데이터베이스

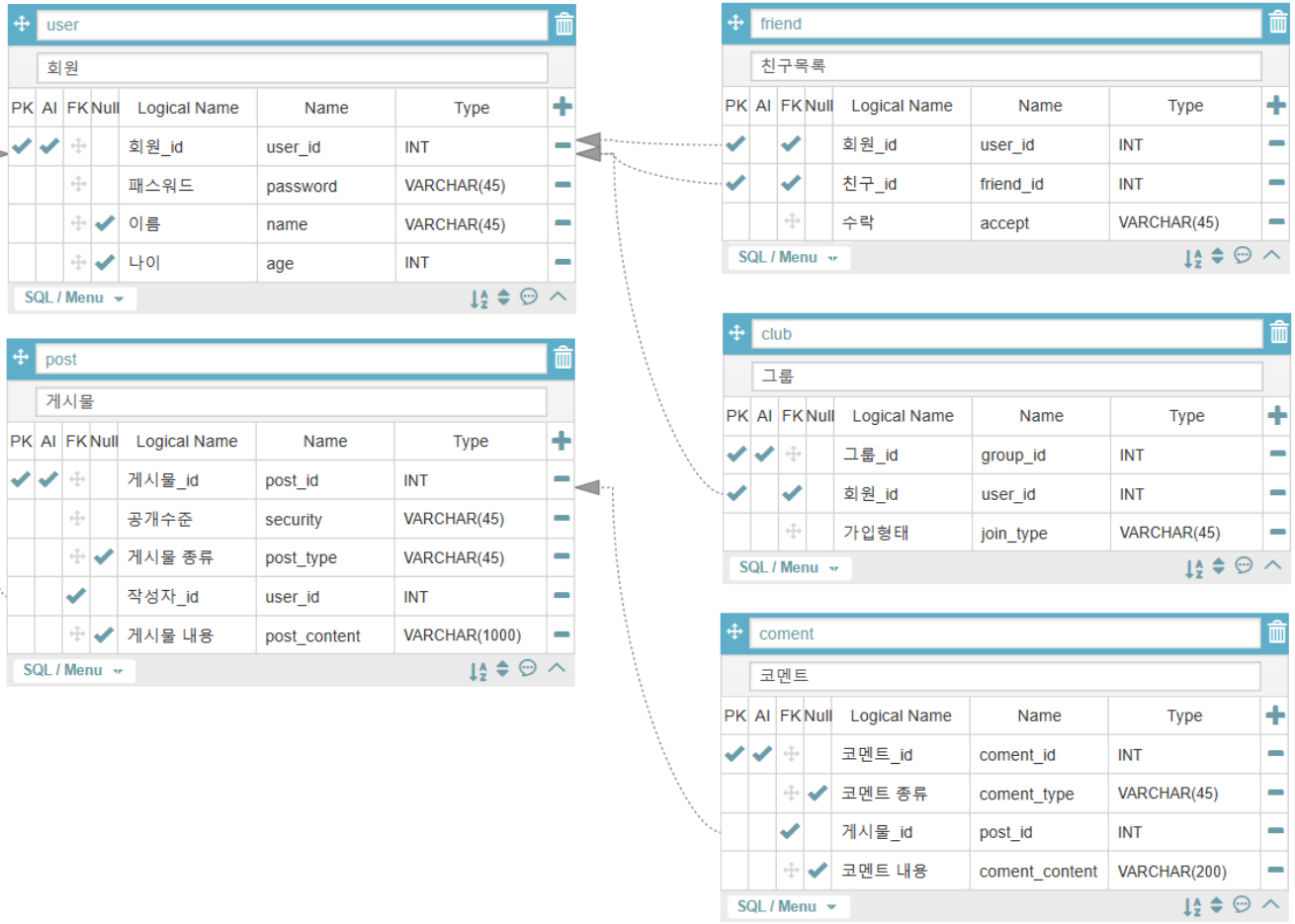
- MSQL프로젝트-

정보융합전공
2017053245 정은지

CONTENTS

1. ERD

1) 테이블 정의


user
회원
PK AI FK Null Logical Name Name Type
회원_id user_id INT
패스워드 password VARCHAR(45)
이름 name VARCHAR(45)
나이 age INT
SQL / Menu
post
게시물
PK AI FK Null Logical Name Name Type
게시물_id post_id INT
공개수준 security VARCHAR(45)
게시물 종류 post_type VARCHAR(45)
작성자_id user_id INT
게시물 내용 post_content VARCHAR(1000)
SQL / Menu
friend
친구목록
PK AI FK Null Logical Name Name Type
회원_id user_id INT
친구_id friend_id INT
수락 accept VARCHAR(45)
SQL / Menu
club
그룹
PK AI FK Null Logical Name Name Type
그룹_id group_id INT
회원_id user_id INT
가입형태 join_type VARCHAR(45)
SQL / Menu
coment
코멘트
PK AI FK Null Logical Name Name Type
코멘트_id coment_id INT
코멘트 종류 coment_type VARCHAR(45)
게시물_id post_id INT
코멘트 내용 coment_content VARCHAR(200)
SQL / Menu

1. ERD

1) 테이블 정의

테이블명	설명
회원	회원의 정보 저장
친구목록	친구 신청 및 관리 정보 저장
그룹	그룹의 정보 저장
게시물	회원이 생성하는 게시물 정보 저장
코멘트	게시물에 대한 코멘트 정보 저장

1. ERD

2) 속성 설명

(1) 회원

속성명 (Logical name)	속성명 (Name)	Type	Key	설명
회원_id	user_id	int (not null)	PK	회원의 ID, 회원 정보를 유일하게 식별할 수 있는 속성이므로 **PK로 지정, PK이므로 not null로 지정**
