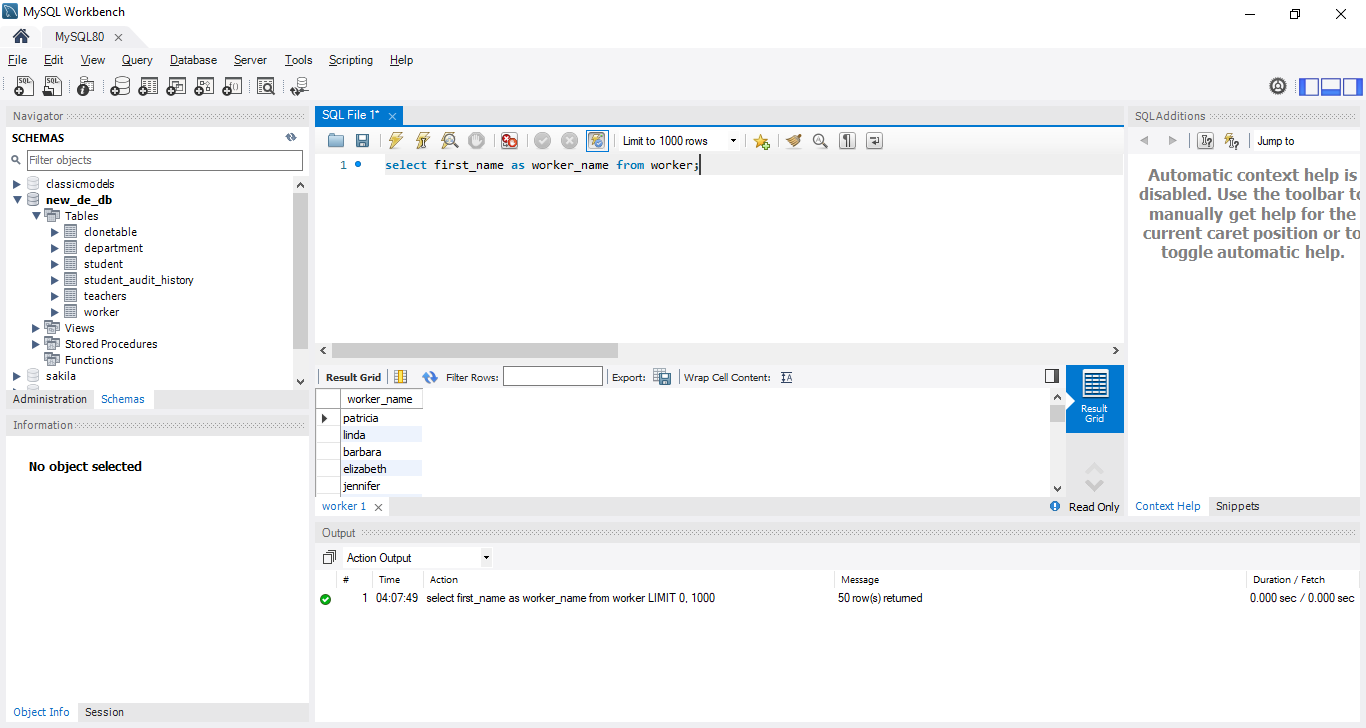
# **Task-1**

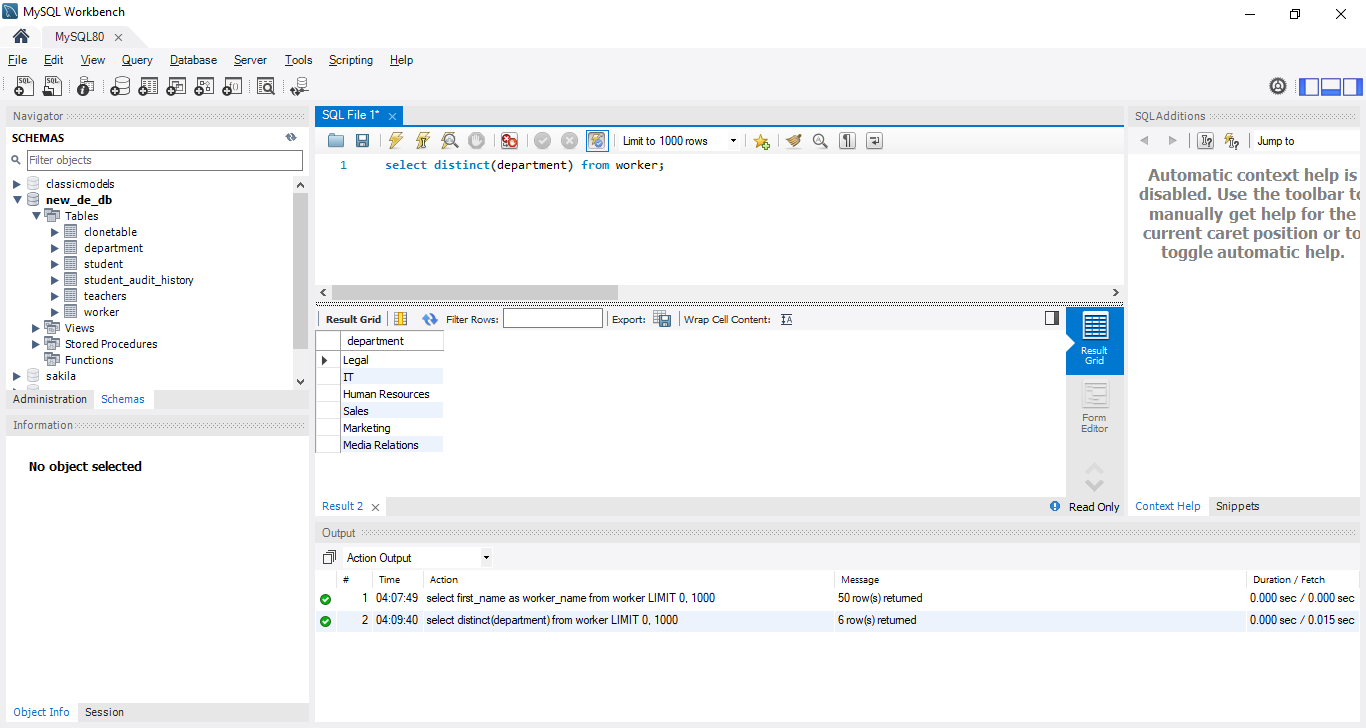
1. Write an SQL query to fetch “FIRST\_NAME” from the Worker table using the alias name as <WORKER\_NAME>.

**Query – select first\_name as worker\_name from worker;**



1. Write an SQL query to fetch unique values of DEPARTMENT from the Worker table.

**Query - select distinct(department) from worker;**



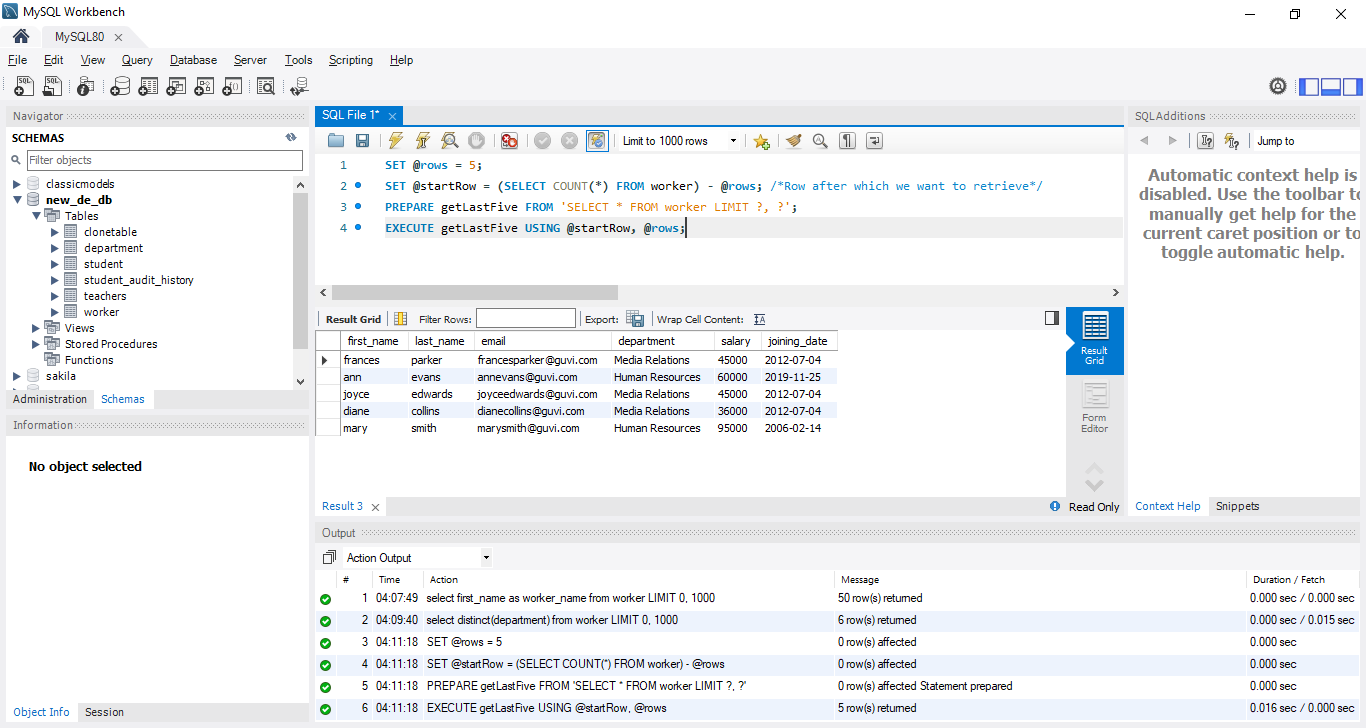
1. Write an SQL query to show the last 5 records from a table.

