

<데이터베이스 프로젝트 수행보고서>

컴퓨터인공지능학부
202312720 안수빈

<목차>

- 1) 프로젝트 주제 설명(목적, 필요성, 기대효과 등)
- 2) 요구 수집 목록 및 수집한 별첨
- 3) 요구 분석 내용
- 4) 개념적 설계 내용
- 5) 논리적 설계 내용
- 6) 물리적 설계 내용
- 7) 테이블 생성 및 초기 데이터 삽입, 테이블 삭제 스크립트 파일 내용
- 8) 데이터 조작을 위한 스크립트 파일 내용 및 각 스크립트 실행 화면 캡처

1) 프로젝트 주제 설명(목적, 필요성, 기대효과 등)

주제: 애견인을 위한 정보 제공 및 애견 용품 거래 기능을 통합한 웹사이트의 데이터베이스를 구축하는 것이 목표입니다. 해당 시스템은 강아지 분양, 애견 카페 및 동물병원 정보, 질병 정보, DIY 콘텐츠, 직거래·공동구매·품앗이 기능 등 다양한 서비스를 데이터베이스를 통해 체계적으로 관리하고자 합니다.

목적: 반려동물을 가족처럼 여기는 인구가 증가함에 따라, 애견 관련 정보와 커뮤니티 활동, 상품 거래 등을 통합적으로 제공할 수 있는 플랫폼의 필요성이 커지고 있습니다. 이러한 수요를 충족시키기 위해, 웹 기반 애견 플랫폼의 데이터베이스를 설계·구현함으로써 안정적이고 효율적인 정보 제공 및 서비스 운영이 가능하도록 하는 것을 목적으로 합니다.

필요성: 현재 애견 관련 정보는 인터넷 및 커뮤니티에 분산되어 있으며, 상품 거래나 질병 정보 등도 체계적으로 제공되지 않는 경우가 많습니다. 이로 인해 사용자들은 필요한 정보를 얻기 위해 여러 플랫폼을 이용해야 하며, 이는 시간과 노력을 요구합니다. 또한, 직거래나 공동구매 등의 커뮤니티 기반 기능은 신뢰성과 시스템적 지원이 부족한 실정입니다. 이러한 문제를 해소하고자, 데이터 중심의 통합형 애견 플랫폼 구축하게 되었습니다.

기대효과:

1. 지역 기반의 애견 카페 및 동물병원 정보 제공으로 사용자 접근성 향상
2. 질병 및 DIY 콘텐츠 제공을 통한 애견 돌봄 정보의 질적 향상
3. 회원 간 직거래 및 공동구매 기능을 통한 자율적 커머스 생태계 조성
4. 친구 매칭, 품앗이, 훈련 프로그램 등 커뮤니티 활성화를 통한 사회적 가치 실현
5. 데이터베이스 기반의 통합 관리 시스템을 통해 신속하고 정확한 정보 제공 가능

2) 요구 수집 목록 및 수집한 별첨

요구 수집 방법: 본 프로젝트의 데이터베이스 구축을 위해 다양한 경로에서 요구사항을 수집 하였습니다. 주요 수집 방법으로는 애견 관련 커뮤니티 조사, 애견 앱 사용자의 불만 사항 분석, 기존 웹사이트 분석 등이 있습니다. 이를 통해 애견인들이 필요로 하는 정보와 서비스 기능을 도출하였습니다.

주요 요구 목록:

1. 회원 관리
 - 회원 가입 및 정보 관리
 - 회원별 애견 정보 등록 및 관리
2. 강아지 분양 정보 제공
 - 품종, 나이, 건강 상태 등 상세 분양 정보
 - 분양 일정 및 연락처 제공
3. 지역 기반 정보 제공
 - 애견 카페 위치 및 이용 정보
 - 동물병원 위치, 진료과목, 운영 시간 등 정보
4. 강아지 질병 정보
 - 주요 질병 목록 및 증상, 치료법 안내
 - 예방 및 관리 방법 제공
5. 애견 용품 거래 기능
 - 회원 간 직거래 시스템
 - 공동 구매 기능 지원
 - 거래 내역 관리
6. DIY 애견 용품 정보 제공
 - 집, 방석, 매트, 간식, 옷 등 제작 방법
 - 관련 재료 및 도구 안내
7. 커뮤니티 기능
 - 강아지 친구 만들기 매칭 서비스
 - 애견 돌봄 품앗이 운영 지원
 - 애견 훈련 프로그램 안내 및 신청

별첨 자료:

요구 수집을 위해 참고한 기존 반려동물 앱 사용자의 불만 사항

- 믹스 견종에 대한 상세 정보 및 분류 항목이 부족하여 사용자가 정확한 정보를 얻기 어려움
- 지역별 애견 카페 및 동물병원 정보가 최신 상태로 유지되지 않아 방문 시 혼란 발생
- 애견 질병 관련 정보가 전문적이지 않고 범위가 좁아, 초보 애견인들이 이해하기 어려움
- 직거래 및 공동구매 기능 사용 시 신뢰할 수 있는 거래 보장 시스템 부재로 이용자 불안감 존재
- DIY 애견 용품 제작 정보가 한정적이고, 다양한 난이도별 자료가 부족함
- 강아지 친구 매칭이나 돌봄 품앗이 기능이 활성화되어 있지 않아 커뮤니티 기능 활용도 저조
- 모바일 환경 최적화가 미흡하여 앱 사용 중 불편함 호소 사례 다수 보고됨

3) 요구 분석 내용

3-1. 데이터 요구 분석

데이터 항목	설명	속성
회원정보	서비스 이용 회원의 개인 및 애견 정보 관리	회원ID, 이름, 연락처, 주소, 애견이름, 품종, 나이, 건강상태
강아지 분양 정보	분양 가능한 강아지에 대한 상세 정보	분양ID, 품종, 나이, 성별, 건강 상태, 가격, 분양 일정, 연락처
애견 카페 정보	지역별 애견 카페 정보	카페ID, 이름, 위치(주소, 지역), 운영 시간, 전화번호, 리뷰
동물병원 정보	지역별 동물병원 정보	병원ID, 이름, 위치, 진료 과목, 운영 시간, 연락처
질병 정보	강아지 질병 관련 상세 정보	질병ID, 질병명, 증상, 원인, 치료법, 예방 방법
애견 용품 정보	판매 및 공동구매 가능한 애견 용품 정보	용품ID, 종류(집, 간식, 옷 등), 가격, 재고, 판매자ID
DIY 정보	DIY 애견 용품 제작 관련 정보	DIY ID, 제목, 내용, 난이도, 필요 재료 목록
친구 매칭 정보	강아지 친구 만들기 기능을 위한 매칭 데이터	회원ID, 강아지 특성, 선호 조건, 매칭 결과
돌봄 품앗이 정보	돌봄 서비스 참여 및 일정 정보	품앗이ID, 참여 회원ID, 일정, 서비스 내용
훈련 프로그램 정보	강아지 훈련 관련 프로그램 정보	프로그램ID, 제목, 설명, 일정, 신청 회원ID

3-2. 트랜잭션 요구 분석

트랜잭션 종류	설명	관련 데이터 및 처리 내용
회원 가입 및 정보 수정	회원이 서비스에 가입하거나 개인정보를 변경함	회원 정보 입력, 수정, 검증
강아지 분양 정보 조회	회원이 분양 가능한 강아지 목록을 조회함	분양 정보 검색, 필터링(품종, 나이 등)
애견 카페/병원 정보 조회	지역별 애견 카페 및 동물병원 정보 조회	위치 기반 조회, 운영 시간 및 리뷰 정보 제공
질병 정보 조회	강아지 질병 상세 정보 조회	질병명 검색, 증상 및 치료법 조회
애견 용품 거래	회원 간 직거래 및 공동구매 신청 및 처리	상품 등록, 주문, 결제, 재고 관리
DIY 정보 조회	DIY 애견 용품 제작 정보 열람	난이도별, 종류별 DIY 콘텐츠 조회
친구 매칭 신청 및 관리	강아지 친구 만들기 신청 및	회원 매칭 신청, 조건 비교,

	매칭	매칭 결과 통보
돌봄 품앗이 일정 등록 및 관리	돌봄 서비스 일정 등록, 신청 및 관리	일정 등록, 참여자 확인 및 일정 변경
훈련 프로그램 신청 및 운영	강아지 훈련 프로그램 신청 및 관리	프로그램 신청, 참가자 명단 관리, 일정 안내

3-3. 요구 분석 요약

이 데이터베이스에서는 애견인의 편의성을 극대화하기 위해 다양한 데이터 항목을 통합 관리하고, 각종 조회 및 거래, 커뮤니티 기능을 트랜잭션으로 처리할 수 있어야 합니다. 데이터 무결성 유지와 신속한 서비스 제공을 위해 각 트랜잭션은 적절한 동시성 제어와 보안 정책을 동반해야 합니다.

4) 개념적 설계 내용

4-1. 개체 후보 도출

요구사항을 바탕으로 주요 개체(Entity)를 후보로 뽑기

개체명	설명
회원(Member)	서비스 이용자 정보 관리
강아지(Dog)	회원이 등록한 강아지 정보
분양정보(Adoption)	분양 가능한 강아지 정보
애견카페(DogCafe)	지역별 애견 카페 정보
동물병원(Hospital)	지역별 동물병원 정보
질병(Disease)	강아지 질병 상세 정보
용품(Product)	애견 용품 정보 및 거래 관련 데이터
DIY정보(DIY)	DIY 애견 용품 제작 정보
친구매칭(Match)	강아지 친구 매칭 기록
돌봄품앗이(CareGroup)	애견 돌봄 품앗이 일정 및 참여 정보
훈련프로그램(Training)	애견 훈련 프로그램 정보
거래(Transaction)	애견 용품 직거래 및 공동구매 거래 기록

