



데이터 베이스 기초 실습 및 과제 #6

2020.04.02. 목
B반 송명훈

목 차

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



[실습 #1]
가계부 작성
테이블 구성

[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

[실습 #1] 가계부 작성 테이블 구성

```
mysql> # 실습1번 가계부
mysql> CREATE TABLE household_ledger (
  -> No INT AUTO_INCREMENT PRIMARY KEY,
  -> 날짜 CHAR(20),
  -> 항목 CHAR(30) DEFAULT '' NOT NULL,
  -> 품목명 CHAR(100) DEFAULT '' NOT NULL,
  -> 가격 INT(40)
  -> );
Query OK, 0 rows affected, 1 warning (0.06 sec)
```

```
mysql> DESCRIBE household_ledger;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| No    | int(11)| NO   | PRI | NULL    | auto_increment |
| 날짜  | char(20)| YES  |     | NULL    |               |
| 항목  | char(30)| NO   |     |         |               |
| 품목명| char(100)| NO  |     |         |               |
| 가격  | int(40)| YES  |     | NULL    |               |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

```
mysql> ALTER TABLE household_ledger
  -> RENAME COLUMN No TO `Index`;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> DESCRIBE household_ledger;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Index | int(11)| NO   | PRI | NULL    | auto_increment |
| 날짜  | char(20)| YES  |     | NULL    |               |
| 항목  | char(30)| NO   |     |         |               |
| 품목명| char(100)| NO  |     |         |               |
| 가격  | int(40)| YES  |     | NULL    |               |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

- CREATE TABLE household_ledger
 - ✓ household_ledger TABLE 생성
 - ✓ FIELD: No, 날짜, 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

[실습 #1] 가계부 작성 테이블 구성

```
mysql> ALTER TABLE household_ledger
-> CHANGE 항목
-> Category INT(10);
Query OK, 0 rows affected, 1 warning (0.09 sec)
Records: 0 Duplicates: 0 Warnings: 1
```

```
mysql> DESCRIBE household_ledger;
```

Field	Type	Null	Key	Default	Extra
Index	int(11)	NO	PRI	NULL	auto_increment
날짜	char(20)	YES		NULL	
Category	int(10)	YES		NULL	
품목명	char(100)	NO			
가격	int(40)	YES		NULL	

5 rows in set (0.00 sec)

```
mysql> ALTER TABLE household_ledger
-> ADD COLUMN 수량 INT
-> AFTER Category;
Query OK, 0 rows affected (0.06 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> DESCRIBE household_ledger;
```

Field	Type	Null	Key	Default	Extra
Index	int(11)	NO	PRI	NULL	auto_increment
날짜	char(20)	YES		NULL	
Category	int(10)	YES		NULL	
수량	int(11)	YES		NULL	
품목명	char(100)	NO			
가격	int(40)	YES		NULL	

6 rows in set (0.00 sec)

- ALTER TABLE household_ledger
CHANGE 항목

Category INT(10);

✓ 항목 컬럼의 이름과 data type을 변경

✓ 변경 전: 항목 CHAR(30)

변경 후: Category INT(10)

- DESCRIBE household_ledger;

✓ 명령이 의도한대로 적용되었는지 확인

- ALTER TABLE household_ledger
ADD COLUMN 수량 INT AFTER Category;

✓ 수량 column을 Category column 뒤에 추가

- DESCRIBE household_ledger;

✓ 수량 column이 의도한대로 추가되었는지 확인

[실습 #1] 가계부 작성 테이블 구성

```
mysql> ALTER TABLE household_ledger
-> ADD COLUMN 비교 CHAR(20)
-> AFTER 가격;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> DESCRIBE household_ledger;
```

Field	Type	Null	Key	Default	Extra
Index	int(11)	NO	PRI	NULL	auto_increment
날짜	char(20)	YES		NULL	
Category	int(10)	YES		NULL	
수량	int(11)	YES		NULL	
품목명	char(100)	NO			
가격	int(40)	YES		NULL	
비교	char(20)	YES		NULL	

7 rows in set (0.00 sec)

```
mysql> ALTER TABLE household_ledger
-> RENAME AS mymoney;
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> DESCRIBE mymoney;
```

