

07

CHAPTER

테이블과 뷰



[실습 7-1] SQL 문으로 테이블 생성하기

1 테이블 생성하기

1-1 회원 테이블과 구매 테이블 생성(테이블의 기본적인 틀만 구성)

```
USE tableDB;
DROP TABLE IF EXISTS buyTBL, userTBL;
CREATE TABLE userTBL -- 회원 테이블
( userID CHAR(8), -- 사용자 아이디
  userName VARCHAR(10), -- 이름
  birthYear INT, -- 출생 연도
  addr CHAR(2), -- 지역(경기, 서울, 경남 등으로 글자만 입력)
  mobile1 CHAR(3), -- 휴대폰의 국번(011, 016, 017, 018, 019, 010 등)
  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 CHAR(6), -- 물품
  groupName CHAR(4), -- 분류
  price INT, -- 단가
  amount SMALLINT -- 수량
);
```

[실습 7-1] SQL 문으로 테이블 생성하기

1-2 몇 가지 옵션 추가

```
USE tableDB;
DROP TABLE IF EXISTS buyTBL, userTBL;
CREATE TABLE userTBL
( userID CHAR(8) NOT NULL,
  userName 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
);
```

[실습 7-1] SQL 문으로 테이블 생성하기

1-3 각 테이블에 기본키 설정

```
DROP TABLE IF EXISTS buyTBL, userTBL;
CREATE TABLE userTBL
( userID CHAR(8) NOT NULL PRIMARY KEY,
  userName 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
);
```

[실습 7-1] SQL 문으로 테이블 생성하기

1-4 구매 테이블(buyTBL)의 순번(num) 열에 AUTO_INCREMENT 설정

```
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  
);
```

1-5 구매 테이블(buyTBL) 아이디(userID) 열을 회원 테이블(userTBL) 아이디(userID) 열의 외래키로 설정

```
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)  
);
```

[실습 7-1] SQL 문으로 테이블 생성하기

2 테이블에 데이터 입력하기

2-1 회원 테이블에 데이터 3건 입력

```
INSERT INTO userTBL VALUES ('YJS', '유재석', 1972, '서울', '010', '11111111', 178, '2008-8-8');  
INSERT INTO userTBL VALUES ('KHD', '강호동', 1970, '경북', '011', '22222222', 182, '2007-7-7');  
INSERT INTO userTBL VALUES ('KKJ', '김국진', 1965, '서울', '019', '33333333', 171, '2009-9-9');
```

2-2 구매 테이블에 데이터 3건 입력

```
INSERT INTO buyTBL VALUES (NULL, 'KHD', '운동화', NULL, 30, 2);  
INSERT INTO buyTBL VALUES (NULL, 'KHD', '노트북', '전자', 1000, 1);  
INSERT INTO buyTBL VALUES (NULL, 'KYM', '모니터', '전자', 200, 1);
```

실행 결과

Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ...

