

▶ Chapter 08 : 제약조건, 테이블

SQL 활용 프로그래밍

SQL Application Programming



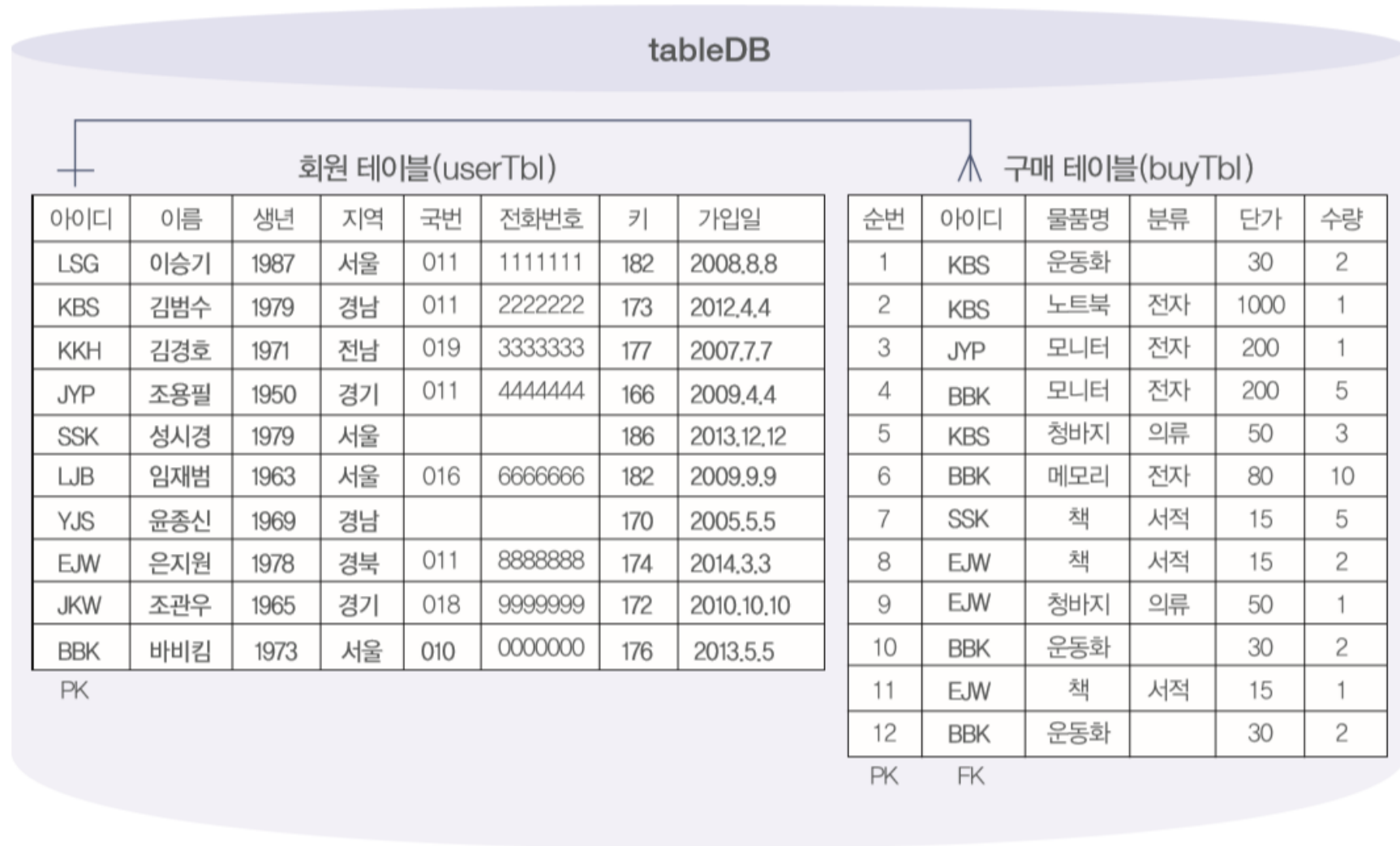


CHAPTER 08 제약조건, 테이블

데이터베이스의 핵심 개체인 테이블과 가상의 테이블인 뷰에 대해서도 알아본다.