패스워드	password	varchar (not null)	-	회원의 비밀번호, 여러 회원이 동일한 패스워드를 사용할 수 있으므로 PK로 지정 불가함 그러나 **로그인 필수 정보이므로 not null**로 지정
이름	name	varchar	-	회원의 이름, 동명이인이 있을 수 있으므로 PK로 지정 불가 회원 **부가 정보이므로 null값 허용**
나이	age	int	-	회원의 나이, 중복값이 있을 수 있으므로 PK로 지정 불가 회원 **부가 정보이므로 null값 허용**

1. ERD

2) 속성 설명

(2) 친구목록

속성명 (Logical name)	속성명 (Name)	Type	Key	설명
회원_id	user_id	int (not null)	PK, FK	회원의 ID와 친구의 ID, 여러 회원과 여러 친구가 친구 관계를 맺을 수 있고 **특정 회원과 친구의 관계를 유일하게 식별해야 하므로 두 키를 PK로 지정**하고, PK이므로 **not null**로 지정 또한, 친구 신청의 매개체가 회원이므로 회원 테이블에 **회원_ID 속성을 참조**함
친구_id	friend_id	int (not null)	PK, FK	
수락	accept	varchar (not null)	-	친구 관계를 나타내는 속성, 친구 신청 단계에는 '신청'으로 표시, 친구 수락 시 '지금부터 친구'로 표시됨 친구 관계에서 **필수 사항이므로 not null**로 지정

1. ERD

2) 속성 설명

(3) 그룹

속성명 (Logical name)	속성명 (Name)	Type	Key	설명
그룹_id	group_id	int (not null)	PK	그룹의 ID, 한 회원이 여러 그룹을 생성, 가입할 수 있으므로 특정 회원과 특정 그룹의 관계를 유일하게 식별해야 함 **따라서, 두 키를 PK로 지정**하고, PK이므로 **not null**로 지정 단, 그룹을 생성, 가입하는 매개체는 회원이므로 **회원 ID만 회원 테이블에 회원_ID 속성을 참조**함
회원_id	user_id	int (not null)	PK, FK	
가입형태	join_type	varchar (not null)	-	그룹의 가입형태, 회원이 그룹을 생성했을 경우(ex.그룹장) '생성'으로 표기 회원이 그룹에 가입했을 경우 '가입'으로 표기 그룹 가입에서 **필수 사항이므로 not null**로 지정

