

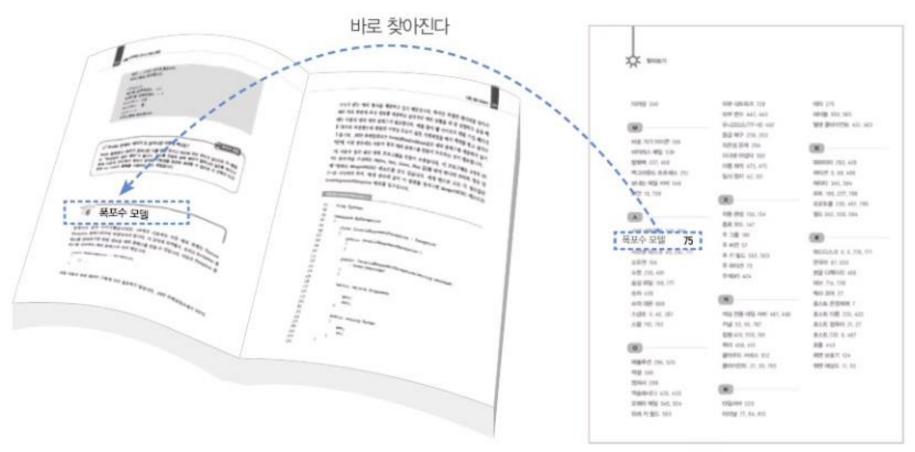
2024년 상반기 K-디지털 트레이닝

인덱스

[KB] IT's Your Life



◎ 인덱스의 개념



책의 제일 뒷부분

인덱스의 장점

- 검색 속도 향상
- 시스템 전체 성능 향상

인덱스의 단점

- 추가적인 공간 필요
- o 처음 인덱스 생성시 많은 시간 소요
- 데이터의 변경 작업(insert, update, delete)이 자주 일어날 경우 성능 저하

☑ 인덱스의 종류

- 클러스터형 인덱스
 - 행 데이터를 자신의 열을 기준으로 정렬함
 - 테이블 당 하나만 생성 가능
- 보조 인덱스
 - 별도의 공간을 추가하여 지정한 컬럼만 정렬하여 인덱스를 운영
 - 컬럼 단위로 여러 개의 인덱스 생성 가능

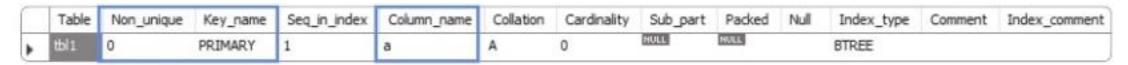
♡ 자동으로 생성되는 인덱스

- o Primary key 제약 조건
 - 클러스터형 인덱스 자동 생성
- o Unique 제약 조건
 - 보조 인덱스 자동 생성

- o Primary Key가 없고 Unique, Not Null 제약조건이 설정된 경우
 - 클러스터형 인덱스로 생성

🗸 인덱스 확인

SHOW INDEX FROM 테이블명



```
USE sqldb;

CREATE TABLE tbl1 (
   a INT PRIMARY KEY,
   b INT,
   c INT
);

SHOW INDEX FROM tbl1;
```

1	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
	tbl1	0	PRIMARY	1	a	A	0	HULL	NULL		BTREE		



```
CREATE TABLE tb12 (
   a INT PRIMARY KEY,
   b INT UNIQUE,
   c INT UNIQUE,
   d INT
);

SHOW INDEX FROM tb12;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
tbl2	0	PRIMARY	1	a	A	0	HULL	MULL		BTREE		
tbl2	0	b	1	b	A	0	NULL	NULL	YES	BTREE		
tbl2	0	c	1	c	A	0	HULL	HUGE	YES	BTREE		

```
CREATE TABLE tbl3 (
a INT UNIQUE,
b INT UNIQUE,
c INT UNIQUE,
d INT
);

SHOW INDEX FROM tbl3;
```

	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
٠	tbl3	0	a	1	a	A	0	MULL	MULL	YES	BTREE		
	tbl3	0	b	1	b	A	0	NULL	HULL	YES	BTREE		
	tbl3	0	c	1	c	A	0	EROUSE .	MULL	YES	BTREE		

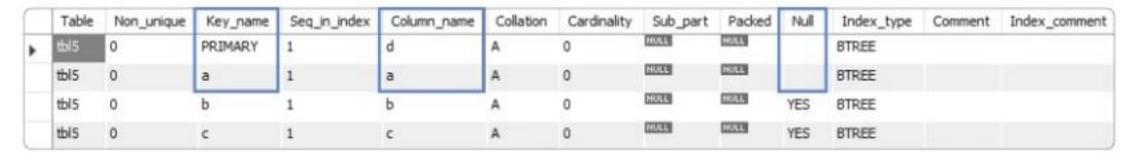
```
CREATE TABLE tbl4 (
a INT UNIQUE NOT NULL,
b INT UNIQUE,
c INT UNIQUE,
d INT
);

