Assignment-6

Data cleaning and Transformation

SQL offers a wide variety of techniques for cleaning and transforming data efficiently.

Some of the techniques include the following:

- Check for Missing Values
- Check for Duplicates
- Standardizing and Transforming Data
- Updating Data / Column Data Types

1) **Check for Missing values**
   SQL provides powerful functions and techniques for handling missing values.

   Ex: **IS NULL**: It is used to indicate missing values in a column.

   COALESCE: It is used to replace missing values with more suitable values.

2) **Check for Duplicates**
   Duplicate records can distort analysis results and skew decision-making. SQL offers efficient ways to identify and remove duplicates.

   Ex: **GROUP BY and HAVING**: They are used to find duplicate records.

   DISTINCT: It eliminates duplicate rows from the result set.

   DELETE FROM: It removes certain rows from the result set.

3) **Standardizing and Transforming Data**
   Data standardization and transformation are crucial for consistency and accurate analysis. SQL provides functions and expressions for data manipulation.

   Ex: **UPPER or LOWER:** They convert a string to uppercase or lowercase.

   REPLACE: It is used to replace occurrences of a substring in a string with a new substring.

   SUBSTRING: It returns specified portions of text.

   TRIM: It removes leading and trailing spaces from a string.

   DATE_FORMAT: You can format a date/time value into a specific format using this.

   CASE WHEN: It performs conditional data transformations. Similar to IF in Excel.

4) **Updating Data / Column Data Types**
   UPDATE: You can use the UPDATE statement to update existing data in a table.

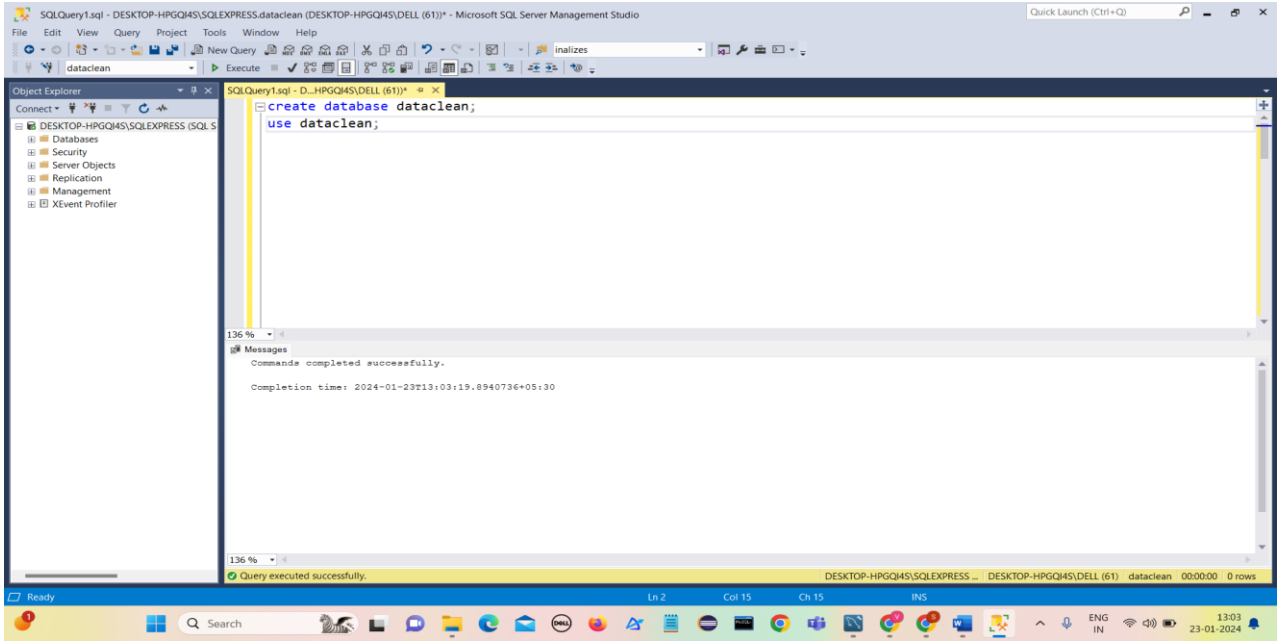
ALTER TABLE & MODIFY COLUMN: If you need to modify the data type of a column, you can use the ALTER TABLE statement with the MODIFY COLUMN clause.

ALTER TABLE & DROP COLUMN: To remove a column from a table, you can use the ALTER TABLE statement with the DROP COLUMN clause.

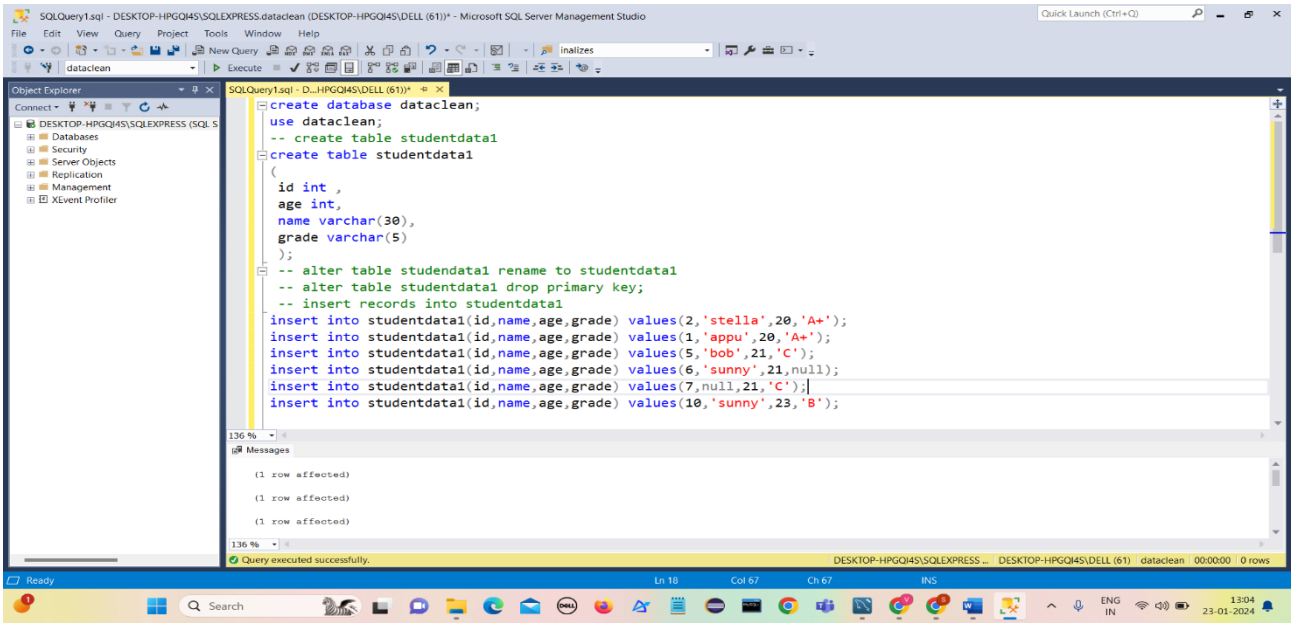
Example for data cleaning and transformation

