

푸드코드 앱과 데이터베이스 설계

2022 SUMMER

데이터베이스(03) 12 조

이인기 교수

1828013 송혜준 1976239 우송은 2022008 김민지 2071087 배주연 2076364 전해승

목차

요구사항 분석 보고서	1
앱 사용자 화면 설계서	3
ER 다이어그램 보고서	9
릴레이션 사상 보고서	14
정규화 과정 보고서	17
데이터베이스 설계 SQL 문 정리 보고서	19
릴레이션 스키마 정보 & 테이블 생성 코드	19
데이터 삽입(import) 코드	23
인덱스 생성 코드	26
앱 실제 기능 구현 코드	27
인덱스 생성 SQL 문 및 실행 결과	32
고객 유형별 베스트 메뉴	34
SQL 및 실행 결과	34
웹 보고서 링크 및 화면	35
데이터 분석 결과	38
매장의 문제점	39
매출 증대 방안	40
시간대별 주문 건수와 매출	41
SQL 및 실행 결과	41
웹 보고서 링크 및 화면	42
데이터 분석 결과	45
매장의 문제점	46
매출 증대 방안	46

코너 별 메뉴 별 주문 건수와 매출	48
SQL 및 실행 결과	48
웹 보고서 링크 및 화면	49
데이터 분석 결과	52
매장의 문제점	53
매출 증대 방안	54

요구사항 분석 보고서

1. 푸드코트 앱을 사용하기 위해서는 회원가입을 해야 한다. 회원으로 가입하려면 휴대폰 번호, 비밀번호, 성별, 나이, 주소를 입력해야 한다.

- 푸드코트 앱 사용자는 로그인이 되어있지 않으면, 푸드코트 어플의 기능을 사용할 수 없다. 회원가입이 되어 있을 경우 본인의 휴대폰 번호와 비밀번호로 로그인한다. 회원가입이 되어 있지 않은 경우, 휴대폰 번호, 비밀번호, 성별, 나이, 주소를 앱에서 입력하여 가입한다. 이때 주소는 시와 구로 세분하여 나타낸다.

2. 회원은 휴대폰 번호로 식별한다.

- 회원의 휴대폰 번호는 ID 로 사용되기 때문에, 중복된 휴대폰 번호로 새로 회원가입을 할 수 없다.

3. 주문은 주문번호로 식별한다.

4. 주문은 주문번호, 주문 시간 정보, 식사유형, 완료여부를 유지해야 한다.

- 회원은 주문을 할때, 식사 유형을 선택할 수 있다. 주문의 식사유형으로는 코트에서 식사 또는 포장에 있다.
- 음식이 완성되었을 때, 푸드코트 코너 측에서 주문번호를 단말기에 입력하면, 해당 주문의 완료여부가 O 로 바뀐다. 주문의 완료여부가 O 로 바뀌면, 앱은 해당 주문을 한 회원에게 음식이 완성되었다는 알림을 전송한다.

5. 각 주문의 주문번호는 주문이 들어온 순서대로 값이 부여된다.

- 주문번호는 첫 주문일 경우 1 번이 부여되며, 그 다음 주문부터는 이전 주문번호에서 1 씩 증가한다.

6. 주문이 생성될 때 주문 시간 정보는 실시간으로 자동 입력된다.

- 주문의 주문 시간 정보는 주문날짜와 주문시각으로 세분화한다.

7. 한 회원은 여러 번 주문을 할 수 있고, 각 주문은 한 회원에게만 속한다.

8. 푸드코트 코너는 코너명을 유지해야하고 코너명으로 식별한다.

- 푸드코트 코너에는 중국집, 한식집, 왕돈까스, 샌드위치, 커피 전문점과 같은 5 개 코너가 있다.

9. 푸드코트 메뉴는 메뉴명, 가격을 나타내고 메뉴명으로 식별한다

10. 각 푸드코트 메뉴는 하나의 푸드코트 코너에서만 공급되고, 푸드코트 코너 하나는 여러 푸드코트 메뉴를 공급할 수 있다.

- 푸드코트 메뉴는 총 25 개로, 각 푸드코트 코너는 5 개의 메뉴를 공급한다.

11. 한 주문에는 하나의 푸드코트 코너만을 선택할 수 있고, 푸드코트 코너는 여러 개의 주문을 가질 수 있다.

- 각 주문은 한 음식 코너의 메뉴들로만 이루어진다. 즉, 다른 코너의 음식을 주문하기 위해서는 새로운 주문을 해야 한다.


12. 한 주문 안에는 (한 코너 내의) 여러 푸드코트 메뉴가 포함될 수 있고, 각 푸드코트 메뉴는 여러 개의 주문에 포함될 수 있다. 이때, 주문에 속하는 메뉴 각각의 수량을 정할 수 있다.

13. 주문한 음식이 나오면 푸드코트 앱으로 알림이 전송된다. 이때 주문번호를 통해 회원에게 주문한 음식이 나왔음을 알려주도록 한다.

14. 한 회원은 한 코너 페이지에서 여러 개의 푸드코트 메뉴를 장바구니에 담을 수 있고, 푸드코트 메뉴 또한 여러 회원들의 장바구니에 담길 수 있다. 이때 메뉴의 수량도 함께 설정한다.

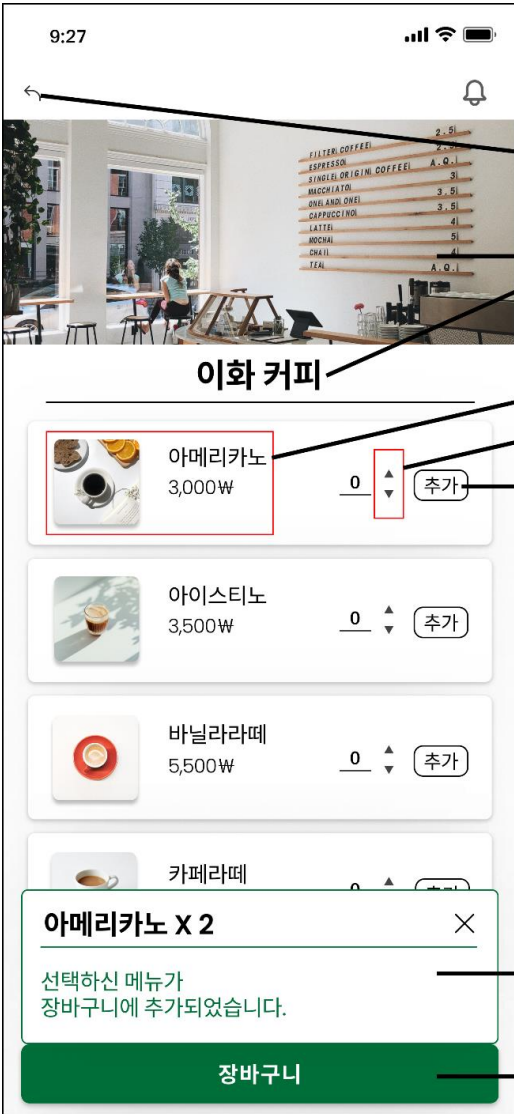
- 장바구니는 임시 저장소로서, 회원은 UI 상 한 코너 페이지 내에서 여러 메뉴를 장바구니에 담을 수 있고, 코너 페이지에서 '뒤로가기' 버튼을 눌러 페이지를 벗어나면 장바구니가 리셋된다.

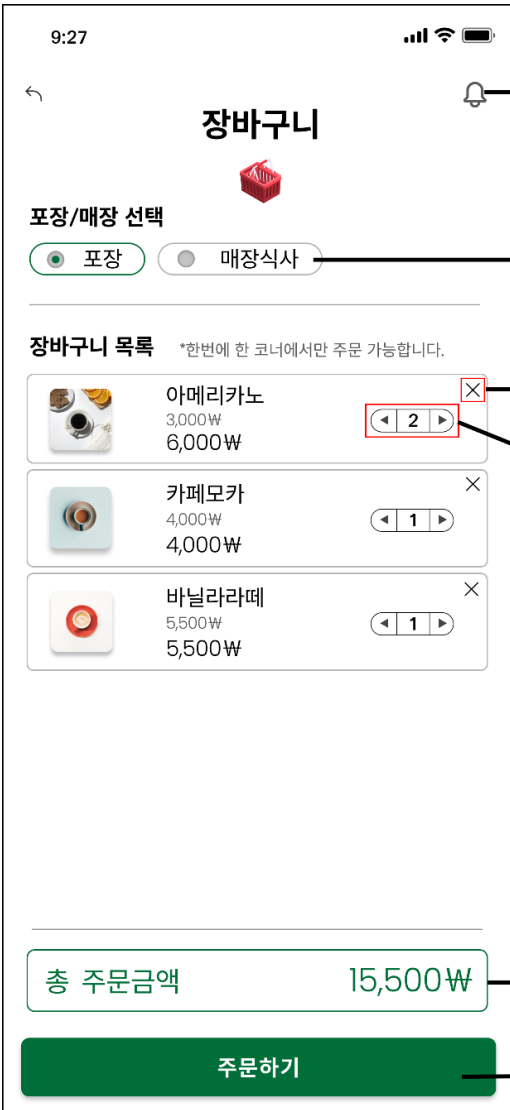
앱 사용자 화면 설계서

화면 ID	LOG	Page name	초기화면	Type	페이지
Navigation	앱 접속				
	정의	LOG - 사용자가 앱의 아이콘을 통해 처음 들어왔을 때 가장 먼저 접하는 페이지			
	디폴트	이화여대의 로고와 앱 이름인 '이대 푸드코트' 명칭 로그인을 위해 핸드폰 번호와 비밀번호를 입력하는 입력 칸 로그인 버튼과 회원가입 버튼			
	1	정보	핸드폰 번호를 입력할 수 있는 입력 칸		
	2	정보	비밀번호를 입력할 수 있는 입력 칸		
	3	정보	로그인을 진행할 수 있는 버튼		
		기능	핸드폰 번호와 비밀번호를 모두 입력했을 경우 활성화된다. 클릭 시 적절한 핸드폰 번호와 비밀번호 여부를 확인, 매칭이 이뤄지지 않을 경우 → 로그인에 이루어지지 않았음을 알리는 팝업 제공, 매칭이 이뤄질 경우 → '코너화면' 페이지로 전환		
	4	정보	회원가입을 진행할 수 있는 버튼		
		기능	'이화여대 푸드코트'에 회원가입이 되어 있지 않는 사용자가 해당 버튼을 클릭하면 '회원정보' 페이지로 전환된다.		

화면 ID	REGISTER	Page name	회원정보	Type	페이지
Navigation	초기화면 페이지				
<div><div>9:27</div><div>회원가입 정보 입력</div><div><div>핸드폰 번호</div><div>010-1234-5678</div></div><div><div>비밀번호</div><div>....</div></div><div><div>비밀번호 재확인</div><div>....</div></div><div><div>성별</div><div>여성</div></div><div><div>주소</div><div>시</div><div>구</div></div><div><div>나이</div><div>22</div></div><div><div><input checked="" type="checkbox"/> 서비스 이용 약관</div><div>자세히 보기</div></div><div><div><input checked="" type="checkbox"/> 개인정보 처리 방침</div><div>자세히 보기</div></div><div>회원가입</div></div>	정의	REGISTER - 회원가입이 되어 있지 않은 사용자가 가입을 위한 개인정보를 입력하고 회원가입을 진행하는 페이지			
	디폴트		핸드폰 번호, 비밀번호, 비밀번호 재확인을 입력할 수 있는 입력 칸 성별, 주소, 나이를 선택할 수 있는 목록 서비스 이용 약관, 개인정보 처리 방침에 동의하는 버튼		
	1	정보	핸드폰 번호를 입력할 수 있는 입력 칸		
	2	정보	비밀번호를 입력할 수 있는 입력 칸		
	3	정보	비밀번호를 재확인할 수 있는 입력 칸		
	4	기능	위에 입력한 비밀번호와 일치하지 않을 경우 주의 문구를 입력 칸 아래 띄운다.		
		기능	성별을 선택할 수 있는 목록 제공		
	5	기능	시, 구를 선택할 수 있는 목록 제공		
	6	기능	나이를 선택할 수 있는 목록 제공		
	7	기능	클릭 시 서비스 이용 약관/개인정보 처리 방침의 자세한 내용을 팝업으로 제공		
	8	정보	서비스 이용 약관/개인 정보 처리 방침에 동의 여부를 선택할 수 있는 버튼		
	9	기능	위의 1~8 에서 입력하거나 선택한 사항에 문제가 없을 경우 버튼이 활성화되고 클릭 시 코너화면으로 전환된다.		

화면 ID	COURT	Page name	코너화면	Type	페이지
Navigation	초기화면 페이지/회원가입 페이지				
	정의		COURT - '이화여대 푸드코트'에 있는 코너를 나열하는 페이지		
	디폴트		카페, 샌드위치, 한식, 돈까스, 중국집의 5 가지 코너 나열 각 코너의 이름과 대표 사진 제공		
	1	정보	각 코너명을 집약적으로 제시하는 버튼		
		기능	버튼을 클릭할 시 각 코너의 '메뉴화면' 페이지로 전환된다.		
	2	정보	각 코너의 대표 사진과 코너명을 요약적으로 제시하는 버튼		
		기능	버튼을 클릭할 시 각 코너의 '메뉴화면' 페이지로 전환된다.		
	3	정보	'코너화면'으로 전환하는 버튼		
		기능	현 화면이 이미 '코너화면' 페이지이므로, 특정한 기능을 수행하지 않는다.		
	4	정보	메뉴 준비 완료 팝업을 띄우는 버튼		
		기능	클릭 시, 주문한 음식이 준비가 완료되었을 경우 → '준비 완료' 팝업 띄움, 주문한 음식이 준비가 완료되지 않았을 경우 → 아직 준비되지 않았음을 알리는 팝업 띄움		
	5	기능	클릭 시, '주문내역' 페이지로 전환된다.		
	6	기능	개인 정보나 기타 앱 설정을 관리할 수 있는 '설정화면' 페이지로 전환		

화면 ID	MENU-01 (01/05)	Page name	메뉴화면	Type	페이지
Navigation	코너화면 페이지				
<div><div>1. 대표 사진과 코너명</div><div>2. 메뉴 이름, 가격, 대표 사진</div><div>3. 수량 조절 버튼</div><div>4. 추가 버튼</div><div>5. 장바구니 버튼</div><div>6. 팝업 메시지</div><div>7. 뒤로가기 버튼</div></div>	정의	MENU-01 - 각 코너의 메뉴 이름과 가격을 나열하는 페이지('01'은 카페를 뜻함)			
	디폴트		각 코너의 대표 사진과 코너명을 제공 각 코너의 메뉴 5 가지와 가격 나열		
	1	정보	코너의 대표 사진과 코너명 제시		
	2	정보	각 메뉴의 이름, 가격, 대표 사진 제시		
	3	정보	각 메뉴의 수량을 조절할 수 있는 버튼		
		기능	위 버튼 클릭 → 수량 증가, 아래 버튼 클릭 → 수량 감소		
	4	정보	각 메뉴를 장바구니에 추가할 수 있는 버튼		
		기능	클릭 시, 수량이 1 이상일 경우 → 장바구니에 추가되고 6 의 팝업을 제시, 수량이 0 일 경우 → 변화 없음		
	5	기능	클릭 시, '결제화면' 페이지로 이동		
	6	정보	3 의 버튼을 클릭할 시 추가된 메뉴를 사용자에게 알리기 위해 나타나는 팝업		
		기능	추가된 메뉴의 이름, 수량을 가시적으로 보여준다. '장바구니' 버튼 바로 위에 나타나며, 엑스 버튼을 눌러 팝업을 제거할 수 있다.		
	7	정보	뒤로가기 버튼		
		기능	해당 코너에서 장바구니에 넣었던 모든 메뉴 목록이 삭제된다.		