1. ERD

2) 속성 설명

(4) 게시물

속성명 (Logical name)	속성명 (Name)	Type	Key	설명
게시물_id	post_id	int (not null)	PK	게시물의 ID, 게시물 정보를 유일하게 식별할 수 있는 속성이므로 **PK로 지정, PK이므로 not null**로 지정
공개수준	security	varchar (not null)	-	게시물의 공개수준, 게시물 생성 시 **기본값 '공개'**로 설정 **회원이 변경할 경우 '비공개'**로 설정 가능 게시물 생성 **필수 정보이므로 not null**로 지정
게시물 종류	post_ type	varchar	-	게시물의 종류, 종류(ex. 사진 또는 동영상)에 대한 정보가 들어있음 게시물 **부가 정보이므로 null값 허용**
작성자_id	user_id	int (not null)	FK	게시물을 작성한 회원의 ID, 게시물 생성 **필수 정보이자 외래키이므로 not null**로 지정 게시물 작성 매개체가 회원이므로 **회원 테이블에** **회원_ID 속성을 참조**함
게시물 내용	post_ content	varchar	-	게시물의 내용, 게시물에 기재되는 텍스트나 추가 내용 (ex. 사진과 동영상 파일의 이름) 게시물 **부가 정보이므로 null값 허용**

1. ERD

2) 속성 설명

(5) 코멘트

속성명 (Logical name)	속성명 (Name)	Type	Key	설명
코멘트_id	coment_id	int (not null)	PK	코멘트의 ID, 코멘트 정보를 유일하게 식별할 수 있는 속성이므로 **PK로 지정, PK이므로 not null**로 지정 한 게시물에 여러 코멘트가 생성될 수 있으므로 코멘트_id만을 PK로 지정
코멘트 종류	coment_type	varchar	-	코멘트의 종류, 종류(ex. 댓글 또는 좋아요)에 대한 정보가 들어있음 코멘트 **부가 정보이므로 null값 허용**
게시물_id	post_id	int (not null)	FK	코멘트가 달려있는 게시물의 ID, 코멘트 생성 **필수 정보이자 외래키이므로 not null**로 지정 코멘트가 달린 매개체가 게시물이므로 **게시물 테이블에 게시물_ID 속성을 참조**함
코멘트 내용	coment_content	varchar	-	코멘트의 내용, 코멘트 속 부가내용(ex. 댓글 내용과 좋아요) **코멘트 부가 정보이므로 null값 허용**

1. ERD

3) 기능 구현 설명

(1) 회원가입

: 페이스북 회원인지 확인 후, 만약 회원이 아니면 회원가입 진행

(1-1). 사용자에게 입력 받은 ID 체크

(1-2). 입력 받은 ID가 DB에 없을 경우 새로운 ID 생성

(2) 그룹 생성

: 회원이 페이스북 내에서 그룹을 만들거나 원하는 그룹에 가입

(2-1). 페이스북 내 동일한 그룹이 있는지 먼저 체크

(2-2). 만약 원하는 그룹이 있는 경우 그 그룹에 가입

(2-3). 만약 원하는 그룹이 없는 경우 새 그룹을 생성

(3) 친구 관리

: 페이스북 내 다른 회원과 친구 관계 생성

회원은 다른 회원에게 친구 요청을 보낼 수 있고 상대가 요청을 수락하면 친구 관계가 됨

(3-1). 본인의 ID와 친구 신청하려는 사람의 ID를 사용자에게 입력 받음

(3-2). 만약 친구 신청 기록이 없다면 친구 신청 가능

(3-3). 친구 신청 기록이 있는 경우는 이미 친구이거나
기존에 친구 신청을 보낸 적이 있는데 상대가 수락하지 않았을 경우임
후자의 경우, 동일인에게 다시 친구 신청을 보낼 수 없으며 상대가 수락할 경우만 친구가 됨

1. ERD

3) 기능 구현 설명

(4) 게시물 작성

: 회원은 사진 또는 동영상 형태의 게시물을 생성할 수 있음
이 때, 게시물의 공개 수준을 설정할 수 있음

(4-1). 생성하려는 게시물의 ID가 기존에 존재하는지 체크

(4-2). 위 게시물 ID가 존재하지 않을 경우 새로운 게시물 생성 진행
이 때, 게시물의 종류를 설정하여 생성할 수 있음(사진, 동영상)

