
테이블과 뷰 2

뷰

❖ 뷰(View)

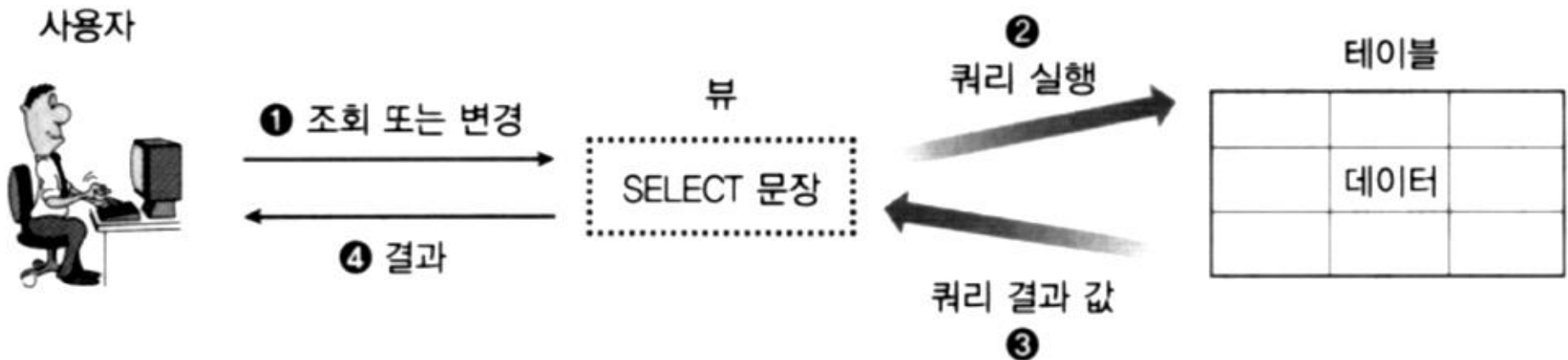
- 테이블과 동일하게 사용되는 개체
- SELECT 쿼리 결과를 하나의 테이블로 간주
 - 여기에 이름을 지정하고 이를 통해 사용

[형식]

CREATE VIEW 뷰이름

AS

SELECT 쿼리



뷰

❖ 뷰 만들기

```
USE tableDB;
```

```
CREATE VIEW v_userTBL  
AS
```

```
    SELECT userid, name, addr FROM userTBL;
```

```
SELECT * FROM v_userTBL;  -- 뷰를 테이블이라고 생각해도 무방
```

뷰

❖ 뷰 만들기

```
SELECT U.userid, U.name, B.prodName, U.addr,  
       CONCAT(U.mobile1, U.mobile2) AS '연락처'  
FROM userTBL U  
     INNER JOIN buyTBL B  
       ON U.userid = B.userid ;
```

```
CREATE VIEW v_userbuyTBL  
AS  
SELECT U.userid, U.name, B.prodName, U.addr,  
       CONCAT(U.mobile1, U.mobile2) AS '연락처'  
FROM userTBL U  
     INNER JOIN buyTBL B  
       ON U.userid = B.userid ;
```

```
SELECT * FROM v_userbuyTBL WHERE name = '김범수';
```

뷰

❖ 뷰 만들기

```
USE sqlDB;
CREATE VIEW v_userbuyTBL
AS
    SELECT U.userid AS 'USER ID', U.name AS 'USER NAME',
           B.prodName AS 'PRODUCT NAME', U.addr,
           CONCAT(U.mobile1, U.mobile2) AS 'MOBILE PHONE'
    FROM userTBL U
         INNER JOIN buyTBL B
         ON U.userid = B.userid;
```

```
SELECT `USER ID`, `USER NAME` FROM v_userbuyTBL; -- 주의! 백틱을 사용한다.
```

Alias를 통해 뷰의 컬럼명 수정 가능

뷰

❖ 뷰 만들기

```
ALTER VIEW v_userbuyTBL
AS
    SELECT U.userid AS '사용자 아이디', U.name AS '이름',
           B.prodName AS '제품 이름', U.addr,
           CONCAT(U.mobile1, U.mobile2) AS '전화 번호'
    FROM userTBL U
         INNER JOIN buyTBL B
           ON U.userid = B.userid ;

SELECT `이름`, `전화 번호` FROM v_userbuyTBL;
```

뷰

❖ 뷰의 삭제

[형식]

```
DROP VIEW 뷰이름
```

```
DROP VIEW v_userbuyTBL;
```