

SQL(Oracle DBMS)

작업형 수행평가

(부산은행)금융 DT아카데미 개발자 양성과정

2025/08/13

이수연

목차

1. 요구사항

2. 요구사항 구현

1. ERD

2. TABLE

3. 수행평가 1~10번 문제 풀이

개요 및 요구사항 구현

1. 목적

- OO은행의 기본적인 은행 업무(계좌 관리, 고객 관리, 카드 관리 등)를 효율적으로 처리
- 고객과 내부 직원 간 원활한 업무 처리를 위해 데이터베이스를 구축.

2. 범위 및 요구사항

- 고객 정보 관리 | Customer
- 카드 정보 관리 | Card
- 예금 계좌 관리 | Account
- 거래 내역 관리 | Transaction

1. OO은행은 예금 및 카드 서비스를 고객에게 제공한다
2. 고객은 고객주민번호를 부여하여 식별하며 고객명, 고객 주소, 고객 생년월일, 고객 이메일, 고객전화번호, 고객직업의 정보를 가진다.
3. 카드는 카드번호를 부여하여 식별하며 카드 신청일자, 카드한도금액, 카드결제일자, 카드종류, 주민번호, 연동 예금계좌ID를 가진다.
4. 예금계좌는 예금계좌ID를 부여하여 식별하며 예금계좌종류, 예금잔고, 카드신청여부, 예금개설일자, 고객주민번호 정보를 가진다.
5. 예금거래내역은 거래번호를 부여하여 식별하며 예금계좌ID, 예금부분, 예금내용, 거래금액, 거래 날짜시간 정보를 가진다.

요구사항 구현



명세서 & ERD

Card

| 구분 | Logical | Physical | | | |
|-----|-----------|--------------------|-----------|------|----------|
| | | Column | Data type | Size | 기타 |
| | | | | | |
| 기본키 | 카드번호 | card_no | char | 14 | Not Null |
| 외래키 | 고객주민번호 | cust_jumin | char | 14 | |
| | 연동 예금계좌ID | acc_id | char | 14 | |
| 속성 | 카드신청일자 | card_register_date | date | | |
| | 카드한도금액 | card_limit | number | - | |
| | 카드결제일자 | card_approve_date | date | | |
| | 카드종류 | card_type | varchar | 10 | Not Null |

| | 카드번호 | 주민번호 | 연동 예금계좌ID | 카드신청일자 | 카드한도금액 | 카드결제일자 | 카드종류 |
|---|----------------|----------------|----------------|------------|-----------|------------|-------|
| 1 | 2111-1001-1001 | 760121-1234567 | 1011-1001-1001 | 2020-01-21 | 1,000,000 | 2020-02-10 | check |
| 2 | 2041-1001-1002 | 890530-1234567 | 1011-1001-1002 | 2020-06-11 | 3,000,000 | 2020-06-15 | check |
| 3 | 2011-1001-1003 | 790413-1234567 | 1011-1001-1003 | 2020-05-30 | 5,000,000 | 2020-06-25 | check |
| 4 | 2611-1001-1005 | 750611-1234567 | 1011-1002-1005 | 2020-09-12 | 1,500,000 | 2020-10-10 | check |

Customer

고객 테이블 명세서

| 개체 | Customer | | | | | |
|-----|----------|------------|-----------|------|----------|--|
| 구분 | Logical | Physical | | | | |
| | | Column | Data type | Size | 기타 | |
| 기본키 | 고객주민번호 | cust_jumin | char | 14 | Not Null | |
| 속성 | 이름 | name | varchar | 20 | Not Null | |
| | 주소 | address | varchar | 100 | Not Null | |
| | 생년월일 | birth | char | 10 | Not Null | |
| | 이메일(UK) | email | varchar | 100 | | |
| | 휴대폰(UK) | hp | varchar | 20 | Not Null | |
| | 직업 | job | varchar | 20 | | |