create database dataclean

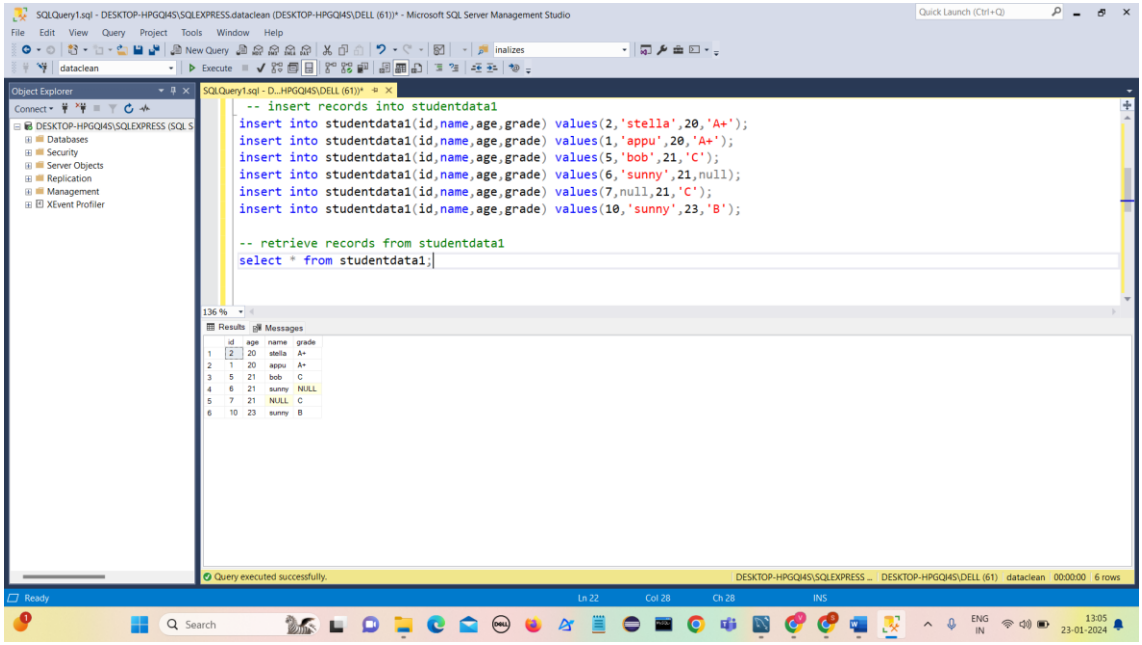
use dataclean to make it as current working database



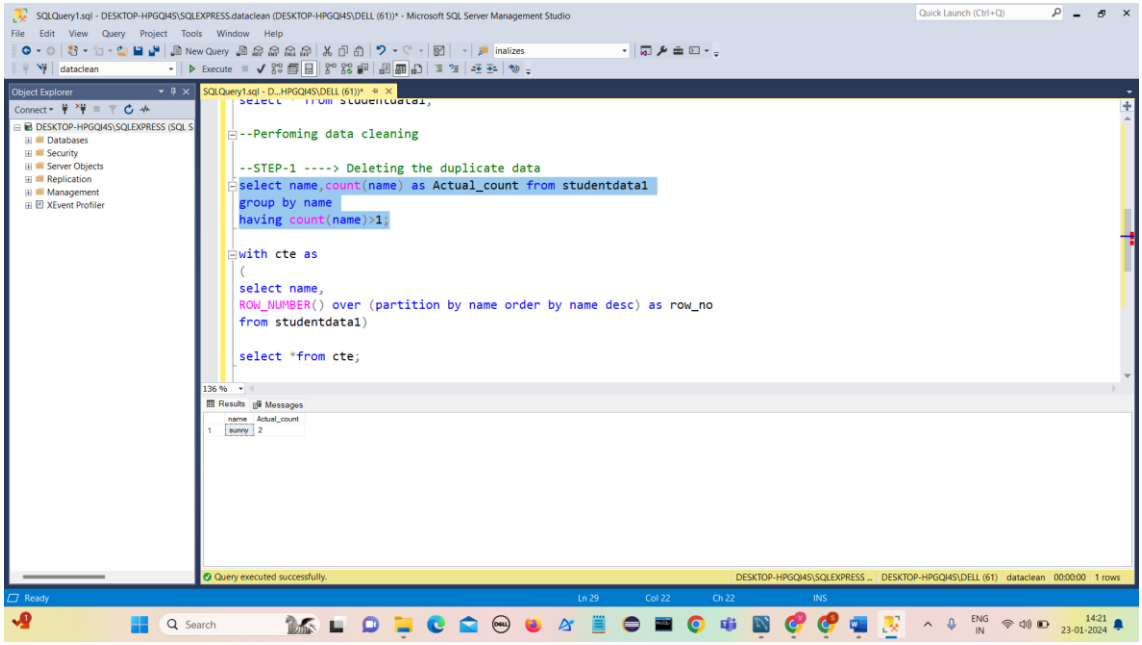
Create table studentdata1 and insert data into it



Retrieve studentdata1 records



Performing Data Cleaning

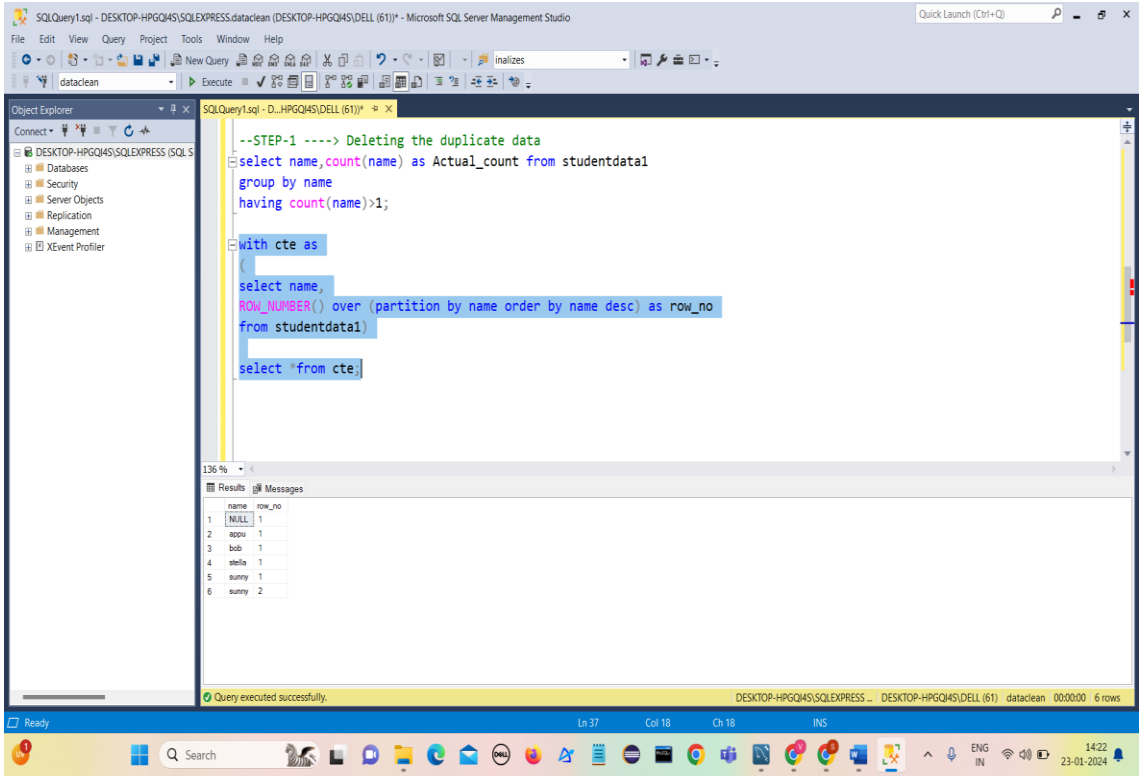


This query gives the name and count of each student.