**Query - SET @rows = 5;**

**SET @startRow = (SELECT COUNT(\*) FROM worker) - @rows; /\*Row after which we want to retrieve\*/**

**PREPARE getLastFive FROM 'SELECT \* FROM worker LIMIT ?, ?';**

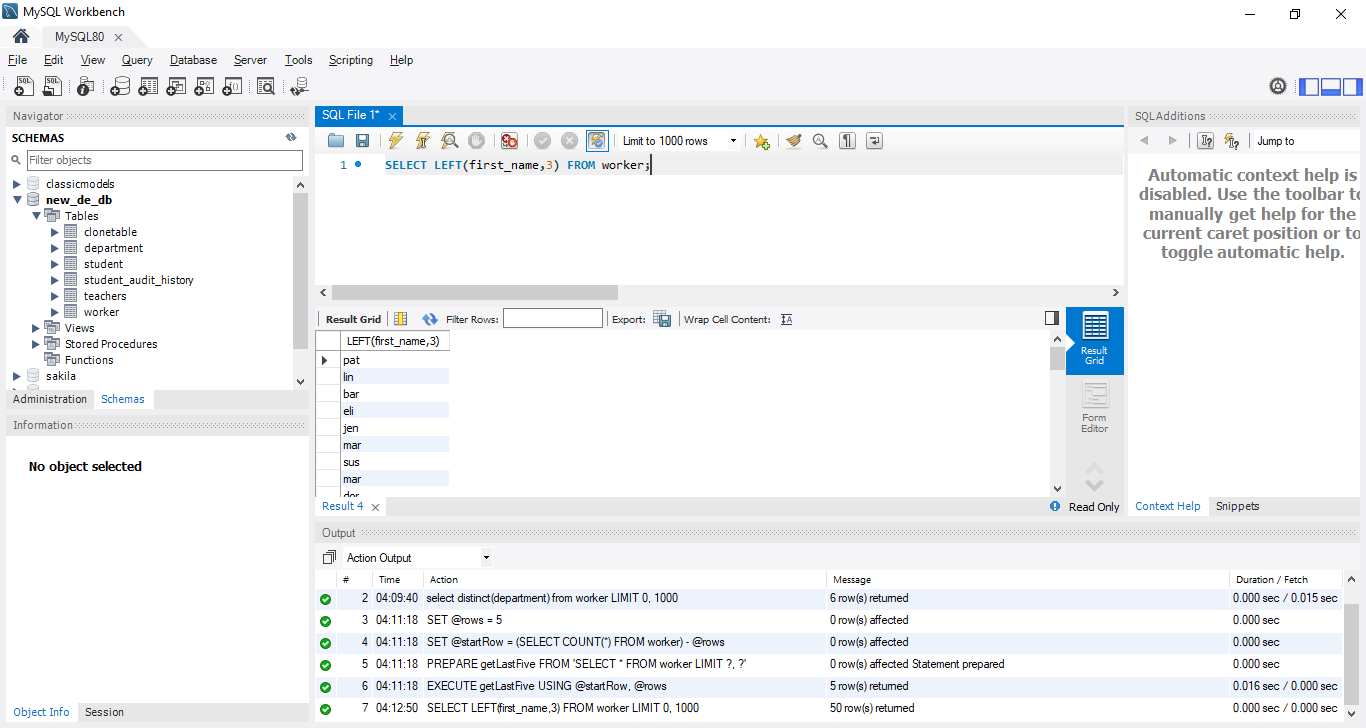
**EXECUTE getLastFive USING @startRow, @rows;**



# **Task-2**

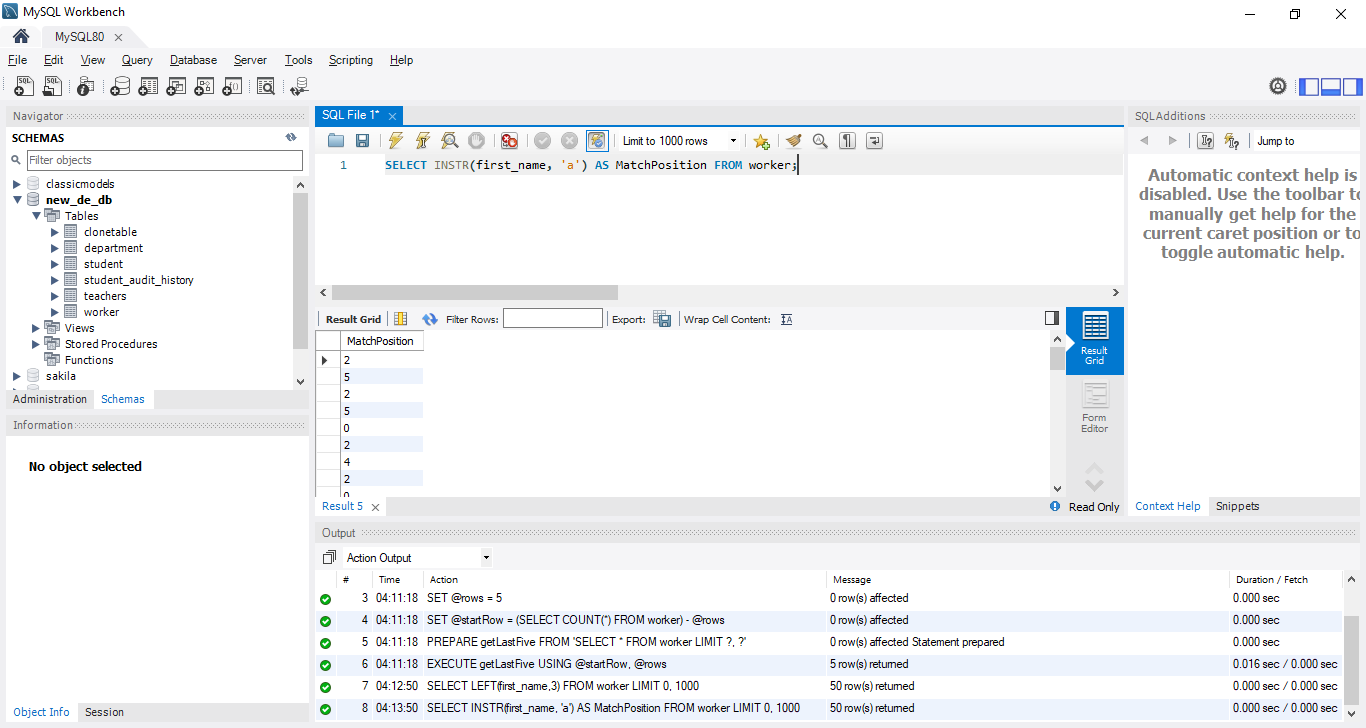
1. Write an SQL query to print the first three characters of FIRST\_NAME from Worker

**Query - SELECT LEFT(first\_name,3) FROM worker;**



1. Write an SQL query to find the position of the alphabet (‘a’) in the first name

**Query – SELECT INSTR(first\_name, 'a') AS MatchPosition FROM worker;**



1. Write an SQL query to print the name of employees who have the highest salary in each department

**Query – select concat(worker.first\_name,' ', worker.last\_name) AS worker\_name,worker.department,worker.salary**

**FROM worker,**

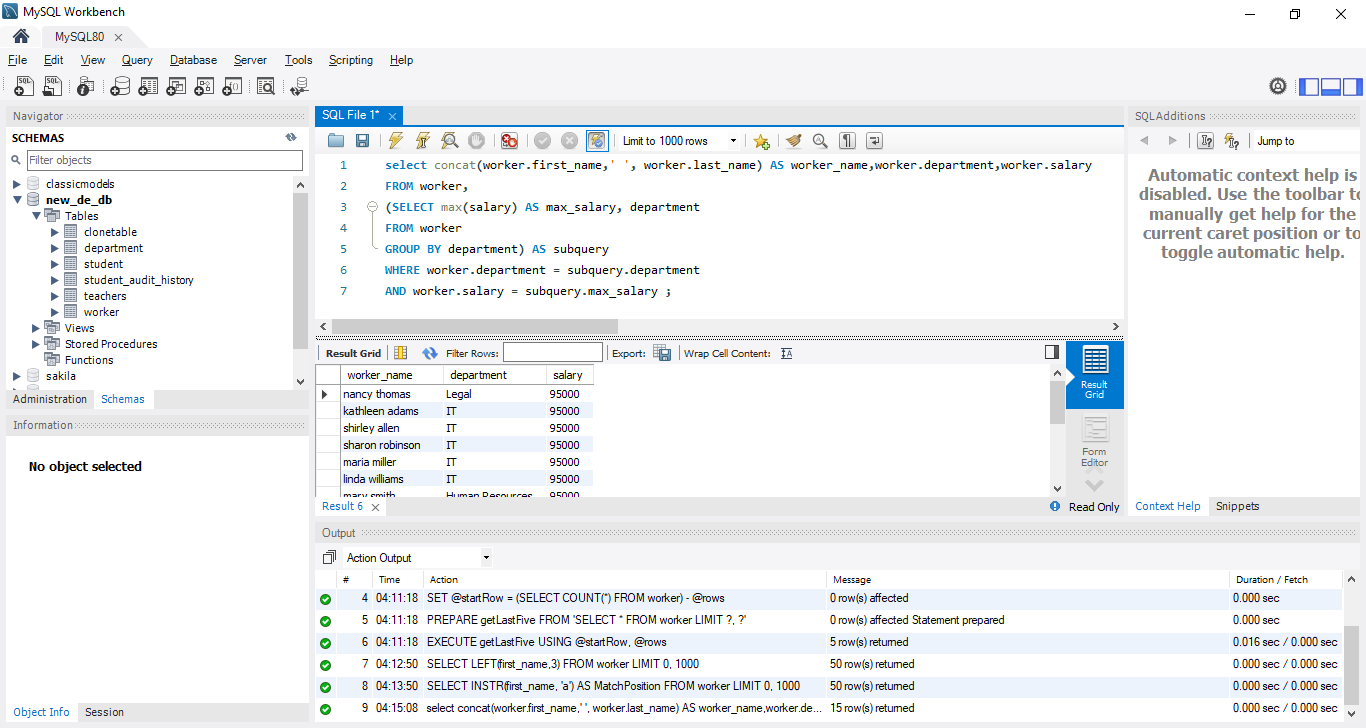
**(SELECT max(salary) AS max\_salary, department**

