

Weekly coding assignment

1Q)

The screenshot shows the Eclipse IDE with two Java files open: `Library.java` and `BusinessLogic.java`. The `Library.java` file defines a `Library` class with attributes `id`, `name`, and `author`, and methods for getting and setting these attributes. The `BusinessLogic.java` file defines a `BusinessLogic` class with methods for adding, displaying, updating, and deleting books. The console output shows the results of these operations.

```
package com.codingassignment3;

public class Library {
    int id;
    String name;
    String author;

    public Library(int id, String name, String author) {
        super();
        this.id = id;
        this.name = name;
        this.author = author;
    }

    public Library() {
        super();
    }

    @Override
    public String toString() {
        return "Library [id=" + id + ", name=" + name + ", author=" + author + "]";
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }
}
```

```
package com.codingassignment3;
import java.util.ArrayList;

public class BusinessLogic {

    public ArrayList<Library> addBooks() {
        Library p = new Library(112, "The Christmas Pig", "JK Rowling");
        Library p1 = new Library(223, "Whereabouts", "Amar");
        Library p2 = new Library(334, "Queen of Fire", "Devika Rangachari");
        Library p3 = new Library(445, "Listen to Your Heart: The London Adventure", "Ruskin Bond");
        Library p4 = new Library(556, "A Place Called Home", "Preeti Shenoy");

        ArrayList<Library> a = new ArrayList<Library>();
        a.add(p);
        a.add(p1);
        a.add(p2);
        a.add(p3);
        a.add(p4);
        return a;
    }

    void displayBooks(ArrayList<Library> b) {
        System.out.println("List of Books");
        for (Library s : b) {
            System.out.println(s);
        }
    }

    void updateBooks(ArrayList<Library> p, String a, String b) {
        System.out.println("List of Books data after Update::: ");
        for (Library e : p) {
            if (e.getAuthor().equals(a)) {
                e.setAuthor(b);
                System.out.println(e);
            }
        }
    }

    void deleteBook(ArrayList<Library> p, String a) {
        System.out.println("List of Books data after Deletion::: ");
        for (Library k : p) {
            if (k.getAuthor().equals(a)) {
                k.setName("");
                k.setId(0);
                k.setAuthor("");
            }
        }
    }
}
```

