

**JAVA AWT BASED- Online MOOC's year wise student
database management system - SQL CONNECTIVITY
USING JDBC**

A

Report

*Submitted in partial fulfilment of the
Requirements for the award of the Degree of*

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

S.Hemanth Kumar <1602-18-737-072>



Department of Information Technology

Vasavi College of Engineering (Autonomous)

Ibrahimbagh, Hyderabad-31

2020

BONAFIDE CERTIFICATE

This to Certify that the project report titled
**"Online MOOC's year wise student database
management system"** project work of
Mr.S.HEMANTH KUMAR bearing Roll.no:**1602-18-
737-072** who carried out this project under my
supervision in the IV semester for the academic
year 2019-2020.

Signature

external examiner

Signature

internal examiner

B.LEELAVATHY

Assistant Professor

Department of Information Technology

ABSTRACT

Online MOOC's year wise Student Management System is a database management system which is helpful for students as well as the Mooc's providers. In the current system all the activities are done manually. It is very time consuming and costly. Our online Mooc's Student Management System deals with the various activities related to the students and mooc's provider. In the database can register as a user and user has of two types, student and administrator. Administrator has the power to add new user and can edit and delete a user. A student can register as user and can add edit and delete his profile. The administrator can add edit and delete marks for the student. All the users can see the marks.

INTRODUCTION

➤ REQUIREMENTS FOR ONLINE MOOC'S YEAR WISE DATABASE MANAGEMENT SYSTEM:

List of tables :

- Online MOOC's provider
- Courses
- Student
- Enrolls
- Assignments
- Results

List of attributes with their domain types:

ENTITY	ATTRIBUTES	DOMAIN
Online Mooc's provider	1. P_id 2. P_name 3. Type 4. Headquartes 5. Found	Number(5) Varchar2(20) Varchar2(20) Varchar2(20) Number(5)
Courses	1. C_id 2. C_name 3. Duration 4. Min_grade 5. Price 6. Status	Number(5) Varchar2(20) Varchar2(20) Char(5) Number(5) Varchar(10)
Student	1. S_id 2. First_name 3. Last_name 4. User_name 5. Password	Number(5) Varchar(10) Varchar(10) Varchar(10) Varchar(10)
Enrolls	1. S_id 2. C_id 3. Year	Number(5) Number(5) Number(5)
Assignments	1. A_id 2. C_id 3. Deadline 4. Score	Number(5) Number(5) Varchar(10) Number(5)

Results	1. S_id 2. C_id 3. A_id 4. Score 5. Grade	Number(5) Number(5) Number(5) Number(5) Char(5)
---------	---	---

- Online mooc's provider offers as many courses to the students who want to persue the course so, it is a one to many mapping. As it is not necessary that one provider should offer only one course.
- Student enrolls into courses, it is many to many mapping as any number of students can enrol into any number of courses.
- Student submits assignment, it is one to one as one student should submit one assignment as per the provider instructions.
- Student gets results if he/submit the assignments and attend the exam, it is one to one mapping as one student get only one result.

➤ **SPECIFIC GOAL OF THE PROJECT:**

The main goal to be achieved through this project was to provide a facility to the Online MOOC's providers to display the details of various courses, students who enroll into those courses, assignments they do, and the results they get based on their assignment submissions online.

The project also ensure that the details of the students are confidential and are stored in the database.

SQL particular Online MOOC's provider, courses, student, assignments and results can be executed.

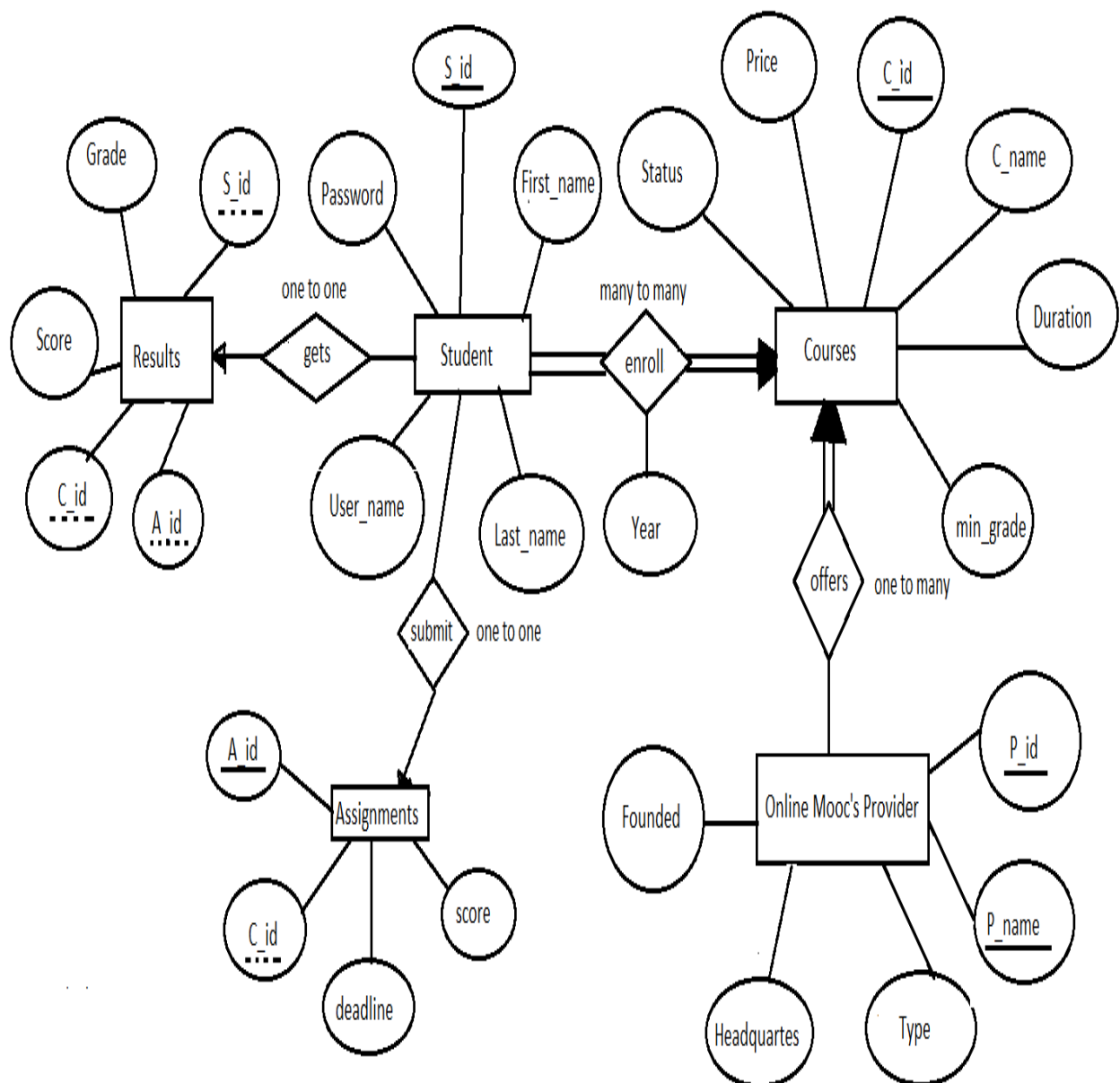
➤ **Architecture and technology used:**

SQL Plus is the most basic Oracle Database utility with a basic command-line interface, commonly used by users, administrators and programmers.

The interface of SQL Plus is used for creating the database. DDL and DML commands are implemented for operations being executed. The details of various Online MOOC's provider, courses, student, assignments, and results are stored in the form of tables in the database.

Eclipse is an integrated development environment(IDE) used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages via plug-ins, including Erlang, JavaScripts etc.

The front end application code is written in "**Java**" using Eclipse. The portal for front end application is designed through Eclipse, runs and has the capacity to connect with the database which has data inserted using SQL.

➤ **DESIGN:**i) ER DIAGRAM:

MAPPING CARDINALITIES AND PARTICIPATION

CONSTRAINTS:

- Online mooc's provider offers as many courses to the students who want to pursue the course so, it is a one to many mapping. As it is not necessary that one provider should offer only one course.
- Student enrolls into courses, it is many to many mapping as any number of students can enrol into any number of courses.
- Student submits assignment, it is one to one as one student should submit one assignment as per the provider instructions.
- Student gets results if he/submit the assignments and attend the exam, it is one to one mapping as one student get only one result.

