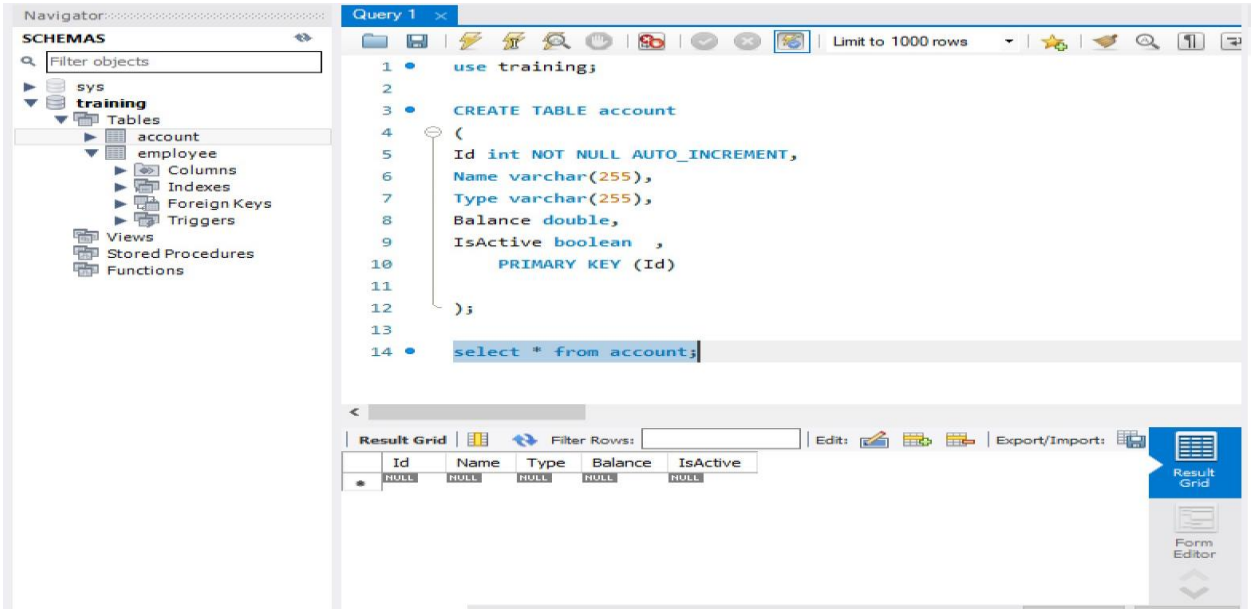
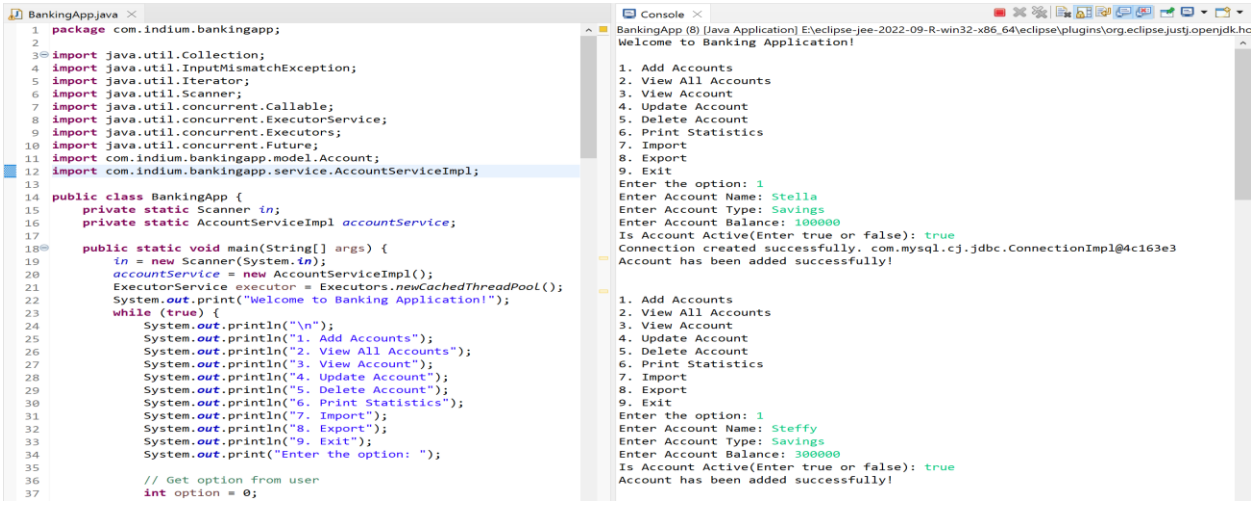
