

电子科技大学信息与软件工程学院

实 验 报 告

学 号 2015220103022

姓 名 张健顺

(实验) 课程名称 数据库原理及应用

理论教师 陈安龙

实验教师 陈安龙

电子科技大学

实验报告

学生姓名：张健顺 学号：20152201303022 指导教师：陈安龙

实验地点：学院楼 房间：306 实验时间：2017.05.08

一、实验一：在线图书销售系统 CDM/LDM 数据模型设计

二、实验学时：4

三、实验目的

通过对在线图书销售系统的业务处理分析,了解系统数据处理及其数据库开发的数据需求。理解系统概念数据模型(CDM)和逻辑数据模型(LDM)的功能及作用,特别是 E-R 图表示方法。掌握 CDM/LDM 数据模型设计方法及其转换关系,培养系统数据架构建模的分析与设计能力。

四、实验原理

通过使用 PowerDesigner 这款优秀的建模工具,将在线图书销售系统的业务分析抽象成相应的数据关系和实体,建立相应的数据模型,掌握 CDM 和 LDM 各自的功能和特点。

五、实验内容

基于对在线图书销售系统业务的基本数据需求分析,使用 Power Designer 设计工具创建在线图书销售系统概念数据模型和逻辑数据模型,为设计数据库提供依据。具体实验内容如下:

- 1) 利用 PowerDesigner 工具创建数据模型文件
- 2) 在线图书销售系统数据实体抽取
- 3) 在线图书销售系统数据实体与联系定义
- 4) 在线图书销售系统 CDM 数据模型创建
- 5) 在线图书销售系统 CDM 数据模型到 LDM 数据模型转换设计
- 6) 在线图书销售系统 CDM/LDM 数据模型设计完善

7) 在线图书销售系统 LDM 数据模型设计验证 在实验计算机上,利用专业数据库建模工具 Power Designer,完成在线图书销售系统的概念数据模型和逻辑数据模型设计。

辑数据模型设计。同时记录实验过程的设计思路、操作方法、运行结果界面等数据，为撰写实验报告提供素材。

六、实验器材（设备、元器件）

“数据库原理及应用”实验所涉及的机房硬件设备为 pc 计算机、服务器以及网络环境，pc 计算机与服务器在同一局域网络。

操作系统： Windows 7 / Windows XP

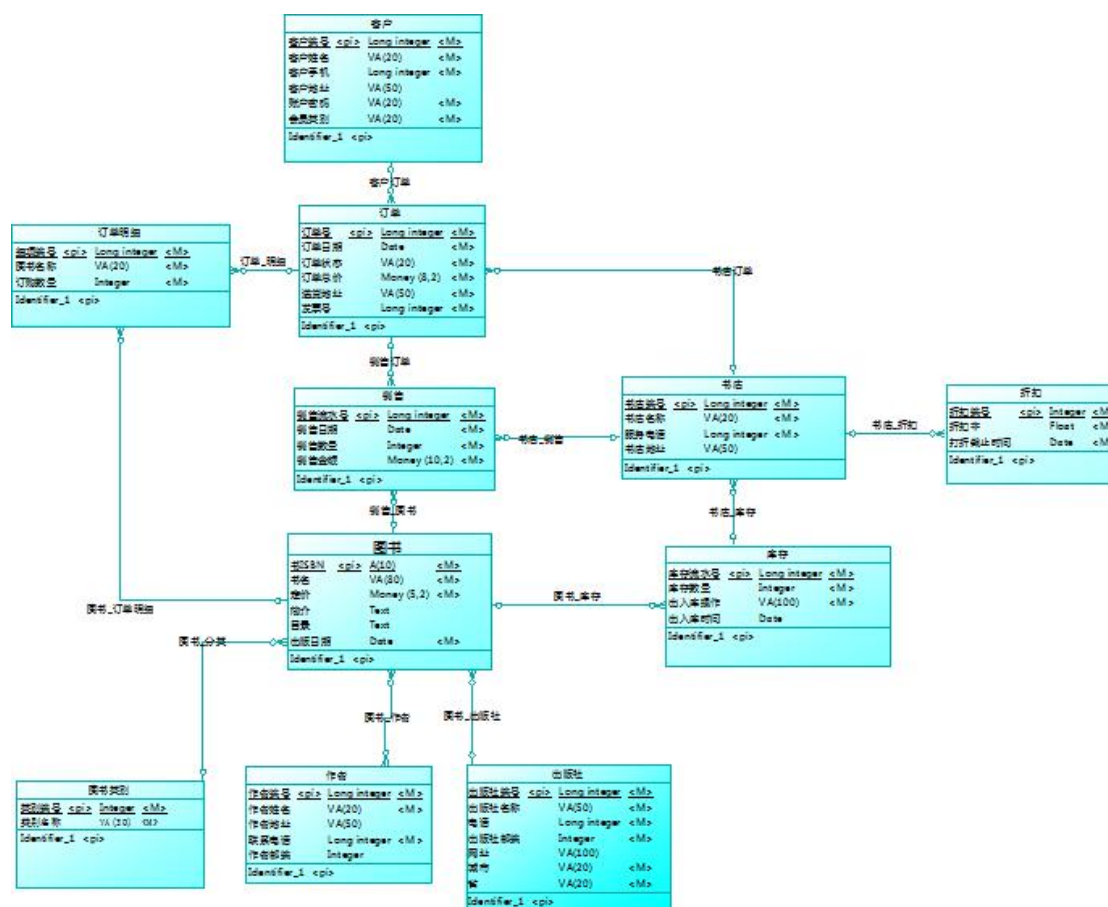
数据库建模工具：SYBASE Power Designer16.5

七、实验步骤

对于在线图书销售系统数据模型设计，可通过使用 Sybase 公司的 Power Designer 数据库设计工具，其基本实验步骤如下：

- 1) 启动 Power Designer 软件运行，进入工具环境。
- 2) 在文件（File）中的“新建模型（New Model）”菜单项中，选取概念数据模型进行创建，并定义模型名称。
- 3) 在概念数据模型工作区中，通过点击 E-R 模型元素图标，建立相应的实体对象，如作者、出版社、图书等。
- 4) 针对模型中每个实体定义属性，输入相应数据信息。
- 5) 定义实体之间的关系，可双击关联线，弹出关联对话框，输入相应关联信息。
- 6) 完善在线图书销售系统概念数据模型设计。
- 7) 将在线图书销售系统 CDM 转换为 LDM 数据模型。
- 8) 检查在线图书销售系统 LDM 数据模型的正确性。
- 9) 完善在线图书销售系统 LDM 数据模型设计。

