



VI. 인덱스

이 장에서 다룰 내용



1 조회를 빠르게 하는 인덱스

2 인덱스 생성

01. 조회를 빠르게 하는 인덱스

❖ 인덱스를 왜 사용하는 것일까요?



- 이에 대한 답은 “빠른 검색을 위해서 인덱스를 사용합니다.” 입니다.
 - 여러분이 테이블 생성 방법을 책에서 찾으려고 할 때 어떻게 합니까? 책 첫 페이지부터 한 장씩 넘겨가면서 테이블 생성 방법이 기술되어 있는지 일일이 살펴보는 사람은 드물 것입니다.
 - 일반적으로 책 맨 뒤에 있는 색인(인덱스, 찾아보기)에서 해당 단어(테이블)를 찾아 그 페이지로 이동합니다.
 - 이렇게 원하는 단어를 쉽게 찾는 방법으로 색인, 인덱스가 사용되는 것처럼 오라클의 인덱스 역시 원하는 데이터를 빨리 찾기 위해서 사용됩니다.
-
- ❖ 인덱스란 SQL 명령문의 처리 속도를 향상시키기 위해서 컬럼에 대해서 생성하는 오라클 객체입니다.

01. 조회를 빠르게 하는 인덱스

❖ 하지만 인덱스는 장점만 있는 것이 아닙니다.

- 오라클에서의 인덱스의 내부 구조는 B* 트리 형식으로 구성되어 있습니다.
- 트리란 나무의 뿌리 모양을 생각해 보시면 쉽게 이해할 수 있습니다.
- 뿌리(루트)를 근거로 아래로 나무뿌리 들이 뻗어 있는 모양을 하고 있습니다.
- 컬럼에 인덱스를 설정하면 이를 위한 B* 트리도 생성되어야 하기 때문에 인덱스를 생성하기 위한 시간도 필요하고 인덱스를 위한 추가적인 공간이 필요하게 됩니다.
- 인덱스가 생성된 후에 새로운 행을 추가하거나 삭제할 경우 인덱스로 사용된 컬럼 값도 함께 변경되는 경우가 발생합니다.
- 인덱스로 사용된 컬럼 값이 변경되는 이를 위한 내부 구조(B* 트리) 역시 함께 수정 돼야 합니다.
- 이 작업은 **오라클 서버에 의해 자동으로 일어나는데 그렇기 때문에 인덱스가 없는 경우 보다 인덱스가 있는 경우에 DML 작업이 훨씬 무거워지게 됩니다.**



01. 인덱스의 사용 용도와 장단점



❖ 인덱스의 장점

- 검색 속도가 빨라진다.
- 시스템에 걸리는 부하를 줄여서 시스템 전체 성능을 향상시킨다.

❖ 인덱스의 단점

- 인덱스를 위한 추가적인 공간이 필요하다.
- 인덱스를 생성하는데 시간이 걸린다.
- 데이터의 변경 작업(INSERT/UPDATE/DELETE)이 자주 일어날 경우에는 오히려 성능이 저하된다.

<탄탄히 다지기>

1. _____는 조회의 성능을 향상시키는 객체이다.

1.2 조회 속도 비교하기



- ❖ 인덱스가 조회 속도를 빠르게 해 준다는 것을 증명하기 위해서 기본 키나 유일 키로 지정하지 않는 컬럼인 사원 이름으로 검색해 봅시다.
- ❖ 아마도 시간이 어느 정도 소요될 것입니다.
- ❖ 검색을 위해서 WHERE 절에 사용되는 컬럼인 사원 이름 컬럼을 인덱스로 생성한 후에 다시 한번 사원 이름으로 검색해보면 검색시간이 현저하게 줄어드는 것을 확인할 수 있습니다.

02. 인덱스 생성하기



- ❖ 제약 조건에 의해 자동으로 생성되는 인덱스 외에 CREATE INDEX 명령어로 직접 인덱스를 생성할 수도 있습니다.
- ❖ 다음은 인덱스를 생성하기 위한 기본 형식입니다.

```
CREATE INDEX index_name
ON table_name (column_name);
```

- CREATE INDEX 다음에 인덱스 객체 이름을 지정합니다. 어떤 테이블의 어떤 컬럼에 인덱스를 설정할 것인지를 결정하기위해서 ON 절 다음에 테이블 이름과 컬럼 이름을 기술합니다.

