8.28 외래키, Union, Sub Query, ERD

데이터 베이스 엔진

테이블 엔진 조회

WebMail

[root@Linux-04 바탕화면]# mysql -uroot -pkeroro2424.

```
mysql> show engines;
+------------+---------+----------------------------------------------------------------
----+--------------+------+------------+
| Engine     | Support | Comment
   | Transactions | XA   | Savepoints |
+------------+---------+----------------------------------------------------------------
----+--------------+------+------------+
| MRG_MYISAM | YES     | Collection of identical MyISAM tables
   | NO           | NO   | NO         |
| CSV        | YES     | CSV storage engine
    | NO           | NO   | NO         |
| MyISAM     | DEFAULT | Default engine as of MySQL 3.23 with great performance
    | NO           | NO   | NO         |
| InnoDB     | YES     | Supports transactions, row-level locking, and foreign
keys | YES          | YES  | YES        |
| MEMORY     | YES     | Hash based, stored in memory, useful for temporary
tables  | NO           | NO   | NO         |
+------------+---------+----------------------------------------------------------------
----+--------------+------+------------+
5 rows in set (0.00 sec)

지금 기본값은 MyISAM
InnoDB로 기본값 바꾼다

mysql> set global storage_engine=InnoDB;
Query OK, 0 rows affected (0.01 sec)
exit

[root@Linux-04 바탕화면]# mysql -uroot -pkeroro2424.
mysql> show engines;
+------------+---------+----------------------------------------------------------------
----+--------------+------+------------+
| Engine     | Support | Comment
   | Transactions | XA   | Savepoints |
+------------+---------+----------------------------------------------------------------
----+--------------+------+------------+
| MRG_MYISAM | YES     | Collection of identical MyISAM tables
   | NO           | NO   | NO         |
| CSV        | YES     | CSV storage engine
    | NO           | NO   | NO         |
| MyISAM     | YES     | Default engine as of MySQL 3.23 with great performance
    | NO           | NO   | NO         |
| InnoDB     | DEFAULT | Supports transactions, row-level locking, and foreign
keys | YES          | YES  | YES        |
```

```
| MEMORY     | YES     | Hash based, stored in memory, useful for temporary
tables  | NO           | NO   | NO         |
+------------+---------+-----------------------------------------------------
----+--------------+------+------------+
5 rows in set (0.00 sec)

-> 일시적인 바꿈. reboot 하면 다시 기본값이 MyISAM으로 돌아옴
```

Reboot 해도 기본값 InnoDB로 설정하기

[root@Linux-04 바탕화면]# vim /etc/my.cnf

```
 1 [mysqld]
 2 datadir=/var/lib/mysql
 3 socket=/var/lib/mysql/mysql.sock
 4 user=mysql
 5 # Disabling symbolic-links is recommended to prevent assorted security risks
 6 symbolic-links=0
 7
 8 default-storage-engine=InnoDB // 여기 추가해주기
 9
10 skip-character-set-client-handshake=FALSE
11
12 init_connect=SET collation_connection=utf8_general_ci
13 init_connect=SET NAMES utf8
14 character-set-server=utf8
15 collation-server=utf8_general_ci
16
17
18 [client]
19 default-character-set=utf8
20
21 [mysqldump]
22 default-character-set=utf8
23
24 [mysql]
25 default-character-set=utf8
26
27 [mysqld_safe]
28 log-error=/var/log/mysqld.log
29 pid-file=/var/run/mysqld/mysqld.pid
```

[root@Linux-04 바탕화면]# service mysqld restart

[root@Linux-04 바탕화면]# mysql -uroot -pkeroro2424.

```
mysql> show engines;
+------------+---------+----------------------------------------------------------------
----+--------------+------+------------+
| Engine     | Support | Comment
   | Transactions | XA   | Savepoints |
+------------+---------+----------------------------------------------------------------
----+--------------+------+------------+
| MRG_MYISAM | YES     | Collection of identical MyISAM tables
   | NO           | NO   | NO         |
| CSV        | YES     | CSV storage engine
    | NO           | NO   | NO         |
| MyISAM     | YES     | Default engine as of MySQL 3.23 with great performance
    | NO           | NO   | NO         |
| InnoDB     | DEFAULT | Supports transactions, row-level locking, and foreign
keys | YES          | YES  | YES        |
| MEMORY     | YES     | Hash based, stored in memory, useful for temporary
tables  | NO           | NO   | NO         |
+------------+---------+----------------------------------------------------------------
----+--------------+------+------------+
5 rows in set (0.00 sec)
```

```
mysql> use naver_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show create table member;
+--------+--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
----+
| Table  | Create Table




    |
+--------+--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
----+
| member | CREATE TABLE `member` (
  `no` int(11) NOT NULL AUTO_INCREMENT,
  `id` varchar(10) NOT NULL,
  `name` varchar(20) NOT NULL,
  `sex` char(1) DEFAULT NULL,
  `post_num` char(8) DEFAULT NULL,
  `address` varchar(80) DEFAULT NULL,
  `tel` varchar(15) DEFAULT NULL,
  `age` int(11) DEFAULT NULL,
```

```
  PRIMARY KEY (`no`),
  UNIQUE KEY `id` (`id`)
) ENGINE=MyISAM AUTO_INCREMENT=18 DEFAULT CHARSET=utf8 |   //지금 기본엔진 MyISAM
+--------+-------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
----+
1 row in set (0.00 sec)

mysql>


mysql> alter table member engine=InnoDB; // 기본 엔진 InnoDB로 바꾸기
Query OK, 16 rows affected (0.02 sec)
Records: 16  Duplicates: 0  Warnings: 0

mysql> show create table member;
+--------+-------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
----+
| Table  | Create Table




    |
+--------+-------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
----+
| member | CREATE TABLE `member` (
  `no` int(11) NOT NULL AUTO_INCREMENT,
  `id` varchar(10) NOT NULL,
  `name` varchar(20) NOT NULL,
  `sex` char(1) DEFAULT NULL,
  `post_num` char(8) DEFAULT NULL,
  `address` varchar(80) DEFAULT NULL,
  `tel` varchar(15) DEFAULT NULL,
  `age` int(11) DEFAULT NULL,
  PRIMARY KEY (`no`),
  UNIQUE KEY `id` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=18 DEFAULT CHARSET=utf8 | //기본엔진 바뀜!
+--------+-------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
----+
1 row in set (0.00 sec)
```

데이터 베이스 설계 단계

1. 요구사항 분석
    - 조직의 구성원들이 데이터 베이스를 사용하는 용도를 파악
    - 다양한 요구사항을 수집하고 이를 분석한 결과를 요구사항 명세서로 작성

2. 개념적 설계
    - 요구 사항을 조금 더 정형화하기 위해 사용
    - 중요한 데이터 요소와 데이터 요소 간의 관계를 표현
    - 일반적으로 E-R 모델을 많이 사용 하며, 데이터 요소 간의 관계를 E-R 다이어 그램으로 표현
    - 개념적 데이터 모델로 결과물을 개념적 구조, 개념적 스키마 라고 함. (개념적 스키마=전체DB 구조)
    - 결과물 : ERD (Entity-Relationship Diagram)
        - ERD 기본 개념
            - Entity : 개체 (테이블)
            - Attribute : 속성
            - UID : 식별자 (키)
            - Relationship : 관계

3. 논리적 설계
    - 개념적 설계 단계에서 설계한 개념적 구조를 기반으로 논리적 구조를 설계
    - ERD를 릴레이션(테이블) 스키마로 변환하여 DBMS가 처리할 수 있도록 하는 것이 논리적 설계단계의 주 작업
    - 결과물 : 관계형 데이터 모델

4. 물리적 설계
    - 저장 장치에 적합한 레코드 방식과 인덱스 등의 구조를 설계
    - 물리적 구조를 물리적 스키마/내부 스키마 라고 함
    - 결과물 : SQL

5. 구현
    - DBMS에 SQL로 작성한 명령문을 실행하여 데이터베이스를 실제로 생성

외래키 생성 방법 및 옵션

- 왜래키 생성
    - create table [테이블명] (

      [컬럼명] [데이터타입], …… ,

      constraint [제약 조건 이름] ---- fk를 이름에 포함 시켜주는 것이 좋음

      foreign key (컬럼명)

references [부모테이블명] (컬럼명) ---- 컬럼은 부모테이블의 기본키

on delete [옵션] ---- 값이 삭제 된 경우

on update [옵션] ---- 값이 변경된 경우

);

