Name:= Madhuri Dattatray bagade

Student Id:- 14051

Project name:= SMS Based Remote Server Monitoring System

Project

SMS Based Remote Server Monitoring System

For Organizations with huge data centers having a lot of servers hosting numerous applications, it is always a major problem to monitor if each of the servers is up and functional all the time. The problem is more acute during late night shifts when the usual number of network/systems engineers working is less.

Usually, when organizations host the applications on their servers on behalf of their clients, they sign-up a service level agreement (SLA), specifying the allowed down time for each of the applications. Any lack of commitment on the part of the organizations in meeting the SLA could result in loss of business or legal action or both. So, it becomes very important for the organizations to know if a server is down or non-functional and take corrective action immediately.

Unfortunately, for some less time critical applications, it is usually the client who informs that there is a problem with the server when we tries to login to the application organisations would be very interested in knowing about server failures immediately and take corrective action before the client start complaining.

## **Existing system**

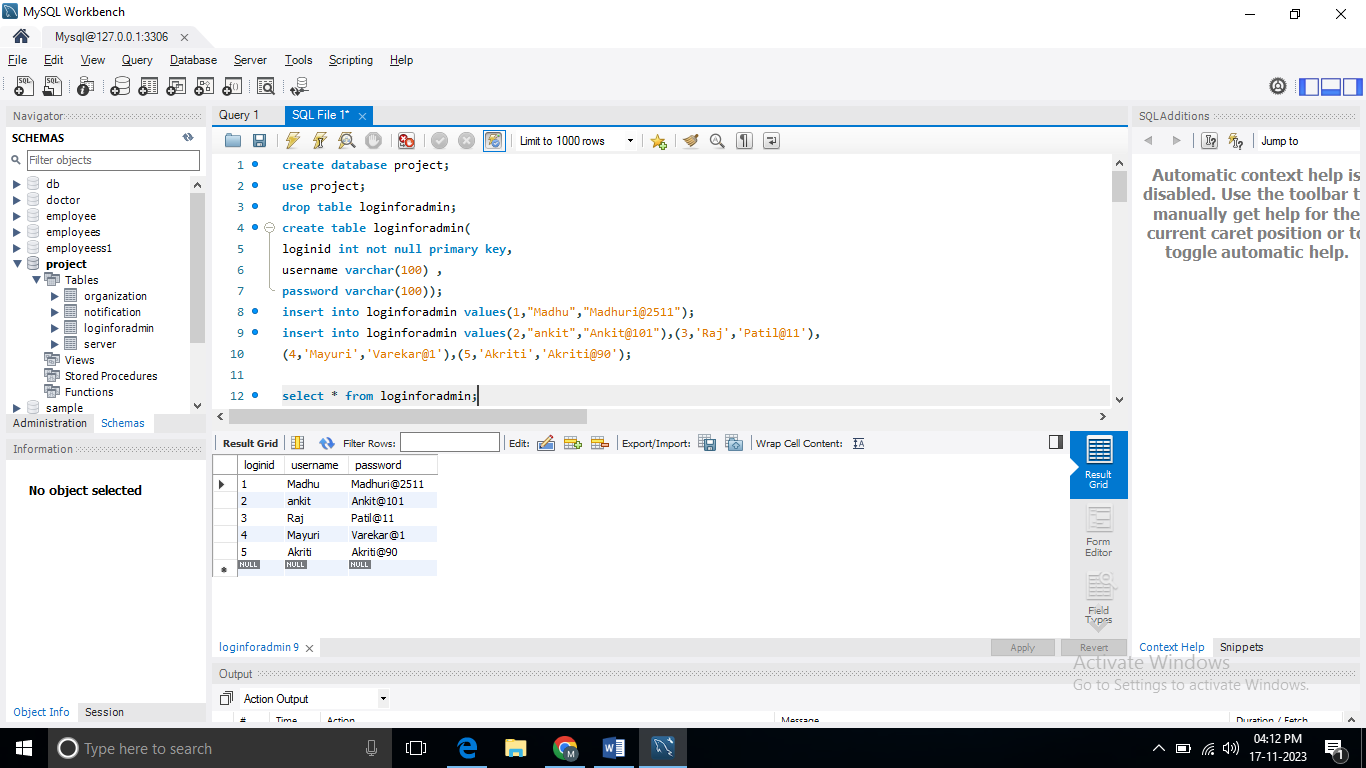
In the present Remote Server Monitoring System when the server failure happens then the network engineer makes the system working. But at the night time when there is less engineer available organization faces the problem, whereas the problem on the server is identified by the client and not by the organization. Even when the organization knows about the server problem, they have difficulty in removing them as the engineer at first needs to find out about the bad server. Whereas the problem can be temporarily, but it can occur multiple time because of this the engineer will face difficulty in the finding of the fault in the server.

## **Proposed system**

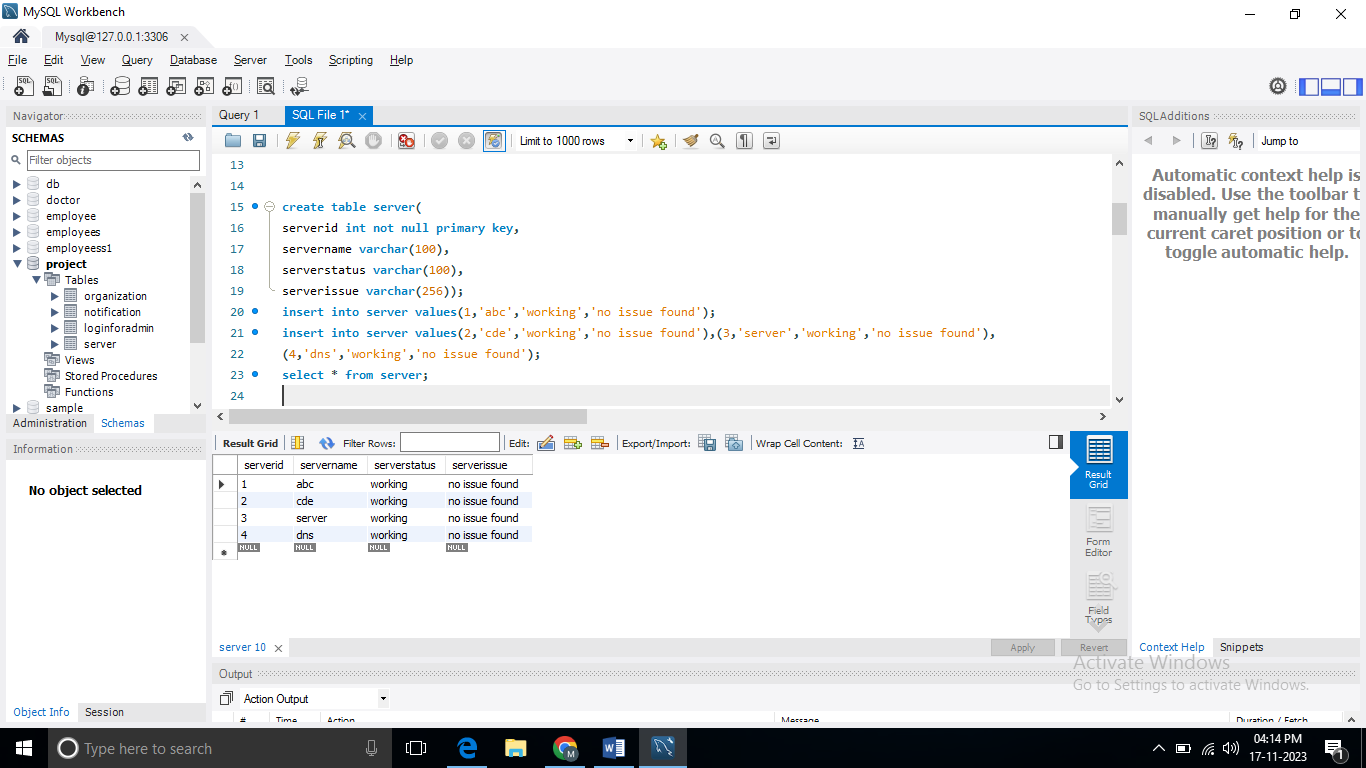
The proposed Remote Server Monitoring System will help the engineers about the server which is not working by providing them the details of the fault server. This Remote Server Monitoring System will monitor each server of the organization by sending a message to the servers. Then the server will respond to that message within the specified time, but if the server does not respond, then the system will send a notification to the admin. The notification will contain the information about the servers such that the time of failure, which server was/is not working.

