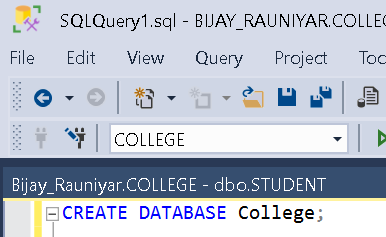
**Create Database :**

Query: CREATE DATABASE College;



**Creating Table:**

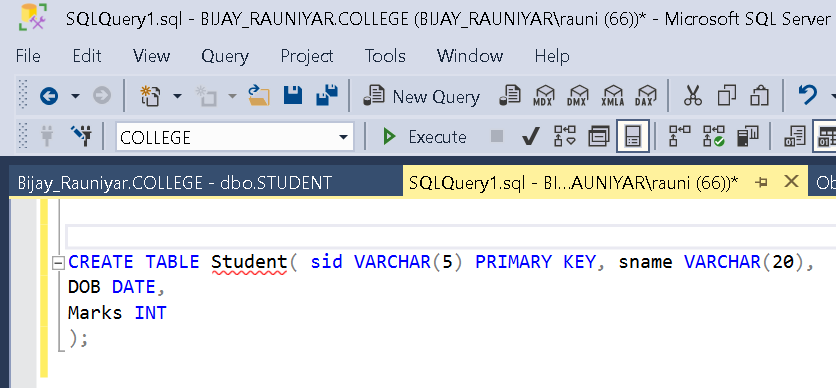
Query:

CREATE TABLE Student( sid VARCHAR(5) PRIMARY KEY, sname VARCHAR(20),

DOB DATE,

Marks INT

# );



**Inserting into Table:**

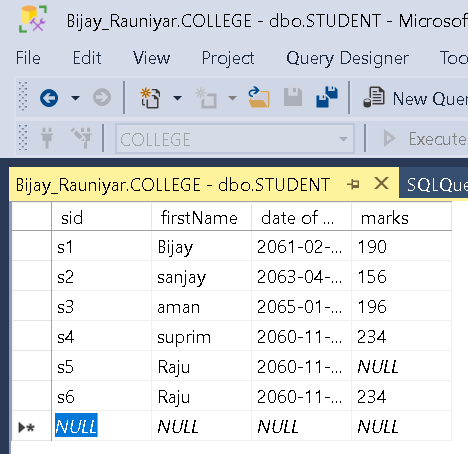
Query: INSERT INTO Student

VALUES ('s1','Bijay','2061-02-28', 190),

('s2', 'sanjay','2063-04-22', 156),

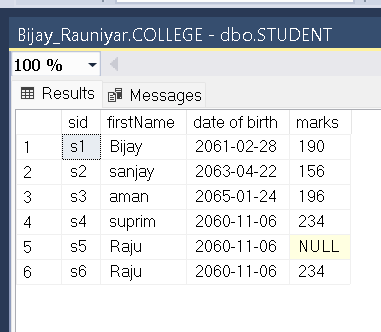
('s3','aman', '2065-01-24',1196);

**Inserting Data Through GUI:**



**Displaying all data from Table:**

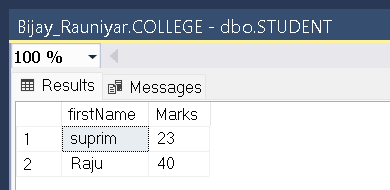
Query: SELECT \* FROM Student;



**Displaying Selected Data From Table:**

Query: SELECT firstname,Marks FROM Student

WHERE Marks < 100;

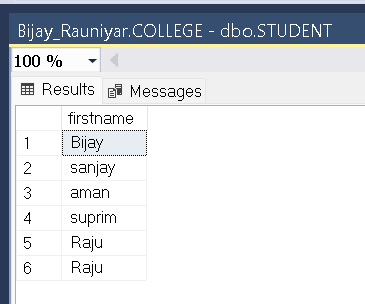


**Display name of all students of marks not equal to 200 or of dob less than ‘2070-01-01’.**

Query: SELECT firstname

FROM student

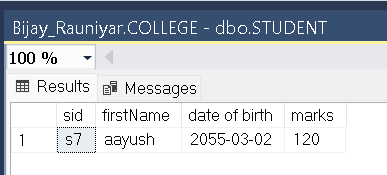
WHERE marks < > 200 OR [date of birth] < '2070-01-01';



**Display all students of marks equal to 120 or 200 or 150 or 220 or 199.**

Query: SELECT \* FROM student

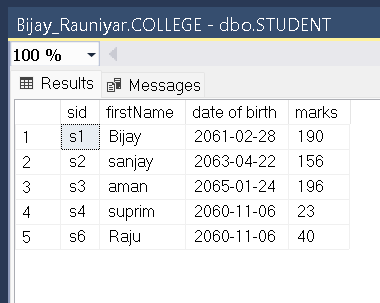
WHERE marks IN(120,200,150,220,199);



**9. Display all students of marks not equal to 120 or 200 or 150 or 220 or 199.**

Query : SELECT \* FROM student

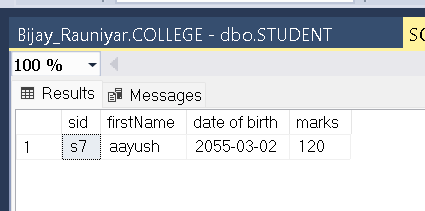
WHERE marks NOT IN(120,200,150,220,199);



**Display all students of dob between ‘2050-01-01’ and ‘2060-01-01’ .**

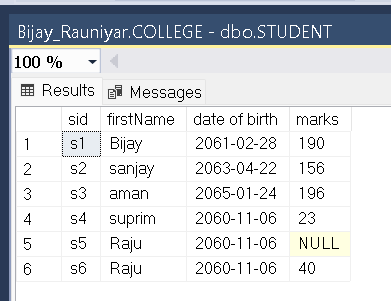
Query : SELECT \* FROM student

WHERE marks NOT IN(120,200,150,220,199);



**Display all students of dob not between ‘2050-01-01’ and ‘2060-01-01’**

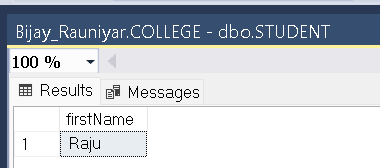
Query : SELECT \* FROM student

WHERE [date of birth] NOT BETWEEN '2050-01-01' AND '2060-01-01'; 

**Display name of all students whose marks is Null and Not Null** Query :

For marks is Null-

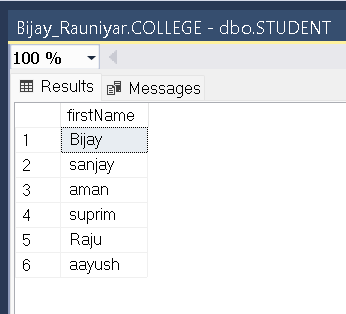
SELECT firstName FROM student WHERE marks IS NULL ;



For marks is NOT NULL:

SELECT firstName FROM student

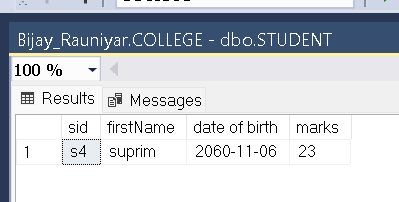
# WHERE marks IS NOT NULL;



