**Lab 1: Installation of MS-SQL Server.**

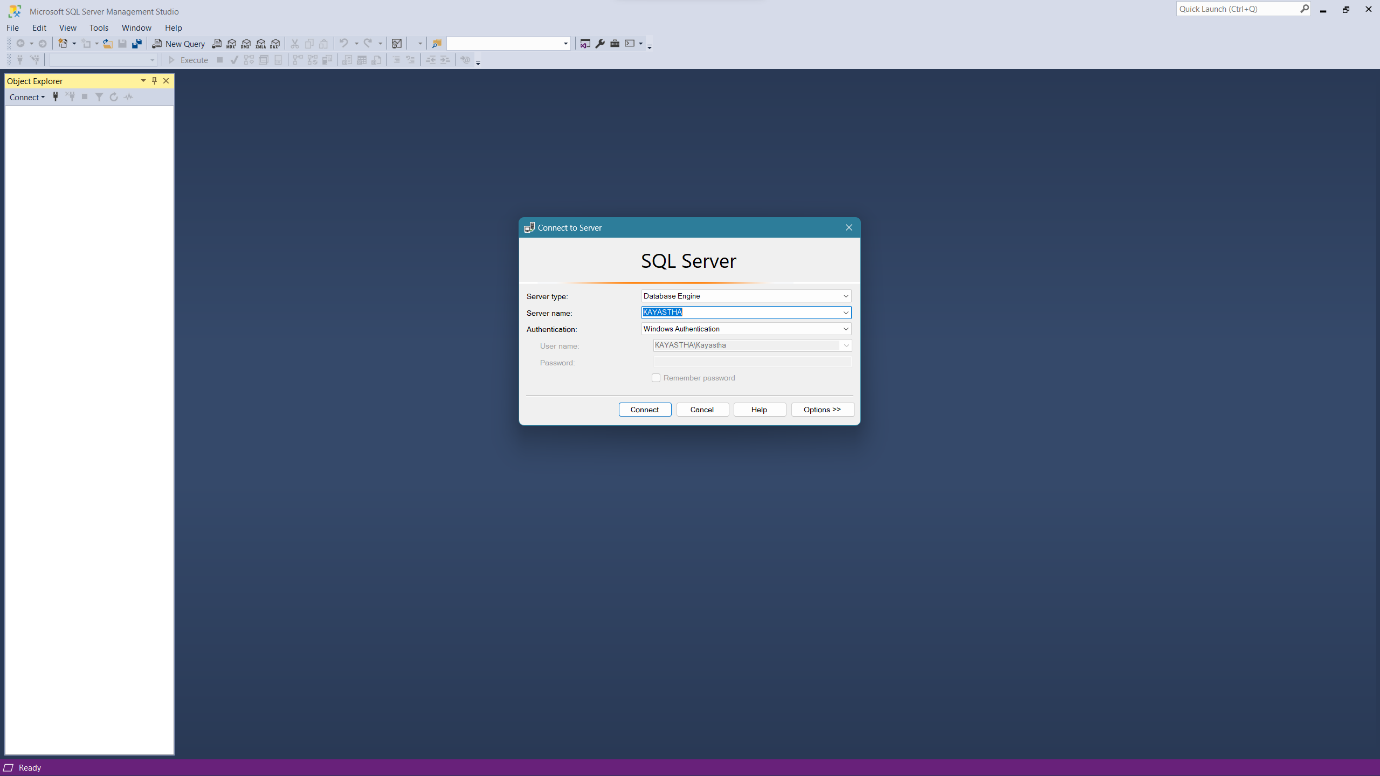
1. Firstly, we download MS-SQL server file.
2. Then, we click on “SQLEXPRA\_X64\_ENU” application to extract “SQLEXPRADV\_X64\_ENU” file.
3. We click on the file and then, click ‘setup’.
4. Click on ‘Installation’ and ‘New SQL server stand-alone’ installation or add features to an existing installation.
5. Then click on, ‘I accept the license term’ and ‘Next’.
6. After clicking few ‘Next’ options, we have to choose ‘Instance Features’. Choose according to the need and ‘Next’.
7. We then choose ‘Default Instance’ and click ‘Next’.
8. After specifying the ‘authentication mode’, click ‘Next’. After clicking, it takes some time to install supporting files and then displays ‘Install Successful’.

Now, We need to install SSMS for connecting SQL Server database.

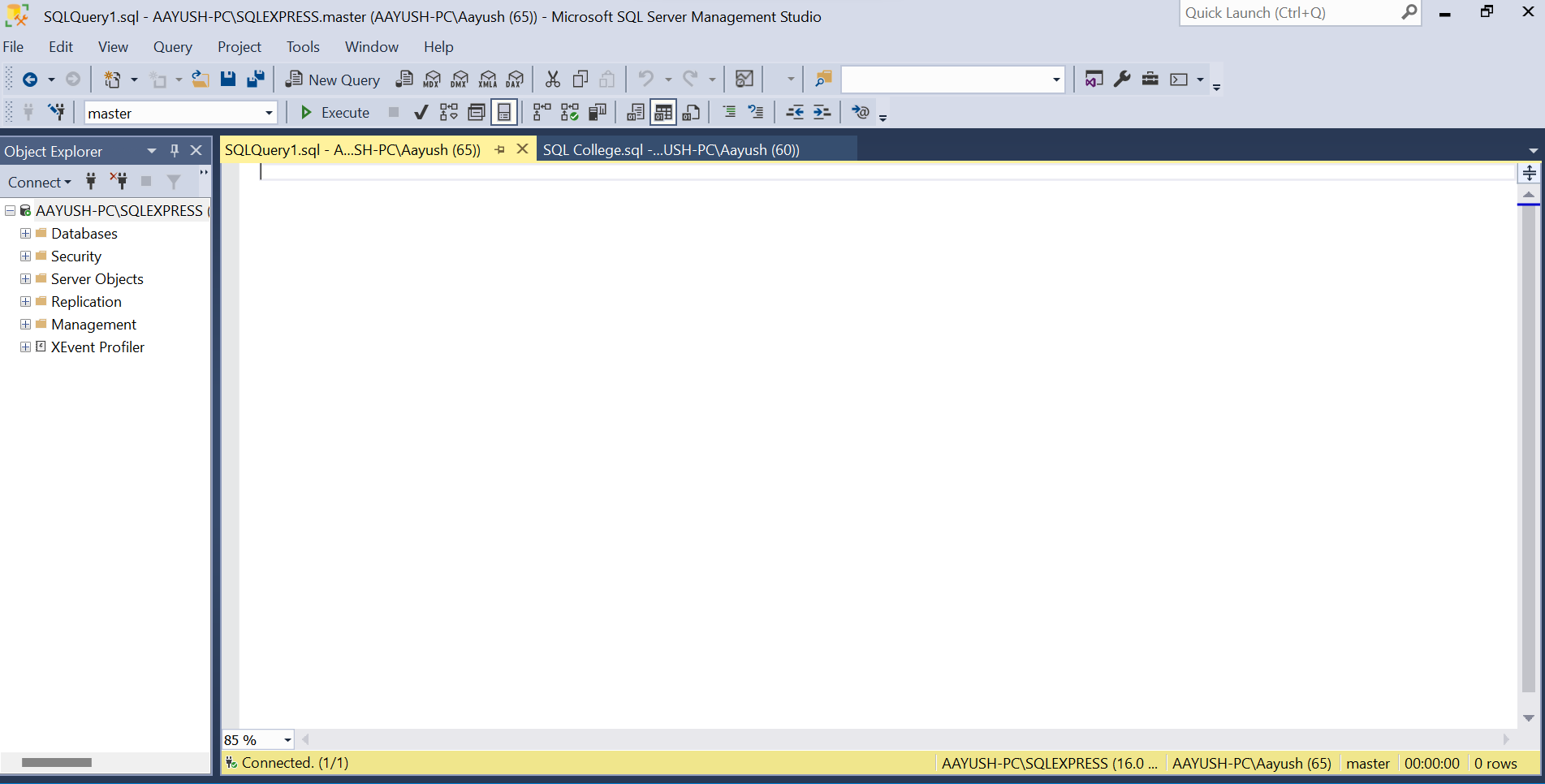
1. Download SQL Server Management Studio (SSMS).
2. Then, click on SSMS-Setup-ENU, choose the location where to install and click ‘INSTALL’.
3. After some time, a window is displayed saying ‘The computer needs to restart before setup can continue’. Click ‘Restart’.
4. After Restart, the file is downloaded. Now click on ‘Microsoft SQL Server Management Studio 18’.
5. After doing installation, we now specify the server type, server name, Authentication and click ‘Connect’.

