



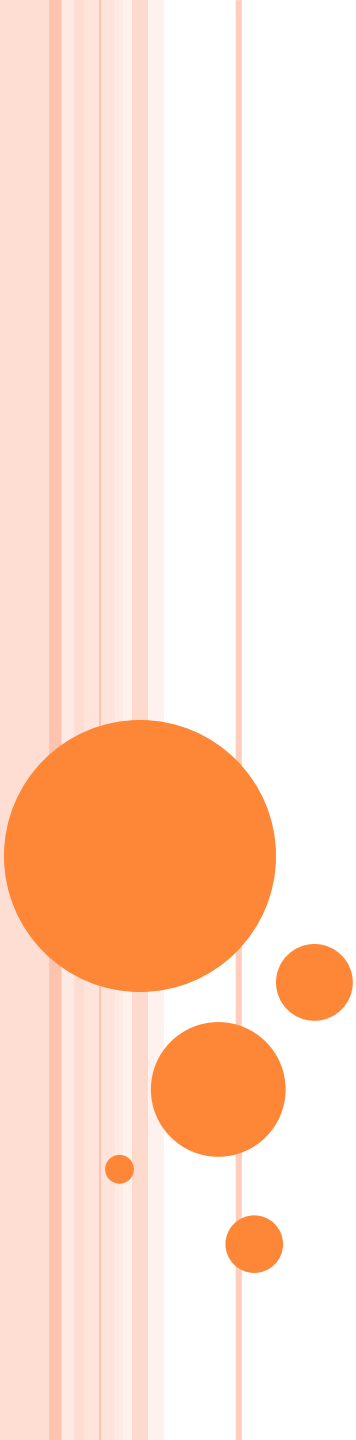
# 数据库系统

Database System

主讲：张仲楠 教授

Email: [zhongnan\\_zhang@xmu.edu.cn](mailto:zhongnan_zhang@xmu.edu.cn)

Office: 海韵A416



# 实验五

## 数据库设计

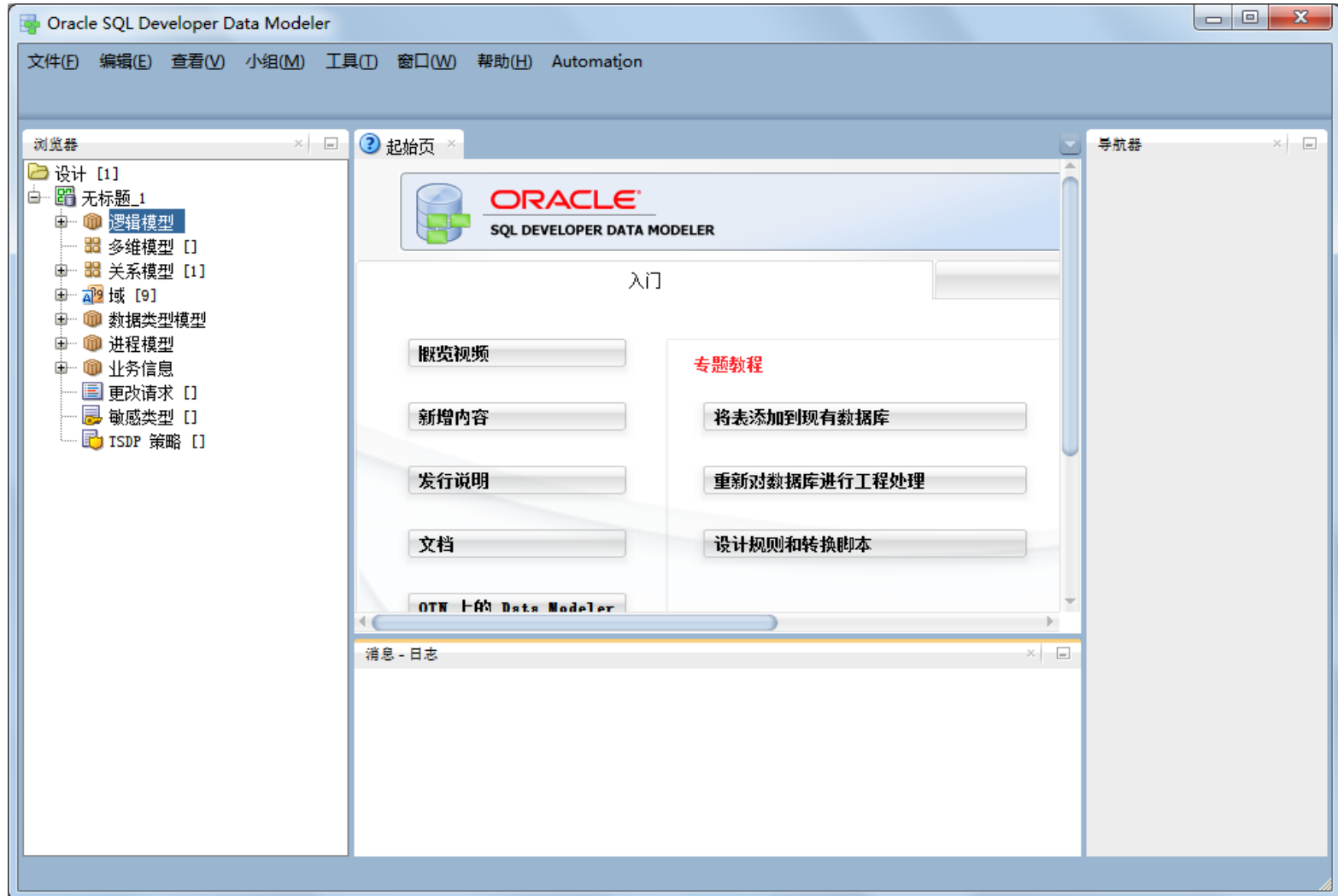
# DATA MODELER



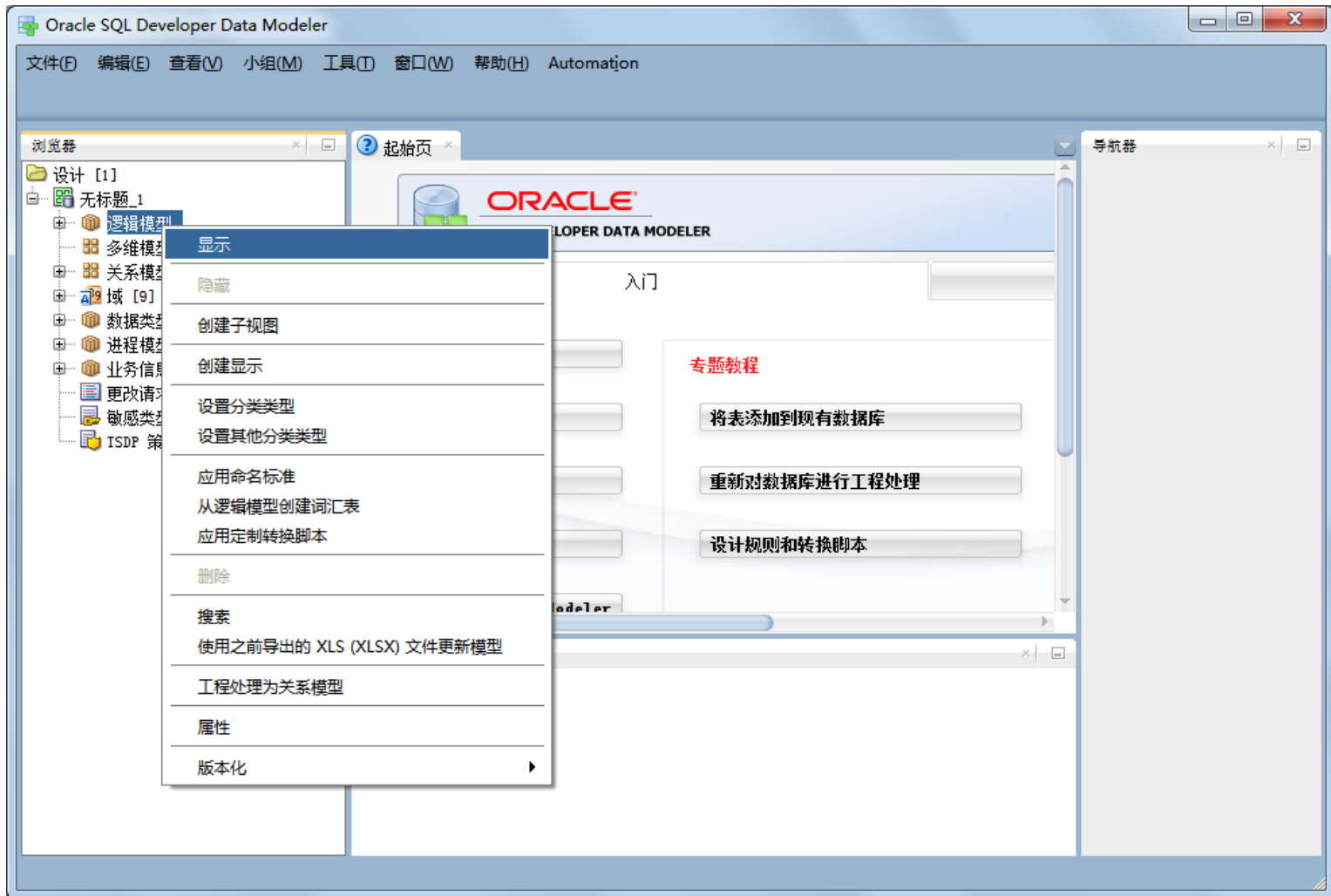
下载地址: <ftp://121.192.180.66/教学课件/张仲楠/数据库>