- 옵션
  - cascade : 부모데이터가 삭제/변경 시 자식 데이터도 동시 삭제/변경
  - restrct : 자식 테이블에 데이터가 남아있는 경우 (참조된 값)
  - set null : 부모 테이블에 값이 삭제 시 자식 테이블의 값을 null로 처리
  - set default : 부모 테이블에 값이 삭제 시 자식 테이블의 값을 기본 값으로 변경

```
mysql> use test;
Database changed
mysql> create table customer (
    -> c_id char(30) primary key comment '아이디',
    -> c_name char(30) not null comment '이름',
    -> c_age tinyint unsigned not null comment '나이',
    -> rank enum ('GOLD', 'SILVER', 'BRONZE', 'NO HUMAN') not null default 'NO
HUMAN' comment '등급'
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> create table ord (
    -> o_id int auto_increment primary key comment '주문번호',
    -> o_name char(30) not null comment '주분고객',
    -> o_prod varchar(50) not null comment '상품평',
    -> o_date datetime not null comment '주문일시',
    -> constraint ord_fk_id
    -> foreign key (o_name)
    -> references customer (c_id)
    -> on delete restrict
    -> on update restrict
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> desc ord;
+--------+-------------+------+-----+---------+----------------+
| Field  | Type        | Null | Key | Default | Extra          |
+--------+-------------+------+-----+---------+----------------+
| o_id   | int(11)     | NO   | PRI | NULL    | auto_increment |
| o_name | char(30)    | NO   | MUL | NULL    |                |
| o_prod | varchar(50) | NO   |     | NULL    |                |
| o_date | datetime    | NO   |     | NULL    |                |
+--------+-------------+------+-----+---------+----------------+
4 rows in set (0.01 sec)


mysql> insert into customer (c_id, c_name, c_age)
    -> values ('kim', '김영수', 28);
Query OK, 1 row affected (0.00 sec)

mysql> insert into customer (c_id, c_name, c_age) values ('ari','아리수','28');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into customer (c_id,c_name,c_age) values ('anzi','안지환',28);
Query OK, 1 row affected (0.00 sec)

mysql> select * from customer; // 이게 부모테이블
+------+-----------+-------+----------+
| c_id | c_name    | c_age | rank     |
+------+-----------+-------+----------+
| anzi | 안지환    |    28 | NO HUMAN |
| ari  | 아리수    |    28 | NO HUMAN |
| kim  | 김영수    |    28 | NO HUMAN |
+------+-----------+-------+----------+
3 rows in set (0.00 sec)

mysql> insert into ord (o_name, o_prod, o_date) values ('ari','물',now());
Query OK, 1 row affected (0.01 sec)

mysql> insert into ord (o_name, o_prod, o_date) values ('anzi','음료수',now());
Query OK, 1 row affected (0.00 sec)

mysql> insert into ord (o_name, o_prod, o_date) values ('ktest','코코넛 음
료',now());
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint
fails (`test`.`ord`, CONSTRAINT `ord_fk_id` FOREIGN KEY (`o_name`) REFERENCES
`customer` (`c_id`)) // 부모테이블에 있는 값만 들어갈 수 있기 때문에 오류뜸!!!
mysql>

mysql> update customer set c_id='kill' where c_id='kim';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from customer;
+------+-----------+-------+----------+
| c_id | c_name    | c_age | rank     |
+------+-----------+-------+----------+
| anzi | 안지환    |    28 | NO HUMAN |
| ari  | 아리수    |    28 | NO HUMAN |
| kill | 김영수    |    28 | NO HUMAN |
+------+-----------+-------+----------+
3 rows in set (0.00 sec)

mysql> update customer set c_id='muri' where c_id='ari';
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key
constraint fails (`test`.`ord`, CONSTRAINT `ord_fk_id` FOREIGN KEY (`o_name`)
REFERENCES `customer` (`c_id`))  // 외래키 때문에 오류남 ->  ari 값이 자식테이블 참조
하고 있기 때문

왜래키가 적용된걸 지우고싶어!
mysql> alter table ord drop foreign key ord_fk_id;
Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0


mysql> alter table ord add
    -> constraint ord_fk_id
    -> foreign key (o_name)
    -> references customer (c_id)
    -> on delete cascade
```

```
    -> on update cascade;
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

이제는 없데이트 잘됨 ari -> muri
mysql> select * from customer;
+------+-----------+-------+----------+
| c_id | c_name    | c_age | rank     |
+------+-----------+-------+----------+
| anzi | 안지환    |    28 | NO HUMAN |
| ari  | 아리수    |    28 | NO HUMAN |
| kill | 김영수    |    28 | NO HUMAN |
+------+-----------+-------+----------+
3 rows in set (0.00 sec)

mysql> update customer set c_id='muri' where c_id='ari';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from customer;
+------+-----------+-------+----------+
| c_id | c_name    | c_age | rank     |
+------+-----------+-------+----------+
| anzi | 안지환    |    28 | NO HUMAN |
| kill | 김영수    |    28 | NO HUMAN |
| muri | 아리수    |    28 | NO HUMAN |
+------+-----------+-------+----------+
3 rows in set (0.00 sec)

여기도 같이 바뀜 부모가 변경되면 cascade도 같이 변경되기 때문
mysql> select * from ord;
+------+--------+-----------+---------------------+
| o_id | o_name | o_prod    | o_date              |
+------+--------+-----------+---------------------+
|    1 | muri   | 물        | 2020-08-28 10:25:27 |
|    2 | anzi   | 음료수    | 2020-08-28 10:25:44 |
+------+--------+-----------+---------------------+
2 rows in set (0.00 sec)
```

JOIN

- select 컬럼명 from [테이블 A] [Join 종류] [테이블 B] on [조건];
- inner join (내부조인) -join
- left outer join (왼쪽 외부 조인) --- left join
- right outer join (오른쪽 외부 조인) --- right join

108p 테스트 테이블 생성 (JOIN)

[root@Linux-04 바탕화면]# mysql -uroot -pkeroro2424.

```
mysql> create database bns;
```

```
Query OK, 1 row affected (0.00 sec)
mysql> use bns;

mysql> create table market ( m_no int unsigned not null auto_increment primary
key,
    -> m_category varchar(32) not null ,
    -> m_name varchar(52) not null, m_seller varchar(32) not null );
Query OK, 0 rows affected (0.02 sec)

mysql> create table sword ( s_name varchar(52) not null primary key,
    -> s_level int unsigned not null,
    -> s_attack int unsigned not null);
Query OK, 0 rows affected (0.00 sec)

mysql> insert into market (m_no, m_category,m_name,m_seller) values ('','지팡
이','촉마지팡이','merry');
Query OK, 1 row affected, 1 warning (0.00 sec)

mysql> insert into market (m_no, m_category,m_name,m_seller) values ('','검','곤
륜검','evernick');
Query OK, 1 row affected, 1 warning (0.00 sec)

mysql>  insert into market (m_no, m_category,m_name,m_seller) values ('','기공
패','공륜기공패','jamienick');
Query OK, 1 row affected, 1 warning (0.01 sec)

mysql>  insert into market (m_no, m_category,m_name,m_seller) values ('','검','풍
뢰검','ruina');
Query OK, 1 row affected, 1 warning (0.00 sec)

mysql> insert into sword (s_name,s_level,s_attack) values ('곤륜검',50,540);
Query OK, 1 row affected (0.00 sec)

mysql> insert into sword (s_name,s_level,s_attack) values ('염화검',36,147);
Query OK, 1 row affected (0.00 sec)

mysql> insert into sword (s_name,s_level,s_attack) values ('요마검',20,64);
Query OK, 1 row affected (0.00 sec)

mysql> insert into sword (s_name,s_level,s_attack) values ('풍뢰검',45,263);
Query OK, 1 row affected (0.00 sec)

mysql> select * from market;
+------+------------+-----------------+-----------+
| m_no | m_category | m_name          | m_seller  |
+------+------------+-----------------+-----------+
|    1 | 지팡이     | 촉마지팡이      | merry     |
|    2 | 검         | 곤륜검          | evernick  |
|    3 | 기공패     | 공륜기공패      | jamienick |
|    4 | 검         | 풍뢰검          | ruina     |
+------+------------+-----------------+-----------+
4 rows in set (0.00 sec)

mysql> select * from sword;
+-----------+---------+----------+
| s_name    | s_level | s_attack |
+-----------+---------+----------+
| 곤륜검    |      50 |      540 |
```

```
| 염화검    |      36 |      147 |
| 요마검    |      20 |       64 |
| 풍뢰검    |      45 |      263 |
+-----------+---------+----------+
4 rows in set (0.00 sec)



mysql> select m_category, m_name, m_seller, s_name, s_level, s_attack from
market
    ->  inner join sword on m_name=s_name;
+------------+-----------+----------+-----------+---------+----------+
| m_category | m_name    | m_seller | s_name    | s_level | s_attack |
+------------+-----------+----------+-----------+---------+----------+
| 검         | 곤륜검    | evernick | 곤륜검    |      50 |      540 |
| 검         | 풍뢰검    | ruina    | 풍뢰검    |      45 |      263 |
+------------+-----------+----------+-----------+---------+----------+
2 rows in set (0.00 sec)

mysql> select m_category, m_name, m_seller, s_name, s_level, s_attack from
market
    -> inner join sword on m_name = s_name;
+------------+-----------+----------+-----------+---------+----------+
| m_category | m_name    | m_seller | s_name    | s_level | s_attack |
+------------+-----------+----------+-----------+---------+----------+
| 검         | 곤륜검    | evernick | 곤륜검    |      50 |      540 |
| 검         | 풍뢰검    | ruina    | 풍뢰검    |      45 |      263 |
+------------+-----------+----------+-----------+---------+----------+
2 rows in set (0.00 sec)

mysql> select m_category, m_name, m_seller, s_name, s_level, s_attack from
market, sword
    -> where m_name = s_name;
+------------+-----------+----------+-----------+---------+----------+
| m_category | m_name    | m_seller | s_name    | s_level | s_attack |
+------------+-----------+----------+-----------+---------+----------+
| 검         | 곤륜검    | evernick | 곤륜검    |      50 |      540 |
| 검         | 풍뢰검    | ruina    | 풍뢰검    |      45 |      263 |
+------------+-----------+----------+-----------+---------+----------+
2 rows in set (0.00 sec)
```

