DBMS LAB ASSIGNMENT-7

Submitted by:
R.Mounika,
19BCS124.

1. Write two stored Procedures relevant to your database.

QUERY:

```
CREATE PROCEDURE Selectall
AS
SELECT*FROM T5_Studentinfo
GO;

EXEC Selectall;
```

OUTPUT:

```
CREATE PROCEDURE Selectall
AS
SELECT*FROM T5_Studentinfo
GO;

EXEC Selectall;
```

```
%
Messages
Commands completed successfully.

Completion time: 2021-04-30T19:03:06.7246345+05:30
```

QUERY:

```
CREATE PROCEDURE selectstudent @dob date, @age int
AS
SELECT*FROM T5_Studentinfo WHERE [D.O.B] = @dob  AND Age = @age;
GO

EXEC selectstudent @dob="2002-05-30", @age=19;
```

OUTPUT:

```
CREATE PROCEDURE selectstudent @dob date, @age int
AS
SELECT*FROM T5_Studentinfo WHERE [D.O.B] = @dob  AND Age = @age;
GO

EXEC selectstudent @dob="2002-05-30", @age=19;
```

100 %

Results | Messages

	Student_ID	First_Name	Last_Name	D.O.B	Age	Email_ID	Phone_No.	Study_Year
1	104	Bhavitha	Reddy	2002-05-30	19	@nitap.ac.in	9999900000	2021

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

QUERY:

```
USE SCHOOL;
GO
BEGIN TRANSACTION
UPDATE T5_Studentinfo SET First_Name='Sai' WHERE Student_ID=108
INSERT INTO T5_Studentinfo VALUES(125,'Saritha','Kumari','2002-03-
04','@nitk.ac.in','9988004567',2021)
COMMIT TRANSACTION
SELECT*FROM T5_Studentinfo;
```

OUTPUT:

```
USE SCHOOL;
GO
BEGIN TRANSACTION
UPDATE T5_Studentinfo SET First_Name='Sai' WHERE Student_ID=108
INSERT INTO T5_Studentinfo VALUES(125,'Saritha','Kumari','2002-03-04','@nitk.ac.in','9988004567',2021)
COMMIT TRANSACTION
SELECT*FROM T5_Studentinfo;
```

100 %

Results | Messages

	Student_ID	First_Name	Last_Name	DOB	Age	Email_ID	Phone_No.	Study_Year
1	104	Bhavitha	Reddy	2002-05-30	19	@nitap.ac.in	9999900000	2021
2	105	Prasanna	Varanasi	2002-01-11	19	@nitrr.ac.in	9966331363	2020
3	106	Anu	Kirti	2002-05-14	19	2002@gmail.com	5555533322	2019
4	107	Krithika	Sen	2002-05-25	19	@iiitdwd.ac.in	4444666678	2019
5	108	Sai	Priya	2000-01-02	21	@nitr.ac.in	9993336660	2018
6	109	Ananya	Bhaskar	2001-01-20	20	@nitk.ac.in	8888666680	2019
7	110	Pragnya	Nayini	2000-03-20	21	@ou.ac.in	5678901234	2018
8	111	Swetha	Sree	2002-03-25	19	@au.ac.in	4555666432	2020
9	112	Swarna	Reddy	2001-06-24	20	@anits.ac.in	7865402139	2019
10	113	Indu	Priya	2003-09-05	18	@jntuk.ac.in	9099883344	2021
11	114	Sonal	Rai	2002-06-06	19	@iiitdwd.ac.in	9133790522	2020
12	115	Abhishek	Yadav	2001-04-05	20	@anits.ac.in	3456789021	2019
13	116	Lalith	Gupta	2000-12-12	21	@iiitdwd.ac.in	4444777700	2018
14	117	Sanjay	Shetty	2001-05-20	20	@rsali.ac.in	3333777788	2020
15	124	Mounika	Rayudu	2002-05-20	19	@iiitdwd.ac.in	9908535176	2020
16	125	Saritha	Kumari	2002-03-04	19	@nitk.ac.in	9988004567	2021

Now, we will insert wrong information in the “T5_Studentinfo” Table to fail the transaction.

