# Web数据库、XML数据、空间数据语义近似查询技术

孟祥福　张霄雁　唐延欢　著

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

# Web 数据库、XML 数据、空间数据语义近似查询技术

孟祥福　张霄雁　唐延欢　著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

## 内 容 简 介

本书结合当前数据库查询领域的研究热点和最新研究成果，较为全面和系统地介绍了 Web 数据库、XML 数据、空间数据语义近似查询技术。本书内容分三部分，共 9 章，第一部分是 Web 数据库柔性查询、结果排序与分类、关键字查询推荐方法，主要内容包括：Web 数据库基础理论和相关技术、Web 数据库自适应查询松弛方法、Web 数据库多查询结果排序方法、Web 数据库多查询结果分类方法、Web 数据库关键字查询推荐方法。第二部分是 XML 数据近似查询方法，主要内容包括：XML 基础理论和查询技术、XML 数据近似查询方法。第三部分是空间关键字查询和兴趣点聚类分析方法，主要内容包括：空间关键字查询方法、空间兴趣点聚类分析方法。

本书可作为高等学校计算机专业、数据分析专业本科生和研究生提升研究能力的参考资料，也可供相关领域的研究人员学习和参考。



**图书在版编目（CIP）数据**

Web 数据库、XML 数据、空间数据语义近似查询技术/孟祥福，张霄雁，唐延欢著. —北京：电子工业出版社，2018.6
ISBN 978-7-121-34458-9

Ⅰ. ①W… Ⅱ. ①孟… ②张… ③唐… Ⅲ. ①数据检索－高等学校－教材 Ⅳ. ①G254.926

中国版本图书馆 CIP 数据核字（2018）第 125599 号

策划编辑：王羽佳
责任编辑：张 京
印 刷：北京京师印务有限公司
装 订：北京京师印务有限公司
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编：100036
开 本：787×1092 1/16 印张：10.5 字数：269 千字
版 次：2018 年 6 月第 1 版
印 次：2018 年 6 月第 1 次印刷
定 价：59.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：（010）88254535，wyj@phei.com.cn。

# 前　言

随着 World Wide Web 的迅速膨胀和 Internet 的普遍应用，Web 上出现了越来越多面向不同应用领域的在线数据库，如购物（如淘宝、京东）、房产（如链家网、雅虎房产）、二手车交易（如瓜子网、人人车、雅虎汽车）、股票（如新浪财经、VantageWire）、旅游（如携程、艺龙）、图书（如亚马逊、当当）等，这些领域的数据库的一个共同特点是允许大量外部用户通过 PC 或移动设备等对其进行访问。这种类型数据库的数量及其蕴含的信息量都呈快速增长的态势，它们的用户群也逐渐从相关熟悉领域知识的专业人员扩大到需要即时满足的普通用户。为了便于普通用户与数据库之间的交互，这种类型的数据库通常提供一种简单易用、基于 Web 表单形式的查询接口，用户通过数据库提供的查询接口指定查询要求（即查询条件）。这些 Web 中只能通过查询接口访问的在线数据库称为 **Web 数据库，其后台数据库通常是关系数据库**。现有的 Web 数据库查询处理模式通常假定用户明确自己的查询意图并且仅支持严格查询匹配，但随着 Web 数据库规模和复杂性的增加，要求大量普通用户了解 Web 数据库的结构和内容已不现实。这种情况下，即使用户使用明确的查询条件，Web 数据库仍有可能返回过少甚至**空查询结果**，此时大多数（特别是需要即时满足的）普通用户希望 Web 数据库系统能够自动放松初始查询条件，提供语义上近似匹配的查询结果。而查询松弛后，用户又可能会面临**多查询结果问题**，此时用户希望 Web 数据库系统能够对查询结果自动进行排序或分类，以避免信息过载。另外，由于 Web 数据库中存储了大量文本数据，因此像谷歌和百度那样使用关键字查询的 Web 数据库更为方便，现有的关键字查询技术主要利用全文索引的搜索方式从形式上匹配关键字，然而数据库中有些条目在内容上可能与关键字十分相关但形式上不匹配，目前的关键字查询技术无法检索到这些信息，这就是需要解决的**关键字语义查询问题**。