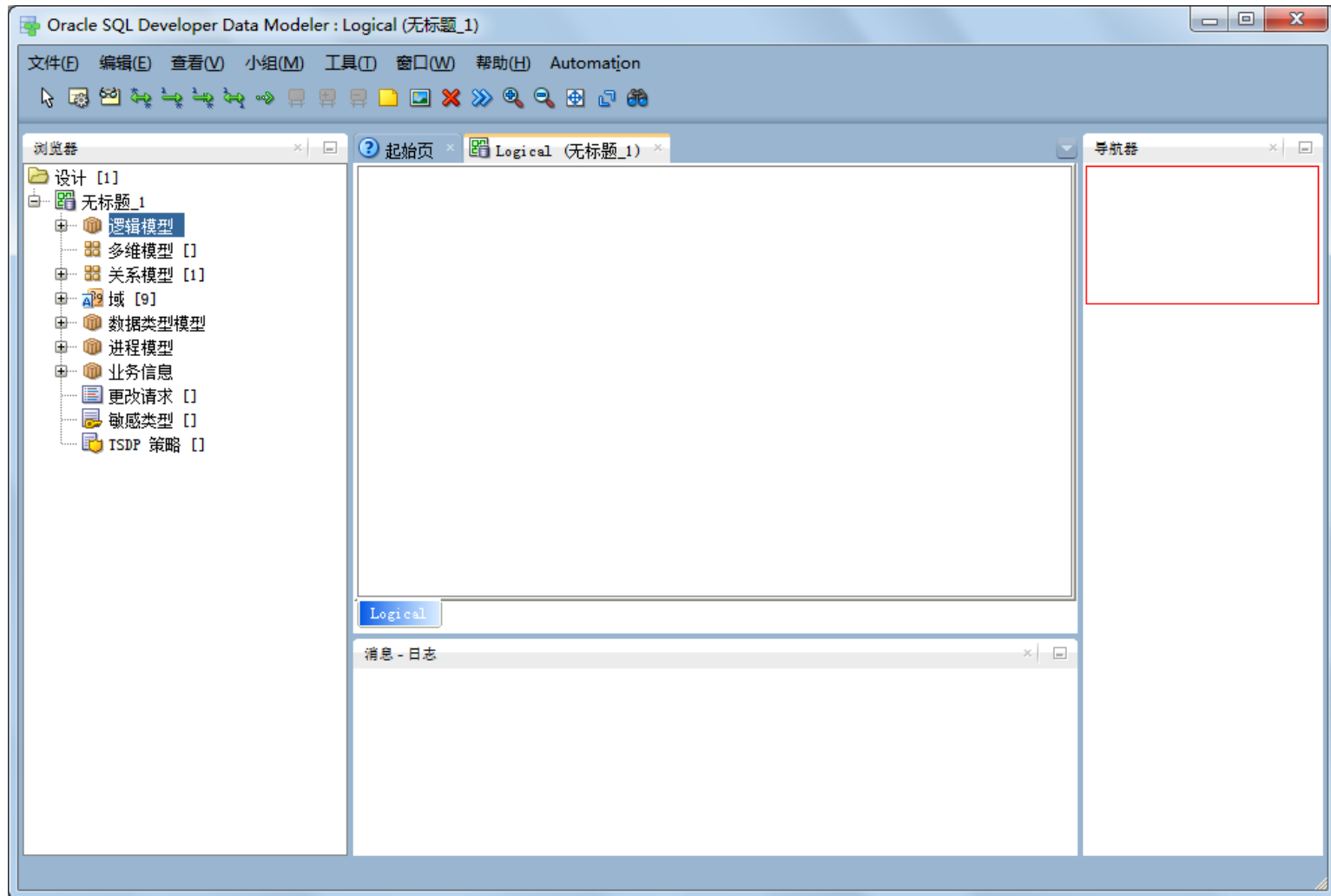
# DATA MODELER



# DATA MODELER



# DATA MODELER

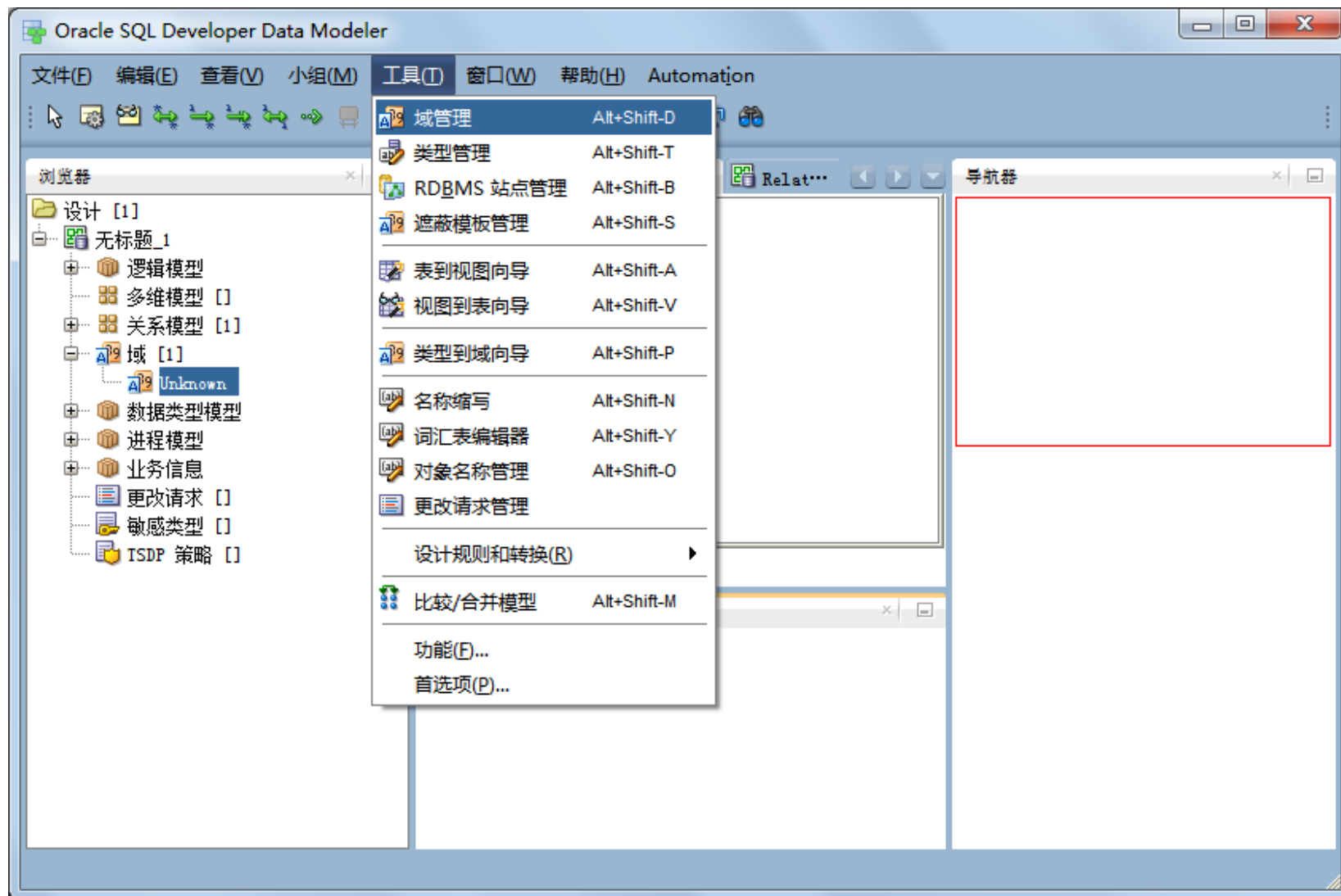


# 简介

- **注意：**在这个软件里，“**逻辑模型**”就是我们在课堂上说的“**概念模型**”，即**E-R模型**.
- 为图书馆设计一个数据库
- 定义如下3个实体：
  - **Books**（描述图书馆里的每本书）
  - **Patrons**（描述每一个用户）
  - **Transactions**（描述每一个事务）



# 增加域





# 增加域

域管理

选择域文件

选择

可用域

Unknown

添加 删除 修改

域属性

名称: Unknown

逻辑类型: ☐ 首选 unknown

大小:

单位:

精度:

小数位数:

域文件: defaultdomains

敏感类型:

同义词:

注释:

度量单位:

默认值:

检查约束条件

范围

值列表

敏感数据说明

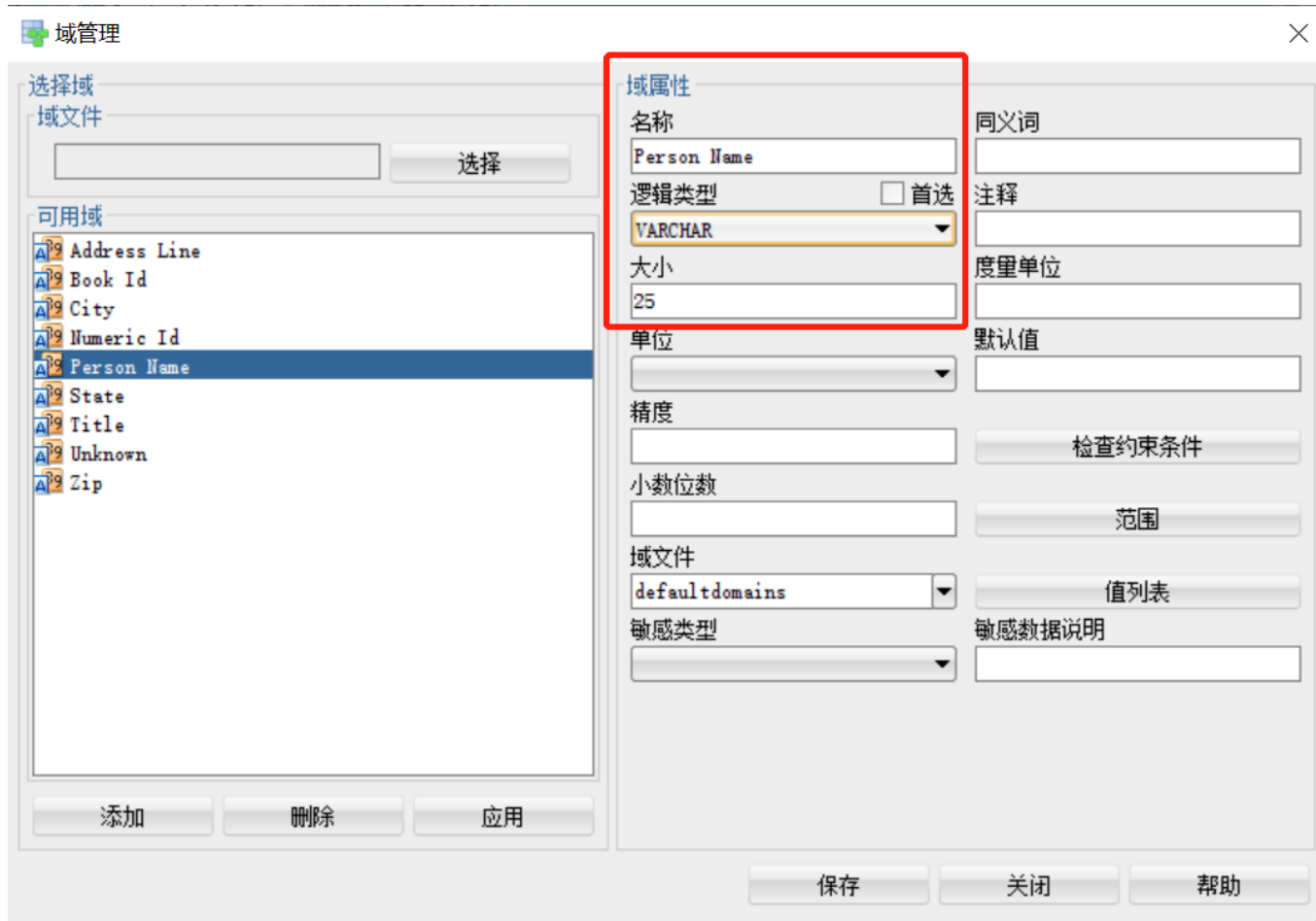
保存 关闭 帮助

## 增加域

Name	Logical Type	Other Information
Person Name	VARCHAR	Size: 25
Address Line	VARCHAR	Size: 40
City	VARCHAR	Size: 25
State	VARCHAR	Size: 2
Zip	VARCHAR	Size: 10
Book Id	VARCHAR	Size: 20
Numeric Id	NUMERIC	Precision: 7, Scale: 0
Title	VARCHAR	Size: 50

