

튜닝 기초

- *Database Basic* -

2014.12

이 익 훈

E-mail: ihlee90@nhn.com

ihlee90@gmail.com

학습 목표

- I. DB 튜닝의 기초 개념을 이해하고 설명할 수 있다.
- II. 질의 실행계획(query plan) 보기, 인덱스를 이용한 성능 개선을 할 수 있다.

DB 튜닝

◆ 데이터베이스 튜닝:

데이터베이스 시스템의 성능을 최적화하기 위해 하는 작업(DB 애플리케이션, 데이터베이스 자체, 운영체제 등의 개선).

◆ DB 튜닝에 필요한 지식

응용 프로그램 개발 지식부터 운영체제, 하드웨어 시스템까지 다양한 분야의 지식이 필요합니다.

◆ 데이터베이스 시스템의 성능 개선은 애플리케이션 영역과 시스템 영역이 있습니다.

시스템 영역은 주로 Disk, OS, DBMS 등을 포함합니다.

애플리케이션 영역은 애플리케이션 설계, 로직(알고리즘), 데이터 스키마 설계, SQL 등을 포함합니다.

Query Plan 보기

- ◆ DB 성능 문제 발생시, 다양한 각도에서 조사해야 하지만, 대부분의 경우 쿼리 실행시간이 느린 경우에 대한 확인 및 대응이 많음
 - ◆ 쿼리 플랜보기
 1. Explain 사용하기
 2. Workbench, Toad 등의 DB 도구 사용하기
- ※ DBMS, 도구에 따라 약간씩 다름

EXPLAIN SELECT * FROM book;

쿼리 내에서의
Select
순차번호

사용가능한
인덱스들

인덱스와
비교되는 열

Result Set Filter: Export: Wrap Cell Content										
	id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
▶	1	SIMPLE	book	ALL	NULL	NULL	NULL	NULL	2	

Select 종류:
SIMPLE,
SUBQUERY,
UNION 등

Join 종류:
ALL, INDEX,
SUBQUERY
INDEX 등

실제 사용하는
인덱스

쿼리 실행을
위해 조사해야
하는
투플수(추정)

[실습] 책 제목이 '이기적유전자'인 책에 대한 정보를 검색하시오.

단, 쿼리가 실행되는 질의실행 계획을 확인하고 문제점이나 개선이 필요한 부분을 찾으시오.

인덱스 사용

◆ 인덱스만 잘 사용해도 많은 성능 문제 해결이 가능

◆ 인덱스 생성

```
CREATE INDEX idx_name ON tbl (col1,  
col2, ...)
```

* 인덱스 삭제

```
ALTER TABLE tbl DROP INDEX idx_name;
```


[실습] book 테이블의 title 열에 인덱스 생성 후 인덱스 정보를 확인하시오.

(hints) 인덱스 정보 보기

SHOW INDEX FROM *tbl_name*;

[실습] 100만건 레코드에 인덱스 사용 예

- (1) Book **data 100만건** 생성하는 프로그램 작성, **대량 삽입** 실행
(id는 primary key, title은 순차생성 title1, title2, ...,
author는 난수로 생성 auth3, ... ,auth123
나머지는 default값을 NULL로)
- (2) `SELECT * FROM book WHERE title = 'title123456';`
실행하고 시간 측정, 쿼리 플랜 확인
- (3) **인덱스 생성**
title 열에 인덱스 생성
- (4) `SELECT * FROM book WHERE title = 'title123456';`
실행하고 시간 측정, 쿼리 플랜 확인

* 100만건 book 데이터 생성 프로그램 예
이후에, 생성된 book_data.csv 파일을 대량 삽입

```
#include "stdafx.h"
```

```
#include <iostream>
```

```
#include <fstream>
```

```
#include <string>
```

```
#include <time.h>
```

```
#define NUMOFRECORDS 1000000
```

```
using namespace std;
```

```
int _tmain(int argc, _TCHAR* argv[])
```

```
{
```

```
    ofstream outstream;
```

```
    string output;
```

```
    char sint[20];
```

