

# 시스템 분석설계 교과목 포트폴리오

20202701 노승준

# 목차

## Github 학습

- 버전관리와 커밋
- Push와 Pull
- Fork 와 Clone
- Fetch and merge
- Pull request

## 소프트웨어 공학

- 소프트웨어 개요
- 소프트웨어의 특징
- 선형 순차적 모델
- V 모델
- 프로토타입 모델

## MySQL 학습

- MySQL의 구조
- 테이블의 생성
- 컬럼을 편집
- INSERT / UPDATE / SELECT
- 관계형 데이터베이스의 필요성
- JOIN
- 인터넷과 데이터베이스
- MySQL 클라이언트 비교

# Github 학습

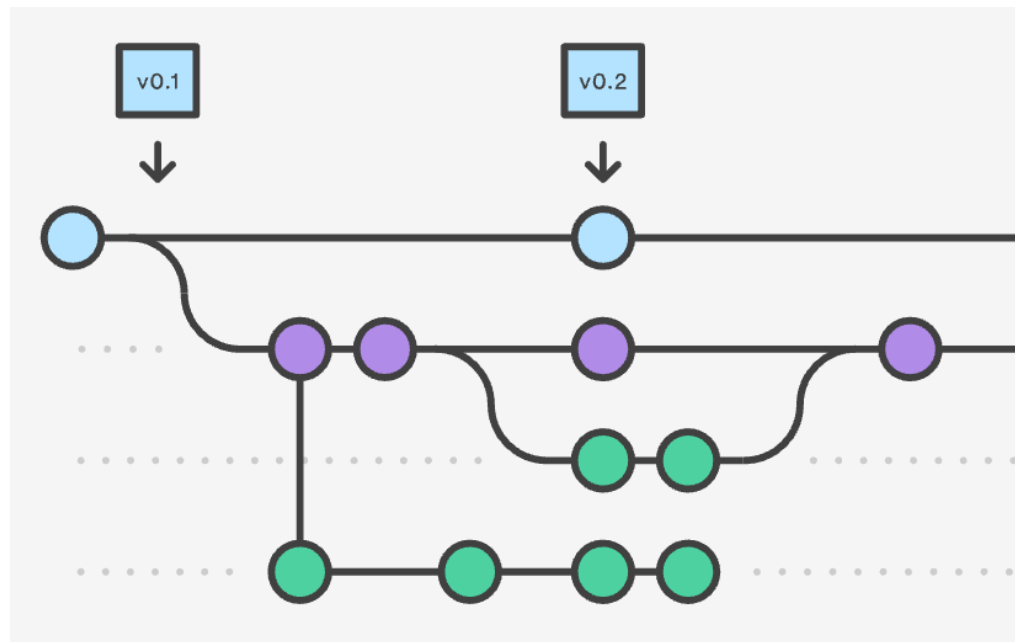
- 버전관리와 커밋
- Push와 Pull
- Fork 와 Clone
- Fetch and merge
- Pull request

# 버전 관리

동일한 정보에 대한 여러 버전을 관리하는 것으로, 팀 단위로 개발 중인 소스코드 등의 디지털 문서를 관리하는데 사용된다.

시간 경과에 따른 파일 또는 파일 집합의 변경사항을 기록하는 접근방식으로 파일의 추가 및 수정이력을 추적 및 관리할 목적으로 사용된다.

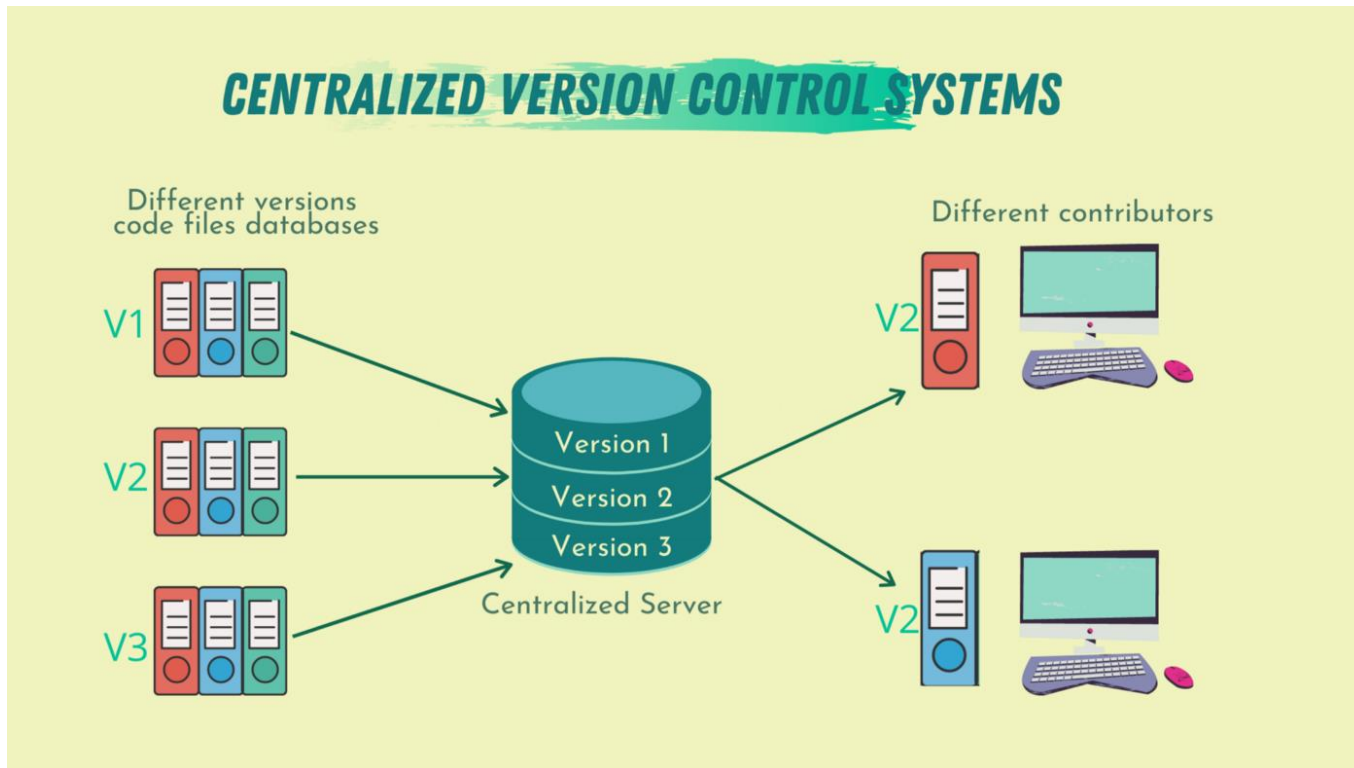
버전을 저장함으로써 필요에 따라 이전 버전으로 되돌려 백업을 할 수 있으며 변경사항을 자세히 확인 할 수 있다.