SECTION 01 테이블

테이블 만들기



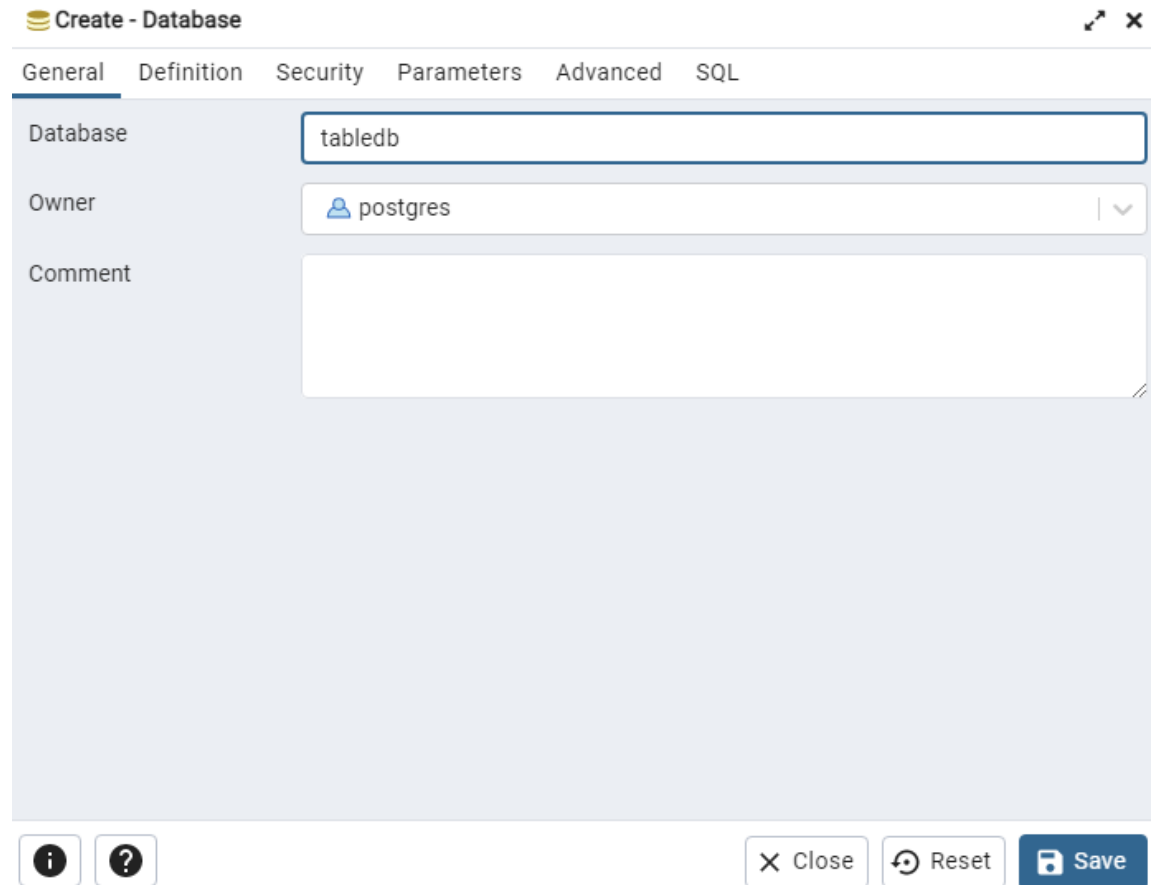
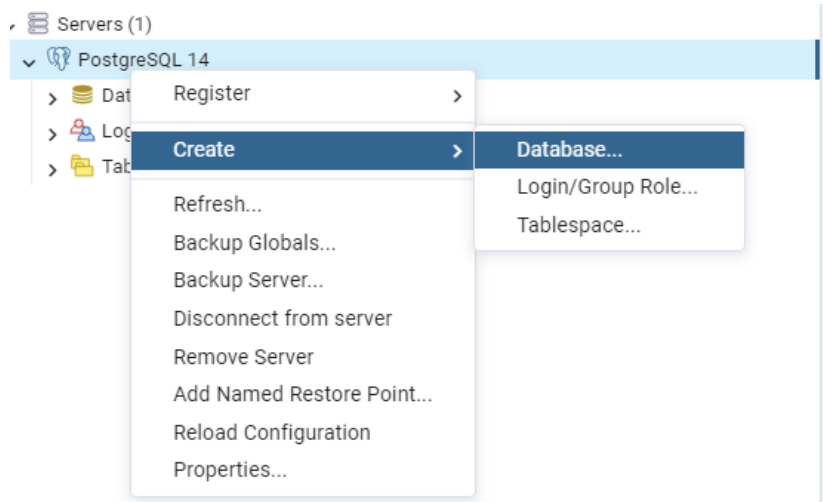
[그림 8-1] 샘플로 사용할 tabledb

[실습 1-1] PgAdmin 의 Navigator에서 테이블을 생성하고, 데이터를 입력하자.

테이블 만들기

- tabledb 생성

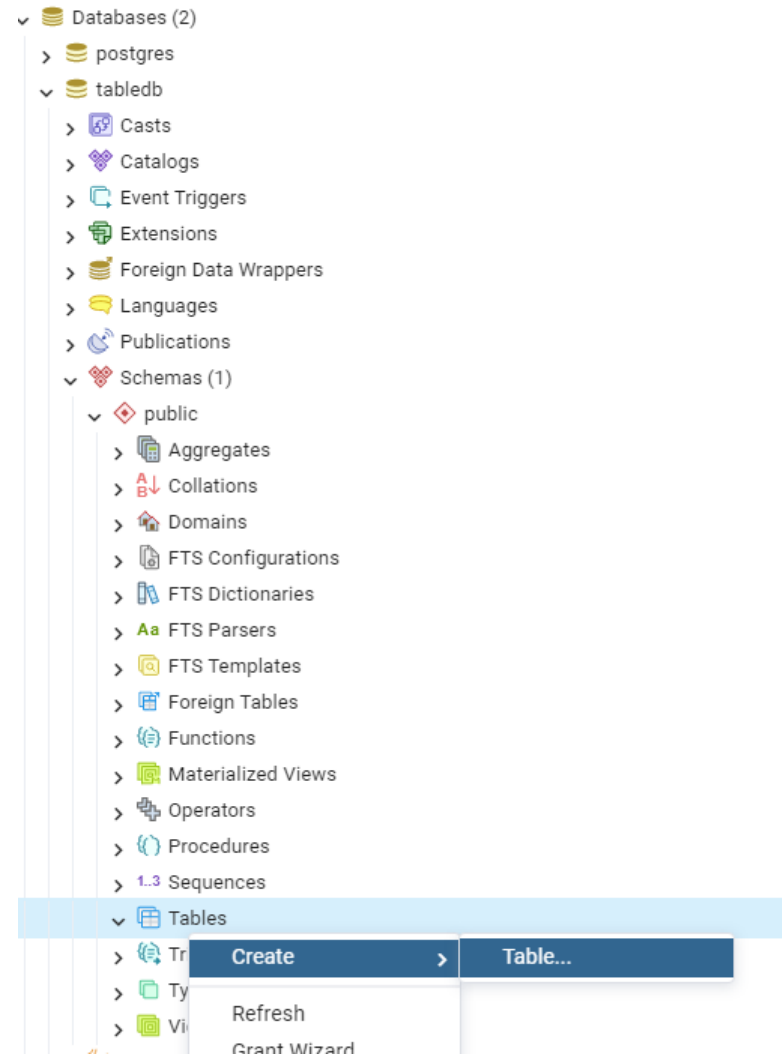
CREATE DATABASE tabledb;



[실습 1-1] PgAdmin 의 Navigator에서 테이블을 생성하고, 데이터를 입력하자.

테이블 만들기

- PgAdmin에서 테이블 생성
 'tabledb' 확장 -> Schemas -> public
 -> 'Tables' 마우스 오른쪽 버튼 -> [Create Table] 선택



[실습 1-1] PgAdmin 의 Navigator에서 테이블을 생성하고, 데이터를 입력하자.

- 테이블 만들기
- PgAdmin에서 테이블 생성
 - usertbl 생성
 - Columns로 이동하여 column 생성
 - userID열을 기본 키(Primary Key)로 설정

Create - Table

General Columns Advanced Constraints Partitions Parameters Security SQL

Name: usertbl

Owner: postgres

Schema: public

Tablespace: Select an item...

Partitioned table?: ☐

Comment:

Close Reset Save

General Columns Advanced Constraints Parameters Security SQL

Columns

| | Name | Data type | Length/Precision | Scale | Not NULL? | Primary key? | Default |
|--|-----------|-------------------|------------------|-------|-------------------------------------|-------------------------------------|---------|
| | userid | character | 8 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| | name | character varying | 10 | | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
| | birthyear | integer | | | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
| | addr | character | 2 | | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
| | mobile1 | character | 3 | | <input type="checkbox"/> | <input type="checkbox"/> | |
| | mobile2 | character | 8 | | <input type="checkbox"/> | <input type="checkbox"/> | |
| | height | smallint | | | <input type="checkbox"/> | <input type="checkbox"/> | |
| | mdate | date | | | <input type="checkbox"/> | <input type="checkbox"/> | |

[실습 1-1] PgAdmin 의 Navigator에서 테이블을 생성하고, 데이터를 입력하자.

