

2. 데이터베이스 활용과 클라우드 서버 환경 준비

Section 1 . 데이터베이스 시스템 / 데이터 모델 / SQL 기초 실습

SQL - 조인

두 개 이상 테이블에서 SQL

- Customer 테이블을 Orders 테이블과 조건 없이 연결해보자.
Customer와 Orders 테이블의 합체 결과 튜플의 개수는 고객이 다섯 명이고 주문이 열 개이므로 5×10 해서 50이 된다.

custid	name	address	phone	orderid	custid	bookid	saleprice	orderdate
1	박지성	영국 맨체스타	000-5000-0001	1	1	1	6000	2014-07-01
2	김연아	대한민국 서울	000-6000-0001	1	1	1	6000	2014-07-01
3	장미란	대한민국 강원도	000-7000-0001	1	1	1	6000	2014-07-01
4	추신수	미국 클리블랜드	000-8000-0001	1	1	1	6000	2014-07-01
5	박세리	대한민국 대전	NULL	1	1	1	6000	2014-07-01
1	박지성	영국 맨체스타	000-5000-0001	2	1	3	21000	2014-07-03
2	김연아	대한민국 서울	000-6000-0001	2	1	3	21000	2014-07-03
3	장미란	대한민국 강원도	000-7000-0001	2	1	3	21000	2014-07-03
4	추신수	미국 클리블랜드	000-8000-0001	2	1	3	21000	2014-07-03
5	박세리	대한민국 대전	NULL	2	1	3	21000	2014-07-03
1	박지성	영국 맨체스타	000-5000-0001	3	2	5	8000	2014-07-03
2	김연아	대한민국 서울	000-6000-0001	3	2	5	8000	2014-07-03
3	장미란	대한민국 강원도	000-7000-0001	3	2	5	8000	2014-07-03
4	추신수	미국 클리블랜드	000-8000-0001	3	2	5	8000	2014-07-03
5	박세리	대한민국 대전	NULL	3	2	5	8000	2014-07-03
1	박지성	영국 맨체스타	000-5000-0001	4	3	6	6000	2014-07-04
2	김연아	대한민국 서울	000-6000-0001	4	3	6	6000	2014-07-04
3	장미란	대한민국 강원도	000-7000-0001	4	3	6	6000	2014-07-04
4	추신수	미국 클리블랜드	000-8000-0001	4	3	6	6000	2014-07-04
... 중략 ...								
1	박지성	영국 맨체스타	000-5000-0001	10	3	8	13000	2014-07-10
2	김연아	대한민국 서울	000-6000-0001	10	3	8	13000	2014-07-10
3	장미란	대한민국 강원도	000-7000-0001	10	3	8	13000	2014-07-10
4	추신수	미국 클리블랜드	000-8000-0001	10	3	8	13000	2014-07-10
5	박세리	대한민국 대전	NULL	10	3	8	13000	2014-07-10

Customer와 Orders 테이블의 합체

두 개 이상 테이블에서 SQL

❖ 조인_2개의 테이블을 합쳐해보자

고객과 고객의 주문에 관한 데이터를 모두 보이시오.

custid	name	address	phone	orderid	custid	bookid	saleprice	orderdate
1	박지성	영국 맨체스터	000-5000-0001	1	1	1	6000	2014-07-01
1	박지성	영국 맨체스터	000-5000-0001	2	1	3	21000	2014-07-03
2	김연아	대한민국 서울	000-6000-0001	3	2	5	8000	2014-07-03
3	장미란	대한민국 강원도	000-7000-0001	4	3	6	6000	2014-07-04
4	추신수	미국 클리블랜드	000-8000-0001	5	4	7	20000	2014-07-05
1	박지성	영국 맨체스터	000-5000-0001	6	1	2	12000	2014-07-07
4	추신수	미국 클리블랜드	000-8000-0001	7	4	8	13000	2014-07-07
3	장미란	대한민국 강원도	000-7000-0001	8	3	10	12000	2014-07-08
2	김연아	대한민국 서울	000-6000-0001	9	2	10	7000	2014-07-09
3	장미란	대한민국 강원도	000-7000-0001	10	3	8	13000	2014-07-10

두 개 이상 테이블에서 SQL

❖ 조인_2개의 테이블을 합쳐해보자

고객과 고객의 주문에 관한 데이터를 고객번호 순으로 정렬하여 보이시오.