(4-3). 사용자가 원할 경우 게시물의 보안단계를 수정
게시물 생성 시 기본값 '공개'로 지정되며, 사용자가 원하면 '비공개'로 수정 가능
게시물은 '공개'일 때 모든 회원에게 열람이 가능함

(5) 코멘트 작성

: 게시물에 달리는 코멘트를 생성할 수 있고, 코멘트는 댓글 또는 좋아요 형태로 생성됨

(5-1). 생성하려는 코멘트 ID가 존재하는지 체크

(5-2). 기존에 동일한 ID가 없을 경우 새로운 코멘트 생성 가능

(5-3). 코멘트 종류는 댓글 또는 좋아요로 생성

(5-4). 코멘트에는 부가 내용(댓글의 내용) 등을 추가할 수 있음

2. 테이블 생성

1) User : 회원 정보가 들어있는 테이블

```
mysql> desc user
    -> ;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| user_id  | int(20)     | NO   | PRI | NULL    |       |
| password | varchar(45) | NO   |     | NULL    |       |
| name     | varchar(45) | YES  |     | NULL    |       |
| age      | int(10)     | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

```
mysql> create table user(
    -> user_id int(20) not null,
    -> password varchar(45) not null,
    -> name varchar(45),
    -> age int(10),
    -> primary key(user_id));
```

2. 테이블 생성

2) friend : 친구 정보, 친구 관계 관리를 위한 테이블

```
mysql> desc friend;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| user_id   | int(20)     | NO   | PRI | NULL    |       |
| friend_id | int(20)     | NO   | PRI | NULL    |       |
| accept    | varchar(45) | NO   |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

```
mysql> create table friend(
    -> user_id int(20) not null,
    -> friend_id int(20) not null,
    -> accept varchar(45) not null,
    -> primary key(user_id, friend_id),
    -> foreign key(user_id)
        references user(user_id),
    -> foreign key(friend_id)
        references user(user_id));
```

3) club : 그룹 생성 등 관리를 위한 테이블

```
mysql> desc club;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| group_id  | int(20)     | NO   | PRI | NULL    |       |
| user_id   | int(20)     | NO   | PRI | NULL    |       |
| join_type | varchar(45) | NO   |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

```
mysql> create table club(
    -> group_id int(20) not null,
    -> user_id int(20) not null,
    -> join_type varchar(45) not null,
    -> primary key(group_id, user_id),
    -> foreign key(user_id)
        references user(user_id));
```

2. 테이블 생성

4) post : 게시글 관련 정보가 담긴 테이블

```
mysql> desc post;
+--------------+---------------+------+-----+---------+-------+
| Field        | Type          | Null | Key | Default | Extra |
+--------------+---------------+------+-----+---------+-------+
| post_id      | int(20)       | NO   | PRI | NULL    |       |
| security     | varchar(45)   | NO   |     | NULL    |       |
| post_type    | varchar(45)   | YES  |     | NULL    |       |
| user_id      | int(20)       | NO   | MUL | NULL    |       |
| post_content | varchar(1000) | YES  |     | NULL    |       |
+--------------+---------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

```
mysql> create table post(
    -> post_id int(20) not null,
    -> security varchar(45) not null,
    -> post_type varchar(45),
    -> user_id int(20) not null,
    -> post_content varchar(1000),
    -> primary key(post_id),
    -> foreign key(user_id)
       references user(user_id));
```

5) Coment : 댓글 관리를 위한 테이블

```
mysql> desc coment;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| coment_id     | int(20)     | NO   | PRI | NULL    |       |
| coment_type   | varchar(45) | YES  |     | NULL    |       |
| post_id       | int(20)     | NO   | MUL | NULL    |       |
| coment_content| varchar(200)| YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