# 버전 관리의 필요성

소스코드의 변경사항을 보존하므로 과거의 소스 파일 및 디렉토리의 내용을 쉽게 확인할 수 있기 때문에, 버그 및 문제가 발생했을 때 추적하여 해결하는 것에 유용하다.

팀원이 하나의 소스를 가지고 효율적으로 협동작업을 할 수 있다.



# 커바 Commit

저장소의 변경을 저장하는 작업으로, 어느 시점의 파일 및 폴더의 변경사항을 저장소에 기록한다.

시간순으로 저장되어 과거 변경 이력과 내용을 알 수 있다.

프로젝트의 상태를 저장하고 파일이 달라지지 않았으면 새로 저장하지 않는다

이전 상태의 파일에 대한 링크만 저장한다.

hauntsaninja committed 3 hours ago ✓	Verified	<a href="#">/4b35be</a>	<>
stubtest: do not treat py files as source for mypy definitions (#12889) ...	Verified	<a href="#">f08c57e</a>	<>
hauntsaninja and KotlinIsland committed 5 hours ago ✓			
stubtest: add error summary, other output nits (#12855) ...	Verified	<a href="#">e661890</a>	<>
3 people committed 5 hours ago ✓			
Commits on May 27, 2022			
Put filelock around PEP 561 tests (fixes #12615) (#12857) ...	Verified	<a href="#">07edc92</a>	<>
erikkemperman committed yesterday ✓			
[mypy] Replace integer floor division by a power of two with a shift (...)	Verified	<a href="#">d3ef642</a>	<>
JukkaL committed yesterday ✓			
Commits on May 26, 2022			
Fix crash with nested attrs class (#12872) ...	Verified	<a href="#">0054046</a>	<>
JukkaL committed 2 days ago ✓			
stubtest: check typevar and paramspec (#12851) ...	Verified	<a href="#">b55dcf5</a>	<>
hauntsaninja committed 2 days ago ✓			
[mypy] Clean up support for debugging mypyc runtests (#12849) ...	Verified	<a href="#">4024748</a>	<>
pranavrajpal committed 3 days ago ✓			
Disallow assignments to awaited coroutines that do not return (#12853) ...	Verified	<a href="#">87e73fe</a>	<>
hauntsaninja committed 3 days ago ✓			
Commits on May 25, 2022			

# 저장소

## Git repository

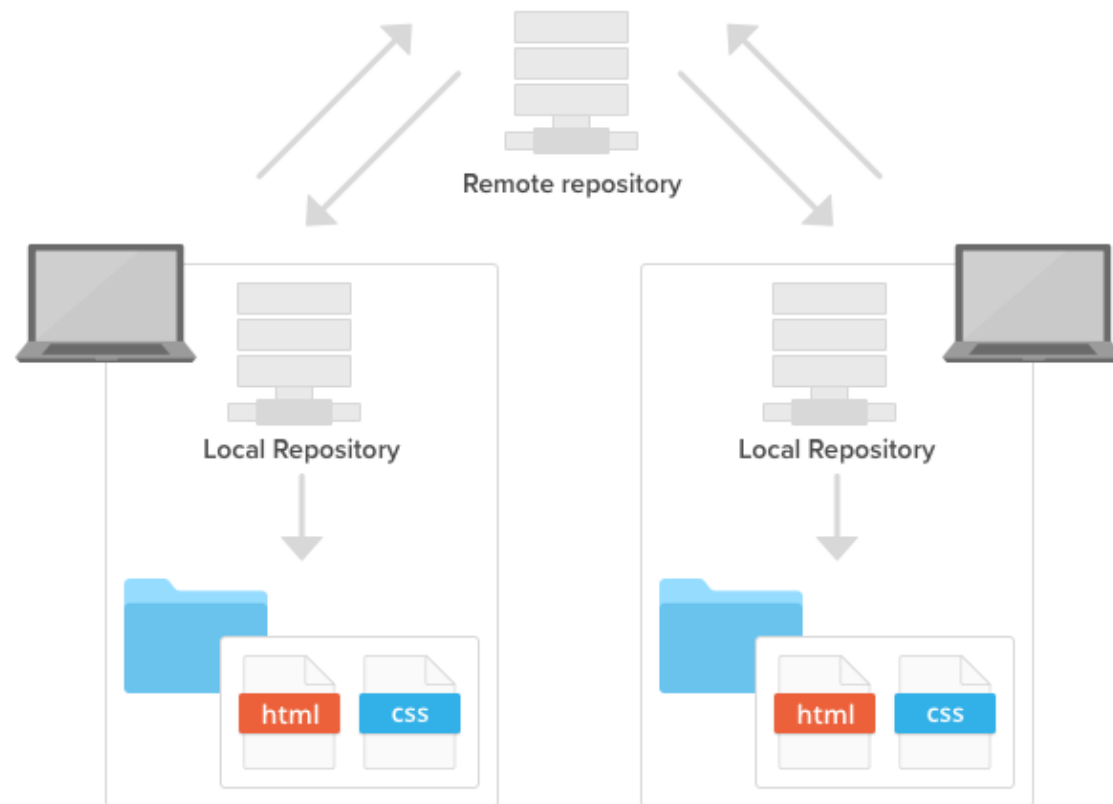
파일이나 폴더를 저장하는 곳으로 파일이 변경 이력별로 구분되어 저장된다.

