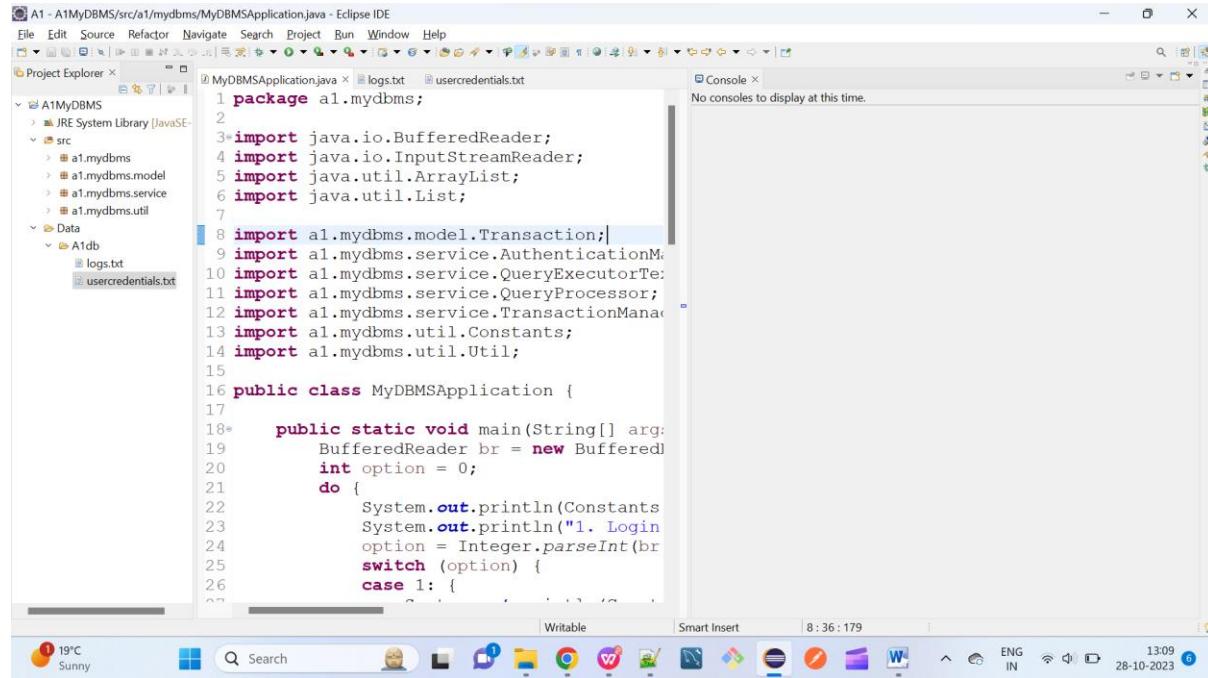


Application Structure and Format

As shown in below screenshots, ‘Data’ folder of my dbms application ‘A1MyDBMS’ will contain ‘A1db’ database. This database will contain logs and usercredentials tables by default, as they are system required tables. logs table is represented by logs.txt and it will store logs of the operations performed and usercredentials table is represented by usercredentials.txt which will store user’s login information.



The screenshot shows the Eclipse IDE interface. The Project Explorer view on the left displays a project named 'A1MyDBMS' with several source packages under 'src'. One package, 'a1.mydbms', contains three files: 'MyDBMSApplication.java', 'logs.txt', and 'usercredentials.txt'. The code editor window shows the content of 'MyDBMSApplication.java'. The code imports various classes from the 'a1.mydbms' package and other standard Java libraries. It defines a main method that reads input from a BufferedReader and processes it according to user options. The status bar at the bottom right shows the date and time as '28-10-2023 13:09'.

```
1 package a1.mydbms;
2
3 import java.io.BufferedReader;
4 import java.io.InputStreamReader;
5 import java.util.ArrayList;
6 import java.util.List;
7
8 import a1.mydbms.model.Transaction;
9 import a1.mydbms.service.AuthenticationManager;
10 import a1.mydbms.service.QueryExecutorManager;
11 import a1.mydbms.service.QueryProcessor;
12 import a1.mydbms.service.TransactionManager;
13 import a1.mydbms.util.Constants;
14 import a1.mydbms.util.Util;
15
16 public class MyDBMSApplication {
17
18    public static void main(String[] args) {
19        BufferedReader br = new BufferedReader(
20            new InputStreamReader(
21                System.in));
22        int option = 0;
23        do {
24            System.out.println(Constants.WELCOME_MESSAGE);
25            System.out.println("1. Login");
26            option = Integer.parseInt(br.readLine());
27            switch (option) {
28                case 1: {
29                    String id = Util.readInput("Enter User ID: ");
30                    String password = Util.readInput("Enter Password: ");
31                    AuthenticationManager authManager =
32                        TransactionManager.getTransactionManager()
33                            .getAuthenticationManager();
34                    if (authManager.authenticate(id, password)) {
35                        System.out.println("Login successful!");
36                    } else {
37                        System.out.println("Login failed!");
38                    }
39                }
40            }
41        } while (option != 0);
42    }
43}
```

Figure 1: Project database structure and default system tables

Tables structure

logs table

This table will store following 3 information:

1. Type of query performed i.e create, insert, delete or update
2. Name of the table on which query is executed
3. User id who executed the query

All these information will be stored in following format (i.e. separated by colon) :

querytype : tablename : userid
e.g. create table : student : Roshni

usercredentials table

This table will store user login details. For all the registered users, it will store following details:

1. Unique user id

2. Password in encrypted format

Both these information will be stored in following format (i.e. separated by space):

Userid EncryptedPassword

e.g. Roshni cec49593ca2c784c3ad8b3ea443ff642

User created tables

Consider following create table and insert query as an example to show how data is stored in user created tables.

- create table student (id int(5), name varchar(20), birthdate date(10), primary key (id));
- insert into student (id, name, birthdate) values (10, Roshni, 06/07/2023);

All the tables created by user will store data in following format.

1. **First Row** will contain details about all columns datatype and their size in following format. (i.e. datatype and size separated by space, columns data separated by pipe)

column1Datatype column1Size | column2Datatype column2Size | column3Datatype
column3Size |

e.g. int 5 | date 10 | varchar 20

2. **Second Row** will contain all the column names separated by pipe as shown in below.

column1Name | column2Name | column3Name |

e.g. id | birthdate | name

3. **All the other rows** after this will store user data separated by pipe as shown in below format:

column1Data | column2Data | column3Data |

e.g. 10 | 06/07/2023 | Roshni

In all the text files for user created tables, primary column will be added first.

e.g. PrimaryKeyColumn | Column1 | Column2 |

User authentication

- Before starting the application, both logs and usercredentials tables are empty as shown in below screenshots.

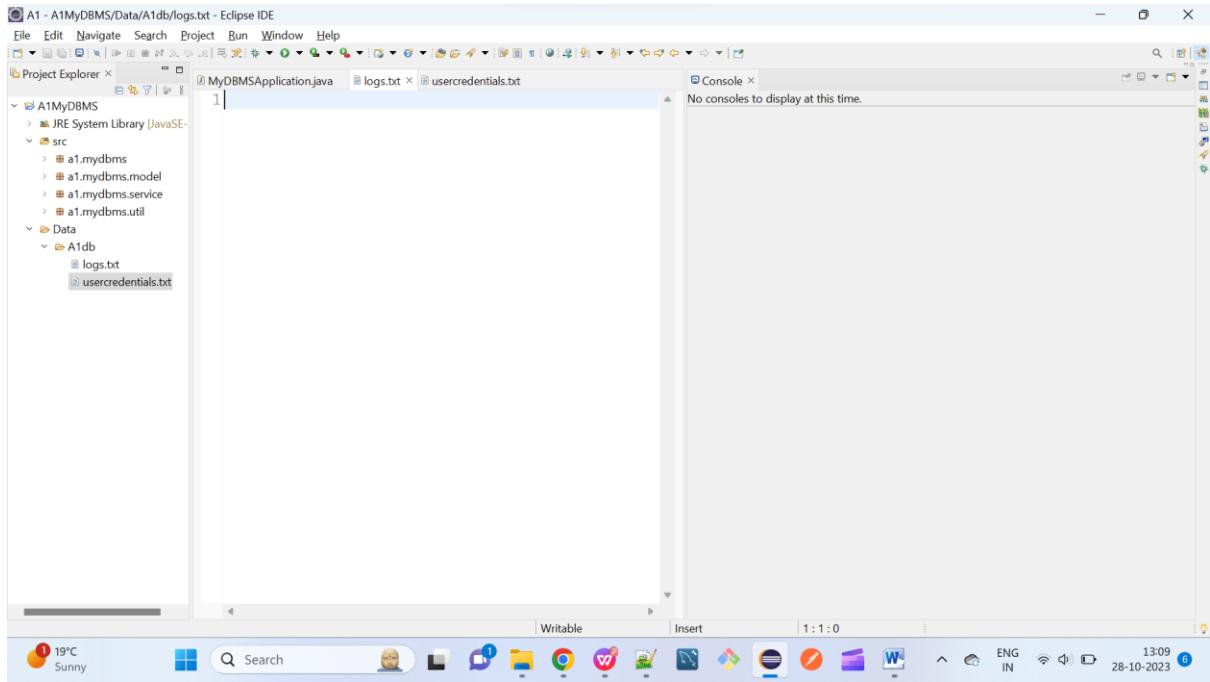


Figure 2: Empty logs table before starting application

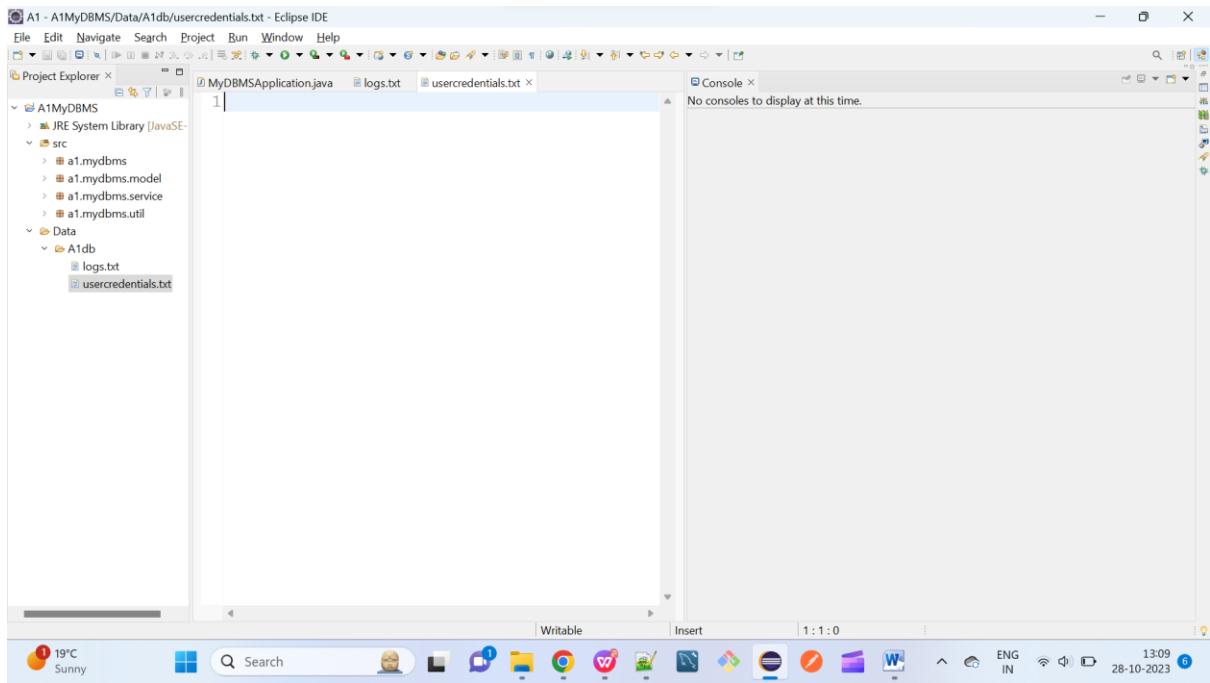


Figure 3: Empty usercredentials table before starting application

- Now, I have started the application and I am trying to login with userid and password. But, as I am not a registered user, this will throw error of ‘Invalid credentials’ as shown in below screenshot.

