

# 数据库设计与开发

补充讲义

Copyright © 2000-2025年3月

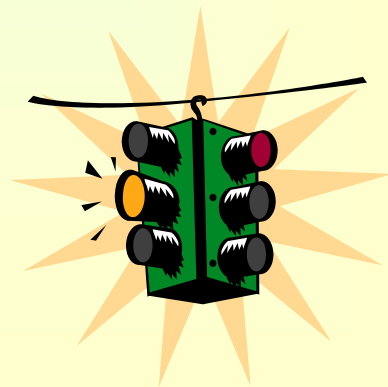
版权所有 请勿复制、传播

赵 小 林

北 京 理 工 大 学

# 补充 数据库设计

- § 1 数据库设计概述
- § 2 需求分析
- § 3 概念结构设计
- § 4 逻辑结构设计
- § 5 数据库的物理设计
- § 6 数据库的实施和维护
- § 7 小结



# §1 数据库设计概述

- 数据库设计概述
- 数据库和信息系统
- 数据库设计的特点
- 数据库设计方法简述
- 数据库设计的基本步骤

# §1 数据库设计概述

➤数据库设计有两个含义:

⑩一个是指数据库系统设计, 即设计一个DBMS系统,

⑩一个是指数据库应用系统的设计。

➤我们这里要讨论的数据库设计是指后者, 即在现有DBMS的基础上建立数据库及应用系统的整个过程。

# §1 数据库设计概述

- 要建立一个数据库应用系统，需要根据数据处理的规模，对应用系统的性能要求等选择合适的计算机硬件配置(如计算机的选型，是否上网等)、软件配置(如操作系统、汉字系统等)、选定数据库管理系统，组织开发人员小组，在熟悉计算机软硬件及DBMS的基础上，完成整个应用系统的设计工作。

# §1 数据库设计概述

- 以数据库为核心的、普通的应用系统人们通常称为管理信息系统（Management Information System, MIS系统）。它一般具有对信息的存储、检索、加工等功能。随着数据库技术的发展和广泛普及, 各行各业都有大量的信息需要进行管理, 要求建立信息管理系统。

# §1 数据库设计概述

- 如何建立一个高效实用的数据库应用系统，是数据库应用领域研究的一个主要课题。
- 在数据库应用初期，数据库设计往往是凭设计者的经验、知识、水平，因此设计出的应用系统性能好坏差别很大，常常不能满足应用要求。数据库工作者经过大量探索和研究，提出了不少的数据库设计方法，如新奥尔良(New Orleans)方法，规范化方法，基于E-R模型的数据库设计方法等。

# §1 数据库设计概述

- 实践表明，数据库设计是一项软件工程，应该把软件工程的原理和方法应用到数据库设计中。与一般软件工程相比，数据库设计与应用环境联系紧密，应用系统的信息结构复杂，加之数据库系统本身的复杂性，因此数据库设计具有自身的特点，逐渐形成了数据库设计方法学。





# §1.1 数据库和信息系统

- 从使用者角度看，信息系统是提供信息、辅助人们对环境进行控制和进行决策的系统。数据库是信息系统的核心和基础。它把信息系统中大量得数据按一定的模型组织起来，提供储存、维护、检索数据的功能，使信息系统可以方便及时准确的从数据库中获得所需的信息。一个信息系统的各个部分能否紧密的结合在一起以及如何集合，关键在数据库。因此只有对数据库进行合理的逻辑设计和有效的物理设计才能开发出完善而高效的信息系统。数据库设计是信息系统开发和建设的重要组成部分。

# §1.1 数据库和信息系统

- 专业人员应具备的技术和知识
  - 数据库的基本知识和数据库设计技术
  - 计算机科学的基础知识和程序设计的方法 and 技巧
  - 软件工程的原理和方法
  - 应用领域的知识



## §1.2 数据库设计的特点

- 数据库建设是硬件、软件和干件（技术与管理的界面）的结合。这是数据库设计的特点之一。
- 数据库设计应该和应用系统设计相结合，也就是说，整个设计过程中要把结构设计和行为设计密切结合起来。这是数据库设计的特点之二。

## §1.2 数据库设计的特点

- 传统的软件工程忽视对应用中数据语义的分析和抽象。如结构化设计方法（SD）和逐步求精的方法，只要有可能就尽量推迟数据结构的设计。
- 数据库应用系统要求各应用程序共享数据库模式，因而数据库（模式）设计质量直接影响系统中各处理过程的性能和质量。



## §1.3 数据库设计方法简述

- 规范设计法中比较著名的有新奥尔良方法。它将数据库设计分为四个阶段：需求分析、概念设计、逻辑设计和物理设计。其后，S.B.Yao等又将数据库设计分为五个步骤。又有I.R.Palmer等主张把数据库设计当成一步接一步的过程，并采用一些辅助手段实现每一过程。
- 基于E-R模型的数据库设计方法，基于3NF的设计方法，基于抽象语法规范的设计方法等，是在数据库设计的不同阶段上支持实现的具体技术和方法。

## §1.3 数据库设计方法简述

- 规范化设计方法从本质上说仍然是手工方法，其基本思想是过程迭代和逐步求精。
- 著名的数据库开发工具
  - ⑩ Design
  - ⑩ Power Designer
- 计算机辅助软件工程（CASE）已经开始强调数据库自动设计。



# §1.4 数据库设计的基本步骤

- 早期数据库设计

- 由于应用涉及面小，通常只是处理某一方面的应用，如工资管理系统，人事管理系统等，需要比较简单，数据结构不太复杂。设计人员在了解用户的信息要求，处理要求，数据量之后，就可以经过综合、分析，建立起数据模型。然后结合DBMS，将数据的逻辑结构、物理结构、系统性能一起考虑，直接编程，完成应用系统的设计。这种手工设计方法，数据库设计的好坏完全取决于设计者的经验、水平，缺少科学根据，因而很难保证设计质量。

# §1.4 数据库设计的基本步骤

- 现代数据库系统与早期数据库系统比较
  - 数据间的关系十分复杂，仅凭设计者的经验很难准确表达不同用户的要求和数据间的关系。
    - ⑩ 直接把逻辑结构，物理结构、系统性能一起考虑，涉及的因素太多，设计过程复杂，难以控制
  - 在设计中缺乏文档资料，很难与用户交流。而准确了解用户需求是数据库应用系统是否成功的关键。
  - 大型数据库系统一般设计周期都比较长，有的可能需要二三年时间。如果在设计后期才发现错误，轻则影响系统质量，重则导致整个设计失败。因此在设计过程中需要进行阶段评审，及时发现错误，及时纠正。