화면 ID	CART	Page name	장바구니	Type	페이지		
Navigation	메뉴화면 페이지						
				정의	CART - 주문하기 전, 주문 옵션을 선택하고 선택한 메뉴들의 목록 및 총 가격을 확인하는 페이지		
				디폴트		선택한 메뉴 목록 나열 메뉴들의 총가격 제공	
				1	기능	주문할 메뉴의 식사유형 (포장/매장식사)을 선택하는 버튼	
				2	기능	클릭 시 각 메뉴가 장바구니에서 삭제됨	
				3	기능	각 메뉴의 수량을 조절하는 버튼 왼쪽 클릭 시 → 수량 감소 오른쪽 클릭 시 → 수량 증가	
				4	정보	선택한 코너에서 담은 메뉴들의 총 가격 정보 제공	
				5	기능	클릭 시, '주문내역' 페이지로 이동	
				6	정보	최신 알림 버튼	
				기능	클릭 시, 가장 최근에 받은 '주문알림'을 확인할 수 있다.		

화면 ID	ORDERED	Page name	주문내역/주문알림	Type	페이지/알림
Navigation	장바구니 페이지/				
<div><div><div><div><div>9:27</div><div>주문이 완료되었습니다</div><div>주문번호: 93번</div><div><div><div><div>이화 커피</div><div><div>아메리카노 X 2</div><div>카페 모카 X 1</div><div>바닐라 라떼 X 1</div></div></div></div></div><div><div>홈 화면</div><div>알림</div><div>주문내역</div><div>설정</div></div></div></div><div><div><div>9:27</div><div>주문이 완료되었습니다</div><div>주문번호: 93번</div><div><div><div>주문번호 93</div><div>주문하신 메뉴가 모두 준비 완료되었습니다</div><div>픽업대에서 메뉴를 픽업해주세요. 픽업대 방문시 마스크를 착용해주세요!</div></div></div></div><div><div>홈 화면</div><div>알림</div><div>주문내역</div><div>설정</div></div></div></div></div>					
정의		ORDERED - 주문을 완료한 후, 해당 주문건의 메뉴 목록과 부여된 주문번호를 확인하고 메뉴 준비 완료 알림을 제공받는 페이지			
디폴트		주문 번호 및 주문한 메뉴 목록 정보 제공 주문 메뉴 준비 완료 시 알림 팝업 제공			
1	정보	해당 주문의 주문 번호			
2	정보	주문한 코너의 대표 이미지			
3	정보	주문한 메뉴의 소속 코너 및 메뉴 목록, 수량			
4	정보	해당 주문의 메뉴가 준비 완료됐을 시 제공되는 주문번호 알림 팝업			

ER 다이어그램 보고서

ER 모델 설계 과정

1. 엔티티 타입 및 애트리뷰트의 식별
1-1. 회원 엔티티
<p>회원에 대한 요구사항 분석서의 내용은 다음과 같다.</p> <p>푸드코트 앱을 사용하기 위해서는 회원가입을 해야 한다. <u>회원</u>으로 가입하려면 <u>휴대폰 번호</u>, <u>비밀번호</u>, <u>성별</u>, <u>나이</u>, <u>주소</u>를 입력해야 한다. <u>회원</u>은 <u>휴대폰 번호</u>로 식별한다.</p> <p>이를 통해 <u>회원 엔티티</u>의 애트리뷰트는 <u>휴대폰번호</u>, <u>비밀번호</u>, <u>성별</u>, <u>나이</u>, <u>주소</u>이며, 회원 엔티티의 기본키는 <u>휴대폰번호</u>가 된다는 것을 알 수 있다. 해당 요구사항을 바탕으로 설계한 ER 다이어그램은 다음과 같다.</p>
1-2. 주문 엔티티
<p>주문에 대한 요구사항 분석서의 내용은 다음과 같다.</p> <p><u>주문</u>은 <u>주문번호</u>로 식별한다. <u>주문</u>은 <u>주문번호</u>, <u>주문 시간 정보</u>, <u>식사유형</u>, <u>완료여부</u>를 유지해야 한다.</p> <p>이를 통해 <u>주문 엔티티</u>의 애트리뷰트는 <u>주문번호</u>, <u>주문 시간 정보</u>, <u>식사유형</u>, <u>완료여부</u>이며, 주문 엔티티의 기본키는 <u>주문번호</u>가 된다는 것을 알 수 있다. 해당 요구사항을 바탕으로 설계한 ER 다이어그램은 다음과 같다.</p>

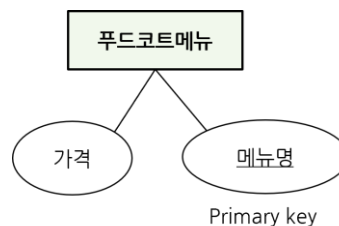
ER 모델 설계 과정

1-3. 푸드코트 메뉴 엔티티

푸드코트 메뉴에 대한 요구사항 분석서의 내용은 다음과 같다.

푸드코트 메뉴는 메뉴명, 가격을 나타내고 메뉴명으로 식별한다.

이를 통해 푸드코트 메뉴 엔티티의 애트리뷰트는 메뉴명, 가격이 되며, 푸드코트 메뉴 엔티티의 기본키는 메뉴명이 된다는 것을 알 수 있다. 해당 요구사항을 바탕으로 설계한 ER 다이어그램은 다음과 같다.

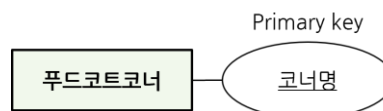


1-4. 푸드코트 코너 엔티티

푸드코트 코너에 대한 요구사항 분석서의 내용은 다음과 같다.

푸드코트 코너는 코너명을 유지해야하고 코너명으로 식별한다.

이를 통해 푸드코트 코너 엔티티의 애트리뷰트는 코너명이 되며, 푸드코트 메뉴 엔티티의 기본키는 코너명이 된다는 것을 알 수 있다. 해당 요구사항을 바탕으로 설계한 ER 다이어그램은 다음과 같다.



2. 관계와 애트리뷰트들을 식별

2-1. 주문-푸드코트 코너 관계

주문-푸드코트 코너 관계에 대한 요구사항 분석서의 내용은 다음과 같다.

한 주문에는 하나의 푸드코트 코너만을 선택할 수 있고, 푸드코트 코너는 여러 개의 주문을 가질 수 있다.

이를 통해 주문과 푸드코트 코너 관계가 N:1 이라고 할 수 있다. 하나의 주문은 하나의 푸드코트 코너에서만 주문 가능하지만, 한 푸드코트 코너에 대한 주문은 여러 개가 가능하기 때문이다. 이때 주문만 전체 참여 관계가 되는데, 그 이유는 모든 주문은 반드시 어떤 한 코너에 속해야 하지만, 어느 한 코너는 주문이 전혀 들어오지 않을 수도 있기 때문이다. 전체 해당 요구사항을 바탕으로 설계한 ER 다이어그램은 다음과 같다.

ER 모델 설계 과정

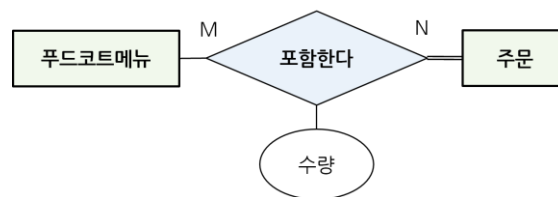


2-2. 주문-푸드코트 메뉴 관계

주문-푸드코트 메뉴 관계에 대한 요구사항 분석서의 내용은 다음과 같다.

한 주문 안에는 (한 코너 내의) 여러 **푸드코트 메뉴**가 포함될 수 있고, 각 **푸드코트 메뉴**는 여러 개의 **주문**에 포함될 수 있다. 이때, 주문에 속하는 메뉴 각각의 수량을 정할 수 있다.

이를 통해 주문과 푸드코트 메뉴 관계가 M:N 이라는 것을 알 수 있다. 하나의 주문 안에 여러 개의 메뉴가 포함될 수 있고, 한 메뉴는 여러 번 주문될 수 있기 때문이다. 이때, 수량 정보를 주문 세부사항 관계에 포함해야 한다. 또한 이 관계에서 주문만이 전체 참여하게 되는데, 그 이유는 주문은 반드시 하나 이상의 메뉴를 가지고 있어야 하지만, 어느 한 메뉴는 주문이 전혀 들어오지 않을 수도 있기 때문이다. 해당 요구사항을 바탕으로 설계한 ER 다이어그램은 다음과 같다.



2-3. 푸드코트 코너-푸드코트 메뉴 관계

푸드코트 코너-푸드코트 메뉴 관계에 대한 요구사항 분석서의 내용은 다음과 같다.

각 **푸드코트 메뉴**는 하나의 **푸드코트 코너**에서만 공급되고, **푸드코트 코너** 하나는 여러 **푸드코트 메뉴**를 공급할 수 있다.

이를 통해 푸드코트 코너와 푸드코트 메뉴 관계가 1:N 관계라는 것을 알 수 있다. 하나의 코너 안에는 여러 메뉴가 속하지만, 한 메뉴는 한 코너에서만 공급되기 때문이다. 이때 푸드코트 코너와 푸드코트 코너는 모두 전체 참여 관계인데, 그 이유는 모든 메뉴는 반드시 하나의 코너에는 속해야 하고, 모든 코너는 반드시 하나 이상의 메뉴를 가지고 있어야 하기 때문이다. 해당 요구사항을 바탕으로 설계한 ER 다이어그램은 다음과 같다.



ER 모델 설계 과정

2-4. 회원-주문 관계

회원-주문 관계에 대한 요구사항 분석서의 내용은 다음과 같다.

한 **회원**은 여러 번 **주문**을 할 수 있고, 각 주문은 한 회원에게만 속한다.

이를 통해 회원과 주문 관계가 1:N 이라는 것을 알 수 있다. 한 명의 회원은 여러 번 주문하는 것이 가능하지만, 한 주문이 여러 회원에게 주문되는 것은 불가능하기 때문이다. 이때 이 관계에서 주문만 전체 참여하게 되는데, 그 이유는 모든 주문은 반드시 주문한 사람, 즉 회원이 있어야 하지만, 회원 중에는 주문을 아직 하지 않은 사람이 있을 수 있기 때문이다. 해당 요구사항을 바탕으로 설계한 ER 다이어그램은 다음과 같다.

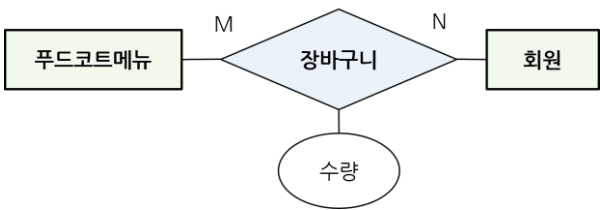


2-5. 회원-푸드코트 메뉴

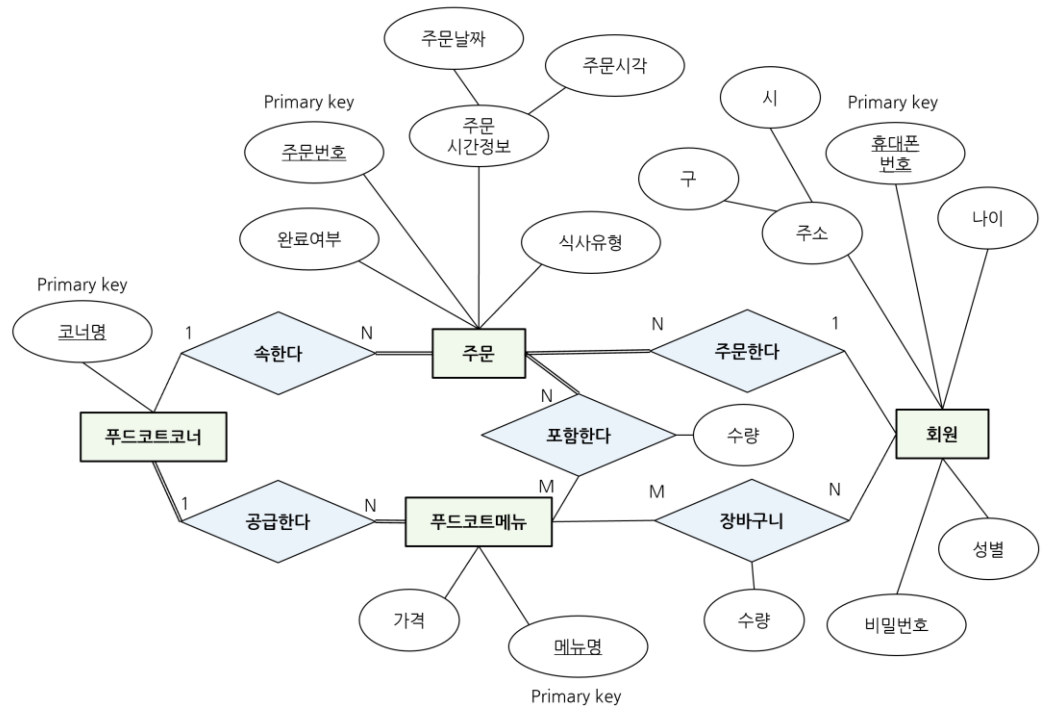
회원-푸드코트 메뉴에 대한 요구사항 분석서의 내용은 다음과 같다.

회원은 주문을 확정짓기 전까지 장바구니에 메뉴들을 담을 수 있다. 이때, 해당 메뉴의 수량도 함께 설정한다. 한 **회원**은 여러 개의 **푸드코트 메뉴**를 장바구니에 담을 수 있고, 푸드코트 메뉴 또한 여러 회원들의 장바구니에 담길 수 있다.

이를 통해 회원과 푸드코트 메뉴 관계는 '장바구니' 관계를 맺고 있으며, M:N 관계라는 것을 알 수 있다. 한 회원은 여러 메뉴를 장바구니에 넣은 것이 가능하고, 한 메뉴 또한 여러 회원의 장바구니에 있는 것이 가능하기 때문이다. 이때 회원과 푸드코트 메뉴는 모두 부분 참여하게 되는데, 어떤 회원이 반드시 장바구니에 메뉴를 담고 있지 않아도 되고, 어떤 메뉴가 반드시 장바구니에 담겨 있을 필요는 없기 때문이다. 해당 요구사항을 바탕으로 설계한 ER 다이어그램은 다음과 같다.



따라서 요구사항 분석서를 바탕으로 ER 모델을 완성한 결과는 다음과 같다.



