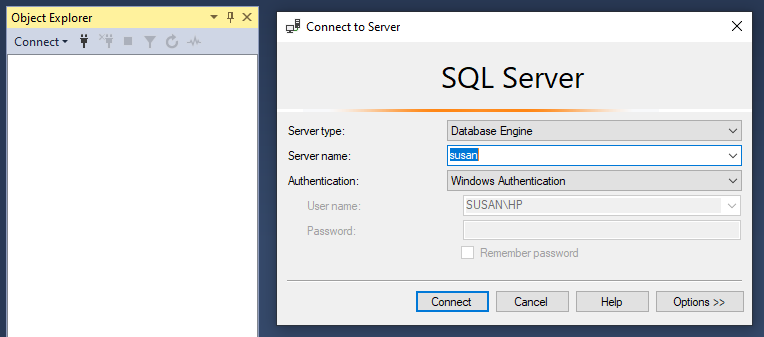
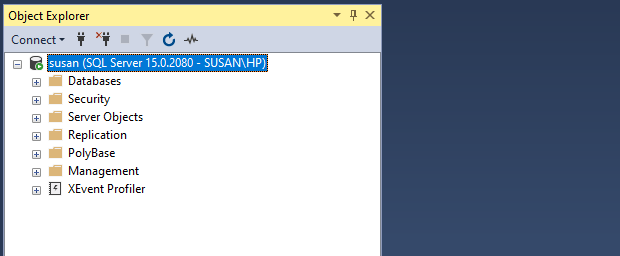
Connecting to server



Connected to server

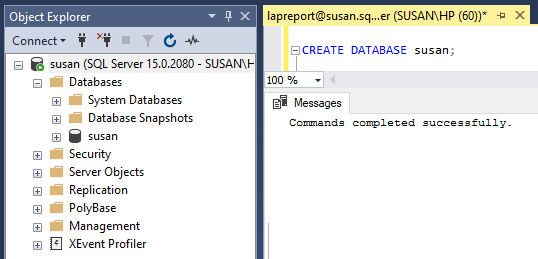


1. Creating and removing database queries

Syntax:

CREATE DATABASE databasename;

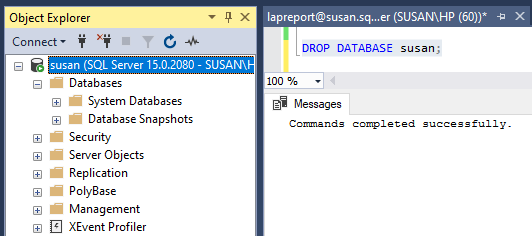
Query with Output:



Syntax:

DROP DATABASE databasename;

Query with Output:



1. Creating and Removing Table Queries

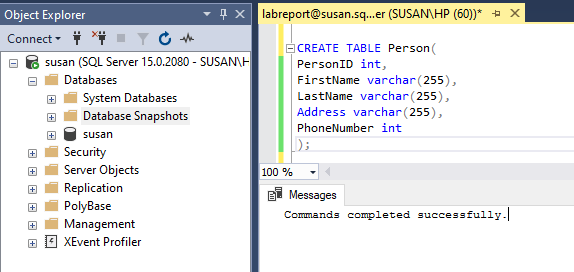
Syntax:

CREATE TABLE tbl\_name(

column datatype,

column2 datatype..);

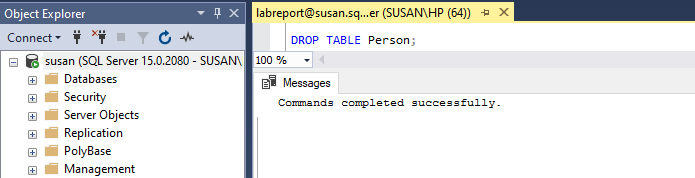
Query with Output:



Syntax:

DROP TABLE tbl\_name;

Query with Output:



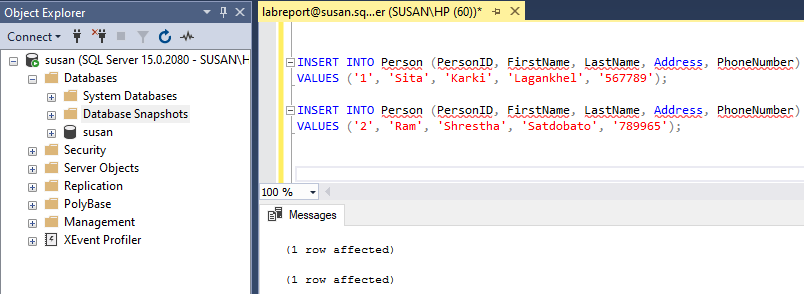
3.Insert into the table, select all, and select distinct from the table.

Syntax:

INSERT INTO table\_name(column1, column2, column3,…)

VALUES (value1, value2, value3,....);

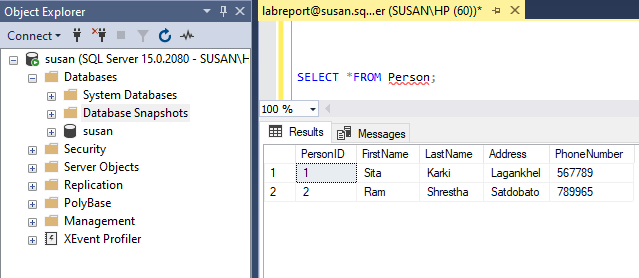
Query with Output:



Syntax:

SELECT \* FROM table\_name;

Query with Output:

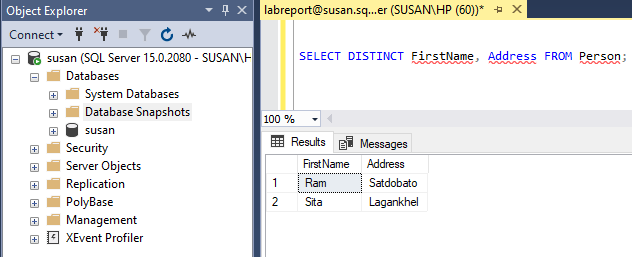


Syntax:

SELECT DISTINCT column1, column2, ....