## §1.4 数据库设计的基本步骤

- 数据库的设计可分为以下六个阶段：
  - 1. 需求分析
  - 2. 概念设计
  - 3. 逻辑设计
  - 4. 物理设计
  - 5. 数据库实施和运行
  - 6. 数据库的使用和维护

# §1.4 数据库设计的基本步骤

- 数据库设计的基本步骤

- 需求分析阶段

- ⑩ 主要是获得用户对所要建立数据库的信息要求和处理要求的全面描述。通过调查研究，了解用户业务流程，与用户取得对需求的一致认识。

- ⑩ 是整个设计过程的基础，是最困难、最耗费时间的一步。

# §1.4 数据库设计的基本步骤

- 数据库设计的基本步骤

- 概念设计阶段

⑩ 要对收集的信息、数据进行分析抽象，确定实体、属性及它们之间的联系，将各个用户的局部视图合并成一个总的全局视图，形成独立于计算机的反映用户观点的概念模式。概念模式与具体DBMS无关，接近现实世界，结构稳定，用户容易理解，能较准确地反映用户的信息需求。

# §1.4 数据库设计的基本步骤

- 数据库设计的基本步骤

- 逻辑设计

- ⑩ 要在概念模式的基础上导出具体DBMS可处理的数据库的逻辑结构，即确定数据库模式和子模式。包括确定数据项，记录及记录间的联系，安全性，完整性，一致性约束等。
    - ⑩ 要检查导出的逻辑结构是否与概念模式一致，从功能和性能上是否能满足用户要求，要进行模式评价。如果达不到用户要求，还要反复、修工或重新进行设计。

# §1.4 数据库设计的基本步骤

- 数据库设计的基本步骤

- 数据库物理设计阶段

- ⑩任务是确定数据在介质上的物理存储结构，即数据在介质上如何存放，包括存取方法及存取路径的选择。物理设计的结果将导出数据库的存储模式。
- ⑩逻辑设计和物理设计的好坏对数据库的性能影响很大，在物理设计完后，要进行性能分析，测试。如果需要，要重新设计逻辑结构和物理结构。在逻辑结构和物理结构确定后，就可以建立数据库了。

# §1.4 数据库设计的基本步骤

- 数据库设计的基本步骤

- 数据库实施阶段

- ⑩ 包括建立实际数据库结构，装入数据，完成编码，进行测试。然后就可以投入运行。在这之间进行的结构特性设计可以和行为特性即应用程序的设计结合起来进行，以相互参照，补充完善各阶段的设计。

# §1.4 数据库设计的基本步骤

- 数据库设计的基本步骤

- 数据库运行和维护阶段

- ⑩数据库应用系统经过试运行以后可投入正式运行。在数据库系统运行过程中必须不断的对其进行评价、调整和修改。

# §1.4 数据库设计的基本步骤

- 数据库生存期

- 按照软件工程的设计思想，软件生存期指软件从开始分析、设计直到停止使用的整个时间。使用和维护阶段是整个生存期的最长时间段。数据库使用和维护阶段需要不断完善系统性能和改进系统功能，进行数据库的再组织和重构造，以延长数据库使用时间。





## §2 需求分析

- 需求分析
- 需求分析的任务
- 需求分析的方法
- 数据字典

## §2 需求分析

- 需求分析

- 简单的说就是分析用户的要求。
- 需求分析是设计数据库的起点，需求分析的结果是否准确的反映了用户的实际要求，将直接影响到后面各个阶段的设计，并影响到设计结果是否合理和实用。



## §2.1 需求分析的任务

- 需求分析的任务

- 需求分析要获得数据库设计所必需的数据信息。这些信息包括用户的：

- ⑩ 信息要求

- ∞ 信息需求指设计范围内涉及的所有信息的内容、特征、需要存储的数据。

- ⑩ 处理要求

- ∞ 处理需要是用户对信息加工处理的要求，包括发生的频度，响应时间，批处理还是联机处理。

# §2.1 需求分析的任务

- 需求分析的任务

- 需求分析要获得数据库设计所必需的数据信息。这些信息包括用户的：

- ⑩完整性、安全性要求

- 对数据约束的要求，安全保密要求等。

- ⑩确定用户的需求是很难的，用户不清楚计算能做什么，不能做什么，不清楚计算机的运行机制，甚至不完全明白自己的需求；而设计人员不了解用户领域的知识，可能会误解用户的需求。



## §2.2 需求分析的方法

- 调查用户需求的具体步骤：
  - (1) 调查组织机构情况。
  - (2) 调查各部门的业务活动情况。
  - (3) 在熟悉了业务活动的基础上，协助用户明确对新系统的各种要求，包括信息要求、处理要求、完全性和完整性要求，这是调查的又一重点。
  - (4) 确定新系统的边界。

## §2.2 需求分析的方法

⑩在调查过程中，可以根据不同的问题和条件，使用不同的调查方法。

- 常用的调查方法有：
  - (1) 跟班作业开调查会
  - (2) 请专人介绍
  - (3) 询问
  - (4) 设计调查表请用户填写
  - (5) 查阅记录

## §2.2 需求分析的方法

- 了解用户需求以后，还需要进一步分析和表达用户的需求。
- 常用的分析方法有结构化分析（SA）方法，采用自顶向下、逐层分解的方式。
- 应借助数据字典（DD）和数据流图（DFD）来进行描述。
- 需求分析过程是一个反复的过程。



## §2.3 数据字典

- 数据流图表达了数据和处理的关系，数据字典则是系统中各类数据描述的集合，是进行详细的数据收集和数据分析所获得的主要成果。
- 数据字典在数据库设计中占有很重要的地位。



## §2.3 数据字典

- 数据字典通常包括：

- 1. 数据项

- ⑩数据项是不可再分割的数据单位。

- 2. 数据结构

- ⑩数据结构反映了数据之间的组合关系。

- 3. 数据流

- ⑩数据流是数据结构在系统内传输的路径。

- 4. 数据储存

- ⑩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。

- 5. 处理过程

- ⑩处理过程的具体处理逻辑一般用判定表或判定树来描述。

## §2.3 数据字典

- 元数据
  - 关于数据库中数据的描述
  - 元数据不是用户应用中的数据
- 需求分析中应该注意的问题
  - 设计的可修改性、可扩充性
  - 用户的参与
- 需求分析的结果
  - 由设计人员（需求人员）和用户共同认可的需求文档



