

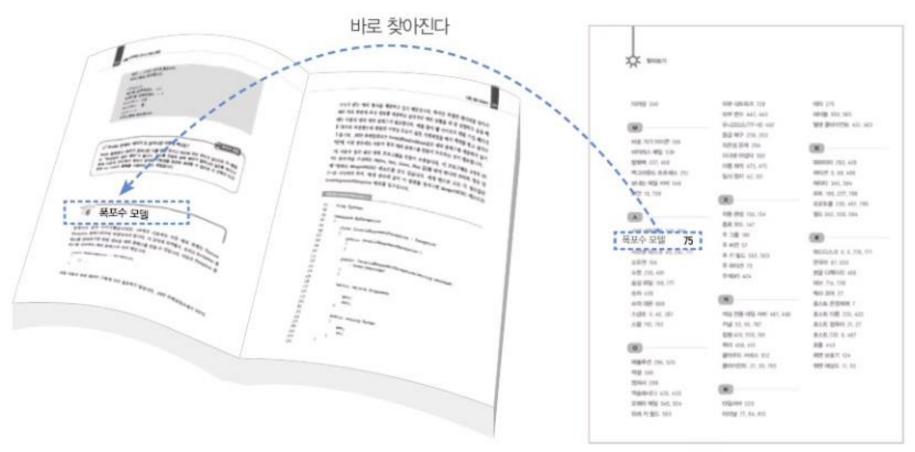
2025년 상반기 K-디지털 트레이닝

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

 - 테이블 당 하나만 생성 가능 ▶
- 보조 인덱스
 - 별도의 공간을 추가하여 지정한 컬럼만 정렬하여 인덱스를 운영
 - 컬럼 단위로 여러 개의 인덱스 생성 가능

unique 키워드를 사용하면 보조 인덱스가 자동 생성되고

수동으로 create index 명령을 통해 직접 생성할 수도 있다

☑ 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



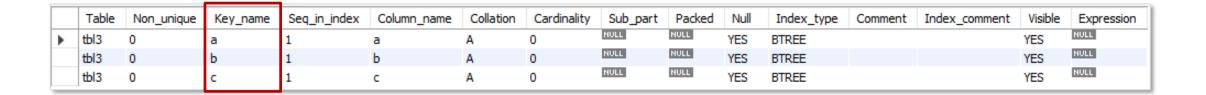
```
USE sqldb;
CREATE TABLE tbl1 (
  a INT PRIMARY KEY,
  b INT,
  c INT
);
SHOW INDEX FROM tbl1;
```

	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
•	tbl1	0	PRIMARY	1	a	Α	0	NULL	NULL		BTREE			YES	NULL

```
CREATE TABLE tbl2 (
  a INT PRIMARY KEY,
  b INT UNIQUE,
 c INT UNIQUE,
  d INT
SHOW INDEX FROM tbl2;
```

	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
•	tbl2	0	PRIMARY	1	a	Α	0	NULL	NULL		BTREE			YES	NULL
1	tbl2	0	b	1	Ь	Α	0	NULL	NULL	YES	BTREE			YES	NULL
1	tbl2	0	С	1	c	Α	0	NULL	NULL	YES	BTREE			YES	NULL

```
CREATE TABLE tbl3 (
  a INT UNIQUE,
  b INT UNIQUE,
  c INT UNIQUE,
  d INT
SHOW INDEX FROM tbl3;
```



2 인덱스의 종류와 자동 생성

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

	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
•	tbl4	0	a	1	a	Α	0	NULL	NULL		BTREE			YES	NULL
	tbl4	0	Ь	1	b	Α	0	NULL	NULL	YES	BTREE			YES	NULL
	tbl4	0	C	1	c	Α	0	NULL	NULL	YES	BTREE			YES	NULL

2 인덱스의 종류와 자동 생성

🗹 실습

```
CREATE TABLE tbl5 (
  a INT UNIQUE NOT NULL,
  b INT UNIQUE,
  c INT UNIQUE,
  d INT PRIMARY KEY
);
```

SHOW INDEX FROM tbl5;