```
mysql> create table coment(
    -> coment_id int(20) not null,
    -> coment_type varchar(45),
    -> post_id int(20) not null,
    -> coment_content varchar(200),
    -> primary key(coment_id),
    -> foreign key(post_id)
       references post(post_id));
```

3. 쿼리 결과

1) 데이터 추가 과정

User 테이블에 회원 정보 추가 (회원 ID 0001, 0002, 0003)

```
mysql> use final_task;
Database changed
mysql> insert into user values(0001, '1234', '한양이', 20);
Query OK, 1 row affected (0.07 sec)

mysql> insert into user values(0002, '5678', '공학이', 21);
Query OK, 1 row affected (0.01 sec)

mysql> insert into user values(0003, '1234', '공돌이', 22);
Query OK, 1 row affected (0.01 sec)

mysql> select * from user;
+---------+----------+-----------+------+
| user_id | password | name      | age  |
+---------+----------+-----------+------+
|       1 | 1234     | 한양이    |   20 |
|       2 | 5678     | 공학이    |   21 |
|       3 | 1234     | 공돌이    |   22 |
+---------+----------+-----------+------+
3 rows in set (0.01 sec)
```

3. 쿼리 결과

1) 데이터 추가 과정

friend 테이블에 친구 정보 추가
(회원 0001 -> 회원 0002, 회원 0002 -> 회원 0003)

```
mysql> insert into friend values(0001, 0002, '신청');
Query OK, 1 row affected (0.02 sec)

mysql> insert into friend values(0002, 0003, '신청');
Query OK, 1 row affected (0.01 sec)

mysql> select * from friend;                          ;
+---------+-----------+--------+
| user_id | friend_id | accept |
+---------+-----------+--------+
|       1 |         2 | 신청   |
|       2 |         3 | 신청   |
+---------+-----------+--------+
2 rows in set (0.00 sec)

mysql> _
```

3. 쿼리 결과

1) 데이터 추가 과정

dub 테이블에 그룹 정보 추가
(그룹 1001 : 회원 0001이 생성, 회원 0002는 가입)

```
mysql> insert into club values(1001, 0001, '생성');
Query OK, 1 row affected (0.01 sec)

mysql> insert into club values(1001, 0002, '가입');
Query OK, 1 row affected (0.01 sec)

mysql> select * from club;
+----------+---------+-----------+
| group_id | user_id | join_type |
+----------+---------+-----------+
|     1001 |       1 | 생성      |
|     1001 |       2 | 가입      |
+----------+---------+-----------+
```

3. 쿼리 결과

1) 데이터 추가 과정

post 테이블에 게시물 정보 추가
(post 2001 : 0001 회원이 게시한 사진,
post 2002 : 0002 회원이 게시한 동영상)

```
mysql> insert into post values(2001, '공개', '사진', 0001, '동해물과 백두산이.jpg');
Query OK, 1 row affected (0.02 sec)

mysql> insert into post values(2002, '공개', '동영상', 0002, '산업융합학부 홍보영상.mp4');;
Query OK, 1 row affected (0.01 sec)

mysql> select * from post;
+---------+----------+-----------+---------+-----------------------------------+
| post_id | security | post_type | user_id | post_content                      |
+---------+----------+-----------+---------+-----------------------------------+
|    2001 | 공개     | 사진      |       1 | 동해물과 백두산이.jpg             |
|    2002 | 공개     | 동영상    |       2 | 산업융합학부 홍보영상.mp4         |
+---------+----------+-----------+---------+-----------------------------------+
2 rows in set (0.00 sec)
```

3. 쿼리 결과

1) 데이터 추가 과정

Coment 테이블에 코멘트 정보 추가
(coment 3001 : post 2001에 달린 댓글,
coment 3002 : post 2001에 달린 좋아요)