另外，随着 XML 成为 Web 上信息表示与交换的标准及其在众多领域的广泛应用，如何对 XML 文档进行高效的查询和处理变得愈加重要。XML 数据的两种重要查询模式是 XPath 和 XQuery，其本质是对 XML 数据单元之间的结构关系和内容进行精确匹配，返回的查询结果也必须严格满足所提出的查询条件。然而，在实际应用中，大量的普通用户对 XML 文档结构和内容并不了解，所提出的查询要求往往是对其查询意图的部分描述或近似描述，因此需要解决 **XML 数据的语义近似查询问题**。

近年来，随着移动互联网的普遍应用，Web 上有超过 1/5 的查询都与位置相关（如百度地图、美团等），这些查询同时包含了位置和文本查询要求，也就是空间数据查询。空间数据查询同样面临语义近似查询问题，如用户查询某个位置附近的“肯德基”，那么为其提供该位置附近的“麦当劳”，这在很大程度上也与用户的查询语义相关。此外，空间数据的聚类分析在诸多领域，如城市规划、市场营销、社区发现等都有着重要的应用价值。现有方法通常根据空间对象（兴趣点，point of interest）在位置上的相近性进行聚类，而实际上空间对象的相关性不仅体现在位置方面，更重要的是它们之间的社会联系，**因此需要综合考虑如**

何根据空间对象在位置上的相近性和社会关系上的紧密度进行聚类分析。

通过上述分析可见，大量需要即时满足的普通用户在查询 Web 数据库、XML 数据和空间数据获取信息时，希望查询系统能够提供灵活智能的方式为其提供语义相关的信息。当前，Web 数据库、XML 数据、空间数据查询处理模式无论是在查询形式方面还是在查询处理与分析方面，都难以满足当前用户的智能化、个性化查询分析需求，这方面的研究工作已经引起国内外研究者的广泛重视。

本书针对当前 Web 数据库查询中亟待解决的空查询结果、多查询结果、关键字语义查询推荐、XML 数据的查询松弛问题、空间数据查询与兴趣点聚类分析问题进行讨论，结合关系数据、XML 数据和空间数据等数据模型，按照查询松弛、松弛查询下的多查询结果排序与分类、关键字语义查询推荐、XML 数据近似查询、空间关键字查询与兴趣点聚类分析的顺序，阐述一套行之有效的 Web 数据库、XML 数据和空间数据语义近似查询的解决方案并提供具体实现技术。

本书内容分三大部分，共 9 章，第一部分是 Web 数据库柔性查询、结果排序与分类、关键字查询推荐方法，主要内容包括：Web 数据库基础理论和相关技术、Web 数据库自适应查询松弛方法、Web 数据库多查询结果排序方法、Web 数据库多查询结果分类方法、Web 数据库关键字查询推荐方法；第二部分是 XML 数据近似查询方法，主要内容包括：XML 基础理论和查询技术、XML 数据近似查询方法；第三部分是空间关键字查询和兴趣点聚类分析方法，主要内容包括：空间关键字查询方法、空间兴趣点聚类分析方法。

本书的撰写参考了大量近年的国内外相关技术资料，吸取了许多专家和同人的宝贵经验，在此深表谢意。

由于 Web 数据库、XML 数据、空间数据查询技术发展迅速，加之作者学识有限，书中难免有误漏之处，望广大读者批评指正。

作　者

2018 年 6 月

# 目　录

## 第一部分

## 第二部分



## 第三部分

# 第一部分

- 第1章 Web数据库基础理论和相关技术
- 第2章 Web数据库自适应查询松弛方法
- 第3章 Web数据库多查询结果排序方法
- 第4章 Web数据库多查询结果分类方法
- 第5章 Web数据库关键字查询推荐方法

