

北京邮电大学 实验报告



课程名称：

数据库系统原理

实验名称：

E/R 建模与关系表转换

计算机科学与技术 系 1
班

姓名： 李智盛

计算机科学与技术 系 1
班

姓名： 李旺

教师： 吴启凡

得分：

2019 年 4 月 1 日

目录

（一）实验目的：	2
（二）实验内容：	2
（三）实验环境：	2
（四）实验要求：	3
（五）实验步骤：	3
一、建立概念模型 CDM.....	3
一、新建概念数据模型.....	3
二、创建新实体.....	5
三、添加实体属性.....	7
四、定义属性的标准检查约束.....	9
五、添加联系.....	11
1.对于没有属性的联系：用 Relationship;.....	11
2.对于有属性的联系： Association:	12
六、 定义联系的特性.....	12
七、 定义联系的角色名.....	13
八、 有关联系的基数.....	13
二、建立物理数据模型 PDM	14
1.生成.....	14
2.设置.....	15
3.完成.....	15
三、使用将物理模型转化为生成数据库中的表和视图的脚本	16
1.生成.....	16
2.设置.....	16
四、执行 SQL 脚本，生成表和视图。	17
1.新建数据库.....	17
2.执行脚本文件.....	17
3.查看表、关系视图.....	17
（六）实验总结：	19
1.数据类型的对应关系.....	19
2.约束条件的相应表示.....	22
3.DataItam 重用.....	22
4. 小结.....	23

（一）实验目的：

1. 理解和掌握E-R图的基本概念。
2. 培养根据实际应用领域数据对象描述，抽取数据对象特征、关联关系等信息，设计数据库概念结构的能力。
3. 选做：学习Power Designer或其它工具，进行数据模型转换和关系表的自动创建，培养软件辅助设计工具的使用能力。

（二）实验内容：

1. 根据数据需求描述抽象出E-R图
2. 阅读《TD-LTE网络配置数据库》课程实验背景资料及数据建模-17-v3.doc，根据LTE的基本概念，分析其中的数据需求，将其描述抽象成实体和联系，并确定实体和联系的属性，特别要注意标明其主键和外键等约束关系，最终形成E-R图。
3. 将E-R图输入相关设计工具（ERWin、Power Design、SQL Server自带设计工具或者Visio等）形成概念模型。
4. 使用工具将E-R图转换为数据库物理结构。
5. 使用将物理模型转化为生成数据库中的表和视图的脚本，注意要选择数据库为SQL Server。
6. 执行SQL脚本，生成表和视图。
7. 成功后，查看生成的表和视图的情况。

