**데이터베이스 기말과제 보고서**

202014889 김민상

1. 기말 과제 제안서 제출된 내용과 대비 차이점, 발전성

리뷰만 있으면 음식에 대한 리뷰인지 가게에 대한 리뷰인지 불분명하므로 리뷰 개체를 가게리뷰로 이름을 바꾸어 의미를 명확히 하였다. 가게리뷰와 가게를 존재 관계로 연결하였다. 주문목록 개체를 삭제하고 회원이 메뉴를 주문하는 관계로 수정했다. 약한 개체였던 멤버십을 수정하였다. 가게 개체의 속성으로 총 리뷰 수, 합산 별점이 추가되었다. 또한, 속성을 명확하게 표현하기 위해 평점을 평균평점으로 수정했다. 유도 속성의 오류를 줄이기 위한 발전이라고 볼 수 있다. 또한, 평균 평점의 속성을 구하는 과정에서 총 리뷰 수 속성의 데이터로 나누는 부분이 있는데, 수학적인 오류를 없애고 논리적인 모순을 제거하기 위해(데이터 무결성) 나누는 값은 0보다 크다는 문장을 추가하였다.

2. 요구 사항 명세서

온라인 음식 배달 앱 회원은 회원ID, 비밀번호, 전화번호, 종류, 이메일, 현주소, 생성날짜 정보를 갖는다. 회원은 회원ID 로 식별한다. 종류는 일반, 사장 정보로 세분화된다. 현주소는 도, 시, 동, 상세주소 정보로 세분화된다.

회원은 가게리뷰를 작성할 수 있다. 가게리뷰는 리뷰번호, 별점, 내용, 생성일 속성을 갖는다. 가게리뷰는 리뷰번호로 식별한다. 한 명의 회원은 여러 개의 가게리뷰를 작성할 수 있고, 가게리뷰 한 개는 한 명의 회원에 의해서만 작성된다. 모든 가게리뷰는 반드시 회원에 의해서만 작성되어야 한다.

회원은 멤버십을 구독할 수 있다. 멤버십은 멤버십코드(고유함), 등급, 생성날짜, 소멸날짜, 상태 속성을 갖는다. 등급은 브론즈, 실버, 골드 정보로 세분화 될 수 있다.

한 명의 회원은 하나의 멤버십을 구독할 수 있고, 하나의 멤버십은 한 명의 회원에 의해 구독된다.

