



데이터베이스 실무 기말 과제

데이터베이스 설계



2501110203

김호석

2025.12.03



목차

1. 과제 개요	1
1-1. 과제 요구사항	1
1-2. 과제 개요	1
2. 요구사항분석 명세서	3
2-1. 요구사항분석 명세서	3
3. 개념적 설계	5
3-1. Entity	5
3-2. 개념 ERD	8
4. 논리적 설계	9
4-1. 논리 ERD	9
4-2. 테이블 명세서	10
5. 구현	13
5-1. 테이블 생성 SQL 문	13
5-2. 구현 테스트	16

1. 과제 개요

1-1. 과제 요구사항

1-1-1. 데이터베이스를 아래의 단계를 따라 설계하시오

- 1 단계 : 요구사항분석 명세서 (개별 주제)
- 2 단계 : 개념적설계 (draw.io(E-R Diagram))
- 3 단계 : 논리적설계 (draw.io(릴레이션 스키마), 테이블 명세서(draw.io 또는 PPT))
- 5 단계 : 구현 (테이블 생성한 sql 문)

1-2. 과제 개요

1-2-1. 과제 주제



[“오늘의 학식” 로고]

본 데이터베이스는 학식 메뉴 열람 및 평가 프로그램 “**오늘의 학식**”을 설계하여, 해당 프로그램에 필요한 데이터를 저장하기 위한 데이터베이스입니다. “**오늘의 학식**”은 학생들에게 금주 메뉴 목록, 금일 메뉴 목록을 학생들에게 제공합니다. 학생들은 평가를 위한 계정을 생성하여 그날의 학식 메뉴를 평가하고 코멘트를 등록할 수 있습니다.

