0

| Experiment No.3 |
| --- |
| DDL & DML Commands |
| Date of Performance: 13/2/24 |
| Date of Submission: 20/2/24 |

**Experiment No. 3:** Create and populate database using Data Definition Language (DDL) and DML Commands for you‘re the specified System.

**Course Outcome [CSL503.2]:** Create database tables with different DDL and DML statements and apply integrity constraints

**Aim**: Create and populate database using Data Definition Language (DDL) and DML

Commands for you‘re the specified System.

**Theory**: DDL Commands

**Commands -**

create - creating database and table - DDL

desc - showing the table at the beginning

insert - inserting the data - DML

select - to display the updated table

update - update the existing column

alter - changes/modifications (eg. adding a column)

Operations in alter:

1. to add a column

2. to update the data

3. remove a specific column - using drop

4. modifying the data

rename - changing the name of the DB or table or column etc

Operations in rename

1. rename the column name

2. rename the table name

truncate - delete the entire records from the table

drop - drop/delete the entire table with structure

Integrety Constraints -

primary key after table is created

setting not null for a column

setting null value

setting unique key

setting foreign key

adding the integrity constraints at the same time

**Code:**

-- Creating the student table

CREATE TABLE Student\_Details (

std\_id INT(10),

std\_name CHAR(30),

branch CHAR(30)

);

DESC Student\_Details; -- show the structure in the table

-- Inserting the data

INSERT INTO Student\_Details(std\_id, std\_name, branch) VALUES

(29, "John Doe", "Computer Science"),

(30, "Jane Smith", "Computer Science");

SELECT \* FROM Student\_Details; -- show the data in the table

-- Updating / adding a column in the table

ALTER TABLE Student\_Details

ADD address VARCHAR(100);

-- Adding data in the updated column

UPDATE Student\_Details

SET address = "123 Main St"

WHERE std\_id IN (29, 30);

-- To remove a particular column

ALTER TABLE Student\_Details

DROP COLUMN address;

-- To modify anything related to column

ALTER TABLE Student\_Details

MODIFY std\_name CHAR(50);

-- To rename the column

ALTER TABLE Student\_Details

CHANGE COLUMN std\_id student\_id INT; -- existing , changing name

-- To rename the table name

RENAME TABLE Student\_Details

TO Students;

DESC Students; -- show the data in the table

-- Defining primary key after the table is created

ALTER TABLE Student\_Details

ADD PRIMARY KEY (student\_id);

SELECT \* FROM Student\_Details; -- show the data in the table

-- Setting a column as not null

ALTER TABLE Students

MODIFY std\_name CHAR(50) NOT NULL;

INSERT INTO Students(student\_id, std\_name, branch) VALUES

(31, "Sam Johnson", "Computer Science");

SELECT \* FROM Student\_Details; -- show the data in the table

-- Setting the column as unique

ALTER TABLE Students

MODIFY std\_name CHAR(50) UNIQUE;

INSERT INTO Students(student\_id, std\_name, branch) VALUES

(32, "John Doe", "Computer Science");

-- Creating another table

CREATE TABLE Faculty\_Details(

fac\_id INT(10),

fac\_name CHAR(30),

fac\_salary INT,

student\_id INT

);

ALTER TABLE Faculty\_Details

ADD PRIMARY KEY (fac\_id);

SELECT \* FROM Faculty\_Details;

-- Creating a foreign key

ALTER TABLE Faculty\_Details

ADD FOREIGN KEY (student\_id)

REFERENCES Students(student\_id);

-- Dropping and truncating

TRUNCATE TABLE Faculty\_Details;

SELECT \* FROM Faculty\_Details;

DROP TABLE Faculty\_Details;

-- Adding the integrity constraints at the same time

CREATE TABLE Table1(

id1 INT(20) UNIQUE,

name CHAR(30) NOT NULL,

PRIMARY KEY (id1)

);