릴레이션 사상 보고서

릴레이션 사상 과정

단계 1: 정규 엔티티 타입과 단일 값 애트리뷰트
<p>단계 1에서는 ER 스키마에의 각 정규 엔티티 타입에 대해 하나의 릴레이션을 생성하고, 엔티티에 있던 단순 애트리뷰트를 해당 릴레이션에 포함시키고, 복합 애트리뷰트의 경우 그 복합 애트리뷰트를 구성하는 단순 애트리뷰트를 릴레이션에 포함해야 한다. 이때 원래 엔티티의 기본 키가 릴레이션의 기본 키가 된다.</p> <p>ER 다이어그램에 존재하는 모든 엔티티, 즉, ‘회원’, ‘푸드코트 코너’, ‘푸드코트 메뉴’, ‘주문’은 정규 엔티티 타입이므로 각각 릴레이션이 생성되고, 단순 애트리뷰트들은 각 릴레이션에 포함된다. 이때 ‘주소’와 ‘주문 시간 정보’는 복합 애트리뷰트이므로, 이를 구성하는 단순 애트리뷰트들이 릴레이션에 포함된다. 원래 엔티티의 기본 키였던 ‘휴대폰번호’, ‘코너명’, ‘메뉴명’, ‘주문번호’들은 각각의 릴레이션의 기본 키가 된다.</p> <p>따라서 단계 1에서는 다음과 같은 릴레이션들이 사상된다.</p> <div>회원(휴대폰번호, 성별, 나이, 시, 구, 비밀번호) 푸드코트 코너(코너명) 푸드코트 메뉴(메뉴명, 가격) 주문(주문번호, 주문날짜, 주문시각, 식사유형, 완료 여부)</div>
단계 2: 약한 엔티티 타입과 단일 값 애트리뷰트
<p>단계 2에서는 약한 엔티티 타입에 대하여 릴레이션을 생성하고, 엔티티에 있던 단순 애트리뷰트를 해당 릴레이션에 포함시키고, 소유 엔티티 타입에 해당하는 릴레이션의 기본 키를 생성된 릴레이션의 외래 키로 포함시킨다. 생성된 릴레이션의 기본 키는 약한 엔티티 타입의 부분 키와 소유 엔티티 타입의 기본 키를 참조하는 외래 키의 조합이 된다.</p> <p>이때 앞서 본 ER 다이어그램 상에서는 약한 엔티티 타입이 존재하지 않으므로, 단계 2는 거치지 않는다.</p>
단계 3: 2 진 1:1 관계 타입
<p>단계 3에서는 2 진 1:1 관계를 맺고 있는 릴레이션들 중 관계 타입에 완전하게 참여하는 릴레이션을 선택하고, 다른 릴레이션의 기본 키를 선택한 릴레이션의 외래 키로 포함시킨다. 이때 관계 타입이 가지고 있던 모든 단순 애트리뷰트들을 선택한 릴레이션에 포함시킨다.</p> <p>이때 앞서 본 ER 다이어그램 상에서는 2 진 1:1 관계를 맺고 있는 릴레이션이 존재하지 않으므로, 단계 3는 거치지 않는다.</p>

릴레이션 사상 과정

<div>단계 4: 정규 2 진 1:N 관계 타입</div> <div><p>단계 4에서는 정규 2 진 1:N 관계를 맺고 있는 릴레이션들 중 1 측의 참여 엔티티 타입의 기본 키를 N 측의 참여 엔티티 타입에 외래 키로 포함시킨다. 이때 관계 타입이 가지고 있던 모든 단순 애트리뷰트들을 선택한 릴레이션에 포함시킨다.</p><p>ER 다이어그램에 존재하는 정규 2 진 1:N 관계는 ‘푸드코트 코너-주문’, ‘푸드코트 코너-푸드코트 메뉴’, ‘회원-주문’이다. 따라서 1 측의 기본 키인 ‘코너명, 휴대폰번호’를 N 측에 외래 키로 포함시키면 된다.</p><p>따라서 단계 4에서는 다음과 같은 릴레이션들이 사상된다.</p><div><div>회원(휴대폰번호, 성별, 나이, 시, 구, 비밀번호)</div><div>푸드코트 코너(코너명)</div><div>푸드코트 메뉴(메뉴명, 가격, 코너명)</div><div>주문(주문번호, 주문날짜, 주문시각, 식사유형, 완료 여부, 휴대폰번호, 코너명)</div></div></div>
<div>단계 5: 2 진 M:N 관계 타입</div> <div><p>단계 5에서는 2 진 M:N 관계 타입에 대해 새로운 릴레이션을 생성하고, 참여 엔티티 타입에 해당하는 릴레이션들의 기본 키를 생성된 릴레이션의 외래 키로 포함시키고, 이들의 조합을 생성된 릴레이션의 기본 키로 만든다.</p><p>ER 다이어그램에 존재하는 2 진 M:N 관계는 ‘주문-푸드코트 메뉴’, ‘회원-푸드코트 메뉴’이다. 따라서 이를 위해 새로운 릴레이션인 ‘주문세부사항’, ‘장바구니’를 생성하고, ‘메뉴명, 주문번호’, ‘휴대폰번호, 메뉴명’을 각각 외래 키이자 기본 키로 포함시킨다.</p><p>따라서 단계 5에서는 다음과 같은 릴레이션들이 사상된다.</p><div><div>회원(휴대폰번호, 성별, 나이, 시, 구, 비밀번호)</div><div>푸드코트 코너(코너명)</div><div>푸드코트 메뉴(메뉴명, 가격, 코너명)</div><div>주문(주문번호, 주문날짜, 주문시각, 식사유형, 완료 여부, 휴대폰번호, 코너명)</div><div>주문세부사항(메뉴명, 주문번호, 수량)</div><div>장바구니(휴대폰번호, 메뉴명, 수량)</div></div></div>

릴레이션 사상 과정

단계 6: 3 진 이상의 관계 타입

단계 6에서는 3 진 이상의 관계 타입에 대해 새로운 릴레이션을 생성하고, 참여 엔티티 타입에 해당하는 릴레이션들의 기본 키를 생성된 릴레이션의 외래 키로 포함시키고, 이들의 조합을 생성된 릴레이션의 기본 키로 만든다. 만약 1:N:N의 관계일 경우, 1 측의 릴레이션의 기본 키를 참조하는 외래 키를 제외한 나머지 외래 키들의 조합이 생성된 릴레이션의 기본 키가 된다.

이때 앞서 본 ER 다이어그램 상에서는 3 진 이상의 관계를 맺고 있는 릴레이션이 존재하지 않으므로, 단계 6 는 거치지 않는다.

단계 7: 다치 애트리뷰트

단계 7에서는 각 다치 애트리뷰트에 대하여 새로운 릴레이션을 생성하고, 다리 애트리뷰트를 릴레이션에 포함시키고, 다치 애트리뷰트를 애트리뷰트로 갖는 엔티티 타입이나 관계 타입에 해당하는 릴레이션의 기본 키를 생성된 릴레이션에 외래 키로 포함시킨다. 생성된 릴레이션의 기본 키는 다치 애트리뷰트와 외래 키의 조합이 된다.

이때 앞서 본 ER 다이어그램 상에서는 다치 애트리뷰트를 가지고 있는 릴레이션이 존재하지 않으므로, 단계 7 는 거치지 않는다.

따라서 ER 다이어그램을 바탕으로 릴레이션을 사상한 결과는 다음과 같다.



정규화 과정 보고서

1. 제 1 정규형

제 1 정규형의 필요충분조건은 다음과 같다. 모든 애트리뷰트가 **원자값**만을 가져야 한다. 즉 릴레이션의 모든 애트리뷰트에 반복 그룹이 나타나지 않아야 한다. 제 1 정규형의 경우에는 자연스럽게 충족되었는데, 처음 데이터를 생성할 때부터 모든 릴레이션의 모든 애트리뷰트에서 중복된 값을 저장한 경우는 없었기 때문이다.

2. 제 2 정규형

제 2 정규형의 필요충분조건은 다음과 같다. 제 1 정규형을 만족하면서, 어떤 후보 키에도 속하지 않는 모든 애트리뷰트들이 R의 기본 키에 **완전히 함수적으로 종속**해야 한다. 따라서 기본 키가 두개 이상의 애트리뷰트로 구성되었을 경우에만 제 1 정규형이 제 2 정규형을 만족하는가 고려하면 된다. 기존에 구상했던 릴레이션에서는 '주문세부사항' 릴레이션에 '코너분류' 애트리뷰트가 속해 있었는데, 이 애트리뷰트가 제 2 정규형을 거치면서 사라지게 되었다. 그 이유는 '주문세부사항' 릴레이션에 기본 키가 '주문번호'와 '메뉴명'이었는데, '코너분류' 애트리뷰트는 '메뉴명'하나만 있어도 구분된다는 속성을 가지고 있기 때문에 완전히 함수적으로 종속하는 상황이 아니었다. 따라서 해당 애트리뷰트를 제외하게 되었고, 그 결과는 다음과 같다.

관리자: XAMPP for Windows - mysql -u web -pweb_admin

```
Setting environment for using XAMPP for Windows.
user@X3L-28 c:\xampp
# mysql -u web -pweb_admin
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 21
Server version: 10.4.24-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use ewhafoodcourt3;
Database changed
MariaDB [ewhafoodcourt3]> alter table 주문세부사항 drop 코너분류;
Query OK, 0 rows affected (0.031 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [ewhafoodcourt3]> select * from 주문세부사항;
```

주문번호	메뉴명	수량
1	BLT 샌드위치	1
27	BLT 샌드위치	2
74	BLT 샌드위치	2
96	BLT 샌드위치	1
105	BLT 샌드위치	1
146	BLT 샌드위치	2

이외에도 기본 키가 두개 이상의 애트리뷰트로 구성된 릴레이션에는 ‘장바구니’ 릴레이션이 있다. 이 경우에는 ‘수량’ 애트리뷰트가 ‘휴대폰번호’ 혹은 ‘메뉴명’ 하나로만 결정되지 않고, 반드시 ‘휴대폰번호, 메뉴명’으로만 결정되기 때문에 제 2 정규형을 만족하는 상황이다.

3. 제 3 정규형

제 3 정규형의 필요 충분 조건은 다음과 같다. 릴레이션 R 이 제 2 정규형을 만족하면서, 키가 아닌 모든 애트리뷰트가 릴레이션 R 의 기본 키에 **이행적으로 종속하지 않아야** 한다.

이때 모든 릴레이션에서 이미 제 3 정규형을 만족하는 상황이었으므로, 정규화를 거칠 부분이 없었다. 푸드코트 코너의 경우, 애트리뷰트가 하나만 존재하므로 제 3 정규화 과정에 포함되지 않는다. 푸드코트 메뉴의 경우, ‘가격’ 애트리뷰트는 기본 키인 ‘메뉴명’에만 종속하고, ‘코너명’에는 종속할 수 없다. 그 이유는 코너명으로 음식의 가격이 결정되지는 않기 때문이다. 회원의 경우, ‘성별’, ‘나이’, ‘시’, ‘구’, ‘비밀번호’는 전부 다른 애트리뷰트를 결정할 수 없다. 이는 이 애트리뷰트들이 모두 사용자가 입력한 값으로, 중복 가능하기 때문이다. 따라서 기본 키인 ‘핸드폰 번호’에만 종속함을 알 수 있다. 주문의 경우도 마찬가지로, ‘주문날짜’, ‘주문요일’, ‘주문시각’, ‘식사유형’, ‘완료여부’는 모두 다른 애트리뷰트를 결정할 수 없다. 그 이유는 사용자가 각각 독립적으로 입력하는 값이기 때문에 다른 값을 필수적으로 결정하지 않기 때문이다. 따라서 기본 키인 ‘주문번호’만 결정자가 된다. 마지막으로, 주문세부사항의 경우 ‘수량’이 ‘주문번호’나 ‘메뉴명’을 결정할 수 없기 때문에 결정자가 되지 않는다. 이를 통해 제 3 정규형이 만족되었다는 것을 알 수 있다.

데이터베이스 설계 SQL 문 정리 보고서

릴레이션 스키마 정보 & 테이블 생성 코드

1. 정규화 이전

릴레이션 1. 회원

성별	나이	구	시	휴대폰번호	비밀번호
----	----	---	---	-------	------

[SQL 문]

```
create table 회원(
    성별 char(4),
    나이 int,
    시 char(10),
    구 char(10),
    휴대폰번호 varchar(15) primary key,
    비밀번호 char(4) );
```

[실행 결과]

```
MariaDB [ewhafoodcourt]> create table 회원(
-> 성별 char(4),
-> 나이 int,
-> 시 char(10),
-> 구 char(10),
-> 휴대폰번호 varchar(15) primary key,
-> 비밀번호 char(4) );
Query OK, 0 rows affected (0.026 sec)
```

릴레이션 2. 푸드코트코너

코너명

[SQL 문]

```
create table 푸드코트코너(
    코너명 char(10) primary key);
```

[실행 결과]

```
MariaDB [ewhafoodcourt]> create table 푸드코트코너(
-> 코너명 char(10) primary key
-> );
Query OK, 0 rows affected (0.073 sec)
```

릴레이션 3. 푸드코트메뉴

코너명	메뉴명	가격
-----	-----	----

[SQL 문]

```
create table 푸드코트메뉴(
    코너명 char(10),
    메뉴명 char(15) primary key,
    가격 int,
    foreign key(코너명) references 푸드코트코너(코너명));
```

[실행 결과]

```
MariaDB [ewhafoodcourt]> create table 푸드코트메뉴(
-> 코너명 char(10),
-> 메뉴명 char(15) primary key,
-> 가격 int,
->
-> Foreign key(코너명) references 푸드코트코너(코너명) );
Query OK, 0 rows affected (0.083 sec)
```

릴레이션 4. 주문

코너	주문번호	주문날짜	주문시각	식사유형	주문자번호	완료여부
----	------	------	------	------	-------	------

[SQL 문]

```
create table 주문(
    코너 char(10),
    주문번호 int not null auto_increment,
    주문날짜 date default current_date,
    주문시각 time default current_time,
    식사유형 char(4),
    주문자번호 varchar(15),
```

완료여부 char(1),

primary key(주문번호),

foreign key(코너) references 푸드코트코너(코너명),

foreign key(주문자번호) references 회원(휴대폰번호));

**** auto_increment** : 실시간으로 주문이 들어올 때마다 해당 주문에 부여되는 주문번호가 1 씩 증가하여 주문 순서를 나타낼 수 있도록 한다.

**** current_date, current_time** : 실시간으로 주문이 들어올 때 해당 주문이 생성된 날짜와 시각이 자동 입력된다.

- 실행 결과

```
MariaDB [ewhafoodcourt]> create table 주문(
-> 코너 char(10),
-> 주문번호 int not null auto_increment,
-> 주문날짜 date default current_date,
-> 주문시각 time default current_time,
-> 식사유형 char(4),
-> 주문자번호 varchar(15),
-> 완료여부 char(1),
->
-> primary key(주문번호),
-> foreign key(코너) references 푸드코트코너(코너명),
-> foreign key(주문자번호) references 회원(휴대폰번호) );
Query OK, 0 rows affected (0.029 sec)
```

릴레이션 5. 주문세부사항

주문번호	메뉴명	코너분류	수량
------	-----	------	----

[SQL 문]

```
create table 주문세부사항(
주문번호 int,
메뉴명 char(15),
코너분류 char(10),
수량 int,
primary key(주문번호, 메뉴명),
foreign key(주문번호) references 주문(주문번호),
foreign key(메뉴명) references 푸드코트메뉴(메뉴명) );
```


