

**Academic Year: 2022**

**Semester: 3<sup>rd</sup>**

**Class: BBIT**

**Course Title: Database Systems**

# TABLE OF CONTENTS

## Table of Contents

<b>Installation of SQL Server .....</b>	<b>4</b>
Download Setup: .....	4
Installation: .....	4
<b>Creating Database tables and Inserting Data .....</b>	<b>19</b>
Creating Database .....	19
Creating Table .....	21
Dropping Table .....	22
Setting Primary Key .....	22
Inserting Data .....	22
<b>Retrieving and Updating Data .....</b>	<b>24</b>
Retrieving Data .....	24
Updating Data .....	25
Setting Foreign Key .....	26
<b>Deleting, Dropping and Altering Tables .....</b>	<b>27</b>
Delete Data .....	27
Drop Table: .....	28
.....	29
Alter Table .....	29
<b>Aliasing .....</b>	<b>32</b>
ALIASING TABLES AND COLUMNS .....	32
ALIASING COLUMNS .....	32
ALIASING TABLES .....	33
<b>Clauses .....</b>	<b>35</b>
Group By Clause .....	35
Order By Clause: .....	36
Having Clause: .....	37
<b>Aggregation .....</b>	<b>39</b>
Aggregation: .....	39
AVG function: .....	39
Count function: .....	40
Max function: .....	41

Min function:.....	42
Sum function: .....	43
<b>Connecting SQL and C#.....</b>	<b>45</b>
Import sql package: .....	45
Get Connection URL.....	46
Establish Connection With DataBase .....	46
Prepare Command .....	47
Execute Query .....	47
Processing the result using sql Data Reader.....	47
Close the Connection .....	48
<b>Joins.....</b>	<b>49</b>
Joins:.....	49
Inner Join.....	49
Left Join: .....	50
Right Join: .....	51
Full outer Join.....	52
<b>Subqueries .....</b>	<b>54</b>
Subquery.....	54
<b>Views .....</b>	<b>56</b>
Views:.....	56
Retrieve data from View .....	57
Drop Views: .....	58
Update Views: .....	59
<b>Project A .....</b>	<b>65</b>
<b>Project B .....</b>	<b>66</b>

## ACKNOWLEDGEMENT

*We would like to express our special gratitude to our teacher Dr. Awais Hassan who initiated the standard lab manual project for Database System. He gave us the golden opportunity to make lab manual which also helped us in doing a lot of research and we came across so many new things. We are really thankful to him. He worked very hard to make this manual, standard document for the students and teachers of the department.*

## CS-363 Database Systems Lab 01

Type of Lab: Open Ended

Weightage: 5%

### Installation of SQL Server

#### Objectives

Install SQL Server and Management Studio

#### Processing steps

##### Download Setup

Download MS SQL Server 2019 Developer Edition from the following link

<https://www.microsoft.com/en-us/sql-server/sql-server-downloads>

Download MS SQL Server Management Studio 18.4 from the following link

<https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?redirectedfrom=MSDN&view=sql-server-ver15>

#### Installation:

Open MS SQL Server 2019 Developer Edition and select ISO Package

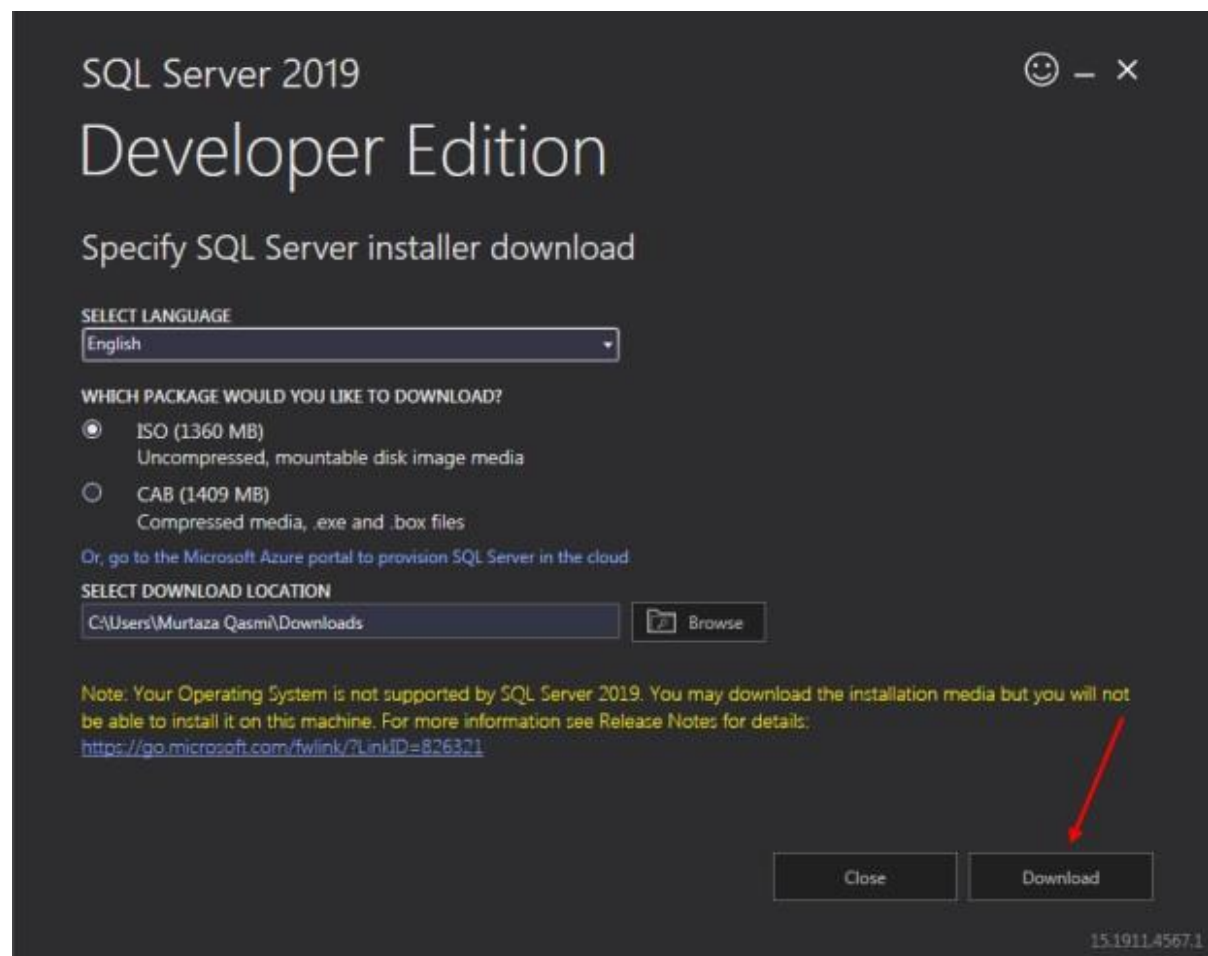
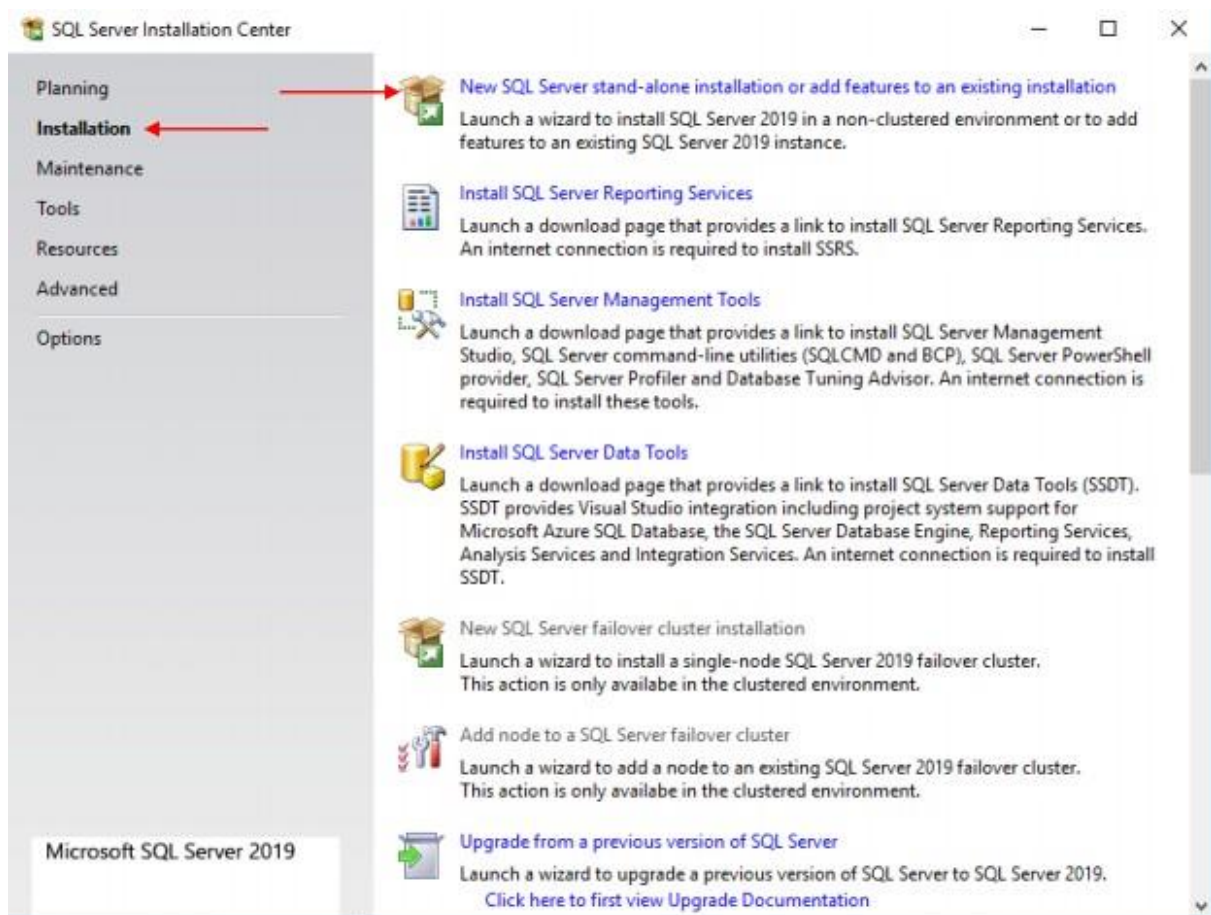


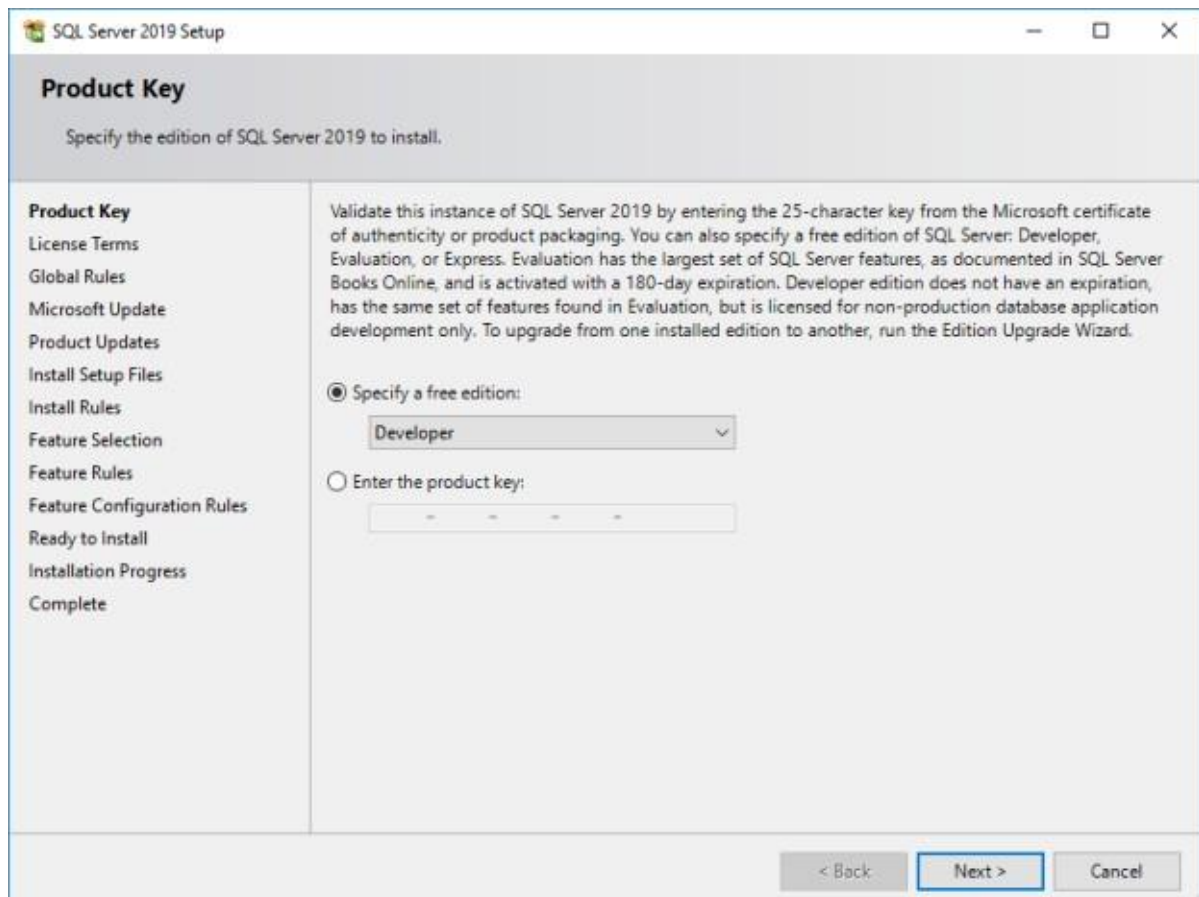
Figure 1 Install SQL Server

Click on the Installation page and then on 'New SQL Server stand-alone installation or add features to an existing installation'



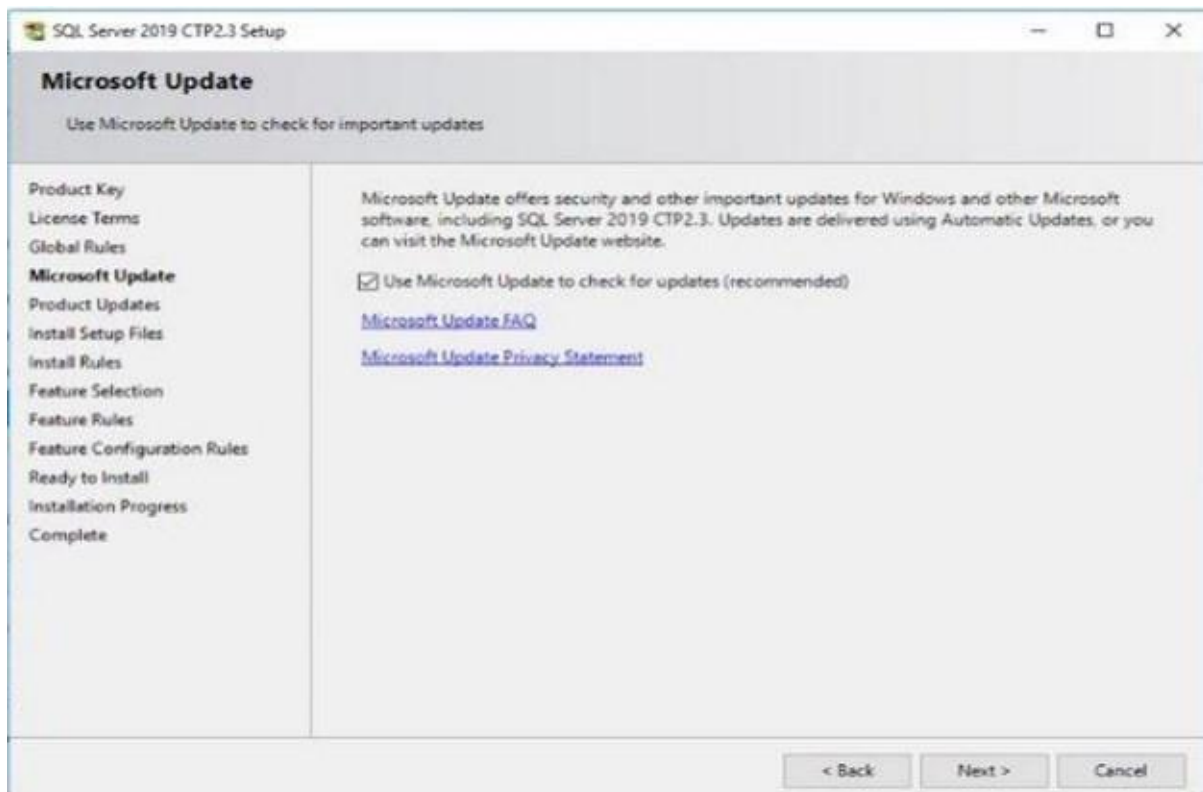
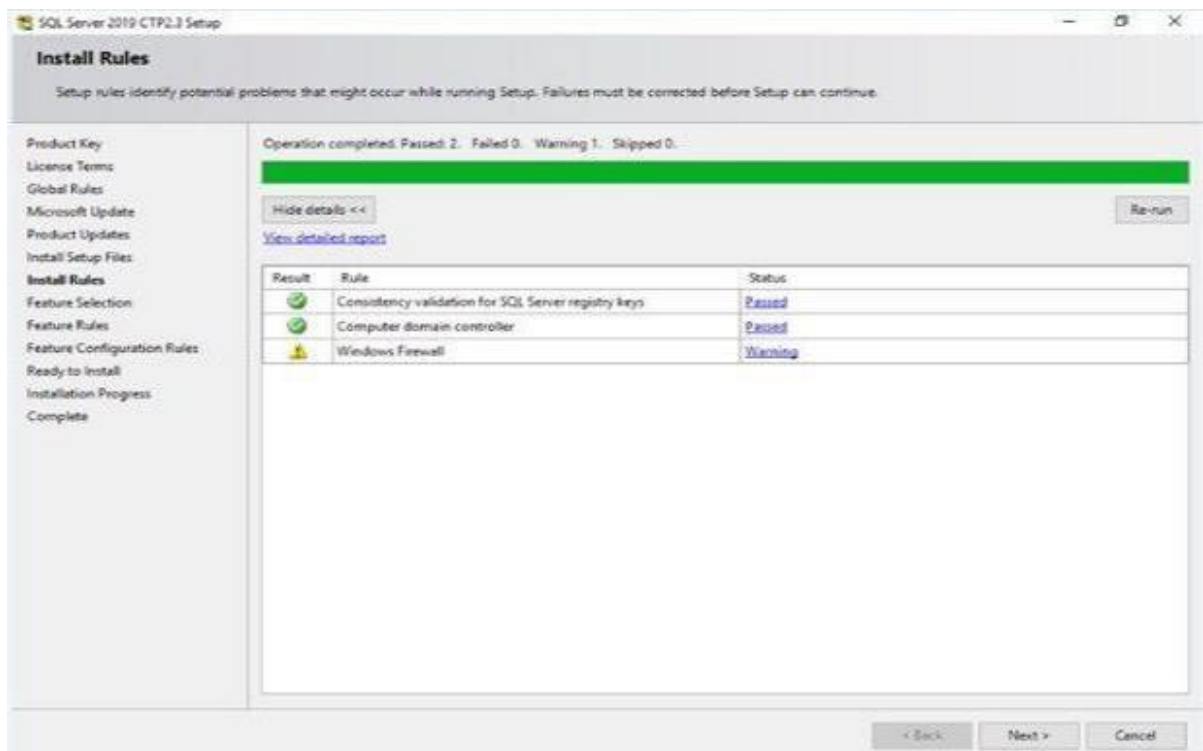
*Figure 2 Installing SQL Server*

Since we are installing Developer edition of SQL Server, we will move to the default option "Developer" and click next.



*Figure 3* Installing SQL Server

Accept the license terms and conditions. Please note the message written here that SQL Server Transmits information about installation experience in performance data to Microsoft to help improve the product. Click on Next. In the next page, it checks for the Global rules and give status information about successful, failed rules. If there are any failed rules, we need to fix those before proceeding with the installation. Next Check checkbox “use Microsoft update to check for updates”

*Figure 4 Installing SQL Server**Figure 5 Installing SQL Server*



Now we need to select the features, we need to install with this installation.

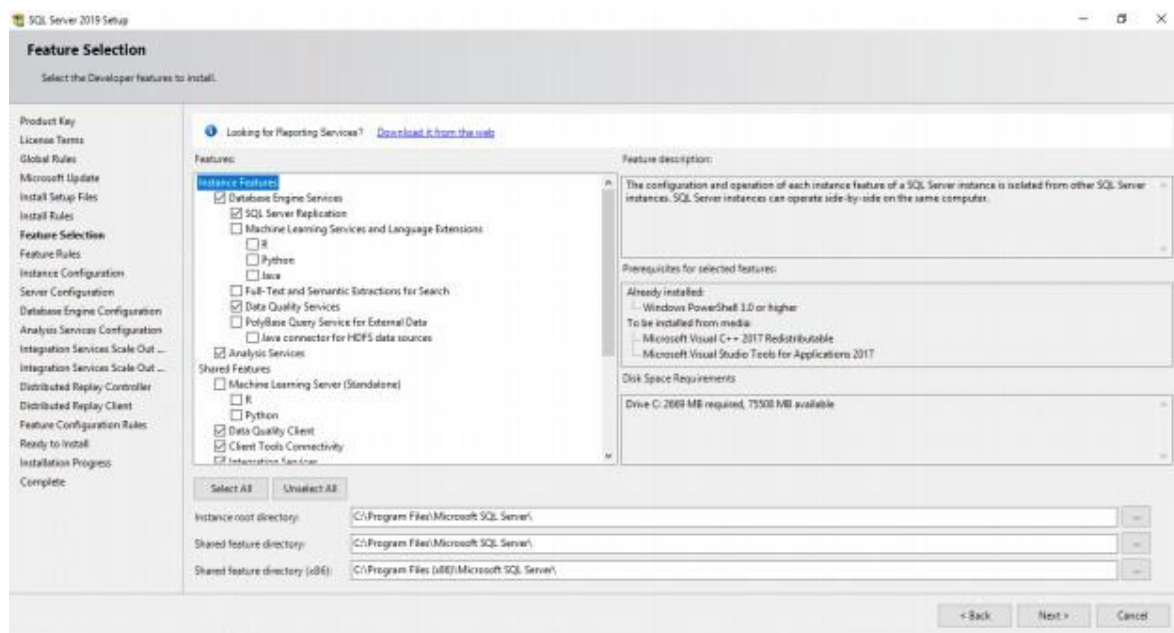


Figure 6 Installing SQL Server

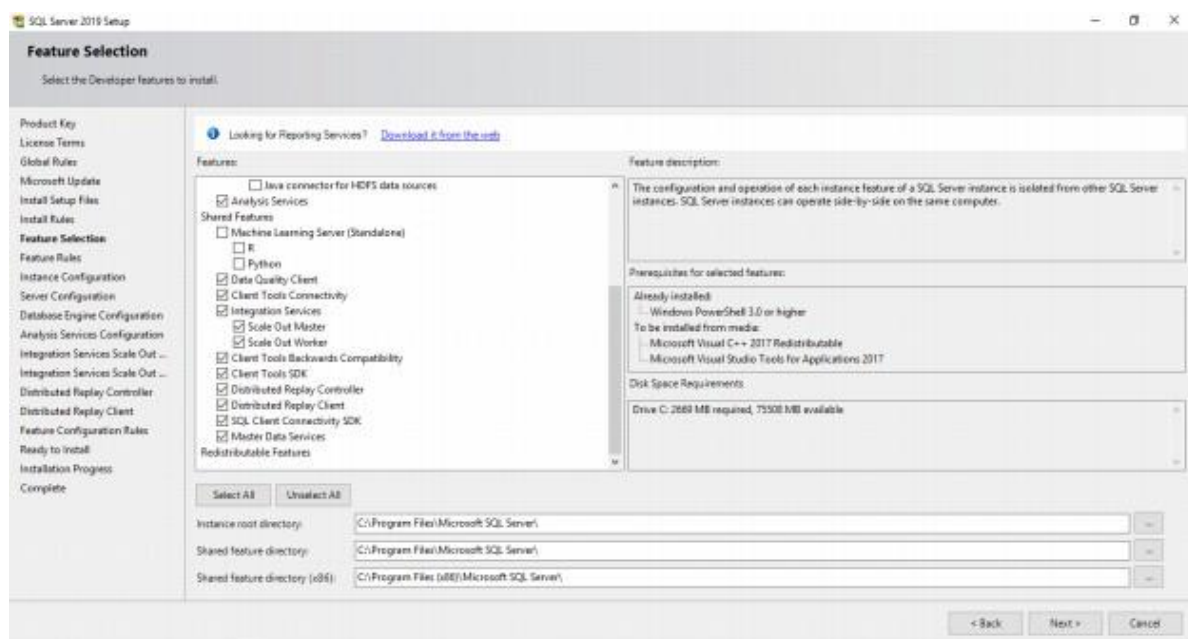


Figure 7 Installing SQL Server

You can choose to install a default or named instance name. I have selected default instance. If you want can only install named instance, Provide an appropriate name for the named instance.

