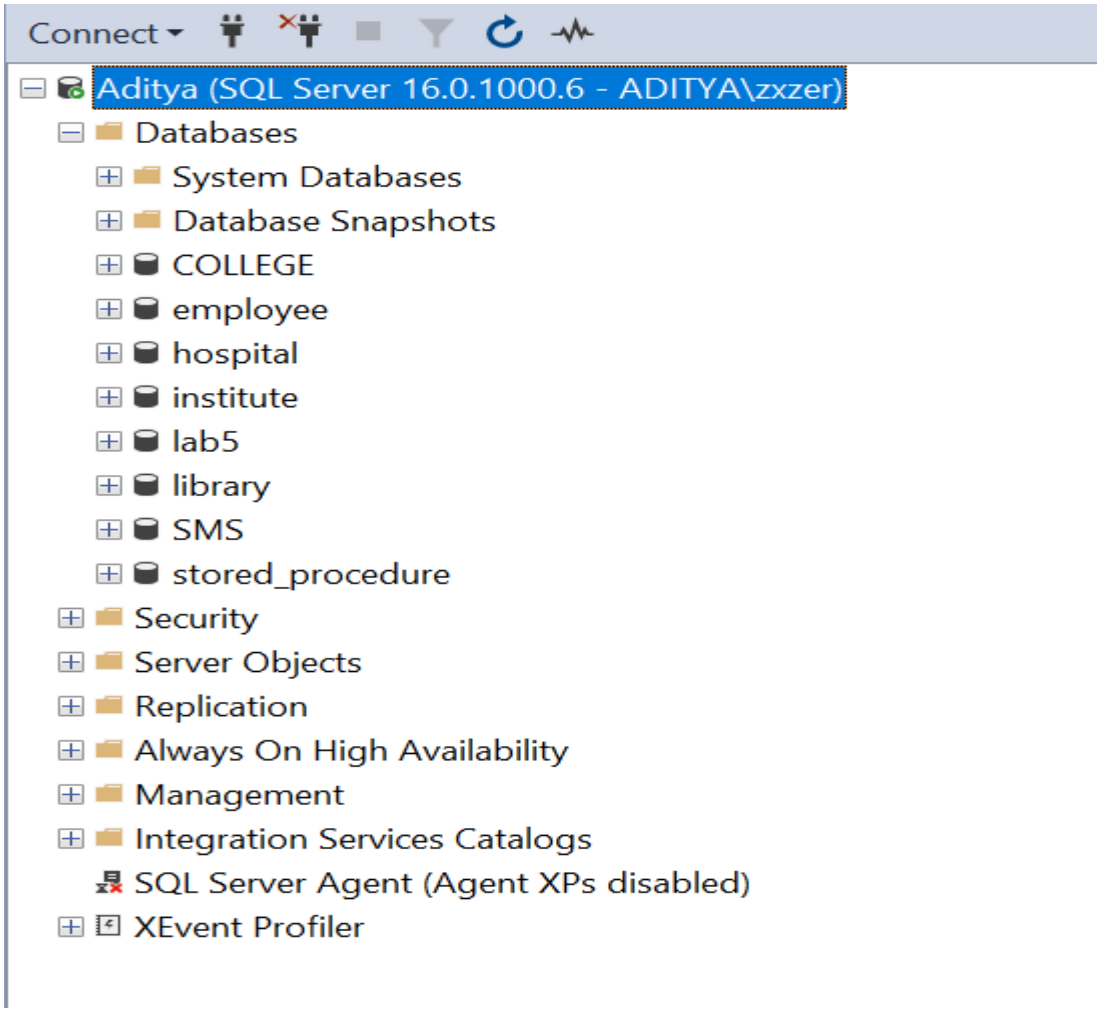
Lab-2:Use of Multiple Table

1. Create Database CMS.

Query: CREATE DATABASE CMS ;



2. Create Multiple Tables : Department, Student, Staff, Subjects and Marks.

Query : CREATE TABLE Department(Did INT NOT NULL identity (1,1) PRIMARY KEY ,
Dname VARCHAR(20),
Db_no INT);

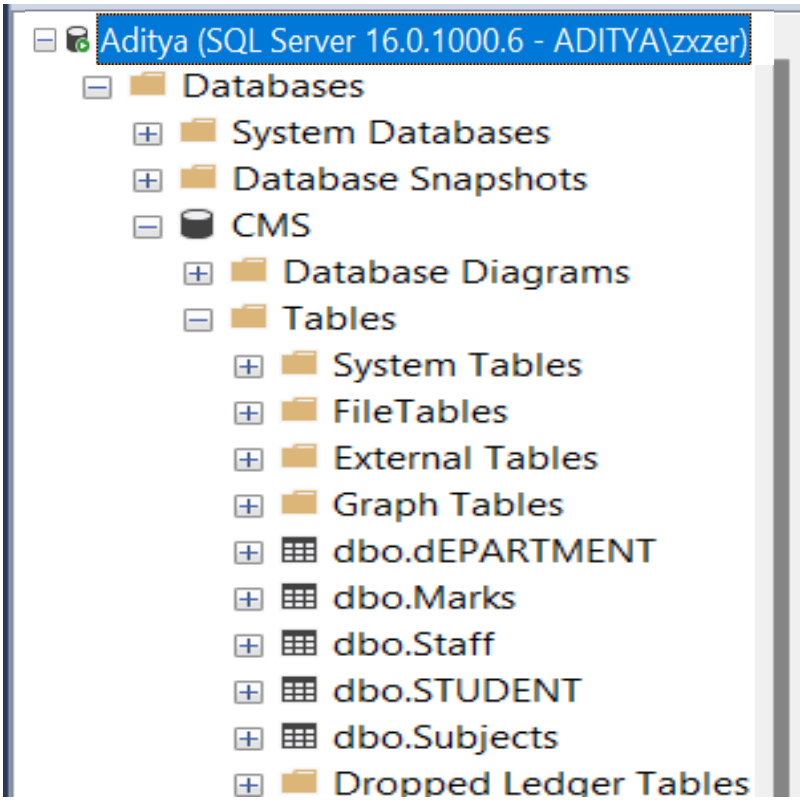
CREATE TABLE Student(Sid INT NOT NULL identity (20,1) PRIMARY KEY ,
Sname VARCHAR(20),

```
                                    Address VARCHAR(20),
                                     Dob DATE ,
                                    Did INT,
                                    FOREIGN KEY (Did) references Department(Did) );


    CREATE TABLE Staff(Staff_id INT NOT NULL identity (40,1)  PRIMARY KEY ,
                                    Staff_name VARCHAR(20),
                                    Did INT,
                                    FOREIGN KEY (Did) references Department(Did) );

CREATE TABLE  Subjects (  Sub_id VARCHAR(10)   PRIMARY KEY ,
                                     Sub_name VARCHAR(20),
                                     Credit_hr INT,
                                    Staff_id INT
                                     FOREIGN KEY (Staff_id) references Staff(Staff_id) );


CREATE TABLE Marks(Obatained_marks INT,
                                   Sub_id VARCHAR(10),
                                    Sid INT,
                                    FOREIGN KEY (Sub_id) references Subjects(Sub_id),
                                    FOREIGN KEY (Sid) references Student(Sid) );
```



3. Insert any 10/10 records to each of the tables through GUI and display.

Query : SELECT * FROM Marks

SELECT * FROM Department

SELECT * FROM Student

SELECT * FROM Staff

SELECT * FROM Subjects

Results Messages

	Obatained_marks	Sub_id	Sid
4	88	CS104	24
5	92	CS105	25
6	76	CS106	26
7	84	CS107	27
8	80	CS108	28
9	75	CS109	29
10	95	CS110	30

	DID	Dname	DB_NO
6	6	Opera...	106
7	7	Legal	107
8	8	Custo...	108
9	9	R&D	109
10	10	Logisti...	110

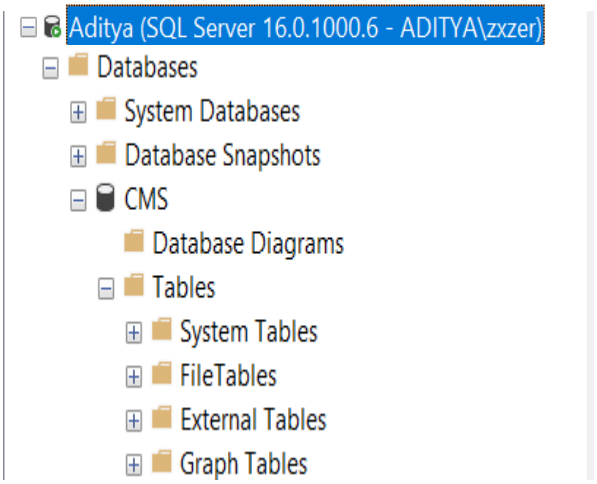
	SID	SNAME	ADDRESS	DOB	DID
6	26	Rita	Bhaktapur	1998-12-12	6
7	27	Mina	Dharan	2000-06-06	7
8	28	Sunita	Hetauda	2002-08-09	8
9	29	Rame...	Janakpur	2003-09-22	9
10	30	Suresh	Nepalgunj	1999-10-30	10

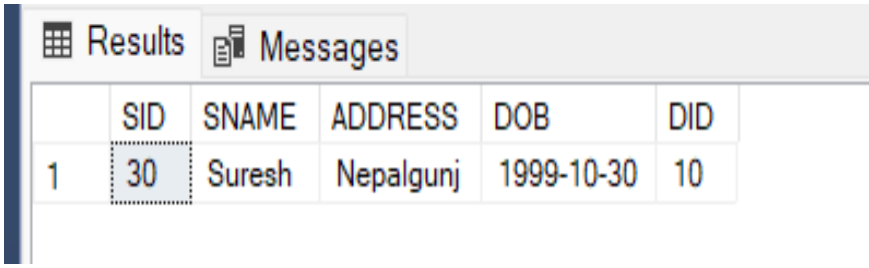
	Staff_id	Staff_name	Did
7	46	Aditi Singh	7
8	47	Sanjay Ya...	8
9	48	Ravi Shres...	9
10	49	Deepa Bh...	10

	Sub_id	Sub_name	Credit_hr	Staff_id
4	CS104	Web	4	43
5	CS105	OS	3	44
6	CS106	Networking	3	45
7	CS107	Software	4	46
8	CS108	AI	3	47
9	CS109	ML	4	48
10	CS110	Cybersec...	3	49

4.Display records of those student who get maximum marks.

