



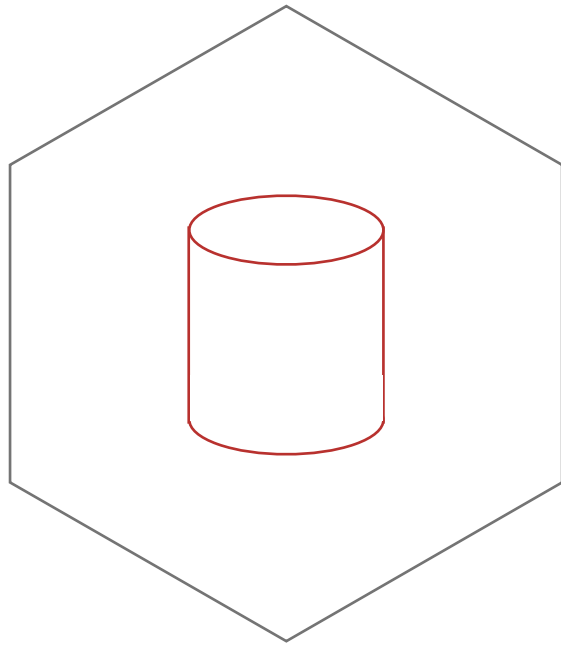
Database

Database

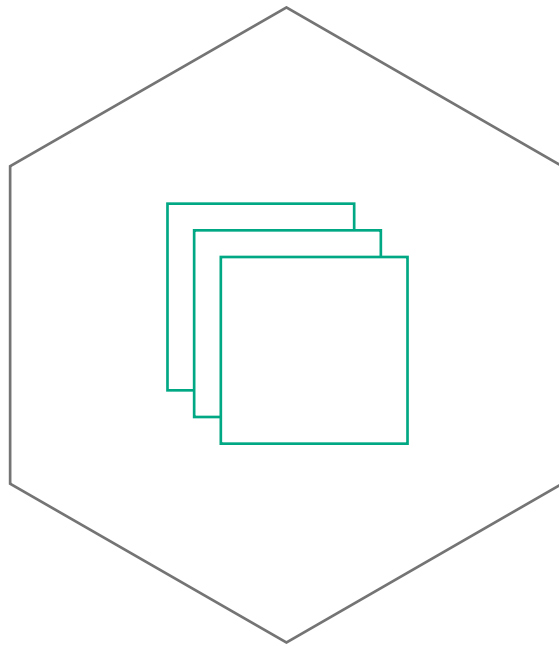
- MVC > Model
- Database
- 목표 : SNS 만들기

Review - MVC

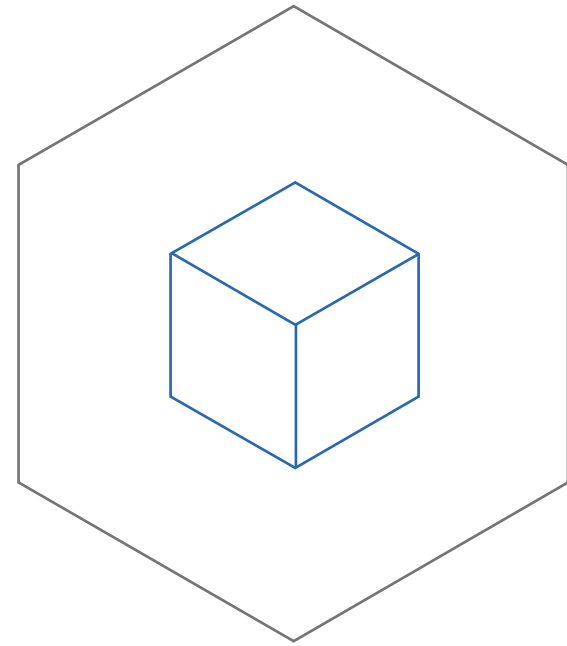
MVC



Model

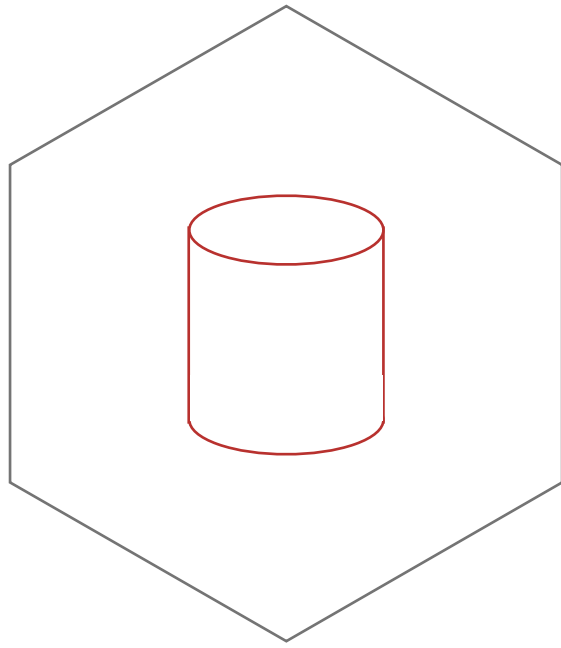


View

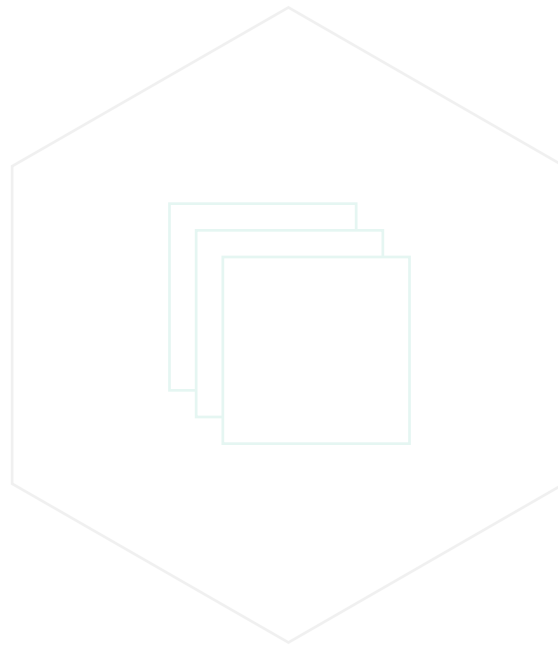


Controller

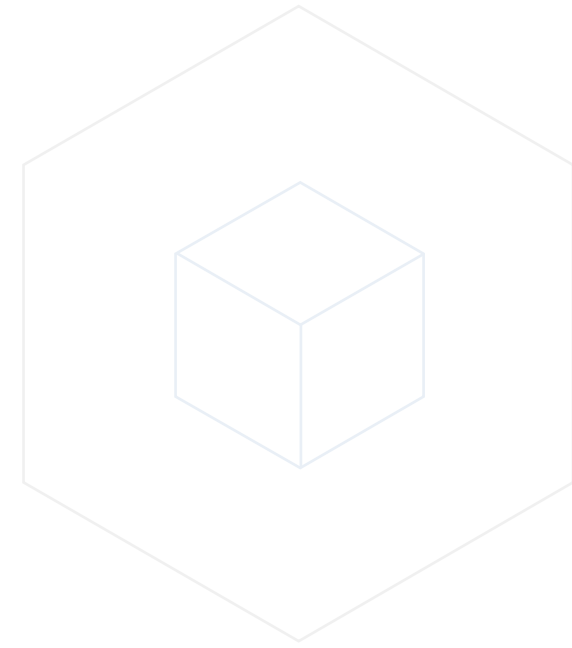
MVC



Model



View



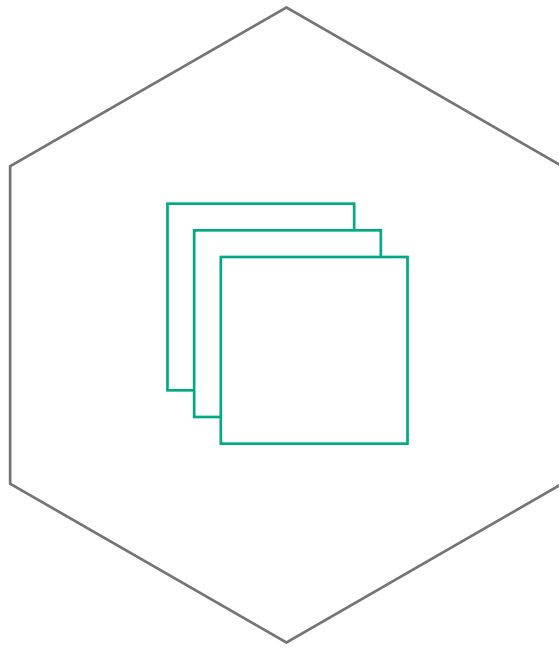
Controller

1. 사용자가 저장한 데이터를 예쁘게 변환해 **저장소**에 저장
2. 사용자가 요청하는 데이터를 **저장소**에서 꺼내와 전달

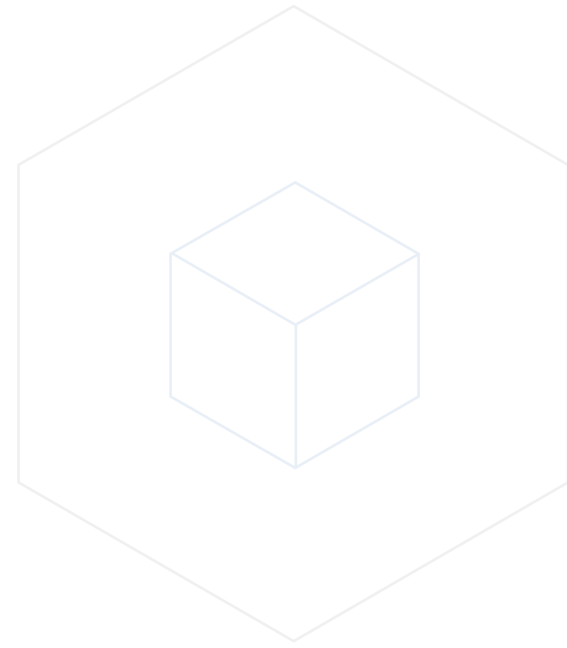
MVC



Model



View