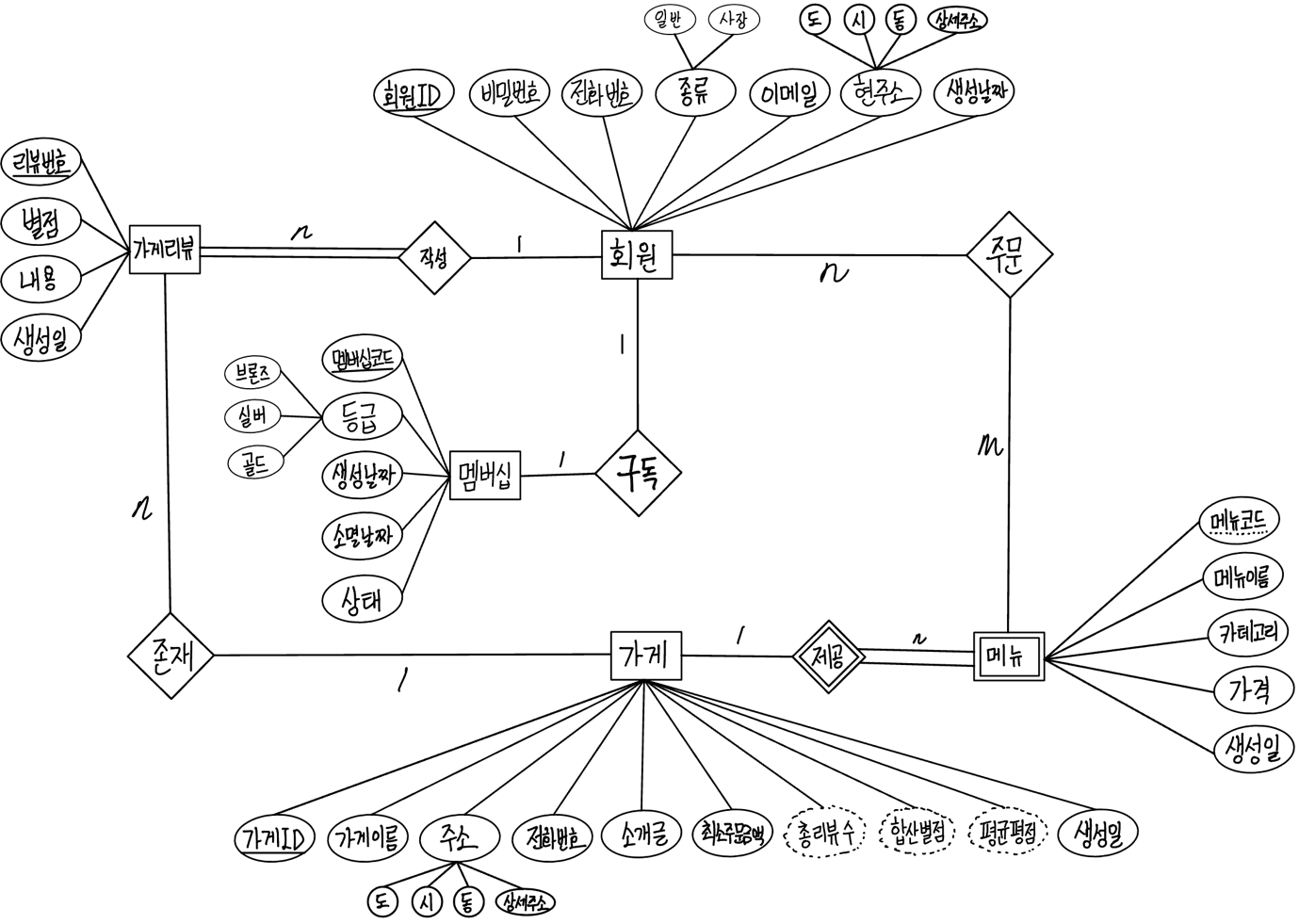
가게는 가게ID, 가게이름, 주소, 전화번호, 소개글, 최소주문금액, 총 리뷰 수, 합산 별점, 평균 평점, 생성일 정보를 갖는다. 가게는 가게ID로 식별한다. 주소는 도, 시, 동, 상세주소 정보로 세분화된다. 총 리뷰 수는 입력받는 것이 아니라, 해당 가게에 부여된 가게리뷰의 개수에 따라 결정된다. 합산 별점은 입력받는 것이 아니라, 해당 가게에 부여된 가게리뷰 중 별점 속성 데이터들의 합으로 결정된다. 평균 평점은 입력받는 것이 아니라, 합산 별점 속성의 값을 총 리뷰 수로 나눈 값으로 결정된다. 이때, 총 리뷰 수는 0보다 크다.

가게에는 가게리뷰가 존재한다. 하나의 가게에는 여러 개의 가게리뷰가 존재할 수 있고, 가게리뷰 한 개는 하나의 가게만을 대상으로 존재한다.

가게는 메뉴를 제공한다. 메뉴는 메뉴코드(고유함), 메뉴이름, 카테고리, 가격, 생성일 정보를 갖는다. 하나의 가게는 여러 개의 메뉴를 제공할 수 있으며, 하나의 메뉴는 하나의 가게에 의해서만 제공된다. 가게가 폐점하면 가게에서 제공하던 메뉴 정보도 함께 소멸된다.

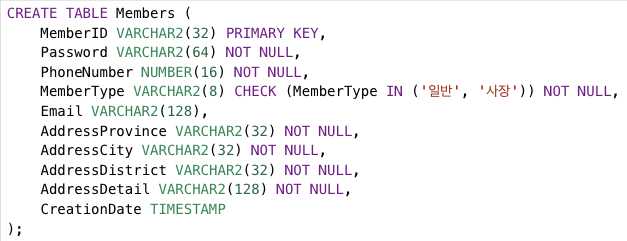
회원은 메뉴를 주문한다. 한 명의 회원은 여러 개의 메뉴를 주문 할 수 있고, 하나의 메뉴는 여러 명의 회원에게 주문될 수 있다.