# §3 概念结构设计

- 概念结构设计
- 概念结构
- 概念结构设计的方法与步骤
- 数据抽象与局部视图设计
- 视图的集成

## §3 概念结构设计

- 概念结构设计

- 将需求分析得到的用户需求抽象为信息结构即概念模型的过程就是概念结构设计。它是整个数据库设计的关键。
- 独立于数据模型（层次、网状、关系），更与采用的DBMS无关。



## §3.1 概念结构

- 概念结构的主要特点是：
  - (1) 能真实、充分的反映现实世界，包括事物和事物之间的联系，能满足用户对数据的处理要求。
  - (2) 易于理解，从而可以用它和不熟悉计算机的用户交换意见，用户的积极参与是数据库的设计成功的关键。
  - (3) 易于更改，当应用环境和应用要求改变时，容易对概念模型修改和扩充。
  - (4) 易于向关系、网状、层状等各种数据模型转换。

# §3.1 概念结构

➤描述概念结构的工具

⑩E-R图



## §3.2 概念结构设计的方法与步骤

- 设计概念结构通常有四类方法：
  - 自顶向下
  - 自底向上
  - 逐步扩张
  - 混合策略



## §3.3 数据抽象与局部视图设计

- 概念结构是对现实世界的一种抽象。所谓抽象是对人、物、事和概念进行人为处理，抽取所关心的共同特性，忽略非本质的细节，并把这些特性用各种概念精确的加以描述，这些概念组成了某种模型。



## §3.3 数据抽象与局部视图设计

- 一般有三种抽象：

- 分类

- ⑩ 定义某一类概念作为现实世界中一组对象的类型。

- 聚集

- ⑩ 定义某一类型的组成成分。

- 概括

- ⑩ 定义类型之间的一种子集联系。

## §3.3 数据抽象与局部视图设计

- 设计分E-R图具体做法是：
  - 1. 选择局部应用
  - 2. 逐一设计分E-R图

## §3.3 数据抽象与局部视图设计

- 为了简化ER图的处置，应尽量将现实世界的事物作为属性对待。
- 划分实体与属性的准则
  - 属性不具备需要再描述的性质，是不可再分的数据项。
  - 属性不能与其它实体有联系。

## §3.3 数据抽象与局部视图设计

- 例

➤设计选课管理的数据库，包含以下信息：

⑩课程名称、课程代号、课程类型（必修、选修、任选）、学分、任课教师姓名、教师编号、教师职称、教师所属院名称、教师所属院代号、教师所授课程、学生姓名、学生学号、学生性别、学生所属院名称、学生所属院代号、学生所选课程。

## §3.3 数据抽象与局部视图设计

- 例（续）

- 其中语义有：

- ⑩每个学院有唯一代号，教师、学生仅有唯一编号，仅属于一个学院，一门课程可以由多个老师同时教授，一名教师可以教授多门课程，一名学生可以选修多门课程，学生选修课程时应同时选择课程和老师，并取得成绩。

- 设计E-R图。

# §3.3 数据抽象与局部视图设计-- next

- 例（续）

- 首先进行信息分类，找出实体

- ⑩ 课程

- ∞ 课程名称、课程代号、课程类型（必修、选修、任选）、学分、

- ⑩ 教师

- ∞ 任课教师姓名、教师编号、教师职称、教师所属学院名称、教师所属学院代号、

- ⑩ 学生

- ∞ 学生姓名、学生学号、学生性别、学生所属学院名称、学生所属学院代号。

## §3.3 数据抽象与局部视图设计

- 例（续）

- 首先进行信息分类，找出实体（续）

- ⑩ 学生所在学院

- ∞ 学院名称、学院代号

- ⑩ 教师所在学院

- ∞ 学院名称、学院代号

- 找出实体间的联系

- ⑩ 学院与教师；学院与学生；教师与课程；  
学生与课程

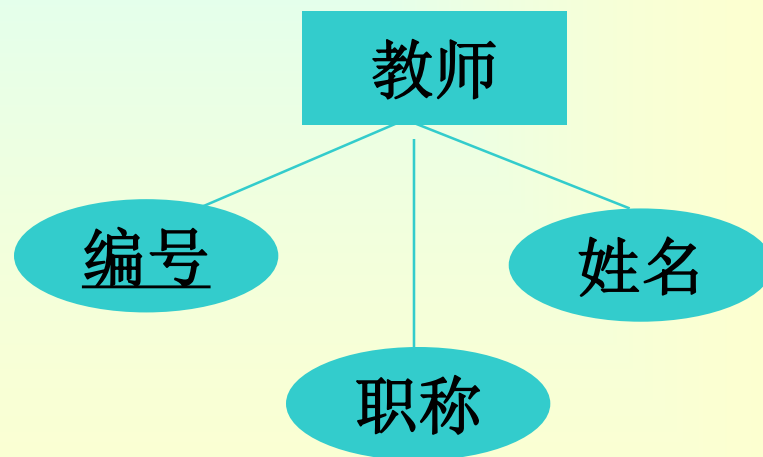
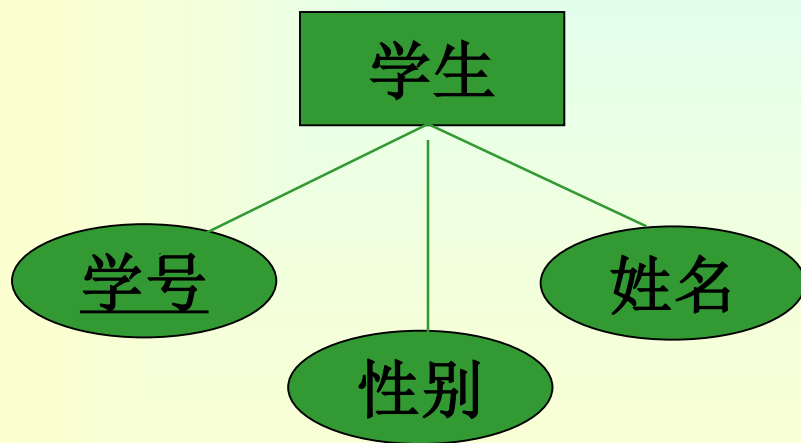
- ∞ 注意学生选的课是某教师教授的课程

