

-- Table & Constraints

DROP DATABASE IF EXISTS ShopDB;

DROP DATABASE IF EXISTS ModelDB;

DROP DATABASE IF EXISTS sqlDB;

DROP DATABASE IF EXISTS tableDB;

DROP DATABASE tableDB;

CREATE DATABASE tableDB;

USE tableDB;

DROP TABLE IF EXISTS buyTbl, userTbl;

CREATE TABLE userTbl -- 회원 테이블

( userID char(8), -- 사용자 아이디 -- 컬럼명 데이터타입, - 반복

name nvarchar(10), -- 이름

birthYear int, -- 출생년도

addr nchar(2), -- 지역(경기,서울,경남 등으로 글자만 입력) -- unicode(전세계 문자 표현 표준), utf-8

mobile1 char(3), -- 휴대폰의국번(011, 016, 017, 018, 019, 010 등) -- ascii - 영문 문자 인코딩 방식

mobile2 char(8), -- 휴대폰의 나머지 전화번호(하이픈 제외)

height smallint, -- 키

mDate date -- 회원 가입일

);

CREATE TABLE buyTbl -- 구매 테이블

( num int, -- 순번(PK)

userid char(8),-- 아이디(FK)

```
    prodName nchar(6), -- 물품명
    groupName nchar(4) , -- 분류
    price      int , -- 단가
    amount     smallint -- 수량
);
```

```
USE tableDB;
```

```
DROP TABLE IF EXISTS buyTbl, userTbl;
```

```
CREATE TABLE userTbl
```

```
( userID  char(8) NOT NULL ,
  name    varchar(10) NOT NULL,
  birthYear  int NOT NULL,
  addr     char(2) NOT NULL,
  mobile1 char(3) NULL,
  mobile2  char(8) NULL,
  height   smallint NULL,
  mDate    date NULL
```

```
);
```

```
CREATE TABLE buyTbl
```

```
(  num int NOT NULL ,
  userid  char(8) NOT NULL ,
  prodName char(6) NOT NULL,
  groupName char(4) NULL ,
  price    int  NOT NULL,
  amount   smallint  NOT NULL
```

```
);
```

```
-- 제약조건(constraints) -
```

/\* 제약조건

primary key - unique, not null

foreign key -

unique

default

null, not null

\*/

DROP TABLE IF EXISTS buyTbl, userTbl;

CREATE TABLE userTbl

( userID char(8) NOT NULL PRIMARY KEY, -- 회원 아이디, 대부분 테이블에 설정, 하나 이상의 열에 가능

name varchar(10) NOT NULL,

birthYear int NOT NULL,

addr char(2) NOT NULL,

mobile1 char(3) NULL,

mobile2 char(8) NULL,

height smallint NULL,

mDate date NULL

);

CREATE TABLE buyTbl

( num int NOT NULL PRIMARY KEY,

userid char(8) NOT NULL ,

prodName char(6) NOT NULL,

groupName char(4) NULL ,

price int NOT NULL,

amount smallint NOT NULL

);

```
use tabledb;

DROP TABLE IF EXISTS buyTbl;

CREATE TABLE buyTbl
( num int AUTO_INCREMENT NOT NULL PRIMARY KEY, -- auto_increment - primary key or unique Key
  userid char(8) NOT NULL ,
  prodName char(6) NOT NULL,
  groupName char(4) NULL ,
  price int NOT NULL,
  amount smallint NOT NULL
);
```

```
DROP TABLE IF EXISTS buyTbl;

CREATE TABLE buyTbl
( num int AUTO_INCREMENT NOT NULL PRIMARY KEY ,
  userid char(8) NOT NULL,
  prodName char(6) NOT NULL,
  groupName char(4) NULL ,
  price int NOT NULL,
  amount smallint NOT NULL
```

```
, FOREIGN KEY(userid) REFERENCES userTbl(userID) -- foreign key  
);
```

```
INSERT INTO userTbl VALUES('LSG', '이승기', 1987, '서울', '011', '11111111', 182, '2008-8-8');
```

```
INSERT INTO userTbl VALUES('KBS', '김범수', 1979, '경남', '011', '22222222', 173, '2012-4-4');
```

```
INSERT INTO userTbl VALUES('KKH', '김경호', 1971, '전남', '019', '33333333', 177, '2007-7-7');
```

```
select * from usertbl;
```

```
INSERT INTO buyTbl VALUES(NULL, 'JYP', '모니터', '전자', 200, 1);
```

```
INSERT INTO buyTbl VALUES(NULL, 'KBS', '노트북', '전자', 1000, 1);
```

```
INSERT INTO buyTbl VALUES(NULL, 'KBS', '운동화', NULL, 30, 2); -- error
```

```
select * from buytbl;
```

```
INSERT ignore INTO buyTbl VALUES(NULL, 'JYP', '모니터', '전자', 200, 1);
```

```
INSERT ignore INTO buyTbl VALUES(NULL, 'KBS', '노트북', '전자', 1000, 1);
```

```
INSERT ignore INTO buyTbl VALUES(NULL, 'KBS', '운동화', NULL, 30, 2); -- error
```

```
select * from buytbl;
```