**VIEWS**

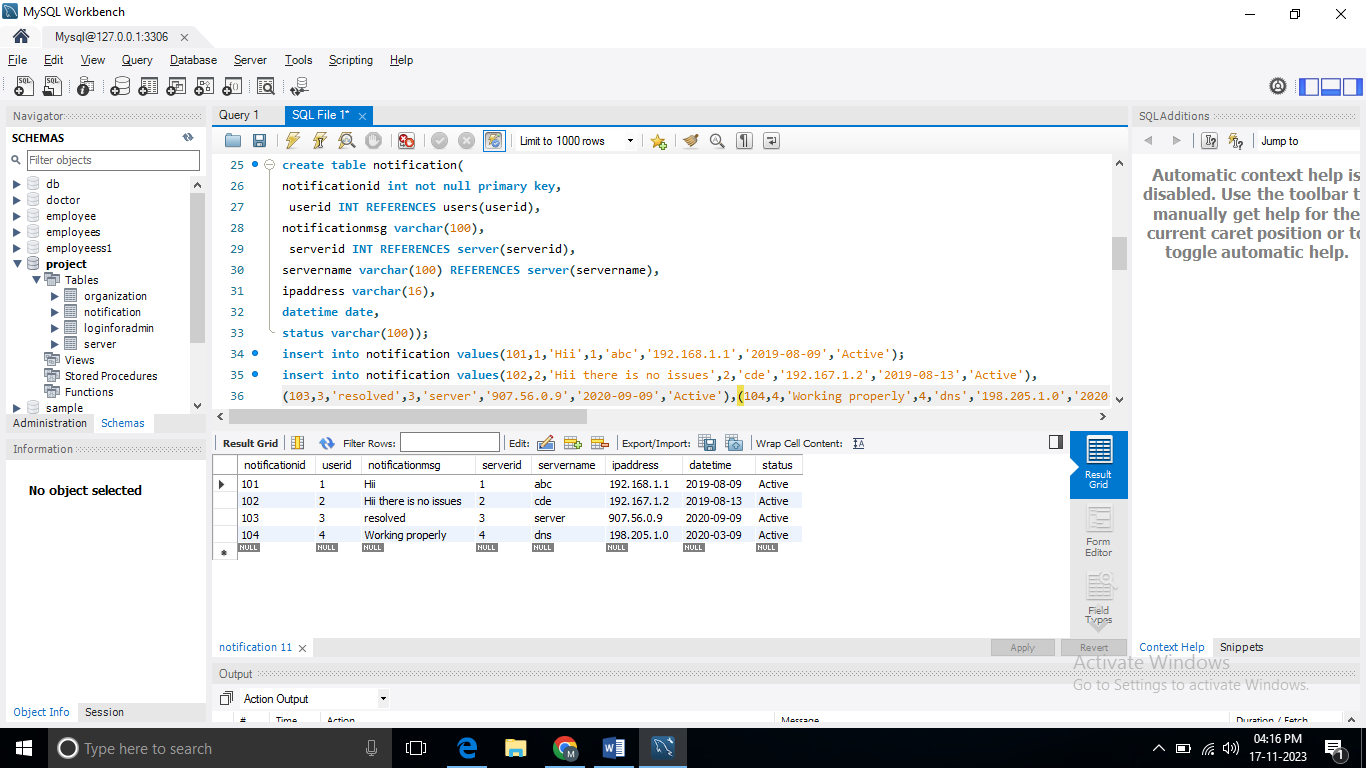
**Table Of Login :-**



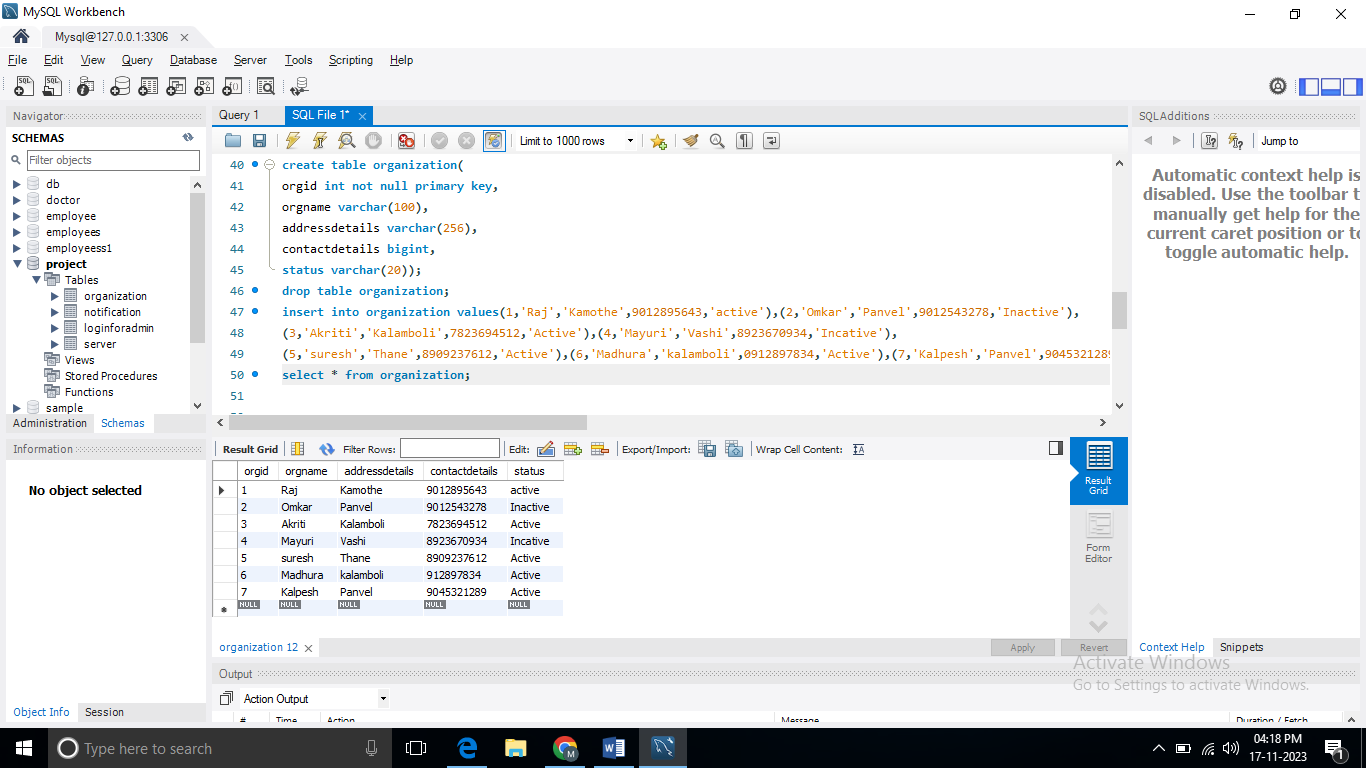
**Table Of Server:-**



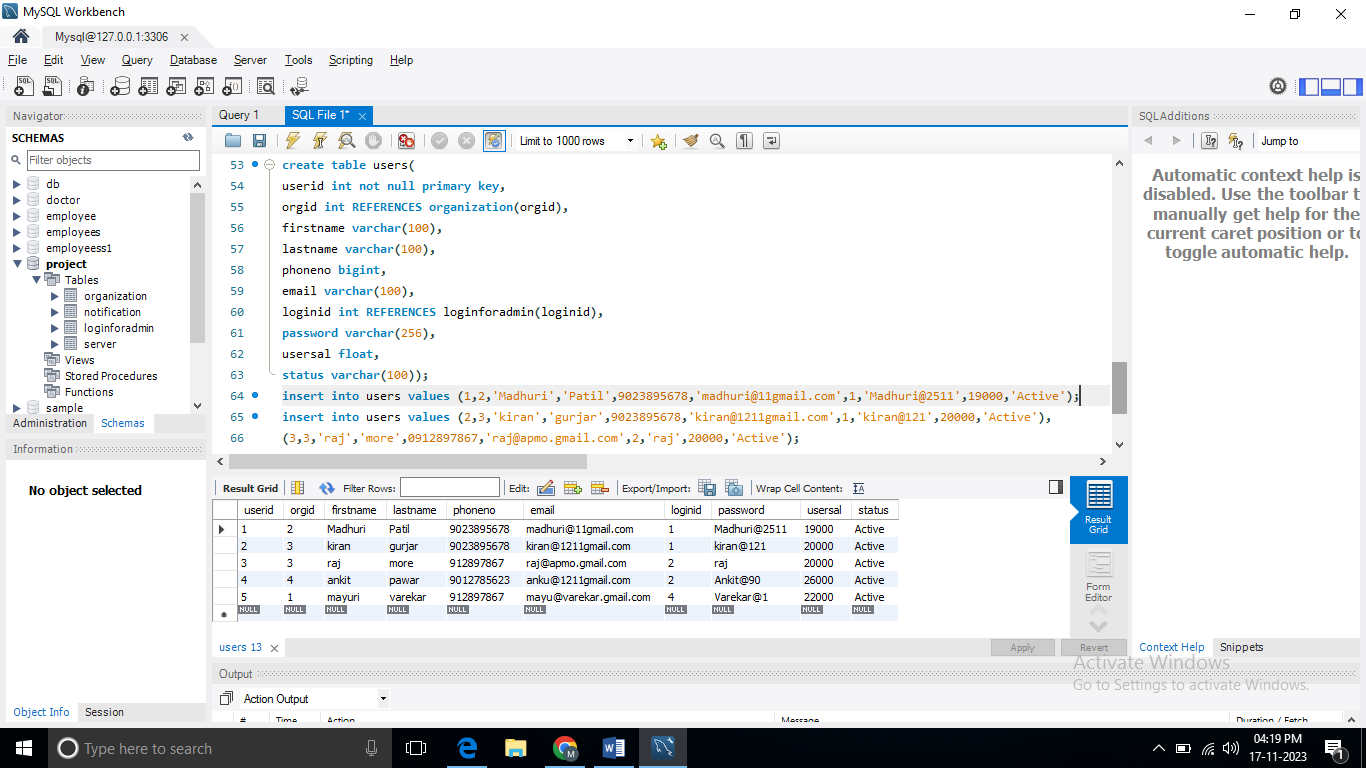
**Table Of Notification:-**



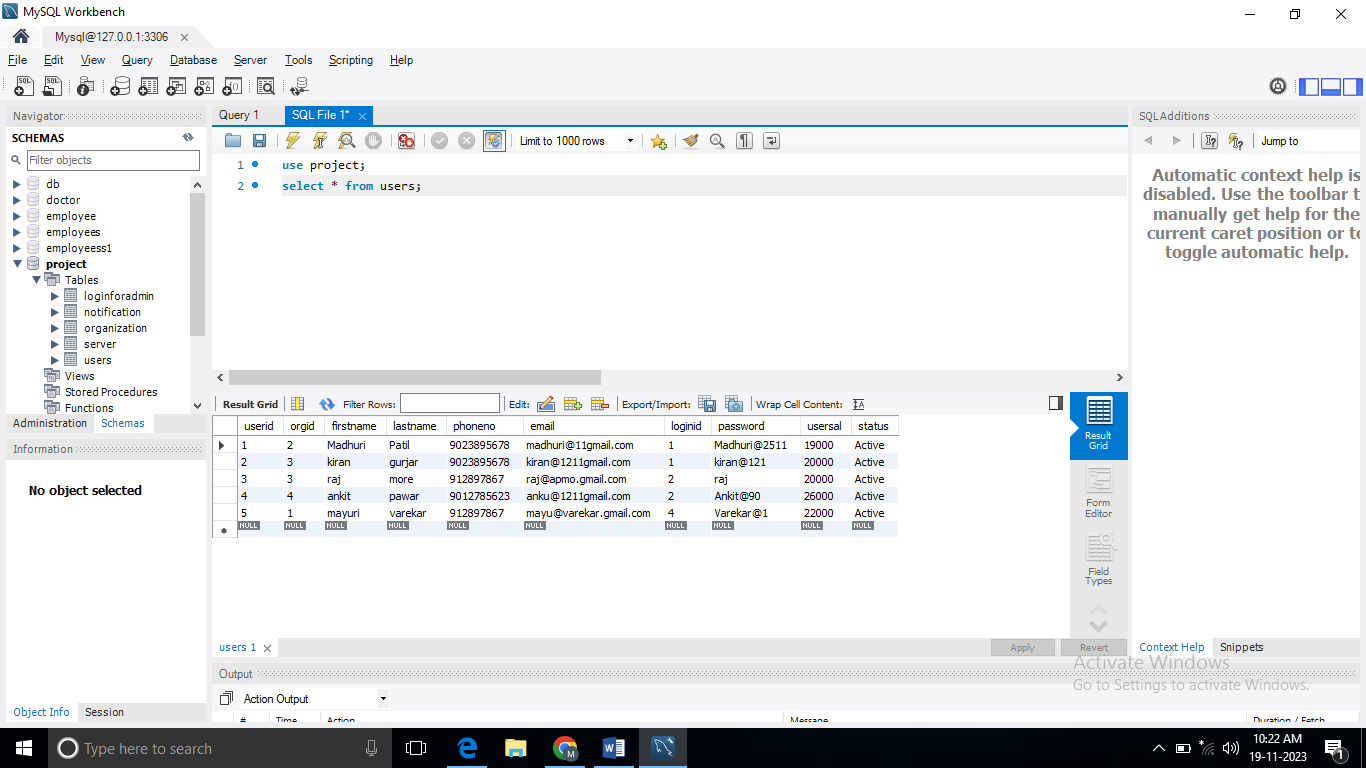
**Table Of Organization:-**



**Table Of Users:-**



**Show the data in users table :-**

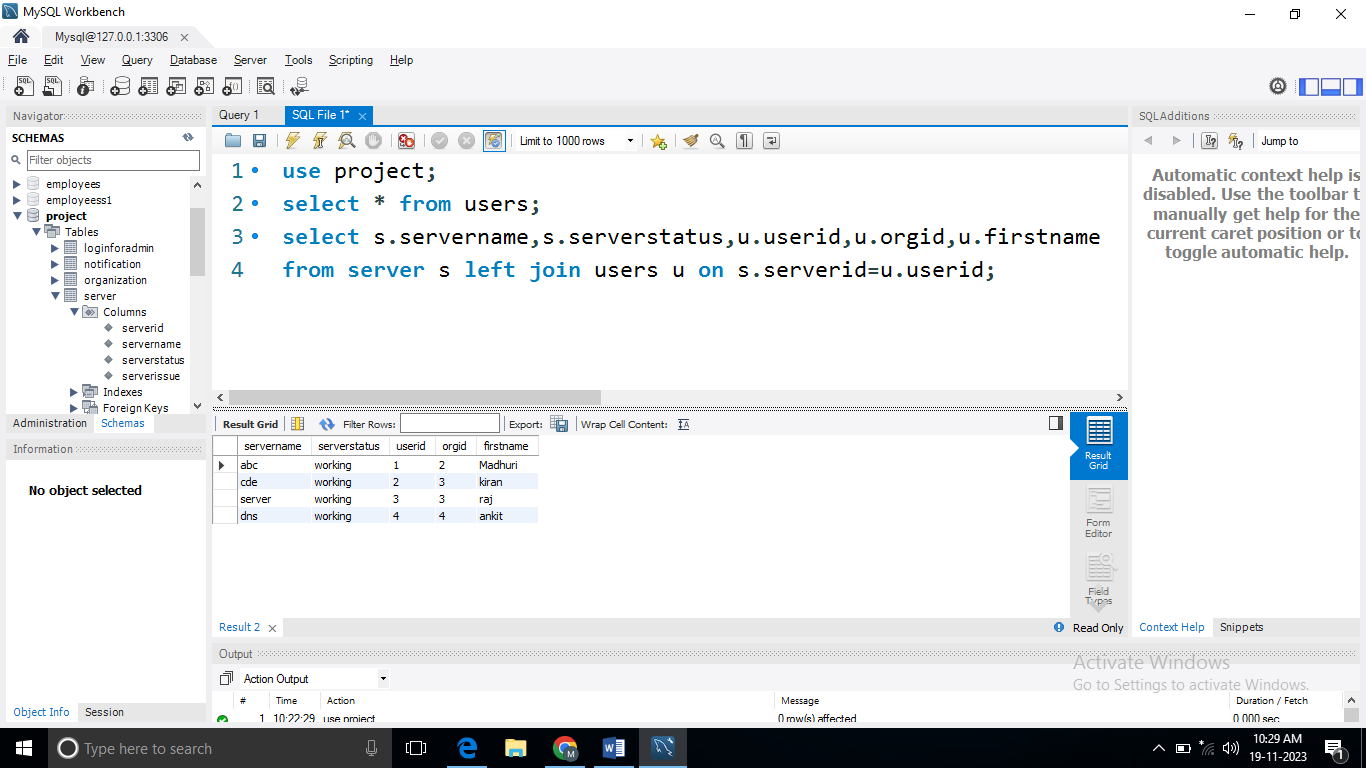


**Joins**

1. **LEFT JOIN:-**

**QUERY:**

* **select s.servername,s.serverstatus,u.userid,u.orgid,u.firstname**
* **from server s left join users u on s.serverid=u.userid;**

