

材料科学数据共享工程标准草案

材料科学数据库设计规范

（征求意见稿）

（本稿完成日期：2010年11月）

2010-11 发布

目 录

前 言	1
1. 范围	2
2. 参考文档.....	2
3. 术语与缩略语.....	2
3.1 术语.....	2
3.1.1 实体 (Entity)	2
3.1.2 属性 (Attribute)	2
3.1.3 值域 (Domain)	3
3.1.4 联系 (Relationship)	3
3.1.5 主键 (Primary Key)	3
3.1.6 外键 (Foreign Key)	3
3.1.7 数据集 (Data Set)	3
3.1.8 数据元 (Data Element)	3
3.1.9. 国家科学数据中心 (Scientific data center)	3
3.1.10 国家科学数据网 (Scientific data network)	4
3.1.11 主体数据库 (Core database)	4
3.1.12 接口 (Interface)	4
3.1.13 数据模式字典 (Data Schema Dictionary)	4
3.1.14 粒度 (Granularity)	4
3.1.15 E-R 数据模型 (E-R Data Model)	4
3.1.16 面向对象数据模型 (Object-oriented Data Model)	4
3.1.17 第一范式 (1NF)	4
3.1.18 第二范式 (2NF)	5
3.1.19 第三范式 (3NF)	5
3.1.20 数据模式 (Data Schema)	5
3.1.21 触发器 (Trigger)	5
3.1.21 存储过程 (Stored Procedure)	5
3.2 缩略语.....	5
3.2.1 UML.....	5
3.2.2 XML.....	5
3.2.3 SQL.....	6
4. 材料科学数据对象命名规范.....	6
4.1 通用规范.....	6

4.2 材料科学数据命名特殊规范.....	6
4.2.1 表名（实体）的命名.....	6
4.2.2 视图命名.....	6
4.2.3 存储过程命名.....	6
4.2.4 函数命名.....	7
4.2.5 触发器命名.....	7
4.2.6 索引命名.....	7
4.2.7 唯一索引命名.....	7
4.2.8 主键命名.....	7
4.2.9 外键命名.....	7
4.2.10 序列命名.....	7
4.2.11 注释规范.....	8
5. 材料科学数据库设计流程.....	8
5.1 数据库设计阶段划分与概述.....	8
5.2 概念设计阶段.....	11
5.2.1 目标.....	11
5.2.2 设计策略.....	11
5.2.3 实施步骤.....	11
5.2.3.1 进行数据抽象.....	12
5.2.3.2 设计局部概念模型.....	13
5.2.3.3 集成局部概念模型，形成全局概念模型.....	13
5.2.3.4 全局 E-R 的优化.....	14
5.2.3.5 材料科学概念模型扩展特性.....	14
5.3 逻辑设计阶段.....	16
5.3.1 目标.....	16
5.3.2 设计要求.....	16
5.3.3 实施步骤.....	16
5.3.3.1 E-R 图向关系数据模型转换.....	18
5.3.3.2 关系数据模型优化.....	18
5.3.3.3 模式规范化.....	18
5.3.3.4 定义完整性和安全性约束.....	18
5.3.3.5 设计用户子模式.....	19
5.3.3.6 性能估计预评价.....	19
5.4 物理设计阶段.....	19

5.4.1 目标.....	19
5.4.2 实施步骤.....	19
5.4.2.1 确定数据库的物理结构.....	20
5.4.2.2 评价物理结构.....	20
5.4.2.3 完整性和安全性考虑.....	20
5.4.2.4 相关程序设计.....	20
5.4.2.5 形成物理设计文档.....	20
6. 材料科学数据库设计原则.....	21
6.1 基本原则.....	21
6.2 材料实体和属性设计原则.....	21
6.2.1 材料实体设计原则.....	21
6.2.2 材料实体属性设置.....	22
6.2.3 材料实体联系定义.....	23
6.3 表的设计.....	23
6.3.1 表的设计原则.....	23
6.3.2 主键和外键选取.....	25
6.4 索引的设计.....	25
6.5 视图的设计.....	25
6.6 数据完整性设计.....	26
7. 运用设计.....	27
7.1 实施步骤.....	27
7.1.1 目标.....	27
7.1.2 实施步骤.....	27
7.1.2.1 定义数据库结构.....	27
7.1.2.2 数据装载.....	28
7.1.2.3 编制与调试应用程序.....	28
7.1.2.4 数据库试运行.....	28
7.2 运行维护.....	28
7.2.1 目标.....	28
7.2.2 人员要求.....	28
7.2.3 工作任务.....	29
7.3 安全设计.....	29
7.4 数据库设计文档规范.....	29
7.4.1 数据库规划报告编写规定.....	29

7.4.2 需求分析文档编写规定	30
7.4.2.1 需求调查表	30
7.4.2.2 数据流图	30
7.4.2.3 数据字典	31
7.4.3 概念设计文档编写规定	32
7.4.4 逻辑设计文档编写规定	32
7.4.5 物理设计文档编写规定	32
附录 A 数据集制作应用实例	33
A.1 数据库命名	33
A.2 定义实体	33
A.3 确定主键	34
A.4 定义域	34
A.5 确定实体属性	34
A.6 确定实体间的联系	38
A.7 选择数据库	40
A.8 E-R 图转换为实际关系数据结构	41
A.9 范式检查与完整性检查	41
A.9.1 范式检查	41
A.9.2 完整性检查	42
A.10 将数据结构转化为数据库表	43

前 言

本标准第一次制定。

本标准主要包括材料科学数据对象命名规范、材料科学数据库设计流程、材料科学数据库设计原则。

本标准是在科学数据共享标准化工作组的数据库标准制定人员的协助下，同材料领域的专家共同制定完成的。

本标准由中国材料科学数据共享工程标准组提出并归口。

本标准主要起草单位：北京科技大学，中科院金属研究所，西北工业大学。

材料科学数据库设计规范

1. 范围

本标准规定了材料科学数据库设计的统一要求。

本标准适用于材料科学领域各类常用数据库的设计、构建和运营维护的全过程。

本标准适用于指导以材料科学领域各类数据资源为主要内容,以共享为主要目的共享数据集规范化的制作、描述、表示、操作方法和步骤,以确保数据集制作人员及数据集用户对共享数据集内容准确、无歧义的理解。

2. 参考文档

下列文件中的条款通过本标准的引用而成为本标准的条款。

SDS/T 2132—2004 数据元标准化原则与方法(科学数据共享技术标准)

SDS/T 2134—2004 数据交换格式设计规则(科学数据共享技术标准)

SDS/T 2321—2004 科学数据中心建设规范(科学数据共享技术标准)

SDS/T 2322—2004 科学数据网建设规范(科学数据共享技术标准)

GB/T 13725-2001 建立术语数据库的一般原则与方法(中华人民共和国国家标准)

GB/T 17532-1998 术语工作 计算机应用 词汇(中华人民共和国国家标准)

GB 17859-1999 计算机信息系统安全保护等级划分准则(中华人民共和国国家标准)

SY/T 6227-2005 石油工业数据库设计规范(中华人民共和国石油天然气行业标准)

Unified Modeling Language 1.5 统一建模语言1.5(OMG)

3. 术语与缩略语

3.1 术语

3.1.1 实体(Entity)

客观存在可以相互区别的事物,也是具有相同性质的一类对象的集合。如金属材料,材料的力学性能。在数据集、关系型数据库中表现为表(Table),而一个具体实例则可以表现为表中的一行(Row)或一条记录(Record)。

3.1.2 属性(Attribute)

描述和表示实体的特征或值。如材料的牌号、用途,材料的屈服点、伸长率、硬度等。在数据集、关系型数据库中表现为列(Column)或字段(Field)。

3.1.3 值域 (Domain)

属性可以取值的范围。值域是单独定义的,用于重用,即多个属性可以使用同一个值域。值域可以是有限的,也可以是无限制的。例如材料的化学成份中, Si 的含量其取值范围应该在 0 到 1 之间。因为对于属性来说,虽然有时可以满足数据的类型要求,但是其取值可能没有任何意义,所以合理的设置材料数据的值域会直接影响到材料数据的质量。

3.1.4 联系 (Relationship)

实体间存在的关联关系的抽象表示。在实体-联系数据模型的二元联系中,把联系区分为一对一,一对多和多对多三种,且在模型中明确地给出这些语义。例如:材料实体与材料的物理特性实体之间存在着一对一的联系;而一个试验试样与其产生的试验数据之间存在着一对多的联系。

3.1.5 主键 (Primary Key)

对取值做出了唯一性限制的、能够区分实体中不同实例的属性。例如材料的牌号。所有实体实例的该属性取值不会出现重复。通过该键值可以唯一的确定一个实体实例。

3.1.6 外键 (Foreign Key)

外键是实体的一个属性,这个属性同时也是与该实体相关联的另一个实体的主键。外键用于提供互见参照 (Cross-reference),也是实现一个关系的约束,它一定是另外一个实体的主键。实体间的联系是数据的很重要的语义,在关系数据模型中,实体之间的联系隐含在它们之间的公共属性中,特别是外键中。这种表示方法不能显示地表示出事物之间的联系。

3.1.7 数据集 (Data Set)

可以标识的数据集合。

3.1.8 数据元 (Data Element)

通过定义、标识、表示和值域等一系列属性描述的一个数据单元。在实体中表现属性,在关系型数据库中表现为列,或字段。

3.1.9. 国家科学数据中心 (Scientific data center)

属于国家科学数据共享平台的组成部分。以国家部门、行业系统为基础,按不同科学技术领域建立的社会公益型的科学数据主中心以及根据需要设立的科学数据分中心,统称国家科学数据中心;主要负责国家长期布局的公益性、基础性科学数据的汇交、管理、交换与共享服务。

3.1.10 国家科学数据网 (Scientific data network)

是国家科学数据共享平台的组成部分。面向国家重大科技计划、重点区域以及基础科学领域，基于因特网连接分布于各科研院所、高等院校和国际组织的相关专业数据库，开展数据组织、加工与服务，所构建的物理上分布、逻辑上统一的科学数据网。

3.1.11 主体数据库 (Core database)

依据国际标准、国家标准或行业标准分类体系构建的二级学科及其分支学科的科学数据集，并基于计算机系统运行的数据库。

3.1.12 接口 (Interface)

是被命名的操作的集合，它们表示一个元素行为的特性。

3.1.13 数据模式字典 (Data Schema Dictionary)

字典形式从名称、定义、英文名称、英文短名、版本标识、状态、来源、注释等多个方面来描述模型中的实体、属性，从而能够严格的对数据模型中的实体和属性作出描述。

3.1.14 粒度 (Granularity)

用于测量数据详细水平。数据综合或概括的程度越高，数据的粒度越低。数据越详细，数据的粒度越高。低粒度的数据常用于整合不同来源的数据，高粒度的数据用于表现数据的细节。

3.1.15 E-R 数据模型 (E-R Data Model)

即实体联系数据模型，其最成功和广泛的应用是作为数据库概念设计的数据模型。该模型不是面向实现，而是面向现实世界。

3.1.16 面向对象数据模型 (Object-oriented Data Model)

一种可扩充的数据模型，它吸收了语义数据模型和知识表示模型的一些基本概念，同时又借鉴了面向对象程序设计语言和抽象数据类型的一些思想。

3.1.17 第一范式 (1NF)

如果一个关系模式中的每个属性的值域皆由不可分的单值数值构成，则称该关系具有第一范式。

3.1.18 第二范式 (2NF)

设 R 是一个具有1NF形式关系模式，如果R中的任何一个非主属性都完全函数依赖于R的任意一个候选关键字，则R具有第二规范形式，简记为2NF。

3.1.19 第三范式 (3NF)

设 R是一个具有2NF形式关系，如果R中的任何一个非主属性对任意一个候选关键字都不存在传递依赖，则R具有第三规范形式，简记为3NF。

3.1.20 数据模式 (Data Schema)

对某一类数据的结构、联系和约束的描述是型的描述，型的描述称为数据模式。它既不涉及数据的物理存储和硬件环境，也与具体的应用无关。模式也称为逻辑模式。

3.1.21 触发器 (Trigger)

触发器 (trigger) 是个特殊的存储过程，它的执行不是由程序调用，也不是手工启动，而是由事件来触发，比如当对一个表进行操作 (insert, delete, update) 时就会激活它执行。触发器经常用于加强数据的完整性约束和业务规则等。

3.1.21 存储过程 (Stored Procedure)

存储过程 (Stored Procedure) 是一组为了完成特定功能的 SQL 语句集，经编译后存储在数据库中。用户通过指定存储过程的名字并给出参数 (如果该存储过程带有参数) 来执行它。存储过程是数据库中的一个重要对象。

3.2 缩略语

3.2.1 UML

统一建模语言， Unified Modeling Language。

3.2.2 XML

可扩展的标记语言， Extensible Markup Language。

3.2.3 SQL

结构化查询语言，Structured Query Language。

4. 材料科学数据对象命名规范

4.1 通用规范

数据库命名时应遵循以下通用的原则：

- 使用通用词汇。命名时应尽量使用材料领域通用词汇。
- 使用英文。要用**简单明了的英文单词，不要用拼音，特别是拼音缩写**。主要目的很明确，让人**容易明白这个对象**是做什么用的。
- 使用**中英文对照**。
- 详尽的注释。尽量写出详尽的、规范的注释。尤其是字面上看不出含义的名称命名时，注释要写的完整、具体。
- 命名**尽量一律大写**。主要是因为有些数据库，表的命名乃至其他数据对象的命名是大小写敏感的，为了避免不必要的麻烦，并且尊重通常的习惯，最好一律用大写。
- 所有命名不得超过**30 字符**。**命名时应尽量使用缩略语**。
- 不得使用数据库的保留字。
- 应该遵循各自的规范。

4.2 材料科学数据命名特殊规范

4.2.1 表名（实体）的命名

T_作为前缀。格式为：T_表名。

其中**表名用英文**。

命名应尽量反映存储的数据内容。数据库表名及表缩写应在数据库设计说明的相关文档中加以描述。

一个表中的各列列名都采用统一的前缀，不设统一规范，列名命名时不宜过长，以能够清楚表达列中内容的含义为原则。

4.2.2 视图命名

视图以**V_**作为前缀。格式为：**V_视图名**。

4.2.3 存储过程命名

存储过程应以**SP_**开头，后续部分主要以动宾形式构成，并用下划线分割各个组成部分。

4.2.4 函数命名

函数命名以 **FUN_为前缀**。格式为：FUN_函数名称。

4.2.5 触发器命名

触发器以 **TR_为前缀**。需要时可增加后缀区分不同触发器。Insert 触发器加_I, Delete 触发器加_D, Update 触发器加_U, 如：TR_触发器名_I 表示 insert 触发器。