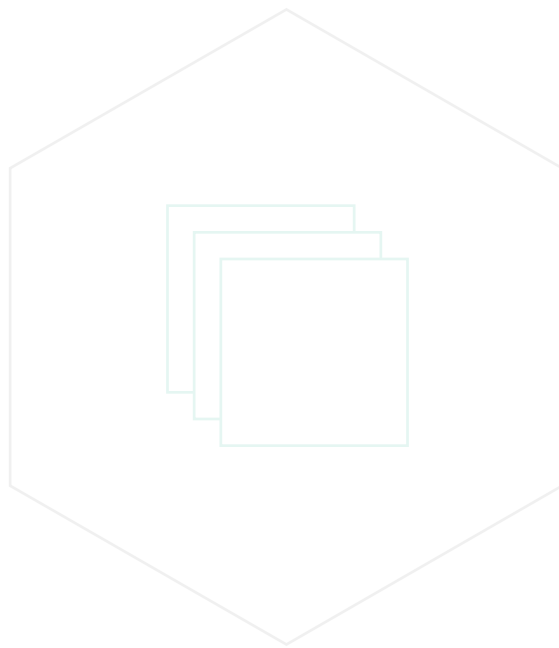
Controller

사용자의 요청에 대한 결과를 최종적으로 보여주는 역할

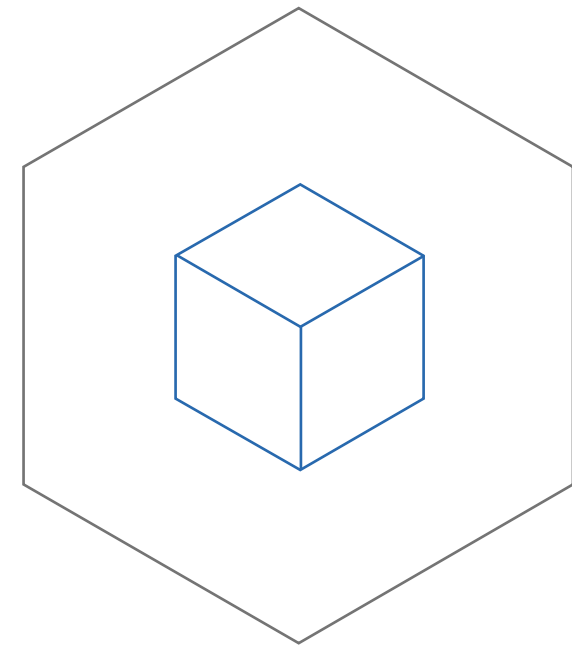
MVC



Model



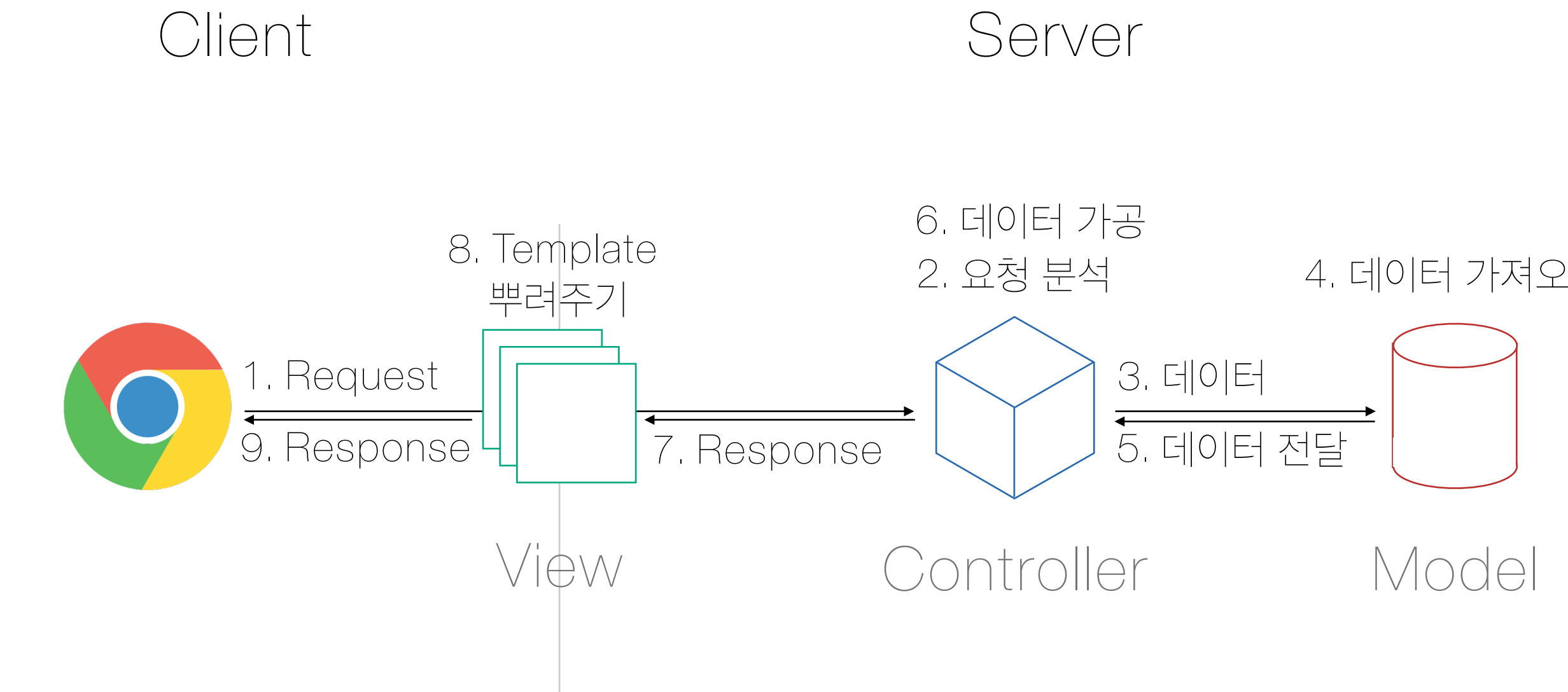
View



Controller

1. 사용자의 요청을 분석
2. 사용자가 제공한 데이터를 가공해 Model에게 전달
3. Model이 보내준 데이터를 가공해 View로 전달

MVC



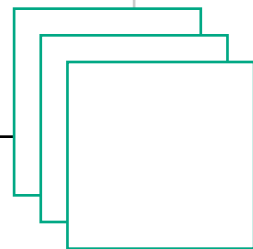
MVC

Client

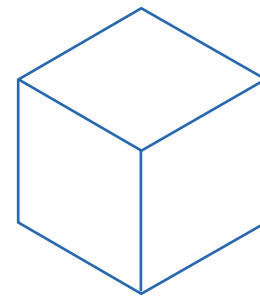
Server



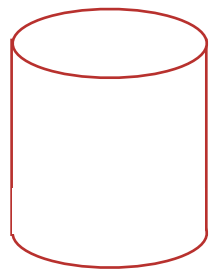
1. Request



View



Controller



Model

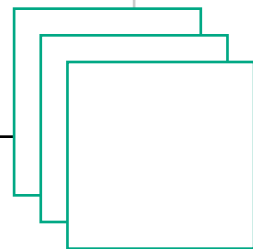
MVC

Client

Server

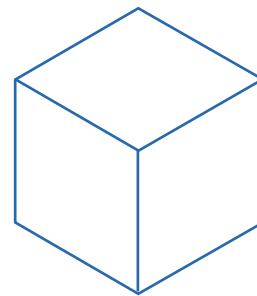


1.