**FROM worker**

**GROUP BY department) AS subquery**

**WHERE worker.department = subquery.department**

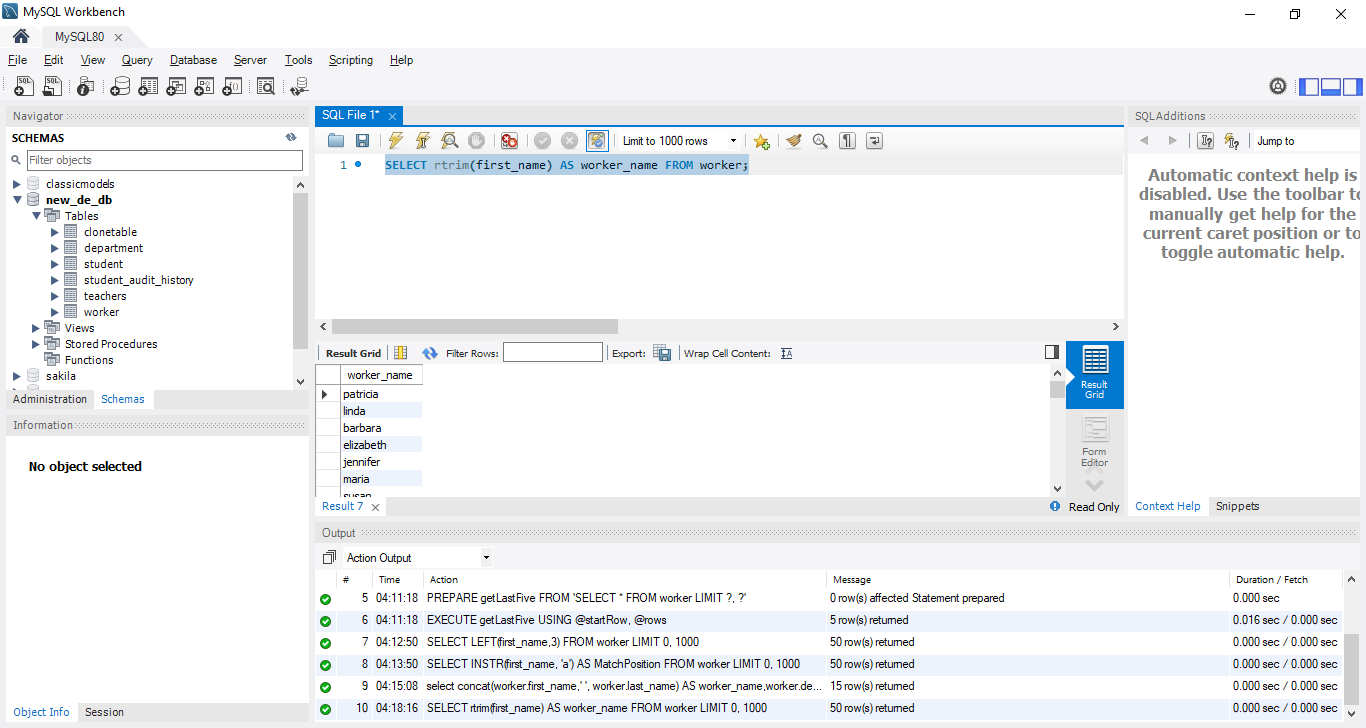
**AND worker.salary = subquery.max\_salary ;**



# **Task-3**

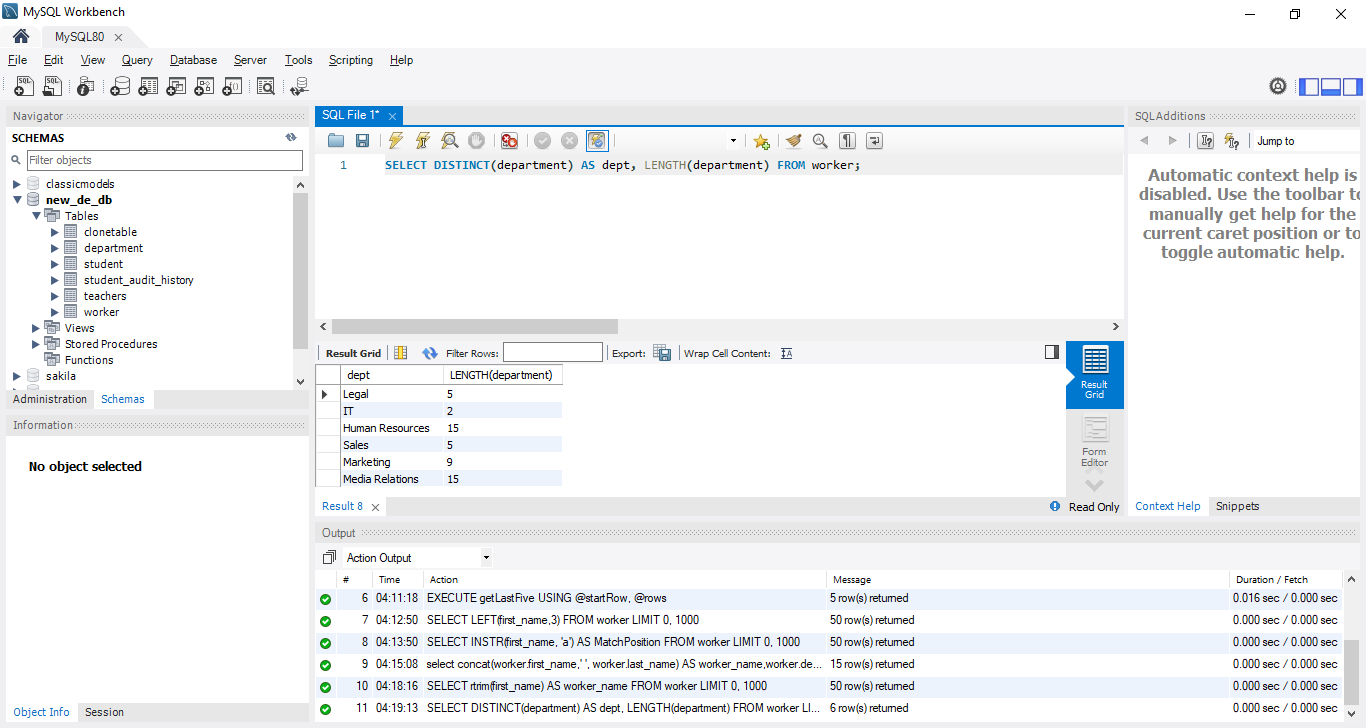
1. Write an SQL query to print the FIRST\_NAME from the Worker table after removing white spaces from the right side.

**Query - SELECT rtrim(first\_name) AS worker\_name FROM worker;**



1. Write an SQL query that fetches the unique values of DEPARTMENT from the Worker table and prints its length.

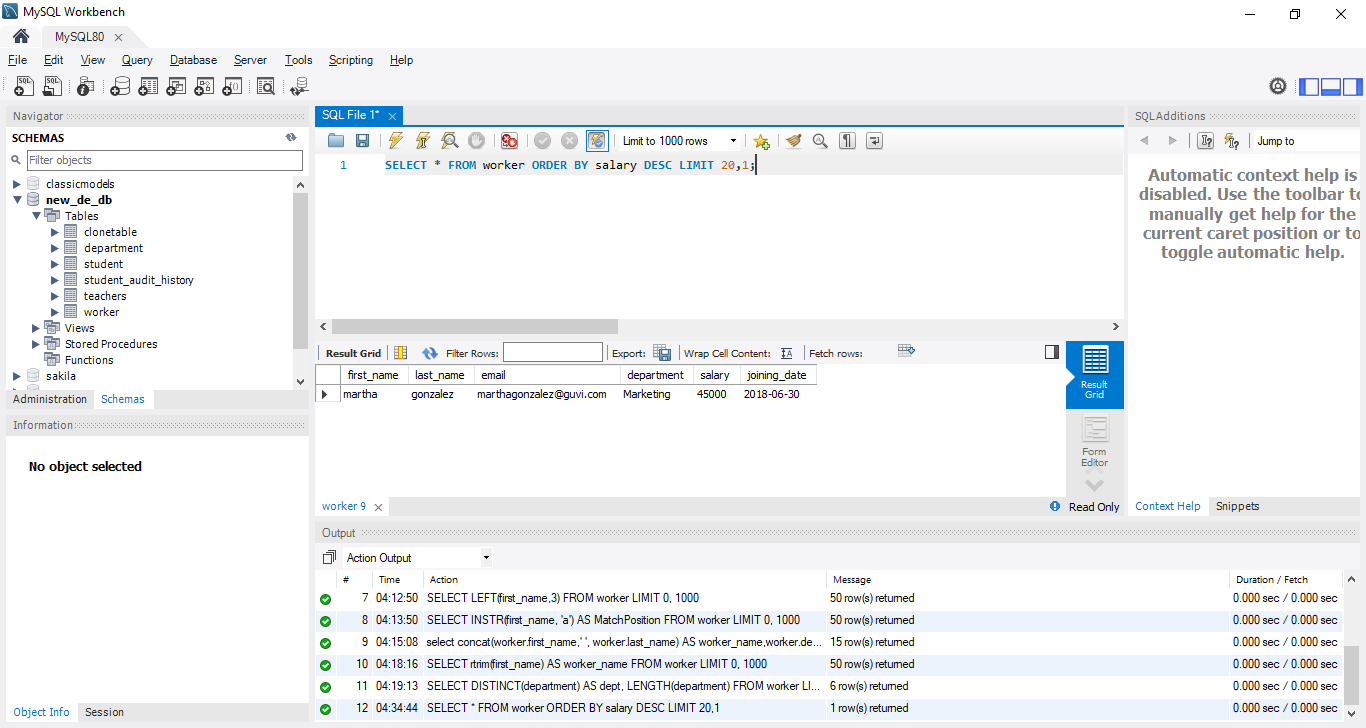
**Query - SELECT DISTINCT(department) AS dept, LENGTH(department) FROM worker;**



