

## Assignment 1

Name: Smith Chokshi

Banner ID: B00923763

Submitted at: 28<sup>th</sup> Feb 2023

### Problem 1

Before Normalization

#### 1) Conceptual Diagram

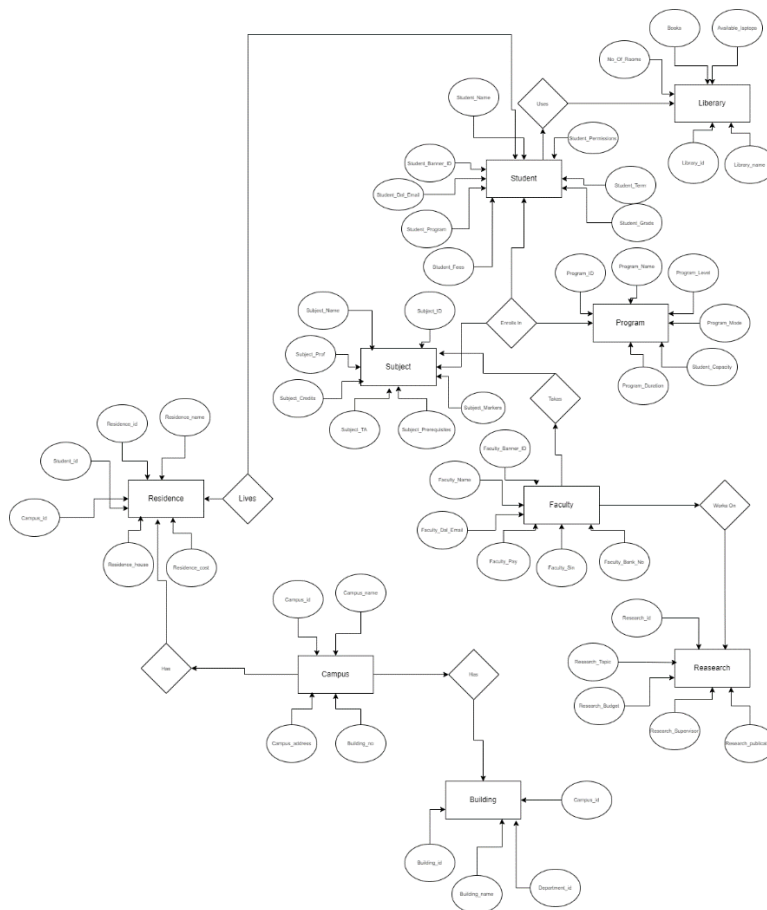


Figure 1 Conceptual Diagram [1]