The screenshot shows the Eclipse IDE interface with a Java application named 'MyDBMSApplication'. In the Project Explorer, there is a 'src' folder containing packages like 'a1.mydbms', 'a1.mydbms.model', 'a1.mydbms.service', and 'a1.mydbms.util'. A 'logs.txt' file is also present. The 'MyDBMSApplication.java' file is open in the editor. The 'Console' tab shows the following interaction:

```

Please select from below option :
1. Login
2. Register
3. Logout
1

Enter userId:
Roshni
Enter password:
Joshi
Enter captcha (3aZ9Yta):
3aZ9Yta
[Invalid credentials]

Please select from below option :
1. Login
2. Register
3. Logout

```

Figure 4: Non registered user trying to login resulting into error

- So, now I selected the option 2 and I registered using userid and password. My registration is successful as shown by ‘Registered’ message in below screenshot. Also, my userid and encrypted password are inserted into usercredentials table as shown in below screenshot.

The screenshot shows the Eclipse IDE interface with the same Java application. The 'Console' tab shows the following interaction after selecting option 2:

```

1
2

Enter password:
Joshi
Enter captcha (3aZ9Yta):
3aZ9Yta
[Invalid credentials]

Please select from below option :
1. Login
2. Register
3. Logout
2

Enter new userId:
Roshni
Enter new password:
Joshi
Enter captcha (oo1mQwo):
oo1mQwo
Registered

Please select from below option :
1. Login
2. Register
3. Logout

```

Figure 5: Registration successful and data inserted into usercredentials table

- Now, I am trying to login again after registration, but by providing incorrect password. In this case, system will again throw error message of ‘Invalid credentials’ and will not allow me to login.

```

A1 - A1MyDBMS/Data/A1db/usercredentials.txt - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Project Explorer X MyDBMSApplication.java logs.txt usercredentials.txt X
A1MyDBMS
  JRE System Library [JavaSE-...
  src
    a1.mydbms
    a1.mydbms.model
    a1.mydbms.service
    a1.mydbms.util
Data
  A1db
    logs.txt
    usercredentials.txt

Console X
MyDBMSApplication [Java Application] D:\Eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.l...
Enter new password:
Joshi
Enter captcha (oo1mQwo):
oo1mQwo
Registered

Please select from below option :
1. Login
2. Register
3. Logout
1

Enter userId:
Roshni
Enter password:
test
Enter captcha (uhtYFMq):
uhtYFMq
Invalid credentials

Please select from below option :
1. Login
2. Register
3. Logout

```

Figure 6: Trying to login again after registration but with incorrect password

- **Captcha Verification :** In below screenshot, I am trying to login with correct userid and password, but have provided the invalid captcha. In this case, system will not allow me to login and throw ‘Invalid Captcha’ message.

```

A1 - A1MyDBMS/Data/A1db/usercredentials.txt - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Project Explorer X MyDBMSApplication.java logs.txt usercredentials.txt X
A1MyDBMS
  JRE System Library [JavaSE-...
  src
    a1.mydbms
    a1.mydbms.model
    a1.mydbms.service
    a1.mydbms.util
Data
  A1db
    logs.txt
    usercredentials.txt

Console X
MyDBMSApplication [Java Application] D:\Eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.l...
Enter password:
test
Enter captcha (uhtYFMq):
uhtYFMq
Invalid credentials

Please select from below option :
1. Login
2. Register
3. Logout
1

Enter userId:
Roshni
Enter password:
Joshi
Enter captcha (sQmzo2Y):
egdggd
Invalid captcha

Please select from below option :
1. Login
2. Register
3. Logout

```

Figure 7: Trying to login using incorrect captcha

- After that, I have provided correct userid, password and captcha to login. In this case, I got successfully logged in, as system has started providing me the option to enter queries as shown in below screenshot.

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS".
- MyDBMSApplication.java:** The current file being edited.
- Console:** Displays the following interaction:


```

Enter userId:
Roshni
Enter password:
Joshi
Enter captcha (sQmzO2Y):
egdggd
Invalid captcha

Please select from below option :
1. Login
2. Register
3. Logout
1

Enter userId:
Roshni
Enter password:
Joshi
Enter captcha (HnbVqtT):
HnbVqtT

Press 'stop' to exit
Enter Query:
      
```
- System Tray:** Shows the date (28-10-2023), time (13:13), and system status (ENG IN).

Figure 8: Successfully logged in

Create database

Below screenshot demonstrates that system will not allow to create new databases. Only one database will be used for all tables. Also, for this query, logs file will not be updated as it is not allowed to create new databases.

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS".
- MyDBMSApplication.java:** The current file being edited.
- Console:** Displays the following interaction:


```

Joshi
Enter captcha (sQmzO2Y):
egdggd
Invalid captcha

Please select from below option :
1. Login
2. Register
3. Logout
1

Enter userId:
Roshni
Enter password:
Joshi
Enter captcha (HnbVqtT):
HnbVqtT

Press 'stop' to exit
Enter Query:
create database test;
New database cannot be created. Please use
A1db only.

Enter Query:
      
```
- System Tray:** Shows the date (28-10-2023), time (13:14), and system status (ENG IN).

Figure 9: User trying to create new database

Create Table Query

- Now, I am executing the create table query for student table and query is successfully executed as shown by success message in console. Also, student.txt is created in A1db database and logs file is updated for create query as shown in below screenshot.

The screenshot shows the Eclipse IDE interface. In the Project Explorer, there is a Java project named 'A1MyDBMS' containing a package 'src' with classes 'a1.mydbms', 'a1.mydbms.model', 'a1.mydbms.service', and 'a1.mydbms.util'. Inside 'Data/A1db', there are files 'logs.txt', 'student.txt', and 'usercredentials.txt'. In the editor, 'MyDBMSApplication.java' contains the following code:

```
1create table : student : Roshni
2|
```

In the Console window, the output is:

```
Please select from below option :
1. Login
2. Register
3. Logout
1

Enter userId:
Roshni
Enter password:
Joshi
Enter captcha (HnbVqtT):
HnbVqtT

Press 'stop' to exit
Enter Query:
create database test;
New database cannot be created. Please use
A1db only.

Enter Query:
create table student (id int(5), name varchar
(20), birthdate date(10), primary key (id));
student: Table Created

Enter Query:
```

Figure 10: Successfully created student table, log file updated

Below screenshot shows the content of student.txt file. It will contain columns and its metadata in the format as mentioned in ‘Table structures’ section.

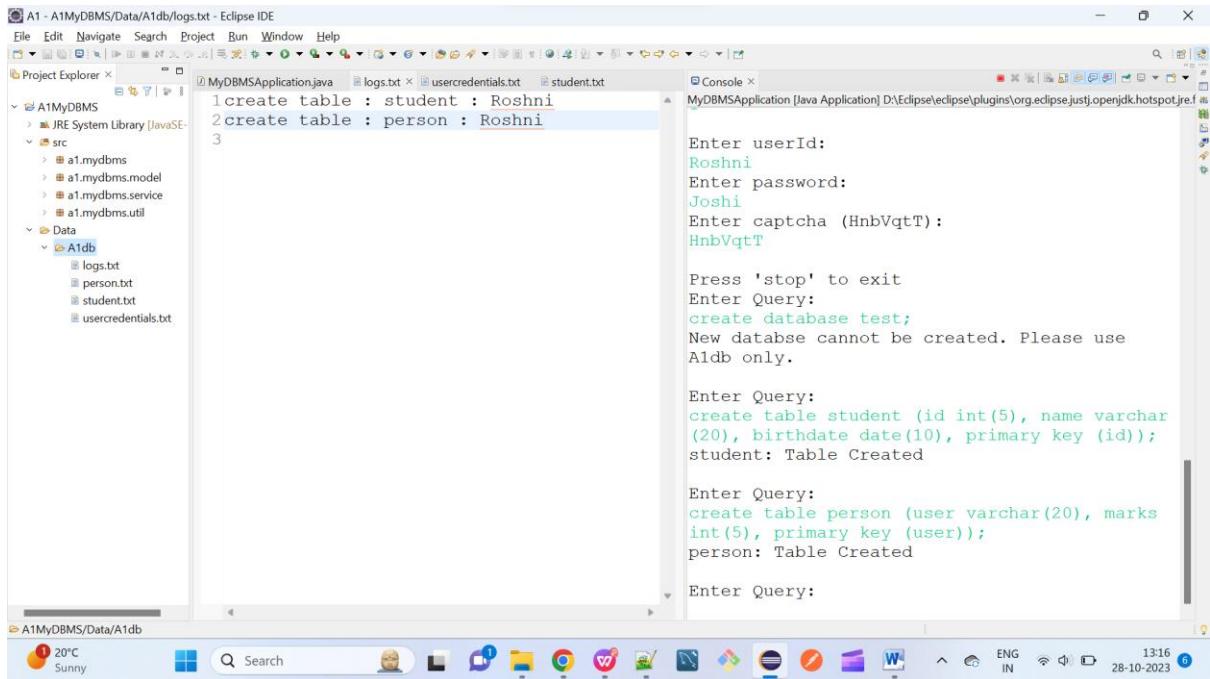
The screenshot shows the Eclipse IDE interface. In the Project Explorer, there is a Java project named 'A1MyDBMS' containing a package 'src' with classes 'a1.mydbms', 'a1.mydbms.model', 'a1.mydbms.service', and 'a1.mydbms.util'. Inside 'Data/A1db', there are files 'logs.txt', 'student.txt', and 'usercredentials.txt'. In the editor, 'MyDBMSApplication.java' contains the following code:

```
1int 5 | date 10 | varchar 20
2id | birthdate | name
3
```

In the Console window, the output is identical to Figure 10, showing the creation of the 'student' table and the update of the 'student.txt' file.

Figure 11: Student table text file contents

- In the similar way, I have successfully created the person table as shown in below screenshots.



The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS".
- MyDBMSApplication.java:** Contains the following SQL code:


```
1 create table : student : Roshni
2 create table : person : Roshni
3
```
- Console:** Displays the output of the database creation process:


```
Enter userId:
Roshni
Enter password:
Joshi
Enter captcha (HnbVqT):
HnbVqT

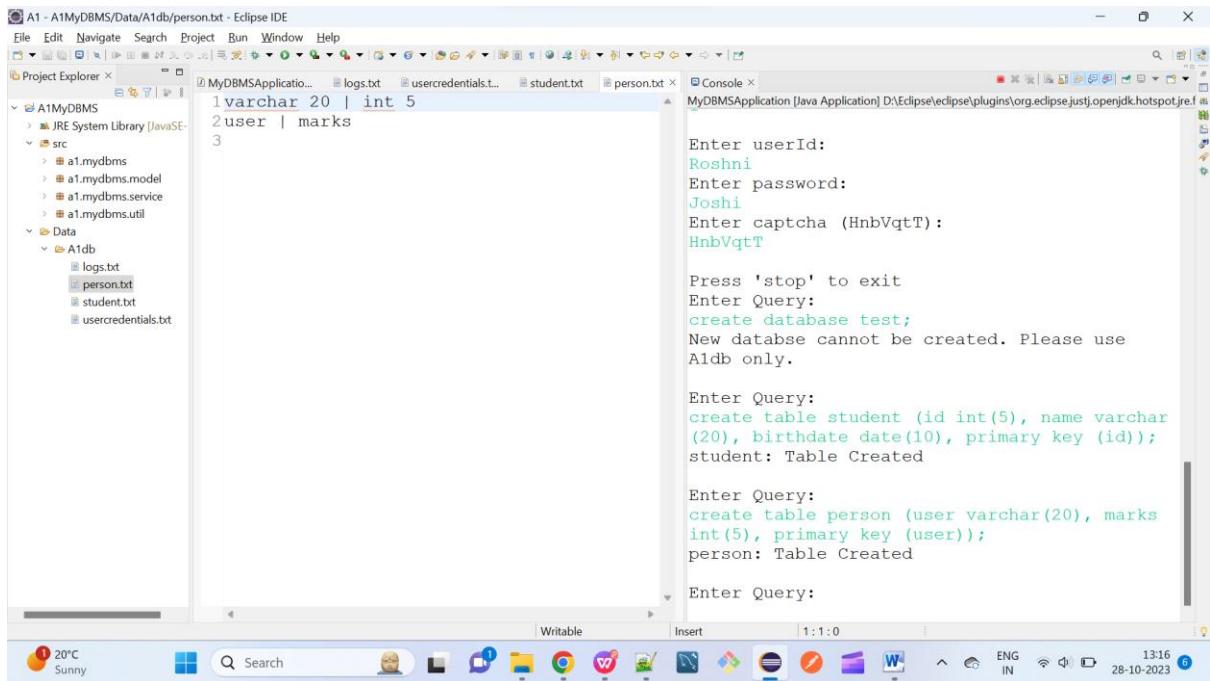
Press 'stop' to exit
Enter Query:
create database test;
New database cannot be created. Please use
A1db only.

Enter Query:
create table student (id int(5), name varchar
(20), birthdate date(10), primary key (id));
student: Table Created

Enter Query:
create table person (user varchar(20), marks
int(5), primary key (user));
person: Table Created

Enter Query:
```
- Windows Taskbar:** Shows the date (28-10-2023), time (13:16), and system status (ENG IN).