## Remote Repository

파일이 원격 저장소 전용 서버에서 관리되며  
여러 사람이 함께 공유하기 위한 저장소.

## Local Repository

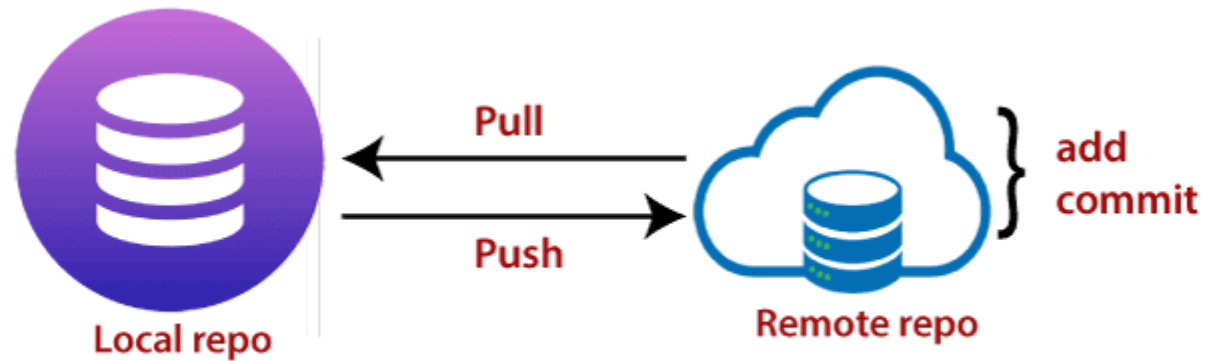
내 PC에 저장되는 개인 전용 저장소



# Push 와 Pull

**Push** : 원격 저장소에 올리는 것  
로컬저장소에서 원격 저장소에 업로드  
원격 저장소에 밀어서(push )올림

**Pull**: 로컬 저장소에 내리는 것  
원격 저장소에서 로컬 저장소로 다운로드  
로컬 저장소로 당겨서(pull) 내림





# Fork 와 Clone

Upstream : 오픈소스 프로젝트의 저장소

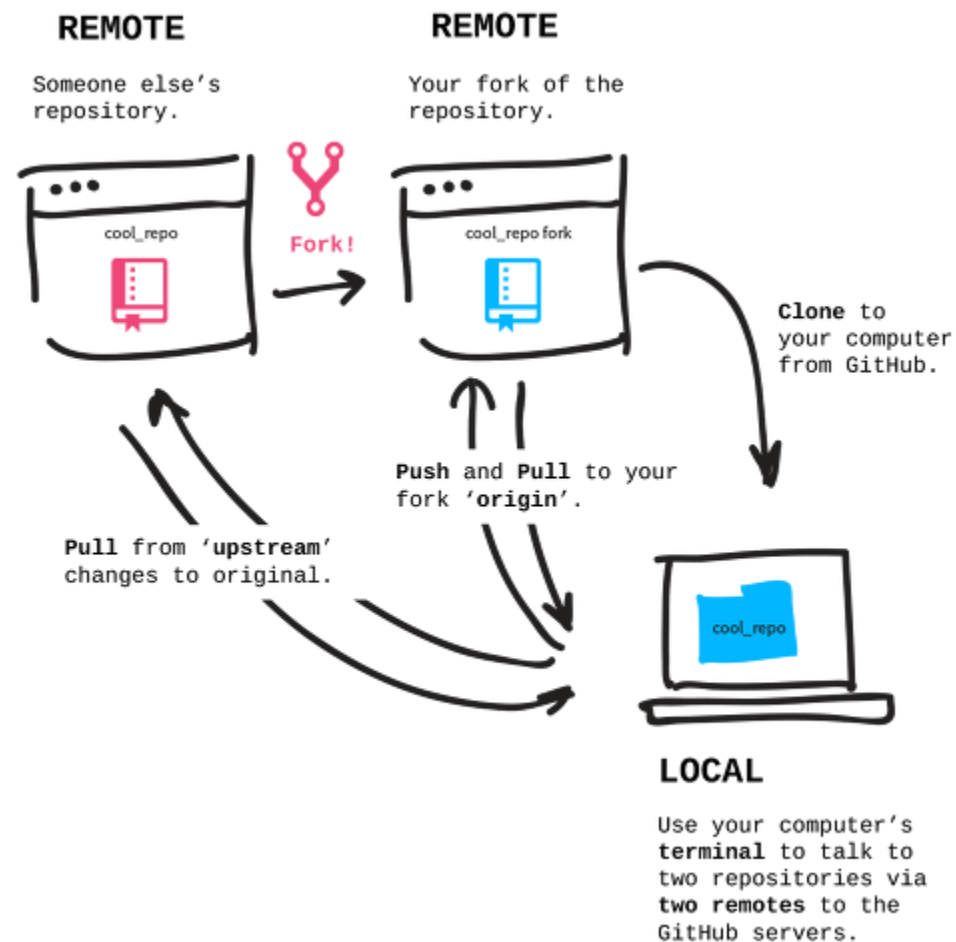
Origin : 나의 github 저장소

Fork : upstream을 나의 origin에 복사하는 작업.

Clone : 나의 origin을 나의 로컬 저장소에 복사하는 작업

로컬 저장소에서 작업 후 pull/push를 통해

변경된 사항을 원격 저장소에 반영



# Fetch and Merge

Upstream 의 변경 사항을 나의 Origin 에 반영하는 작업

main

3 branches

0 tags

Go to file

Add file

Code

This branch is 9 commits ahead of YuHaanSung:main.

Contribute

Fetch upstream

qwert00123 Add files via upload

개인별 졸업작품 기획서	Merge pull request <a href="#">ai7dnn#28</a> f	
선배 졸작 기획 제안 샘플	Add files via upload	
졸업작품 분석설계서 초안	Add files via upload	11 days ago
졸업작품집	Add files via upload	3 months ago
팀별 졸업작품 기획서	2조 졸업작품 프로젝트 기획제안입니다.	26 days ago

✓

This branch is not behind the upstream  
YuHaanSung:main.

No new commits to fetch. Enjoy your day!

Compare

Fetch and merge

# Pull Request

오픈소스 프로젝트나 협업을 하는 과정에서 기여를 하게 된다.

기능 제안 및 추가, 버그 수정, 문서화 작업 등의 기여한 것을 반영할 것을 요청하는 작업이다.

Pull 할 것을 요청(Request)하는 것.

**Contribute**

This branch is 9 commits ahead of YuHaanSung:main.  