## 2) Logical Diagram

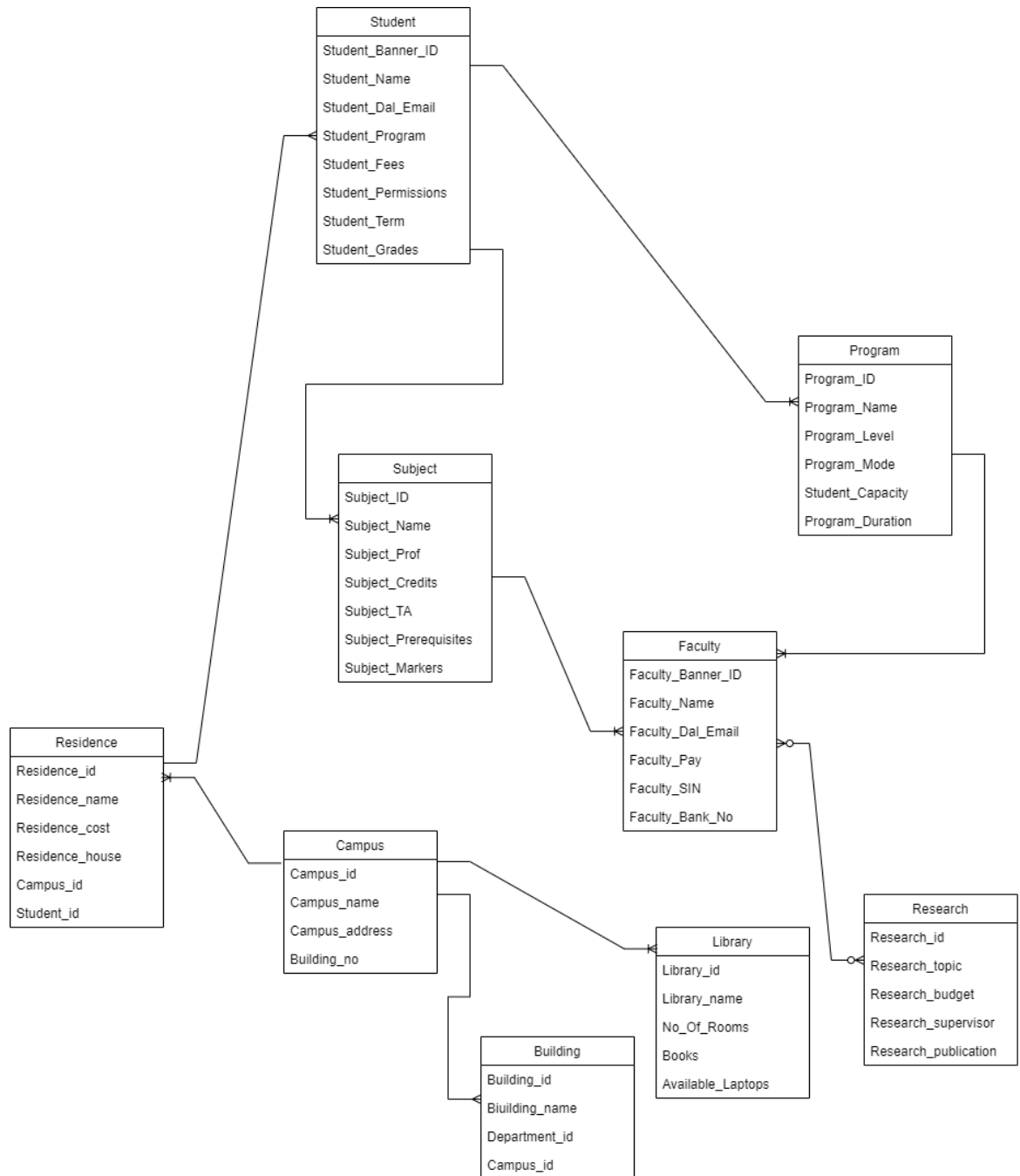


Figure 2 Logical Diagram [1]