Union 유니온 114p

```
mysql> select m_category, m_name, m_seller from market
    -> union
    -> select s_name, s_level, s_attack from sword;
+------------+-----------------+-----------+
| m_category | m_name          | m_seller  |
+------------+-----------------+-----------+
| 지팡이     | 촉마지팡이      | merry     |
| 검         | 곤륜검          | evernick  |
| 기공패     | 공륜기공패      | jamienick |
| 검         | 풍뢰검          | ruina     |
```

```
| 곤륜검     | 50              | 540       |
| 염화검     | 36              | 147       |
| 요마검     | 20              | 64        |
| 풍뢰검     | 45              | 263       |
+------------+-----------------+-----------+
8 rows in set (0.00 sec)
```

115P staff 테이블 추가

```
mysql> create table staff (
    -> f_name varchar(52) primary key,
    -> f_level int unsigned not null,
    -> f_attack int unsigned not null
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> insert into staff values ('촉마지팡이', 50,311);
Query OK, 1 row affected (0.00 sec)

mysql> insert into staff values ('유성지팡이', 50,400);
Query OK, 1 row affected (0.00 sec)

mysql> select * from staff;
+-----------------+---------+----------+
| f_name          | f_level | f_attack |
+-----------------+---------+----------+
| 유성지팡이      |      50 |      400 |
| 촉마지팡이      |      50 |      311 |
+-----------------+---------+----------+
2 rows in set (0.00 sec)

mysql> select * from sword
    -> union
    -> select * from staff;
+-----------------+---------+----------+
| s_name          | s_level | s_attack |
+-----------------+---------+----------+
| 곤륜검          |      50 |      540 |
| 염화검          |      36 |      147 |
| 요마검          |      20 |       64 |
| 풍뢰검          |      45 |      263 |
| 유성지팡이      |      50 |      400 |
| 촉마지팡이      |      50 |      311 |
+-----------------+---------+----------+
6 rows in set (0.00 sec)
```

```
mysql> select * from market left join sword on m_name=s_name
    -> union
    -> select * from market right join sword on m_name=s_name;
+------+------------+-----------------+-----------+-----------+---------+-------
---+
| m_no | m_category | m_name          | m_seller  | s_name    | s_level |
s_attack |
```

```
+------+------------+-----------------+-----------+-----------+---------+-------
---+
|    1 | 지팡이     | 촉마지팡이      | merry     | NULL      |    NULL |     NULL
|
|    2 | 검         | 곤륜검          | evernick  | 곤륜검    |      50 |      540
|
|    3 | 기공패     | 공륜기공패      | jamienick | NULL      |    NULL |     NULL
|
|    4 | 검         | 풍뢰검          | ruina     | 풍뢰검    |      45 |      263
|
| NULL | NULL       | NULL            | NULL      | 염화검    |      36 |
147 |
| NULL | NULL       | NULL            | NULL      | 요마검    |      20 |
64 |
+------+------------+-----------------+-----------+-----------+---------+-------
---+
6 rows in set (0.01 sec)

mysql> select * from market left join sword on m_name=s_name
    -> union all
    -> select * from market right join sword on m_name=s_name;
+------+------------+-----------------+-----------+-----------+---------+-------
---+
| m_no | m_category | m_name          | m_seller  | s_name    | s_level |
s_attack |
+------+------------+-----------------+-----------+-----------+---------+-------
---+
|    1 | 지팡이     | 촉마지팡이      | merry     | NULL      |    NULL |     NULL
|
|    2 | 검         | 곤륜검          | evernick  | 곤륜검    |      50 |      540
|
|    3 | 기공패     | 공륜기공패      | jamienick | NULL      |    NULL |     NULL
|
|    4 | 검         | 풍뢰검          | ruina     | 풍뢰검    |      45 |      263
|
|    2 | 검         | 곤륜검          | evernick  | 곤륜검    |      50 |      540
|
| NULL | NULL       | NULL            | NULL      | 염화검    |      36 |
147 |
| NULL | NULL       | NULL            | NULL      | 요마검    |      20 |
64 |
|    4 | 검         | 풍뢰검          | ruina     | 풍뢰검    |      45 |      263
|
+------+------------+-----------------+-----------+-----------+---------+-------
---+
8 rows in set (0.00 sec)
```

Sub Query 서브쿼리

- 하나의 SQL문 안에 포함되어 있는 또 다른 SQL문
- 알려지지 않는 조건에 근거한 값들을 검색하는 select문을 작성하는데 유용
  (다른 테이블에 있는 값을 조건으로 부여)

- select, update, delete, insert 와 같은 DML문과 create table, view 에서 사용 가능
- 서브쿼리는 반드시 소괄호로 묶어야 함

  (select ....)
- 서브쿼리는 연산자 오른쪽에 위치
- 서브쿼리 내에서는 order by 문법 지원 안되지만 집계함수는 사용 가능
  - select [A.컬럼명] from [테이블A] where [A.컬럼명] = (select [B.컬럼명] from [테이블B] where [B.컬럼명])
- 서브쿼리 종류 = (select [B.컬럼명] from [테이블B] where [B.컬럼명]) 에 넘어오는 값에 따라
  - 단일 행 서브쿼리 : 서브쿼리 실행 결과가 1건 이하 (단일 컬럼)
  - 다중 행 서브쿼리 : 서브쿼리 실행 결과가 여러 건 (단일 컬럼)
  - 다중 열 서브쿼리 : 서브쿼리 실행 결과가 여러 컬럼

    로 나뉘어 진다

착용레벨 추가하기

```
mysql> select * from market;
+------+------------+-----------------+-----------+
| m_no | m_category | m_name          | m_seller  |
+------+------------+-----------------+-----------+
|    1 | 지팡이     | 촉마지팡이      | merry     |
|    2 | 검         | 곤륜검          | evernick  |
|    3 | 기공패     | 공륜기공패      | jamienick |
|    4 | 검         | 풍뢰검          | ruina     |
+------+------------+-----------------+-----------+
4 rows in set (0.00 sec)

mysql> alter table market add m_level int unsigned not null after m_name;
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> select * from market;
+------+------------+-----------------+---------+-----------+
| m_no | m_category | m_name          | m_level | m_seller  |
+------+------------+-----------------+---------+-----------+
|    1 | 지팡이     | 촉마지팡이      |       0 | merry     |
|    2 | 검         | 곤륜검          |       0 | evernick  |
|    3 | 기공패     | 공륜기공패      |       0 | jamienick |
|    4 | 검         | 풍뢰검          |       0 | ruina     |
+------+------------+-----------------+---------+-----------+
4 rows in set (0.00 sec)

mysql> update market set m_level=45 where m_name='풍뢰검';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from market;
+------+------------+-----------------+---------+-----------+
| m_no | m_category | m_name          | m_level | m_seller  |
+------+------------+-----------------+---------+-----------+
|    1 | 지팡이     | 촉마지팡이      |      50 | merry     |
|    2 | 검         | 곤륜검          |      50 | evernick  |
```

```
|    3 | 기공패     | 공륜기공패      |      50 | jamienick |
|    4 | 검         | 풍뢰검          |      45 | ruina     |
+------+------------+-----------------+---------+-----------+
4 rows in set (0.00 sec)

mysql> select * from sword
    -> where s_name=(select m_name from market where m_seller='evernick');
+-----------+---------+----------+
| s_name    | s_level | s_attack |
+-----------+---------+----------+
| 곤륜검    |      50 |      540 |
+-----------+---------+----------+
1 row in set (0.00 sec)

mysql> select m_name from market where m_seller='evernick'; //단일행 서브쿼리
+-----------+
| m_name    |
+-----------+
| 곤륜검    |
+-----------+
1 row in set (0.00 sec)

mysql> select * from market;
+------+------------+-----------------+---------+-----------+
| m_no | m_category | m_name          | m_level | m_seller  |
+------+------------+-----------------+---------+-----------+
|    1 | 지팡이     | 촉마지팡이      |      50 | merry     |
|    2 | 검         | 곤륜검          |      50 | evernick  |
|    3 | 기공패     | 공륜기공패      |      50 | jamienick |
|    4 | 검         | 풍뢰검          |      45 | ruina     |
+------+------------+-----------------+---------+-----------+
4 rows in set (0.00 sec)

mysql> select * from sword where
    -> s_name in (select m_name from market where m_category='검');
+-----------+---------+----------+
| s_name    | s_level | s_attack |
+-----------+---------+----------+
| 곤륜검    |      50 |      540 |
| 풍뢰검    |      45 |      263 |
+-----------+---------+----------+
2 rows in set (0.00 sec)

여러개 쓸 때는 in () 이용하기!!!!!

mysql> select * from sword where s_name in ('곤륜검','풍뢰검');
+-----------+---------+----------+
| s_name    | s_level | s_attack |
+-----------+---------+----------+
| 곤륜검    |      50 |      540 |
| 풍뢰검    |      45 |      263 |
+-----------+---------+----------+
2 rows in set (0.00 sec)

in 안쓰면 오류뜸

mysql> select * from sword where s_name = '곤륜검','풍뢰검';
```

```
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 ''풍뢰
검'' at line 1


mysql> select * from sword where
    -> (s_name, s_level) in (select m_name, max(m_level) from market where
m_category='검');
+-----------+---------+----------+
| s_name    | s_level | s_attack |
+-----------+---------+----------+
| 곤륜검    |      50 |      540 |
+-----------+---------+----------+
1 row in set (0.01 sec)

mysql> select m_name, max(m_level) from market where m_category='검';
+-----------+--------------+
| m_name    | max(m_level) |
+-----------+--------------+
| 곤륜검    |           50 |
+-----------+--------------+
1 row in set (0.00 sec)

-> 다중 열 서브쿼리
```