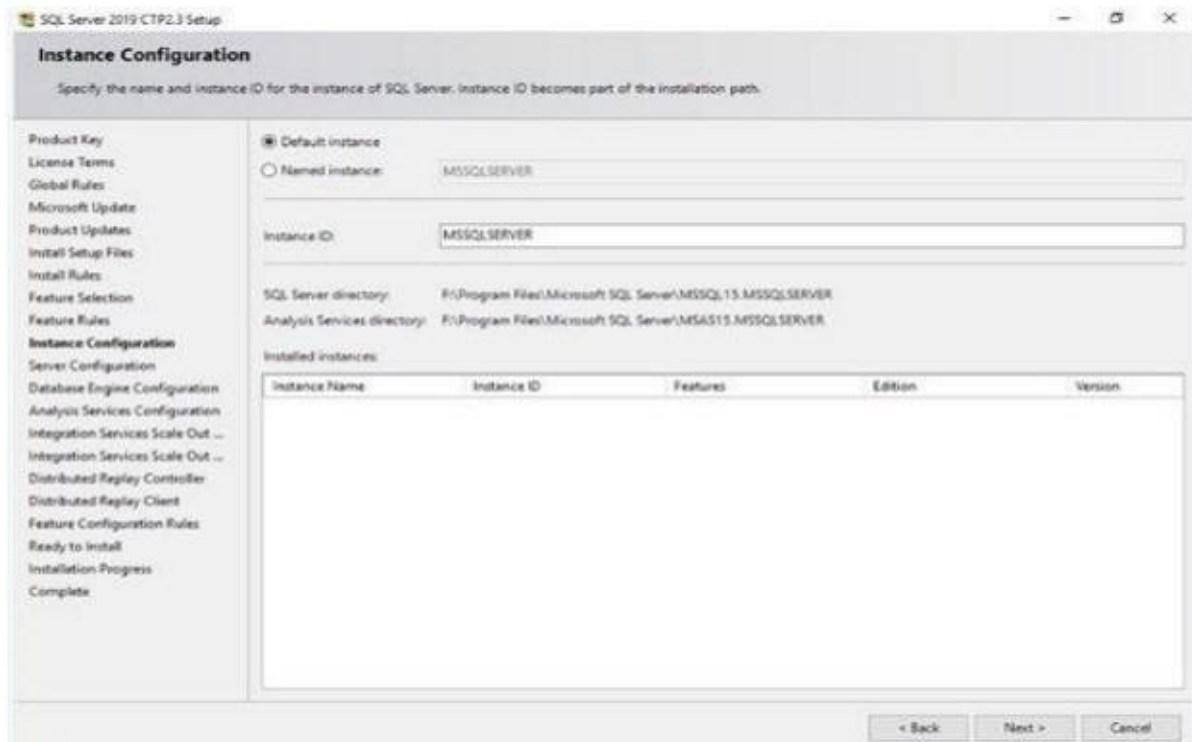


Figure 8 Installing SQL Server

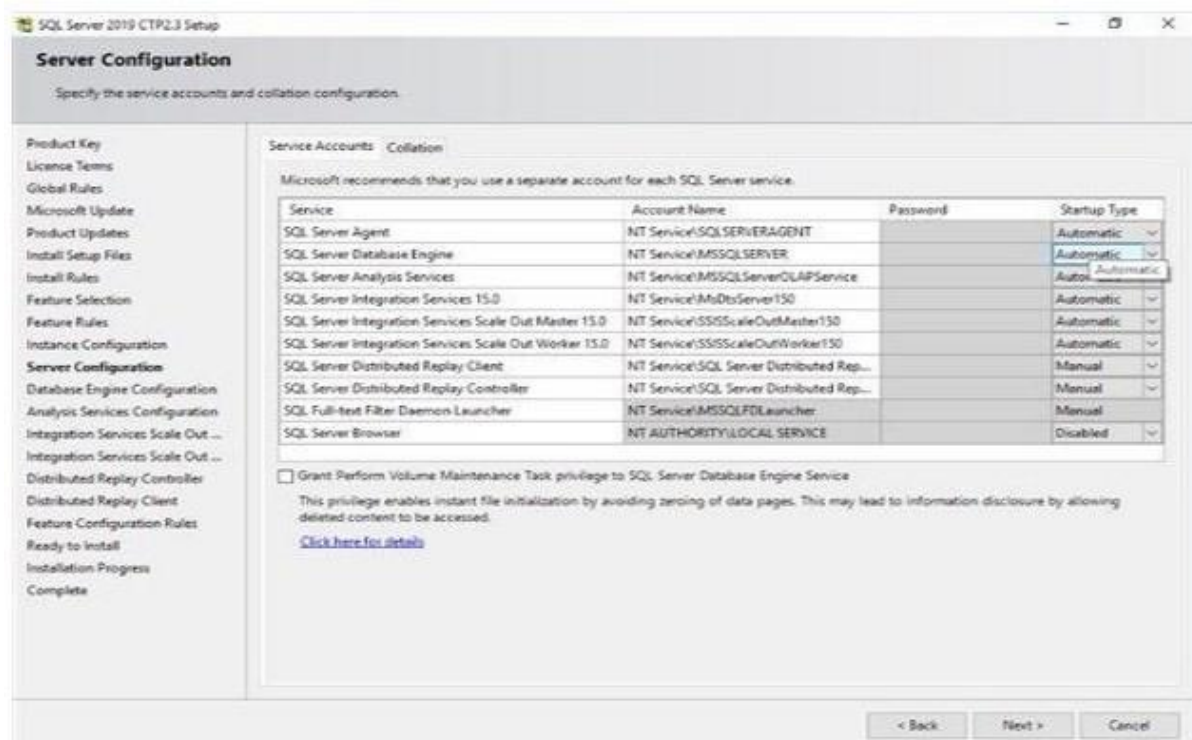


Figure 9 Installing SQL Server

Check properly about Collation.

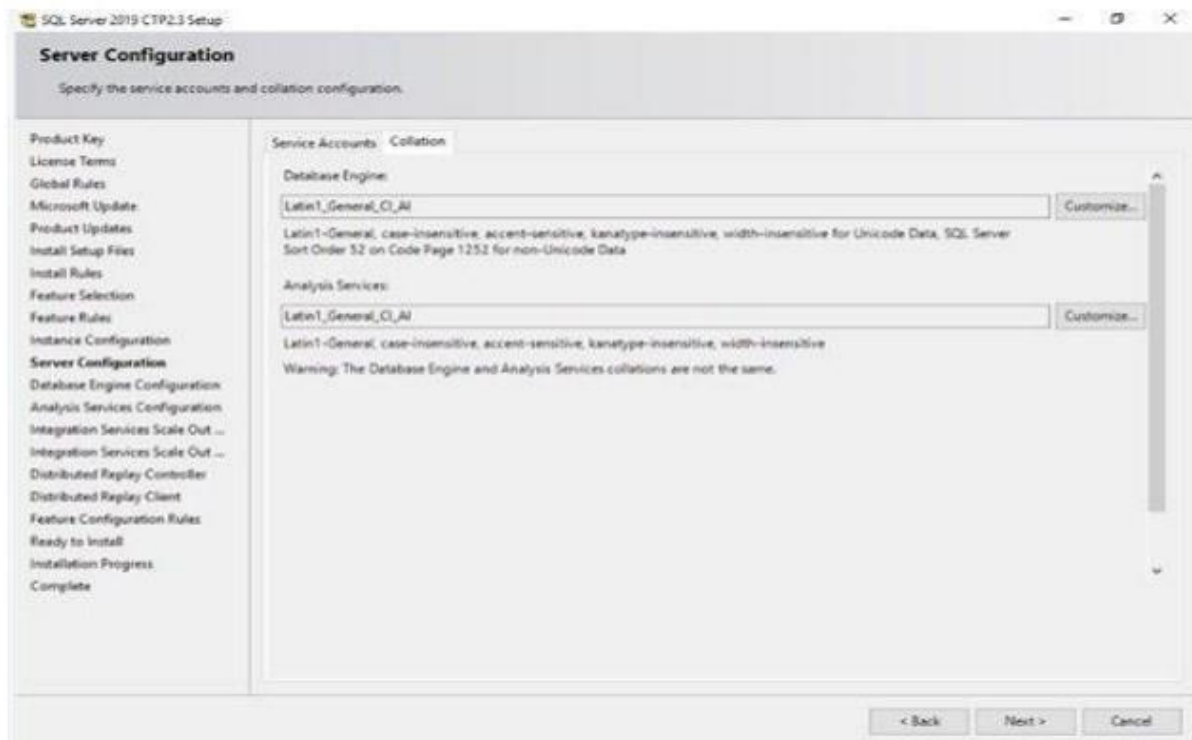
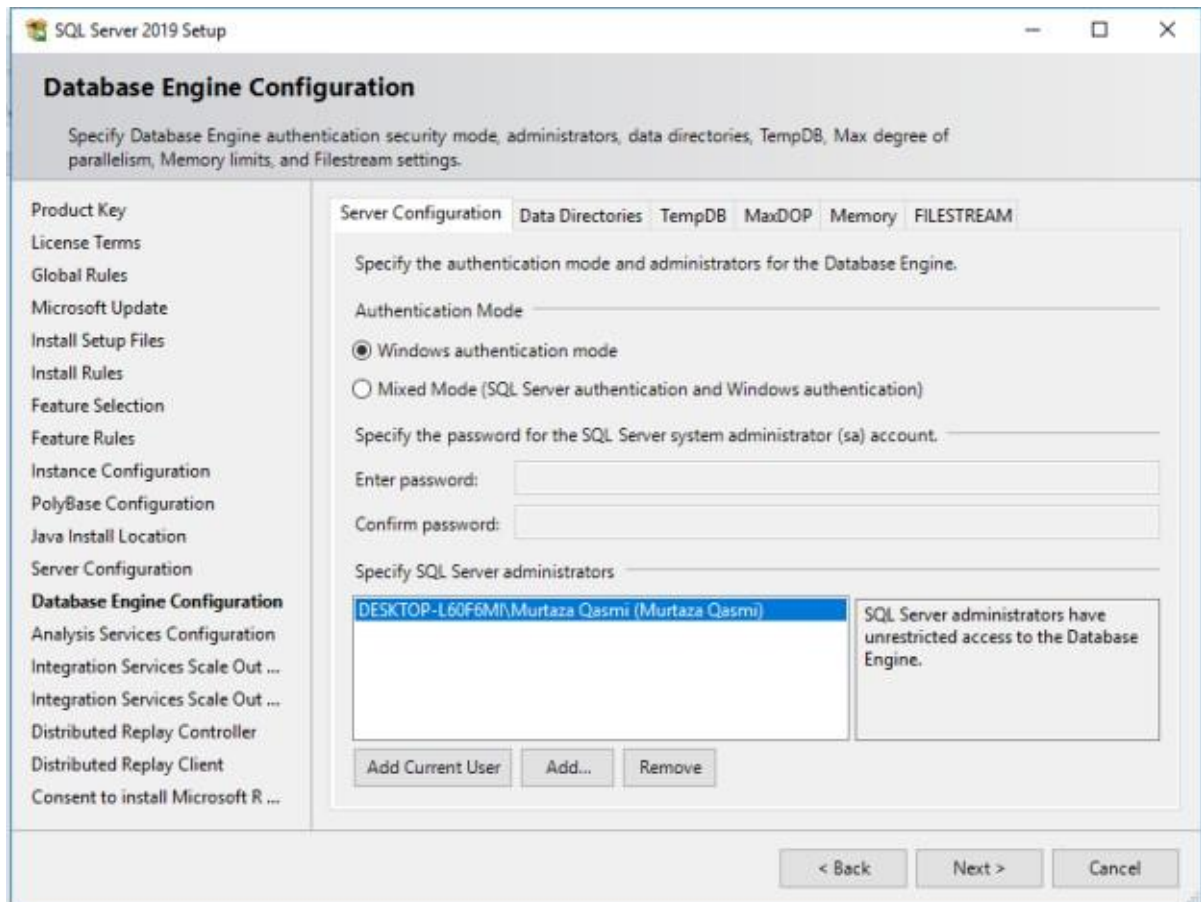


Figure 10 Server Configuration

Specify Authentication modes (Windows or Mixed mode) and add users to have SQL Server administrators.



*Figure 11 Database Engine Configuration*

Click on Data Directories and we can specify the data, log files, backup directories here. You can mention their directories in a different drive. We have local drive one drive so selected only one drive.

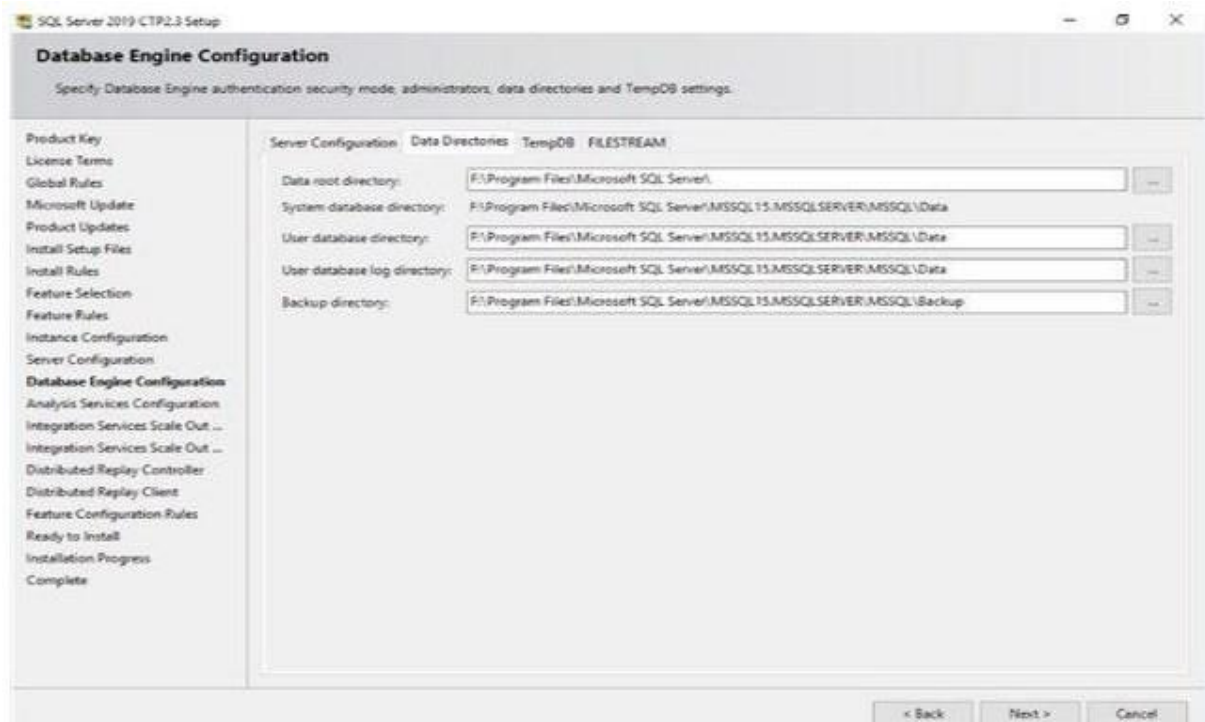


Figure 12 Database Engine Configuration

Click on TempDB to configure the TempDB configurations. Ideally, Tempdb should in a different drive.

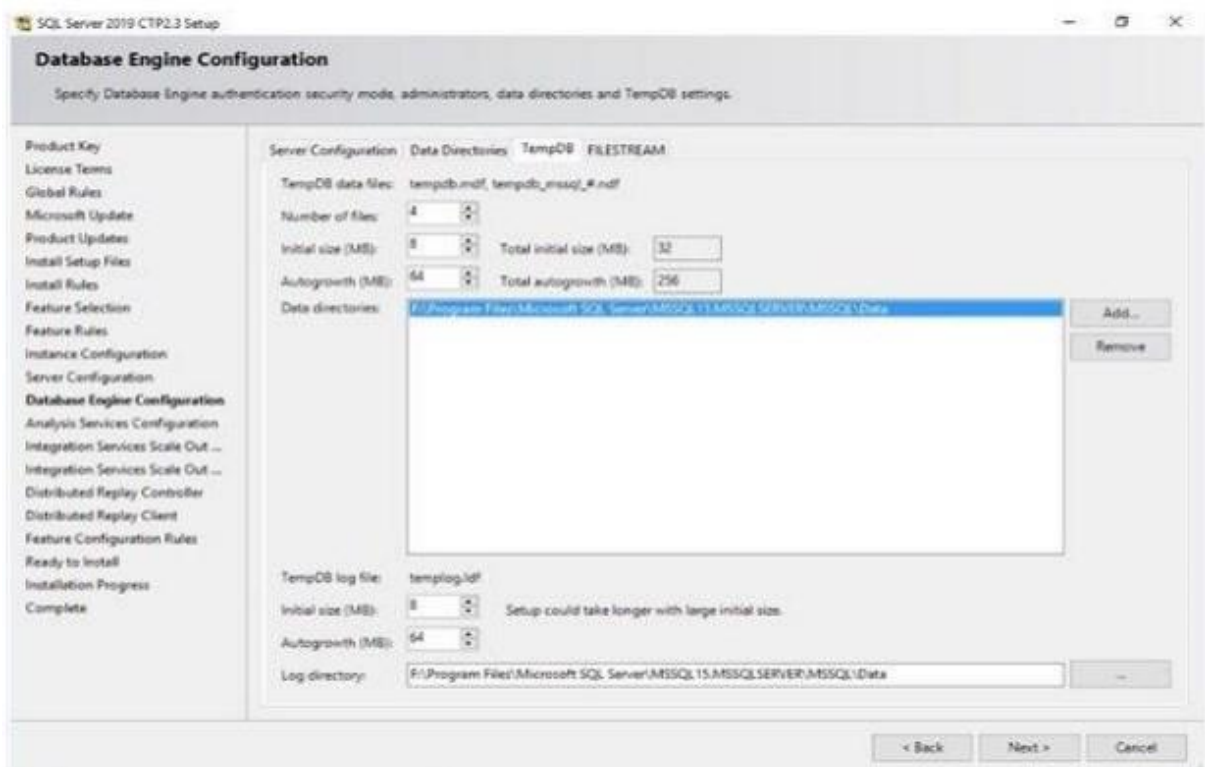


Figure 13 Database Engine Configuration

We have selected Tabular mode for SSAS.

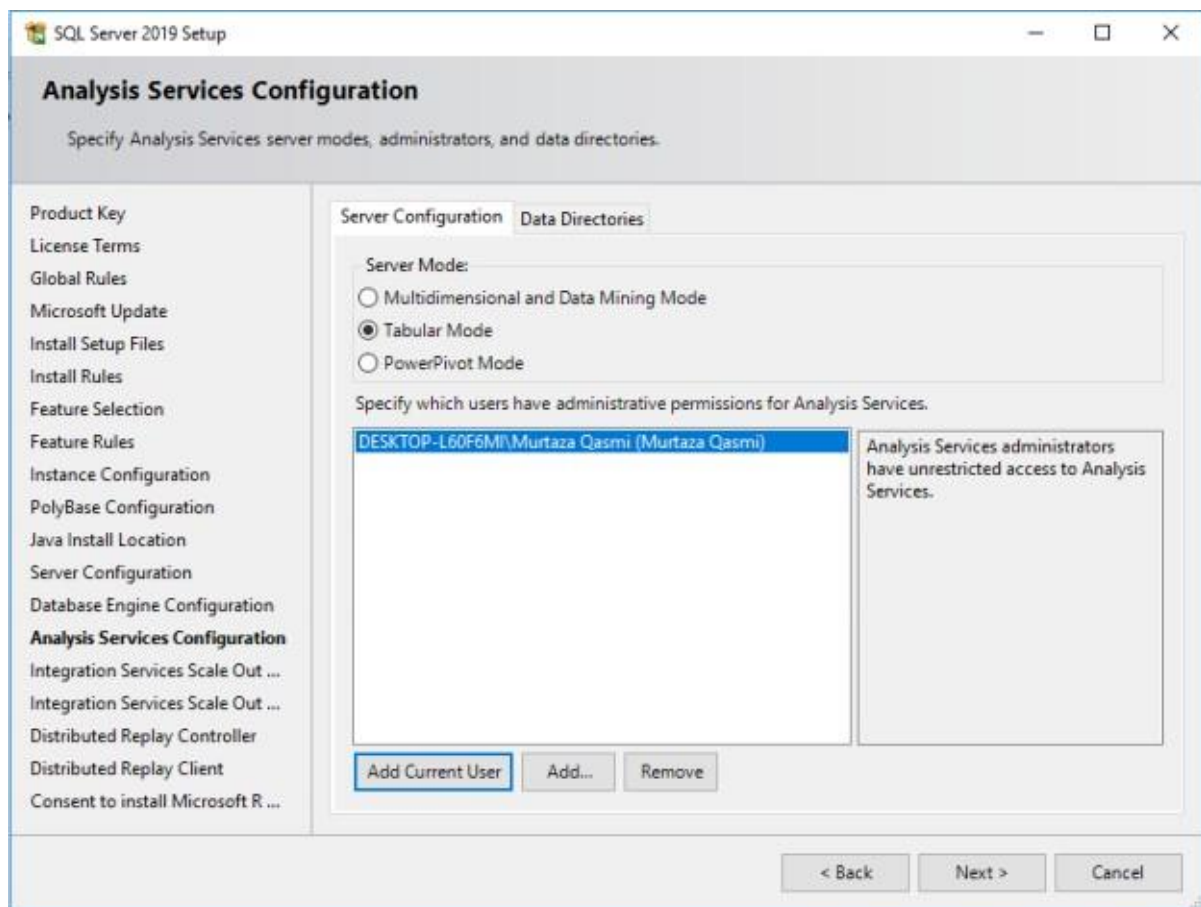


Figure 14 Analysis Services Configuration

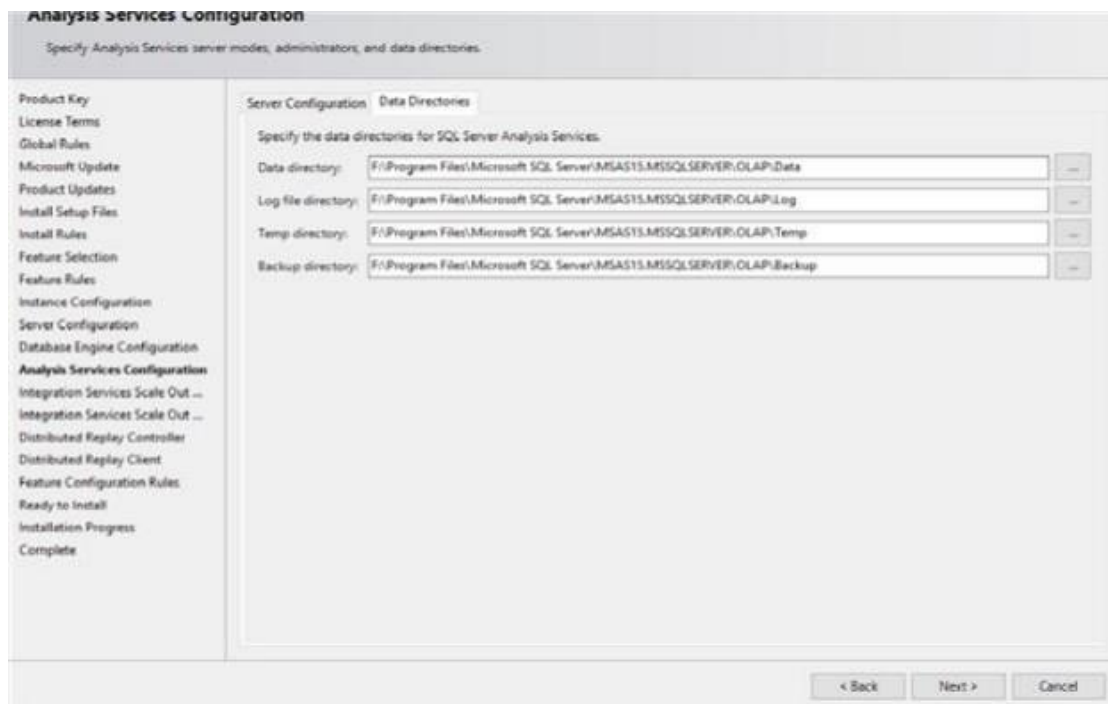


Figure 15 Analysis Services Configuration



Make sure that 8391 port should open or you can give a different port

The screenshot shows the 'Integration Services Scale Out Configuration - Master Node' window in the SQL Server 2019 Setup. The window has a title bar with the text 'SQL Server 2019 Setup'. Below the title bar, the main heading is 'Integration Services Scale Out Configuration - Master Node'. Underneath the heading, it says 'Specify the port number and security certificate for the Scale Out Master node.'.