After Normalization

### 3) Physical Model

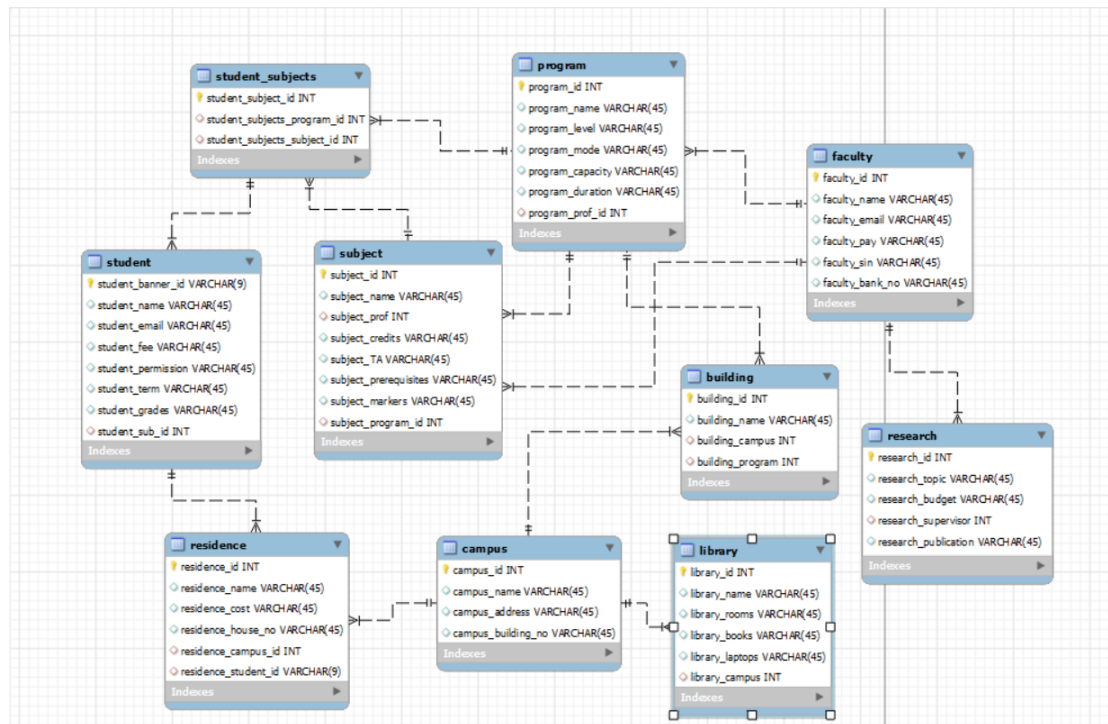


Figure 3 Physical Diagram [2]

- Normalized Students Entity

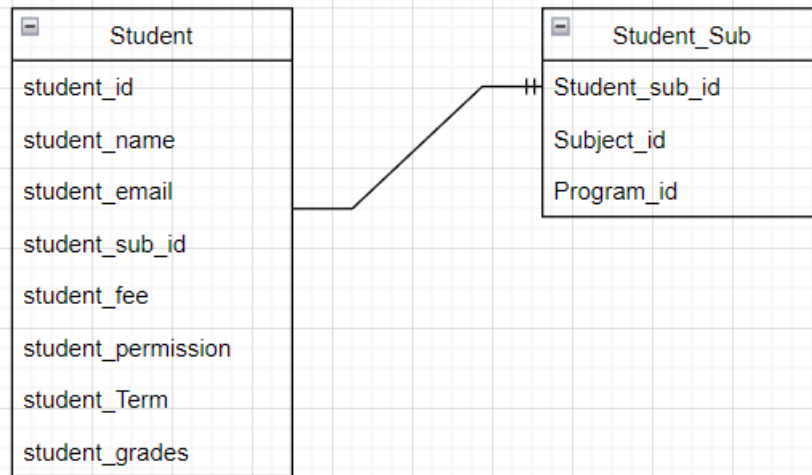


Figure 4 Normalization Table [1]

#### 4) Entities

##### 1) student

- student\_banner\_id PK
- student\_name
- student\_email
- student\_fee
- student\_permission
- student\_term
- student\_grades
- student\_sub\_id FK

##### 2) student\_subjects

- student\_subject\_id PK
- student\_subject\_program\_id FK
- student\_subjects\_subject\_id FK

##### 3) subject

- subject\_id PK
- subject\_name

- subject\_prof FK
- subject\_credits
- subject\_TA
- subject\_prerequisites
- subject\_markers
- subject\_program\_id FK

4) program

- program\_id PK
- program\_name
- program\_level
- program\_mode
- program\_capacity
- program\_duration
- program\_prof FK

5) faculty

- faculty\_id PK
- faculty\_name
- faculty\_email
- faculty\_pay
- faculty\_sin
- faculty\_bank\_no

6) research

- research\_id PK
- research\_topic
- research\_budget
- research\_supervisor FK
- research\_publication

7) building

- building\_id PK
- building\_name
- building\_campus FK
- building\_program FK

8) library

- library\_id PK

- library\_name
- library\_rooms
- library\_books
- library\_laptops
- library\_campus FK

#### 9) campus

- campus\_id PK
- campus\_name
- campus\_address
- campus\_building\_no

#### 10) residence

- residence\_id PK
- residence\_name
- residence\_cost
- residence\_house\_no
- residence\_campus\_id FK
- residence\_student\_id FK