Open a pull request to contribute your changes upstream.

**Comparing changes**  
Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base repository: YuHaanSung/2022-Sys-AnD\_3 | base: main | head repository: qwert00123/2022-Sys-AnD\_3 | compare: main

✓ **Able to merge.** These branches can be automatically merged.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#)

**Create pull request**

3 files changed | 1 contributor

**Pull request**

Write | Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

☒ Allow edits by maintainers

**Create pull request**

Verified | a5ea74d

Verified | b9c8ee6

# 소프트웨어 공학

- 소프트웨어 개요
- 소프트웨어의 특징
- 선형 순차적 모델
- V 모델
- 프로토타입 모델

# 소프트웨어 개요

## 프로그램

원시코드를 지칭하며 컴퓨터에서 실행될 때 특정 작업을 수행하는 명령어들의 집합.

특정 문제 해결을 위해 처리방법과 순서를 기술하여 컴퓨터에 입력되는 명령문 집합체이다.

## 소프트웨어

응용 프로그램과 데이터처럼, 컴퓨터의 하드웨어 상에서 구동되거나 처리되는 무형물.

원시코드 뿐만 아니라 자료구조, DB구조, 테스트 결과 등의 모든 산출물과 각 단계마다 생산되는 문서 사용자 메뉴얼 까지 포함하는 매우 포괄적인 개념이다.

# 소프트웨어의 특징

## 제조가 아닌 개발

제조: 정해진 프로세스에 맞춰 균등한 품질로 일정하게 생산하는 것.

많은 인력을 필요로 하고 개인의 능력이 품질에 미치는 영향이 크지 않음.

개발: 정해진 틀에 맞추지 않아도 진행할 수 있다.

개인의 능력에 따른 결과물 차이가 매우 크다

## 소모가 아닌 품질저하

H/W: 사용할수록 부품이 닳고, 고장 발생빈도가 높아지며, 기능이 저하됨.

S/W: 오래 사용해도 부품이 닳지 않고, 고장 발생빈도가 낮으며, 기능도 동일함.

시간이 지나면서 타 소프트웨어에 비해 경쟁력이 떨어짐.

# 소프트웨어 프로세스 모델 - 선형 순차적 모델

## 장점

절차가 간결하여 이해가 쉽다.

단계별 진척 사항에 대한 관리가 용이하다.

체계적인 문서화가 가능하다.

