

Database Term Project

1단계 : Requirement Analysis & Design(수정)
2단계 : System Implementation

산업경영공학부
2020170822
권민석

Dept. of Computer Science and Engineering,
College of Informatics, Korea University

Requirement Analysis & Conceptual Design

1차 제출물에서 수정된 부분

1. Using_for_cook 을 약개체가 아닌 relationship으로 표현하고 attribute를 별도로 표시함.
2. Inventory를 약개체로 수정하여 ingredient_id, restaurant_id 로 식별할 수 있게 수정함.
3. Do 테이블에서 Dish_id를 Dish_name으로 수정(ERD와 불일치 수정), 약개체를 참조하고 있으므로 Restaurant_id또한 Menu 테이블에서 참조하여 PK로 사용(식별문제 해결)
4. p_order_id의 자료형을 int로 수정하고 auto increment를 적용함
5. 모든 참조 제약조건을 cascade로 설정함.

업무개요서

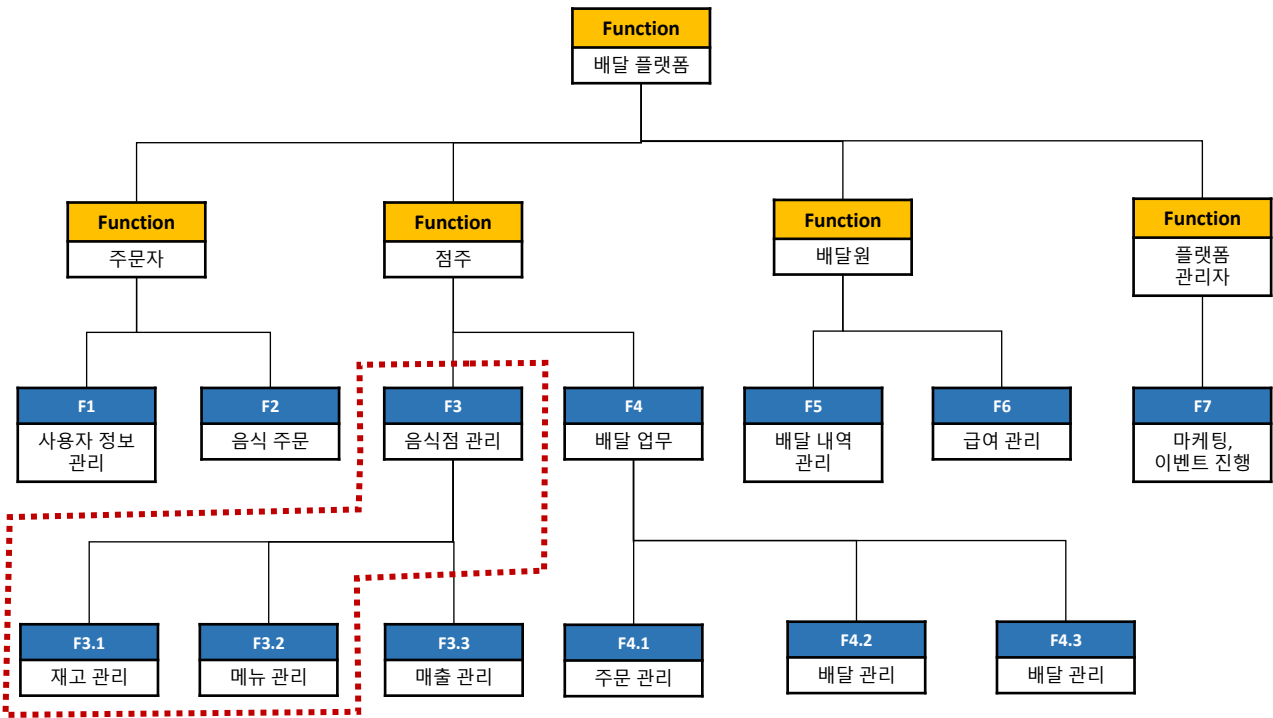
❖ 개요

- 배달 플랫폼은 총 세명의 사용자인 주문자(Customer), 점주(Manager), 배달원(Rider)을 연결하는 통합 시스템이다. 이 시스템은 크게 주문자 기능, 점주 기능, 배달원 기능, 플랫폼 관리자 기능으로 나누어져 있으며, 이번 시스템 개발 대상은 점주 기능 업무이다.

❖ 개발 대상

- 음식점 기능은 배달 플랫폼을 통해 음식점을 운영하는 점주의 입장에서 작동하는 기능이다. 점주는 자신의 음식점을 관리하고 들어온 주문을 처리하는 작업을 진행한다. 관리할 내용으로는 재고 정보, 주문내역 및 매출 정보, 음식점의 메뉴가 있으며, 주문을 처리하는 작업에는 사용자로부터 입력된 주문정보에 대하여 적절한 배달원을 매칭하고, 배달 후 변경되는 재고데이터를 반영하는 기능이 포함되어 있다. 이번 개발 대상으로는 음식점을 관리하는 작업으로, 재고 관리, 주문 내역 관리, 매출 관리 기능을 구현하고 관련 데이터베이스를 구축하는 것이 목표이다.

