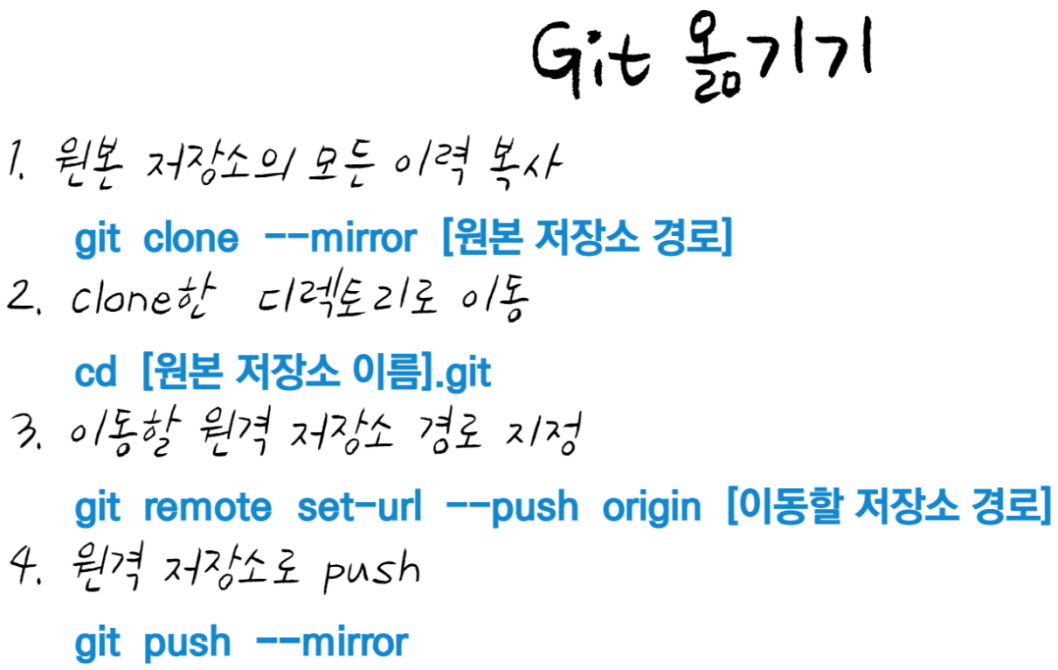
Git flow / NoSQL

Category	
DATE	@2022/03/08
Complete	☐
etc	
열	

Git mirror



Git flow

https://blog.gangnamunni.com/post/understanding_git_flow/

1. 새로운 feature 브랜치를 생성

```
git clone ~
git flow init
엔터 다다다닥
git flow feature start <feature-name>
```

2. merge 후에 브랜치 삭제

```
git flow feature finish <feature-name>
```

- merge했지만 브랜치 기록 남기는 옵션

```
git flow feature finish --no-ff <feature-name>
```

NoSQL

https://khj93.tistory.com/entry/Database-RDBMS와-NOSQL-차이점

> 💡 NoSQL에서는 RDBMS와는 달리 테이블 간 관계를 정의하지 않습니다.
>
> 데이터 테이블은 그냥 하나의 테이블이며 테이블 간의 관계를 정의하지 않아 일반적으로 테이블 간 Join도 불가능합니다.