FROM table\_name;

Query with Output:



4.To show the auto increment in the primary key

Syntax:

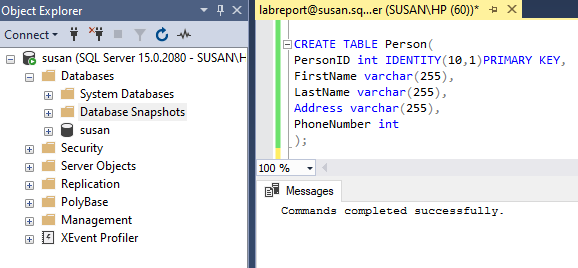
CREATE TABLE table\_name(

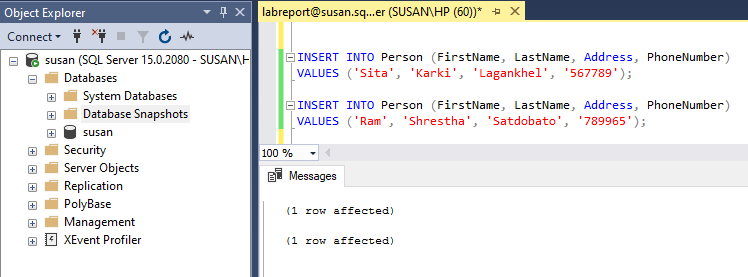
Column int INDENTITY(starting digits, increment by number) PRIMARY,

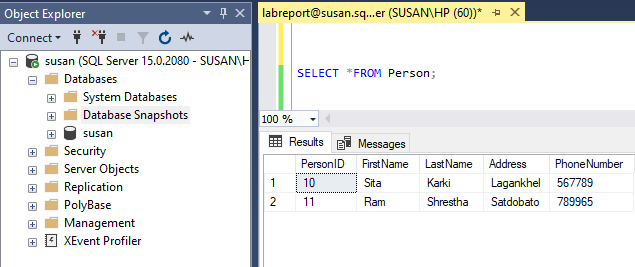
Coulmn 2,

Coulmn 3….);

Query with Output:







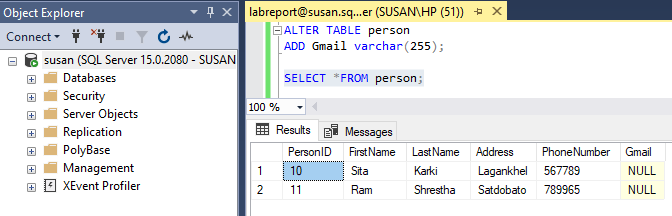
1. Altering the table and adding a column to the table

Syntax:

ALTER TABLE tb\_tablename

ADD column\_name datatype ;

Query with Output:



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

1. Syntax:(Unique)

CREATE TABLE tb\_tablename(

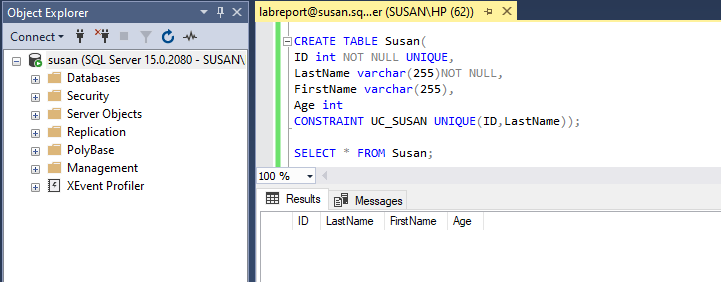
Column 1 datatype NOT NULL UNIQUE,

Coulmn 2 NOT NULL,

Coulmn 3….

CONSTRAINT UC\_TABLENAME UNIQUE (column\_name,column\_name));

Query with Output:

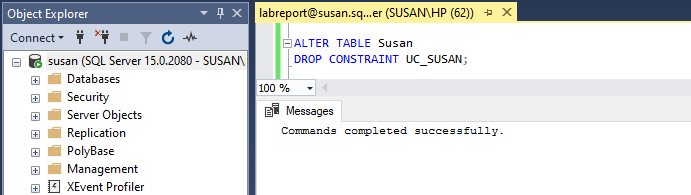


1. Syntax: (Droping Unique)

ALTER TABLE table\_name

DROP CONSTRAINT UC\_TABLENAME;

Query with Output:



1. Syntax:(Primary Key)

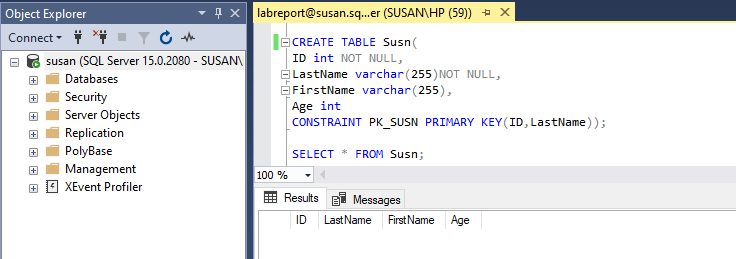
CREATE TABLE tb\_tablename(

Column 1 datatype NOT NULL UNIQUE,

Coulmn 2 NOT NULL,

CONSTRAINT PK\_TABLENAME PRIMARY KEY (column\_name,column\_name));

Query with Output:



1. Syntax:(Function Key)

CREATE TABLE tb\_tablename(

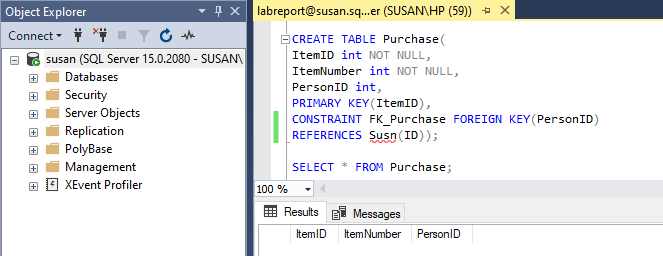
Column 1 datatype NOT NULL UNIQUE,

Coulmn 2 NOT NULL,

CONSTRAINT FK\_TABLENAME FOREIGN KEY (column\_name)

REFERENCES tablename(column\_name));

Query with Output:



7.Use the WHERE clause in the query.

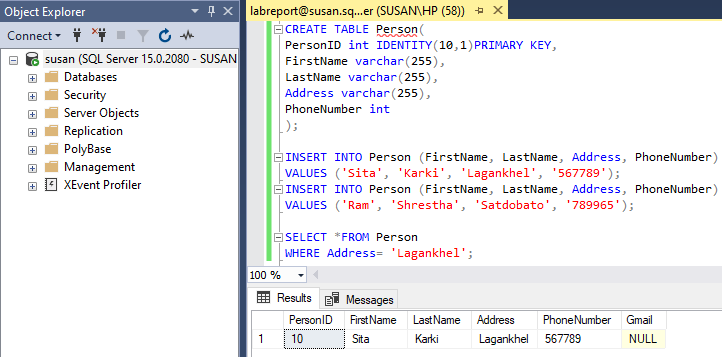
Syntax:

SELECT column1, column2,….

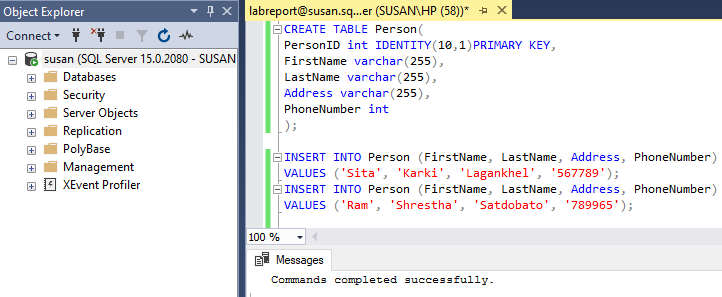
FROM table\_name

WHERE condition;

Query with Output:



1. Use AND, OR, NOT queries



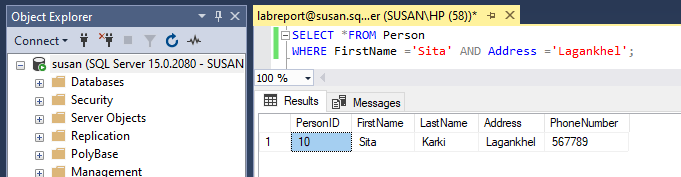
1. Syntax:(AND)

SELECT column1, column2, …

FROM table\_name

WHERE condition1 AND condition2 AND condition3….;

Query with Output:



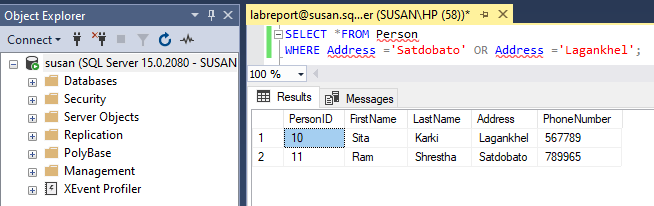
1. Syntax:(OR)

SELECT column1, column2, …

FROM table\_name

WHERE condition1 OR condition 2 or condition3….;

Query with Output:



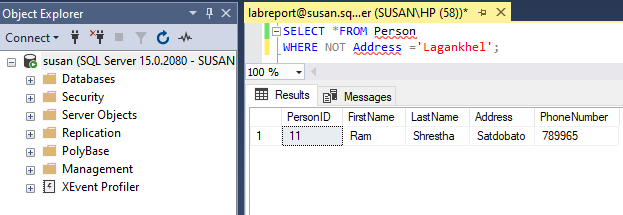
1. Syntax:(NOT)

SELECT column1, column2, …

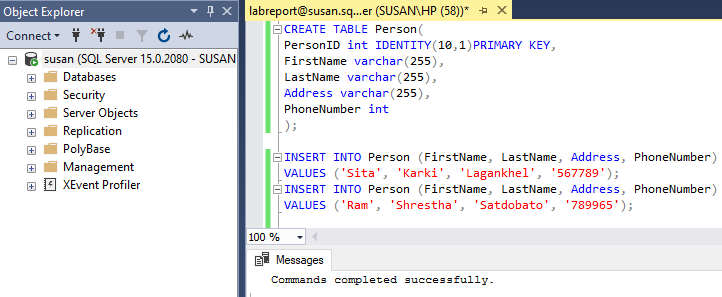
FROM table\_name

WHERE NOT condition;