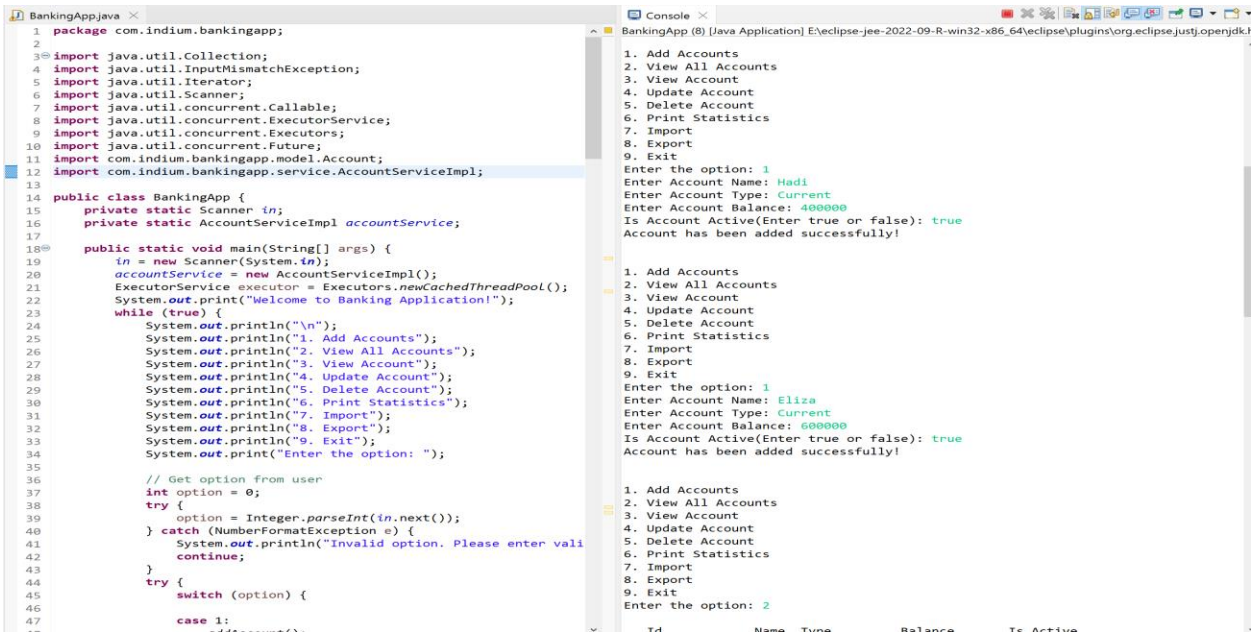
Banking App Program using JDBC Screenshots