custid	name	address	phone	orderid	custid	bookid	saleprice	orderdate
1	박지성	영국 맨체스타	000-5000-0001	6	1	2	12000	2014-07-07
1	박지성	영국 맨체스타	000-5000-0001	1	1	1	6000	2014-07-01
1	박지성	영국 맨체스타	000-5000-0001	2	1	3	21000	2014-07-03
2	김연아	대한민국 서울	000-6000-0001	3	2	5	8000	2014-07-03
2	김연아	대한민국 서울	000-6000-0001	9	2	10	7000	2014-07-09
3	장미란	대한민국 강원도	000-7000-0001	4	3	6	6000	2014-07-04
3	장미란	대한민국 강원도	000-7000-0001	8	3	10	12000	2014-07-08
3	장미란	대한민국 강원도	000-7000-0001	10	3	8	13000	2014-07-10
4	추신수	미국 클리블랜드	000-8000-0001	5	4	7	20000	2014-07-05
4	추신수	미국 클리블랜드	000-8000-0001	7	4	8	13000	2014-07-07

두 개 이상 테이블에서 SQL

❖ 조인_2개의 테이블을 합체해보자

고객의 이름과 고객이 주문한 도서의 판매가격을 검색하시오.

name	saleprice
박지성	6000
박지성	21000
김연아	8000
장미란	6000
추신수	20000
박지성	12000
추신수	13000
장미란	12000
김연아	7000
장미란	13000

두 개 이상 테이블에서 SQL

❖ 조인_2개의 테이블을 합체해보자

고객별로 주문한 모든 도서의 총 판매액을 구하고, 고객별로 정렬하시오.

name	SUM(saleprice)
김연아	15000
박지성	39000
장미란	31000
추신수	33000

두 개 이상 테이블에서 SQL

❖ 조인_2개의 테이블을 합체해보자

고객의 이름과 고객이 주문한 도서의 이름을 구하시오.

CUSTID	NAME	ADDRESS	PHONE
1	박지성	영국 맨체스터	000-5000-0001
2	김연아	대한민국 서울	000-6000-0001
3	장미란	대한민국 강원도	000-7000-0001
4	추신수	미국 캘리포니아	000-8000-0001
5	박세리	대한민국 대전	{null}

BOOKID	BOOKNAME	PUBLISHER	PRICE
1	축구의 역사	굿스포츠	7000
2	축구하는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000
6	역도 단계별기술	굿스포츠	6000
7	야구의 추억	이상미디어	20000
8	야구를 부탁해	이상미디어	13000
9	올림픽 이야기	삼성당	7500
10	Olympic Champions	Pearson	13000

ORDERID	CUSTID	BOOKID	SALEPRICE	ORDERDATE
1	1	1	6000	14/07/01
2	1	3	21000	14/07/03
3	2	5	8000	14/07/03
4	3	6	6000	14/07/04
5	4	7	20000	14/07/05
6	1	2	12000	14/07/07
7	4	8	13000	14/07/07
8	3	10	12000	14/07/08
9	2	10	7000	14/07/09
10	3	8	13000	14/07/10

서점 데이터 간의 연결

두 개 이상 테이블에서 SQL

❖ 조인_2개의 테이블을 합체해보자

고객의 이름과 고객이 주문한 도서의 이름을 구하시오.

name	bookname
박지성	축구의 역사
박지성	축구의 이해
김연아	피겨 교본
장미란	역도 단계별기술
추신수	야구의 추억
박지성	축구하는 여자
추신수	야구를 부탁해
장미란	Olympic Champions
김연아	Olympic Champions
장미란	역도를 부탁해

두 개 이상 테이블에서 SQL

❖ 조인_2개의 테이블을 합체해보자

가격이 20,000원인 도서를 주문한 고객의 이름과 도서의 이름을 구하시오.

name	bookname
추신수	야구의 추억

두 개 이상 테이블에서 SQL

❖ 조인_2개의 테이블을 합체해보자

■ 외부조인

도서를 구매하지 않은 고객을 포함하여 고객의 이름과 고객이 주문한 도서의 판매가격을 구하시오.

name	saleprice
박지성	6000
박지성	21000
김연아	8000
장미란	6000
추신수	20000
박지성	12000
추신수	13000
장미란	12000
김연아	7000
장미란	13000
박세리	NULL

