

데이터베이스 최종 보고서

프로젝트명		냉장고를 부탁해																	
1. 개요 및 필요성		팀 : 수학통계학부 수학과 설민욱, 수학통계학부 통계학과 권태양																	
<div>○ 주제 선정이유</div> <p>우리는 요리를 만들기 위해 많은 재료를 사지만, 정작 무엇을 만들지 잘 모르는 경우가 많습니다. 그리고 최근 요리 관련 서적이 많지만, 내가 가지고 있는 재료로 어떤 요리를 만들 수 있을지 확인하기 힘든 점이 많습니다. 요리를 못하는 사람들에게 아무리 재료가 많이 있다고 해도 어떤 요리를 고를지 결정하는 것은 무척이나 힘든 일입니다. 그래서 냉장고에 있는 재료를 입력해서 검색을 하면 현재 가지고 있는 재료를 확인하여 지금 만들 수 있는 요리를 제시해주는 기능을 구현해 보려고 합니다. 또한 데이터베이스 안에 저장되어있는 레시피를 확인하고 수정할 수 있습니다.</p>																			
2. 프로젝트 내용																			
<div>○ 시스템 구성도(데이터흐름도)</div> <div>-화면흐름도</div> <div><pre>graph LR MM[메인 메뉴] --> TR[전체 레시피] MM --> NH[냉장고 현황] MM --> YG[요리 검색] MM --> ZT[종료] TR --> TRS[레시피 검색] TR --> TRN[새로운 레시피 추가] TR --> TRU[기존 레시피 수정] TR --> TRM[메인 메뉴] NH --> NHZ[재료 추가] NH --> NHK[재료 삭제] NH --> NHU[재료 확인] NH --> NHM[메인 메뉴] YG --> YG1[가능한 요리 검색] YG --> YG2[세부 검색] YG --> YGM[메인 메뉴]</pre></div>																			
<div>○ 요구사항분석 명세서</div> <table><tr><th colspan="4">업무 기술서</th></tr><tr><td>시스템명</td><td>냉장고를 부탁해 DB버전</td><td>작성일</td><td>2019년 6월 9일</td></tr><tr><td>서비스시스템명</td><td>ForTheCcook_db</td><td>작성자</td><td>설민욱</td></tr><tr><td>단위업무명</td><td colspan="3">레시피 검색,추가,삭제,수정 및 재료 확인 후, 추천레시피 제시</td></tr></table> <div>1. 목적</div> <p>사람들이 레시피를 찾을 때, 제일 많이 쓰는 어플인 “만개의레시피” 어플의 내용을 참고하여 요리 재료만 입력하면 추천 레시피를 제시 할 수 있게 구현해내고 기존에 있는 레시피를 검색, 수정, 삭제 그리고 새로운 레시피를 추가할 수 있게 만들어준다.</p> <div>2. 프로세스</div> <ul style="list-style-type: none">- 요리 이름을 기준으로 데이터를 등록한다.- 요리 이름을 기준으로 데이터를 추가할 때, NULL값이 만들어지지 않도록 한다.- 요리 이름과 요리 재료는 1대1로 대응시킬 수 있도록 한다. ex)떡볶이 - 떡, 떡볶이 - 양파- 요리 재료를 입력할 때, 물,설탕,소금,간장,식초,후추,고춧가루,간마늘,식용유은 있다는 가정하에 입력할 수 있도록 한다. <div>3. 노트</div> <ul style="list-style-type: none">- 요리 이름에 대응되는 데이터 : 요리 종류(cook_kind), 요리 시간(cook_tme), 요리 난이도				업무 기술서				시스템명	냉장고를 부탁해 DB버전	작성일	2019년 6월 9일	서비스시스템명	ForTheCcook_db	작성자	설민욱	단위업무명	레시피 검색,추가,삭제,수정 및 재료 확인 후, 추천레시피 제시		
업무 기술서																			
시스템명	냉장고를 부탁해 DB버전	작성일	2019년 6월 9일																
서비스시스템명	ForTheCcook_db	작성자	설민욱																
단위업무명	레시피 검색,추가,삭제,수정 및 재료 확인 후, 추천레시피 제시																		

(cook_difficulty), 나라별 요리(cook_world), 요리 재료(cook_material)

- 내 냉장고에 있는 재료가 들어 있는 데이터 : 요리 재료(My_refrigerator_material)