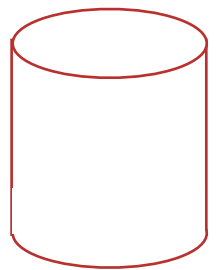


View

2. 요청 분석



Controller

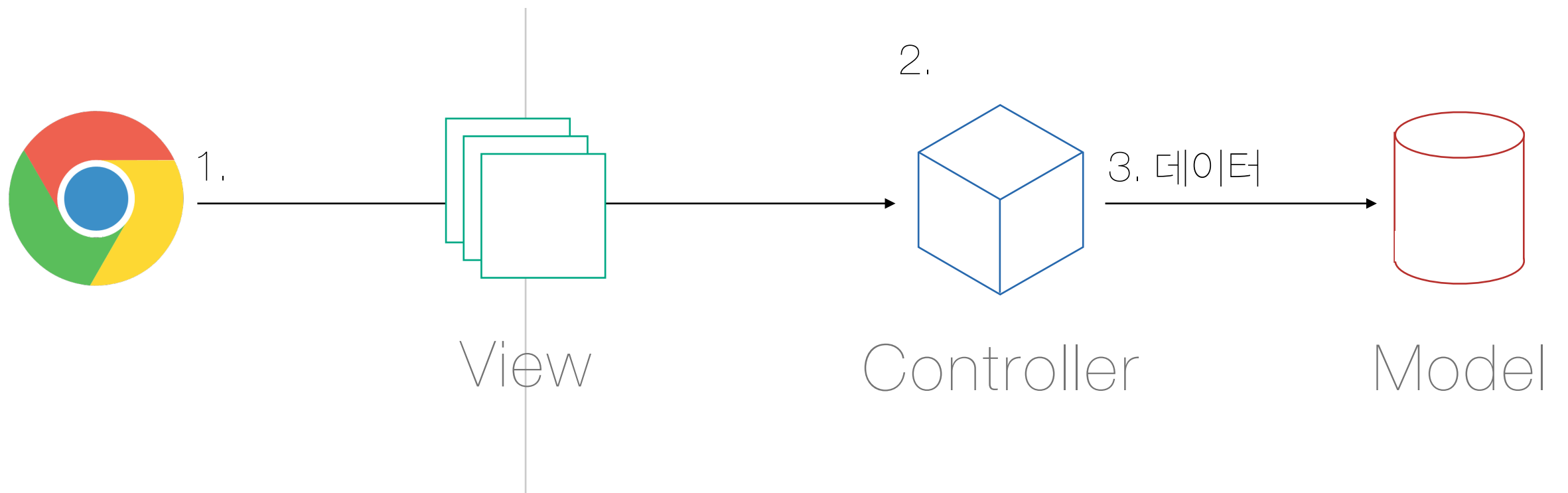


Model

MVC

Client

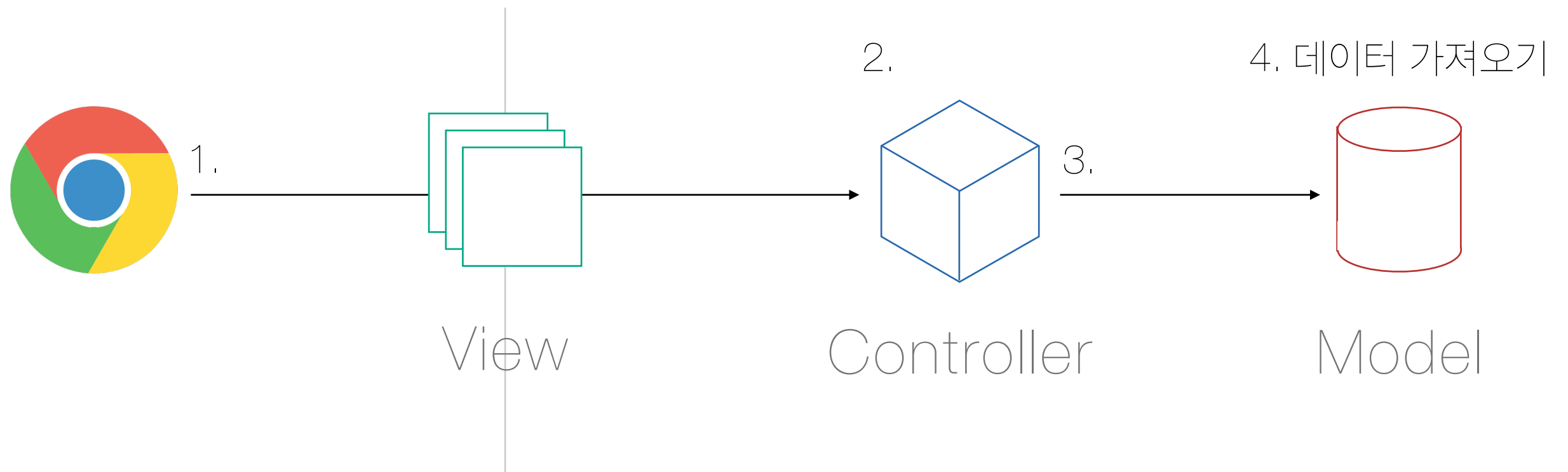
Server



MVC

Client

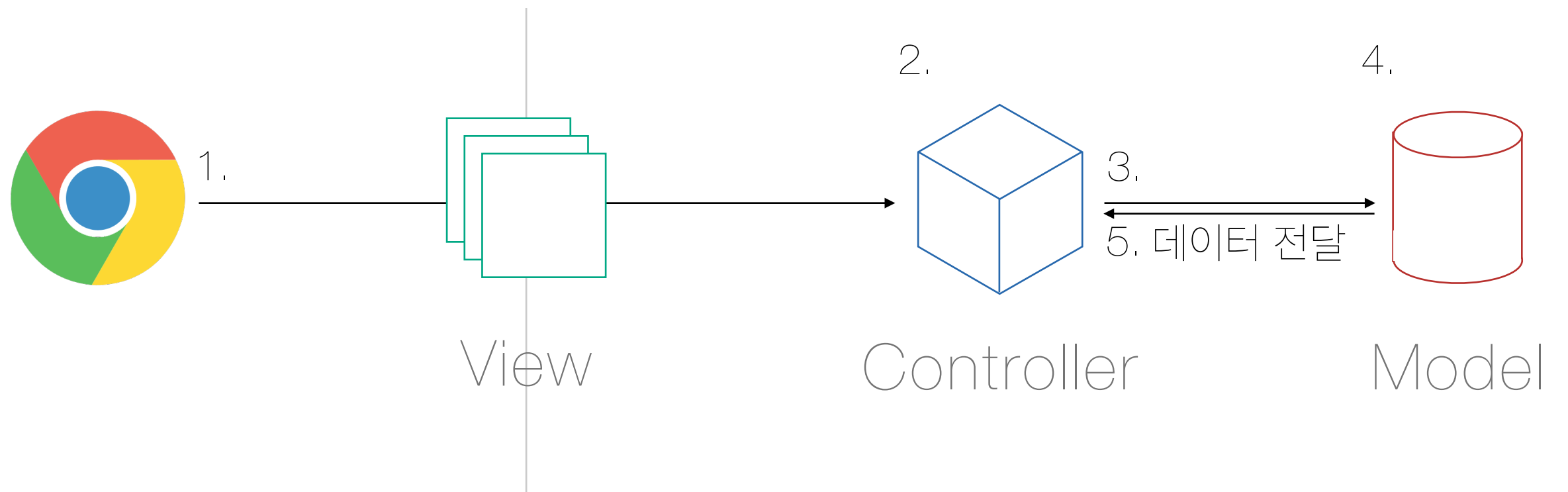
Server



MVC

Client

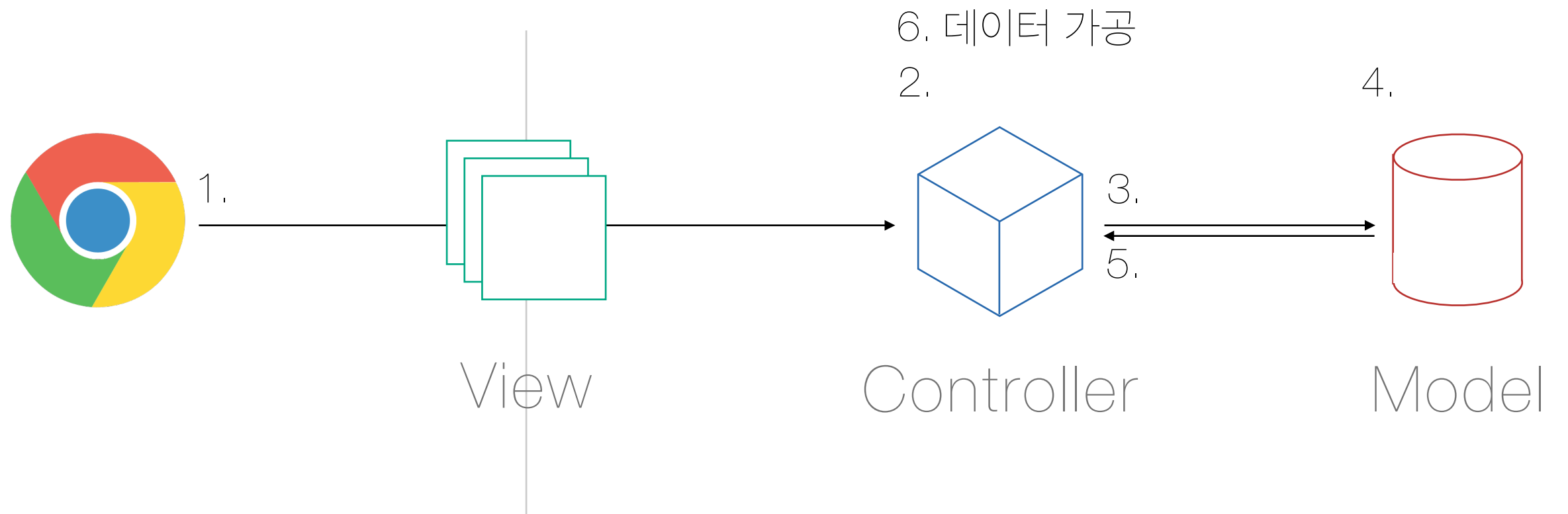
Server



MVC

Client

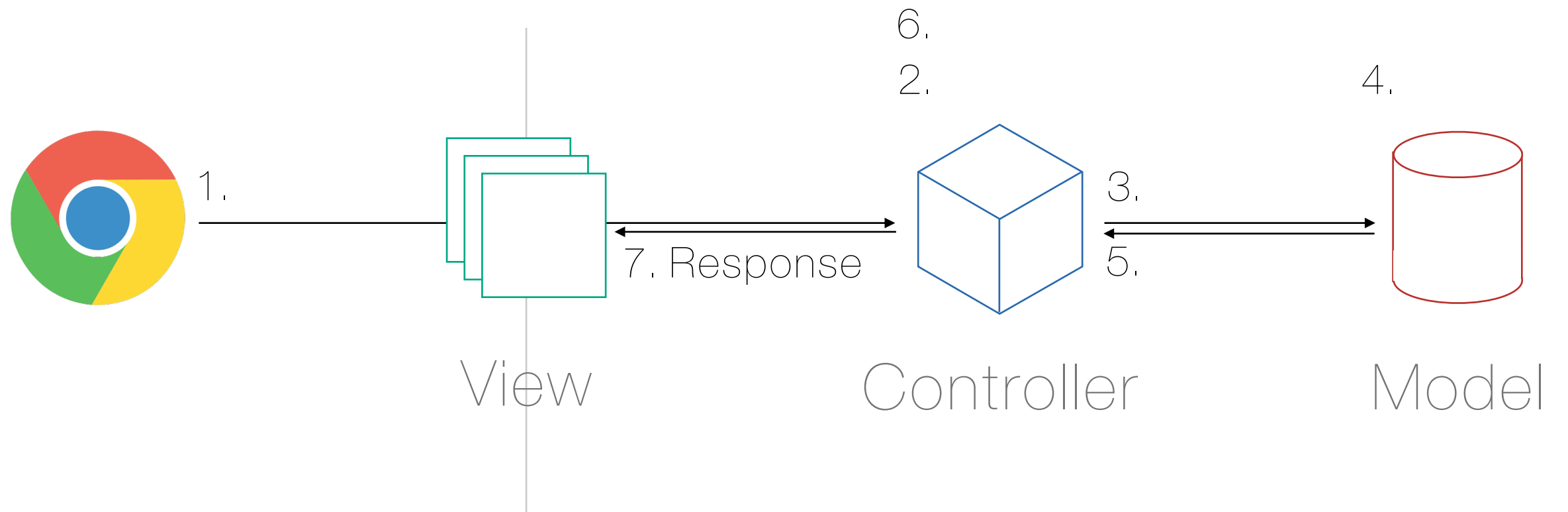
