

목 차

1. 프로젝트 개요

- 1.1. 프로젝트 배경
- 1.2. 문제점 분석 및 새로운 요구사항
- 1.3. 프로젝트의 목표

2. 연구 내용

- 2.1. 주요 기능
- 2.2. 화면 흐름도

3. 데이터베이스 설계

- 3.1. 요구사항 분석
- 3.2. 개념적 데이터 모델링
 - (1) 개체, 속성 및 특징 추출
 - (2) 개체 간의 관계 유형 선별
 - (3) ER 다이어그램
- 3.3. 논리적 데이터 모델링
 - (1) 릴레이션 스키마로 변환
- 3.4. 물리적 데이터 모델링
 - 테이블 명세서 용어 사전
 - 물리 ERD

4. 결론

5. 부록

1. 프로젝트 개요

1.1. 프로젝트 배경

2016년 대비 2020년의 간편결제 서비스 이용건수와 이용 금액은 크게 늘어났다. 실제로 2020년, 간편 결제 이용 건수는 일 평균 기준으로 2016년보다 약 7배가 많은 1455만건에 달한다. 우리는 일상생활 속에서도 삼성페이, 네이버 페이, 카카오 페이 등 다양한 간편결제 시스템을 사용하고 있다. 간편 결제 시스템을 가진 어플리케이션 중 하나인 스타벅스 어플리케이션은 2014년에 출시되어 현재까지도 성장세를 지속하고 있다.



많은 고객이 다른 결제 수단이 있음에도 스타벅스 어플리케이션을 사용하는 이유는 '간편'하기 때문이다. 실제로는 많은 데이터를 담고 있음에도 사용자는 어플리케이션을 편리하게 이용한다. 많은 데이터를 편리하게 이용할 수 있는 그 이면에는 데이터베이스 설계가 있다.

따라서 본 프로젝트는 스타벅스 어플리케이션의 사용자 관점에서 데이터베이스를 설계한다.

1.2. 문제점 분석 및 새로운 요구사항

스타벅스 어플리케이션은 우수한 어플리케이션이지만 타 어플리케이션에 비해서 아쉬운 부분도 있다. 스타벅스 어플리케이션은 간편 결제 시에만 사용하고 그 이외의 시간은 사용하는 경우가 드물다. 사용자가 어플리케이션에 계속해서 흥미를 가지도록 하기 위해서는 주문하기 기능을 통한 주문 뿐만이 아니라 다양한 기능 또는 콘텐츠가 필요하다.

스타벅스 어플리케이션은 스타벅스매장 리뷰와 상품들의 별점 등 리뷰평가 서비스가 없다. 배달의 민족이나 요기요, 다른 카페 어플서비스를 확인해보면 리뷰에 따라 주문량이 달라지고 처음 먹어보는 상품이라도 별점을 참고해 주문하면 믿음이 생기고 다양하게 상품을 경험해 볼 수 있다. 스타벅스 어플리케이션에 리뷰를 작성하고 볼 수 있도록 하면 사용자들은 흥미를 느낄 것이다.

현재 스타벅스는 많은 이벤트들을 진행 중이다. 대표적으로 크리스마스를 맞이해서 프리퀀시 이벤트를 준비중인데 머그컵이나 다이어리 컴퓨터등등 스타벅스 로고가 들어간 상품들을 디자인해서 미션음료 3잔 포함 17잔을 마시면 증정하는 이벤트를 진행 중이다. 하지만 스타벅스에서는 고객의 기념일에 따른 한정 프로모션은 제공하지 않고 있다. 그렇기 때문에 고객의 생일

에 맞춘 기념일 한정 상품 구매 이벤트가 있다면 사용자들은 기념일마다 본 서비스를 사용하게 될 것이다.

1.3. 프로젝트의 목표

모바일 '스타벅스 애플리케이션을 기반으로 커피 주문 어플리케이션 데이터베이스 설계 및 기획 하는 것이 목표이다. 스타벅스 모바일 애플리케이션은 스타벅스 리워드, 스타벅스 카드, e-Gift 카드, 사이렌 오더 메뉴 주문 및 e-Gift Item 선물, Gift, Coupon, e-Frequency, 홀케이크 선물/예약 등의 서비스를 편리하고 스마트하게 사용할 수 있는 스타벅스 커피 코리아의 공식 애플리케이션이다. 회원은 물론, 회원이 아니더라도 애플리케이션을 이용하는 누구나 사이렌 오더를 통해 음료, 푸드 등 다양한 메뉴를 비대면으로 주문할 수 있으며, 스타벅스 카드를 등록하고 충전하여 바코드를 통해 결제할 수 있다. 팀원들 기준으로 불필요할 것 같은 기능들은 제외하였고 기존 애플리케이션에 있던 기본적인 기능과는 별개로 이벤트 기능과 리뷰 기능을 직접 추가하여 본래의 애플리케이션과 차이를 두었다. 기념일 이벤트 기능은 스타벅스 어플리케이션을 사용하는 고객들의 생일 데이터를 이용하여 생일을 맞이한 애플리케이션 사용자들을 대상으로 이벤트가 진행될 수 있도록 데이터베이스를 설계해보았다. 스타벅스 상품 리뷰 기능은 애플리케이션을 사용하는 사용자 모두가 구입했던 스타벅스 상품에 대한 리뷰를 작성할 수 있는 기능을 추가하였다. 정리하여, 기능이 추가된 스타벅스 어플을 데이터베이스 설계 및 기획을 하는 것을 목표로 프로젝트를 진행하였다.

2. 연구 내용

2.1. 주요 기능

기존 어플에 있던 기능들에 추가한 주요기능을 정리해보았다.

상품 정보 (상품-음료,푸드): 주문을 하기 위해 상품(음료,푸드)를 선택하면 다양한 상품 정보들을 확인할 수 있습니다. 상품명, 상품의 가격, 상품마다 다른 영양정보(칼로리, 탄수화물, 당류, 나트륨, 단백질, 지방, 콜레스테롤, 트랜스지방, 카페인, 포화지방), 상품의 성분에 따라 알레르기 유발요인(대두, 우유, 난류, 밀)이 있는 상품인지 알 수 있다.

상품 정보 (상품-기타): 주문을 하기 위해 상품(기타)를 선택하면 상품 정보와 상품의 원산지, 가격 등을 확인할 수 있다.

회원가입 기능: 성명, 닉네임, 성별, 생년월일, 나이, 전화번호, 주소, 알레르기 유발 요인, 결제정보를 입력하여 스타벅스 애플리케이션 회원가입을 할 수 있다.

옵션 선택 기능: 상품을 선택한 후 개인 취향에 알맞는 옵션을 선택할 수 있습니다. 옵션엔 사이즈, 컵 종류, 커피 종류, 시럽 여부, 음료 온도 및 얼음 양 등의 보기가 있

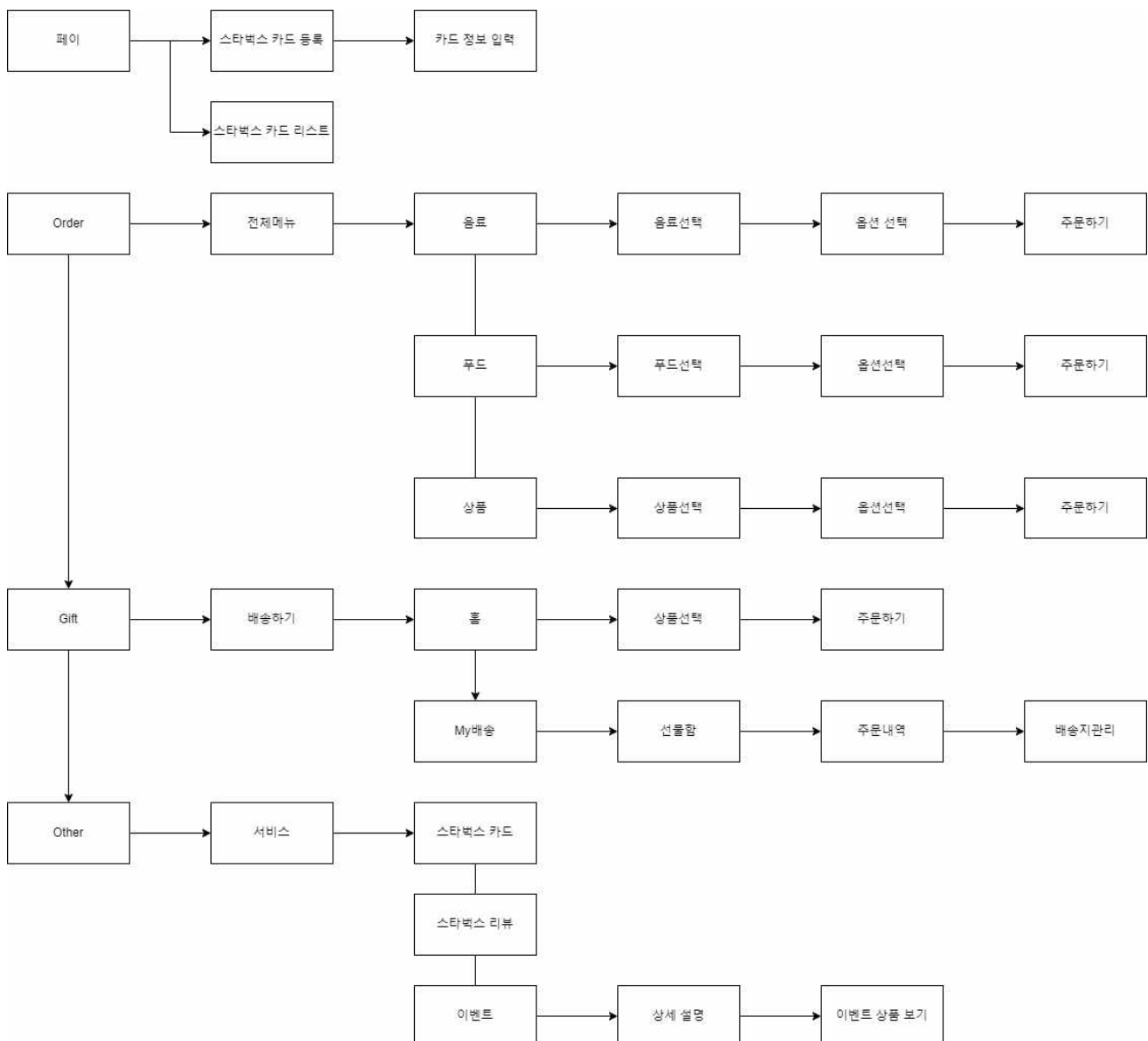
다.