두 개 이상 테이블에서 SQL

❖ 조인_2개의 테이블을 합체해보자

조인 문법

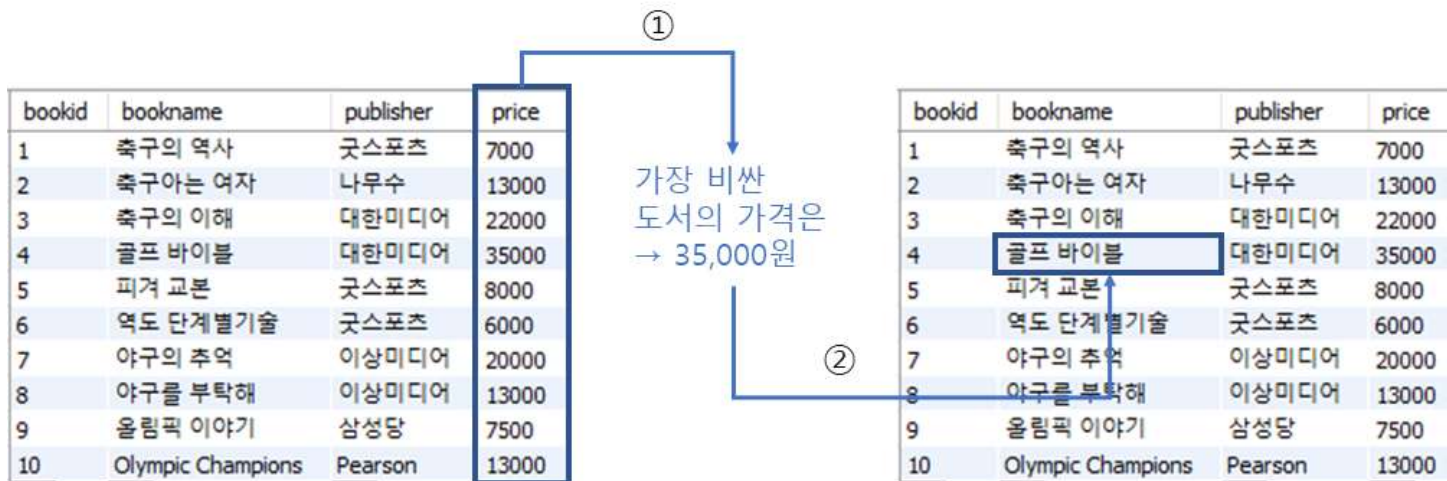
명령	문법	설명
일반 조인	SELECT <속성들> FROM 테이블1, 테이블2 WHERE <조인조건> AND <검색조건>	SQL 문에서는 주로 동등조인을 사용함. 두 가지 문법 중 하나를 사용할 수 있음.
	SELECT <속성들> FROM 테이블1 INNER JOIN 테이블2 ON <조인조건> WHERE <검색조건>	
외부조인	SELECT <속성들> FROM 테이블1 {LEFT RIGHT FULL [OUTER]} JOIN 테이블2 ON <조인조건> WHERE <검색조건>	외부조인은 FROM 절에 조인 종류를 적고 ON을 이용하여 조인조건을 명시함.

두 개 이상 테이블에서 SQL

❖ 부속 SQL 문 내에 또 다른 SQL 문을 작성해보자

가장 비싼 도서의 이름을 보이시오.

bookname
골프 바이블



부속의 실행 순서

두 개 이상 테이블에서 SQL

❖ 부속_SQL 문 내에 또 다른 SQL 문을 작성해보자

도서를 구매한 적이 있는 고객의 이름을 검색하시오.