Server



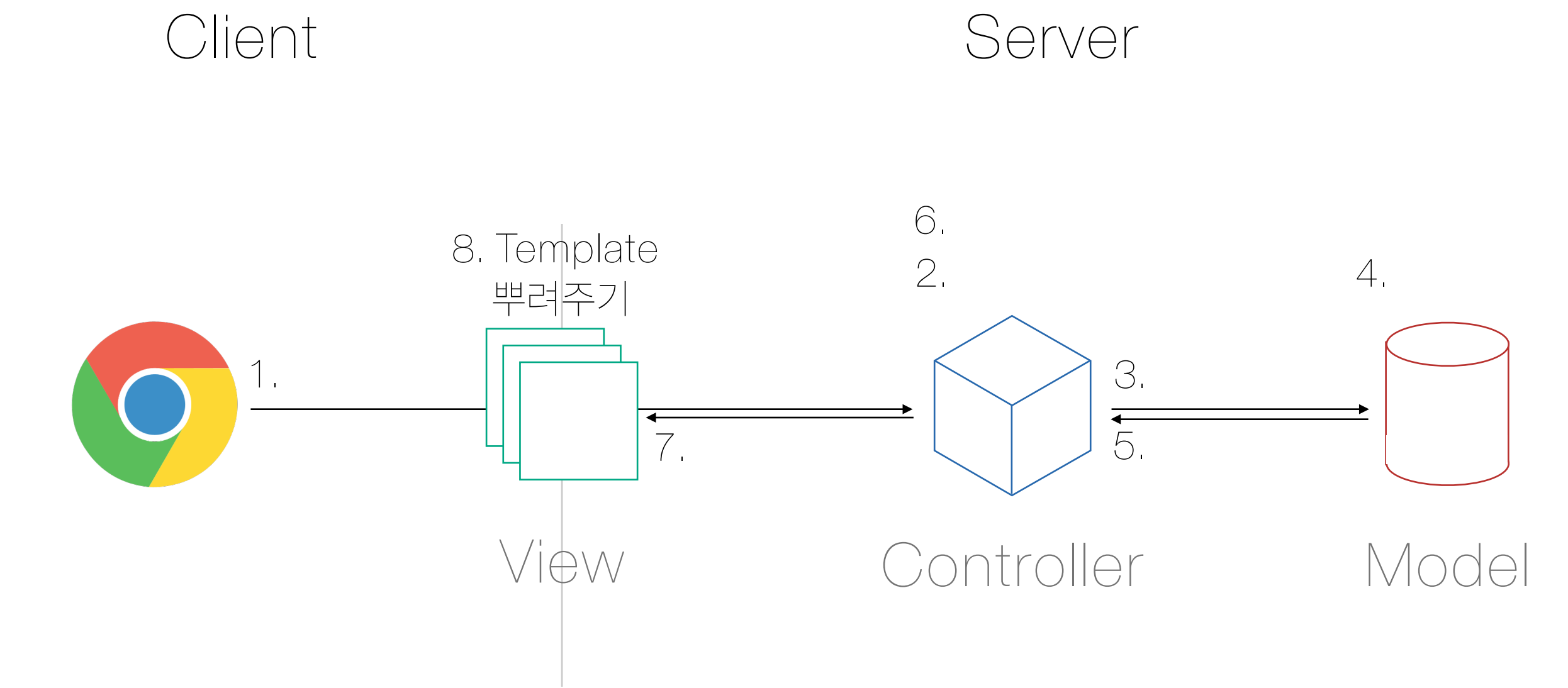
MVC

Client

Server



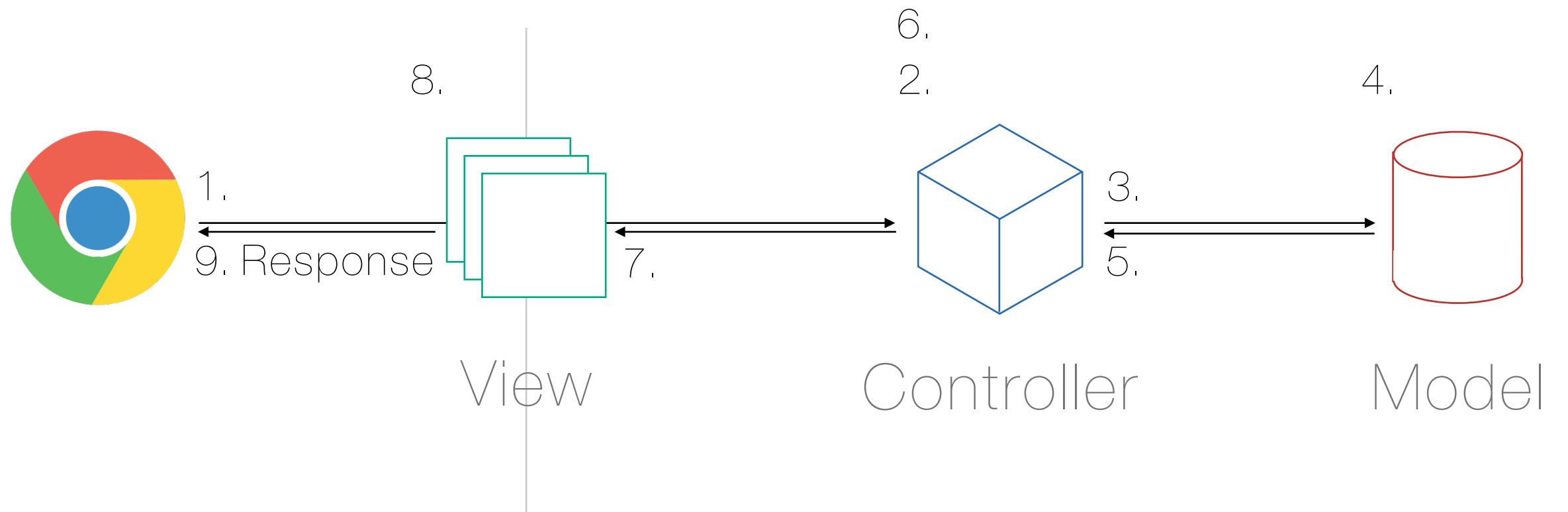
MVC



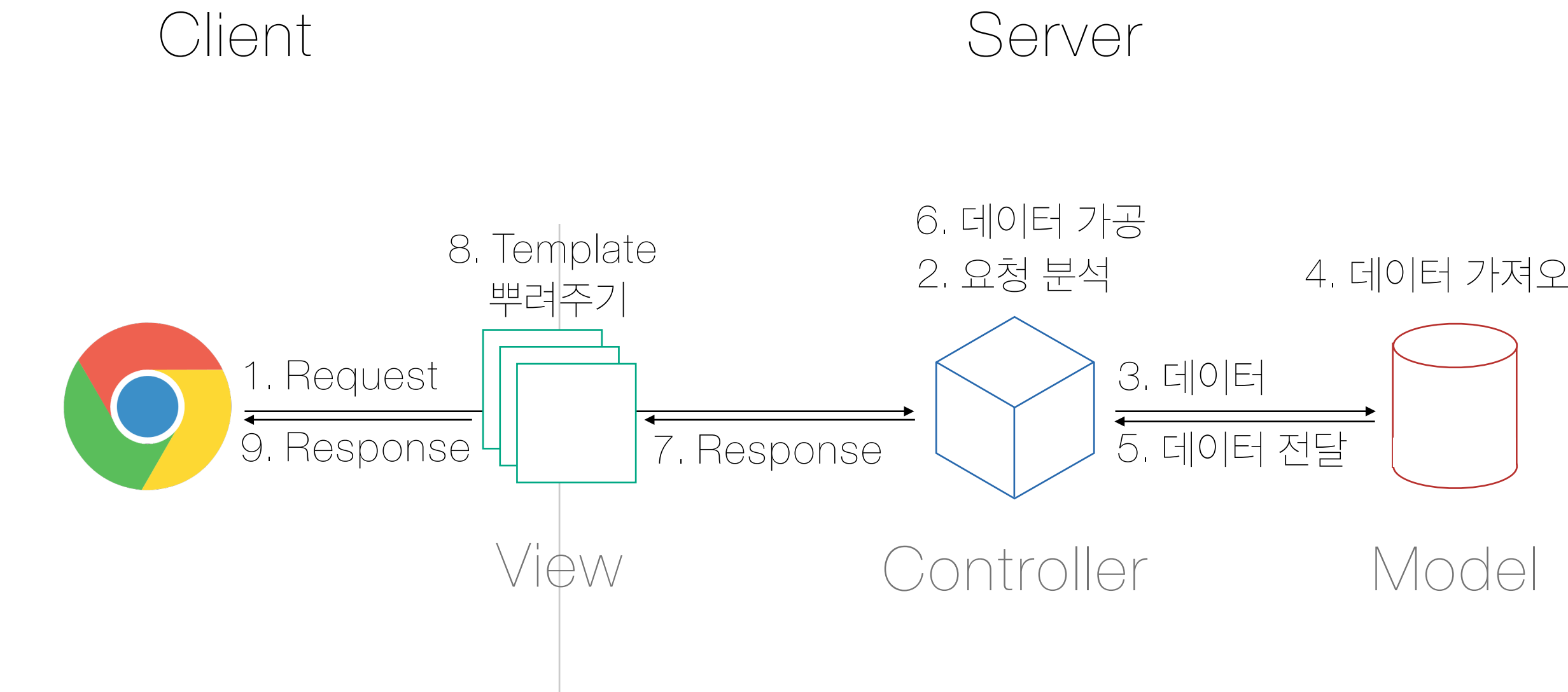
MVC

Client

Server

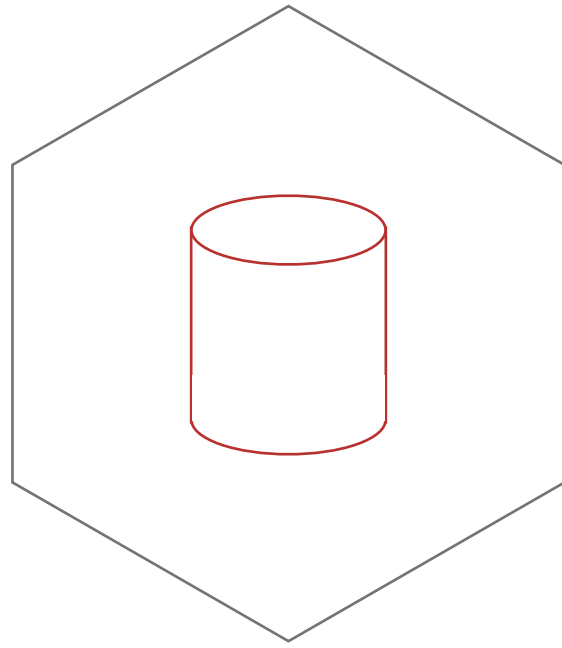


MVC



Database

MVC > Model



Model

- 서비스의 자료를 저장 · 관리
- 사용자가 입력하거나 미리 준비된 자료들을 일정한 형태로 **데이터베이스**에 저장, 출력, 수정, 삭제(CRUD)하는 부분