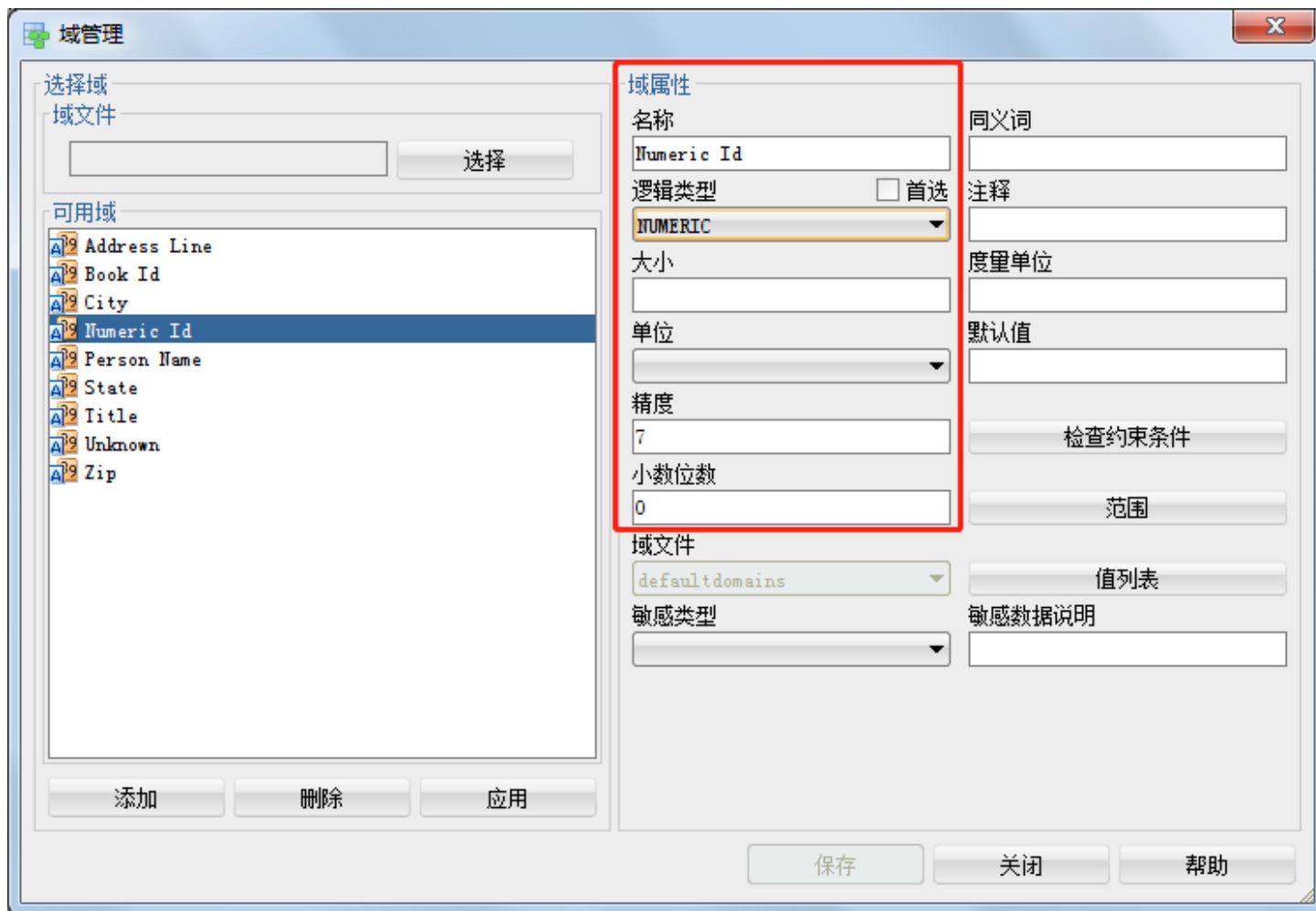


# 增加域



\\datamodeler\\datamodeler\\types\\defaultdomains.xml

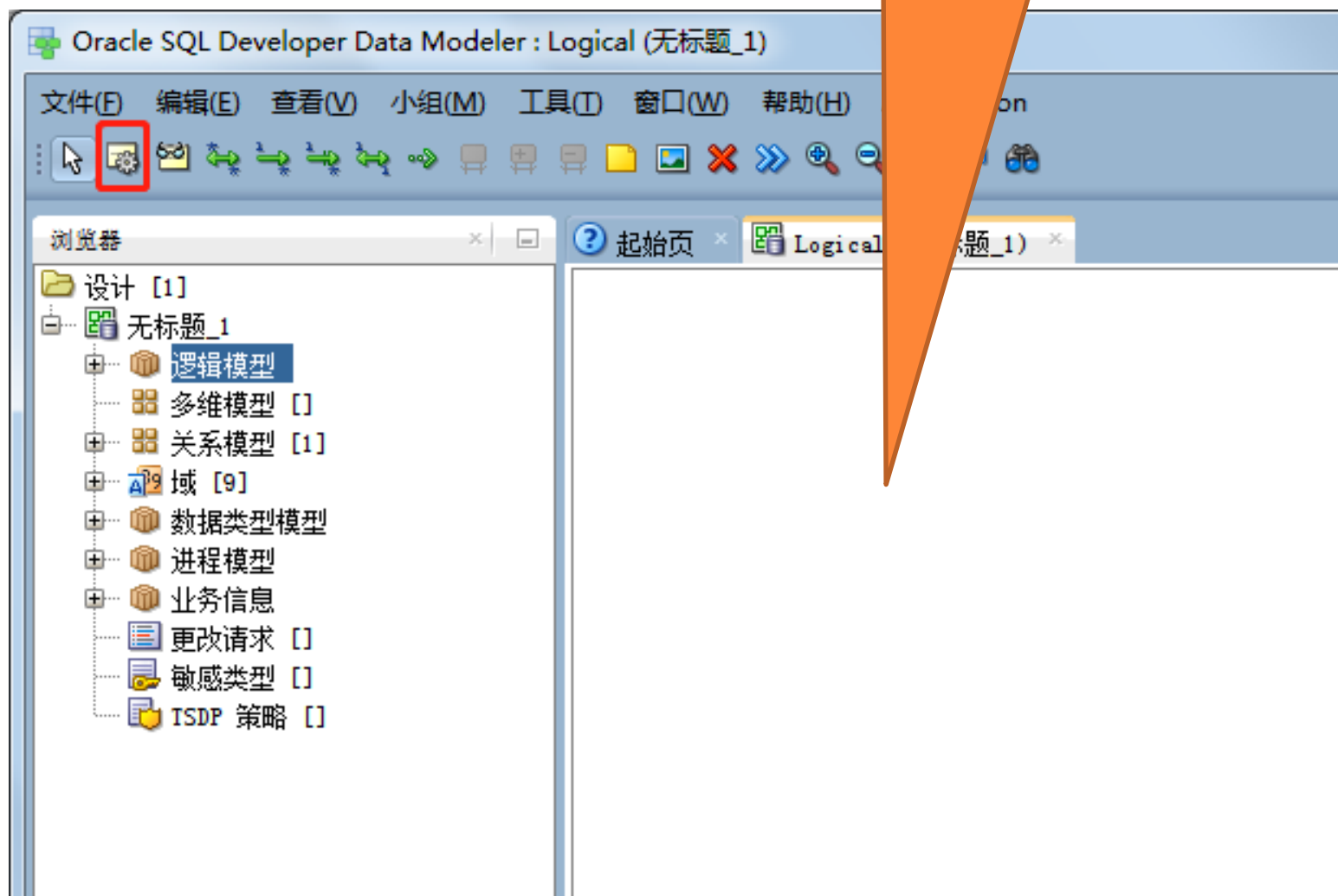
# 增加域



\datamodeler\datamodeler\types\defaultdomains.xml

## 创建BOOKS 实体

点完”创建实体”按钮,在工作区会出现一个”十字”,点鼠标左键拉出一个矩形框,松鼠标



# 创建BOOKS 实体

实体属性 - Entity\_2

一般信息

属性

唯一标识符

关系

子类型

卷属性

工程处理目标

注释

RDBMS 中的注释

重叠属性

备注

影响分析

度量

更改请求

责任方

文档

动态属性

用户定义的属性

分类类型

概要

一般信息

名称 Books

短名称

同义词

要显示的同义词

首选缩写

长名称 Entity\_2

基于结构化类型

超类型

源

允许类型替换: ☒

创建代理关键字: ☐

已废弃 ☐

选择

确定 应用 命名规则 取消 帮助

# 创建BOOKS 实体

实体属性 - Entity\_2

一般信息  
**属性**  
唯一标识符  
关系  
子类型  
卷属性  
工程处理目标  
注释  
RDBMS 中的注释  
重叠属性  
备注  
影响分析  
度量  
更改请求  
责任方  
文档  
动态属性  
用户定义的属性  
分类类型  
概要

属性

详细资料 概览 UDP

属性:

名称

数据类型

名称

数据类型

属性特性

名称

数据类型

源类型

域 逻辑 相异  
结构化 集合

Unknown 首选

主 UID 关系 UID 必需 已废弃

注释 RDBMS 中的注释 备注

确定 应用 命名规则 取消 帮助

## 创建BOOKS 实体

主码

Name	Datatype	Other Information and Notes
book_id	<u>Domain: Book Id</u>	Primary UID (unique identifier) (The Dewey code or other book identifier.)
title	<u>Domain: Title</u>	M ( <u>mandatory</u> , that is, must not be null).
author_last_name	<u>Domain: Person Name</u>	M ( <u>mandatory</u> , that is, must not be null).
author_first_name	<u>Domain: Person Name</u>	(Author's first name; not mandatory, but enter it if the author has a first name.)
rating	<u>Logical type: NUMERIC</u> (Precision=2, Scale= 0)	(Librarian's personal rating of the book, from 1 (poor) to 10 (great).)

