

# 프로젝트 4:

## DBMS 프로그램 개발

과목: Database Systems ( )

교수: 교수님

조교: 조교님, 조교님

소속:

학번:

이름: 송재현

## 목차

1. Environment
2. 프로그램 코드에 대한 자세한 설명
3. 수행되는 기능 스크린샷
4. 사용되는 SQL 문 명세
5. 변경사항

## 1. Environment

- OS: Windows 10
- Language: Java (openjdk, Amazon Corretto 17), MySQL 8.0
- IDE: IntelliJ IDEA 2020.3
- Encoding: UTF-8

## 2. 프로그램 코드에 대한 자세한 설명

Bank 프로그램은 은행을 모방해 만든 프로그램입니다.

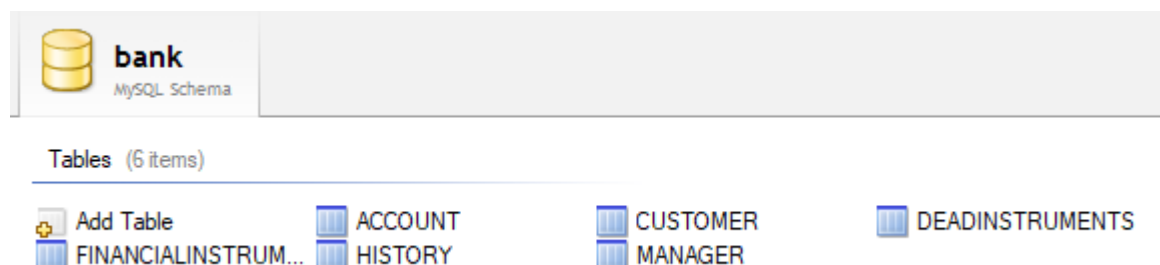
크게 main class 인 Bank class 와 Display class 로 나뉩니다.

Display class 는 주로 '길게 출력되는 선택지'를 화면에 출력할 때 사용됩니다.

Bank class 에서는 각종 함수를 이용해 다양한 기능을 구현합니다.

DB 스키마 설명 이후, 프로그램 구성을 설명하겠습니다.

SQL DB 에 대해 설명하겠습니다.



위 사진과 같이 총 6개의 테이블로 BANK DB 스키마를 구성했습니다.

MANAGER - Table										
Table Name: <b>MANAGER</b>		Schema: <b>bank</b>								
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Mssn	CHAR(5)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Mname	VARCHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Position	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
Password	VARCHAR(4)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

### Manager 테이블

Mssn 은 Primary Key 로 관리자들 개개인을 구분합니다. Mname 은 이름입니다. Position 은 0과 1이 있고, 0은 최고 관리자 1명을 의미하며, 다른 관리자들은 1의 Position 을 가집니다. Password 는 로그인, 고객의 고액 인출 등에 사용됩니다.

CUSTOMER - Table										
Table Name: <b>CUSTOMER</b>		Schema: <b>bank</b>								
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Cssn	CHAR(5)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Id	CHAR(13)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Cname	VARCHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Cphonenum	VARCHAR(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Credit	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	500
Cmssn	CHAR(5)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

### Customer 테이블

Cssn 은 Primary Key 로 고객들 개개인을 구분합니다. Id 는 주민번호를 의미합니다. Cname 은 이름을 의미합니다. Cphonenum 은 휴대전화번호를 의미합니다. Credit 은 신용점수로, 최대 계좌 수, 가입 가능 상품, 최대 인출 금액 제한, 담당 관리자 등에 영향을 끼칩니다. 기본 값은 500이고, 주기적으로 업데이트됩니다. Cmssn 은 담당 관리자를 의미하는 Foreign Key 입니다. 900점 이상의 신용점수를 가진 고객은 최고 관리자를 담당 관리자로 갖게 됩니다.

FINANCIALINSTRUMENTS - Table										
Table Name: <b>FINANCIALINSTRUMENTS</b>		Schema: <b>bank</b>								
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Instrumentcode	CHAR(3)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Category	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Standard	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Interest	DECIMAL(4,3)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

### Financial Instruments 테이블

Instrumentcode 는 Primary Key 로 각각의 상품을 구분합니다. Category 는 0과 1로 구성되는데, 0은 자유입출금상품을, 1은 대출상품을 의미합니다. 자유입출금상품은 잔고에 상관없이 입금, 출금이 가능합니다. 대출상품은 잔고가 0 이상이 되면 주기적 업데이트 시에 사라집니다.

DEADINSTRUMENTS - Table										
Table Name: DEADINSTRUMENTS		Schema: bank								
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Dinstrumentscode	CHAR(3)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

#### Dead Instruments 테이블

삭제한 은행 상품 테이블입니다. Dinstrumentscode 는 Financial Instruments 를 가리키는 Primary Key 이자 Foreign Key 입니다.

ACCOUNT - Table										
Table Name: ACCOUNT		Schema: bank								
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Ainstrumentscode	CHAR(3)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Accountnumber	CHAR(13)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Password	CHAR(4)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Acssn	CHAR(5)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Value	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

#### Account 테이블

Ainstrumentscode 는 identifying owner 를 의미하는 값으로 은행 상품을 가리킵니다. Accountnumber 은 계좌번호로 Ainstrumentscode 와 함께 Primary Key 를 구성합니다. Password 는 계좌의 비밀번호입니다. Acssn 은 계좌를 갖고 있는 고객을 의미하는 Foreign Key 입니다. Value 는 잔고를 의미합니다.

HISTORY - Table										
Table Name: HISTORY		Schema: bank								
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Haccountnumber	CHAR(13)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Hindex	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Hvaluechange	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

#### History 테이블

Haccountnumber 은 계좌를 가리키는 Foreign Key 입니다. Hindex 는 계좌기록들에 순서대로 붙인 인덱스 값입니다. 1부터 시작합니다. Hvaluechange 는 잔고 변경 값입니다.

프로그램 코드는 java 로 작성되었으며, 전체가 하나의 큰 while 문으로 구성되어 있고, 내부에 여러 while 문으로 각각의 기능들이 돌아갑니다. labeling 을 통해 필요한 반복문으로 jump 하는 내부 구조를 갖고 있습니다.

프로그램의 메뉴 흐름에 따라 설명하겠습니다.

1. 관리자: 관리자 로그인을 통해 최고관리자/일반관리자로 나뉩니다.
  - A. 최고관리자: Position 0으로 1명만 존재합니다.
    - i. 관리자 생성: Position1의 일반관리자들을 생성합니다.
    - ii. 관리자 정보 변경: 관리자의 정보를 변경합니다.
    - iii. 관리자 삭제: 관리자를 삭제합니다.
    - iv. 고객 생성: 고객을 생성합니다.
    - v. 은행 상품 생성: 은행 상품을 생성합니다.
    - vi. 은행 상품 삭제: 은행 상품을 삭제합니다. (DeadInstruments 테이블에 등록함으로써 신규 가입은 막고, 기존 상품 보유 고객은 그대로 이용할 수 있습니다.)
    - vii. 관리자 나열: 관리자 정보를 나열합니다.
    - viii. 고객 나열: 고객 정보를 나열합니다.
    - ix. 은행 상품 나열: 은행 상품 정보를 나열합니다.
    - x. 계좌 내역 관리: 최고 관리자의 고객의 계좌기록을 보고, 삭제할 수 있습니다.
  - B. 일반관리자
    - i. 계좌 내역 관리: 해당 관리자의 고객의 계좌기록을 보고, 삭제할 수 있습니다.
2. 사용자
  - A. 정보 변경: 고객의 정보를 변경합니다.
  - B. 계좌 생성: 계좌를 만듭니다. 계좌의 최대 개수는 (신용점수/100)개로 제한됩니다. 제한 조건에 해당하지 않으면, 상품 정보를 나열하고, 중복되지 않는 계좌번호로 계좌를 생성합니다.
  - C. 계좌 나열: 고객 자신의 계좌들의 정보를 나열합니다.
  - D. 입금: 계좌에 돈을 입금합니다. 계좌번호 입력 후 자신의 계좌인지 확인합니다.
  - E. 출금: 계좌에서 돈을 출금합니다. 계좌번호와, 계좌 비밀번호를 입력 후 본인 계좌가 아닐 시에 거부합니다. 금액이 자신의 신용점수보다 큰 값이라면, 담당 관리자의 비밀번호 입력이 필요합니다.
3. 한 달 후...
  - A. 이자, 신용 정보 업데이트 등의 작업을 수행합니다.