[실행 결과]

```
MariaDB [ewhafoodcourt]> create table 주문세부사항(
-> 주문번호 int,
-> 메뉴명 char(15),
-> 코너분류 char(10),
-> 수량 int,
-> primary key(주문번호, 메뉴명),
-> foreign key(주문번호) references 주문(주문번호),
-> foreign key(메뉴명) references 푸드코트메뉴(메뉴명)
-> );
Query OK, 0 rows affected (0.030 sec)
```

릴레이션 6. 장바구니

<u>회원번호</u>	<u>메뉴명</u>	수량
-------------	------------	----

[SQL 문]

```
create table 장바구니(
    회원번호 varchar(15),
    메뉴명 char(15),
    수량 int,
    primary key(회원번호, 메뉴명),
    foreign key(회원번호) references 회원(휴대폰번호),
    foreign key(메뉴명) references 푸드코트메뉴(메뉴명));
```

```
alter table 장바구니 add unique(회원번호, 메뉴명);
```

[실행 결과]

```
MariaDB [ewhafoodcourt]> create table 장바구니(
-> 회원번호 varchar(15),
-> 메뉴명 char(15),
-> 수량 int,
-> primary key(회원번호, 메뉴명),
-> foreign key(회원번호) references 회원(휴대폰번호),
-> foreign key(메뉴명) references 푸드코트메뉴(메뉴명) );
Query OK, 0 rows affected (0.048 sec)

MariaDB [ewhafoodcourt]> alter table 장바구니 add unique(회원번호, 메뉴명);
Query OK, 0 rows affected (0.032 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

2. 정규화 이후 변경된 릴레이션

릴레이션 5. 주문세부사항

주문번호	메뉴명	코너분류(삭제)	수량
------	-----	----------	----

[SQL 문]

alter table 주문세부사항 drop column 코너분류;

[실행 결과]

```
MariaDB [ewhafoodcourt]> alter table 주문세부사항 drop column 코너분류;
Query OK, 0 rows affected (0.017 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

데이터 삽입(import) 코드

[엑셀 파일을 이용한 데이터 삽입 과정]

- 1) 각 릴레이션별로 생성한 엑셀 파일에 데이터를 작성 및 입력함.
- 2) 각 엑셀파일을 csv 파일로 변환함.
- 3) 다음 SQL 문을 활용하여 각 csv 파일을 DB 의 릴레이션으로 import 함.

```
LOAD DATA LOCAL INFILE '파일 경로' replace into table `테이블명` columns terminated by
',' enclosed by '';
```

[SQL 문]

```
LOAD DATA LOCAL INFILE 'C:/Users/user/Desktop/db/회원.csv' replace into table `회원`
columns terminated by ',' enclosed by '';
```

```
LOAD DATA LOCAL INFILE 'C:/Users/user/Desktop/db/푸드코트코너.csv' replace into
table `푸드코트코너` columns terminated by ',' enclosed by '';
```

```
LOAD DATA LOCAL INFILE 'C:/Users/user/Desktop/db/푸드코트메뉴.csv' replace into
table `푸드코트메뉴` columns terminated by ',' enclosed by '';
```

```
LOAD DATA LOCAL INFILE 'C:/Users/user/Desktop/db/주문.csv' replace into table `주문`
columns terminated by ',' enclosed by '';
```

LOAD DATA LOCAL INFILE 'C:/Users/user/Desktop/db/주문세부사항.csv' replace into table '주문세부사항' columns terminated by ',' enclosed by '"';

[실행 결과]

릴레이션 1. 회원

성별

나이

시

구

휴대폰번호

비밀번호

여

51

서울특별시

서초구

010-1097-3391

2996

여

40

서울특별시

동대문구

010-1136-7889

4502

여

29

서울특별시

성북구

010-1143-2689

4803

여

63

서울특별시

강남구

010-1435-4455

2183

여

31

서울특별시

영등포구

010-1492-5944

5850

여

57

서울특별시

중랑구

010-1544-1908

4335

여

29

서울특별시

금천구

010-1689-2750

2757

여

19

서울특별시

마포구

010-1803-3282

8131

여

30

서울특별시

성동구

010-1856-6041

2918

여

39

서울특별시

성북구

010-1943-1798

8074

여

53

서울특별시

노원구

010-1972-5796

3679

여

24

서울특별시

중랑구

010-2022-9353

1471

여

20

서울특별시

강남구

010-2037-7243

8655

여

50

서울특별시

성북구

010-2081-1624

2123

여

49

서울특별시

성북구

010-2089-1213

8803

여

14

서울특별시

도봉구

010-2324-5793

4683

여

56

서울특별시

강서구

010-2355-7099

9347

여

48

서울특별시

관악구

010-2373-8065

2151

여

51

서울특별시

마포구

010-2592-6327

1372

여

57

서울특별시

서대문구

010-2635-7529

6395

여

32

서울특별시

은평구

010-2662-1919

5013

여

44

서울특별시

중랑구

010-2786-5505

6867

여

55

서울특별시

강서구

010-2951-9937

1156

여

17

서울특별시

구로구

010-3242-2453

5422

여

57

서울특별시

강서구

010-3338-1272

7635

여

21

서울특별시

양천구

010-3375-8272

9187

여

63

서울특별시

양천구

010-3503-7181

4261

여

30

서울특별시

노원구

010-3568-2703

4625

여

60

서울특별시

성동구

010-3742-7078

7229

여

14

서울특별시

중구

010-3823-3042

5519

여

35

서울특별시

영등포구

010-3840-4495

2897

여

68

서울특별시

서초구

010-3884-4211

8030

여

33

서울특별시

서대문구

010-5974-3533

5883

여

58

서울특별시

관악구

010-6016-9780

7282

여

55

서울특별시

종로구

010-6140-2479

9557

여

47

서울특별시

강북구

010-6380-8597

2909

여

59

서울특별시

강남구

010-6423-3851

5476

여

19

서울특별시

관악구

010-6506-4980

4411

여

47

서울특별시

마포구

010-6692-3989

9771

여

40

서울특별시

동작구

010-6739-6814

7469

여

21

서울특별시

서초구

010-6788-5410

3664

여

33

서울특별시

서대문구

010-6813-3366

3478

여

63

서울특별시

송파구

010-6813-7291

4296

여

45

서울특별시

구로구

010-7083-1433

6542

여

49

서울특별시

중랑구

010-7096-5257

4006

여

56

서울특별시

강남구

010-7203-5243

3974

여

37

서울특별시

강남구

010-7215-2159

2892

여

18

서울특별시

중랑구

010-7277-4438

4326

여

41

서울특별시

강남구

010-7376-8545

2275

여

16

서울특별시

서대문구

010-7410-2470

8847

여

23

서울특별시

송파구

010-7445-4895

4076

여

25

서울특별시

중대문구

010-7591-4276

2998

여

59

서울특별시

중랑구

010-7770-4022

5623

여

60

서울특별시

노원구

010-7771-3980

9724

여

20

서울특별시

노원구

010-7873-8800

1278

여

20

서울특별시

중구

010-7912-3526

7102

여

33

서울특별시

강남구

010-8527-9549

3791

여

23

서울특별시

성북구

010-8533-3887

4895

여

20

서울특별시

노원구

010-8643-4043

2972

여

46

서울특별시

도봉구

010-8935-2003

1892

여

14

서울특별시

성동구

010-8941-3154

7613

여

52

서울특별시

강남구

010-8987-6458

2236

여

60

서울특별시

영등포구

010-9037-7321

6254

여

63

서울특별시

양천구

010-9197-1343

2371

여

33

서울특별시

노원구

010-9275-1643

4003

여

45

서울특별시

강북구

010-9339-1552

7230

여

25

서울특별시

서초구

010-9439-5197

9089

여

24

서울특별시

종로구

010-9502-3066

5695

여

37

서울특별시

은평구

010-9608-8514

2953

여

51

서울특별시

강남구

010-9653-7230

2659

여

16

서울특별시

종로구

010-9942-5536

8487

100 rows in set (0.000 sec)

릴레이션 2. 푸드코트코너

코너명

비밀번호

샌드위치

2996

왕도넛

4502

중국집

4803

커피

2183

한식집

5850

5 rows in set (0.000 sec)

릴레이션 3. 푸드코트메뉴

코너명

메뉴명

가격

샌드위치

BLT 샌드위치

6000

왕도넛

고구마도넛

8000

중국집

간짜개

6000

중국집

간짜개

15000

커피

바닐라라떼

5500

커피

바닐라라떼

7000

한식집

비빔밥

7000

한식집

순두부찌개

6000

커피

아메리카노

3000

커피

아이스티노

3500

샌드위치

에그샌드위치

3500

왕도넛

왕도넛

6500

중국집

우육면

8000

중국집

짜장면

7000

중국집

짜장면

8000

샌드위치

치즈샌드위치

4000

왕도넛

치즈도넛

8000

왕도넛

치킨도넛

9000

샌드위치

치킨샌드위치

5000

왕도넛

카레도넛

8000

커피

카페라떼

5000

커피

카페모카

5000

중국집

탕수육

12000

샌드위치

햄샌드위치

4000

25 rows in set (0.001 sec)

릴레이션 4. 주문

```
MariaDB [ewhafoodcourt]> LOAD DATA LOCAL INFILE 'C:/Users/user/Desktop/주문세부사항.csv' REPLACE INTO TABLE `주문` COLUMNS TERMINATED BY ',' ENCLOSED BY '' ;
Query OK, 230 rows affected (0.013 sec)
Records: 230 Deleted: 0 Skipped: 0 Warnings: 0
```

MariaDB [ewhafoodcourt]> select * from 주문;

코너	주문번호	주문날짜	주문시각	식사유형	주문지번호	음식명
샌드위치	1	2022-05-16	07:40:56	모장	010-3375-8272	X
한식집	2	2022-05-16	07:14:21	모장	010-4430-7648	X
왕도넛스	3	2022-05-16	08:10:12	모장	010-3823-3042	X
한식집	4	2022-05-16	08:10:42	매장	010-5766-5094	X
샌드위치	5	2022-05-16	08:13:46	매장	010-3840-4495	X
커피	6	2022-05-16	09:50:21	매장	010-9439-5197	X
커피	7	2022-05-16	09:53:00	매장	010-6739-6814	X
왕도넛스	8	2022-05-16	10:26:14	모장	010-7445-4895	X
한식집	9	2022-05-16	10:42:07	모장	010-3503-7181	X
한식집	10	2022-05-16	11:11:11	모장	010-3823-3042	X
중국집	11	2022-05-16	11:32:11	매장	010-2022-3363	X
한식집	12	2022-05-16	11:40:33	매장	010-3338-1272	X
샌드위치	13	2022-05-16	11:50:21	매장	010-7376-8545	X
왕도넛스	14	2022-05-16	12:10:10	매장	010-4131-9971	X
왕도넛스	15	2022-05-16	12:24:18	매장	010-9942-5536	X
한식집	16	2022-05-16	12:30:57	모장	010-9339-1552	X
한식집	17	2022-05-16	12:33:56	모장	010-4430-7648	X
샌드위치	18	2022-05-16	13:00:27	모장	010-3840-4495	X
커피	19	2022-05-16	13:09:56	매장	010-7083-1433	X
한식집	20	2022-05-16	13:22:09	매장	010-1435-4455	X
커피	21	2022-05-16	14:20:07	매장	010-9608-8514	X
커피	22	2022-05-16	15:15:47	매장	010-6788-5410	X
왕도넛스	23	2022-05-16	17:12:18	모장	010-5693-8898	X
왕도넛스	24	2022-05-16	17:30:01	모장	010-7376-8545	X
샌드위치	25	2022-05-16	17:31:45	매장	010-2355-7099	X
한식집	26	2022-05-16	18:01:06	매장	010-3503-7181	X
샌드위치	27	2022-05-16	18:09:41	모장	010-3742-7078	X
한식집	28	2022-05-16	18:23:39	모장	010-7203-5243	X
왕도넛스	29	2022-05-16	18:30:23	매장	010-3242-2453	X
한식집	30	2022-05-16	19:06:08	모장	010-3338-1272	X

커피	194	2022-05-21	17:21:12	매장	010-8841-3154	X
샌드위치	195	2022-05-21	17:48:22	모장	010-9037-7321	X
한식집	196	2022-05-21	18:02:59	매장	010-2951-9937	X
왕도넛스	197	2022-05-21	18:05:54	모장	010-7771-9980	X
한식집	198	2022-05-21	19:23:15	매장	010-3503-7181	X
커피	199	2022-05-21	20:11:45	매장	010-6423-3851	X
커피	200	2022-05-21	20:51:16	모장	010-3975-1477	X
한식집	201	2022-05-22	08:15:54	모장	010-3742-7078	X
왕도넛스	202	2022-05-22	09:23:01	모장	010-7096-5257	X
한식집	203	2022-05-22	09:32:43	매장	010-7215-2159	X
한식집	204	2022-05-22	10:18:10	모장	010-9037-7321	X
커피	205	2022-05-22	10:39:23	모장	010-8649-4083	X
커피	206	2022-05-22	10:45:23	매장	010-4981-8871	X
샌드위치	207	2022-05-22	11:08:42	모장	010-7912-3526	X
중국집	208	2022-05-22	11:23:02	매장	010-2951-9937	X
한식집	209	2022-05-22	11:32:11	모장	010-9339-1552	X
왕도넛스	210	2022-05-22	12:01:08	모장	010-4981-8871	X
한식집	211	2022-05-22	12:09:10	매장	010-7083-1433	X
왕도넛스	212	2022-05-22	12:19:31	모장	010-2355-7099	X
샌드위치	213	2022-05-22	12:29:18	모장	010-3840-4495	X
한식집	214	2022-05-22	12:30:57	모장	010-9037-7321	X
왕도넛스	215	2022-05-22	12:21:18	모장	010-5585-5771	X
샌드위치	216	2022-05-22	12:54:11	매장	010-7591-4276	X
샌드위치	217	2022-05-22	13:01:31	매장	010-5766-5094	X
커피	218	2022-05-22	13:02:12	매장	010-5342-9915	X
왕도넛스	219	2022-05-22	14:34:56	모장	010-9942-5536	X
커피	220	2022-05-22	14:36:31	모장	010-5914-8330	X
커피	221	2022-05-22	14:44:20	매장	010-6788-5410	X
샌드위치	222	2022-05-22	15:23:01	모장	010-5806-5096	X
커피	223	2022-05-22	16:04:12	모장	010-7215-2159	X
샌드위치	224	2022-05-22	17:11:24	매장	010-7912-3526	X
한식집	225	2022-05-22	18:00:01	모장	010-2786-5605	X
왕도넛스	226	2022-05-22	18:08:17	매장	010-7445-4895	X
한식집	227	2022-05-22	18:34:21	매장	010-9608-8514	X
한식집	228	2022-05-22	18:46:13	모장	010-4843-8157	X
한식집	229	2022-05-22	19:44:11	모장	010-9037-7321	X
커피	230	2022-05-22	20:45:27	매장	010-5974-3533	X

230 rows in set (0.001 sec)

릴레이션 5. 주문세부사항

```
MariaDB [ewhafoodcourt]> LOAD DATA LOCAL INFILE 'C:/Users/user/Desktop/주문세부사항.csv' REPLACE INTO TABLE `주문세부사항` COLUMNS TERMINATED BY ',' ENCLOSED BY '' ;
Query OK, 252 rows affected, 252 warnings (0.023 sec)
Records: 252 Deleted: 0 Skipped: 0 Warnings: 252
```