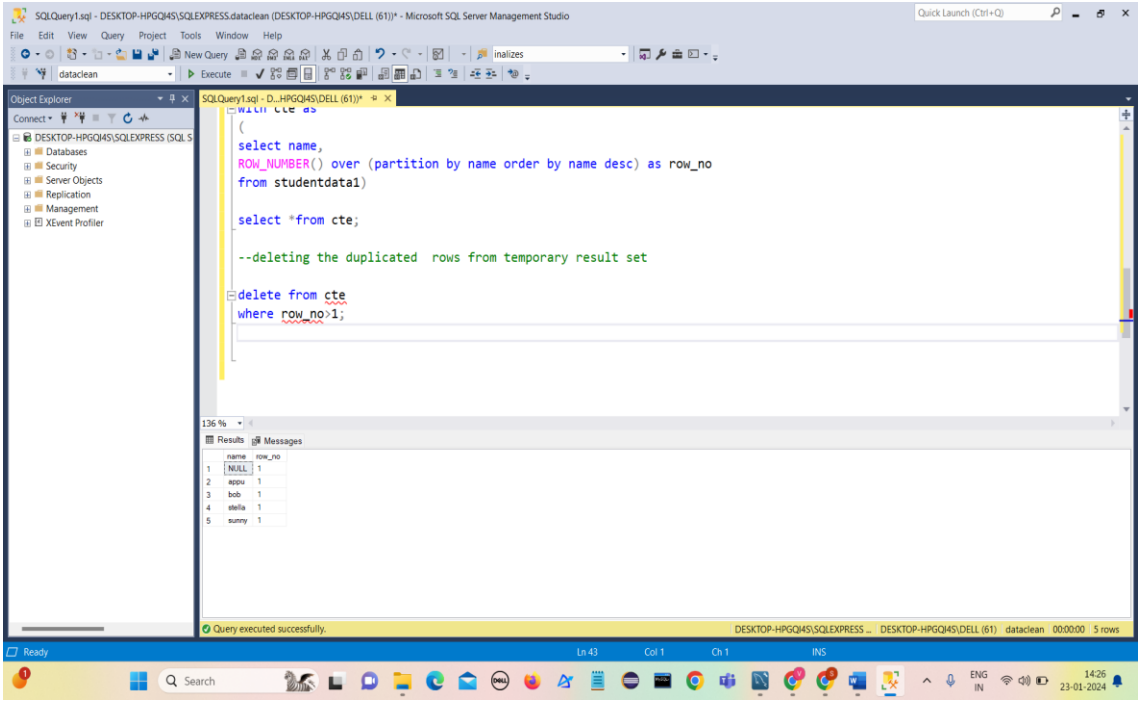
Step-1 Deleting duplicate data

Creating a Common Table Expression (CTE) named **cte** using the **ROW_NUMBER()** window function over the studentdata1 table.

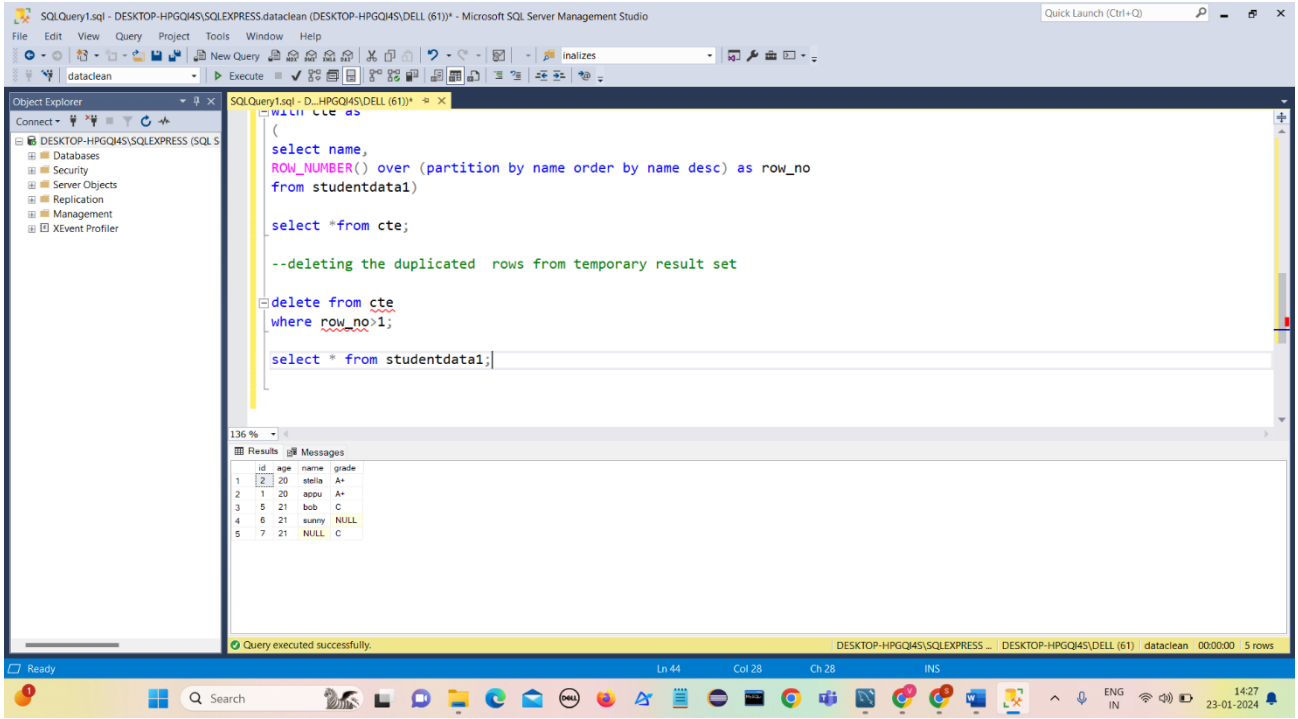
The **ROW_NUMBER()** function assigns a unique number to each row within a partition of the result set based on the specified ordering.



The row_no for sunny is 2 i.e., it is repeating and it is duplicate. We need to eliminate duplicate values.

