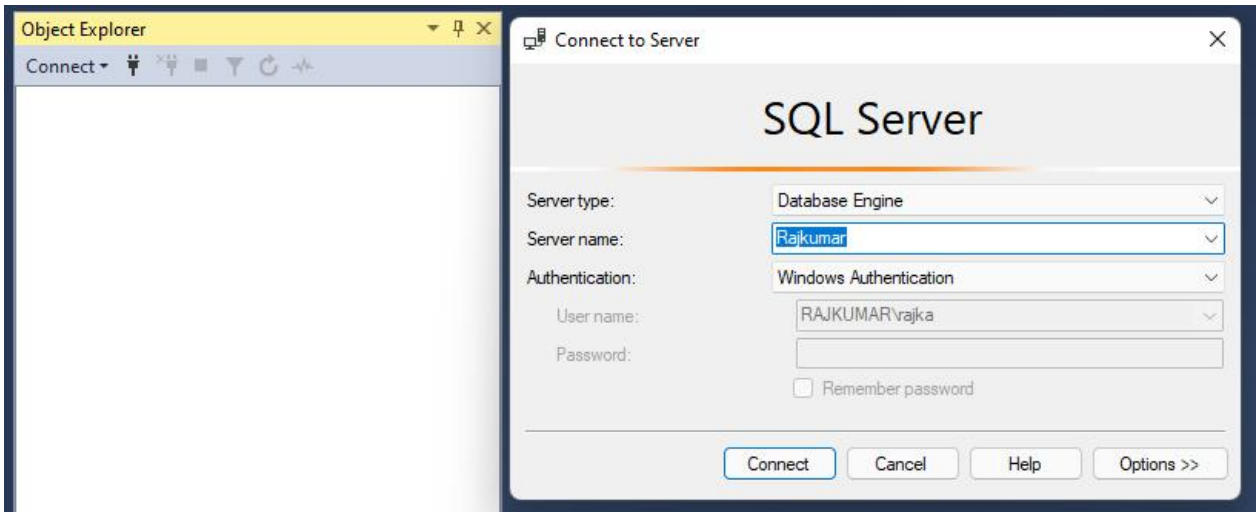
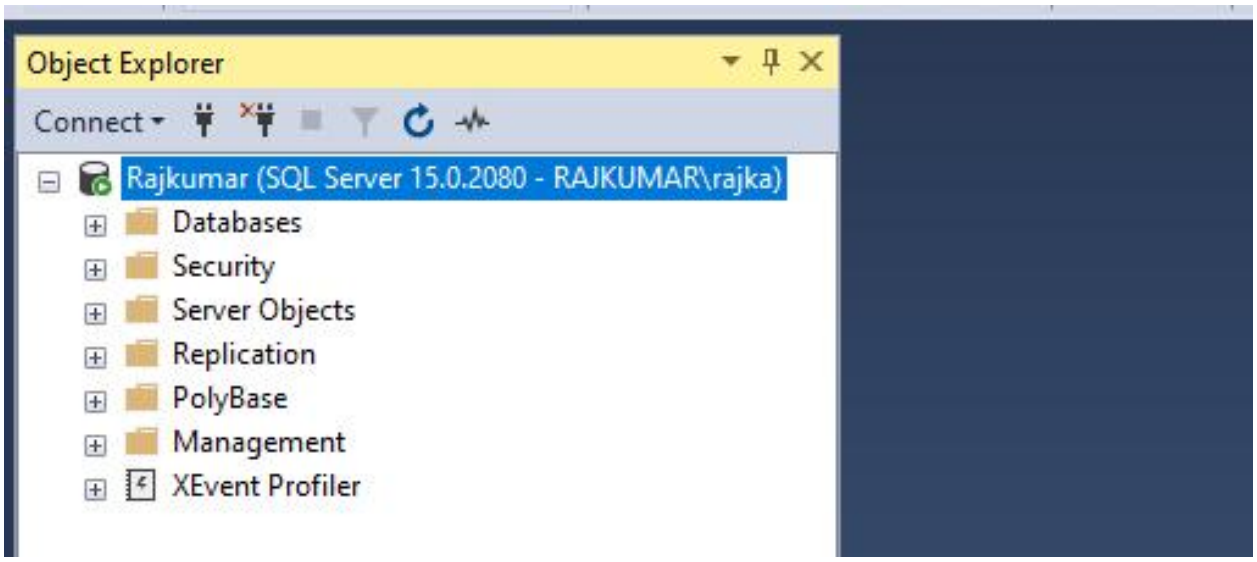
Connecting to server:



Connected to Server:

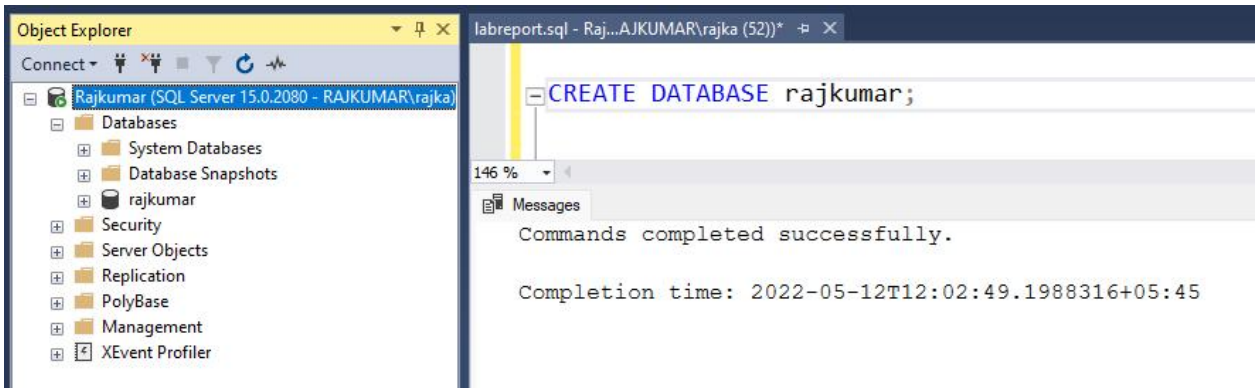


1. Creating and removing database queries

i. Synatx: (CREATE)

CREATE DATABASE db_name;

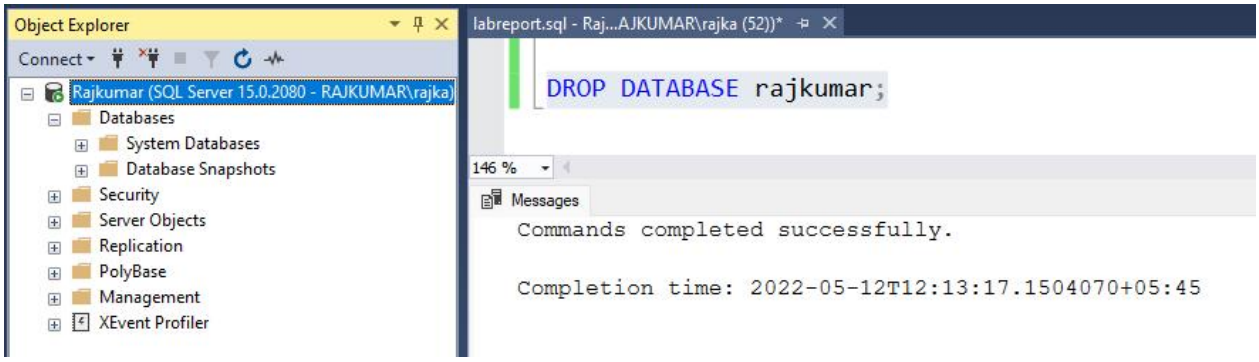
Query with Output:



ii. Syntax: (DROP)

DROP DATABASE db_name;

Query with Output:



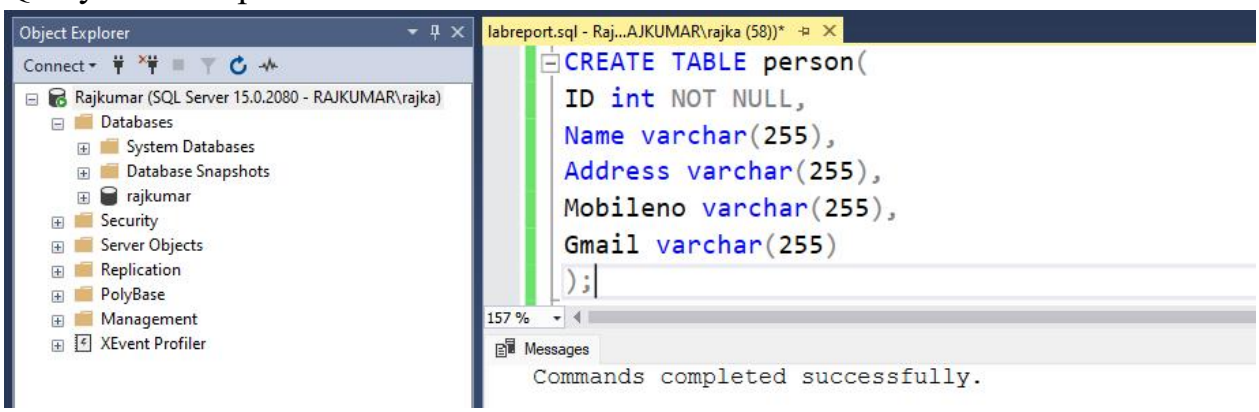
Handwritten Code:

2. Creating and Removing Table Queries

i. Syntax: (CREATE)

CREATE TABLE tb_tablename(
column_name1,
Column_name2…);

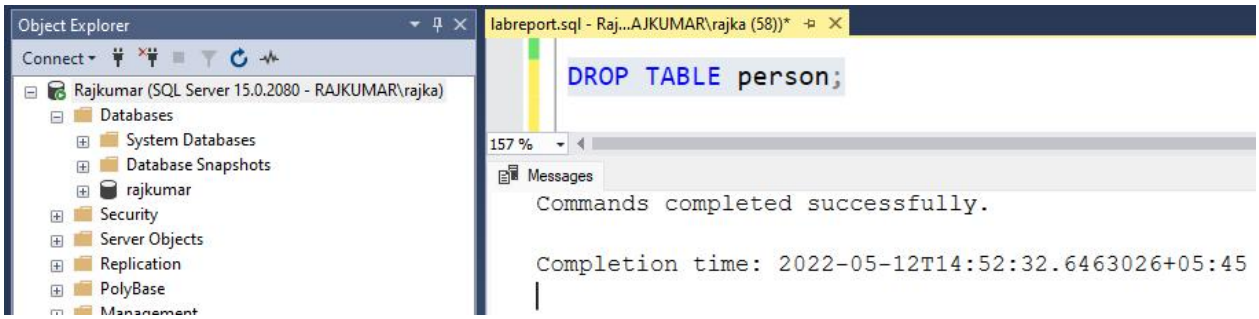
Query with Output:



ii. Syntax: (DROP)

DROP TABLE tb_tablename;

Query with Output:



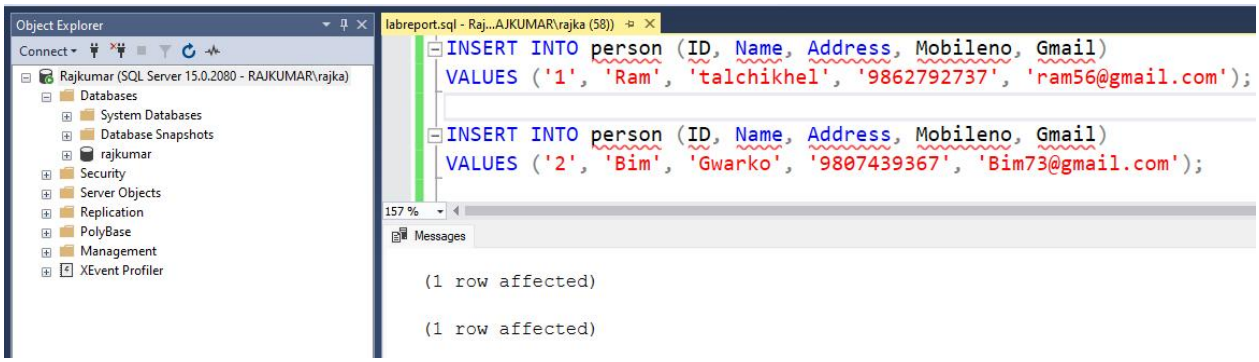
Handwritten Code:

3. Insert into the table, select all, and select distinct from the table.

i. Syntax: (INSERT INTO)

INSERT INTO table_name(column1, column2, column3,…)
VALUES (value1, value2, value3,....);

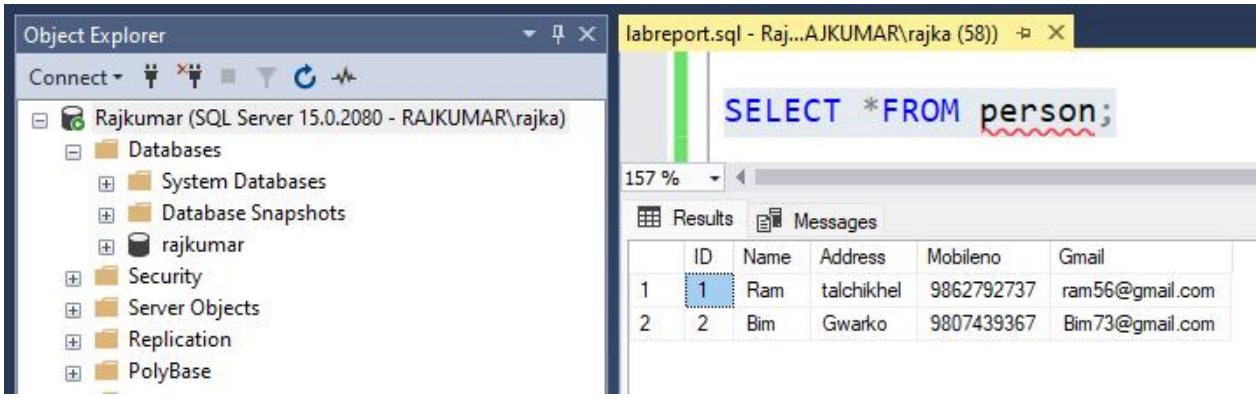
Query with Output:



ii. Syntax: (SELECT)

SELECT * FROM table_name;