八、实验结果与分析

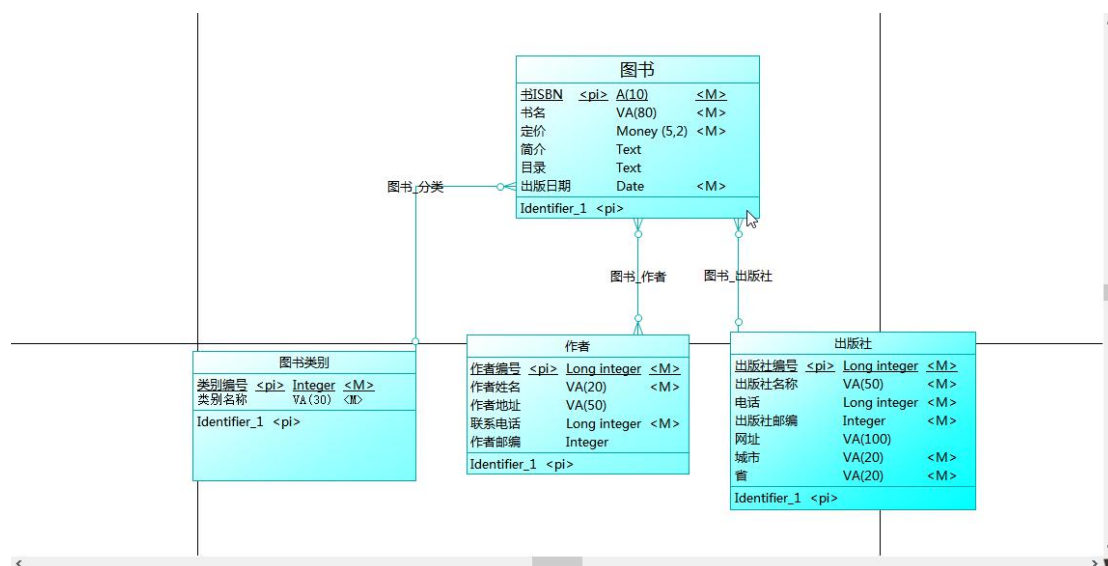


图书销售系统整体模型

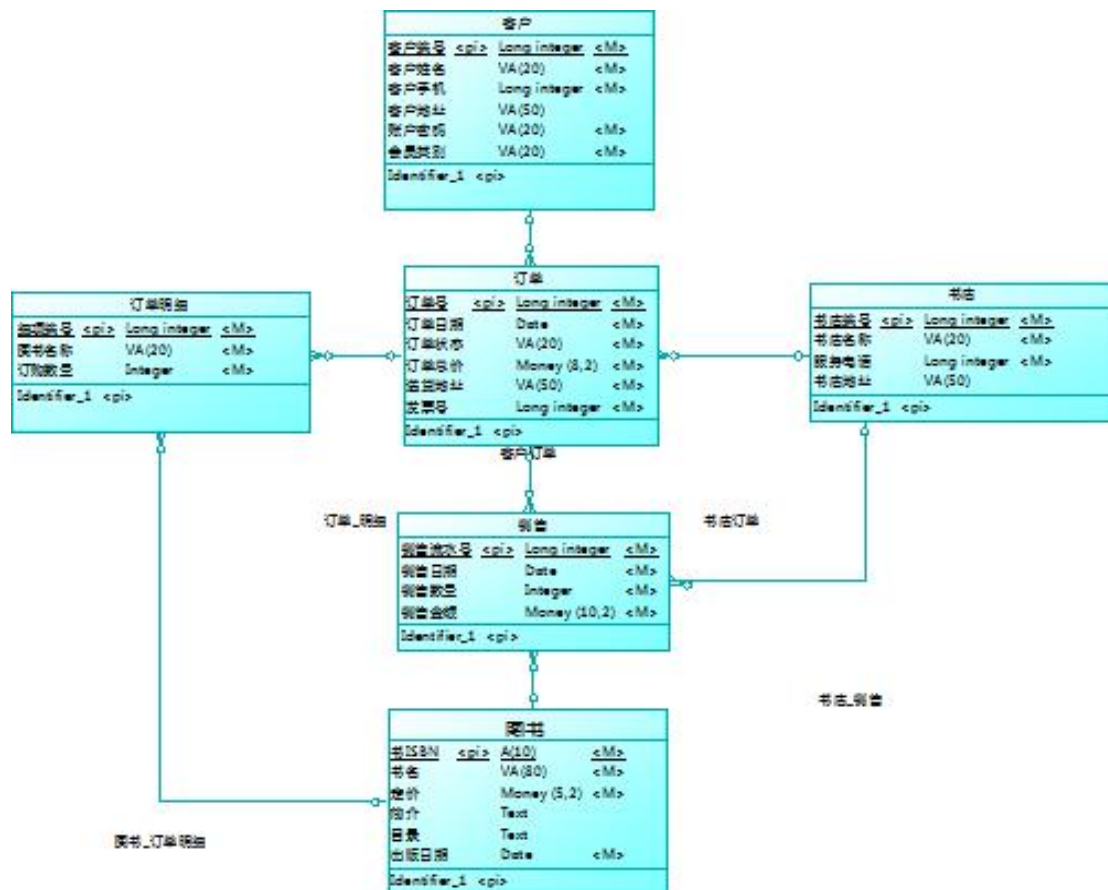
将整体模型进行相应的模块划分，简化概念模型并进行设计。

划分为三个子模块：

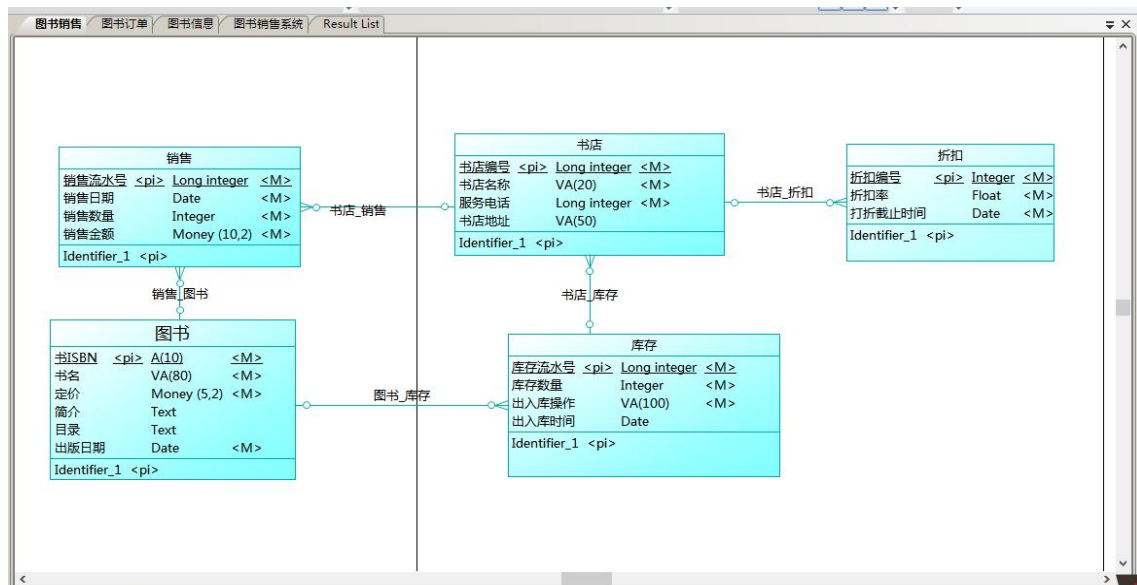
- ①、图书信息；
- ②、图书订单；
- ③、图书销售



图书信息子模块模型



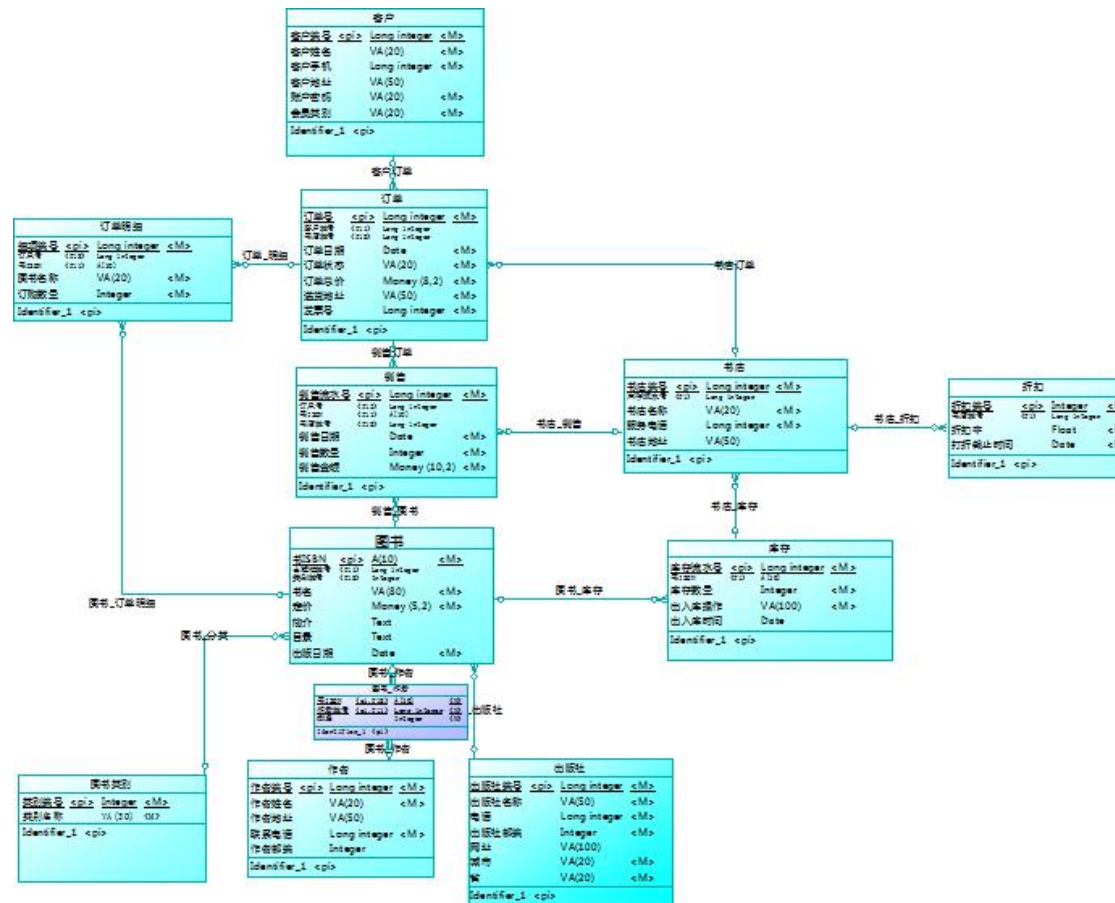
图书订单子模块



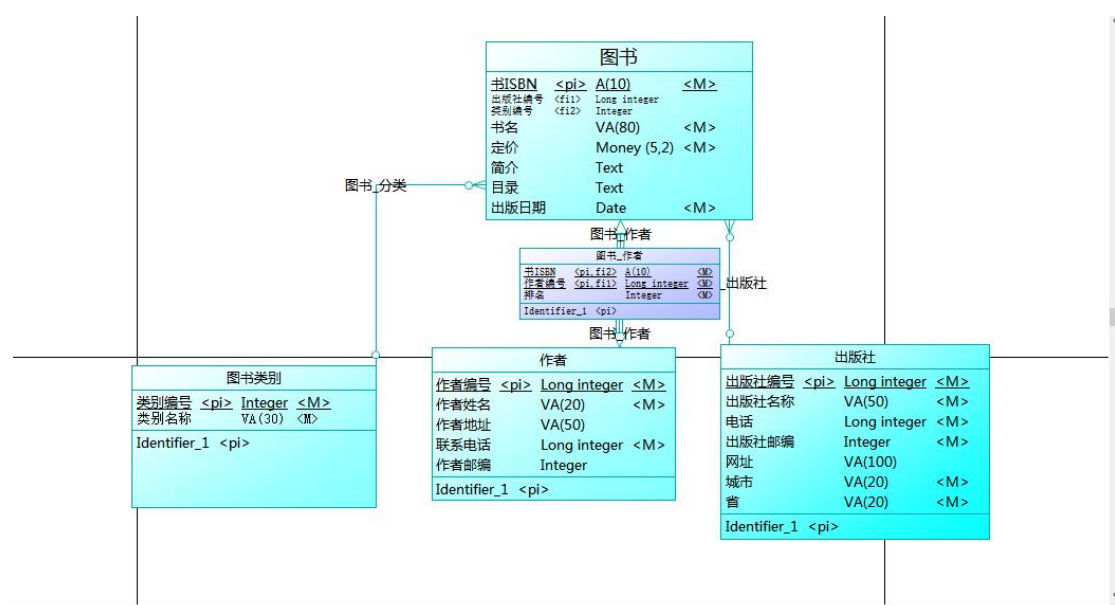
图书销售子模块

在建立模型的过程中，对于模型中的各个实体需要按照数据库中表的设计原

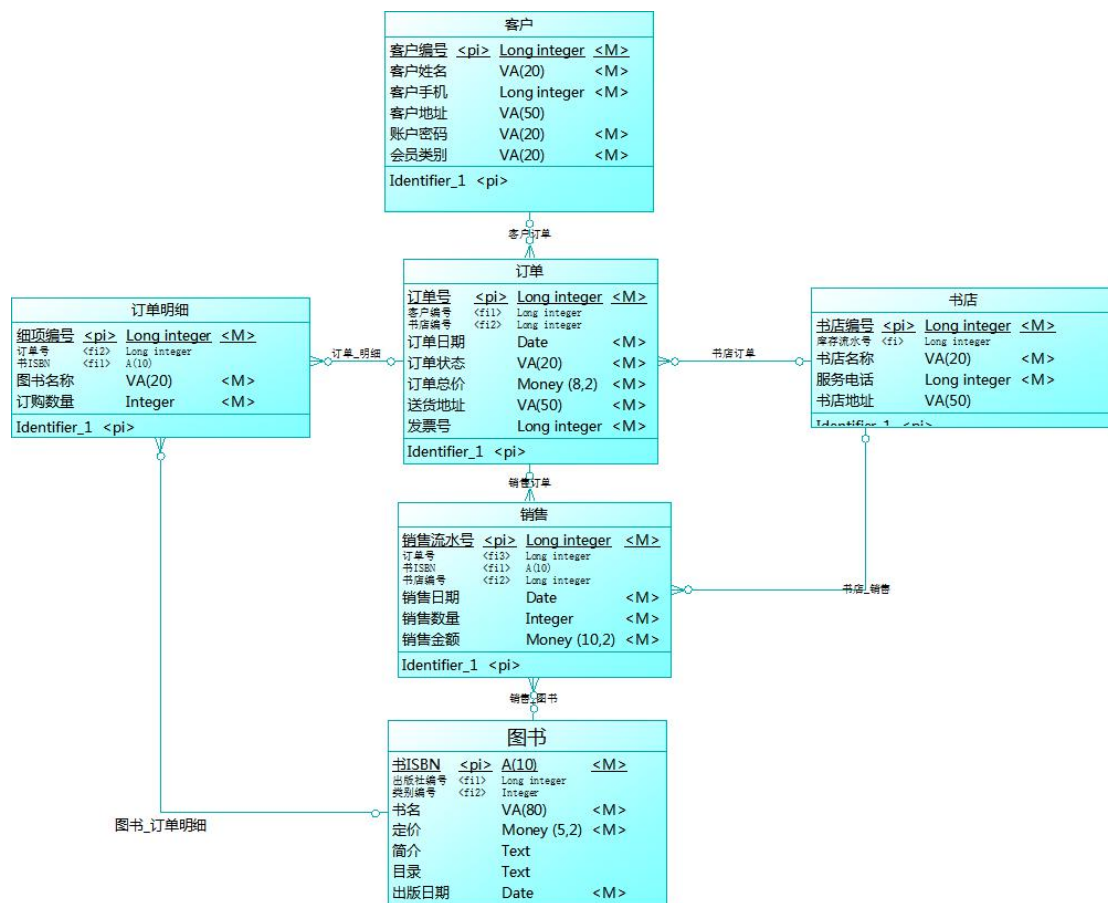
则来进行相关设计，对于实体的属性的类型、长度以及限定条件都需要进行相关的设置。在建立好的实体基础之上，需要建立实体之间的对应关系，一对一、多对多、一对多、多对一，根据实际情况而定，只有在建立好实体以及对应关系的基础之上，才能将 CDM 模型转为 LDM 模型。



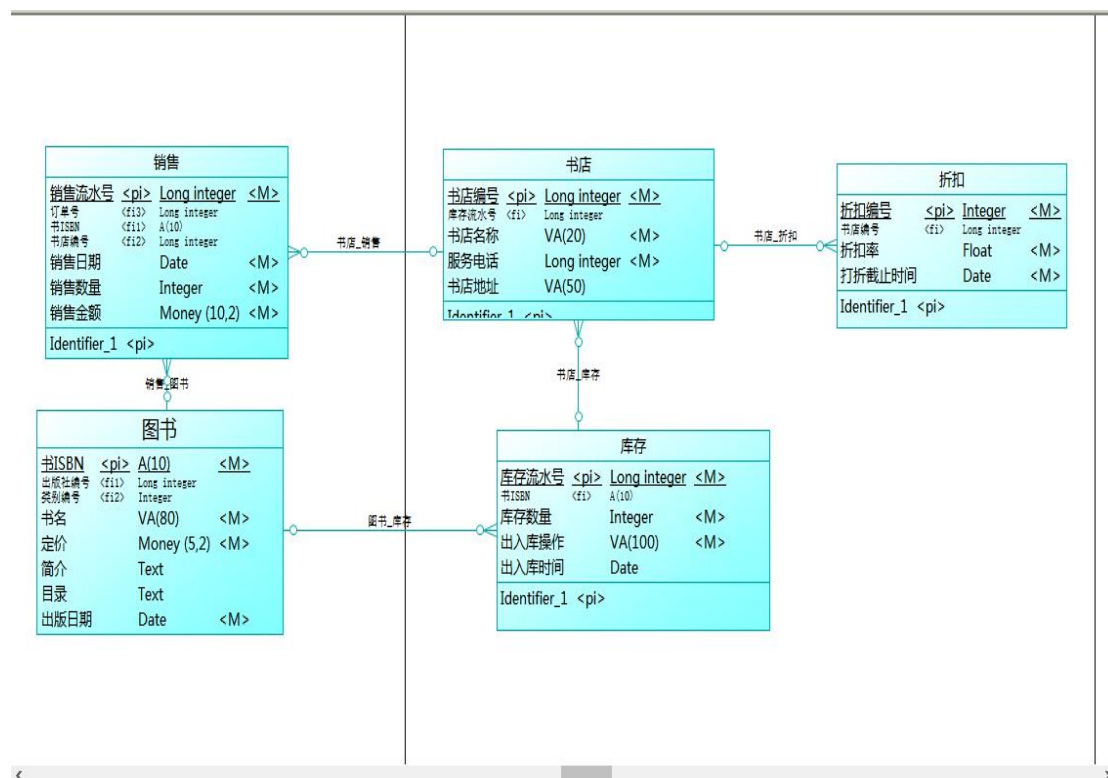
图书销售整体 LDM 模型



图书信息 LDM 模型



图书订单 LDM 模型



图书销售 LDM 模型

通过 powerDesinger 中的 Tools 工具中的 Generate Logical Data Model 选项, 将建好的 CDM 模型转换成物理模型。通过比较 LDM 和 CDM 发现在 LDM 模型中实体中的属性因为和其他实体之间的关联关系的存在而增加了一些属性, 这些属性又是唯一标识相关联的实体的主键类属性。对于实体之间存在的不同种类的关系, 会有相应的不同的结果生成。譬如图书和作者之间的多对多关系, 会生成一个新的实体, 而该实体的属性则是由图书和作者两个表中的唯一标识属性组成, 以表示图书和作者相应的对应关系。

九、总结及心得体会

①、该实验的核心就是要首先明确所要建立的模型的实体之间的对应关系, 其次是实体各自的属性的相应的限定。只有在建立好实体以及实体之间的关系的的基础之上, 才能正确地生成相应的逻辑模型;

②、在逻辑模型生成以后, 需要各个实体的属性进一步完善, 从最开始概念数据模型的建立到逻辑模型的成功建立其实是一个不断完善的过程。

③、对于相对较为复杂的系统, 可以进行相应的系统划分来简化系统, 从而提高开发效率。

④、要学会使用 PowerDesinger 建模工具, 利用好该工具的强大的功能完成数据库模型的相应设计, 可以达到事半功倍的效果。

十、对本实验过程及方法、手段的改进建议

对于本次实验, 由于核心是各个实体之间的关系以及相应的限定条件, 希望对于不同的关系建立出来的不同的 LDM 逻辑数据模型的结果进行相应的比较, 充分理解各种逻辑关系。对于生成的逻辑数据模型, 和概念数据模型进行比较, 充分体会概念数据模型到逻辑数据模型的转换过程。

电子科技大学

实验报告

学生姓名：张健顺 学号：20152201303022 指导教师：陈安龙

实验地点：学院楼 房间：306 实验时间：2017.05.15

一、实验二：在线图书销售系统 PDM 数据模型设计与数据库 SQL 实现

二、实验学时：4

三、实验目的

针对 ORACLE 数据库平台，了解与掌握系统物理数据模型设计及其数据库实现方法，同时理解系统物理数据模型（PDM）的功能与作用，特别是数据物理模型实现数据库方法。培养数据库物理数据模型设计及其数据库实现能力。

四、实验原理

利用 PowerDesigner 这款优秀的建模工具对相应的 CDM、LDM 设计之后，生成相应的 PDM，并利用自动化工具导出相应的 sql 语句，oracle 数据库平台便可直接导入相应的 sql 文件进行数据库的生成。

五、实验内容

在所设计的系统 CDM/LDM 数据模型基础上，针对 Oracle Database 数据库管理系统平台进行物理数据模型设计，并完成在线图书销售系统数据库表创建生成。具体实验内容如下：

- 1) 在线图书销售系统 CDM/LDM 到物理数据模型 PDM 转换设计
- 2) 在线图书销售系统物理数据模型 PDM 设计完善
- 3) 在线图书销售系统 PDM 到 ORACLE 数据库对象实现 SQL 转换
- 4) 在线图书销售系统 PDM 数据模型设计报告创建在实验计算机上，利用专业数据库建模工具 Power Designer，完成在线图书销售系统的物理数据模型设计及实现。同时记录实验过程的设计思路、操作方法、运行结果界面等数据，为撰写实验报告提供素材。

六、实验器材（设备、元器件）

“数据库原理及应用”实验所涉及的机房硬件设备为 pc 计算机、服务器以及网络环境，pc 计算机与服务器在同一局域网。

操作系统： Windows 7 / Windows XP

数据库建模工具：SYBASE Power Designer16.5

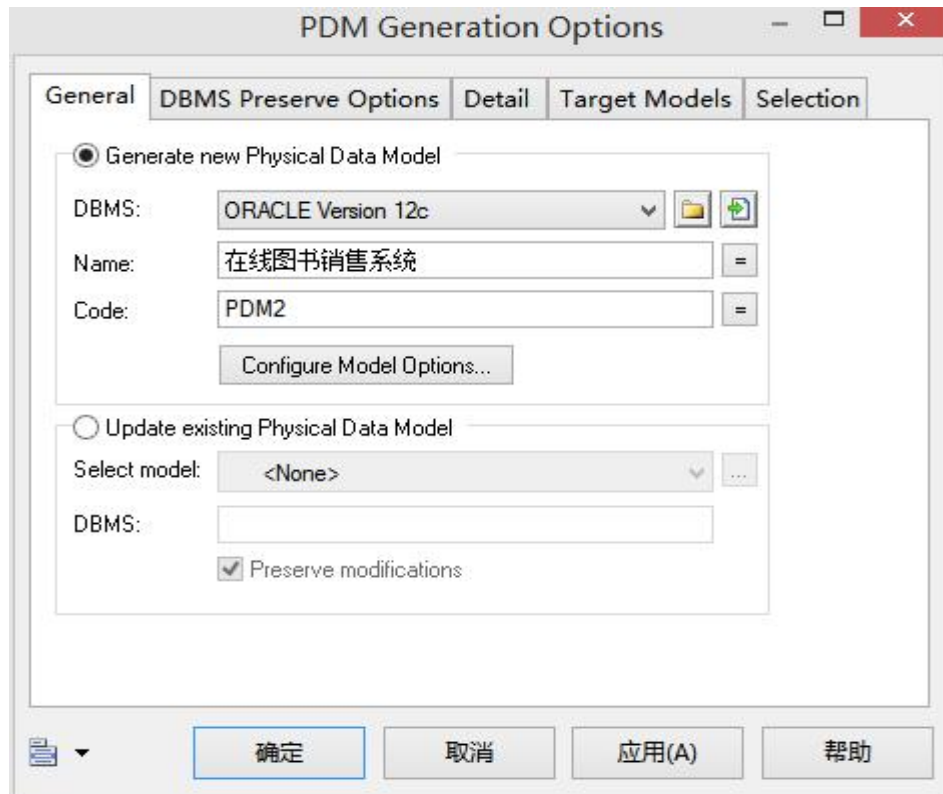
七、实验步骤

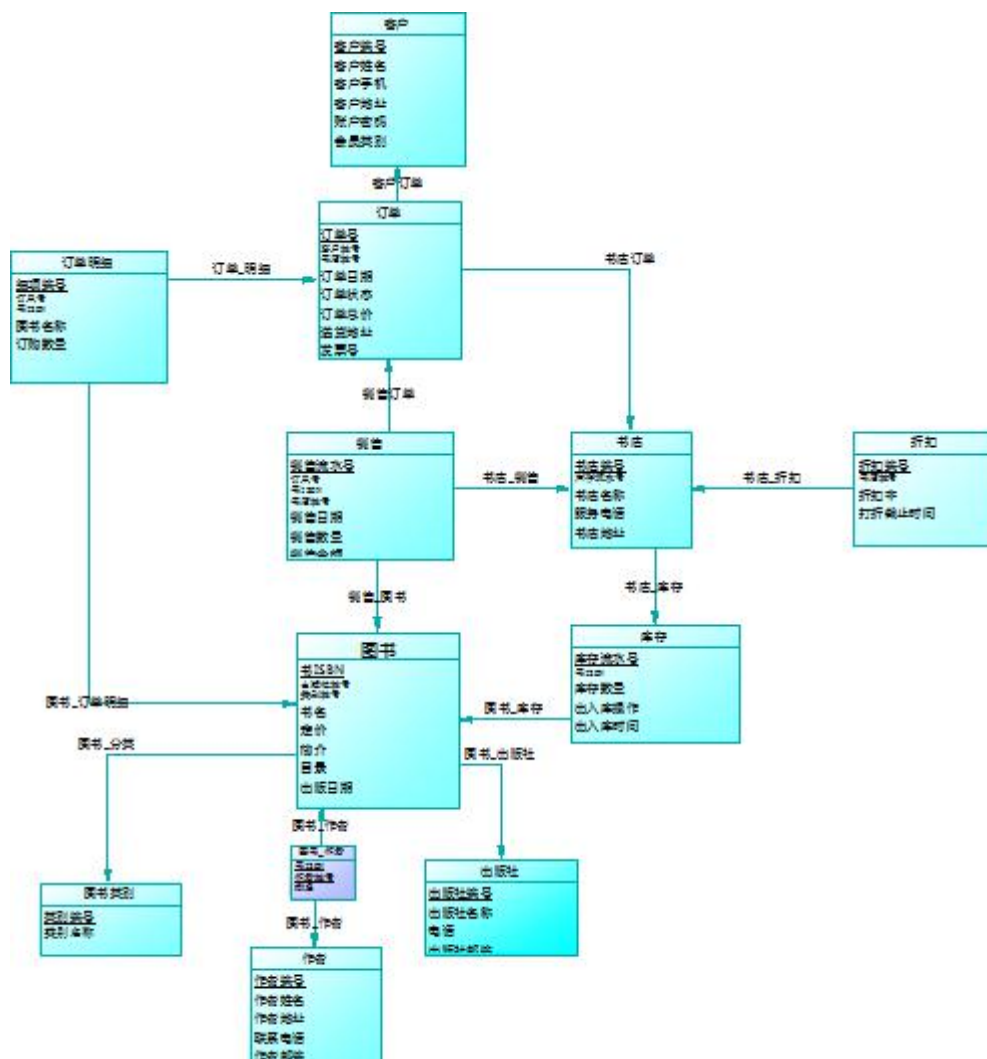
在所设计的 CDM/LDM 数据模型基础上，使用 Power Designer 数据库设计工具继续完成物理数据模型设计和数据库表创建，其步骤如下：