SHOW INDEX FROM tbl3;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
tbl4	0	a	1	a	A	0	HULL	NULL		BTREE		
tbi4	0	b	1	b	A	0	RULE	ROBE	YES	BTREE		
tbl4	0	c	1	c	A	0	HULL	HUSE	YES	BTREE		



```
CREATE TABLE tbl5 (
    a INT UNIQUE NOT NULL,
    b INT UNIQUE,
    c INT UNIQUE,
    d INT PRIMARY KEY
SHOW INDEX FROM tbl3;
```



2 인덱스의 종류와 자동 생성

```
CREATE DATABASE IF NOT EXITSTS testdb;

USE testdb;

DROP TABLE IF EXISTS usertbl;

CREATE TABLE usertbl(
    userID CHAR(8) NOT NULL PRIMARY KEY,
    name VARCHAR(10) NOT NULL,
    birthYear INT NOT NULL
    addr NCHAR(2) NOT NULL
);
```

```
INSERT INTO usertbl VALUES('LSG', '이승기', 1987, '서울');
INSERT INTO usertbl VALUES('KBS', '김범수', 1979, '경남');
INSERT INTO usertbl VALUES('KKH', '김경호', 1971, '전남');
INSERT INTO usertbl VALUES('JYP', '조용필', 1950, '경기');
INSERT INTO usertbl VALUES('S나', '성시경', 1979, '서울');
SELECT * FROM usertbl;
```

	userID	name	birthYear	addr
۰	JYP	조용필	1950	경기
	KBS	김범수	1979	경남
	ккн	김경호	1971	전남
	LSG	이승기	1987	서울
	SSK	성시경	1979	서울

```
ALTER TABLE usertbl DROP PRIMARY KEY;

ALTER TABLE usertbl

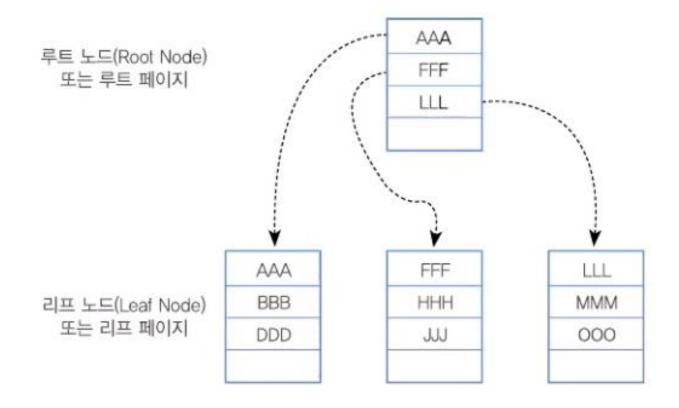
ADD CONSTRAINT pk_name PRIMARY KEY(name);

SELECT * FROM usertbl;
```

	userID	name	birthYear	addr
۰	кжн	김경호	1971	전남
	KBS	김범수	1979	경남
	SSK	성시경	1979	서울
	LSG	이승기	1987	서울
	JYP	조용필	1950	경기

B-Tree(Balanced Tree, 균형 트리)

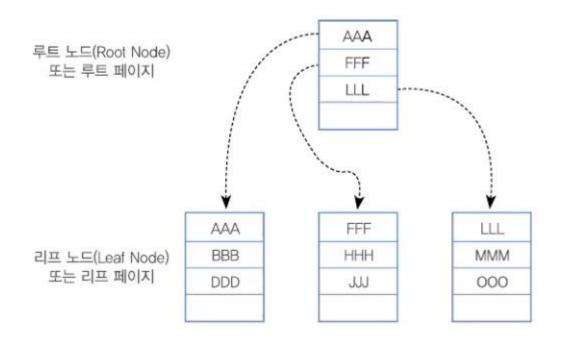
- o 페이지 Page
 - B-Tree의 노드
 - 16Kbyte의 고정 크기를 가짐
 - 데이터를 한 개만 저정하더라도 페이지 단위로 할당됨

