

**Konstantinos Kousounnis**  
**Java-Application-PrivateSchool-Data-Management-System-Mysql**  
**Documentation**

## **Project Brief**

Following PART A you need to implement the below functionality:

1. Design the ERD of your system and verify it through an online tool such as <https://sqldbm.com/> (it requires a free account) [15 marks]
2. Identify any other tables you need based on your implementation and construct them [15 marks]
3. Make the schema of a database that can keep data for the main entities of the assignment and name the tables as: Students, Trainers, Assignments, Courses [15 marks]
4. Populate the tables of the database with enough data [10 marks]
5. You need to produce sql queries that output the following [19 marks in total]:
  - A list of all the students [2 marks]
  - A list of all the trainers [2 marks]
  - A list of all the assignments [2 marks]
  - A list of all the courses [2 marks]
  - All the students per course [2 marks]
  - All the trainers per course [2 marks]
  - All the assignments per course [2 marks]
  - All the assignments per course per student [2 marks]
  - A list of students that belong to more than one courses [3 marks]
6. You also need to produce a small project that [26 marks]
  - a. makes a connection to the database and executes the above sql queries [9 marks]
  - b. makes a connection to the database and inserts input data from the keyboard to the following tables,
    - i. students [2 marks]
    - ii. trainers [2 marks]
    - iii. assignments [2 marks]
    - iv. courses [2 marks]
    - v. students per course [3 marks]
    - vi. trainers per course [3 marks]
    - vii. assignments per student per course [3 marks]

## Project - Documentation

Github:

<https://github.com/kkousounnis/Java-Application-PrivateSchool-Data-Management-System-Mysql.git>

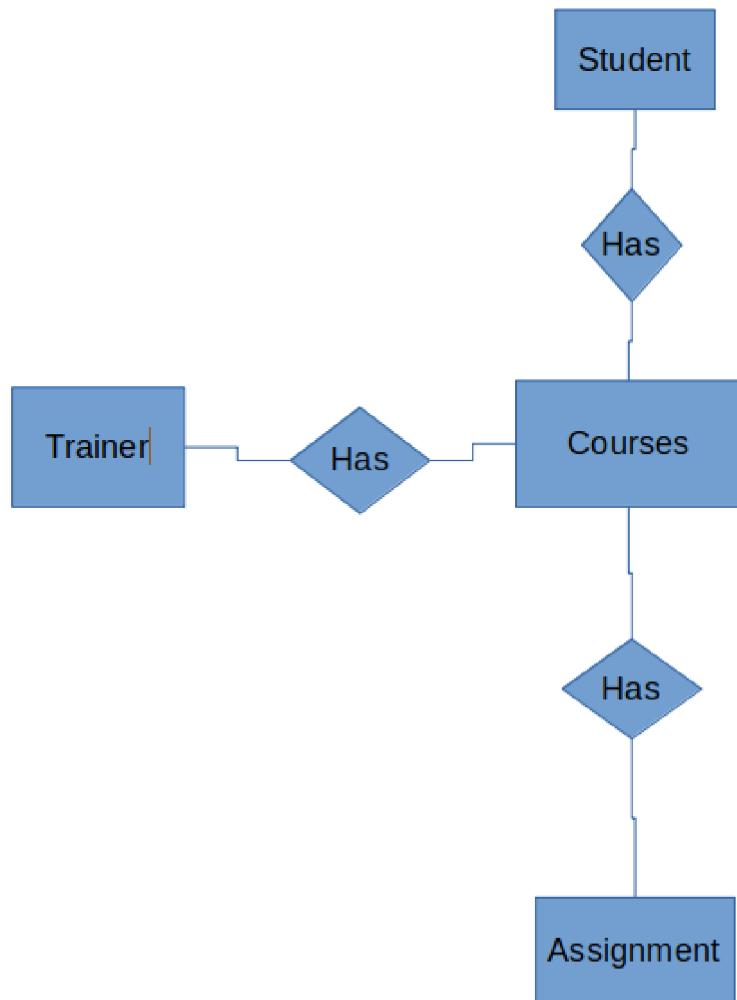
1. Design the ERD of your system and verify it through an online tool such as <https://sqldbm.com/> (it requires a free account) [15 marks]
2. Identify any other tables you need based on your implementation and construct them [15 marks]
3. Make the schema of a database that can keep data for the main entities of the assignment and name the tables as: Students, Trainers, Assignments, Courses [15 marks]
4. Populate the tables of the database with enough data [10 marks]

First Thought Design:

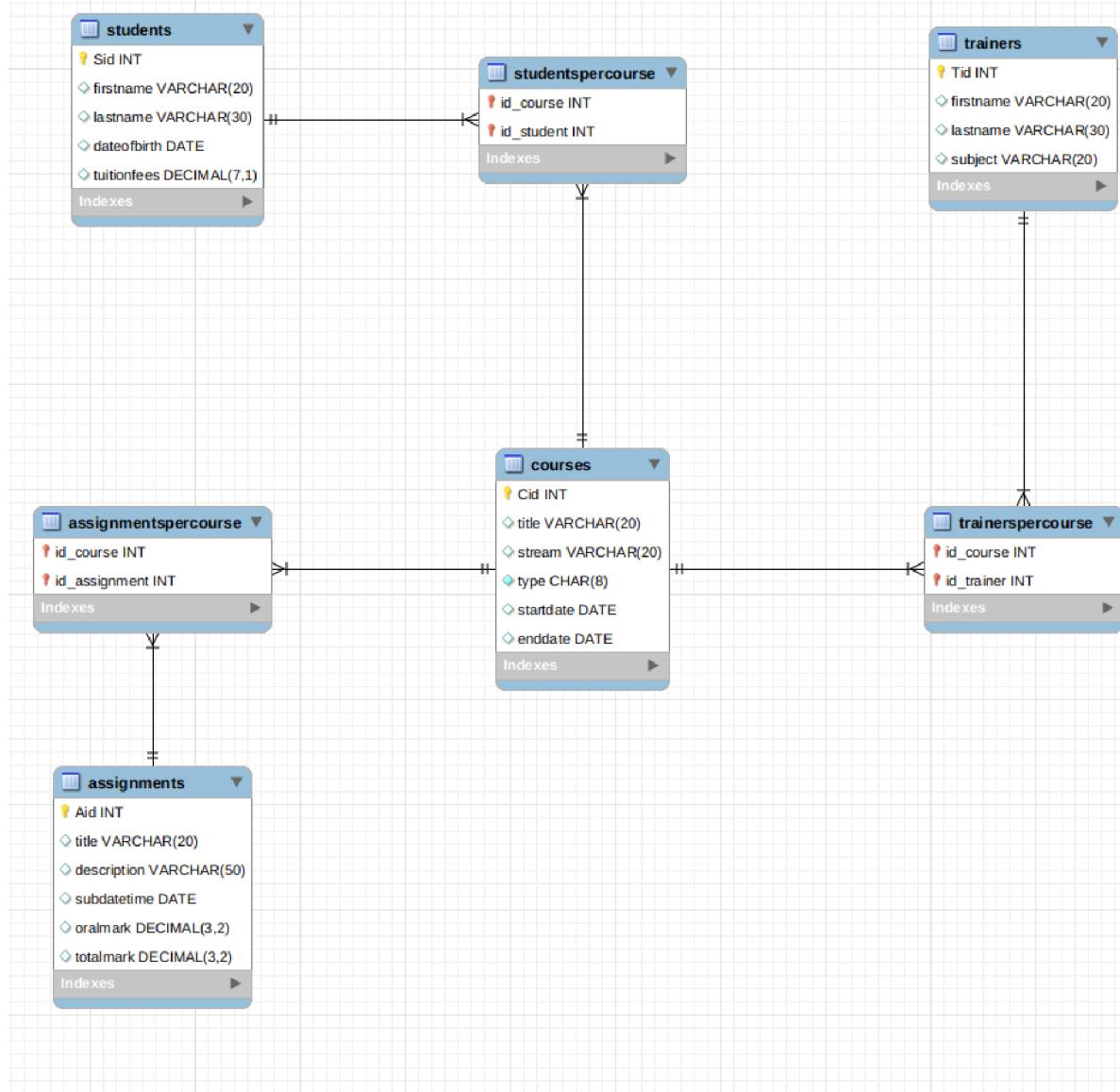
Student Has Courses.

