

(二) 关系数据库及 数据库系统设计

本节主要内容

- 数据库结构概述
- 关系模型、关系
- E-R模型
- 数据模型的基本概念
- 数据库设计方法概述
- 需求分析、概念设计、实现设计、实施与维护

数据库系统概况

数据库系统是由计算机软、硬件资源组成的系统，它实现了有组织地、动态地存储大量关联数据，方便多用户访问。通俗地讲，数据库系统可把日常的一些表格、卡片等的数据库系统有组织地集合在一起，输入到计算机，然后通过计算机处理，再按一定要求输出结果。所以，对于数据库来说，主要解决三个问题：

- 1.有效地组织数据，即对数据进行合理设计，以便计算机存取；
- 2.方便地将数据输入到计算机中；
- 3.根据用户的要求将数据从计算机中抽取出来。

数据模型

- **作用**：对现实世界进行抽象。找到所关心的数据，并将其组织起来，使之可以存放到数据库中。
- **模型**：是描述数据的概念集合。按应用的不同目的分为两类：概念模型、数据模型。
- **概念模型**：也称信息模型，它按用户的观点来数据和信息进行建模，用于信息世界，它强调语义表达能力，是现实世界到信息世界的第一次抽象，是用户和数据库设计人员之间进行交流的语言。

关系型数据库理论

什么是关系？

关系数据库是用表来表示数据的数据库，“关系”只是表的数学术语。

使用“关系”的原因如下：

- 关系系统基于关系模型，关系模型基于数学方面的数据抽象理论
- 关系模型理论是IBM研究员、数学家E.F.Codd在1969年提出的将数学理论和严密的逻辑应用到数据管理领域
- 同样，在其它领域，如人工智能、自然语言处理和硬

关系型数据库理论

关系模型

关系系统基于关系理论，即关系数据模型。体现在如下方面：

- 结构化方面：数据库中的数据对用户来说是表，并且只是表；
- 完整性方面：数据库中的这些表满足一定的完整性约束；
- 操纵性方面：用户可以使用用于表操作的操作符。

关系型数据库理论

关系模型包括：

- 一个可扩展的标量类型的集合；
- 关系类型生成器的对应这些关系类型的解释器；
- 实用程序，用于定义生成关系类型的关系变量；
- 向关系变量赋关系值的关系赋值操作；
- 从其他关系值中产生关系值的、可扩充的关系操作符集合。

实体－关系模型概述

- 为了面向用户提供更丰富、更具表达能力的概念和技术，以便在更高层次上对数据对象的性质和联系作出清晰和简明的描述，20世纪70年代，人们做了大量的研究工作，并建立的语义数据模型。
- P.P.Chen1976年提出了E-R模型是第一个出现的语义模型，不受任何DBMS约束的面向用户的表达机制。

实体关系模型的基本概念

现实世界中的事物在人们头脑中反映的信息世界是用文字和符号记载下来的，描述事物用以下术语。

(1) 实体(Entity)

客观事物在信息世界中称为实体。实体可以是具体的，如一个学生，一本书，也可以是抽象的事件，如一场足球比赛。实体用类型(Type)和值(Value)表示，例如学生是一个实体，而具体的学生李明、王力是实体值。

(2) 实体集(Entity Set)

性质相同的同类实体的集合称为实体集。如一班学生，一批书籍。

实体关系模型的基本概念

(3) 属性(Attribute)

实体有许多特性，每一特性在信息世界中都称为属性。每个属性都有一个值，值的类型可以是整数、实数或字符型，例如学生的姓名、年龄都是学生这个实体的属性，姓名的类型为字符型，年龄的类型为整数。