Figure 12: Successfully created person table, log file updated



The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS".
- MyDBMSApplication.java:** Contains the following SQL code:


```
1 varchar 20 | int 5
2 user | marks
3
```
- Console:** Displays the output of the database creation process:


```
Enter userId:
Roshni
Enter password:
Joshi
Enter captcha (HnbVqT):
HnbVqT

Press 'stop' to exit
Enter Query:
create database test;
New database cannot be created. Please use
A1db only.

Enter Query:
create table student (id int(5), name varchar
(20), birthdate date(10), primary key (id));
student: Table Created

Enter Query:
create table person (user varchar(20), marks
int(5), primary key (user));
person: Table Created

Enter Query:
```
- Windows Taskbar:** Shows the date (28-10-2023), time (13:16), and system status (ENG IN).

Figure 13: person table text file contents

- Now, I am trying to create student table again. In this case, system will not create new table and throws an error message stating that table already exists.

A1 - A1MyDBMS/Data/A1db/logs.txt - Eclipse IDE

File Edit Navigate Search Project Run Window Help

Project Explorer

MyDBMSApplication logs.txt usercredentials.txt student.txt person.txt

1 create table : student : Roshni
2 create table : person : Roshni
3

Console

MyDBMSApplication [Java Application] D:\Eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.l

Enter captcha (HnbVqtT):
HnbVqtT

Press 'stop' to exit
Enter Query:
`create database test;`
New database cannot be created. Please use A1db only.

Enter Query:
`create table student (id int(5), name varchar(20), birthdate date(10), primary key (id));`
student: Table Created

Enter Query:
`create table person (user varchar(20), marks int(5), primary key (user));`
person: Table Created

Enter Query:
`create table student (id int(5), name varchar(20), birthdate date(10), primary key (id));`
Table already exists

Enter Query:

13:17 28-10-2023

Figure 14: Trying to create student table again

- Now, in below scenarios, I am providing incorrect create table query. In all these cases, system will not create tables and will display the respective error message in console.

(1) Invalid tablename (e.g. student\$%)

A1 - A1MyDBMS/Data/A1db/logs.txt - Eclipse IDE

File Edit Navigate Search Project Run Window Help

Project Explorer

MyDBMSApplication logs.txt usercredentials.txt student.txt person.txt

1 create table : student : Roshni
2 create table : person : Roshni
3

Console

MyDBMSApplication [Java Application] D:\Eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.l

Enter Query:
`create database test;`
New database cannot be created. Please use A1db only.

Enter Query:
`create table student (id int(5), name varchar(20), birthdate date(10), primary key (id));`
student: Table Created

Enter Query:
`create table person (user varchar(20), marks int(5), primary key (user));`
person: Table Created

Enter Query:
`create table student (id int(5), name varchar(20), birthdate date(10), primary key (id));`
Table already exists

Enter Query:
`create table student$% (id int(5));`
Invalid table name

Enter Query:

13:18 28-10-2023

Figure 15: Create table query with invalid tablename

(2) Not providing size for varchar datatype column (i.e. name column)

The screenshot shows the Eclipse IDE interface with a Java project named 'A1MyDBMS'. In the 'Console' view, several 'Enter Query:' prompts are shown, each followed by a 'create table' command. The first two queries, 'create table : student : Roshni' and 'create table : person : Roshni', result in 'Table Created' messages. The third query, 'create table student (id int(5), name varchar (20), birthdate date(10), primary key (id));', results in an error message: 'Table already exists'. The fourth query, 'create table person (user varchar(20), marks int(5), primary key (user));', results in an error message: 'person: Table Created'. The fifth query, 'create table student (id int(5), name varchar (20), birthdate date(10), primary key (id));', results in an error message: 'Table already exists'. The sixth query, 'create table student\$% (id int(5));', results in an error message: 'Invalid table name'. The seventh query, 'create table demo (id int(5), name varchar, birthdate date(10), primary key (id));', results in an error message: 'Please provide size for column: name'. The eighth query, 'create table demo (id int(5), name varchar (20), birthdate date(10), primary key (id));', results in an error message: 'Invalid datatype key for columnn primary'. The system status bar at the bottom indicates it's 13:18 on 28-10-2023.

Figure 16: Create table query without size for name column

(3) Invalid primary key format

The screenshot shows the Eclipse IDE interface with a Java project named 'A1MyDBMS'. In the 'Console' view, several 'Enter Query:' prompts are shown, each followed by a 'create table' command. The first two queries, 'create table : student : Roshni' and 'create table : person : Roshni', result in 'Table Created' messages. The third query, 'create table person (user varchar(20), marks int(5), primary key (user));', results in an error message: 'person: Table Created'. The fourth query, 'create table student (id int(5), name varchar (20), birthdate date(10), primary key (id));', results in an error message: 'Table already exists'. The fifth query, 'create table student\$% (id int(5));', results in an error message: 'Invalid table name'. The sixth query, 'create table demo (id int(5), name varchar, birthdate date(10), primary key (id));', results in an error message: 'Please provide size for column: name'. The seventh query, 'create table demo (id int(5), name varchar (20), birthdate date(10), primary key (id));', results in an error message: 'Invalid datatype key for columnn primary'. The system status bar at the bottom indicates it's 13:20 on 28-10-2023.

Figure 17: Create table query with invalid primary key format

(4) Invalid column name (e.g. id&)

The screenshot shows the Eclipse IDE interface with a Java project named "A1MyDBMS". The "Console" tab displays several failed SQL create table statements due to invalid column names:

- create table student : Roshni
- create table person : Roshni
- create table student\$% (id int(5)); Invalid table name
- create table demo (id int(5), name varchar, birthdate date(10), primary key (id)); Please provide size for column: name
- create table demo (id int(5), name varchar(20), birthdate date(10), primary key (id)); Invalid datatype key for columnn primary
- create table test (id& int(5), name varchar(20), primary key (id&)); Invalid column name: id&

Figure 18: Create table query with invalid column name

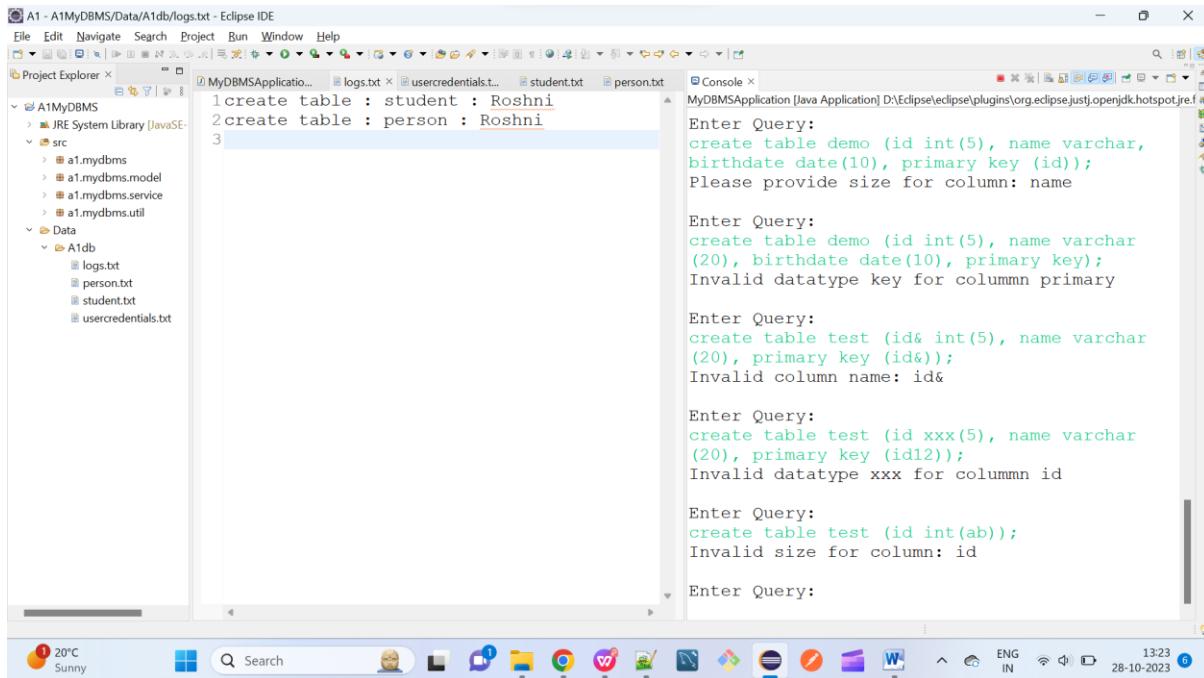
(5) Invalid datatype (i.e. xxx)

The screenshot shows the Eclipse IDE interface with a Java project named "A1MyDBMS". The "Console" tab displays failed SQL create table statements due to invalid datatypes:

- create table student : Roshni
- create table person : Roshni
- create table student\$% (id int(5)); Invalid table name
- create table demo (id int(5), name varchar, birthdate date(10), primary key (id)); Please provide size for column: name
- create table demo (id int(5), name varchar(20), birthdate date(10), primary key (id)); Invalid datatype key for columnn primary
- create table test (id& int(5), name varchar(20), primary key (id&)); Invalid column name: id&
- create table test (id xxx(5), name varchar(20), primary key (id12)); Invalid datatype xxx for columnn id

Figure 19: Create table query with invalid datatype

(6) Invalid size (i.e. non numeric size)



The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS".
- Console:** Displays several failed SQL create table queries due to invalid column sizes.

```

1 create table : student : Roshni
2 create table : person : Roshni
3

Enter Query:
create table demo (id int(5), name varchar,
birthdate date(10), primary key (id));
Please provide size for column: name

Enter Query:
create table demo (id int(5), name varchar
(20), birthdate date(10), primary key);
Invalid datatype key for columnn primary

Enter Query:
create table test (id& int(5), name varchar
(20), primary key (id&));
Invalid column name: id&

Enter Query:
create table test (id xxx(5), name varchar
(20), primary key (id12));
Invalid datatype xxx for colummn id

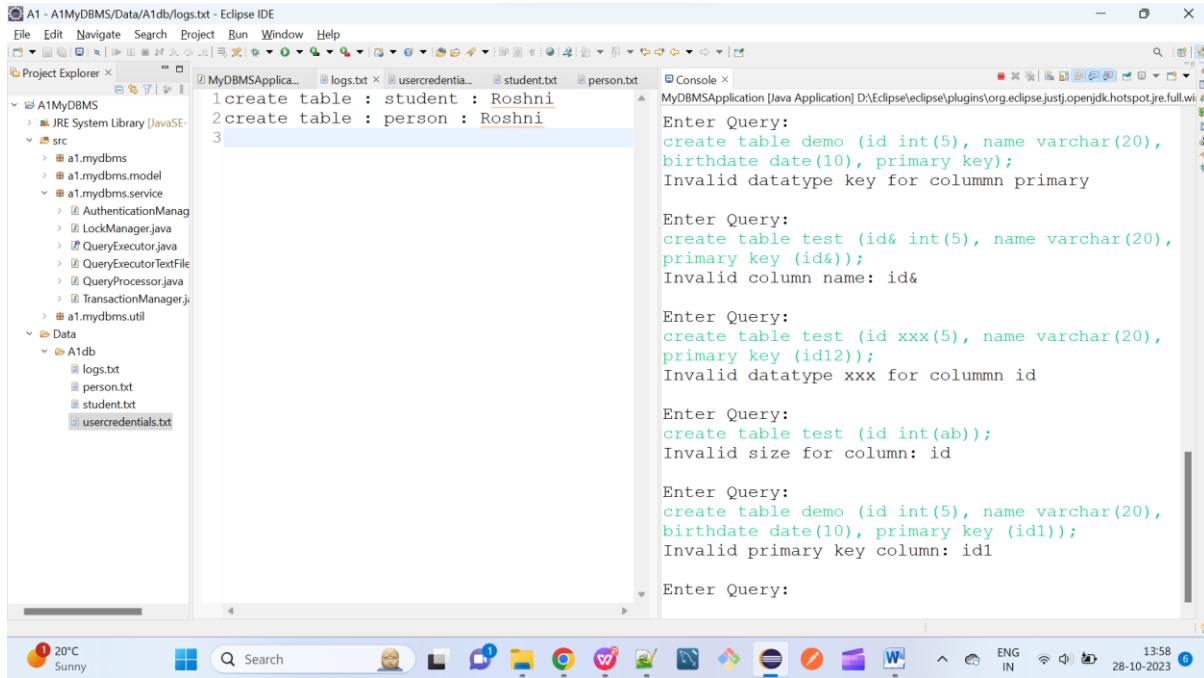
Enter Query:
create table test (id int(ab));
Invalid size for column: id

Enter Query:

```

Figure 20: Create table query with invalid size value

(7) Invalid primary key column (e.g. id1)



The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS".
- Console:** Displays several failed SQL create table queries due to invalid primary key column names.

```

1 create table : student : Roshni
2 create table : person : Roshni
3

Enter Query:
create table demo (id int(5), name varchar(20),
birthdate date(10), primary key);
Invalid datatype key for columnn primary

Enter Query:
create table test (id& int(5), name varchar(20),
primary key (id&));
Invalid column name: id&

Enter Query:
create table test (id xxx(5), name varchar(20),
primary key (id12));
Invalid datatype xxx for colummn id

Enter Query:
create table test (id int(ab));
Invalid size for column: id

Enter Query:
create table demo (id int(5), name varchar(20),
birthdate date(10), primary key (id1));
Invalid primary key column: id1

Enter Query:

```

Figure 21: Create table query with invalid primary key column

- As shown in below screenshots, no tables are created for above mentioned invalid scenarios. Only student and person table was created.

