

데이터 베이스 기초 실습 및 과제 #6

> 2020.04.02. 목 B반 송명훈

- 1. [실습 #1] 가계부 테이블 구성
- 2. [실습 #2] 전화번호부의 즐겨찿기 테이블 구성
- 3. [실습 #3] 물품 가격 정리 테이블 구축
- 4. [복습과제] 회원 관리 테이블 구축



[실습 #1] 가계부 작성

테이블 구성

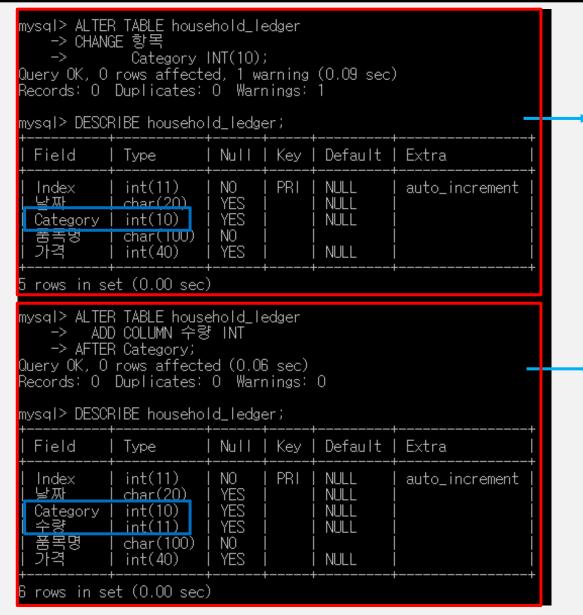
실습 내용

- 가계부 작성을 위한 테이블 구성하기
 - 1. No(int), 날짜(char), 항목(char), 품목명(char), 가격(int)으로 column 5개
 - 2. 아래 조건대로 출력하기
 - 1. 현재 상태 그대로 테이블 구조 출력
 - 2. 'No' 칼럼을 'Index'로 변경하고 테이블 구조 출력
 - 3. '항목(char)' column 'Category(int)'로 변경하고 테이블 구조 출력
 - 4. 'Category' column 뒤에 '수량(int)'를 넣고 테이블 구조 출력
 - 5. '가격' column 뒤에 '비고(char)'를 넣고 테이블 구조 출력
 - 6. 가계부 테이블 이름을 'mymoney'로 변경하고 테이블 구조 출력

```
TABLE household_ledger (
AUTO_INCREMENT PRIMARY KEY,
         항목 CHAR(30) DEFAULT ' ' NOT NULL,
품목명 CHAR(100) DEFAULT ' ' <u>NOT NULL</u>,
        가격 INT(40)
Query OK, O rows affected, 1 warning (0.06 sec)
mysgl> DESCRIBE household ledger;
  Field | Type
                        | Null | Key | Default | Extra
            int(11)
                          NO
                                  PRI | NULL
                                                     auto increment
            har (20)
                         YES
                                        NULL
                          NO
            chai (100)
                          YES
                                         NULL
  rows in set (0.01 sec)
 nvsal> ALTER TABLE household ledger
    -> RENAME COLUMN No TO `Index`;
Query OK, O rows a<mark>ffected (0.02 sec)</mark>
Records: O Duplidates: O Warnings: O
mysg∣> DESCRIBE ∦ousehold_ledger;
  Field | Type
                        | Null | Key | Default | Extra
                                  PRI | NULL
                                                     auto increment
                         YES
            char (20)
                                         NULL
            char (30)
            char (100)
                          NO
                                        NULL
 rows in set (0.00 sec)
```

- CREATE TABLE household_ledger
 - ✓ household_ledger TABLE 생성
 - ✓ FIELD: No, 날씨, 항목, 품목명, 가격

- ALTER TABLE household_ledger
 RENAME COLUMN No TO `Index`;
 - ✓ No 컬럼의 이름을 Index로 변경
 - ➤ Index가 예약어이므로 `(Apostrophe)를 사용
- DESCRIBE household_ledger;
 - ✔ No가 Index로 변경되었는지 확인



