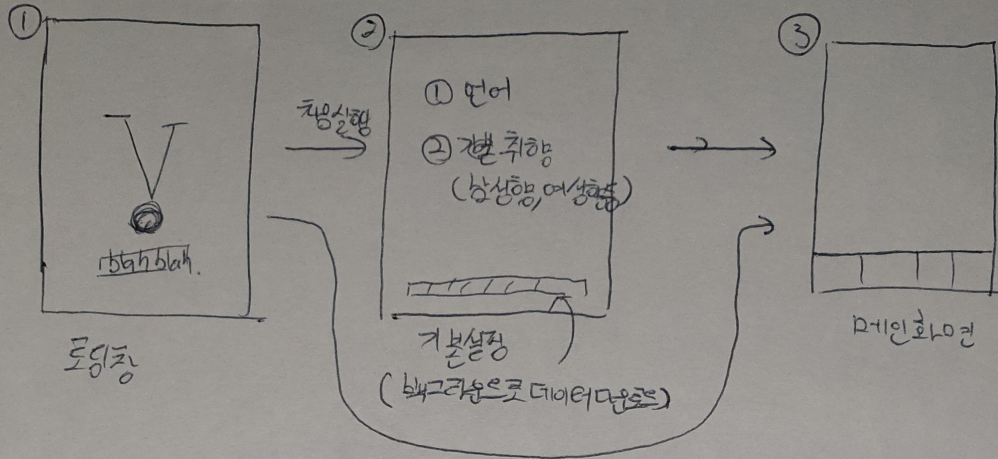
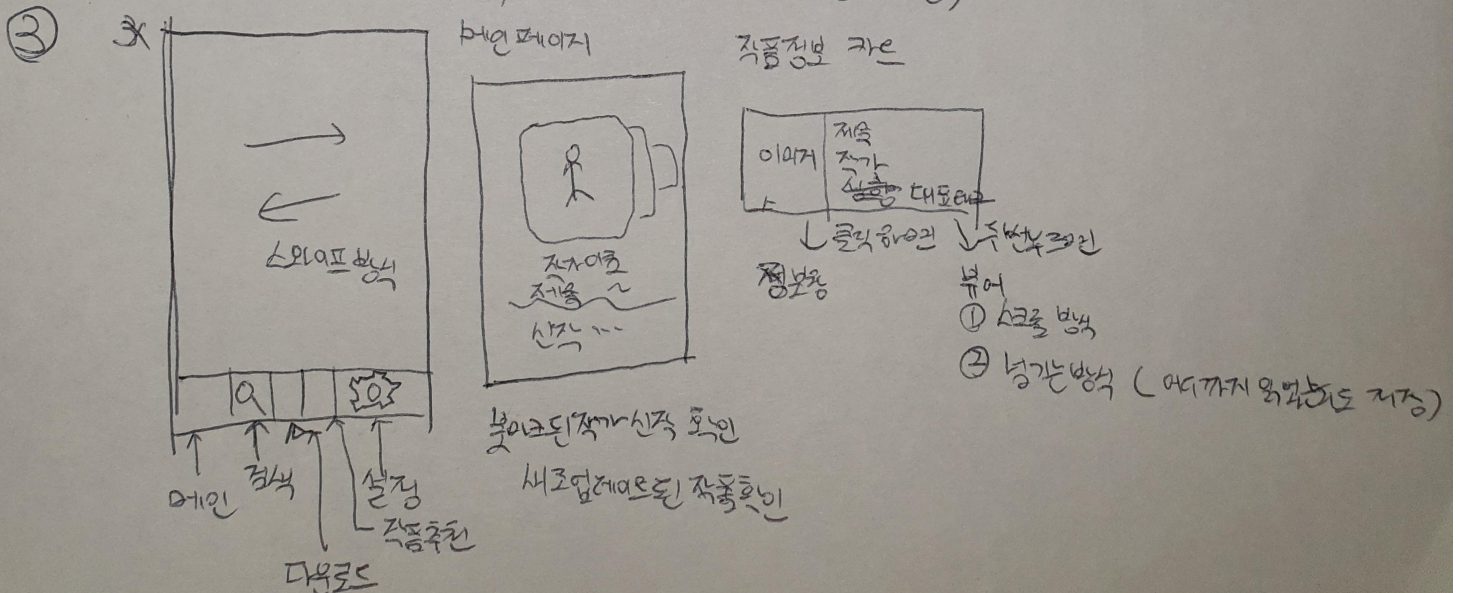


# Project Violet



- ①
  - 1) 처음 실행시 화면의 데이터를 다운로드 해야함 → ②로
  - 2) 다운로드 데이터 로딩 (인덱싱 데이터) ← 데이터 서버쪽에서 초신데이터로 동기화
  - 3) 유저데이터 로딩
  - 4) 북마크된 작가 신작들 불러옴
  - 5) 스크립트 다운로드 (Data Script 나, Lua, JS 등 여러 사용)

- ②
  - 1) 인덱싱 데이터와 작품정보를 다운로드 해야함
  - 어떤 작가의 작품수
    - 태그와 수
    - 등
    - 크기가 작아 빠른 메모리에 로딩
    - 형식
    - 종류가 크기 때문에 불러올 때
    - SQL / ~~JSON~~ / JSON / Message pack 중 하나로 사용
    - 로컬 저장
    - 초신데이터 동기화
  - 2) 사용자 언어 및 제외태그, 취향설정 (디자인 미팅)





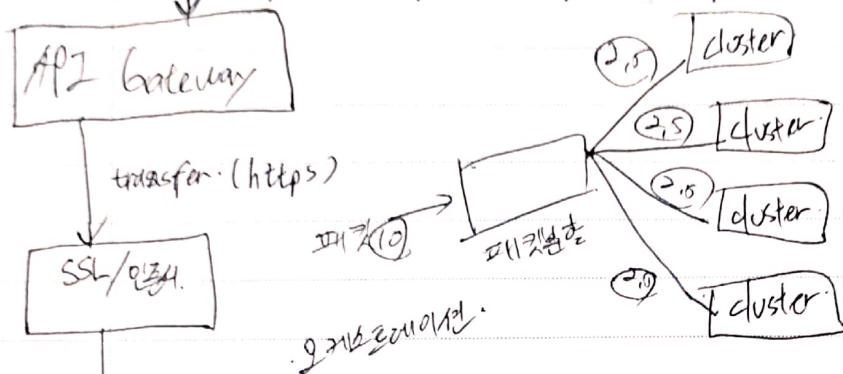
2024 07 07

NO. \_\_\_\_\_  
year month day ( )

고성능 서버 아키텍처 설계

HTTP/HTTPS Raw

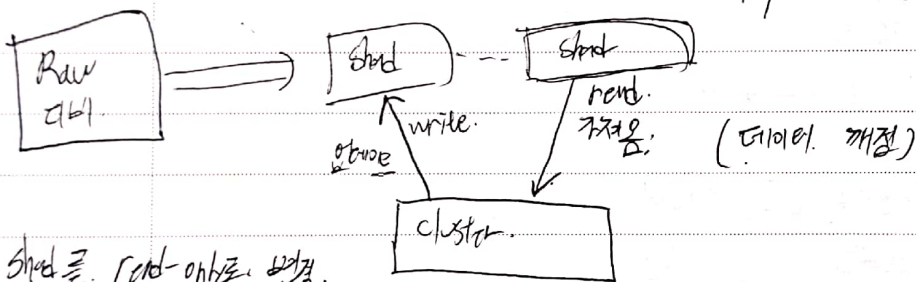
Rachel.



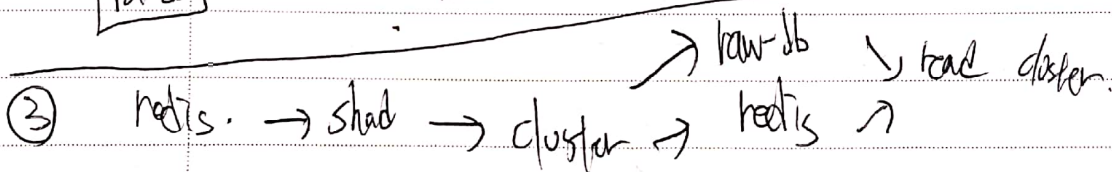
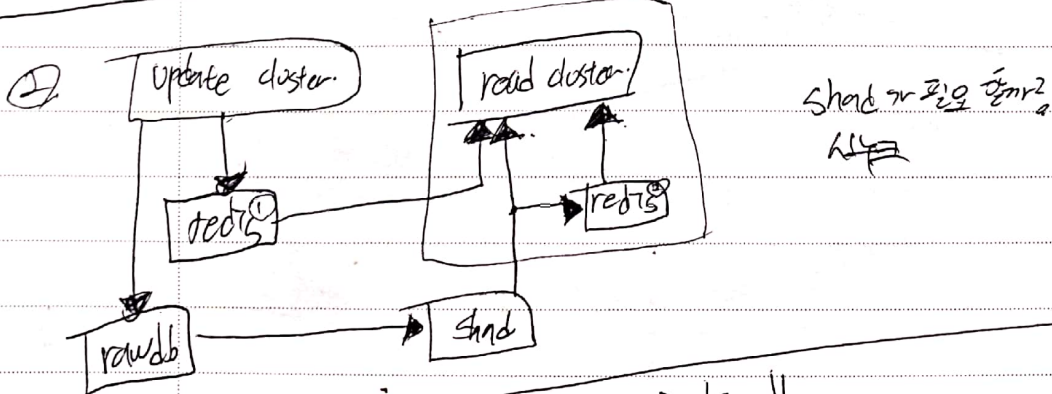
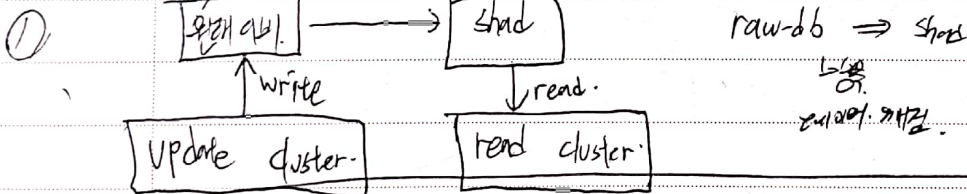
~~클러스터 (4.5) / 클러스터 (4.5) / 클러스터 (4.5) / 클러스터 (4.5)~~

고성능 데이터 처리

~~id~~



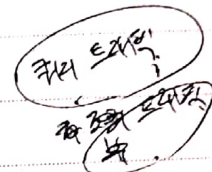
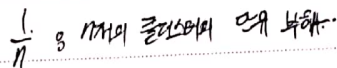
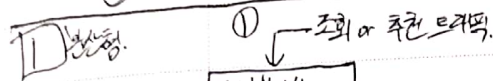
shard 클러스터 read-아키텍처 변경



③

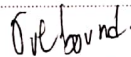
$$\frac{1}{n} \times n^2$$

year month day ( )



Sorted Set

Sorted list





const compare = (a, b) => (a, b) => {  
return

NO.  
year month day ( )

④ var arr = [];

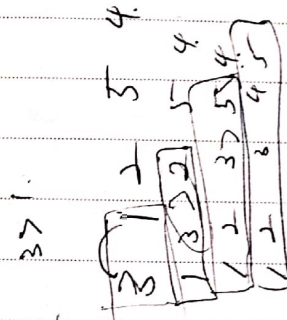
arr[0] = 'zero';

⋮  
arr.splice(a, b) => a-b;

for (var i = 0; i < arr.length; i++)

arr.push('asd');

Array.push() splice()  
unshift() = push - front  
splice()  
pop()  
shift()



Plain text

SHA-256 → 프로그램 암호화 및 데이터 전송  
SHA-512 → 암호화 및 서명  
SHA-256 → AES-256 암호

SHA-256  
AES-256



Real-Time - Ranking:

Daily, Weekly, Monthly, All Time.

Real-Time Update

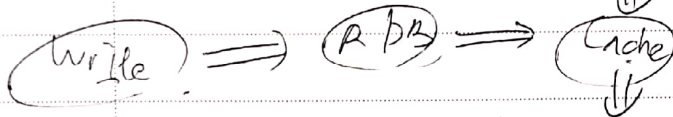
Write - Read - Cache

↓  
100,000    100,000

Sorted Set (4) Query

① ~~Ranking~~ Leader  
②

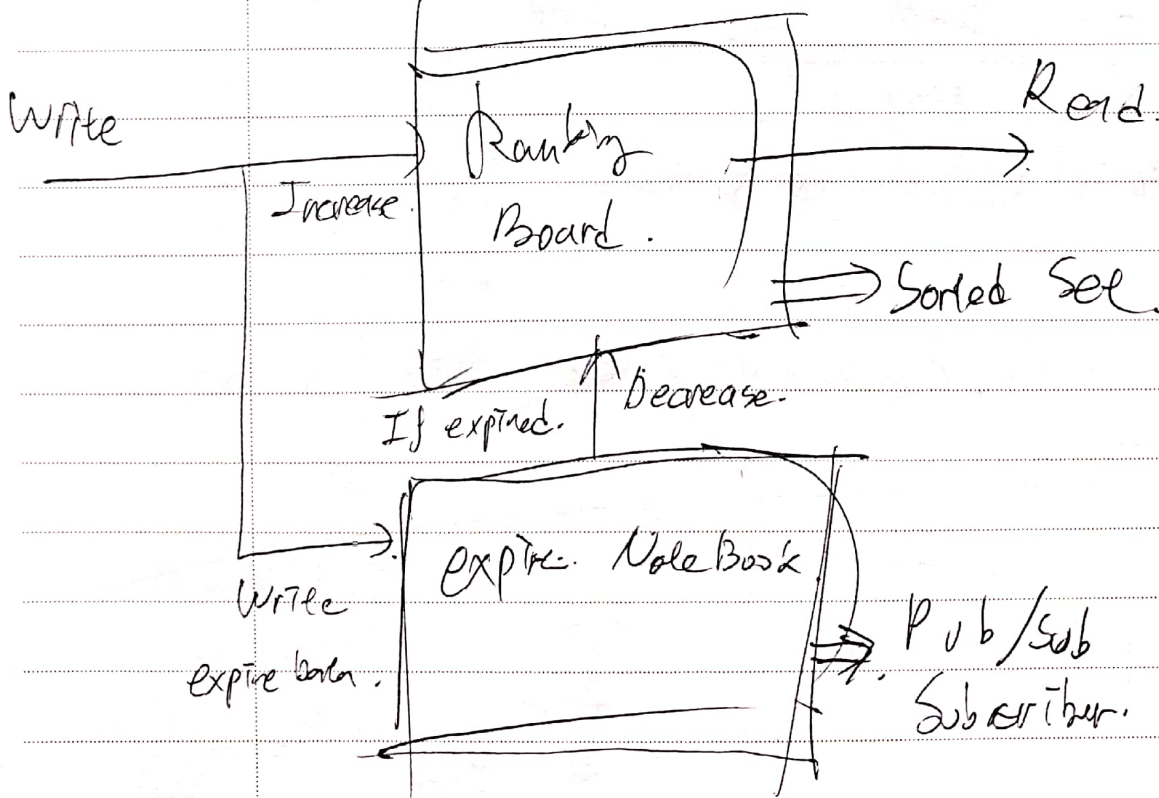
Read



Cache Miss 0%.

Output Data

① Daily ② Weekly ③ Monthly ④ All Time



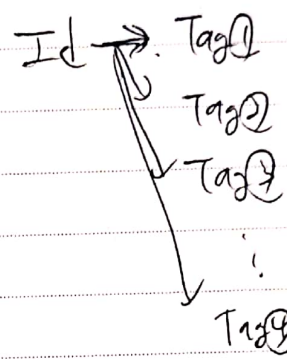
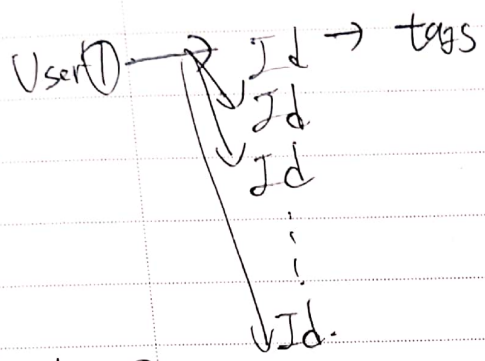
Neural  
style.  
Transfer

SVD →  $\lambda_1, \lambda_2, \dots$

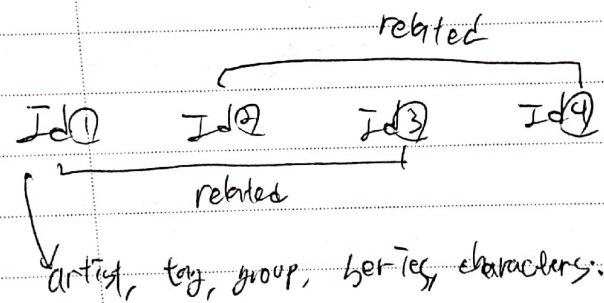
NO.  
year month day ( )

Refresh

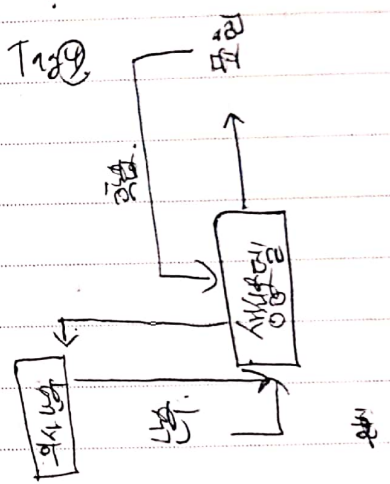
Make Daily, Weekly, Monthly, AllTime



User 2



artist, tag, group, her-ies, characters.



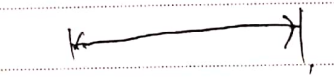
1708.05509.pdf

Trends

60x60x24

36x36x24

144x144x24



864x864

674

Account 14 hours  
 51de 3200



df

TF-IDF

문단당  
단어 빈도

문단별  
중요도

미국  
리볼루션

지명

NO.

9471

year month day ( )

Jarcard, Cosine

Similarity Query

유저 → 작품 집합 (작품, Rank)

유저 = {작품(1), 작품(2), ...}

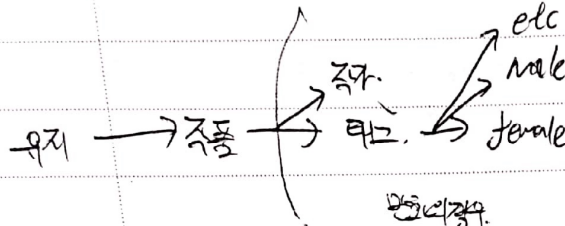
작품의 가중치에 1씩 추가

① 유저 (User)

② 작품 (Article)

③ 태그 (Tag)

최소성



질적분류

문단별 단어 빈도

유저의 작품별 점수  
유저의 작품별 점수, 작품, 태그 점수

작가 → 특징 추출 (문단별, 문장별, 문구별)

다른작가의 작품 → 그림체, 배경화, 작풍, 작품

대역폭

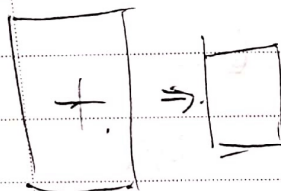


사진 양쪽 끝 큰따옴표 사용

128x128

키워드, 번호 인식 → 그림체 추출

그래서 소개할

그림체 추출

유저 및 순위

작가 ⇒ 대표작의 순위

작가 번호를 줄 나열하는 것

⑥ 정형식, 레이블 서비스. ⑦. 좋은 디자인 서비스 망상인 극대화를 위해.

## II. 웹 서비스

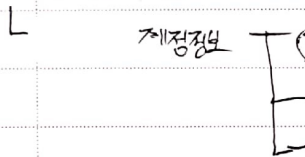
① 데이터 베이스.

② 회원관리.

③ 고객지원(?)

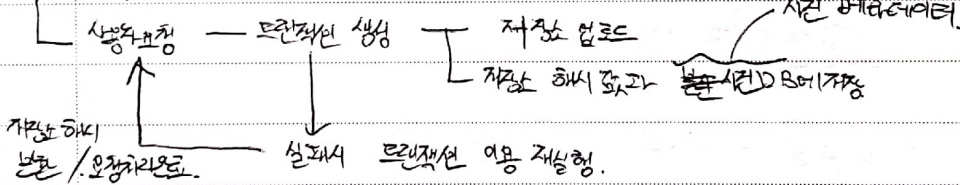
DB 클라이언트 서버로 저장 관련자, 시간응답, 영속성, 맞춤 정보.

비밀번호 전송을 위해 서버상 암호화. 사용.



게시물 작성

사진, 음성, 영상 등 2차 정보의 등록.



저장 해시  
본문 / 동영상 파일

실제 트랜잭션 이용 가능함.

자식들 조회.

Graph 기반 사용자 행동의 SQL 요청.

제. L. 데이터를 처리할 위한 트랜잭션 및. Shad DB 이용.

사용자 연결 서비스.

L. 비슷한 사용자 찾기, 연결하기 기반 사용자 연결 추천, 사용자 ID (전화번호, 이름 등) 이용해 추천연락, 카카오톡, 네이버 ~~카카오톡~~ 서비스 이용

서버 세션

L. 로그인 사용자 정보 저장, 세션 만료 시간, 자식들 조회, 수정, 삭제 관련 인증서.

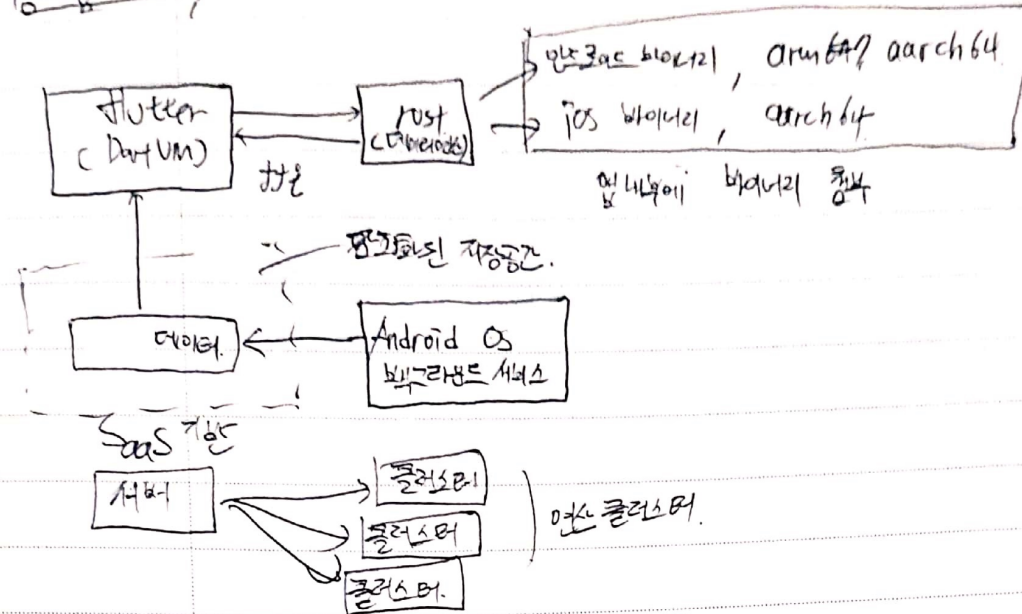


$$\frac{2\sqrt{2}}{154}$$

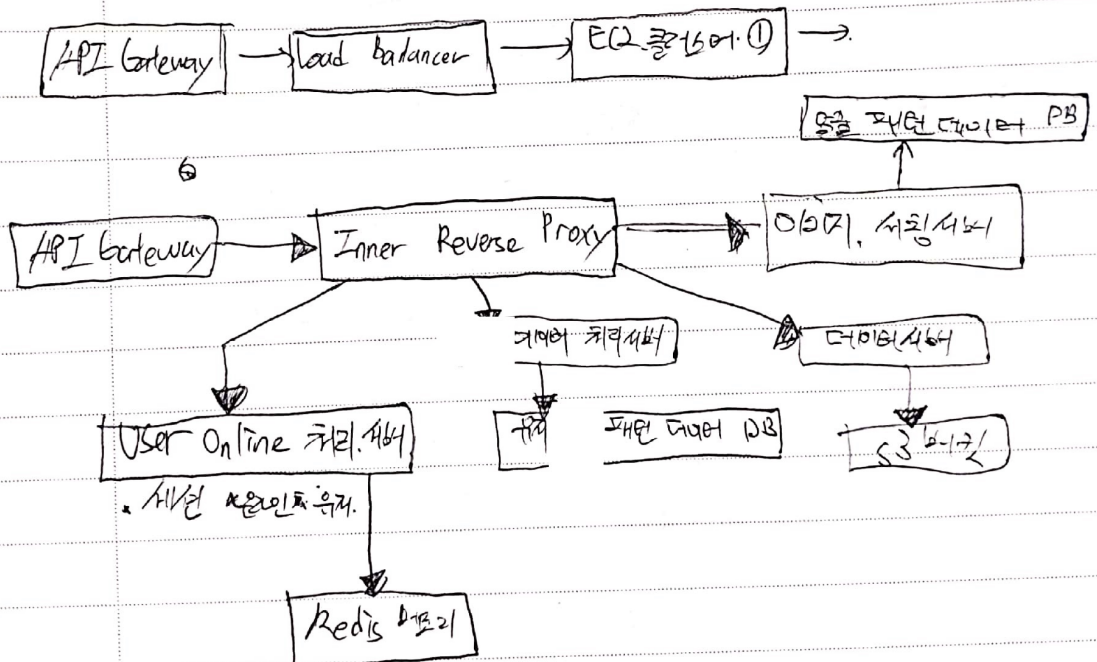
1,440기,  
520년

48

NO. \_\_\_\_\_  
year month day ( )



~~사서~~ 사용자 데이터 저장, 온라인 정보.



20 June

Elavir Search, Red's, Morning DB

동작할 60 시간만 사용하길 2/3로 줄이기  
이유

25 + 200  
NO

year month day ( )  
2020 06 20

상업적(?) 자료. 그해

①. 반박

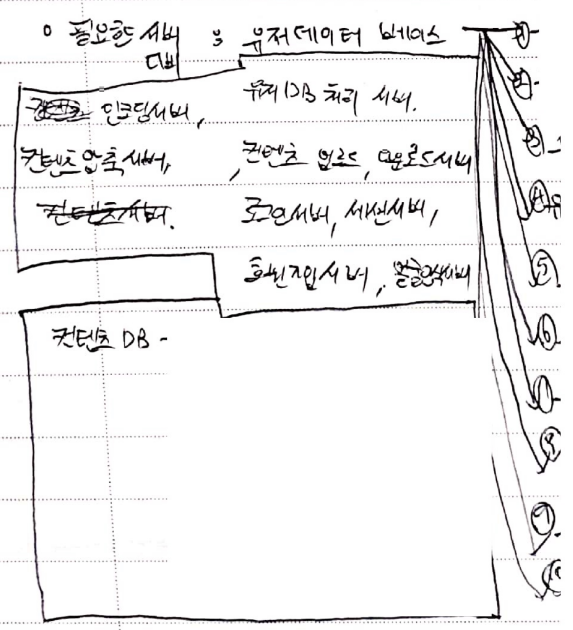
하버드에서 기록한 것 재조사 중인 프로젝트임을 이렇기 때문에 재조사해야  
높은 생산성을 증명해, 인포로이드, IT, 머신러닝, 웹 등 모든 다방면이 급속도로 진행해,  
정확도 향상을 통해 정확도가 불충분한 장점이 있다. ~~이~~ 많은 사람들이  
플라즈마 사용중이며, 미처한 IT와 높은 ~~생산성~~ <sup>생산성</sup>이 특징이다

②. 클라이언트 안전 조치화

애플과 플러머는 내부의 data 바깥에서 동그아 복제본 만든 버전보다 느리고  
비밀 보존해야 한다. Net 기반 하기 큰 장점을 통해, 상용이 필요한 작업에 대해서  
data를 가지고 하기. 연결이 ~~강하다~~ <sup>강하다</sup> 하여. 대외 작업 수행이 가능하다.

③. 서버 설계 및 운용

~~서버~~ 클라우드 서버는 주문대로부터. 대저장까지 ~~보안~~ <sup>보안</sup>으로 쓰이는 서버다. 다양한 서버의  
환경을 비용으로 고정성의 서버를 사용할 수. ~~있다~~ <sup>있다</sup> ~~또한~~ <sup>또한</sup> ~~중요한~~ <sup>중요한</sup> ~~클라우드~~  
~~아기~~ ~~클라우드를~~ ~~사용하는~~ ~~클라우드를~~ ~~사용하고~~ ~~서버를~~ ~~제작하는~~ ~~정보~~ ~~클라우드~~ ~~이~~ ~~아기~~ ~~클라우드~~  
설계하는데 가장 중요하다. ~~로드~~ <sup>로드</sup> ~~밸런싱~~ <sup>밸런싱</sup> ~~서버~~ <sup>서버</sup>.





대시 오버플로우

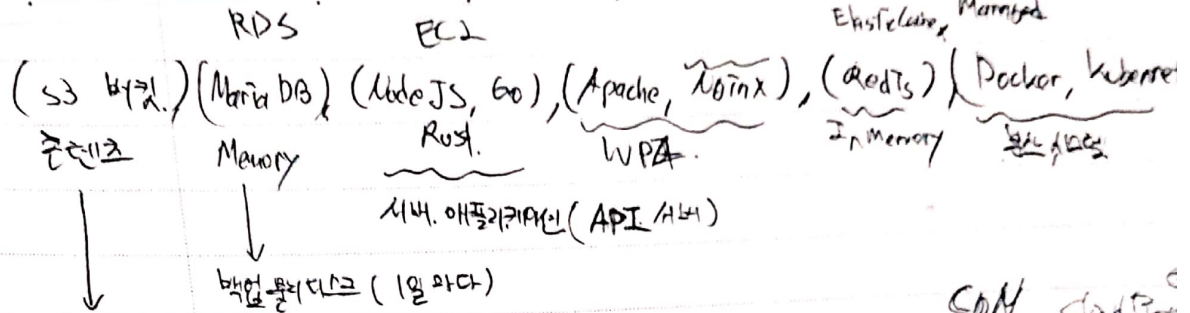
301

NO.

year month day ( )

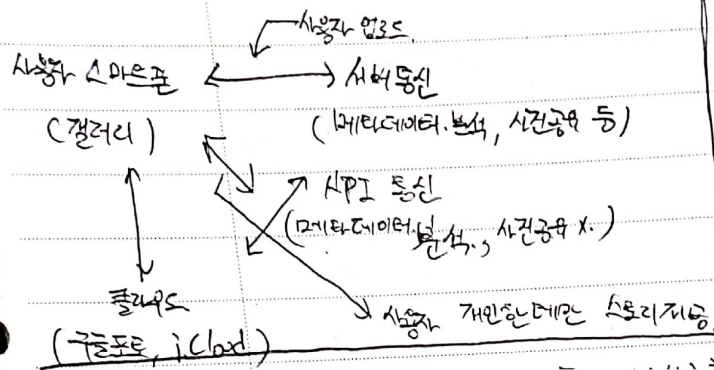
연령상 국대화 ⇒ Rust, 생신상 국대화 ⇒ NodeJS, Go

API Gateway, Cloud Front, CDN ..



백업 디스크 (한달마다)

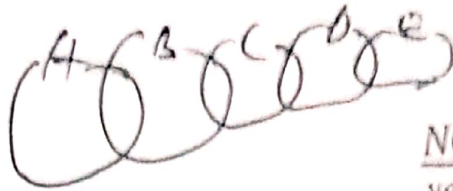
CDN, CloudFront



0,140 \$ per GB  
 $1024 \text{ GB} = 143.36\$ = 1 \text{ TB}$   
 $10 \text{ TB} = 1433.6\$$   
 $1433.6 \times 128 = 183,491.2\$$

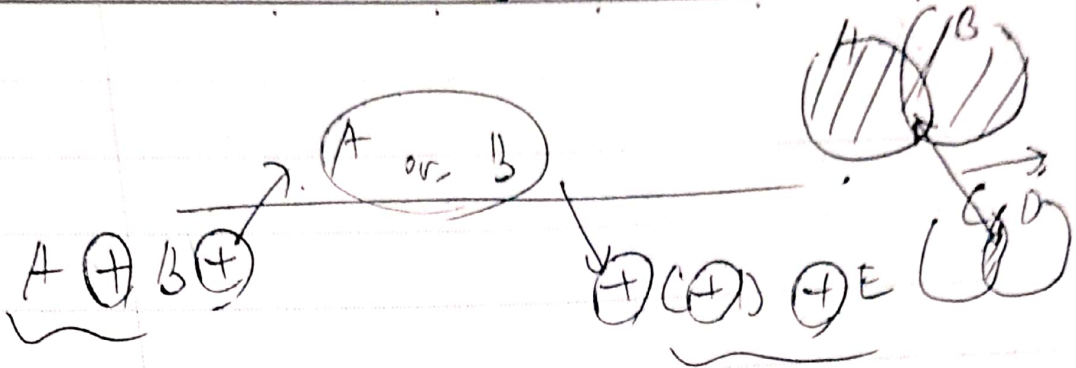
↑  
장분

name by 34



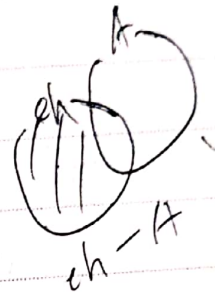
NO. \_\_\_\_\_  
year month day ( )

EH - A - B - C - D



And Join

A, B, C, D, E, A or B and Join



E ((A or B) or (C D))

SQL로 작성하기

E.F. [(A or B) or (C D)]

E.F. (A or B) or (C D)

(A or B) or (C D)



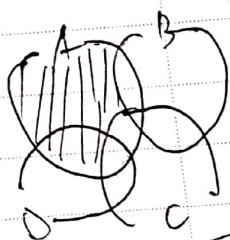
(C Join D)

[A full outer join B]

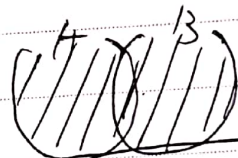
[C inner join D]

full outer join

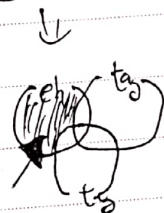
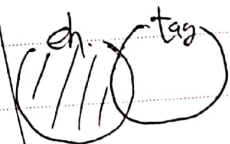
A B - (X Y)



A - B - C + D

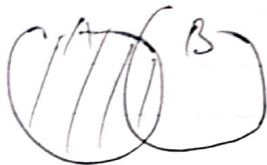


eh - tag - tag

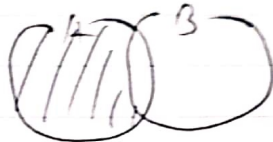




$A \text{ or } B \Rightarrow$  full outer join  
 $A \text{ and } B \Rightarrow$  inner join.



Left Join.



Left Join with,  
 $B.\text{key} = \text{NULL}$



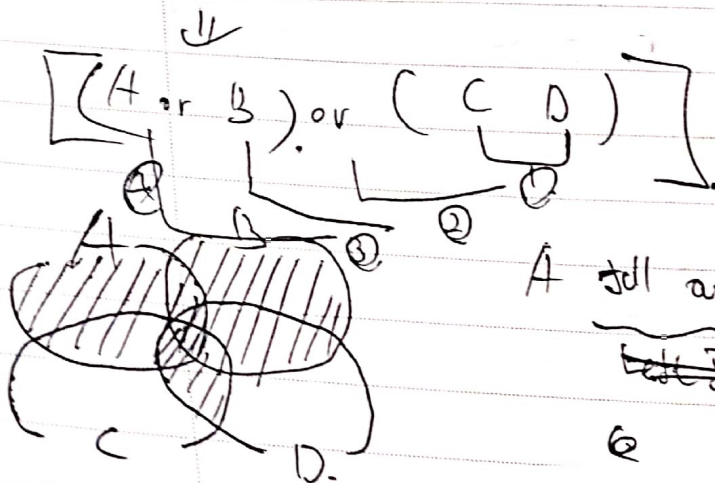
Full outer Join (+)

$A.pk = \text{NULL}$  with

$B.pk = \text{NULL}$ .

$(A \text{ or } B) \text{ or } (C \text{ D})$

optimizer.



$A$  full outer join  $B$

~~Left Join.~~

6

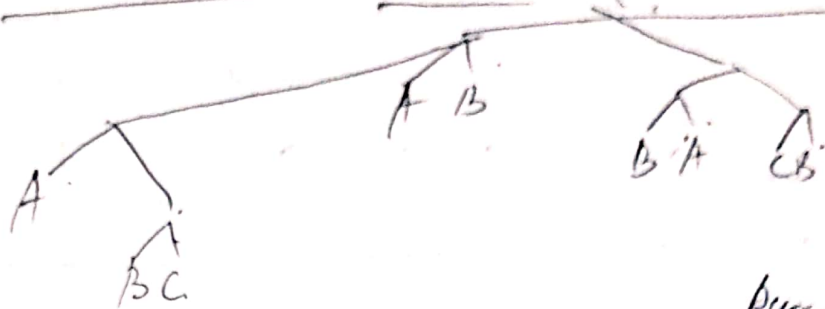
5D

Inner Join.  
Outer Join.

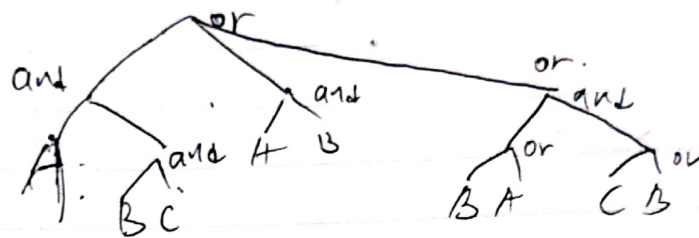
(length of name is length of A)

NO.  
year month day ( )

$(A \text{ and } (B \text{ and } C)) \text{ or } (A \text{ and } B) \text{ or } ((B \text{ or } A) \text{ and } (C \text{ or } B))$



Purple Tree.



Optimizer.

$(A \text{ and } B \text{ and } C) \text{ or } (A \text{ and } B) \text{ or } ((B \text{ or } A) \text{ and } (C \text{ or } B))$

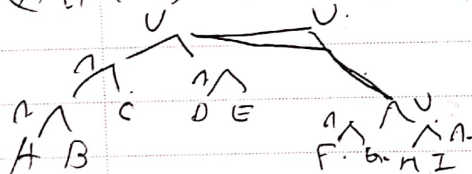
$(A \text{ and } B \text{ and } C) \cup (A \text{ and } B) \cup (B \cup A) \cap (C \cup B) \cap A \cap C$

$(A \cap B) \cup (A \cup B)$

$\textcircled{1} \Rightarrow \textcircled{ALL}$

Set optimizer.

$(A \cap B \cap C) \cup (A \cap B) \cup ((B \cap A) \cup (C \cap B))$



$(A \cap B \cap C) \cup (B \cap A) \cup (C \cap B)$



⑤

$$(A \cap B)^c = A^c \cup B^c$$

year month day ( )

Inter dot  
over

$$(A \cup B) \cup (C \cap D)$$

$$(A \cup B \cup (C \cap D))$$

or and

-X and (Y or)

starts → Var  
1.

(A and

A

$$(A \cup B \cup C) \cap (A \cup B \cup D) \cap (C \cup D \cup E)^c$$

Pattern ①

$$A \cap B \cap (C \text{ or } D \text{ or } E) \cap (F \cap G \cap H) \cap (I \text{ or } J)$$

$$A \cap B \cap (C \cap D \cap E)^c \cap (F \cap G \cap H) \cap (I \cup J)$$

$$(C^c \cap D^c \cap E^c) \text{ converse } \textcircled{1}$$

and merge

① merge landscape and operation.

$$(A \cap B \cap F \cap G \cap H) \cap (I \cup J) \cap (C \cap D \cap E)^c$$

$$(C \cup D \cup E)^c$$

이런데

except tags,

include, exclude. 이걸로

↓

$$(A \cap B \cap F \cap G \cap H) \cap (C^c \cap D^c \cap E^c) \cap (I \cup J)$$

⑤

1

(C.F)

$$(A \cap B)^c = A^c \cup B^c$$

4/30/21

year month day ( )

$$(A \cup B) \cup (C \cap D)$$

$$\downarrow$$
$$\boxed{A \cup B \cup (C \cap D)}$$

$$(A \cup B \cup C) \cap (A \cup B \cup D) \quad \cap (C \cup D \cup E)^c$$

Pattern ①

$$A \quad B \quad \neg (C \text{ or } D \text{ or } E) \quad (F \text{ and } G \text{ and } H) \quad (I \text{ or } J)$$

and  $\downarrow$

$$A \cap B \cap (C \cap D \cap E)^c \cap (F \cap G \cap H) \cap (I \cup J)$$

$$\downarrow \quad \uparrow \quad \uparrow \quad \text{Converse} \quad \text{①}$$
$$(C^c \cap D^c \cap E^c)$$

and merge

① merge landscape and operation.

$$(A \cap B \cap F \cap G \cap H) \cap (I \cup J) \cap (C^c \cap D^c \cap E^c)^c$$

이러한데 except tags,

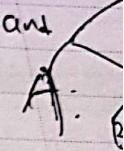
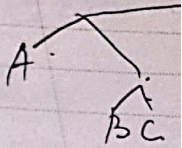
include, exclude. 이걸 다는 것은

$\downarrow$

$$\boxed{(A \cap B \cap F \cap G \cap H) \cap (C^c \cap D^c \cap E^c) \cap (I \cup J)}$$

Inner Join.  
Outer Join.

$$(A \text{ and } (B \text{ and } C))$$



$$\frac{(A)}{(A)}$$

$$(A \cap B)$$

$$\cap A$$



$$\neg(A \vee \neg B \vee C) \Rightarrow \neg A \vee \neg B \vee C$$

NO.  
year month day ( )

expr  $\rightarrow$  term  
1. term (+|-) expr term

term  $\rightarrow$  factor (\*|/) factor  
1 factor

A or B

$$(A^c \vee B^c)$$

factor  $\rightarrow$  token  
1 token  
1 (expr)  
1 factor

$$(A \cap B)^c$$

$$\begin{aligned} C \text{ or } A \text{ or } B & \Rightarrow C \cup A \cup B \\ C \cap (A \cup B) & \\ C \cap A \cup C \cap B & \end{aligned}$$

db, os, ~~serv~~ network, cloud service;  
distributing.

~~tag & format~~

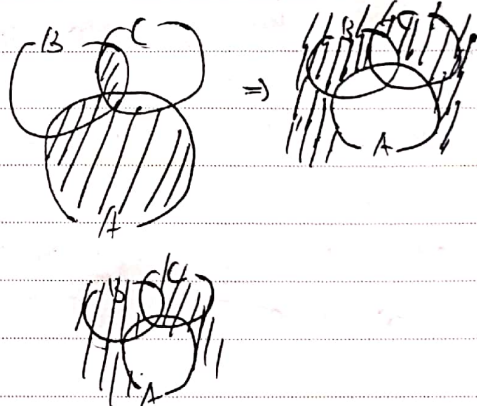
tag & format: vx.

$$\neg(A \text{ or } B) \Rightarrow \neg A \text{ and } \neg B$$

$$\neg A \text{ or } (B \cap C)$$

$$\neg A \vee (B \cap C)$$

$$\neg A \cap (\neg B \vee \neg C)$$



$$\begin{aligned} ① C \cap (A \cup B) & \\ C \cap A \cup C \cap B & \\ ② C \cap A & \end{aligned}$$

Not In  
 A (larger set or  
 larger set)  
 INNER  
 a. JOIN  
 $A \cap B$   
 $A \cap (B \cup C)$

NO.  
 year month day ( )

FULL JOIN  
 FULL OUTER JOIN  
 inner  
 left join A  
 left join B  
 left join C  
 left join D  
 and connecting  
 (type or type)  
 (ch type = or or ch type = x.) A or -B  
 $A \cap B$   
 $A \cap B \cup C$   
 where C.Id IS NULL  
 where C.Id IS NULL  
 V-taken

- or  
 A  
 B  
 U.  
 and -A  
 B.  
 - (A and -B)  
 (A and +B)  
 (-A or -B)

and A  
 and B  
 C.  
 and -A  
 or B





NO.

year month day ( )

(Article, Artist)

Select \* from articles - artist - junction.

where Artist IN (select Id from articles - artist -

where ~~id~~ name = { } " , ... )

Permission -

Select \* from

(select \* from articles - artist - junction.

where Artist = (select Id from articles - artist.

where name = { } ) Order by Article desc.

limit 1

Union.

"

?

~~Id~~ ~~Artist~~ Name.

~~id~~

~~with 5 rows~~

Select all 5 rows.

} "Artists" : [ ~ ],

"Groups" : [ ~ ],

"characters" : [ ~ ],

"series" : [ ~ ],

"tags" : [ ~ ].

}

[

return. ~~Artist~~

"Artists" : [ ~ ],

{ "Name" : <ArtistId> },

{ "Name" : [ ~ ] },

↓

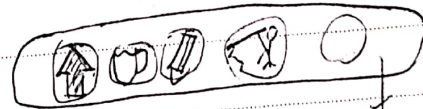
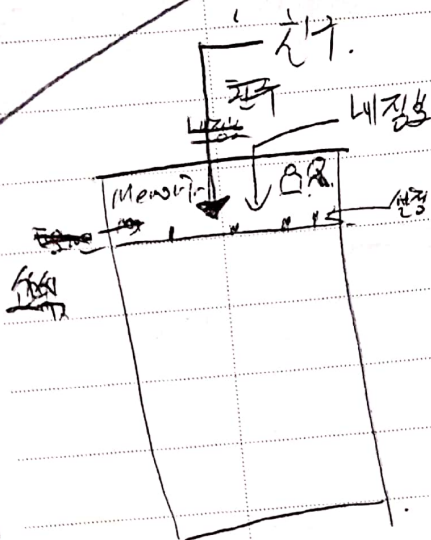
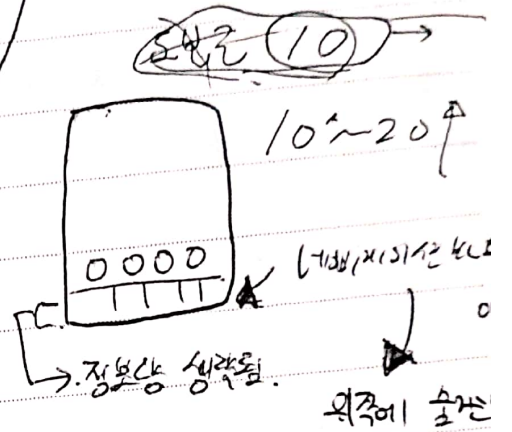
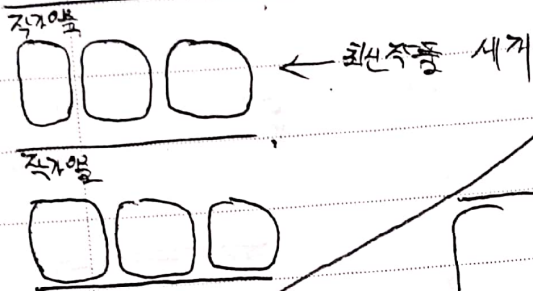


○ ○ ○ ○  
 } 정렬 방법  
 ○ ⊕

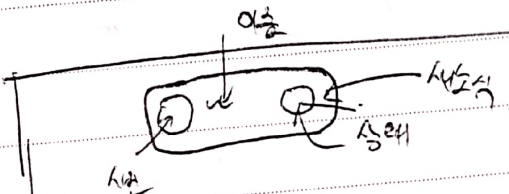
수준 이동 시  
 순서 & 길.  
 Id 번호 정렬

주사, 이동, 최신, 랜덤.  
 (올바름)

직접, 더하기 ~~정렬~~ 필요함 가능.



한국인 전체  
 30%



Q.2.2.2.2, 프로젝트 시작

Memphis.

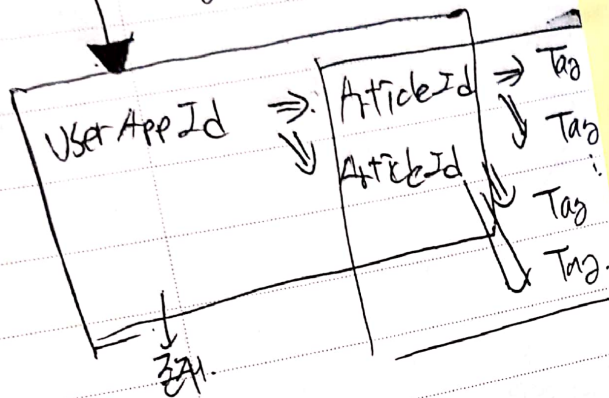
작성 예정  
작성 일자

NO.   
 year month day ( )

articles

articles - artists - junction

View time. - UserAppId, ArticleId  
group



UserAppId, ArticleId.