We have successfully installed both the SQL server and SSMS.

1. Connection of SQL Server and SSMS:



1. Code writing area:



**Lab 2: Creation of a Database and Table.**

After installing both the SQL server and SSMS, we now create a database and a Table within the database.

1. **Creation of Database**
   1. Open SSMS.
   2. Connect with the servers.
   3. Click on ‘New Query’.
   4. Type ‘CREATE DATABASE student1’ to create a database named ‘student1’. Then click ‘Execute’.

Now, We can create as many tables as we want inside the database ‘student1’.

1. **Creation of a Table**

We setup the server in order to save the formed tables in ‘student1’ database. We then, type the code for the creation of table named ‘p1’.

CREATE TABLE p1

(

roll\_no int,

name varchar(20),

birth\_date int

) ;

*/\*This creates the table ‘p1’ having rows ‘roll\_no’,’name’,’birth\_date’.*

*Now, we insert the values into table accordingly\*/*

*/\*Code\*/*

INSERT INTO p1

VALUES(1,’RABIN’,20);

*/\* Here, roll\_no is ‘1’, name is ‘RABIN’, birth\_date is ’20’.*

*To assign values to the respective row, we execute the code\*/*

Select \* from p1

*/\*This code replaces the respective values with the correct rows.\*/*

*/\*For deleting the values from the table, we execute the code:\*/*

DROP TABLE p1

*/\*This code deletes all values from ‘p1’ table.\*/*

**Combined code:**

CREATE TABLE p1

(

roll\_no int,

name varchar(20),

birth\_date int

) ;

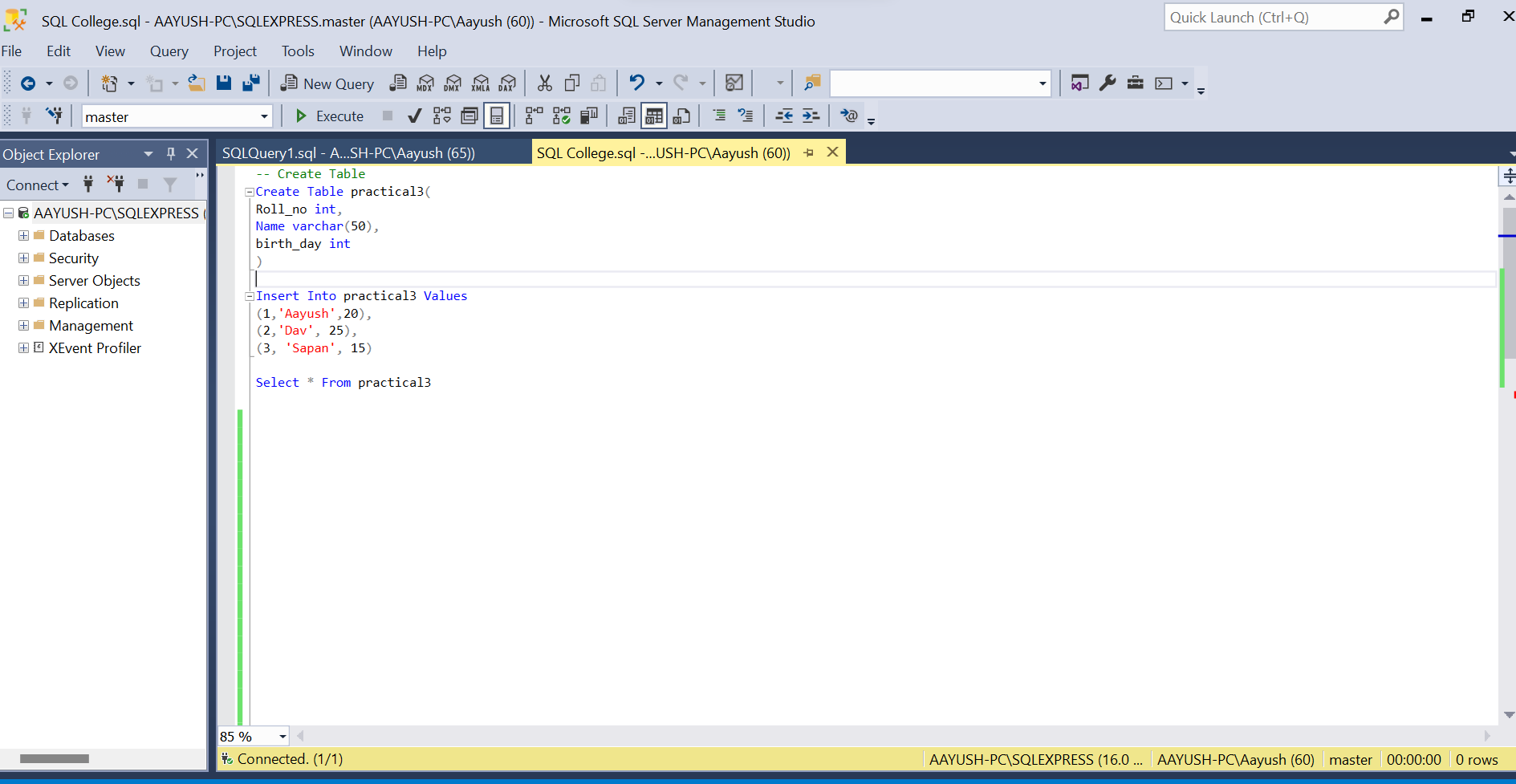
INSERT INTO p1

VALUES(1,’Aayush’,20);

