데이터베이스시스템 project2 - Convenience Stores

20221212 이준석

1. Logical Schema Design (BCNF)

텍스트, 도표, 스크린샷, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

(bold체 + 밑줄친 속성 = primary key , bold체만 = foreign key, 나머지 = 일반속성)

1. stores: store\_id는 name, address, open\_time, close\_time, ownership\_type을 결정한다.

그리고 다른 일반 속성들간에 함수적 종속성이 없으므로 bcnf를 만족한다.

2. product: product\_upc는 name, brand, package\_type, size, price를 결정한다.

그리고 다른 일반 속성들간에 함수적 종속성이 없으므로 bcnf를 만족한다.

추가로 erd에는 없었던 name이라는 속성을 이번 과제에서 추가했다.

3. inventory: store\_id와 product\_upc가 각각 외래키이며, 합쳐서 Primary Key로 설정했다. 그리고 이 primary key는 inventory\_level, reorder\_thresholds, reorder\_quantitys, recent\_order\_history를 결정한다. 그리고 다른 일반 속성들간에 함수적 종속성이 없으므로 bcnf를 만족한다.

4. sales transactions: store\_id, product\_upc, customer\_id 는 각각 외래키이며 따로 sale\_id라는 primary key로 인덱스 하였다. 그리고 다른 일반 속성들간에 함수적 종속성이 없으므로 bcnf를 만족한다. 추가로 erd에는 없었던 buy\_date이라는 속성을 추가했다.

5. customer: customer\_id는 name, phone\_num, email, loyalty를 결정한다.

그리고 다른 일반 속성들간에 함수적 종속성이 없으므로 bcnf를 만족한다.

6. vendor: vendor\_id는  name, phone\_num, email을 결정한다.

그리고 다른 일반 속성들간에 함수적 종속성이 없으므로 bcnf를 만족한다.

7. product\_vendor: product\_upc, vendor\_id는 각각 외래키이며 합쳐서 Primary Key로 설정했다. 이 primary key는 product\_num을 결정한다. 그리고 다른 일반 속성들간에 함수적 종속성이 없으므로 bcnf를 만족한다.

2. Physical Schema Design

텍스트, 폰트, 스크린샷, 친필이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

check관련 내용은 아래 mysql에 적어두었다.

1. stores:

Store\_id : 편의점의 id, int 정수 숫자이며 PIRMARY KEY이다.

name: 편의점의 이름 , VARCHAR(50) 문자열이며 NOT NULL 이다.

Address: 편의점의 주소, VARCHAR(50) 문자열이며 NOT NULL 이다.

Open\_time: 편의점의 오픈시간 , TIME인 00:00:00으로 표현되며 NOT NULL 이다.

Close time: 편의점의 닫는시간, time 인 00:00:00으로 표현되며 NOT NULL 이다.

Ownership\_type: 편의점의 운영주체, VARCHAR(50) 문자열이며 NOT NULL 이다. 프랜차이즈, 직영점, 개인으로 나뉜다.

2. product:

Product\_upc: 물건의 id값, VARCHAR(50)인 문자열이며 primary key이다.

Name: 물건의 이름, VARCHAR(50) 문자열이며 NOT NULL 이다.

Brand: 물건의 브랜드, VARCHAR(50) 문자열이며 NOT NULL 이다.

Package\_type, 물건의 포장 방식, VARCHAR(50) 문자열이며 NOT NULL 이다. 박스, 병, 팩으로 나뉜다.

Size: 물건의 크기, VARCHAR(50) 문자열이며 NOT NULL 이다. Large, medium,small으로 나뉜다.

Price: 물건의 가격, DECIMAL(10,2) 인 소수 둘째자리까지 나타내는 숫자이며 NOT NULL이다.

3. inventory:

Store\_id: 편의점의 id, int 정수 숫자이며 FOREIGN KEY이다. Stores 테이블에서 참조하고 stores에서 삭제된다면 여기서도 삭제된다.

Product\_upc: 물건의 id값, VARCHAR(50)인 문자열이며 foreign key이다. products 테이블에서 참조하고 products에서 삭제된다면 여기서도 삭제된다.

Inventory\_level: 현재 재고수량, int 정수 숫자이며 NOT NULL이다. 0이상의값이다.

Reorder\_\_thresholds: 최소 재고 유지 수량, int 정수 숫자이며 NOT NULL이다. 0초과의 값이다.

Reorder\_quantitys: 주문시 오는 재고 수량, int 정수 숫자이며 NOT NULL이다. 0초과의 값이다.

Recent\_order\_history: 최근 주문일자, DATE 이며 0000-00-00 식으로 년원일이 표시된다. NOT NULL이다.

Store\_id, product\_upc를 묶어서 primary key이다.

4. sales transactions:

Sale\_\_Id: sale 의 id값, int 정수 숫자이며 PIRMARY KEY이다.

Store\_id: 편의점의 id, int 정수 숫자이며 FOREIGN KEY이다. Stores 테이블에서 참조하고 stores에서 삭제된다면 여기서도 삭제된다.

Product\_upc: 물건의 id값, VARCHAR(50)인 문자열이며 foreign key이다. products 테이블에서 참조하고 products에서 삭제된다면 여기서도 삭제된다.

customer\_id: 고객의 id, int 정수 숫자이며 FOREIGN KEY이다. customers 테이블에서 참조하고 cusotmers에서 삭제된다면 여기서도 삭제된다.

payment\_method: 어떠한 방식으로 구매했는지, VARCHAR(50) 문자열이며 NOT NULL 이다. 현금과 신용카드로 나뉜다.

total\_amount: 판매한 물품의 개수, int 정수 숫자이며 NOT NULL이다. 0초과의 값이다.

Buy\_date: 물품을 구매한 날짜, date인 0000-00-00형식이며 NOT NULL이다.

Store\_id,product\_upc,customer\_id가 묶여서 primary key의 역할을 할 수 있지만 너무 많아서 따로 sale\_id라는 인덱스를 만들었다.

5. customer:

Customer\_id: 고객의 id, int 정수 숫자이며 PRIMARY KEY이다.

Name ,고객의 이름, VARCHAR(50) 문자열이며 NOT NULL 이다.

phone\_num : 고객의 핸드폰 번호, VARCHAR(50) 문자열이며 NOT NULL 이다.

Email : 고객의 이메일, VARCHAR(50) 문자열이며 NOT NULL 이다.

Loyalty: 고객의 충성도 등급, VARCHAR(50) 문자열이며 NOT NULL 이다. Vip와 일반으로 나뉜다.

6. vendor:

vendor\_id 공급업체의 id값, int 정수 숫자이며 PRIMARY KEY이다.

Name: 공급업체의 이름, VARCHAR(50) 문자열이며 NOT NULL 이다.

phone\_num: 공급업체의 전화번호, VARCHAR(50) 문자열이며 NOT NULL 이다.

Email: 공급업체의 이메일, VARCHAR(50) 문자열이며 NOT NULL 이다.

7. product\_vendor:

product\_upc: 물건의 id값, VARCHAR(50) 문자열이며 foreign key이다. products 테이블에서 참조하고 products에서 삭제된다면 여기서도 삭제된다.

vendor\_id: 공급업체의 id값 , int 정수 숫자이며 foreign key이다. vendors 테이블에서 참조하고 vendors에서 삭제된다면 여기서도 삭제된다.

product\_num: 물건의 공급 개수 , int 정수 숫자이며 NOT NULL이다. 0초과의 값이다.

Product\_upc, vendor\_id 묶여서 primary key이다.

3. Database Implementation

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

위와 같이 table을 create해서 sql파일에서 실행하여 table을 생성할 수 있다.

또한 적절한 데이터 샘플을 생성하여 데이터를 insert할 수 있다.

sql파일의 처음 부분에 drop table if exist를 추가하여 기존에 이러한 이름의 테이블이 존재한다면 삭제하는 쿼리를 처음에 넣어준다.

4. Application Development

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

Server, user, password, database를 입력하고 mysql\_init을 통해 초기화한 뒤 mysql\_real\_connect를 이용해 연결한다.

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

쿼리와 connection을 입력받는 함수를 만들어서 들어온 쿼리 요청에 대해 쿼리를 출력하는 함수를 만들었다.

그리고 반복문을 만들고 매번 메뉴화면을 출력하고 아래에 0~7중 값을 입력받고 각각의 값마다 쿼리를 설정해서 쿼리함수를 불러오는 반복문 코드를 만들었다. 예외처리로 0~7이외의 값이 온다면 다시 반복문의 시작으로 돌아가 메뉴화면을 다시 출력한다.

5. Query Implementation

텍스트, 스크린샷이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

UPC일때와 name,brand일때를 구분짓는다. 같이 했을 때 문자열 입력시 upc를 확인하는 where 부분에서 오류가 난다.   
먼저 재고 테이블과 매장 테이블을 join한 다음 product도 join해주면 product의 어떠한 정보를 얻으면 그 제품이 있는 매장의 id와 재고 수량을 알 수 있다.

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

매매 테이블에서 store\_id, product\_upc 그룹화해서 저번달의 판매량의 합을 구하고 각 매장마다 가장 많이 팔린 양을 max로 구하고 having을 이용해 매장마다 가장 많이 팔린 물건들을 구한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

올해 쿼터 즉 4월 5월 6월에 팔린 양을 store\_id로 그룹화하고 sum한다. 그 후 판매량을 기준으로 내림차순으로 정렬해주고 limit 1을 이용해 상위 하나의 store를 출력한다.

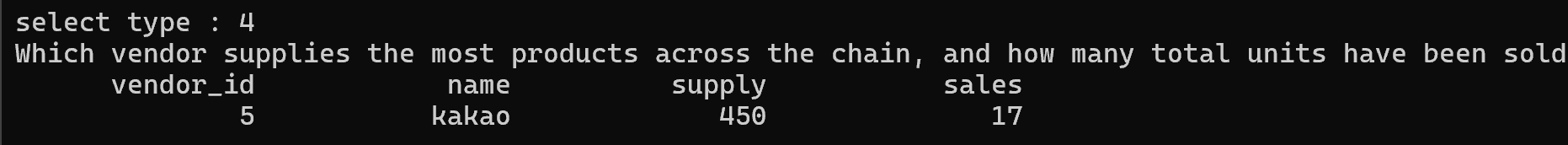
텍스트, 스크린샷, 폰트, 블랙이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

Product\_vendor 테이블에서 vendor\_id로 그룹화해서 총 공급량을 구하고 limit 1을 이용해 가장 공급량이 큰 vendor를 출력한다. 매매 테이블에서 판매량을 구하고 product와 join해서 x가격의 합을 구하여 가장 공급량이 큰 vendor의 판매량을 구한다.



텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

재고 테이블에서 inventory\_level < reorder\_threseholds를 이용하여 재고가 부족한 store와 물품의 정보를 출력한다.

텍스트, 스크린샷, 폰트, 블랙이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

Loyalty 등급이 VIP인 사람들의 정보를 알아낸뒤 매매 테이블과 product테이블을 join해 이 정보를 이용한 사람이 coffee를 결제했을때의 store\_id,payment\_method,buy\_date,customer\_id를 구해서 이와 같은 거래를 했을때의 총 합을 sum을 이용하여 구한다. 이때 물품이름이 coffee일 때를 빼야한다. 판매량을 기준으로 내림차순으로 정렬하고 LIMIT 3을 이용해 탑3 제품을 출력한다.

텍스트, 스크린샷, 폰트, 블랙이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

텍스트, 폰트, 스크린샷이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

재고 테이블에서 store\_id마다의 count(\*)을 구해서 다양성을 구한다. Store와 join을 한 후 ownership\_type마다의 평균 다양성을 구한다.텍스트, 스크린샷, 블랙, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.