```
mysql> insert into coment values(3001, '댓글', 2001, '아름다워요');
Query OK, 1 row affected (0.01 sec)

mysql> insert into coment values(3002, '좋아요', 2001, '좋아요');;;);
Query OK, 1 row affected (0.01 sec)

mysql> select * from coment;
+-----------+-------------+---------+-----------------+
| coment_id | coment_type | post_id | coment_content  |
+-----------+-------------+---------+-----------------+
|      3001 | 댓글        |    2001 | 아름다워요      |
|      3002 | 좋아요      |    2001 | 좋아요          |
+-----------+-------------+---------+-----------------+
2 rows in set (0.00 sec)

mysql> _
```

3. 쿼리 결과

2) SQL 쿼리문

(1) 회원가입

(1-1). ID 체크

- select count(*) from users where user_id = '사용자 입력';

(1-2). ID 생성

- insert into user values(user_id, password, name, age);

(2) 그룹 생성

(2-1). group 체크

- select count(*) from club where group_id = '사용자 입력';

(2-2). group 가입

- insert into club values(group_id, user_id, join_type);

(2-3). group 생성

- insert into club values(group_id, user_id, join_type);

3. 쿼리 결과

2) SQL 쿼리문

(3) 친구 목록

(3-1). 친구 목록 확인

- select accept from friend where user_id = '본인 ID 입력' and friend_id = '친구 ID 입력';
  (사용자 입력 아이디가 있을 경우 accept '신청'을 출력, 없을 경우 친구 신청 가능)

(3-2). 친구 신청

- insert into friend values(user_id, friend_id, accept);

(3-3). 친구 수락

- update friend set accept = '지금부터 친구' where friend_id = '현재 수락한 ID'
  and user_id = '수락할 ID';
- insert into friend values(user_id, friend_id, accept);

3. 쿼리 결과

2) SQL 쿼리문

(4) 게시물 작성

(4-1). 게시물 확인

- select count(*) from post where post_id = '현재 생성 ID';

(4-2). 게시물 작성

- insert into post values(post_id, security, post_type, user_id);

(4-3). 게시물 보안단계 수정(기본값 공개로 생성)

- update post set security = '비공개' where security = '공개' and user_id = '사용자 ID' and post_id = '바꾸려는 post_id';

3. 쿼리 결과

3) 실행 결과

(1) 회원가입

(1-1). ID 체크 : 0004 회원이 있는지 확인

(1-2). ID 생성 : 회원이 없으므로 신규 ID 생성 -> 생성 후 회원 테이블에 추가된 결과 확인

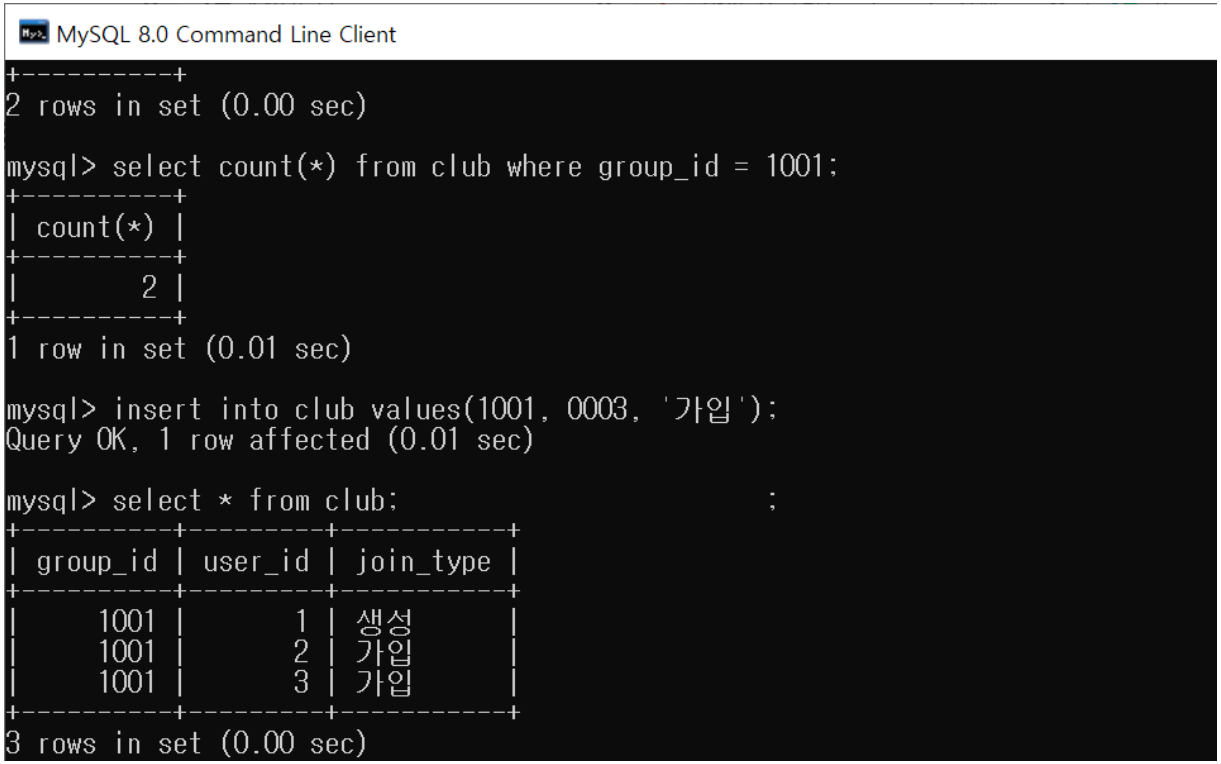
```
MySQL 8.0 Command Line Client
mysql> insert into user values(0003, '1234', '공돌이', 22);
Query OK, 1 row affected (0.01 sec)

