

▶ Chapter 09 : 뷰

SQL 활용 프로그래밍

SQL Application Programming





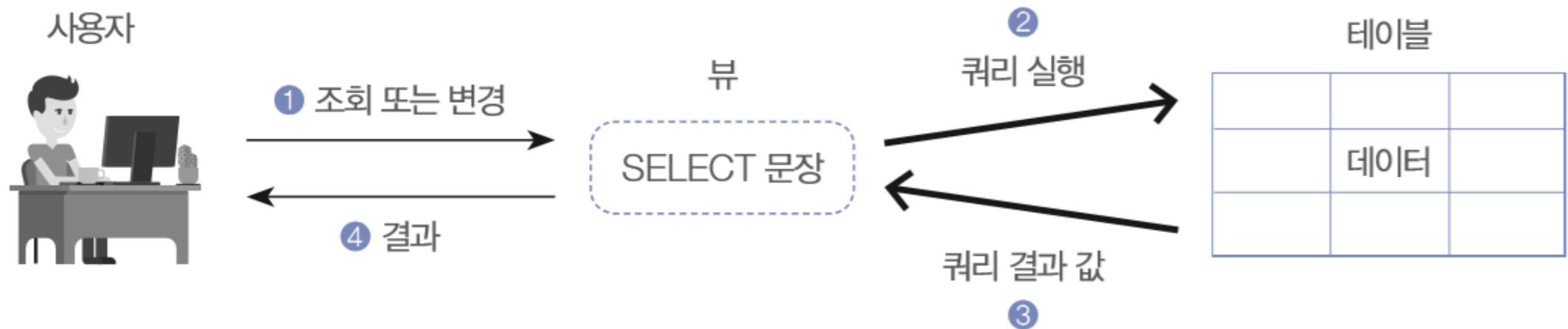
CHAPTER 09 VIEW

SECTION 03 VIEW

뷰의 개념

- 일반 사용자 입장에서 테이블과 동일하게 사용하는 개체
 - 뷰를 생성한 후에는 테이블처럼 접근 가능하여 동일한 결과 얻을수 있음

뷰의 작동 방식



[그림 8-27] 뷰의 작동 방식

SECTION 03 VIEW

뷰의 개념

- 뷰 생성 구문

```
CREATE VIEW v_usertbl  
AS SELECT userid, name, addr FROM usertbl;
```

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

	userid character (8) 🔒	name character varying (10) 🔒	addr character (2) 🔒
1	LSG	이승기	서울
2	JYP	조용필	경기
3	KBS	김범수	경남
4	KKH	김경호	전남
5	EJW	은지원	경북
6	VVK	바비킴	서울

SECTION 03 VIEW

뷰의 장점

- 보안에 도움
 - 사용자가 중요한 정보에 바로 접근하지 못함
- 복잡한 쿼리 단순화
 - 긴 쿼리를 뷰로 작성, 뷰를 테이블처럼 사용 가능

```
SELECT U.userid, U.name, B.prodName, U.addr, CONCAT(U.mobile1, U.mobile2) AS PH
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 PH
FROM usertbl U
INNER JOIN buytbl B
ON U.userid = B.userid;
```

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

	userid character (8) 🔒	name character varying (10) 🔒	prodname character (6) 🔒	addr character (2) 🔒	ph text 🔒
1	KBS	김범수	운동화	경남	01122222222
2	KBS	김범수	노트북	경남	01122222222
3	KBS	김범수	청바지	경남	01122222222

[실습 1]

뷰를 생성해서 활용하자.

뷰 생성 및 활용

- 기본적인 뷰 생성 및 뷰에서 사용될 열 이름 변경

```
DROP VIEW v_userbuytbl;
```

```
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;
```

	user_id character (8) 🔒	user_name character varying (10) 🔒
1	KBS	김범수
2	KBS	김범수
3	JYP	조용필
4	KBS	김범수
5	VVK	바비킴
6	VVK	바비킴
7	VVK	바비킴