Trainer Has Courses.

Assignment Has Courses.



## Complete project



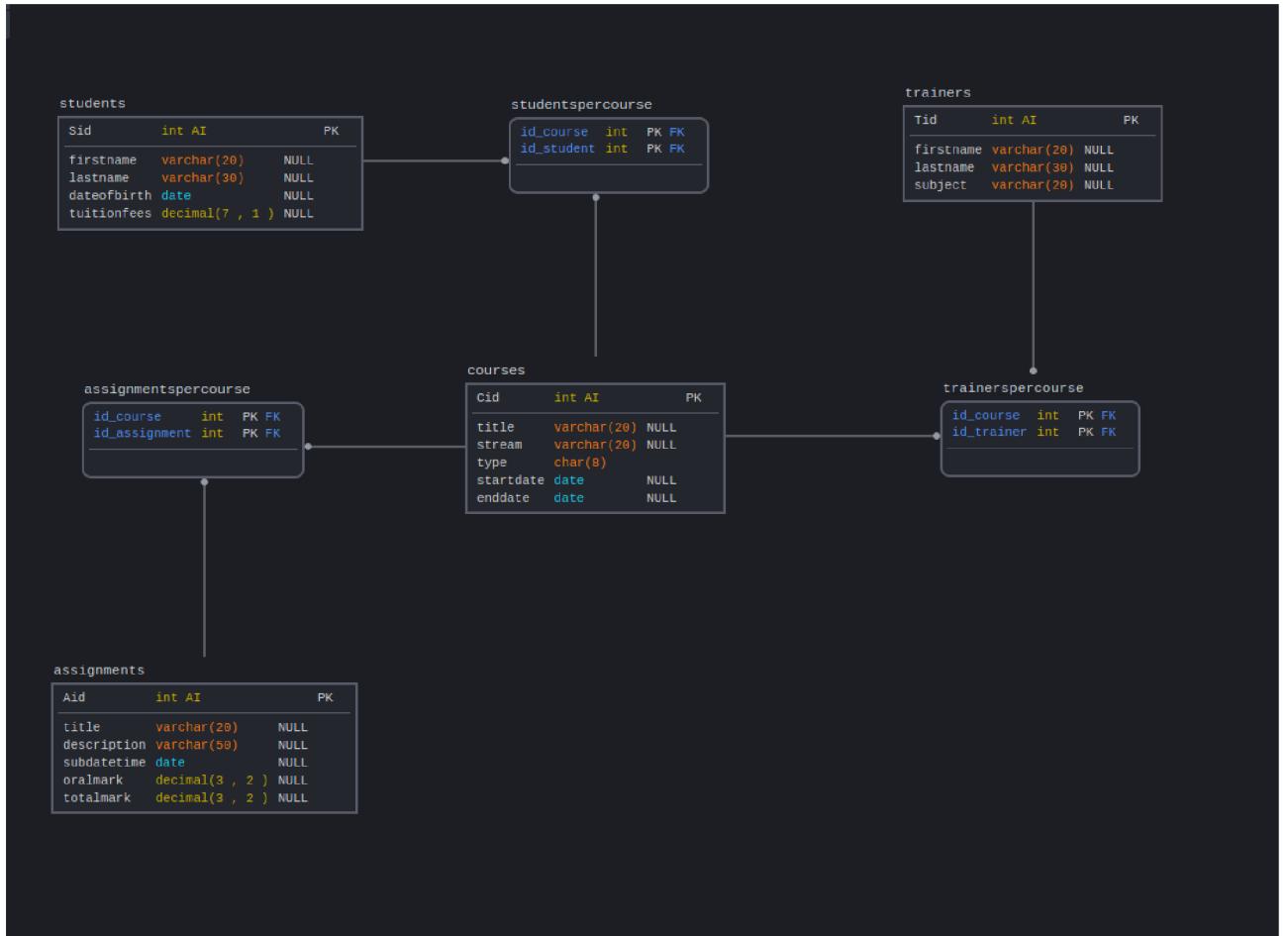
The extra tables that we need are **students\_per\_course**, **trainers\_per\_course**, **assignments\_per\_course**.

We need these tables in order to achieve the rules of normalization in our database.

As a result in our database we have the following rules.

1. We don't have loss of information.
2. We don't have information redundancy and our tables have been split exactly in the point where we don't have data repetition and all data are unique.
3. Maintaining dependencies all tables are interconnected.

Erd Verification through an online tool.



Erd diagramm succesfully created.

```

Local instance 3306 | MySQL Model | EER Diagram
File Edit View Query Database Server Tools Scripting Help
Administration Schemas
Filter objects
eshop
eshop1
privateschool
Tables
Views
Stored Procedures
Functions
sales
sys

Query 2 × privateschooldb ×
Limit to 1000 rows
1 • DROP DATABASE IF EXISTS `privateschool`;
2 • CREATE DATABASE `privateschool`;
3 • USE `privateschool` ;
4
5 -- Create Table `courses`
6
7 • CREATE TABLE IF NOT EXISTS `courses` (
8     `Cid` INT NOT NULL AUTO_INCREMENT,
9     `title` VARCHAR(20) NULL DEFAULT NULL,
10    `stream` VARCHAR(20) NULL DEFAULT NULL,
11    `type` CHAR(8) NOT NULL,
12    `startdate` DATE NULL DEFAULT NULL,
13    `enddate` DATE NULL DEFAULT NULL,
14    PRIMARY KEY (`Cid`)
15 );
16
17 -- Create Table `students`
18
19 • CREATE TABLE IF NOT EXISTS `students` (
20     `Sid` INT NOT NULL AUTO_INCREMENT,
21     `firstname` VARCHAR(20) NULL DEFAULT NULL,
22     `lastname` VARCHAR(30) NULL DEFAULT NULL,
23     `dateofbirth` DATE NULL DEFAULT NULL,
24     `tuitionfees` DECIMAL(7 , 1 ) UNSIGNED NULL DEFAULT NULL,
25     PRIMARY KEY (`Sid`)
26 );
27
28 -- Create Table `trainers`