1. Create Accounts & View all Accounts.

No Data in Accounts Table



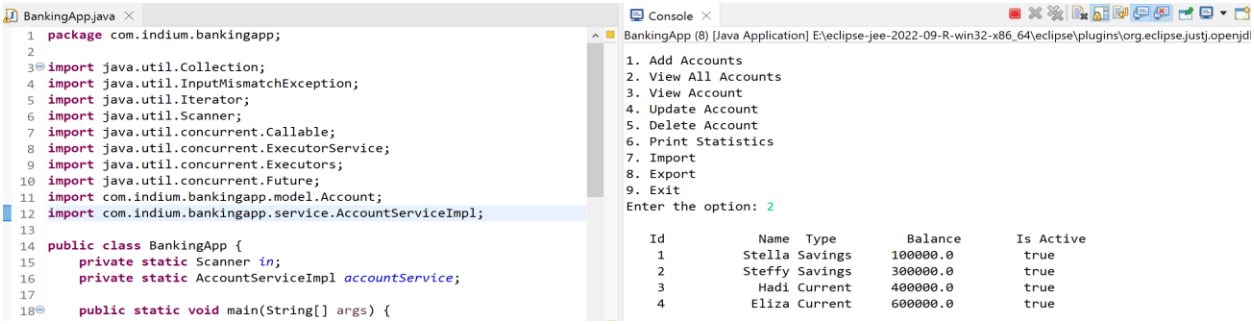




```
BankingApp.java
```

```java
package com.indium.bankingapp;

import java.util.Collection;
import java.util.InputMismatchException;
import java.util.Iterator;
import java.util.Scanner;
import java.util.concurrent.Callable;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.Future;
import com.indium.bankingapp.model.Account;
import com.indium.bankingapp.service.AccountServiceImpl;

public class BankingApp {
    private static Scanner in;
    private static AccountServiceImpl accountService;

    public static void main(String[] args) {
        in = new Scanner(System.in);
        accountService = new AccountServiceImpl();
        ExecutorService executor = Executors.newCachedThreadPool();
        System.out.print("Welcome to Banking Application!");
        while (true) {
            System.out.println("\n");
            System.out.println("1. Add Accounts");
            System.out.println("2. View All Accounts");
            System.out.println("3. View Account");
            System.out.println("4. Update Account");
            System.out.println("5. Delete Account");
            System.out.println("6. Print Statistics");
            System.out.println("7. Import");
            System.out.println("8. Export");
            System.out.println("9. Exit");
            System.out.print("Enter the option: ");

            // Get option from user
            int option = 0;
            try {
                option = Integer.parseInt(in.next());
            } catch (NumberFormatException e) {
                System.out.println("Invalid option. Please enter vali
                continue;
            }
            try {
                switch (option) {

                case 1:
```

```
Console
BankingApp (8) [Java Application] E:\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.h

1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 1
Enter Account Name: Hadi
Enter Account Type: Current
Enter Account Balance: 400000
Is Account Active(Enter true or false): true
Account has been added successfully!