属性用类型和值表示，例如学号、姓名、年龄是属性的类型，而具体的数值870101、王小艳、19是属性值。

实体关系模型的基本概念

实体关系图（E-R图）的表示方法：

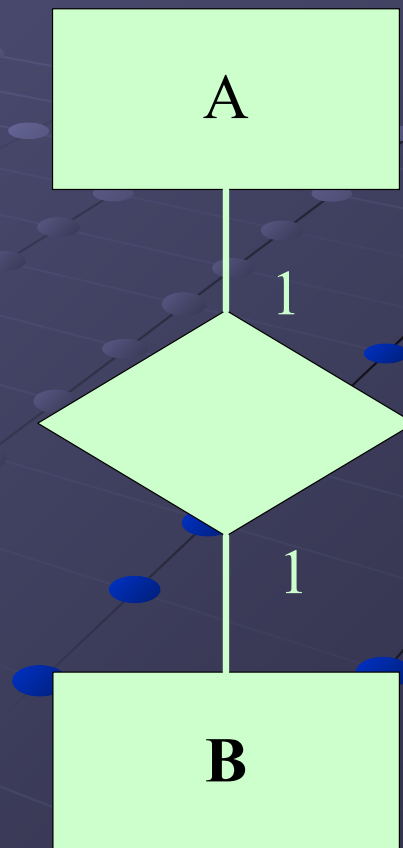
- (1) 实体集用矩形框表示，矩形框中写上实体的名称
- (2) 实体的属性用圆或椭圆表示，圆内写上属性名，并用连线与实体相连
- (3) 实体事件的联系用菱形框表示，联系用适当的含义命名，名称写在菱形中。并在连线上表明联系的类型，即1:1，1:N，M:N。

实体关系模型的基本概念

(4) 实体联系

i. 一对一联系(1:1)

如果A中的任一属性至多对应B中的一属性，且B中的任一属性至多对应A中的一属性，则称A与B是一对一联系，如图所示。例如电影院中观众与座位之间、乘客旅客与车票之间、病人与病床之间都是一对一联系。

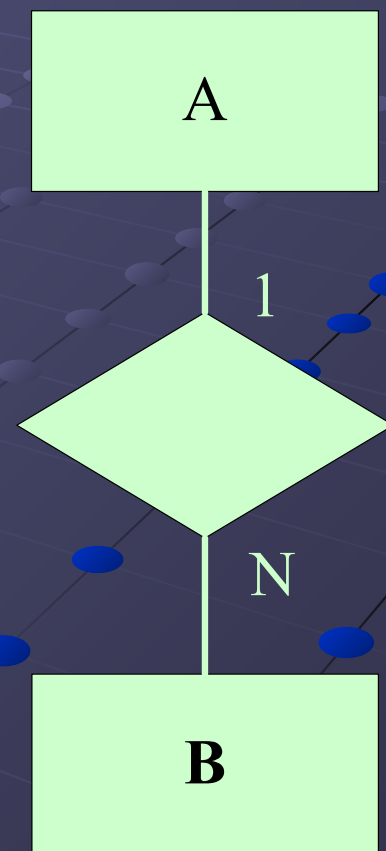


实体关系模型的基本概念

(4) 实体联系

ii. 一对多联系(1:N)

如果A中至少有一个属性对应B中一个以上属性，且B中任一属性至少对应A中一个属性，则称A与B是一对多联系，如图所示。例如，学校对系，班级对学生等都是—对多联系。

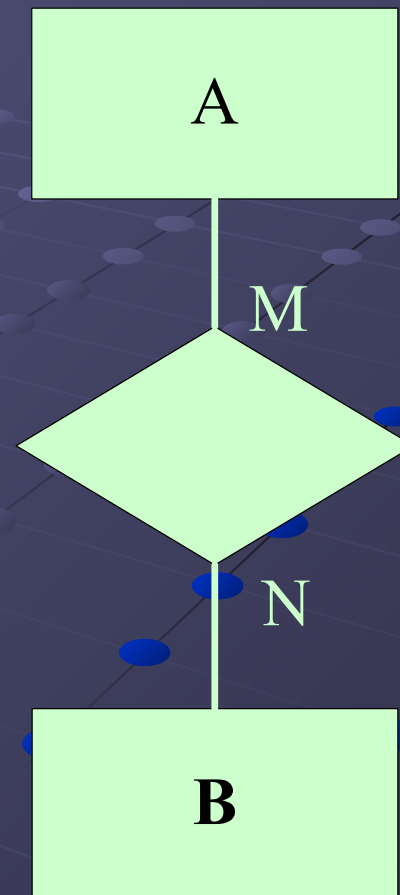


实体关系模型的基本概念

(4) 实体联系

iii. 多对多联系(M:N)

如果A中至少有一个属性对应B中一个以上属性，且B中也至少有一个属性对应A中一个以上属性，则称A与B是多对多联系，如图所示。例如，学生与课程、工厂与产品、商店与顾客都是多对多联系