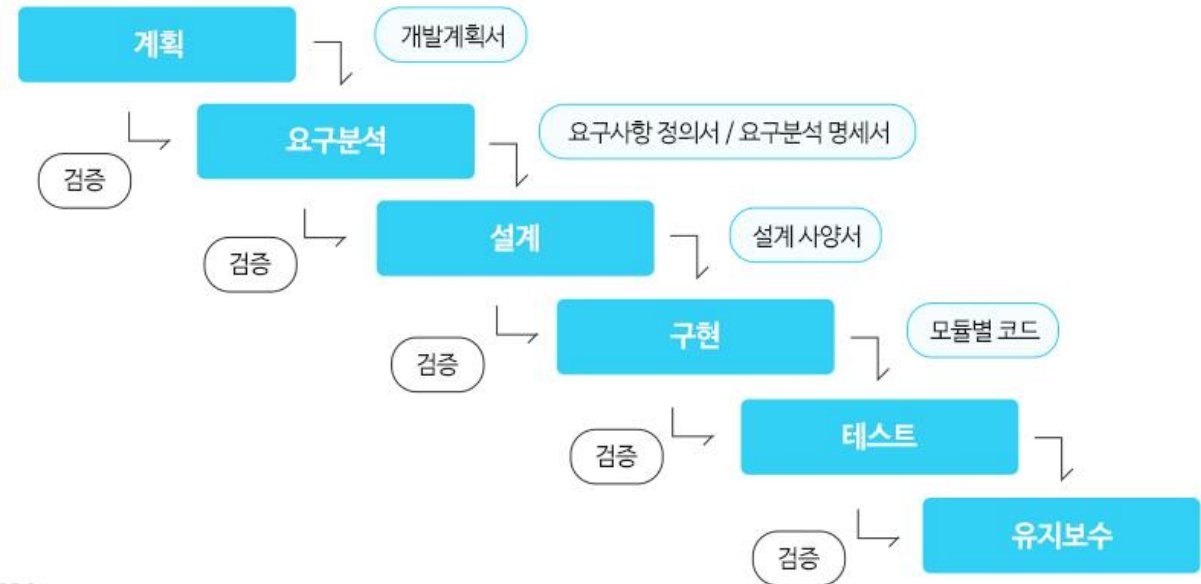
## 단점

사용자 요구 사항이 바뀔 경우 반영이 어렵다.

결과물이 완벽한 수준이어야 다음 단계에 오류를 넘겨 주지 않는다.

사용자가 중간에 가시적인 결과를 볼 수 없어 답답할 수 있다.

초기에 어느정도 요구사항이 확정된 형태의 프로젝트,  
요구 사항의 변동이 없는 형태의 프로젝트에 적합하다.



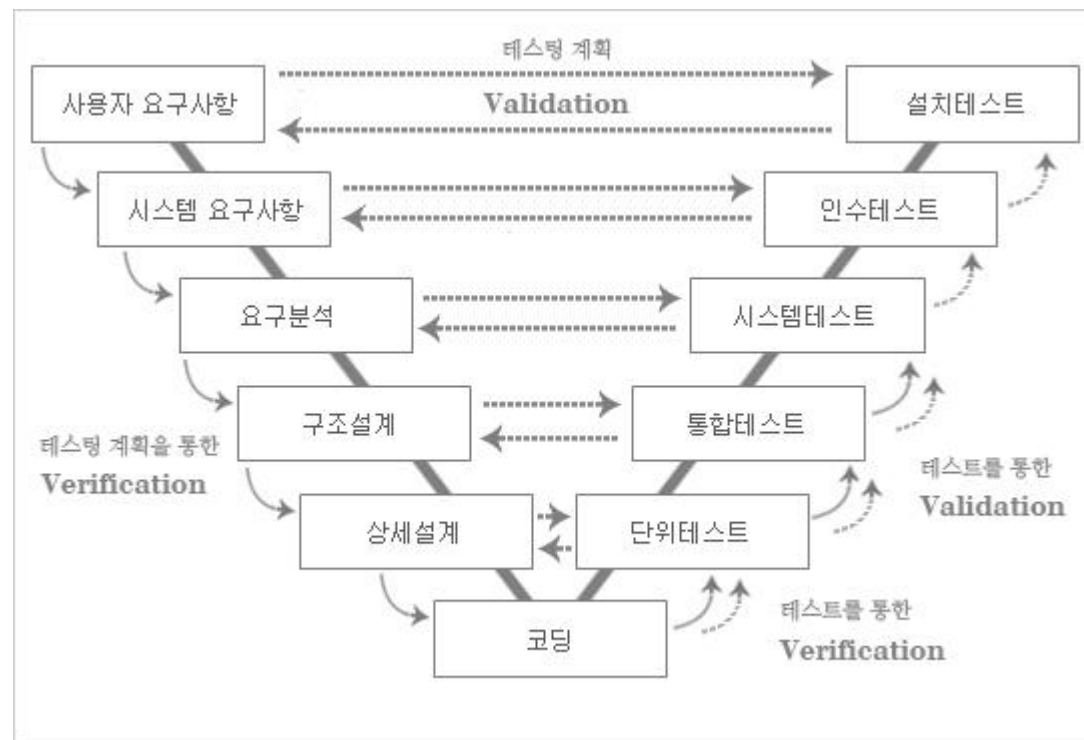
# 소프트웨어 프로세스 모델 - V 모델

폭포수 모델의 확장된 형태 중 하나이다.

아래 방향으로 선형적으로 내려가면서 진행되는 폭포수 모델과 달리,  
코딩 단계에서 위쪽으로 꺾어서 알파벳 V자 모양으로 진행된다.

폭포수 모델에 테스트 단계가 추가되어 확장된 모델이다.

각 개발 단계를 검증하는데 초점을 둔다.