[실습 7-1] SQL 문으로 테이블 생성하기

2-3 회원 테이블(userTBL)에 나머지 데이터를 입력한 후 구매 테이블(buyTBL)의 세 번째 데이터 부터 다시 입력

```
INSERT INTO userTBL VALUES ('KYM', '김용만', 1967, '서울', '010', '44444444', 177, '2015-5-5');
INSERT INTO userTBL VALUES ('KJD', '김제동', 1974, '경남', NULL, NULL, 173, '2013-3-3');
INSERT INTO userTBL VALUES ('NHS', '남희석', 1971, '충남', '016', '66666666', 180, '2017-4-4');
INSERT INTO userTBL VALUES ('SDY', '신동엽', 1971, '경기', NULL, NULL, 176, '2008-10-10');
INSERT INTO userTBL VALUES ('LHJ', '이휘재', 1972, '경기', '011', '88888888', 180, '2006-4-4');
INSERT INTO userTBL VALUES ('LKK', '이경규', 1960, '경남', '018', '99999999', 170, '2004-12-12');
INSERT INTO userTBL VALUES ('PSH', '박수홍', 1970, '서울', '010', '00000000', 183, '2012-5-5');
```

```
INSERT INTO buyTBL VALUES (NULL, 'KYM', '모니터', '전자', 200, 1);
INSERT INTO buyTBL VALUES (NULL, 'PSH', '모니터', '전자', 200, 5);
INSERT INTO buyTBL VALUES (NULL, 'KHD', '청바지', '의류', 50, 3);
INSERT INTO buyTBL VALUES (NULL, 'PSH', '메모리', '전자', 80, 10);
INSERT INTO buyTBL VALUES (NULL, 'KJD', '책', '서적', 15, 5);
INSERT INTO buyTBL VALUES (NULL, 'LHJ', '책', '서적', 15, 2);
INSERT INTO buyTBL VALUES (NULL, 'LHJ', '청바지', '의류', 50, 1);
INSERT INTO buyTBL VALUES (NULL, 'PSH', '운동화', NULL, 30, 2);
INSERT INTO buyTBL VALUES (NULL, 'LHJ', '책', '서적', 15, 1);
INSERT INTO buyTBL VALUES (NULL, 'PSH', '운동화', NULL, 30, 2);
```

2-1 제약 조건의 개요

- 제약 조건(constraint)
 - 데이터의 무결성을 지키기 위해 제한하는 조건
 - 특정 데이터를 입력할 때 무조건 입력되는 것이 아니라 제약 조건을 만족했을 때만 입력되도록 설정하는 것
- 대부분의 DBMS에서는 데이터의 무결성을 보장하기 위해 다음과 같은 제약 조건을 제공
 - 기본키 제약 조건
 - 외래키 제약 조건
 - UNIQUE 제약 조건
 - DEFAULT 제약 조건
 - NULL 값 허용

2-2 기본키 제약 조건

■ 기본키 제약조건

- 기본키에 입력되는 값은 중복될 수 없으며 NULL 값이 올 수도 있음
- 회원 테이블에 입력되는 회원 아이디는 당연히 중복될 수 없고 비어 있을 수도 없음

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

■ 기본키에 설정된 제약 조건의 이름을 DESCRIBE 문으로 확인

```
DESCRIBE userTBL;
```

	Field	Type	Null	Key	Default	Extra
▶	userID	char(8)	NO	PRI	NULL	
	userName	varchar(10)	NO		NULL	
	birthYear	int(11)	NO		NULL	

2-2 기본키 제약 조건

- 이미 만들어진 테이블을 수정하는 ALTER TABLE 문을 사용하여 제약 조건 설정

```
DROP TABLE IF EXISTS userTBL;  
CREATE TABLE userTBL  
( userID CHAR(8) NOT NULL,  
  userName VARCHAR(10) NOT NULL,  
  birthYear INT NOT NULL  
);  
ALTER TABLE userTBL  
ADD PRIMARY KEY (userID);
```

- ALTER TABLE userTBL: 회원 테이블(userTBL)을 변경
- ADD PRIMARY KEY (userID): 추가할 제약 조건은 기본키 제약 조건이고 제약 조건을 설정할 열은 userID

2-3 외래키 제약 조건

■ 외래키 제약 조건

- 두 테이블 사이의 관계를 선언함으로써 데이터의 무결성을 보장하는 역할을 함
- 테이블 사이에 외래키 관계를 설정하면 하나의 테이블이 다른 테이블에 의존하게 됨
- 외래키를 정의하는 테이블을 '외래키 테이블'이라 하고, 외래키에 의해서 참조가 되는 테이블을 '기준 테이블'이라 함
- 외래키 테이블에 데이터를 입력할 때는 기준 테이블을 참조
- 외래키 테이블이 참조하는 기준 테이블의 열은 반드시 기본키(PK)이거나 UNIQUE 제약 조건으로 설정되어 있어야 함