| | 주민번호 | 고객명 | 주소 | 생년월일 | 이메일 | 전화번호 | 직업 |
|---|----------------|-----|----|------------|---------------|---------------|-----|
| 1 | 760121-1234567 | 정우성 | 서울 | 1976-01-21 | - | 010-1101-7601 | 배우 |
| 2 | 750611-1234567 | 이정재 | 서울 | 1975-06-11 | - | 010-1102-7506 | 배우 |
| 3 | 890530-1234567 | 전지현 | 대전 | 1989-05-30 | ji@naver.com | 010-1103-8905 | 자영업 |
| 4 | 790413-1234567 | 이나영 | 대전 | 1979-04-13 | lee@naver.com | 010-2101-7904 | 회사원 |
| 5 | 660912-1234567 | 원빈 | 대전 | 1966-09-12 | one@daum.net | 010-2104-6609 | 배우 |

Account

| 구분 | Logical | Physical | | | |
|-----|---------|-------------------|-----------|------|----------|
| | | Column | Data type | Size | 기타 |
| | | | | | |
| 기본키 | 예금계좌ID | acc_id | char | 14 | Not Null |
| 외래키 | 고객주민번호 | cust_jumin | char | 14 | |
| | 예금계좌종류 | acc_type | varchar | 20 | Not Null |
| 속성 | 예금잔고 | acc_balance | number | - | Not Null |
| | 카드신청여부 | acc_card | char | 1 | Not Null |
| | 계좌개설일 | acc_register_date | datetime | | |

| | 예금계좌ID | 주민번호 | 예금계좌종류 | 예금잔고 | 카드신청여부 | 계좌개설일 |
|---|----------------|----------------|--------|-----------|--------|---------------------|
| 1 | 1011-1001-1001 | 760121-1234567 | 자유입출금 | 4,160,000 | Y | 2020-01-21 13:00:02 |
| 2 | 1011-1001-1002 | 890530-1234567 | 자유입출금 | 376,000 | Y | 2020-06-11 13:00:02 |
| 3 | 1011-1001-1003 | 790413-1234567 | 자유입출금 | 1,200,000 | Y | 2020-05-30 13:00:02 |
| 4 | 1011-2001-1004 | 660912-1234567 | 정기적금 | 1,000,000 | N | 2020-04-13 13:00:02 |
| 5 | 1011-1002-1005 | 750611-1234567 | 자유입출금 | 820,000 | Y | 2020-09-12 13:00:02 |

Transaction

예금계좌 거래내역 테이블 명세서

| 개체 | Transaction | | | | |
|-----|-------------|---------------|-----------|------|----------|
| 구분 | Logical | Physical | | | |
| | | Column | Data type | Size | 기타 |
| 기본키 | 거래번호 | trans_id | number | - | sequence |
| 외래키 | 예금계좌ID | trans_acc_id | char | 14 | |
| | 예금구분 | trans_type | varchar | 20 | |
| 속성 | 예금내용 | trans_message | varchar | 20 | |
| | 거래금액 | trans_money | number | | |
| | 거래날짜시간 | trans_date | date | | Not Null |

| | 거래번호 | 예금계좌ID | 예금구분 | 예금내용 | 거래금액 | 거래날짜시간 |
|---|----------------|--------|---------|-----------|---------------------|--------|
| 1 | 1011-1001-1001 | 입금 | 2월 정기급여 | 3,500,000 | 2020-02-10 12:36:12 | |
| 2 | 1011-1001-1003 | 출금 | ATM 출금 | 300,000 | 2020-02-10 12:37:21 | |
| 3 | 1011-1001-1002 | 입금 | 2월 급여 | 2,800,000 | 2020-02-10 12:38:21 | |
| 4 | 1011-1001-1001 | 출금 | 2월 공과금 | 116,200 | 2020-02-10 12:39:21 | |
| 5 | 1011-1002-1005 | 출금 | ATM 출금 | 50,000 | 2020-02-10 12:40:21 | |