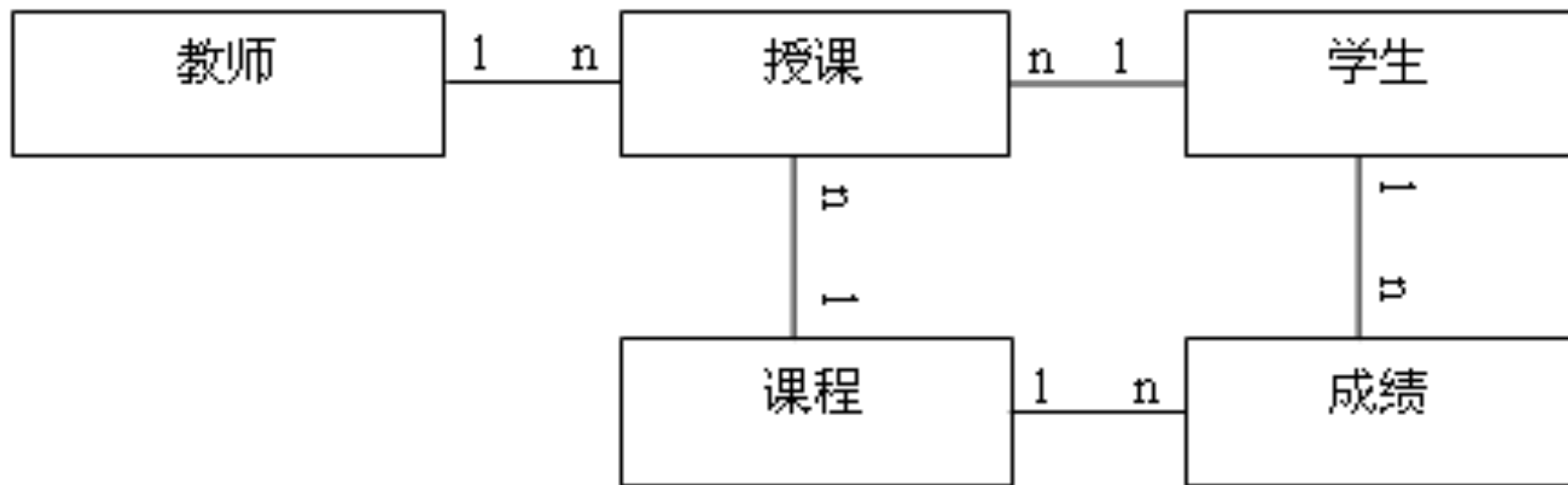
实体关系模型的基本概念

(5) 实体模型(Entity Model)

反映实体之间联系的模型称为实体模型。

教学情况可由学生、课程、教师、授课、成绩等方面情况组成。其中，学生具有属性学号、姓名、年龄、性别(当然还有其它属性，这里省略)，课程具有属性课号、课程名称，授课具有属性教师姓名、课号、课时、班级，成绩具有属性学号、课号、分数。

学生对课程是多对多联系，因为一个学生可以学习多门课程，而一门课程又有多个学生学习。教师对课程假设是一对多联系，即一个教师可以讲授多门课程，但一门课程只能由一个教师任教。教学情况的实体模型如图所示。



数据模型的基本概念

描述数据有以下术语：

(1) 字段(Field)

对应于信息世界中的属性，也称数据项。字段的命名往往与属性名相同。

(2) 记录(Record)

字段的有序集合称为记录，它用来描述一个实体，是相应于这一实体的数据。例如，组成一个学生记录的字段(数据项)有学号、姓名、年龄和性别字段，这是记录的类型。而870101王小艳18女就是一个记录的值。

数据模型的基本概念

(3) 文件(File)

同一类记录的集合，例如所有的学生记录的集合就是一个学生文件。

(4) 数据模型(Data Model)

实体模型的数据化。

(5) 关键字(Key)

能唯一标识文件中每一个记录的一个或多个字段的最小组合称为关键字。例如学生文件中，学号可以唯一地标识每个学生记录，所以学号是关键字。

数据模型的建立实际上是实体模型的数据化，为了使模型能清晰、准确地反映客观事物，并能用于数据库设计，一般应做如下工作：

- a. 给数据模型命名，使不同模型得以区别；
- b. 给每个数据项命名，以说明和区分每个记录类型具有的数据项，并确定作为记录类型主关键词的数据项；
- c. 指出数据项性质，即类型、长度、值域。

