

Prototype of a light-weight DBMS using Java programming language (no 3rd party libraries used).

IDE used: IntelliJ IDEA

JDK version: 19

JavaDocs specification for commenting styles, such as @param, @return etc.:

```
/**
 * This method is used to retrieve the list of attributes for a given table.
 *
 * @param tableName Name of the table.
 * @param emailAddress Email Address of the logged-in user.
 * @return returns the List<Attribute> object.
 */
2 usages  Vatsal Jain
public static List<Attribute> retrieveAttributes(String tableName, String emailAddress) {...}
```

Fig 2.1: JavaDocs comments example

Design principles used: I have used some of the SOLID design principles in my code. I have created an interface to handle the queries which has different implementations based on the type of the query entered by the user. I have created each module independently to reduce the coupling.

Application type: console-based

Project structure:

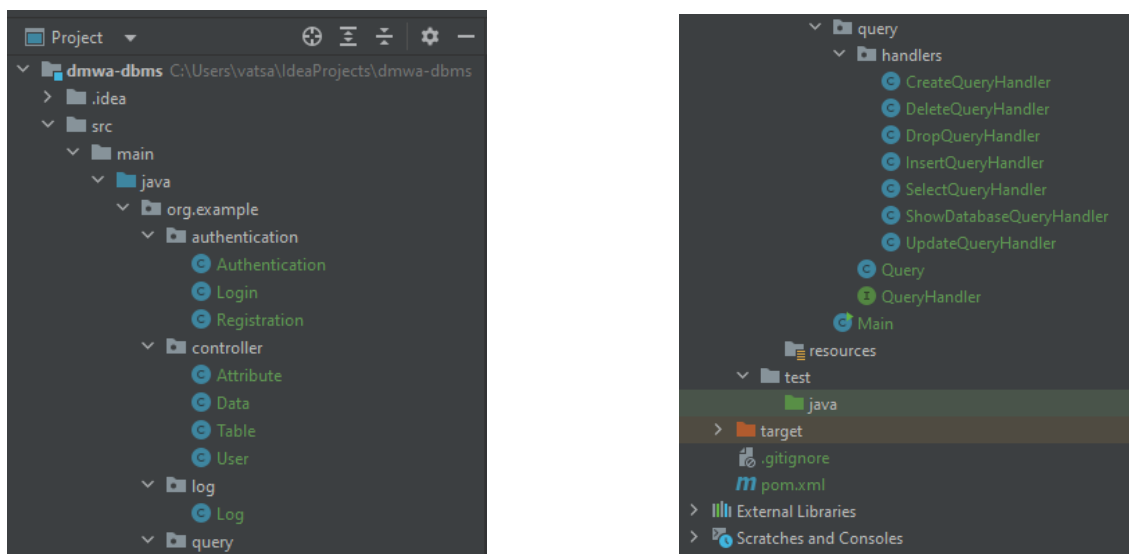
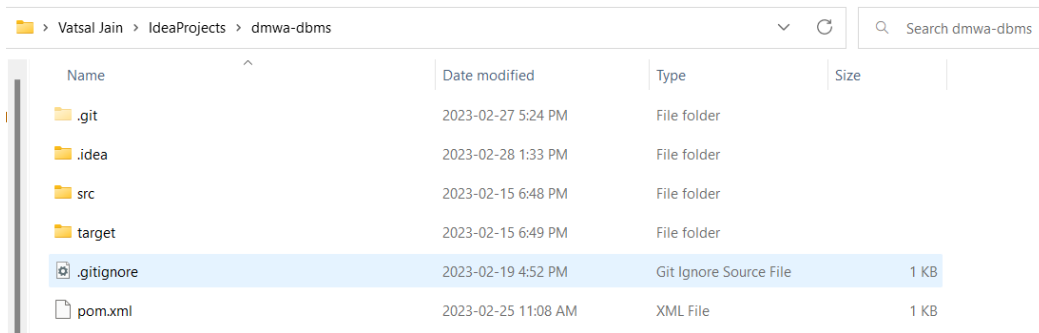


Fig 2.2: Project Structure

User Authentication:

1. User Registration:



Vatsal Jain > IdeaProjects > dmwa-dbms					Search dmwa-dbms
Name	Date modified	Type	Size		
.git	2023-02-27 5:24 PM	File folder			
.idea	2023-02-28 1:33 PM	File folder			
src	2023-02-15 6:48 PM	File folder			
target	2023-02-15 6:49 PM	File folder			
.gitignore	2023-02-19 4:52 PM	Git Ignore Source File	1 KB		
pom.xml	2023-02-25 11:08 AM	XML File	1 KB		

