

数据库设计实验

16337113

劳马东

计算机科学与技术（超算方向）

一、实验目的

掌握数据库设计基本方法及数据库设计工具。

二、实验内容和要求

掌握数据库设计基本步骤，包括数据库概念结构设计、逻辑结构设计、物理结构设计和数据库模式 SQL 语句生成。能够使用数据库设计工具进行数据库设计。

三、实验环境

系统	Windows 10
SQL	MySQL 8.0
工具	PowerDesigner

四、实验过程

设计一个采购、销售和客户管理应用数据库。其中，一个供应商可以供应多种零件，一种零件也可以有多个供应商。一个客户订单可以订购多种供应商供应的零件。客户和供应商都分属不同的国家，而国家按世界五大洲八大洋划分地区。请利用 PowerDesigner 或者 Erwin 等数据库设计工具设计数据库。

1. 概念结构设计

识别出零件 Part、供应商 Supplier、客户 Customer、订单 Order、订单项 Lineitem、国家 Nation、地区 Region 等 7 个实体集及其拥有的属性。

- 零件 Part：零件编号 partkey、零件名称 name、零件制造商 mfg、品牌 brand、类型 type、大小 size、零售价格 retailprice、包装 container、备注 comment。主码：零件编号 partkey。
- 供应商 Supplier：供应商编号 suppkey、供应商名称 name、地址 address、电话 phone、备注 comment（不需要 nation 属性）。主码：供应商编号 suppkey。
- 客户 Customer：客户编号 custkey、客户名称 name、地址 address、电话 phone、备注 comment（不需要 nation 属性）。主码：客户编号 custkey。
- 订单 Order：订单编号 orderkey、订单状态 status、订单总价 totalprice、订单日期

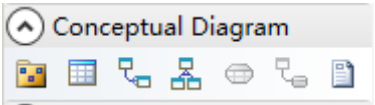
orderdate、订单优先级 orderpriority、记账员 clerk、运送优先级 shippriority、备注 comment。主码：订单编号 orderkey。

● 订单项 Lineitem：订单项编号 linenumber、零件数量 quantity、零件总价 extendedprice、折扣 discount、税率 tax、退货标记 returnflag 等（不需要 partkey 和 supkey 属性）。主码：订单项编号 linenumber。

● 国家 Nation：国家编号 nationkey、国家名称 name、备注 comment（不需要 region 属性）。主码：国家编号 nationkey。

● 地区 Region：地区编号 regionkey、地区名称 name、备注 comment。主码：地区编号 regionkey。

在 File->New Model 中新建一个 Conceptual Data Model。双击图中的第二个图标，即可在 Diagram 中新建一个实体集，同理双击第三个图标可新建一个联系集。



新建 Part 实体集，双击进入 Attribute 页添加实体集的属性，分别是 partkey、name、mfgr、brand 等。在最右侧，勾选 P(rimary)以表示 partkey 是 Part 实体集的主码，M 表示该属性的取值是否唯一（显然主码是 Unique 的），D(isplay)表示该属性是否显示在实体集上。

Entity Properties - Part (Part)

Entity Properties - Part (Part)										
General Attributes Identifiers Notes Rules										
	Name	Code	Data Type	Length	Preci	M	P	D		
→	partkey	partkey	Integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
2	name	name	Variable characters (25)	25		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3	mfgr	mfgr	Characters (25)	25		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
4	brand	brand	Characters (10)	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
5	type	type	Variable characters (25)	25		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
6	size	size	Integer			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
7	retailprice	retailprice	Float			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
8	container	container	Characters (10)	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
9	comment	comment	Variable characters (152)	152		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

以同样的方法新建 Supplier 实体集，它具有 supkey、name、address 等属性，以 supkey 为主码。

Entity Properties - Supplier (Supplier)

Entity Properties - Supplier (Supplier)										
General Attributes Identifiers Notes Rules										
	Name	Code	Data Type	Length	Preci	M	P	D		
→	supkey	supkey	Integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
2	name	name	Variable characters (25)	25		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3	address	address	Variable characters (40)	40		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
4	phone	phone	Characters (15)	15		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
5	acctbal	acctbal	Float			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
6	comment	comment	Variable characters (152)	152		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

创建完 Part 实体集和 Supplier 实体集后，在 Tools->Model Options 中设置 Notation 为 Entity/Relationship，就能在 Diagram 中看到如下两张图（默认 Notation（Barker）下