- i. 모든 계좌에 대해 이자를 잔고에 반영합니다. 이자는 잔고 \* 이율의 식으로 계산합니다. 기록이 남습니다.
- ii. 잔고가 0 이상이 된 대출 계좌들을 삭제합니다. 삭제된 계좌들을 화면에 출력합니다.
- iii. 신용점수를 업데이트합니다. 신용점수는 기존 신용점수 + 자유입출금계좌 잔고의 합(sum)/100의 식으로 계산합니다. 변경 내역을 화면에 출력합니다.
- iv. 신용점수가 900 이상인 고객들의 담당 관리자를 최고관리자로 변경합니다. 변경 내역을 화면에 출력합니다.

### 3. 수행되는 기능 스크린샷

컴파일 및 실행

```
C:\Users\송재현\Desktop\tmp>javac -encoding UTF-8 Bank.java

C:\Users\송재현\Desktop\tmp>java -cp ".;.\mysql-connector-java-8.0.27.jar" Bank

Initial Menu
0. Exit
1. Manager Menu(Log In)
2. Customer Menu(Log In)
3. One Month Later...
Input: 1

Manager LogIn
0. Go to Initial Menu
Insert your Ssn and Password one by one
0 or Ssn: 12345
Password: 1234
```

## 1. 최고관리자

A. 관리자 생성: Position1의 일반관리자들을 생성합니다.

	Mssn	Mname	Position	Password
▶	12345	Boss	0	1234
	14088	Pepe	1	1234
	23456	Rupi	1	1234
	42099	Sangdi	1	1234
	57922	Sancho	1	1234
	58475	Mina	1	1234
✱	NULL	NULL	NULL	NULL

기존 상태

```

Chief Manager Menu
0. Go to Initial Menu
1. Create Manager
2. Modify Manager
3. Delete Manager
4. New Customer Sign Up
5. Make Financial Instruments
6. Delete Financial Instruments
7. List Managers
8. List Customers
9. List Financial Instruments
10. Manage Account History
1

Create Manager
Set the new manager's ssn: 77777
Set the new manager's name:
aespa
Set the new manager's password:
1234
New Manager Has Been Added!

Chief Manager Menu
0. Go to Initial Menu
  
```

기능 실행 화면

	Mssn	Mname	Position	Password
▶	12345	Boss	0	1234
	14088	Pepe	1	1234
	23456	Rupi	1	1234
	42099	Sangdi	1	1234
	57922	Sancho	1	1234
	58475	Mina	1	1234
	77777	aespa	1	1234
✱	NULL	NULL	NULL	NULL

변경 상태



B. 관리자 정보 변경: 관리자의 정보를 변경합니다.

	Mssn	Mname	Position	Password
▶	12345	Boss	0	1234
	14088	Pepe	1	1234
	23456	Rupi	1	1234
	42099	Sangdi	1	1234
	57922	Sancho	1	1234
	58475	Mina	1	1234
	77777	aespa	1	1234
*	NULL	NULL	NULL	NULL

기존상태

```

Chief Manager Menu
0. Go to Initial Menu
1. Create Manager
2. Modify Manager
3. Delete Manager
4. New Customer Sign Up
5. Make Financial Instruments
6. Delete Financial Instruments
7. List Managers
8. List Customers
9. List Financial Instruments
10. Manage Account History
2

Modify Manager
Insert the manager's ssn: 77777
Set the manager's new name:
Exo
Set the manager's new password:
4321
Manager's Information Has Been Updated!

Chief Manager Menu
0. Go to Initial Menu
1. Create Manager
2. Modify Manager
  
```

기능 실행 화면

	Mssn	Mname	Position	Password
▶	12345	Boss	0	1234
	14088	Pepe	1	1234
	23456	Rupi	1	1234
	42099	Sangdi	1	1234
	57922	Sancho	1	1234
	58475	Mina	1	1234
	77777	Exo	1	4321
*	NULL	NULL	NULL	NULL

변경 상태

C. 관리자 삭제: 관리자를 삭제합니다.

	Mssn	Mname	Position	Password
▶	12345	Boss	0	1234
	14088	Pepe	1	1234
	23456	Rupi	1	1234
	42099	Sangdi	1	1234
	57922	Sancho	1	1234
	58475	Mina	1	1234
	77777	Exo	1	4321
*	NULL	NULL	NULL	NULL

기존상태

	Cssn	Id	Cname	Cphonenum	Credit	Cmssn
▶	13802	4612309281662	Ning	23957975831	555	57922
	14911	3411303825195	Wendy	39081084101	899	23456
	20591	5507138372922	Chen	48920481481	512	23456
	51098	5601147582912	Winter	01029394857	899	57922
	58723	5502133481922	Carina	43466267323	905	12345
	62039	9703223512795	Kai	60260249622	788	57922
	69038	8711303825195	Zizel	39571957915	450	23456
	92332	8303223512795	Jin	25235255266	114	23456
*	NULL	NULL	NULL	NULL	NULL	NULL

기존 상태

```

Chief Manager Menu
0. Go to Initial Menu
1. Create Manager
2. Modify Manager
3. Delete Manager
4. New Customer Sign Up
5. Make Financial Instruments
6. Delete Financial Instruments
7. List Managers
8. List Customers
9. List Financial Instruments
10. Manage Account History
3

Delete Manager
Insert the deleting manager's ssn:
23456
Insert the alternating manager's ssn:
58475
Manager '23456' Has Been Deleted!

Chief Manager Menu
0. Go to Initial Menu
1. Create Manager

```

기능 실행 화면

	Mssn	Mname	Position	Password
▶	12345	Boss	0	1234
	14088	Pepe	1	1234
	42099	Sangdi	1	1234
	57922	Sancho	1	1234
	58475	Mina	1	1234
	77777	Exo	1	4321
*	NULL	NULL	NULL	NULL

변경상태

	Cssn	Id	Cname	Cphonenum	Credit	Cmssn
▶	13802	4612309281662	Ning	23957975831	555	57922
	14911	3411303825195	Wendy	39081084101	899	58475
	20591	5507138372922	Chen	48920481481	512	58475
	51098	5601147582912	Winter	01029394857	899	57922
	58723	5502133481922	Carina	43466267323	905	12345
	62039	9703223512795	Kai	60260249622	788	57922
	69038	8711303825195	Zizel	39571957915	450	58475
	92332	8303223512795	Jin	25235255266	114	58475
*	NULL	NULL	NULL	NULL	NULL	NULL

변경 상태

D. 고객 생성: 고객을 생성합니다.

	Cssn	Id	Cname	Cphonenum	Credit	Cmssn
▶	13802	4612309281662	Ning	23957975831	555	57922
	14911	3411303825195	Wendy	39081084101	899	58475
	20591	5507138372922	Chen	48920481481	512	58475
	51098	5601147582912	Winter	01029394857	899	57922
	58723	5502133481922	Carina	43466267323	905	12345
	62039	9703223512795	Kai	60260249622	788	57922
	69038	8711303825195	Zizel	39571957915	450	58475
	92332	8303223512795	Jin	25235255266	114	58475
•	NULL	NULL	NULL	NULL	NULL	NULL