실습 서브쿼리

```
mysql> use TestDB;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+------------------+
| Tables_in_TestDB |
+------------------+
| buyTbl           |
| userTbl          |
+------------------+
2 rows in set (0.00 sec)

mysql> select * from buyTbl;
+-----+--------+-----------+-----------+-------+--------+
| num | userid | prodname  | groupname | price | amount |
+-----+--------+-----------+-----------+-------+--------+
|   1 | PJY    | 운동화    | NULL      |    45 |      2 |
|   2 | PJY    | 노트북    | 전자      |  1500 |      1 |
|   3 | LCS    | 모니터    | 전자      |   300 |      1 |
|   4 | CJC    | 노트북    | 전자      |   300 |      5 |
|   5 | PJY    | 청바지    | 의류      |    75 |      3 |
|   6 | CJC    | 메모리    | 전자      |   120 |     10 |
|   7 | AJH    | 책        | 서적      |    23 |      5 |
|   8 | SKH    | 책        | 서적      |    23 |      2 |
|   9 | SKH    | 청바지    | 의류      |    75 |      1 |
```

```
|  10 | CJC    | 운동화    | NULL      |    45 |      2 |
|  11 | SKH    | 책        | 서적      |    23 |      1 |
|  12 | CJC    | 운동화    | NULL      |    45 |      2 |
+-----+--------+-----------+-----------+-------+--------+
12 rows in set (0.00 sec)
```

1. 책을 구매한 사람 중 5개를 구매한 사람의 이름을 출력하여라

```
mysql> select name from userTbl
    -> where userid=(select userid from buyTbl where prodname='책' and
amount=5);
+-----------+
| name      |
+-----------+
| 안정환    |
+-----------+
1 row in set (0.00 sec)
```

2. 1986년생 이면서 경기에 살고 있는 사람의 구매 물품중에 50원 보다 비싼 물건의 물건이름 및 가격 정보를 출력하라.

```
mysql> select prodname, price from buyTbl
    -> where userid=(select userid from userTbl where birthyear=1986 and
addr='경기')
    -> and price > 50;
+-----------+-------+
| prodname  | price |
+-----------+-------+
| 노트북    |  1500 |
| 청바지    |    75 |
+-----------+-------+
2 rows in set (0.00 sec)
```

3. 최진철이 구매한 메모리 가격보다 작은 물건의 이름 및 가격을 출력하고 가격 별로 내림차순으로 정렬하여 출력하여라.

```
mysql> select prodname, price from buyTbl
    -> where price < (select price from buyTbl
    -> where userid = (select userid from userTbl where name='최진철')
    -> and prodname='메모리') order by price desc;
+-----------+-------+
| prodname  | price |
+-----------+-------+
| 청바지    |    75 |
| 청바지    |    75 |
| 운동화    |    45 |
| 운동화    |    45 |
| 운동화    |    45 |
| 책        |    23 |
```

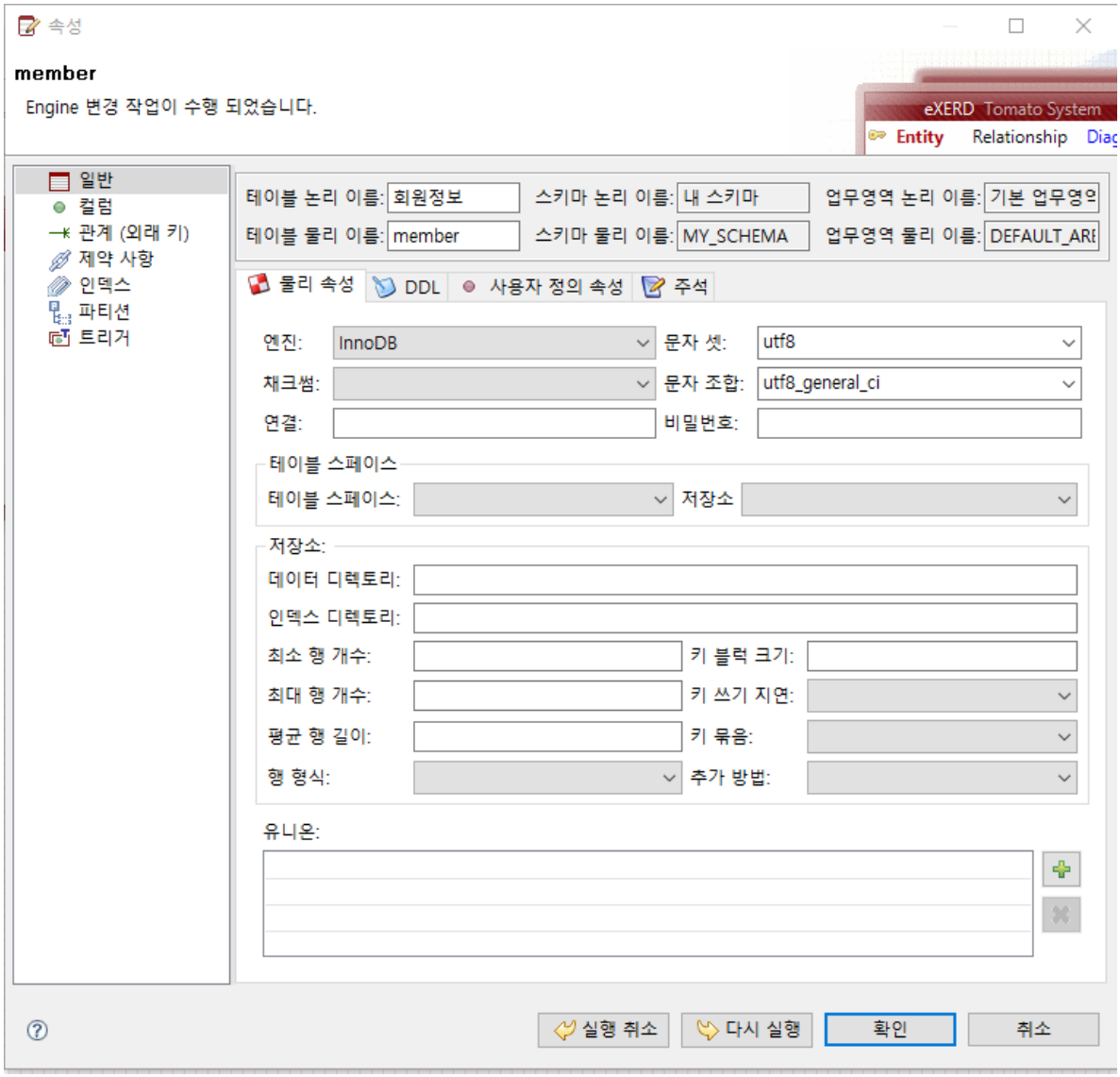
```
| 책         |    23 |
| 책         |    23 |
+-----------+-------+
8 rows in set (0.00 sec)

중복 제거해주기

mysql> select distinct prodname, price from buyTbl
    -> where price < (select price from buyTbl    -> where userid = (select
userid from userTbl where name='최진철')    -> and prodname='메모리') order
by price desc;+-----------+-------+
| prodname  | price |
+-----------+-------+
| 청바지     |    75 |
| 운동화     |    45 |
| 책         |    23 |
+-----------+-------+
3 rows in set (0.00 sec)
```