Fig 2.3.1: Root folder before the registration of a user

```
*****Welcome to the DBMS*****

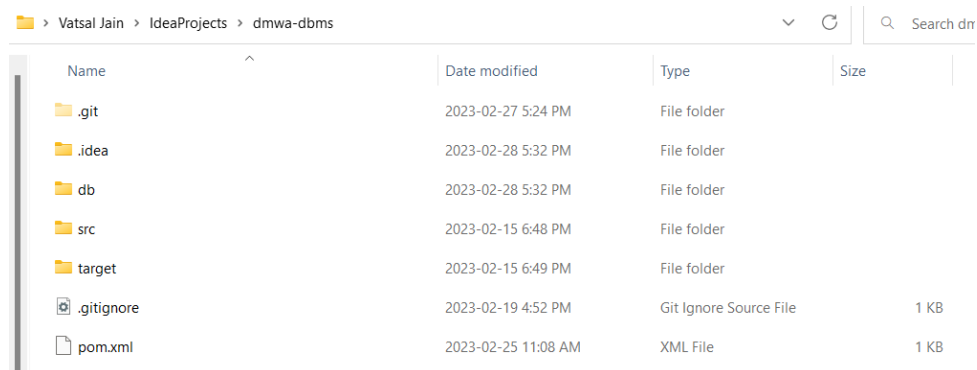
1. Log in
2. Register as a new user
3. Exit

Enter you choice: 2

Email address: vatsal@abc.com
Password: vatsal
First Name: Vatsal
Last Name: Jain
Enter a security question: Which city do you live in?
Enter your answer for the security question: Halifax

Vatsal Jain, you have been registered. Please log in now!
```

Fig 2.3.2: Console-based user registration



Vatsal Jain > IdeaProjects > dmwa-dbms					Search dmwa-dbms
Name	Date modified	Type	Size		
.git	2023-02-27 5:24 PM	File folder			
.idea	2023-02-28 5:32 PM	File folder			
db	2023-02-28 5:32 PM	File folder			
src	2023-02-15 6:48 PM	File folder			
target	2023-02-15 6:49 PM	File folder			
.gitignore	2023-02-19 4:52 PM	Git Ignore Source File	1 KB		
pom.xml	2023-02-25 11:08 AM	XML File	1 KB		

Fig 2.3.3: Root folder after the registration of a user

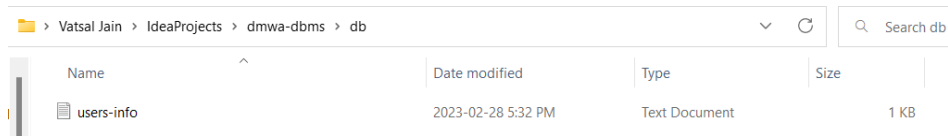


Fig 2.3.4: “db” directory has been created



Fig 2.3.5: “users-info.txt” file has been created to store the user details

- The password of the user has been stored after computing the hash code using the standard Java library MD5.

2. User Login

```
1. Log in
2. Register as a new user
3. Exit

Enter you choice: 1

Email address: vatsal@abc.com
Password: vatsal

Which city do you live in? Halifax
```

Fig 2.4: Console-based user login

Requirement 7(1):

Query Implementation:

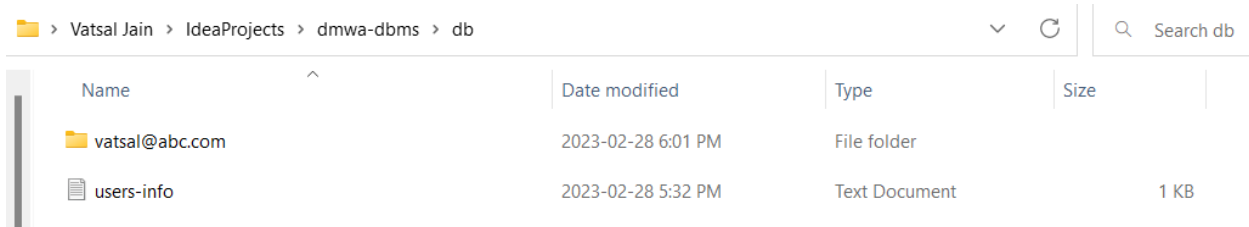
1. CREATE

```
1. Execute a query
2. Perform a transaction
3. Logout

Enter you choice: 1

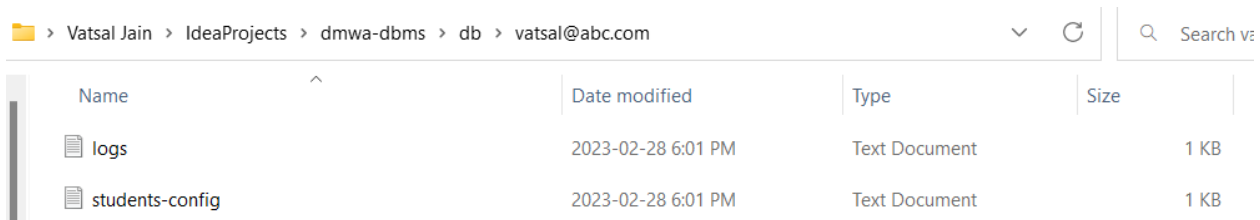
SQL Query: create table students (id int primary_key not_null, name varchar not_null, city varchar, program varchar, contact varchar, enrollment_date varchar, age int not_null);
Query executed successfully!
```

Fig 2.5.1: Query execution in console



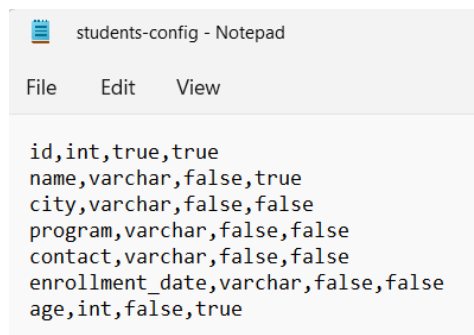
Name	Date modified	Type	Size
folder vatsal@abc.com	2023-02-28 6:01 PM	File folder	
file users-info	2023-02-28 5:32 PM	Text Document	1 KB

Fig 2.5.2: Database folder with user's id gets created after query execution



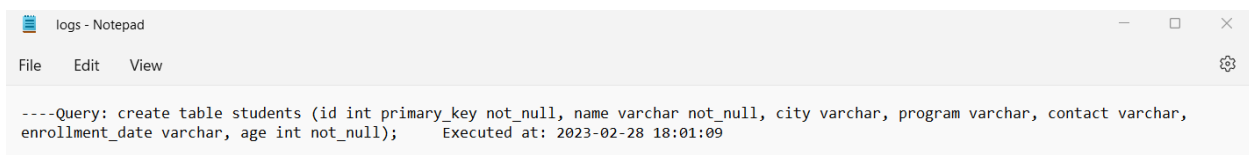
Name	Date modified	Type	Size
file logs	2023-02-28 6:01 PM	Text Document	1 KB
file students-config	2023-02-28 6:01 PM	Text Document	1 KB

Fig 2.5.3: Files for table configuration and storing the logs have been created



```
id,int,true,true
name,varchar,false,true
city,varchar,false,false
program,varchar,false,false
contact,varchar,false,false
enrollment_date,varchar,false,false
age,int,false,true
```

Fig 2.5.4: Table configuration file



```
----Query: create table students (id int primary_key not_null, name varchar not_null, city varchar, program varchar, contact varchar,
enrollment_date varchar, age int not_null);      Executed at: 2023-02-28 18:01:09
```

Fig 2.5.5: Log statements for the executed query

```
1. Execute a query
2. Perform a transaction
3. Logout

Enter you choice: 1

SQL Query: create table students (id int primary key not null, name varchar not null, city varchar, program varchar, contact varchar, enrollment_date varchar, age int not null)
Invalid syntax!

1. Execute a query
2. Perform a transaction
3. Logout
```

Fig 2.5.6: CREATE query syntax error

2. INSERT

```
SQL Query: insert into students values (1,Vatsal,Ahmedabad,MCS,9423355455,2022-24-05,21);
Query executed successfully!

1. Execute a query
2. Perform a transaction
3. Logout

Enter you choice: 1

SQL Query: insert into students values (2,Virat,Capetown,MACS,5465463233,2023-10-05,35);
Query executed successfully!
```