-- <Primary Key> -- 3가지 방법

drop table usertbl;

alter table buytbl drop foreign key buytbl\_ibfk\_1;

drop table usertbl;

-- 1

CREATE TABLE userTbl

( userID char(8) NOT NULL PRIMARY KEY,

name varchar(10) NOT NULL,

birthYear int NOT NULL,

addr char(2) NOT NULL,

mobile1 char(3) NULL,

mobile2 char(8) NULL,

height smallint NULL,

mDate date NULL

);

DESCRIBE userTBL;

DROP TABLE IF EXISTS userTbl;

-- 2

CREATE TABLE userTbl

( userID char(8) NOT NULL,

name varchar(10) NOT NULL,

```
birthYear    int NOT NULL,

addr         char(2) NOT NULL,

mobile1      char(3) NULL,

mobile2      char(8) NULL,

height       smallint NULL,

mDate        date NULL,

CONSTRAINT PRIMARY KEY PK_userTbl_userID (userID)

);
```

```
DROP TABLE IF EXISTS userTbl;
```

```
-- 3
```

```
CREATE TABLE userTbl

( userID      char(8) NOT NULL,

  name        varchar(10) NOT NULL,

  birthYear   int NOT NULL,

  addr        char(2) NOT NULL,

  mobile1     char(3) NULL,

  mobile2     char(8) NULL,

  height      smallint NULL,

  mDate       date NULL

);
```

```
ALTER TABLE userTbl

    ADD CONSTRAINT PK_userTbl_userID

    PRIMARY KEY (userID);
```

-- 2개의 칼럼을 묶어서 primary key지정

DROP TABLE IF EXISTS prodTbl;

CREATE TABLE prodTbl

( prodCode CHAR(3) NOT NULL,

prodID CHAR(4) NOT NULL,

prodDate DATETIME NOT NULL,

prodCur CHAR(10) NULL

);

ALTER TABLE prodTbl

ADD CONSTRAINT PK\_prodTbl\_proCode\_prodID

PRIMARY KEY (prodCode, prodID) ;

DROP TABLE IF EXISTS prodTbl;

CREATE TABLE prodTbl

( prodCode CHAR(3) NOT NULL,

prodID CHAR(4) NOT NULL,

prodDate DATETIME NOT NULL,

prodCur CHAR(10) NULL,

CONSTRAINT PK\_prodTbl\_proCode\_prodID

PRIMARY KEY (prodCode, prodID)

);

DROP TABLE IF EXISTS prodTbl;

CREATE TABLE prodTbl

( prodCode CHAR(3) NOT NULL,



```
prodID CHAR(4) NOT NULL,  
prodDate DATETIME NOT NULL,  
prodCur CHAR(10) NULL,  
PRIMARY KEY (prodCode, prodID)  
);
```