**Display records of all students whose name contains ‘m’ as substring and dob is less than ‘2065-01-05’.**

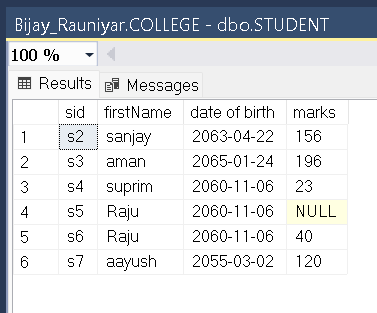
**Query :** SELECT \* FROM Student

WHERE firstName LIKE '%m%' AND [date of birth] < '2065-01-05'

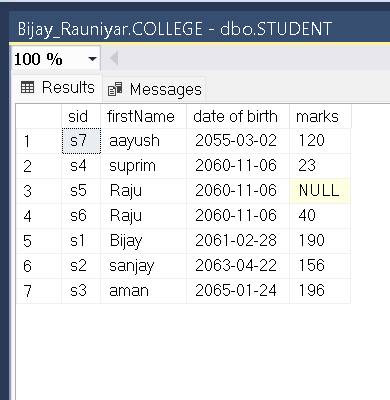


**Display records of all students whose name length is not equal to 5.**

Query :



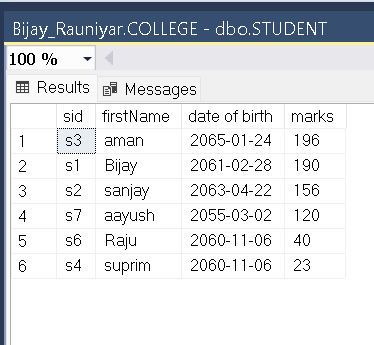
**Display records of all students in ascending order of their dob.**

Query: SELECT \* FROM student ORDER by [date of birth] ASC; 

**Display records of all students of marks less than than 200 and arrange the data in descending order of their marks.**

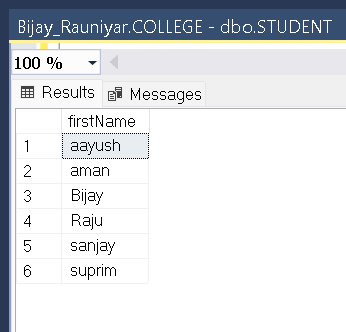
Query: SELECT \* FROM STUDENT WHERE marks < 200

ORDER by marks DESC;



**Display records of all students by displaying unique names.**

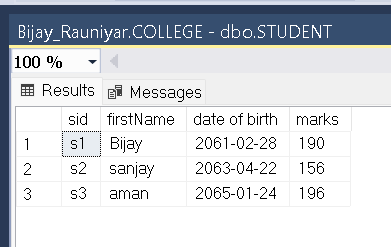
Query: SELECT DISTINCT (firstName) FROM student;



**Display top 3 records of student.**

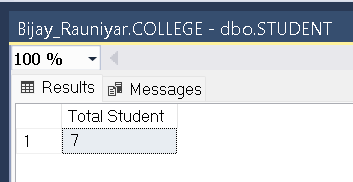
Query: Select TOP(3) \*

From student ;



**Find total no of students.**

Query: SELECT COUNT(\*) AS 'Total Student' FROM student ;

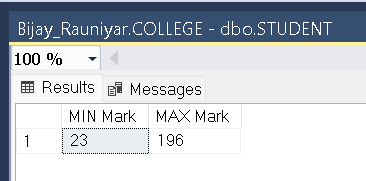


**Find maximum and minimum marks of students.**

Query : SELECT MIN(marks) AS 'MIN Mark',

MAX(marks) AS 'MAX Mark'

FROM student ;



|  |  |  |
| --- | --- | --- |
| **LAB 1 – SINGLE TABLE -DBMS** |  | **Bijay Rauniyar** |

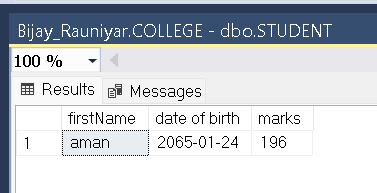
**Find name and dob of those student who get maximum marks.**

Query: SELECT firstName, [date of birth], marks

FROM student

WHERE marks IN (SELECT MAX(marks)

FROM student ) ;

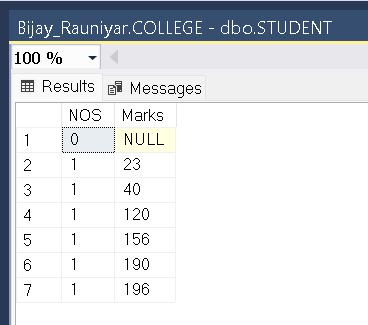


**Display no of student with same Marks.**

Query: SELECT COUNT(Marks) as 'NOS' ,Marks

FROM student

GROUP BY (Marks);

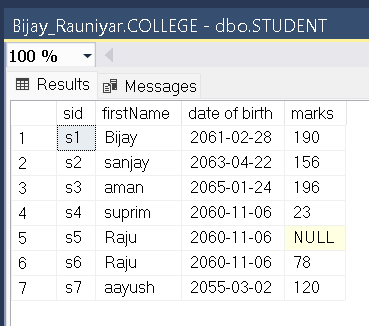


**Increase marks of all student by 40% of name start with 's'**

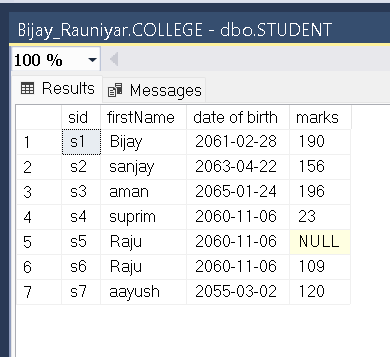
Query : UPDATE student

SET marks = marks + marks \*0.4 WHERE firstName LIKE 'R%' ;

**Before :**

****

**AFTER:**

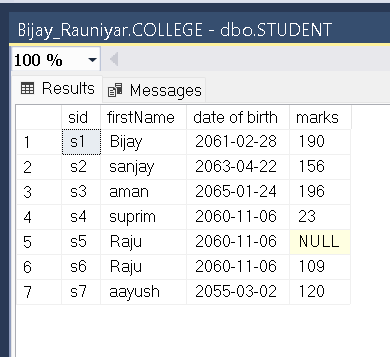
****

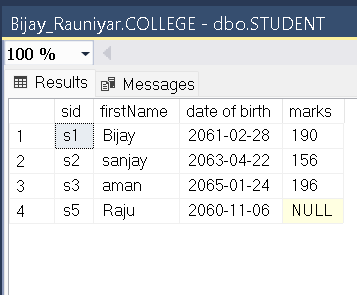
**Delete record of all student marks with less than 150.**

Query: DELETE FROM student

WHERE (marks) < 150 ;