- 1) 打开在线图书销售系统逻辑数据模型。
- 2) 使用 Power Designer, 选取 Oracle Database 12c 为实现目标数据库 DBMS。
- 3) 将在线图书销售系统逻辑数据模型转换物理数据模型。
- 4) 检查所设计的物理数据模型是否存在错误。
- 5) 完善系统物理数据模型设计
- 6) 将在线图书销售系统物理数据模型转换为 Oracle Database 11g/12c 数据库对象创建 SQL 程序。
- 7) 生成在线图书销售系统数据库物理数据模型设计报告

八、实验结果与分析

根据实验一生成并完善后的 LDM，利用 PowerDesigner 相应的工具生成 PDM。

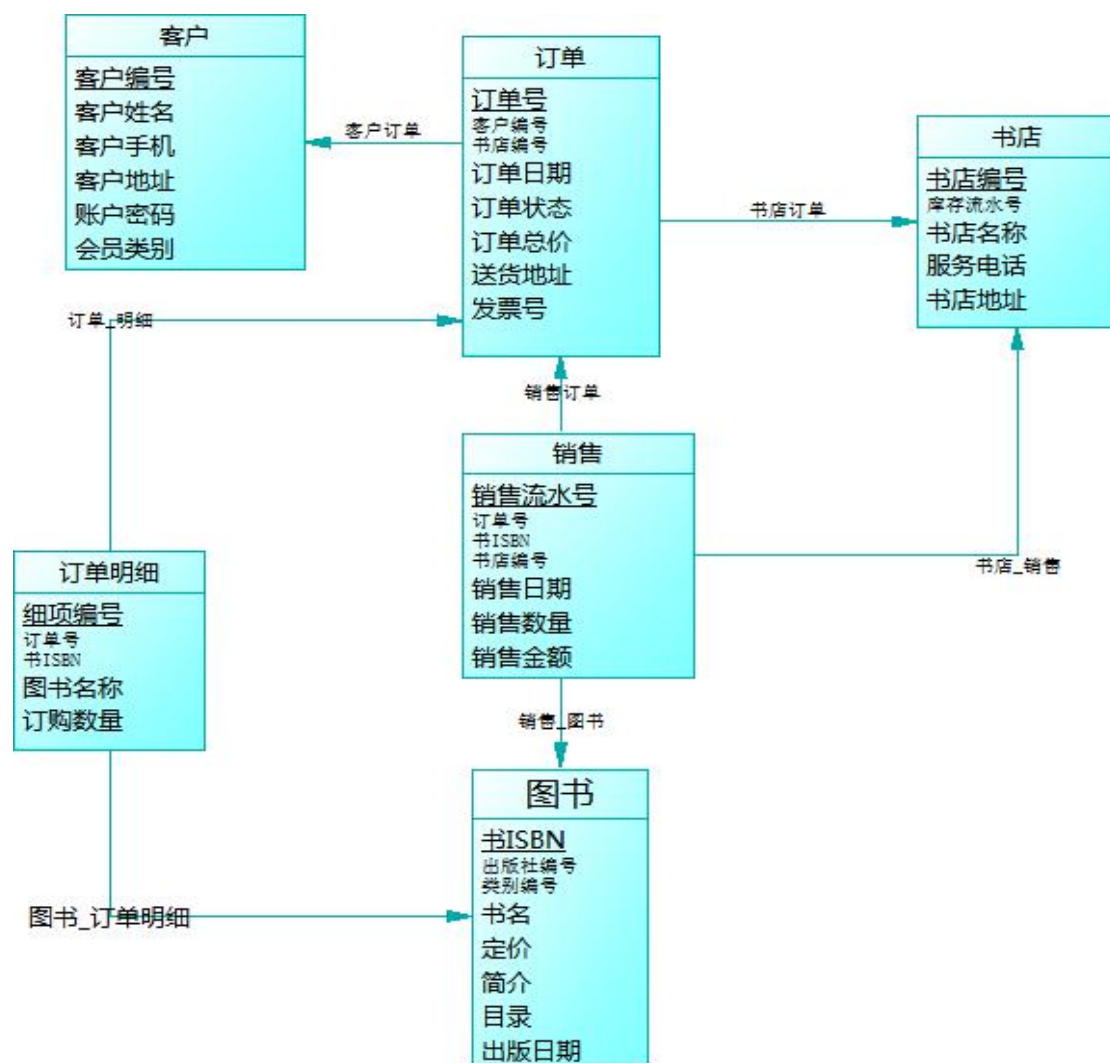




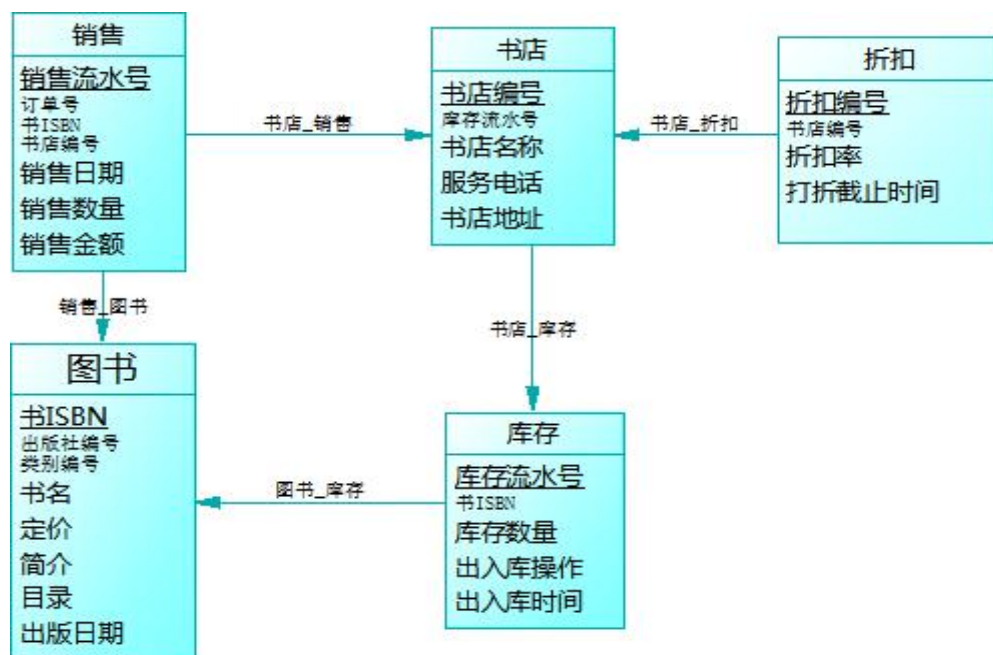
图书销售系统整体 PDM 模型



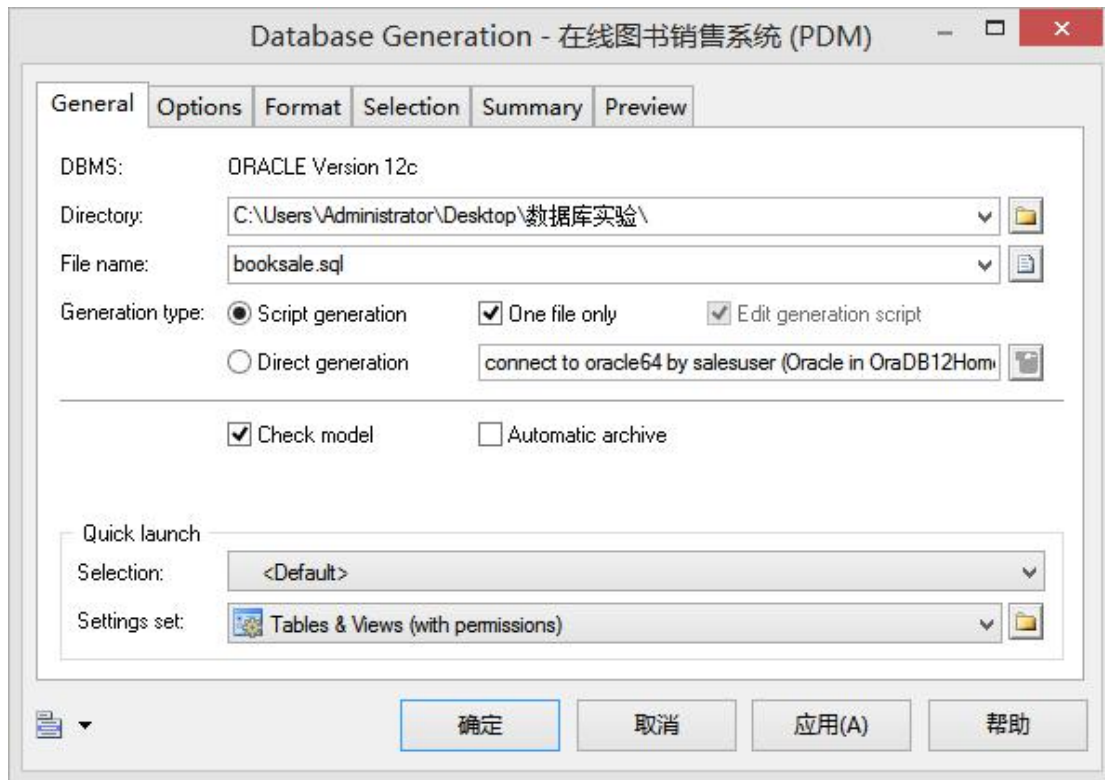
图书信息模块 PDM 模型



图书订单模块 PDM 模型

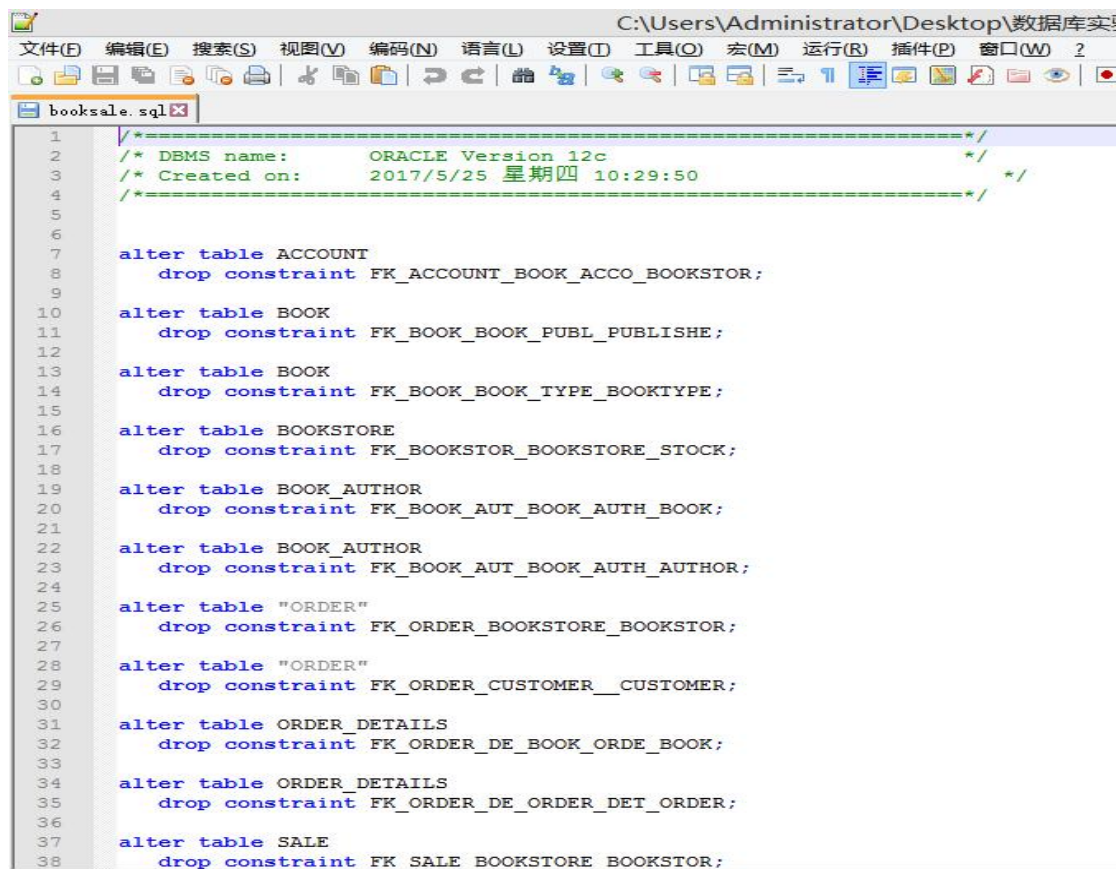


图书销售模块 PDM 模型

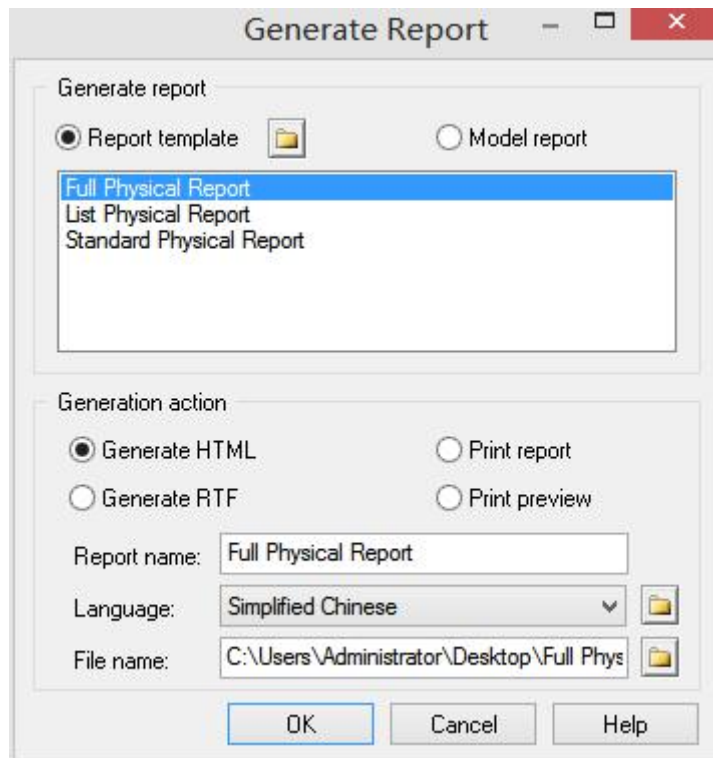


将 PDM 转换为目标数据库相应的 sql 语句

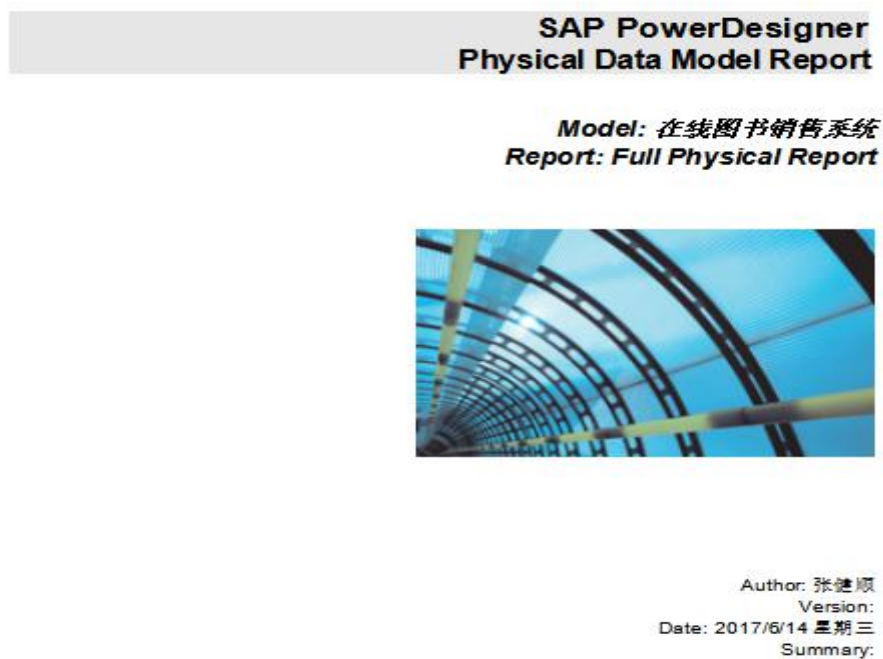
(sql 语句文件见附件)