1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 1
Enter Account Name: Eliza
Enter Account Type: Current
Enter Account Balance: 600000
Is Account Active(Enter true or false): true
Account has been added successfully!


1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 2

   Id            Name  Type          Balance       Is Active
```

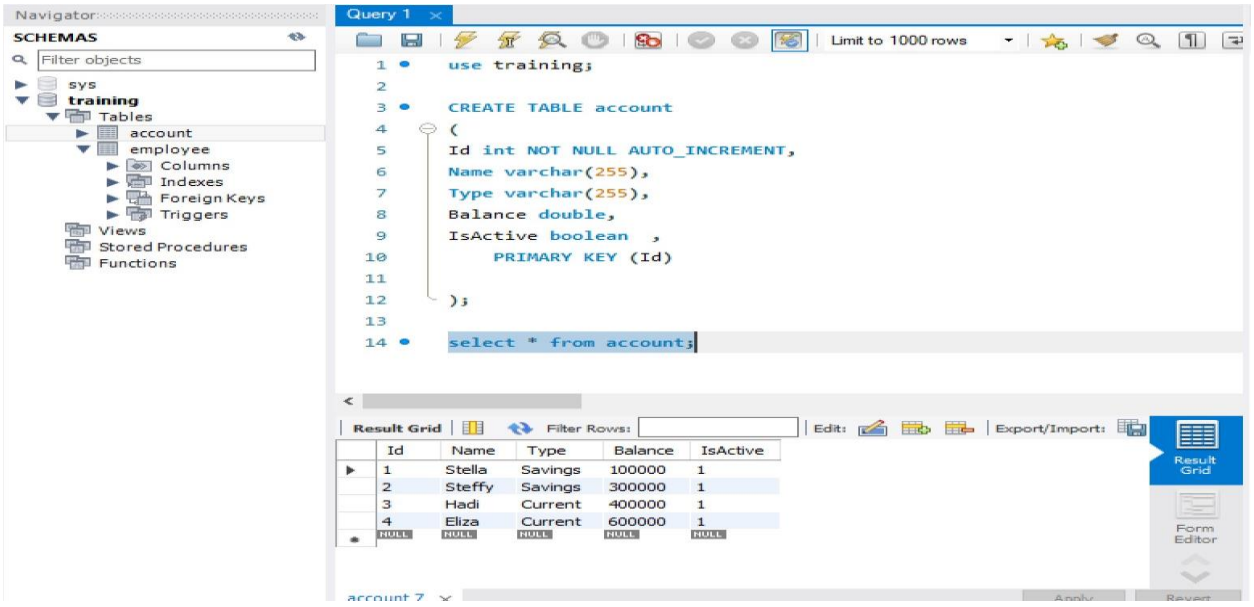


```
BankingApp.java
```

```java
package com.indium.bankingapp;

import java.util.Collection;
import java.util.InputMismatchException;
import java.util.Iterator;
import java.util.Scanner;
import java.util.concurrent.Callable;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.Future;
import com.indium.bankingapp.model.Account;
import com.indium.bankingapp.service.AccountServiceImpl;

public class BankingApp {
    private static Scanner in;
    private static AccountServiceImpl accountService;

    public static void main(String[] args) {
```

```
Console
BankingApp (8) [Java Application] E:\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk

1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 2

   Id            Name  Type          Balance       Is Active
    1          Stella Savings      100000.0          true
    2          Steffy Savings      300000.0          true
    3            Hadi Current      400000.0          true
    4           Eliza Current      600000.0          true
```

After Creation of accounts



```
Navigator
SCHEMAS
Filter objects
sys
training
Tables
account
employee
Columns
Indexes
Foreign Keys
Triggers
Views
Stored Procedures
Functions
```

```
Query 1
Limit to 1000 rows
```

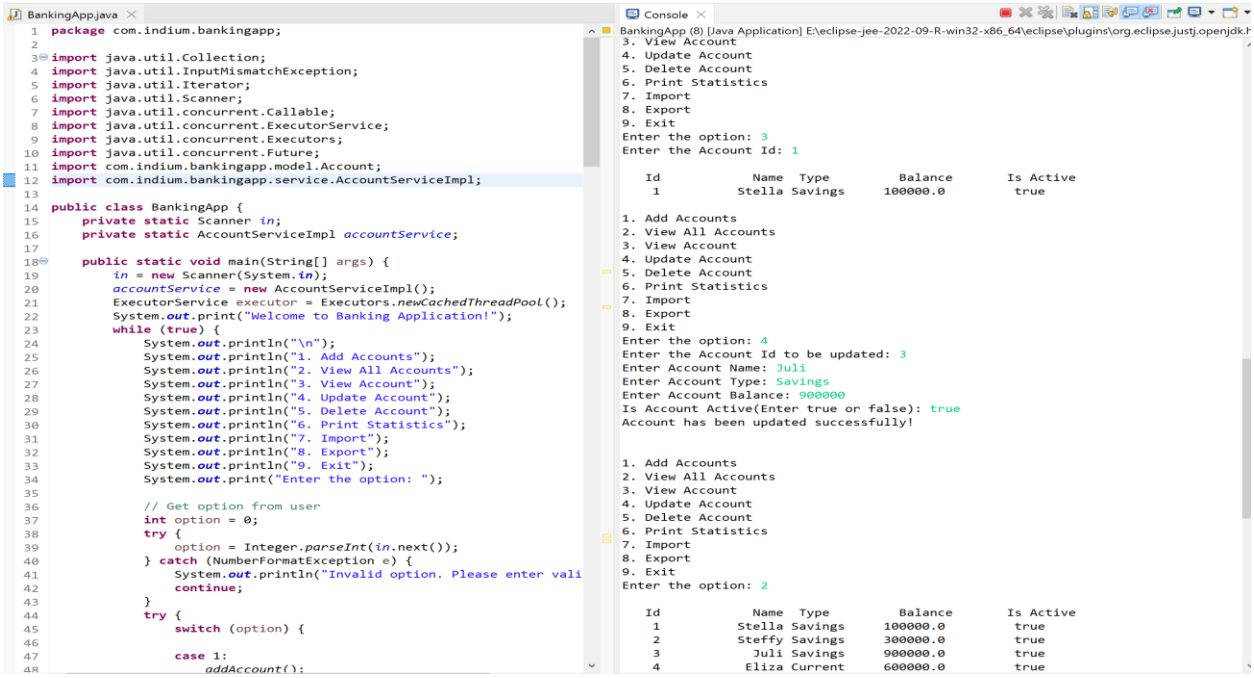
```sql
use training;

CREATE TABLE account
(
Id int NOT NULL AUTO_INCREMENT,
Name varchar(255),
Type varchar(255),
Balance double,
IsActive boolean  ,
    PRIMARY KEY (Id)

);

select * from account;
```

```
Result Grid | Filter Rows: | Edit: | Export/Import:
```

Id	Name	Type	Balance	IsActive
1	Stella	Savings	100000	1
2	Steffy	Savings	300000	1
3	Hadi	Current	400000	1
4	Eliza	Current	600000	1
NULL	NULL	NULL	NULL	NULL

```
Result Grid
Form Editor
account 7
Apply
Revert
```