（三）实验环境：

1. Windows10 操作系统
2. Power designer 16.5 数据库概念设计工具

3. Microsoft SQL Server 2012 数据库管理系统

（四）实验要求：

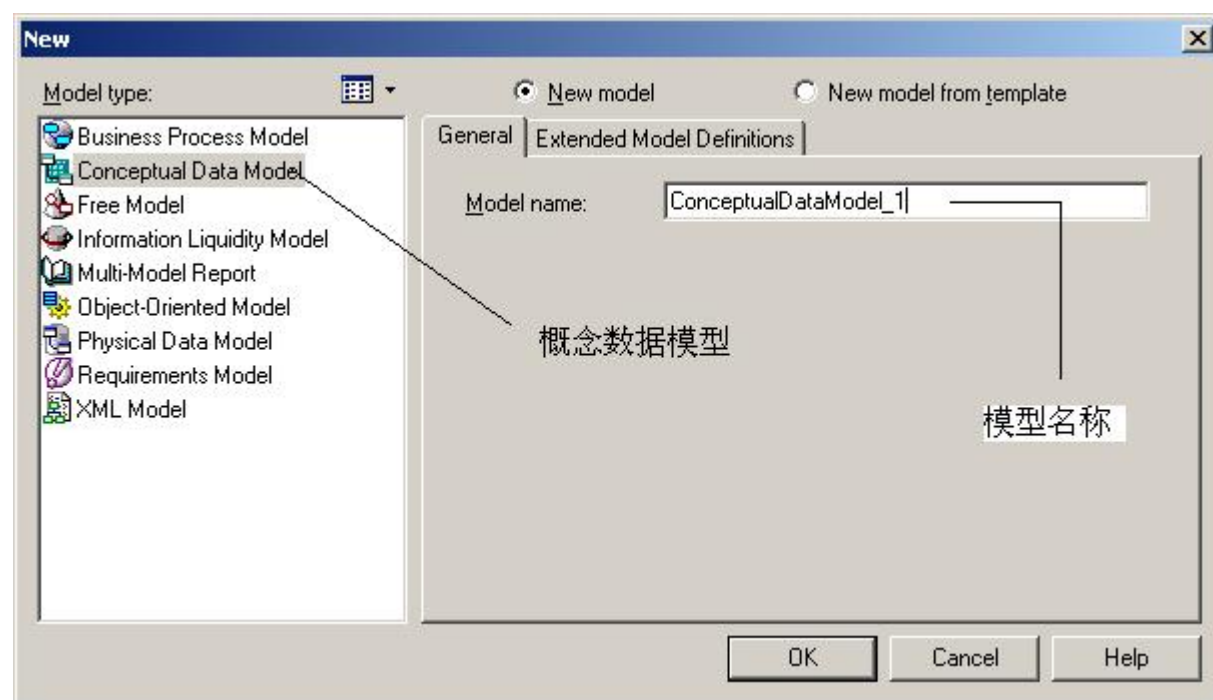
要求在实验报告中详细记录整个过程并详细描述遇到的主要问题。

（五）实验步骤：

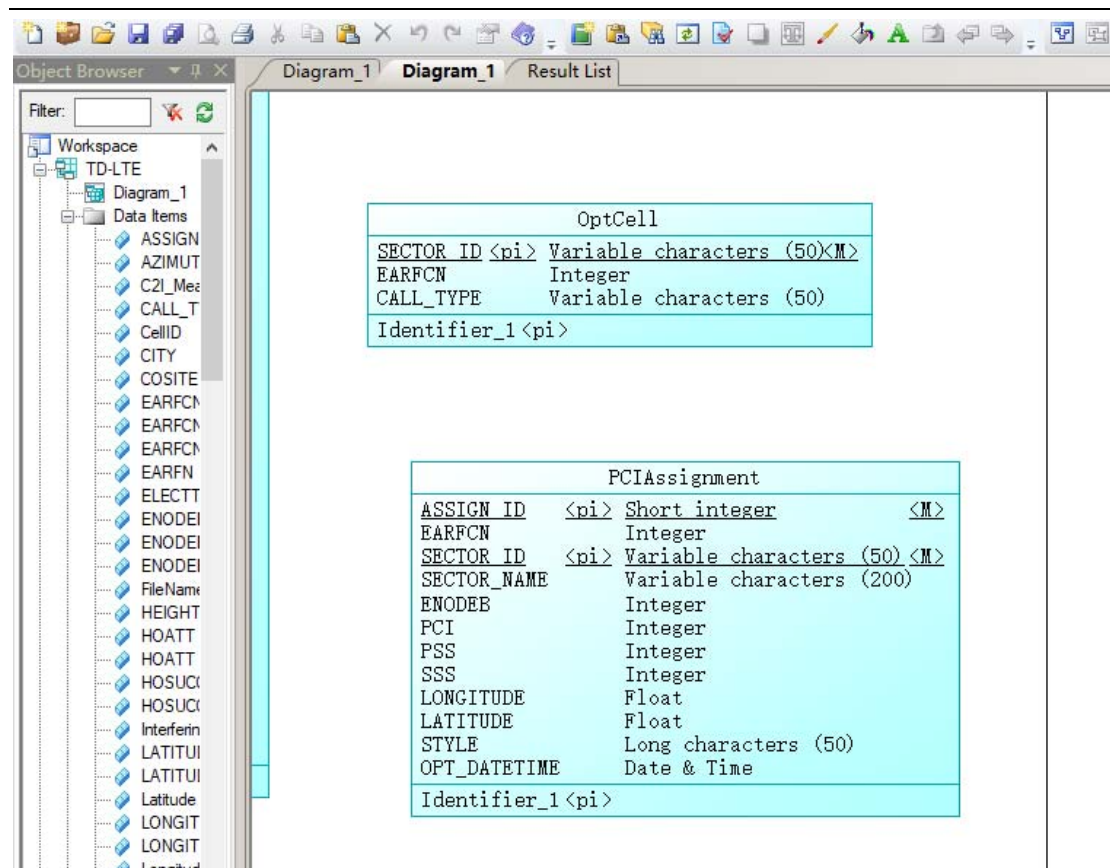
一、建立概念模型 CDM

一、新建概念数据模型

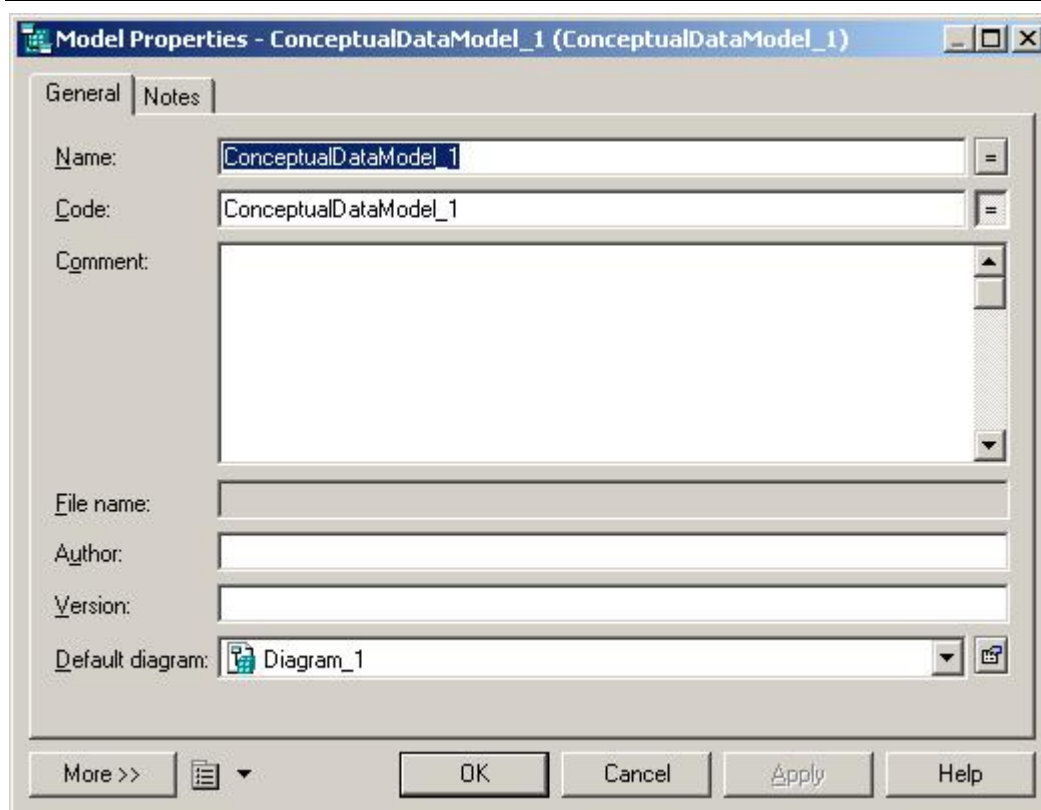
1) 选择 File-->New,弹出如图所示对话框，选择 CDM 模型（即概念数据模型）建立模型。



2) 完成概念数据模型的创建。以下图示

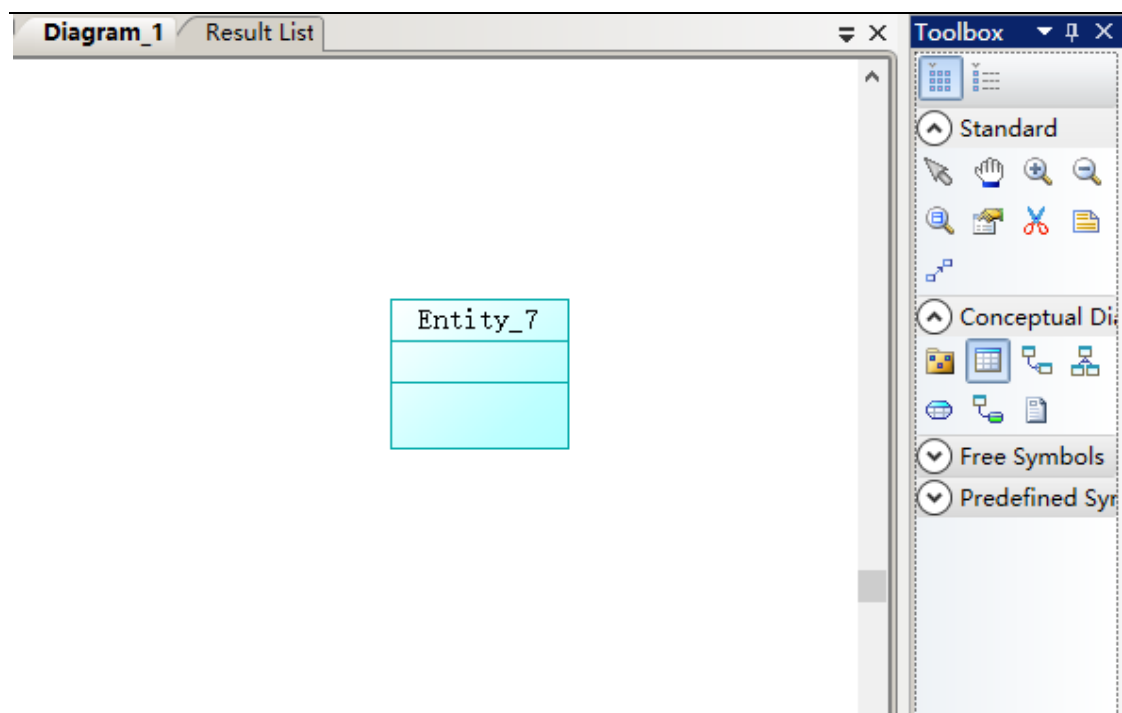


3) 选择新增的 CDM 模型，右击，在弹出的菜单中选择 “Properties” 属性项，弹出如图所示对话框。在 “General” 标签里可以输入所建模型的名称、代码、描述、创建者、版本以及默认的图表等等信息

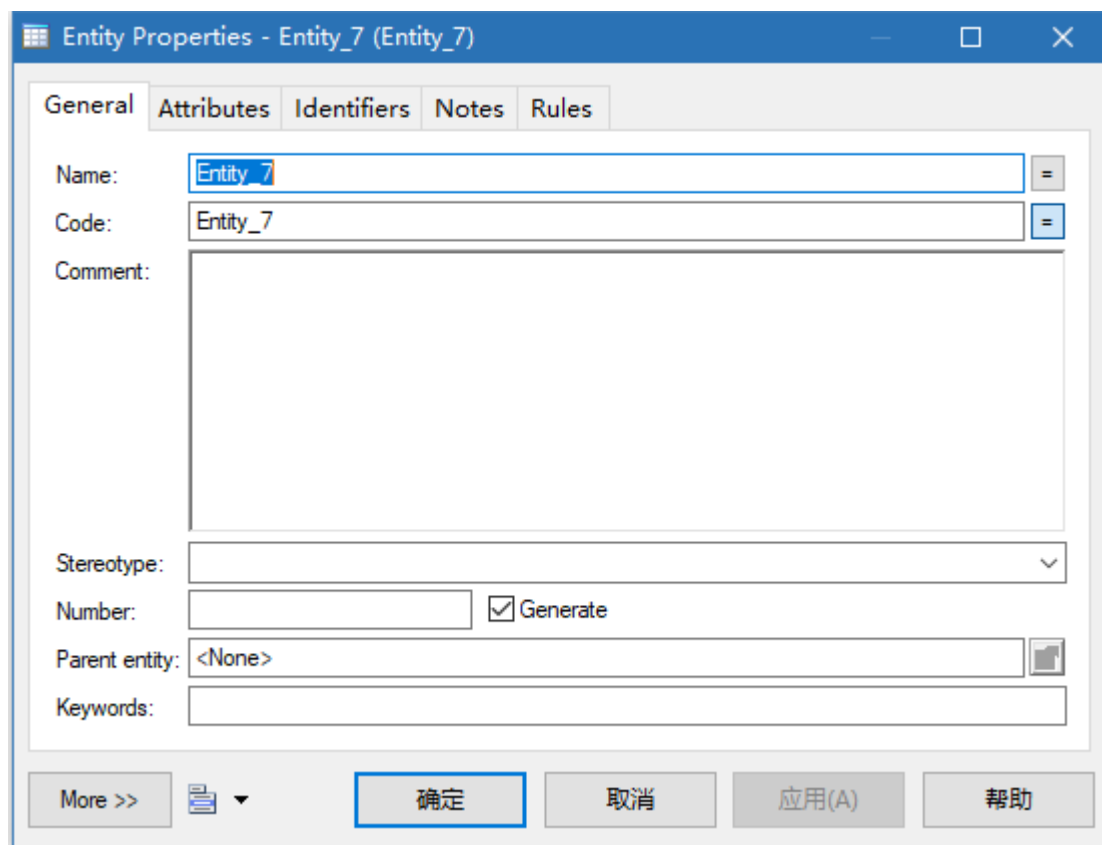


二、创建新实体

1) 在 CDM 的图形窗口中, 单击工具选项版上的 Entity 工具, 再单击图形窗口的空白处, 在单击的位置就出现一个实体符号。点击 Pointer 工具或右击鼠标, 释放 Entity 工具。如图所示



