

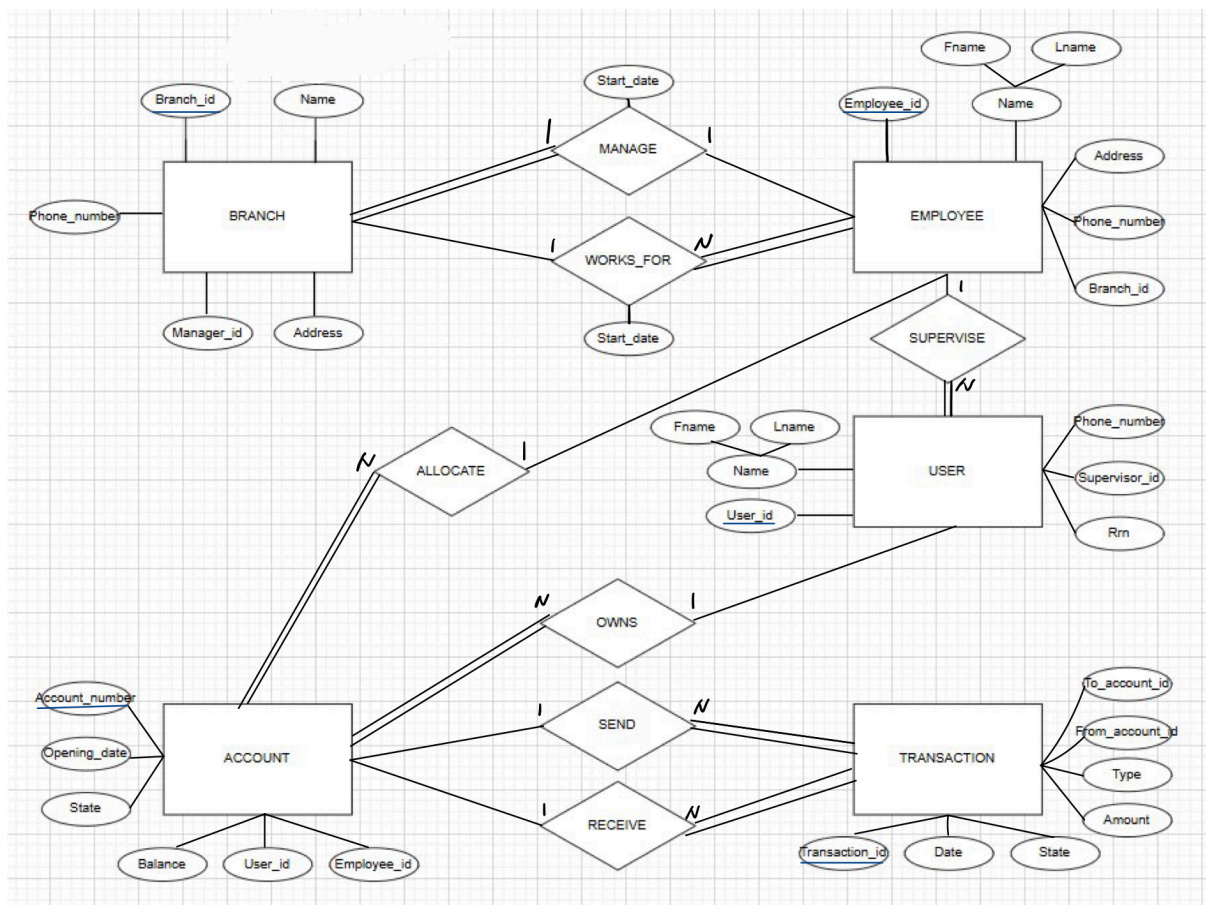
# [데이터베이스시스템] Project 2

이름: 권도현

학번: 2023065350

학과: 컴퓨터소프트웨어학부

## Concpetual DB Design using ER Model



## Entity

### 1. BRANCH

- **Simple attribute:** Branch\_id, Phone\_number, Address, Manager\_id
  - Manager\_id는 MANAGE Relationship의 attribute로, BRANCH Entity의 Simple attribute로 추가한 것이다.
- **Composite attribute:** Name

- Fname, Lname으로 나뉜다.

## 2. EMPLOYEE

- **Simple attribute: Employee\_id, Phone\_number, Address, Branch\_id**
  - Branch\_id는 WORKS\_FOR Relationship의 attribute로, EMPLOYEE Entity의 Simple attribute로 추가한 것이다.
- **Composite attribute: Name**
  - Fname, Lname으로 나뉜다.

