

# DBMS LAB ASSIGNMENT-7

NAME – B. Ieeshasree

ROLL NO. – 19BCS021

BRANCH - CSE

DB – Travel agency

1. Write two stored Procedures relevant to your database.

CODE:

```
CREATE PROCEDURE Namestarting_with_C
AS
SELECT * FROM T10_customerdetailsNEW WHERE customer_name LIKE 'C%';
GO
CREATE PROCEDURE Selectbyhotelname @hotel_name varchar(max) , @hotel_rent int
AS
SELECT * FROM T10_hoteldetails WHERE hotel_name= @hotel_name AND
hotel_rent=@hotel_rent;
GO
EXEC Namestarting_with_C;
EXEC Selectbyhotelname @hotel_name='RUCHIKA', @hotel_rent=500
```

OUTPUT:

Results		Messages					
customer_id	customer_name	customer_mobile	customer_email	customer_username	customer_password	customer_address	
1	1	CHAITRA	88815	chara	chaieddy	msaichatra	kaapa
2	101	CHAITRA	88815	chara	chaieddy	msaichatra	kaapa

hotel_id	hotel_name	hotel_type	hotel_rent	hotel_description	hotel_address	booking_id	
1	203	RUCHIKA	FAMILY	500	NULL	CHSOLE	NULL

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

T10\_packagedetails table before transaction:

Results Messages										
	package_id	package_tour_id	package_name	package_amount	package_total	package_type	package_description	package_amountFinal	package_Final	hotel_id
1	31	21	RIVERPAC	440	590	DUAL	NULL	0	NULL	NULL
2	76	33	NEWLY WED	700	1890	DUAL	NULL	2500	1100	77
3	77	33	RUCIASS	770	1800	DUAL	NULL	2000	100	78
4	89	213	COUPLE	1788	2890	DUAL	NULL	3500	5100	90
5	203	28	RUCHIKAPAC	550	1000	NEW	NULL	NULL	NULL	NULL
6	331	21	RIVERPAC	440	590	DUAL	NULL	30	NULL	NULL

CODE:

```
USE T10_Travel;
GO
BEGIN TRANSACTION
UPDATE T10_packagedetails SET package_name='RIVERPAC' WHERE
package_id=31
INSERT INTO T10_packagedetails
VALUES(32,212,'RIDSVERPAC',4440,5904,'DUAL TIME',null,9000,NULL,NULL)
COMMIT TRANSACTION
SELECT * FROM T10_packagedetails
```

OUTPUT:

85	USE T10_Travel;
86	GO
87	BEGIN TRANSACTION
88	UPDATE T10_packagedetails SET package_name='RIVERPAC' WHERE
89	package_id=31
90	INSERT INTO T10_packagedetails
91	VALUES(32,212,'RIDSVERPAC',4440,5904,'DUAL TIME',null,9000,NULL,NULL)
92	COMMIT TRANSACTION
93	SELECT * FROM T10_packagedetails

Results Messages										
	package_id	package_tour_id	package_name	package_amount	package_total	package_type	package_description	package_amountFinal	package_Final	hotel_id
1	31	21	RIVERPAC	440	590	DUAL	NULL	0	NULL	NULL
2	32	212	RIDSVERPAC	4440	5904	DUAL TIME	NULL	9000	NULL	NULL
3	76	33	NEWLY WED	700	1890	DUAL	NULL	2500	1100	77
4	77	33	RUCIASS	770	1800	DUAL	NULL	2000	100	78
5	89	213	COUPLE	1788	2890	DUAL	NULL	3500	5100	90
6	203	28	RUCHIKAPAC	550	1000	NEW	NULL	NULL	NULL	NULL
7	331	21	RIVERPAC	440	590	DUAL	NULL	30	NULL	NULL

AFTER TRANSACTION (when given input is not valid):  
QUERY:

CODE:

```
USE T10_Travel;
GO
BEGIN TRANSACTION
UPDATE T10_packagedetails SET package_name='RIVERPAC' WHERE
package_id=31
INSERT INTO T10_packagedetails
VALUES('RIDSVERPAC',4440,5904,'DUAL TIME',null,9000,NULL,NULL)
COMMIT TRANSACTION
SELECT * FROM T10_packagedetails
```

OUTPUT:

```
7:03:00 PM Started executing query at line 88
Msg 213, Level 16, State 1, Line 6
Column name or number of supplied values does not match table definition.
Total execution time: 00:00:00.248
```