기존 상태

```

Chief Manager Menu
0. Go to Initial Menu
1. Create Manager
2. Modify Manager
3. Delete Manager
4. New Customer Sign Up
5. Make Financial Instruments
6. Delete Financial Instruments
7. List Managers
8. List Customers
9. List Financial Instruments
10. Manage Account History
4

New Customer Sign Up
Set the customer's ssn: 53535
Insert the customer's ID:
8102126749815
Insert the customer's name:
Koon
Insert the customer's phone number:
07084858485
Set the customer's manager's ssn:
77777
New Customer Has Been Added!

Chief Manager Menu
0. Go to Initial Menu

```

기능 실행 화면

	Cssn	Id	Cname	Cphonenum	Credit	Cmssn
▶	13802	4612309281662	Ning	23957975831	555	57922
	14911	3411303825195	Wendy	39081084101	899	58475
	20591	5507138372922	Chen	48920481481	512	58475
	51098	5601147582912	Winter	01029394857	899	57922
	53535	8102126749815	Koon	07084858485	500	77777
	58723	5502133481922	Carina	43466267323	905	12345
	62039	9703223512795	Kai	60260249622	788	57922
	69038	8711303825195	Zizel	39571957915	450	58475
	92332	8303223512795	Jin	25235255266	114	58475
★	NULL	NULL	NULL	NULL	NULL	NULL

변경 상태

(Credit default: 500)

E. 은행 상품 생성: 은행 상품을 생성합니다.

	Instrumentscode	Category	Standard	Interest
▶	123	0	999	0.099
	366	1	999	0.005
	386	1	0	0.563
	563	0	500	0.055
	687	1	500	0.067
	864	0	0	0.021
*	NULL	NULL	NULL	NULL

기존 상태

```

Chief Manager Menu
0. Go to Initial Menu
1. Create Manager
2. Modify Manager
3. Delete Manager
4. New Customer Sign Up
5. Make Financial Instruments
6. Delete Financial Instruments
7. List Managers
8. List Customers
9. List Financial Instruments
10. Manage Account History
5

Make Financial Instruments
Insert the new instrument's code: 666
Insert the new instrument's category:
0
Insert the new instrument's credit standard:
450
Insert the new instrument's interest:
0.049
New Financial Instrument Has Been Added!

Chief Manager Menu
  
```

기능 실행 화면

	Instrumentscode	Category	Standard	Interest
▶	123	0	999	0.099
	366	1	999	0.005
	386	1	0	0.563
	563	0	500	0.055
	666	0	450	0.049
	687	1	500	0.067
	864	0	0	0.021
*	NULL	NULL	NULL	NULL

변경 상태

F. 은행 상품 삭제: 은행 상품을 삭제합니다. (DeadInstruments 테이블에 등록함으로써 신규 가입은 막고, 기존 상품 보유 고객은 그대로 이용할 수 있습니다.)

	Dinstrumentscode
▶	563
•	HULL

기존 상태

```
Chief Manager Menu
0. Go to Initial Menu
1. Create Manager
2. Modify Manager
3. Delete Manager
4. New Customer Sign Up
5. Make Financial Instruments
6. Delete Financial Instruments
7. List Managers
8. List Customers
9. List Financial Instruments
10. Manage Account History
6

Delete Financial Instruments
Insert the instrument's code: 666
Financial Instrument 666 Has Been Added!

Chief Manager Menu
0. Go to Initial Menu
```

기능 실행 화면

	Dinstrumentscode
▶	563
	666
•	HULL

변경 상태

G. 관리자 나열: 관리자 정보를 나열합니다.

	Mssn	Mname	Position	Password
▶	12345	Boss	0	1234
	14088	Pepe	1	1234
	42099	Sangdi	1	1234
	57922	Sancho	1	1234
	58475	Mina	1	1234
	77777	Exo	1	4321
★	NULL	NULL	NULL	NULL

기존상태

```

Chief Manager Menu
0. Go to Initial Menu
1. Create Manager
2. Modify Manager
3. Delete Manager
4. New Customer Sign Up
5. Make Financial Instruments
6. Delete Financial Instruments
7. List Managers
8. List Customers
9. List Financial Instruments
10. Manage Account History
7

List Managers
Index Ssn      Name Position
1 12345      Boss 0
2 77777      Exo 1
3 58475      Mina 1
4 14088      Pepe 1
5 57922      Sancho 1
6 42099      Sangdi 1

Chief Manager Menu
0. Go to Initial Menu

```

기능 실행 화면(이름 오름차순)



H. 고객 나열: 고객 정보를 나열합니다.

	Cssn	Id	Cname	Cphonenum	Credit	Cmssn
▶	13802	4612309281662	Ning	23957975831	555	57922
	14911	3411303825195	Wendy	39081084101	899	58475
	20591	5507138372922	Chen	48920481481	512	58475
	51098	5601147582912	Winter	01029394857	899	57922
	53535	8102126749815	Koon	07084858485	500	77777
	58723	5502133481922	Carina	43466267323	905	12345
	62039	9703223512795	Kai	60260249622	788	57922
	69038	8711303825195	Zizel	39571957915	450	58475
	92332	8303223512795	Jin	25235255266	114	58475
*	NULL	NULL	NULL	NULL	NULL	NULL

기존상태

```

Chief Manager Menu
0. Go to Initial Menu
1. Create Manager
2. Modify Manager
3. Delete Manager
4. New Customer Sign Up
5. Make Financial Instruments
6. Delete Financial Instruments
7. List Managers
8. List Customers
9. List Financial Instruments
10. Manage Account History
8

List Customers
Index Ssn      Name Phonenum Credit Mng_ssn Mng_name
1 58723      Carina 43466267323 905 12345      Boss
2 14911      Wendy 39081084101 899 58475      Mina
3 51098      Winter 01029394857 899 57922      Sancho
4 62039      Kai 60260249622 788 57922      Sancho
5 13802      Ning 23957975831 555 57922      Sancho
6 20591      Chen 48920481481 512 58475      Mina
7 53535      Koon 07084858485 500 77777      Exo
8 69038      Zizel 39571957915 450 58475      Mina
9 92332      Jin 25235255266 114 58475      Mina

Chief Manager Menu

```

기능 실행 화면 (Credit 내림차순)

- I. 은행 상품 나열: 은행 상품 정보를 나열합니다.

	Instrumentscode	Category	Standard	Interest
▶	123	0	999	0.099
	366	1	999	0.005
	386	1	0	0.563
	563	0	500	0.055
	666	0	450	0.049
	687	1	500	0.067
	864	0	0	0.021
*	NULL	NULL	NULL	NULL

기존 상태

	Dinstrumentscode
▶	563
	666
*	NULL

기존상태

```

Chief Manager Menu
0. Go to Initial Menu
1. Create Manager
2. Modify Manager
3. Delete Manager
4. New Customer Sign Up
5. Make Financial Instruments
6. Delete Financial Instruments
7. List Managers
8. List Customers
9. List Financial Instruments
10. Manage Account History
9

List Financial Instruments
Index InstrumentsCode Category Standard Interest
1 123 Check 999 0.099
2 864 Check 0 0.021
3 366 Loan 999 0.005
4 687 Loan 500 0.067
5 386 Loan 0 0.563
List Dead Instruments
Index InstrumentsCode Category Standard Interest
1 563 Check 500 0.055
2 666 Check 450 0.049

```

기능 실행 화면 (삭제한 은행 상품은 따로 나열)

J. 계좌 내역 관리: 최고 관리자의 고객의 계좌기록을 보고, 삭제할 수 있습니다.

	Haccountnumber	Hindex	Hvaluechange
▶	4937201947271	6	-2311
	7489147294477	2	2321
	7849734718171	1	50
*	NULL	NULL	NULL