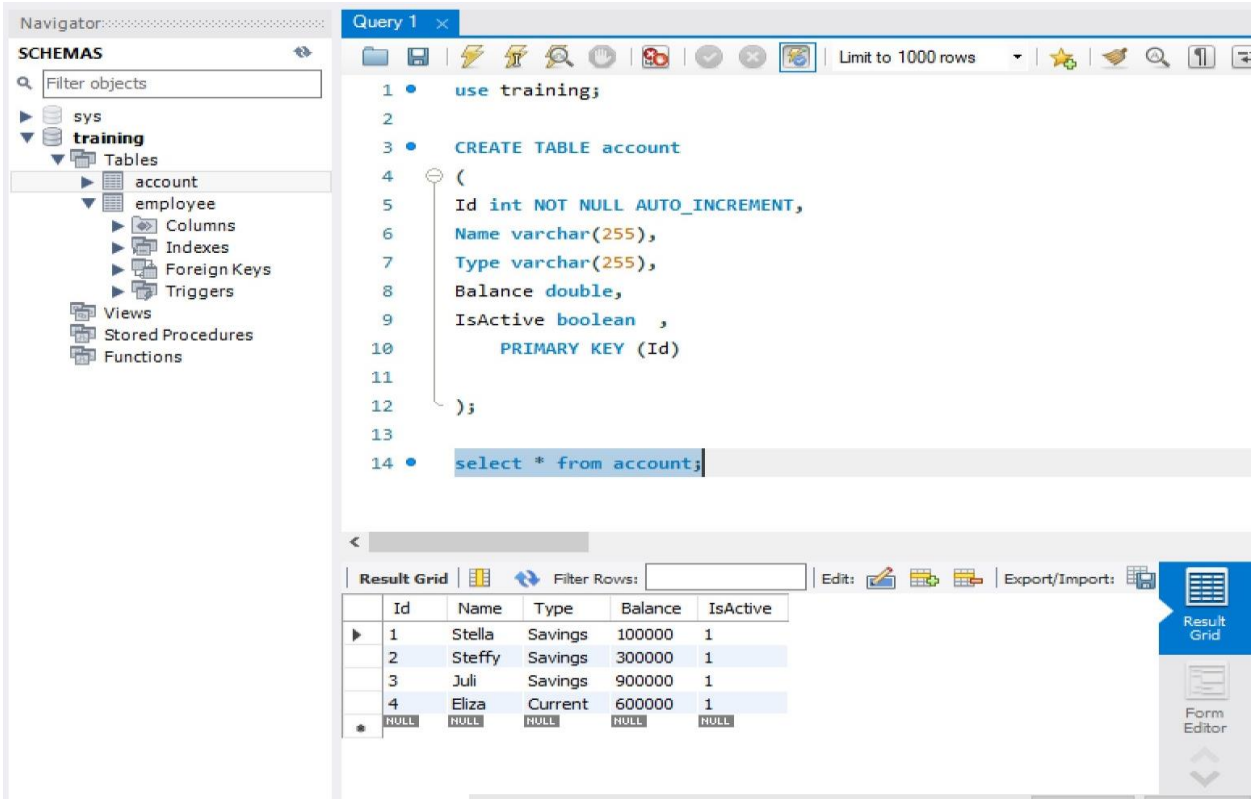
2. View Account, Update Accounts & View all accounts.



```
BankingApp.java

package com.indium.bankingapp;

import java.util.Collection;
import java.util.InputMismatchException;
import java.util.Iterator;
import java.util.Scanner;
import java.util.concurrent.Callable;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.Future;
import com.indium.bankingapp.model.Account;
import com.indium.bankingapp.service.AccountServiceImpl;

public class BankingApp {
    private static Scanner in;
    private static AccountServiceImpl accountService;

    public static void main(String[] args) {
        in = new Scanner(System.in);
        accountService = new AccountServiceImpl();
        ExecutorService executor = Executors.newCachedThreadPool();
        System.out.print("Welcome to Banking Application!");
        while (true) {
            System.out.println("\n");
            System.out.println("1. Add Accounts");
            System.out.println("2. View All Accounts");
            System.out.println("3. View Account");
            System.out.println("4. Update Account");
            System.out.println("5. Delete Account");
            System.out.println("6. Print Statistics");
            System.out.println("7. Import");
            System.out.println("8. Export");
            System.out.println("9. Exit");
            System.out.print("Enter the option: ");

            // Get option from user
            int option = 0;
            try {
                option = Integer.parseInt(in.next());
            } catch (NumberFormatException e) {
                System.out.println("Invalid option. Please enter vali
                continue;
            }
            try {
                switch (option) {

