MYSQL Assignment :

1. Write a stored procedure to paginate the data that comes out of a query

- The queries for creation of table and insertion of data are as below :

  create table teacher(id int primary key,name varchar(10));

  insert into teacher values(1,"Rohit");

  insert into teacher values(2,"Rahul");

  insert into teacher values(3,"Rama"); and so on.

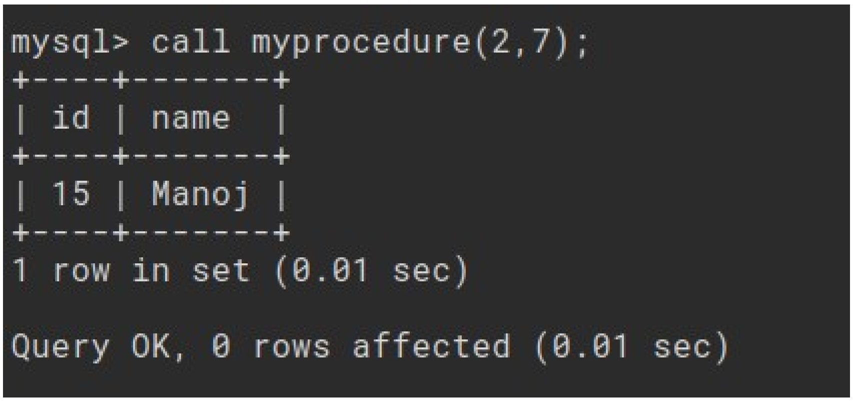
- The syntax for the procedure is as below :

```
CREATE PROCEDURE myprocedure (PageNumber int, NumberOFRows int)
BEGIN
DECLARE DataLim INT DEFAULT 1;
set DataLim = NumberOfRows*(PageNumber);
select * from teacher limit DataLim, NumberOfRows;
END
```

- The syntax to call it is :

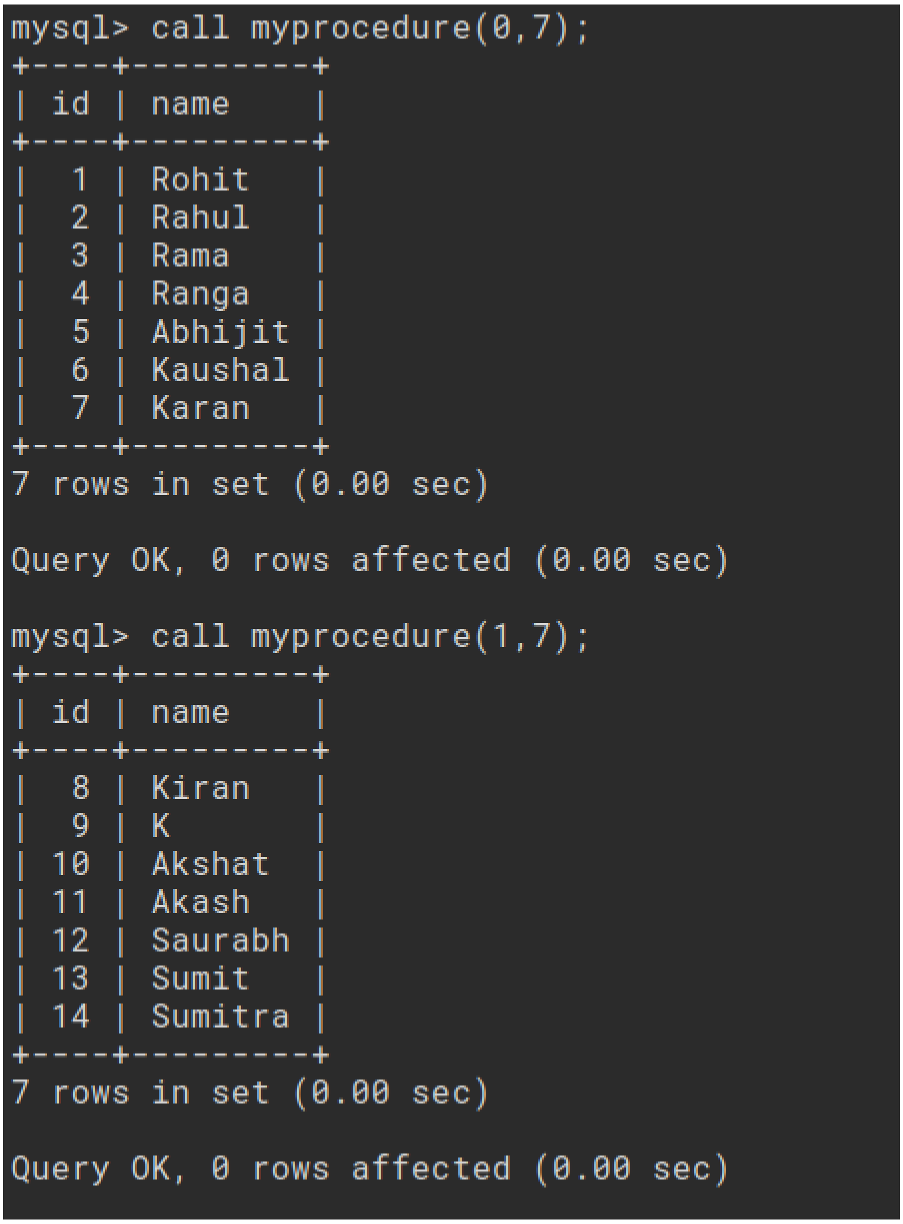
  call myprocedure(3,8);