1. Write an SQL query to fetch nth max salaries from a table.

**Query - SELECT \* FROM worker ORDER BY salary DESC LIMIT 20,1;**

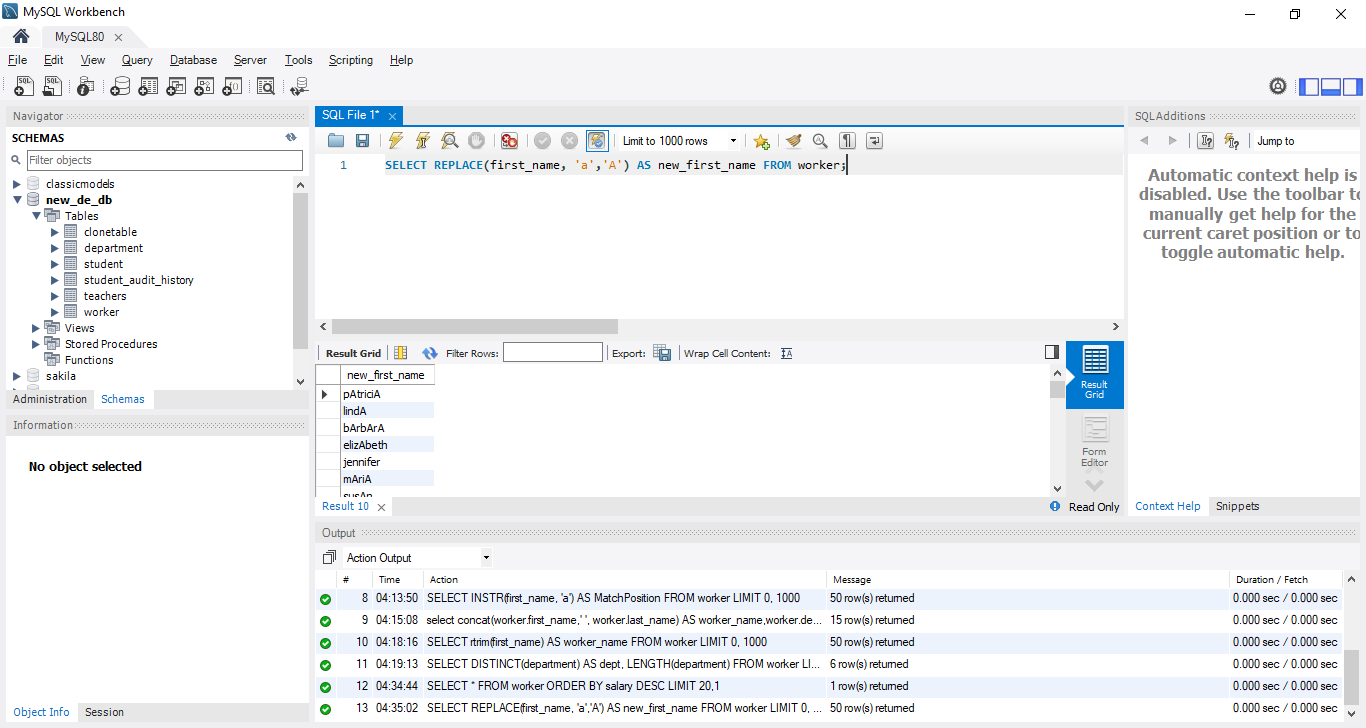
Here n = 20



# **Task-4**

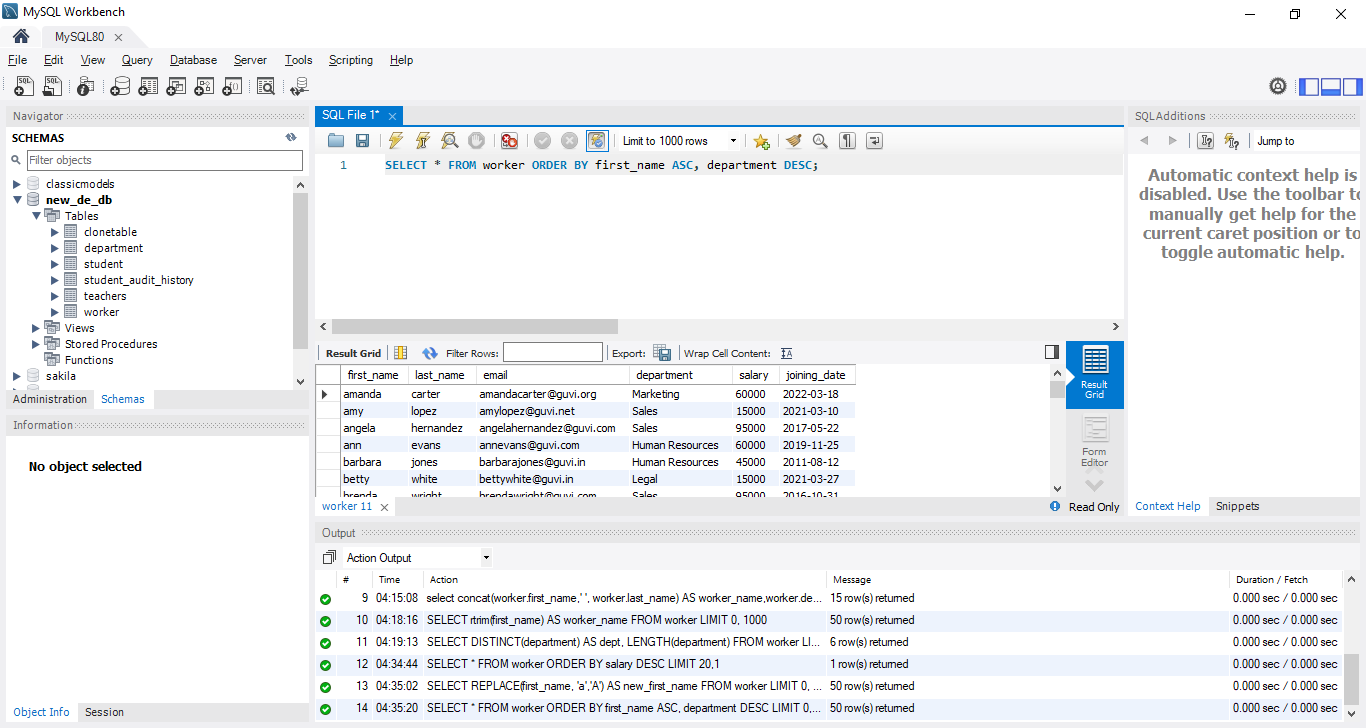
1. Write an SQL query to print the FIRST\_NAME from the Worker table after replacing ‘a’ with ‘A’.

**Query - SELECT REPLACE(first\_name, 'a','A') AS new\_first\_name FROM worker;**



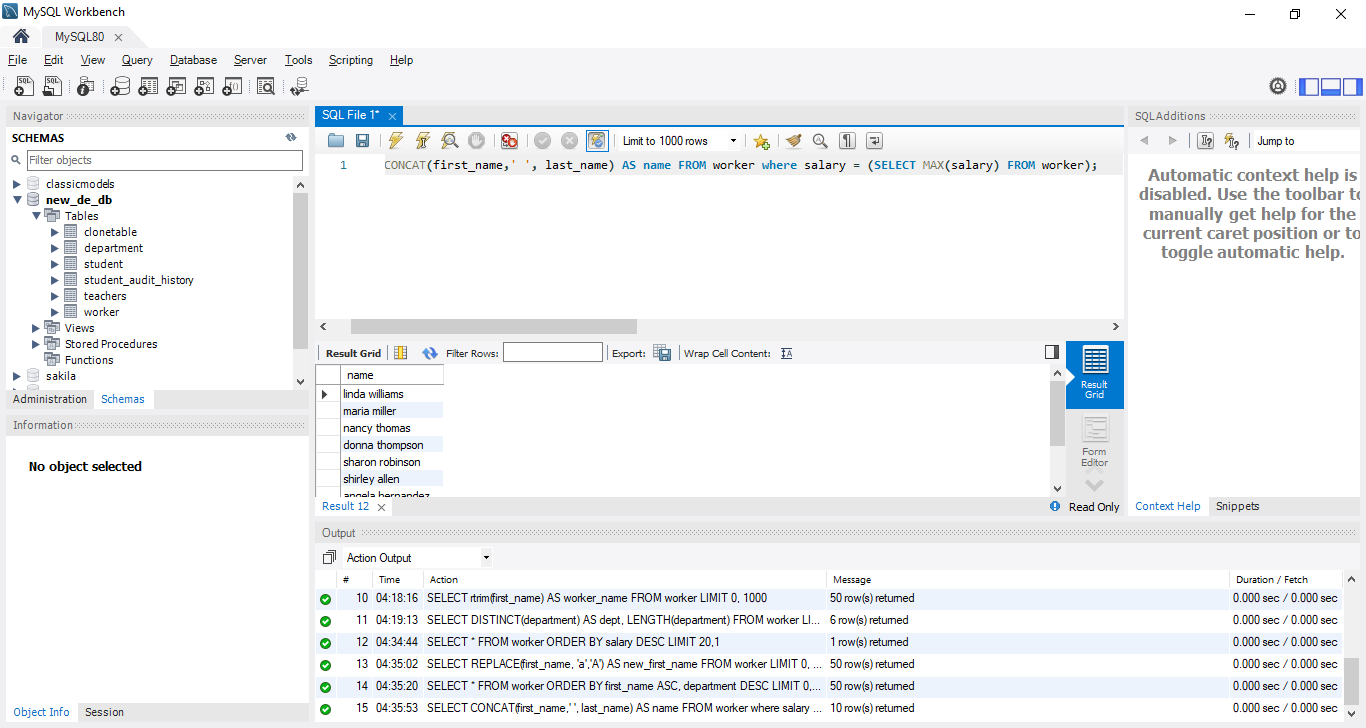
1. Write an SQL query to print all Worker details from the Worker table order FIRST\_NAME Ascending and DEPARTMENT Descending.

**Query - SELECT \* FROM worker ORDER BY first\_name ASC, department DESC;**



1. Write an SQL query to fetch the names of workers who earn the highest salary.

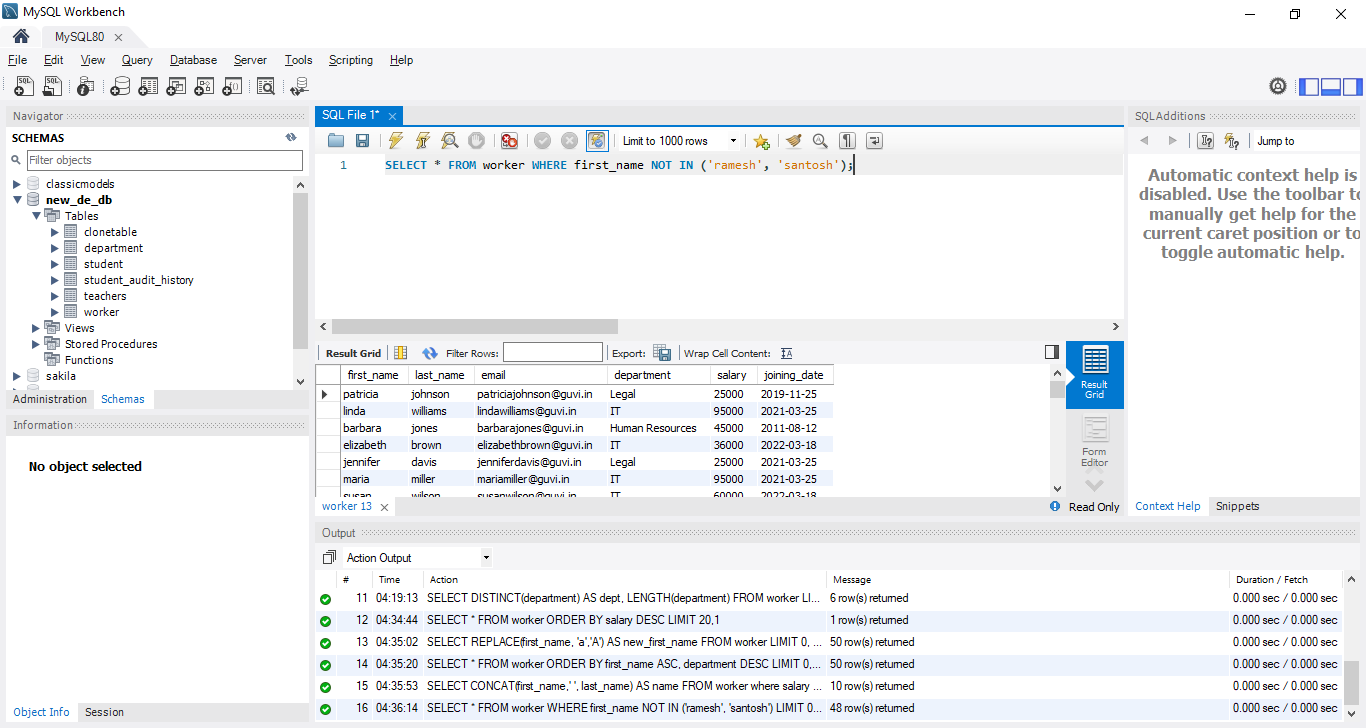
**Query - SELECT CONCAT(first\_name,' ', last\_name) AS name FROM worker where salary = (SELECT MAX(salary) FROM worker);**



