



Database for SNS

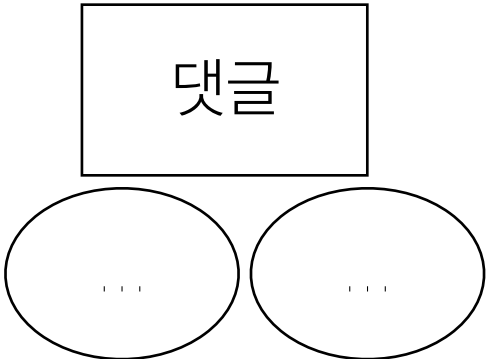
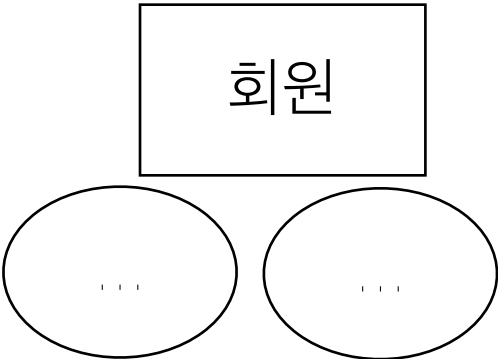
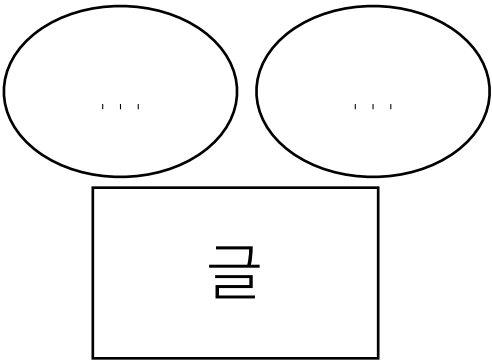
Database for SNS

