
테이블과 뷰

테이블 만들기

❖ SQL로 테이블 만들기

```
DROP DATABASE IF EXISTS tableDB;  
CREATE DATABASE tableDB;
```

```
USE tableDB;
```

```
DROP TABLE IF EXISTS buyTBL, userTBL;  
CREATE TABLE userTBL -- 회원 테이블  
( userID      char(8),      -- 사용자 아이디  
  name        varchar(10), -- 이름  
  birthYear   int,          -- 출생연도  
  addr        char(2),      -- 지역(경기,서울,경남 등으로 글자만 입력)  
  mobile1     char(3),      -- 휴대폰의국번(011, 016, 017, 018, 019, 010 등)  
  mobile2     char(8),      -- 휴대폰의 나머지 전화번호(하이픈 제외)  
  height      smallint,    -- 키  
  mDate       date         -- 회원 가입일  
);
```

테이블 만들기

❖ SQL로 테이블 만들기

```
CREATE TABLE buyTBL -- 구매 테이블
(
  num          int,          -- 순번(PK)
  userid       char(8),      -- 아이디(FK)
  prodName     char(6),      -- 물품명
  groupName    char(4),      -- 분류
  price        int ,         -- 단가
  amount       smallint      -- 수량
);
```

테이블 만들기

❖ SQL로 테이블 만들기(NULL, NOT NULL 제약 조건)

```
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
);
```

테이블 만들기

❖ SQL로 테이블 만들기 (NULL, NOT NULL 제약 조건)

```
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
);
```

테이블 만들기

❖ SQL로 테이블 만들기(PRIMARY KEY 제약 조건)

- PRIMARY KEY
 - 중복을 허용하지 않으며, NOT NULL 임

```
USE tableDB;
DROP TABLE IF EXISTS buyTBL, userTBL;
CREATE TABLE userTBL
( userID      char(8) 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
);
```

테이블 만들기

❖ SQL로 테이블 만들기 (NULL, NOT NULL 제약 조건)

```
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  
);
```

테이블 만들기

❖ SQL로 테이블 만들기(AUTO_INCREMENT)

```
USE tableDB;
DROP TABLE IF EXISTS buyTBL, userTBL;
CREATE TABLE userTBL(
    userID      char(8) 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
);
```


테이블 만들기

❖ SQL로 테이블 만들기(AUTO_INCREMENT)

- AUTO_INCREMENT
 - 값을 지정하지 않은경우 자동 증가값이 지정

```
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  
);
```

테이블 만들기

❖ SQL로 테이블 만들기(FOREIN KEY)

- FOREIGN KEY
 - 다른 테이블에 대한 참조를 의미
 - 참조되는 필드는 반드시 그 테이블에서 KEY여야 함(유일해야 함)
 - **FOREIGN KEY(필드명) REFERENCES 참조테이블명(참조테이블_필드명)**

```
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)
);
```

테이블 만들기

❖ SQL로 테이블 만들기

```
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 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 );
```

테이블 만들기

❖ 제약 조건

- 데이터의 무결성을 지키기 위한 제한된 조건을 의미
 - 값이 반드시 있어야 함
 - 값이 중복되면 안됨 등
- PRIMARY KEY 제약조건
- FOREIGN KEY 제약조건
- UNIQUE 제약조건
- CHECK 제약조건
- DEFAULT 정의
- NULL 값 허용
- CONSTRAINT로 제약조건에 명칭 부여 가능
 - 에러 발생시 어떤 제약조건을 위반했는지 디버깅 하기 쉬움
 - **CONSTRAINT 제약조건 제약조건_명칭(컬럼명)**
 - **제약 조건 명칭 관례: 제약조건타입_테이블명_컬럼명**

