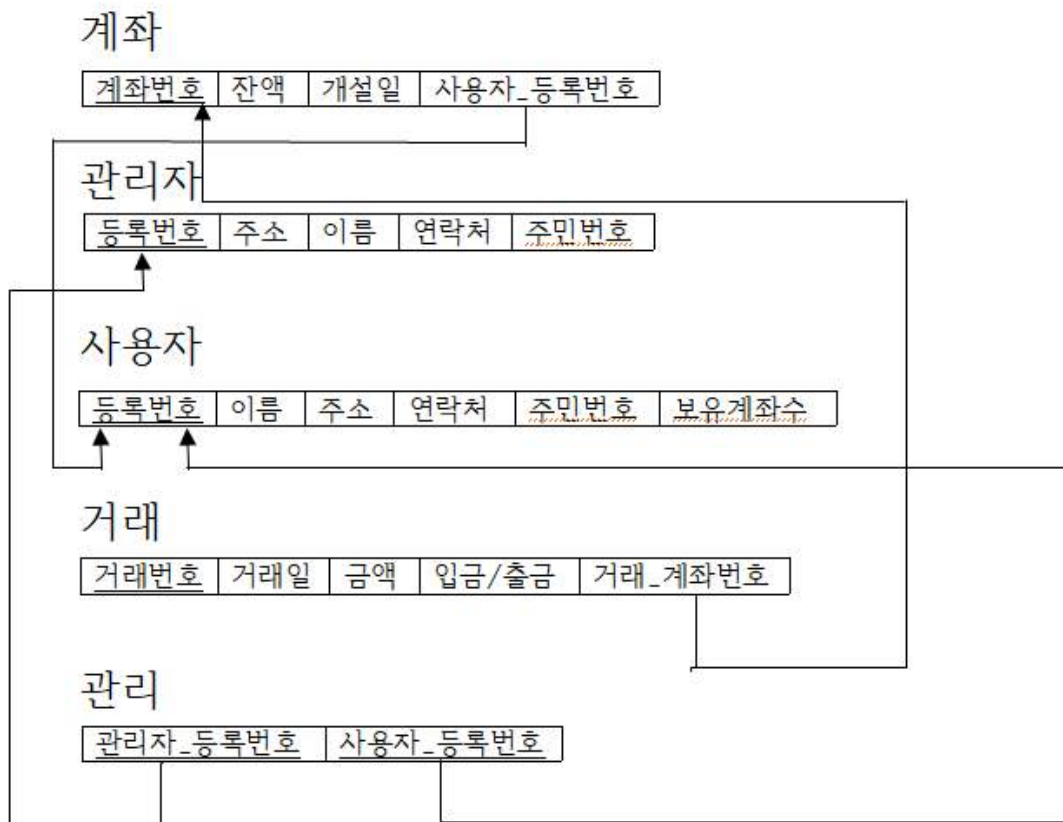


Summary of Project4

수업시간 : 월 13:00 ~ 15:00 목 17:00 ~ 19:00
2018008559 컴퓨터소프트웨어학부 신상윤

0. 수정사항 및 설명



- 계좌안에 거래일, 거래금액, 입금/출금등이 attribute로 설정되어있었는데 이를 '거래' entity로 만들어 계좌와 거래가 1:N의 관계를 갖게 했다. 거래의 primary key는 거래번호이고, foreign key는 계좌번호를 가져온 값이다.
- 원래 대출 entity가 있었으나 구현하기 까다롭고, 거래 entity가 추가됨에 따라 삭제하였다.

1. Import

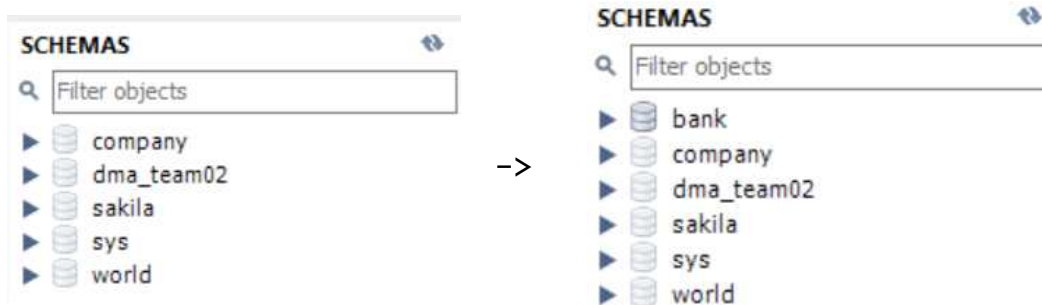
```
import mysql.connector //mysql과 연결할 수 있도록 한다.  
import datetime //현재 시각을 가져올 수 있게 한다.
```