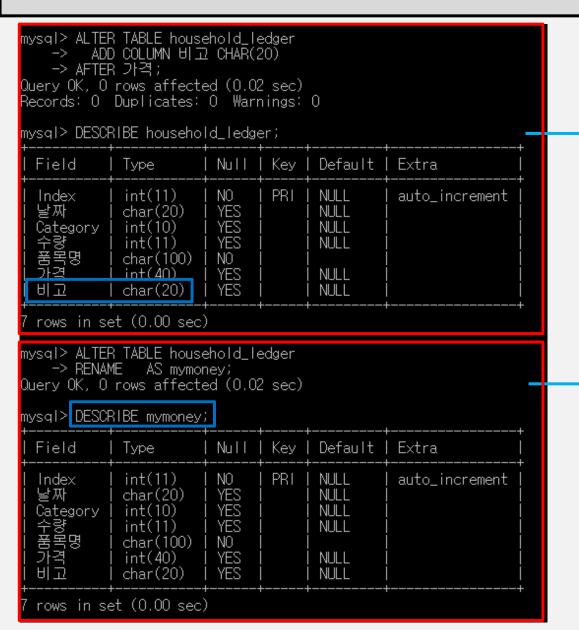
ALTER TABLE household_ledger
CHANGE 항목
Category INT(10);

✓ 항목 컬럼의 이름과 data type을 변경
✓ 변경 전: 항목 CHAR(30)

- DESCRIBE household_ledger;
 - ✓ 명령이 의도한대로 적용되었는지 확인
- ALTER TABLE household_ledger ADD COLUMN 수량 INT AFTER Category;
 - ✓ 수량 column을 Category column 뒤에 추가

변경후: Category INT(10)

- DESCRIBE household_ledger;
 - ✓ 수량 column이 의도한대로 추가되었는지 확인



- ALTER TABLE household_ledger
 ADD COLUMN 비고 CHAR(20)

 AFTER 가격;
 - ✓ 비고 column을 가격 column의 뒤에 추가
- DESCRIBE household_ledger;
 - ✓ 명령이 의도한대로 적용되었는지 확인

- ALTER TABLE household_ledger RENAME AS mymoney;
 - ✓ TABLE의 이름을 household_ledger에서 mymoney로 변경
- DESCRIBE mymoney;
 - ✓ TABLE 이름이 의도한대로 변경되었는지 확인