On the left side, there is a list of configuration steps: Product Key, License Terms, Global Rules, Microsoft Update, Install Setup Files, Install Rules, Feature Selection, Feature Rules, Instance Configuration, PolyBase Configuration, Java Install Location, Server Configuration, Database Engine Configuration, Analysis Services Configuration, **Integration Services Scale Ou...**, Integration Services Scale Out ..., Distributed Replay Controller, Distributed Replay Client, and Consent to install Microsoft R ....

The main area of the window contains the following instructions and options:

- 'Specify a port number that the master node uses to communicate with the worker nodes.'
- A text box labeled 'Port Number:' with the value '8391' entered.
- 'Select a SSL certificate that is used for the communication between the master node and worker nodes in the scale out topology. A default self-signed certificate is created if you choose to create a new SSL certificate.'
- A radio button labeled 'Create a new SSL certificate' is selected.
- A text box labeled 'CNs in the certificate:' with the value 'CN=desktop-l60f6m; CN=127.0.0.1' entered.
- A radio button labeled 'Use an existing SSL certificate' is unselected.
- A text box for an existing certificate is empty.
- A 'Browse...' button is next to the empty text box.

At the bottom right, there are three buttons: '< Back', 'Next >', and 'Cancel'.

Figure 16 Master Node

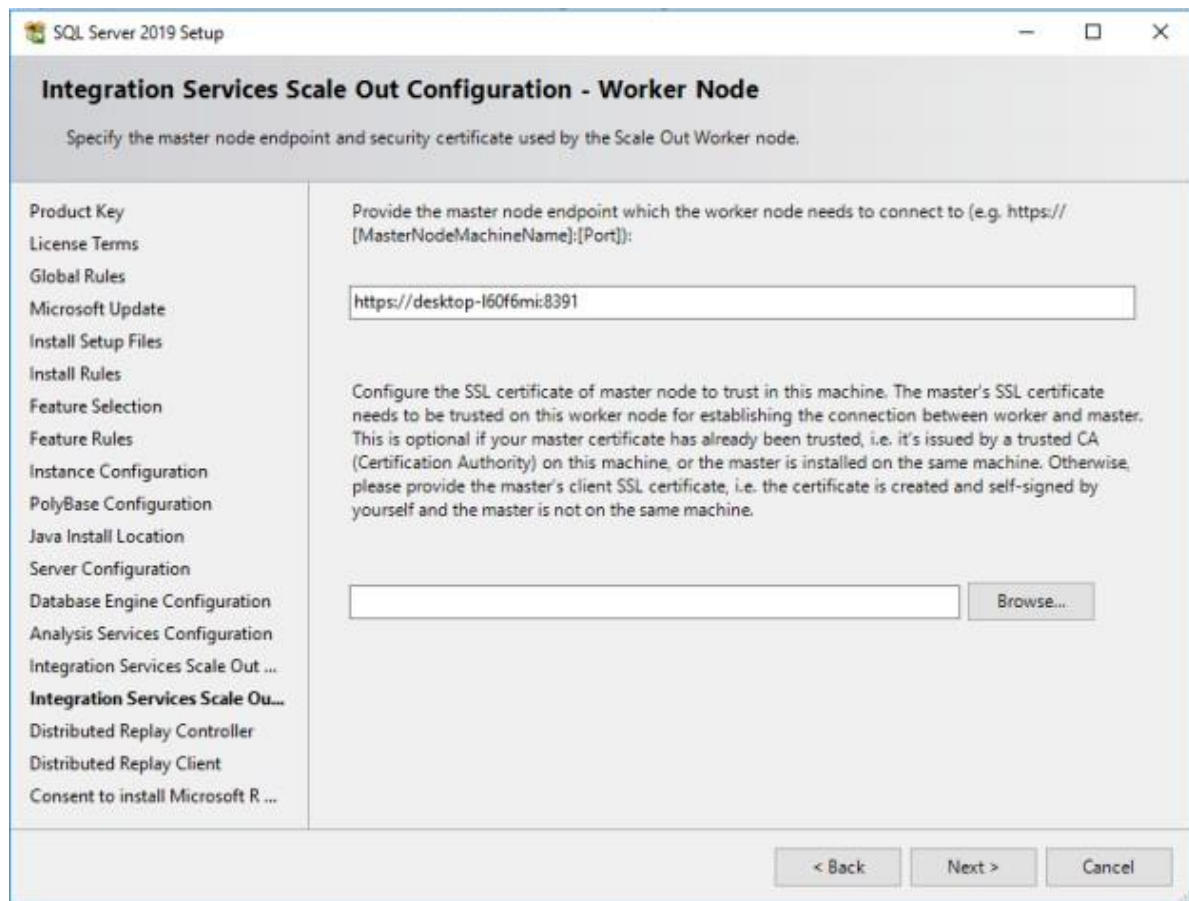


Figure 17 Worker Node



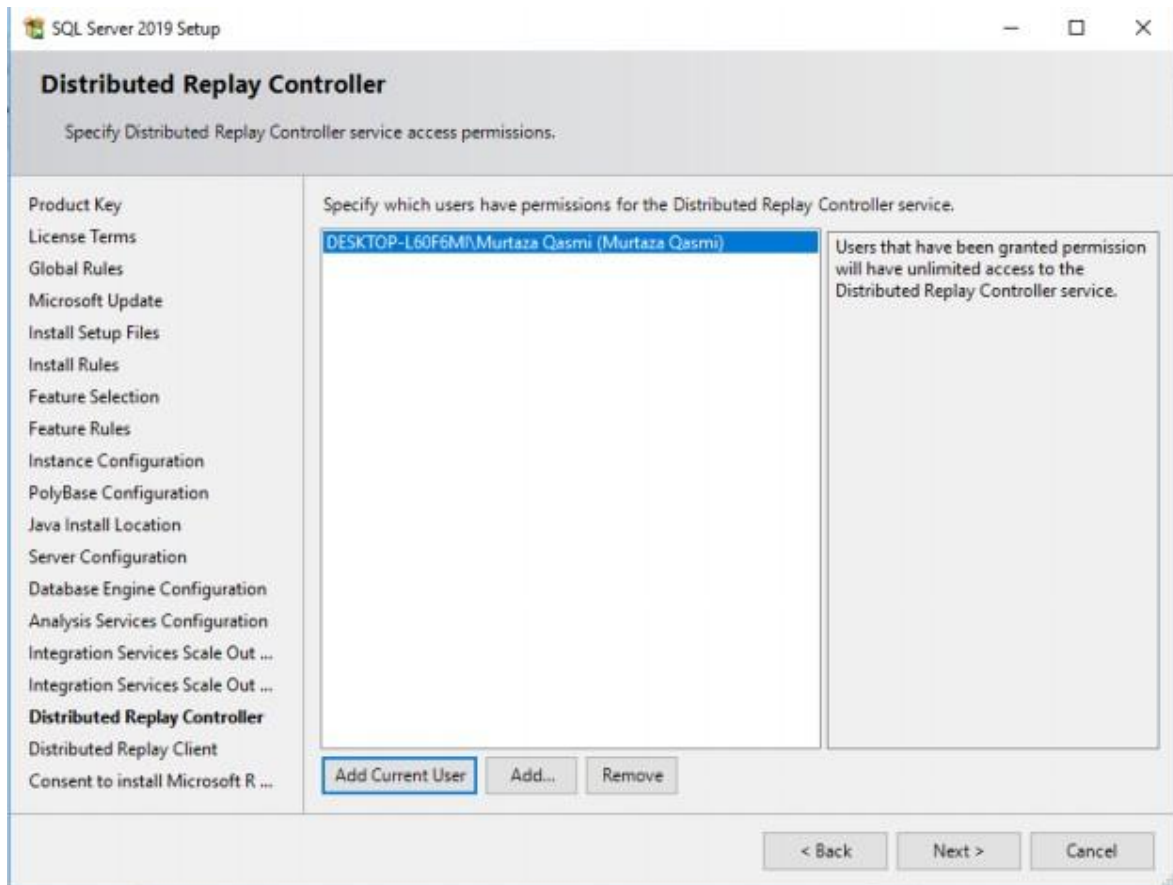


Figure 18 Distributed Replay Controller

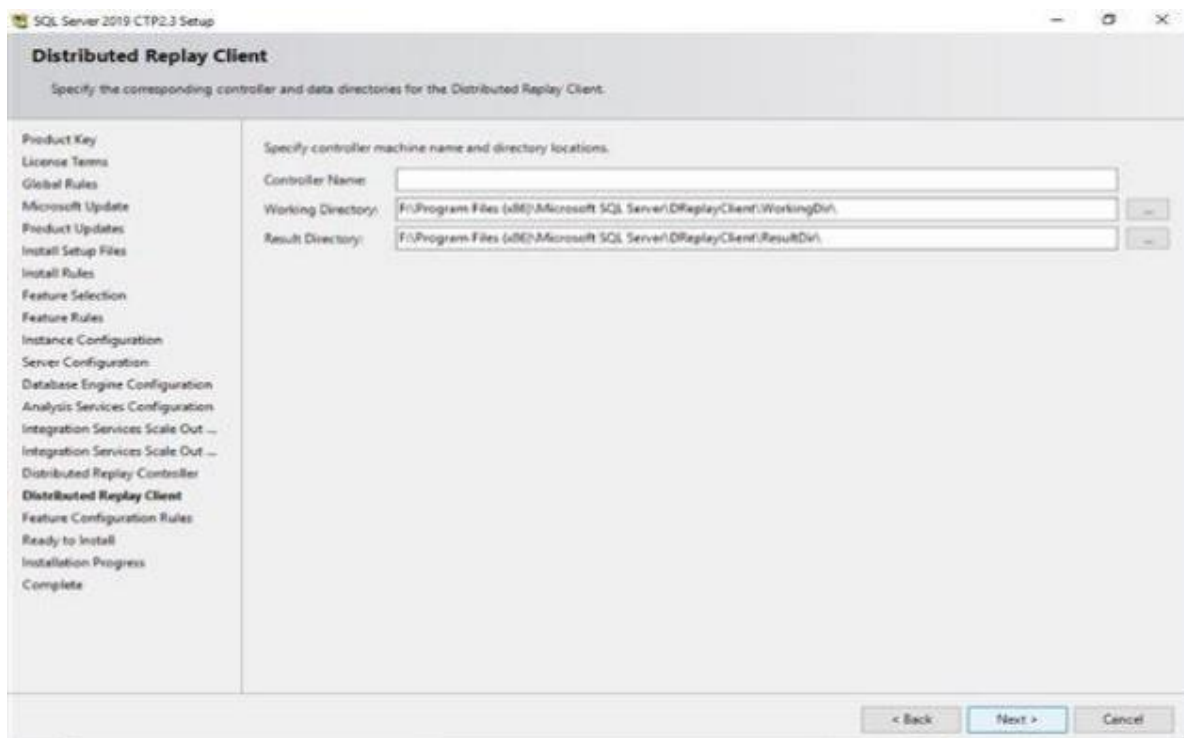
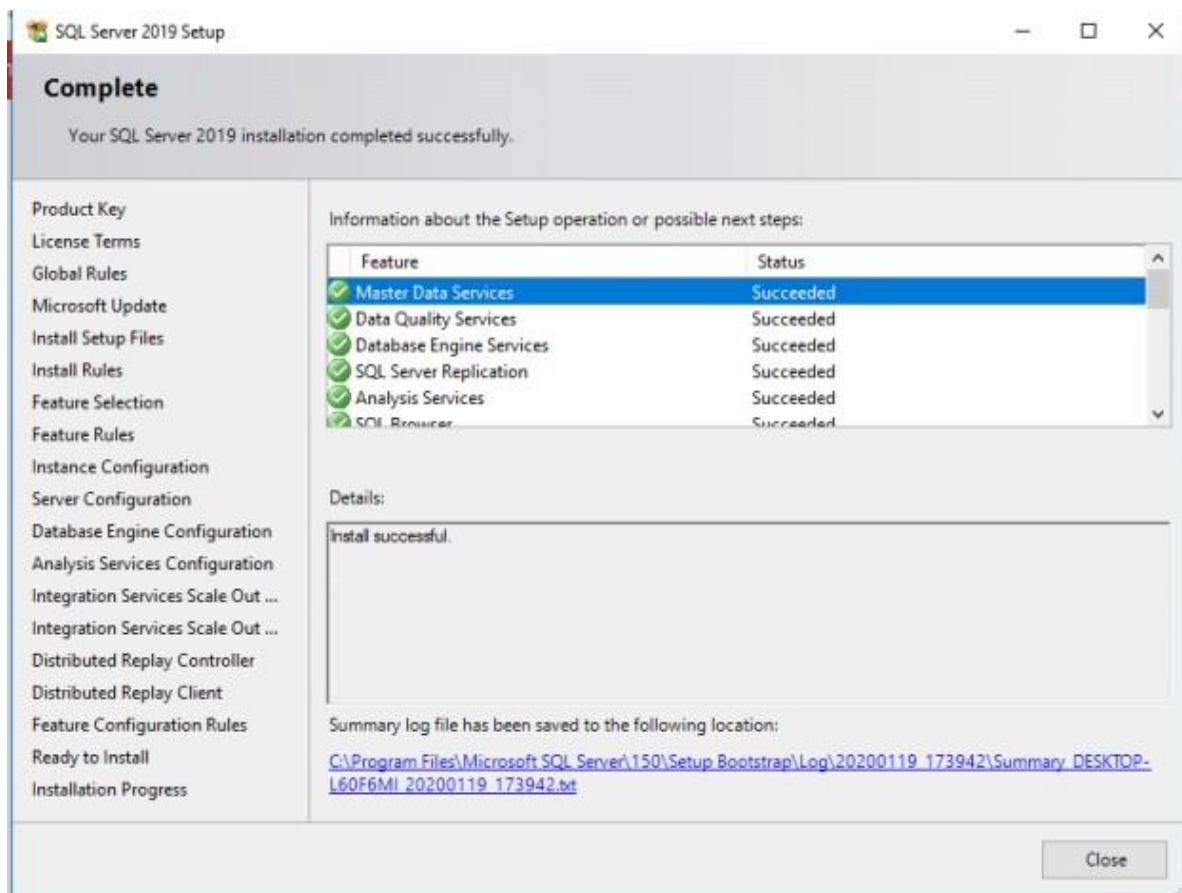


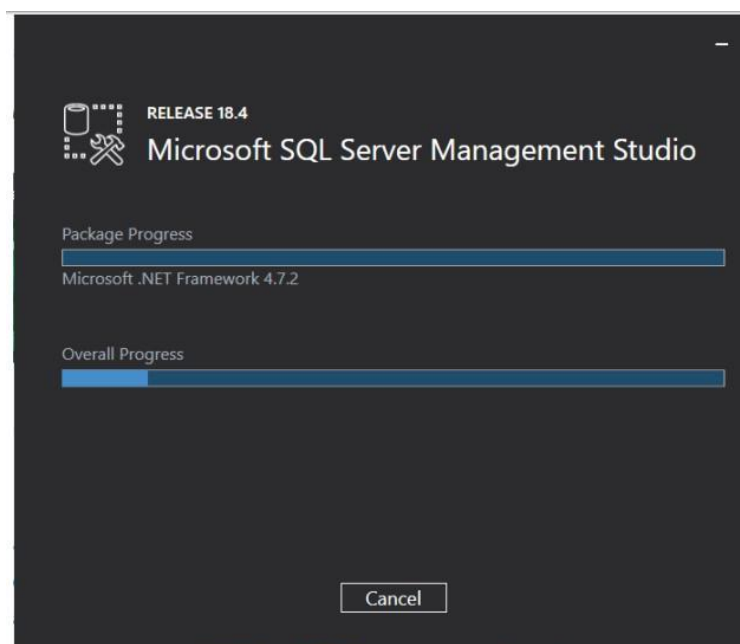
Figure 19 Distributed Replay Controller

The complete installation will require reboot server.

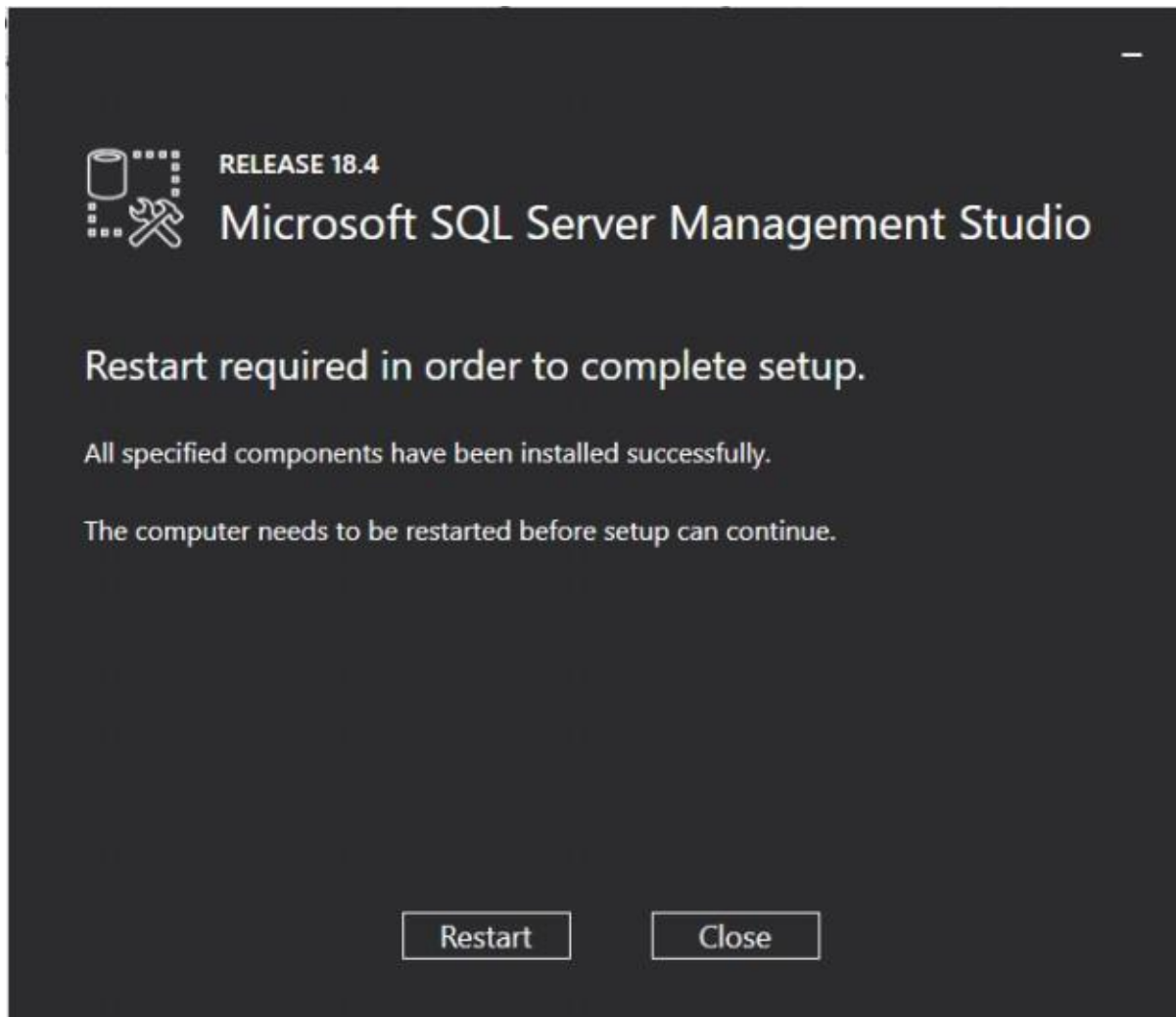


*Figure 20 Installation Complete*

Now we need to Install SQL Server management studio



*Figure 21 SQL Server*



*Figure 22 MSSMS*

## Conclusion

After following the above steps, you will be able to install sql server and sql server management studio

## CS-363 Database Systems Lab 02

**Type of Lab: Open Ended**

**Weightage: 5%**

**CLO 1:** Construct DML queries to retrieve and store data in different relations

creating tables and inserting data	<b>Cognitive/Understanding</b>	CLO1	Rubric A
------------------------------------	--------------------------------	------	----------

### Rubric A: Cognitive Domain

Evaluation Method: GA shall evaluate the students for Question according to following rubrics.

CLO	0	1	2	3	4
CLO	Student is unable to the problem	There are errors in the query	Queries are correct	The task is partially correct	Output is according to query.

## Creating Database tables and Inserting Data

### Objectives

To get familiar Structure query language, and get to know how to run simple SQL quires, make tables, insert data and make primary keys in table.

### Processing steps

#### Creating Database

1- Open Microsoft SQL Server Management Studio

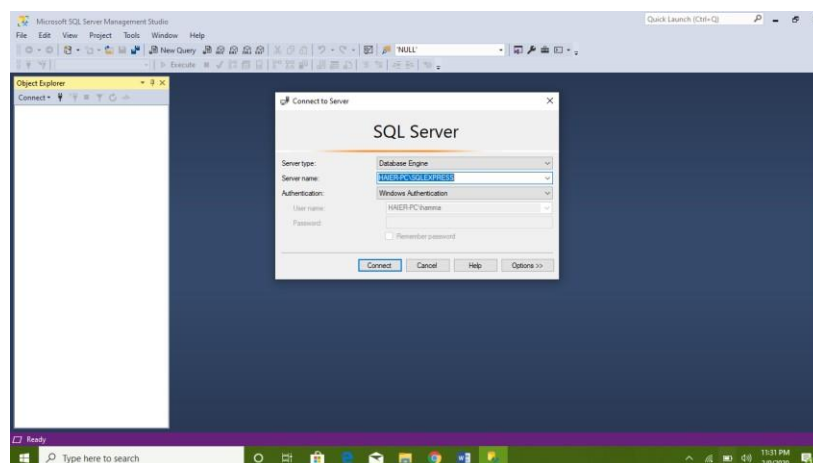


Figure 23 Creating Database

You will see this screen. Connect your Management Tool client application with server by clicking connect button. If you installed as SQL Server Authentication then enter username and password and then connect with your server application.

2-Object Explore is where you can manage your databases, Server Objects and Security. Right Click on your connection from Object Explorer and choose New Query from List

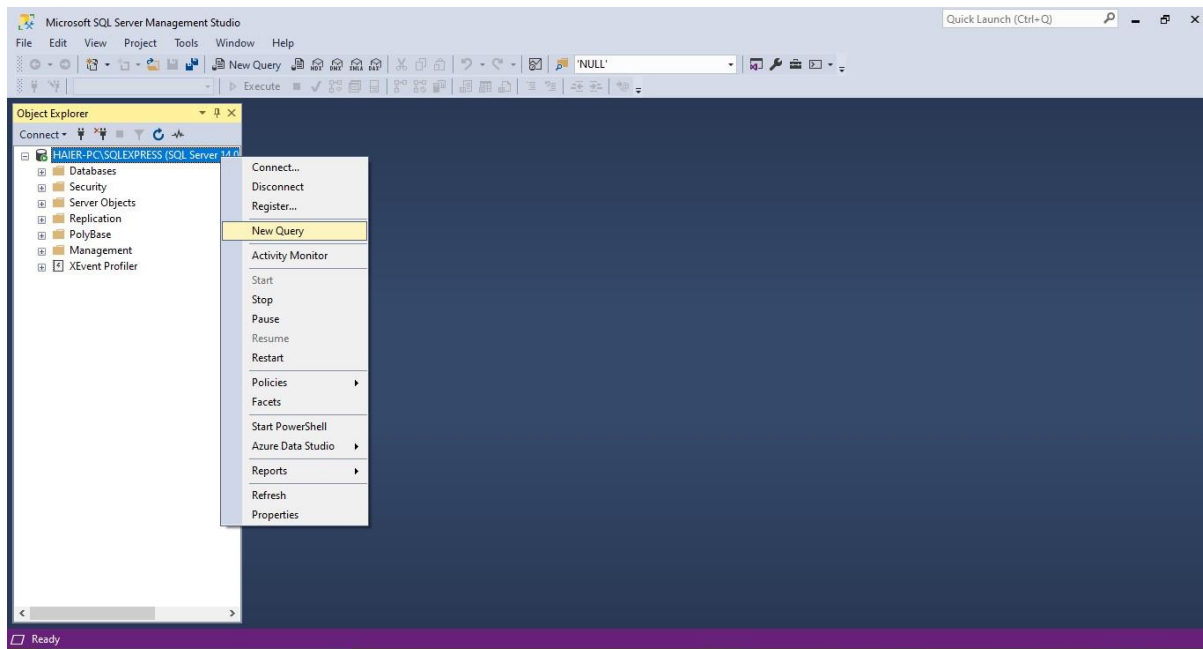


Figure 24 Query

3- Write query in Query Writing Area shown in below figure.