4. 입력사항 : 요리 재료, 요리 이름, 냉장고에 있는 재료

5. 출력사항 : 재료에 맞게 추천되는 기존 레시피, 요리 이름에 대응되어 있는 모든 정보

○ 테이블 기술서

테이블명	cook_kind	테이블 기술서	작성일	19년 6월 9일	page/	
System	ForTheCcook_db		작성자	설민욱		
테이블 설명		요리 이름과 요리 종류를 대응시킨 테이블입니다.				
번호	속성	데이터형	널	키	기본값	설명
1	cook_name	varchar(32)	NN	PK		요리 이름
2	cook_kind	varchar(32)	NN			요리 종류

테이블명	cook_time	테이블 기술서	작성일	19년 6월 9일	page/	
System	ForTheCcook_db		작성자	설민욱		
테이블 설명		요리 이름과 요리 시간을 대응시킨 테이블입니다.				
번호	속성	데이터형	널	키	기본값	설명
1	cook_name	varchar(32)	NN	PK		요리 이름
2	cook_time	varchar(32)	NN			요리 시간

테이블명	cook_difficulty	테이블 기술서	작성일	19년 6월 9일	page/	
System	ForTheCcook_db		작성자	설민욱		
테이블 설명		요리 이름과 요리 난이도를 대응시킨 테이블입니다.				
번호	속성	데이터형	널	키	기본값	설명
1	cook_name	varchar(32)	NN	PK		요리 이름
2	cook_difficulty	varchar(32)	NN			요리 난이도

테이블명	cook_world	테이블 기술서	작성일	19년 6월 9일	page/	
System	ForTheCcook_db		작성자	설민욱		
테이블 설명		요리 이름과 나라별 요리를 대응시킨 테이블입니다.				
번호	속성	데이터형	널	키	기본값	설명
1	cook_name	varchar(32)	NN	PK		요리 이름
2	cook_world	varchar(32)	NN			나라별 요리

테이블명	cook_material	테이블 기술서	작성일	19년 6월 9일	page/	
System	ForTheCcook_db		작성자	설민욱		
테이블 설명		요리 이름과 요리 재료를 대응시킨 테이블입니다.				
번호	속성	데이터형	널	키	기본값	설명
1	cook_name	varchar(32)	NN	PK		요리 이름
2	cook_materials	varchar(32)	NN			요리 재료

테이블명	My_refrigerator	테이블 기술서	작성일	19년 6월 9일	page/	
System	ForTheCcook_db		작성자	설민욱		
테이블 설명		내 냉장고에 있는 재료가 들어있는 테이블입니다.				
번호	속성	데이터형	널	키	기본값	설명
1	My_materials	varchar(32)	NN			요리 재료