DDL Commands:

Creating all the required tables.

```
SQL> create table courses( C_id number(5),C_name varchar2(20),Duration Varchar(20), min_grade Char(10),Status char(10),price Number(3,2));
```

Table created.

```
SQL> desc courses;
```

Name	Null?	Type
C_ID		NUMBER(5)
C_NAME		VARCHAR2(20)
DURATION		VARCHAR2(20)
MIN_GRADE		CHAR(10)
STATUS		CHAR(10)
PRICE		NUMBER(3,2)

```
SQL> create table student( S_id number(5),First_name varchar2(20),Last_name Varchar2(20), User_name varchar2(10),Password varchar2(10),price Number(3,2));
```

Table created.

```
SQL> desc student;
```

Name	Null?	Type
S_ID		NUMBER(5)
FIRST_NAME		VARCHAR2(20)
LAST_NAME		VARCHAR2(20)
USER_NAME		VARCHAR2(10)
PASSWORD		VARCHAR2(10)
PRICE		NUMBER(3,2)

```
SQL> create table results( S_id number(5),C_id number(5),Score varchar2(20),Grade char(20));
```

Table created.

```
SQL> desc results;
```

Name	Null?	Type
S_ID		NUMBER(5)
C_ID		NUMBER(5)
SCORE		VARCHAR2(20)
GRADE		CHAR(20)

```
SQL>
```

DBMS ASSIGNMENT 2

```
SQL> desc assignments;
```

Name	Null?	Type
A_ID		NUMBER(5)
C_ID		NUMBER(5)
DEADLINE		VARCHAR2(20)
SCORE		CHAR(20)

```
SQL> create table enrolls(S_id Number(5),C_id Number(5),Year Number(5));
```

Table created.

```
SQL> desc enrolls;
```

Name	Null?	Type
S_ID		NUMBER(5)
C_ID		NUMBER(5)
YEAR		NUMBER(5)

```
SQL> alter table Online_Moocs_Provider add primary key(P_id);
```

Table altered.

```
SQL> desc Online_Moocs_Provider;
```

Name	Null?	Type
P_ID	NOT NULL	NUMBER(5)
P_NAME		VARCHAR2(20)
TYPE		VARCHAR2(20)
HEADQUARTERS		VARCHAR2(20)
FOUNDED		NUMBER(5)

```
SQL> alter table courses add primary key(C_id);
```

Table altered.

```
SQL> alter table Student add primary key (S_id);
```

Table altered.

```
SQL> alter table Results add foreign key (S id,C id) references
```

Enforcing constraints to primary, foreign key constraints:

```

SQL> desc Online_Moocs_Provider;
Name                                         Null?    Type
-----
P_ID                                         NOT NULL NUMBER(5)
P_NAME                                       VARCHAR2(20)
TYPE                                         VARCHAR2(20)
HEADQUARTERS                               VARCHAR2(20)
FOUNDED                                     NUMBER(5)

SQL> alter table courses add primary key(C_id);

Table altered.

SQL> alter table Student add primary key (S_id);

Table altered.

SQL> alter table Results add foreign key (S_id,C_id) references student,courses;
alter table Results add foreign key (S_id,C_id) references student,courses
*
ERROR at line 1:
ORA-01735: invalid ALTER TABLE option

SQL> alter table Results add foreign key (S_id) references student;

Table altered.

SQL> alter table Results add foreign key (C_id) references courses;

Table altered.

SQL> alter table enrolls add foreign key (C_id) references courses;

Table altered.

SQL> alter table enrolls add foreign key (S_id) references student;

Table altered.

SQL> alter table Assignments add foreign key (C_id) references courses;

Table altered.

```

DML commands:**Inserting values into the tables.**

```

SQL> Insert into Online_Moocs_Provider values(&P_id,&P_name,&Type,&headquartes,&founded');
Enter value for p_id: 101
Enter value for p_name: SWAYAM
Enter value for type: Non-profit
Enter value for headquartes: India
Enter value for founded: 2017
old 1: Insert into Online_Moocs_Provider values(&P_id,&P_name,&Type,&headquartes,&founded')
new 1: Insert into Online_Moocs_Provider values(101,'SWAYAM','Non-profit','India','2017')

1 row created.

SQL> /
Enter value for p_id: 102
Enter value for p_name: Udemy
Enter value for type: Commercial
Enter value for headquartes: USA
Enter value for founded: 2010
old 1: Insert into Online_Moocs_Provider values(&P_id,&P_name,&Type,&headquartes,&founded')
new 1: Insert into Online_Moocs_Provider values(102,'Udemy','Commercial','USA','2010')

1 row created.

SQL> /
Enter value for p_id: 103
Enter value for p_name: Khanacadamy
Enter value for type: Non-profit
Enter value for headquartes: USA
Enter value for founded: 2006
old 1: Insert into Online_Moocs_Provider values(&P_id,&P_name,&Type,&headquartes,&founded')
new 1: Insert into Online_Moocs_Provider values(103,'Khanacadamy','Non-profit','USA','2006')

1 row created.

SQL> /
Enter value for p_id: 104
Enter value for p_name: Coursera
Enter value for type: Commercial
Enter value for headquartes: USA
Enter value for founded: 2012
old 1: Insert into Online_Moocs_Provider values(&P_id,&P_name,&Type,&headquartes,&founded')
new 1: Insert into Online_Moocs_Provider values(104,'Coursera','Commercial','USA','2012')

1 row created.

```

```

SQL> /
Enter value for p_id: 105
Enter value for p_name: Udacity
Enter value for type: Commercial
Enter value for headquartes: USA
Enter value for founded: 2012
old 1: Insert into Online_Moocs_Provider values(&P_id,&P_name,&Type,&headquartes,&founded')
new 1: Insert into Online_Moocs_Provider values(105,'Udacity','Commercial','USA','2012')

1 row created.

```

DBMS ASSIGNMENT 2

```
SQL> select * from Online_MOOCs_provider;
```

P_ID	P_NAME	TYPE	HEADQUARTERS
FOUNDED			
101 2017	SWAYAM	Non-profit	India
102 2010	Udemy	Commercial	USA
103 2006	Khanacademy	Non-profit	USA
P_ID	P_NAME	TYPE	HEADQUARTERS
FOUNDED			
104 2012	Coursera	Commercial	USA
105 2012	Udacity	Commercial	USA

SQL Plus

```
SQL> desc assignments;
```

Name	Null?	Type
A_ID		NUMBER(5)
C_ID		NUMBER(5)
DEADLINE		VARCHAR2(20)
SCORE		CHAR(20)

```
SQL> alter table assignments add primary key(A_id);
```

Table altered.

```
SQL> desc assignments;
```

Name	Null?	Type
A_ID	NOT NULL	NUMBER(5)
C_ID		NUMBER(5)
DEADLINE		VARCHAR2(20)
SCORE		CHAR(20)

```
SQL> alter table results add foreign key(A_id) references assignments;
alter table results add foreign key(A_id) references assignments
```

```
ERROR at line 1:
ORA-00904: "A_ID": invalid identifier
```

```
SQL> alter table results add (A_id Number(5));
```

Table altered.

```
SQL> alter table results add foreign key(A_id) references assignments;
```

Table altered.

```
SQL> desc results;
```

Name	Null?	Type
S_ID		NUMBER(5)
C_ID		NUMBER(5)
SCORE		VARCHAR2(20)
GRADE		CHAR(20)
A_ID		NUMBER(5)

```
SQL> desc student;
```

Name	Null?	Type
S_ID	NOT NULL	NUMBER(5)
FIRST_NAME		VARCHAR2(20)
LAST_NAME		VARCHAR2(20)
USER_NAME		VARCHAR2(10)
PASSWORD		VARCHAR2(10)
PRICE		NUMBER(3,2)

```
SQL> alter table students drop column price;
```

```
alter table students drop column price
```

```
ERROR at line 1:
```

Windows taskbar showing search bar, task icons, and system tray with date 12-02-2020 and time 06:56.