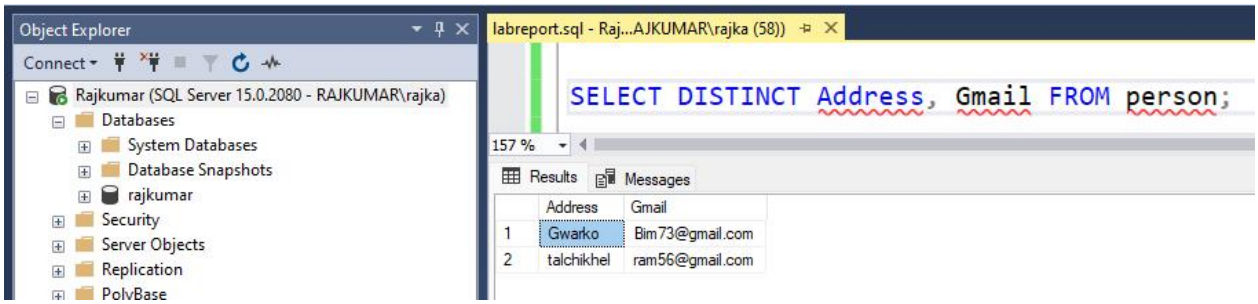
Query with Output:



iii. Syntax: (SELECT DISTINCT)

SELECT DISTINCT column1, column2, …. FROM table_name;

Query with Output:



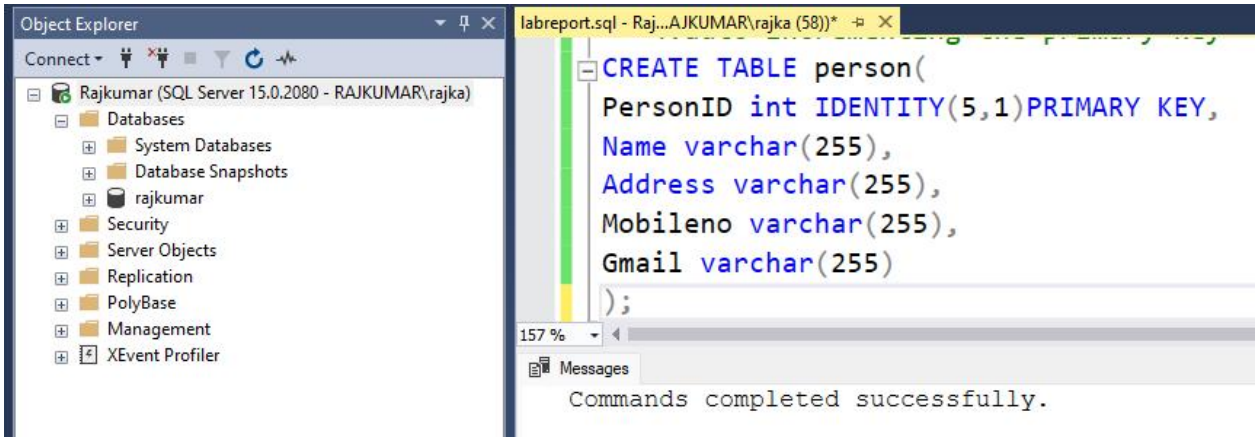
Handwritten Code:

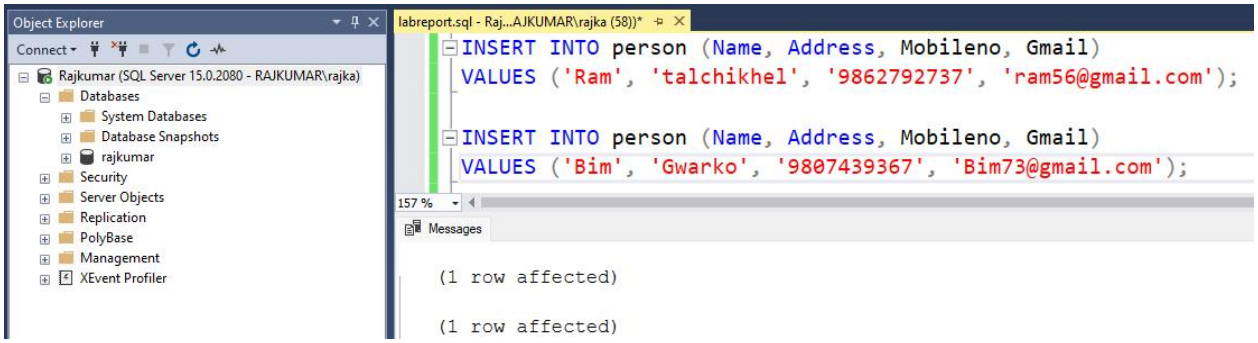
4. To show the auto increment in the primary key

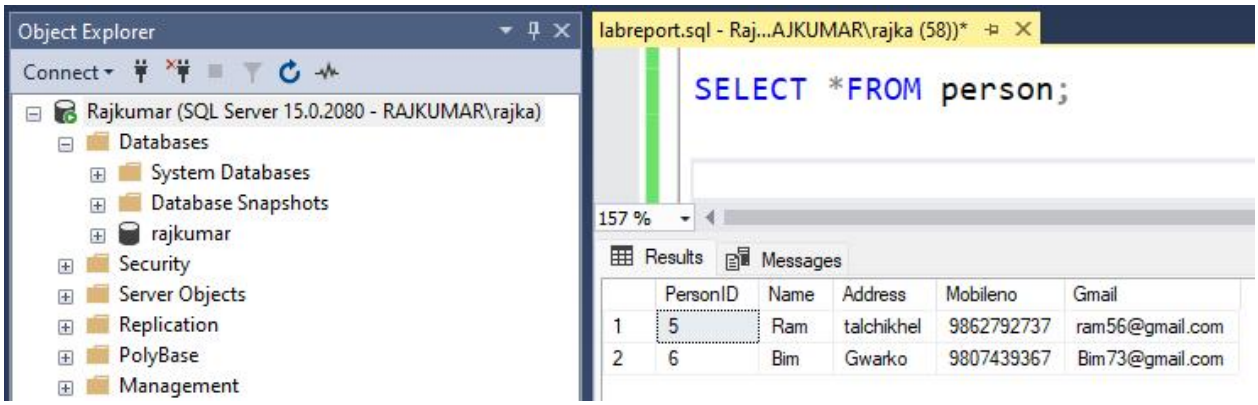
i. Syntax: (AUTO INCREMENT)

CREATE TABLE tb_tablename(
Column 1 datatype INDENTITY(starting number, increment by number) PRIMARY KEY,
Coulmn 2,
Coulmn 3….);

Query with Output:

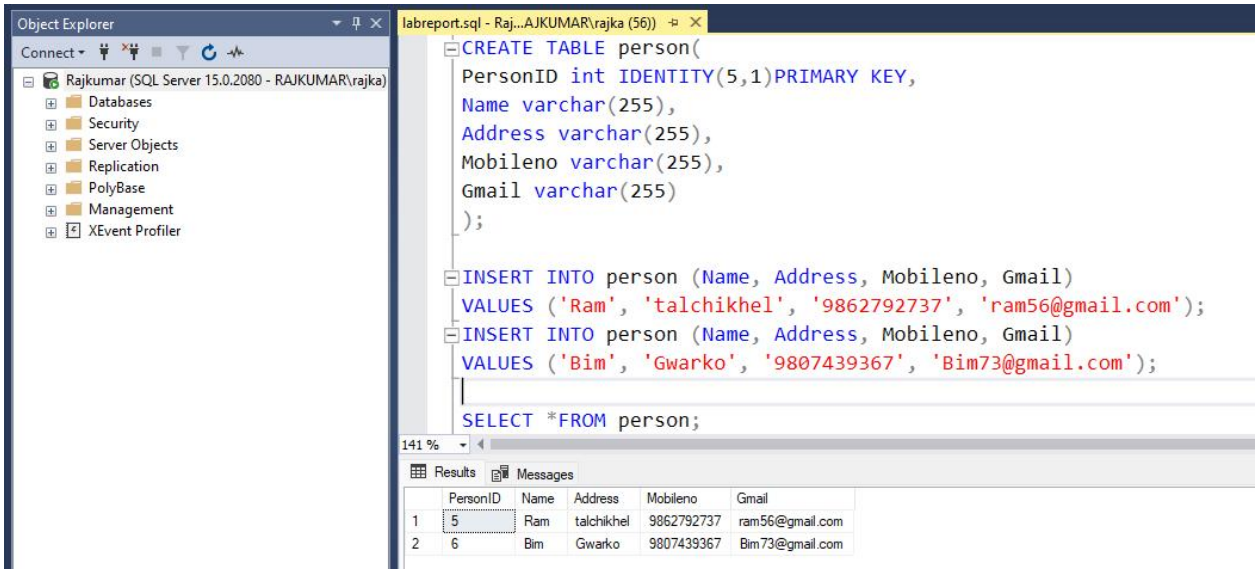






Handwritten Code:

5. Altering the table and adding a column to the table

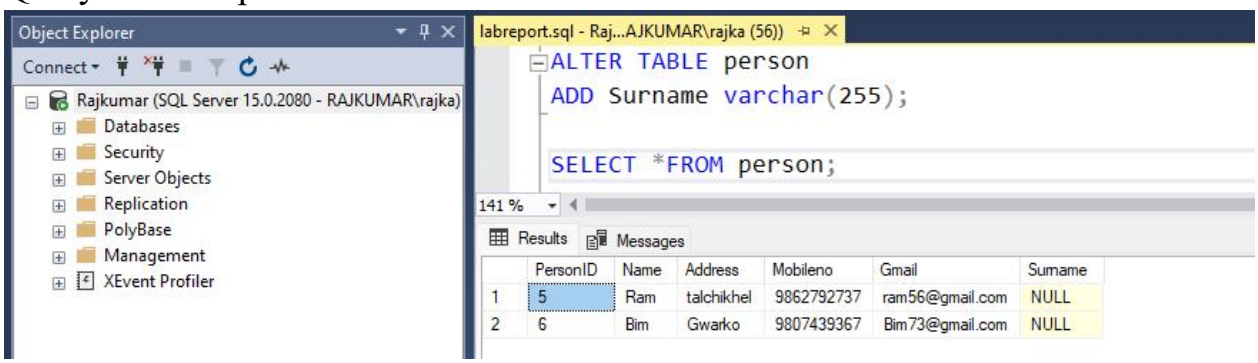


Syntax:

ALTER TABLE tb_tablename

ADD column_name datatype ;

Query with Output:



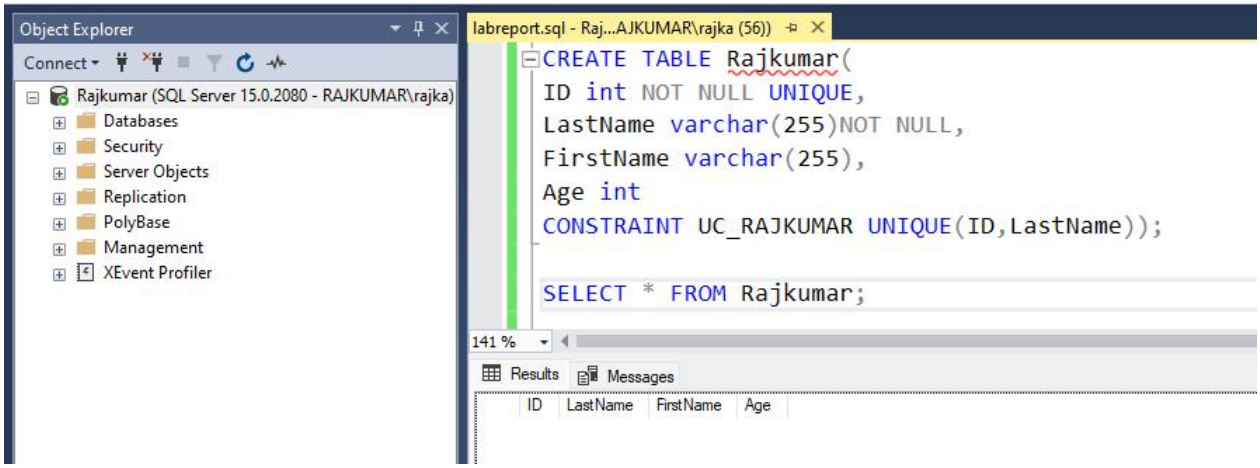
Handwritten Code:

6. Declare constraints in the table and remove one of them.

i. Syntax:(Unique)

CREATE TABLE tb_tablename(
Column 1 datatype NOT NULL UNIQUE,
Coulmn 2 NOT NULL,
Coulmn 3….
CONSTRAINT UC_TABLENAME UNIQUE (column_name,column_name));

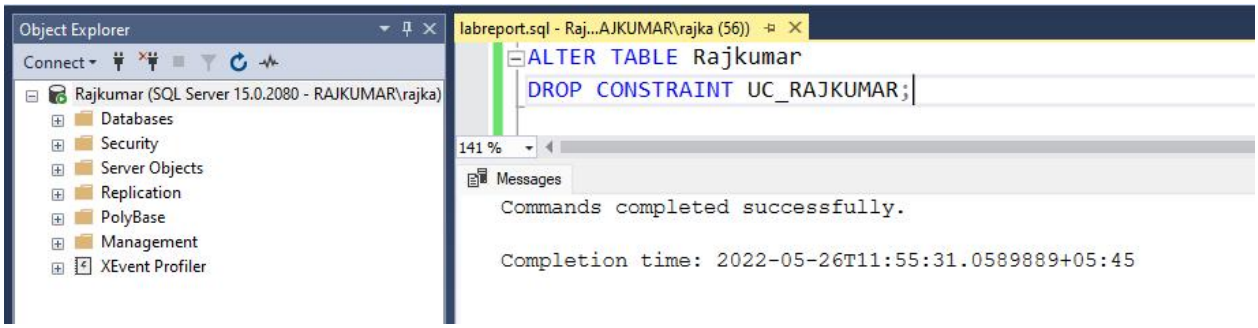
Query with Output:



Syntax: (Droping Unique)

ALTER TABLE table_name
DROP CONSTRAINT UC_TABLENAME;

