



Ch 08. 테이블과 뷰

이것이 MariaDB다

❖ 핵심 개념

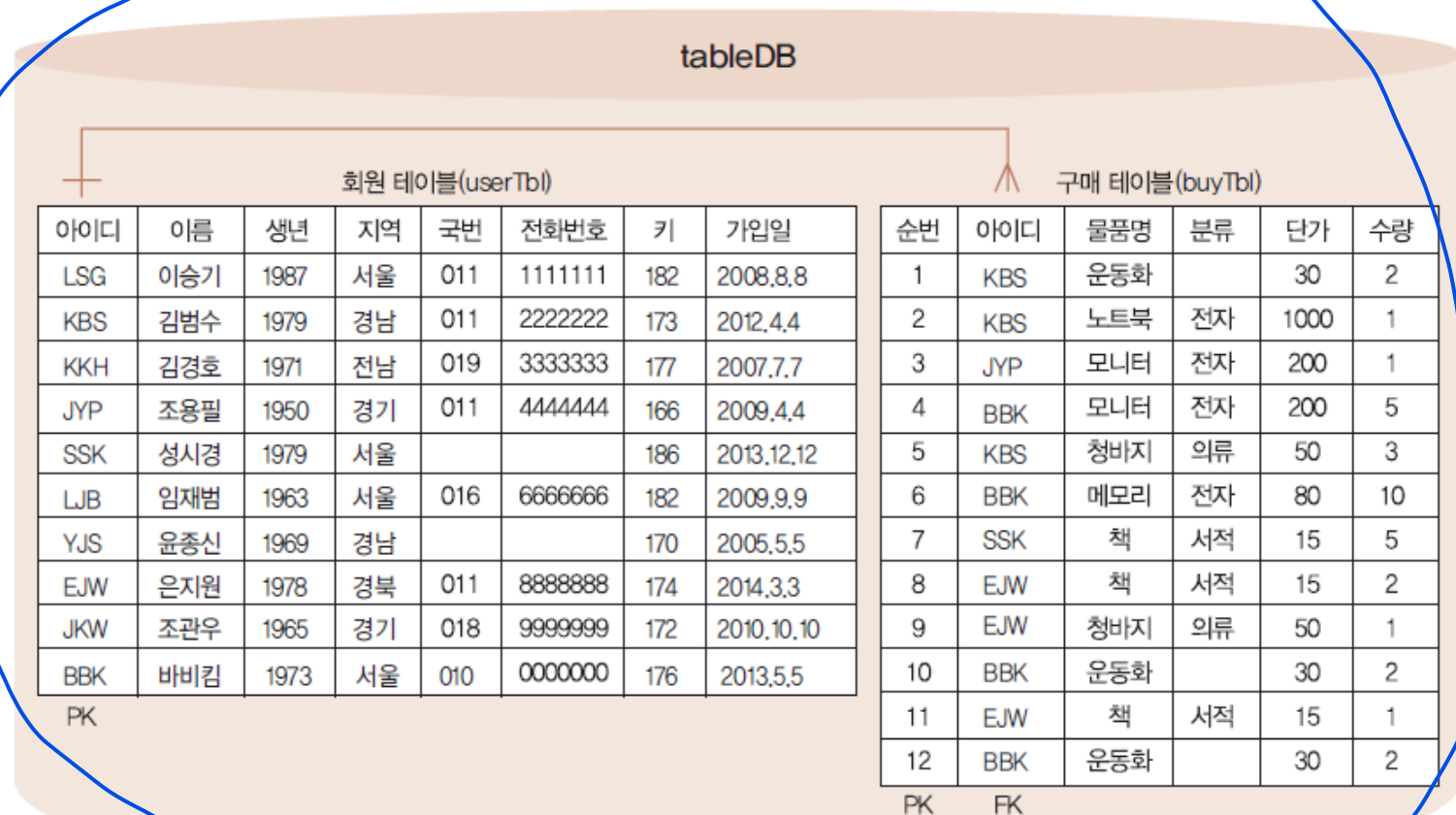
- 테이블의 생성
- 제약 조건: 기본 키, 외래 키 등
- 테이블 압축과 효율성 및 임시 테이블의 활용
- 뷰의 개념과 장단점



8.1 테이블

8.1.1 테이블 만들기

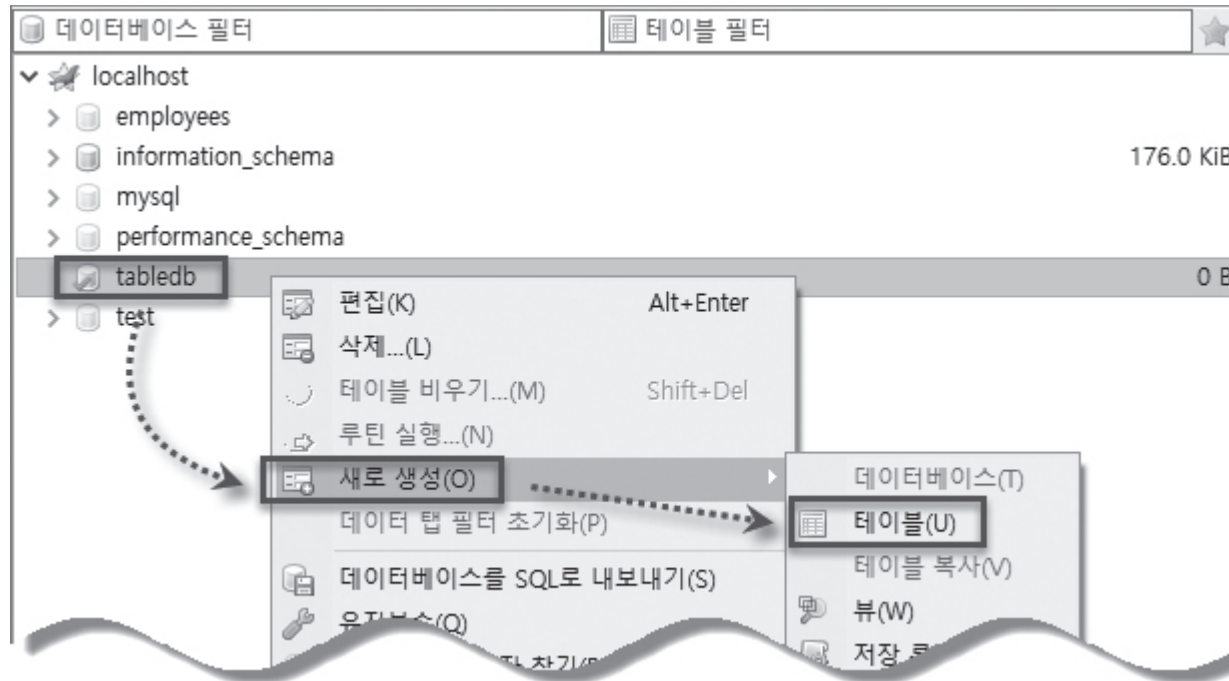
- HeidiSQL에서 테이블 생성 *직접 만들기*
- 6장의 sqlDB와 동일한 형식의 데이터베이스



8.1 테이블

8.1.1 테이블 만들기

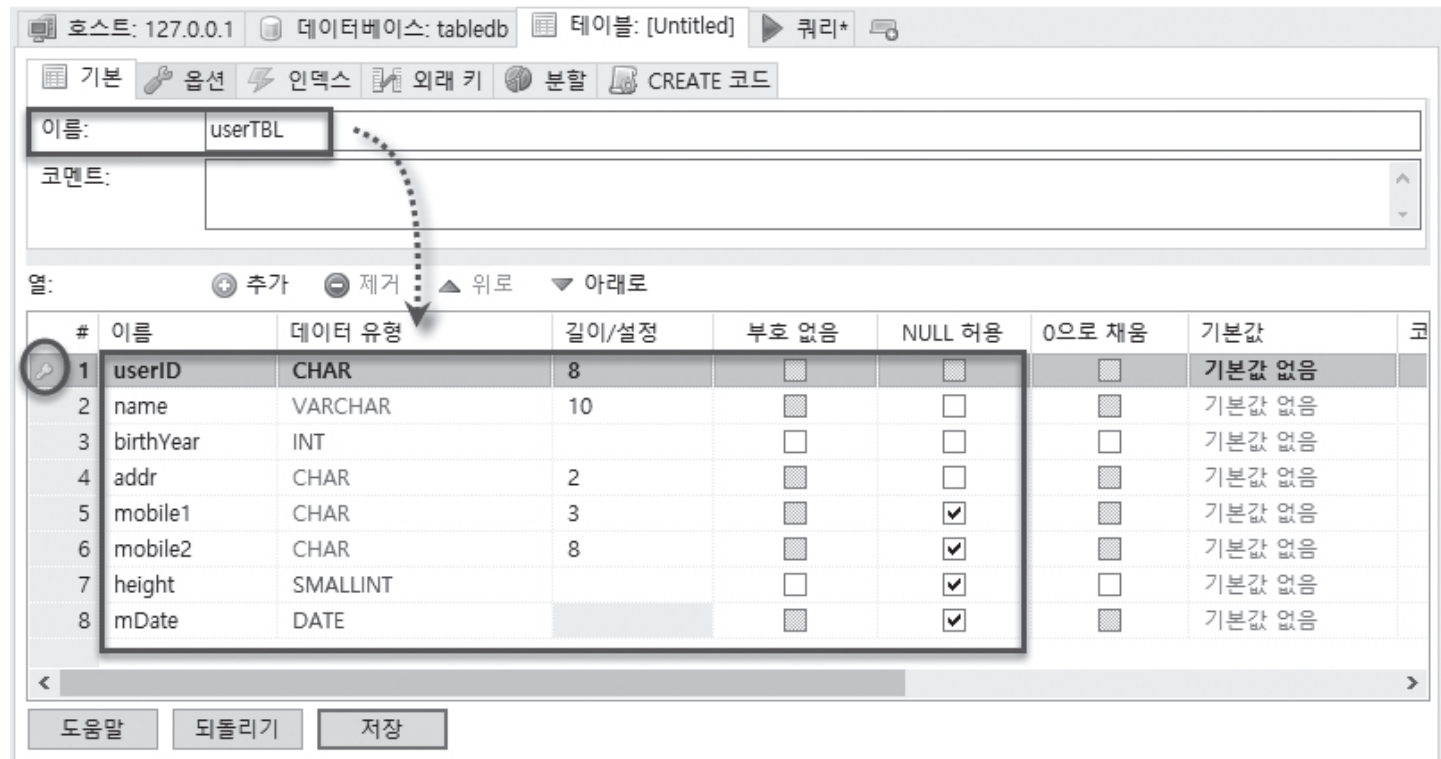
- HeidiSQL에서 테이블 생성
 - GUI 환경에서 테이블 생성



8.1 테이블

8.1.1 테이블 만들기

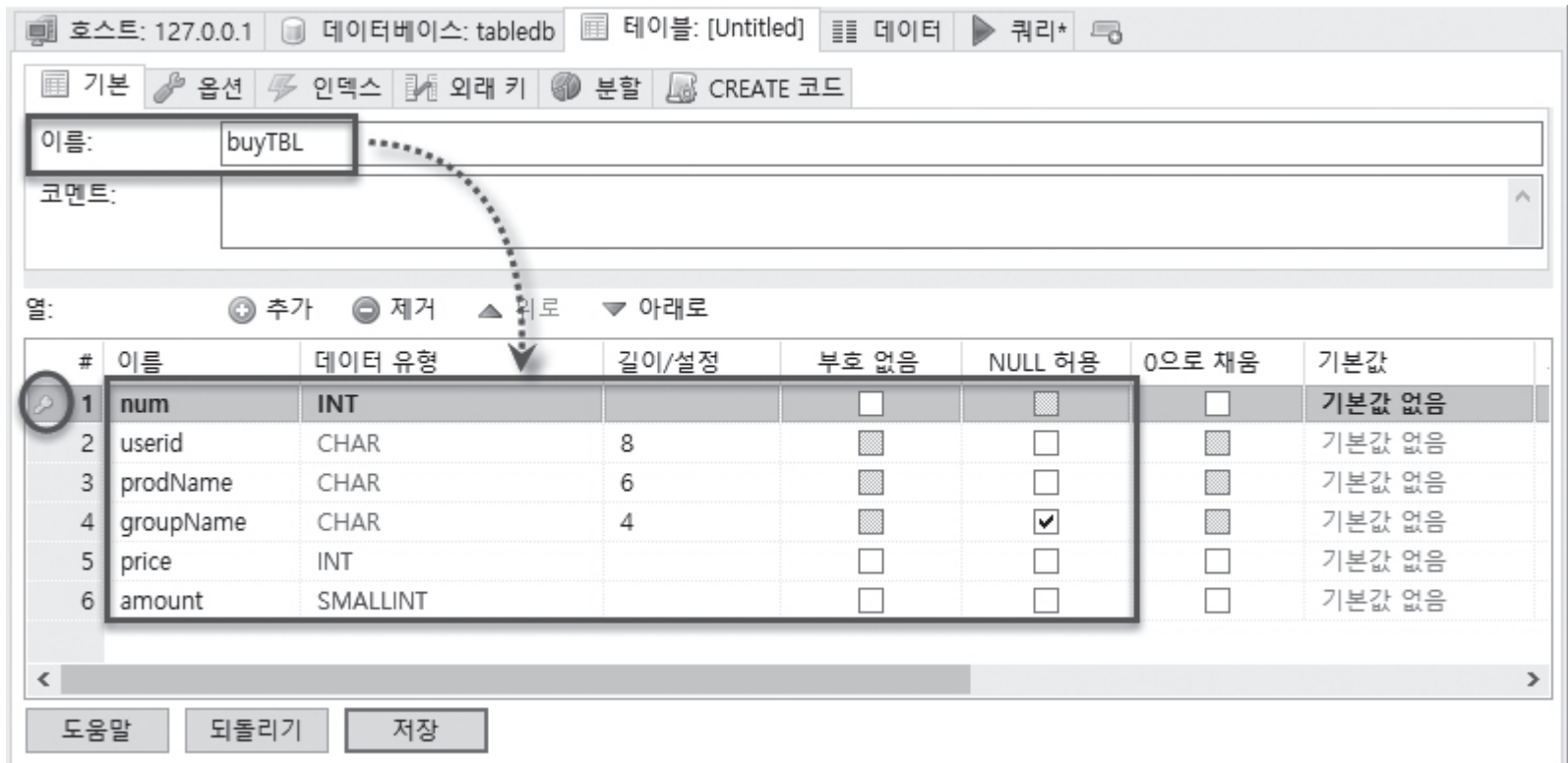
- HeidiSQL에서 테이블 생성
 - GUI 환경에서 테이블 입력
 - 회원테이블. Primary Key는 userID



8.1 테이블

8.1.1 테이블 만들기

- HeidiSQL에서 테이블 생성
 - GUI 환경에서 테이블 입력
 - 구매테이블. Primary Key는 num



8.1 테이블

8.1.1 테이블 만들기

- HeidiSQL에서 테이블 생성

- GUI 환경에서 테이블 입력

– 구매테이블의 순번 열은 자동증가 설정

열: + 추가 - 제거 ▲ 위로 ▼ 아래로

#	이름	데이터 유형	길이/설정	부호 없음	NULL 허용	0으로 채움	기본값
1	num	INT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	기본값 없음
2	userid	CHAR	8	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
3	prodName	CHAR	6	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
4	groupName	CHAR	4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
5	price	INT		<input type="checkbox"/>	<input type="checkbox"/>		
6	amount	SMALLINT		<input type="checkbox"/>	<input type="checkbox"/>		

☐ 기본값 없음
☐ Custom:
☐ NULL
☐ CURRENT_TIMESTAMP
☐ ON UPDATE CURRENT_TIMESTAMP
☒ AUTO_INCREMENT

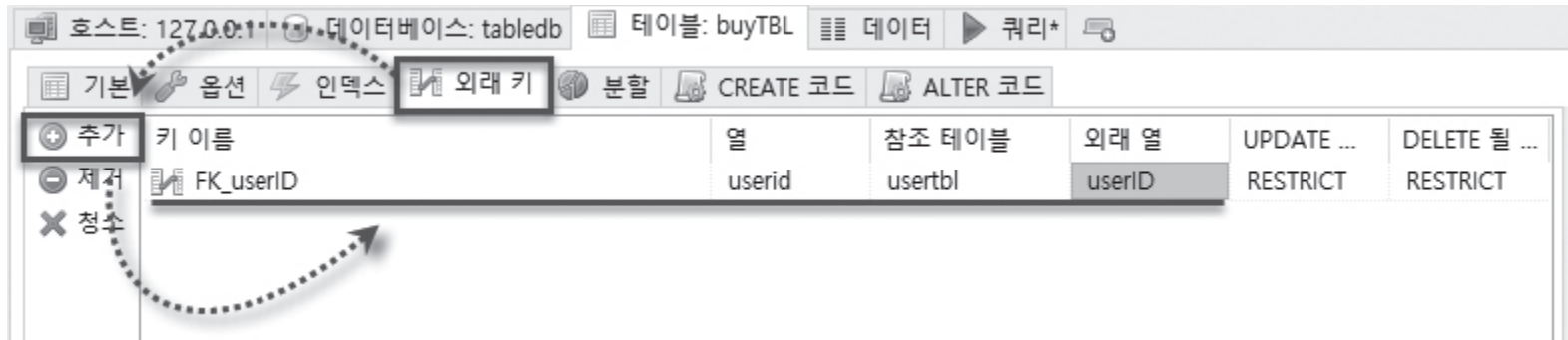
확인 취소



8.1 테이블

8.1.1 테이블 만들기

- HeidiSQL에서 테이블 생성
 - GUI 환경에서 테이블 입력
 - 구매 테이블의 userID에 외래 키 설정



8.1 테이블

8.1.1 테이블 만들기

- HeidiSQL에서 테이블 생성
 - GUI 환경에서 테이블 입력

– HeidiSQL에서 데이터 입력하기 (외래 키 설정 확인)

HeidiSQL interface showing two database tables:

Table 1: usertbl

userID	name	birthYear	addr	mobile1	mobile2	height	mDate
KBS	김범수	1,979	경남	011	22222222	173	2012-04-04
KKH	김경호	1,971	전남	019	33333333	177	2007-07-07
LSG	이승기	1,987	서울	011	11111111	182	2008-08-08

Table 2: buytbl

num	userid	prodName	groupName	price	amount
(NULL)			(NULL)		

A dropdown menu is open for the 'userid' column in 'buytbl', showing options: KKH: 김경호, KBS: 김범수, LSG: 이승기.

8.1 테이블

8.1.1 테이블 만들기

■ SQL로 테이블 생성

- 처음에는 열 이름, 형식 지정하는 기본 틀 구성
- NOT NULL 과 NULL 표시
- 테이블에 기본 키, AUTO_INCREMENT 지정
- 외래 키 설정

– FOREIGN KEY REFERENCES
userTBL(userID)문

» userTBL 테이블의 userID열과 외래 키 관계
를 맺어라

- 데이터 입력까지 완성해 SQL 실습 완료



8.1 테이블

8.1.1 테이블 만들기

■ SQL로 테이블 생성

```
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)
);
```

```
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
);
```



8.1 테이블

8.1.1 테이블 만들기

■ SQL로 테이블 생성

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

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

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

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

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

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



8.1 테이블

8.1.2 제약 조건

- 제약 조건 – Constraint
- 데이터의 무결성을 지키기 위한 제한된 조건
- 특정 데이터를 입력할 때 어떠한 조건을 만족했을 때 입력 되도록 제약 가능
- MariaDB가 제공하는 6가지 제약 조건
 - PRIMARY KEY 제약 조건 NOT NULL / VN 1 QVE
 - FOREIGN KEY 제약 조건
 - UNIQUE 제약 조건
 - CHECK 제약 조건
 - DEFAULT 정의
 - NULL 값 허용



8.1 테이블

8.1.2 제약 조건

■ 기본 키 제약 조건

- 테이블에 존재하는 많은 행 데이터 구분 가능한 식별자
 - Ex) 회원테이블의 회원 아이디, 학생 테이블의 학번
- 기본 키에 입력되는 값은 중복 불가
- NULL 값 입력 불가
- 자동으로 클러스터형 인덱스 생성
- 테이블에서는 기본 키를 하나 이상의 열에 설정 가능
- PRIMARY KEY의 이름을 사용자가 지정 가능

```
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)
);
```



8.1 테이블

8.1.2 제약 조건

■ 기본 키 제약 조건

- CREATE TABLE 문 안에 보통 위치
- ALTER TABLE 문 사용해 수정 구문 사용 가능

```
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);
```