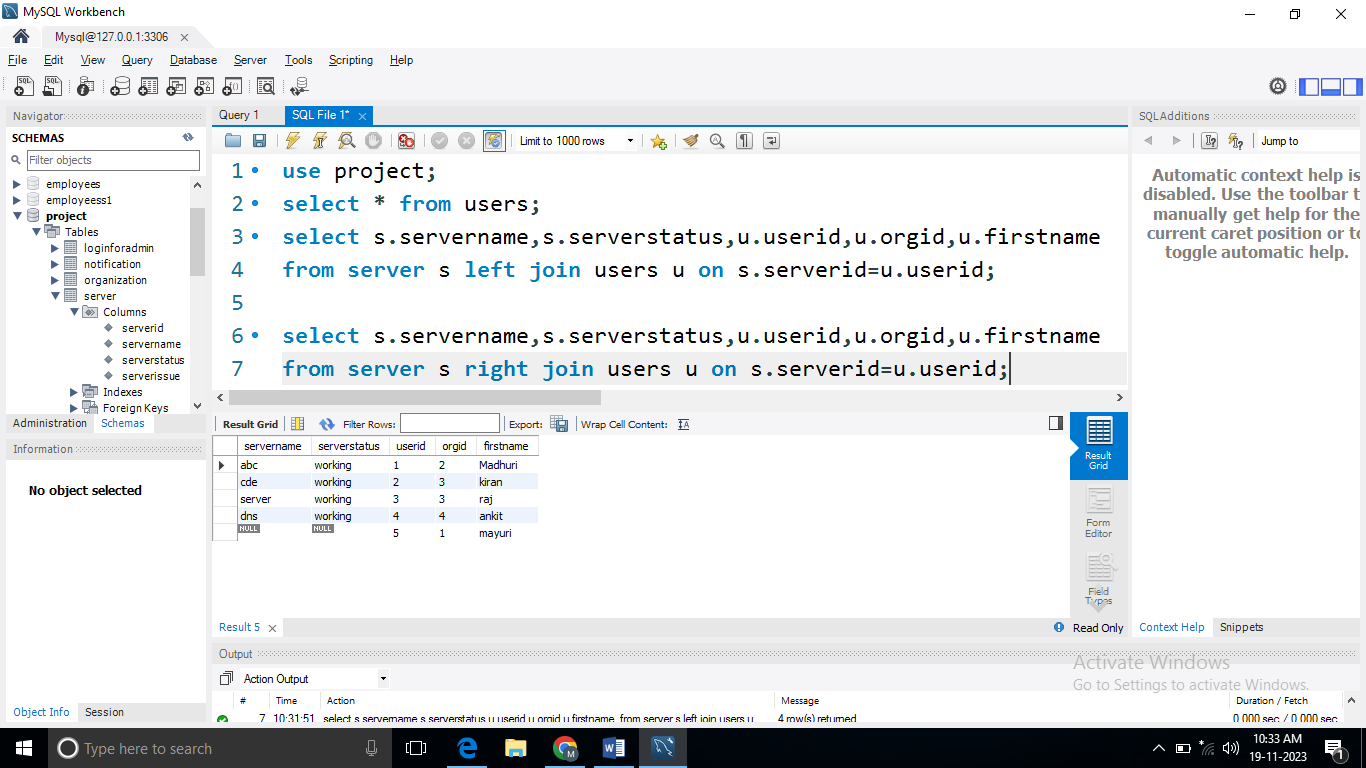


1. **RIGHT JOIN:-**

**QUERY:-**

* **select s.servername,s.serverstatus,u.userid,u.orgid,u.firstname**

**from server s right join users u on s.serverid=u.userid;**

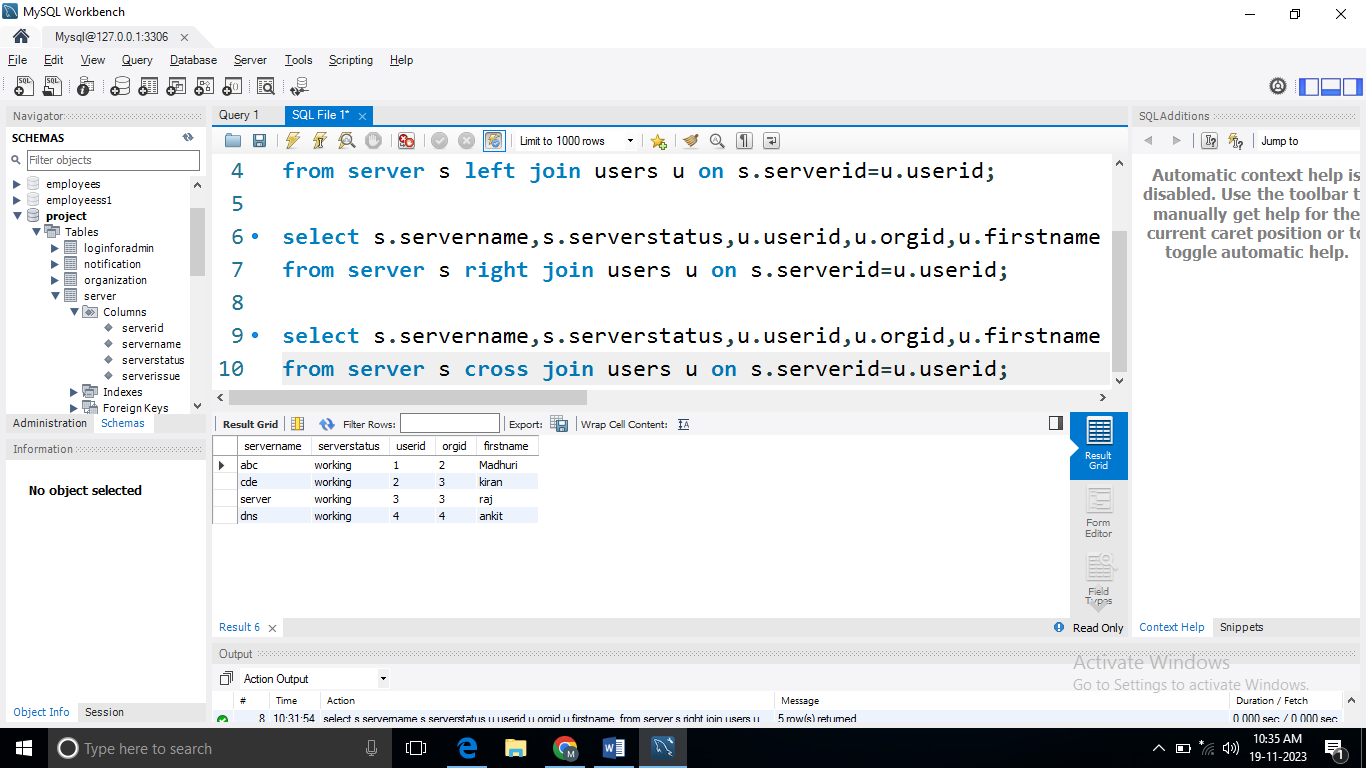


1. **CROSS JOIN:-**

**QUERY:-**

* **select s.servername,s.serverstatus,u.userid,u.orgid,u.firstname**

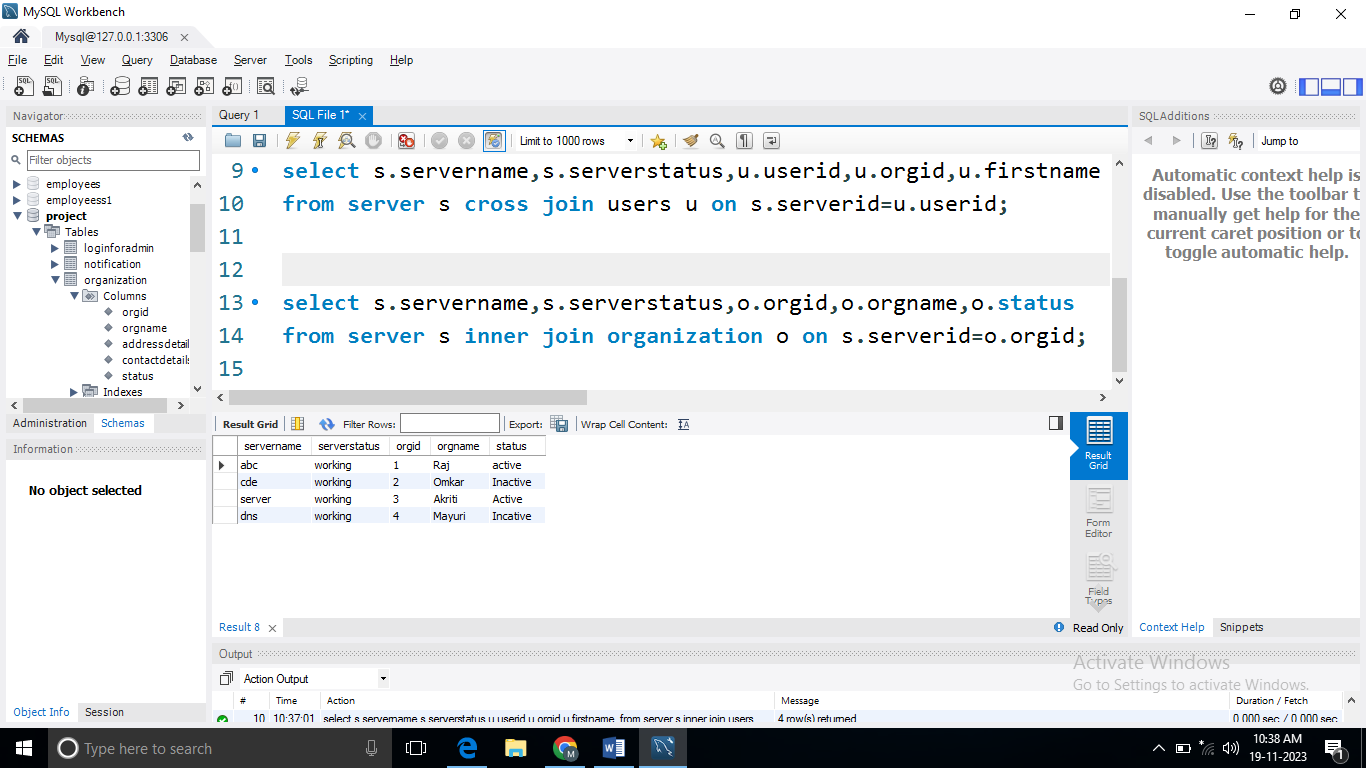
**from server s cross join users u on s.serverid=u.userid;**



1. **INNER JOIN:-**

**QUERY:-**

* **select s.servername,s.serverstatus,o.orgid,o.orgname,o.status**
* **from server s inner join organization o on s.serverid=o.orgid;**

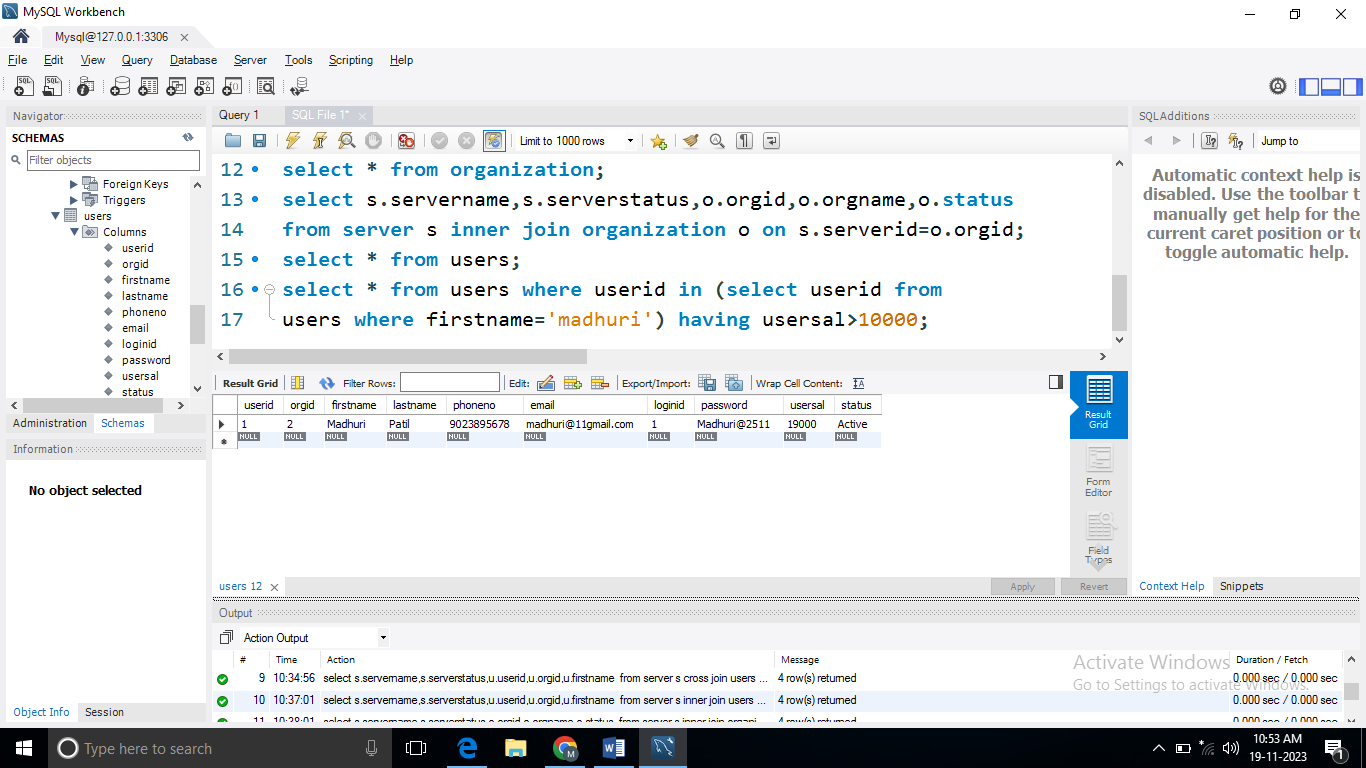


**SUBQUERIES**

1. **Show the all the details of all the users where username= ‘’Madhuri” having salary> 10000**

**QUERY:-**

* **select \* from users where userid in (select userid from**
* **users where firstname='madhuri') having usersal>10000;**