**Before :**

** AFTER:**

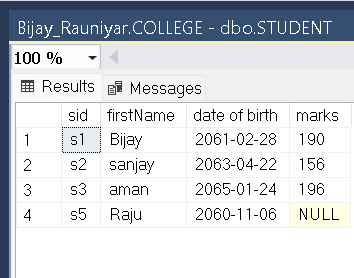
****

**Insert new attribute address to the student table.**

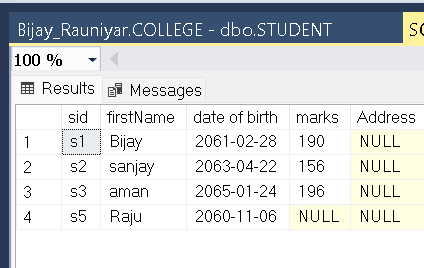
Query: ALTER TABLE student

ADD Address VARCHAR(20);

**Before :**



**After:**

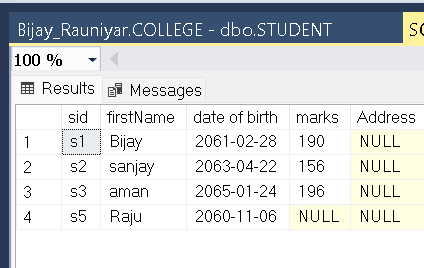


**Remove address attribute from student table.**

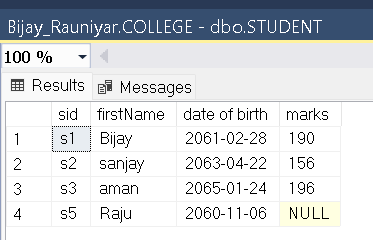
Query: ALTER TABLE student

DROP COLUMN Address;

**Before :**



**After:**

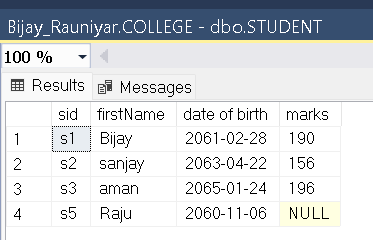


**1 – Single Table -DBMS**

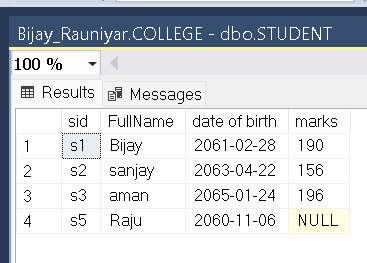
**Rename the attribute to FullName'.**

Query: EXEC sp\_rename 'Student.sname' ,'FullName'

**Before :**

****

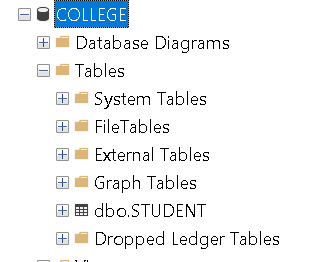
**After:**



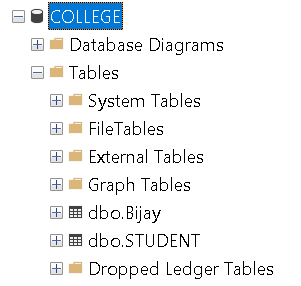
**1 – Single Table -DBMS**

**Delete table ‘Bijay’ from database**

Query : Drop table **Aadarsh’**

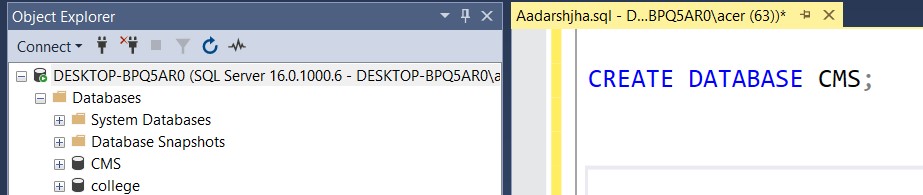
**Before : **

**AFTER:**

****

1 **2- Multiple Table – DBMS**

**Create Database CMS :-**



**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,

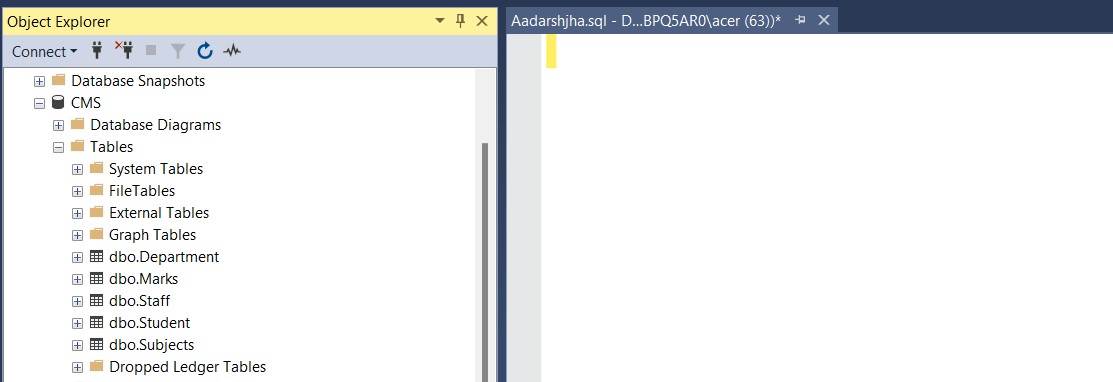
Sub\_id VARCHAR(10),

Sid INT,

FOREIGN KEY (Sub\_id) references Subjects(Sub\_id),

FOREIGN KEY (Sid) references Student(Sid)

);



**Insert records to each of the tables** Query :

INSERT INTO Department (Dname, Db\_no) VALUES

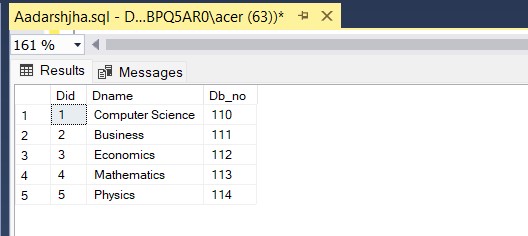
('Computer Science', 110),

('Business', 111),

('Economics', 112),

('Mathematics', 113),

('Physics', 114);



INSERT INTO Student (Sname, Address, Dob, Did) VALUES

('Rita Adhikari', 'Birgunj', '1997-08-15', 2),

('Kavi Karna', 'Gorkha', '1996-03-12', 1),

('Susmita Giri', 'Biratnagar', '2011-05-05', 3),

('Suman Khadka', 'Kathmandu', '1997-11-30', 4),

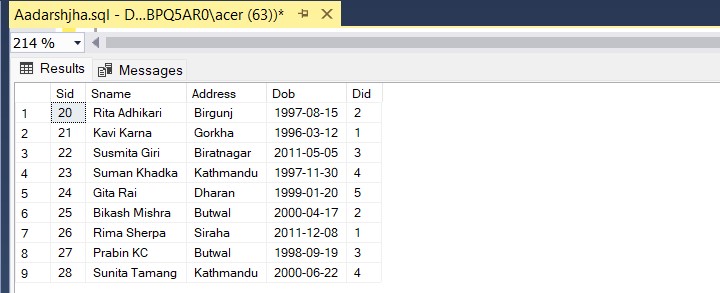
('Gita Rai', 'Dharan', '1999-01-20', 5),

('Bikash Mishra', 'Butwal', '2000-04-17', 2),

('Rima Sherpa', 'Siraha', '2011-12-08', 1),

('Prabin KC', 'Butwal', '1998-09-19', 3),

('Sunita Tamang', 'Kathmandu', '2000-06-22', 4);