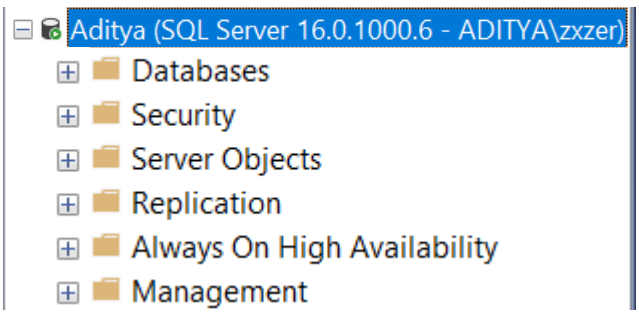
```
Query: SELECT *
       FROM Student AS s
       WHERE s.sid IN (SELECT m.sid
                       FROM Marks AS m
                  WHERE Obatained_marks  In (SELECT    MAX(Obatained_marks)
                                             FROM Marks )    );
```

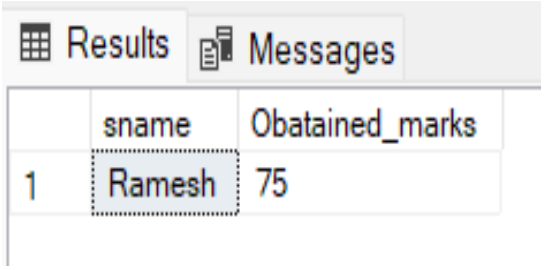




5.Find name and marks of all students who get minimum marks.

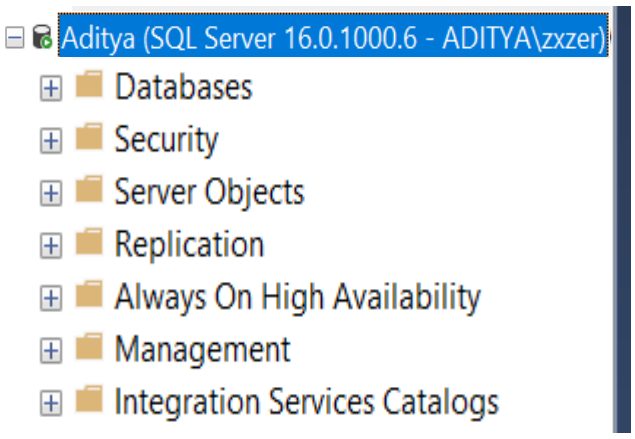
```
Query: SELECT s.sname , m.Obtained_marks
       FROM Student AS s INNER JOIN Marks AS m
       ON s.sid=m.sid
       WHERE Obatained_marks IN (SELECT MIN(Obatained_marks)
                                 FROM Marks);
```





6.Display list of subjects learned by student of dob less than 2045-10-10.

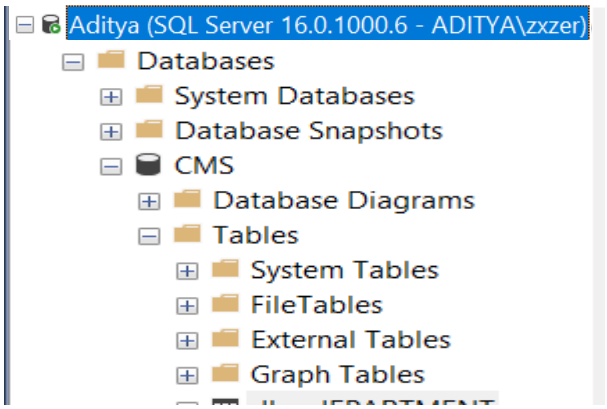
Query: SELECT sub.Sub_id,sub.Sub_name ,sub.credit_hr ,sub.Staff_id , s.sname
FROM Subjects AS sub INNER JOIN Marks AS m
ON sub.sub_id=m.sub_id
INNER JOIN Student AS s
ON s.sid =m.sid
WHERE s.Dob < '2045-10-10' ;



	Sub_id	Sub_name	credit_hr	Staff_id	sname
1	CS101	CS	3	40	Ram
2	CS102	DS	4	41	Shyam
3	CS103	DBMS	3	42	Hari
4	CS104	Web	4	43	Gita
5	CS105	OS	3	44	Sita
6	CS106	Networking	3	45	Rita
7	CS107	Software	4	46	Mina
8	CS108	AI	3	47	Sunita
9	CS109	ML	4	48	Ramesh
10	CS110	Cybersecurity	3	49	Suresh

7.Display name of all students of department 'IT' or of address start with 'k'

Query: SELECT s.sname , s.Address
FROM Department AS d INNER JOIN Student AS s
ON d.Did=s.Did
WHERE s.Address LIKE 'K%' OR d.Dname = 'IT';



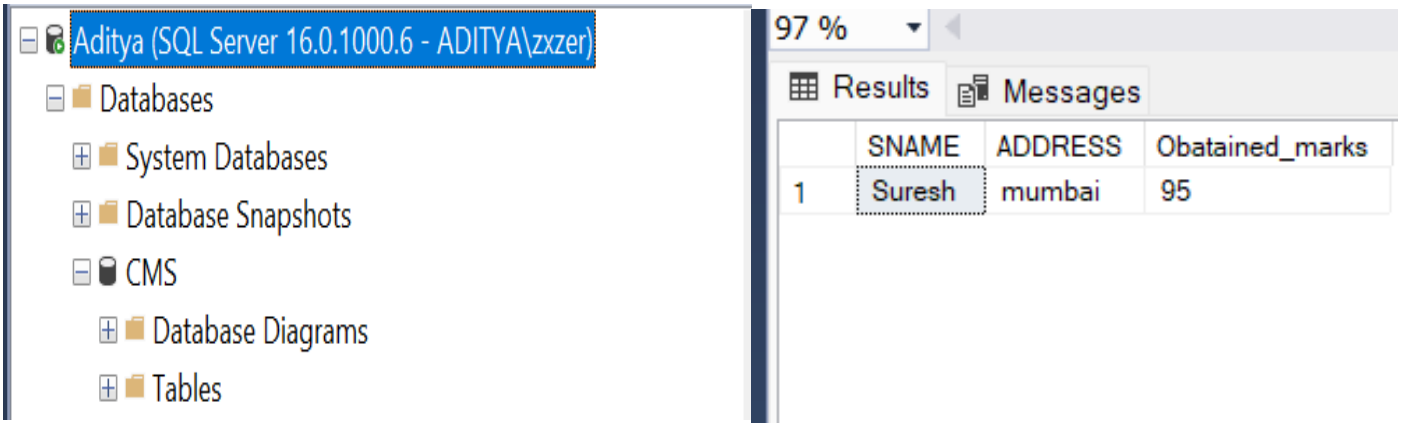
Results Messages

	sname	Address
1	Ram	Kathmandu
2	Kita	Lalitpur

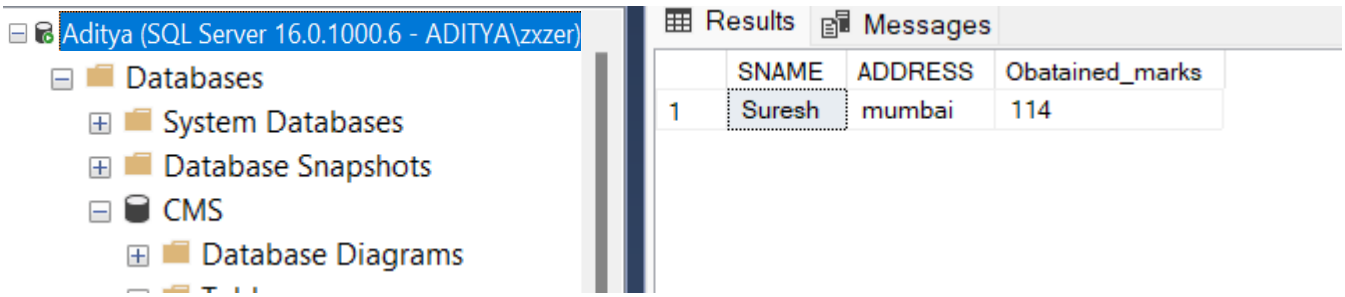
8.Increase marks of all students of address 'Mumbai'by 20%.

Query : SELECT s.Sname, s.Address ,m.Obatained_marks
FROM Student s INNER JOIN Marks AS m
ON s.sid =m.sid
WHERE s.Address ='Mumbai' ;
UPDATE Marks
SET Obatained_marks =Obatained_marks + 0.2 * Obatained_marks
WHERE Sid IN (SELECT sid
FROM student
WHERE Address ='Mumbai');

Before

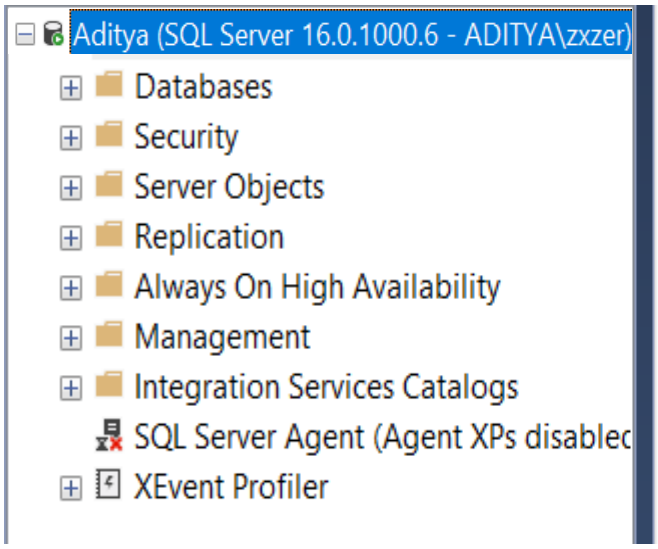


After



9.Display record of all student in descending order of their dob.