글

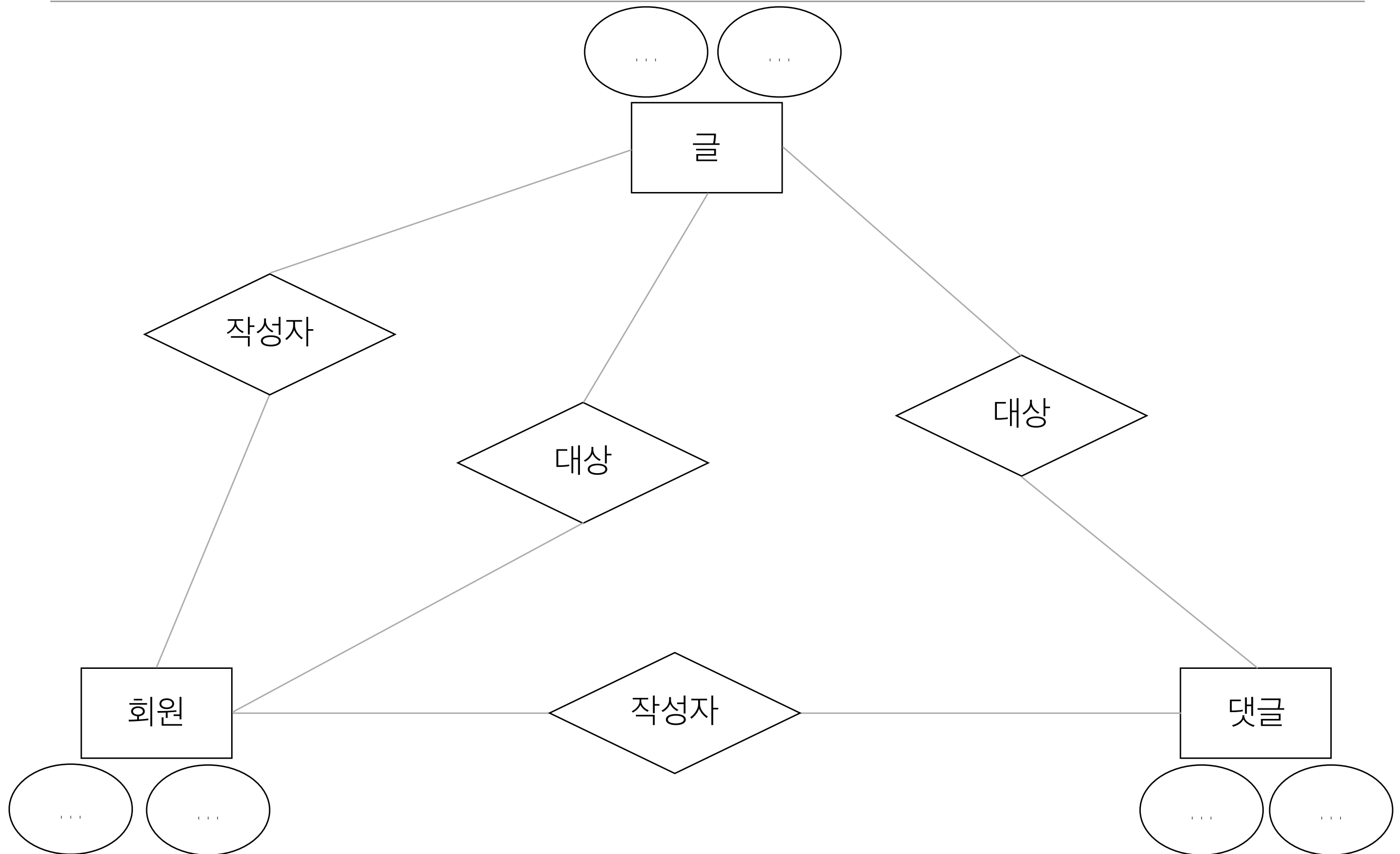
회원

댓글

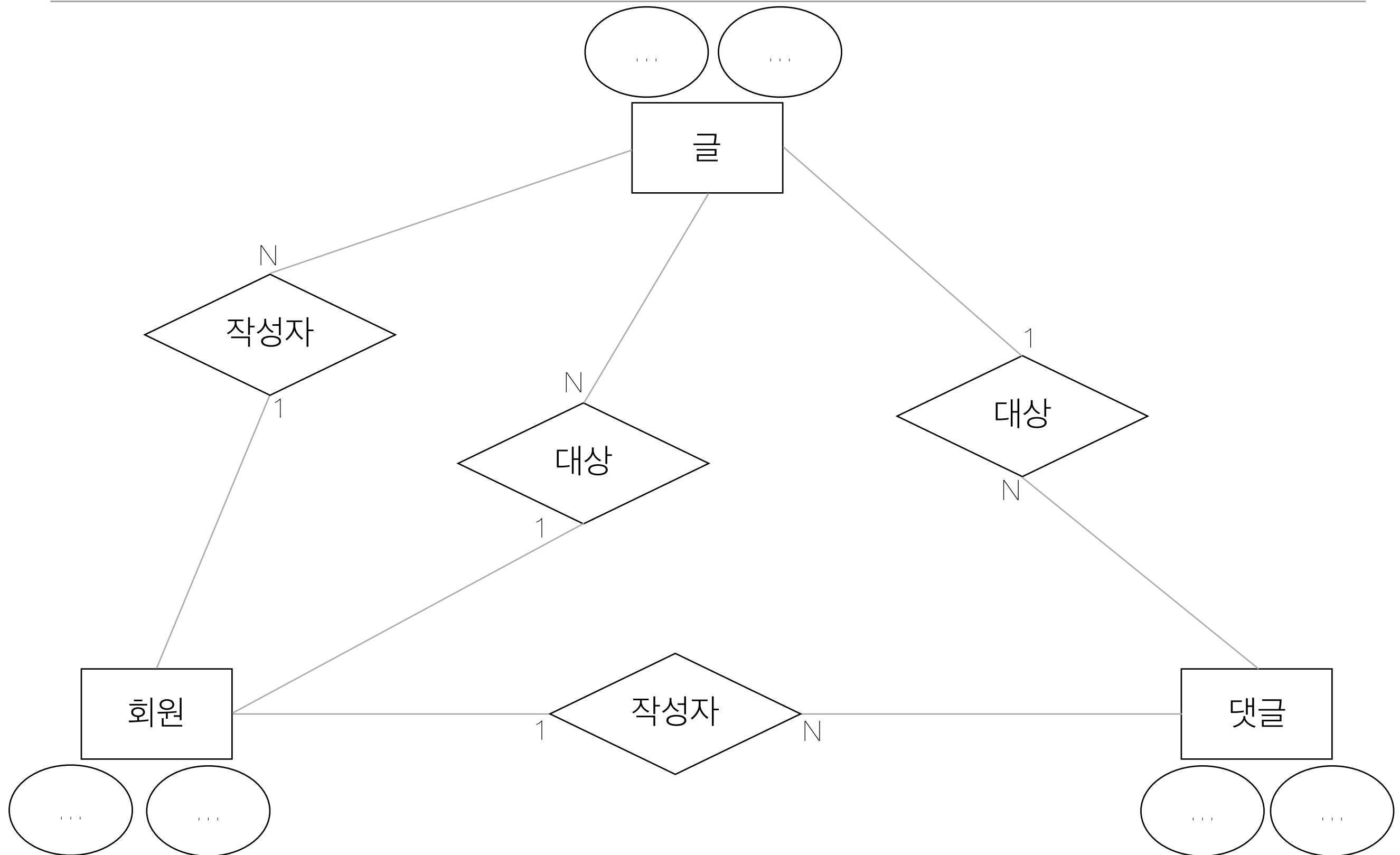
Database for SNS



Database for SNS



Database for SNS



Database for SNS

사용자

ID	이름	이메일	...
----	----	-----	-----

글

ID	내용	시간	...
----	----	----	-----

댓글

ID	내용	시간	...
----	----	----	-----

Database for SNS

사용자

ID	이름	이메일	...
----	----	-----	-----

글

ID	내용	시간	작성자ID	답벼락ID	...
----	----	----	-------	-------	-----

댓글

ID	내용	시간	글ID	작성자ID	...
----	----	----	-----	-------	-----



Database for SNS

Timeline | **우동** User List

홍지호

2014-08-06 14:58:53

×

메롱

Comments

홍지호

댓글 달기

우동 오올ㅋ

2014-08-06 15:05:04

홍지호 후후

2014-08-06 15:05:26

Database for SNS

Timeline | **우동**

User List

홍지호

2014-08-06 14:58:53

×

메롱

Comments

홍지호

댓글 달기

우동 오올ㅋ

2014-08-06 15:05:04

홍지호 후후

2014-08-06 15:05:26

Database for SNS

Timeline | **우동** User List

홍지호

2014-08-06 14:58:53

×

메롱

Comments

홍지호

댓글 달기