Tag is ArticleId.

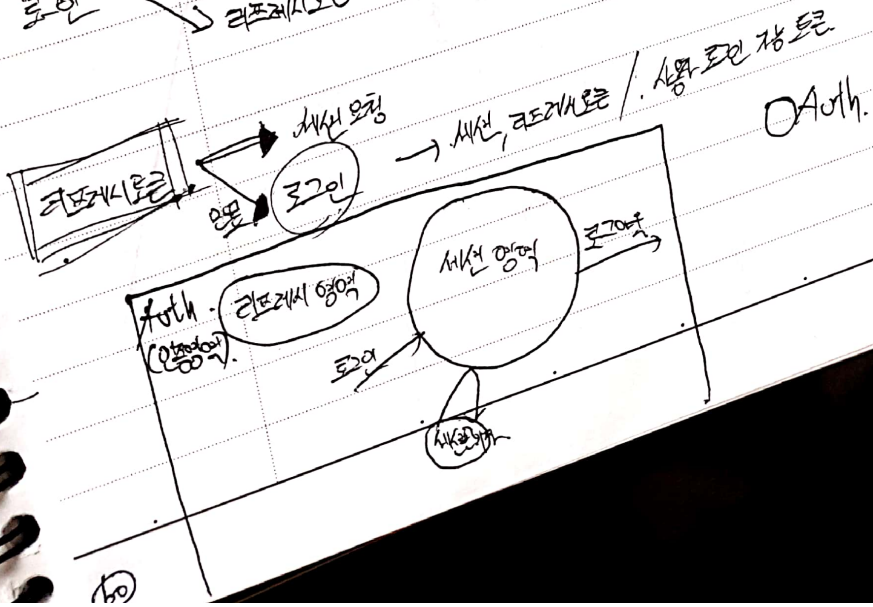
상호작용

전체 사용 - 개별 기록

유한한 정보, 대역폭. 로우 대역로 제한 → 대역폭 초과 발생. 처리량 증가

로그인 → 새션 생성  
→ 리프레시토큰

Sign  
→ in → 새션 생성  
out → 새션 만료  
up → 로그인  
down → 로그아웃

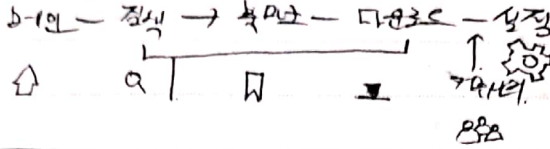
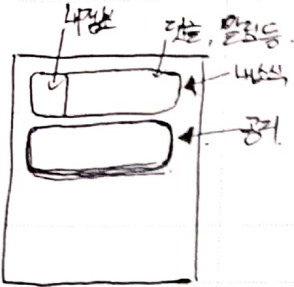


OAuth.



Violet ~~프로젝트~~ TODO LIST

• 커뮤네이션 디자인 : Main 화면에서 댓글 → 댓글리스트 → 댓글보기  
Bottom Navigation Bar에 추가 → 댓글리스트



추천 [주제] 관련 게시물

이슈 게시물 - 알림수신

댓글 게시물

실시간 댓글 게시물

뉴스 관련 게시물

사용자 정보 게시물 : 회원정보 - 비밀번호 - 권한관리 (X)  
↳ 비밀번호변경      ↳ 사용자정보

작가 정보 게시물 (승인)

↳ 작가정보, 성향설치, 댓글

미국의 작가

↳ 그랜저로 복귀  
타코노기? → 생애기록 활동판.  
→ 특정 태그가 포함됨  
특정 태그가 50% 이상 포함됨.  
→ 스토리로 복귀 → 모든데이터를,  
어떤 태그의 사용량/어떤 태그의 비용이 높은지

작가 검색기능 - 작가검색 / 바꿀것

어떤 태그의 사용량/어떤 태그의 비용이 높은지

작품 ①. 작품의 상호. 작가의 상호와 방향  
 작품 ②. 그 반대는 상반인용

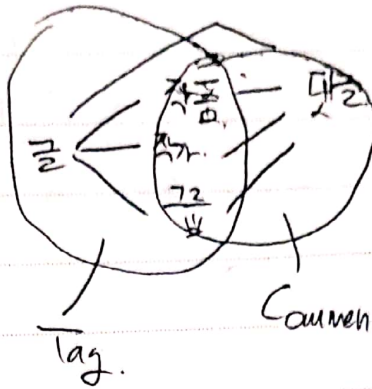
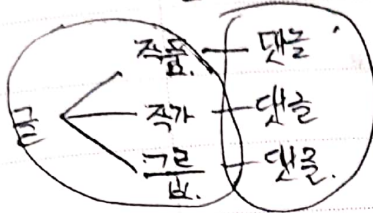
NO.

year month day ( )

단일 작품으로 분류. 작자로 분류. 그림자로 분류

역시 그림체, 풍운소리 (필요 X), 시그 그리는 작가.

글 - 작가 - 작품  
 글 - 그림



새로운 작품이 등장할수록  
 중요한

작품 분류 기준

소리, 그림체

소리, 그림체

소리, 그림체

작가 (그림)

그림체

소리

분류

인물 관계

등장인물

인물

인물

인물

신호그림체, 비신호그림체

작화기법

작화기법

작화기법

작화기법

작화기법

작화기법

작화기법

작화기법

작화기법

작화기법

작화기법

작화기법

작화기법

작화기법

작화기법

작화기법

작화기법

작화기법

작화기법

작품 시퀀스

연속

연속

추진 수

다른 정보

연속

연속

추진 수

다른 정보

연속

연속

추진 수

다른 정보

연속

연속

추진 수

다른 정보

연속

연속

추진 수

다른 정보



이 필터 J.O. 대용, 14 입대자로

0. 커뮤니티 기능. UI 설계 및 추가

블로그 기능

0. 서버 도킹, 테스트

0. 앱 코드. 디자인 코딩

0. 프로토타입 설계 및

0. ~~웹사이트~~ ~~기능~~ ~~추가~~ ~~및~~ ~~이~~ ~~기능~~ ~~가~~ ~~개선~~ (픽시브, 인스타그램, 유튜브 등 외부 사이트)

커뮤니티 기능. UI 설계 및 추가

커뮤니티 기능 → ~~프로토타입~~ ~~디자인~~, ~~회화~~ ~~및~~ ~~이~~ ~~기능~~ ~~가~~ ~~개선~~

↓  
 커뮤니티 디자인  
 업데이트  
 (이차, 안타, 픽시브)  
 무사한

Splash → 회원 가입 에러시 → 유저 ID, PW 보안 확인

기존에 사용하지 않던 ID, PW 사용권장

→ ~~설정~~ ~~중~~ ~~에~~ ~~포함~~ ~~추가~~ ~~(~~ ~~추가~~ ~~항~~ ~~)~~

커뮤니티 탭 | 글 작성 → 제목, 본문 (클릭 후)

태깅 → 작가, 작품, ~~책~~ ~~만~~

댓글 작성 → 태깅 (해시, 작가 작품만 허용)

0. 편의성 개선 → 댓글 링크 클릭시, 자동으로 이동

댓글 형식. 비공식 + ~~댓글~~ ~~같이~~ ~~보여~~ ~~주기~~

(~~댓글~~ ~~인~~ ~~한~~ ~~장~~ ~~위~~ → ~~댓글~~ ~~작성~~ ~~,~~

외부 앱에서 링크 클릭시 이동  
 비공식으로

Sal, olive, dix, zine, ~~the~~, ~~the~~, ~~the~~, ~~the~~, ~~the~~

---

Keun Young C



h x x 2 - 514 501 - 28 2

■ 재발, 덧글 등. 키워드 설정 비밀 (2)

→ 농에 많은 짐들 외에는, 귀뚜라미 꼬리바실,  
한양과 한명 우리 생일 등, 세력·정체질.