DBMS ASSIGNMENT 2

```
SQL> insert into courses values(&c_id,&c_name','&duration','&min_grade','&status',&price);
Enter value for c_id: 2
Enter value for c_name: DAA
Enter value for duration: 8weeks
Enter value for min_grade: D
Enter value for status: yes
Enter value for price: 1000
old 1: insert into courses values(&c_id,&c_name','&duration','&min_grade','&status',&price)
new 1: insert into courses values(2,'DAA','8weeks','D','yes',1000)
```

1 row created.

```
SQL> /
Enter value for c_id: 3
Enter value for c_name: SocialNetworking
Enter value for duration: 10weeks
Enter value for min_grade: C
Enter value for status: No
Enter value for price: 1500
old 1: insert into courses values(&c_id,&c_name','&duration','&min_grade','&status',&price)
new 1: insert into courses values(3,'SocialNetworking','10weeks','C','No',1500)
```

1 row created.

```
SQL> /
Enter value for c_id: 4
Enter value for c_name: AI
Enter value for duration: 20weeks
Enter value for min_grade: D
Enter value for status: yes
Enter value for price: 3500
old 1: insert into courses values(&c_id,&c_name','&duration','&min_grade','&status',&price)
new 1: insert into courses values(4,'AI','20weeks','D','yes',3500)
```

1 row created.

```
SQL> /
Enter value for c_id: 5
Enter value for c_name: C_programming
Enter value for duration: 8weeks
Enter value for min_grade: E
Enter value for status: yes
Enter value for price: free
old 1: insert into courses values(&c_id,&c_name','&duration','&min_grade','&status',&price)
new 1: insert into courses values(5,'C_programming','8weeks','E','yes',free)
insert into courses values(5,'C_programming','8weeks','E','yes',*)
```

ERROR at line 1:
ORA-00984: column not allowed here

```
SQL> /
Enter value for c_id: 5
Enter value for c_name: C_programming
```

```
SQL> /
Enter value for c_id: 5
Enter value for c_name: C_programming
Enter value for duration: 8weeks
Enter value for min_grade: E
Enter value for status: yes
Enter value for price: 0
old 1: insert into courses values(&c_id,&c_name','&duration','&min_grade','&status',&price)
new 1: insert into courses values(5,'C_programming','8weeks','E','yes',0)
```

1 row created.

```
SQL> select *from courses;
```

C_ID	C_NAME	DURATION	MIN_GRADE	STATUS
2	DAA	8weeks	D	yes
3	SocialNetworking	10weeks	C	No
4	AI	20weeks	D	yes
5	C_programming	8weeks	E	yes
1	DBMS	12weeks	D	YES

DBMS ASSIGNMENT 2

SQL Plus

```

61 Abhiraj          dusari          dusariabhi @bhir@j
SQL> insert into student values (&s_id,&first_name','&last_name','&user_name','&password');
Enter value for s_id: 72
Enter value for first_name: Hemanth
Enter value for last_name: Sherla
Enter value for user_name: sherla001
Enter value for password: hem@nth
old 1: insert into student values (&s_id,&first_name','&last_name','&user_name','&password')
new 1: insert into student values (72,'Hemanth','Sherla','sherla001','hem@nth')

1 row created.

SQL> /
Enter value for s_id: 83
Enter value for first_name: mohammad
Enter value for last_name: razzaq
Enter value for user_name: mrzzaq
Enter value for password: r@zz@q
old 1: insert into student values (&s_id,&first_name','&last_name','&user_name','&password')
new 1: insert into student values (83,'mohammad','razzaq','mrzzaq','r@zz@q')

1 row created.

SQL> /
Enter value for s_id: 94
Enter value for first_name: mallik
Enter value for last_name: reddy
Enter value for user_name: saimallik
Enter value for password: s@im@llik
old 1: insert into student values (&s_id,&first_name','&last_name','&user_name','&password')
new 1: insert into student values (94,'mallik','reddy','saimallik','s@im@llik')

1 row created.

SQL> /
Enter value for s_id: 105
Enter value for first_name: sujitha
Enter value for last_name: tadi
Enter value for user_name: sujithatadi
Enter value for password: sujith@
old 1: insert into student values (&s_id,&first_name','&last_name','&user_name','&password')
new 1: insert into student values (105,'sujitha','tadi','sujithatadi','sujith@')
insert into student values (105,'sujitha','tadi','sujithatadi','sujith@')
*
ERROR at line 1:
ORA-12899: value too large for column "HEMANTH"."STUDENT"."USER_NAME" (actual:
11, maximum: 10)

SQL> /
Enter value for s_id: 105
Enter value for first_name: sujitha
Enter value for last_name: tadi
Enter value for user_name: tsujitha
Enter value for password: sujith@
old 1: insert into student values (&s_id,&first_name','&last_name','&user_name','&password')
new 1: insert into student values (105,'sujitha','tadi','tsujitha','sujith@')

1 row created.

SQL> select *from student;

   S_ID FIRST_NAME   LAST_NAME   USER_NAME   PASSWORD
-----
    61 Abhiraj      dusari      dusariabhi @bhir@j
    72 Hemanth      Sherla      sherla001  hem@nth
    83 mohammad      razzaq      mrzzaq     r@zz@q
    94 mallik        reddy       saimallik  s@im@llik
   105 sujitha      tadi        tsujitha   sujith@

SQL> desc enrolls;
Name                                     Null?   Type
-----

```

```
YEAR                                NUMBER(5)

SQL> insert into enrolls values(&s_id,&c_id,&year);
Enter value for s_id: 72
Enter value for c_id: 4
Enter value for year: 2018
old 1: insert into enrolls values(&s_id,&c_id,&year)
new 1: insert into enrolls values(72,4,2018)

1 row created.

SQL> /
Enter value for s_id: 94
Enter value for c_id: 5
Enter value for year: 2020
old 1: insert into enrolls values(&s_id,&c_id,&year)
new 1: insert into enrolls values(94,5,2020)

1 row created.

SQL> /
Enter value for s_id: 105
Enter value for c_id: 3
Enter value for year: 2019
old 1: insert into enrolls values(&s_id,&c_id,&year)
new 1: insert into enrolls values(105,3,2019)

1 row created.

SQL> /
Enter value for s_id: 61
Enter value for c_id: 1
Enter value for year: 2019
old 1: insert into enrolls values(&s_id,&c_id,&year)
new 1: insert into enrolls values(61,1,2019)

1 row created.

SQL> /
Enter value for s_id: 83
Enter value for c_id: 2
Enter value for year: 2019
old 1: insert into enrolls values(&s_id,&c_id,&year)
new 1: insert into enrolls values(83,2,2019)

1 row created.

SQL> select * from enrolls;

  S_ID    C_ID    YEAR
-----
    72      4    2018
    94      5    2020
   105      3    2019
    61      1    2019
    83      2    2019
```


DBMS ASSIGNMENT 2

```

SQL> insert into assgnments values(&a_id,&c_id,&'deadline','&score');
Enter value for a_id: 101
Enter value for c_id: 4
Enter value for deadline: 12-2-2018
Enter value for score: 100
old 1: insert into assgnments values(&a_id,&c_id,&'deadline','&score')
new 1: insert into assgnments values(101,4,'12-2-2018','100')
insert into assgnments values(101,4,'12-2-2018','100')
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> insert into assignments values(&a_id,&c_id,&'deadline','&score');
Enter value for a_id: 101
Enter value for c_id: 4
Enter value for deadline: 12-02-2018
Enter value for score: 100
old 1: insert into assignments values(&a_id,&c_id,&'deadline','&score')
new 1: insert into assignments values(101,4,'12-02-2018','100')

1 row created.

SQL> /
Enter value for a_id: 102
Enter value for c_id: 5
Enter value for deadline: 11-02-2020
Enter value for score: 100
old 1: insert into assignments values(&a_id,&c_id,&'deadline','&score')
new 1: insert into assignments values(102,5,'11-02-2020','100')

1 row created.

SQL> /
Enter value for a_id: 105
Enter value for c_id: 3
Enter value for deadline:
Enter value for score:
old 1: insert into assignments values(&a_id,&c_id,&'deadline','&score')
new 1: insert into assignments values(105,3,',','')

1 row created.

SQL> /
Enter value for a_id: 61
Enter value for c_id: 1
Enter value for deadline: 15-02-2019
Enter value for score: 80
old 1: insert into assignments values(&a_id,&c_id,&'deadline','&score')
new 1: insert into assignments values(61,1,'15-02-2019','80')

1 row created.

