| Experiment No.5 |
| --- |
| String Manipulation, Aggregrate Function, Group By |
| Date of Performance: 27/2/24 |
| Date of Submission: 19/3/24 |

**Experiment No. 5: Perform Simple queries, string manipulation operations.**

**Course Outcome [CSL503.3]:** Apply SQL queries ,triggers and procedures for specific module/task

Aim:- Perform Simple queries, string manipulation operations.

Theory:

1. Union: Union combines the results of two or more SELECT statements, returning only distinct rows from the combined result set, thereby eliminating duplicate records.

2. Union All: Union All also combines the results of multiple SELECT statements, but it includes all rows from each query, even if duplicates are present, resulting in potentially more rows in the output.

3. Intersect: Intersect returns the common rows between two SELECT statements, providing the intersection of their result sets, effectively showing only records that exist in both tables.

4. Minus / Except: Minus (or Except) returns rows from the first SELECT statement that are not present in the second SELECT statement, helping to identify differences between datasets.

5. ASCII Value: ASCII Value retrieves the numeric ASCII code for a given character, providing a standardized representation of characters used in computing.

6. Padding: Padding adds characters, usually spaces, to the beginning, end, or both sides of a string to make it a specified length, commonly used for formatting output.

7. Trimming: Trimming removes leading, trailing, or both leading and trailing spaces from a string, helping to clean and normalize data stored in databases.

8. Repeat: Repeat duplicates a string a specified number of times, useful for generating repeated patterns or formatting strings in a desired manner.

9. Replace: Replace substitutes occurrences of a specified substring within a string with another substring, facilitating data manipulation and cleanup tasks.

10. Reverse: Reverse changes the order of characters in a string, flipping it backwards, which can be helpful for various text processing tasks, such as generating palindromes or reversing input strings.

Selecting Substrings:

- SUBSTRING(): SUBSTRING extracts a portion of a string, starting from a specified position and optionally with a specified length, useful for extracting substrings or segments of text.

- TRIM(): TRIM removes specified characters, such as leading or trailing spaces, from the beginning, end, or both sides of a string, aiding in data cleaning and formatting.

- UPPER(): UPPER converts all characters in a string to uppercase, facilitating case-insensitive comparisons or enforcing uniform casing.

- LOWER(): LOWER converts all characters in a string to lowercase, similar to UPPER but for lowercase representation.

- LENGTH(): LENGTH calculates the length of a string, counting the number of characters, which is valuable for validation or data manipulation based on string length.

Group By Syntax:

The Group By syntax is used in SQL to group rows that have the same values into summary rows, typically used in conjunction with aggregate functions to perform operations on grouped data.

**Code:**

create table Emp(ID integer(20),Name char(20), Salary integer(20));

desc Emp;

insert into Emp(ID , Name,Salary)values(1,'ABC',10000);

insert into Emp(ID , Name,Salary)values(2,'XYZ',20000);

insert into Emp(ID , Name,Salary)values(3,'PQR',30000);

insert into Emp(ID , Name,Salary)values(4,'MNO',40000);

Select \* from Emp;

select \*from Emp Union select \*from Staff;

select \*from Emp Union All select \*from Staff;

select \*from Emp Union All select \*from Staff;

-- select \*from Emp inner join  Staff on  ;

--String Operations

select ASCII ('t');

select left('world',4);

select right('india',3);

select lpad('hi',5,'!');

select rpad('anjali',8,'!');

select ltrim('      anjali gupta');

select rtrim('anjali gupta       ');

select replace("Anjali",'j','s');

select substring('Quadratically',5);

select substring('Quadratically',5,6);

select substring('Quadractically',-6,3);

select trim(  '  ABC    ');

Select trim(leading 'x' from 'xxxxabcxxxx');

Select trim(trailing 'x' from 'xxxxabcxxxx');

Select trim(both'x' from 'xxxxabcxxxx');

select upper('anjali');

select LOWER('ANJALI');

Select length('SQL TUTORIAL');

create table Employe(ID integer(20),Name char(20), Quantity integer (10),Price integer(20),City char(20));

desc Employe;

insert into Employe(ID , Name,Quantity,Price,City)values(1,'ABC',2,2000,'Mumbai'),(2,'PQR',3,4000,'Jaipur'),(3,'XYZ',4,5000,'Haryana'),(4,'LMN',2,60000,'Mumbai'),(5,'ABC',4,2000,'Pune');

select \*from Employe;

select count(\*) from Employe ;

select count(\*) from Employe where city="Mumbai";

select Avg(Price) from Employe;

select min(price) from Employe;

select min(price) from Employe where Quantity = 2;

select max(price) from Employe where Quantity = 2;

select Name ,count(\*) from Employe  groupbyName ;

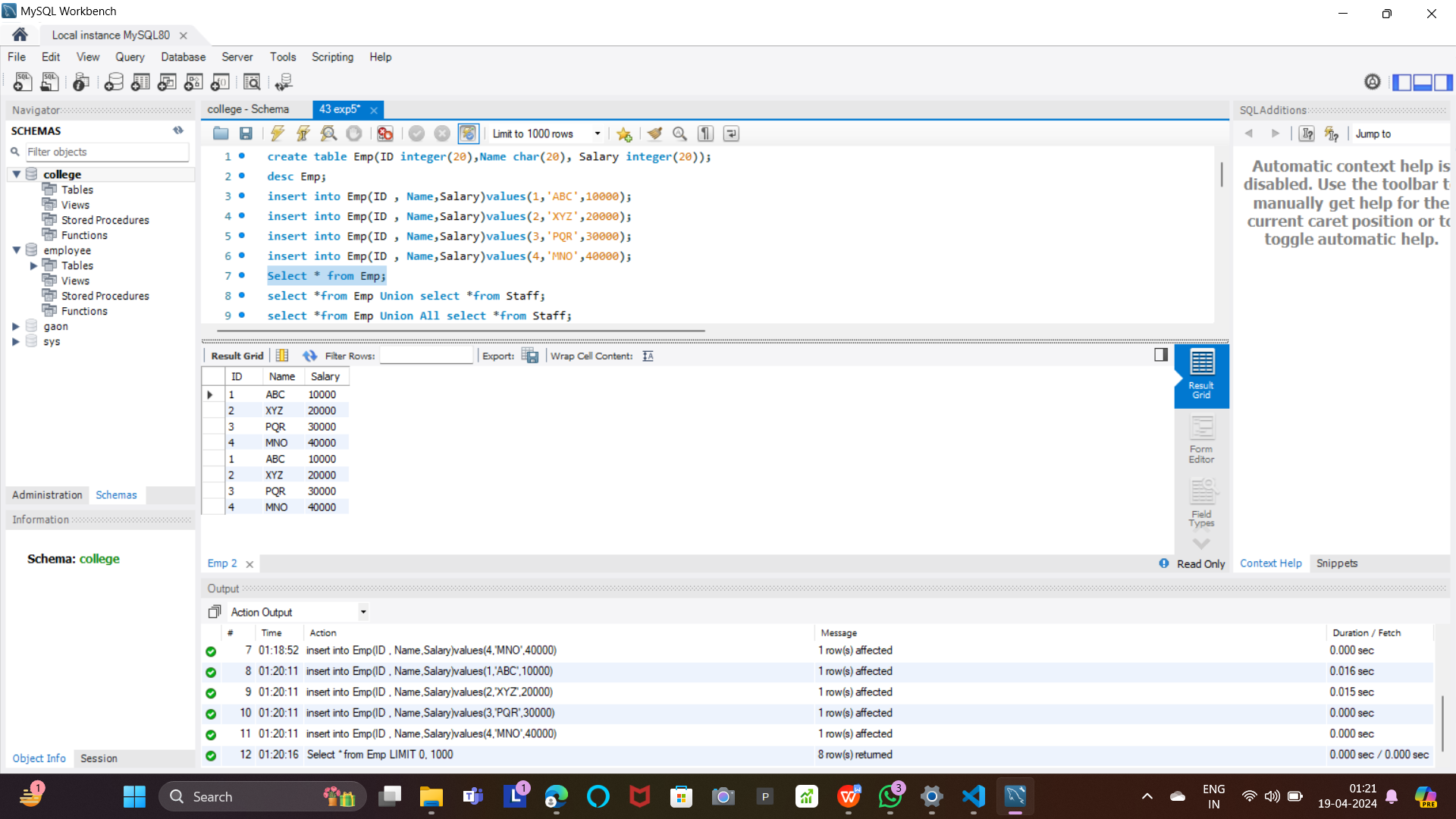
select city ,max(price) from Employe  group by city ;