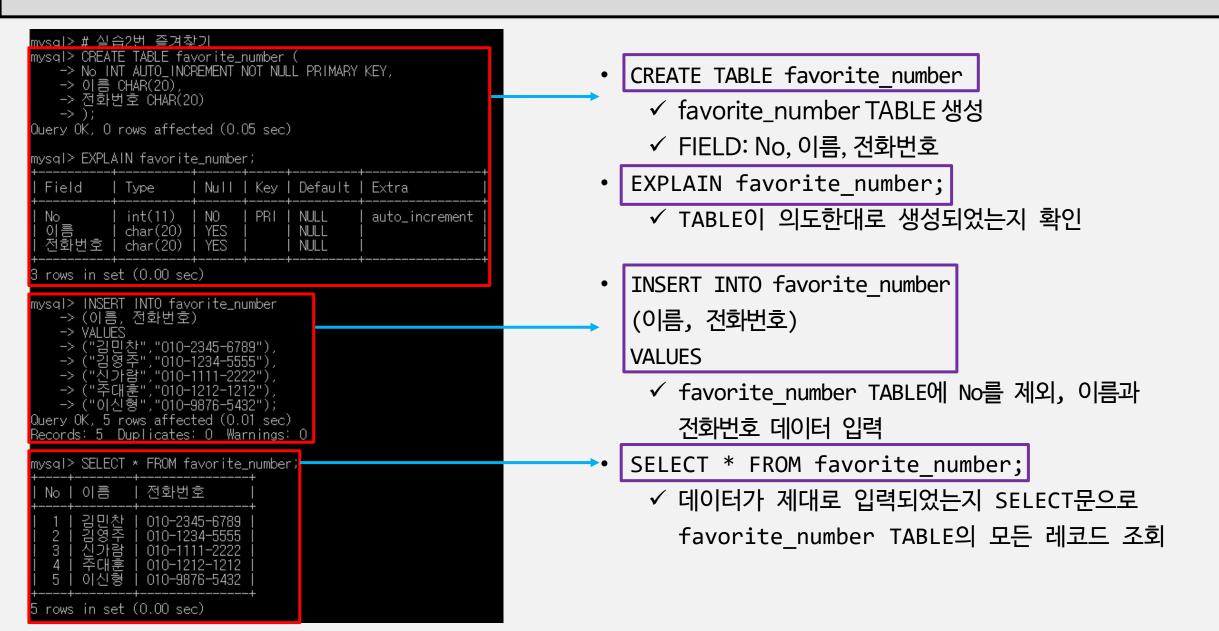
[실습 #2] 전화번호부의 즐겨찾기 테이블 구성하기

[실습 #2] 전화번호부의 즐겨찿기 테이블 구성

실습 내용

- 전화번호부의 즐겨찾기 테이블 구성하기
 - 1. No(AUTO_INCREMENT, NOT NULL, PRIMARY KEY), 이름, 전화번호 필드 생성
 - 2. 아래 조건대로 출력하기
 - ① 현재 상태의 테이블 구조 출력
 - ② 'No'를 제외한 이름과 전화번호만 입력하여 데이터 5개 입력 후 내용 전체 출력