- Some screenshots of the output are :



```
mysql> call myprocedure(2,7);
+----+-------+
| id | name  |
+----+-------+
| 15 | Manoj |
+----+-------+
1 row in set (0.01 sec)

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

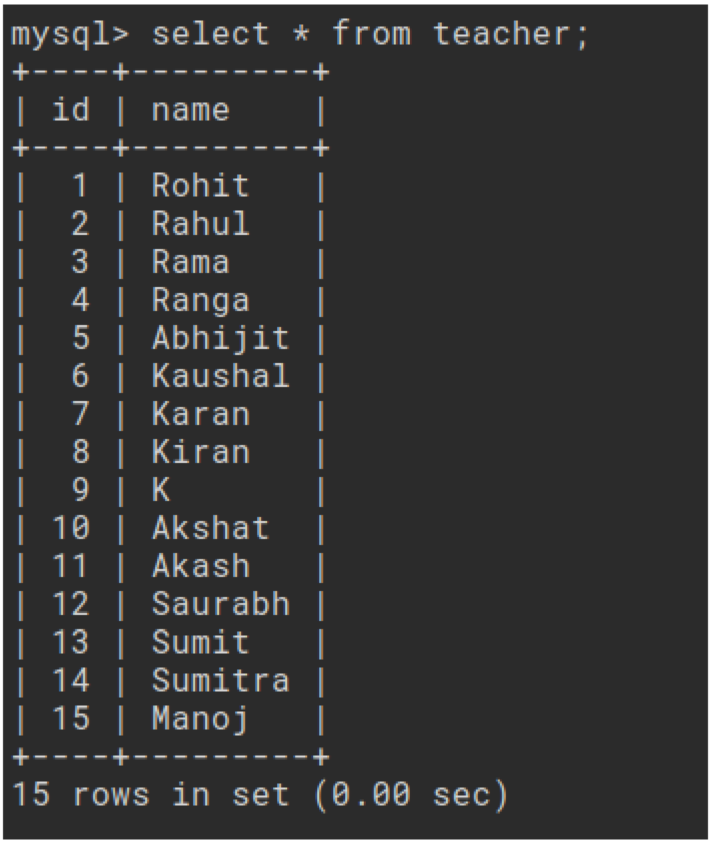


```
mysql> call myprocedure(0,7);
+----+---------+
| id | name    |
+----+---------+
|  1 | Rohit   |
|  2 | Rahul   |
|  3 | Rama    |
|  4 | Ranga   |
|  5 | Abhijit |
|  6 | Kaushal |
|  7 | Karan   |
+----+---------+
7 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> call myprocedure(1,7);
+----+---------+
| id | name    |
+----+---------+
|  8 | Kiran   |
|  9 | K       |
| 10 | Akshat  |
| 11 | Akash   |
| 12 | Saurabh |
| 13 | Sumit   |
| 14 | Sumitra |
+----+---------+
7 rows in set (0.00 sec)

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



```
mysql> select * from teacher;
+----+---------+
| id | name    |
+----+---------+
|  1 | Rohit   |
|  2 | Rahul   |
|  3 | Rama    |
|  4 | Ranga   |
|  5 | Abhijit |
|  6 | Kaushal |
|  7 | Karan   |
|  8 | Kiran   |
|  9 | K       |
| 10 | Akshat  |
| 11 | Akash   |
| 12 | Saurabh |
| 13 | Sumit   |
| 14 | Sumitra |
| 15 | Manoj   |
+----+---------+
15 rows in set (0.00 sec)
```

2.Write a Brief Note on Before triggers.

- **Before Insert :**
    - The Create Before Insert Trigger in SQL fires the trigger before we insert the value into the table.
    - The syntax is :

      ```
      CREATE TRIGGER <TriggerName>

      Before Insert ON <TableName> FOR EACH ROW

      BEGIN

      END;
      ```
    - The above statement can not return any values. That is it cannot execute select statements.
- **Before Update :**
    - MySQL BEFORE UPDATE triggers are invoked automatically before an update event occurs on the table associated with the triggers.
    - The Syntax is :

      ```
      CREATE TRIGGER <TriggerName>

      Before Update ON <TableName> FOR EACH ROW

      BEGIN

      END;
      ```
    - Similarly this cannot return any values.

- **Before Delete :**
  - MySQL BEFORE Delete triggers are invoked automatically before a delete event occurs on the table associated with the triggers.
  - The Syntax is :

    ```
    CREATE TRIGGER <TriggerName>

    Before Delete ON <TableName> FOR EACH ROW

    BEGIN

    END;
    ```
  - Similarly this can not return values.