                case 1:
                    addAccount();
```

```
Console
BankingApp (8) [Java Application] E:\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.h
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 3
Enter the Account Id: 1

   Id            Name  Type          Balance       Is Active
    1          Stella Savings     100000.0          true

1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 4
Enter the Account Id to be updated: 3
Enter Account Name: Juli
Enter Account Type: Savings
Enter Account Balance: 900000
Is Account Active(Enter true or false): true
Account has been updated successfully!


1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 2

   Id            Name  Type          Balance       Is Active
    1          Stella Savings     100000.0          true
    2          Steffy Savings     300000.0          true
    3            Juli Savings     900000.0          true
    4           Eliza Current     600000.0          true
```

Account table post update

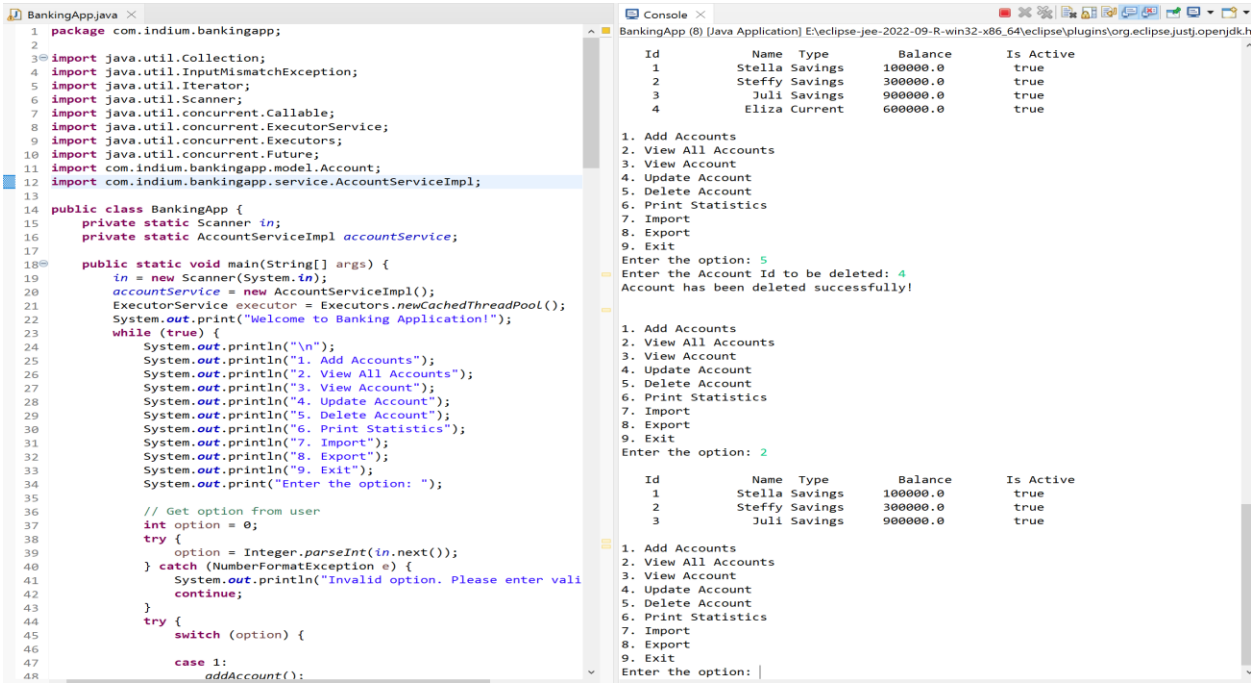


```
Navigator:
SCHEMAS
Filter objects
sys
training
  Tables
    account
    employee
      Columns
      Indexes
      Foreign Keys
      Triggers
  Views
  Stored Procedures
  Functions

Query 1
Limit to 1000 rows

1  use training;
2
3  CREATE TABLE account
4  (
5  Id int NOT NULL AUTO_INCREMENT,
6  Name varchar(255),
7  Type varchar(255),
8  Balance double,
9  IsActive boolean  ,
10     PRIMARY KEY (Id)
11
12 );
13
14 select * from account;

Result Grid | Filter Rows: | Edit: | Export/Import:
```

Id	Name	Type	Balance	IsActive
1	Stella	Savings	100000	1
2	Steffy	Savings	300000	1
3	Juli	Savings	900000	1
4	Eliza	Current	600000	1
NULL	NULL	NULL	NULL	NULL

```
Result Grid