-- Foreign Key

-- 두 테이블의 관계 선언, 데이터의 무결성을 보장

-- 기준키 테이블, 외래 키 테이블

-- 외래키 테이블에 데이터를 입력 시, 기준키 테이블에 데이터가 존재해야alter

-- 기준키 테이블의 참조 열은 반드시 unique or primary key이어야

```
DROP TABLE IF EXISTS buyTbl, userTbl;
```

```
CREATE TABLE userTbl
```

```
( userID char(8) NOT NULL PRIMARY KEY,
```

```
name varchar(10) NOT NULL,
```

```
birthYear int NOT NULL,
```

```
addr char(2) NOT NULL,
```

```
mobile1 char(3) NULL,
```

```
mobile2 char(8) NULL,
```

```
height smallint NULL,
```

```
mDate date NULL
```

```
);
```

-- 1

```

CREATE TABLE buyTbl
(
  num int AUTO_INCREMENT NOT NULL PRIMARY KEY ,

  userid char(8) NOT NULL ,

                                FOREIGN KEY(userid) REFERENCES userTbl(userID),

  prodName char(6) NOT NULL,

  groupName char(4) NULL ,

  price      int  NOT NULL,

  amount     smallint  NOT NULL

);

```

```

DROP TABLE IF EXISTS buyTbl;

```

```

-- 2

```

```

CREATE TABLE buyTbl
(
  num int AUTO_INCREMENT NOT NULL PRIMARY KEY,

  userid char(8) NOT NULL,

  prodName char(6) NOT NULL,

  groupName char(4) NULL ,

  price      int  NOT NULL,

  amount     smallint  NOT NULL,

  CONSTRAINT FK_userTbl_buyTbl FOREIGN KEY(userid) REFERENCES userTbl(userID)

);

```

```

-- 3

```

```

CREATE TABLE buyTbl
(
  num int AUTO_INCREMENT NOT NULL PRIMARY KEY,

  userid char(8) NOT NULL,

  prodName char(6) NOT NULL,

  groupName char(4) NULL ,

  price      int  NOT NULL,

```

```
amount    smallint NOT NULL,  
  
FOREIGN KEY(userid) REFERENCES userTbl(userID)  
  
);
```

```
DROP TABLE IF EXISTS buyTbl, userTbl ;
```

```
-- 4
```

```
CREATE TABLE userTbl  
  
( userID char(8) NOT NULL PRIMARY KEY,  
  
  name    nvarchar(10) NOT NULL,  
  
  birthYear int NOT NULL,  
  
  addr    char(2) NOT NULL,  
  
  mobile1 char(3) NULL,  
  
  mobile2 char(8) NULL,  
  
  height  smallint NULL,  
  
  mDate   date NULL  
  
);
```

```
CREATE TABLE buyTbl  
  
( num int AUTO_INCREMENT NOT NULL PRIMARY KEY,  
  
  userid char(8) NOT NULL,  
  
  prodName char(6) NOT NULL,  
  
  groupName char(4) NULL ,  
  
  price    int NOT NULL,  
  
  amount   smallint NOT NULL  
  
);
```

```
ALTER TABLE buyTbl  
  
  ADD CONSTRAINT FK_userTbl_buyTbl
```

```
FOREIGN KEY (userid)
```

```
REFERENCES userTbl(userID);
```

```
show index from buytbl;
```

```
show index from usertbl;
```

```
-- on delete cascade, on update cascade -- ₩
```

```
-- 기존 테이블의 데이터가 변경 시 외래키 테이블에도 자동 반영
```

```
ALTER TABLE buyTbl
```

```
    DROP FOREIGN KEY FK_userTbl_buyTbl; -- 외래 키 제거
```

```
ALTER TABLE buyTbl
```

```
    ADD CONSTRAINT FK_userTbl_buyTbl
```

```
    FOREIGN KEY (userID)
```

```
    REFERENCES userTbl (userID)
```

```
    ON UPDATE CASCADE;
```

```
-- Unique
```

```
CREATE TABLE userTbl
```

```
( userID  char(8) NOT NULL PRIMARY KEY,
```

```
  name    nvarchar(10) NOT NULL,
```

```
  birthYear  int NOT NULL,
```

```
    addr      char(2) NOT NULL,

    mobile1 char(3) NULL,

    mobile2   char(8) NULL,

    height    smallint NULL,

    mDate     date NULL,

    email     char(30) NULL  UNIQUE

);
```

```
CREATE TABLE userTbl

( userID  char(8) NOT NULL PRIMARY KEY,

    name   nvarchar(10) NOT NULL,

    birthYear  int NOT NULL,

    addr     char(2) NOT NULL,

    mobile1 char(3) NULL,

    mobile2   char(8) NULL,

    height    smallint NULL,

    mDate     date NULL,

    email     char(30) NULL ,

    CONSTRAINT AK_email  UNIQUE (email)

);
```

```
-- Defualt
```

```
drop database testdb;
```

```
CREATE DATABASE IF NOT EXISTS testDB;
```

```
use testDB;
```

```
DROP TABLE IF EXISTS userTbl;
```

```
-- 1
```

```
CREATE TABLE userTbl
```

```
( userID    char(8) NOT NULL PRIMARY KEY,  
  
  name      varchar(10) NOT NULL,  
  
  birthYear  int NOT NULL DEFAULT -1,  
  
  addr      char(2) NOT NULL DEFAULT '서울',  
  
  mobile1   char(3) NULL,  
  
  mobile2   char(8) NULL,  
  
  height    smallint NULL DEFAULT 170,  
  
  mDate     date NULL  
);
```

```
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      char(2) NOT NULL,  
  
  mobile1   char(3) NULL,  
  
  mobile2   char(8) NULL,  
  
  height    smallint NULL,  
  
  mDate     date NULL  
);
```

