

3.데이터 입출력 구현

김은표

목록

1. Exerd URL 복사
2. MYSQL 다운로드
3. Java에 실행
4. 논리, 물리 데이터 모델링
5. 데이터 모델링 데이터 추가
6. 데이터 모델링 데이터 실행
7. SQL실행

1.Exerd URL 복사

1. <http://ko.exerd.com> 접속 후 다운로드
2. 이클립스 플러그인으로 설치 누르기
3. eXERD 3.x Update Site URL 복사하기



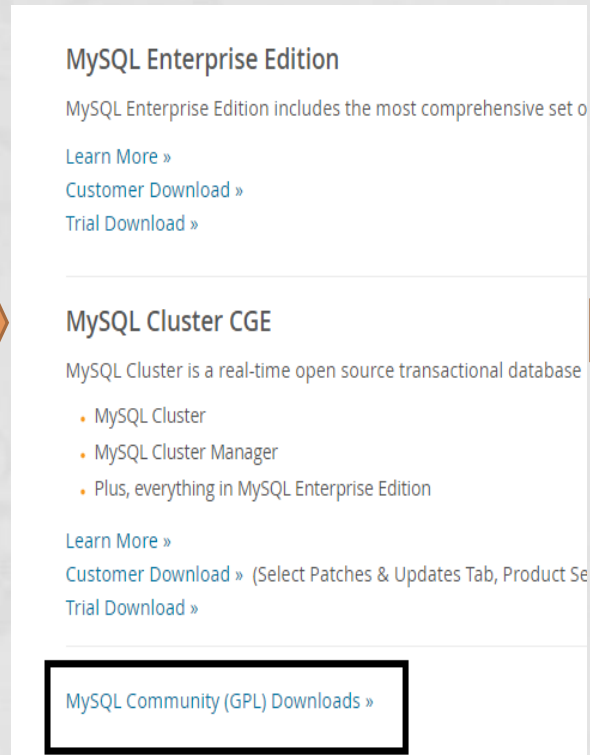
Eclipse 플러그인으로 설치

Windows 이외의 운영체제를 사용하시거나, 이클립스 기반의 원래 작업 환경에 eXERD를 이용하시려면, 플러그인 형태로 설치하실 수 있습니다.

eXERD 3.x Update Site URL (이클립스 버전 3.4 ~ 3.7, 4.3 또는 이상 버전)	http://exerd.com/update/3.x
eXERD 2.x Update Site URL (이클립스 버전 3.4 ~ 3.7, 4.3 또는 이상 버전)	http://exerd.com/update

2. MYSQL 다운로드

1. <https://www.mysql.com> 접속 후 다운로드 클릭
2. MySQL Community (GPL) Downloads 클릭
3. [Connector/J](#) 클릭



MySQL Community Downloads

- MySQL Yum Repository
- MySQL APT Repository
- MySQL SUSE Repository
- MySQL Community Server
- MySQL Cluster
- MySQL Router
- MySQL Shell
- MySQL Workbench
- MySQL Installer for Windows
- MySQL for Visual Studio
- C API (libmysqlclient)
- Connector/C++
- **Connector/J**
- Connector/NET
- Connector/Node.js
- Connector/ODBC
- Connector/Python
- MySQL Native Driver for PHP
- MySQL Benchmark Tool
- Time zone description tables
- Download Archives

2. MYSQL 다운로드

1. Looking for the latest GA version? 클릭 후 버전 확인
2. Platform Independent (Architecture Independent), ZIP Archive 클릭

[General Availability \(GA\) Releases](#) [Archives](#)

Connector/J 5.1.49

Select Operating System:
Platform Independent

①

Looking for the latest GA version?

Recommended Windows Download:

MySQL Installer for Windows

All MySQL Products. For All Windows Platforms. In One Package.

Starting with MySQL 5.6 the MySQL Installer package replaces the standalone MSI packages.