기존 상태

```

Chief Manager Menu
0. Go to Initial Menu
1. Create Manager
2. Modify Manager
3. Delete Manager
4. New Customer Sign Up
5. Make Financial Instruments
6. Delete Financial Instruments
7. List Managers
8. List Customers
9. List Financial Instruments
10. Manage Account History
10

Manage Account History
Index Category Customer AccountNumber Manager
1 Check 58723 4937201947271 12345
2 Check 58723 5310589038559 12345
3 Loan 58723 6408649268026 12345
Which account would you like to manage?(put 0 to cancel): 4937201947271
HistoryIndex AccountNumber ValueChange
        6 4937201947271
Which index would you remove?: 6
History Has Been Removed!

Chief Manager Menu
0. Go to Initial Menu
1. Create Manager

```

기능 실행 화면

	Haccountnumber	Hindex	Hvaluechange
▶	7489147294477	2	2321
	7849734718171	1	50
*	NULL	NULL	NULL

변경 상태

## 2. 일반관리자

A. 계좌 내역 관리: 해당 관리자의 고객의 계좌기록을 보고, 삭제할 수 있습니다.

	Haccountnumber	Hindex	Hvaluechange
▶	7489147294477	2	2321
	7849734718171	1	50
✱	NULL	NULL	NULL

기존 상태

```
Manager LogIn
0. Go to Initial Menu
Insert your Ssn and Password one by one
0 or Ssn: 57922
Password: 1234

Manager Menu ( Mssn: 57922 Position: 1 )
0. Go to Initial Menu
1. Manage Account History
1

Manage Account History
Index Category Customer AccountNumber Manager
1 Check 13802 4968240680924 57922
2 Loan 13802 7849734718171 57922
3 Check 51098 4096804268266 57922
4 Loan 51098 0194021941294 57922
5 Check 62039 4756295724755 57922
6 Loan 62039 1788468468888 57922
Which account would you like to manage?(put 0 to cancel): 7489147294477
HistoryIndex AccountNumber ValueChange
2 7489147294477
Which index would you remove?(0 to cancel): 2
History Has Been Removed!

Manager Menu ( Mssn: 57922 Position: 1 )
0. Go to Initial Menu
1. Manage Account History
|
```

기능 실행 화면

	Haccountnumber	Hindex	Hvaluechange
▶	7849734718171	1	50
✱	NULL	NULL	NULL

변경 상태

### 3. 사용자

A. 정보 변경: 고객의 정보를 변경합니다.

	Cssn	Id	Cname	Cphonenum	Credit	Cmssn
▶	13802	4612309281662	Ning	23957975831	555	57922
	14911	3411303825195	Wendy	39081084101	899	58475
	20591	5507138372922	Chen	48920481481	512	58475
	51098	5601147582912	Winter	01029394857	899	57922
	53535	8102126749815	Koon	07084858485	500	77777
	58723	5502133481922	Carina	43466267323	905	12345
	62039	9703223512795	Kai	60260249622	788	57922
	69038	8711303825195	Zizel	39571957915	450	58475
	92332	8303223512795	Jin	25235255266	114	58475
*	NULL	NULL	NULL	NULL	NULL	NULL

기존 상태

```
Customer LogIn
0. Go to Initial Menu
Insert your Ssn
0 or Ssn: 92332

Customer Menu
0. Go to Initial Menu
1. Modify Information
2. Create Account
3. List My Account
4. Save
5. Withdraw
Choice: 1

Modify Information
Insert the customer's new name:
Jimin
Insert the customer's new phone number:
00011112222
Insert the customer's new manager's ssn:
77777
Customer's Information Has Been Modified!

Customer Menu
0. Go to Initial Menu
1. Modify Information
2. Create Account
3. List My Account
4. Save
5. Withdraw
Choice: |
```

기능 실행 화면

	Cssn	Id	Cname	Cphonenum	Credit	Cmssn
▶	13802	4612309281662	Ning	23957975831	555	57922
	14911	3411303825195	Wendy	39081084101	899	58475
	20591	5507138372922	Chen	48920481481	512	58475
	51098	5601147582912	Winter	01029394857	899	57922
	53535	8102126749815	Koon	07084858485	500	77777
	58723	5502133481922	Carina	43466267323	905	12345
	62039	9703223512795	Kai	60260249622	788	57922
	69038	8711303825195	Zizel	39571957915	450	58475
	92332	8303223512795	Jimin	00011112222	114	77777
✱	NULL	NULL	NULL	NULL	NULL	NULL

변경 상태

- B. 계좌 생성: 계좌를 만듭니다. 계좌의 최대 개수는 (신용점수/100)개로 제한됩니다. 제한 조건에 해당하지 않으면, 상품 정보를 나열하고, 중복되지 않는 계좌번호로 계좌를 생성합니다.

	Ainstrumentscode	Accountnumber	Password	Acssn	Value
	563	3513515353444	1234	14911	4000
	563	4096804268266	1234	51098	3599
	563	4756295724755	1234	62039	1822
	563	4937201947271	1234	58723	6162
	563	4968240680924	1234	13802	2463
	563	5310589038559	1234	58723	6484
▶	563	8467926748296	1234	20591	3522
	687	0194021941294	1234	51098	-132
	687	1788468468888	1234	62039	15
	687	6408649268026	1234	58723	-511
	687	7849734718171	1234	13802	20
	864	1670829682282	1234	69038	6642
	864	5636537482828	1234	92332	100
	864	6702769427629	1234	69038	7675
	864	7489147294477	1234	69038	6622
*	NULL	NULL	NULL	NULL	NULL

기존 상태

```

Customer Menu
0. Go to Initial Menu
1. Modify Information
2. Create Account
3. List My Account
4. Save
5. Withdraw
Choice: 2

Create Account
You Have Too Many Accounts!

Customer Menu
0. Go to Initial Menu

```

Jimin 은 개수 제한으로 계좌 생성 불가

```
Create Account

Create Account Menu
0. Go to Initial Menu
1. Cancel
2. Cat0: Check
3. Cat1: Loan
Choice: 2
Index InstrumentsCode Category Standard Interest
1 123 Check 999 0.099
2 864 Check 0 0.021
3 366 Loan 999 0.005
4 687 Loan 500 0.067
5 386 Loan 0 0.563
Choose InstrumentsCode Choice: 123
The Standard Is Above Your Credit!

Customer Menu
```

Wendy는 Credit 부족으로 123번 은행상품 가입 불가



```

Create Account

Create Account Menu
0. Go to Initial Menu
1. Cancel
2. Cat0: Check
3. Cat1: Loan
Choice: 2
Index InstrumentsCode Category Standard Interest
1 123 Check 999 0.099
2 864 Check 0 0.021
3 366 Loan 999 0.005
4 687 Loan 500 0.067
5 386 Loan 0 0.563
Choose InstrumentsCode Choice: 864
Set Account Number: 8467926748296
Existing Account Number!
Set Account Number: 6487154879635
Set the new account's password (4 Numbers / ex.1234): 4321
Account Created!

Customer Menu
0. Go to Initial Menu
1. Modify Information
2. Create Account
3. List My Account
4. Save
5. Withdraw
Choice: |

```

존재하는 계좌번호는 계좌생성불가. 없는 계좌번호로 계좌 생성 완료.