-- 2

ALTER TABLE userTbl

ALTER COLUMN birthYear SET DEFAULT -1;

ALTER TABLE userTbl

ALTER COLUMN addr SET DEFAULT '서울';

ALTER TABLE userTbl

ALTER COLUMN height SET DEFAULT 170;

-- default 문은 DEFAULT로 설정된 값을 자동 입력한다.

INSERT INTO userTbl VALUES ('LHL', '이혜리', default, default, '011', '1234567', default, '2019.12.12');

-- 열이름이 명시되지 않으면 DEFAULT로 설정된 값을 자동 입력한다

INSERT INTO userTbl(userID, name) VALUES('KAY', '김아영');

-- 값이 직접 명기되면 DEFAULT로 설정된 값은 무시된다.

INSERT INTO userTbl VALUES ('WB', '원빈', 1982, '대전', '019', '9876543', 176, '2017.5.5');

SELECT \* FROM userTbl;

-- <데이터 압축> --

-- 시스템변수 확인

SHOW VARIABLES LIKE 'innodb\_file\_format';

SHOW VARIABLES LIKE 'innodb\_large\_prefix';

```
CREATE DATABASE IF NOT EXISTS compressDB;

USE compressDB;

CREATE TABLE normalTBL( emp_no int , first_name varchar(14));

CREATE TABLE compressTBL( emp_no int , first_name varchar(14))

    ROW_FORMAT=COMPRESSED ;
```

```
INSERT INTO normalTbl

    SELECT emp_no, first_name FROM employees.employees;

INSERT INTO compressTBL

    SELECT emp_no, first_name FROM employees.employees;
```

```
SHOW TABLE STATUS FROM compressDB;
```

```
DROP DATABASE IF EXISTS compressDB;
```

```
-- 임시 테이블 - 잠시 사용하는 테이블

-- 세션 내에서만 사용, 생성한 클라이언트만 사용 가능

-- 임시테이블 삭제 - drop table, workbench 종료, mysql서비스 재시작
```



```
USE employees;
```

```
CREATE TEMPORARY TABLE IF NOT EXISTS tempTBL (id INT, name CHAR(5));
```

```
CREATE TEMPORARY TABLE IF NOT EXISTS employees (id INT, name CHAR(5));
```

```
DESCRIBE tempTBL;
```

```
DESCRIBE employees;
```

```
INSERT INTO tempTBL VALUES (1, 'This');
```

```
INSERT INTO employees VALUES (2, 'MySQL');
```

```
SELECT * FROM tempTBL;
```

```
SELECT * FROM employees;
```

```
USE employees;
```

```
SELECT * FROM tempTBL;
```

```
SELECT * FROM employees;
```

```
USE employees;
```

```
SELECT * FROM employees;
```

```
-- 테이블 삭제
```

```
-- drop table 테이블 이름
```

```
-- 외래키 제약 조건의 기준 테이블은 삭제할 수 없다
```

-- 먼저 외래키 테이블을 삭제해야 한다.