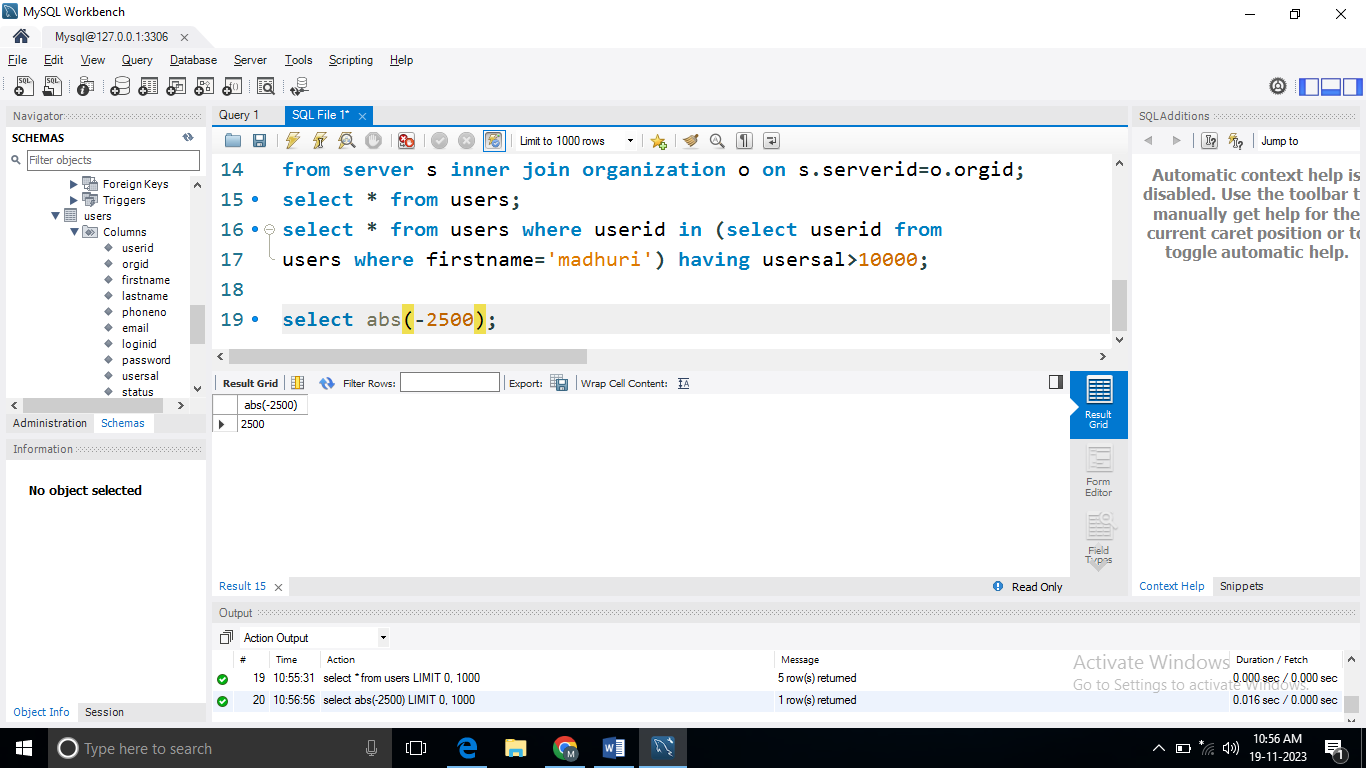


**BUILT IN FUNCTIONS**

1. **MATH**
2. **STRING**
3. **DATETIME**
4. **AGGREGATE**
5. **ABS:-**

**Query:-**

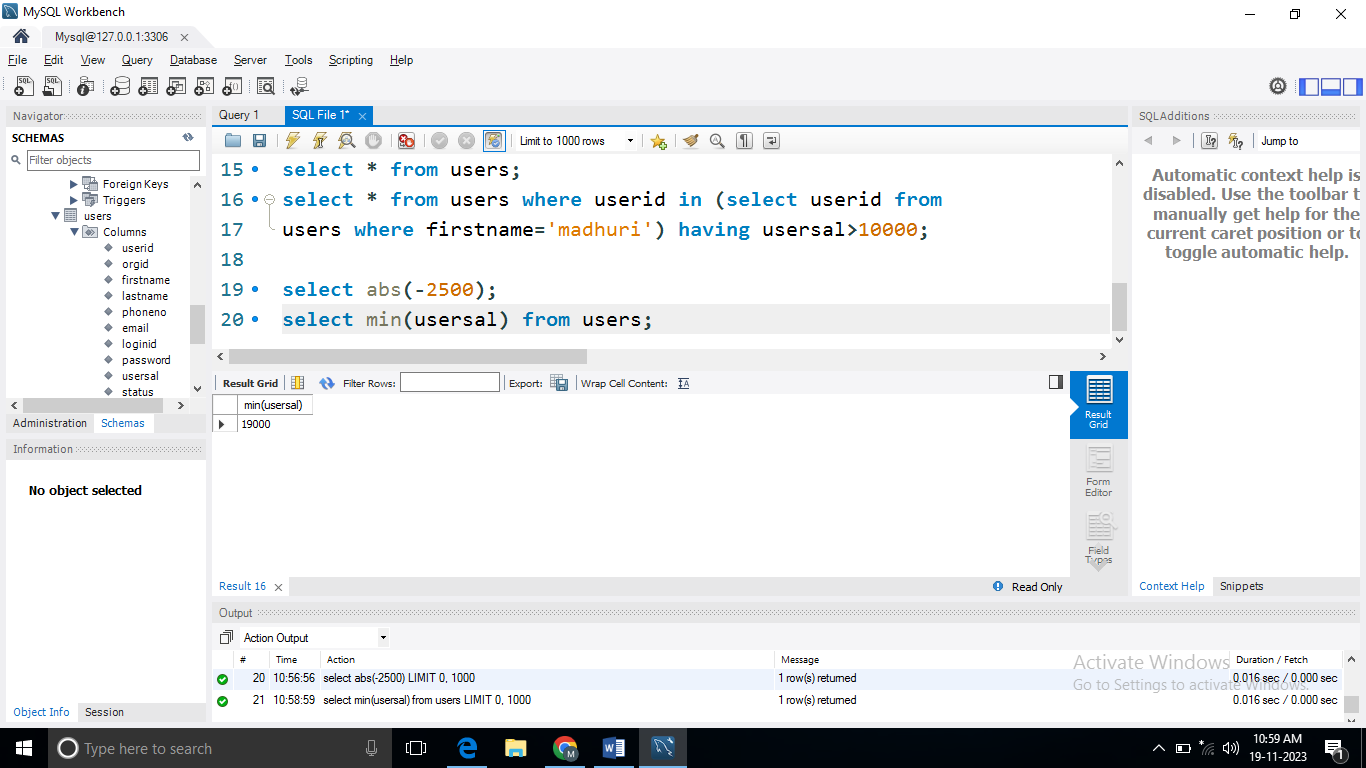
* select abs(-2500);



1. **MIN:=**

**Query:-**

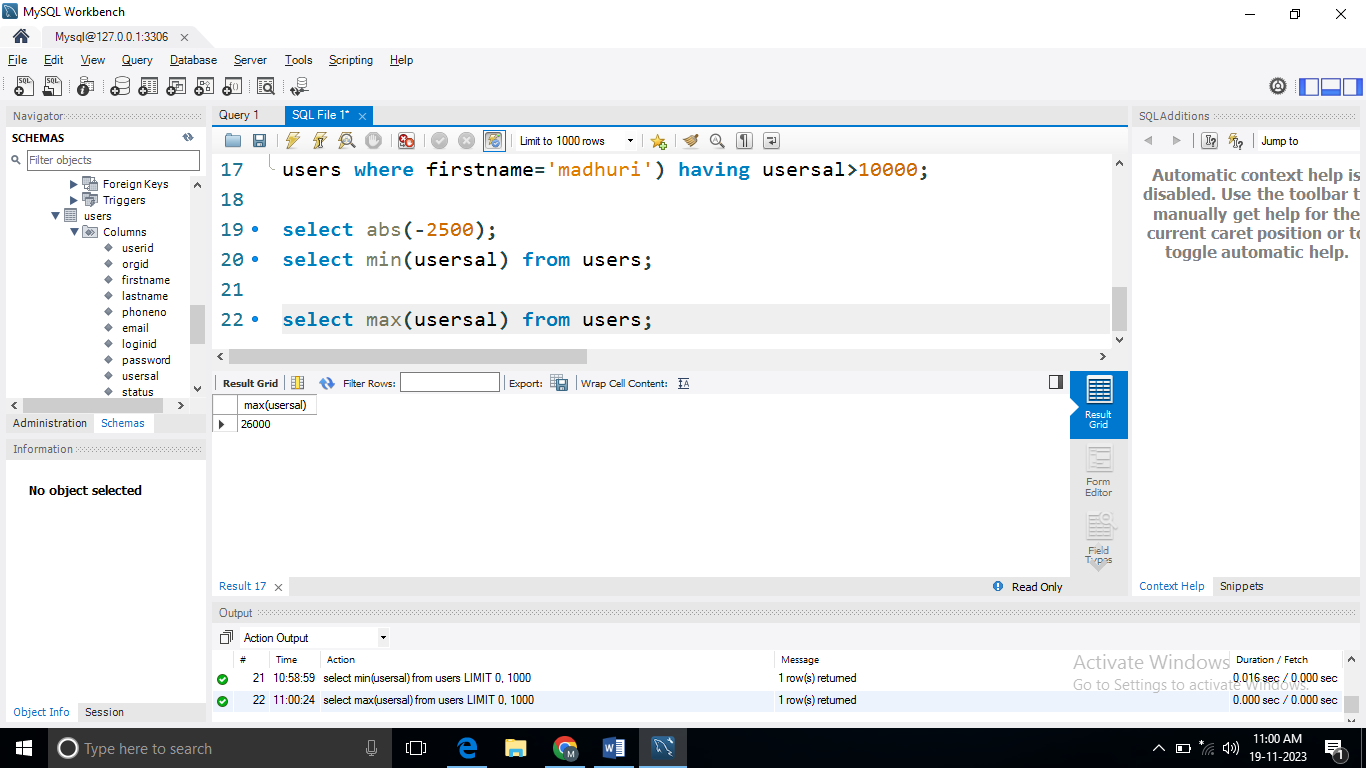
* select min(usersal) from users;



1. **MAX:=**

**Query:-**

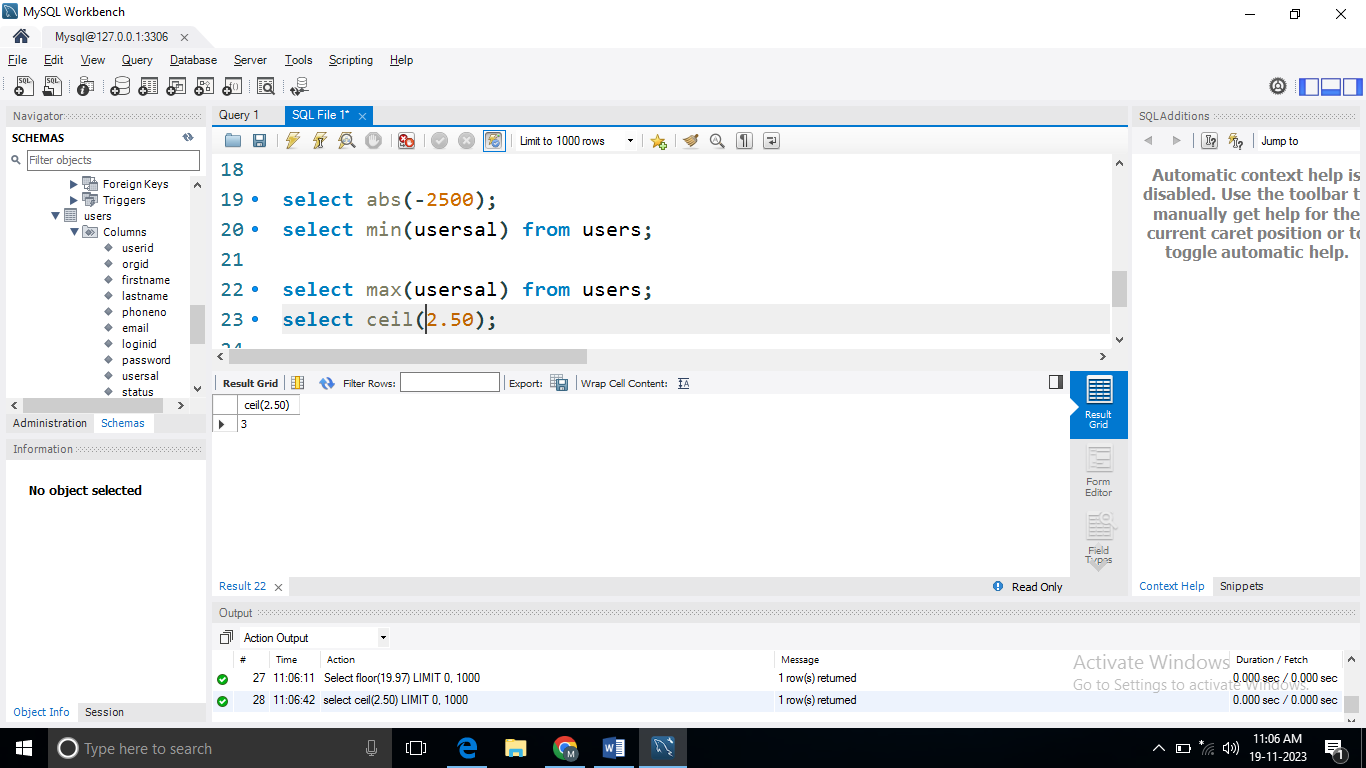
* select max(usersal) from users;