| Customer | |
|--------------------------|-----------------------------|
| P | * cust_jumin CHAR (14 BYTE) |
| | * name VARCHAR2 (20 BYTE) |
| | * address VARCHAR2 (100) |
| | * birth CHAR (10 BYTE) |
| | * email VARCHAR2 (100 BYTE) |
| | * hp VARCHAR2 (20 BYTE) |
| | * job VARCHAR2 (20 BYTE) |
| Customer_PK (cust_jumin) | |

| Account | |
|----------------------------------|-------------------------------|
| P | * acc_id CHAR (14 BYTE) |
| F | * cust_jumin CHAR (14 BYTE) |
| | * acc_type VARCHAR2 (20 BYTE) |
| | * acc_balance NUMBER |
| | * acc_card CHAR (1 BYTE) |
| | * acc_register_date DATE |
| Account_PK (acc_id) | |
| Account_Customer_FK (cust_jumin) | |

| Transaction | |
|---------------------------------------|------------------------------------|
| P | * trans_id NUMBER |
| F | * trans_acc_id CHAR (14 BYTE) |
| | * trans_type VARCHAR2 (20 BYTE) |
| | * trans_message VARCHAR2 (20 BYTE) |
| | * trans_money NUMBER |
| | * trans_date DATE |
| Transaction_PK (trans_id) | |
| Transaction_Account_FK (trans_acc_id) | |

TABLE

```
Microsoft Windows [Version 10.0.19045.6093]
(c) Microsoft Corporation. All rights reserved.

C:\Users\#GGG>sqlplus

SQL>Plus: Release 21.0.0.0.0 - Production on 수 8월 13 09:22:31 2025
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

사용자명 인력: system
비밀번호 인력:
마지막 성공한 로그인 시간: 화 8월 12 2025 12:20:11 +09:00

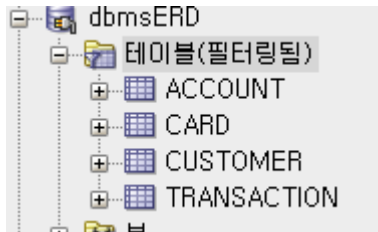
다음에 접속할:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0
```

```
SQL> alter session set "_ORACLE_SCRIPT"=true;

세션이 변경되었습니다.

SQL> create user dbmsERD identified by 1234;

사용자가 생성되었습니다.
```



```
CREATE TABLE account (
    acc_id          CHAR(14 BYTE) NOT NULL,
    cust_jumin      CHAR(14 BYTE),
    acc_type        VARCHAR2(20 BYTE) NOT NULL,
    acc_balance     NUMBER NOT NULL,
    acc_card        CHAR(1 BYTE) NOT NULL,
    acc_register_date DATE
);
```

```
ALTER TABLE account ADD CONSTRAINT account_pk PRIMARY KEY ( acc_id );
```

```
CREATE TABLE card (
    card_no          CHAR(14 BYTE) NOT NULL,
    cust_jumin      CHAR(14 BYTE),
    acc_id          CHAR(14 BYTE),
    card_register_date DATE,
    card_limit       NUMBER,
    card_approve_date DATE,
    card_type        VARCHAR2(10 BYTE) NOT NULL
);
```

```
ALTER TABLE card ADD CONSTRAINT card_pk PRIMARY KEY ( card_no );
```

```
CREATE TABLE customer (
    cust_jumin CHAR(14 BYTE) NOT NULL,
    name       VARCHAR2(20 BYTE) NOT NULL,
    address    VARCHAR2(100) NOT NULL,
    birth      CHAR(10 BYTE) NOT NULL,
    email      VARCHAR2(100 BYTE),
    hp         VARCHAR2(20 BYTE) NOT NULL,
    job        VARCHAR2(20 BYTE)
);
```

```
ALTER TABLE customer ADD CONSTRAINT customer_pk PRIMARY KEY ( cust_jumin );
```

```
CREATE TABLE transaction (
    trans_id          NUMBER NOT NULL,
    trans_acc_id     CHAR(14 BYTE) NOT NULL,
    trans_type        VARCHAR2(20 BYTE),
    trans_message     VARCHAR2(20 BYTE),
    trans_money       NUMBER,
    trans_date        DATE NOT NULL
);
```

```
ALTER TABLE transaction ADD CONSTRAINT transaction_pk PRIMARY KEY ( trans_id );
```

```
ALTER TABLE account
    ADD CONSTRAINT account_customer_fk FOREIGN KEY ( cust_jumin )
    REFERENCES customer ( cust_jumin );
```

```
ALTER TABLE card
    ADD CONSTRAINT card_account_fk FOREIGN KEY ( acc_id )
    REFERENCES account ( acc_id );
```

```
ALTER TABLE card
    ADD CONSTRAINT card_customer_fk FOREIGN KEY ( cust_jumin )
    REFERENCES customer ( cust_jumin );
```

```
ALTER TABLE transaction
    ADD CONSTRAINT transaction_account_fk FOREIGN KEY ( trans_acc_id )
    REFERENCES account ( acc_id );
```

```
CREATE SEQUENCE transaction_trans_id_seq START WITH 1 NOCACHE ORDER;
```

```
CREATE OR REPLACE TRIGGER transaction_trans_id_trg BEFORE
    INSERT ON transaction
    FOR EACH ROW
    WHEN ( new.trans_id IS NULL )
BEGIN
    :new.trans_id := transaction_trans_id_seq.nextval;
END;
```