### 5) Tables Created

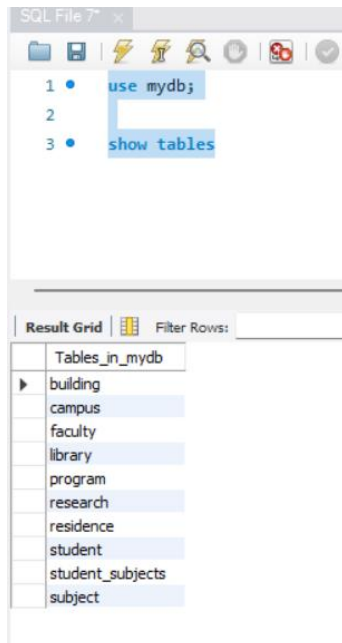
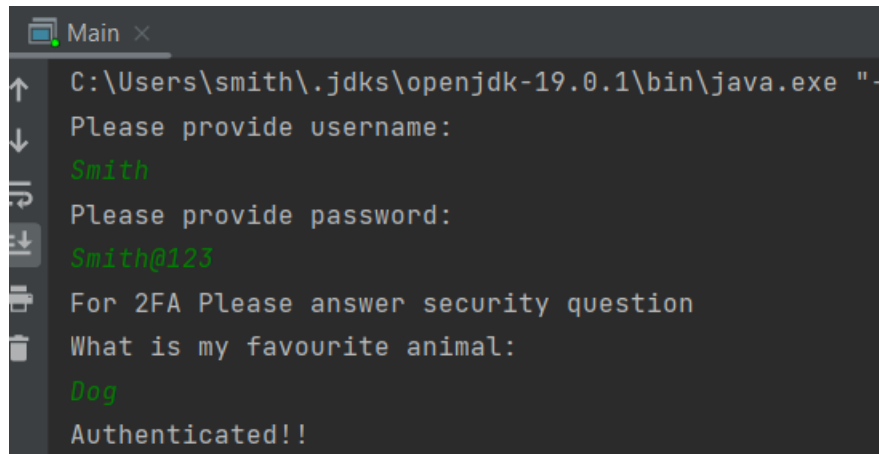


Figure 5 Table created [2]

## Problem 2

### Test Case

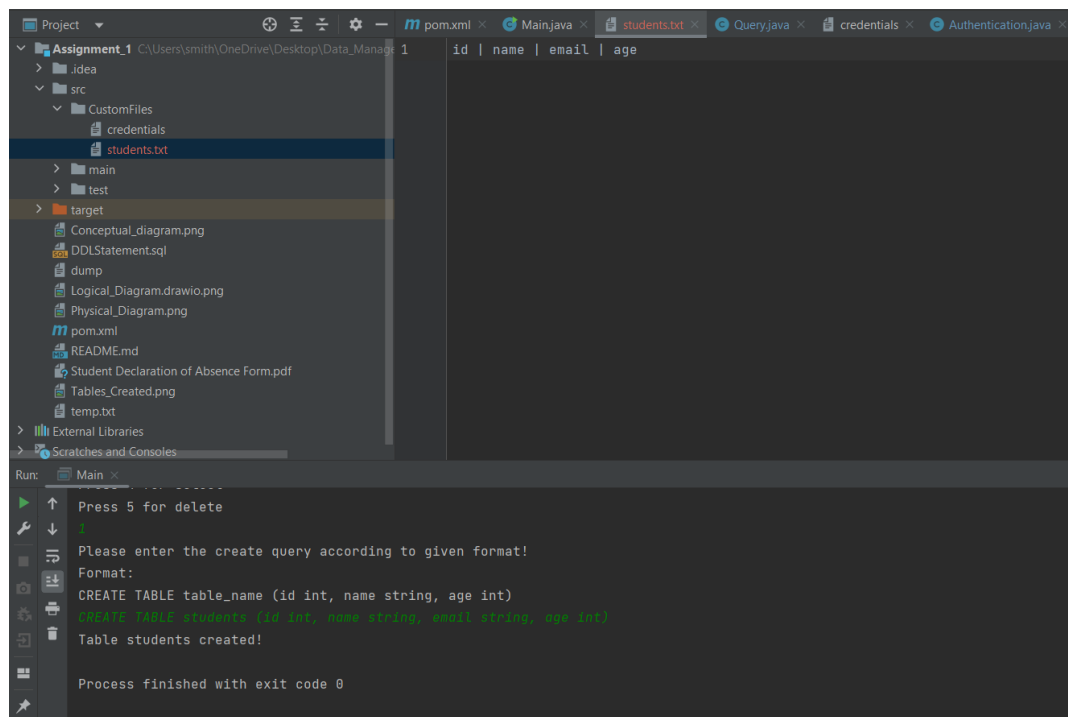
#### 1) 2 Factor Authentication



```
Main x
C:\Users\smith\.jdk\openjdk-19.0.1\bin\java.exe "-
Please provide username:
Smith
Please provide password:
Smith@123
For 2FA Please answer security question
What is my favourite animal:
Dog
Authenticated!!
```