生成的 sql 文件截图



生成相应的模型报告
(报告文档见附件)



报告封面截图

九、总结及心得体会

①、本次实验的核心是数据库与 PowerDesigner 的连接，并生成相应的 sql 文件，在 Oracle 中的 SQL Developer 里再进行 sql 文件的导入，创建相应的数据库和数据表；

②、完善逻辑模型之后，再进行 LDM 到 PDM 的转换，充分利用 PowerDesigner 这款建模工具；

③、生成相应的 sql 文件和报告文件时可以根据实际情况需要进行相应的配置；

④、报告文件中的内容不仅包含了模型，还包含了数据表的详细信息，以及数据表之间的关系，十分详细。

⑤、生成 sql 文件时注意数据库名和数据表名可能会带来的影响，也可以通过配置相应的格式取消引号的影响。

十、对本实验过程及方法、手段的改进建议

希望能在实验过程中了解 PowerDesigner 这款优秀的建模工具将物理模型转换成相应的 sql 语句的原理，以及希望能更进一步理解三种模型（CDM, LDM, PDM）之间的异同。除此之外，对于生成的报告在实际开发中的应用希望有一个大致的了解。

电子科技大学

实验报告

学生姓名：张健顺 学号：20152201303022 指导教师：陈安龙

实验地点：学院楼 房间：306 实验时间：2017.05.22

一、实验三：在线图书销售系统数据库 SQL 访问操作

二、实验学时：5

三、实验目的

针对在线图书销售数据库开发，了解 ORACLE 对 SQL 语言的支持，掌握 DDL、DML、DQL、DCL 类型语句在数据库操作访问中的应用方法，培养数据库 SQL 编程能力。同时也掌握基本的数据库触发器、存储过程编程方法，培养数据库后端编程能力。本实验完成在线图书销售系统数据库的 SQL 数据操作访问和后端数据处理功能。

四、实验原理

利用 ORACLE 数据库平台对相应的数据库进行有关的基本配置，并进行相关增删改查的基本操作以及创建数据库对象和部分高级用法，譬如触发器存储过程

五、实验内容

使用 SQL Developer 数据库开发工具对在线图书销售系统数据库进行数据操作访问，并完成触发器、存储过程后端编程，具体实验内容如下：

- 1) 在线图书销售系统数据库方案创建及配置。
- 2) 在线图书销售系统数据库方案中创建数据库对象（表、视图、索引、序列等）的 SQL 操作。
- 3) 在线图书销售系统数据库表的数据增、删、查、改 SQL 操作。
- 4) 在线图书销售系统数据库的触发器编程，以实现业务规则数据处理。
- 5) 在线图书销售系统数据库的存储过程编程，以实现数据库后端数据处理。

在实验计算机上，利用 SQL Developer 数据库系统工具，完成在线图书销售系统数据库操作访问，同时记录实验过程的规划、操作、运行结果界面等数据，为撰写实验报告提供素材。

六、实验器材（设备、元器件）

“数据库原理及应用”实验所涉及的机房硬件设备为 pc 计算机、服务器以及网络环境，pc 计算机与服务器在同一局域网络。

操作系统： Windows 8.1 / Windows XP

数据库产品软件工具： Oracle 11g/Oracle Database 12c

七、实验步骤

使用 Oracle SQL Developer 数据库系统工具开展在线图书销售系统数据库操作访问，其步骤如下：

1) 在现有 ORACLE 数据库中，创建一个方案对象 BOOKSALE，并进行必要配置。

2) 在 BOOKSALE 方案中，通过执行 SQL 语句创建在线图书销售系统数据库的表、视图、索引、序列等对象。

3) 在线图书销售系统数据库表中，通过执行 SQL 语句实现数据增、删、查、改 SQL 操作。

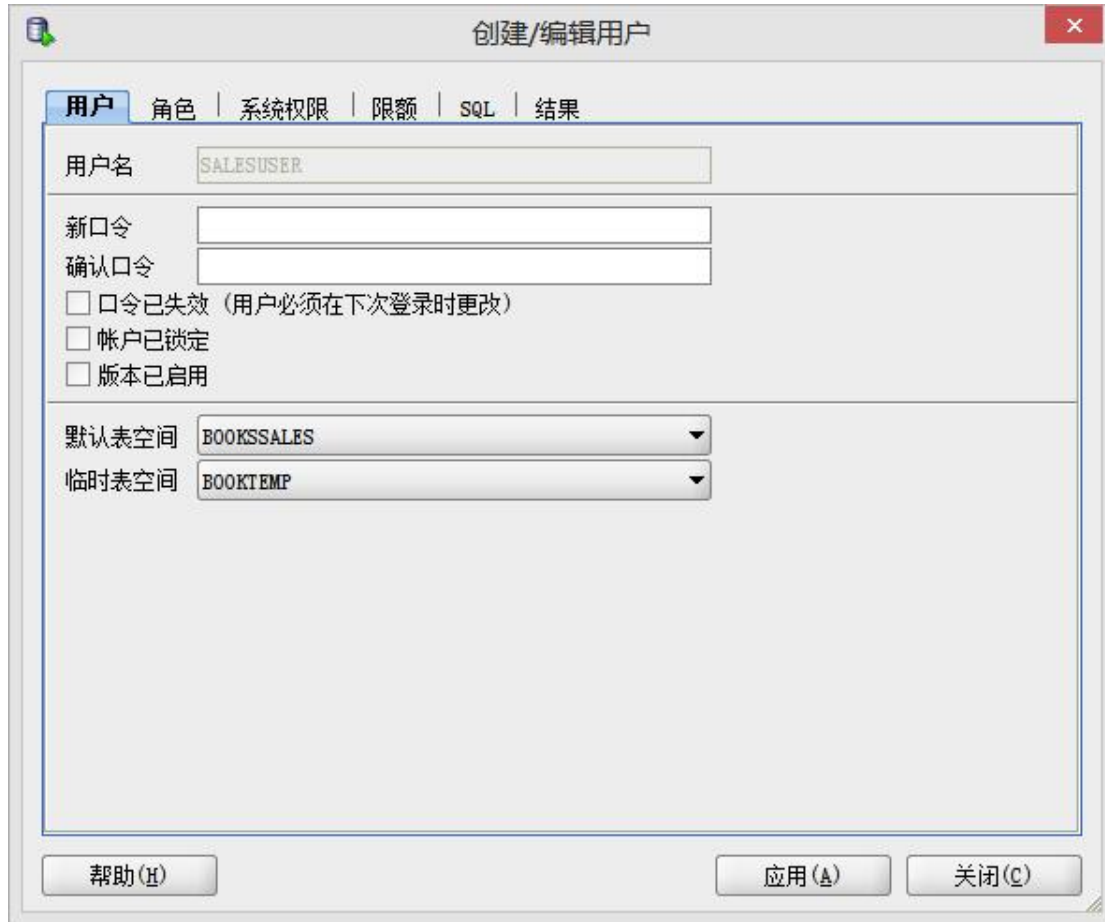
4) 在 BOOKSALE 方案中，通过执行 SQL 语句实现在线图书销售系统数据库的触发器，以实现业务规则（如依据图书销售数据对图书库存数据进行同步修改）数据处理。

5) 在 BOOKSALE 方案中，通过执行 SQL 语句实现在线图书销售系统数据库的存储过程，以实现数据库后端数据（如图书销售统计）处理。

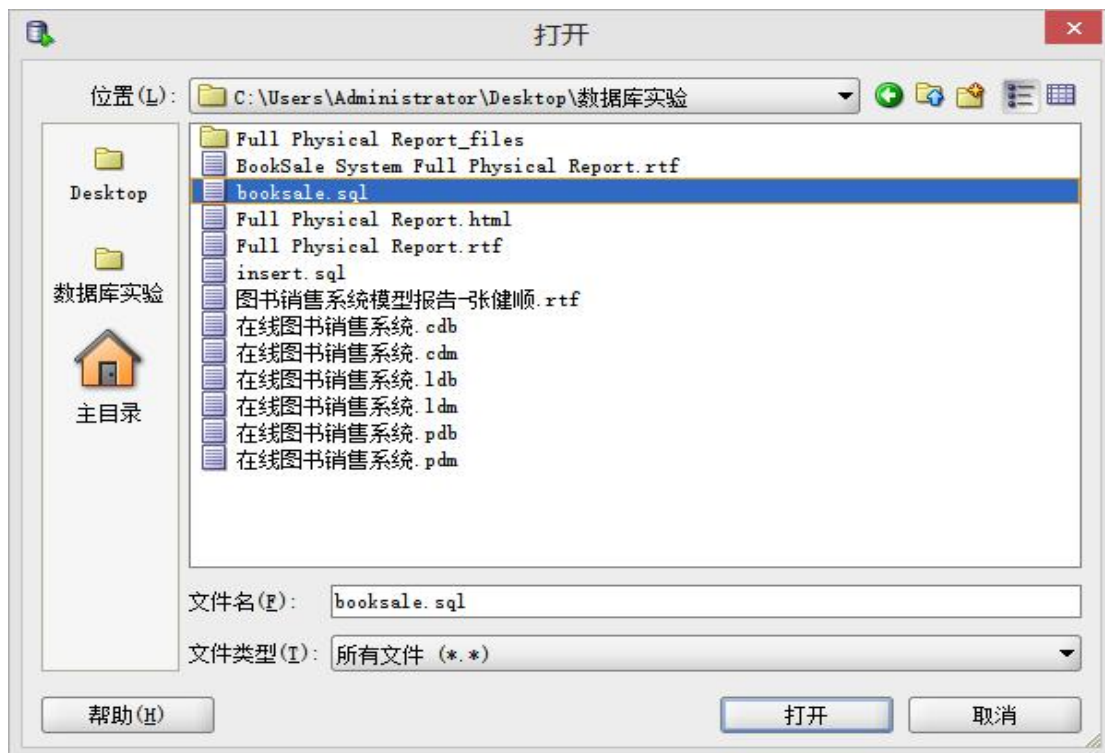
八、实验结果与分析

创建一个方案对象 BOOKSALE，并进行必要配置。

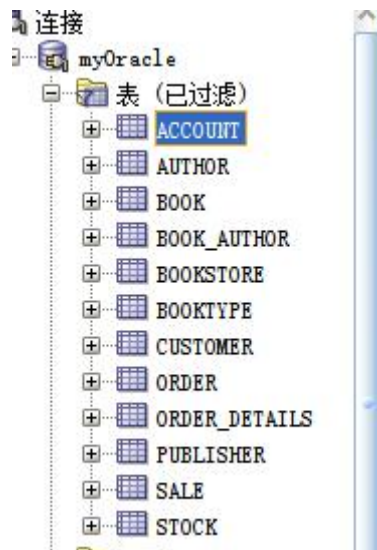
```
CREATE TABLESPACE bookSales DATAFILE 'D:\myOracleDB\bookSales.DBF' SIZE 50M;--创建表空间
CREATE TEMPORARY TABLESPACE bookTemp TEMPFILE 'D:\myOracleDB\bookTemp.DBF' SIZE 20M;--创建临时表空间
CREATE USER salesuser IDENTIFIED BY user
  DEFAULT TABLESPACE bookSales
  TEMPORARY TABLESPACE bookTemp
  ACCOUNT UNLOCK;--创建用户对象
grant dba to salesuser;--将 dba 权限授予给创建的用户对象
```



配置用户



导入 PowerDesigner 生成的 sql 文件



得到相应的表

```
INSERT INTO BOOK VALUES (111111, 1, 1, '三重门
',30,'introduction','content',to_date('2017-6-1 11:36:29','yyyy-mm-dd
hh24:mi:ss'));
INSERT INTO BOOK VALUES (222222, 2, 2, '女权
',33,'introduction1','content1',to_date('2017-6-1
11:37:49','yyyy-mm-dd hh24:mi:ss'));
INSERT INTO BOOK VALUES (333333, 3, 3, '活着
',30,'introduction2','content3',to_date('2017-6-1
11:38:30','yyyy-mm-dd hh24:mi:ss'));
```

```
INSERT INTO book_author VALUES (111111, 1, 1);
INSERT INTO book_author VALUES (222222, 2, 2);
INSERT INTO book_author VALUES (333333, 3, 3);
```

```
INSERT INTO STOCK VALUES (1, 111111, 100, 'operation',to_date('2017-6-1
11:36:29','yyyy-mm-dd hh24:mi:ss'));
INSERT INTO STOCK VALUES (2, 222222, 110, 'operation2',to_date('2017-6-1
11:47:11','yyyy-mm-dd hh24:mi:ss'));
```

```

INSERT INTO STOCK VALUES (3, 333333, 101, 'operation3',to_date('2017-6-1
11:47:16','yyyy-mm-dd hh24:mi:ss'));

INSERT INTO STOCK VALUES (4, 111111, 101, 'operation3',to_date('2017-6-1
11:47:16','yyyy-mm-dd hh24:mi:ss'));


INSERT INTO BOOKSTORE VALUES (1,1, '新华文轩',1888888888,'电子科大');
INSERT INTO BOOKSTORE VALUES (2,1, '新华文轩',1888888888,'电子科大');
INSERT INTO BOOKSTORE VALUES (3,3, '新华文轩',1888888888,'电子科大');


INSERT INTO ACCOUNT VALUES (1,1,0.8,to_date('2017-12-1
11:47:16','yyyy-mm-dd hh24:mi:ss'));
INSERT INTO ACCOUNT VALUES (2,1,0.8,to_date('2017-12-1
11:47:16','yyyy-mm-dd hh24:mi:ss'));
INSERT INTO ACCOUNT VALUES (3,3,0.8,to_date('2017-12-1
11:47:16','yyyy-mm-dd hh24:mi:ss'));


INSERT INTO CUSTOMER VALUES (1,'小白',13000000000,'四川省成都市成华区
','password1','vip');
INSERT INTO CUSTOMER VALUES (2,'小红',13111111111,'四川省成都市郫县
','password2','svip');
INSERT INTO CUSTOMER VALUES (3,'小黑',13222222222,'四川省成都市锦江区
','password3','ssvip');


INSERT INTO "ORDER" VALUES (1,2,1,to_date('2017-12-1
11:47:16','yyyy-mm-dd hh24:mi:ss'),'true','100.25','电子科技大学',101);
INSERT INTO "ORDER" VALUES (2,1,3,to_date('2017-12-1
11:47:16','yyyy-mm-dd hh24:mi:ss'),'true','120.25','西南财经大学',102);
INSERT INTO "ORDER" VALUES (3,3,2,to_date('2017-12-1
11:47:16','yyyy-mm-dd hh24:mi:ss'),'true','45.25','四川大学',103);