결제 기능: 결제할시 스타벅스 카드나 신용카드로 결제가 가능하며 쿠폰, 통신사 제휴할인, 현금영수증을 선택하면 그에 맞는 할인이 가능하다.

이벤트 기능: 스타벅스 회원인 고객들을 대상으로 회원고객 데이터 중 생일 데이터를 이용하여 생일 이벤트가 진행된다. 생일은 맞이한 고객들이 이벤트 상품을 받아볼 수 있다. 이벤트 상품으로는 스타벅스 상품 이 지급되며 이벤트 상품의 정보(상품명, 원산지, 가격)를 알 수 있다.

리뷰 기능: 스타벅스 상품에 대하여 리뷰를 작성/확인 이 가능하다. 고객이 이용했던 상품을 평가하여 리뷰 작성이 가능합니다. 작성하지 않은 고객들도 타인의 리뷰를 확인할 수 있다.

2.2. 화면 흐름도



3. 데이터베이스 설계

3.1. 요구사항 명세서

- ① 상품(음료, 푸드)을 선택하면 상품 정보와 가격, 영양정보(음료,푸드), 알레르기 유발요인(음료, 푸드)를 확인 할 수있다.
- ② 상품(기타)을 선택하면 상품 정보와 원산지, 가격등을 확인할 수 있다.
- ③ 상품(음료,푸드)의 영양정보(칼로리, 탄수화물, 당류, 나트륨, 단백질, 지방, 콜레스테롤, 트랜스지방, 카페인, 포화지방)의 정보와 알레르기 유발 요인을 확인 할 수 있다.
- ④ 고객 회원가입시 성명, 닉네임 ,성별 ,생년월일, 나이, 전화번호, 주소, 알레르기 유발 요인, 결제정보를 입력해야한다.
- ⑤ 상품을 선택한 후에는 옵션을 선택할 수 있다.
- ⑥ 옵션은 사이즈, 컵 종류, 커피 종류, 시럽 여부, 음료 온도 및 얼음 양을 선택할 수 있다.
- ⑦ 결제할 때에는 결제 수단(스타벅스 카드, 신용카드) 쿠폰, 통신사 제휴할인, 현금영수증을 입력(선택)해야한다.
- ⑧ 고객은 상품에 리뷰를 작성할 수 있다.
- ⑨ 고객의 생일 전후 한 달 이내에 기념일 이벤트페이지에 접속할 수 있다.
- ⑩ 특정 이벤트에는 이벤트 한정 이벤트 상품 리스트가 있다.

3.2. 개념적 데이터 모델링

(1) 개체, 속성 및 특징 추출

① 상품(음료, 푸드)을 선택하면 상품 정보와 가격, 영양정보(음료,푸드), 알레르기 유발요인을 확인 할 수있다.

상품(음료, 푸드)은 개체로 분류하고 상품 정보, 가격, 영양정보(음료, 푸드), 알레르기 유발요인(음료, 푸드)는 '상품(음료, 푸드)'개체의 속성으로 분류한다. 영양정보와 알레르기 유발요인은 다중 값 속성을 가진다.

② 상품(기타)을 선택하면 상품 정보와 원산지, 가격등을 확인할 수 있다.

'기타 상품'은 개체로 분류하고 상품 정보, 원산지, 가공 방식, 가격, 프로필을 '상품'의 속성으로 분류한다.

③ 상품(음료,푸드)의 영양정보(칼로리, 탄수화물, 당류, 나트륨, 단백질, 지방, 콜레스테롤, 트랜스지방, 카페인, 포화지방)의 정보와 알레르기 유발 요인(대두, 우유, 난류, 밀)을 확인 할 수 있다.

영양정보(칼로리, 당류, 나트륨)는 상품(음료, 푸드)의 약한 개체로 분류되고, 칼로리, 탄수화물, 당류, 나트륨, 단백질, 지방, 콜레스테롤, 트랜스지방, 카페인, 포화지방 속성을 가진다. 알레르기 유발 요인은 상품(음료, 푸드)의 약한 개체로 분류되고, 대두, 우유, 난류, 밀을 속성으로 가진다. 고객의 알레르기 유발 요인과 '알레르기 유발 요인'개체를 참고하여 상품 섭취가 괜찮은지에 대한 여부를 유도 속성으로 확인할 수 있다.

④ 고객회원가입시 성명, 닉네임, 성별, 생년월일, 나이, 전화번호, 주소, 알레르기 유발 요인, 결제정보를 입력해야한다.

'고객'은 개체로 분류되고, 성명, 닉네임, 성별, 생년월일, 나이, 전화번호, 주소, 알레르기 유발 요인을 속성으로 가진다. '생년월일' 속성을 참고하여 '나이'속성을 유도 속성으로 가진다. 알레르기 유발 요인은 여러가지 유발 요인에 대한 값을 입력해야 하기 때문에 복합 속성이다. '결제 정보'는 한 고객 당 여러 개를 가질 수 있기 때문에 다중값 속성을 가지고, 결제 정보에는 카드별칭, 카드 번호, 유효기간, cvc번호, 비밀번호 앞 두 자리 속성을 가지기 때문에 복합 속성을 가진다.

⑤ 상품을 선택한 후에는 옵션을 선택하여 주문할 수 있다.

'주문'은 개체로 분류되고 '옵션'은 주문 개체에 종속적이기 때문에 '주문'개체의 약한 개체가 된다.

⑥ 옵션은 사이즈, 컵 종류, 커피 종류, 시럽 여부, 음료 온도 및 얼음 양을 선택할 수 있다.

'옵션'은 개체로 분류되고, 사이즈, 컵 종류, 커피 종류, 시럽 여부, 음료 온도 및 얼음 양을 속성으로 가진다.

⑦ 결제할 때에는 결제 수단(스타벅스 카드, 신용카드), 쿠폰, 통신사 제휴할인, 현금영수증을 입력(선택)해야한다.

고객 개체는 '고객 결제정보'개체를 약한 개체로 가지고 있으며, 고객 결제 정보에는 스타벅스 카드, 신용카드, 쿠폰, 통신사 제휴 할인, 현금영수증 속성을 가지고 있다. 스타벅스 카드, 신용카드 속성은 다중값 속성과 복합 속성을 가지며, 쿠폰 속성은 다중값 속성을 가지고 있다.

⑧ 고객은 상품에 리뷰를 작성할 수 있다.

리뷰 개체를 생성한다.

⑨ 고객의 생일 전후 한 달 이내에 기념일 이벤트페이지에 접속할 수 있다.

이벤트 페이지 개체를 생성한다.

⑩ 특정 이벤트에는 이벤트 한정 이벤트 상품 리스트가 있다.

이벤트 개체와 연결되는 이벤트 상품 개체를 생성한다.

상품(음료, 푸드)	상품ID, 상품명, 상품 정보, 가격, 영양정보, 알레르기 유발 요인	약한 개체: 영양정보, 알레르기 유발요인
영양정보	칼로리, 탄수화물, 당류, 나트륨, 단백질, 지방, 콜레스테롤, 트랜스지방, 카페인, 포화지방	
알레르기 유발요인	대두, 우유, 난류, 밀, 고객 알레르기 여부 확인	유도 속성: 고객 알레르기 여부 확인
상품(기타)	상품ID, 상품 정보, 원산지, 가격	
고객	고객ID, 성명, 닉네임, 성별, 생년월일, 나이, 전화번호, 주소, 알레르기 유발 요인, 결제정보	약한개체: 결제 정보 유도 속성: 나이
주문	고객ID, 상품ID	
옵션	사이즈, 컵 종류, 커피 종류, 시럽 여부, 음료 온도 및 얼음 양	
고객 결제 정보	스타벅스 카드, 신용카드, 쿠폰, 통신사 제휴 할인, 현금영수증	다중값 속성: 스타벅스 카드, 신용카드, 쿠폰 복합 속성: 신용카드, 스타벅스 카드
리뷰	리뷰ID, 리뷰 내용, 별점, 리뷰 날짜	
이벤트 상품	이벤트 상품명, 이벤트 상품 정보, 이벤트 상품 원산지, 이벤트 상품 가격	
기념일 이벤트	이벤트 ID, 이벤트 내용, 이벤트 이름	

표1. 개체, 속성 및 특징 추출

(2) 개체 간의 관계 유형 선별

① 고객 - 고객 결제 정보

결제는 고객 개체와 고객 결제 정보 개체가 맺는 관계를 설명한다. 따라서 고객 개체와 고객 결제 정보 개체가 맺고 있는 결제 관계를 추출할 수 있다. 고객 한명이 하나의 고객 결제 정보를 선택할 수 있고, 하나의 고객 결제 정보는 고객 한 명에 의해서 결제 되므로 고객 개체와 고객 결제 정보가 맺는 결제 관계는 일대일(1:1)이 된다. 고객은 상품을 결제하는 조건이므로 고객 개체는 결제 관계에 필수적으로 참여한다고 볼 수 있다. 고객에 의해서 하나의 상품을 결제하므로 상품은 결제 관계에 필수적으로 참여한다고 볼 수 있다.

② 상품(음료, 푸드) - 영양정보

상품 하나에 영양정보 리스트(칼로리, 탄수화물, 당류, 나트륨 등등) 을 가질 수 있고, 영양정보 리스트는 상품 하나를 대상으로 해야한다. 그러므로 영양정보와 상품은 일대일(1:1) 관계가 된다. 상품이 삭제되면 영양정보도 삭제되므로 종속적인 특징을 가진다. 상품 개체가 강한개체이고,영양정보 개체가 약한 개체(종속 개체)이다.

③ 상품(음료, 푸드) - 알레르기 유발요인

상품 하나가 알레르기 유발요인 리스트(대두, 우유, 난류, 밀) 하나를 가질 수 있고, 알레르기 유발 요인 리스트는 하나의 상품을 대상으로 한다. 해당 관계는 일대일(1:1)관계가 된다. 상품이 삭제되면 알레르기 유발 요인도 삭제되므로 종속적인 특징을 가진다. 상품 개체가 강한개체

이고, 알레르기 유발 요인 개체가 약한 개체(종속 개체)이다.

④ 주문 - 상품(음료,푸드)

'주문' 한 번에 여러 개의 상품(음료,푸드)를 주문할 수 있으며, '상품(음료,푸드)'하나는 하나의 주문을 할 수 있다. 따라서, '주문'개체와 '상품(음료,푸드)'개체의 관계는 일대다(1:n)관계가 된다. '상품(음료,푸드)'개체가 존재하여야 '주문'개체가 존재할 수 있기 때문에 상품(음료,푸드) 개체는 강한 개체(오너 개체)이고, 주문 개체는 약한 개체(종속 개체)이다.