4.2.6 索引命名

索引以 **I_为前缀**。格式为：I_表名_字段名。

如果存在多字段索引，取每字段前三个字符加下划线组合，如：在 hyid,hynome,hymobile 上建立联合索引，命名：I_表名_HYI_HYN_HYM,如果前三个截取字符相同，就从字段名称中不同的字符开始取三个字符加下划线组合，如：在 zhyid, zhynome, zhymobile 上建立联合索引，命名：I_表名_ID_NAM_MOB。

4.2.7 唯一索引命名

以 **UI_为前缀**。格式为：UI_表名_字段名。

如果存在多字段唯一索引，取每字段前三个字符加下划线组合，如：在 hyid, hynome, hymobile 上建立唯一索引，命名：UI_表名_HYI_HYN_HYM, 如果前三个截取字符相同，就从字段名称中不同的字符开始取三个字符加下划线组合，如：在 zhyid,zhynome,zhymobile 上建立唯一索引，命名：UI_表名_ID_NAM_MOB。

4.2.8 主键命名

以 **PK_为前缀**。格式为：PK_表名_字段名。

如果存在多字段主键，取每字段前三个字符加下划线组合，如：在 hyid, hynome, hymobile 上建立主键，命名：PK_表名_HYI_HYN_HYM, 如果前三个截取字符相同，就从字段名称中不同的字符开始取三个字符加下划线组合，如：在 zhyid, zhynome, zhymobile 上建立主键，命名：PK_表名_ID_NAM_MOB。

4.2.9 外键命名

外键以 **FK_为前缀**。格式为：FK_表名_主键名_字段名。

4.2.10 序列命名

序列以 **SEQ_为前缀**，格式为：SEQ_表名_列名。

可以不采用表名加列名的形式，可以根据需要另取名字。

4.2.11 注释规范

注释可以包含在批处理中。在触发器、存储过程中包含描述性注释将大大增加文本的可读性和可维护性。本规范建议：

- 注释尽可能详细、全面。

创建每一数据对象前，应具体描述该对象的功能和用途。传入参数的含义应该有所说明。如果取值范围确定，也应该一并说明。取值有特定含义的变量（如 `boolean` 类型变量），应给出每个值的含义。

- 注释语法包含两种情况：单行注释、多行注释。

单行注释时注释前有两个连字符（--），最后以行尾序列（CR-LF）结束。一般，对变量、条件子句可以采用该类注释；多行注释以符号`/*`和`*/`之间的内容为注释内容。对某项完整的操作建议使用该类注释。

- 注释简洁，同时应描述清晰。

- 函数注释：

编写函数文本如触发器、存储过程以及其他数据对象时，必须为每个函数增加适当注释。该注释以多行注释为主，主要结构如下：

```
/*  
*****  
*name      :          --函数名  
*function  :          --函数功能  
*input     :          --输入参数  
*output    :          --输出参数  
*author    :          --作者  
*CreateDate :        --创建时间  
*UpdateDate :        --函数更改信息（包括作者、时间、更改内容等）  
*****  
*/  
  
CREATE PROCEDURE SP_XXX
```

5. 材料科学数据库设计流程

5.1 数据库设计阶段划分与概述

结合软件工程生命周期的概念，本规范将数据库系统的设计步骤分为如下七个阶段，如图 5.1 所示：

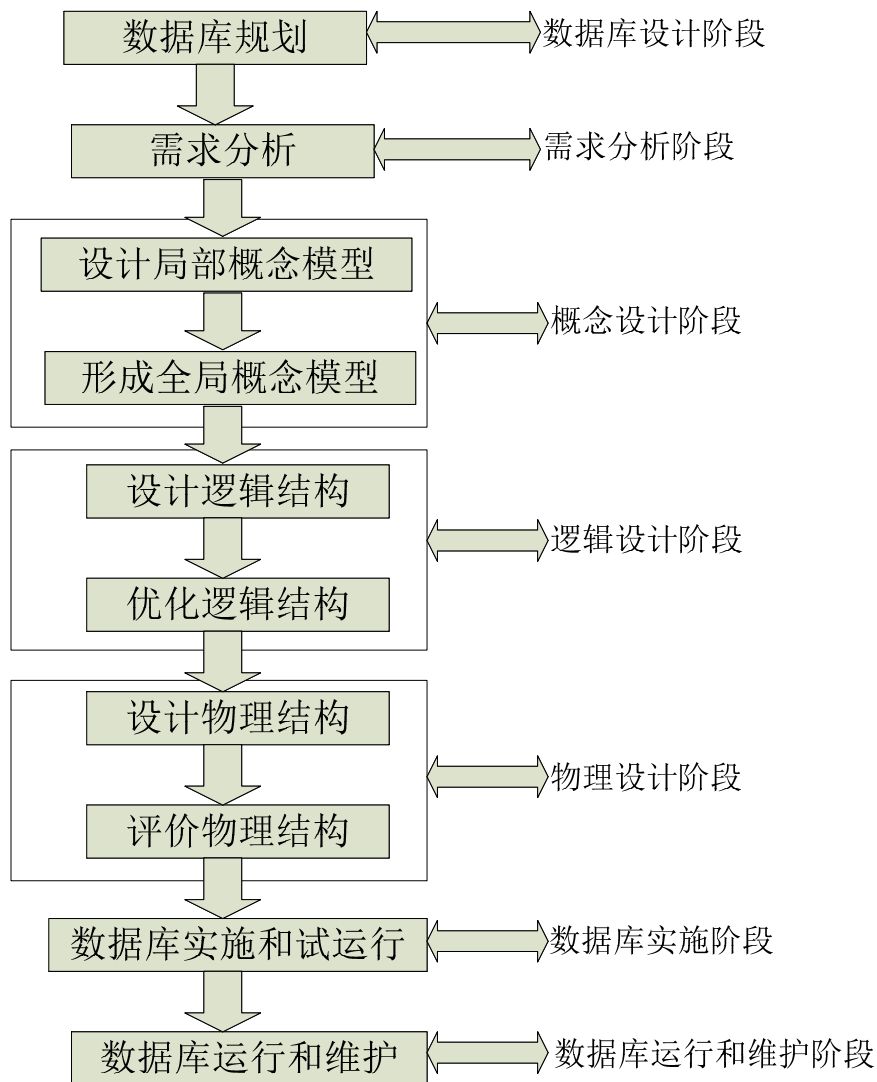


图 5.1 数据库设计过程

- 数据库规划阶段：主要进行的工作是材料科学数据库可行性分析及目标和范围。
- 需求分析阶段：数据库设计首先必须准确了解与分析共享网用户需求(包括数据和处理)。需求分析是整个设计过程的基础，是最困难、最耗费时间的一步，需求分析的好坏直接影响整个数据库的建设速度与质量。
- 概念设计阶段：概念结构设计是整个数据库设计的关键，它通过对共享网用户需求进行综合、归纳与抽象，形成一个独立于具体 DBMS 的概念模型。
- 逻辑设计阶段：逻辑结构设计是将概念结构转换为某个 DBMS 所支持的数据模型，并对其进行优化。
- 物理设计阶段：物理设计是为逻辑数据模型选取一个最适合应用环境的物理结构(包括存储结构和存取方法)。
- 数据库实施阶段：在本阶段，设计人员运用 DBMS 提供的数据库语言及其宿主语言，根据逻辑设计和物理设计的结果建立数据库，编制与调试应用程序，组织数据入库，并进行试运行。
- 数据库运行维护阶段：数据库正式投入运行后，要不断地对其进行评价、调整 and 修改。

上述的数据库系统的设计过程既是数据库设计的过程，也是数据库应用系统的设计过程。在整个设计过程中，数据库的设计和对数据库中数据处理的设计需要紧密结合起来，相互参照，相互补充，从而达到相互完善的目的。

数据库系统设计的七个阶段是不循环反复的过程，经过过程迭代和逐步求精后，逐步形成数据库的各级模式。整个数据库系统设计阶段过程中的输入、输出如图 5.3 所示。

数据库设计过程的输入包括四方面的内容：

- 总体信息需求：与材料领域专家和用户进行深入的沟通，构建材料科学数据库的可行性分析；
- 需求描述，数据流图，数据字典；
- 数据库概念结构，材料概念模型；
- 数据库逻辑结构，材料逻辑模型。

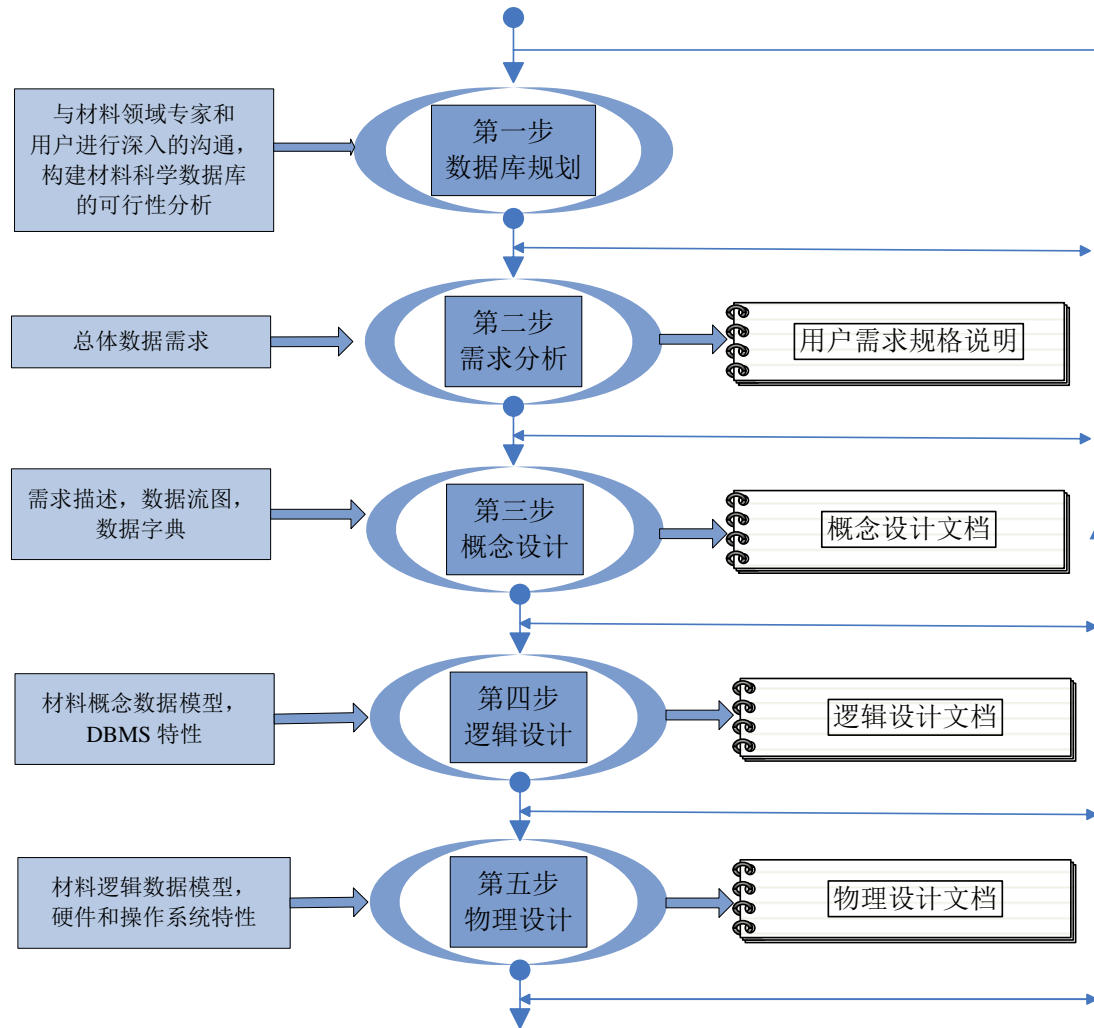


图 5.3 数据库设计的输入输出

在数据库设计阶段之前，最重要的步骤是数据库规划和数据库需求分析。

数据库规划阶段的主要任务是对数据库系统建设的各种因素做全面的分析和权衡，保障系统建设的正确方向。包括：

- 确定系统的名称、范围、目标、功能；
- 确定系统所需要的资源(如人员、设备、资金等)；
- 估计系统开发的成本；
- 确定系统实施计划及进度；
- 分析估算系统可能达到的效益；
- 确定系统设计的原则和技术路线等。

通过了解共享网用户的实际需求，结合材料技术发展的历史与前景，最终确定系统的需

求，为后面数据库的设计奠定坚实的基础，为数据服务提供可靠的依据。

5.2 概念设计阶段

概念结构独立于数据库的逻辑结构，也独立于支持数据库的 DBMS。它是现实世界和机器世界的中介，是现实世界的一个真实模型，易于理解，便于和不熟悉计算机的用户讨论交流意见，并能很容易地进行调整来适应现实世界的需求改变。概念结构设计是整个数据库设计的关键所在。

5.2.1 目标

以需求分析阶段的成果（主要是数据流图和数据字典）为基础，对现实世界进行抽象，借助实体联系模型（E-R 图）或面向对象模型来描述现实应用领域的信息结构，建立一个既可为最终用户所理解，又可在多种数据库管理系统上实现的概念模型。

5.2.2 设计策略

概念结构设计通常包含四类设计策略：

- 自顶向下：首先定义全局概念结构的框架，然后逐步细化。比如先将材料按照某种特性分类，然后细化每个类。
- 自底向上：首先定义各具体材料的概念结构，然后将它们集成起来，得到全局概念结构。
- 混合策略：自顶向下和自底向上的策略相结合，用自顶向下策略设计一个全局概念结构框架，并以此为骨架集成自底向上策略中设计的各局部概念结构。

5.2.3 实施步骤

最普遍的自底向上的设计策略，即自顶向下地进行需求分析，然后自底向上地进行概念设计，并采用 E-R 工具来描述概念模型。图 5.4 给出了概念设计的主要的三个步骤：数据抽象、局部概念模型、局部概念模型的集成。

