DBMS-LAB ASSIGNMENT 7

NAME: KASHA SINGH

REG. NO.: 19BCS051

1. Write two stored Procedures relevant to your database.

QUERY:

USE T3\_Travel;

CREATE PROCEDURE Names\_starting\_with\_M

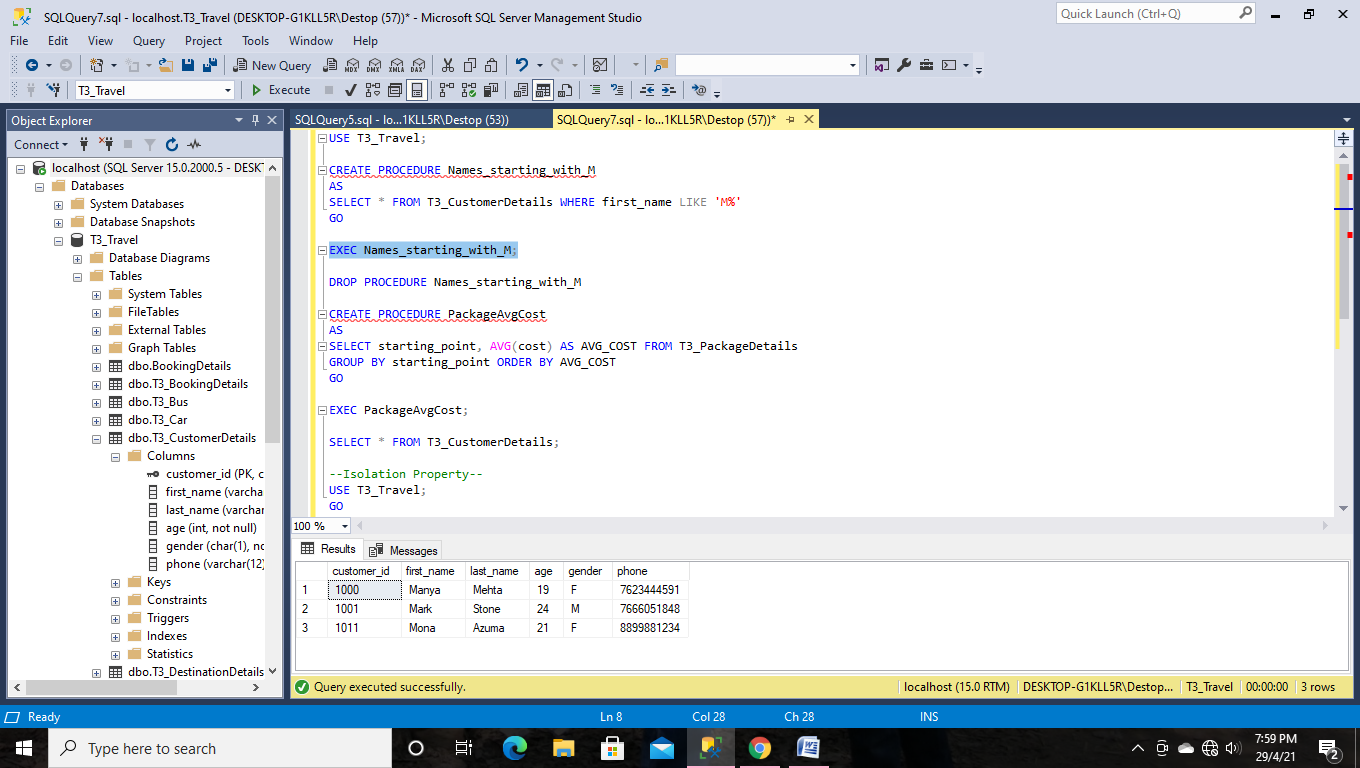
AS

SELECT \* FROM T3\_CustomerDetails WHERE first\_name LIKE 'M%'

GO

EXEC Names\_starting\_with\_M;

DATABASE OUTPUT:



CREATE PROCEDURE PackageAvgCost

AS

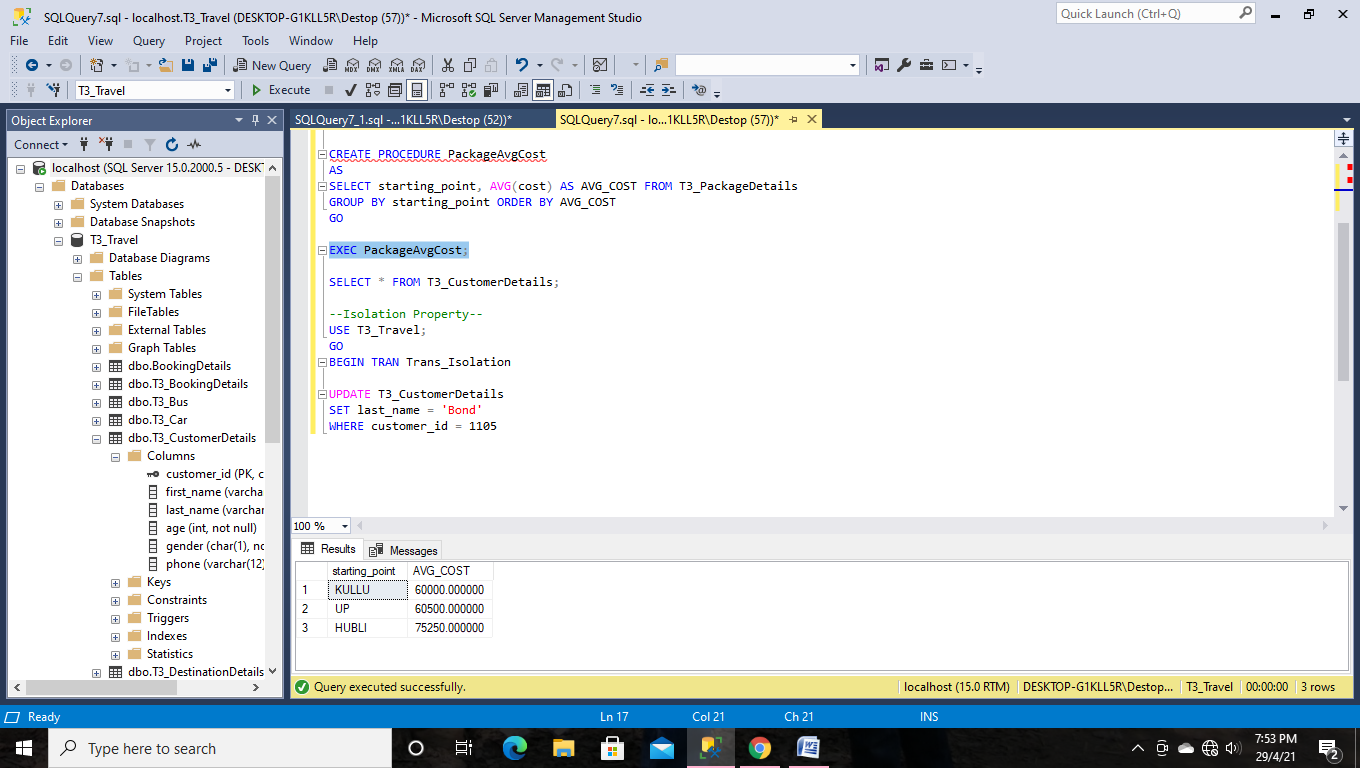
SELECT starting\_point, AVG(cost) AS AVG\_COST FROM T3\_PackageDetails

GROUP BY starting\_point ORDER BY AVG\_COST

GO

EXEC PackageAvgCost;

DATABASE OUTPUT:



2. Write a transaction to illustrate atomicity (related to your database)

QUERY:

--Atomicity--

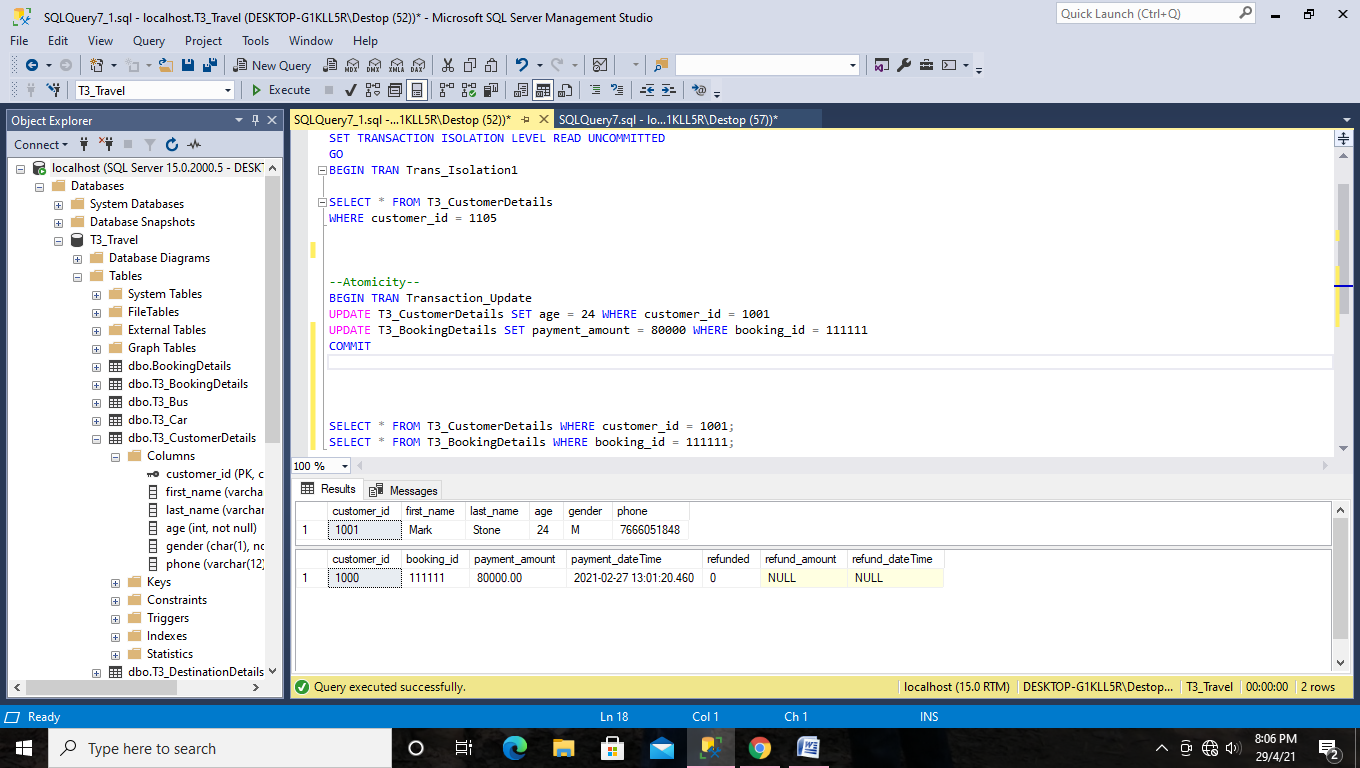
BEGIN TRAN Transaction\_Update

UPDATE T3\_CustomerDetails SET age = 24 WHERE customer\_id = 1001

UPDATE T3\_BookingDetails SET payment\_amount = 80000 where booking\_id = 111111

COMMIT

DATABASE OUTPUT:



As the transaction is atomic, both of the updates on the two separate tables will commit together, or they will rollback together.