3. ER 다이어그램



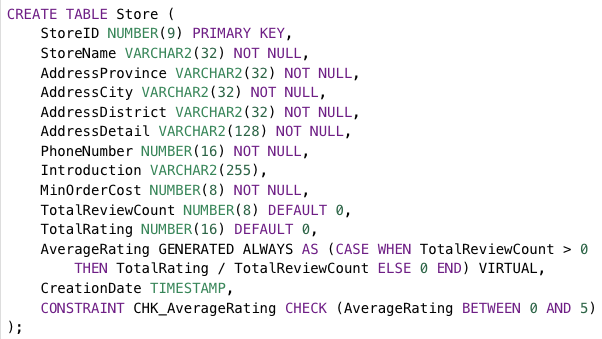
4. 논리 모델링(관계형 모델) - 테이블 정의서(+속성 및 제약조건에 대한 간단한 설명)

1) 회원 테이블



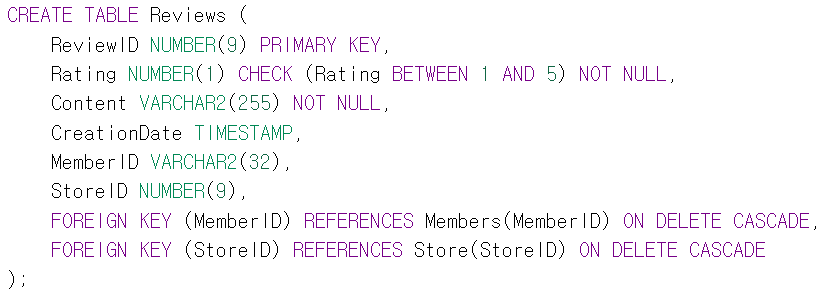
회원ID를 기본키로 하였고 복합 속성인 종류 속성과 현주소 속성은 하위 속성들로 나누어 작성하였다. 종류 속성에 데이터 삽입 또는 수정 시 ‘일반’ 또는 ‘사장’ 문자만 가능하도록 CHECK를 이용했다. 생성날짜 속성은 TIMESTAMP를 이용하여 표현했다. 배달 음식 앱이므로 주소 속성에서 나뉜 하위 속성들은 NULL값을 허용하지 않았고, 이메일 속성은 생략 가능하도록 했다.

2) 가게 테이블



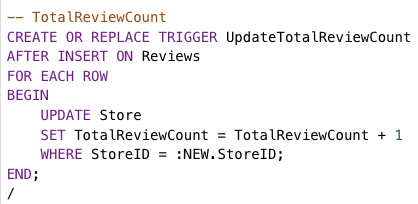
가게ID를 기본키로 하였고, 복합 속성인 주소는 하위 속성들로 나눠서 작성했다. 역시나, 주소 속성에서 파생된 하위 속성들은 NULL값을 허용하지 않았고, 소개글 속성은 NULL값을 허용했다. 생성일 속성은 TIMESTAMP를 이용하여 표현했으며 유도 속성인 총리뷰수, 합산별점, 평균평점을 표현하기 위해 트리거를 이용했다. 유도 속성 관련 설명들은 코드 특성상 가게리뷰 테이블의 내용이 필요하므로 아래에서 설명하겠다.

3) 가게리뷰 테이블

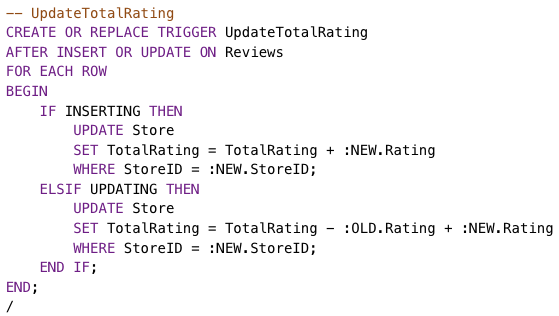


리뷰번호를 기본키로 하며, 별점은 1부터 5사이의 수만 가질 수 있도록 CHECK를 사용했다. 가게리뷰 개체는 회원 개체, 가게 개체와 각각 다대일 관계이므로 외래키로 각 개체의 기본키를 참조했다. 그리고 외래키로 연결된 데이터들이 일관성을 유지할 수 있도록 ON DELETE CASCADE 옵션을 적용했다.

\*\*\*(아래 두 사진은 테이블이 아니지만 앞서 언급한 유도 속성들을 설명하는 부분)\*\*\*

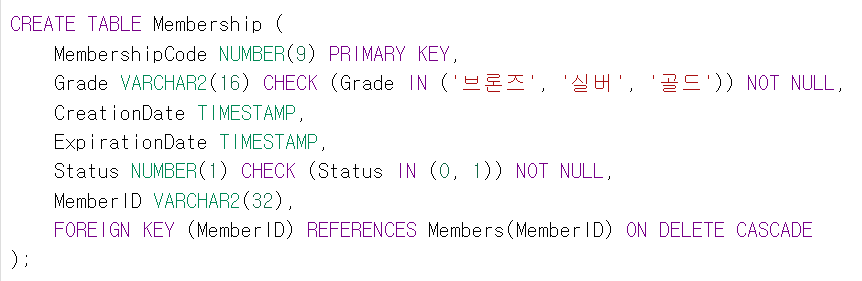


총리뷰수를 업데이트하는 트리거다. 리뷰 테이블에 새로운 행이 삽입되면, 삽입된 리뷰의 가게ID에 해당하는 가게의 총 리뷰수를 1증가시킨다.



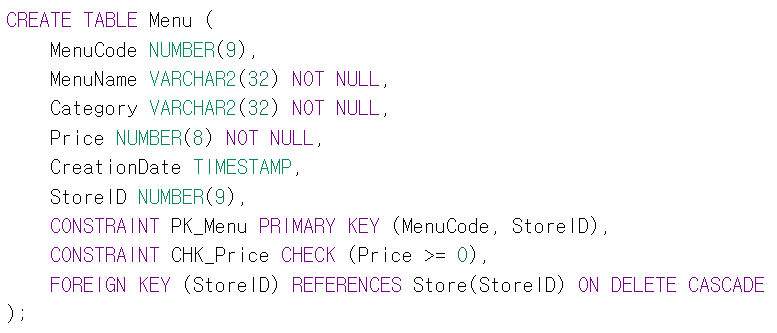
합산별점을 업데이트하는 트리거다. 리뷰 테이블에 새로운 행이 삽입되거나 기존 행이 갱신되는 경우 2가지로 나뉜다. 삽입되는 경우는, 해당 리뷰의 가게ID에 해당하는 가게 테이블의 합산별점에 새로운 평점을 더한다. 갱신되는 경우는, 갱신된 리뷰의 가게ID에 해당하는 가게의 합산별점에서 이전 별점을 빼고 새 별점을 더한다.

4) 멤버십 테이블



멤버십코드를 기본키로 하며, 등급에 대해 ‘브론즈’, ‘실버’, ‘골드’만 입력받을 수 있도록 CHECK로 작성했다. 생성날짜와 소멸날짜는 TIMESTAMP로 작성, 상태에 대해서는 0과 1(0은 사용불가, 1은 사용가능)의 숫자만 가질 수 있도록 만들었다. 외래키로 회원 테이블의 회원ID를 참조한다. 그리고 외래키로 연결된 데이터들이 일관성을 유지할 수 있도록 ON DELETE CASCADE 옵션을 적용했다.

5) 메뉴 테이블



메뉴 개체는 약한 개체이다. 메뉴 코드는 구별자이며, 강한 개체인 가게의 기본키 가게ID와 함께 있어야 메뉴의 기본키가 된다. 이는 CONSTRAINT로 제약조건을 검사하도록 했다. 가게ID는 외래키로 참조하였다. 생성일 속성은 TIMESTAMP를 이용했다. 가격 속성은 0이상의 값을 가져야 하는 CHECK 제약 조건을 걸었다. 그리고 외래키로 연결된 데이터들이 일관성을 유지할 수 있도록 ON DELETE CASCADE 옵션을 적용했다.

6) 주문 테이블

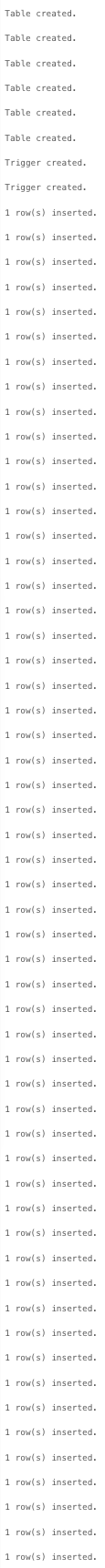
텍스트, 스크린샷, 폰트이(가) 표시된 사진

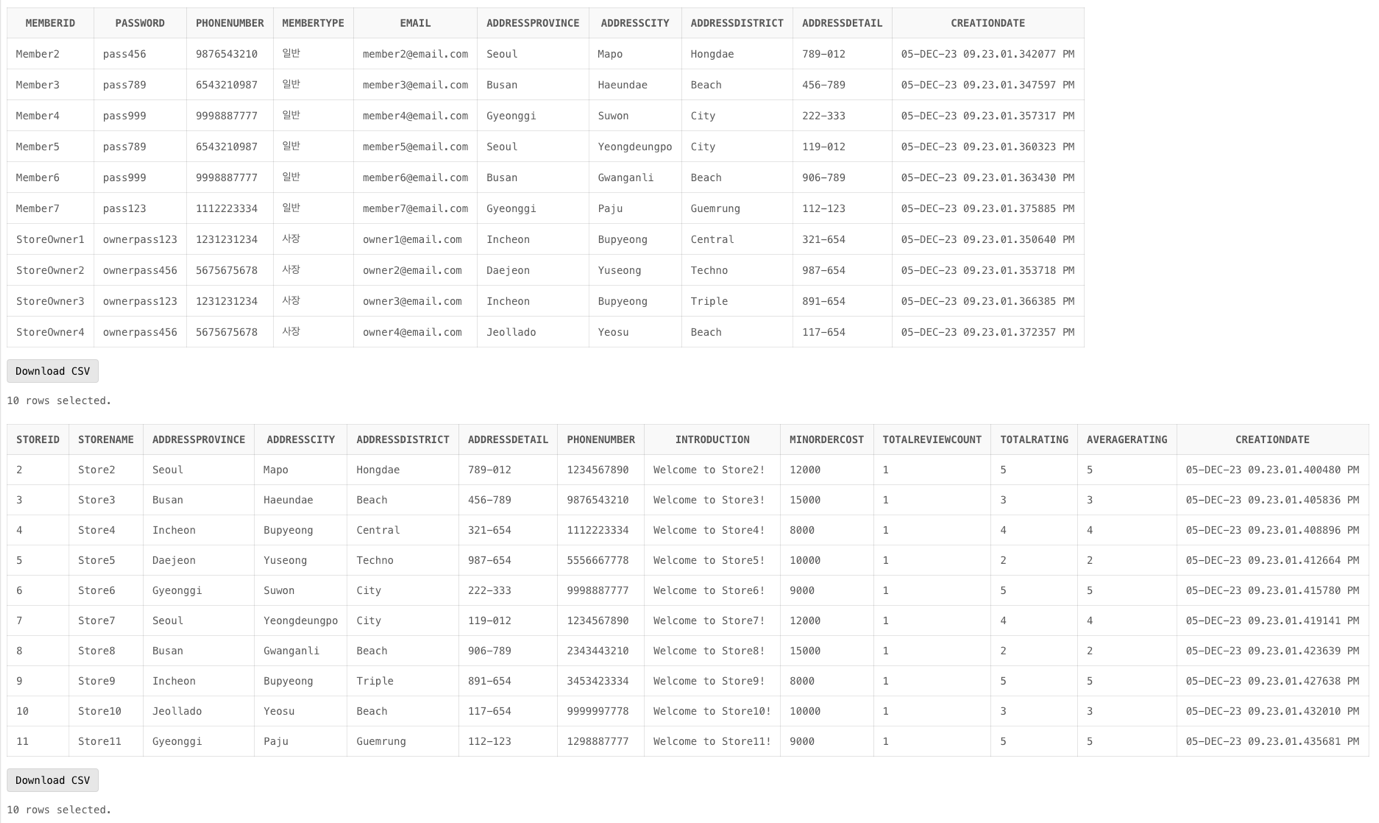
자동 생성된 설명

회원 개체와 메뉴 개체가 다대다 관계이기 때문에 주문 테이블을 작성했다. 메뉴 개체는약한 개체라서 가게ID와 메뉴코드 기본키를 가진다. 이에 맞추어 제약조건을 작성했고, 외래키로 연결된 데이터들이 일관성을 유지할 수 있도록 ON DELETE CASCADE 옵션을 적용했다.

5. 구현 - SQL문 실행(실행 결과 캡쳐)

1) 데이터 잘 들어갔는지 확인





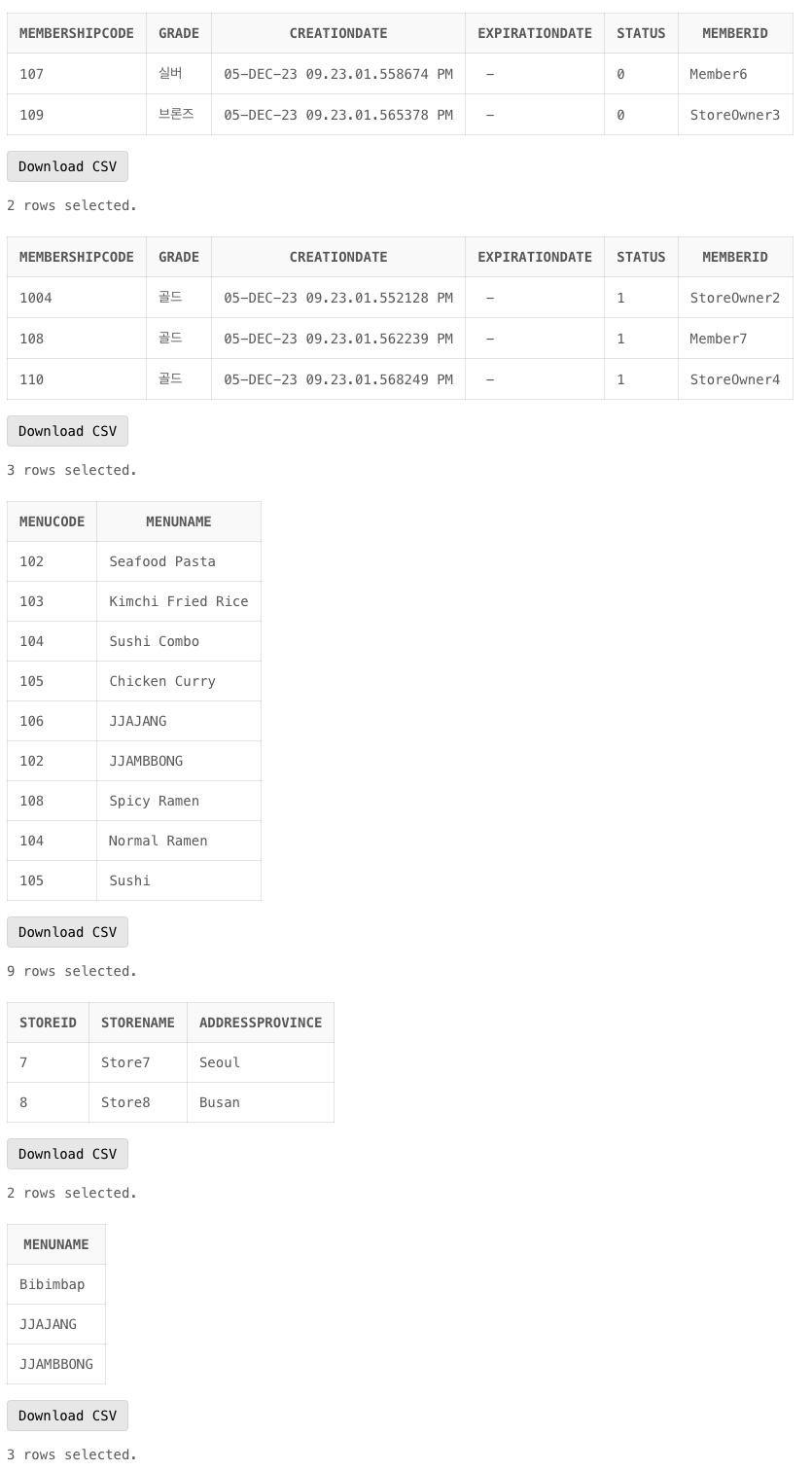


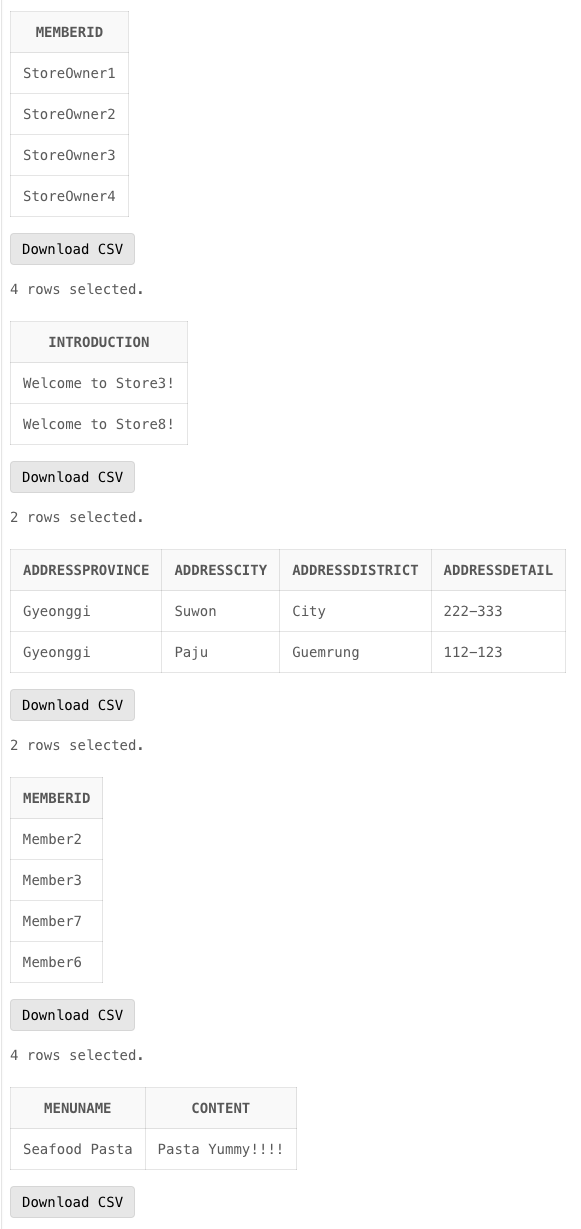


텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

2) 데이터 검색





3) 데이터 삽입

텍스트, 영수증, 문서, 디자인이(가) 표시된 사진

자동 생성된 설명

텍스트, 영수증, 문서이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

4) 데이터 수정

텍스트, 문서, 영수증이(가) 표시된 사진

자동 생성된 설명

텍스트, 영수증, 문서이(가) 표시된 사진

자동 생성된 설명

5) 데이터 삭제

텍스트, 문서, 영수증, 패턴이(가) 표시된 사진

자동 생성된 설명

텍스트, 영수증, 문서이(가) 표시된 사진

자동 생성된 설명

6. 소스 파일

압축하여 제출하였습니다!