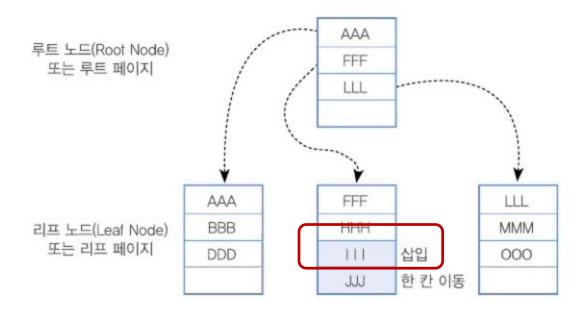


3 인덱스의 내부 작동

페이지 분할

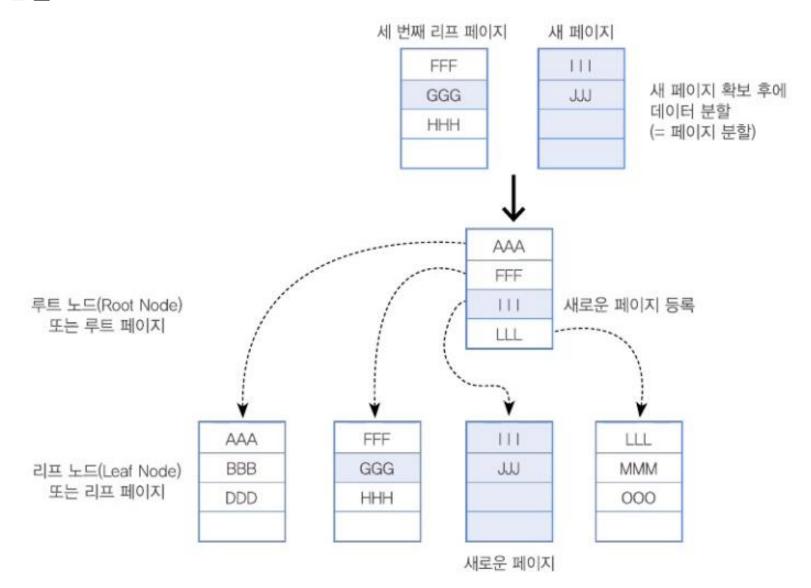
- 페이지에 공간이 없는 경우
 - 새로운 페이지를 생성
 - 기존 데이터를 양분하여 각각의 페이지에 재 배치
 - 링크 수정





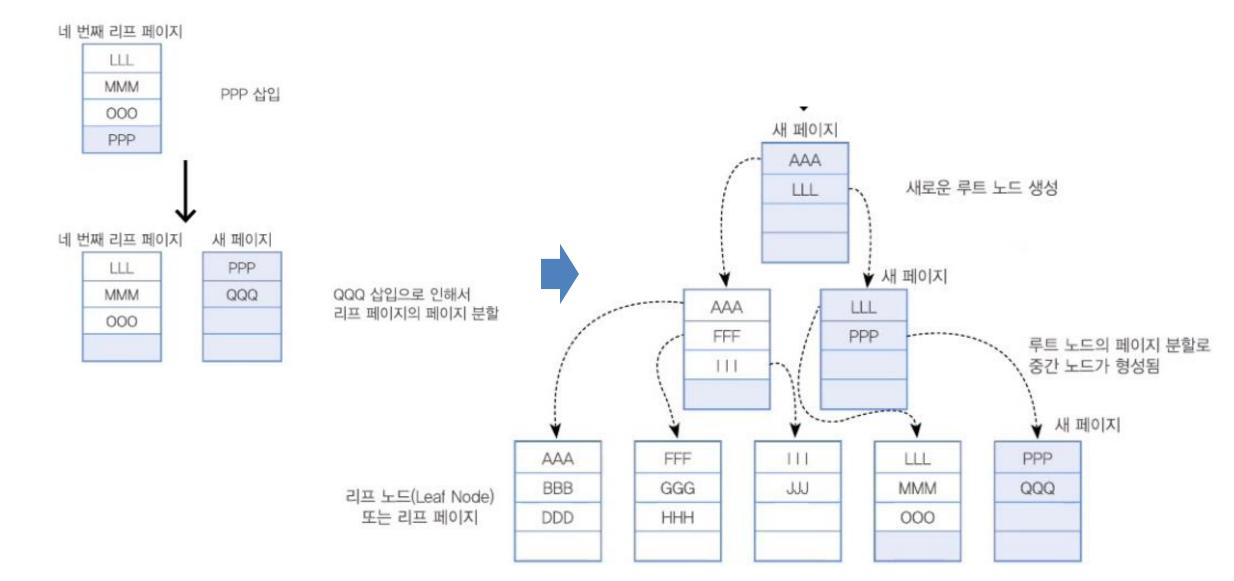
3 인덱스의 내부 작동

페이지 분할



3 인덱스의 내부 작동

페이지 분할



🥑 인덱스 생성

CREATE [UNIQUE | FULLTEXT | SPATIAL] INDEX 인덱스명 ON 테이블명(컬럼명, ...);

