世界著名计算机教材精选

数据库系统基础教程

[美] Jeffrey D. Ullman, Jennifer Widom 著

史嘉权 等译

清华大学出版社
http://www.tup.tsinghua.edu.cn

PRENTICE HALL
http://www.prenhall.com

需要全本请访问 www.ertongbook.com

(京)新登字 158 号

内 容 简 介

本书是由美国斯坦福大学两位著名的计算机学者 Jeffrey D. Ullman 和 Jennifer Widom 为初学数据库的人编写的基本教材。内容以对数据库的使用为主,讲述了数据建模,关系数据模型,SQL 语言以及面向对象数据库的查询语言 OQL 的基本概念。作者根据当前数据库领域的发展,对全书内容做了较大调整,删除了大量旧内容,增加了面向对象的新技术。本书内容简洁,概念清楚,适合作大学本科生学习数据库的参考书。

A First Course in Database Systems
Jeffrey D. Ullman, Jennifer Widom

Copyright © 1997 by Prentice Hall, Inc.
Original English Language Edition Published by Prentice Hall, Inc.
All Rights Reserved.

本书中文简体字版由 Prentice Hall 出版公司授权清华大学出版社独家出版、发行。未经出版者书面许可,不得以任何方式复制或抄袭本书的任何部分。
本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。
北京市版权局著作权合同登记号: 图字 01-98-006 号

图书在版编目(CIP)数据

数据库系统基础教程/(美)额尔曼,(美)威多姆著;史嘉权译. —北京: 清华大学出版社, 1999. 8
(世界著名计算机教材精选)
ISBN 7-302-03646-2

Ⅰ. 数… Ⅱ. ①额… ②威… ③史… Ⅲ. 数据库系统-教材 Ⅳ. TP311. 13

中国版本图书馆 CIP 数据核字(1999)第 30397 号

出版者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)
http://www.tup.tsinghua.edu.cn
印刷者: 清华大学印刷厂
发行者: 新华书店总店北京发行所
开 本: 787× 1092 1/16 印张: 21. 75 字数: 501 千字
版 次: 1999 年 9 月第 1 版 2001 年 11 月第 4 次印刷
书 号: ISBN 7-302-03646-2/TP・2029
印 数: 16001～19000
定 价: 36. 00 元

译 者 前 言

数据库技术近年来发展非常迅速,特别是提出信息高速公路以来,所谓“3C”即计算机、通信和信息内容(Computer、Communication 和 Contents)已成为信息技术的核心。而信息高速公路的价值正体现在信息内容上,若没有大量的数据库存放这些“内容”并提供迅速、简便、高效的查询手段,则信息高速公路就只能“跑空车”了。

面向对象的数据库技术是近年来数据库技术发展的重要方向和热点,目前国内在该领域的科研方面已在积极开展,但在教材中尚无反映,因此急需有关的教材,而本书正是雪中送炭。

本书是从斯坦福大学 1997 年的教材《数据库系统基础教程》(《A First Course in Database Systems》)翻译过来的。本书以当前的主流数据库——关系数据库——为基础,以数据库系统的最基本内容——数据库的设计与编程——为重点,以引进数据库领域的最新成果——比如面向对象的数据库技术——为特点,系统地阐述了数据库建模、关系数据库的理论和设计、结构化查询语言 SQL 及其最新的标准 SQL2 和 SQL3,阐述了递归查询等最新内容,特别是以相当多的篇幅阐述了面向对象数据库的对象定义语言 ODL 和对象查询语言 OQL。

本书的主要特点是新颖、丰富、系统、实用,把数据库技术的最新成果迅速反映到教材中。斯坦福大学是世界一流大学,世界著名的硅谷与斯坦福有不解之缘。我们及时引进国外的最新教材,对提高国内的计算机教学和科研水平会起到积极的推动作用。

为培养同学直接从英文资料获取信息的能力,清华大学出版社已于 98 年出版了本书英文原著的影印本,现在又出版该书的中译本,这样可使不同程度的读者都能从中有所收益。读者不仅可以从中学到最新的专业知识,也能从中提高英文的专业阅读能力。

