**DBMS LAB ASSIGNMENT-7**

**B NEELAKANTESWAR**

**19bcs020 T12**

1)Write two stored Procedures relevant to your database.

**Query1**:

***CREATE PROCEDURE Regional\_phn\_no***

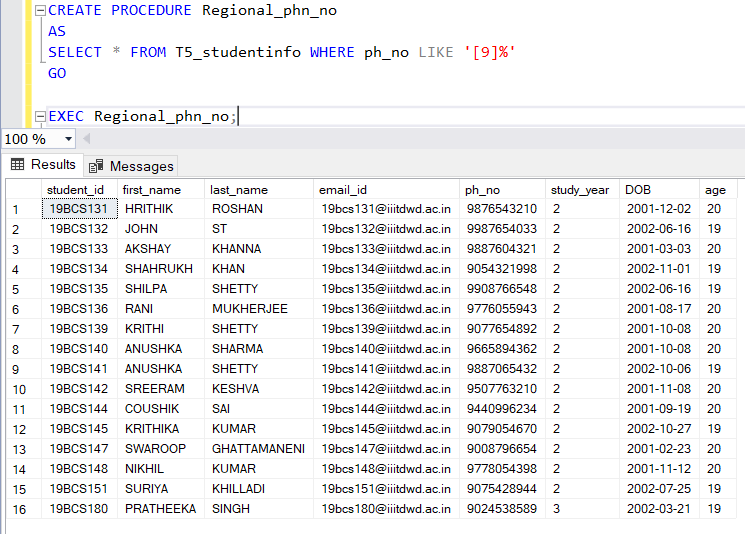
***AS***

***SELECT \* FROM T12\_studentinfo WHERE ph\_no LIKE '[9]%'***

***GO***

***EXEC Regional\_phn\_no;***

OUTPUT:



**Query2**:

*CREATE PROCEDURE avg\_fee\_paid\_by\_region*

*AS*

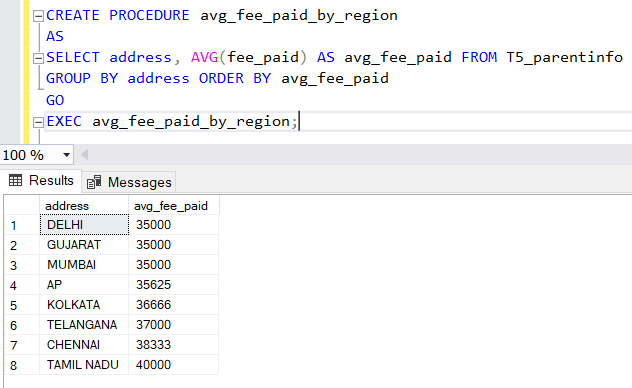
*SELECT address, AVG(fee\_paid) AS avg\_fee\_paid FROM T12\_parentinfo*

*GROUP BY address ORDER BY avg\_fee\_paid*

*GO*

*EXEC avg\_fee\_paid\_by\_region;*

OUTPUT:



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

**Query**:

*BEGIN TRAN Transaction\_grades*

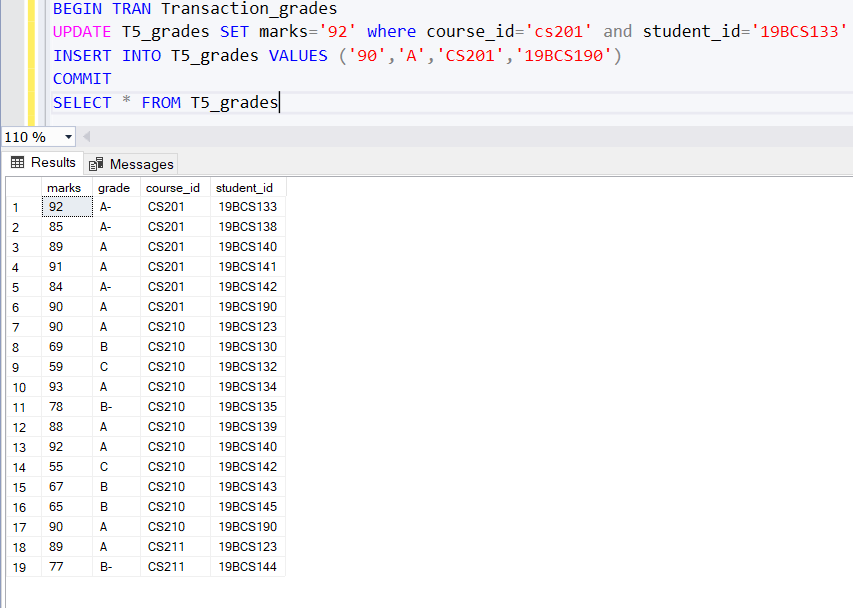
*UPDATE T12\_grades SET marks='92' where course\_id='cs201' and student\_id='19BCS133'*