- 두 개의 열을 같이 기본 키로 사용할 경우 ,로 구분
- SHOW INDEX FROM 테이블_이름 - 설정 확인 가능



8.1 테이블

8.1.2 제약 조건

```
SET foreign_key_checks = 0;
```

■ 외래 키 제약 조건

- 두 테이블 사이의 관계를 선언함으로써, 데이터의 무결성을 보장해 주는 역할
- 하나의 테이블이 다른 테이블에 의존
- 외래 키 테이블에 데이터를 입력할 때는 꼭 기준 테이블을 참조해 입력
 - 기준 테이블에 이미 데이터가 존재해야
- 외래 키 테이블이 참조하는 기준 테이블의 열
 - 반드시 Primary Key
 - Unique 제약 조건이 설정되어 있어야 함



8.1 테이블

8.1.2 제약 조건

■ 외래 키 제약 조건

- 외래 키 설정은 CREATE TABLE 내부에 FOREIGN KEY 키워드로 설정
- 외래 키의 이름을 지정할 필요가 없다면
– REFERENCES userTBL(userID)

```
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)
);
```

식별자



8.1 테이블

8.1.2 제약 조건

■ 외래 키 제약 조건

- 역시 ALTER TABLE 구문으로도 사용 가능
 - 테이블 구조 먼저 선언 후 수정 형식
- SHOW INDEX FROM 테이블_이름으로 확인 가능
- ON UPDATE CASCADE, ON DELETE CASCADE
 - 회원 테이블 변화에 따라 외래 키 테이블도 변화
 - 지정하지 않으면 ON UPDATE NO ACTION, ON DELETE NO ACTION



8.1 테이블

8.1.2 제약 조건

■ 외래 키 제약 조건

```
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 ;
```

```
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;
```

➔ 자식 테이블도 같이 변경



8.1 테이블

8.1.2 제약 조건

■ UNIQUE 제약 조건

- ‘중복되지 않는 유일한 값’ 을 입력해야 하는 조건
- PRIMARY KEY와 차이점
 - UNIQUE는 NULL 값 허용
 - Ex) 회원 정보 테이블의 E-mail
- Alternate Key 라고도 부름

$\frac{1}{0} \frac{4}{1} \frac{7}{1}$



8.1 테이블

8.1.2 제약 조건

■ UNIQUE 제약 조건

```
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 →윜명자
);
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) →윜명자
);
```



8.1 테이블

8.1.2 제약 조건

★ CHECK 제약 조건

- 입력되는 데이터를 점검하는 기능
- ex) 키(height)에 마이너스 값 불가, 출생연도가 1900년 이후이고 현재 시점 이전이어야 한다든지 등의 조건 지정
- 열 정의한 후 바로 CHECK 예약어로 조건 지정
- 정의가 끝난 후 ALTER TABLE 로 수정하는 방법
- CHECK 제약 조건을 무시하려면 시스템 변수 중 `check_constraint_checks = 0`으로 설정
- P.322의 SQL 문을 이해하고 넘어가도록 함

→ 하지만
무시X



8.1 테이블

8.1.2 제약 조건

■ CHECK 제약 조건

```
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)
);
```

//이름이 반드시 값이
있도록

-- 휴대폰 국번 체크

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



8.1 테이블

8.1.2 제약 조건

■ DEFAULT 정의

- 값을 입력하지 않았을 때, 자동 입력되는 기본값 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 NOT NULL DEFAULT 170,
  mDate   date NULL
);
```



8.1 테이블

8.1.2 제약 조건

▪ DEFAULT 정의

- ALTER TABLE 문을 이용할 때
 - ALTER COLUMN문으로 사용됨