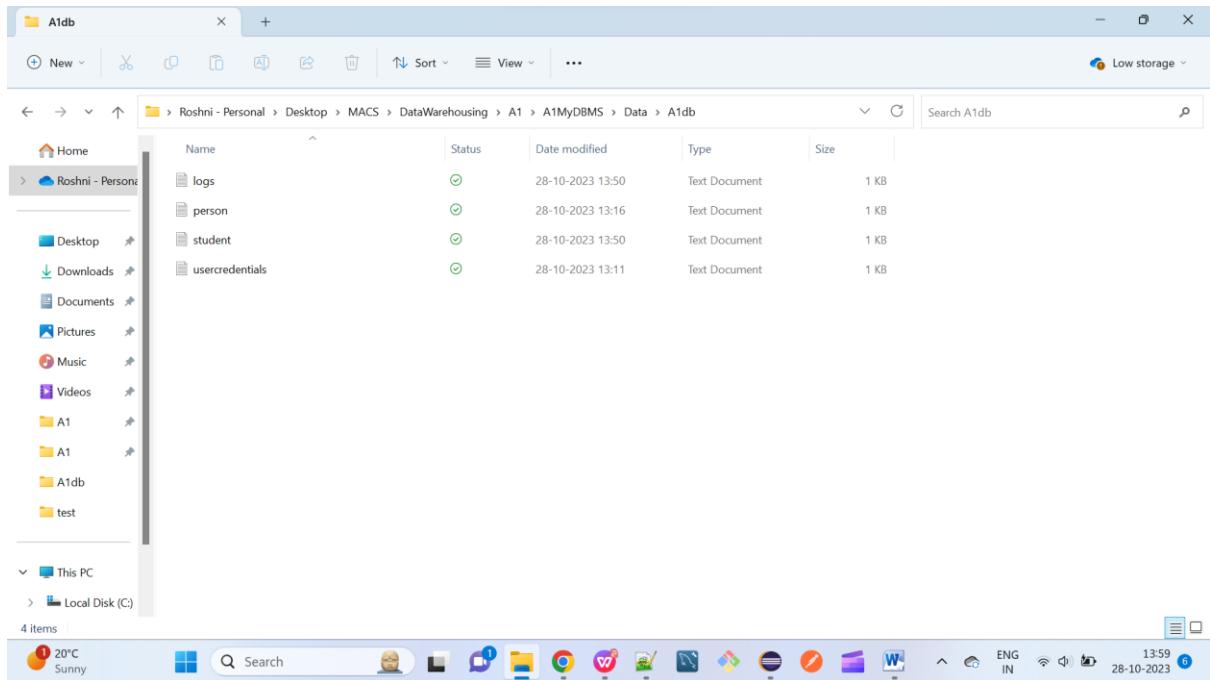


Figure 22: Database folder after all above queries execution

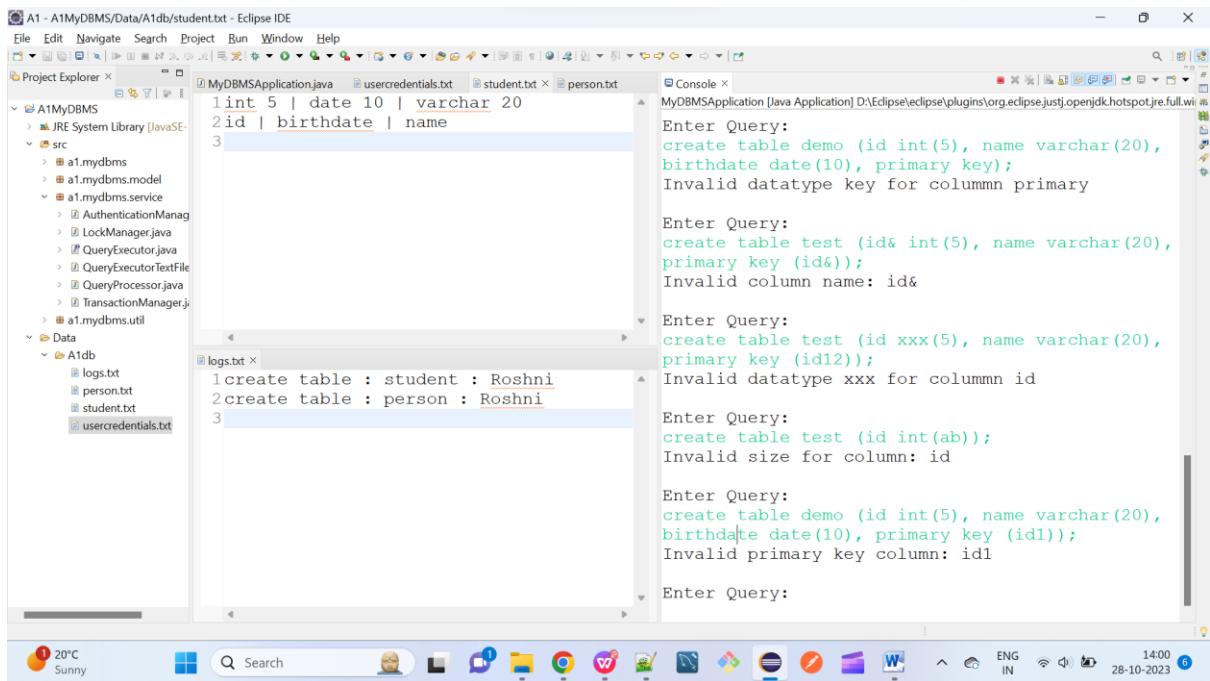


Figure 23: Project database tables after all above queries execution

Insert Query

- Now, I am inserting 5 data rows into student table. All five records are inserted into student table and logs file is getting updated for each query as shown in below screenshots. System will provide the status message for each query execution.

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with packages like A1MyDBMS, a1.mydbms, and a1.mydbms.service.
- MyDBMSApplication.java:** Contains code for creating a table named 'test' with columns 'id' (int), 'name' (varchar(20)), and a primary key on 'id'. The 'id' column is highlighted in red.
- Console:** Displays the error message: "Invalid column name: id&"
- logs.txt:** Shows the log entries for table creation and the first insert operation.
- student.txt:** Shows the log entry for the insert operation.
- person.txt:** Shows the log entry for the insert operation.
- usercredentials.txt:** Shows the log entry for the insert operation.
- Logs:** Shows the log file with the inserted data row.
- System Tray:** Shows the date and time as 28-10-2023 14:03.

Figure 24: Inserted first data row in student table, log file updated

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with packages like A1MyDBMS, a1.mydbms, and a1.mydbms.service.
- MyDBMSApplication.java:** Contains code for creating a table named 'test' with columns 'id' (int), 'name' (varchar(20)), and a primary key on 'id'. The 'id' column is highlighted in red.
- Console:** Displays the error message: "Invalid datatype xxx for columnmn id"
- logs.txt:** Shows the log entries for table creation and the second insert operation.
- student.txt:** Shows the log entry for the insert operation.
- person.txt:** Shows the log entry for the insert operation.
- usercredentials.txt:** Shows the log entry for the insert operation.
- Logs:** Shows the log file with the inserted data row.
- System Tray:** Shows the date and time as 28-10-2023 14:04.

Figure 25: Inserted second data row in student table, log file updated

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with packages like A1MyDBMS, src, and A1db.
- MyDBMSApplication.java:** Contains code for database operations.
- logs.txt:** Log file showing the creation of tables and the insertion of three rows into the student table.
- Console:** Displays SQL queries and their execution results, including an invalid primary key error for the id column.
- Taskbar:** Shows system status including weather (20°C, Sunny), date (28-10-2023), and time (14:04).

```

1 create table : student : Roshni
2 create table : person : Roshni
3 insert into : student : Roshni
4 insert into : student : Roshni
5 insert into : student : Roshni
6

```

Figure 26: Inserted third data row in student table, log file updated

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with packages like A1MyDBMS, src, and A1db.
- MyDBMSApplication.java:** Contains code for database operations.
- logs.txt:** Log file showing the creation of tables and the insertion of four rows into the student table.
- Console:** Displays SQL queries and their execution results, including an invalid primary key error for the id column.
- Taskbar:** Shows system status including weather (20°C, Sunny), date (28-10-2023), and time (14:05).

```

1 create table : student : Roshni
2 create table : person : Roshni
3 insert into : student : Roshni
4 insert into : student : Roshni
5 insert into : student : Roshni
6 insert into : student : Roshni
7

```

Figure 27: Inserted fourth data row in student table, log file updated

```

A1 - A1MyDBMS/Data/A1db/student.txt - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Project Explorer
MyDBMSApplication.java usercredentials.txt student.txt person.txt
A1MyDBMS JRE System Library JavaSE-...
src a1.mydbms a1.mydbms.model a1.mydbms.service a1.mydbms.util
A1db logs.txt person.txt student.txt usercredentials.txt
MyDBMSApplication [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64\bin\java -jar C:\Users\DELL\OneDrive\Desktop\A1MyDBMS\bin\A1MyDBMS.jar
insert into student (id, name, birthdate) values
(10, abc, 06/07/2023);
student: Insert executed

Enter Query:
insert into student (id, name, birthdate) values
(20, Roshni, 06/08/2010);
student: Insert executed

Enter Query:
insert into student (id, name, birthdate) values
(30, Bhavee, 04/09/2019);
student: Insert executed

Enter Query:
insert into student (id, name, birthdate) values
(40, Roshni, 20/10/1999);
student: Insert executed

Enter Query:
insert into student (id, name, birthdate) values
(50, Dhruvi, 17/05/2023);
student: Insert executed

Enter Query:

```

Figure 28: Inserted fifth data row in student table, log file updated

- Similarly, I am inserting 2 data rows into person table. Both records are inserted into person table and logs file is getting updated for each query as shown in below screenshots. System will provide the status message for each query execution.

```

A1 - A1MyDBMS/Data/A1db/logs.txt - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Project Explorer
MyDBMSApplication.java usercredentials.txt student.txt person.txt
A1MyDBMS JRE System Library JavaSE-...
src a1.mydbms a1.mydbms.model a1.mydbms.service a1.mydbms.util
A1db logs.txt person.txt student.txt usercredentials.txt
MyDBMSApplication [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64\bin\java -jar C:\Users\DELL\OneDrive\Desktop\A1MyDBMS\bin\A1MyDBMS.jar
1varchar 20 | int 5
2user | marks
3testuser | 20
4

Enter Query:
insert into student (id, name, birthdate) values
(20, Roshni, 06/08/2010);
student: Insert executed

Enter Query:
insert into student (id, name, birthdate) values
(30, Bhavee, 04/09/2019);
student: Insert executed

Enter Query:
insert into student (id, name, birthdate) values
(40, Roshni, 20/10/1999);
student: Insert executed

Enter Query:
insert into person (user, marks) values
(testuser, 20);
person: Insert executed

Enter Query:

```

Figure 29: Inserted first data row in person table, log file updated

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with packages like A1MyDBMS, a1.mydbms, and a1.mydbms.service.
- Code Editor:** Displays the `MyDBMSApplication.java` file containing database schema definitions and log entries from `logs.txt`. One entry is highlighted: "9 insert into : person : Roshni".
- Console:** Shows the output of the application's database operations. It includes several `insert into` statements and their corresponding results, such as "student: Insert executed" and "person: Insert executed".
- System Tray:** Shows the Windows taskbar with various icons and system status information.

Figure 30: Inserted second data row in person table, log file updated

- Now, I am trying to insert new row with id=10 in student table. But the system will not allow me to insert, as data with id=10 is already present in student table. In student table, id is the primary key and it must store unique values. Hence, system will throw me error message as ‘Data already exists’ as shown in below screenshot.