Fig 2.6.1: Query execution in console

```
students - Notepad

File Edit View

1,Vatsal,Ahmedabad,MCS,9423355455,2022-24-05,21
2,Virat,Capetown,MACS,5465463233,2023-10-05,35
3,Rohit,Mumbai,MCS,5465877846,2022-02-10,30
4,Rahul,Mumbai,MACS,5467575455,2021-29-09,40
5,Hardik,Jamaica,BACS,8745641564,2020-25-08,32
6,AB,Capetown,MACS,9427357539,2023-20-01,25
7,Chris,Mumbai,BACS,8745641564,2019-20-04,27
```

Fig 2.6.2: Rows inserted into the table

```
---Query: insert into students values (1,Vatsal,Ahmedabad,MCS,9423355455,2022-24-05,21); Executed at: 2023-02-28 18:09:14
---Query: insert into students values (2,Virat,Capetown,MACS,5465463233,2023-10-05,35); Executed at: 2023-02-28 18:09:20
---Query: insert into students values (3,Rohit,Mumbai,MCS,5465877846,2022-02-10,30); Executed at: 2023-02-28 18:10:29
---Query: insert into students values (4,Rahul,Mumbai,MACS,5467575455,2021-29-09,40); Executed at: 2023-02-28 18:10:34
---Query: insert into students values (5,Hardik,Jamaica,BACS,8745641564,2020-25-08,32); Executed at: 2023-02-28 18:10:40
---Query: insert into students values (6,AB,Capetown,MACS,9427357539,2023-20-01,25); Executed at: 2023-02-28 18:10:45
---Query: insert into students values (7,Chris,Mumbai,BACS,8745641564,2019-20-04,27); Executed at: 2023-02-28 18:10:52
```

Fig 2.6.3: Log statements for the executed query

```
1. Execute a query
2. Perform a transaction
3. Logout

Enter you choice: 1

SQL Query: insert into students values 4,Rahul,Mumbai,MACS,5467575455,2021-29-09,40;
Invalid syntax!

1. Execute a query
2. Perform a transaction
3. Logout

Enter you choice:
```

Fig 2.6.4: INSERT query syntax error

3. SELECT

```
SQL Query: select * from students;

id  name    city      program contact enrollment_date age
1   Vatsal  Ahmedabad MCS  9423355455 2022-24-05 21
2   Virat   Capetown  MACS   5465463233 2023-10-05 35
3   Rohit   Mumbai   MCS   5465877846 2022-02-10 30
4   Rahul   Mumbai   MACS   5467575455 2021-29-09 40
5   Hardik  Jamaica  BACS   8745641564 2020-25-08 32
6   AB      Capetown  MACS   9427357539 2023-20-01 25
7   Chris   Mumbai   BACS   8745641564 2019-20-04 27

Query executed successfully!
```

```
SQL Query: select city,name,age from students;

city  name    age
Vatsal Ahmedabad 21
Virat  Capetown 35
Rohit  Mumbai  30
Rahul  Mumbai  40
Hardik Jamaica 32
AB     Capetown 25
Chris  Mumbai  27

Query executed successfully!
```

```
SQL Query: select id from students;

id
1
2
3
4
5
6
7

Query executed successfully!
```

```
SQL Query: select * from students where program=MACS;
```

id	name	city	program	contact	enrollment_date	age
2	Virat	Capetown	MACS	5465463233	2023-10-05	35
4	Rahul	Mumbai	MACS	5467575455	2021-29-09	40
6	AB	Capetown	MACS	9427357539	2023-20-01	25

Query executed successfully!

```
SQL Query: select * from students where program=MACS and city=Capetown;
```

id	name	city	program	contact	enrollment_date	age
2	Virat	Capetown	MACS	5465463233	2023-10-05	35
6	AB	Capetown	MACS	9427357539	2023-20-01	25

Query executed successfully!

```
SQL Query: select city from students where program=BACS;
```

city
Jamaica
Mumbai

Query executed successfully!

```
SQL Query: select name,city,program from students where program=MACS;
```

name	city	program
Virat	Capetown	MACS
Rahul	Mumbai	MACS
AB	Capetown	MACS

Query executed successfully!