4-2. 개체 후보 검토

- 회원(Member): 서비스 핵심 사용자인 회원 정보는 반드시 필요.
- 강아지(Dog): 회원이 키우는 강아지별 정보 관리 필요.
- 분양정보(Adoption): 강아지 분양 정보는 회원과는 별도로 분리 관리.
- 애견카페/동물병원: 지역기반 장소 정보, 조회 및 관리 용도.
- 질병(Disease): 강아지 질병 정보 제공, 상세 설명 필수.
- 용품(Product)과 거래(Transaction): 상품 관리 및 거래 이력 기록을 위해 분리 필요.
- DIY정보(DIY): 별도의 콘텐츠 관리용 개체.
- 친구매칭/돌봄품앗이/훈련프로그램: 각각 커뮤니티 기능 관련 개체로 분리해 관리.

4-3. 개체 및 관계 명세서

개체명	주요 속성	관계 및 설명
회원(Member)	회원ID(PK), 이름, 연락처, 이메일	1:N 관계로 강아지 보유 (Member - Dog)
강아지(Dog)	강아지ID(PK), 이름, 품종, 생년월일	N:1 관계로 회원에 소속
분양정보(Adoption)	분양ID(PK), 품종, 나이, 상태	분양가능 강아지 정보를 별도 관리
애견카페(DogCafe)	카페ID(PK), 이름, 위치, 운영시간	지역 정보에 따른 조회 가능
동물병원(Hospital)	병원ID(PK), 이름, 진료과목, 위치	지역기반 진료정보 제공
질병(Disease)	질병ID(PK), 이름, 증상	애견 건강정보 제공

	치료법	
용품(Product)	용품ID(PK), 이름, 종류, 가격, 재고	거래 기록과 연동
거래(Transaction)	거래ID(PK), 판매자ID(FK), 구매자ID(FK), 용품ID(FK), 수량, 거래일	회원 간 용품 거래 기록
DIY정보(DIY)	DIY_ID(PK), 제목, 내용, 난이도	DIY 콘텐츠 제공
친구매칭(Match)	매칭ID(PK), 회원ID1(FK), 회원ID2(FK), 매칭일, 상태	강아지 친구 매칭 기록
돌봄품앗이(CareGroup)	품앗이ID(PK), 일정, 참여회원ID(FK), 내용	돌봄 서비스 관리
훈련프로그램(Training)	프로그램ID(PK), 제목, 일정, 설명	강아지 훈련 프로그램 정보

5) 논리적 설계 내용

5-1. 개체 간 관계 명세

관계	관계 형태	관계 설명
회원(Member) - 강아지(Dog)	1:N	한 명의 회원은 여러 마리의 강아지를 등록할 수 있음. → 회원(부모), 강아지(자식)
회원(Member) - 거래(Transaction)	1:N	한 회원은 여러 건의 판매자/구매자로 참여 가능. 판매자ID, 구매자ID 모두 Member를 참조함.
용품(Product) - 거래(Transaction)	1:N	하나의 용품은 여러 건의 거래에 포함될 수 있음. → 용품(부모), 거래(자식)
회원(Member) - 친구매칭(Match)	N:M	한 회원은 여러 명과 친구가 될 수 있고, 한 친구 매칭에 2명의 회원이 참여함. (연결 테이블 형태 Match 사용)
회원(Member) - 돌봄품앗이(CareGroup)	1:N	한 회원은 여러 개의 돌봄 일정에 참여 가능. → 회원(부모), 돌봄품앗이(자식)
훈련프로그램(Training) - 회원(Member)	N:M	여러 회원이 하나의 훈련 프로그램에 신청 가능하고, 한 회원이 여러 프로그램에 참여 가능. (→ 별도 신청 테이블 필요할 수 있음)
강아지(Dog) - 질병(Disease)	N:M (선택적)	한 강아지가 여러 질병을 겪을 수 있고, 한 질병이 여러 강아지에게 해당될 수 있음. (→ 진단 또는 이력 테이블 필요)
애견카페(DogCafe), 동물병원(Hospital)	-	독립적으로 지역 기반 정보 제공용 테이블 (회원과 직접적인 외래키 관계는 없음)
분양정보(Adoption) - 강아지(Dog)	선택적 1:1 또는 1:N	한 분양 건에 강아지 1마리 / 또는 다수 강아지 포함 가능. 구조에 따라 선택 (간단하게 하려면 1:1)
DIY정보(DIY)	-	단독 콘텐츠 테이블로 독립적 사용 (회원과 연결 필요 시 작성자ID 속성 추가 가능)

→ Transaction(거래ID, 판매자ID, 구매자ID, 용품ID, 수량, 거래일)로 3NF 만족

5-4. 릴레이션 스키마 정의

Member(회원ID PK, 이름, 연락처, 이메일)