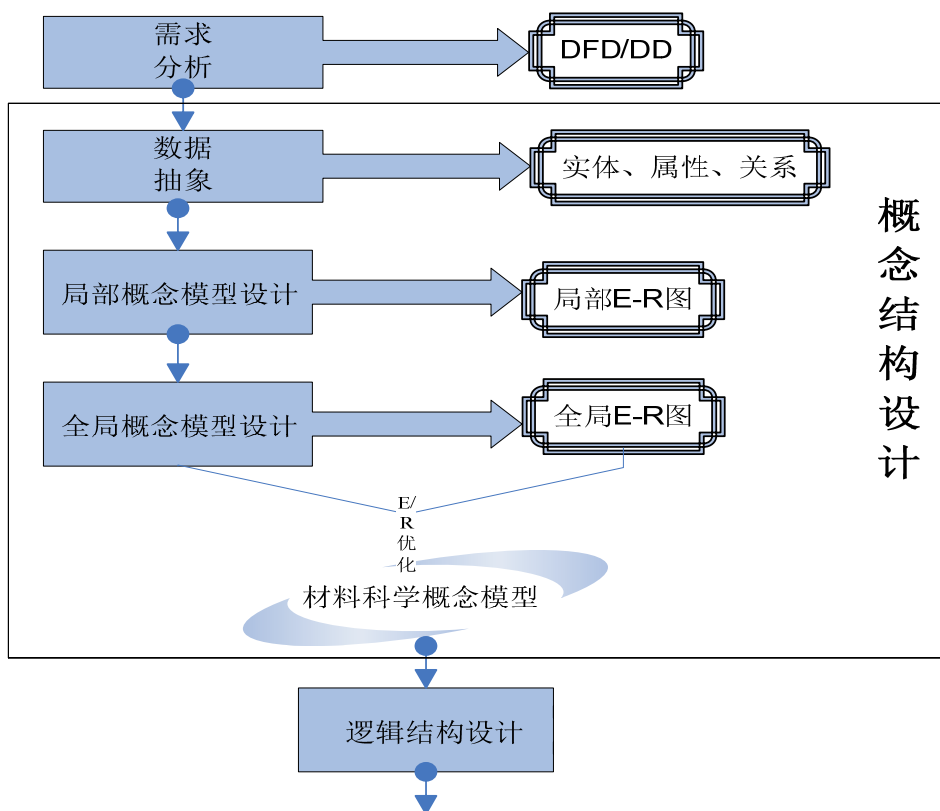


图 5.4 概念结构设计

- 进行数据抽象，确定实体、属性和关系；
- 设计局部概念模型，形成 E-R 图；
- 集成局部概念模型，形成全局 E-R 图；
- 进行全局 E-R 的优化；
- 形成概念设计文档。

5.2.3.1 进行数据抽象

概念设计的第一步就是利用抽象，对需求阶段收集到的数据进行分类、组织和聚集，形成实体、属性和标识，然后确定实体之间的联系。

数据抽象分为三类：

- **分类(classification)**: 对现实世界进行分类，即定义某一组对象的类型，这些对象具有某些共同的特性和行为。它抽象了对象值和型之间的“is member of”的语义。在 E-R 模型中，通过分类抽象形成了实体。例如，钢铁材料是对铁、碳钢、合金钢抽象形成的实体。
- **聚集(aggregation)**: 定义某一类型的组成成分。它抽象了对象内部类型和成分之间的“is part of”语义。在 E-R 模型中，若干属性的聚集组成了实体型。
- **概括(generalization)**: 定义类型之间的一种子集联系，它抽象了类型之间的“is subset of”的语义。例如，钢铁材料是一个实体型，铁和碳钢也是实体型，它们都是钢铁材料实体型的子类实体，并称钢铁材料为超类。概括的一个很重要特性是继承性，子类继承超类的所有特性，还可以增加自己的特性。

5.2.3.2 设计局部概念模型

设计局部概念模型分为以下几步：

- 选择局部应用：在数据流图中选择适当层次的子数据流程图，作为设计局部概念模型的出发点；
- 确定实体：一般按照自然习惯来划分确定实体，如铁、材料力学性能和制备工艺等，这些都是自然存在的实体；
- 确定属性：分析子数据流程图，将属性合理地分配给实体和联系，确定实体的关键字属性；
- 确定联系：根据子数据流程图和相应的数据字典，考虑实体之间是否存在联系，并确定是 1:1, 1:n 或 m:n 的联系，同时消除冗余的联系；
- 确定继承：分析实体间的分类关系与包含关系；
- 确定嵌套：分析实体对属性的依赖关系，确认是否存在聚合与分解关系；
- 消除属性冗余：对于在多个实体中出现的属性，应将它分配给其中的一个实体，以避免数据冗余，影响数据的一致性和完整性。

在设计局部概念模型时，要特别注意区分实体和属性，能够作为属性对待的，应尽量作为属性对待，目的在于简化 E-R 图。区分实体和属性可以遵循以下三条基本准则：

- 描述性原则：一般实体需要有进一步的性质描述，而属性则无，属性一般不能再具有需要描述的性质，即属性不能是另一些属性的聚集；
- 依赖性原则：一般属性仅单向依赖于某个实体，且这种依赖是包含性依赖；
- 一致性原则：一实体有若干个属性组成，这些属性间有内在的关联性与一致性。

5.2.3.3 集成局部概念模型，形成全局概念模型

从全局的观点出发，进行局部概念模型的综合和归并，并消除不一致和冗余，形成一个完整的、能支持各个局部概念模型的全局概念模型。

集成局部概念模型可以分步完成，先归并联系较紧密的两个或多个局部模型，形成中间局部模型，再将中间局部模型归并，最终形成全局概念模型。

局部模型归并主要采用三种方法：

- 等同(identity)：指两个或多个数据对象有相同的语义，包括属性等同、实体等同和联系等同；
 - 聚合(aggregation)：也称为聚集，指将不同实体聚合成合成对象；
 - 泛化(generalization)：指将相似对象提取成一个新实体，并构成具有继承关系的结构。
- 局部模型归并分为实体类的合并和联系的合并。实体类合并用于重点解决以下问题：
- 命名冲突：包括实体类型名、联系类型名之间同名异义或异名同义等命名冲突；
- 标识符冲突：同一实体类的标识符应一致；
 - 属性冲突：包括属性域冲突、取值单位冲突、取值范围冲突等；
 - 约束冲突：不同局部模型存在不同的约束；
 - 结构冲突：如既作为实体又作为联系或属性，同一实体其属性成分不同等。
- 联系的合并用于消除合并后的冗余联系和属性。

5.2.3.4 全局 E-R 的优化

一个好的全局 E-R 图除了能反映用户功能需求外，还应满足下列条件：

- 实体类型个数尽可能少；
- 实体类型所含属性尽可能少；
- 实体类型间联系无冗余。

全局 E-R 的优化就是要达到上述三个目的，为此，要合并相关实体类型，一般把 1:1 关系的两个实体类型合并，合并具有相同键的实体类型，消除冗余属性，消除冗余联系。但有时为提高效率，根据具体情况可存在适当冗余。

5.2.3.5 材料科学概念模型扩展特性

数据间关联复杂是材料数据不同于其它数据的显著特点之一。一方面，材料数据不是孤立的，数据的意义与试验场景、环境、试件的尺寸等密切相关，即各种材料科学数据都是与特定的上下文相关的，例如陶瓷材料的加工和生产是一个复杂的动态过程，也就是说仅仅由陶瓷原材料的基本属性根本无法得到陶瓷产品的性能，只有综合考虑陶瓷原材料的基本属性和生产加工的特性，才能得出陶瓷产品的性能。另一方面，材料数据源单位的数据库能够满足一定范围的应用需求，但是由于这些数据库都是根据单一材料、单一应用或单一行业的特点构建的，因此孤立、无序、不可控，不能全面反映材料科学数据的整体情况。

面对这些问题，传统的 E-R 建模技术不能很好的满足需求，而需要使用扩展的实体联系（EE-R）建模技术或面向对象的数据建模技术，增加描述客观复杂数据对象的能力，同时综合考虑材料学科特色和应用特点，构建统一的数据模型。

面向对象的数据模型是用面向对象观点来描述现实世界（对象）的逻辑组织、对象间的限制、联系等的模型。一系列面向对象的核心概念构成了面向对象数据模型的基础。面向对象的数据模型是可扩充的数据模型，能方便自然的模拟客观世界；对象内部数据及方法实现方式的变化不影响对象外界的使用，提高了数据的独立性；外界不能直接访问修改对象内部数据，有利于保证数据的完整性和安全性。为此，我们可以采用语义描述能力更强的面向对象数据模型。

在面向对象模型中可以利用继承、聚合和关联表达实体对象之间的联系。

继承联系，表示若干数据实体继承自某个数据实体，并因此具有该实体的部分属性，例如，铝合金是一种有色金属，所以铝合金实体继承自有色金属（如图 5.5 所示）。

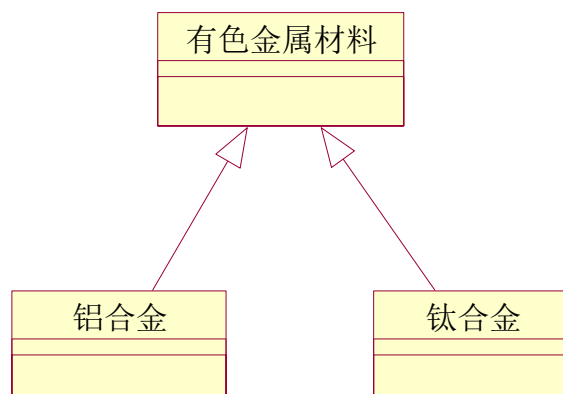


图 5.5 继承

聚合联系，表示实体（数据单元）包含其它的实体（数据单元），被包含的实体（数据单元）是前者的一个组成部分。例如，材料科学数据包括材料的组成结构、材料的加工与制备、材料的基础性能以及材料的使用效能等部分组成（如图 5.6 所示）。

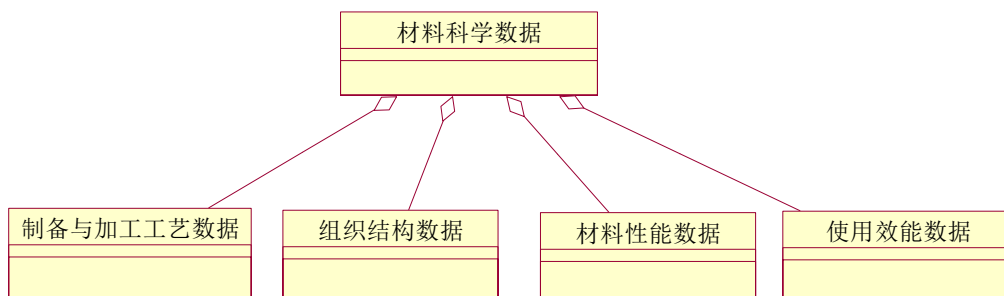


图 5.6 聚合

关联联系，一般用于表示除泛化、包含等联系以外的数据实体之间的关系，例如，一个试验所用的试件可能作一次获多次试验，每次试验可以出一个试验结果，就可以利用关联表达（如图 5.7 所示）。

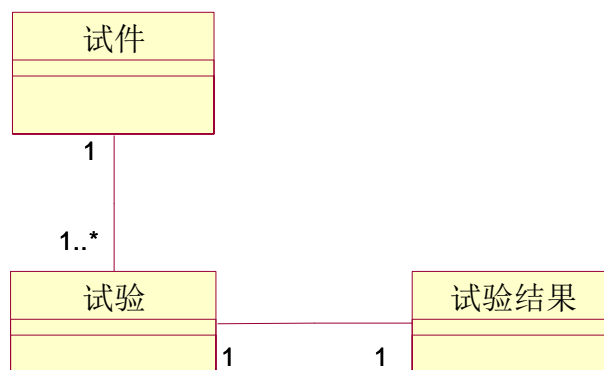


图 5.7 关联

可以将材料数据模型设计为一个面向对象模型，它由超类、类、对象、对象之间的关联、约束等几部分组成。从应用的需求出发，抽象出若干个超类，如事物对象、活动、特性等。事物对象就是指我们感兴趣的一切对象，如某种材料、等等，而活动是对对象进行的操作，如对某种材料的加工、试验等等。当然对同一个事物对象，可以在不同时期、不同环境下进行多次“活动”。特性是对事物对象属性以及活动产生结果的描述。每个超类可以包含若干个子类，可以按实际应用需求划分。在这些超类的基础上，可以进一步衍生出多个类和约束，根据业务需求，形成完整的面向对象模型，它描述了数据之间的逻辑关系，使任何一个数据都有了背景意义，从而解决复杂的数据关联问题。

例如，当我们描述一个工程材料从诞生、到加工、到服役过程、到失效、到消亡的生命周期全过程时，可以利用该模型表示出各种材料数据实体之间的联系，一个金属材料具备重量、尺寸、颜色等金属共有的特性信息，可以将其视为我们感兴趣的对象，通过特定的生产活动，将该金属材料生产为 347 不锈钢，就可以得到不锈钢型号、成分等特性信息；接着，

该 347 不锈钢通过加工活动被加工为一个 347 不锈钢海水泵止回阀,它所具备的各种设计参数、描述信息等各种特性信息可以被记录下来;该止回阀进入服役环境进行服役后,可以得到其服役过程中的载荷、运行参数等信息;在服役的过程中,要对该 347 不锈钢海水泵止回阀进行多次检修活动,并产生检修记录信息;当该止回阀在服役过程中发生失效,会产生与其失效相关的描述和参数信息;为了能够改善产品的性能,需要对该止回阀进行失效分析,通过断口显微分析、海水腐蚀分析、磨损分析分别得到其疲劳状况、腐蚀状况和磨损状况,综合这些状况信息可以得出该止回阀的失效结论,并将该失效案例的相关数据保存起来;最后,失效后的 347 不锈钢海水泵止回阀通过回收回归成为原材料。在这个过程中,此金属对象与各种特性对象、止回阀对象、服役对象、检修记录对象、失效情况对象、各种分析数据对象等之间可以利用关联关系进行表示。虽然在对象之间都是关联,但是不同对象之间的关联蕴含着不同语义,利用面向对象模型可以更好的表达这种语义关系。

对于复杂语义的表达,可以利用本体对 E-R 模型进行扩展。本体是面向领域的概念模型,它包含领域内通用的、普遍的概念,并规定了领域级别上的约束,这些约束可以被用来进行逻辑推理,因此本体是可以表达更高级别的信息需求。可以通过本体显式地、形式化地表达材料领域的概念模型,捕捉同一领域内 ER 模型间的共性;并且通过把 ER 模型和本体结合起来,将本体作为 ER 模型的语义基础,使 ER 模型中实体、属性和联系中隐含的语义显式的表达出来,方便 ER 模型间的集成。例如,材料的分类情况十分复杂,按照材料的组成、应用等角度可以把材料分成不同的类别,同一种材料可能分别属于不同的类别,表达这些类别之间的复杂的语义关系是一个困难的问题。可以利用本体中的类来描述材料领域的类别,并利用描述逻辑公理定义类之间的联系,从而可以表达材料领域中的复杂语义知识。

5.3 逻辑设计阶段

5.3.1 目标

逻辑结构设计是将概念结构模型转换成材料科学数据共享网数据库系统可支持的逻辑数据模型。

5.3.2 设计要求

逻辑结构设计产生的逻辑数据模型应符合下列要求:

- 选用的数据库管理系统(DBMS)完全支持;
- 模式具有一定的规范化,尽可能满足第三规范形式(3NF)的要求;
- 满足用户的完整性和安全性要求;
- 支持用户高效率的数据操作要求;
- 具有较高的存储空间利用率。

5.3.3 实施步骤

图 5.8 给出了逻辑设计的主要三个步骤:转化为一般的数据模型、转化为特定 DBMS 支持的数据模型、优化模型。

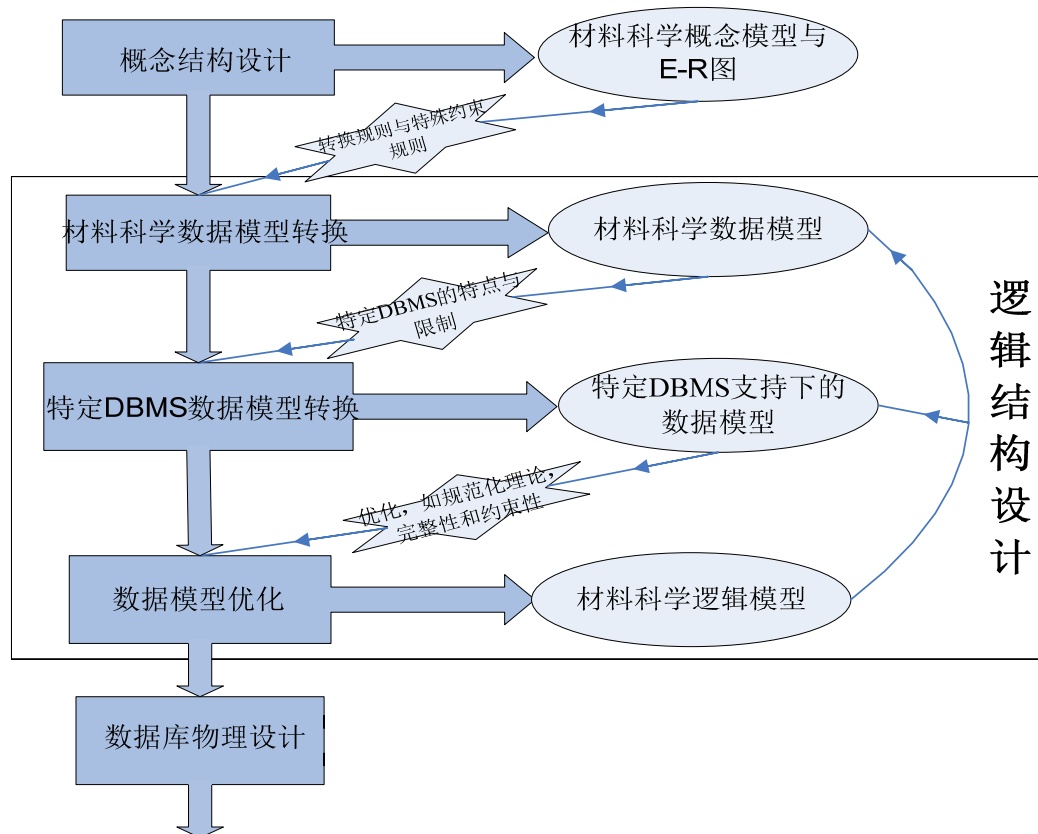


图 5.8 逻辑结构设计

- 将概念模型转换为一般的关系数据模型；
- 将转化来的关系模型向特定的 DBMS 支持下的关系数据模型转换；
- 对关系数据模型进行优化；
- 模式规范化；
- 设计用户子模式；
- 定义完整性和安全性约束；
- 性能估计与评价；
- 形成逻辑设计文档。

注：逻辑设计如果采用基于 UML 的面向对象的方法，通过下列转换规则将面向对象模型转换成关系数据模型：

- 将对象类映射为一个以上的关系模式；
- 将关系(一对一、一对多、多对多等)映射为一个关系模式或其关联的对象类模式之间的外键；
- 将单一继承的超类和子类分别映射为一个关系模式，或是只映射为一个关系模式(父类/子类)，让其拥有父类和子类全部的属性；
- 将多重继承的超类和子类分别映射为一个关系模式；
- 对映射后的关系模式进行冗余控制调整，使其达到合理的关系范式。

5.3.3.1 E-R 图向关系数据模型转换

将 E-R 图转换成关系数据模型实际上就是将实体、实体的属性和实体间的联系转换成关系模式。这种转换一般遵循如下规则：

- 一个实体型转换为一个关系模式。实体的属性就是关系的属性，实体的码就是关系的码；
- 一个联系可以转换为一个关系模式，与该联系相关的各实体码及属性转换为关系的属性，该关系的码有三种情况：
 - ◆ 1:1 联系：即每个实体的码均是该关系的候选码；
 - ◆ 1:n 联系：即关系的码为 n 端的实体码；
 - ◆ m:n 联系：即关系的码为诸实体的码的组合。
- 扩充的继承和嵌套联系的转换；
- 具有相同码的关系模式可合并。

5.3.3.2 关系数据模型优化

对关系数据模型的优化是指适当地修改和调整数据模型的结构，提高数据库应用系统的性能。

对数据模型的修改方法有三种，即垂直分割、水平分割和适当增加冗余，三种方法可以混合使用。

- 垂直分割是把关系模式 R 的属性分解为若干个子集合，形成若干个关系模式。垂直分割的原则是，经常在一起使用的属性从 R 中分解出来形成一个子关系模式。垂直分割可以提高某些事务的效率。
- 水平分割是把关系的元组分为若干子集合，定义每个子集合为一个关系，以提高系统的效率。

5.3.3.3 模式规范化

对生成的关系模型进行规范化，以期进一步优化模型，消除异常，改善完整性、一致性和存储效率，应力求达到第三规范形式(3NF)。

规范化过程实际上就是单一化的过程，即让一个关系描述一个概念，若多余一个概念就把它分离出来。

具体方法包括：

- 确定关系模式的动态或静态性；
- 确定静态关系模式的规范化级别，确保为第一规范形式(1NF)；
- 将动态关系模式转换为第三规范形式(3NF)。

5.3.3.4 定义完整性和安全性约束

定义关系模式上的约束包括：

- 定义属性上的完整性约束；
- 定义属性间的完整性约束；
- 定义属性上的安全性约束；
- 定义属性间的安全性约束。

5.3.3.5 设计用户子模式

用户子模式是应用程序与数据库的接口，允许有效访问数据库而不破坏数据库的安全性。应根据局部应用的需求，结合具体 DBMS 的特点，设计用户的外模式。步骤包括：

- 根据实际局部需求和使用情况进行应用划分；
- 根据具体应用确定子模式的信息结构。

一般利用 DBMS 提供的视图功能来完成用户子模式的设计工作。

5.3.3.6 性能估计预评价

对关系模式进行性能估计与评价的目的是检查关系模式是否满足用户的要求。一般采用定量分析和性能测算的方法对逻辑关系模式作出评价。

- 定量分析是指处理频率和数据容量及其增长情况；
- 性能测算是指逻辑记录访问的数目，一个应用程序传输的总字节数和数据库的总字节数等。

5.4 物理设计阶段

5.4.1 目标

在逻辑数据模型的基础上，结合特定的数据库管理系统和计算机软硬件环境，确定数据库的物理结构，包括数据的存储结构和存取方法。

5.4.2 实施步骤

数据库的物理设计阶段通常分为两个重要部分确定数据库的物理结构和评价数据库的物理结构，如图 5.9 所示：

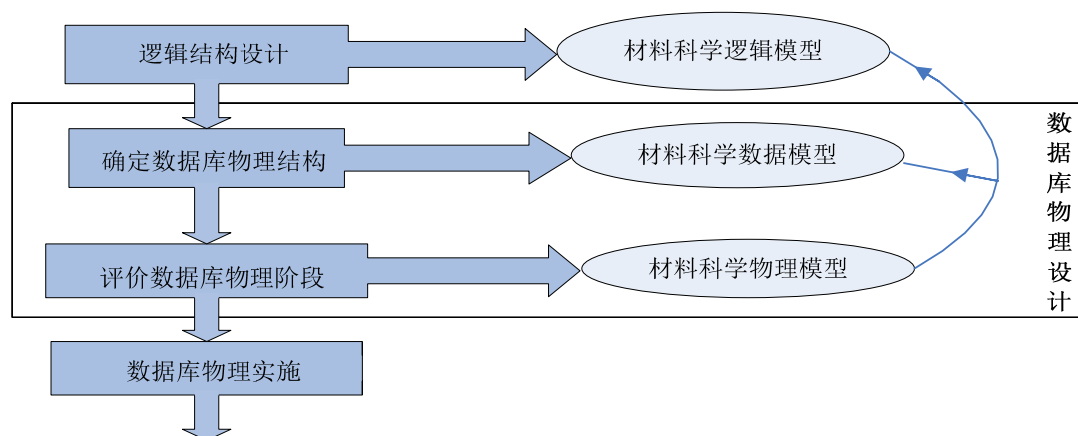


图 5.9 数据库的物理设计

具体的实施步骤包括：

- 确定数据库的物理结构；
- 对物理结构进行评价，重点是时间和空间效率；
- 安全性和完整性考虑；
- 程序设计；
- 形成物理设计文档。

5.4.2.1 确定数据库的物理结构

结合给定的计算机系统，在充分了解所用 DBMS 的内部特征后，确定数据库的以下几方面内容：

确定数据的存储结构：确定存储结构时，要综合考虑存取时间、存储利用率和维护代价三方面的因素。确定存储结构一般包含索引设计、聚簇设计和分区设计等三方面：

- 确定数据的存放路径：主要是确定建立哪些索引来提高访问效率；
- 确定数据的存放位置：分析数据的稳定性，分盘或分区存储数据，以加快存取速度；
- 确定系统配置：按照 DBMS 提供的配置参数，结合数据情况，对包括数据库大小、使用缓冲区个数与大小、时间片大小、锁数目、用户数等参数进行合理配置，以使系统性能最优。

5.4.2.2 评价物理结构

从时间效率、安全性、完整性、有效性、维护代价和其他各种用户要求等方面对数据库物理结构进行权衡，以便对设计方案进行细致的评价，优化数据库的物理结构的设计。

评价数据库物理结构的方法依赖于所选用的 DBMS，主要是从定量估算各种方案的存取空间、存取时间和维护代价入手，对估算结果进行权衡、比较，选择出一个较优的、合理的物理结构。

5.4.2.3 完整性和安全性考虑

根据逻辑设计说明书中提供的对数据库的约束条件、具体的 DBMS，OS 的性能特征和硬件环境，设计数据库的完整性和安全措施。

5.4.2.4 相关程序设计

包括人机接口设计，如菜单、屏幕设计、I/O 格式设计、代码设计、处理加工设计等。

5.4.2.5 形成物理设计文档

物理设计的结构文档是物理设计说明书。

6. 材料科学数据库设计原则

6.1 基本原则

结合共享网的实际情况，在该类型数据库设计应遵循如下基本原则：

- 数据的共享性。数据的共享是建立数据库的最基本动因，数据库建设要强调多种应用、各个业务单位共建共用，共享数据服务。
- 数据的整体观念。数据库存储、管理和操作的对象是数据，和设计其他系统一样，必须具有整体的观念。
- 数据库设计应与应用系统设计相结合。数据最终总是通过应用系统来满足用户的需求，在设计数据库结构时，应充分考虑应用系统的要求。
- 结构特性和行为特性密切结合。要充分了解对数据的处理和使用两个层面的特性，在整个设计过程中要把结构（数据）设计和行为（处理）设计紧密结合起来，同时考虑数据及其处理，便于系统达到整体最优。
- 数据库设计应面向数据，而不应面向处理过程。只要采集的数据地点性质和目标不变，其数据类相对是稳定的，而处理过程是经常变化的。特定数据库的共享数据的提交和发布，都是对数据类的数据输入和数据输出。
- 设计“主体数据库”，而不是“应用数据库”。“主体数据库”是共享的、一致的、本来意义的数据库，它面向业务主题，而不是面向应用程序，所以数据独立于程序，数据本身基本稳定，不会随应用系统的变化而改变。“主体数据库”强调分析企业各业务层次上的数据源，要求数据从源头就地采集、处理、使用和存储，以及必要的电子传输、汇总，必须保证数据一次一处录入，杜绝数据多次录入，造成数据重复。“主体数据库”应当由多个“基本表”组成，“基本表”具有原子性、演绎性和规范性。
- 关系模型规范化。设计完成的关系模型的好坏直接影响到数据库的使用性能。对于关系模型而言，在设计过程中，要尽量满足第一规范形式(1NF)、第二规范形式(2NF)和第三规范形式(3NF)的要求。

6.2 材料实体和属性设计原则

6.2.1 材料实体设计原则

如前所述，实体是现实世界的事物或现象的抽象表示。具体到材料科学数据共享网，它应既能够表示实际材料，也能够表示抽象的材料性能的概念。同一材料实体的不同具体材料具备相同的属性。分析和确定实体是建立数据模式的第一步工作，然后确认实体的定义和命名。

实体的定义包括三部分内容：实体命名、实体定义、实体别名。对于确认后的实体信息，分别记录归档。

1) 实体命名

给予实体一个统一的规范化名称。在实际命名时要注意：实体名称能够唯一标识实体。不能有重复的名称，每一个实体必须有唯一的名称，意味着只要是相同名称就拥有相同的含义。反过来，同样含义就需要相同的名称，使用别名时例外；实体名称具有一定的含义。实体对应于现实世界中的一个物体，或者是一个抽象的概念。实体的名称最好具有代表性，是

可以表明它内涵的一个名词。

2) 实体定义

准确描述实体的定义，包括对实体的内容、范围、使用等多方面的描述。由于实体代表一组真实的或抽象的事物（人、物体、地点、事件、概念、多个事物的结合体等），它们拥有共同的属性和特征。那么在定义实体时，必须注意它的定义的内涵和外延是否足够概括这一类型的所有事物，是属于逻辑层次上的实体定义。

3) 实体别名

在现实生活中大量存在相同的事物拥有不同名称的情况，为了使人们能够更加方便地了解实体的含义，可以根据不同的使用习惯，给出该实体的别名。

根据以上内容，列出数据集内的所有实体，填写表 6.1。

表 6.1 实体列表

序号	实体名称	定义
1.		
2.		
3.		