# 第 1 章　Web 数据库基础理论和相关技术

内容关键词

- Web 数据库
- Deep Web
- 关系数据模型
- 关联规则、直方图、Top-$k$ 排序、决策树分类

## 1.1　Web 数据库

随着 World Wide Web 的迅速膨胀，网络上出现了越来越多面向不同应用领域的数据库，如购物（淘宝、京东）、房产（链家网、雅虎房产）、二手车（瓜子网、人人车、雅虎汽车）、股票（新浪财经、VantageWire）、旅游（携程、艺龙、Foursquare）、图书数据库（亚马逊、当当）等，这些数据库的一个共同特点是允许大量外部用户对其进行访问。随着 Internet 的普遍应用，这种类型数据库的数量及其蕴含的信息量都呈快速增长的态势，它们的用户群也逐渐从熟悉领域知识的专业人员扩大到需要即时满足的普通用户。为便于普通用户与数据库之间交互，这种类型数据库通常提供一种简单易用的、基于 Web 表单的查询接口（见图 1.1）。用户通过查询接口指定查询要求（即查询条件），然后查询条件被自动地转换成标准的查询语句提交给查询接口的后台数据库（underlying database）系统执行。这些 Web 中在线可用的、只能通过基于 Web 表单形式（Web form based）的查询接口访问的、存储海量信息的数据库称为 Web 数据库或 WDB[1, 2]。


图 1.1　Web 数据库查询接口和查询结果页面示例（Amazon.cn）

Web 数据库中的内容只有在被查询时才会由 Web 服务器动态生成页面，并把结果返回给用户（见图 1.2）。由此可见，查询接口是外部访问 Web 数据库的门户，是从 Web 数据库中获取数据的主要途径[2, 3]。

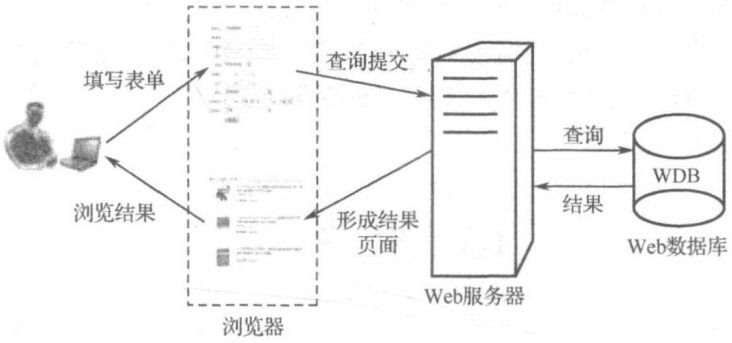


图 1.2　Web 数据库查询处理模式

查询接口支持 Web 数据库中若干属性上的查询，通过对其中若干属性的赋值形成一个对该查询接口所代表的 Web 数据库的查询。以图 1.1 所示的中文亚马逊图书 Web 数据库（Amazon.cn）提供的查询接口为例，用户可以通过该查询接口根据书名、作者、价格等方面的信息查询需要的图书。应当指出的是，查询接口虽然易于使用，但降低了查询的表达能力。例如，用户必须以精确的形式在查询接口表单中某个属性所对应的文本框（或下拉列表框）中填写（或选择）查询值，该值与其对应的属性及二者之间的关系构成一个基本查询条件（如商品名="Web 数据库"），多个基本查询条件之间由"and"连接，构成一个合取查询，即满足该查询的查询结果必须同时满足其中每个基本查询条件。这种查询方式很可能导致不理想的查询结果，因为用户必须以精确的、合取的形式表达他们的查询需求。在实际应用中，随着 Web 数据库规模和复杂性的增加，要求大量普通用户了解 Web 数据库的结构和内容已不现实，而现有的 Web 数据库查询处理模式又通常假定用户明确自己的查询意图并仅支持严格查询匹配，此时即使用户使用明确的查询条件，Web 数据库仍有可能返回过少甚至空查询结果。在这种情况下，大多数（特别是需要即时满足的）普通用户希望 Web 数据库系统能够自动放松初始查询条件来提供近似匹配的查询结果，此时查询条件应该是对查询结果的一个柔性限制，并不一定要求查询结果完全匹配。然而，查询松弛后，用户又可能面临多查询结果问题。在这种情况下，用户则希望 Web 数据库系统能够按照结果元组对用户需求和偏好的满足程度自动地对查询结果进行排序或分类以避免信息过载，从而使用户能够快速定位到其感兴趣的少量信息。另外，由于 Web 数据库中存储了大量文本数据，因此像谷歌和百度那样使用关键字查询 Web 数据库的查询技术已成为当前数据库查询领域研究的热点，现有的关键字查询技术主要利用全文索引的搜索方式从形式上匹配关键字，然而数据库中有些条目在内容上可能与关键字十分相关但形式上不匹配，目前的关键字查询技术无法检索到这些信息，因此需要解决关键字语义查询问题。