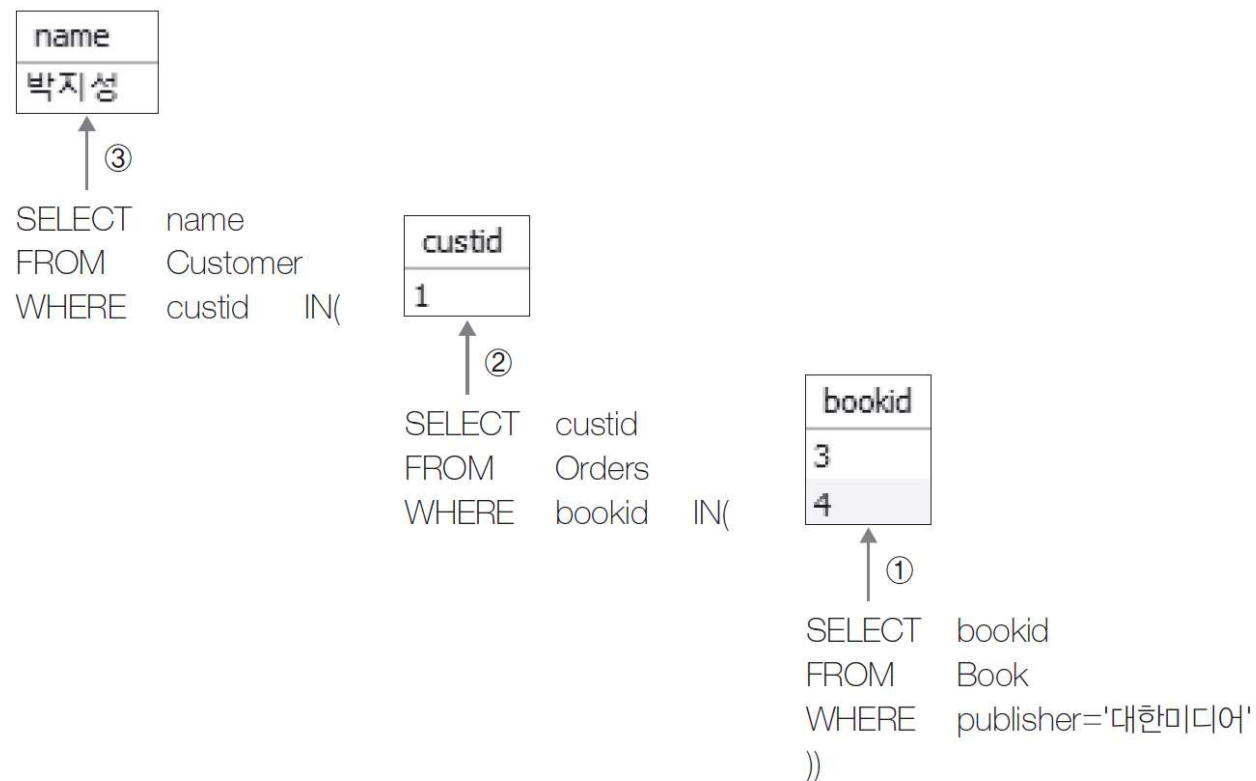
name
박지성
김연아
장미란
추신수

대한미디어에서 출판한 도서를 구매한 고객의 이름을 보이시오.