1-2-2. 과제 수행 목표

A	B	C	D	E
날짜	요일	조식	중식	석식
1 2025-09-29	월요일	쌀밥, 김치어묵국, 대파삼겹살청경재볶음, 풍조림, 짙곡밥, 쇠고기근대국, 제육느타리볶음, 잡채, 풍나물무침, 김쌀밥, 버섯계란국, 닭갈비, 스크램블에그, 호박맛살볶음, 김치		
3 2025-09-30	화요일	잡곡밥, 쇠고기무우국, 허제오리볶음, 어묵야채볶음쌀밥, 스팸강자짜글이, 닭도리탕, 해물볶음수동, 무밀랭이무침찹곡밥, 끝비지찌개, 가라아게, 미트볼야채볶음, 단호박샐러드, 김치		
4 2025-10-01	수요일	닭죽, 강자베이컨볶음, 지침만두, 양념깻잎지, (추기쌀밥), 옥수수크림스프, 토마토파스타, 면치까스, 췄러스, 김치계란라면, 김밥, 단무지, 야쿠르트, 김치		
5 2025-10-02	목요일	쌀밥, 볶어해장국, 치킨겟&미스터드, 햄강자볶음쌀밥	통합미역국, 대파삼겹살제육볶음, 오징어링튀김, 야식o 케이준샌드위치, 음료	
6 2025-10-03	금요일			
7 2025-10-13	월요일	쌀밥, 어우새우국, 떡갈비&소스, 해식브리운/김구이 쌀밥, 버섯사브국, 삼겹살제육볶음, 찐만두, 풍나물무침, 김치쌀밥, 미소장국, 닭드리탕, 들기름막국수, 마늘쫑무침, 김치		
8 2025-10-14	화요일	잡곡밥, 매운콩나물국, 제육볶음, 무밀랭이무침, 바 행아채볶음밥&짜장소스, 유부국, 통살새우까스, 숙주굴소스볶집곡밥, 통태매운탕, 향수육&소스, 두부구이, 부추김절이, 김치		
9 2025-10-15	수요일	잡곡밥, 원장국, 달걀데리야끼볶음, 고구마고로케, 짜장밥, 쇠고기근대국, 제육느타리볶음, 오징어콩나물김, 엔두쌀밥	김치무우국, 맵기슴살구이, 그린빈매일조림, 양념깻잎지, 김치	
10 2025-10-16	목요일	쌀밥, 예호박된장찌개, 인양식바싹불고기, 비엔나봉 치마요덮밥, 콩나물김치국, 국물떡볶이, 고구마맛탕, 마늘쫑잡곡밥, 김치찌개, 쇠고기야채볶음, 계란찜, 오징어젓부무침, 김치		
11 2025-10-17	금요일	잡곡밥, 통합무우국, 닭갈비, 어묵볶음, 계란후라이 쌀밥, 사골배추된장국, 순살치킨가스오스스, 도토리육무침, 코쌀밥, 오징어국, 향수육, 김치전, 그린샐러드, 김치		
12 2025-10-20	월요일	쌀밥*, 쇠고기대파국, 쪽찰미트불볶음, 열치건과볶음잡곡밥, 쇠고기미역국, 제육볶음, 등그란전, 부추무침, 김치 (쌀밥)*, 꼬치어묵탕, 안동닭찜, 꽃맛살샐러드, 양념깻잎지, 김치		
13 2025-10-21	화요일	후리카케밥, 쌀국수, 짜조&카레고로케, 가공커피우:잡곡밥, 김치유부국, 파인애플&치즈합박스테이크, 토마토판너쌀밥, 콩나물국, 대파삼겹살볶음, 계란찜, 상추쌈강저염쌈장, 김치		
14 2025-10-22	수요일	잡곡밥, 쇠고기떡만두국, 달다리살볶음, 에그그린샐 어향가지식고기덮밥, 계살스프, 남자지침만두, 연두부샐러드, 짜곡밥, 콩비지찌개, 달バラ케구이, 계란만두, 부추김절이, 김치		
15 2025-10-23	목요일	쌀밥, 순두부찌개, 돈육볶음수동, 고마돈까스, 풍동 짜곡밥, 김치두부국, 달장각바베큐구이, 오이양파무침, 스위트각두기돈육볶음밥, 미트볼야채볶음, 참치사리다빵, 짜사이무침, 쥬시클		
16 2025-10-24	금요일	잡곡밥*, 매콤시금치국, 맘볶음탕, 어묵볶음, 그린샐 카레라이스, 통합무국, 고추김재&또마야, 단호박샐러드, 야쿠쌀밥*	열무된장국, 온기스, 미트볼야채볶음, 오이파출, 김치	
17 2025-10-27	월요일	쌀밥, 쇠고기무국, 제육볶음, 허게소버터감자, 무말 짜곡밥, 부대찌개, 삼겹살세송이구이, 실곤약무침, 숙주나물, 쇠고기아체죽, 우동국물, 너비아니구이, 모듬풀라워호조장, 야쿠르트, 애		
18 2025-10-28	화요일	세우야채볶음밥, 두부랑이버섯국, 치즈합박스테이크쌀밥, 어죽얼같이국, 면치까스, 두부된육조림, 오이양파무침, 짜곡밥, 쇠고기미역국, 제육떡조림, 엿날소시지전, 콩나물무침, 김치		
19 2025-10-29	수요일	쌀밥*, 쇠고기백종국, 달걀데리야끼볶음, 스크램블에베이간길육볶음밥, 우동국물, 치킨까스, 둘깨양배추샐러드, 오접곡밥, 얼큰콩나물국, 순불바싹불고기, 어묵계란전, 양상추샐러드, 김치		
20 2025-10-30	목요일	잡곡밥*, 자돌버섯국, 김치삼겹살볶음, 그린샐러드, 짜곡밥, 볶어콩나물국, 허제오리볶음, 볶음짬뽕, 무비트무침, 어제볶음밥, 옥수수스프, 바게트빵, 순살돈까스, 그린샐러드, 김치		
21 2025-10-31	금요일	쌀밥, 쇠고기미역국, 토율육찹볶음, 베이컨강자채볶 짜곡밥, 계란버섯국, 블미바싹불고기볶음, 도토리油田, 콩나물국, 간풍기, 동그란전, 얼갈이무침, 김치		
22 2025-11-03	월요일	쌀밥, 얼큰쇠고기근대국, 떡갈비구이, 강자채볶음, 하이라이스, 매콤콩나물국, 생선까스, 참나물무침, 봉어빵, 김밥국, 데지짜글이, 오징어가스, 계란찜, 상추김절이, 김치		
23 2025-11-04	화요일	잡곡밥, 끓개찌개, 쇠고기볶음, 부추양파무침, 엔두 짜곡밥, 뚝배기, 양배추쌈&쌈장, 얼갈이된장무침밥, 어묵무우국, 너비아니구이, 오징어링튀김, 오이양파무침, 김치		
24 2025-11-05	수요일	닭죽, 치킨겟야채볶음, 김구이, 삶은계란, 요플레 청양오자기밀밥, 얼큰고기장국, 타코야끼, 맛살걸리볶음, 짜곡밥, 김치두부국, 향수육, 잡채, 양념깻잎지, 김치		

[기준 학식 메뉴 저장 방식]

위 이미지는 데이터를 엑셀파일로 저장했던 기준의 형식입니다. 본 과제에서는 해당 형식을 데이터베이스의 테이블 형식으로 변환하고 정규화하며, 기능을 확장하여 학생, 계정, 리뷰 테이블을 설계하고 구현하는 것을 목표로 합니다.