Dog(강아지ID PK, 이름, 품종, 생년월일, 회원ID FK)

Adoption(분양ID PK, 품종, 나이, 상태, 강아지ID FK)

DogCafe(카페ID PK, 이름, 위치, 운영시간)

Hospital(병원ID PK, 이름, 진료과목, 위치)

Disease(질병ID PK, 이름, 증상, 치료법)

Product(용품ID PK, 이름, 종류, 가격, 재고)

Transaction(거래ID PK, 판매자ID FK, 구매자ID FK, 용품ID FK, 수량, 거래일)

DIY(DIY_ID PK, 제목, 내용, 난이도)

Match(매칭ID PK, 회원ID1 FK, 회원ID2 FK, 매칭일, 상태)

CareGroup(품앗이ID PK, 일정, 참여회원ID FK, 내용)

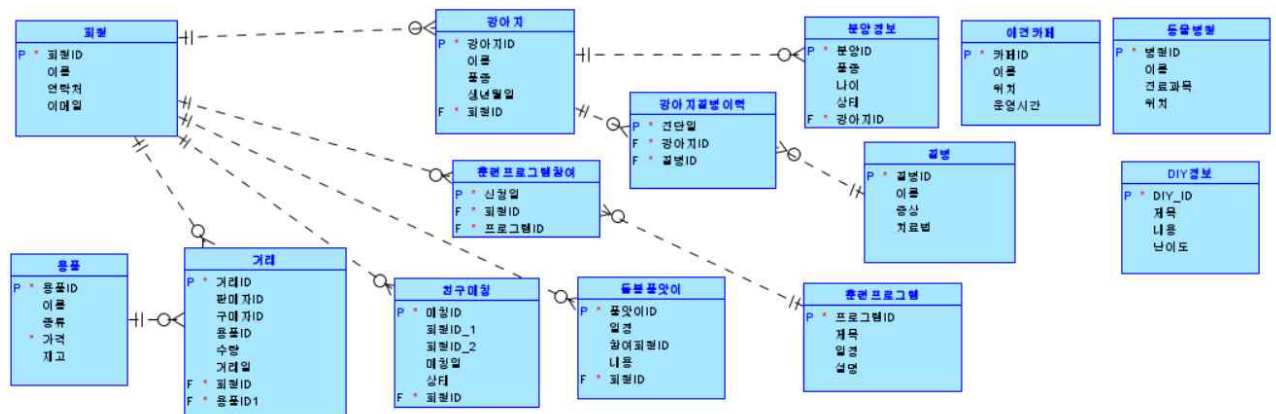
Training(프로그램ID PK, 제목, 일정, 설명)

(선택적으로 추가)