name
박지성

두 개 이상 테이블에서 SQL

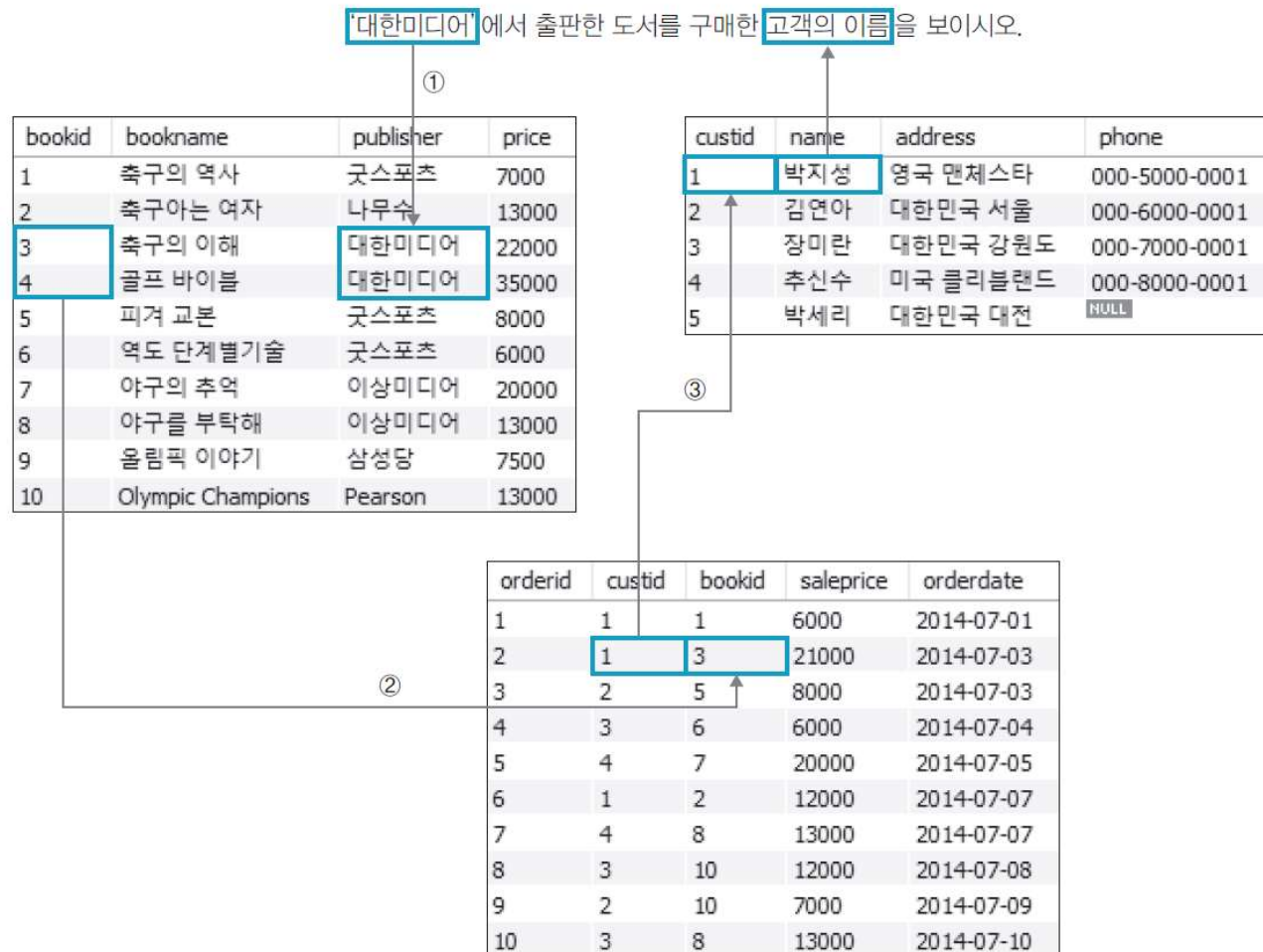
❖ **부속_SQL** 문 내에 또 다른 SQL 문을 작성해보자



3단계 부속의 실행 순서

두 개 이상 테이블에서 SQL

❖ 부속 SQL 문 내에 또 다른 SQL 문을 작성해보자



3단계 부속의 실행 순서와 데이터 예

두 개 이상 테이블에서 SQL

❖ 부속_SQL 문 내에 또 다른 SQL 문을 작성해보자

- 상관 부속(correlated subquery)는 상위 부속의 튜플을 이용하여 하위 부속를 계산함. 즉 상위 부속와 하위 부속가 독립적이지 않고 서로 관련을 맺고 있음.

출판사별로 출판사의 평균 도서 가격보다 비싼 도서를 구하시오.

bookname
골프 바이블
피겨 교본
야구의 추억

두 개 이상 테이블에서 SQL

❖ 부속 SQL 문 내에 또 다른 SQL 문을 작성해보자

상관 부속의 데이터 예

Book 테이블 : b1으로 나타냄

bookid	bookname	publisher	price
1	축구의 역사	굿스포츠	7000
2	축구하는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000
6	역도 단계별기술	굿스포츠	6000
7	야구의 추억	이상미디어	20000
8	야구를 부탁해	이상미디어	13000
9	올림픽 이야기	삼성당	7500
10	Olympic Champions	Pearson	13000

b1 테이블의
특정 t에
해당하는
출판사를
b2 테이블로
가져가서,
같은 출판사를
가진 튜플들의
price 평균을
구한다.

Book 테이블 : b2로 나타냄

bookid	bookname	publisher	price
1	축구의 역사	굿스포츠	7000
2	축구하는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000
6	역도 단계별기술	굿스포츠	6000
7	야구의 추억	이상미디어	20000
8	야구를 부탁해	이상미디어	13000
9	올림픽 이야기	삼성당	7500
10	Olympic Champions	Pearson	13000

avg(b2.price)
28500

b1.price > avg(b2.price)

두 개 이상 테이블에서 SQL

❖ 집합 연산_도서를 주문한 고객을 알고 싶다

■ 합집합 UNION, 차집합 MINUS, 교집합 INTERSECT

대한민국에서 거주하는 고객의 이름과 도서를 주문한 고객의 이름을 보이시오.