mandatory:必需，也就是不能为null



# 创建BOOKS 实体

属性特性

名称

数据类型 ☒ 域 ☐ 逻辑 ☐ 相异  
☐ 结构化 ☐ 集合

源类型  首选 ☐

☒ 主 UID ☐ 关系 UID ☒ 必需 ☐ 已废弃

属性特性

名称

数据类型 ☒ 域 ☐ 逻辑 ☐ 相异  
☐ 结构化 ☐ 集合

源类型  首选 ☐

☐ 主 UID ☐ 关系 UID ☒ 必需 ☐ 已废弃

属性特性

名称

数据类型 ☒ 域 ☐ 逻辑 ☐ 相异  
☐ 结构化 ☐ 集合

源类型  首选 ☐

☐ 主 UID ☐ 关系 UID ☒ 必需 ☐ 已废弃

属性特性

名称

数据类型 ☒ 域 ☐ 逻辑 ☐ 相异  
☐ 结构化 ☐ 集合

源类型  首选 ☐

☐ 主 UID ☐ 关系 UID ☐ 必需 ☐ 已废弃

## 创建BOOKS 实体

属性特性

名称

数据类型 ☐ 域 ☒ 逻辑 ☐ 相异  
☐ 结构化 ☐ 集合

源类型  首选 ☐

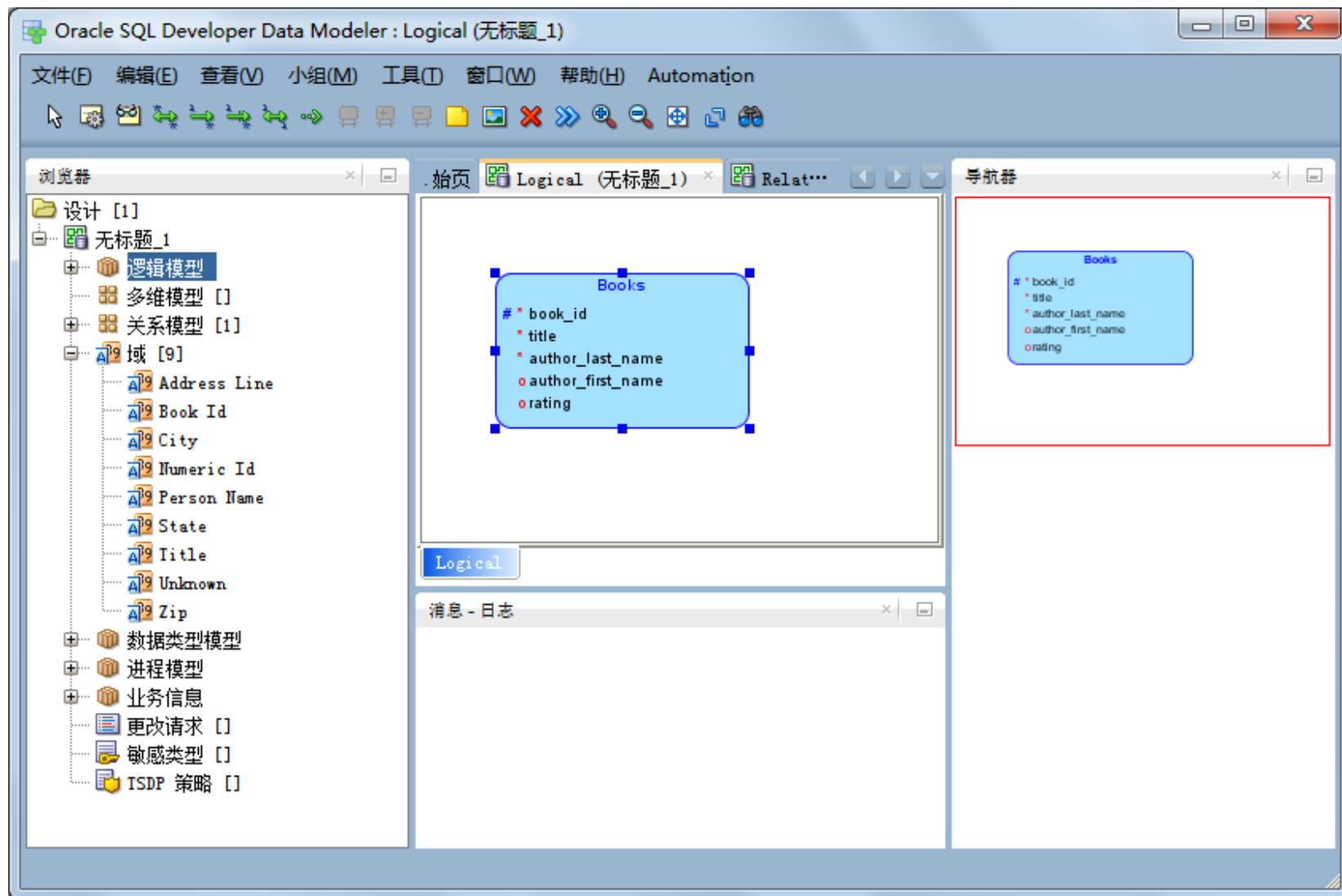
精度

小数位数

☐ 主 UID ☐ 关系 UID ☐ 必需 ☐ 已废弃

	名称	数据类型
1	book_id	Book Id
2	title	Title
3	author_last_name	Person Name
4	author_first_...	Person Name
5	rating	NUMERIC (2)

# 创建BOOKS 实体



# 创建PATRONS 实体

主码

Attribute Name	Type	Other Information and Notes
patron_id	Domain: Numeric Id	<u>Primary UID (unique identifier)</u> . (Unique patron ID number, also called the library card number.)
last_name	Domain: Person Name	M ( <u>mandatory</u> , that is, must not be null). 25 characters maximum.
first_name	Domain: Person Name	(Patron's first name.)
street_address	Domain: Address Line	(Patron's street address.)
city	Domain: City	(City or town where the patron lives.)
state	Domain: State	(2-letter code for the state where the patron lives.)
zip	Domain: Zip	(Postal code where the patron lives.)
location	Structured type: SDO_GEOMETRY	Oracle Spatial and Graph geometry object representing the patron's geocoded address.

mandatory:必需，也就是不能为null

# 创建PATRONS 实体

属性特性

名称

数据类型 ☒ 域 ☐ 逻辑 ☐ 相异  
☐ 结构化 ☐ 集合

源类型  首选 ☐

☒ 主 UID ☐ 关系 UID ☒ 必需 ☐ 已废弃

属性特性

名称

数据类型 ☒ 域 ☐ 逻辑 ☐ 相异  
☐ 结构化 ☐ 集合

源类型  首选 ☐

☐ 主 UID ☐ 关系 UID ☒ 必需 ☐ 已废弃

属性特性

名称

数据类型 ☒ 域 ☐ 逻辑 ☐ 相异  
☐ 结构化 ☐ 集合

源类型  首选 ☐

☐ 主 UID ☐ 关系 UID ☐ 必需 ☐ 已废弃

属性特性

名称

数据类型 ☒ 域 ☐ 逻辑 ☐ 相异  
☐ 结构化 ☐ 集合

源类型  首选 ☐

☐ 主 UID ☐ 关系 UID ☐ 必需 ☐ 已废弃

# 创建PATRONS 实体

属性特性

名称

数据类型 ☒ 域 ☐ 逻辑 ☐ 相异  
☐ 结构化 ☐ 集合

源类型  首选 ☐

☐ 主 UID ☐ 关系 UID ☐ 必需 ☐ 已废弃

属性特性

名称

数据类型 ☒ 域 ☐ 逻辑 ☐ 相异  
☐ 结构化 ☐ 集合

源类型  首选 ☐

☐ 主 UID ☒ 关系 UID ☐ 必需 ☐ 已废弃

属性特性

名称

数据类型 ☒ 域 ☐ 逻辑 ☐ 相异  
☐ 结构化 ☐ 集合

源类型  首选 ☐

☐ 主 UID ☐ 关系 UID ☐ 必需 ☐ 已废弃

属性特性

名称

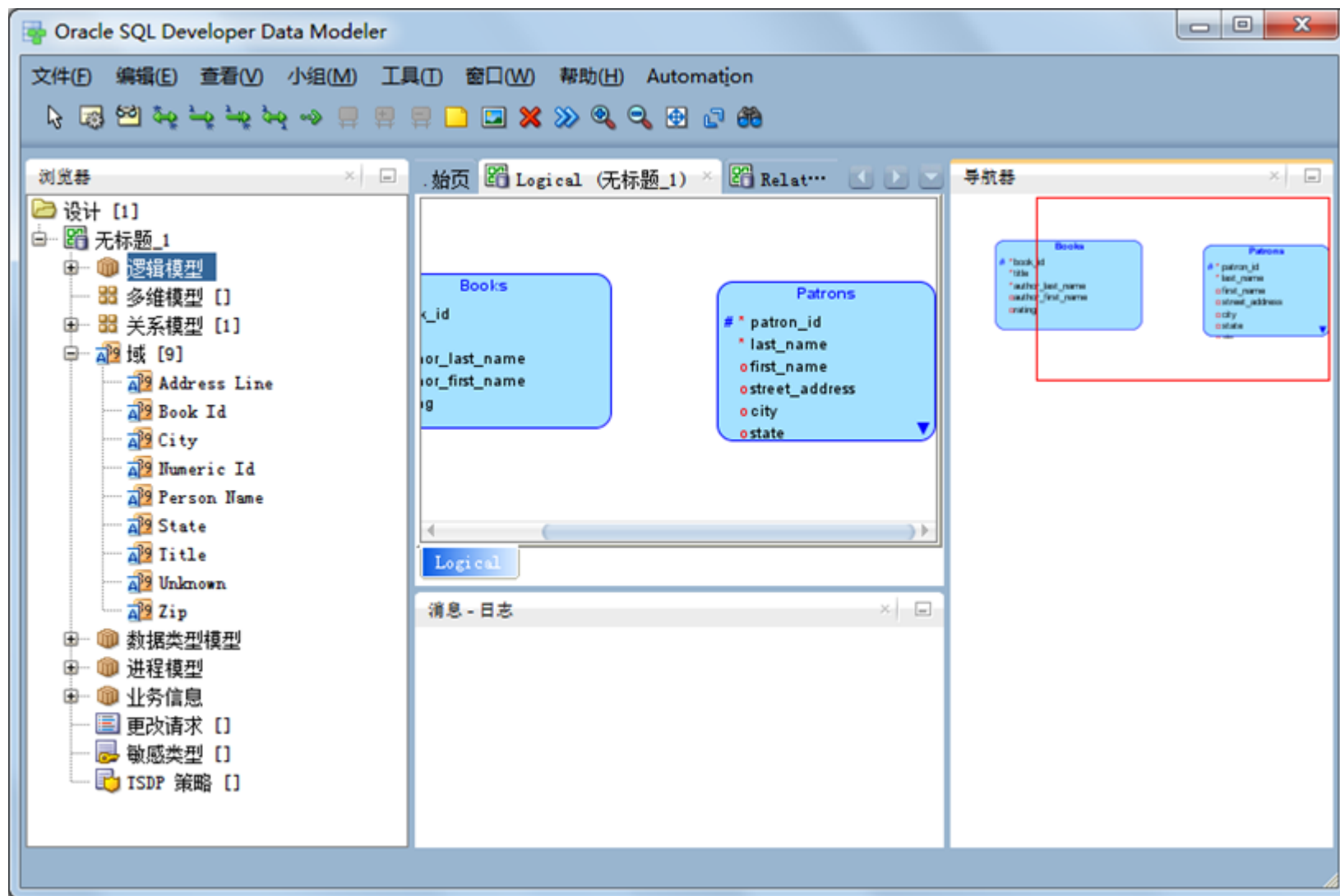
数据类型 ☐ 域 ☐ 逻辑 ☐ 相异  
☒ 结构化 ☐ 集合

源类型  首选 ☐

☐ 引用

☐ 主 UID ☐ 关系 UID ☐ 必需 ☐ 已废弃

# 创建PATRONS 实体



## 创建TRANSACTIONS 实体

Attribute Name	Type	Other Information and Notes
transaction_id	Domain: Numeric Id	<u>Primary UID (unique identifier)</u> . (Unique transaction ID number)
transaction_date	Logical type: Datetime	M ( <u>mandatory</u> , that is, must not be null). Date and time of the transaction.
transaction_type	Domain: Numeric Id	M ( <u>mandatory</u> , that is, must not be null). (Numeric code indicating the type of transaction, such as 1 for checking out a book.)