INSERT INTO Staff (Staff\_name, Did) VALUES

('Ramesh Sarki', 1),

('Tina Lama', 2),

('Kiran Lama', 3),

(' Sudha Khan', 4),

('Anil Mishra', 5),

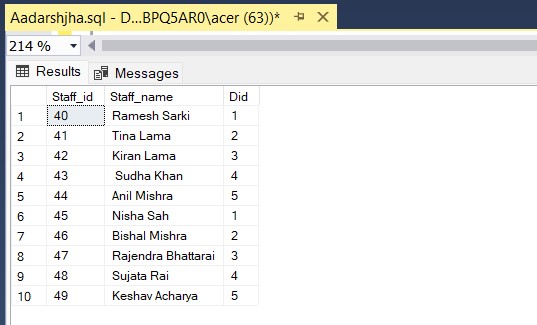
('Nisha Sah', 1),

('Bishal Mishra', 2),

('Rajendra Bhattarai', 3),

('Sujata Rai', 4),

('Keshav Acharya', 5);



INSERT INTO Subjects (Sub\_id, Sub\_name, Credit\_hr, Staff\_id) VALUES

('BUS201', 'Marketing', 3, 41),

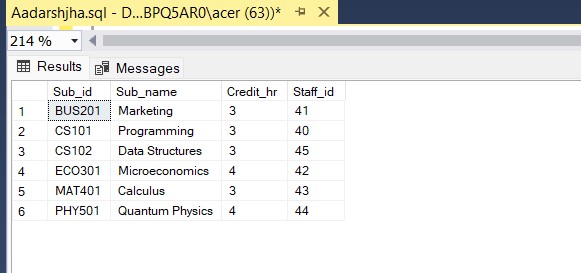
('CS101', 'Programming', 3, 40),

('CS102', 'Data Structures', 3, 45),

('ECO301', 'Microeconomics', 4, 42),

('MAT401', 'Calculus', 3, 43),

('PHY501', 'Quantum Physics', 4, 44);



INSERT INTO Marks (Obatained\_marks, Sub\_id, Sid) VALUES

(78, 'BUS201', 21),

(92, 'ECO301', 22),

(88, 'MAT401', 23),

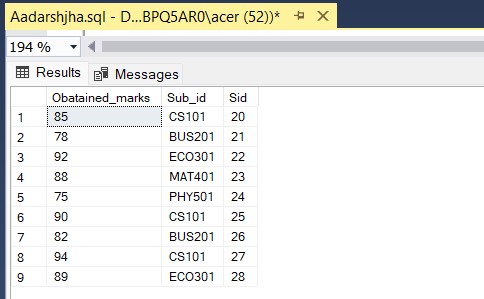
(75, 'PHY501', 24),

(90, 'CS101', 25),

(82, 'BUS201', 26),

(94, 'CS101', 27),

(89, 'ECO301', 28),



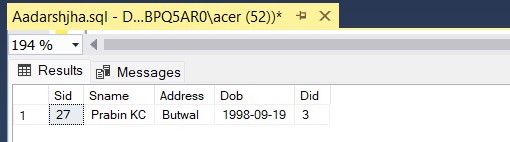
**Display records of those student who get maximum marks.**

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 ));



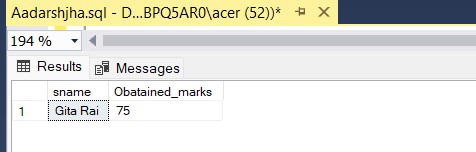
**Find name and marks of all students who get minimum marks.**

Query:

SELECT s.sname , m.Obatained\_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);

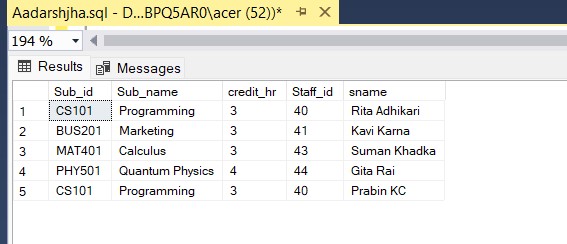


**Display list of subjects learned by student of dob less than 2045-10-10.** Query:

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 < '1999-10-10' ;



**Display name of all students of department 'Computer Science' or of address start with**

**'k'**

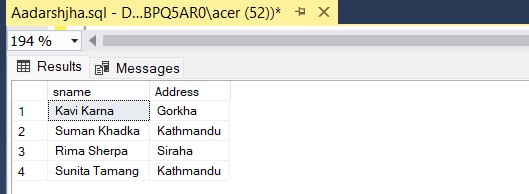
Query :

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 = 'Computer Science';



**Increase marks of all students of address 'Kathmandu' by 20%.**

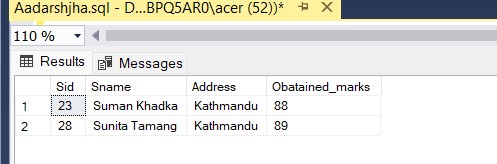
Query:

UPDATE Marks

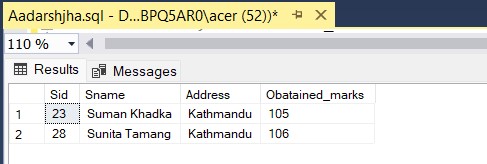
SET Obatained\_marks =Obatained\_marks + 0.2 \* Obatained\_marks

WHERE Sid IN (SELECT sid FROM student WHERE Address ='Kathmandu');

Before:



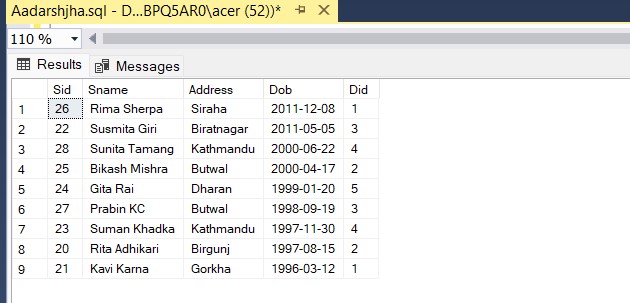
After:



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

Query: SELECT \* FROM Student s

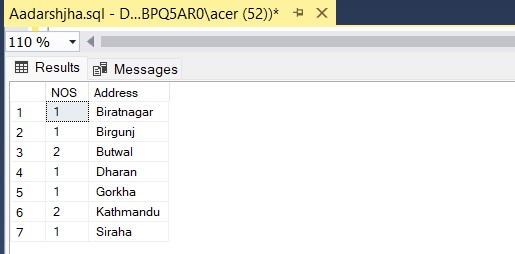
ORDER by s.Dob DESC;



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

Query : SELECT COUNT (sid) AS NOS ,Address

FROM Student GROUP BY (Address);

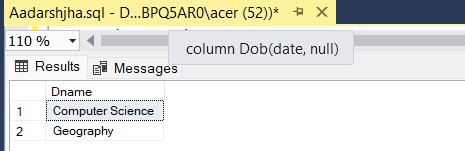


**Display all department with no students.**

Query : SELECT Dname FROM Student s

FULL OUTER JOIN Department d

ON s.did=d.did WHERE s.did is NULL;