通过上述分析可见，随着 Internet 和移动网络的普遍应用及 Web 数据库的数量及其所蕴含信息量的迅速增长，Web 数据库的用户群也在发生着改变，大量需要即时满足的普通用户通过访问 Web 数据库获取信息，并期望 Web 数据库系统提供灵活、智能的方式让用户查询 Web 数据库。当前，Web 数据库查询处理模式无论是在查询形式还是在查询处理方面，都还难以满足当前普通用户对 Web 数据库系统提供智能化个性化查询服务的需求，因此使 Web 数据库系统支持更高级的智能化个性化查询已经变得十分必要，这方面的研究工作当前已经引起国内外研究者的广泛重视[1, 4, 5, 6]。

## 1.2 相 关 定 义

### 1.2.1 Deep Web

Web 数据库与 Deep Web 密切相关，Deep Web 是指 Web 中不能被传统的搜索引擎索引到的那部分内容，其内容主要源自对 Web 数据库的查询而得到的动态页面[7]。近年来，随着 Web 相关技术的日益成熟和 Deep Web 所蕴含信息量的快速增长，对 Deep Web 数据集成的研究越来越受到人们的关注。Deep Web 数据集成类似于谷歌、百度等搜索引擎，目的是为用户提供一个统一的访问途径来自动地获取并集成自由分布在整个 Web 上的多个 Web 数据库中丰富的信息[2]。可见，Web 数据库是 Deep Web 数据集成的信息来源，它们之间的关系如图 1.3 所示。

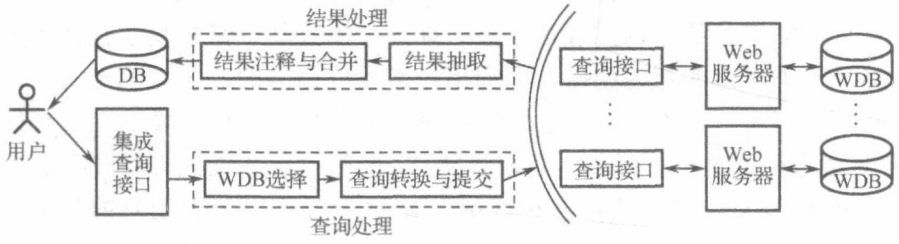


图 1.3　Web 数据库与 Deep Web 数据集成之间的关系

### 1.2.2 Web 数据库查询

查询接口是外部访问 Web 数据库的门户，是从 Web 数据库中获取数据的主要途径。目前，由 Web 查询接口产生的查询通常为合取查询（即基本查询条件之间由“and”连接）。由于大多数 Web 数据库的后台数据库采用关系数据模型，因此结合关系数据模型给出 Web 数据库查询的形式化定义：令 $R$ 为 Web 数据库中一个包含 $n$ 条元组$\{t_1,\cdots,t_n\}$的关系，其模式由 $m$ 个分类型（Categorical）和数值型（Numerical）相混合的属性 $A=\{A_1,A_2,\cdots,A_m\}$构成，$Q$ 为 $R$ 上一个合取查询，形式为 $Q=\sigma_{\wedge_{i\in\{1,\cdots,k\}}}(A_i\ \theta\ a_i)$，其中 $k\leqslant m$ 且 $\theta\in\{>,<,=,\geqslant,\leqslant,\text{between}\}$，如果 $\theta$ 是操作符“between”，则 $a_i$ 用区间$[a_{i1}, a_{i2}]$表示，$A_i\ \theta\ a_i$ 的形式为“$A_i$ between $a_{i1}$ and $a_{i2}$”。