2) 双击刚创建的实体符号，打开下列图标窗口，在此窗口“General”标签中可以输入实体的名称、代码、描述等信息。



Cell		
CITY	Variable multibyte (255)	
SECTOR_ID <pi>	Variable multibyte (50) <M>	
SECTOR_NAME	Variable characters (255) <M>	
EARFN	Integer	<M>
PCI	Integer	
PSS	Integer	
SSS	Integer	
TAC	Integer	
AZIMUTH	Float	<M>
HEIGHT	Float	
ELECTTILT	Float	
MECHTILT	Float	
TOTLETILT	Float	<M>
Identifier_1 <pi>		

2) 在上图所示窗口中, 点击插入属性按钮, 弹出属性对话框, 如下图所示。

Entity Attribute Properties - Attribute_91 (Attribute_91)

General Standard Checks Notes Rules

Name: Attribute_91 =

Code: Attribute_91 =

Comment:

Stereotype:

Entity: Entity_7

Data Item: Attribute_91

Data type: <Undefined> ...

Length: Precision:

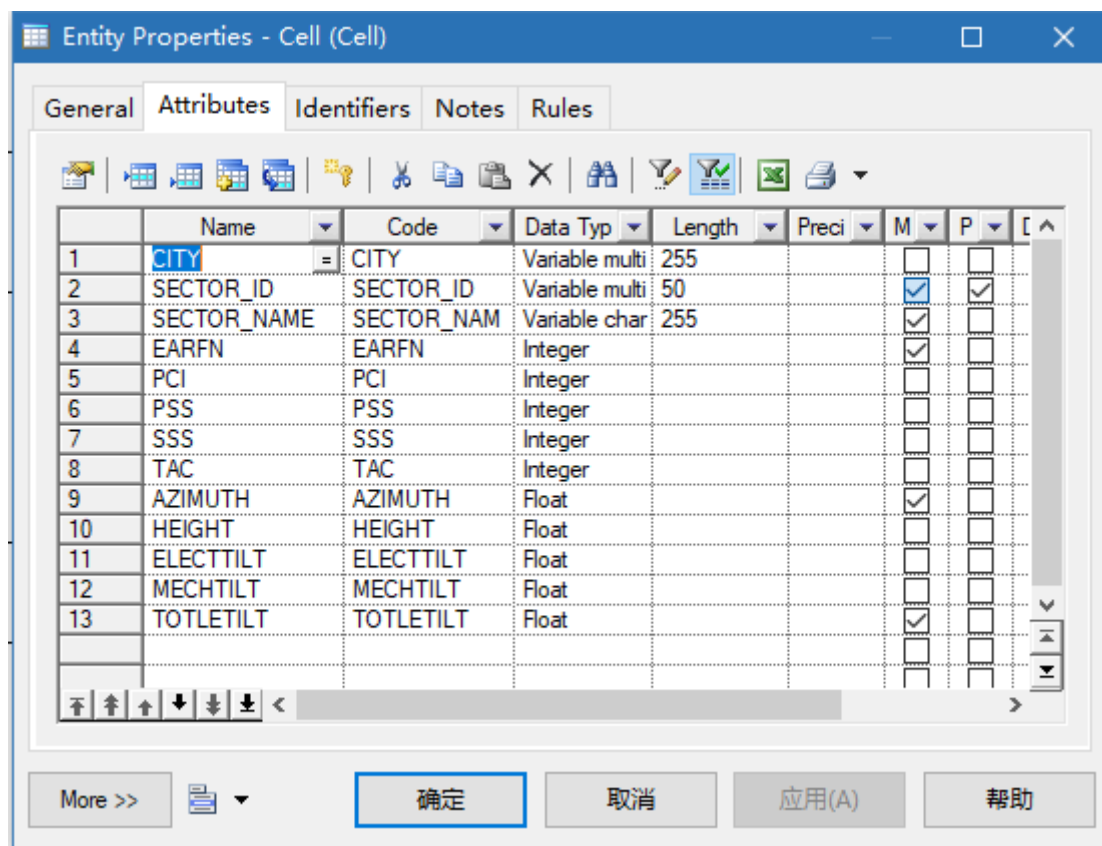
Domain: <None> ...

☐ Primary identifier ☐ Mandatory ☒ Displayed

Keywords:

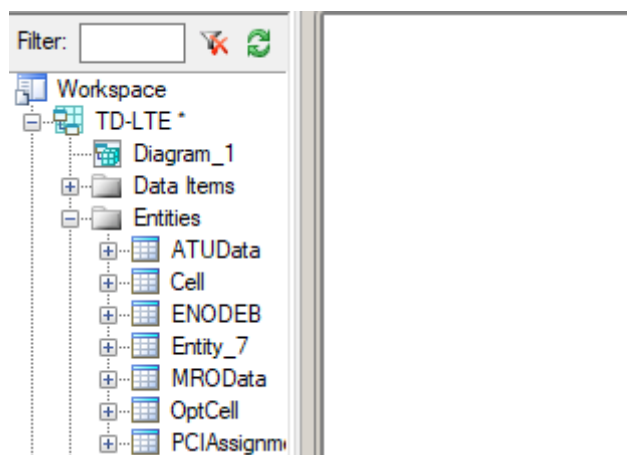
More >> 确定 取消 应用(A) 帮助

在 Arributes 中添加相应的属性：



依次继续创建其他实体并添加属性

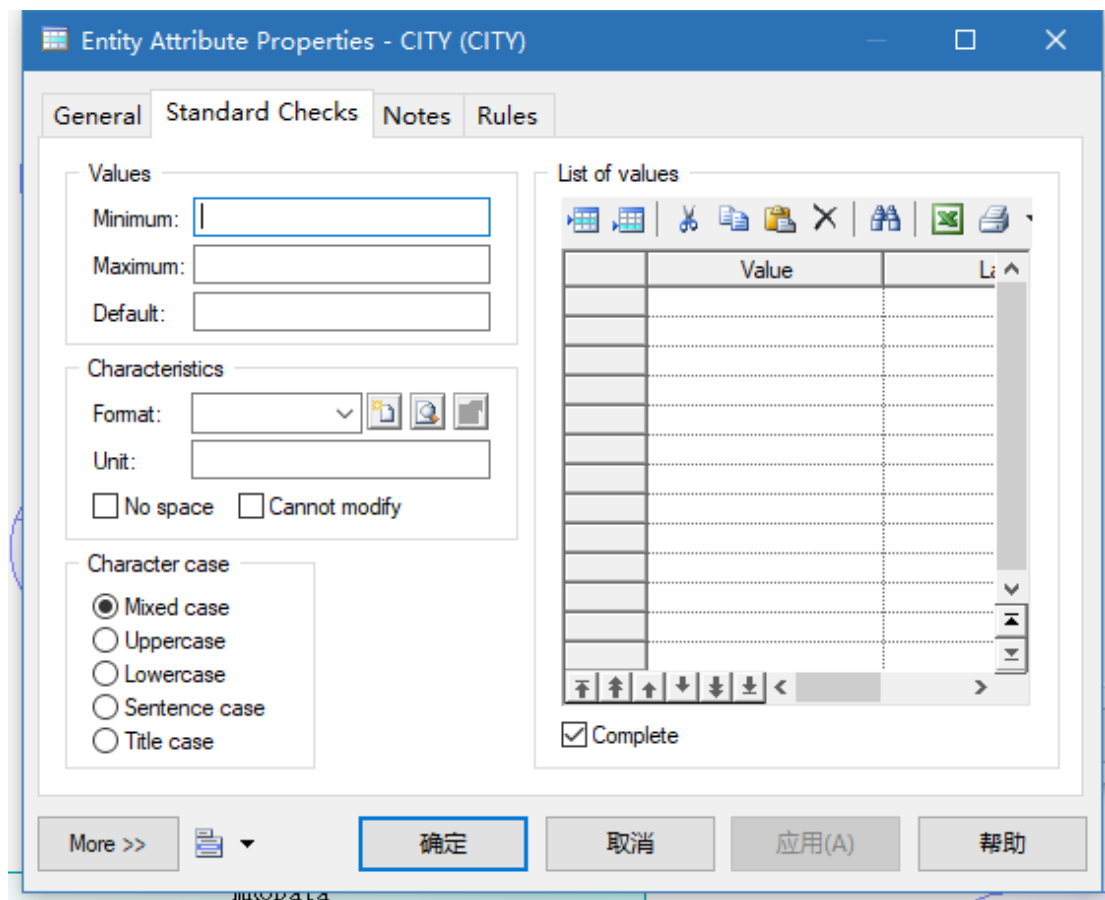
至此六个实体全部添加完成；



四、定义属性的标准检查约束

标准检查约束是一组确保属性有效的表达式。在实体属性的特性窗口，打开如图

所示的检查选项卡。



在这个选项卡可以定义属性的标准检查约束，窗口中每项的参数的含义，

参数说明

Minimum	属性可接受的最小数
Default	属性不赋值时, 系统提供的默认值 Unit 单位, 如公里、吨、元
Lowercase	属性的赋值全部变为小写字母
Cannot modify	该属性一旦赋值不能再修改
Label	属性列表值的标签

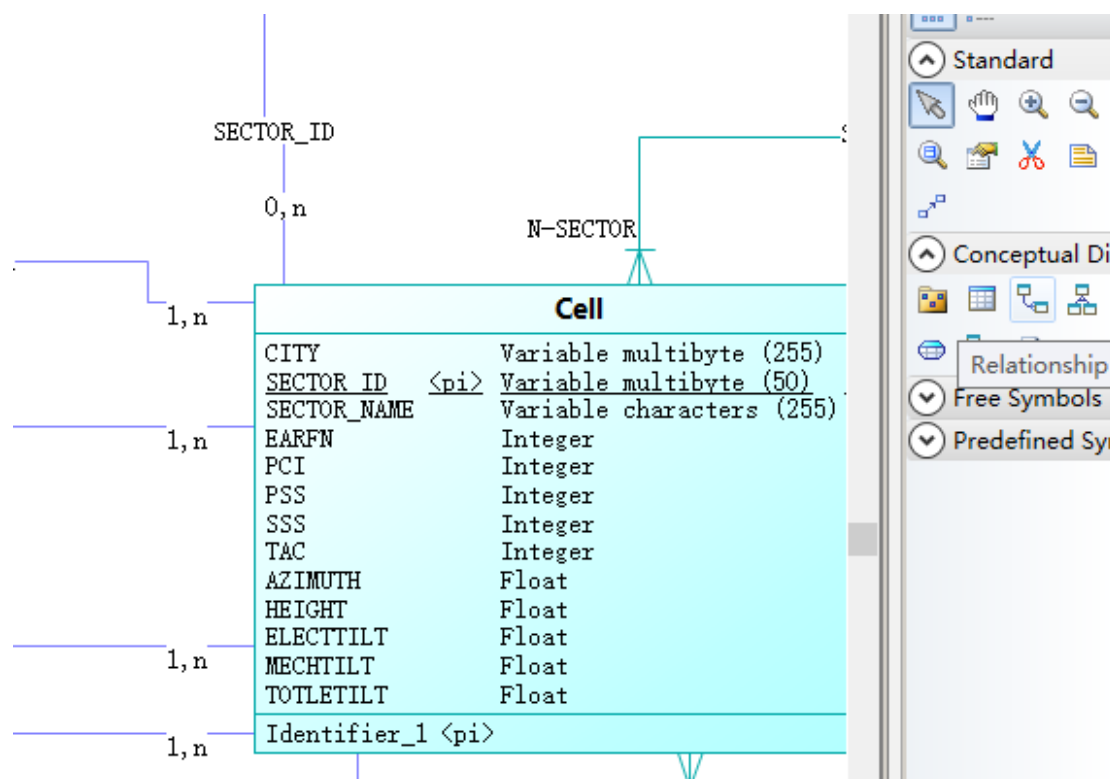
五、添加联系

联系 (Relationship) 是指实体集这间或实体集内部实例之间的连接。

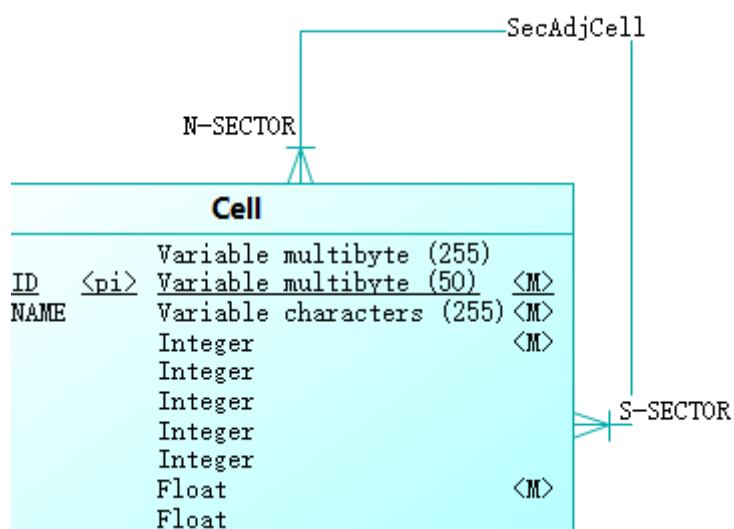
实体之间可以通过联系来相互关联。与实体和实体集对应，联系也可以分为联系和联系集，联系集是实体集之间的联系，联系是实体之间的联系，联系是具有方向性的。联系和联系集在含义明确的情况之下均可称为联系。

按照实体类型中实例之间的数量对应关系，通常可将联系分为 4 类，即一对一 (ONE TO ONE) 联系、一对多 (ONE TO MANY) 联系、多对一 (MANY TO ONE) 联系和多对多联系 (MANY TO MANY)。

在 CDM 工具选项板中除了公共的工具外，还包括如下图所示的其它对象产生工具。

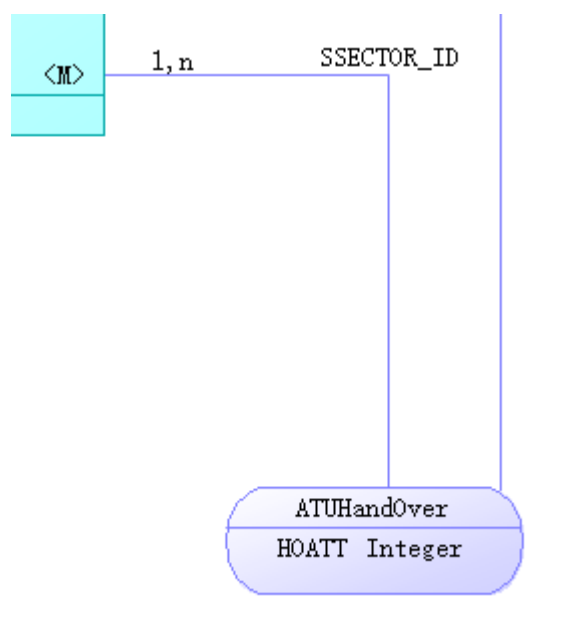


1.对于没有属性的联系：用 Relationship;