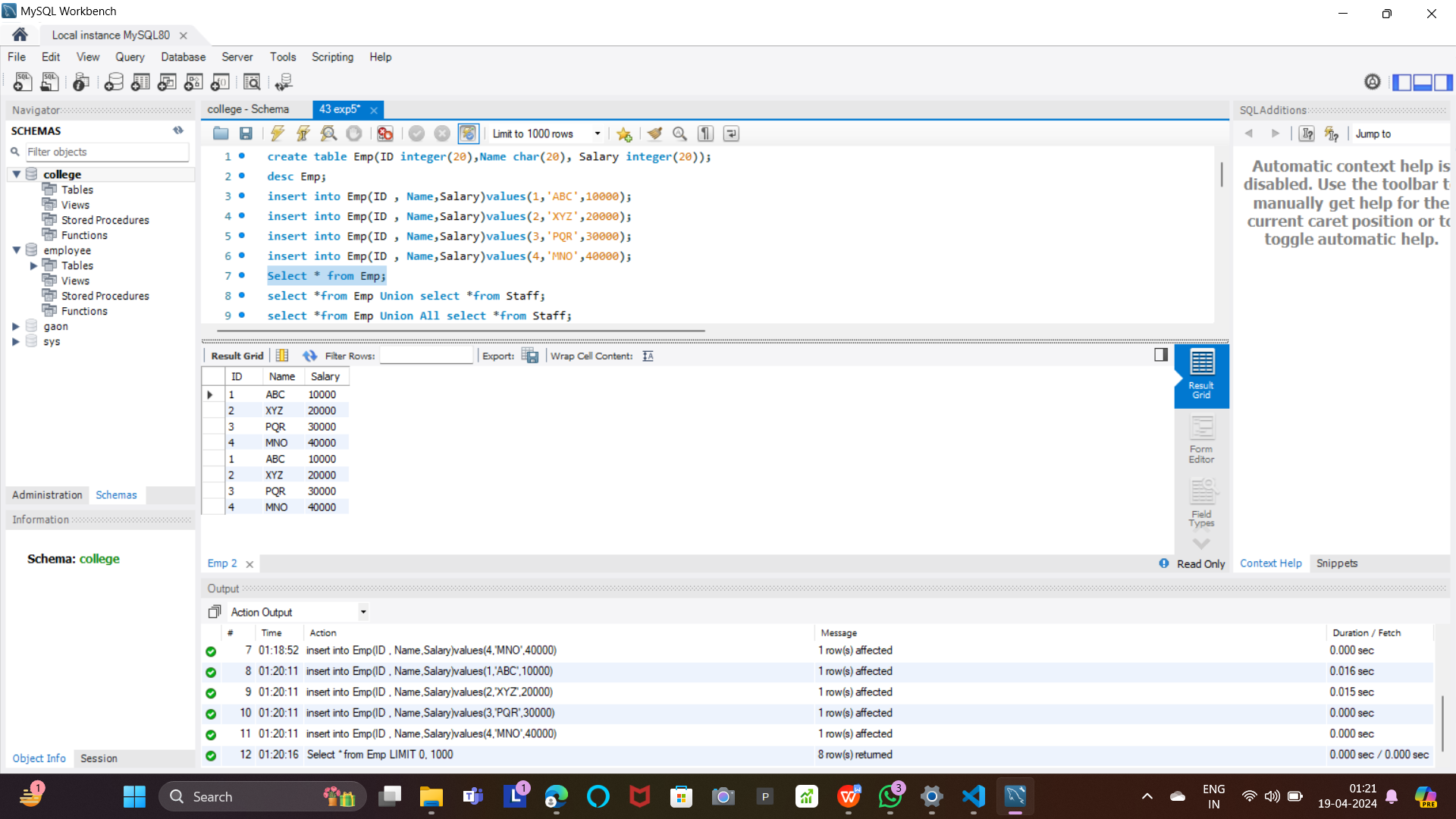
select city ,min(price) from Employe  group by city ;

select city ,sum(price) from Employe  group by city ;