Windows (x86, 32 & 64-bit), MySQL Installer MSI

[Go to Download Page >](#)

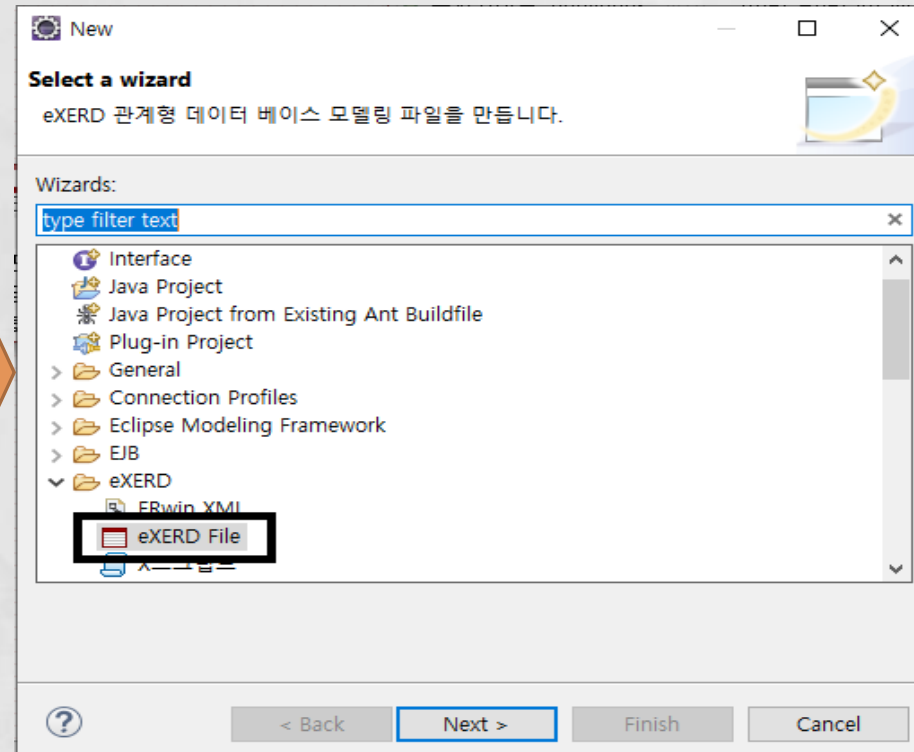
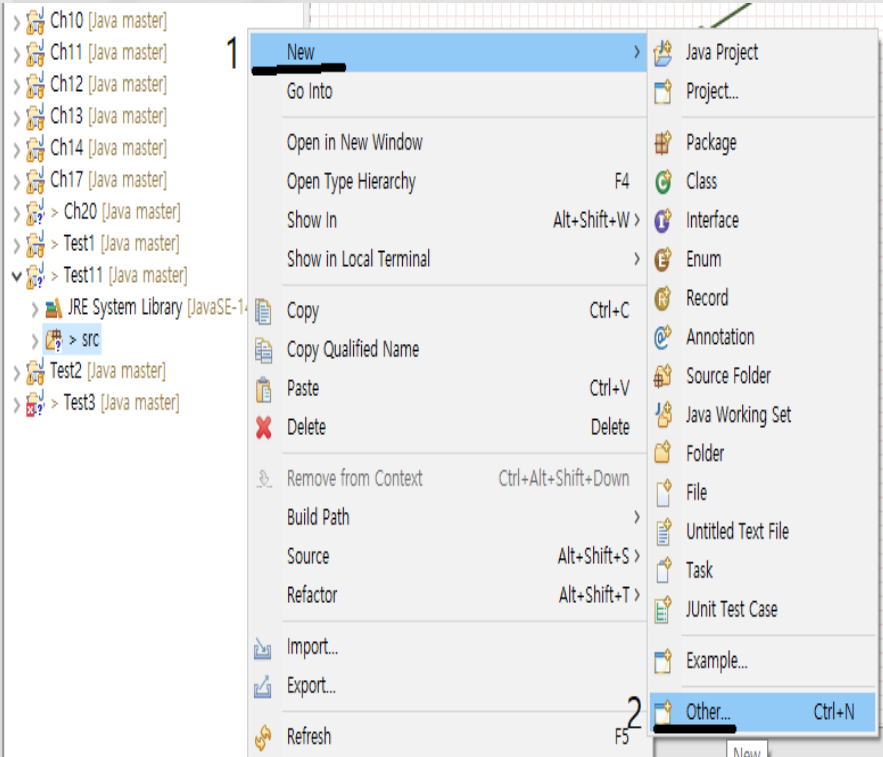
Platform Independent (Architecture Independent), Compressed TAR Archive (mysql-connector-java-5.1.49.tar.gz)	5.1.49	3.2M	Download
Platform Independent (Architecture Independent), ZIP Archive (mysql-connector-java-5.1.49.zip)	5.1.49	3.5M	Download

②

We suggest that you use the [MD5 checksums](#) and [GnuPG signatures](#) to verify the integrity of the packages you download.