```

```

INSERT INTO order_details VALUES (1,3,222222,'女权',3);
INSERT INTO order_details VALUES (2,2,111111,'三重门',2);
INSERT INTO order_details VALUES (3,1,333333,'活着',5);

INSERT INTO SALE VALUES (1,1,222222,1,to_date('2017-12-1
11:47:16','yyyy-mm-dd hh24:mi:ss'),3,100.25);
INSERT INTO SALE VALUES (2,2,111111,1,to_date('2017-12-1
11:27:16','yyyy-mm-dd hh24:mi:ss'),2,120.25);
INSERT INTO SALE VALUES (3,1,333333,2,to_date('2017-12-1
11:37:16','yyyy-mm-dd hh24:mi:ss'),5,45.25);

```

插入数据

```
select * from book; --查询 book 表中的内容
```

查... x

SQL | 提取的所有行: 3, 用时 0.352 秒

	ISBN	PublisherId	TypeId	BookName	BookPrice	Introduction	Content	PubDate
1	111111	1	1	三重门	30	introduction	content	01-6月 -17
2	222222	2	2	女权	33	introduction1	content1	01-6月 -17
3	333333	3	3	活着	29	introduction2	content3	01-6月 -17

```
select * from BOOK where "BookName" LIKE '活%';--查询 bookname 以活开头的
```

查... x

SQL | 提取的所有行: 1, 用时 0.004 秒

	ISBN	PublisherId	TypeId	BookName	BookPrice	Introduction	Content	PubDate
1	333333	3	3	活着	29	introduction2	content3	01-6月 -17

```
UPDATE BOOK SET "BookPrice" = 29 WHERE ISBN = 333333;--修改数据
```

查询结果 x | 脚本输出 x

任务已完成, 用时 0.118 秒

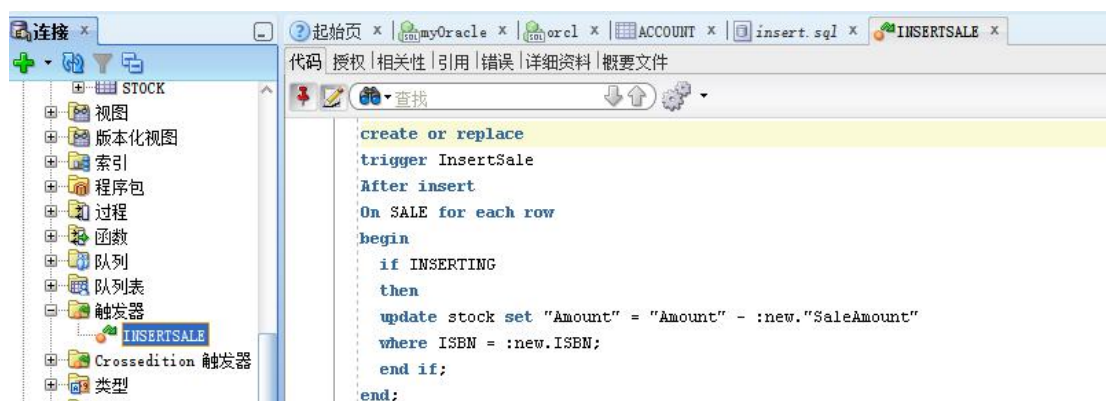
1 行已更新。

简单的数据库操作

```

create or replace
trigger InsertSale
After insert
On SALE for each row
begin
    if INSERTING
    then
        update stock set "Amount" = "Amount" - :new."SaleAmount"
        where ISBN = :new.ISBN;
    end if;
end;

```



创建触发器

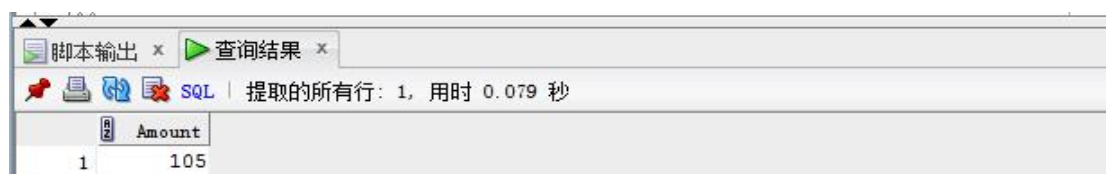
```

//触发器效果验证
select "Amount" from stock where isbn = 222222;

INSERT INTO SALE VALUES (4,1,222222,2,to_date('2017-12-1
11:37:16','yyyy-mm-dd hh24:mi:ss'),5,45.25);

select "Amount" from stock where isbn = 222222;

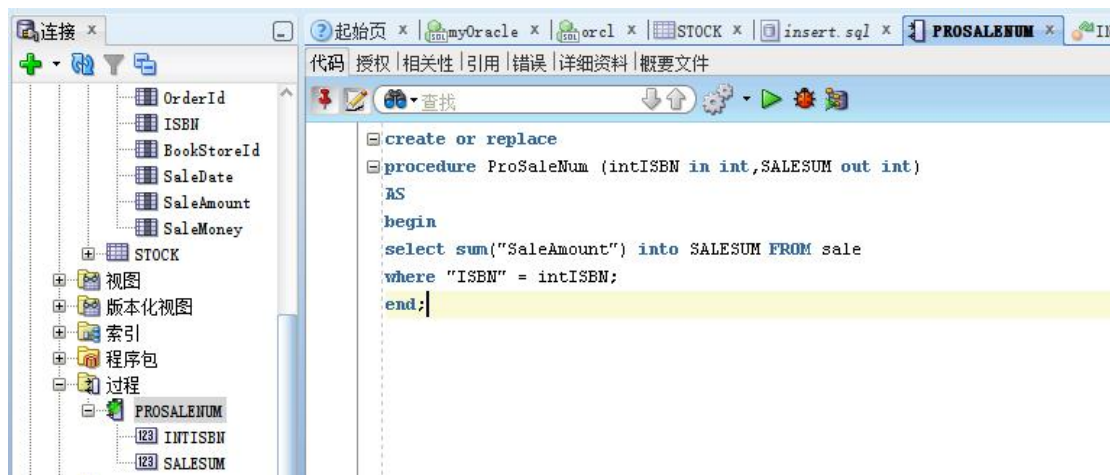
```





触发器效果验证

```
create or replace
procedure ProSaleNum (intISBN in int,SALESUM out int)
AS
begin
select sum("SaleAmount") into SALESUM FROM sale
where "ISBN" = intISBN;
end;
```



存储过程的创建

```

//存储过程效果验证

select * from sale where isbn = 222222;

set serveroutput on;

declare SALESUM int;--申请变量 SALESUM

begin

    prosalenum (222222,SALESUM);--调用存储过程

    dbms_output.put_line(SALESUM);--输出变量值

end;

```



SaleId	OrderId	ISBN	BookStoreId	SaleDate	SaleAmount	SaleMoney
1	1	222222	122	01-12月-17	3	100.25
2	4	222222	2	01-12月-17	5	45.25
3	5	222222	2	01-12月-17	5	45.25

任务已完成, 用时 0.002 秒

匿名块已完成

13

存储过程效果验证

九、总结及心得体会

通过本次实验，了解到了 oracle 数据库基本的增删改查操作，以及触发器和存储过程的创建和使用。通过触发器和存储过程的使用可以有效减少数据库的访问次数，同时实现多条 sql 语句执行的相同的效果，大大提高了效率。

十、对本实验过程及方法、手段的改进建议

希望能在实验过程中能了解到存储过程和触发器等高级数据库操作的应用场景，便于理解和熟悉使用规则。

电子科技大学

实验报告

学生姓名：张健顺 学号：20152201303022 指导教师：陈安龙

实验地点：学院楼 房间：306 实验时间：2017.06.01

一、实验四：在线图书销售系统 Oracle 数据库管理

二、实验学时：5

三、实验目的

通过本上机实验，熟悉典型数据库管理工具使用，掌握应用系统的数据库管理技术与方法，如数据库角色管理、用户管理、权限管理、数据库备份与恢复等系统管理操作，以及完成数据库对象操作访问。

四、实验原理

ORACLE 数据库平台为用户提供了功能强大的 GUI 界面方便用户对数据库进行相应的管理。

五、实验内容

使用 Oracle SQL Developer 数据库开发工具对在线图书销售系统数据库进行数据库管理，并完成数据库备份/恢复处理，具体实验内容如下：

- 1) 在线图书销售系统数据库所有者用户（BOOKSALE）创建
- 2) 在线图书销售系统数据库角色创建
- 3) 在线图书销售系统数据库角色权限设置
- 4) 在线图书销售系统数据库其他用户创建与角色分派
- 5) 在线图书销售系统用户的数据库对象访问
- 6) 在线图书销售系统数据库备份管理
- 7) 在线图书销售系统数据库恢复管理

在实验计算机上，利用 Oracle SQL Developer 数据库系统工具，完成在线图书销售系统数据库管理与维护，同时记录实验过程的规划、操作、运行结果界面等数据，为撰写实验报告提供素材。

六、实验器材（设备、元器件）

“数据库原理及应用”实验所涉及的机房硬件设备为 pc 计算机、服务器以及网络环境，pc 计算机与服务器在同一局域网络。

操作系统： Windows 8.1 / Windows XP

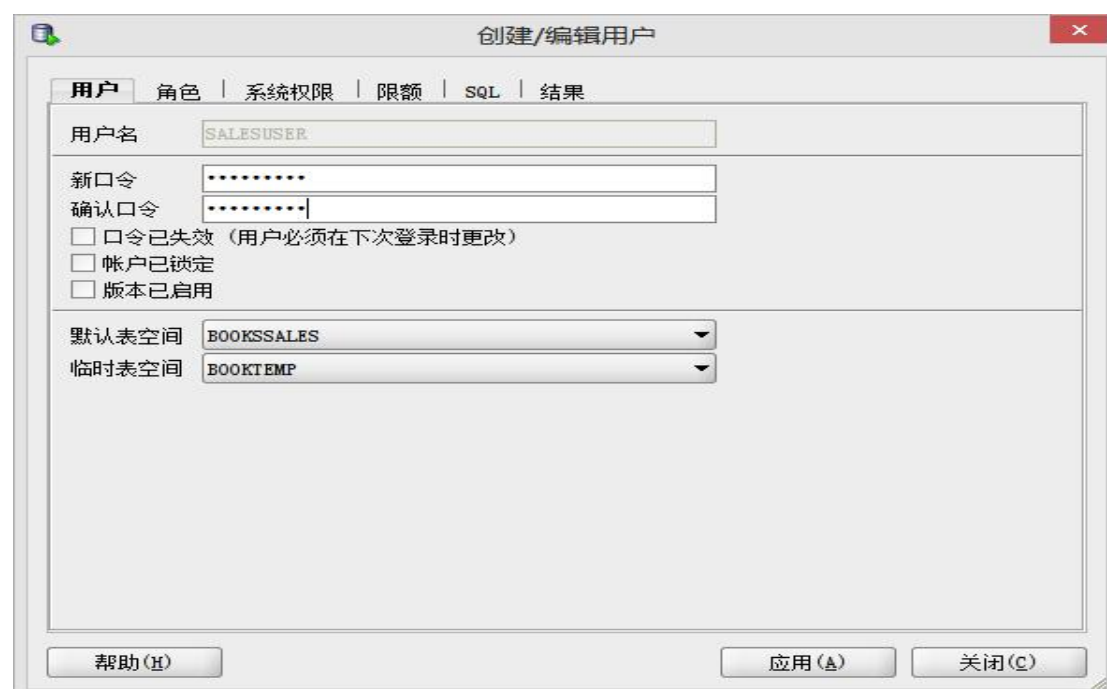
数据库产品软件工具： Oracle 11g/Oracle Database 12c

七、实验步骤

使用 Oracle SQL Developer 数据库系统工具开展在线图书销售系统数据库管理与维护，其步骤如下：

- 1) 创建数据库所有者用户 BOOKSALE
- 2) 设置 BOOKSALE 数据库用户的角色与权限
- 3) 在 BOOKSALE 方案中实现在线图书销售系统数据库对象创建
- 4) 创建在线图书销售系统数据库客户、经理角色
- 5) 分派创建角色的权限
- 6) 创建在线图书销售系统用户
- 7) 分派创建用户的角色
- 8) 验证创建用户的数据库对象访问权限
- 9) 数据库备份管理
- 10) 数据库恢复管理

八、实验结果与分析



创建/编辑用户

用户 | 角色 | 系统权限 | 限额 | SQL | 结果

用户名: SALESUSER

新口令:

确认口令:

☐ 口令已失效 (用户必须在下次登录时更改)

