01.16

📌인덱스

클러스터링 인덱스

- PK 를 기준으로 정렬된다. ⇒ 기본 키(primary key) = 클러스터드 인덱스
  - PK 값에 의해 저장 위치가 결정된다.
- leaf 페이지 = 데이터 페이지
  - **Leaf페이지의 주소**로 구성하고, Leaf페이지는 **실제 데이터 페이지**로 구성된다.
  - 따로 인덱스 페이지를 만들지 않는다.
- 데이터 입력, 수정, 삭제 시 항상 정렬 상태를 유지
  - 인덱스를 생성/수정/삭제할 때 페이지 분할되어 데이터 페이지 전체를 다시 정렬해야 하기 때문에 느려진다.
- 검색속도가 높다.
- 사용경우
  - 테이블 데이터가 자주 업데이트 되는 경우
  - 읽기 작업이 월등히 많은 경우
  - 정렬된 상태로 데이터를 반환해야하는 경우
  - max,min, count 쿼리로 범위가 정해져있거나 groupby로 조회할 경우

넌클러스터링 인덱스

- 인덱스 페이지 생성해야 한다. → 인덱스 페이지만을 위한 **추가 저장공간이 필요하다.**
  - Non-Clustered Index의 인덱스 페이지(리프 페이지)는 키값과 데이터가 위치하는 포인터(RID)로 구성
- 레코드의 원본은 정렬되지 않고, 인덱스 페이지만 정렬된다.
- 검색속도는 느리나 , 입력, 삭제, 수정이 빠르다.

    - 클러스터드 인덱스보다 페이지 분할이 적게 일어난다.
    - 리프페이지가 모두 차 있어도 페이지 분할은 일어나지 않는다.
- 사용경우
    - where절이나 Join 절과 같이 조건문을 활용하여 테이블을 필터링 하고자 할 경우
    - 데이터가 자주 업데이트 될 경우
    - 특정 컬럼이 쿼리에서 자주 사용 될 경우

📌네트워크 통신 순서 (SSL 핸드셰이크 과정)

1) Client Hello - 클라이언트

Cyper suit(암호화 알고리즘 목록)와 브라우저의 ssl/tls 버전정보와 클라이언트에서 생성한 난수를 서버에 준다

2) Server Hello - 서버

cyper suit 선택을 하고 SSL 인증서와 서버 공개키와 서버에서 생성한 난수를 클라이언트에게 준다.

3) SSL 인증서 확인 - 클라이언트

(1) 현재 클라이언트는 난수, CA 목록 리스트, 서버 인증서, 서버공개키, SSL 인증서를 가지고 있다.

(2) 클라이언트에서 서버가 준 서버 인증서가 클라이언트가 내장한 CA 목록에 있는지 확인한다.

(3) SSL 인증서를 서버공개키를 이용하여 복호화한다. SSL 인증서 안에 전자서명에서 CA 공개키를 얻을 수 있게 된다.

4) 브라우저가 생성한 난수와 서버의 난수를 이용하여 Premaster secret key를 생성한다. (공유키) - 클라이언트

5) 서버 공개키로 premaster secret key를 암호화하여 서버로 전송한다.. - 클라이언트

6) 비공개키를 이용하여 premaster secret key를 복호화 한다. - 서버

7) 복호화 한 값을 master secret 값으로 저장한다. - 서버

8) master secret 을 이용해 session key(공유키)를 생성한다.

9) session key를 이용하여 브라우저와 서버사이에 주고 받는 데이터를 암호화하고 복호화 한다.