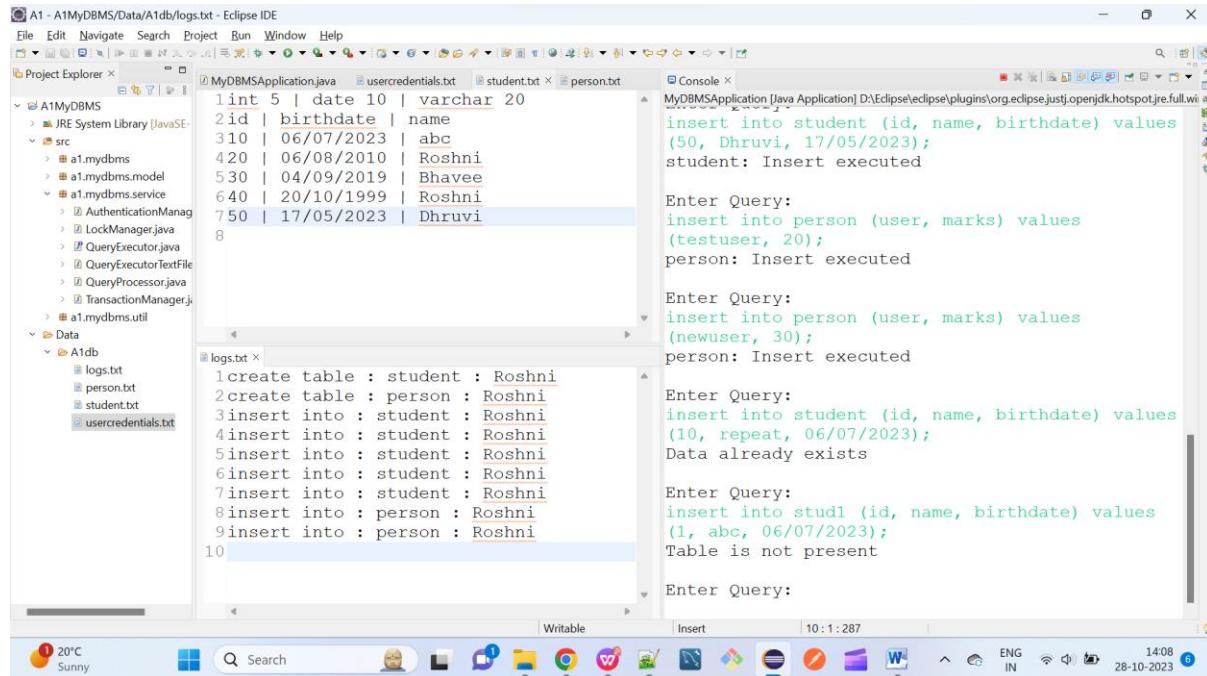
The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with packages like A1MyDBMS, a1.mydbms, and a1.mydbms.service.
- Code Editor:** Displays the `MyDBMSApplication.java` file containing database schema definitions and log entries from `logs.txt`. One entry is highlighted: "10 insert into : student : Roshni".
- Console:** Shows the output of the application's database operations. It includes several `insert into` statements and their corresponding results, such as "student: Insert executed" and "person: Insert executed".
- Output:** Shows an error message: "Data already exists" when attempting to insert a row with id=10 into the student table.
- System Tray:** Shows the Windows taskbar with various icons and system status information.

Figure 31: Trying to insert data with existing primary key value

- Now, in below scenarios, I am providing incorrect insert query. In all these cases, system will not insert data and will display the respective error message in console.

(1) Providing table name that doesn't exist

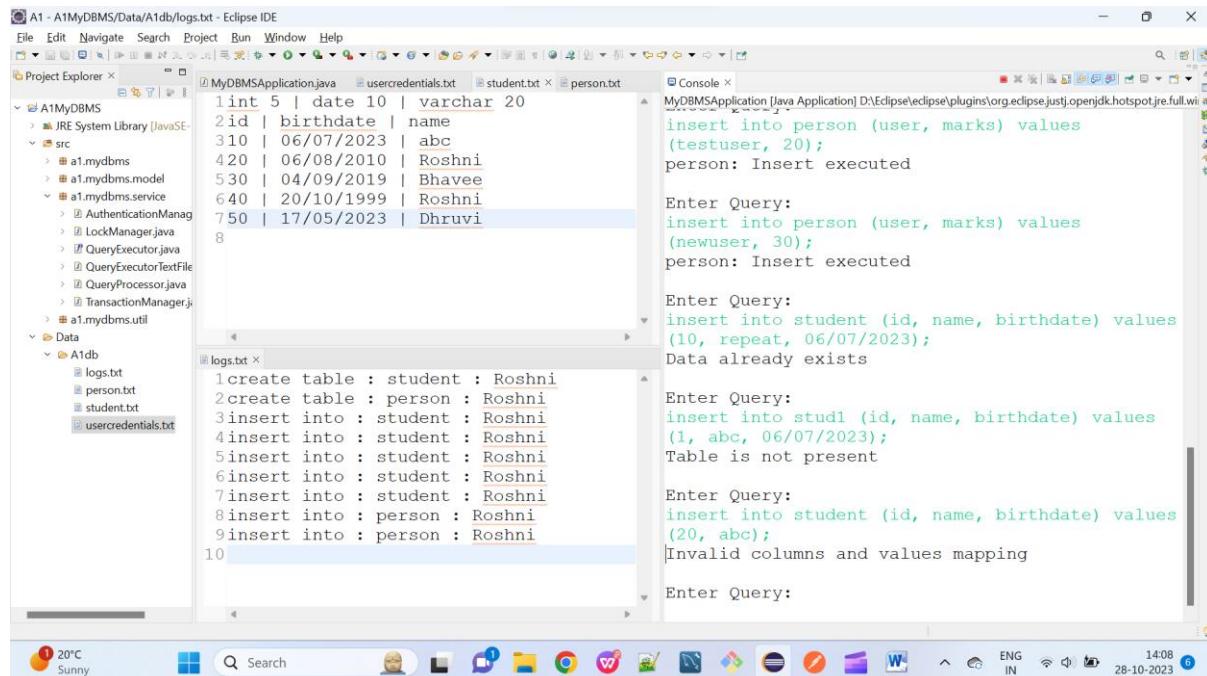


The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with files like MyDBMSApplication.java, usercredentials.txt, student.txt, and person.txt.
- Console:** Displays the following SQL queries and their results:
 - insert into student (id, name, birthdate) values (50, Dhruvi, 17/05/2023); student: Insert executed
 - Enter Query: insert into person (user, marks) values (testuser, 20); person: Insert executed
 - Enter Query: insert into person (user, marks) values (newuser, 30); person: Insert executed
 - Enter Query: insert into student (id, name, birthdate) values (10, repeat, 06/07/2023); Data already exists
 - Enter Query: insert into stud1 (id, name, birthdate) values (1, abc, 06/07/2023); Table is not present
 - Enter Query:
- logs.txt:** Shows log entries for table creation and insertions.
- System Tray:** Shows weather (20°C, Sunny), date (28-10-2023), and time (14:08).

Figure 32: Insert query with invalid tablename

(2) Different columns and values count



The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with files like MyDBMSApplication.java, usercredentials.txt, student.txt, and person.txt.
- Console:** Displays the following SQL queries and their results:
 - insert into person (user, marks) values (testuser, 20); person: Insert executed
 - Enter Query: insert into person (user, marks) values (newuser, 30); person: Insert executed
 - Enter Query: insert into student (id, name, birthdate) values (10, repeat, 06/07/2023); Data already exists
 - Enter Query: insert into stud1 (id, name, birthdate) values (1, abc, 06/07/2023); Table is not present
 - Enter Query: insert into student (id, name, birthdate) values (20, abc); Invalid columns and values mapping
 - Enter Query:
- logs.txt:** Shows log entries for table creation and insertions.
- System Tray:** Shows weather (20°C, Sunny), date (28-10-2023), and time (14:08).

Figure 33: Insert query with different columns and values count

(3) Value that doesn't match datatype of a column (e.g. giving non numeric value for id)

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS".
- Console:** Displays the following SQL queries and their results:
 - insert into person (user, marks) values (newuser, 30);
person: Insert executed
 - Enter Query:
insert into student (id, name, birthdate) values (10, repeat, 06/07/2023);
Data already exists
 - Enter Query:
insert into stud1 (id, name, birthdate) values (1, abc, 06/07/2023);
Table is not present
 - Enter Query:
insert into student (id, name, birthdate) values (20, abc);
Invalid columns and values mapping
 - Enter Query:
insert into student (id, name, birthdate) values (xyz, abc, 06/07/2023);
Invalid data type for xyz
 - Enter Query:
- logs.txt:** Shows log entries related to table creation and insertions.
- System Tray:** Shows weather (20°C, Sunny), date (28-10-2023), and time (14:08).

Figure 34: Insert query with invalid datatype

(4) Value having larger size than specified for a column

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS".
- Console:** Displays the following SQL queries and their results:
 - insert into student (id, name, birthdate) values (10, repeat, 06/07/2023);
Data already exists
 - Enter Query:
insert into stud1 (id, name, birthdate) values (1, abc, 06/07/2023);
Table is not present
 - Enter Query:
insert into student (id, name, birthdate) values (20, abc);
Invalid columns and values mapping
 - Enter Query:
insert into student (id, name, birthdate) values (xyz, abc, 06/07/2023);
Invalid data type for xyz
 - Enter Query:
insert into student (id, name, birthdate) values (46666666, Roshni, 05/03/2010);
Invalid length for 46666666
 - Enter Query:
- logs.txt:** Shows log entries related to table creation and insertions.
- System Tray:** Shows weather (20°C, Sunny), date (28-10-2023), and time (14:09).

Figure 35: Insert query with invalid size

Select query

- In below screenshots, I am executing the select queries for student table.

(1) Select all data query

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with packages like A1MyDBMS, JRE System Library, and src containing various Java classes.
- MyDBMSApplication.java:** Contains a main method with a loop that reads from a file named student.txt.

```
int i; date d; varchar name;
for(i=0; i<10; i++) {
    d = new date();
    name = "Roshni";
    System.out.println("insert into student (id, name, birthdate) values (" + i + ", " + name + ", " + d + ");");
}
```
- student.txt:** A file containing 10 rows of data.

ID	Date	Name
1	06/07/2023	Roshni
2	06/08/2010	Roshni
3	04/09/2019	Bhavee
4	20/10/1999	Roshni
5	17/05/2023	Dhrushi
6		
7		
8		
9		
10		
- Console:** Displays the output of the Java application as SQL statements being executed.

```
insert into student (id, name, birthdate) values (20, abc);
Invalid columns and values mapping
```

```
Enter Query:
insert into student (id, name, birthdate) values (xyz, abc, 06/07/2023);
Invalid data type for xyz
```

```
Enter Query:
insert into student (id, name, birthdate) values (46666666, Roshni, 05/03/2010);
Invalid length for 46666666
```

```
Enter Query:
select * from student;
id | birthdate | name
-----
```

- logs.txt:** A log file showing the execution of SQL statements.

```
1.create table : student : Roshni
2.create table : person : Roshni
3.insert into : student : Roshni
4.insert into : student : Roshni
5.insert into : student : Roshni
6.insert into : student : Roshni
7.insert into : student : Roshni
8.insert into : person : Roshni
9.insert into : person : Roshni
10
```
- usercredentials.txt:** A file containing a single row of credentials.

```
abc
```

Figure 36: Select all data query

(2) Select only name and birthdate columns

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** A1 - A1MyDBMS/Data/A1db/student.txt - Eclipse IDE
- Menu Bar:** File Edit Navigate Search Project Run Window Help
- Project Explorer:** Shows the project structure with packages like A1MyDBMS, JRE System Library, and src containing various Java classes.
- MyDBMSApplication.java:** Contains a main method with a try-catch block. The catch block prints an error message: "Invalid length for 46666666".
- student.txt:** A file containing student data with columns id, birthdate, and name.
- person.txt:** A file containing person data with columns id, birthdate, and name.
- logs.txt:** A log file showing multiple insert operations into the student and person tables, all performed by user Roshni.
- Console:** Displays the output of the application, including the error message from the catch block and the log entries from logs.txt.