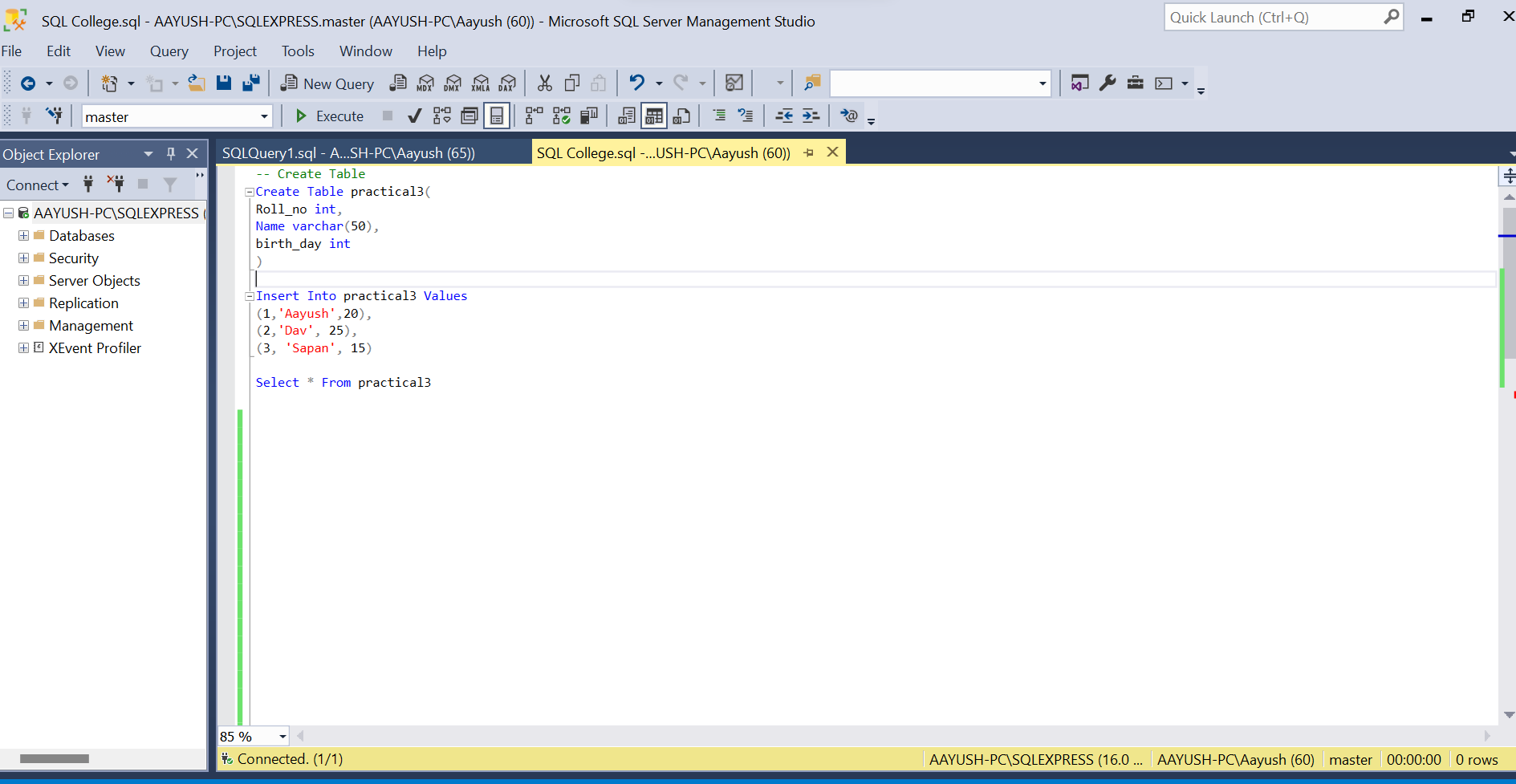
Select \* from p1

DROP TABLE p1

1. Creation of Table:



1. Insertion of values in the table:



**Lab 3: Creation of Table named ‘Emp’, adding multiple rows in the table and altering the table.**

We have successfully created a table and have put a row of values in the table. Here, we will create a table having multiple row values and finally, alter the table by adding extra column.

1. */\*Creation of ‘Emp’\*/*

*/\*CODE:\*/*

CREATE TABLE Emp

(

EMP\_ID int,

Name varchar(20),

Age int,

Address varchar(100),

Salary numeric(10,2)

);

1. *\*/Inserting the values in the table ‘Emp’\*/*

INSERT INTO Emp

VALUES(1,’RABIN’,30,’Canada’,10000),

(2,’Raunak’,22,’Birgunj’,1000000),

(3,’Bini’,21,’Kotwashor’,1000000),

(4,’Jeny’,21,’Teaching’,1000000),

(5,’Kiran’,20,’Seto Pul’,1000000),

(6,’Anubhav’,21,’Sukhadhara’,1000000);

Select \* from Emp

*/\*By executing the above code, the table gets filled with values.\*/*

1. */\*Altering ‘Emp’ by adding a column ‘Email’\*/*

*/\*This is used if we must add an extra column in the table after the table is created. The code is: \*/*

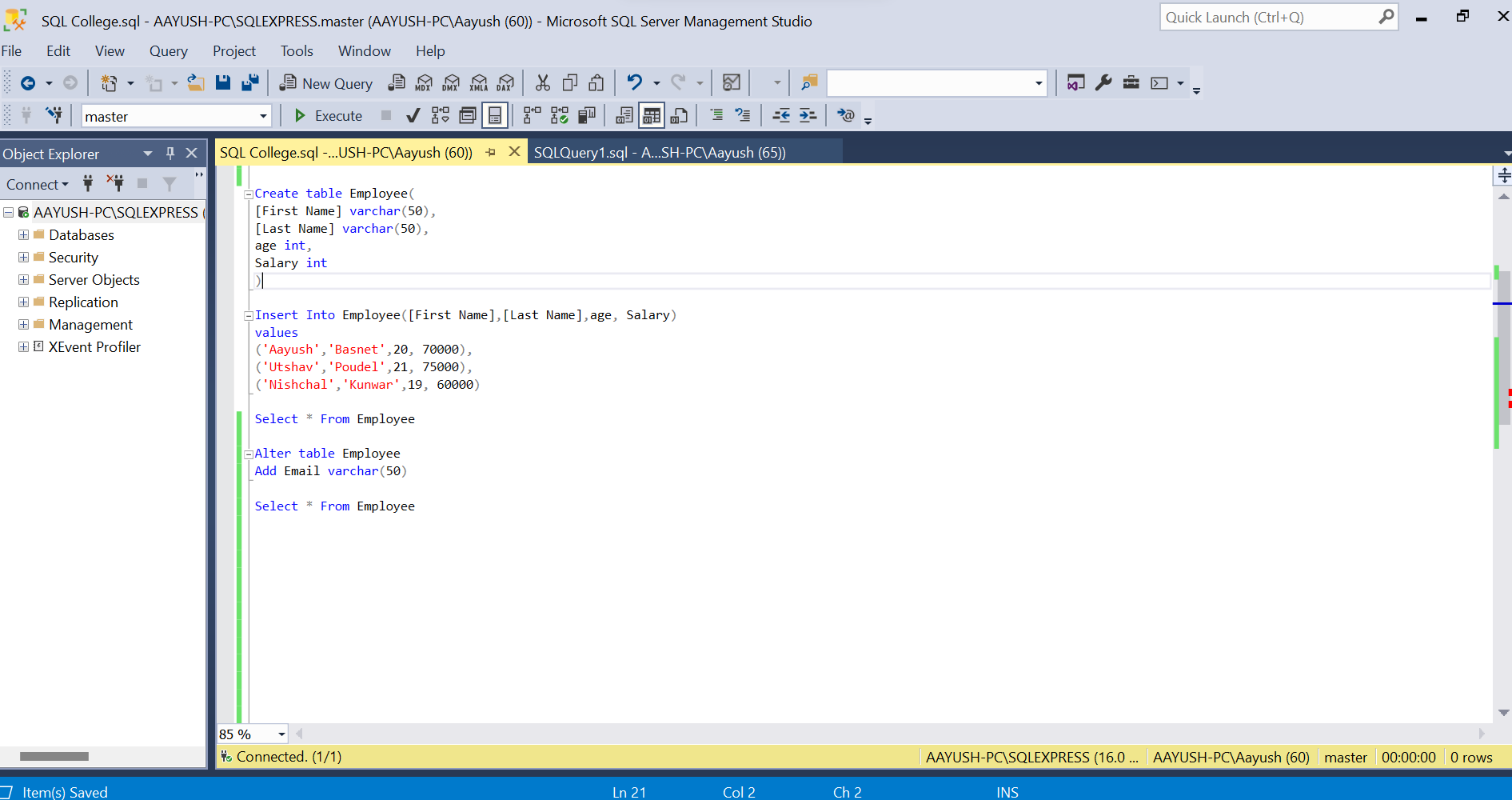
ALTER TABLE Emp

ADD Email varchar(100);