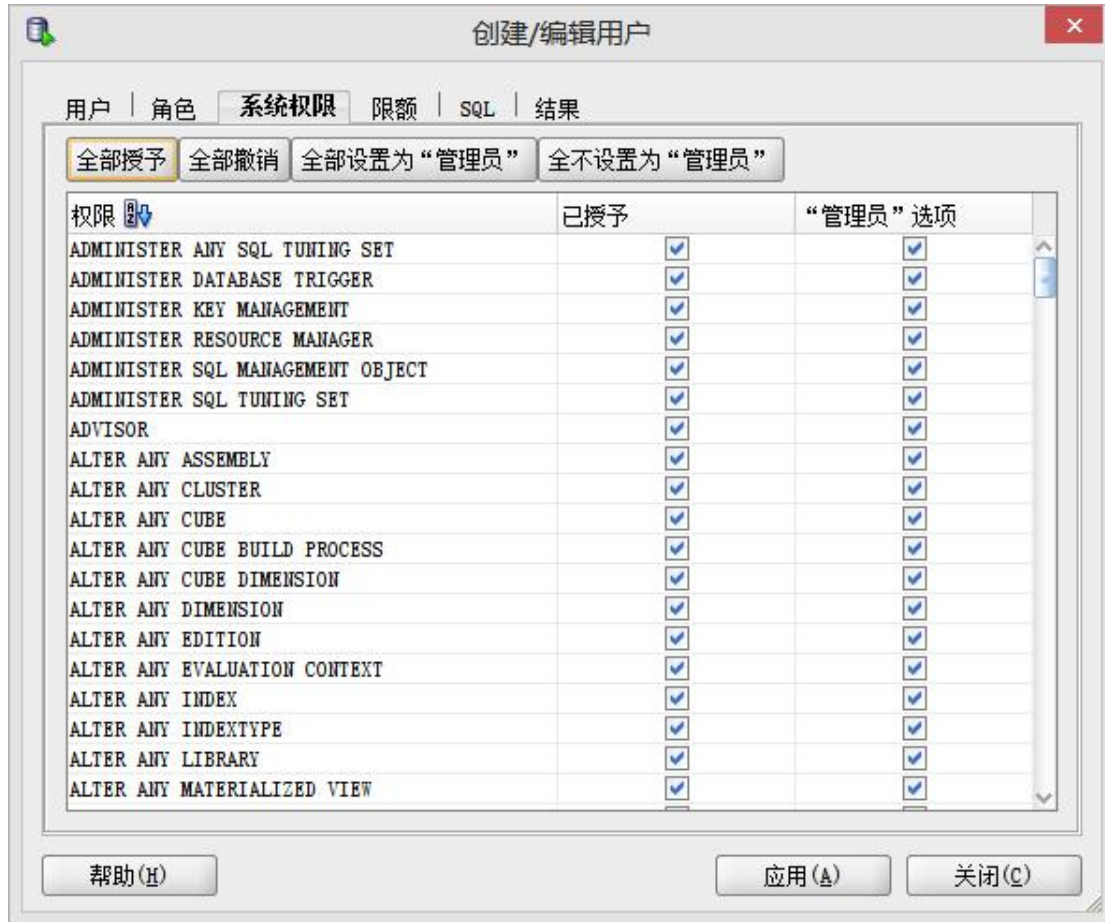
☐ 帐户已锁定

☐ 版本已启用

默认表空间: BOOKSSALES

临时表空间: BOOKTEMP

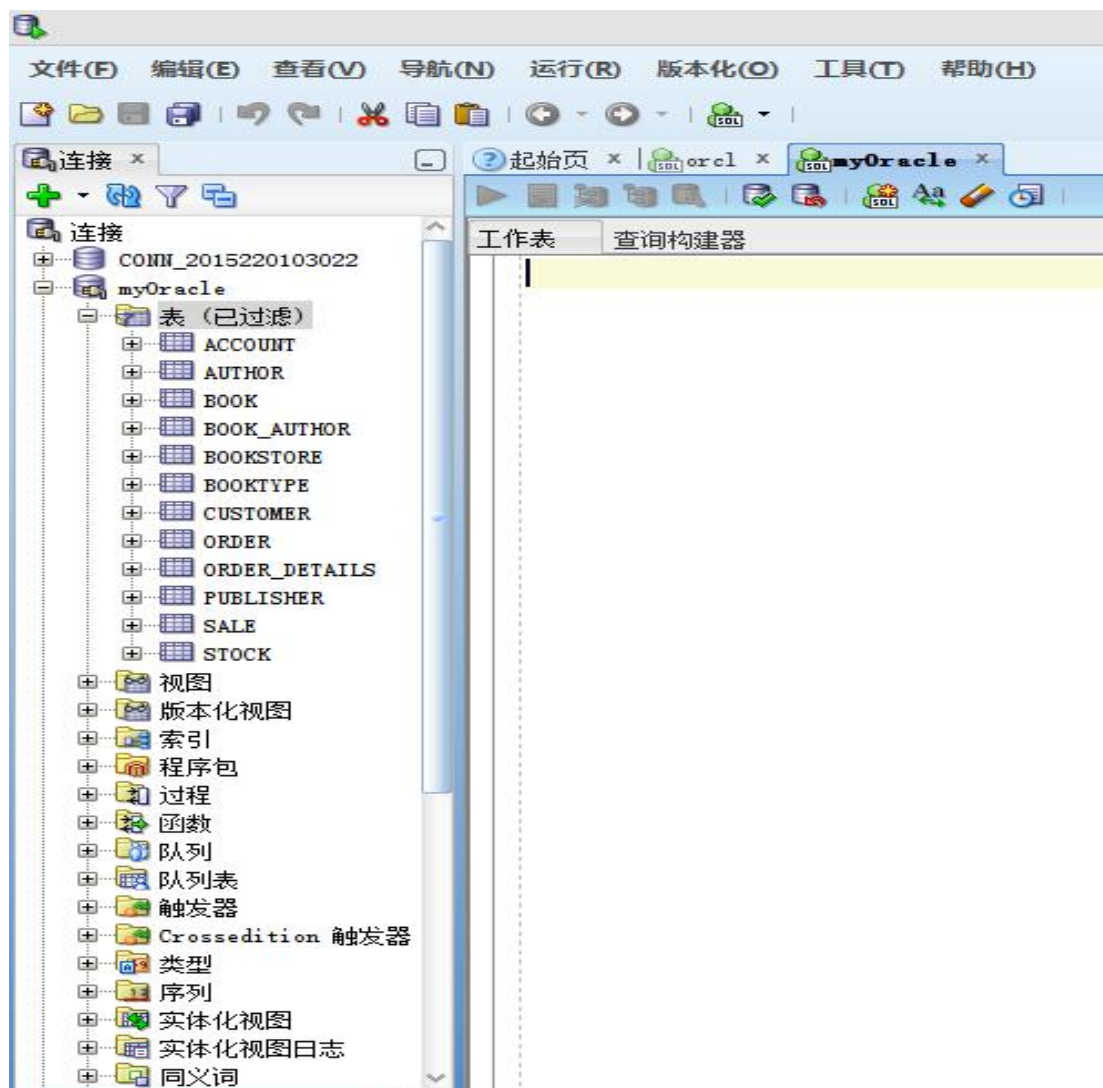
帮助(H) | 应用(A) | 关闭(C)



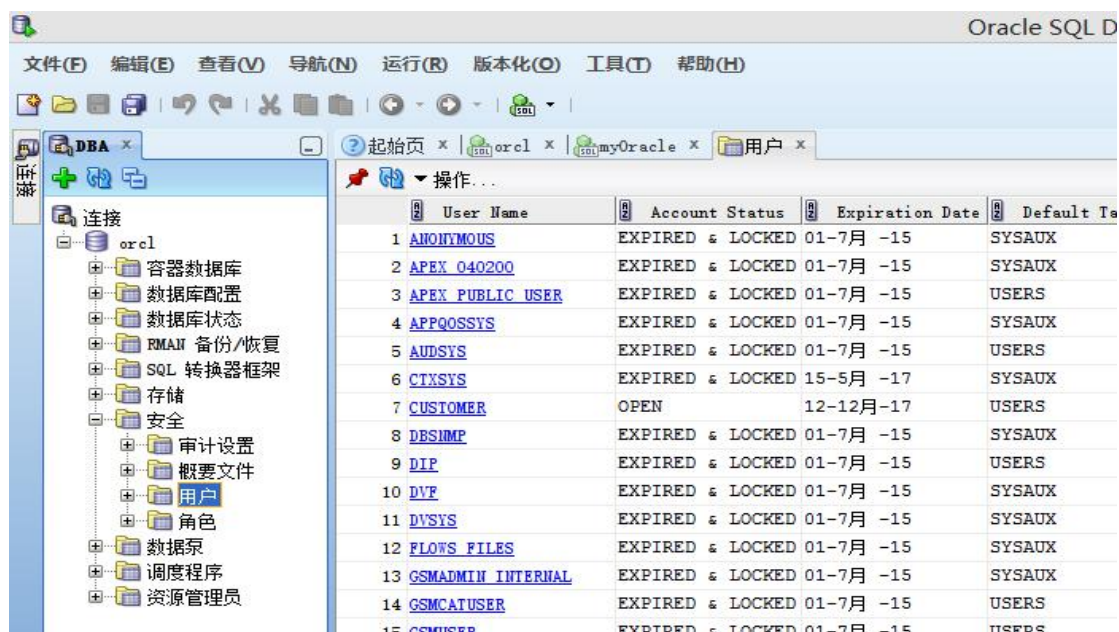
权限管理



登录对应的用户所创建的连接



数据库对象



DBA 视图下目录

创建角色

角色 | 授予的角色 | 系统权限 | SQL

角色名: Manager

新口令:

确认口令:

创建角色

授予角色

属性 | SQL

角色名: RESOURCE

是管理员: ☐

授予角色

授予系统权限

属性 | SQL

```
GRANT ADMINISTER DATABASE TRIGGER TO "MANAGER";
GRANT DELETE ON "SALESUSER"."AUTHOR" TO "MANAGER";
GRANT UPDATE ON "SALESUSER"."AUTHOR" TO "MANAGER";
GRANT INSERT ON "SALESUSER"."AUTHOR" TO "MANAGER";
```

授予系统权限

创建用户

用户 | 授予的角色 | 系统权限 | 限额 | SQL

用户名: USER1

新口令:

确认口令:

☐ 口令已失效 (用户必须在下次登录时更改)

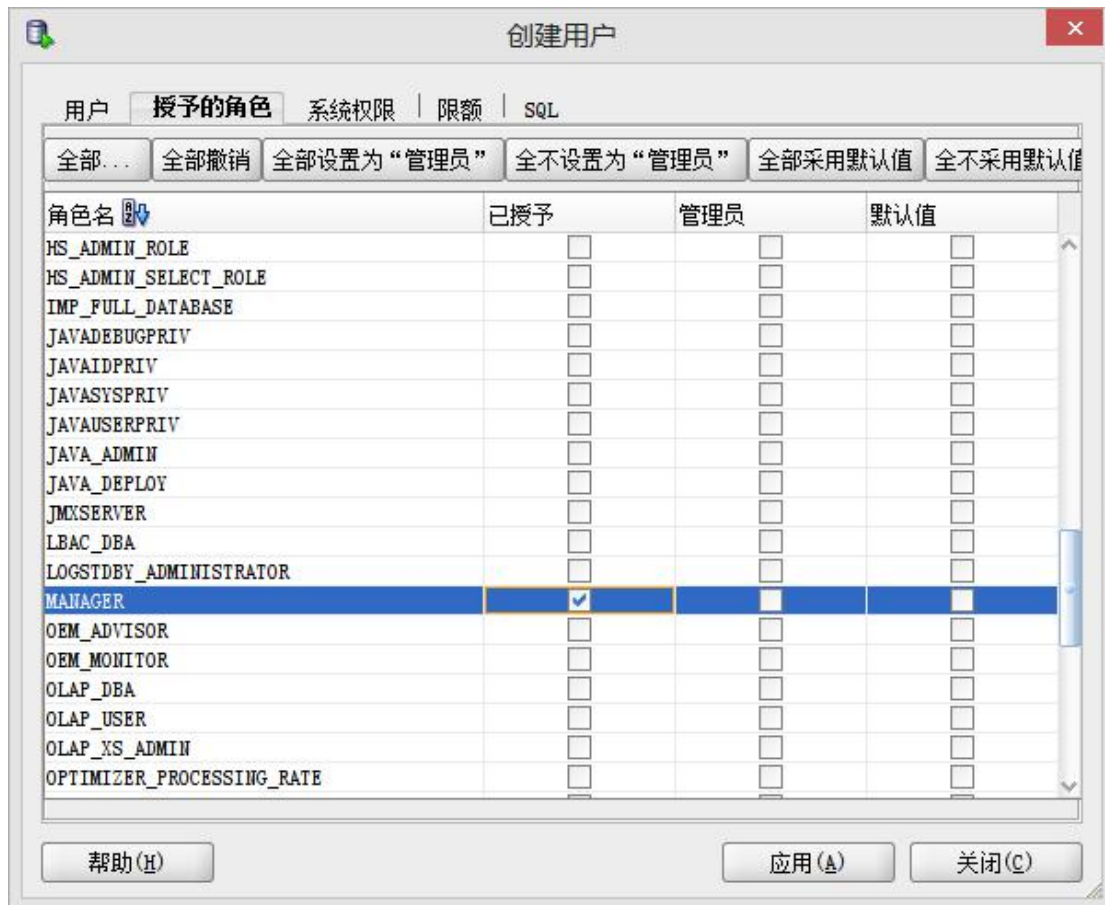
☐ 帐户已锁定

☐ 版本已启用

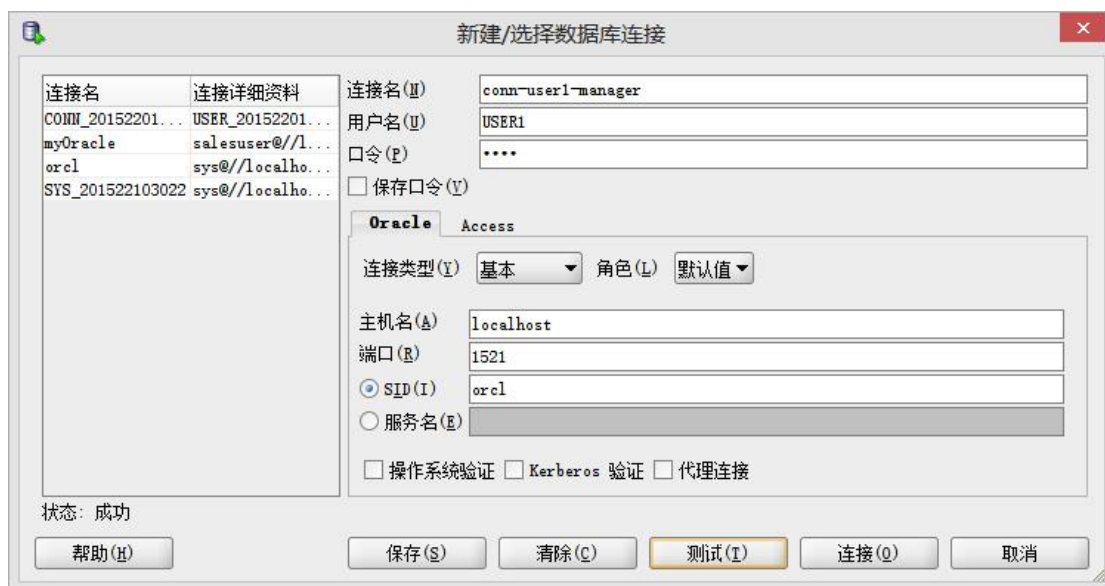
默认表空间: USERS

临时表空间: TEMP

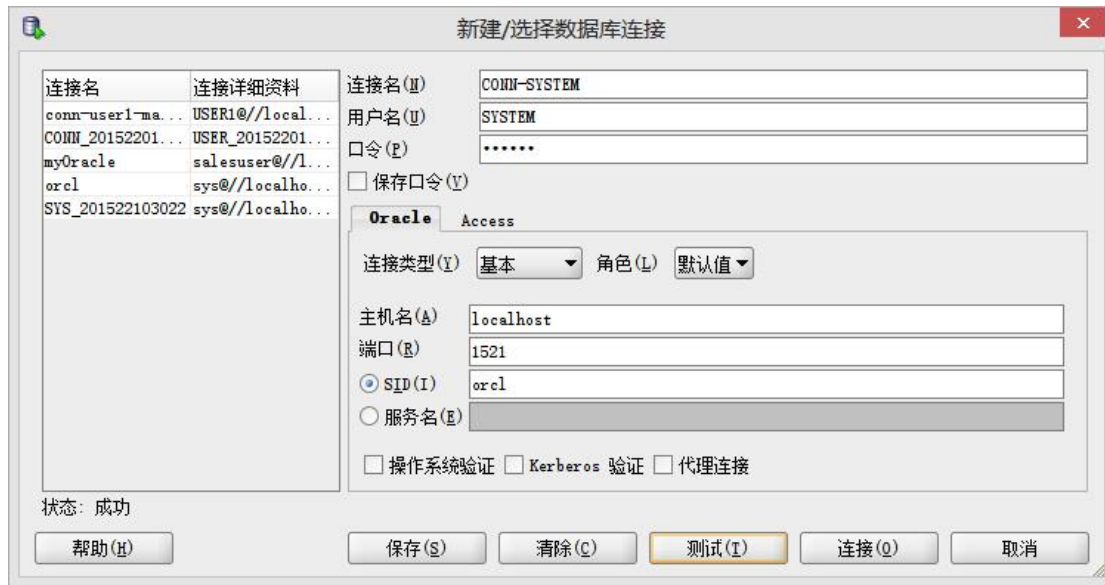
创建用户



授予 MANAGER 角色



创建连接



创建 system 用户对应的连接



SQL Developer 的数据泵导出设置

导出向导 - 步骤 4/8

表数据

源
方案
过滤器
表数据
选项
输出文件
作业调度
概要

名称(A): %
方案(S): SALESUSER
类型(Y): TABLE

数据库对象	列	对象 Where 子句
SALESUSER.ACCOUNT	*	
SALESUSER.AUTHOR	*	
SALESUSER.BOOK	*	
SALESUSER.BOOKSTORE	*	
SALESUSER.BOOKTYPE	*	

全局 Where 子句:

选取备份的数据库表

导出向导 - 步骤 5/8

选项

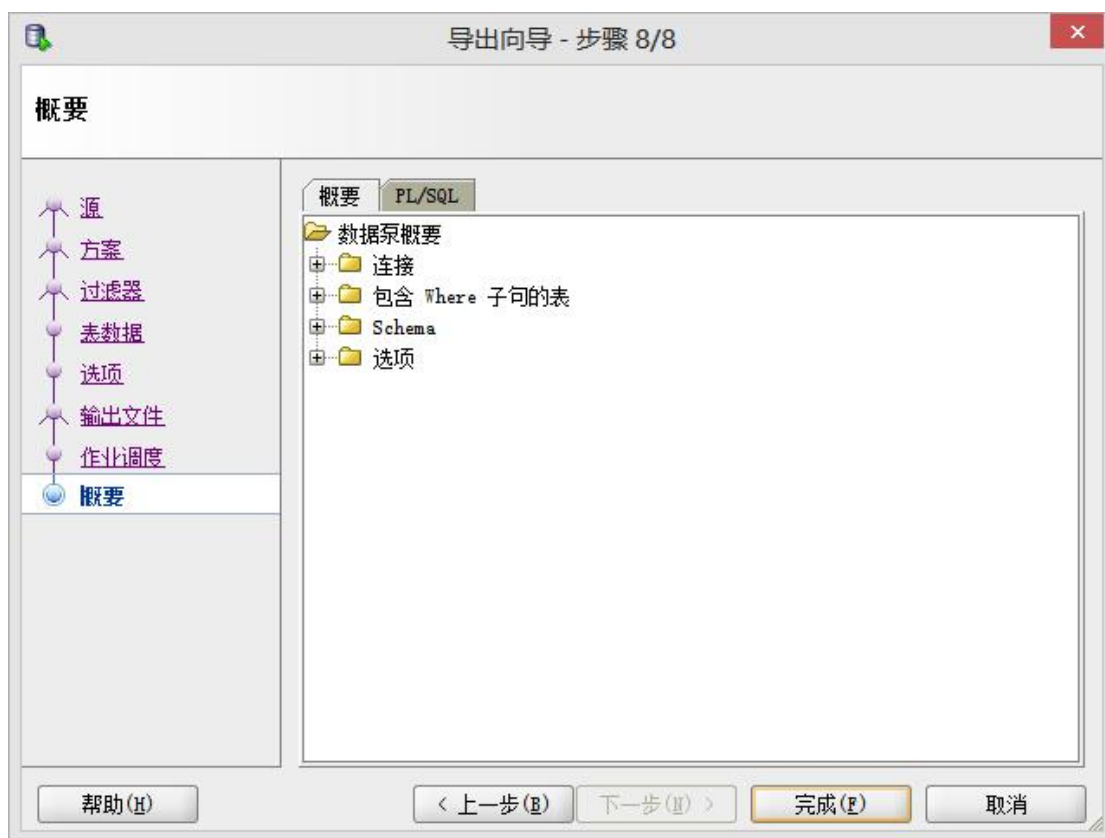
源
方案
过滤器
表数据
选项
输出文件
作业调度
概要

线程号
估计 (尚未计算)
☒ 启用日志记录
日志文件
☐ 导出数据的读取一致性视图
按 SCN
按日期

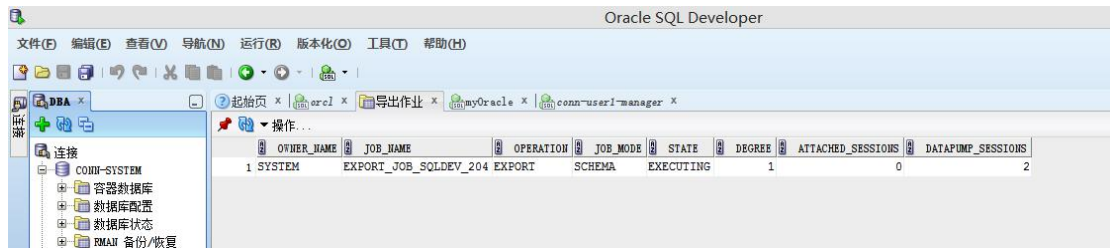
备份选项设置



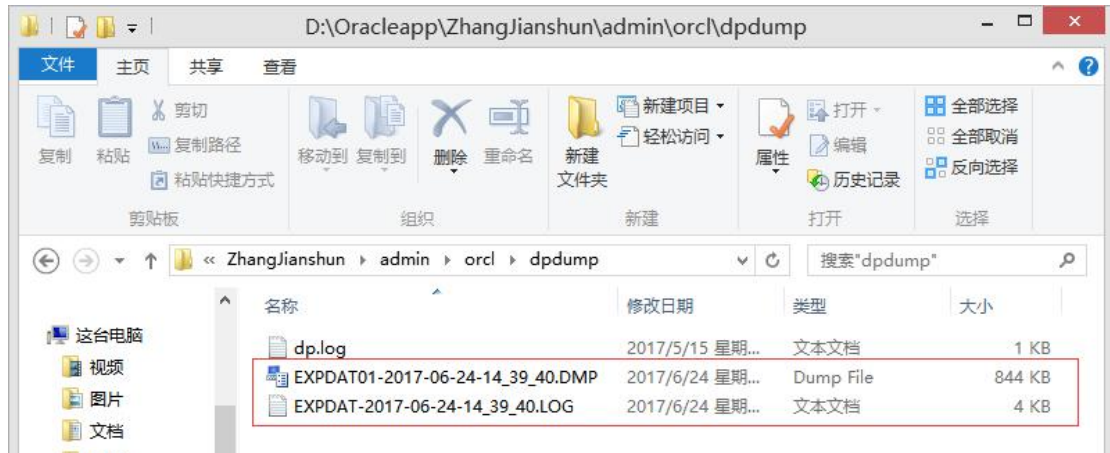
输出文件设置



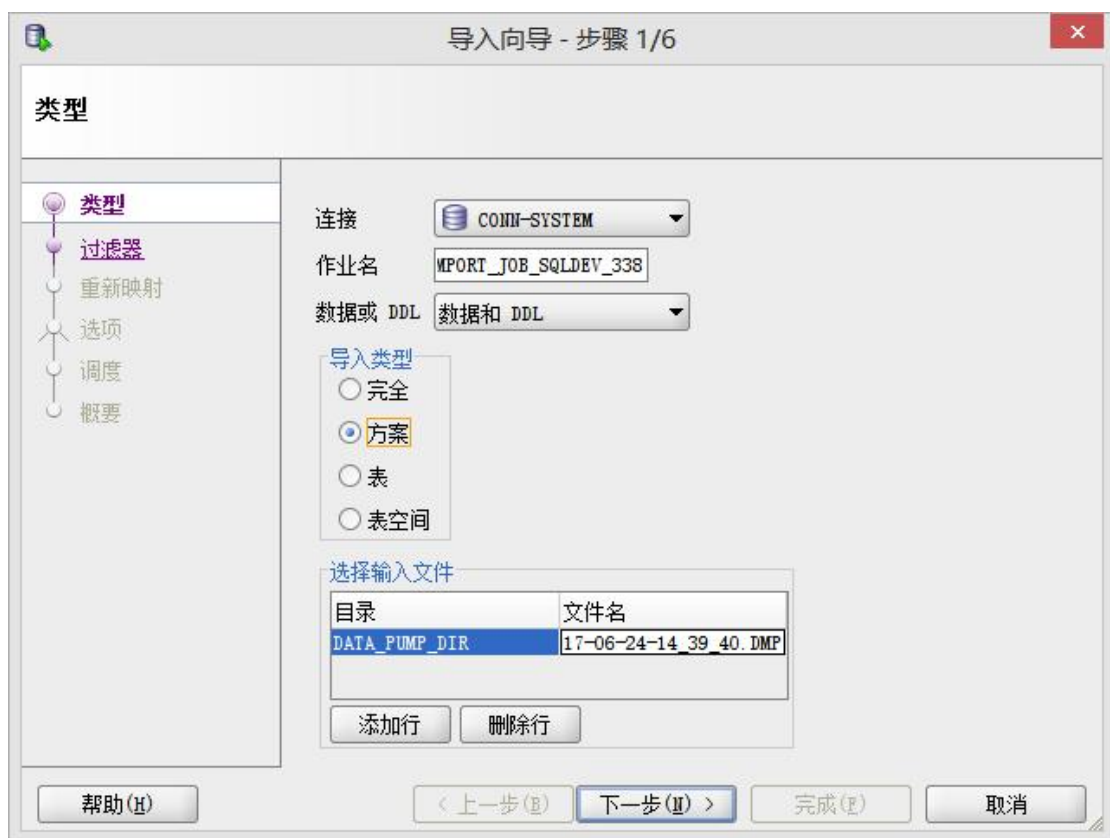
概要



备份调度执行



导出的文件



导入类型设置



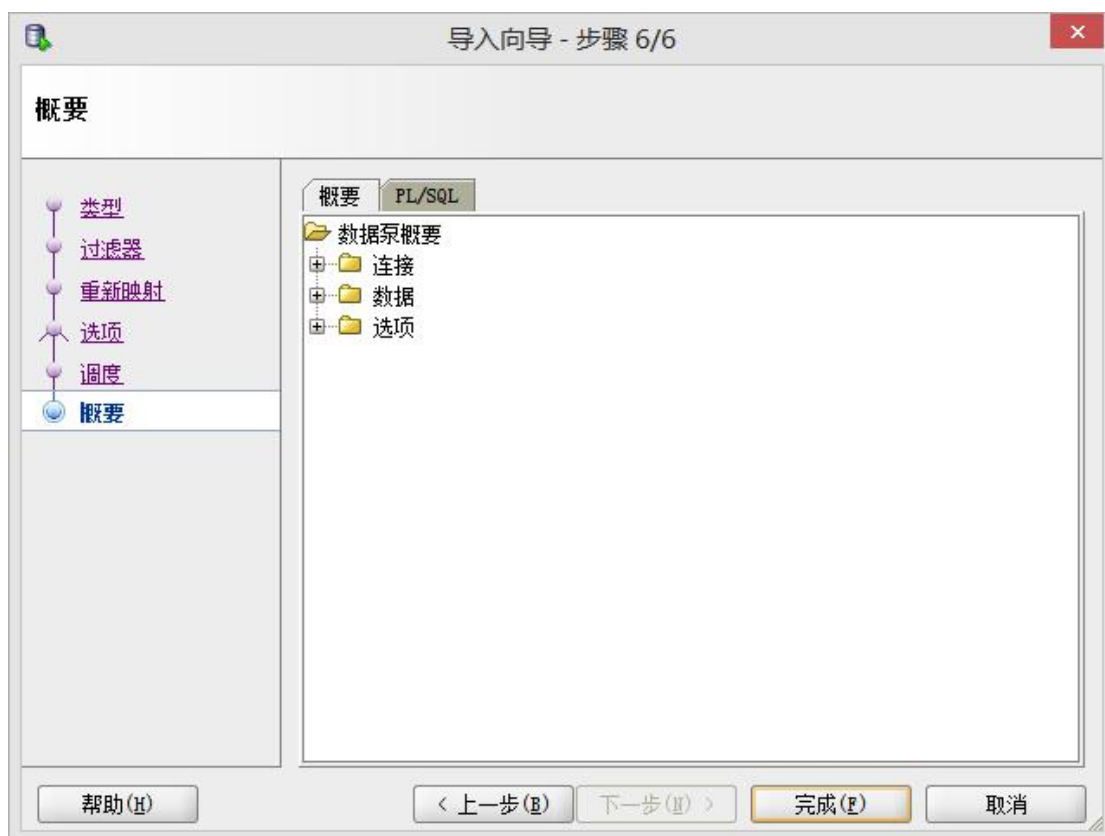
方案导入过滤设置



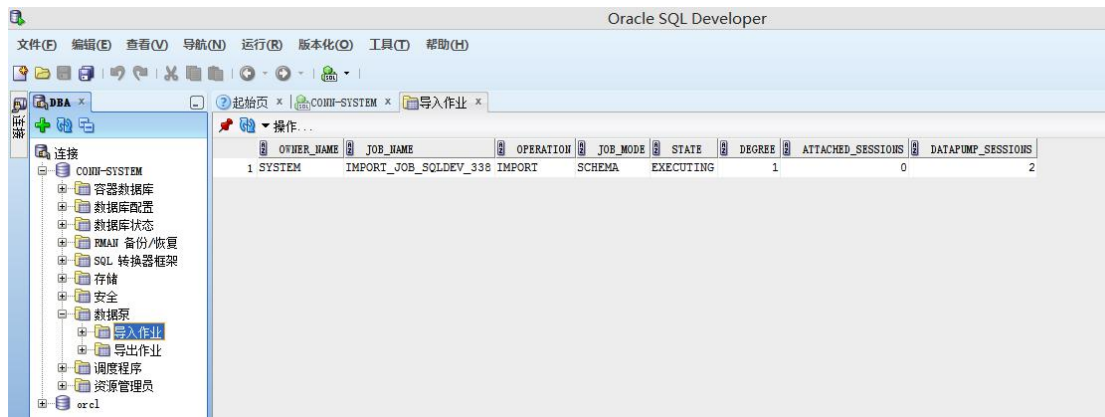
重新映射设置



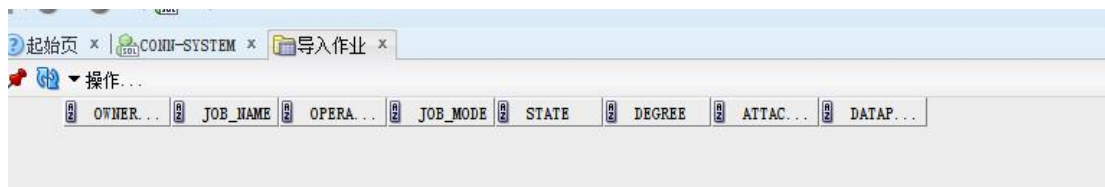
选项设置



概要



导入作业执行



执行结束



实验过程中发现的错误，仔细排查错误后发现，如果采用默认配置，其中的文件名和一开始导出的文件名（或者本地磁盘上存在的文件名）是不一致的，所以需要修改默认配置中的文件名为本地文件的文件名。否则会报错。

九、总结及心得体会

通过本次实验，了解到对数据库除了增删改查的一系列对数据库中的数据的基本操作以外还有很多关于数据库管理的相关知识和技巧。权限、角色等重要数据库对象的管理以及数据库的备份和恢复（导入和导出）操作。

十、对本实验过程及方法、手段的改进建议

希望能结合实际的开发中的应用场景进一步地阐释数据库的管理扮演的重要角色，便于更深入地理解数据库管理的必要性。

电子科技大学

实验报告

学生姓名：张健顺 学号：20152201303022 指导教师：陈安龙

实验地点：学院楼 房间：306 实验时间：2017.06.15

一、实验五：在线图书销售系统数据库 web 应用编程访问

二、实验学时：6

三、实验目的

掌握基本的 WEB 应用程序连接数据库、数据查询访问的编程实现方法。理解 WEB 应用程序结构，以及应用程序对数据库的操作访问方法。掌握 WEB 环境部署、JDBC 标准接口、数据库连接、JSP 页面编程等基本技术方法，培养数据库应用的基本编程访问能力。

四、实验原理

利用 Java 提供的 JDBC 组件包来建立程序和数据库之间的联系，以及利用相关操作来对数据库进行数据上的管理（增删改查）。

五、实验内容

针对所设计实现的在线图书销售系统数据库进行 web 编程访问，并完成基本功能的 JSP 页面程序，具体实验内容如下：

- 1) 在线图书销售系统 WEB 应用开发环境配置
- 2) 在线图书销售系统数据库的访问连接代码编写
- 3) 在线图书销售系统的出版社（PUBLISHER）信息添加 JSP 页面代码编写
- 4) 在线图书销售系统的出版社（PUBLISHER）信息查看 JSP 页面代码编写
- 5) 在线图书销售系统的出版社（PUBLISHER）信息修改 JSP 页面代码编写
- 6) 在线图书销售系统的出版社（PUBLISHER）信息删除 JSP 页面代码编写

在实验计算机上，利用应用开发工具，完成在线图书销售系统数据库表的功能编程访问实现。同时记录实验过程的编程、操作、运行结果界面等数据，

为撰写实验报告提供素材。

六、实验器材（设备、元器件）

“数据库原理及应用”实验所涉及的机房硬件设备为 pc 计算机、服务器以

及网络环境，pc 计算机与服务器在同一局域网络。

操作系统： Windows 8.1 / Windows XP

数据库产品软件工具： Oracle 11g/Oracle Database 12c

开发环境： Eclipse luna、Tomcat

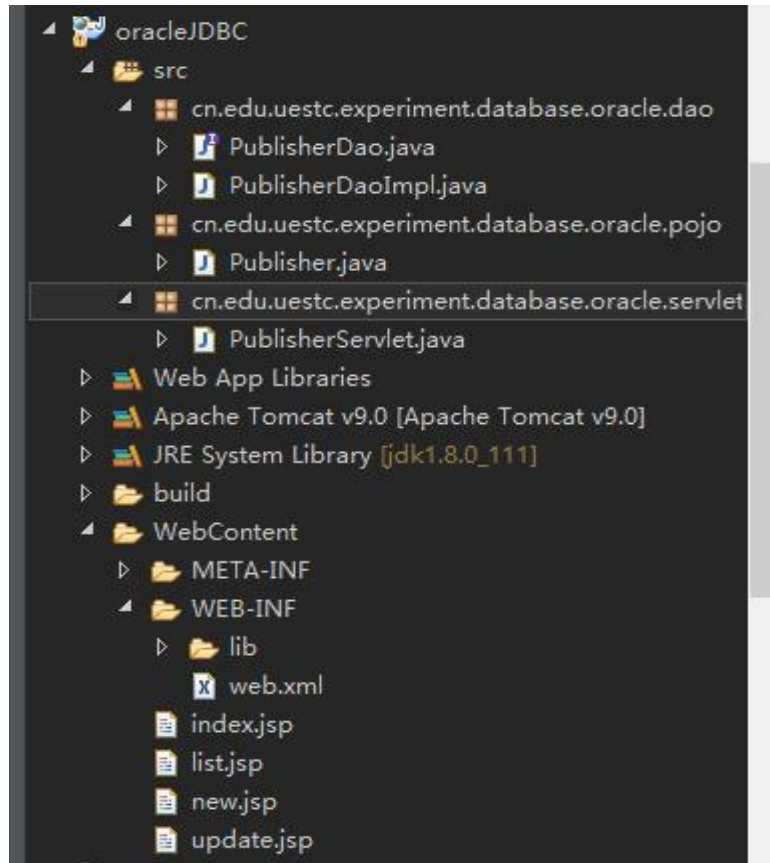
七、实验步骤

在 Eclipse 开发环境中，使用 JSP 编程技术，完成在线图书销售系统的基本功能页面开发，其步骤如下：

- 1) Eclipse 开发环境配置
- 2) 在线图书销售系统数据库的访问连接代码编写
- 3) 在线图书销售系统的出版社（PUBLISHER）信息添加 JSP 页面代码编写
- 4) 在线图书销售系统的出版社（PUBLISHER）信息查看 JSP 页面代码编写
- 5) 在线图书销售系统的出版社（PUBLISHER）信息修改 JSP 页面代码编写
- 6) 在线图书销售系统的出版社（PUBLISHER）信息删除 JSP 页面代码编写