-- buytbl을 먼저 삭제 후 usertbl을 삭제해야

-- 테이블 수정--

-- cf. insert, delete, update

-- alter table table\_name add column

-- alter table table\_name change column

-- alter table table\_name drop column/ primary key/ foreign key

USE tableDB;

ALTER TABLE userTbl

ADD homepage VARCHAR(30) -- 열추가

DEFAULT 'http://www.hanbit.co.kr' -- 디폴트값

NULL; -- Null 허용함

select \* from usertbl;

ALTER TABLE userTbl

DROP COLUMN mobile1;

select \* from usertbl;

ALTER TABLE userTbl

CHANGE COLUMN name uName VARCHAR(20) NULL ;

```
select * from usertbl;
```

```
show index from usertbl;
```

```
/*
```

```
ALTER TABLE userTbl
```

```
    ADD CONSTRAINT PK_userTbl_userID
```

```
    PRIMARY KEY (userID);
```

```
ALTER TABLE buyTbl
```

```
    ADD CONSTRAINT FK_userTbl_buyTbl
```

```
    FOREIGN KEY (userID)
```

```
    REFERENCES userTbl (userID)
```

```
*/
```

```
ALTER TABLE userTbl
```

```
    DROP PRIMARY KEY; -- error
```

```
show index from usertbl;
```

```
show index from buytbl;
```

```
ALTER TABLE buyTbl
```

```
    DROP FOREIGN KEY fk_usertbl_buytbl;
```

```
ALTER TABLE userTbl
```

```
    DROP PRIMARY KEY;
```

```
show index from usertbl;
```

```
create database tabledb;
```

```
USE tableDB;
```

```
DROP TABLE IF EXISTS buyTbl, userTbl;
```

```
CREATE TABLE userTbl
```

```
( userID   char(8),  
  
   name    nvarchar(10),  
  
   birthYear  int,  
  
   addr     nchar(2),  
  
   mobile1 char(3),  
  
   mobile2   char(8),  
  
   height    smallint,  
  
   mDate     date
```

```
);
```

```
CREATE TABLE buyTbl
```

```
(  num int AUTO_INCREMENT PRIMARY KEY,  
  
   userid   char(8),  
  
   prodName nchar(6),  
  
   groupName nchar(4),  
  
   price     int ,  
  
   amount    smallint
```

```
);
```

```
INSERT INTO userTbl VALUES('LSG', '이승기', 1987, '서울', '011', '1111111', 182, '2008-8-8');
```

```
INSERT INTO userTbl VALUES('KBS', '김범수', NULL, '경남', '011', '2222222', 173, '2012-4-4');
```

```
INSERT INTO userTbl VALUES('KKH', '김경호', 1871, '전남', '019', '3333333', 177, '2007-7-7');
```

```
INSERT INTO userTbl VALUES('JYP', '조용필', 1950, '경기', '011', '4444444', 166, '2009-4-4');
```

```
INSERT INTO buyTbl VALUES(NULL, 'KBS', '운동화', NULL, 30, 2);
```

```
INSERT INTO buyTbl VALUES(NULL, 'KBS', '노트북', '전자', 1000, 1);
```

```
INSERT INTO buyTbl VALUES(NULL, 'JYP', '모니터', '전자', 200, 1);
```

```
INSERT INTO buyTbl VALUES(NULL, 'BBK', '모니터', '전자', 200, 5);
```

```
select * from usertbl;
```

```
select * from buytbl;
```

```
ALTER TABLE userTbl
```

```
ADD CONSTRAINT PK_userTbl_userID
```

```
PRIMARY KEY (userID);
```

```
ALTER TABLE buyTbl
```

```
ADD CONSTRAINT FK_userTbl_buyTbl
```

```
FOREIGN KEY (userID)
```

```
REFERENCES userTbl (userID); -- error - BBK
```

```
DELETE FROM buyTbl WHERE userid = 'BBK';
```

```
ALTER TABLE buyTbl
```

```
ADD CONSTRAINT FK_userTbl_buyTbl
```

```
FOREIGN KEY (userID)
```

```
REFERENCES userTbl (userID);
```

```
INSERT INTO buyTbl VALUES(NULL, 'BBK', '모니터', '전자', 200, 5); -- 오류
```

```

SET foreign_key_checks = 0; -- 외래키조건 해제

INSERT INTO buyTbl VALUES(NULL, 'BBK', '모니터', '전자', 200, 5);

INSERT INTO buyTbl VALUES(NULL, 'KBS', '청바지', '의류', 50, 3);

INSERT INTO buyTbl VALUES(NULL, 'BBK', '메모리', '전자', 80, 10);

INSERT INTO buyTbl VALUES(NULL, 'SSK', '책', '서적', 15, 5);

INSERT INTO buyTbl VALUES(NULL, 'EJW', '책', '서적', 15, 2);

INSERT INTO buyTbl VALUES(NULL, 'EJW', '청바지', '의류', 50, 1);

INSERT INTO buyTbl VALUES(NULL, 'BBK', '운동화', NULL, 30, 2);

INSERT INTO buyTbl VALUES(NULL, 'EJW', '책', '서적', 15, 1);

INSERT INTO buyTbl VALUES(NULL, 'BBK', '운동화', NULL, 30, 2);

SET foreign_key_checks = 1; -- 외래키조건 재설정

