

形式化关系查询语言

第 2 章到第 5 章，我们介绍了关系模型，并详细讲述了 SQL。在本章中我们将介绍 SQL 所基于的形式化模型，同时它也是其他关系查询语言的基础。

本章内容包括三种形式化语言。我们首先介绍关系代数 (Relational Algebra, RA)，它构成了广泛使用的 SQL 查询语言的基础。然后我们继续介绍“纯”查询语言，特别是元组关系演算和域关系演算，它们都基于数学逻辑的声明式查询语言。

6.1 关系代数

关系代数由一组运算组成，这些运算接受一个或两个关系作为输入，并生成一个新的关系作为它们的结果。

其中一些运算 (如选择、投影和更名运算) 称为一元 (unary) 运算，因为它们只在一个关系上进行运算。另一些运算 (如并、笛卡儿积和集差) 在一对关系上进行，因此称为二元 (binary) 运算。

尽管关系代数运算构成了广泛使用的 SQL 的基础，但是数据库系统并不允许用户用关系代数来编写查询。然而，也存在为便于学生练习关系代数查询而构建的关系代数实现。在本书英文版的网站 db-book.com 上，标题为“实验室材料”的链接下提供了几个这样的实现的网址。

在此值得回顾的是，由于关系是元组的集合，因此关系不能包含重复的元组。然而，在实践中，除非施加了特定的约束来禁止，否则数据库系统中的表允许包含重复元组。但是在讨论正式的关系代数时，如集合的数学定义所要求的那样，我们要求去除重复项。在第 3 章中，我们讨论了如何将关系代数扩展到多重集合上，多重集合是可以包含重复项的集合。

6.1.1 选择运算

选择 (select) 运算选出满足给定谓词的元组。我们使用小写的希腊字母 sigma (σ) 来代表选择。谓词写在 σ 的下标中。作为参数的关系写在 σ 后的括号内。因此，为了选出 *instructor* 关系的那些元组，使得它们对应于“物理”系中的教师，我们写为：

$$\sigma_{dept_name = \text{"Physics"}}(instructor)$$

如果 *instructor* 关系如图 2-1 所示，则上述查询得到的关系如图 6-1 所示。

通过编写如下语句，我们可以找出薪水高于 90 000 美元的所有教师：

$$\sigma_{salary > 90000}(instructor)$$

通常，我们允许在选择谓词中使用 =、 \neq 、<、 \leq 、> 和 \geq 来进行比较。此外，可以通过使用连接词 and (\wedge)、or (\vee) 和 not (\neg) 将几个谓词组合成一个更长的谓词。因此，为了查找薪水超过 90 000 美元的物理系的教师，我们写为：