基本查询条件“$A_i\theta a_i$”中的属性 $A_i$ 是属性集 $A$ 中的属性，$a_i$ 包含于属性 $A_i$ 的值域当中。$X=\{X_1,X_2,\cdots,X_k\}\subseteq A$ 是被查询指定的属性集合，简称指定的属性集；$Y=A-X$ 是未被查询指定的属性集合，简称未指定的属性集。相应地，被查询指定的属性值简称为指定的属性值，未被查询指定的属性值简称为未指定的属性值。

为了方便叙述，这里规定了与 Web 数据库查询相关的术语说明（见表 1.1），本书第 2、3、4、5 章的内容都是以关系数据模型为基础阐述相关技术方法的，这些方法可直接或经过适当修改应用到后续的 XML 和空间数据查询方法中。

表 1.1　Web 数据库查询相关的术语说明

| 定　义 | 说　明 |
| --- | --- |
| 基本查询条件<br>（basic query condition） | 一个查询指定在关系中某个属性上的原子条件 $A_i\theta a_i$ 称为基本查询条件，$A_i$ 代表属性，$\theta\in\{>,<,=,\geqslant,\leqslant,\text{between}\}$，$a_i$ 代表查询值 |
| 查询条件<br>（query condition） | 通过查询接口指定在 Web 数据库中多个（或至少一个）属性上的基本查询条件的合取构成一个查询条件，也称为查询（query）或查询要求，表示为 $Q=\sigma\wedge_{i\in\{1,\cdots,k\}}(A_i\ \theta\ a_i)$，其中，$1\leqslant k\leqslant m$，$A_i\theta a_i$ 为一个基本查询条件。按照基本查询条件指定的查询范围不同，可将其分为等值查询条件和范围查询条件 |
| 等值查询条件<br>（equality query condition） | 形如“$A_i=a_i$”的基本查询条件称为等值查询条件，其中 $a_i$ 是用户指定在属性 $A_i$ 上的一个查询值；如果一个查询仅由等值查询条件构成，则称该查询为等值查询，如 $Q=\sigma\wedge_{i\in\{1,\cdots,k\}}(A_i=a_i)$ |
| 范围查询条件<br>（range query condition） | 形如“$A_i$ in $(a_{i1},\cdots,a_{ik})$”或“$A_i$ between $a_{i1}$ and $a_{i2}$”的基本查询条件称为范围查询条件，这类查询条件在某个属性上指定多个值或一个区间，由至少一个范围查询条件构成的查询称为范围查询 |

### 1.2.3　查询历史

Web 数据库查询历史（Query history）是指所有使用过该 Web 数据库系统的用户提出的查询记录集合，用 $H$ 表示。查询历史可由后台数据库管理系统（DBMS）提供的查询日志功能收集。查询日志记录了用户与系统交互的整个过程，不同系统的日志记录格式略有不同，但一般都包括 Session ID、用户的访问时间、输入的查询条件、用户退出系统的时间等。需要说明的是，一个 Session 以用户连接到数据库开始，并以用户断开连接结束。在一个 Session 中，用户可能提出多条查询，假设同一 Session 中的查询是由同一个用户提出的。

由于需要从查询历史中获取用户偏好，而查询日志以字符串形式记载的查询历史不便于用户偏好的挖掘，因此对其进行规范化处理。对于查询历史中的一条查询记录，假设该查询为一个等值查询，其对应的关系模式为 $R(A_1,A_2,\cdots,A_m)$，且 $R$ 中包含 $n$ 条元组$\{t_1,t_2,\cdots,t_n\}$，则可将其分解为一个 $m$ 维向量，其中，每个分量上的取值就是该查询指定在对应属性上的查询值。例如，如果该查询在属性 $A_i$ 上指定了查询值 $a_i$，则对应属性 $A_i$ 上的分量取值即为 $a_i$；如果该查询在属性 $A_i$ 上没指定任何值，则对应属性 $A_i$ 上的分量取值为空。注意，对于一个范围查询，如果该查询指定在属性 $A_i$ 上的基本查询条件为范围查询条件，则将其转换成一个区间形式。例如，用$(a_{i1}, a_{i2}, a_{i3})$表示基本查询条件“$A_i$ in $(a_{i1}, a_{i2}, a_{i3})$”，用$[0, a_i)$表示基本查询条件“$A_i < a_i$”。根据上述方法，查询历史中的所有查询记录分解后可构成一个二维表形式