图形是圆角矩形，而且实体集名字和属性之间没有横线相隔，identifier 也不显示）。其中主码用下划线标出，<pi>表示主码，<M>表示属性是否取值唯一，最下方是实体集上的标识符（identifier）。

Part		
<u>partkey</u>	<pi> Integer	<M>
name	Variable characters (25)	
mfg	Characters (25)	
brand	Characters (10)	
type	Variable characters (25)	
size	Integer	
retailprice	Float	
container	Characters (10)	
comment	Variable characters (152)	
id_partkey <pi>		

Supplier		
<u>suppkey</u>	<pi> Integer	<M>
name	Variable characters (25)	
address	Variable characters (40)	
phone	Characters (15)	
acctbal	Float	
comment	Variable characters (152)	
id_suppkey <pi>		

接下来，考虑 Part 实体集和 Supplier 实体集之间的联系。一个供应商可以供应多种零件，一种零件也可以有多个供应商，因此二者之间存在一个多对多联系，记为 PartSupp。而由于不允许供应商不供应零件，不允许零件没有供应商，因此 PartSupp 是一个全参与的联系集。

点击右侧的图标双击联系集图标，然后单击 Part 和 Supplier 两个实体集，就在二者之间建立了一个联系集（默认为多对 1）。双击命名为 PartSupp，在 Cardinalities 页选择多对多（Many-many），修改 Cardinality 的值为[1, n]（或者勾选 Mandatory），以表示实体集 Part 和 Supplier 上的所有实体都参与到联系集 PartSupp 中。

Relationship Properties - PartSupp (PartSupp)

Entity 1 Entity 2

Part Supplier

General Cardinalities Notes Rules

Each Part must have one or more Supplier.
Each Supplier must have one or more Part.

Cardinalities

☐ One - one ☐ One - many ☐ Many - one ☒ Many - many

Dominant role: <None>

Part to Supplier

Role name:

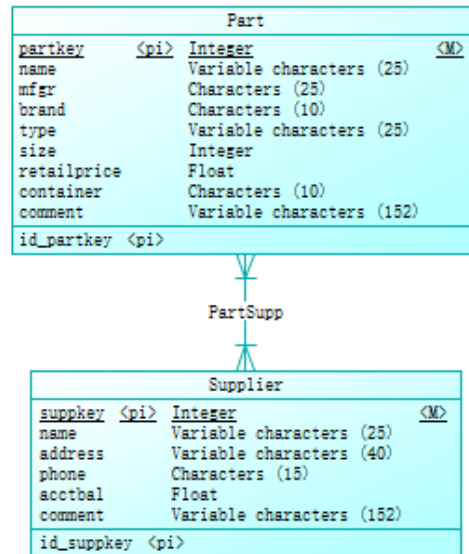
☐ Dependent ☒ Mandatory Cardinality: 1,n

Supplier to Part

Role name:

☐ Dependent ☒ Mandatory Cardinality: 1,n

设置好 PartSupp 后，返回 Diagram 页面，就能看到如下图。Part 和 Supplier 实体集之间的线表示联系集 PartSupp，线两端的三角形表示多（many），横线/竖线表示全参与（部分参与用圆表示）。