Figure 6 2FA Testing

#### 2) Executing CREATE statement



```
Project
Assignment_1 C:\Users\smith\OneDrive\Desktop\Data_Manage 1
  .idea
  src
    CustomFiles
      credentials
      students.txt
    main
    test
    target
      Conceptual_diagram.png
      DDLStatement.sql
      dump
      Logical_Diagram.drawio.png
      Physical_Diagram.png
      pom.xml
      README.md
      Student Declaration of Absence Form.pdf
      Tables_Created.png
      temp.txt
  External Libraries
  Scratches and Consoles

Run: Main x
Press 5 for delete
5
Please enter the create query according to given format!
Format:
CREATE TABLE table_name (id int, name string, age int)
CREATE TABLE students (id int, name string, email string, age int)
Table students created!
Process finished with exit code 0
```

Figure 7 Create query testing

- Create statement is used to create a table under the name of students.
- A new file named students is created and column id, name, email, and age are created.

### 3) Executing INSERT statement

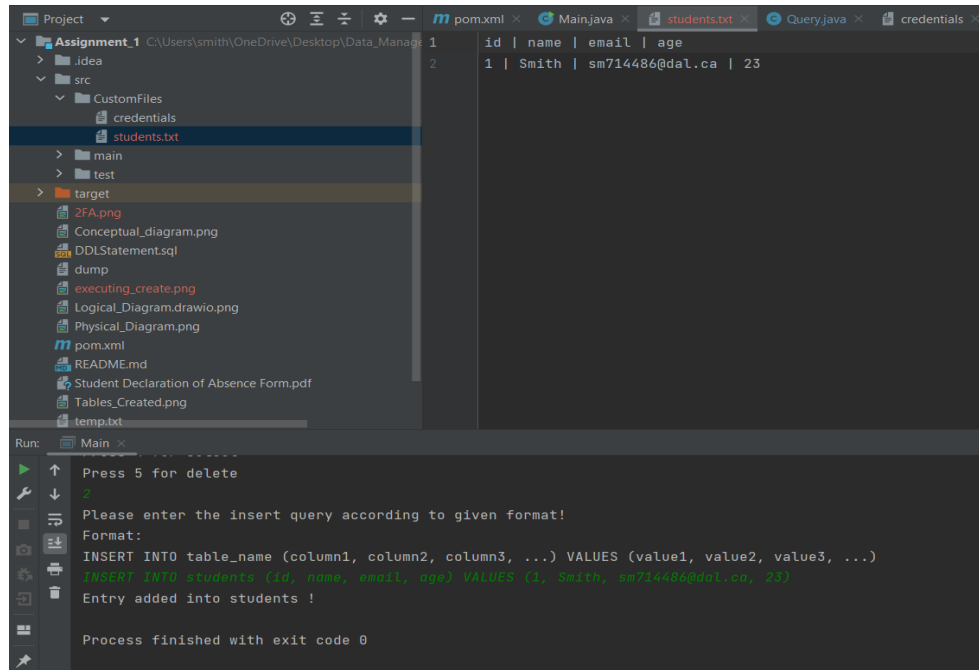


Figure 8 Insert query testing

- Insert statement is used to insert the data into students table.
- A new entry has been created into the students file.