Dog_Disease(강아지ID FK, 질병ID FK, 진단일 PK) -- 강아지-질병 N:M 관계 해결용 연결 테이블

TrainingApply(회원ID FK, 프로그램ID FK, 신청일 PK) -- 회원-훈련 N:M 관계 해결용 연결 테이블

<정규화 후 최종 논리적 ERD>

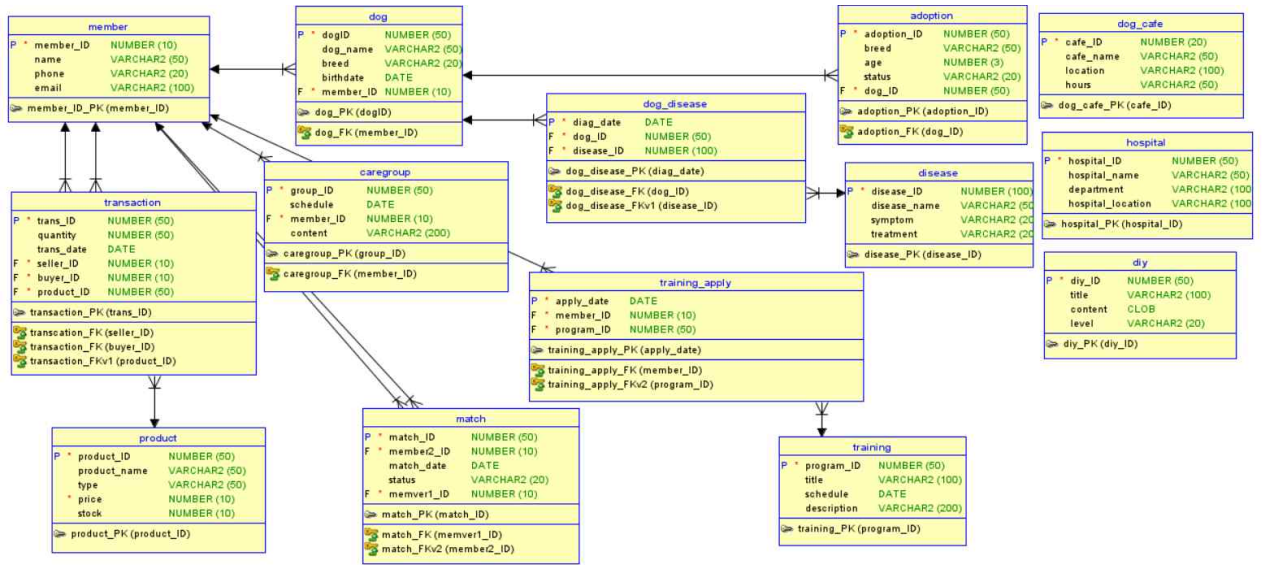


5-5. 무결성 제약 조건 정의

릴레이션	무결성 제약 조건
Member	회원ID는 기본키, 중복 불가, 이메일은 유일 제약 가능
Dog	회원ID는 외래키 → Member의 회원ID 참조, 삭제 시 cascade 금지
Transaction	판매자ID, 구매자ID는 모두 Member 참조 (FK), 용품ID는 Product 참조
Match	회원ID1 ≠ 회원ID2, 중복 매칭 방지를 위한 UNIQUE 제약 고려
Product	재고는 0 이상 정수, 가격은 음수 불가 (check 제약)
Dog_Disease	복합 기본키: (강아지ID, 질병ID, 진단일)
TrainingApply	복합 기본키: (회원ID, 프로그램ID), 중복 신청 방지

6) 물리적 설계 내용

<물리적 ERD>



7) 테이블 생성 및 초기 데이터 삽입, 테이블 삭제 스크립트 파일 내용

(파일 따로 첨부하였습니다.)

-- 1. Member

```
CREATE TABLE Member (  
    회원ID NUMBER PRIMARY KEY,  
    이름 VARCHAR2(50),  
    연락처 VARCHAR2(20),  
    이메일 VARCHAR2(100)  
);
```

```
INSERT INTO Member VALUES (1, '김철수', '010-1234-5678', 'cs@example.com');  
INSERT INTO Member VALUES (2, '이영희', '010-2345-6789', 'yh@example.com');  
INSERT INTO Member VALUES (3, '박민수', '010-3456-7890', 'ms@example.com');  
INSERT INTO Member VALUES (4, '최지현', '010-4567-8901', 'jh@example.com');  
INSERT INTO Member VALUES (5, '정우성', '010-5678-9012', 'ws@example.com');
```

-- 2. Dog

```
CREATE TABLE Dog (  
    강아지ID NUMBER PRIMARY KEY,  
    이름 VARCHAR2(50),  
    품종 VARCHAR2(50),  
    생년월일 DATE,  
    회원ID NUMBER,  
    FOREIGN KEY (회원ID) REFERENCES Member(회원ID)  
);
```

```
INSERT INTO Dog VALUES (101, '콩이', '푸들',  
    TO_DATE('2020-05-01','YYYY-MM-DD'), 1);  
INSERT INTO Dog VALUES (102, '보리', '말티즈',  
    TO_DATE('2019-07-15','YYYY-MM-DD'), 1);  
INSERT INTO Dog VALUES (103, '초코', '비숱',  
    TO_DATE('2021-03-10','YYYY-MM-DD'), 2);  
INSERT INTO Dog VALUES (104, '두부', '시츄',  
    TO_DATE('2020-11-22','YYYY-MM-DD'), 3);  
INSERT INTO Dog VALUES (105, '밤이', '진돗개',  
    TO_DATE('2018-08-30','YYYY-MM-DD'), 4);
```

-- 3. Adoption

```
CREATE TABLE Adoption (  
    분양ID NUMBER PRIMARY KEY,
```

```
    품종 VARCHAR2(50),
    나이 NUMBER,
    상태 VARCHAR2(20),
    강아지ID NUMBER,
    FOREIGN KEY (강아지ID) REFERENCES Dog(강아지ID)
);
```

```
INSERT INTO Adoption VALUES (201, '푸들', 3, '가능', 101);
INSERT INTO Adoption VALUES (202, '비송', 2, '완료', 103);
INSERT INTO Adoption VALUES (203, '진돗개', 5, '가능', 105);
INSERT INTO Adoption VALUES (204, '말티즈', 4, '대기', 102);
INSERT INTO Adoption VALUES (205, '시츄', 3, '가능', 104);
```

-- 4. DogCafe

```
CREATE TABLE DogCafe (
    카페ID NUMBER PRIMARY KEY,
    이름 VARCHAR2(100),
    위치 VARCHAR2(100),
    운영시간 VARCHAR2(50)
);
```

```
INSERT INTO DogCafe VALUES (301, '멍멍카페', '서울', '10:00-20:00');
INSERT INTO DogCafe VALUES (302, '강아지놀이터', '부산', '09:00-18:00');
INSERT INTO DogCafe VALUES (303, '댕댕하우스', '대구', '11:00-21:00');
INSERT INTO DogCafe VALUES (304, '퍼피카페', '광주', '10:00-22:00');
INSERT INTO DogCafe VALUES (305, '도그존', '대전', '09:30-19:00');
```

