

FINAL 프로젝트: DBMS 프로그램

I. ER Diagram/Relational Diagram

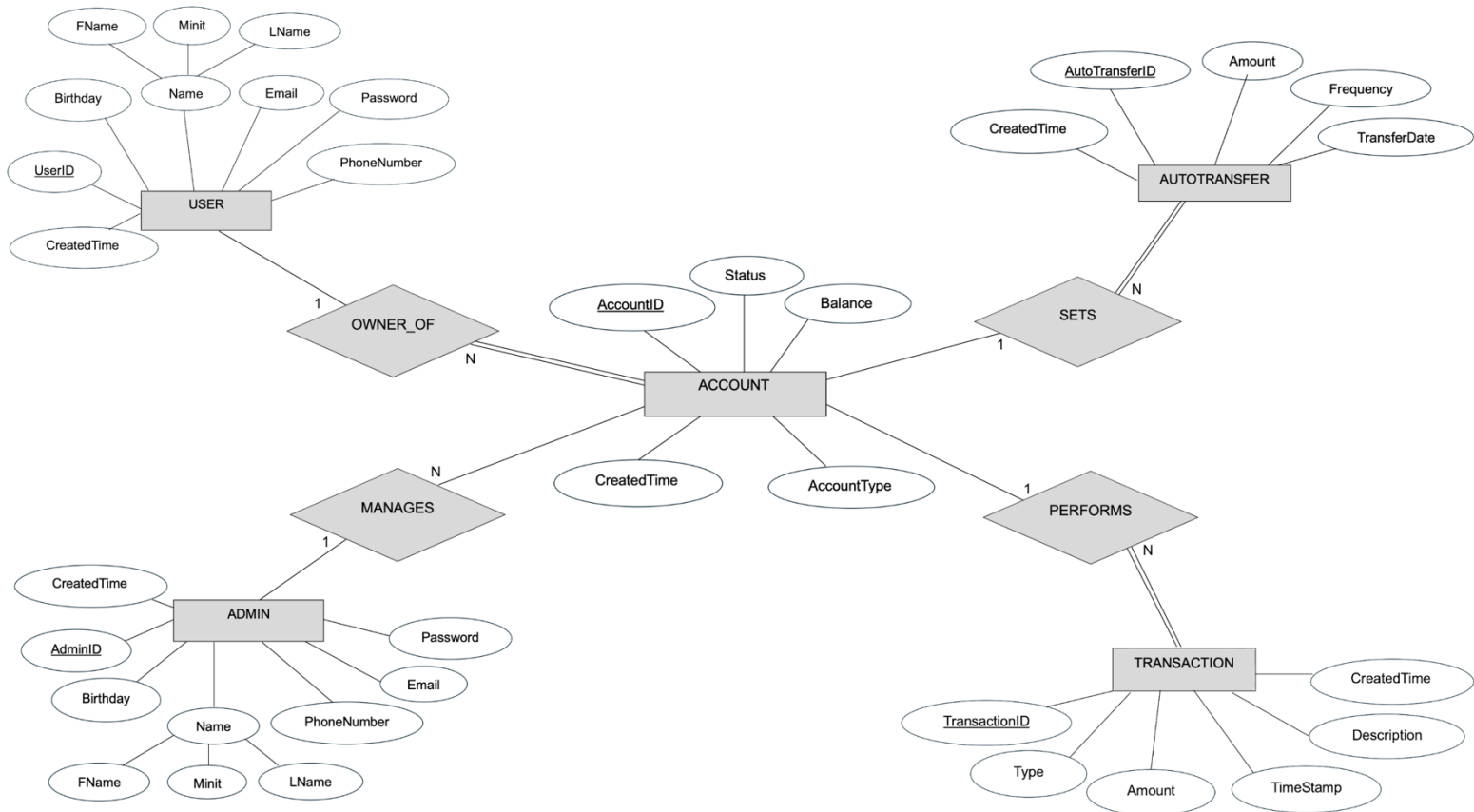


Figure 1. ER Diagram

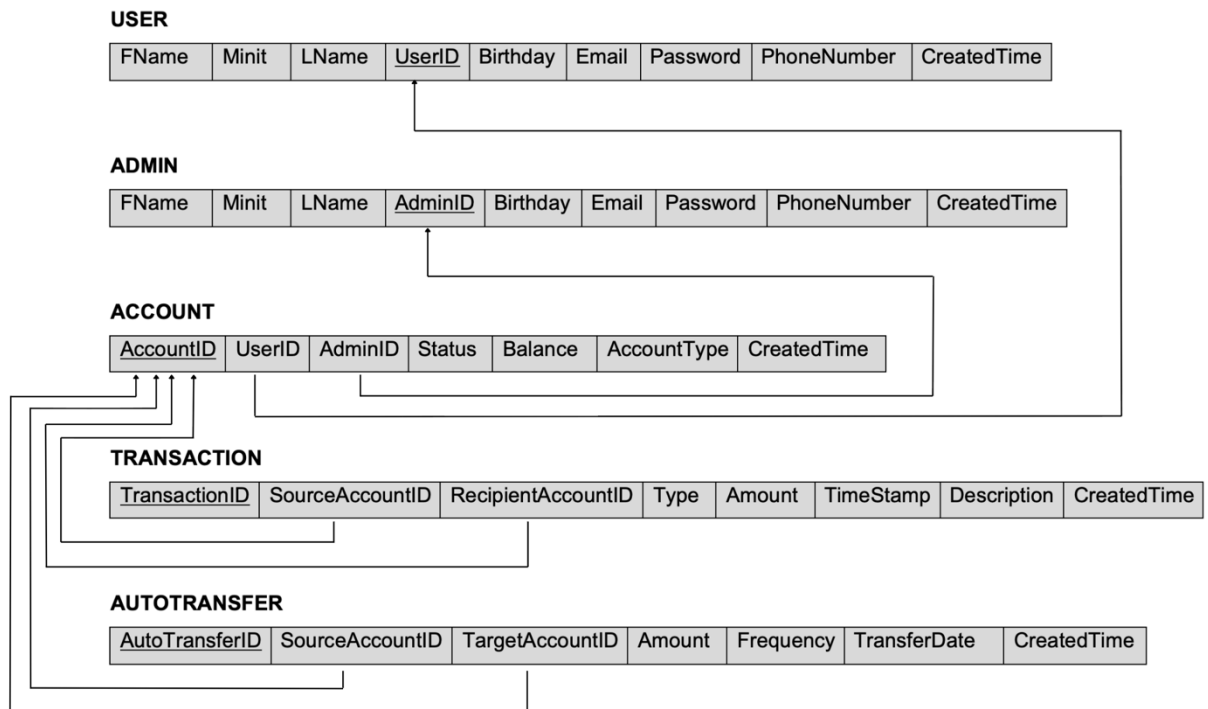


Figure 2. Relational Diagram

Main Changes to the Diagrams

- Added CreatedTime entities in USER, ADMIN, TRANSACTION, and AUTOTRANSFER to signal when each of these were created.
- Changed “Type” → “AccountType” in ACCOUNT for clarity.
- Changed “AccountID” → “SourceAccountID” in TRANSACTION for clarity.

II. Explanation of DBMS Program

The DBMS program, bank.py, is a CLI application that connects to the bank_app MySQL database. In essence, a person can log into the application as either a user (has access to only their own account) or as an admin (has access to user's history) in the login page. The program uses simple text prompts as the interface and stores monetary values as decimal values to avoid any errors.

As a user, you can sign up and create your own bank application accounts, login to these accounts and create/delete your own bank accounts. You can also transfer money from one bank account to another one of your bank accounts. Moreover, you can withdraw and deposit money into their own accounts. Also, users can view all the transactions made on your account and set autotransfers. On the other hand, as an admin, you are given access to an admin account made from the bank and login through the application. Once logged in, you can see and list out all of the users' histories or search a specific user's history (transactions, accounts, autotransfers).