2. 요구사항분석 명세서

2-1. 요구사항분석 명세서

2-1-1. 학식

번호	요구사항
1	학식 정보는 매주 등록되며 (학식번호, 날짜, 구분, 메뉴) 정보를 유지해야 합니다.
2	구분에 들어가는 데이터는 "조식", "중식", "석식" 중 하나입니다.
3	동일 날짜에 구분이 같은 학식 정보는 등록할 수 없습니다.
4	학식은 학식번호로 식별합니다.

2-1-2. 학생

번호	요구사항
1	학생은 (학번, 이름, 성별, 생년월일) 정보를 유지해야 합니다.
2	학생은 학번으로 식별합니다.

2-1-3. 계정

번호	요구사항
1	학생은 "오늘의 학식"을 이용하기 위해 계정을 등록해야 합니다.
2	계정 생성 시 (학번, 비밀번호)를 등록해야 합니다.
3	계정은 (학번, 비밀번호, 생성날짜, 최근 접속일) 정보를 유지해야 합니다.
4	계정의 비밀번호는 데이터베이스에 암호화하여 저장해야 합니다.
5	학생 데이터가 삭제되면 계정도 함께 삭제되어야 합니다.
6	계정은 학번으로 식별합니다.

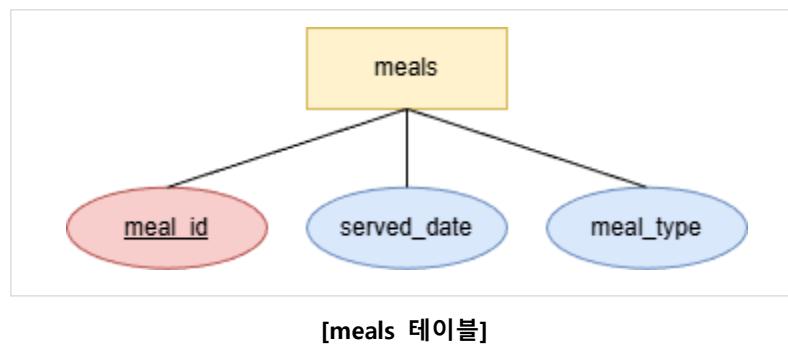
2-1-4. 평가

번호	요구사항
1	여러 학생은 여러 학식을 평가할 수 있습니다.
2	한 명의 학생이 하나의 학식을 여러 번 평가할 수 없습니다.
3	학식이 평가되면 (평가번호, 학식번호, 학생학번, 평점, 코멘트, 평가날짜) 정보를 기록해야 합니다.
4	평점은 1~10 사이의 값으로 기록합니다.
5	학생과 학식 데이터가 삭제되어도 평가 정보는 유지되어야 합니다.
6	평가는 평가번호로 식별합니다.

3. 개념적 설계

3-1. Entity

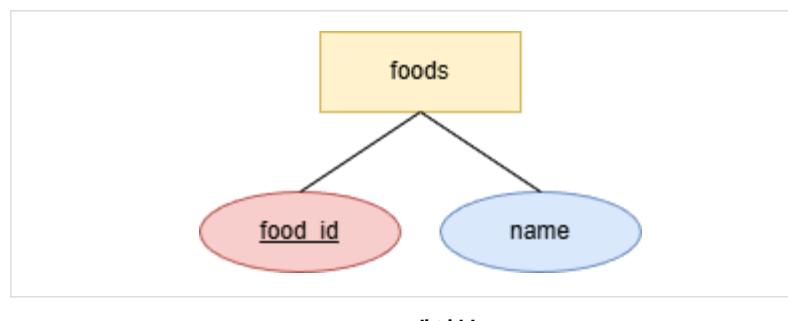
3-1-1. meals 테이블



[meals 테이블]

meals 테이블은 학식번호와 날짜, 구분 속성을 갖고 있습니다. meal_foods 테이블을 통해 foods 테이블과 연결하여 각 날짜별 학식 메뉴를 표시하고자 합니다. 구분 속성의 도메인은 “조식”, “중식”, “석식”입니다.