[실습 #2] 전화번호부의 즐겨찿기 테이블 구성

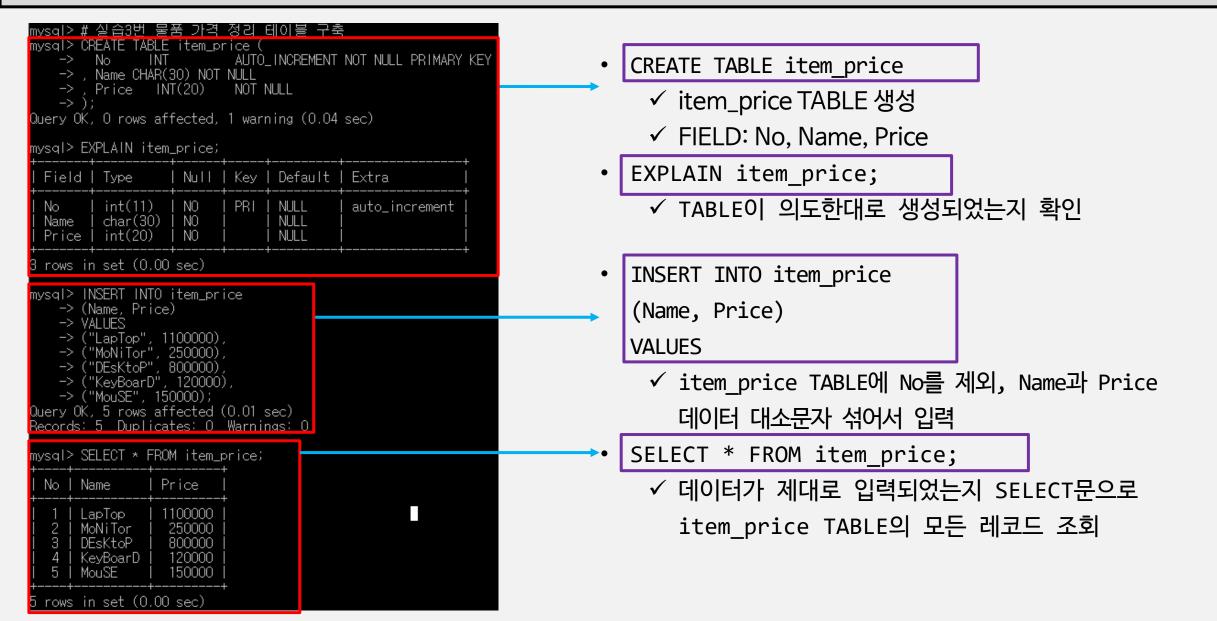


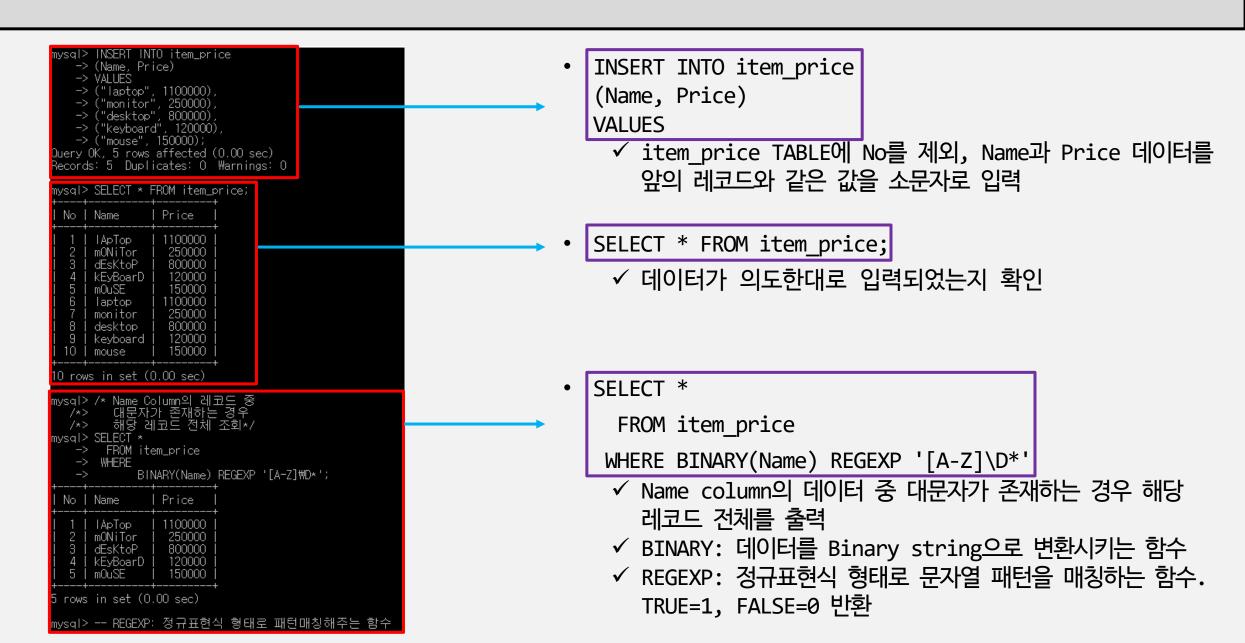


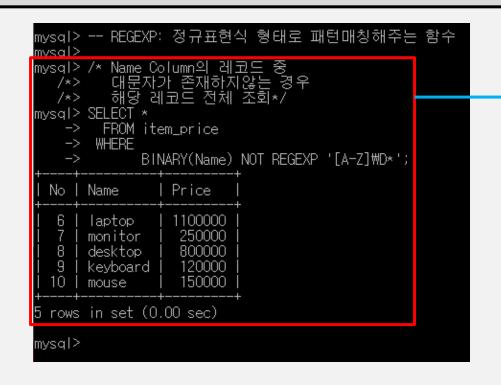
[실습 #3] 물품 가격 정리 테이블 구축하기

실습 내용

- 물품 가격 정리 테이블 구축하기
 - 1. No(AUTO_INCREMENT, NOT NULL, PRIMARY KEY), 물품명, 가격 필드 생성
 - 2. 데이터를 아래와 같이 입력하기
 - ① 대소문자를 섞은 물품명을 활용하여 데이터 5개 입력
 - ② 1번에서 입력한 똑같은 물품명을 소문자로만 하여 데이터 5개 입력
 - 3. 아래 조건대로 출력하기
 - ① 대소문자 구분없이 전체 목록 출력하기
 - ② 대소문자 구분하여 데이터 출력해보기(5개)







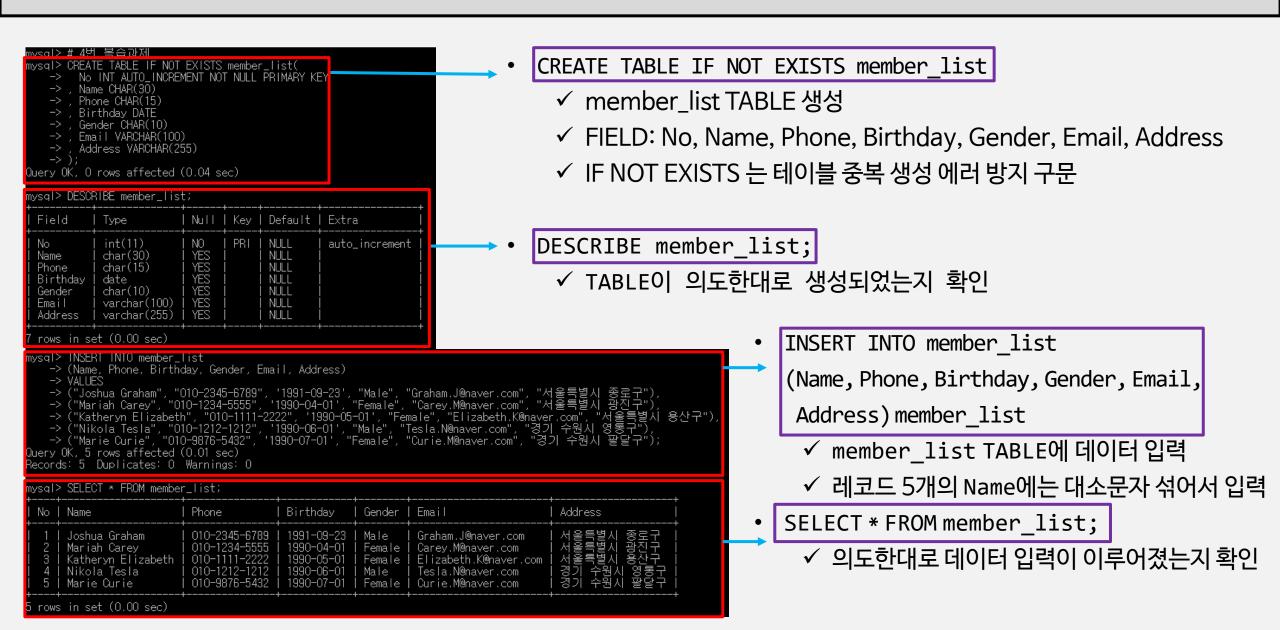
SELECT *
FROM item_price
WHERE BINARY(Name) REGEXP '[A-Z]\D*'

