

2025년 상반기 K-디지털 트레이닝

인덱스

[KB] IT's Your Life

 sqldb 데이터베이스에 다음과 같은 컬럼을 가지는 테이블 tbl1을 생성하고, 자동 생성된 인덱스 목록을 확인하세요.

○ 컬럼 목록

- a INT PRIMARY KEY
- b INT
- c INT

 sqldb 데이터베이스에 다음과 같은 컬럼을 가지는 테이블 tbl1을 생성하고, 자동 생성된 인덱스 목록을 확인하세요.

```
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
▶	tbl1	0	PRIMARY	1	a	A	0	NULL	NULL		BTREE		

 sqldb 데이터베이스에 다음과 같은 컬럼을 가지는 테이블 tbl2를 생성하고, 자동 생성된 인덱스 목록을 확인하세요.

○ 컬럼 목록

- a INT PRIMARY KEY
- b INT UNIQUE
- c INT UNIQUE
- d INT

📌 sqldb 데이터베이스에 다음과 같은 컬럼을 가지는 테이블 tbl2를 생성하고, 자동 생성된 인덱스 목록을 확인하세요.

```
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
▶	tbl2	0	PRIMARY	1	a	A	0	NULL	NULL		BTREE		
	tbl2	0	b	1	b	A	0	NULL	NULL	YES	BTREE		
	tbl2	0	c	1	c	A	0	NULL	NULL	YES	BTREE		

 sqldb 데이터베이스에 다음과 같은 컬럼을 가지는 테이블 tbl3를 생성하고, 자동 생성된 인덱스 목록을 확인하세요.

○ 컬럼 목록

- a INT UNIQUE,
- b INT UNIQUE,
- c INT UNIQUE,
- d INT

📌 sqldb 데이터베이스에 다음과 같은 컬럼을 가지는 테이블 tbl3를 생성하고, 자동 생성된 인덱스 목록을 확인하세요.

```
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
▶	tbl3	0	a	1	a	A	0	NULL	NULL	YES	BTREE		
	tbl3	0	b	1	b	A	0	NULL	NULL	YES	BTREE		
	tbl3	0	c	1	c	A	0	NULL	NULL	YES	BTREE		

 sqldb 데이터베이스에 다음과 같은 컬럼을 가지는 테이블 tbl4를 생성하고, 자동 생성된 인덱스 목록을 확인하세요.

○ 컬럼 목록

- a INT UNIQUE NOT NULL,
- b INT UNIQUE,
- c INT UNIQUE,
- d INT

📌 sqldb 데이터베이스에 다음과 같은 컬럼을 가지는 테이블 tbl4를 생성하고, 자동 생성된 인덱스 목록을 확인하세요.

```
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	NULL	NULL		BTREE		
	tbl4	0	b	1	b	A	0	NULL	NULL	YES	BTREE		
	tbl4	0	c	1	c	A	0	NULL	NULL	YES	BTREE		

 sqldb 데이터베이스에 다음과 같은 컬럼을 가지는 테이블 tbl5를 생성하고, 자동 생성된 인덱스 목록을 확인하세요.

○ 컬럼 목록

- a INT UNIQUE NOT NULL,
- b INT UNIQUE,
- c INT UNIQUE,
- d INT PRIMARY KEY

 sqldb 데이터베이스에 다음과 같은 컬럼을 가지는 테이블 tbl5를 생성하고, 자동 생성된 인덱스 목록을 확인하세요.

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

```
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
▶	tbl5	0	PRIMARY	1	d	A	0	NULL	NULL		BTREE		
	tbl5	0	a	1	a	A	0	NULL	NULL		BTREE		
	tbl5	0	b	1	b	A	0	NULL	NULL	YES	BTREE		
	tbl5	0	c	1	c	A	0	NULL	NULL	YES	BTREE		

 testdb에 다음 컬럼 목록을 가지는 usertbl을 만드세요.

○ 컬럼 목록

- userID CHAR(8) NOT NULL PRIMARY KEY
- name VARCHAR(10) NOT NULL
- birthYear INT NOT NULL
- addr NCHAR(2) NOT NULL

 아래 데이터를 추가하고, 클러스터형 인덱스의 테이블 내용을 확인하세요.

'LSG', '이승기', 1987, '서울'
'KBS', '김범수', 1979, '경남'
'KKH', '김경호', 1971, '전남'
'JYP', '조용필', 1950, '경기'
'S나', '성시경', 1979, '서울'

 testdb에 다음 컬럼 목록을 가지는 usertbl을 만드세요.

```
CREATE DATABASE IF NOT EXISTS 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  
);
```

✓ 아래 데이터를 추가하고, 클러스터형 인덱스의 테이블 내용을 확인하세요.

```
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('SL나', '성시경', 1979, '서울');
```

```
SELECT * FROM usertbl;
```

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

- ✎ ALTER를 사용하여 usertbl에서 PRIMARY KEY 제약조건을 제거하고, name 컬럼에 pk_name이라는 제약조건명으로 기본키를 설정하세요.
- ✎ usertbl의 내용을 출력하여, 새로운 클러스터형 인덱스를 확인하세요

✎ ALTER를 사용하여 usertbl에서 PRIMARY KEY 제약조건을 제거하고, name 컬럼에 pk_name이라는 제약조건명으로 기본키를 설정하세요.

✎ usertbl의 내용을 출력하여, 새로운 클러스터형 인덱스를 확인하세요

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

```
SELECT * FROM usertbl;
```

	userID	name	birthYear	addr
▶	KOH	김경호	1971	전남
	KBS	김범수	1979	경남
	SSK	성시경	1979	서울
	LSG	이슬기	1987	서울
	JYP	조용필	1950	경기