MariaDB [ewhafoodcourt]> select * from 주문세부사항;

주문번호	메뉴명	수량
1	BLT 샌드위치	1
1	치킨샌드위치	1
2	부대찌개	1
3	고구마도넛스	1
3	왕도넛스	1
4	순두부찌개	1
5	햄샌드위치	1
6	아이스티노	1
7	아메리카노	1
8	왕도넛스	1
9	비빔밥	1
10	짜장면	1
11	짜장면	1
11	탕수육	1
12	부대찌개	1
13	치킨샌드위치	1
14	고구마도넛스	1
15	왕도넛스	1
16	순두부찌개	1
17	비빔밥	1
18	햄샌드위치	1
19	바나나라떼	1
20	부대찌개	1
21	아메리카노	1
22	아이스티노	2
23	왕도넛스	1
24	치즈도넛스	1
25	참치밥	1
26	비빔밥	2
27	BLT 샌드위치	2
27	햄샌드위치	2

197	카레도넛스	1
198	비빔밥	1
199	카페라떼	1
200	아메리카노	2
201	부대찌개	1
202	치킨까스	1
203	순두부찌개	1
204	비빔밥	1
205	아이스티노	1
206	카페라떼	1
207	BLT 샌드위치	1
208	완종기	1
209	김치찌개	1
210	왕도넛스	2
211	부대찌개	1
212	고구마도넛스	1
213	햄샌드위치	1
214	원장찌개	1
214	비빔밥	1
215	치킨까스	1
216	참치샌드위치	2
217	에그샌드위치	1
218	아메리카노	1
219	왕도넛스	1
220	아메리카노	1
221	아이스티노	1
222	햄샌드위치	2
223	바나나라떼	1
224	에그샌드위치	2
225	된장찌개	1
225	순두부찌개	1
226	왕도넛스	2
227	순두부찌개	1
228	부대찌개	1
229	비빔밥	1
230	아메리카노	2
230	아이스티노	2

252 rows in set (0.001 sec)

인덱스 생성 코드

[SQL 문]

```
create index 메뉴i on 푸드코트메뉴(메뉴명);
```

[실행 결과]

```
MariaDB [ewhafoodcourt]> create index 메뉴i on 푸드코트메뉴(메뉴명);  
Query OK, 0 rows affected (0.019 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

앱 실제 기능 구현 코드

1. 회원가입

[기능]

*** 기능**

사용자가 회원가입을 완료하면, '회원' 테이블에 해당 회원이 입력한 가입 정보를 포함한 튜플이 삽입된다.

*** SQL 문(예시)**

```
INSERT INTO 회원 (성별, 나이, 시, 구, 휴대폰번호, 비밀번호) VALUES ('여', '22', '서울특별시', '강동구', '010-1234-5678', '0000');
```

[실행 결과 (예시)]

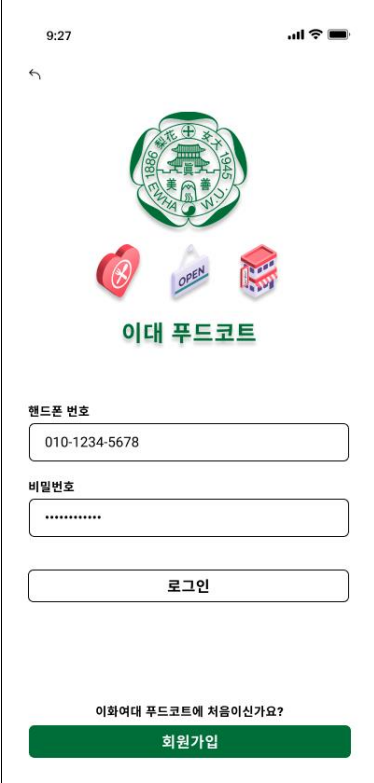
```
MariaDB [ewhafoodcourt]> INSERT INTO 회원(성별,나이,시,구,휴대폰번호,비밀번호) VALUES('여','22','서울특별시','강동구','010-1234-5678','0000');
Query OK, 1 row affected (0.003 sec)

MariaDB [ewhafoodcourt]> SELECT * FROM 회원 WHERE 휴대폰번호='010-1234-5678';

+-----+-----+-----+-----+-----+-----+
| 성별 | 나이 | 시      | 구      | 휴대폰번호 | 비밀번호 |
+-----+-----+-----+-----+-----+-----+
| 여   | 22  | 서울특별시 | 강동구 | 010-1234-5678 | 0000     |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)
```


2. 로그인

[기능]

	<p>* 기능</p> <p>사용자가 로그인을 시도하면, '회원 테이블'에서 휴대폰번호 및 비밀번호 애트리뷰트의 값이 사용자가 입력한 값과 모두 일치하는 튜플의 개수를 조회한다. 조회 결과 1 이 반환되면 로그인 성공, 0 이 반환되면 로그인 실패이다.</p> <p>* SQL 문(예시)</p> <pre>SELECT COUNT(*) FROM 회원 WHERE 휴대폰번호 = '010-1234-5678' AND 비밀번호 = '1234';</pre> <p>Select 문 결과: 1: authentication succeeded 0: authentication failed</p>
--	---

[실행 결과(예시)]

```
MariaDB [ewhafoodcourt]> SELECT COUNT(*)
-> FROM 회원
-> WHERE 휴대폰번호 = '010-1234-5678' AND 비밀번호='1234';
+-----+
| COUNT(*) |
+-----+
|         0 |
+-----+
1 row in set (0.002 sec)
```

3. 장바구니
[기능]

<div><div>9:27</div><div>장바구니</div><div>포장/매장 선택</div><div><div>포장</div><div>매장식사</div></div><div>장바구니 목록</div><div><div>아메리카노</div><div>3,000W</div><div>6,000W</div><div>2</div></div><div><div>카페모카</div><div>4,000W</div><div>4,000W</div><div>1</div></div><div><div>바닐라라떼</div><div>5,500W</div><div>5,500W</div><div>1</div></div><div><div>총 주문금액</div><div>15,500W</div></div><div>주문하기</div></div>	<div><div>* 기능</div><div>1) 회원이 장바구니에 메뉴를 추가하면, DB의 '장바구니' 테이블에 해당 회원의 휴대폰번호와 담은 메뉴의 이름 및 수량 정보가 포함된 튜플이 삽입된다.</div><div>2) 회원이 코너 페이지에서 메뉴 선택 중 뒤로가기를 눌러 페이지를 벗어나면, DB의 '장바구니' 테이블에서 해당 회원의 정보가 포함된 튜플이 모두 삭제된다.</div><div>3) 회원이 장바구니에서 메뉴를 삭제하거나 메뉴 주문을 완료하면, DB의 '장바구니' 테이블에서 해당 메뉴와 회원의 정보가 포함된 튜플이 삭제된다.</div><div>4) 회원이 장바구니에서 담은 메뉴의 수량을 변경하면, DB의 '장바구니' 테이블에서 해당 메뉴와 회원의 정보가 포함된 튜플의 '수량' 애트리뷰트 값이 변경된다.</div></div> <div><div>* SQL 문</div><div>1) 장바구니에 추가시 : INSERT INTO 장바구니(회원번호, 메뉴명, 수량) VALUES ('010-1234-5678', '아메리카노', 2);</div><div>2) 코너페이지에서 뒤로가기 버튼 터치시 or 장바구니 페이지에서 '주문하기' 버튼 터치시: DELETE FROM 장바구니 WHERE 회원번호 = '010-1234-5678';</div><div>3) 개별 메뉴 삭제시: DELETE FROM 장바구니 WHERE 회원번호 = '010-1234-5678' AND 메뉴명 = '바닐라라떼';</div><div>4) 장바구니에서 메뉴 수량 수정시: UPDATE 장바구니 SET 수량 = 1 WHERE 회원번호 = '010-1234-5678' AND 메뉴명 = '아메리카노';</div></div>
--	--

[실행 결과(예시)]

```
MariaDB [ewhafoodcourt]> INSERT INTO 회원(성별,나이,시,구,휴대폰번호,비밀번호) VALUES('여',22,'서울특별시','강동구','010-1234-5678','0000');
Query OK, 1 row affected (0.003 sec)
```

```
MariaDB [ewhafoodcourt]> SELECT * FROM 회원 WHERE 휴대폰번호='010-1234-5678';
```

성별	나이	시	구	휴대폰번호	비밀번호
여	22	서울특별시	강동구	010-1234-5678	0000

1 row in set (0.001 sec)

```
MariaDB [ewhafoodcourt]> SELECT COUNT(*)
-> FROM 회원
-> WHERE 휴대폰번호 = '010-1234-5678' AND 비밀번호='1234';
```

COUNT(*)
0

1 row in set (0.002 sec)

```
MariaDB [ewhafoodcourt]> INSERT INTO 장바구니(회원번호, 메뉴명, 수량) VALUES('010-1234-5678','아메리카노',2);
Query OK, 1 row affected (0.009 sec)
```

```
MariaDB [ewhafoodcourt]> INSERT INTO 장바구니(회원번호, 메뉴명, 수량) VALUES('010-1234-5678','카페모카',1);
Query OK, 1 row affected (0.003 sec)
```

```
MariaDB [ewhafoodcourt]> INSERT INTO 장바구니(회원번호, 메뉴명, 수량) VALUES('010-1234-5678','바닐라라떼',1);
Query OK, 1 row affected (0.003 sec)
```

```
MariaDB [ewhafoodcourt]> SELECT * FROM 장바구니 WHERE 회원번호='010-1234-5678';
```

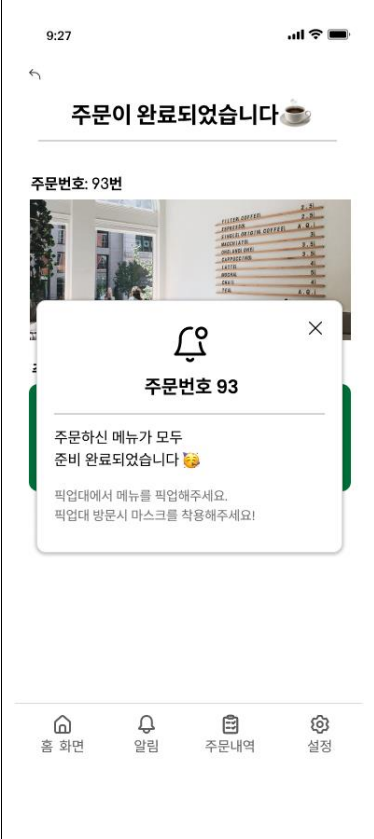
회원번호	메뉴명	수량
010-1234-5678	아메리카노	2
010-1234-5678	카페모카	1
010-1234-5678	바닐라라떼	1

3 rows in set (0.000 sec)

```
Empty set (0.000 sec)
```

4. 주문 준비 완료 알림

[기능]



*** 기능**

1) 푸드코트 측 앱에서 메뉴 준비가 완료된 주문건의 주문번호를 입력하면, DB의 '주문' 테이블에서 해당 주문번호를 키 값으로 갖는 튜플의 '완료여부' 애트리뷰트 값이 X에서 O로 변경된다.

2) DB에서 주문번호 애트리뷰트 값이 푸드코트 측 앱에서 입력한 값과 같고 완료여부 값이 'O'인 튜플의 주문자번호를 조회하면, 해당 주문자번호를 휴대폰번호 애트리뷰트 값으로 갖는 회원의 앱으로 알림을 보낸다.

*** SQL 문(예시)**

```
UPDATE 주문
SET 완료여부 = 'O'
WHERE 주문번호= 93;

SELECT 주문.주문자번호
FROM 주문
WHERE 주문.완료여부 = 'O' and
주문.주문번호 = 93;
```

[실행 결과(예시)]

```
MariaDB [ewhafoodcourt]> UPDATE 주문
-> SET 완료여부 = 'O'
-> WHERE 주문번호=93;
Query OK, 0 rows affected (0.001 sec)
Rows matched: 1 Changed: 0 Warnings: 0

MariaDB [ewhafoodcourt]> SELECT * FROM 주문 WHERE 주문번호=93;
+-----+-----+-----+-----+-----+-----+
| 코너   | 주문번호 | 주문날짜 | 주문시각 | 식사유형 | 주문자번호 | 완료여부 |
+-----+-----+-----+-----+-----+-----+
| 왕돈까스 | 93 | 2022-05-18 | 18:33:27 | 매장   | 010-4981-8871 | 0 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.000 sec)

MariaDB [ewhafoodcourt]> SELECT 주문.주문자번호 FROM 주문
-> WHERE 주문.완료여부 = 'O' AND 주문.주문번호=93;
+-----+
| 주문자번호 |
+-----+
| 010-4981-8871 |
+-----+
1 row in set (0.000 sec)
```

인덱스 생성 SQL 문 및 실행 결과

[SQL 문]

create index 메뉴i on 푸드코트메뉴(메뉴명);

[실행 결과]

```
MariaDB [ewhafoodcourt]> create index 메뉴i on 푸드코트메뉴(메뉴명);
Query OK, 0 rows affected (0.021 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

[인덱스 생성 이유]

인덱스 생성은 검색 속도를 향상시킨다는 장점이 있는 반면, 인덱스를 저장하기 위한 공간이 추가로 필요하고 삽입, 삭제, 수정 연산과 같은 갱신의 속도를 저하시킨다는 단점이 있다.

인덱스의 장점을 이용하기 위해, 웹 보고서 출력을 위한 SQL 문 수행시, 검색 속도 향상을 위한 인덱스를 생성하고자 하였다. 따라서, SQL 문의 JOIN 연산을 수행할때 사용되는 애틀리뷰트에 대한 인덱스 생성을 고려하였다. JOIN 연산 수행시 사용되는 애틀리뷰트로는 (1)주문 릴레이션의 기본 키인 주문번호 (2)회원 릴레이션의 기본 키인 휴대폰번호 (3)푸드코트메뉴 릴레이션의 기본 키인 메뉴명이 있다.

이 중, 주문번호와 휴대폰번호 각각의 애틀리뷰트의 값들끼리는 서로 중복되지 않기 때문에 해당 애틀리뷰트들에 대하여 인덱스를 생성하는 것은 적절하지 않다. 주문번호는 주문 릴레이션의 기본 키이며, 휴대폰번호는 회원 릴레이션의 기본 키이기에 주문번호 애틀리뷰트 값들끼리 서로 중복되지 않으며, 휴대폰번호 애틀리뷰트의 값들도 마찬가지이다. 예를 들어, 만약 주문번호 애틀리뷰트에 대하여 인덱스를 생성할 경우, 현재의 DB 상에서는 서로 중복되지 않는 230 개의 값에 대한 인덱스를 생성하는 것이다. 하지만 인덱스는 저장 공간이 추가로 필요하다는 단점이 있기에 서로 중복되지 않는 값들을 갖는 애틀리뷰트에 대한 인덱스 생성은 비효율적이다.

또한, 주문번호와 휴대폰번호는 각각 매번 새로운 주문이 생성될 때마다, 새로운 회원이 가입할때마다 갱신(갱신 유형: insert)이 이루어진다. 데이터가 많아질수록 저장 공간이 추가로 많이 필요하기에 해당 애틀리뷰트에 대한 인덱스 생성은 비효율적이다. 앞서 언급한 인덱스 생성의 단점으로, 인덱스는 갱신의 속도를 저하시킨다는 점이 있다. 이러한 상황 속에서 다음과 같은 '인덱스 선정 지침'들까지 참고했을 때, 주문번호와 휴대폰번호에 대해 인덱스를 생성하는 것은 적절하지 않다:

- 인덱스 선정 지침 5: 자주 갱신되는 애틀리뷰트에는 인덱스를 정의하지 않는 것이 좋다.
- 인덱스 선정 지침 6: 갱신이 빈번하게 이루어지는 릴레이션에는 인덱스를 많이 만드는 것을 피한다.