TABLE

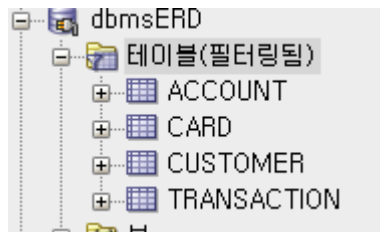
-- 데이터 입력

```
insert into customer VALUES ('760121-1234567', '정우성', '서울', '1976-01-21', null, '010-1101-7601', '배우');
insert into customer VALUES ('750611-1234567', '이정재', '서울', '1975-06-11', null, '010-1102-7506', '배우');
insert into customer VALUES ('890530-1234567', '전지현', '대전', '1989-05-30', 'jjh@naver.com', '010-1103-8905', '자영업');
insert into customer VALUES ('790413-1234567', '이나영', '대전', '1979-04-14', 'lee@naver.com', '010-2101-7904', '회사원');
insert into customer VALUES ('660912-1234567', '원빈', '대전', '1966-09-12', 'one@daum.net', '010-2104-6609', '배우');

insert into account VALUES ('1011-1001-1001', '760121-1234567', '자유입출금', 4160000, 'Y', '2020-01-21 13:00:02');
insert into account VALUES ('1011-1001-1002', '890530-1234567', '자유입출금', 376000, 'Y', '2020-06-11 13:00:02');
insert into account VALUES ('1011-1001-1003', '790413-1234567', '자유입출금', 1200000, 'Y', '2020-05-30 13:00:02');
insert into account VALUES ('1011-1001-1004', '660912-1234567', '자유입출금', 1000000, 'N', '2020-04-13 13:00:02');
insert into account VALUES ('1011-1001-1005', '750611-1234567', '자유입출금', 820000, 'Y', '2020-09-12 13:00:02');

insert into card values ('2111-1001-1001', '760121-1234567', '1011-1001-1001', '2020-01-21', 1000000, '2020-02-10', 'check');
insert into card values ('2041-1001-1002', '890530-1234567', '1011-1001-1002', '2020-06-11', 3000000, '2020-06-15', 'check');
insert into card values ('2011-1001-1003', '790413-1234567', '1011-1001-1003', '2020-05-30', 5000000, '2020-06-25', 'check');
insert into card values ('2611-1001-1005', '750611-1234567', '1011-1001-1005', '2020-09-12', 1500000, '2020-10-10', 'check');

insert into transaction VALUES (1, '1011-1001-1001', '입금', '2월 정기급여', 3500000, '2020-02-10 12:36:12');
insert into transaction VALUES (2, '1011-1001-1003', '출금', 'ATM 출금', 300000, '2020-02-10 12:37:12');
insert into transaction VALUES (3, '1011-1001-1002', '입금', '2월 급여', 2800000, '2020-02-10 12:38:12');
insert into transaction VALUES (4, '1011-1001-1001', '출금', '2월 공과금', 116200, '2020-02-10 12:39:12');
insert into transaction VALUES (5, '1011-1001-1005', '출금', 'ATM 출금', 50000, '2020-02-10 12:40:12');
```



SQL 문제 풀이

문제 1.