Query: SELECT *
FROM Student s
ORDER by s.Dob DESC;

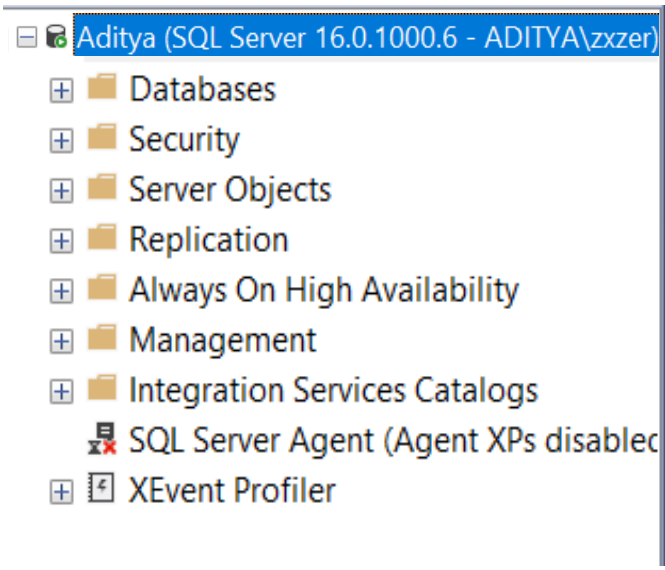


Results Messages

	SID	SNAME	ADDRESS	DOB	DID
1	29	Ramesh	Janakpur	2003-09-22	9
2	28	Sunita	Hetauda	2002-08-09	8
3	24	Gita	Butwal	2002-07-05	4
4	25	Kita	Lalitpur	2001-03-18	5
5	22	Shyam	Pokhara	2001-02-15	2
6	27	Mina	Dharan	2000-06-06	7
7	21	Ram	Kathmandu	2000-01-01	1
8	23	Hari	Biratnagar	1999-11-23	3
9	30	Suresh	mumbai	1999-10-30	10
10	26	Rita	Bhaktapur	1998-12-12	6

10.Display total no of student and their address in every address level.

Query: SELECT COUNT (sid) AS NOS ,Address
FROM Student
GROUP BY (Address);

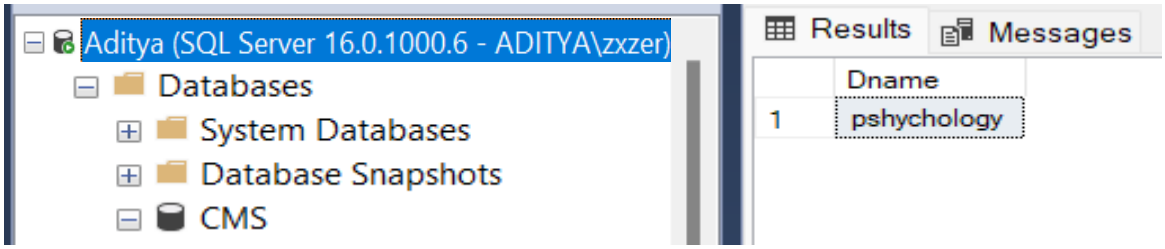


Results Messages

	NOS	Address
1	1	Bhaktapur
2	1	Biratnagar
3	1	Butwal
4	1	Dharan
5	1	Hetauda
6	1	Janakpur
7	1	Kathmandu
8	1	Lalitpur
9	1	mumbai
10	1	Pokhara

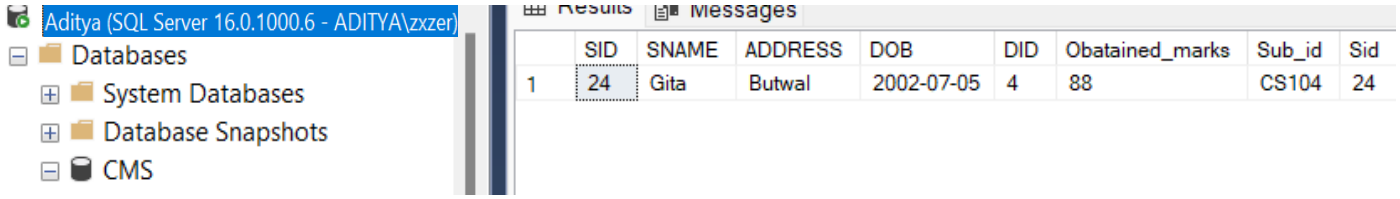
11.Display all department with no students.

Query: SELECT Dname
FROM Student s FULL OUTER JOIN Department d
ON s.did=d.did
WHERE s.did is NULL;



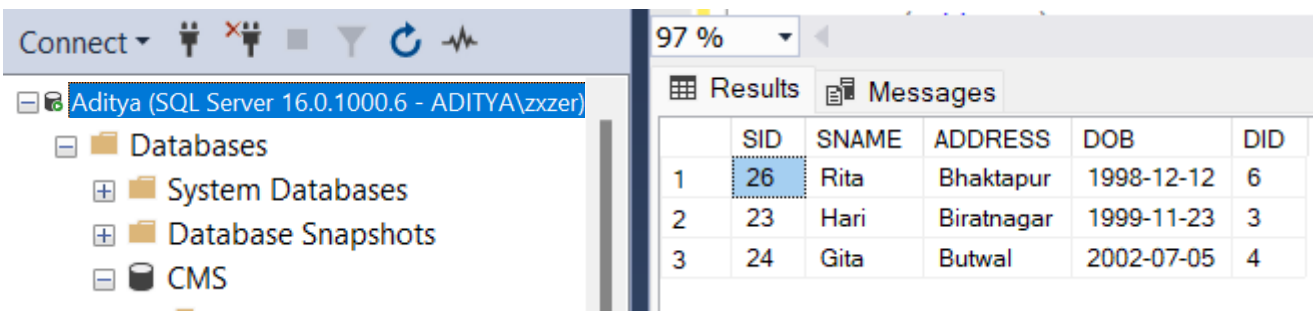
12.Display records of all student of address start with 'B' and get greater or equal than average marks.

Query: SELECT *
FROM Student s INNER JOIN Marks m
ON s.sid=m.sid
WHERE s.Address LIKE 'B%' AND
Obatained_marks >= (SELECT AVG(Obatained_marks)
FROM Marks);



13.Display 5 oldest student of address start with 'B'

Query: SELECT TOP 5*
FROM student s
WHERE s.ADDRESS LIKE 'B%'
ORDER BY Dob ASC ;



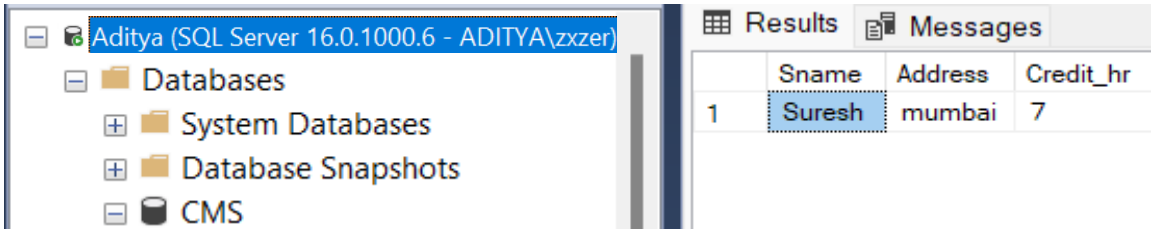
14.Increase the credit hour of all subjects of name contains letter 's' and study by student of address 'Mumbai' by 2 hr.

Query:
```
UPDATE Subjects
SET Credit_hr = Credit_hr +2
FROM Student s INNER JOIN Marks AS m
ON s.sid =m.sid
 INNER JOIN Subjects AS sub
ON sub.Sub_id =m.Sub_id
WHERE sub.Sub_name LIKE '%S%' AND s.Address ='Mumbai' ;

SELECT s.Sname, s.Address,sub.Credit_hr
FROM Student s INNER JOIN Marks AS m
ON s.sid =m.sid
INNER JOIN Subjects AS sub
ON sub.Sub_id =m.Sub_id
WHERE sub.Sub_name LIKE '%S%' AND s.Address ='Mumbai' ;
```

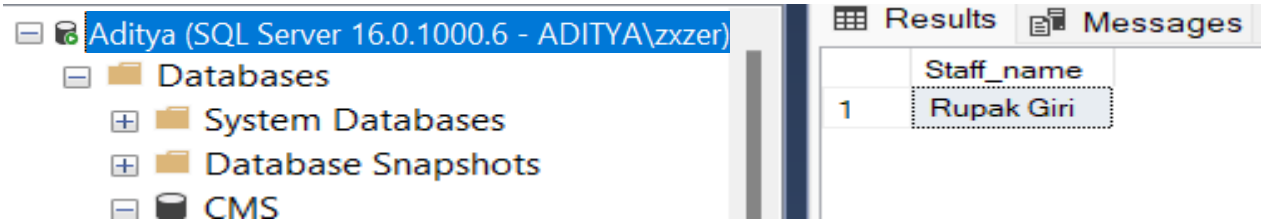
Before

After



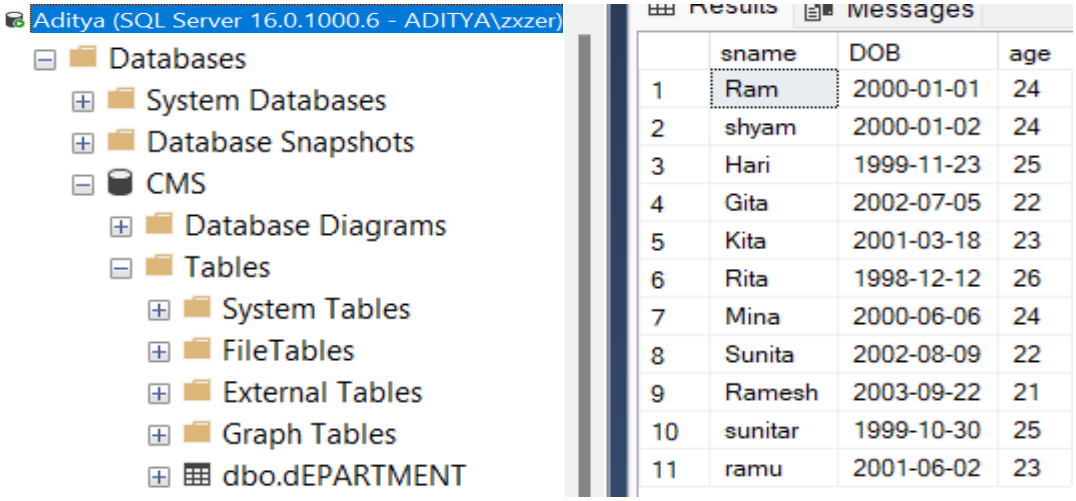
15.Display the record all staffs who not help to any students.