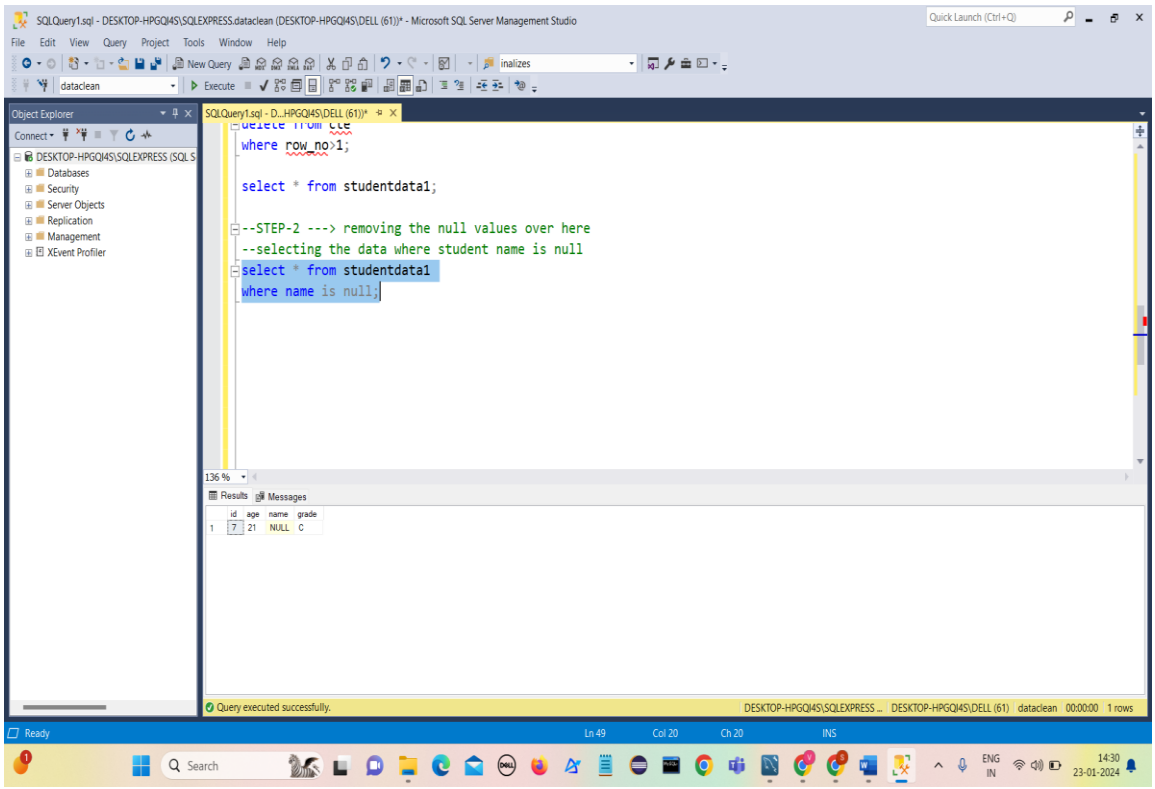


The value of sunny is 1.

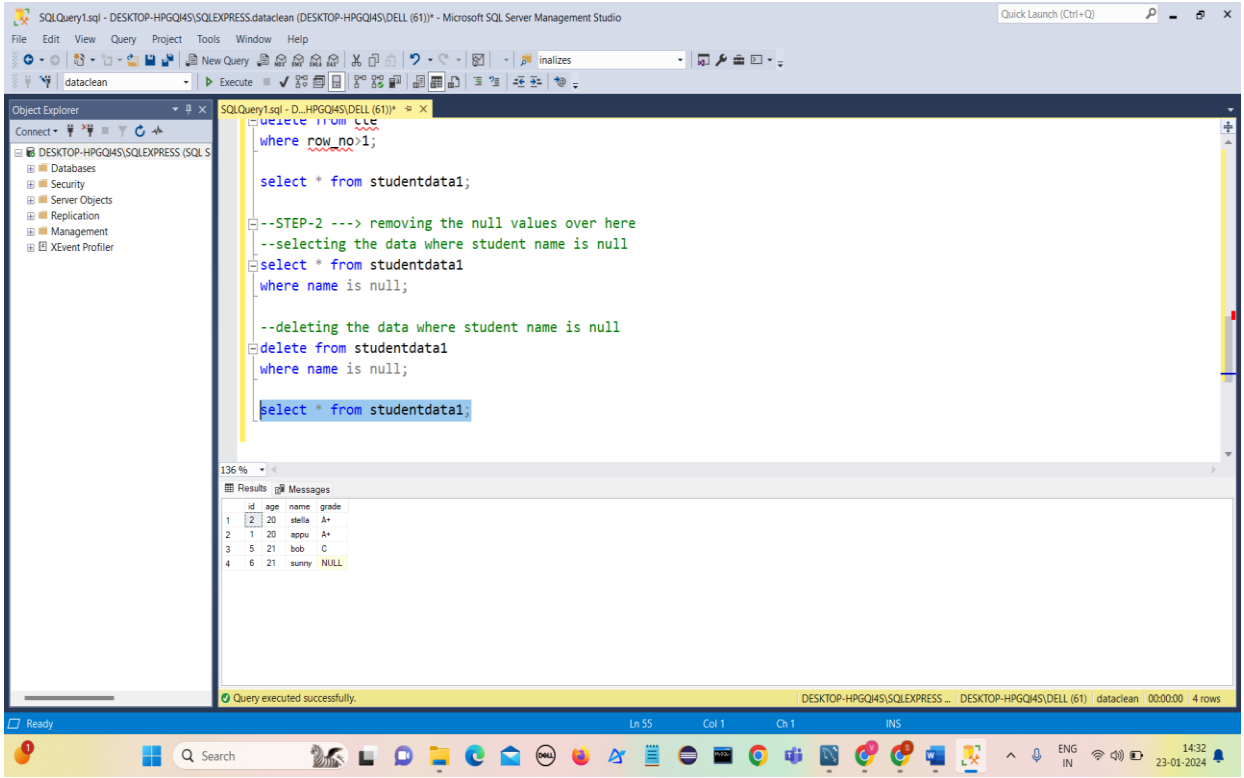


Duplicate values are removed.