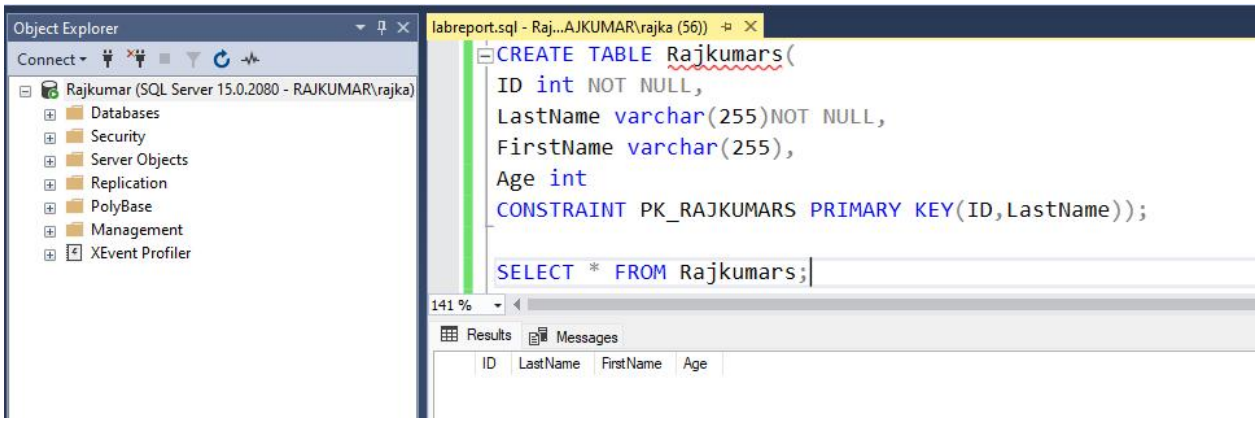
Query with Output:



ii. Syntax:(Primary Key)

CREATE TABLE tb_tablename(
Column 1 datatype NOT NULL UNIQUE,
Coulmn 2 NOT NULL,
Coulmn 3….
CONSTRAINT PK_TABLENAME PRIMARY KEY (column_name,column_name));

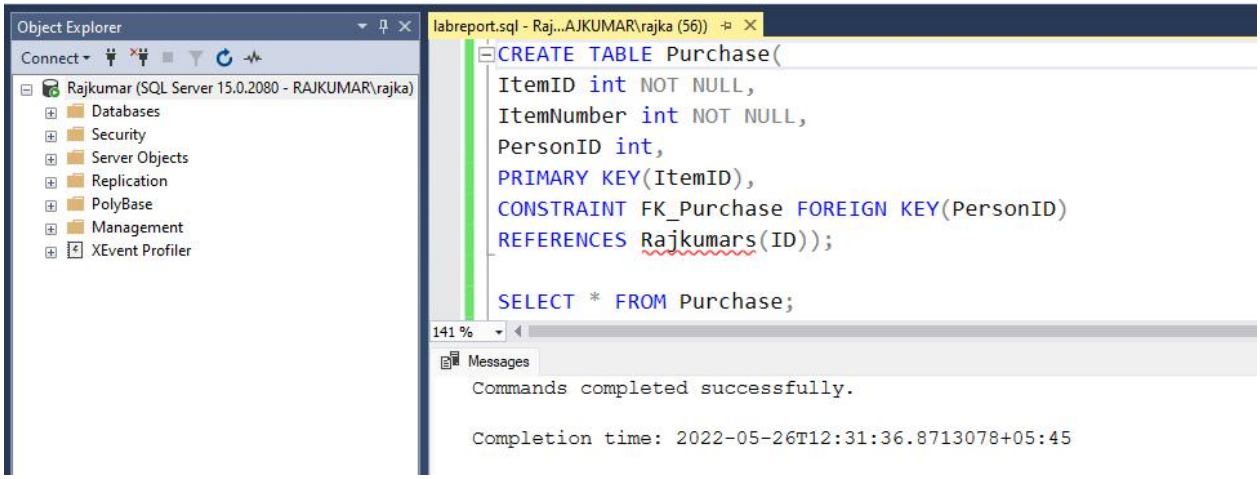
Query with Output:



iii. Syntax:(Function Key)

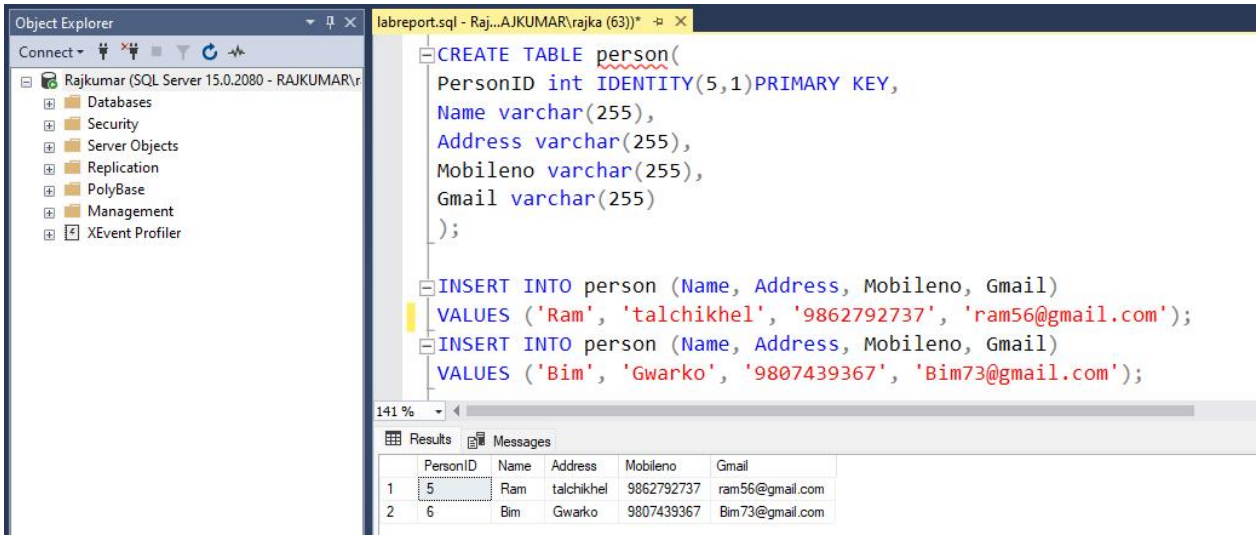
CREATE TABLE tb_tablename(
Column 1 datatype NOT NULL UNIQUE,
Coulmn 2 NOT NULL,
Coulmn 3….
CONSTRAINT FK_TABLENAME FOREIGN KEY (column_name)
REFERENCES tablename(column_name));

Query with Output:



Handwritten Code:

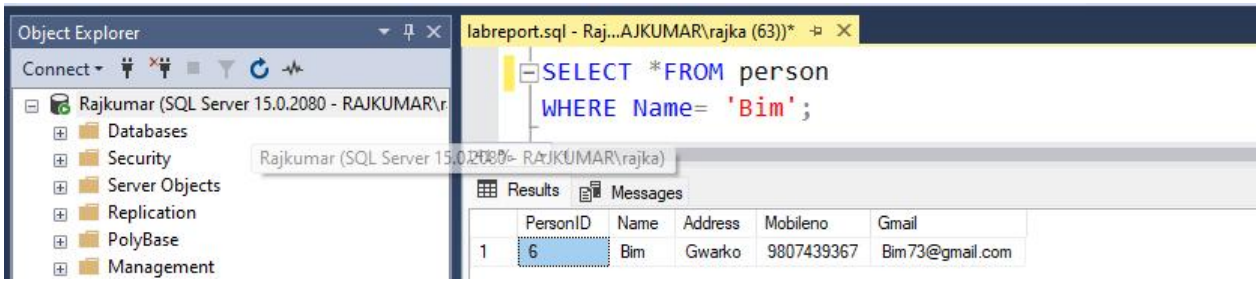
7. Use the WHERE clause in the query.



Syntax:

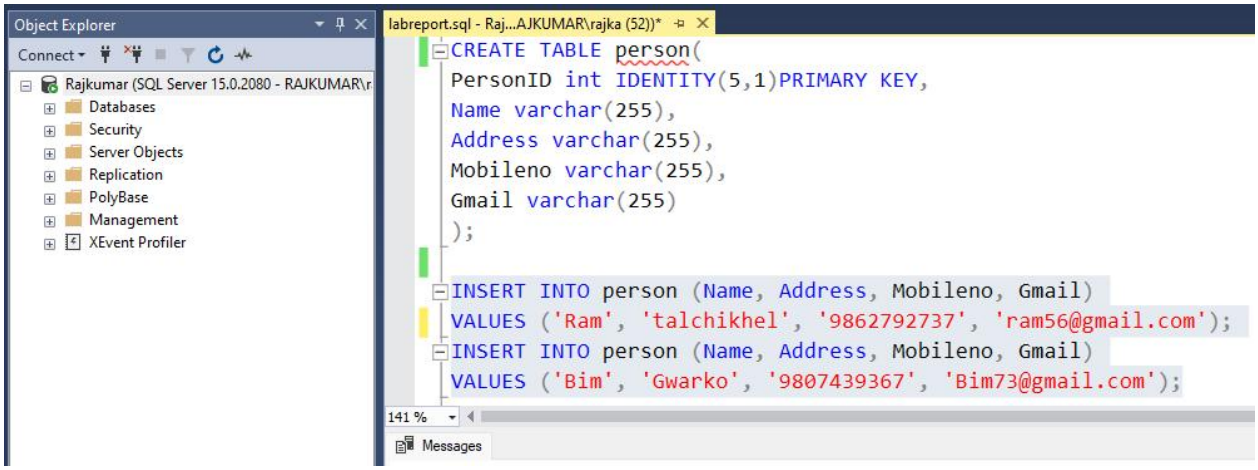
SELECT column1, column2,….
FROM table_name
WHERE condition;

Query with Output:



Handwritten Code:

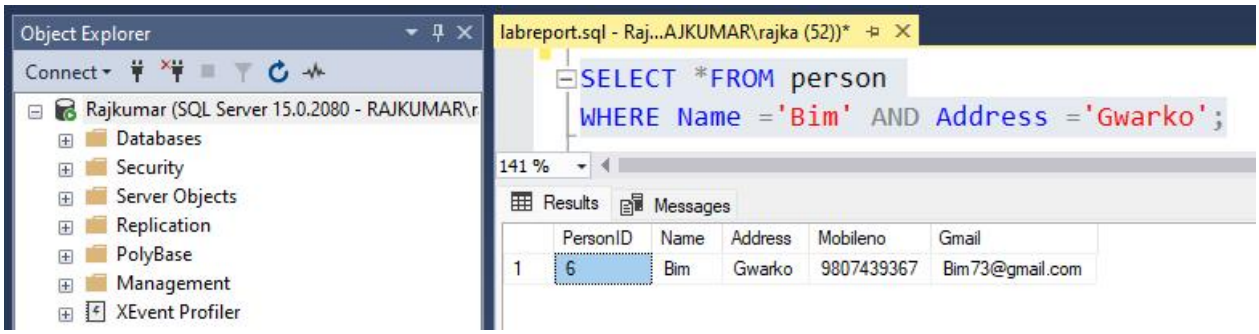
8. Use AND, OR, NOT queries



i. Syntax: (AND)

SELECT column1, column2, …
FROM table_name
WHERE condition1 AND condition2 AND condition3….;

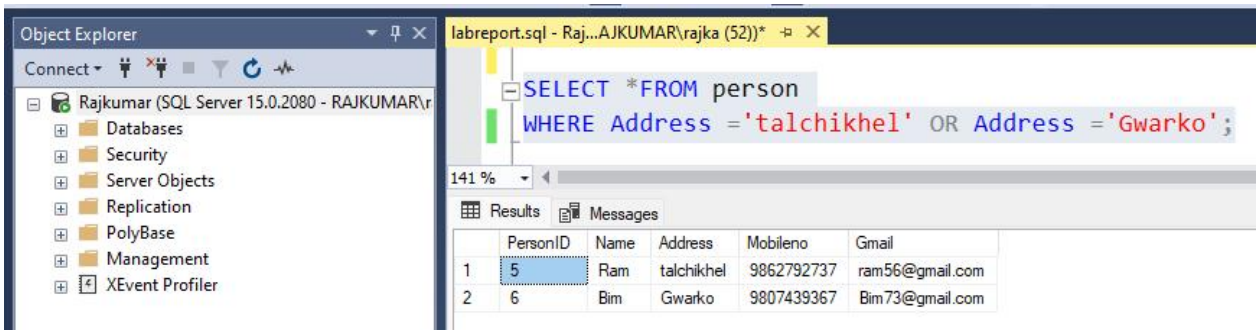
Query with Output:



ii. Syntax: (OR)

SELECT column1, column2, …
FROM table_name
WHERE condition1 OR condition 2 or condition3….;

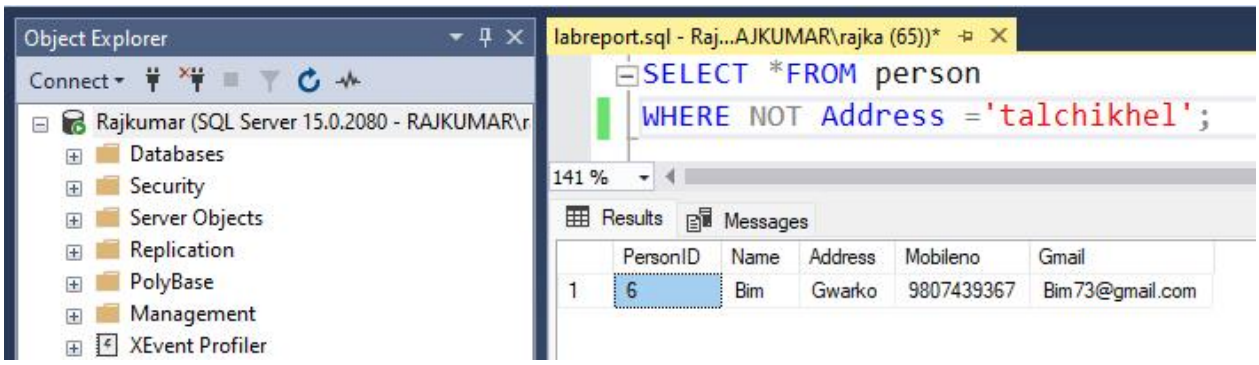
Query with Output:



iii. Syntax: (NOT)