Query: SELECT st.Staff_name
FROM Student s RIGHT JOIN Marks AS m
ON s.sid =m.sid
RIGHT JOIN Subjects AS sub
ON sub.Sub_id =m.Sub_id
Right JOIN Staff AS st
ON st.Staff_id =sub.Staff_id
WHERE sub.Staff_id is NULL ;



16.Find out current age from dob of all students.

Query :SELECT sname, DOB, DATEDIFF (year,DOB,GETDATE()) AS age
FROM Student ;

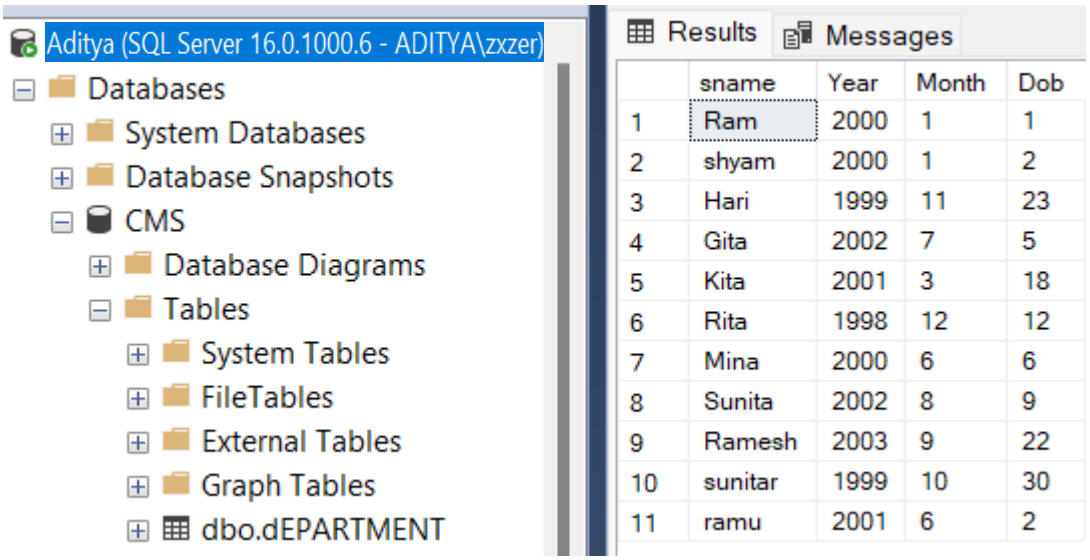


17.Display only those students whose dob contain 2003 yr.

Query: SELECT sname, year (DOB)
FROM Student
WHERE year(DOB) = '2003'

18.Display year, month and day of all students and their names.

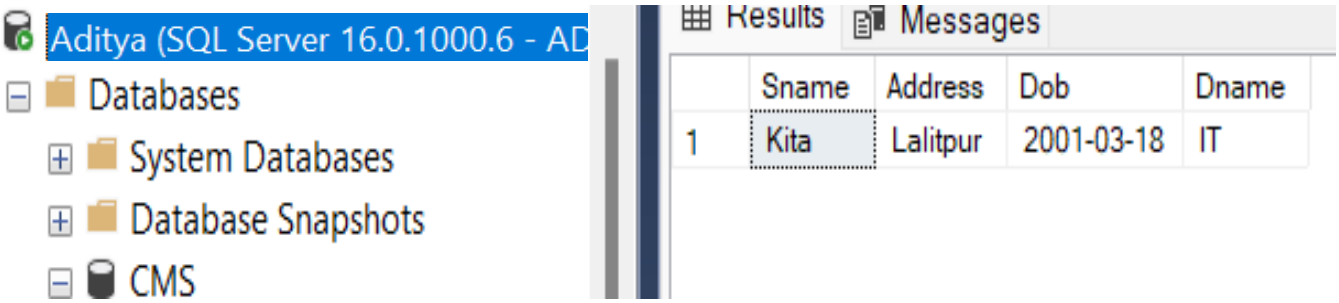
Query : SELECT sname, year(DOB) AS Year ,MONTH(DOB) AS Month ,
DAY(DOB) AS Dob
FROM Student ;



	sname	Year	Month	Dob
1	Ram	2000	1	1
2	shyam	2000	1	2
3	Hari	1999	11	23
4	Gita	2002	7	5
5	Kita	2001	3	18
6	Rita	1998	12	12
7	Mina	2000	6	6
8	Sunita	2002	8	9
9	Ramesh	2003	9	22
10	sunitar	1999	10	30
11	ramu	2001	6	2

19.Display all student who associated with 'IT' department.

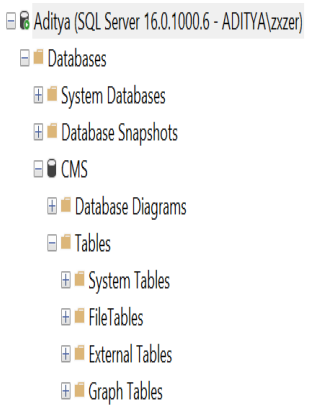
Query : SELECT Sname,Address,Dob,Dname
FROM Department AS d INNER JOIN Student AS s
ON d.Did=s.Did
WHERE d.Dname = 'IT';



	Sname	Address	Dob	Dname
1	Kita	Lalitpur	2001-03-18	IT

20.Find join of above 5-tables.

Query:
```
SELECT *
FROM Student s INNER JOIN Marks AS m
ON s.sid =m.sid
INNER JOIN Subjects AS sub
ON sub.Sub_id =m.Sub_id
INNER JOIN Staff AS st
ON st.Staff_id =sub.Staff_id
INNER JOIN Department AS d
ON d.Did =s.Did ;
```



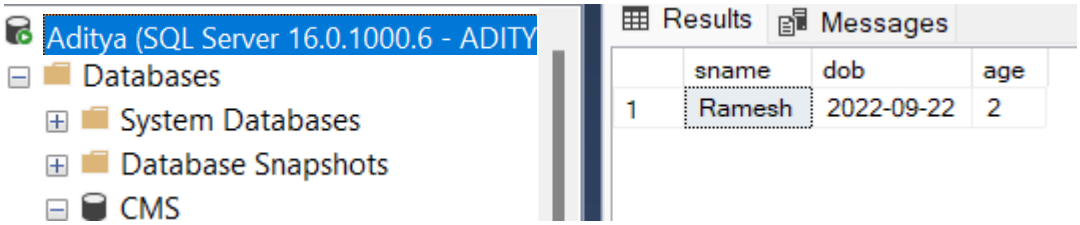
esults Messages

SID	SNAME	ADDRESS	DOB	DID	Obatained_marks	Sub_id	Sid	Sub_id	Sub_name	Credit_hr	Staff_id	Staff_id	Staff_name	Did	DID	Dname	DB_NO
21	Ram	Kathmandu	2000-01-01	1	85	CS101	21	CS101	CS	3	40	40	Sita Sharma	1	1	HR	101
22	shyam	lalitpur	2000-01-02	2	90	CS102	22	CS102	DS	4	41	41	Ram Thapa	2	2	Finance	102
23	Hari	Biratnagar	1999-11-23	3	78	CS103	23	CS103	DBMS	3	42	42	Anita Rai	3	3	Marketing	103
24	Gita	Butwal	2002-07-05	4	88	CS104	24	CS104	Web	4	43	43	Bishal Gurung	4	4	Sales	104
25	Kita	Lalitpur	2001-03-18	5	92	CS105	25	CS105	OS	3	44	44	Pooja Joshi	5	5	IT	105
26	Rita	Bhaktapur	1998-12-12	6	76	CS106	26	CS106	Networking	3	45	45	Manoj Karki	6	6	Operatio...	106
27	Mina	Dharan	2000-06-06	7	84	CS107	27	CS107	Software	4	46	46	Aditi Singh	7	7	Legal	107
28	Sunita	Hetauda	2002-08-09	8	80	CS108	28	CS108	AI	3	47	47	Sanjay Yadav	8	8	Custom...	108
29	Rame...	Janakpur	2003-09-22	9	75	CS109	29	CS109	ML	4	48	48	Ravi Shrestha	9	9	R&D	109
30	sunitar	mumbai	1999-10-30	10	114	CS110	30	CS110	Cybersec...	7	49	49	Deepa Bhatt...	10	10	Logistics	110

Lab-3:Use of View

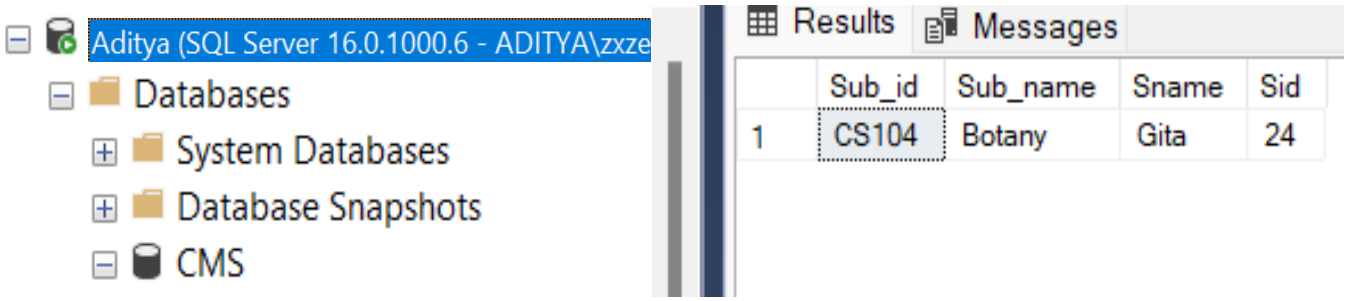
1.Create a view ‘student view’ that display all student of age less than 20.