**CREATE DATABASE lab2**

Run this query by clicking Execute button from Toolbar. Then click on Refresh icon to refresh your content in Object Explorer.

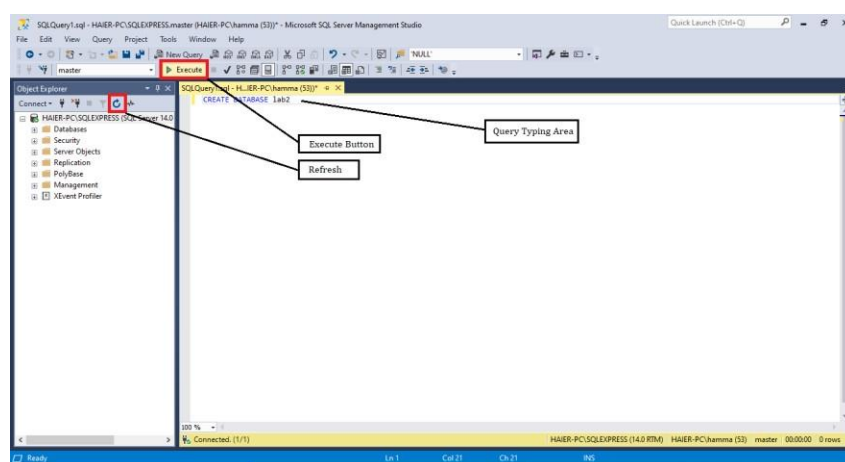
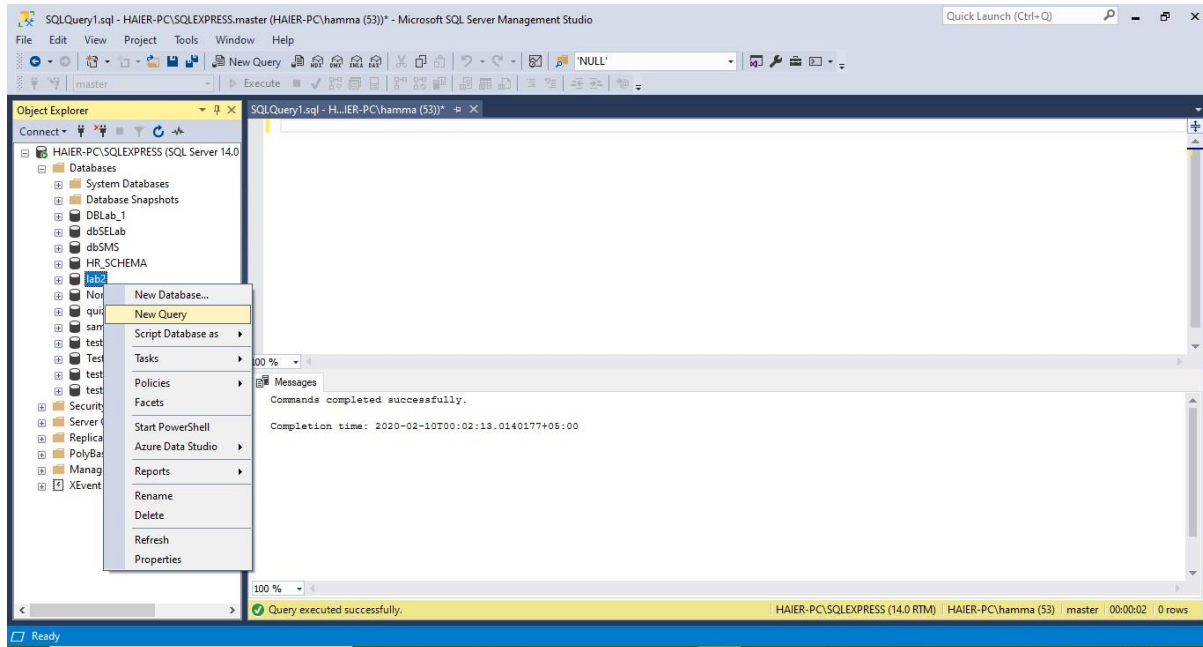


Figure 25 Execute Query

Expand Database option in Object Explorer then you will see your created database. In our case its lab2.

## Creating Table

4- Right click on the database you created and click on new query.



*Figure 26 Query*

5- Write following query to create a table to store student data and press execute.

```
create table student(  
    student_roll varchar(40),  
    student_name varchar(40),  
    student_address varchar(120),  
    age int  
)
```

Press refresh button and expand tables in labs2 database and you will the table created there.

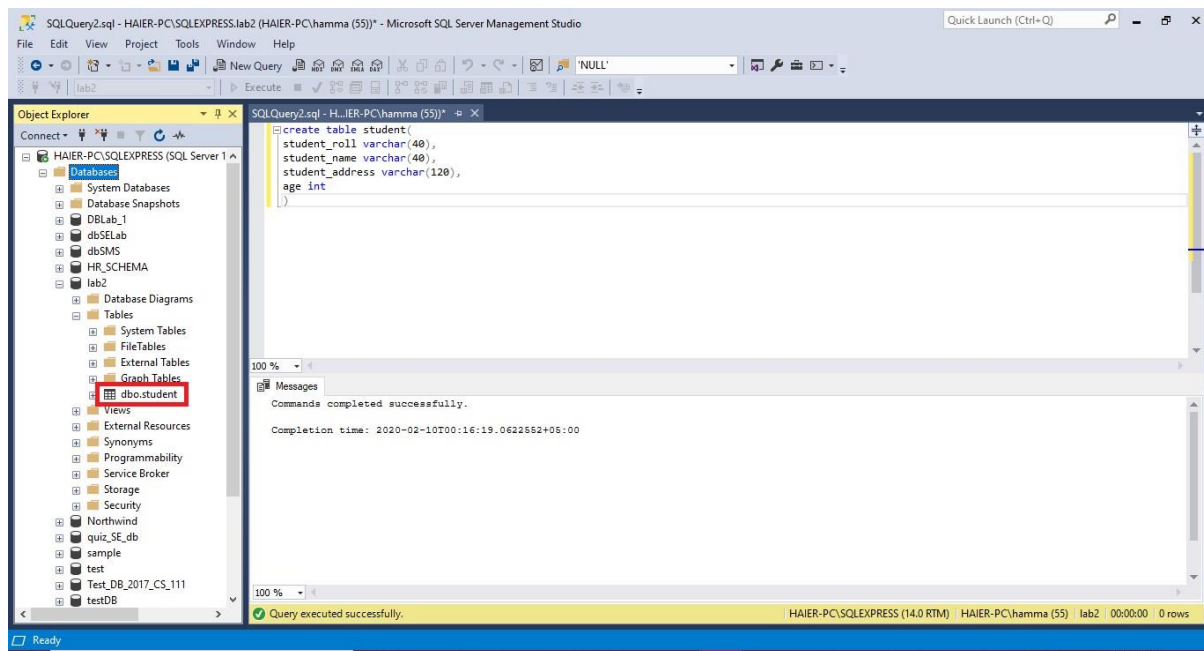


Figure 27 Creating Table

## Dropping Table

6- To drop a table use the following query.

**drop table** student

This will completely remove the table.

## Setting Primary Key

7- To create a student with primary we can use the following query

```
create table student(
student_roll varchar(40) Primary key,
student_name varchar(40),
student_address varchar(120),
age int
)
```

In this example student\_roll will be the primary key.

**Note:** If you have multiple queries in your Query File then you can run specific query from all of them just by selecting that query and then press execute button.

## Inserting Data

8- There are two ways to insert data in table

In this technique you have to take care of two things, total number of passing values should be equal to the total number of columns and sequence of value also match with column sequence. **insert into** student **values**('1', 'Lae Khan', 'Mianwali', 23)

In this technique you can define your own sequence and total number of values to pass, like in below query I am just passing two values to table. But remember you cannot leave or pass NULL value to primary key attribute. Your column names and passing values sequence must match.

```
insert into student(student_name, student_roll) values('Ali Junaid', '2')
```

Execute these two queries

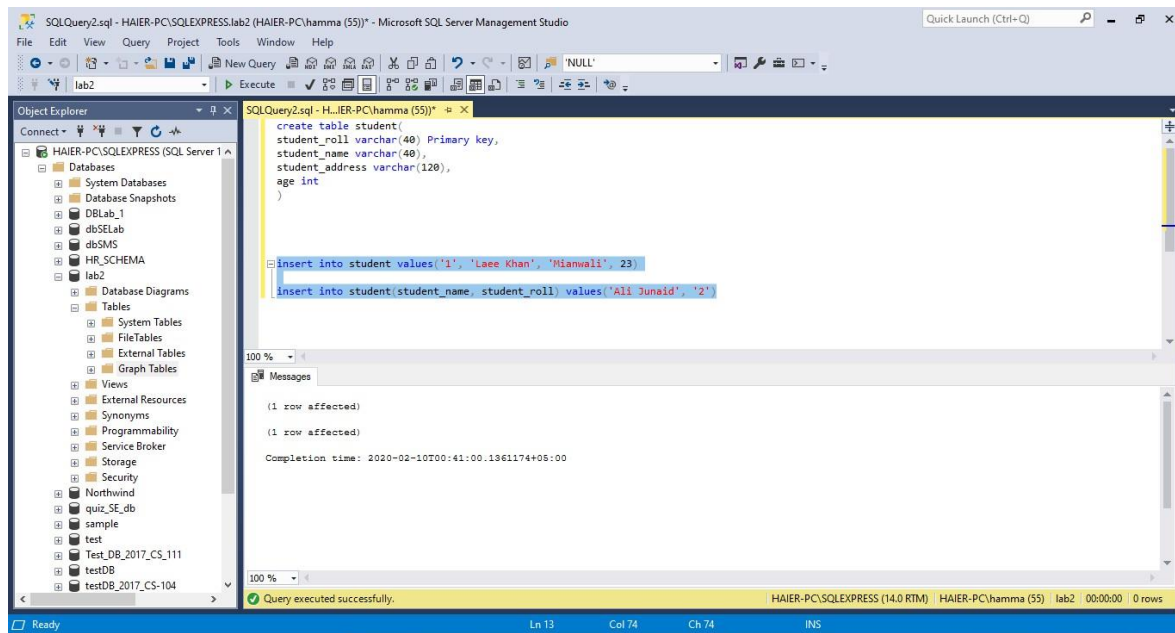


Figure 28 Inserting into table

If you run this query twice then this error will appear

These will not run because we cannot enter duplicate values in primary key for this you have to change first primary key value and then execute query otherwise database system will allow you to enter duplicate primary key.

Msg 2627, Level 14, State 1, Line 17 Violation of PRIMARY KEY constraint 'PK\_student\_7D7D75DB62D3F718'.

Primary Key data duplication error.

## Conclusion

After following the above steps, user would be able to create databases and tables and insert data into tables.

## Task

Write query to create a teacher table and insert data in it.



## CS-363 Database Systems Lab 03

**Type of Lab: Open Ended**

**Weightage: 5%**

**CLO 1:** Construct DML queries to retrieve and store data in different relations.

Retrieving and updating Data	<b>Cognitive/Understanding</b>	CLO1	Rubric A
------------------------------	--------------------------------	------	----------

### Rubric A: Cognitive Domain

Evaluation Method: GA shall evaluate the students for Question according to following rubrics.

CLO	0	1	2	3	4
CLO	Student is unable to the problem	There are errors in the query	Queries are correct	The task is partially correct	Output is according to query.

## Retrieving and Updating Data

### Objectives

Understand how to retrieve and update data and set foreign keys

### Processing steps

### Retrieving Data

To select data from table use following command

`select * from student`

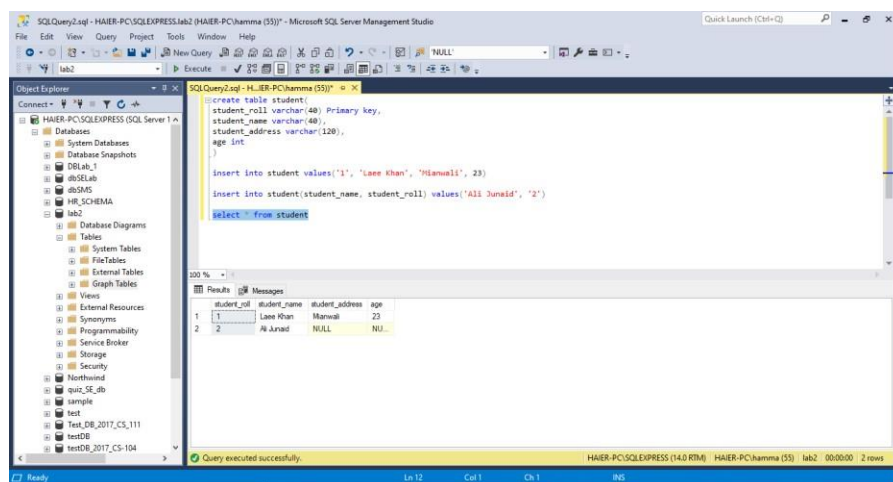


Figure 29 Retrieving Data

You will the table data in result window.

If you want to retrieve data from specific columns then you can replace this \* with Column names. For Example

```
select student_roll, student_name from student
```

Where clause

```
select * from student where student_roll=1
```

```
select * from student where student_roll=1 and student_name='Lae Khan'
```

```
select * from student where age > 20
```

these operations can be used with =, >, =, <=, and, or Where clause use to select specific data or group of data from table base on condition

## Updating Data

To update record in our student we use following query

```
update student set student_name='Lae Khan', student_address='Lahore' where  
student_roll=1
```

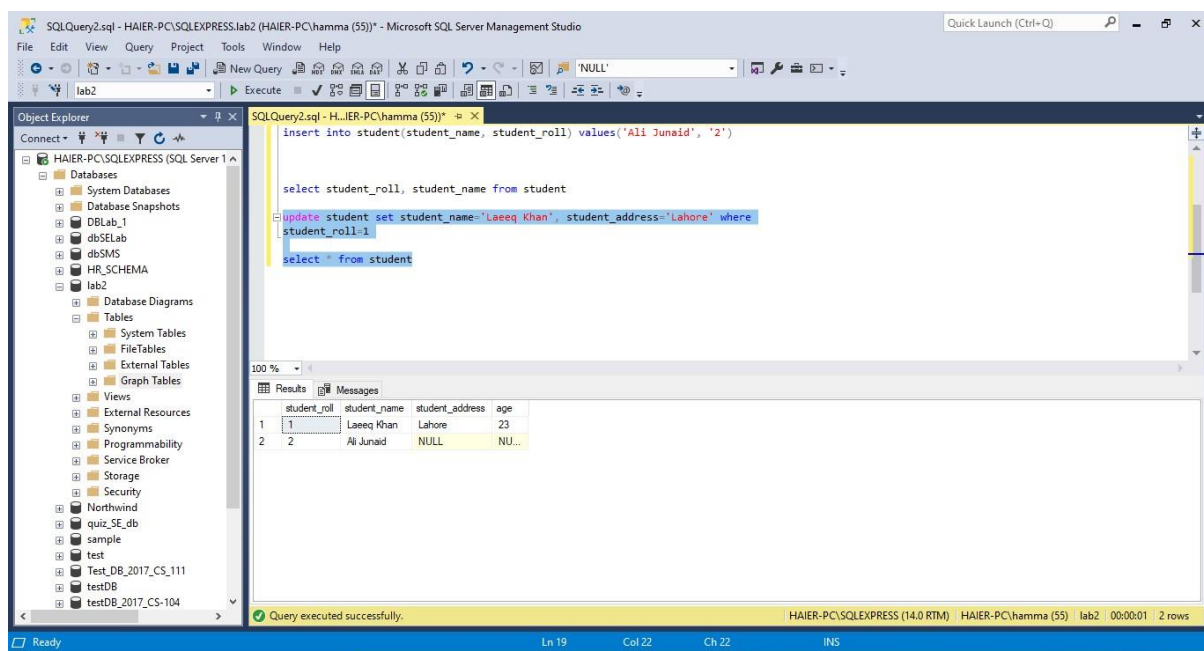


Figure 30 Updating Data

Note: Update your records with help of primary key selection otherwise your all data or more records will be updated. Like you can update all student names to Ali who age is greater than 20.

## Setting Foreign Key

In below example you will learn to create foreign key in table. Table name will be student\_details with following attributes Student\_contat, student\_dob, student\_nickname, student\_hobby, and student\_favsubject

```
create table student_details(  
  details_id int primary key,  
  student_contact varchar(40),  
  student_dob date,  
  student_nickname varchar(40),  
  student_hobby varchar(50),  
  student_fav_subject varchar(50),  
  student_rollnumber varchar(40) foreign key references student(student_roll)  
)
```

When creating foreign key there are some important things to keep in mind. Column name of foreign key can be different from referencing Primary key but data type should be same with same size

## Conclusion

After following the above steps, user would be retrieve and update data and set foreign keys

## Task

Write queries to select and update data from teacher table.

**CS-363 Database Systems Lab 04****Type of Lab: Open Ended****Weightage: 5****CLO 2:** Construct DDL queries to manage relations, constraints and indexes

Deleting, dropping and altering tables	<b>Cognitive/Understanding</b>	CLO2	Rubric A
--	--------------------------------	------	----------

**Rubric A: Cognitive Domain**

Evaluation Method: GA shall evaluate the students for Question according to following rubrics.

CLO	0	1	2	3	4
CLO	Student is unable to the problem	There are errors in the query	Queries are correct	The task is partially correct	Output is according to query.

**Deleting, Dropping and Altering Tables****Objectives**

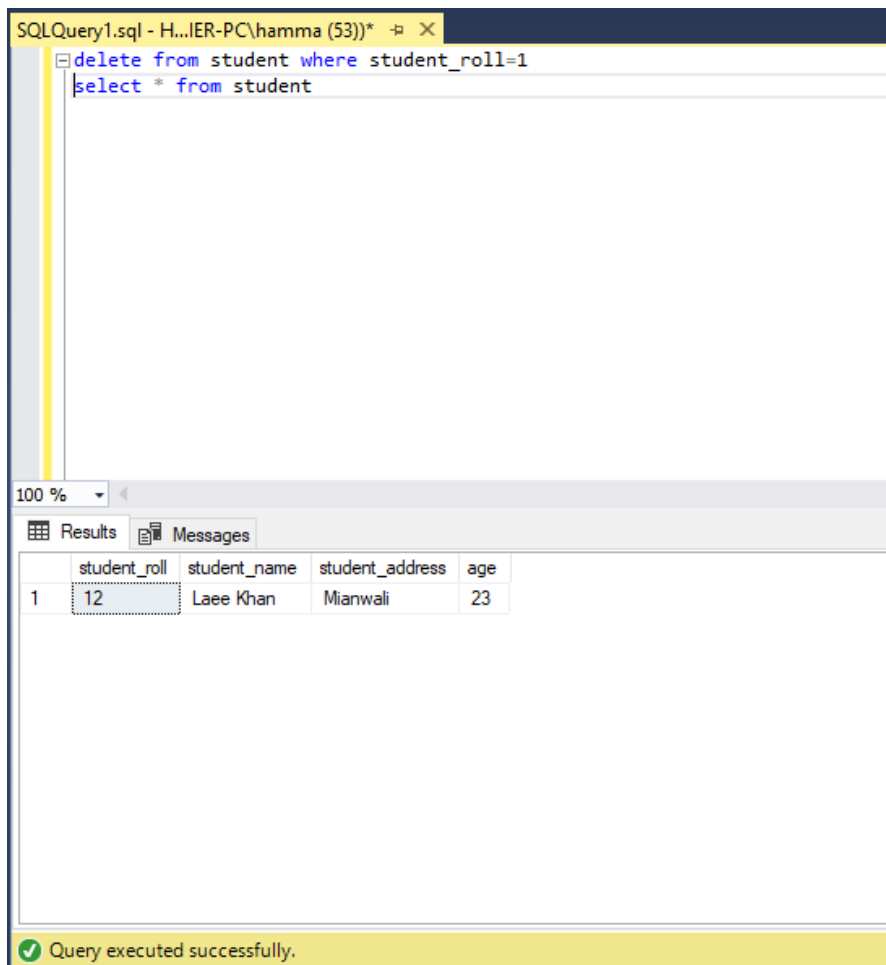
To delete data from table, drop and altar tables.

**Processing steps****Delete Data**

`delete from student where student_roll=1`

`select * from student`

Here delete query will delete records from table and select query will help use to view records in table

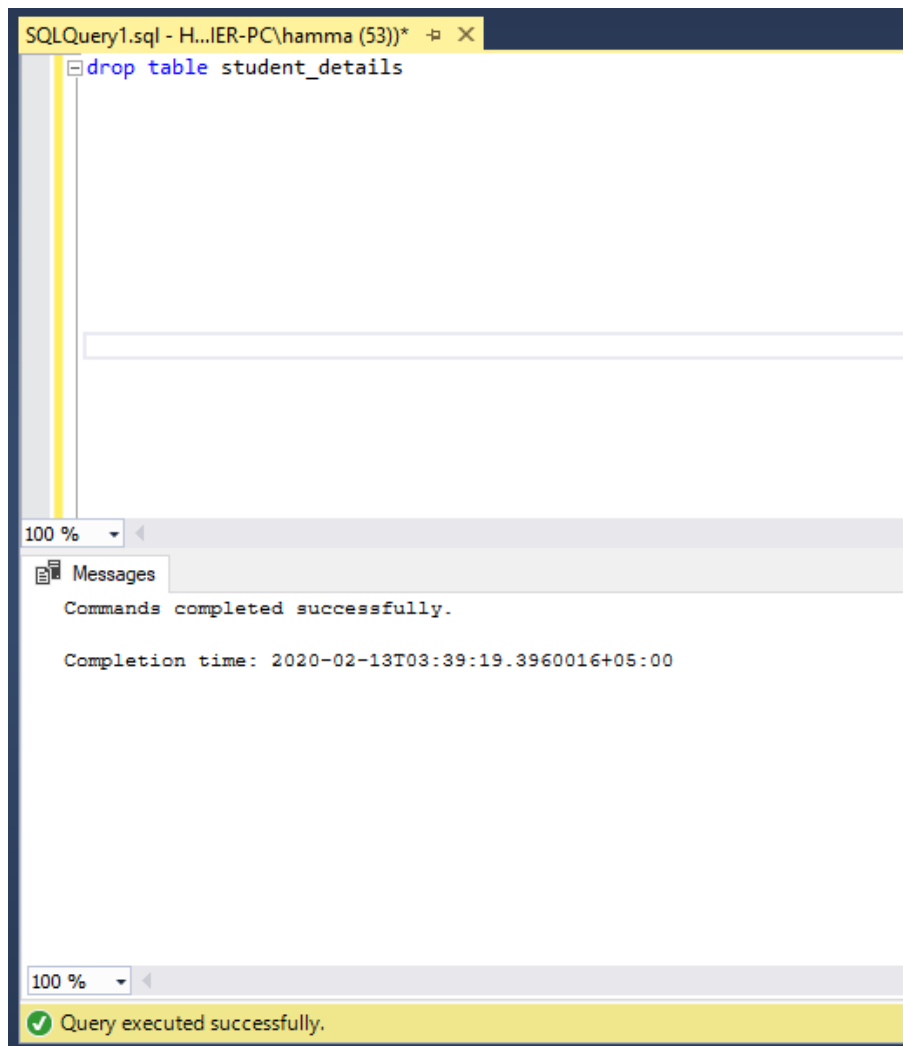


*Figure 31Deleting table*

### Drop Table:

To drop a table use following query

`drop table student_details`



*Figure 32 Drop Table*

### **Alter Table:**

If you want to add new column in existing table then here is query to add new column