모든 고객 정보를 조회하시오.

```
--1.
select * from customer;
```

| | CUST_JUMIN | NAME | ADDRESS | BIRTH | EMAIL | HP | JOB |
|---|----------------|------|---------|------------|---------------|---------------|-----|
| 1 | 760121-1234567 | 정우성 | 서울 | 1976-01-21 | (null) | 010-1101-7601 | 배우 |
| 2 | 750611-1234567 | 이정재 | 서울 | 1975-06-11 | (null) | 010-1102-7506 | 배우 |
| 3 | 890530-1234567 | 전지현 | 대전 | 1989-05-30 | jjh@naver.com | 010-1103-8905 | 자영업 |
| 4 | 790413-1234567 | 이나영 | 대전 | 1979-04-14 | lee@naver.com | 010-2101-7904 | 회사원 |
| 5 | 660912-1234567 | 원빈 | 대전 | 1966-09-12 | one@daum.net | 010-2104-6609 | 배우 |

문제 2.

모든 카드 정보를 조회하시오.

```
--2.
select * from card;
```

| | CARD_NO | CUST_JUMIN | ACC_ID | CARD_REGISTER_DATE | CARD_LIMIT | CARD_APPROVE_DATE | CARD_TYPE |
|---|----------------|----------------|----------------|---------------------|------------|---------------------|-----------|
| 1 | 2111-1001-1001 | 760121-1234567 | 1011-1001-1001 | 2020-01-21 00:00:00 | 1000000 | 2020-02-10 00:00:00 | check |
| 2 | 2041-1001-1002 | 890530-1234567 | 1011-1001-1002 | 2020-06-11 00:00:00 | 3000000 | 2020-06-15 00:00:00 | check |
| 3 | 2011-1001-1003 | 790413-1234567 | 1011-1001-1003 | 2020-05-30 00:00:00 | 5000000 | 2020-06-25 00:00:00 | check |
| 4 | 2611-1001-1005 | 750611-1234567 | 1011-1001-1005 | 2020-09-12 00:00:00 | 1500000 | 2020-10-10 00:00:00 | check |

문제 3.

모든 예금계좌 정보를 조회하시오.

```
--3.
select * from account;
```

| | ACC_ID | CUST_JUMIN | ACC_TYPE | ACC_BALANCE | ACC_CARD | ACC_REGISTER_DATE |
|---|----------------|----------------|----------|-------------|----------|---------------------|
| 1 | 1011-1001-1001 | 760121-1234567 | 자유입출금 | 4160000 Y | | 2020-01-21 13:00:02 |
| 2 | 1011-1001-1002 | 890530-1234567 | 자유입출금 | 376000 Y | | 2020-06-11 13:00:02 |
| 3 | 1011-1001-1003 | 790413-1234567 | 자유입출금 | 1200000 Y | | 2020-05-30 13:00:02 |
| 4 | 1011-1001-1004 | 660912-1234567 | 자유입출금 | 1000000 N | | 2020-04-13 13:00:02 |
| 5 | 1011-1001-1005 | 750611-1234567 | 자유입출금 | 820000 Y | | 2020-09-12 13:00:02 |

SQL 문제 풀이

문제 4.

