

데이터 입출력 구현

물리 데이터 저장소 설계 및
데이터 프로시저 작성하기

수행 순서

1. 논리, 물리 데이터 모델링
2. 데이터 모델링으로 생성한 테이블에 데이터 추가 및 개체 관계에 따른 무결성 방안 설명

1) 논리, 물리 데이터 모델링

- **개요**

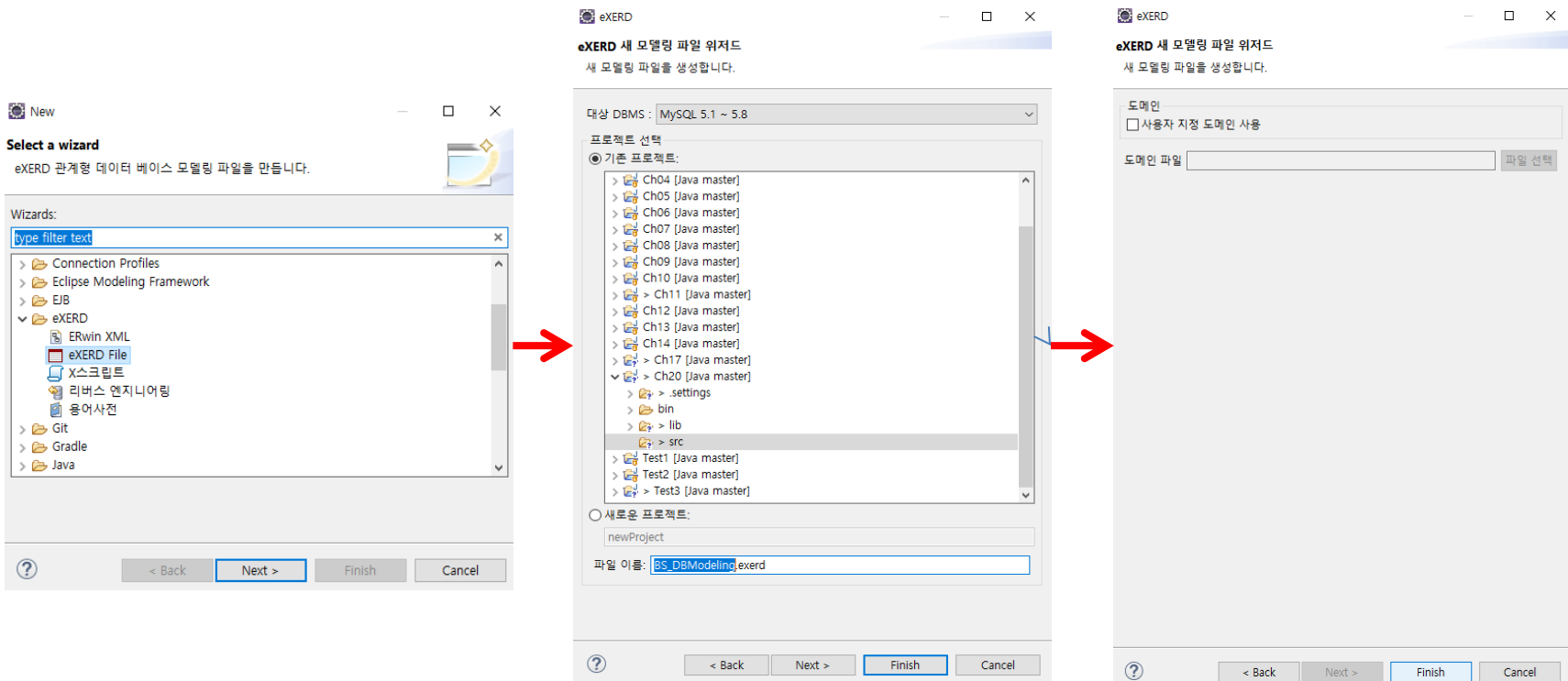
OO서점은 도서관리 시스템을 구축하여 도서관리를 효율적이고 체계적으로 관리한다.

- **요구사항**

1. 도서 목록에는 도서번호, 도서이름, 출판사이름, 도서단가를 기록한다.
2. 출판사 목록에는 출판사이름, 담당자이름, 전화번호를 기록한다.
3. 고객 목록에는 고객번호, 고객이름, 주소, 전화번호를 기록한다.
4. OO서점은 출판사에서 공급한 도서만 등록하여 관리한다. (출판사와 도서의 관계는 1:N)
5. 고객은 여러 권의 도서를 구입할 수 있다. (고객과 도서의 관계는 N:N)
6. 주문 목록은 고객이 도서를 구입한 날(주문날짜)과 구매한 가격(주문금액)을 따로 저장한다.