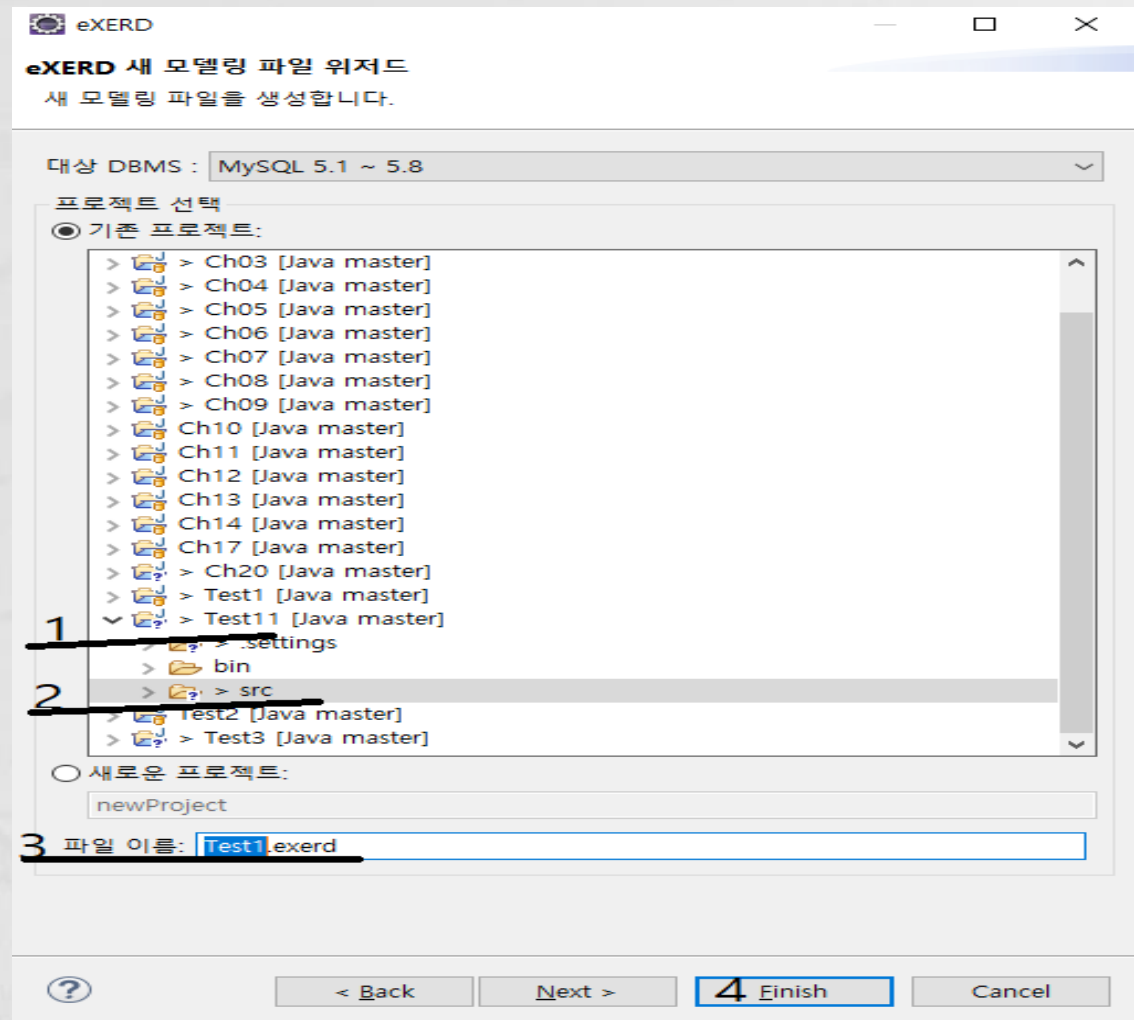
3. Java에 실행

1. Java project 폴더 생성
2. src에서 new Other 클릭
3. eXERD FILE 클릭 후 Next



3.Java에 실행

1. 대상 DBMS에 MYSQL 5.1~5.8버전 확인
2. 지정할 폴더 src 클릭
3. 파일이름 지정 후 finish



4.논리, 물리 데이터 모델링

고객		TABLE3		
논리 이름*	물리 이름*	도메인	데이터 타입	널 허용
● 고객번호	custid	N/A	INT	N-N
● 고객이름	name	N/A	VARCHAR(20)	NULL
● 주소	address	N/A	VARCHAR(30)	NULL