常见的数据模型有三种：**层次模型、网状模型和关系模型**，根据这三种数据模型建立的数据库分别为层次数据库、网状数据库和关系数据库。

自20世纪80年代以来，大多数数据库系统都是建立在关系模型之上。关系模型建立在严密的数学概念之上，它用“二维表格”来表示实体及实体之间的联系。数据是二维表中的元素，而二维表即表示关系。表格中的一行称为一个元组，相当于一个记录；表格中的一列称为一个属性，相当于记录中的一个字段。一个或若干个属性的集合称为关键词，它唯一标识一个元组。关系应满足：

- (1) 二维表格中每一列中的元素是类型相同的数据；
- (2) 行和列的顺序可以任意；
- (3) 表中元素是不可再分的最小数据项(描述对象属性的数据)；
- (4) 表中任意两行的记录不能完全相同，表中不允许有表。

关系中的每个记录是唯一的，所有记录具有相同个数和类型的字段，即所有记录具有相同的固定长度和格式。

下图是典型的关系模型。模型中有4个关系模式，为关系中的每个数据项指定了属性之后，就得到一个二维关系表。表1-1是一个学生关系表的示例。

学生关系

学号	姓名	年龄	性别	班级
----	----	----	----	----

课程关系

课号	课程名称
----	------

授课关系

老师姓名	课号	课时	班级
------	----	----	----

成绩关系

学号	课号	分数
----	----	----

表1-1 学生关系表

学号	姓名	年龄	性别	班级
870101	王小艳	18	女	微机871
870201	李 明	18	男	微机872
870102	司马奋进	19	男	微机871
870202	李 明	18	女	微机872
870103	成 功	18	男	微机871

关系模型的最大优点是简单，一个关系就是一个数据表格，用户容易掌握，只需用简单的查询语句就能对数据库进行操作。用关系模型设计的数据库系统是用查表方法查找数据。

数据库管理系统

数据库管理系统(DBMS)是在操作系统支持下运行的，它与操作系统之间的接口称为储存记录接口，DBMS借助于操作系统实现数据的储存管理。它与用户之间的接口称为用户接口，DBMS提供用户作用的数据库语言。

DBMS是在操作系统的支持下，把用户对数据库的操作从应用程序带到用户级、概念级，再导向物理级，最终实现对内存中数据的操作。DBMS应使数据能被各种不同的用户所共享，保证用户得到的数据是完整可靠的。