Field	Type	Null	Key	Default	Extra
Index	int(11)	NO	PRI	NULL	auto_increment
날짜	char(20)	YES		NULL	
Category	int(10)	YES		NULL	
수량	int(11)	YES		NULL	
품목명	char(100)	NO			
가격	int(40)	YES		NULL	
비교	char(20)	YES		NULL	

7 rows in set (0.00 sec)

- ALTER TABLE household_ledger
ADD COLUMN 비교 CHAR(20)
AFTER 가격;
✓ 비교 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
즐거찾기 테이블
구성하기

[실습 #2] 전화번호부의 즐겨찾기 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 즐겨찾기 테이블 구성

```
mysql> # 실습2번 즐겨찾기
mysql> CREATE TABLE favorite_number (
  -> No INT AUTO_INCREMENT NOT NULL PRIMARY KEY,
  -> 이름 CHAR(20),
  -> 전화번호 CHAR(20)
  -> );
Query OK, 0 rows affected (0.05 sec)

mysql> EXPLAIN favorite_number;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| No     | int(11) | NO   | PRI | NULL    | auto_increment |
| 이름   | char(20) | YES  |     | NULL    |                |
| 전화번호 | char(20) | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> INSERT INTO favorite_number
  -> (이름, 전화번호)
  -> VALUES
  -> ("김민찬", "010-2345-6789"),
  -> ("김영주", "010-1234-5555"),
  -> ("신가람", "010-1111-2222"),
  -> ("주대훈", "010-1212-1212"),
  -> ("이신형", "010-9876-5432");
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

```
mysql> SELECT * FROM favorite_number;
+-----+-----+-----+
| No | 이름 | 전화번호 |
+-----+-----+-----+
| 1  | 김민찬 | 010-2345-6789 |
| 2  | 김영주 | 010-1234-5555 |
| 3  | 신가람 | 010-1111-2222 |
| 4  | 주대훈 | 010-1212-1212 |
| 5  | 이신형 | 010-9876-5432 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

- CREATE TABLE favorite_number
 - ✓ favorite_number TABLE 생성
 - ✓ FIELD: No, 이름, 전화번호
- EXPLAIN favorite_number;
 - ✓ TABLE이 의도한대로 생성되었는지 확인
- INSERT INTO favorite_number (이름, 전화번호) VALUES
 - ✓ favorite_number TABLE에 No를 제외, 이름과 전화번호 데이터 입력
- SELECT * FROM favorite_number;
 - ✓ 데이터가 제대로 입력되었는지 SELECT문으로 favorite_number TABLE의 모든 레코드 조회



[실습 #3]
물품 가격
정리 테이블
구축하기

[실습 #3] 물품 가격 정리 테이블 구축

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)

[실습 #3] 물품 가격 정리 테이블 구축

```
mysql> # 실습3번 물품 가격 정리 테이블 구축
mysql> CREATE TABLE item_price (
  -> No INT AUTO_INCREMENT NOT NULL PRIMARY KEY
  -> , Name CHAR(30) NOT NULL
  -> , Price INT(20) NOT NULL
  -> );
Query OK, 0 rows affected, 1 warning (0.04 sec)

mysql> EXPLAIN item_price;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| No    | int(11)| NO   | PRI | NULL    | auto_increment |
| Name  | char(30)| NO   |     | NULL    |                |
| Price | int(20)| NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> INSERT INTO item_price
  -> (Name, Price)
  -> VALUES
  -> ("LapTop", 1100000),
  -> ("MoNiTor", 250000),
  -> ("DESkToP", 800000),
  -> ("KeyBoarD", 120000),
  -> ("MouSE", 150000);
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

```
mysql> SELECT * FROM item_price;
+-----+-----+-----+
| No | Name   | Price |
+-----+-----+-----+
| 1  | LapTop | 1100000 |
| 2  | MoNiTor | 250000 |
| 3  | DESkToP | 800000 |
| 4  | KeyBoarD | 120000 |
| 5  | MouSE | 150000 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