	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
•	tbl5	0	PRIMARY	1	d	Α	0	NULL	NULL		BTREE			YES	NULL
	tbl5	0	a	1	a	Α	0	NULL	NULL		BTREE			YES	NULL
	tbl5	0	b	1	b	Α	0	NULL	NULL	YES	BTREE			YES	NULL
	tbl5	0	С	1	c	Α	0	NULL	NULL	YES	BTREE			YES	NULL

☑ 실습

```
CREATE DATABASE IF NOT EXISTS testdb;
USE testdb;
DROP TABLE IF EXISTS usertbl;
CREATE TABLE usertbl(
            CHAR(8) NOT NULL PRIMARY KEY,
 userID
            VARCHAR(10) NOT NULL,
  name
 birthYear INT NOT NULL
            NCHAR(2) NOT NULL
  addr
```



☑ 실습

```
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('SSK', '성시경', 1979, '서울');
```

SELECT * FROM usertbl;



☑ 실습

ALTER TABLE usertbl DROP PRIMARY KEY;

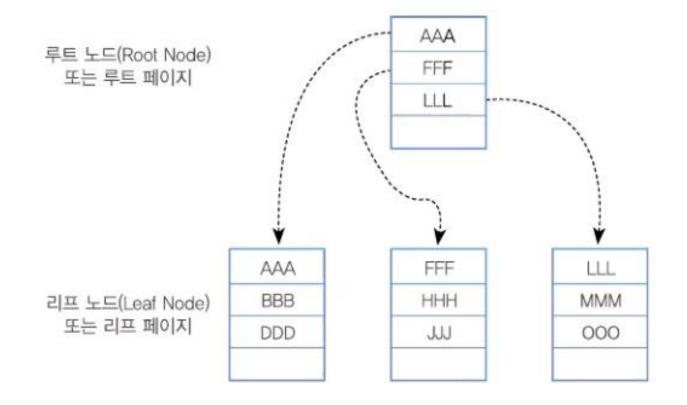
ALTER TABLE usertbl ADD CONSTRAINT pk_name PRIMARY KEY(name);

SELECT * FROM usertbl;



B-Tree(Balanced Tree, 균형 트리)

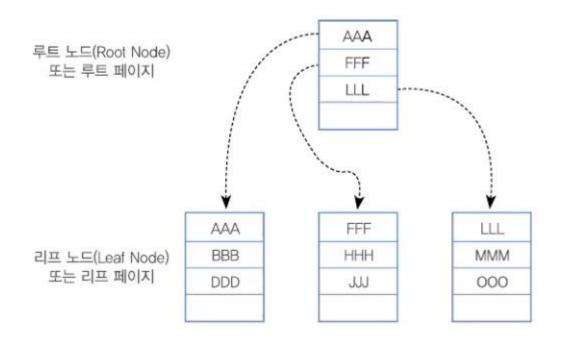
- o 페이지 Page
 - B-Tree의 노드
 - 16Kbyte의 고정 크기를 가짐
 - 데이터를 한 개만 저정하더라도 페이지 단위로 할당됨