Database

- 서버 상에 데이터를 저장할 수 있는 공간
- 데이터는 테이블 형태로 저장됨

Database > Table

- Table 'users'

| ID | email | password | schoolId | userLevel |
|----|-----------------------|---|----------|-----------|
| 19 | s_polaris@naver.com | *8CC4F3A11708BF5BB08D58A545EA2953CD0ECD9E | 7 | 2 |
| 20 | polaris341@gmail.com | *8CC4F3A11708BF5BB08D58A545EA2953CD0ECD9E | 7 | 1 |
| 21 | jiho.hong@hotmail.com | *8CC4F3A11708BF5BB08D58A545EA2953CD0ECD9E | 7 | 1 |
| 22 | pjhjohn@gmail.com | *4EA558B931ED8C50A6EA48AFB514401C2D911389 | 7 | 2 |
| 23 | namdy0429@gmail.com | *734EA8818426E68FE909C89F9801039434560532 | 7 | 2 |
| 24 | alexhan46@gmail.com | *EE7411B683D1BD7DD47AEDCFF2315E1323148A69 | 7 | 2 |
| 25 | vhfpqj327@naver.com | *FF531076B5E1ADBF390159FF8AA7D9A3AF0D1656 | 9 | 2 |
| 26 | rishubil@gmail.com | *B89FFCF610F9B892992C9EBB4373035C210ABC58 | 11 | 2 |
| 27 | dusgmldutls@gmail.com | *62FDDE25F3B2D0E52DCB67C1013ADE0F5536C759 | 3 | 1 |
| 28 | lsywind3@gmail.com | *B974B4CBD851BFB62DFB72518A9B4EF589F5B7B1 | 4 | 1 |
| 29 | khulikelion@gmail.com | *31BD9D075FFE841DF06E9E22799D097E491A922C | 3 | 2 |
| 30 | bok3104@naver.com | *B89FFCF610F9B892992C9EBB4373035C210ABC58 | 11 | 1 |

Database > Schema

- 데이터베이스의 구조
- 데이터베이스 설계 시 정의되며 자주 변경되지 않음
- Schema Diagram

PERSON

| | | | | |
|------|---------|-------|-----|--------|
| Name | Address | Phone | Fax | Mobile |
|------|---------|-------|-----|--------|

COMPANY

| | | | | |
|------|---------|--------|---------|----------|
| Name | Address | VAT ID | UKCO ID | Turnover |
|------|---------|--------|---------|----------|

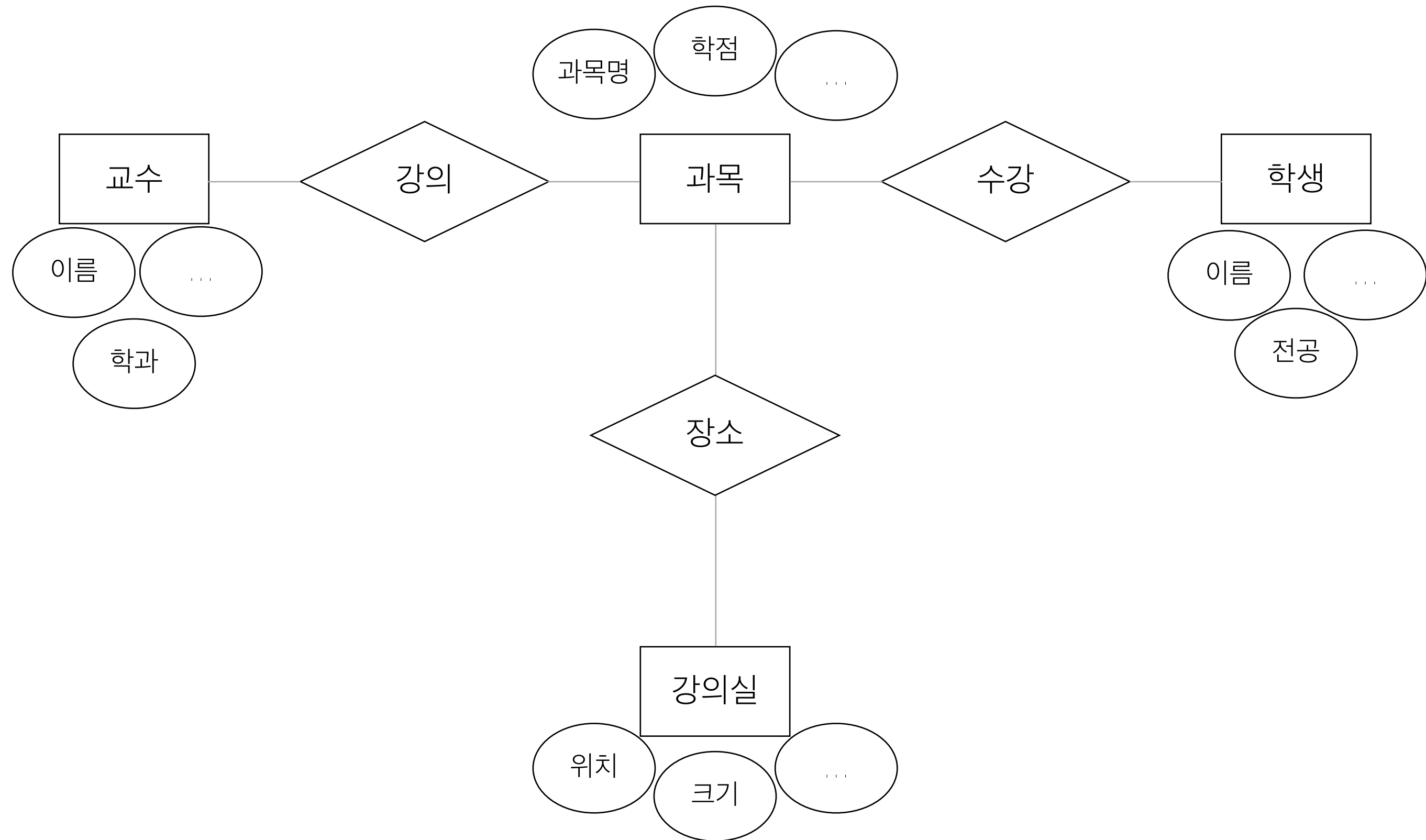
Database - Entity, Attribute, Relationship

- 개체(Entity) : 실세계의 물건, 개념
- 속성(Attribute) : 개체의 성질, 속성
- 관계(Relationship) : 개체 사이의 관계

Database - Entity, Attribute, Relationship

- 개체(Entity) : 학생, 선생님, 수업, 강의실, ...
- 속성(Attribute) : 이름, 나이, 학교, 전공, 주제, 위치, ...
- 관계(Relationship) : 학생 - 수업, 선생님 - 수업, 강의실 - 수업, ...

Database Example - 수강신청 시스템



Database Example - 수강신청 시스템

‘교수’ 테이블

| ID | 이름 | 학과 | ... |
|-----|----------|------------------|-----|
| 1 | 박준호 | 컴퓨터공학부 | ... |
| 2 | Alex Han | Computer Science | ... |
| 3 | 남다예 | 컴퓨터과학 | ... |
| ... | ... | ... | ... |

Database Example - 수강신청 시스템

‘교수’ 테이블

| ID | 이름 | 학과 | ... |
|-------------------------|----------|------------------|-----|
| Row : 테이블에 들어 있는 각각의 자료 | | | |
| 2 | Alex Han | Computer Science | ... |
| 3 | 남다예 | 컴퓨터과학 | ... |
| ... | ... | ... | ... |

Database Example - 수강신청 시스템

‘교수’ 테이블

| ID | 이름 | 학과 | ... |
|----|----------|------------------|-----|
| 1 | 박준호 | 컴퓨터공학부 | ... |
| 2 | Alex Han | Computer Science | ... |
| 3 | 남다예 | 컴퓨터과학 | ... |

Column : 개체의 속성들이 각각 column이 됨

Database Example - 수강신청 시스템

‘교수’ 테이블

| ID | 이름 | 학과 | ... |
|-----|----------|------------------|-----|
| 1 | 박준호 | 컴퓨터공학부 | ... |
| 2 | Alex Han | Computer Science | ... |
| ... | ... | ... | ... |

Value : 실제 값이 들어있는 부분

Database Example - 수강신청 시스템

‘교수’ 테이블

| ID | 이름 | 학과 | ... |
|----------------------------|----------|------------------|-----|
| ID 속성 : 이 테이블의 primary key | | | |
| 2 | Alex Han | Computer Science | ... |
| 3 | 남다예 | 컴퓨터과학 | ... |
| ... | ... | ... | ... |

