

Internet of
Things

数据模型建立



目录 Contents



相关知识

| 数据模型建立



任务实践

| 创建云平台用的数据表



数据模型建立

数据模型

- **模型概念**

模型（model）：一般而言，模型是现实世界某些特征的模拟和抽象，分为实物模型与抽象模型。建筑模型，汽车模型，飞机模型等都是实物模型，它们通常是客观事物的某些外观特征或者功能的模拟与刻画；数学模型 $s=\pi r^2$ 是一种抽象模型，它抽象描述了圆的面积和圆的半径之间的数量关系，揭示客观事物的某些本质的、内部的特征。

- **数据模型**

在实现数据库管理的过程中，数据模型起着关键作用。整个数据库技术的发展就是沿着数据模型的主线展开的。现有的数据库均是基于某种数据模型，了解数据模型的基本概念是学习数据库的基础。根据模型应用的不同目的，可以将这些模型划分为两类，他们分属于两个不同的层次。

第一类模型是概念模型，也称信息模型。它是按用户的观点来对数据和信息建模，主要用于数据库设计。

另一类模型是数据模型，主要包括网状模型、层次模型、关系模型等，它是按计算机系统的观点对数据建模，主要用于DBMS的实现，这种数据模型也叫结构数据模型。

1.2 数据库管理的实现

数据模型的要素：

- **数据结构**

数据结构是所研究的对象类型的集合，这些对象是数据库的组成成分，它们包括两类，一类是与数据类型、内容、性质有关的对象；一类是与数据之间联系有关的对象。

- **数据操作**

数据操作是指对数据库中各种对象的（型）的实例（值）允许执行的操作的集合，包括操作及有关的操作规则。数据库主要有检索和更新（包括插入、删除、修改）两大类操作，数据模型必须定义这些操作的确切含义、操作符号、操作规则（如优先级）以及实现操作的语言。数据操作是对系统动态特性的描述。

- **数据的约束条件**

数据约束条件是一组完整性规则的集合。完整性规则是给定的数据模型中数据及其联系所具有的制约和依存规则，用以限定符合数据模型的数据库状态以及状态的变化，以保证数据的正确、有效、相容。如：入学年龄不超过30岁，成绩不得有三门以上不及格等。

1.2 数据库管理的实现

1.2.3 概念模型

概念模型的相关概念

- (1) **实体 (Entity)**：客观存在并相互区别的事物及其事物之间的联系。例如，一个学生、一门课程、学生的一次选课、一次考试等都是实体。
- (2) **属性 (Attribute)**：实体所具有的某一特性。例如，学生的学号、姓名、性别、出生年份、系、入学时间等。
- (3) **码 (Key)**：唯一标示实体的属性集。例如，学号是学生实体的码。
- (4) **域 (Domain)**：属性的取值范围。例如，年龄的域为15至35之间。
- (5) **实体型 (Entity Type)**：用实体名及其属性名集合来抽象和刻画同类实体，称为实体型。例如，学生(学号，姓名，性别，出生年份，系，入学时间)就是一个实体型。
- (6) **实体集 (Entity Set)**：同型实体的集合称为实体集。如，全体学生就是一个实体集。
- (7) **联系 (Relationship)**：实体与实体之间以及实体与组成它的各属性间的关系。

1.2 数据库管理的实现

实体间联系的三种情况

(1) 一对一联系 (1:1)

如果对于实体集A中的每一个实体，实体集B中至少有一个（也可以没有）实体与之联系，反之亦然，则称实体集A与实体集B具有一对一联系，记为1:1。例如，一个学生只能有一个学号，而一个学号只能指向一个学生，则学生与学号之间具有一对一联系。

(2) 一对多联系 (1:n)

如果对于实体集A中的每一个实体，实体集B中有n个实体 ($n \geq 0$) 与之联系，反之，对于实体集B中的每一个实体，实体集A中至多只有一个实体与之联系，则称实体集A与实体集B有一对多联系，记为1:n。例如，一个班级中有若干名学生，而每个学生只在一个班级中学习，则班级与学生之间具有一对多联系。

(3) 多对多联系 (m:n)

如果对于实体集A中的每一个实体，实体集B中有n个实体 ($n \geq 0$) 与之联系，反之，对于实体集B中的每一个实体，实体集A中也有m个实体 ($m \geq 0$) 与之联系，则称实体集A与实体集B具有多对多联系，记为m:n。例如，一门课程同时有若干个学生选修，而一个学生可以同时选修多门课程，则课程与学生之间具有多对多联系。