## §3.3 数据抽象与局部视图设计

- 例（续）

- E-R图

- ⑩ 首先画出各个实体



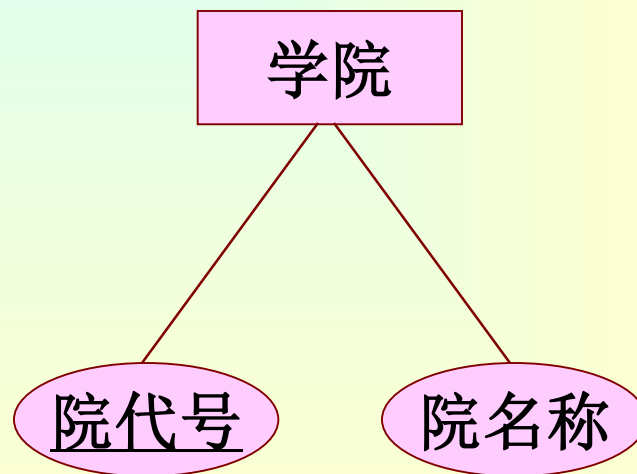
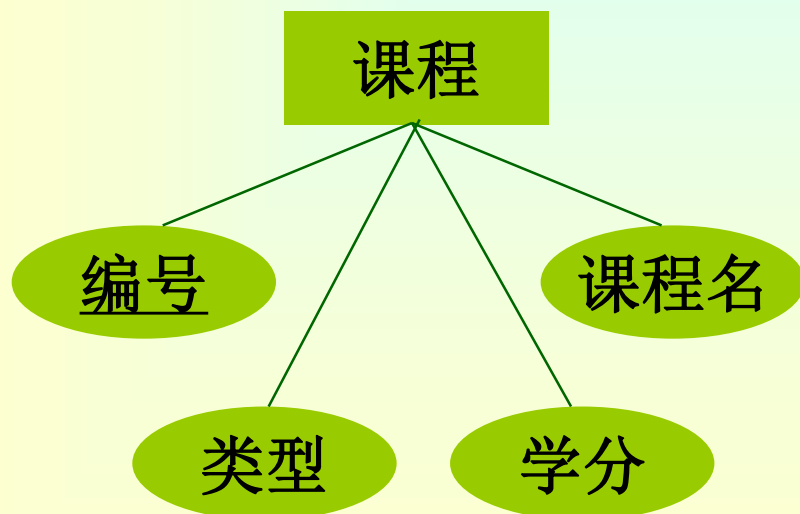