1. **CEIL:=**

**Query:-**

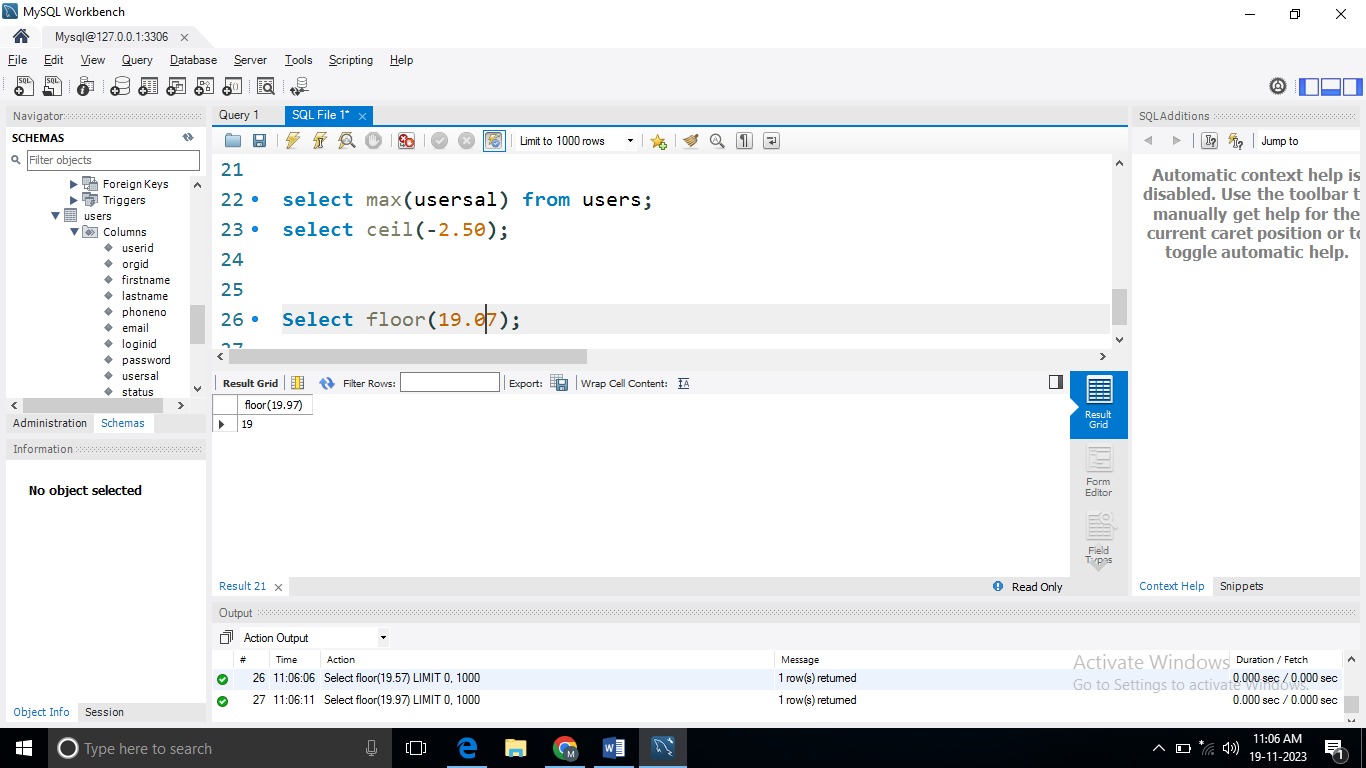
* select ceil(-2.50);



1. **FLOOR:=**

**Query:-**

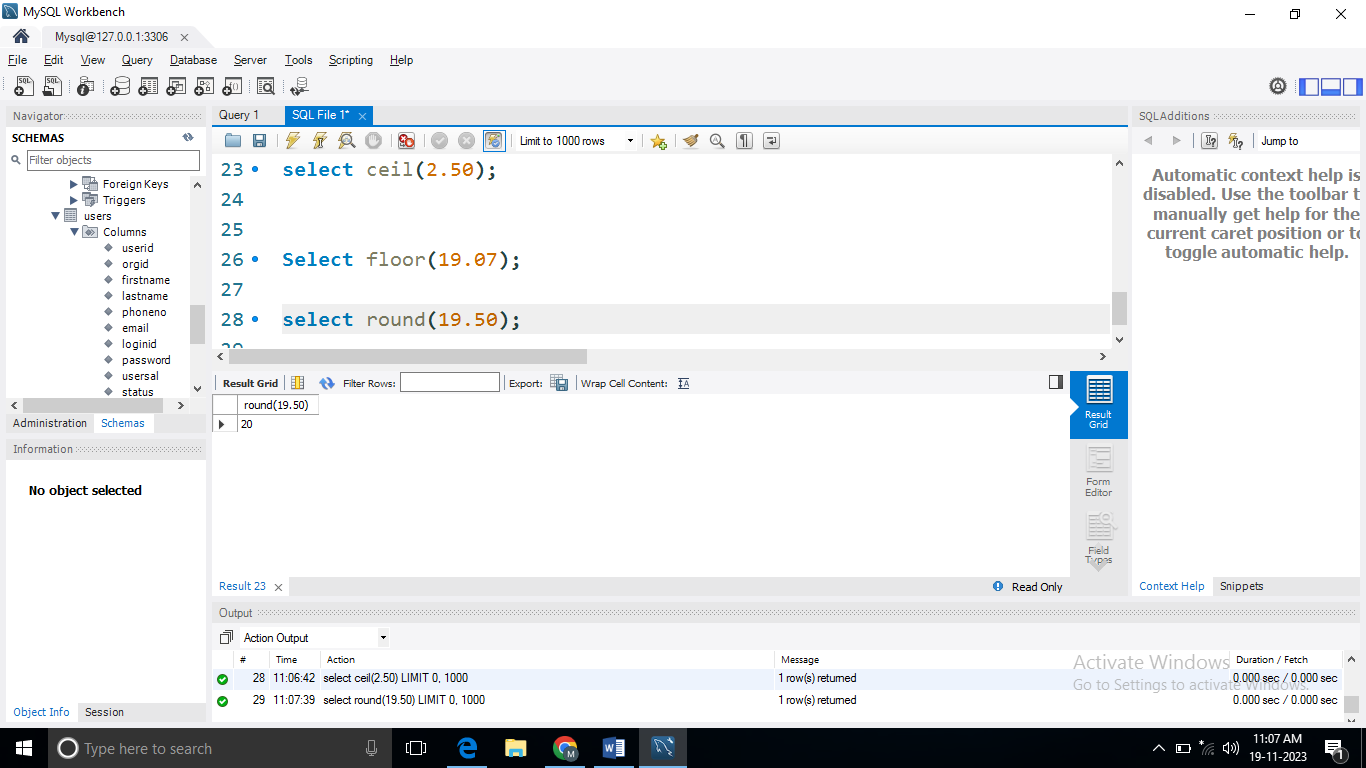
* Select floor(19.07);



**6.ROUND:=**

**Query:-**

* select round(19.50);

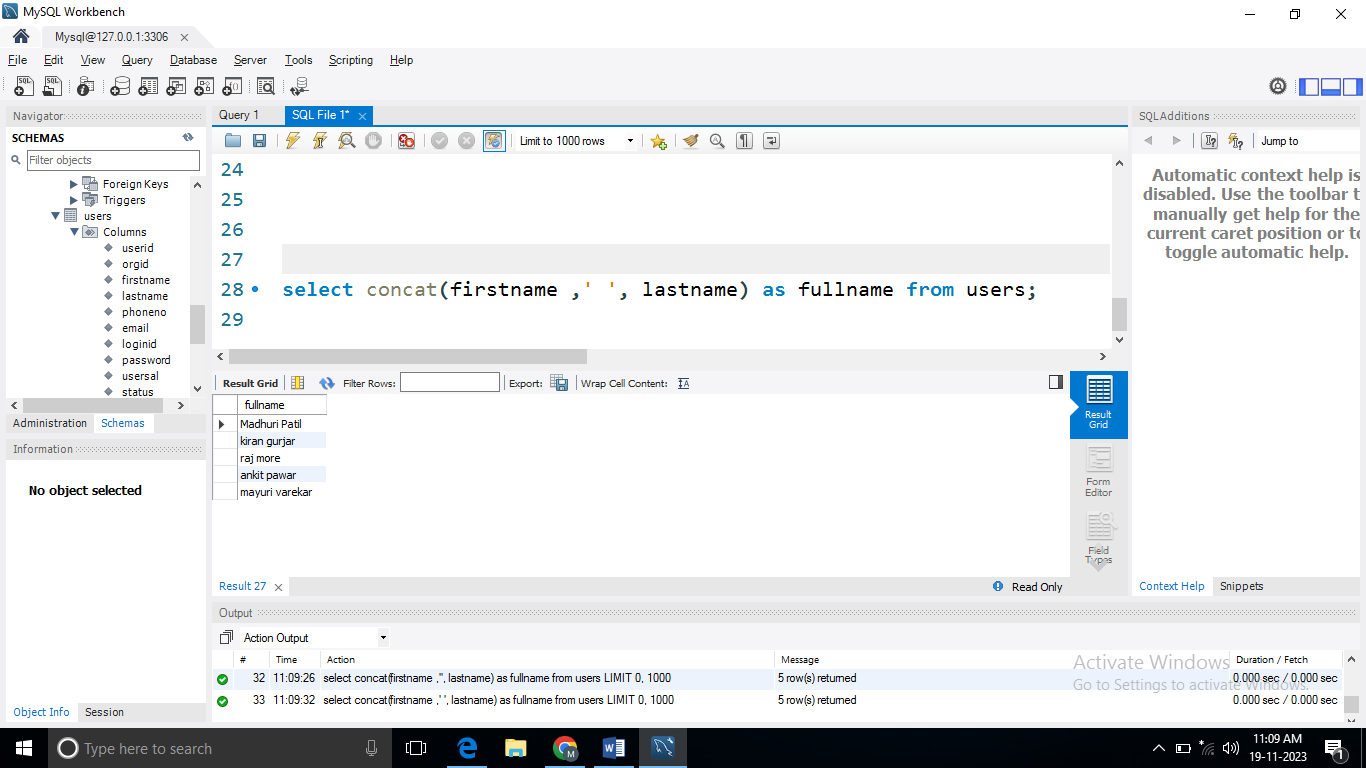


**String Function:-**

1. **CONCAT:-**

**Query:-**

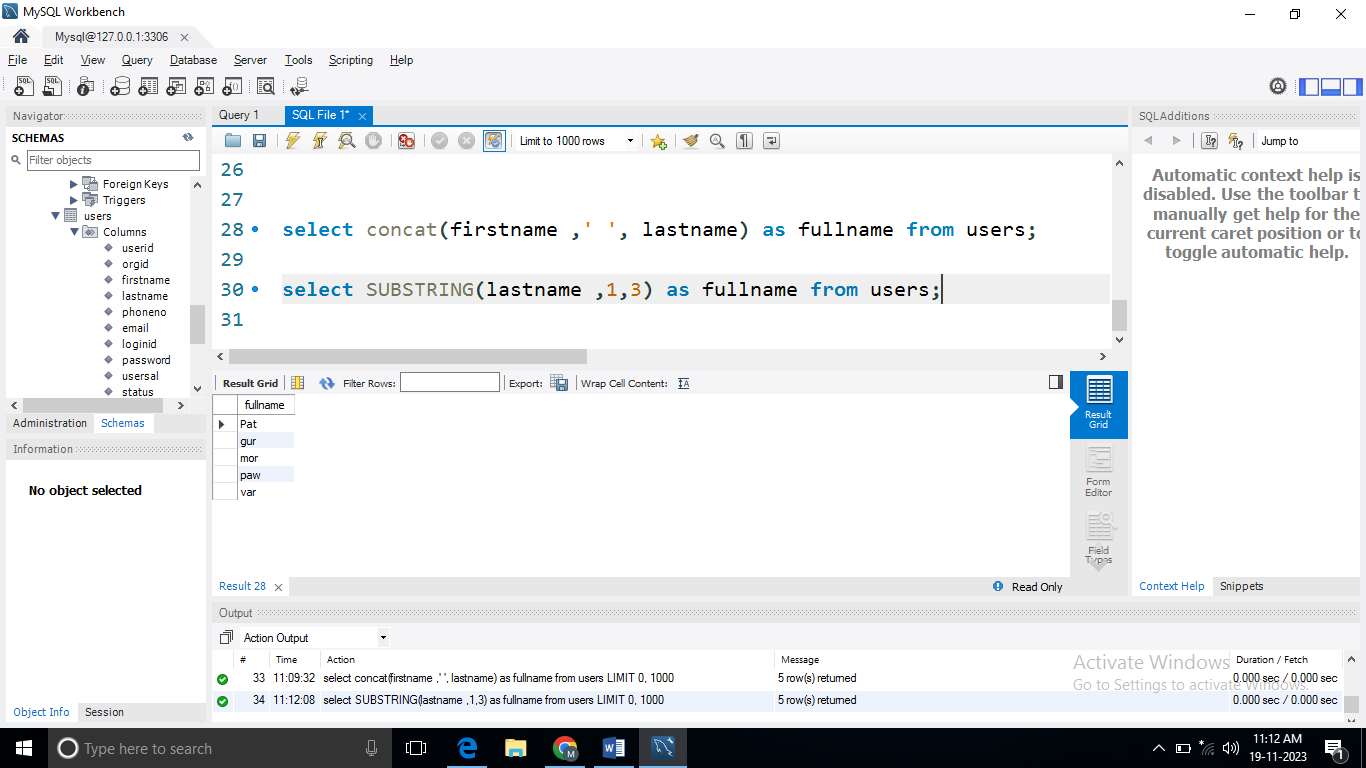
* select concat(firstname ,' ', lastname) as fullname from users;