Query with Output:



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



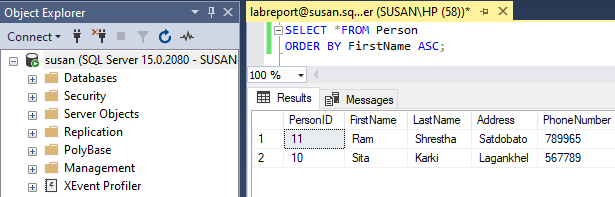
1. Syntax:(.Ascending)

SELECT column1, column2,….

FROM table\_name

ORDER BY column1, column2,….ASC;

Query with Output:



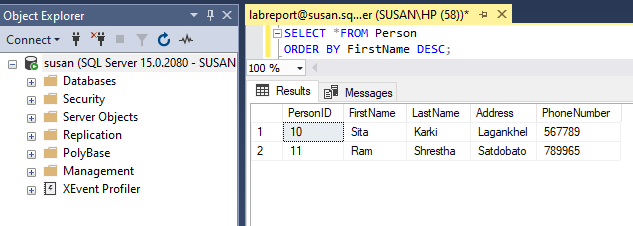
1. Syntax: (Descending)

SELECT column1, column2,….

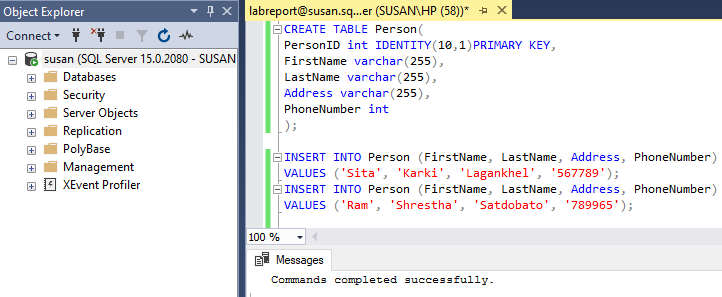
FROM table\_name

ORDER BY column1, column2,…. DESC;

Query with Output:



10.Update and delete table fields.



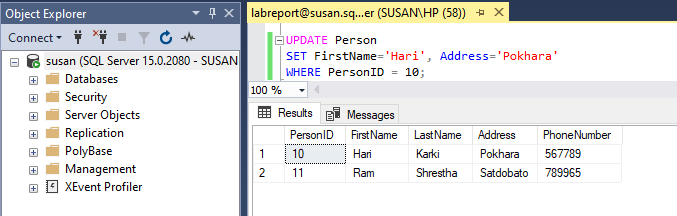
1. Syntax:(UPDATE)

UPDATE table\_name

SET column1= value1, column2=value2,…..

WHERE condition;

Query with Output:

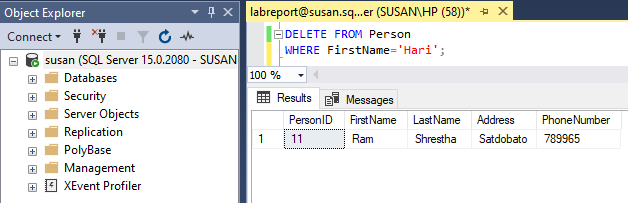


1. Syntax:(DELETE)

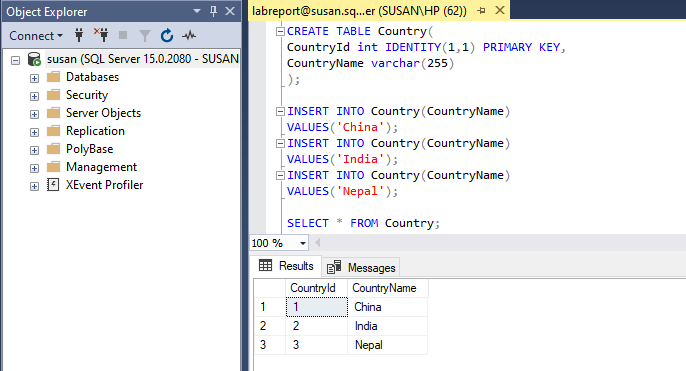
DELETE FROM table\_name

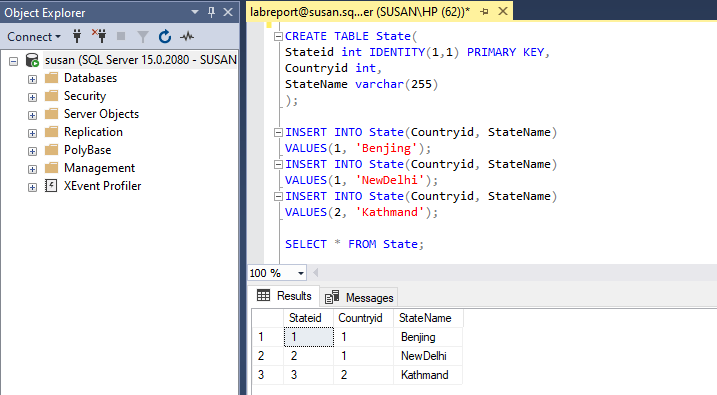
WHERE condition;