（见表 1.2）。

表 1.2　查询历史举例

| UID | QID | $A_1$ | $A_2$ | … | $A_m$ |
|---|---|---|---|---|---|
| $U_1$ | $Q_1$ | | $[0, a_{25}]$ | … | |
| $U_1$ | $Q_2$ | $(a_{11}, a_{13})$ | $[a_{22}, a_{25}]$ | … | $(a_{m1},a_{m2},a_{m3})$ |
| $U_1$ | $Q_3$ | $a_{11}$ | $[a_{23}, a_{25}]$ | … | $a_{m1}$ |
| … | … | … | … | … | … |
| $U_2$ | $Q_1$ | | $[a_{2k}, a_{2n}]$ | … | $a_{mn}$ |
| $U_2$ | $Q_2$ | $a_{1k}$ | $(a_{2k+1}, a_{2n}]$ | … | $a_{mn}$ |
| … | … | … | … | … | … |

在表 1.2 中，UID 代表 Session ID，QID 代表查询 ID，UID 与 QID 的组合能够唯一标识一条查询记录，表中的每条元组都代表一个查询。如果元组在某个属性上取值为空，则表示与其对应的查询在该属性上没指定任何基本查询条件；如果元组在某个属性上取值非空（这里假设取值为 $a_i$ 或$[a_{i1}, a_{i2}]$），则表示与其对应的查询在该属性上指定的基本查询条件为 $A_i = a_i$ 或 $A_i$ between $a_{i1}$ and $a_{i2}$。

Web 数据库查询历史中的查询记录，有些具有代表性意义，有些是无意义的，而无意义的查询记录将影响整个查询历史的数据质量。本书后续介绍的查询方法在很大程度上依赖于查询历史（如查询历史将作为挖掘隐式用户偏好的数据源），因此查询历史的数据质量将直接影响所提方法的有效性。为了提高查询历史的数据质量，需要对查询历史进行修剪，以便保存那些具有代表性意义的查询记录。对查询历史的修剪基于如下两个原则：

- 导致空查询结果的查询是无意义的，从查询历史中移除这样的查询记录；
- 属于同一 Session 的查询通常是由同一个用户提出的，而且该用户通常会以一个宽泛的查询开始，然后通过观察查询结果再逐步完善以前的查询，直到返回满意的查询结果为止，即同一个用户的查询通常是渐进式的。所以，在同一 Session 中的查询，通常只有最后一个查询是重要且有意义的，因此仅保留查询历史在同一 Session 中的最后一条查询。

修剪后的查询历史形式为 $H=\{(U_1,Q_1),\cdots,(U_i,Q_i),\cdots,(U_k,Q_k)\}$，其中 $U_i$ 代表 Session ID，$Q_i$ 代表一条历史查询记录。

## 1.3　关系数据模型

关系数据模型是当前 Web 数据库使用的最重要的一种数据模型，关系数据库系统采用关系模型作为数据的组织方式。下面简单介绍关系数据模型和关系数据库基本理论[8]。

### 1.3.1 关系

（1）域（Domain）

**定义 1.1**　域是一组具有相同数据类型的值的集合。例如，实数、长度在 0～255 字节的字符串集合、{0, 0.2, 1}等，都可以是域。

（2）笛卡儿积（Cartesian product）

**定义 1.2**　给定一组域 $D_1$, $D_2$, …, $D_n$，这些域中可以有相同的，$D_1$, $D_2$, …, $D_n$ 的笛卡儿积为

$$D_1 \times D_2 \times \cdots \times D_n = \{(d_1, d_2, \cdots, d_n) \mid d_i \in D_i, i = 1, 2, \cdots, n\}$$

其中的每一个元素$(d_1, d_2, \cdots, d_n)$叫作一个 $n$ 元组（$n$-tuple）或简称元组（Tuple）。

元组中的每一个值 $d_i$ 叫作一个分量（Component）。

若 $D_i(i = 1, 2, \cdots, n)$为有限集，其基数（Cardinal number）为 $m_i(i = 1, 2, \cdots, n)$，则 $D_1 \times D_2 \times \cdots \times D_n$ 的基数 $M$ 为：$M = \prod_{i=1}^{m} m_i$。