mandatory:必需，也就是不能为null





# 创建TRANSACTIONS 实体

属性特性

名称

数据类型 ☒ 域 ☐ 逻辑 ☐ 相异  
☐ 结构化 ☐ 集合

源类型  首选 ☐

☒ 主 UID ☐ 关系 UID ☒ 必需 ☐ 已废弃

属性特性

名称

数据类型 ☐ 域 ☒ 逻辑 ☐ 相异  
☐ 结构化 ☐ 集合

源类型  首选 ☐

☐ 主 UID ☐ 关系 UID ☒ 必需 ☐ 已废弃

属性特性

名称

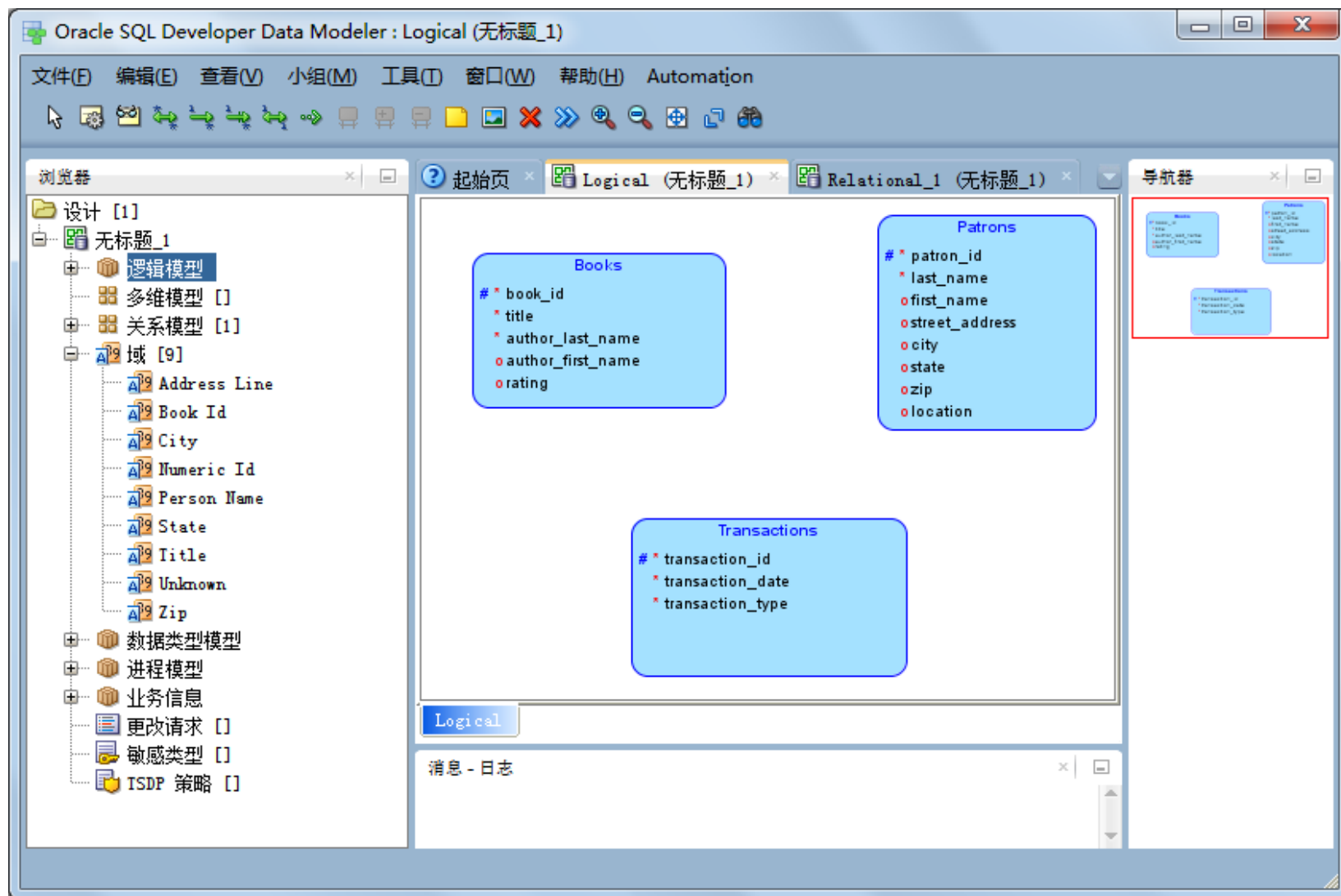
数据类型 ☒ 域 ☐ 逻辑 ☐ 相异  
☐ 结构化 ☐ 集合

源类型  首选 ☐

☐ 主 UID ☐ 关系 UID ☒ 必需 ☐ 已废弃



# 创建TRANSACTIONS 实体



## 实体间的联系约束

- 对实体之间联系描述的细化
- 基数约束 (Cardinality constraint)
  - 说明实体型中的任何一个实体可以在联系中出现的**最多**次数(**上限**)
- 参与约束 (Participation constraint )
  - 说明实体型中的任何一个实体可以在联系中出现的**最少**次数(**下限**)
  - 类型: 强制 (mandatory) — 至少参与一次  
可选 (optional) — 可以参与, 也可以不参与



## 创建实体间的联系

The following relationships exist between the entities:

- Books and Transactions: **one-to-many**. Each book can be involved in multiple transactions. Each book can have **zero** transaction; a book that is checked out should have a **return due date**.
- Books: 可选
- Transactions: 强制



联系的属性

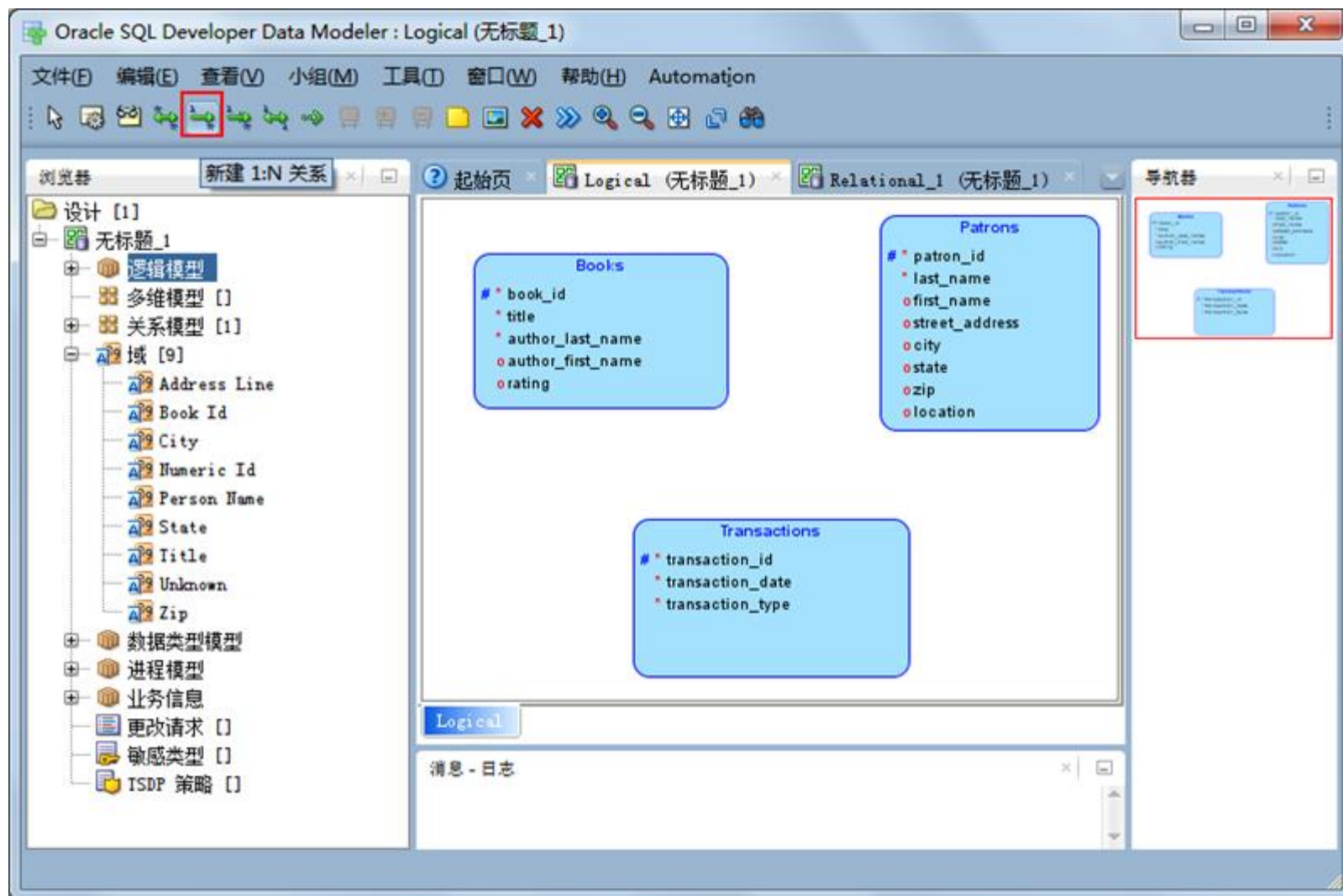
## 创建实体间的联系

The following relationships exist between the entities:

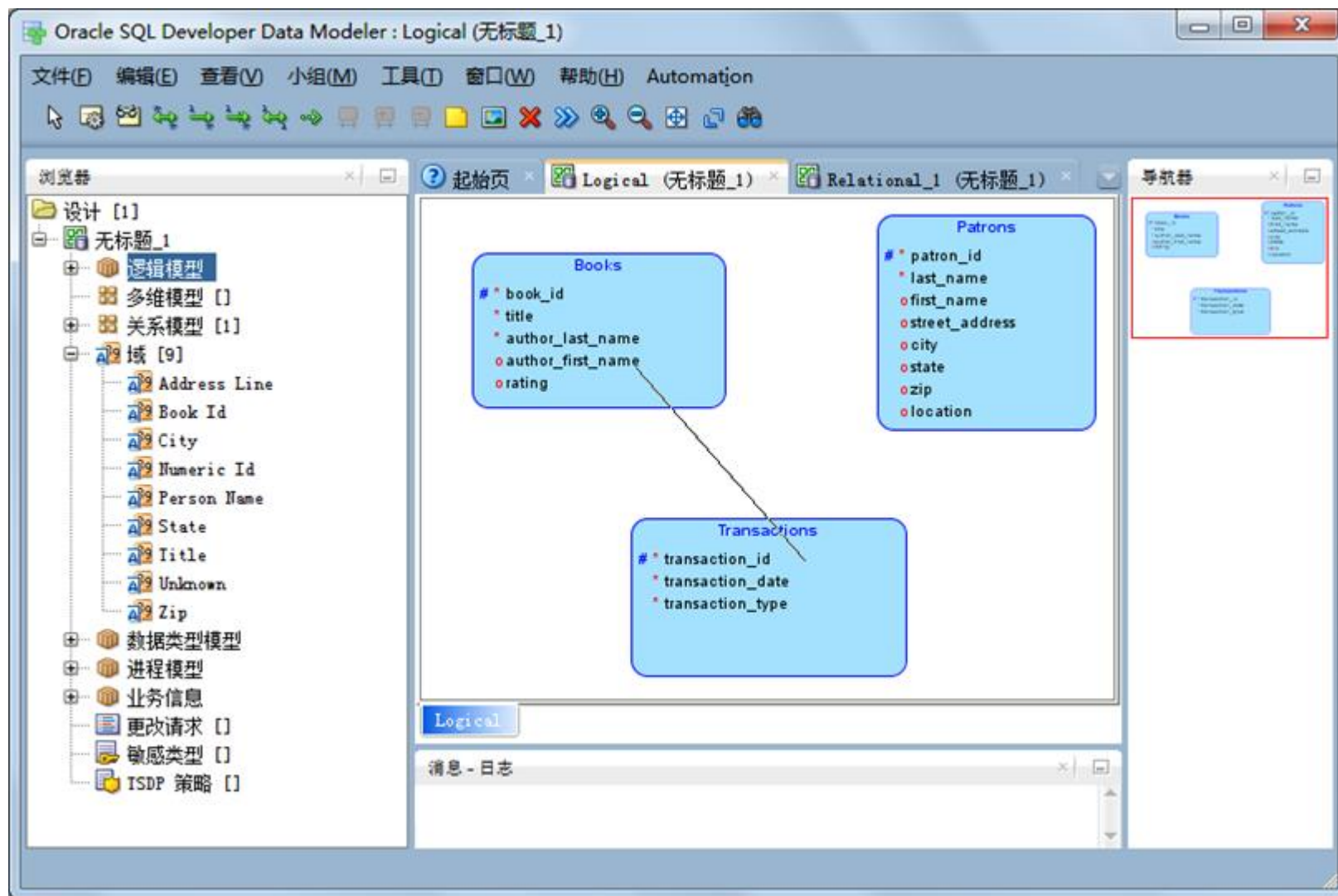
- Patrons and Transactions: **one-to-many**. Each patron can be involved in multiple transactions. Each patron can have **zero**; each patron can also return checked out books at anytime.
- Patrons: 参与约束--可选
- Transactions: 参与约束--强制



# 创建实体间的联系



# 创建实体间的联系



# 创建实体间的联系

关系属性 - Involve

一般信息

名称: Involve

使用代理关键字: ☐

源基数

源: Books

源关键字: Books. Books PK

源上的名称:

源实体同义词: Books

源到目标基数: 1 < \*

源 (可选): ☒

可传输: ☒

属性角色: 无

标识: ☐

删除规则: NO ACTION

目标基数

目标: Transactions

目标关键字: Transactions. Tra...