2-3 외래키 제약 조건

■ 외래키 설정 방법

- 외래키는 CREATE TABLE 문 내부에 FOREIGN KEY 키워드를 이용하여 설정

```
DROP TABLE IF EXISTS buyTBL, userTBL;
CREATE TABLE userTBL
( userID CHAR(8) NOT NULL PRIMARY KEY,
  userName 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)
);
```

■ 외래키 제약 조건의 이름 지정

- 마지막 행에서 쉼표로 분리한 후 아래에 작성

```
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)
);
```

2-3 외래키 제약 조건

- ALTER TABLE문을 이용하여 외래키 제약 조건 지정

```
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);
```

- ALTER TABLE buyTBL: 구매 테이블(buyTBL)을 수정
- ADD CONSTRAINT FK_userTBL_buyTBL: 제약 조건을 추가하고 제약 조건 이름을 ' FK_userTBL_buyTBL'로 명명
- FOREIGN KEY (userID): 외래키 제약 조건을 구매 테이블(buyTBL)의 아이디(userID) 열에 설정
- REFERENCES userTBL (userID): 참조할 기준 테이블은 회원 테이블(userTBL)의 아이디 (userID) 열

2-3 외래키 제약 조건

- SHOW INDEX FROM buyTBL; 문으로 확인 결과

	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible
▶	buytbl	0	PRIMARY	1	num	A	0	NULL	NULL		BTREE			YES
	buytbl	1	FK_userTbl_buyTbl	1	userID	A	0	NULL	NULL		BTREE			YES

- 외래키 설정시 ON DELETE CASCADE 또는 ON UPDATE CASCADE 옵션 사용

```
ALTER TABLE buyTBL
  DROP FOREIGN KEY FK_userTBL_buyTBL; -- 외래키 제거
ALTER TABLE buyTBL
  ADD CONSTRAINT FK_userTBL_buyTBL
  FOREIGN KEY (userID)
  REFERENCES userTBL (userID)
  ON UPDATE CASCADE ON DELETE CASCADE;
```

2-4 UNIQUE 제약 조건

- UNIQUE 제약 조건
 - 중복되지 않는 유일한 값을 입력해야 하는 조건
 - 기본키 제약 조건과 거의 비슷하지만 NULL 값은 허용
- 기존 회원 테이블에 email 열을 추가한 경우(두 CREATE TABLE 문은 동일한 결과를 출력)

```
USE tableDB;  
DROP TABLE IF EXISTS buyTBL, userTBL;
```

```
CREATE TABLE userTBL  
( userID CHAR(8) NOT NULL PRIMARY KEY,  
  userName VARCHAR(10) NOT NULL,  
  birthYear INT NOT NULL,  
  email CHAR(30) NULL UNIQUE  
);
```

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

```
CREATE TABLE userTBL  
( userID CHAR(8) NOT NULL PRIMARY KEY,  
  userName VARCHAR(10) NOT NULL,  
  birthYear INT NOT NULL,  
  email CHAR(30) NULL,  
  CONSTRAINT AK_email UNIQUE (email)  
);
```

2-5 DEFAULT 제약 조건

- DEFAULT 제약 조건

- 값을 입력하지 않았을 때 자동으로 입력되는 기본 값을 정의하는 조건

- 예

- 출생 연도를 입력하지 않았다면 '-1', 주소를 입력하지 않았다면 '서울', 키를 입력하지 않았다면 '170'을 자동으로 입력하는 구문

```
DROP TABLE IF EXISTS userTBL;
CREATE TABLE userTBL
( userID CHAR(8) NOT NULL PRIMARY KEY,
  userName 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
);
```


2-5 DEFAULT 제약 조건

- DEFAULT 제약 조건이 설정된 열에는 다음과 같은 방법으로 데이터 입력

-- 입력 데이터가 default이면 DEFAULT 문으로 설정된 값을 자동 입력한다.

```
INSERT INTO userTBL VALUES ('YBJ', '유병재', default, default, '010', '12345678', default, '2019.12.12');
```