⑤ 상품(음료,푸드) - 옵션

상품(음료,푸드)가 하나의 옵션을 선택할 수 있고, 하나의 상품(음료,푸드)은 옵션에 의해서 주문 된다. 따라서, '상품(음료,푸드)'개체와 '옵션'개체가 맺는 관계는 일대일(1:1)관계가 된다.

⑥ 주문 - 상품(기타)

'주문' 한 번에 여러 개의 상품(기타)를 주문할 수 있으며, '상품(기타)'하나는 하나의 주문을 할 수 있다. 따라서, '주문'개체와 '상품(기타)'개체의 관계는 일대다(1:n)관계가 된다. '상품(기타)'개체가 존재하여야 '주문'개체가 존재할 수 있기 때문에 상품(기타) 개체는 강한 개체(오너 개체)이고, 주문 개체는 약한 개체(종속 개체)이다.

⑦ 주문 - 고객

주문 한 번에 고객 한 명의 정보만이 필요하고, 고객은 여러 번 주문을 할 수 있기 때문에 주문과 고객과의 관계는 일대다(1:n)관계이다.

⑧ 고객 - 리뷰

리뷰 한 번에 고객 한 명의 정보만이 필요하고, 고객은 여러 번 리뷰를 작성할 수 있기 때문에 리뷰와 고객과의 관계는 일대다(1:n)관계이다.

⑨ 상품(음료, 푸드) - 리뷰

상품 하나에 리뷰를 하나만 달 수 있고, 상품에는 여러 개의 리뷰가 달릴 수 있기 때문에 상품과 리뷰와의 관계는 일대다(1:n)관계이다.

⑩ 이벤트 상품 - 주문

'주문' 한 번에 여러 개의 이벤트 상품을 주문할 수 있으며, '이벤트 상품'하나는 하나의 주문을 할 수 있다. 따라서, '주문'개체와 '이벤트 상품'개체의 관계는 일대다(1:n)관계가 된다. '이벤트 상품'개체가 존재하여야 '주문'개체가 존재할 수 있기 때문에 이벤트 상품 개체는 강한 개체(오너 개체)이고, 주문 개체는 약한 개체(종속 개체)이다.

⑪ 이벤트 - 이벤트 상품

이벤트 상품 하나에 이벤트 한 개의 정보만이 필요하고, 이벤트는 여러 개의 이벤트 상품을 가질 수 있기 때문에 이벤트와 이벤트 상품과의 관계는 일대다(1:n)관계이다.

	관계에 참여하는 개체	참여도	관계 유형	속성	오너 개체, 종속 개체 분류
1	고객	선택	일대일(1:1)	종속성	오너 개체-고객 종속 개체-고객 결제정보
	고객 결제정보	필수			
2	상품(음료, 푸드)	필수	일대일(1:1)	종속성	오너 개체-상품 종속 개체-영양정보
	영양정보	필수			
3	상품(음료, 푸드)	필수	일대일(1:1)	종속성	오너 개체-상품 종속 개체-알레르기 유발 요인
	알레르기 유발요인	필수			
4	주문	필수	일대다(1:n)	종속성	오너 개체-상품 종속 개체-주문
	상품(음료, 푸드)	선택			
5	상품(음료, 푸드)	선택	일대일(1:1)	종속성	오너 개체-상품 종속 개체-옵션
	옵션	필수			
6	주문	필수	일대다(1:n)	종속성	오너 개체-상품 종속 개체-주문
	상품(기타)	선택			
7	고객	선택	일대다(1:n)	.	.
	주문	필수			
8	고객	선택	일대다(1:n)	종속성	오너개체-고객 종속개체-리뷰
	리뷰	필수			
9	상품(음료, 푸드)	선택	일대다(1:n)	종속성	오너개체-상품(음료, 푸드) 종속개체-리뷰
	리뷰	필수			
10	이벤트 상품	필수	일대다(1:n)	종속성	오너 개체-이벤트 상품 종속 개체-주문
	주문	선택			
11	이벤트	선택	일대다(1:n)	종속성	오너 개체-이벤트 종속 개체-이벤트 상품
	이벤트 상품	필수			

표 2. 개체 관계 유형 선별

(3) ER 다이어그램

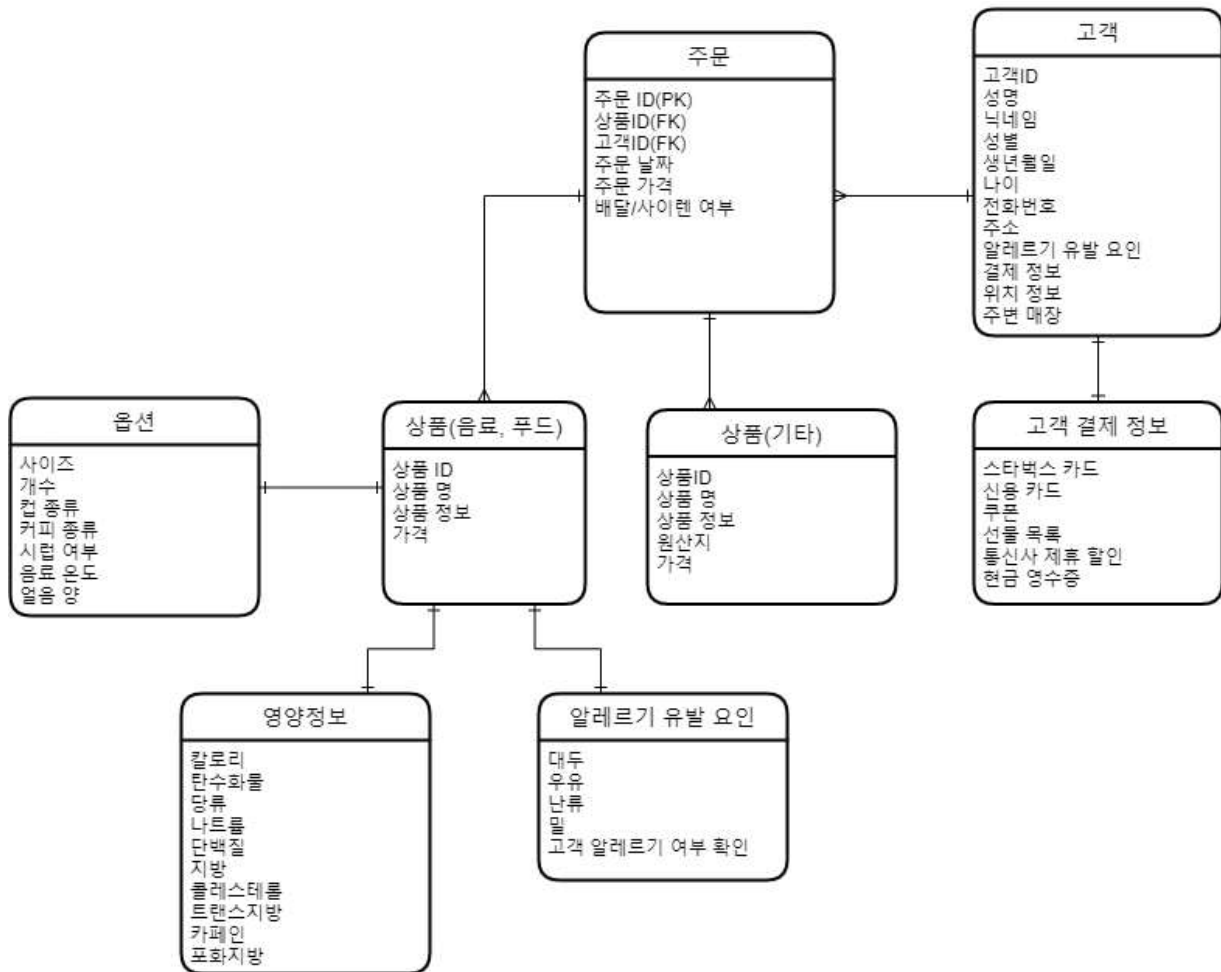


그림 3 개념적 설계 ER 다이어그램

3.3. 논리적 데이터 모델링

(1) 릴레이션 스키마로 변환

릴레이션 스키마 변환 규칙에 따라서 ER다이어그램을 릴레이션 스키마로 변환한다. 릴레이션 스키마 변환 규칙은 아래와 같다.

- 규칙 1 : 모든 개체는 릴레이션으로 변환한다.
- 규칙 2 : 다대다 관계는 릴레이션으로 변환한다.
- 규칙 3 : 일대다 관계는 외래키로 표현한다.
 - ① 일반적인 일대다 관계는 외래키로 표현한다.
 - ② 약한 개체가 참여하는 일대다 관계는 외래키를 포함해서 기본키로 지정한다.
- 규칙 4 : 일대일 관계는 외래키로 표현한다.
 - ① 일반적인 일대일 관계는 외래키를 서로 주고 받는다.
 - ② 일대일 관계에 필수적으로 참여하는 개체의 릴레이션만 외래키를 받는다.
- 규칙 5 : 다중 값 속성은 릴레이션으로 반환한다.

표 3. 릴레이션 스키마 변환 규칙

1) 모든 개체는 릴레이션으로 변환한다.

상품(음료, 푸드)	영양정보	알레르기 유발요인	상품(기타)
상품(음료, 푸드) ID	영양정보 ID	알레르기 ID	상품(기타)ID
상품명	칼로리	대두	상품 정보
상품정보	탄수화물	우유	원산지
가격	당류	난류	가격
영양정보	나트륨	밀	
알레르기 유발요인	단백질	고객 알레르기 여부 확인	
	지방		
	콜레스테롤		
	트랜스지방		
	카페인		
	포화지방		

고객	주문	옵션	고객결제정보
고객 ID	주문 ID	옵션 ID	결제 정보 ID
성명	고객 ID	사이즈	스타벅스 카드
닉네임	상품 ID	컵 종류	신용카드
성별		커피 종류	쿠폰
생년월일		시럽 여부	통신사 제휴 할인
나이		음료 온도 및 얼음 양	현금영수증
전화번호			
주소			
알레르기 유발 요인			
결제정보			

리뷰	이벤트 상품	기념일 이벤트
리뷰ID	이벤트 상품 ID	이벤트 ID
리뷰내용	이벤트 상품명	이벤트 내용
별점	이벤트 상품 원산지	이벤트 이름
리뷰날짜	이벤트 상품 가격	

2) 다대다 관계는 릴레이션으로 변환한다.

다대다 관계는 개념적 설계 과정에서 고려해서 일대다 관계로 변환하였다. 해당 데이터베이스 내에는 다대다 관계가 없으므로 다음 단계를 진행한다.

3) 일대다 관계는 외래키로 표현한다.