The program is separated into seven main categories that contain several defining functions:

1. User Creation
 - Allows a user to create their own account (sign up) at the login page by inserting the values (entities) to the user table in bank_app database.
2. User/Admin Login
 - Validates user/admin login by checking the database if such user/admin exists.
 - Successfully user/admin in if the email and password are in the database.
3. Account Management
 - Allows a user to create and delete their bank accounts. When creating an account, it inserts the values into the account table. When deleting an account, it first checks if the user inputted their own proper account number, then proceeds to delete transactions, autotransfers, and deletes the account entirely.
4. Transactions (Deposit, Withdraw, Transfer)
 - These allow a user to update their balance by depositing/withdrawing money into their account and transferring their money into another account.
 - Can only transfer 10,000,000 at once.
5. Autotransfer
 - Allows a user to create an autotransfer.
6. Lists
 - These functions consist of select queries that fetches and lists out all users and user-specific accounts, transactions, and autotransfers
7. Admin Controls
 - search_user allows the admin to search for a user and if the user exists, the admin can view the user's accounts, transaction history, and autotransfer history.

III. Screenshot of Results

```
=====
                        LOGIN MENU
=====
1) Login as User
2) Login as Admin
3) Signup as User
0) Exit

Choice: █
```

Figure 3. Login Menu

```
=====
                        LOGIN MENU
=====
1) Login as User
2) Login as Admin
3) Signup as User
0) Exit

Choice: 3

===== SIGN UP =====
First name: Jane
Middle initial (optional):
Last name: Doe
Birthday (YYYY-MM-DD): 1999-12-12
Email: janedoe@gmail.com
Password:
Phone number (010-xxxx-xxxx): 010-4321-8765
```

Figure 4. Signup Menu

----- USER LOGIN -----

```
=== USER LOGIN ===  
Email: jminhwang@hanyang.ac.kr  
Password:  
Welcome, Jimin!  
  
===== USER MENU =====  
===== 1) Create my account  
2) View my accounts  
3) Create my transaction  
4) View my transactions  
5) Create AutoTransfer  
6) View my AutoTransfers  
7) Delete my account  
0) Logout  
Choice: █
```

Figure 5. Logging in as a User

```
Create your account:  
Initial balance: 500  
  
Choose type:  
1) Checking (입출금)  
2) Savings (예금)  
3) Installment Savings (적금)  
  
Choice: 1  
Account successfully created.
```

Figure 6. Creating a bank account

AccountID	AccountType	Balance	Status	CreatedTime
1	Checking Account	500.00	Open	2025-12-08 21:44:07

Figure 7. Viewing created bank account

```

=====
TRANSACTION
=====
1) Deposit
2) Withdraw
3) Transfer

Choice: 3
Source Account ID: 1
Recipient Account ID: 2
Amount: 50
Description: transfer from checking to savings

Transfer successful!

```

Figure 8. Transferring money from one account to another

```

Choice: 4
TransactionID SourceAccountID RecipientAccountID Type Amount Description CreatedTime
-----
1 1 2 Transfer 50.00 transfer from checking to savings 2025-12-08 21:46:02

```

Figure 9. Viewing transaction

```

AccountID AccountType Balance Status CreatedTime
-----
1 Checking Account 450.00 Open 2025-12-08 21:44:07
2 Savings Account 1050.00 Open 2025-12-08 21:45:12

```

Figure 10. Viewing balance after transaction

```

Choice: 7
Enter Account ID to delete: 2
Account successfully deleted.

=====
USER MENU
=====
1) Create my account
2) View my accounts
3) Create my transaction
4) View my transactions
5) Create AutoTransfer
6) View my AutoTransfers
7) Delete my account
0) Logout

Choice: 2

AccountID AccountType Balance Status CreatedTime
-----
1 Checking Account 450.00 Open 2025-12-08 21:44:07

```

Figure 11. Deleting an account

----- ADMIN LOGIN -----

```

=== ADMIN LOGIN ===
Email: admin@gmail.com
Password:
Admin logged in: John

```

===== ADMIN MENU =====

```

1) List all ...
2) View a user's ...
0) Logout

```

Figure 12. Logging in as an Admin

```

=====
List all...
=====
1) List all users
2) List all accounts
3) List all transactions
4) List all autotransfers

Choice: 2
AccountID  FName  LName  AccountType  Balance  Status  CreatedTime
-----
1          Jimin  Hwang  Checking Account  450.00  Open    2025-12-08 21:44:07
3          Jane   Doe    Checking Account  1000.00 Open    2025-12-08 22:02:29

