

Index

Dongseop Kwon

DBMS 저장 구조

- Heap File
 - Table 저장
 - Disk Block 단위로 접근
 - 레코드의 순서/위치는 언제라도 바뀔 수 있음 (예: 데이터의 변경)



- Index
 - 테이블 검색을 빠르게 하기 위하여 미리 구축해 두는 저장 구조
 - 종류: Tree, Hash, Bitmap ...

Index

- 장점
 - 대량의 데이터에서 특별한 값을 빠르게 검색
 - 검색, 조인, 정렬...
- 단점
 - 유지 비용 (데이터 변경 시 인덱스도 변경하여야 함)
 - 저장 공간
- 언제?
 - WHERE 절의 조건이나 Join에 자주 사용되는 컬럼
 - 매우 큰 테이블에서 2~4%레코드만 선택될 때
 - 값의 종류가 다양할 때 (예. 이름 vs. 성별?)
 - 자주 변경되지 않을 때

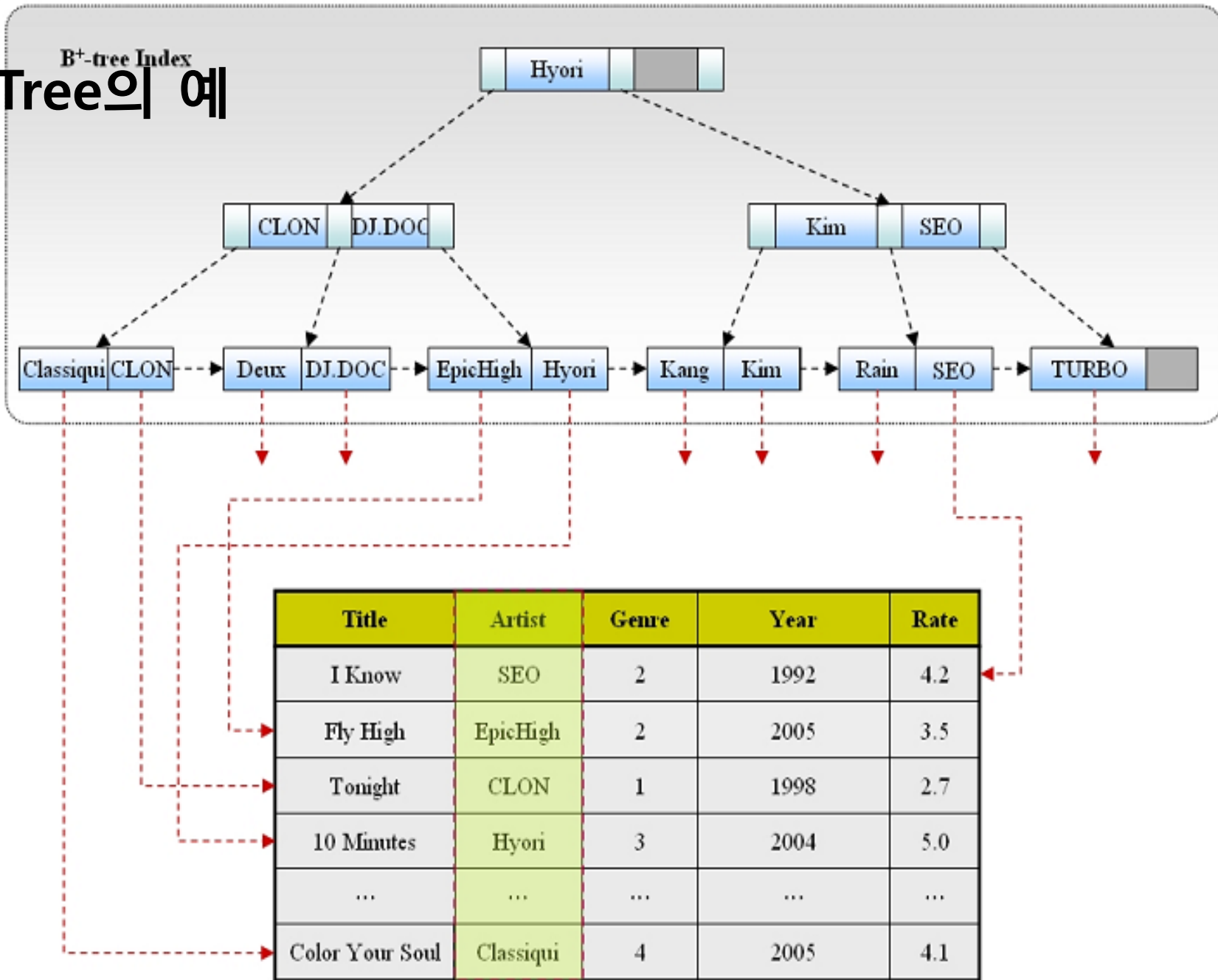
Oracle INDEX

- 종류
 - B⁺-Tree: 일반적 인덱스 (트리 기반)
 - Bitmap 인덱스: 특수 용도 (OLAP 등에 사용)
- 구분
 - Single 컬럼 vs. Composite 컬럼
 - Unique vs. Non-Unique
 - Column Data vs. Function-Based
 - Automatic-Created vs. User-Created