가장 최근 거래 내역 3건 조회하시오.

```
--4.
select *
from transaction
order by trans_date
fetch first 3 row ONLY;
```

| TRANS_ID | TRANS_ACC_ID | TRANS_TYPE | TRANS_MESSAGE | TRANS_MONEY | TRANS_DATE |
|----------|-----------------|------------|---------------|-------------|---------------------|
| 1 | 11011-1001-1001 | 입금 | 2월 정기급여 | 3500000 | 2020-02-10 12:36:12 |
| 2 | 21011-1001-1003 | 출금 | ATM 출금 | 300000 | 2020-02-10 12:37:12 |
| 3 | 31011-1001-1002 | 입금 | 2월 급여 | 2800000 | 2020-02-10 12:38:12 |

문제5.

카드한도금액이 200만원 이상인 고객의 이름과
카드종류 조회하시오.

```
--5.
select
    b.name as 고객이름,
    a.card_type as 카드종류
from card a
join customer b on a.cust_jumin = b.cust_jumin
where card_limit >= 2000000;
```

| | 고객이름 | 카드종류 |
|---|------|-------|
| 1 | 전지현 | check |
| 2 | 이나영 | check |

SQL 문제 풀이

문제 6.

예금계좌별 거래 건수 조회하시오.

```
-- 6.
select
    a.acc_id as 예금계좌,
    count(trans_acc_id) as "거래 건수"
from account a
join transaction b on a.acc_id = b.trans_acc_id
group by a.acc_id;
```

| 예금계좌 | 거래 건수 |
|------------------|-------|
| 1 1011-1001-1001 | 2 |
| 2 1011-1001-1003 | 1 |
| 3 1011-1001-1002 | 1 |
| 4 1011-1001-1005 | 1 |

문제7.

거래금액이 100만원 이상인 거래 내역 조회하시오.
(최근 거래순)

```
--7.
SELECT *
from transaction
where trans_money >= 1000000
order by trans_date;
```

| TRANS_ID | TRANS_ACC_ID | TRANS_TYPE | TRANS_MESSAGE | TRANS_MONEY | TRANS_DATE |
|----------|------------------|------------|---------------|-------------|---------------------|
| 1 | 1 1011-1001-1001 | 입금 | 2월 정기급여 | 3500000 | 2020-02-10 12:36:12 |
| 2 | 3 1011-1001-1002 | 입금 | 2월 급여 | 2800000 | 2020-02-10 12:38:12 |

SQL 문제 풀이

문제 8.

계좌와 연결된 카드 정보 조회(계좌ID, 카드ID, 카드종류) 하시오.

```
--8.
SELECT
    a.acc_id as 계좌ID,
    b.card_no as 카드ID,
    b.card_type as 카드종류
from account a
join card b on a.acc_id = b.acc_id;
```

질의 결과 x

SQL | 인출된 모든 행: 4(0,002초)

| | 계좌ID | 카드ID | 카드종류 |
|---|----------------|----------------|-------|
| 1 | 1011-1001-1001 | 2111-1001-1001 | check |
| 2 | 1011-1001-1002 | 2041-1001-1002 | check |
| 3 | 1011-1001-1003 | 2011-1001-1003 | check |
| 4 | 1011-1001-1005 | 2611-1001-1005 | check |

문제9.

예금구분이 '입금'인거래의 총합 조회하시오.

```
--9.
SELECT
    sum(trans_money) as "입금 거래의 총합"
from transaction
where trans_type = '입금';
```

스크립트 출력 x

질의 결과 x

SQL | 인출된 모든 행: 1(0,002초)

| | 입금 거래의 총합 |
|---|-----------|
| 1 | 6300000 |

SQL 문제 풀이

문제 10.

예금잔고가 4,000,000원 이상 고객에 대한 고객명, 주민번호, 전화번호, 주소를 조회하시오.

```
--10.
SELECT
    a.name as 고객명,
    a.cust_jumin as 주민번호,
    a.hp as 전화번호,
    a.address as 주소
from customer a
join account b on a.cust_jumin = b.cust_jumin
where b.acc_balance >= 4000000;
```

스크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 1(0.001초)

| | 고객명 | 주민번호 | 전화번호 | 주소 |
|---|-----|----------------|---------------|----|
| 1 | 정우성 | 760121-1234567 | 010-1101-7601 | 서울 |

감사합니다 😊