Fig 2.7.1: SELECT query console

```
----Query: select * from students; Executed at: 2023-02-28 18:17:09
----Query: select city,name,age from students; Executed at: 2023-02-28 18:17:55
----Query: select id from students; Executed at: 2023-02-28 18:18:29
----Query: select * from students where program=MACS; Executed at: 2023-02-28 18:19:03
----Query: select * from students where program=MACS and city=Capetown; Executed at: 2023-02-28 18:19:31
----Query: select city from students where program=BACS; Executed at: 2023-02-28 18:20:04
----Query: select name,city,program from students where program=MACS; Executed at: 2023-02-28 18:20:37
```

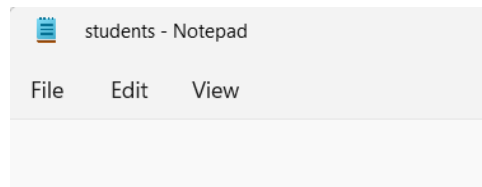
Fig 2.7.2: Log statements for the executed queries

```
SQL Query: select * from students where program=MACS & city=Capetown;
Invalid syntax!
```

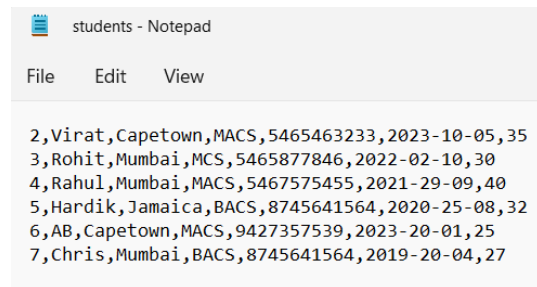
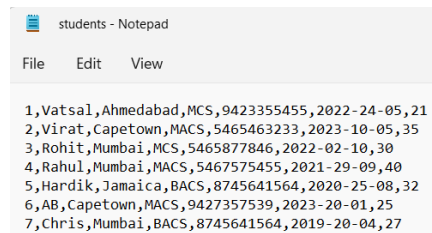
Fig 2.7.3: SELECT query syntax error

4. DELETE

```
SQL Query: delete from students;
Query executed successfully!
```

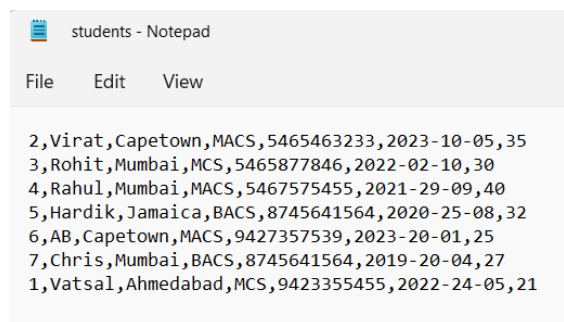


----Query: delete from students; Executed at: 2023-02-28 18:22:40



```
SQL Query: delete from students where city=Ahmedabad;
Query executed successfully!
```

----Query: delete from students where city=Ahmedabad; Executed at: 2023-02-28 18:32:59



```
SQL Query: delete from students where city=Ahmedabad and name=Vatsal;
Query executed successfully!
```



```
students - Notepad
File Edit View

2, Virat, Capetown, MACS, 5465463233, 2023-10-05, 35
3, Rohit, Mumbai, MCS, 5465877846, 2022-02-10, 30
4, Rahul, Mumbai, MACS, 5467575455, 2021-29-09, 40
5, Hardik, Jamaica, BACS, 8745641564, 2020-25-08, 32
6, AB, Capetown, MACS, 9427357539, 2023-20-01, 25
7, Chris, Mumbai, BACS, 8745641564, 2019-20-04, 27
```

----Query: delete from students where city=Ahmedabad and name=Vatsal; Executed at: 2023-02-28 18:36:33

Fig 2.8: DELETE query console, file changes and logs

5. UPDATE:

```
students - Notepad
File Edit View

1, Vatsal, Ahmedabad, MCS, 9423355455, 2022-24-05, 21
2, Virat, Capetown, MACS, 5465463233, 2023-10-05, 35
3, Rohit, Mumbai, MCS, 5465877846, 2022-02-10, 30
4, Rahul, Mumbai, MACS, 5467575455, 2021-29-09, 40
5, Hardik, Jamaica, BACS, 8745641564, 2020-25-08, 32
6, AB, Capetown, MACS, 9427357539, 2023-20-01, 25
7, Chris, Mumbai, BACS, 8745641564, 2019-20-04, 27
```

```
SQL Query: update students set program=MENG where program=MACS and city=Capetown;
Query executed successfully!
```

```
students - Notepad
File Edit View

2, Virat, Capetown, MENG, 5465463233, 2023-10-05, 35
3, Rohit, Mumbai, MCS, 5465877846, 2022-02-10, 30
4, Rahul, Mumbai, MACS, 5467575455, 2021-29-09, 40
5, Hardik, Jamaica, BACS, 8745641564, 2020-25-08, 32
6, AB, Capetown, MENG, 9427357539, 2023-20-01, 25
7, Chris, Mumbai, BACS, 8745641564, 2019-20-04, 27
1, Vatsal, Ahmedabad, MCS, 9423355455, 2022-24-05, 21
```