mysql> select * from user;
+---------+----------+-----------+------+
| user_id | password | name      | age  |
+---------+----------+-----------+------+
|       1 | 1234     | 한양이    |   20 |
|       2 | 5678     | 공학이    |   21 |
|       3 | 1234     | 공돌이    |   22 |
+---------+----------+-----------+------+
3 rows in set (0.01 sec)

mysql> select count(*) from users where user_id = 0004;
ERROR 1146 (42S02): Table 'final_task.users' doesn't exist
mysql> insert into user values(0004, '3456', '정융이', 21);
Query OK, 1 row affected (0.01 sec)

mysql> select * from user;                                );
+---------+----------+-----------+------+
| user_id | password | name      | age  |
+---------+----------+-----------+------+
|       1 | 1234     | 한양이    |   20 |
|       2 | 5678     | 공학이    |   21 |
|       3 | 1234     | 공돌이    |   22 |
|       4 | 3456     | 정융이    |   21 |
+---------+----------+-----------+------+
4 rows in set (0.00 sec)
```

3. 쿼리 결과

3) 실행 결과

(2) 그룹 생성

(2-1). group 체크 : 0003 회원이 1001 그룹이 있는지 확인
(2-2). group 가입 : 이미 있는 그룹을 확인함. 가입을 진행



3. 쿼리 결과

3) 실행 결과

(2) 그룹 생성

(2-3). group 생성 : 0003 회원이 1002 그룹이 있는지 확인하고
생성 가능한 그룹이므로 생성함

```
mysql> select count(*) from club where group_id = 1002;  );
+----------+
| count(*) |
+----------+
|        0 |
+----------+
1 row in set (0.00 sec)

mysql> insert into club values(1002, 0003, '생성');
Query OK, 1 row affected (0.01 sec)

mysql> select * from club;
+----------+---------+-----------+
| group_id | user_id | join_type |
+----------+---------+-----------+
|     1001 |       1 | 생성      |
|     1001 |       2 | 가입      |
|     1001 |       3 | 가입      |
|     1002 |       3 | 생성      |
+----------+---------+-----------+
4 rows in set (0.00 sec)
```

3. 쿼리 결과

3) 실행 결과

(3) 친구 목록

(3-1). 친구 목록 확인 : 0001 회원이 0002 회원에게 친구 신청을 보낸 상태이나
아직 0002 회원이 수락 전임 -> 다시 친구 신청 불가, 친구의 수락을 기다려야함
0001 회원이 0003 회원에게 친구 신청을 보내려하는데 친구 신청 이력이 없음

(3-2). 친구 신청 : 0001 회원이 0003 회원에게 친구 신청을 보냄

(3-3). 친구 수락 : 0002 회원이 0001 회원이 보내온 친구 신청을 수락함
0001 회원의 결과창에도 친구 수락이 된 걸 확인할 수 있음
친구는 서로 관계를 맺으므로 0002 회원의 정보도 0001 회원과
'지금부터 친구' 관계로 추가 삽입