	Ainstrumentscode	Accountnumber	Password	Acssn	Value
	563	3513515353444	1234	14911	4000
	563	4096804268266	1234	51098	3599
	563	4756295724755	1234	62039	1822
	563	4937201947271	1234	58723	6162
	563	4968240680924	1234	13802	2463
	563	5310589038559	1234	58723	6484
▶	563	8467926748296	1234	20591	3522
	687	0194021941294	1234	51098	-132
	687	1788468468888	1234	62039	15
	687	6408649268026	1234	58723	-511
	687	7849734718171	1234	13802	20
	864	1670829682282	1234	69038	6642
	864	5636537482828	1234	92332	100
	864	6487154879635	4321	14911	0
	864	6702769427629	1234	69038	7675
	864	7489147294477	1234	69038	6622
★	NULL	NULL	NULL	NULL	NULL

변경 상태

C. 계좌 나열: 고객 자신의 계좌들의 정보를 나열합니다.

```
Customer Menu
0. Go to Initial Menu
1. Modify Information
2. Create Account
3. List My Account
4. Save
5. Withdraw
Choice: 3

List My Account
Index Category AccountNumber Value Interest Manager
1 Check 3513515353444 4000 0.055 58475
2 Check 6487154879635 0 0.021 58475

Customer Menu
```

기능 실행 화면

D. 입금: 계좌에 돈을 입금합니다. 계좌번호 입력 후 자신의 계좌인지 확인합니다.

	Ainstrumentscode	Accountnumber	Password	Acssn	Value
	563	3513515353444	1234	14911	4000
	563	4096804268266	1234	51098	3599
	563	4756295724755	1234	62039	1822
	563	4937201947271	1234	58723	6162
	563	4968240680924	1234	13802	2463
	563	5310589038559	1234	58723	6484
▶	563	8467926748296	1234	20591	3522
	687	0194021941294	1234	51098	-132
	687	1788468468888	1234	62039	15
	687	6408649268026	1234	58723	-511
	687	7849734718171	1234	13802	20
	864	1670829682282	1234	69038	6642
	864	5636537482828	1234	92332	100
	864	6487154879635	4321	14911	0
	864	6702769427629	1234	69038	7675
	864	7489147294477	1234	69038	6622
*	NULL	NULL	NULL	NULL	NULL

기존 상태

```

Customer Menu
0. Go to Initial Menu
1. Modify Information
2. Create Account
3. List My Account
4. Save
5. Withdraw
Choice: 4

Save
Choose The Account: 15487
Not Your Account!
  
```

타인 계좌 입력 시 거부

```

Customer Menu
0. Go to Initial Menu
1. Modify Information
2. Create Account
3. List My Account
4. Save
5. Withdraw
Choice: 4

Save
Choose The Account: 3513515353444
How Much Money Would You Save?: 1000
Saving Is Done

```

입금 성공

	Ainstrumentscode	Accountnumber	Password	Acssn	Value
	563	3513515353444	1234	14911	5000
	563	4096804268266	1234	51098	3599
	563	4756295724755	1234	62039	1822
	563	4937201947271	1234	58723	6162
	563	4968240680924	1234	13802	2463
	563	5310589038559	1234	58723	6484
▶	563	8467926748296	1234	20591	3522
	687	0194021941294	1234	51098	-132
	687	1788468468888	1234	62039	15
	687	6408649268026	1234	58723	-511
	687	7849734718171	1234	13802	20
	864	1670829682282	1234	69038	6642
	864	5636537482828	1234	92332	100
	864	6487154879635	4321	14911	0
	864	6702769427629	1234	69038	7675
	864	7489147294477	1234	69038	6622
•	NULL	NULL	NULL	NULL	NULL

변경 상태

- E. 출금: 계좌에서 돈을 출금합니다. 계좌번호와, 계좌 비밀번호를 입력 후 본인 계좌가 아닐 시에 거부합니다. 금액이 자신의 신용점수보다 큰 값이라면, 담당 관리자의 비밀번호 입력이 필요합니다.

```
Customer LogIn
0. Go to Initial Menu
Insert your Ssn
0 or Ssn: 14911

Customer Menu
0. Go to Initial Menu
1. Modify Information
2. Create Account
3. List My Account
4. Save
5. Withdraw
Choice: 5
Withdraw
Choose The Account: 4096804268266
Insert The Password: 1234
Not Your Account!
```

타인 계좌 인출 거부

```
Customer Menu
0. Go to Initial Menu
1. Modify Information
2. Create Account
3. List My Account
4. Save
5. Withdraw
Choice: 5
Withdraw
Choose The Account: 3513515353444
Insert The Password: 1234
How Much Money Would You Withdraw?: 1000
The Amount Is Above Your Limit.
Insert Your Manager's Password1234
Withdrawing Is Done

Customer Menu
```

큰 금액 인출 시 관리자 비밀번호 입력 후 출금 성공

```

Customer Menu
0. Go to Initial Menu
1. Modify Information
2. Create Account
3. List My Account
4. Save
5. Withdraw
Choice: 5
Withdraw
Choose The Account: 3513515353444
Insert The Password: 1234
How Much Money Would You Withdraw?: 100
Withdrawing Is Done

Customer Menu

```

소액 인출 시 출금 성공

	Haccountnumber	Hindex	Hvaluechange
	3513515353444	4	-100
▶	3513515353444	5	1100
	3513515353444	6	-1000
	3513515353444	7	-100
	7849734718171	1	50
•	NULL	NULL	NULL

관련 계좌기록 생성

#### 4. 한 달 후...

```
Initial Menu
0. Exit
1. Manager Menu(Log In)
2. Customer Menu(Log In)
3. One Month Later...
Input: 3

You Chose 'One Month Later...'
Interests Calculation...
Removing Fully Paid Loan Accounts
Update: Account With Accountnumber 1788468468888 Has Been Removed
Update: Account With Accountnumber 7849734718171 Has Been Removed
Recalculating The Credit
Update: 14911's Credit Has Changed From 899 to 941
Update: 51098's Credit Has Changed From 899 to 936
Update: 62039's Credit Has Changed From 788 to 807
Update: 58723's Credit Has Changed From 905 to 1038
Update: 13802's Credit Has Changed From 555 to 580
Update: 20591's Credit Has Changed From 512 to 549
Update: 69038's Credit Has Changed From 450 to 663
Update: 92332's Credit Has Changed From 114 to 115
Credit Above 900 Customers' Will Be Belonged To Chief Manager
Update: Wendy's Manager Is The Chief Manager
Update: Winter's Manager Is The Chief Manager
... Complete

Initial Menu
0. Exit
1. Manager Menu(Log In)
2. Customer Menu(Log In)
3. One Month Later...
Input: |
```

기능 실행 화면

- A. 모든 계좌에 대해 이자를 잔고에 반영합니다. 이자는 잔고 \* 비율의 식으로 계산합니다.  
기록이 남습니다.

	Ainstrumentscode	Accountnumber	Password	Acssn	Value
▶	563	3513515353444	1234	14911	4000
	563	4096804268266	1234	51098	3599
	563	4756295724755	1234	62039	1822
	563	4937201947271	1234	58723	6162
	563	4968240680924	1234	13802	2463
	563	5310589038559	1234	58723	6484
	563	8467926748296	1234	20591	3522
	687	0194021941294	1234	51098	-132
	687	1788468468888	1234	62039	15
	687	6408649268026	1234	58723	-511
	687	7849734718171	1234	13802	20
	864	1670829682282	1234	69038	6642
	864	5636537482828	1234	92332	100
	864	6702769427629	1234	69038	7675
	864	7489147294477	1234	69038	6622
★	NULL	NULL	NULL	NULL	NULL

기존 상태

	Haccountnumber	Hindex	Hvaluechange
▶	4937201947271	6	-2311
	7489147294477	2	2321
	7849734718171	1	50
★	NULL	NULL	NULL

기존 상태

	Ainstrumentscode	Accountnumber	Password	Acssn	Value
▶	563	3513515353444	1234	14911	4220
	563	4096804268266	1234	51098	3797
	563	4756295724755	1234	62039	1922
	563	4937201947271	1234	58723	6501
	563	4968240680924	1234	13802	2598
	563	5310589038559	1234	58723	6841
	563	8467926748296	1234	20591	3716
	687	0194021941294	1234	51098	-141
	687	6408649268026	1234	58723	-545
	864	1670829682282	1234	69038	6781
	864	5636537482828	1234	92332	102
	864	6702769427629	1234	69038	7836
	864	7489147294477	1234	69038	6761
★	NULL	NULL	NULL	NULL	NULL