CREATE TABLE Table2(

id2 INT(20) UNIQUE,

t2\_id INT(20),

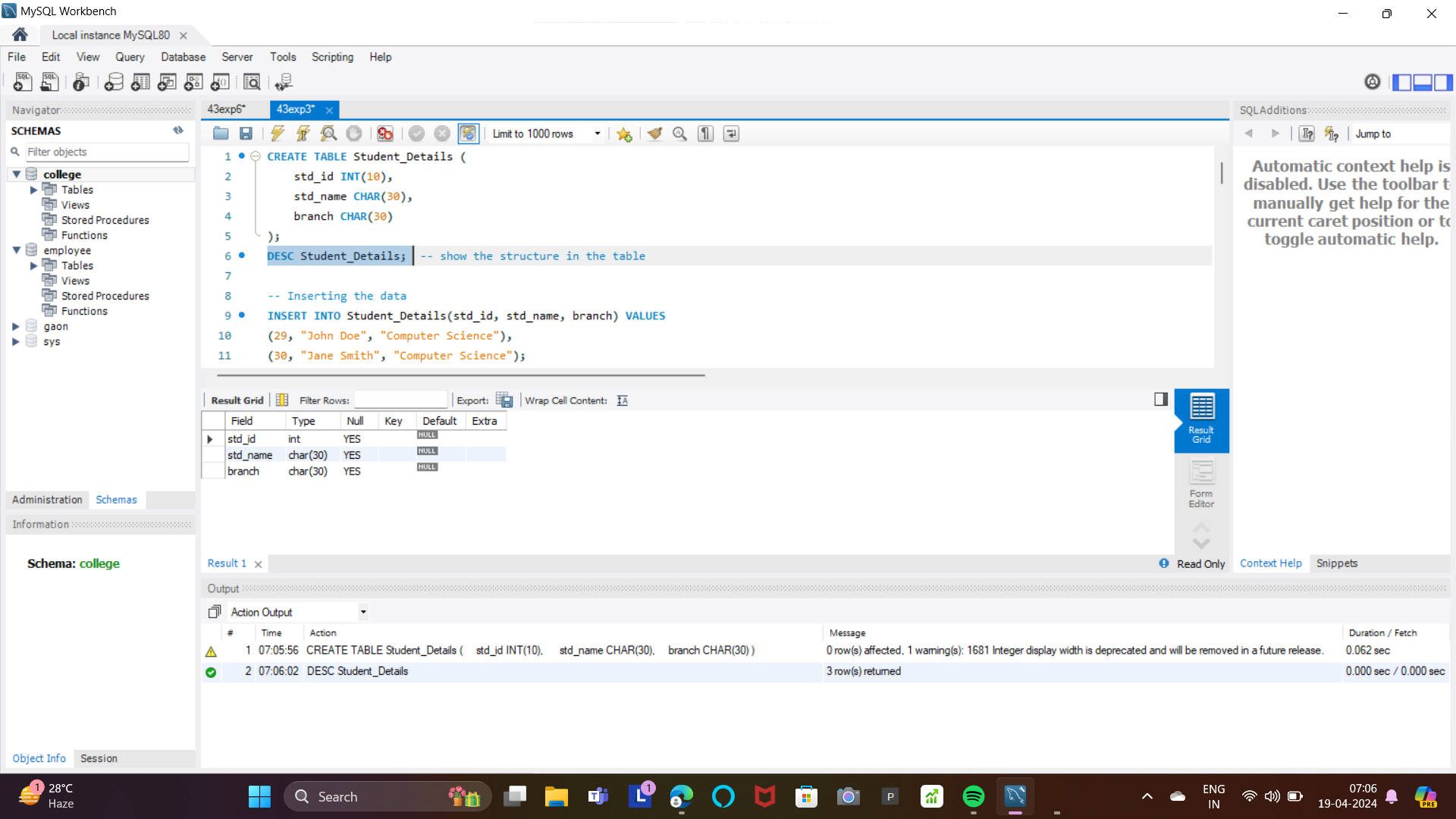