2. 함수

가. schema(host,user,password)

```
cnx = mysql.connector.connect(host=host, user=user, password=password)
cursor = cnx.cursor()
cursor.execute('SET GLOBAL innodb_buffer_pool_size=2*1024*1024*1024;')
cursor.execute('DROP DATABASE IF EXISTS bank;')
cursor.execute('CREATE DATABASE IF NOT EXISTS bank;')
cursor.close()
```

mysql과 연결하고, bank 스키마가 존재하면 삭제하고, bank 스키마가 존재하지 않으면 생성한다. 각각 drop과 create로 실행했다.



나. table(host, user, password)

mysql과 연결하고, bank를 사용하겠다고 쿼리를 작성한다.

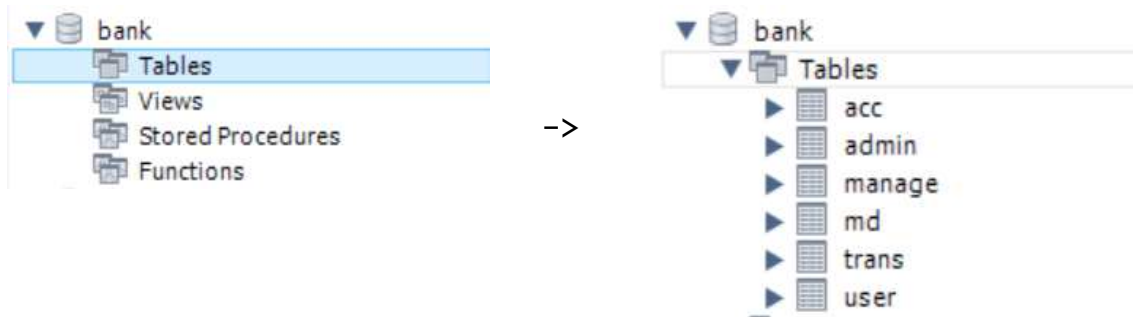
이후 계좌, 거래, 관리자, 사용자, 관리, meta_data table을 작성한다.

최소 한명의 관리자가 있어야 한다고 생각하여 처음에 data를 넣어준다.

meta_data? b+tree에서 index.dat 역할을 하는 meta_data를 구현했다. table의 역할은 하지 않는다. 각 table의 primary key인 acc_number, user_number 등등을 자동으로 할당해주기 위해서 순차적으로 할당한다 가정했다. 이를 위해서 몇 번째 번호까지 사용했는지를 저장해야 하므로 meta_data table을 만들어 관리한다. 처음 값은 1, 1, 2, 1로 관리자가 이미 한 명이 있으므로 2번부터 할당해준다.

create로 table을 만들고 insert로 data를 넣어주었다.

```
INSERT INTO md VALUES ();
```



다. new_user(host, user, password)

mysql과 연결하고, meta_data에서 다음으로 할당받아야 할 사용자번호를 받는다.

```
SELECT md_use FROM md;
```

이후 meta_data의 md_user 값을 1 증가시킨 값으로 update 한다.

```
UPDATE md SET md_use = user_number+1;
```

받은 사용자번호는 새로운 사용자의 primary key 값이 된다.

이후 이름, 주소, 연락처, 주민번호를 받아 user table에 insert 한다.

```
INSERT INTO user VALUES {};
```

이때 초기 계좌 개수는 1개이다. 새로 사용자가 만들어질 때 계좌 개설도 같이 진행하기 때문이다. 계좌 개설도 같이 진행하기 때문에 계좌번호도 meta_data에서 가져오고,

```
SELECT md_acc FROM md;
```

update 한다.

```
UPDATE md SET md_acc = acc_number+1;
```

잔액은 0, 개설일은 현재 날짜인 계좌를 insert한다.

```
INSERT INTO acc VALUES {};
```

마지막으로 담당 관리자와 연결을 하는데,

```
SELECT admin_number FROM admin;
```

모든 관리자와 연결을 한다. 즉, 관계 table을 모두 생성한다.

```
INSERT INTO manage VALUES (admin_number, user_number);
```