변경 상태



	Haccountnumber	Hindex	Hvaluechange
▶	7489147294477	2	2321
	4937201947271	6	-2311
	3513515353444	7	220
	4096804268266	8	198
	4756295724755	9	100
	4937201947271	10	339
	4968240680924	11	135
	5310589038559	12	357
	8467926748296	13	194
	0194021941294	14	-9
	6408649268026	16	-34
	1670829682282	18	139
	5636537482828	19	2
	6702769427629	20	161
	7489147294477	21	139
★	NULL	NULL	NULL

변경 상태

B. 잔고가 0 이상이 된 대출 계좌들을 삭제합니다. 삭제된 계좌들을 화면에 출력합니다.

	Ainstrumentscode	Accountnumber	Password	Acssn	Value
▶	563	3513515353444	1234	14911	4000
	563	4096804268266	1234	51098	3599
	563	4756295724755	1234	62039	1822
	563	4937201947271	1234	58723	6162
	563	4968240680924	1234	13802	2463
	563	5310589038559	1234	58723	6484
	563	8467926748296	1234	20591	3522
	687	0194021941294	1234	51098	-132
	687	1788468468888	1234	62039	15
	687	6408649268026	1234	58723	-511
	687	7849734718171	1234	13802	20
	864	1670829682282	1234	69038	6642
	864	5636537482828	1234	92332	100
	864	6702769427629	1234	69038	7675
	864	7489147294477	1234	69038	6622
★	NULL	NULL	NULL	NULL	NULL

기존 상태

	Ainstrumentscode	Accountnumber	Password	Acssn	Value
▶	563	3513515353444	1234	14911	4220
	563	4096804268266	1234	51098	3797
	563	4756295724755	1234	62039	1922
	563	4937201947271	1234	58723	6501
	563	4968240680924	1234	13802	2598
	563	5310589038559	1234	58723	6841
	563	8467926748296	1234	20591	3716
	687	0194021941294	1234	51098	-141
	687	6408649268026	1234	58723	-545
	864	1670829682282	1234	69038	6781
	864	5636537482828	1234	92332	102
	864	6702769427629	1234	69038	7836
	864	7489147294477	1234	69038	6761
*	NULL	NULL	NULL	NULL	NULL

변경 상태

- C. 신용점수를 업데이트합니다. 신용점수는 기존 신용점수 + 자유입출금계좌 잔고의 합(sum)/100의 식으로 계산합니다.

	Cssn	Id	Cname	Cphonenum	Credit	Cmssn
▶	13802	4612309281662	Ning	23957975831	555	57922
	14911	3411303825195	Wendy	39081084101	899	58475
	20591	5507138372922	Chen	48920481481	512	58475
	51098	5601147582912	Winter	01029394857	899	57922
	53535	8102126749815	Koon	07084858485	500	77777
	58723	5502133481922	Carina	43466267323	905	12345
	62039	9703223512795	Kai	60260249622	788	57922
	69038	8711303825195	Zizel	39571957915	450	58475
	92332	8303223512795	Jimin	00011112222	114	77777
*	NULL	NULL	NULL	NULL	NULL	NULL

기존 상태

	Cssn	Id	Cname	Cphonenum	Credit	Cmssn
▶	13802	4612309281662	Ning	23957975831	580	57922
	14911	3411303825195	Wendy	39081084101	941	12345
	20591	5507138372922	Chen	48920481481	549	23456
	51098	5601147582912	Winter	01029394857	936	12345
	58723	5502133481922	Carina	43466267323	1038	12345
	62039	9703223512795	Kai	60260249622	807	57922
	69038	8711303825195	Zizel	39571957915	663	23456
	92332	8303223512795	Jin	25235255266	115	23456
*	NULL	NULL	NULL	NULL	NULL	NULL

변경 상태

- D. 신용점수가 900 이상인 고객들의 담당 관리자를 최고관리자로 변경합니다. 변경 내역을 화면에 출력합니다.

	Cssn	Id	Cname	Cphonenum	Credit	Cmssn
▶	13802	4612309281662	Ning	23957975831	555	57922
	14911	3411303825195	Wendy	39081084101	899	23456
	20591	5507138372922	Chen	48920481481	512	23456
	51098	5601147582912	Winter	01029394857	899	57922
	58723	5502133481922	Carina	43466267323	905	12345
	62039	9703223512795	Kai	60260249622	788	57922
	69038	8711303825195	Zizel	39571957915	450	23456
	92332	8303223512795	Jin	25235255266	114	23456
*	NULL	NULL	NULL	NULL	NULL	NULL

기존 상태

	Cssn	Id	Cname	Cphonenum	Credit	Cmssn
▶	13802	4612309281662	Ning	23957975831	580	57922
	14911	3411303825195	Wendy	39081084101	941	12345
	20591	5507138372922	Chen	48920481481	549	23456
	51098	5601147582912	Winter	01029394857	936	12345
	58723	5502133481922	Carina	43466267323	1038	12345
	62039	9703223512795	Kai	60260249622	807	57922
	69038	8711303825195	Zizel	39571957915	663	23456
	92332	8303223512795	Jin	25235255266	115	23456
*	NULL	NULL	NULL	NULL	NULL	NULL

변경 상태

#### 4. 사용되는 SQL 문 명세

1. Hindex 는 History 테이블의 Primary Key 를 구성하는 attribute 중 하나입니다. DB 가동 시에 가장 높은 계좌 기록 인덱스를 찾아 그 다음 넘버링을 저장합니다.  
`rs = stmt.executeQuery("select Hindex from history");`
2. // Manager 테이블에서 Mssn, Password, Position attribute 를 가져옵니다.  
`rs = stmt.executeQuery("select Mssn, Password, Position from Manager");`
3. // 필요한 value 들을 넣어 새 Manager tuple 을 생성합니다.  
`sql = String.format("insert into manager values ('%s', '%s', 1, '%s') ",  
inputStr, inputStr1, inputStr2);`
4. // 필요한 value 들을 넣어 Manager tuple 을 수정합니다.  
`sql = String.format("update manager set Mname = '%s', Password = '%s' where  
Mssn = '%s'", inputStr1, inputStr2, inputStr);`
5. // 관리자를 삭제하려고 합니다. 담당 고객들의 관리자 정보를 업데이트하기 위해  
Manager, Customer 테이블을 equijoin 해 정보를 가져옵니다.  
`sql = String.format("select Ccssn from customer, manager where Ccssn = Mssn");`
6. // Referential Constraint 위배 방지를 위해, 삭제할 Manager 의 Customer 들의  
Manager 를 다른 Manager 로 변경합니다.  
`sql = String.format("update customer set Ccssn = '%s' where Ccssn = '%s'",  
inputStr1, inputStr);`
7. // 삭제하고자하는 Manager tuple 을 DB 에서 삭제합니다.  
`sql = String.format("delete from manager where Mssn = '%s'", inputStr);`
8. // 필요한 value 들을 넣어 Customer tuple 을 생성합니다. Default Credit 은 500  
입니다.  
`sql = String.format("insert into customer values ('%s', '%s', '%s', '%s',  
DEFAULT, '%s')", inputStr, inputStr1, inputStr2, inputStr3, inputStr4);`
9. // 필요한 value 들을 넣어 Financial Instruments tuple 을 생성합니다.  
`sql = String.format("insert into financialinstruments values ('%s', '%s',  
'%s', '%s') ", inputStr, input, input1, inputdbl);`
10. // 삭제한 Financial Instrument 의 Primary Key 인 Instrumentscode 를 Dead  
Instruments 테이블에 추가합니다.  
`sql = String.format("insert into deadinstruments values ('%s')", inputStr);`
11. // 관리자들 나열해 출력하려고 합니다. Manager 테이블에서 Position, 이름에 따라  
정렬해 정보를 가져옵니다.  
`rs = stmt.executeQuery("select Mssn, Mname, Position from Manager order by  
Position asc, Mname asc");`
12. // 전체 고객들을 나열해 출력하려고 합니다. Manager 와 담당하는 Customer 들의  
정보를 equijoin 을 통해 연결하고, Credit 내림차순으로 정렬해 정보를  
가져옵니다.  
`rs = stmt.executeQuery("select Ccssn, Cname, Cphonenumber, Credit, Mssn, Mname  
from Manager join Customer on manager.Mssn = customer.Ccssn order by Credit  
desc ");`