name CHAR(30) NOT NULL,

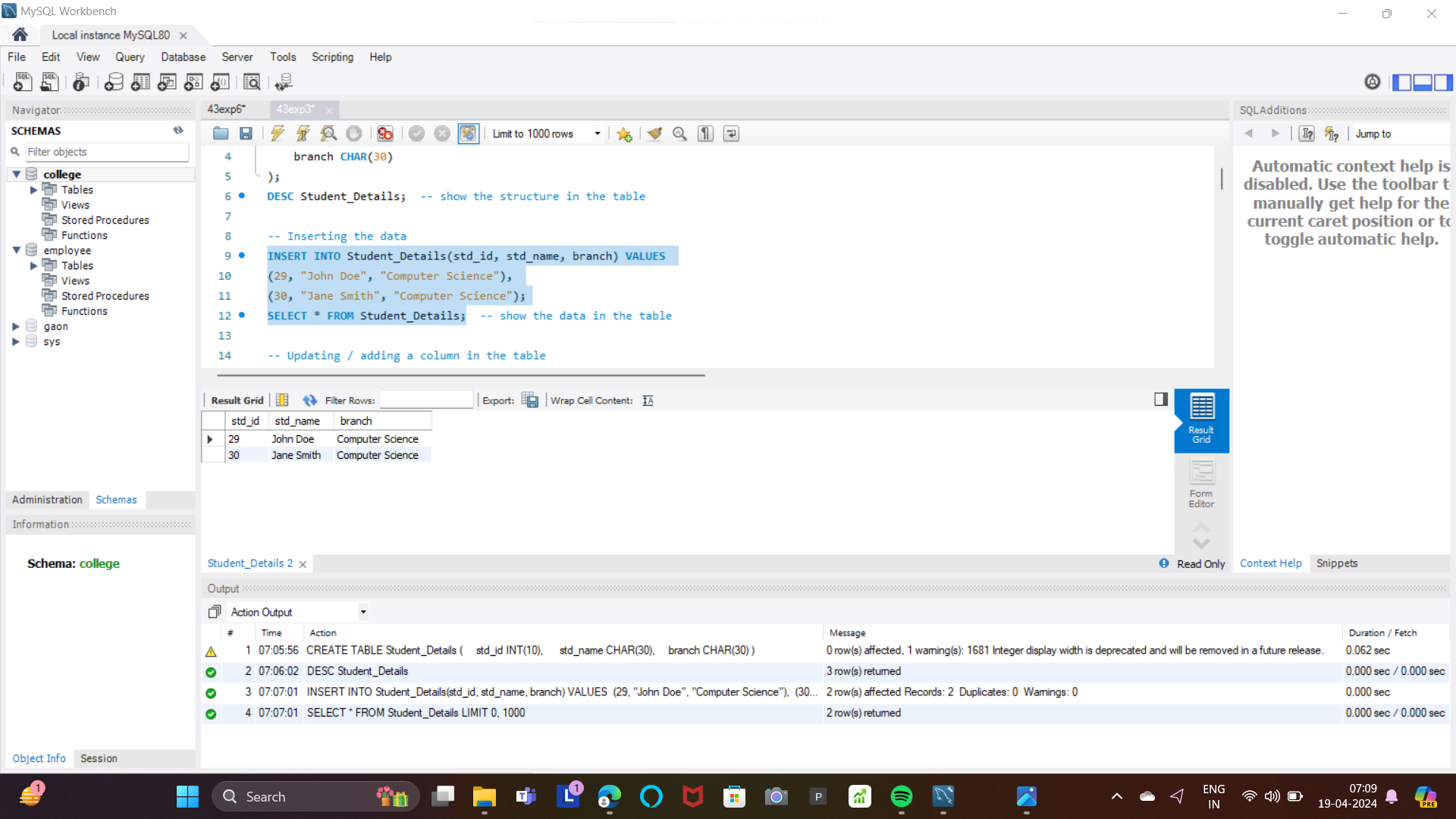
PRIMARY KEY (id2),