# **Task-5**

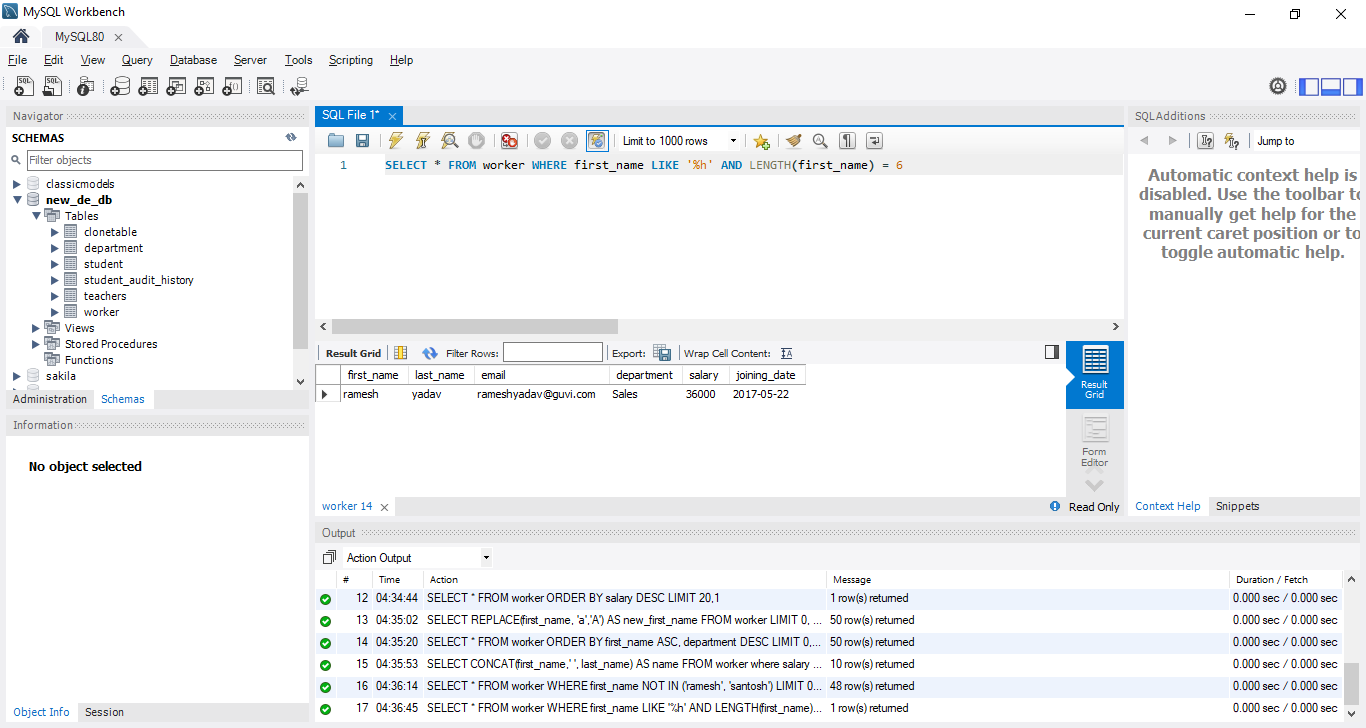
1. Write an SQL query to print details of workers excluding first names, “Ramesh” and “Santhosh” from the Worker table.

**Query - SELECT \* FROM worker WHERE first\_name NOT IN ('ramesh', 'santosh');**



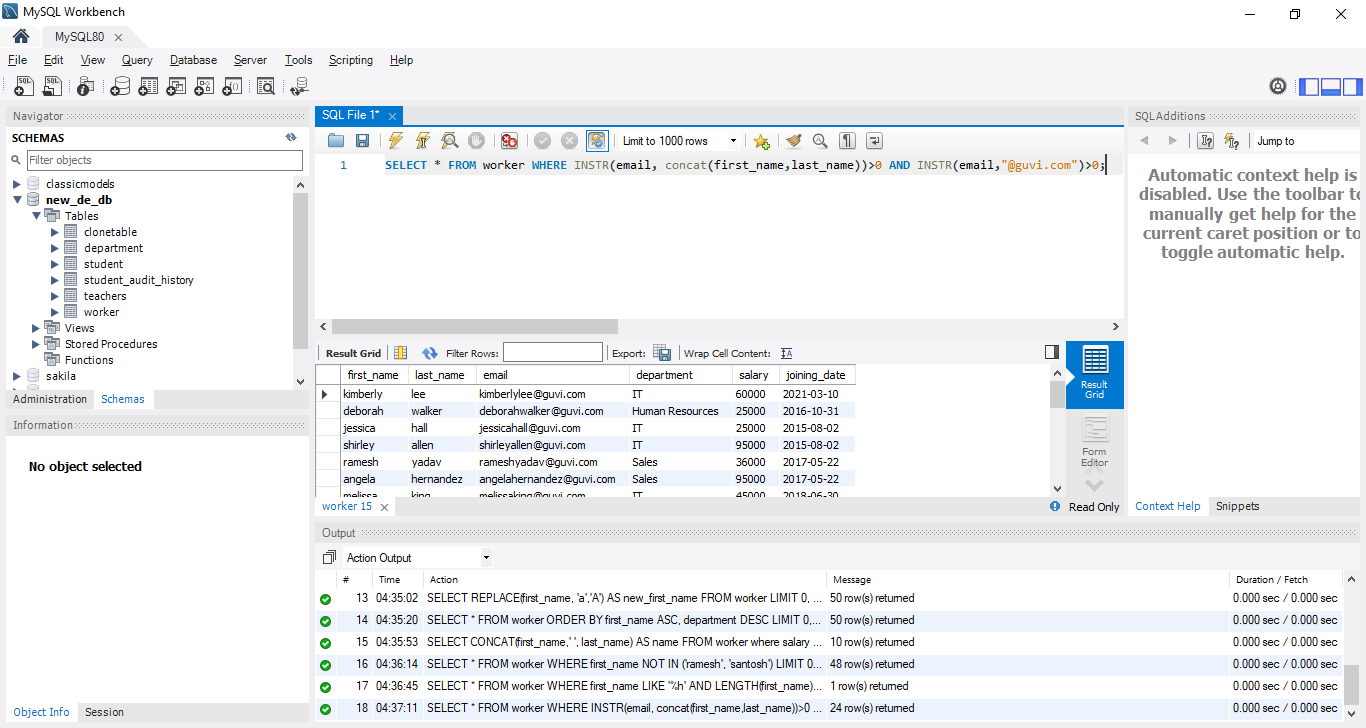
1. Write an SQL query to print details of the Workers whose FIRST\_NAME ends with ‘h’ and contains six alphabets.

**Query - SELECT \* FROM worker WHERE first\_name LIKE '%h' AND LENGTH(first\_name) = 6**



1. Write a query to validate Email of Employee (email should have first name last name and guvi.com example (first name=Kamal last name= raja and the mail id should be kamalraja@guvi.com).

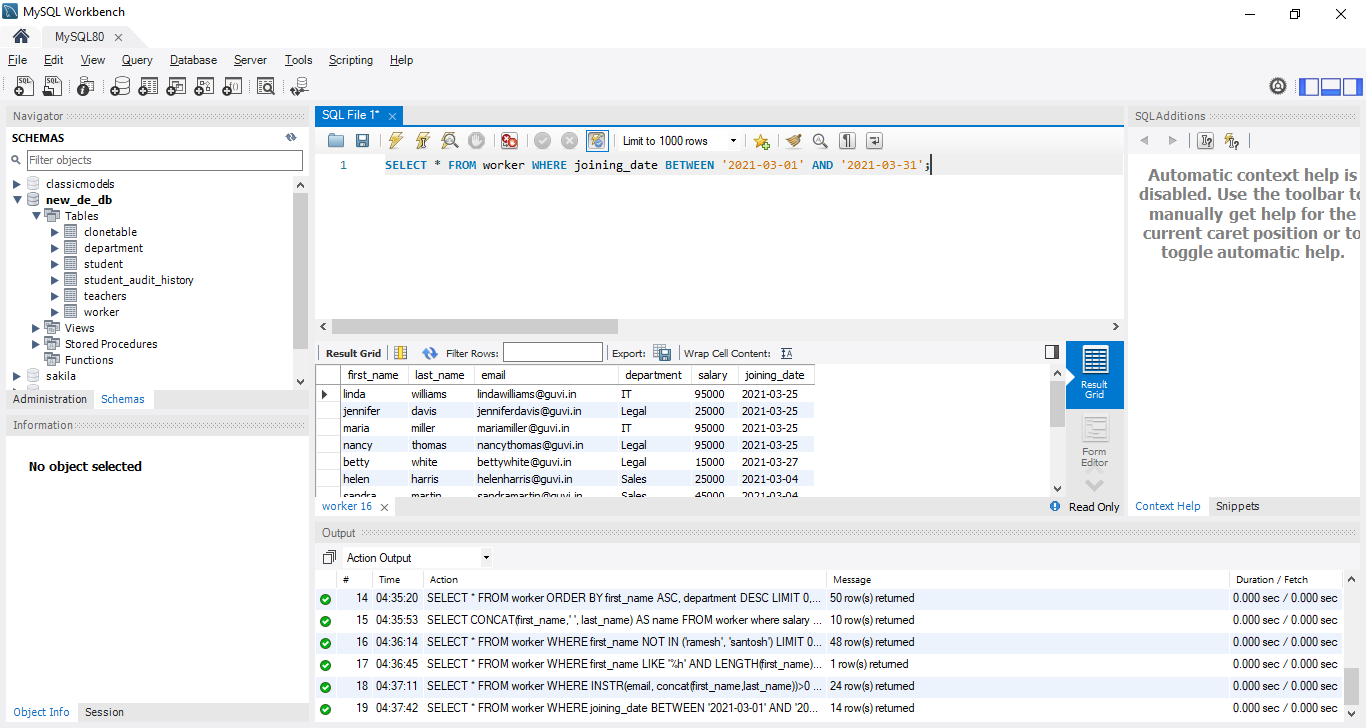
**Query - SELECT \* FROM worker WHERE INSTR(email, concat(first\_name,last\_name))>0 AND INSTR(email,"@guvi.com")>0;**