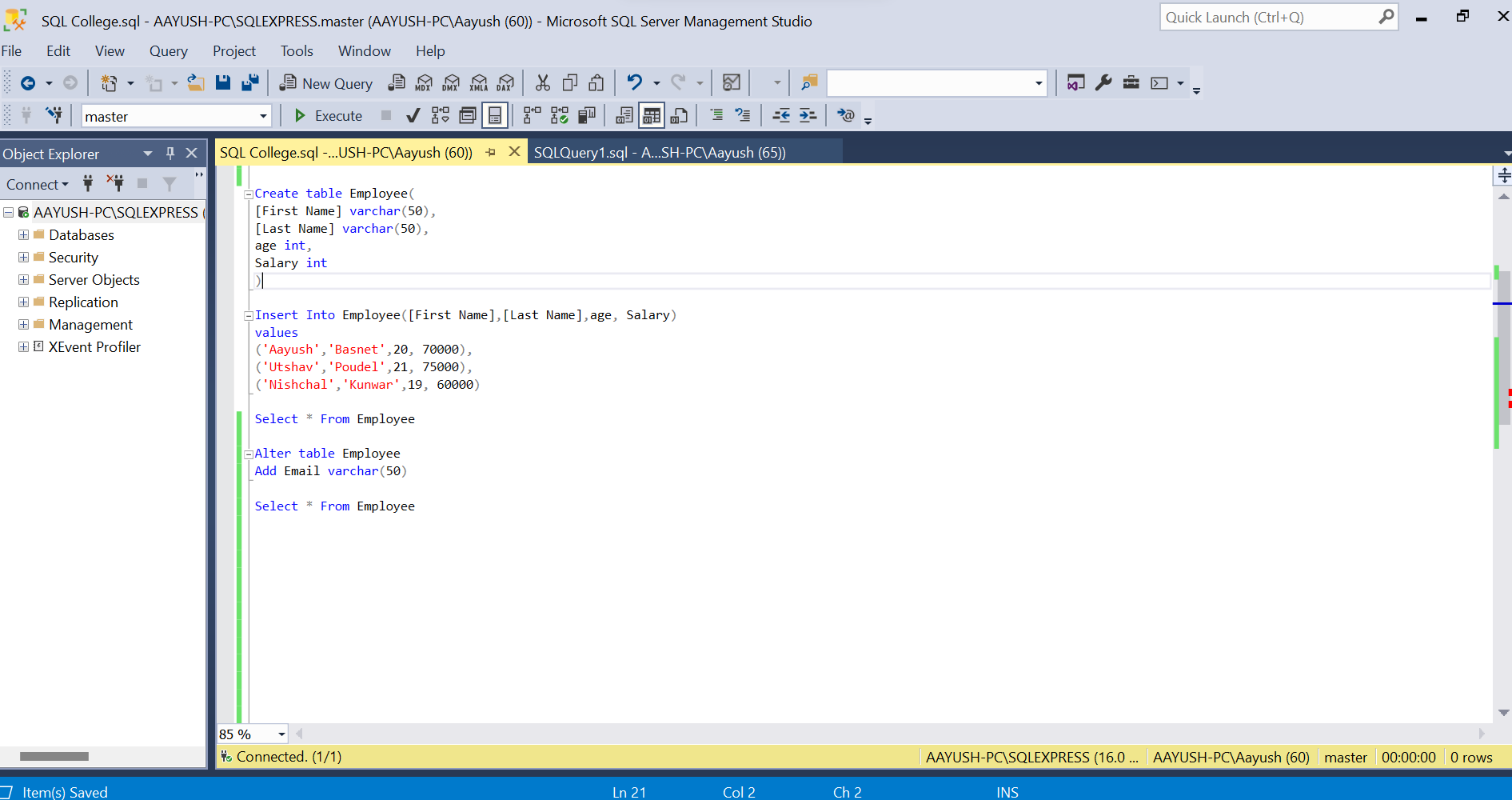
Select \* from Emp

*/\*By executing this code , we will add an extra column ‘Email’ in the already made table\*/*

1. Insertion of values in the table ‘Emp’:



1. Altering the table by adding a column ‘Email’:



**Lab 4: Creation of Table named ‘p3’, and altering the table using ‘TRUNCATE’.**

1. */\*The code for creation of table is as follows:\*/*

CREATE TABLE p3

(

roll int,

name varchar(20),

birth int

);

INSERT INTO p3

VALUES(1,’Bini’,21,’bini@gmail.com’),

(2,’Raunak’,22,’raunakshrivastabking@gmail.com’),

(3,’Shreeja’,20,’shreeja@gmail.com’),

(4,’Jeny’,21,’jeny@gmail.com’),

(5,’Kiran’,20,’kiran@gmail.com),

(6,’Anubhav’,22,’aban@gmail.com’);

1. */\*Until now, we have columns for roll, name and birth(date) only. Now, we add a column for email address in the existing table.\*/*

ALTER TABLE p3

ADD email varchar(100);

1. */\*Now, we use ‘TRUNCATE’ to remove the data present in the table without deleting the whole table, unlike ‘DROP’.\*/*

TRUNCATE TABLE p3;

1. */\*To add phone no. and address column:\*/*

ALTER TABLE p3

ADD phone\_no varchar(100);

ALTER TABLE p3

ADD Address varchar(100);

**Combined code:**

CREATE TABLE p3

(

roll int,

name varchar(20),

birth int

);

INSERT INTO p3

VALUES(1,’Bini’,21,’bini@gmail.com’),

(2,’Raunak’,22,’raunakshrivastabking@gmail.com’),

(3,’Shreeja’,20,’shreeja@gmail.com’),

(4,’Jeny’,21,’jeny@gmail.com’),

(5,’Kiran’,20,’kiran@gmail.com’),

(6,’Anubhav’,22,’aban@gmail.com’);

ALTER TABLE p3

ADD email varchar(100);

TRUNCATE TABLE p3;

ALTER TABLE p3

ADD phone\_no varchar(100);