```

Privateschooldb.sql is the file which has the source code of the database and populates the database with enough data.

```

157      (7,10);
158
159      -- Insert values to trainerspercourse
160
161 •  INSERT INTO `trainerspercourse`(`id_course`,`id_trainer`) VALUES(4,7),
162      (3,7),
163      (2,1),
164      (1,3),
165      (7,4),
166      (8,2),
167      (5,8),
168      (6,6);
169
170      -- Insert values to assignmentspercourse
171 •  INSERT INTO `assignmentspercourse`(`id_course`,`id_assignment`)
172      VALUES(1,1),(1,2),(1,3),(1,4),
173      (2,1),(2,2),(2,3),(2,4),
174      (3,1),(3,2),(3,3),(3,4),
175      (4,1),(4,2),(4,3),(4,4),
176      (5,5),(5,6),(5,7),(5,8),
177      (6,5),(6,6),(6,7),(6,8),
178      (7,5),(7,6),(7,7),(7,8),
179      (8,5),(8,6),(8,7),(8,8);
180

```

5. You need to produce sql queries that output the following [19 marks in total]:

- A list of all the students [2 marks]
- A list of all the trainers [2 marks]
- A list of all the assignments [2 marks]
- A list of all the courses [2 marks]
- All the students per course [2 marks]
- All the trainers per course [2 marks]
- All the assignments per course [2 marks]
- All the assignments per course per student [2 marks]
- A list of students that belong to more than one courses [3 marks]

A list of all the students [2 marks]

The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below the toolbar is a query editor window titled "1-ListStudents". The SQL code in the editor is:

```
1 | - A list of all the students
2
3 • USE `privateschool`;
4
5 • SELECT
6     `firstname` AS `First_Name`,
7     `lastname` AS `Last_Name`,
8     `dateofbirth` AS `Date_Of_Birth`,
9     `tuitionfees` AS `TuitionFees`
10    FROM
11        `students`
```

Below the query editor is a "Result Grid" window. It has a header row with columns: #, First\_Name, Last\_Name, Date\_Of\_Birth, TuitionFees. The data grid contains 10 rows of student information:

#	First_Name	Last_Name	Date_Of_Birth	TuitionFees
1	Klemens	Habgood	1994-05-12	5000.0
2	Rina	Vasichenko	1994-06-06	5000.0
3	Tabina	Steinhammer	1992-02-08	5000.0
4	Johanna	Stanislaw	1994-03-07	5000.0
5	Leila	Pierce	1993-04-10	5000.0
6	Marten	Kowalski	1995-05-09	5000.0
7	Elladine	Shorrock	1996-02-08	5000.0
8	Sanderson	Tankard	1992-07-04	5000.0
9	Lindsay	Matches	1995-10-02	5000.0
10	Teador	Middup	1995-07-11	5000.0

At the bottom of the interface is an "Action Output" window titled "students 1". It displays the execution log:

#	Time	Action	Message	Duration / Fetch
1	15:42:09	USE `privateschool`	0 row(s) affected	0,00039 sec
2	15:42:09	SELECT `firstname` AS `First_Name`, `lastname` AS ...	10 row(s) returned	0,00057 sec / 0,000...

A list of all the trainers [2 marks]

The screenshot shows the MySQL Workbench interface. At the top is a toolbar with various icons. Below it is a query editor window titled "2-ListTrainers". The query code is:

```
1 -- A list of all the trainers
2
3 • USE `privateschool`;
4
5 • SELECT
6     `firstname` AS `First_Name`,
7     `lastname` AS `Last_Name`,
8     `subject` AS `Subject`
9 FROM
10    `trainers`
```

Below the query editor is a "Result Grid" window. It has a header row with columns: #, First\_Name, Last\_Name, Subject. The data rows are:

#	First_Name	Last_Name	Subject
1	Corissa	Abrahim	Backend
2	Judith	Howatt	Database
3	Merry	Kenewell	Backend
4	Isahella	Gockelen	Database
5	Judith	Howatt	Backend
6	Merry	Kenewell	Database
7	Denys	Lambden	Backend
8	Neal	Hairsnape	Database

At the bottom is an "Action Output" window titled "trainers 2". It contains a table with two rows:

#	Time	Action	Message
1	15:43:41	USE `privateschool`	0 row(s) affected
2	15:43:41	SELECT `firstname` AS `First_Name`, `lastname` AS ...	8 row(s) returned

A list of all the assignments [2 marks]

The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below the toolbar, the title bar says "3-ListAssignments". The main area contains a SQL query:

```
1 -- A list of all the assignments
2
3 • USE `privateschool`;
4
5 • SELECT
6     `title` AS `Title`,
7     `description` AS `Description`,
8     `subdatetime` AS `SubDateTime`,
9     `oralmark` AS `OralMark`,
10    `totalmark` AS `TotalMark`
11 FROM
12     `assignments`
```

Below the query, there is a "Result Grid" section with the following data:

#	Title	Description	SubDateTime	OralMark	TotalMark
1	Assignment1	Create a hello world app	2021-01-08	NULL	NULL
2	Assignment2	Create an ArrayList of animals	2021-01-16	NULL	NULL
3	Assignment3	Create an Sorting Algorithm	2021-01-24	NULL	NULL
4	Assignment4	Create an java consola Appli...	2021-02-02	NULL	NULL
5	Assignment5	Create a Select From qeury	2021-01-08	NULL	NULL
6	Assignment6	Create a quyrt with aggregati...	2021-01-16	NULL	NULL
7	Assignment7	Create a nested query	2021-01-24	NULL	NULL
8	Assignment8	Create an inner join query	2021-02-02	NULL	NULL

At the bottom, there is another tab titled "assignments 2" which shows the "Action Output" with the following data:

#	Time	Action	Message
1	15:44:57	USE `privateschool`	0 row(s) affected
2	15:44:57	SELECT `title` AS `Title`, `description` AS `Descripti...	8 row(s) returned

A list of all the courses [2 marks]

The screenshot shows the MySQL Workbench interface. In the top-left window, titled '4-ListCourses', a SQL query is written:

```
1 -- A list of all the courses
2
3 • USE `privateschool`;
4
5 • SELECT
6     `title` AS `Title`,
7     `stream` AS `Stream`,
8     `type` AS `Type`,
9     `startdate` AS `StartDate`,
10    `enddate` AS `EndDate`
11 FROM
12     `courses`
```

In the bottom-left window, titled 'courses 1', the 'Action Output' tab is selected, showing the execution history:

#	Time	Action	Message
1	15:47:09	USE `privateschool`	0 row(s) affect
2	15:47:09	SELECT `title` AS `Title`, `stream` AS `Stream`, `type` AS `Type`, `startdate` AS `StartDate`, `enddate` AS `EndDate` FROM `courses`	8 row(s) return

The main area displays the 'Result Grid' with the following data:

#	Title	Stream	Type	StartDate	EndDate
1	course1	Javascript	Fulltime	2021-01-01	2021-03-31
2	course1	Javascript	Parttime	2021-01-01	2021-05-31
3	course2	Go	Fulltime	2021-01-01	2021-03-31
4	course2	Go	Parttime	2021-01-01	2021-05-31
5	course3	Postgresql	Fulltime	2021-01-01	2021-03-31
6	course3	Postgresql	Parttime	2021-01-01	2021-05-31
7	course4	Mysql	Fulltime	2021-01-01	2021-03-31
8	course4	Mysql	Parttime	2021-01-01	2021-05-31

All the students per course [2 marks]

The screenshot shows the MySQL Workbench interface with a query editor and a result grid.