注意：实际上，一对一联系是一对多联系的特例，而一对多联系又是多对多联系的特例。

1.2 数据库管理的实现

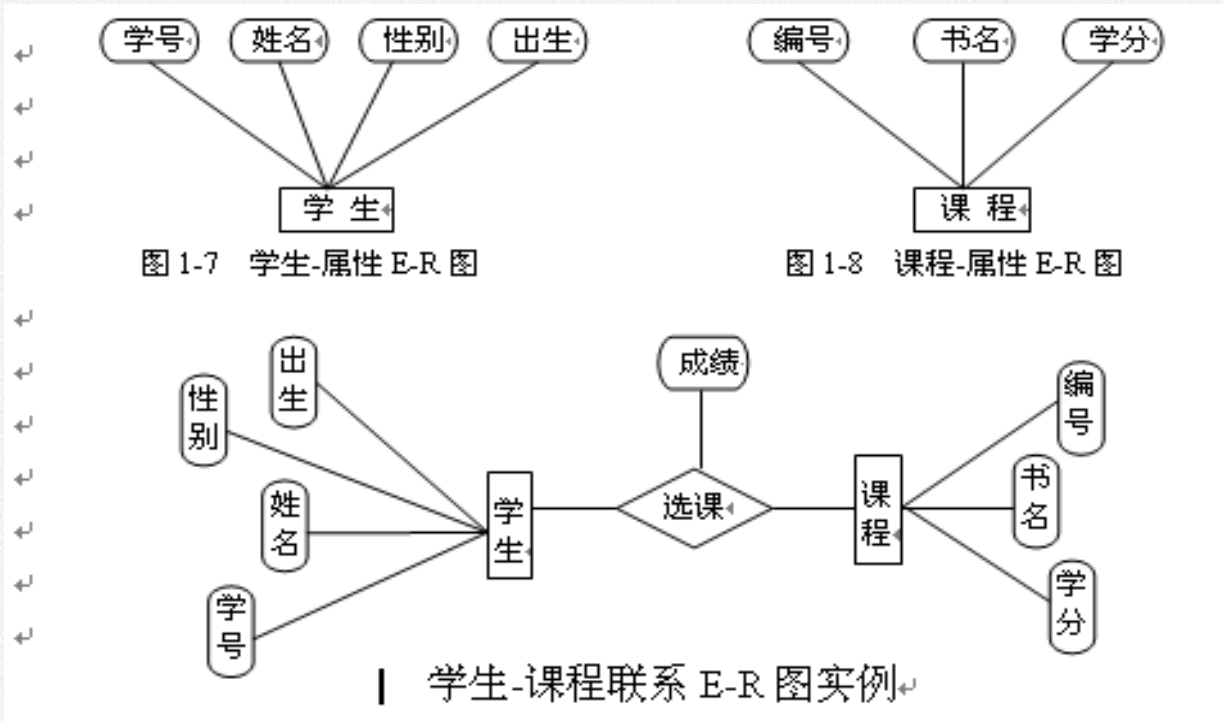
- 概念模型的表示方法

概念模型的表示方法很多，最常用的是实体—联系方法。该方法用E-R(Entity-Relationship Approach) 图来描述现实世界的概念模型。

E-R图提供了表示实体型、属性和联系的方法。E-R图有三个要素：

- (1) 实体型：用矩形表示，矩形框内写明实体名。
- (2) 属性：用椭圆形表示，并用无向边将其与相应的实体连接起来。
- (3) 联系：用菱形表示，菱形框内写明联系名，并用无向边分别与有关实体连接起来，同时在无向边旁标上联系的类型(1: 1, 1: n或m: n)。

1.2 数据库管理的实现



1.2 数据库管理的实现

1.2.4 数据模型

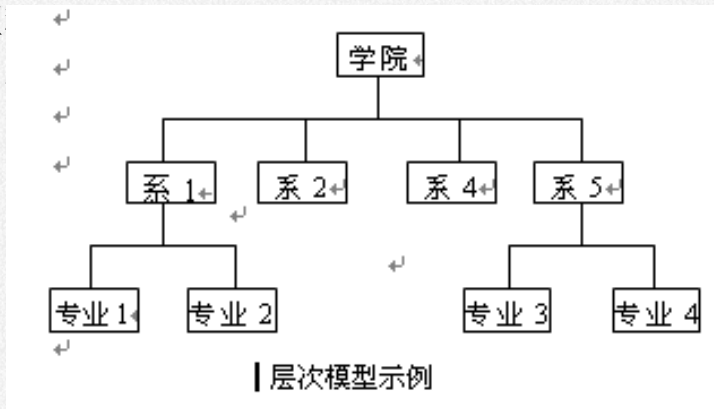
这里讲的数据模型是在概念级的模型（概念模型）的基础上形成的，为计算机上某一DBMS支持的数据模型。本书主要介绍前三种模型。

- **层次模型**

若用图来表示，层次模型是一棵倒立的树。在数据库中，满足以下条件的

数据模型
点：②

，这个结点称为根结



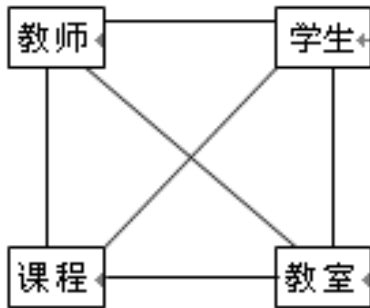
1.2 数据库管理的实现

- 网状模型

在现实世界中，事物之间的联系更多的是非层次关系的，用层次模型表示非树型结构是很不直接的，网状模型则可以克服这一弊病。网状模型是一个网络。在数据库中，满足以下两个条件的数据模型称为网状模型。① 允许一个以上的结点无父结点；② 一个结点可以有多个父结点。