```

Figure 13. Listing all accounts in database (note: account#2 is removed from deleting the account in figure 11)

```

===== ADMIN MENU =====
=====
1) List all ...
2) View a user's ...
0) Logout

Choice: 2

=== USER SEARCH ===
Enter first name, last name, or email: jane

Found 1 user(s):

UserID  FName  LName  Email                PhoneNumber  CreatedTime
-----
2       Jane   Doe    janedoe@gmail.com    010-4321-8765  2025-12-08 21:59:48

Enter UserID to manage (0 = cancel): 2

===== VIEW USER (ID: 2, Name: Jane Doe) =====
=====
1) View user's accounts
2) View user's transactions
3) View user's autotransfers
0) Back

Choice: 1
AccountID  AccountType  Balance  Status  CreatedTime
-----
3          Checking Account  1000.00  Open    2025-12-08 22:02:29

```

Figure 14. Searching for a user and then checking their account list

IV. SQL Query Specifications

There are four main SQL queries are the following: INSERT, UPDATE, SELECT, DELETE. These queries are used in the following:

1. User

a. Create user

```
INSERT INTO user (FName, Minit, LName, Birthday, Email,
Password, PhoneNumber)
VALUES (%s,%s,%s,%s,%s,%s,%s,%s)
```

b. User login

```
SELECT *
FROM user
WHERE Email=%s AND Password=%s
```

2. Admin

a. Admin login

```
SELECT *
FROM admin
WHERE Email=%s AND Password=%s
```

b. Search for a user

```
SELECT *
FROM user
WHERE FName LIKE %s
      OR LName LIKE %s
      OR Email LIKE %s
```

3. Account

a. Create account

```
INSERT INTO account (UserID, AdminID, Balance, AccountType)
VALUES (%s, %s, %s, %s)
```

b. Delete account

```
DELETE FROM transaction
WHERE SourceAccountID=%s OR
RecipientAccountID=%sTransaction
```

```
DELETE FROM autotransfer
WHERE SourceAccountID=%s OR TargetAccountID=%s
```

```
DELETE FROM account
WHERE AccountID=%s
```

c. List user accounts

```
SELECT * FROM account
WHERE UserID = %s
ORDER BY AccountID
```

d. List all accounts


```
SELECT a.*, u.FName, u.LName
FROM account a
JOIN user u ON a.UserID = u.UserID
ORDER BY AccountID
```

4. Transaction

a. Check balance

```
SELECT Balance
FROM account
WHERE AccountID=%s FOR UPDATE
```

b. Update balance

```
UPDATE account
SET Balance=%s
WHERE AccountID=%s
```

c. Deposit

```
INSERT INTO transaction
(SourceAccountID, Type, Amount, Description)
VALUES (%s, 'Deposit', %s, %s)
```

d. Withdraw

```
INSERT INTO transaction
(SourceAccountID, Type, Amount, Description)
VALUES (%s, 'Withdraw', %s, %s)
```

e. Transfer

```
INSERT INTO transaction
(SourceAccountID, RecipientAccountID, Type, Amount,
Description)
VALUES (%s, %s, 'Transfer', %s, %s)
```

f. List user transactions

```
SELECT *
FROM transaction
WHERE SourceAccountID IN (
    SELECT AccountID FROM account WHERE UserID=%s
)
ORDER BY TransactionID DESC
```

g. List all transactions

```
SELECT *
FROM transaction
ORDER BY TransactionID DESC
```

5. Autotransfer

a. Create autotransfer

```
INSERT INTO autotransfer
(SourceAccountID, TargetAccountID, Amount, Frequency,
TransferDate)
VALUES (%s,%s,%s,%s,%s)
```

- b. List user autotransfers

```
SELECT at.*  
FROM autotransfer at  
JOIN account a ON at.SourceAccountID = a.AccountID  
WHERE a.UserID = %s  
ORDER BY AutoTransferID
```

- c. List all autotransfers

```
SELECT *  
FROM autotransfer  
ORDER BY AutoTransferID
```