**Query Editor:**

```
5-StudentsPerCourse
1 -- All the students per course
2
3 • use `privateschool`;
4
5 • SELECT
6     `courses`.`cid` AS `Course_ID`,
7     `courses`.`title` AS `Title`,
8     `courses`.`stream` AS `Sream`,
9     `courses`.`type` AS `Type`,
10    `students`.`firstname` AS `Students_First_Name`,
11    `students`.`lastname` AS `Students_Last_Name`
12 FROM
13     `studentspercourse`
```

**Result Grid:**

#	Course_ID	Title	Sream	Type	Students_First_Name	Students_Last_Na
1	1	course1	Javascript	Fulltime	Klemens	Habgood
2	1	course1	Javascript	Fulltime	Rina	Vasilchenko
3	2	course1	Javascript	Parttime	Klemens	Habgood
4	2	course1	Javascript	Parttime	Tabina	Steinhammer
5	3	course2	Go	Fulltime	Klemens	Habgood
6	3	course2	Go	Fulltime	Rina	Vasilchenko
7	3	course2	Go	Fulltime	Elladine	Shorrock
8	4	course2	Go	Parttime	Tabina	Steinhammer
9	4	course2	Go	Parttime	Sanderson	Tankard
10	5	course3	Postgresql	Fulltime	Johanna	Stanislaw
11	5	course3	Postgresql	Fulltime	Lindsay	Matches

**Action Output:**

#	Time	Action	Me
1	15:58:31	use `privateschool`	0 ro
2	15:58:31	SELECT `courses`.`cid` AS `Course_ID`, `courses`.`ti...`	16 r

All the trainers per course [2 marks]

6-TrainersPerCourse

```
1 -- All the trainers per course
2
3 • use `privateschool`;
4
5 • SELECT
6     `courses`.`Cid` AS `Course_ID`,
7     `courses`.`title` AS `Title`,
8     `courses`.`stream` AS `Stream`,
9     `courses`.`type` AS `Type`,
10    `trainers`.`firstname` AS `Trainers_First_Name`,
11    `trainers`.`lastname` AS `Trainers_Last_Name`,
12    `trainers`.`subject` AS `Subject`
13 FROM
```

Result Grid Filter Rows: Export: Wrap Cell Content:

#	Course_ID	Title	Stream	Type	Trainers_First_Name	Trainers_Last_Name	Subject
1	1	course1	Javascript	Fulltime	Merry	Kenewell	Backend
2	2	course1	Javascript	Parttime	Corissa	Abrahim	Backend
3	3	course2	Go	Fulltime	Denys	Lambden	Backend
4	4	course2	Go	Parttime	Denys	Lambden	Backend
5	5	course3	Postgresql	Fulltime	Neal	Hairsnape	Database
6	6	course3	Postgresql	Parttime	Merry	Kenewell	Database
7	7	course4	Mysql	Fulltime	Isahella	Gockelen	Database
8	8	course4	Mysql	Parttime	Judith	Howatt	Database

Result 2

Action Output

#	Time	Action	Message
1	16:00:11	use `privateschool`	0 row(s) affected
2	16:00:11	SELECT `courses`.`Cid` AS `Course_ID`, `courses`.`ti...`	8 row(s) returned

All the assignments per course [2 marks]

The screenshot shows the MySQL Workbench interface with a query editor and a results grid.

**Query Editor:**

```
7-AssignmentsPerCourse
1 -- All the Assignments per course
2
3 • use `privateschool`;
4
5 • SELECT
6     `courses`.`Cid` AS `Course_ID`,
7     `courses`.`title` AS `Title`,
8     `courses`.`stream` AS `Sream`,
9     `courses`.`type` AS `Type`,
10    `assignments`.`title` AS `Title`,
11    `assignments`.`description` AS `Description`,
12    `assignments`.`subdatetime` AS `Subdatetime`
13 FROM
```

**Result Grid:**

#	Course_ID	Title	Sream	Type	Title	Description	Subdatetime
1	1	course1	Javascript	Fulltime	Assignment1	Create a hello world app	2021-01-08
2	1	course1	Javascript	Fulltime	Assignment2	Create an Array List of animals	2021-01-16
3	1	course1	Javascript	Fulltime	Assignment4	Create an java consola Appli...	2021-02-02
4	1	course1	Javascript	Fulltime	Assignment3	Create an Sorting Algorithm	2021-01-24
5	2	course1	Javascript	Parttime	Assignment1	Create a hello world app	2021-01-08
6	2	course1	Javascript	Parttime	Assignment2	Create an Array List of animals	2021-01-16
7	2	course1	Javascript	Parttime	Assignment4	Create an java consola Appli...	2021-02-02
8	2	course1	Javascript	Parttime	Assignment3	Create an Sorting Algorithm	2021-01-24
9	3	course2	Go	Fulltime	Assignment3	Create an Sorting Algorithm	2021-01-24
10	3	course2	Go	Fulltime	Assignment4	Create an java consola Appli...	2021-02-02
11	3	course2	Go	Fulltime	Assignment2	Create an Array List of animals	2021-01-16

**Action Output:**

#	Time	Action	Message
1	16:01:10	use `privateschool`	0 row(s) affected
2	16:01:10	SELECT `courses`.`Cid` AS `Course_ID`, `courses`.`ti...	32 row(s) returned

## All the assignments per course per student [2 marks]

The screenshot shows the MySQL Workbench interface with a query editor and a results grid.

```

8-AssignmentsPerCoursePerStudent
Limit to 1000 rows
1 -- All the assignments per course per student
2
3 • use `privateschool`;
4
5 • SELECT
6     `assignmentspercourse`.`id_assignment` AS `Assignment_ID`,
7     `assignments`.`title` AS `Title`,
8     `assignments`.`description` AS `Description`,
9     `assignments`.`subdatetime` AS `Subdatetime`,
10    `courses`.`Cid` AS `Courses_Id`,
11    `courses`.* AS `Courses`

```

**Result Grid:**

#	Assignment_ID	Title	Description	Subdatetime	Courses_Id	Title	Stream	Type	Student_ID	First_Name	Last_Name	Date_Of_Birth	TuitionFees
1	1	Assignment1	Create a hello world app	2021-01-08	1	course1	Javascript	Fulltime	1	Klemens	Habgood	1994-05-12	5000.0
2	2	Assignment2	Create an Array List of animals	2021-01-16	1	course1	Javascript	Fulltime	1	Klemens	Habgood	1994-05-12	5000.0
3	3	Assignment3	Create an Sorting Algorithm	2021-01-24	1	course1	Javascript	Fulltime	1	Klemens	Habgood	1994-05-12	5000.0
4	4	Assignment4	Create an java consola Appli...	2021-02-02	1	course1	Javascript	Fulltime	1	Klemens	Habgood	1994-05-12	5000.0
5	1	Assignment1	Create a hello world app	2021-01-08	1	course1	Javascript	Fulltime	2	Rina	Vasilchenko	1994-06-06	5000.0
6	2	Assignment2	Create an Array List of animals	2021-01-16	1	course1	Javascript	Fulltime	2	Rina	Vasilchenko	1994-06-06	5000.0
7	3	Assignment3	Create an Sorting Algorithm	2021-01-24	1	course1	Javascript	Fulltime	2	Rina	Vasilchenko	1994-06-06	5000.0
8	4	Assignment4	Create an java consola Appli...	2021-02-02	1	course1	Javascript	Fulltime	2	Rina	Vasilchenko	1994-06-06	5000.0
9	1	Assignment1	Create a hello world app	2021-01-08	2	course1	Javascript	Parttime	1	Klemens	Habgood	1994-05-12	5000.0

**Action Output:**

#	Time	Action	Message	Duration / Fetch
1	16:02:33	use `privateschool`	0 row(s) affected	0,00035 sec
2	16:02:33	SELECT `assignmentspercourse`.`id_assignment` AS ...	64 row(s) returned	0,0022 sec / 0,0000...