회원번호를 입력해주세요. 신규등록을 원하시면 -1을 입력해주세요.-1

이름,주소,연락처,주민번호를 입력해주세요: 신상윤 의왕시 01047405160 990428

회원가입을 축하합니다!! 당신의 회원번호는 1
계좌번호는 1입니다 잊지 않도록 주의하세요!!

	user_number	user_name	user_adress	user_phone	user_ssn	user_accnum
▶	1	신상윤	의왕시	01047405160	990428	1
*	NULL	NULL	NULL	NULL	NULL	NULL

라. whoareyou(host, user, password,user_number)

mysql과 연결하고, user table에서 user_number가 변수 user_number와 같을 때 사용자 이름을 return 한다.

```
SELECT user_name
```

```
FROM user
```

```
WHERE user_number= user_number;
```

만약 같은 row가 없으면 빈 string이므로 등록되지 않은 user이다.

회원번호를 입력해주세요. 신규등록을 원하시면 -1을 입력해주세요.1
안녕하세요 신상윤님

마. check_user_accnum(host, user, password,user_number)

사용자가 보유하고 있는 계좌의 개수를 return 한다.

```
SELECT user_accnum
```

```
FROM user
```

```
WHERE user_number = user_number;
```

한 사람당 만들 수 있는 계좌를 3개로 설정했기 때문에 확인하는 것이다.

어떤 업무로 오셨나요?

- 1.계좌생성
- 2.잔액확인
- 3.입금
- 4.출금
- 5.계좌 해지
- 6.회원탈퇴
- 7.최근 입출금 내역1

이미 등록된 계좌가 3개입니다!!!

바. new_acc(host, user, password,user_number)

새로 계좌를 하나 만든다. 먼저 meta_data에서 사용할 계좌번호를 가져오고,

```
SELECT md_acc FROM md;
```

update 한다. UPDATE md SET md_acc = acc_number + 1;

마찬가지로 잔액이 0, 개설일은 오늘 날짜인 계좌가 만들어진다.

```
INSERT INTO acc VALUES {};
```

계좌를 개설한 사용자의 보유 계좌 수를 하나 늘려야하기 때문에 update 해준다.

```
SELECT user_accnum
```

```
FROM user
```

```
WHERE user_number = user_number;
```

```
UPDATE user SET user_accnum = user_accnum+1
```

```
WHERE user_number = user_number;
```

어떤 업무로 오셨나요?

1. 계좌생성
2. 잔액확인
3. 입금
4. 출금
5. 계좌 해지
6. 회원탈퇴
7. 최근 입출금 내역1

성공적으로 계좌가 만들어졌습니다! 계좌번호는 3입니다

	acc_number	acc_money	acc_date	acc_user_number
▶	1	0	2021-12-03	1
	2	0	2021-12-03	1
	3	0	2021-12-03	1
•	NULL	NULL	NULL	NULL

사. my_acc(host, user, password, user_number)

사용자가 현재 보유 중인 모든 계좌를 출력한다.

```
SELECT acc_number
```

```
FROM acc, user
```

```
WHERE user_number = user_number and acc_user_number = user_number;
```

어떤 업무로 오셨나요?

1. 계좌생성
2. 잔액확인
3. 입금
4. 출금
5. 계좌 해지
6. 회원탈퇴
7. 최근 입출금 내역2

보유중인 계좌는 1 2 3 입니다.

아. `check_acc_money(host, user, password, user_number, acc_number)`

사용자번호와 계좌번호를 입력받아 계좌의 잔액을 출력한다. 사용자번호도 받는 이유는 본인 계좌가 아닌 다른 계좌를 볼 수 없게 하기 위함이다.

```
SELECT acc_money
```

```
FROM acc, user
```

```
WHERE user_number = user_number and acc_user_number = user_number  
and acc_number = acc_number
```

어떤 업무로 오셨나요?

1. 계좌생성
2. 잔액확인
3. 입금
4. 출금
5. 계좌 해지
6. 회원탈퇴
7. 최근 입출금 내역2

보유중인 계좌는 1 2 3 입니다.