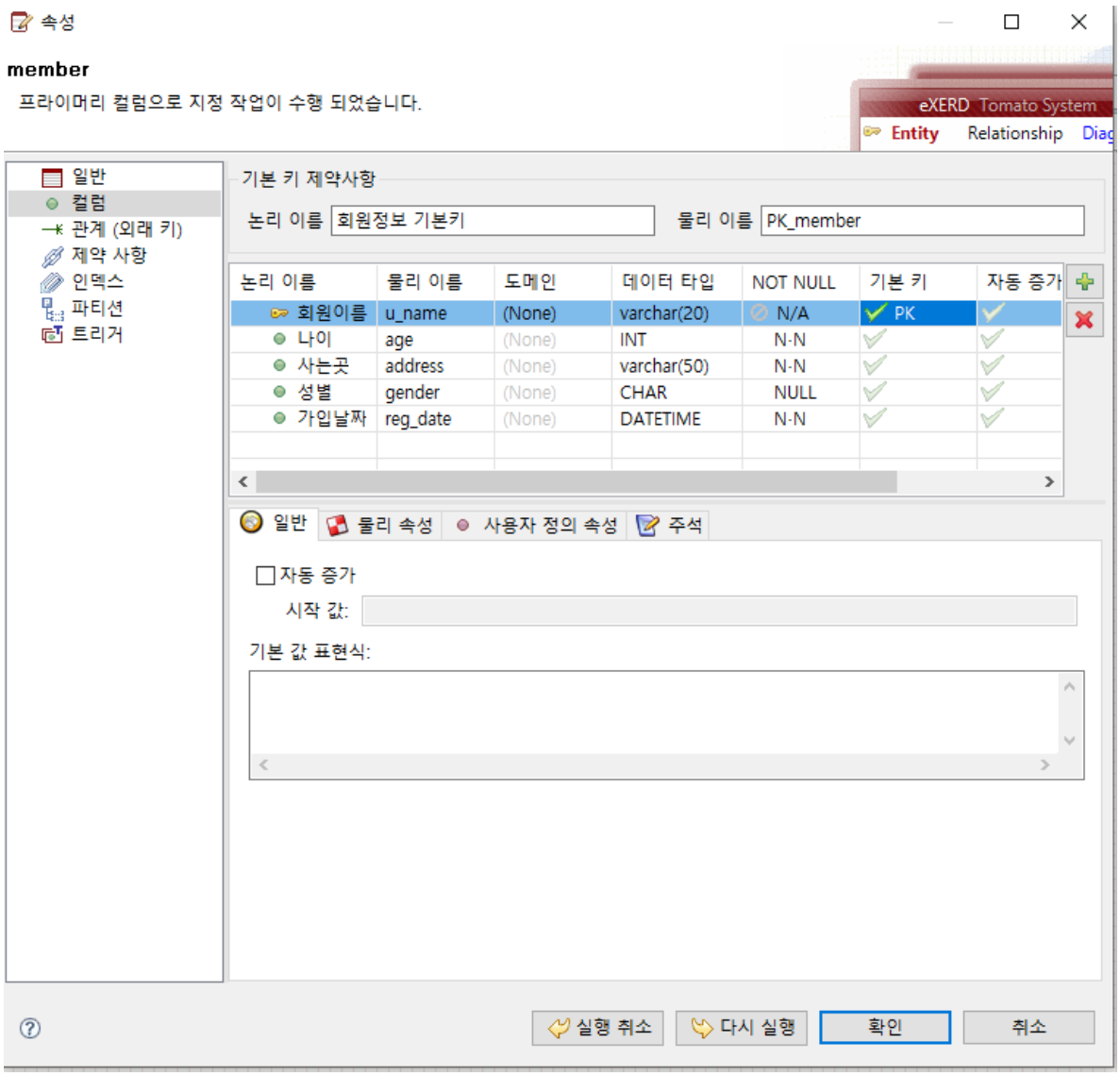
ERD 만들기

파일 -> 새로만들기 -> exERD -> 대상DBMS : Mysql 선택 -> 파일 이름 : exam.exerd

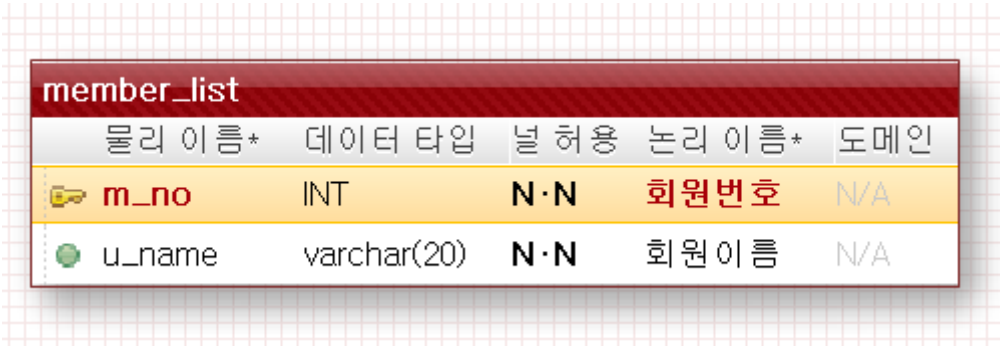
오른쪽 new Project -> exeam.erd 클릭

3 누르고 클릭 -> p -> 컬럼추가

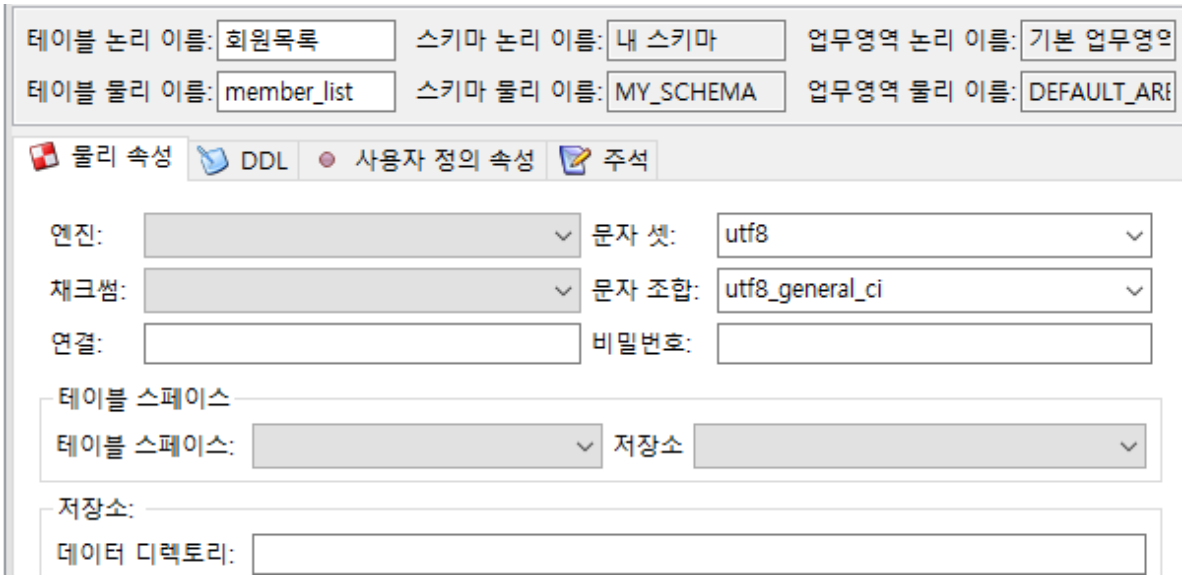

속성
member
Engine 변경 작업이 수행 되었습니다.
eXERD Tomato System
Entity
Relationship
일반
컬럼
관계 (외래 키)
제약 사항
인덱스
파티션
트리거
테이블 논리 이름: 회원정보
스키마 논리 이름: 내 스키마
업무영역 논리 이름: 기본 업무영역
테이블 물리 이름: member
스키마 물리 이름: MY_SCHEMA
업무영역 물리 이름: DEFAULT_ARE
물리 속성
DDL
사용자 정의 속성
주석
엔진: InnoDB
문자 셋: utf8
채크썸:
문자 조합: utf8_general_ci
연결:
비밀번호:
테이블 스페이스
테이블 스페이스:
저장소
저장소:
데이터 디렉토리:
인덱스 디렉토리:
최소 행 개수:
키 블럭 크기:
최대 행 개수:
키 쓰기 지연:
평균 행 길이:
키 묶음:
행 형식:
추가 방법:
유니온:
실행 취소
다시 실행
확인
취소