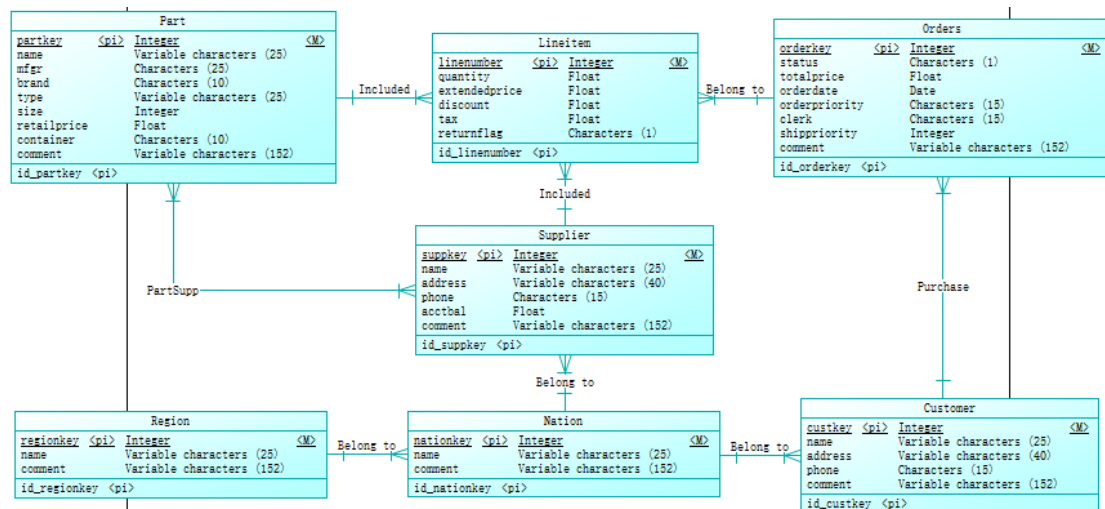


以同样的方法创建其他实体集和联系集，需要注意的是**外键属性不需要添加到实体集上**。例如，教材中 LineItem 实体集具有 partkey 和 supkey 两个外键属性，如果将他们添加到 LineItem 实体集上，一方面会由于重复使用名为 partkey、supkey 这样的 Data Item 而报错，另一方面即使添加成功，它们也不会被当成外键来对待，之后生成逻辑模型时不会在实体集上以外键显示。

Entity Properties - Lineitem (Lineitem)

General Attributes Identifiers Notes Rules										
	Name	Code	Data Type	Length	Preci	M	P	D		
1	linenumber	linenumber	Integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
2	quantity	quantity	Float			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
3	extendedprice	extendedprice	Float			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
4	discount	discount	Float			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
5	tax	tax	Float			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
6	returnflag	returnflag	Characters (1)	1		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

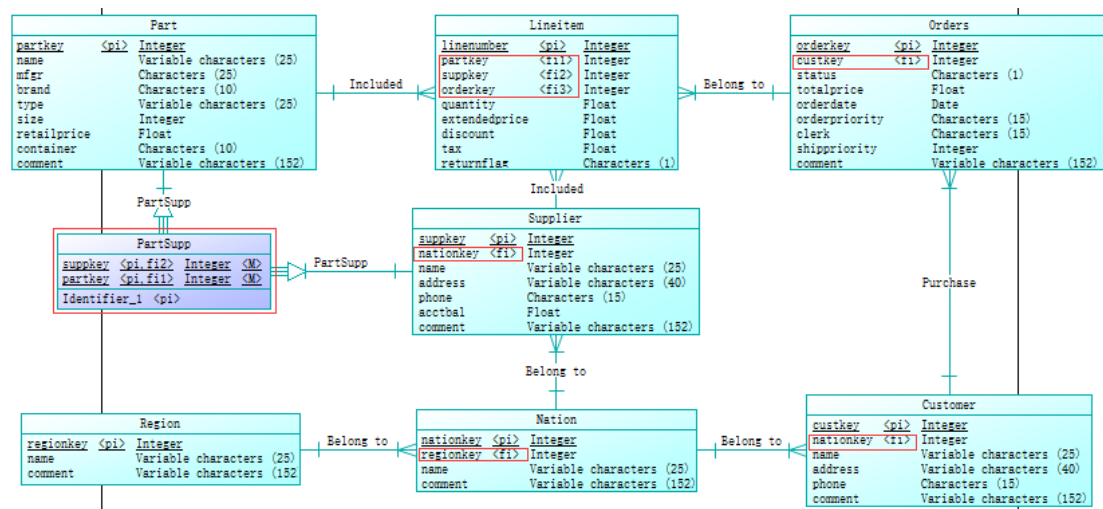
所有实体集和联系集创建完毕，在 Symbol->Auto layout 中调整图片的布局，能得到下图。