Query: SELECT sname, dob, DATEDIFF (year, DOB, GETDATE ()) AS age
FROM Student
WHERE DATEDIFF (year, DOB, GETDATE ()) < 20;



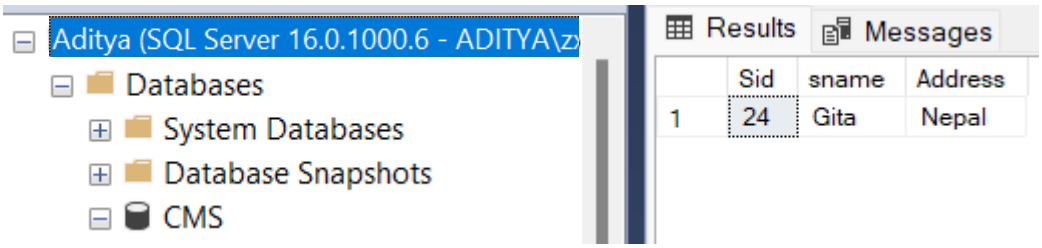
2.Create a view 'Student subjects' that display all student who takes 'BOTANY' subject.

Query : CREATE VIEW student_subjects AS
SELECT sub.Sub_id,sub.Sub_name, s.Sname, s.Sid
FROM Student s INNER JOIN Marks AS m
ON s.sid =m.sid
INNER JOIN Subjects AS sub
ON sub.Sub_id =m.Sub_id
WHERE Sub_name ='BOTANY' ;



3.Create a view 'student details' that contain sid,sname and address of those student of address NEPAL.

Query: CREATE VIEW student_Details AS
SELECT Sid, sname, Address
FROM Student
WHERE address ='Nepal'

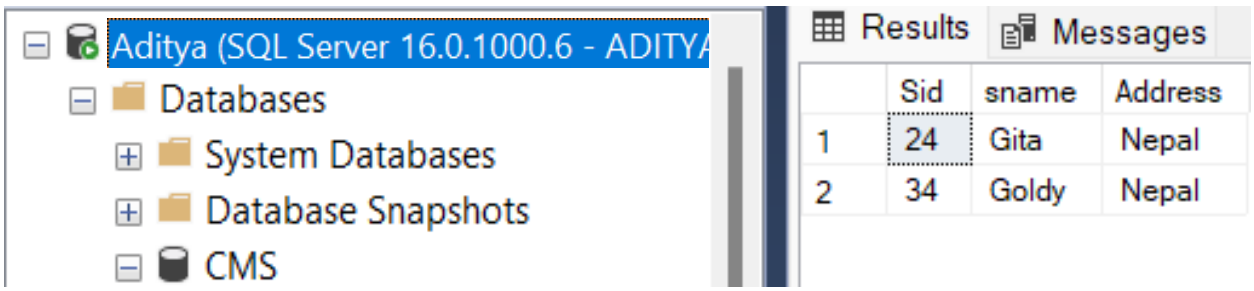


4.Insert any 3 additional records to student_details view.

Query: INSERT into student_Details VALUES('samir', 'kathmandu'),
('Reven', 'Lalitpur'),
('Goldy' , 'Nepal') ;

5.Display the view ‘ student_details’

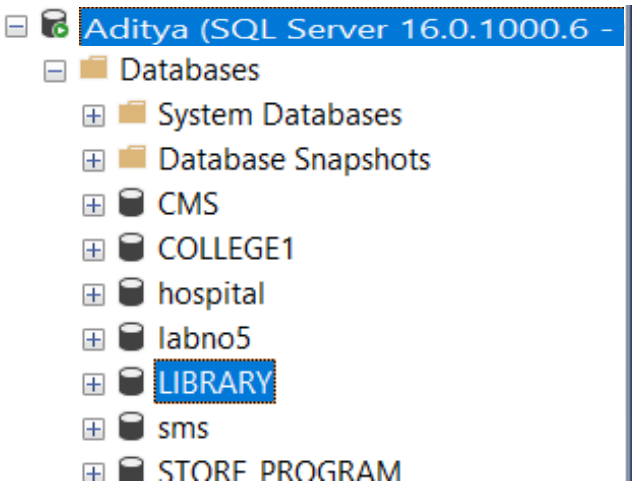
Query: SELECT *
FROM student_Details ;



Lab-4:Use of Constraints

1.Create database name ‘ Library ’.

Query: CREATE DATABASE Library ;

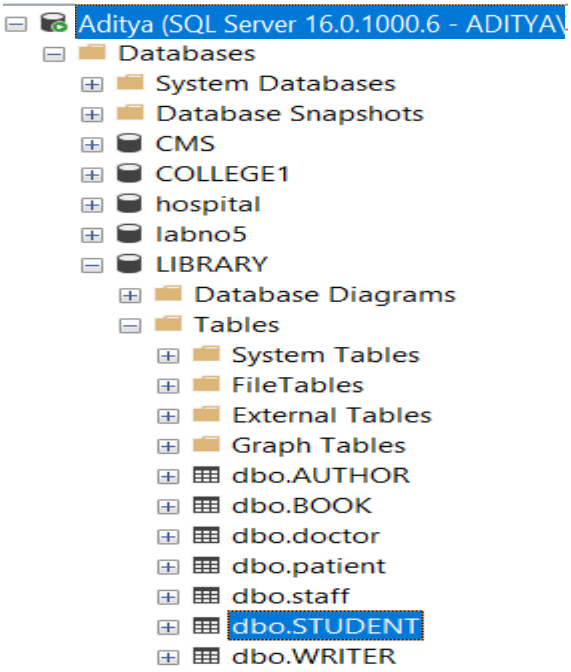


2. Create table Books,Student and Author with proper constraints.

```
Query:CREATE TABLE Book (ISBN VARCHAR(10) PRIMARY KEY,
                         bname VARCHAR (20),
                         price INT NOT NULL,
                         noP INT UNIQUE,
                         sid INT,
                         FOREIGN KEY(sid) references Student (sid));


      CREATE TABLE Student (sid  INT  identity (11,1)  PRIMARY KEY ,
                            sname VARCHAR(20),
                            age INT CHECK(age > 0 and age < 110) ,
                           address VARCHAR(20) DEFAULT 'Baneshwor')


      CREATE TABLE Author(aid  INT identity (21,1)  PRIMARY KEY ,
                          aname VARCHAR(20)  NOT NULL,
                          Phone_no VARCHAR(10) UNIQUE ,
                         address  VARCHAR(20),
                         ISBN VARCHAR(10),
```



FOREIGN KEY (ISBN) references Book (ISBN));

3. Insert any 5 data into table Books,Student and Author.

Query:INSERT into Student (sname,age) VALUES(‘Aditya’,20),
('Lujana',18),
('Ram',18),
('Shyam',19),
(‘Hari’,21) ;

INSERT into Author VALUES('Bhupendra Singh Saud','9818345672','kathmandu','1'),
('Tej Bahadur Shahi','9876543720','Bhaktapur','3'),
('Pearson','9875436271','Lalitpur','4'),
('Indra Chaudhary','9876543768','Illam','2'),
('Dipak Bhatta','9876583456','Pokhara','1');

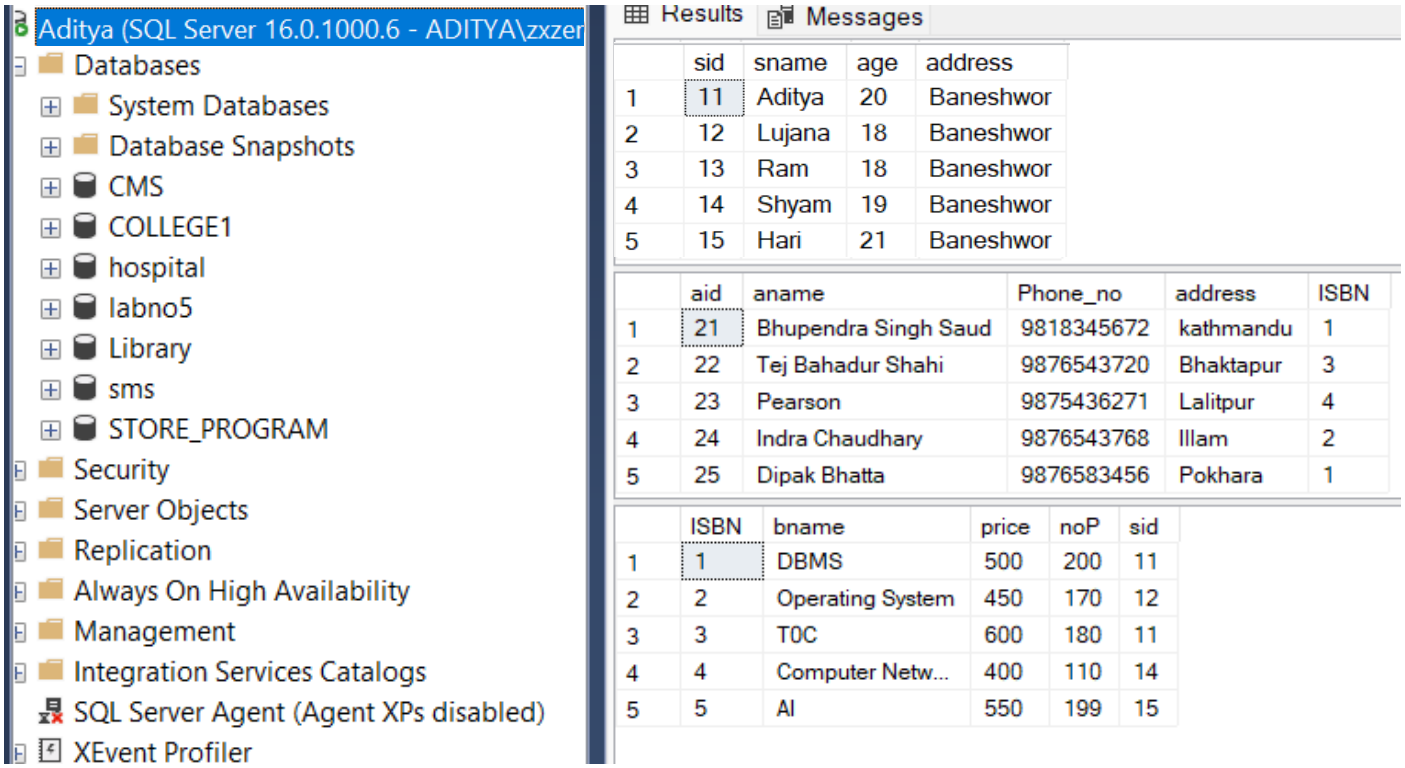
INSERT into Book VALUES ('1','DBMS',500,200,11),
('2','Operating System',450,170,12),
('3','T0C',600,180,11),
('4','Computer Network',400,110,14),
('5','AI',550,199,15);