```

```

SQL> update assignments SET '&deadline' where a_id=105 and c_id=3;
Enter value for deadline: 12-02-2019
old 1: update assignments SET '&deadline' where a_id=105 and c_id=3
new 1: update assignments SET '12-02-2019' where a_id=105 and c_id=3
update assignments SET '12-02-2019' where a_id=105 and c_id=3
*
ERROR at line 1:
ORA-01747: invalid user.table.column, table.column, or column specification

SQL> update assignments SET deadline ='&deadline' where a_id=105 and c_id=3;
Enter value for deadline: 12-08-2019
old 1: update assignments SET deadline ='&deadline' where a_id=105 and c_id=3
new 1: update assignments SET deadline ='12-08-2019' where a_id=105 and c_id=3

1 row updated.

SQL> update assignments SET score =&score where a_id=105 and c_id=3;
Enter value for score: 100
old 1: update assignments SET score =&score where a_id=105 and c_id=3
new 1: update assignments SET score =100 where a_id=105 and c_id=3

1 row updated.

SQL> delete from assignments where a_id=23;

1 row deleted.

SQL> select * from assignments;

   A_ID    C_ID DEADLINE    SCORE
-----
    101         4 12-02-2018      100
    102         5 11-02-2020      100
    105         3 12-08-2019      100
     61         1 15-02-2019       80
     83         2 20-02-2019       75

```

DBMS ASSIGNMENT 2

```

SQL> desc results;
Name                                Null?    Type
-----
S_ID                                NUMBER(5)
C_ID                                NUMBER(5)
SCORE                              VARCHAR2(20)
GRADE                              CHAR(20)
A_ID                                NUMBER(5)

SQL> insert into results values(&s_id,&c_id,'&score','&char',&a_id);
Enter value for s_id: 72
Enter value for c_id: 4
Enter value for score: 92
Enter value for char: A
Enter value for a_id: 101
old 1: insert into results values(&s_id,&c_id,'&score','&char',&a_id)
new 1: insert into results values(72,4,'92','A',101)

1 row created.

SQL> /
Enter value for s_id: 94
Enter value for c_id: 5
Enter value for score: 82
Enter value for char: B
Enter value for a_id: 102
old 1: insert into results values(&s_id,&c_id,'&score','&char',&a_id)
new 1: insert into results values(94,5,'82','B',102)

1 row created.

SQL> /
Enter value for s_id: 105
Enter value for c_id: 3
Enter value for score: 95
Enter value for char: A
Enter value for a_id: 105
old 1: insert into results values(&s_id,&c_id,'&score','&char',&a_id)
new 1: insert into results values(105,3,'95','A',105)

1 row created.

SQL> /
Enter value for s_id: 61
Enter value for c_id: 1
Enter value for score: 91
Enter value for char: A
Enter value for a_id: 61
old 1: insert into results values(&s_id,&c_id,'&score','&char',&a_id)
new 1: insert into results values(61,1,'91','A',61)

1 row created.

SQL> /
Enter value for s_id: 83
Enter value for c_id: 2
Enter value for score: 67
Enter value for char: D
Enter value for a_id: 83
old 1: insert into results values(&s_id,&c_id,'&score','&char',&a_id)
new 1: insert into results values(83,2,'67','D',83)

1 row created.

SQL> select_

```

DBMS ASSIGNMENT 2

```

SQL> insert into results values(&s_id,&c_id,'&score','&char',&a_id);
Enter value for s_id: 72
Enter value for c_id: 4
Enter value for score: 92
Enter value for char: A
Enter value for a_id: 101
old 1: insert into results values(&s_id,&c_id,'&score','&char',&a_id)
new 1: insert into results values(72,4,'92','A',101)

1 row created.

SQL> /
Enter value for s_id: 94
Enter value for c_id: 5
Enter value for score: 82
Enter value for char: B
Enter value for a_id: 102
old 1: insert into results values(&s_id,&c_id,'&score','&char',&a_id)
new 1: insert into results values(94,5,'82','B',102)

1 row created.

SQL> /
Enter value for s_id: 105
Enter value for c_id: 3
Enter value for score: 95
Enter value for char: A
Enter value for a_id: 105
old 1: insert into results values(&s_id,&c_id,'&score','&char',&a_id)
new 1: insert into results values(105,3,'95','A',105)

1 row created.

SQL> /
Enter value for s_id: 61
Enter value for c_id: 1
Enter value for score: 91
Enter value for char: A
Enter value for a_id: 61
old 1: insert into results values(&s_id,&c_id,'&score','&char',&a_id)
new 1: insert into results values(61,1,'91','A',61)

1 row created.

SQL> /
Enter value for s_id: 83
Enter value for c_id: 2
Enter value for score: 67
Enter value for char: D
Enter value for a_id: 83
old 1: insert into results values(&s_id,&c_id,'&score','&char',&a_id)
new 1: insert into results values(83,2,'67','D',83)

1 row created.

SQL> select * from results;

  S_ID    C_ID SCORE      GRADE      A_ID
-----
    72      4  92         A        101
    94      5  82         B        102
   105      3  95         A        105
    61      1  91         A         61
    83      2  67         D         83

```

Implementation

➤ Front end programs:

1) Insert a Student:

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class InsertStudent extends Panel
{
    Button insertStudentButton;
    TextField sidText, fnameText, lnameText, unameText, passwordText;
    TextArea errorText;
    Connection connection;
    Statement statement;
    public InsertStudent()
    {
        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
        }
        catch (Exception e)
        {
            System.err.println("Unable to find and load driver");
            System.exit(1);
        }
        connectToDB();
    }
    public void connectToDB()
    {

```

```
try
{
    connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","hemanth","orac
le");

    statement = connection.createStatement();

}
catch (SQLException connectException)
{
    System.out.println(connectException.getMessage());
    System.out.println(connectException.getSQLState());
    System.out.println(connectException.getErrorCode());
    System.exit(1);
}
}

public void buildGUI()
{
    //Handle Insert Account Button
    insertStudentButton = new Button("Submit");
    insertStudentButton.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent e)
        {
            try
            {
                Statement statement = connection.createStatement();
```

DBMS ASSIGNMENT 2

```

        String query= "INSERT INTO student VALUES(" +
sidText.getText() + ", " + "" + fnameText.getText() + "," + "" + lnameText.getText() +
", "+""+unameText.getText()+", " +""+passwordText.getText()+""+"");

        int i = statement.executeUpdate(query);

        errorText.append("\nInserted " + i + " rows successfully");

    }

    catch (SQLException insertException)

    {

        displaySQLErrors(insertException);

    }

}

});

```

```

fnameText = new TextField(15);
sidText = new TextField(15);
lnameText = new TextField(15);
unameText = new TextField(15);
passwordText = new TextField(15);

```

```

errorText = new TextArea(10,40);
errorText.setEditable(false);

```

```

Panel first = new Panel();

first.setLayout(new GridLayout(5,2));

first.add(new Label("Student ID:"));

first.add(sidText);

first.add(new Label("FirstName:"));

```

```
first.add(fnameText);
first.add(new Label("LastName:"));
first.add(lnameText);
first.add(new Label("Username:"));
first.add(unameText);
first.add(new Label("Password:"));
first.add(passwordText);

first.setBounds(125,90,300,150);

Panel second = new Panel(new GridLayout(4, 1));
second.add(insertStudentButton);
second.setBounds(195,290,150,100);

Panel third = new Panel();
third.add(errorText);
third.setBounds(80,410,430,300);
setLayout(null);

add(first);
add(second);
add(third);

setSize(400,180);
setVisible(true);
}
```

```
private void displaySQLExceptions(SQLException e)
{
    errorText.append("\nSQLException: " + e.getMessage() + "\n");
    errorText.append("SQLState:  " + e.getSQLState() + "\n");
    errorText.append("VendorError: " + e.getErrorCode() + "\n");
}
```

```
public static void main(String[] args)
{
    InsertStudent ins = new InsertStudent();
    ins.buildGUI();
}
}
```

2)Delete a Student:

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class DeleteStudent extends Panel{

    Button deleteStudentButton;

    List studentIDList;

    TextField sidText, fnameText, lnameText, unameText,passwordText;

    TextArea errorText;
```


Connection connection;

Statement statement;

ResultSet rs;

public DeleteStudent()

{

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

}

catch (Exception e)

{

System.err.println("Unable to find and load driver");

System.exit(1);

}

connectToDB();

}

public void connectToDB()

{

try

{

connection =

DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","hemanth","oracle")