- CREATE TABLE item_price
 - ✓ item_price TABLE 생성
 - ✓ FIELD: No, Name, Price
- EXPLAIN item_price;
 - ✓ TABLE이 의도한대로 생성되었는지 확인
- INSERT INTO item_price (Name, Price) VALUES
 - ✓ item_price TABLE에 No를 제외, Name과 Price 데이터 대소문자 섞어서 입력
- SELECT * FROM item_price;
 - ✓ 데이터가 제대로 입력되었는지 SELECT문으로 item_price TABLE의 모든 레코드 조회

[실습 #3] 물품 가격 정리 테이블 구축

```
mysql> INSERT INTO item_price
-> (Name, Price)
-> VALUES
-> ("laptop", 1100000),
-> ("monitor", 250000),
-> ("desktop", 800000),
-> ("keyboard", 120000),
-> ("mouse", 150000);
Query OK, 5 rows affected (0.00 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

```
mysql> SELECT * FROM item_price;
+----+-----+-----+
| No | Name  | Price |
+----+-----+-----+
| 1  | lApTo | 1100000 |
| 2  | mONiTo | 250000 |
| 3  | dEsKto | 800000 |
| 4  | kEyBoarD | 120000 |
| 5  | mOuSE | 150000 |
| 6  | laptop | 1100000 |
| 7  | monitor | 250000 |
| 8  | desktop | 800000 |
| 9  | keyboard | 120000 |
| 10 | mouse | 150000 |
+----+-----+-----+
10 rows in set (0.00 sec)
```

```
mysql> /* Name Column의 레코드 중
/*> 대문자가 존재하는 경우
/*> 해당 레코드 전체 조회*/
mysql> SELECT *
-> FROM item_price
-> WHERE
-> BINARY(Name) REGEXP '[A-Z]\D*';
+----+-----+-----+
| No | Name  | Price |
+----+-----+-----+
| 1  | lApTo | 1100000 |
| 2  | mONiTo | 250000 |
| 3  | dEsKto | 800000 |
| 4  | kEyBoarD | 120000 |
| 5  | mOuSE | 150000 |
+----+-----+-----+
5 rows in set (0.00 sec)

mysql> -- REGEXP: 정규표현식 형태로 패턴매칭해주는 함수
```

- INSERT INTO item_price
(Name, Price)
VALUES

- ✓ item_price TABLE에 No를 제외, Name과 Price 데이터를 앞의 레코드와 같은 값을 소문자로 입력

- SELECT * FROM item_price;

- ✓ 데이터가 의도한대로 입력되었는지 확인

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

- ✓ Name column의 데이터 중 대문자가 존재하는 경우 해당 레코드 전체를 출력
- ✓ BINARY: 데이터를 Binary string으로 변환시키는 함수
- ✓ REGEXP: 정규표현식 형태로 문자열 패턴을 매칭하는 함수. TRUE=1, FALSE=0 반환

[실습 #3] 물품 가격 정리 테이블 구축

```
mysql> -- REGEXP: 정규표현식 형태로 패턴매칭해주는 함수
mysql>
mysql> /* Name Column의 레코드 중
/*> 대문자가 존재하지 않는 경우
/*> 해당 레코드 전체 조회*/
mysql> SELECT *
-> FROM item_price
-> WHERE
-> BINARY(Name) NOT REGEXP '[A-Z]\D*';

+----+-----+-----+
| No | Name  | Price |
+----+-----+-----+
| 6  | laptop | 1100000 |
| 7  | monitor | 250000 |
| 8  | desktop | 800000 |
| 9  | keyboard | 120000 |
| 10 | mouse  | 150000 |
+----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

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

✓ Name column의 데이터 중 대문자가 존재하지 않는 경우 해당 레코드 전체를 출력

※ 정규표현식을 사용하지 않고 해결할 수 있는 방법이 있으면 꼭 좀 알려주시면 감사하겠습니다 교수님...ㅠㅠ



**[복습 과제]**  
**회원 관리**  
**테이블 구축**



# [복습과제] 회원 관리 테이블 구축

## 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

# [복습과제] 회원 관리 테이블 구축

```
mysql> # 4번 복습과제
mysql> CREATE TABLE IF NOT EXISTS member_list(
-> No INT AUTO_INCREMENT NOT NULL PRIMARY KEY
-> , Name CHAR(30)
-> , Phone CHAR(15)
-> , Birthday DATE
-> , Gender CHAR(10)
-> , Email VARCHAR(100)
-> , Address VARCHAR(255)
->);
Query OK, 0 rows affected (0.04 sec)
```