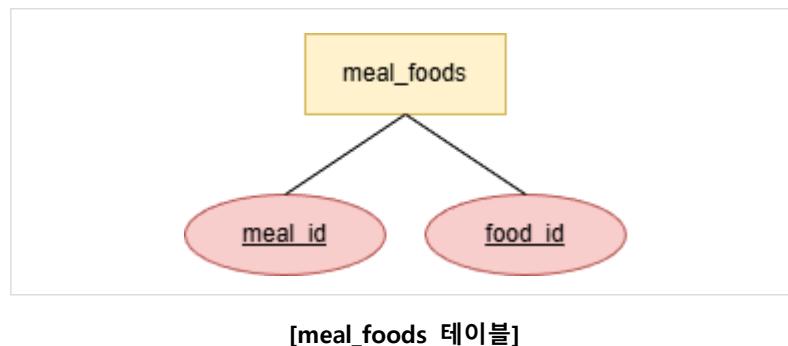
3-1-2. foods 테이블



[foods 테이블]

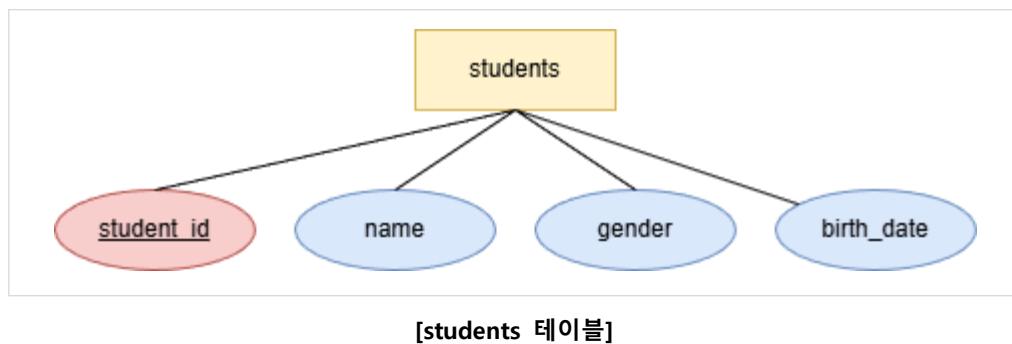
foods 테이블은 음식에 대한 데이터를 저장하는 테이블입니다. 음식 정보를 따로 저장하여 중복되는 음식 정보가 없도록 유지하려 하며, meal_foods 테이블을 통해 meals 테이블과 연결하여 각 날짜별 학식 메뉴를 표시하고자 합니다.

3-1-3. meal_foods 테이블



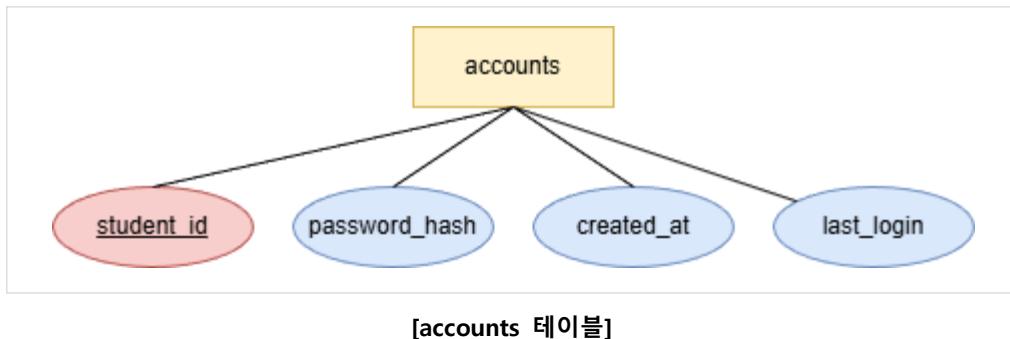
meal_foods 테이블은 기존에 설계한 테이블을 정규화하는 과정에서 분해된 테이블로 meals 테이블과 foods 테이블을 연결하기 위한 테이블입니다.

3-1-4. students 테이블



students 테이블은 학생에 대한 정보를 저장하는 테이블입니다. 학번, 이름, 성별, 생년월일을 속성으로 가집니다. 나이를 속성으로 설정하면 매년 나이를 증가시켜줘야 하므로 생년월일을 속성으로 설정하였습니다.