기능분해도



개발 업무범위

요구사항 명세서

기능명

재고관리

1. 음식점에 대한 정보는 음식점 번호, 음식점 이름, 점주 이름, 점주 전화번호를 포함하고있다.
2. 음식점 번호당 고유한 음식점 이름이 존재한다.
3. 음식점은 개별적으로 요리에 필요한 식자재를 저장하여 관리하고있다.
4. 관리중인 식자재 정보로는 식재료번호, 식자재이름, 식자재종류, 수량이 포함된다.
5. 음식점은 준비한 재료가 떨어지지 않도록 관리하기 위해 정기적으로 식자재 마트에 발주를 넣어 재료를 보충한다.
6. 발주는 한번에 여러 개 마트에 넣을 수 있지만, 하나의 발주는 하나의 식자재에 대한 정보만 입력되어있다. 식자재 마트는 여러 개의 음식점으로부터 발주를 받을 수 있다.(발주를 받지 못한 마트가 존재할 수도 있다.)
7. 식자재 마트에 대한 정보는 마트번호, 마트명, 지역, 대표전화번호가 있다.
8. 마트 번호당 고유한 마트명이 존재한다.
9. 발주 내용에는 발주번호, 발주신청일이 포함되어있으며, 어떤 식자재를 얼마나 구매할지에 대한 내용을 다룬다.
- 10.발주 결과에 대해서 식자재 정보가 갱신된다.