테이블 만들기

❖ 기본키(PRIMARY KEY) 제약조건

- 테이블당 1개의 컬럼에만 배정
- 값은 중복 될 수 없으며, NULL을 가질 수 없음

```
DROP TABLE IF EXISTS userTBL;  
CREATE TABLE userTBL  
(  
    userID CHAR(8) NOT NULL,  
    name VARCHAR(10) NOT NULL,  
    birthYear INT NOT NULL,  
    CONSTRAINT PRIMARY KEY PK_userTBL_userID (userID)  
);
```

cf) PRIMARY KEY는 테이블당 1개만 존재하므로 일반적으로 제약조건명을 지정하지 않음

테이블 만들기

❖ 제약조건

- 테이블 생성 이후 제약조건 추가

[형식]

```
ALTER TABLE 테이블명  
ADD CONSTRAINT [제약조건명] PRIMARY KEY(컬럼명)
```

```
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 PRIMARY KEY (prodCode, prodID); -- 복합키
```

테이블 만들기

❖ 기본키(PRIMARY KEY) 제약조건

- 자동으로 인덱스 객체가 생성됨

```
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)  
);
```

```
SHOW INDEX FROM prodTbl ;
```

테이블 만들기

❖ 외래 키(FOREIGN KEY) 제약조건

- 두 테이블 사이의 관계를 선언함으로써, 데이터의 무결성을 보장
- 하나의 테이블(자식 테이블)이 다른 테이블(부모 테이블)에 의존하게 됨
- 자식 테이블에 외래 키가 존재
 - 참조하는 부모 테이블에 반드시 값이 존재해야 함
 - 삽입/수정 시 부모 테이블에 값이 없다면 에러
 - 자식 테이블에 참조하는 행이 있다면 해당 부모 테이블의 행은 삭제 불가
 - 삭제 시 에러 발생
 - 자식 테이블의 해당 값을 NULL로 바꾸거나, 먼저 자식 테이블의 행을 삭제

테이블 만들기

❖ 외래 키(FOREIGN KEY) 제약조건

```
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
);

CREATE TABLE buyTBL
(  num INT AUTO_INCREMENT NOT NULL PRIMARY KEY ,
   userID CHAR(8) NOT NULL,
   prodName CHAR(6) NOT NULL,
   FOREIGN KEY(userID) REFERENCES userTBL(userID)
);
```

```
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,
   CONSTRAINT FK_userTBL_buyTBL FOREIGN KEY(userID) REFERENCES userTBL(userID)
);
```

테이블 만들기

❖ 외래 키(FOREIGN KEY) 제약조건

```
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  
);  
  
ALTER TABLE buyTBL  
  ADD CONSTRAINT FK_userTBL_buyTBL  
  FOREIGN KEY (userID) REFERENCES userTBL(userID);  
  
SHOW INDEX FROM buyTBL ;
```

테이블 만들기

❖ 외래 키(FOREIGN KEY) 제약조건

- 부모 테이블의 행을 수정/삭제할 때
 - 그 행을 참조하는 자식 테이블에 대한 처리 옵션 지정
 - ON DELETE CASCADE
 - 부모 행이 삭제될 때 자식 테이블의 행도 자동 삭제
 - ON UPDATE CASCADE
 - 부모 수정 시 자식도 해당 값으로 자동 수정
 - 디폴트
 - ON UPDATE NO ACTION
 - ON DELETE NO ACTION

테이블 만들기

❖ 외래 키(FOREIGN KEY) 제약조건

```
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;
```

테이블 만들기

❖ UNIQUE 제약조건

- 중복을 허용하지 않음
- NULL은 허용

cf) 키의 종류

- primary key
- unique key
- foreign key

테이블 만들기

❖ UNIQUE 제약조건

```
USE tableDB;
```

```
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,  
  email CHAR(30) NULL UNIQUE  
);
```

```
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,  
  email CHAR(30) NULL ,  
  CONSTRAINT AK_email UNIQUE (email)  
);
```