`alter table student add data_of_birth varchar(60)`

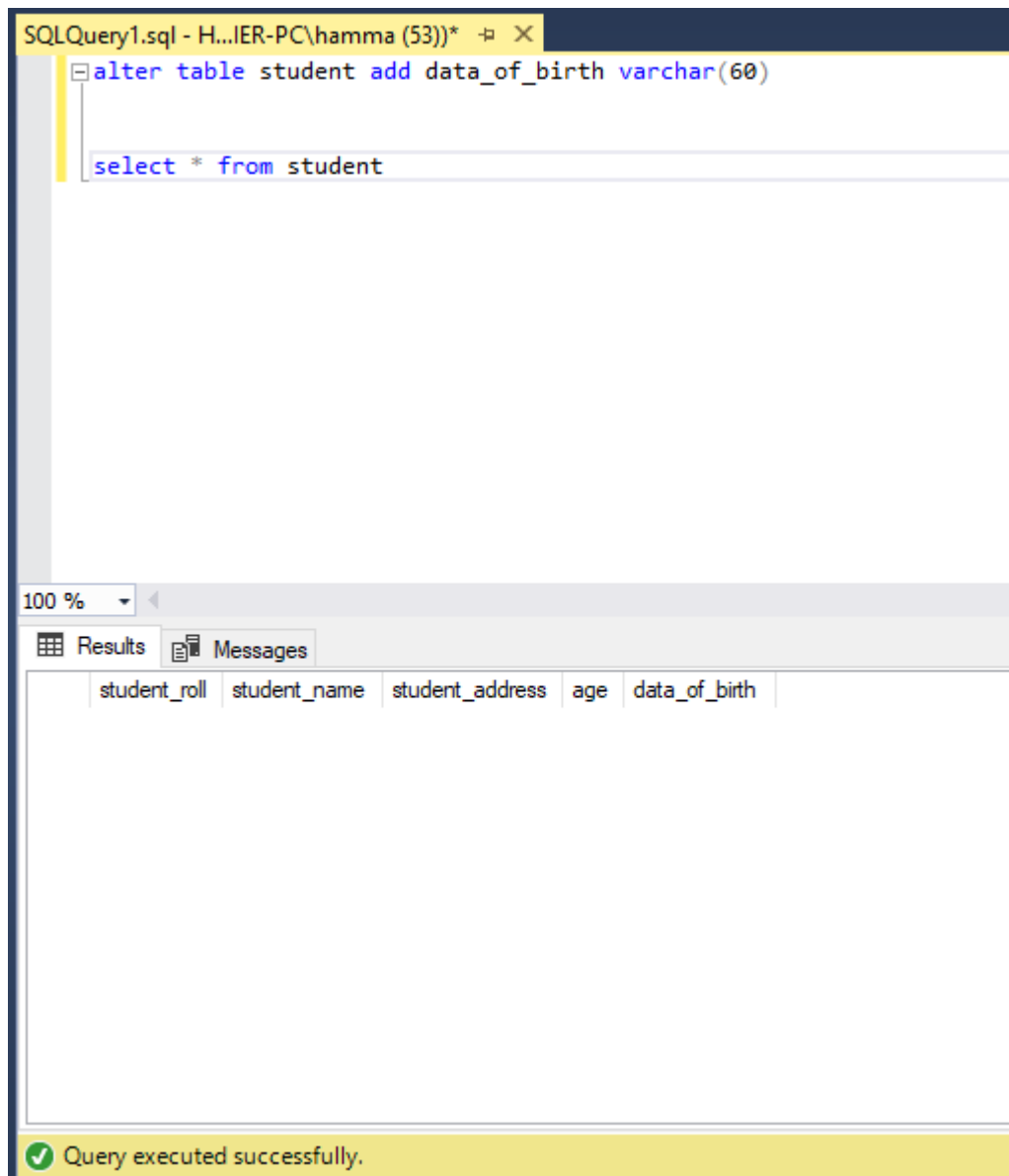


Figure 33 Alter Table

If you have multiple columns to add just separate columns with comma

```
alter table student add data_of_birth varchar(60), marks int
```

If you want to change datatype of column then here is query to change column datatype

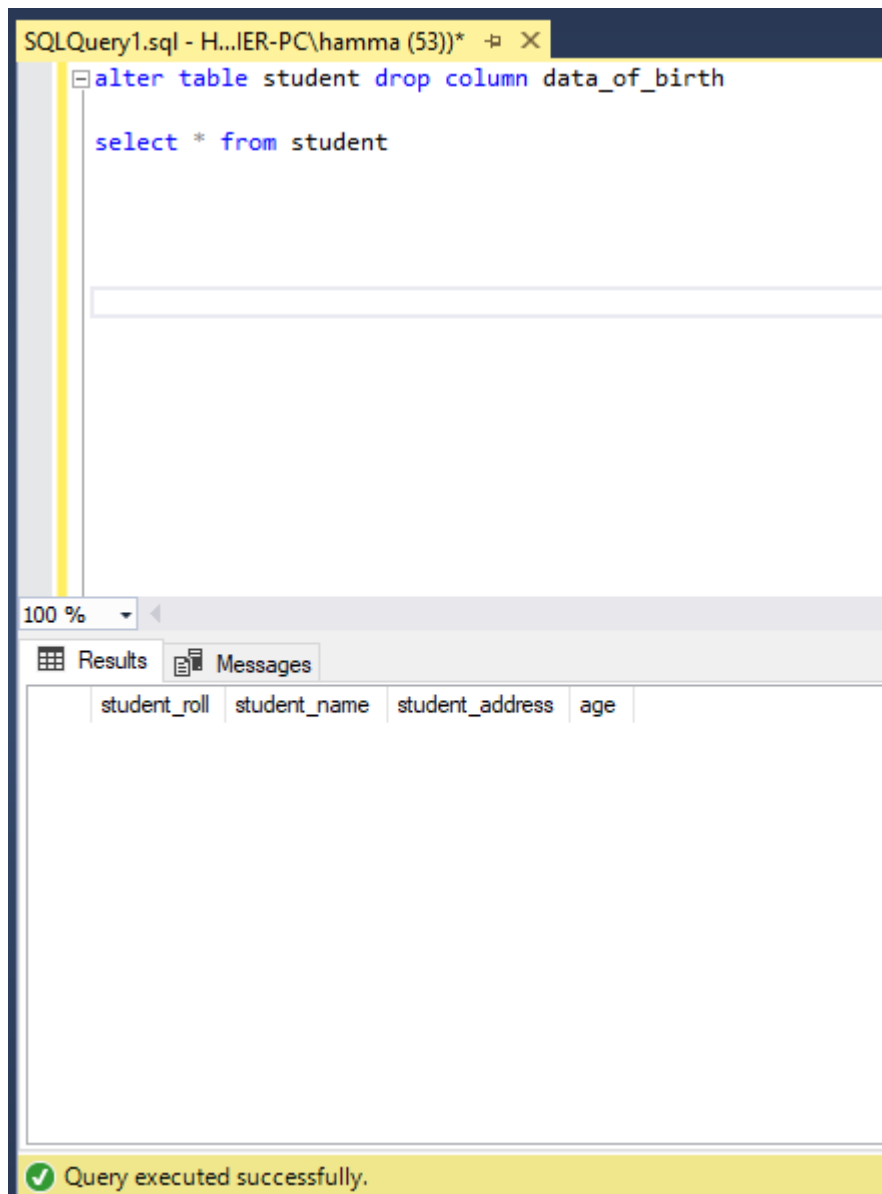
```
alter table student alter column marks float
```

This query will change marks column datatype from int to float.

```
alter table student drop column marks
```

This query will drop marks column from student table. If want to drop multiple columns once then separate columns name with comma. Here is query to drop multiple columns

```
alter table student drop column st_address, data_of_birth
```



*Figure 34 Alter Successfully*

## Conclusion

After following the above steps, user would be able to delete data, drop tables and update tables.

## Task

- Delete previous data from teacher table.
- Add a new row in teacher table



## CS-363 Database Systems Lab 05

**Type of Lab: Open Ended**

**Weightage: 5%**

**CLO 1:** Construct DML queries to retrieve and store data in different relations.

Use Aliasing	<b>Cognitive/Understanding</b>	CLO1	Rubric A
--------------	--------------------------------	------	----------

### Rubric A: Cognitive Domain

Evaluation Method: GA shall evaluate the students for Question according to following rubrics.

CLO	0	1	2	3	4
CLO	Student is unable to the problem	There are errors in the query	Queries are correct	The task is partially correct	Output is according to query.

## Aliasing

### Objectives

Develop understanding of column and Table Alias

### Processing steps

#### ALIASING TABLES AND COLUMNS

Aliases provide database administrators, as well as other database users, with the ability to reduce the amount of code required for a query, and to make queries simpler to understand. In addition, aliasing can be used as an obfuscation technique to protect the real names of database fields.

#### ALIASING COLUMNS

**SELECT** column\_name **AS** alias\_name **FROM** table\_name

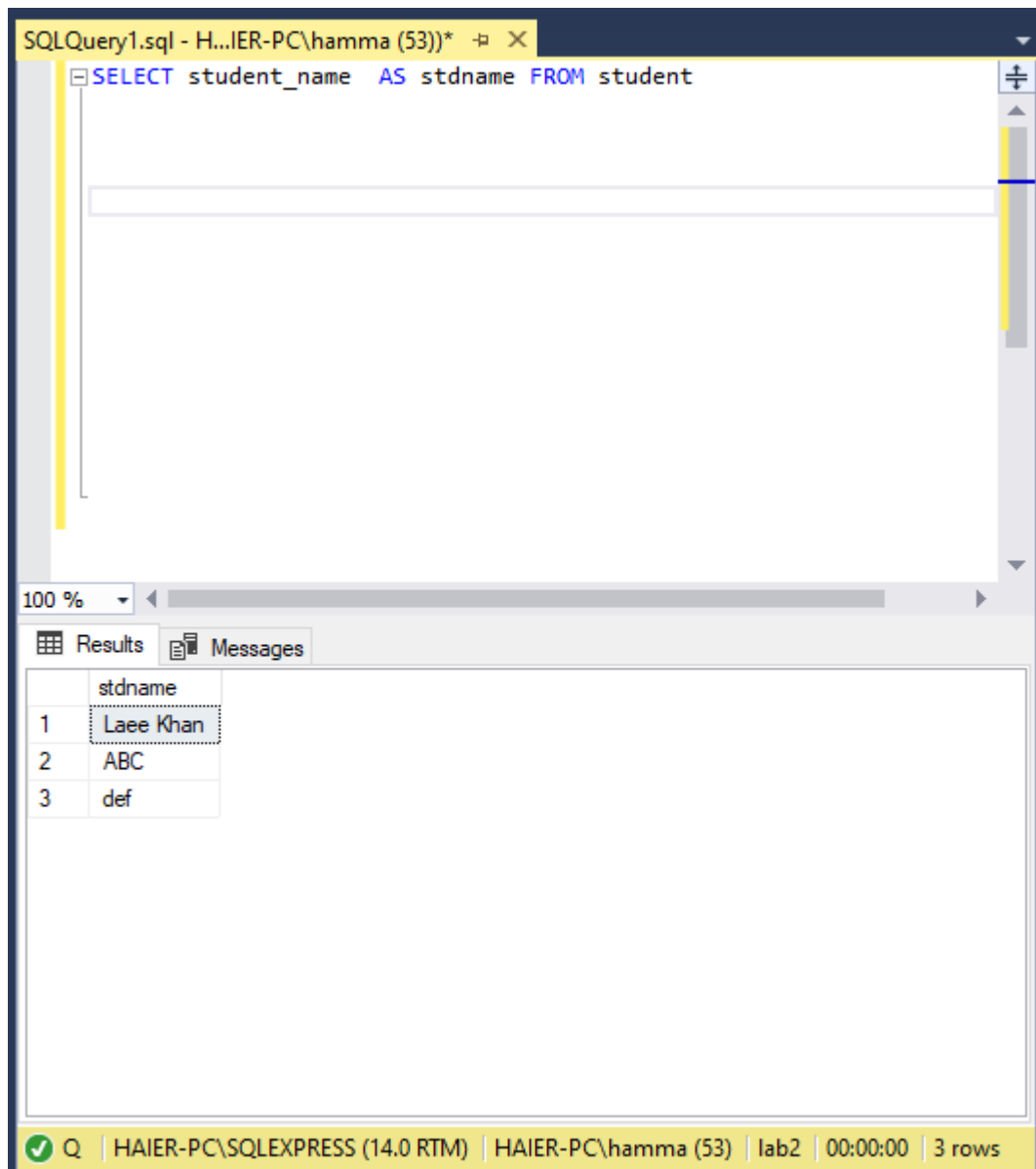


Figure 35 Aliasing

## ALIASING TABLES

`SELECT` column\_name(s) `FROM` table\_name `AS` alias\_name;

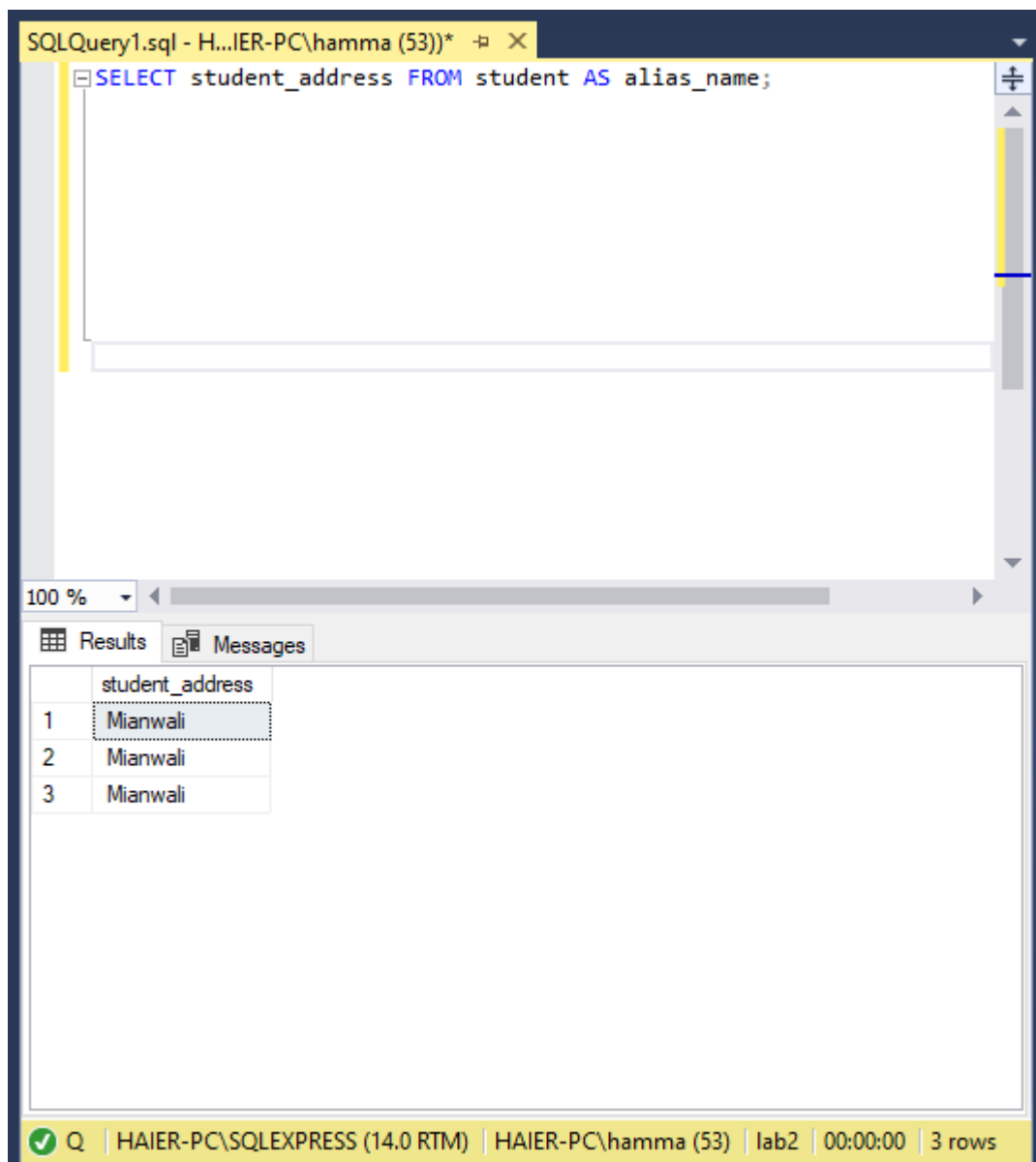


Figure 36 Aliasing

## Conclusion

After following the above steps, user would be able to alias tables and columns.

## Task

Use aliasing to show teacher's name and age from teacher table.

## CS-363 Database Systems Lab 06

**Type of Lab: Open Ended**

**Weightage: 5%**

**CLO 1:** Construct DML queries to retrieve and store data in different relations.

Restricting and Sorting Data	<b>Cognitive/Understanding</b>	CLO1	Rubric A
------------------------------	--------------------------------	------	----------

### Rubric A: Cognitive Domain

Evaluation Method: GA shall evaluate the students for Question according to following rubrics.

CLO	0	1	2	3	4
CLO	Student is unable to the problem	There are errors in the query	Queries are correct	The task is partially correct	Output is according to query.

## Clauses

### Objectives

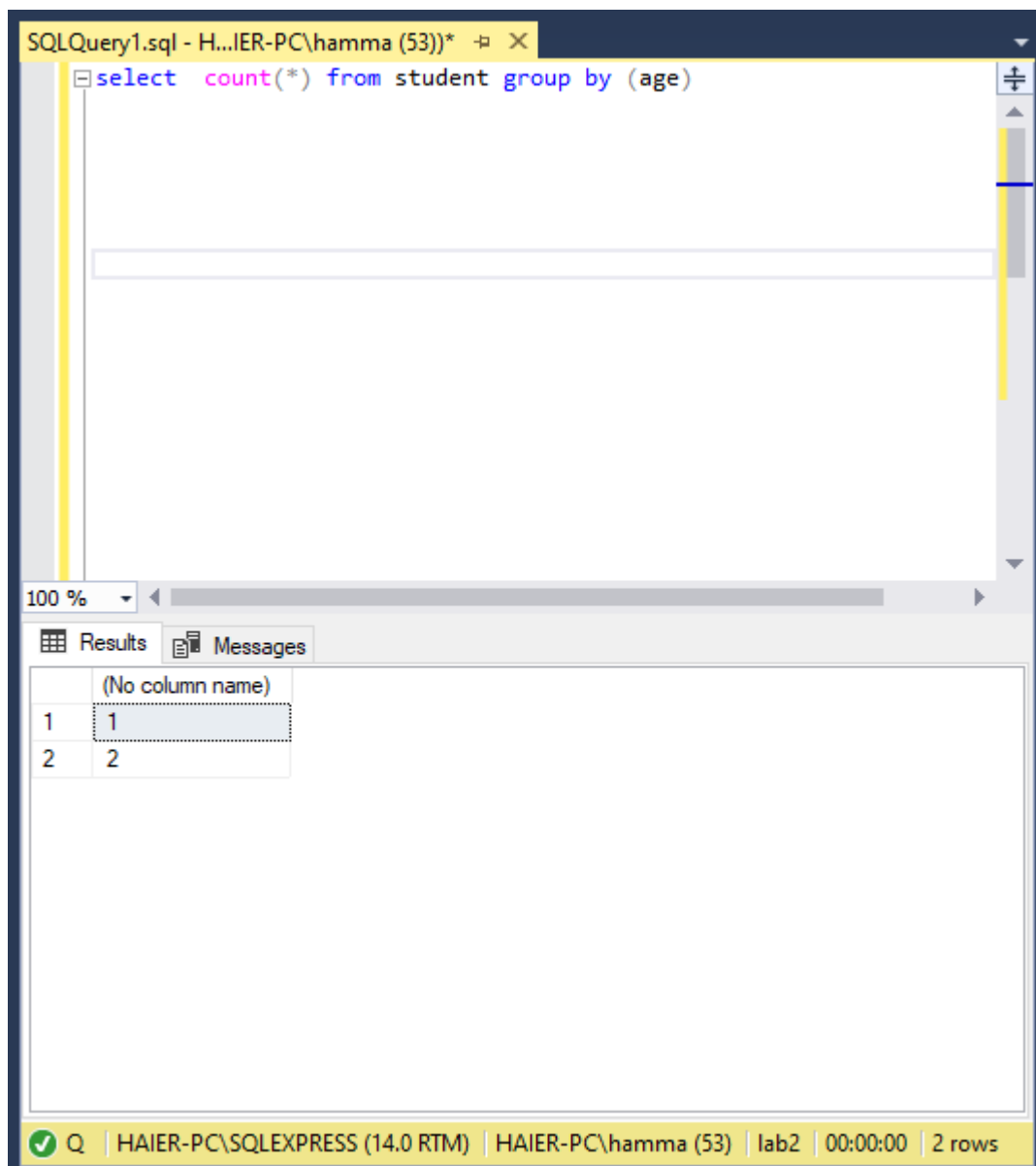
Develop understanding of group by, order by and having Clause

### Processing steps

#### Group By Clause

The SQL GROUP BY clause is used in collaboration with the SELECT statement to arrange identical data into groups. This GROUP BY clause follows the WHERE clause in a SELECT statement and precedes the ORDER BY clause.

```
select count(*) from student group by (age)
```

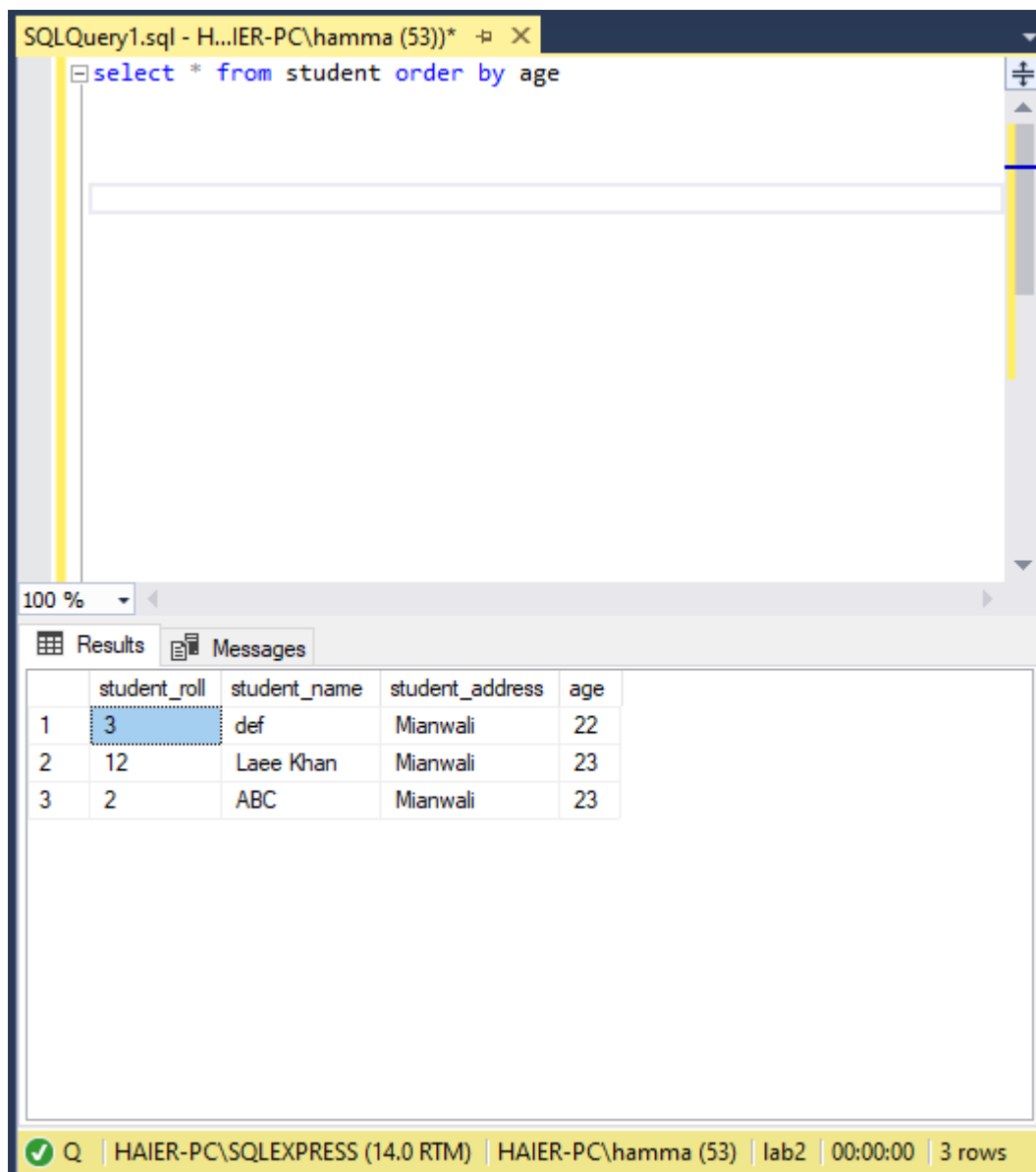


*Figure 37 Clauses*

### Order by Clause

The ORDER BY is used to sort the result-set in ascending or descending order.