- 뷰 수정 및 삭제

```
ALTER VIEW [ IF EXISTS ] name ALTER [ COLUMN ] column_name SET DEFAULT expression  
ALTER VIEW [ IF EXISTS ] name ALTER [ COLUMN ] column_name DROP DEFAULT  
ALTER VIEW [ IF EXISTS ] name OWNER TO { new_owner | CURRENT_ROLE | CURRENT_USER | SESSION_USER }  
ALTER VIEW [ IF EXISTS ] name RENAME [ COLUMN ] column_name TO new_column_name  
ALTER VIEW [ IF EXISTS ] name RENAME TO new_name  
ALTER VIEW [ IF EXISTS ] name SET SCHEMA new_schema  
ALTER VIEW [ IF EXISTS ] name SET ( view_option_name [= view_option_value] [, ... ] )  
ALTER VIEW [ IF EXISTS ] name RESET ( view_option_name [, ... ] )
```

```
ALTER VIEW v_userbuytbl RENAME TO v_userbuynewtbl;
```

```
SELECT USER_ID, USER_NAME from v_userbuytbl;
```

- 뷰 수정 ALTER VIEW
- 뷰 삭제 DROP VIEW

	user_id character (8) 🔒	user_name character varying (10) 🔒
1	KBS	김범수
2	KBS	김범수
3	JYP	조용필
4	KBS	김범수
5	VVK	바비킴
6	VVK	바비킴
7	VVK	바비킴

[실습 1]

뷰를 생성해서 활용하자.

뷰 생성 및 활용

- 뷰를 통한 데이터 변경

```
UPDATE v_usertbl SET addr = '부산' WHERE userid='JYP';
```

```
SELECT * from v_userbuytbl;
```

```
INSERT INTO v_usertbl(userid, name, addr) VALUES('KBM', '김병만', '충북');
```

	user_id character (8)	user_name character varying (10)	product_name character (6)	addr character (2)	mobile_phone text
1	KBS	김범수	운동화	경남	01122222222
2	KBS	김범수	노트북	경남	01122222222
3	JYP	조용필	모니터	부산	01111111111
4	KBS	김범수	청바지	경남	01122222222
5	VVK	바비킴	모니터	서울	01000000000
6	VVK	바비킴	메모리	서울	01000000000
7	VVK	바비킴	운동화	서울	01000000000

- 데이터 수정은 정상적으로 되지만 데이터 입력 시 오류 발생!!
- V_usertbl이 참조하는 테이블 usertbl의 열 중에서 birthYear열은 NOT NULL로 설정되어서 반드시 값을 입력 해주거나 DEFAULT가 지정되어 있으면 됨
- 하지만 현재는 birthYear를 참조하고 있지 않으므로 값 입력 불가
- 값을 v_usertbl을 통해서 입력하고 싶을 경우 v_usertbl에 birthYear를 포함하도록 재정의하거나 usertbl에서 birthYear를 NULL 또는 DEFAULT 값을 지정

- 그룹 함수를 포함하는 뷰 정의

```
CREATE VIEW v_sum
```

```
AS
```

```
SELECT userid AS USERID, SUM(price*amount) AS TOTAL  
FROM buytbl GROUP BY userid;
```

```
SELECT * FROM v_sum;
```

	userid character (8)	total bigint
1	KBS	1210
2	JYP	200
3	VVK	1860

[실습 1]

뷰를 생성해서 활용하자.

뷰 생성 및 활용

- 뷰 정보 확인

```
SELECT * FROM INFORMATION_SCHEMA.VIEWS
WHERE TABLE_SCHEMA = 'public' AND table_NAME = 'v_sum';
```

- SUM() 함수를 사용한 뷰 수정 불가
- 시스템 DB 중 하나인 INFORMATION_SCHEMA의 VIEWS 테이블에서 전체 시스템에 저장된 뷰에 대한 다양한 정보 포함

	table_catalog name	table_schema name	table_name name	view_definition character varying	check_option character varying	is_updatable character varying (3)	is_insertable_into character varying (3)	is_trigger_updatable character varying (3)	is_trigger_deletable character varying (3)	is_trigger_insertable_into character varying (3)
1	tabledb	public	v_sum	SELECT buytbl.useri...	NONE	NO	NO	NO	NO	NO

- IS_UPDATABLE이 NO로 되어있음
- 이 뷰를 통해 데이터 변경(INSERT, UPDATE, DELETE) 불가
- 집계 함수를 사용한 뷰
- UNION ALL, JOIN 등을 사용한 뷰
- DISTINCT, GROUP BY 등을 사용한 뷰 또한 뷰를 통해 데이터 수정/삭제 불가

[실습 1]

뷰를 생성해서 활용하자.