## §3.3 数据抽象与局部视图设计

- 例（续）

- E-R图（续）

- ⑩ 首先画出各个实体（续）



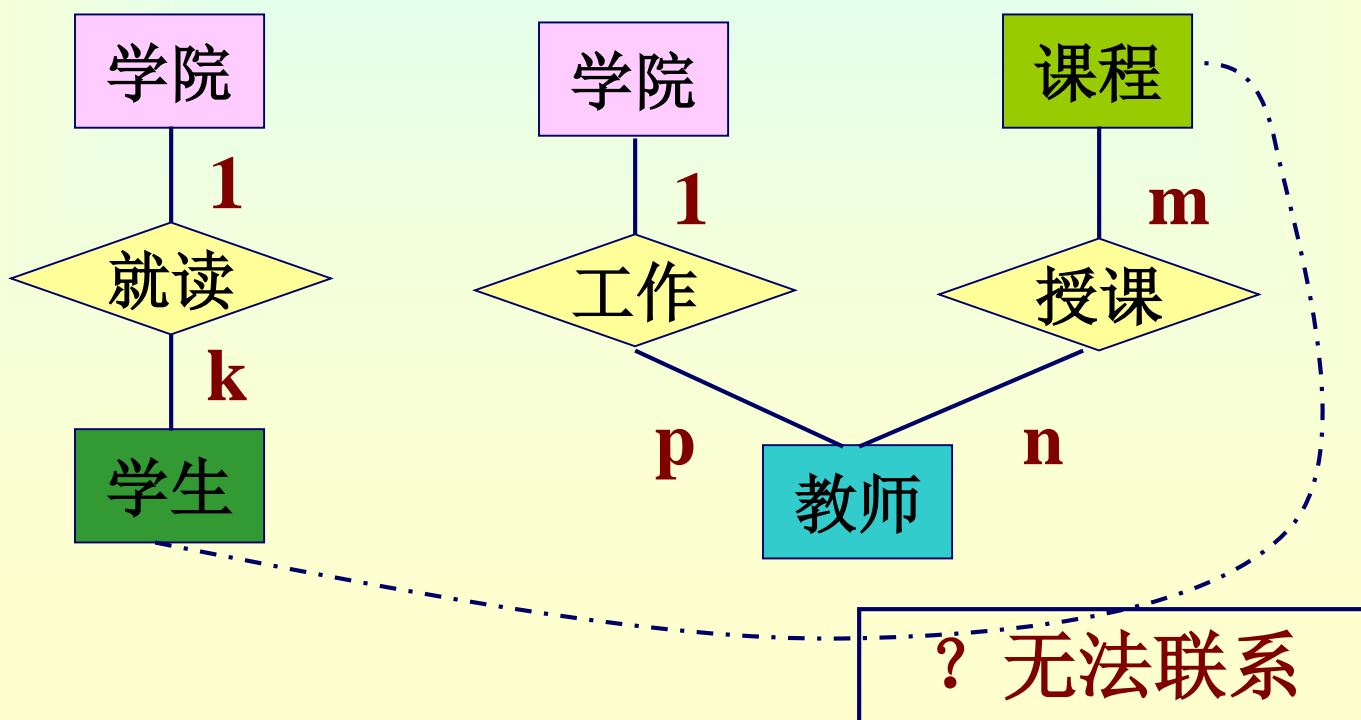
## §3.3 数据抽象与局部视图设计

- 例（续）

- E-R图（续）

⑩再画出各个实体间的联系（忽略了属性）

∞该图存在疑问和不明确的地方



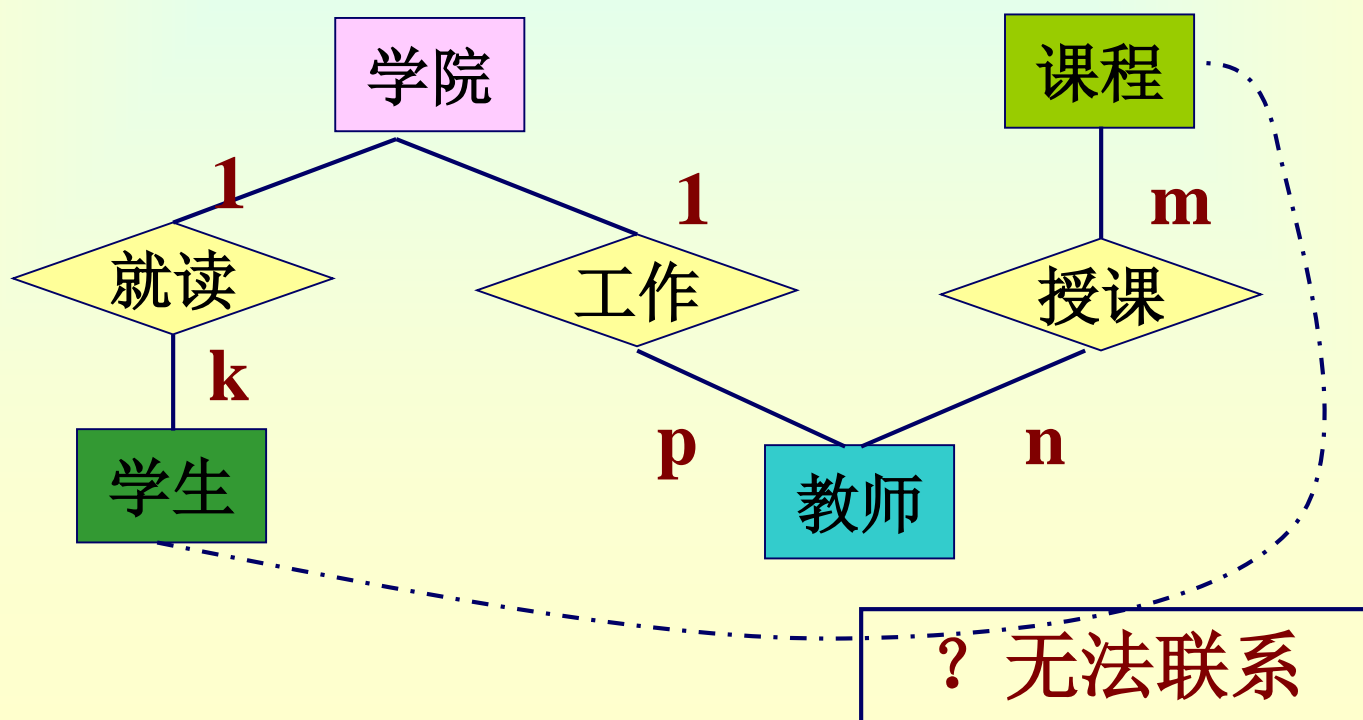
## §3.3 数据抽象与局部视图设计

- 例（续）

- E-R图（续）

⑩再画出各个实体间的联系（忽略了属性）

∞该图存在疑问和不明确的地方



## §3.4 视图的集成

- 各子系统的分E-R图设计好以后，下一步就是要将所有的分E-R图综合成一个系统的总E-R图。
- 一般来说，视图集成可以有两种方式：
  - ⑩ 多个分E-R图一次集成
  - ⑩ 逐步集成，用累加的方式一次集成两个分E-R图

## §3.4 视图的集成

- 每次集成个分E-图时都需要分两步走
  - (1) 合并
    - ⑩解决各分E-R图之间的冲突，将各分E-R合成初步E-R图。
  - (2) 修改和重构
    - ⑩消除不必要的冗余，生成基本E-R图。