- 테이블 만들기
- PgAdmin에서 테이블 생성
 - usertbl 생성
 - Columns로 이동하여 column 생성
 - userID열을 기본 키(Primary Key)로 설정

| Query | Query History |
|-------|---|
| 1 | CREATE TABLE usertbl(2 userID CHAR(8) primary key not null, 3 name varchar(10) not null, 4 birthYear integer not null, 5 addr char(2) not null, 6 mobile1 char(3), 7 mobile2 char(8), 8 height smallint, 9 mDate DATE 10); |

[실습 1-1]

PgAdmin의 Navigator 에서 테이블을 생성하고, 데이터를 입력하자.

테이블 만들기

- PgAdmin 에서 테이블 생성
 - buytbl 생성
 - num열을 기본 키(Primary Key)로 설정
 - num열이 자동으로 값이 증가하도록 Serial 타입으로 정의
 - num열에 PRIMARY KEY, NOT NULL

| Query | Query History |
|-------|--|
| 11 | |
| 12 | CREATE TABLE buytbl (|
| 13 | num SERIAL PRIMARY KEY NOT NULL , |
| 14 | userid CHAR(8) NOT NULL , |
| 15 | probName CHAR(8) NOT NULL , |
| 16 | groupName CHAR(8) , |
| 17 | price INTEGER NOT NULL , |
| 18 | amount INTEGER NOT NULL , |
| 19 |); |

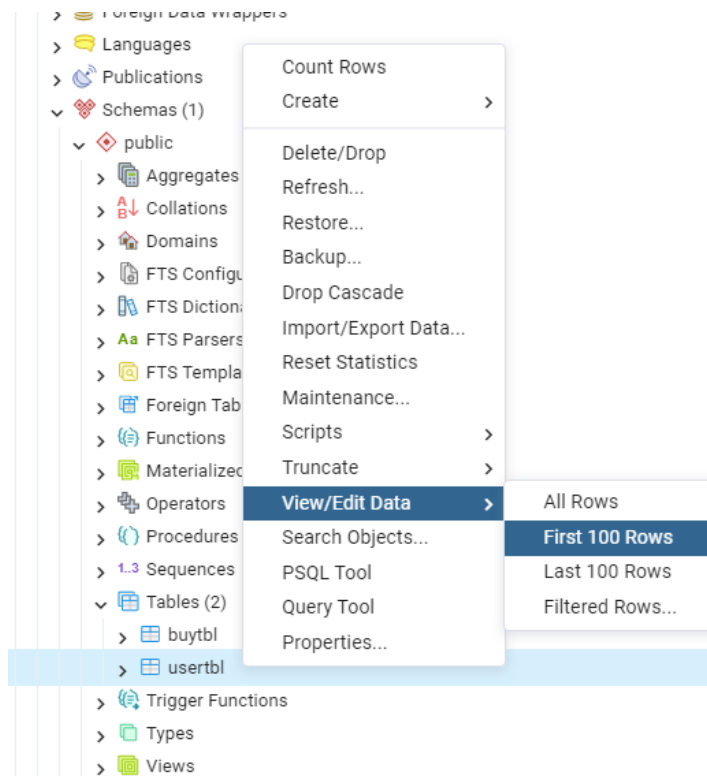
[실습 1-1]

PgAdmin의 Navigator 에서 테이블을 생성하고, 데이터를 입력하자.

테이블 만들기

○ PgAdmin 에서 데이터 입력

- Navigator에서 usertbl 선택 - 마우스 오른쪽 버튼 클릭 - [View/Edit Data-> First 100 Rows] 선택
- <Add row> 아이콘 클릭한 후, 3개 행 입력 - <Save Data Change> 클릭



| Data output Messages Notifications | | | | | | | | |
|------------------------------------|--------------------|------------------------|-----------|---------------|---------------|---------------|----------|--------|
| | userid | name | birthyear | addr | mobile1 | mobile2 | height | mdate |
| | [PK] character (8) | character varying (10) | integer | character (2) | character (3) | character (8) | smallint | date |
| 1+ | [null] | [null] | [null] | [null] | [null] | [null] | [null] | [null] |

| Data output Messages Notifications | | | | | | | | |
|------------------------------------|--------------------|------------------------|-----------|---------------|---------------|---------------|----------|------------|
| | userid | name | birthyear | addr | mobile1 | mobile2 | height | mdate |
| | [PK] character (8) | character varying (10) | integer | character (2) | character (3) | character (8) | smallint | date |
| 1+ | [null] | [null] | [null] | [null] | [null] | [null] | [null] | [null] |
| 2 | KKH | 김경호 | 1971 | 전남 | 019 | 33333333 | 177 | 2007-07-07 |
| 3 | LSG | 이승기 | 1987 | 서울 | 011 | 11111111 | 182 | 2008-08-08 |
| 4 | KBS | 김범수 | 1979 | 경남 | 011 | 22222222 | 173 | 2012-04-04 |

[실습 1-1]

PgAdmin의 Navigator 에서 테이블을 생성하고, 데이터를 입력하자.

테이블 만들기

- PgAdmin 에서 데이터 삭제
 - 해당 row를 지우고 싶을 경우 왼쪽 Index 번호를 클릭 후 Delete 클릭
 - <Save Data Change> 클릭

| Data output Messages Notifications | | | | | | | | |
|------------------------------------|---------------------------------------|--------------------------------|----------------------|-----------------------|--------------------------|--------------------------|--------------------|---------------|
| | userid [PK] character varying (10) | name character varying (10) | birthyear integer | addr character (2) | mobile1 character (3) | mobile2 character (8) | height smallint | mdate date |
| 1 | KKH | 김경호 | 1971 | 전남 | 019 | 33333333 | 177 | 2007-07-07 |
| 2 | LSG | 이승기 | 1987 | 서울 | 011 | 11111111 | 182 | 2008-08-08 |
| 3 | KBS | 김범수 | 1979 | 경남 | 011 | 22222222 | 173 | 2012-04-04 |

[실습 2]

SQL로 테이블을 생성하고, 데이터를 입력하자.