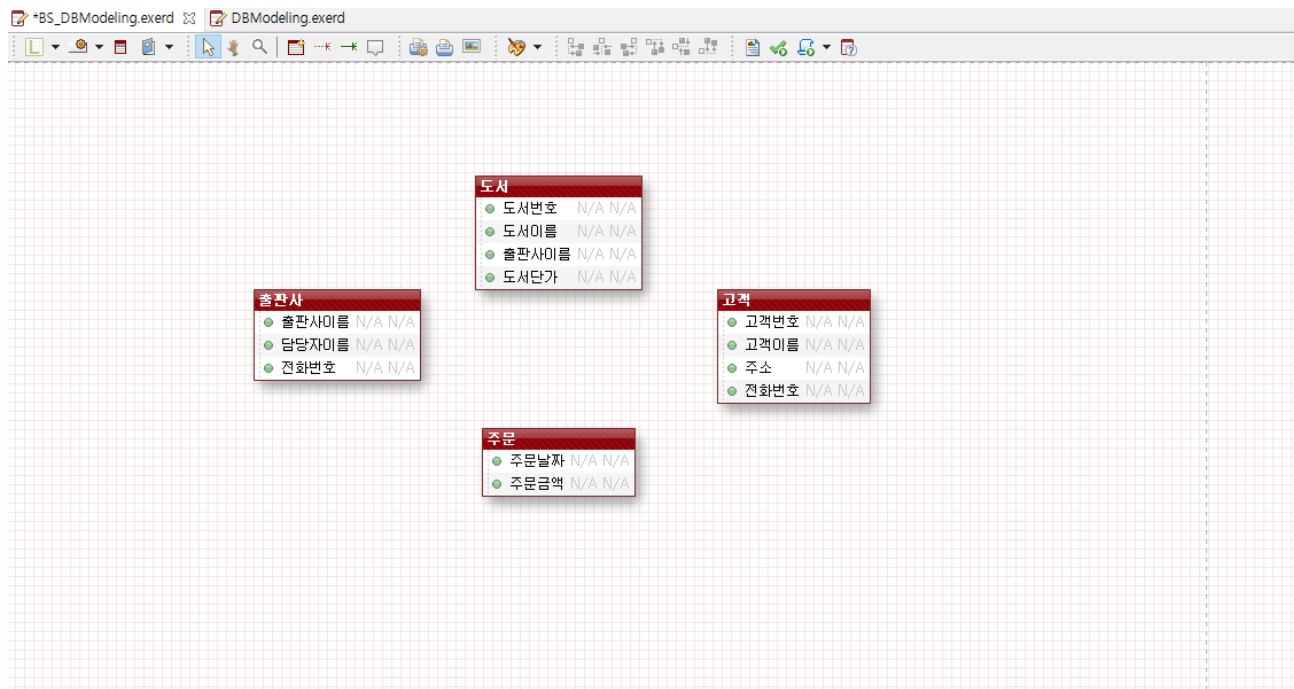
1) 논리, 물리 데이터 모델링

- 다음과 같이 eXERD 프로그램을 통해 DB모델을 생성한다.