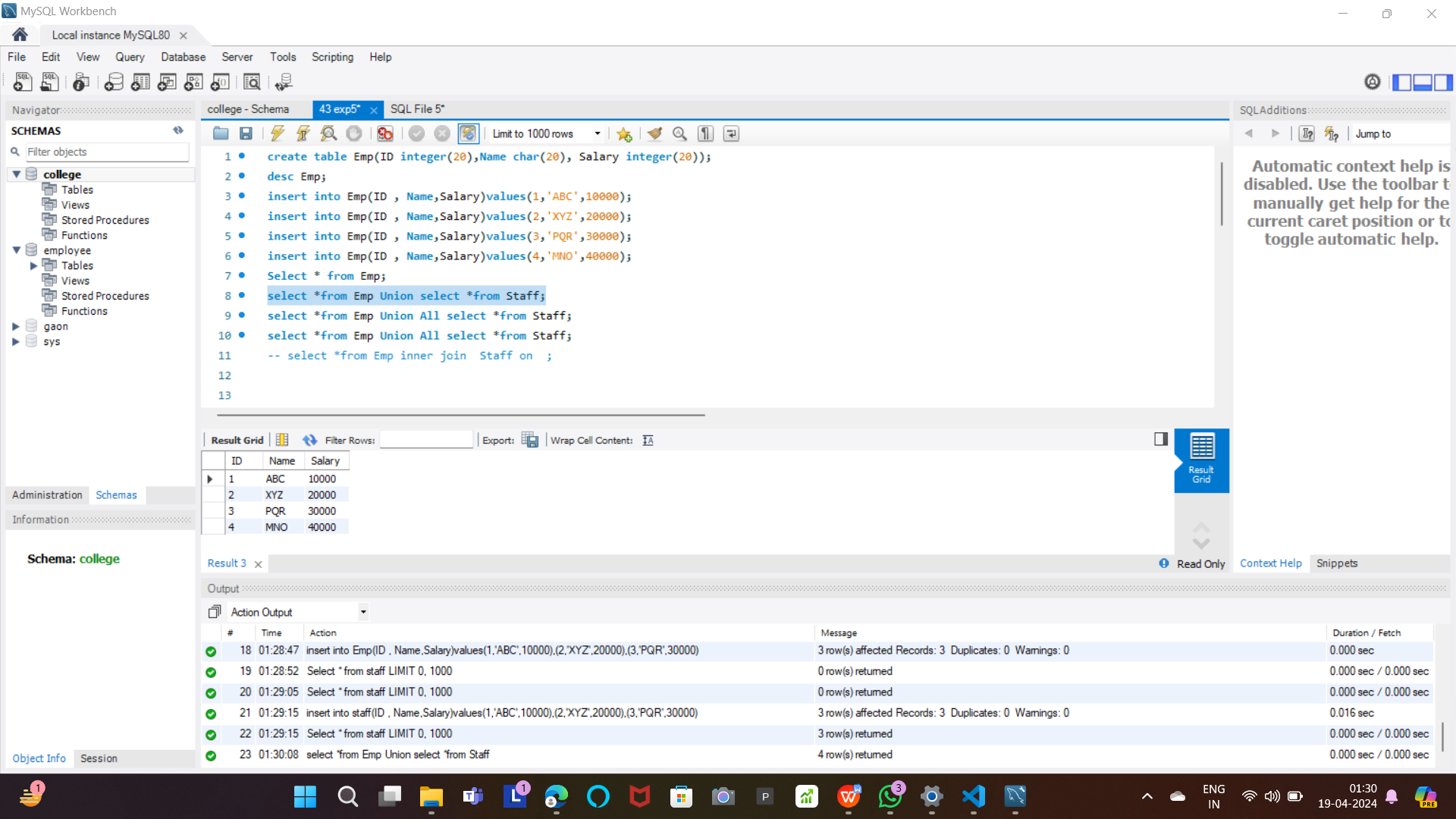
**Output:**

**Table Creation:**

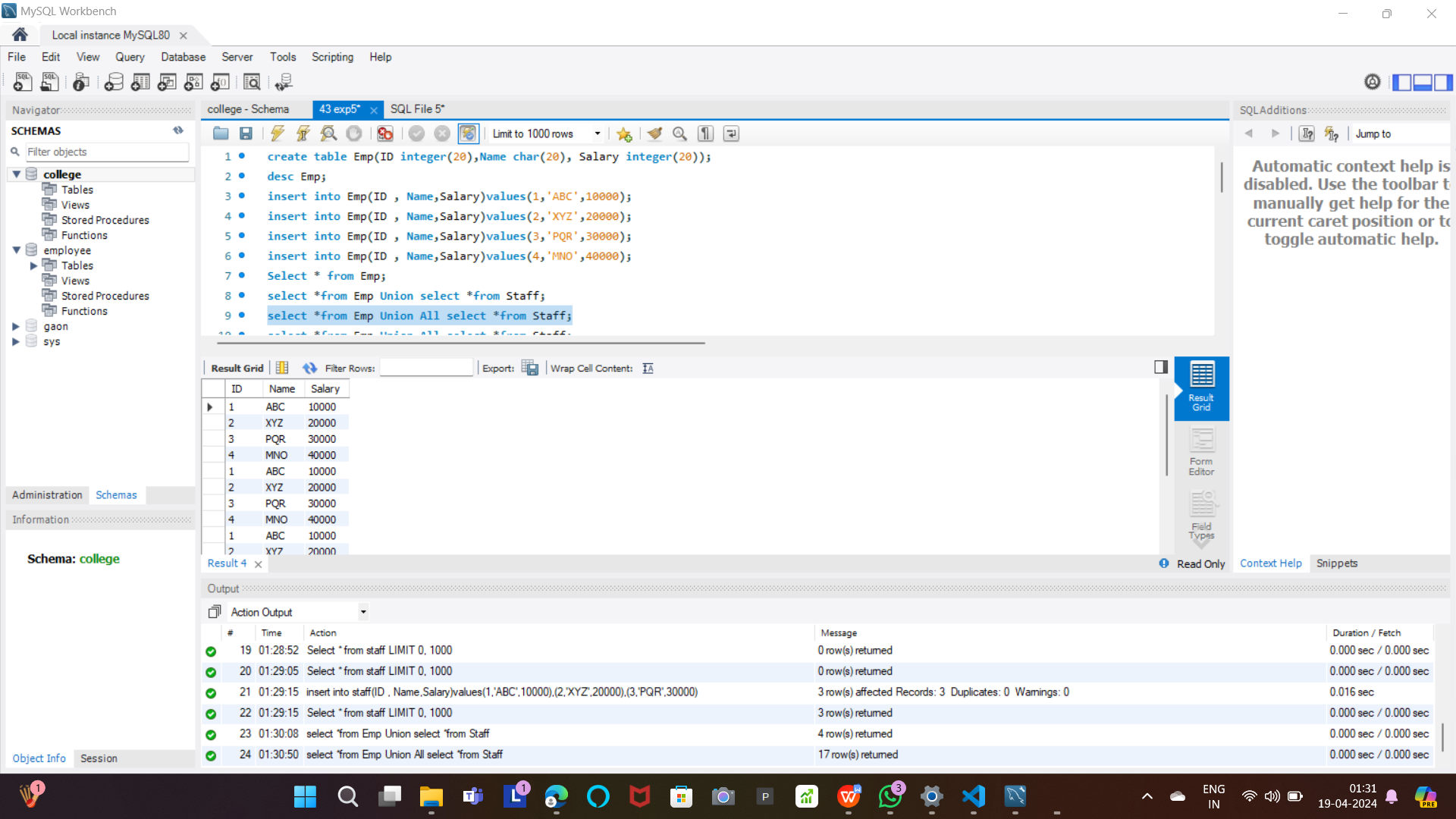




**Union:**

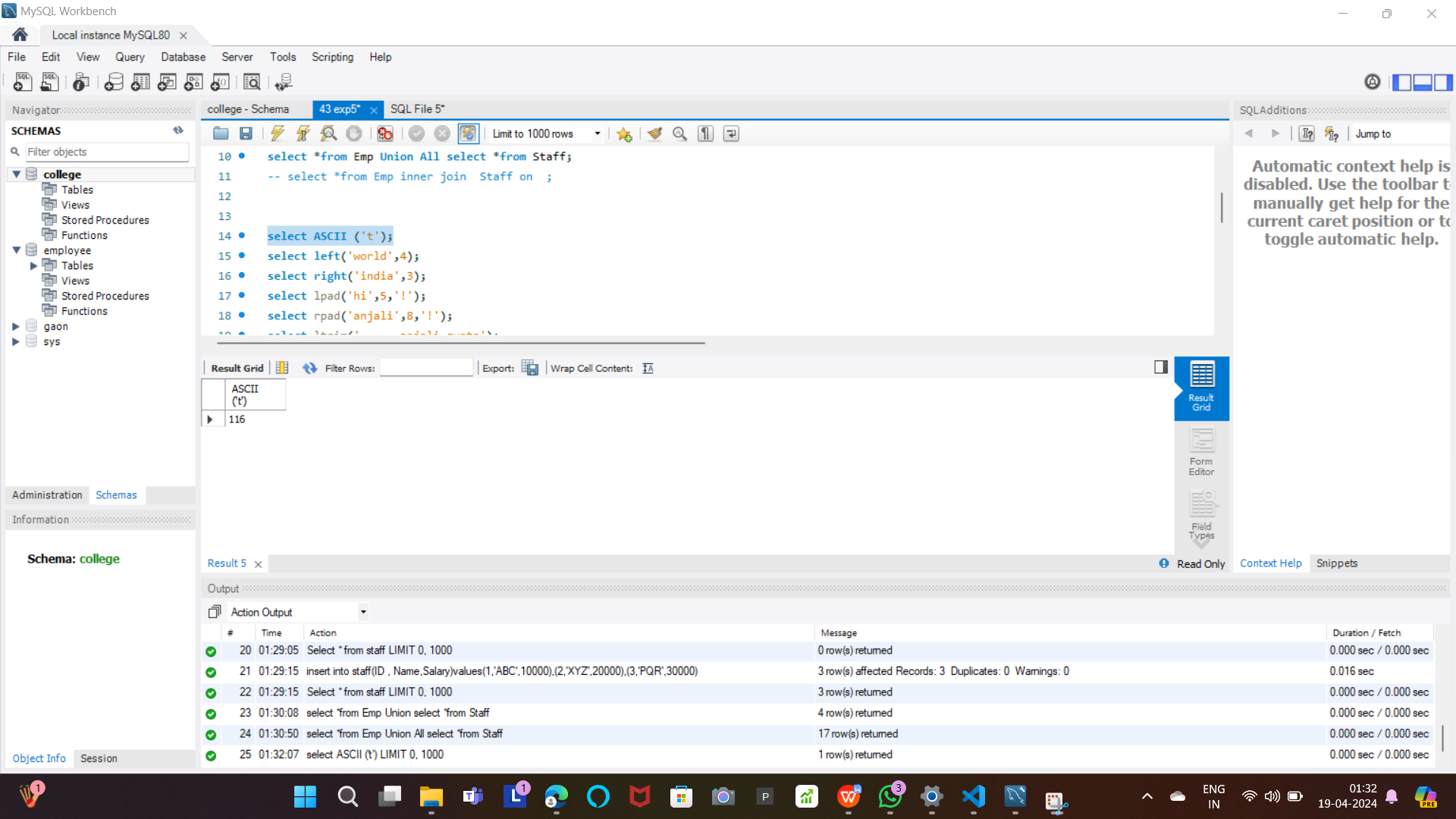