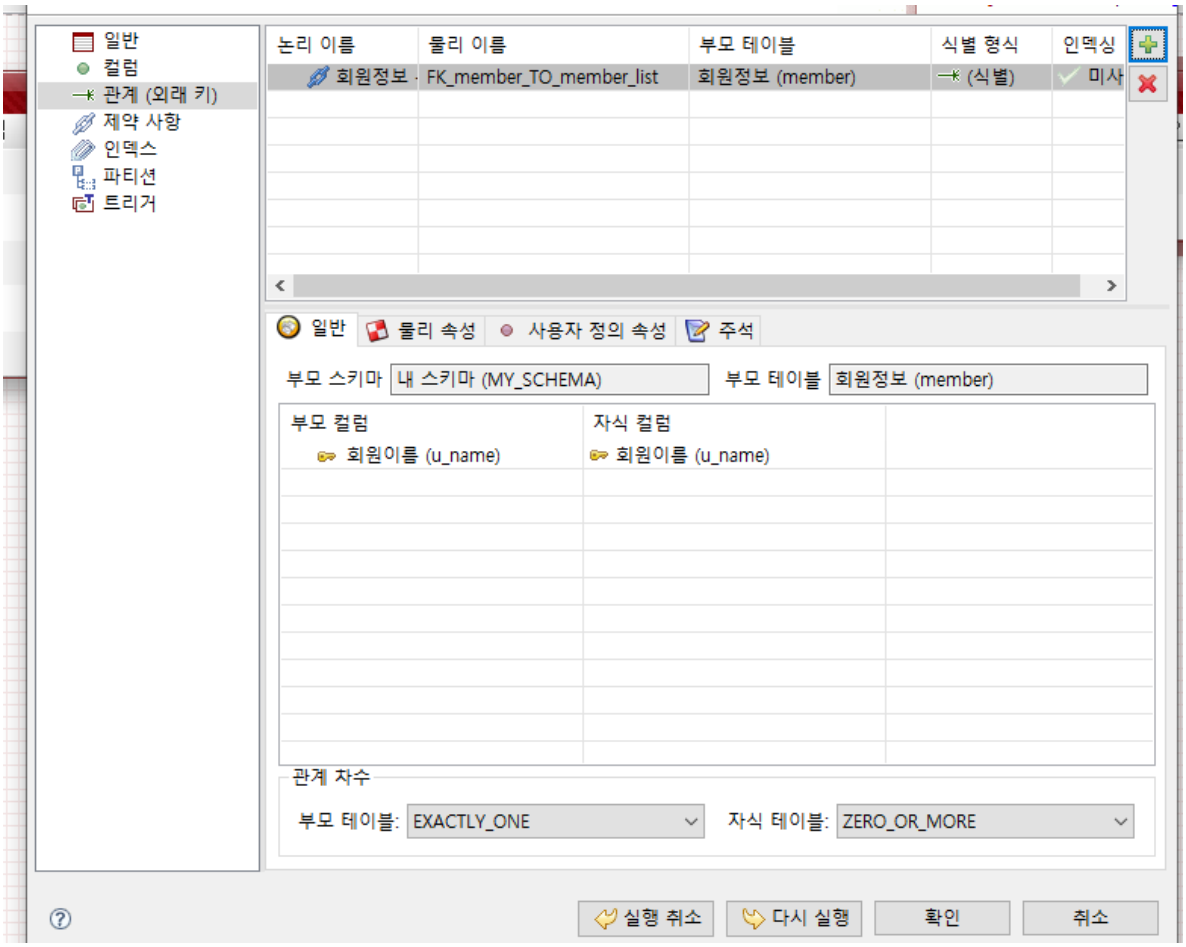


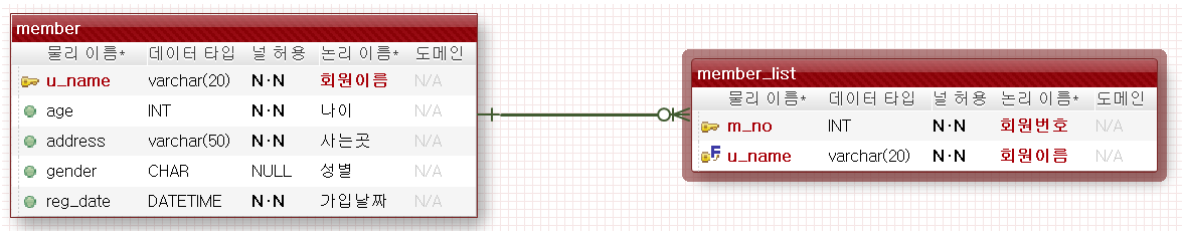
하나더 추가



외래키로 만들기

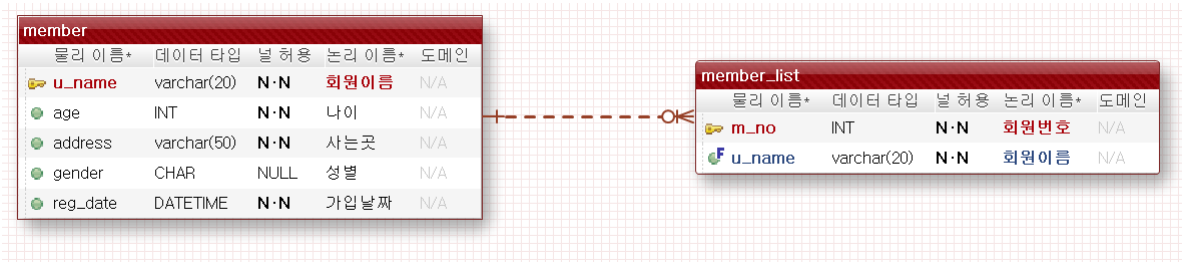






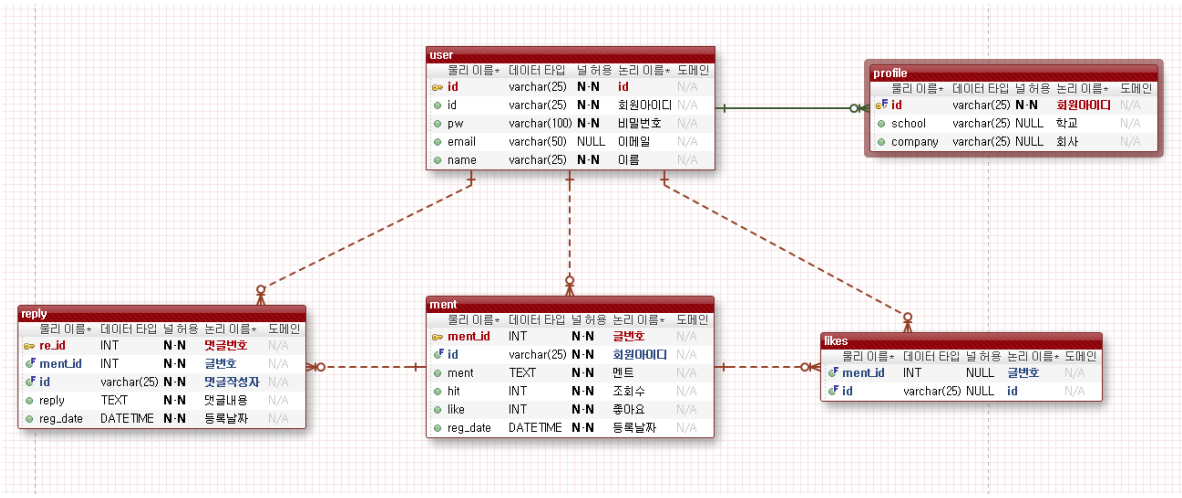
근데 보면 u_name이 기본키면서 외래키로 잡힘 이러면 안됨

비식별단계는 점선 식별단계는 실선

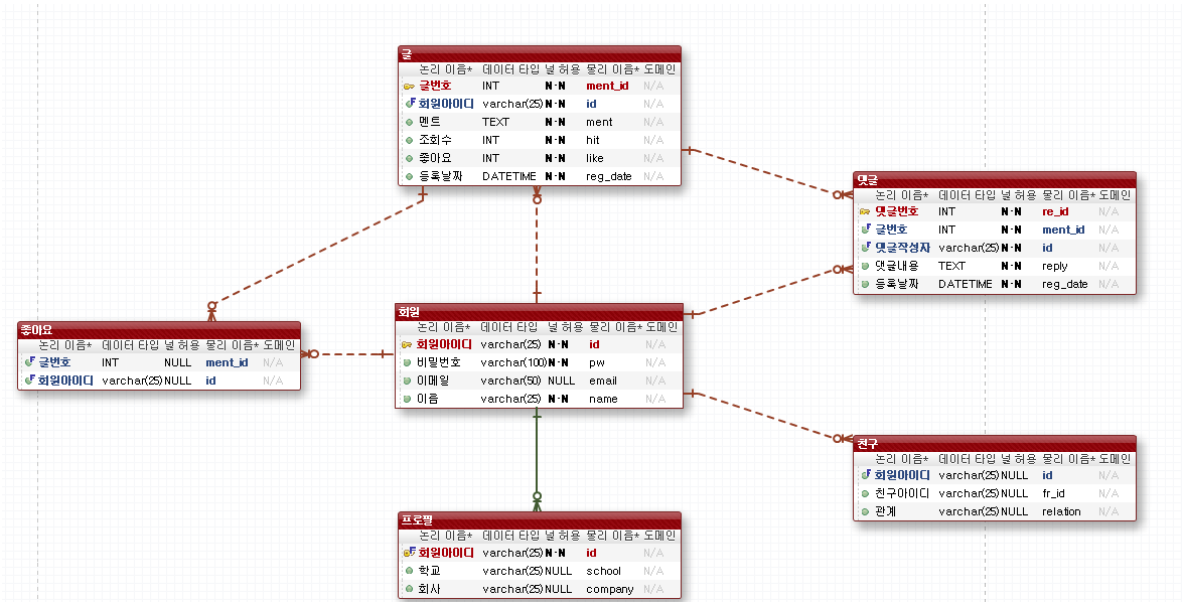


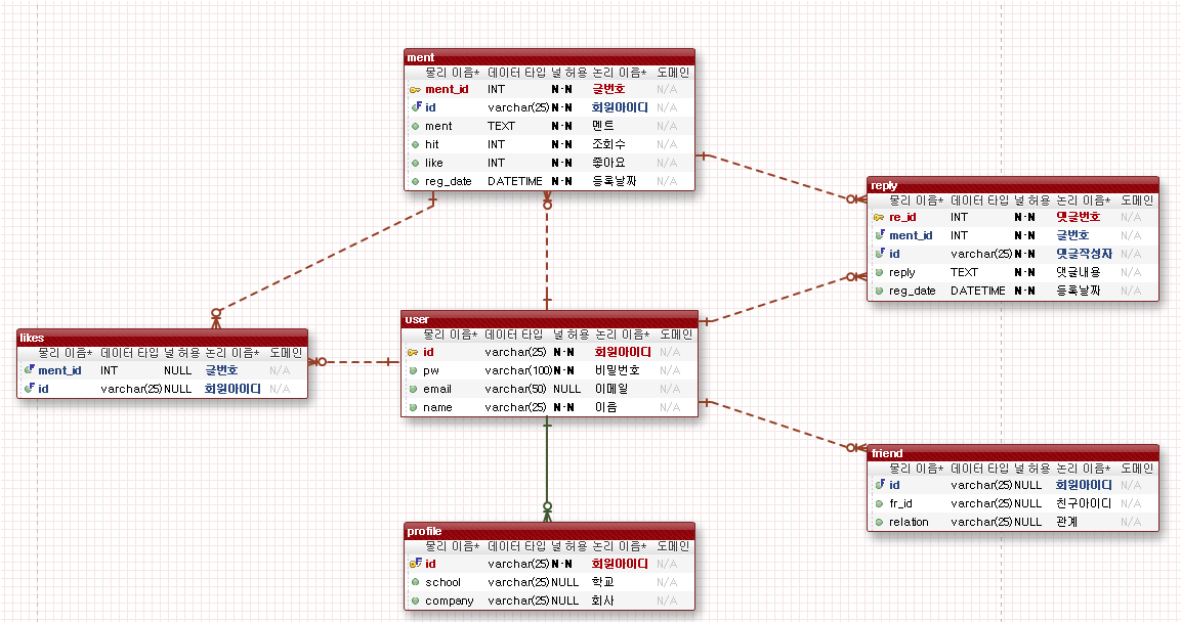
실습 SNS DB 설계

1. 회원가입은 id, 비밀번호, email, 이름을 입력하여 진행 한다.
2. 선택적으로 프로필을 등록할 수 있다.
3. 프로필 등록에는 다니는 학교와 회사를 입력해야 한다.
4. 글을 작성 할 수 있다. 간단한 멘트를 남길 수 있다.
5. 이 글에는 등록날짜, 조회 수 , 좋아요 숫자를 보여준다.
6. "좋아요" 는 로그인 한 회원이 아이디 별로 한번만 가능하다
7. 해당 글에는 댓글을 달 수 있다.