우동 오올ㅋ

2014-08-06 15:05:04

홍지호 후후

2014-08-06 15:05:26

Database for SNS

Timeline | **우동** User List

홍지호

2014-08-06 14:58:53

×

메롱

Comments

홍지호

댓글 달기

우동 오올ㅋ

2014-08-06 15:05:04

홍지호 후후

2014-08-06 15:05:26

Database for SNS

Timeline | **우동** User List

홍지호

2014-08-06 14:58:53

×

메롱

Comments

홍지호

댓글 달기

우동 오올ㅋ

2014-08-06 15:05:04

홍지호 후후

2014-08-06 15:05:26

Database for SNS

사용자

ID	이름	...
1	홍지호	...
2	우동	...
...

글

ID	내용	작성자ID	답벼락ID	...
1	메롱	1	2	...
...

댓글

ID	내용	작성자ID	글ID	...
1	오올ㅋ	2	1	...
2	후후	1	1	...
..

Database for SNS

사용자

ID	이름	...
1	홍지호	...
2	우동	...
...

글

ID	내용	작성자ID	답벼락ID	...
1	메롱	1	2	...
...

댓글

ID	내용	작성자ID	글ID	...
1	오올ㅋ	2	1	...
2	후후	1	1	...
..

Database for SNS

사용자

ID	이름	...
1	홍지호	...
2	우동	...
...

글

ID	내용	작성자ID	답벼락ID	...
1	메롱	1	2	...
...

댓글

ID	내용	작성자ID	글ID	...
1	오올ㅋ	2	1	...
2	후후	1	1	...
..

Database for SNS

사용자

ID	이름	...
1	홍지호	...
2	우동	...
...