● 전화번호	csthph	N/A	CHAR(13)	NULL

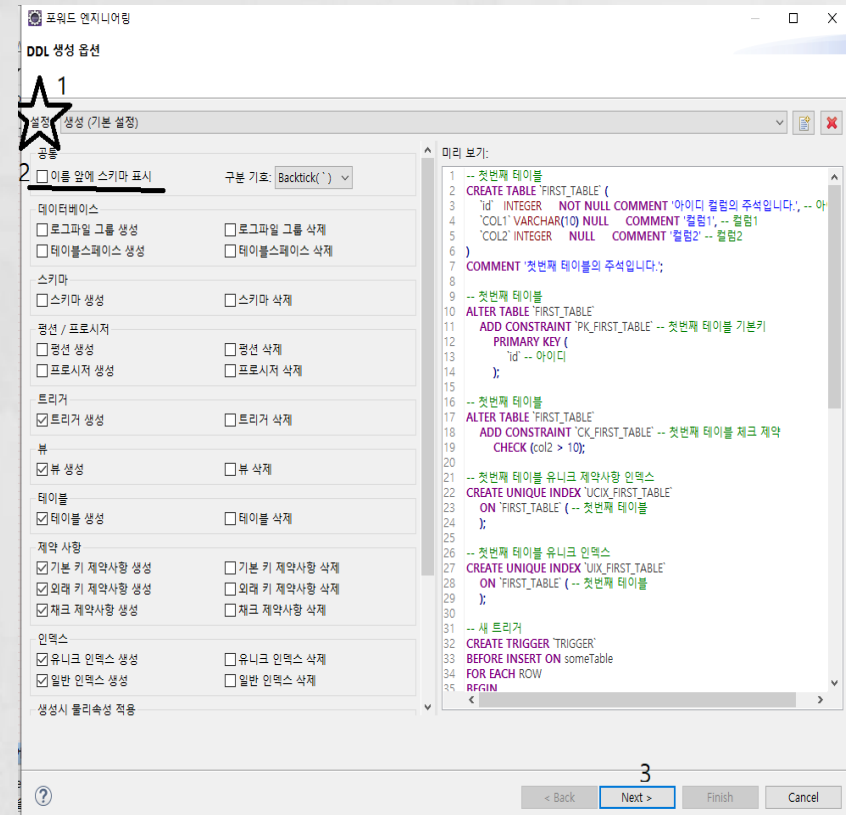
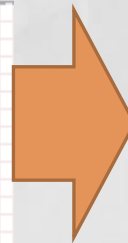
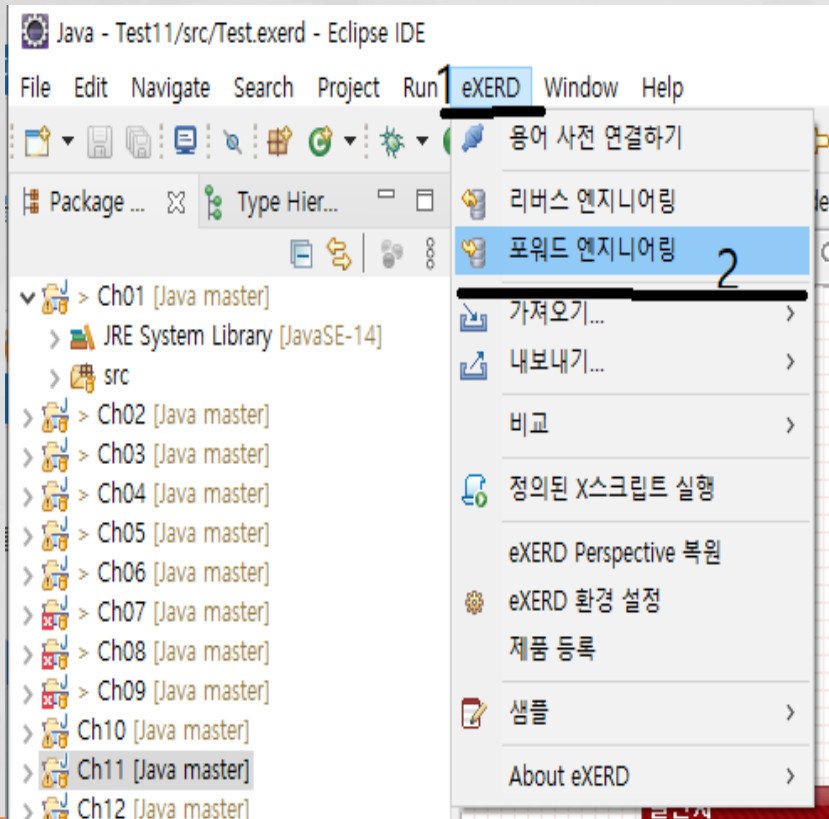
주문		TABLE4		
논리 이름*	물리 이름*	도메인	데이터 타입	널 허용
● 고객번호	custid	N/A	INT	N-N
● 주문번호	orderid	N/A	INT	NULL
● 도서이름	bookid	N/A	INT	NULL
● 주문금액	saleprice	N/A	INT	NULL
● 주문날짜	orderdate	N/A	DATE	NULL

