UserDetails class

```
package com.bankapplication;


public class UserDetails {


	private Integer id;

	private String name;

	private String email;

	private String password;

	@Override

	public String toString() {

		return "UserDetails [name=" + name + ", email=" +
email + ", password=" + password + "]";

	}


	public UserDetails(Integer id, String name, String email,
String password) {

		super();

		this.id = id;

		this.name = name;

		this.email = email;

		this.password = password;

	}
```

```
public String getName() {
	return name;
}
public void setName(String name) {
	this.name = name;
}
public String getEmail() {
	return email;
}
public void setEmail(String email) {
	this.email = email;
}
public String getPassword() {
	return password;
}
public void setPassword(String password) {
	this.password = password;
}

public UserDetails() {
	// TODO Auto-generated constructor stub
}

public Integer getId() {
		return id;
	}

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



}
```

MoneyDetails class

```
package com.bankapplication;

import java.sql.Date;

public class MoneyDetails {

	private Date date;
	private Integer id;
	private Float balance;
	private String category;
	@Override
	public String toString() {
		return "MoneyDetails [date=" + date + ", id=" + id +
", balance=" + balance + ", category=" + category + "]";
	}
	public MoneyDetails(Date date, Integer id, Float balance,
String category) {
		super();
		this.date = date;
		this.id = id;
		this.balance = balance;
		this.category = category;
	}
	public Date getDate() {
		return date;
	}
	public void setDate(Date date) {
		this.date = date;
	}
	public Integer getId() {
		return id;
	}
	public void setId(Integer id) {
		this.id = id;
	}
	public Float getBalance() {
		return balance;
	}
	public void setBalance(Float balance) {
		this.balance = balance;
	}
	public String getCategory() {
		return category;
	}
	public void setCategory(String category) {
		this.category = category;
	}

	public MoneyDetails() {
		// TODO Auto-generated constructor stub
	}


}
```

Operations.java

```
package com.bankapplication;


import javax.sql.DataSource;


import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.jdbc.core.JdbcTemplate;

import
org.springframework.jdbc.core.namedparam.MapSqlParameterSource;

import
org.springframework.jdbc.core.namedparam.NamedParameterJdbcTempla
te;

import org.springframework.stereotype.Component;


@Component

public class Operations {


	DataSource dataSource;

	NamedParameterJdbcTemplate namedParameterJdbcTemplate;

	JdbcTemplate jdbcTemplate;


	@Autowired

	public void setDataSource(DataSource dataSource) {

		namedParameterJdbcTemplate=new
NamedParameterJdbcTemplate(dataSource);

		jdbcTemplate=new JdbcTemplate(dataSource);
	}

	// registering the user

	int registerUser(UserDetails user) {
		String qry="insert into userinfo values
(:id,:name,:email,:password)";
		MapSqlParameterSource source=new
MapSqlParameterSource()
					.addValue("id", user.getId())
					.addValue("name", user.getName())
					.addValue("email", user.getEmail())
					.addValue("password", user.getPassword());
		return namedParameterJdbcTemplate.update(qry, source);

	}



	Float depositing(Float depositBalance, Integer id) {

	String qry="select balance from moneydetails where id=";
	MapSqlParameterSource source=new MapSqlParameterSource()
			.addValue("id", id);
	Float
balance=jdbcTemplate.queryForObject(qry,Float.class,source);
```

```
      return balance+depositBalance;

      }

}
```