4) 实体的内容描述

数据集实体的内容描述根据该实体的定义和说明信息确定，包括实体名称、别名、英文名称、短名等，具体填写内容见表 6.2 的实体描述字典。

表 6.2 实体描述字典

字段	可选性	描述
实体名称	必选	实体的标识，一般使用名词表达，通常名称都能反映出实体的属性和特征。一般用中文表示
别名	可选	实体的别名，一般使用名词表达。
英文名称	可选	实体的英文全称
短名	可选	实体的英文名称缩写
定义	必选	实体定义的详细描述
备注	可选	和实体相关的其它信息
版本标识	必选	用于实体的配置管理和控制。
实体来源	可选	说明实体定义的来源，来源包括已有的数据模式草案标准、已有的信息系统和其它来源。

6.2.2 材料实体属性设置

材料属性是对材料实体基本特征的描述，是数据集结构的基本元素。属性的定义内容包括属性名称、属性定义等：

1)序号。序号分两部分，前一部分对应于实体号，后一部分对应于该属性的编号。例如序号为 1 的实体的第一个属性即为 1.1。

2)属性名称：用于标识属性，不同的属性具备不同的属性名称。属性名称应当具备一个含义。如材料的牌号、用途，材料的屈服点、伸长率、硬度等。

3)属性定义：对属性的含义进行详细的解释，列出该材料实体的所有属性。

- 4)数据类型：表明该属性的数据类型。
 5)值域（可选）：表明属性的值所取值的范围。

表 6.3 属性列表

序号	属性名称	英文名称	定义	数据类型	值域（可选）
1.1					
1.2					
1.3					

6.2.3 材料实体联系定义

实体之间存在着相互联系，实体与实体间的关系抽象为联系。E-R 数据模型在实体的联系方面提供了较多的语义。在二元联系中，E-R 数据模型把联系区分为一对一（1:1）、一对多（1:N）和多对多（M:N）三种。这种方法还可以推广到多元联系，如三元联系也可以区分为 M:N:P，1:1:1，1:N:P，1:1:P 等联系。这种实体间联系的约束称为基数比约束。

材料应用的复杂性决定了材料实体间联系的复杂性，同样对应了 E-R 数据模型中的实体间的联系的多样性。E-R 图中矩形框代表实体，菱形框代表联系，联系与实体相连的弧上所注明的基数表示联系的基数比。例如，在材料科学中，材料实体与牌号对照表实体表示为一对一的关系；材料实体与硬度性能实体，材料实体与热学性能实体等表示为一对多的关系；材料实体与元素组成实体表示为多对多的关系。

6.3 表的设计

6.3.1 表的设计原则

■ 标准化和规范化

数据的标准化有助于消除数据库中的数据冗余。关系数据库范式理论是数据库设计的一种理论指南和基础，是数据库设计优劣的判断依据。“范式”是数据库设计过程中要依据的准则，数据库结构必须满足这种准则，在这里准则被称为规范化形式。在数据库进行检查和修改并使之符合范式的过程叫规范化。

范式按照规范化从宽松到严格分为多种不同级别的范式，通常使用的有第一范式、第二范式、第三范式及 BC 范式等范式是建立在函数依赖基础上的，本规范仅要求到第三范式，对 BC 范式暂不讨论。

■ 第一范式（1NF）

如果关系模式的每个关系的属性都是不可分的数据项，那么就称该关系模式是第一范式的模式。关系数据库设计研究的关系规范化是在第一范式的基础上进行的。

所谓第一范式是指数据库表的每一列都是不可分割的基本数据项，同一列中不能有多个值。

例如：

国标牌号	材料描述	
国标牌号	材料名称	说明
GB/T 2046-1980	锰青铜板	磁电转速表零件和高温工作零件

GB/T 2046-1980	镉青铜板	用于电器、仪表等工业
----------------	------	------------

以上表就不符合第一范式，主键重复，材料字段可再分，正确的应为：

国标牌号	材料名称	说明
GB/T 2046-1980	镉青铜板	磁电转速表零件和高温工作零件
GB/T 2044-1980	镉青铜板	用于电器、仪表等工业

■ 第二范式

如果关系模式是第一范式，而且每个非主属性完全函数依赖于候选键，那么就称该模式为第二范式。第二范式是在第一范式的基础上建立起来的。第二范式要求数据库表中的每个实例或行必须可以被唯一的区分。这个唯一属性列被称为主关键字。

第二范式要求实体的属性完全依赖于主关键字。所谓完全依赖是指不能存在仅依赖主关键字一部分的属性，如果存在，那么这个属性和主关键字的这一部分应该分离出来形成一个新的实体，新实体与原实体之间是一对多的关系。第二范式的主要任务就是在满足第一范式的前提下，消除部分函数依赖。

例如：假定元素组成关系表为 T_CONSTITUTE (国标牌号，材料名称，说明，元素 ID，元素名称，元素符号，元素比例)，关键字为组合关键字 (国标牌号，元素 ID)，因为存在如下决定关系：

(国标牌号，元素 ID) -> (材料名称，说明，元素名称，元素符号，元素比例)

这个数据库表不满足第二范式，因为存在如下决定关系：

(国标牌号) -> (材料名称，说明)

(元素 ID) -> (元素名称，元素符号)

这里存在组合关键字中的字段决定非关键字的情况。如果这样设计，将会存在数据冗余，更新、插入和删除异常等问题。所以我们把元素组成关系表改为如下三个表：

材料：T_MATERIAL (国标牌号，材料名称，说明)；

元素：T_ELEMENT (元素 ID，元素名称，元素符号)；

元素组成关系：T_CONSTITUTE (国标牌号，元素 ID，元素比例)。

这样的数据库表才是符合第二范式的。

■ 第三范式

如果关系模式是第二范式的，且关系模式中的所有非主属性对任何候选关键字都不存在传递依赖，则称该关系是属于第三范式的。

第三范式是在首先满足第二范式，其次非主属性之间不存在函数依赖。所谓的传递函数依赖，指的是如果存在“A->B->C”的决定关系，则C传递函数依赖于A。因此，满足第三范式的数据库表应该不存在如下依赖关系：

关键字段->非关键字段 x->非关键字段 y

例如：

假定材料类别关系表为 T_MATERIALB (国标牌号，材料名称，说明，所属材料类别，类别主要成分，类别用途)，关键字为单一关键字“国标牌号”，因为存在如下决定关系：

(国标牌号) -> (材料名称，说明，所属材料类别，类别主要成分，类别用途)

这个数据库是符合第二范式的，但是不符合第三范式，因为存在如下决定关系：

(国标牌号) -> (材料类别) -> (类别主要成分，类别用途)

即存在非关键字段“类别主要成分”“类别用途”对关键字段“国标牌号”的传递函数依赖。这样也会存在数据冗余，更新、插入和删除异常的情况。这样可以把材料类别关系表分为如下两个表：

材料：(国标牌号，材料名称，说明)；

材料类别：（材料类别，类别主要成分，类别用途）。

6.3.2 主键和外键选取

键选择原则：

■ 键设计 4 原则

- ◆ 为关联字段创建外键。
- ◆ 所有的键都必须唯一。
- ◆ 避免使用复合键。
- ◆ 外键总是关联唯一的键字段。

■ 使用系统生成的主键

设计数据库的时候采用系统生成的键作为主键，那么实际控制了数据库的索引完整性。这样，数据库和非人工机制就有效地控制了对存储数据中每一行的访问。采用系统生成键作为主键还有一个优点：当拥有一致的键结构时，找到逻辑缺陷很容易。

■ 不要用用户的键(不让主键具有可更新性)

在确定采用什么字段作为表的键的时候，可一定要小心用户将要编辑的字段。通常的情况下不要选择用户可编辑的字段作为键。

■ 可选键有时可做主键

把可选键进一步用做主键，可以拥有建立强大索引的能力。

6.4 索引的设计

索引使用原则：

- 索引是从数据库中获取数据的最高效方式之一。95%的数据库性能问题都可以采用索引技术得到解决。
- 逻辑主键使用唯一的成组索引，对系统键（作为存储过程）采用唯一的非成组索引，对外键列采用非成组索引。考虑数据库的空间有多大，表如何进行访问，还有这些访问是否主要用作读写。
- 大多数数据库都索引自动创建的主键字段，但是可别忘了索引外键，它们也是经常使用的键，比如运行查询显示主表和所有关联表的某条记录就用得上。
- 不要索引 memo/note 字段，不要索引大型字段（有很多字符），这样作会让索引占用太多的存储空间。
- 不要索引常用的小型表
- 不要为小型数据表设置任何键，假如它们经常有插入和删除操作就更别这样作了。对这些插入和删除操作的索引维护可能比扫描表空间消耗更多的时间。

6.5 视图的设计

与基本表、代码表、中间表不同，视图是一种虚表，它依赖数据源的实表而存在。视图是供程序员使用数据库的一个窗口，是基表数据综合的一种形式，是数据处理的一种方法，是用户数据保密的一种手段。

为了进行复杂处理、提高运算速度和节省存储空间，视图的定义深度一般不得超过三层。若三层视图仍不够用，则应在视图上定义临时表，在临时表上再定义视图。这样反复交迭定

义，视图的深度就不受限制了。

对于安全性要求较高的系统，视图的作用更加重要。对待不同人员使用不同视图，可以有效的保护数据库的安全性。

6.6 数据完整性设计

数据的完整性是指数据的正确性、有效性和相容性。数据的完整性约束是在数据库系统中，数据模型定义数据时所必须满足的一组规则。该规则是给定的数据模型中数据及其联系所具有的制约和储存规则，用以限定符合数据模型的数据状态以及状态的变化，以保证数据的正确、有效和相容。

在数据库中构建数据的完整性约束是确保数据完整有效的最可靠的方法。实际的系统中，一般将数据完整性约束分为标准的数据完整性约束和定制的数据完整性约束。标准的数据完整性约束是 DBS 已经实现的内部规则，包括域完整性、实体完整性和引用完整性。

域完整性(Field Integrity)保证一个数据库不包含无意义的或不合理的值，即保证表的某一列的任何值是该列域(即合法的数据集合)的成员。方法是限制列的数据类型、精度、范围、格式和长度等。

实体完整性(Entity Integrity)保证一个表中的每一行必须是唯一的(元组的唯一性)。为保证实体完整性，需指定一个表中的一列或一组列作为它的主键(Primary Key)。一个表中每行的主键必须确实含有一个值。一个表只能含有一个主键，如需要从其它列中除去重复的值，可以将一个或一组非主键列指定为一个候选键或唯一值键。

引用完整性(Reference Integrity)定义了一个关系数据库中不同的列和不同的表之间的关系(主键与外键)。要求一列或一组列中的值必须要与相关的一列或一组列中的值相匹配。从属的一列或一组列称之为外键(Foreign Key)。被引用的列或一组列称之为父键，父键必须是一个主键或唯一键。外键属于子表或明细表，父键属于父表或主表。若父键和外键属于同一表，则称之为自引用完整性。子表某行的外键必须与主表的主键相匹配，只要依赖于某主键的外键仍存在，主表中包含该主键的行就不能删除。

如果数据具有完整性，则表示数据有效(正确并且准确)，而且数据库的关系结构是完整的。引用完整性约束可以加强数据库的关系结构。这些规则使数据在各个表之间保持一致。

数据库完整性由各种各样的完整性约束来保证，因此可以说数据库完整性设计就是数据库完整性约束的设计。数据库完整性约束可以通过 DBMS 或应用程序来实现，基于 DBMS 的完整性约束作为模式的一部分存入数据库中。通过 DBMS 实现的数据库完整性按照数据库设计步骤进行设计，而由应用软件实现的数据库完整性则纳入应用软件设计。

数据库完整性对于数据库应用系统非常关键，其作用主要体现在以下几个方面：

- 数据库完整性约束能够防止合法用户使用数据库时向数据库中添加不合语义的数据。
- 利用基于 DBMS 的完整性控制机制来实现业务规则，易于定义，容易理解，而且可以降低应用程序的复杂性，提高应用程序的运行效率。同时，基于 DBMS 的完整性控制机制是集中管理的，因此比应用程序更容易实现数据库的完整性。
- 合理的数据库完整性设计，能够同时兼顾数据库的完整性和系统的效能。比如装载大量数据时，只要在装载之前临时使基于 DBMS 的数据库完整性约束失效，此后再使其生效，就能保证既不影响数据装载的效率又能保证数据库的完整性。
- 在应用软件的功能测试中，完善的数据库完整性有助于尽早发现应用软件的错误。

7. 运用设计

7.1 实施步骤

7.1.1 目标

根据逻辑设计和物理设计阶段的成果，在目标计算机上建立起实际的数据库结构、装载数据，并实际测试和运行数据库系统。

7.1.2 实施步骤

用 DBMS 提供的 DDL 来定义数据库结构；

数据装载；

编制与调试应用程序；

数据库试运行。

数据库实施步骤如图 7.1 所示：

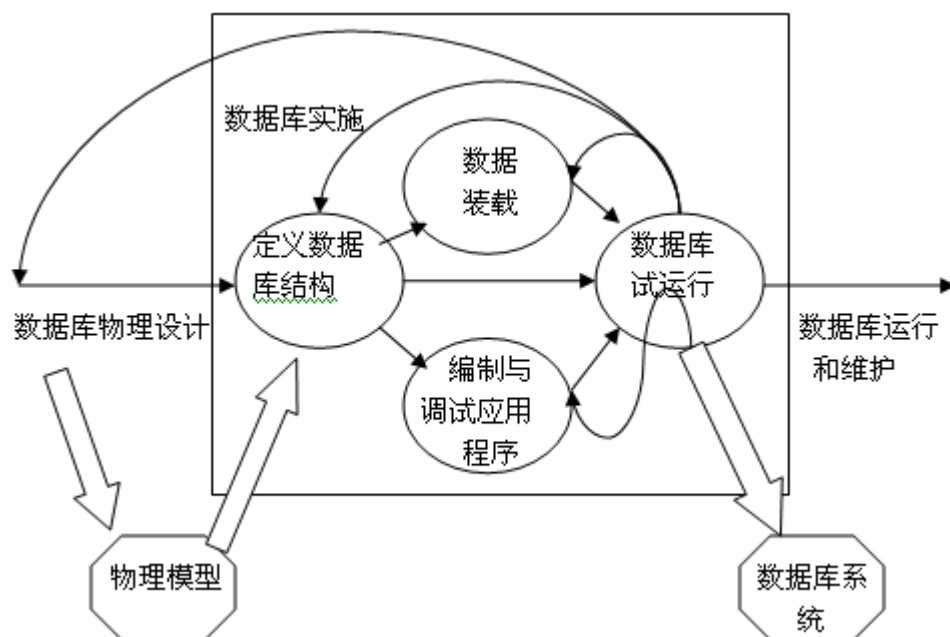


图 7.1 数据库实施步骤

7.1.2.1 定义数据库结构

根据数据库的逻辑结构与物理结构设计结果，就可以用所选用的 DBMS 提供的数据库定义语言(DDL)来严格描述数据库结构。

7.1.2.2 数据装载

数据库结构建立好后，就可以向数据库中装载数据了。数据装载的主要步骤包括：

- 筛选数据：原始数据可能存在于各个部门的文件或凭证中，首先将需要入库的数据筛选出来；
- 输入数据：借助系统提供的输入界面，将原始数据输入到计算机中；
- 校验数据：系统应采用多种检验技术，来保证输入数据的正确性，防止非法的、不一致的错误数据进入到数据库中；
- 转换数据：对于不符合数据库要求的数据，往往需要进行数据格式转换；
- 综合数据：对转换好的数据根据系统的要求进一步综合成最终数据。