本书的翻译得到了我系周立柱教授的大力支持,在此表示衷心的感谢!在本书的翻译过程中,王霞、张勇、张劲飞和武志光同学为初稿的翻译和文稿的录入协助做了很多工作。本书的译文难免有不妥之处,敬请读者予以指正。

目　　录

第 1 章　数据库系统的世界 …… 1
1.1　数据库系统的发展 …… 1
1.1.1　早期的数据库管理系统 …… 1
1.1.2　关系数据库系统 …… 3
1.1.3　越来越小的系统 …… 4
1.1.4　越来越大的系统 …… 4
1.2　数据库管理系统的结构 …… 5
1.2.1　DBMS 的组成概述 …… 5
1.2.2　存储管理程序 …… 7
1.2.3　查询处理程序 …… 7
1.2.4　事务管理程序 …… 8
1.2.5　客户程序-服务程序体系结构 …… 10
1.3　未来的数据库系统 …… 10
1.3.1　类型、类和对象 …… 10
1.3.2　约束和触发程序 …… 13
1.3.3　多媒体数据 …… 13
1.3.4　数据集成 …… 14
1.4　本书概要 …… 15
1.4.1　设计 …… 15
1.4.2　编程 …… 15
1.5　本章总结 …… 16
1.6　本章参考文献 …… 17

第 2 章　数据库建模 …… 18
2.1　ODL 介绍 …… 18
2.1.1　面向对象的设计 …… 19
2.1.2　接口说明 …… 20
2.1.3　ODL 中的属性 …… 20
2.1.4　ODL 中的联系 …… 22
2.1.5　反向联系 …… 22
2.1.6　联系的多重性 …… 24
2.1.7　ODL 中的类型 …… 26
2.1.8　本节练习 …… 27
2.2　实体联系图 …… 29

2.2.1 E/R 联系的多重性 …… 30
2.2.2 联系的多向性 …… 30
2.2.3 联系中的角色 …… 31
2.2.4 联系中的属性 …… 32
2.2.5 把多向联系转换成二元联系 …… 33
2.2.6 本节练习 …… 34
2.3 设计原则 …… 35
2.3.1 真实性 …… 36
2.3.2 避免冗余 …… 36
2.3.3 对简单性的考虑 …… 36
2.3.4 选择合适的元素类型 …… 37
2.3.5 本节练习 …… 38
2.4 子类 …… 40
2.4.1 ODL 中的子类 …… 40
2.4.2 在 ODL 中的多重继承 …… 40
2.4.3 实体联系图中的子类 …… 42
2.4.4 E/R 模型中的继承 …… 42
2.4.5 本节练习 …… 43
2.5 对约束的建模 …… 44
2.5.1 键码 …… 45
2.5.2 在 ODL 中说明键码 …… 46
2.5.3 在 E/R 模型中表示键码 …… 47
2.5.4 单值约束 …… 47
2.5.5 参照完整性 …… 48
2.5.6 E/R 图中的参照完整性 …… 48
2.5.7 其他类型的约束 …… 49
2.5.8 本节练习 …… 49
2.6 弱实体集 …… 50
2.6.1 产生弱实体集的原因 …… 50
2.6.2 对弱实体集的要求 …… 52
2.6.3 弱实体集的表示法 …… 52
2.6.4 本节练习 …… 53
2.7 历史上有影响的模型 …… 53
2.7.1 网状模型 …… 53
2.7.2 网状模式的表示 …… 54
2.7.3 层次模型 …… 55
2.7.4 本节练习 …… 56
2.8 本章总结 …… 56
2.9 本章参考文献 …… 57