반면, 메뉴명은 갱신이 이루어지지 않는 애트리뷰트이기에, 이에 대한 인덱스를 생성하기에 적합하다. 저장 공간의 측면에서도, 상대적으로 애트리뷰트의 값들 자체가 적고 한정되어있기에 효율적이다. 또한, 메뉴명은 CHAR 애트리뷰트로서 다음 인덱스 선정 지침을 만족하기에 인덱스 생성 대상 애트리뷰트로서 적합하다:

- ‘인덱스 선정 지침 10: VARCHAR 애트리뷰트에 대해 인덱스를 생성하지 않는다.

데이터가 적어서 성능 분석은 할 수 없지만 SQL 문에서 인덱스가 where 의 join 에서 활용되는 만큼, 속도가 향상되는데 도움이 될 것으로 예상된다.

고객 유형별 베스트 메뉴

SQL 및 실행 결과

```
CREATE VIEW 나이대_성별_메뉴별

AS SELECT R1.나이대, R1.성별, R1.메뉴명, SUM(R1.수량) as 총수량

FROM (SELECT floor(회원.나이/10)*10 AS 나이대, 회원.성별, 주문세부사항.메뉴명,
주문세부사항.수량

FROM 회원, 주문, 주문세부사항

Where 회원.휴대폰번호 =주문.휴대폰번호

and 주문.주문번호 = 주문세부사항.주문번호) R1

group by R1.나이대, R1.성별, R1.메뉴명;

select 성별, 나이대 ,메뉴명

from(

SELECT 나이대 ,성별, 메뉴명,

ROW_NUMBER() OVER(PARTITION BY 나이대,성별 ORDER BY 총수량 DESC) AS row_id

FROM 나이대_성별_메뉴별) R1

where row_id =1

order by 성별, 나이대;
```

성별	나이대	메뉴명
남	10	치즈돈까스
	20	아이스티노
	30	아메리카노
	40	순두부찌개
	50	부대찌개
	60	부대찌개
여	10	왕돈까스
	20	아이스티노
	30	햄샌드위치
	40	아메리카노
	50	순두부찌개
	60	비빔밥


12 rows in set (0.015 sec)

웹 보고서 링크 및 화면

[웹 보고서 링크]

<http://localhost/ewhafoodcourt/고객유형별.php>

[실행 화면]



데이터베이스 웹 리포트

By 데이터베이스 12조

고객 유형별(성별, 나이) 베스트 메뉴

성별	나이대	메뉴명
남	10	치즈돈까스
남	20	아이스티노
남	30	아메리카노
남	40	순두부찌개
남	50	부대찌개
남	60	부대찌개
여	10	왕돈까스
여	20	아이스티노
여	30	행센드위치
여	40	아메리카노
여	50	순두부찌개
여	60	비빔밥

[결과 분석]

I. 10대 이하의 연령대의 경우, 남녀 모두 돈까스 메뉴를 선호하는 것으로 나타났다. 이는 이와예대 푸트코트 특성상 10대 아하는 이와예대 학생이 아니라 방문객일 확률이 높기 때문으로 보인다. 학생이 아니기 때문에 커피를 주류로 소모할 필요가 없으며, 또한 어린 연령층이기 때문에 한식이나 중식보다는 돈까스 메뉴를 선호하는 경향이 있는 것으로 파악된다.

II. 남녀 모두에서 20대 이상 40대 이하 층은 커피 메뉴를 선호하는 것으로 나타났다. 구체적으로는 여성 20대와 40대, 남성 20대와 30대 층에서 커피 메뉴가 베스트 메뉴로 선정되었다. 이는 20대에서 40대가 주로 학부생, 대학원생과 교수가 분포하는 연령대이기 때문에 카페인 소모량이 많은 것으로 보인다.

III. 커피 메뉴는 20대의 젊은 층에게는 아이스티노와 같은 최근 유행하는 메뉴가, 30대나 40대에게는 아메리카노와 같은 기본적인 메뉴가 인기였다. 이를 통해서 같은 카페고리의 음료에서도 세대 차이가 나타난다는 것을 알 수 있었다.

IV. 남녀 모두에서 40대에서 60대 이상 층은 한식 메뉴를 선호했다. 구체적으로는 여성은 50대와 60대 이상에서, 남성은 40대, 50대, 60대 이상 연령층에서 한식 메뉴를 베스트 메뉴로 꼽았다.

V. 남성의 경우 한식 중에서도 찌개 메뉴를 더욱 선호하는 것을 알 수 있었다. 여성의 경우 한식을 베스트 메뉴로 꼽은 연령층 중 60대 이상은 비빔밥, 즉 찌개가 아닌 메뉴를 선택했지만, 남성의 경우 한식을 베스트 메뉴로 꼽은 모든 연령층이 찌개 종류를 선택했다.

[소스코드]

고객유형별.php

```
<html>
<head>
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
```

```

<link
href="https://fonts.googleapis.com/css2?family=Gowun+Dodum&family=IBM+Plex+Sans+KR:wght@300&family=Noto+Serif+KR&display=swap" rel="stylesheet">
<body bgcolor=white>
<div style="text-align:center; padding-top: 13;">

</div>
<h1 style="font-family: 'Gowun Dodum', sans-serif; color: green;",
align=center>데이터베이스 웹 리포트</h1>
<h5 style="font-family: 'Gowun Dodum', sans-serif; text-align:right; color: 3a3b3c;
font-size: 16; text-shadow:0px 0px 1px lightgreen;">By 데이터베이스 12 조</h5>
<h3 style="font-family: 'Gowun Dodum', sans-serif; border:2px double green; padding:
10px; text-align:center; background-color:#f1ffc4; color:3a3b3c; font-size: 17">고객
유형별(성별, 나이) 베스트 메뉴
</h3>
<?php
$conn=mysqli_connect("localhost", "web", "web_admin", "ewhafoodcourt");

if(!$conn){
    echo "데이터베이스 연결에 문제가 생겼습니다.";}

if(mysqli_connect_errno()){
    echo "연결에 실패했습니다.".mysqli_connect_error();
    exit();
}
$query="select 성별, 나이대 ,메뉴명
from(
SELECT 나이대 ,성별, 메뉴명,
ROW_NUMBER() OVER(PARTITION BY 나이대,성별 ORDER BY 총수량 DESC) AS row_id
FROM 나이대_성별_메뉴별) R1
where row_id =1
order by 성별, 나이대";
$result=mysqli_query($conn, $query);
?>
<table style="margin: auto;"," border=1, bordercolor=black, cellpadding=10,
cellspacing=2>
<tr>

```

```

        <td align="center", valign="center", style="font-family: 'Noto Serif KR', serif;
background-color: #fcf5c7; color: darkgreen; font-weight: bold;">성별</td>
        <td align="center", valign="center", style="font-family: 'Noto Serif KR', serif;
background-color: #fce1e4; color: darkgreen; font-weight: bold;">나이대</td>
        <td align="center", valign="center", style="font-family: 'Noto Serif KR', serif;
background-color: #bbf1f1; color: darkgreen; font-weight: bold;">메뉴명</td>
    </tr>
<?php
    while($row=mysqli_fetch_array($result))
    {
        ?>
        <tr>
            <td align="center", valign="center", style="font-family: 'IBM Plex Sans KR', sans-serif;
font-size: 15; background-color: #fcf5c7;"><?=$row['성별']?></td>
            <td align="center", valign="center", style="font-family: 'IBM Plex Sans KR', sans-serif;
font-size: 15; background-color: #fce1e4"><?=$row['나이대']?></td>
            <td align="center", valign="center", style="font-family: 'IBM Plex Sans KR', sans-serif;
font-size: 15; background-color: #bbf1f1;"><?=$row['메뉴명']?></td>
        </tr>
    <?php } ?>
</table>

<h3 align="center", style="padding-top: 30px; font-family: 'Gowun Dodum', sans-
serif; color:#3a3b3c; ">[결과 분석]</h3>
<div align="left", style="background-color: #f4e7d8; padding-left: 30px; padding-
right:30px; padding-top: 15px; padding-bottom: 15px; margin-left: 30px; margin-right:
30px;">
    <h4 style="font-family: 'IBM Plex Sans KR', sans-serif; color: darkgreen;">I. 10 대
이하의 연령대의 경우, 남녀 모두 돈까스 메뉴를 선호하는 것으로 나타났다. 이는 이화여대
푸트코트 특성상 10 대 이하는 이화여대 학생이 아니라 방문객일 확률이 높기 때문으로 보인다.
학생이 아니기 때문에 커피를 주류로 소모할 필요가 없으며, 또한 어린 연령층이기 때문에
한식이나 중식보다는 돈까스 메뉴를 선호하는 경향이 있는 것으로 파악된다.
    </h4>
    <h4 style="font-family: 'IBM Plex Sans KR', sans-serif; color: darkgreen;">
        II. 남녀 모두에서 20 대 이상 40 대 이하 층은 커피 메뉴를 선호하는 것으로 나타났다.
구체적으로는 여성 20 대와 40 대, 남성 20 대와 30 대 층에서 커피 메뉴가 베스트 메뉴로
선정되었다. 이는 20 대에서 40 대가 주로 학부생, 대학원생과 교수가 분포하는 연령대이기
때문에 카페인 소모량이 많은 것으로 보인다.
    </h4>

```


<h4 style="font-family: 'IBM Plex Sans KR', sans-serif; color: darkgreen;">III. 커피 메뉴는 20 대의 젊은 층에게는 아이스티노와 같은 최근 유행하는 메뉴가, 30 대나 40 대에게는 아메리카노와 같은 기본적인 메뉴가 인기였다. 이를 통해서 같은 카테고리의 음료에서도 세대 차이가 나타난다는 것을 알 수 있었다.

</h4>

<h4 style="font-family: 'IBM Plex Sans KR', sans-serif; color: darkgreen;">IV. 남녀 모두에서 40 대에서 60 대 이상 층은 한식 메뉴를 선호했다. 구체적으로는 여성은 50 대와 60 대 이상에서, 남성은 40 대, 50 대, 60 대 이상 연령층에서 한식 메뉴를 베스트 메뉴로 꼽았다.

</h4>

<h4 style="font-family: 'IBM Plex Sans KR', sans-serif; color: darkgreen;">V. 남성의 경우 한식 중에서도 찌개 메뉴를 더욱 선호하는 것을 알 수 있었다. 여성의 경우 한식을 베스트 메뉴로 꼽은 연령층 중 60 대 이상은 비빔밥, 즉 찌개가 아닌 메뉴를 선택했지만, 남성의 경우 한식을 베스트 메뉴로 꼽은 모든 연령층이 찌개 종류를 선택했다.

</h4>

```
</div>
<?php
mysqli_free_result($result);
mysqli_close($conn);
?>
</body>
</html>
```

데이터 분석 결과

[고객 유형별 베스트 메뉴 표]

성별	나이대	메뉴
여성	10대 이하	왕돈까스
여성	20대	아이스티노
여성	30대	햄샌드위치
여성	40대	아메리카노
여성	50대	순두부찌개
여성	60대 이상	비빔밥
남자	10대 이하	치즈돈까스
남자	20대	아이스티노
남자	30대	아메리카노
남자	40대	순두부찌개
남자	50대	부대찌개
남자	60대 이상	부대찌개

[분석]

- I. 0 대 이하의 연령대의 경우, 남녀 모두 돈까스 메뉴를 선호하는 것으로 나타났다. 이는 이화여대 푸트코트 특성상 10 대 이하는 이화여대 학생이 아니라 방문객일 확률이 높기 때문으로 보인다. 학생이 아니기 때문에 커피를 주류로 소모할 필요가 없으며, 또한 어린 연령층이기 때문에 한식이나 중식보다는 돈까스 메뉴를 선호하는 경향이 있는 것으로 파악된다.
- II. 남녀 모두에서 20 대 이상 40 대 이하 층은 커피 메뉴를 선호하는 것으로 나타났다. 구체적으로는 여성 20 대와 40 대, 남성 20 대와 30 대 층에서 커피 메뉴가 베스트 메뉴로 선정되었다. 이는 20 대에서 40 대가 주로 학부생, 대학원생과 교수가 분포하는 연령대이기 때문에 카페인 소모량이 많은 것으로 보인다.
- III. 커피 메뉴는 20 대의 젊은 층에게는 아이스티노와 같은 최근 유행하는 메뉴가, 30 대나 40 대에게는 아메리카노와 같은 기본적인 메뉴가 인기였다. 이를 통해서 같은 카테고리의 음료에서도 세대 차이가 나타난다는 것을 알 수 있었다.
- IV. 남녀 모두에서 40 대에서 60 대 이상 층은 한식 메뉴를 선호했다. 구체적으로는 여성은 50 대와 60 대 이상에서, 남성은 40 대, 50 대, 60 대 이상 연령층에서 한식 메뉴를 베스트 메뉴로 꼽았다.
- V. 남성의 경우 한식 중에서도 찌개 메뉴를 더욱 선호하는 것을 알 수 있었다. 여성의 경우 한식을 베스트 메뉴로 꼽은 연령층 중 60 대 이상은 비빔밥, 즉 찌개가 아닌 메뉴를 선택했지만, 남성의 경우 한식을 베스트 메뉴로 꼽은 모든 연령층이 찌개 종류를 선택했다.

매장의 문제점

[문제점 1]

중국집 코너의 경우 베스트 메뉴에 선정되지 않았다. 이는 여러가지 원인이 있겠지만, 중국집 코너의 메뉴가 다른 코너의 메뉴 가격과 비교해봤을 때 전체적으로 비싼 것이 주된 원인으로 보인다. 따라서 1) 중국집 메뉴의 가격을 조정하거나, 2) 중국집 코너를 축소 개편할 필요가 있다.

[문제점 2]

베스트 메뉴의 다양성이 부족한 것으로 분석된다. 현재 총 12 개의 베스트 메뉴 중 총 4 개 종류의 메뉴, 즉 아이스티노, 아메리카노, 순두부찌개와 부대찌개가 중복되어 나타나는 것으로 분석되었다. 이 메뉴의 선호도가 높은 이유를 추가적으로 조사하고 그 이유를 다른 메뉴에도 적용하여 고객들이 선호하는 메뉴의 다양성이 높아지게 만들 필요성이 있다.

매출 증대 방안

[방안 1]

여성 20 대와 40 대는 커피를, 여성 30 대는 샌드위치를 가장 많이 소비하는 것으로 나타났다. 이를 고려했을 때 샌드위치와 커피를 같이 소비할 수 있도록 하면 두 코너의 매출이 같이 증대할 것으로 분석된다. 따라서 커피나 샌드위치를 구매할 경우 다음 번 주문에서 사용할 수 있는 샌드위치나 커피 쿠폰을 제공하는 것이 매출 증대에 도움이 될 것이다. 만약 하나의 주문은 한 코너에서만 가능하다는 제약 조건이 사라진다면 베스트 메뉴인 아이스티노, 아메리카노와 햄샌드위치를 할인 세트로 구성하는 방법을 사용할 수 있다.

