

# 냉장고 속 재료로 찾는 레시피 추천 시스템

## 에이콘 아카데미 미니 프로젝트

*Food Recipe Finder*  
*Mini Project by Acorn Academy*

우동훈 · 박성욱 개발자(진)

# 목차

*Table of Contents*

## 1 프로젝트 개요

프로젝트 주제 : "냉장고 속 재료로 만들 수 있는 요리  
추천 시스템"

## 2 데이터베이스 설계

다이어그램 (테이블 관계도)

## 3 핵심 SQL 기능

## 4 개발 중 배운 코드

## 5 기능 시연

# 프로젝트 개요

*Status Page*

```
SELECT
  r.food_recipe_name AS 음식명,
  a.alcohol_name     AS 추천주종,
  a.alcohol_abv      AS 도수,
  rec.reason         AS 추천이유
FROM recommend_v2 rec
JOIN food_recipe_v2 r
  ON r.food_recipe_id = rec.food_recipe_id
```

```
);
-- ㉠ 메뉴-술 추천 매핑 (N:M) ※ FK 참조 컬럼명 정합성 수정
CREATE TABLE RECOMMEND_V2 (
  food_recipe_id NUMBER NOT NULL,
  alcohol_id     NUMBER NOT NULL,
  reason        VARCHAR2(200),
  CONSTRAINT PK_RECOMMEND_V2 PRIMARY KEY (food_recipe_id),
  CONSTRAINT FK_REC_RECIPE_V2 FOREIGN KEY (food_recipe_id)
```



## “냉장고 속 재료로 만들 수 있는 요리 추천 시스템”



- 집에 있는 재료를 효율적으로 활용
- 불필요한 식재료 낭비 감소
- 사용자에게 간단하고 직관적인 요리 정보 제공

# 데이터베이스 설계

Analysis Page

## "개념 다이어그램"

부모 테이블

food\_item(재료), food\_recipe(음식), alcohol(주류)

food\_recipe\_ingredient

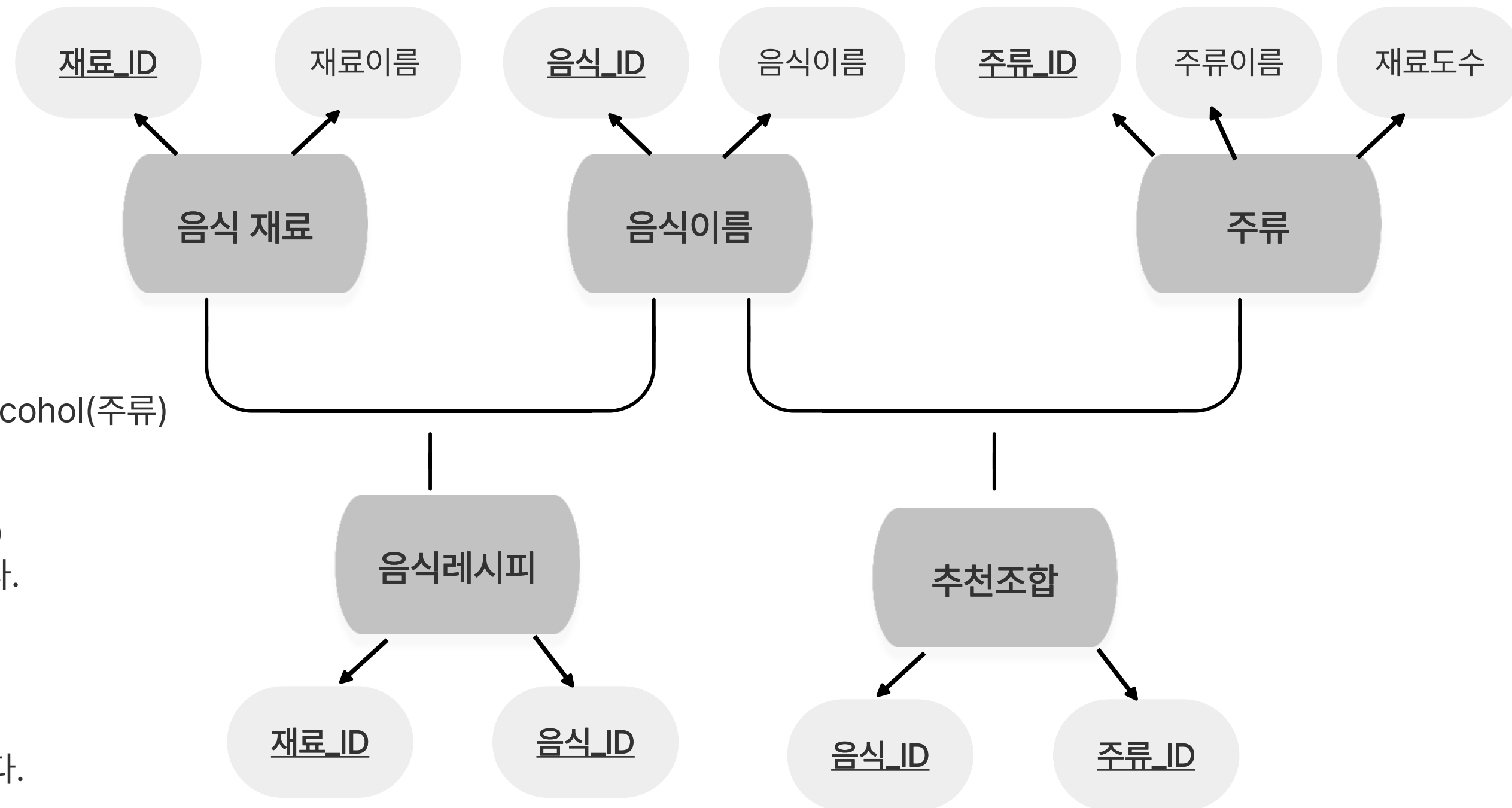
재료(food\_item) ↔ 음식(food\_recipe)