Figure 37: select specific columns data

(3) Select all data with where clause

```

MyDBMSApplication [Java Application] D:\Eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64\bin\java -jar MyDBMSApplication.jar
id | birthdate | name
-----
10 | 06/07/2023 | abc
20 | 06/08/2010 | Roshni
30 | 04/09/2019 | Bhavee
40 | 20/10/1999 | Roshni
50 | 17/05/2023 | Dhruvi

Enter Query:
select name, birthdate from student;
name | birthdate
-----
abc | 06/07/2023
Roshni | 06/08/2010
Bhavee | 04/09/2019
Roshni | 20/10/1999
Dhruvi | 17/05/2023

Enter Query:
select * from student where id = 10;
id | birthdate | name
-----
10 | 06/07/2023 | abc

Enter Query:

```

Figure 38: select all columns with where clause

(4) Select only name, birthdate with where clause

```

MyDBMSApplication [Java Application] D:\Eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64\bin\java -jar MyDBMSApplication.jar
Enter Query:
select name, birthdate from student;
name | birthdate
-----
abc | 06/07/2023
Roshni | 06/08/2010
Bhavee | 04/09/2019
Roshni | 20/10/1999
Dhruvi | 17/05/2023

Enter Query:
select * from student where id = 10;
id | birthdate | name
-----
10 | 06/07/2023 | abc

Enter Query:
select name, birthdate from student where name = Roshni;
name | birthdate
-----
Roshni | 06/08/2010
Roshni | 20/10/1999

Enter Query:

```

Figure 39: select specific columns with where clause

(5) Select non existing data

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS".
- Console:** Displays the output of a database query. The query "select * from student where id = 10;" returns a single row: "10 | 06/07/2023 | abc".
- Logs.txt:** A file containing SQL statements used to create tables and insert data. It includes:


```

1create table : student : Roshni
2create table : person : Roshni
3insert into : student : Roshni
4insert into : student : Roshni
5insert into : student : Roshni
6insert into : student : Roshni
7insert into : student : Roshni
8insert into : person : Roshni
9insert into : person : Roshni
10
      
```

Figure 40: select non existing data

- In below screenshot, I am trying to select from non existing table, which will throw error message.

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS".
- Console:** Displays the output of a database query. The query "select * from student where id = 10;" returns a single row: "10 | 06/07/2023 | abc".
- Logs.txt:** A file containing SQL statements used to create tables and insert data. It includes:

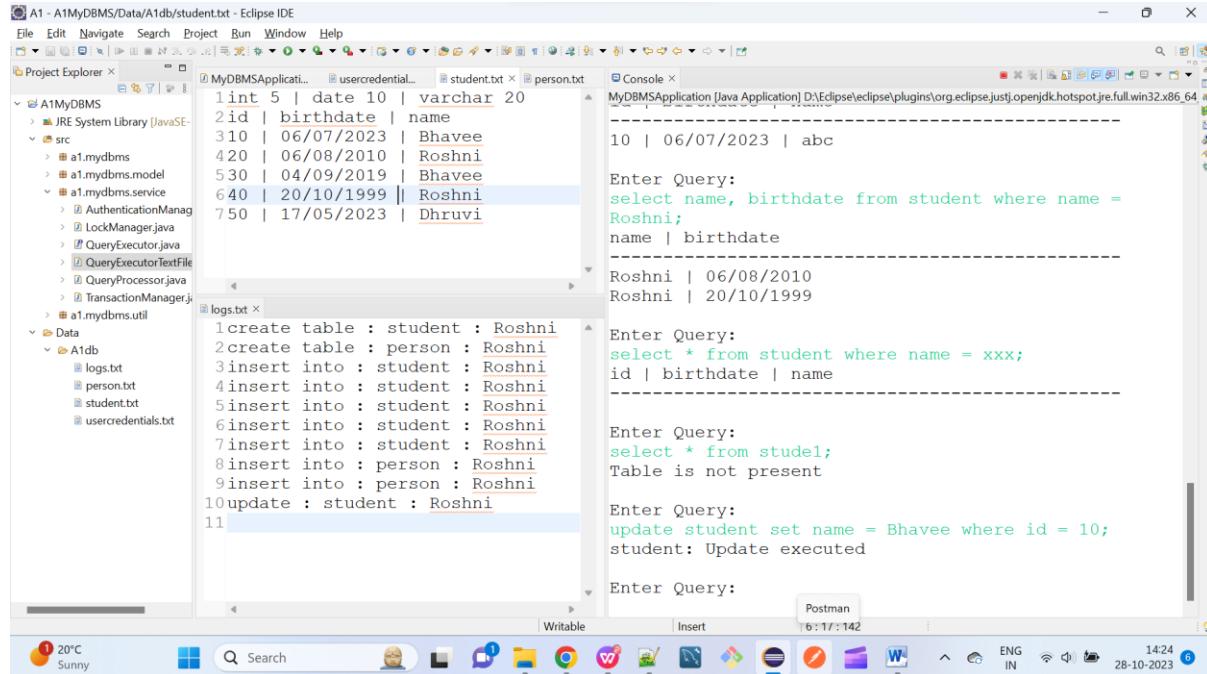

```

1create table : student : Roshni
2create table : person : Roshni
3insert into : student : Roshni
4insert into : student : Roshni
5insert into : student : Roshni
6insert into : student : Roshni
7insert into : student : Roshni
8insert into : person : Roshni
9insert into : person : Roshni
10
      
```
- Console (continued):** Shows an error message: "Table is not present" when attempting to select from a non-existing table named "stude1".

Figure 41: select from non existing table

Update Query

- In below two screenshots, I am updating the student table data. Queries are executed successfully as shown by success message in console and data got updated in student table for both the queries. Moreover, log file is also updated.



The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS" with various Java files like `a1.mydbms`, `a1.mydbms.model`, `a1.mydbms.service`, etc.
- Console:** Displays the output of the query execution. It shows the original data in `student.txt` and the result of the update query `update student set name = Bhavee where id = 10;`. The log file `logs.txt` also reflects this update.
- Logs:** The log file `logs.txt` contains the history of database operations, including the update command.

```

10 | 06/07/2023 | abc
Enter Query:
select name, birthdate from student where name =
Roshni;
name | birthdate
Roshni | 06/08/2010
Roshni | 20/10/1999

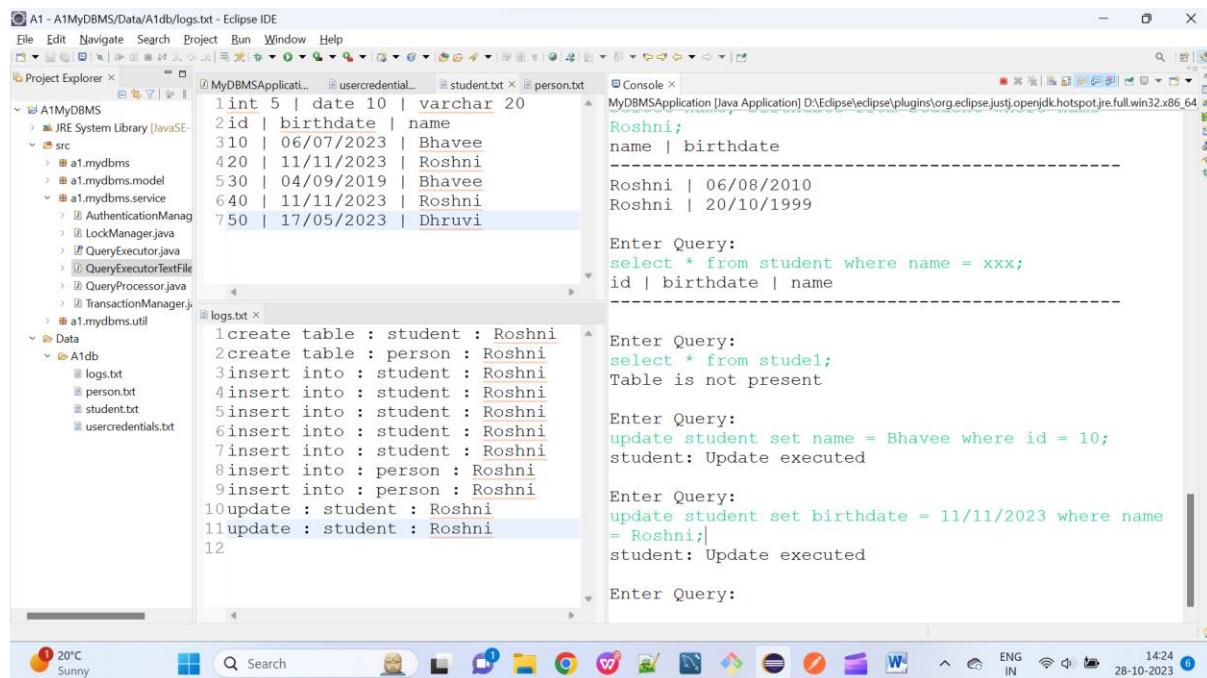
Enter Query:
select * from student where name = xxx;
id | birthdate | name
-----
Enter Query:
select * from studel;
Table is not present

Enter Query:
update student set name = Bhavee where id = 10;
student: Update executed

Enter Query:

```

Figure 42: name updated to Bhavee for id 10, log file updated



The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS" with various Java files like `a1.mydbms`, `a1.mydbms.model`, `a1.mydbms.service`, etc.
- Console:** Displays the output of the query execution. It shows the original data in `student.txt` and the result of the update query `update student set birthdate = 11/11/2023 where name = Roshni;`. The log file `logs.txt` also reflects this update.
- Logs:** The log file `logs.txt` contains the history of database operations, including the update command.

```

Roshni;
name | birthdate
Roshni | 06/08/2010
Roshni | 20/10/1999

Enter Query:
select * from student where name = xxx;
id | birthdate | name
-----
Enter Query:
select * from studel;
Table is not present

Enter Query:
update student set name = Bhavee where id = 10;
student: Update executed

Enter Query:
update student set birthdate = 11/11/2023 where name
= Roshni;
student: Update executed

Enter Query:

```

Figure 43: birthdate updated to 11/11/2023 for name Roshni, log file updated

- In below screenshot, I am trying to update non existing table, which throws error message.

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS".
- Console:** Displays the following SQL queries and their results:
 - Enter Query: `select * from student where name = xxx;`
 - Table is not present
 - Enter Query: `update student set name = Bhavee where id = 10;`
 - student: Update executed
 - Enter Query: `update student set birthdate = 11/11/2023 where name = Roshni;`
 - student: Update executed
 - Enter Query: `update studl set name = Bhavee where id = 2;`
 - Table is not present
 - Enter Query:
- Logs.txt:** Shows log entries corresponding to the queries executed.

Figure 44: Updating non existing table

Delete Query

- In below screenshots, I am performing the delete operation on student and person table data. Queries are executed successfully as shown by success message in console and data got deleted in student and person table. Moreover, log file is also updated.

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS".
- Console:** Displays the following SQL queries and their results:
 - Enter Query: `select * from student;`
 - Table is not present
 - Enter Query: `update student set name = Bhavee where id = 10;`
 - student: Update executed
 - Enter Query: `update student set birthdate = 11/11/2023 where name = Roshni;`
 - student: Update executed
 - Enter Query: `update studl set name = Bhavee where id = 2;`
 - Table is not present
 - Enter Query: `delete from student where id = 30;`
 - student: Delete executed
 - Enter Query:
- Logs.txt:** Shows log entries corresponding to the queries executed.

Figure 45: data with id 30 is deleted from student, log file updated

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS".
- Console:** Displays the following SQL queries and their results:
 - `select * from stud1;` - Table is not present
 - `update student set name = Bhavee where id = 10;` - student: Update executed
 - `update student set birthdate = 11/11/2023 where name = Roshni;` - student: Update executed
 - `update stud1 set name = Bhavee where id = 2;` - Table is not present
 - `delete from student where id = 30;` - student: Delete executed
 - `delete from student where name = Dhruvi;` - student: Delete executed
- logs.txt:** Shows log entries corresponding to the executed queries.

Figure 46: data with name Dhruvi is deleted from student, log file updated

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS".
- Console:** Displays the following SQL queries and their results:
 - `update student set birthdate = 11/11/2023 where name = Roshni;` - student: Update executed
 - `update stud1 set name = Bhavee where id = 2;` - Table is not present
 - `delete from student where id = 30;` - student: Delete executed
 - `delete from student where name = Dhruvi;` - student: Delete executed
 - `delete from stud1 where id = 3;` - Table is not present
 - `delete from person;` - person: Delete executed
- logs.txt:** Shows log entries corresponding to the executed queries.