The screenshot shows the Eclipse IDE with the `Tester.java` file open. The `Tester` class contains a `main` method that demonstrates the functionality of the `Library` and `BusinessLogic` classes. The console output shows the results of these operations.

```
package com.codingassignment3;
import java.util.ArrayList;

public class Tester {

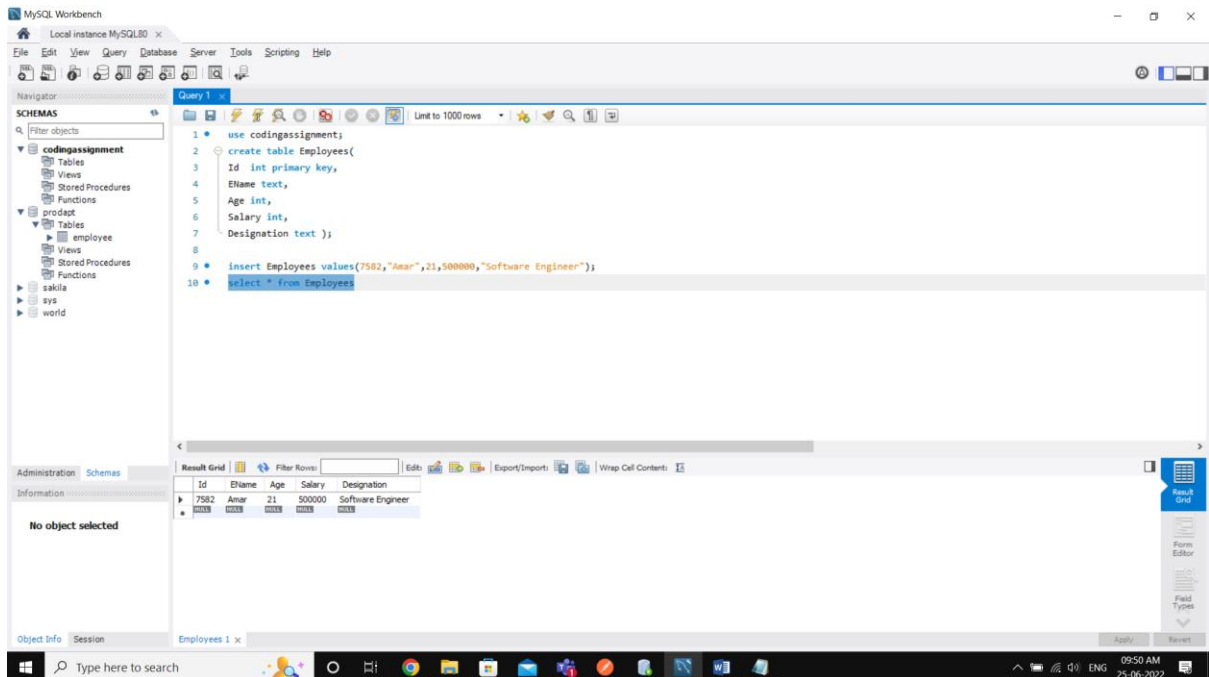
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        BusinessLogic o = new BusinessLogic();
        ArrayList<Library> z = o.addBooks();
        o.displayBooks(z);
        o.updateBooks(z, "Amar", "Jhumpa Lahiri");
        o.deleteBook(z, "Jhumpa Lahiri");
    }
}
```

```
terminated- tester [Java Application] C:\Users\amarpraveen\p2\pools\plugins\org.eclipse.justi.openjdk.hotspot...
List of Books
Library [id=112, name=The Christmas Pig, author=JK Rowling]
Library [id=223, name=Whereabouts, author=Amar]
Library [id=334, name=Queen of Fire, author=Devika Rangachari]
Library [id=445, name=Listen to Your Heart: The London Adventure, author=Ruskin Bond]
Library [id=556, name=A Place Called Home, author=Preeti Shenoy]

List of Books data after Update:::
Library [id=223, name=Whereabouts, author=Jhumpa Lahiri]

List of Books data after Deletion:::
Library [id=112, name=The Christmas Pig, author=JK Rowling]
Library [id=0, name=, author=]
Library [id=334, name=Queen of Fire, author=Devika Rangachari]
Library [id=445, name=Listen to Your Heart: The London Adventure, author=Ruskin Bond]
Library [id=556, name=A Place Called Home, author=Preeti Shenoy]
```

2Q)