**Union All:**

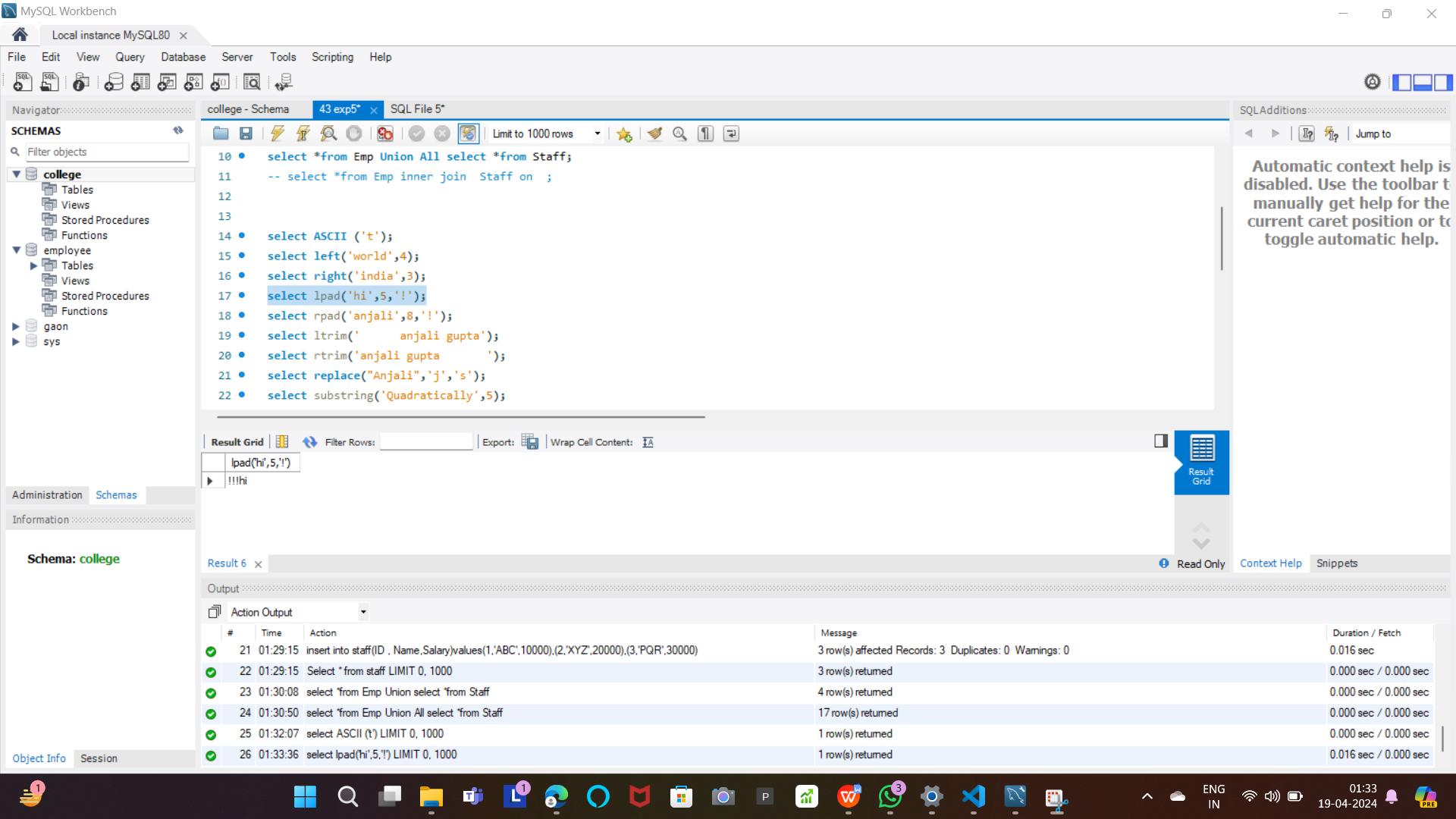


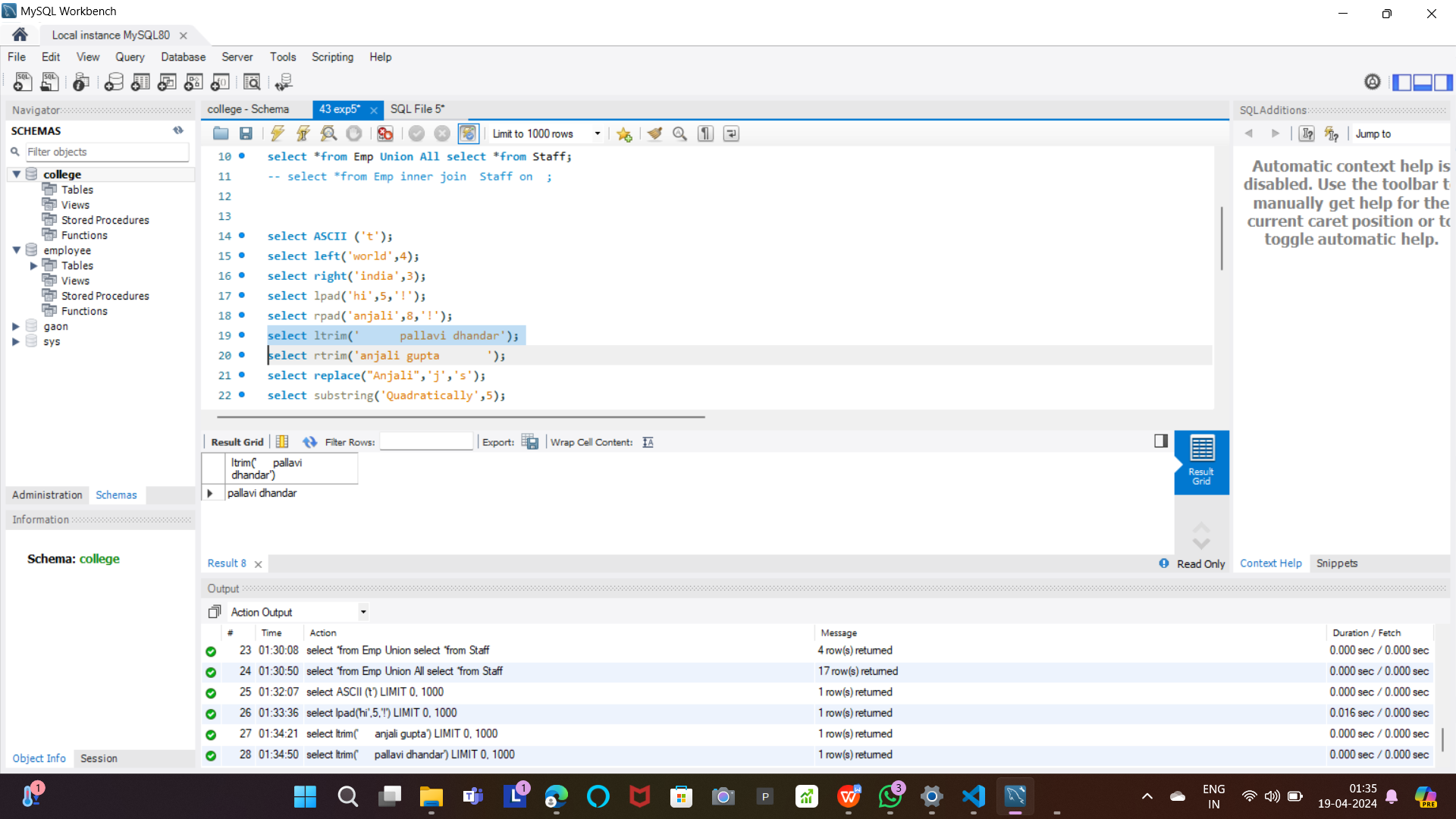
***String Operations:***

ASCII of Characters:

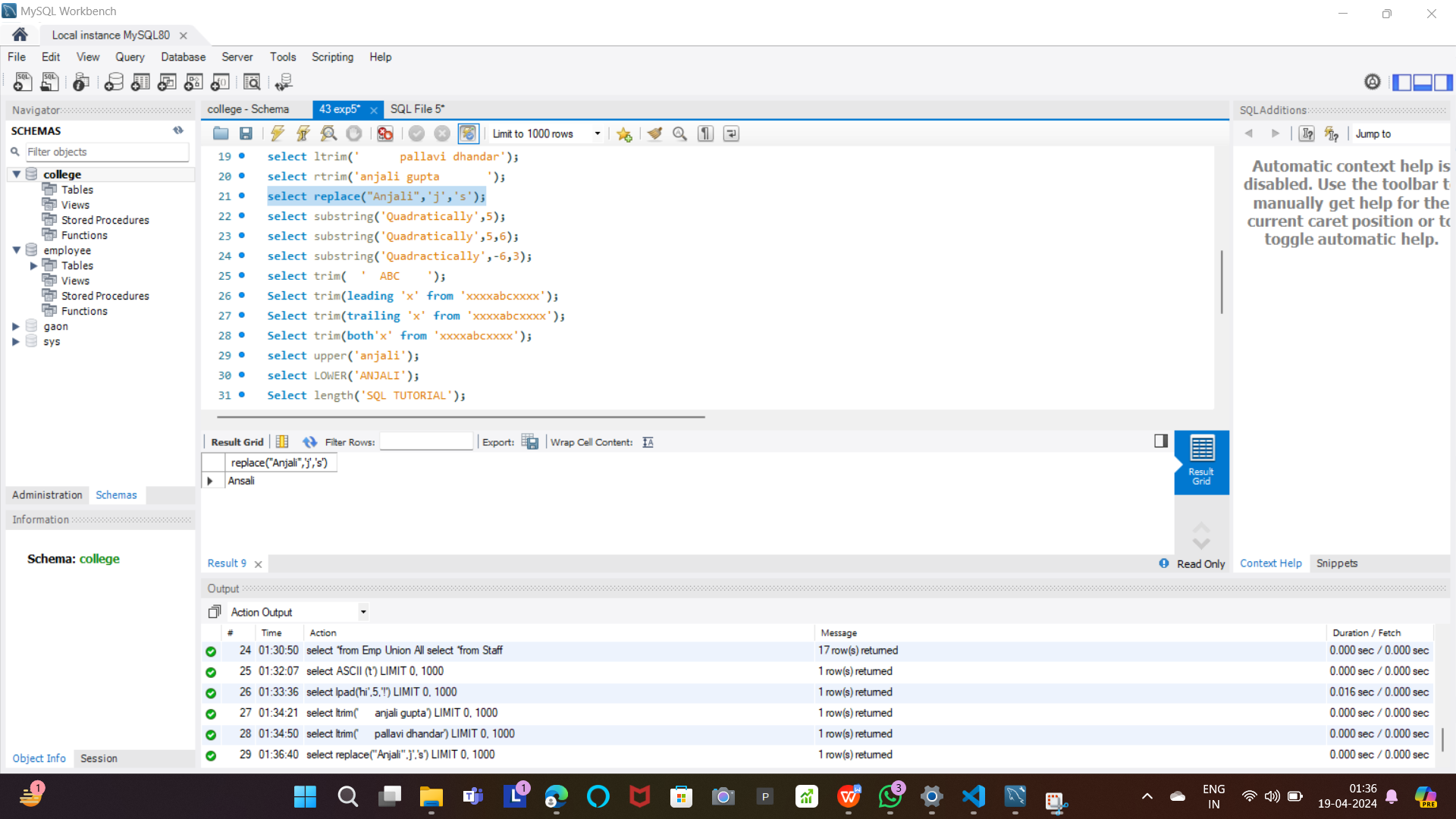


Padding into String:

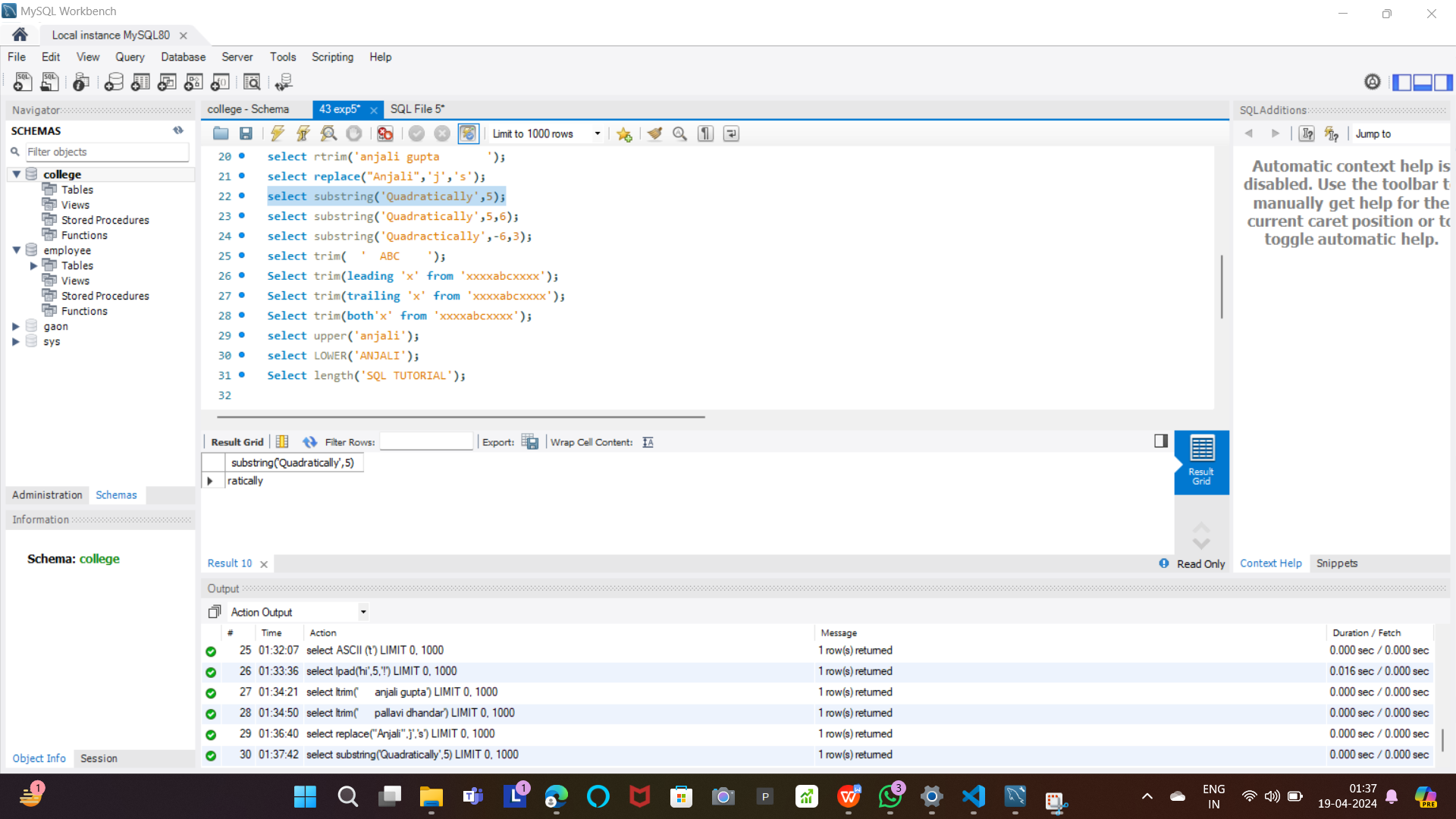


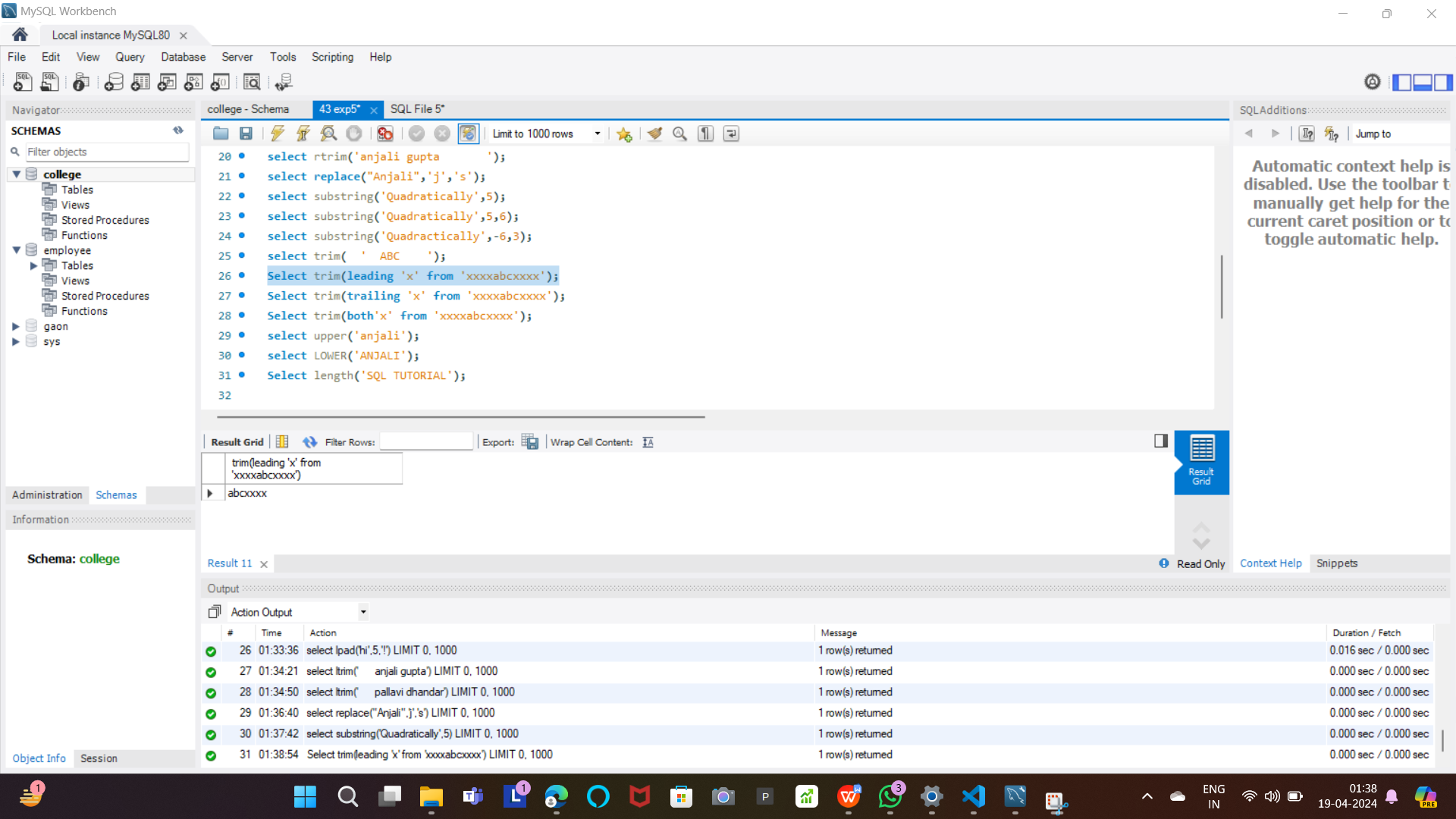


Replace:

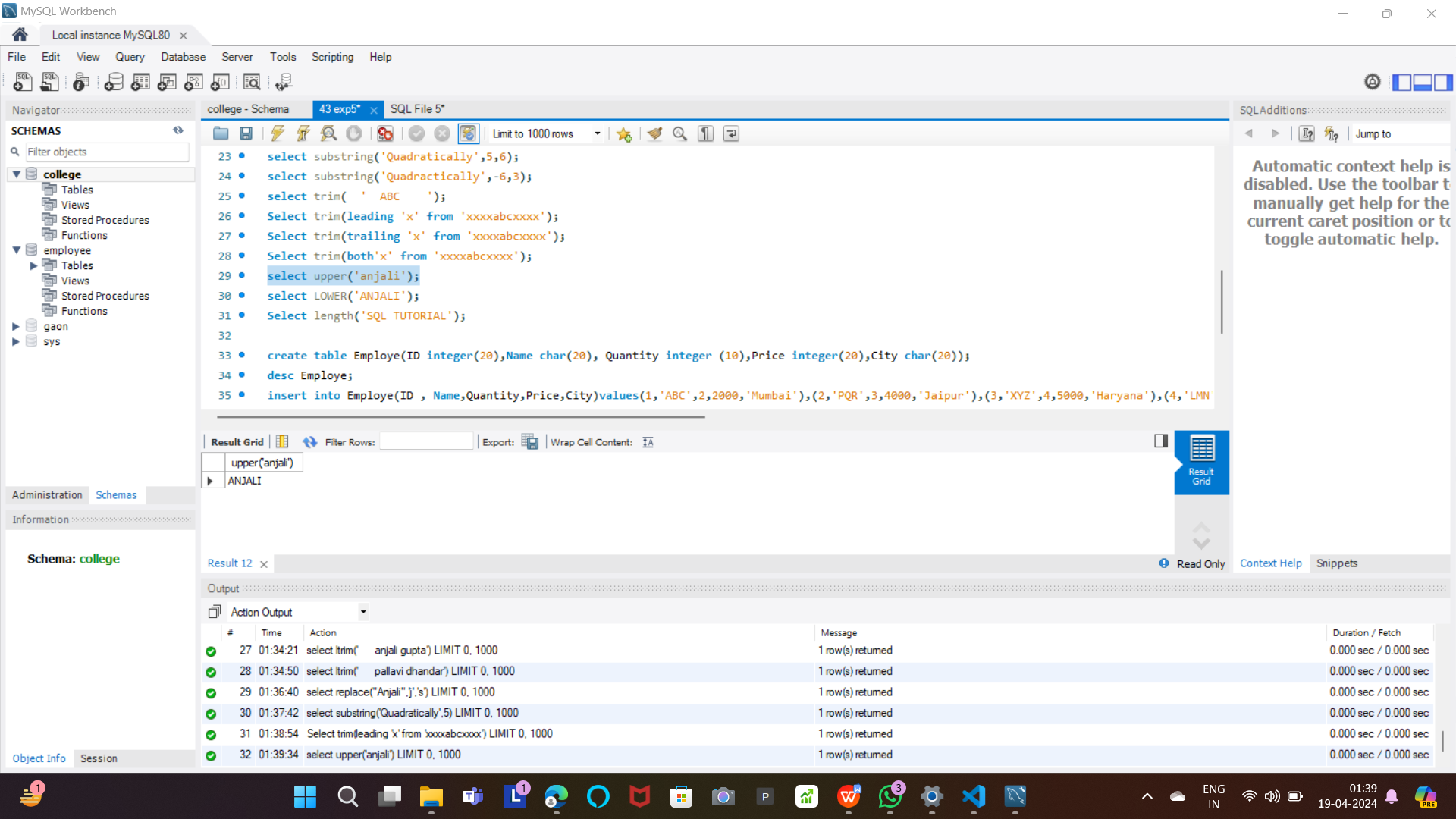


Selecting a Substring:



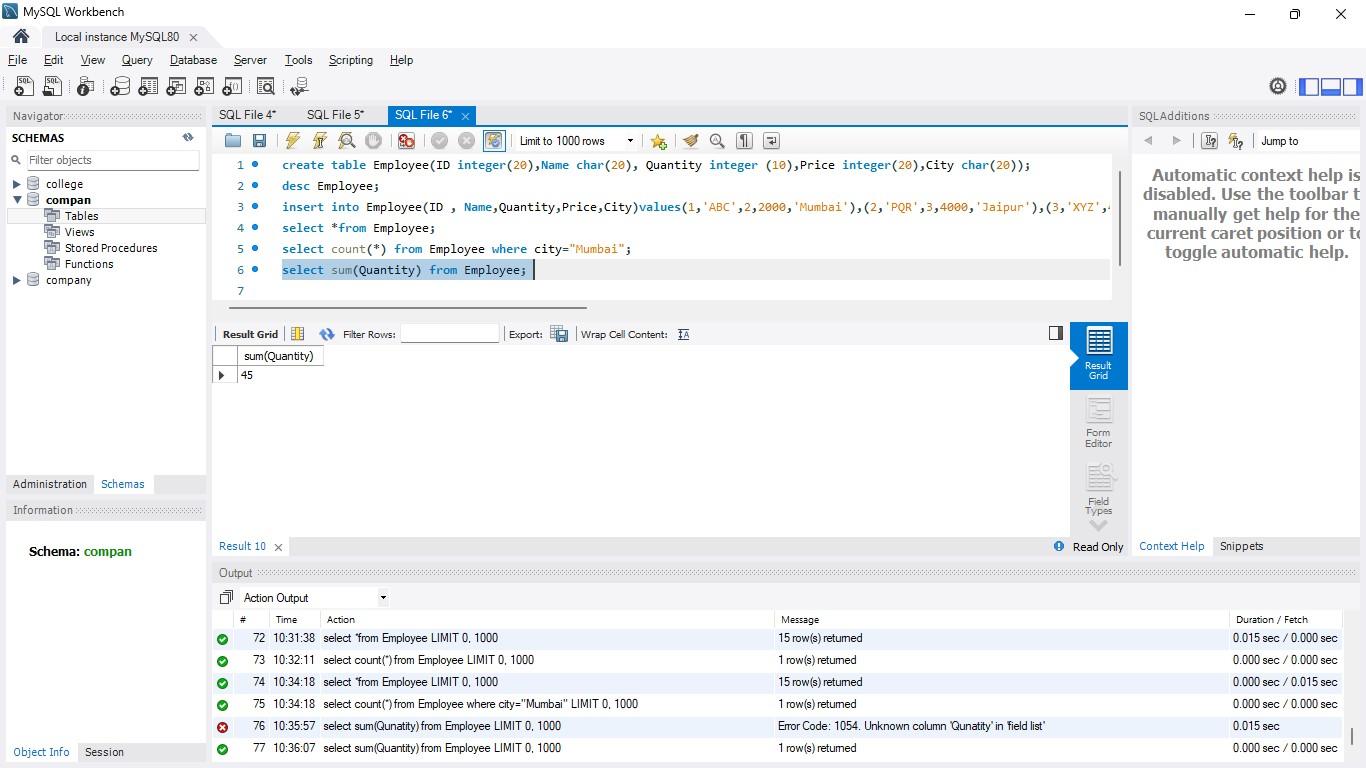
Trim :  