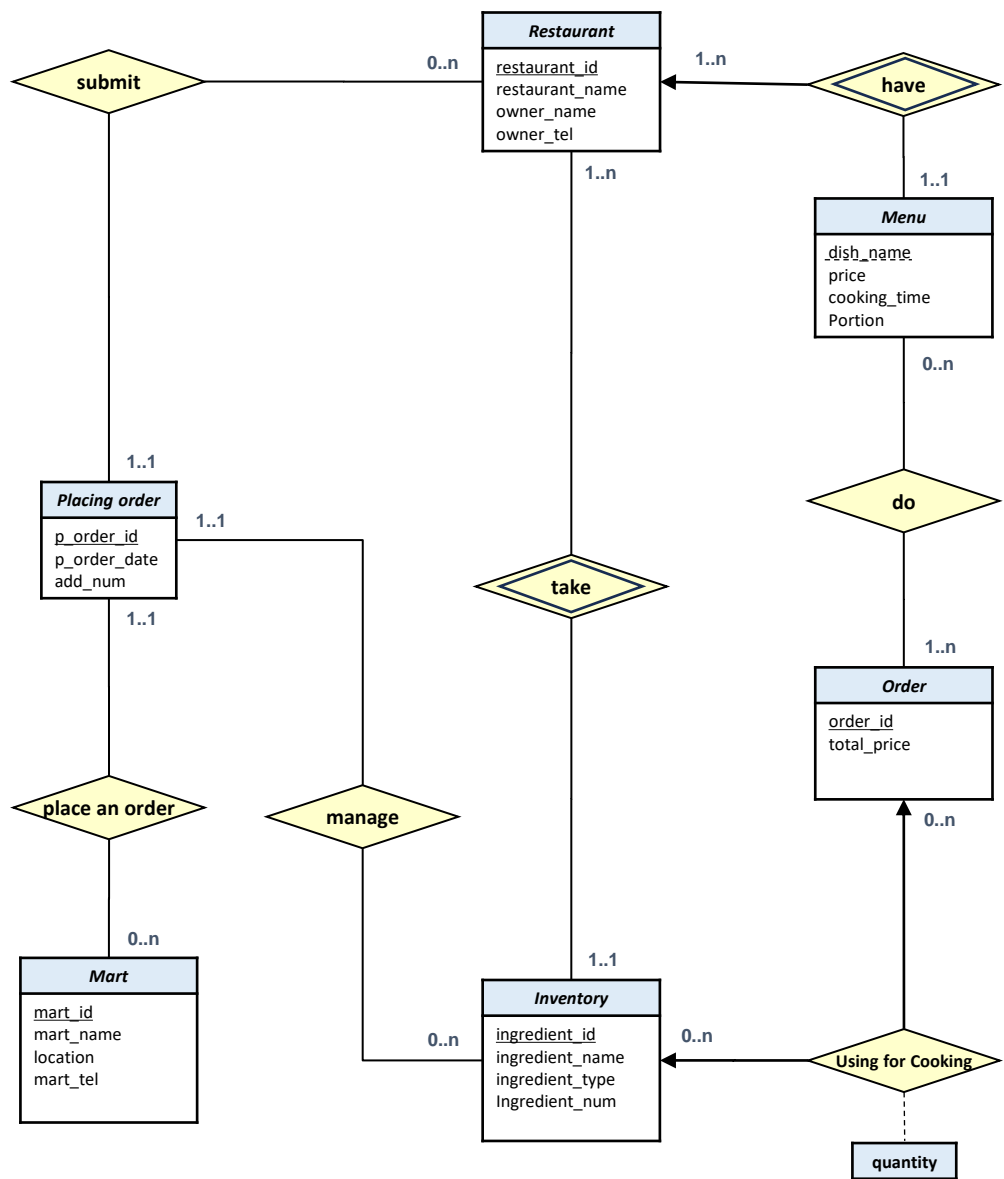
요구사항 명세서

기능명

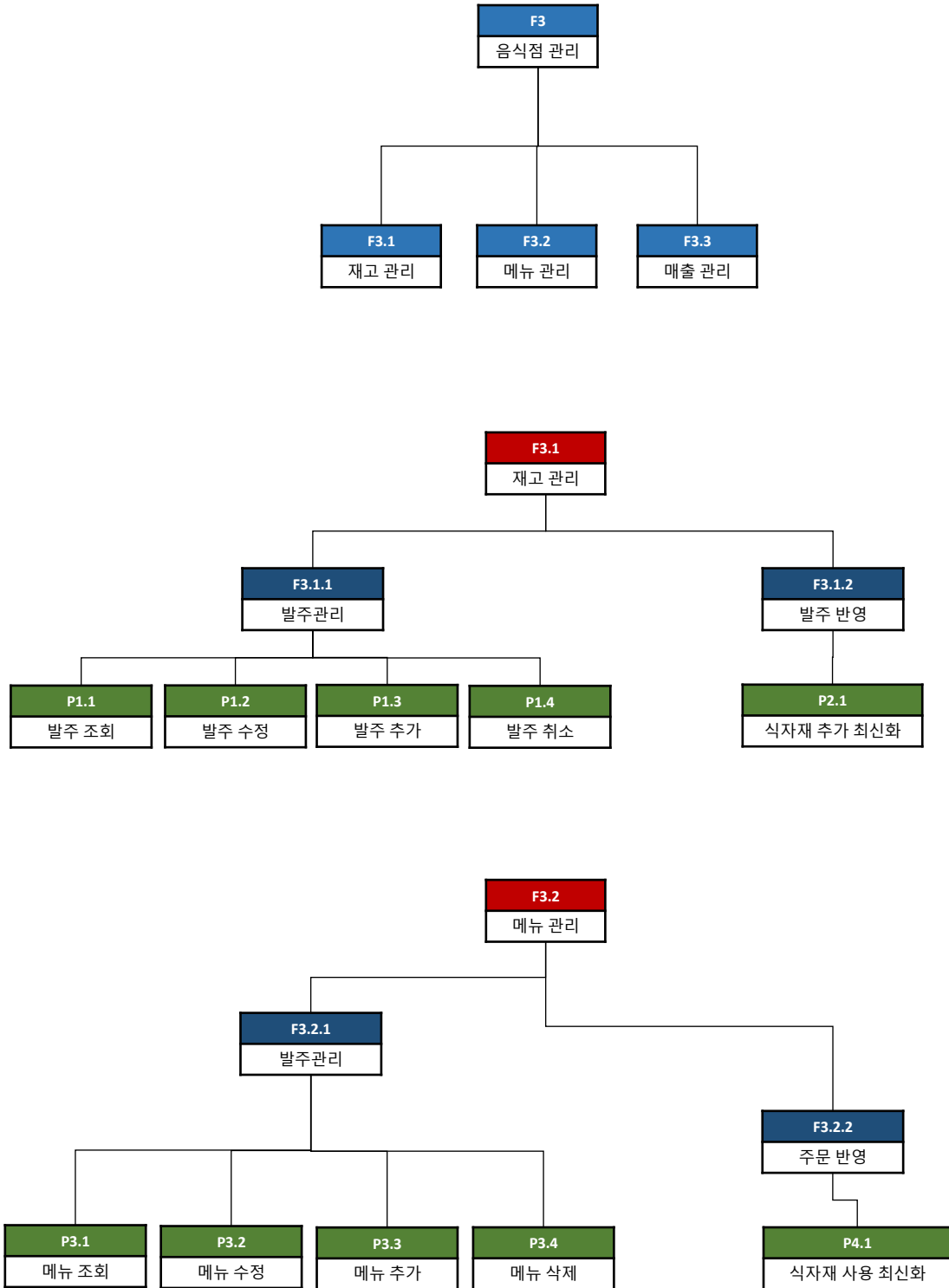
메뉴관리

1. 음식점은 여러 곳이 존재할 수 있으며 각 음식점의 메뉴에는 여러 요리정보를 아울러 관리한다.
2. 점주는 자신의 음식점 메뉴에 요리를 추가하고 삭제하거나 요리의 가격을 바꿀 수 있다.
3. 사용자는 동시에 한 음식점에서만 주문을 할 수 있다.
4. 사용자가 음식점에 주문을 하면 주문 하나당 하나 이상의 요리가 포함되어 있다.
5. 주문 내역에는 주문 번호와 총 금액이 포함되어 있다.
6. 음식점이 주문을 처리하기 위해 요리를 진행하면 하나 이상의 식자재를 사용하며, 식자재는 여러 주문에 사용될 수 있다.(사용 안 되는 식자재가 존재할 수도 있다)
7. 메뉴정보에는 요리고유번호, 요리이름, 가격, 필요한 재료의 수량,예상 조리시간, 그리고 몇 인분인지가 존재한다.
8. 주문 결과에 따라서 식자재 정보가 갱신된다