## §3.4 视图的集成

- 一、合并E-R图，生成初步E-R图

⑩ 各分E-R图之间的冲突主要有三类：属性冲突、命名冲突、结构冲突。

➤ 属性冲突

⑩ (1) 属性域冲突，即属性值的类型、取值范围或取值集合不同。

⑩ (2) 属性取值单位冲突

## §3.4 视图的集成

- 一、合并E-R图，生成初步E-R图（续）

- 命名冲突

- ⑩（1）同名异义

- ⑩（2）异名同义

- 结构冲突

- ⑩（1）同一对象在不同应用中具有不同的抽象。

- ⑩（2）同一实体在不同分E-R图中包含的属性个数和属性排列次序不完全相同。

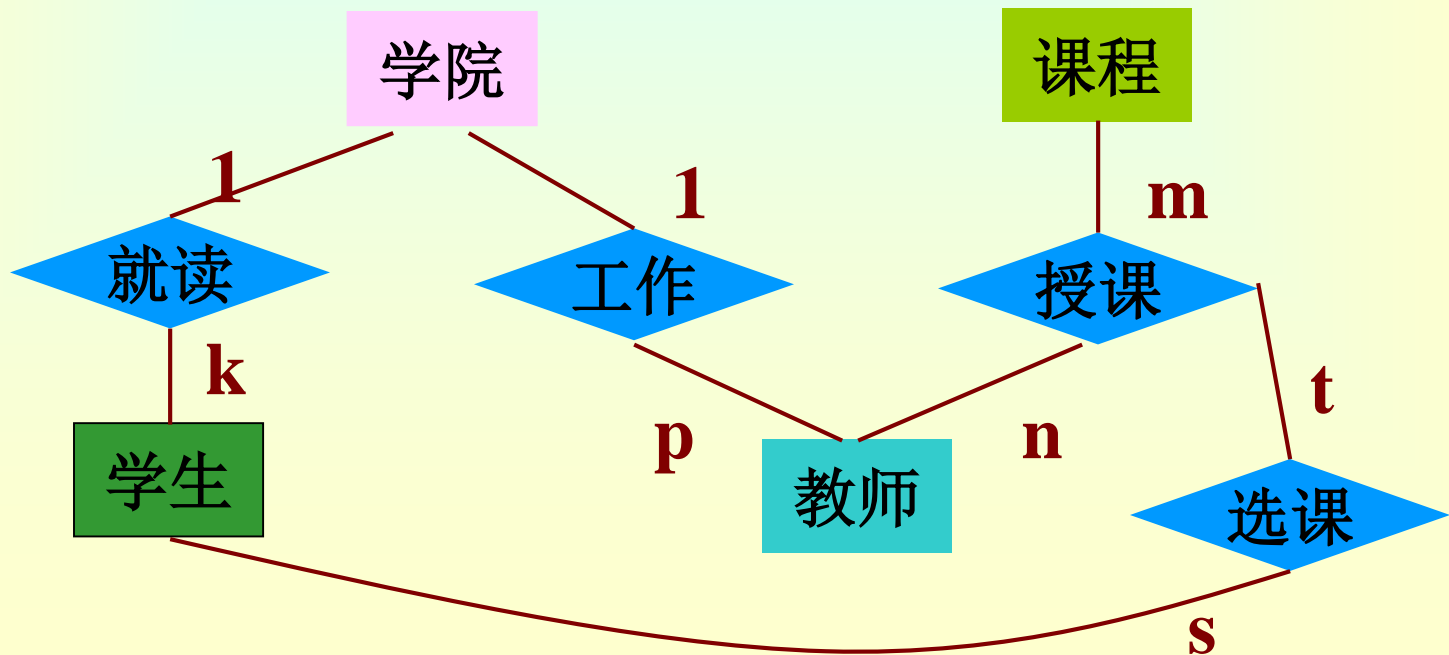
- ✎解决方案：取属性的并集。

## §3.4 视图的集成

- 例

- 合并E-R图

⑩ 某学生选的是某老师上的某课程，因此可以讲授课联系看作实体





## §3.4 视图的集成

- 二、消除不必要的冗余，设计基本E-R图
  - 在初步E-R图中，可能存在一些冗余的数据和实体间冗余的联系。冗余数据和冗余联系容易破坏数据库的完整性，给数据库的维护增加困难，因当消除。消除了冗余的初步E-R图称为基本E-R图。
  - 消除冗余主要采用分析方法，即以数据字典和数据流图为依据，根据数据字典中关于数据项之间逻辑关系的说明来消除冗余。

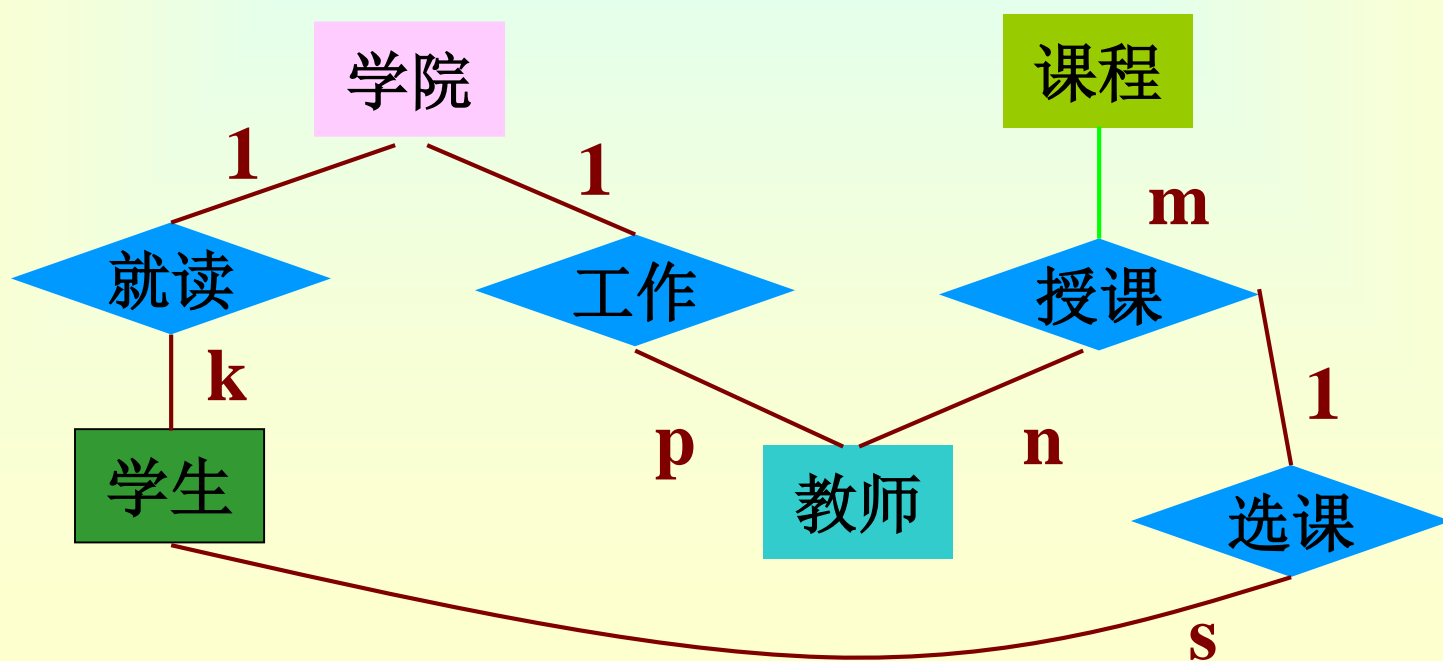
## §3.4 视图的集成

- 二、消除不必要的冗余，设计基本E-R图
  - 除分析方法外，还可以用规范化理论来消除冗余。具体方法如下：
    - ⑩ 确定分E-R图实体之间的数据依赖 $F_L$ 。
    - ⑩ 求 $F_L$ 的最小覆盖 $G_L$ ，差集为 $D = F_L - G_L$ 。  
逐一考察 $D$ 中的函数依赖，确定是否是冗余的联系，若是，就把它去掉。

## §3.4 视图的集成

- 例

- 合并E-R图



# §4 逻辑结构设计

- 逻辑结构设计
- E-R图向关系模型的转换
- 数据模型的优化
- 设计用户子模式
- 评价数据模型

## §4 逻辑结构设计

- 逻辑结构设计

- 就是把概念结构设计阶段设计好的基本E-R图转换为与选用DBMS产品所支持的数据模型相符合的逻辑结构。

- ⑩概念结构是独立于任何一种数据模型的信息结构。

- ⑩逻辑结构设计是与具体的DBMS相关的。

- ⑩实际情况中往往由用户来选择DBMS。

## §4 逻辑结构设计

- 设计逻辑结构时一般要分三步进行：
  - 1. 将概念结构转换为一般的关系、网状、层状模型；
  - 2. 将转换来的关系、网状、层状模型向特定DBMS支持下的数据模型转换；
  - 3. 对数据模型进行优化；



## §4.1 E-R图向关系模型的转换

- E-R图向关系模型的转换要解决的问题是  
如何将实体和实体间的联系转换为关系  
模式，如何确定这些关系模式的属和码。

## §4.1 E-R图向关系模型的转换

➤ 将实体和实体间的联系转换为关系模式，这种转换一般遵循如原则：

- ⑩ (1) 一个实体型转换为一个关系模式。
- ⑩ (2) 一个1: 1联系可以转为一个独立的关系模式。
- ⑩ (3) 一个1: n联系可以转为一个独立的关系模式。
- ⑩ (4) 一个m: n联系转换为一个关系模式。
- ⑩ (5) 三个或三个以上实体间的一个多元联系可以转换为一个关系模式。
- ⑩ (6) 具有相同码的关系模式可合并。