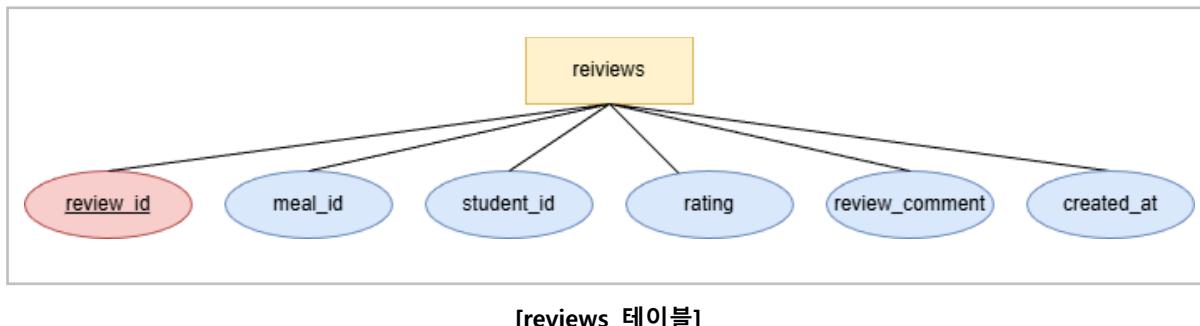
3-1-5. accounts 테이블



accounts 테이블은 학생이 학식 메뉴 프로그램을 사용하기 위해 등록하는 계정 정보를 저장하는 테이블입니다. 기존에는 students 테이블 안에 비밀번호까지 속성으로 설정하여 하나의 테이블로 사용하려 했습니다. 하지만 학생 정보가 존재해야만 계정을 생성할 수 있도록 하고 역할을 명확히 구분하기 위해 테이블을 분리하였습니다.

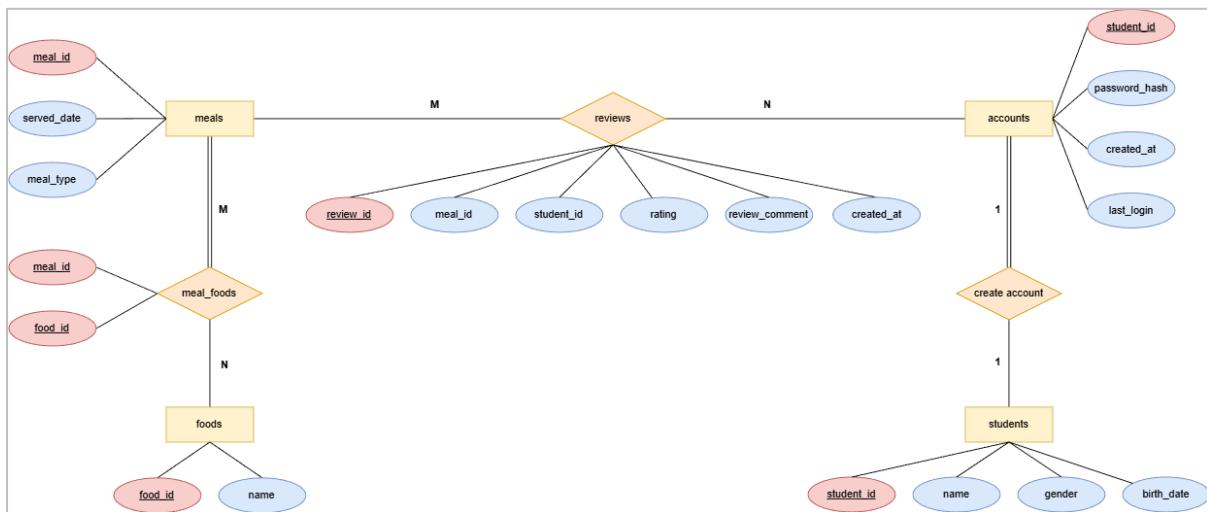
계정 등록 시 학번을 ID로 사용하고 비밀번호를 설정하도록 하고, 비밀번호는 해싱을 적용하여 암호화할 계획입니다.

3-1-6. reviews 테이블



review 테이블은 학생이 학식을 평가한 정보를 저장하는 테이블입니다. 리뷰번호, 평가한 학생의 학번, 평가된 학식의 번호, 평점, 코멘트, 평가 일시를 속성으로 가지고 있습니다.

3-2. 개념 ERD



[개념 ERD]

위 다이어그램은 각 테이블의 관계를 나타낸 E-R 다이어그램입니다. 정규화로 인해 분해된 학식 관련 테이블들을 연결하고 학생과 계정 테이블을 연결하였으며, "리뷰" 행위를 기준으로 학식 테이블과 계정 테이블을 연결하였습니다.

