

# 数据库系统开发实验报告

## 售后服务管理系统数据建模

### 1. 实验目的

通过本实验，学习使用数据建模工具，建立企业合同管理数据模型，学会数据库的基本设计方法。

### 2. 实验内容

(1) 使用 powerdesigner16.5 设计数据库逻辑模型。在原数据模型基础上修改数据模型，重点设计完善了售后服务部分，以下是售后服务模块部分表的设计：

售后服务表 Service\_After\_Sale

属性名	属性含义	类型	备注
Service_ID	服务 ID	Int	主键
Service_Begin_Date	服务开始时间	Date	
Service_End_Date	服务结束时间	Date	
Service_Type	服务类型	Char(16)	
Service_Detail	服务内容	Varchar(256)	
Contact_Name	客户联系人姓名	Varchar(64)	
Contact_Phone	客户联系人电话	Char(11)	
Contract_ID	合同号	Contract_ID	外键 1
Parent_Service_ID	上级服务 ID	Parent_Service_ID	外键 2，对于小服务该属性存储大服务 ID，对于大服务该属性为 NULL

该表为售后服务基本信息表，其中服务 ID 作为主键自动生成，服务开始时间、服务结束时间、服务类型、服务内容、客户联系人姓名、客户联系人电话、合同号为基本信息，上级服务 ID 作为引用自身表主键的外键，通过递归联系的方式构建大服务和小服务之间的联系。

售后服务专员表 Employee\_Service

属性名	属性含义	类型	备注
Service_ID	服务 ID	Int	外键 1，联合主键属性之一

Emp_ID	专员员工 ID	Int	外键 2，联合主键属性之一
--------	---------	-----	---------------

由于售后服务和服务员工是多对多联系，所以设计该表以存储各服务的员工信息，外键 1 引用 Service\_After\_Sale 表主键 Service\_ID，外键 2 引用 Emp 表主键 Emp\_ID。

售后服务产品表 Product\_Service

属性名	属性含义	类型	备注
Service_ID	服务 ID	Int	外键 1，联合主键之一
Contract_ID	合同 ID	Int	联合外键 2 属性之一，联合主键之一
Product_ID	产品 ID	Int	联合外键 2 属性之一，联合主键之一

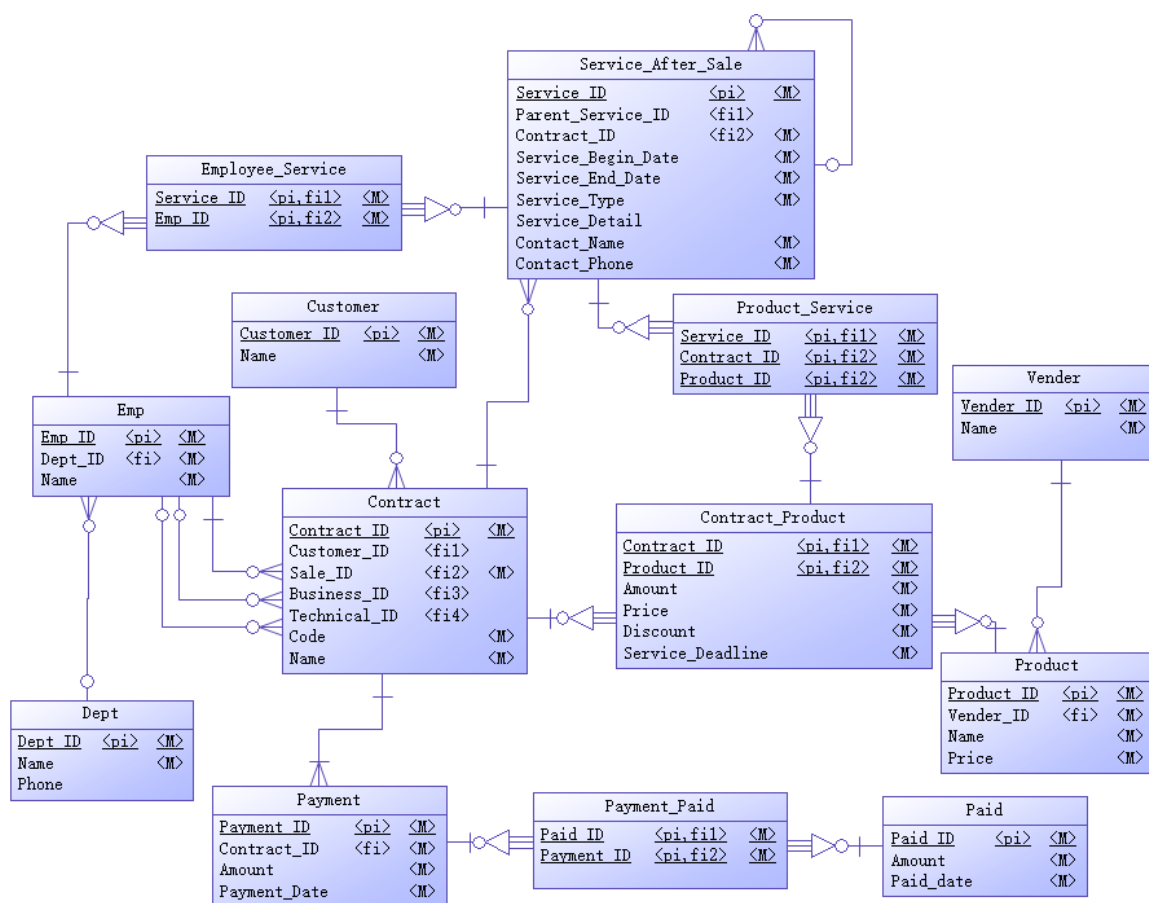
由于售后服务和售后中的产品是多对多联系，所以设计该表以存储各服务的产品信息，外键 1 引用 Service\_After\_Sale 表主键 Service\_ID，外键 2 作为联合外键引用 Contract\_Product 表联合主键 (Contract\_ID, Product\_ID)。