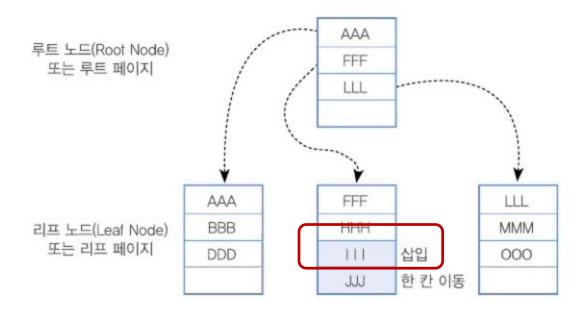


3 인덱스의 내부 작동

페이지 분할

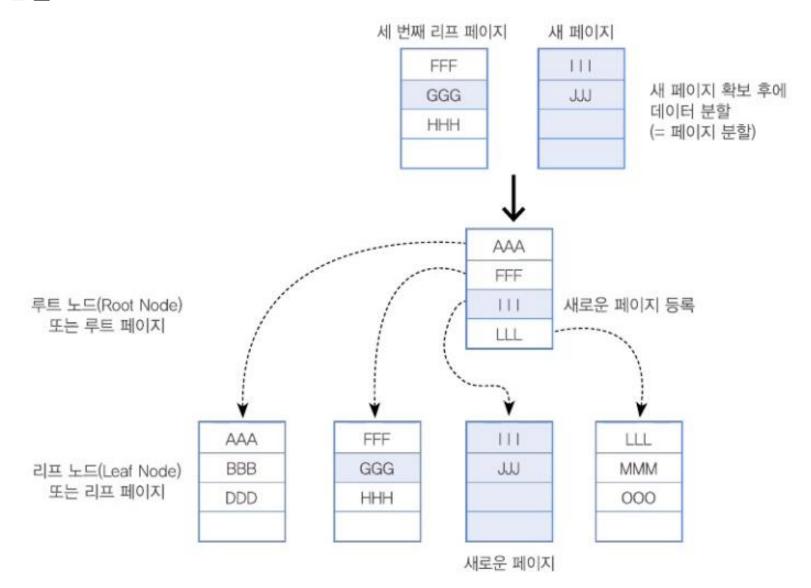
- 페이지에 공간이 없는 경우
 - 새로운 페이지를 생성
 - 기존 데이터를 양분하여 각각의 페이지에 재 배치
 - 링크 수정





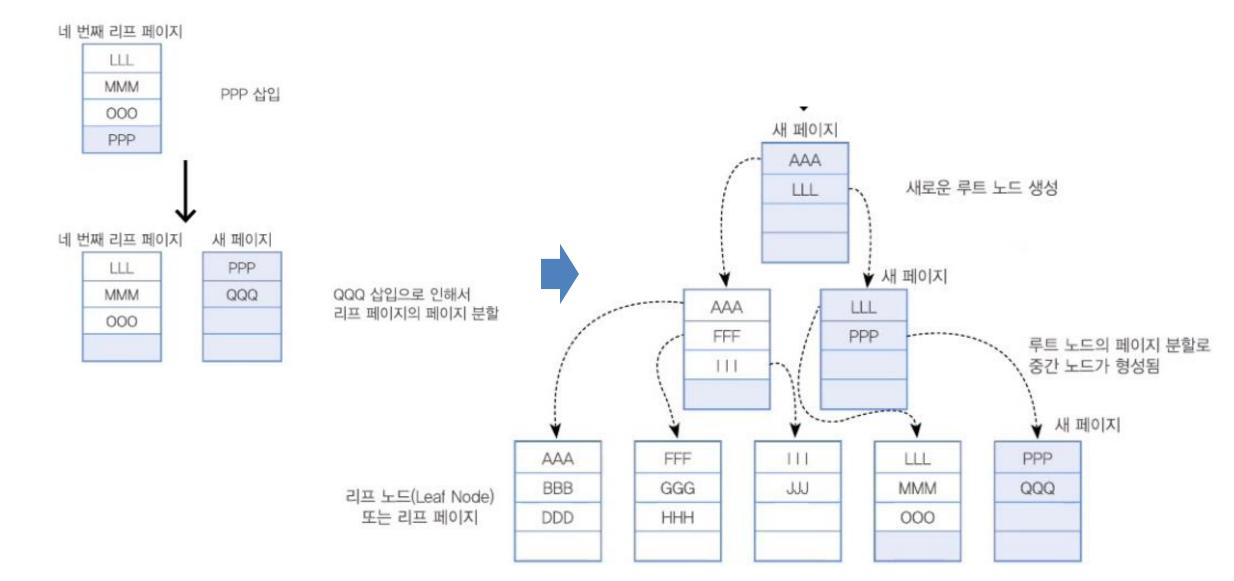
3 인덱스의 내부 작동

페이지 분할



3 인덱스의 내부 작동

페이지 분할



💟 인덱스 생성

옵션들

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

인덱스는 특정 기준으로 정렬하기에 orderby 연산시 추가적인 비용이 안드느느 편이다 하지만 인덱스 없는 테이블을 정렬하면orderby로 추가적인 비용이 많이 든다

하지만 공간 오버헤드가 생긴다.

추가적인 인덱스를 만들지말고 primary key를 이용하게끔 설계하는게 좋다

☑ 인덱스 제거

DROP INDEX 인덱스명 ON 테이블명;

인덱스는 물리적으로 어디에 저장될까

각 테이블의 테이블 스페이스 안에 같이 저장된다.