第 3 章 关系数据模型 …… 58
3.1 关系模型的基本概念 …… 58

需要全本请访问 www.ertongbook.com

3.1.1 属性 …… 59
3.1.2 模式 …… 59
3.1.3 元组 …… 59
3.1.4 域 …… 60
3.1.5 关系的等价表示法 …… 60
3.1.6 关系实例 …… 61
3.1.7 本节练习 …… 62
3.2 从 ODL 设计到关系设计 …… 62
3.2.1 从 ODL 属性到关系属性 …… 63
3.2.2 类中的非原子属性 …… 63
3.2.3 其他类型构造符的表示 …… 66
3.2.4 单值联系的表示 …… 67
3.2.5 多值联系的表示 …… 68
3.2.6 假如没有键码 …… 69
3.2.7 联系与反向联系的表示 …… 70
3.2.8 本节练习 …… 71
3.3 从 E/R 图到关系的设计 …… 72
3.3.1 实体集到关系的转换 …… 72
3.3.2 E/R 联系到关系的转换 …… 73
3.3.3 处理弱实体集 …… 75
3.3.4 本节练习 …… 77
3.4 子类结构到关系的转换 …… 78
3.4.1 用关系表示 ODL 子类 …… 78
3.4.2 在关系模型中表示“属于”联系 …… 79
3.4.3 方法的比较 …… 80
3.4.4 使用 NULL 值合并关系 …… 80
3.4.5 本节练习 …… 81
3.5 函数依赖 …… 82
3.5.1 函数依赖的定义 …… 82
3.5.2 关系的键码 …… 83
3.5.3 超键码 …… 84
3.5.4 寻找关系的键码 …… 85
3.5.5 由 ODL 设计导出的关系的键码 …… 86
3.5.6 本节练习 …… 87
3.6 函数依赖规则 …… 88
3.6.1 分解/合并规则 …… 88
3.6.2 平凡依赖 …… 89
3.6.3 计算属性的闭包 …… 90
3.6.4 传递规则 …… 92
3.6.5 函数依赖的闭包 …… 93
3.6.6 本节练习 …… 94
3.7 关系数据库模式设计 …… 95

3.7.1 异常 …… 96
3.7.2 关系分解 …… 96
3.7.3 BC 范式 …… 98
3.7.4 分解成 BCNF …… 99
3.7.5 函数依赖的投影 …… 102
3.7.6 从分解中恢复信息 …… 103
3.7.7 第三范式 …… 105
3.7.8 本节练习 …… 107
3.8 多值依赖 …… 108
3.8.1 属性的独立性及其带来的冗余 …… 108
3.8.2 多值依赖的定义 …… 109
3.8.3 多值依赖的推论 …… 111
3.8.4 第四范式 …… 112
3.8.5 分解成第四范式 …… 113
3.8.6 范式间的联系 …… 114
3.8.7 本节练习 …… 114
3.9 数据库模式实例 …… 116
3.10 本章总结 …… 118
3.11 本章参考文献 …… 119

第 4 章 关系模型中的运算 …… 121
4.1 关系代数 …… 121
4.1.1 关系的集合运算 …… 122
4.1.2 投影 …… 123
4.1.3 选择 …… 124
4.1.4 笛卡尔积 …… 124
4.1.5 自然连接 …… 125
4.1.6 θ连接 …… 127
4.1.7 查询中的复合运算 …… 128
4.1.8 改名 …… 129
4.1.9 基本和导出运算 …… 130
4.1.10 本节练习 …… 131
4.2 关系的逻辑 …… 136
4.2.1 谓词和原子 …… 136
4.2.2 算术原子 …… 137
4.2.3 Datalog 规则和查询 …… 137
4.2.4 Datalog 规则的含义 …… 138
4.2.5 外延和内涵谓词 …… 140
4.2.6 本节练习 …… 140
4.3 从关系代数到 Datalog …… 140
4.3.1 交集 …… 141
4.3.2 并集 …… 141

4.3.3 差集 …… 141
4.3.4 投影 …… 142
4.3.5 选择 …… 142
4.3.6 乘积 …… 144
4.3.7 连接 …… 144
4.3.8 用 Datalog 模拟多重运算 …… 145
4.3.9 本节练习 …… 146
4.4 Datalog 中的递归编程 …… 147
4.4.1 固定点运算符 …… 147
4.4.2 计算最小固定点 …… 148
4.4.3 Datalog 中的固定点方程 …… 149
4.4.4 递归规则中的求反 …… 153
4.4.5 本节练习 …… 156
4.5 对关系的约束 …… 157
4.5.1 用关系代数作为约束语言 …… 158
4.5.2 参照完整性约束 …… 158
4.5.3 附加约束的例子 …… 159
4.5.4 本节练习 …… 160
4.6 包的关系运算 …… 161
4.6.1 为什么用包? …… 162
4.6.2 包的并集、交集和差集 …… 163
4.6.3 包的投影 …… 164
4.6.4 包的选择 …… 165
4.6.5 包的乘积 …… 165
4.6.6 包的连接 …… 166
4.6.7 包的运算用于 Datalog 规则 …… 167
4.6.8 本节练习 …… 168
4.7 关系模型的其他外延 …… 169
4.7.1 更新 …… 169
4.7.2 聚合 …… 169
4.7.3 视图 …… 169
4.7.4 空值 …… 170
4.8 本章总结 …… 170
4.9 本章参考文献 …… 171