4. 논리적 설계

4-1. 논리 ERD

meals 테이블 릴레이션 스키마	<table border="1"><tr><td>meal_id</td><td>served_date</td><td>meal_type</td></tr></table>	meal_id	served_date	meal_type			
meal_id	served_date	meal_type					
foods 테이블 릴레이션 스키마	<table border="1"><tr><td>food_id</td><td>name</td></tr></table>	food_id	name				
food_id	name						
meal_foods 테이블 릴레이션 스키마	<table border="1"><tr><td>meal_id *외래키</td><td>food_id *외래키</td></tr></table>	meal_id *외래키	food_id *외래키				
meal_id *외래키	food_id *외래키						
students 테이블 릴레이션 스키마	<table border="1"><tr><td>student_id</td><td>name</td><td>gender</td><td>birth_date</td></tr></table>	student_id	name	gender	birth_date		
student_id	name	gender	birth_date				
accounts 테이블 릴레이션 스키마	<table border="1"><tr><td>student_id *외래키</td><td>password_hash</td><td>created_at</td><td>last_login</td></tr></table>	student_id *외래키	password_hash	created_at	last_login		
student_id *외래키	password_hash	created_at	last_login				
reviews 테이블 릴레이션 스키마	<table border="1"><tr><td>review_id</td><td>meal_id *외래키</td><td>student_id *외래키</td><td>rating</td><td>review_comment</td><td>created_at</td></tr></table>	review_id	meal_id *외래키	student_id *외래키	rating	review_comment	created_at
review_id	meal_id *외래키	student_id *외래키	rating	review_comment	created_at		

[논리 ERD]

개념 ERD 에 근거하여 논리 ERD 를 작성하였습니다. 각 테이블의 PK 와 FK 를 표시하였습니다.

4-2. 테이블 명세서

4-2-1. meals 테이블

테이블 명세서						
테이블 이름			meals			
속성 이름	데이터 타입	널 허용 여부	기본 값	기본 키	외래 키	제약 조건
meal_id	Int	N		PK		
served_date	Date	N				
meal_type	Varchar(10)	N				"조식" "중식" "석식"

UNIQUE

[meals 테이블 명세서]

meals 테이블은 날짜와 구분("조식","중식","석식") 컬럼을 끌어 UNIQUE로 설정하였습니다.
하루에 같은 구분의 학식이 존재하면 안 되기 때문입니다. (ex. 하루에 조식 두 번)

4-2-2. foods 테이블

테이블 명세서						
테이블 이름			foods			
속성 이름	데이터 타입	널 허용 여부	기본 값	기본 키	외래 키	제약 조건
food_id	Int	N		PK		
name	Varchar(30)	N				UNIQUE

[foods 테이블 명세서]

4-2-3. meal_foods 테이블

테이블 명세서						
테이블 이름			meal_foods			
속성 이름	데이터 타입	널 허용 여부	기본값	기본키	외래키	제약 조건
meal_id	Int	N		PK	FK	ON DELETE CASCADE
food_id	Int	N		PK	FK	ON DELETE CASCADE

[meal_foods 테이블 명세서]

meal_foods 테이블은 meals 테이블의 meal_id 와 foods 테이블의 food_id 를 외래키로 설정하여 두 테이블을 연결하는 역할을 합니다. 참조한 외래키가 부모 테이블에서 삭제될 경우 meal_foods 테이블에서도 데이터를 삭제하도록 ON DELETE CASCADE 제약 조건을 설정하였습니다.

4-2-4. students 테이블

테이블 명세서						
테이블 이름			students			
속성 이름	데이터 타입	널 허용 여부	기본값	기본키	외래키	제약 조건
student_id	Varchar(10)	N		PK		
name	Varchar(20)	N				
gender	Varchar(1)	Y				"F" "M"
birth_date	Date	Y				

[students 테이블 명세서]

4-2-5. accounts 테이블

테이블 명세서						
테이블 이름			accounts			
속성 이름	데이터 타입	널 허용 여부	기본값	기본키	외래키	제약 조건
student_id	Varchar(10)	N		PK	FK	ON DELETE CASCADE
password_hash	Varchar(255)	N				
created_at	Date	N				
last_login	Date	N				

[accounts 테이블 명세서]

accounts 테이블은 students 테이블의 student_id 를 외래키로 설정하여, 학생 정보가 없을 경우 계정을 등록할 수 없도록 설정했습니다. students 테이블에서 학생정보가 삭제될 경우 계정 정보도 삭제되도록 ON DELETE CASCADE 제약조건을 설정하였습니다.