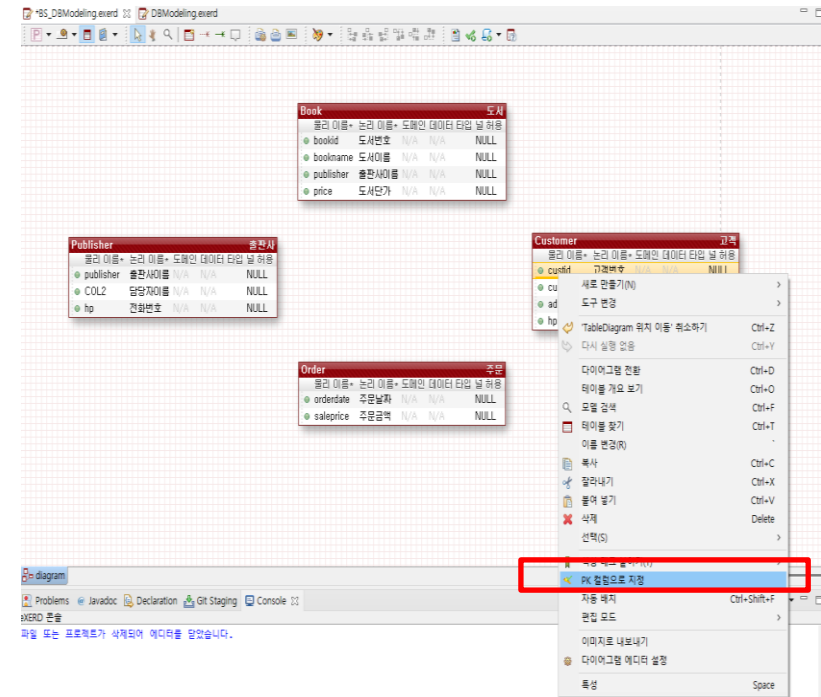
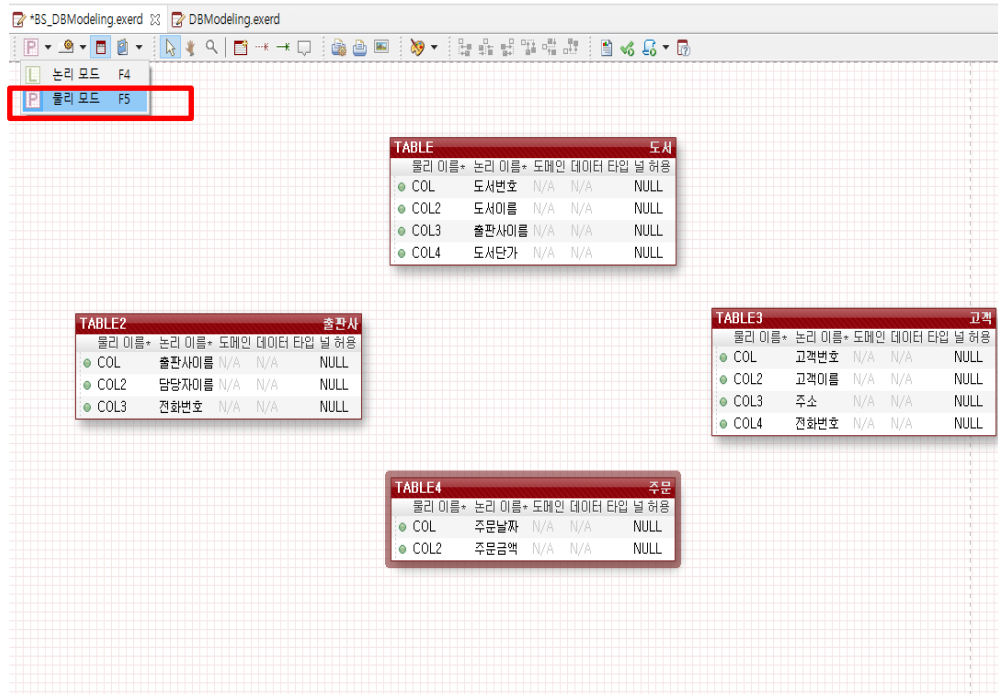
1) 논리, 물리 데이터 모델링

- 도서 목록에는 도서번호, 도서이름, 출판사이름, 도서단가를 기록한다.
- 출판사 목록에는 출판사이름, 담당자이름, 전화번호를 기록한다.
- 고객 목록에는 고객번호, 고객이름, 주소, 전화번호를 기록한다.
- 주문 목록은 고객이 도서를 구입한 날(주문날짜)과 구매한 가격(주문금액)을 따로 저장한다.