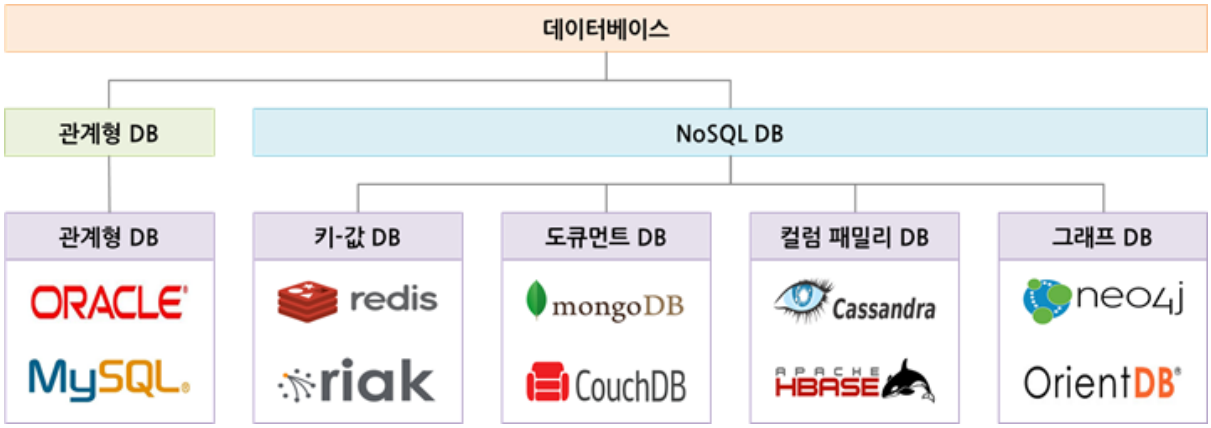
▼ NoSQL vs RDBMS

항목	NoSQL DB	관계형 DB
적합업무	- 오프라인에서 정형 및 비정형 데이터 분석 업무 - 초당 동시 처리가 중요한 업무 - 로그 및 이력 등의 단순 기록형 업무	- 데이터 무결성 및 일관성이 중요한 트랜잭션 업무 - 온라인에서 다양한 집계 및 통계를 분석하는 업무 - 복잡한 계산 및 실시간 데이터 정합성이 필요한 업무
데이터 모델	- 서비스에 맞는 DB 선택이 중요함 - 반정 규화에 의한 설계를 기본으로 함 - 비정형화 스키마 구조로 미리 스키마를 선언하지 않음	- 엔티티 및 각 엔티티 간 관계를 정의함 - 엔티티 정의 시 정규화에 의한 설계가 중요함 - 테이블\, 칼럼 등 DB요소에 대한 스키마를 엄격히 관리함
성능	- 클러스터 크기\, 네트워크 및 애플리케이션에 의해 성능이 결정됨	- 성능 향상을 위해서는 성능 최적화 작업이 필요함
인터페이스	- 쿼리 외 다양한 API를 통한 데이터 저장 및 검색이 가능함	- SQL을 통해서만 데이터 저장 및 검색이 가능함
장점	- 쿼리 프로세싱이 단순화되어 대용량 데이터 처리 성능이 향상됨	- 데이터 중복 배제로 데이터 이상 발생 및 용량 증가를 최소화함
단점	- 데이터 중복에 의해 데이터 일관성이 저하되고 용량이 증가함	- 조인이 복잡한 경우 쿼리 프로세싱도 복잡해져 성능이 저하됨

https://meetup.toast.com/posts/274

- **NoSQL을 언제 사용?**
  - 정확한 데이터 구조를 알 수 없고 데이터가 변경/확장이 될 수 있는 경우에 사용하는 것이 좋습니다. 또한 단점에서도 명확하듯이 데이터 중복이 발생할 수 있으며 중복된 데이터가 변경될 시에는 모든 컬렉션에서 수정을 해야 합니다. 이러한 특징들을 기반으로 Update가 많이 이루어지지 않는 시스템이 좋으며 또한 Scale-out이 가능하다는 장점을 활용해 막대한 데이터를 저장해야 해서 Database를 Scale-Out를 해야 되는 시스템에 적합합니다.
  - scale-up, scale-out 이란? https://tecoble.techcourse.co.kr/post/2021-10-12-scale-up-scale-out/

NoSQL 종류



1. 키-값 데이터베이스

- 키와 값으로 구성된 배열구조의 데이터베이스로 NoSQL 데이터베이스 중 가장 단순한 구조

▼ 언제 사용?!

https://velog.io/@swhan9404/NoSQL-의-종류별-특징

1. 성능 향상을 위해 관계형 데이터베이스에서 데이터 캐싱
2. 장바구니 같은 웹 애플리케이션에서 일시적인 속성 추적
3. 모바일 애플리케이션용 사용자 데이터 정보와 구성 정보 저장
4. 이미지나 오디오 파일 같은 대용량 객체 저장

https://happyer16.tistory.com/entry/레디스Redis의-다양한-활용-사례

1. 좋아요 처리

2. 일일 방문자 수 구하기
3. 출석 이벤트 구현
4. 최근 검색 목록

▼ Redis

Redis(REmote DIctionary Server)

https://meetup.toast.com/posts/224

- 레디스는 오픈소스이고 **In-Memory 데이터베이스 ⇒** 즉, 모든 데이터를 메모리에 저장하고 조회
- 관계형 데이터베이스(Oracle, MySQL) 보다 훨씬 빠름!!
  - 그 이유는 메모리 접근이 디스크 접근보다 빠르기 때문
- 다른 In-Memory 데이터베이스(ex. Memcached) 와의 가장 큰 차이점
  - String, Bitmap, Hash, List, Set, Sorted Set 등 다양한 자료구조를 제공
  - 높은 버전은 Geospatial Index, Hyperloglog, Stream 등의 자료형도 지원
    - **Sorted set :** 실시간 랭킹 서버를 사용하는데 알맞음
    - **String** : **모든 종류의 문자열**(이진 데이터 포함) 을 저장 가능.

      따라서 **JPEG** 이미지를 저장, **HTML fragment 를 캐시**하는 용도로 자주 사용.

