

프로젝트 2: Entity-Relationship (ER) 모델을 이용한 개념적 (conceptual) DB 설계

2015005078 정진교

프로젝트 1을 통해 다음과 같은 요구사항을 제출하였습니다.

- 개체 : 관리자, 사용자, 계좌, 입금, 출금, 대출
- 요구사항
 - ✓ 관리자의 입금 번호, 출금 번호, 대출 번호를 저장한다.
 - ✓ 관리자는 각 사용자의 계좌 입출금을 승인한다.
 - ✓ 관리자는 각 사용자의 대출을 승인한다.
 - ✓ 사용자의 이름, 주민등록번호, 성별, 주소, 생일, 급여, 대출을 저장한다.
 - 대출은 남은 대출 금액과 기간을 포함한다.
 - ✓ 사용자는 자신의 계좌를 n개까지 개설하거나 삭제한다.
 - ✓ 사용자는 자신의 계좌에 돈을 입금한다.
 - ✓ 사용자는 자신의 계좌에서 돈을 출금한다.
 - ✓ 사용자는 n명의 사용자들에게 돈을 송금한다.
 - ✓ 사용자는 관리자로부터 대출 받는다.
 - 대출금이 남아있다면 대출을 받을 수 없다.
 - 대출할 수 있는 금액은 사용자의 급여에 따라 달라진다.
 - ✓ 계좌의 계좌 번호, 카테고리, 잔액을 저장한다.
 - 카테고리는 보통예금, 정기예금, 적금을 포함한다.
 - 정기예금, 적금은 만기일, 이자를 포함한다.
 - ✓ 입금의 입금 번호, 입금 계좌, 입금할 금액을 저장한다.
 - ✓ 출금의 출금 번호, 출금 계좌, 출금할 금액을 저장한다.
 - ✓ 대출의 대출 번호, 대출 계좌, 대출기간, 대출금액, 대출이자를 저장한다.

하지만 ER 모델을 설계하면서 약간의 수정이 불가피하였습니다.

1. 관리자의 attribute를 이름, 주민등록번호, 성별, 주소, 생일, 급여로 바꾸었습니다.
2. 관리자가 각 사용자의 계좌 입금, 출금, 송금, 대출을 승인 대신 관리로 바꾸었습니다.
3. 계좌의 카테고리에서 적금을 삭제하였습니다.
4. 사용자의 attribute에서 대출을 빚으로 이름을 변경하였습니다.

따라서 수정된 요구사항은 다음과 같습니다.

- 개체 : 관리자, 사용자, 계좌, 입금, 출금, 대출
- 요구사항
 - ✓ 관리자의 이름, 주민등록번호, 성별, 주소, 생일, 급여를 저장한다.
 - ✓ 관리자는 각 사용자의 계좌 입금, 출금, 송금, 대출을 관리한다.
 - ✓ 사용자의 이름, 주민등록번호, 성별, 주소, 생일, 급여, 빚을 저장한다.
 - 빚은 남은 금액과 남은 기간을 포함한다.
 - ✓ 사용자는 자신의 계좌를 n개까지 개설하거나 삭제한다.
 - ✓ 사용자는 자신의 계좌에 돈을 입금한다.
 - ✓ 사용자는 자신의 계좌에서 돈을 출금한다.
 - ✓ 사용자는 n명의 사용자들에게 돈을 송금한다.
 - ✓ 사용자는 대출 받는다.
 - 빚이 남아있다면 대출을 받을 수 없다.
 - 대출할 수 있는 금액은 사용자의 급여에 따라 달라진다.
 - ✓ 계좌의 계좌 번호, 카테고리, 잔액을 저장한다.
 - 카테고리는 보통예금, 정기예금, 적금을 포함한다.
 - 정기예금, 적금은 만기일, 이자를 포함한다.
 - ✓ 입금의 입금 번호, 입금 계좌, 입금할 금액을 저장한다.
 - ✓ 출금의 출금 번호, 출금 계좌, 출금할 금액을 저장한다.
 - ✓ 대출의 대출 번호, 대출 계좌, 대출기간, 대출금액, 대출이자를 저장한다.

수정된 요구사항에 따른 ER 모델은 사진으로 첨부하였습니다.

1. 요구사항의 개체를 entity로 설정하였습니다.
2. 관리자 entity의 key attribute는 주민등록번호입니다.
3. 사용자 entity의 key attribute는 주민등록번호입니다.
4. 계좌 entity의 key attribute는 계좌 번호입니다.
5. 송금 entity의 key attribute는 송금 번호입니다.
6. 입금 entity의 key attribute는 입금 번호입니다.
7. 출금 entity의 key attribute는 출금 번호입니다.
8. 대출 entity의 key attribute는 대출 번호입니다.
9. 사용자가 계좌를 개설하는 것을 1:N relationship으로 설정하였습니다.
10. 사용자가 송금하는 것을 송금 신청 1:N relationship으로 설정하였습니다.
11. 사용자가 입금하는 것을 입금 신청 1:N relationship으로 설정하였습니다.
12. 사용자가 출금하는 것을 출금 신청 1:N relationship으로 설정하였습니다.
13. 사용자가 대출받는 것을 대출 신청 1:N relationship으로 설정하였습니다.
14. 관리자가 사용자의 입금을 관리하는 것을 입금 관리 1:N relationship으로 설정하였습니다.
15. 관리자가 사용자의 출금을 관리하는 것을 출금 관리 1:N relationship으로 설정하였습니다.
16. 관리자가 사용자의 송금을 관리하는 것을 송금 관리 1:N relationship으로 설정하였습니다.
17. 관리자가 사용자의 대출을 관리하는 것을 대출 관리 1:N relationship으로 설정하였습니다.