```

-- check - mysql에서 지원하지 않는다

```
select * from usertbl;
```

```
ALTER TABLE userTbl
```

```
ADD CONSTRAINT CK_birthYear
```

```
CHECK (birthYear >= 1900 AND birthYear <= YEAR(CURDATE()));
```

```

INSERT INTO userTbl VALUES('SSK', '성시경', 1979, '서울', NULL, NULL, 186, '2013-12-12');

INSERT INTO userTbl VALUES('LJB', '임재범', 1963, '서울', '016', '6666666', 182, '2009-9-9');

INSERT INTO userTbl VALUES('YJS', '윤종신', 1969, '경남', NULL, NULL, 170, '2005-5-5');

INSERT INTO userTbl VALUES('EJW', '은지원', 1972, '경북', '011', '8888888', 174, '2014-3-3');

INSERT INTO userTbl VALUES('JKW', '조관우', 1965, '경기', '018', '9999999', 172, '2010-10-10');

INSERT INTO userTbl VALUES('BBK', '바비킴', 1973, '서울', '010', '0000000', 176, '2013-5-5');

```

```
select * from usertbl;
```

```
-- update
```

```
UPDATE userTbl SET userID = 'VVK' WHERE userID='BBK'; -- error
```

```
SET foreign_key_checks = 0;
```

```
UPDATE userTbl SET userID = 'VVK' WHERE userID='BBK';
```

```
SET foreign_key_checks = 1;
```

```
SELECT B.userid, U.name, B.prodName, U.addr, U.mobile1 + U.mobile2 AS '연락처' -- 4건 부족
```

```
FROM buyTbl B
```

```
INNER JOIN userTbl U
```

```
ON B.userid = U.userid ;
```

```
SELECT COUNT(*) FROM buyTbl;
```

```
select * from buytbl;
```

```
SELECT B.userid, U.name, B.prodName, U.addr, U.mobile1 + U.mobile2 AS '연락처'
```

```
FROM buyTbl B
```

```
LEFT OUTER JOIN userTbl U
```

```
ON B.userid = U.userid
```

```
ORDER BY B.userid ;
```

```
SET foreign_key_checks = 0;
```

```
UPDATE userTbl SET userID = 'BBK' WHERE userID='VVK';
```

```
SET foreign_key_checks = 1;
```

```
ALTER TABLE buyTbl
```

```
    DROP FOREIGN KEY FK_userTbl_buyTbl;
```

```
-- on update cascade
```

```
alter table usertbl
```

```
add constraint primary key (userid);
```

```
ALTER TABLE buyTbl
```

```
    ADD CONSTRAINT FK_userTbl_buyTbl
```

```
        FOREIGN KEY (userID)
```

```
        REFERENCES userTbl (userID)
```

```
        ON UPDATE CASCADE;
```

```
UPDATE userTbl SET userID = 'VVK' WHERE userID='BBK';
```

```
SELECT B.userid, U.name, B.prodName, U.addr, U.mobile1 + U.mobile2 AS '연락처' -- 합계 수정
```

```
FROM buyTbl B
```

```
    INNER JOIN userTbl U
```

```
        ON B.userid = U.userid
```

```
ORDER BY B.userid;
```

```
DELETE FROM userTbl WHERE userID = 'VVK'; -- 삭제 안 됨
```



```
ALTER TABLE buyTbl
```

```
    DROP FOREIGN KEY FK_userTbl_buyTbl;
```

```
ALTER TABLE buyTbl
```

```
    ADD CONSTRAINT FK_userTbl_buyTbl
```

```
        FOREIGN KEY (userID)
```

```
        REFERENCES userTbl (userID)
```

```
        ON UPDATE CASCADE
```

```
        ON DELETE CASCADE;
```

```
DELETE FROM userTbl WHERE userID = 'VVK'; -- 함께 삭제됨
```

```
SELECT * FROM buyTbl ;
```

```
ALTER TABLE userTbl
```

```
    DROP COLUMN birthYear ;
```

```
-- < view > --
```

```
--
```

```
USE tableDB;
```

```
CREATE VIEW v_userTbl
```

```
AS
```