      저장가능 최대 사이즈 **512MB**
- 이러한 자료구조를 Key-Value 형태로 저장
- 에어비엔비, Instagram, 쿠팡에서도 사용

2. 도큐먼트 데이터베이스

- 필드와 값의 형태로 구성된 데이터를 JSON 포맷으로 관리하는 데이터베이스

▼ 특징

- NoSQL 데이터베이스 중 가장 인기가 높다
- Documnet Database 데이터는 Key와Document의 형태로 저장된다. Key-Value 모델과 다른 점이라면 Value가 계층적인 형태인 도큐먼트로 저장된다는 것이다. 객체지향에서의 객체와 유사하며, 이들은 하나의 단위로 취급되어 저장된다. 다시 말해 하나의 객체를 여러 개의 테이블에 나눠 저장할 필요가 없어진다는 뜻이다.

- 주요한 특징으로는 객체-관계 매핑이 필요하지 않다. 객체를 Document의 형태로 바로 저장 가능하기 때문이다. 또한 **검색에 최적화**되어 있는데, 이는 Ket-Value 모델의 특징과 동일하다. 단점이라면 사용이 번거롭고 쿼리가 SQL과는 다르다는 점이다. 도큐먼트 모델에서는 질의의 결과가 JSON이나 xml 형태로 출력되기 때문에 그 사용 방법이 RDBMS에서의 질의 결과를 사용하는 방법과 다르다

▼ 언제 사용!?

https://velog.io/@swhan9404/NoSQL-의-종류별-특징

1. 대용량 데이터를 읽고 쓰는 웹 사이트용 백엔드 지원
2. 제품처럼 다양한 속성이 있는 데이터 관리
3. 다양한 유형의 메타데이터 추적
4. JSON 데이터 구조를 사요하는 애플리케이션
5. 비정규화된 중첩 구조의 데이터를 사용하는 애플리케이션

▼ mongoDB

- mongoDB 특징과 쿼리: https://meetup.toast.com/posts/275

항목	mongoDB	관계형 DB
입력	db.member.**insert**({ no : "C001", age : 45, status : "A" })	**insert** into member(no, age, status) values('C001', 45, 'A')
수정	db.member.**update**({ age : { $gt : 25 } }, { $set : { status : "C" } }, { multi : true })	**update** member set status = 'C' where age > 25
삭제	db.member.**remove**({ no : "C001" })	**delete** from member where no = 'C001'
조회	db.member.**find**({ no : "C001" })	**select** * from member where no = 'C001'

- mongoDB 모델링
  - https://meetup.toast.com/posts/276
  - https://kciter.so/posts/about-mongodb

3. 칼럼 패밀리 데이터베이스

▼ 칼럼과 로우로 구성된 데이터베이스

- 칼럼은 이름과 값으로 구성
- 로우는 각기 다른 칼럼으로 구성이 가능

- 이렇게 저장된 데이터는 하나의 커다란 테이블로 표현이 가능

4. 그래프 데이터베이스

- 노드와 관계로 구성된 데이터베이스

▼ 특징

- 연관된 객체를 모델링할 목적으로 설계되었음
- 개체와 관계를 그래프 형태로 표현한 것이므로 관계형 모델이라고 할 수 있음

▼ 언제사용?!

데이터 간의 관계가 탐색의 키일 경우에 적합하다. 페이스북이나 트위터 같은 **소셜 네트워크**에서(내 친구의 친구를 찾는 질의 등) 적합하고, **연관된 데이터**를 추천해주는 **추천 엔진(연관검색어)**나 패턴 인식 등의 데이터베이스로도 적합하다.

- 장점 1. 다양한 데이터를 참조하는 질의에 적합
- 장점 2. 데이터 확장이 용이

- 그래프데이터 베이스 특징 : https://d2.naver.com/helloworld/8446520
- Neo4JS, Cyper : https://wikidocs.net/50728

유사 프로젝트

빅데이터 분석 프로젝트 예시

https://www.koreascience.or.kr/article/CFKO201835372170725.pdf

데이터 관리 예시 : 수집한 데이터는 제목, 기사가 작성된 시간, 기사 내용, 언론사, URL, 카테고리, 댓글 수, 추천 수로 분류하여 NoSQL에 저장한다. 이 때, 동일한 정보를 가진 데이터가 저장되는 것을 방지하기 위하여 제목과 URL로 중복 여부를 검사한다.

분모자 기능

- 기사 검색 -> 추천 알고리즘
- 이슈 랭킹 (시간 기준)
- 시각화 (유튜브 또는 사진)
- 필터 기능 (기간, 언론사, 카테고리)

- 기사내용, 제목으로 키워드 추출-> 추천 알고리즘, 검색,
- 키워드 입력 -> 구독

제목, 시간, 기사 내용, 언론사, URL, 카테고리, 추천 수, 댓글 수로 분류하여 NoSQL에 저장한다.