### 4) Executing Update statement

- Before update query



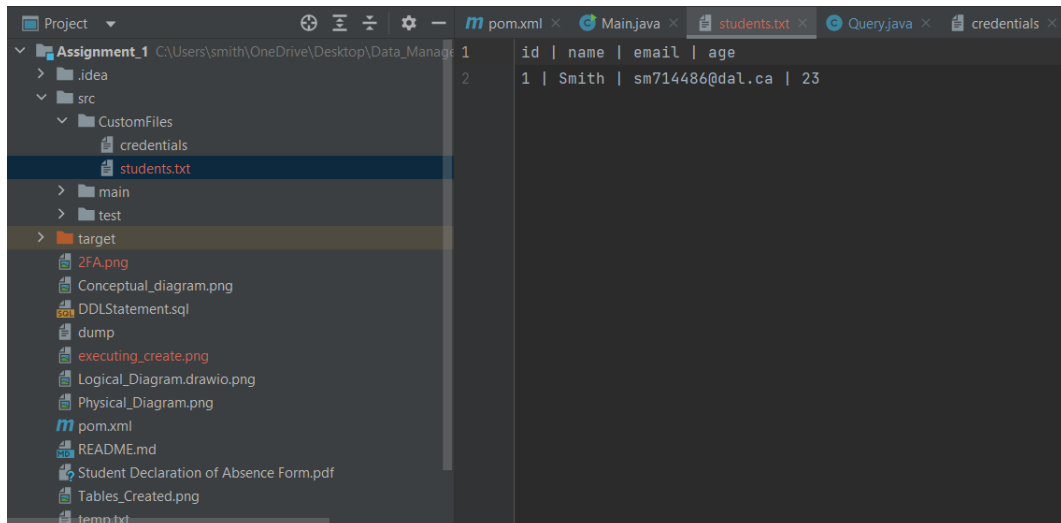


Figure 9 Before update