E-R模型的转换

E-R模型主要用于数据库逻辑设计，由于它独立于任何数据库系统，在实际的设计过程中必须将E-R模型转换为具体的数据库管理系统所支持的数据模型。

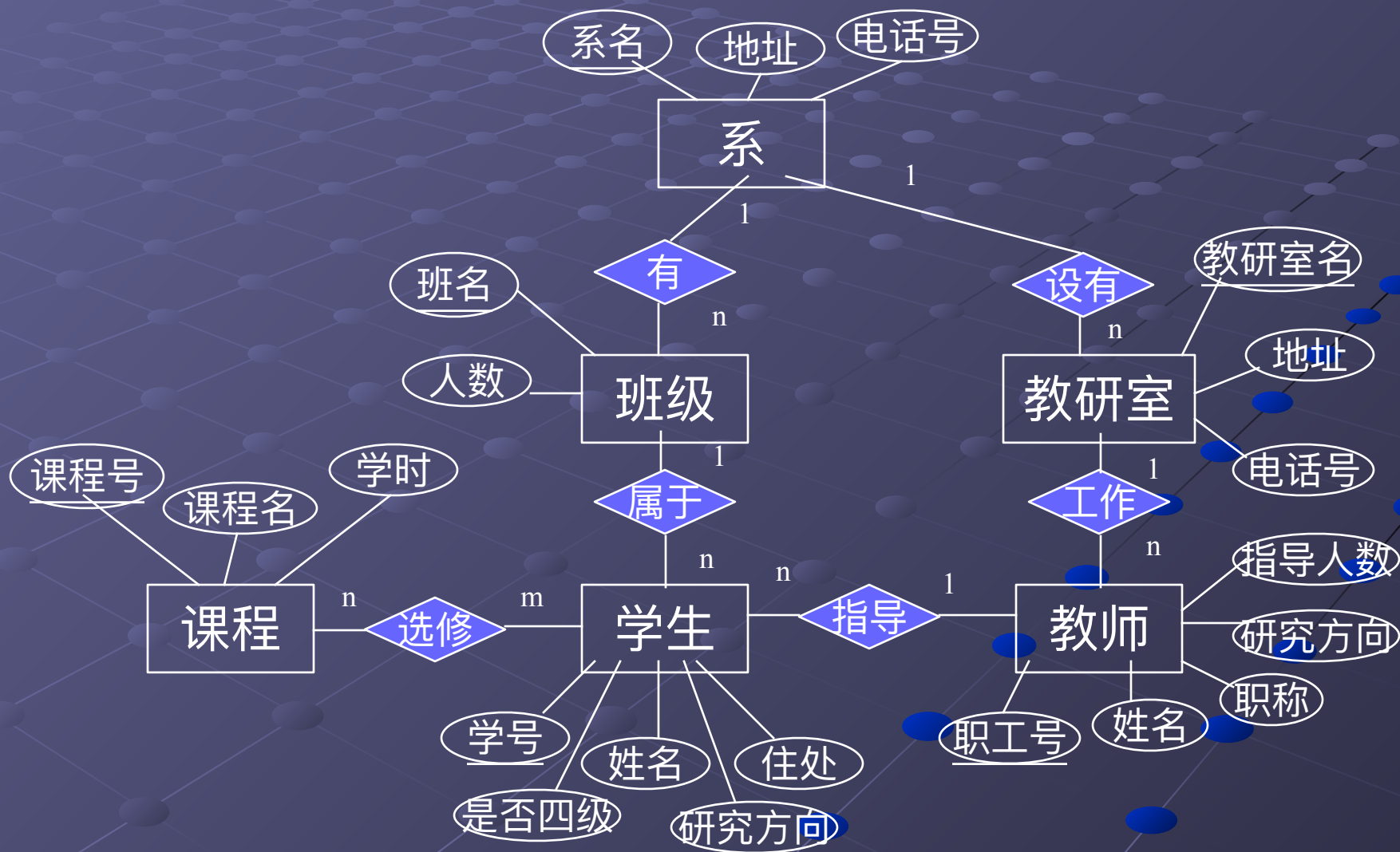
基本转换规则如下：

- (1) 把E-R图中所有的实体及其属性用关系模式表示；
- (2) 把实体之间的联系及与被联系的两个实体中的相关信息也用一个关系模式来表示。

实例分析

- 例子：开发学校信息管理系统。学校中有若干系，每个系有若干班级和教研室，每个教研室有若干教师，其中有教授和副教授每人各带若干名研究生，每个班有若干学生，每个学生选若干课程，每门可由若干学生选修。

实例分析 (E-R图)