4. Display table Books,student and Author.

Query: SELECT * FROM student

SELECT * FROM Author

SELECT * FROM Book



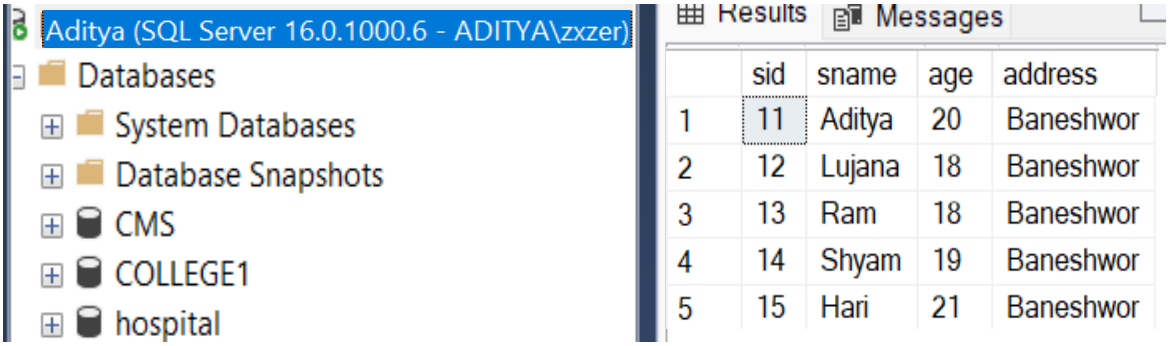
	sid	sname	age	address
1	11	Aditya	20	Baneshwor
2	12	Lujana	18	Baneshwor
3	13	Ram	18	Baneshwor
4	14	Shyam	19	Baneshwor
5	15	Hari	21	Baneshwor

	aid	aname	Phone_no	address	ISBN
1	21	Bhupendra Singh Saud	9818345672	kathmandu	1
2	22	Tej Bahadur Shahi	9876543720	Bhaktapur	3
3	23	Pearson	9875436271	Lalitpur	4
4	24	Indra Chaudhary	9876543768	Illam	2
5	25	Dipak Bhatta	9876583456	Pokhara	1

	ISBN	bname	price	noP	sid
1	1	DBMS	500	200	11
2	2	Operating System	450	170	12
3	3	TOC	600	180	11
4	4	Computer Netw...	400	110	14
5	5	AI	550	199	15

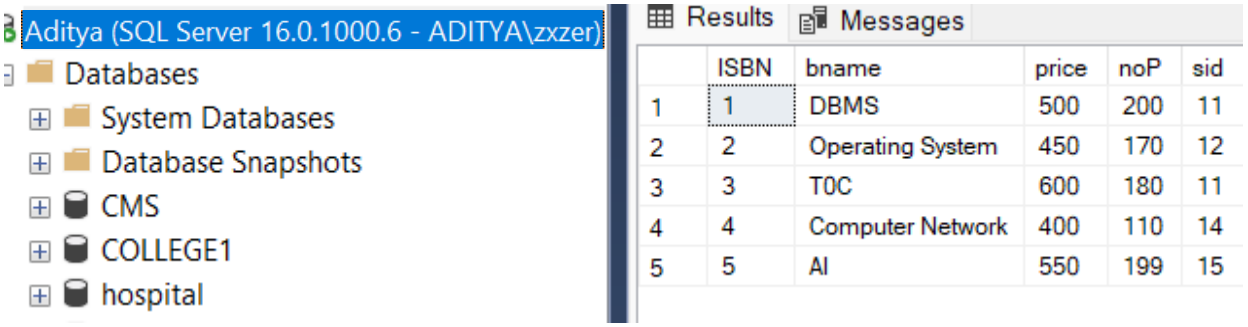
5. Test the ‘default’ constraints.

Query: address VARCHAR(20) DEFAULT 'Baneshwor' (In table : Student)



	sid	sname	age	address
1	11	Aditya	20	Baneshwor
2	12	Lujana	18	Baneshwor
3	13	Ram	18	Baneshwor
4	14	Shyam	19	Baneshwor
5	15	Hari	21	Baneshwor

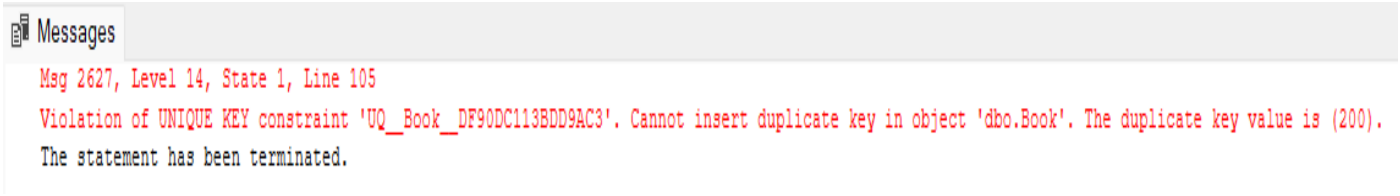
6. Test for ‘Unique’ constraint.



	ISBN	bname	price	noP	sid
1	1	DBMS	500	200	11
2	2	Operating System	450	170	12
3	3	T0C	600	180	11
4	4	Computer Network	400	110	14
5	5	AI	550	199	15

Query: noP INT UNIQUE (In table: Book)

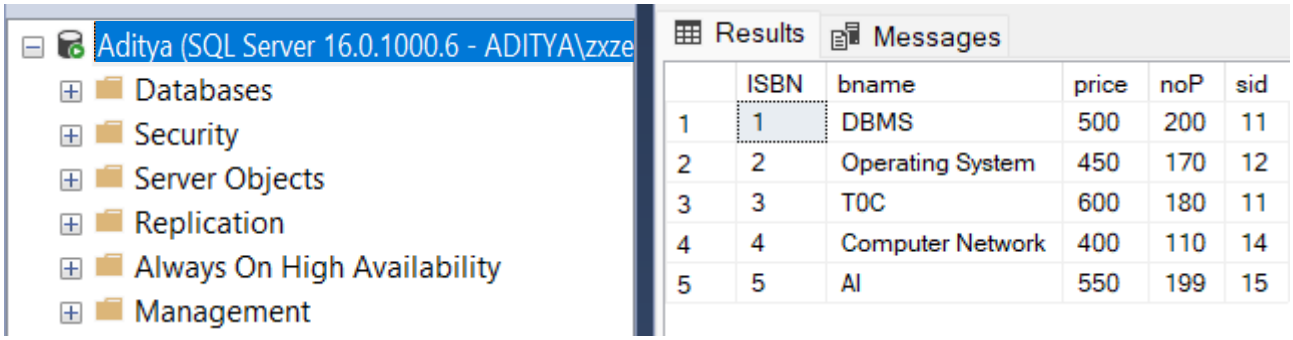
Query: INSERT into Book VALUES (‘6','Dsa',700,200,15);



Here, we are unable to insert data as ‘noP’ should be unique i.e 200 is already in table.

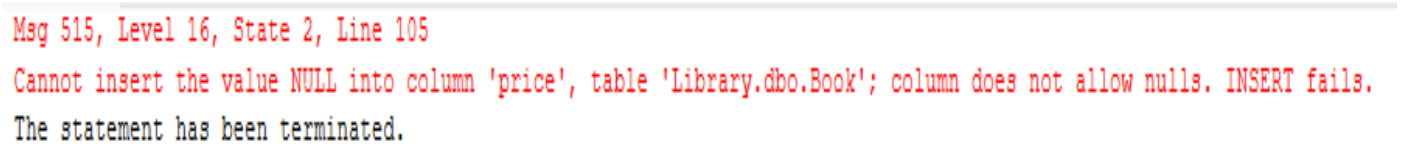
7. Test for ‘Not Null ’ constraint.

Query: price INT NOT NULL, (In table: Book)



	ISBN	bname	price	noP	sid
1	1	DBMS	500	200	11
2	2	Operating System	450	170	12
3	3	T0C	600	180	11
4	4	Computer Network	400	110	14
5	5	AI	550	199	15

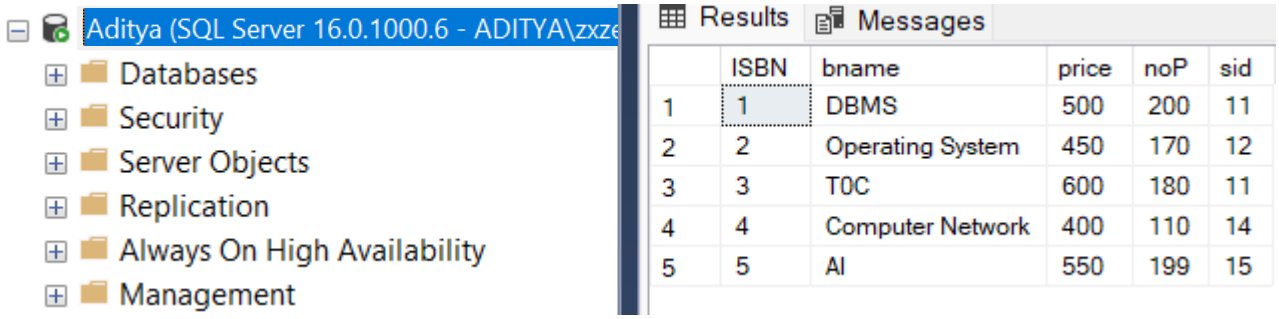
Query: INSERT into Book VALUES ('6','Dsa',NULL,230,15);



Here ,we are unable to insert data as ‘price’ should not be NUll.