-- 5. Hospital

```
CREATE TABLE Hospital (
    병원ID NUMBER PRIMARY KEY,
    이름 VARCHAR2(100),
    진료과목 VARCHAR2(50),
    위치 VARCHAR2(100)
);
```

```
INSERT INTO Hospital VALUES (401, '24시동물병원', '내과', '서울');
INSERT INTO Hospital VALUES (402, '펫케어', '외과', '부산');
INSERT INTO Hospital VALUES (403, '멍멍병원', '피부과', '대구');
INSERT INTO Hospital VALUES (404, '러브펫', '종합', '광주');
INSERT INTO Hospital VALUES (405, '헬로동물병원', '정형외과', '대전');
```

-- 6. Disease

```
CREATE TABLE Disease (  
    질병ID NUMBER PRIMARY KEY,  
    이름 VARCHAR2(50),  
    증상 VARCHAR2(100),  
    치료법 VARCHAR2(100)  
);
```

```
INSERT INTO Disease VALUES (501, '피부병', '가려움', '연고');  
INSERT INTO Disease VALUES (502, '감기', '기침', '약물치료');  
INSERT INTO Disease VALUES (503, '장염', '설사', '식이요법');  
INSERT INTO Disease VALUES (504, '구토', '구토', '수액');  
INSERT INTO Disease VALUES (505, '탈수', '무기력', '수분섭취');
```

-- 7. Product

```
CREATE TABLE Product (  
    용품ID NUMBER PRIMARY KEY,  
    이름 VARCHAR2(100),  
    종류 VARCHAR2(50),  
    가격 NUMBER,  
    재고 NUMBER  
);
```

```
INSERT INTO Product VALUES (601, '리드줄', '외출', 12000, 20);  
INSERT INTO Product VALUES (602, '강아지간식', '식품', 5000, 50);  
INSERT INTO Product VALUES (603, '방석', '생활', 15000, 15);  
INSERT INTO Product VALUES (604, '장난감', '놀이', 7000, 30);  
INSERT INTO Product VALUES (605, '삼푸', '위생', 9000, 25);
```

-- 8. Transaction

```
CREATE TABLE Transaction (  
    거래ID NUMBER PRIMARY KEY,  
    판매자ID NUMBER,  
    구매자ID NUMBER,  
    용품ID NUMBER,  
    수량 NUMBER,  
    거래일 DATE,  
    FOREIGN KEY (판매자ID) REFERENCES Member(회원ID),  
    FOREIGN KEY (구매자ID) REFERENCES Member(회원ID),  
    FOREIGN KEY (용품ID) REFERENCES Product(용품ID)  
);
```



```

INSERT INTO Transaction VALUES (701, 1, 3, 601, 1,
TO_DATE('2024-01-01','YYYY-MM-DD'));
INSERT INTO Transaction VALUES (702, 2, 1, 602, 2,
TO_DATE('2024-02-14','YYYY-MM-DD'));
INSERT INTO Transaction VALUES (703, 2, 5, 603, 1,
TO_DATE('2024-03-03','YYYY-MM-DD'));
INSERT INTO Transaction VALUES (704, 4, 2, 604, 3,
TO_DATE('2024-03-20','YYYY-MM-DD'));
INSERT INTO Transaction VALUES (705, 5, 3, 605, 2,
TO_DATE('2024-04-01','YYYY-MM-DD'));

```

-- 9. DIY

```

CREATE TABLE DIY (
    DIY_ID NUMBER PRIMARY KEY,
    제목 VARCHAR2(100),
    내용 VARCHAR2(200),
    난이도 VARCHAR2(10)
);

```

```

INSERT INTO DIY VALUES (801, '강아지옷 만들기', '바느질로 옷 제작', '중');
INSERT INTO DIY VALUES (802, '장난감 DIY', '천 조각 활용', '하');
INSERT INTO DIY VALUES (803, '강아지 침대', '솜 채워 만들기', '중');
INSERT INTO DIY VALUES (804, '간식 제조', '오븐 간식', '상');
INSERT INTO DIY VALUES (805, '방석 만들기', '솜+천', '중');

```

-- 10. Match

```

CREATE TABLE Match (
    매칭ID NUMBER PRIMARY KEY,
    회원ID1 NUMBER,
    회원ID2 NUMBER,
    매칭일 DATE,
    상태 VARCHAR2(20),
    FOREIGN KEY (회원ID1) REFERENCES Member(회원ID),
    FOREIGN KEY (회원ID2) REFERENCES Member(회원ID)
);

```

```

INSERT INTO Match VALUES (901, 1, 2, TO_DATE('2024-03-01','YYYY-MM-DD'), '성공');
INSERT INTO Match VALUES (902, 2, 3, TO_DATE('2024-03-05','YYYY-MM-DD'), '대기');