○ 용어 사전

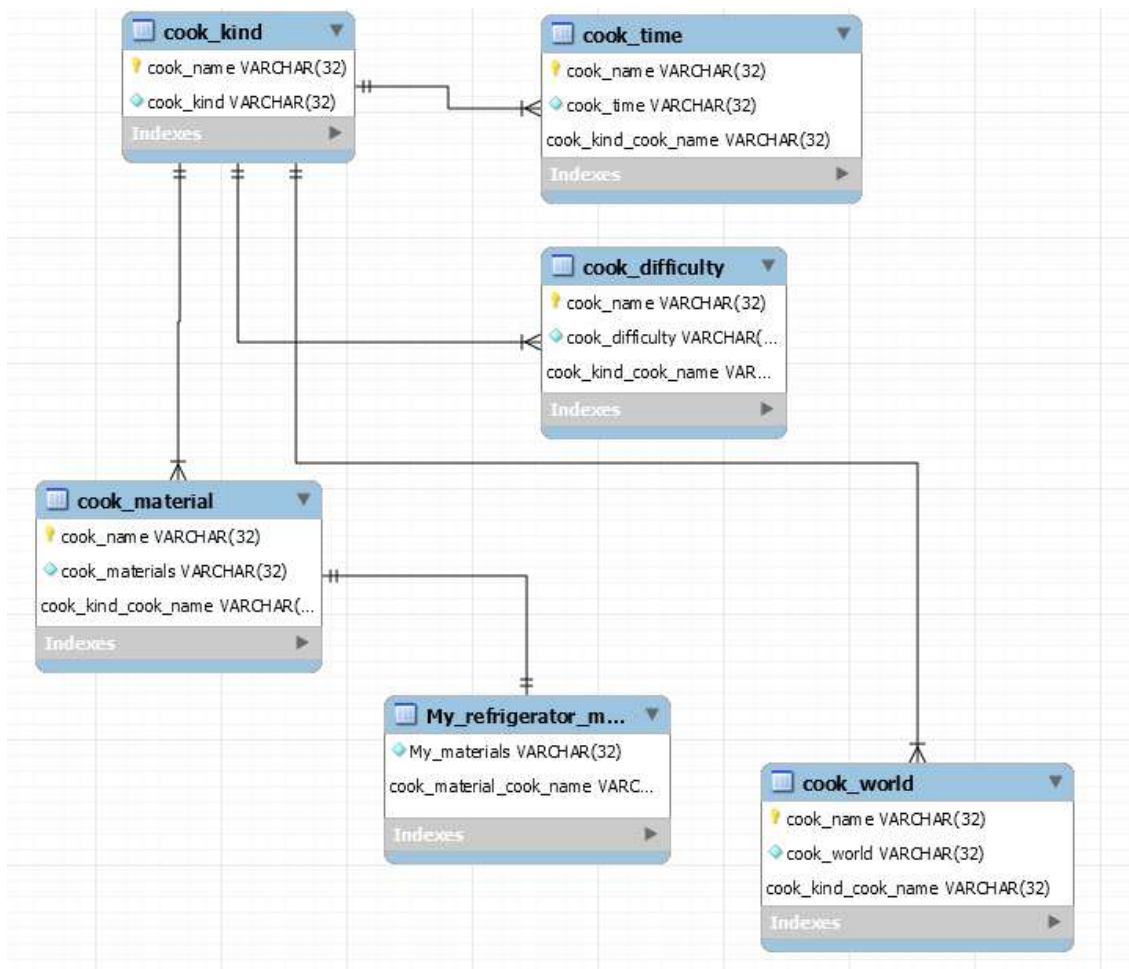
System	ForTheCcook_db	용어사전	작성일	19년 6월 9일	page/
			작성자	설민욱	
논리명		물리명		약어	설명
요리 이름		cook_name		cookname	요리 이름

요리 종류	cook_kind	cookkind	요리 종류
요리 시간	cook_time	cooktime	요리 시간
요리 난이도	cook_difficuty	cookdifficulty	요리 난이도
나라별 요리	cook_world	cookworld	나라별 요리
요리 재료	cook_materials	cookmaterials	레시피에 들어 있는 요리재료
냉장고 안 재료	My_materials	Mymaterials	냉장고 안에 있는 요리재료

○ 도메인기술서

System	ForTheCok_db	도메인기술서	작성일	19년 6월 9일	page/
			작성자	설민옥	
논리명	물리명	데이터형	설명		
요리 이름	cook_name	varchar(32)	요리 이름		
요리 종류	cook_kind	varchar(32)	메인,반찬,디저트로 나눈다.		
요리 시간	cook_time	varchar(32)	10분이하,10분에서30분이하,30분에서1시간이하,1시간이상으로 나눈다.		
요리 난이도	cook_difficuty	varchar(32)	상,중,하로 나눈다		
나라별 요리	cook_world	varchar(32)	한식,중식,양식,일식으로 나눈다		
요리 재료	cook_materials	varchar(32)	레시피에 들어 있는 요리재료		
냉장고 안 재료	My_materials	varchar(32)	냉장고 안에 있는 요리재료		

○ ER-다이어그램



○ 정규화 과정

- 제 1 정규형의 조건은 릴레이션에 속한 모든 속성의 도메인이 원자값으로만 구성되어 있으면 제 1정규형에 속한다. 위에 있는 테이블기술서에 보이듯이 모든값이 varchar(32)로 다 원자값으로만 구성되어 있다. 따라서 성립한다.
- 제 2 정규형의 조건은 릴레이션이 제 1 정규형에 속하고 기본키가 아닌 모든 속성이 디본키에 완전 함수 종속되면 제 2 정규형에 속한다. 테이블 기술서에 보면 모든 기본키

는 cook_name이고 cook_name에 다른 데이터가 1:1대응을 이루고 있는 것을 볼 수 있다. 따라서 기본키에 완전 함수 종속이 된다. 따라서 성립한다.