## 소프트웨어 프로세스 모델 - 프로토타입 모델

요구사항을 듣고 설계하며, 평가를 받는 과정을 요구사항 정의 및 분석 과정에서 진행하여 커뮤니케이션 오류를 줄여준다.  
개략적인 요구사항 정의 후 개선을 반복하면서 최종 프로토타입을 개발한다.  
프로토타입은 완전한 설계 대신, 사용자와 대화할 수 있는 수준으로 설계한다.



# MySQL 학습

- MySQL의 구조
- 테이블의 생성
- 컬럼을 편집
- INSERT / UPDATE / SELECT
- 관계형 데이터베이스의 필요성
- JOIN
- 인터넷과 데이터베이스
- MySQL 클라이언트 비교

# MySQL의 구조

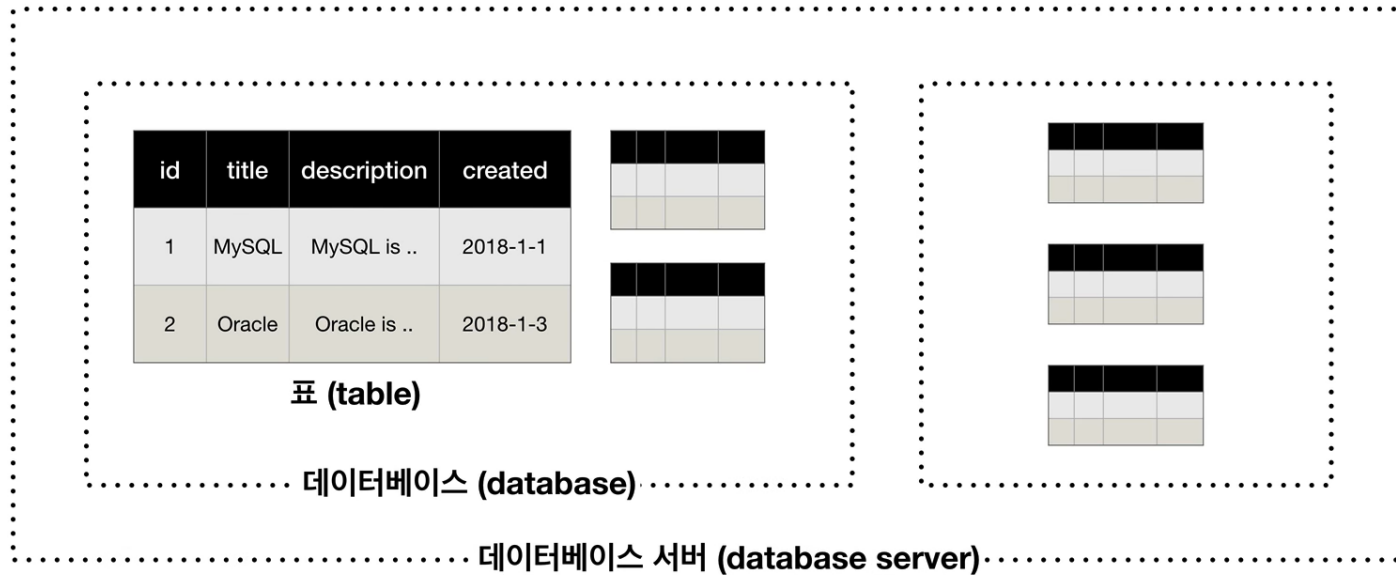


table : 정보가 최종적으로 저장되는 곳

database, schema(MySQL) : 서로 연관된 표들을 그룹화하여 관리하는데 사용

database sever : 데이터베이스들을 저장

# 테이블의 생성

```
1
2 create table book (
3     id int(11) not null auto_increment,
4     title varchar(50) not null,
5     writer varchar(50) not null,
6     primary key (id)
7 );
```

생성하고자 하는 테이블과  
테이블에 속할 컬럼(column),  
컬럼의 자료형을 지정

```
9 ✓ desc book;
```

테이블 book에 속한 컬럼과 그 속성들을 표시

	Field	Type	Null	Key	Default	Extra
1	id	int(11)	NO	PRI	<null>	auto_increment
2	title	varchar(50)	NO		<null>	
3	writer	varchar(50)	NO		<null>	

# 테이블의 컬럼을 편집

```
1  
2 alter table book add country varchar(50) null ;  
3 alter table book add date date null;  
4
```

country와 date 이름을 가진 컬럼을 추가

```
5 alter table book modify title text not null;  
6
```

title 컬럼의 자료형을 text로 수정, null 허용하지 않음

```
7 alter table book change date year int(11) null;  
8
```

date 컬럼의 이름을 year로 변경, 자료형을 int 로 변경

```
9 alter table book drop id;  
10
```

id 컬럼을 삭제

```
11 ✓ desc book;
```

	Field	Type	Null	Key	Default	Extra
1	title	text	NO		<null>	
2	writer	varchar(50)	NO		<null>	
3	country	varchar(50)	YES		<null>	
4	year	int(11)	YES		<null>	

