

학습목표

1. INDEX

학습내용

- 인덱스(INDEX)에 대해 알 수 있습니다.
- 인덱스를 정의하는 방법을 알 수 있습니다.

사전퀴즈

1. 인덱스(index)는 데이터를 빠르게 업데이트하기 위해서 사용한다.

정답 : X

해설 : 인덱스(index)는 데이터를 빠르게 찾기 위해 사용하는 보조 자료구조이다.

2. 인덱스는 업데이트가 잦을 경우에도 유용하다.

정답 : X

해설 : 인덱스는 업데이트가 잦을 경우에는 오히려 없는 것보다 성능이 낮을 경우도 있다.

수업

1. 인덱스

인덱스의 개념

- 검색을 빠르게 하기 위한 자료구조(주로 B트리 계열)

· 레코드 추가/수정/삭제 시 **해당 컬럼으로 다시 정렬한 후** 검색에 필요한 값들을 미리 계산해 검색 속도를 높임 (trade off)

→ 레코드가 추가/수정/삭제될 때마다 인덱스가 변경되므로, 레코드가 자주 추가/수정/삭제 되는 경우 인덱스의 효과가 떨어짐

왜냐하면 인덱스는 데이터가 큰 변화없이 저장되어 있고 많은 검색을 하는경우에만 의미를 가지기 때문이다.

· 검색보다 추가/삭제가 빨라야 한다면?

→ NoSQL을 사용하거나, RDBMS를 사용하는 경우 인덱스를 사용하지 않는 방법이 있다.

- 기본키는 자동으로 인덱스 설정(Unique index)

- 인덱스 ON/OFF를 통해 검색속도 차이 체감 (대량의 데이터일 수록 큰 영향)

- 조인(JOIN)시에도 영향을 줌

인덱스 추가/삭제

- CREATE INDEX 인덱스명 ON 테이블명(컬럼명)

- CREATE INDEX 인덱스명 ON 테이블명(컬럼명1, 컬럼명2 ...)

→ city 테이블의 name, population 컬럼을 동시에 조회하는 경우(where name='값' and population='값')

city 테이블의 name 컬럼에 대해서만 생성된 인덱스를 사용하지 못함

즉, 2개의 컬럼(name, population 컬럼)에 대한 인덱스를 새로 생성해주어야 함

- CREATE UNIQUE INDEX 인덱스명 ON 테이블명(컬럼명)

→ 중복값을 허용하지 않는 인덱스를 생성하고 싶은 경우 사용(인덱스는 기본적으로 중복값을 허용)

기본키와 unique 제약조건시 생성되는 인덱스

- ALTER TABLE 테이블명 DROP INDEX 인덱스명

예제 1-1

BusinessCard의 전화번호(Telephone)컬럼에 인덱스를 추가하시오.

→ BusinessCard 테이블의 전화번호(Telephone) 컬럼으로 조회를 많이 하는데, 해당 컬럼에 인덱스가 생성되어 있지 않아서 검색 성능이 떨어진다고 판단하여 인덱스를 생성하기로 함.

"성능이 떨어진다"는 판단은 보통 2가지로 할 수 있다.

1) 쿼리의 실행속도(보통 쿼리시간이 0.5 ~1초 이상 걸리는 경우)

2) 쿼리가 어떻게 수행되는지 자세히 알 수 있는 explain

예를 들어, explain을 보고 전화번호(Telephone) 컬럼에 대한 인덱스를 사용하지 않고 검색을 하여 성능이 떨어진다는 사실을 알 수 있다.

```

1 use sampleDB;
2
3 create table BusinessCard(
4     Name varchar(255),
5     Address varchar(255),
6     Telephone varchar(255)
7 )
8 ;
9
10 insert into BusinessCard
11 values('Bob', 'Seocho-dong 123', '123-456')
12 ;
13
14 insert into BusinessCard
15 values('Sam', 'Sinsa-dong 456', '321-456')
16 ;
17
18 create index BusinessCard_Telephone_index
19 on BusinessCard(Telephone)
20 ;

```

인덱스 실습

- 성능차이를 확인하려면 INDEX 관련 A/B(on/off) 테스트
- 인덱스 설정하지 않고 검색 / 설정하고 검색
- 쿼리 수행시간 비교
 - 실제 성능 차이가 확연히 나려면 수 십만 개 이상의 레코드 필요
 - 데이터가 수 백 ~ 수 천 개의 경우 차이가 미미함
 - 데이터가 100만 개가 넘어가면 인덱스 유/무가 커다란 차이
- 조인연산도 테이블 합치는 연산 중 검색이 많이 들어감
- 샘플 데이터베이스
 - <https://launchpad.net/test-db>
 - 직원/부서 테이블
 - 테이블당 약 30만 개의 샘플 DB
 - 인덱스(B+ 트리)의 효과검증

인덱스를 효율적으로 사용하는 방법

전문가의견

기본키와 외래키는 인덱스가 기본적으로 생성되며, 그 외 컬럼으로 검색이 자주 일어나는 경우에는 해당 컬럼에 인덱스를 달아주는 것이 효율적입니다.

인덱스는 조인연산에도 쓰입니다. 조인연산은 테이블 여러 개를 외래키를 기반으로 합치다 보니 한 테이블의 레코드 별로 대응하는 테이블의 해당 레코드를 찾아서 연결해야 합니다. 이 때 대응하는

테이블의 컬럼이 인덱스가 달려있다면 빠르게 조인이 실행되지만, 만약 조인연산의 방향이 반대인 경우에는 조인시에 많은 시간이 걸릴 수도 있습니다. 만약 실행시간이 많이 걸린다면 인덱스를 추가해주는 것이 좋습니다.