1. **SUBSTRING:-**

**Query:-**

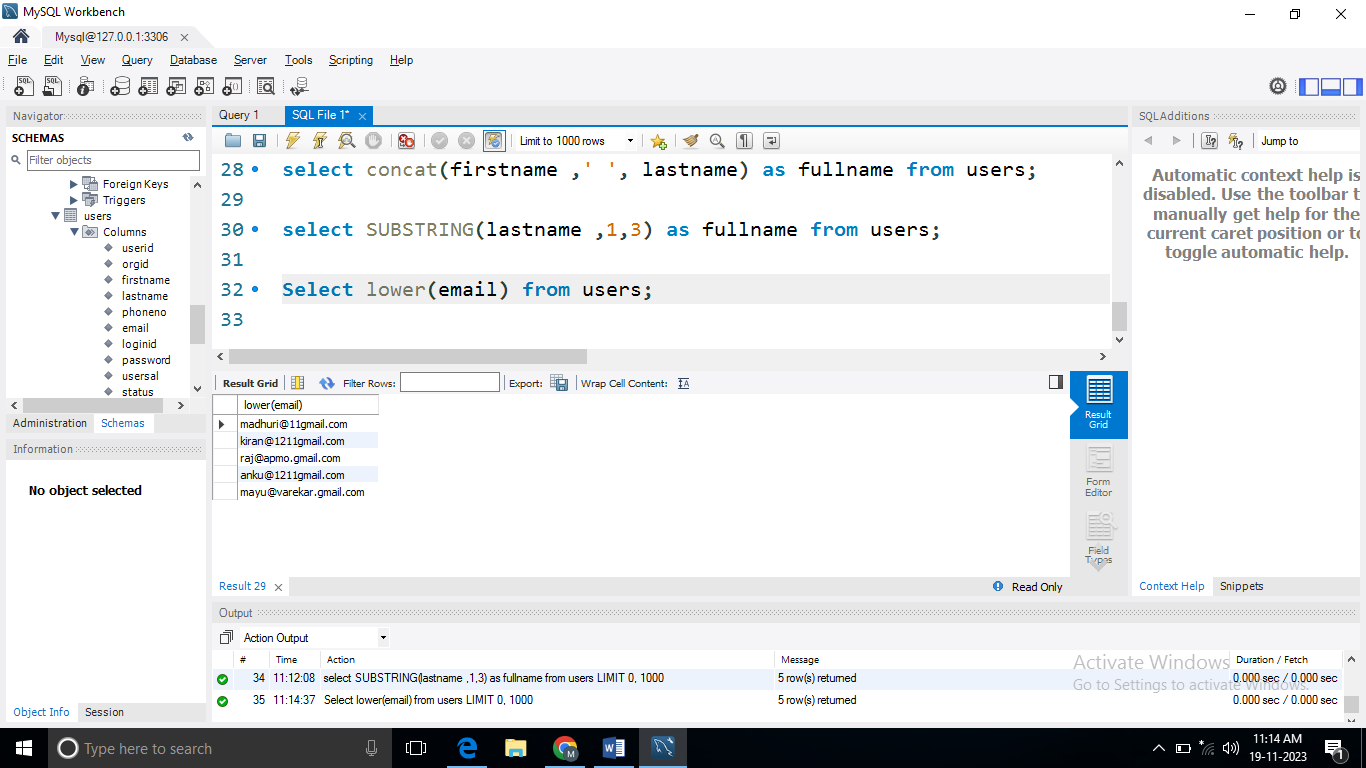
* select SUBSTRING(lastname ,1,3) as fullname from users;



1. **LOWER:-**

**Query:-**

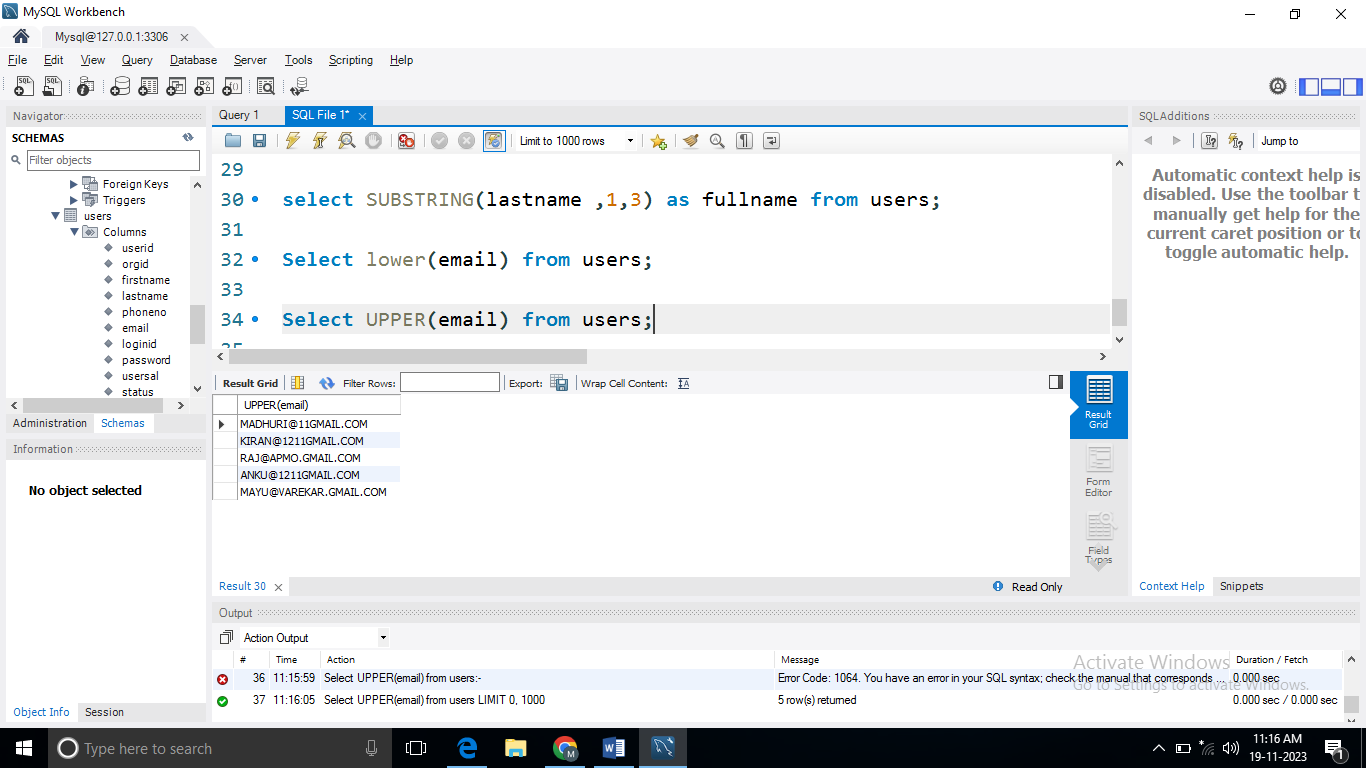
* Select lower(email) from users;



1. **UPPER:-**

**Query:-**

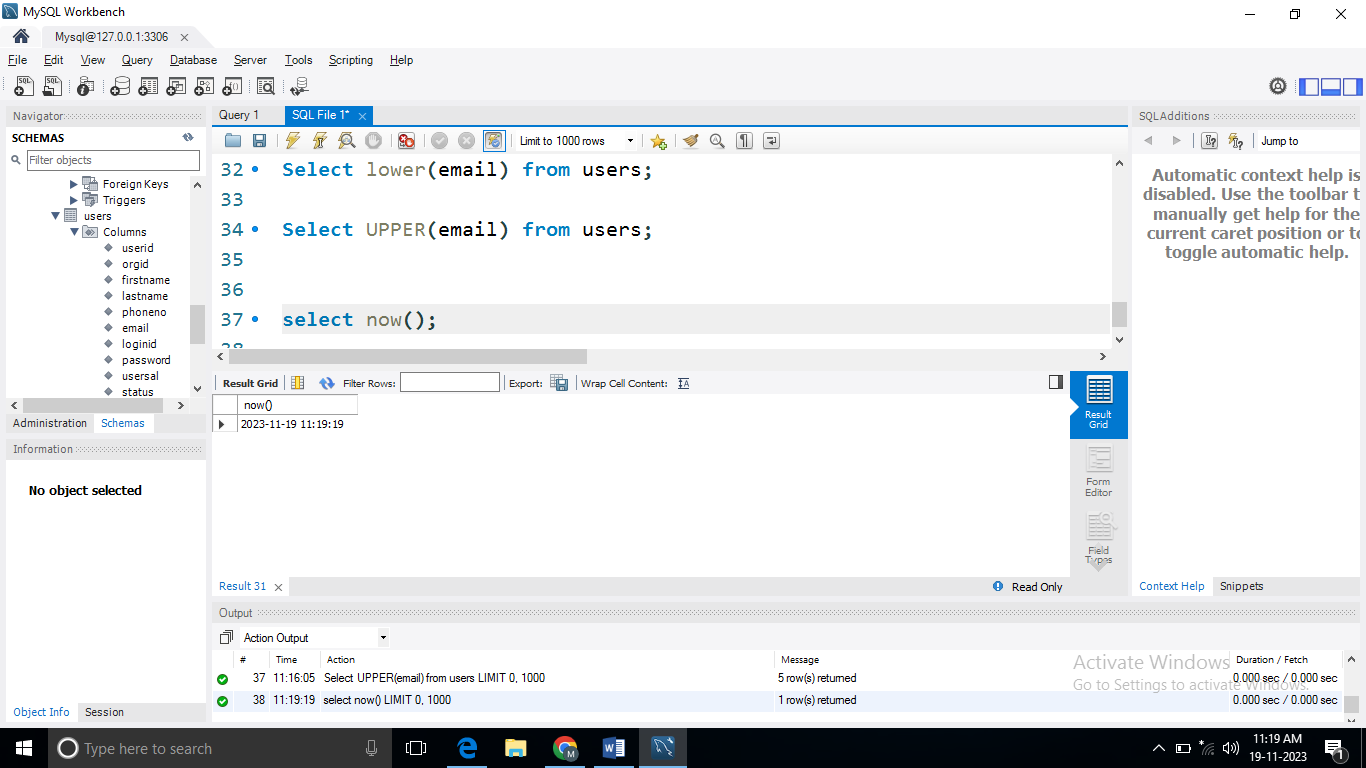
* Select UPPER(email) from users:-



1. **NOW():=**

**Query:-**

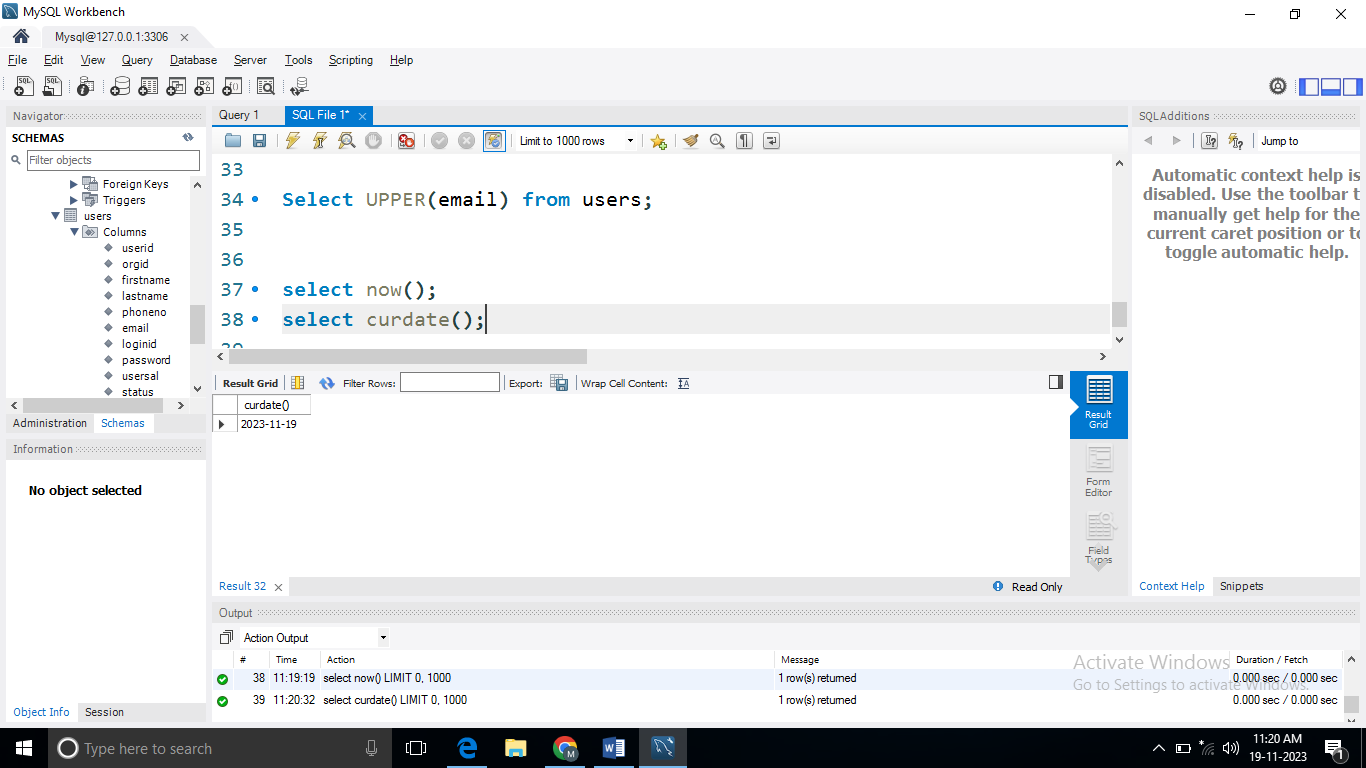
* select now();