테이블 만들기

- SQL로 테이블 수정

- 회원 테이블 데이터 입력

- 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');

- 구매 테이블 데이터 입력

- INSERT INTO buytbl VALUES(1, 'KBS', '운동화', NULL, 30, 2);
 - INSERT INTO buytbl VALUES(2, 'KKH', '노트북', '전자', 1000, 1);

[실습 2]

SQL로 테이블을 생성하고, 데이터를 입력하자.

테이블 만들기

- SQL로 테이블 수정
 - 구매 테이블 데이터 삭제
 - DELETE FROM buytbl WHERE userid = 'KBS';
 - DELETE FROM buytbl WHERE userid = 'KBS';

SECTION 02 제약조건

제약 조건

- 제약 조건(Constraint)이란?
 - 데이터의 무결성을 지키기 위한 제한된 조건 의미
 - 특정 데이터를 입력 시 어떠한 조건을 만족했을 때에 입력되도록 제약
 - ex) 동일한 아이디로 다시 회원 가입이 안되는 것
 - 데이터 무결성을 위한 6가지 제약조건
 - PRIMARY KEY 제약 조건
 - FOREIGN KEY 제약 조건
 - UNIQUE 제약 조건
 - CHECK 제약 조건(MySQL 8.0.16부터 지원)
 - DEFAULT 정의
 - NULL 값 허용

SECTION 02 제약조건

데이터 무결성을 위한 제약 조건

◦ 기본 키(Primary Key) 제약 조건

- 기본 키(Primary Key) 란?
 - 테이블에 존재하는 많은 행의 데이터를 구분할 수 있는 식별자
 - 중복이나 NULL값이 입력될 수 없음
 - ex) 회원 테이블의 회원 아이디, 학생 테이블이 학번
- 기본 키로 생성한 것은 자동으로 클러스터형 인덱스 생성
- 테이블에서는 기본 키를 하나 이상 열에 설정 가능
- 기본 키 생성 방법

```
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)  
);
```

CONSTRAINT 생략 가능, PK가 userTBL 테이블의 userID 열에 지정됨

SECTION 02 제약조건

데이터 무결성을 위한 제약 조건

```
DROP TABLE IF EXISTS usertbl;  
CREATE TABLE usertbl  
(  userID  CHAR(8) NOT NULL,  
   name    VARCHAR(10) NOT NULL,  
   birthYear INT NOT NULL  
);  
ALTER TABLE usertbl  
  ADD CONSTRAINT PK_usertbl_userID  
  PRIMARY KEY (userID);
```

기본키는 각 테이블 별로 하나만 존재해야 하지만, 기본 키를 하나의 열로만 구성해야 하는 것은 아님
필요에 따라 두 개 또는 그 이상의 열을 합쳐서 하나의 기본키로 설정하는 경우도 종종 있음

- ALTER TABLE usertbl
usertbl을 변경하자.
- ADD CONSTRAINT PK_usertbl_userID
제약 조건을 추가하자. 추가할 제약 조건의 이름은 'PK_usertbl_userID'이다.
- PRIMARY KEY(userID)
추가할 제약 조건은 기본 키 제약 조건이다. 그리고 제약 조건을 설정할 열은 userID열이다.

SECTION 02 제약조건

데이터 무결성을 위한 제약 조건

- 기본 키(Primary Key) 제약 조건

- ex) 제품 테이블

- 기본 키 = 제품코드 + 제품일련번호

제품코드만으로는 중복 → 기본키 설정 불가

제품 일련번호도 0001부터 부여하는 체계로 기본 키 설정 불가

| 제품 코드 | 제품 일련 번호 | 제조일자 | 현 상태 |
|-------|----------|------------|------|
| AAA | 0001 | 2023.10.10 | 판매완료 |
| AAA | 0002 | 2023.10.11 | 매장진열 |
| BBB | 0001 | 2023.10.12 | 재고창고 |
| CCC | 0001 | 2023.10.13 | 판매완료 |
| CCC | 0002 | 2023.10.14 | 매장진열 |

SECTION 02 제약조건

데이터 무결성을 위한 제약 조건

- 기본 키(Primary Key) 제약 조건

```
CREATE TABLE prodTbl  
( prodCode CHAR(3) NOT NULL,  
  prodID CHAR(4) NOT NULL,  
  prodDate DATE NOT NULL,  
  prodCur CHAR(10) NULL,  
  CONSTRAINT PK_prodTbl_proCode_prodID  
  PRIMARY KEY (prodCode, prodID)  
);
```

| Query | Query History |
|-------|--|
| 1 | CREATE TABLE prodTbl |
| 2 | (prodCode CHAR(3) NOT NULL, |
| 3 | prodID CHAR(4) NOT NULL, |
| 4 | prodDate DATE NOT NULL, |
| 5 | prodCur CHAR(10) NULL |
| 6 |); |
| 7 | |
| 8 | ALTER TABLE prodTbl |
| 9 | ADD CONSTRAINT pk_prodTbl_proCode_prodID |
| 10 | PRIMARY KEY (prodCode, prodID); |

prodtbl

General Columns Advanced Constraints Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

| Name | | Columns |
|---------------------------|--|-----------------|
| pk_prodtbl_procode_prodid | | prodcode,prodid |