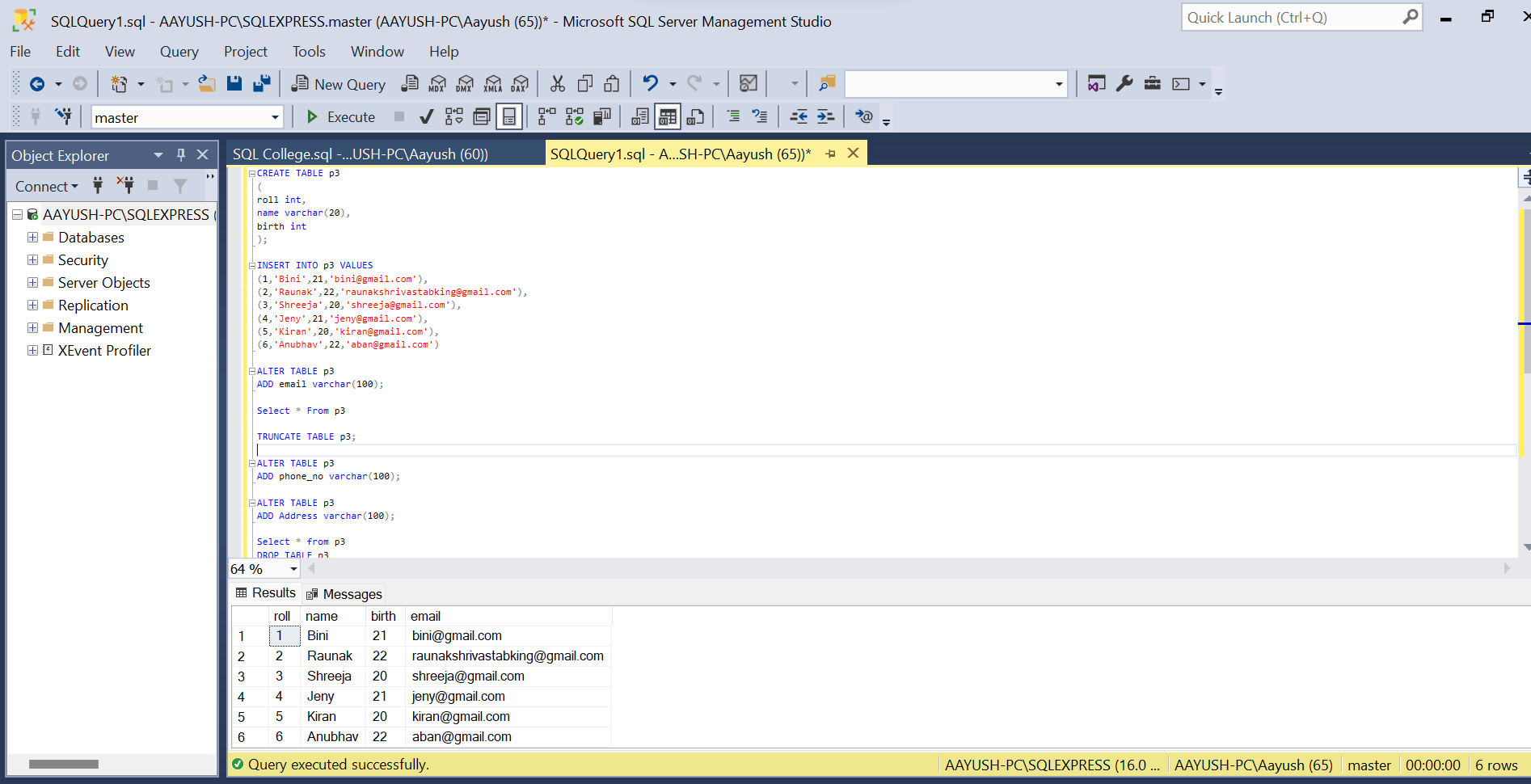
ALTER TABLE p3

ADD Address varchar(100);

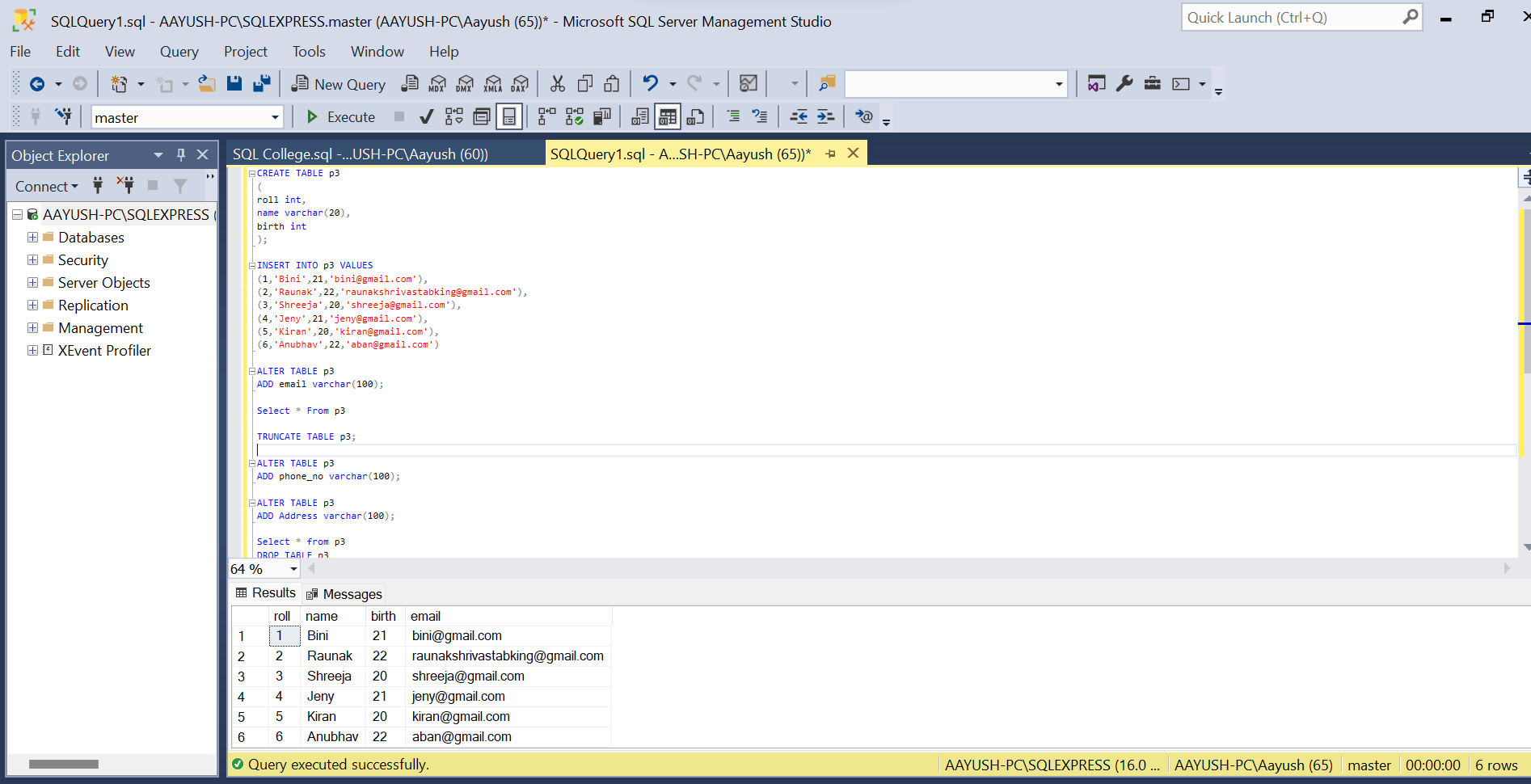
Select \* from p3

DROP TABLE p3

1. Creation of table and then insertion of extra column named ‘email’:



1. Deletion of data of the table using “TRUNCATE”:

****

**Lab 5: Insertion of data in table using assigned attributes.**

The code for the creation of the table and insertion of data in the respective columns using assigned attributes is as follows:

CREATE TABLE emp

(

[FirstName] varchar(50),

[MiddleName] varchar(50),

[LastName] varchar(50),

[Age] int

);

INSERT INTO emp(FirstName,MiddleName,LastName,Age)