ERD



프로세스 계층도



프로세스 명세서

프로세스명	설명
발주 조회	발주 신청서를 조회한다.
발주 수정	특정 발주 신청서의 내용을 수정한다.
발주 추가	새로운 발주 신청서를 추가한다.
발주 취소	발주 신청서를 취소한다.
식자재 추가 최신화	발주 결과에 따른 식자재 수 변화를 최신화 한다.

프로세스 명세서

프로세스명	설명
메뉴 조회	등록된 메뉴를 조회한다
메뉴 수정	특정 메뉴의 내용을 수정한다
메뉴 추가	새로운 메뉴를 추가한다.
메뉴 삭제	기존 메뉴를 삭제한다.
식자재 사용 최신화	주문 결과에 따른 식자재 수 변화를 최신화 한다.

Logical Design

릴레이션 명세서 (1)

[illegible]

릴레이션 명세서 (2)

[illegible]

릴레이션 명세서 (3)

릴레이션명	order
릴레이션 설명	주문에 대한 정보

릴레이션 속성

[illegible]

릴레이션 명세서 (6)

릴레이션명	Mart
릴레이션 설명	식자재 마트에 대한 정보

릴레이션 속성

[illegible]

[illegible]

모듈 명세서 (1)

모듈명	발주 조회
모듈ID	M1.1.1
설명	사용자가 원하는 발주 정보를 조회

관련정보	입력	출력	비고
	Restaurant -restaurant_id Placing order -p_order_id	발주신청서 -p_order_id, p_order_date, ingredient_name, add_num, mart_name	- 발주신청서 display

모듈 내역

Read
Restaurant
-restaurant_id
Placing order
-p_order_id

Restaurant, Placing order, Mart, Inventory를 전부 natural join 한 table 사용

If : p_order_id 로 식별되는 row 존재

- Display 발주신청서 : 식별된 row의 p_order_id, p_order_date, ingredient_name, add_num, mart_name

Else : 에러문구 출력

모듈 명세서 (2)

모듈명	발주 수정		
모듈ID	M1.1.2		
설명	특정 발주 신청서의 정보를 수정한다.		
관련정보	입력	출력	비고
	Restaurant - restaurant_id Placing order: - p_order_id - add_num Ingredient: - ingredient_name Mart: - mart_name	수정된 발주신청서 - p_order_id, p_order_date, ingredient_name, add_num, mart_name	- 기존의 발주 신청서의 ingredient_name, add_num, mart_name 등을 입력받은 값으로 update - 공란일경우 error(기존것을 유지하려면 그대로 입력해야함) - 수정된 발주신청서 display

모듈 내역

Read

Restaurant

- restaurant_id

Placing order:

- p_order_id

- add_num

Ingredient:

- ingredient_name

Mart:

- mart_name

Restaurant, Placing order, Mart, Inventory를 전부 natural join 한 table 사용

If : p_order_id 로 식별되는 row 존재

- Update 해당 row의 add_num, ingredient_name, mart_name To 입력받은 내용

- Display 수정된 row의 p_order_id, p_order_date, ingredient_name, add_num, mart_name

Else : 에러문구 출력

모듈 명세서 (3)

모듈명	발주 추가
모듈ID	M1.1.3
설명	입력 받은 발주 신청서의 정보를 추가

관련정보	입력	출력	비고
	Restaurant - restaurant_id Placing order: - add_num Ingredient: - ingredient_name Mart: - mart_name	추가된 발주신청서 - p_order_id, p_order_date, ingredient_name, add_num, mart_name	- p_order_id는 고유해야 하므로 마지막으로 추가된 date의 p_order_id에 1 더하여 입력 - p_order_date는 입력 완 료할 시의 시간을 입력 - display 추가된 발주신청 서

모듈 내역

Read

Restaurant

- restaurant_id

Placing order:

- add_num

Ingredient:

- ingredient_name

Mart:

- mart_name

Restaurant, Placing order, Mart, Inventory를 전부 natural join 한 table 사용

If : restaurant_id가 Restaurant 테이블에 존재

- table에서 p_order_id를 오름차 정렬하였을 때 마지막 값 + 1하여 추가되는 p_order_id로 사용

- p_order_date는 입력 완료할 시의 시간으로 사용

- Insert restaurant_id, p_order_id, p_order_date, ingredient_name, add_num, mart_name
into Placing order (이후 다시 join하여 최신화된 table 생성)