-- 열 이름이 명시되지 않으면 DEFAULT 문으로 설정된 값을 자동 입력한다.

```
INSERT INTO userTBL (userID, userName) VALUES ('PNR', '박나래');
```

-- 값이 직접 명시되어 있으면 DEFAULT 문으로 설정된 값을 무시한다.

```
INSERT INTO userTBL VALUES ('WB', '원빈', 1982, '대전', '010', '98765432', 176, '2020.5.5');
```

```
SELECT * FROM userTBL;
```

[illegible]

3-1 테이블 삭제

- DROP TABLE 문
 - 테이블 삭제
- 테이블 삭제시 주의점
 - 외래키 제약 조건에 걸려 있는 기준 테이블은 삭제할 수 없음(외래키가 생성된 외래키 테이블을 삭제한 후 기준 테이블을 삭제해야 함)
 - 여러 개의 테이블을 동시에 삭제할 때는 DROP TABLE 테이블1, 테이블2, 테이블3;과 같이 계속 나열

3-2 테이블 수정

■ ALTER TABLE 문

- 이미 생성된 테이블의 구조에 무엇인가를 추가하거나 삭제하거나 변경

```
ALTER [IGNORE] TABLE tbl_name
    [alter_specification [, alter_specification] ...]
    [partition_options]

alter_specification:
    table_options
| ADD [COLUMN] col_name column_definition
    [FIRST | AFTER col_name]
| ADD [COLUMN] (col_name column_definition, ...)
| ADD {INDEX | KEY} [index_name]
    [index_type] (index_col_userName, ...) [index_option] ...
| ADD [CONSTRAINT [symbol]] PRIMARY KEY
    [index_type] (index_col_userName, ...) [index_option] ...
| ADD [CONSTRAINT [symbol]]
    UNIQUE [INDEX | KEY] [index_name]
    [index_type] (index_col_userName, ...) [index_option] ...
| ADD FULLTEXT [INDEX | KEY] [index_name]
    (index_col_userName, ...) [index_option]
| ADD [CONSTRAINT [symbol]]
    FOREIGN KEY [index_name] (index_col_userName, ...)
    reference_definition
| ALTER [COLUMN] col_name {SET DEFAULT literal | DROP DEFAULT}
| CHANGE [COLUMN] old_col_name new_col_name column_definition
    [FIRST | AFTER col_name]
| MODIFY [COLUMN] col_name column_definition
    [FIRST | AFTER col_name]
| DROP [COLUMN] col_name
| DROP PRIMARY KEY
| DROP {INDEX | KEY} index_name
| DROP FOREIGN KEY fk_symbol
| DISABLE KEYS
| ENABLE KEYS
| RENAME [TO | AS] new_tbl_name
| RENAME {INDEX | KEY} old_index_name TO new_index_name
| ORDER BY col_name [, col_name] ...
```

3-2 테이블 수정

- 회원 테이블(userTBL)에 회원의 홈페이지 주소 추가

```
USE tableDB;  
ALTER TABLE userTBL  
    ADD COLUMN homepage VARCHAR(30) -- 열 추가  
    DEFAULT 'http://www.hanbit.co.kr' -- 디폴트 값  
    NULL; -- NULL 허용
```

- 회원 테이블(userTBL)에서 전화번호 열 삭제

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

- 회원 테이블(userTBL)에서 회원 이름(name) 열의 이름을 uName으로, 데이터 형식을 VARCHAR(20)으로 변경하고 NULL 값도 허용

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

- 회원 테이블(userTBL)의 기본키 삭제

```
ALTER TABLE userTBL  
    DROP PRIMARY KEY;
```

3-2 테이블 수정

- 회원 테이블(userTBL)의 외래키 삭제

```
ALTER TABLE buyTBL  
  DROP FOREIGN KEY 외래키이름;
```

```
ALTER TABLE buyTBL  
  DROP CONSTRAINT 외래키이름;
```

```
set foreign_key_checks = 1;  
set foreign_key_checks = 0;
```

. ->

가 .