A list of students that belong to more than one courses [3 marks]

The screenshot shows a MySQL Workbench interface with a query editor and a results grid.

**Query Editor:**

```
9-StudentMoreThan1Courses
1 -- A list of students that belong to more than one courses
2
3 • use `privateschool`;
4
5 • SELECT
6     `students`.`Sid` AS `Student_ID`,
7     `students`.`firstname` AS `First_Name`,
8     `students`.`lastname` AS `Last_Name`,
9     COUNT(`studentspercourse`.`id_student`) AS `HowManyCourses`
10    FROM
11        `studentspercourse`
12        INNER JOIN
13            `students` ON `students`.`Sid` = `studentspercourse`.`id_student`
14        GROUP BY `studentspercourse`.`id_student`
15        HAVING `HowManyCourses` > 1;
```

**Result Grid:**

#	Student_ID	First_Name	Last_Name	HowManyCourses
1	1	Klemens	Habgood	3
2	2	Rina	Vasilchenko	2
3	3	Tabina	Steinhammer	2
4	6	Marten	Kowalski	2

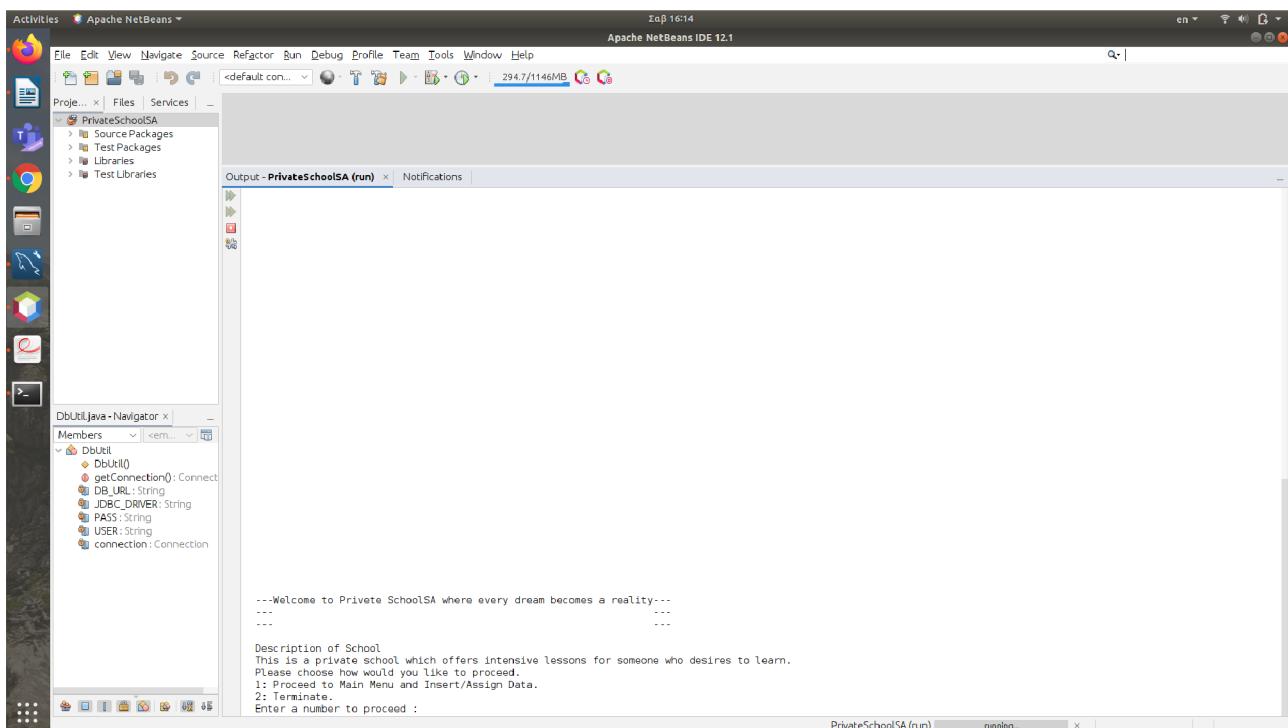
**Action Output:**

	#	Time	Action	Message
✓	1	16:04:02	use `privateschool`	0 row(s) affected
✓	2	16:04:02	SELECT `students`.`Sid` AS `Student_ID`, `students`....	5 row(s) returned

6. You also need to produce a small project that [26 marks]  
 a. makes a connection to the database and executes the above sql queries [9 marks]

<https://github.com/kkousounnis/Java-Application-PrivateSchool-Data-Management-System-Mysql.git>

Project install it to netbeans.



## Set up Database

The screenshot shows the Apache NetBeans IDE interface with the "DbUtil.java" file open in the editor. The code defines a static Connection object and a static method "getConnection()" that uses DriverManager to connect to a MySQL database. The code is annotated with comments explaining the variables and the connection process.

```

package util;
import java.sql.Connection;
import java.sql.DriverManager;
public class DbUtil {
    private static Connection connection = null;
    private static final String JDBC_DRIVER = "com.mysql.cj.jdbc.Driver";
    private static final String DB_URL = "jdbc:mysql://localhost/priveschool";
    private static final String USER = "root";
    private static final String PASS = "password";
    public static Connection getConnection() {
        try {
            System.out.println("Connecting to database...");
            connection = DriverManager.getConnection(DB_URL, USER, PASS);
            System.out.println("Connection Established");
        } catch (Exception e) {
            e.printStackTrace();
        }
        return connection;
    }
}

```

Package:util  
 → DbUtil.java  
 Se  
 USER->  
 Username  
 PASS->  
 Password

Specify your database username and password.

Shift+f6 to run application.

```
--Welcome to Private SchoolsSA where every dream becomes a reality---
---
Description of School
This is a private school which offers intensive lessons for someone who desires to learn.
Please choose how would you like to proceed.
1: Proceed to Main Menu and Insert/Assign Data.
2: Terminate.
Enter a number to proceed :
```

```
Main Menu
Please choose how would you like to proceed
1: Proceed to Course Menu.
2: Proceed to Student Menu.
3: Proceed to Trainer Menu.
4: Proceed to Assignment Menu.
5: Return to strat Menu.
6: Terminate.
Enter a number to proceed : |
```

## List of Courses

```
Course Menu
Please choose how would you like to proceed
1: Course Manually create and insert to Database.
2: Show a list of all courses.
3: Show all students per course.
4: Show all trainers per course.
5: Show all assignments per course.
6: Return.
Enter a number to proceed : 2
Connecting to database...
Connection Established
--Courses--
1: Course{ title=course1, stream=Javascript, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
2: Course{ title=course1, stream=Javascript, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
3: Course{ title=course2, stream=Go, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
4: Course{ title=course2, stream=Go, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
5: Course{ title=course3, stream=Postgresql, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
6: Course{ title=course3, stream=Postgresql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
7: Course{ title=course4, stream=Mysql, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
8: Course{ title=course4, stream=Mysql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
Press Enter key to continue...
```

## List Of Students

```

Student Menu
Please choose how would you like to proceed
1: Create Manually student and insert to Database.
2: Show a list of all the students.
3: Assign students per course.
4: Show assignments per course per student
5: Show students who attend in more than one course.
6: Return.
Enter a number to proceed : 2
---Students---
1: Student{KlemensHabgood, dateOfBirth=12-May-1994, tuitionFees=5000}
2: Student{RinaVasilchenko, dateOfBirth=06-June-1994, tuitionFees=5000}
3: Student{TabinaSteinhammer, dateOfBirth=08-February-1992, tuitionFees=5000}
4: Student{JohannaStanislaw, dateOfBirth=07-March-1994, tuitionFees=5000}
5: Student{LeilaPierce, dateOfBirth=10-April-1993, tuitionFees=5000}
6: Student{MartenKowalski, dateOfBirth=09-May-1995, tuitionFees=5000}
7: Student{ElladineShorrock, dateOfBirth=08-February-1996, tuitionFees=5000}
8: Student{SandersonTankard, dateOfBirth=04-July-1992, tuitionFees=5000}
9: Student{LindsayMatches, dateOfBirth=02-October-1995, tuitionFees=5000}
10: Student{TeadorMiddup, dateOfBirth=11-July-1995, tuitionFees=5000}
Press Enter key to continue...