The ORDER BY sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

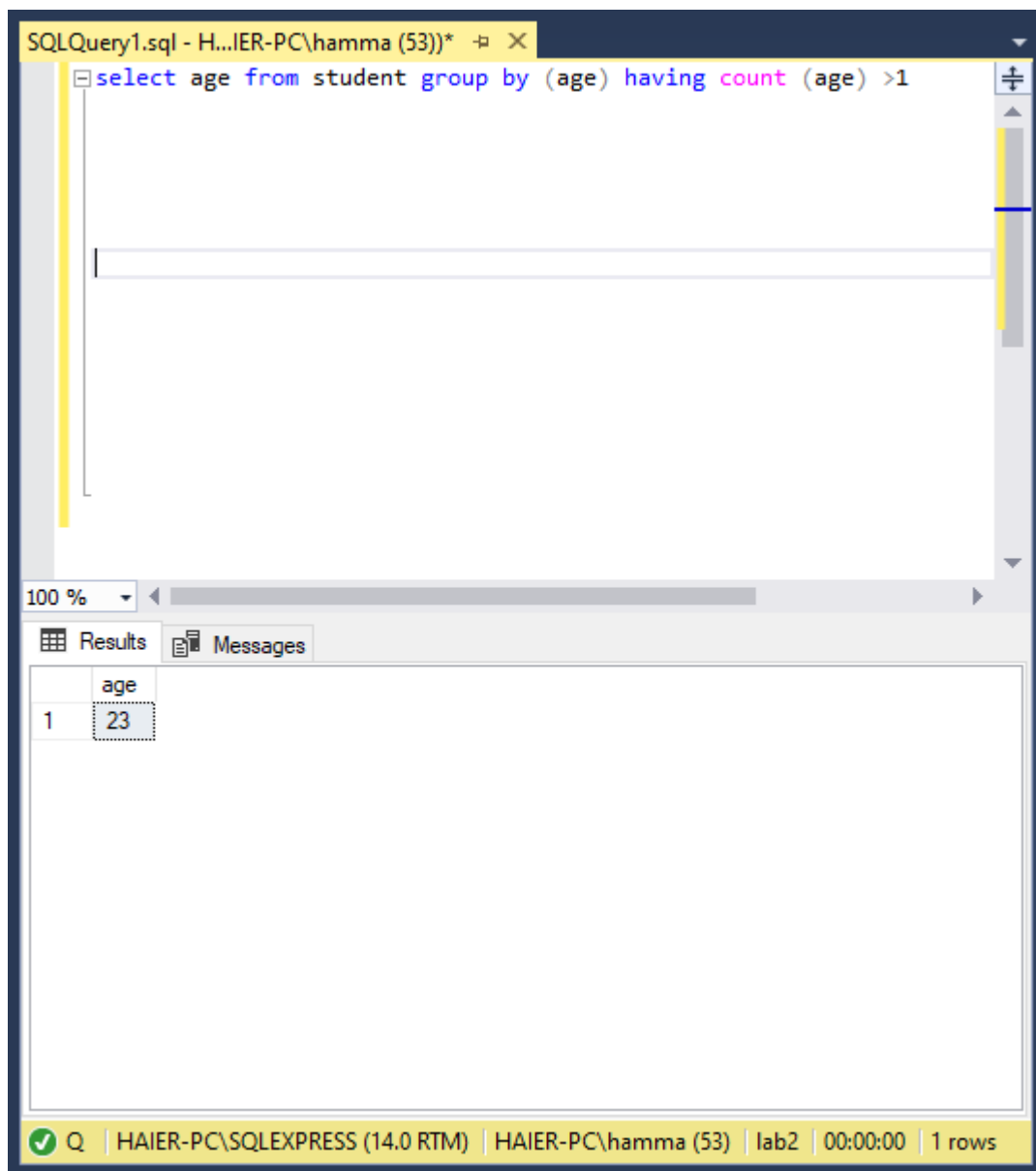


*Figure 38 Order by Clauses*

## Having Clause

The WHERE clause is a row filter, the HAVING clause is a group filter. Only groups for which the HAVING predicate evaluates to TRUE are returned by the HAVING phase to the next logical query processing phase. Groups for which the predicate evaluates to FALSE or UNKNOWN are discarded. Because the HAVING clause is processed after the rows have been grouped, you can refer to aggregate functions in the logical expression.

`select age from student group by (age) having count (age) >1`



*Figure 39 Having Clauses*

## Conclusion

After following the above steps, user would be able to use group by, order by and having clauses.

## Tasks

- Use clauses to order teacher data by age.
- Use clauses to filter out teachers with at least 4 years' experience.

## CS-363 Database Systems Lab 07

**Type of Lab: Open Ended**

**Weightage: 5%**

**CLO 1:** Construct DML queries to retrieve and store data in different relations.

Using single row and group aggregation Functions to Customize Output	<b>Cognitive/Understanding</b>	CLO1	Rubric A
--	--------------------------------	------	----------

### Rubric A: Cognitive Domain

Evaluation Method: GA shall evaluate the students for Question according to following rubrics.

CLO	0	1	2	3	4
CLO	Student is unable to the problem	There are errors in the query	Queries are correct	The task is partially correct	Output is according to query.

## Aggregation

### Objectives

Develop understanding of aggregation functions

### Processing steps

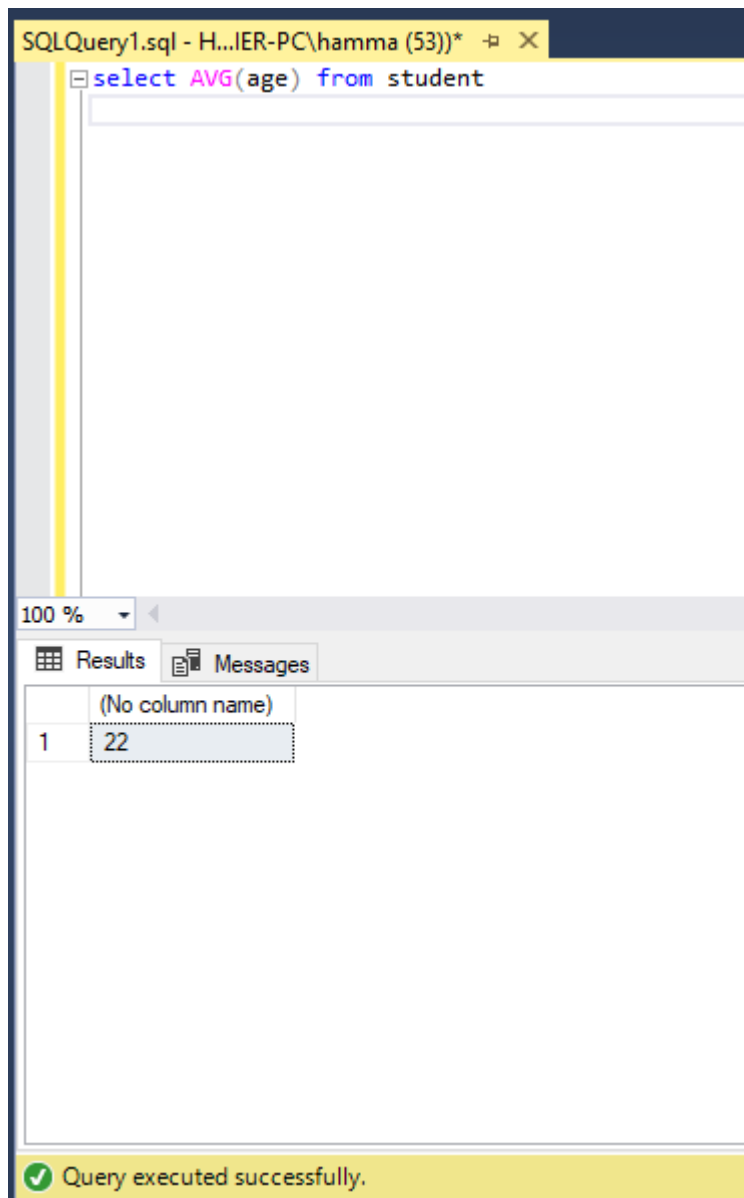
#### Aggregation

Aggregate functions allow you to perform a calculation on a set of values to return a single scalar value. The most common aggregate functions will be performed today.

#### AVG function

AVG function is used to get average of the values in a column

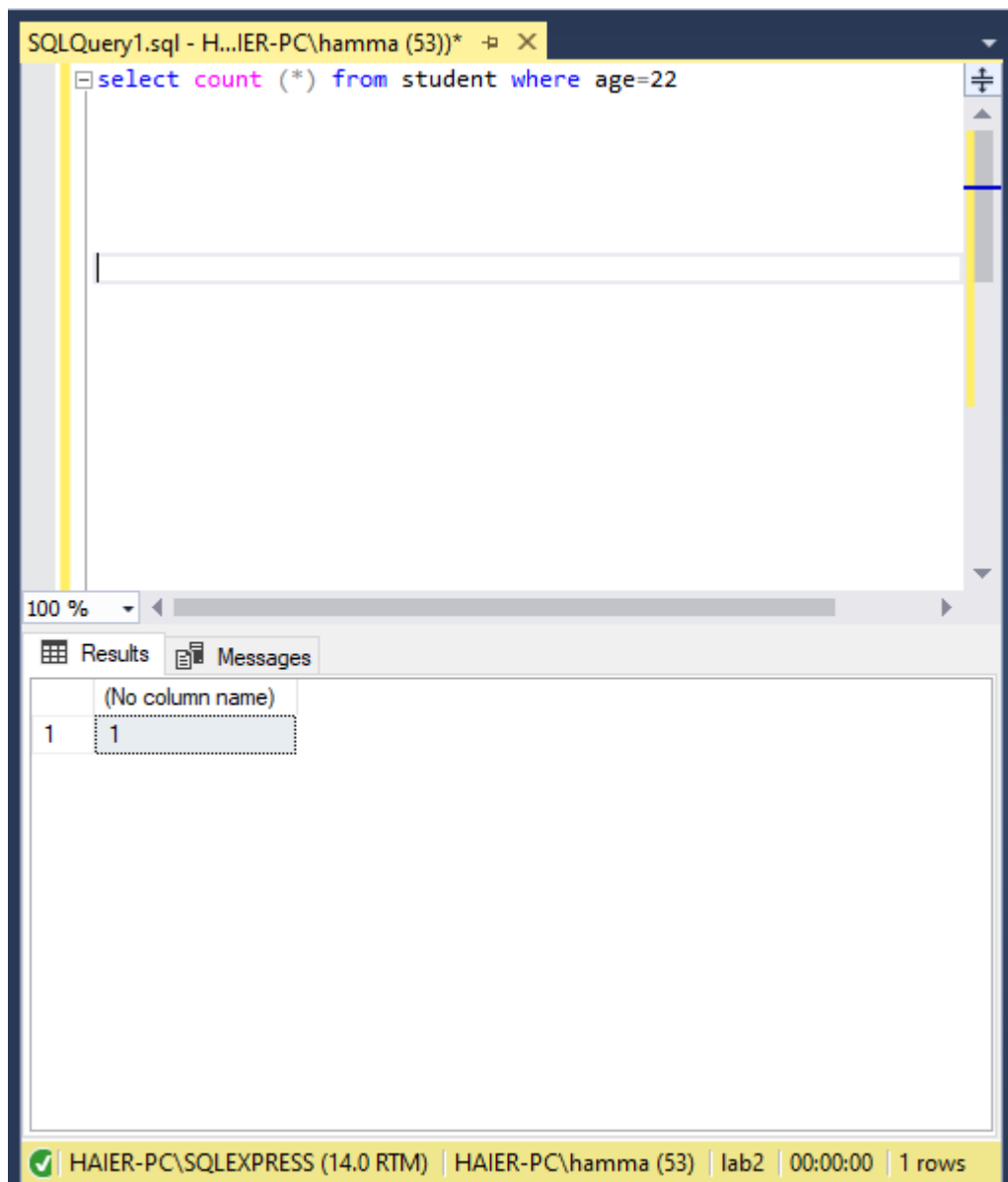




*Figure 40 Aggregation*

### Count function

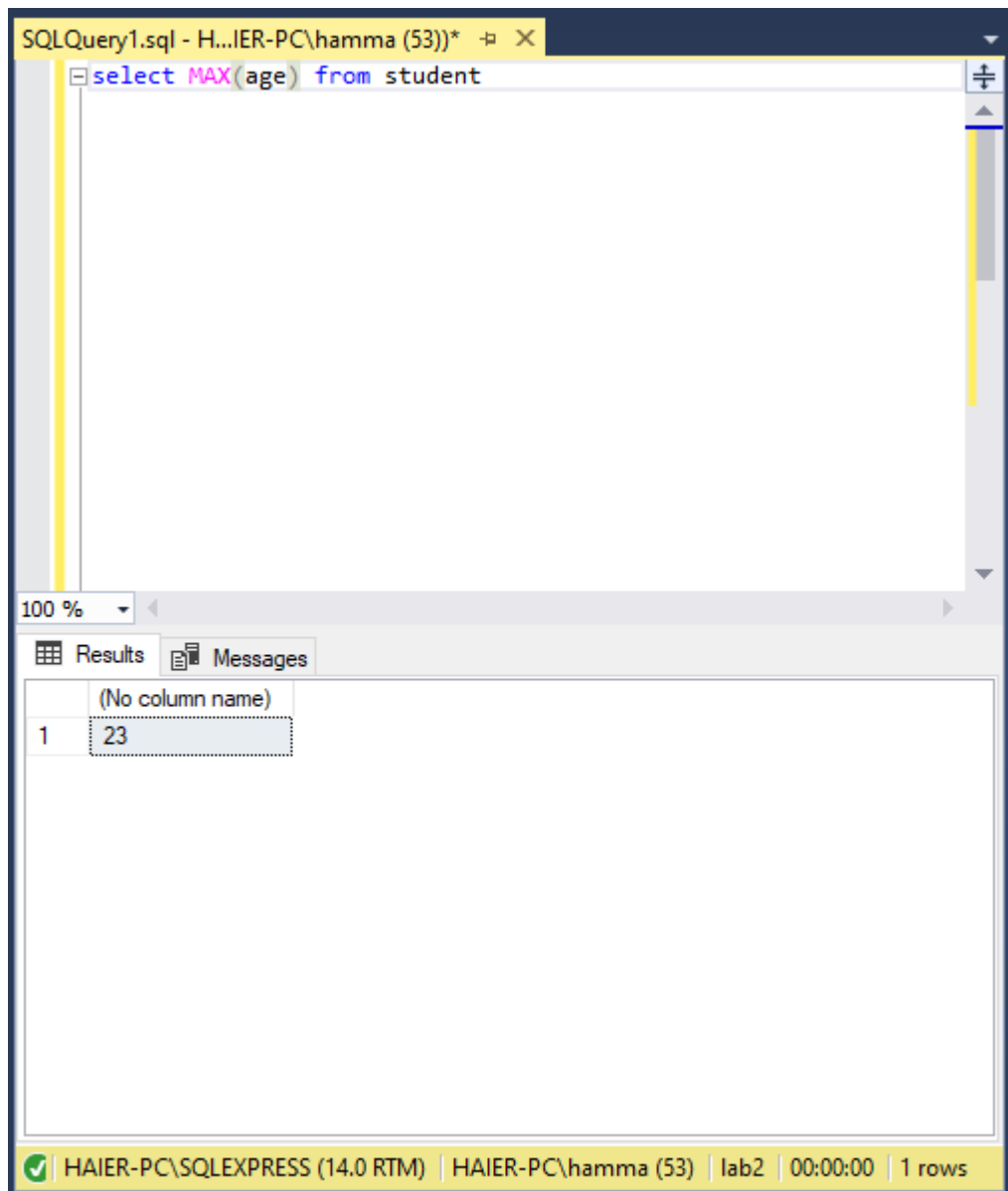
Count function is used to count the number of rows.



*Figure 41 Count*

### Max function

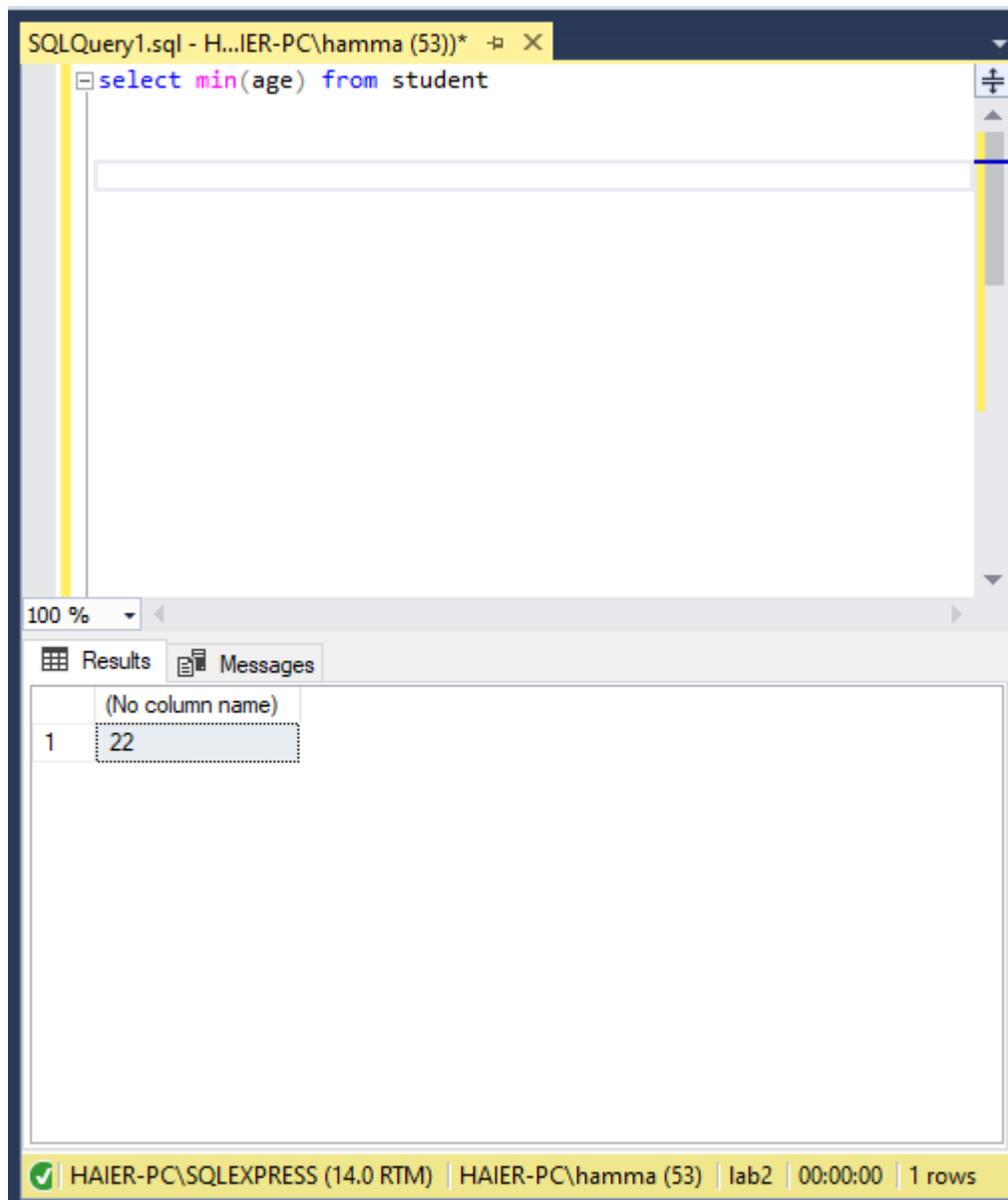
Max function is used to get maximum value from a column.



*Figure 42 Max Function*

### Min function

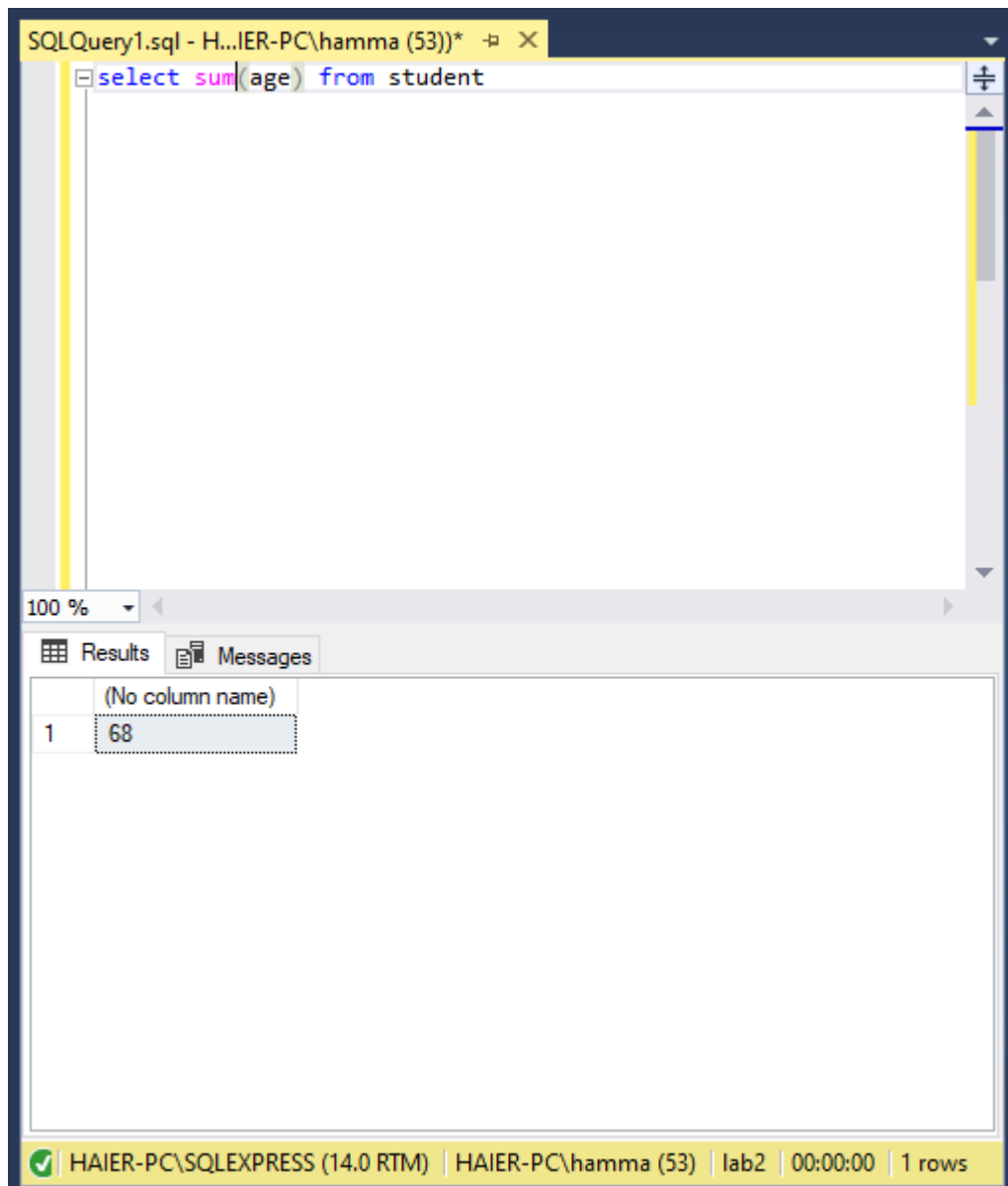
Min function is used to get minimum value from a column.



*Figure 43 Min Function*

### Sum function

Sum function is used to get sum of values from a column.



*Figure 44 Sum Function*

## Conclusion

After following the above steps, user would be able to use aggregation functions

## Task

Implement all above mentioned functions for teacher table.

## CS-363 Database Systems Lab 08

**Type of Lab: Open Ended**

**Weightage: 5%**

**CLO 1:** Construct DML queries to retrieve and store data in different relations.

Use sql server with C#	<b>Cognitive/Understanding</b>	CLO1	Rubric A
------------------------	--------------------------------	------	----------

### Rubric A: Cognitive Domain

Evaluation Method: GA shall evaluate the students for Question according to following rubrics.

CLO	0	1	2	3	4
CLO	Student is unable to the problem	Connection to server was successfully build	Queries are correct and properly executed	The task is partially correct	Output is according to query.

## Connecting SQL and C#

### Objectives

Understand how to connect sql server database with a c#.

### Processing steps

#### Import sql package

Create a c# console app. Import the package System.Data.SqlClient that contains useful classes and interfaces to access & work with database.

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using System.IO;
7  using System.Data.SqlClient;
8
9  namespace ConsoleApp3
10 {
11     class Program
12     {
13         static void Main(string[] args)
14         {
15         }
16     }
17 }
18

```

Figure 45 Import SQL Package

### Get Connection URL

To get a connection, we need to specify the URL of a database (Actually we need to specify the address of the database which is in the form of URL) which is done by

```
String conURL = "Data Source=(addressToServer);Initial Catalog= dbName;Integrated Security=True;MultipleActiveResultSets=True";
```

Example:

```

namespace ConsoleApp3
{
    class Program
    {
        static void Main(string[] args)
        {
            string conURL = "Data Source=HAIER-PC\\SQLEXPRESS;Initial Catalog=lab2;Integrated Security=True; MultipleActiveResultSets=True";
        }
    }
}

```

Figure 46 Get Connection URL

### Establish Connection With DataBase

Use SqlConnection to get the connection object. The URL of the database is passed to the SqlConnection constructor. You need to open the connection to use it.