• CREATE TABLE IF NOT EXISTS member\_list

- ✓ member\_list TABLE 생성
- ✓ FIELD: No, Name, Phone, Birthday, Gender, Email, Address
- ✓ IF NOT EXISTS 는 테이블 중복 생성 에러 방지 구문

```
mysql> DESCRIBE member_list;
```

| Field    | Type         | Null | Key | Default | Extra          |
|----------|--------------|------|-----|---------|----------------|
| No       | int(11)      | NO   | PRI | NULL    | auto_increment |
| Name     | char(30)     | YES  |     | NULL    |                |
| Phone    | char(15)     | YES  |     | NULL    |                |
| Birthday | date         | YES  |     | NULL    |                |
| Gender   | char(10)     | YES  |     | NULL    |                |
| Email    | varchar(100) | YES  |     | NULL    |                |
| Address  | varchar(255) | YES  |     | NULL    |                |

• DESCRIBE member\_list;

- ✓ TABLE이 의도한대로 생성되었는지 확인

```
mysql> INSERT INTO member_list
```

```
-> (Name, Phone, Birthday, Gender, Email, Address)
-> VALUES
-> ("Joshua Graham", "010-2345-6789", '1991-09-23', "Male", "Graham.J@naver.com", "서울특별시 종로구"),
-> ("Mariah Carey", "010-1234-5555", '1990-04-01', "Female", "Carey.M@naver.com", "서울특별시 광진구"),
-> ("Katheryn Elizabeth", "010-1111-2222", '1990-05-01', "Female", "Elizabeth.K@naver.com", "서울특별시 용산구"),
-> ("Nikola Tesla", "010-1212-1212", '1990-06-01', "Male", "Tesla.N@naver.com", "경기 수원시 영통구"),
-> ("Marie Curie", "010-9876-5432", '1990-07-01', "Female", "Curie.M@naver.com", "경기 수원시 팔달구");
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

• INSERT INTO member\_list

(Name, Phone, Birthday, Gender, Email, Address) member\_list

- ✓ member\_list TABLE에 데이터 입력
- ✓ 레코드 5개의 Name에는 대소문자 섞어서 입력

```
mysql> SELECT * FROM member_list;
```

| No | Name               | Phone         | Birthday   | Gender | Email                 | Address    |
|----|--------------------|---------------|------------|--------|-----------------------|------------|
| 1  | Joshua Graham      | 010-2345-6789 | 1991-09-23 | Male   | Graham.J@naver.com    | 서울특별시 종로구  |
| 2  | Mariah Carey       | 010-1234-5555 | 1990-04-01 | Female | Carey.M@naver.com     | 서울특별시 광진구  |
| 3  | Katheryn Elizabeth | 010-1111-2222 | 1990-05-01 | Female | Elizabeth.K@naver.com | 서울특별시 용산구  |
| 4  | Nikola Tesla       | 010-1212-1212 | 1990-06-01 | Male   | Tesla.N@naver.com     | 경기 수원시 영통구 |
| 5  | Marie Curie        | 010-9876-5432 | 1990-07-01 | Female | Curie.M@naver.com     | 경기 수원시 팔달구 |

• SELECT \* FROM member\_list;

- ✓ 의도한대로 데이터 입력이 이루어졌는지 확인

```
5 rows in set (0.00 sec)
```

# [복습과제] 회원 관리 테이블 구축

```
mysql> INSERT INTO member_list
-> (Name, Phone, Birthday, Gender, Email, Address)
-> VALUES
-> ("joshua graham", "010-2345-6789", '1991-09-23', "Male", "graham.j@naver.com", "서울특별시 종로구"),
-> ("mariah carey", "010-1234-5555", '1990-04-01', "Female", "carey.m@naver.com", "서울특별시 광진구"),
-> ("katheryn elizabeth", "010-1111-2222", '1990-05-01', "Female", "elizabeth.k@naver.com", "서울특별시 용산구"),
-> ("nikola tesla", "010-1212-1212", '1990-06-01', "Male", "tesla.n@naver.com", "경기 수원시 영통구"),
-> ("marie curie", "010-9876-5432", '1990-07-01', "Female", "curie.m@naver.com", "경기 수원시 팔달구");
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

```
mysql> SELECT * FROM member_list;
```