4 인덱스 생성/변경/삭제

☑ 인덱스 제거

DROP INDEX 인덱스명 ON 테이블명;

USE sqldb;

SELECT * FROM usertbl;

userID	name	birthYear	addr	mobile 1	mobile2	height	mDate
BBK	바비킴	1973	서울	010	0000000	176	2013-05-05
EJW	은지원	1972	경북	011	8888888	174	2014-03-03
ЗKW	조관우	1965	경기	018	9999999	172	2010-10-10
JYP	조용필	1950	경기	011	4444444	166	2009-04-04
KBS	김범수	1979	경남	011	2222222	173	2012-04-04
KKH	김경호	1971	전남	019	3333333	177	2007-07-07
LJB	임재범	1963	서울	016	6666666	182	2009-09-09
LSG	이승기	1987	서울	011	1111111	182	2008-08-08
SSK	성시경	1979	서울	HULL	MULL	186	2013-12-12
YJS	윤종신	1969	경남	NULL	HULL	170	2005-05-05



USE sqldb;

SHOW INDEX FROM usertbl;



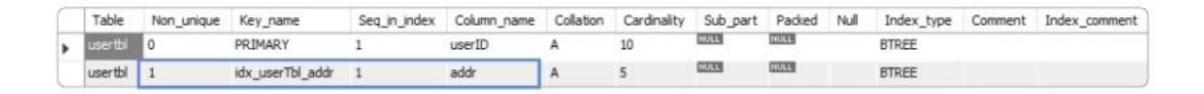
-- 인덱스의 크기 확인

SHOW TABLE STATUS LIKE 'usertbl';



-- addr 컬럼에 인덱스 만들기 CREATE INDEX idx_usertbl_addr ON usertbl(addr);

SHOW INDEX FROM usertbl;

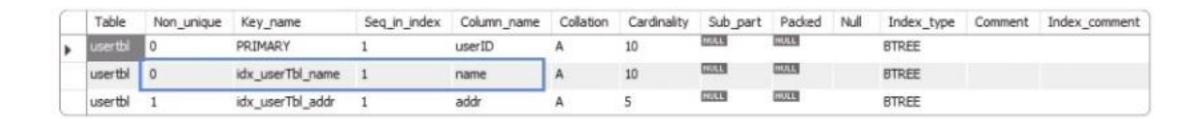


```
-- 인덱스 크기 확인
SHOW TABLE STATUS LIKE 'usertbl'; -- Index_lengt가 0
-- 생성한 인덱스를 실제 적용하려면 ANALYZE TABLE 문으로 먼저 테이블을 분석/처리해줘야 함
ANALYZE TABLE usertbl;
SHOW TABLE STATUS LIKE 'usertbl';
```

	Name	Engine	Version	Row_format	Rows	Avg_row_length	Data_length	Max_data_length	Index_length	Data_free	Auto_increment	Create_time
•	usertbl	InnoDB	10	Dynamic	10	1638	16384	0	16384	0	HULL	2016-03-05 20:53:55

- -- 출생년도(birthYear)에 보조 인덱스 생성 CREATE UNIQUE INDEX idx_usertbl_birthYear ON usertbl(birthYear);
- -- 중복값이 있어 인덱스 생성 에러 발생
- -- 이름(name)에 보조 인덱스 생성 CREATE UNIQUE INDEX idx_usertbl_name ON usertbl(name);

SHOW INDEX FROM usertbl;



```
-- 이름(name)에 보조 인덱스 삭제
DROP INDEX idx_usertbl_name ON usertbl;

-- name, birthYear 조합으로 인덱스 생성
CREATE UNIQUE INDEX idx_usertbl_name_birthYear
ON usertbl(name, birthYear);

SHOW INDEX FROM usertbl;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment
usertbl	0	PRIMARY	1	userID	A	10	HULL	MULL		BTREE	
usertbl	1	idx_usertbl_addr	1	addr	A	5	NULL	HULL		BTREE	
usertbl	1	idx_usertbl_name_birthYear	1	name	A	10	MULL	MULL		BTREE	
usertbl	1	idx_usertbl_name_birthYear	2	birthYear	A	10	HULL	MALE		BTREE	

```
-- 인덱스 삭제
SHOW INDEX FROM usertbl;

DROP INDEX idx_usertbl_addr ON usertbl;

DROP INDEX idx_usertbl_name_birthYear ON usertbl;

또는

ALTER TABLE usertbl DROP INDEX idx_usertbl_addr;
ALTER TABLE usertbl DROP INDEX idx_usertbl_name_birthYear;
```