App.java

```
package com.bankapplication;

import java.util.Scanner;

import
org.springframework.context.support.ClassPathXmlApplicationContex
t;

/**

 * Hello world!

 *

 */

public class App {

      public static void main( String[] args )

    {

        System.out.println( "Hello World!" );
        ClassPathXmlApplicationContext context=new
ClassPathXmlApplicationContext("config.xml");


        Operations
operations=context.getBean("operations",Operations.class);

        String signing;

        Scanner sc=new Scanner(System.in);

        String runOrStop;

        Integer chocieOfOperation;


        do {


           System.out.println("Please enter signup for adding and
signin for signing in");

           signing=sc.next();

           if(signing.equalsIgnoreCase("signup")) {

                System.out.println("Please neter name email
password");

                String name=sc.next();

                String email=sc.next();

                String password=sc.next();

                Integer id=sc.nextInt();

                operations.registerUser(new
UserDetails(id,name,email,password));
```

```
            }

            if(signing.equalsIgnoreCase("signin")) {

                System.out.println("Please enter your choice");
                chocieOfOperation=sc.nextInt();
                switch(chocieOfOperation) {

                case 1:
                    System.out.println("Deposting the money");
                    Float depositbalance=sc.nextFloat();
                    Integer id=sc.nextInt();
                    System.out.println("updated Balance: 
"+operations.depositing(depositbalance,id));

                }
            }
            System.out.println("Enter y to run and anyother key to 
stop");
            runOrStop=sc.next();
        }while(runOrStop.equalsIgnoreCase("y"));
    }
}
```

Database Tables

Userinfo table;

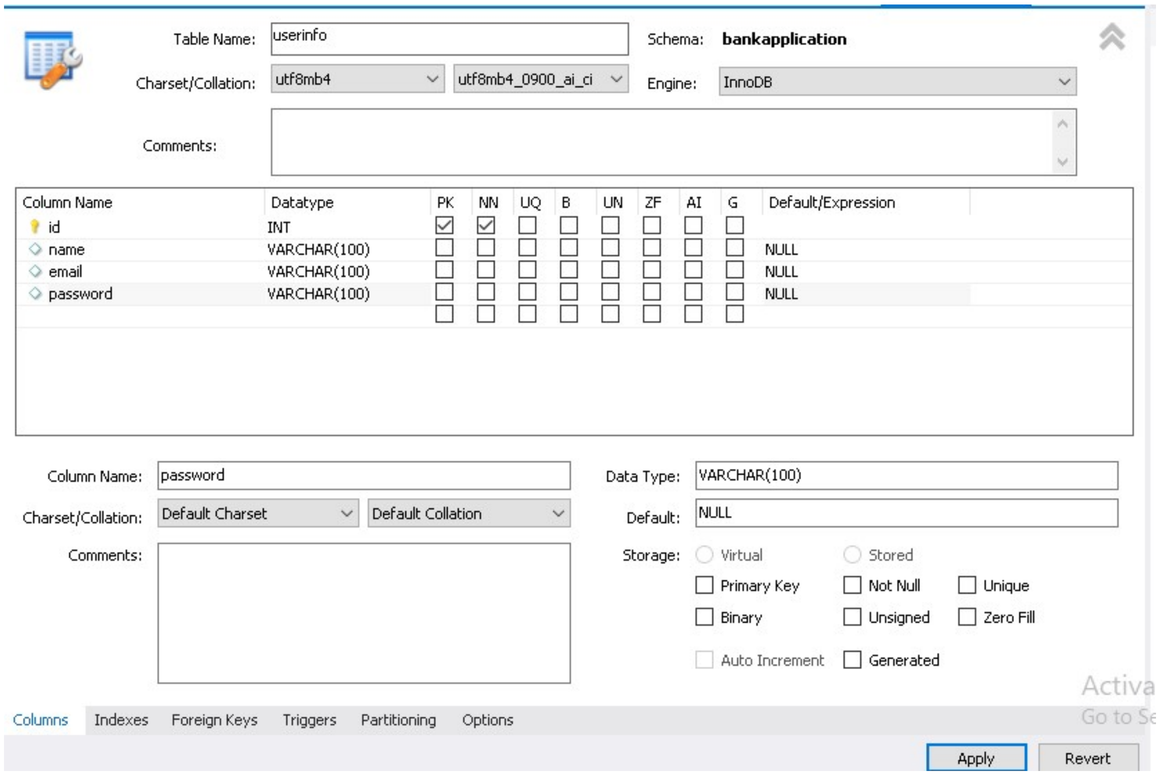

Table Name: userinfo

Schema: **bankapplication**

Charset/Collation: utf8mb4 | utf8mb4_0900_ai_ci

Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	✓	✓	☐	☐	☐	☐	☐	☐	
name	VARCHAR(100)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
email	VARCHAR(100)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
password	VARCHAR(100)	☐	☐	☐	☐	☐	☐	☐	☐	NULL