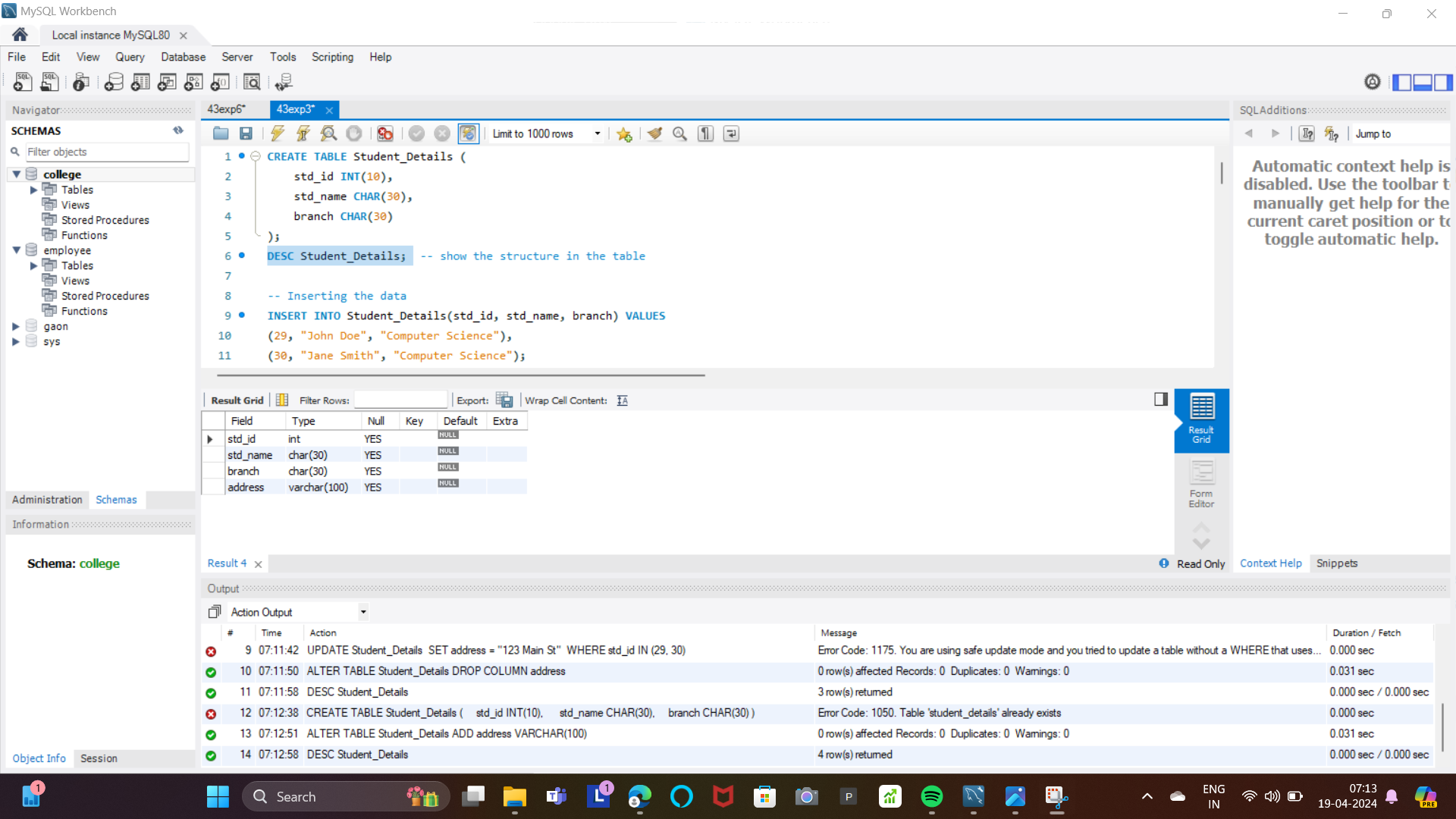
FOREIGN KEY (t2\_id) REFERENCES Table1(id1)

);