| No | Name               | Phone         | Birthday   | Gender | Email                 | Address    |
|----|--------------------|---------------|------------|--------|-----------------------|------------|
| 1  | Joshua Graham      | 010-2345-6789 | 1991-09-23 | Male   | Graham.J@naver.com    | 서울특별시 종로구  |
| 2  | Mariah Carey       | 010-1234-5555 | 1990-04-01 | Female | Carey.M@naver.com     | 서울특별시 광진구  |
| 3  | Katheryn Elizabeth | 010-1111-2222 | 1990-05-01 | Female | Elizabeth.K@naver.com | 서울특별시 용산구  |
| 4  | Nikola Tesla       | 010-1212-1212 | 1990-06-01 | Male   | Tesla.N@naver.com     | 경기 수원시 영통구 |
| 5  | Marie Curie        | 010-9876-5432 | 1990-07-01 | Female | Curie.M@naver.com     | 경기 수원시 팔달구 |
| 6  | joshua graham      | 010-2345-6789 | 1991-09-23 | Male   | graham.j@naver.com    | 서울특별시 종로구  |
| 7  | mariah carey       | 010-1234-5555 | 1990-04-01 | Female | carey.m@naver.com     | 서울특별시 광진구  |
| 8  | katheryn elizabeth | 010-1111-2222 | 1990-05-01 | Female | elizabeth.k@naver.com | 서울특별시 용산구  |
| 9  | nikola tesla       | 010-1212-1212 | 1990-06-01 | Male   | tesla.n@naver.com     | 경기 수원시 영통구 |
| 10 | marie curie        | 010-9876-5432 | 1990-07-01 | Female | curie.m@naver.com     | 경기 수원시 팔달구 |

10 rows in set (0.00 sec)

```
mysql> /* Name Column의 레코드 중
/*> 대문자가 존재하는 경우
/*> 해당 레코드 전체 조회*/
mysql> SELECT *
-> FROM member_list
-> WHERE
-> BINARY(Name) REGEXP '[A-Z]\D*';
```

| No | Name               | Phone         | Birthday   | Gender | Email                 | Address    |
|----|--------------------|---------------|------------|--------|-----------------------|------------|
| 1  | Joshua Graham      | 010-2345-6789 | 1991-09-23 | Male   | Graham.J@naver.com    | 서울특별시 종로구  |
| 2  | Mariah Carey       | 010-1234-5555 | 1990-04-01 | Female | Carey.M@naver.com     | 서울특별시 광진구  |
| 3  | Katheryn Elizabeth | 010-1111-2222 | 1990-05-01 | Female | Elizabeth.K@naver.com | 서울특별시 용산구  |
| 4  | Nikola Tesla       | 010-1212-1212 | 1990-06-01 | Male   | Tesla.N@naver.com     | 경기 수원시 영통구 |
| 5  | Marie Curie        | 010-9876-5432 | 1990-07-01 | Female | Curie.M@naver.com     | 경기 수원시 팔달구 |

5 rows in set (0.00 sec)

• 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

# [복습과제] 회원 관리 테이블 구축

```
mysql> /* Name Column의 레코드 중
/*> 대문자가 존재하지 않는 경우
/*> 해당 레코드 전체 조회*/
```

```
mysql> SELECT *
-> FROM member_list
-> WHERE
-> BINARY(Name) NOT REGEXP '[A-Z]#D*';
```

| No | Name               | Phone         | Birthday   | Gender | Email                 | Address    |
|----|--------------------|---------------|------------|--------|-----------------------|------------|
| 6  | joshua graham      | 010-2345-6789 | 1991-09-23 | Male   | graham.j@naver.com    | 서울특별시 중랑구  |
| 7  | mariah carey       | 010-1234-5555 | 1990-04-01 | Female | carey.m@naver.com     | 서울특별시 강남구  |
| 8  | katheryn elizabeth | 010-1111-2222 | 1990-05-01 | Female | elizabeth.k@naver.com | 서울특별시 양천구  |
| 9  | nikola tesla       | 010-1212-1212 | 1990-06-01 | Male   | tesla.n@naver.com     | 경기 수원시 영통구 |
| 10 | marie curie        | 010-9876-5432 | 1990-07-01 | Female | curie.m@naver.com     | 경기 수원시 팔달구 |