해당 프로젝트의 데이터베이스에서 일대다 관계는 아래와 같다.

- 주문 - 상품(음료, 푸드)
- 주문 - 상품(기타)
- 고객 - 주문

① 일반적인 일대다 관계는 외래키로 표현한다.

→ '고객'과 '주문'의 관계는 종속성을 가지지 않는 일반적인 일대다 관계이다. 따라서 주문 개체는 고객 개체의 기본키를 외래키로 가진다.

고객	주문
고객 ID(PK)	주문 ID(PK)
성명	고객 ID(FK)
닉네임	상품 ID
성별	
생년월일	
나이	
전화번호	
주소	
알레르기 유발 요인	
결제정보	

② 약한 개체가 참여하는 일대다 관계는 외래키를 포함해서 기본키로 지정한다.

→ 주문은 상품(음료, 푸드)이 있어야만 존재할 수 있는 일대다 관계이므로, 주문은 상품(음료, 푸드)의 약한 개체이다.

주문	상품(음료, 푸드)
주문 ID(PK)	상품(음료, 푸드) ID(PK)
상품(음료, 푸드) ID(PK/FK)	상품명
고객 ID(FK)	상품정보
	가격
	영양정보
	알레르기 유발요인

→ 주문은 상품(기타)이 있어야만 존재할 수 있는 일대다 관계이므로, 주문은 상품(기타)의 약한 개체이다.

주문	상품(기타)
주문 ID(PK)	상품(기타)ID(PK)
상품(음료, 푸드) ID(PK/FK)	상품 정보
상품(기타) ID(PK/FK)	원산지
고객 ID(FK)	가격

4) 일대일 관계는 외래키로 표현한다.

해당 프로젝트의 데이터베이스에서 일대일 관계는 아래와 같다.

→ 고객 - 고객 결제정보

→ 상품(음료, 푸드) - 영양정보

→ 상품(음료, 푸드) - 알레르기 유발요인

→ 상품(음료, 푸드) - 옵션

① 일반적인 일대일 관계는 외래키를 서로 주고받는다.

일반적인 일대일 관계란 모든 개체가 필수적으로 관계에 참여하는 것을 의미한다.

→ 상품(음료, 푸드) 개체와 영양정보 개체는 각각 필수적으로 관계에 참여하고 있다.

상품(음료, 푸드)	영양정보
상품(음료, 푸드) ID(PK)	영양정보 ID(PK)
영양정보 ID(FK)	상품(음료, 푸드) ID(FK)
상품명	칼로리
상품정보	탄수화물
가격	당류
	나트륨
	단백질
	지방
	콜레스테롤
	트랜스지방
	카페인
	포화지방

→ 상품(음료, 푸드) 개체와 알레르기 유발요인 개체는 각각 필수적으로 관계에 참여하고 있다.

상품(음료, 푸드)	알레르기 유발요인
상품(음료, 푸드) ID(PK)	알레르기 ID(PK)
알레르기 ID(FK)	상품(음료, 푸드) ID(FK)
영양정보 ID(FK)	대두
상품명	우유
상품정보	난류
가격	밀
	고객 알레르기 여부 확인

② 일대일 관계에 필수적으로 참여하는 개체의 릴레이션만 외래키를 받는다.

→ 고객 개체와 고객 결제정보 개체간의 관계에서 고객은 선택적으로 관계에 참여하고, 고객 결제정보는 필수적으로 관계에 참여한다.

고객 - 선택 참여	고객결제정보 - 필수 참여
고객 ID(PK)	결제 정보 ID(PK)
성명	고객 ID(FK)
닉네임	스타벅스 카드
성별	신용카드
생년월일	쿠폰
나이	통신사 제휴 할인
전화번호	현금영수증
주소	
알레르기 유발 요인	
결제정보	

→ 상품 개체와 옵션 개체간의 관계에서 상품은 선택적으로 관계에 참여하고, 옵션은 필수적으로 관계에 참여한다.

상품(음료, 푸드) - 선택 참여	옵션 - 필수 참여
상품(음료, 푸드) ID(PK)	옵션 ID(PK)
알레르기 ID(FK)	상품(음료, 푸드) ID(FK)
영양정보 ID(FK)	사이즈
상품명	컵 종류
상품정보	커피 종류
가격	시럽 여부
	음료 온도 및 얼음 양

5) 다중 값 속성은 릴레이션으로 반환한다.

고객 결제정보 개체의 스타벅스 카드, 신용카드, 쿠폰은 다중 값 속성을 가지므로 3개의 릴레이션을 생성한다.

고객결제정보
결제 정보 ID(PK)
고객 ID(FK)
스타벅스 카드
신용카드
쿠폰
통신사 제휴 할인
현금영수증

→

스타벅스 카드	신용카드	쿠폰
스타벅스카드 ID(PK)	신용카드 ID(PK)	쿠폰 ID(PK)
스타벅스 카드 이름	신용카드 이름	쿠폰 할인 금액
스타벅스 카드 번호	신용카드 번호	쿠폰 번호
		유효기간

기존 다중값 속성을 가지고 있던 개체의 기본키를 외래키로 가져와 새로운 릴레이션 스키마

의 기본키로 한다.

고객결제정보	스타벅스 카드	신용카드	쿠폰
결제 정보 ID(PK)	스타벅스카드 ID(PK)	신용카드 ID(PK)	쿠폰 ID(PK)
고객 ID(FK)	결제 정보 ID(FK/PK)	결제 정보 ID(FK/PK)	결제 정보 ID(FK/PK)
통신사 제휴 할인	스타벅스 카드 이름	신용카드 이름	쿠폰 할인 금액
현금영수증	스타벅스 카드 번호	신용카드 번호	쿠폰 번호
			유효기간

본 프로젝트의 데이터베이스는 최종적으로 14개의 릴레이션 스키마(개체)로 구성되었다.

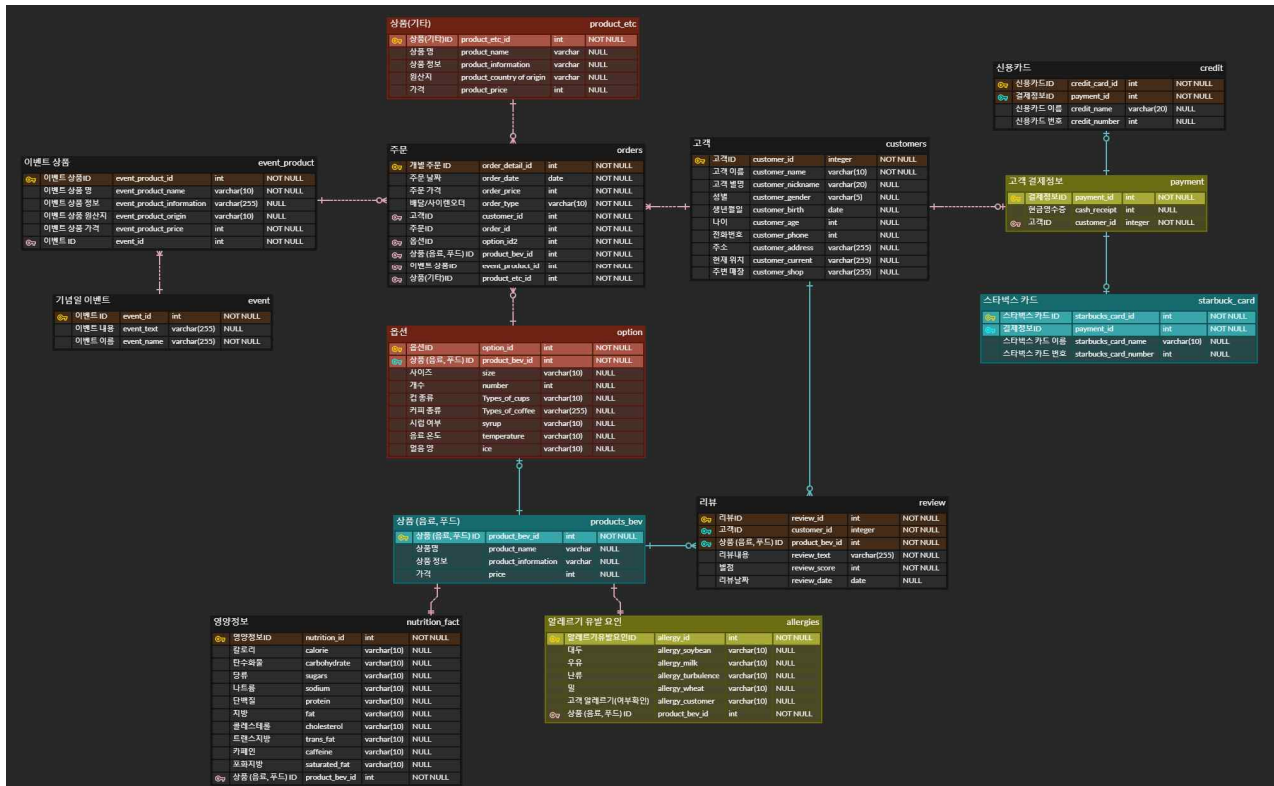
상품(음료, 푸드)	영양정보	알레르기 유발요인	상품(기타)
상품(음료, 푸드) ID(PK)	영양정보 ID(PK)	알레르기 ID(PK)	상품(기타)ID(PK)
알레르기 ID(FK)	상품(음료, 푸드) ID(FK)	상품(음료, 푸드) ID(FK)	상품 정보
영양정보 ID(FK)	칼로리	대두	원산지
상품명	탄수화물	우유	가격
상품정보	당류	난류	
가격	나트륨	밀	
	단백질	고객 알레르기 여부 확인	
	지방		
	콜레스테롤		
	트랜스지방		
	카페인		
	포화지방		

고객	주문	옵션
고객 ID(PK)	주문 ID(PK)	옵션 ID(PK)
성명	상품(음료, 푸드) ID(PK/FK)	상품(음료, 푸드) ID(FK)
닉네임	상품(기타) ID(PK/FK)	사이즈
성별	고객 ID(FK)	컵 종류
생년월일		커피 종류
나이		시럽 여부
전화번호		음료 온도 및 얼음 양
주소		
알레르기 유발 요인		
결제정보		

고객결제정보	스타벅스 카드	신용카드	쿠폰
결제 정보 ID(PK)	스타벅스카드 ID(PK)	신용카드 ID(PK)	쿠폰 ID(PK)
고객 ID(FK)	결제 정보 ID(FK/PK)	결제 정보 ID(FK/PK)	결제 정보 ID(FK/PK)
통신사 제휴 할인	스타벅스 카드 이름	신용카드 이름	쿠폰 할인 금액
현금영수증	스타벅스 카드 번호	신용카드 번호	쿠폰 번호
			유효기간

리뷰	이벤트 상품	기념일 이벤트
리뷰ID(PK)	이벤트 상품 ID(PK)	이벤트 ID(PK)
고객ID(PK/FK)	이벤트 ID(FK)	이벤트 내용
상품(음료, 푸드)ID(PK/FK)	이벤트 상품명	이벤트 이름
리뷰내용	이벤트 상품 원산지	
별점	이벤트 상품 가격	
리뷰날짜		

본 프로젝트의 데이터베이스는 최종적으로 13개의 릴레이션으로 구성되었다.