VALUES('Raunak','Kumar','Shrivastab',22),

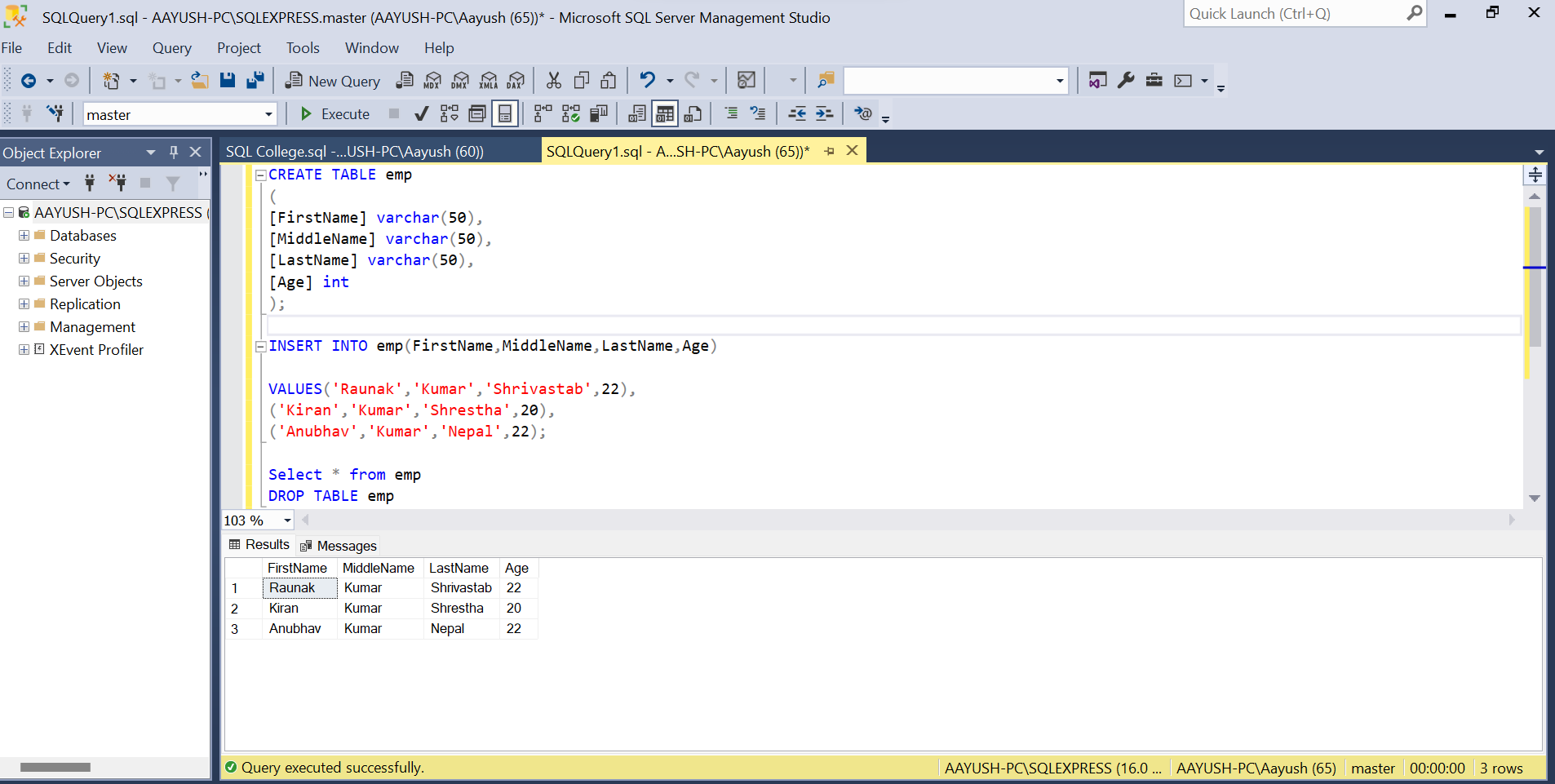
('Kiran','Kumar','Shrestha',20),

('Anubhav','Kumar','Nepal',22);

Select \* from emp

DROP TABLE emp

1. Insertion of data in the table using assigned attributes:



**Lab 6: Create table, insert, update, delete in SQL along with primary key.**

1. */\*The code for creation of table is as follows:\*/*

CREATE TABLE emp\_tbl

(

empId int primary key,

FirstName varchar(50),

LastName varchar(50),

salary float,

MobileNo varchar(15)

);

select \* from emp\_tbl

insert into emp\_tbl

VALUES(6,'Kiran','shrestha',2000,'9845071651'),

(7,'Anubhav','Nepal',10000,'966656678'),

(9,'Raunak','Shrivastab',500000,'966765665');

1. */\*delete empID=1 from the existing table \*/*

delete from emp\_tbl where empID=1

1. */\*updating the data of single row from table using its empID\*/*

UPDATE emp\_tbl set FirstName='Bini' where empID=3

DROP TABLE emp\_tbl

**Combined code:**

CREATE TABLE emp\_tbl

(

empId int primary key,

FirstName varchar(50),

LastName varchar(50),

salary float,

MobileNo varchar(15)

);

select \* from emp\_tbl

insert into emp\_tbl

VALUES(6,'Kiran','shrestha',2000,'9845071651'),

(7,'Anubhav','Nepal',10000,'966656678'),

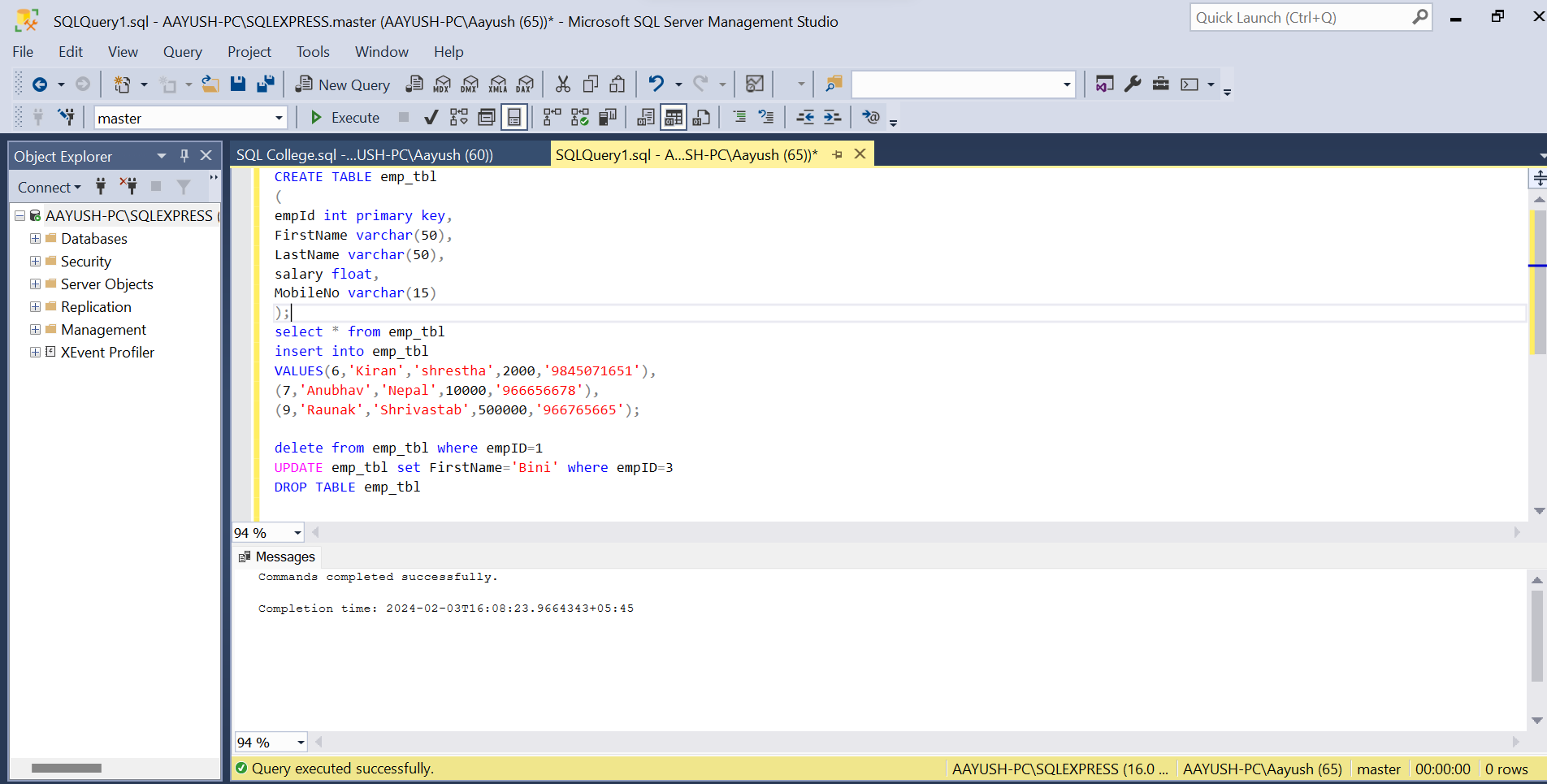
(9,'Raunak','Shrivastab',500000,'966765665');