```

class Program
{
    static void Main(string[] args)
    {
        string conURL = "Data Source=HAIER-PC\\SQLEXPRESS;Initial Catalog=lab2;Integrated Security=True; MultipleActiveResultSets=True";
        SqlConnection connection = new SqlConnection(conURL);
        connection.Open();
    }
}

```

Figure 47 Establish Connection With Database

## Prepare Command

The query is written in string than command is created from this query string and the connection. That command is then executed to get desired result. Following example is for a simple select command on our student table from previous lab

```
0 references
static void Main(string[] args)
{
    string conURL = "Data Source=HAIER-PC\\SQLEXPRESS;Initial Catalog=lab2;Integrated Security=True; MultipleActiveResultSets=True";
    SqlConnection connection = new SqlConnection(conURL);
    connection.Open();
    String query = "SELECT * FROM student";
    SqlCommand command = new SqlCommand(query, connection);
}
```

*Figure 48 Prepare Command*

## Execute Query

Two methods are generally used for executing SQL queries. These are:

- ExecuteReader() method Used for SQL SELECT queries. Returns the SqlDataReader object that contains the results of the query which are to be accessed. This is done by: `SqlDataReader reader = command.ExecuteReader();`
- command.ExecuteNonQuery() method This method is used for executing an update statement like INSERT, UPDATE or DELETE. Returns an Integer value representing the number of rows updated which is done by:
- `int count = command.ExecuteNonQuery();`

## Processing the result using sql Data Reader

The SqlDataReader object provides array index that takes a column index or name and returns the data as object which can then be casted into different types. The SqlDataReader object maintains the data in the form tables (rows & columns). First row has index 1, not 0. The read method of SqlDataReader object returns true or false depending upon whether the next row is available (exist) or not and moves the cursor. Always remember to call read() method at-least once. To retrieve the data of the column of the current row you need to use the various getters provided by the SqlDataReader object. For example, the following code snippet will iterate over the whole SqlDataReader object, read and print values from the first and second column:

```
string conURL = "Data Source=HAIER-PC\\SQLEXPRESS;Initial Catalog=lab2;Integrated Security=True; MultipleActiveResultSets=True";
SqlConnection connection = new SqlConnection(conURL);
connection.Open();
String query = "SELECT * FROM student";
SqlCommand command = new SqlCommand(query, connection);
SqlDataReader reader = command.ExecuteReader();
while (reader.Read())
{
    Console.WriteLine(reader[0] + " " + reader[1]);
}
```

*Figure 49 SQL data Reader*



## Close the Connection

An open connection is expensive, postpone this step if additional database operations are expected. This is done by:

```
string conURL = "Data Source=HAIER-PC\\SQLEXPRESS;Initial Catalog=lab2;Integrated Security=True; MultipleActiveResultSets=True";
SqlConnection connection = new SqlConnection(conURL);
connection.Open();
String query = "SELECT * FROM student";
SqlCommand command = new SqlCommand(query, connection);
SqlDataReader reader = command.ExecuteReader();
connection.Close();
```

*Figure 50 Close Connection*

## Conclusion

After following the above steps, user would be able to use sql data in c# for different type of tasks.

## Task

Create a signup and login system using c# and sql.

## CS-363 Database Systems Lab 09

**Type of Lab: Open Ended**

**Weightage: 10%**

**CLO 1:** Construct DML queries to retrieve and store data in different relations.

Displaying Data from Multiple Tables	<b>Cognitive/Understanding</b>	CLO1	Rubric A
--------------------------------------	--------------------------------	------	----------

### Rubric A: Cognitive Domain

Evaluation Method: GA shall evaluate the students for Question according to following rubrics.

CLO	0	1	2	3	4
CLO	Student is unable to the problem	There are errors in the query	Queries are correct	The task is partially correct	Output is according to query.

## Joins

### Objectives

Understand how use of joins.

### Processing steps

#### Joins

A JOIN clause is used to combine rows from two or more tables, based on a related column between them

#### Inner Join

The INNER JOIN keyword selects records that have matching values in both tables.

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

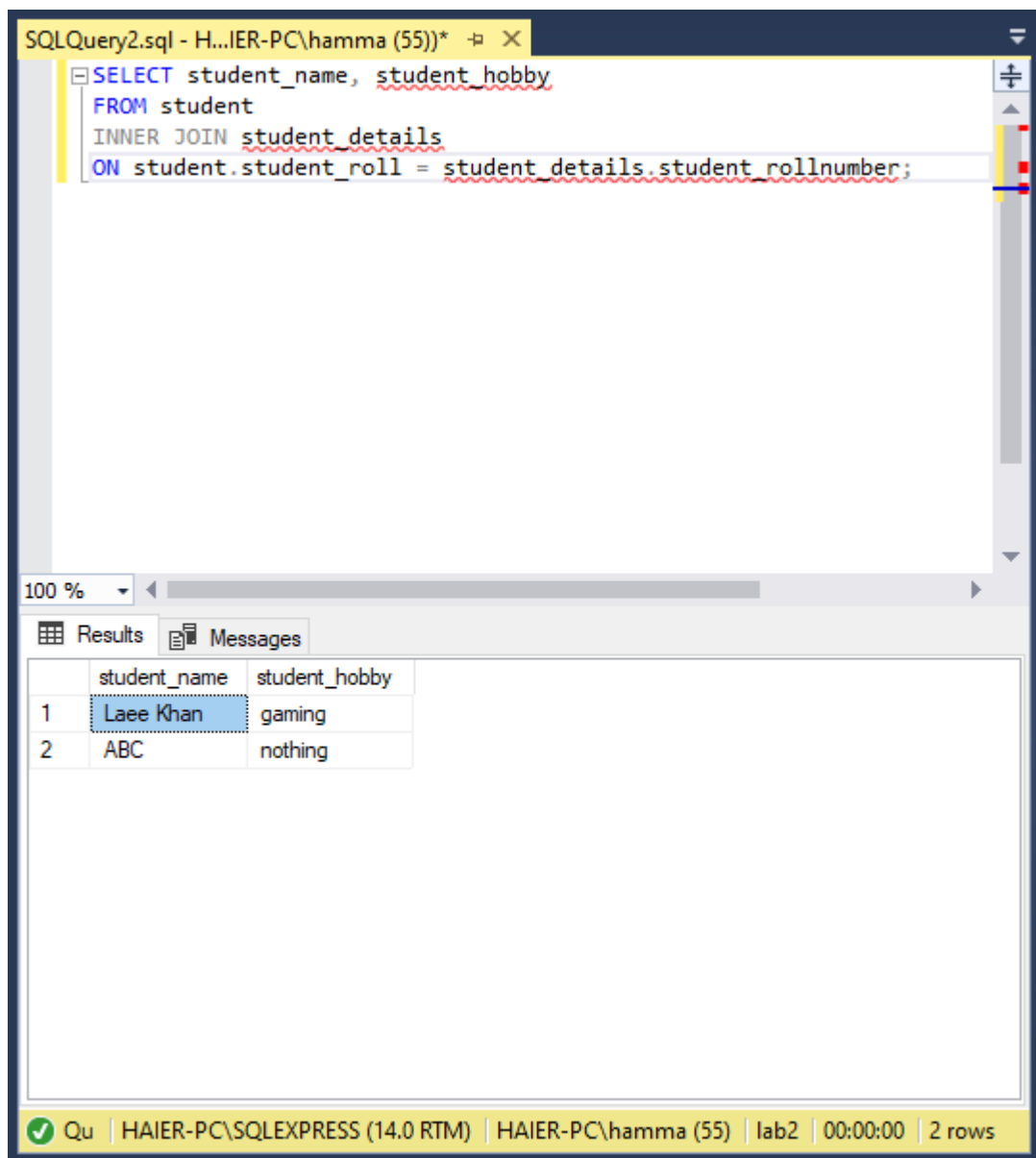


Figure 51 Joins

### Left Join

The LEFT JOIN keyword returns all records from the left table (table1), and the matched records from the right table (table2). The result is NULL from the right side, if there is no match.

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

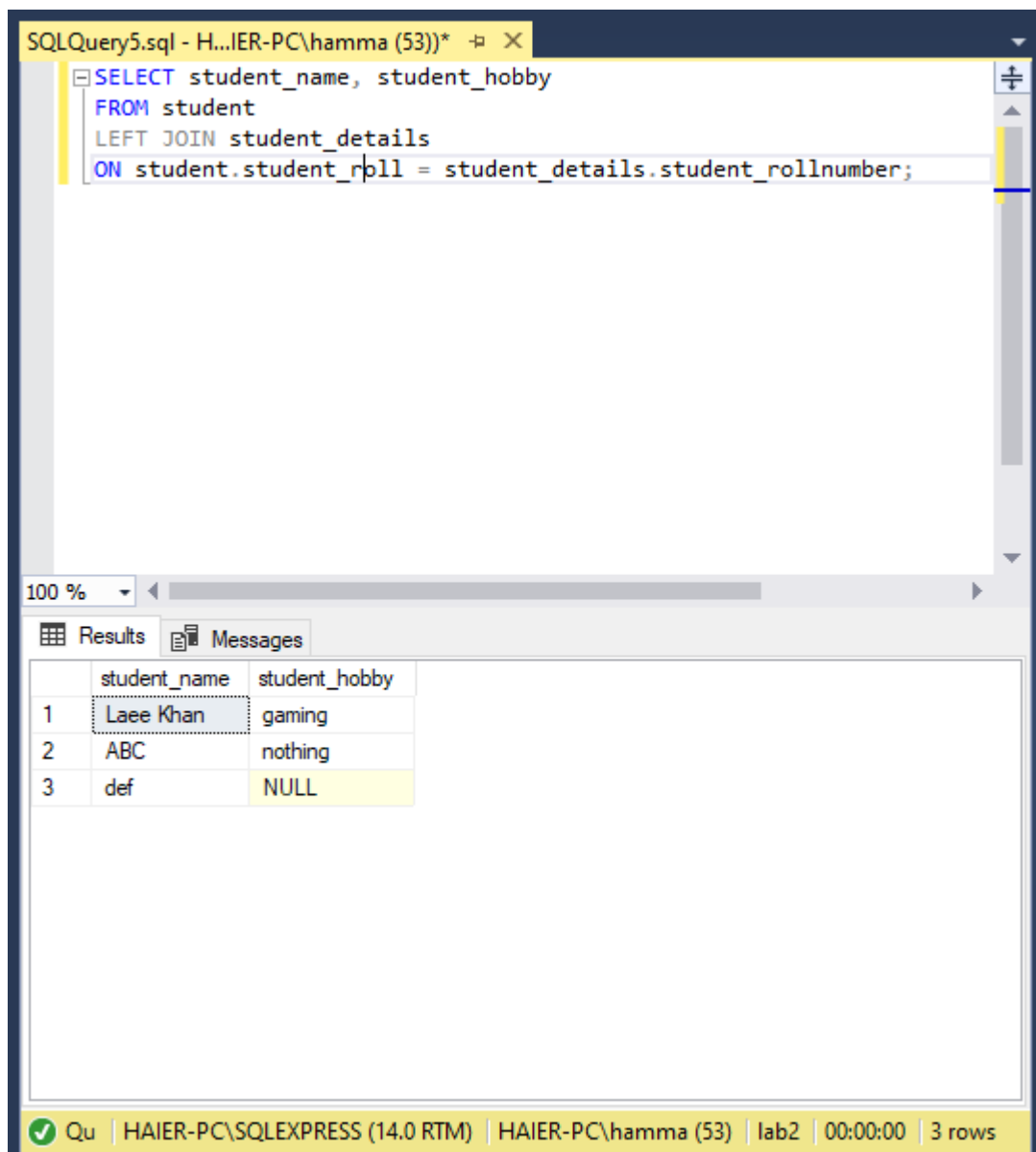


Figure 52 Left Join

### Right Join

The RIGHT JOIN keyword returns all records from the right table (table2), and the matched records from the left table (table1). The result is NULL from the left side, when there is no match.

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

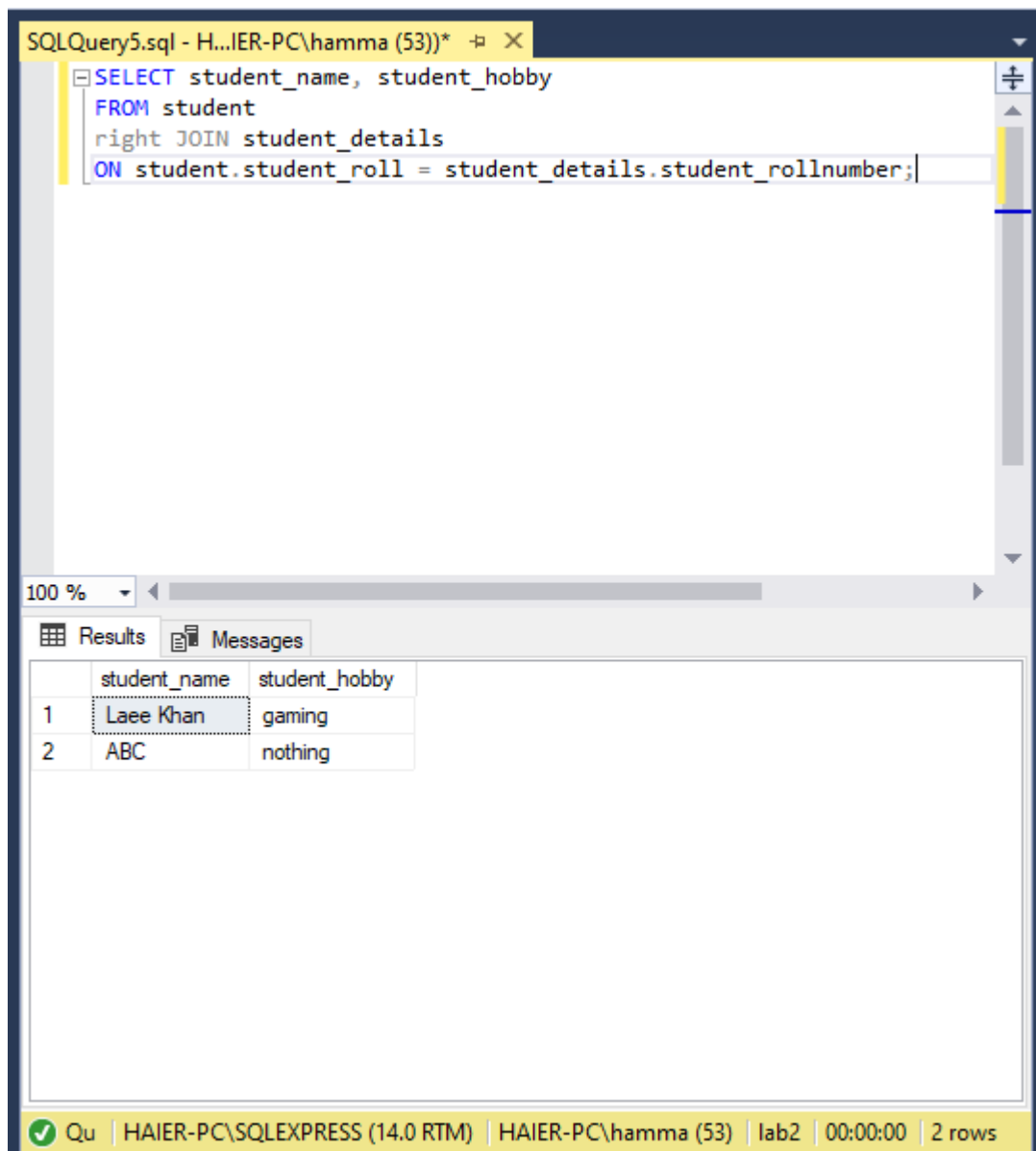


Figure 53 Right Join

### Full outer Join

The FULL OUTER JOIN keyword returns all records when there is a match in left (table1) or right (table2) table records.

```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name
WHERE condition;
```

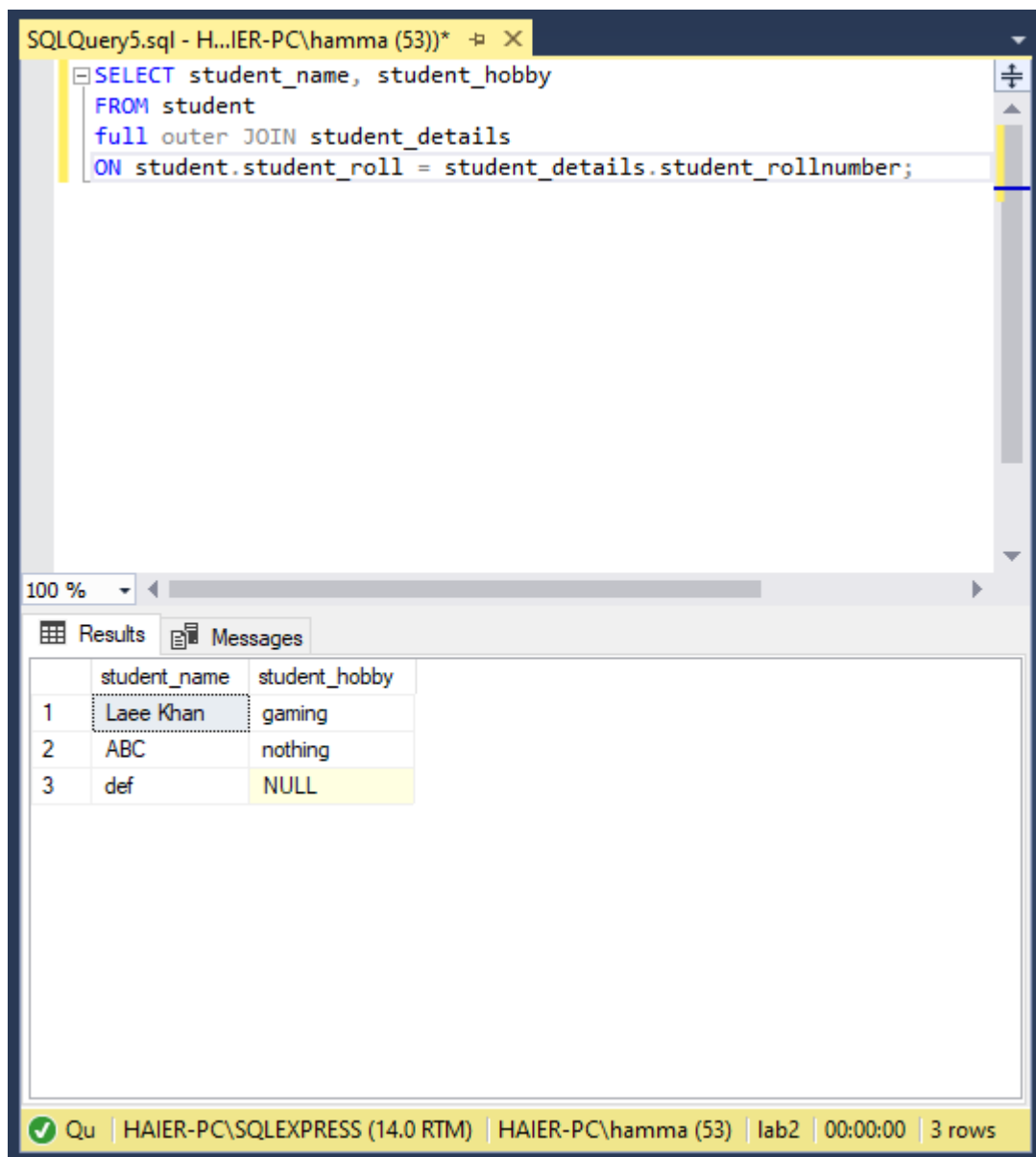


Figure 54 Full Outer Join

## Conclusion

After following the above steps, user would be able to use joins

## Task

Create a teacher\_details table and use joins to show combined data from teacher and teacher\_details table.

**CS-363 Database Systems Lab 10****Type of Lab: Open Ended****Weightage:10%****CLO 1:** Construct DML queries to retrieve and store data in different relations.

Using Sub queries to Solve Queries	<b>Cognitive/Understanding</b>	CLO1	Rubric A
------------------------------------	--------------------------------	------	----------

**Rubric A: Cognitive Domain**

Evaluation Method: GA shall evaluate the students for Question according to following rubrics.

CLO	0	1	2	3	4
CLO	Student is unable to the problem	There are errors in the query	Queries are correct	The task is partially correct	Output is according to query.

**Subqueries****Objectives**

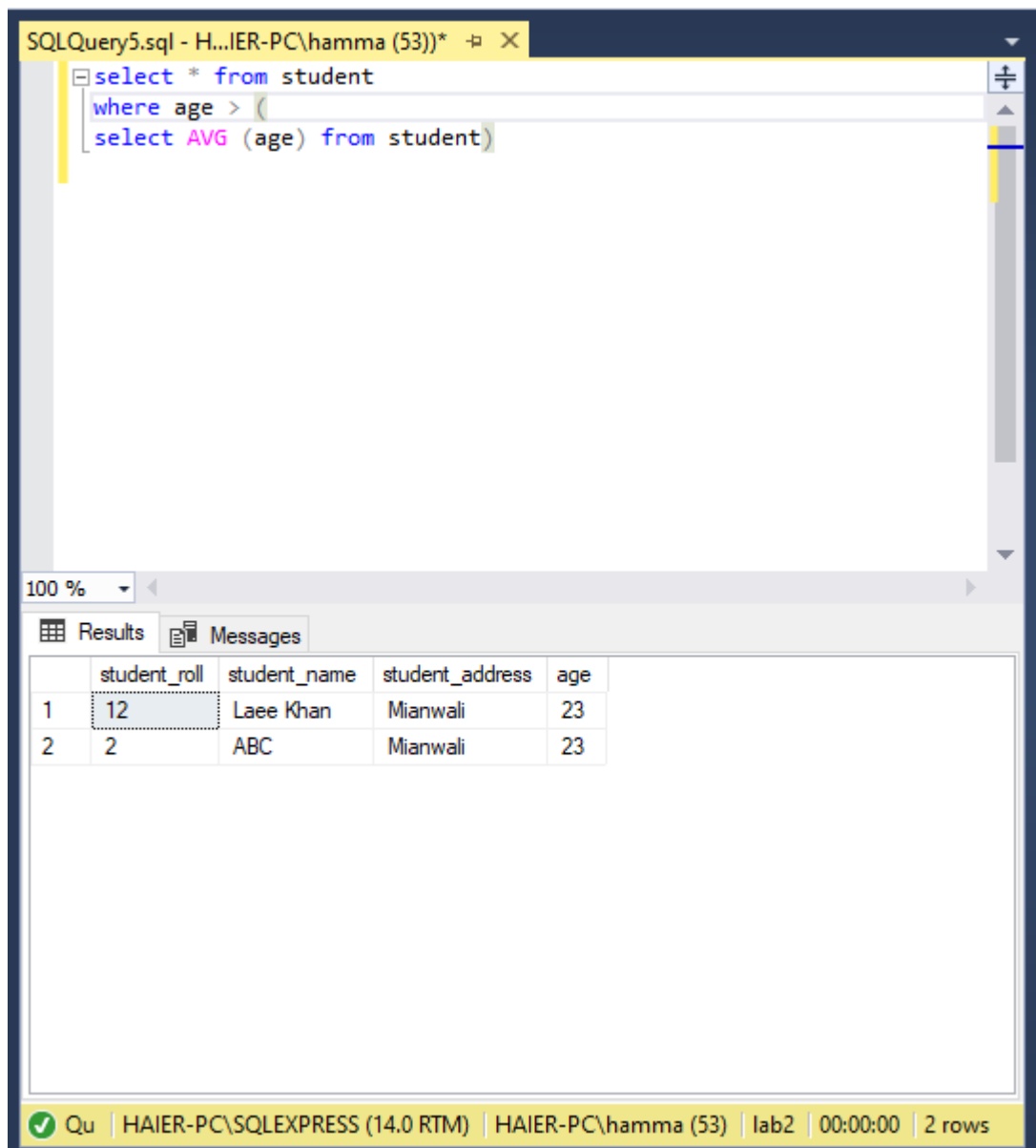
Understand how use of subqueries.

**Processing steps****Subquery**

A subquery is a query that is nested inside a SELECT, INSERT, UPDATE, or DELETE statement, or inside another subquery. A subquery can be used anywhere an expression is allowed.

Example:

In the following query we are selecting all data of students whose age is greater than average.



*Figure 55 Sub-Queries*

## Conclusion

After following the above steps, user would be able to use subqueries

## Tasks

Only select query example is shown in this lab. Implement the use of other queries in sub query structure.



## CS-363 Database Systems Lab 11

**Type of Lab: Open Ended**

**Weightage: 10%**

**CLO 2:** Construct DDL queries to manage relations, constraints and indexes

Creating Other Schema Objects, Views, Sequences etc	<b>Cognitive/Understanding</b>	CLO2	Rubric A
---	--------------------------------	------	----------

### Rubric A: Cognitive Domain

Evaluation Method: GA shall evaluate the students for Question according to following rubrics.

CLO	0	1	2	3	4
CLO	Student is unable to the problem	There are errors in the query	Queries are correct	The task is partially correct	Output is according to query.

## Views

### Objectives

Understand how use of views.

### Processing steps

### Views

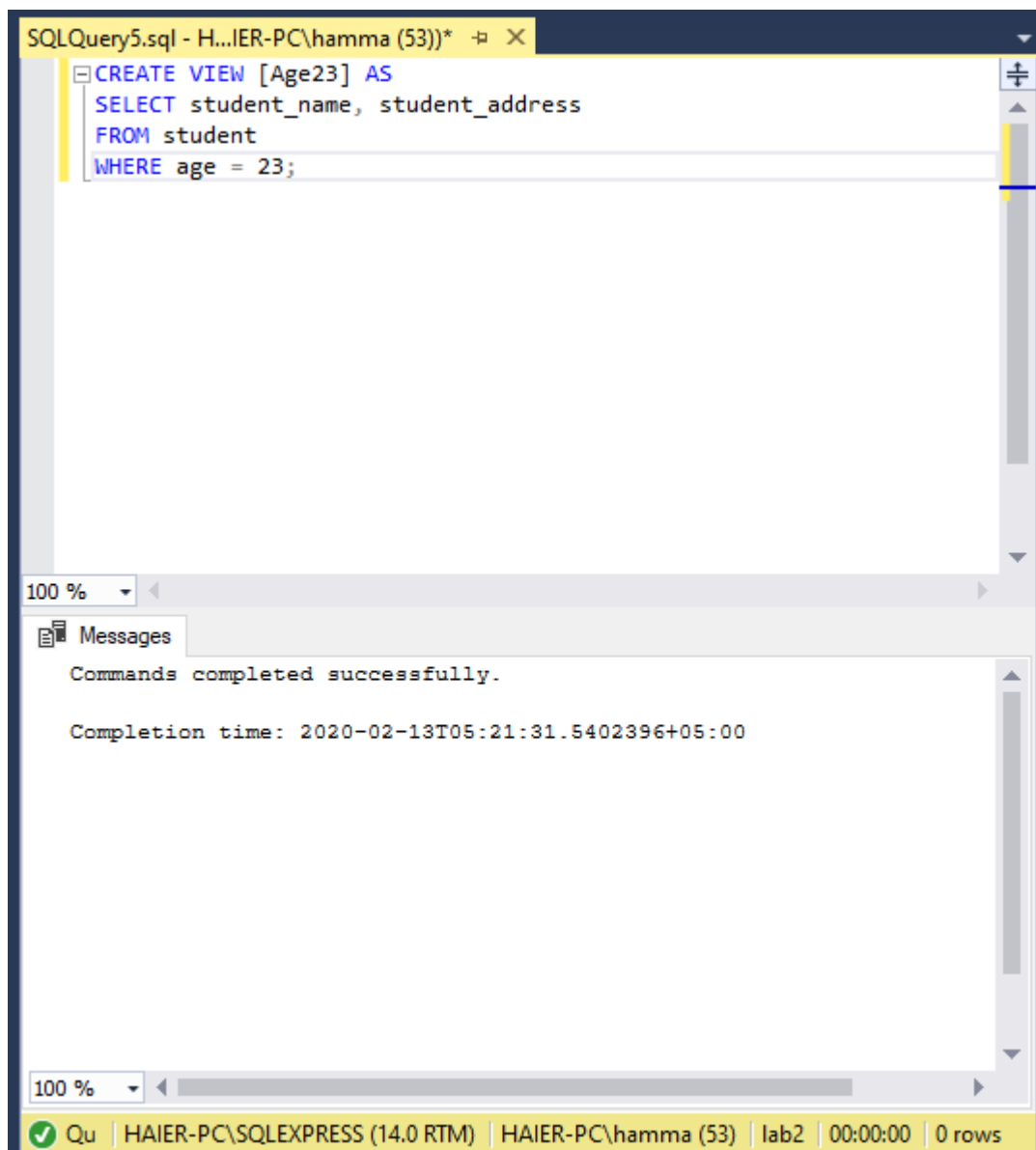
In SQL, a view is a virtual table based on the result-set of an SQL statement.

A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

You can add SQL functions, WHERE, and JOIN statements to a view and present the data as if the data were coming from one single table.