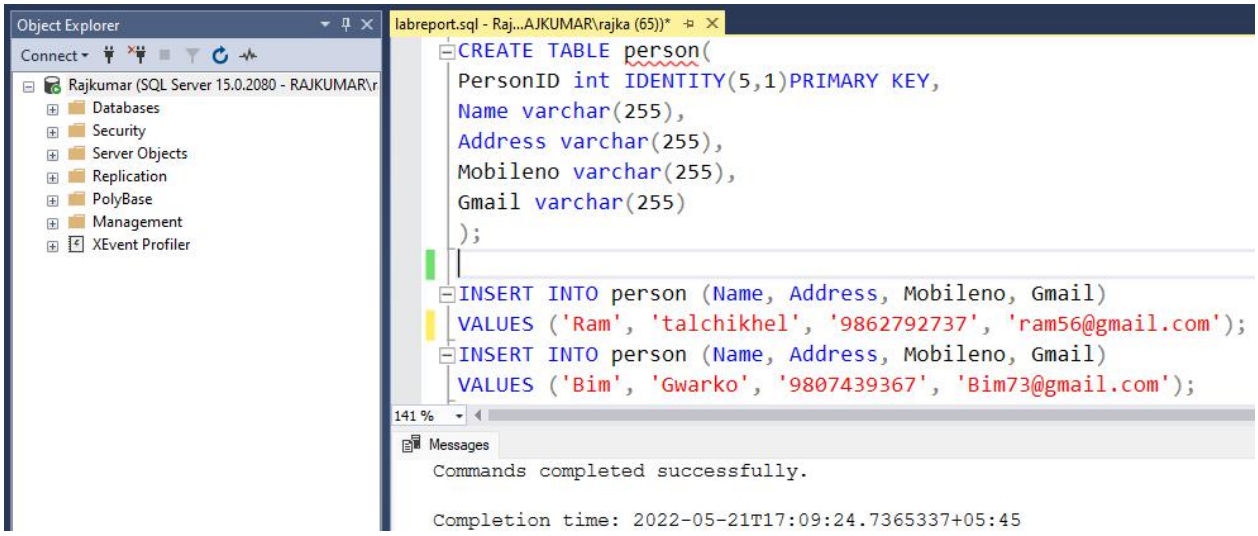
SELECT column1, column2, …
FROM table_name
WHERE NOT condition;

Query with Output:



Handwritten Code:

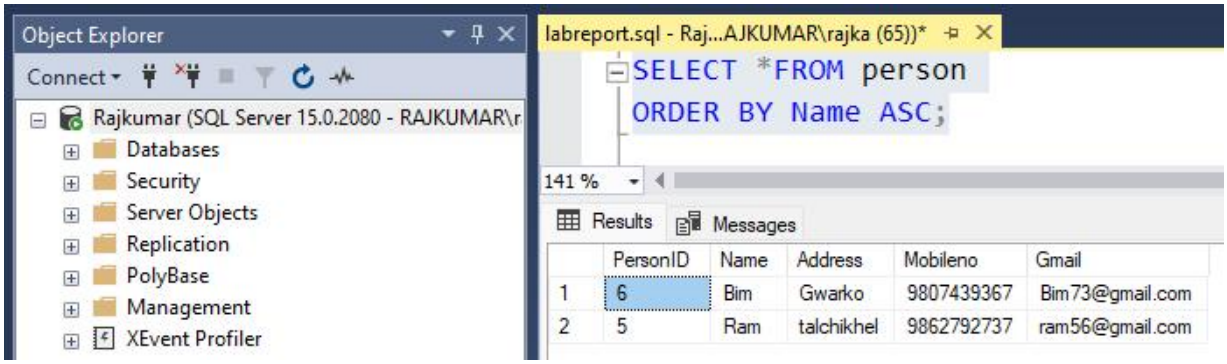
9. Use “ ORDER BY ” query for both ascending and descending orders.



i. Syntax: (Ascending)

SELECT column1, column2,…. FROM table_name
ORDER BY column1, column2,….ASC

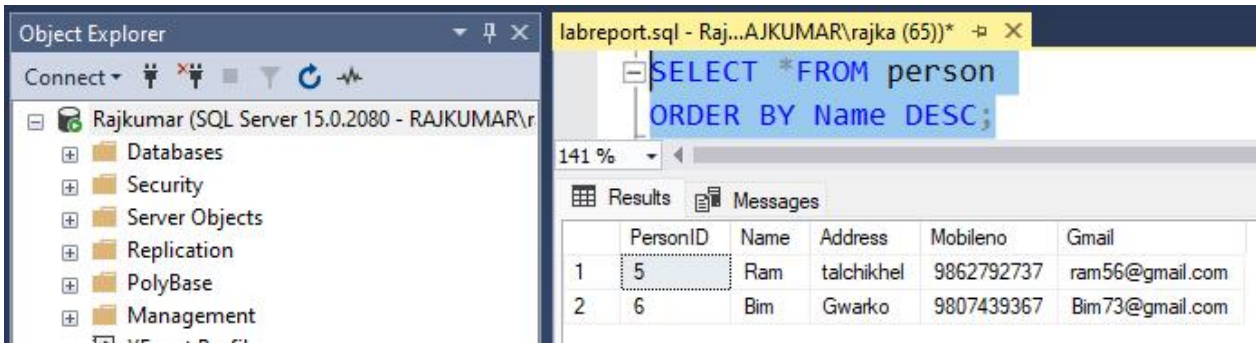
Query with Output:



ii. Syntax: (Descending)

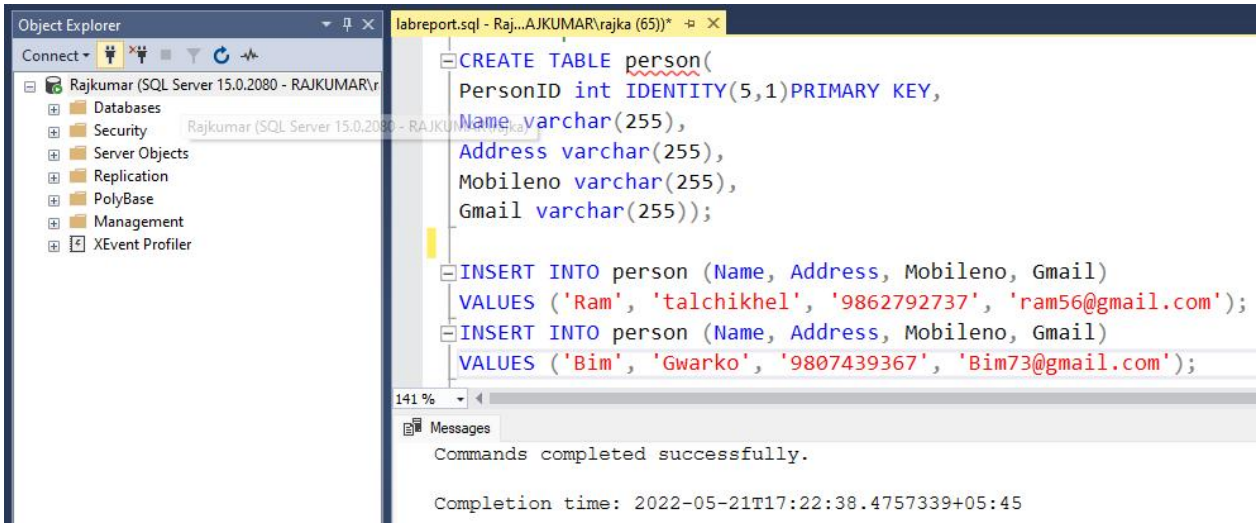
SELECT column1, column2,…. FROM table_name
ORDER BY column1, column2,….DESC;

Query with Output:



Handwritten Code:

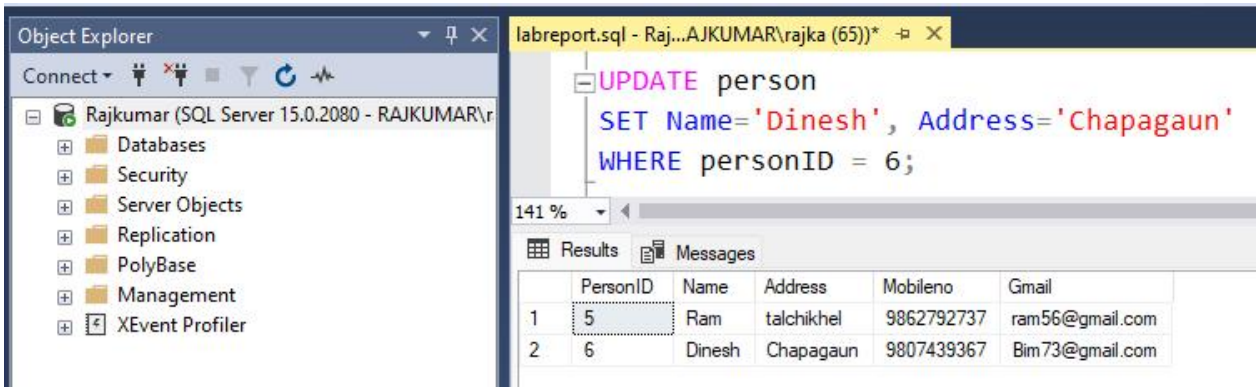
10. Update and delete table fields.



i. Syntax: (UPDATE)

UPDATE table_name
SET column1= value1, column2=value2,…..
WHERE condition;

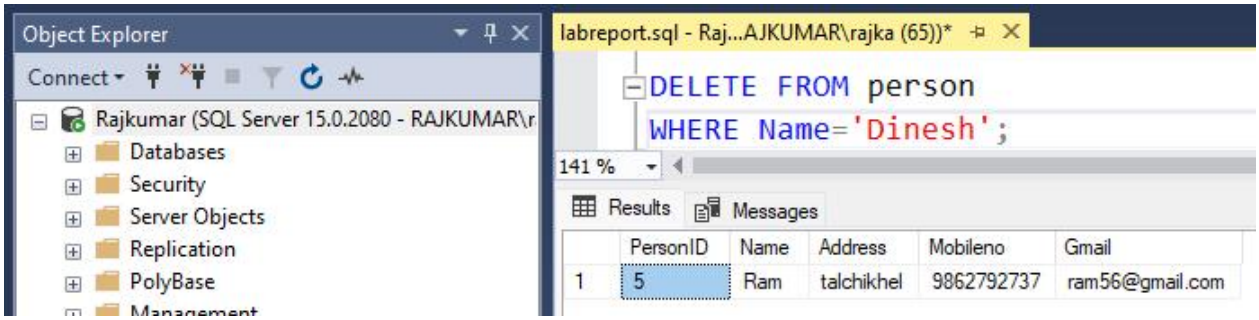
Query with Output:



ii. Syntax: (DELETE)

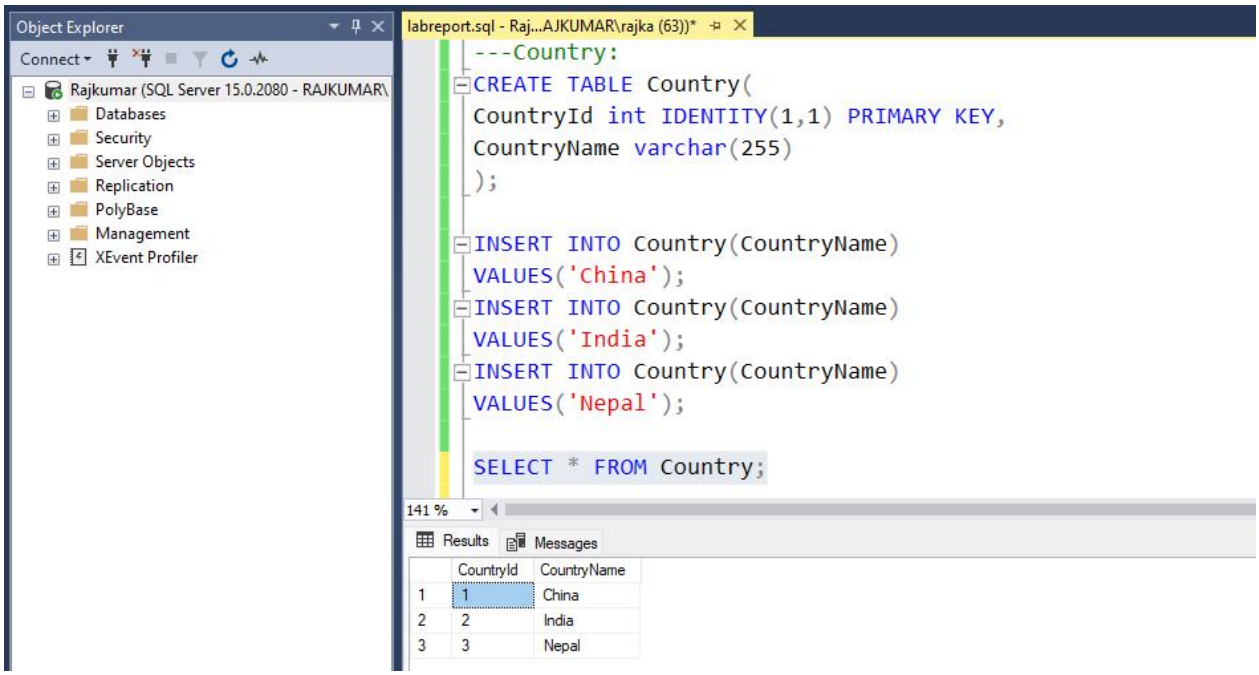
DELETE FROM table_name
WHERE condition;

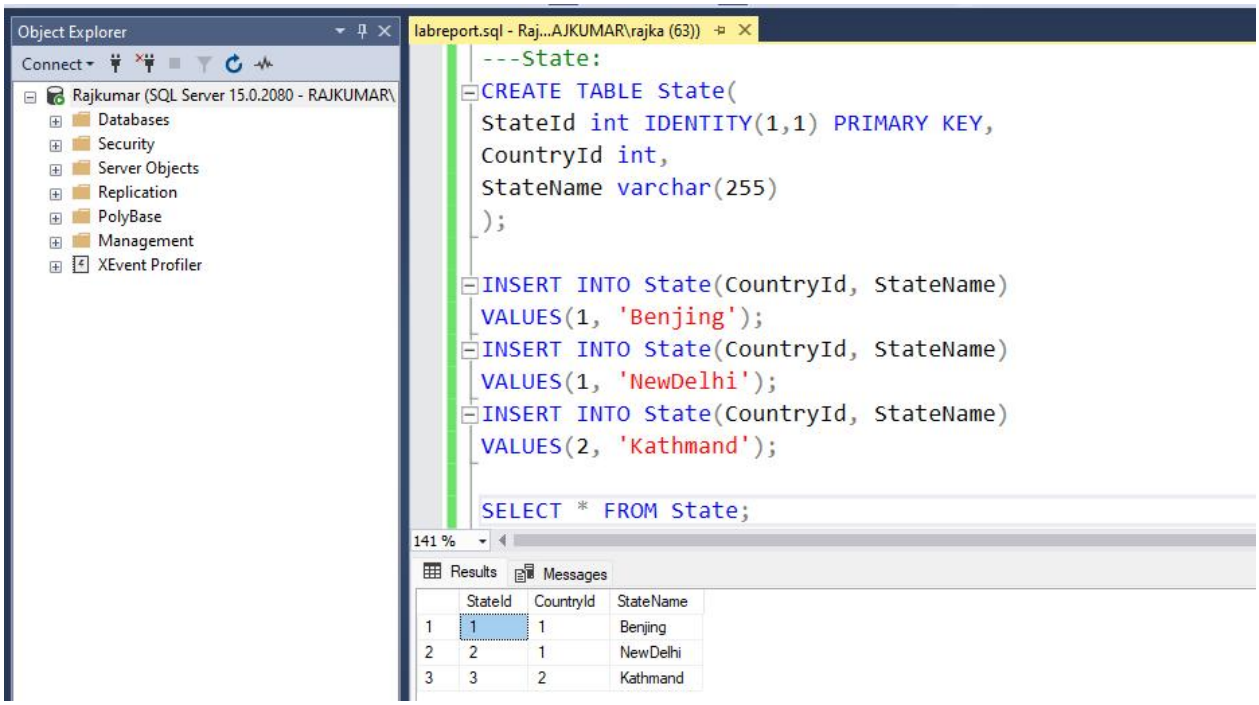
Query with Output:

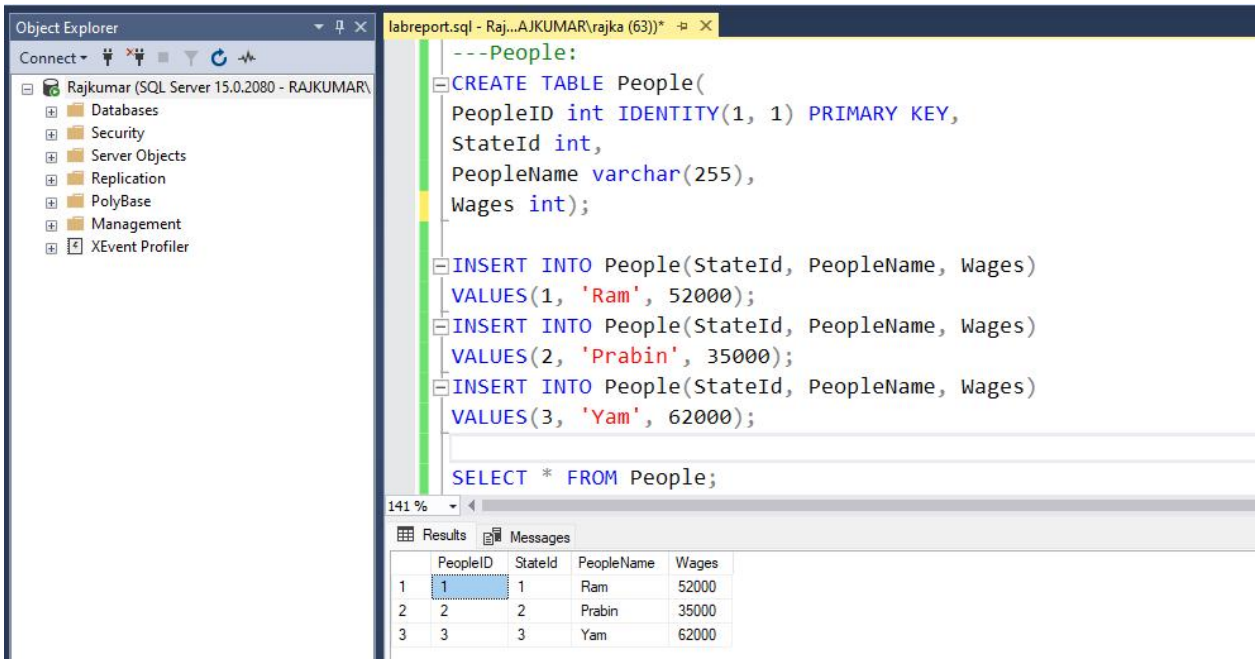


Handwritten Code:

11. Use Join queries



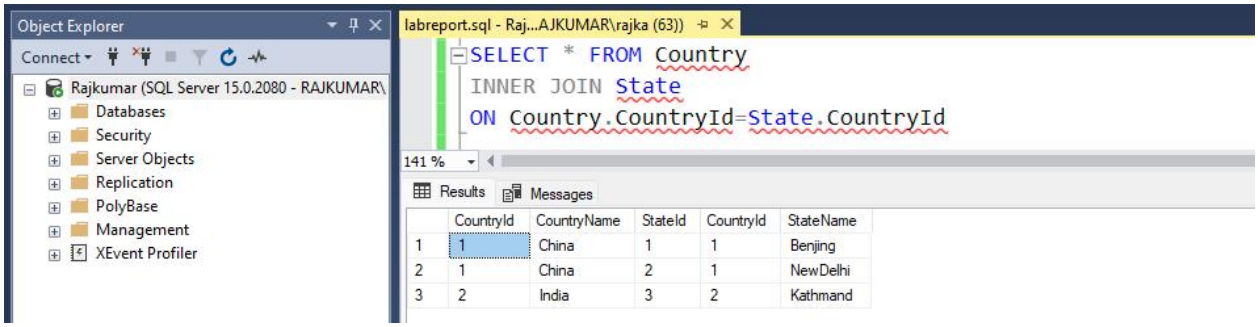




i. Syntax: (INNER JOIN)

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

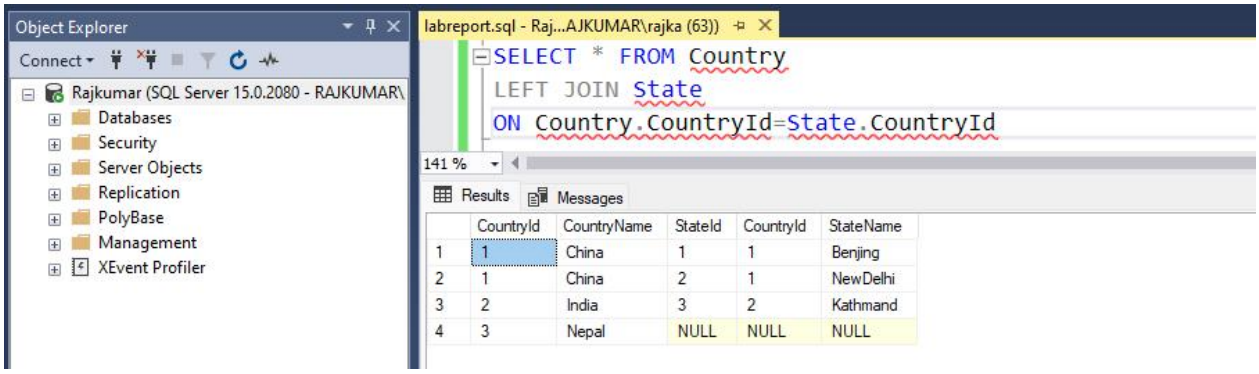
Query with Output:



ii. Syntax: (LEFT JOIN)

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

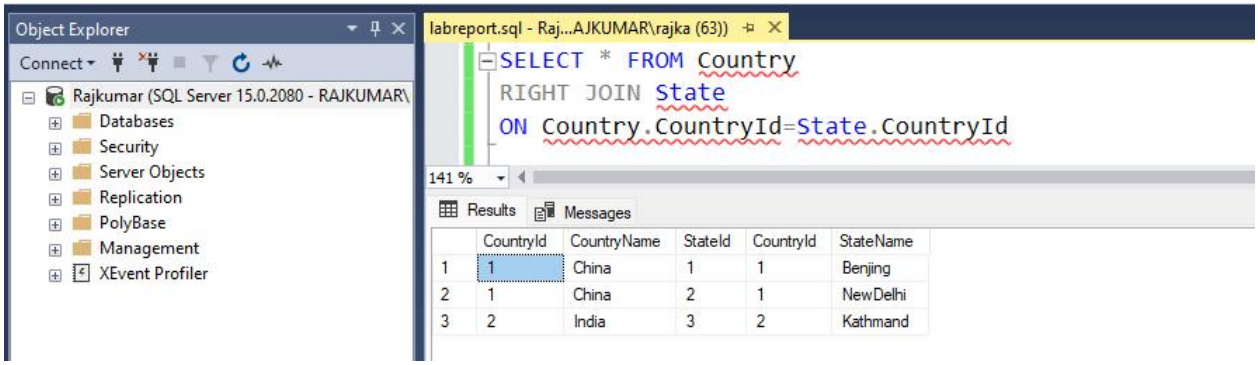
Query with Output:



iii. Syntax: (RIGHT JOIN)

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

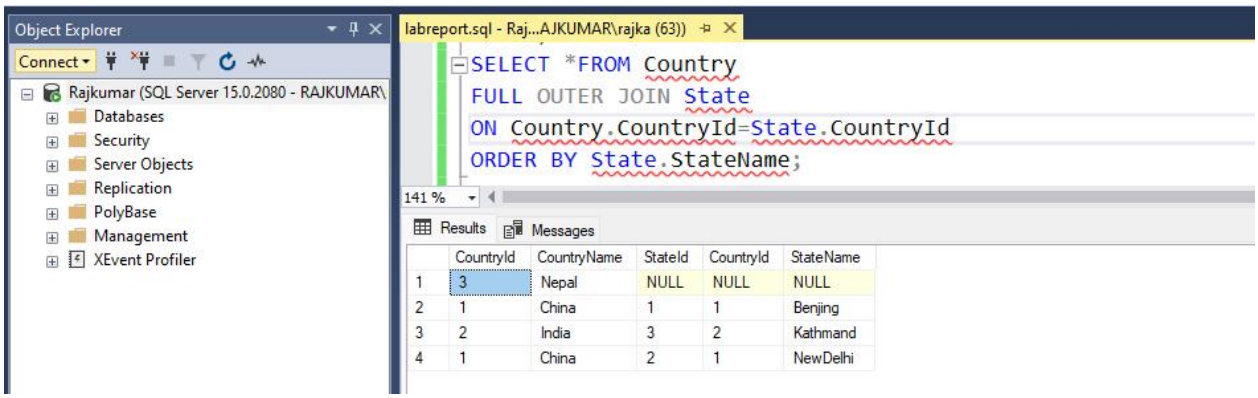
Query with Output:



iv. Syntax: (FULL OUTER JOIN)

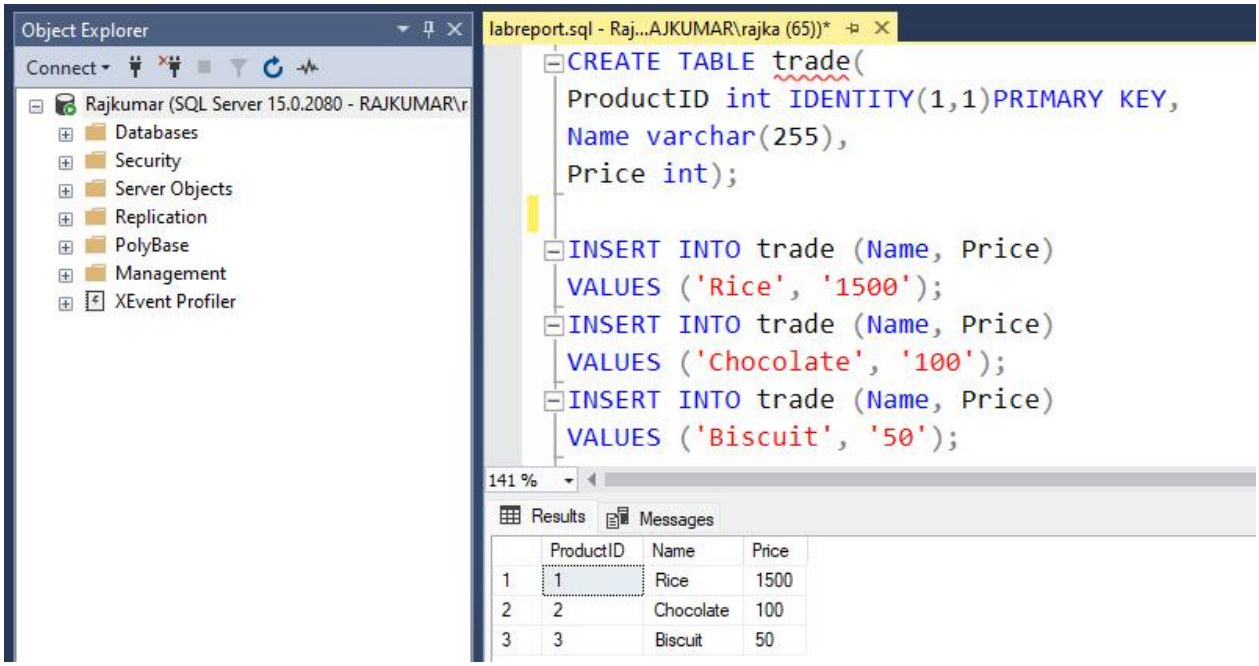
SELECT column_name(s) FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name;
WHERE condition;

Query with Output:



Handwritten Code:

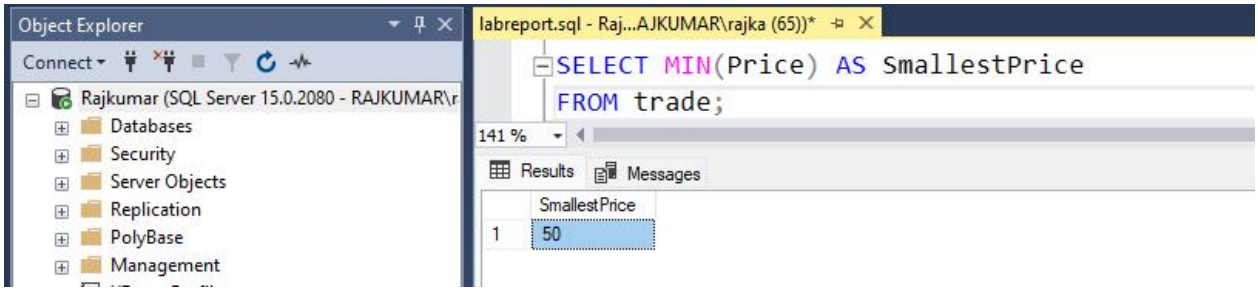
12. Use the MIN and MAX functions.



i. Syntax: (MIN Function)

SELECT MIN(column_name)
FROM table_name
WHERE condition;