Form Editor
```

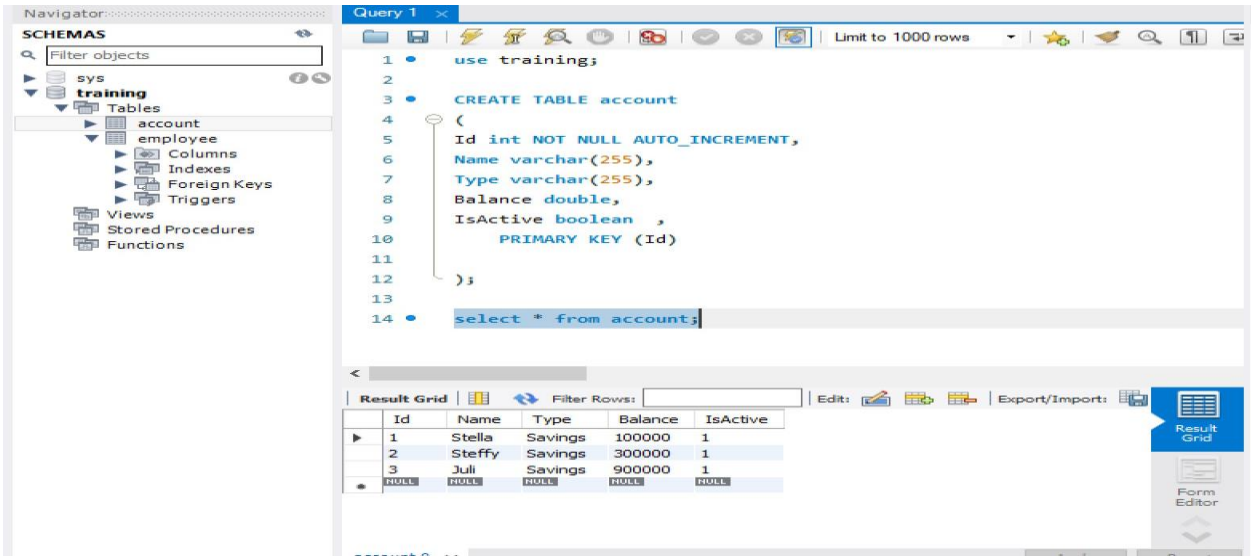
3. Delete Accounts



```
BankingApp.java

package com.indium.bankingapp;

import java.util.Collection;
import java.util.InputMismatchException;
import java.util.Iterator;
import java.util.Scanner;
import java.util.concurrent.Callable;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.Future;
import com.indium.bankingapp.model.Account;
import com.indium.bankingapp.service.AccountServiceImpl;

public class BankingApp {
    private static Scanner in;
    private static AccountServiceImpl accountService;

    public static void main(String[] args) {
        in = new Scanner(System.in);
        accountService = new AccountServiceImpl();
        ExecutorService executor = Executors.newCachedThreadPool();
        System.out.print("Welcome to Banking Application!");
        while (true) {
            System.out.println("\n");
            System.out.println("1. Add Accounts");
            System.out.println("2. View All Accounts");
            System.out.println("3. View Account");
            System.out.println("4. Update Account");
            System.out.println("5. Delete Account");
            System.out.println("6. Print Statistics");
            System.out.println("7. Import");
            System.out.println("8. Export");
            System.out.println("9. Exit");
            System.out.print("Enter the option: ");

            // Get option from user
            int option = 0;
            try {
                option = Integer.parseInt(in.next());
            } catch (NumberFormatException e) {
                System.out.println("Invalid option. Please enter vali
                continue;
            }
            try {
                switch (option) {