第 5 章 数据库语言 SQL …… 172
5.1 SQL 的简单查询 …… 172
5.1.1 SQL 的投影 …… 173
5.1.2 SQL 的选择 …… 175
5.1.3 字符串的比较 …… 176
5.1.4 日期和时间的比较 …… 178
5.1.5 输出的排序 …… 178

5.1.6 本节练习 …… 179
5.2 涉及多个关系的查询 …… 180
5.2.1 SQL 中的乘积和连接 …… 180
5.2.2 消除属性的二义性 …… 181
5.2.3 元组变量 …… 182
5.2.4 多关系查询的解释 …… 183
5.2.5 查询的并、交、差 …… 185
5.2.6 本节练习 …… 186
5.3 子查询 …… 188
5.3.1 产生标量值的子查询 …… 188
5.3.2 涉及到关系的条件 …… 189
5.3.3 涉及到元组的条件 …… 190
5.3.4 相关子查询 …… 191
5.3.5 本节练习 …… 192
5.4 副本 …… 193
5.4.1 副本的删除 …… 193
5.4.2 并、交、差中的副本 …… 194
5.4.3 本节练习 …… 195
5.5 聚合 …… 195
5.5.1 聚合运算符 …… 195
5.5.2 分组 …… 196
5.5.3 HAVING 子句 …… 198
5.5.4 本节练习 …… 199
5.6 数据库更新 …… 200
5.6.1 插入 …… 200
5.6.2 删除 …… 202
5.6.3 修改 …… 203
5.6.4 本节练习 …… 203
5.7 用 SQL 定义关系模式 …… 204
5.7.1 数据类型 …… 205
5.7.2 表的简单说明 …… 205
5.7.3 删除表 …… 206
5.7.4 更改关系模式 …… 206
5.7.5 默认值 …… 207
5.7.6 域 …… 207
5.7.7 索引 …… 208
5.7.8 本节练习 …… 209
5.8 视图的定义 …… 211
5.8.1 视图的说明 …… 211
5.8.2 视图的查询 …… 212
5.8.3 属性改名 …… 213
5.8.4 视图的更新 …… 213

5.8.5 对涉及到视图的查询的解释 …… 216
5.8.6 本节练习 …… 217
5.9 空值和外部连接 …… 218
5.9.1 对空值的运算 …… 218
5.9.2 真值 UNKNOWN …… 219
5.9.3 SQL2 中的连接表达式 …… 221
5.9.4 自然连接 …… 222
5.9.5 外部连接 …… 222
5.9.6 本节练习 …… 224
5.10 SQL3 中的递归 …… 225
5.10.1 在 SQL3 中定义 IDB 关系 …… 225
5.10.2 线性递归 …… 228
5.10.3 在 WITH 语句中使用视图 …… 228
5.10.4 分层求反 …… 229
5.10.5 SQL3 递归中的未定表达式 …… 230
5.10.6 本节练习 …… 232
5.11 本章总结 …… 233
5.12 本章参考文献 …… 235