- Display 추가된 row의 p_order_id, p_order_date, ingredient_name, add_num, mart_name

Else : 에러문구 출력

모듈 명세서 (4)

모듈명	발주 취소		
모듈ID	M1.1.3		
설명	입력 받은 발주 신청서의 정보를 추가		
관련정보	입력	출력	비고
	Restaurant -restaurant_id Placing order -p_order_id	발주 취소 문구	- 식별된 row 삭제
모듈 내역			

Read

Restaurant

-restaurant_id

Placing order

-p_order_id

Restaurant, Placing order, Mart, Inventory를 전부 natural join 한 table 사용

If : p_order_id 로 식별되는 row 존재

- Drop p_order_id의 row From Placing order (참조하고 있는 FK의 원본 데이터는 유지)

- display 발주 취소 문구

Else : 에러문구 출력

모듈 명세서 (5)

모듈명	식자재 추가 최신화		
모듈ID	M1.2.1		
설명	발주 결과에 따른 식자재 수의 증가 반영		
관련정보	입력	출력	비고
	Restaurant -restaurant_id Placing order -p_order_id	식자재 정보 - ingredient_name - ingredient_type - ingredient_num	- 식자재 정보 display

모듈 내역

Read

Restaurant
-restaurant_id

Placing order
-p_order_id

Restaurant, Placing order, Inventory 를 natural join 한 table 사용

Update Inventory.ingredient_num

To restaurant_id, p_order_id에 의해 식별되는 row 의 add_num값 만큼 해당 row의 ingredient_num값에 더한 값

Display restaurant_id에 대한 식자재 정보

- ingredient_name
- ingredient_type
- ingredient_num

모듈 명세서 (6)

모듈명	메뉴 조회		
모듈ID	M2.1.1		
설명	사용자가 원하는 메뉴 정보를 조회		
관련정보	입력	출력	비고
	Restaurant -restaurant_id	메뉴 조회 - restaurant_id로 식별되는 모든 dish_name, price, cooking time, portion	- 메뉴 조회 display

모듈 내역

Read

Restaurant

-restaurant_id

Restaurant, Menu를 natural join 한 table 사용

If : restaurant_id 로 식별되는 row 존재

- Display : 해당 restaurant_id를 가진 모든 dish_name, price, cooking time, portion

Else : 에러문구 출력

모듈 명세서 (7)			
모듈명	메뉴 수정		
모듈ID	M2.1.2		
설명	메뉴의 특정 요리의 정보를 수정한다.		
관련정보	입력	출력	비고
	Restaurant - restaurant_id Menu - dish_name - price - cooking_time - portion	수정된 요리 정보 - dish_name - price - cooking_time - portion	- 공란일경우 error(기존것을 유지하려면 그대로 입력해야함) - 수정된 요리 정보 display
모듈 내역			
<p>Read</p> <p>Restaurant</p> <p>- restaurant_id</p> <p>Menu</p> <p>- dish_name</p> <p>- price</p> <p>- cooking_time</p> <p>- portion</p> <p>Restaurant, Menu를 natural join 한 table 사용</p> <p>If : restaurant_id, dish_name으로 식별되는 row 존재</p> <p>- Update 해당 row의 price, cooking_time, portion To 입력받은 내용</p> <p>- Display 수정된 row의 dish_name, price, cooking_time, portion</p> <p>Else : 에러문구 출력</p>			

모듈 명세서 (8)			
모듈명	요리 추가		
모듈ID	M2.1.3		
설명	메뉴에 입력 받은 요리의 정보를 추가		
관련정보	입력	출력	비고
	Restaurant - restaurant_id Menu - dish_name - price - cooking_time - portion	추가된 요리 정보 - dish_name - price - cooking_time - portion	- restaurant_id, dish_name으로 식별되는 row 존재할 경우 Error - 갱신된 메뉴 display
모듈 내역			
<p>Restaurant - restaurant_id</p> <p>Menu - dish_name - price - cooking_time - portion</p> <p>Restaurant, Menu를 natural join 한 table 사용</p> <p>If : restaurant_id로 식별되는 row 존재, restaurant_id, dish_name으로 식별되는 row 미존재</p> <p>- Insert dish_name, price, cooking_time, portion into table</p> <p>- Display restaurant_id를 가진 모든 dish_name, price, cooking time, portion</p> <p>Else : 에러문구 출력</p>			

모듈 명세서 (9)

모듈명	요리 삭제
모듈ID	M2.1.4
설명	메뉴에 존재하는 요리 정보를 삭제

관련정보	입력	출력	비고
	Restaurant - restaurant_id Menu - dish_name	요리 삭제 문구	- 식별된 row 삭제

모듈 내역

Read
Restaurant
- restaurant_id
Menu
- dish_name

Restaurant, Menu를 natural join 한 table 사용

If : restaurant_id, dish_name으로 식별되는 row 존재

- Drop restaurant_id, dish_name의 row From table(참조하고 있는 FK의 원본 데이터는 유지)
- display 요리 삭제 문구