4-2-6. reviews 테이블

테이블 명세서						
테이블 이름			reviews			
속성 이름	데이터 타입	널 허용 여부	기본값	기본키	외래키	제약 조건
review_id	Int	N		PK		
meal_id	Int	Y			FK	ON DELETE SET NULL
student_id	Varchar(10)	Y			FK	ON DELETE SET NULL
rating	Int	N				1 이상 10 이하
review_comment	Varchar(500)	Y				
created_at	Date	N				

UNIQUE

[reviews 테이블 명세서]

reviews 테이블은 students 테이블의 student_id 와 meals 테이블의 meal_id 를 외래키로 설정하였으며, 한 학생이 하나의 학식을 여러 번 평가하는 것을 방지하기 위해 두 컬럼을 묶어 UNIQUE 설정을 하였습니다. rating(평점)은 1~10 의 범위를 가지도록 설정했습니다.

meals 테이블과 accounts 테이블이 삭제되어도 리뷰 데이터를 유지하기 위해 ON DELETE SET NULL 제약조건을 설정하였습니다.

5. 구현

5-1. 테이블 생성 SQL 문

5-1-1. meals 테이블

```
CREATE TABLE meals (
    meal_id INT NOT NULL,
    served_date DATE NOT NULL,
    meal_type VARCHAR2(10) NOT NULL,
    PRIMARY KEY(meal_id),
    UNIQUE(served_date, meal_type),
    CONSTRAINT check_type CHECK(meal_type in ('조식', '중식', '석식'))
);
```

[meals 테이블 생성 SQL문]

served_date 컬럼과 meal_type 컬럼을 묶어 UNIQUE 제약조건을 설정하였습니다. 또한 meal_type 컬럼에 “조식”, “중식”, “석식”만 입력이 가능하도록 CHECK 옵션을 설정하였습니다.

5-1-2. foods 테이블

```
CREATE TABLE foods (
    food_id INT NOT NULL,
    name VARCHAR2(30) NOT NULL,
    PRIMARY KEY(food_id),
    UNIQUE(name)
);
```

[foods 테이블 생성 SQL문]

같은 이름의 음식이 등록되지 않도록 name 컬럼에 UNIQUE 제약조건을 설정하였습니다.

5-1-3. meal_foods 테이블

```
CREATE TABLE meal_food (
    meal_id INT NOT NULL,
    food_id INT NOT NULL,
    PRIMARY KEY(meal_id, food_id),
    FOREIGN KEY(meal_id) REFERENCES meals(meal_id) ON DELETE CASCADE,
    FOREIGN KEY(food_id) REFERENCES foods(food_id) ON DELETE CASCADE
);
```

[meal_foods 테이블 생성 SQL문]

meals 테이블과 foods 테이블의 데이터가 삭제될 경우 두 테이블을 연결하는 데이터까지 함께 삭제하기 위해 ON DELETE CASCADE 옵션을 적용하였습니다.

5-1-4. students 테이블

```
CREATE TABLE students (
    student_id VARCHAR2(10) NOT NULL,
    name VARCHAR2(20) NOT NULL,
    gender VARCHAR2(1),
    birth_date DATE,
    PRIMARY KEY(student_id),
    CONSTRAINT check_gender CHECK(gender IN ('F', 'M') OR gender IS NULL)
);
```

[students 테이블 생성 SQL문]

성별이 'F' or 'M'만 입력되거나 아예 입력되지 않도록 CHECK 옵션을 설정하였습니다.

5-1-5. accounts 테이블

```
CREATE TABLE accounts (
    student_id VARCHAR2(10) NOT NULL,
    password_hash VARCHAR2(255) NOT NULL,
    created_at DATE NOT NULL,
    last_login DATE NOT NULL,
    PRIMARY KEY(student_id),
    FOREIGN KEY(student_id) REFERENCES students(student_id) ON DELETE CASCADE
);
```

[accounts 테이블 생성 SQL문]

외래키인 student_id 가 부모 테이블인 students 테이블에서 삭제될 경우 accounts 테이블에서도 삭제되도록 ON DELETE CASCADE 옵션을 설정하였습니다.