Database Example - 수강신청 시스템

교수

| ID | 이름 | 학과 | ... |
|----|----|----|-----|
|----|----|----|-----|

학생

| ID | 이름 | 전공 | ... |
|----|----|----|-----|
|----|----|----|-----|

과목

| ID | 과목명 | 학점 | ... |
|----|-----|----|-----|
|----|-----|----|-----|

강의실

| ID | 위치 | 크기 | ... |
|----|----|----|-----|
|----|----|----|-----|

Database Example - 수강신청 시스템

교수

| ID | 이름 | 학과 | ... |
|----|----|----|-----|
|----|----|----|-----|

학생

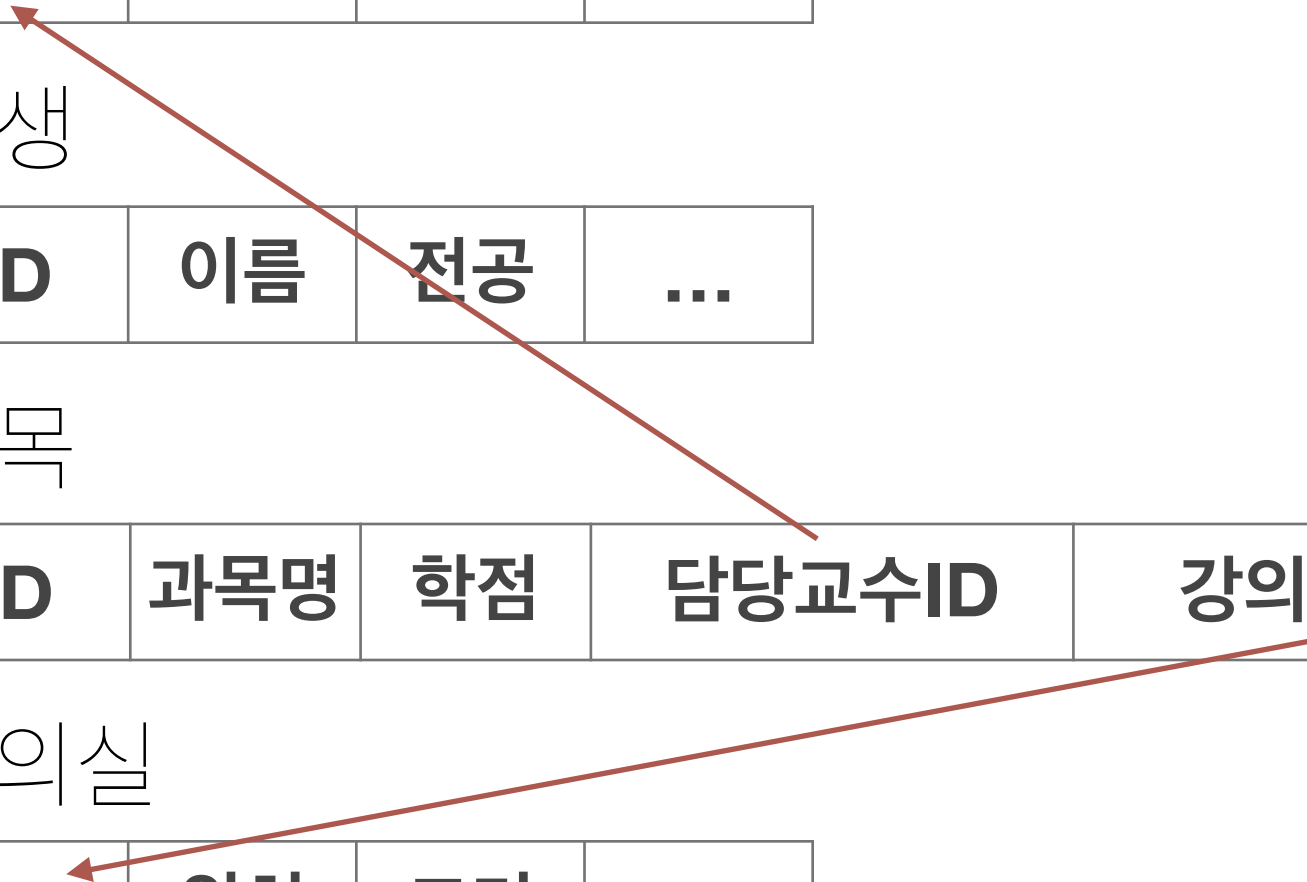
| ID | 이름 | 전공 | ... |
|----|----|----|-----|
|----|----|----|-----|

과목

| ID | 과목명 | 학점 | 담당교수ID | 강의실ID | ... |
|----|-----|----|--------|-------|-----|
|----|-----|----|--------|-------|-----|

강의실

| ID | 위치 | 크기 | ... |
|----|----|----|-----|
|----|----|----|-----|



Database Example - 수강신청 시스템

학생

| ID | 이름 | ... |
|-----|-----|-----|
| 1 | 홍지호 | ... |
| 2 | 박준호 | ... |
| 3 | 한영후 | ... |
| 4 | 남다예 | ... |
| ... | ... | ... |

과목

| ID | 과목명 | ... |
|-----|--------|-----|
| 1 | HTML | ... |
| 2 | CSS | ... |
| 3 | JS | ... |
| 4 | Python | ... |
| ... | ... | ... |

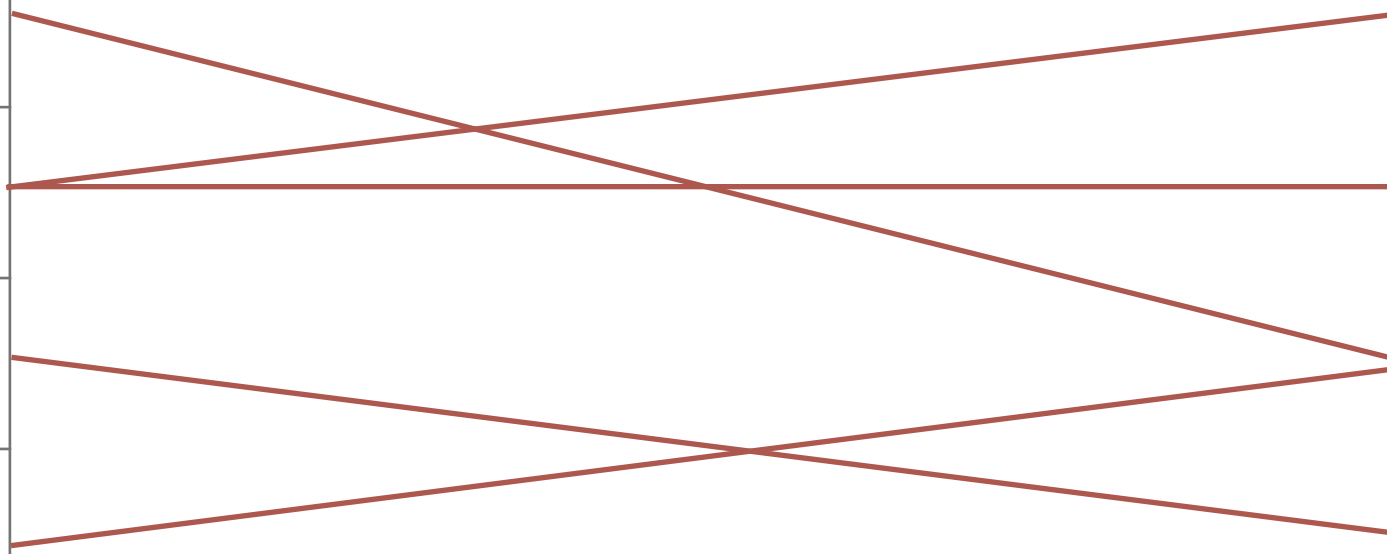
Database Example - 수강신청 시스템

학생

| ID | 이름 | ... |
|-----|-----|-----|
| 1 | 홍지호 | ... |
| 2 | 박준호 | ... |
| 3 | 한영후 | ... |
| 4 | 남다예 | ... |
| ... | ... | ... |

과목

| ID | 과목명 | ... |
|-----|--------|-----|
| 1 | HTML | ... |
| 2 | CSS | ... |
| 3 | JS | ... |
| 4 | Python | ... |
| ... | ... | ... |



Database Example - 수강신청 시스템

학생

| ID | 이름 | ... |
|-----|-----|-----|
| 1 | 홍지호 | ... |
| 2 | 박준호 | ... |
| 3 | 한영후 | ... |
| 4 | 남다예 | ... |
| ... | ... | ... |

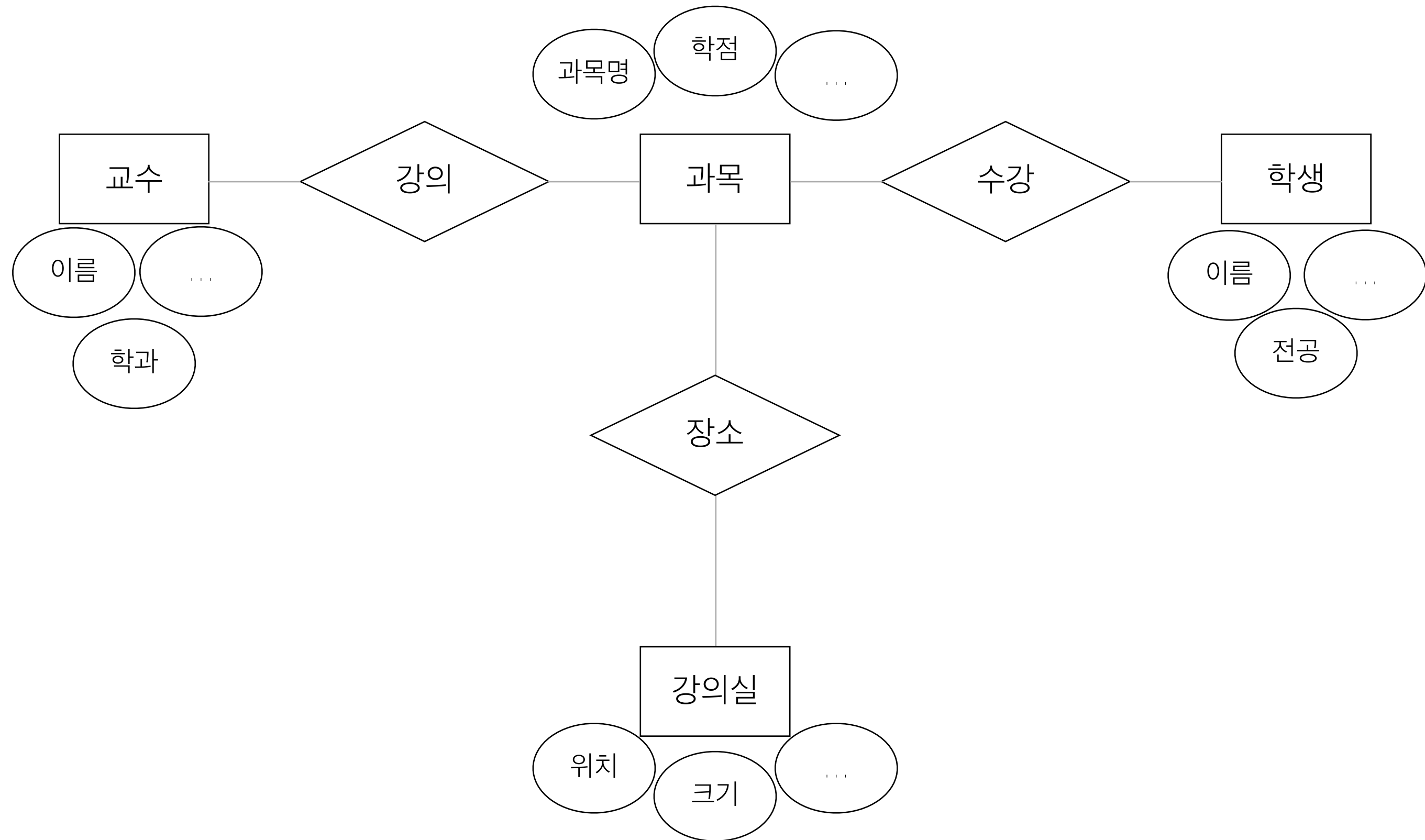
수강신청

| 학생ID | 과목ID | ... |
|------|------|-----|
| 1 | 3 | ... |
| 2 | 1 | ... |
| 2 | 2 | ... |
| 3 | 4 | ... |
| 4 | 3 | ... |
| ... | ... | ... |