;

```
statement = connection.createStatement();

}

catch (SQLException connectException)

{

    System.out.println(connectException.getMessage());

    System.out.println(connectException.getSQLState());

    System.out.println(connectException.getErrorCode());

    System.exit(1);

}

}

private void loadStudent()

{

    try

    {

        rs = statement.executeQuery("SELECT * FROM STUDENT");

        while (rs.next())

        {

            studentIDList.add(rs.getString("S_ID"));

        }

    }

    catch (SQLException e)

    {

        displaySQLErrors(e);

    }

}
```

```
    }  
}  
  
public void buildGUI()  
{  
    studentIDList = new List(10);  
    loadStudent();  
    add(studentIDList);  
  
    //When a list item is selected populate the text fields  
    studentIDList.addItemListener(new ItemListener()  
    {  
        public void itemStateChanged(ItemEvent e)  
        {  
            try  
            {  
                rs = statement.executeQuery("SELECT * FROM  
STUDENT");  
  
                while (rs.next())  
                {  
                    if  
(rs.getString("S_ID").equals(studentIDList.getSelectedItem()))  
  
                        break;  
                }  
  
                if (!rs.isAfterLast())
```

DBMS ASSIGNMENT 2

```
{

    sidText.setText(rs.getString("S_ID"));

    fnameText.setText(rs.getString("First_NAME"));

    lnameText.setText(rs.getString("Last_Name"));

    unameText.setText(rs.getString("User_Name"));

passwordText.setText(rs.getString("Password"));

}

}

catch (SQLException selectException)

{

    displaySQLErrors(selectException);

}

}

});

deleteStudentButton = new Button("Delete");

deleteStudentButton.addActionListener(new ActionListener()

{

    public void actionPerformed(ActionEvent e)

    {

        try

        {

            Statement statement = connection.createStatement();
```

DBMS ASSIGNMENT 2

```

student WHERE S_ID = "
                                + studentIDList.getSelectedItem());

errorText.append("\nDeleted " + i + " rows
successfully");

sidText.setText(null);

fnameText.setText(null);

lnameText.setText(null);

unameText.setText(null);

passwordText.setText(null);

studentIDList.removeAll();

loadStudent();

}

catch (SQLException insertException)

{

    displaySQLErrors(insertException);

}

}

});

sidText = new TextField(15);

fnameText = new TextField(15);

lnameText = new TextField(15);

unameText = new TextField(15);

passwordText= new TextField(15);

```

```
errorText = new TextArea(10, 40);

errorText.setEditable(false);


Panel first = new Panel();

first.setLayout(new GridLayout(6, 1));

first.add(new Label("Student ID:"));

first.add(sidText);

sidText.setEditable(false);

first.add(new Label("FirstName:"));

first.add(fnameText);

fnameText.setEditable(false);

first.add(new Label("LastName:"));

first.add(lnameText);

lnameText.setEditable(false);

first.add(new Label("Username:"));

first.add(unameText);

unameText.setEditable(false);

first.add(new Label("password:"));

first.add(passwordText);

passwordText.setEditable(false);
```

DBMS ASSIGNMENT 2

```
Panel second = new Panel(new GridLayout(4, 1));

second.add(deleteStudentButton);


Panel third = new Panel();

third.add(errorText);


add(first);

add(second);

add(third);

setSize(450, 600);

setLayout(new FlowLayout());

setVisible(true);

}

private void displaySQLErrors(SQLException e)

{

    errorText.append("\nSQLException: " + e.getMessage() + "\n");

    errorText.append("SQLState:   " + e.getSQLState() + "\n");

    errorText.append("VendorError: " + e.getErrorCode() + "\n");

}

public static void main(String[] args)

{

    DeleteStudent dels = new DeleteStudent();

    dels.buildGUI();

}
```

```
}
```

3) Update a Student:

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import java.sql.*;
```

```
public class UpdateStudent extends Panel
```

```
{
```

```
    Button updateStudentButton;
```

```
    List studentIDList;
```

```
    TextField sidText,fnameText,lnameText,unameText,passwordText;
```

```
    TextArea errorText;
```

```
    Connection connection;
```

```
    Statement statement;
```

```
    ResultSet rs;
```

```
    public UpdateStudent()
```

```
    {
```

```
        try
```

```
        {
```

```
            Class.forName("oracle.jdbc.driver.OracleDriver");
```

```
        }
```

```
        catch (Exception e)
```

```
        {
```


DBMS ASSIGNMENT 2

```
        System.err.println("Unable to find and load driver");

        System.exit(1);

    }

    connectToDB();

}

public void connectToDB()

{

    try

    {

        connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","hemanth","oracle")
;

        statement = connection.createStatement();

    }

    catch (SQLException connectException)

    {

        System.out.println(connectException.getMessage());

        System.out.println(connectException.getSQLState());

        System.out.println(connectException.getErrorCode());

        System.exit(1);

    }

}
```

```
private void loadStudent()
{
    try
    {
        rs = statement.executeQuery("SELECT S_ID FROM STUDENT");
        while (rs.next())
        {
            studentIDList.add(rs.getString("S_ID"));
        }
    }
    catch (SQLException e)
    {
        displaySQLErrors(e);
    }
}

public void buildGUI()
{
    studentIDList = new List(10);

    loadStudent();

    add(studentIDList);

    //When a list item is selected populate the text fields
    studentIDList.addItemListener(new ItemListener()
```

```
{  
  
public void itemStateChanged(ItemEvent e)  
  
{  
  
    try  
  
    {  
  
        rs = statement.executeQuery("SELECT * FROM Student");  
  
        while (rs.next())  
  
        {  
  
            if  
(rs.getString("S_ID").equals(studentIDList.getSelectedItem()))  
  
                break;  
  
        }  
  
        if (!rs.isAfterLast())  
  
        {  
  
            sidText.setText(rs.getString("S_ID"));  
  
            fnameText.setText(rs.getString("First_NAME"));  
  
            lnameText.setText(rs.getString("Last_name"));  
  
            unameText.setText(rs.getString("USER_NAME"));  
  
            passwordText.setText(rs.getString("PASSWORD"));  
  
        }  
  
    }  
  
    catch (SQLException selectException)  
  
    {  
  
        displaySQLErrors(selectException);  
  
    }  
  
}
```

```

    }

});

//Handle Update Sailor Button

updateStudentButton = new Button("Modify");

updateStudentButton.addActionListener(new ActionListener()

{

    public void actionPerformed(ActionEvent e)

    {

        try

        {

            Statement statement = connection.createStatement();

            int i = statement.executeUpdate("UPDATE STUDENT

SET First_name='" + fnameText.getText() + "', Last_name='" + lnameText.getText() + "', "

+ "User_name

='"+unameText.getText()+"',"+ "password ='"+passwordText.getText()+"'"+" WHERE S_id ="

+ studentIDList.getSelectedItem());

            errorText.append("\nUpdated " + i + " rows

successfully");

            studentIDList.removeAll();

            loadStudent();

        }

        catch (SQLException insertException)

        {

            displaySQLErrors(insertException);

```

```
        }  
    }  
});  
  
sidText = new TextField(15);  
sidText.setEditable(false);  
fnameText = new TextField(15);  
lnameText = new TextField(15);  
unameText = new TextField(15);  
passwordText=new TextField(15);  
  
errorText = new TextArea(10, 40);  
errorText.setEditable(false);  
  
Panel first = new Panel();  
first.setLayout(new GridLayout(5, 2));  
first.add(new Label("Student ID:"));  
first.add(sidText);  
first.add(new Label("FirstName:"));  
first.add(fnameText);  
first.add(new Label("Lastname:"));  
first.add(lnameText);  
first.add(new Label("Username:"));  
first.add(unameText);
```

```
first.add(new Label("Password:"));

first.add(passwordText);


Panel second = new Panel(new GridLayout(5, 1));

second.add(updateStudentButton);


Panel third = new Panel();

third.add(errorText);


add(first);

add(second);

add(third);


setSize(500, 600);

setLayout(new FlowLayout());

setVisible(true);

}


private void displaySQLErrors(SQLException e)

{

    errorText.append("\nSQLException: " + e.getMessage() + "\n");

    errorText.append("SQLState:  " + e.getSQLState() + "\n");

    errorText.append("VendorError: " + e.getErrorCode() + "\n");
```

```
}

public static void main(String[] args)
{
    UpdateStudent us = new UpdateStudent();

    us.buildGUI();
}
}

4)Main method

import java.awt.*;

import java.awt.event.*;

class OnlineMoocsProvider extends Frame implements ActionListener
{
    String msg = "";

