

2025년 상반기 K-디지털 트레이닝

테이블

[KB] IT's Your Life

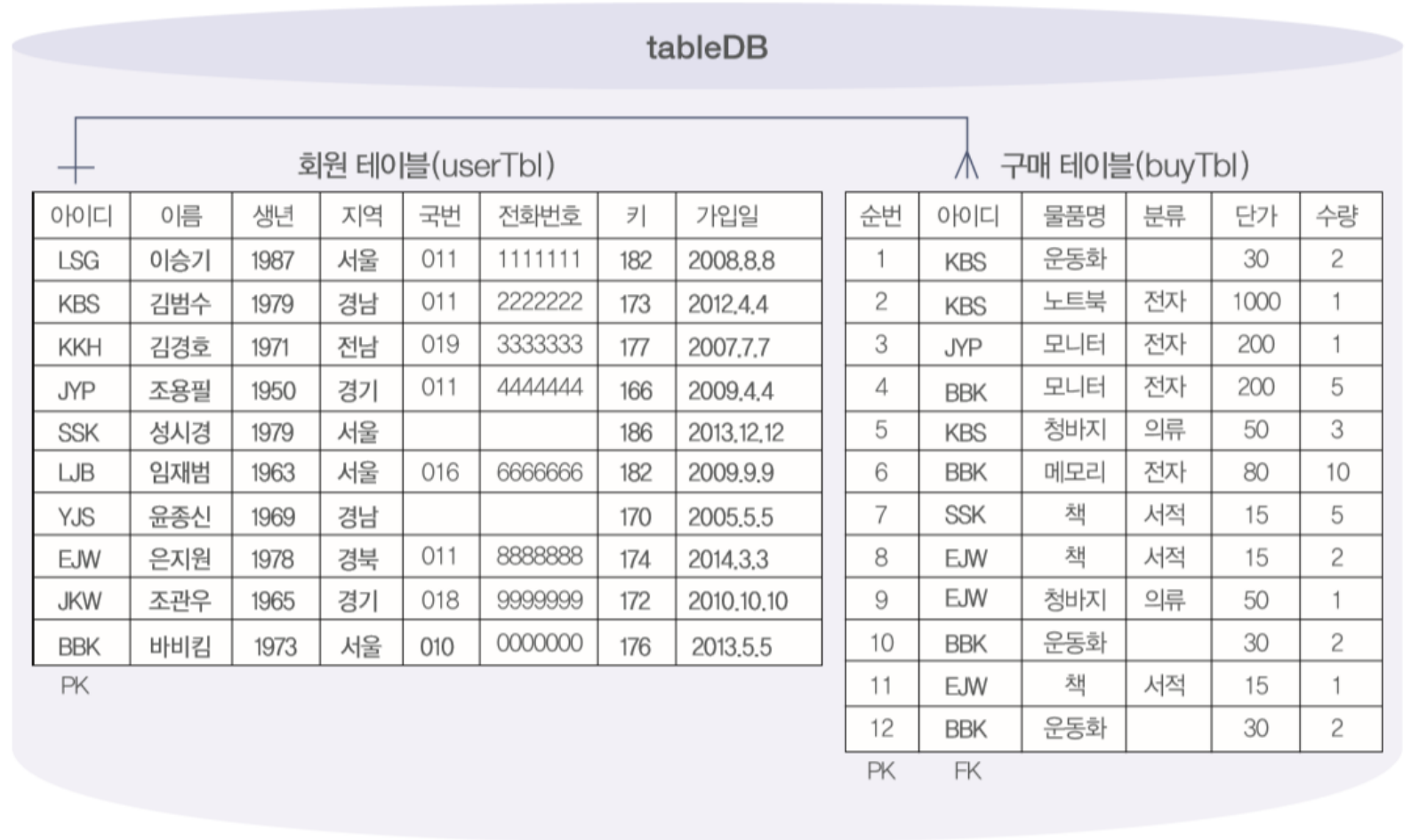
1 테이블

✓ 데이터베이스 만들기

```

DROP DATABASE tabledb;
CREATE DATABASE tabledb;
USE tabledb;

```



✔ usertbl만들기

```
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  
);
```

✅ buytbl 만들기

```
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)  
);
```

1 테이블

✓ 테이블 만들기

○ 회원 테이블 데이터 입력

```
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');
```

○ 구매 테이블 데이터 입력

```
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);
```

○ 구매 테이블 데이터 입력 시 3번째 행은 앞과 같이 에러 발생하므로 삭제하고 입력

1 테이블

✓ 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 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

1 테이블

✓ 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 usertbl2;
```

```
CREATE TABLE usertbl2 (  
    userID CHAR(8) NOT NULL,  
    name VARCHAR(10) NOT NULL,  
    birthYear INT NOT NULL,  
    CONSTRAINT PRIMARY KEY PK_userTBL_userID(userID)  
);
```

1 테이블

- ✓ 데이터 무결성을 위한 제약 조건
 - 기본 키(Primary Key) 제약 조건
 - ex) 제품 테이블
 - 기본 키 = 제품코드 + 제품일련번호

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

✓ 데이터 무결성을 위한 제약 조건

- 기본 키(Primary Key) 제약 조건 - 복합키
ex) 제품 테이블

```
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_prodCode_prodID  
        PRIMARY KEY (prodCode, prodID)  
);
```

```
SHOW INDEX FROM prodtbl;
```

	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
▶	prodtbl	0	PRIMARY	1	prodCode	A	0	NULL	NULL		BTREE		
	prodtbl	0	PRIMARY	2	prodID	A	0	NULL	NULL		BTREE		

✓ 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

1 테이블

✓ 데이터 무결성을 위한 제약 조건

○ 외래 키(Foreign Key) 제약 조건

- 외래 키 생성 방법 1
 - CREATE TABLE 끝에 FOREIGN KEY 키워드로 설정

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

```
CREATE TABLE userTBL(  
    userID    CHAR(8) NOT NULL PRIMARY KEY,  
    name      VARCHAR(1) 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)  
);
```

✓ 데이터 무결성을 위한 제약 조건

○ 외래 키(Foreign Key) 제약 조건

- 외래 키 생성 방법 2
 - ALTER TABLE 구문 이용

```
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);
```

1 테이블

✓ 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

```
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  
);
```

1 테이블

✓ 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

-- 출생년도가 1900년 이후 그리고 2023년 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 <= 2023),  
    mobile1   CHAR(3) NULL,  
    CONSTRAINT CK_name CHECK ( name IS NOT NULL)  
);
```

✓ 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 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  
);
```

✓ 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('KAY', '김아영');
```

-- 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	name	birthYear	addr	mobile1	mobile2	height	mDate
▶	KAY	김아영	-1	서울	NULL	NULL	170	NULL
	LHL	이혜리	-1	서울	011	1234567	170	2023-12-12
	WB	원빈	1982	대전	019	9876543	176	2020-05-05

1 테이블

✓ 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

✓ 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;

1 테이블

✓ 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
```

✓ 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;
```

✓ 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