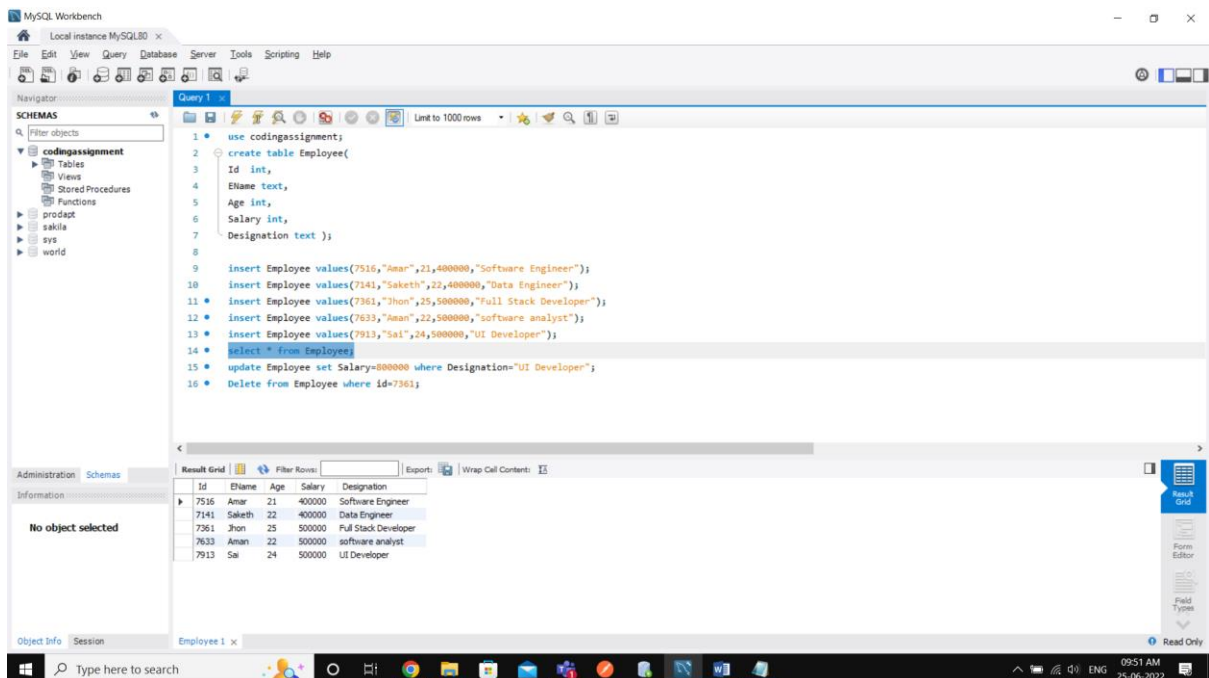
MySQL Workbench interface showing a query window with the following SQL code:

```
1 use codingassignment;
2 create table Employees(
3   Id int primary key,
4   EName text,
5   Age int,
6   Salary int,
7   Designation text );
8
9 insert Employees values(7582,"Amar",21,500000,"Software Engineer");
10 select * from Employees
```

The result grid shows one row of data:

7582	Amar	21	500000	Software Engineer	

3Q)



MySQL Workbench interface showing a query window with the following SQL code:

```
1 use codingassignment;
2 create table Employee(
3   Id int,
4   EName text,
5   Age int,
6   Salary int,
7   Designation text );
8
9 insert Employee values(7516,"Amar",21,400000,"Software Engineer");
10 insert Employee values(7141,"Saketh",22,400000,"Data Engineer");
11 insert Employee values(7361,"Jhon",25,500000,"Full Stack Developer");
12 insert Employee values(7633,"Aman",22,500000,"software analyst");
13 insert Employee values(7913,"Sai",24,500000,"UI Developer");
14 select * from Employee;
15 update Employee set Salary=800000 where Designation="UI Developer";
16 Delete from Employee where id=7361;
```

The result grid shows five rows of data:

7516	Amar	21	400000	Software Engineer	
7141	Saketh	22	400000	Data Engineer	
7361	Jhon	25	500000	Full Stack Developer	
7633	Aman	22	500000	software analyst	
7913	Sai	24	500000	UI Developer	

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- codingassignment
 - Tables
 - Views
 - Stored Procedures
 - Functions
- prodapt
- sakila
- sys
- world

Query 1

```
1 use codingassignment;
2 create table Employee(
3   Id int,
4   EName text,
5   Age int,
6   Salary int,
7   Designation text );
8
9 insert Employee values(7516,"Amar",21,400000,"Software Engineer");
10 insert Employee values(7141,"Saketh",22,400000,"Data Engineer");
11 insert Employee values(7361,"Jhon",25,500000,"Full Stack Developer");
12 insert Employee values(7633,"Aman",22,500000,"software analyst");
13 insert Employee values(7913,"Sai",24,800000,"UI Developer");
14 select * from Employee;
15 update Employee set Salary=800000 where Designation="UI Developer";
16 Delete from Employee where id=7361;
```

Result Grid

	Id	EName	Age	Salary	Designation
▶	7516	Amar	21	400000	Software Engineer
	7141	Saketh	22	400000	Data Engineer
	7633	Aman	22	500000	software analyst
	7913	Sai	24	800000	UI Developer

Administration Schemas Information

No object selected

Employee 2 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
10	09:52:01	Delete from Employee where id=7361	1 row(s) affected	0.015 sec
11	09:52:04	select * from Employee LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

09:52 AM 25-06-2022