Step-2: Removing null values

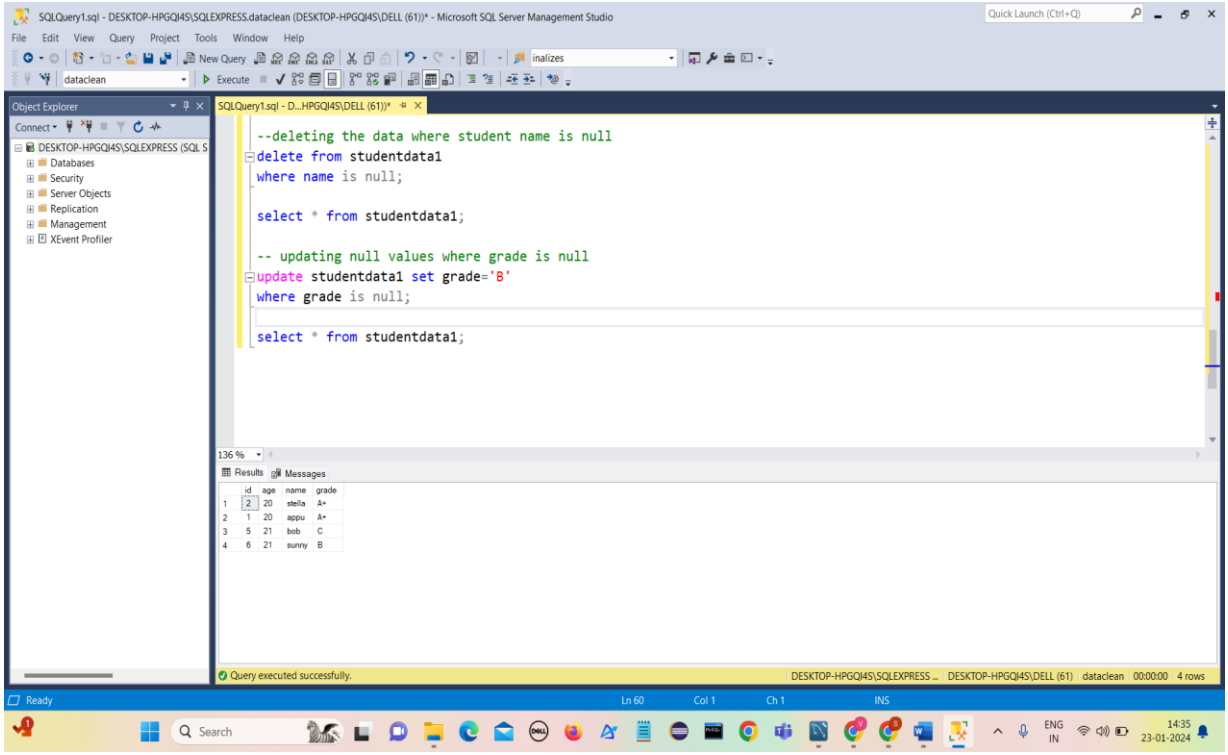


Here name is null for student with id 7. We have to eliminate null values



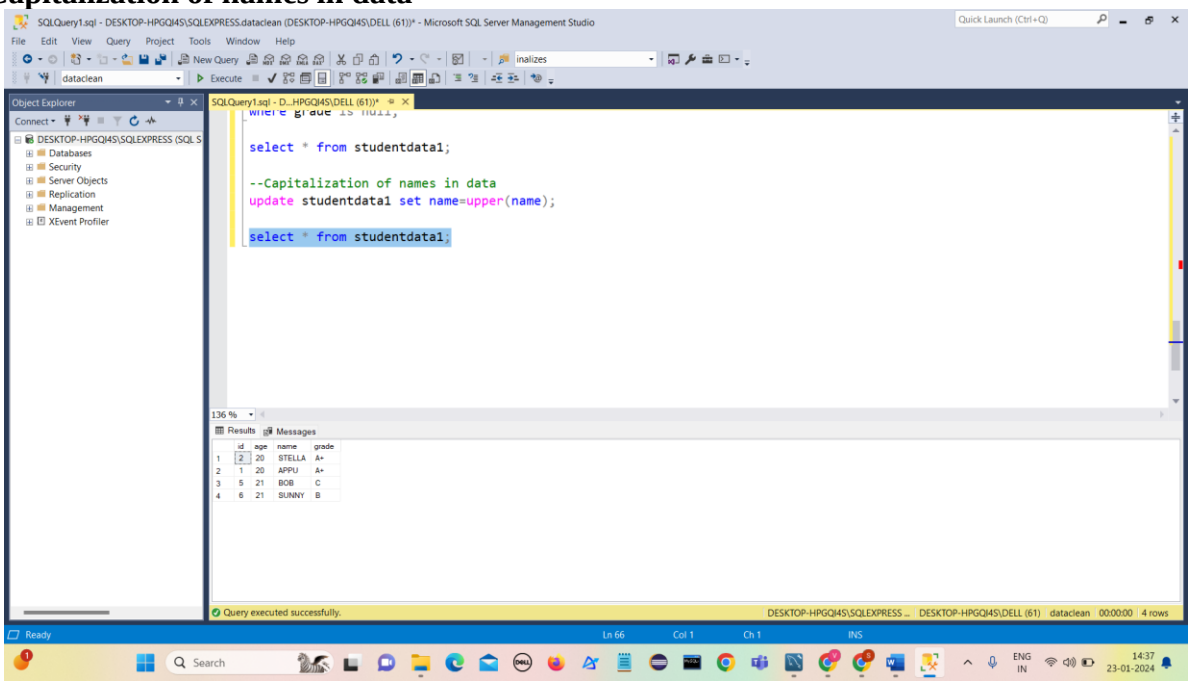
Student with id 7 is deleted.

Updating null values where grade is null



Hence grade is updated.

Capitalization of names in data



RANK() FUNCTION:

The **RANK() function** is a window function could be used in SQL Server to calculate a rank for each row within a partition of a result set.

SQL Sever provides SQL RANK functions to specify rank for individual fields as per the categorizations.

Rank Functions are:

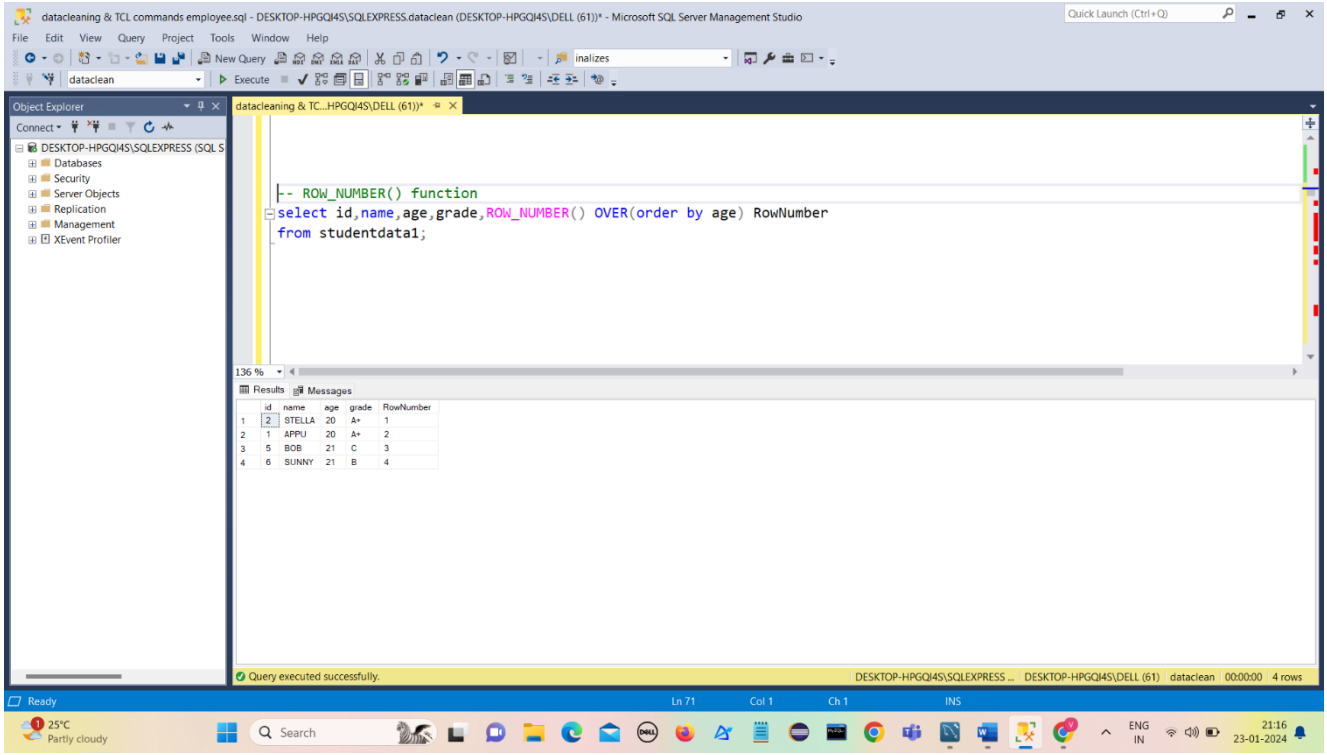
1) **ROW_NUMBER()**
2) **RANK()**
3) **DENSE_RANK()**
4) **NTILE()**

1) ROW_NUMBER():

