7. 뷰 및 인덱스 활용

뷰(VIEW) 의 개요

- **'가상의 테이블' – 물리적으로 구현되어 있지 않음.**
- 쿼리 창에서 **SELECT 문을 실행하여 나온 결과가 뷰**가 됨.
- 뷰의 실체는 SELECT 문
- 조인문 사용 최소화로 사용상 편의성을 최대화 함.
- 뷰에 나타나지 않는 데이터를 안전하게 보호할 수 있음.

뷰(VIEW) 생성

CREATE VIEW *뷰_이름* **AS SELECT문...**

emp_no	emp_nm	dept_no	pos	gender	hire_date	Salary
1001	구창민	1	과장	M	1995-05-01	5000000
1002	김민서	1	사원	M	2017-09-01	2500000
1003	이은영	2	부장	F	1990-09-01	5500000
1004	한성일	2	과장	M	1993-04-01	5000000

직원번호와 이름만 검색하여
뷰 생성

emp_no	emp_nm
1001	구창민
1002	김민서
1003	이은영
1004	한성일

```
CREATE VIEW emp_view as
SELECT emp_no, emp_nm FROM employee;
```

뷰(VIEW)

employee

emp_no	emp_nm	dept_no	pos	gender	hire_date	Salary
1001	구창민	1	과장	M	1995-05-01	5000000
1002	김민서	1	사원	M	2017-09-01	2500000
1003	이은영	2	부장	F	1990-09-01	5500000
1004	한성일	2	과장	M	1993-04-01	5000000

직급이 '과장 ' 인 직원들의 뷰(post_view)를 만드세요.

```
CREATE VIEW post_view as
SELECT * FROM employee WHERE position ='과장'
```

emp_no	emp_nm	dept_no	pos	gender	hire_date	Salary
1001	구창민	1	과장	M	1995-05-01	5000000
1004	한성일	2	과장	M	1993-04-01	5000000

뷰(VIEW)

- 뷰는 복잡한 질의를 간단하게 표현할 수 있게 함. 보안에도 좋음
- 뷰는 다중테이블에서 질의하여 표현할 수 있음.

employee

emp_no	emp_nm	dept_no	pos	gender	hire_date	Salary
1001	구창민	1	과장	M	1995-05-01	5000000
1002	김민서	1	사원	M	2017-09-01	2500000
1003	이은영	2	부장	F	1990-09-01	5500000
1004	한성일	2	과장	M	1993-04-01	5000000

department

dept_no	dept_nm	loc
1	영업부	대구
2	인사부	서울
3	총무부	대구
4	기획부	서울

depart_view

emp_no	emp_nm	dept_nm	pos
1001	구창민	영업부	과장
1002	김민서	영업부	사원
1003	이은영	인사부	부장
1004	한성일	인사부	과장

사원번호, 사원명, 부서명, 직급을 가진 뷰(depart_view)

```
CREATE VIEW depart_view as
SELECT emp_no, emp_nm, dept_nm, pos
FROM employee e join department d on e.dept_no=d.dept_no;
```

뷰(VIEW)

직급이 '사원'인 직원의 직원번호, 이름, 부서명, 직책, 월급, 작업위치를 조회하여 emp_dept_view를 생성하고 생성되었는지 확인하세요.

```
create view emp_dept_view as select emp_no, emp_nm, dept_nm, pos,
salary, loc from employee as e join department as d on
e.dept_no=d.dept_no where pos = '사원';
```

```
>> show tables;
>> desc emp_dept_view;
>> select * from emp_dept_view;
로 확인
```

emp_dept_view에서 사원의 이름과 작업 위치를 조회하세요.

```
select emp_nm, loc from emp_dept_view;
```