# §4.1 E-R图向关系模型的转换

- 例

- E-R图向关系模型进行转换

- ⑩ 学生 (学号, 姓名, 性别)
- ⑩ 就读 (学号, 所在学院代号)
- ⑩ 教师 (教师编号, 教师姓名, 职称)
- ⑩ 工作 (教师编号, 所在学院代号)
- ⑩ 课程 (课程编号, 课程名称, 课程类型, 学分)
- ⑩ 学院 (学院名称, 学院代号)
- ⑩ 授课 (教师编号, 课程编号)
- ⑩ 选课 (学号, 教师编号, 课程编号, 成绩)



## §4.2 数据模型的优化

- 数据模型的优化

- ⑩1. 确定数据依赖

- ⑩2. 对于各个关系模式之间的数据依赖进行极小化处理，消除冗余的联系。

- ✎取消多余的属性

- ⑩3. 按照数据依赖的理论对关系模式逐一进行分析，考察是否存在部分函数依赖、传递函数依赖、多值依赖等，确定各关系模式分别属于第几范式。

- ✎并非规范化程度越高越好，一般的，设计到3NF或BCNF。

## §4.2 数据模型的优化

- 数据模型的优化（续）

⑩4. 按照需求分析阶段得到的处理要求，分析这些模式对于这样的应用环境是否合适确定是否要对某些模式进行合并或分解。

- ⌘根据空间或时间需要可能将表分拆或合并（合并会降低范式）。

⑩5. 对关系模式进行必要分解，提高数据库操作的效率和储存空间的利用率。

- ⌘根据80/20原则水平分解以提高效率

- ⌘水平分解可以实现数据分布

- ⌘垂直分解要保持无损连接性和保持函数依赖

## §4.2 数据模型的优化

- 例

- 数据模型的优化

- ⑩ 学生 (学号, 姓名, 性别, 所在院代号)

- ⑩ 教师 (教师编号、教师姓名, 所在院代号)

- ⑩ 课程 (课程编号, 课程名称, 学分)

- ⑩ 学院 (学院名称, 学院代号)

- ⑩ 授课 (教师编号, 课程编号)

- ∞ 可以增加“教室”等属性。

- ⑩ 选课 (学号, 教师编号, 课程编号, 成绩)

## §4.2 数据模型的优化

- 例

- 写出函数依赖

- ⑩ 学院代号 → 学院名称

- ⑩ 学号 → 姓名、性别、出生日期、学院代号

- ⑩ 教师编号 → 姓名、性别、学院代号

- ⑩ 课程编号 → 课程名称、课程类型学分)

- ⑩ (教师编号、课程编号) → 教师编号、课程编号

- ⑩ (学号、教师编号、课程编号) → 成绩



## §4.3 设计用户子模式

- 定义数据库全局模式主要是从系统的时间效率、空间效率、易维护等角度出发。
- 由于用户子模式与模式是相对独立的，因此在定义用户子模式是时注重考虑用户的方便与习惯。包括：
  - ⑩（1）使用更符合用户习惯的别名。
  - ⑩（2）可以对不同级别的用户定义不同的View以保证系统的安全性。
  - ⑩（3）简化用户对系统的使用。



## §4.4 评价数据模型

- 如何评价一个数据模型？

- ⑩ 一个好的模型应该具有最小定义、完备定义及适应性。另外，应该容易实现。

- **最小定义：**

- ⑩ 用尽可能少的实体描述尽可能多的对象。

- ⑩ 任何数据和信息应该且只能用一种方法进行定义，以便尽可能地消除冗余：消除冗余的实体和冗余的关系。

- ⑩ 规范化理论为消除冗余提供了强有力的手段。

## §4.4 评价数据模型

- 如何评价一个数据模型？（继续）

- 完备定义

- ⑩ 现实世界中所有感兴趣的概念和对象是否都已经定义在模型之中，如果没有，系统的某些功能必将无法实现。

- 适应性

- ⑩ 当系统的功能或需求发生变化时，是否能够做最小的修改以适应新的要求，降低维护费用。





# §5 数据库的物理设计

- 数据库的物理设计
- 数据库物理设计的内容和方法
- 关系模式存取方法选择
- 确定数据库的存储结构
- 评价物理结构
- 物理设计的必要工作

# §5 数据库的物理设计

- 数据库的物理设计

- 为一个给定的逻辑数据模型选取一个最适合应用要求的物理结构的过程。

- 数据库物理设计的步骤:

- (1) 确定数据库的物理结构，在关系数据库中主要指存取方法和储存结构。

- (2) 对物理结构进行评价，评价的重点是时间和空间效率。



# §5.1 数据库物理设计的内容方法

## ➤ 要根据应用环境来决定设计方案

### ⑩ 决策支持系统

∞ 较少空间，较多索引，不强调速度

### ⑩ 在线事务处理

∞ 强调速度，适当索引

### ⑩ 读型（主要读）

∞ 强调速度，可以有冗余

### ⑩ 写型（主要写）

∞ 冗余小，数据备份

# §5.1 数据库物理设计的内容方法

➤ 要根据DBMS决定设计方案

⑩ 存取方法

⑩ 存储结构

# §5.1 数据库物理设计的内容方法

➤ 对于数据库查询事务，需要得到如下信息

- ⑩ 查询的关系
- ⑩ 查询条件所涉及属性
- ⑩ 连接条件所涉及属性
- ⑩ 查询的投影属性

# §5.1 数据库物理设计的内容方法

➤ 对于数据的更新，需要得到如下信息

⑩ 被更新的关系

⑩ 每一关系上的更新操作条件所涉及的属性

⑩ 修改操作要改变的属性值

# §5.1 数据库物理设计的内容方法

➤ 通常对于关系数据库物理设计的内容主要包括：

⑩ 为关系模式选择存取方法

⑩ 设计关系、索引等数据库文件的物理储存结构



## §5.2 关系模式存取方法选择

- 数据库物理设计的任务之一就是要确定选择哪些存取方法，即建立哪些存取路径。
- 存取方法是快速存取数据库中数据的技术，主要的方法有三类：
  - ⑩ 索引方法
    - ∞ 主要是B+树。
  - ⑩ 聚簇（Cluster）方法
  - ⑩ HASH方法