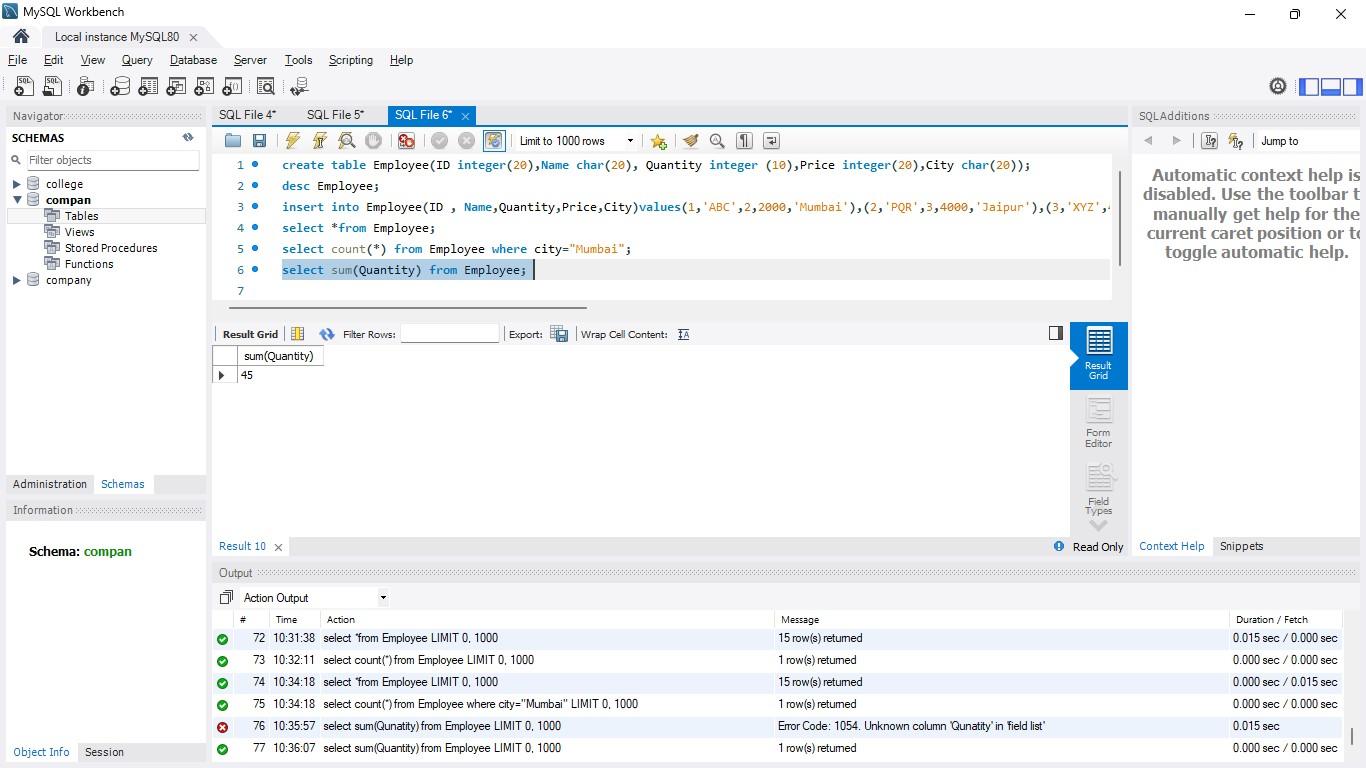
Upper and Lower Case:



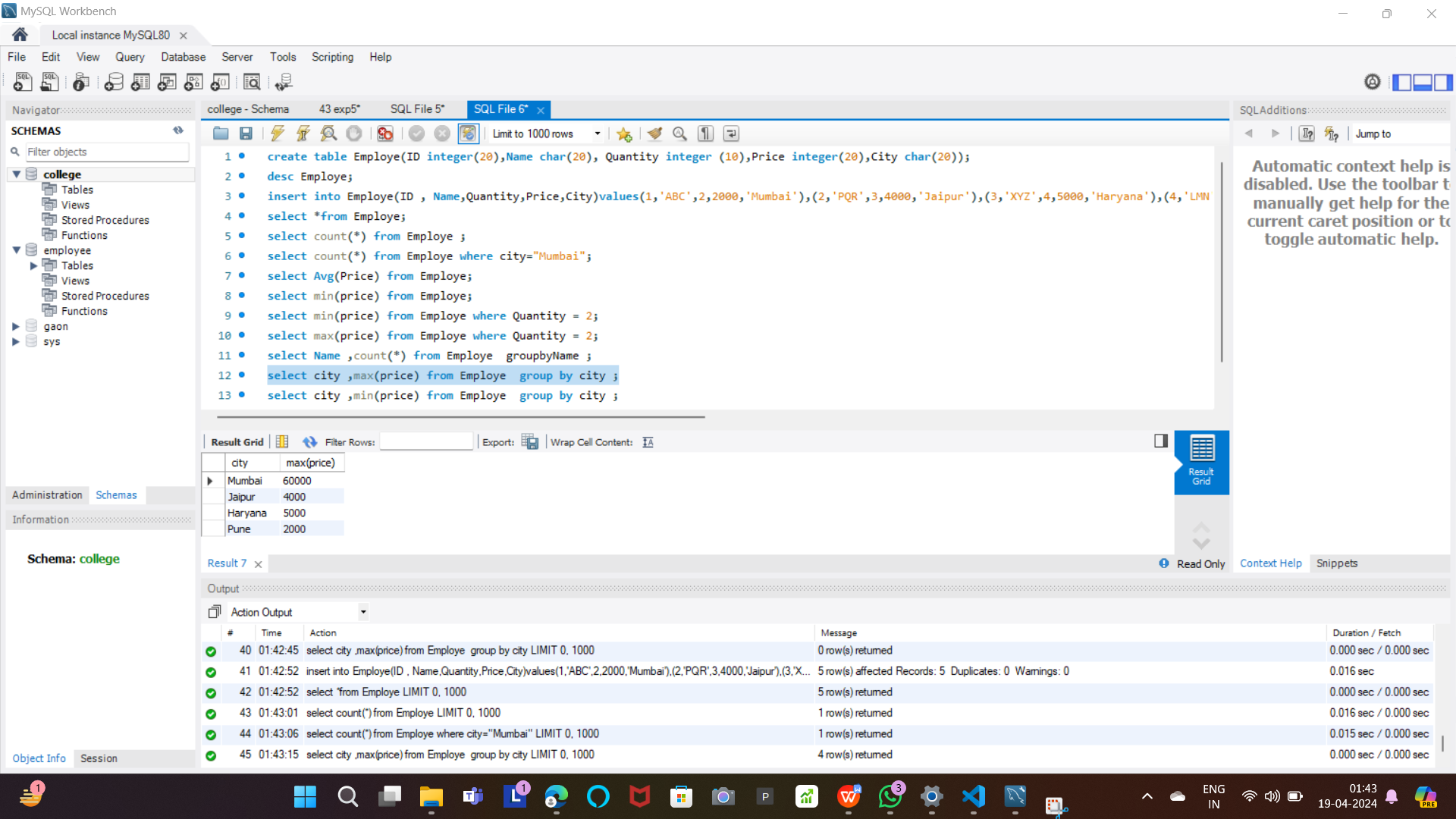
***Aggregate Function:***

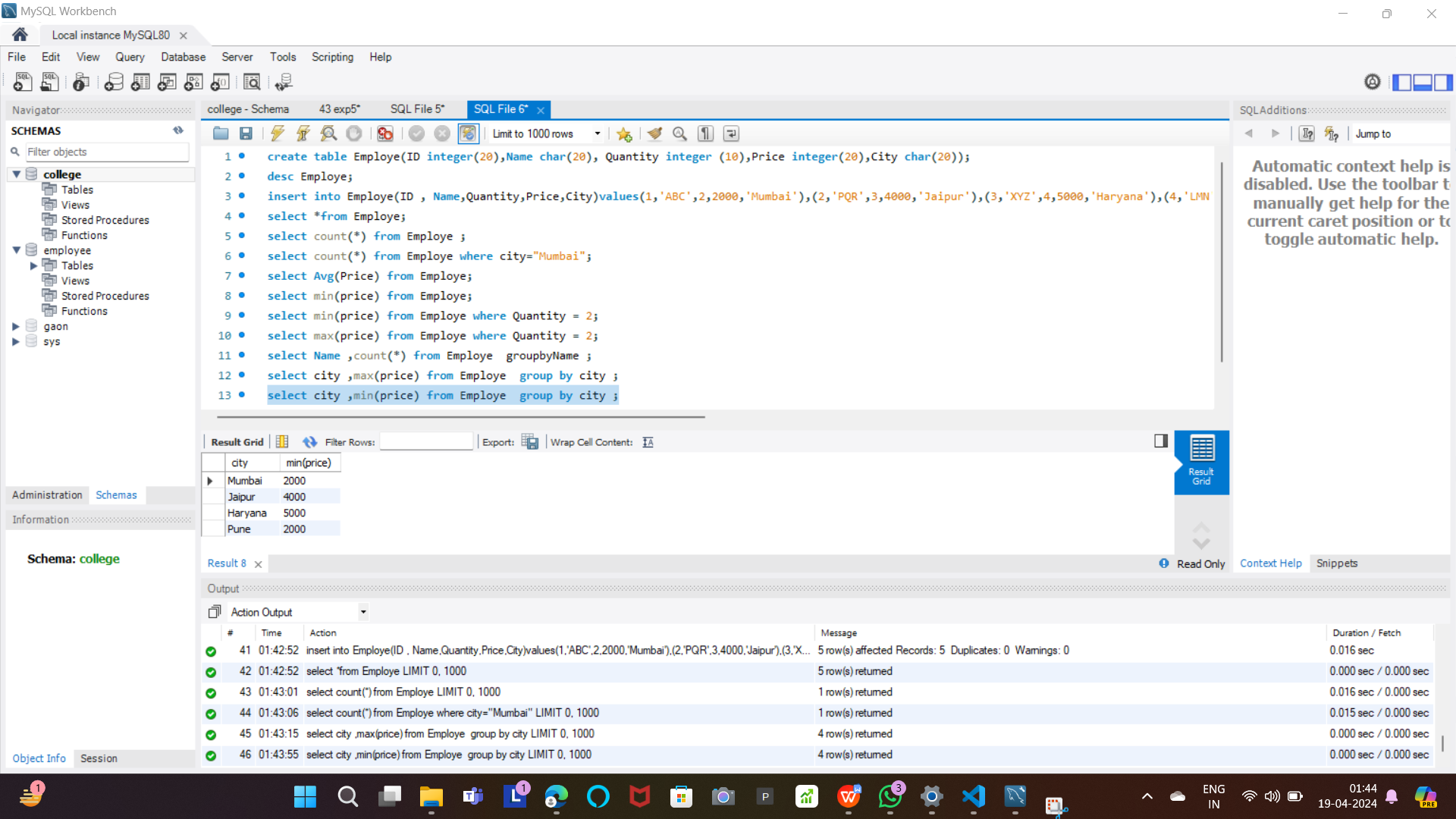
Sum Operation:



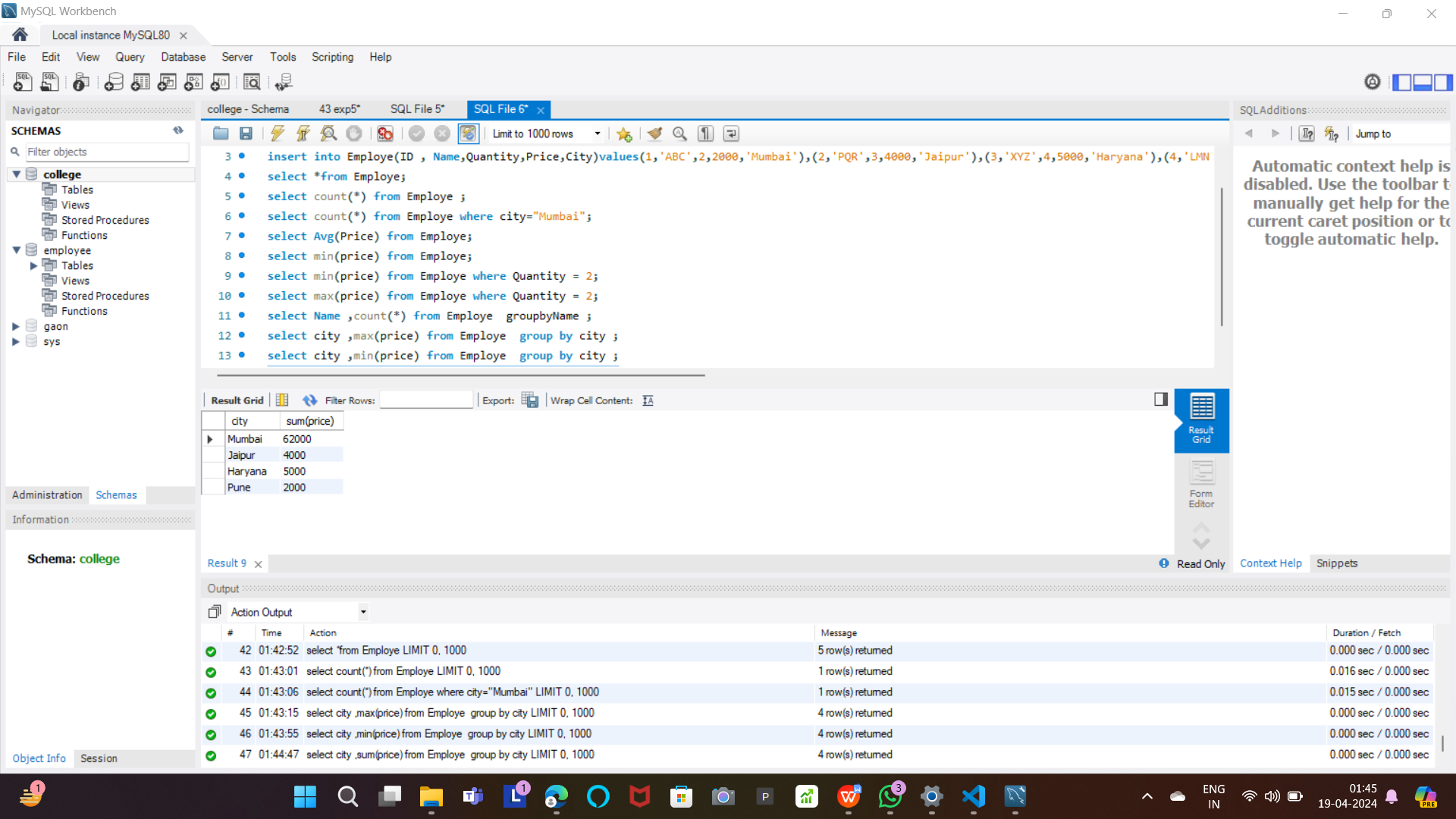


Minimum and Maximum:





***Group By:***



**Conclusion:**

After performing Experiment No. 5, it's evident that SQL queries and string manipulation operations are essential tools for data retrieval, transformation, and analysis. Through the application of SQL concepts such as Union, Intersect, Minus, ASCII Value retrieval, padding, trimming, repetition, replacement, and string reversal, along with functions like SUBSTRING, TRIM, UPPER, LOWER, and LENGTH, we can efficiently manipulate and extract valuable information from databases. These operations enable us to perform various data processing tasks, including data cleaning, formatting, and transformation, enhancing the usability and integrity of the stored data. Overall, this experiment underscores the significance of SQL and string manipulation techniques in database management and data-driven decision-making processes.