글

ID	내용	작성자ID	답벼락ID	...
1	메롱	1	2	...
...

댓글

ID	내용	작성자ID	글ID	...
1	오올ㅋ	2	1	...
2	후후	1	1	...
..

Database for SNS

사용자

ID	이름	...
1	홍지호	...
2	우동	...
...

글

ID	내용	작성자ID	답벼락ID	...
1	메롱	1	2	...
...

댓글

ID	내용	작성자ID	글ID	...
1	오올ㅋ	2	1	...
2	후후	1	1	...
..

Database for SNS

사용자

ID	이름	...
1	홍지호	...
2	우동	...
...

글

ID	내용	작성자ID	답벼락ID	...
1	메롱	1	2	...
...

댓글

ID	내용	작성자ID	글ID	...
1	오올ㅋ	2	1	...
2	후후	1	1	...
..

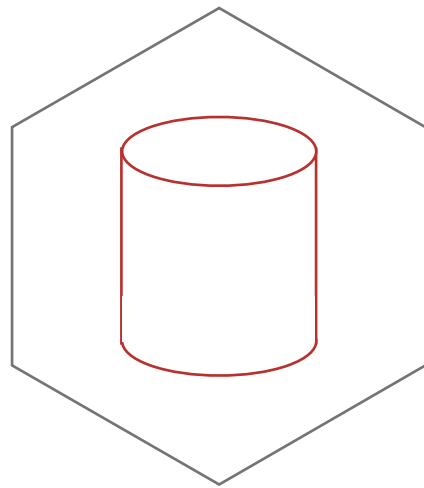
...

ORM

CRUD

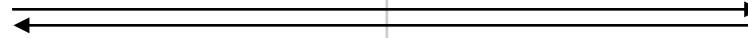
- Create : 데이터베이스에 데이터를 추가
- Read : 데이터베이스에서 데이터를 읽음
- Update : 데이터베이스의 데이터를 수정
- Delete : 데이터베이스의 데이터를 삭제