*INSERT INTO T12\_grades VALUES ('90','A','CS201','19BCS190')*

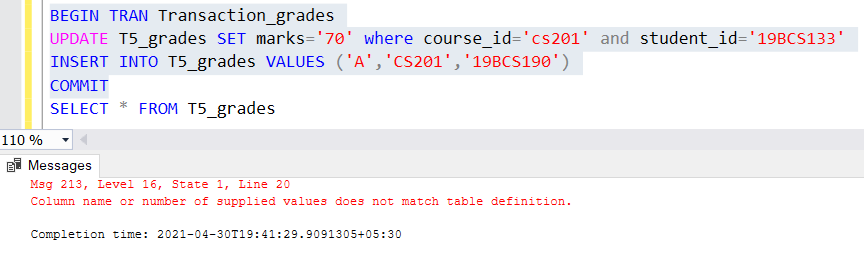
*COMMIT*

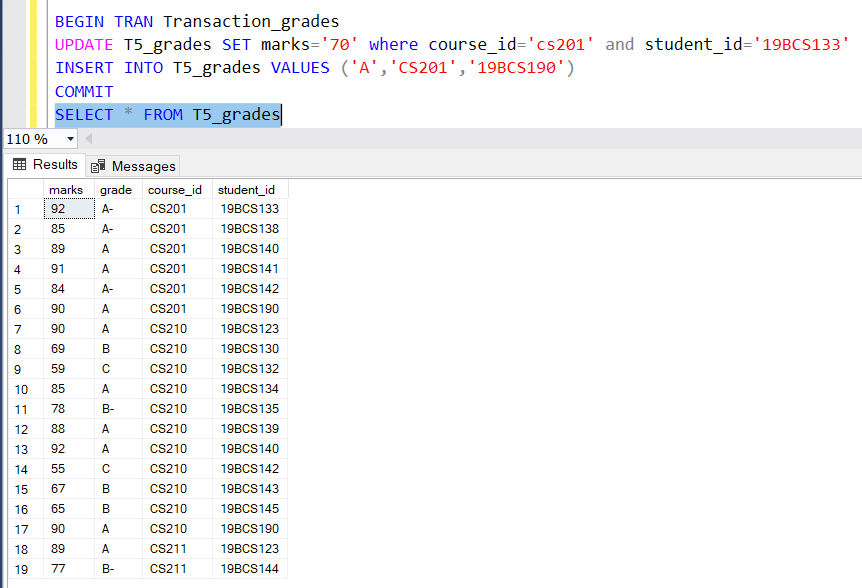
*SELECT \* FROM T12\_grades*

OUTPUT:



Now, let us we will insert wrong information in the T12\_grades table to fail the insertion deliberately.





Here, we can clearly see that the transaction got rolled back as error have been occurred in insert operation. And thus, update have not been worked due to atomic property and the previous values of the table are displayed.

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

**Query**:

*USE school;*

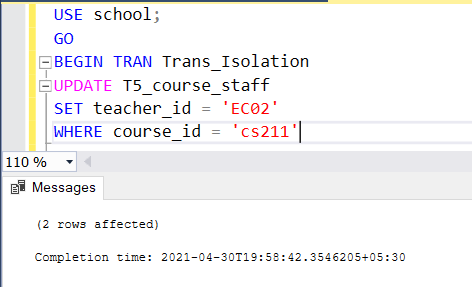
*GO*

*BEGIN TRAN Trans\_Isolation*

*UPDATE T12\_course\_staff*

*SET teacher\_id = 'EC02'*

*WHERE course\_id = 'cs211'*



*USE school;*

*GO*

*SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED*

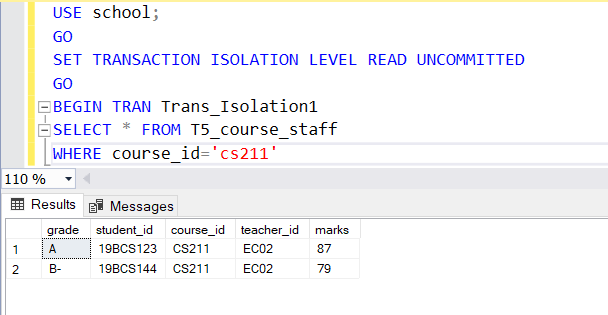
*GO*

*BEGIN TRAN Trans\_Isolation1*

*SELECT \* FROM T12\_course\_staff*

*WHERE course\_id='cs211'*

OUTPUT:



* When we set the isolation level to read uncommitted, we will be able to see the teacher\_id set to ‘EC02’, called Dirty Read.