도서		TABLE		
논리 이름*	물리 이름*	도메인	데이터 타입	널 허용
● 도서번호	bookid	N/A	INT	N-N
● 도서이름	bookname	N/A	VARCHAR(20)	NULL
● 출판사이름	publisher	N/A	VARCHAR(20)	NULL
● 도서단가	price	N/A	INT	NULL

출판사		TABLE2		
논리 이름*	물리 이름*	도메인	데이터 타입	널 허용
● 출판사이름	pubname	N/A	INT	NULL
● 담당자이름	staffname	N/A	VARCHAR(20)	NULL
● 전화번호	pubhph	N/A	CHAR(13)	NULL

4.논리, 물리 데이터 모델링

1. eXERD 클릭
2. 포워드 엔지니어링 클릭
3. 설정에 생성(기본설정) 확인
4. 이름앞에 스키마 표시 X
5. Next



4.논리, 물리 데이터 모델링

1. 파일선택
2. MYSQL 다운 받은 파일 설정하기
3. 드라이버 클래스 설정
4. 호스트 입력
5. 데이터 데이스, 사용자, 비번 확인
6. 연결 테스트 확인 후 끝내기

포워드 엔지니어링

연결 설정

연결 테스트를 진행하여 연결이 올바른지 확인해드립니다.

연결: 마지막으로 구성된 연결

JDBC 드라이버: 2 C:\Users\bigdata\Desktop\workspace\Java\Ch20\lib\mysql-connector-java-5.1.49.jar 1 파일 선택

드라이버 클래스: 3 com.mysql.jdbc.Driver

URL: jdbc:mysql://192.168.10.114:3306/kep?characterEncoding=utf8

호스트: 4 192.168.10.114 포트: 3306

문자 인코딩: utf8

데이터베이스: 5 kep

사용자: kep

비밀번호: ●●●●

키	값

속성 추가

속성 제거

6 연결 테스트

< Back Next > 7 Finish Cancel

5. 데이터 모델링 데이터 추가

수업DB#kep#Orders# - HeidiSQL 11.1.0.6116

파일 편집 검색 쿼리 도구 이동 도움말

호스트: 192.168.10.114 데이터베이스: kep 테이블: Orders 데이터 SQL실습과제.sql SQL수행평가.sql

Donate

```
32 );
33 #실습 4
34 INSERT INTO `Booklist` (`bookname`, `publisher`, `price`) VALUES ('자바의 이해', '삼성', 17000);
35 INSERT INTO `Booklist` (`bookname`, `publisher`, `price`) VALUES ('자바 기초', '삼성', 22000);
36 INSERT INTO `Booklist` (`bookname`, `publisher`, `price`) VALUES ('자바 활용', '두산', 13000);
37 INSERT INTO `Booklist` (`bookname`, `publisher`, `price`) VALUES ('파이선 기초', '대한', 35000);
38 INSERT INTO `Booklist` (`bookname`, `publisher`, `price`) VALUES ('컴퓨터공학', '컴공', 28000);
39 INSERT INTO `Booklist` (`bookname`, `publisher`, `price`) VALUES ('MY SQL', '대한', 26000);
40 INSERT INTO `Booklist` (`bookname`, `publisher`, `price`) VALUES ('c언어', '컴공', 20000);
41 INSERT INTO `Booklist` (`bookname`, `publisher`, `price`) VALUES ('파이선 활용', '두산', 13000);
42 INSERT INTO `Booklist` (`bookname`, `publisher`, `price`) VALUES ('데이터베이스', '두산', 17500);
43 INSERT INTO `Booklist` (`bookname`, `publisher`, `price`) VALUES ('정보처리기사', '두산', 20000);
44
45
46 INSERT INTO `Publisher` (`publname`, `staffname`, `hp`) VALUES ('삼성', '박지성', '000-5000-0001');
47 INSERT INTO `Publisher` (`publname`, `staffname`, `hp`) VALUES ('두산', '김연아', '000-6000-0001');
48 INSERT INTO `Publisher` (`publname`, `staffname`, `hp`) VALUES ('대한', '장미란', '000-7000-0001');
49 INSERT INTO `Publisher` (`publname`, `staffname`, `hp`) VALUES ('컴공', '주신수', '000-8000-0001');
50 INSERT INTO `Publisher` (`publname`, `staffname`, `hp`) VALUES ('두산', '박세리', '000-9000-0001');
51
52
53 INSERT INTO `Customer` (`name`, `address`, `hp`) VALUES ('홍길동', '부산', '010-1111-1111');
54 INSERT INTO `Customer` (`name`, `address`, `hp`) VALUES ('김말자', '서울', '010-1111-2222');
55 INSERT INTO `Customer` (`name`, `address`, `hp`) VALUES ('장보고', '창원', '010-1111-3333');
56 INSERT INTO `Customer` (`name`, `address`, `hp`) VALUES ('아이유', '마산', '010-1111-4444');
57 INSERT INTO `Customer` (`name`, `address`, `hp`) VALUES ('소지섭', '대전', '010-1111-5555');
58 INSERT INTO `Customer` (`name`, `address`, `hp`) VALUES ('강호동', '부산', '010-1111-6666');
59 INSERT INTO `Customer` (`name`, `address`, `hp`) VALUES ('유아인', '서울', '010-1111-7777');
60 INSERT INTO `Customer` (`name`, `address`, `hp`) VALUES ('강동원', '창원', '010-1111-8888');
61 INSERT INTO `Customer` (`name`, `address`, `hp`) VALUES ('박종원', '마산', '010-1111-9999');
62 INSERT INTO `Customer` (`name`, `address`, `hp`) VALUES ('서태지', '대전', '010-1111-0000');
63
64
65 INSERT INTO `Orders` (`custid`, `bookid`, `saleprice`, `orderdate`) VALUES (1,1,6000, '2014-07-01');
66 INSERT INTO `Orders` (`custid`, `bookid`, `saleprice`, `orderdate`) VALUES (1,3,21000, '2014-07-03');
67 INSERT INTO `Orders` (`custid`, `bookid`, `saleprice`, `orderdate`) VALUES (2,5,8000, '2014-07-03');
68 INSERT INTO `Orders` (`custid`, `bookid`, `saleprice`, `orderdate`) VALUES (3,6,6000, '2014-07-04');
69 INSERT INTO `Orders` (`custid`, `bookid`, `saleprice`, `orderdate`) VALUES (4,7,20000, '2014-07-05');
70 INSERT INTO `Orders` (`custid`, `bookid`, `saleprice`, `orderdate`) VALUES (1,2,12000, '2014-07-07');
71 INSERT INTO `Orders` (`custid`, `bookid`, `saleprice`, `orderdate`) VALUES (4,8,13000, '2014-07-07');
72 INSERT INTO `Orders` (`custid`, `bookid`, `saleprice`, `orderdate`) VALUES (3,10,12000, '2014-07-08');
73 INSERT INTO `Orders` (`custid`, `bookid`, `saleprice`, `orderdate`) VALUES (2,10,7000, '2014-07-09');
74 INSERT INTO `Orders` (`custid`, `bookid`, `saleprice`, `orderdate`) VALUES (3,8,13000, '2014-07-10');
75
```