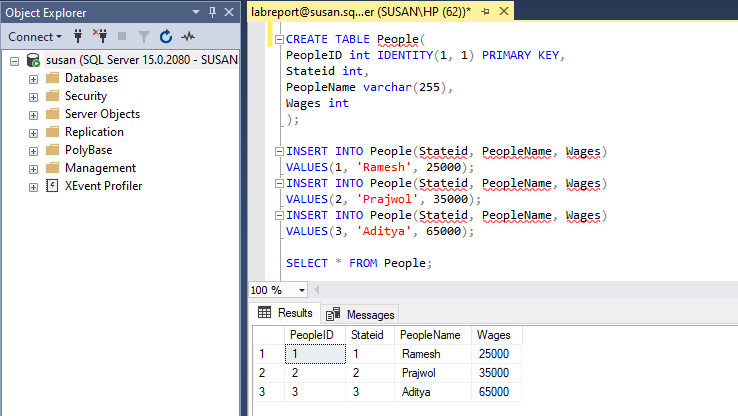
Query with Output:



1. Use Join queries







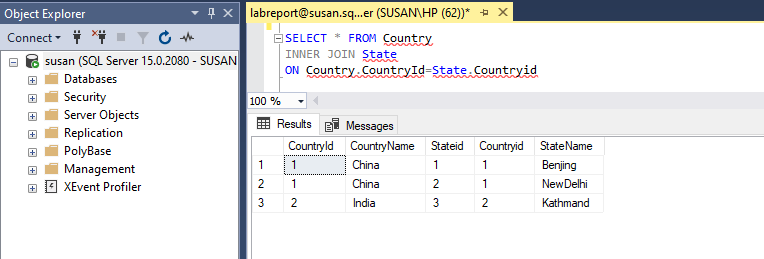
i.Syntax: (INNER JOIN)

SELECT column\_name(s) FROM table1

INNER JOIN table2

ON table1.column\_name = table2.column\_name;

Query with Output:



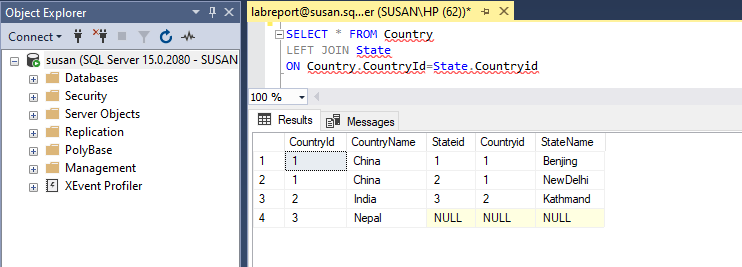
ii.Syntax: (LEFT JOIN )

SELECT column\_name(s) FROM table1

LEFT JOIN table2

ON table1.column\_name = table2.column\_name;

Query with Output:



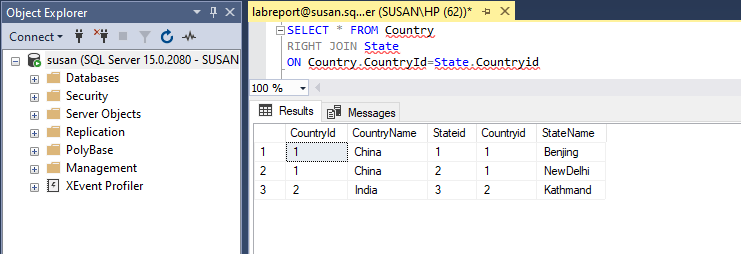
iii.Syntax: (RIGHT JOIN)

SELECT column\_name(s) FROM table1

RIGHT JOIN table2

ON table1.column\_name = table2.column\_name;

Query with Output:



iv.Syntax: (FULL OUTER JOIN)

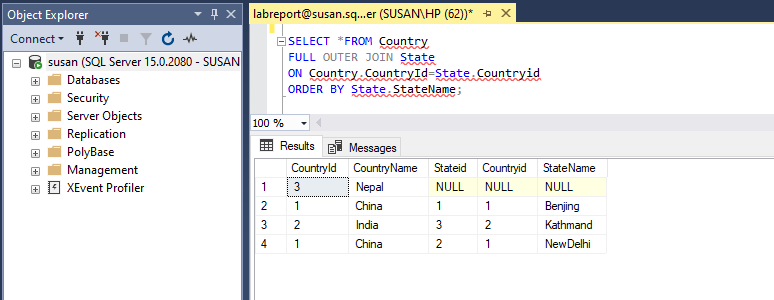
SELECT column\_name(s) FROM table1

FULL OUTER JOIN table2

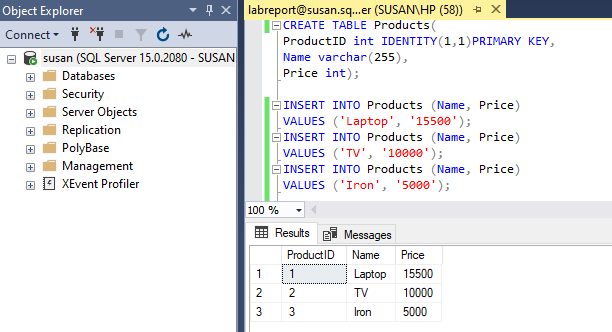
ON table1.column\_name = table2.column\_name;

WHERE condition;

Query with Output:



12. Use the min and max functions.



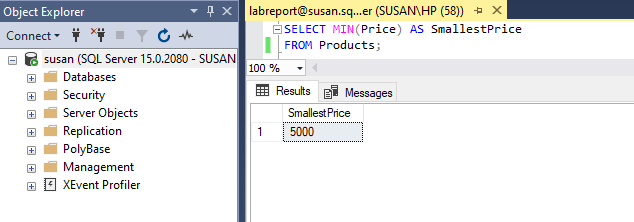
1. Syntax: (MIN)

SELECT MIN(column\_name)

FROM table\_name

WHERE condition;

Query with Output:



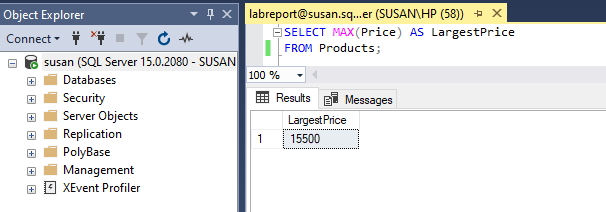
1. Syntax: (MAX)

SELECT MAX(column\_name)

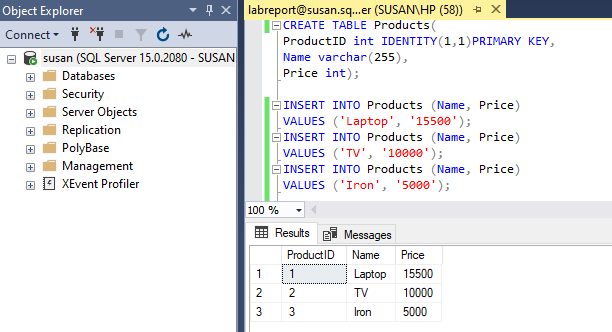
FROM table\_name

WHERE condition;

Query with Output:



13. Use the count, average, and sum functions.



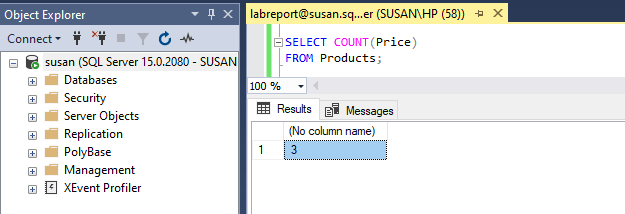
1. Syntax:(COUNT)

SELECT COUNT (column\_name)

FROM table\_names

WHERE condition;

Query with Output:



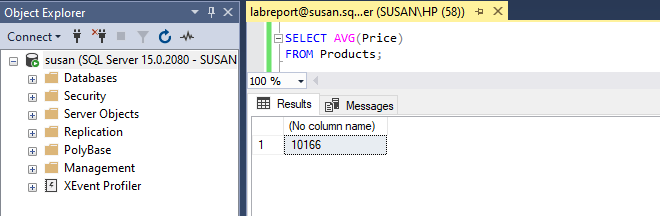
1. Syntax:(AVERAGE)

SELECT AVG(column\_name)

FROM table\_name

WHERE condition;

Query with Output:



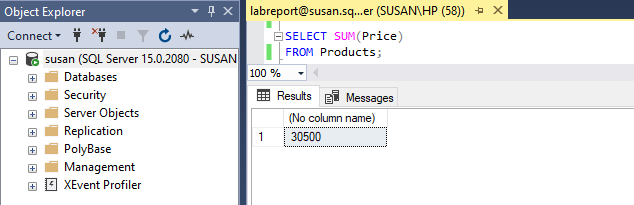
1. Syntax:(SUM)

SELECT SUM (column\_name)

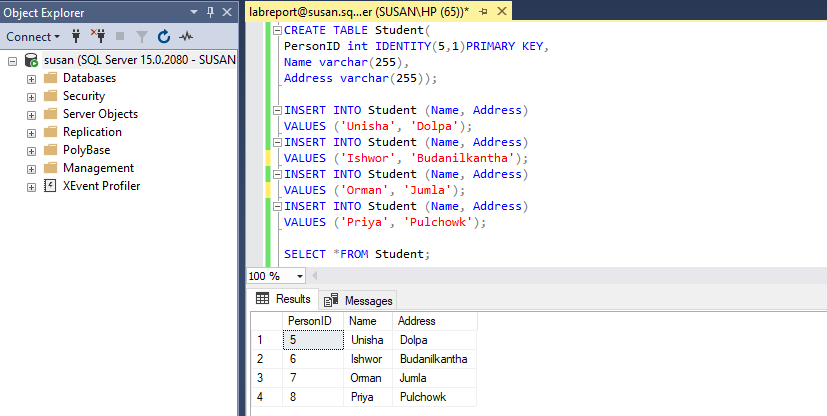
FROM table\_name

WHERE condition;

Query with Output:



1. Display students' names with "or" in any position.

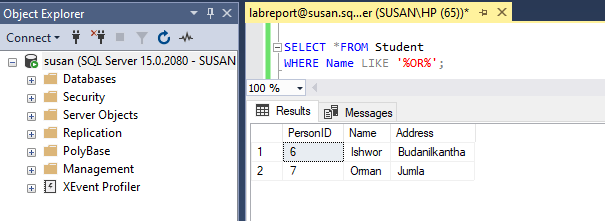


Syntax:

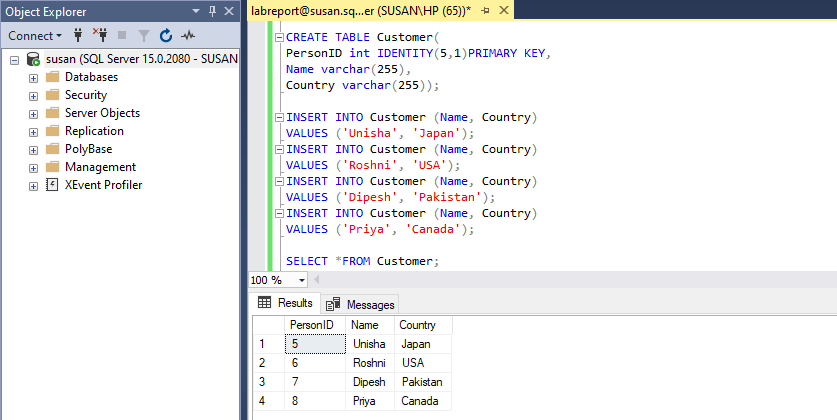
SELECT \* FROM table\_name

WHERE column\_name LIKE ‘%a%’;

Query with Output:



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



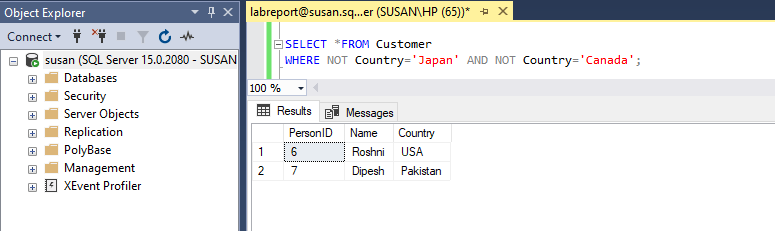
Syntax:

SELECT column1, column2, .....

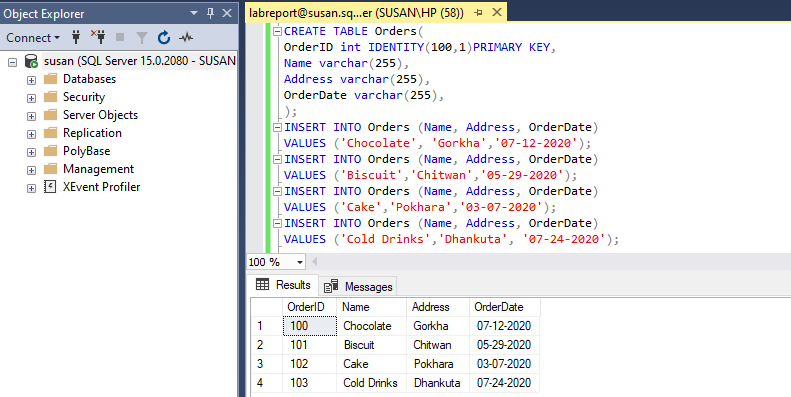
FROM table\_name

WHERE NOT condition ;

Query with Output:



16. Display all orders with an OrderDate between '01-July-2020' and '31-July-2020'.



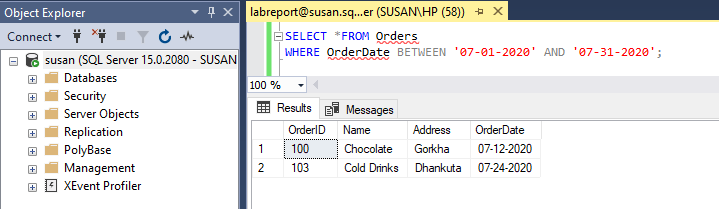
Syntax:

SELECT column\_name(s)

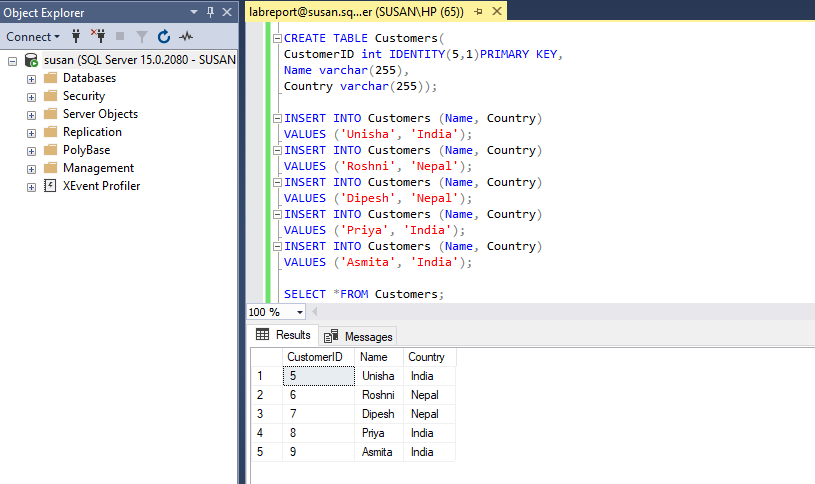
FROM table\_name

WHERE column\_name BETWEEN value1 AND value2;

Query with Output:



17. Count the number of customers in each country.



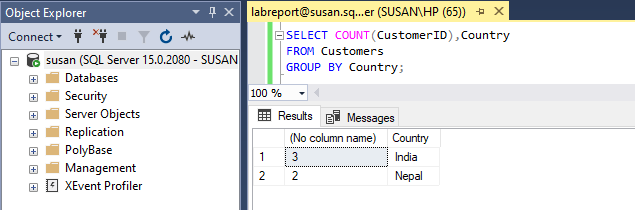
Syntax:

SELECT COUNT (column\_name)

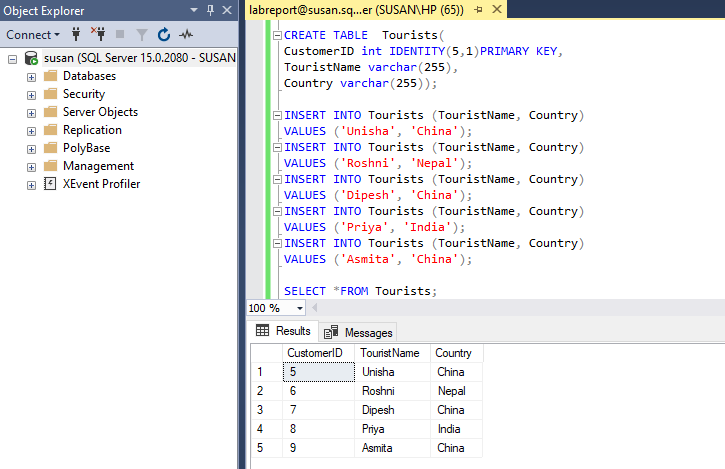
FROM table\_name

GROUP BY column\_name;

Query with Output:



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



Syntax:

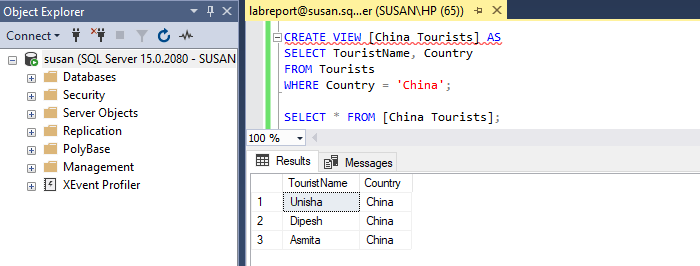
CREATE VIEW view\_name AS

SELECT column1, column2, …..

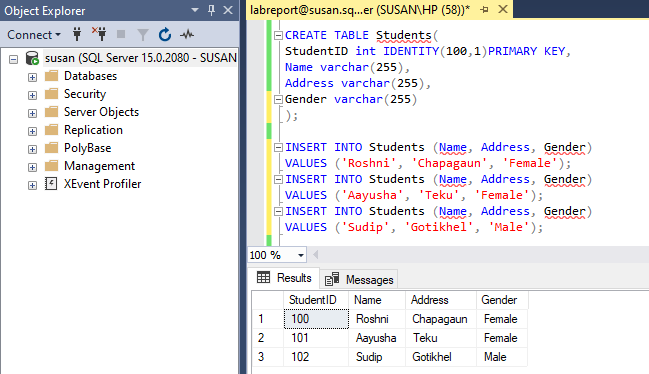
FROM table\_name

WHERE condition;

Query with Output:



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



Syntax:

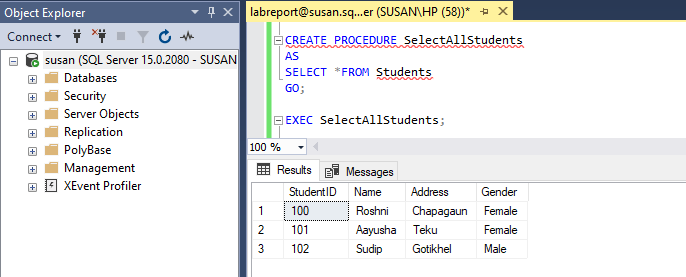
CREATE PROCEDURE procedure\_name

AS

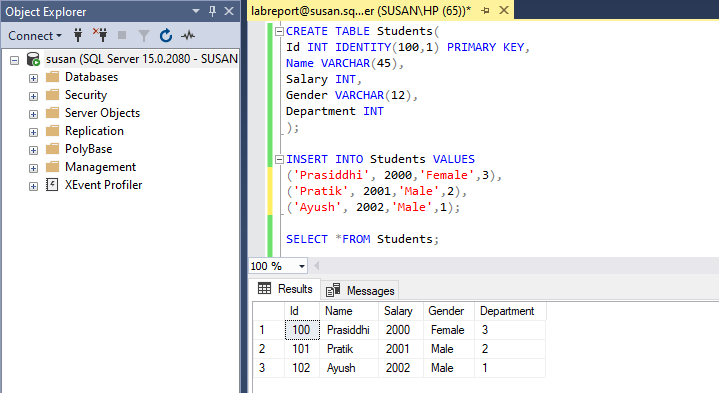
sql\_statement

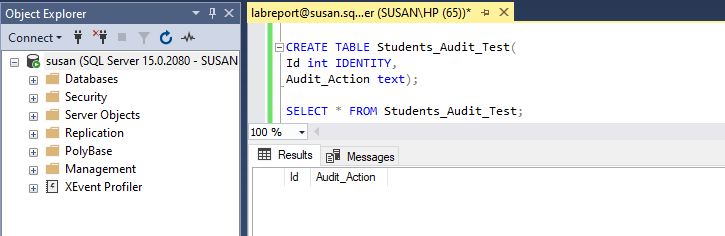
GO;

Query with Output:



1. 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:

