Homework2 보고서

SQL & Relational Algebra

|  |  |
| --- | --- |
| **과목명** | 데이터베이스시스템및응용 |
| **담당 교수님** | 차재혁 교수 |
| **제출일** | 2021년 10월 15일(금요일) |
| **소속** | 한양대학교 공과대학 |
| 컴퓨터소프트웨어학부 |
| **학번** | **이름** |
| 2019009261 | 최가온(CHOI GA ON) |

Ⅰ. Tables

sellers.csv, stores.csv, menus.csv 데이터를 입력할 sellers, stores, menus 테이블을 만들기

* 테이블 간의 관계를 고려할 것
* 각 컬럼에 알맞다고 생각하는 데이터 타입과 제약을 사용할 것
  + Foreign Key에 대해 delete cascade 옵션을 꼭 사용할 것
* 참고) 이 예제 데이터들은 “배달의 한양”의 샘플 데이터인데, 이 스키마가 최선인 것은 아니며 나중의 과제는 재가공해서 쓸 필요가 있다.

**STEP 1**

「homework2」 데이터베이스를 만든다.

유저를 통해 만든 homework2 데이터베이스에 접속한다.

CREATE DATABASE homework2;

\c homework2

**STEP 2**

seller 테이블에서 만들 SEQUENCE 개체를 하나 생성한다.

CREATE SEQUENCE seq\_seller\_id INCREMENT 1 START 0 MINVALUE 0;

ALTER SEQUENCE seq\_seller\_id OWNER TO postgres;

seller.csv 파일로부터 seller 테이블을 정의한다.

CREATE TABLE sellers (

seller\_id INTEGER NOT NULL DEFAULT nextval (

'seq\_seller\_id'::regclass

),

name VARCHAR (20) NOT NULL,

phone VARCHAR (20) NOT NULL,

local VARCHAR (20) NOT NULL,

domain VARCHAR (20) NOT NULL,

passwd VARCHAR (20) NOT NULL,

PRIMARY KEY (seller\_id),

CONSTRAINT seller\_email UNIQUE (local, domain)

);

이후 seller.csv 파일로부터 sellers 테이블에 데이터를 추가한다.

COPY sellers (name, phone, local, domain, passwd)

FROM '/home/example\_db/sellers.csv' DELIMITER ',' CSV HEADER;

**STEP 3**

seller 테이블에서 만들 SEQUENCE 개체를 하나 생성한다.

CREATE SEQUENCE seq\_store\_id INCREMENT 1 START 0 MINVALUE 0;

ALTER SEQUENCE seq\_store\_id OWNER TO postgres;

stores.csv 파일로부터 stores 테이블을 정의한다.

CREATE TABLE stores (

store\_id INTEGER NOT NULL DEFAULT nextval (

'seq\_store\_id' :: regclass

),

address VARCHAR (100) NOT NULL,

sname VARCHAR (40) NOT NULL,

latitude FLOAT,

longitude FLOAT,

phone\_nums VARCHAR (50) NOT NULL,

seller\_id INTEGER,

PRIMARY KEY (store\_id),

CONSTRAINT fk\_seller FOREIGN KEY (seller\_id)

REFERENCES sellers (seller\_id) ON DELETE CASCADE

);

이후 store.csv 파일로부터 stores 테이블에 데이터를 추가한다.

COPY stores

(address, sname, latitude, longitude, phone\_nums, seller\_id)

FROM '/home/example\_db/stores.csv' DELIMITER ',' CSV HEADER;

**STEP 4**

menus.csv 파일로부터 menus 테이블을 정의한다.

CREATE TABLE menus (

menu VARCHAR (20) NOT NULL,

store\_id INTEGER NOT NULL,

CONSTRAINT fk\_store

FOREIGN KEY (store\_id) REFERENCES stores (store\_id)

ON DELETE CASCADE

);

**STEP 5**

아래의 그림은 sellers, stores, menus 테이블에 대한 E-R 다이어그램이다.

텍스트, 모니터, 노트북, 화면이(가) 표시된 사진

자동 생성된 설명

만약, 위의 설치를 제거하고 재설치할 경우 아래의 코드를 입력한다.

|  |
| --- |
| DROP TABLE menus;  DROP TABLE stores;  DROP TABLE num\_store;  DROP TABLE sellers;  DROP SEQUENCE seq\_seller\_id;  DROP SEQUENCE seq\_store\_id; |

Ⅱ. SQL & Relational Algebra

위 테이블을 이용해 다음 조건들을 SQL과 Relational Algebra로 나타내기

단, Relational Algebra는 밑줄 친 부분에 대해서만 나타내면 된다.

1. sellers 테이블의 이름과 전화번호를, 이름을 기준으로 내림차순으로 출력할 것

**SQL Sentence**

SELECT s.name, s.phone

FROM sellers s

ORDER BY s.name DESC;

텍스트이(가) 표시된 사진

자동 생성된 설명

**Relational Algebra**

1. sellers 테이블과 stores 테이블을 연결해 점주 이름이 ‘신동화’인 가게명을 모두 출력할 것

**SQL Sentence**

SELECT s2.sname

FROM sellers s1, stores s2

WHERE s2.seller\_id = s1.seller\_id AND s1.name = '신동화';

텍스트이(가) 표시된 사진

자동 생성된 설명

**Relational Algebra**

1. stores 테이블의 주소에서 특별광역시도와 시군구를 분리해 서울특별시 중구에 있는 가게명을 출력할 것

**SQL Sentence**

SELECT s.sname

FROM stores s

WHERE SPLIT\_PART (s.address, ' ', 1) = '서울특별시'

AND SPLIT\_PART (s.address, ' ', 2) = '중구';

텍스트이(가) 표시된 사진

자동 생성된 설명

**Relational Algebra**

1. sellers 테이블에 본인 정보 넣기(사실일 필요 없음)

**SQL Sentence**

INSERT INTO sellers (name, phone, local, domain, passwd)

VALUES ('최가온', '01036969592', 'choigaon1028', 'hanyang.ac.kr', 'dbdbdeep');

텍스트이(가) 표시된 사진

자동 생성된 설명

**Relational Algebra**

1. 판매자의 비밀번호 길이가 5 이하인 행을 찾아서 비밀번호를 6자리 이상의 임의의 영문자로 수정하기

**SQL Sentence**

**STEP 1**

비밀번호의 길이가 5 이하인 행을 sellers 테이블에서 찾는다.

SELECT \* FROM sellers

WHERE length( passwd ) <= 5;

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**STEP 2**

비밀번호의 길이가 5 이하인 행에 대해 영어 소문자/대문자 조합의 비밀번호로 passwd 내용을 바꾼다. 이때 비밀번호의 길이는 6 이상 10 이하의 임의의 길이를 갖는다.

UPDATE sellers

SET passwd = (

SELECT array\_to\_string (ARRAY (

SELECT chr ((

65 + 32\*round(random()) + round(random()\*25)

)::INTEGER)

FROM generate\_series (

1, CAST(6 + floor(random()\*5) AS INT)

)), ''

)

)

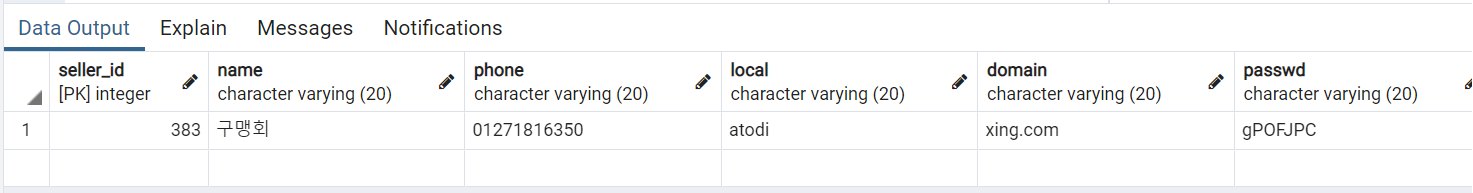
WHERE LENGTH(passwd) <= 5;

**STEP 3**

위의 STEP 2를 실행하기 전 passwd의 길이가 5 이하였던 행을 재검색하여 passwd를 확인한다. 비밀번호가 6자리 이상으로 새롭게 변경된 것을 확인할 수 있다.

SELECT \* FROM sellers

WHERE name = '구맹회';



**Relational Algebra**

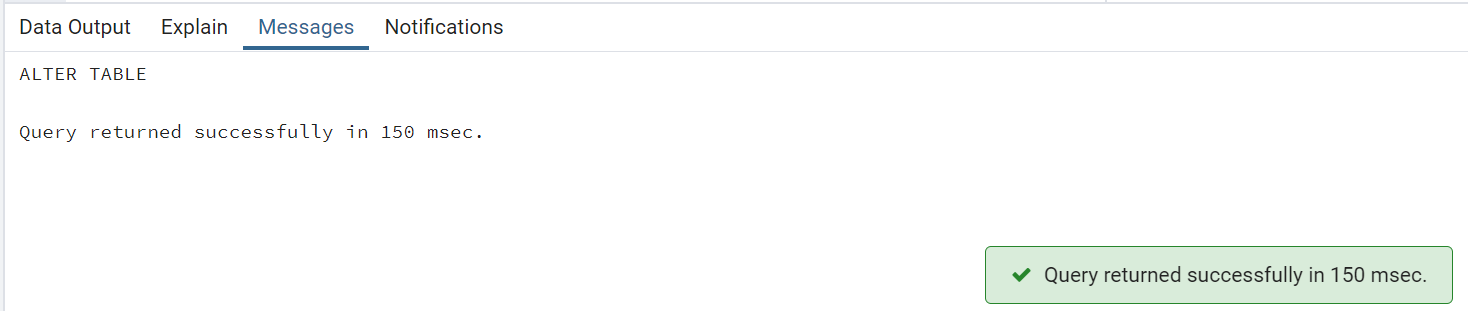
1. stores 테이블에 ‘rating’이라는 column을 추가하고, 0 이상 5 이하의 랜덤한 정수로 채워 넣기

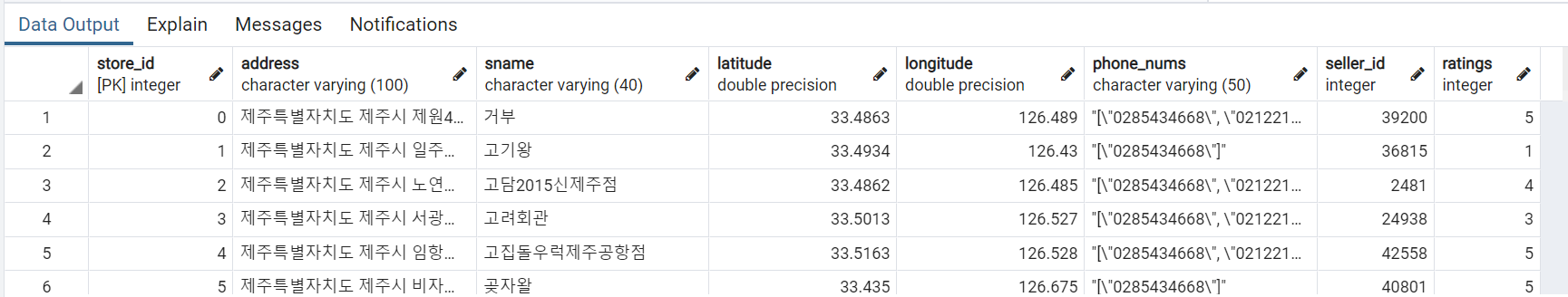
**SQL Sentence**

ALTER TABLE stores

ADD COLUMN ratings INT

DEFAULT floor( random () \* 6 );





1. 필요한 테이블들을 연결해 판매자 이름, 판매자 id, 소유한 가게 수를 내림차순으로 출력하는 select query를 만들고, 그 결과를 새로운 테이블(num\_store) 만들기

**SQL Sentence**

CREATE TABLE num\_store AS (

SELECT s1.name, s1.seller\_id, COUNT (\*) AS store\_num

FROM sellers s1, stores s2

WHERE s1.seller\_id = s2.seller\_id

GROUP BY s1.seller\_id

ORDER BY store\_num DESC

);

텍스트이(가) 표시된 사진

자동 생성된 설명

**Relational Algebra**

1. num\_store 테이블에서 가장 상위에 있는 판매자 아이디를 참고해, 판매자 테이블에서 해당 판매자를 삭제하기. 그를 참조하는 다른 테이블까지 cascading 되어야 한다.

**SQL Sentence**

**STEP 1**

num\_store 테이블에서 store\_num(소유 가게 수)의 최댓값을 찾는다.

SELECT MAX (store\_num) FROM num\_store;

**STEP 2**

num\_store 테이블에서 소유 가게수가 최댓값에 해당하는 점주의 seller\_id를 추출한다.

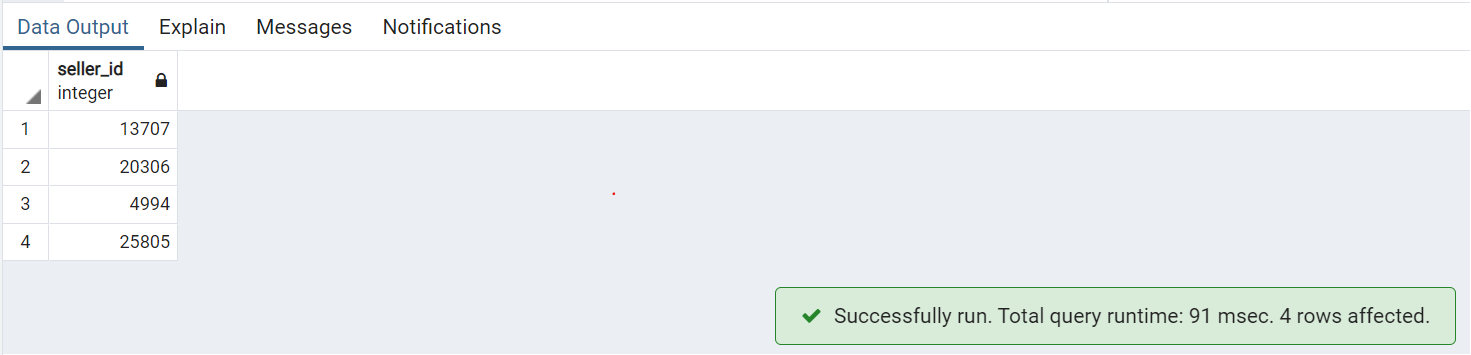
SELECT seller\_id FROM num\_store

WHERE store\_num = (

SELECT MAX(store\_num)

FROM num\_store

);



**STEP 3**

STEP 2에서 추출한 seller\_id에 해당하는 점주에 대한 모든 column을 sellers 테이블에서 가져온다.

SELECT \* FROM sellers s

WHERE s.seller\_id IN (

SELECT seller\_id

FROM num\_store

WHERE store\_num = (

SELECT MAX (store\_num)

FROM num\_store

)

);