## 3. USER

- **Simple attribute: User\_id, Phone\_number, Rrn, Supervisor\_id**
  - Rrn: 주민등록번호
  - Supervisor\_id는 SUPERVISE Relationship의 attribute로, User Entity의 Simple attribute로 추가한 것이다.
- **Composite attribute: Name**
  - Fname, Lname으로 나뉜다.

## 4. Account

- **Simple attribute: Account\_number, Balance, State, Opening\_data Rrn, Employee\_id, User\_id**
  - State: 계좌 상태
  - Balance: 잔액
  - Employee\_id는 ALLOCATE Relationship의 attribute로, Account Entity의 Simple attribute로 추가한 것이다.
  - User\_id는 OWNS Relationship의 attribute로, Account Entity의 Simple attribute로 추가한 것이다.

## 5. TRANSACTION

- **Simple attribute:** Transaction\_id, Date, state, Amount(거래 금액), Type(거래 유형), From\_account\_id, To\_account\_id
  - From\_account\_id는 SEND relationship의 attribute로, Transaction Entity의 Simple attribute로 추가한 것이다.
  - To\_account\_id는 RECIEVE relationship의 attribute로, Transaction Entity의 Simple attribute로 추가한 것이다.

## Relationship

### 1. MANAGE

- 한 명의 Employee가 하나의 Branch의 지점장임을 나타내는 Relationship
- Branch에는 반드시 Manager가 존재해야 하기 때문에 Branch쪽은 Total participation이다.
- **Requirements에 한 명의 관리자는 하나의 Branch만 관리할 수 있다는 가정을 추가해서 1:1 Relationship으로 설정했다.**
- Simple attribute: Start\_date

### 2. WORKS\_FOR

- 한 Branch에 속하는 Employee를 나타내는 Relationship
- Employee는 반드시 하나의 지점에 속하고, 하나의 Branch는 최대 20명의 Employee를 가질 수 있다.
  - 1:N Relationship
  - Employee 쪽은 Total participation이다.
- Simple attribute: Start\_date

### 3. SUPERVISEE

- Employee가 관리하는 User를 나타내는 Relationship
- 한 관리자는 최대 10명의 사용자를 관리할 수 있고, 한 사용자는 1명의 관리자에 의해서 관리되어야 한다.
  - 1:N Relationship
  - USER 쪽은 Total particiaption이다.

#### 4. ALLOCATE

- Employee가 관리하는 Account를 나타내기 위한 Relationship
- 한 Employee는 최대 100개의 Account를 관리할 수 있고 한 Account는 한 명의 Employee에 의해 관리되어야 한다.
  - 1:N relationship
  - Account쪽은 Total participation이다.

#### 5. OWNS

- User가 갖는 Account를 나타내기 위한 Relationship
- 한 User는 최대 100개의 Account를 가질 수 있고 한 Account는 한 명의 User에 의해 소유되어야 한다.
  - 1:N relationship
  - Account쪽은 Total participation이다.

#### 6. SEND

- Account에서 돈이 빠져나가는 Transaction을 나타내기 위한 Relationship
- Transaction은 반드시 Account를 통해서 발생해야 하고, Account 하나에 여러 개의 Transaction 내역이 저장될 수 있어야 한다.
  - 1:N relationship
  - Transaction쪽은 Total participation이다.