뷰(VIEW) 삭제

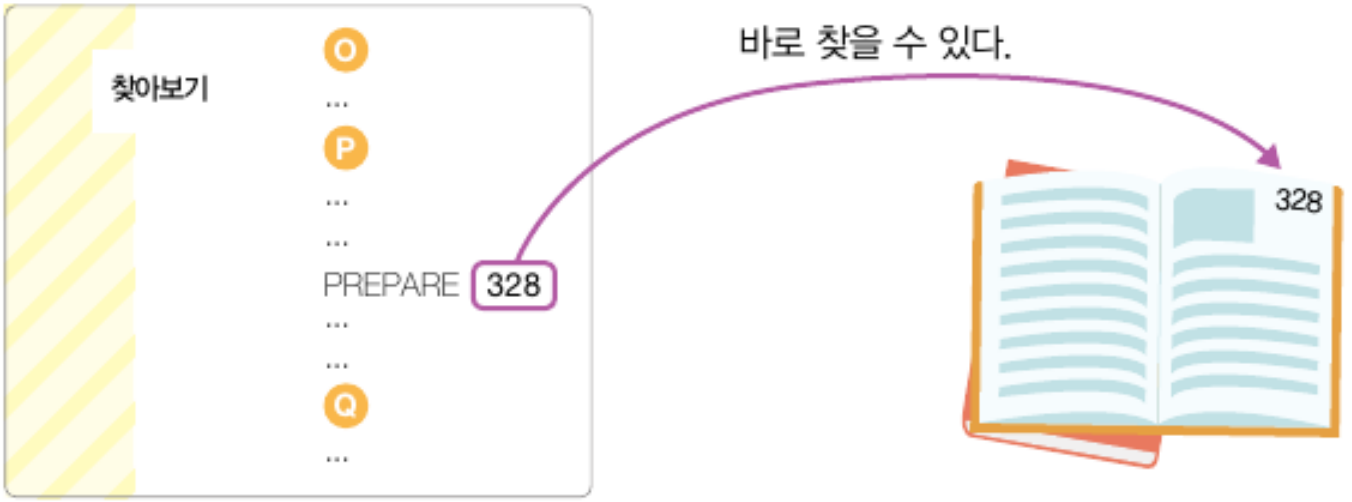
DROP VIEW *뷰_이름* ;

```
DROP VIEW emp_view;
DROP VIEW emp_dept_view;
```

데이터는 삭제되지 않음.

인덱스의 개요

- 인덱스(index)는 데이터베이스 테이블에 대한 **검색 성능의 속도를 높여주는** 자료구조
- 책의 목차 또는 찾아보기와 비슷한 개념.
- '**데이터를 좀 더 빨리 찾을 수 있도록 도와주는 도구**'



책의 찾아보기

인덱스 장단점

장점

- 테이블 검색 속도와 성능이 향상됨.

단점

- 인덱스를 관리하기 위한 추가 작업이 필요.(테이블에 변화가 생기면 값들을 다시 정렬해야 함)
- 추가 저장 공간 필요(대략 데이터베이스 크기의 10% 정도 추가 공간이 필요)
- 잘못 사용하는 경우 오히려 검색 성능 저하

인덱스 종류

인덱스 종류

- MySQL/mariaDB에서 사용하는 인덱스에는 **클러스터형 인덱스(clustered index)**와 **보조 인덱스 (secondary index)**가 있음

클러스터형 인덱스

- 테이블이 기본키(primary key)에 의해 정렬되고, 그 기본키에 대해 만들어진 인덱스
- 기본키(또는 not null의 unique키) 선언될 때 자동 생성
- **테이블당 하나만 생성**할 수 있음.(기본키가 있으면 unique 그리고 not null 이라도 보조 인덱스임)

보조 인덱스

- 기본키를 제외한 나머지 모든 인덱스
- **create index** 문장에 의해 생성됨.
- unique, foreign key 선언할 때 자동 생성.
- 테이블당 여러 개를 생성할 수 있음

클러스터형 인덱스

```
CREATE TABLE v1(t1 int, t2 int, t3 int,primary key(t1));
SHOW INDEX from v1;
```

클러스터형 인덱스 – t1

```
MariaDB [olympicsdb]> show index from v1;
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
| v1    |          0 | PRIMARY  |            1 | t1          | A         |           0 |     NULL | NULL   |      | BTREE      |         |               |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
```

```
CREATE TABLE v2(t1 int, t2 int unique, t3 int,primary key(t1));
SHOW INDEX from v2;
```