1. **CURDATE():=**

**Query:-**

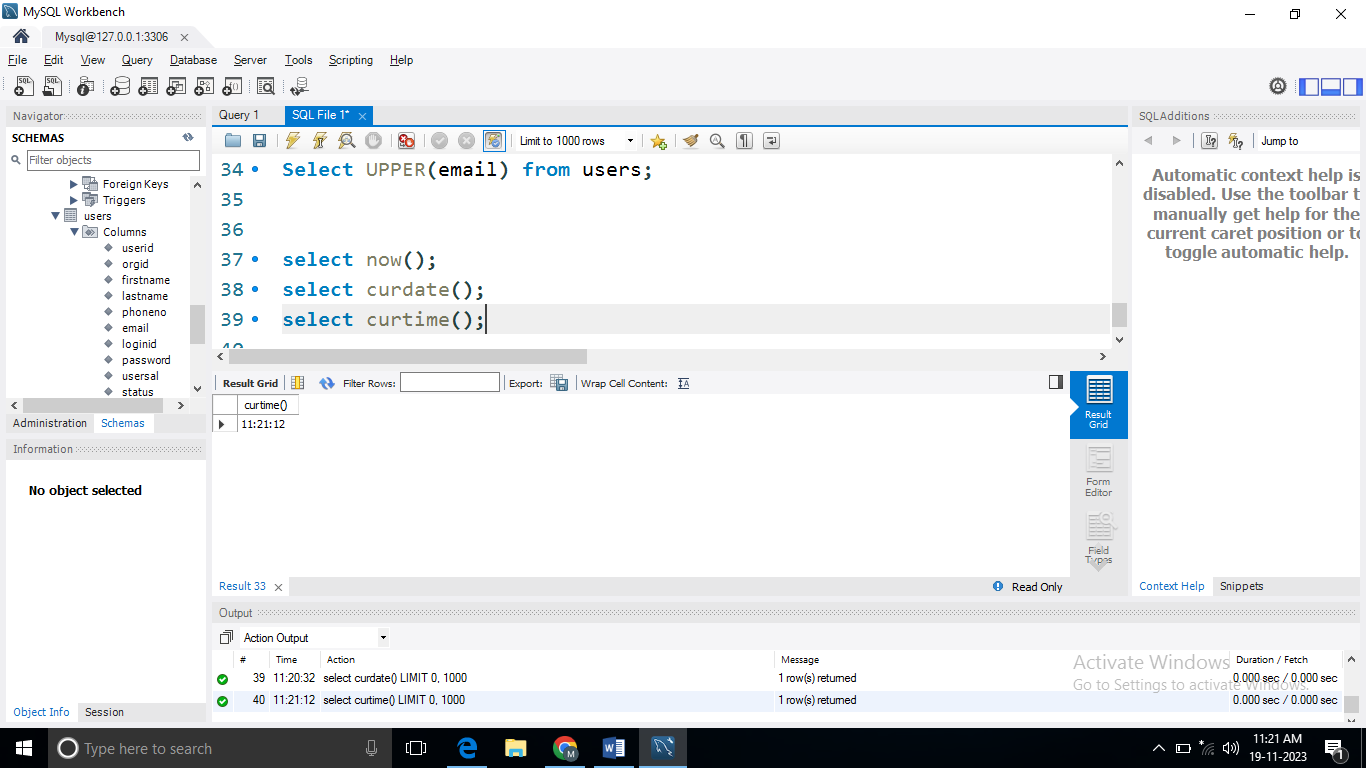
* select curdate();



1. **CURTIME():=**

**Query:-**

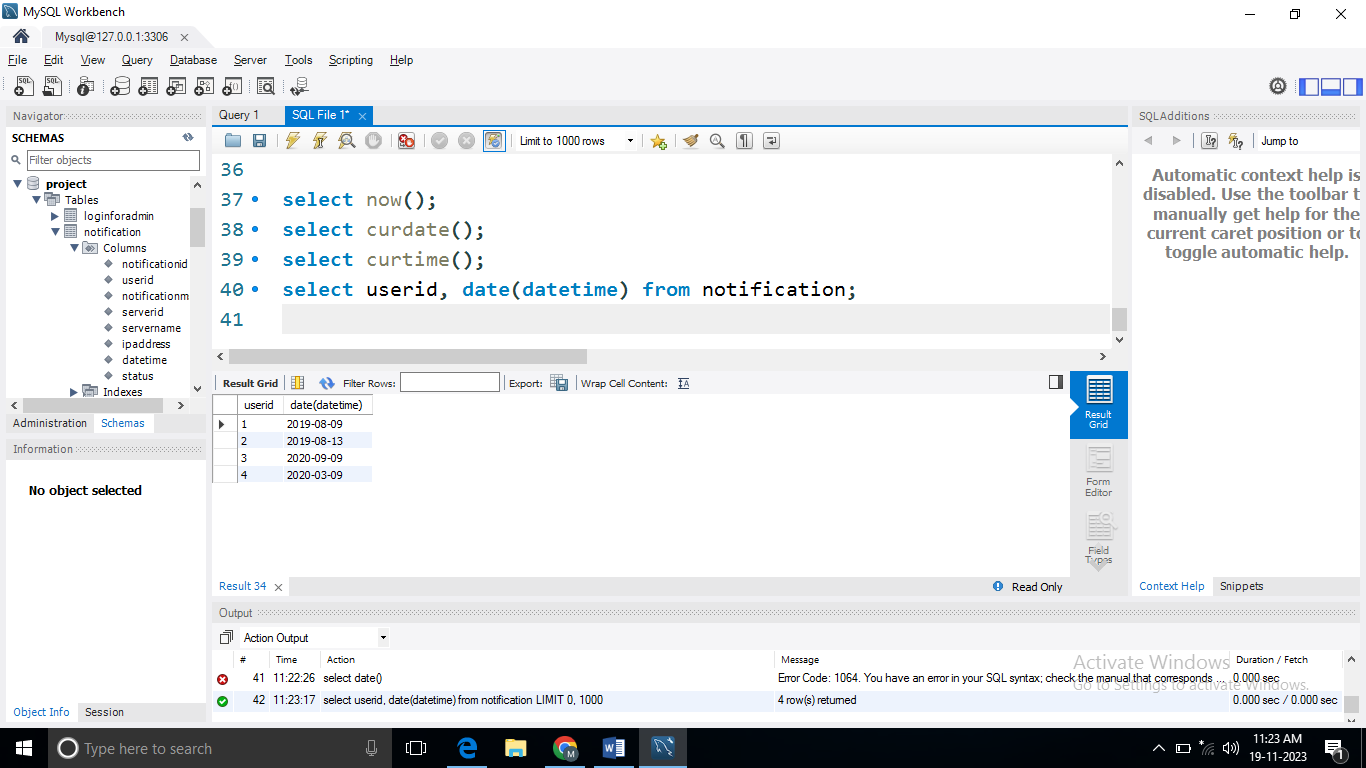
* select curtime();



1. **DATE():=**

**Query:-**

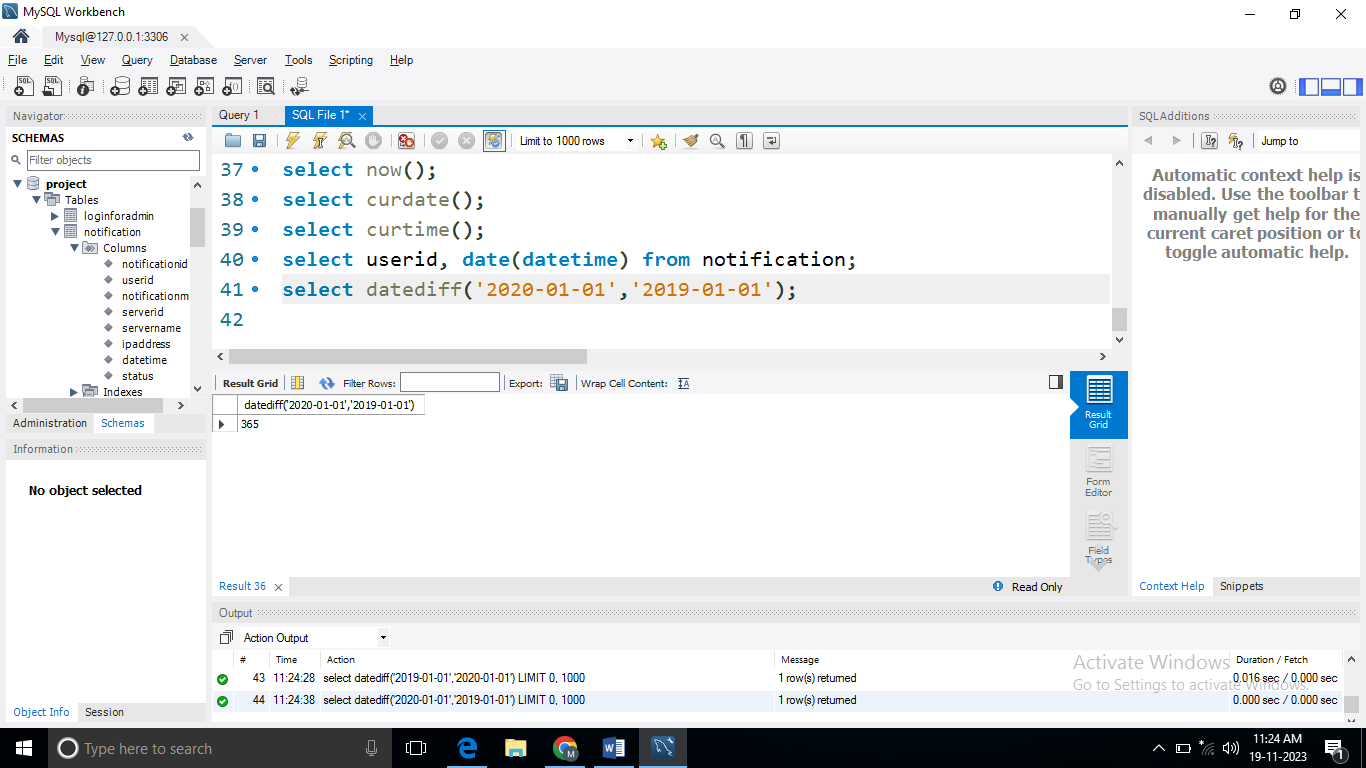
* select userid, date(datetime) from notification;



1. **DATEDIFF():=**

**Query:-**

* select datediff('2020-01-01','2019-01-01');