텍스트이(가) 표시된 사진

자동 생성된 설명

**(최종정답코드)**

**STEP 4**

DELETE FROM sellers

WHERE seller\_id IN (

SELECT seller\_id

FROM num\_store

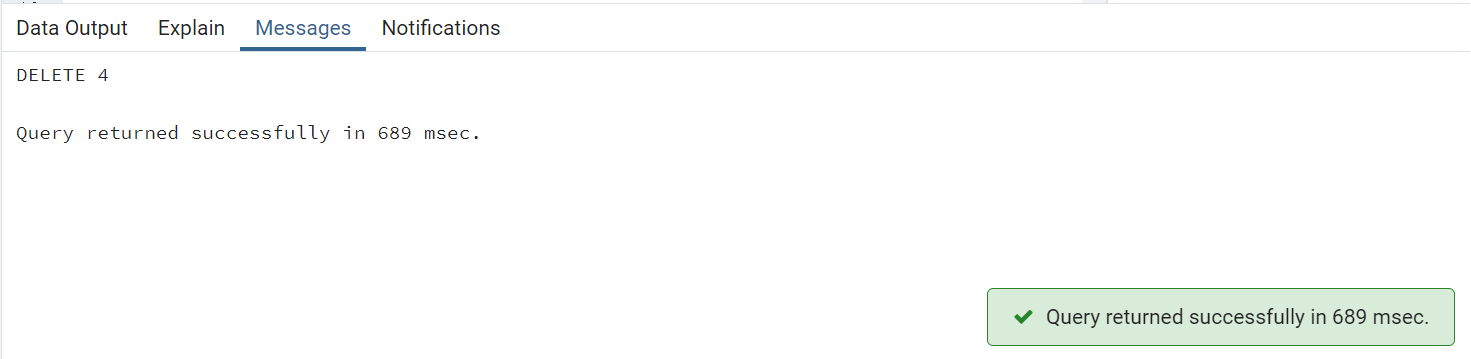
WHERE store\_num = (

SELECT MAX (store\_num)

FROM num\_store

)

)



**STEP 5**

이제 제대로 sellers 테이블에서 해당 점주의 정보가 지워졌는지 확인한다.

SELECT \*

FROM sellers

WHERE seller\_id IN ( 4994, 13707, 20306, 25805 );

텍스트이(가) 표시된 사진

자동 생성된 설명

**STEP 6**

해당 점주가 소요한 가게의 정보가 stores 테이블에서 지워졌는지 확인한다.

SELECT \* FROM stores

WHERE seller\_id IN (4994, 13707, 20306, 25805);

텍스트이(가) 표시된 사진

자동 생성된 설명

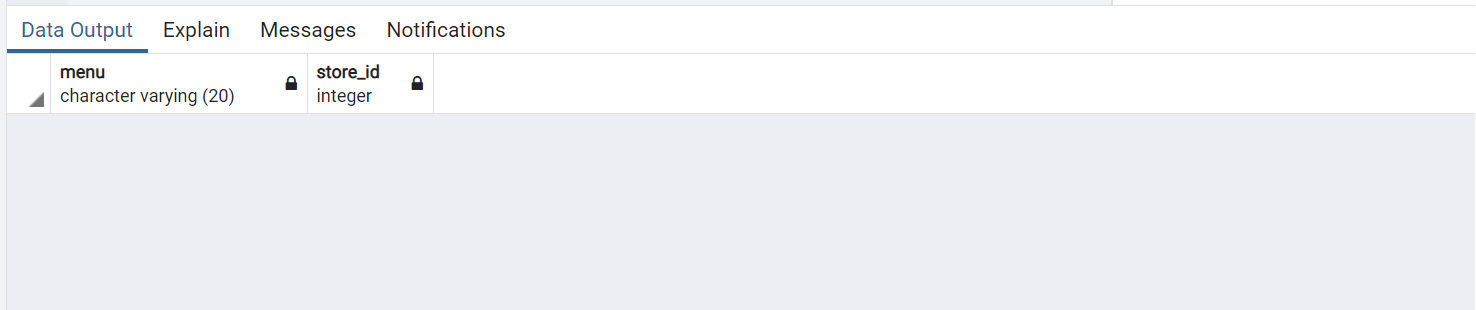
**STEP 7**

해당 점주가 소요한 가게에서 제공하는 메뉴에 대한 정보가 모두 지워졌는지 확인한다.

SELECT \* FROM menus

WHERE store\_id IN (3458, 4560, 5465, 6147, 8353, 10476, 14510, 15883, 17661, 19675, 22279, 22313, 23759, 25692, 27816, 29117, 29551, 30565, 31989, 33218, 34209, 34770, 35714, 37308

);



**Relational Algebra**

위의 SQL문은 nested query 형태를 갖는다.

The most-inner query is:

The second query is:

The third query is:

The final query is(최종정답):

(1)

(2)