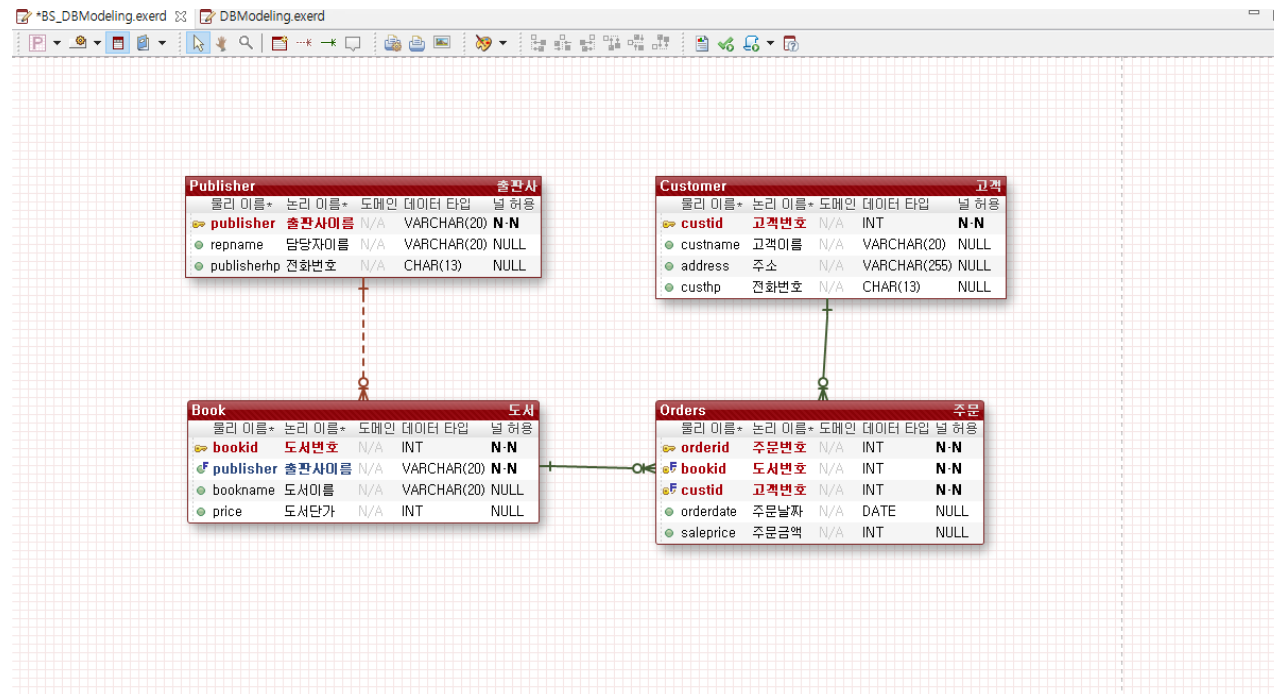
1) 논리, 물리 데이터 모델링

- 논리 모드에서 물리 모드로 변경 후 물리 이름과 데이터 타입을 편집 후 각 테이블에 특정 튜플을 유일하게 구별 할 수 있는 속성인 기본 키(Primary Key)를 지정해준다.