# **Task-6**

1. Write an SQL query to print details of the Workers who have joined in March ’2021.

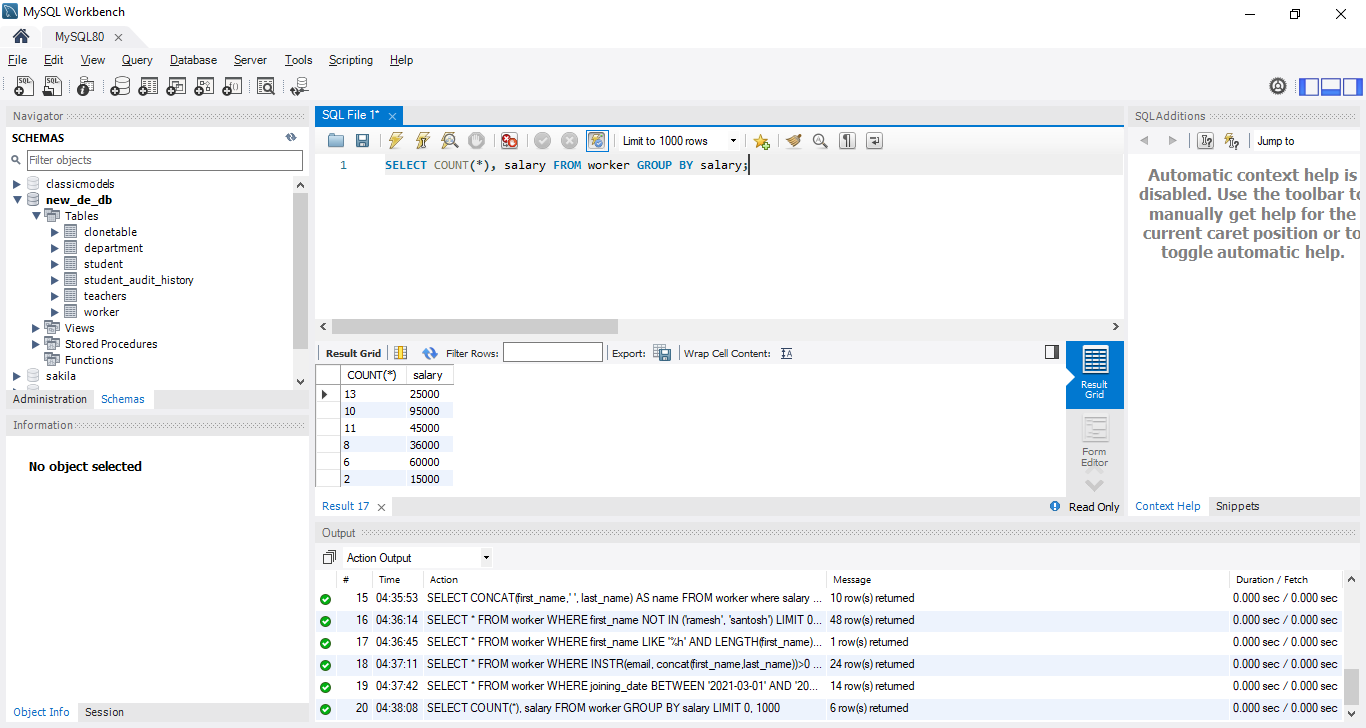
**Query - SELECT \* FROM worker WHERE joining\_date BETWEEN '2021-03-01' AND '2021-03-31';**



1. Write an SQL query to fetch duplicates that have matching data in some fields of a table.

**Query - SELECT COUNT(\*), salary FROM worker GROUP BY salary;**

**# here looking for duplicate data in salary column**



1. How to remove duplicate rows from the Employees table

**Query - CREATE TABLE if not exists removeduplicate AS**

**SELECT \* FROM worker**

**GROUP BY first\_name,**

**last\_name,**

**email,**

**department,**

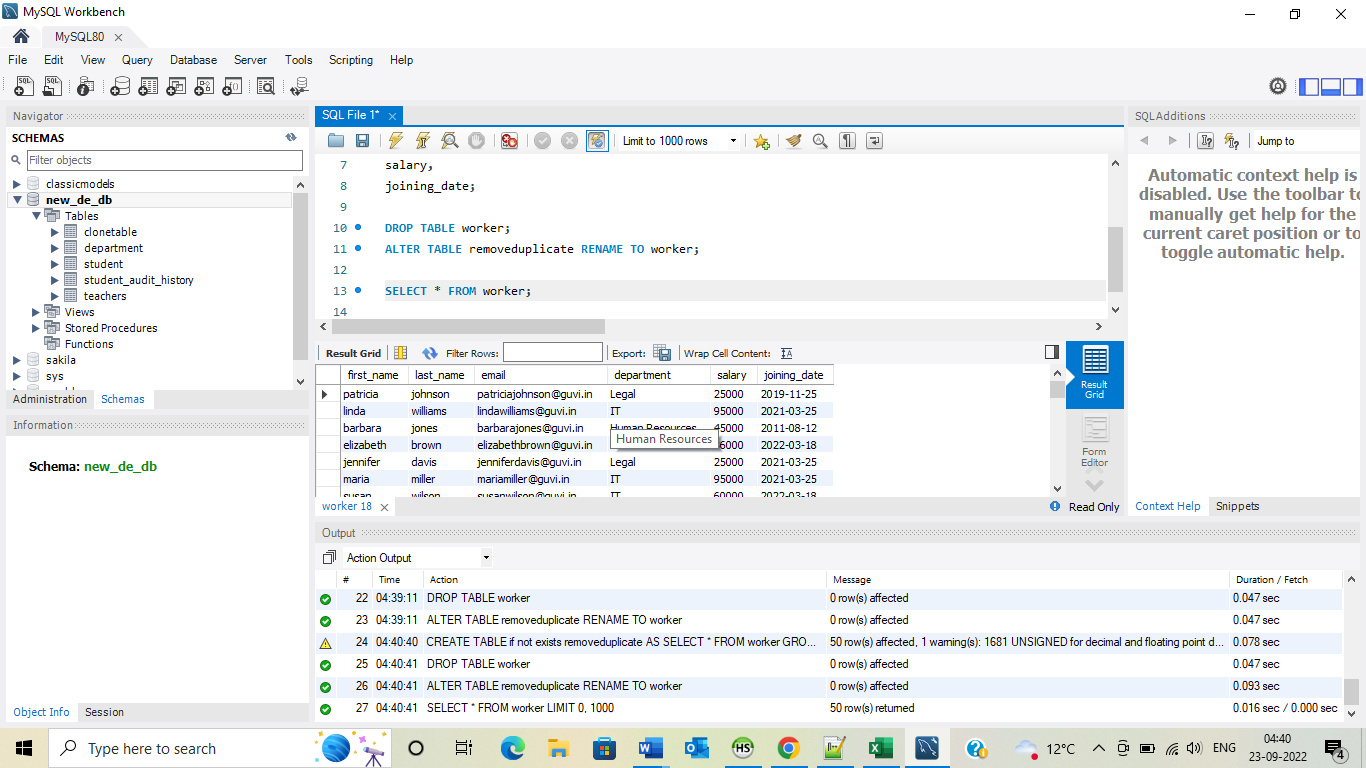
**salary,**

**joining\_date;**

**DROP TABLE worker;**

**ALTER TABLE removeduplicate RENAME TO worker;**

Created a new table with distinct rows and dropped the original table and renamed the new table to original table.



# **Task-7**

1. Write an SQL query to show only odd rows from a table.

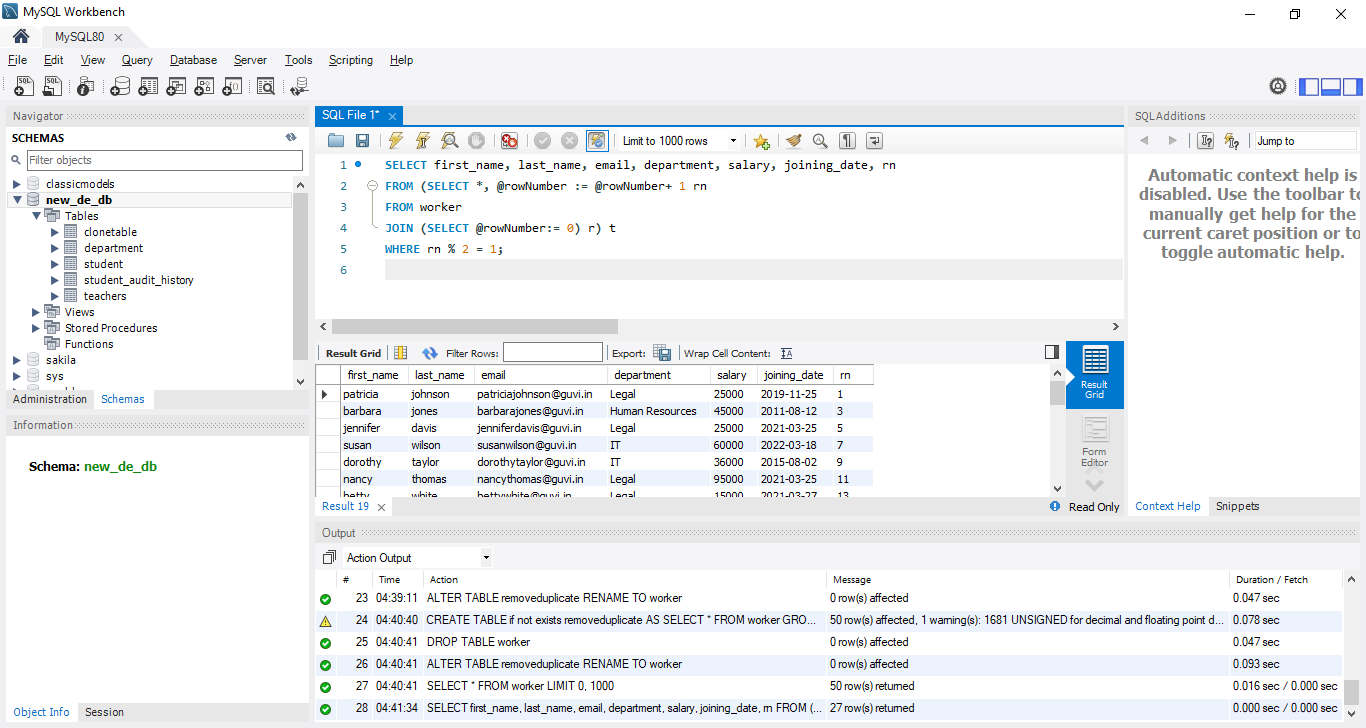
**Query - SELECT first\_name, last\_name, email, department, salary, joining\_date, rn**