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

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

```
/*
```

1. 보안에 도움
2. 복잡한 쿼리를 단순화

\*/

```
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';
```

/\*

```
CREATE DATABASE sqlDB;
```

```
USE sqlDB;
```

```
CREATE TABLE userTbl -- 회원 테이블
```

```
( userID    CHAR(8) NOT NULL PRIMARY KEY, -- 사용자아이디
```

```
  name      VARCHAR(10) NOT NULL, -- 이름
```

```
  birthYear INT NOT NULL, -- 출생년도
```

```
  addr      CHAR(2) NOT NULL, -- 지역(경기,서울,경남 식으로 2글자만입력)
```

```
  mobile1   CHAR(3), -- 휴대폰의 국번(011, 016, 017, 018, 019, 010 등)
```

```
  mobile2   CHAR(8), -- 휴대폰의 나머지 전화번호(하이픈제외)
```

```

height          SMALLINT, -- 키

mDate           DATE  -- 회원 가입일

);

CREATE TABLE buyTbl -- 회원 구매 테이블

( num           INT AUTO_INCREMENT NOT NULL PRIMARY KEY, -- 순번(PK)

  userID        CHAR(8) NOT NULL, -- 아이디(FK)

  prodName      CHAR(6) NOT NULL, -- 물품명

  groupName     CHAR(4) , -- 분류

  price         INT  NOT NULL, -- 단가

  amount        SMALLINT  NOT NULL, -- 수량

  FOREIGN KEY (userID) REFERENCES userTbl(userID)

);

INSERT INTO userTbl VALUES('LSG', '이승기', 1987, '서울', '011', '1111111', 182, '2008-8-8');

INSERT INTO userTbl VALUES('KBS', '김범수', 1979, '경남', '011', '2222222', 173, '2012-4-4');

INSERT INTO userTbl VALUES('KKH', '김경호', 1971, '전남', '019', '3333333', 177, '2007-7-7');

INSERT INTO userTbl VALUES('JYP', '조용필', 1950, '경기', '011', '4444444', 166, '2009-4-4');

INSERT INTO userTbl VALUES('SSK', '성시경', 1979, '서울', NULL , NULL      , 186, '2013-12-12');

INSERT INTO userTbl VALUES('LJB', '임재범', 1963, '서울', '016', '6666666', 182, '2009-9-9');

INSERT INTO userTbl VALUES('YJS', '윤종신', 1969, '경남', NULL , NULL      , 170, '2005-5-5');

INSERT INTO userTbl VALUES('EJW', '은지원', 1972, '경북', '011', '8888888', 174, '2014-3-3');

INSERT INTO userTbl VALUES('JKW', '조관우', 1965, '경기', '018', '9999999', 172, '2010-10-10');

INSERT INTO userTbl VALUES('BBK', '바비킴', 1973, '서울', '010', '0000000', 176, '2013-5-5');

INSERT INTO buyTbl VALUES(NULL, 'KBS', '운동화', NULL , 30, 2);

INSERT INTO buyTbl VALUES(NULL, 'KBS', '노트북', '전자', 1000, 1);

INSERT INTO buyTbl VALUES(NULL, 'JYP', '모니터', '전자', 200, 1);

INSERT INTO buyTbl VALUES(NULL, 'BBK', '모니터', '전자', 200, 5);

INSERT INTO buyTbl VALUES(NULL, 'KBS', '청바지', '의류', 50, 3);

INSERT INTO buyTbl VALUES(NULL, 'BBK', '메모리', '전자', 80, 10);

INSERT INTO buyTbl VALUES(NULL, 'SSK', '책'      , '서적', 15, 5);