13. // 은행 상품들을 출력하려고 합니다. (Not In)을 통해 삭제되어 Dead Instruments 에 등록된 Financial Instruments 들을 제외합니다.  
Category(자유입출금/대출)와 가입가능 신용점수 기준으로 정렬합니다.  
rs = stmt.executeQuery("select Instrumentscode, Category, Standard, Interest  
from financialinstruments where Instrumentscode NOT IN (select  
Dinstrumentscode from deadinstruments where Dinstrumentscode =  
Instrumentscode) order by Category asc, Standard desc");
14. // 삭제되어 Dead Instruments 에 등록된 은행 상품들도 따로 출력하려고 합니다.  
DeadInstruments 테이블의 tuple 들을 정렬해 가져옵니다.  
rs = stmt.executeQuery("select Instrumentscode, Category, Standard, Interest  
from financialinstruments join deadinstruments where Instrumentscode =  
Dinstrumentscode order by Category asc, Standard desc");
15. // 매니저가 담당하는 고객들의 계좌 정보를 출력하려고 합니다. Account,  
Customer, FinancialInstruments 테이블에서 정보를 equijoin 합니다.  
sql = String.format("select Category, Accountnumber, Ccssn, Cmsn from account,  
customer, financialinstruments where Ainstrumentcode = Instrumentscode AND  
Acssn = Ccssn AND Cmsn = '%s' order by Ccssn asc", tmpSsn);
16. // 계좌기록을 출력하려고 합니다. History 테이블에서 계좌기록 정보를 가져와  
Hindex 오름차순으로 정렬합니다.  
sql = String.format("select \* from history where Haccountnumber = '%s' order  
by Hindex asc", inputStr);
17. // 제거한 계좌의 계좌기록을 History 테이블에서 삭제합니다.  
sql = String.format("delete from history where Hindex = '%s'", input);
18. // 매니저가 담당하는 고객들의 계좌 정보를 출력하려고 합니다. Account,  
Customer, FinancialInstruments 테이블에서 정보를 equijoin 합니다.  
sql = String.format("select Category, Accountnumber, Ccssn, Cmsn from account,  
customer, financialinstruments where Ainstrumentcode = Instrumentscode AND  
Acssn = Ccssn AND Cmsn = '%s' order by Ccssn asc", tmpSsn);
19. // 계좌기록을 출력하려고 합니다. History 테이블에서 계좌기록 정보를 가져와  
Hindex 오름차순으로 정렬합니다.  
sql = String.format("select \* from history where Haccountnumber = '%s' order  
by Hindex asc", inputStr);
20. // 제거하고자하는 계좌기록을 History 테이블에서 삭제합니다.  
sql = String.format("delete from history where Hindex = '%s'", input);
21. // Customer 테이블에서 Ssn 정보를 가져옵니다.  
rs = stmt.executeQuery("select Ccssn from Customer");
22. // Customer 의 정보를 업데이트합니다.  
sql = String.format("update customer set Cname = '%s', Cphonenum = '%s', Cmsn  
= '%s' where Ccssn = '%s'", inputStr1, inputStr2, inputStr3, inputStr);
23. // 계좌를 만들고자 합니다. Account, Customer 테이블을 equijoin 하여 계좌 갯수,  
신용점수 정보를 가져옵니다. 계좌 갯수 제한 조건을 확인하기 위해, count 문을  
통해 고객의 계좌 갯수를 셉니다.

```

sql = String.format("select count(Accountnumber) as totalCount, Credit from
account join customer on Acssn = Csn where Csn = '%s' ", inputStr);
24. // 삭제되지 않은 은행 상품을 출력하려고 합니다. FinancialInstruments
테이블에서 Not In 구문을 통해 DeadInstruments 테이블에 존재하는 은행 상품은
제외하고 정보를 가져옵니다.
sql = String.format("select Instrumentscode, Category, Standard, Interest from
financialinstruments where Instrumentscode NOT IN (select Dinstrumentscode
from deadinstruments where Dinstrumentscode = Instrumentscode) order by
Category asc, Standard desc");
25. // 고객의 신용점수가 만들고자하는 은행상품의 신용점수기준 이상인지 확인하기
위해 FinancialInstruments 테이블에서 해당 은행상품의 신용점수기준 attribute 를
가져옵니다.
sql = String.format("select Standard from financialinstruments where
Instrumentscode = '%s'", inputStr1);
26. // 계좌번호 중복을 피하기 위해, Account 테이블에 존재하는 계좌들의 계좌번호
정보들을 가져옵니다.
rs = stmt1.executeQuery("select Accountnumber from account");
27. // 필요한 value 들을 넣어 새로운 Account 테이블의 tuple 을 생성합니다.
sql = String.format("insert into account values ('%s', '%s', '%s', '%s',
'0')",inputStr1, inputStr2, inputStr3, inputStr);
28. // 고객의 계좌정보를 출력하려고 합니다. Customer, Account,
FinancialInstruments 테이블을 equijoin 해 고객의 계좌정보들을 가져옵니다.
sql = String.format("select Category, Accountnumber, Cmsn, Value, Interest
from account, financialinstruments, customer where account.Ainstrumentscode =
financialinstruments.Instrumentscode AND account.Acscn = customer.Cscn AND
Acscn = '%s'", inputStr);
29. // 입력한 계좌번호가 고객의 계좌인지 확인하기 위해 count 문을 통해 고객의 계좌
중 해당 계좌번호를 가진 tuple 이 있는지 확인합니다. cnt 값이 0 이면 해당
계좌번호를 가진 고객의 계좌가 존재하지 않습니다.
sql = String.format("select count(*) as cnt from account join customer on Acscn
= Cscn where Accountnumber = '%s' AND Cscn = '%s'", inputStr1, inputStr);
30. // 고객의 계좌 잔고에 입금한 금액을 업데이트합니다.
sql = String.format("update Account set Value = Value + '%s' where
Accountnumber = '%s'", input, inputStr1);
31. // History 테이블에 입금 기록 tuple 을 추가합니다.
sql = String.format("insert into history values ('%s', '%s', '%s') ",
inputStr1, hIdx++, input);
32. // 입력한 계좌번호가 고객의 계좌인지 확인하기 위해 count 문을 통해 고객의 계좌
중 해당 계좌번호를 가진 tuple 이 있는지 확인합니다. cnt 값이 0 이면 해당
계좌번호를 가진 고객의 계좌가 존재하지 않습니다.
sql = String.format("select count(*) as cnt from account join customer on Acscn
= Cscn where Accountnumber = '%s' AND Cscn = '%s'", inputStr1, inputStr);