3.4. 물리적 데이터 모델링

(1) 테이블 명세서 용어 사전

고객	customer			
논리명	물리명	약어	설명	타입
고객아이디	customer_id	cust_id	고객아이디	int
고객이름	customer_name	cust_name	고객이름	varchar
고객별명	customer_nickname	cust_nickname	고객별명	varchar
성별	customer_gender	cust_gender	고객 성별	varchar
생년월일	customer_birth	cust_birth	고객 생년월일	date
나이	customer_age	cust_age	고객 나이	int
전화번호	customer_phone	cust_num	고객 전화번호	int
주소	customer_address	cust_addr	고객 주소	varchar
현재위치	customer_current	cust_current	고객 현재위치	varchar
주변매장	customer_shop	cust_shop	고객 주변매장	varchar

주문	orders			
논리명	물리명	약어	설명	타입
개별 주문 ID	order_detail_id	order_detail_id	개별 주문 ID	int
주문ID	order_id	order_id	주문ID	int
상품(기타)ID	product_etc_id	product_etc_id	상품(기타)아이디	int
이벤트 상품ID	event_product_id	event_product_id	이벤트 상품ID	int
옵션ID	option_id	option_id	옵션ID	int
주문 날짜	order_date	order_date	주문 날짜	date
주문 가격	order_price	order_price	주문 가격	int
배달/사이렌오더	order_type	order_type	배달/사이렌오더	varchar
고객ID	customer_id	cust_id	고객ID	int

영양정보	nutrition fact			
논리명	물리명	약어	설명	타입
영양정보ID	nutrition_id	nutrition_id	영양정보아이디	varchar
칼로리	calorie	calorie	칼로리	varchar
탄수화물	carbohydrate	carbohydrate	탄수화물	varchar
당류	sugars	sugars	당류	varchar
나트륨	sodium	sodium	나트륨	varchar
단백질	protein	protein	단백질	varchar
지방	fat	fat	지방	varchar
콜레스테롤	cholesterol	cholesterol	콜레스테롤	varchar
트랜스지방	trans_fat	trans_fat	트랜스지방	varchar
카페인	caffeine	caffeine	카페인	varchar
포화지방	saturated_fat	saturated_fat	포화지방	varchar
상품(음료,푸드)ID	product_bev_id	product_bev_id	상품(음료,푸드) 아이디	int

옵션	option			
논리명	물리명	약어	설명	타입
옵션ID	option_id	option_id	옵션 아이디	int
상품 (음료, 푸드) ID	product_bev_id	product_bev_id	상품 아이디	int
사이즈	size	size	사이즈	varchar
개수	number	num	개수	int
컵 종류	Types of cups	Types of cups	컵 종류	varchar
커피 종류	Types of coffee	Types of coffee	커피 종류	varchar
시럽 여부	syrup	syrup	시럽 유/무	varchar
음료 온도	temperature	temp	온도	varchar
얼음 양	ice	ice	얼음 양	varchar

알레르기 유발 요인	allergies			
논리명	물리명	약어	설명	타입
알레르기유발요인ID	allergy_id	allergy_id	알레르기유발요인ID	int
대두	allergy_soybean	allergy_soybean	대두	varchar
우유	allergy_milk	allergy_milk	우유	varchar
난류	a l l e r g y _ turbulence	a l l e r g y _ turbulence	난류	varchar
밀	allergy_wheat	allergy_wheat	밀	varchar
고객 알레르기(여부확인)	allergy_customer	allergy_cust	고객알레르기(여부확인)	varchar
상품 (음료, 푸드) ID	product_bev_id	product_bev_id	상품 (음료, 푸드) ID	int

상품(기타)	product_etc			
논리명	물리명	약어	설명	타입
상품(기타)ID	product_etc_id	product_etc_id	상 품 (기 타)ID	int
상품 명	product_name	product_name	상품 명	varchar
상품 정보	product_information	product_information	상품 정보	varchar
원산지	product_country of origin	product_country of origin	원산지	varchar
가격	product_price	product_price	가격	int

신용카드	credit			
논리명	물리명	약어	설명	타입
신용카드ID	credit_card_id	credit_card_id	신용카드ID	int
결제정보ID	payment_id	payment_id	결제정보ID	int
신용카드 이름	credit_name	credit_name	신용카드 이름	varchar
신용카드 번호	credit_number	credit_num	신용카드 번호	int

스타벅스 카드	starbucks card			
논리명	물리명	약어	설명	타입
스타벅스카드 ID	starbucks card_id	s t a r b u c k s card_id	스타벅스 카드 ID	int
결제정보ID	payment_id	payment_id	결제정보ID	int
스타벅스카드 이름	starbucks card_name	s t a r b u c k s card_name	스타벅스 카드 이름	varchar
스타벅스카드 번호	starbucks card_number	s t a r b u c k s card_num	스타벅스 카드 번호	int

고객 결제정보	payment			
논리명	물리명	약어	설명	타입
결제정보ID	payment_id	payment_id	결제정보ID	int
현금영수증	cash_receipt	cash_receipt	현금영수증	int
고객ID	customer_id	cust_id	고객ID	int

상품 (음료, 푸드)	products_bev			
논리명	물리명	약어	설명	타입
상품 (음료, 푸드) ID	product_bev_id	product_bev_id	상품(음료, 푸드) ID	int
상품명	product_name	product_name	상품명	int
상품 정보	product_information	product_inform ation	상품 정보	varchar
가격	price	price	가격	int

(2) 물리적 설계

①상품(기타)

-입력 데이터

product_etc_id	product_name	product_information
3	리유지할 것 음료	음료식용 밀키와 함께 제공되며 구매받지 않고 사용할 수 있어 활용성이 높은 591ml 용량의 리유지할 것 음료입니다.
4	콜라 사이렌 콜드캔	콜라식용 밀키와 함께 제공되며 구매받지 않고 사용할 수 있어 활용성이 높은 710ml 용량의 콜라식용 콜드캔입니다.
5	그린 사이렌 보온병	보온/보냉 기능이 우수한 다양한 보온병입니다.
8	제너 원두	원대성 고유의 향기와 신비로운 맛으로 감동할 주스와 같은 느낌을 가진 커피
9	콜롬비아 원두	진과류의 풍미를 간직한 부드러운 균형 잡힌 커피
10	에스프레소 포스트	풍부하고 고화로운 느낌으로 스타벅스 에스프레소 음료의 기본이 되는 커피

product_country_of_origin	product_price
korea	40000
korea	19000
korea	22000
kenya	11000
colombia	12000
ethiopia	14000

-테이블 상세정보

Field	Type	Null	Key	Default	Extra
product_etc_id	int(11)	NO	PRI	NULL	
product_name	varchar(10)	YES		NULL	
product_information	varchar(255)	YES		NULL	
product_country_of_origin	varchar(10)	YES		NULL	
product_price	int(11)	YES		NULL	

필드에는 product_etc_id, product_name, product_information, product_countryof

origin,product_price이렇게 5개가 있고

각각 type에는 int(11),varchar(10),varchar(255),varchar(10)int(11)이다
product_etc_id는 primary key이다.

②상품(음료, 푸드)

- 입력 데이터

product_bev_id	product_name
1	카라멜 마키아도
2	카페 라떼
3	자바칩 프라푸치노
4	자몽에이드
5	아메리카노
6	카푸치노
7	바닐라빈라떼
8	플레인라떼
9	플랫화이트
10	얼그레이티

product_information	price
향긋한 베닐라 시럽과 부드러운 스팀 밀크 위에 풍성한 우유 거품을 얹고 밀크 토포의 에스프레소를 부은 후 토포의 크림으로 가늠된 드리핑을 올린 달콤한 커피 음료	5000
달콤하고 진한 에스프레소가 신선한 스팀 밀크를 만나 부드러운 커피 위에 우유 거품을 얹어 맛있게 즐길 수 있는 달콤한 커피 음료	7000
향긋한 베닐라 시럽과 부드러운 스팀 밀크 위에 풍성한 우유 거품을 얹고 밀크 토포의 에스프레소를 부은 후 토포의 크림으로 가늠된 드리핑을 올린 달콤한 커피 음료	5500
향긋한 베닐라 시럽과 부드러운 스팀 밀크 위에 풍성한 우유 거품을 얹고 밀크 토포의 에스프레소를 부은 후 토포의 크림으로 가늠된 드리핑을 올린 달콤한 커피 음료	5000
향긋한 베닐라 시럽과 부드러운 스팀 밀크 위에 풍성한 우유 거품을 얹고 밀크 토포의 에스프레소를 부은 후 토포의 크림으로 가늠된 드리핑을 올린 달콤한 커피 음료	6000
향긋한 베닐라 시럽과 부드러운 스팀 밀크 위에 풍성한 우유 거품을 얹고 밀크 토포의 에스프레소를 부은 후 토포의 크림으로 가늠된 드리핑을 올린 달콤한 커피 음료	5000
향긋한 베닐라 시럽과 부드러운 스팀 밀크 위에 풍성한 우유 거품을 얹고 밀크 토포의 에스프레소를 부은 후 토포의 크림으로 가늠된 드리핑을 올린 달콤한 커피 음료	6000
향긋한 베닐라 시럽과 부드러운 스팀 밀크 위에 풍성한 우유 거품을 얹고 밀크 토포의 에스프레소를 부은 후 토포의 크림으로 가늠된 드리핑을 올린 달콤한 커피 음료	7000
향긋한 베닐라 시럽과 부드러운 스팀 밀크 위에 풍성한 우유 거품을 얹고 밀크 토포의 에스프레소를 부은 후 토포의 크림으로 가늠된 드리핑을 올린 달콤한 커피 음료	6000
향긋한 베닐라 시럽과 부드러운 스팀 밀크 위에 풍성한 우유 거품을 얹고 밀크 토포의 에스프레소를 부은 후 토포의 크림으로 가늠된 드리핑을 올린 달콤한 커피 음료	5000
향긋한 베닐라 시럽과 부드러운 스팀 밀크 위에 풍성한 우유 거품을 얹고 밀크 토포의 에스프레소를 부은 후 토포의 크림으로 가늠된 드리핑을 올린 달콤한 커피 음료	4000

- 테이블 상세 정보

Field	Type	Null	Key	Default	Extra
product_bev_id	int(11)	NO	PRI	NULL	
product_name	varchar(10)	YES		NULL	
product_information	varchar(255)	YES		NULL	
price	int(11)	YES		NULL	

필드에는 product_bev_id,product_name,product_information,price 이렇게 4개가 있고

각각 type에는 int(11),varchar(10),varchar(255),int(11)이다.

product_bev_id는 primary key이다.