修改合同产品表 Contract\_Product 后如下

属性名	属性含义	类型	备注
Contract_ID	合同 ID	Int	外键 1，联合主键属性之一
Product_ID	产品 ID	Int	外键 2，联合主键属性之一
Amount	产品数量	Int	
Price	单价	Number	
Discount	折扣	Number	
Service_Deadline	产品售后服务期限	Date	

添加 Service\_Deadline 数据项，以记录某次合同中某产品的售后服务期限。可通过 Product\_Service 表用 Service\_ID 获得该次服务的合同 ID 和产品 ID，在 Contract\_Product 表中检测后即可判断该服务中的某项产品是否在服务期限内。其余部分并未作出较大修改。

现展示 ERD 图如下



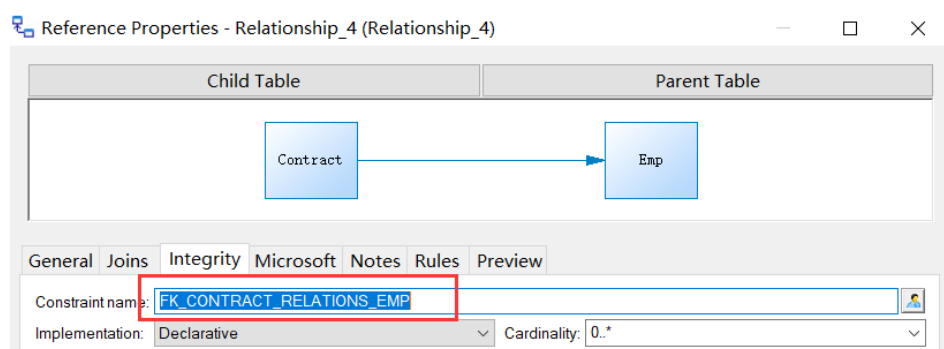
保存逻辑模型名为 LogicalDataModel\_1.ldm

(2) 使用 Powerdesignner 将逻辑模型转化为物理模型，使用的 DBMS 为 Microsoft SQL Server 2012，保存为 PhysicalDataModel\_1.pdm。

使用该物理模型生成 SQL 命令行代码，生成错误信息如下

Diagram_1 Result List			
Category	Check	Object	Location
Package	Constraint name uniqueness	Reference 'Relationship_4'	<Model>
Package	Constraint name uniqueness	Reference 'Relationship_10'	<Model>
Package	Constraint name uniqueness	Reference 'Relationship_11'	<Model>

经检查发现是由于表 Emp 与表 Contract 之间建立三个外键联系，powerdesignner 自动生成的三个对应的约束名称重复，所以报错，自动生成的约束名如下图所示：