Else : 에러문구 출력

모듈 명세서 (10)			
모듈명	식자재 사용 최신화		
모듈ID	M2.2.1		
설명	주문 결과에 따른 식자재 수 감소 반영		
관련정보	입력	출력	비고
	Restaurant - restaurant_id Order - order_id	식자재 정보 - ingredient_name - ingredient_type - ingredient_num	- 식자재 정보 display
모듈 내역			
<p>Read</p> <p>Restaurant</p> <p>- restaurant_id</p> <p>Order</p> <p>- order_id</p> <p>order_id에 식별되는 모든 Using for Cooking 테이블</p> <p>Update Inventory의 해당되는 모든 ingredient_num</p> <p>각각의 order_id, using_num, ingredient_id로 식별되는 ingredient_id에 대한 using_num을 기존의 ingredient_num에 더한 값</p> <p>Display restaurant_id에 대한 식자재 정보</p> <p>- ingredient_name</p> <p>- ingredient_type</p> <p>- ingredient_num</p>			

System Implementation

DB 구현 내역서

테이블명 Restaurant

테이블 스키마

❖ “describe table “을 이용하여 출력된 내용을 캡처하여 제시

```
MariaDB [db2020170822]> describe Restaurant;
+-----+-----+-----+-----+-----+-----+
| Field          | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| restaurant_id  | char(10)  | NO   | PRI | NULL     |       |
| restaurant_name | char(50)  | NO   |     | NULL     |       |
| owner_name      | char(30)  | NO   |     | NULL     |       |
| owner_tel       | int(11)   | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

테이블 데이터

❖ “select * from customer; “을 이용하여 출력된 내용을 캡처하여 제시

```
MariaDB [db2020170822]> SELECT * FROM Restaurant;
+-----+-----+-----+-----+
| restaurant_id | restaurant_name | owner_name | owner_tel |
+-----+-----+-----+-----+
| R001          | 연 어 이 야 기 | 권 민 석   | 1031806103 |
| R002          | 애 기 능 생 활 관 | 백 중 원   | 1012345678 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

DB 구현 내역서

테이블명

Menu

테이블 스키마

❖ “describe table “을 이용하여 출력된 내용을 캡처하여 제시

```
MariaDB [db2020170822]> describe Menu;
+-----+-----+-----+-----+-----+-----+
| Field          | Type      | Null  | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| dish_name      | char(50)   | NO    | PRI  |          |       |
| restaurant_id  | char(10)   | NO    | PRI  |          |       |
| price          | int(11)    | NO    |      | NULL    |       |
| cooking_time   | int(11)    | YES   |      | NULL    |       |
| portion        | int(11)    | YES   |      | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

테이블 데이터

❖ “select * from customer; “을 이용하여 출력된 내용을 캡처하여 제시

```
MariaDB [db2020170822]> SELECT * FROM Menu;
+-----+-----+-----+-----+-----+
| dish_name      | restaurant_id | price | cooking_time | portion |
+-----+-----+-----+-----+-----+
| 김 치 짜 개    | R002          | 24000 | 15           | 1       |
| 된 장 짜 개    | R002          | 9000  | 15           | 1       |
| 불 고 기       | R002          | 24000 | 20           | 1       |
| 비빔밥         | R001          | 15000 | 10           | 1       |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```


DB 구현 내역서

테이블명 Orders

테이블 스키마

❖ “describe table “을 이용하여 출력된 내용을 캡처하여 제시

```
MariaDB [db2020170822]> describe Orders;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| order_id   | char(10)  | NO   | PRI | NULL    |       |
| total_price | int(11)   | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

테이블 데이터

❖ “select * from customer; “을 이용하여 출력된 내용을 캡처하여 제시

```
MariaDB [db2020170822]> select * from Orders;
+-----+-----+
| order_id | total_price |
+-----+-----+
| O001     | 15000       |
| O002     | 24000       |
| O003     | 9000        |
+-----+-----+
3 rows in set (0.00 sec)
```

DB 구현 내역서

테이블명

Inventory

테이블 스키마

❖ “describe table “을 이용하여 출력된 내용을 캡처하여 제시

```
MariaDB [db2020170822]> DESCRIBE Inventory;
```

Field	Type	Null	Key	Default	Extra
ingredient_id	char(10)	NO	PRI	NULL	
ingredient_name	char(50)	NO		NULL	
ingredient_type	char(30)	YES		NULL	
ingredient_num	int(11)	YES		NULL	
restaurant_id	char(10)	NO	PRI	NULL	

테이블 데이터

❖ “select * from customer; “을 이용하여 출력된 내용을 캡처하여 제시