③ 옵션

입력 데이터

option_id	product_bev_id	size	number	Types_of_cups	Types_of_coffee	syrup	temperature	ice
1	2	tall	1	밀크	카페인	있음	hot	적게
2	3	grande	1	밀크	카페인	있음	ice	많이
3	4	tall	1	머그컵	카페인	없음	hot	적게
4	1	small	1	머그컵	카페인	없음	hot	많이
5	3	null	1	밀크	카페인	없음	ice	보통
6	2	small	1	머그컵	카페인	있음	ice	적게
7	5	tall	1	테이크아웃	카페인	없음	ice	보통
8	7	tall	2	테이크아웃	카페인	없음	ice	보통
9	8	tall	2	머그컵	카페인	없음	hot	보통
10	7	small	1	머그컵	카페인	없음	ice	적게

테이블 상세 정보

Field	Type	Null	Key	Default	Extra
option_id	int(11)	NO	PRI	NULL	
product_bev_id	int(11)	NO	PRI	NULL	
size	varchar(10)	YES		NULL	
number	int(11)	YES		NULL	
Types_of_cups	varchar(10)	YES		NULL	
Types_of_coffee	varchar(10)	YES		NULL	
syrup	varchar(10)	YES		NULL	
temperature	varchar(10)	YES		NULL	
ice	varchar(10)	YES		NULL	

field에는 option_id,product_bev_id,size,number,Types_of_cups,Types_of_coffee,,syrup, temperature,ice 이렇게 9개가 있고 각각 type에는 int(11),int(11),varchar(10),int(11), varchar(10),varchar(10),varchar(10),varchar(10),varchar(10)이다.
option_id는 primary key이고 product_bev_id는 primary key이다.

④ 고객

입력 데이터

customer_id	customer_name	customer_nickname	customer_gender	customer_birth	customer_age	customer_phone	customer_address	customer_current	customer_shop
1 김수영	gayesong		여성	2009-10-20	22	1011111111	서울구 역삼동로 1번 지	★강남구 ★건 2층 동명대로 472	서면로점
2 전세민	yejin		여성	2001-05-05	22	1022222222	서울구 역삼동로 2번 지	★강남구 ★건 2층 동명대로 473	서면로점
3 김민수	gemu		남자	1999-07-09	23	1033333333	서울구 역삼동로 3번 지	★강남구 ★건 2층 동명대로 474	서면동명대로점
4 김민수	taetae		남자	1998-05-13	24	1044444444	서울구 역삼동로 4번 지	★강남구 ★건 2층 동명대로 475	서면역삼로점
5 김민수	taeh		남자	1990-04-10	32	1055555555	서울구 역삼동로 5번 지	★강남구 ★건 2층 동명대로 476	서면로점
6 김민수	One		남자	2010-09-20	12	1066666666	서울구 역삼동로 6번 지	★강남구 ★건 2층 동명대로 477	서면로점
7 박민수	jjeong		여성	2005-01-12	17	1077777777	서울구 역삼동로 7번 지	★강남구 ★건 2층 동명대로 478	서면동명대로점
8 박민수	hyeon		남자	1999-11-25	23	1088888888	서울구 역삼동로 8번 지	★강남구 ★건 2층 동명대로 479	서면역삼로점
9 김민수	doyeon		남자	1990-12-25	32	1099999999	서울구 역삼동로 9번 지	★강남구 ★건 2층 동명대로 480	서면동명대로점
10 김민수	ubin		남자	1989-02-24	34	1000000000	서울구 역삼동로 10번 지	★강남구 ★건 2층 동명대로 481	서면동명대로점

-테이블 상세 정보

Field	Type	Null	Key	Default	Extra
customer_id	int(11)	NO	PRI	NULL	
customer_name	varchar(10)	NO		NULL	
customer_nickname	varchar(20)	YES		NULL	
customer_gender	varchar(5)	YES		NULL	
customer_birth	date	YES		NULL	
customer_age	int(11)	YES		NULL	
customer_phone	int(11)	YES		NULL	
customer_address	varchar(255)	YES		NULL	
customer_current	varchar(255)	YES		NULL	
customer_shop	varchar(255)	YES		NULL	

field에는customer_id,customer_name,customer_nickname,customer_gender, customer_birth,customer_age,customer_phone,customer_address,customer_current, customer_shop 이렇게 10개가 있고 각각 type에는 int(11),varchar(10),varchar(20) varchar(5),date,int(11),int(11),varchar(255),varchar(255),varchar(255)이다.
customer_id는 primary key이다.

⑤ 이벤트

입력 데이터

event_id	event_text	event_name
1	봄에 생일인 사람들을 위한 이벤트	봄 생일자 이벤트
2	여름에 생일인 사람들을 위한 이벤트	여름 생일자 이벤트
3	가을에 생일인 사람들을 위한 이벤트	가을 생일자 이벤트
4	겨울에 생일인 사람들을 위한 이벤트	겨울 생일자 이벤트

테이블 상세 정보

Field	Type	Null	Key	Default	Extra
event_id	int(11)	NO	PRI	NULL	
event_text	varchar(255)	YES		NULL	
event_name	varchar(255)	NO		NULL	

field에는 event_id,event_text,event_name 이렇게 3개가 있고 각각 type에는 int(11),varchar(255),varchar(255)이다.
event_id는 primary key이다.

⑥ 이벤트 상품

입력 데이터

event_product_id	event_product_name	event_product_information	event_product_price	event_id	event_product_origin
1	딸꾹 케이크	딸꾹만 먹을 땀 케이크는 어떨까요?	10000	1	한국
2	졸고래 케이크	졸고래만 먹을 졸고래 케이크는 어떨까요?	10000	2	중국
3	낙엽 케이크	떨어지는 낙엽만 먹을 낙엽 케이크는 어떨까요?	18000	3	한국
4	눈 케이크	떨어지는 눈만 먹을 눈 케이크는 어떨까요?	18000	4	한국
5	소풍 머그컵	떨어지는 겨울 소풍 머그컵은 어떨까요?	11000	1	한국

테이블 상세 정보

Field	Type	Null	Key	Default	Extra
event_product_id	int(11)	NO	PRI	NULL	
event_product_name	varchar(10)	NO		NULL	
event_product_information	varchar(255)	YES		NULL	
event_product_price	int(11)	NO		NULL	
event_id	int(11)	NO	MUL	NULL	
event_product_origin	varchar(255)	YES		NULL	

field에는 event_product_id,event_product_name,
event_product_information, event_product_origin,event_product_price,event_id 이렇게 6개가
있고 각각 type에는 int(11),varchar(10),varchar(255),int(11),int(11),varchar(255) 이다.
event_product_id는 primary key이고 event_id는 multi key를 가진다.

⑦ 주문

입력 데이터

order_detail_id	order_id	product_etc_id	event_product_id	option_id	order_date	order_price	order_type	customer_id
1	1	3	NULL	NULL	2021-11-01	40000	배달	1
2	2	NULL	NULL	1	2021-11-02	5000	배달	2
3	3	4	NULL	NULL	2021-11-03	19000	배달	3
4	4	NULL	NULL	2	2021-11-04	7000	사미권오더	4
5	5	NULL	NULL	3	2021-11-05	5500	사미권오더	5
6	6	NULL	NULL	4	2021-11-01	6000	사미권오더	6
7	7	NULL	NULL	5	2021-11-07	5000	사미권오더	7
8	8	5	NULL	NULL	2021-11-08	22000	사미권오더	8
9	9	NULL	NULL	6	2021-11-09	6000	사미권오더	9
10	10	8	NULL	NULL	2021-11-11	11000	사미권오더	10
11	11	NULL	NULL	7	2021-11-11	7000	사미권오더	1
12	12	9	NULL	NULL	2021-11-12	1200	사미권오더	2
13	13	NULL	NULL	8	2021-11-13	6000	사미권오더	3
14	14	NULL	NULL	9	2021-11-14	5000	사미권오더	4
15	15	NULL	NULL	10	2021-11-15	4000	사미권오더	5
16	16	NULL	1	NULL	2021-11-16	10000	사미권오더	6
17	17	NULL	NULL	2	2021-11-17	7000	사미권오더	8
18	18	NULL	NULL	3	2021-11-18	5500	배달	9
19	19	10	NULL	NULL	2021-11-19	14000	배달	10
20	20	NULL	NULL	5	2021-11-21	5000	배달	2
21	21	NULL	2	NULL	2021-11-21	10000	배달	7
22	22	NULL	3	NULL	2021-11-22	18000	배달	8
23	23	NULL	4	NULL	2021-11-23	18000	배달	9
24	1	NULL	5	NULL	2021-11-01	11000	배달	1
25	6	NULL	4	NULL	2021-11-01	18000	사미권오더	6
26	5	NULL	NULL	1	2021-11-05	5000	사미권오더	5
27	4	NULL	NULL	2	2021-11-04	7000	사미권오더	4
28	3	NULL	NULL	4	2021-11-03	5500	사미권오더	3
29	2	5	NULL	NULL	2021-11-02	22000	사미권오더	2
30	1	NULL	1	NULL	2021-11-01	10000	사미권오더	1

-테이블 상세 정보

Field	Type	Null	Key	Default	Extra
order_detail_id	int(11)	NO	PRI	NULL	
order_id	int(11)	NO		NULL	
product_etc_id	int(11)	YES	MUL	NULL	
event_product_id	int(11)	YES	MUL	NULL	
option_id	int(11)	YES	MUL	NULL	
order_date	date	NO		NULL	
order_price	int(11)	NO		NULL	
order_type	varchar(10)	NO		NULL	
customer_id	int(11)	NO		NULL	

field에는 order_detail_id,order_date,order_price,order_type,customer_id,order_id,option_id2,product_bev_id,event_product_id,product_etc_id 이렇게 9개가 있고
 각각 type에는 int(11),int(11),int(11),int(11),int(11),date,int(11),,varchar(10),int(11)이다.
 order_detail_id는 primary key이고 product_etc_id는 multi key,event_product_id 도multi key,option_id도 multi key를 가진다.