    Label l1,l2;

    CardLayout cardLO;

    //Create Panels for each of the menu items, welcome screen panel and home
screen panel with CardLayout

    InsertProvider provide;

    UpdateProvider upp;
```

DBMS ASSIGNMENT 2

DeleteProvider delp;

InsertCourses inc;

UpdateCourses upc;

DeleteCourses delc;

InsertStudent ins;

DeleteStudent dels;

UpdateStudent us;

Enroll mks;

UpdateEnroll upe;

DeleteEnroll dele;

InsertAssignments ina;

UpdateAssignments upa;

DeleteAssignment dela;

InsertResults inr;

DeleteResults delr;

UpdateResults upr;

Panel home,welcome;

OnlineMoocsProvider()

{

cardLO = new CardLayout();

//Create an empty home panel and set its layout to card layout


```
home = new Panel();
```

```
home.setLayout(cardLO);
```

```
l1 = new Label();
```

```
l2 = new Label();
```

```
l1.setAlignment(Label.CENTER);
```

```
l2.setAlignment(Label.CENTER);
```

```
l1.setText("Welcome to Online MOOC's Provider");
```

```
l2.setText("All @rights are reserved");
```

```
//Create welcome panel and add the label to it
```

```
welcome = new Panel();
```

```
welcome.add(l1);
```

```
welcome.add(l2);
```

```
//create panels for each of our menu items and build them with  
respective components
```

```
provide = new InsertProvider(); provide.buildGUI();
```

```
upp = new UpdateProvider(); upp.buildGUI();
```

```
delp = new DeleteProvider(); delp.buildGUI();
```

```
inc = new InsertCourses();inc.buildGUI();
```

```
upc= new UpdateCourses();upc.buildGUI();
```

```
delc = new DeleteCourses();delc.buildGUI();
```

```
ins = new InsertStudent();ins.buildGUI();
```

```
dels = new DeleteStudent();dels.buildGUI();
```

DBMS ASSIGNMENT 2

```
us= new UpdateStudent();us.buildGUI();

mks= new Enroll();    mks.buildGUI();

upe= new UpdateEnroll();upe.buildGUI();

dele = new DeleteEnroll(); dele.buildGUI();

ina = new InsertAssignments();ina.buildGUI();

upa = new UpdateAssignments();upa.buildGUI();

dela = new DeleteAssignment();dela.buildGUI();

inr = new InsertResults();inr.buildGUI();

delr = new DeleteResults();delr.buildGUI();

upr = new UpdateResults();upr.buildGUI();


//add all the panels to the home panel which has a cardlayout

home.add(welcome, "Welcome");

home.add(provide, "InsertProvider");

home.add(upp, "UpdateProvider");

home.add(delp, "DeleteProvider");

home.add(inc,"InsertCourses");

home.add(upc,"UpdateCourses");

home.add(delc,"DeleteCourses");

home.add(ins,"InsertStudent");

home.add(dels,"DeleteStudent");

home.add(us,"UpdateStudent");

home.add(mks,"Enroll");
```

DBMS ASSIGNMENT 2

```
home.add(upe,"UpdateEnroll");  
home.add(dele,"DeleteEnroll");  
home.add(ina,"InsertAssignments");  
home.add(upa,"UpdateAssignments");  
home.add(dela,"DeleteAssignment");  
home.add(inr,"InsertResults");  
home.add(delr,"DeleteResults");  
home.add(upr,"UpdateResults");
```

```
//home.add(upb, "UpdateBoat");  
//home.add(mks, "MakeReserve");
```

```
// add home panel to main frame  
add(home);
```

```
// create menu bar and add it to frame  
MenuBar mbar = new MenuBar();  
setMenuBar(mbar);
```

```
// create the menu items and add it to Menu
```

DBMS ASSIGNMENT 2

```
Menu provider = new Menu("OnlineMOOC'sProvider");  
  
MenuItem item1, item2, item3;  
  
provider.add(item1 = new MenuItem("Insert Provider"));  
provider.add(item2 = new MenuItem("View Provider"));  
provider.add(item3 = new MenuItem("Delete Provider"));  
  
mbar.add(provider);
```

```
Menu courses = new Menu("Courses");  
  
MenuItem item4, item5, item6;  
  
courses.add(item4 = new MenuItem("Insert Courses"));  
courses.add(item5 = new MenuItem("View Courses"));  
courses.add(item6 = new MenuItem("Delete Courses"));  
  
mbar.add(courses);
```

```
Menu student = new Menu("Student");  
  
MenuItem item7, item8, item9;  
  
student.add(item7 = new MenuItem("Insert Student"));  
student.add(item8 = new MenuItem("View Student"));  
student.add(item9 = new MenuItem("Delete Student"));  
  
mbar.add(student);
```

```
Menu enroll= new Menu("Enroll");  
  
MenuItem item10, item11, item12;  
  
enroll.add(item10 = new MenuItem("Insert Enroll"));
```

DBMS ASSIGNMENT 2

```
enroll.add(item11= new MenuItem("View Enroll"));

enroll.add(item12 = new MenuItem("Delete Enroll"));

mbar.add(enroll);


Menu assignments= new Menu("Assignments");

MenuItem item13, item14, item15;

assignments.add(item13 = new MenuItem("Insert Assignments"));

assignments.add(item14= new MenuItem("View Assignments"));

assignments.add(item15 = new MenuItem("Delete Assignment"));

mbar.add(assignments);


Menu results= new Menu("Results");

MenuItem item16, item17, item18;

results.add(item16 = new MenuItem("Insert Results"));

results.add(item17= new MenuItem("View Results"));

results.add(item18 = new MenuItem("Delete Results"));

mbar.add(results);


// register listeners

item1.addActionListener(this);

item2.addActionListener(this);

item3.addActionListener(this);

item4.addActionListener(this);

item5.addActionListener(this);
```

DBMS ASSIGNMENT 2

```
item6.addActionListener(this);  
item7.addActionListener(this);  
item8.addActionListener(this);  
item9.addActionListener(this);  
item10.addActionListener(this);  
item11.addActionListener(this);  
item12.addActionListener(this);  
item13.addActionListener(this);  
item14.addActionListener(this);  
item15.addActionListener(this);  
item16.addActionListener(this);  
item17.addActionListener(this);  
item18.addActionListener(this);
```

```
// Anonymous inner class which extends WindowAdaptor to handle  
the Window event: windowClosing
```

```
addWindowListener(new WindowAdapter(){  
    public void windowClosing(WindowEvent we)  
    {  
        System.exit(0);  
    }  
});  
  
//Frame properties  
  
setTitle("Online MOOC's Provider");
```

DBMS ASSIGNMENT 2

```
        Color clr = new Color(50, 150, 100);

        setBackground(clr);

        setFont(new Font("Monaco", Font.BOLD, 20));

        setSize(900, 1000);

        setVisible(true);

    }

    public void actionPerformed(ActionEvent ae)
    {

        String arg = ae.getActionCommand();

        if(arg.equals("Insert Provider"))
        {

            cardLO.show(home, "InsertProvider");

        }

        else if(arg.equals("View Provider"))
        {

            cardLO.show(home, "UpdateProvider");

        }

        else if(arg.equals("Delete Provider"))
        {

            cardLO.show(home, "DeleteProvider");
```

```
}  
  
else if(arg.equals("Insert Courses"))  
{  
  
    cardLO.show(home, "InsertCourses");  
  
}  
  
else if(arg.equals("Delete Courses"))  
{  
  
    cardLO.show(home, "DeleteCourses");  
  
}  
  
else if(arg.equals("View Courses"))  
{  
  
    cardLO.show(home, "UpdateCourses");  
  
}  
  
else if(arg.equals("Insert Student"))  
{  
  
    cardLO.show(home, "InsertStudent");  
  
}  
  
else if(arg.equals("Delete Student"))  
{  
  
    cardLO.show(home, "DeleteStudent");  
  
}
```



```
else if(arg.equals("View Student"))
{
    cardLO.show(home, "UpdateStudent");
}
else if(arg.equals("Insert Enroll"))
{
    cardLO.show(home, "Enroll");
}
else if(arg.equals("View Enroll"))
{
    cardLO.show(home, "UpdateEnroll");
}
else if(arg.equals("Delete Enroll"))
{
    cardLO.show(home, "DeleteEnroll");
}
else if(arg.equals("Insert Assignments"))
{
    cardLO.show(home, "InsertAssignments");
}
else if(arg.equals("View Assignments"))
{
    cardLO.show(home, "UpdateAssignments");
}
```