delete from emp\_tbl where empID=1

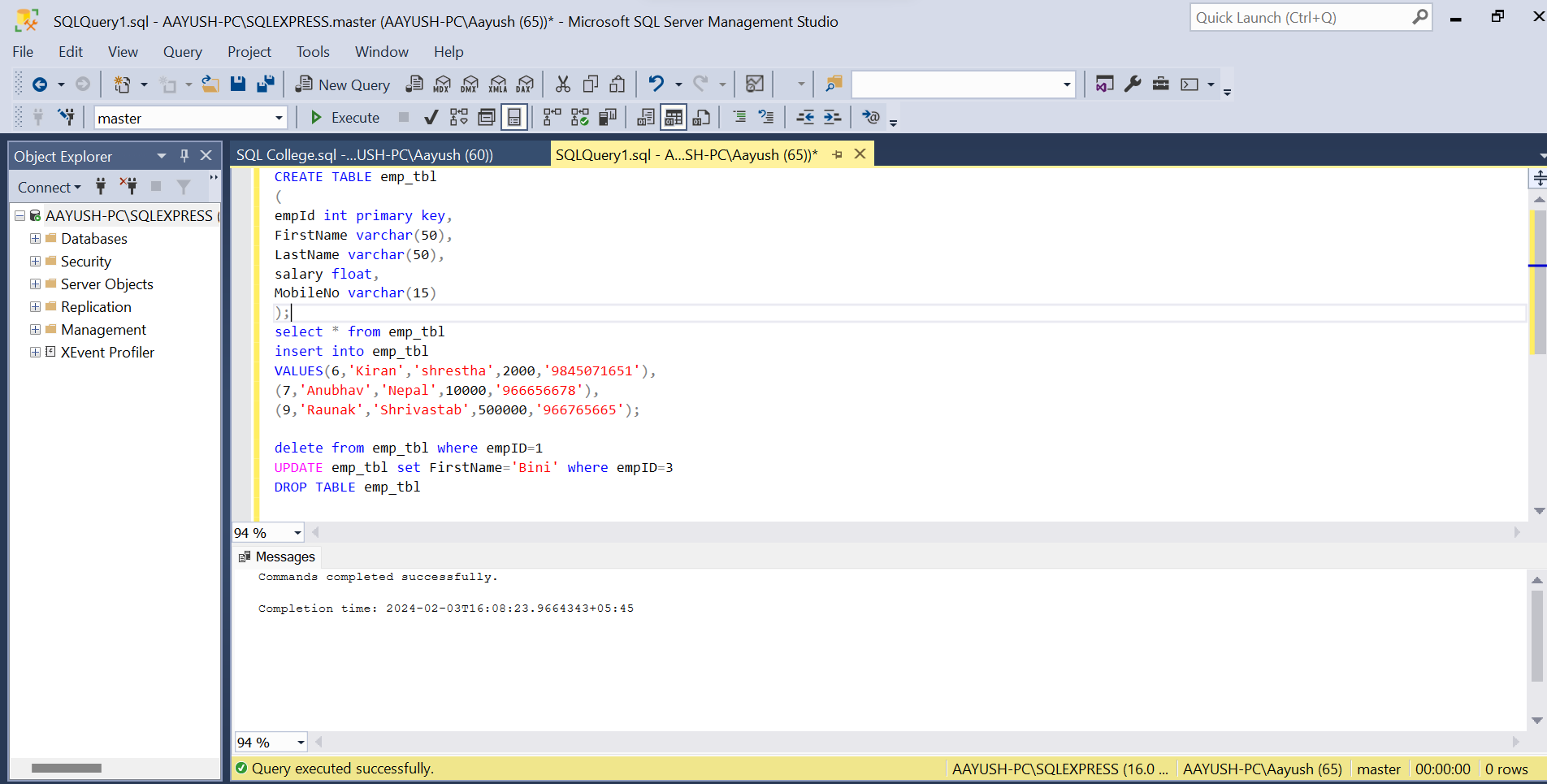
UPDATE emp\_tbl set FirstName='Bini' where empID=3

DROP TABLE emp\_tbl

1. Creation of table and then remove any individual row from table using ‘empID’ attributes:



1. Updating of data of individual row from table using ‘empID’ attributes:



**Lab 7: Create a table customer5 with following field with conditional statement.**

**Code:**

CREATE TABLE CUSTOMER5

(

CUST\_NO numeric(5) PRIMARY KEY,

LNAME VARCHAR(10),

FNAME VARCHAR(10) NOT NULL,

ADDR VARCHAR(20),

CITY VARCHAR(10),

STATES VARCHAR(10),

PIN VARCHAR(5),

BIRTH\_DATE DATE,

STATUS VARCHAR(1),

CHECK (STATUS IN ('V','I'))

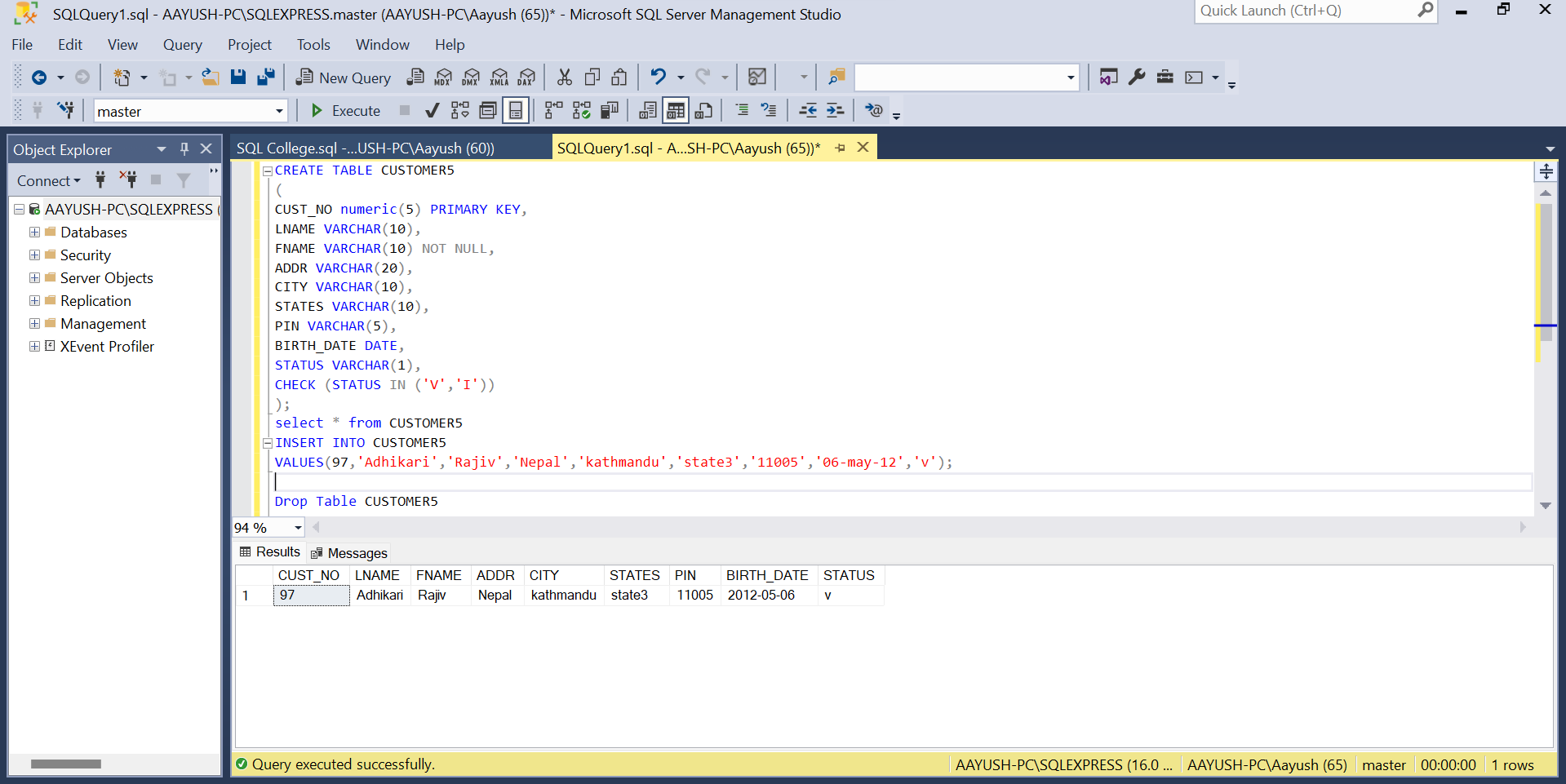
);