친구테이블 만들고싶어



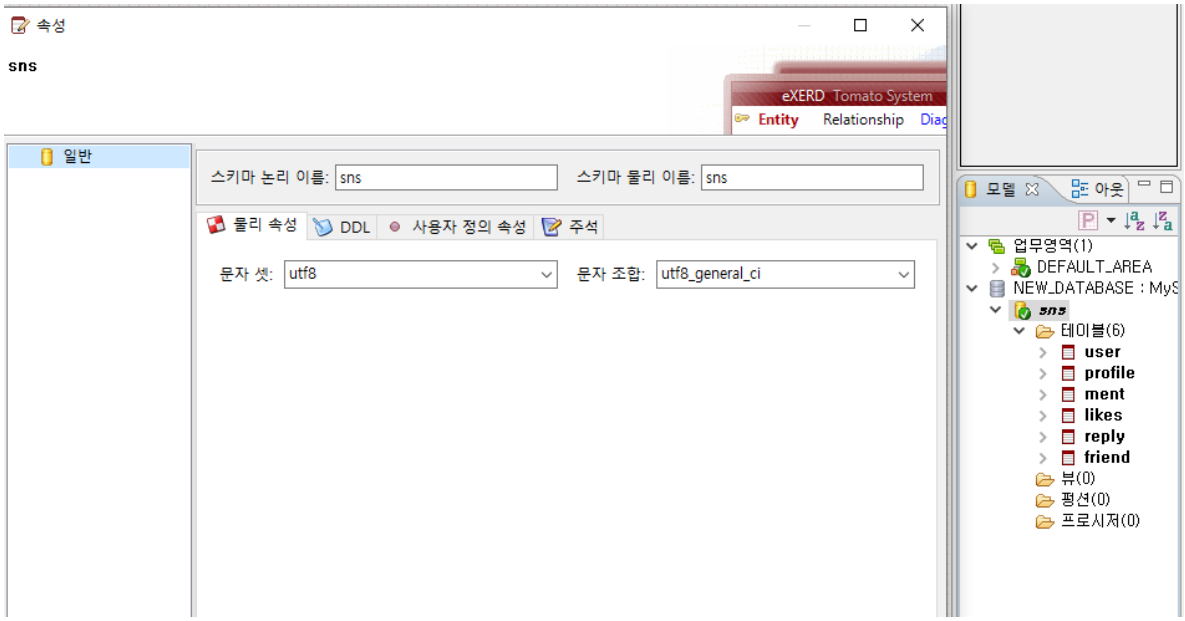


다 만들고 디비로 접근하기 -> 아직은 권한이 없어

우리 db 계정 봐봐야

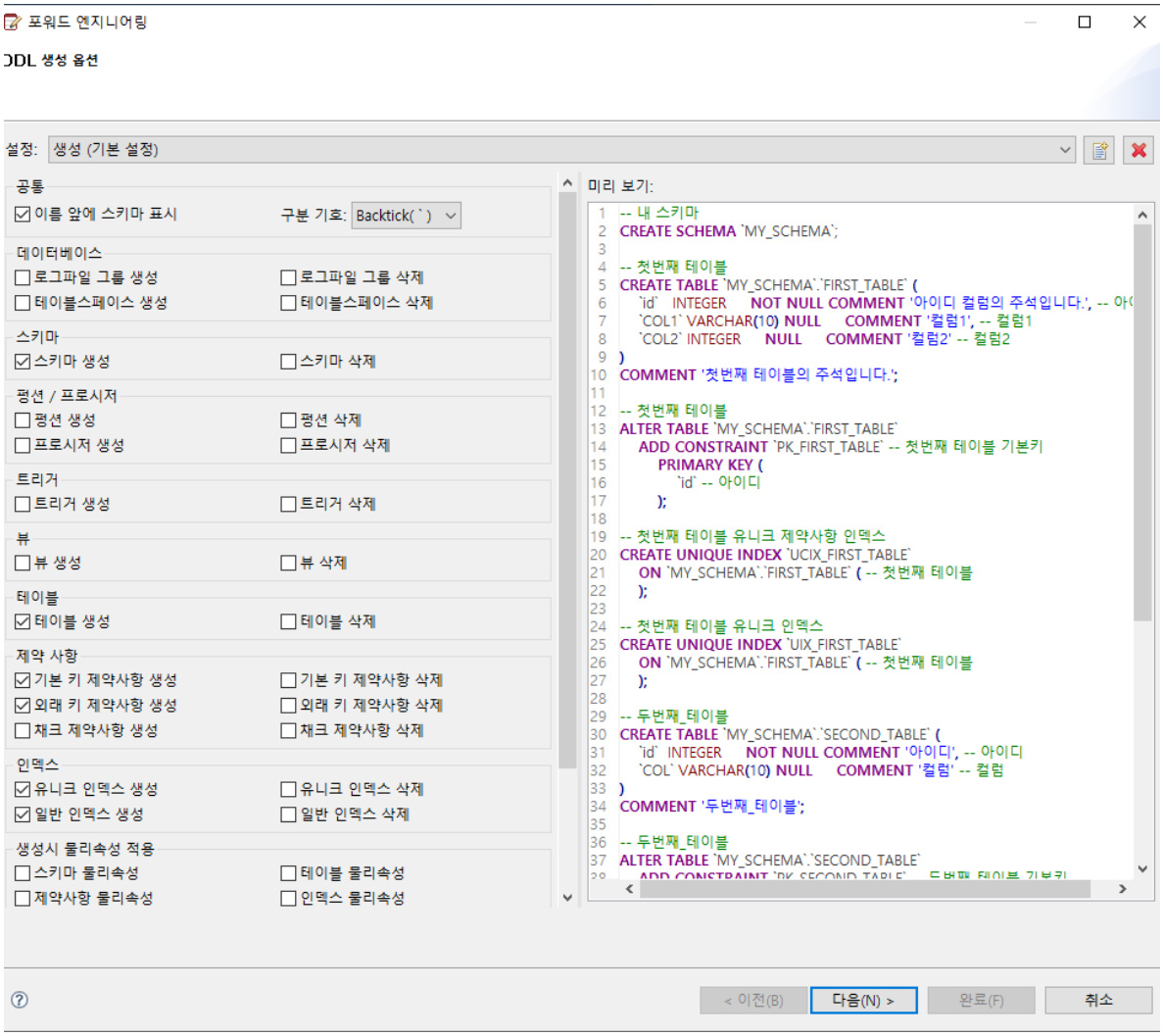
```
mysql> select user, host, password from mysql.user;
+-----------+---------------+-------------------------------------------+
| user      | host          | password                                  |
+-----------+---------------+-------------------------------------------+
| root      | localhost     | *A849201DB02204C54339B1AF6F0E3BEF14EE74BC |
| root      | linux-04      | *A849201DB02204C54339B1AF6F0E3BEF14EE74BC |
| root      | 127.0.0.1     | *A849201DB02204C54339B1AF6F0E3BEF14EE74BC |
| kuser1    | localhost     | *A4B6157319038724E3560894F7F932C8886EBFCF |
| root      | 100.100.100.% | *A4B6157319038724E3560894F7F932C8886EBFCF |
| testuser1 | localhost     | *A849201DB02204C54339B1AF6F0E3BEF14EE74BC |
| testuser2 | localhost     | *A849201DB02204C54339B1AF6F0E3BEF14EE74BC |
+-----------+---------------+-------------------------------------------+
7 rows in set (0.00 sec)

mysql> show grants for root@'100.100.100.%';
+-----------------------------------------------------------------------------------
----------------------------------------+
| Grants for root@100.100.100.%
                                        |
+-----------------------------------------------------------------------------------
----------------------------------------+
| GRANT ALL PRIVILEGES ON *.* TO 'root'@'100.100.100.%' IDENTIFIED BY PASSWORD
'*A4B6157319038724E3560894F7F932C8886EBFCF' |
+-----------------------------------------------------------------------------------
----------------------------------------+
1 row in set (0.00 sec)

-> 하나 있어서 접근 가능하겠다! 권한 따로 안줘도 됨 우리 Vmnet8 이 100.100.100.1 이라 가능
```