----Query: update students set program=MENG where program=MACS and city=Capetown; Executed at: 2023-02-28 20:32:21

```
students - Notepad
File Edit View

2, Virat, Capetown, MACS, 5465463233, 2023-10-05, 35
3, Rohit, Mumbai, MCS, 5465877846, 2022-02-10, 30
4, Rahul, Mumbai, MACS, 5467575455, 2021-29-09, 40
5, Hardik, Jamaica, BACS, 8745641564, 2020-25-08, 32
6, AB, Capetown, MACS, 9427357539, 2023-20-01, 25
7, Chris, Mumbai, BACS, 8745641564, 2019-20-04, 27
1, Vatsal, Ahmedabad, MCS, 9423355455, 2022-24-05, 21
```

```
SQL Query: update students set program=MENG,city=Delhi where program=MACS and city=Capetown;
Query executed successfully!
```

students - Notepad

File Edit View

```
2, Virat, Delhi, MENG, 5465463233, 2023-10-05, 35
3, Rohit, Mumbai, MCS, 5465877846, 2022-02-10, 30
4, Rahul, Mumbai, MACS, 5467575455, 2021-29-09, 40
5, Hardik, Jamaica, BACS, 8745641564, 2020-25-08, 32
6, AB, Delhi, MENG, 9427357539, 2023-20-01, 25
7, Chris, Mumbai, BACS, 8745641564, 2019-20-04, 27
1, Vatsal, Ahmedabad, MCS, 9423355455, 2022-24-05, 21
```

```
----Query: update students set program=MENG,city=Delhi where program=MACS and city=Capetown; Executed at: 2023-02-28 20:39:25
```

students - Notepad

File Edit View

```
1, Vatsal, Ahmedabad, MCS, 9423355455, 2022-24-05, 21
2, Virat, Capetown, MACS, 5465463233, 2023-10-05, 35
3, Rohit, Mumbai, MCS, 5465877846, 2022-02-10, 30
4, Rahul, Mumbai, MACS, 5467575455, 2021-29-09, 40
5, Hardik, Jamaica, BACS, 8745641564, 2020-25-08, 32
6, AB, Capetown, MACS, 9427357539, 2023-20-01, 25
7, Chris, Mumbai, BACS, 8745641564, 2019-20-04, 27
```

```
SQL Query: update students set program=MENG,city=Delhi where program=MACS;
Query executed successfully!
```

students - Notepad

File Edit View

```
1, Vatsal, Ahmedabad, MCS, 9423355455, 2022-24-05, 21
2, Virat, Delhi, MENG, 5465463233, 2023-10-05, 35
3, Rohit, Mumbai, MCS, 5465877846, 2022-02-10, 30
4, Rahul, Delhi, MENG, 5467575455, 2021-29-09, 40
5, Hardik, Jamaica, BACS, 8745641564, 2020-25-08, 32
6, AB, Delhi, MENG, 9427357539, 2023-20-01, 25
7, Chris, Mumbai, BACS, 8745641564, 2019-20-04, 27
```

```
----Query: update students set program=MENG,city=Delhi where program=MACS; Executed at: 2023-02-28 20:49:30
```