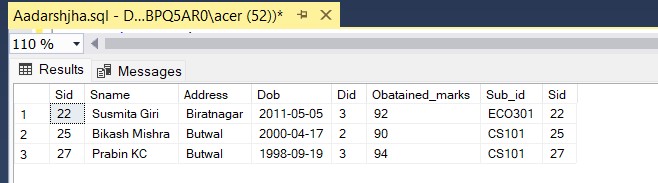


**Display records of all student of address start with 'B' and get greater or equal than average marks.**

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);



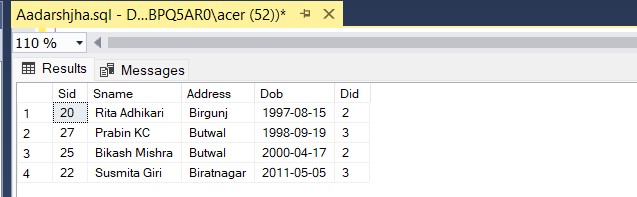
**Display 5 oldest student of address start with 'B'**

Query : SELECT TOP 5\*

FROM student s

WHERE s.ADDRESS LIKE 'B%'

ORDER BY Dob ASC ;



**Increase the credit hr of all subjects of name contains letter 'a' and study by student of address ‘Kathmandu’ by 2 hr.**

Query : UPDATE Subjects SET Credit\_hr = Credit\_hr +2 FROM Student s INNER JOIN Marks AS m

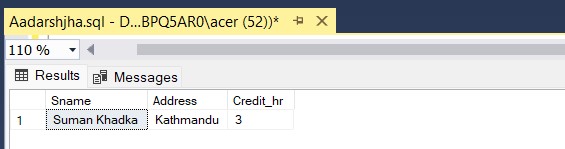
ON s.sid =m.sid INNER JOIN Subjects AS sub

ON sub.Sub\_id =m.Sub\_id WHERE sub.Sub\_name

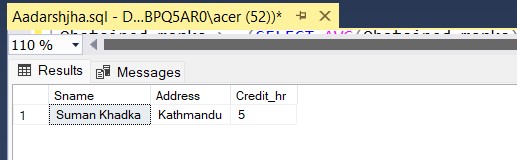
LIKE '%A%' AND s.Address ='Kathmandu' ;

1. **Lab 2- Multiple Table – DBMS**

Before:

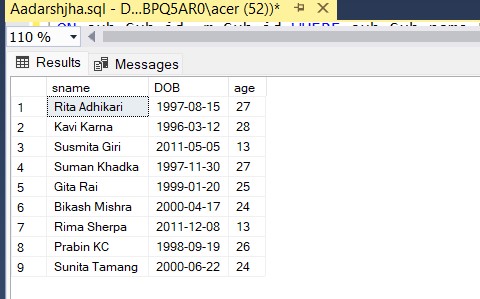


After:



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

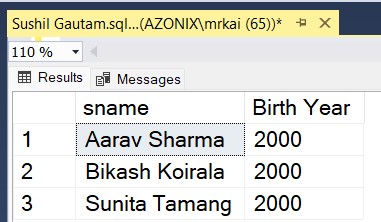
Query : SELECT sname, DOB, DATEDIFF (year,DOB,GETDATE()) AS age FROM Student ;



**Display only those student whose dob contain 2000 yr.**

Query : SELECT sname, year(DOB) as Birth\_Year

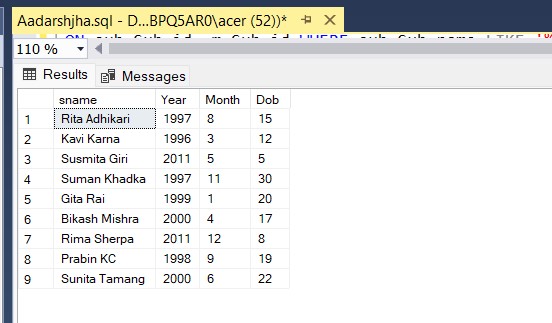
FROM Student WHERE year(DOB) = '2000'



1. **Lab 2- Multiple Table – DBMS**

**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 ;

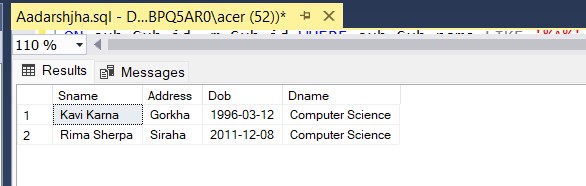


**Display all student who associated with ‘Computer Science’ department.**

Query : SELECT Sname,Address,Dob,Dname

FROM Department AS d INNER JOIN Student AS s

ON d.Did=s.Did WHERE d.Dname = 'Computer Science';



**Find join of above 5-tables.**

Query : SELECT \*

FROM Student s INNER JOIN Marks AS m

ON s.sid =m.sid

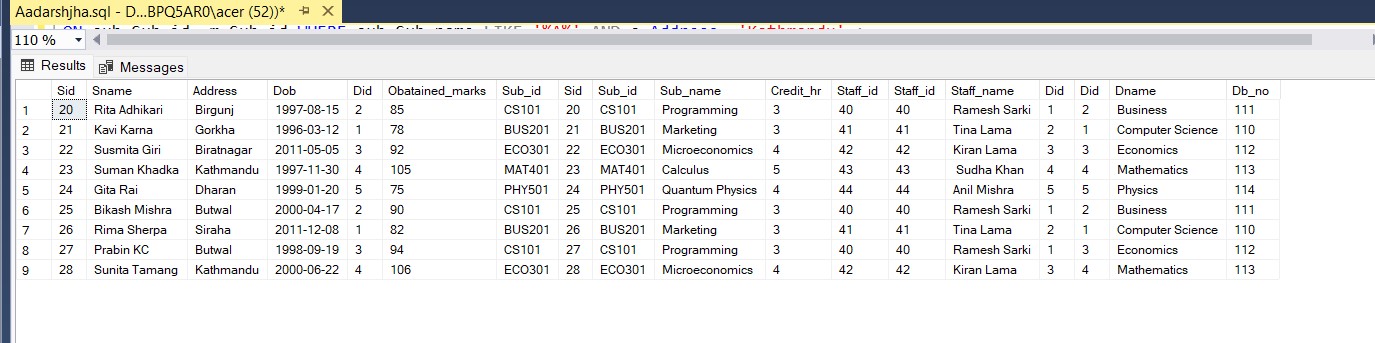
INNER JOIN Subjects AS sub

ON sub.Sub\_id =m.Sub\_id

INNER JOIN Staff AS st

ON st.Staff\_id =sub.Staff\_idINNER JOIN Department AS d ON d.Did =s.Did ;

11 **Lab 2- Multiple Table – DBMS**



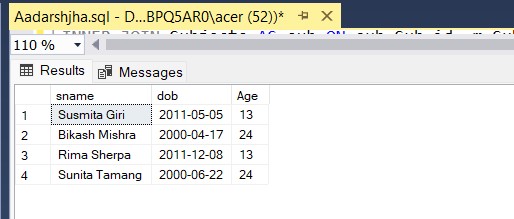
1 **Lab-3:Use of View- DBMS**

**1.Create a view ‘student\_view’ that display all student of age less than 25.**

Query : CREATE VIEW Student\_view as

SELECT sname ,dob,DATEDIFF (year,DOB,GETDATE()) AS Age

FROM Student WHERE DATEDIFF (year,DOB,GETDATE()) < 25;