7.1.2.3 编制与调试应用程序

数据库应用程序的设计应与数据库设计并行进行，数据库构建好后，就可以编制与调试数据库应用程序。编制与调试应用程序可以与组织数据入库同步进行，调试应用程序可先使用模拟数据。

7.1.2.4 数据库试运行

应用程序调试完成。并已有小部分数据入库后，就可以开始数据库试运行。主要工作包括：

- 功能测试：实际运行应用程序，执行各类数据库操作，检验各种功能是否正确无误，满足设计要求；
- 性能测试：测量系统的性能指标，检验是否符合设计指标；
- 备份和恢复：在数据库试运行阶段，由于系统还不稳定，软硬件故障随时都可能发生，因此必须做好数据库的备份和恢复工作。

7.2 运行维护

7.2.1 目标

当数据库试运行结果符合设计目标后，数据库就可以真正投入运行并进入维护阶段了。数据库运行维护将针对应用环境和物理存储的不断变化，对数据库设计进行评价、调整、修改等维护工作，是数据库设计的继续和提高。

7.2.2 人员要求

数据库的维护工作主要由 DBA 来完成。

7.2.3 工作任务

- 数据库备份和恢复：DBA 要针对不同的应用要求制定不同的数据备份计划，定期对数据库和日志文件进行备份，以保证一旦发生故障，能利用数据库备份和日志文件备份，尽快将数据库恢复到某种一致性状态，并尽可能减少对数据库的破坏；
- 数据库安全性和完整性控制：DBA 必须根据用户的实际需要授予其不同的操作权限，并在数据库运行过程中，根据环境的变化适当调整原有的安全性控制。同样由于应用环境的变化，数据库的完整性约束也会变化，需要 DBA 不断修正，以便满足用户要求；
- 数据库性能的监控、分析和改进：DBA 应借助 DBMS 系统性能监测工具，来监督系统运行状态，并仔细分析监测数据，判断当前系统是否处于最佳运行状态；如果不是，则需要通过调整某些参数来进一步改进数据库性能；
- 数据库的重组和重构：在数据库运行一段时间后，由于数据记录的不断增、删、改，会使数据库的物理存储变坏，从而降低数据库存储空间的利用率和数据的存取效率，使数据库性能下降，这时 DBA 需要借助 DBMS 提供的实用程序对数据库进行重组，或部分重组，数据库重组不会改变原设计的数据逻辑结构和物理结构，只是按原设计要求重新安排存储位置，回收垃圾，减少指针链，提高系统性能。同样随着数据库的应用环境的变化，可能会导致实体或对象发生变化，从而不得不适当调整数据库的模式和内模式，这时 DBA 需要对数据库进行重新构造，例如增加数据项，改变数据项的类型，增加或删除索引等。

7.3 安全设计

在数据库的设计中，说明将如何通过区分不同的访问者、不同的访问类型和不同的数据对象，进行分别对待而获得的数据库安全保密的设计考虑。

应根据 GB 17859-1999 《计算机信息系统安全保护等级划分准则》相关标准、GA/T 389-2002 《计算机信息系统安全等级保护数据库管理系统技术要求》的保密规定，确定具体材料数据库的安全保护等级并采取相应的用户访问控制、防病毒和备份等技术措施予以保障。

7.4 数据库设计文档规范

7.4.1 数据库规划报告编写规定

该报告是数据库规划阶段的成果文档。其主要内容包括：

- 系统简述：概述系统名称、任务提出、系统范围等；
- 系统特性：
 - ◆ 复杂性(系统是否设计多媒体、实时检索等复杂技术内容)；
 - ◆ 连续性(系统与现行系统关系、投资保护、可扩展性等)；
 - ◆ 移植性(不同计算机系统环境下的应用情况)。
- 系统目标：设计系统总目标、分目标(子目标)，并逐层分解，以求目标明确具体。
- 系统所需资源：现有情况、系统需求和落实情况。
- 成本估算：分期估算是成本估算常采用的方法，其内容涉及各工期中的人力资源，包括

不同水平的人员所要花费的时间及应花费的费用，必要的物质资源消耗即总的成本是人员投入成本和物资投入成本的总合。

- 效益评估：效益包括社会效益和经济效益。
- 可行性分析包括：技术可行性涉及硬软件问题和其他技术性问题；系统开发和运行环境的可行性包括需求的迫切性，领导与职工的支持，现行管理体制和管理水平能否保证系统顺利开发和实现。
- 设计原则：用户第一原则，系统开发的各个阶段尽可能吸收用户参加，倾听用户意见，交流问题，以便统一认识，提高设计质量。
- 技术路线：尽可能采用较为成熟的先进技术和产品，尽可能采用国际和国家及本企业的标准代码，尽可能保护现有投资(包括信息投资)，尽可能与周边环境相适应等。
- 系统开发在组织落实各类人员比例，每一成员能投入的时间等。
- 系统开发计划及进度安排等。

7.4.2 需求分析文档编写规定

需求分析文档常以需求分析说明书的形式出现。其主要内容包括：

- 需求调查表；
- 系统流程图；
- 数据流图(面向对象方法应包括用例图、构件图、顺序图、协作图)；
- 数据字典；
- 数据性能分析；
- 系统边界、远景描述等。

7.4.2.1 需求调查表

需求调查表一般包括五个方面的内容：

- 岗位职责及业务处理流程表：内容包括部门名称、岗位名称、本岗位职责和业务处理流程。
- 信息输入表内容包括部门名称、岗位名称、输入表名称、原始资料简单说明、信息输入数据项说明、信息提供者、提供方式以及提供周期等。
- 信息处理表：内容包括部门名称、岗位名称、输入表名称、信息处理名称、信息处理描述、信息输出名称以及信息输出方式等。
- 信息输出表：内容包括部门名称、岗位名称、输出表名称、信息处理名称、信息输出数据项说明、输出表接收者部门名称、输出表接收者岗位名称以及输出表接收者信息处理名称。
- 功能要求说明表：用户对系统的功能要求，包括功能分类及其简短描述。

7.4.2.2 数据流图

在需求分析阶段，数据流图(data flow diagram)被用来描述系统的组成及各部分之间的关系。数据流图表达了数据和处理过程之间的关系。在数据流图中主要用到的图形符号见表7.1。

符号	说明
	表示数据源点或终点，N 为名称
	表示数据变换的处理过程，T 为处理过程名称
	表示存储数据，名称写在框内
	表示数据流，名称写在上面

表 7.1 数据流图中的主要符号

7.4.2.3 数据字典

数据字典是各类数据描述的集合它是进行详细的数据收集和数据分析所获得的主要结果。因此在数据库设计中占有很重要的地位。

数据字典通常包括数据项、数据结构、数据流、数据存储和处理过程 5 个部分。其中数据项是数据的最小组成单位。若干个数据项可以组成一个数据结构，数据字典通过对数据项和数据结构的定义来描述数据流、数据存储的逻辑内容。

■ 数据项：是不可再分的数据单位。对数据项的描述通常包括以下内容：

数据项描述= {数据项名，数据项含义说明，别名，数据类型，长度，取值范围，取值含义，与其他数据项的逻辑关系}

其中取值含义、与其他数据项的逻辑关系数据结构(如该数据项等于另几个数据项的和)定义了数据的完整性约束条件，是设计数据检验功能的依据。

■ 数据结构：反映了数据之间的组合关系。一个数据结构可以由若干个数据项组成，也可以由若干个数据结构组成，或由若干个数据结构和数据项混合组成。

对数据结构的描述通常包括以下内容：

数据结构描述= {数据结构名，含义说明，组成：{数据项或数据结构}}

■ 数据流：是数据结构在系统内传输的途径。对数据流的描述通常包括以下内容：

数据流描述= {数据流名，说明，数据流来源，数据流去向，组成：{数据结构}，平均流量，高峰期流量}

其中数据流来源是说明该数据流来自哪个过程。数据流去向是说明该数据流将到哪个过程去。平均流量是指在单位时间里的传输次数。高峰期流量是指在高峰时期的数据流量。

■ 数据存储：是数据结构停留或保存的地方，也是数据流的来源和去向之一。对数据存储的描述通常包括以下内容：

数据存储描述= {数据存储名，说明，编号，流入的数据流，流出的数据流，组成：{数据结构}，数据量，存取方式}

其中数据量是指每次存取多少数据，每单位时间存取多少次等信息。存取方法包括是批处理还是联机处理，是检索还是更新，是顺序检索还是随机检索等。

■ 处理过程：具体处理逻辑一般用判定表或判定树来描述，数据字典中只需描述处理过程的说明性信息，通常包括以下内容：

处理过程描述= {处理过程名，说明，输入：{数据流}，输出：{数据流}，处理：{简要说明}}

其中简要说明中主要说明该处理过程的功能及处理要求，功能是指该处理过程用来做什么，处理要求包括处理频度要求。这些要求是后面物理设计的输入及性能评价标准。

可见，数据字典是关于数据库中数据的描述，即元数据，而不是数据本身。
数据字典是在需求分析阶段建立，在数据库设计过程中不断修改、充实、完善的。

7.4.3 概念设计文档编写规定

概念设计文档常以概念设计说明书的形式出现。其主要内容包括：

- 实体和属性定义表，内容包括：
 - ◆ 实体名称；
 - ◆ 所辖属性集合；
 - ◆ 实体的关键字属性。

- 联系定义表(包括继承和嵌套)。

- 域定义表。

- 实体联系图 (E-R 图)。

E-R 图主要包括如下几个主要构件：

- 矩形，表示实体集；
- 椭圆，表示属性；
- 菱形，表示联系集；
- 线段，将属性连接到实体集或将实体集连接到联系集；
- 双椭圆，表示多值属性；
- 虚椭圆，表示派生属性；
- 双线，表示一个实体集全部参与到联系集中；
- 双矩形，表示弱实体集。

7.4.4 逻辑设计文档编写规定

逻辑设计文档常以逻辑设计说明书的形式出现。其主要内容包括：

- 关系模式定义表（数据表）：内容包括关系名称、规范化级别、属性名称、代码、数据类型、长度、精度、是否主键、说明等。
- 完整性约束和函数依赖定义表：内容包括关系名称、属性名称、唯一性、空值、值约束、函数依赖等。
- 安全性约束定义表：内容包括关系名称、关系上的安全性约束、属性名称、属性上的安全性约束等。
- 域定义表：内容包括域名称、代码、数据类型、长度、精度等。
- 用户子模式定义表：内容包括子模式名称、属性、基础关系名称、基础关系属性、条件、应用等。
- 性能评价表：内容包括任务名称、存取字节数、占用磁盘空间字节数等。

7.4.5 物理设计文档编写规定

物理设计文档的内容视选用的 DBMS 的不同而存在差异，本节规定了比较通用的一些文档内容和要求。物理设计文档主要内容包括：

- 数据库对象命名及编程规范约定表。
- 数据库表空间定义表。

- 数据库文件说明表：包括服务器名称、数据库名称、使用网络协议、库文件初始大小、日志文件初始大小、备份策略等。
- 索引定义表：包括表名称、索引名称、索引代码、说明等。
- 聚簇定义表。
- 分区定义表。
- 数据库用户定义表：包括用户名称、代码、权限、说明等。

附录 A 数据集制作应用实例

本章将以某类材料数据库的设计过程，来说明数据库系统设计的有关理论与实际开发过程的对应关系。为了突出重点，本示例将简化描述需求，不建立需求模型，同时不考虑应用模型即不建立视图与索引，也不考虑与性能有关模型建立。

为了示例的典型性，考虑到材料科学数据的特点，假定已知的某类材料的以下相关信息。

(1) 该类材料的基本属性。其中一些属性（比如材料名称）可以把该类材料与其它材料区分开，而另有一些属性（如材料牌号）则可以区分这类材料中的各个材料。同时不同的国家的材料牌号有不同的定义。

(2) 该类材料的组织结构。由于组织结构较为复杂，本示例以元素组成作为代表。一个材料可以有多个元素组成，而不同的元素又可以组成不同的材料，即材料和元素组成之间存在多对多的联系。

(3) 该类材料的相关性能指标。本示例以该类材料的硬度性能和热学性能性能指标为代表。同时，如果条件不同比如温度不同，某个材料可以具有不同的硬度性能或是热学性能即这些指标和单个材料存在一对多的联系。

(4) 该类材料的制备和加工工艺。本示例主要假设了与该类材料制备相关实验的实验数据，数据以图形图像数据为例。

(5) 使用性能。假设已知腐蚀率等信息。

(6) 需要特殊考虑的信息。

- ◆ 有些数据需要特殊的考虑，材料数据的特点决定了要特别注意数据和实验的相关性的处理。有些数据离开特定的实验环境便变得没有任何意义，因此设计数据库时需要考虑实验和数据的相关性问题；

- ◆ 域值问题。一方面，准确的域定义对数据库域完整性约束的可靠保证。另一方面，材料数据域值问题又不同于其它科学数据。

在这些信息的基础上我们建立一个示例数据库，描述数据之间的关系，并将数据装载到数据库中。

A.1 数据库命名

数据库名称：MaterialADb

A.2 定义实体

实体一：材料 A（表示一类材料，如铁、碳钢等）；

实体二：硬度性能；

实体三：热学性能；

实体四：元素；
 实体五：牌号对照表；
 实体六：腐蚀；
 实体七：制备实验；

表 8.1 实体内容描述字典

序号	实体名称	英文名称	定义	版本标识
1	材料 A	T_MATERIALA	表示一类材料的实体	1.0
2	硬度性能	T_SOLIDITY	表示材料 A 硬度性能的相关指标	1.0
3	热学性能	T_CALORIFICS	表示材料 A 热学性能的相关指标	1.0
4	元素	T_ELEMENT	所有元素列表	1.0
5	牌号对照表	T_ANTITHESES	不同国家的牌号对照表	1.0
6	腐蚀	T_ERODE	表示材料 A 的有关腐蚀的使用性能	1.0
7	制备实验	T_EXPERIMENT	表示与材料 A 相关的制备实验	1.0