```
MariaDB [db2020170822]> SELECT * FROM Inventory;
```

ingredient_id	ingredient_name	ingredient_type	ingredient_num	restaurant_id
I001	쌀	곡류	106	R001
I002	김치	반찬	164	R001
I003	된장	장류	43	R002
I004	소고기	육류	45	R002

```
4 rows in set (0.00 sec)
```

DB 구현 내역서

테이블명 Placing_order

테이블 스키마

❖ “describe table “을 이용하여 출력된 내용을 캡처하여 제시

```
MariaDB [db2020170822]> DESCRIBE Placing_order;
+-----+-----+-----+-----+-----+-----+
| Field          | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| p_order_id     | int(11)   | NO   | PRI | NULL     | auto_increment |
| p_order_date   | date      | NO   |     | NULL     |               |
| add_num        | int(11)   | NO   |     | NULL     |               |
| restaurant_id  | char(10)  | NO   | MUL | NULL     |               |
| mart_id        | char(10)  | NO   | MUL | NULL     |               |
| ingredient_id  | char(10)  | NO   | MUL | NULL     |               |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

테이블 데이터

❖ “select * from customer; “을 이용하여 출력된 내용을 캡처하여 제시

```
MariaDB [db2020170822]> SELECT * FROM Placing_order;
+-----+-----+-----+-----+-----+-----+
| p_order_id | p_order_date | add_num | restaurant_id | mart_id | ingredient_id |
+-----+-----+-----+-----+-----+-----+
| 1 | 2024-06-25 | 1313 | R002 | M001 | I001 |
| 2 | 2024-06-25 | 1000 | R001 | M001 | I003 |
| 3 | 2024-06-25 | 123 | R001 | M002 | I003 |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

DB 구현 내역서

테이블명 Mart

테이블 스키마

❖ “describe table “을 이용하여 출력된 내용을 캡처하여 제시

```
MariaDB [db2020170822]> describe Mart;
+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| mart_id    | char(10)   | NO   | PRI | NULL     |       |
| mart_name  | char(50)   | NO   |     | NULL     |       |
| location   | char(100)  | YES  |     | NULL     |       |
| tel_num    | int(11)    | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

테이블 데이터

❖ “select * from customer; “을 이용하여 출력된 내용을 캡처하여 제시

```
MariaDB [db2020170822]> SELECT * FROM Mart;
+-----+-----+-----+-----+
| mart_id | mart_name | location | tel_num |
+-----+-----+-----+-----+
| M001    | 마 트 1   | 서 울   | 12345678 |
| M002    | 마 트 2   | 부 산   | 87654321 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

DB 구현 내역서

테이블명 Using_for_cook

테이블 스키마

❖ “describe table “을 이용하여 출력된 내용을 캡처하여 제시

```
MariaDB [db2020170822]> describe Using_for_cook;
+-----+-----+-----+-----+-----+
| Field          | Type      | Null  | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| order_id       | char(10)  | NO    | PRI  |          |       |
| ingredient_id  | char(10)  | NO    | PRI  |          |       |
| quantity       | int(11)   | YES   |      | NULL    |       |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

테이블 데이터

❖ “select * from customer; “을 이용하여 출력된 내용을 캡처하여 제시

```
MariaDB [db2020170822]> SELECT * FROM Using_for_cook;
+-----+-----+-----+
| order_id | ingredient_id | quantity |
+-----+-----+-----+
| O001     | I001         | 2        |
| O002     | I002         | 3        |
| O002     | I004         | 4        |
| O003     | I003         | 1        |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

DB 구현 내역서

테이블명

Do

테이블 스키마

❖ “describe table “을 이용하여 출력된 내용을 캡처하여 제시

```
MariaDB [db2020170822]> describe Do;
```

Field	Type	Null	Key	Default	Extra
dish_name	char(50)	NO	PRI		
restaurant_id	char(10)	NO	PRI		
order_id	char(10)	NO	PRI		

테이블 데이터

❖ “select * from customer; “을 이용하여 출력된 내용을 캡처하여 제시