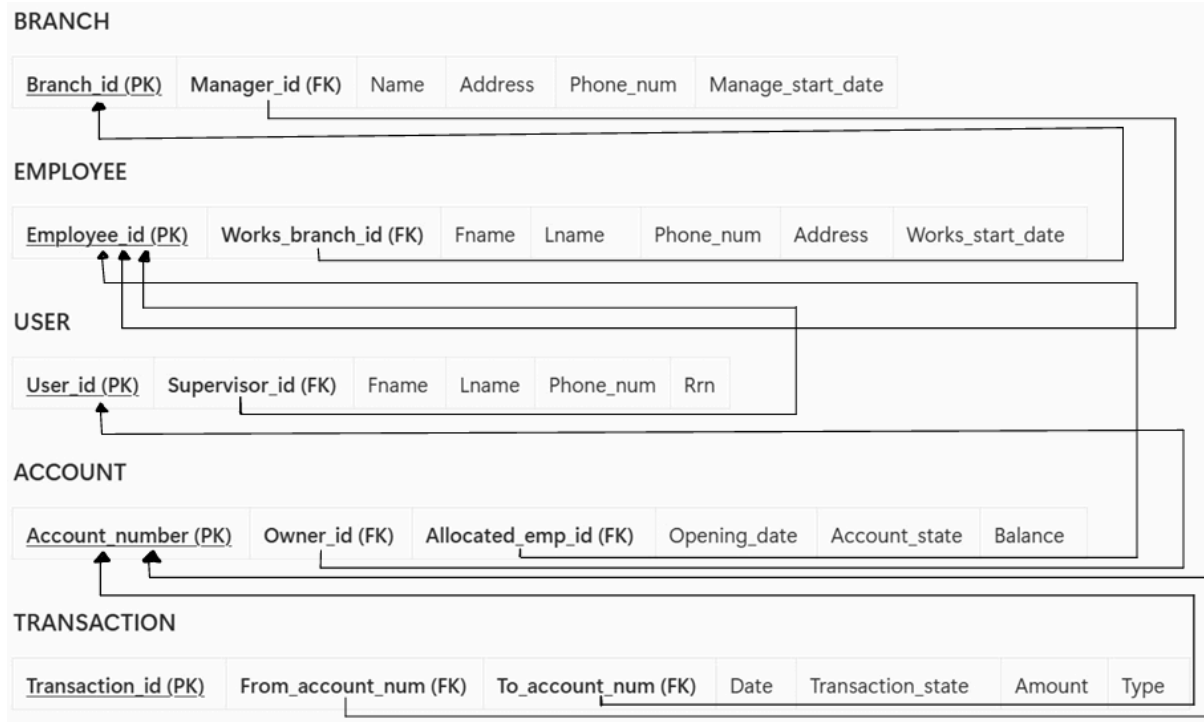
#### 7. RECIEVE

- Account로 돈이 들어오는 Transaction을 나타내기 위한 Relationship
- Transaction은 반드시 Account를 통해서 발생해야 하고, Account 하나에 여러 개의 Transaction 내역이 저장될 수 있어야 한다.
  - 1:N relationship
  - Transaction쪽은 Total participation이다.

## Constraints

1. **WORKS\_FOR** Relationship에서 BRANCH Entity 한 개는 최대 20명의 EMPLOYEE Entity와 Relationship을 맺을 수 있다.
2. **SUPERVISE** Relationship에서 EMPLOYEE Entity 한 개는 최대 10명의 USER Entity와 Relationship을 맺을 수 있다.
3. **ALLOCATE** Relationship에서 BRANCH Entity 한 개는 최대 100개의 ACCOUNT Entity와 Relationship을 맺을 수 있다.
4. **OWNS** Relationship에서 USER entity 한 개는 최대 10개의 ACCOUNT entity와 Relationship을 맺을 수 있다.
5. **ACCOUNT** Relationship에서 State attribute는 Open/Stop/Close 세 가지 중 하나를 값으로 가질 수 있다.
6. **ACCOUNT** Relationship에서 Balance attribute의 값은 0보다 작아질 수 없다.
7. **TRANSACTION** Relationship에서 Amount attribute의 값은 1,000,000을 초과할 수 없고 0보다 커야 한다.
8. **TRANSACTION** Relationship에서 Type attribute는 이체/출금/입금 세 가지 중 하나를 값으로 가질 수 있다.

## Logical DB Design using Relation Model



위의 ER diagram을 기반으로 Relational database schema를 만들었다.

- 변경 사항 없고, ER diagram을 Relational database schema로 변경하는 방법대로 변경하였다.
- MANAGE relationship (1:1)의 Simple attribute는 Total participation 쪽인 BRANCH Entity에 Simple attribute로 추가하였다.
  - Manage\_start\_date으로 Renaming했다.
- WORKS\_For relationship (1:N)의 Simple attribute는 N쪽 side이면서 Total participation 쪽인 EMPLOYEE Entity에 Simple attribute로 추가하였다.
  - Works\_start\_date으로 Renaming했다.
- EMPLOYEE entity와 USER entity의 Composite attribute인 Name은 Simple attribute 2개로 분해하여 원래의 Entity에 대응되는 Relation에 Simple attribute로 추가하였다.

Relational database schema로 변경하는 과정에서 중복되었을 때 문제가 될 만한 User\_id, Employee\_id, State 등은 Rename하였다.

- Works\_branch\_id (FK), Works\_branch\_id (FK), Allocated\_emp\_id (FK)
- Account\_state, Transaction state

**Constraint는 위와 동일하다.**