잔액확인을 하고싶은 계좌번호를 입력해주세요. 2

보유하고 계신 잔액은 123원 입니다.

	acc_number	acc_money	acc_date	acc_user_number
▶	1	0	2021-12-03	1
	2	123	2021-12-03	1
	3	0	2021-12-03	1
★	NULL	NULL	NULL	NULL

자. in_acc_money(host, user, password, user_number, acc_number)

```
SELECT acc_money
FROM acc, user
WHERE user_number = user_number and acc_user_number = user_number
and acc_number = acc_number
```

선택한 계좌에 입금을 진행한다. 입금하고 싶은 금액을 입력받고 계좌의 잔액을 (원래 잔액 + 입력받은 금액)의 값으로 update 시켜준다.

```
UPDATE acc SET acc_money = total_money WHERE acc_number = acc_number
```

입금을 성공적으로 했다면, 거래 table에 '입금' 거래기록을 insert 해준다.

```
UPDATE md SET md_trans = trans_number+1;
```

```
INSERT INTO trans VALUES {};
```

어떤 업무로 오셨나요?

1. 계좌생성
2. 잔액확인
3. 입금
4. 출금
5. 계좌 해지
6. 회원탈퇴
7. 최근 입출금 내역3

보유중인 계좌는 1 2 3 입니다.

입금을 하고싶은 계좌번호를 입력해주세요. 1

입금하고싶은 금액을 입력해주세요. 345

입금이 완료되었습니다

	acc_number	acc_money	acc_date	acc_user_number
▶	1	345	2021-12-03	1
	2	123	2021-12-03	1
	3	0	2021-12-03	1
●	NULL	NULL	NULL	NULL

차. out_acc_money(host, user, password, user_number, acc_number)

```
SELECT acc_money
```

```
FROM acc, user
```

```
WHERE user_number = user_number and acc_user_number = user_number  
and acc_number = acc_number;
```

선택한 계좌에 출금을 진행한다. 출금하고 싶은 금액을 입력받는다.

이때 잔액과 비교를 했을 때 잔액이 더 작다면 출금을 할 수 없으므로 잔액이 부족하다고 출력한다. 만약 잔액이 더 크다면 잔액을 (잔액 - 입력받은 금액)으로 update 시켜준다. UPDATE acc SET acc_money = total_money WHERE acc_number = acc_number

마찬가지로 출금을 성공적으로 했다면, 거래 table에 '출금' 거래기록을 insert 해준다. SELECT md_trans FROM md;

```
UPDATE md SET md_trans = trans_number+1;
```

```
INSERT INTO trans VALUES {};
```

보유중인 계좌는 1 2 3 입니다.

출금을 하고싶은 계좌번호를 입력해주세요. 2

출금하고싶은 금액을 입력해주세요. 23

출금이 완료되었습니다

보유중인 계좌는 1 2 3 입니다.

출금을 하고싶은 계좌번호를 입력해주세요. 2

출금하고싶은 금액을 입력해주세요. 3000

잔액이 부족합니다!!

	acc_number	acc_money	acc_date	acc_user_number
▶	1	345	2021-12-03	1
	2	100	2021-12-03	1
	3	0	2021-12-03	1
●	NULL	NULL	NULL	NULL

카. del_acc_number(host, user, password, user_number, acc_number)

```
SELECT acc_money
FROM acc, user
WHERE user_number = user_number and acc_user_number = user_number
and acc_number = acc_number;
```

계좌 삭제를 진행한다. 계좌 삭제를 진행하면서 잔액이 남아있다면 잔액을 모두 출금하고, SELECT md_trans FROM md;

거래기록에도 '계좌해지로인한출금'이라 적는다.

```
UPDATE md SET md_trans = trans_number+1;
```

이후에 계좌에 관련된 table을 모두 삭제하고,

```
DELETE FROM acc WHERE acc_number = acc_number;
```

user가 보유하고 있는 계좌도 -1 값으로 update 해준다.

```
SELECT user_accnum
```

```
FROM user
```

```
WHERE user_number = user_number;
```

```
UPDATE user SET user_accnum = user_accnum - 1 WHERE user_number =
user_number
```

보유중인 계좌는 1 2 3 입니다.