目标上的名称:

目标实体同义词: Transactions

目标到源基数: 1

目标 (可选): ☐

可传输: ☒

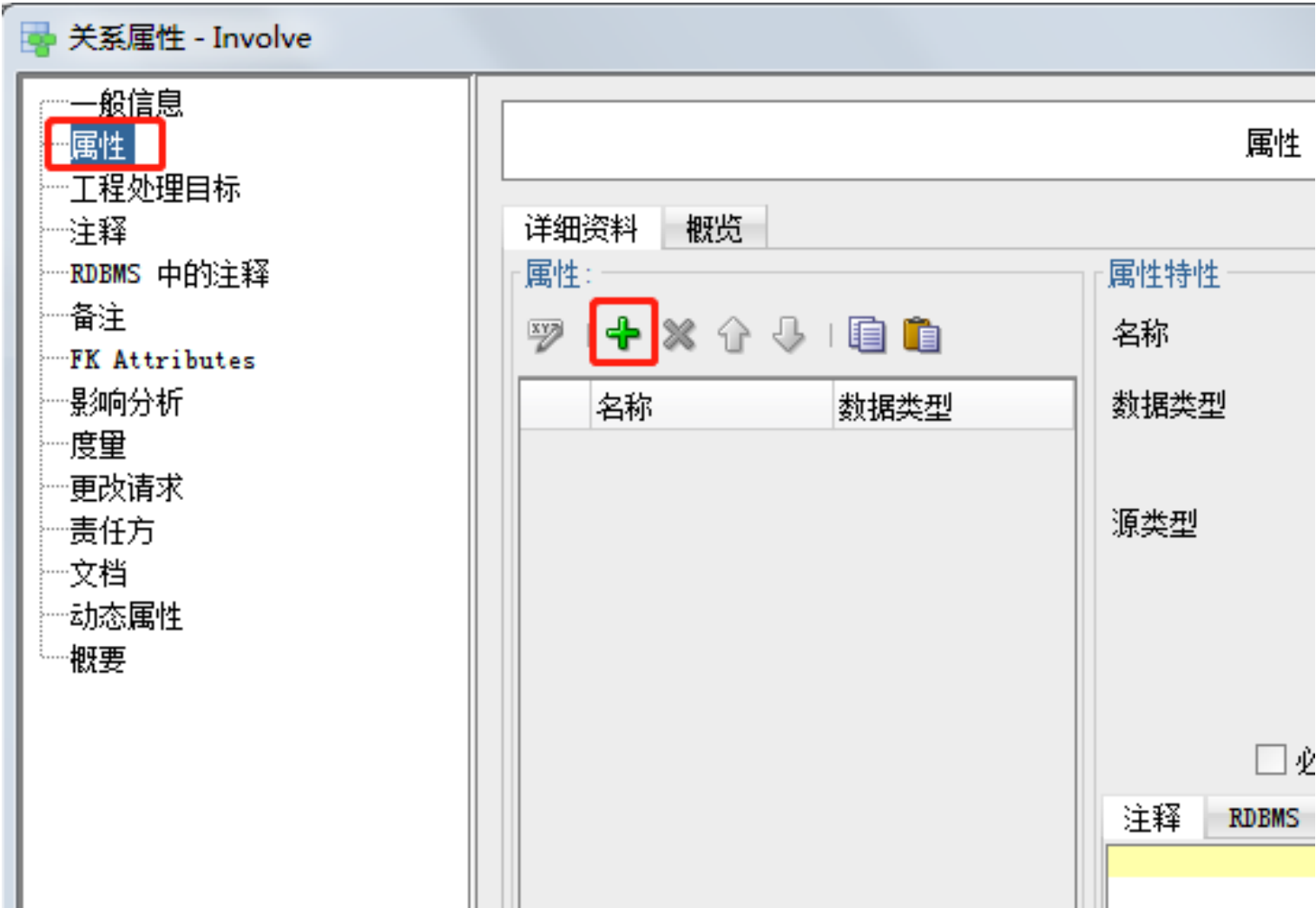
在弧线中: ☐

“打勾”表示参与“可选”

确定 应用 取消 帮助



# 为联系增加属性



## 为联系增加属性

详细资料 概览

属性:

名称 数据类型

1	Due_Date	Date
---	----------	------

属性特性

名称 Due\_Date

数据类型

☐ 域 ☒ 逻辑 ☐ 相异

☐ 结构化 ☐ 集合

源类型 Date 首选 ☐

☐ 必需

注释 RDBMS 中的注释 备注

确定 应用 取消 帮助

# 创建实体间的联系

关系属性 - Has

一般信息

名称: Has

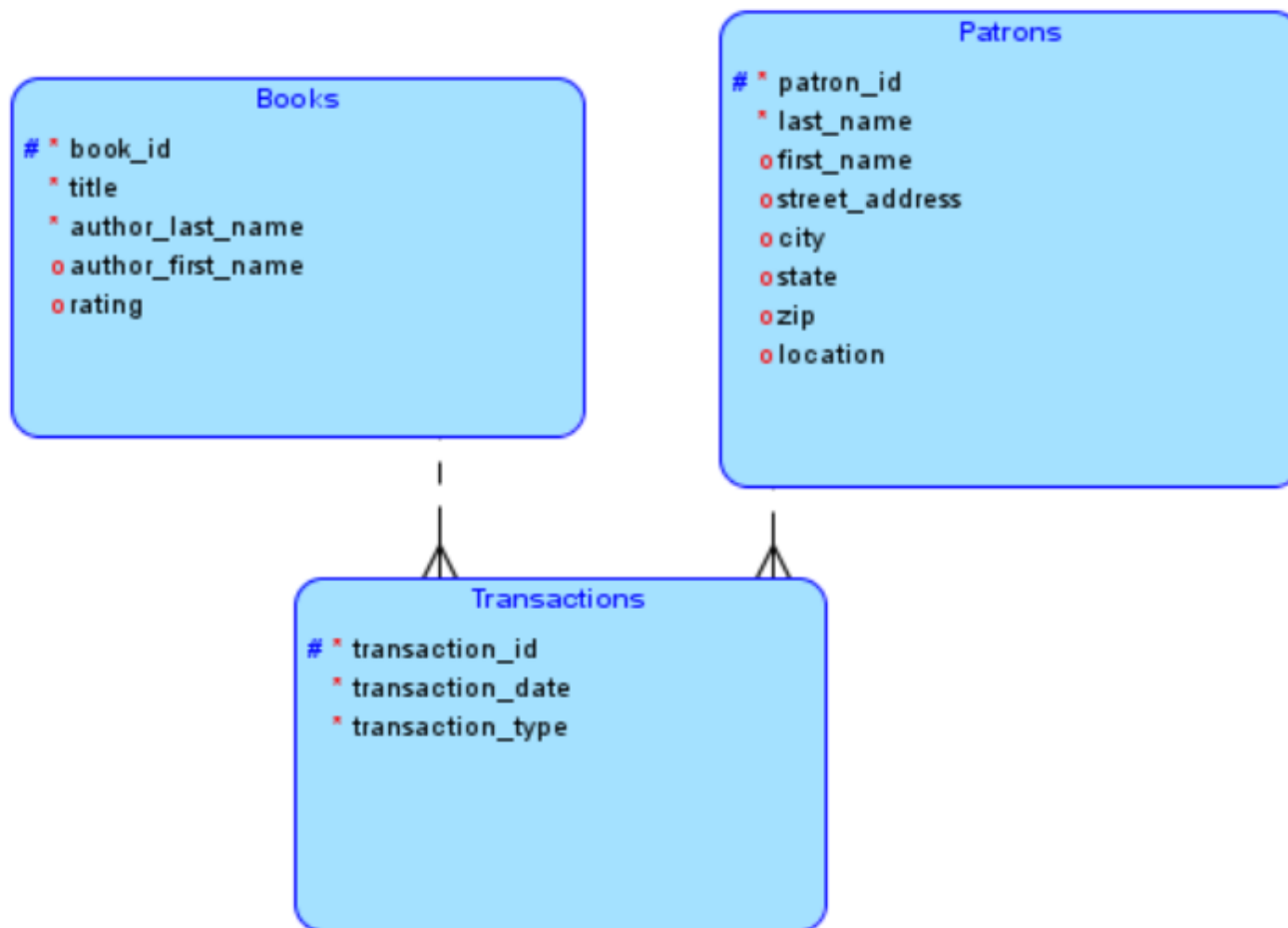
使用代理关键字: ☐

源基数: 源: Patrons 源关键字: Patrons.Patrons PK 源上的名称: 源实体同义词: Patrons 源到目标基数: 1-<\* 源 (可选): ☒ 可传输: ☒ 支配性角色: 无 标识: ☐ 删除规则: NO ACTION