{고객 이름} = {대한민국에 거주하는 고객 이름} ∪ {도서를 주문한 고객 이름}

name
김연아
장미란
박세리
박지성
추신수

두 개 이상 테이블에서 SQL

■ MINUS, INTERSECT 연산자

MySQL에는 MINUS, INTERSECT 연산자가 없으므로 다음과 같이 표현한다.

MINUS 연산을 수행한 "대한민국에서 거주하는 고객의 이름에서 도서를 주문한 고객의 이름 빼고 보이시오." 를 NOT IN 연산자를 사용

INTERSECT 연산을 수행한 "대한민국에서 거주하는 고객 중 도서를 주문한 고객의 이름 보이시오." 를 IN 연산자를 사용

두 개 이상 테이블에서 SQL

❖ EXISTS_주문이 있는 고객을 알고 싶다

- EXISTS는 원래 단어에서 의미하는 것과 같이 조건에 맞는 튜플이 존재하면 결과에 포함시킴. 즉 부속문의 어떤 행이 조건에 만족하면 참임.
- NOT EXISTS는 부속문의 모든 행이 조건에 만족하지 않을 때만 참임.

주문이 있는 고객의 이름과 주소를 보이시오.

name	address
박지성	영국 맨체스터
김연아	대한민국 서울
장미란	대한민국 강원도
추신수	미국 클리블랜드

두 개 이상 테이블에서 SQL

❖ EXISTS_주문이 있는 고객을 알고 싶다

Customer

custid	name	address	phone
1	박지성	영국 맨체스터	000-5000-0001
2	김연아	대한민국 서울	000-6000-0001
3	장미란	대한민국 강원도	000-7000-0001
4	추신수	미국 클리블랜드	000-8000-0001
5	박세리	대한민국 대전	NULL

Orders

orderid	custid	bookid	saleprice	orderdate	
1	1	1	6000	2014-07-01	true
2	1	3	21000	2014-07-03	
3	2	5	8000	2014-07-03	true
4	3	6	6000	2014-07-04	true
5	4	7	20000	2014-07-05	true
6	1	2	12000	2014-07-07	
7	4	8	13000	2014-07-07	
8	3	10	12000	2014-07-08	
9	2	10	7000	2014-07-09	
10	3	8	13000	2014-07-10	

①
②
③
④
⑤
x
없음
false

EXISTS 상관 부속문 데이터 예

데이터 정의어

1. CREATE 문
2. ALTER 문
3. DROP 문

1. 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]
}
)
```


1. CREATE 문

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);
```

1. 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));
```

※ 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));
```

1. CREATE 문

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) );
```

1. CREATE 문

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 );
```

1. 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.

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.	

2. 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]
```

2. ALTER 문

NewBook 테이블에 VARCHAR(13)의 자료형을 가진 isbn 속성을 추가하시오.

NewBook 테이블의 isbn 속성의 데이터 타입을 INTEGER형으로 변경하시오.

NewBook 테이블의 isbn 속성을 삭제하시오.

NewBook 테이블의 bookid 속성에 NOT NULL 제약조건을 적용하시오.

NewBook 테이블의 bookid 속성을 기본키로 변경하시오.

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
```

NewBook 테이블을 삭제하시오.

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

데이터 조작용어 – 삽입, 수정, 삭제

1. INSERT 문
2. UPDATE 문
3. DELETE 문

1. INSERT 문

- INSERT 문은 테이블에 새로운 튜플을 삽입하는 명령임.
- INSERT 문의 기본 문법

```
INSERT INTO 테이블이름[(속성리스트)]  
VALUES (값리스트);
```

Book 테이블에 새로운 도서 '스포츠 의학'을 삽입하시오. 스포츠 의학은 한솔의학서적에서 출간했으며 가격은 90,000원이다.

bookid	bookname	publisher	price
1	축구의 역사	굿스포츠	7000
2	축구하는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000
6	역도 단계별기술	굿스포츠	6000
7	야구의 추억	이상미디어	20000
8	야구를 부탁해	이상미디어	13000
9	올림픽 이야기	삼성당	7500
10	Olympic Champions	Pearson	13000
11	스포츠 의학	한솔의학...	90000