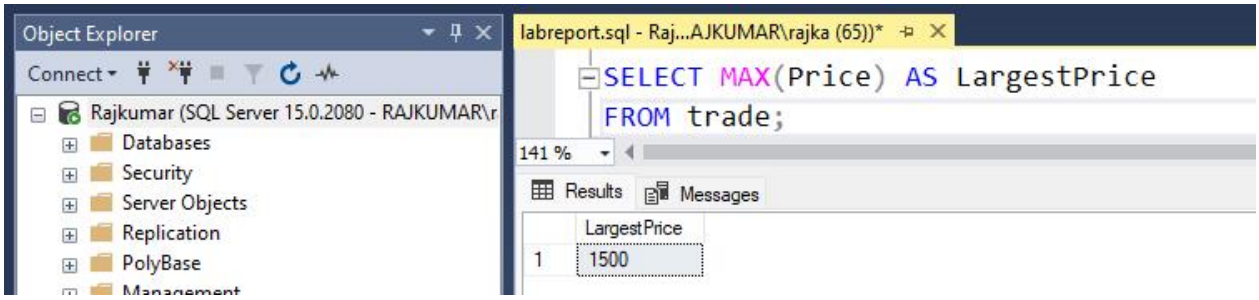
Query with Output:



ii. Syntax: (MAX Function)

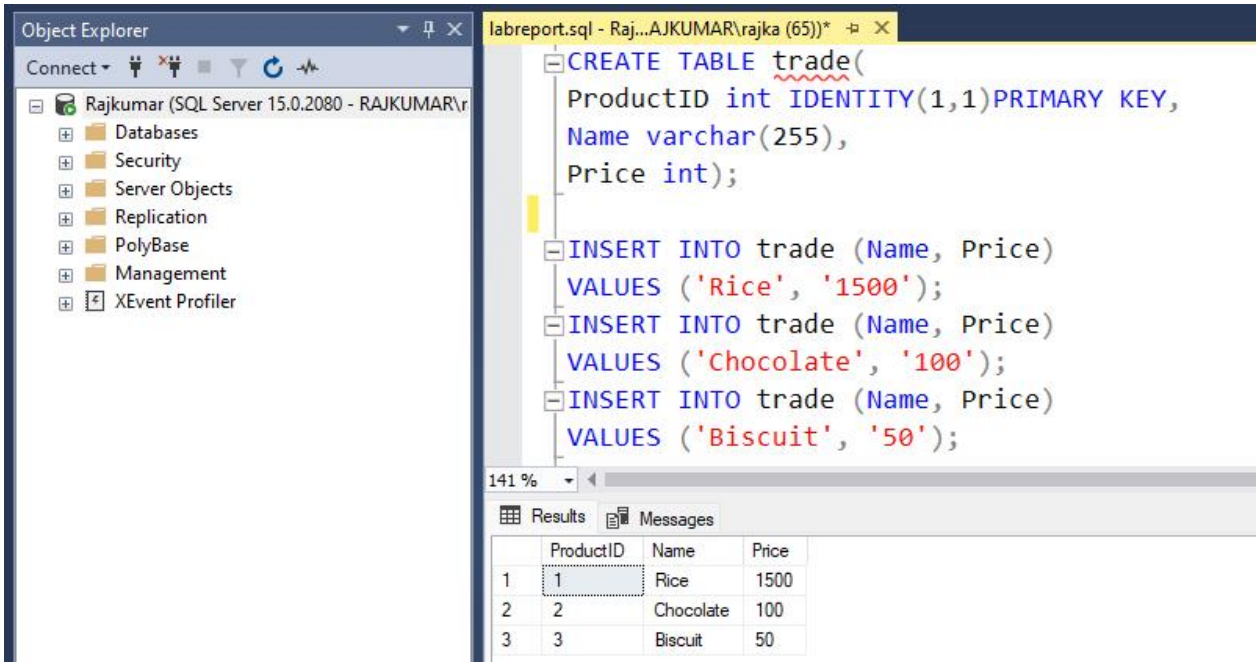
SELECT MAX(column_name)
FROM table_name
WHERE condition;

Query with Output:



Handwritten Code:

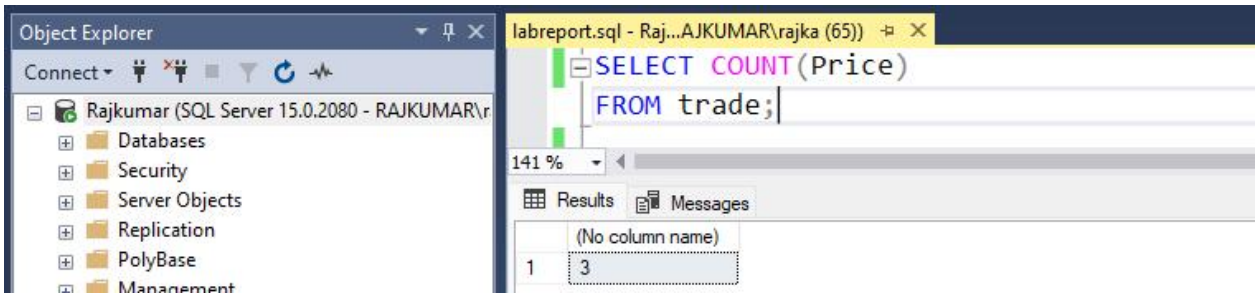
13. Use the COUNT, AVERAGE, and SUM functions.



i. Syntax: (COUNT)

SELECT COUNT (column_name)
FROM table_name
WHERE condition;

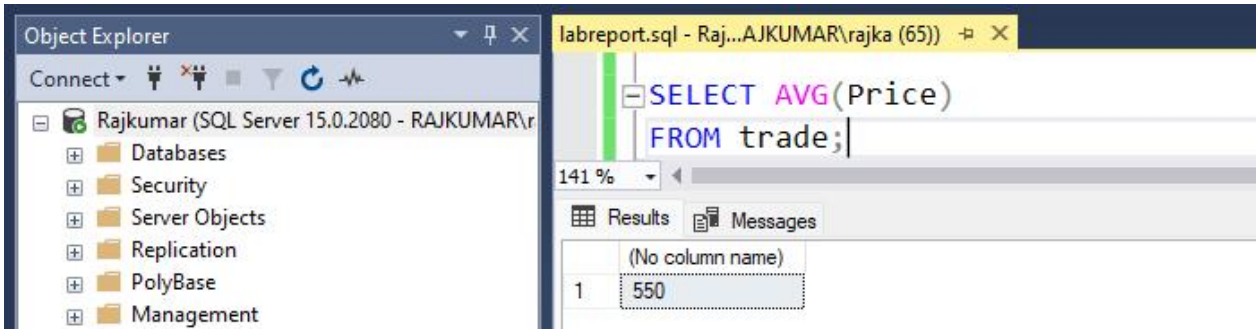
Query with Output:



ii. Syntax: (AVERAGE)

SELECT AVG(column_name)
FROM table_name
WHERE condition;

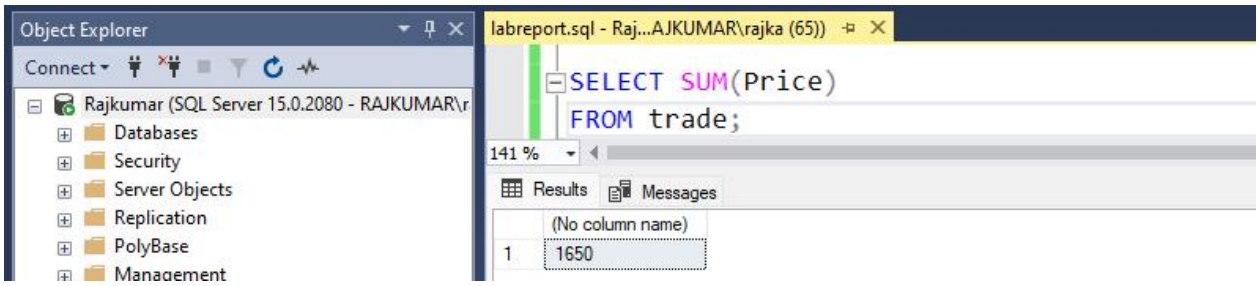
Query with Output:



iii. Syntax: (SUM)

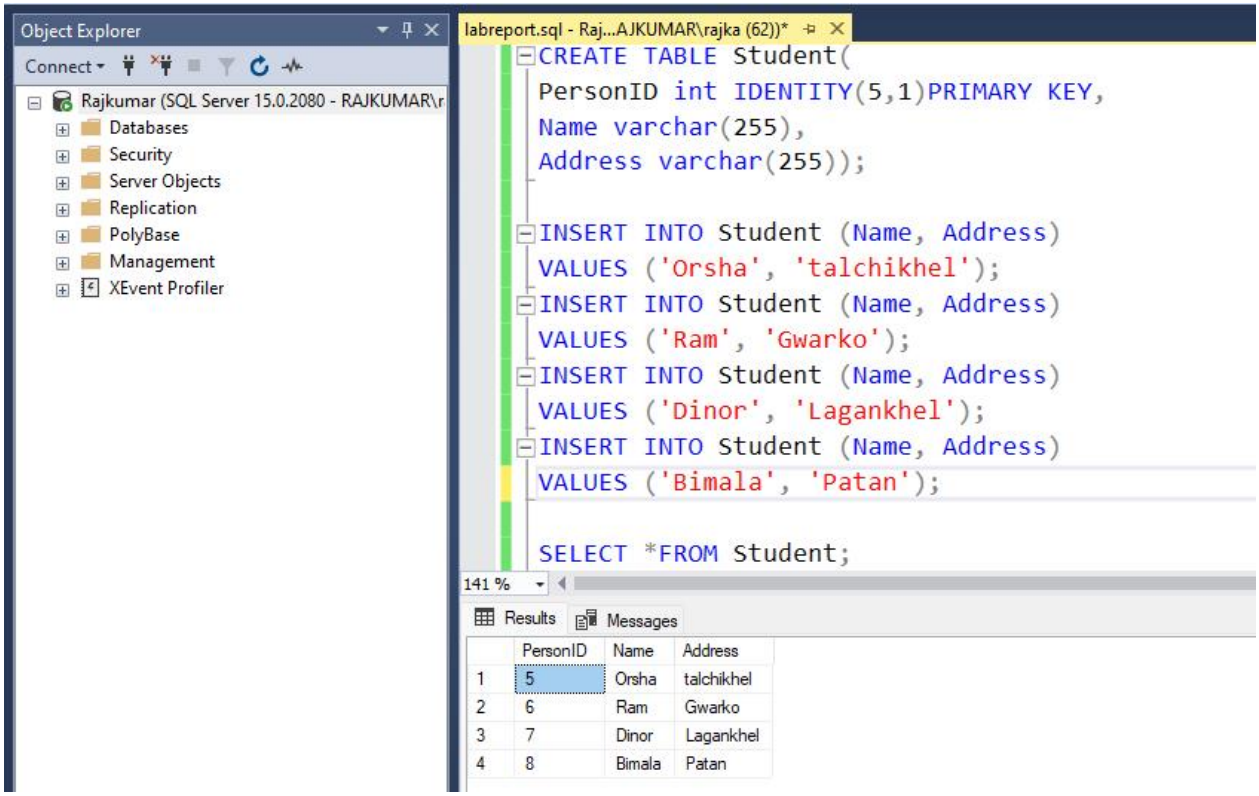
SELECT SUM (column_name)
FROM table_name
WHERE condition;

Query with Output:



Handwritten Code:

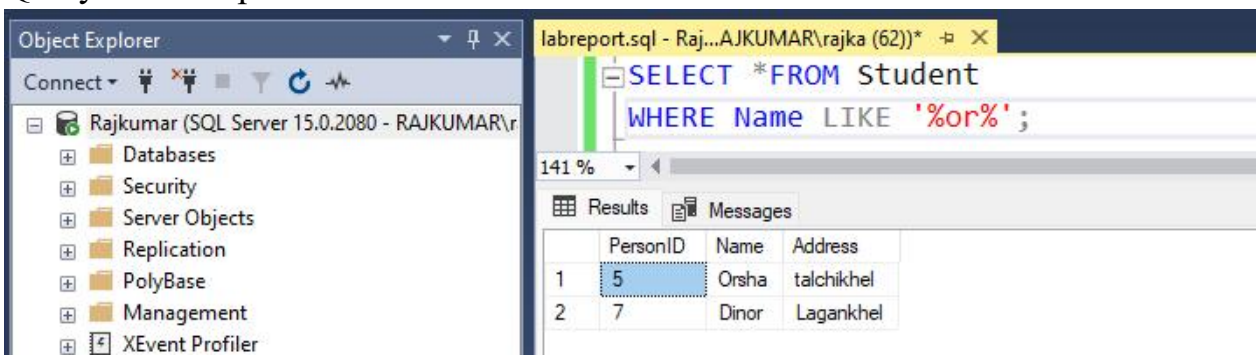
14. Display students' names with "OR" in any position.