```
        else if(arg.equals("Insert Results"))
        {
            cardLO.show(home, "InsertResults");
        }
        else if(arg.equals("View Results"))
        {
            cardLO.show(home, "UpdateResults");
        }
        else if(arg.equals("Delete Results"))
        {
            cardLO.show(home, "DeleteResults");
        }
    }

    public static void main(String ... args)
    {
        new OnlineMoocsProvider();
    }
}
```

Connectivity with the Database:

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database and is oriented towards relational databases.

Block of code for JAVA- SQL connectivity with JDBC:

```
public void connectToDB()
{
    try
    {
        connection=DriverManager.getConnection("jdbc:oracle:thin:@localhost:
1521:orcl","hemanth","oracle");

        statement=connection.createStatement();
    }
    catch(SQLException connectException)
    {
        System.out.println(connectException.getMessage());
        System.out.println(connectException.getSQLState());
        System.out.println(connectException.getErrorCode());
        System.exit(1);
    }
}
```

GITHUB LINK:

<https://github.com/sherlahemanth001/DBMS-project.git>

Folder Structure:

This project contains a folder named src in which it has 19 .java files which are from 6 different tables which include insert , delete, update and a main file. By which we can navigate easily to reach the java code and we can make changes easily.

Name	Date modified	Type	Size
.settings	22-04-2020 19:19	File folder	
bin	22-04-2020 19:19	File folder	
src	22-04-2020 19:19	File folder	
.classpath	20-03-2020 13:59	CLASSPATH File	1 KB
.project	14-03-2020 23:04	PROJECT File	1 KB

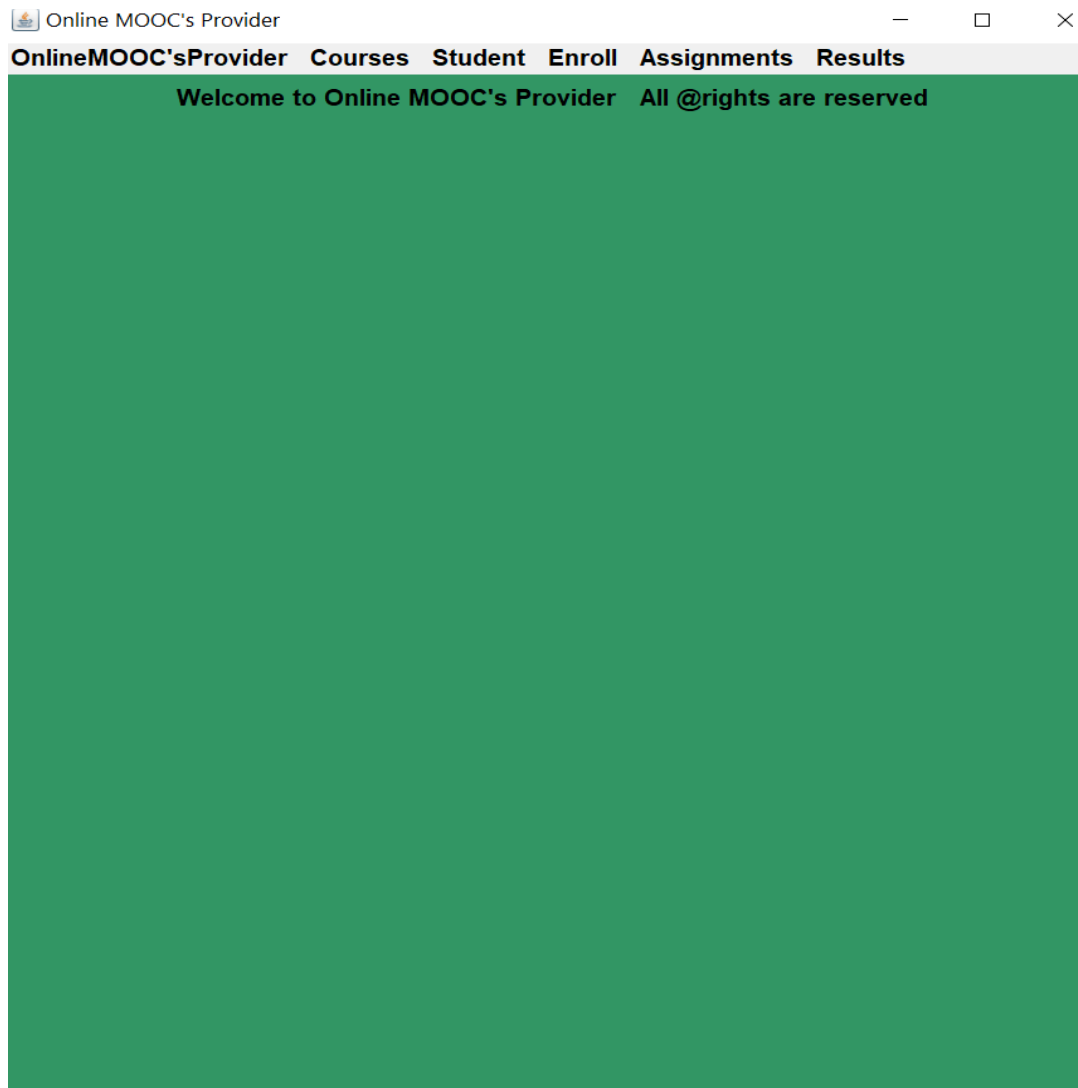
Name	Date modified	Type	Size
DeleteAssignment	19-04-2020 16:04	JAVA File	5 KB
DeleteCourses	19-04-2020 16:05	JAVA File	5 KB
DeleteEnroll	19-04-2020 16:05	JAVA File	4 KB
DeleteProvider	19-04-2020 16:05	JAVA File	5 KB
DeleteResults	19-04-2020 16:05	JAVA File	5 KB
DeleteStudent	19-04-2020 16:06	JAVA File	5 KB
Enroll	19-04-2020 16:03	JAVA File	4 KB
InsertAssignments	19-04-2020 16:03	JAVA File	4 KB
InsertCourses	19-04-2020 16:03	JAVA File	4 KB
InsertProvider	19-04-2020 16:01	JAVA File	4 KB
InsertResults	19-04-2020 16:03	JAVA File	4 KB
InsertStudent	19-04-2020 16:03	JAVA File	4 KB
OnlineMoocsProvider	19-04-2020 18:41	JAVA File	8 KB
UpdateAssignments	19-04-2020 16:08	JAVA File	5 KB
UpdateCourses	19-04-2020 16:08	JAVA File	5 KB
UpdateEnroll	19-04-2020 16:11	JAVA File	4 KB
UpdateProvider	19-04-2020 16:15	JAVA File	5 KB
UpdateResults	19-04-2020 16:15	JAVA File	5 KB
UpdateStudent	19-04-2020 16:18	JAVA File	5 KB

TESTING

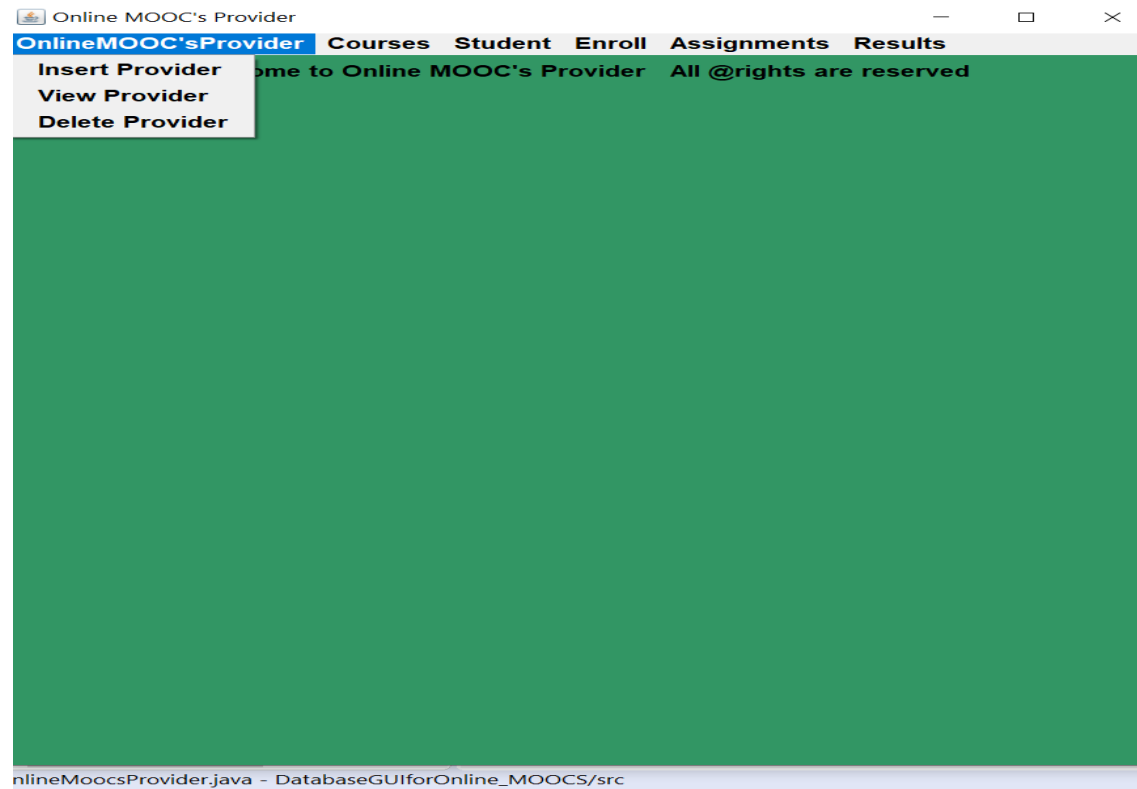
The program runs for execution of three basic operations of insertion, update and delete on 5 different table. Along with this, it also has a output column which gives the information about how many rows have been edited. Errors, syntactical or exceptional will be shown if occurred.

The code written for building GUI and connecting with database ensures that the values entered by the user are of correct data types. It prompts an error message in the text message box.

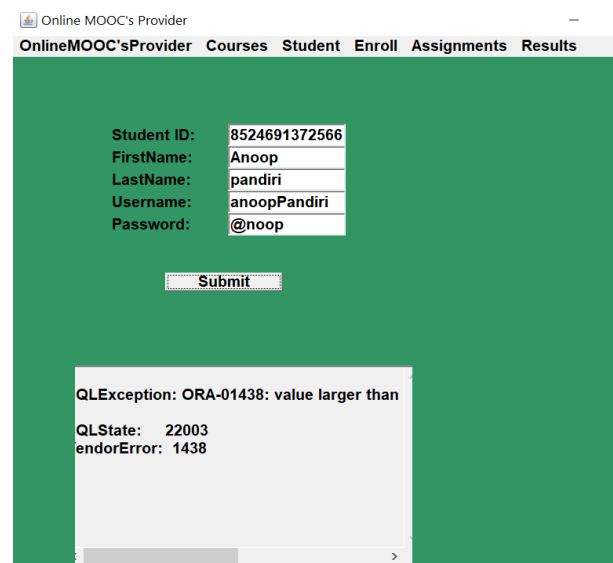
HOME PAGE:



DBMS ASSIGNMENT 2

**INSERT STUDENT:****ERROR:**

If user enters a invalid content it will through an error .



PROPER ENTRY:

Online MOOC's Provider

OnlineMOOC'sProvider Courses Student Enroll Assignments Results

Student ID: 106
 FirstName: Pandiri
 LastName: Anoop
 Username: anoop@001
 Password: @noop

Submit

Inserted 1 rows successfully