* + 1. **Create a view 'Student\_subjects' that display all student who takes ‘Programming’ subject.**

Query : CREATE VIEW student\_subjects AS

SELECT sub.Sub\_id,sub.Sub\_name, s.Sname, s.Sid

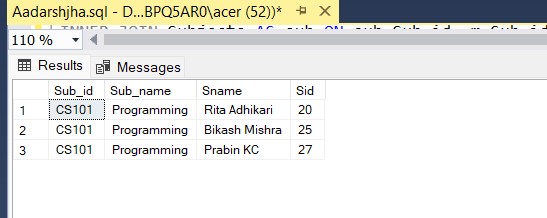
FROM Student s INNER JOIN Marks AS m

ON s.sid =m.sid

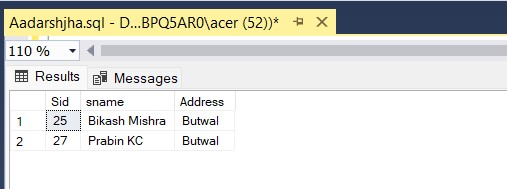
INNER JOIN Subjects AS sub

ON sub.Sub\_id =m.Sub\_id

WHERE Sub\_name ='Programming' ;



* + 1. **Create a view 'student details' that contain sid,sname and address of those student of address Butwal.**

Query :

CREATE VIEW student\_Details AS

SELECT Sid, sname,Address

FROM Student

WHERE address ='Butwal';

2 **Lab-3:Use of View- DBMS**

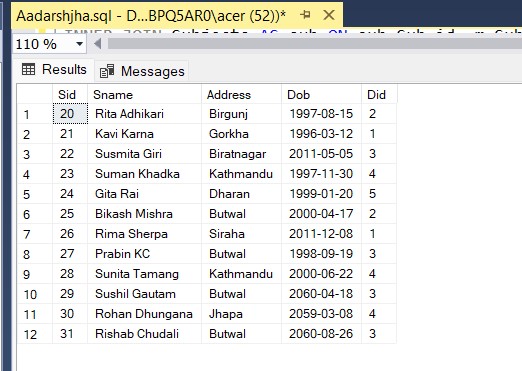
* + 1. **Insert any 3 additional records to Student table.**

Query : INSERT INTO Student(Sname, Address, DOB , DID)

VALUES ('Sushil Gautam' , 'Butwal' , '2060-04-18', 3),

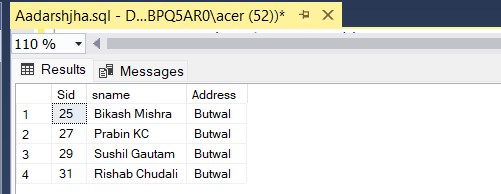
('Rohan Dhungana' , 'Jhapa' , '2059-03-08', 4),

('Rishab Chudali' , 'Butwal' , '2060-08-26', 3)



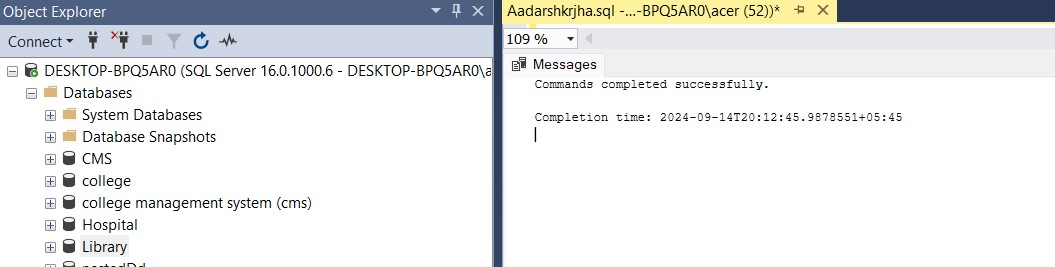
* + 1. **Display the view ‘ student\_details’**

Query : SELECT \* FROM student\_Details ;



1. **Create database name ‘ Library ’.**

Query : CREATE DATABASE Library;



1. **Create table Books,Student and Author with proper constraints.** Query :

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 Book (

ISBN VARCHAR(10) PRIMARY KEY,

bname VARCHAR (20), price INT NOT NULL, noP INT UNIQUE , sid INT,

FOREIGN KEY(sid) references Student (sid)

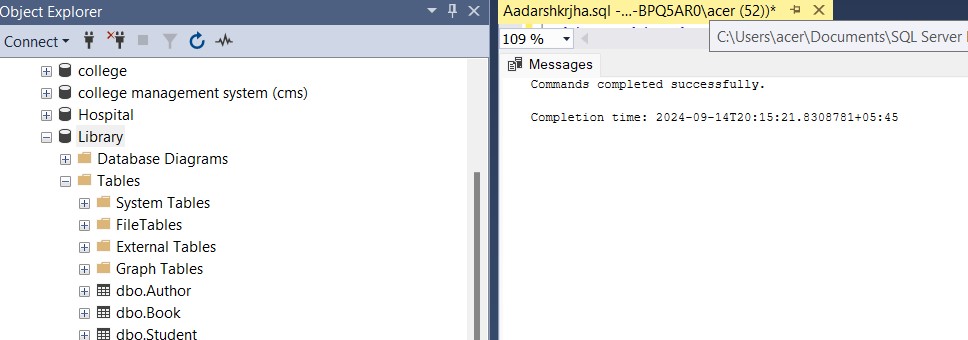
);

CREATE TABLE 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, address) VALUES

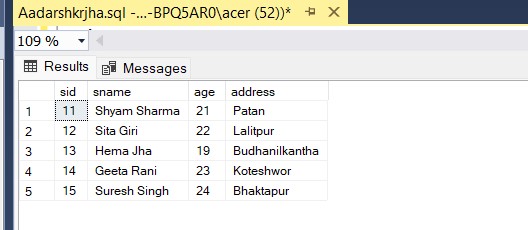
('Shyam Sharma', 21, 'Patan'),

('Sita Giri', 22, 'Lalitpur'),

('Hema Jha', 19, 'Budhanilkantha'),

('Geeta Rani', 23, 'Koteshwor'),

('Suresh Singh', 24, 'Bhaktapur');



INSERT INTO Book (ISBN, bname, price, noP, sid) VALUES

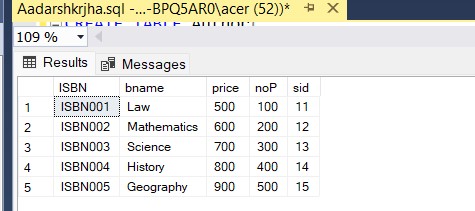
('ISBN001', 'Law', 500, 100, 11),

('ISBN002', 'Mathematics', 600, 200, 12),

('ISBN003', 'Science', 700, 300, 13),

('ISBN004', 'History', 800, 400, 14),

('ISBN005', 'Geography', 900, 500, 15);