Column Name: password

Data Type: VARCHAR(100)

Charset/Collation: Default Charset | Default Collation

Default: NULL

Comments:

Storage: ○ Virtual ○ Stored

☐ Primary Key ☐ Not Null ☐ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

moneydetails:

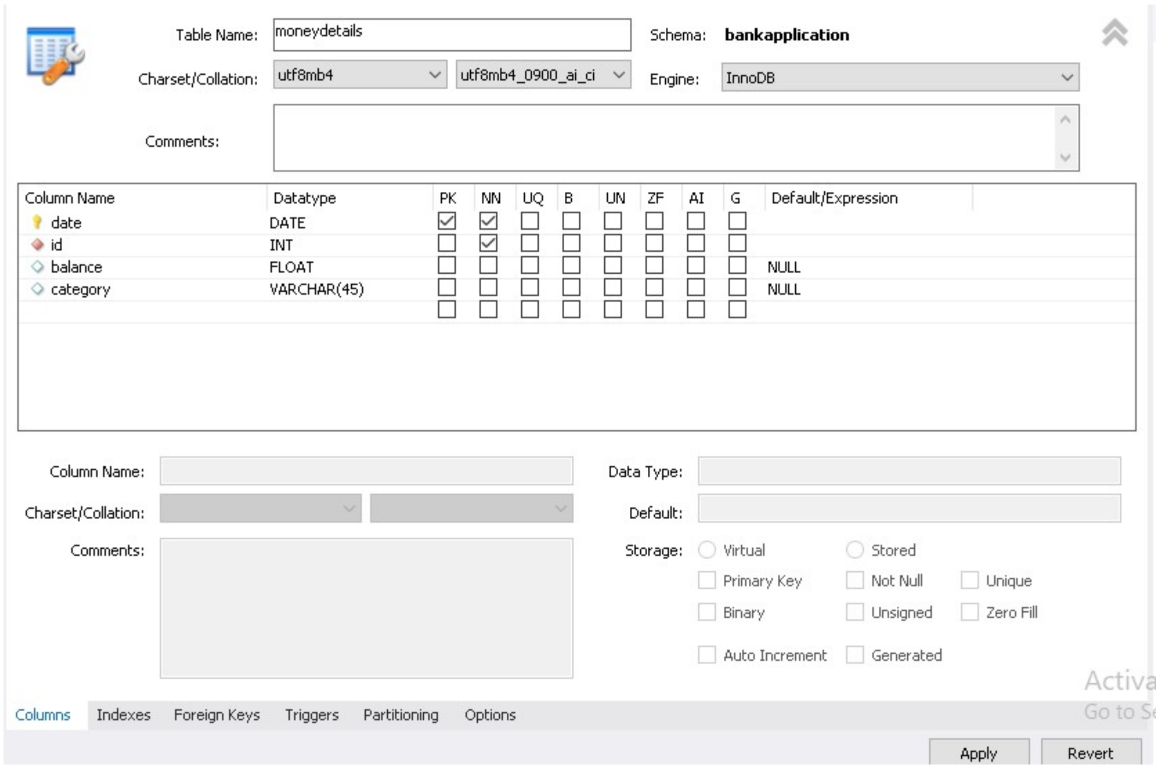

Table Name: moneydetails
Schema: **bankapplication**
Charset/Collation: utf8mb4 utf8mb4_0900_ai_ci
Engine: InnoDB
Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
date	DATE	✓	✓							
id	INT		✓							
balance	FLOAT									NULL
category	VARCHAR(45)									NULL

Column Name:
Data Type:
Charset/Collation:
Default:
Comments:
Storage: Virtual Stored
Primary Key Not Null Unique
Binary Unsigned Zero Fill
Auto Increment Generated