query execution

After update query

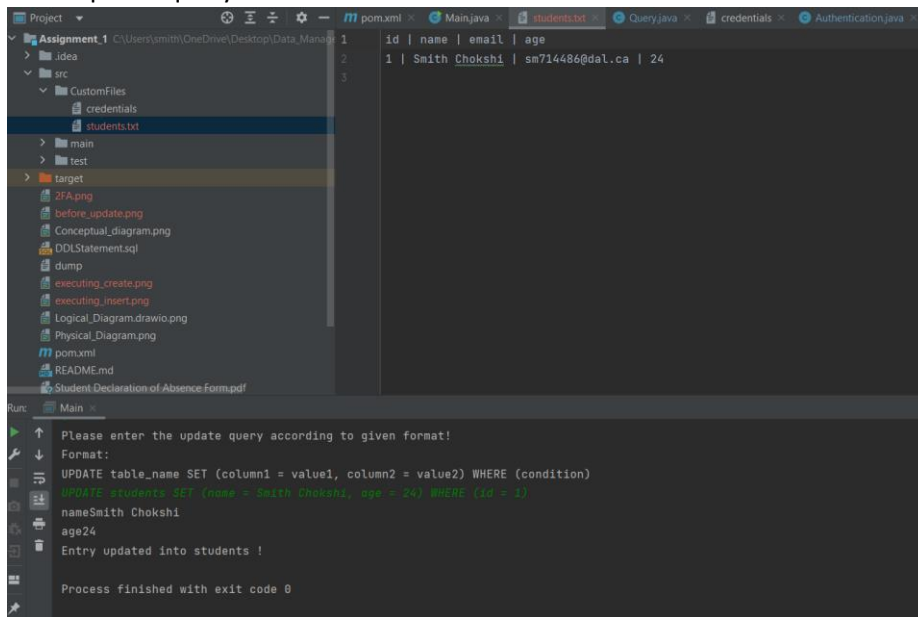
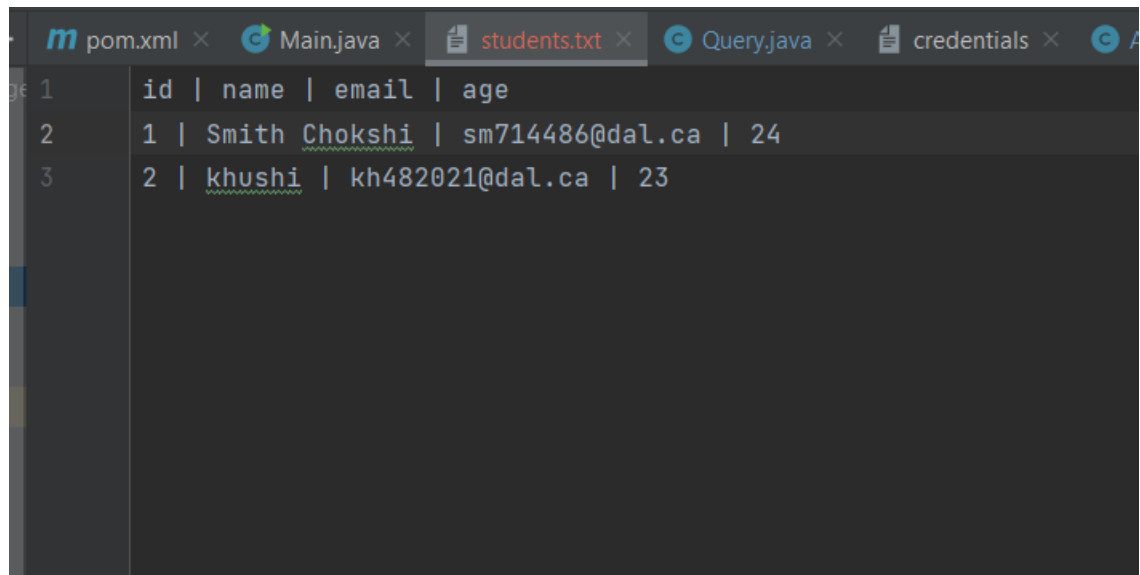