```
mysql> select accept from friend where user_id = 0001 and friend_id = 0002;
+--------+
| accept |
+--------+
| 신청   |
+--------+
1 row in set (0.00 sec)

mysql> select accept from friend where user_id = 0001 and friend_id = 0003;
Empty set (0.00 sec)

mysql> insert into friend values(0001, 0003, '신청');
Query OK, 1 row affected (0.01 sec)
```

3. 쿼리 결과

3) 실행 결과

(3) 친구 목록

```
mysql> update set friend accept = '지금부터 친구' where friend_id = 0002 and user_id = 0001;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version
 for the right syntax to use near 'set friend accept = '지금부터 친구' where friend_id = 0002 and user_id = 0' at line 1


mysql> update friend set accept = '지금부터 친구' where friend_id = 0002 and user_id = 0001;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select accept from friend where user_id = 0001 and friend_id = 0002;           0001;
+--------------------+
| accept             |
+--------------------+
| 지금부터 친구      |
+--------------------+
1 row in set (0.00 sec)

mysql> insert into friend values(0002, 0001, '지금부터 친구');
Query OK, 1 row affected (0.01 sec)

mysql> select * from friend;
+---------+-----------+--------------------+
| user_id | friend_id | accept             |
+---------+-----------+--------------------+
|       1 |         2 | 지금부터 친구      |
|       1 |         3 | 신청               |
|       2 |         1 | 지금부터 친구      |
|       2 |         3 | 신청               |
+---------+-----------+--------------------+
```

3. 쿼리 결과

3) 실행 결과

(4) 게시물 작성

(4-1). 게시물 확인 : 게시물 2001이 있는지 확인 -> 있으므로 새로 생성 불가
게시물 2003이 있는지 확인 -> 없으므로 새로 생성 가능

(4-2). 게시물 작성 : 게시물 2003을 생성 -> 게시물이 생성된 것을 확인할 수 있음

(4-3). 게시물 보안단계 수정(기본값 공개로 생성) : 0002 회원이 작성한 게시물 2002의 공개수준을 수정 (공개 -> 비공개) -> 결과값이 반영된 것을 확인할 수 있음

```
mysql> select count(*) from post where post_id = 2001;
+----------+
| count(*) |
+----------+
|        1 |
+----------+
1 row in set (0.01 sec)

mysql> select count(*) from post where post_id = 2003;
+----------+
| count(*) |
+----------+
|        0 |
+----------+
1 row in set (0.00 sec)

mysql> insert into post values(2003, '공개', '동영상', 0002, '산업융합학부 MT 영상.mp4');;
Query OK, 1 row affected (0.01 sec)
```

3. 쿼리 결과

3) 실행 결과

(4) 게시물 작성

```
mysql> select * from post;                                        개';
+---------+----------+-----------+---------+-----------------------------------+
| post_id | security | post_type | user_id | post_content                      |
+---------+----------+-----------+---------+-----------------------------------+
|    2001 | 공개     | 사진      |       1 | 동해물과 백두산이.jpg             |
|    2002 | 공개     | 동영상    |       2 | 산업융합학부 홍보영상.mp4         |
|    2003 | 공개     | 동영상    |       2 | 산업융합학부 MT 영상.mp4          |
+---------+----------+-----------+---------+-----------------------------------+
3 rows in set (0.00 sec)

mysql> update post set security = '비공개' where security = '공개' and user_id = 0002 and post_id = 2002;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from post;                                        개';                                  002;
+---------+-----------+-----------+---------+-----------------------------------+
| post_id | security  | post_type | user_id | post_content                      |
+---------+-----------+-----------+---------+-----------------------------------+
|    2001 | 공개      | 사진      |       1 | 동해물과 백두산이.jpg             |
|    2002 | 비공개    | 동영상    |       2 | 산업융합학부 홍보영상.mp4         |
|    2003 | 공개      | 동영상    |       2 | 산업융합학부 MT 영상.mp4          |
+---------+-----------+-----------+---------+-----------------------------------+
3 rows in set (0.00 sec)
```