Syntax:

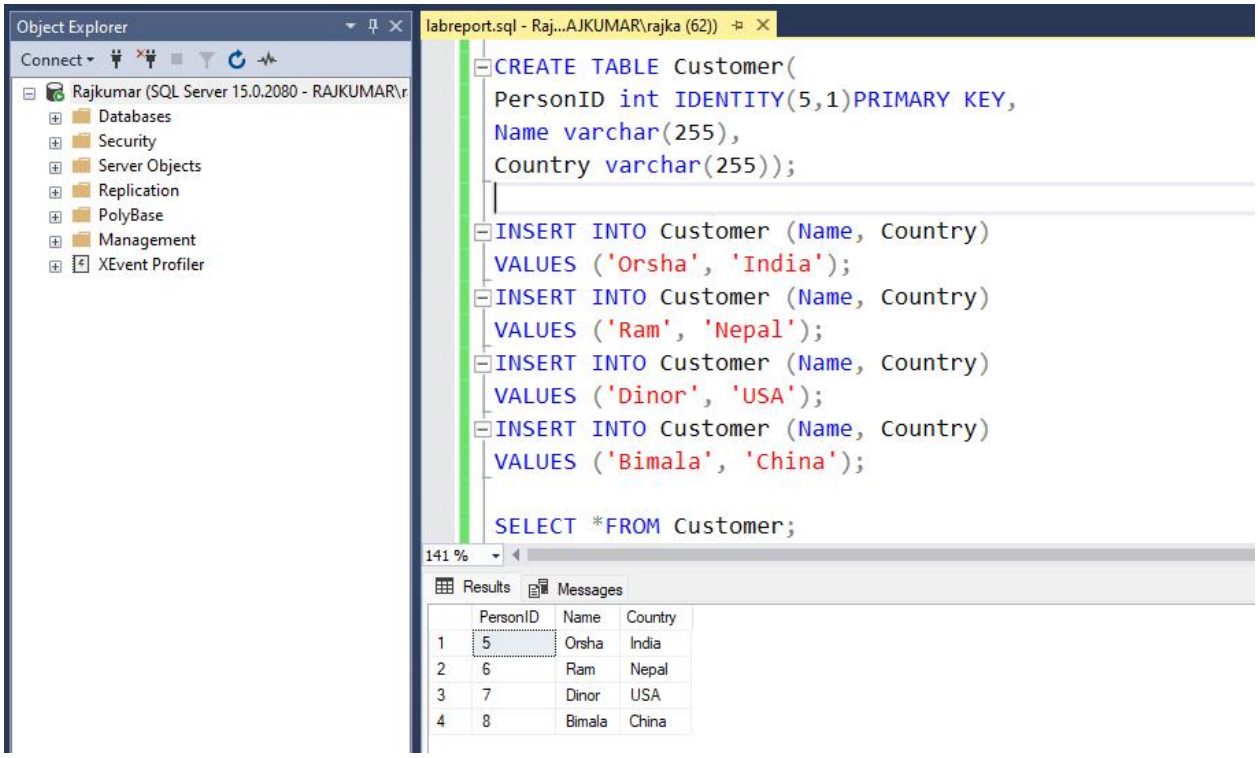
SELECT * FROM table_name
WHERE column_name LIKE '%a%';

Query with Output:



Handwritten Code:

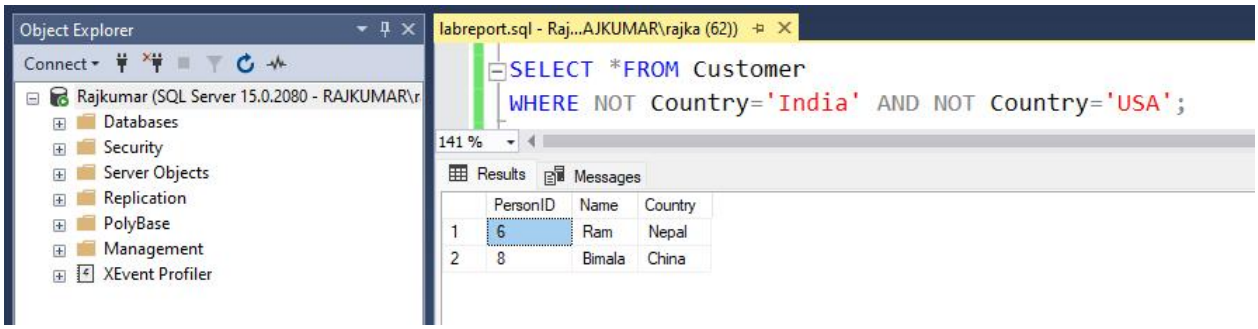
15. Display all customers that are not located in India and the USA.



Syntax:

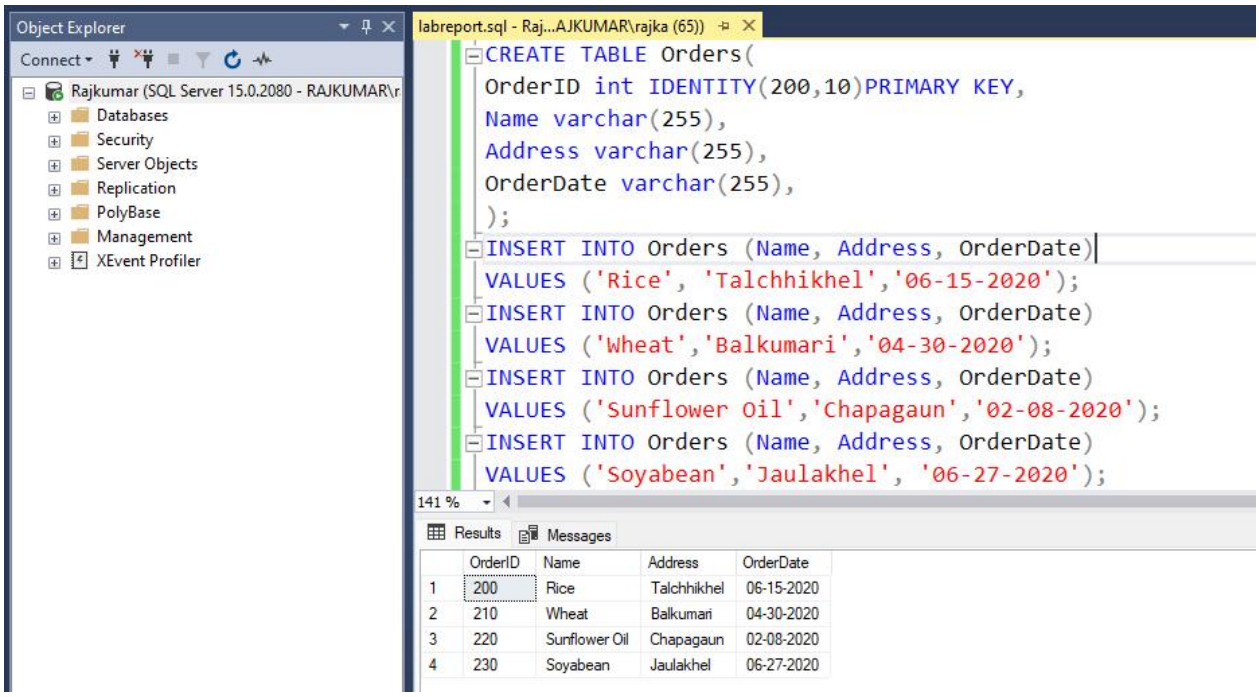
SELECT column1, column2, ...
FROM table_name
WHERE NOT condition ;

Query with Output:



Handwritten Code:

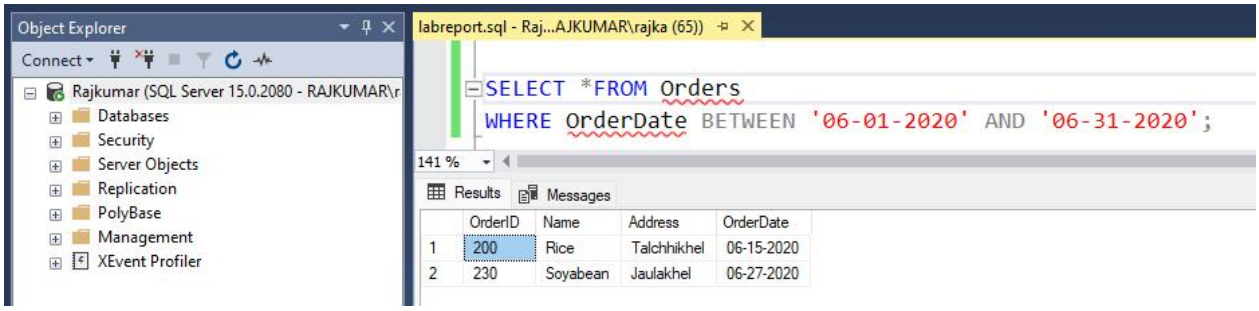
16. Display all orders with an OrderDate BETWEEN '06-01-2020' AND '06-31-2020'



Syntax:

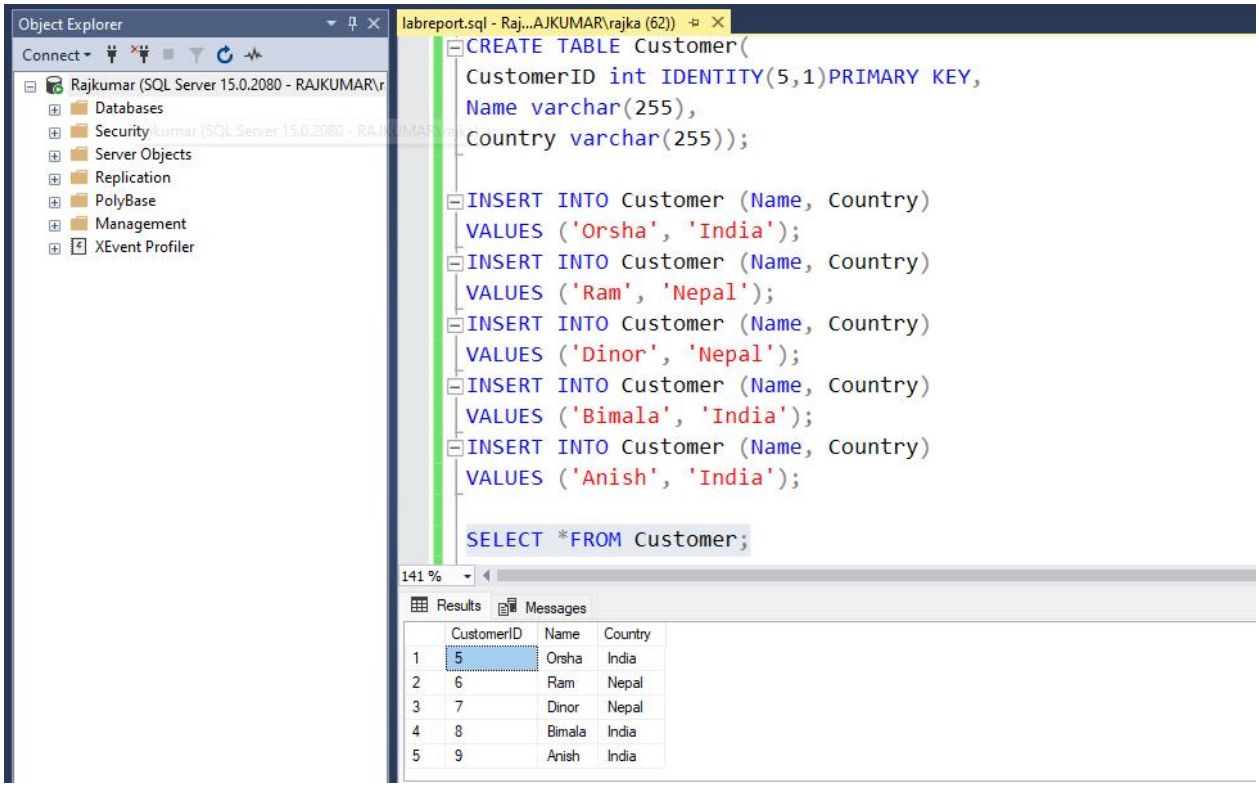
SELECT column_name(s)
FROM table_name
WHERE column_name BETWEEN value1 AND value2;

Query with Output:



Handwritten Code:

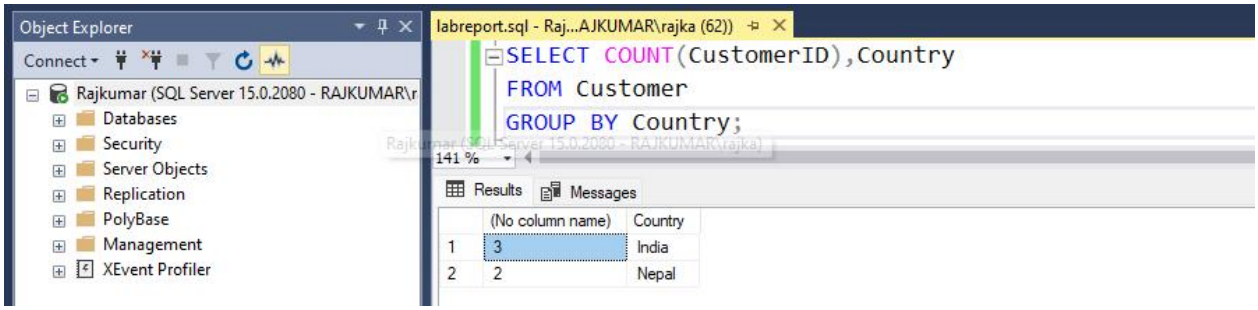
17. Count the number of customers in each country.



Syntax:

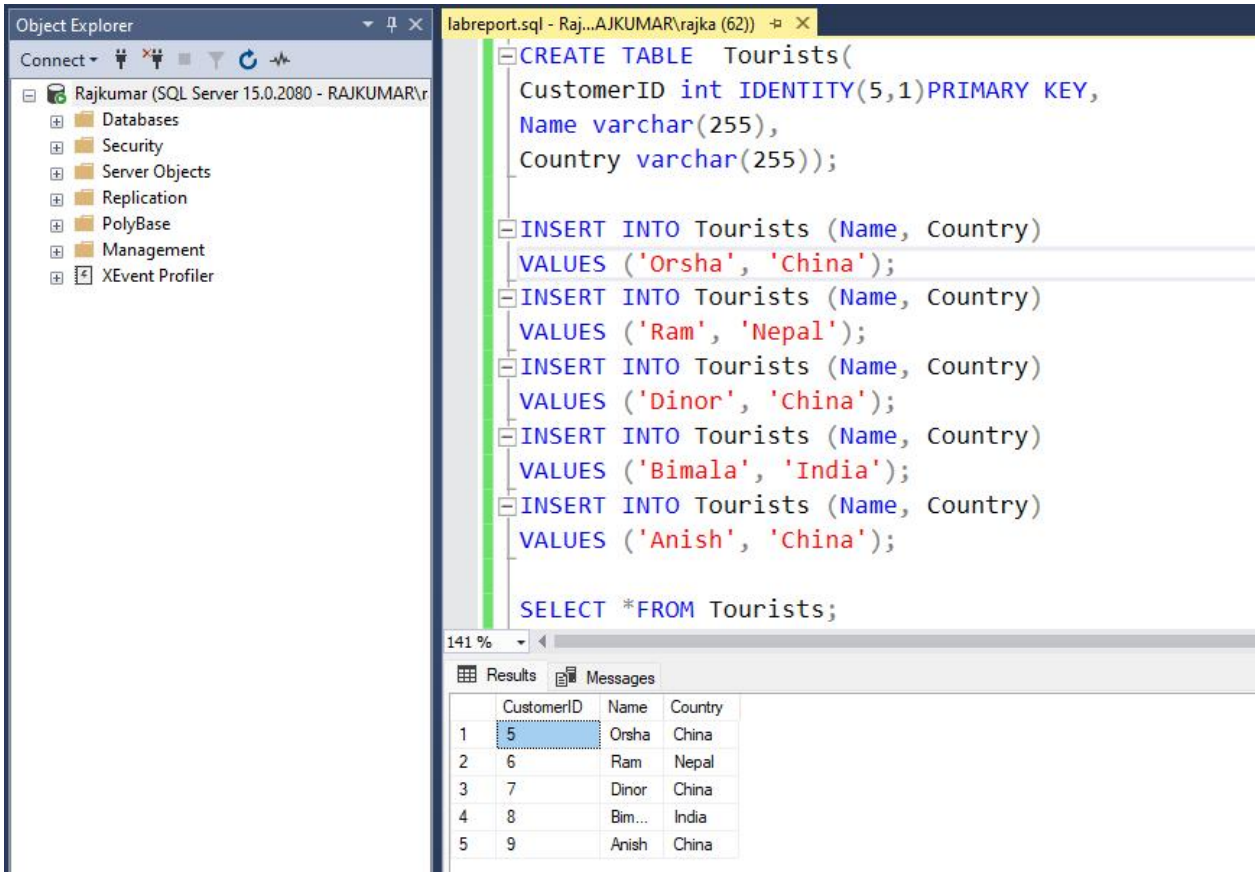
SELECT COUNT (column_name)
FROM table_name
GROUP BY column_name;