```
SQL> select * from student;
```

S_ID	FIRST_NAME	LAST_NAME	USER_NAME	PASSWORD
105	Sujitha	Tadi	SujithaT	@sujitha
83	mohammad	razzaq	Mohammad	r@zz@q
94	mallik	reddy	saimallik	s@im@llik
106	Pandiri	Anoop	anoop@001	@noop
61	Abhiraj	dusari	dusariabhi	@bhir@j
72	Hemanth	Sherla	Hemanth	Hem@nth

```
6 rows selected.
```

```
SQL> _
```

UPDATE STUDENT:

ERROR:

If user enters a invalid content it will through an error .

Online MOOC's Provider

OnlineMOOC'sProvider Courses Student Enroll Assignments Results

61	Student ID:	72	Modifi
72	FirstName:	Hemanth	
83	LastName:	Sherla	
94	Username:	Hemanth	
105	Password:	@nthKumarlaeopimj	

ENT"."PASSWORD" (actual: 20, maximum: 10)

Online MOOC's Provider

OnlineMOOC'sProvider Courses Student Enroll Assignments Results

61	Student ID:	72	Modify
72	FirstName:	Hemanth	
83	LastName:	Sherla	
94	Username:	Hemanth	
105	Password:	@nthKumarlaeopimj	

Updated 0 rows successfully
SQLException: ORA-12899: value too large fo
SQLState: 72000
VendorError: 12899

PROPER ENTRY:

Online MOOC's Provider

OnlineMOOC'sProvider Courses Student Enroll Assignments Results

61
72
83
94
105
106

Student ID: 106

FirstName: ANOOP

LastName: PANDIRI

Username: anoop@001

Password: @noop

Updated 1 rows successfully

```
SQL> SELECT * FROM STUDENT;
```

S_ID	FIRST_NAME	LAST_NAME	USER_NAME	PASSWORD
105	Sujitha	Tadi	SujithaT	@sujitha
83	mohammad	razzaq	Mohammad	r@zz@g
94	mallik	reddy	saimallik	s@im@llik
106	ANOOP	PANDIRI	anoop@001	@noop
61	Abhiraj	dusari	dusariabhi	@bhir@j
72	Hemanth	Sherla	Hemanth	Hem@nth

```
6 rows selected.
```

DELETE STUDENT:

Online MOOC's Provider

OnlineMOOC'sProvider Courses Student Enroll Assignments Results

106	Student ID:	106	Delete
105	FirstName:	ANOOP	
83	LastName:	PANDIRI	
94	Username:	anoop@001	
61	password:	@noop	
72			

Deleted 1 rows successfully

Online MOOC's Provider

OnlineMOOC'sProvider Courses Student Enroll Assignments Results

105	Student ID:		Delete
83	FirstName:		
94	LastName:		
61	Username:		
72	password:		

Deleted 1 rows successfully

```
SQL> SELECT * FROM STUDENT;
```

S_ID	FIRST_NAME	LAST_NAME	USER_NAME	PASSWORD
105	Sujitha	Tadi	SujithaT	@sujitha
83	mohammad	razzaq	Mohammad	r@zz@q
94	mallik	reddy	saimallik	s@im@llik
106	ANOOPI	PANDIRI	anoop@001	@noop
61	Abhiraj	usari	abhiraj	@bhiraj
72	Hemanth	Sherla	Hemanth	Hem@nth

```
6 rows selected.

SQL> SELECT * FROM STUDENT;
```

S_ID	FIRST_NAME	LAST_NAME	USER_NAME	PASSWORD
105	Sujitha	Tadi	SujithaT	@sujitha
83	mohammad	razzaq	Mohammad	r@zz@q
94	mallik	reddy	saimallik	s@im@llik
61	Abhiraj	usari	abhiraj	@bhiraj
72	Hemanth	Sherla	Hemanth	Hem@nth

```
SQL>
```

RESULTS

The DML commands, Insert, update and delete for one of the tables in given below:

For student table: (in java, as per the application)

```
Insert: "INSERT INTO student VALUES(" + sidText.getText() + ", " + "'" +
+ fnameText.getText() + "', " + "'" + lnameText.getText() +
+ "', " + "'" + unameText.getText() + "', "
+ "'" + passwordText.getText() + "'" + ")";
```

```
Update: "UPDATE STUDENT SET First_name=" + fnameText.getText() +
+ "', Last_name=" + lnameText.getText() + "', " + "User_name
+ "'" + unameText.getText() + "'" + "password
+ "'" + passwordText.getText() + "'" + " WHERE S_id =" +
+ studentIDList.getSelectedItem());
```

```
Delete: "DELETE FROM student WHERE S_ID = " +
+ studentIDList.getSelectedItem());
```

- 1.Connection with database is established.
- 2.The values given for tables in the GUI components by the user are saved in the database.

REFERENCES

- 1.<http://sociallearningcommunity.com/10-of-the-best-mooc-providers/>
- 2.https://en.wikipedia.org/wiki/List_of_MOOC_providers
- 3.<https://github.com/sherlahemanth001/DBMS-project.git>(assignment 1)