## §5.2 关系模式存取方法选择

- 存取方法的选择

- 所谓选择索引存取方法实际上就是根据应用要求确定对关系的哪些属性列建立索引、哪些属性列建立组合索引、哪些索引要设计为唯一索引等。

- 常建立索引的属性（组）

- ⑩ 经常查询的属性（组）

- ⑩ 经常在聚集函数中使用的属性（组）

- ⑩ 在连接条件中出现的属性（组）

- ∞ 索引不是越多越好，DBMS必须为索引的维护付出代价

## §5.2 关系模式存取方法选择

- 聚簇存取方法的选择

- 聚簇

- ⑩ 为了提高某个属性的查询速度，把这个或这些属性上具有相同值的元组集中存放在连续的物理块称为聚簇。
- ⑩ 聚簇功能可以用于经常连接的多个关系，形成“聚簇码”，以提高连接速度。

- 聚簇索引

- ⑩ 强迫系统把这个或这些属性上具有相同值的元组集中存放在连续的物理块且整体按升序或降序排列的索引。

## §5.2 关系模式存取方法选择

- 聚簇存取方法的选择

- 设计候选聚簇

- ⑩（1）对经常在一起进行连接操作的关系可以建立聚簇
- ⑩（2）如果一个关系的一个属性经常出现在相等比较条件中，则单个关系可建立聚簇
- ⑩（3）如果一个关系的一个属性上的值重复率很高，则此单个关系可建立聚簇。

## §5.2 关系模式存取方法选择

- 聚簇存取方法的选择

- 检查候选聚族中的关系，取消其中不必要的的关系：

- ⑩ (1) 从聚簇中删除经常进行全表扫描的关系
    - ⑩ (2) 从聚簇中删除更改新操作远多于连接操作的关系
    - ⑩ (3) 不同的聚簇中可能包含相同的关系，一个关系可以在某一个聚簇中但不能同时加入多个聚簇

## §5.2 关系模式存取方法选择

- 聚簇存取方法的选择

- 聚簇适用的语句

- ∞ 主要是排序类

- ⑩ ORDER BY

- ⑩ GROUP BY

- ⑩ UNION

- ⑩ DISTINCT

## §5.2 关系模式存取方法选择

- HASH存取方法的选择

- 选择HASH存取方法的规则如下：

- 如果一个关系属性主要出现在等连接条件中或主要出现在相等比较选择条件中，而且满足下列两个条件之一，则此关系可以选择HASH存取方法：

- ⑩（1）如果一个关系的大小可预知，而且不变。

- ⑩（2）如果关系的大小动态改变，而且数据库管理系统提供了动态HASH存取方法。

## §5.3 确定数据库的存储结构

- 确定数据库物理结构
  - 主要指确定数据的存放位置和存储结构, 包括确定关系、索引、聚簇、日志、备份等的储存安排和存储结构; 确定系统配置等。
  - 主要考虑的因素有存取时间、存储空间利用率、维护代价、数据可恢复性。这几方面常常相互矛盾, 需要折中选择。

## §5.3 确定数据库的存储结构

- 确定数据的存放位置
  - 磁盘的数量和容量
    - ⑩ 磁盘阵列
  - 数据是否镜像
  - 将表分割存放在不同的磁盘上
  - 多个日志的分开存放
  - 表和索引分别放在不同的磁盘
  - 磁带或磁盘备份



## §5.3 确定数据库的存储结构

- 确定系统配置

- 内存
- 同时连接用户数
- 数据缓冲区（大小、个数）
- 磁盘物理块（大小）
- 数据填装因子
- 时间片
- 数据库大小
- 锁规模、层次



## §5.4 评价物理结构

- 评价物理结构

- 数据库物理设计过程中需要对时间效率、时空效率、维护代价和各种用户要求进行权衡，其结果可以产生多种方案，数据库设计人员必须对这些方案进行细致的评价，从中选出一个较优的方案为数据库的物理结构。
- 评价数据库物理结构的方法完全依赖于所选用的DBMS。



## §5.5 物理设计的必要工作

- 物理设计的必要工作
  - 物理设计的工作除了要实现实现概念模型向DBMS的转化的任务，同时加入概念模型中未考虑或未全部考虑的因素：PK（主键），FK（外键），View（视图），Index（索引），Trigger（触发器），Stored Procedure（存储过程）等等，以实现系统的完整性。



# §6 数据库的实施和维护

- 数据库的实施和维护
- 数据的载入和应用程序的调试
- 数据库的试运行
- 数据库的运行和维护

# §6 数据库的实施和维护

- 数据库的实施

- 完成数据库的物理设计之后，设计人员就要用RDBMS提供的数据库定义语言和其它实用程序将数据逻辑设计和物理设计结果严格描述出来，成为DBMS可以接受的源代码，在经过调试产生目标模式。然后就可以组织数据入库，这就是数据库实施阶段。
- 目标代码可能存在与数据库服务器端，也可能存在与客户端。



# §6.1 数据载入和应用程序的调试

- 数据的载入和应用程序的调试

- ⑩数据库实施阶段包括两项重要的工作，  
一项是数据库的载入，另一项是应用程序  
的编码和调试。

- 数据库的载入

- ⑩录入或从旧系统转换而来

- 应用程序的编码和调试

- ⑩应与数据库设计同时进行，结合已经载  
入的数据进行调试。



## §6.2 数据库的试运行

- 数据库的试运行

- 在原有系统的数据有一小部分已输入数据库后，就可以开始对数据库系统进行联合调试，这又称之为数据库的试运行。

- ⑩ 测试应用程序功能是否满足设计要求

- ⑩ 测试系统性能是否与设计目标相符

- 试运行期间的注意事项

- ⑩ 分批组织数据入库，防止修改设计时需要大量修正数据或是数据作废

- ⑩ 做好系统的备份工作



## §6.3 数据库的运行和维护

➤在数据库运行阶段，对数据库经常性的维护工作主要是由DBA来完成。

⑩数据库的转储和恢复

⑩数据库的安全性、完整性控制

⑩数据库性能的监督、分析和改造

    ∞性能调整或数据库重构

⑩数据库的重组与重构造

    ∞修改模式或外模式

➤当系统发生重大变化，重构也无济于事，则数据库系统生命周期结束。



## §6.3 数据库的运行和维护

### ➤ 维护工作的分类

- ⑩ 1. 改正性维护;
- ⑩ 2. 适应性维护;
- ⑩ 3. 完善性维护;
- ⑩ 4. 预防性维护。



## §7 小结

- 数据库的设计过程
  - 各阶段的任务和步骤
- E-R图
  - 能够给出应用系统的E-R图，并转化为关系模型
- 对数据模型的评价