Query with Output:



Handwritten Code:

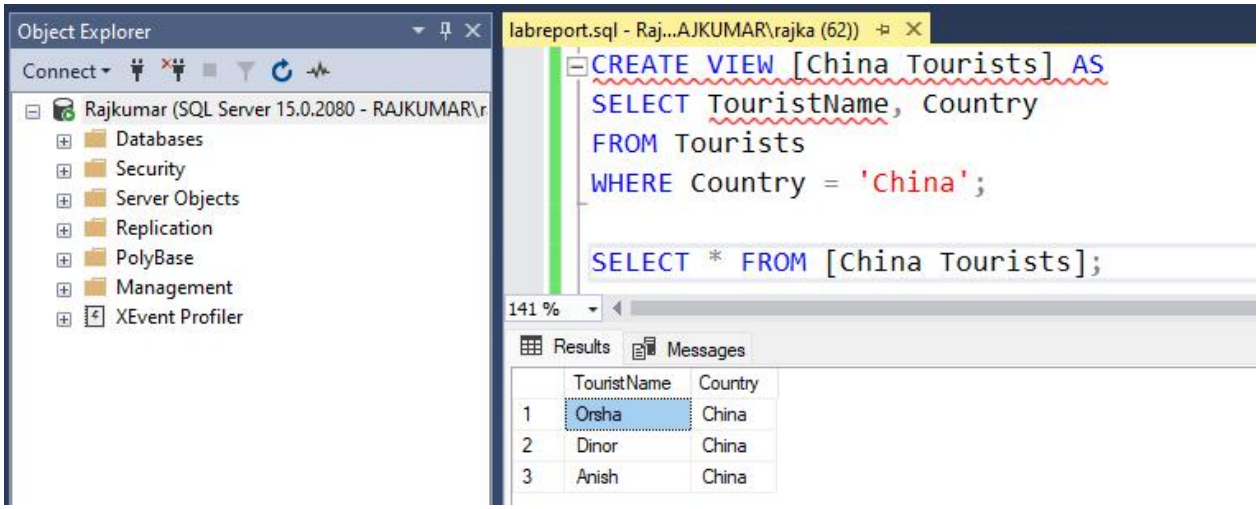
18. Create a view to display all tourists from China.



Syntax:

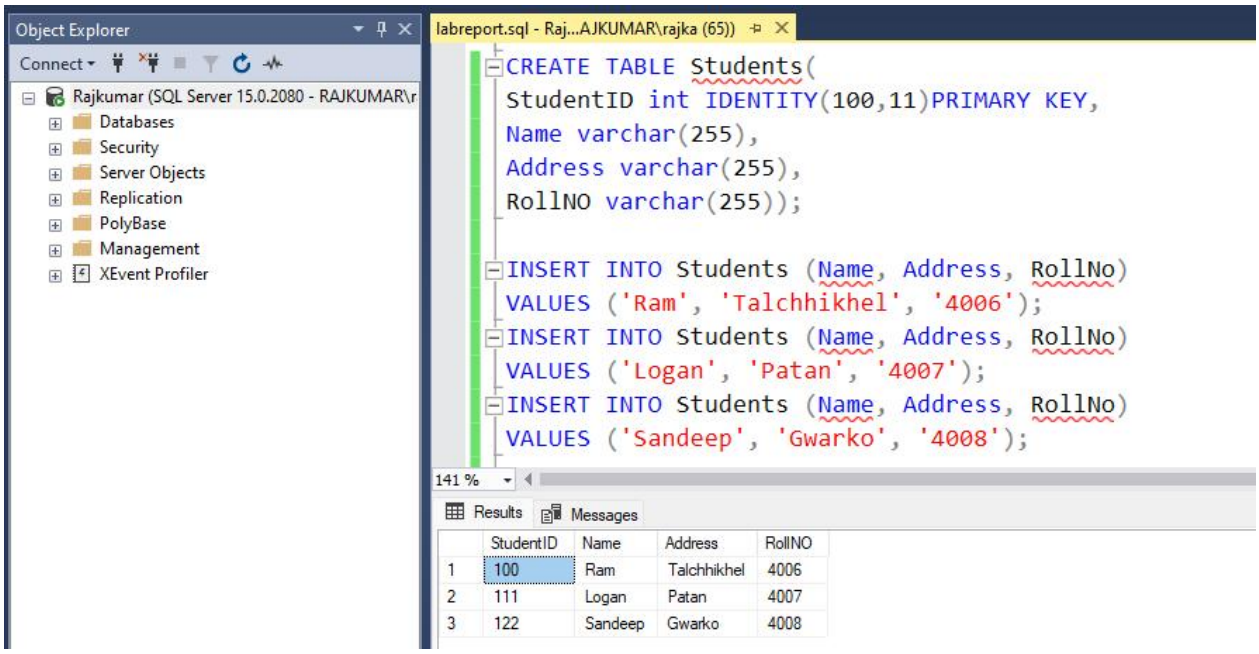
CREATE VIEW view_name AS
SELECT column1, column2, …..
FROM table_name
WHERE condition;

Query with Output:



Handwritten Code:

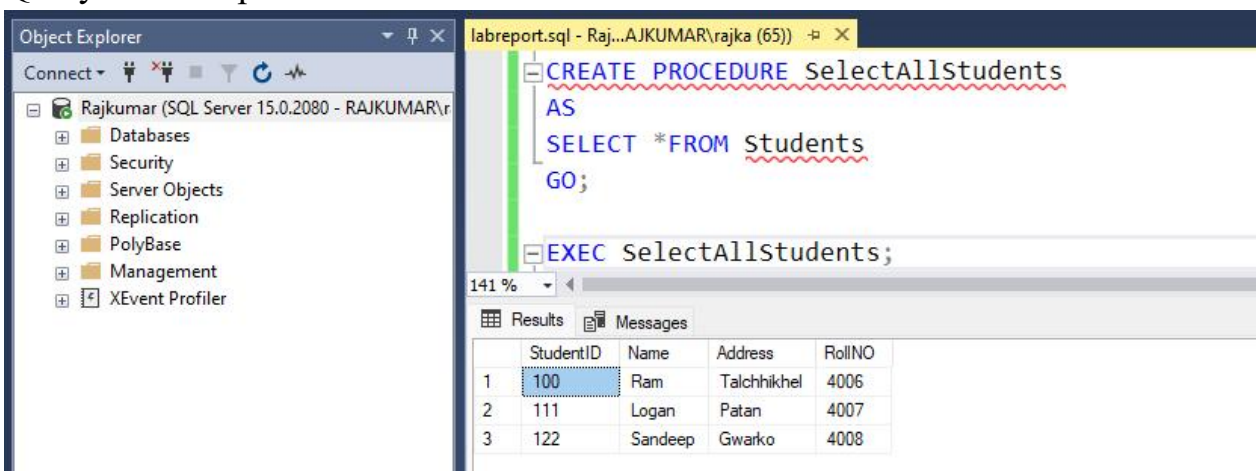
19. Create a stored procedure named "SelectAllStudents" that selects all records from the "Students" table.



Syntax:

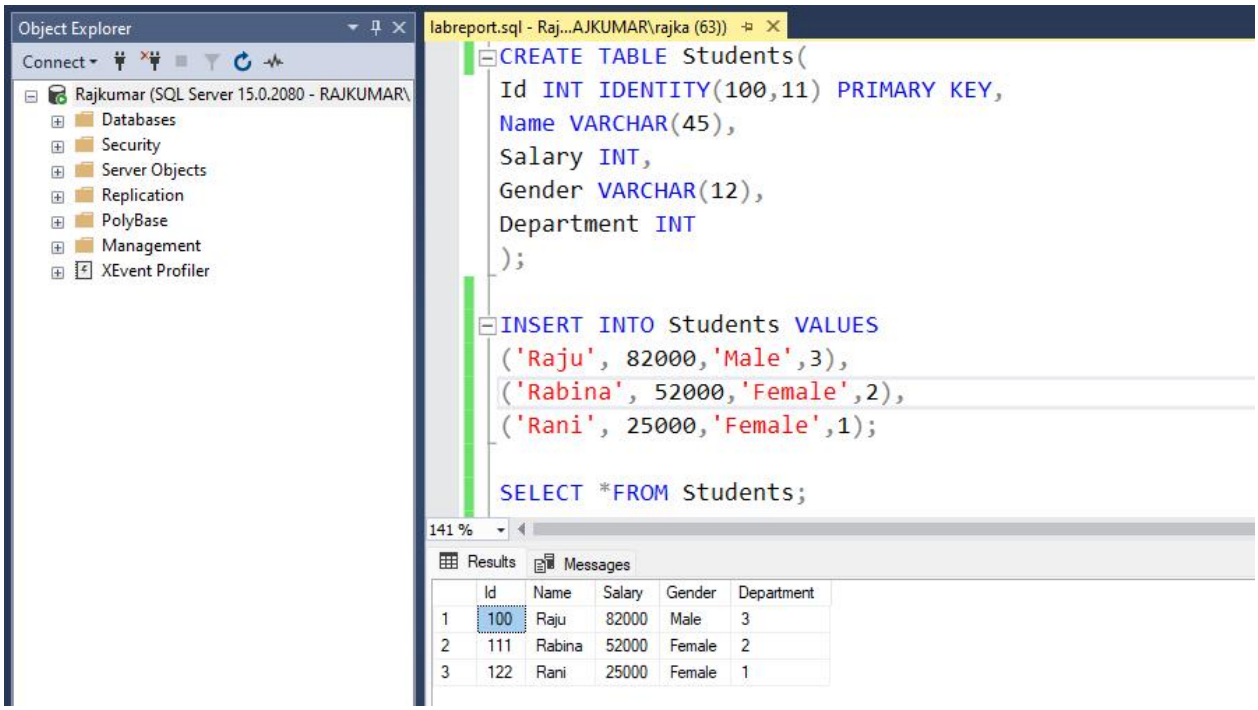
CREATE PROCEDURE procedure_name
AS
sql_statement
GO;

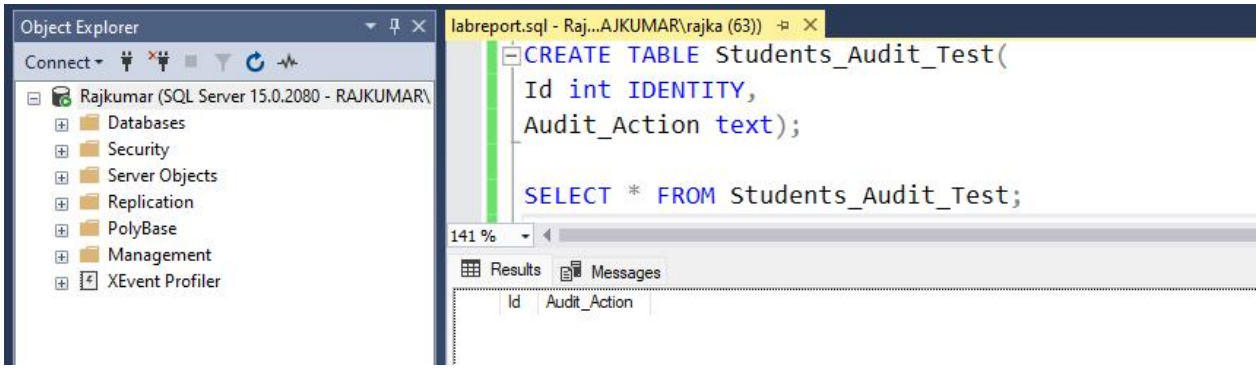
Query with Output:



Handwritten Code:

20. Create a trigger that stores transaction records of each inserted operation on the Students table.

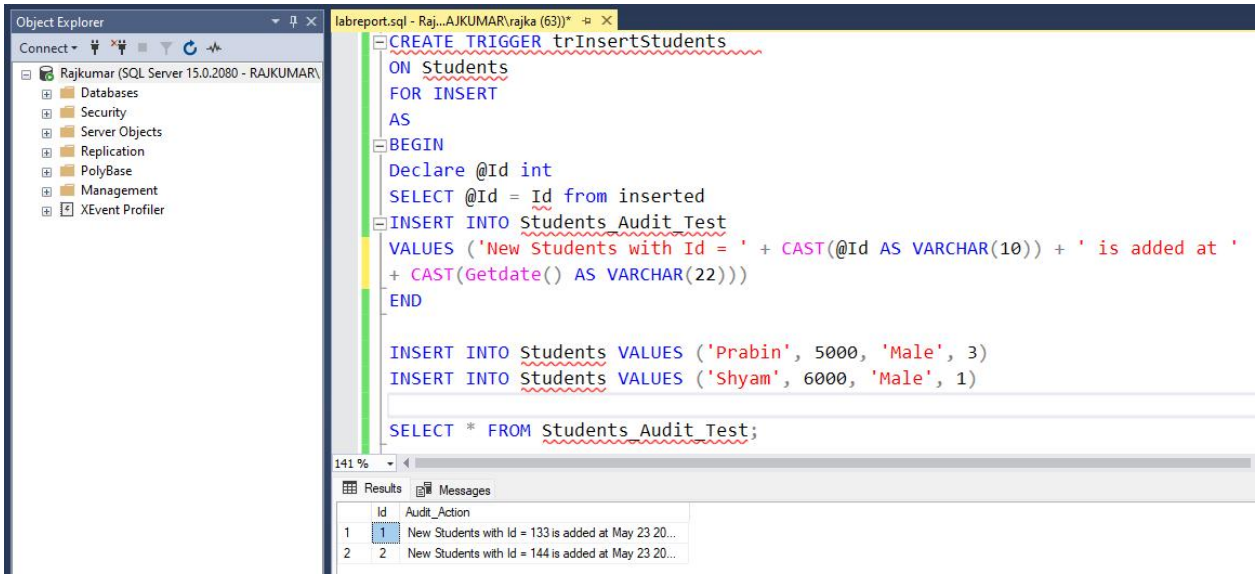




Syntax:

CREATE TRIGGER schema.trigger_name
ON table_name
AFTER {INSERT}
{NOT FOR REPLICATION}
AS
{SQL Statements}

Query with Output:



Handwritten Code: