



데이터베이스 Database

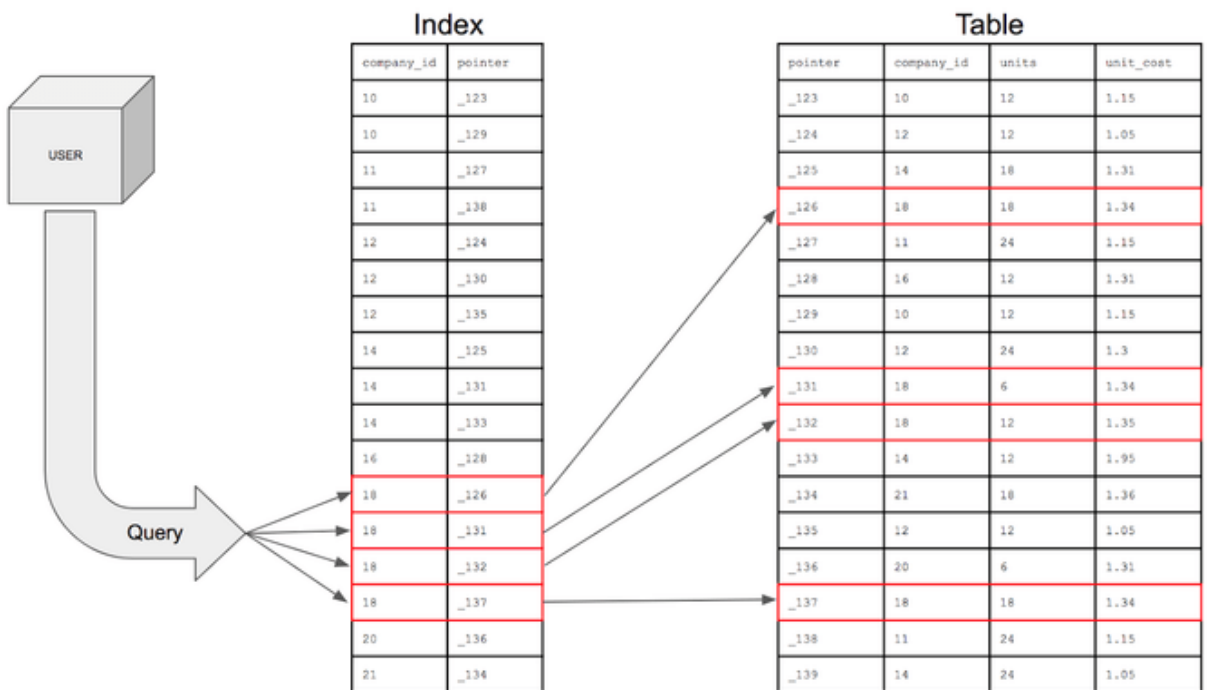
15 인덱스(Index)

INDEX
UNIQUE INDEX

인덱스(Index)

• 인덱스(Index)

- 데이터베이스에서 테이블 검색 속도를 향상시키기 위한 자료구조
- 별도의 추가 저장 공간을 활용하여 인덱스 생성
- 데이터와 데이터 위치를 포함한 자료구조 생성
- 데이터를 빠르게 찾을 수 있기 때문에 디스크 액세스 횟수 감소
- DBMS에서 인덱스를 자동으로 사용하며 유지 보수 수행
- 인덱스는 언제든지 생성 및 삭제가 가능하며 다른 테이블이나 인덱스에 영향을 주지 않음
- 인덱스를 통해 데이터 조회를 위한 SELECT 뿐만 아니라 UPDATE와 DELETE의 성능도 향상