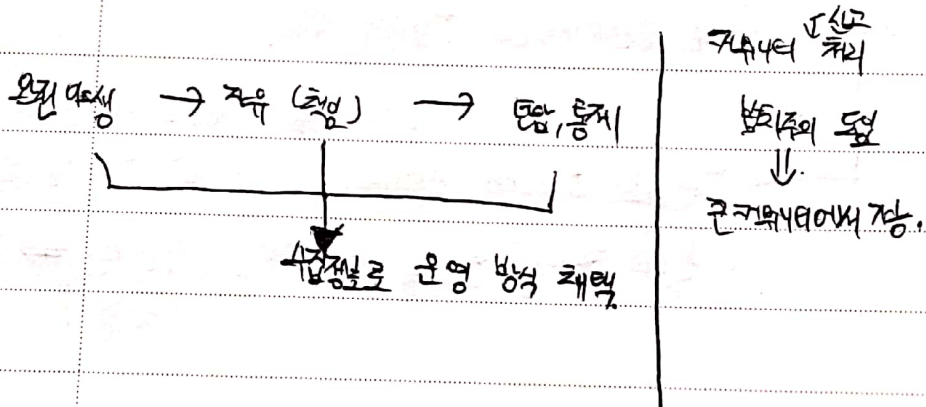
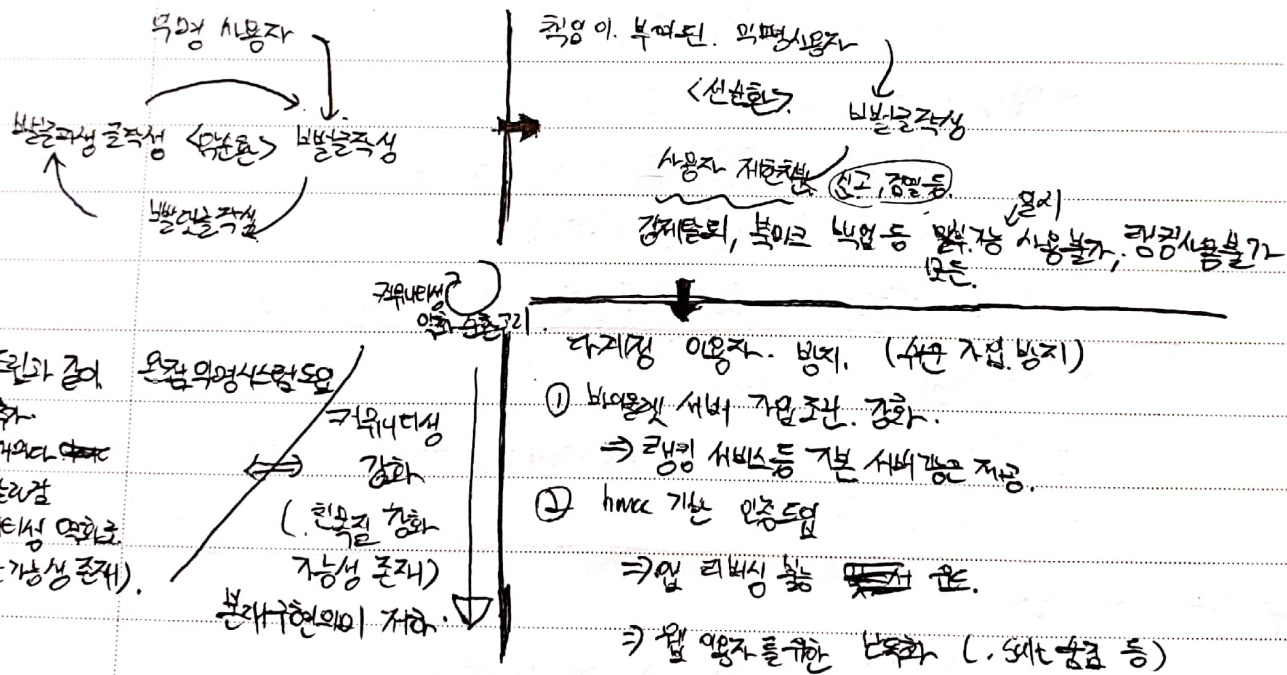
⇒ 4명의 사람들이 2017 대충에 발생하는 원인.

→ 주사광선, 미생물 매개하면 ID, 북미크 등 원생 속제조리

→ 근거·관련 초시, 탄생성 권리. (주제 내 이해하는 것)

(조표, 작자, 원, 복간, 명, 야수 등.)

$\Rightarrow$  ~~이론~~ Q & A ~~은~~ <sup>이</sup> ~~가~~ <sup>가</sup> ~~있~~ <sup>있</sup> ~~다~~ <sup>다</sup> ~~는~~ <sup>는</sup> ~~것~~ <sup>것</sup> ~~이다~~ <sup>이다</sup>



Hybrid 한쌍에서  $\Rightarrow$  30개 개체  
중 40개는  $\frac{1}{2}$

1. 연금준비금 100원

NO.

year month day ( )

1000B = 1MB      1000B = 1MB      1000B = 1MB

사물자 모으기. (현악)

이동자 보충)

07농복3

독립

하루한 김 영웅자 4% → 북미크, 김영웅

ଆପଣେ ପାଠ୍ୟପୁସ୍ତକ ୧୬% - ଅପସ୍ତମ୍ବ ୧୭ ୪୨%

이탈산의 장 → 대변검색, 대변원도

⇒ 자료 구조 및 정리 열거 기능 (검색기능 없음)

- 인삼, 표본 등 기타 방법으로 (독립 사기).

⇒ 카탈라니 다변량더 오픈스호나 (2, 5, 10 예정)

⇒ 적극적 행동이 양호소화. (11.5.10 예제).

- 5월 25일. 화요일.

⇒ 대비 때문에 느끼는 부정적 느낌을 줄

3월 31일 4월 5일

바다를 많이 건넌 (하루는 한 마리)

○ ~~가장~~ 시발점, ~~가장~~ 다양한 개성, 높은 완성도, 커뮤니케이션, ~~가장~~ <sup>높은 완성도</sup>

가까 붙어 있을 (소프트한) 앤 이미지).

• 간편한 뷰어, 볼 것 많을 위한 연, 구독 저장 공유

대중의 생애를 연구한 것은 24년

만일 실시시 유익을 가장 먼저 보여줘야 하므로. — 86일자 정보통신 (3)

(일 정서생, 절정 상황).

- 160자 정보창 (3)

정색충(3),  $\frac{1}{2}$ 이소충(3), 2번

다양한 도덕적 가치관

※ 다음으로 감동제 (1차 제외)은 모든 웹사이트들

다음으로 여기서는 김석재의 이와 동일하게 실장.

→ 생애는 죽음 (마음으로 생애 보아줘야 함)

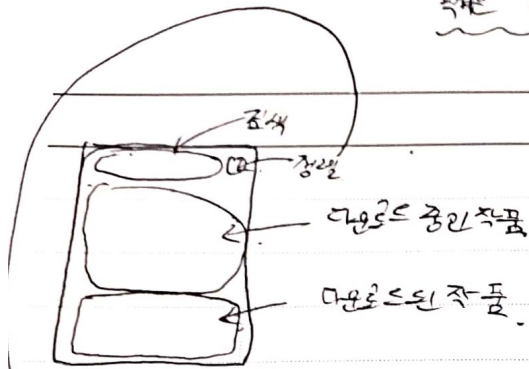
→ CLASS II에서 시작하여: 북마크 페이지 포함.

북.미.코. 동.남.극. 전.세계. 원.전. 개.관 (플루토늄  $\Rightarrow$  질.소)  
(변.환)

Young Co.

Keun'





대문로드 인스턴스

대문로드 프로젝트 다각화

대문로드 엔지니어 (UI) → 사인

연동

태그 (방자)

대문로드 태스크 (PAC)

작업장 피드백

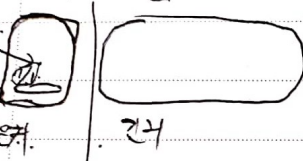
(UI와 관련 스텝에서)

상용 인스턴스

대문로드 태스크 매거

주제 표현

① 인스턴스



표제 22146

주제

인

① 인스턴스  
대문로드 인스턴스  
인스턴스  
표제

비즈니스 인스턴스

대문로드 인스턴스

인스턴스 (SAF)

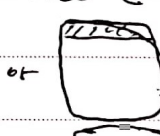
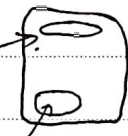
Read Record

표제, 주제, 이미지, 사진, 동영상

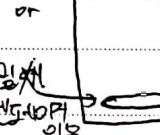
프로젝트 인스턴스

주제

화면



표제



Read Time - 개편

인스턴스. 대기 상태

태스크 인스턴스. 아...

인스턴스

인스턴스

인스턴스

인스턴스