```

```
INSERT INTO Match VALUES (903, 3, 4, TO_DATE('2024-03-10','YYYY-MM-DD'), '실패');
INSERT INTO Match VALUES (904, 4, 5, TO_DATE('2024-03-15','YYYY-MM-DD'), '성공');
INSERT INTO Match VALUES (905, 5, 1, TO_DATE('2024-03-20','YYYY-MM-DD'), '대기');
```

-- 11. CareGroup

```
CREATE TABLE CareGroup (
    품앗이ID NUMBER PRIMARY KEY,
    일정 DATE,
    참여회원ID NUMBER,
    내용 VARCHAR2(100),
    FOREIGN KEY (참여회원ID) REFERENCES Member(회원ID)
);
```

```
INSERT INTO CareGroup VALUES (1001, TO_DATE('2024-05-01','YYYY-MM-DD'), 1, '산책돌봄');
INSERT INTO CareGroup VALUES (1002, TO_DATE('2024-05-02','YYYY-MM-DD'), 2, '식사지원');
INSERT INTO CareGroup VALUES (1003, TO_DATE('2024-05-03','YYYY-MM-DD'), 3, '목욕');
INSERT INTO CareGroup VALUES (1004, TO_DATE('2024-05-04','YYYY-MM-DD'), 4, '미용');
INSERT INTO CareGroup VALUES (1005, TO_DATE('2024-05-05','YYYY-MM-DD'), 5, '산책');
```

-- 12. Training

```
CREATE TABLE Training (
    프로그램ID NUMBER PRIMARY KEY,
    제목 VARCHAR2(100),
    일정 DATE,
    설명 VARCHAR2(200)
);
```

```
INSERT INTO Training VALUES (1101, '기본 훈련', TO_DATE('2024-06-01','YYYY-MM-DD'), '앉아, 기다려');
INSERT INTO Training VALUES (1102, '배변 훈련', TO_DATE('2024-06-03','YYYY-MM-DD'), '패드 사용');
INSERT INTO Training VALUES (1103, '사회성 훈련', TO_DATE('2024-06-05','YYYY-MM-DD'), '다른 강아지와 교류');
```

```

INSERT INTO Training VALUES (1104, '짚음 방지',
TO_DATE('2024-06-07','YYYY-MM-DD'), '짚음 억제법');
INSERT INTO Training VALUES (1105, '트릭 훈련',
TO_DATE('2024-06-09','YYYY-MM-DD'), '돌아, 손');

```

-- 13. Dog_Disease (N:M)

```

CREATE TABLE Dog_Disease (
    강아지ID NUMBER,
    질병ID NUMBER,
    진단일 DATE,
    PRIMARY KEY (강아지ID, 질병ID, 진단일),
    FOREIGN KEY (강아지ID) REFERENCES Dog(강아지ID),
    FOREIGN KEY (질병ID) REFERENCES Disease(질병ID)
);

```

```

INSERT INTO Dog_Disease VALUES (101, 501,
TO_DATE('2024-02-01','YYYY-MM-DD'));
INSERT INTO Dog_Disease VALUES (102, 502,
TO_DATE('2024-02-02','YYYY-MM-DD'));
INSERT INTO Dog_Disease VALUES (103, 503,
TO_DATE('2024-02-03','YYYY-MM-DD'));
INSERT INTO Dog_Disease VALUES (104, 504,
TO_DATE('2024-02-04','YYYY-MM-DD'));
INSERT INTO Dog_Disease VALUES (105, 505,
TO_DATE('2024-02-05','YYYY-MM-DD'));

```

-- 14. TrainingApply (N:M)

```

CREATE TABLE TrainingApply (
    회원ID NUMBER,
    프로그램ID NUMBER,
    신청일 DATE,
    PRIMARY KEY (회원ID, 프로그램ID, 신청일),
    FOREIGN KEY (회원ID) REFERENCES Member(회원ID),
    FOREIGN KEY (프로그램ID) REFERENCES Training(프로그램ID)
);

```

```

INSERT INTO TrainingApply VALUES (1, 1101,
TO_DATE('2024-05-20','YYYY-MM-DD'));
INSERT INTO TrainingApply VALUES (2, 1102,
TO_DATE('2024-05-21','YYYY-MM-DD'));
INSERT INTO TrainingApply VALUES (3, 1103,

```

```
TO_DATE('2024-05-22','YYYY-MM-DD'));
INSERT      INTO      TrainingApply      VALUES      (4,      1104,
TO_DATE('2024-05-23','YYYY-MM-DD'));
INSERT      INTO      TrainingApply      VALUES      (5,      1105,
TO_DATE('2024-05-24','YYYY-MM-DD'));
```

--테이블 삭제 코드

```
DROP TABLE DIY CASCADE CONSTRAINTS;
```

8) 데이터 조작을 위한 스크립트 파일 내용 및 각 스크립트 실행 화면 캡처

<삽입/삭제/변경>

1. INSERT / DELETE / UPDATE

-- (1) Product 테이블에 새로운 강아지 용품 추가

```
INSERT INTO Product (용품ID, 이름, 종류, 가격, 재고)
VALUES (999, '강아지 우비', '외출', 18000, 10);
```