Fig 2.9: UPDATE query console, file changes and logs

6. DROP TABLE:

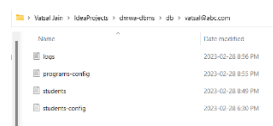


Fig 2.10: DROP query before



Fig 2.10.2: DROP query after

7. DROP DATABASE:

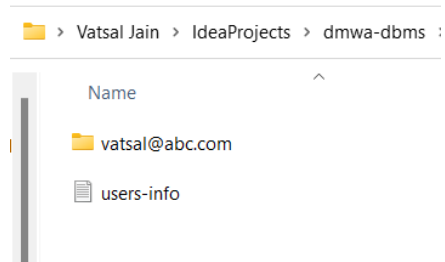


Fig 2.11.1: DROP query before

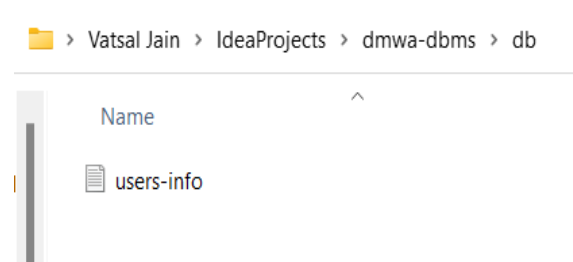


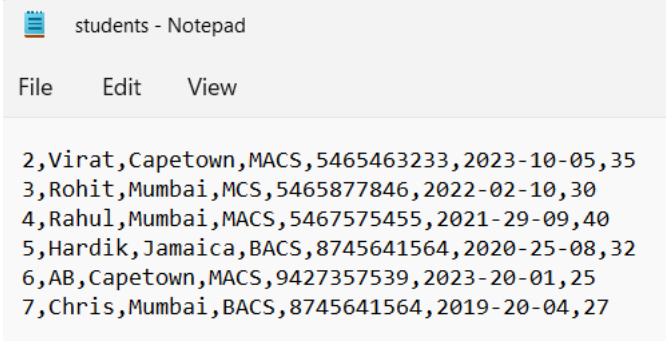
Fig 2.11.2: DROP query after

Transaction Implementation:

1. Successful transaction:

```
1. Execute a query
2. Perform a transaction
3. Logout

Enter you choice: 2
Transaction started! Enter 'commit' to end the transaction and 'rollback' to abort.
insert into students values (1,Vatsal,Ahmedabad,MCS,9423355455,2022-24-05,21);
insert into students values (2,Virat,Capetown,MACS,5465463233,2023-10-05,35);
insert into students values (3,Rohit,Mumbai,MCS,5465877846,2022-02-10,30);
insert into students values (4,Rahul,Mumbai,MACS,5467575455,2021-29-09,40);
insert into students values (5,Hardik,Jamaica,BACS,8745641564,2020-25-08,32);
insert into students values (6,AB,Capetown,MACS,9427357539,2023-20-01,25);
insert into students values (7,Chris,Mumbai,BACS,8745641564,2019-20-04,27);
delete from students where city=Ahmedabad;
commit;
Query executed successfully!
Query executed successfully!
Query executed successfully!
Query executed successfully!
Query executed successfully!
Query executed successfully!
Query executed successfully!
Query executed successfully!
```



```
2, Virat, Capetown, MACS, 5465463233, 2023-10-05, 35
3, Rohit, Mumbai, MCS, 5465877846, 2022-02-10, 30
4, Rahul, Mumbai, MACS, 5467575455, 2021-29-09, 40
5, Hardik, Jamaica, BACS, 8745641564, 2020-25-08, 32
6, AB, Capetown, MACS, 9427357539, 2023-20-01, 25
7, Chris, Mumbai, BACS, 8745641564, 2019-20-04, 27

----Query: insert into students values (1,Vatsal,Ahmedabad,MCS,9423355455,2022-24-05,21); Executed at: 2023-02-28 21:03:21
----Query: insert into students values (2, Virat, Capetown, MACS, 5465463233, 2023-10-05, 35); Executed at: 2023-02-28 21:03:21
----Query: insert into students values (3, Rohit, Mumbai, MCS, 5465877846, 2022-02-10, 30); Executed at: 2023-02-28 21:03:21
----Query: insert into students values (4, Rahul, Mumbai, MACS, 5467575455, 2021-29-09, 40); Executed at: 2023-02-28 21:03:21
----Query: insert into students values (5, Hardik, Jamaica, BACS, 8745641564, 2020-25-08, 32); Executed at: 2023-02-28 21:03:21
----Query: insert into students values (6, AB, Capetown, MACS, 9427357539, 2023-20-01, 25); Executed at: 2023-02-28 21:03:21
----Query: insert into students values (7, Chris, Mumbai, BACS, 8745641564, 2019-20-04, 27); Executed at: 2023-02-28 21:03:21
----Query: delete from students where city=Ahmedabad; Executed at: 2023-02-28 21:03:21
```