```

## List of Trainers

```

---Trainers---
1: Trainer{CorissaAbrahim, subject=Backend}
2: Trainer{JudithHowatt, subject=Database}
3: Trainer{MerryKenewell, subject=Backend}
4: Trainer{IsahellaGockelen, subject=Database}
5: Trainer{JudithHowatt, subject=Backend}
6: Trainer{MerryKenewell, subject=Database}
7: Trainer{DenysLambden, subject=Backend}
8: Trainer{NealHairsnape, subject=Database}
Press Enter key to continue...

```

## List of Assignments

```

---Assignments---
1: Assignment{title=Assignment1, description=Create a hello world app, subDateTime=08-January-2021, oralMark=0, totalMark=0}
2: Assignment{title=Assignment2, description=Create an Array List of animals, subDateTime=16-January-2021, oralMark=0, totalMark=0}
3: Assignment{title=Assignment3, description=Create an Sorting Algorithm, subDateTime=24-January-2021, oralMark=0, totalMark=0}
4: Assignment{title=Assignment4, description=Create an java consola Application, subDateTime=02-February-2021, oralMark=0, totalMark=0}
5: Assignment{title=Assignment5, description=Create a Select From query, subDateTime=08-January-2021, oralMark=0, totalMark=0}
6: Assignment{title=Assignment6, description=Create a query with aggregation, subDateTime=16-January-2021, oralMark=0, totalMark=0}
7: Assignment{title=Assignment7, description=Create a nested query, subDateTime=24-January-2021, oralMark=0, totalMark=0}
8: Assignment{title=Assignment8, description=Create an inner join query, subDateTime=02-February-2021, oralMark=0, totalMark=0}
Press Enter key to continue...

```

## All the students per course

```
Course Menu
Please choose how would you like to proceed
1: Course Manually create and insert to Database.
2: Show a list of all courses.
3: Show all students per course.
4: Show all trainers per course.
5: Show all assignments per course.
6: Return.
Enter a number to proceed : 3

Course{ title=course1, stream=Javascript, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
Student{KlemensHabgood, dateOfBirth=12-May-1994, tuitionFees=5000}Student{RinaVasilchenko, dateOfBirth=06-June-1994, tuitionFees=5000}

Course{ title=course1, stream=Javascript, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
Student{KlemensHabgood, dateOfBirth=12-May-1994, tuitionFees=5000}Student{TabinaSteinhammer, dateOfBirth=08-February-1992, tuitionFees=5000}

Course{ title=course2, stream=Go, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
Student{KlemensHabgood, dateOfBirth=12-May-1994, tuitionFees=5000}Student{RinaVasilchenko, dateOfBirth=06-June-1994, tuitionFees=5000}Student{ElladineShorrock, dateOfBirth=10-April-1993, tuitionFees=5000}

Course{ title=course2, stream=Go, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
Student{TabinaSteinhammer, dateOfBirth=08-February-1992, tuitionFees=5000}Student{SandersonTankard, dateOfBirth=04-July-1992, tuitionFees=5000}

Course{ title=course3, stream=Postgresql, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
Student{JohannaStanislaw, dateOfBirth=07-March-1994, tuitionFees=5000}Student{LindsayMatches, dateOfBirth=02-October-1995, tuitionFees=5000}

Course{ title=course3, stream=Postgresql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
Student{LeileaPierce, dateOfBirth=10-April-1993, tuitionFees=5000}

Course{ title=course4, stream=Mysql, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
Student{MartenKowalski, dateOfBirth=09-May-1995, tuitionFees=5000}Student{TeadorMiddup, dateOfBirth=11-July-1995, tuitionFees=5000}

Course{ title=course4, stream=Mysql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
Student{MartenKowalski, dateOfBirth=09-May-1995, tuitionFees=5000}Student{ElladineShorrock, dateOfBirth=08-February-1996, tuitionFees=5000}

Press Enter key to continue...
```

## All the trainers per course.

```
Course Menu
Please choose how would you like to proceed
1: Course Manually create and insert to Database.
2: Show a list of all courses.
3: Show all students per course.
4: Show all trainers per course.
5: Show all assignments per course.
6: Return.
Enter a number to proceed : 4

Course{ title=course1, stream=Javascript, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
Trainer{MerryKenewell, subject=Backend}

Course{ title=course1, stream=Javascript, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
Trainer{CorissaAbrahim, subject=Backend}

Course{ title=course2, stream=Go, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
Trainer{DenysLambden, subject=Backend}

Course{ title=course2, stream=Go, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
Trainer{DenysLambden, subject=Backend}

Course{ title=course3, stream=Postgresql, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
Trainer{NealHairsnape, subject=Database}

Course{ title=course3, stream=Postgresql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
Trainer{MerryKenewell, subject=Database}

Course{ title=course4, stream=Mysql, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
Trainer{IsahellaGockelen, subject=Database}

Course{ title=course4, stream=Mysql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
Trainer{JudithHowatt, subject=Database}

Press Enter key to continue...
```

## All the assignments per course

```
Course Menu
Please choose how would you like to proceed
1: Course Manually create and insert to Database.
2: Show a list of all courses.
3: Show all students per course.
4: Show all trainers per course.
5: Show all assignments per course.
6: Return.
Enter a number to proceed : 5

Course{ title=course1, stream=Javascript, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
Assignment{title=course1, description=Create a hello world app, subDateTime=08-January-2021, oralMark=0, totalMark=0}Assignment{title=course1, description=Create an Array}
Course{ title=course1, stream=Javascript, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
Assignment{title=course1, description=Create a hello world app, subDateTime=08-January-2021, oralMark=0, totalMark=0}Assignment{title=course1, description=Create an Array}
Course{ title=course2, stream=Golang, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
Assignment{title=course2, description=Create a Sorting Algorithm, subDateTime=24-January-2021, oralMark=0, totalMark=0}Assignment{title=course2, description=Create an array}
Course{ title=course2, stream=Golang, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
Assignment{title=course2, description=Create a Sorting Algorithm, subDateTime=24-January-2021, oralMark=0, totalMark=0}Assignment{title=course2, description=Create an array}
Course{ title=course3, stream=Postgresql, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
Assignment{title=course3, description=Create a Select From query, subDateTime=08-January-2021, oralMark=0, totalMark=0}Assignment{title=course3, description=Create an inner join}
Course{ title=course3, stream=Postgresql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
Assignment{title=course3, description=Create a query with aggregation, subDateTime=16-January-2021, oralMark=0, totalMark=0}Assignment{title=course3, description=Create a query}
Course{ title=course4, stream=Mysql, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
Assignment{title=course4, description=Create a query with aggregation, subDateTime=16-January-2021, oralMark=0, totalMark=0}Assignment{title=course4, description=Create a query}
Course{ title=course4, stream=Mysql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
Assignment{title=course4, description=Create a Select From query, subDateTime=08-January-2021, oralMark=0, totalMark=0}Assignment{title=course4, description=Create a query}
Press Enter key to continue...
```