从以上定义看出，网状模型构成了比层次结构复杂的网状结构，适宜表示多

上的“超链接数据”模式有类似



网状模型示例

1.2 数据库管理的实现

- **关系模型**

以二维表的形式表示实体和实体之间联系的数据模型称为关系数据模型。从模型的三

要素角度看，关系模型的内容为：（1）数据结构：一张二维表格。（2）数据操作：数据的定义，检索、维护等。（3）数据约束条件：表中列的取值范围即域值的限制条件。

关系模型的几个常见概念：

- （1）关系：一个关系就是一张二维表，每个关系都有一个关系名，即数据表名。
- （2）元组：表中的行称为元组，一行就是一个元组，对应表中一条记录。
- （3）属性：表中的列称为属性，即字段。字段名称为属性名，字段值称为属性值。
- （4）域：属性的取值范围，如，分数在0~100之间。
- （5）关键字：表中的一个属性（组），它的值可以唯一地标志一个元组。如：学号。
- （6）候选码：表中的某一个属性，它的值可以唯一地标志一个元组。一个表中可能有

多个候选码，选择一个作为主键，主键的属性称为主属性。

- （7）外关键字：如果一个关系中的属性或属性组并非该关系的关键字，但它们是另

1.2 数据库管理的实现



1.2 数据库管理的实现

- **关系模型的完整性**

关系模型对**数据**一般都具有一定的限制，这种限制称为完整性或完整性约束。

关系模型的完整性是保证关系数据表正确的关键。关系模型支持实体完整性约束、参照完整性约束和域约束3种完整性约束。

- (1) 实体完整性约束**

假设A是一个表R的主键，则A不能接收空值，即单列主键的值不能为空，复合主键的任何列也不能接收空值。例如，在学生信息表中，“学号”为该表的主键，那么在数据库的任何记录中，“学号”列的值都不能为空。这样的约束称为实体完整性约束。

1.2 数据库管理的实现

(2) 参照完整性约束

参照完整性约束关心的是逻辑相关的表中值与值之间的关系。假设X是一个表A的主键，在表B中是外键，那么若K是表B中一个外部键值，则表A中必然存在在X上的值为K的记录。例如，“系编码”是院系信息表的主键，而在学生信息表中是相对于

院系

信息表的外键（学生信息表中的主关键字是由“学号”和“系编码”组合而成），对于学生

信息表的任何记录，其所包含的“系编码”的值，在院系信息表的“系编码”列中必

一个相同

学号	系编码	姓名	性别
+ 20051201	0201	王意宝	女
+ 20051202	0201	李树财	男
+ 20051203	0101	李方浩	女
+ 20051204	0102	张敬秋	女
+ 20051205	0202	田文涛	男

系编码	系名
0101	信息工程系
0102	机械工程系
0201	外语系
0202	汽车工程系
0103	土木工程系

参照完整性约束

1.2 数据库管理的实现

(3) 域约束

域是逻辑相关的值的集合，从域中可以得出特定列的值。

例如，在学生信息表中：

- [出生日期]域的值必须按照特定的统一格式存放，而不能有时用：1986.12.23格式，有时用：12/23/1986格式，造成数据混乱；
- [学生名字]、[院系名称]等域的值必须属于字符集合；
- 对于[性别]，该域中的值必须局限于男、女等。

1.2 数据库管理的实现

1.2.5 关系运算

关系的基本运算有两类：传统的集合运算和专门的关系运算。

- **传统的集合运算**

(1) 并 (Union)：设有两个关系R和S，它们具有相同的结构。R和S的并是由属于R
或

属于S的元组组成的集合，运算符为 \cup 。例如：设每个班有一个学生关系R1, R2, R3, ...则全校学生关系T是： $T = R1 \cup R2 \cup R3 \cup \dots$

(2) 差 (Difference)：设有两个关系R和S，它们具有相同的结构。R和S的差是由属于R

但不属于S的元组组成的集合。运算符为 $-$ 。例如：设有参加计算机小组的学生关系R，参加桥牌小组的学生关系S。求参加了计算机小组但没有参加桥牌小组的学生。结果放在关系T1中，则 $T1 = R - S$ 。

1.2 数据库管理的实现

(3) 交 (Intersection)：设有两个关系R和S，它们具有相同的结构。R和S的交是由既属于R又属于S的元组组成的集合，运算符为 \cap 。交运算的结果是R和S的共同元组。

例如：设有参加计算机小组的学生关系 R，参加桥牌小组的学生关系 S。求既参加了计算机

小组又参加桥牌小组的学生。结果放在关系 T2中，则 $T2 = R \cap S$ 。关系的交可以用关系的差来表示。即 $R \cap S = R - (R - S)$ ，所以，上式也可以写成： $T2 = R - (R - S)$ 。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/195014221111011132>