3. Write a transaction to illustrate isolation level. It can be on commit or uncommit read (related to your database)

Window 1:

--Isolation Property--

USE T3\_Travel;

GO

BEGIN TRAN Trans\_Isolation

UPDATE T3\_CustomerDetails

SET last\_name = 'Bond'

WHERE customer\_id = 1105

Window 2:

USE T3\_Travel;

GO

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED

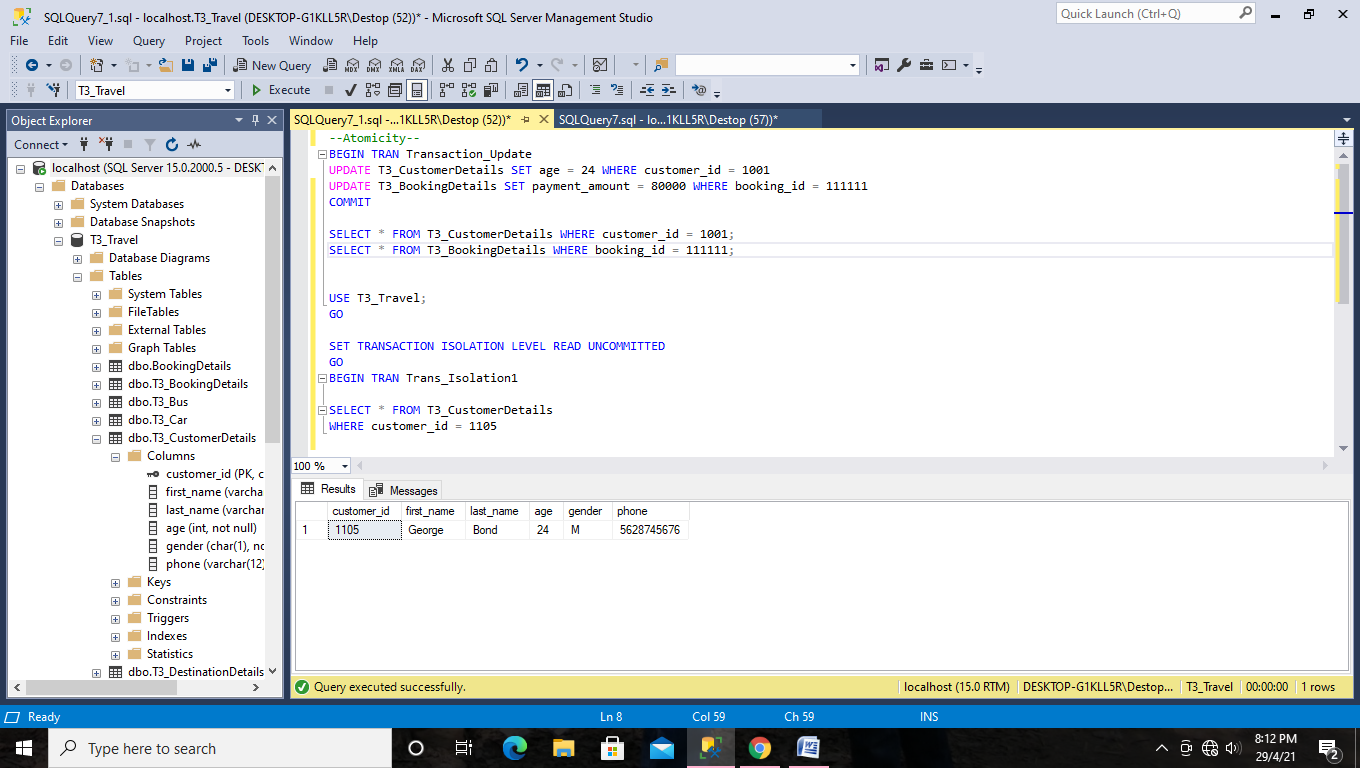
GO

BEGIN TRAN Trans\_Isolation1

SELECT \* FROM T3\_CustomerDetails

WHERE customer\_id = 1105

DATABASE OUTPUT:



When we set the isolation level to read uncommitted, we will be able to see the last\_name set to ‘Bond’, called Dirty Read.