# MySQL 의 INSERT

```
1
2 insert into book (title, writer)
3 values ('죄와 벌', '도스토옙스키'),
4        ('인간실격', '다자이 오사무'),
5        ('1984', '조지 오웰'),
6        ('이방인', '알베르 카뮈'),
7        ('설국', '가와바타 야스나리'),
8        ('라쇼몽', '아쿠타가와 류노스케'),
9        ('카라마조프가의 형제들', '도스토옙스키'),
10       ('위대한 개츠비', 'F.스콧피츠 제럴드');
11
12 ✓ select * from book;
```

	title	writer	country	year
1	죄와 벌	도스토옙스키	<null>	<null>
2	인간실격	다자이 오사무	<null>	<null>
3	1984	조지 오웰	<null>	<null>
4	이방인	알베르 카뮈	<null>	<null>
5	설국	가와바타 야스나리	<null>	<null>
6	라쇼몽	아쿠타가와 류노스케	<null>	<null>
7	카라마조프가의 형제들	도스토옙스키	<null>	<null>
8	위대한 개츠비	F.스콧피츠 제럴드	<null>	<null>

INSERT INTO TABLE\_NAME (COLUMN\_1, COLUMN\_2 ...)  
VALUES (VALUE\_1, VALUE\_2 ...);

title, writer 컬럼에 해당하는 값을 INSERT 를 활용하여 추가

# MySQL 의 UPDATE

```
2 update book
3   set country = 'Japan'
4   where writer = '다자이 오사무' or writer = '가와바타 야스나리';
5
6 update book
7   set country = 'USA'
8   where writer like '%스콧피츠%';
```

UPDATE TABLE\_NAME 을 이용하여 수정  
SET 에 수정할 컬럼 이름과 값을 지정  
WHERE 로 조건을 지정

```
10 ✓ select * from book;
```

	title	writer	country	year
1	죄와 벌	도스토옙스키	Russia	<null>
2	인간실격	다자이 오사무	Japan	<null>
3	1984	조지 오웰	UK	<null>
4	이방인	알베르 카뮈	France	<null>
5	설국	가와바타 야스나리	Japan	<null>
6	라쇼몽	아쿠타가와 류노스케	Japan	<null>
7	카라마조프가의 형제	도스토옙스키	Russia	<null>
8	위대한 개츠비	F. 스콧피츠 제럴드	USA	<null>

# MySQL 의 SELECT

```
1
2 ✓ select title, writer, country from book
3   where not country = 'France'
4   order by title asc
5   limit 4
6 ;
```

SELECT 보고자 하는 컬럼

FROM 테이블 이름

WHERE 조건

ORDER BY 내림 및 오름차순

LIMIT 개수 제한

	title	writer	country
1	1984	조지 오웰	UK
2	라쇼몽	아쿠타가와 류노스케	Japan
3	설국	가와바타 야스나리	Japan
4	위대한 개츠비	F.스콧 피츠 제럴드	USA



# 관계형 데이터베이스의 필요성

데이터가 커지고 많아질수록 데이터의 중복은 다양한 문제를 발생시킨다.

복잡하고 용량이 큰 데이터가 중복된다면 기술적, 경제적 손해이다.

또한 많은 양의 중복된 데이터의 수정이 필요할 때,

관계형 데이터베이스를 사용하지 않는다면 수정이 굉장히 어려울 것이다.

	id	title	description	created	author	profile
1	1	MySQL	MySQL is ...	2022-05-28 17:20:01	Watermelon	developer
2	2	Oracle	Oracle is ...	2018-01-03 13:01:10	Watermelon	developer
3	3	SQL Server	SQL Server is ...	2018-01-20 11:01:10	Peach	database administrator
4	4	Postgre SQL	Postgre SQL is ...	2020-07-13 12:11:03	Mango	developer
5	5	MongoDB	MongoDB is ...	2019-04-07 17:03:10	Melon	data scientist, developer
6	6	Redis	Redis is ...	2022-05-28 17:20:23	Apple	data scientist
7	7	IBM Db2	IBM Db2 is ...	2022-05-28 17:18:17	Melon	data scientist, developer
8	8	SQLite	SQLite is ...	2022-05-28 17:18:17	Apple	data scientist

# 관계형 데이터베이스의 필요성

표를 분리하면 원래의 table의 중복된 데이터는 사라지고  
각각의 데이터들에 대한 author table의 id 값으로 대체가 된다.  
이전의 표와 비교했을 때, 유지 및 보수가 편리해진다.

	id	title	description	created	author_id
1	1	MySQL	MySQL is ...	2022-05-28 17:20:01	1
2	2	Oracle	Oracle is ...	2018-01-03 13:01:10	1
3	3	SQL Server	SQL Server is ...	2018-01-20 11:01:10	2
4	4	Postgre SQL	Postgre SQL is ...	2020-07-13 12:11:03	4
5	5	MongoDB	MongoDB is ...	2019-04-07 17:03:10	3
6	6	Redis	Redis is ...	2022-05-28 17:20:23	5
7	7	IBM Db2	IBM Db2 is ...	2022-05-28 17:18:17	3
8	8	SQLite	SQLite is ...	2022-05-28 17:18:17	

	id	name	profile
1	1	Watermelon	developer
2	2	Peach	database administrator
3	3	Melon	data scientist, developer
4	4	Mango	developer
5	5	Apple	data scientist

# MySQL 의 JOIN

```
2 ✓ select * from topic
3 left join author on topic.author_id = author.id;
```

	topic.id	title	description	created	author_id	author.id	name	profile
1	1	MySQL	MySQL is ...	2022-05-28 17:20:01	1	1	Watermelon	developer
2	2	Oracle	Oracle is ...	2018-01-03 13:01:10	1	1	Watermelon	developer
3	3	SQL Server	SQL Server is ...	2018-01-20 11:01:10	2	2	Peach	database administrator
4	4	Postgre SQL	Postgre SQL is ...	2020-07-13 12:11:03	4	4	Mango	developer
5	5	MongoDB	MongoDB is ...	2019-04-07 17:03:10	3	3	Melon	data scientist, developer
6	6	Redis	Redis is ...	2022-05-28 17:20:23	5	5	Apple	data scientist
7	7	IBM Db2	IBM Db2 is ...	2022-05-28 17:18:17	3	3	Melon	data scientist, developer
8	8	SQLite	SQLite is ...	2022-05-28 17:18:17	5	5	Apple	data scientist

두 table의 결합고리는 topic. author\_id 와 author.id 값

WHERE, ORDER BY, LIMIT 모두 사용할 수 있다.