Figure 47: Deleting all the data from person table

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS".
- Console:** Displays the following SQL queries and their results:
 - Enter Query: `update studl set name = Bhavee where id = 2;` Table is not present
 - Enter Query: `delete from student where id = 30;` student: Delete executed
 - Enter Query: `delete from student where name = Dhruvi;` student: Delete executed
 - Enter Query: `delete from studl where id = 3;` Table is not present
 - Enter Query: `delete from person;` person: Delete executed
 - Enter Query: `delete from student;` student: Delete executed
- Logs.txt Content:**

```

1 create table : student : Roshni
2 create table : person : Roshni
3 insert into : student : Roshni
4 insert into : student : Roshni
5 insert into : student : Roshni
6 insert into : student : Roshni
7 insert into : student : Roshni
8 insert into : person : Roshni
9 insert into : person : Roshni
10 update : student : Roshni
11 update : student : Roshni
12 delete from : student : Roshni
13 delete from : student : Roshni
14 delete from : person : Roshni
15 delete from : student : Roshni
16

```

Figure 48: Deleting all data from student table

- In below screenshot, I am trying to delete from non existing table, which throws error message

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS".
- Console:** Displays the following SQL queries and their results:
 - Enter Query: `update student set name = Bhavee where id = 10;` student: Update executed
 - Enter Query: `update student set birthdate = 11/11/2023 where name = Roshni;` student: Update executed
 - Enter Query: `update studl set name = Bhavee where id = 2;` Table is not present
 - Enter Query: `delete from student where id = 30;` student: Delete executed
 - Enter Query: `delete from student where name = Dhruvi;` student: Delete executed
 - Enter Query: `delete from studl where id = 3;` Table is not present
- Logs.txt Content:**

```

1 create table : student : Roshni
2 create table : person : Roshni
3 insert into : student : Roshni
4 insert into : student : Roshni
5 insert into : student : Roshni
6 insert into : student : Roshni
7 insert into : student : Roshni
8 insert into : person : Roshni
9 insert into : person : Roshni
10 update : student : Roshni
11 update : student : Roshni
12 delete from : student : Roshni
13 delete from : student : Roshni
14

```

Figure 49: deleting from non existing table

- Finally, I am exiting from the 'enter query' option using 'stop' as shown in below screenshot.

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS".
- logs.txt:** Contains log entries from the application's execution.
- Console:** Displays the following command-line interactions:

 - student: Delete executed
 - Enter Query:
delete from student where name = Dhrushi;
student: Delete executed
 - Enter Query:
delete from stud1 where id = 3;
Table is not present
 - Enter Query:
delete from person;
person: Delete executed
 - Enter Query:
delete from student;
student: Delete executed
 - Enter Query:
stop
 - Please select from below option :
1. Login
2. Register
3. Logout

Figure 50: Exiting from enter query option

Creating new tables for transaction

- Here, I have restarted the application and deleted previously created student and person tables. Now, my database has only logs table and user credentials table with my previously stored credentials as shown in below screenshots. After restarting, I am successfully login using correct credentials.

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS".
- usercredentials.txt:** Contains the following data:

```
1 Roshni cec49593ca2c784c3ad8b3ea443ff642
2
```

- logs.txt:** Contains the following log entry:

```
1
```

- Console:** Displays the message: "No consoles to display at this time."

Figure 51: usercredentials table with previously provided login details

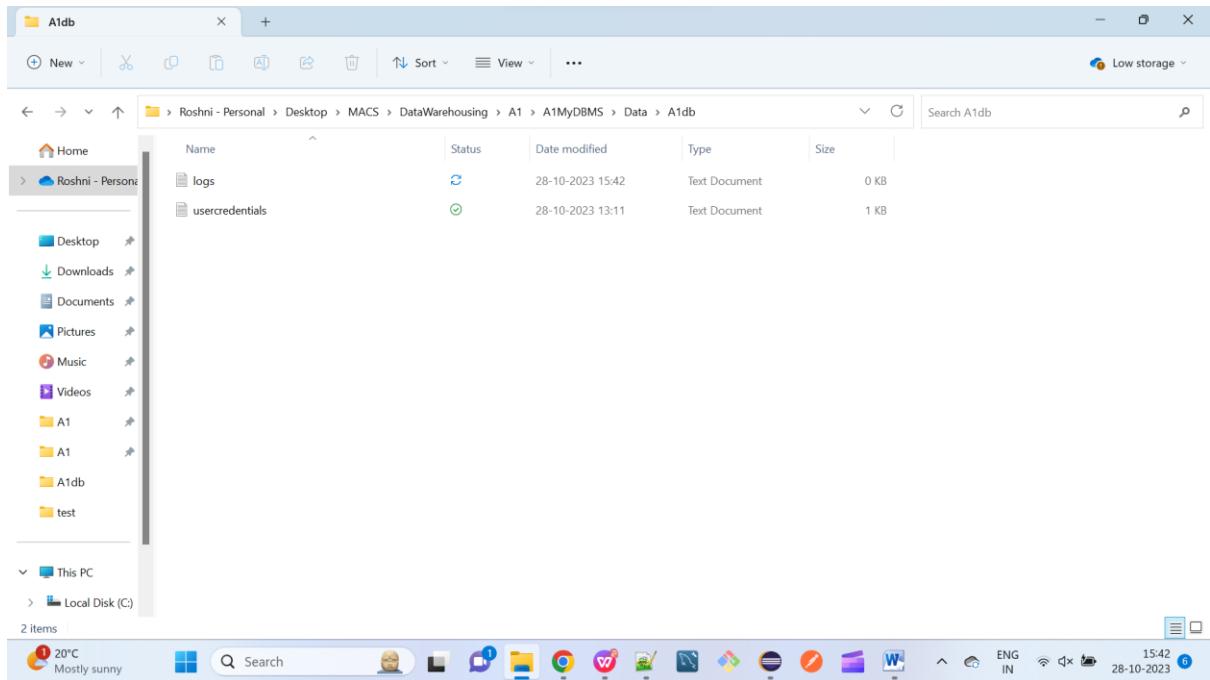


Figure 52: database containing only logs and usercredentials table

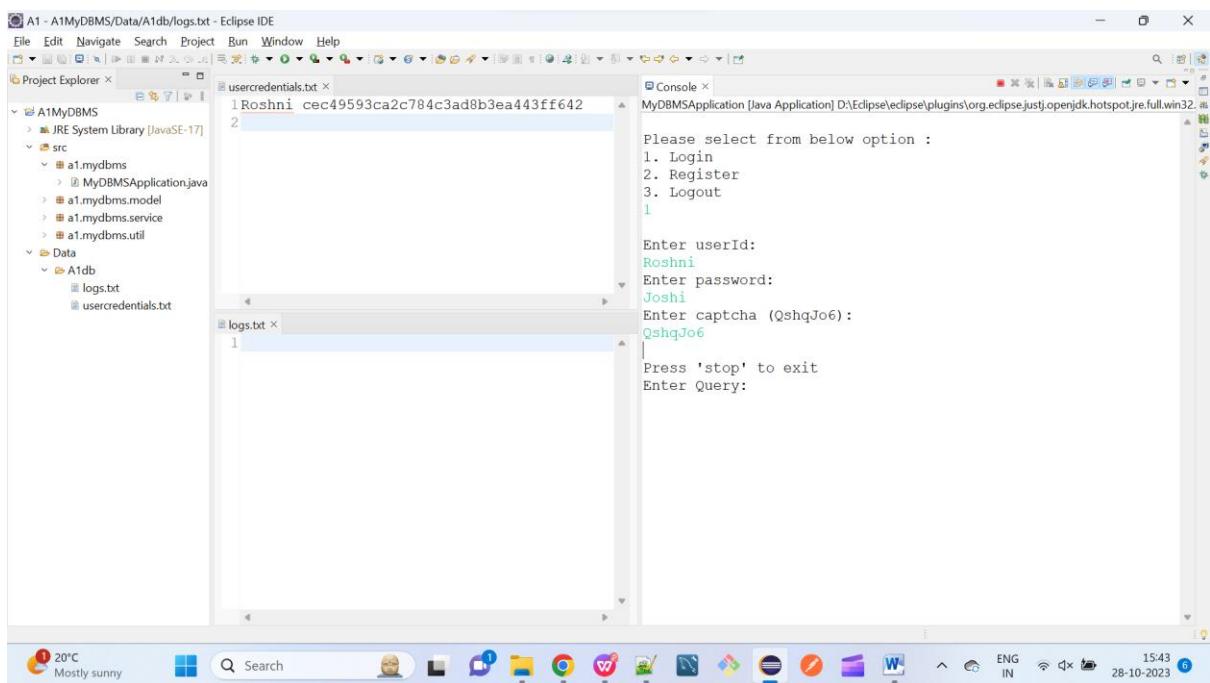


Figure 53: Login using correct credentials

- Now, I have created new student and person tables and inserted data in both of them as shown in below screenshots.

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with files like `A1MyDBMS`, `src`, `Data`, and `A1db`.
- Logs.txt:** Contains log entries for table creation and insertions:


```
1create table : student : Roshni
2insert into : student : Roshni
3insert into : student : Roshni
4insert into : student : Roshni
5insert into : student : Roshni
6insert into : student : Roshni
7
```
- Console:** Shows the execution of SQL queries and their results:


```
student: Table Created
Enter Query:
insert into student (id, name, birthdate) values (10, abc, 06/07/2023);
student: Insert executed
Enter Query:
insert into student (id, name, birthdate) values (20, Roshni, 06/08/2010);
student: Insert executed
Enter Query:
insert into student (id, name, birthdate) values (30, Bhavee, 04/09/2019);
student: Insert executed
Enter Query:
insert into student (id, name, birthdate) values (40, Roshni, 20/10/1999);
student: Insert executed
Enter Query:
insert into student (id, name, birthdate) values (50, Dhrushi, 17/05/2023);
student: Insert executed
Enter Query:
```
- System Tray:** Shows weather (20°C, Mostly sunny), date (28-10-2023), and time (15:44).

Figure 54: created student table and inserted data

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with files like `A1MyDBMS`, `src`, `Data`, and `A1db`.
- Logs.txt:** Contains log entries for table creation and insertions:


```
1varchar 20 | int 5
2user | marks
3testuser | 20
4newuser | 30
5
```
- Console:** Shows the execution of SQL queries and their results:


```
student: Insert executed
Enter Query:
insert into student (id, name, birthdate) values (40, Roshni, 20/10/1999);
student: Insert executed
Enter Query:
insert into student (id, name, birthdate) values (50, Dhrushi, 17/05/2023);
student: Insert executed
Enter Query:
create table person (user varchar(20), marks int(5), primary key (user));
person: Table Created
Enter Query:
insert into person (user, marks) values (testuser, 20);
person: Insert executed
Enter Query:
insert into person (user, marks) values (newuser, 30);
person: Insert executed
Enter Query:
Invalid query
Enter Query:
```
- System Tray:** Shows weather (20°C, Mostly sunny), date (28-10-2023), and time (15:45).

Figure 55: created person table and inserted data

Executing Transactions

- As shown in below screenshots, I am executing the transactions. Here, when I give ‘begin transaction’ statement, system will allow me to insert more queries without executing them right away, until I provide ‘end transaction’ statement.

- Furthermore, I implemented two-phase locking with table-level lock in this system. Hence, all the required tables data will be fetched first into buffer and lock will be acquired on all that tables. Then, all the modifications by queries will be done on the buffer data, instead of direct text file. Here, logs will also be stored in buffer instead of logs.txt file.
- Now, if rollback is performed in transaction, the buffer will be emptied and no changes will be applied to text files .
- If commit is performed in transaction, then all the buffer data will be updated into text files of the tables and logs.
- Lastly, after transaction is completed, lock manager will release all lock and a status message will be displayed in console.

A1 - A1MyDBMS/Data/A1db/logs.txt - Eclipse IDE

File Edit Navigate Search Project Run Window Help

Project Explorer X

- A1MyDBMS
 - JRE System Library [JavaSE-17]
 - src
 - a1.mydbms
 - MyDBMSApplication.java
 - a1.mydbms.model
 - a1.mydbms.service
 - a1.mydbms.util
 - Data
 - A1db
 - logs.txt
 - person.txt
 - student.txt
 - usercredentials.txt

Console X

```

MyDBMSApplication [Java Application] D:\Eclipse\MyDBMS\MyDBMSApplication\bin\org.eclipse.justj.openjdk.java.full.win32
insert into student (id, name, birthdate) values (40, Roshni, 20/10/1999);
student: Insert executed

Enter Query:
insert into student (id, name, birthdate) values (50, Dhruvi, 17/05/2023);
student: Insert executed

Enter Query:
create table person (user varchar(20), marks int(5), primary key (user));
person: Table Created

Enter Query:
insert into person (user, marks) values (testuser, 20);
person: Insert executed

Enter Query:
insert into person (user, marks) values (newuser, 30);
person: Insert executed

Enter Query:
Invalid query

Enter Query:
begin transaction;
|
```

Figure 56: executed begin transaction query

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS" with files like "MyDBMSApplication.java", "a1.mydbms", "a1.mydbms.model", "a1.mydbms.service", and "a1.mydbms.util".
- Console:** Displays database queries and their execution results.
- Logs:** Shows log entries from "logs.txt" and "student.txt".
- System Bar:** Includes weather information ("20°C Mostly sunny"), system icons, and a date/time stamp ("28-10-2023 15:46").

```

Enter Query:
insert into student (id, name, birthdate) values (50, Dhruvi, 17/05/2023);
student: Insert executed

Enter Query:
create table person (user varchar(20), marks int(5), primary key (user));
person: Table Created

Enter Query:
insert into person (user, marks) values (testuser, 20);
person: Insert executed

Enter Query:
insert into person (user, marks) values (newuser, 30);
person: Insert executed

Enter Query:
Invalid query

Enter Query:
begin transaction;
update student set name = Bhavee where id = 10;
delete from person;
update student set birthdate = 11/11/2023 where name =
Roshni;

```

Figure 57: System allows to add more queries after begin transaction

Transaction with Rollback

As shown in below screenshot, query changes are not applied to the table text files as transaction is rollbacked (i.e. no data is updated in student or person table). Also, log.txt is not updated.