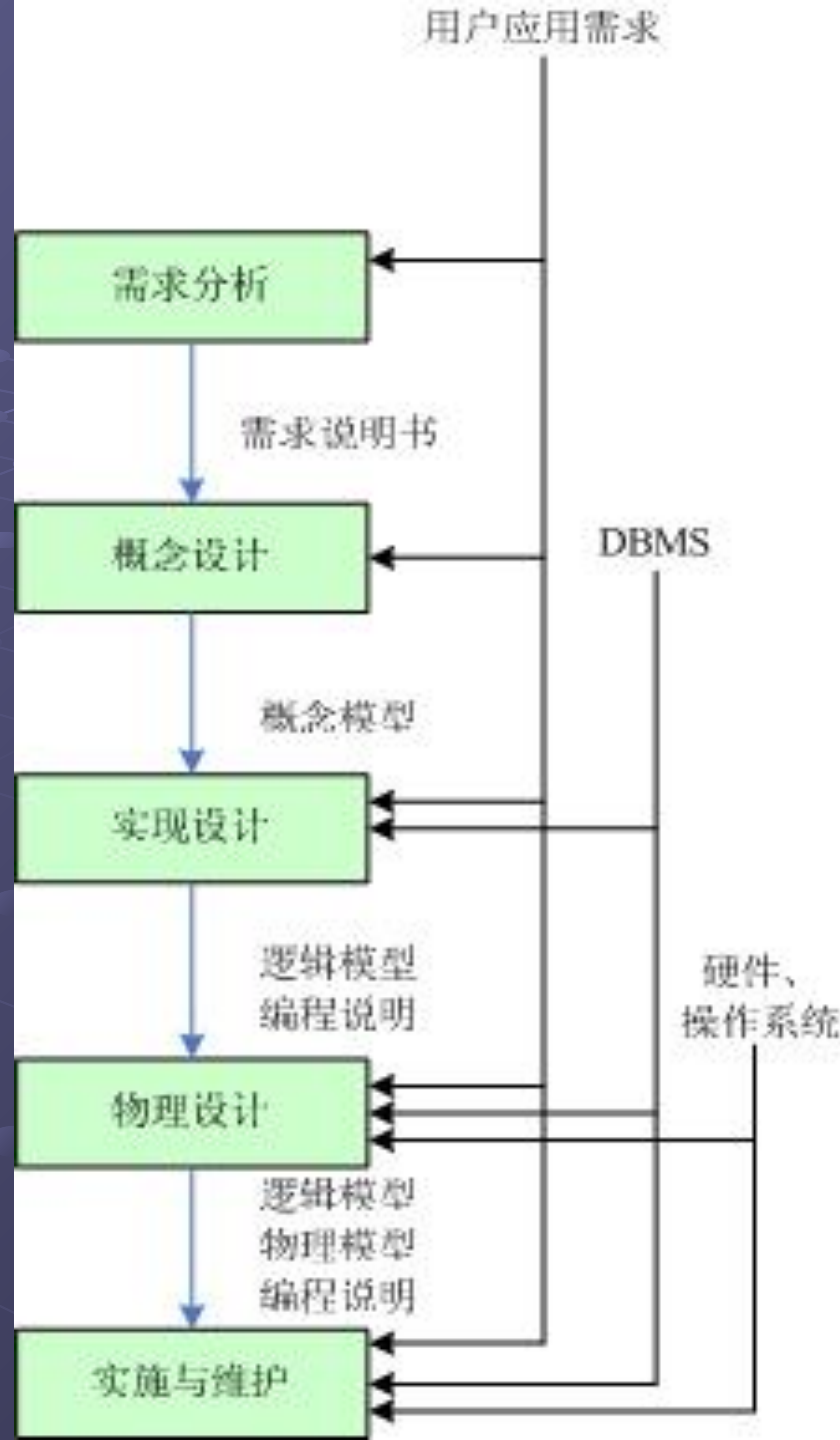
数据库设计方法概述

分阶段设计方法，该方法遵循自顶向下、逐步求精的原则，将数据库设计过程分解为若干个相互依存的阶段。每一个阶段采用不同的技术与工具，解决不同的问题。

基本步骤：

- (1) 需求分析
- (2) 概念设计
- (3) 实现设计
- (4) 物理设计
- (5) 实施与维护

数据库设计步骤



需求分析

需求分析的根本任务是了解用户需要哪些数据以及对数据如何处理的要求。

一般可分为四个步骤：

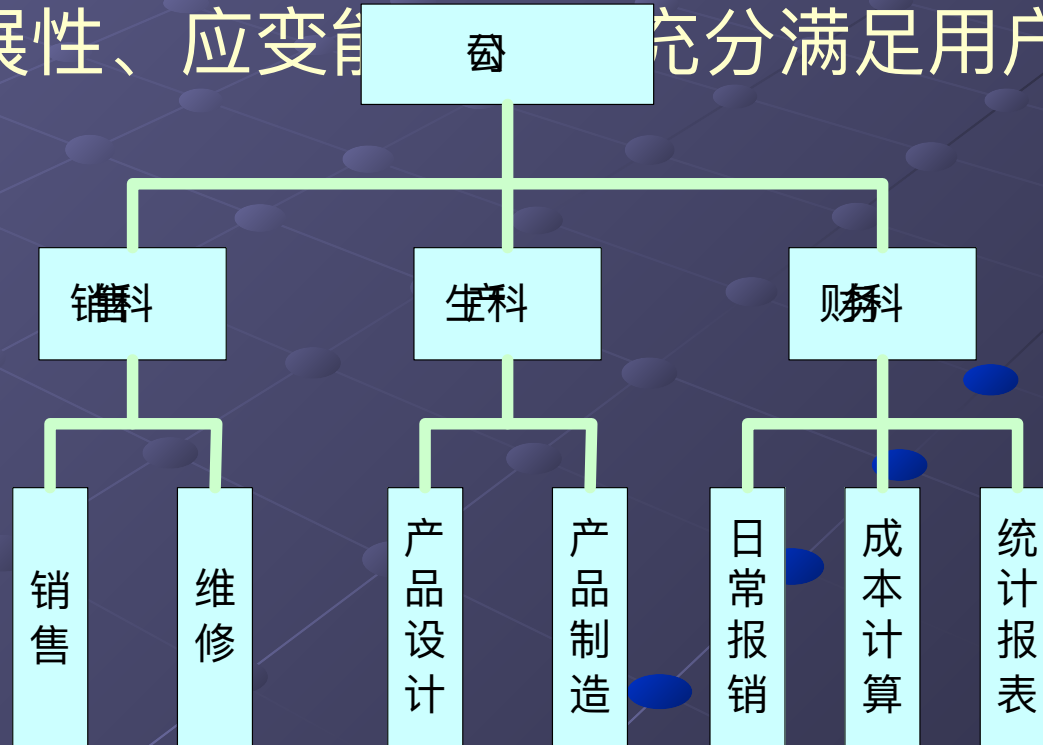
- (1) 确定数据库范围
- (2) 分析数据应用过程
- (3) 收集与分析数据
- (4) 整理文档