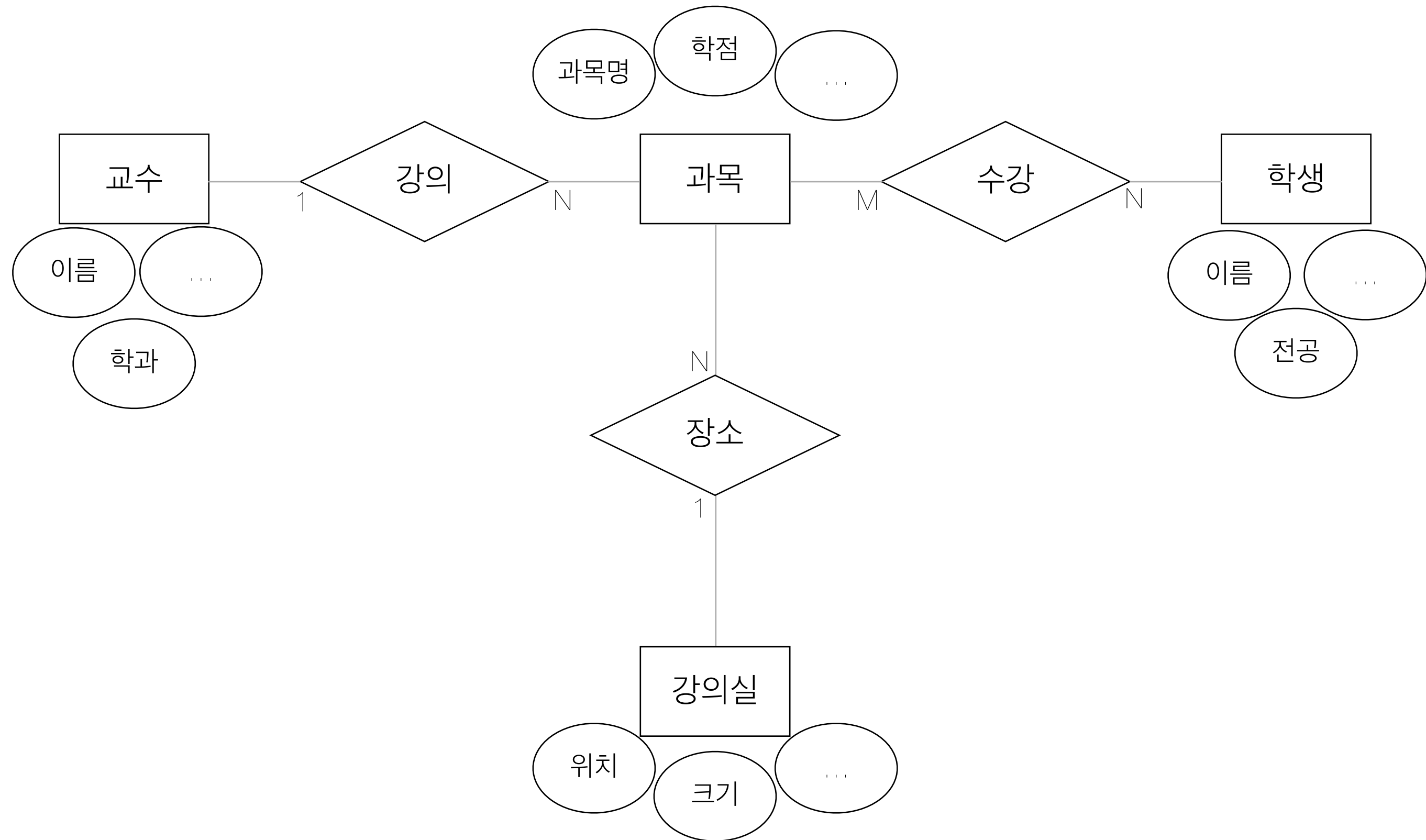
과목

| ID | 과목명 | ... |
|-----|--------|-----|
| 1 | HTML | ... |
| 2 | CSS | ... |
| 3 | JS | ... |
| 4 | Python | ... |
| ... | ... | ... |

Database Example - 수강신청 시스템



Database Example - 수강신청 시스템



Database Example - 수강신청 시스템

교수

| ID | 이름 | 학과 | ... |
|----|----|----|-----|
|----|----|----|-----|

학생

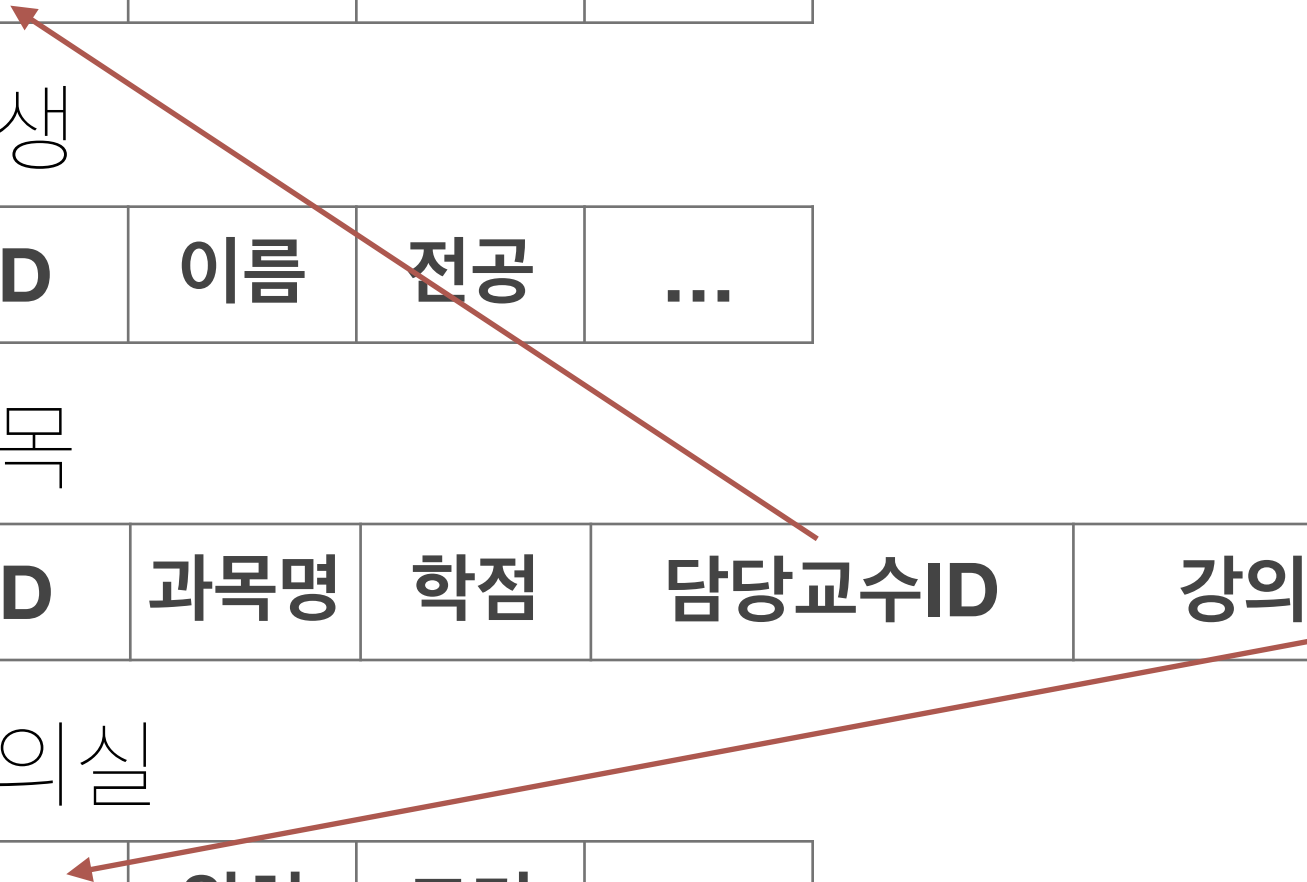
| ID | 이름 | 전공 | ... |
|----|----|----|-----|
|----|----|----|-----|

과목

| ID | 과목명 | 학점 | 담당교수ID | 강의실ID | ... |
|----|-----|----|--------|-------|-----|
|----|-----|----|--------|-------|-----|

강의실

| ID | 위치 | 크기 | ... |
|----|----|----|-----|
|----|----|----|-----|



Database Example - 수강신청 시스템

교수

| ID | 이름 | 학과 | ... |
|----|----|----|-----|
|----|----|----|-----|

학생

| ID | 이름 | 전공 | ... |
|----|----|----|-----|
|----|----|----|-----|

과목

| ID | 과목명 | 학점 | 담당교수ID | 강의실ID | ... |
|----|-----|----|--------|-------|-----|
|----|-----|----|--------|-------|-----|

강의실

| ID | 위치 | 크기 | ... |
|----|----|----|-----|
|----|----|----|-----|

수강신청

| 학생ID | 과목ID | 학기 | ... |
|------|------|----|-----|
|------|------|----|-----|



Database Example - 수강신청 시스템

교수

| ID | 이름 | 학과 | ... |
|----|----|----|-----|
|----|----|----|-----|

학생

| ID | 이름 | 전공 | ... |
|----|----|----|-----|
|----|----|----|-----|

과목

| ID | 과목명 | 학점 | 담당교수ID | 강의실ID | ... |
|----|-----|----|--------|-------|-----|
|----|-----|----|--------|-------|-----|

강의실

| ID | 위치 | 크기 | ... |
|----|----|----|-----|
|----|----|----|-----|

수강신청

| 학생ID | 과목ID | 학기 | ... |
|------|------|----|-----|
|------|------|----|-----|

Database > Key

- 각 테이블은 키를 가짐
- 키 : 각각의 개체를 구분할 수 있는 속성
- Primary key, unique key, foreign key
 - Primary key : 테이블에서 각각의 개체들을 구분할 수 있는 속성
 - 강의실 : 호수, 학생 : 학번, ...
 - Unique Key : 테이블 내의 개체들끼리 중복될 수 없는 속성
 - 학생 : 주민등록번호, ...
 - Foreign key : 다른 테이블의 PK를 참조하는 속성

Database Example - 수강신청 시스템

교수

| <u>ID</u> | 이름 | 학과 | ... |
|-----------|----|----|-----|
|-----------|----|----|-----|