Figure 10 After update query execution test

## 5) Executing Delete statement

- Before Delete

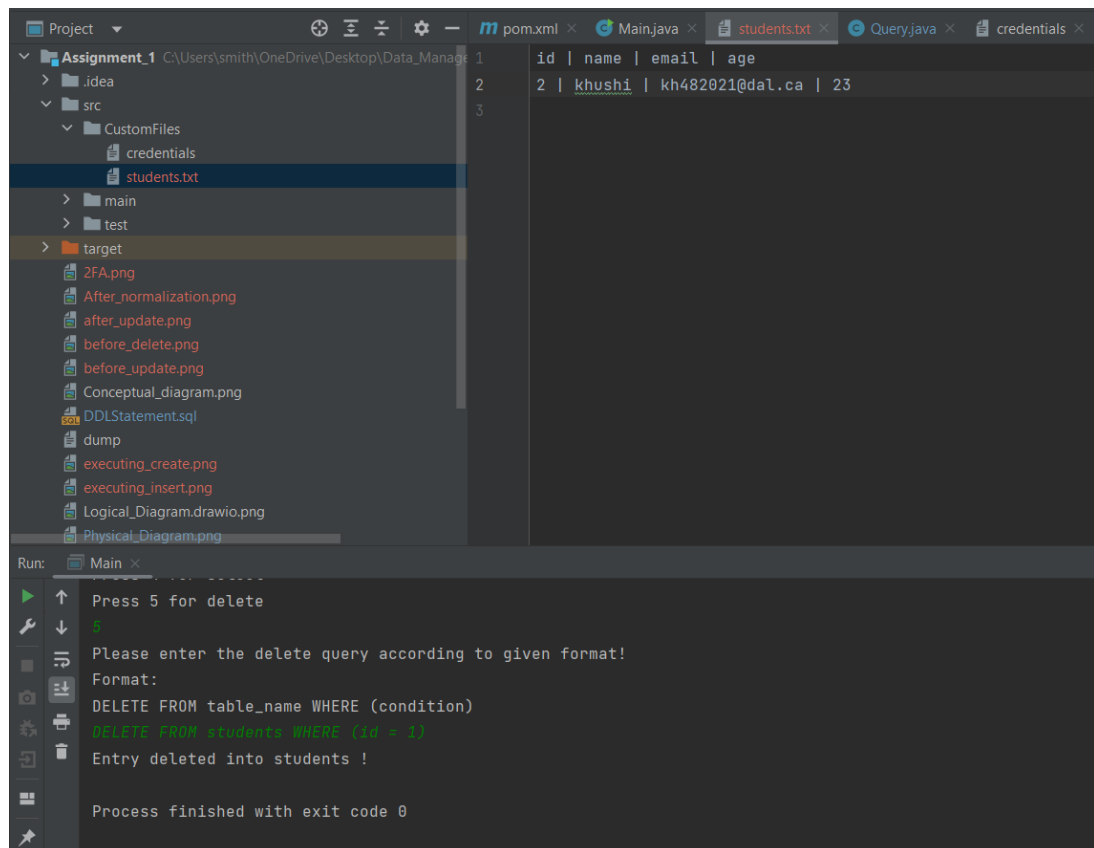


The screenshot shows an IDE with several tabs: pom.xml, Main.java, students.txt, Query.java, credentials, and A. The 'students.txt' tab is active, displaying a table with 4 columns: id, name, email, and age. The table contains 3 rows of data.

| id | name          | email           | age |
|----|---------------|-----------------|-----|
| 1  | Smith Chokshi | sm714486@dal.ca | 24  |
| 2  | khushi        | kh482021@dal.ca | 23  |

Figure 11 Before delete query execution

- After Delete



The screenshot shows the IDE after the deletion of the entry with id=1. The 'students.txt' tab now displays only 2 rows of data. The Run console at the bottom shows the execution of a Java application that prompts the user to press 5 for delete, displays the delete query, and confirms the deletion.

| id | name   | email           | age |
|----|--------|-----------------|-----|
| 2  | khushi | kh482021@dal.ca | 23  |

Run: Main

```
Press 5 for delete
5
Please enter the delete query according to given format!
Format:
DELETE FROM table_name WHERE (condition)
DELETE FROM students WHERE (id = 1)
Entry deleted into students !

Process finished with exit code 0
```

Figure 12 After delete query execution test

## 6) Executing Select statement

The screenshot shows an IDE with the following components:

- Project Explorer:** Shows a project named 'Assignment\_1' with a file tree including 'src', 'main', 'test', and 'target' directories. The 'students.txt' file is highlighted in the 'src' directory.
- Code Editor:** Displays the content of 'students.txt' as a CSV file with the following data:

| id | name   | email           | age |
|----|--------|-----------------|-----|
| 2  | khushi | kh482021@dal.ca | 23  |
- Run Console:** Shows the execution output of the 'Main' class. The output includes:

```
Please enter the select query according to given format!
Format:
SELECT * FROM table_name
SELECT * FROM students
id | name | email | age
2 | khushi | kh482021@dal.ca | 23
Process finished with exit code 0
```

Figure 13 Select query execution test

## Reference

- [1] "Diagrams.net - free flowchart maker and diagrams online," Flowchart Maker & Online Diagram Software. [Online]. Available: <https://app.diagrams.net>. [Accessed: 28-Feb-2023].
- [2] "MySQL Workbenchdownload now "," MySQL. [Online]. Available: <https://www.mysql.com/products/workbench/>. [Accessed: 28-Feb-2023].