테이블 만들기

❖ CHECK 제약조건

- 입력되는 데이터를 점검하는 기능
 - 특정 값 중 하나인지, 음수인지, 출생연도가 1900년 이전인지 등 조사

```
-- 출생연도가 1900년 이후 그리고 2020년 이전, 이름은 반드시 넣어야 함.
```

```
DROP TABLE IF EXISTS userTBL;
```

```
CREATE TABLE userTBL
```

```
( userID      CHAR(8) PRIMARY KEY,
```

```
  name        VARCHAR(10) ,
```

```
  birthYear   INT CHECK (birthYear >= 1900 AND birthYear <= 2020),
```

```
  mobile1     CHAR(3) NULL,
```

```
  CONSTRAINT CK_name CHECK ( name IS NOT NULL)
```

```
);
```

```
-- 휴대폰 국번 체크
```

```
ALTER TABLE userTbl
```

```
  ADD CONSTRAINT CK_mobile1
```

```
  CHECK (mobile1 IN ('010','011','016','017','018','019')) ;
```

테이블 만들기

❖ DEFAULT 정의

- 값을 입력하지 않았을 때, 자동으로 입력되는 기본 값을 정의

```
DROP TABLE IF EXISTS userTBL;
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
);
```


테이블 만들기

❖ DEFAULT 정의

```
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
);
```

```
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;
```

테이블 만들기

❖ DEFAULT 정의

-- default 문은 DEFAULT로 설정된 값을 자동 입력한다.

```
INSERT INTO userTBL VALUES
```

```
('LHL', '이혜리', default, default, '011', '1234567', default, '2022.12.12');
```

-- 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, '2023.5.5');
```

```
SELECT * FROM userTBL;
```

테이블 만들기

❖ 임시 테이블

- 임시로 잠깐 사용되는 테이블
- 세션 내에서만 존재
 - 세션: 로그인 해서 로그아웃 될 때까지의 기간
 - 로그아웃 하면 임시 테이블은 자동으로 삭제

[형식]

```
CREATE TEMPORARY TABLE [IF NOT EXISTS] 테이블이름  
(열 정의 ...)
```

테이블 만들기

❖ 임시 테이블

```
USE employees;
CREATE TEMPORARY TABLE IF NOT EXISTS tempTBL (id INT, name CHAR(7));
CREATE TEMPORARY TABLE IF NOT EXISTS employees (id INT, name CHAR(7));
DESCRIBE tempTBL;
DESCRIBE employees;

INSERT INTO tempTBL VALUES (1, 'This');
INSERT INTO employees VALUES (2, 'MariaDB');
SELECT * FROM tempTBL;
SELECT * FROM employees;

-- 로그아웃 후 다시 로그인
USE employees;
SELECT * FROM tempTBL;
SELECT * FROM employees;
```

테이블 삭제

❖ 테이블 삭제

[형식]

DROP TABLE 테이블명

외래키 제약조건이 있는 경우 자식 테이블을 먼저 삭제하거나, 제약조건 제거 후 실행해야 함

테이블 수정

❖ 테이블 수정

[형식]

```
ALTER TABLE 테이블명  
수정사항...
```

테이블 수정

❖ 테이블 수정

○ 열의 추가

```
USE tableDB;  
ALTER TABLE userTBL  
  ADD homepage VARCHAR(30)  -- 열추가  
    DEFAULT 'http://www.hanbit.co.kr' -- 디폴트값  
    NULL; -- Null 허용함
```

○ 열의 삭제

```
ALTER TABLE userTBL  
  DROP COLUMN mobile1;
```

○ 열의 수정

```
ALTER TABLE userTBL  
  CHANGE COLUMN name uName VARCHAR(20) NULL ;
```

테이블 수정

❖ 테이블 수정

- 제약조건의 삭제

```
ALTER TABLE userTBL
```

```
DROP PRIMARY KEY; -- 외래키 제약조건이 있으면 에러 발생
```

```
ALTER TABLE buyTBL
```

```
DROP FOREIGN KEY FK_userTBL_buyTBL;
```