03. 인덱스 제거하기



```
DROP TABLE EMP01;  
  
CREATE TABLE EMP01 AS SELECT * FROM EMP;  
  
INSERT INTO EMP01 SELECT * FROM EMP01; -- 2번+11번 수행  
  
SELECT COUNT(*) FROM EMP01; -- 11만개  
  
INSERT INTO EMP01 (EMPNO, ENAME) VALUES (1111,'HONG'); --HONG 일부  
  
INSERT INTO EMP01 SELECT * FROM EMP01; -- 3번 수행(90만 ROW정도)  
  
SELECT TO_CHAR(COUNT(*), '9,999,999') FROM EMP01;
```

```
SELECT * FROM EMP01 WHERE ENAME='HONG'; -- 검색속도 느림  
  
CREATE INDEX IDX_EMP_ENAME ON EMP01(ENAME);  
  
SELECT * FROM EMP01 WHERE ENAME= 'HONG' ; -- 검색속도 빠름  
  
INSERT INTO EMP01 SELECT * FROM EMP01; -- insert 속도 엄청 느림  
  
DROP INDEX IDX_EMP_ENAME;  
  
INSERT INTO EMP01 SELECT * FROM EMP01; -- insert 속도 빨라짐
```

<실습하기> 인덱스 설정하기

인덱스가 지정하지 않은 컬럼인 ENAME으로 조회하여 어느 정도의 시간은 소요됨을 확인하였으므로 이번에는 ENAME 컬럼으로 인덱스를 지정하여 조회 시간이 단축됨을 확인해봅시다.

1. 이번에는 테이블 EMP01의 컬럼 중에서 이름(ENAME)에 대해서 인덱스를 생성해봅시다.

```
CREATE INDEX IDX_EMP01_ENAME  
ON EMP01(ENAME);
```

2. 사원 이름이 'HONG'인 행을 검색해 봅시다.

```
SELECT DISTINCT EMPNO, ENAME  
FROM EMP01  
WHERE ENAME='HONG';
```

3. 인덱스를 생성한 후에 사원 이름이 'HONG'인 행을 다시 검색하면 수행속도가 매우 감소함을 알 수 있습니다.

02. 인덱스를 사용해야 하는 경우 판단하기

- ❖ 인덱스가 검색을 위한 처리 속도만 향상시킨다고 했습니다.
- ❖ 하지만, 무조건 인덱스를 사용한다고 검색 속도가 향상되는 것은 아닙니다.
- ❖ 계획성 없이 너무 많은 인덱스를 지정하면 오히려 성능을 저하시킬 수도 있습니다.
- ❖ 언제 인덱스를 사용하는 것이 좋을까요?



02. 인덱스를 사용해야 하는 경우 판단하기



인덱스를 사용해야 하는 경우	인덱스를 사용하지 말아야 하는 경우
테이블에 행의 수가 많을 때	테이블에 행의 수가 적을 때
WHERE 문에 해당 컬럼이 많이 사용될 때	WHERE 문에 해당 컬럼이 자주 사용되지 않을 때
검색 결과가 전체 데이터의 2%~4% 정도 일 때	검색 결과가 전체 데이터의 10%~15% 이상 일 때
JOIN에 자주 사용되는 컬럼이나 NULL을 포함하는 컬럼이 많은 경우	테이블에 DML 작업이 많은 경우 즉, 입력 수정 삭제 등이 자주 일어 날 때

02. 인덱스를 사용해야 하는 경우 판단하기

- ❖ 다음과 같은 조건에서 사원 테이블의 부서 번호에 인덱스를 거는 것이 좋을까요?
 - 테이블에 전체 행의 수는 100000 건이다.
 - 위의 쿼리문을 전체 쿼리문 들 중에서 95% 사용된다.
 - 쿼리문의 결과로 구해지는 행은 100건 정도이다.
- ❖ 조건을 위 표를 비추어보고 판단해 보면 DEPTNO 컬럼을 인덱스로 사용하기에 알맞다는 결론이 납니다.