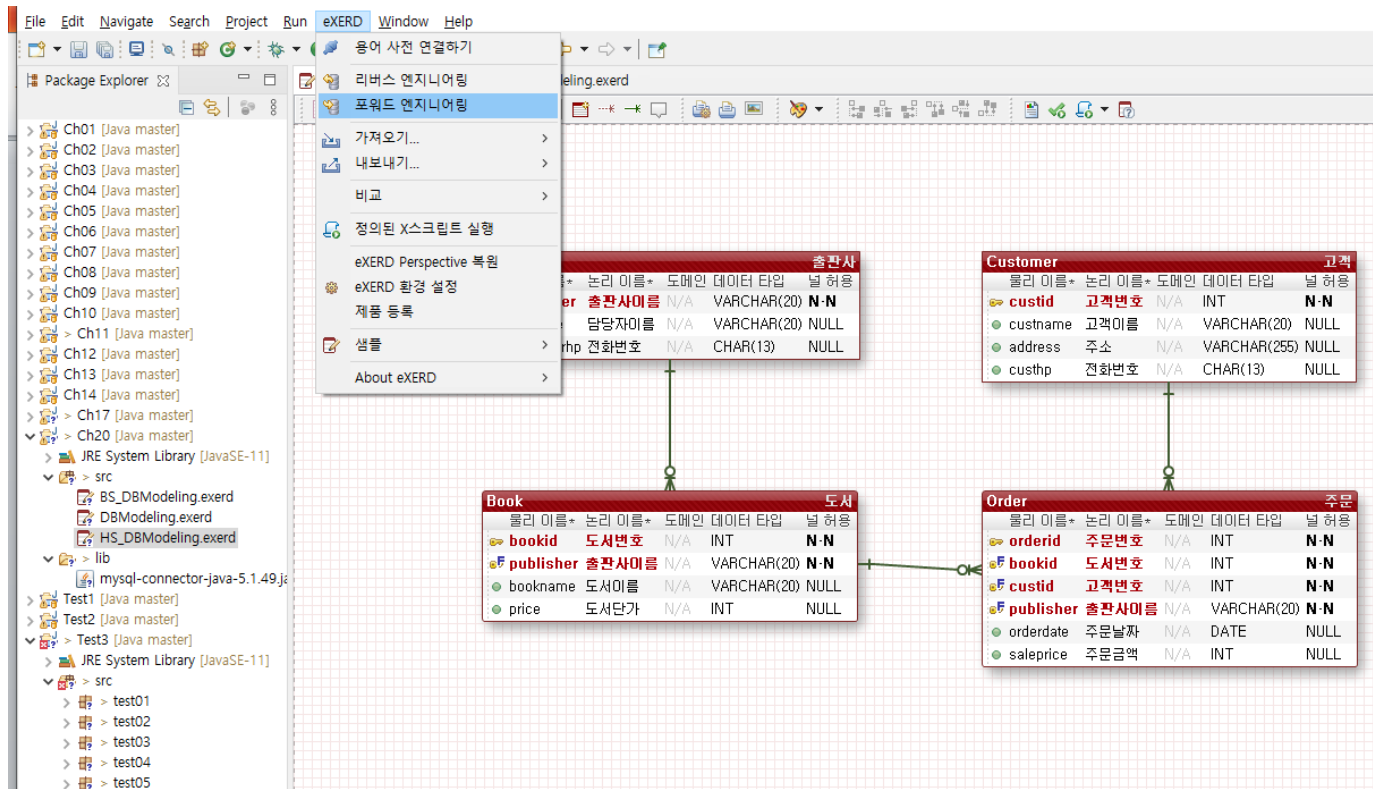
1) 논리, 물리 데이터 모델링

- 식별/비식별 관계를 지정해준다.



1) 논리, 물리 데이터 모델링

- eXERD 탭에서 포워드 엔지니어링을 통해 완성된 DB모델을 데이터 베이스 프로그램에 모델을 자동으로 등록해준다.



2) 데이터 모델링으로 생성한 테이블 데이터 추가

- 물리 데이터 모델을 기반으로 아래와 같이 HeidiSQL에서 테이블이 생성된 것을 찾아볼 수 있다.

The screenshot shows the HeidiSQL interface with the 'lse' database selected. The 'Book' table is highlighted in the left sidebar. The table structure is displayed in the main window:

#	이름	데이터 유형	길이/설정	부호	NULL	0...	기본값	코멘트	조합
1	bookid	INT	11				기본값 없음	도서번호	
2	publisher	VARCHAR	20				기본값 없음	출판사이름	utf8_unicode_ci
3	bookname	VARCHAR	20				NULL	도서이름	utf8_unicode_ci
4	price	INT	11				NULL	도서단가	

The screenshot shows the HeidiSQL interface with the 'lse' database selected. The 'Order' table is highlighted in the left sidebar. The table structure is displayed in the main window:

#	이름	데이터 유형	길이/설정	부호	NULL	0...	기본값	코멘트	조합
1	orderid	INT	11				기본값 없음	주문번호	
2	bookid	INT	11				기본값 없음	도서번호	
3	custid	INT	11				기본값 없음	고객번호	
4	publisher	VARCHAR	20				기본값 없음	출판사이름	utf8_unicode_ci
5	orderdate	DATE					NULL	주문날짜	
6	saleprice	INT	11				NULL	주문금액	

The screenshot shows the HeidiSQL interface with the 'lse' database selected. The 'Customer' table is highlighted in the left sidebar. The table structure is displayed in the main window:

#	이름	데이터 유형	길이/설정	부호	NULL	0...	기본값	코멘트	조합
1	custid	INT	11				기본값 없음	고객번호	
2	custname	VARCHAR	20				NULL	고객이름	utf8_unicode_ci
3	address	VARCHAR	255				NULL	주소	utf8_unicode_ci
4	custhp	CHAR	13				NULL	전화번호	utf8_unicode_ci

The screenshot shows the HeidiSQL interface with the 'lse' database selected. The 'Publisher' table is highlighted in the left sidebar. The table structure is displayed in the main window:

#	이름	데이터 유형	길이/설정	부호	NULL	0...	기본값	코멘트	조합
1	publisher	VARCHAR	20				기본값 없음	출판사이름	utf8_unicode_ci
2	repname	VARCHAR	20				NULL	담당자이름	utf8_unicode_ci
3	publisherhp	CHAR	13				NULL	전화번호	utf8_unicode_ci

1) 논리, 물리 데이터 모델링

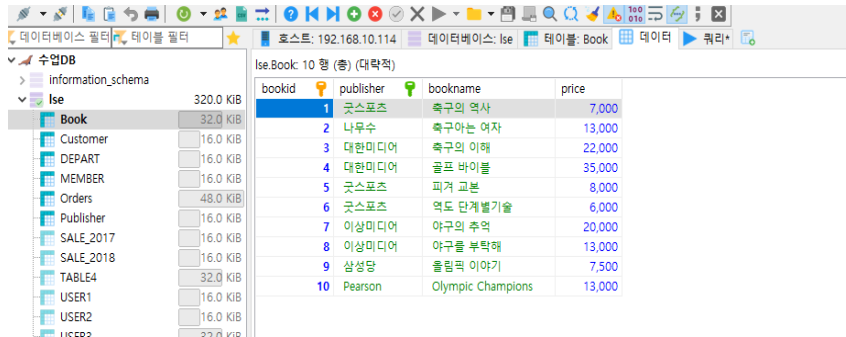
- 생성된 테이블에 다음과 같이 데이터를 추가한다.

```
호스트: 192.168.10.114  데이터베이스: lse  테이블: Orders  데이터  쿼리*

1  #날 짜 : 2020/11/20
2  #이 름 : 이 승 이
3  #내 용 : 데 이 터 입 출 력 구 현 - 수 령 평 가
4
5  INSERT INTO `Publisher` (`publisher`, `repname`, `publisherhp`) VALUES ('굿스포츠', '김유신', '010-1234-1111');
6  INSERT INTO `Publisher` (`publisher`, `repname`, `publisherhp`) VALUES ('나무수', '김춘추', '010-1234-2222');
7  INSERT INTO `Publisher` (`publisher`, `repname`, `publisherhp`) VALUES ('대한미디어', '장보고', '010-1234-3333');
8  INSERT INTO `Publisher` (`publisher`, `repname`, `publisherhp`) VALUES ('이상미디어', '강감찬', '010-1234-4444');
9  INSERT INTO `Publisher` (`publisher`, `repname`, `publisherhp`) VALUES ('삼성당', '이성계', '010-1234-5555');
10 INSERT INTO `Publisher` (`publisher`, `repname`, `publisherhp`) VALUES ('Pearson', '이순신', '010-1234-5555');
11
12 INSERT INTO `Book` (`publisher`, `bookname`, `price`) VALUES ('굿스포츠', '축구의 역사', 7000);
13 INSERT INTO `Book` (`publisher`, `bookname`, `price`) VALUES ('나무수', '축구하는 여자', 13000);
14 INSERT INTO `Book` (`publisher`, `bookname`, `price`) VALUES ('대한미디어', '축구의 이해', 22000);
15 INSERT INTO `Book` (`publisher`, `bookname`, `price`) VALUES ('대한미디어', '골프 바이블', 35000);
16 INSERT INTO `Book` (`publisher`, `bookname`, `price`) VALUES ('굿스포츠', '피겨 교본', 8000);
17 INSERT INTO `Book` (`publisher`, `bookname`, `price`) VALUES ('굿스포츠', '역도 단계별 기술', 6000);
18 INSERT INTO `Book` (`publisher`, `bookname`, `price`) VALUES ('이상미디어', '야구의 추억', 20000);
19 INSERT INTO `Book` (`publisher`, `bookname`, `price`) VALUES ('이상미디어', '야구를 부탁해', 13000);
20 INSERT INTO `Book` (`publisher`, `bookname`, `price`) VALUES ('삼성당', '올림픽 이야기', 7500);
21 INSERT INTO `Book` (`publisher`, `bookname`, `price`) VALUES ('Pearson', 'Olympic Champions', 13000);
22
23 INSERT INTO `Customer` (`custname`, `address`, `custhp`) VALUES ('박지성', '영국 맨체스터', '010-5000-0001');
24 INSERT INTO `Customer` (`custname`, `address`, `custhp`) VALUES ('김연아', '대한민국 서울', '010-6000-0001');
25 INSERT INTO `Customer` (`custname`, `address`, `custhp`) VALUES ('장미란', '대한민국 강원도', '010-7000-0001');
26 INSERT INTO `Customer` (`custname`, `address`, `custhp`) VALUES ('추신수', '미국 클리블랜드', '010-8000-0001');
27 INSERT INTO `Customer` (`custname`, `address`) VALUES ('박세리', '대한민국 대전');
28
29 INSERT INTO `Orders` (`custid`, `bookid`, `saleprice`, `orderdate`) VALUES (1, 1, 6000, '2014-07-01');
30 INSERT INTO `Orders` (`custid`, `bookid`, `saleprice`, `orderdate`) VALUES (1, 3, 21000, '2014-07-03');
31 INSERT INTO `Orders` (`custid`, `bookid`, `saleprice`, `orderdate`) VALUES (2, 5, 8000, '2014-07-03');
32 INSERT INTO `Orders` (`custid`, `bookid`, `saleprice`, `orderdate`) VALUES (3, 6, 6000, '2014-07-04');
33 INSERT INTO `Orders` (`custid`, `bookid`, `saleprice`, `orderdate`) VALUES (4, 7, 20000, '2014-07-05');
34 INSERT INTO `Orders` (`custid`, `bookid`, `saleprice`, `orderdate`) VALUES (1, 2, 12000, '2014-07-07');
35 INSERT INTO `Orders` (`custid`, `bookid`, `saleprice`, `orderdate`) VALUES (4, 8, 13000, '2014-07-07');
36 INSERT INTO `Orders` (`custid`, `bookid`, `saleprice`, `orderdate`) VALUES (3, 10, 12000, '2014-07-08');
37 INSERT INTO `Orders` (`custid`, `bookid`, `saleprice`, `orderdate`) VALUES (2, 10, 7000, '2014-07-09');
38 INSERT INTO `Orders` (`custid`, `bookid`, `saleprice`, `orderdate`) VALUES (3, 8, 13000, '2014-07-10');
39
```