手动修改约束名，在其末尾加上数字后缀并保存。

再次生成 SQL 命令行代码，生成成功，保存为 crebas.sql。由于代码过长，故放在附加文件内。

使用系统管理员登录名连接 SQL Server，新建数据库 E6，打开 crebass.sql 文件，可以在查询编辑框内看到具体的代码。切换当前数据库至 E6，执行该代码，E6 数据库即可构建此前设计好的数据库对象。构建完成后，E6 数据库内表如下图所示



如图可见设计表格生成完全，具体表格信息就不一一展示，仅选择 Service\_After\_Sale 表展示。

Service\_After\_Sale 表列属性如下，如图可见 Service\_ID 为主键，各属性均和设计一致：

	列名	数据类型	允许 Null 值
PK	Service_ID	int	<input type="checkbox"/>
	Parent_Service_ID	int	<input checked="" type="checkbox"/>
	Contract_ID	int	<input type="checkbox"/>
	Service_Begin_Date	datetime	<input type="checkbox"/>
	Service_End_Date	datetime	<input type="checkbox"/>
	Service_Type	char(16)	<input type="checkbox"/>
	Service_Detail	varchar(256)	<input checked="" type="checkbox"/>
	Contact_Name	varchar(64)	<input type="checkbox"/>
	Contact_Phone	char(11)	<input type="checkbox"/>
			<input type="checkbox"/>

与该表有关的四个外键约束均正确构建，其具体信息如下四图所示，均与设计一致：

▼ 表和列规范	...
外键基表	Employee_Service
外键列	Service_ID
主/唯一键基表	Service_After_Sale
主/唯一键列	Service_ID
在创建或重新启用时检查现有 是	
▼ 标识	
(名称)	FK_EMPLOYEE_RELATIONS_SERVICE_

▼ 表和列规范	...
外键基表	Product_Service
外键列	Service_ID
主/唯一键基表	Service_After_Sale
主/唯一键列	Service_ID
在创建或重新启用时检查现有 是	
▼ 标识	
(名称)	FK_PRODUCT_RELATIONS_SERVICE_

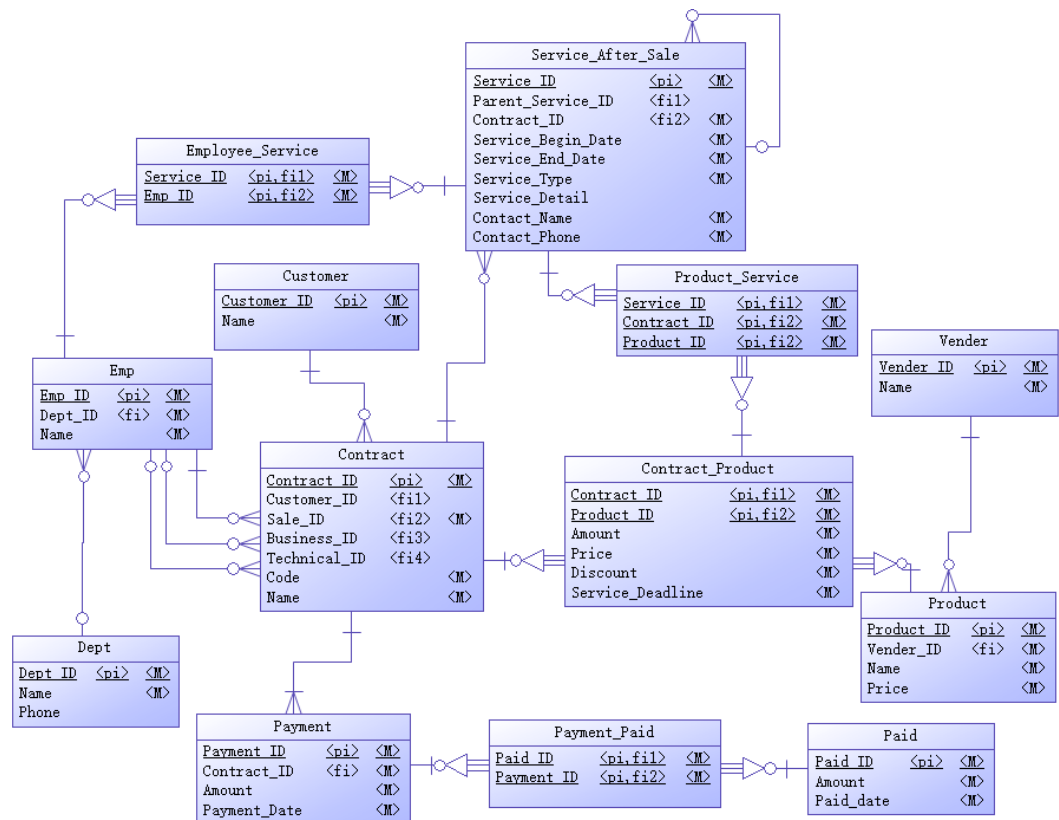
▼ 表和列规范	...
外键基表	Service_After_Sale
外键列	Contract_ID
主/唯一键基表	Contract
主/唯一键列	Contract_ID
在创建或重新启用时检查现有 是	
▼ 标识	
(名称)	FK_SERVICE_RELATIONS_CONTRACT