음식과 재료를 묶어주는 중간 테이블입니다.

recommend

음식(food\_recipe) ↔ 주류(alcohol)

음식과 주류를 연결하는 자식 테이블입니다.



# 데이터베이스 설계

Analysis Page

## "논리 다이어그램"

### 부모 테이블

food\_item(재료), food\_recipe(음식), alcohol(주류)

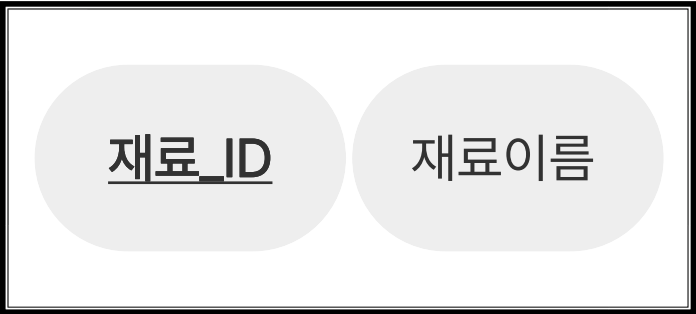
### food\_recipe\_ingredient

재료(food\_item) ↔ 음식(food\_recipe)  
음식과 재료를 묶어주는 중간 테이블입니다.

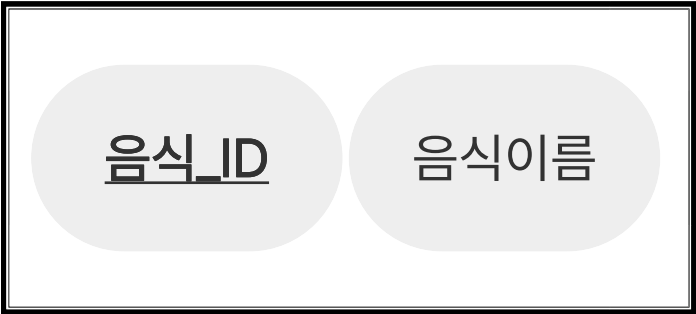
### recommend

음식(food\_recipe) ↔ 주류(alcohol)  
음식과 주류를 연결하는 자식 테이블입니다.

### 재료



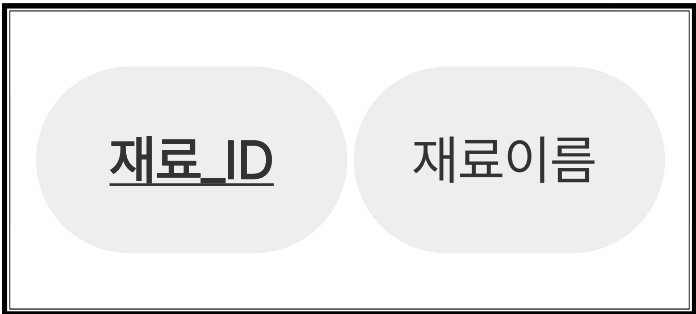
### 음식



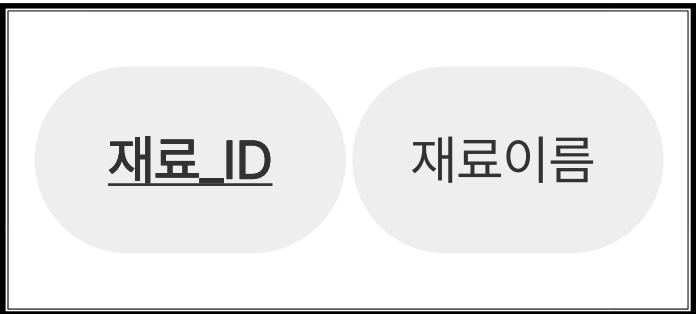
### 주류



### 음식레시피



### 추천조합



# 데이터베이스 설계

*Analysis Page*

## "물리 다이어그램"

부모 테이블

food\_item(재료), food\_recipe(음식), alcohol(주류)

food\_recipe\_ingredient


재료(food\_item) ↔ 음식(food\_recipe)  
음식과 재료를 묶어주는 중간 테이블입니다.

recommend


음식(food\_recipe) ↔ 주류(alcohol)  
음식과 주류를 연결하는 자식 테이블입니다.