CRUD



Model

CRUD



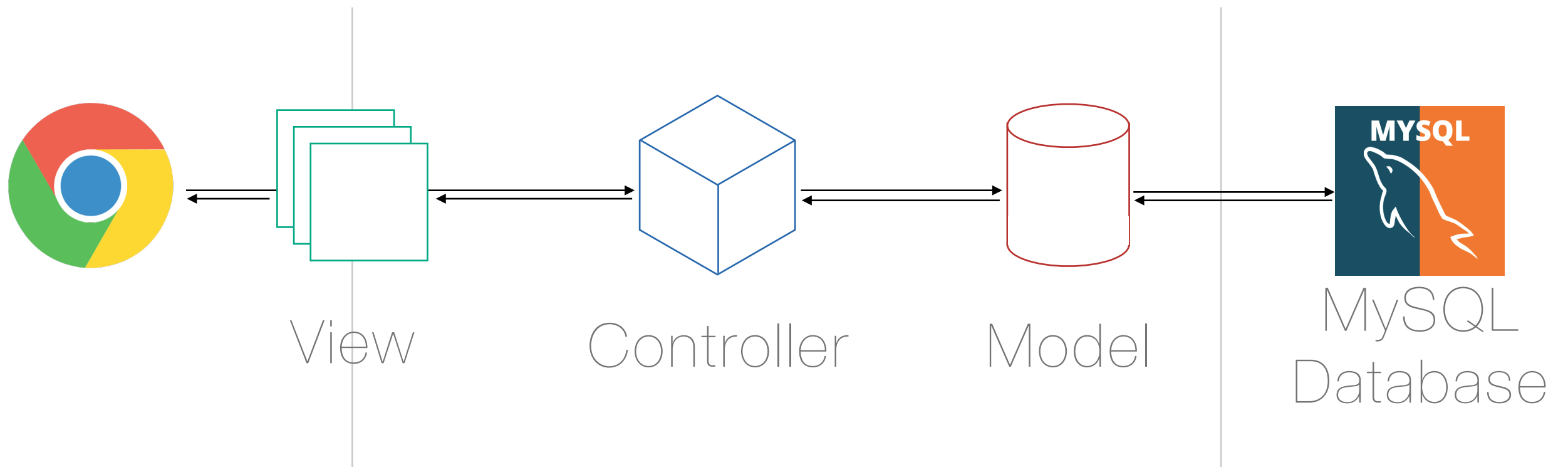
Database

MVC with Database

Client

Server

DB Server

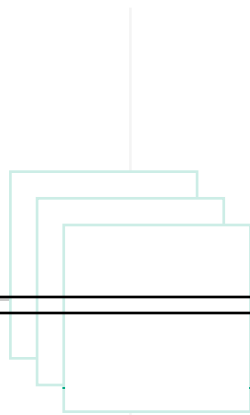


MVC with Database

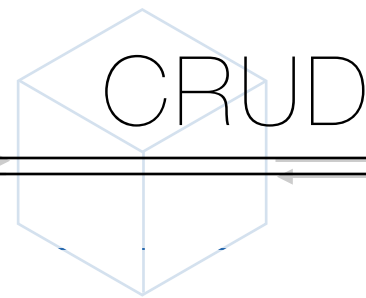
Client

Server

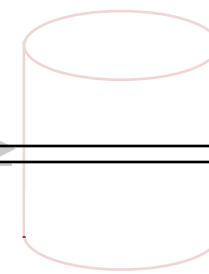
DB Server



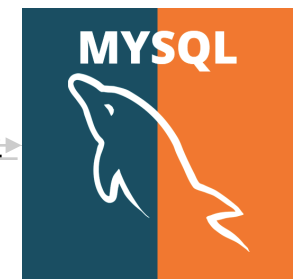
View



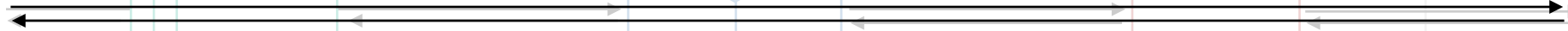
Controller



Model



MySQL
Database



SQL

- Structured Query Language
- 데이터베이스 CRUD에 사용되는 언어

SQL - Example

```
SELECT
    [ALL | DISTINCT | DISTINCTROW ]
    [HIGH_PRIORITY]
    [STRAIGHT_JOIN]
    [SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
    [SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
    select_expr [, select_expr ...]
    [FROM table_references
    [WHERE where_condition]
    [GROUP BY {col_name | expr | position}
    [ASC | DESC], ... [WITH ROLLUP]]
    [HAVING where_condition]
    [ORDER BY {col_name | expr | position}
    [ASC | DESC], ...]
    [LIMIT {[offset,] row_count | row_count OFFSET offset}]
    [PROCEDURE procedure_name(argument_list)]
    [INTO OUTFILE 'file_name' export_options
    | INTO DUMPFILE 'file_name'
    | INTO var_name [, var_name]]
    [FOR UPDATE | LOCK IN SHARE MODE]]
```

ORM

Object-Relational Mapping

쿼리를 간단하게 수행할 수 있는 방법
CRUD에 해당하는 파이썬 함수 제공

ORM > 테이블 생성

```
class User(db.Model):  
    id = db.Column(db.Integer, primary_key = True)  
    username = db.Column(db.String(45))  
    gender = db.Column(db.Enum('F', 'M'))  
    password = db.Column(db.String(100))  
    mobile = db.Column(db.String(15))  
    birthday = db.Column(db.Date)
```

ORM > Create

```
user = User(  
    email = data['email'],  
    username = data['username'],  
    gender = data['gender'],  
    password = db.func.md5(data['password']),  
    mobile = data['mobile'],  
    birthday = data['birthday']  
)  
  
db.session.add(user)  
db.session.commit()
```

ORM > Read

```
User.query.filter(User.email == 's_polaris@naver.com')
```

ORM > Update

```
user = User.query.get(id)
user.username = 'Alex'

db.session.commit()
```


ORM > Delete

```
user = User.query.get(id)
db.session.delete(user)
db.session.commit()
```

Flask-SQLAlchemy

- Flask에서 주로 사용하는 ORM 라이브러리

<https://pypi.python.org/pypi>

Flask-SQLAlchemy

- **MAC은 앞에 sudo 를 붙여주세요**
 - pip install flask-sqlalchemy
 - pip install flask-migrate

...