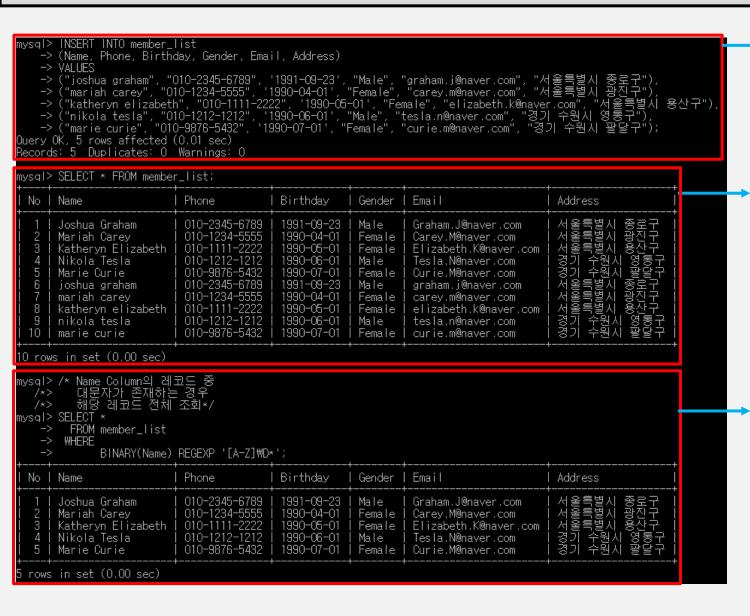
- ✓ Name column의데이터중대문자가존재하지않는경우해당 레코드전체를출력
 - ※ 정규표현식을 사용하지 않고 해결할 수 있는 방법이 있으면 꼭 좀 알려주시면 감사하겠습니다 교수님...ㅠㅠ



과제 내용

- 회원관리 테이블 구축하기
 - 1. No(AUTO_INCREMENT, NOT NULL, PRIMARY KEY), 이름, 전화번호, 생년월일 등
 - 2. 컬럼 최소 7개 생성
 - 3. 데이터는 아래와 같이 입력하기
 - ① 대소문자를 섞은 이름으로 5개 입력
 - ② 1번에서 입력한 이름을 소문자로만 5개 입력
 - 4. 아래 조건대로 출력하기
 - ① 대소문자 구분없이 전체 목록 출력하기
 - ② 대소문자 구분하여 데이터 출력해보기(5개)
 - ③ 맨 마지막에 'etc' 컬럼을 생성하여 테이블 구조 출력





INSERT INTO member_list
(Name, Phone, Birthday, Gender, Email,
Address) member_list

✓ 레코드 5개의 Name에 소문자만 쓰인 레코드 5개 추가입력

SELECT * FROM member_list;

- ✓ member_list 내 대소문자 구분없이 모든 데이터 조회
- ✓ TABLE이 의도한대로 생성되었는지 확인

SELECT *
FROM item_price
WHERE BINARY(Name) REGEXP '[A-Z]\D*';

✓ Name column의 데이터 중 대문자가 존재하는 경우 해당 레코드 전체를 출력