Fig 2.12: Successful Transaction

2. Failed Transaction:

```
Enter you choice: 2
Transaction started! Enter 'commit' to end the transaction and 'rollback' to abort.
insert into students values (1,Vatsal,Ahmedabad,MCS,9423355455,2022-24-05,21);
insert into students values (2, Virat, Capetown, MACS, 5465463233, 2023-10-05, 35);
insert into students values (3, Rohit, Mumbai, MCS, 5465877846, 2022-02-10, 30);
insert into students values (4, Rahul, Mumbai, MACS, 5467575455, 2021-29-09, 40);
insert into students values (5, Hardik, Jamaica, BACS, 8745641564, 2020-25-08, 32);
insert into students values (6, AB, Capetown, MACS, 9427357539, 2023-20-01, 25);
insert into students values (7, Chris, Mumbai, BACS, 8745641564, 201
update students set program=NENG where program=MACS and city=Capetown;
commit;
Query executed successfully!
Query executed successfully!
Query executed successfully!
Query executed successfully!
Query executed successfully!
Query executed successfully!
Invalid syntax!
```

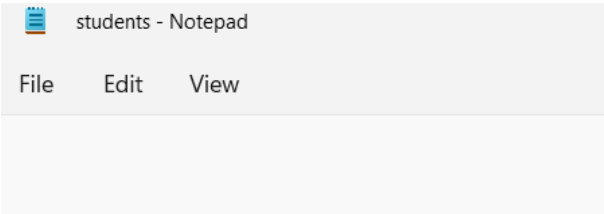


Fig 2.13: Failed Transaction

3. Transaction Rollback:

```
Enter you choice: 2
Transaction started! Enter 'commit' to end the transaction and 'rollback' to abort.
insert into students values (1,Vatsal,Ahmedabad,MCS,9423355455,2022-24-05,21);
insert into students values (2,Virat,CapeTown,MACS,5465463233,2023-10-05,35);
insert into students values (3,Rohit,Mumbai,MCS,5465677846,2022-02-10,30);
insert into students values (4,Rahul,Mumbai,MACS,5467575455,2021-29-09,40);
insert into students values (5,Hardik,Jamaica,BACS,8745641564,2020-25-08,32);
insert into students values (6,Ad,CapeTown,MACS,9427357539,2023-20-01,25);
update students set program=MENG where program=MACS and city=CapeTown;
rollback;
```

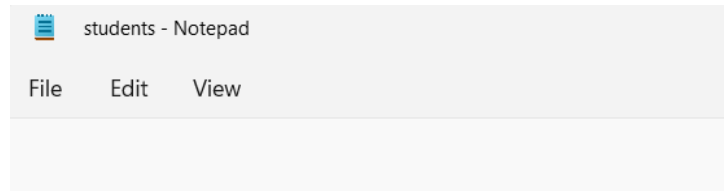


Fig 2.14: Transaction Rollback

Current state of the database:

```
SQL Query: show database;

Table: programs

Attribute  Datatype  Primary Key Not Null
-----
id         int       true       true
name       varchar   false      true
duration   varchar   false      false
number_of_courses int       false      false

Table: students

Attribute  Datatype  Primary Key Not Null
-----
id         int       true       true
name       varchar   false      true
city       varchar   false      false
program    varchar   false      false
contact    varchar   false      false
enrollment_date varchar   false      false
age        int       false      true

Query executed successfully!
```

```
----Query: show database;      Executed at: 2023-02-28 20:56:16
```

Fig 2.15: Current state of the database

References:

- “draw.io”, [Online]. Available: <https://app.diagrams.net/>. [Accessed: 02-Mar-2023].
- “Different ways of Reading a text file in Java”, [Online]. Available: <https://www.geeksforgeeks.org/different-ways-reading-text-file-java/>. [Accessed: 02-Mar-2023].
- “Regex Generator”, [Online]. Available: <https://regex-generator.olafneumann.org>. [Accessed: 02-Mar-2023].
- “Copying files from one directory to another in Java”, [Online]. Available: <https://stackoverflow.com/questions/1146153/copying-files-from-one-directory-to-another-in-java>. [Accessed: 02-Mar-2023].
- “A Solid Guide to SOLID Principles”, [Online]. Available: <https://www.baeldung.com/solid-principles>. [Accessed: 02-Mar-2023].
- “SOLID Principles — explained with examples”, [Online]. Available: <https://medium.com/mindorks/solid-principles-explained-with-examples-79d1ce114ace>. [Accessed: 02-Mar-2023].
- “MD5 hash in Java”, [Online]. Available: <https://www.geeksforgeeks.org/md5-hash-in-java/>. [Accessed: 02-Mar-2023].