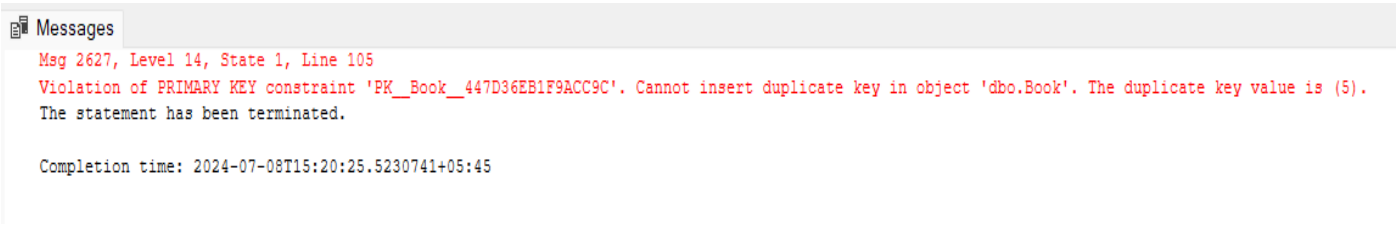
8. Test for ‘Primary Key’ constraint.

Query: ISBN VARCHAR (10) PRIMARY KEY



	ISBN	bname	price	noP	sid
1	1	DBMS	500	200	11
2	2	Operating System	450	170	12
3	3	T0C	600	180	11
4	4	Computer Network	400	110	14
5	5	AI	550	199	15

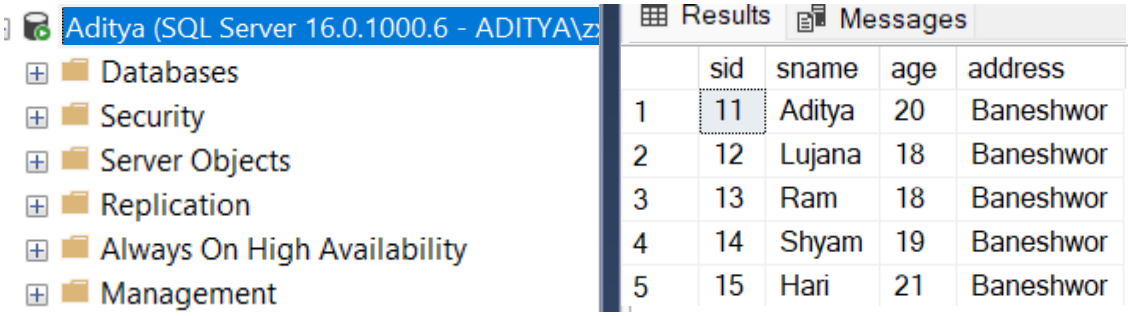
Query: INSERT into Book VALUES ('5','Dsa',240,230,15)



Here , we are unable to insert data as Primary Key can’t be same that is 5 is already in table.

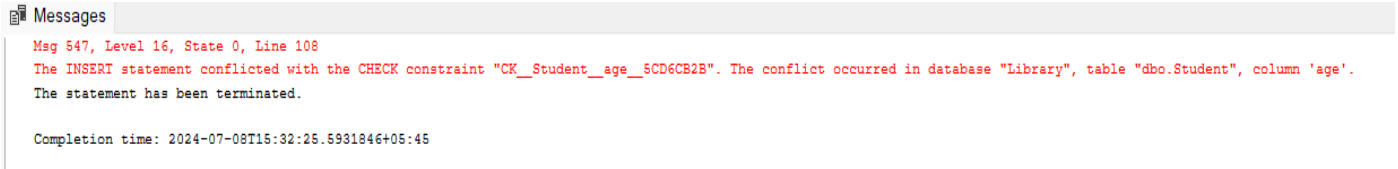
9. Test for ‘Check’ constraint.

Query: age INT CHECK(age > 0 and age < 110) (In table: Student)



	sid	sname	age	address
1	11	Aditya	20	Baneshwor
2	12	Lujana	18	Baneshwor
3	13	Ram	18	Baneshwor
4	14	Shyam	19	Baneshwor
5	15	Hari	21	Baneshwor

Query: INSERT into Student(sname,age) VALUES ('Ram',-20);

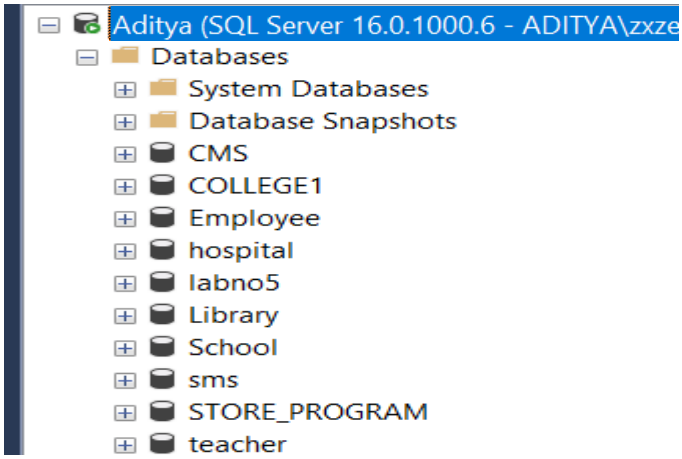


Here, we are unable to insert data with age -20 as age must be 0 < age < 110 .

Lab-5:Use of Triggers

1.Create database name ‘teacher’

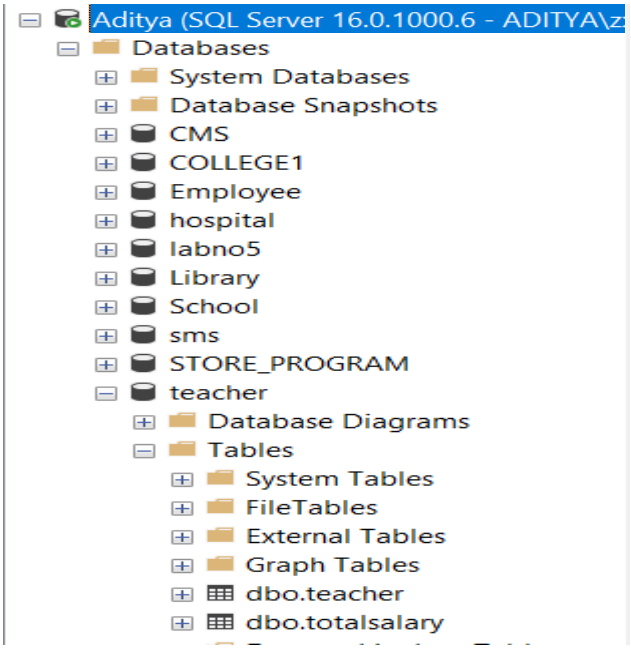
Query: CREATE DATABASE teacher;



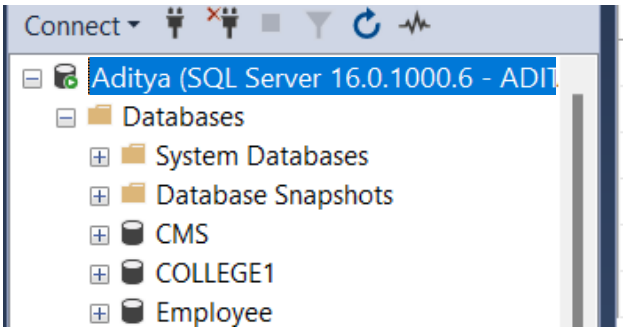
2.Create table Teacher and Total_salary with proper constraints.

Query: create table teacher
(
tid varchar(5) primary key,
tname varchar(20),
salary float,
address varchar(20)
)

create table totalsalary
(
salary_sum float
)



3.Insert any 5 data into Employee table through GUI.



	tid	tname	salary	address
	1	Ram	50	Sanepa
	2	Shyam	50	Banepa
	3	Rami	500	Santinagar
	4	Kamal	1500	Bhaktapur
	5	Kamala	100	Rara
▶*	NULL	NULL	NULL	NULL

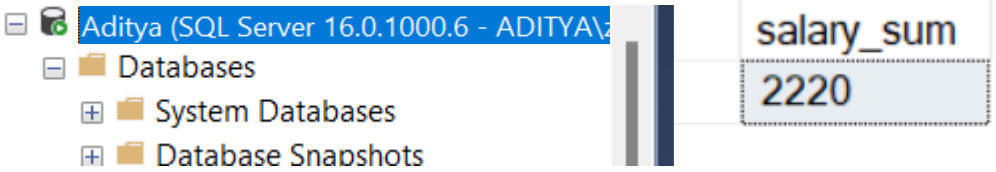
4.Create a trigger to find total sum of salary and store to total_salary table.

Query:

```
create trigger auto_sum on teacher after insert,delete,update as
begin declare @total float select @total = sum(salary) from
teacher; update totalsalary set salary_sum = @total; end
```

5.Display total_salary table after activation of trigger total_salary_update.

Query: SELECT*
FROM TOTAL_SALARY



6. create trigger Employee_log_update .