```
CREATE VIEW view_name AS
SELECT column1, column2,
FROM table_name
WHERE condition;
```

...

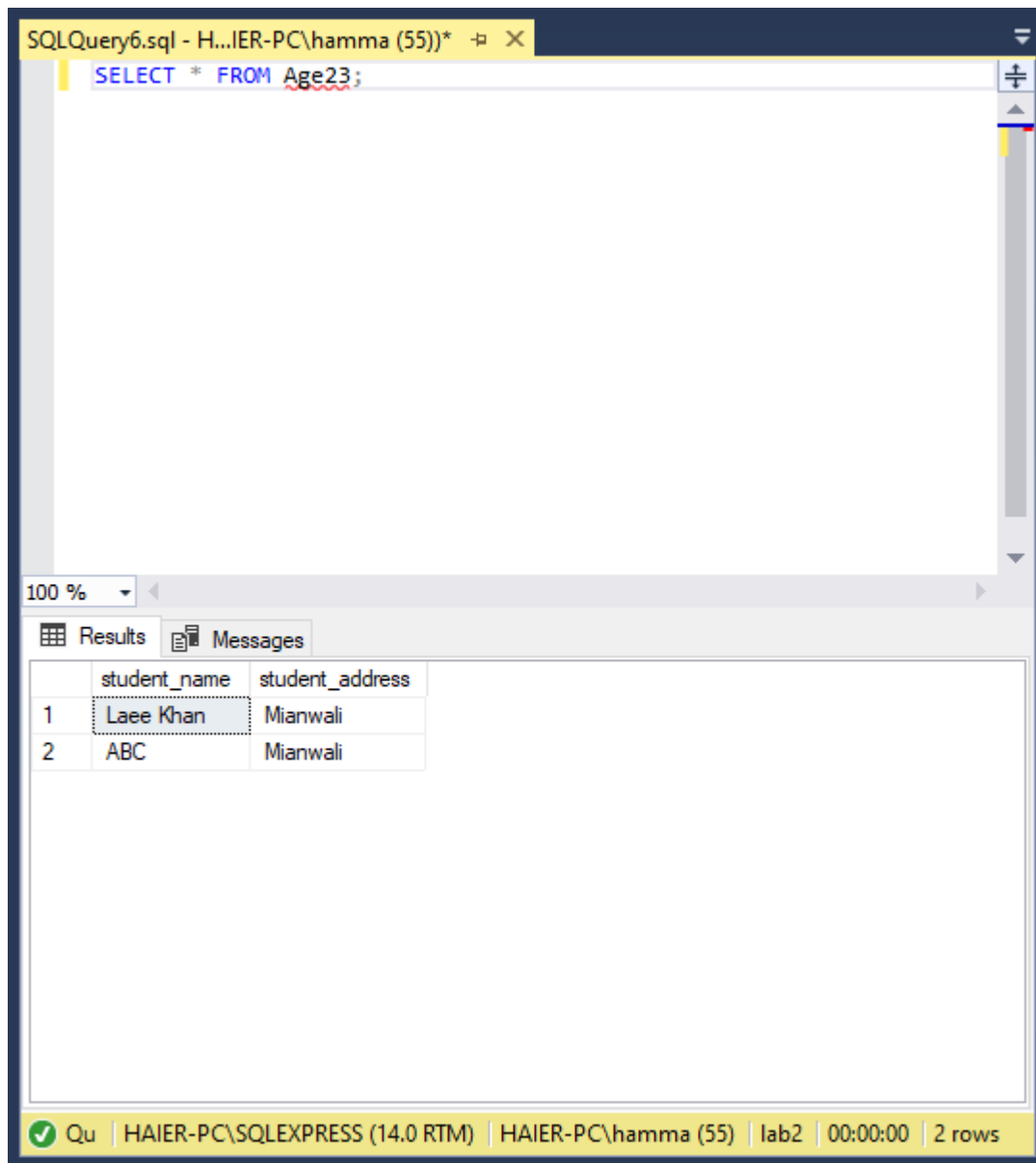


*Figure 56 Views*

## Retrieve data from View

We can retrieve data from views in the same as tables.

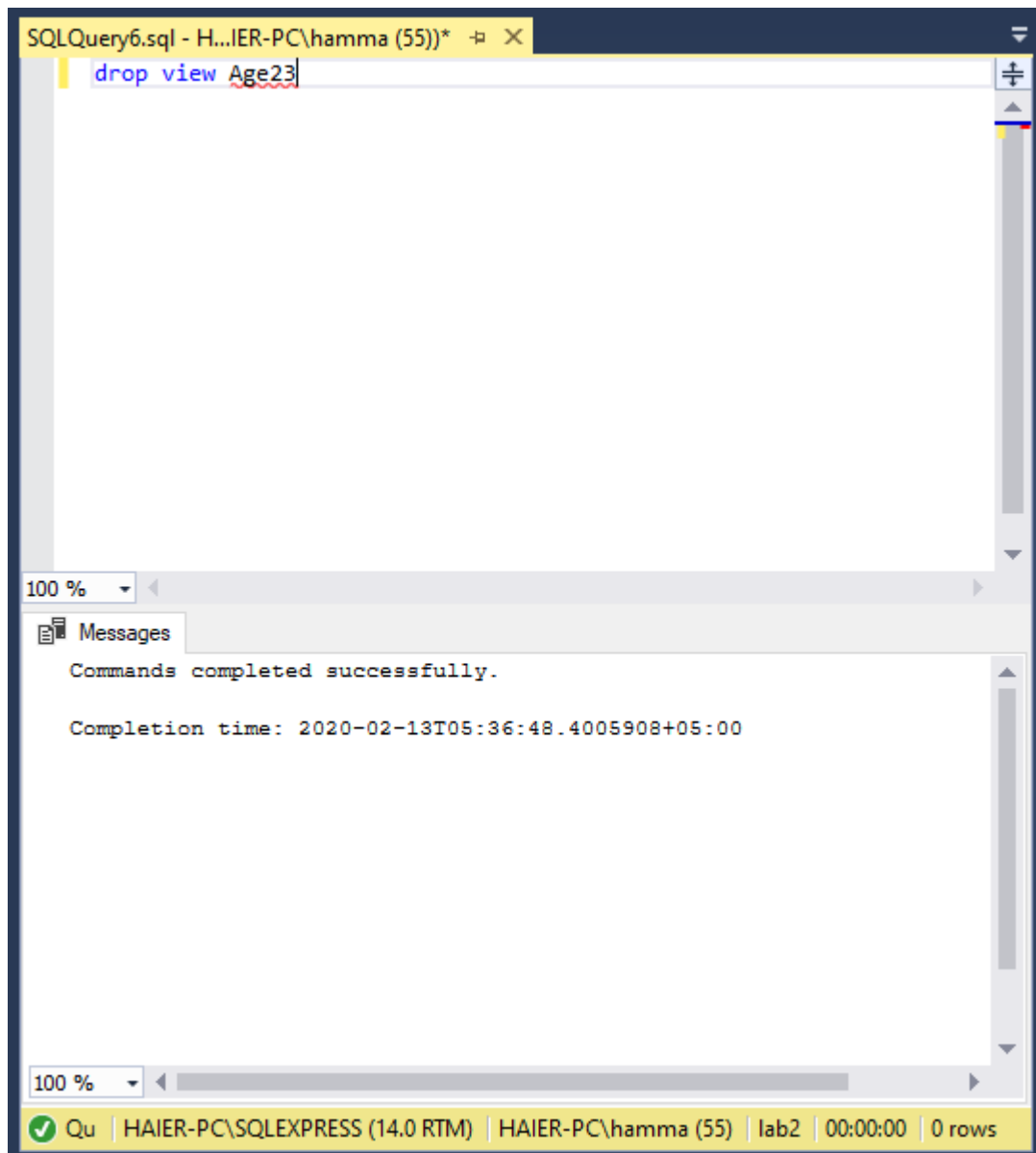
Example:



*Figure 57 Retrieve View*

## Drop Views

A view is deleted with the DROP VIEW command.

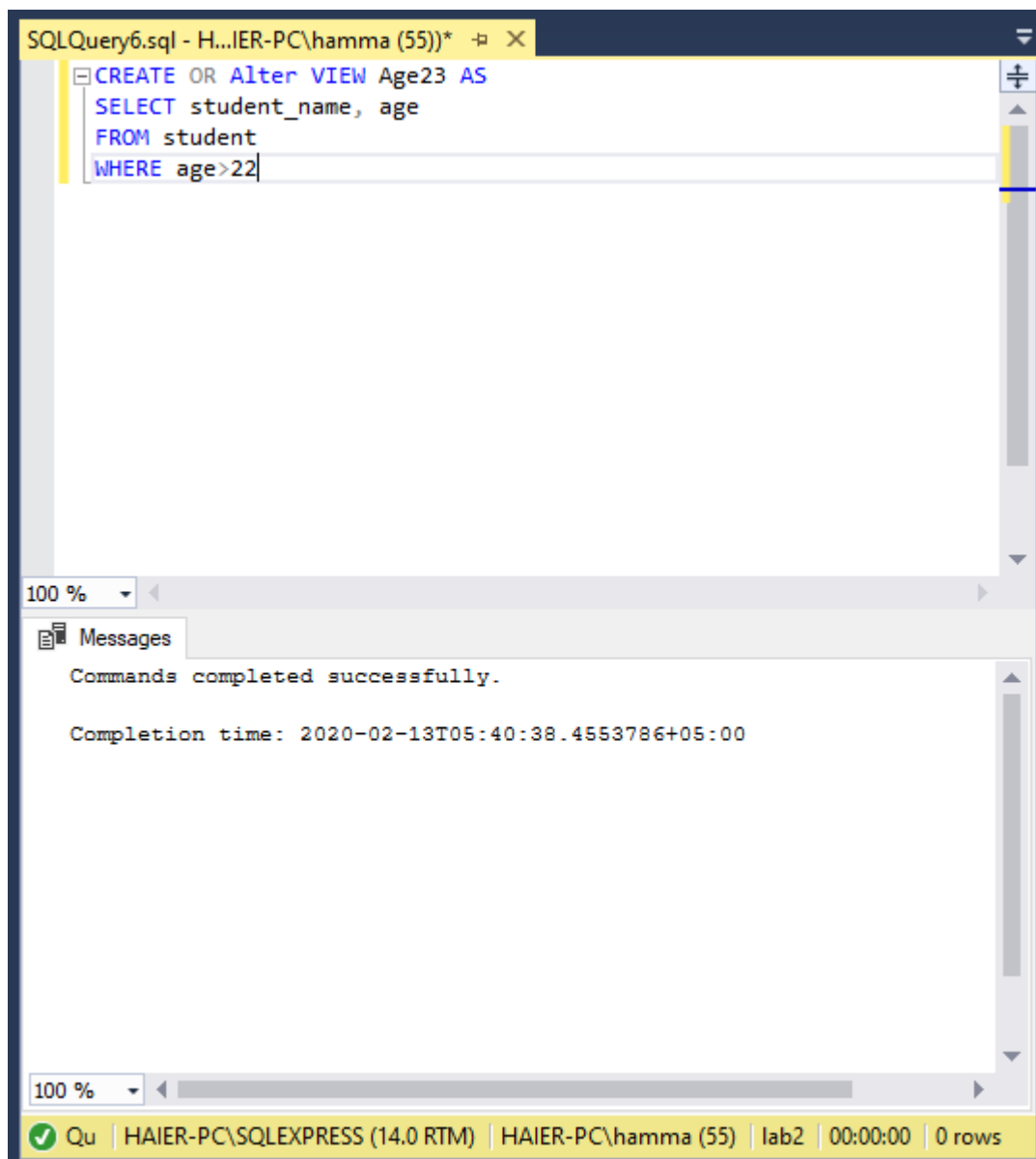


*Figure 58 Drop View*

## Update Views

A view can be updated with the CREATE OR ALTER VIEW command.

```
CREATE OR ALTER VIEW view_name AS  
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```



*Figure 59 Update View*

## Conclusion

After following the above steps, user would be able to use views

## Task

Demonstrate the use views with teacher table.

## CS-363 Database Systems Lab 12

**Type of Lab: Open Ended**

**Weightage: 10%**

**CLO 2:** Construct DDL queries to manage Procedural Language, Creating Procedures, Functions, Triggers.

Creating Procedural Language, Creating Procedures, Functions, Triggers	<b>Cognitive/Understanding</b>	CLO2	Rubric A
--	--------------------------------	------	----------

### Rubric A: Cognitive Domain

Evaluation Method: GA shall evaluate the students for Question according to following rubrics.

CLO	0	1	2	3	4
CLO	Student is unable to the problem	There are errors in the query	Queries are correct	The task is partially correct	Output is according to query.

### Objectives

Understand Procedural Language, Creating Procedures, Functions, Triggers.

### Processing steps

### Procedural Language

A procedural language is a computer programming language that follows, in order, a set of commands. Examples of computer procedural languages are BASIC, C, FORTRAN, Java, and Pascal.

Procedural languages are some of the common types of programming languages used by script and software programmers. They make use of functions, conditional statements, and variables to create programs that allow a computer to calculate and display a desired output.

## 1. The basic PL/SQL code structure

DECLARE -- optional, which declares and define variables, cursors. BEGIN -- mandatory SQL statements PL/SQL statements EXCEPTION -- optional, which specifies what actions to take when error occurs. END; -- mandatory	<b>DECLARE</b> <b>v1 NUMBER(3);</b> <b>BEGIN</b> <b>v1 := 3;</b> <b>DBMS_OUTPUT.PUT_LINE('v1='    v1);</b> <b>END;</b>
--	---

## 2. Control Structures

IF <condition> THEN [ELSIF <condition> THEN] [ELSE <condition> THEN] END IF;	LOOP ... EXIT WHEN <condition> END LOOP;	FOR counter IN lower_bound .. upper_bound ... END LOOP;
---	---	---

## 3. Cursors

A SQL cursor is a private Oracle SQL working area. We can perform programming task on these.

```

DECLARE
CURSOR csr_ac (p_name VARCHAR2) IS
SELECT empno, name, sal
FROM employee
WHERE name LIKE '%p_name%';
v_a employee.empno%TYPE; //its type will be type of filed
v_b employee.name%TYPE;
v_c employee.sal%TYPE;
BEGIN
OPEN csr_ac ('LE');
LOOP
FETCH csr_ac INTO a, b, c;
EXIT WHEN csr_ac%NOTFOUND;
DBMS_OUTPUT.PUT_LINE(v_a || ' ' || v_b || ' '||v_c);
END LOOP;
CLOSE csr_ac;
  
```

## 4. SQL Procedure

```

PROCEDURE hire_employee (emp_id INTEGER, name VARCHAR2) IS
BEGIN
  
```

```
INSERT INTO employee VALUES (emp_id, name, 1000);
END hire_employee;
```

## 5. Function

```
FUNCTION sal_ok (salary REAL, title REAL) RETURN BOOLEAN IS
min_sal REAL;
max_sal REAL;
BEGIN
SELECT losal, hisal INTO min_sal, max_sal
FROM sals
WHERE job = title;
RETURN (salary >= min_sal) AND (salary <= max_sal);
END sal_ok;
END;
```

## 6. Triggers

Triggers are stored programs, which are automatically executed or fired when some events occur. Triggers are in fact, written to be executed in response to any of the following events:

- A database manipulation (DML) statement (DELETE, INSERT, or UPDATE).
- A database definition (DDL) statement (CREATE, ALTER, or DROP).
- A database operation (SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN).
- Triggers could be defined on the table, view, schema, or database with which the event is associated.

### Trigger Example:

ID	NAME	AGE	ADDRESS	SALARY
----	------	-----	---------	--------

Let's consider following table

we want whenever we insert,delete or update records in table we want to output the information

at console or me sent as text message

```
CREATE OR REPLACE TRIGGER display_salary_changes
BEFORE DELETE OR INSERT OR UPDATE ON customers
FOR EACH ROW
WHEN (NEW.ID > 0)
DECLARE
sal_diff number;
BEGIN
sal_diff := :NEW.salary - :OLD.salary;
dbms_output.put_line('Old salary: ' || :OLD.salary);
dbms_output.put_line('New salary: ' || :NEW.salary);
dbms_output.put_line('Salary difference: ' || sal_diff);
END;
```



**Conclusion**

After following the above steps, user would be able to understand Procedural Languages, Create Procedures, Functions and Triggers.

**Task**

Demonstrate the use views with teacher table.

**Project A****Weightage: 10%**

Department of Computer Science UET Lahore holds committee for management of final year project. Each year, list of project titles is opened for the students by the committee after the compilation of ideas from the faculty members. Students are privileged to form the group and select any of the ideas from the list after approval from the faculty advisor. Once the project is selected by a group of students, an advisory board is assigned to the project which consists of main advisor, co-advisor, and industry advisor. Throughout the year, multiple evaluations are taken against the project. Currently, this process is managed through spread sheets. You are supposed to streamline the above process and develop desktop application (Windows Form Application using C#) that will be operated by the member of project committee to manage data at one place. Following features will be implemented in the application.

- Manage Students
- Manage Advisors
- Manage Projects
- Formation of Student Group and its management
- Assignment of project to a group of students
- Assignment of multiple advisors to the project
- Manage Evaluations
- Mark the evaluations against a group

Any other feature that can be helpful for the management of final year projects after discussion with the committee. Final year project committee also requires multiple reports in pdf form that may include.

- List of projects along with advisory board and list of students
- Marks sheet of projects that shows the marks in each evaluation against each student and project

Any other reports that you can help the committee to streamline the process.

**Note:**

Database scripts are available at <http://bit.ly/ProjectADb>

## Project B

**Weightage: 10%**

Department of Computer Science UET Lahore follows the Outcome Based Education where each subject is mapped with multiple CLOs. For the Lab work, these CLOs are further mapped to multiple rubrics. Rubrics are the rules that measure the students at different levels in particular component of an assessment. Example of rubrics for object-oriented programming are as follow:

Criteria	Sub-criteria	Exceptional	Good	Fair	Unsatisfactory
Level		4	3	2	1
CLO 2: Implement abstraction and encapsulation to develop reusable classes for objects of real world problems	<b>Design</b> Program should be properly decomposed in reusable components. That either be functions , classes or files or or any other paradigm as per the course requirement	Functionalities are divided properly in coherent and cohesive components	Functionalities are divided into proper coherent units but the are either redundant or lack cohesion	Code is divided into modules but no consideration is put into reusability and cohesion of the modules	No such division of responsibility is visible in the code structure
	<b>Execution</b> Code is correct, the required programming techniques are implemented accurately according to rules of language.	No Errors, programs compiles and executes perfectly and efficiently	Program does compiles but could have been coded in more efficient way	Program does not compiles have minor errors due to missing semicolons or mis-alignments or missing brackets or any such issue	Program does not compile or interpret due to lack of syntax knowledge
	<b>Testing</b> Program executes and all scenarios are tested with no logical errors	All test cases are clear for functionalities and their boundary conditions	All test cases are clear for functionalities but might show erroneous behaviour on boundary conditions	Majority of the test cases are clear, but there might be few failed ones	Majority test cases are failed

*Figure 60 Project Criteria*

Each student is being evaluated against rubrics in each assessment. For example, Lab1 has three components/Questions with 10 total marks and these questions are mapped with the rubrics as follow:

Component	Rubric	Component Marks
Question 1	Design	3
Question 2	Execution	4
Question 3	Testing	3

Now the student X is evaluated against assessment and student will be assigned with the rubric level. Based on the rubric level, obtained marks will be automatically calculated according to following formula

$$\text{ObtainedMarks} = \frac{\text{ObtainedRubricLevel}}{\text{MaxRubricLevel}} \times \text{Component Marks}$$

Component	Rubric	Component Marks	Student Rubric Level	Obtained Marks (Automatically calculated)
Question 1	Design	3	2	1.5
Question 2	Execution	4	3	3
Question 3	Testing	2	3	1.5

Currently, the above process is being managed manually. You are supposed to streamline the above process and develop desktop application(Windows Form Application using C#) that will be operated by the teacher to manage data at one place. Following features will be implemented in the application.

- Manage Students
- Manage CLOs
- Manage Rubrics
- Manage Assessments
- Manage Rubric Levels
- Mark the evaluations against a student

And any other feature that can be helpful for the management of evaluations. Instructor also requires multiple reports in pdf form that may include.

And any other feature that can be helpful for the management of evaluations. Instructor also requires multiple reports in pdf form that may include.

- CLO wise class result
- Assessment wise class result

And any other reports that you can help the committee to streamline the process.

**Note:**

Database scripts are available at <http://bit.ly/ProjectADb>