## All the assignments per course per student

```
Output - PrivateSchoolSA (run) | Notifications
▶ Course{ title=course2, stream=Golang, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
Assignment{title=Assignment1, description=Create a hello world app, subDateTime=08-January-2021, oralMark=0, totalMark=0}Assignment{title=Assignment2, description=Create an array}
Student(ElledineShorrock, dateOfBirth=08-February-1996, tuitionFees=5000)
Course{ title=course2, stream=Golang, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
Assignment{title=Assignment1, description=Create a hello world app, subDateTime=08-January-2021, oralMark=0, totalMark=0}Assignment{title=Assignment2, description=Create an array}
Student(TabinoSteinhammer, dateOfBirth=08-February-1992, tuitionFees=5000)
Course{ title=course2, stream=Golang, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
Assignment{title=Assignment1, description=Create a hello world app, subDateTime=08-January-2021, oralMark=0, totalMark=0}Assignment{title=Assignment2, description=Create an array}
Student(SandersonTankard, dateOfBirth=04-July-1992, tuitionFees=5000)
Course{ title=course2, stream=Golang, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
Assignment{title=Assignment1, description=Create a hello world app, subDateTime=08-January-2021, oralMark=0, totalMark=0}Assignment{title=Assignment2, description=Create an array}
Student(JohannaStanislaw, dateOfBirth=07-March-1994, tuitionFees=5000)
Course{ title=course3, stream=Postgresql, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
Assignment{title=Assignment1, description=Create a Select From query, subDateTime=08-January-2021, oralMark=0, totalMark=0}Assignment{title=Assignment2, description=Create an inner join}
Student(LindsayMatches, dateOfBirth=02-October-1995, tuitionFees=5000)
Course{ title=course3, stream=Postgresql, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
Assignment{title=Assignment1, description=Create a Select From query, subDateTime=08-January-2021, oralMark=0, totalMark=0}Assignment{title=Assignment2, description=Create an inner join}
Student(LeilaPierce, dateOfBirth=10-April-1993, tuitionFees=5000)
Course{ title=course3, stream=Postgresql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
Assignment{title=Assignment1, description=Create a Select From query, subDateTime=08-January-2021, oralMark=0, totalMark=0}Assignment{title=Assignment2, description=Create an inner join}
Student(MartenKowalski, dateOfBirth=09-May-1995, tuitionFees=5000)
Course{ title=course4, stream=Mysql, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
Assignment{title=Assignment1, description=Create a Select From query, subDateTime=08-January-2021, oralMark=0, totalMark=0}Assignment{title=Assignment2, description=Create an inner join}
Student(TeadorMiddup, dateOfBirth=11-July-1995, tuitionFees=5000)
Course{ title=course4, stream=Mysql, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
Assignment{title=Assignment1, description=Create a Select From query, subDateTime=08-January-2021, oralMark=0, totalMark=0}Assignment{title=Assignment2, description=Create an inner join}
Student(MartenKowalski, dateOfBirth=09-May-1995, tuitionFees=5000)
Course{ title=course4, stream=Mysql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
Assignment{title=Assignment1, description=Create a Select From query, subDateTime=08-January-2021, oralMark=0, totalMark=0}Assignment{title=Assignment2, description=Create an inner join}
Student(ElledineShorrock, dateOfBirth=08-February-1996, tuitionFees=5000)
Course{ title=course4, stream=Mysql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
Assignment{title=Assignment1, description=Create a Select From query, subDateTime=08-January-2021, oralMark=0, totalMark=0}Assignment{title=Assignment2, description=Create an inner join}
Press Enter key to continue...
```

A list of students that belong to more than one courses.

```
{ Student_Id:1, Firstname:Klemens, Lastname:Habgood, Courses:3
{ Student_Id:2, Firstname:Rina, Lastname:Vasilchenko, Courses:2
{ Student_Id:3, Firstname:Tabina, Lastname:Steinhammer, Courses:2
{ Student_Id:6, Firstname:Marten, Lastname:Kowalski, Courses:2
{ Student_Id:7, Firstname:Elladine, Lastname:Shorrock, Courses:2
```

Press Enter key to continue...

b. makes a connection to the database and inserts input data from the keyboard to the following tables,

i. students [2 marks]

ii. trainers [2 marks]

iii. assignments [2 marks]

iv. courses [2 marks]

v. students per course [3 marks]

vi. trainers per course [3 marks]

vii. assignments per student per course [3 marks]

i. students [2 marks]

```
Choose the way you want to insert your values
Type 1 for step by step input Press 2 for multiple inputs
Enter a number to proceed : 2
Give inputs with this priority
-> [firstname lastname (DD/MM/YYYY) tuitionfees]:
Warning there must be absolutely four inputs if not the system will ask you to retype all four inputs
Warning in third input below please enter DoB like this (DD/MM/YYYY)

Konstantinos Kousounnis 14/12/1996 5000
The input is: 5000
Student successfully inserted to Database.
Press Enter key to continue...
```

ii. trainers [2 marks]

```
Choose the way you want to insert your values
Type 1 for step by step input Press 2 for multiple inputs
Enter a number to proceed : 2
[firstname lastname subject]
Warning there must be absolutely three inputs if not the system will ask you to retype all three inputs
Nikos Alepis Backend
Trainer successfully inserted to Database.
Press Enter key to continue...
```

### iii. assignments [2 marks]

```
Choose the way you want to insert your values
Type 1 for step by step input Press 2 for multiple inputs
Enter a number to proceed : 2

Type like this ->[assignmenttitle assignmentdate description(sentence) ]

Warning there must be three inputs but in third input we can have a sentence consisted of 12 words.
So you are allowed to type maximum 15 words.
If they are more than fifteen words the system will ask you to retype all three inputs consisted of fifteen words.
For example type: --> Assignment1 01/01/2021 Create a simple java program that prints hello world.
Assignment1 01/01/2021 Create a MongoDb database.
Assignment successfully inserted to Database.
Press Enter key to continue...
```

### iv. courses [2 marks]

```
Please give the title of course.
Course4
Please give the name of course
MongoDb
Please type (f) if it is FullTime or type (p) if it is PartTime
f
Please give me the start date of the course
please enter date like this (DD/MM/YYYY)
01/01/2021
Please give me the ending date of the course
please enter date like this (DD/MM/YYYY)
01/03/2021
Course successfully inserted to Database.
Press Enter key to continue...
```