▼ 表和列规范	...
外键基表	Service_After_Sale
外键列	Parent_Service_ID
主/唯一键基表	Service_After_Sale
主/唯一键列	Service_ID
在创建或重新启用时检查现有 是	
▼ 标识	
(名称)	FK_SERVICE_RELATIONS_SERVICE_
说明	

综上，完成本次数据建模，逻辑模型文件 LogicalDataModel\_1.ldm、物理模型文件 PhysicalDataModel\_1.pdm、SQL 命令行文件 crebas.sql 均保存在附件文件夹内。

### 3. 问题回答

为方便查看，故在此处再次放置 ERD 图



#### (1) 如何实现售后服务中的产品服务期限的？

Contract\_Product 表记录了合同中产品的信息，因为售后服务的对象是某合同中一个或多个产品，其具体期限由合同和合同中的某项产品决定，故可以给 Contract\_Product 表添加 Service\_Deadline 属性，这样该表的一条记录便可以保存在某一合同中某一项产品的服务期限。

#### (2) 如何记录售后服务的产品信息？

一次售后服务可能要处理多项合同中的产品，一个合同中的产品也可能在多个售后服务中被处理，是多对多的联系，故设计 Product\_Service 表记录售后服务中产品信息，其两个外键分别联系 Service\_After\_Sale 表中的 Service\_ID 和 Contract\_Product 表中的 (Contract\_ID, Product)，这样既可以根据服务 ID 确认对应的产品基本信息（合同 ID 和产品 ID），并根据基本信息从 Contract\_Service 表获得对应的产品详细信息（包括服务期限）也可以确保服务的合同中的产品的完整性。

#### (3) 如何记录技术服务工程师（包括服务专员）的信息？

统一记录某次服务中的技术人员服务人员等为服务中的员工，一次售后服务可能有多个员工来进行处理，一个员工也可能处理多个售后服务，是多对多的联系，故设计 Employee\_Service 表来记录售后服务中处理的员工信息，其两个外键分别联系 Service\_After\_Sale 表中的 Service\_ID 和 Emp 表中的 Emp\_ID，这样既可以根

据服务 ID 确认对应的员工基本信息(员工 ID)，并根据基本信息获得从 Emp 表中获得员工详细信息，也可以确保服务员工信息的完整性。

- (4) 一个完整的服务信息（如维修一个磁盘可能包括第一次电话咨询、维修等由多个小服务组成一个大服务）是如何记录的？

大服务和小服务在存储结构上几乎没有差别，仅仅在信息内容上有差异和约束，故记录在同一张表 Service\_After\_Sale 中。写入某次服务的基本信息，将合同号、服务开始时间、服务结束时间、服务类型、服务内容、联系人姓名、联系人电话写入 Service\_After\_Sale 表，并将自动生成的服务 ID 和服务的产品 ID 及对应的合同号写入 Product\_Service 表，将服务 ID 和服务员工 ID 写入 Employee\_Service 表。

为构建表中大服务和小服务之间的联系，在表 Service\_After\_Sale 中设置外键 Parent\_Service\_ID，联系表自身主键 Service\_ID，其含义为本次服务的上级服务的 ID，可以通过这一属性构成一项大服务和其对应的小服务之间的联系。

比如写入一项大服务 A 和其对应的小服务 A1、A2, 先根据 A1、A2 信息写入大服务 A 的基本信息，并设置其 Parent\_Service\_ID 项为 NULL，然后写入小服务 A1，设置其 Parent\_Service\_ID 项为 A 的 Service\_ID，A2 同理。如果大服务 A 仍有后续添加的小服务 A3，那么除了将 A3 的 Parent\_Service\_ID 项设为 A 的 Service\_ID 外，还需要相应修改 A 的信息。

这一设置可以记录完整的服务信息，进一步还可以形成多级服务信息记录，查询操作方便，但更新操作相对困难。