[방안 2]

남성의 경우 찌개 종류가 인기가 많은 것으로 나타났으므로, 찌개 메뉴에 적용되는 식권을 판매하는 방식을 도입하는 것이 매출 증대에 도움이 될 것으로 분석된다. 즉, 찌개 메뉴 식권 10 장을 구매 시 11 장을 주는 할인을 도입하여 충성심 높은 소비자를 유치하는 방향으로 매출을 높일 수 있다

시간대별 주문 건수와 매출

SQL 및 실행 결과

```
SELECT ROUND(HOUR(L.주문시각) / 2)*2-1 AS 주문시간대_2 시간단위, COUNT(DISTINCT
N.주문번호) AS 총주문량, SUM(N.수량*F.가격) AS 총매출
FROM 주문 AS L, 주문세부사항 AS N, 푸드코트메뉴 AS F
WHERE L.주문시각 BETWEEN '07:00:00' AND '22:00:00'
      AND L.주문번호 = N.주문번호
      AND N.메뉴명 = F.메뉴명
GROUP BY 주문시간대_2 시간단위
ORDER BY 주문시간대_2 시간단위 ASC;
```

*ROUND(HOUR(주문시각) / 2)*2-1 은 시간대를 2 시간 단위로 나누어 준다.

**DISTINCT 는 같은 주문번호로 다른 메뉴를 주문한 중복 케이스를 제외해 준다.

[실행 결과]

주문시간대_2시간단위	총주문량	총매출
7	12	99000
9	37	276500
11	66	619500
13	33	205500
15	11	55000
17	49	337000
19	22	154000

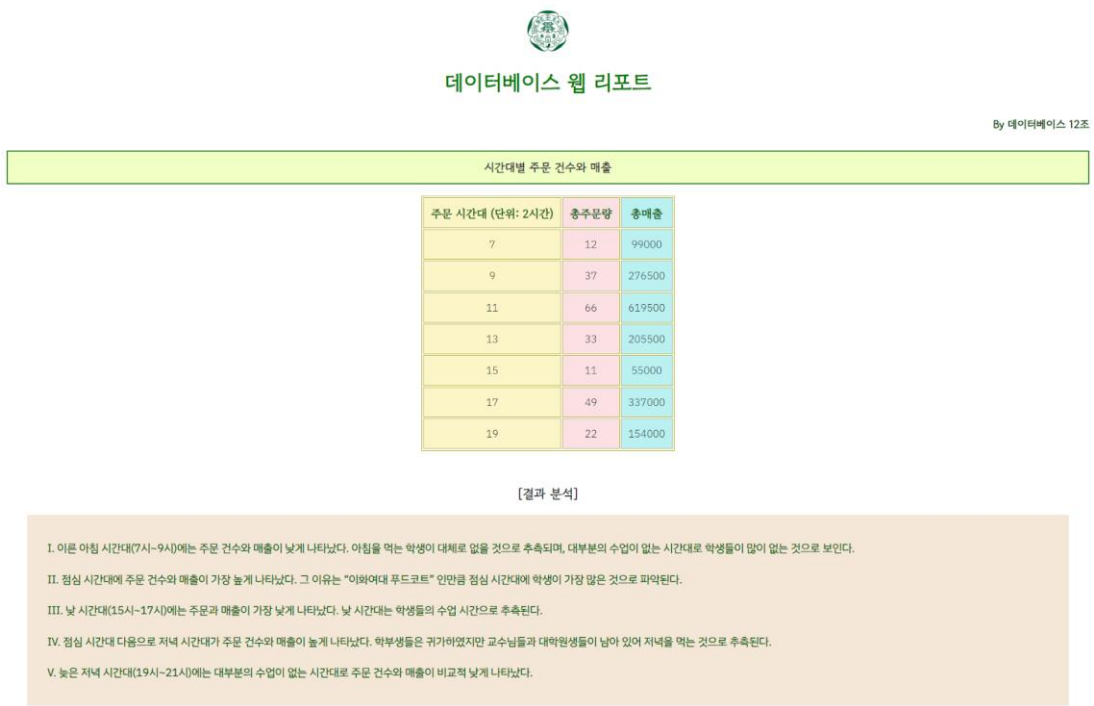
7 rows in set (0.002 sec)

웹 보고서 링크 및 화면

[웹 보고서 링크]

http://localhost/ewhafoodcourt/시간대별.php

[실행 화면]



[소스코드]

시간대별.php

```
<html>
<head>
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Gowun+Dodum&family=IBM+Plex+Sans+KR:wght@300&family=Noto+Serif+KR&display=swap" rel="stylesheet">
<body bgcolor=white>
<div style="text-align:center; padding-top: 13px">
```

```


</div>
<h1 style="font-family: 'Gowun Dodum', sans-serif;color: green;",
align=center>데이터베이스 웹 리포트</h1>
<h5 style="font-family: 'Gowun Dodum', sans-serif; text-align:right; color: 3a3b3c;
font-size: 16; text-shadow:0px 0px 1px lightgreen;">By 데이터베이스 12 조</h5>
<h3 style="font-family: 'Gowun Dodum', sans-serif; border:2px double green; padding:
10px; text-align:center; background-color:#f1ffc4; color:3a3b3c; font-size: 17">시간대별
주문 건수와 매출</h3>
<?php
$conn=mysqli_connect("localhost", "web", "web_admin", "ewhafoodcourt");

if(!$conn){
    echo "데이터베이스 연결에 문제가 생겼습니다.";}

if(mysqli_connect_errno()){
    echo "연결에 실패했습니다.".mysqli_connect_error();
    exit();
}
$query="SELECT ROUND(HOUR(L.주문시각) / 2)*2-1 AS 주문시간대_2 시간단위,
COUNT(DISTINCT N.주문번호) AS 총주문량, SUM(N.수량*F.가격) AS 총매출
FROM 주문 AS L, 주문세부사항 AS N, 푸드코트메뉴 AS F
WHERE L.주문시각 BETWEEN '07:00:00' AND '22:00:00'
AND L.주문번호 = N.주문번호
AND N.메뉴명 = F.메뉴명
GROUP BY 주문시간대_2 시간단위
ORDER BY 주문시간대_2 시간단위 ASC";
$result=mysqli_query($conn, $query);
?>
<table style="margin: auto;", border=1, bordercolor=black, cellpadding=10,
cellspacing=2>
<tr>
<td align="center", valign="center", style="font-family: 'Noto Serif KR', serif;
background-color: #fcf5c7; color: darkgreen; font-weight: bold;">주문 시간대 (단위:
2 시간)</td>
<td align="center", valign="center", style="font-family: 'Noto Serif KR', serif;
background-color: #fce1e4; color: darkgreen; font-weight: bold;">총주문량</td>

```

```

        <td align="center", valign="center", style="font-family: 'Noto Serif KR', serif;
background-color: #bbf1f1; color: darkgreen; font-weight: bold;">총매출</td>
    </tr>
<?php
    while($row=mysqli_fetch_array($result))
    {
        ?>
    <tr>
        <td align="center", valign="center", style="font-family: 'IBM Plex Sans KR', sans-serif;
font-size: 15; background-color: #fcf5c7;"><?=$row['주문시간대_2 시간단위']?></td>
        <td align="center", valign="center", style="font-family: 'IBM Plex Sans KR', sans-serif;
font-size: 15; background-color: #fce1e4"><?=$row['총주문량']?></td>
        <td align="center", valign="center", style="font-family: 'IBM Plex Sans KR', sans-serif;
font-size: 15; background-color: #bbf1f1;"><?=$row['총매출']?></td>
    </tr>
<?php } ?>
</table>

<h3 align="center", style="padding-top: 30px; font-family: 'Gowun Dodum', sans-
serif; color:#3a3b3c; ">[결과 분석]</h3>
<div align="left", style="background-color: #f4e7d8; padding-left: 30px; padding-
right:30px; padding-top: 15px; padding-bottom: 15px; margin-left: 30px; margin-right:
30px;">
    <h4 style="font-family: 'IBM Plex Sans KR', sans-serif; color: darkgreen;">I. 이른 아침
시간대(7 시~9 시)에는 주문 건수와 매출이 낮게 나타났다. 아침을 먹는 학생이 대체로 없을
것으로 추측되며, 대부분의 수업이 없는 시간대로 학생들이 많이 없는 것으로 보인다.
    </h4>
    <h4 style="font-family: 'IBM Plex Sans KR', sans-serif; color: darkgreen;">
        II. 점심 시간대에 주문 건수와 매출이 가장 높게 나타났다. 그 이유는 “이화여대 푸드코트”
인만큼 점심 시간대에 학생이 가장 많은 것으로 파악된다.</h4>
    <h4 style="font-family: 'IBM Plex Sans KR', sans-serif; color: darkgreen;">
        III. 낮 시간대(15 시~17 시)에는 주문과 매출이 가장 낮게 나타났다. 낮 시간대는 학생들의
수업 시간으로 추측된다.
    </h4>
    <h4 style="font-family: 'IBM Plex Sans KR', sans-serif; color: darkgreen;">
        IV. 점심 시간대 다음으로 저녁 시간대가 주문 건수와 매출이 높게 나타났다. 학부생들은
귀가하였지만 교수님들과 대학원생들이 남아 있어 저녁을 먹는 것으로 추측된다.
    </h4>
    <h4 style="font-family: 'IBM Plex Sans KR', sans-serif; color: darkgreen;">

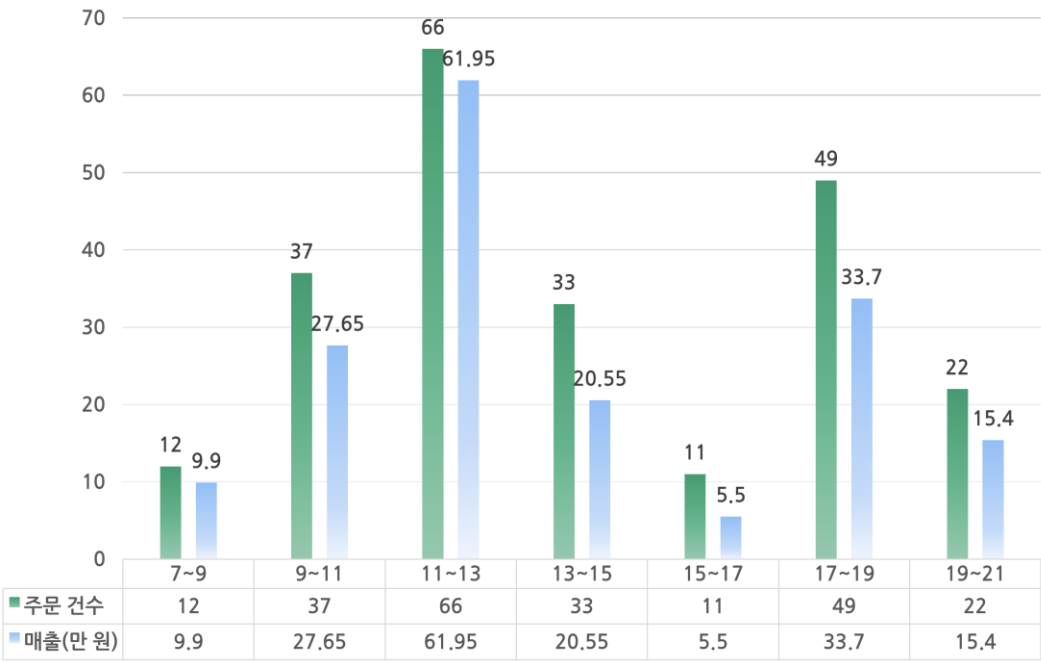
```

V. 늦은 저녁 시간대(19 시~21 시)에는 대부분의 수업이 없는 시간대로 주문 건수와 매출이 비교적 낮게 나타났다.

```
</h4>
</div>
<?php
mysqli_free_result($result);
mysqli_close($conn);
?>
</body>
</html>
```

데이터 분석 결과

[시간대별 주문 건수와 매출 차트]



[분석]

- I. 이른 아침 시간대(7 시~9 시)에는 주문 건수와 매출이 낮게 나타났다. 아침을 먹는 학생이 대체로 없을 것으로 추측되며, 대부분의 수업이 없는 시간대로 학생들이 많이 없는 것으로 보인다.

- II. 점심 시간대에 주문 건수와 매출이 가장 높게 나타났다. 그 이유는 “이화여대 푸드코트” 인만큼 점심 시간대에 학생이 가장 많은 것으로 파악된다.
- III. 낮 시간대(15 시~17 시)에는 주문과 매출이 가장 낮게 나타났다. 낮 시간대는 학생들의 수업 시간으로 추측된다.
- IV. 점심 시간대 다음으로 저녁 시간대가 주문 건수와 매출이 높게 나타났다. 학부생들은 귀가하였지만 교수님들과 대학원생들이 남아 있어 저녁을 먹는 것으로 추측된다.
- V. 늦은 저녁 시간대(19 시~21 시)에는 대부분의 수업이 없는 시간대로 주문 건수와 매출이 비교적 낮게 나타났다.

매장의 문제점

[문제점 1]

7 시~9 시, 15 시~17 시 시간대의 주문량이 다른 시간대에 비해 매우 저조하다. 아침 시간대에는 아침을 먹는 학생이나 사람들이 거의 없는 것으로 파악되며, 낮 시간대는 대부분이 수업 시간으로 인한 것으로 보인다. 따라서 매장에서 아침을 챙기지 않는 이유를 파악하고 보완할 필요가 있다.

[문제점 2]

11 시~13 시, 17 시~19 시 시간대의 주문량으로 보아 매장이 매우 혼잡할 것으로 추측된다. 이러한 이유로 식사를 포기하는 사람들도 있을 것으로 예상된다. 매장의 직원을 늘려야 할 필요가 있다.

매출 증대 방안

[방안 1] - 분석 1, 분석 5

이른 아침 시간대(7 시~9 시)와 늦은 저녁 시간대(19 시~21 시)의 매출을 증대시키기 위해 아침 시간이나 늦은 저녁 시간대에 가볍게 먹을 수 있는 간단한 메뉴를 추가한다.

[방안 2] - 분석 2, 분석 4

점심 시간대(11 시~13 시)와 저녁 시간대(17 시~19 시) 주문량으로 보아 매장이 매우 혼잡할 것이다. 이를 해결하고 피크 시간대의 매출 증대를 위해 다른 시간대의 직원을 점심 시간대와 저녁 시간대로 파트로 변경하거나 추가로 새로운 직원을 고용한다.

[방안 3] - 분석 3

대부분의 수업이 있는 낮 시간대 주문 건수와 매출이 낮다. 수업에 들고 들어갈 수 있거나 가는 길에 먹을 수 있는 가벼운 메뉴를 추가한다.