A.3 确定主键

材料 A 实体的主键：国标牌号 (PK_MATERIALA_GBTRADEMARK)；
 硬度性能实体的主键：硬度性能 ID (PK_SOLIDITY_ID)；
 热学性能实体的主键：热学性能 ID (PK_CALORIFICS_ID)；
 元素实体的主键：元素 ID (PK_ELEMENT_ID)；
 牌号对照表实体的主键：牌号 ID (PK_TRADEMARK_ID)；
 腐蚀实体的主键：腐蚀 ID (PK_ERODE_ID)；
 制备实验实体的主键：实验编号 (PK_EXPERIMENT_ID)；

A.4 定义域

材料数据有自己独特的特点，因此在域的设计时应该根据实际情况进行设计。在本示例中只取一个简单例子。

以“元素组成”关系中的属性“比例”为例，一个材料可能有多种元素组成，但比例的取值范围为[0,1]，因此应该定义域[0,1]，“比例”属性应该受该域的约束。

A.5 确定实体属性

1. 列出各个实体属性

- 材料 A 实体：材料名称、国标牌号、说明；

材料A	
材料名称	A15
国标牌号	<pi> A20 <M>
说明	VA1024
国标牌号	<pi>

图 8.1 材料 A 实体

- 硬度性能实体：硬度性能 ID，布氏硬度，洛氏硬度，维氏硬度，努氏硬度，肖氏硬度，温度；

硬度性能	
硬度性能ID	<pi> <M>
布氏硬度	F
洛氏硬度	F
维氏硬度	F
努氏硬度	F
肖氏硬度	F
温度	F
硬度性能ID	<pi>

图 8.2 硬度性能实体

- 热学性能实体：热学性能 ID，热导率，比热容，规定非比例扭转应力，热辐射系数，热膨胀系数，温度；

热学性能	
热学性能ID	<pi> <M>
热导率	F
比热容	F
规定非比例扭转应力	F
热辐射系数	F
热膨胀系数	F
温度	F
热学性能ID	<pi>

图 8.3 热学性能实体

- 元素实体：元素 ID，元素名称，元素符号；

元素	
元素名称	A15
元素符号	A10
元素ID	<pi> <M>
元素ID	<pi>

图 8.4 元素实体

- 牌号对照表实体：牌号 ID，国际牌号，日本牌号，德国 DIN 牌号；

牌号对照表		
牌号ID	<u><pi> I</u>	<u><M></u>
国际牌号	A20	
日本牌号	A20	
德国DIN牌号	A20	
牌号ID	<pi>	

图 8.5 牌号对照表实体

- 腐蚀实体：腐蚀 ID，平均腐蚀率；

腐蚀		
腐蚀ID	<u><pi> I</u>	<u><M></u>
平均腐蚀率	F	
腐蚀ID	<pi>	

图 8.6 腐蚀实体

- 制备实验实体：实验编号，实验条件 1，实验条件 2，图形图像数据；

制备实验		
实验编号	<u><pi> I</u>	<u><M></u>
实验条件1	VA1024	
实验条件2	VA1024	
图形图像数据	LBIN	
实验编号	<pi>	

图 8.7 制备实验实体

2. 进行属性定义

属性定义表中，数据类型使用了通用的类型，在确定 ER 图时并不需要考虑与某个数据库的相关性。比如图形图像数据，用 SQL Server2000 数据库时数据类型为 ntext，而使用 Oracle 数据库时数据类型为 BLOB。而本表中用 LBIN（long binary）通用表示。

表 8.2 属性定义表

序号	属性名称	英文名称	定义	数据类型
■ 材料 A				
1.1	材料名称	MATERIALA_MATNAME	材料的名称	Characters(15)
1.2	国标牌号	PK_MATERIALA_GBTRAD EMARK	中国自定义牌号	Characters(20)
1.3	说明	MATERIALA_DESCRIBE	关于材料的说明	Variable Characters(1024)
硬度性能实体				
2.1	硬度性	PK_SOLIDITY_ID	硬度性能	Integer

	能 ID		的 ID 号	
2.2	布氏硬度	SOLIDITY_BUSSOL	布氏硬度	Float
2.3	洛氏硬度	SOLIDITY_LUOSSOL	洛氏硬度	Float
2.4	维氏硬度	SOLIDITY_WEISSOL	维氏硬度	Float
2.5	努氏硬度	SOLIDITY_NUSSOL	努氏硬度	Float
2.6	肖氏硬度	SOLIDITY_XIAOSSOL	肖氏硬度	Float
2.7	温度	SOLIDITY_TEMP	温度	Float
热学性能实体				
3.1	热学性能 ID	PK_CALORIFICS_ID	热学性能 ID	Integer
3.2	热导率	CALORIFICS_LEAD	热导率	Float
3.3	比热容	CALORIFICS_SPECIFICHEAT	比热容	Float
3.4	规定非比例扭转应力	CALORIFICS_REGNOPRO	规定非比例扭转应力	Float
3.5	热辐射系数	CALORIFICS_CERADIATE	热辐射系数	Float
3.6	热膨胀系数	CALORIFICS_CEXPAND	热膨胀系数	Float
3.7	温度	CALORIFICS_TEMP	热膨胀系数	Float
元素实体				
4.1	元素 ID	PK_ELEMENT_ID	元素 ID	Integer
4.2	元素名称	ELEMENT_NAME	元素名称	Characters(15)
4.3	元素符号	ELEMENT_SYMBOL	元素符号	Characters(20)
牌号对照表实体				
5.1	牌号 ID	PK_TRADEMARK_ID	牌号 ID	Integer
5.2	国际牌号	TRADEMARK_ISO	国际牌号	Character
5.3	日本牌号	TRADEMARK_JIS	日本牌号	Characters(20)
5.4	德国 DIN 牌号	CALORIFICS_REGNOPRO	德国 DIN 牌号	Characters(20)
腐蚀实体				

6.1	腐蚀 ID	PK_ERODE_ID	腐蚀 ID	Integer
6.2	平均腐蚀率	ERODE_RATE	平均腐蚀率	Float
制备实验实体				
7.1	热学性能 ID	PK_CALORIFICS_ID	热学性能 ID	Integer 20
7.2	实验条件 1	EXPERIMENT_CONDITION_1	实验条件 1	Variable Characters(1024)
7.3	实验条件 2	EXPERIMENT_CONDITION_2	实验条件 2	Variable Characters(1024)
7.4	图形图像数据	EXPERIMENT_GRAPHICS_DATA	图形图像数据	LBIN