无属性的联系：SecAdjCell

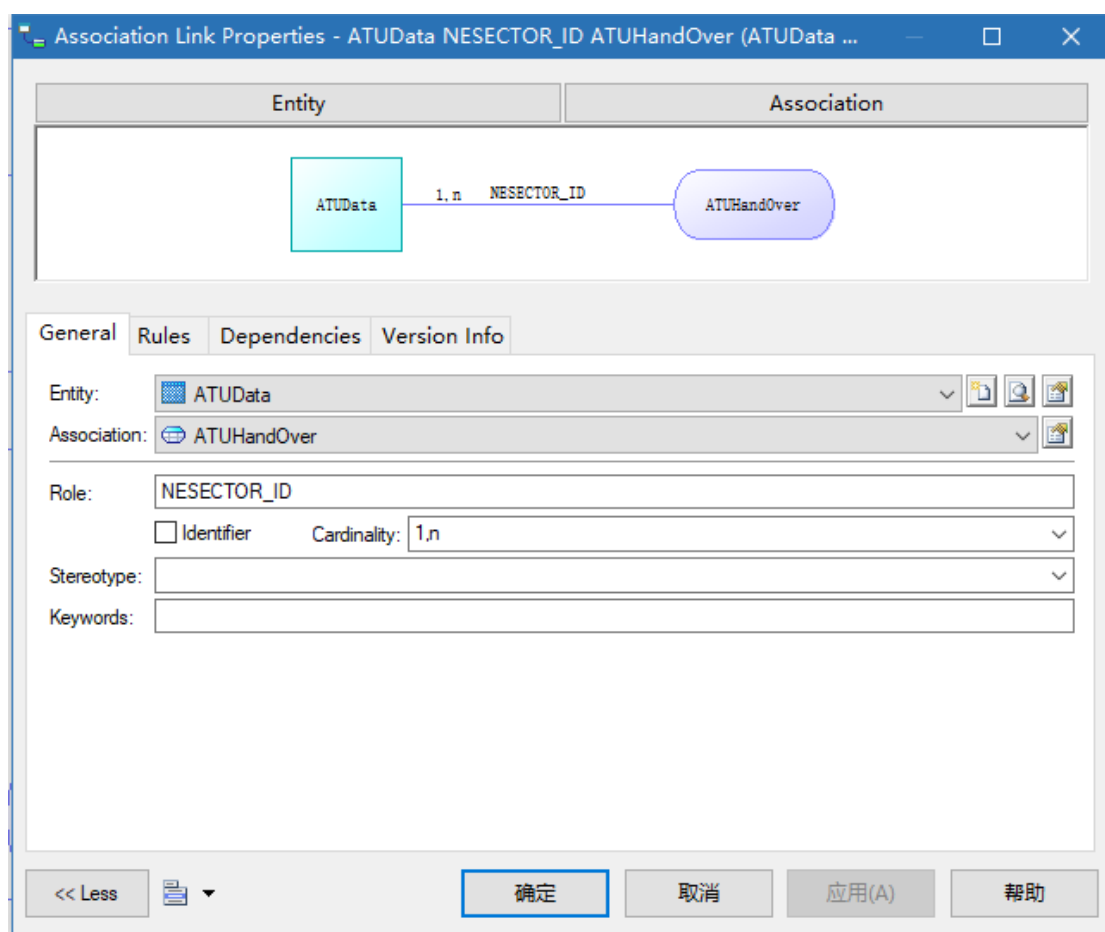
2.对于有属性的联系：Association:



有属性的联系：ATUHandOver

六、 定义联系的特性

在两个实体间建立了联系后，双击联系线，打开联系特性窗口，如图所示



七、 定义联系的角色名

在联系的两个方向上各自包含有一个分组框，其中的参数只对这个方向起作用，

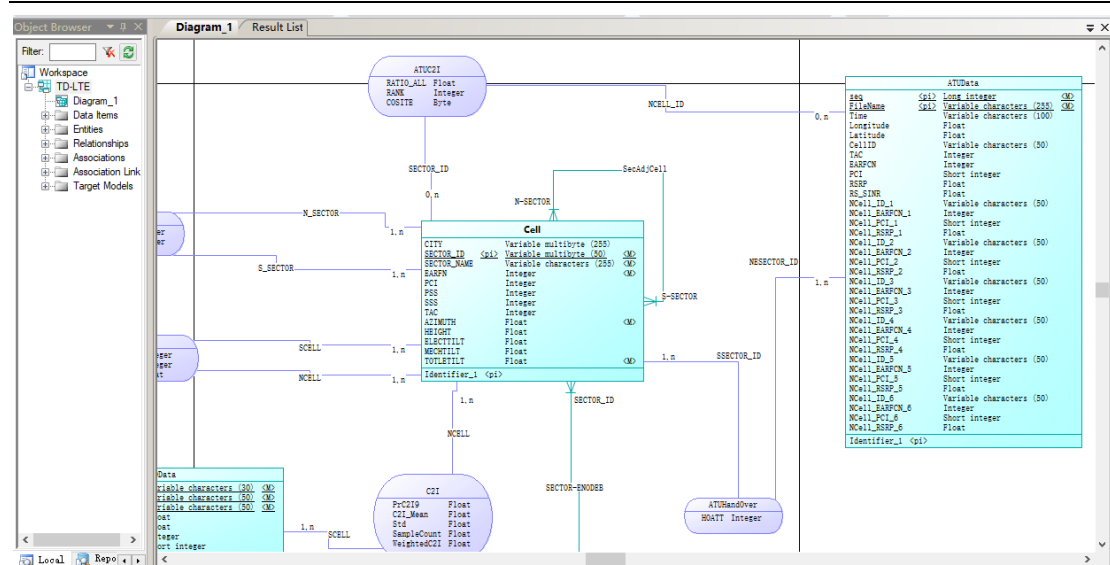
Role Name 为角色名，描述该方向联系的作用

八、 有关联系的基数

联系具有方向性，每个方向上都有一个基数。具体参考课本《数据库系统原理》

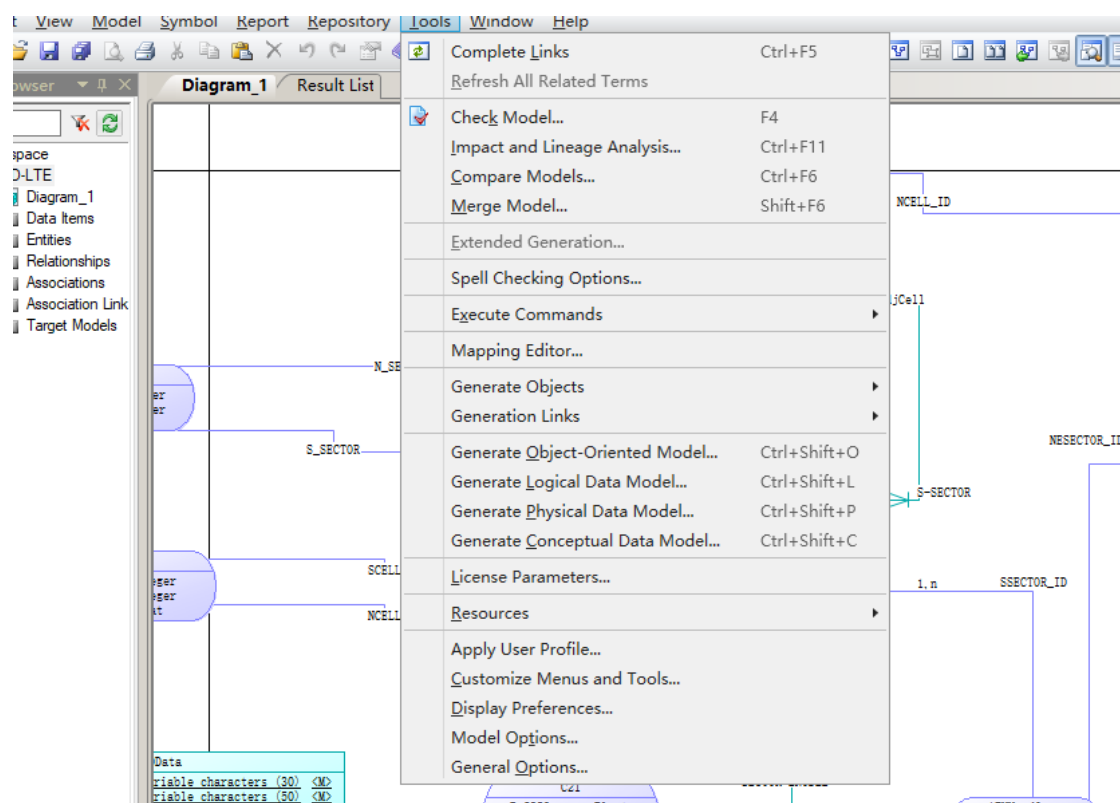
第七章：映射基数

至此，概念模型已基本完成：



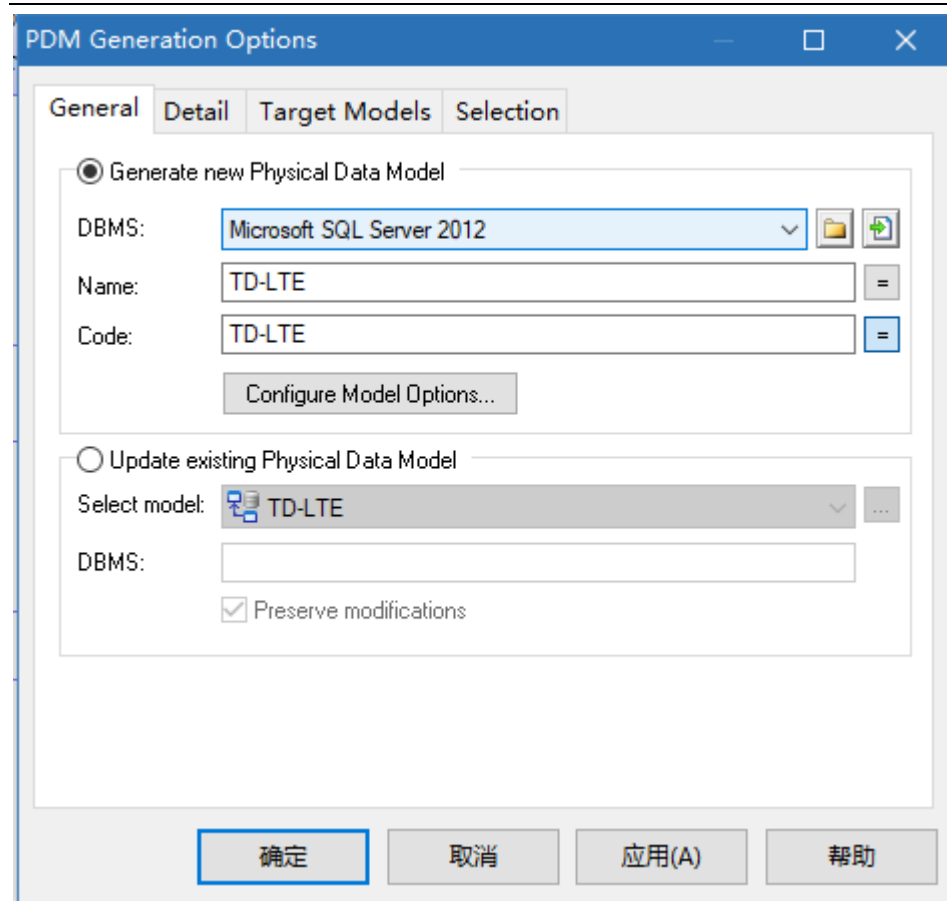
二、建立物理数据模型 PDM

将概念模型 E-R 图 (CDM) 转化为 PDM 步骤如下:



1.生成

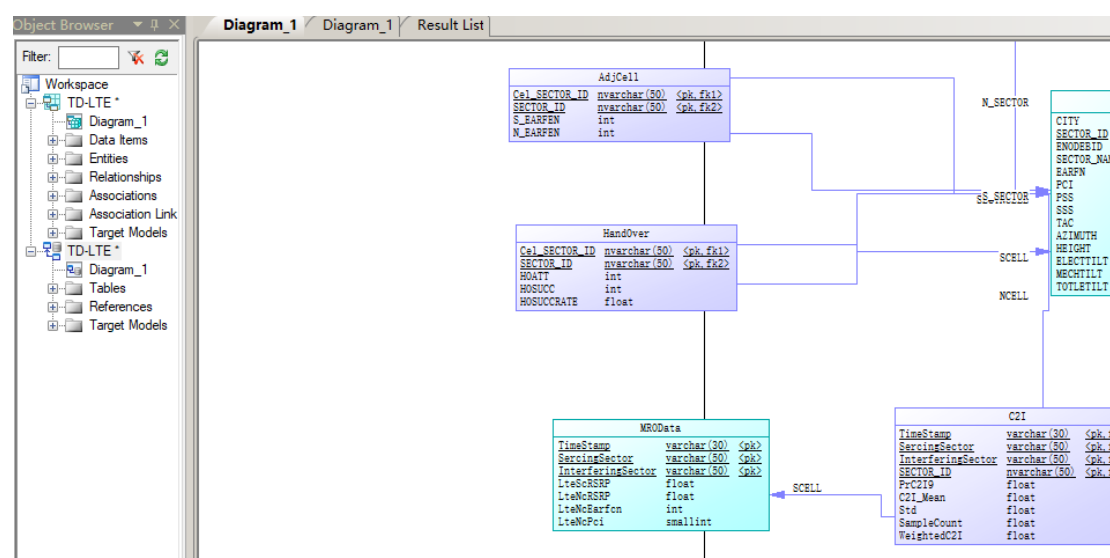
点击 Generate Physical Data Model



2.设置

设置适合自己 SQL 的版本, 物理模型的名称

3.完成

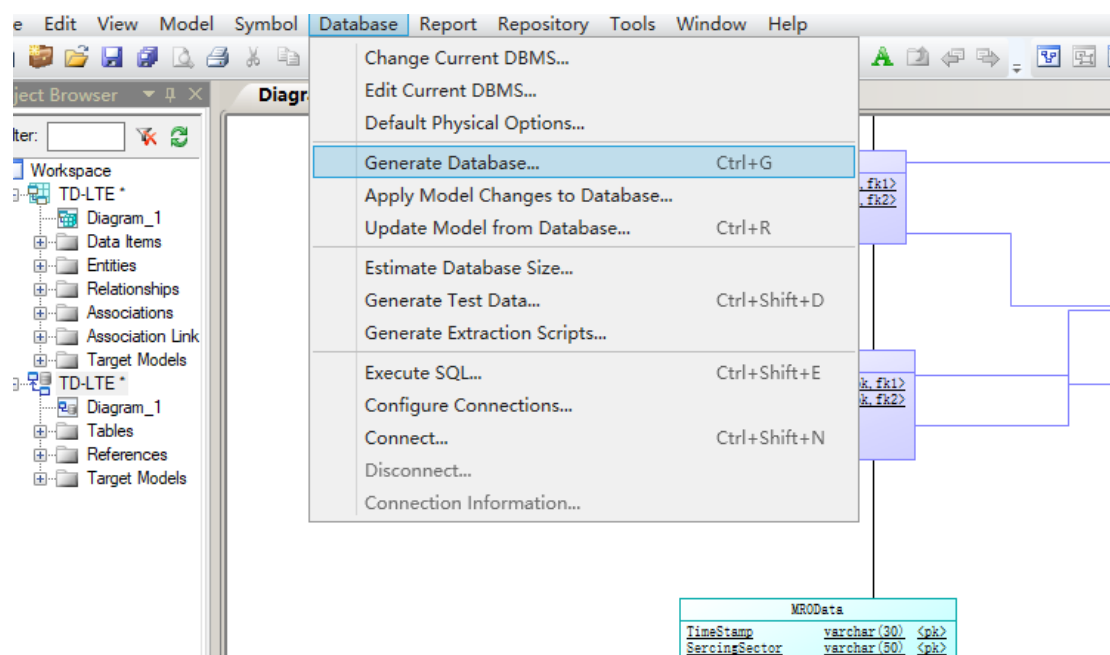


左侧 Browser 资源栏中生成了物理模型的资源列表, 右侧生成了对应的物理模

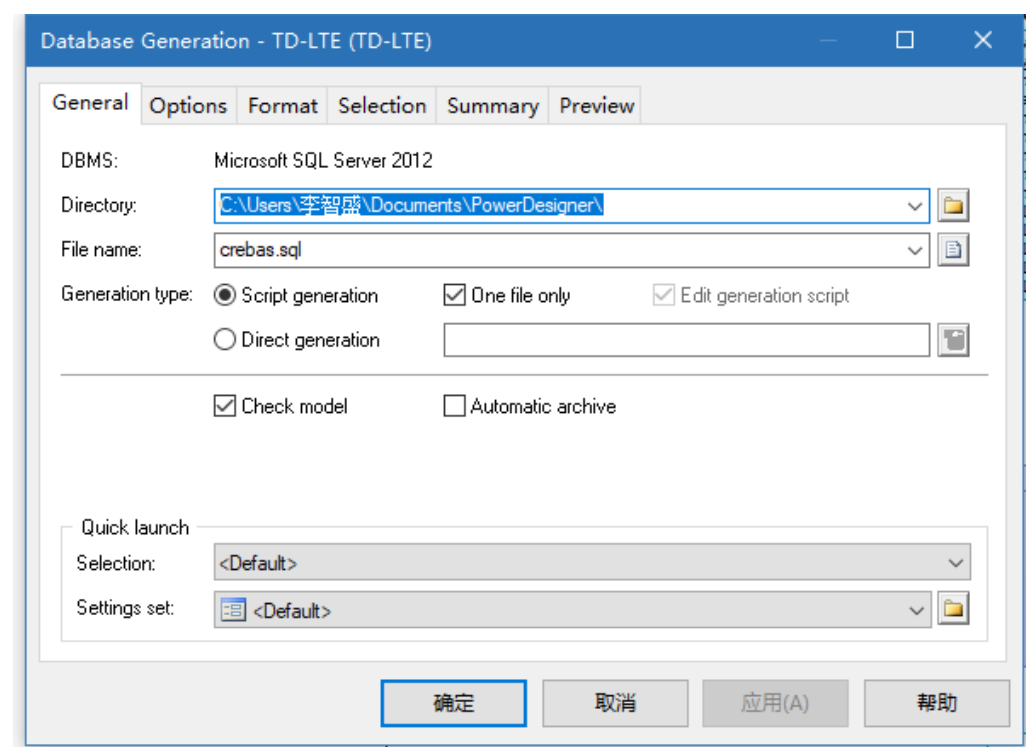
型图。

三、使用将物理模型转化为生成数据库中的表和视图的脚本

1.生成



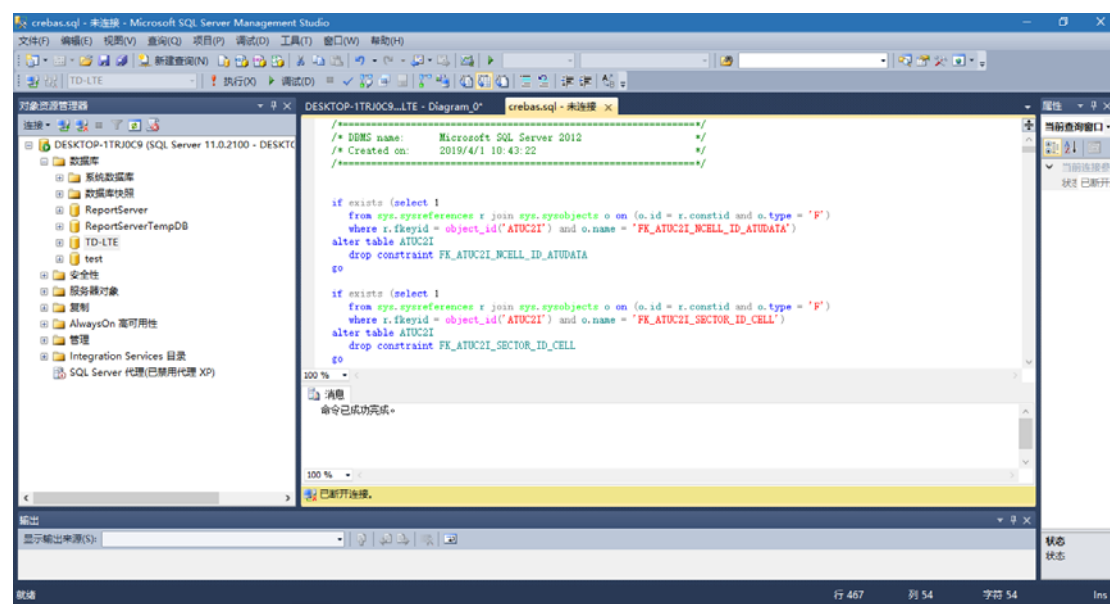
2.设置



设置生成脚本文件的路径，名称等；其他选择默认

四、执行 SQL 脚本，生成表和视图。

1.新建数据库

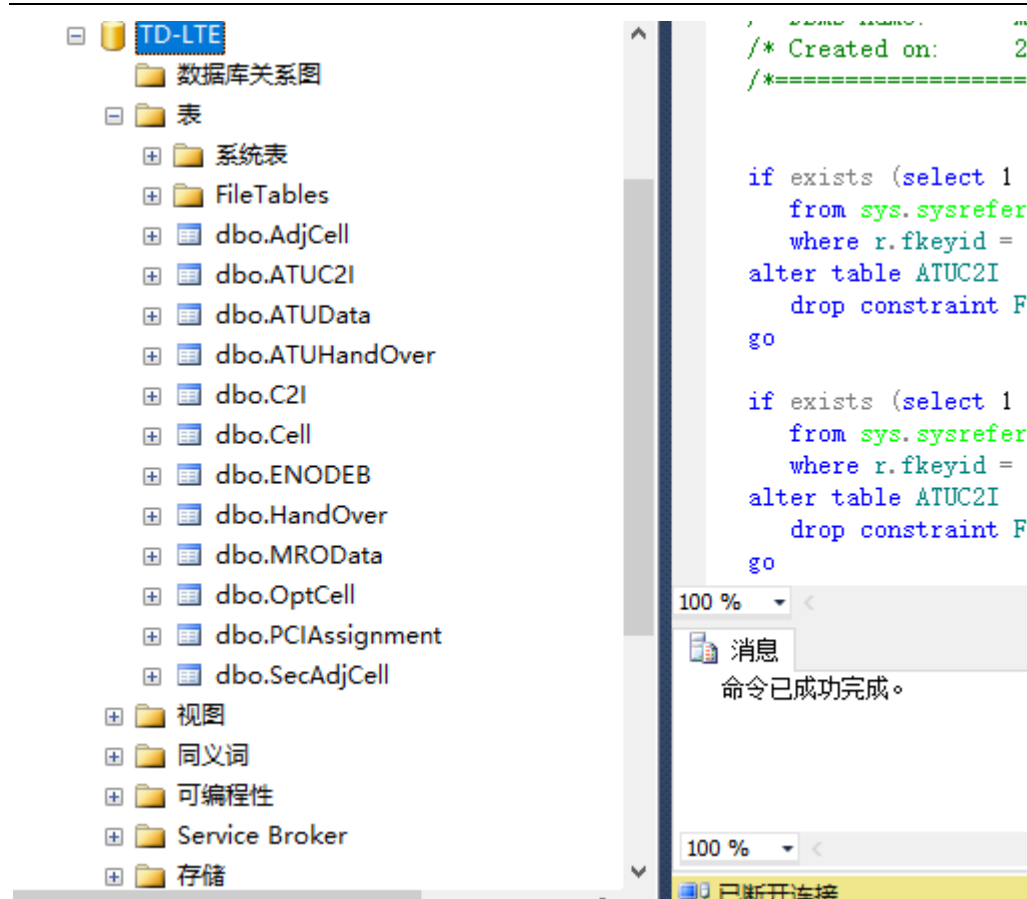


新建数据库 TD-LTE

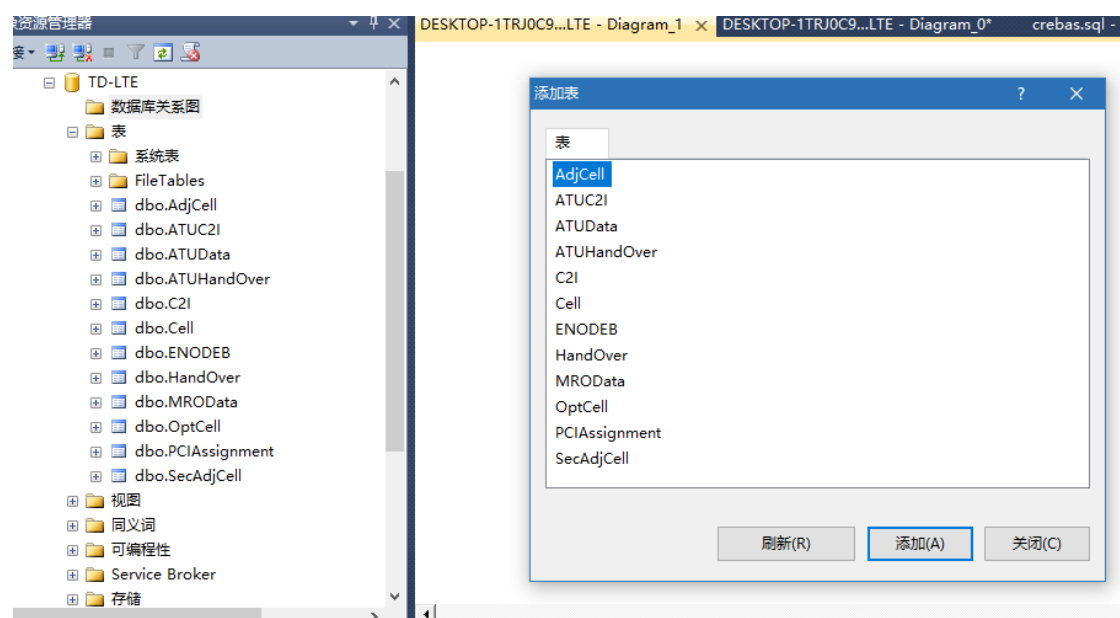
2.执行脚本文件

打开生成的脚本文件，运行环境在下拉菜单中选择刚才创建的数据库 TD-LTE，执行文件；

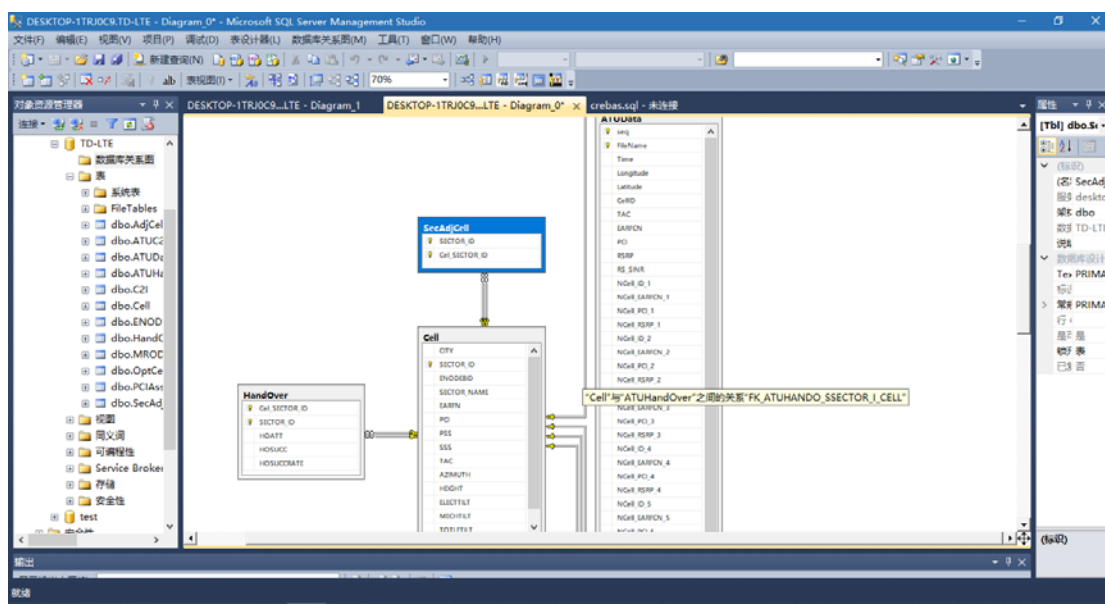
3.查看表、关系视图



这里可以在数据库关系图-右键-新建数据库关系图



全部选择，点击生成；



（六）实验总结：

1.数据类型的对应关系

TD-LTE 要求文档中的数据类型，如 nvarchar 等，在 Powerdesigner 实体属性的数据类型选项卡中找不到对应的名字；查阅相应资料发现，是命名的原因

Numeric data types

Standard datatype	DBMS-specific physical data type	Content	Length
Integer	int / INTEGER	32-bit integer	—
Short Integer	smallint / SMALLINT	16-bit integer	—
Long Integer	int / INTEGER	32-bit integer	—
Byte	tinyint / SMALLINT	256 values	—
Number	numeric / NUMBER	Numbers with a fixed decimal point	Fixed
Decimal	decimal / NUMBER	Numbers with a fixed decimal point	Fixed

Float	float / FLOAT	32- bit floating point numbers	Fixed
Short Float	real / FLOAT	Less than 32- bit point decimal number	—
Long Float	double precision / BINARY DOUBLE	64- bit floating point numbers	—
Money	money / NUMBER	Numbers with a fixed decimal point	Fixed
Serial	numeric / NUMBER	Automatically incremented numbers	Fixed
Boolean	bit / SMALLINT	Two opposing values (true/false; yes/no; 1/0)	—

Character data types

The following character data types are available:

Standard data type	DBMS-specific physical data type	Content	Length
Characters	char / CHAR	Character strings	Fixed
Variable Characters	varchar / VARCHAR2	Character strings	Maximum
Long Characters	varchar / CLOB	Character strings	Maximum
Long Var Characters	text / CLOB	Character strings	Maximum