해지 하고싶은 계좌번호를 입력해주세요. 2

계좌 해지 완료

	acc_number	acc_money	acc_date	acc_user_number
▶	1	345	2021-12-03	1
	3	0	2021-12-03	1
●	NULL	NULL	NULL	NULL

	trans_number	trans_type	trans_date	trans_money	trans_acc_number
▶	1	입금	2021-12-03	123	2
	2	입금	2021-12-03	345	1
	3	출금	2021-12-03	23	2
	4	계좌해지로인한출금	2021-12-03	100	2
•	NULL	NULL	NULL	NULL	NULL

	user_number	user_name	user_adress	user_phone	user_ssn	user_accnum
▶	1	신상운	의왕시	01047405160	990428	2
•	NULL	NULL	NULL	NULL	NULL	NULL

타. del_user_number(host, user, password, user_number)

```
SELECT user_accnum
```

```
FROM user
```

```
WHERE user_number = user_number;
```

사용자 정보를 모두 지워준다. 이때 계좌가 모두 해지된 상태 이어야한다.

이를 먼저 검사하고, 이후 남아있는 계좌가 없다면 user와 관련된 table을 모두 삭제 해준다.

```
DELETE FROM user WHERE user_number = user_number;
```

어떤 업무로 오셨나요?

- 1.계좌생성
- 2.잔액확인
- 3.입금
- 4.출금
- 5.계좌 해지
- 6.회원탈퇴
- 7.최근 입출금 내역6

보유중인 계좌가 남아있습니다! 계좌 해지부터 해주세요!

어떤 업무로 오셨나요?

- 1.계좌생성
- 2.잔액확인
- 3.입금
- 4.출금
- 5.계좌 해지
- 6.회원탈퇴
- 7.최근 입출금 내역6

회원탈퇴가 완료되었습니다. 그동안 이용해주셔서 감사합니다

	user_number	user_name	user_adress	user_phone	user_ssn	user_accnum
*	NULL	NULL	NULL	NULL	NULL	NULL

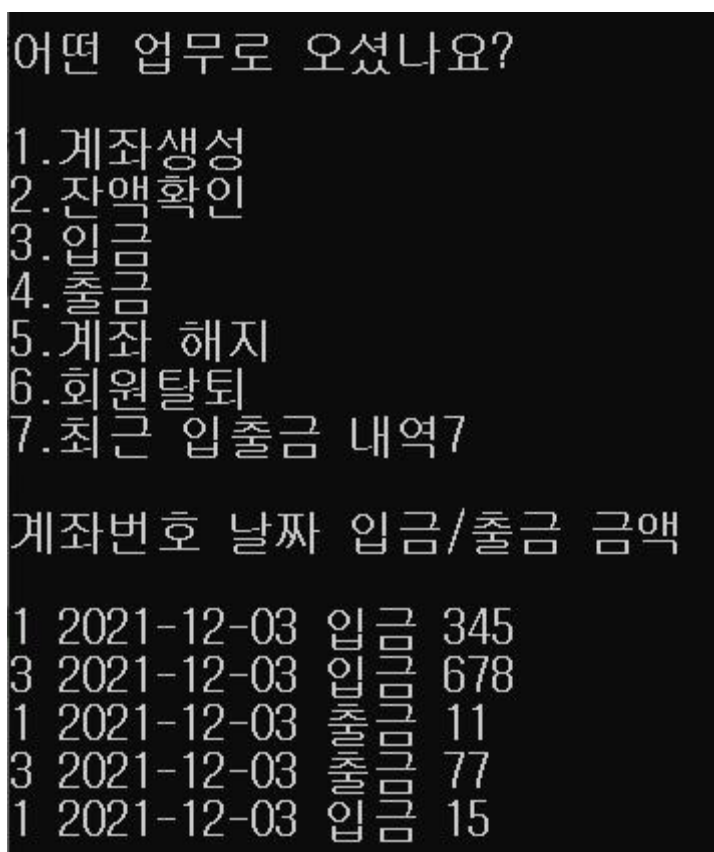
파. user_my_trans(host, user, password, user_number)

user의 최근 20건 이내의 거래기록을 출력한다.

```
SELECT trans_acc_number, trans_date, trans_type, trans_money
```

```
FROM user,acc,trans
```

```
WHERE user_number = user_number and acc_user_number = user_number  
and acc_number = trans_acc_number;
```