八、实验结果与分析

采用 MVC 设计模式完成本次实验。（Model-View-Controller），Model 即为 Publisher.java, View 即为 .jsp 文件, Controller 即为 PublisherServlet.java。



项目结构

其中对数据库的访问操作使用对 Oracle 的 JDBC 方式,导入 ojdbc 的 jar 包。对数据层（持久层）采用 DAO 设计模式, PublisherDao.java 提供相关数据操作接口, 相关实现放在 PublisherDaoImpl.java 中, 而在 PublisherServlet 中则是逻辑处理的相关操作, WEB-INF 下的 .jsp 文件呈现相关页面。

PublisherDao 提供相应的增删改查接口:

```
public interface PublisherDao {  
    void save(Publisher s) throws Exception;  
    void update(Publisher s) throws Exception;  
    List<Publisher> findAll() throws Exception;  
    void delete(int id) throws Exception;  
    Publisher findById(int id) throws Exception;  
}
```

PublisherDaoImpl 实现具体操作:

```
public class PublisherDaoImpl implements PublisherDao {  
  
    // 数据库相关常量（可移动到接口中）  
    public final static String URL  
        ="jdbc:oracle:thin:@localhost:1521:orcl";  
    public final static String USERNAME = "salesuser";  
    public final static String PASSWORD = "salesuser";  
    public final static String DRIVER =  
"oracle.jdbc.driver.OracleDriver";  
  
    // 数据库连接操作，获取数据库连接  
    public static Connection connect() throws Exception {  
        Class.forName(DRIVER);  
        Connection connection = DriverManager.getConnection(URL,  
        USERNAME, PASSWORD);  
        connection.setAutoCommit(false);  
        return connection;  
    }  
}
```

```
// 主键生成器

public static int getPrimaryKey() throws Exception {
    Connection connection = connect();
    String sql = "select max(\"PublisherId\") from PUBLISHER";
    PreparedStatement preparedStatement =
        connection.prepareStatement(sql);
    ResultSet rSet = preparedStatement.executeQuery();
    int id = 0;
    if(rSet.next()){
        id = rSet.getInt(1);
    }
    rSet.close();
    preparedStatement.close();
    connection.close();
    return id + 1;
}
```

添加出版社信息的 save 方法

```
public void save(Publisher s) throws Exception {
    Connection connection = connect();
    String sql = "insert into PUBLISHER values (?, ?, ?, ?, ?, ?, ?)";
    PreparedStatement preparedStatement =
connection.prepareStatement(sql);
    preparedStatement.setInt(1, getPrimaryKey());
    preparedStatement.setString(2, s.getPublisherName());
    preparedStatement.setLong(3, s.getTelephone());
    preparedStatement.setLong(4, s.getPostCode());
    preparedStatement.setString(5, s.getWebsite());
    preparedStatement.setString(6, s.getCity());
    preparedStatement.setString(7, s.getProvince());
    preparedStatement.executeUpdate();
    preparedStatement.close();
    connection.close();
}
```

修改出版社信息的方法

```
@Override
    public void update(Publisher s) throws Exception {
        Connection connection = connect();
        String sql = "update PUBLISHER set \"PublisherName\"
= ?,\"Telephone\" = ?,\"PostCode\" = ?,\"
        + "\"Website\" = ?,\"City\" = ?,\"Province\" = ? where
\"PublisherId\" = ?";
        PreparedStatement preparedStatement =
connection.prepareStatement(sql);
        preparedStatement.setString(1, s.getPublisherName());
        preparedStatement.setLong(2, s.getTelephone());
        preparedStatement.setLong(3, s.getPostCode());
        preparedStatement.setString(4, s.getWebsite());
        preparedStatement.setString(5, s.getCity());
        preparedStatement.setString(6, s.getProvince());
        preparedStatement.setInt(7, s.getPublisherId());

        preparedStatement.executeUpdate();
        preparedStatement.close();
        connection.close();
    }
```

查询出版社信息列表的方法 findAll

```
public List<Publisher> findAll() throws Exception {
    Connection connection = connect();
    String sql = "select * from PUBLISHER";
    PreparedStatement preparedStatement =
        connection.prepareStatement(sql);
    ResultSet rSet = preparedStatement.executeQuery();
    List<Publisher> publishers = new ArrayList<>();
    Publisher publisher = null;
```

```

while(rSet.next()) {
    publisher = new Publisher();
    publisher.setPublisherId(rSet.getInt(1));
    publisher.setPublisherName(rSet.getString(2));
    publisher.setTelephone(rSet.getLong(3));
    publisher.setPostCode(rSet.getLong(4));
    publisher.setWebsite(rSet.getString(5));
    publisher.setCity(rSet.getString(6));
    publisher.setProvince(rSet.getString(7));
    publishers.add(publisher);
}
rSet.close();
preparedStatement.close();
connection.close();
return publishers;
}

```

删除单个出版社信息的 delete 方法

```

@Override
public void delete(int id) throws Exception {
    Connection connection = connect();
    String sql = "delete * from PUBLISHER where \"PublisherId\"
= ?";

    PreparedStatement preparedStatement =
        connection.prepareStatement(sql);

    preparedStatement.setInt(1, id);

    preparedStatement.executeUpdate();
    preparedStatement.close();
    connection.close();

}

```


查询单个出版社信息（主要用于回显）

```
@Override
public Publisher findById(int id) throws Exception {
    Connection connection = connect();
    String sql = "select * from PUBLISHER where \"PublisherId\"
= ?";
    PreparedStatement preparedStatement =
connection.prepareStatement(sql);
    preparedStatement.setInt(1, id);

    ResultSet rSet = preparedStatement.executeQuery();
    Publisher publisher = null;
    if (rSet.next()) {
        publisher = new Publisher();
        publisher.setPublisherId(rSet.getInt(1));
        publisher.setPublisherName(rSet.getString(2));
        publisher.setTelephone(rSet.getLong(3));
        publisher.setPostCode(rSet.getLong(4));
        publisher.setWebsite(rSet.getString(5));
        publisher.setCity(rSet.getString(6));
        publisher.setProvince(rSet.getString(7));
    }
    rSet.close();
    preparedStatement.close();
    connection.close();
    return publisher;
}
}
```

对于页面展示，采用 jsp 的方式，主要是将查询到的数据展示出来以及需要修改的数据以表单的形式进行相关提交。index.jsp 主要用于跳转，list.jsp 用于展示出版社信息列表，new.jsp 用于添加出版社信息，update.jsp 用于修改出版社信息

Index.jsp

```
<body>
<h1>click 下面啦啦啦啦啦! </h1>
<!-- 通过传递 method 的方法来告知 servlet 调用哪种具体操作 -->
    <a href="myServlet?method=FindAll">查看出版社列表</a>
</body>
```

List.jsp

```
<title>查看全部信息</title>
</head>
<body>
<h2>出版社列表</h2>
<table>
    <tr>
        <th>ID</th>
        <th>Name</th>
        <th>Telephone</th>
        <th>PostCode</th>
        <th>Website</th>
        <th>City</th>
        <th>Province</th>
    </tr>
<c:forEach items="${requestScope.list}" var="Publisher">
    <tr>
        <td>${Publisher.publisherId}</td>
        <td>${Publisher.publisherName}</td>
        <td>${Publisher.telephone}</td>
        <td>${Publisher.postCode}</td>
        <td>${Publisher.website}</td>
        <td>${Publisher.city}</td>
        <td>${Publisher.province}</td>
        <td>
```

```

<!-- 点击 Update 后，调用 servlet 中的 Update 方法，并传递动态参数 id -->
<a
href="myServlet?method=find&id=${Publisher.publisherId }">Update</a>
<a
href="myServlet?method=delete&id=${Publisher.publisherId }">Delete</a
>

        </td>
    </tr>
</c:forEach>
<tr>
    <td colspan="4"><a href="new.jsp">Add</a></td>
</tr>
</table>

```

New.jsp

```

<title>添加</title>
</head>
<body>
<h2>添加学生信息</h2>
<!-- 响应信息 -->
<form action="myServlet" method="post">
<!-- method 作为一个参数名，传递 Add 方法告知 Servlet 属性 hidden -->
<input type="hidden" name="method" value="Add"/>
<table>
    <tr>
        <td>ID</td>
        <td><input type="hidden" name="id" value=0/></td>
    </tr>
    <tr>
        <td>Name</td>
        <td><input type="text" name="name"/></td>
    </tr>

```

```

        <tr>
            <td>Telephone</td>
            <td><input type="text" name="telephone"/></td>
        </tr>
        <tr>
            <td>PostCode</td>
            <td><input type="text" name="postCode"></td>
        </tr>
        <tr>
            <td>Website</td>
            <td><input type="text" name="website"></td>
        </tr>
        <tr>
            <td>City</td>
            <td><input type="text" name="city"></td>
        </tr>
        <tr>
            <td>Province</td>
            <td><input type="text" name="province"></td>
        </tr>
        <tr>
            <td colspan="2">
                <input type="submit" value="submit"/>
                <input type="reset" value="reset"/>
            </td>
        </tr>
    </table>
</form>
</body>

```

Update.jsp

```

<title>修改</title>
</head>
<body>
<h2>修改学生信息</h2>

```

```

<!-- 响应信息 -->
<form action="myServlet" method="post">
<!-- method 作为一个参数名, 传递 Update 方法告知 Servlet 属性 hidden -->
<input type="hidden" name="method" value="update"/>
<table>
  <tr>
    <!-- ID 为只读, 显示为选中的对象的 ID, 故应为动态 ID -->
    <td>ID</td>
    <td><input type="text" name="id" readonly="readonly"
      value="${publisher.publisherId }"/></td>
  </tr>
  <tr>
    <!-- 显示为选中的对象的 name, 故应为动态 name -->
    <td>Name</td>
    <td><input type="text" name="name"
      value="${publisher.publisherName }"/></td>
  </tr>
  <tr>
    <td>Telephone</td>
    <td><input type="text" name="telephone"
      value="${publisher.telephone }"/></td>
  </tr>
  <tr>
    <td>PostCode</td>
    <td><input type="text" name="postCode"
      value="${publisher.postCode }"/></td>
  </tr>
  <tr>
    <td>Website</td>
    <td><input type="text" name="website"
      value="${publisher.website }"/></td>
  </tr>

```

```

        <tr>
            <td>City</td>
            <td><input type="text" name="city"
                value="${publisher.city    }"></td>
        </tr>
        <tr>
            <td>Province</td>
            <td><input type="text" name="province"
                value="${publisher.province    }"></td>
        </tr>
        <tr>
            <!-- 提交和重置按钮 -->
            <td colspan="2">
                <input type="submit" value="submit"/>
                <input type="reset" value="reset"/>
            </td>
        </tr>
    </table>
</form>
</body>

```

最终结果截图

Index.jsp



click下面啦啦啦啦啦！

[查看出版社列表](#)

List.jsp



出版社列表

ID	Name	Telephone	PostCode	Website	City	Province	
1	高等教育出版社	2811111111	10000	www.publisher1.com	中关村	北京	Update Delete
2	电子工业出版社	2822222223	11111	www.publisher2.com	西安	陕西	Update Delete
3	电子科技大学出版社	2833333333	12222	www.publisher3.com	成都	四川	Update Delete
4	长江出版社	123456789	12312	www.changjiang.com	南京	江苏	Update Delete

[Add](#)

New.jsp



添加学生信息

ID

Name

Telephone

PostCode

Website

City

Province



出版社列表

ID	Name	Telephone	PostCode	Website	City	Province	
1	高等教育出版社	2811111111	10000	www.publisher1.com	中关村	北京	Update Delete
2	电子工业出版社	2822222223	11111	www.publisher2.com	西安	陕西	Update Delete
3	电子科技大学出版社	2833333333	12222	www.publisher3.com	成都	四川	Update Delete
4	长江出版社	123456789	12312	www.changjiang.com	南京	江苏	Update Delete
5	信软学院出版社	123456789	12312	www.baidu.com	成都	四川	Update Delete

[Add](#)

Update.jsp



修改学生信息

ID	<input type="text" value="5"/>
Name	<input type="text" value="信软学院出版社"/>
Telephone	<input type="text" value="123456789"/>
PostCode	<input type="text" value="12312"/>
Website	<input type="text" value="www.baidu.com"/>
City	<input type="text" value="成都电子科技大学"/>
Province	<input type="text" value="四川"/>
<input type="button" value="submit"/> <input type="button" value="reset"/>	



出版社列表

ID	Name	Telephone	PostCode	Website	City	Province	
1	高等教育出版社	2811111111	10000	www.publisher1.com	中关村	北京	Update Delete
2	电子工业出版社	2822222223	11111	www.publisher2.com	西安	陕西	Update Delete
3	电子科技大学出版社	2833333333	12222	www.publisher3.com	成都	四川	Update Delete
4	长江出版社	123456789	12312	www.changjiang.com	南京	江苏	Update Delete
5	信软学院出版社	123456789	12312	www.baidu.com	成都电子科技大学	四川	Update Delete

[Add](#)

删除



出版社列表

ID	Name	Telephone	PostCode	Website	City	Province	
1	高等教育出版社	2811111111	10000	www.publisher1.com	中关村	北京	Update Delete
2	电子工业出版社	2822222223	11111	www.publisher2.com	西安	陕西	Update Delete
3	电子科技大学出版社	2833333333	12222	www.publisher3.com	成都	四川	Update Delete
4	长江出版社	123456789	12312	www.changjiang.com	南京	江苏	Update Delete
5	信软学院出版社	123456789	12312	www.baidu.com	成都电子科技大学	四川	Update Delete

[Add](#)

出版社列表

ID	Name	Telephone	PostCode	Website	City	Province	
1	高等教育出版社	2811111111	10000	www.publisher1.com	中关村	北京	Update Delete
2	电子工业出版社	2822222223	11111	www.publisher2.com	西安	陕西	Update Delete
3	电子科技大学出版社	2833333333	12222	www.publisher3.com	成都	四川	Update Delete
4	长江出版社	123456789	12312	www.changjiang.com	南京	江苏	Update Delete

[Add](#)

九、总结及心得体会

此次实验是自我的一次大胆尝试，不单纯地利用 JDBC 和 JSP 的简单知识，而是采用 JavaEE 中一种常见的设计模式 MVC 来完成本次实验。程序主要是负责和用户进行交互，获取页面传递过来的数据进行相关逻辑处理，以及将数据库中存储的信息展示到页面，即 MVC 中的 Controller 层,而对于需要直接和用户进行交互的展示 jsp 则是所谓的 view 层，将后台传递来的数据以较为友好的方式传递给用户以及将用户键入的数据传递给后台，当然在数据库中存储的数据和我们程序中的数据需要存在一定的映射关系，两个对象类型不可能完全一致，所以程序需要根据数据库中存储的对象进行相关的抽象操作，转化成能由程序直接处理的类，也就是所谓的 Model 层。而对于本学科所侧重的数据库，无论类型（ORACLE/MYSQL/SQ SERVER），都可以采用相同的设计模式来对数据库进行操作，只是需要导入不同的数据库对应的 JDBC 的组件包。

十、对本实验过程及方法、手段的改进建议

希望能通过更多的应用场景的实际开发中的例子来阐释数据库在我们的开发中所扮演的重要角色，以及各种类型的应用对数据库的操作的大致实现过程。

报告评分：

指导教师签字：