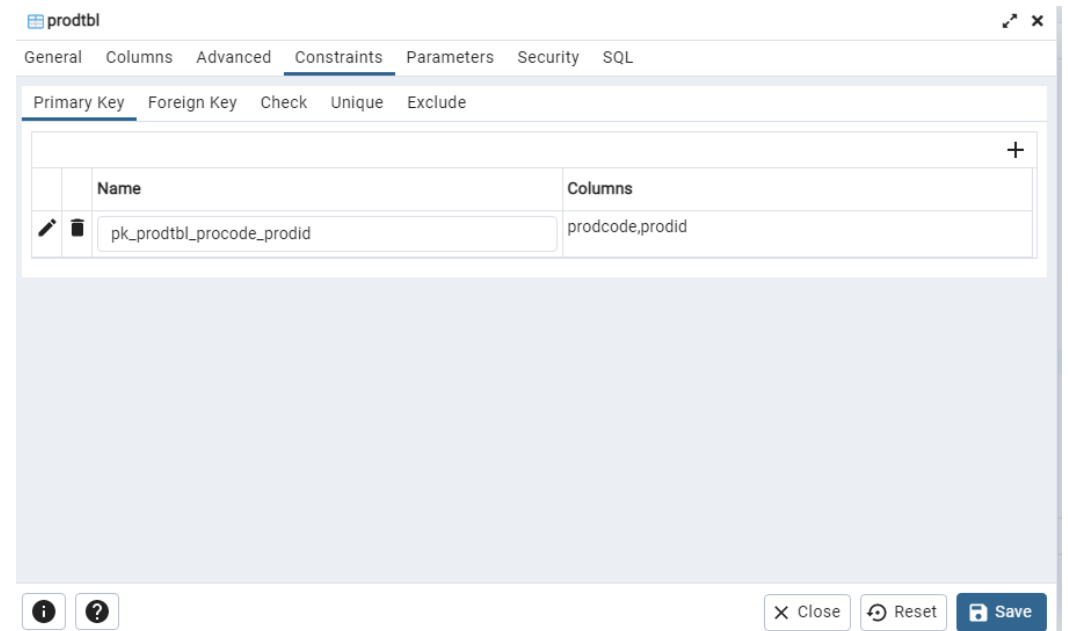
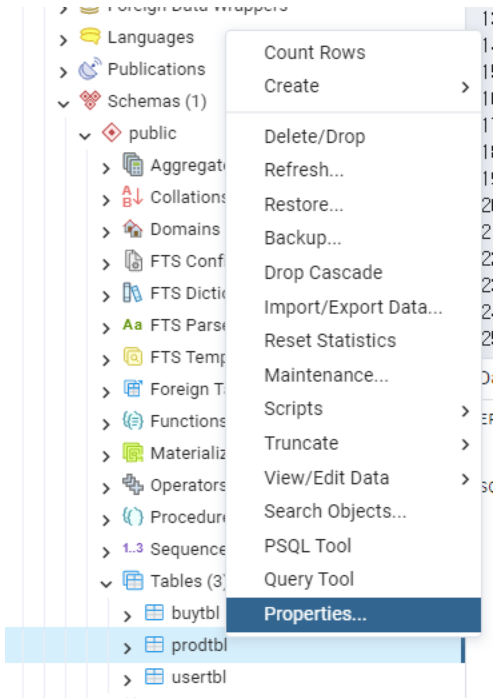
두 열이 합쳐져서 하나의 기본키 제약 조건을 설정하고 있음을 확인

SECTION 02

제약조건

Table 에 제약조건을 PgAdmin에서 확인하는 법

- PgAdmin 에서 제약조건은 table Properties안에 Constraints 에서 확인가능



SECTION 02 제약조건

데이터 무결성을 위한 제약 조건

- 외래 키(Foreign Key) 제약 조건
 - 두 테이블 사이의 관계 선언하여 데이터의 무결성 보장해주는 역할
 - 외래 키 관계를 설정하면 하나의 테이블이 다른 테이블에 의존
 - 외래 키 테이블이 참조하는 기준 테이블의 열은 반드시 Primary Key이거나 Unique 제약 조건이 설정되어 있어야 함
 - 외래 키의 옵션 중 ON DELETE CASCADE 또는 ON UPDATE CASCADE
 - 기준 테이블의 데이터가 변경되었을 때 외래 키 테이블도 자동으로 적용되도록 설정

SECTION 02 제약조건

데이터 무결성을 위한 제약 조건

- 외래 키(Foreign Key) 제약 조건
 - CREATE TABLE 끝에 FOREIGN KEY 키워드로 설정
 - 기존 테이블(userTBL)에 userID가 Primary Key 혹은 Unique가 아니라면 외래 키 설정 불가

```
DROP TABLE IF EXISTS buyTBL, userTBL;
CREATE TABLE userTBL
( userID CHAR(8) NOT NULL PRIMARY KEY,
  name VARCHAR(10) NOT NULL,
  birthYear INTEGER NOT NULL
);
CREATE TABLE buyTBL
(
  num SERIAL 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)
);
```

제약조건에 이름을 추가하고 싶다면 위와 같이 작성

외래 키의 이름을 지정할 필요가 없을 경우 제일 마지막 행에 간단히 FOREIGN KEY(userID) REFERENCES userTBL(userID)만 써줘도 됨

SECTION 02 제약조건

데이터 무결성을 위한 제약 조건

- 외래 키(Foreign Key) 제약 조건

- 외래 키 생성 방법 2

- ALTER TABLE 구문 이용

- ON DELETE CASCADE

- ON UPDATE CASCADE

기준 테이블의 데이터가 변경되었을 때 외래 키 테이블로 자동으로 적용되도록 설정
별도로 지정하지 않을 경우 ON UPDATE NO ACTION 및
ON DELETE NO ACTION을 지정한 것과 동일

```
CREATE TABLE buyTBL
( num SERIAL 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);
```

SECTION 02 제약조건

데이터 무결성을 위한 제약 조건

- UNIQUE 제약 조건
 - '중복되지 않는 유일한 값'을 입력해야 하는 조건
 - PRIMARY KEY와 비슷하나 UNIQUE는 NULL 값 허용
 - NULL은 여러 개가 입력되어도 상관 없음
 - ex) 회원 테이블 Email 주소 Unique로 설정

```
CREATE TABLE userTBL
(userID CHAR(8) NOT NULL PRIMARY KEY,
name VARCHAR(10) NOT NULL,
birthyear INTEGER NOT NULL,
email CHAR(30) NULL UNIQUE
);
```

```
DROP TABLE IF EXISTS 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)
```

); 모든 열의 정의가 끝난 상태에서 별도로 Unique제약 조건 추가
Email 정의 끝난 후 콤마(,)로 구분되어 있는 것 확인

SECTION 02 제약조건

데이터 무결성을 위한 제약 조건

◦ CHECK 제약 조건

- 입력되는 데이터를 점검하는 기능
 - ex) 키(Height) 제한 - 마이너스 값이 들어올수 없도록,
 - 출생년도 제한 - 1900년 이후이고 현재시점 이전
- ALTER TABLE문으로 제약 조건 추가 가능

테이블 정의 후 ALTER TABLE로 제약조건 추가
(전화번호 앞자리는 확인)

```
ALTER TABLE usertbl  
ADD CONSTRAINT CK_mobile1  
CHECK (mobile1 IN ('010','011','018','016','019','017'));
```

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

```
CREATE TABLE userTBL  
( userID CHAR(8) PRIMARY KEY,  
  name VARCHAR(10),  
  birthYear INTEGER CHECK (birthYear >= 1900 AND birthYear <= 2023),  
  mobile1 CHAR(3) NULL,  
  CONSTRAINT CK_name CHECK (name IS NOT NULL)  
);
```