 : PK(Primary Key, 기본키)

 : FK(Foreign Key, 외래키)  
= 다른 테이블의 기본키

food_item				
	재료아이디	food_item_id	Domain	number
	재료이름	food_item_name	Domain	varchar2(100)

food_recipe_ingredient				
	음식아이디	recipe_id	Domain	number
	재료아이디	item_id	Domain	number
	재료아이디	food_item_id	Domain	number
	음식아이디	food_recipe_id	Domain	number

food_recipe				
	음식아이디	food_recipe_id	Domain	number
	음식이름	food_recipe_name	Domain	varchar2(150)

alcohol				
	주류아이디	al_id	Domain	number
	주류이름	al_name	Domain	varchar(50)
	주류도수	al_content	Domain	number

recommend				
	추천음식아이디	recom_recipe_id	Domain	number
	추천주류아이디	recom_al_id	Domain	number
	음식아이디	food_recipe_id	Domain	number
	주류아이디	al_id2	Domain	number
	추천이유	recom_food_com	Domain	varchar(50)

# 핵심 SQL 기능

## 코드

```
CREATE TABLE FOOD_RECIPE_INGREDIENT (
  food_recipe_id NUMBER NOT NULL, -- 레시피 ID (FOOD_RECIPE 참조)
  food_item_id   NUMBER NOT NULL, -- 재료 ID (FOOD_ITEM 참조)
  CONSTRAINT PK_RECIPE_INGREDIENT PRIMARY KEY (food_recipe_id, food_item_id)
  |
  CONSTRAINT FK_RECIPEING_TO_RECIPE FOREIGN KEY (food_recipe_id)
    REFERENCES FOOD_RECIPE(food_recipe_id),
  ...
)
```

## 결과

	FOOD_ITEM_ID	FOOD_ITEM_NAME
1	1	마늘
2	2	양파
3	3	대파
4	4	감자
5	5	고구마

## 코드 설명

### NUMBER GENERATED ALWAYS AS IDENTITY

새로운 행이 추가시, 순서대로 번호를 생성합니다.  
번호를 넣어줄 필요 없이 \*\*고유 ID(기본키) 관리 할수 있습니다.



### CONSTRAINT FK\_RECIPEING\_TO\_ITEM FOREIGN KEY(food\_item\_id)

'CONSTRAINT+ 제약조건 이름+ 제약조건+ 컬럼이름'  
테이블에 저장되는 데이터에 규칙을 부여하는 장치입니다.

# 핵심 SQL 기능

## 코드

```
INSERT INTO FOOD_RECIPE_INGREDIENT (food_recipe_id, food_item_id)  
SELECT r.food_recipe_id, i.food_item_id  
FROM FOOD_RECIPE r  
JOIN FOOD_ITEM i ON i.food_item_name IN ('베이컨',  
WHERE r.food_recipe_name = '베이컨말이');
```

## 코드 설명

### INSERT 에서 VALUES 아닌 SELECT

VALUES 아닌 SELECT로 값을 받을수 있다.



# 핵심 SQL 기능

## 코드

```
LISTAGG(i.food_item_name, ', ' ) AS  
T fri  
food_recipe_id = r.food_recipe_id  
od_item_id = i.food_item_id  
'된장찌개'
```

## 결과

SQL   인출된 모든 행: 1(0.001초)	
FOOD_RECIPE_NAME	INGREDIENTS
1 된장찌개	양파, 대파, 된장, 두부

## 코드 설명

**LISTAGG(i.food\_item\_name, ', ')**

여러 행을 하나의 문자열로 합치는 집계 함수  
반드시 GROUP BY가 필요

→ **WITHIN GROUP (ORDER BY -  
컬럼이름 )**

집계 함수 결과를 낼 때 합치는 순서를 지정하는 옵션

# 핵심 SQL 기능

## 코드

```
HAVING COUNT(CASE WHEN fi.food_item_name
                IN ('두부','홍합','마늘','대파') THEN 1 END) > 0;
```

## 결과

음식명	전체재료목록
1 간장두부조림	간장, 두부, 마늘
2 계란말이	계란, 대파, 소금
3 고등어조림	간장, 감자, 고등어, 마늘
4 고추장어묵볶음	고추장, 두부, 양파

## 코드 설명

**CASE WHEN fi.food\_item\_name  
IN ('두부','홍합','마늘','대파')  
THEN 1 END**

CASE WHEN ... THEN 1 = 체크박스: 조건 맞으면 체크(1),  
아니면 비워둠(NULL)

COUNT = 체크된 개수 세기

HAVING ... > 0 = 체크된 게 하나 이상 있으면 통과