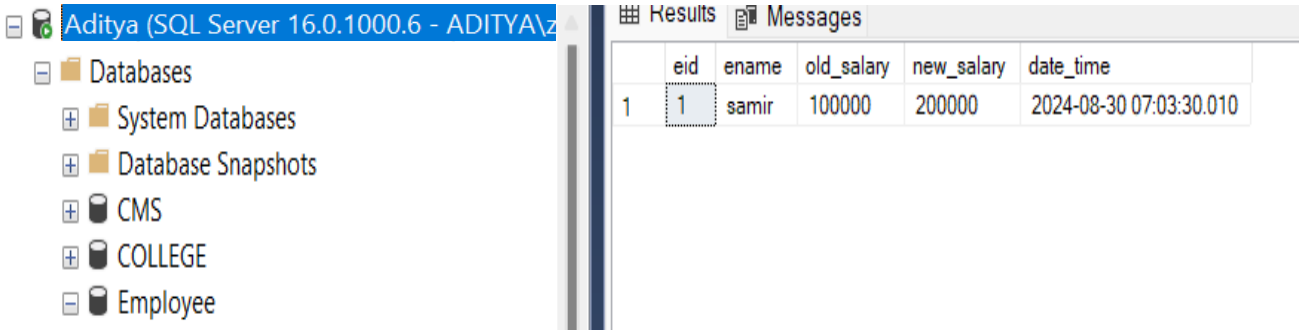
Query:CREATE trigger Employee_Log_update
ON Employee
AFTER UPDATE
AS
BEGIN
Insert into Employee_log(eid,ename,old_Salary,new_salary,date_time)

SELECT deleted.eid,deleted.ename,deleted.salary AS
old_salary,inserted.salary AS new_salary,GETDATE()
FROM inserted

JOIN deleted
on inserted.eid=deleted.eid
END ;

7.Display Employee_log table after activation of trigger Employee_Log_Update.

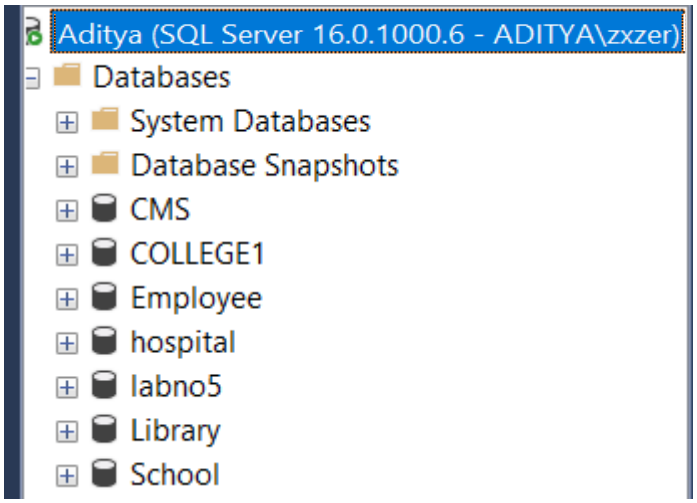
Query: SELECT*
FROM Employee_log



Lab-6:Use of Store Procedure

1.Create database name ‘School’

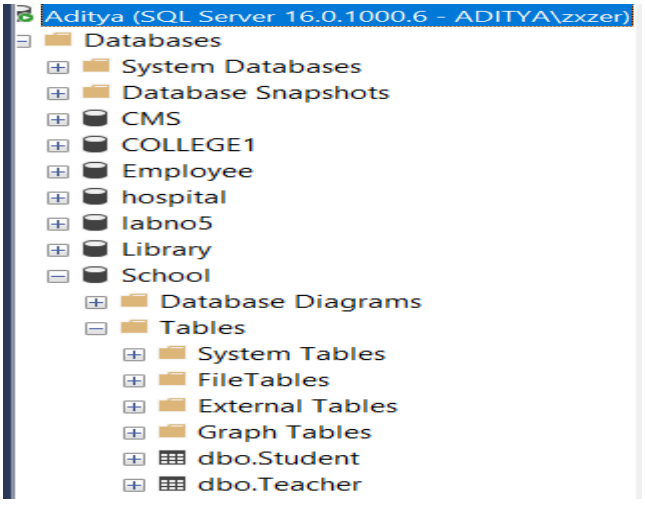
Query: CREATE DATABASE School;



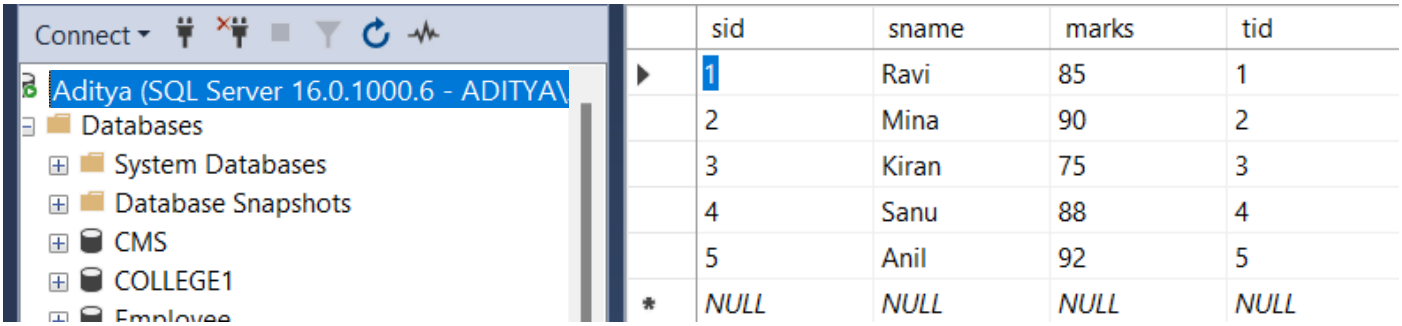
2.Create table Teacher and Student with proper constraints.

Query:CREATE TABLE Teacher
(tid INT NOT NULL PRIMARY KEY ,
tname VARCHAR(20),
salary FLOAT,
Address VARCHAR(20));

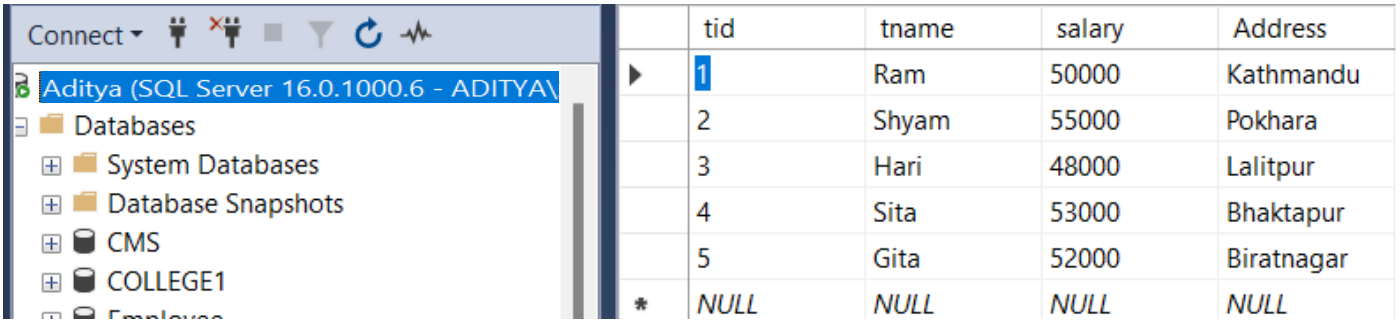
CREATE TABLE Student
(sid INT,
sname VARCHAR(20),
marks FLOAT,
tid INT,
FOREIGN KEY (tid) references Teacher(tid));



3.Insert any 5 data into Teacher and Student table through GUI.



	sid	sname	marks	tid
▶	1	Ravi	85	1
	2	Mina	90	2
	3	Kiran	75	3
	4	Sanu	88	4
	5	Anil	92	5
*	NULL	NULL	NULL	NULL



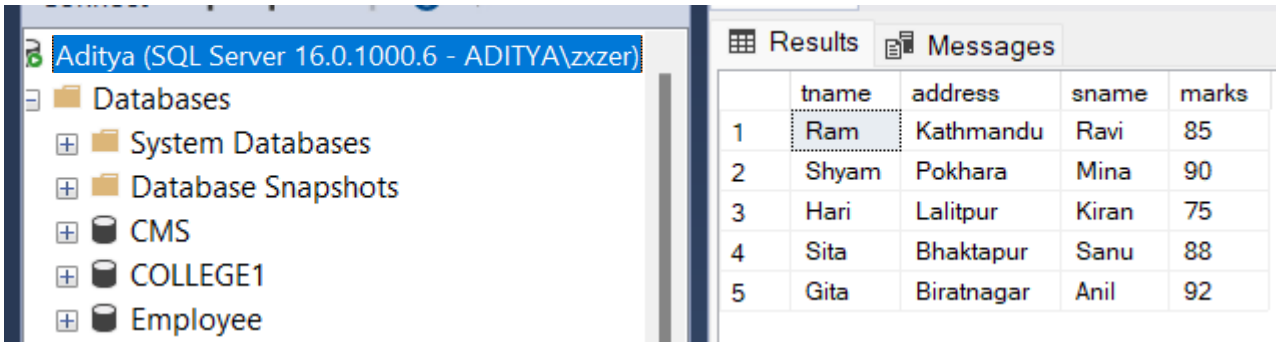
	tid	tname	salary	Address
▶	1	Ram	50000	Kathmandu
	2	Shyam	55000	Pokhara
	3	Hari	48000	Lalitpur
	4	Sita	53000	Bhaktapur
	5	Gita	52000	Biratnagar
*	NULL	NULL	NULL	NULL

Teacher and Student tables based on their tid (teacher ID).

Query: CREATE procedure teacher_student
AS
BEGIN
SELECT tname,address,sname,marks
FROM Teacher t inner join Student s
on t.tid = s.tid
END

5.Display procedure teacher_student.

Query:EXEC teacher_student;



	tname	address	sname	marks
1	Ram	Kathmandu	Ravi	85
2	Shyam	Pokhara	Mina	90
3	Hari	Lalitpur	Kiran	75
4	Sita	Bhaktapur	Sanu	88
5	Gita	Biratnagar	Anil	92