Text	text / CLOB	Character strings	Maximum
Multibyte	nchar / NCHAR	Multibyte character strings	Fixed
Variable Multibyte	nvarchar / NVARCHAR2	Multibyte character strings	Maximum

Time data types

Standard data type	DBMS-specific physical data type	Content	Length
Date	date / DATE	Day, month, year	—
Time	time / DATE	Hour, minute, and second	—
Date & Time	datetime / DATE	Date and time	—
Timestamp	timestamp / TIMESTAMP	System date and time	—

Other data types

The following other data types are available:

Standard data type	DBMS-specific physical data type	Content	Length
Binary	binary / RAW	Binary strings	Maximum
Long Binary	image / BLOB	Binary strings	Maximum
Bitmap	image / BLOB	Images in bitmap format (BMP)	Maximum
Image	image / BLOB	Images	Maximum
OLE	image / BLOB	OLE links	Maximum
Other	—	User-defined data type	—
Undefined	undefined	Undefined. Replaced by the default data type at generation.	—

2.约束条件的相应表示

PowerDesigner 中的业务规则 (Business Rules) 的种类

1、constraint(约束): 用来描述实体属性之间的限制与约束

例子: 入厂日期要大于出生日期

2、definition(定义): 用来描述实体

例子: 一个作家由一个名字和一个地址来识别

3、fact(事实): 用来描述实体之间的联系

例子: 一个出版社可以出版一个或多个主题的图书

4、formula(公式): 用来描述属性的数值运算约束

例子: 总金额等于所有订单金额之和

5、validation(有效性): 用来描述实体的属性取值约束

例子: 性别只能是“男”或“女”, 不能为空

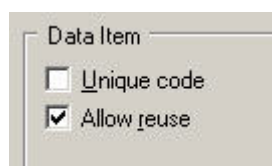
6、requirement(需求): 是信息系统中功能的详细说明

例子: 模型被设计以致版税的总数量不超过总售卖的 10%

3.DataItem 重用

设置实体属性时, 有两个实体属性列表中都有 SECTOR_ID;在第二个实体中添加时, 出现报错。检查后发现是 Power designer 不允许数据项重名导致的。

“工具”菜单中有个选项“Model Option”, 其中有一项设置, 如下图:



其默认选项是“Unique code”处于选中状态。PowerDesigner 默认在 CDM

中不能存在相同名称的实体属性,这也是考虑到可能产生的一些如主键外键等名称冲突问题,但当我们进行实际数据库设计时,可能会多次使用相同数据项 (DataItem) 便于理解各实体。

4. 小结

此次实验学习了 Power Designer 的使用,进行数据模型转换和关系表的自动创建,培养软件辅助设计工具的使用能力;可以提高编程效率和降低软件错误。在数据库系统设计开发中,可以利用多种数据复制工具和直接将 E-R 图转化成数据库表结构的工具。Power Designer 是一种数据库概念设计工具,它支持将数据库概念结构转化为物理结构,然后再转化为 SQL 脚本,从而在数据库中直接生成表结构。同时也熟悉了 SQL Server 数据库的基本操作,脚本的执行、数据库关系图的生成等。实验起初,各步骤不是很清楚明了,具体操作细节需要自己摸索;也遇到了很多问题,通过查找资料、同学合作等途径都逐个解决了,进一步锻炼了动手能力,为后续实验做好了铺垫。