笛卡儿积可表示为一个二维表。表中的每行对应一个元组，表中的每列对应一个域。

（3）关系（Relation）

**定义 1.3**　$D_1 \times D_2 \times \cdots \times D_n$ 的子集叫作在域 $D_1, D_2, \cdots, D_n$ 上的关系，表示为

$$R(D_1, D_2, \cdots, D_n)$$

这里 $R$ 表示关系的名字，$n$ 是关系的目或度（Degree），关系中的每个元素是关系中的元组，通常用 $t$ 表示。

关系是笛卡儿积的有限子集，可用一个二维表表示，表的每行对应一个元组，表的每列对应一个域，每列的名字称为属性（Attribute）。若关系中的某一属性组的值能唯一地表示一个元组，则称该属性组为候选码（Candidate key）。可以在候选码中选定一个作为主码（Primary key），主码的诸属性称为主属性（Primary attribute），不包含在任何候选码中的属性称为非码属性（Non-key attribute）。在最简单的情况下，候选码只包含一个属性。在最极端的情况下，关系模式的所有属性组都是这个关系模式的候选键，称为全码（All-key）。

关系数据模型中主要包括两种类型的属性，即分类型属性（Categorical attribute）和数值型属性（Numerical attribute），下面分别给出它们的定义。

**定义 1.4**　设 $A_1$, $A_2$, …, $A_m$ 是关系模式 $R$ 中的 $m$ 个属性，Dom($A_1$)，Dom($A_2$)，…，Dom($A_m$)分别是对应于这些属性的值域。若值域 Dom($A_j$)是有限（Infinite）且无序的（Unordered），则称属性 $A_j$ 是分类型属性；若值域 Dom($A_j$)是由数值构成的有限集合且值之间具有一个隐含的序，则称属性 $A_j$ 是数值型属性[9]。

例如 Amazon 中文图书网站，其后台数据库中包含一个图书关系表 BookDB（商品名，定价，库存状态……），其中属性“商品名”是分类型属性，它的值域包含了所有图书的名称（如“数据结构”、“信息检索”等）；属性“定价”为数值型属性，它的值域包含了所有图书的价格（如“21”，“25”，“50”）。

### 1.3.2 关系模式

在关系数据库中，关系模式是对关系的描述。关系是指一张二维表，表的每一行为一个元组，每一列为一个属性。一个元组就是该关系所涉及的属性集的笛卡儿积的一个元素。关系是元组的集合，关系模式指出了这个元组集合的结构，即它由哪些属性构成，这些属性来自哪些域，以及属性与域之间的映像关系。

**定义 1.5** 关系的描述称为关系模式（Relation schema）。它可以形式化地表示为：$R(U, D, \text{Dom}, F)$，其中 $R$ 为关系名，$U$ 为组成该关系的属性集合，$D$ 为属性组 $U$ 中属性所来自的域，Dom 为属性向域的映像集合，$F$ 为属性间数据的依赖关系集合。

关系是关系模式在某一时刻的状态或内容。关系模式是静态的、稳定的，而关系是动态的、不断变化的，因为关系操作在不断地更新着数据库中的数据。

### 1.3.3 关系数据库

在关系数据模型中，实体及实体间的联系都是用关系表示的。例如，导师实体、博士生实体、导师与博士生之间的一对多联系都可分别用一个关系表示。在一个给定的应用领域中，所有实体及实体之间联系的关系的集合构成一个关系数据库。关系数据库的型称为关系数据库模式，是对关系数据库的描述，它包括若干域的定义以及在这些域上定义的若干关系模式。关系数据库的值是这些关系模式在某一时刻对应的关系的集合，通常称为关系数据库，在关系数据库中（或以关系模式）存储的数据也称为结构化数据。

## 1.4 Web 数据库查询相关技术

Web 数据库查询涉及的主要技术包括：关联规则挖掘（在查询历史中挖掘带偏好程度的上下文偏好）、直方图（统计原始数据和查询结果中的属性值分布情况）、Top-$k$ 排序算法（对查询结果进行快速排序处理）及决策树分类（对查询结果进行分类处理）。下面简述这些技术的基本思想以便为下文使用提供基础。