```

33. // 비밀번호가 맞는지 확인하기 위해 해당 계좌의 비밀번호 정보를 가져옵니다.  
`sql = String.format("select Password from account where Accountnumber = '%s'", inputStr1);`
34. // 출금은 신용점수의 값보다 많이 출금할 수 없습니다. 그 이상을 출금하기 위해서는 담당 관리자의 비밀번호가 필요합니다. 해당 조건 확인을 위해 Customer, Manager 테이블을 equijoin 해 신용점수와 매니저의 비밀번호를 가져옵니다.  
`sql = String.format("select Credit, Password from customer join manager on customer.Cmsn = manager.Mssn where Cmsn = '%s' ", inputStr);`
35. // 고객의 계좌 잔고에 출금한 금액을 업데이트합니다.  
`sql = String.format("update Account set Value = Value - '%s' where Accountnumber = '%s'", input, inputStr1);`
36. // History 테이블에 입금 기록 tuple 을 추가합니다.  
`sql = String.format("insert into history values ('%s', '%s', '%s') ", inputStr1, hIdx++, input);`
37. // 이자 계산에는 해당 계좌의 잔고와 은행상품의 이자 정보가 필요합니다. 이자 계산을 위해 Account, FinancialInstruments 테이블을 equijoin 해 정보를 가져옵니다.  
`sql = String.format("select * from account join financialinstruments on Ainstrumentcode = Instrumentscode");`
38. // 계좌 잔고를 업데이트합니다.  
`sql = String.format("update account set Value = '%s' where Accountnumber = '%s'", tmp2, rs.getString("Accountnumber") );`
39. // 이자 반영 기록을 남깁니다. History 테이블에 필요한 value 들을 통해 tuple 을 추가합니다.  
`sql = String.format("insert into history values ('%s', '%s', '%s') ", rs.getString("Accountnumber"), hIdx++, (tmp2-tmp));`
40. // 잔고가 0 이상이 된 계좌들을 삭제하려고 합니다. Account, FinancialInstruments 테이블을 equijoin 해 계좌와 계좌 Category(자유입출금/대출) 정보를 가져옵니다.  
`sql = String.format("select * from account join financialinstruments on Ainstrumentcode = Instrumentscode");`
41. // 계좌 Category 가 대출이고 잔고가 0 이상이면 계좌를 삭제합니다.  
`sql = String.format("delete from account where Accountnumber = '%s'", rs.getString("Accountnumber"));`
42. // 삭제한 계좌의 계좌기록들을 History 테이블에서 제거합니다.  
`sql = String.format("delete from history where Haccountnumber = '%s'", rs.getString("Accountnumber"));`
43. // 신용점수를 업데이트하려고 합니다. Customer, Account, FinancialInstruments 테이블을 equijoin 하고, 자유입출금계좌(Category: 0)들의 정보를 가져옵니다. group by 문을 통해 고객 단위로 정보를 묶습니다. sum 문을 통해 잔고의 합 정보를 가져옵니다.

```

sql = String.format("select Ccssn, Credit, sum(Value) as totalValue, Credit
from customer, account, financialinstruments where Ccssn = Accssn AND
Ainstrumentscode = Instrumentscode AND Category = '0' group by Ccssn");
44. // 신용점수(자유입출금계좌의 잔고 합 / 100)의 변경내용을 업데이트합니다.
sql = String.format("update Customer set Credit = '%s' where Ccssn = '%s'",
tmp2, rs.getString("Ccssn") );
45. // 신용점수 900 이상의 고객은 담당관리자가 최고관리자로 변경됩니다. 그를 위해
최고 관리자의 Ssn 정보를 가져옵니다.
sql = String.format("select Mssn from Manager where Position = '0'");
46. // 신용점수 900 이상이고 담당관리자가 최고관리자가 아닌 고객들의 정보를 <>문을
통해 가져옵니다.
sql = String.format("select Ccssn, Cname from customer where Credit > 900 AND
Cmssn <> '%s'", tmpSsn);
47. // 신용점수 900 이상이고, 담당관리자가 최고관리자가 아닌 고객들의 관리자를
최고관리자로 변경합니다.
sql = String.format("update Customer set Cmssn = '%s' where Ccssn = '%s'",
tmpSsn, rs.getString("Ccssn") );

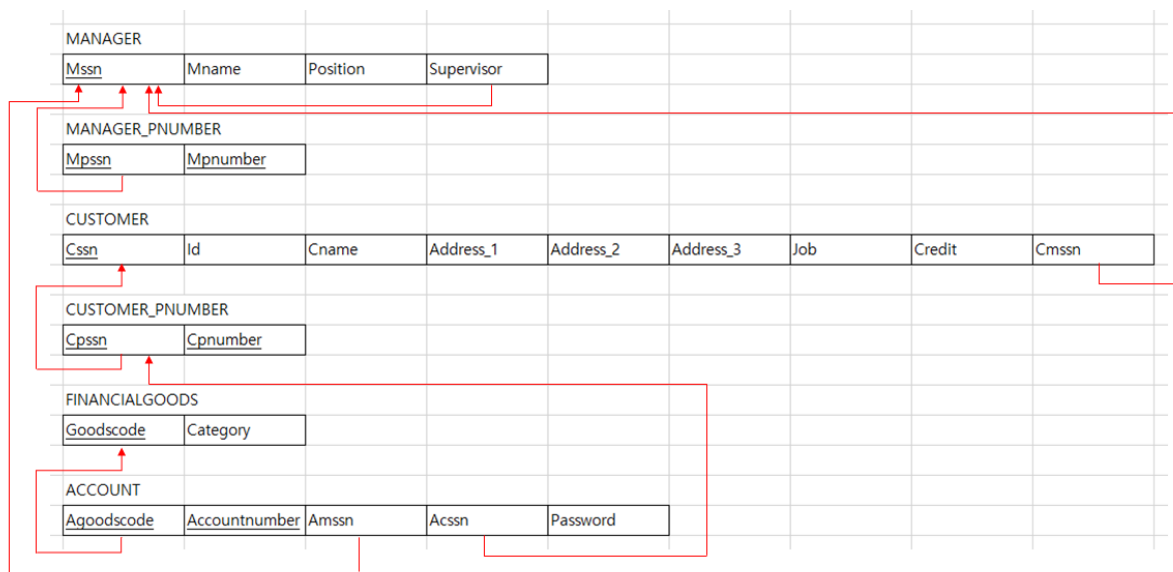
```



## 5. 변경사항

기존 프로젝트를 진행하면서 구상했던 기능들이 다양하게 있었습니다. 하지만, 구현하면서 난이도, 현실성 등에 어려움을 느끼고 많은 기능을 제거하였습니다.

다음 페이지는 기존 프로젝트 구상 자료입니다.



#### 기존 프로젝트 구상

1. 관리자가 사용자의 계좌 입출금 내역 관리의 범위를 어떻게 할지 -> 계좌 개설, 삭제는 관리자의 허가가 필요하다고 설정할까?
2. 계좌 개설의 제한 n개는 신용점수, 나이에 따라 결정됨 -> function
3. 사용자가 자신의 계좌에 입금하는 것은 제한 없이 입금 가능
4. 사용자가 자신의 계좌에서 출금하는 것은 신용에 따른 한도 제한을 두고, 초과 시에 담당 관리자가 승인해야 함.
5. 계좌간 계좌이체 기능. 자신의 계좌 내에서는 한도 제한 없으나, 타인과의 계좌이체는 한도 제한. 초과 시에 담당 관리자가 승인해야 함. 한도는 출금 한도와 동일하다고 설정.
6. 신용점수는 상품 이용 기간등의 크레딧에 따라 변동. 관리자 직급에 따른 조절 가능 신용점수 범위가 존재한다. 신용점수가 범위 넘어가면 관리자 새롭게 선택.
7. 관리자도 password가 필요할 거 같음. 관리자 등급에 따라 허용 가능한 이체의 최대 금액 제한 존재. 그 이상은 상사로부터의 허가 필요.
8. 최대계좌수는 신용점수, 나이, 제일 금액이 많은 예금 혹은 자유입출금계좌의 금액에 따라 결정됨. (적금을 만들어서 몇 개월 후에 가능하다고 알려주는 기능도 넣을까?)
9. 계좌는 자유입출금 필수, 예금, 적금, 대출 계좌도 존재.
10. 은행 상품의 이율은 신용점수에 따라 변화한다. 신용점수가 변화할 때 추천상품 기능에서 더 나은 예금 혹은 대출 상품을 추천받을 수 있다.

읽어 주셔서 감사합니다.