```
DROP TABLE IF EXISTS userTBL;
CREATE TABLE userTBL
( userIDchar(8) NOT NULL PRIMARY KEY,
  namevarchar(10) NOT NULL,
  birthYearint NOT NULL ,
  addrchar(2) NOT NULL,
  mobile1char(3) NULL,
  mobile2char(8) NULL,
  heightsmallint NULL,
  mDateate 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;
-- 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;
```



8.1 테이블

8.1.2 제약 조건

■ NULL 값 허용

- NULL 값을 허용하려면 NULL, 허용하지 않으려면 NOT NULL을 사용
- PRIMARY KEY가 설정된 열 - 자동 NOT NULL
- NULL 값
 - ‘아무 것도 없다’ 라는 의미
 - 공백(‘ ’)이나 0과 같은 값과 다름
- Null을 저장할 때
 - 고정 길이 문자형(char) 은 공간을 모두 차지
 - 가변 길이 문자형(varchar) 은 공간을 차지하지 않음 . Null 값을 많이 입력한다면 가변 길이 데이터 형식을 사용하는 것이 효과적



8.1 테이블

8.1.3 테이블 압축

- 압축 기능 - 대용량 테이블의 공간을 절약하는 효과
- 열 뒤에 ROW_FORMAT = COMPRESSED 를 붙이면 압축되도록 저장.
- 테이블에 데이터를 INSERT 하는 시간이 늘어나며, 같은 양의 데이터를 가진 테이블보다 평균 행 길이나 데이터 길이는 작아짐

```
209 INSERT INTO normalTBL .....SELECT emp_no, first_name FROM employees.employees;  
210 /* Affected rows: 300,024  찾은 행: 0 경고: 0 지속 시간: 쿼리: 0.781 sec. */  
211 INSERT INTO compressTBL .....SELECT emp_no, first_name FROM employees.employees;  
212 /* Affected rows: 300,024  찾은 행: 0 경고: 0 지속 시간: 쿼리: 2.860 sec. */
```

TABLES (20x2)									
Name	Engine	Version	Row_format	Rows	Avg_row_length	Data_length	Max_data_length	Index_length	Data_free
compresstbl	InnoDB	10	Compressed	299,503	25	7,626,752	0	0	2,097,152
normaltbl	InnoDB	10	Dynamic	299,316	40	12,075,008	0	0	4,194,304



8.1 테이블

8.1.3 테이블 압축

```
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;

USE mysql;
DROP DATABASE IF EXISTS compressDB;
```



8.1 테이블

8.1.4 임시 테이블

- 임시로 잠깐 사용되는 테이블
- 세션 내에서만 존재 가능하며 세션 닫히면 자동 삭제
- 생성한 클라이언트에서만 사용 가능
- 임시 테이블이 삭제되는 시점
 - 사용자가 DROP TABLE로 직접 삭제
 - HeidiSQL을 종료하면 삭제됨
 - 클라이언트 프로그램을 종료하면 삭제됨
 - MariaDB 서비스가 재시작되면 삭제됨
- 임시테이블 생성 형식

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



8.1 테이블

8.1.4 임시 테이블

■ 임시로 잠깐 사용되는 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;
```

```
USE employees;
SELECT * FROM tempTBL;
SELECT * FROM employees;
```

```
DROP TABLE tempTBL ;
```

```
USE employees;
SELECT * FROM employees;
```



8.1 테이블

8.1.4 테이블 삭제

■ 주의 사항

- 외래 키 (FOREIGN KEY) 제약 조건의 기준 테이블 삭제 불가 .
 - 외래 키가 생성된 외래 키 테이블을 먼저 삭제
 - Ex) 회원 테이블은 구매테이블 삭제 후 삭제 가능

■ 여러 개의 테이블 동시에 삭제

- DROP TABLE 테이블1, 테이블2, 테이블3;



8.1 테이블

8.1.4 테이블 수정

- ALTER TABLE 다음에 필요한 조건을 나열
 - 이미 만들어진 테이블에 추가/변경/수정/삭제...

add modify drop



8.1 테이블

8.1.4 테이블 수정

■ 열의 추가

- 열을 추가하면 기본적으로 가장 뒤에 추가
- 열을 추가하면서 순서 지정
 - 구문 맨 뒤에 ‘FIRST’ 나 AFTER 열이름’ 지정
 - FIRST는 제일 앞에 열 추가
 - AFTER 열 이름은 열 이름 다음에 추가

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



8.1 테이블

8.1.4 테이블 수정

■ 열의 삭제

- 제약 조건이 걸린 열을 삭제할 경우
– 제약 조건을 먼저 삭제한 후 열 삭제

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

■ 열의 이름 및 데이터 형식 변경

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

■ 열의 제약 조건 추가 및 삭제 – FK 삭제 후 PK 삭제

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