```
MariaDB [db2020170822]> SELECT * FROM Do;
```

dish_name	restaurant_id	order_id
김치찌개	R002	O002
된장찌개	R002	O003
불고기	R002	O002
비빔밥	R001	O001

4 rows in set (0.00 sec)

소스코드 요약 설명서

Code	Module	Description
image	N/A	홈 화면 중앙의 사진에 대한 파일이 들어있는 폴더.
config.php	N/A	예제 코드 참조(데이터베이스 연결 코드).
footer.php	N/A	모든 페이지의 하단 바 디자인.
header.php	N/A	모든 페이지의 상단 바 디자인. 홈 화면과 세 가지 기능에 접근할 수 있음.
index.php	N/A	홈 화면. 사진과 기타 설명글이 있음.
order_management.php	발주 조회	발주를 관리하는 기능에 대한 페이지. 기본적으로 모든 음식점에 대한 발주 정보를 조회할 수 있으며, 음식점 선택을 통해 해당 음식점에 발주 정보를 선택해서 조회할 수 있음. 모든 발주정보에는 '수정' 과 '삭제' 버튼이 존재하며 각각의 기능으로 연결됨. 하단에는 '추가' 버튼이 존재하여 발주를 추가하는 페이지로 연결됨.
menu_management.php	메뉴 조회	메뉴를 관리하는 기능에 대한 페이지. 기본적으로 모든 음식점에 대한 메뉴 정보를 조회할 수 있으며, 음식점 선택을 통해 해당 음식점에 메뉴 정보를 선택해서 조회할 수 있음. 모든 메뉴정보에는 '수정' 과 '삭제' 버튼이 존재하며 각각의 기능으로 연결됨. 하단에는 '추가' 버튼이 존재하여 메뉴를 추가하는 페이지로 연결됨.
inventory_management.php	N/A	식자재 정보 및 관련 발주와 주문을 모두 조회할 수 있는 페이지. 기본적으로 모든 음식점에 대한 정보가 출력되며, 음식점 선택을 통해 해당 음식점에 대한 정보만 조회할 수 있음. 모든 주문과 발주에는 '반영' 버튼 이 존재하며 관련 기능이 수행 됨.
order_insert.php	발주 추가	추가하고자 하는 발주 정보를 입력받는 페이지로 이동하여 DB에 해당 발주 정보를 추가함. 식자재 종류와 마트 종류는 DB상에 존재하는 데이터에 한해서 선택할 수 있음. 추가하기 버튼을 누르면 해당 트랜지션이 수행되고 정상적으로 commit될 경우 발주 조회페이지로 돌아와 추가된 정보를 확인할 수 있음.
order_delete.php	발주 취소	해당 발주를 취소함. 버튼을 누르면 해당 트랜지션이 수행되고 DB에서 해당 발주 튜플이 삭제됨. 정상적으로 commit될 경우 발주 조회페이지로 돌아와 바뀐 조회정보를 통해 삭제되었음을 확인할 수 있음.
order_modify.php	발주 수정	해당 발주를 수정하는 페이지로 이동하여 수정하고자 하는 발주 정보를 입력받아 DB에 해당 튜플을 수정함. 수정하기 버튼을 누르면 해당 트랜지션이 수행되고 정상적으로 commit될 경우 발주 조회페이지로 돌아와 수정된 정보를 확인할 수 있음.
menu_insert.php	요리 추가	추가하고자 하는 메뉴 정보를 입력받는 페이지로 이동하여 DB에 해당 메뉴 정보를 추가함. 추가하기 버튼을 누르면 해당 트랜지션이 수행되고 정상적으로 commit될 경우 메뉴 조회페이지로 돌아와 추가된 정보를 확인할 수 있음.(필요한 식자재에 대한 정보는 주문사항과 연관이 깊으므로 이번 구현의 범위가 아님. 따라서 추가는 DB에서만 가능함.)
menu_delete.php	요리 삭제	해당 메뉴를 삭제함. 버튼을 누르면 해당 트랜지션이 수행되고 DB에서 해당 메뉴 튜플이 삭제됨. 정상적으로 commit될 경우 메뉴 조회페이지로 돌아와 바뀐 조회정보를 통해 삭제되었음을 확인할 수 있음.
menu_modify.php	메뉴 수정	해당 메뉴를 수정하는 페이지로 이동하여 수정하고자 하는 메뉴 정보를 입력받아 DB에 해당 튜플을 수정함. 수정하기 버튼을 누르면 해당 트랜지션이 수행되고 정상적으로 commit될 경우 메뉴 조회페이지로 돌아와 수정된 정보를 확인할 수 있음.

소스코드 요약 설명서

Code	Module	Description
reflect_order.php	식자재 추가 최신화	발주 정보에 대해 식자재 데이터에 반영하는 기능. 해당 버튼을 누를 시 해당 데이터의 식자재와 그 수량을 그 발주를 넣은 음식점의 자재 데이터에 현재 수량에 더하여 업데이트를 하고, 발주정보는 삭제되는 트랜지션이 수행 됨. 정상적으로 commit될 경우 식자재 관리페이지가 새로고침되어 반영된 식자재 정보를 확인할 수 있음.
reflect_request.php	식자재 사용 최신화	주문 정보에 대해 식자재 데이터에 반영하는 기능. 해당 버튼을 누를 시 해당 데이터의 식자재와 그 수량을 그 주문을 받는 음식점의 자재 데이터에 현재 수량에 빼서 업데이트를 하고, 주문 정보를 삭제하며 , 만약 음수가 될 경우 비정상적으로 종료(롤백)를 하는 트랜지션이 수행 됨. 정상적으로 commit될 경우 식자재 관리 페이지가 새로고침되어 반영된 식자재 정보를 확인할 수 있음.
styles.css	N/A	예제 코드 참조
util.php	N/A	예제 코드 참조