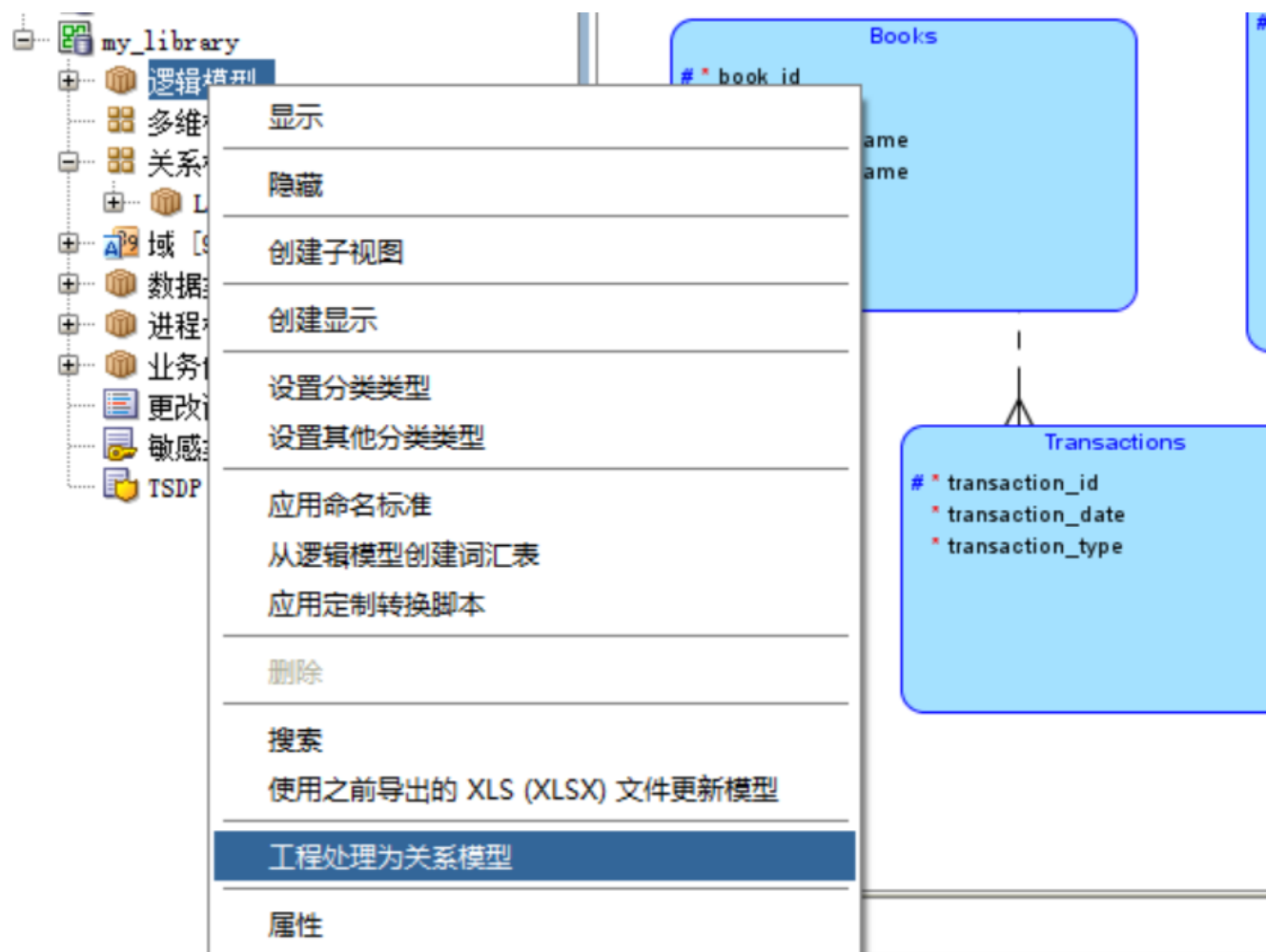
目标基数: 目标: Transactions 目标关键字: Transactions.Tra... 目标上的名称: 目标实体同义词: Transactions 目标到源基数: 1 目标 (可选): ☐ 可传输: ☒ 在弧线中: ☐

确定 应用 取消 帮助

## 完整的E-R图

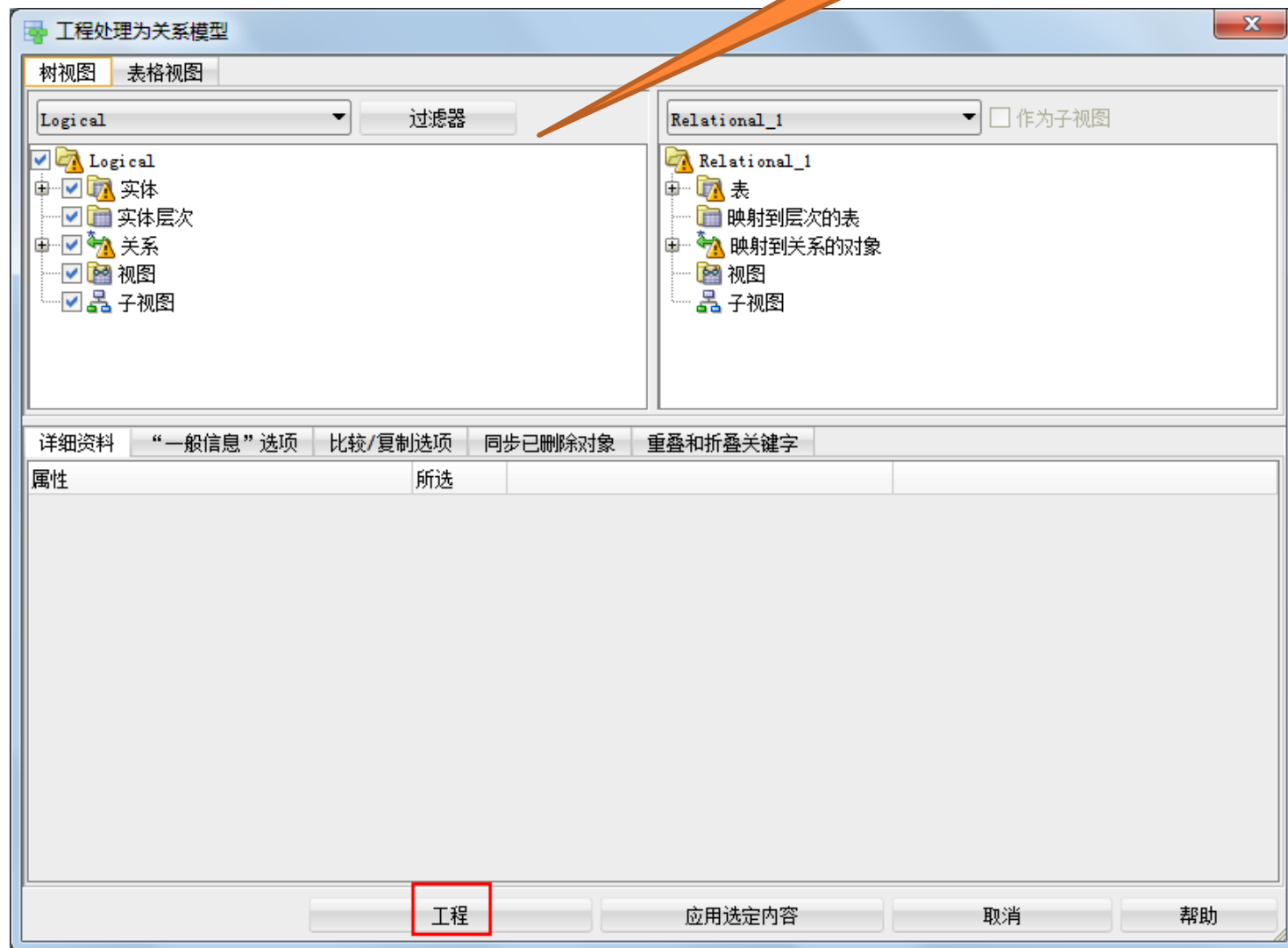


# 生成关系模型



# 生成关系模型

不用改动



# 生成关系模型

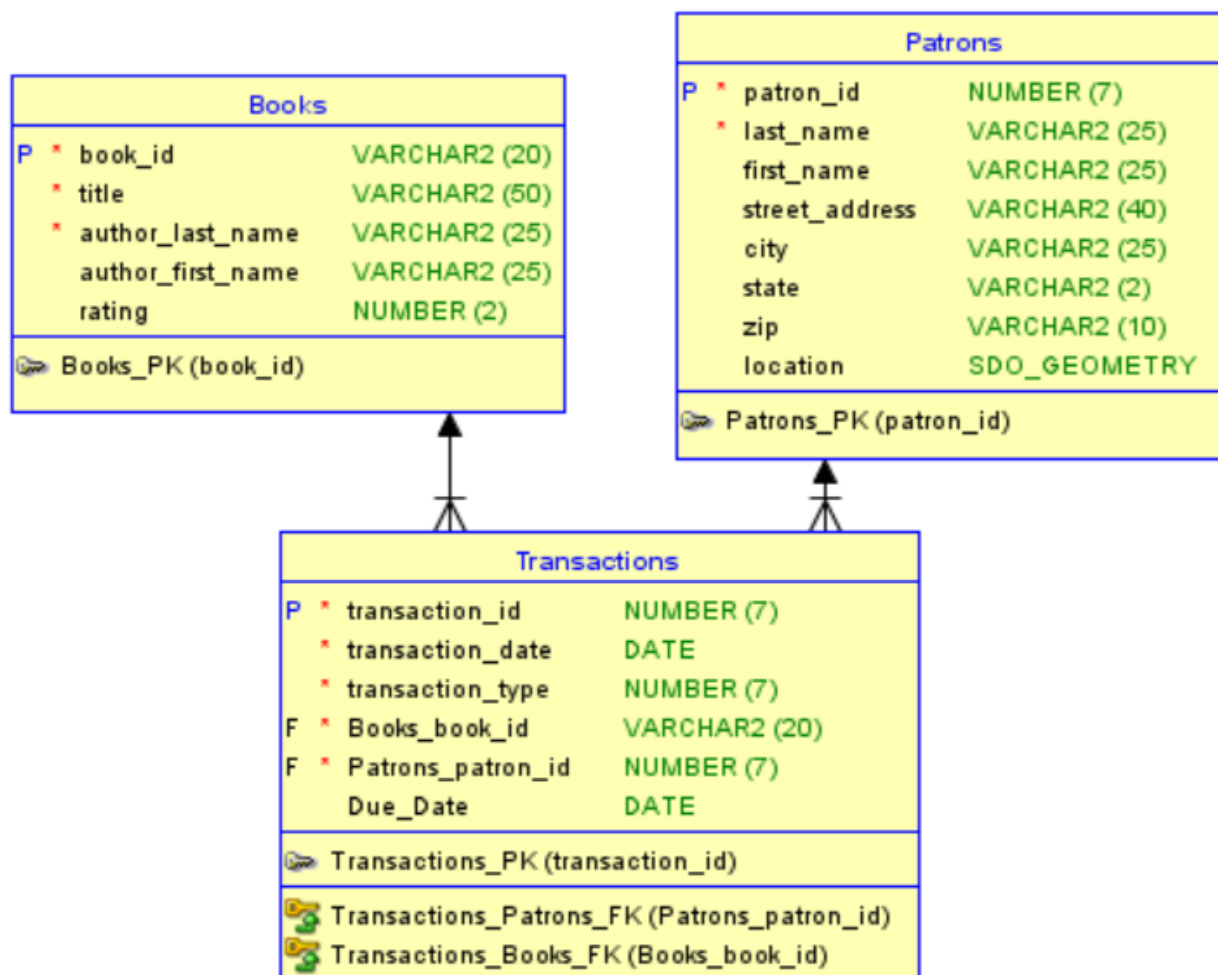
自己命名

模型属性 - Library

一般信息	
名称	Library
可见	<input checked="" type="checkbox"/>
RDMS 类型:	Oracle Database 12c
RDMS 站点:	Oracle Database 12c