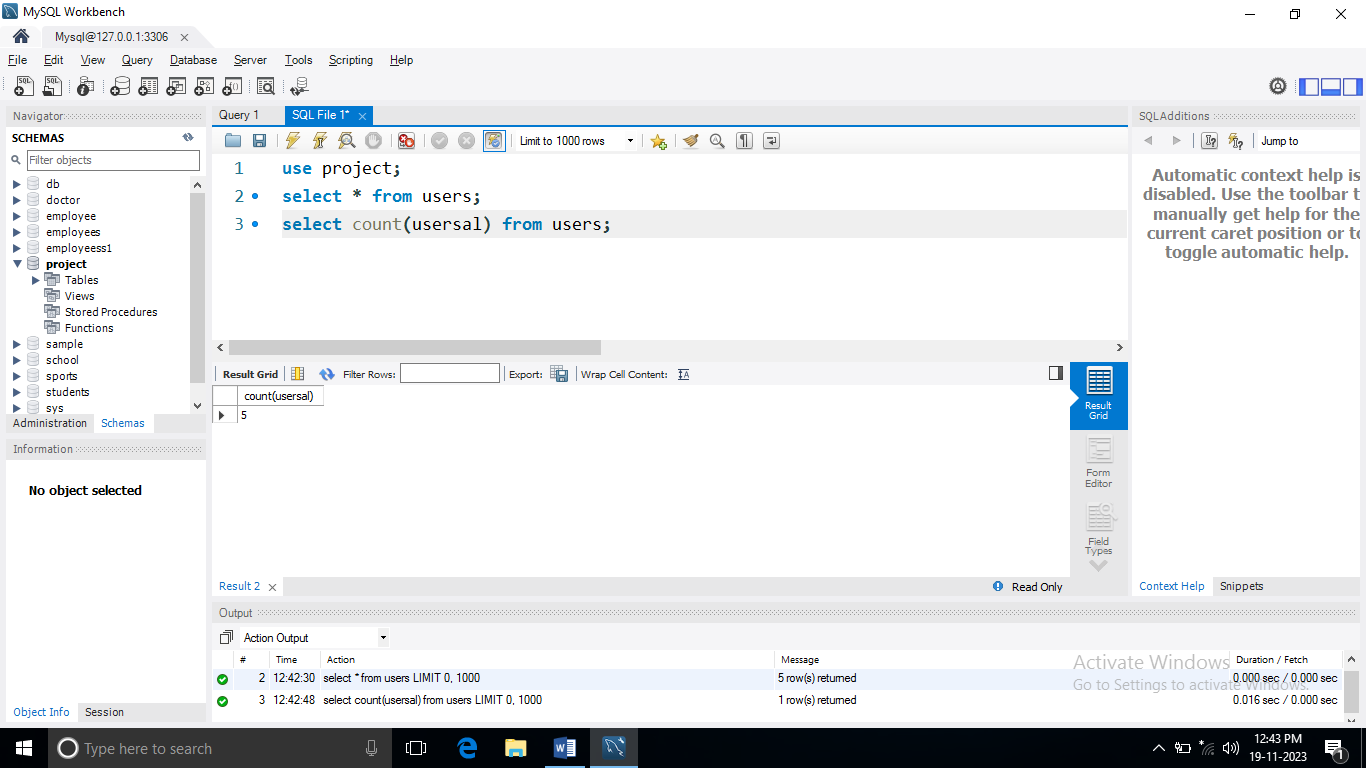


**Aggregate Functions**

1. **Count():-**

**QUERY:-**

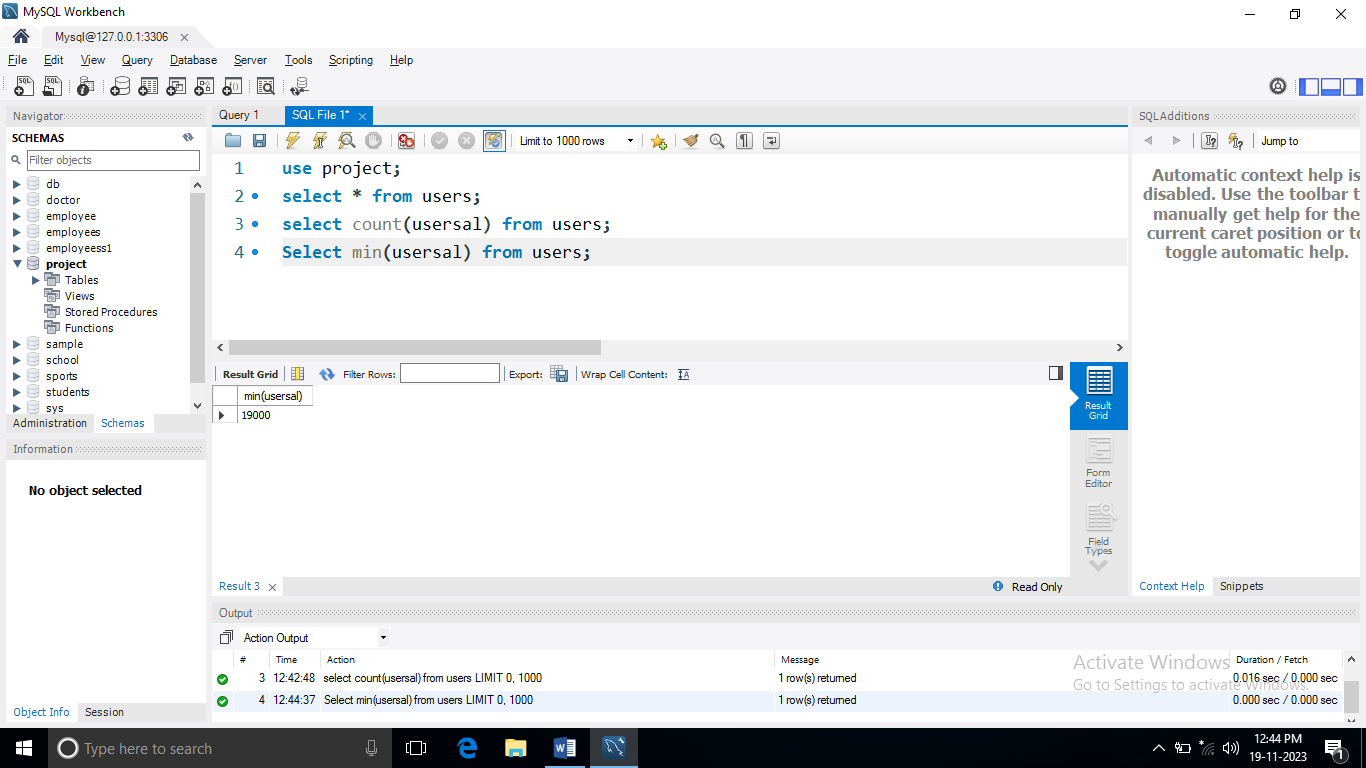
* Select count (usersal) from users;



1. **Min():=**

**Query:-**

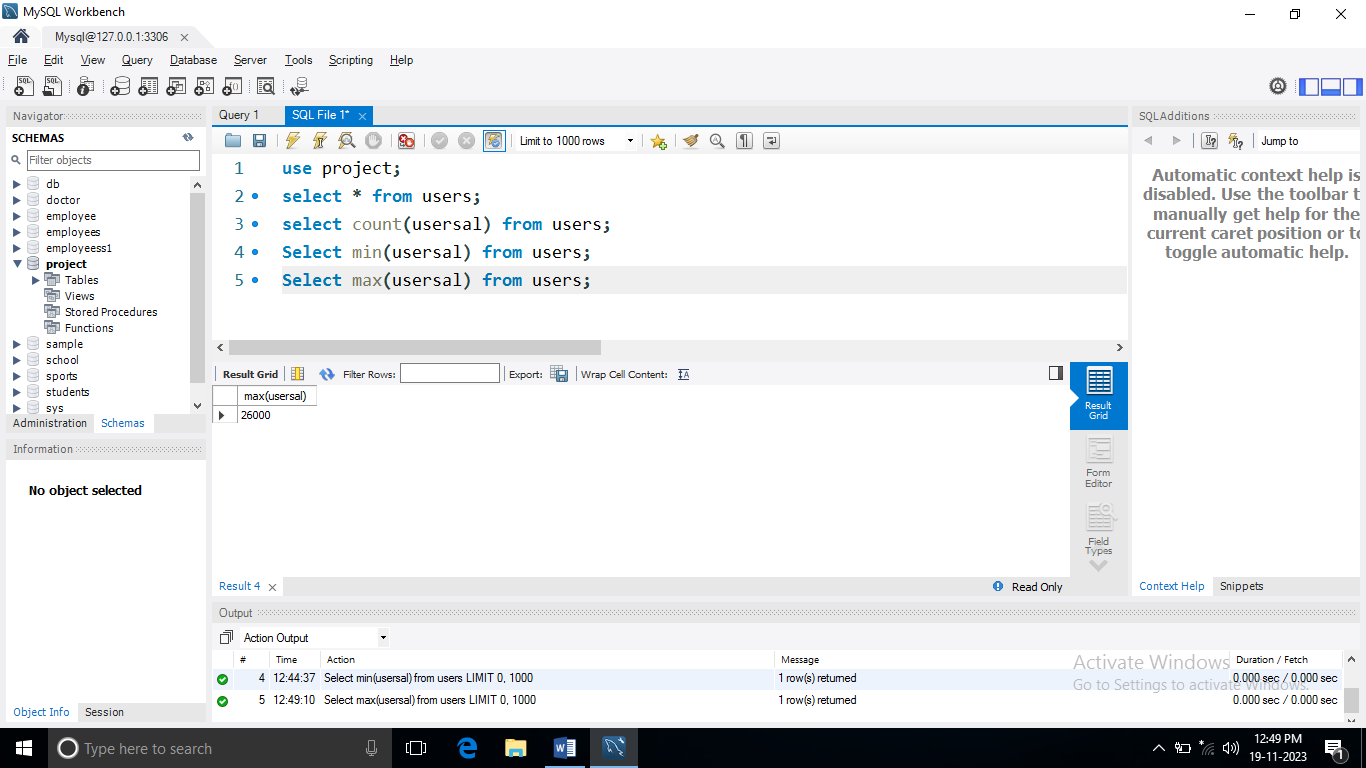
* Select min(usersal) from users;



1. **Max():=**

**Query:-**

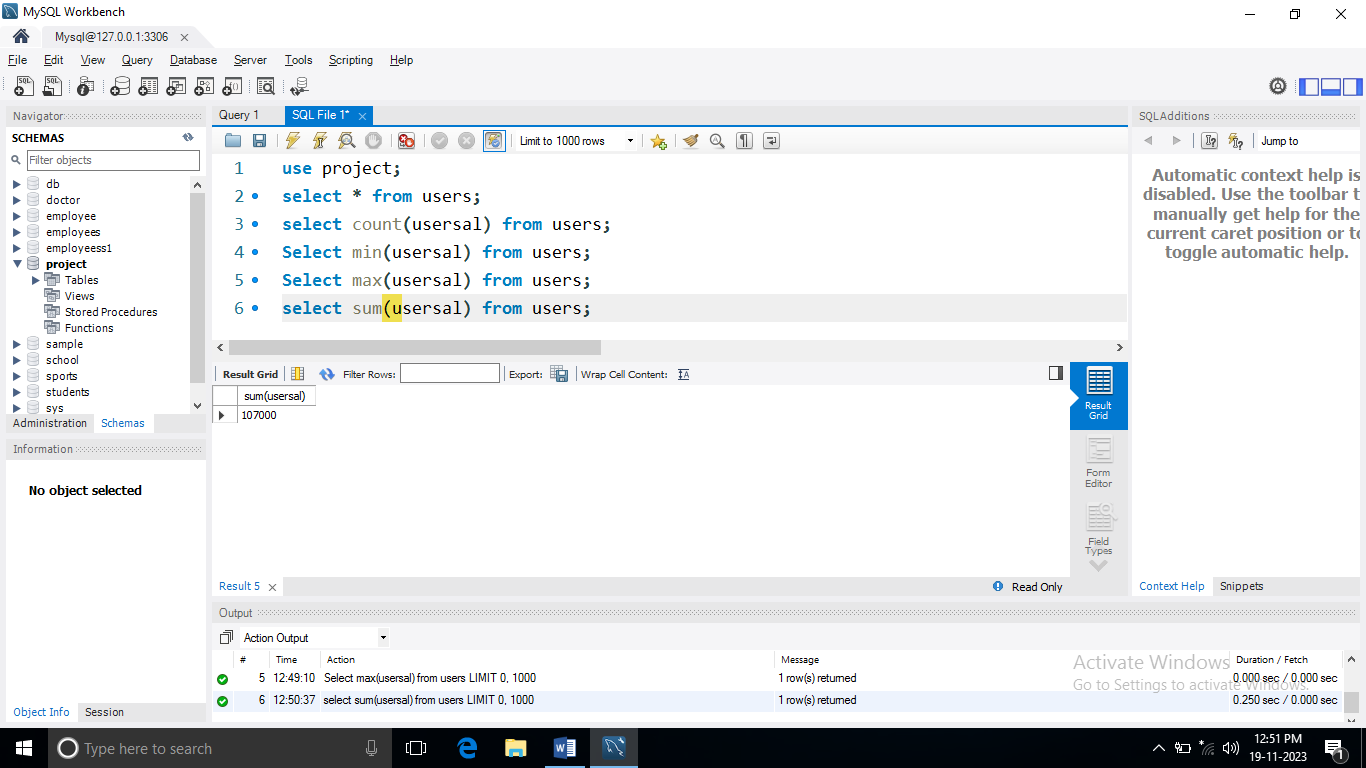
* Select max(usersal) from users;



1. **Sum():=**

**Query:-**

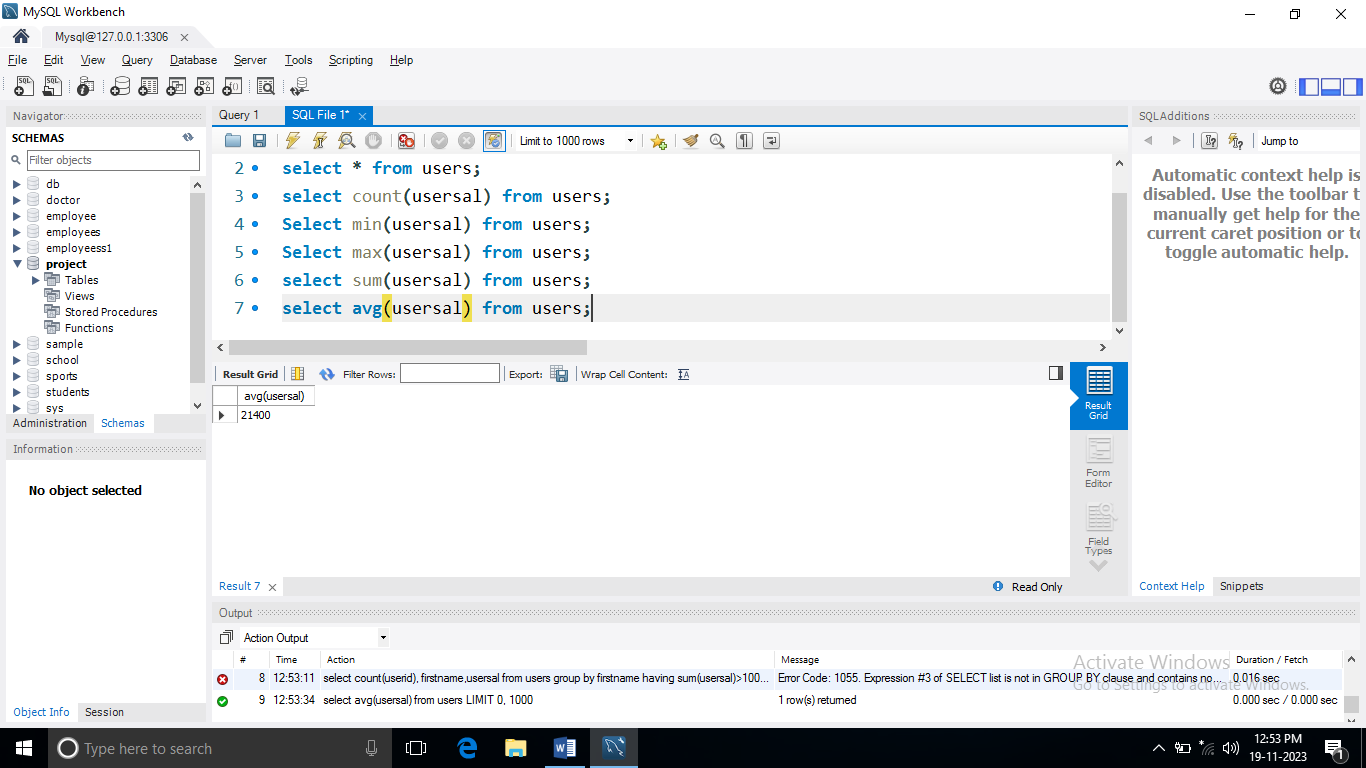
* Select sum(usersal) from users;



1. **Avg():=**

**Query:-**

* Select avg(usersal) from users;



**THANK YOU**