INSERT INTO Author (aname, Phone\_no, address, ISBN) VALUES

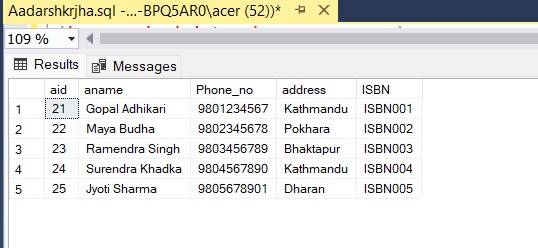
('Gopal Adhikari', '9801234567', 'Kathmandu', 'ISBN001'),

('Maya Budha', '9802345678', 'Pokhara', 'ISBN002'),

('Ramendra Singh', '9803456789', 'Bhaktapur', 'ISBN003'),

('Surendra Khadka', '9804567890', 'Kathmandu', 'ISBN004'),

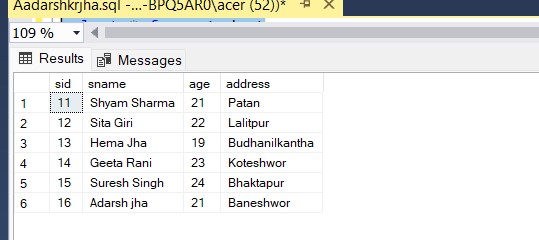
('Jyoti Sharma', '9805678901', 'Dharan', 'ISBN005');



1. **Test the ‘default’ constraints.**

Here, Address attribute of Student table has default value as ‘Baneshwor’ .

Query : INSERT INTO Student(sname, age) values('Shushil Gautam', 21)

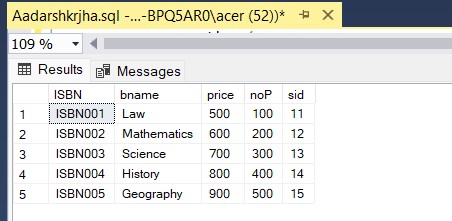


1. **Test for ‘Unique’ constraint.**

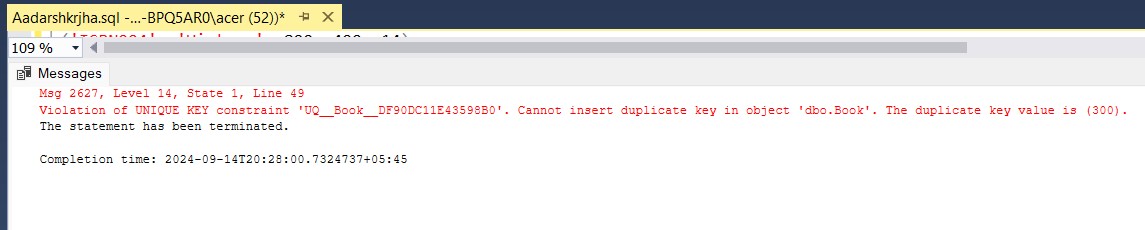
Here , nop attribute of Book table has unique constraint.

Query : INSERT INTO Book (ISBN, bname, price, noP, sid)

VALUES ('ISBN006', 'Philosophy', 1000, 300, 11); Before :



After :

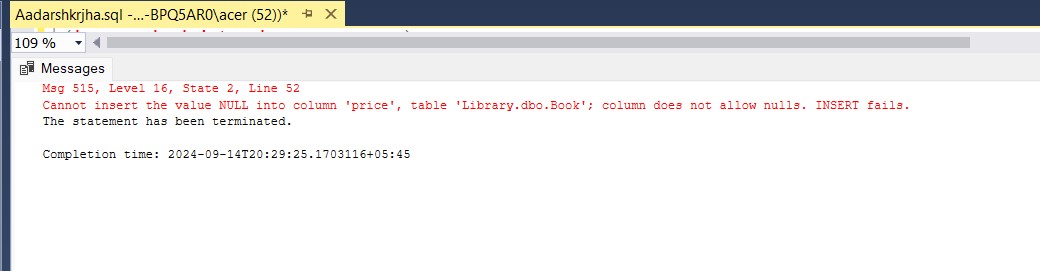


1. **Test for ‘Not Null ’ constraint.**

Here , price attribute of Book table has NOT NULL Constraint.

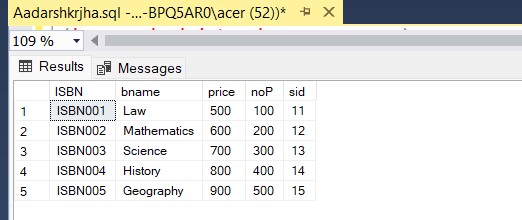
Query : INSERT INTO Book (ISBN, bname, price, noP, sid)

VALUES ('ISBN007', 'Literature', NULL, 600, 12);



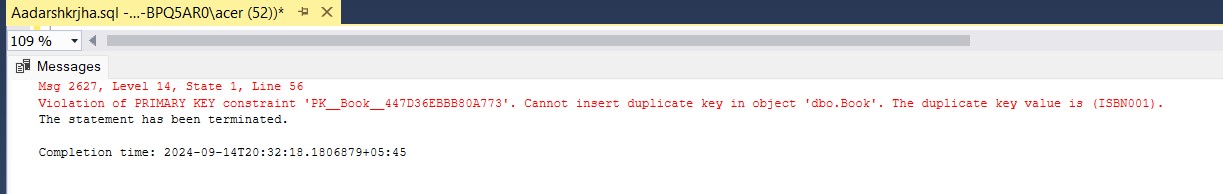
1. **Test for ‘Primary Key’ constraint.**

Here , ISBN attiibute of Book table is a PRIMARY KEY



Query : INSERT INTO Book (ISBN, bname, price, noP, sid) VALUES

('ISBN001', 'Physics', 1100, 600, 13);

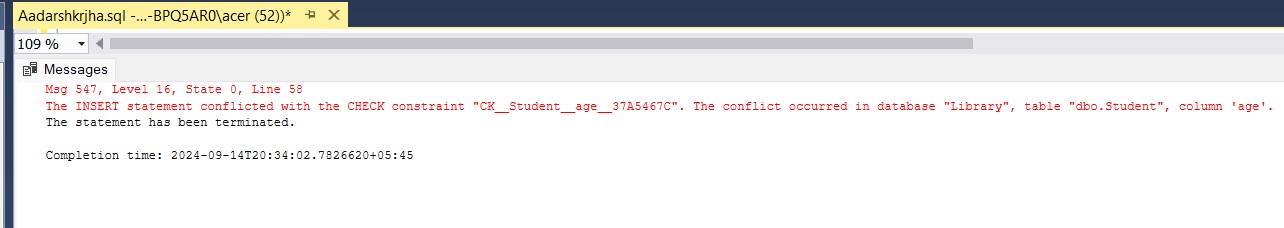


1. **Test for ‘Check’ constraint.**

Here , age attribute of Student table has CHECK Constraint as age INT CHECK(age > 0 and age < 110)

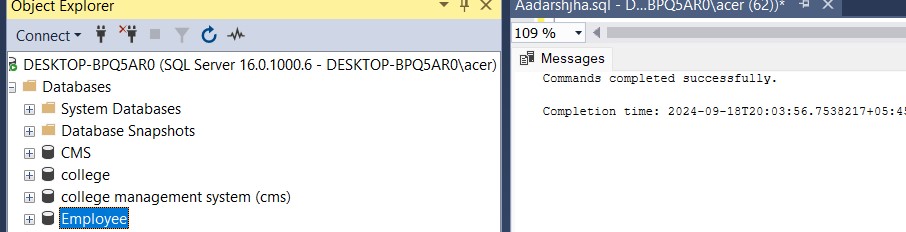
Query : INSERT INTO Student (sname, age, address) VALUES

('Sushil Gautam', 120, 'Santinagar');



1.Create database name ‘Employee’

Query: CREATE DATABASE Employee;



2.Create table Employee,Employee\_log and Total\_salary with proper constraints.