SECTION 02 제약조건

데이터 무결성을 위한 제약 조건

◦ DEFAULT 정의

- 값 입력하지 않았을 때 자동으로 입력되는 기본 값 정의하는 방법
- ALTER TABLE 사용 시에 열에 DEFAULT를 지정하기 위해서 ALTER COLUMN문 사용

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

출생 년도를 입력하지 않으면 -1을 입력하고, 주소를 특별히 입력하지 않았다면 '서울'이 입력되며, 키를 입력하지 않으면 170이라고 입력되도록 함

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

ALTER TABLE 사용 시
열에 DEFAULT 지정하기 위해
ALTER COLUMN문 사용

SECTION 02 제약조건

데이터 무결성을 위한 제약 조건

- DEFAULT 정의
 - 디폴트 설정된 열에는 다음과 같은 방법으로 데이터 입력

```
-- default문은 DEFAULT로 설정된 값을 자동 입력한다.  
  
INSERT INTO usertbl VALUES ('LHL','이혜리',default,  
                             default, '011','1234567',default, '2023.12.12');  
  
-- 열 이름이 명시되지 않으면 DEFAULT로 설정된 값을 자동 입력한다.  
INSERT INTO usertbl (userID, name) VALUES('LMH' , '이문형');  
  
-- 값이 직접 명기되면 DEFAULT로 설정된 값은 무시된다.  
INSERT INTO usertbl VALUES ('WB','원빈',1982,'대전','019','9876543',176,  
                             '2020.5.5');  
  
SELECT * FROM usertbl;
```

| | userid [PK] character (8) | name character varying (10) | birthyear integer | addr character (2) | mobile1 character (3) | mobile2 character (8) | height smallint | mdate date |
|---|------------------------------|--------------------------------|----------------------|-----------------------|--------------------------|--------------------------|--------------------|---------------|
| 1 | LHL | 이혜리 | -1 | 서울 | 011 | 1234567 | 170 | 2023-12-12 |
| 2 | LMH | 이문형 | -1 | 서울 | [null] | [null] | 170 | [null] |
| 3 | WB | 원빈 | 1982 | 대전 | 019 | 9876543 | 176 | 2020-05-05 |

SECTION 02 제약조건

데이터 무결성을 위한 제약 조건

- Null 값 허용
 - NULL 값을 허용하려면 NULL을, 허용하지 않으려면 NOT NULL을 사용
 - PRIMARY KEY가 설정된 열에는 생략하면 자동으로 NOT NULL
 - NULL 값은 '아무 것도 없다'라는 의미, 공백(' ') 이나 0과 다름

SECTION 02 제약조건

테이블 삭제

```
DROP TABLE 테이블이름 ;
```

- 외래 키 제약 조건의 기준 테이블은 삭제할 수가 없음
 - 먼저 외래 키가 생성된 외래 키 테이블을 삭제해야 함
 - 구매 테이블이 존재하는데 회원 테이블을 삭제 할 수 없음, 구매 테이블 삭제가 선행 되어야 함
- 동시에 여러 테이블 삭제도 가능
 - DROP TABLE 테이블1, 테이블2, 테이블3;

SECTION 02 제약조건

테이블 수정

- ALTER TABLE문 사용
 - 테이블에 무엇인가 추가/변경/수정/삭제 모두 ALTER TABLE문 사용
- 열의 추가
 - 기본적으로 가장 뒤에 추가
 - 순서를 지정하려면 제일 뒤에 'FIRST' 또는 'ALTER 열 이름' 지정
 - ex) 회원 테이블(usertbl)에 회원 홈페이지 주소 추가

```
USE tabledb;  
ALTER TABLE usertbl  
    ADD homepage VARCHAR(30) -- 열 추가  
                        DEFAULT 'http://www.hanbit.co.kr' -- 디폴트 값  
                        NULL; -- Null 허용함
```

SECTION 02 제약조건

테이블 수정

- 열의 삭제

```
ALTER TABLE usertbl  
    DROP COLUMN mobile1;
```

- 제약 조건이 걸린 열을 삭제할 경우 제약 조건을 먼저 삭제한 후에 열을 삭제해야 함

- 열의 이름 및 데이터 형식 변경

- ex) 회원 이름(name)의 열 이름을 uName으로 변경하고 데이터 형식을 VARCHAR(20)으로 변경, NULL 값도 허용하는 경우

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

SECTION 02 제약조건

테이블 수정

- 열의 제약 조건 추가 및 삭제
 - ex) 기본 키를 삭제 하는 경우

```
ALTER TABLE usertbl  
    DROP PRIMARY KEY;
```

- 오류가 발생
 - usertbl의 기본 키인 userID열은 buytbl에 외래 키로 연결되어 있기 때문에 외래 키를 제거한 후에 다시 기본 키를 제거해야 함

```
ALTER TABLE buytbl  
    DROP FOREIGN KEY buytbl_ibfk_1;
```

[실습 3]

지금까지 익힌 테이블의 제약 조건 및 수정 방법을 실습을 통해서 익히자.

테이블 제약조건 및 수정 방법

- 조건 제외 테이블 재생성
- 단, 구매 테이블(buytbl)의 num열만 AUTO_INCREMENT 및 PRIMARY KEY 속성 부여

```
CREATE TABLE usertbl
(
    userID CHAR(8),
    name VARCHAR(10),
    birthYear INTEGER,
    addr CHAR(2),
    mobile1 CHAR(3),
    mobile2 CHAR(8),
    height SMALLINT,
    mDate DATE
);
```

```
CREATE TABLE buytbl
(
    num SERIAL PRIMARY KEY,
    userid CHAR(8),
    prodName CHAR(6),
    groupName CHAR(4),
    price INTEGER ,
    amount SMALLINT
);
```

[실습 3]

지금까지 익힌 테이블의 제약 조건 및 수정 방법을 실습을 통해서 익히자.

테이블 제약조건 및 수정 방법

- 각각의 테이블에 테이블당 4건씩만 입력
- 입력 시 김범수의 출생년도는 모르는 것으로 NULL값, 김경호의 출생년도는 1871년으로 잘못 입력

```
INSERT INTO usertbl VALUES('LSG','이승기',1987,'서울','011','11111111',182,'2008-8-8');
INSERT INTO usertbl VALUES('KBS','김범수', NULL ,'경남','011','22222222',173,'2012-4-4');
INSERT INTO usertbl VALUES('KKH','김경호',1871,'전남','019','33333333',177,'2007-7-7');
INSERT INTO usertbl VALUES('JYP','조용필',1950 ,'경기','011','11111111',166,'2009-4-4');

INSERT INTO buytbl VALUES(DEFAULT, 'KBS','운동화',NULL, 30, 2);
INSERT INTO buytbl VALUES(DEFAULT, 'KBS','노트북','전자', 1000, 1);
INSERT INTO buytbl VALUES(DEFAULT, 'JYP','모니터','전자', 200, 1);
INSERT INTO buytbl VALUES(DEFAULT, 'BBK','모니터','전자', 200, 5);
```

- 아직 FOREIGN KEY 제약 조건이 설정된 것이 아니므로, usertbl에 BBK(바비킴) 회원이 없지만, 입력은 잘 됨
- DEFAULT 값으로 넣어주는 것은 SERIAL에 경우, NOT NULL 조건이 자동으로 붙게됨

[실습 3]

지금까지 익힌 테이블의 제약 조건 및 수정 방법을 실습을 통해서 익히자.

테이블 제약조건 및 수정 방법

- 제약 조건 생성

```
ALTER TABLE usertbl  
ADD CONSTRAINT PK_usertbl_userID  
PRIMARY KEY (userID);
```

PRIMARY KEY로 설정하려면 당연히 NOT NULL이어야 하지만 NULL인 열에 PRIMARY KEY를 설정하면 자동으로 NOT NULL까지 설정됨

- 테이블 확인

| Name | Data type | Length/Precision | Scale | Not NULL? | Primary key? | Default |
|-----------|-------------------|------------------|-------|-----------|--------------|---------|
| userid | character | 8 | | Yes | Yes | |
| name | character varying | 10 | | No | No | |
| birthyear | integer | | | No | No | |
| addr | character | 2 | | No | No | |
| mobile1 | character | 3 | | No | No | |
| mobile2 | character | 8 | | No | No | |
| height | smallint | | | No | No | |
| mdate | date | | | No | No | |

userid열이 NOT NULL로 설정된 것 확인

[실습 3]

지금까지 익힌 테이블의 제약 조건 및 수정 방법을 실습을 통해서 익히자.

테이블 제약조건 및 수정 방법

- 외래 키 설정

```
ALTER TABLE buytbl
  ADD CONSTRAINT FK_usertbl_buytbl
  FOREIGN KEY (userID)
  REFERENCES usertbl (userID);
```

Buytbl에는 BBK(바비킴)의 구매 기록이 있는데, 이 BBK 아이디가 usertbl에는 존재하지 않기 때문

ERROR: 오류: "buytbl" 테이블에서 자료 추가, 갱신 작업이 "fk_usertbl_buytbl" 참조키(foreign key) 제약 조건을 위반했습니다
DETAIL: (userid)=(BBK) 키가 "usertbl" 테이블에 없습니다.

- Buytbl BBK행 삭제 후 외래 키 재설정

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

```
ALTER TABLE buytbl
  ADD CONSTRAINT FK_usertbl_buytbl
  FOREIGN KEY (userID)
  REFERENCES usertbl (userID);
```

ALTER TABLE

Query returned successfully in 31 msec.

- Buytbl 네 번째 데이터 재입력

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

ERROR: 오류: "num" 칼럼(해당 릴레이션 "buytbl")의 null 값이 not null 제약조건을 위반했습니다.

DETAIL: 실패한 자료: (null, BBK , 모니터 , 전자 , 200, 5)

외래 키가 연결되어 활성화된 상태이므로 새로 입력하는 데이터는 모두 외래 키 제약 조건을 만족해야 함
BBK가 아직 usertbl에 없기 때문에 발생하는 오류임

[실습 3]

지금까지 익힌 테이블의 제약 조건 및 수정 방법을 실습을 통해서 익히자.

테이블 제약조건 및 수정 방법

- Buytbl 데이터 입력하는 동안 잠시 외래 키 제약조건 비활성화

```
ALTER TABLE buytbl DISABLE TRIGGER ALL;
```

```
INSERT INTO buytbl VALUES(DEFAULT, 'BBK', '모니터', '전자', 200, 5);  
INSERT INTO buytbl VALUES(DEFAULT, 'KBS', '청바지', '의류', 50, 3);  
INSERT INTO buytbl VALUES(DEFAULT, 'BBK', '메모리', '전자', 80, 10);  
INSERT INTO buytbl VALUES(DEFAULT, 'BBK', '운동화', NULL, 30, 2);
```

```
ALTER TABLE buytbl ENABLE TRIGGER ALL;
```

- 외래 키 제약조건 활성화 (마지막 라인)

```
ALTER TABLE buytbl ENABLE TRIGGER ALL;
```

```
ALTER TABLE buytbl DISABLE TRIGGER ALL;
```

[실습 3]

지금까지 익힌 테이블의 제약 조건 및 수정 방법을 실습을 통해서 익히자.

테이블 제약조건 및 수정 방법

- CHECK 제약조건 설정

```
ALTER TABLE userTBL
ADD CONSTRAINT CK_birthYear
CHECK ((birthYear >= 1900 AND birthYear <= 2023) AND (birthYear IS NOT NULL));
```

ERROR: 오류: "ck_birthyear" 체크 제약 조건(해당 릴레이션 "usertbl")을 위반하는 몇몇 자료가 있습니다

김범수(KBS) 출생년도 모르는 것으로 NULL값을 넣고 김경호(KKH)의 출생년도를 1871로 잘못 입력했기 때문

usertbl의 출생년도를 1900~2023까지만 입력되도록 함

NULL값을 허용하지 말고 출생년도는 반드시 입력하도록 설정

```
UPDATE usertbl SET birthYear=1979 WHERE userID='KBS';
UPDATE usertbl SET birthYear=1971 WHERE userID='KKH';
```

```
ALTER TABLE userTBL
ADD CONSTRAINT CK_birthYear
CHECK ((birthYear >= 1900 AND birthYear <= 2023) AND (birthYear IS NOT NULL));
```

- 데이터 수정 후 제약조건 재설정 및 테스트

```
INSERT INTO usertbl VALUES('TKV', '태권뷔', 2999, '우주', NULL, NULL, 186, '2023-12-12');
```

ERROR: 오류: 새 자료가 "usertbl" 릴레이션의 "ck_birthyear" 체크 제약 조건을 위반했습니다
DETAIL: 실패한 자료: (TKV, 태권뷔, 2999, 우주, null, null, 186, 2023-12-12)

출생년도가 잘못된 값을 입력할 경우 오류 발생

[실습 3]

지금까지 익힌 테이블의 제약 조건 및 수정 방법을 실습을 통해서 익히자.