**FROM (SELECT \*, @rowNumber := @rowNumber+ 1 rn**

**FROM worker**

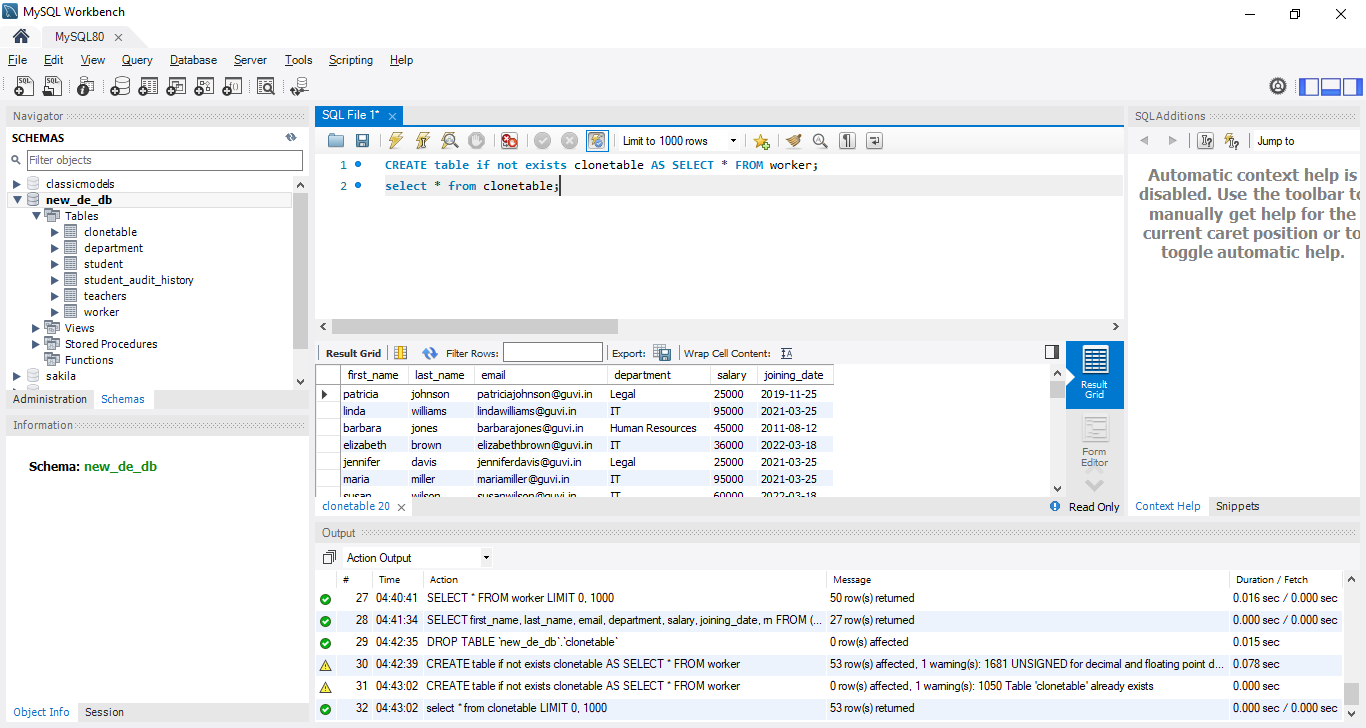
**JOIN (SELECT @rowNumber:= 0) r) t**

**WHERE rn % 2 = 1;**



1. Write an SQL query to clone a new table from another table.

**Query - CREATE table if not exists clonetable AS SELECT \* FROM worker;**



# **Task-8**

1. Write an SQL query to fetch intersecting records of two tables.

**Query - SELECT \* FROM worker w**

**WHERE EXISTS (SELECT \* FROM clonetable ct**

**WHERE ct.first\_name = w.first\_name**

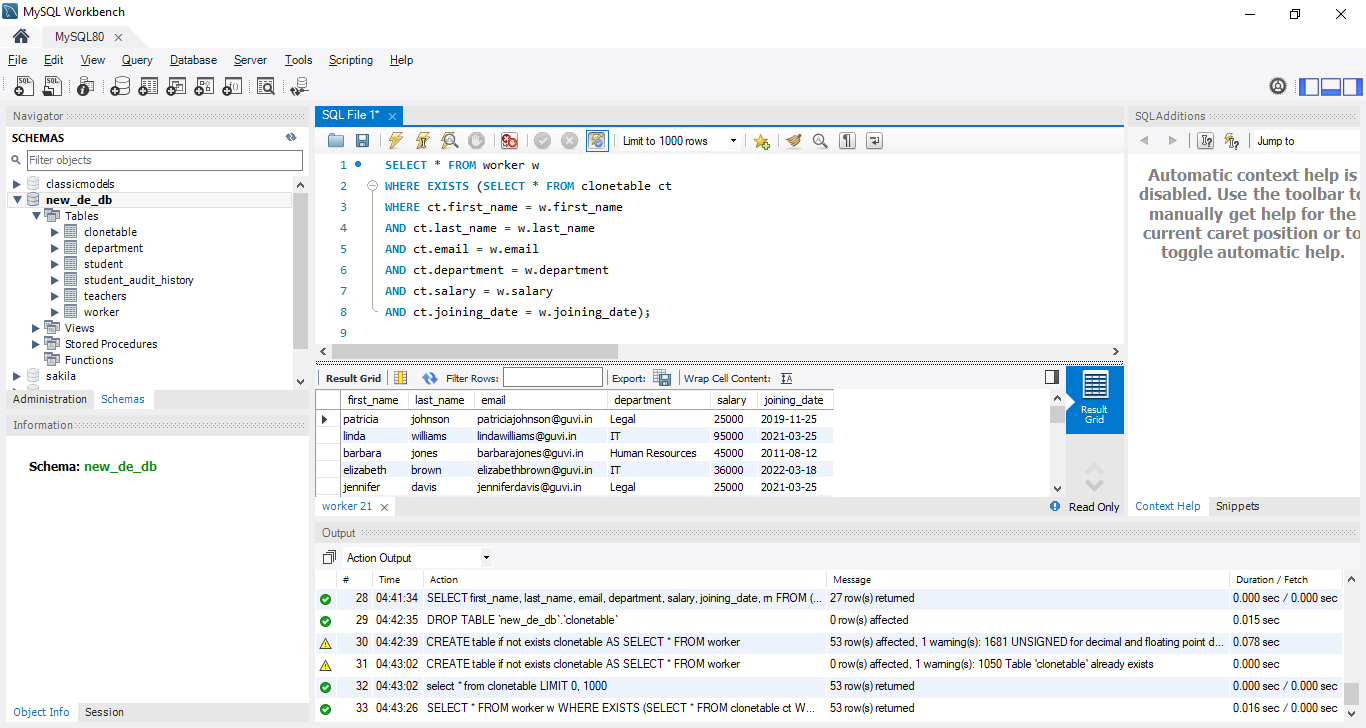
**AND ct.last\_name = w.last\_name**

**AND ct.email = w.email**

**AND ct.department = w.department**

**AND ct.salary = w.salary**

**AND ct.joining\_date = w.joining\_date);**



1. Write an SQL query to show records from one table that another table does not have.

**Query** - **SELECT \* FROM worker w**

**WHERE NOT EXISTS (SELECT \* FROM clonetable ct**

**WHERE ct.first\_name = w.first\_name**

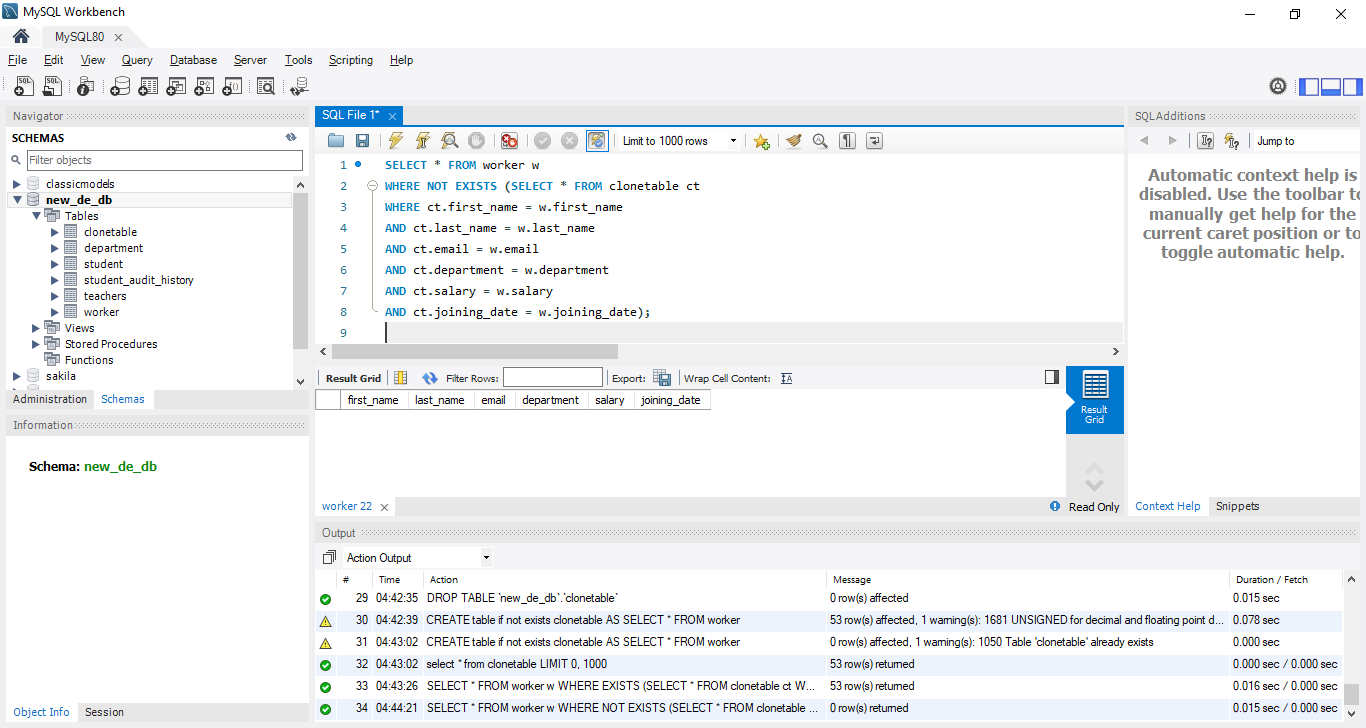