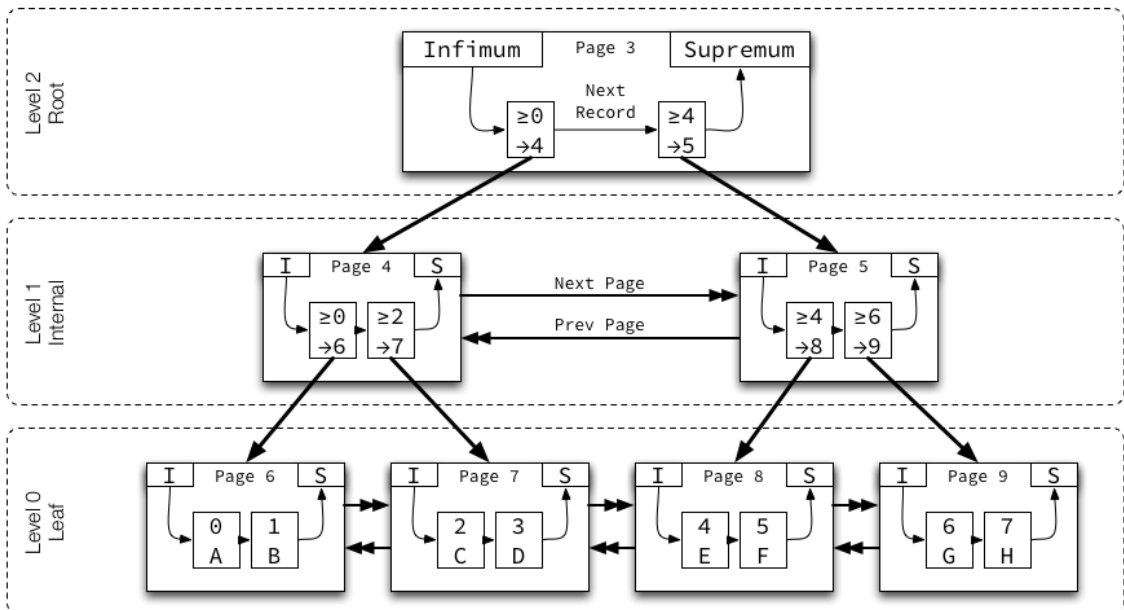


• 인덱스 관리

- 데이터에 변경이 발생하면 최신 상태 유지를 위해 인덱스 관리 필요
- 인덱스가 적용된 컬럼은 최신 데이터를 정렬된 상태로 유지
- INSERT, UPDATE, DELETE 발생 시 인덱스 관리 수행
- INSERT: 새로운 데이터에 대한 인덱스 추가
- UPDATE: 기존의 인덱스는 사용하지 않도록 처리하고, 갱신된 데이터에 대한 인덱스 추가
- DELETE: 삭제하는 데이터의 인덱스는 사용하지 않도록 처리

인덱스(Index)

- 인덱스 장점
 - 테이블 조회 속도 향상
 - 전반적인 시스템 부하 감소
- 인덱스 단점
 - 인덱스 관리를 위한 추가 작업 필요
 - 데이터베이스의 약 10% 정도의 추가 저장 공간 필요
 - 잘못된 인덱스 사용으로 성능이 저하될수도 있음
- 인덱스 사용에 적합한 경우
 - 규모가 큰 테이블
 - INSERT, UPDATE, DELETE가 자주 발생하지 않는 컬럼
 - JOIN이나 WHERE 또는 ORDER BY에 자주 사용되는 컬럼
 - 중복되는 데이터가 최소인 컬럼
- 인덱스 자료구조
 - 트리 형태의 자료구조를 주로 사용하며, 특히 B Tree 계열에서 B* Tree 또는 B+ Tree 구조를 가장 많이 사용



Levels are numbered starting from 0 at the leaf pages, incrementing up the tree.
Pages on each level are doubly-linked with previous and next pointers in ascending order by key.
Records within a page are singly-linked with a next pointer in ascending order by key.
Infimum represents a value lower than any key on the page, and is always the first record in the singly-linked list of records.
Supremum represents a value higher than any key on the page, and is always the last record in the singly-linked list of records.
Non-leaf pages contain the minimum key of the child page and the child page number, called a "node pointer".

SELECT INDEX

- 인덱스 조회

```
SELECT *  
FROM user_indexes;
```

```
SELECT *  
FROM user_indexes  
WHERE table_name = 'EMPLOYEES';
```

- 인덱스 컬럼 조회

```
SELECT *  
FROM user_ind_columns;
```

```
SELECT *  
FROM user_ind_columns  
WHERE table_name = 'EMPLOYEES';
```

- 인덱스 사용 여부

```
SELECT *  
FROM employees  
WHERE employee_id = 100;
```

```
SELECT *  
FROM employees  
WHERE first_name = 'Steven';
```