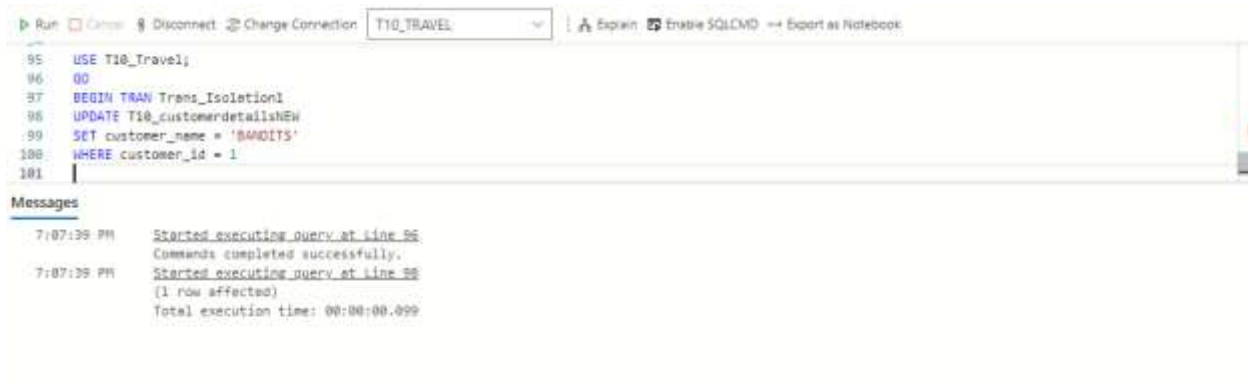
When error occurs in the transaction it gets rolled back.  
So the table did not get updated .It means all statements  
inside a transaction should either get succeed or fail as a  
unit. So this is the atomicity property.

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

CODE:

```
USE T10_Travel;
GO
BEGIN TRAN Trans_Isolation1
UPDATE T10_customerdetailsNEW
SET customer_name = 'BANDITS'
WHERE customer_id = 1
```

## OUTPUT:



The screenshot shows a SQL Server Enterprise Manager interface. The top toolbar includes buttons for Run, Cancel, Disconnect, Change Connection, and a dropdown menu showing 'T10\_TRAVEL'. To the right are links for Explain, Enable SQLCMD, and Export as Notebook. The query window contains the following SQL code:

```
95 USE T10_Travel;  
96 GO  
97 BEGIN TRAN Trans_Isolation1  
98 UPDATE T10_customerdetailsNEW  
99 SET customer_name = 'BANDITS'  
100 WHERE customer_id = 1  
101
```

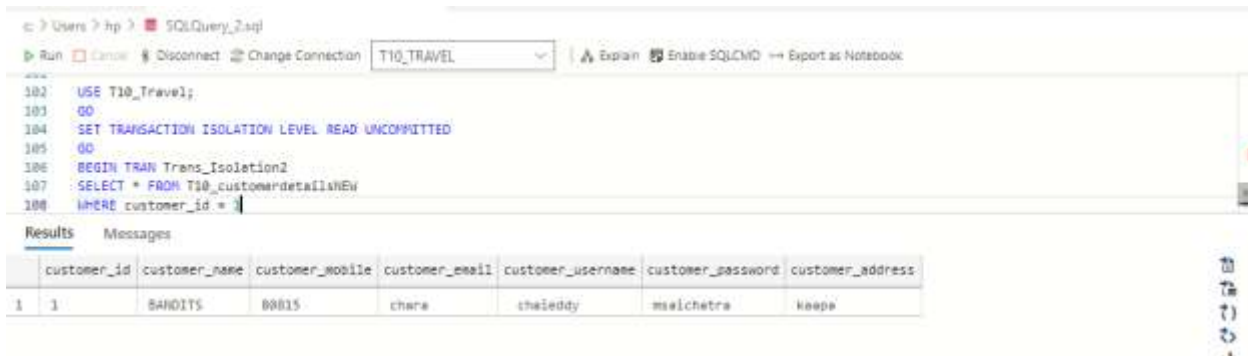
Below the query window is a 'Messages' tab showing the following output:

```
7:07:39 PM Started executing query at line 96  
Commands completed successfully.  
7:07:39 PM Started executing query at line 98  
(1 row affected)  
Total execution time: 00:00:00.099
```

## QUERY-2

```
USE T10_Travel;  
GO  
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED  
GO  
BEGIN TRAN Trans_Isolation2  
SELECT * FROM T10_customerdetailsNEW  
WHERE customer_id = 1
```

## OUTPUT:



The screenshot shows the same SQL Server Enterprise Manager interface. The query window contains the following SQL code:

```
102 USE T10_Travel;  
103 GO  
104 SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED  
105 GO  
106 BEGIN TRAN Trans_Isolation2  
107 SELECT * FROM T10_customerdetailsNEW  
108 WHERE customer_id = 1
```

Below the query window is a 'Results' tab showing the following output:

customer_id	customer_name	customer_mobile	customer_email	customer_username	customer_password	customer_address
1	BANDITS	00015	chere	cheleddy	maichetra	keeps

When we set the isolation level to read uncommitted, we will be able to see the customer\_name set to 'BANDITS', called Dirty Read.