select \* from CUSTOMER5

INSERT INTO CUSTOMER5

VALUES(97,'Adhikari','Rajiv','Nepal','kathmandu','state3','11005','06-may-12','v');

1. Here we have used conditional statements:



**Lab 8: To Implement Primary Key and Foreign Key Concept In SQL.**

**Theory:**

In SQL, a primary key is a column or a set of columns that uniquely identifies each row in a table. A foreign key is a column or a set of columns in one table that refers to the primary key of another table. The primary key and foreign key concepts are used to establish relationships between tables in a database.

To implement primary key and foreign key concept in SQL, follow these steps:

1. Create the tables: Create the tables that you want to establish a relationship between. Each table should have a primary key column that uniquely identifies each row.
2. Define the primary key: Define the primary key column in each table. The primary key column should be unique and not null.
3. Define the foreign key: Define the foreign key column in the table that will refer to the primary key of another table. The foreign key column should have the same data type as the primary key column.
4. Establish the relationship: Establish the relationship between the tables by specifying the foreign key column in the child table and the primary key column in the parent table.
5. Enforce referential integrity: Enforce referential integrity by specifying that the foreign key column in the child table must refer to a valid primary key in the parent table.

**Code:**

CREATE TABLE STD\_ADD

(

Roll\_no numeric(14) primary key,

Names varchar(14),

Addr varchar(14),

Place varchar(14),

pin varchar(10),

);

CREATE TABLE STD\_MARKS

(

Roll\_no numeric(14) References STD\_ADD,

Subjects varchar(14),

Exam\_date date,

Marks numeric(3)

);

SELECT \* from STD\_ADD

SELECT \* from STD\_MARKS

insert into STD\_ADD

values(0,'Binisha','Kotwashor','Kathmandu','45'),

(10,'Raunak','Narephet','Kotwashor','5'),

(20,'Kiran','Setopul','madan','10'),

(30,'Anubhav','Sukadhara','sekwa','15'),

(40,'Jeny','KTM','TIWARIHOUSE','25'),

(50,'Shreeja','Kalanki','LONGROUTE','30');

insert into STD\_MARKS

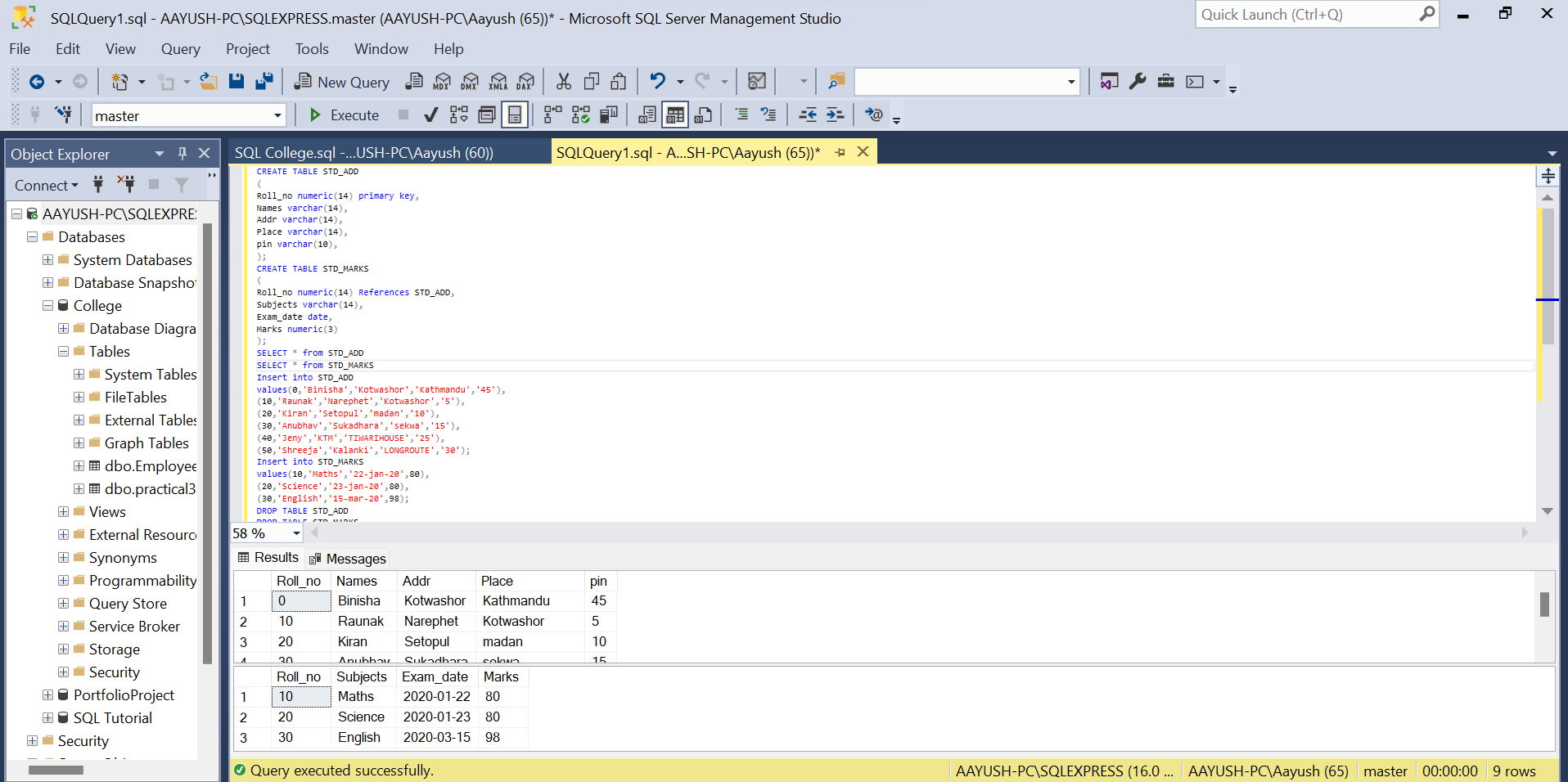
values(10,'Maths','22-jan-20',80),

(20,'Science','23-jan-20',80),

(30,'English','15-mar-20',98);

DROP TABLE STD\_ADD

DROP TABLE STD\_MARKS

****

**Result:**

We successfully implemented the primary key and foreign key concept in SQL. We created two tables, STD\_ADD and STD\_MARKS. The STD\_ADD table had a primary key column called customer\_id, while the STD\_MARKS table had a primary key column called Roll\_no and a foreign key column called Roll\_no that referred to the Roll\_no column in the STD\_ADD table. The FOREIGN KEY constraint specified that the values in the Roll\_no column in the STD\_MARKS table must match the values in the Roll\_no column in the STD\_ADD table. This ensured that each order was associated with a valid student.