**AND ct.last\_name = w.last\_name**

**AND ct.email = w.email**

**AND ct.department = w.department**

**AND ct.salary = w.salary**

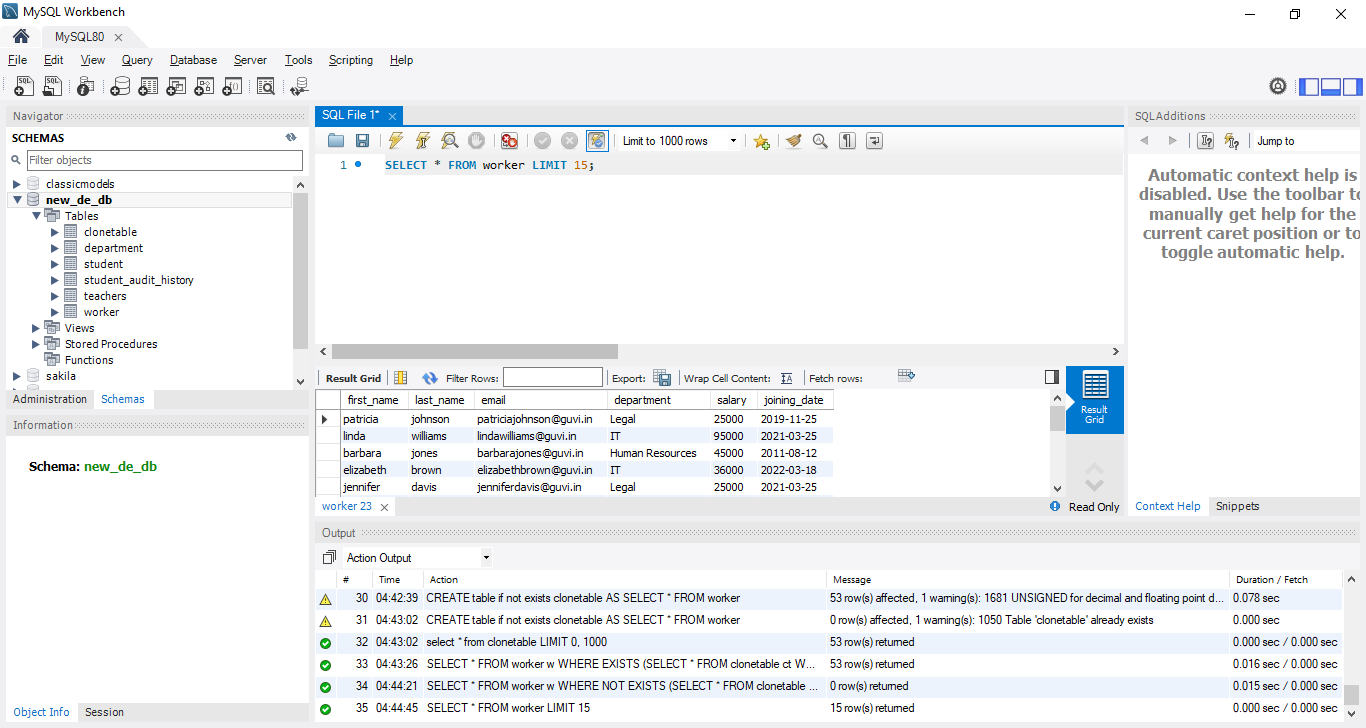
**AND ct.joining\_date = w.joining\_date);**



# **Task-9**

1. Write an SQL query to show the top n (say 15) records of a table.

**Query - SELECT \* FROM worker LIMIT 15;**

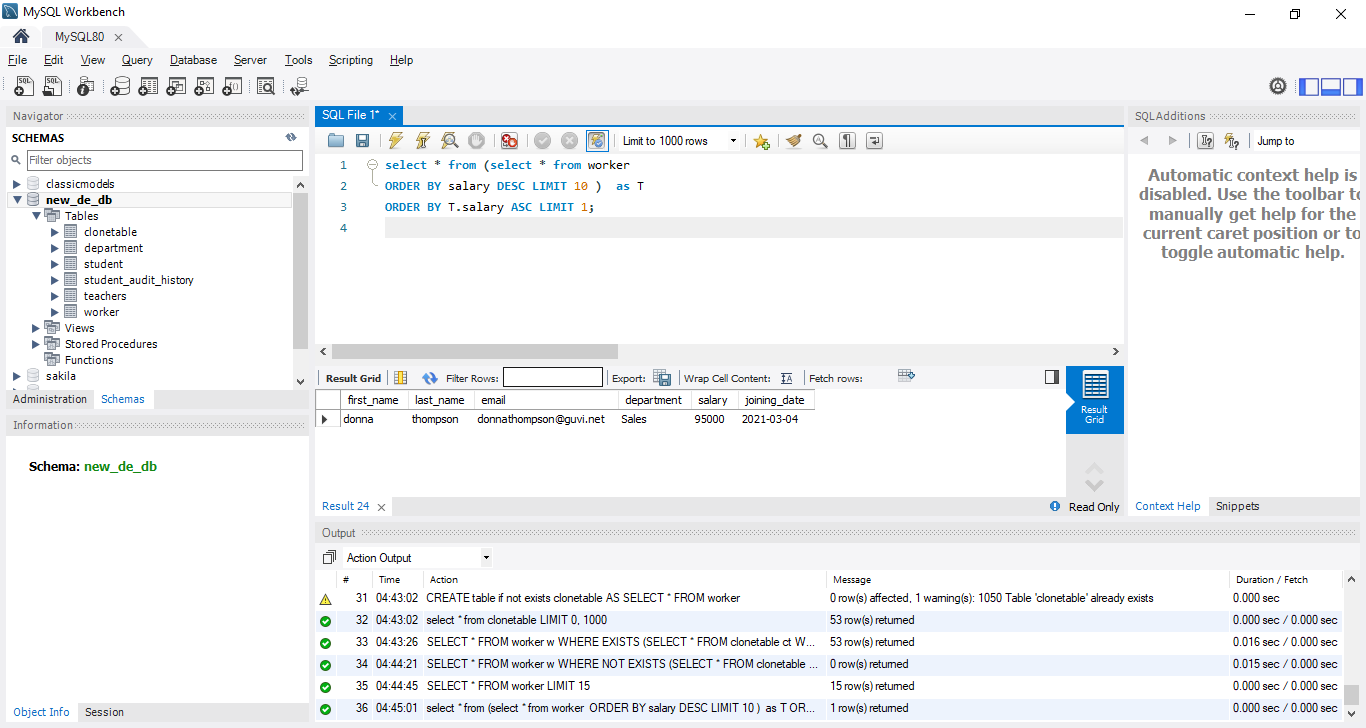


1. Write an SQL query to determine the nth (say n=10) highest salary from a table.

**Query - select \* from (select \* from worker**

**ORDER BY salary DESC LIMIT 10 ) as T**

**ORDER BY T.salary ASC LIMIT 1;**



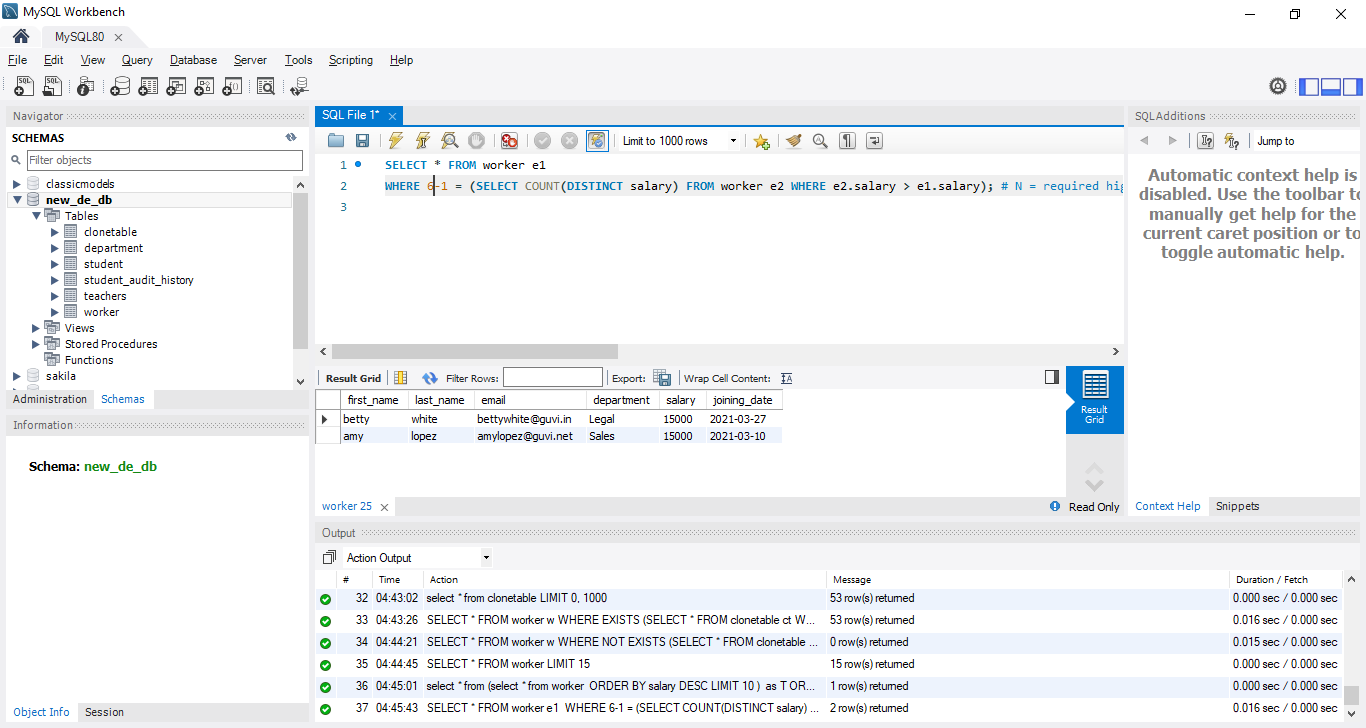
# **Task-10**

1. Write an SQL query to determine the 8th highest salary without using TOP or LIMIT methods.

**Query - SELECT \* FROM worker e1**

**WHERE N-1 = (SELECT COUNT(DISTINCT salary) FROM worker e2 WHERE e2.salary > e1.salary); # N = required highest salary**

**Below example N = 6**



1. Write an SQL query to fetch the list of employees with the same salary.

**Query - select t.\* from (select t.\*, count(\*) over (partition by salary) as salarycnt**

**from worker t) t**

**where salarycnt > 1**

**order by salary;**