⑧ 고객 결제정보

입력 데이터

payment_id	cash_receipt	customer_id
1	59134555	1
2	65328844	2
3	54654412	3
4	87854541	4
5	85515458	5
6	74516391	6
7	46337499	7
8	23212323	8
9	23134153	9
10	85662131	10

테이블 상세 정보

Field	Type	Null	Key	Default	Extra
payment_id	int(11)	NO	PRI	NULL	
cash_receipt	int(11)	YES		NULL	
customer_id	int(11)	NO	MUL	NULL	

field에는 payment_id,cash_receipt,customer_id 이렇게 3개가 있고 type에는 int(11), int(11),int(11)이다.

payment_id는 primary key이고,customer_id는 multi key이다.

⑨ 스타벅스카드

입력 데이터

starbucks_card_id	payment_id	starbucks_card_name	starbucks_card_number
1	1	맹 새	556855215
2	2	홍 길 동	455112256
3	3	김 길 동	448845555
4	4	나 길 동	465875412
5	5	김 수 갈	875643215
6	6	김 수 박	487564215
7	7	김 호 박	878754654
8	8	선 포 도	965233654
9	9	김 트 성	484811556
10	10	우 록 성	787546565

테이블 상세 정보

Field	Type	Null	Key	Default	Extra
starbucks_card_id	int(11)	NO	PRI	NULL	
payment_id	int(11)	NO	PRI	NULL	
starbucks_card_name	varchar(10)	YES		NULL	
starbucks_card_number	int(11)	YES		NULL	

field에는 starbucks_card_id,payment_id,starbucks_card_name, starbucks_card_number 이렇게 4개가 있고

type에는 각각 int(11),int(11),varchar(10),int(11)이다.

starbucks_card_id는 primary key이고,payment_id도 primary key이다.

⑩ 신용카드

입력 데이터

credit_card_id	payment_id	credit_name	credit_number
1	4	농협	4
2	5	신협	5
3	8	국민	6
4	7	기타	7
5	1	카카오	3
6	3	토스	1
7	2	부산	2
8	6	정남	9
9	9	하나	8
10	10	우리	10

테이블 상세 정보

Field	Type	Null	Key	Default	Extra
credit_card_id	int(11)	NO	PRI	NULL	
payment_id	int(11)	NO	PRI	NULL	
credit_name	varchar(20)	YES		NULL	
credit_number	int(11)	YES		NULL	

field에는 credit_card_id,payment_id,credit_name,credit_number 이렇게 4개가 있고
type에는 각각 int(11),int(11),varchar(20),int(11)이다.

credit_card_id는 primary key이고,payment_id도 primary key이다.

⑪ 리뷰

입력 데이터

review_id	customer_id	product_bev_id	review_text	review_score	review_date
1	1	1	정말 맛있어요	5	2021-10-01
2	1	2	맛있어요	1	2021-10-02
3	2	3	노맛	1	2021-10-03
4	2	4	이거 맛있으니 꼭 나눠주세요	5	2021-10-04
5	3	5	시원함을 느껴봐요 맛있게 보세요!	5	2021-10-05
6	3	6	달달하니 좋습니다	5	2021-10-06
7	4	7	편하고 맛있게 맛있어요	5	2021-10-07
8	4	8	오늘은 최고 맛있게 먹었는데 파로해서 맛있었어요	5	2021-10-08
9	5	9	너무 좋아요...	2	2021-10-09
10	5	10	리소넬 옵션이 있어서 인스타 게시글 해왔는데 맛있었어요	5	2021-10-10
11	6	1	소스소스하게 맛있어요	3	2021-10-11
12	6	2	맛있고 좋습니다	5	2021-10-12
13	7	3	친구랑 같이 먹었는데 참 맛있었던 것...	5	2021-10-13
14	7	4	커피는 사랑합니다.	5	2021-10-14
15	8	5	디저트는 사랑합니다	5	2021-10-15
16	8	6	상큼한 자몽은 내가 상큼해	5	2021-10-16
17	9	7	여러분들...	5	2021-10-17
18	9	8	노맛이에요...	2	2021-10-18
19	10	9	개노맛	2	2021-10-19
20	10	10	은맛	5	2021-10-20

테이블 상세 정보

Field	Type	Null	Key	Default	Extra
review_id	int(11)	NO	PRI	NULL	
customer_id	int(11)	NO	PRI	NULL	
product_bev_id	int(11)	NO	PRI	NULL	
review_text	varchar(255)	NO		NULL	
review_score	int(11)	NO		NULL	
review_date	date	YES		NULL	

field에는 review_id,customer_id,product_bev_id,review_text,review_score,review_date 이렇게 4개가 있고 type에는 각각 int(11),int(11),int(11),varchar(255),int(11),date이다.

review_id는 primary key이고,customer_id도 primary key이고,product_bev_id도 primary key이다.

⑫ 알레르기유발요인

입력 데이터

allergy_id	allergy_soybean	allergy_milk	allergy_turbulence	allergy_wheat	allergy_customer	product_bev_id
1	yes	yes	no	no	yes	1
2	yes	no	yes	no	yes	6
3	no	no	no	yes	yes	2
4	yes	yes	yes	no	yes	7
5	yes	no	yes	no	yes	3
6	no	no	no	yes	yes	8
7	yes	no	yes	no	yes	4
8	no	no	no	yes	yes	9
9	no	yes	no	yes	yes	5

테이블 상세 정보

Field	Type	Null	Key	Default	Extra
allergy_id	int(11)	NO	PRI	NULL	
allergy_soybean	varchar(10)	YES		NULL	
allergy_milk	varchar(10)	YES		NULL	
allergy_turbulence	varchar(10)	YES		NULL	
allergy_wheat	varchar(10)	YES		NULL	
allergy_customer	varchar(10)	YES		NULL	
product_bev_id	int(11)	NO	MUL	NULL	

field에는 allergy_id,allergy_soybean,allergy_milk,allergy_turbulence,allergy_wheat,allergy_customer,allergy_customer,product_bev_id 이렇게 7개가 있고

type에는 각각 int(11),

varchar(10),varchar(10),varchar(10),varchar(10),varchar(10),int(11)이다.

allergy_id는 primary key이고,product_bev_id는 multi key이다.

⑬ 영양정보

입력 데이터

nutrition_id	calorie	carbohydrate	sugars	sodium	protein	fat	cholesterol	trans_fat	caffeine	saturated_fat	product_bev_id
1	300	29	120	10	200	11	11	1	11	3	1
2	500	50	70	55	110	22	22	2	11	4	2
3	350	30	40	70	70	33	33	3	11	5	3
4	340	45	20	20	75	44	44	4	11	6	4
5	280	42	100	50	55	55	55	5	11	7	5
6	370	66	80	40	66	66	66	6	11	8	6
7	460	72	90	33	77	77	77	7	11	9	7
8	550	80	75	45	25	88	88	8	11	10	8
9	250	90	25	53	46	99	99	9	11	11	9
10	180	120	40	51	86	110	110	10	11	12	10

테이블 상세정보

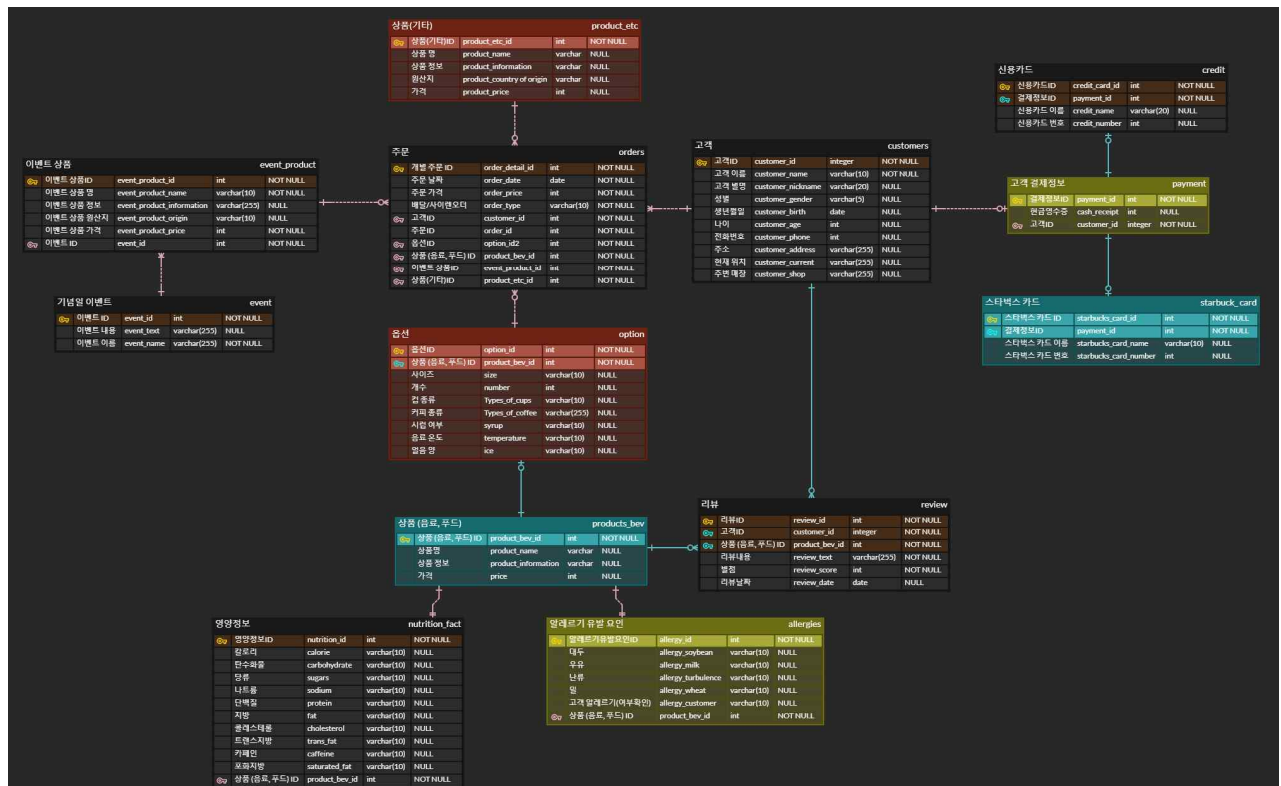
Field	Type	Null	Key	Default	Extra
nutrition_id	int(11)	NO	PRI	NULL	
calorie	varchar(10)	YES		NULL	
carbohydrate	varchar(10)	YES		NULL	
sugars	varchar(10)	YES		NULL	
sodium	varchar(10)	YES		NULL	
protein	varchar(10)	YES		NULL	
fat	varchar(10)	YES		NULL	
cholesterol	varchar(10)	YES		NULL	
trans_fat	varchar(10)	YES		NULL	
caffeine	varchar(10)	YES		NULL	
saturated_fat	varchar(10)	YES		NULL	
product_bev_id	int(11)	NO	MUL	NULL	

field에는 nutrition_id,calorie,carbohydrate,sugars,sodium,protein,fat,cholesterol, trans_fat,caffeine,saturated_fat,product_bev_id 이렇게 12개가 있고
type에는 각각 int(11),
varchar(10),varchar(10),varchar(10),varchar(10),varchar(10),varchar(10),varchar(10),va
rchar(10),varchar(10),int(11)이다.
nutrition_id는 primary key이고, product_bev_id는 multi key이다.