테이블 제약조건 및 수정 방법

- 나머지 usertbl 정상 데이터 입력

```
INSERT INTO usertbl VALUES('SSK','성시경',1979,'서울',NULL,NULL,186,'2008-8-8');
INSERT INTO usertbl VALUES('LJB','임재범',1963,'서울','016','66666666',182,'2009-9-9');
INSERT INTO usertbl VALUES('YJS','윤종신',1969,'경남',NULL,NULL,174,'2005-5-5');
INSERT INTO usertbl VALUES('BBK','바비킴',1973,'서울','010','00000000',176,'2013-5-5');
INSERT INTO usertbl VALUES('EJW','은지원',1972,'경북','011','88888888',174,'2014-3-5');
```

- 바비킴 회원 ID 변경(BBK -> VVK)

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

ERROR: 오류: "usertbl" 테이블의 자료 갱신, 삭제 작업이 "fk_usertbl_buytbl" 참조키(foreign key) 제약 조건 - "buytbl" 테이블 - 을 위반했습니다
DETAIL: (userid)=(BBK) 키가 "buytbl" 테이블에서 여전히 참조됩니다.

BBK는 이미 buytbl에서 구매한 기록이 있으므로 바뀌지 않는 것

[실습 3]

지금까지 익힌 테이블의 제약 조건 및 수정 방법을 실습을 통해서 익히자.

테이블 제약조건 및 수정 방법

- 제약조건 잠시 비활성화, 데이터 변경 후 활성화

```
ALTER TABLE usertbl DISABLE TRIGGER ALL;
```

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

```
ALTER TABLE usertbl ENABLE TRIGGER ALL;
```

- 물품 배송을 위해 회원 테이블, 구매 테이블 조인

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

| | userid character (8) | name character varying (10) | prodname character (6) | addr character (2) | ph text |
|---|-------------------------|--------------------------------|---------------------------|-----------------------|------------|
| 1 | KBS | 김범수 | 운동화 | 경남 | 0112222... |
| 2 | KBS | 김범수 | 노트북 | 경남 | 0112222... |
| 3 | JYP | 조용필 | 모니터 | 경기 | 0111111... |
| 4 | KBS | 김범수 | 청바지 | 경남 | 0112222... |

7건 입력했는데 4건 나왔음, 3건은 어디에 있을까?

- 구매 테이블에 4건만 입력된 건 아닌지 확인

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

| Data output | | Messages | Notifications |
|-------------|-----------------|----------|---------------|
| | | | |
| | count bigint | | |
| 1 | 7 | | |

7건의 데이터가 존재하는 것을 확인할 수 있음

[실습 5]

지금까지 익힌 테이블의 제약 조건 및 수정 방법을 실습을 통해서 익히자.

테이블 제약조건 및 수정 방법

- 외부조인으로 구매 테이블의 내용 모두 출력 및 아이디로 정렬

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

| | userid character (8) | name character varying (10) | prodname character (6) | addr character (2) | ph text |
|---|-------------------------|--------------------------------|---------------------------|-----------------------|------------|
| 1 | BBK | [null] | 모니터 | [null] | |
| 2 | BBK | [null] | 메모리 | [null] | |
| 3 | BBK | [null] | 운동화 | [null] | |
| 4 | JYP | 조용필 | 모니터 | 경기 | 0111111... |
| 5 | KBS | 김범수 | 운동화 | 경남 | 0112222... |
| 6 | KBS | 김범수 | 청바지 | 경남 | 0112222... |
| 7 | KBS | 김범수 | 노트북 | 경남 | 0112222... |

BBK는 이름, 주소, 연락처가 없는 것 확인
→ VVK로 변경했었기 때문

- 바비킴 아이디 원상복구

```
ALTER TABLE usertbl DISABLE TRIGGER ALL;

UPDATE usertbl SET userID = 'BBK' WHERE userID='VVK';

ALTER TABLE usertbl ENABLE TRIGGER ALL;
```

| | userid character (8) | name character varying (10) | prodname character (6) | addr character (2) | ph text |
|---|-------------------------|--------------------------------|---------------------------|-----------------------|-------------|
| 1 | BBK | 바비킴 | 모니터 | 사을 | 01000000000 |
| 2 | BBK | 바비킴 | 메모리 | 사을 | 01000000000 |
| 3 | BBK | 바비킴 | 운동화 | 사을 | 01000000000 |
| 4 | JYP | 조용필 | 모니터 | 경기 | 01111111111 |
| 5 | KBS | 김범수 | 운동화 | 경남 | 01122222222 |
| 6 | KBS | 김범수 | 청바지 | 경남 | 01122222222 |
| 7 | KBS | 김범수 | 노트북 | 경남 | 01122222222 |

[실습 5]

지금까지 익힌 테이블의 제약 조건 및 수정 방법을 실습을 통해서 익히자.

테이블 제약조건 및 수정 방법

- 회원테이블 userID가 바뀔 때 이와 관련되는 구매 테이블의 userID도 자동 변경되도록 함

```
ALTER TABLE buytbl  
DROP CONSTRAINT FK_usertbl_buytbl;
```

```
ALTER TABLE buytbl  
ADD CONSTRAINT FK_usertbl_buytbl  
FOREIGN KEY (userid)  
REFERENCES usertbl (userID)  
ON UPDATE CASCADE;
```

외래키 제약조건 삭제 후 다시 ON UPDATE CASCADE 옵션과 함께 설정

- Usertbl의 바비킴의 ID를 다시 변경하고, buytbl에도 바뀌었는지 확인

```
UPDATE usertbl SET userID = 'VVK' WHERE userID = 'BBK';  
SELECT B.userid, U.name, B.prodName, U.addr, CONCAT(U.mobile1, U.mobile2) as PH  
FROM buytbl B  
INNER JOIN usertbl U  
ON B.userid = U.userid  
ORDER BY B.userid;
```

| | userid character (8) 🔒 | name character varying (10) 🔒 | prodname character (6) 🔒 | addr character (2) 🔒 | ph text 🔒 |
|---|---------------------------|----------------------------------|-----------------------------|-------------------------|--------------|
| 1 | JYP | 조용필 | 모니터 | 경기 | 0111111... |
| 2 | KBS | 김범수 | 노트북 | 경남 | 0112222... |
| 3 | KBS | 김범수 | 운동화 | 경남 | 0112222... |
| 4 | KBS | 김범수 | 청바지 | 경남 | 0112222... |
| 5 | VVK | 바비킴 | 모니터 | 서울 | 0100000... |
| 6 | VVK | 바비킴 | 메모리 | 서울 | 0100000... |
| 7 | VVK | 바비킴 | 운동화 | 서울 | 0100000... |