```

```

INSERT INTO buyTbl VALUES(NULL, 'EJW', '책'      , '서적', 15,  2);

INSERT INTO buyTbl VALUES(NULL, 'EJW', '청바지', '의류', 50,  1);

INSERT INTO buyTbl VALUES(NULL, 'BBK', '운동화', NULL   , 30,  2);

INSERT INTO buyTbl VALUES(NULL, 'EJW', '책'      , '서적', 15,  1);

INSERT INTO buyTbl VALUES(NULL, 'BBK', '운동화', NULL   , 30,  2);

*/

```

```

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.` `

```

```

# SELECT 'USER ID', 'USER NAME' FROM v_userbuyTbl;

```

```

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;

```

```
DROP VIEW v_userbuyTbl;
```

```
/*
```

view를 사용하는 이유

1. 보안에 도움이 된다.
2. 복잡한 쿼리를 단순화한다.

```
*/
```

```
USE sqlDB;
```

```
CREATE OR REPLACE VIEW v_userTbl
```

```
AS
```

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

```
DESCRIBE v_userTbl;
```

```
# SHOW CREATE VIEW v_userTbl;
```

```
UPDATE v_userTbl SET addr = '부산' WHERE userid='JKW' ;
```

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

```
CREATE VIEW v_sum
```

```
AS
```

```
    SELECT userid AS 'userid', SUM(price*amount) AS 'total'
```

```
    FROM buyTbl GROUP BY userid;
```

```
SELECT * FROM v_sum;
```

```
SELECT * FROM INFORMATION_SCHEMA.VIEWS      -- 시스템에 저장된 모든 뷰
```

```
    WHERE TABLE_SCHEMA = 'sqlDB' AND TABLE_NAME = 'v_sum';
```

```
CREATE VIEW v_height177
```

```
AS
```

```
SELECT * FROM userTbl WHERE height >= 177 ;
```

```
SELECT * FROM v_height177 ;
```

```
DELETE FROM v_height177 WHERE height < 177 ;
```

```
INSERT INTO v_height177 VALUES('KBM', '김병만', 1977 , '경기', '010', '5555555', 158, '2019-01-01') ;
```

```
INSERT INTO v_height177 VALUES('KBM', '김병만', 1977 , '경기', '010', '5555555', 158, '2019-01-01') ; -- 뷰에는 보이지 않지만 입력된다
```

```
select * from usertbl;
```

```
ALTER VIEW v_height177
```

```
AS
```

```
SELECT * FROM userTbl WHERE height >= 177
```

```
WITH CHECK OPTION ; -- 입력차단
```

```
INSERT INTO v_height177 VALUES('WDT', '서장훈', 2006 , '서울', '010', '3333333', 155, '2019-3-3') ;
```

```
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 ;
```

```
INSERT INTO v_userbuyTbl VALUES('PKL','박경리','운동화','경기','00000000000','2020-2-2'); -- 두 개 이상의 테이블이 연결된 뷰는 업데이트할 수 없다
```

```
DROP TABLE IF EXISTS buyTbl, userTbl;
```

```
SELECT * FROM v_userbuyTbl;
```

```
CHECK TABLE v_userbuyTbl; -- 뷰의 상태 체크
```