B⁺-Tree

- 특징
 - 대표적인 트리 기반의 인덱스 구조
 - m-원 탐색 트리
 - Block 기반의 접근 구조 (디스크 저장 구조로 적합)
 - 데이터의 삽입/삭제에도 자동적으로 트리 구조 유지
 - 트리 높이가 항상 일정 (skew가 없음)
 - 최소 저장공간 활용 비율 보장 가능 (예: 50%는 최소 사용)
 - Leaf 노드 데이터는 키에 의하여 정렬됨

B⁺-Tree의 예



MUSIC Table

Syntax

- CREATE [UNIQUE] INDEX *index_name* ON *table_name* (*column_list...*);

```
CREATE INDEX idx_emp_ename ON emp(ename);
```

- DROP INDEX *index_name*;

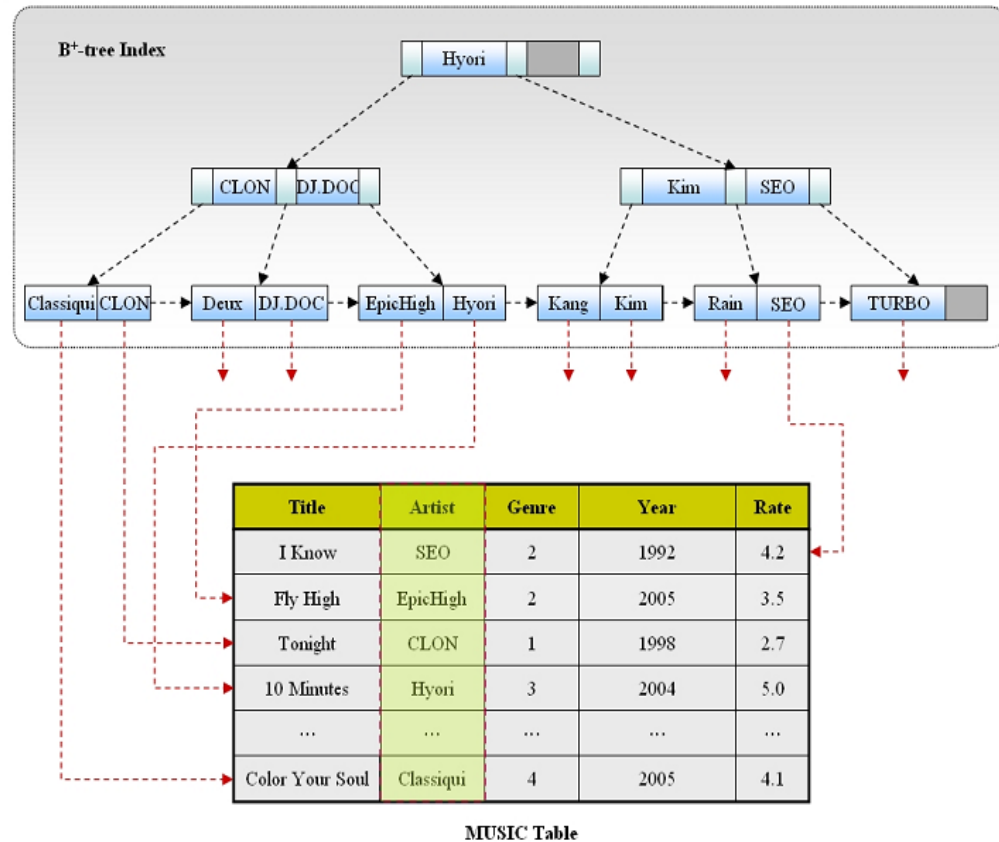
```
DROP INDEX idx_emp_ename;
```

- Dictionary
 - USER_INDEXES

```
SELECT t.index_name, t.uniqueness,  
       c.column_name, c.column_position  
FROM user_indexes t, user_ind_columns c  
WHERE c.index_name = t.index_name AND t.table_name =  
      'EMP';
```

Oracle의 데이터 접근 방법

- Full Table Scan:
 - 전체 테이블 순차적 접근
- Index Scan:
 - 인덱스를 통한 접근
 - Index Unique:
 - Index Range:
 - Index Full Scan:



Query Optimizer

- Query Optimizer가 주어진 질의의 처리방법 결정
 - Cost-Based Optimization
 - Heuristic-Based Optimization
- 일반적인 고려 요소
 - 인덱스의 유무
 - 테이블의 크기
 - 데이터의 분포
 - ...

Composite Index & Covering Index

- **Composite Index**

- 둘 이상의 컬럼의 쌍에 대한 인덱스. **순서 중요!**

- 예)

```
CREATE INDEX idx_emp1 ON emp(dept, name) ;
```

- SELECT * FROM emp WHERE **dept** = 'A' AND **name** = 'B'
 - SELECT * FROM emp WHERE **dept** = 'A'
 - SELECT * FROM emp WHERE **name** = 'B'

- **Covering Index** (covered query)

- Table 참조 없이 Index만으로 처리 가능한 경우

- 일반적으로 성능이 우수

- 예)

- SELECT dept, name FROM emp WHERE dept='AAA';
 - SELECT COUNT(*) FROM emp WHERE dept='AAA';

Hint Index

- 강제로 질의에서 특정 인덱스를 사용하도록 강요함
- Oracle 예

```
SELECT /*+ index (employees EMP_EMAIL_UK) */ email  
FROM employees;
```