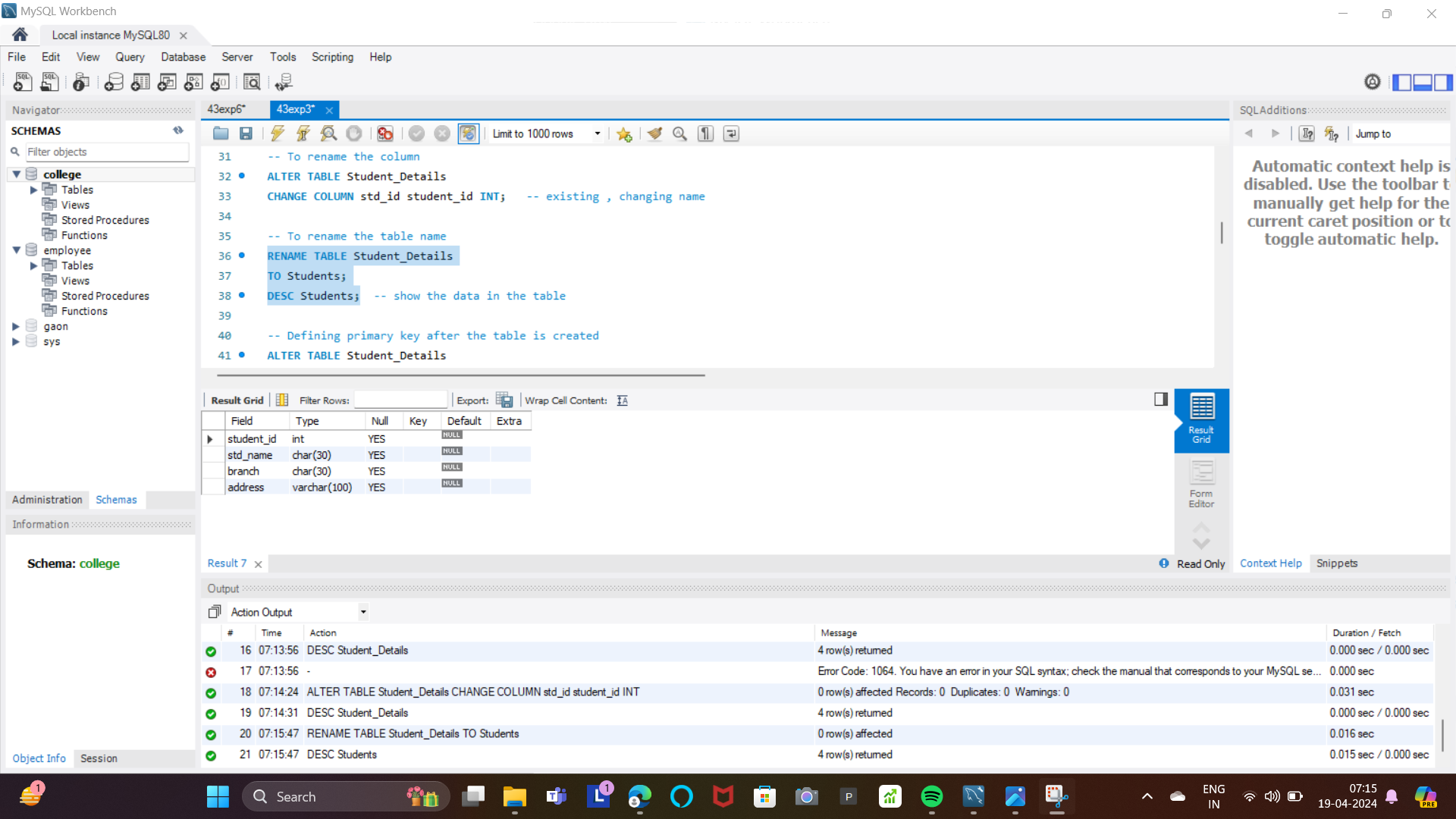
**Create and insert into a table :**

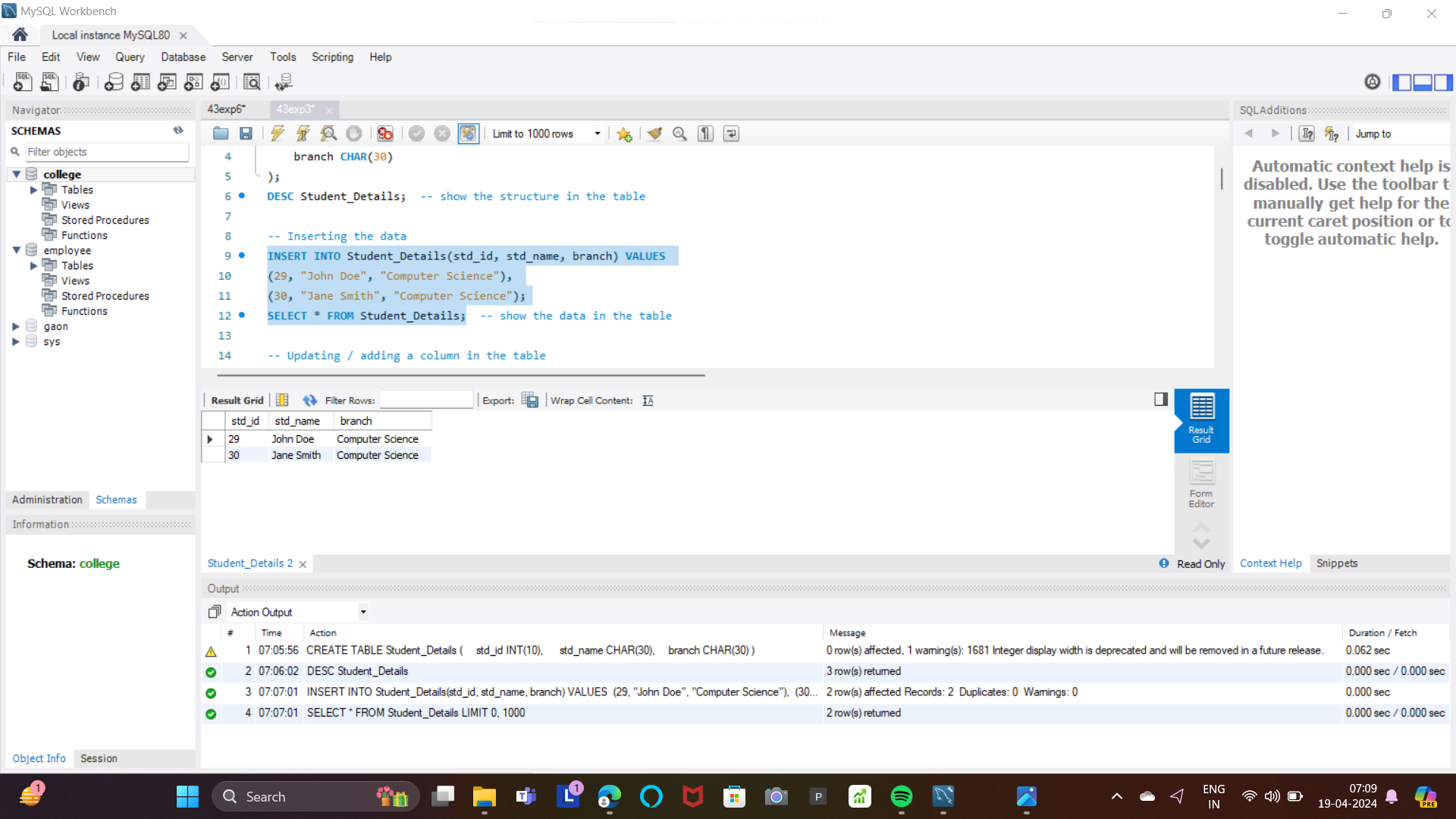


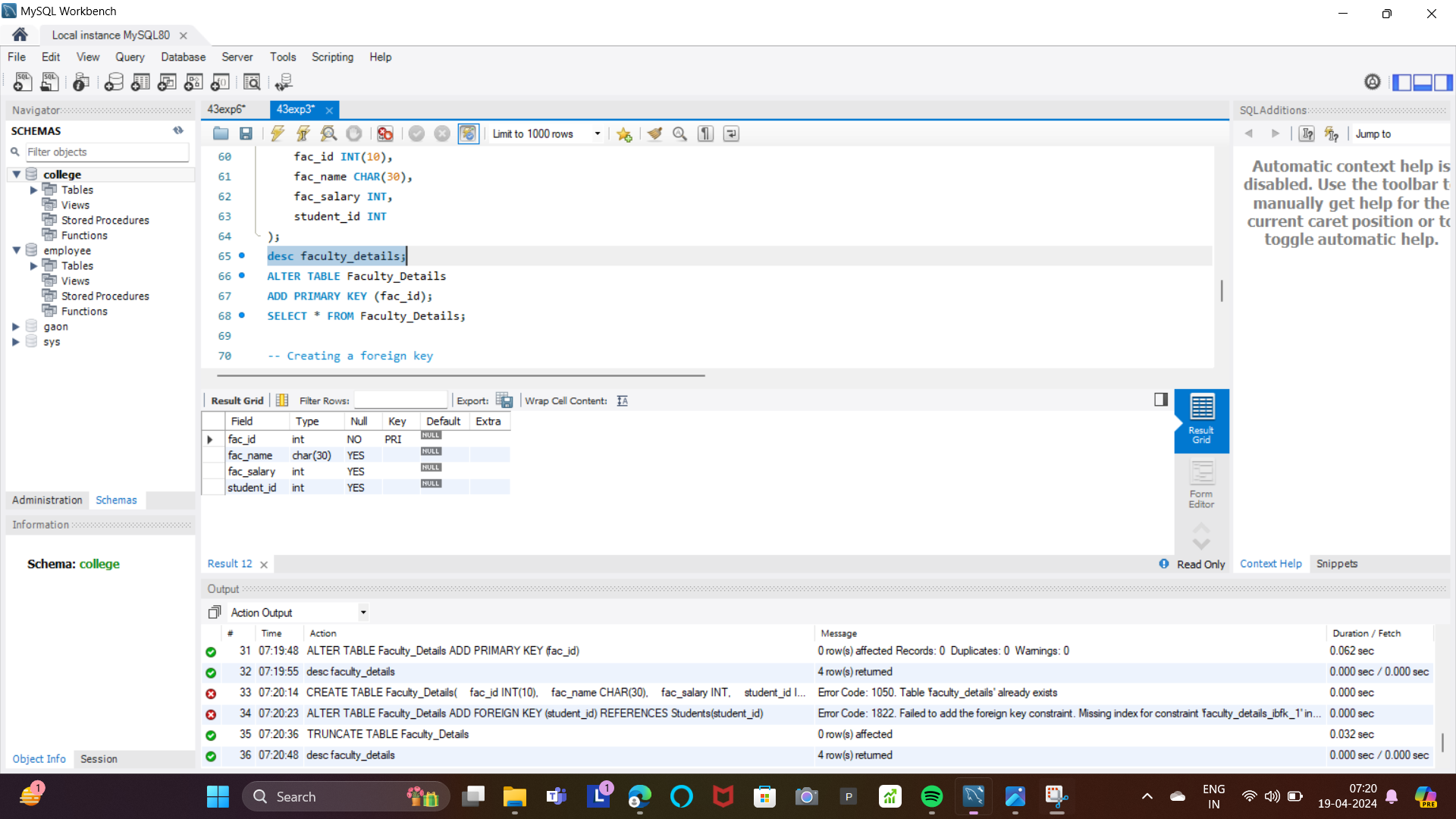
****

**Update the data in table/column:  
**

**Rename the table**

****

**Adding Primary & Foreign key while table creation:**

**Dropping & Truncating the Table:  
**

**CONCLUSION**:

The successful execution of Data Definition Language (DDL) commands ensured the creation, modification, and deletion of database objects according to the specified schema requirements. Likewise, the Data Manipulation Language (DML) commands effectively facilitated the retrieval, insertion, updating, and deletion of data within the database, demonstrating the system's operational integrity and adherence to business logic. This accomplishment underscores the essential role of DDL and DML commands in managing database structure and content, fostering a robust foundation for data-driven operations and decision-making processes.