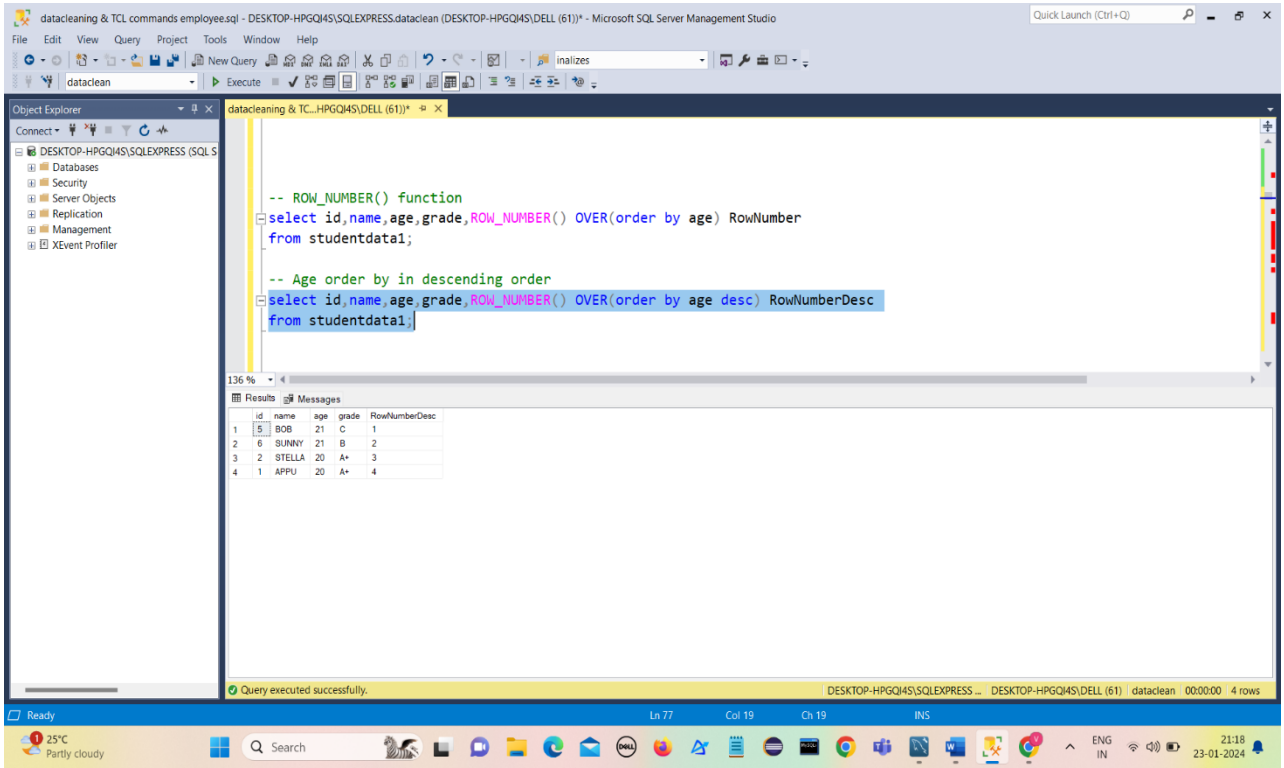
i) OVER() clause - define a set of rows in the result set.

ii) ROW_Number() SQL RANK function gives a unique sequential number for each row in the specified data.

iii) It gives the rank one for the first row and then increments the value by one for each row. We get different ranks for the row having similar values as well.



Age order by in descending order



2) RANK():

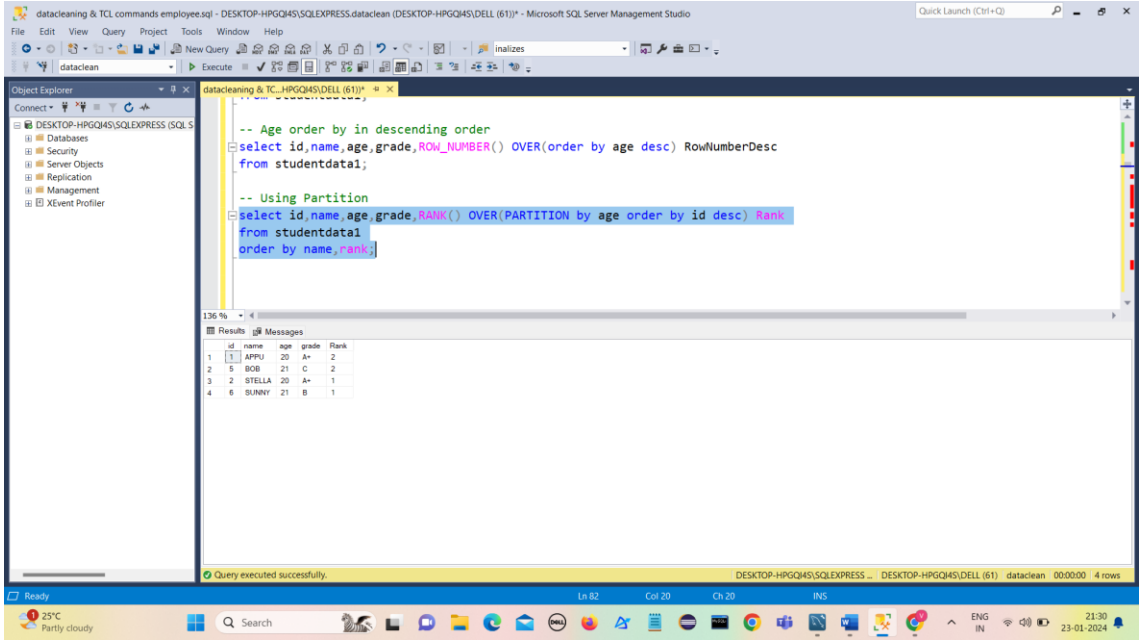
RANK() SQL Rank function to specify rank for each row in the result set.

Using Partition: (divide data into smaller subset)

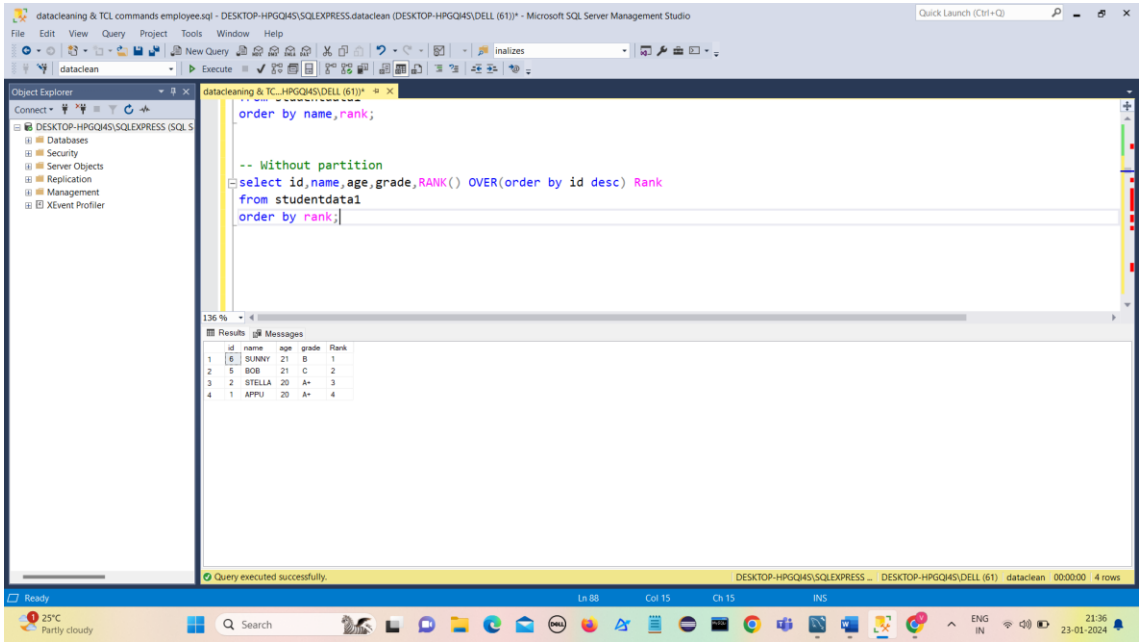
Use partition by age

Each subset should get rank as per id in descending order

Uses order by clause to sort based on name and rank.