클러스터형 인덱스 – t1

보조 인덱스 – t2

```
MariaDB [olympicsdb]> show index from v2;
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
| v2    |          0 | PRIMARY  |            1 | t1          | A         |           0 |     NULL | NULL   |      | BTREE      |         |               |
| v2    |          0 | t2       |            1 | t2          | A         |           0 |     NULL | NULL   | YES  | BTREE      |         |               |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
```

```
CREATE TABLE v3(t1 int, t2 int not null unique, t3 int unique);
SHOW INDEX from v3;
```

클러스터형 인덱스 – t2

보조 인덱스 – t3

```
MariaDB [olympicsdb]> show index from v3;
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
| v3    |          0 | t2       |            1 | t2          | A         |           0 |     NULL | NULL   |      | BTREE      |         |               |
| v3    |          0 | t3       |            1 | t3          | A         |           0 |     NULL | NULL   | YES  | BTREE      |         |               |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
```

클러스터형 인덱스 정렬 확인하기

```
MariaDB [olympicsdb]> insert into v1 values(1,2,3);
Query OK, 1 row affected (0.005 sec)

MariaDB [olympicsdb]> select * from v1;
+----+------+------+
| t1 | t2   | t3   |
+----+------+------+
|  1 |    2 |    3 |
+----+------+------+
1 row in set (0.000 sec)

MariaDB [olympicsdb]> insert into v1 values(5,2,3);
Query OK, 1 row affected (0.004 sec)

MariaDB [olympicsdb]> select * from v1;
+----+------+------+
| t1 | t2   | t3   |
+----+------+------+
|  1 |    2 |    3 |
|  5 |    2 |    3 |
+----+------+------+
2 rows in set (0.000 sec)

MariaDB [olympicsdb]> insert into v1 values(3,2,3);
Query OK, 1 row affected (0.005 sec)

MariaDB [olympicsdb]> select * from v1;
+----+------+------+
| t1 | t2   | t3   |
+----+------+------+
|  1 |    2 |    3 |
|  3 |    2 |    3 |
|  5 |    2 |    3 |
+----+------+------+
3 rows in set (0.000 sec)
```

t1을 기준으로 데이터가 정렬됨.

보조 인덱스 만들기, 인덱스 삭제하기

CREATE INDEX *index_name*
ON *table_name* (*column1*, *column2*, ...);

DROP INDEX *index_name*
ON *table_name*;

employee 테이블의 이름(emp_nm)필드에 인덱스(인덱스명 : nameIDX) 만들기

```
SHOW INDEX from employee;                    // 인덱스 만들기 전 인덱스 확인
CREATE INDEX nameIDX ON employee(emp_nm);
SHOW INDEX from employee;                    // 인덱스 만들기 후 인덱스 확인
```

```
MariaDB [com_1]> show index from employee;
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+
| Table    | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+
| employee |          0 | PRIMARY  |            1 | emp_no      | A         |           5 |     NULL | NULL   |      | BTREE      |
| employee |          1 | nameIDX  |            1 | emp_nm      | A         |           5 |     NULL | NULL   | YES  | BTREE      |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+
```

인덱스(인덱스명 : nameIDX) 삭제하기

```
DROP INDEX nameIDX ON employee;
SHOW INDEX from employee;
```

보조 인덱스 만들기(2개 컬럼에)

employee 테이블의 사원명과 부서필드에 인덱스(인덱스명 : empIDX) 만들기

```
CREATE INDEX empIDX ON employee(emp_nm, dept_no);
SHOW INDEX from employee;
```

```
MariaDB [com_1]> SHOW INDEX from employee;
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---
| Table    | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | C
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---
| employee |          0 | PRIMARY  |            1 | emp_no      | A         |           5 |     NULL | NULL   |      | BTREE      |
| employee |          1 | empIDX   |            1 | emp_nm      | A         |           5 |     NULL | NULL   | YES  | BTREE      |
| employee |          1 | empIDX   |            2 | dept_no     | A         |           5 |     NULL | NULL   | YES  | BTREE      |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---
```