4. 결론

본 프로젝트는 스타벅스 어플리케이션을 참고하여 데이터베이스 개념적 설계, 논리적 설계, 물리적 설계 순서대로 진행되었다.

개념적 설계시에는 개체의 관계와 속성을 추출하였다. 논리적 설계를 진행하기 전, 다대다 관계는 관계테이블을 만들어 일대다 관계로 변환하였다. 논리적 설계는 릴레이션 변환 규칙을 참고하여 진행했다. 그 결과 아래와 같은 다이어그램으로 나타낼 수 있었다.



기존 스타벅스 어플리케이션을 참고하여 데이터베이스를 설계한 후 상품에 리뷰 달기 기능과, 기념일 이벤트 기능을 추가했다. 리뷰 기능과 이벤트 기능을 추가하기 위해서 고객과 상품의 기본키를 외래키로 가지는 리뷰 테이블을 추가하였다. 주문 테이블과 연결되는 이벤트 상품 테이블과 기념일 이벤트 테이블을 추가하였다. 기존의 스타벅스 서비스를 제공하기 위한 데이터베이스 설계를 나타내 보는 것 뿐만 아니라, 새로운 서비스를 제안하기 위한 데이터베이스를 나타내보면서 서비스 제안 시의 데이터베이스 측면을 확인할 수 있다.

5. 부록

- 데이터 정의어

```

DROP TABLE IF EXISTS customers;
CREATE TABLE customers (
  customer_id integer NOT NULL,
  customer_name varchar(10) NOT NULL,
  customer_nickname varchar(20) NULL,
  customer_gender varchar(5) NULL,
  customer_birth date NULL,
  customer_age int NULL,
  customer_phone int NULL,
  customer_address varchar(255) NULL,
  customer_current varchar(255) NULL,
  customer_shop varchar(255) NULL
);
DROP TABLE IF EXISTS product_etc;

```

```
CREATE TABLE product_etc (  
    product_etc_id int NOT NULL,  
    product_name varchar(10) NULL,  
    product_information varchar(255) NULL,  
    product_country_of_origin varchar(10) NULL,  
    product_price int NULL  
);  
  
DROP TABLE IF EXISTS option;  
CREATE TABLE option (  
    option_id int NOT NULL,  
    product_bev_id int NOT NULL,  
    size varchar(10) NULL,  
    number int NULL,  
    Types_of_cups varchar(10) NULL,  
    Types_of_coffee varchar(10) NULL,  
    syrup varchar(10) NULL,  
    temperature varchar(10) NULL,  
    ice varchar(10) NULL  
);  
  
DROP TABLE IF EXISTS orders;  
CREATE TABLE orders (  
    order_id int NOT NULL,  
    product_bev_id int NULL,  
    product_etc_id int NULL,  
    event_product_id int NOT NULL,  
    order_date date NOT NULL,  
    order_price int NOT NULL,  
    order_type varchar(10) NOT NULL,  
    customer_id int NOT NULL  
);  
  
DROP TABLE IF EXISTS products_bev;  
CREATE TABLE products_bev (  
    product_bev_id int NOT NULL,  
    product_name varchar(10) NULL,  
    product_information varchar(255) NULL,  
    price int NULL  
);  
  
DROP TABLE IF EXISTS starbucks_card;  
CREATE TABLE starbucks_card (  
    starbucks_card_id int NOT NULL,  
    payment_id int NOT NULL,  
    starbucks_card_name varchar(10) NULL,  
    starbucks_card_number int NULL  
);
```

```
DROP TABLE IF EXISTS allergies;
CREATE TABLE allergies (
  allergy_id int NOT NULL,
  allergy_soybean varchar(10) NULL,
  allergy_milk varchar(10) NULL,
  allergy_turbulence varchar(10) NULL,
  allergy_wheat varchar(10) NULL,
  allergy_customer varchar(10) NULL,
  product_bev_id int NOT NULL
);
DROP TABLE IF EXISTS payment;
CREATE TABLE payment (
  payment_id int NOT NULL,
  cash_receipt int NULL,
  customer_id integer NOT NULL
);
DROP TABLE IF EXISTS credit;
CREATE TABLE credit (
  credit_card_id int NOT NULL,
  payment_id int NOT NULL,
  credit_name varchar(20) NULL,
  credit_number int NULL
);
DROP TABLE IF EXISTS nutrition_fact;
CREATE TABLE nutrition_fact (
  nutrition_id int NOT NULL,
  calorie varchar(10) NULL,
  carbohydrate varchar(10) NULL,
  sugars varchar(10) NULL,
  sodium varchar(10) NULL,
  protein varchar(10) NULL,
  fat varchar(10) NULL,
  cholesterol varchar(10) NULL,
  trans_fat varchar(10) NULL,
  caffeine varchar(10) NULL,
  saturated_fat varchar(10) NULL,
  product_bev_id int NOT NULL
);
DROP TABLE IF EXISTS event;
CREATE TABLE event (
  event_id int NOT NULL,
  event_text varchar(255) NULL,
  event_name varchar(255) NOT NULL
);
DROP TABLE IF EXISTS event_product;
```

```
CREATE TABLE event_product (  
    event_product_id int NOT NULL,  
    event_product_name varchar(10) NOT NULL,  
    event_product_information varchar(255) NULL,  
    event_product_country_of_origin varchar(10) NULL,  
    event_product_price int NOT NULL,  
    event_id int NOT NULL  
);  
DROP TABLE IF EXISTS review;  
CREATE TABLE review (  
    review_id int NOT NULL,  
    customer_id integer NOT NULL,  
    product_bev_id int NOT NULL,  
    review_text varchar(255) NOT NULL,  
    review_score int NOT NULL,  
    review_date date NULL  
);  
ALTER TABLE customers ADD CONSTRAINT PK_CUSTOMERS PRIMARY KEY (  
    customer_id  
);  
ALTER TABLE product_etc ADD CONSTRAINT PK_PRODUCT_ETC PRIMARY KEY (  
    product_etc_id  
);  
ALTER TABLE option ADD CONSTRAINT PK_OPTION PRIMARY KEY (  
    option_id,  
    product_bev_id  
);  
  
ALTER TABLE orders ADD CONSTRAINT PK_ORDERS PRIMARY KEY (  
    order_id,  
    product_bev_id,  
    product_etc_id,  
    event_product_id  
);  
ALTER TABLE products_bev ADD CONSTRAINT PK_PRODUCTS_BEV PRIMARY KEY (  
    product_bev_id  
);  
ALTER TABLE starbucks_card ADD CONSTRAINT PK_STARBUCKS_CARD PRIMARY KEY (  
    starbucks_card_id,  
    payment_id  
);  
ALTER TABLE allergies ADD CONSTRAINT PK_ALLERGIES PRIMARY KEY (  
    allergy_id  
);  
ALTER TABLE payment ADD CONSTRAINT PK_PAYMENT PRIMARY KEY (  

```



```

    payment_id
);
ALTER TABLE credit ADD CONSTRAINT PK_CREDIT PRIMARY KEY (
    credit_card_id,
    payment_id
);
ALTER TABLE nutrition_fact ADD CONSTRAINT PK_NUTRITION_FACT PRIMARY KEY (
    nutrition_id
);
ALTER TABLE event ADD CONSTRAINT PK_EVENT PRIMARY KEY (
    event_id
);
ALTER TABLE event_product ADD CONSTRAINT PK_EVENT_PRODUCT PRIMARY KEY (
    event_product_id
);
ALTER TABLE review ADD CONSTRAINT PK_REVIEW PRIMARY KEY (
    review_id,
    customer_id,
    product_bev_id
);
ALTER TABLE option ADD CONSTRAINT FK_products_bev_TO_option_1 FOREIGN KEY (
    product_bev_id
)REFERENCES products_bev (
    product_bev_id
);
ALTER TABLE orders ADD CONSTRAINT FK_products_bev_TO_orders_1 FOREIGN KEY (
    product_bev_id
)REFERENCES products_bev (
    product_bev_id
);
ALTER TABLE orders ADD CONSTRAINT FK_product_etc_TO_orders_1 FOREIGN KEY (
    product_etc_id
)REFERENCES product_etc (
    product_etc_id
);
ALTER TABLE orders ADD CONSTRAINT FK_event_product_TO_orders_1 FOREIGN KEY (
    event_product_id
)REFERENCES event_product (
    event_product_id
);
ALTER TABLE orders ADD CONSTRAINT FK_customers_TO_orders_1 FOREIGN KEY (
    customer_id
)REFERENCES customers (
    customer_id
);

```

```
ALTER TABLE starbucks_card ADD CONSTRAINT FK_payment_TO_starbucks_card_1 FOREIGN KEY (
    payment_id
)REFERENCES payment (
    payment_id
);
ALTER TABLE allergies ADD CONSTRAINT FK_products_bev_TO_allergies_1 FOREIGN KEY (
    product_bev_id
)REFERENCES products_bev (
    product_bev_id
);
ALTER TABLE payment ADD CONSTRAINT FK_customers_TO_payment_1 FOREIGN KEY (
    customer_id
)REFERENCES customers (
    customer_id
);
ALTER TABLE credit ADD CONSTRAINT FK_payment_TO_credit_1 FOREIGN KEY (
    payment_id
)REFERENCES payment (
    payment_id
);

ALTER TABLE nutrition_fact ADD CONSTRAINT FK_products_bev_TO_nutrition_fact_1 FOREIGN KEY (
    product_bev_id
)REFERENCES products_bev (
    product_bev_id
);
ALTER TABLE event_product ADD CONSTRAINT FK_event_TO_event_product_1 FOREIGN KEY (
    event_id
)REFERENCES event (
    event_id
);
ALTER TABLE review ADD CONSTRAINT FK_customers_TO_review_1 FOREIGN KEY (
    customer_id
)REFERENCES customers (
    customer_id
);
ALTER TABLE review ADD CONSTRAINT FK_products_bev_TO_review_1 FOREIGN KEY (
    product_bev_id
)REFERENCES products_bev (
    product_bev_id
);
```