Without partitioning

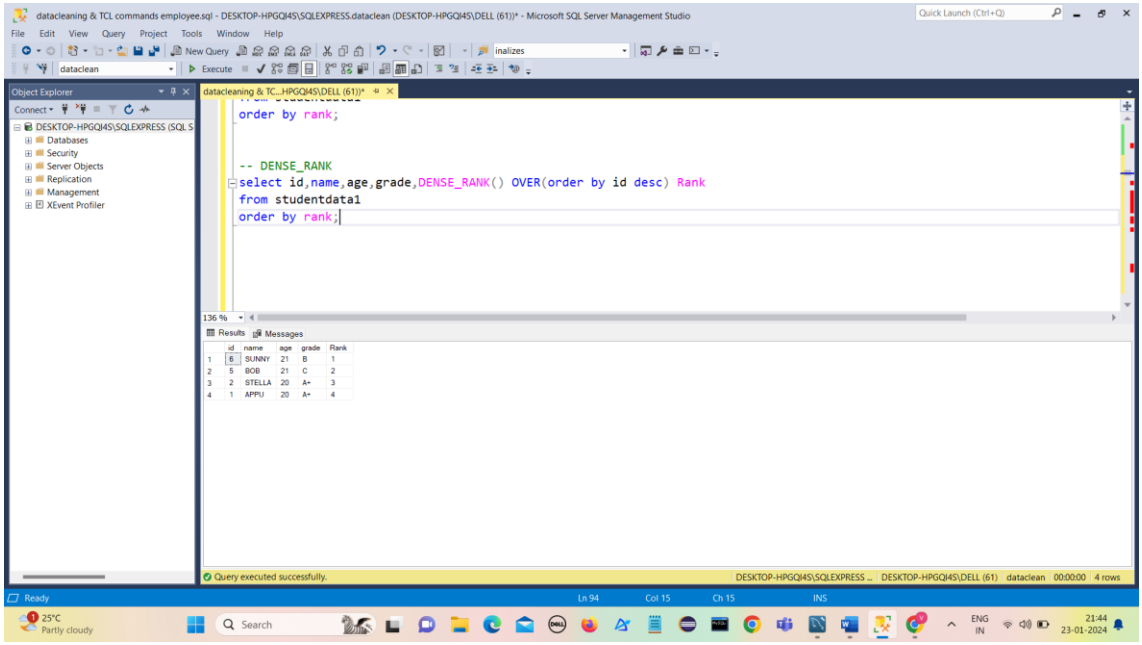


If ids are same then same rank is assigned.

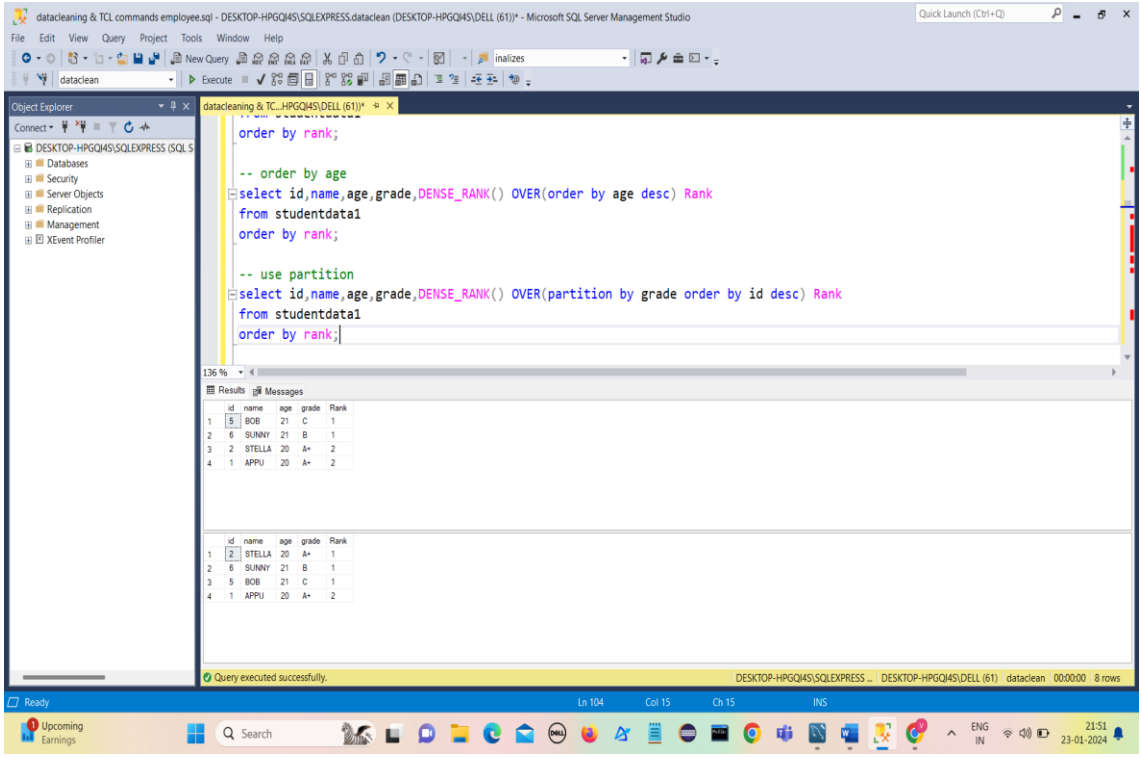
3) DENSE_RANK()

DENSE_RANK() function to specify a unique rank number within the partition as per the specified column value.

In the SQL RANK function DENSE_RANK(), if we have duplicate values, SQL assigns different ranks to those rows as well.



Using partition



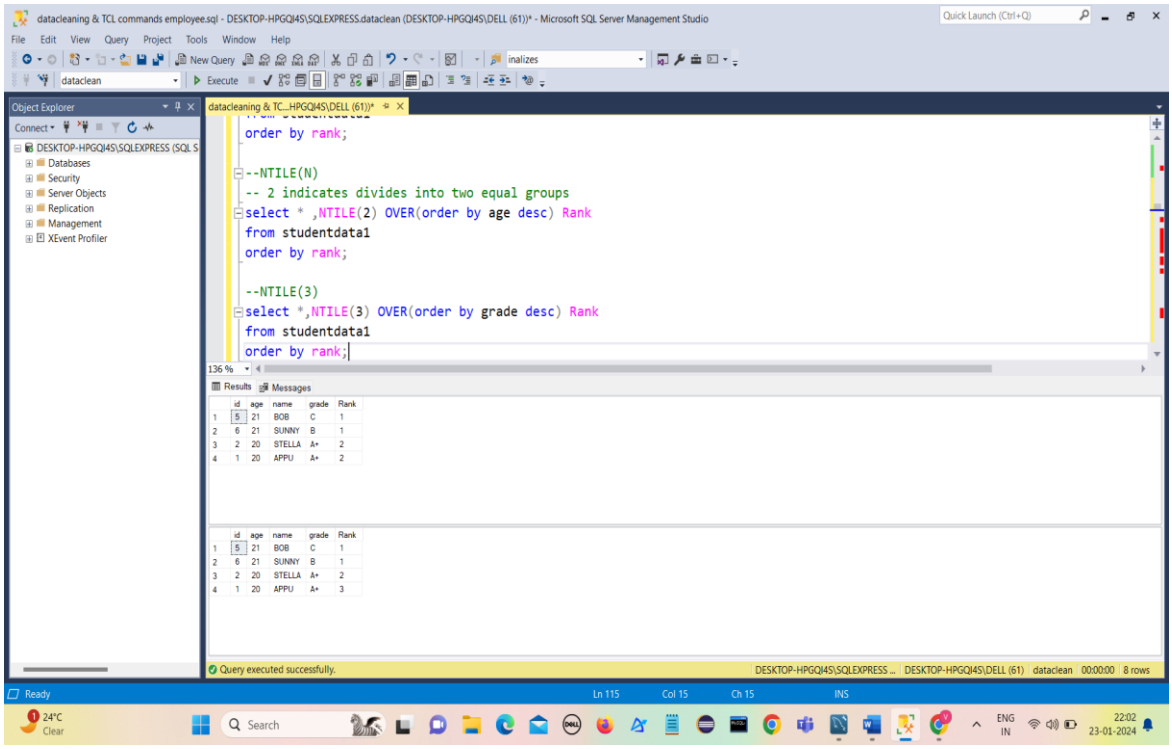
4) NTILE(N) RANK FUNCTION:

NTILE(N) function to distribute the number of rows in the specified (N) number of groups.

Each row group gets its rank as per the specified condition.

We need to specify the value for the desired number of groups.

NTILE(2)-means divides the table into two groups



STORED PROCEDURE:

An SQL **stored procedure** is a group of pre-compiled SQL statements (prepared SQL code) that can be reused by simply calling it whenever needed.

Syntax:

CREATE PROCEDURE *procedure_name*

AS

BEGIN

sql_statement

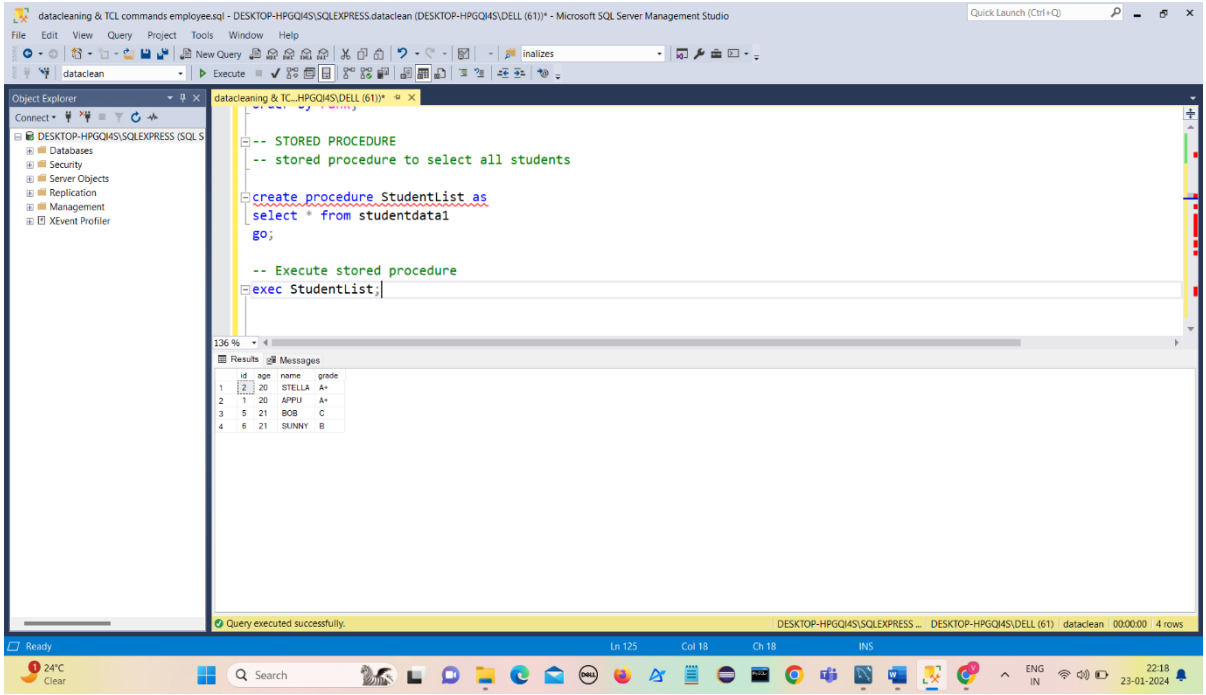
END

GO;

Execute a Stored Procedure

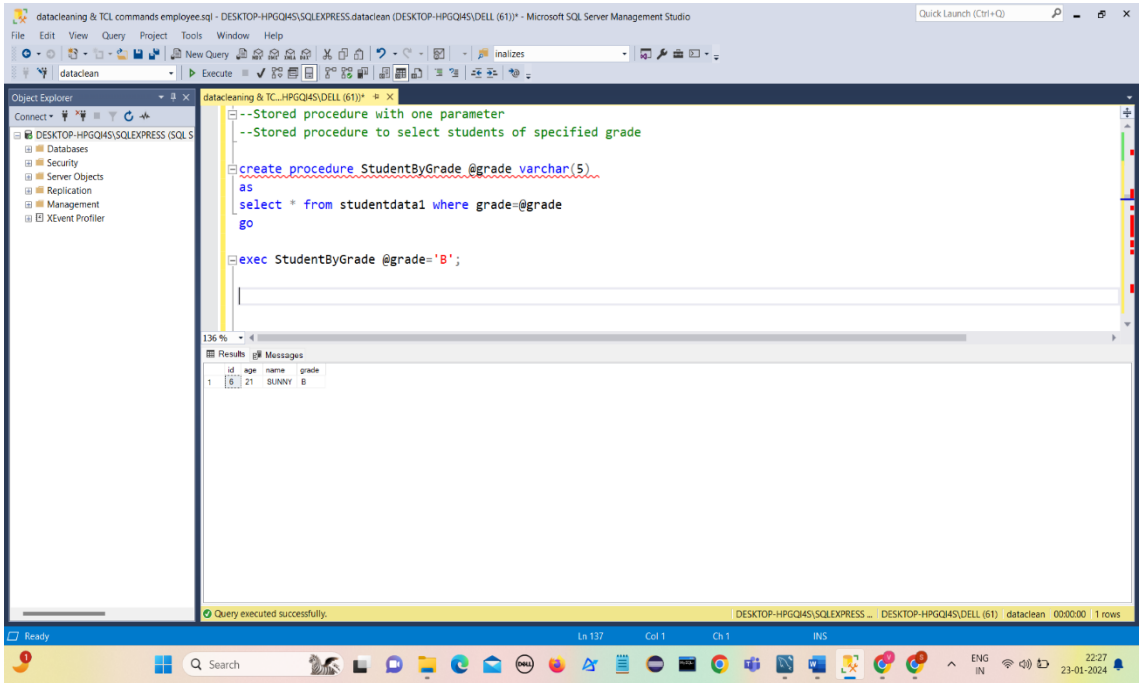
EXEC *procedure_name*;

1) Stored procedure to select all students



2) **Stored procedure with one parameter**

Stored procedure to select students of specified grade.



3) **Stored procedure with multiple parameters.**

Create stored procedure to get students of specific age and id.