需求分析

一、确定数据库范围

数据库设计的第一项工作就是要确定数据库的范围，即确定数据库应支持哪些应用功能。

应该尽可能地考虑到较为广泛的应用部门或应用领域、可扩展性、应变能力，充分满足用户的应用功能要求。



需求分析

二、应用过程分析

应用过程分析是指了解并分析数据与数据处理间的关系。在数据库应用范围确定之后，设计人员可以借助于数据流图画出数据作用的顺序、处理策略及处理结果。

需求分析

三、数据的收集与分析

了解并分析数据的组成格式及操作特征，每个数据元素的语义及关系等。

1.静态结构

数据的静态结构是指数据的原始状况，这可通过数据分类表和数据元素表进行说明。

2.动态结构

动态结构是指将应用操作施加于数据之上后数据的状况，可通过任务分类表和数据操作特征表进行说明。

数据的动态结构对于分析数据的共享性、时间响应要

需求分析

三、数据的收集与分析

3.数据约束

数据约束是指使用数据时的特殊要求。约束主要有下述几个方面：

- 数据的安全保密性
- 数据的完整性
- 响应时间
- 数据恢复

需求分析

三、数据的收集与分析

4. 文档整理

即需求说明书。目标是对用户需求进行一次全面的汇总、调整与系统化，从而形成标准化的统一形式，包括：

- 数据库的应用功能目标
- 标明各不同用户视图范围
- 应用处理过程需求说明
- 数据字典

概念设计

概念设计的任务是在需求分析中产生需求说明书的基础上，设计出满足应用需求的用户信息结构，即独立于任何软件和硬件的概念模型。概念设计的常用方法为自顶向下方法，通过建立局部信息结构和合成全局信息结构的两个步骤完成。