-- (2) 해당 우비 용품 삭제

```
DELETE FROM Product
WHERE 용품ID = 999;
```

-- (3) 기존 장난감 가격을 인상

```
UPDATE Product
SET 가격 = 가격 + 2000
WHERE 이름 = '장난감';
```

[원래 Product 테이블]

용품 ID	이름	종류	가격	재고
601	리드줄	외출	12000	20
602	강아지간식	식품	5000	50
603	방석	생활	15000	15
604	장난감	놀이	7000	30
605	삼푸	위생	9000	25
NULL	NULL	NULL	NULL	NULL

[삽입/삭제/변경 후 Product 테이블]

용품 ID	이름	종류	가격	재고
601	리드줄	외출	12000	20
602	강아지간식	식품	5000	50
603	방석	생활	15000	15
604	장난감	놀이	9000	30
605	삼푸	위생	9000	25
NULL	NULL	NULL	NULL	NULL

2. 다양한 검색 쿼리 (DIY 없음)

-- (1) 품종별 분양 상태가 '가능'인 강아지 리스트

```
SELECT a.품종, d.이름 AS 강아지이름
FROM Adoption a
JOIN Dog d ON a.강아지ID = d.강아지ID
WHERE a.상태 = '가능';
```

[Adoption테이블]

	분양 ID	품종	나이	상태	강아지ID
▶	201	푸들	3	가능	101
	202	비숑	2	완료	103
	203	진돗개	5	가능	105
	204	말티즈	4	대기	102
	205	시츄	3	가능	104
★	NULL	NULL	NULL	NULL	NULL

[검색 결과]

	품종	강아지 이름
▶	푸들	콩이
	진돗개	밤이
	시츄	두부

-- (2) 각 프로그램 난이도별 신청 인원 수 (GROUP BY)

```
SELECT t.제목, COUNT(*) AS 신청자수
FROM TrainingApply ta
JOIN Training t ON ta.프로그램ID = t.프로그램ID
GROUP BY t.제목;
```

[TrainingApply 테이블]

	회원 ID	프로그램ID	신청일
▶	1	1101	2024-05-20
	2	1102	2024-05-21
	3	1103	2024-05-22
	4	1104	2024-05-23
	5	1105	2024-05-24
★	NULL	NULL	NULL

[검색 결과]

	제목	신청 자수
▶	기본 훈련	1
	배변 훈련	1
	사회성 훈련	1
	짚음 방지	1
	트릭 훈련	1

-- (3) 병원 위치별 병원 수 집계

```
SELECT 위치, COUNT(*) AS 병원_수
FROM Hospital
GROUP BY 위치;
```

[Hospital 테이블]

병원 ID	이름	진료 과목	위치
401	24시동물병원	내과	서울
402	켓케어	외과	부산
403	멍멍병원	피부과	대구
404	러브펫	종합	광주
405	헬로동물병원	정형외과	대전
NULL	NULL	NULL	NULL

[검색 결과]

위치	병원_수
서울	1
부산	1
대구	1
광주	1
대전	1

-- (4) 전체 거래의 평균 구매 수량

```
SELECT AVG(수량) AS 평균_구매_수량
FROM Transaction;
```

[Transaction 테이블]

	거래 ID	판매 자ID	구매 자ID	용품 ID	수량	거래일
▶	701	1	3	601	1	2024-01-01
	702	2	1	602	2	2024-02-14
	703	2	5	603	1	2024-03-03
	704	4	2	604	3	2024-03-20
	705	5	3	605	2	2024-04-01
*	NULL	NULL	NULL	NULL	NULL	NULL

[검색 결과]

평균_구매_수량
1.8000

-- (5) '강아지간식'을 구매한 회원의 이름과 이메일 (서브쿼리 + 조인)

```
SELECT m.이름, m.이메일
FROM Member m
WHERE 회원ID IN (
    SELECT 구매자ID
    FROM Transaction
    WHERE 용품ID = (
        SELECT 용품ID FROM Product WHERE 이름 = '강아지간식'
    )
);
```

[Member 테이블]

회원 ID	이름	연락처	이메일
1	김철수	010-1234-5678	cs@example.com
2	이영희	010-2345-6789	yh@example.com
3	박민수	010-3456-7890	ms@example.com
4	최지현	010-4567-8901	jh@example.com
5	정우성	010-5678-9012	ws@example.com
NULL	NULL	NULL	NULL

[Transaction 테이블]

	거래 ID	판매자ID	구매자ID	용품 ID	수량	거래일
▶	701	1	3	601	1	2024-01-01
	702	2	1	602	2	2024-02-14
	703	2	5	603	1	2024-03-03
	704	4	2	604	3	2024-03-20
	705	5	3	605	2	2024-04-01
*	NULL	NULL	NULL	NULL	NULL	NULL

[검색 결과]

이름	이메일
김철수	cs@example.com