뷰 등등 테이블과 관련된것은 테이블스페이스에 저장됨

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
JKW	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 DATA 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	0	0	MOLL	2016-03-05 20:32:04

```
-- addr 컬럼에 인덱스 만들기
CREATE INDEX idx_usertbl_addr 그냥 인덱스
ON usertbl(addr);
                       만들때, 그 후에도 중복데이터 존재해도 됨
```

SHOW INDEX FROM usertbl;

	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
•	usertbl	0	PRIMARY	1	userID	A	10	HULL	HUEL		BTREE		
	usertbl	1	idx_userTbl_addr	1	addr	A	5	MULL	HOLL		BTREE		

4

🗹 실습

- -- 인덱스 크기 확인 SHOW TABLE STATUS LIKE 'usertbl'; -- lndex_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

인덱스 생성/변경/삭제

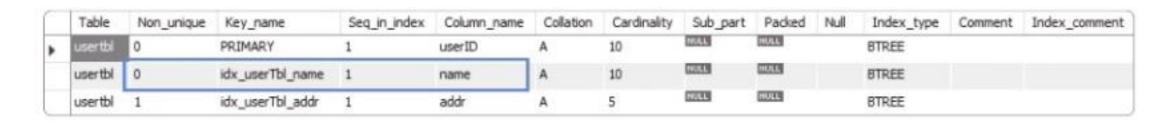
🗹 실습

-- 출생년도(birthYear)에 보조 인덱스 생성 CREATE UNIQUE INDEX idx_usertbl_birthYear ON usertbl(birthYear);

중복 데이터 안돼

- → 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	Α	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;
```



2025년 상반기 K-디지털 트레이닝

사용자 관리

[KB] IT's Your Life



사용자 관리

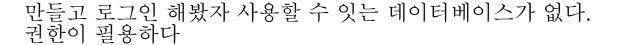
🕜 사용자 관리

○ 관리자(root) 권한으로 실행해야 함

ip나 %를 줄 수 있는데 어느 호스트에 따지는지는 ip. %는 따지지 않는 옵션이다 o **사용자 추가** ip에 대해서 제한을 걸 수 있따. 예) 'a'@'127.0.0.1'이라면 해당 컴퓨터에서만 ㅈ접속할 수 CREATE USER '사용자명'@'호스트명' IDENTIFIED BY '비밀번호';

○ **사용자 삭제**DROP USER '사용자명'@'호스트명'

- 사용자의 접속 호스트 제한 풀기
 - 호스트 명에 '%' 지정



사용자 관리

 ☑ 권한 관리
 모든 권한
 데이버 베이스 단위 혹은

 테이블 단위 로 타켓 범위를 줄 수있음

 GRANT ALL PRIVILEGES ON 데이터베이스명.객체 TO '사용자명'@'호스트명';

FLUSH PRIVILEGES;

- 데이터베이스의 모든 객체 지정 데이터베이스명.*
- o 권한 설정 적용하기 FLUSH PRIVILEGES;

사용자 관리

💟 실습 데이터베이스 및 사용자 추가하기

o 데이터베이스 명: scoula_db school 그리스어. 내일 부터 실습에 쓸 db

o 사용자 명: scoula

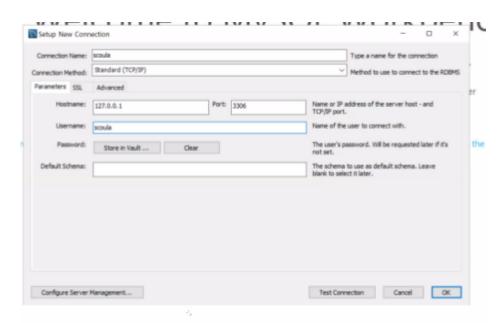
○ 비밀번호: 1234

○ 접속 호스트: 제한 없음

CREATE DATABASE scoula_db;

CREATE USER 'scoula'@'%' IDENTIFIED BY '1234';
GRANT ALL PRIVILEGES ON scoula_db.* TO 'scoula'@'%';
FLUSH PRIVILEGES;

비번 바꾸기근ㄴ alter user 'aaa' @ '...' identified by '바꿀비번내용';



만든 유저를 통해서 생성한db에 접속할 새로운 세션을 만들어서 사용해보자