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS" with files like "MyDBMSApplication.java", "a1.mydbms", "a1.mydbms.model", "a1.mydbms.service", and "a1.mydbms.util".
- Console:** Displays database queries and their execution results.
- Logs:** Shows log entries from "logs.txt" and "student.txt".
- System Bar:** Includes weather information ("20°C Mostly sunny"), system icons, and a date/time stamp ("28-10-2023 15:47").

```

key (user));
person: Table Created

Enter Query:
insert into person (user, marks) values (testuser, 20);
person: Insert executed

Enter Query:
insert into person (user, marks) values (newuser, 30);
person: Insert executed

Enter Query:
Invalid query

Enter Query:
begin transaction;
update student set name = Bhavee where id = 10;
delete from person;
update student set birthdate = 11/11/2023 where name =
Roshni;
delete from student where name = Dhruvi;
rollback;
end transaction;
Transaction completed
All locks released

Enter Query:

```

Figure 58: transaction rollback scenario

Transaction with Commit

As shown in below screenshot, query changes are applied to the table text files as transaction is committed (i.e. data is updated in student and person table). Also, log.txt is updated.

```

A1 - A1MyDBMS/Data/A1db/student.txt - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Project Explorer
A1MyDBMS
  JRE System Library [JavaSE-17]
  src
    a1.mydbms
      MyDBMSApplication.java
    a1.mydbms.model
    a1.mydbms.service
    a1.mydbms.util
  Data
    A1db
      logs.txt
      person.txt
      student.txt
      usercredentials.txt

Console
MyDBMSApplication [Java Application] D:\Eclipse\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32
Invalid query

Enter Query:
begin transaction;
update student set name = Bhavee where id = 10;
delete from person;
update student set birthdate = 11/11/2023 where name =
Roshni;
delete from student where name = Dhruvi;
rollback;
end transaction;
Transaction completed
All locks released

Enter Query:
begin transaction;
update student set name = Bhavee where id = 10;
delete from person;
update student set birthdate = 11/11/2023 where name =
Roshni;
delete from student where name = Dhruvi;
commit;
end transaction;
Transaction completed
All locks released

Enter Query:

```

Figure 59: transaction commit scenario, student and log tables updated

```

A1 - A1MyDBMS/Data/A1db/person.txt - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Project Explorer
A1MyDBMS
  JRE System Library [JavaSE-17]
  src
    a1.mydbms
      MyDBMSApplication.java
    a1.mydbms.model
    a1.mydbms.service
    a1.mydbms.util
  Data
    A1db
      logs.txt
      person.txt
      student.txt
      usercredentials.txt

Console
MyDBMSApplication [Java Application] D:\Eclipse\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32
Invalid query

Enter Query:
begin transaction;
update student set name = Bhavee where id = 10;
delete from person;
update student set birthdate = 11/11/2023 where name =
Roshni;
delete from student where name = Dhruvi;
rollback;
end transaction;
Transaction completed
All locks released

Enter Query:
begin transaction;
update student set name = Bhavee where id = 10;
delete from person;
update student set birthdate = 11/11/2023 where name =
Roshni;
delete from student where name = Dhruvi;
commit;
end transaction;
Transaction completed
All locks released

Enter Query:

```

Figure 60: transaction commit scenario, person and log tables updated

- Now, I am successfully logging out.

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with files like usercredentials.txt, student.txt, person.txt, and logs.txt.
- logs.txt Content:**

```

1create table : student : Roshni
2insert into : student : Roshni
3insert into : student : Roshni
4insert into : student : Roshni
5insert into : student : Roshni
6insert into : student : Roshni
7create table : person : Roshni
8insert into : person : Roshni
9insert into : person : Roshni
10update : student : Roshni
11delete from : person : Roshni
12update : student : Roshni
13delete from : student : Roshni

```
- Console Output:**

```

<terminated> MyDBMSApplication [Java Application] D:\Eclipse\workspace\org.eclipse.justj.openjdk.hotspace
delete from student where name = Dhrushi;
rollback;
end transaction;
Transaction completed
All locks released

Enter Query:
begin transaction;
update student set name = Bhavee where id = 10;
delete from person;
update student set birthdate = 11/11/2023 where name =
Roshni;
delete from student where name = Dhrushi;
commit;
end transaction;
Transaction completed
All locks released

Enter Query:
stop

Please select from below option :
1. Login
2. Register
3. Logout
3
| Successfully logged out

```

Figure 61: Logging out

Multi user system

- As shown in below screenshots, I am registering and logging in as new user. Also, new data will be inserted in the usercredentials tables.

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with files like usercredentials.txt, student.txt, person.txt, and logs.txt.
- logs.txt Content:**

```

1Roshni cec49593ca2c784c3ad8b3ea443ff642
2NewTestUser 098f6bcd4621d373cade4e832627b4f6
3

```
- Console Output:**

```

MyDBMSApplication [Java Application] D:\Eclipse\workspace\org.eclipse.justj.openjdk.hotspace
Please select from below option :
1. Login
2. Register
3. Logout
2

Enter new userId:
NewTestUser
Enter new password:
test
Enter captcha (HSQmdrh):
HSQmdrh
Registered

Please select from below option :
1. Login
2. Register
3. Logout

```

Figure 62: Registering as new user

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS".
- Console:** Displays the following interaction:


```

MyDBMSApplication [Java Application] D:\Eclipse\workspace\org.eclipse.justj.openjdk.hotspot.jtreg\run\logs\jtreg.log
1. Login
2. Register
3. Logout
2

Enter new userId:
NewTestUser
Enter new password:
test
Enter captcha (HSQmdrh):
HSQmdrh
Registered

Please select from below option :
1. Login
2. Register
3. Logout
1

Enter userId:
NewTestUser
Enter password:
test
Enter captcha (5LG0uHU):
5LG0uHU
Press 'stop' to exit
Enter Query:
      
```
- Logs.txt:** Shows database log entries.

Figure 63: Login as new user

- Now, I am performing the delete and update query on student table. In this case, query is executed and log file will be updated with the new user id (i.e. NewTestUser) as shown in below screenshots.

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "A1MyDBMS".
- Console:** Displays the following interaction:


```

MyDBMSApplication [Java Application] D:\Eclipse\workspace\org.eclipse.justj.openjdk.hotspot.jtreg\run\logs\jtreg.log
Enter captcha (HSQmdrh):
HSQmdrh
Registered

Please select from below option :
1. Login
2. Register
3. Logout
1

Enter userId:
NewTestUser
Enter password:
test
Enter captcha (5LG0uHU):
5LG0uHU
Press 'stop' to exit
Enter Query:
select * from student
id | birthdate | name
-----+
10 | 06/07/2023 | Bhavee
20 | 11/11/2023 | Roshni
30 | 04/09/2019 | Bhavee
40 | 11/11/2023 | Roshni
      
```
- Logs.txt:** Shows database log entries.

Figure 64: before updating student as new user

Project Explorer

- A1MyDBMS
 - JRE System Library [JavaSE-17]
 - src
 - a1.mydbms
 - MyDBMSApplication.java
 - a1.mydbms.model
 - a1.mydbms.service
 - a1.mydbms.util
 - Data
 - A1db
 - logs.txt
 - person.txt
 - student.txt
 - usercredentials.txt

Console

```

MyDBMSApplication [Java Application] D:\Eclipse\workspace\org.eclipse.jdt.openjdk.hotspot.jre.full.x86_64\bin
Enter password:
test
Enter captcha (5LG0uHU):
5LG0uHU

Press 'stop' to exit
Enter Query:
select * from student
id | birthdate | name
-----
```

10	06/07/2023	Bhavee
20	11/11/2023	Roshni
30	04/09/2019	Bhavee
40	11/11/2023	Roshni

```

Enter Query:
delete from student where id = 30;
student: Delete executed

Enter Query:
select * from student
id | birthdate | name
-----
```

10	06/07/2023	Bhavee
20	11/11/2023	Roshni
40	11/11/2023	Roshni

```

Enter Query:
```

Figure 65: record deleted from student table, log file updated with new user id

Project Explorer

- A1MyDBMS
 - JRE System Library [JavaSE-17]
 - src
 - a1.mydbms
 - MyDBMSApplication.java
 - a1.mydbms.model
 - a1.mydbms.service
 - a1.mydbms.util
 - Data
 - A1db
 - logs.txt
 - person.txt
 - student.txt
 - usercredentials.txt

Console

```

MyDBMSApplication [Java Application] D:\Eclipse\workspace\org.eclipse.jdt.openjdk.hotspot.jre.full.x86_64\bin
Press 'stop' to exit
Enter Query:
select * from student
id | birthdate | name
-----
```

10	06/07/2023	Bhavee
20	11/11/2023	Roshni
30	04/09/2019	Bhavee
40	11/11/2023	Roshni

```

Enter Query:
delete from student where id = 30;
student: Delete executed

Enter Query:
select * from student
id | birthdate | name
-----
```

10	06/07/2023	Bhavee
20	11/11/2023	abc
30	04/09/2019	Bhavee
40	11/11/2023	Roshni

```

Enter Query:
update student set name = abc where id = 20;
student: Update executed

Enter Query:
```

Figure 66: record updated in student table, log file updated with new user id

- Now, I am trying to register again using the already existing userid. In this case, system will not allow me to register and throws ‘user already exists’ error message, as every user has unique userid. Moreover, usercredentials.txt is also not updated in this case.

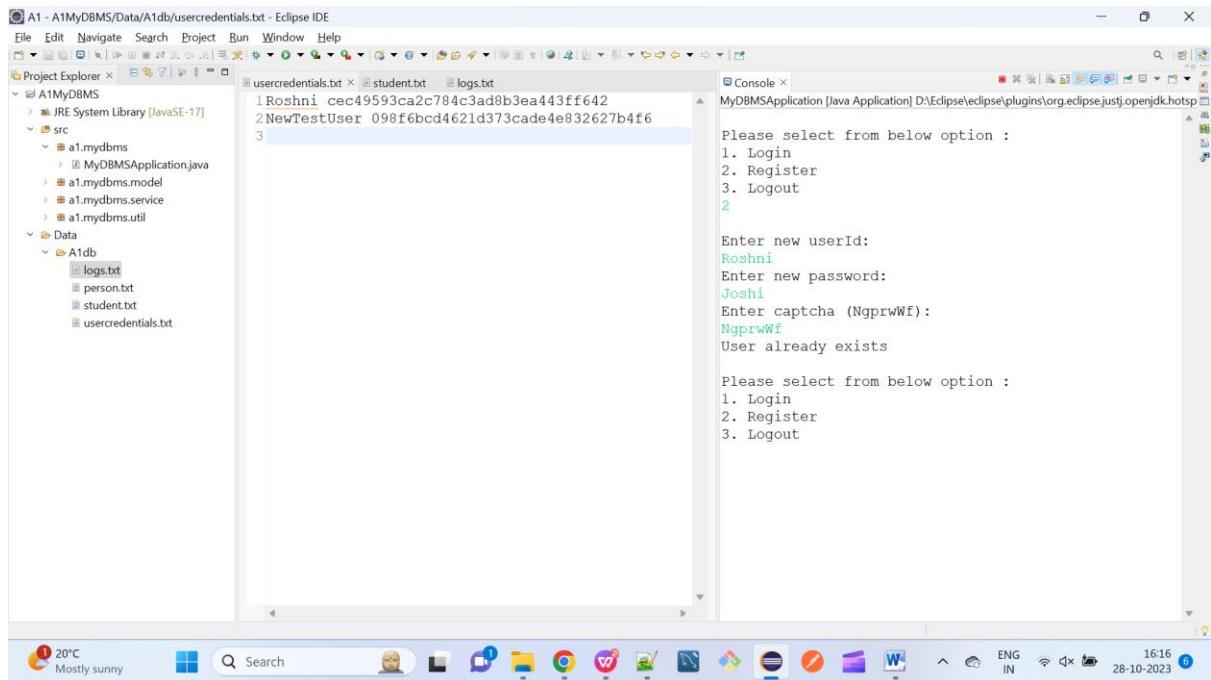


Figure 67: Trying to register using existing user id