코너 별 메뉴 별 주문 건수와 매출

SQL 및 실행 결과

```
SELECT F.코너명, N.메뉴명, SUM(N.수량) AS 총주문량, SUM(N.수량*F.가격) AS 총매출
FROM 주문세부사항 AS N, 푸드코트메뉴 AS F
WHERE N.메뉴명 = F.메뉴명
GROUP BY 코너명, 메뉴명
ORDER BY 코너명 ASC, 메뉴명 ASC;
```

[실행 결과]

코너명	메뉴명	총주문량	총매출
샌드위치	BLT 샌드위치	10	60000
샌드위치	에그샌드위치	12	42000
샌드위치	참치샌드위치	5	20000
샌드위치	치킨샌드위치	6	30000
샌드위치	햄샌드위치	27	108000
왕돈까스	고구마돈까스	6	48000
왕돈까스	왕돈까스	25	162500
왕돈까스	치즈돈까스	15	120000
왕돈까스	치킨까스	7	63000
왕돈까스	카레돈까스	7	56000
중국집	깐풍기	2	30000
중국집	우육면	2	16000
중국집	짜장면	5	35000
중국집	짬뽕	5	40000
중국집	탕수육	3	36000
커피	바닐라라떼	10	55000
커피	아메리카노	29	87000
커피	아이스티노	28	98000
커피	카페라떼	9	45000
커피	카페모카	5	25000
한식집	김치찌개	6	36000
한식집	된장찌개	7	42000
한식집	부대찌개	25	175000
한식집	비빔밥	23	161000
한식집	순두부찌개	26	156000

25 rows in set (0.007 sec)

웹 보고서 링크 및 화면

[웹 보고서 링크]

http://localhost/ewhafoodcourt/코너및메뉴별.php

[실행 화면]



데이터베이스 웹 리포트

By 데이터베이스 12조

코너별 메뉴별 주문 건수와 매출

코너명	메뉴명	총주문량	총매출
샌드위치	BLT 샌드위치	10	60000
샌드위치	에그샌드위치	12	42000
샌드위치	참치샌드위치	5	20000
샌드위치	치킨샌드위치	6	30000
샌드위치	햄샌드위치	27	108000
왕돈까스	고구마돈까스	6	48000
왕돈까스	왕돈까스	25	162500
왕돈까스	치즈돈까스	15	120000
왕돈까스	치킨까스	7	63000
왕돈까스	카레돈까스	7	56000
중국집	간통기	2	30000
중국집	우육면	2	16000
중국집	짜장면	5	35000
중국집	짬뽕	5	40000
중국집	왕수육	3	36000
커피	바닐라라떼	10	55000
커피	아메리카노	29	87000
커피	아이스티노	28	98000
커피	카레라떼	9	45000
커피	카레모카	5	25000
한식집	김치찌개	6	36000
한식집	된장찌개	7	42000
한식집	부대찌개	25	175000
한식집	비빔밥	23	161000
한식집	순두부찌개	26	156000

결과 분석

I. 커피와 한식집은 다른 푸드코트에 비해 주문량이 가장 많은 것으로 나타났다.

II. 중국집은 모든 푸드코트 코너 중 주문 건수가 가장 낮게 나타났다.

III. 돈까스 메뉴와 샌드위치 메뉴는 전체적으로 비슷한 주문량을 보였다.

IV. 앞서 보았던 연령별 인기 메뉴가 전체적으로 많은 주문량을 보였다.

[소스코드]

코너및메뉴별.php

```
<html>
<head>
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Gowun+Dodum&family=IBM+Plex+Sans+KR:wght@300&family=Noto+Serif+KR&display=swap" rel="stylesheet">
<body bgcolor=white>
<div style="text-align:center; padding-top: 13;">

</div>
<h1 style="font-family: 'Gowun Dodum', sans-serif;color: green;",
align=center>데이터베이스 웹 리포트</h1>
<h5 style="font-family: 'Gowun Dodum', sans-serif; text-align:right; color: 3a3b3c;
font-size: 16; text-shadow:0px 0px 1px lightgreen;">By 데이터베이스 12 조</h5>
<h3 style="font-family: 'Gowun Dodum', sans-serif; border:2px double green; padding:
10px; text-align:center; background-color:#f1ffc4; color:3a3b3c; font-size: 17">코너별
메뉴별 주문 건수와 매출</h3>
<?php
$conn=mysqli_connect("localhost", "web", "web_admin", "ewhafoodcourt");

if(!$conn){
    echo "데이터베이스 연결에 문제가 생겼습니다.";}

if(mysqli_connect_errno()){
    echo "연결에 실패했습니다.".mysqli_connect_error();
    exit();
}
$query="SELECT F.코너명, N.메뉴명, SUM(N.수량) AS 총주문량, SUM(N.수량*F.가격) AS
총매출
FROM 주문세부사항 AS N, 푸드코트메뉴 AS F
WHERE N.메뉴명 = F.메뉴명
GROUP BY 코너명, 메뉴명
ORDER BY 코너명 ASC, 메뉴명 ASC";
$result=mysqli_query($conn, $query);
```

?>

```
<table style="margin: auto;", border=1, bordercolor=black, cellpadding=10,
cellspacing=2>
```

```
<tr>
```

```
<td align="center", valign="center", style="font-family: 'Noto Serif KR', serif;
background-color: #fcf5c7; color: darkgreen; font-weight: bold;">코너명</td>
```

```
<td align="center", valign="center", style="font-family: 'Noto Serif KR', serif;
background-color: #fce1e4; color: darkgreen; font-weight: bold;">메뉴명</td>
```

```
<td align="center", valign="center", style="font-family: 'Noto Serif KR', serif;
background-color: #c8e7ff; color: darkgreen; font-weight: bold;">총주문량</td>
```

```
<td align="center", valign="center", style="font-family: 'Noto Serif KR', serif;
background-color: #bbf1f1; color: darkgreen; font-weight: bold;">총매출</td>
```

```
</tr>
```

```
<?php
```

```
while($row=mysqli_fetch_array($result))
```

```
{
```

```
?>
```

```
<tr>
```

```
<td align="center", valign="center", style="font-family: 'IBM Plex Sans KR', sans-serif;
font-size: 15; background-color: #fcf5c7;"><?=$row['코너명']?></td>
```

```
<td align="center", valign="center", style="font-family: 'IBM Plex Sans KR', sans-serif;
font-size: 15; background-color: #fce1e4"><?=$row['메뉴명']?></td>
```

```
<td align="center", valign="center", style="font-family: 'IBM Plex Sans KR', sans-serif;
font-size: 15; background-color: #c8e7ff;"><?=$row['총주문량']?></td>
```

```
<td align="center", valign="center", style="font-family: 'IBM Plex Sans KR', sans-serif;
font-size: 15; background-color: #bbf1f1;"><?=$row['총매출']?></td>
```

```
</tr>
```

```
<?php } ?>
```

```
</table>
```

```
<h3 align="center", style="padding-top: 30px; font-family: 'Gowun Dodum', sans-
serif; color:#3a3b3c; ">[결과 분석]</h3>
```

```
<div align="left", style="background-color: #f4e7d8; padding-left: 30px; padding-
right:30px; padding-top: 15px; padding-bottom: 15px; margin-left: 30px; margin-right:
30px;">
```

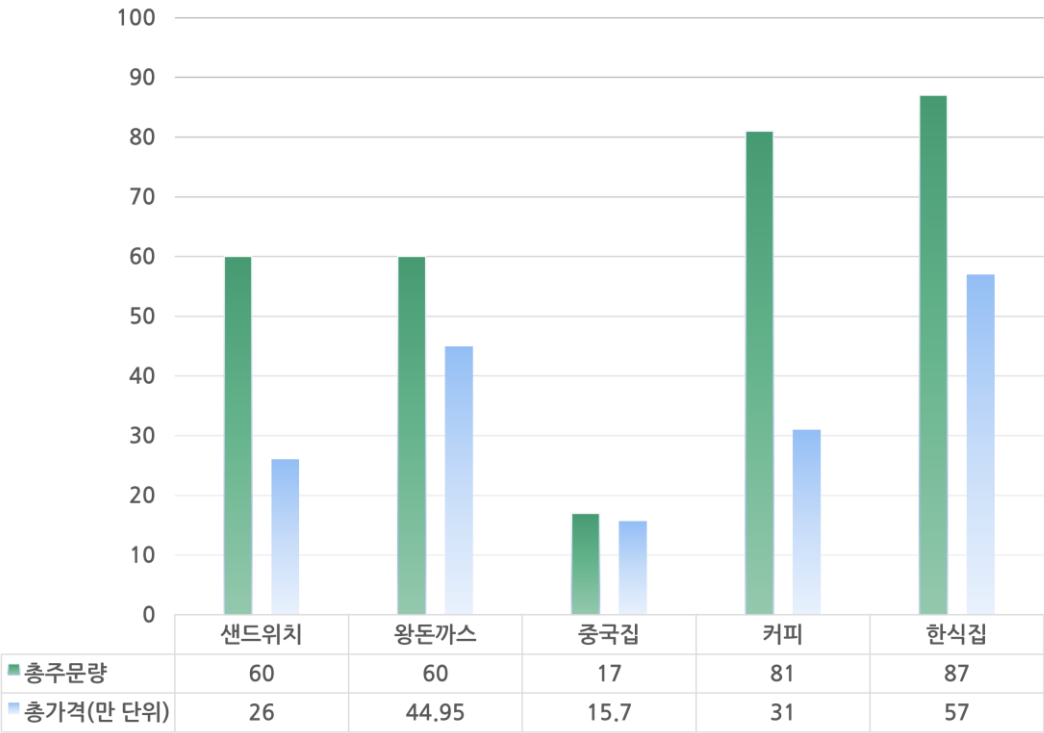
```
<h4 style="font-family: 'IBM Plex Sans KR', sans-serif; color: darkgreen;">I. 커피와
한식집은 다른 푸드코트에 비해 주문량이 가장 많은 것으로 나타났다.</h4>
```

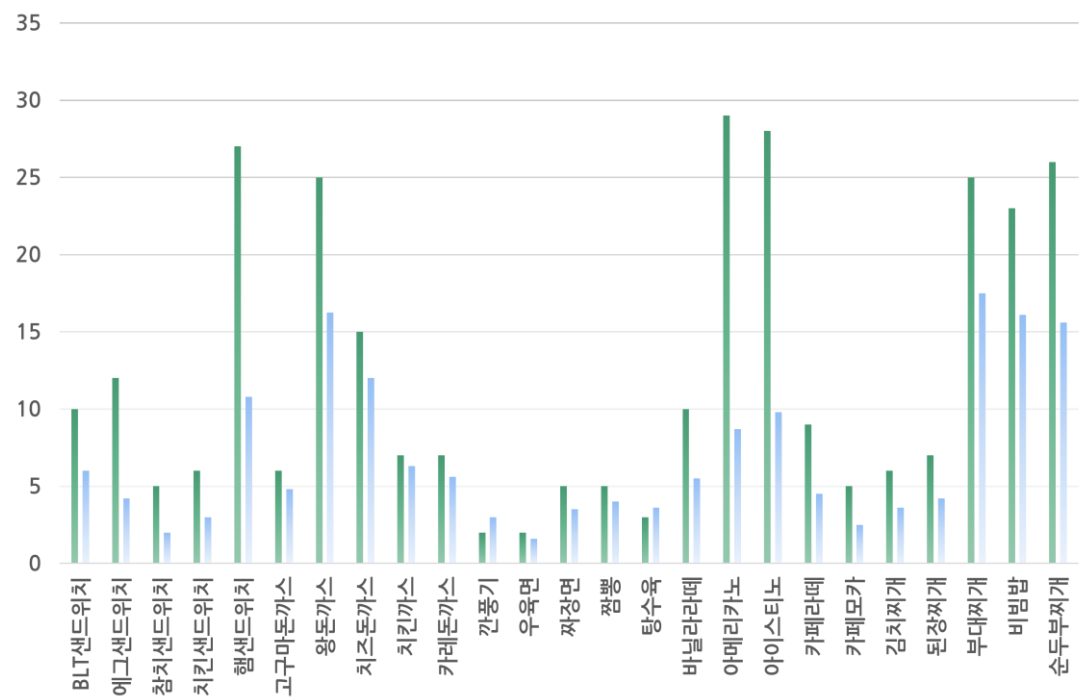
```
<h4 style="font-family: 'IBM Plex Sans KR', sans-serif; color: darkgreen;">
```

II. 중국집은 모든 푸드코트 코너 중 주문 건수가 가장 낮게 나타났다.
</h4>
<h4 style="font-family: 'IBM Plex Sans KR', sans-serif; color: darkgreen;">
III. 돈까스 메뉴와 샌드위치 메뉴는 전체적으로 비슷한 주문량을 보였다.
</h4>
<h4 style="font-family: 'IBM Plex Sans KR', sans-serif; color: darkgreen;">
IV. 앞서 보았던 연령별 인기 메뉴가 전체적으로 많은 주문량을 보였다.</h4>
</div>

```
<?php
mysqli_free_result($result);
mysqli_close($conn);
?>
</body>
</html>
```

데이터 분석 결과
[코너 별 메뉴 별 주문 건수와 매출 차트]





[분석]

- I. 커피와 한식집은 다른 푸드코트에 비해 주문 건수가 장 많은 것으로 나타났다.
- II. 중국집은 모든 푸드코트 코너 중 주문 건수가 가장 적게 나타났다.
- III. 돈까스 메뉴와 샌드위치 메뉴는 전체적으로 비슷한 주문량을 보였다.
- IV. 앞서 보았던 연령별 인기 메뉴가 전체적으로 많은 주문량을 보였다.

매장의 문제점

[문제점 1]

중국집의 모든 메뉴가 다른 코너의 메뉴들에 비해 상대적으로 주문량과 매출이 매우 낮다. 매출이 있기 위해서는 먼저 주문량을 증대시킬 필요가 있다. 중국집의 메뉴 가격 조정이나 다양성 증가, 다양한 이벤트와 같은 주문량을 늘릴 방안이 매우 필요하다.

[문제점 2]

커피 푸드코트는 주문량에 비해 매출이 낮다. 커피의 가격 조정이나 여러 잔을 주문 시 혜택을 주는 방법을 고려해볼 필요가 있다.

[문제점 3]

앞서 보았던 연령별 베스트 메뉴가 전체적으로 많은 주문량을 보였으나 이외의 메뉴 주문량이 저조하다. 베스트 메뉴 이외의 메뉴의 주문량을 증가시킬 방법을 찾는 것이 필요해 보인다.

매출 증대 방안

[방안 1]

연령별 베스트 메뉴의 주문량이 매우 많다. 매출을 증대시키기 위해 주문량이 많은 메뉴의 가격을 500 원씩 인상한다.

[방안 2]

커피 푸드코트 주문량이 많다. 주문량을 더 늘리기 위해 10 잔을 마시면 1 잔 무료인 쿠폰 이벤트를 한다. 또한 커피 주문량에 비해 매출이 적으므로 커피 가격을 500 원 인상한다.

[방안 3]

중국집은 먼저 주문량을 증대시킬 필요가 있다. 따라서 메뉴의 가격을 500 원씩 인하한다. 또한 짜장면과 탕수육, 짬뽕과 간pong 등 세트 메뉴를 출시하여 세트 메뉴 주문 시 할인 이벤트를 한다. 추가로 메뉴의 다양성을 높이기 위해 고객들의 주문 통계량을 분석하여 세 달 간격으로 새로운 메뉴를 도입한다.

[방안 4]

샌드위치 또는 다른 푸드코트에서 식사 후, 커피 푸드코트에서 주문 시 10% 할인하는 방법으로 모든 푸드코트의 주문량과 매출을 증대시킨다.

[방안 5]

코너 별 비인기 메뉴의 주문을 늘리기 위해서 요일을 정해서 비인기 메뉴의 가격을 10% 할인하는 이벤트를 한다.