Query:CREATE TABLE Employee

(eid INT NOT NULL PRIMARY KEY ,

ename VARCHAR(20), salary FLOAT,

Address VARCHAR(20) );

CREATE TABLE Employee\_log

(eid INT,

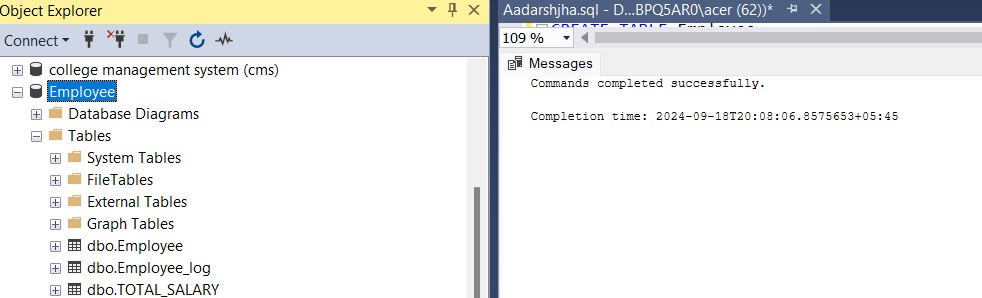
ename VARCHAR(20),

old\_salary FLOAT, new\_salary FLOAT,

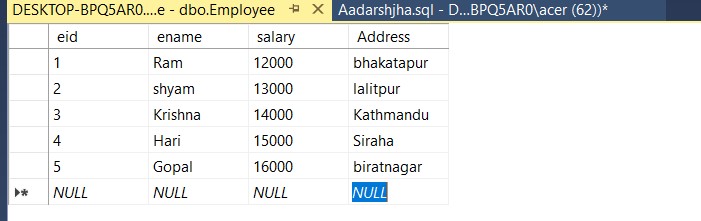
date\_time DATETIME );

CREATE TABLE TOTAL\_SALARY

( salary\_sum FLOAT );



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



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

Query:CREATE trigger total\_salary\_update

ON Employee

AFTER INSERT,UPDATE,DELETE

AS

BEGIN

declare @total float

SELECT @total=SUM(salary)

FROM employee ;

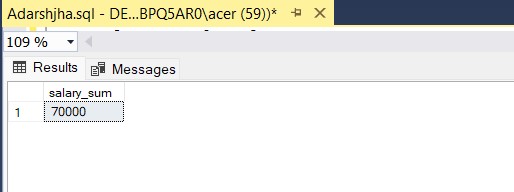
UPDATE total\_salary

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

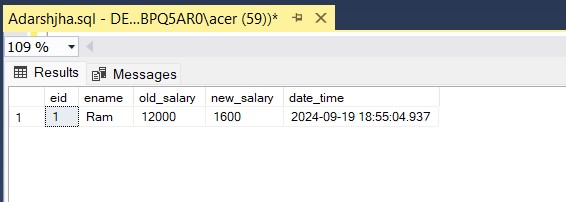
JOIN deleted

on inserted.eid=deleted.eid END ;

7.Display Employee\_log table after activation of trigger Employee\_Log\_Update.

Query: SELECT\*

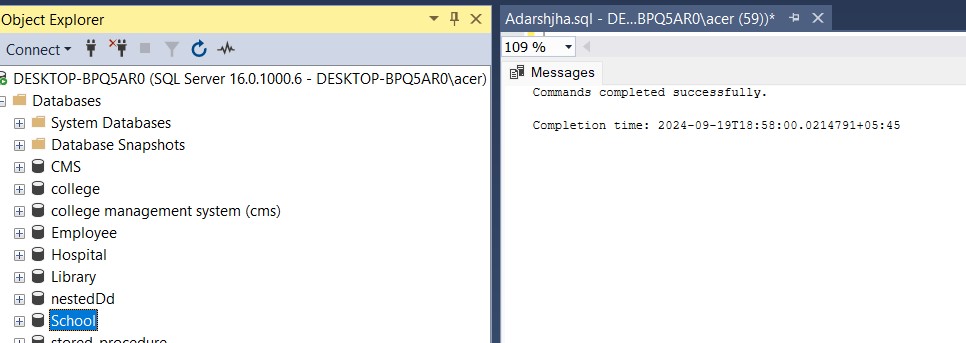
FROM Employee\_log



**Lab-6 Store Procedure - DBMS**

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

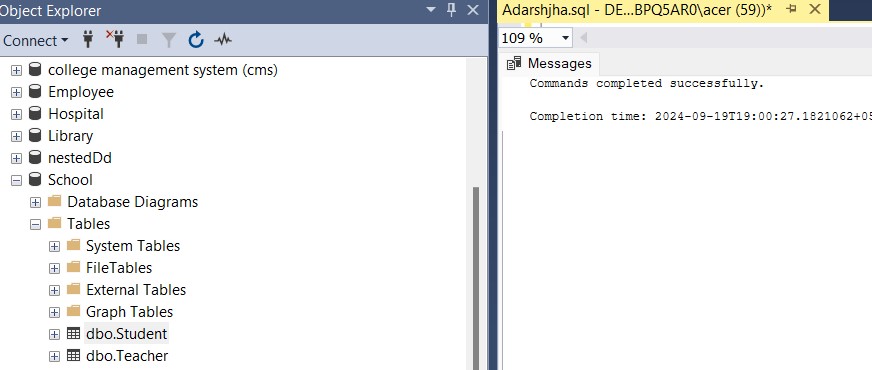
(sid INT,

sname VARCHAR(20),

marks FLOAT,

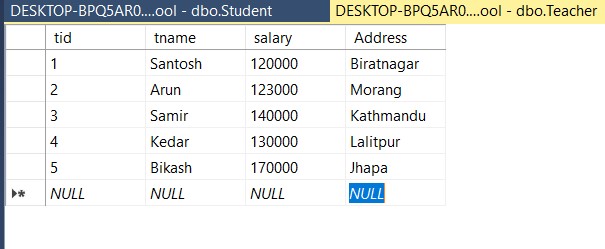
tid INT,

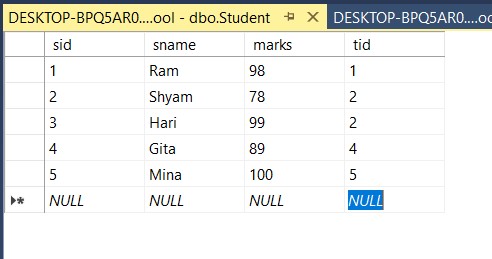
FOREIGN KEY (tid) references Teacher(tid) );



**Lab-6 : Use of Store Procedure - DBMS**

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





4.Create a Stored Procedure teacher\_student that retrieves data by joining the 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;