5-1-6. reviews 테이블

```
CREATE TABLE reviews (
    review_id INT NOT NULL,
    meal_id INT,
    student_id VARCHAR2(10),
    rating INT NOT NULL,
    review_comment VARCHAR2(500),
    created_at DATE NOT NULL,
    PRIMARY KEY(review_id),
    FOREIGN KEY(meal_id) REFERENCES meals(meal_id) ON DELETE SET NULL,
    FOREIGN KEY(student_id) REFERENCES students(student_id) ON DELETE SET NULL,
    UNIQUE(meal_id, student_id),
    CONSTRAINT check_rating CHECK(rating BETWEEN 1 AND 10)
);
```

[reviews 테이블 생성 SQL문]

한 학생이 같은 학식을 여러 번 리뷰할 수 없도록 meal_id 컬럼과 student_id 컬럼을 묶어 UNIQUE 제약조건을 설정하였으며, 외래키를 참조한 테이블에서 데이터가 삭제되어도 리뷰 데이터를 유지할 수 있도록 ON DELETE NULL 옵션을 설정하였습니다. 평점이 1~10 범위를 벗어나지 않도록 CHECK 옵션을 설정하였습니다.

5-2. 구현 테스트

더미 데이터를 Insert하여 의도한 대로 동작하는지 테스트했습니다.

5-2-1. 학식 SELECT 테스트

```
SELECT m.served_date, m.meal_type, f.name  
FROM meals m, foods f, meal_foods mf  
WHERE m.meal_id = mf.meal_id AND mf.food_id = f.food_id;
```

[학식 데이터 SELECT문]

위 SELECT문을 사용하여 meals 테이블과 foods 테이블을 meal_foods 테이블을 통해 연결한 결과를 테스트 하였으며 결과는 아래와 같습니다.

	SERVED_DATE	MEAL_TYPE	NAME
1	25/09/29	조식	김치
2	25/09/29	조식	잡곡밥
3	25/09/29	조식	콩나물국
4	25/09/29	조식	훈제오리고기
5	25/09/29	중식	감자샐러드
6	25/09/29	중식	상추쌈
7	25/09/29	중식	영양보쌈
8	25/09/29	중식	흰쌀밥
9	25/09/29	중식	감자찌개
10	25/09/29	석식	김치
11	25/09/29	석식	잡곡밥
12	25/09/29	석식	콩나물국
13	25/09/29	석식	닭도리탕

[학식 데이터 SELECT 결과]

5-2-2. 학식 GROUP SELECT 테스트

```
SELECT
    m.served_date,
    m.meal_type,
    LISTAGG(f.name, ', ') WITHIN GROUP (ORDER BY f.name) AS food_list
FROM meals m
JOIN meal_foods mf ON m.meal_id = mf.meal_id
JOIN foods f ON mf.food_id = f.food_id
GROUP BY
    m.served_date,
    m.meal_type
ORDER BY
    m.served_date,
    m.meal_type;
```

[학식 데이터 GROUP SELECT문]

5-1-1의 SELECT문의 결과로는 학식 메뉴를 한눈에 파악하기 어렵습니다. 사용자 편의성을 고려하여 학식별로 메뉴를 한눈에 볼 수 있도록 SELECT문을 구성하였으며, 실행결과는 아래와 같습니다.

	SERVED_DATE	MEAL_TYPE	FOOD_LIST
1	25/09/29	석식	김치, 닭도리탕, 잡곡밥, 콩나물국
2	25/09/29	조식	김치, 잡곡밥, 콩나물국, 훈제오리고기
3	25/09/29	중식	감자샐러드, 감자찌개, 상추쌈, 영양보쌈, 흰쌀밥

[학식 데이터 GROUP SELECT 결과]

5-2-3. 리뷰 SELECT 테스트

```
SELECT r.review_id, m.served_date, m.meal_type, r.student_id, r.rating, r.review_comment  
FROM meals m, reviews r  
WHERE m.meal_id = r.meal_id  
ORDER BY r.review_id;
```

[리뷰 데이터 SELECT문]

학생이 학식을 평가한 내용을 확인할 수 있도록 SELECT문을 구성하여 실행 결과를 테스트했습니다. 결과는 아래와 같습니다.

	REVIEW_ID	SERVED_DATE	MEAL_TYPE	STUDENT_ID	RATING	REVIEW_COMMENT
1	1	25/09/29	조식	2501110203	10	밥 맛있어요!
2	2	25/09/29	중식	2501110200	1	최악
3	3	25/09/29	석식	2501110202	5	굿굿
4	4	25/09/29	조식	2501110199	6	고기 많이 주세요.

[리뷰 데이터 SELECT 결과]