5 rows in set (0.00 sec)

```
mysql> ALTER TABLE member_list
-> ADD COLUMN etc CHAR(20) DEFAULT ' ' NOT NULL
-> AFTER Address;
```

Query OK, 0 rows affected (0.03 sec)  
Records: 0 Duplicates: 0 Warnings: 0

```
mysql> DESCRIBE member_list;
```

| Field    | Type         | Null | Key | Default | Extra          |
|----------|--------------|------|-----|---------|----------------|
| No       | int(11)      | NO   | PRI | NULL    | auto_increment |
| Name     | char(30)     | YES  |     | NULL    |                |
| Phone    | char(15)     | YES  |     | NULL    |                |
| Birthday | date         | YES  |     | NULL    |                |
| Gender   | char(10)     | YES  |     | NULL    |                |
| Email    | varchar(100) | YES  |     | NULL    |                |
| Address  | varchar(255) | YES  |     | NULL    |                |
| etc      | char(20)     | NO   |     | NULL    |                |

8 rows in set (0.00 sec)

```
mysql> SELECT * FROM member_list;
```

| No | Name               | Phone         | Birthday   | Gender | Email                 | Address    | etc |
|----|--------------------|---------------|------------|--------|-----------------------|------------|-----|
| 1  | Joshua Graham      | 010-2345-6789 | 1991-09-23 | Male   | Graham.J@naver.com    | 서울특별시 중랑구  |     |
| 2  | Mariah Carey       | 010-1234-5555 | 1990-04-01 | Female | Carey.M@naver.com     | 서울특별시 강남구  |     |
| 3  | Katheryn Elizabeth | 010-1111-2222 | 1990-05-01 | Female | Elizabeth.K@naver.com | 서울특별시 양천구  |     |
| 4  | Nikola Tesla       | 010-1212-1212 | 1990-06-01 | Male   | Tesla.N@naver.com     | 경기 수원시 영통구 |     |
| 5  | Marie Curie        | 010-9876-5432 | 1990-07-01 | Female | Curie.M@naver.com     | 경기 수원시 팔달구 |     |
| 6  | joshua graham      | 010-2345-6789 | 1991-09-23 | Male   | graham.j@naver.com    | 서울특별시 중랑구  |     |
| 7  | mariah carey       | 010-1234-5555 | 1990-04-01 | Female | carey.m@naver.com     | 서울특별시 강남구  |     |
| 8  | katheryn elizabeth | 010-1111-2222 | 1990-05-01 | Female | elizabeth.k@naver.com | 서울특별시 양천구  |     |
| 9  | nikola tesla       | 010-1212-1212 | 1990-06-01 | Male   | tesla.n@naver.com     | 경기 수원시 영통구 |     |
| 10 | marie curie        | 010-9876-5432 | 1990-07-01 | Female | curie.m@naver.com     | 경기 수원시 팔달구 |     |

10 rows in set (0.00 sec)

- ```
SELECT *
FROM item_price
WHERE BINARY(Name) NOT REGEXP '[A-Z]\D*';
```

 - ✓ Name column의 데이터 중 소문자만 존재하는 레코드의 데이터 전체 출력
- ```
ALTER TABLE member_list
ADD COLUMN etc CHAR(20) DEFAULT ' ' NOT NULL
AFTER Address;
```

  - ✓ member\_list 내 대소문자 구분없이 모든 데이터 조회
- ```
DESCRIBE member_list;
```

 - ✓ COLUMN이 의도한대로 추가되었는지 확인
- ```
SELECT * FROM member_list;
```

  - ✓ member\_list TABLE 내 모든 데이터 조회
  - ✓ 최종 확인 결과 이상 없음