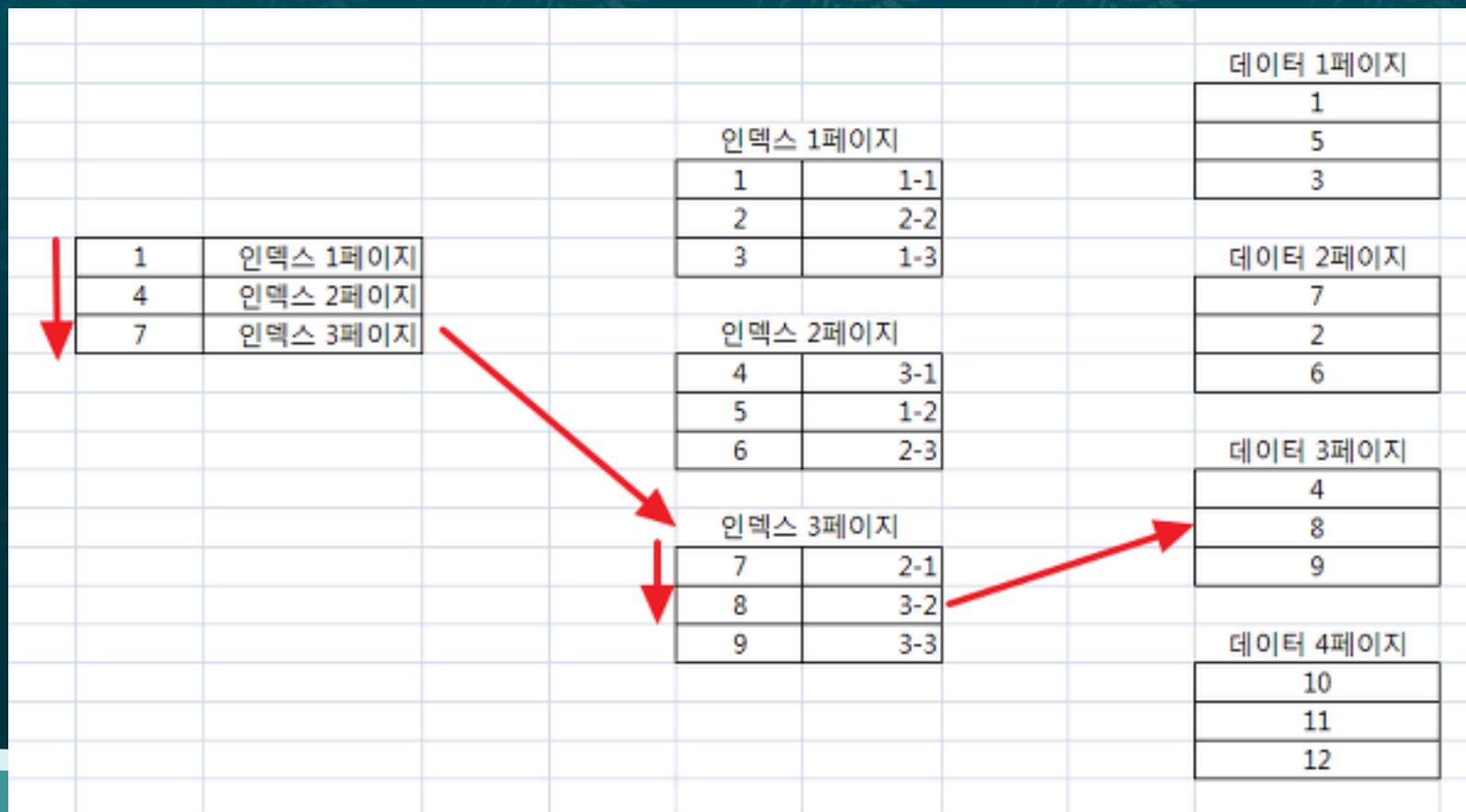
```
ostream.open("C:\\Test\\book_data.csv");
if (ostream.fail()) {
    cout << " error \n";
    return 1;
}
srand((unsigned) time(NULL));

for (int i = 0; i < NUMOFRECORDS; i++) {
    output = "";
    _itoa_s(i, sint, 10); output.append( sint ); // id
    output.append(",");
    output.append("title"); output.append( sint ); // title
    output.append( ",");
    output.append("author");
    _itoa_s((rand() % 1000), sint, 10); output.append( sint ); // author

    ostream << output << endl;
}

ostream.close();
return 0;
}
```


Clustered vs. non-clustered index



						1페이지	
						1	
						2	
						3	
						2페이지	
						4	
						5	
						6	
						3페이지	
						7	
						8	
						9	
						4페이지	
						10	
						11	
						12	

	클러스터드 인덱스	non 클러스터드 인덱스
차이	물리적으로 행을 재배열	물리적으로 재배열 하지 않는다.
크기	인덱스 페이지 용량이 non 클러스터드 인덱스 페이지 용량보다 작다.	클러스터드 인덱스 페이지 용량보다 크다.
선택도	30% 이내에서 사용해야 좋은 선택도	3% 이내에서 사용해야 좋은 선택도
최대 갯수	테이블당 1개	테이블당 여러 개

◆복합 인덱스

여러 칼럼에 대해 인덱스 생성도 가능합니다.

(사용 예)

```
CREATE INDEX idx_titleAuthor  
ON book( title, author);
```


[실습] (복합 인덱스 테스트) book 테이블에서
다음 조건으로 검색하고 실행시간과 쿼리 플
랜을 확인해 보시오.

- 제목이 'title123456'인 책
- 저자가 'author123'인 책
- 제목이 'title123456'이고 저자가 'author333'인 책

※ 복합 인덱스 생성은 심사 숙고해서



**NHN INSTITUTE FOR THE
NEXT NETWORK**