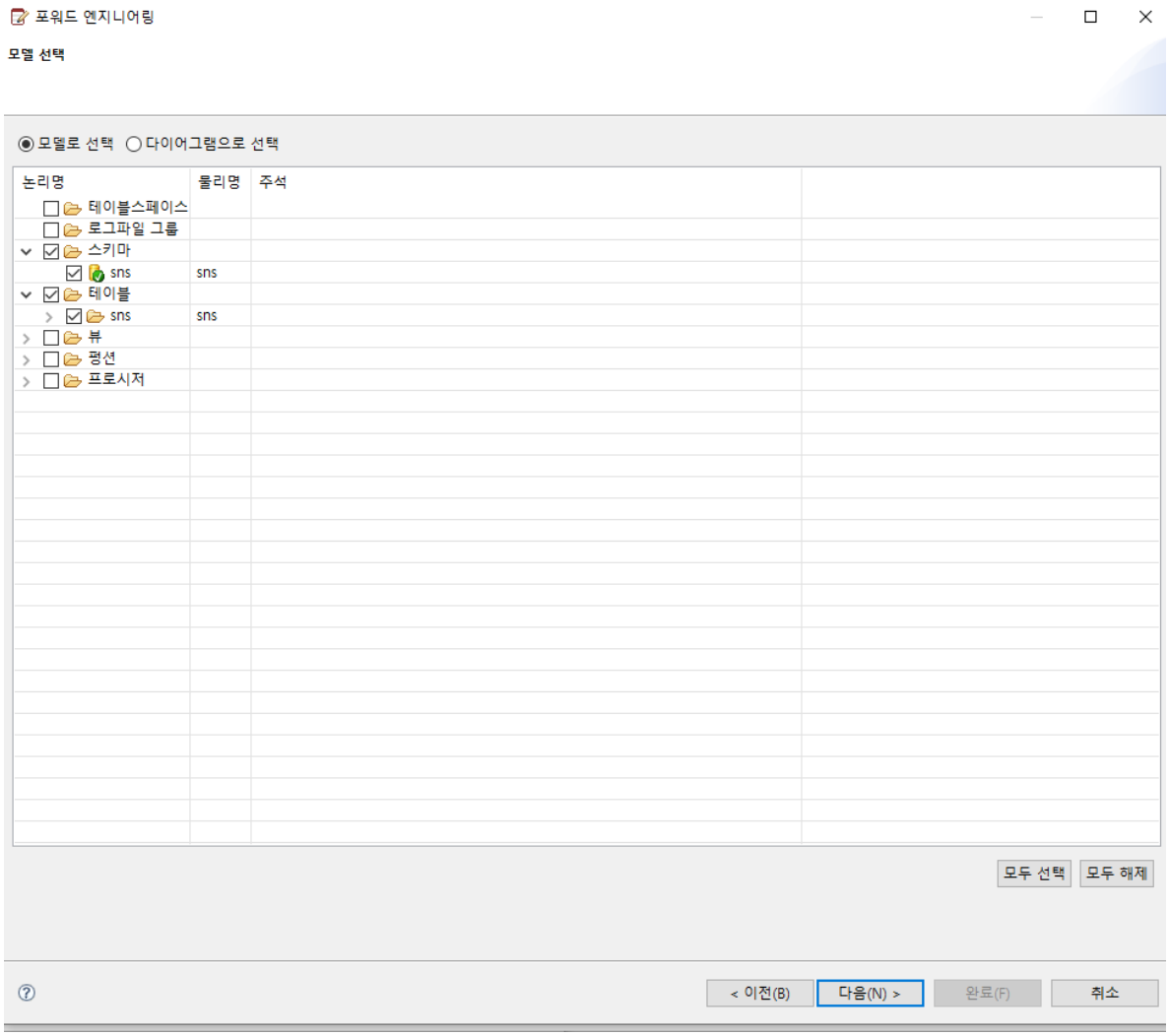
옆에 스키마 오른쪽마우스 -> 속성 -> 스키마 논리 이름이랑 문자셋 바꿔주기



eXERD - 포워드 엔지리어링

밑에 사진에서 스키마 생성 해제하슈!!!!!!!!!!

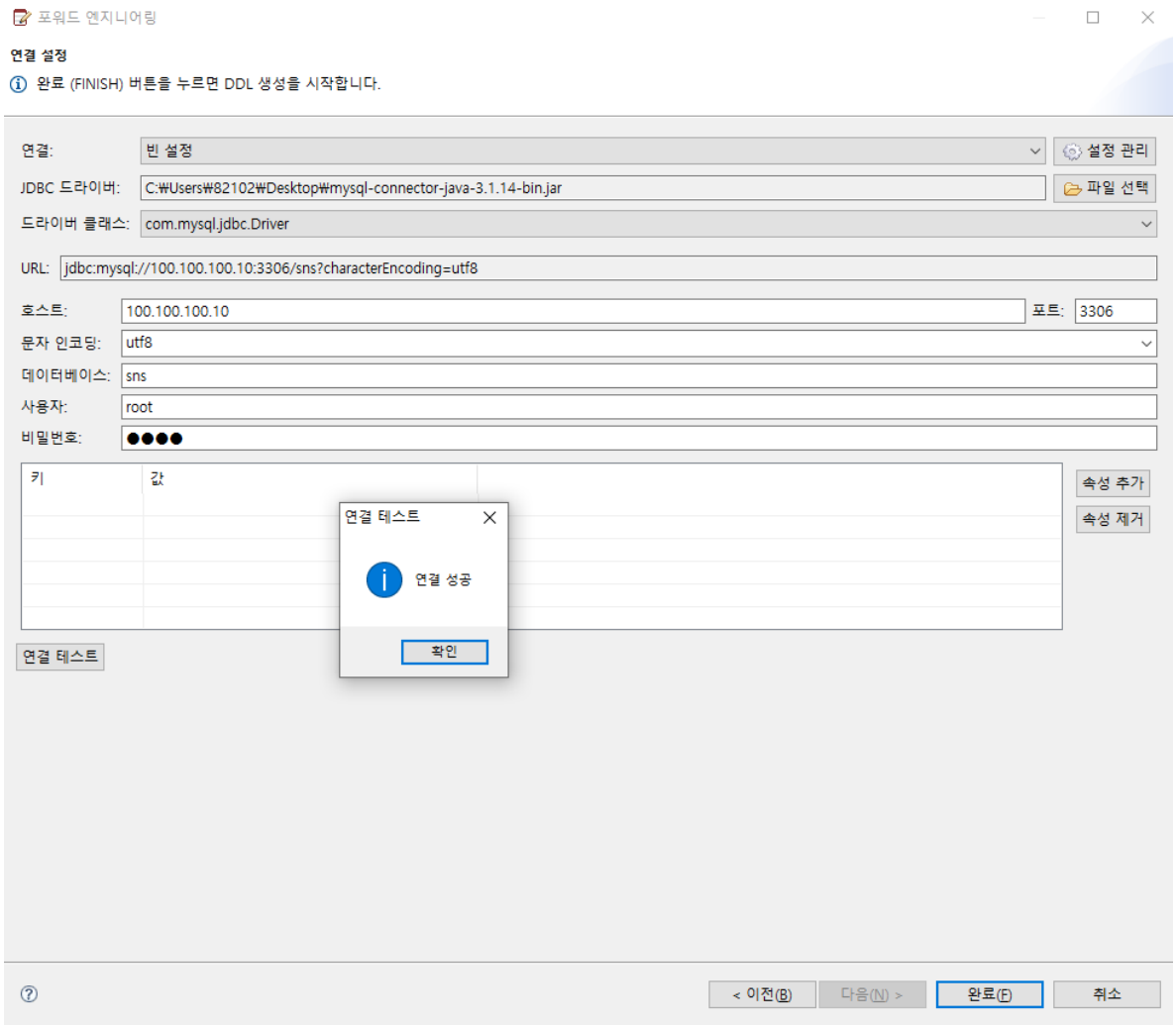




WebMail 가서 먼저 sns DB 만들어주고 오기!!!

```
mysql> create database sns;
Query OK, 1 row affected (0.01 sec)
```

밑에 사진대로 입력하고 연결테스트 클릭해보기 -> 성공 나오면 성공



다시 WebMail로 돌아와서 잘 되었는지 확인해보기

```
mysql> use sns;
Database changed

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| TestDB             |
| bns                |
| entest             |
| entest2            |
| khacademy          |
| mysql              |
| naver_db           |
| sns                |
| test               |
+--------------------+
10 rows in set (0.00 sec)

mysql> desc user;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | varchar(25)  | NO   | PRI | NULL    |       |
| pw    | varchar(100) | NO   |     | NULL    |       |
```

```
| email | varchar(50)  | YES  |     | NULL    |       |
| name  | varchar(25)  | NO   |     | NULL    |       |
+-------+--------------+------+-----+---------+-------+
4 rows in set (0.01 sec)

mysql> desc profile;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| id      | varchar(25) | NO   | PRI | NULL    |       |
| school  | varchar(25) | YES  |     | NULL    |       |
| company | varchar(25) | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> desc ment;
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| ment_id  | int(11)     | NO   | PRI | NULL    | auto_increment |
| id       | varchar(25) | NO   | MUL | NULL    |                |
| ment     | text        | NO   |     | NULL    |                |
| hit      | int(11)     | NO   |     | NULL    |                |
| like     | int(11)     | NO   |     | NULL    |                |
| reg_date | datetime    | NO   |     | NULL    |                |
+----------+-------------+------+-----+---------+----------------+
6 rows in set (0.01 sec)

mysql> desc likes;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| ment_id | int(11)     | YES  | MUL | NULL    |       |
| id      | varchar(25) | YES  | MUL | NULL    |       |
+---------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> show tables;
+---------------+
| Tables_in_sns |
+---------------+
| friend        |
| likes         |
| member        |
| member_list   |
| ment          |
| profile       |
| reply         |
| user          |
+---------------+
8 rows in set (0.00 sec)

mysql> desc member;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| u_name   | varchar(20) | NO   | PRI | NULL    |       |
| age      | int(11)     | NO   |     | NULL    |       |
```

```
| address  | varchar(50) | NO   |     | NULL    |       |
| gender   | char(1)     | YES  |     | NULL    |       |
| reg_date | datetime    | NO   |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

eXERD 에서 member 랑 member_list 지워줌~

mysql> show tables;
+---------------+
| Tables_in_sns |
+---------------+
| friend        |
| likes         |
| ment          |
| profile       |
| reply         |
| user          |
+---------------+
6 rows in set (0.00 sec)
```