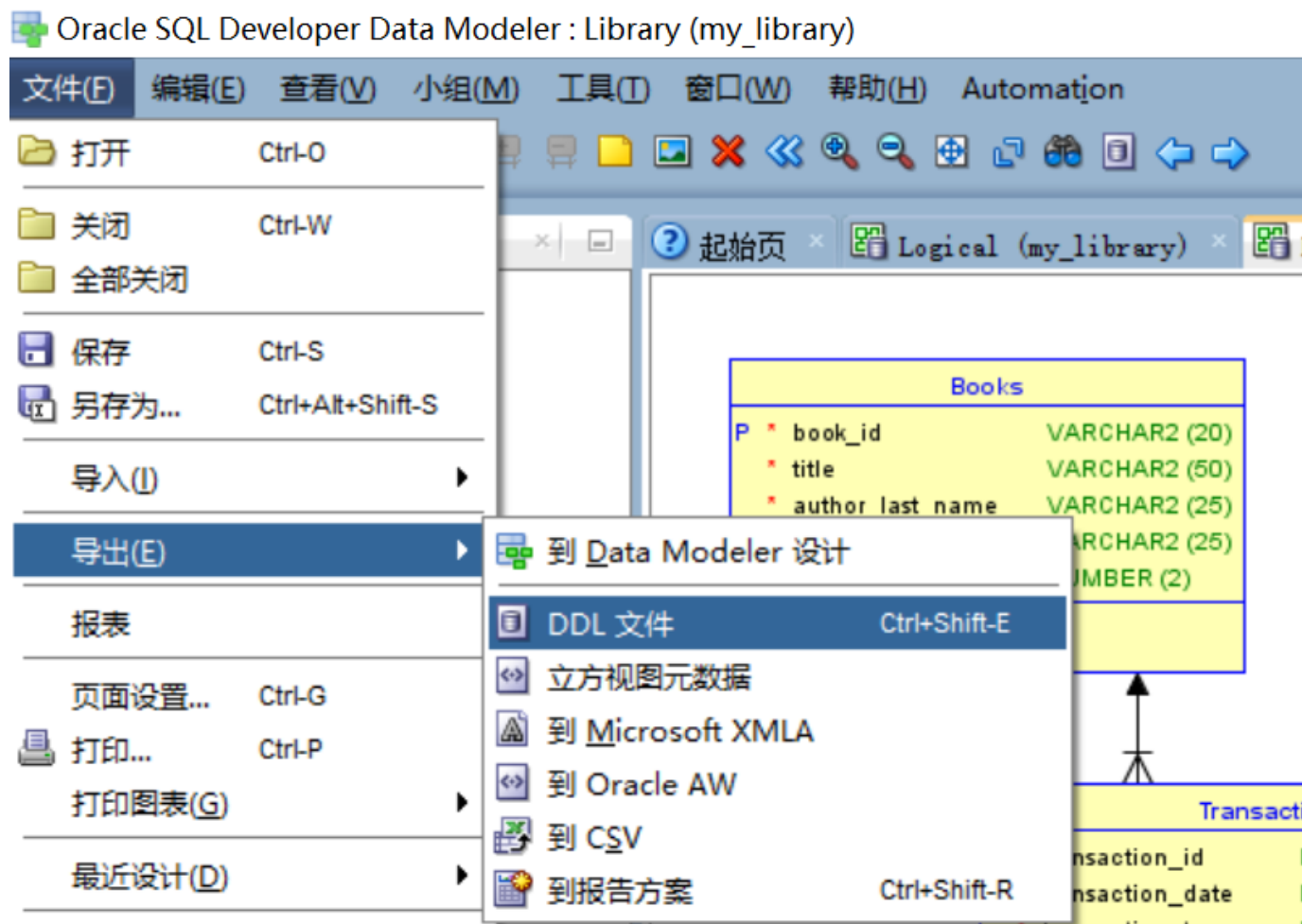
确定 应用 取

# 完整的数据库关系模式





# 生成DDL



# 生成DDL

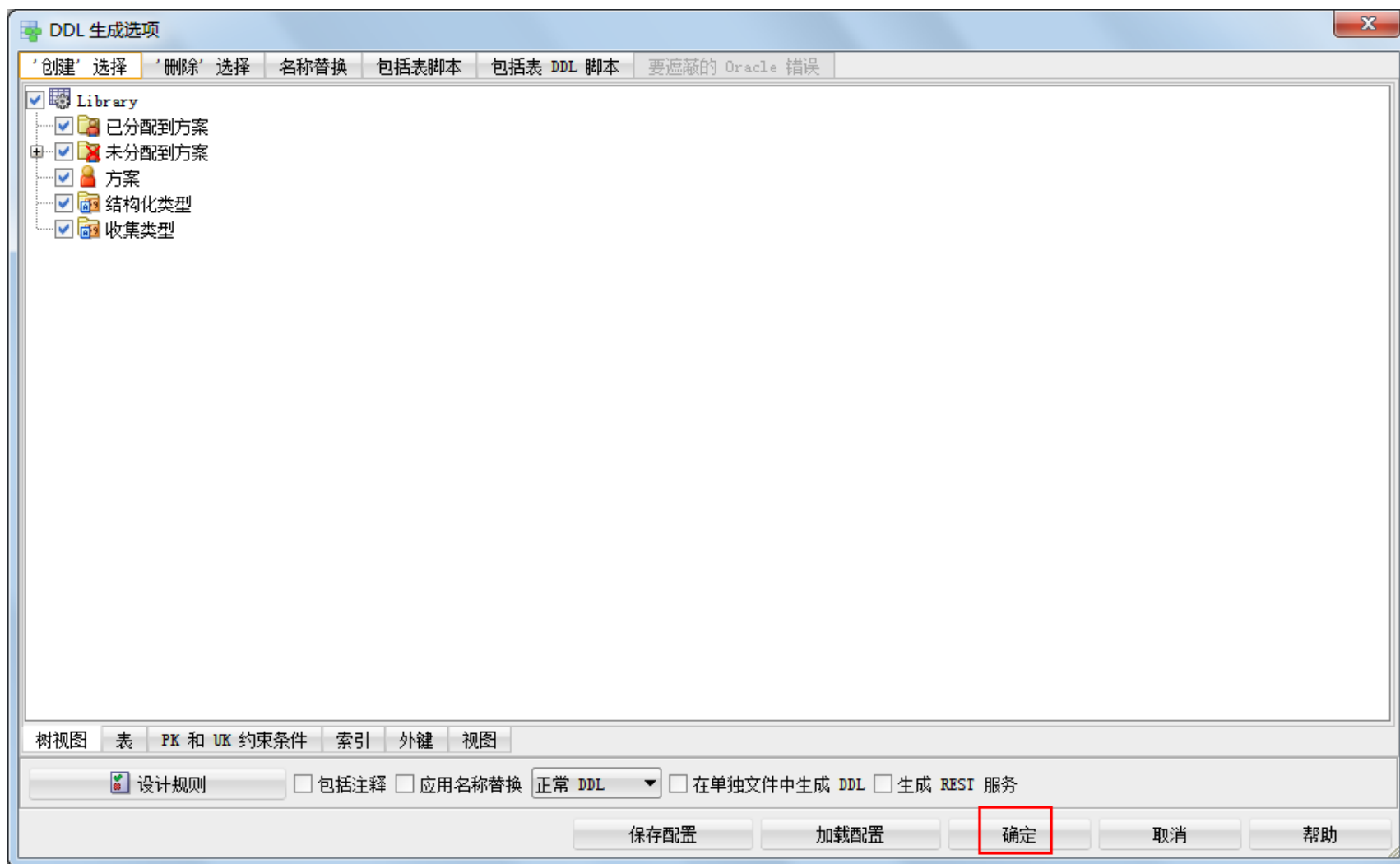
DDL 文件编辑器 -Oracle Database 12c

Oracle Database 12c Library 生成

1



# 生成DDL



## 生成 DDL

```
8 CREATE TABLE books (  
9     book_id          VARCHAR2(20) NOT NULL,  
10    title             VARCHAR2(50) NOT NULL,  
11    author_last_name  VARCHAR2(25) NOT NULL,  
12    author_first_name VARCHAR2(25),  
13    rating            NUMBER(2)  
14 );  
15  
16 ALTER TABLE books ADD CONSTRAINT books_pk PRIMARY KEY ( book_id );
```

Books表的定义



# 保存设计

Oracle SQL Developer Data Modeler

The screenshot displays the Oracle SQL Developer Data Modeler interface. The '文件(F)' (File) menu is open, showing options such as '打开' (Open), '关闭' (Close), '全部关闭' (Close All), '保存' (Save), '另存为...' (Save As...), '导入(I)' (Import), '导出(E)' (Export), '报表' (Reports), '页面设置...' (Page Setup...), '打印...' (Print...), '打印图表(G)' (Print Diagram), '最近设计(D)' (Recent Designs), and '退出(X)' (Exit). The background shows a logical data model with three tables: 'Books', 'Transactions', and 'Patrons'. 'Books' has primary key 'book\_id' and attributes 'title', 'author\_last\_name', 'author\_first\_name', and 'rating'. 'Transactions' has primary key 'transaction\_id' and attributes 'transaction\_date' and 'transaction\_type'. 'Patrons' has primary key 'patron\_id' and attributes 'last\_name', 'first\_name', 'street\_address', 'city', 'state', 'zip', and 'location'. Relationships are shown with crow's foot notation: 'Transactions' is connected to 'Books' and 'Patrons' with one-to-many relationships.

文件(F) 编辑(E) 查看(V) 小组(M) 工具(T) 窗口(W) 帮助(H) Automation

打开 Ctrl-O

关闭 Ctrl-W

全部关闭

保存 Ctrl-S

另存为... Ctrl+Alt+Shift-S

导入(I)

导出(E)

报表

页面设置... Ctrl-G

打印... Ctrl-P

打印图表(G)

最近设计(D)

退出(X) Alt-F4

起始页 Logical (my\_library) Library (my\_library)

**Books**

- P \* book\_id VARCHAR2 (20)
- \* title VARCHAR2 (50)
- \* author\_last\_name VARCHAR2 (25)
- author\_first\_name VARCHAR2 (25)
- rating NUMBER (2)
- Books\_PK (book\_id)

**Transactions**

- P \* transaction\_id NUMBER (7)
- \* transaction\_date DATE
- \* transaction\_type NUMBER (7)
- \* Books book\_id VARCHAR2 (20)

**Patrons**

- P \* patron\_id
- \* last\_name
- first\_name
- street\_address
- city
- state
- zip
- location
- Patrons\_PK (patron\_id)

# 作业

考虑一个电影数据库，其中记录了有关电影行业的数据。数据要求总结如下：

- 每部电影都有片名(title)和发行年份(year of release)。每部电影有一个时长(length)，比如120分钟。每部电影类型都有一个制作公司(production company)，并且属于一种或多种类型(genre)（如恐怖、动作、戏剧等）。每部电影都有一个或多个导演(director)，其中有一个或多个演员(actor)。每部电影都有一个情节大纲(plot)。最后，每部电影都有零条或多条可引用的引语(quote)，每一条引语都是由电影中出现的一位演员所说的，一位演员可以提供多条引语。
- 演员通过名字(name)和出生日期(date of birth)来识别，并出现在一部或多部电影中。每个演员在电影中都有一个角色(role)。
- 导演也可以通过姓名(name)和出生日期来确定，并导演一部或多部电影。导演也可以在电影中表演（包括他或她可能导演的电影）。
- 制作公司按名称(name)标识，每个公司都有地址。制作公司制作一部或多部电影。

## 作业要求

- 属性的数据类型可以自己合理地定义
- 画出E-R图
- 转换成一系列关系模式
- 生成这一系列关系模式的DDL（SQL脚本）



## 作业提交须知

- 提交的文件包括：  
实验报告.doc（包含sql代码文本和两个设计的清晰截图）  
在课程中心(course.xmu.edu.cn)提交实验报告
- 截止时间：2023-05-26 23:59:59



# 作业提交须知

## ○ 迟交作业

```
update sc
set grade=0
where sno='2432018220xxxx'
      and grade is null
      and miss_deadline = 1;
```

## ○ 抄袭作业

```
update sc
set grade=0
where sno='2432018220xxxx'
      and grade is null
      and plagiarize = 1;
```