OUTPUT (after transaction):

```
USE SCHOOL;
GO
BEGIN TRANSACTION
UPDATE T5_Studentinfo SET First_Name='Sai' WHERE Student_ID=108
INSERT INTO T5_Studentinfo VALUES(126,'Sahithi','Sai','@nitw.ac.in','9988004567',2021)
COMMIT TRANSACTION
SELECT*FROM T5_Studentinfo;
```

```
Messages
Msg 213, Level 16, State 1, Line 23
Column name or number of supplied values does not match table definition.

Completion time: 2021-04-30T20:48:42.1072191+05:30
```

```
GO
BEGIN TRANSACTION
UPDATE T5_Studentinfo SET First_Name='Sai' WHERE Student_ID=108
INSERT INTO T5_Studentinfo VALUES(126,'Sahithi','Sai','@nitw.ac.in','9988004567',2021)
COMMIT TRANSACTION
SELECT*FROM T5_Studentinfo;
```

Results | Messages

	Student_ID	First_Name	Last_Name	DOB	Age	Email_ID	Phone_No.	Study_Year
1	104	Bhavitha	Reddy	2002-05-30	19	@nitap.ac.in	9999900000	2021
2	105	Prasanna	Varanasi	2002-01-11	19	@nitrr.ac.in	9966331363	2020
3	106	Anu	Kirti	2002-05-14	19	2002@gmail.com	5555533322	2019
4	107	Krithika	Sen	2002-05-25	19	@iiitdwd.ac.in	4444666678	2019
5	108	Sai	Priya	2000-01-02	21	@nitr.ac.in	9993336660	2018
6	109	Ananya	Bhaskar	2001-01-20	20	@nitk.ac.in	8888666680	2019
7	110	Pragnya	Nayini	2000-03-20	21	@ou.ac.in	5678901234	2018
8	111	Swetha	Sree	2002-03-25	19	@au.ac.in	4555666432	2020
9	112	Swarna	Reddy	2001-06-24	20	@anits.ac.in	7865402139	2019
10	113	Indu	Priya	2003-09-05	18	@jntuk.ac.in	9099883344	2021
11	114	Sonal	Rai	2002-06-06	19	@iiitdwd.ac.in	9133790522	2020
12	115	Abhishek	Yadav	2001-04-05	20	@anits.ac.in	3456789021	2019
13	116	Lalith	Gupta	2000-12-12	21	@iiitdwd.ac.in	4444777700	2018
14	117	Sanjay	Shetty	2001-05-20	20	@risali.ac.in	3333777788	2020
15	124	Mounika	Rayudu	2002-05-20	19	@iiitdwd.ac.in	9908535176	2020
16	125	Saritha	Kumari	2002-03-04	19	@nitk.ac.in	9988004567	2021

Here, we can clearly see the transaction got rolled back as error occurred in insert operation. And thus, update haven't 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:

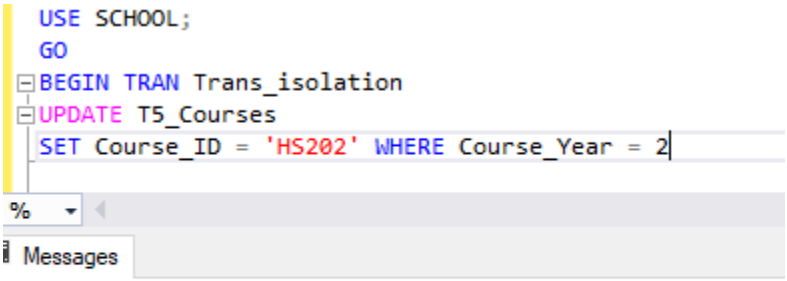
Window 1:

```
USE SCHOOL;
GO
BEGIN TRAN Trans_isolation
UPDATE T5_Courses
SET Course_ID = 'HS202' WHERE Course_Year = 2
```

Window 2:

```
USE SCHOOL;
GO
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED
GO
BEGIN TRAN Trans_isolation1
SELECT*FROM T5_Courses WHERE Course_Year=2
```

OUTPUT:



```
USE SCHOOL;
GO
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED
GO
BEGIN TRAN Trans_isolation1
SELECT*FROM T5_Courses WHERE Course_Year=2
```

100 %

Results | Messages

	Course_ID	Course_Name	Course_Year	Student_ID
1	HS202	Economics	2	104
2	HS202	OOPS	2	105
3	HS202	OS	2	107
4	HS202	ToC	2	108
5	HS202	DAA	2	109
6	HS202	ML	2	111
7	HS202	SE	2	115
8	HS202	DBMS	2	116

- When we set the isolation level to read uncommitted, we will be able to see the "Course_ID" set to "HS202", called Dirty Read.