2. 逻辑结构设计

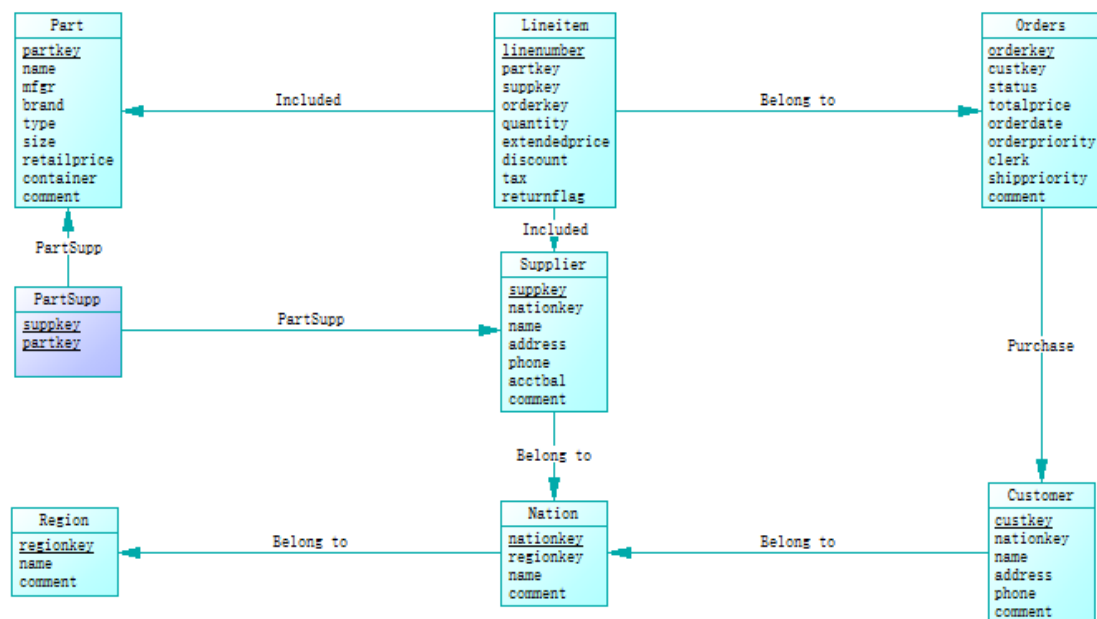
点击 Tools->Generate Logical Data Model, 设置模型名为 Logical Data Model (保证该模型的 code 名与概念模型不同即可), 生成了如下的逻辑模型图。其中, 与在概念模型中不同的是, 多出了一个 PartSupp 实体集, 其他实体集上添加了一些外键属性 (红线框出)。仔细研究, 发现有这样的规律: **如果两个实体集之间存在多对一的联系, “一”对应的实体集会将它的主码添加到“多”对应的实体集上去; 如果是多对多联系集, 二者的主码就会“相遇”而形成一个新的实体集。**这也解释了前面设计概念数据模型时不添加外键属性, 因为外键是自动识别并加上去的。

在逻辑结构图中，实体集的主码以下划线和<pi>标识，外键属性是<fi>（PartSupp 中的两个属性既是主码又是外键）。



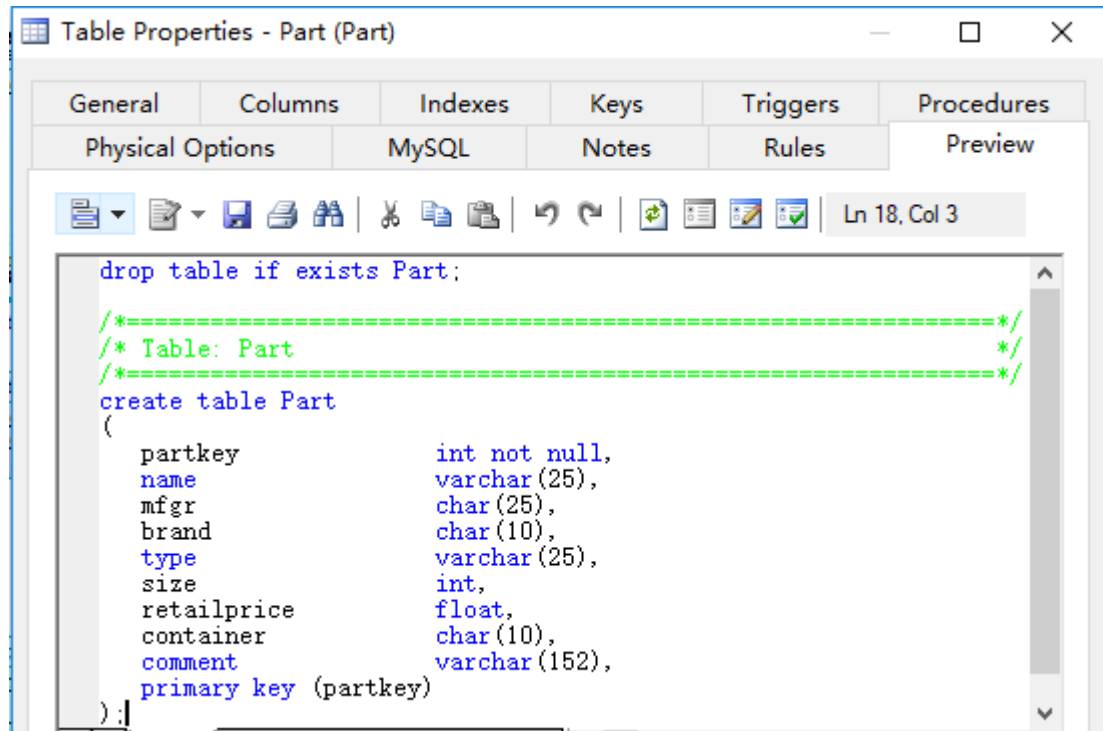
3. 物理结构设计

点击 Tools->Generate Physical Data Model，设置模型名为 Physical Data Model，选择数据库为 MySQL 5.0，生成物理模型如下。主码仍以以下划线标识，联系的箭头从“多”对应的关系指向“一”对应的关系。