뷰 생성 및 활용

- 지정된 범위로 뷰 생성 및 데이터 입력

```
CREATE VIEW v_height177
AS
SELECT * FROM usertbl WHERE height >= 177;

SELECT * FROM v_height177;
```

	userid character (8)	name character varying (10)	birthyear integer	addr character (2)	mobile1 character (3)	mobile2 character (8)	height smallint	mdate date
1	LSG	이승기	1987	서울	011	11111111	182	2008-08-08
2	KKH	김경호	1971	전남	019	33333333	177	2007-07-07

- V_heigthe177 뷰에서 키 177이하인 데이터 삭제 및 177미만 데이터 입력

```
DELETE FROM v_height177 WHERE height < 177; 0 row(s) affected
```

```
INSERT INTO v_height177 VALUES('KBM', '김병만', 1977, '경기', '010', '55555555', 158, '2023-01-01'); 1 row(s) affected
```

	userid character (8)	name character varying (10)	birthyear integer	addr character (2)	mobile1 character (3)	mobile2 character (8)	height smallint	mdate date
1	LSG	이승기	1987	서울	011	11111111	182	2008-08-08
2	KKH	김경호	1971	전남	019	33333333	177	2007-07-07

- 입력된 값이 보이지 않음
- 입력이 되더라도 입력된 값은 키가 177 미만이므로 v_height177 뷰에는 보이지 않고, 직접 usertbl을 확인해야 김병만이 보임

	userid [PK] character (8)	name character varying (10)	birthyear integer	addr character (2)	mobile1 character (3)	mobile2 character (8)	height smallint	mdate date
1	LSG	이승기	1987	서울	011	11111111	182	2008-08-08
2	KBS	김범수	1979	경남	011	22222222	173	2012-04-04
3	KKH	김경호	1971	전남	019	33333333	177	2007-07-07
4	EJW	은지원	1972	경북	011	88888888	174	2014-03-05
5	VVK	바비킴	1973	사울	010	00000000	176	2013-05-05
6	JYP	조용필	1950	부산	011	11111111	166	2009-04-04
7	KBM	김병만	1977	경기	010	55555555	158	2023-01-01

[실습 1]

뷰를 생성해서 활용하자.

뷰 생성 및 활용

- WITH CHECK OPTION

```
CREATE VIEW v_height177
AS SELECT * FROM usertbl WHERE height >= 177
WITH CHECK OPTION;
```

```
INSERT INTO v_height177 VALUES('WDT','서장훈',2006,'서울','010','33333333',155,'2023-3-3');
```

ERROR: 오류: 새 자료가 "v_height177" 뷰의 체크 제약 조건을 위반했습니다

DETAIL: 실패한 자료: (WDT, 서장훈, 2006, 서울, 010, 33333333, 155, 2023-03-03)

키 177 미만 입력되지 않는 것 확인

- 두 개 이상 테이블 관련되는 복합 뷰 생성 및 데이터 입력

```
CREATE VIEW v_userbuytbl
AS
SELECT U.userid, U.name, B.prodName, U.addr, CONCAT(U.mobile1, U.mobile2) AS mobile
FROM usertbl U
INNER JOIN buytbl B
ON U.userid = B.userid;
```

두 개 이상의 테이블이 관련된 뷰는 업데이트 할 수 없음

```
INSERT INTO v_userbuytbl VALUES('PKL','박경리','문동화','01000000000','2023-2-2');
```

ERROR: 오류: "v_userbuytbl" 뷰에 자료를 입력할 수 없습니다

DETAIL: Views that do not select from a single table or view are not automatically updatable.

HINT: 뷰를 통해 자료를 입력하려면, INSTEAD OF INSERT 트리거나 ON INSERT DO INSTEAD 롤을 사용하세요

