

# Lab #5:

## DDL (Data Definition Language)

소프트웨어학부  
노서영



# **CREATE Statement**

# CREATE 문

- 테이블 구성, 속성과 속성에 관한 제약 정의, 기본키 및 외래키를 정의하는 명령
  - PRIMARY KEY : 기본키를 정할 때 사용
  - FOREIGN KEY : 외래키를 지정할 때 사용
  - ON UPDATE와 ON DELETE : 외래키 속성의 수정과 튜플 삭제 시 동작을 나타냄
- CREATE 문의 기본 문법

```
CREATE TABLE 테이블이름  
( { 속성이름 데이터타입  
    [NOT NULL | UNIQUE | DEFAULT 기본값 | CHECK 체크조건]  
    }  
    [PRIMARY KEY 속성이름(들)]  
    {[FOREIGN KEY 속성이름 REFERENCES 테이블이름(속성이름)]  
        [ON DELETE [CASCADE | SET NULL]]  
    }  
)
```

# CREATE 문

**질의 3-34** 다음과 같은 속성을 가진 NewBook 테이블을 생성하시오. 정수형은 INTEGER를 사용하며 문자형은 가변형 문자타입인 VARCHAR을 사용한다.

- bookid(도서번호)-INTEGER
- bookname(도서이름)-VARCHAR(20)
- publisher(출판사)-VARCHAR(20)
- price(가격)-INTEGER

```
CREATE TABLE      NewBook (  
    bookid          INTEGER,  
    bookname        VARCHAR(20),  
    publisher       VARCHAR(20),  
    price           INTEGER);
```

※ 기본키를 지정하고 싶다면 다음과 같이 생성한다.

```
CREATE TABLE NewBook (  
    bookid          INTEGER,  
    bookname        VARCHAR(20),  
    publisher       VARCHAR(20),  
    price           INTEGER,  
    PRIMARY KEY (bookid));
```

```
CREATE TABLE NewBook (  
    bookid          INTEGER PRIMARY KEY,  
    bookname        VARCHAR(20),  
    publisher       VARCHAR(20),  
    price           INTEGER);
```

# CREATE 문

※ **bookid** 속성이 없어서 두 개의 속성 **bookname**, **publisher**가 기본키가 된다면 괄호를 사용하여 복합키를 지정한다.

```
CREATE TABLE      NewBook (  
    bookname        VARCHAR(20),  
    publisher        VARCHAR(20),  
    price            INTEGER,  
    PRIMARY KEY      (bookname, publisher));
```

※ **NewBook** 테이블의 **CREATE** 문에 좀 더 복잡한 제약사항을 추가한다.

bookname은 NULL 값을 가질 수 없고, publisher는 같은 값이 있으면 안 된다. price에 값이 입력되지 않을 경우 기본 값 10000을 저장한다. 또 가격은 최소 1,000원 이상으로 한다.

```
CREATE TABLE      NewBook (  
    bookname        VARCHAR(20)      NOT NULL,  
    publisher        VARCHAR(20)      UNIQUE,  
    price            INTEGER DEFAULT 10000 CHECK(price > 1000),  
    PRIMARY KEY      (bookname, publisher));
```

# CREATE 문

질의 3-35 다음과 같은 속성을 가진 NewCustomer 테이블을 생성하시오.

- custid(고객번호) - INTEGER, 기본키
- name(이름) - VARCHAR(40)
- address(주소) - VARCHAR(40)
- phone(전화번호) - VARCHAR(30)

```
CREATE TABLE      NewCustomer (  
  custid           INTEGER PRIMARY KEY,  
  name             VARCHAR(40),  
  address          VARCHAR(40),  
  phone            VARCHAR(30) );
```

# CREATE 문

**질의 3-36** 다음과 같은 속성을 가진 **NewOrders** 테이블을 생성하시오.

- orderid(주문번호) - INTEGER, 기본키
- custid(고객번호) - INTEGER, NOT NULL 제약조건, 외래키(NewCustomer.custid, 연쇄삭제)
- bookid(도서번호) - INTEGER, NOT NULL 제약조건
- saleprice(판매가격) - INTEGER
- orderdate(판매일자) - DATE

```
CREATE TABLE      NewOrders (  
  orderid  INTEGER,  
  custid   INTEGER  NOT NULL,  
  bookid   INTEGER  NOT NULL,  
  saleprice INTEGER,  
  orderdate DATE,  
  PRIMARY KEY (orderid),  
  FOREIGN KEY (custid) REFERENCES NewCustomer(custid) ON DELETE CASCADE );
```