# 인터넷과 데이터베이스

client 컴퓨터와 sever 컴퓨터가 인터넷을 통해 정보를 주고받는다.

이 관계가 web이라면 web client와 web server가 되는 것과 같이

DB의 작동은 DB client와 sever로 이루어진다.

mysql을 설치하면 두 개의 프로그램을 설치하는데, database client와 database sever이다.

서버에 데이터가 저장되고, 클라이언트를 통해서 서버에 접속할 수 있다.

데이터베이스 서버는 직접 다룰 수 없고,

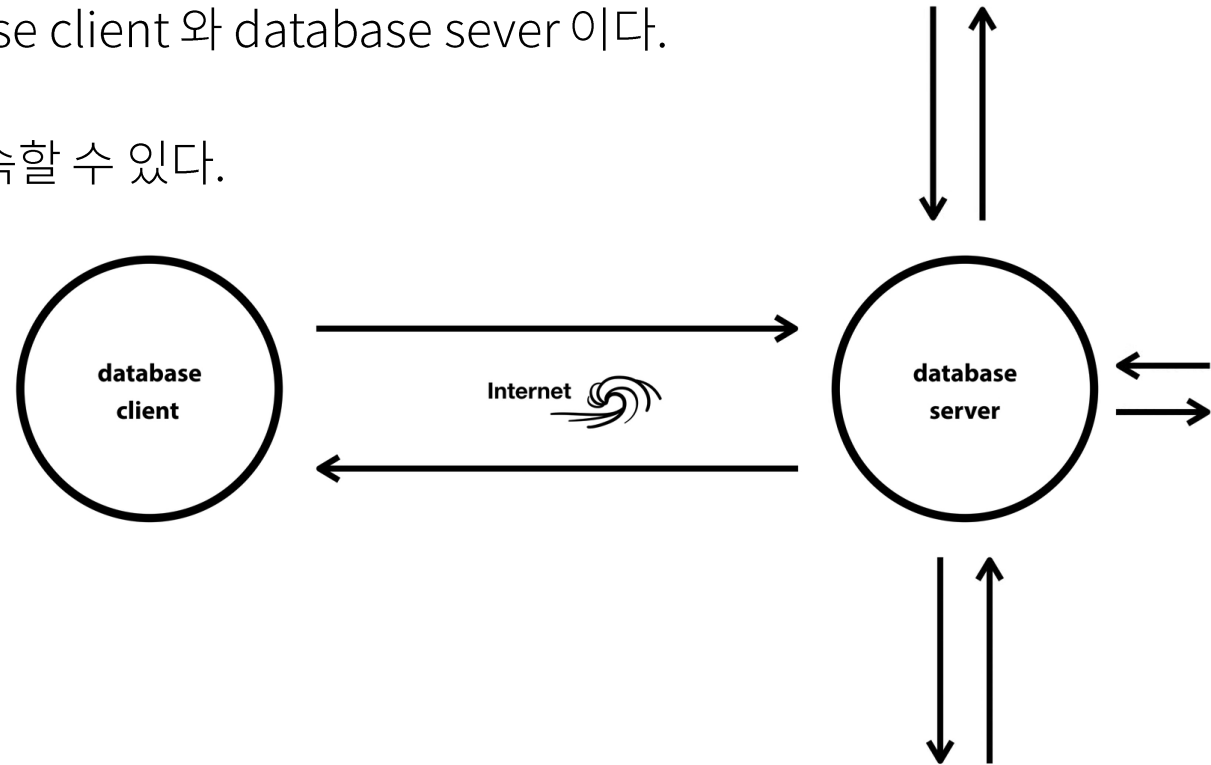
어떤 형태로든 클라이언트를 사용해야 한다.

이러한 구조로 인해 전세계의 수많은 클라이언트들이

하나의 서버를 중심으로

웹이나 앱과 같은 UI를 사용하지 않고도

데이터를 다루는 것이 가능해진다.



# MySQL 클라이언트 비교

## 명령어 기반 (MySQL monitor)

Monitor의 경우, 서버와 함께 설치되기 때문에,  
서버가 있는 곳에 클라이언트가 있다.  
즉, 어디에서나 사용이 가능하다.  
명령어를 기억해야 한다.  
익숙해지기까지 시간이 걸린다.

## GUI 기반 (Workbench)

클릭을 통한 제어로 쉽게 다룰 수 있다.  
서버 컴퓨터들이 컴퓨터의 자원을 일 자체에 투여하기 위해  
GUI 기능을 제공하지 않는 경우가 많다.  
이러한 경우에는 그 컴퓨터에서 실행할 수 없다.

감사합니다