                case 1:
                    addAccount();
```

```
Console
BankingApp (8) [Java Application] E:\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.h

  Id          Name  Type         Balance       Is Active
   1        Stella Savings      100000.0         true
   2        Steffy Savings      300000.0         true
   3          Juli Savings      900000.0         true
   4         Eliza Current      600000.0         true

1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 5
Enter the Account Id to be deleted: 4
Account has been deleted successfully!


1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option: 2

  Id          Name  Type         Balance       Is Active
   1        Stella Savings      100000.0         true
   2        Steffy Savings      300000.0         true
   3          Juli Savings      900000.0         true

1. Add Accounts
2. View All Accounts
3. View Account
4. Update Account
5. Delete Account
6. Print Statistics
7. Import
8. Export
9. Exit
Enter the option:
```

Accounts table Post Delete



```
Navigator
SCHEMAS
Filter objects
sys
training
  Tables
    account
    employee
      Columns
      Indexes
      Foreign Keys
      Triggers
  Views
  Stored Procedures
  Functions

Query 1
Limit to 1000 rows

use training;

CREATE TABLE account
(
Id int NOT NULL AUTO_INCREMENT,
Name varchar(255),
Type varchar(255),
Balance double,
IsActive boolean  ,
    PRIMARY KEY (Id)

);

select * from account;

Result Grid | Filter Rows: | Edit: | Export/Import:

| Id | Name | Type | Balance | IsActive |
|---|---|---|---|---|
| 1 | Stella | Savings | 100000 | 1 |
| 2 | Steffy | Savings | 300000 | 1 |
| 3 | Juli | Savings | 900000 | 1 |
| NULL | NULL | NULL | NULL | NULL |

Result Grid
Form Editor
```