# CREATE 문

- 외래키 제약조건을 명시할 때는 반드시 참조되는 테이블(부모 릴레이션)이 존재해야 하며 참조되는 테이블의 기본키여야 함
- 외래키 지정 시 ON DELETE 또는 ON UPDATE 옵션은 참조되는 테이블의 튜플이 삭제되거나 수정될 때 취할 수 있는 동작을 지정함
- NO ACTION은 어떠한 동작도 취하지 않음.

표 3-10 데이터 타입 종류

데이터 타입	설명	ANSI SQL 표준 타입
INTEGER INT	4바이트 정수형	INTEGER, INT SMALLINT
NUMERIC(m,d) DECIMAL(m,d)	전체자리수 m, 소수점이하 자리수 d를 가진 숫자형	DECIMAL(p, s) NUMERIC[(p,s)]
CHAR(n)	문자형 고정길이, 문자를 저장하고 남은 공간은 공백으로 채운다.	CHARACTER(n) CHAR(n)
VARCHAR(n)	문자형 가변길이	CHARACTER VARYING(n) CHAR VARYING(n)
DATE	날짜형, 연도, 월, 날, 시간을 저장한다.	



# **ALTER Statement**

# ALTER 문

- **ALTER 문은 생성된 테이블의 속성과 속성에 관한 제약을 변경하며, 기본키 및 외래키를 변경함**
  - ADD, DROP은 속성을 추가하거나 제거할 때 사용함
  - MODIFY는 속성의 기본값을 설정하거나 삭제할 때 사용함
  - ADD <제약이름>, DROP <제약이름>은 제약사항을 추가하거나 삭제할 때 사용함
- **ALTER 문의 기본 문법**

```
ALTER TABLE 테이블이름  
    [ADD 속성이름 데이터타입]  
    [DROP COLUMN 속성이름]  
    [MODIFY 속성이름 데이터타입]  
    [MODIFY 속성이름 [NULL | NOT NULL]]  
    [ADD PRIMARY KEY(속성이름)]  
    [[ADD | DROP] 제약이름]
```

# ALTER 문

**질의 3-37** NewBook 테이블에 VARCHAR(13)의 자료형을 가진 isbn 속성을 추가하시오.

```
ALTER TABLE NewBook ADD isbn VARCHAR(13);
```

**질의 3-38** NewBook 테이블의 isbn 속성의 데이터 타입을 INTEGER형으로 변경하시오.

```
ALTER TABLE NewBook MODIFY isbn INTEGER;
```

**질의 3-39** NewBook 테이블의 isbn 속성을 삭제하시오.

```
ALTER TABLE NewBook DROP COLUMN isbn;
```

**질의 3-40** NewBook 테이블의 bookid 속성에 NOT NULL 제약조건을 적용하시오.

```
ALTER TABLE NewBook MODIFY bookid INTEGER NOT NULL;
```

**질의 3-41** NewBook 테이블의 bookid 속성을 기본키로 변경하시오.

```
ALTER TABLE NewBook ADD PRIMARY KEY(bookid);
```

# **DROP Statement**

# DROP 문

- **DROP 문은 테이블을 삭제하는 명령**
  - DROP 문은 테이블의 구조와 데이터를 모두 삭제하므로 사용에 주의해야 함
  - 데이터만 삭제하려면 DELETE 문을 사용함
- **DROP문의 기본 문법**

```
DROP TABLE 테이블이름
```

질의 3-42 NewBook 테이블을 삭제하시오.

```
DROP TABLE NewBook;
```

질의 3-43 NewCustomer 테이블을 삭제하시오. 만약 삭제가 거절된다면 원인을 파악하고 관련된 테이블을 같이 삭제하시오(NewOrders 테이블이 NewCustomer를 참조하고 있음).

```
DROP TABLE NewCustomer;
```