概念设计

一、局部信息结构设计

局部信息结构就是根据需求说明书中标明的不同用户视图范围所建立的用户信息结构。

- 1.确定局部范围
- 2.选择实体
- 3.选择实体的关键字属性
- 4.确定联系实体
- 5.确定实体属性

概念设计

二、全局信息结构设计

全局信息结构设计是将局部信息结构设计中产生的局部信息合并成为一个全局信息结构。

合并的过程是一个不断发现和解决冲突的过程，冲突主要表现在：

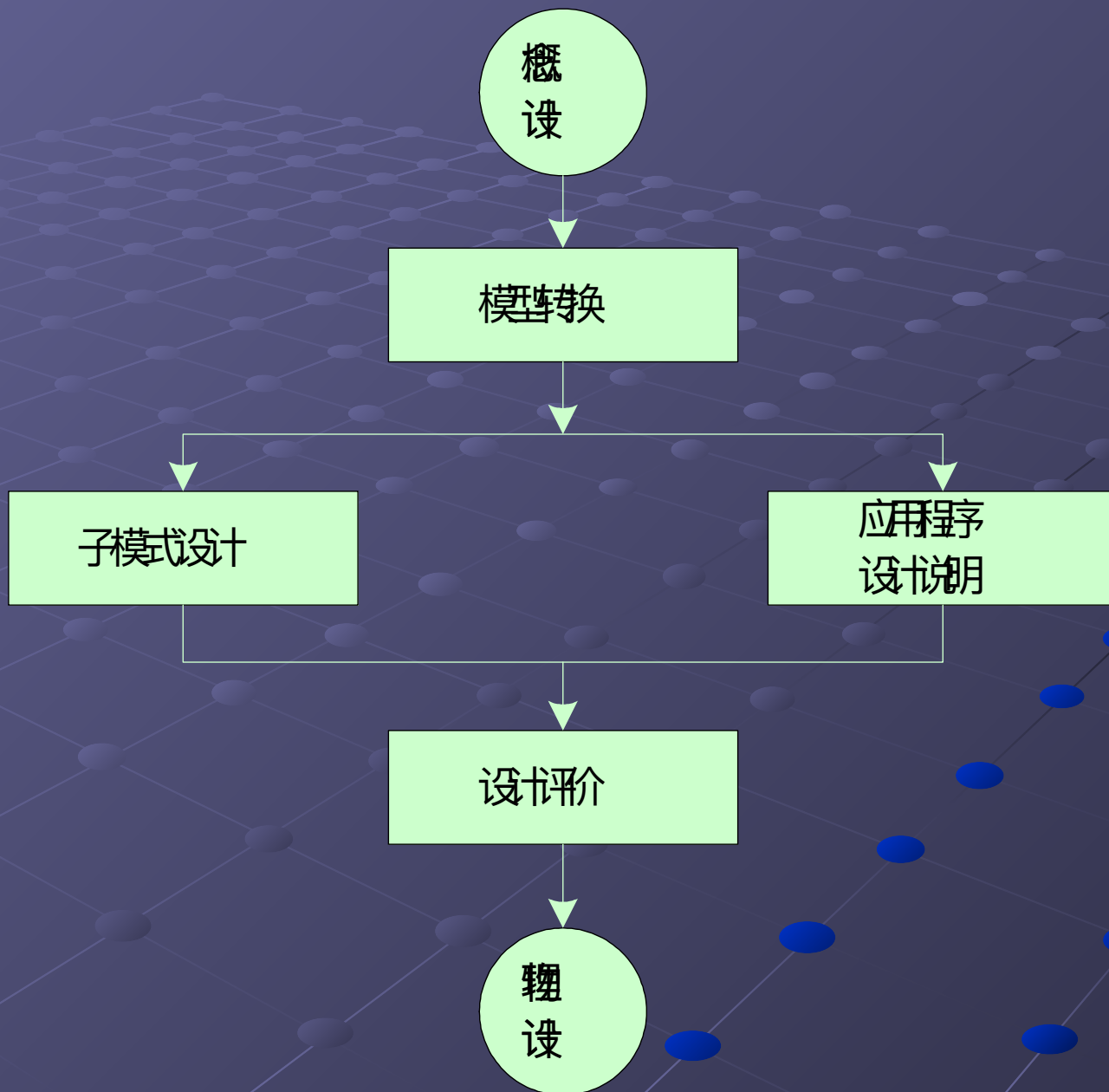
- 1.命名冲突
- 2.特征冲突
- 3.结构冲突

实现设计

实现设计的目标是将概念模型转换为等价的并为特定DBMS所支持数据模型的结构，要求提供如下的输入信息：

1. 独立于特定DBMS的概念模型；
2. 有关响应时间、安全保密性、数据完整性、一致性及恢复方面的要求说明；
3. 数据量及使用频率；
4. 特定DBMS支持的概念模式；
5. 一个或多个外部视图；
6. 物理设计说明，主要包括存入数据库中的数据量、使用频率及响应时间要求；

实现设计的步骤



物理设计

物理设计是指对于一个给定的数据库逻辑结构，研究并构造物理结构的过程。具体任务主要是确定数据库在存储设备上的存储结构及存取方法。因DBMS的不同还可能包括建立索引和聚集，以及物理块的大小、缓冲区个数和大小等。

一般来说，用户通常不需要进行数据存储结构和存取方法的设计，因为物理设计十分简单，只需在数据库系统中设置一下即可

实施与维护

实施是指数据库的定义、数据装载、应用程序的编制与调试，维护是指数据库试运行乃至正式运行过程中的评价、改进和扩展。

一、数据库建立

数据库定义

数据库装载

二、应用程序设计

通常采用通用的设计工具，目前比较常用的有VC++、Delphi、PowerBuilder、VB、.net、Java等

三、数据库试运行

关系规范化

关系规范化的目的在于控制数据冗余，避免插入和删除等操作异常，从而增强数据库结构的稳定性和灵活性。

数据冗余是指同一数据被反复存放的情况，数据冗余将导致存储空间的浪费和潜在数据不一致性及修改麻烦等问题。

数据的插入操作异常是指应该插入到数据库中的数据不能执行操作操作的情形。

数据的删除操作异常是指不应该删去的数据被删去的情形。

本节内容总结

- 数据库结构概述
- 关系模型、关系
- 实体—关系模型
- 数据模型的基本概念
- 数据库设计方法概述
- 需求分析、概念设计、实现设计、实施与amp;维护