학생

| <u>ID</u> | 이름 | 전공 | ... |
|-----------|----|----|-----|
|-----------|----|----|-----|

과목

| <u>ID</u> | 과목명 | 학점 | 담당교수ID | 강의실ID | ... |
|-----------|-----|----|--------|-------|-----|
|-----------|-----|----|--------|-------|-----|

강의실

| <u>ID</u> | 위치 | 크기 | ... |
|-----------|----|----|-----|
|-----------|----|----|-----|

수강신청

| <u>학생ID</u> | <u>과목ID</u> | 학기 | ... |
|-------------|-------------|----|-----|
|-------------|-------------|----|-----|

Database Example - 수강신청 시스템

교수

| ID | 이름 | 학과 | ... |
|----|----|----|-----|
|----|----|----|-----|

학생

| ID | 이름 | 전공 | ... |
|----|----|----|-----|
|----|----|----|-----|

과목

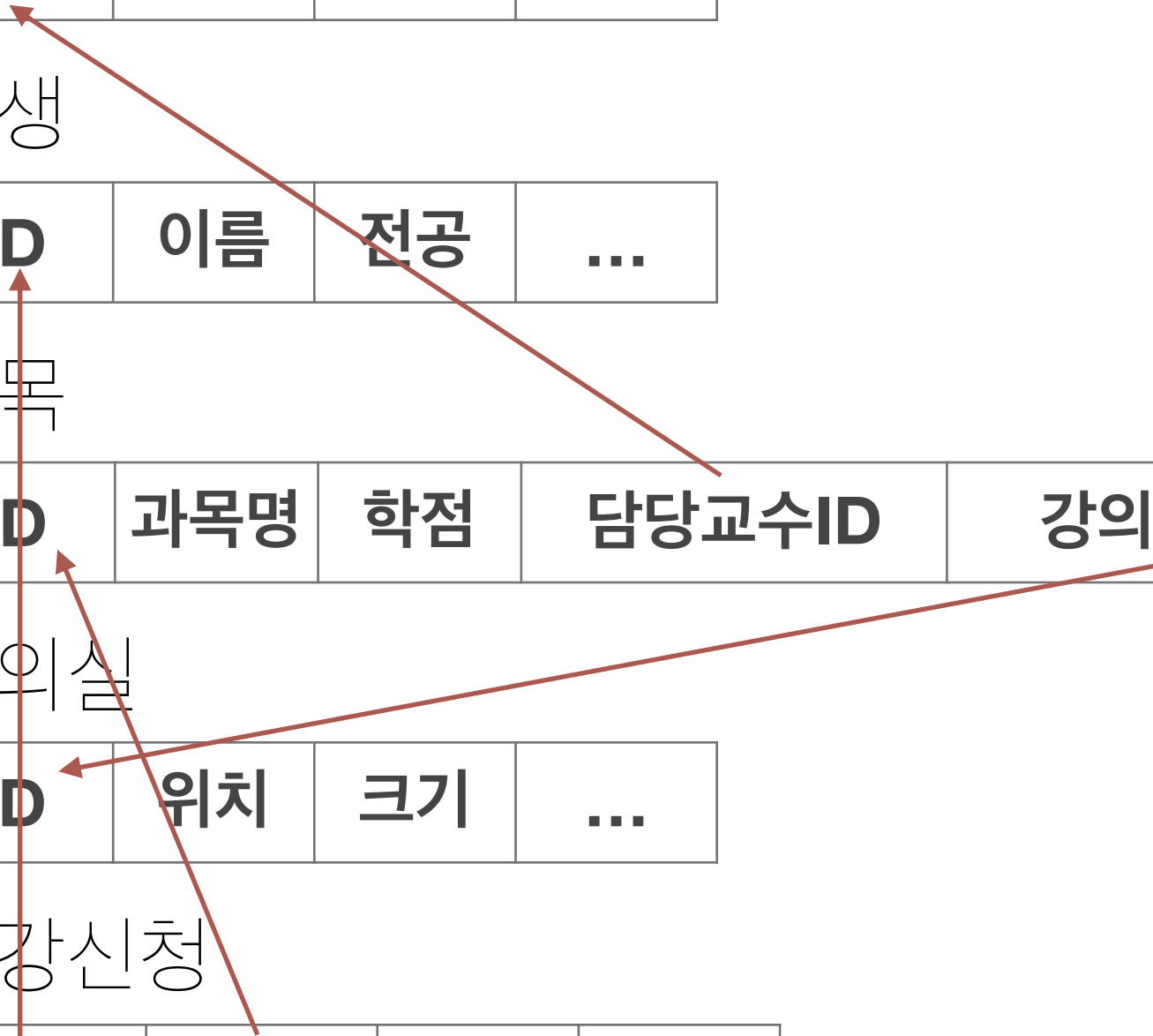
| ID | 과목명 | 학점 | 담당교수ID | 강의실ID | ... |
|----|-----|----|--------|-------|-----|
|----|-----|----|--------|-------|-----|

강의실

| ID | 위치 | 크기 | ... |
|----|----|----|-----|
|----|----|----|-----|

수강신청

| 학생ID | 과목ID | 학기 | ... |
|------|------|----|-----|
|------|------|----|-----|

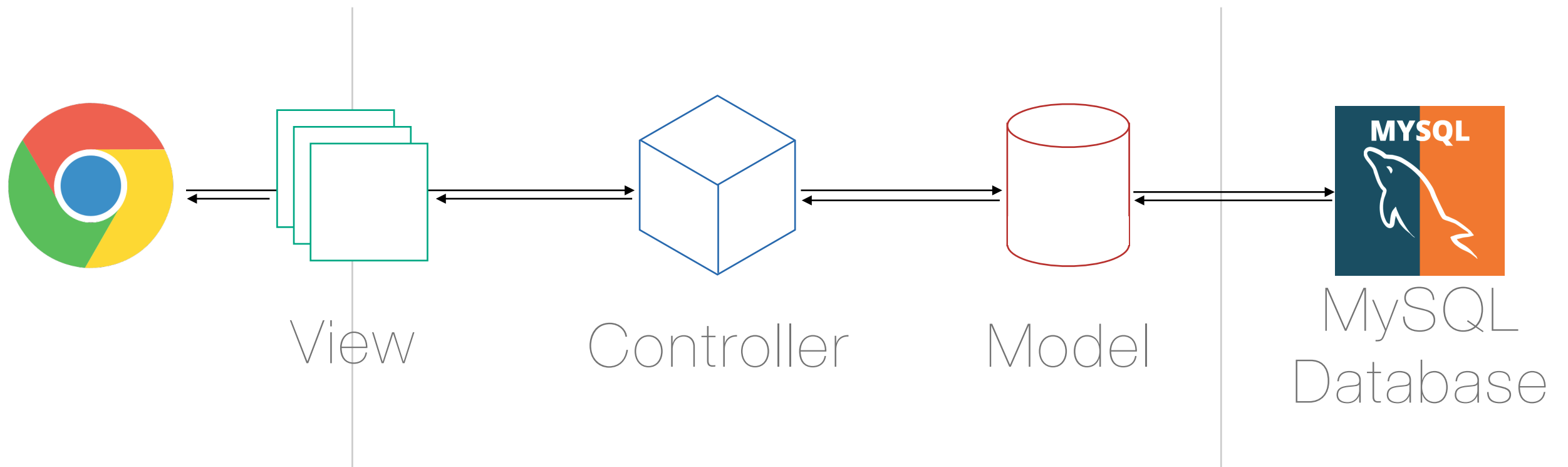


MVC with Database

Client

Server

DB Server



...

MySQL > Data Types

<http://dev.mysql.com/doc/refman/5.6/en/data-types.html>

MySQL > Data Types > Numeric Types

- TINYINT[(M)]
 - Very small integer
 - -128 ~ 127
 - M : Total number of digits
- INT[(M)]
 - Normal-size integer
 - -2147483648 ~ 2147483647
- BIGINT[(M)]
 - Large integer
 - -9223372036854775808 ~ 9223372036854775807

MySQL > Data Types > Numeric Types

- `DECIMAL[(M[,D])]`
 - “Exact” fixed-point number
 - M : Total number of digits
 - Maximum : 65
 - D : Number of digits after decimal point
 - Maximum : 30, default : 0
 - Decimal point and the “-” sign are not counted in M

MySQL > Data Types > Numeric Types

- `FLOAT[(M,D)]`
 - Small floating-point number
- `DOUBLE[(M,D)]`
 - Normal-size floating-point number

MySQL > Data Types > Date and Time Types

- DATE
 - Range : '1000-01-01' to '9999-12-31'
 - MySQL displays DATE values in 'YYYY-MM-DD' format
- TIME, DATETIME, YEAR

MySQL > Data Types > Date and Time Types

- **TIMESTAMP**
 - Range : '1970-01-01 00:00:01.0000000' to '2038-01-19 03:14:07.999999'
 - TIMESTAMP values are stored as the number of seconds since '1970-01-01 00:00:00'
 - '1970-01-01 00:00:00' 은 나타낼 수 없다. 0은 '0000-00-00 00:00:00' 을 나타내기 위해 사용되기 때문

MySQL > Data Types > String Types

- CHAR[(M)]
 - Fixed-length string
 - M : length
 - Always right-padded with spaces
- VARCHAR[(M)]
 - Variable-length string
 - M : maximum length, 0 to 65535

MySQL > Data Types > String Types

- TEXT
 - Maximum length : 65535 characters
 - TINYTEXT, MEDIUMTEXT, LONGTEXT(max 4GB)

MySQL > Data Types > String Types

- ENUM('value1', 'value2', ...)
 - String object that can have only one value, chosen from the list of values ('value1', 'value2', ..., NULL or the special " error value
- SET('value1', 'value2', ...)
 - String object that can have zero or more values

...

목표 : SNS 만들기

<http://astyfx-personal-project.appspot.com/>

SNS

- 회원가입 / 로그인
- 자신 또는 다른 사용자의 담벼락에 글 작성 가능
- 담벼락 글에 댓글 작성 가능
- 비밀글

실습 & 과제

실습 & 과제

- 실습
 - 서비스 개발용 앱엔진 프로젝트 생성
 - 구글 클라우드 SQL 인스턴스 생성 - 앱엔진 프로젝트 연결
 - MySQL 워크벤치 설치, 클라우드 SQL 연결 설정
 - 사용자 정보 테이블 만들기
- 과제
 - DB 스키마 완성해오기
 - HTML 템플릿 만들어오기
 - 담벼락, 글 보기/쓰기