A. 6 确定实体间的联系

实体之间有多种联系，有一对一，一对多，多对多的联系，比如材料 A 实体与硬度性能实体为一对多的关系，如图 8.8 表示了一对多的关系。

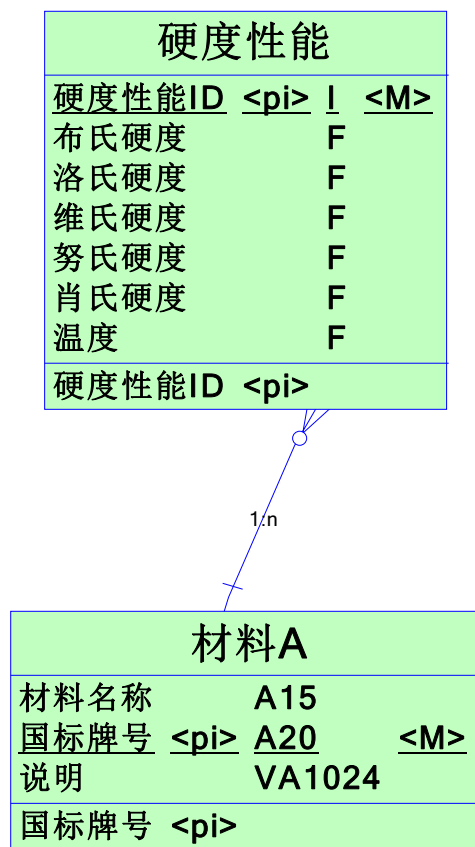


图 8.8 1:n 联系图

而材料 A 与牌号对照表之间是一一对一的关系，它的处理如图 8.9。

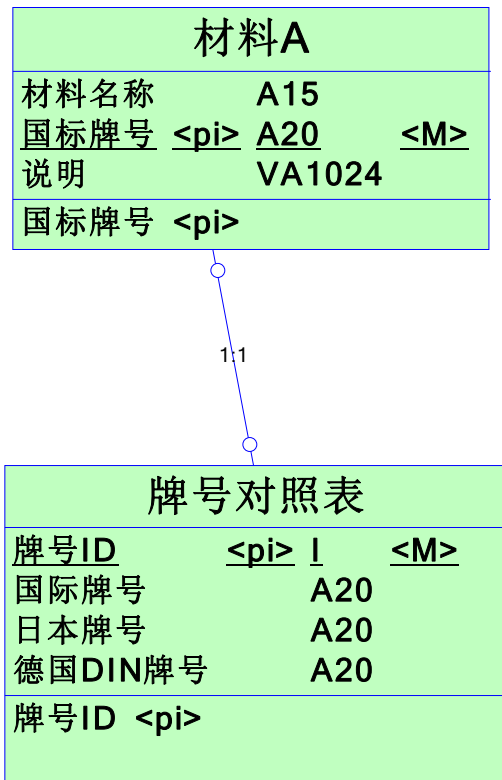


图 8.9 1:1 联系图

对于多对多的处理，需要把多对多转换为两个一对多联系，通过增加一个联系实体来实现，如图 8.10 所示。

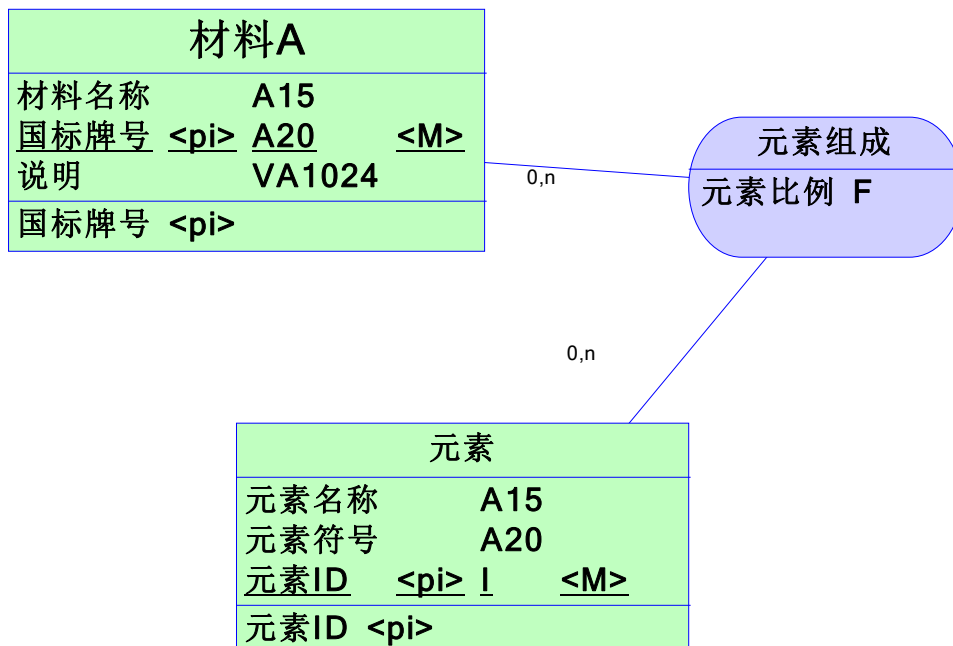


图 8.10 m:n 联系图

通过以上处理，最终实现的实体关系图如图 8.11 所示。

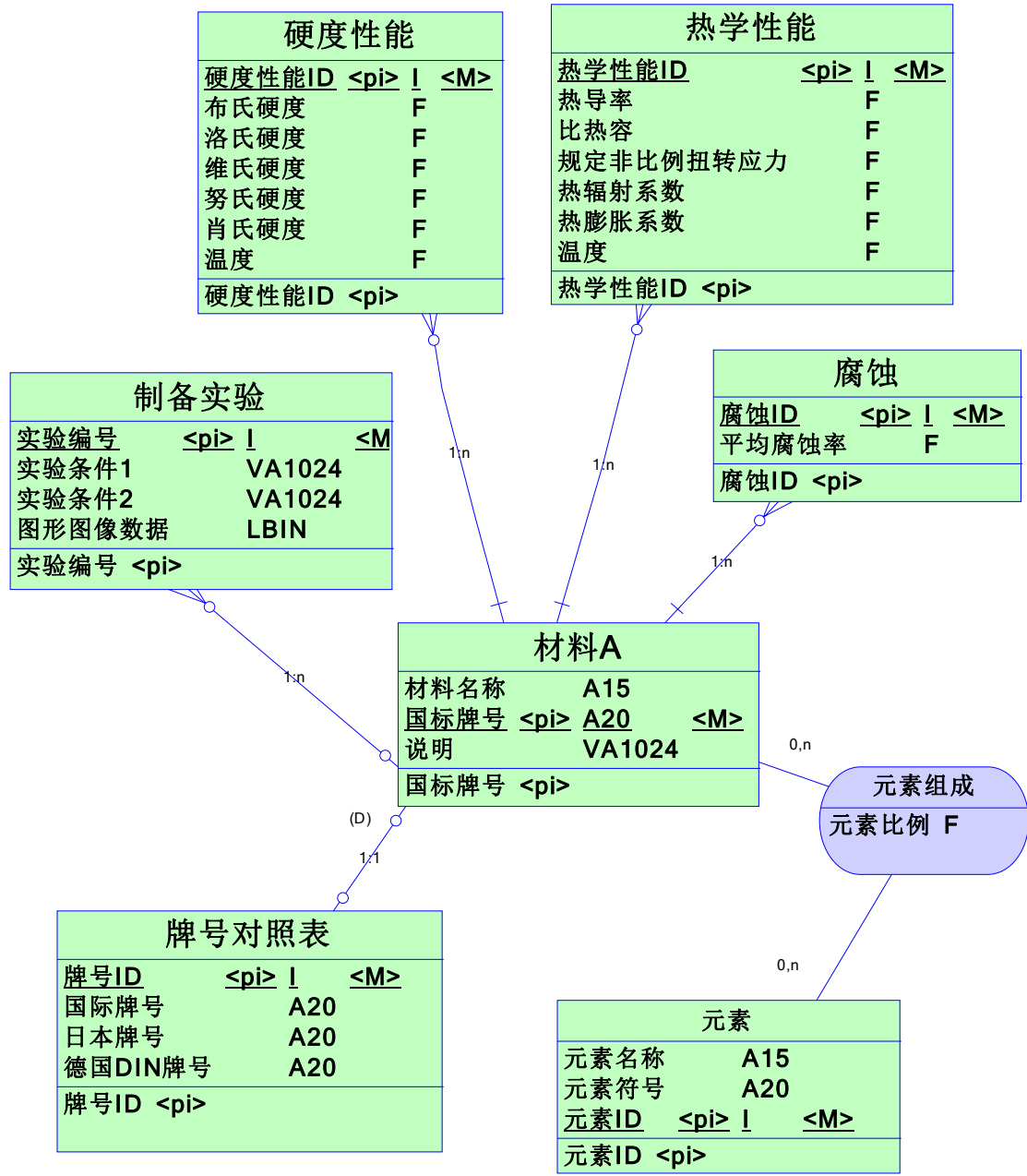


图 8.11 实体联系图

A.7 选择数据库

根据实际需要，选择合适的数据库。本示例中选择 Oracle 10g 数据库。

A.8 E-R 图转换为实际关系数据结构

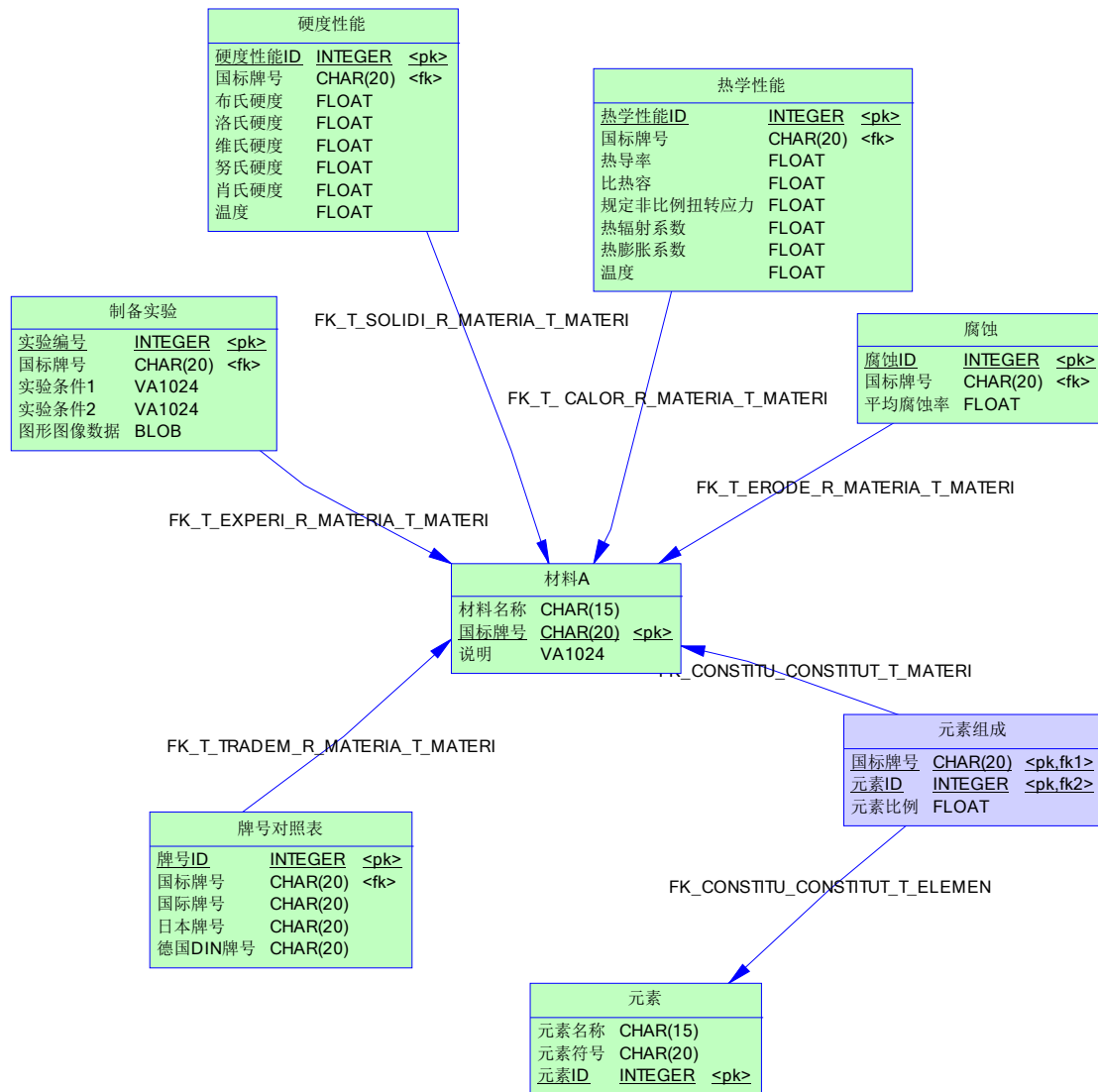


图 8.12 物理模型视图

针对一对多的关系，以及三范式的规范化和引用的完整性，添加外键，如硬度性能、热学性能表添加国标牌号作为外键，与材料 A 表中的国标牌号相对应。

A.9 范式检查与完整性检查

A.9.1 范式检查

严格按照以上各个步骤设计，将符合第二范式的要求，即实体的属性完全依赖于主关键字。而第三范式要求在首先满足第二范式，其次非主属性之间不存在函数依赖。

根据本规范 6.3 章节的要求，要求按照本规范设计的数据库满足第三范式。以材料 A 表为例，国际牌号是主键即主属性，材料名称和说明均依赖于主属性，这样满足了第二范式的定义。而非主属性即材料名称和说明之间不存在函数依赖，故材料 A 表满足第三范式。以此为例，对所有表进行第三范式的检查。

如果不满足第三范式，需要对模型就行修改，通过模式分解，把第二范式分解为第三范式。

A. 9. 2 完整性检查

为了确保设计出的数据库符合关系模型，并且设计出的数据库具有实际意义，需要进行完整性检查。完整性检查主要是对域完整性、实体完整性和引用完整性进行检查。

域完整性(Field Integrity)保证一个数据库不包含无意义的或不合理的值，即保证表的某一行列的任何值是该列域(即合法的数据集合)的成员。方法是限制列的数据类型、精度、范围、格式和长度等。

本示例中进行了域定义，如果不进行域定义，例如元素所占的比例如果输入错误，可能会存在负值，而这种值将没有任何意义。因此要对所定义的域完整性进行检查。一方面，检查域是否包含了所有的取值；另一方面确保无意义的的数据不被输入。

实体完整性(Entity Integrity)保证一个表中的每一行必须是唯一的(元组的唯一性)。为保证实体完整性，需指定一个表中的一列或一组列作为它的主键(Primary Key)。一个表中每行的主键必须确实含有一个值。一个表只能含有一个主键，如需要从其它列中除去重复的值，可以将一个或一组非主键列指定为一个候选键或唯一值键。

实体完整性检查主要是看是否设计时每个表都有主键，并且要求主键具有唯一性。如果不满足实体完整性，例如一个表不设置主键，有两条记录，材料名称相同，都是铅青铜，将无法区分这两种材料。

表 8.3 无主键设置表

材料名称	说明
铅青铜	有高的导电率。
铅青铜	常温及较高的温度下具有较高的强度和硬度。

而当设置材料牌号为主键后，根据牌号很容易搜区分这两个不同的材料。

表 8.4 主键设置表

材料牌号 (主键)	材料名称	说明
QZr0.2	铅青铜	有高的导电率。
QZr0.4	铅青铜	常温及较高的温度下具有较高的强度和硬度。

引用完整性(Reference Integrity)定义了一个关系数据库中不同的列和不同的表之间的关系(主键与外键)。要求一行或一组列中的值必须要与相关的一行或一组列中的值相匹配。从属的一行或一组列称之为外键(Foreign Key)。被引用的列或一组列称之为父键，父键必须是一个主键或唯一键。外键属于子表或明细表，父键属于父表或主表。若父键和外键属于同一表，则称之为自引用完整性。子表某行的外键必须与主表的主键相匹配，只要依赖于某主键的外键仍存在，主表中包含该主键的行就不能删除。

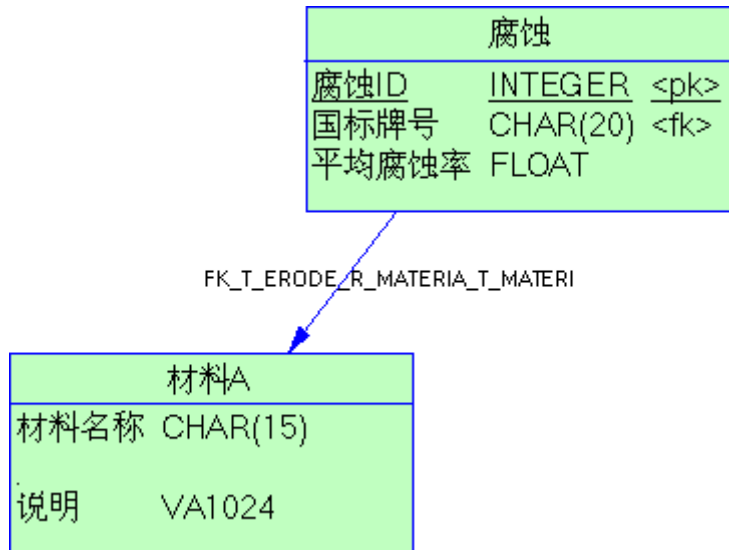


图 8.13 未符合引用完整性图

如图 8.13 不符合引用完整性，腐蚀表中的外键国标牌号，在它依赖的材料 A 表中并没有该列。这样的话，假如我们知道腐蚀表中一条记录为

腐蚀 ID	国标牌号	平均腐蚀率
123	A	X

如果我们想知道该条记录对应的材料名称，由于引用的不完整，我们将无法实现。正确的实现应该如图 8.14 所示。

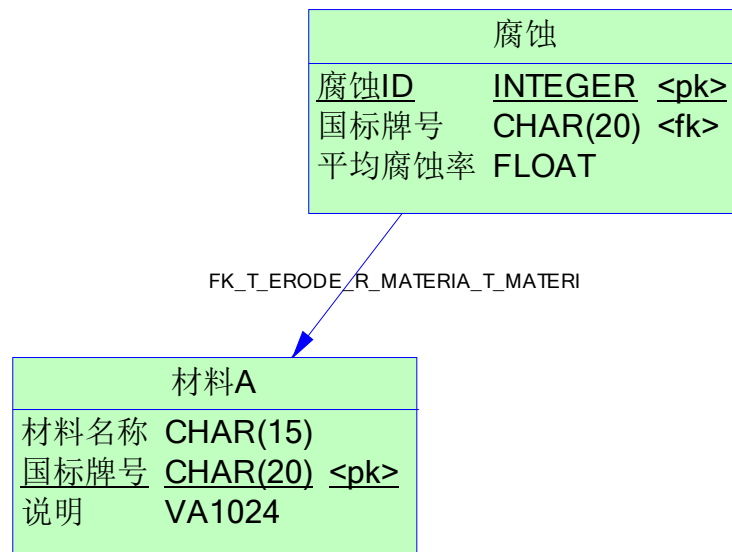


图 8.14 符合引用完整性图

A. 10 将数据结构转化为数据库表

按照 8.7 中选择的特定数据库，根据关系数据结构，编写针对该数据库的 SQL 语句，创建数据库及表，如表 8.5~8.12 所示，最后将现有的原始数据装载到数据库表中。

表 8.5 材料 A 表 (T_MATERIALA 表)

列名	中文名称	数据类型	长度
MATERIALA_MATNAME	材料名称	CHAR	15
PK_MATERIALA_GBTRADEMARK	国标牌号	CHAR	20
MATERIALA_DESCRIBE	说明	VA1024	1024

表 8.6 硬度性能表 (T_SOLIDITY 表)

列名	中文名称	数据类型	长度
PK_SOLIDITY_ID	硬度性能 ID	INTEGER	
PK_MATERIALA_GBTRADEMARK	国标牌号	CHAR	20
SOLIDITY_BUSSOL	布氏硬度	FLOAT	
SOLIDITY_LUOSSOL	洛氏硬度	FLOAT	
SOLIDITY_WEISSOL	维氏硬度	FLOAT	
SOLIDITY_NUSSOL	努氏硬度	FLOAT	
SOLIDITY_XIAOSSOL	肖氏硬度	FLOAT	
SOLIDITY_TEMP	温度	FLOAT	

表 8.7 热学性能表 (T_CALORIFICS 表)

列名	中文名称	数据类型	长度
PK_CALORIFICS_ID	热学性能 ID	INTEGER	
PK_MATERIALA_GBTRADEMARK	国标牌号	CHAR	20
CALORIFICS_LEAD	热导率	FLOAT	
CALORIFICS_SPECIFICHEAT	比热容	FLOAT	
CALORIFICS_REGNOPRO	规定非比例 扭转应力	FLOAT	
CALORIFICS_CERADIATE	热辐射系数	FLOAT	
CALORIFICS_CEXPAND	热膨胀系数	FLOAT	
CALORIFICS_TEMP	温度	FLOAT	

表 8.8 元素表 (T_ELEMENT 表)

列名	中文名称	数据类型	长度
ELEMENT_NAME	元素名称	CHAR	15
PK_ELEMENT_ID	元素 ID	INTEGER	
ELEMENT_SYMBOL	元素符号	CHAR	20

表 8.9 牌号对照表表 (T_TRADEMARK 表)

列名	中文名称	数据类型	长度
PK_TRADEMARK_ID	牌号 ID	INTEGER	
PK_MATERIALA_GBTRADEMARK	国标牌号	CHAR	20
TRADEMARK_ISO	国际牌号	CHAR	20
TRADEMARK_JIS	日本牌号	CHAR	20
TRADEMARK_DIN	德国 DIN 牌号	CHAR	20

表 8.10 腐蚀表 (T_ERODE 表)

列名	中文名称	数据类型	长度
PK_ERODE_ID	腐蚀 ID	INTEGER	
PK_MATERIALA_GBTRADEMARK	国标牌号	CHAR	20
ERODE_RATE	平均腐蚀率	FLOAT	

表 8.11 制备实验表 (T_EXPERIMENT 表)

列名	中文名称	数据类型	长度
PK_EXPERIMENT_ID	实验编号	INTEGER	
PK_MATERIALA_GBTRADEMARK	国标牌号	CHAR	20
EXPERIMENT_CONDITION_1	实验条件 1	VA1024	1024
EXPERIMENT_CONDITION_2	实验条件 2	VA1024	1024
EXPERIMENT_GRAPHICS_DATA	图形图像数据	BLOB	

表 8.12 元素组成表 (T_CONSTITUTE 表)

列名	中文名称	数据类型	长度
PK_CALORIFICS_ID	热学性能 ID	INTEGER	
PK_MATERIALA_GBTRADEMARK	国标牌号	CHAR	20
PK_ELEMENT_ID	元素 ID	INTEGER	
CONSTITUTE_PROPORTION	元素比例	FLOAT	