2) 데이터 모델링으로 생성한 테이블 데이터 추가

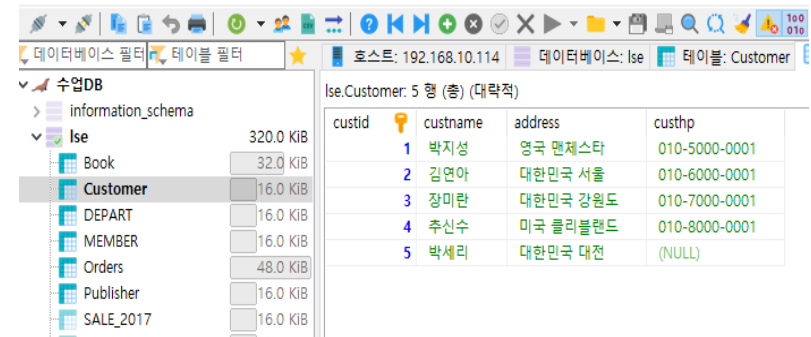
- 데이터를 입력한 결과 다음과 같이 데이터가 생성된다.



Host: 192.168.10.114 | Database: Ise | Table: Book

Ise.Book: 10 행 (총) (대략적)

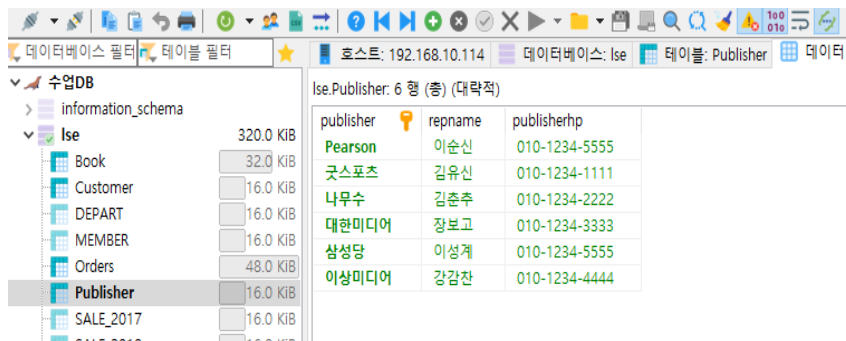
bookid	publisher	bookname	price
1	굿스포츠	축구 역사	7,000
2	나무수	축구는 여자	13,000
3	대한미디어	축구의 이해	22,000
4	대한미디어	골프 바이블	35,000
5	굿스포츠	피겨 교본	8,000
6	굿스포츠	역도 단계별기술	6,000
7	이상미디어	야구의 추억	20,000
8	이상미디어	야구를 부탁해	13,000
9	삼성당	올림픽 이야기	7,500
10	Pearson	Olympic Champions	13,000