- 제 3 정규형의 조건은 릴레이션이 제 2 정규형에 속하고, 기본키가 아닌 모든 속성이 기본키에 이행적 함수 종속이 되지 않으면 제 3 정규형에 속한다. 테이블기술서에 보면 cook_name을 제외한 모든 데이터가 요리 이름이라는 기준이 없으면 서로 연관이 없는 데이터이기 때문에 기본키에 이행적 함수 종속이 되지 않는다. 따라서 성립한다.

○ 과제소스

*DB연동/처리부분

저희 조는 python안에 있는 패키지인 pymysql패키지를 이용해서 DB를 연동시켰습니다.

```
def Search_Recipe():
    print()
    a = input("원하는 요리를 입력하세요 : ")
    conn = pymysql.connect(host='localhost', user='root', password='0000',
                           db='cook_db', charset='utf8')

    # Connection 으로부터 Cursor 생성
    curs = conn.cursor()

    # SQL로 실행
    # cook_kind에 cook_time, cook_time, cook_difficulty, cook_difficulty, cook_world, cook_world를 cookname 별로 Join
    # 원하는 요리의 이름, 종류, 시간, 난이도, 나라별 음식 출력
    sql = """select cook_kind.*, cook_time.cook_time, cook_difficulty.cook_difficulty, cook_world.cook_world from cook_kind left join cook_time
              on cook_kind.cook_name = cook_time.cook_name
              left join cook_difficulty
              on cook_kind.cook_name = cook_difficulty.cook_name
              left join cook_world
              on cook_kind.cook_name = cook_world.cook_name
              where cook_kind.cook_name = %s""" + a + "%s"

    curs.execute(sql)

    # 데이터 Fetch
    rows = curs.fetchall()

    print()

    print(rows[0][0])
    print("요리 종류 : " + rows[0][1])
    print("요리 시간 : " + rows[0][2])
    print("요리 난이도 : " + rows[0][3])
    print(rows[0][4] + "입니다.")

    # Connection 으로부터 Cursor 생성
    curs = conn.cursor()

    # 요리 재료 확인
    sql = "select cook_materials from cook_material where cook_name = %s" + a + "%s"
    curs.execute(sql)

    # 데이터 Fetch
    rows = curs.fetchall()

    234
    235 # 삭제할 요리가 있는지 확인
    236 if len(rows) != 0:
    237     curs = conn.cursor()
    238     # cook_kind에 각 테이블을 Join한 뒤 삭제할 요리를 확인하고 화면에 삭제
    239     sql = """delete cook_kind, cook_time, cook_difficulty, cook_world, cook_material from cook_kind left join cook_time
    240               on cook_kind.cook_name = cook_time.cook_name
    241               left join cook_difficulty
    242               on cook_kind.cook_name = cook_difficulty.cook_name
    243               left join cook_world
    244               on cook_kind.cook_name = cook_world.cook_name
    245               left join cook_material
    246               on cook_kind.cook_name = cook_material.cook_name
    247               where cook_kind.cook_name = %s""" + a + "%s"
    248
    249     curs.execute(sql)
    250     print("삭제되었습니다.")
    251 else:
    252     print("입력하신 요리는 없습니다.")
    253     print()
    254
    255     conn.commit()
    256     conn.close()
    257
    258 def Search_All_Cooks():
    259     conn = pymysql.connect(host='localhost', user='root', password='0000',
    260                           db='cook_db', charset='utf8')
    261
    262     # Connection 으로부터 Cursor 생성
    263     curs = conn.cursor()
    264
    265     # SQL로 실행
    266     sql = "select cook_name from cook_kind"
    267
    268     curs.execute(sql)
    269
    270     # 데이터 Fetch
    271     rows = list(curs.fetchall())
    272
    273     # 전체 요리 이름을 출력하기 위해 리스트로 저장
    274     recipe = []
    275     for i in rows:
    276         recipe.append(*i)
    277
```

- DB: MySQL
- 언어: Python

4. 팀원구성 및 역할

팀 장 설민욱 : 모든 보고서를 작성하고 DB를 설계하고 연동하는 부분을 했고 레시피에 대한 DB가 없어서 DB를 직접 만들었다. 또한, 발표 자료 준비와 발표 담당을 맡게 되었다

팀 원 권태양 : 아이디어를 제시하고 python부분에서 모든 주요기능 개발했고 UI디자인과 기능 검토 담당을 맡았다.