal >=1000 and Monthly_Sal <2000 then income level is “Avg Income”
If Monthly_Sal >= 2000 then income level is “High Income”.

Call this function for employee table and display Emp_ID, Name & income_level.



```
mysql> create function income_level (monthlySal numeric)
    -> returns varchar(30)
    -> begin
    -> if monthlySal <1000 then return 'Low Income';
    -> else if monthlySal >=1000 and monthlySal <2000 then return 'Avg Income';
    -> else return 'High Income';
    -> end if;
    -> end if;
    -> end//
Query OK, 0 rows affected (0.00 sec)

mysql> select emp_id,name,income_level(monthly_sal) from employee//
+--------+--------+---------------------------+
| emp_id | name   | income_level(monthly_sal) |
+--------+--------+---------------------------+
|      1 | John   | Avg Income                |
|      2 | Robert | Avg Income                |
|      3 | Luther | Avg Income                |
|      4 | Betty  | Low Income                |
|      5 | Kim    | High Income               |
|      6 | Ronald | Low Income                |
|      7 | Johny  | Avg Income                |
|      8 | Peter  | Avg Income                |
+--------+--------+---------------------------+
8 rows in set (0.00 sec)
```

Drop database DBMS2019



```
mysql> drop database dbms2019//
Query OK, 1 row affected (0.23 sec)
```