896 SHOW PROCEDURE STATUS WHERE `Db`='kep';

897 SHOW TRIGGERS FROM `kep`;

r91 : c15 (4.9 KiB) 연결됨: 03:07 h

MariaDB 5.5.65

가동 시간: 7 일, 01:57 h

서버 시간: 오후 1:01 유틸

오후 1:01
2020-11-20

6.데이터 모델링 데이터 실행

도서목록

kep.Booklist: 10 행 (총) (대략적) >> 다음

bookid	bookname	publisher	price
1	자바의 이해	더넥서스	17,000
2	자바 기초	더넥서스	22,000
3	자바 활용	IT센터	13,000
4	파이선 기초	대한	35,000
5	컴퓨터공학	کم공	28,000
6	MY SQL	대한	26,000
7	c언어	کم공	20,000
8	파이선 활용	두산	13,000
9	데이터베이스	두산	175,500
10	정보처리기사	IT센터	20,000

출판사 목록

publname	staffname	hp
삼성	박지성	000-5000-0001
무학	김연아	000-6000-0001
대한	장미란	000-7000-0001
کم공	추신수	000-8000-0001
두산	박세리	000-9000-0001

고객 목록

kep.Customer: 10 행 (총) (대략적) >> 다음

custid	name	address	hp
1	홍길동	부산	010-1111-1111
2	김말자	서울	010-1111-2222
3	장보고	창원	010-1111-3333
4	아이유	마산	010-1111-4444
5	소지섭	대전	010-1111-5555
6	강호동	부산	010-1111-6666
7	유아인	서울	010-1111-7777
8	강동원	창원	010-1111-8888
9	백종원	마산	010-1111-9999
10	서태지	대전	010-1111-0000

주문목록

kep.Orders: 10 행 (총) (대략적)

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

7.SQL실행

76 #실습하기 4-1

```
77 SELECT DISTINCT `bookid`, `bookname`, `publisher` FROM `Booklist`;
```

Booklist (0r x 3c)

bookid	bookname	publisher
--------	----------	-----------

78 #실습하기 4-2

```
79 SELECT * FROM `Book` WHERE `price` >= 20000 ORDER BY `price` DESC;
```

Book (3r x 4c)

bookid	bookname	publisher	price
4	골프 바이블	대한미디어	35,000
3	축구의 이해	대한미디어	22,000
7	야구의 추억	이상미디어	20,000

92 #실습하기 4-5

```
93 SELECT * FROM `Customer` AS a
94 JOIN `Orders` AS b
95 ON a.custid = b.custid;
```

Customer (10r x 9c)

custid	name	address	hp	orderid	custid	bookid	saleprice	orderdate
1	홍길동	부산	010-1111-1111	1	1	1	6,000	2014-07-01
1	홍길동	부산	010-1111-1111	2	1	3	21,000	2014-07-03
2	김말자	서울	010-1111-2222	3	2	5	8,000	2014-07-03
3	장보고	창원	010-1111-3333	4	3	6	6,000	2014-07-04
4	아이유	마산	010-1111-4444	5	4	7	20,000	2014-07-05
1	홍길동	부산	010-1111-1111	6	1	2	12,000	2014-07-07
4	아이유	마산	010-1111-4444	7	4	8	13,000	2014-07-07
3	장보고	창원	010-1111-3333	8	3	10	12,000	2014-07-08
2	김말자	서울	010-1111-2222	9	2	10	7,000	2014-07-09
3	장보고	창원	010-1111-3333	10	3	8	13,000	2014-07-10

85 #실습하기 4-4

```
86 SELECT
87     SUM(`saleprice`) AS `총 판매액`,
88     AVG(`saleprice`) AS `평균값`,
89     MIN(`saleprice`) AS `최저가`,
90     MAX(`saleprice`) AS `최고가`
91 FROM `Orders`;
```

Orders (1r x 4c)

총판매액	평균값	최저가	최고가
0000	6,000	21,000	