1. INSERT 문

Book 테이블에 새로운 도서 '스포츠 의학'을 삽입하시오. 스포츠 의학은 한솔의학서적에서 출간했으며 가격은 미정이다.

bookid	bookname	publisher	price
1	축구의 역사	굿스포츠	7000
2	축구하는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000
6	역도 단계별기술	굿스포츠	6000
7	야구의 추억	이상미디어	20000
8	야구를 부탁해	이상미디어	13000
9	올림픽 이야기	삼성당	7500
10	Olympic Champions	Pearson	13000
11	스포츠 의학	한솔의학...	90000
12	스포츠 의학	한솔의학...	90000
13	스포츠 의학	한솔의학...	90000
14	스포츠 의학	한솔의학...	NULL

1. INSERT 문

- 대량 삽입(bulk insert)이란 한꺼번에 여러 개의 튜플을 삽입하는 방법임.

수입도서 목록(Imported_book)을 Book 테이블에 모두 삽입하시오.

(Imported_book 테이블은 스크립트 Book 테이블과 같이 이미 만들어져 있음)

bookid	bookname	publisher	price
1	축구의 역사	굿스포츠	7000
2	축구하는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000
6	역도 단계별기술	굿스포츠	6000
7	야구의 추억	이상미디어	20000
8	야구를 부탁해	이상미디어	13000
9	올림픽 이야기	삼성당	7500
10	Olympic Champions	Pearson	13000
11	스포츠 의학	한솔의학...	90000
12	스포츠 의학	한솔의학...	90000
13	스포츠 의학	한솔의학...	90000
14	스포츠 의학	한솔의학...	NULL
21	Zen Golf	Pearson	12000
22	Soccer Skills	Human Kin...	15000

2. UPDATE 문

- UPDATE 문은 특정 속성 값을 수정하는 명령이다.
- UPDATE 문의 기본 문법

```
UPDATE 테이블이름  
SET    속성이름1=값1[, 속성이름2=값2, ...]  
[WHERE <검색조건>];
```

Customer 테이블에서 고객번호가 5인 고객의 주소를 '대한민국 부산'으로 변경하시오.

custid	name	address	phone
1	박지성	영국 맨체스터	000-5000-0001
2	김연아	대한민국 서울	000-6000-0001
3	장미란	대한민국 강원도	000-7000-0001
4	추신수	미국 클리블랜드	000-8000-0001
5	박세리	대한민국 부산	NULL

2. UPDATE 문

Book 테이블에서 14번 '스포츠 의학'의 출판사를 imported_book 테이블의 21번 책의 출판사와 동일하게 변경하시오.

bookid	bookname	publisher	price
1	축구의 역사	굿스포츠	7000
2	축구하는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000
6	역도 단계별기술	굿스포츠	6000
7	야구의 추억	이상미디어	20000
8	야구를 부탁해	이상미디어	13000
9	올림픽 이야기	삼성당	7500
10	Olympic Champions	Pearson	13000
11	스포츠 의학	한솔의학...	90000
12	스포츠 의학	한솔의학...	90000
13	스포츠 의학	한솔의학...	90000
14	스포츠 의학	Pearson	NULL
21	Zen Golf	Pearson	12000
22	Soccer Skills	Human Kin...	15000

DELETE 문

- DELETE 문은 테이블에 있는 기존 튜플을 삭제하는 명령
- DELETE 문의 기본 문법

```
DELETE FROM 테이블이름  
[WHERE 검색조건];
```

DELETE 문

Book 테이블에서 도서번호가 11인 도서를 삭제하시오.

bookid	bookname	publisher	price
1	축구의 역사	굿스포츠	7000
2	축구아는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000
6	역도 단계별기술	굿스포츠	6000
7	야구의 추억	이상미디어	20000
8	야구를 부탁해	이상미디어	13000
9	올림픽 이야기	삼성당	7500
10	Olympic Champions	Pearson	13000
12	스포츠 의학	한솔의학...	90000
13	스포츠 의학	한솔의학...	90000
14	스포츠 의학	Pearson	NULL
21	Zen Golf	Pearson	12000
22	Soccer Skills	Human Kin...	15000

모든 고객을 삭제하시오.