4. 数据库模式 SQL 语句生成

生成物理模型后，双击 Part 关系，切换到 Preview 页，就能看到 Part 的对应 SQL 代码，如下图。代码首先判断表格是否存在，如果已有就先删除。主码显式指定为 not null，采用表级约束，这样的好处是便于模板化生成代码，不管主码有一个还是多个属性，都只在表级加上主码约束。



```
drop table if exists Part;

/*=====*/
/* Table: Part */
/*=====*/
create table Part
(
    partkey          int not null,
    name             varchar(25),
    mfg              char(25),
    brand            char(10),
    type             varchar(25),
    size             int,
    retailprice      float,
    container         char(10),
    comment          varchar(152),
    primary key (partkey)
);
```

另一个比较有代表性的关系是 Lineitem，它具有外键属性，创建代码如下图。从代码中可以看到，外键约束是在创建表之后添加的，而不是在列级（实际上对 MySQL 来说是无用的）或表级（猜测是 MySQL 5.0 无法添加表级外键约束，因为在 MySQL 8.0 中表级约束比表外约束的语法更加简单，也具有模板化特性）。

```
drop table if exists Lineitem;

/*=====*/
/* Table: Lineitem */
/*=====*/
create table Lineitem
(
    linenum          int not null,
    partkey           int not null,
    supkey            int not null,
    orderkey          int not null,
    quantity          float,
    extendedprice     float,
    discount           float,
    tax               float,
    returnflag        char(1),
    primary key (linenum)
);

alter table Lineitem add constraint "FK_Belong to2" foreign key (orderkey)
references Orders (orderkey) on delete restrict on update restrict;

alter table Lineitem add constraint FK_Included foreign key (supkey)
references Supplier (supkey) on delete restrict on update restrict;

alter table Lineitem add constraint FK_Included2 foreign key (partkey)
references Part (partkey) on delete restrict on update restrict;
```

五、 实验总结

这次实验与以往的实验有所不同。以往的实验都是已经有了一个设计良好的数据库，所有的数据表（关系）都已经被设计好了，我们实验者的任务是创建这些数据表、在这些数据表上执行查询更新等操作、设置数据表的访问权限等。而在这个实验，我们担负了设计数据表的任务，使用的方法是 ER 建模技术。

ER 建模技术将事物分为实体和联系。实体是那些具体的人、物（如课程），也可以是发生的某件事（如开课、选课）；联系是实体之间的关系（如课程之间的先修关系、老师和学生之间的关系）。每个实体都会有一些属性来表征自己的性质，每种属性上都有对应的值。

配合使用 PowerDesigner 软件，我很快就构建了该实验的概念数据模型，因为 ER 建模技术并不难，而且有很强的可解释性。在正式开始实验之前，还发生了一些小插曲。一开始，我安装了 PowerDesigner Viewer，这是一个只能查看模型而不能创建和修改模型的软件，这让我纠结了好一阵，明明网上的教程点击 File 菜单就能新建模型，而我的软件的新建按钮却是灰色的。后来，继续看其他教程才发现了 Viewer 版和正式版的区别，改装正式版，很快就完成了实验。

我认为，在这个实验中，最重要的技巧是识别出实体集、实体集属性以及联系集。考虑实体集创建少了的情况，例如在大学模式中没有课程这一实体集，而又开课这一实体集，那么一门新课只有在开了之后才能存入数据库中；实体集属性设计不好容易造成冗余，例如在课程和开课中都存储课程名和课程 ID，在这里开课的课程名就是冗余的，实际上只需存储课程 ID，就能在课程中查询到与这个唯一的 ID 对应的课程名；联系集的设计比较考验设计者的能力，一方面需要判断两个实体集之间有没有联系，另一方面要设计是何种联系（一对一、多对一、一对多、多对多）。

以上就是我在这次实验中的收获和感想。