Host: 192.168.10.114 | Database: Ise | Table: Customer

Ise.Customer: 5 행 (총) (대략적)

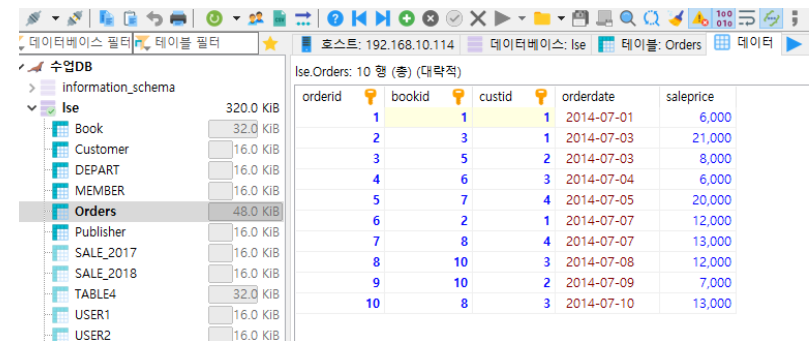
custid	custname	address	custhp
1	박지성	영국 맨체스터	010-5000-0001
2	김연아	대한민국 서울	010-6000-0001
3	장미란	대한민국 강원도	010-7000-0001
4	추신수	미국 클리블랜드	010-8000-0001
5	박세리	대한민국 대전	(NULL)



Host: 192.168.10.114 | Database: Ise | Table: Publisher

Ise.Publisher: 6 행 (총) (대략적)

publisher	repname	publisherhp
Pearson	이순신	010-1234-5555
굿스포츠	김유신	010-1234-1111
나무수	김준주	010-1234-2222
대한미디어	장보고	010-1234-3333
삼성당	이성계	010-1234-5555
이상미디어	강감찬	010-1234-4444



Host: 192.168.10.114 | Database: Ise | Table: Orders

Ise.Orders: 10 행 (총) (대략적)

orderid	bookid	custid	orderdate	saleprice
1	1	1	2014-07-01	6,000
2	3	1	2014-07-03	21,000
3	5	2	2014-07-03	8,000
4	6	3	2014-07-04	6,000
5	7	4	2014-07-05	20,000
6	2	1	2014-07-07	12,000
7	8	4	2014-07-07	13,000
8	10	3	2014-07-08	12,000
9	10	2	2014-07-09	7,000
10	8	3	2014-07-10	13,000

2) 데이터 개체 관계에 따른 무결성 방안 설명

- 무결성(Integrity)이란 데이터베이스에 저장된 데이터 값과 그것이 표현되는 현실 세계의 실제 값이 일치하는 정확성을 의미한다. 논리 데이터 모델의 중복 최소화를 통한 무결성이 보장되었는지 확인한다.
- OO서점은 출판사에서 공급한 도서만 등록하여 관리하기 때문에 등록된 출판사가 아닌 경우 도서에 데이터를 추가할 수 없게 된다.

The screenshot shows a SQL IDE interface. On the left, a tree view displays the database schema '수업DB' with tables like 'Book', 'Customer', 'DEPART', 'MEMBER', 'Orders', 'Publisher', 'SALE_2017', 'SALE_2018', 'TABLE4', 'USER1', 'USER2', 'USER3', 'USER4', 'USER5', and 'USER6'. The 'Orders' table is selected. The main window shows a SQL script with multiple INSERT statements. The script includes comments in Korean and inserts data into 'Publisher' and 'Book' tables. An error message is displayed in the bottom right corner, stating: '수업DB: 오류 SQL 오류 (1452): Cannot add or update a child row: a foreign key constraint fails ('se`.`Book`, CONSTRAINT 'FK_Publisher_TO_Book' FOREIGN KEY (publisher) REFERENCES 'Publisher' (publisher))'. The error message is in Korean and English, with a '확인' (OK) button.