### v. students per course [3 marks]

```
--Students--
1: Student{KlemensHabgood, dateOfBirth=12-May-1994, tuitionFees=5000}
2: Student{RinaVasilchenko, dateOfBirth=06-June-1994, tuitionFees=5000}
3: Student{TabinaSteinhammer, dateOfBirth=08-February-1992, tuitionFees=5000}
4: Student{JohannaStanislaw, dateOfBirth=07-March-1994, tuitionFees=5000}
5: Student{LeilaPierce, dateOfBirth=10-April-1993, tuitionFees=5000}
6: Student{MartenKowalski, dateOfBirth=09-May-1995, tuitionFees=5000}
7: Student{ElladineShorrock, dateOfBirth=08-February-1996, tuitionFees=5000}
8: Student{SandersonTankard, dateOfBirth=04-July-1992, tuitionFees=5000}
9: Student{LindsayMatches, dateOfBirth=02-October-1995, tuitionFees=5000}
10: Student{TeadorMiddup, dateOfBirth=11-July-1995, tuitionFees=5000}
11: Student{KonstantinosKousounnis, dateOfBirth=14-December-1996, tuitionFees=5000}
Please specify student from List by typing number
Enter a number to proceed :
```

First specify student.

```

Please specify student from List by typing number
Enter a number to proceed : 2
---Courses---
1: Course{ title=course1, stream=Javascript, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
2: Course{ title=course1, stream=Javascript, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
3: Course{ title=course2, stream=Go, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
4: Course{ title=course2, stream=Go, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
5: Course{ title=course3, stream=Postgresql, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
6: Course{ title=course3, stream=Postgresql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
7: Course{ title=course4, stream=Mysql, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
8: Course{ title=course4, stream=Mysql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
9: Course{ title=Course4, stream=MongoDb, type=FullTime, startDate=01-January-2021, endDate=01-March-2021}
Please tell me to which course will the student attend please type number of course.
Enter a number to proceed :

```

PrivateSchoolSA (run)

## Specify course

```

---Students---
1: Student{KlemensHabgood, dateOfBirth=12-May-1994, tuitionFees=5000}
2: Student{RinaVasilchenko, dateOfBirth=06-June-1994, tuitionFees=5000}
3: Student{TabinaSteinhammer, dateOfBirth=08-February-1992, tuitionFees=5000}
4: Student{JohannaStanislaw, dateOfBirth=07-March-1994, tuitionFees=5000}
5: Student{LeillaPierce, dateOfBirth=10-April-1993, tuitionFees=5000}
6: Student{MartenKowalski, dateOfBirth=09-May-1995, tuitionFees=5000}
7: Student{ElladineShorrock, dateOfBirth=08-February-1996, tuitionFees=5000}
8: Student{SandersonTankard, dateOfBirth=04-July-1992, tuitionFees=5000}
9: Student{LindsayMatches, dateOfBirth=02-October-1995, tuitionFees=5000}
10: Student{TeadorMiddup, dateOfBirth=11-July-1995, tuitionFees=5000}
11: Student{KonstantinosKousounnis, dateOfBirth=14-December-1996, tuitionFees=5000}
Please specify student from List by typing number
Enter a number to proceed : 2
---Courses---
1: Course{ title=course1, stream=Javascript, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
2: Course{ title=course1, stream=Javascript, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
3: Course{ title=course2, stream=Go, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
4: Course{ title=course2, stream=Go, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
5: Course{ title=course3, stream=Postgresql, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
6: Course{ title=course3, stream=Postgresql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
7: Course{ title=course4, stream=Mysql, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
8: Course{ title=course4, stream=Mysql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
9: Course{ title=Course4, stream=MongoDb, type=FullTime, startDate=01-January-2021, endDate=01-March-2021}
Please tell me to which course will the student attend please type number of course.
Enter a number to proceed : 3
Assignment of students per course successfully inserted to Database.
Student Menu
Please choose how would you like to proceed
1: Create Manually student and insert to Database.
2: Show a list of all the students.
3: Assign students per course.
4: Show assignments per course per student
5: Show students who attend in more than one course.
6: Return.
Enter a number to proceed : |

```

## vi. trainers per course [3 marks]

```

---Trainers---
1: Trainer{CorissaAbrahim, subject=Backend}
2: Trainer{JudithHowatt, subject=Database}
3: Trainer{MerryKenewell, subject=Backend}
4: Trainer{IsahellaGockelen, subject=Database}
5: Trainer{JudithHowatt, subject=Backend}
6: Trainer{MerryKenewell, subject=Database}
7: Trainer{DenysLambden, subject=Backend}
8: Trainer{NealHairsnape, subject=Database}
9: Trainer{NikosAlepis, subject=Backend}
Please specify trainer from List by typing number
Enter a number to proceed : 2
--Courses--
1: Course{ title=course1, stream=Javascript, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
2: Course{ title=course1, stream=Javascript, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
3: Course{ title=course2, stream=Go, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
4: Course{ title=course2, stream=Go, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
5: Course{ title=course3, stream=Postgresql, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
6: Course{ title=course3, stream=Postgresql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
7: Course{ title=course4, stream=Mysql, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
8: Course{ title=course4, stream=Mysql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
9: Course{ title=Course4, stream=MongoDb, type=FullTime, startDate=01-January-2021, endDate=01-March-2021}
Please tell me to which course will the trainer attend please type number of course.
Enter a number to proceed : 3
Assignment of trainers per course successfully inserted to Database.
    Trainer Menu
Please choose how would you like to proceed.
1: Create Manually trainer and insert to Database.
2: Show all trainers
3: Assign trainers per course.
4: Return.
Enter a number to proceed : |

```

## vii. assignments per student per course [3 marks]

```

---Assignments---
1: Assignment{title=Assignment1, description=Create a hello world app, subDateTime=08-January-2021, oralMark=0, totalMark=0}
2: Assignment{title=Assignment2, description=Create an Array List of animals, subDateTime=16-January-2021, oralMark=0, totalMark=0}
3: Assignment{title=Assignment3, description=Create an Sorting Algorithm, subDateTime=24-January-2021, oralMark=0, totalMark=0}
4: Assignment{title=Assignment4, description=Create an java consola Application, subDateTime=02-February-2021, oralMark=0, totalMark=0}
5: Assignment{title=Assignment5, description=Create a Select From query, subDateTime=08-January-2021, oralMark=0, totalMark=0}
6: Assignment{title=Assignment6, description=Create a quyr with aggregation, subDateTime=16-January-2021, oralMark=0, totalMark=0}
7: Assignment{title=Assignment7, description=Create a nested query, subDateTime=24-January-2021, oralMark=0, totalMark=0}
8: Assignment{title=Assignment8, description=Create an inner join query, subDateTime=02-February-2021, oralMark=0, totalMark=0}
9: Assignment{title=Assignment6, description= Create a MongoDb database., subDateTime=01-January-2021, oralMark=0, totalMark=0}
Please specify assignments from List by typing number
Enter a number to proceed : 5
--Courses--
1: Course{ title=course1, stream=Javascript, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
2: Course{ title=course1, stream=Javascript, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
3: Course{ title=course2, stream=Go, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
4: Course{ title=course2, stream=Go, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
5: Course{ title=course3, stream=Postgresql, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
6: Course{ title=course3, stream=Postgresql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
7: Course{ title=course4, stream=Mysql, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
8: Course{ title=course4, stream=Mysql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
9: Course{ title=Course4, stream=MongoDb, type=FullTime, startDate=01-January-2021, endDate=01-March-2021}
Please tell me to which course will the assignments attend please type number of course.
Enter a number to proceed : 6
Assignment of assignments per course successfully inserted to Database.
    Assignment Menu
Please choose how would you like to proceed
1: Create manually assignment and insert toDatabase.
2: Show all assignments.
3: Assign assignments per student per course.
4: Return.
Enter a number to proceed : |

```