SELECT INDEX

- 예제 테이블 customer 생성

```
CREATE TABLE customers
(
  customer_id      number          NOT NULL    PRIMARY KEY,
  first_name       varchar2(10)    NOT NULL,
  last_name        varchar2(10)    NOT NULL,
  email            varchar2(10),
  phone_number     varchar2(20),
  regist_date      date
);
```

```
INSERT INTO customers
VALUES (1, 'Suan', 'Lee', 'suan', '010-1234-1234', '21/01/01');
```

```
INSERT INTO customers
VALUES (2, 'Elon', 'Musk', 'elon', '010-1111-2222', '21/05/01');
```

```
INSERT INTO customers
VALUES (3, 'Steve', 'Jobs', 'steve', '010-3333-4444', '21/10/01');
```

```
INSERT INTO customers
VALUES (4, 'Bill', 'Gates', 'bill', '010-5555-6666', '21/11/01');
```

```
INSERT INTO customers
VALUES (5, 'Mark', 'Zuckerberg', 'mark', '010-7777-8888',
'21/12/01');
```

- customer 테이블 인덱스 조회

```
SELECT *
FROM customers;
```

```
SELECT *
FROM user_indexes
WHERE table_name = 'CUSTOMERS';
```

CREATE INDEX

- 인덱스 생성

```
CREATE INDEX regist_date_idx  
ON customers (regist_date);
```

```
SELECT *  
FROM user_indexes  
WHERE table_name = 'CUSTOMERS';
```

```
SELECT *  
FROM customers  
WHERE regist_date = '21/01/01';
```

- 다중 속성 인덱스 생성

```
CREATE INDEX name_idx  
ON customers (first_name, last_name);
```

```
SELECT *  
FROM user_indexes  
WHERE table_name = 'CUSTOMERS';
```

```
SELECT *  
FROM customers  
WHERE first_name = 'Suan';
```

CREATE UNIQUE INDEX

- 고유 인덱스 생성

```
CREATE UNIQUE INDEX email_idx  
ON customers (email);
```

```
SELECT *  
FROM user_indexes  
WHERE table_name = 'CUSTOMERS';
```

```
SELECT *  
FROM customers  
WHERE email = 'suan';
```

```
CREATE UNIQUE INDEX phone_idx  
ON customers (phone_number);
```

```
SELECT *  
FROM user_indexes  
WHERE table_name = 'CUSTOMERS';
```

```
SELECT *  
FROM customers  
WHERE phone_number = '010-1234-1234';
```


DROP INDEX / TABLE

- 인덱스 삭제

```
SELECT *  
FROM user_indexes  
WHERE table_name = 'CUSTOMERS';
```

```
DROP INDEX regist_date_idx;
```

```
DROP INDEX email_idx;
```

```
DROP INDEX name_idx;
```

```
DROP INDEX phone_idx;
```

```
SELECT *  
FROM user_indexes  
WHERE table_name = 'CUSTOMERS';
```

- 테이블 삭제

```
DROP TABLE customers;
```

[실습] 테이블 생성 및 데이터 삽입

- products 테이블 생성

PRODUCT_ID	PRODUCT_NAME	REG_DATE	WEIGHT	PRICE
Number	Varchar2(10)	Date	Number	Number
NOT NULL	NOT NULL			
PRIMARY KEY				

- products 데이터 삽입

PRODUCT_ID	PRODUCT_NAME	REG_DATE	WEIGHT	PRICE
1	Computer	21/01/01	10	1600000
2	Smartphone	21/02/01	0.2	1000000
3	Television	21/03/01	20	2000000

[실습] 인덱스 생성

- products 테이블의 reg_date 컬럼에 대한 인덱스 reg_date_idx 생성

- products 테이블의 weight 컬럼에 대한 인덱스 weight_idx 생성

- products 테이블의 price 컬럼에 대한 인덱스 price_idx 생성

- products 테이블의 product_name 컬럼에 대한 고유 인덱스 product_name_idx 생성

- products 테이블에 대한 사용자 인덱스 조회

- products 테이블에 대한 사용자 인덱스 컬럼 조회

[실습] 인덱스 / 테이블 삭제

- products 테이블의 reg_date_idx 인덱스 삭제

- products 테이블의 weight_idx 인덱스 삭제

- products 테이블의 price_idx 인덱스 삭제

- products 테이블의 product_name_idx 인덱스 삭제

- products 테이블 삭제



이수안 컴퓨터 연구소

suan computer laboratory