하. new_admin(host, user, password)

새로운 관리자를 만든다. 관리자 번호는 meta_data에서

```
SELECT md_admin
```

```
FROM md;
```

가져오고 update 한다. UPDATE md SET md_admin = admin_number+1;

당신은?
 1.관리자
 2.사용자
 3.프로그램을 종료한다.1
 비밀번호를 입력해주세요 : kas123123
 안녕하세요 관리자님 관리자 번호를 입력해주세요. 신규 관리자 등록을 원하시면 -1을 입력해주세요.-1
 주소,이름,연락처,주민번호를 입력해주세요: 울릉도 홍길동 15881588 201202
 새 관리자님 환영합니다!

	admin_number	admin_adress	admin_name	admin_phone	admin_ssn
▶	1	의왕시	신상윤	01047405160	990428
	2	울릉도	홍길동	15881588	201202
•	NULL	NULL	NULL	NULL	NULL

거. whoareyouadmin(host, user, password, admin_number)

```
SELECT admin_name
FROM admin
WHERE admin_number = admin_number;
```

관리자 번호를 입력받고 관리자의 이름을 return 한다.

안녕하세요 관리자님 관리자 번호를 입력해주세요. 신규 관리자 등록을 원하시면 -1을 입력해주세요.2
 안녕하세요 홍길동 관리자님

너. admin_user(host, user, password, admin_number)

관리자가 담당하고 있는 user들의 이름과 계좌번호를 모두 출력한다.

```
SELECT user_name, acc_number
FROM manage, user, acc
WHERE manage_admin_number = admin_number and manage_user_number
= user_number and user_number = acc_user_number;
```

안녕하세요 신상윤 관리자님
 어떤 업무로 오셨나요?
 1.담당 회원/계좌 확인
 2.최근 거래기록 확인1
 김철수 4
 독수리 5

	user_number	user_name	user_adress	user_phone	user_ssn	user_accnum
▶	2	김철수	서울	7388	12345	1
	3	독수리	하늘로얄오리	9999	13035678	1
●	NULL	NULL	NULL	NULL	NULL	NULL

더. admin_us

최근 거래기록 20건을 출력한다.

```
SELECT trans_acc_number, trans_date, trans_type, trans_money
FROM trans;
```

안녕하세요 신상윤 관리자님

어떤 업무로 오셨나요?

1. 담당 회원/계좌 확인
2. 최근 거래기록 확인

계좌번호 날짜 입금/출금 금액

```
2 2021-12-03 입금 123
3 2021-12-03 계좌해지로인한출금 601
5 2021-12-03 입금 898
5 2021-12-03 출금 11
4 2021-12-03 입금 343
4 2021-12-03 입금 22
5 2021-12-03 계좌해지로인한출금 887
1 2021-12-03 입금 345
2 2021-12-03 출금 23
2 2021-12-03 계좌해지로인한출금 100
3 2021-12-03 입금 678
1 2021-12-03 출금 11
3 2021-12-03 출금 77
1 2021-12-03 입금 15
1 2021-12-03 계좌해지로인한출금 349
```

3. main

mysql에 접속을 하기 위해

host = 'localhost'

user = 'root'

password = 'kas123123'로 설정하였다.

이후 초기화 진행 여부를 묻고, while 문이 시작된다.

while문은 각 print에 써져있는 내용대로 진행할 수 있도록 적절하게 함수들을 설정하였다.

```
당신은?
1.관리자
2.사용자
3.프로그램을 종료한다.1
비밀번호를 입력해주세요 : kas123123
안녕하세요 관리자님 관리자 번호를 입력해주세요. 신규 관리자 등록을 원하시면 -1을 입력해주세요.1
안녕하세요 신상윤 관리자님
어떤 업무로 오셨나요?
1.담당 회원/계좌 확인
2.최근 거래기록 확인
당신은?
1.관리자
2.사용자
3.프로그램을 종료한다.2
회원번호를 입력해주세요. 신규등록을 원하시면 -1을 입력해주세요.2
안녕하세요 김철수님
어떤 업무로 오셨나요?
1.계좌생성
2.잔액확인
3.입금
4.출금
5.계좌 해지
6.회원탈퇴
7.최근 입출금 내역
```

각각 관리자, 사용자 창이다.