### 1.4.1 关联规则挖掘

关联规则（Association rules）反映一个事物与其他事物之间的相互依存性和关联性，如果两个或者多个事物之间存在一定的关联关系，那么其中一个事物就能够通过其他事物预测到。典型的关联规则例子就是“90%的顾客在购买面包和黄油的同时也会购买牛奶”，其直观的意义是，顾客在购买某些东西的时候有多大的倾向也会购买另外一些东西。

（1）关联规则基本模型

设 $I = \{i_1, i_2, \cdots, i_m\}$ 为 $m$ 个不同项目的集合，$D$ 为事务数据库，$D$ 中的一个事务 $T$ 是一个项目子集（$T \subseteq I$）。每一个事务具有唯一的事务标识 $TID$。设 $A$ 是一个由项目构成的集合，称为项集。事务 $T$ 包含项集 $A$，当且仅当 $A \subseteq T$。如果项集 $A$ 中包含 $k$ 个项目，则称其为 $k$ 项集。项集 $A$ 在事务数据库 $D$ 中出现的次数占 $D$ 中总事务数的百分比叫作项集的支持度。如

果项集的支持度超过用户给定的最小支持度阈值，就称该项集是频繁项集（Frequent itemsets）或大项集（Large itemsets）[10]。

关联规则是形如 $X \Rightarrow Y$ 的逻辑蕴含式，其中 $X \subset I$，$Y \subset I$，且 $X \cap Y = \varnothing$。如果以上面的例子为例，那么 $X$ 中就包含了“面包”、“黄油”两个项目，$Y$ 中包含了“牛奶”一个项目。下面给出关联规则的支持度和信任度的定义。

**定义 1.6**　支持度（Support）：事务数据库 $D$ 中包含 $X \cup Y$ 的事务数占 $D$ 中事务总数的百分比（记作 Sup），称为规则 $X \Rightarrow Y$ 在 $D$ 中的支持度，即

$$\text{Sup} = \frac{\text{support}(X \cup Y)}{n} \times 100\% \tag{1.1}$$

式中，support($X \cup Y$)为 $D$ 中包含（或支持）$X \cup Y$ 的事务数，$n$ 为 $D$ 中的事务总数。

**定义 1.7**　信任度（Confidence）：事务数据库 $D$ 中包含 $X$ 的同时也包含 $Y$ 的事务数占 $D$ 中包含 $X$ 的事务数的百分比（记作 Conf），称为规则 $X \Rightarrow Y$ 在 $D$ 中的信任度，即

$$\text{Conf} = \frac{\text{Sup}(X \cup Y)}{\text{Sup}(X)} \times 100\% \tag{1.2}$$

式中，Sup ($X \cup Y$)为 $X \cup Y$ 的支持度，Sup($X$)为 $X$ 的支持度。

实际上，支持度是一个概率值，信任度是一个条件概率值，即

$$\text{Sup}\ (X \Rightarrow Y) = P\ (X \cup Y)$$

$$\text{Conf}\ (X \Rightarrow Y) = P\ (Y \mid X)$$

例如，在表 1.3 所示的商品交易数据库中找出所有最小支持度为 50%、最小信任度为 50%的关联规则，根据上述定义可得到如下两条关联规则。

**Rule 1:** $A \Rightarrow C$（支持度：50%，信任度：66.6%）。

**Rule 2:** $C \Rightarrow A$（支持度：50%，信任度：100%）。

**表 1.3　商品交易数据库表**

| Transactions ID | Merchandises |
|---|---|
| 2000 | *A, B, C* |
| 1000 | *A, C* |
| 4000 | *A, D* |
| 5000 | *B, E, F* |

关联规则的挖掘问题就是生成所有满足用户指定的最小支持度（minsup）和最小信任度（minconf）的关联规则，即这些关联规则的支持度和信任度分别不小于最小支持度和最小信任度。

**定义 1.8**　满足最小支持度和最小信任度的关联规则称为强关联规则（Strong association rules）。

关联规则具有如下两条基本性质。

**性质 1.1**　频繁项集的子集必为频繁项集。

**性质 1.2**　非频繁项集的超集一定是非频繁的。