第 6 章 SQL 中的约束和触发程序 …… 236
6.1 SQL 中的键码 …… 236
6.1.1 说明键码 …… 236
6.1.2 实施键码约束 …… 238
6.1.3 本节练习 …… 238
6.2 参照完整性和外键码 …… 238
6.2.1 说明外键码约束 …… 239
6.2.2 保持参照完整性 …… 240
6.2.3 本节练习 …… 241
6.3 对属性值的约束 …… 243
6.3.1 非空约束 …… 243
6.3.2 基于属性的 CHECK 约束 …… 243
6.3.3 域约束 …… 244
6.3.4 本节练习 …… 245
6.4 全局约束 …… 246
6.4.1 基于元组的 CHECK 约束 …… 246
6.4.2 断言 …… 247
6.4.3 本节练习 …… 250
6.5 约束的更新 …… 251
6.5.1 对约束命名 …… 251
6.5.2 更改表的约束 …… 252
6.5.3 更改域的约束 …… 253
6.5.4 更改断言 …… 253

6.5.5 本节练习 …… 253
6.6 SQL3 中的触发程序 …… 254
6.6.1 触发和约束 …… 254
6.6.2 SQL3 触发程序 …… 254
6.6.3 SQL3 的断言 …… 257
6.6.4 本节练习 …… 258
6.7 本章总结 …… 259
6.8 本章参考文献 …… 260

第 7 章 SQL 系统概况 …… 261
7.1 编程环境中的 SQL …… 261
7.1.1 匹配失衡问题 …… 262
7.1.2 SQL/宿主语言接口 …… 262
7.1.3 说明(DECLARE)段 …… 263
7.1.4 使用共享变量 …… 263
7.1.5 单行查询语句 …… 264
7.1.6 游标 …… 265
7.1.7 通过游标的更新 …… 267
7.1.8 游标选项 …… 267
7.1.9 为取出的元组排序 …… 268
7.1.10 防止并发更新的保护措施 …… 269
7.1.11 滚动游标 …… 269
7.1.12 动态 SQL …… 270
7.1.13 本节练习 …… 271
7.2 SQL 中的事务 …… 273
7.2.1 可串行性 …… 273
7.2.2 原子性 …… 275
7.2.3 事务 …… 276
7.2.4 只读事务 …… 277
7.2.5 读脏数据 …… 278
7.2.6 其他隔离性级别 …… 279
7.2.7 本节练习 …… 280
7.3 SQL 环境 …… 281
7.3.1 环境 …… 281
7.3.2 模式 …… 282
7.3.3 目录 …… 283
7.3.4 SQL 环境中的客户程序和服务程序 …… 284
7.3.5 连接 …… 284
7.3.6 会话 …… 285
7.3.7 模块 …… 285
7.4 SQL2 的安全和用户授权 …… 286
7.4.1 权限 …… 286

需要全本请访问 www.ertongbook.com

7.4.2 建立权限 …… 287
7.4.3 权限检验处理 …… 287
7.4.4 授予权限 …… 289
7.4.5 授权图 …… 290
7.4.6 取消权限 …… 290
7.4.7 本节练习 …… 293
7.5 本章总结 …… 294
7.6 本章参考文献 …… 296

第8章 面向对象查询语言 …… 297
8.1 ODL 中相关查询的特性 …… 297
8.1.1 ODL 对象的操作 …… 297
8.1.2 ODL 中方法署名的说明 …… 298
8.1.3 类的范围 …… 300
8.1.4 本节练习 …… 300
8.2 OQL 介绍 …… 301
8.2.1 面向对象的电影实例 …… 302
8.2.2 OQL 类型系统 …… 302
8.2.3 路径表达式 …… 303
8.2.4 OQL 中的 select-from-where 表达式 …… 304
8.2.5 消除重复 …… 305
8.2.6 复杂的输出类型 …… 305
8.2.7 子查询 …… 306
8.2.8 对结果排序 …… 307
8.2.9 本节练习 …… 307
8.3 OQL 表达式的附加格式 …… 308
8.3.1 量词表达式 …… 308
8.3.2 聚合表达式 …… 308
8.3.3 分组表达式 …… 309
8.3.4 HAVING 子句 …… 311
8.3.5 集合运算符 …… 311
8.3.6 本节练习 …… 312
8.4 OQL 中对象的赋值和建立 …… 313
8.4.1 对宿主语言变量赋值 …… 313
8.4.2 从聚集中提取元素 …… 313
8.4.3 获取聚集的每个成员 …… 314
8.4.4 建立新对象 …… 314
8.4.5 本节练习 …… 316
8.5 SQL3 中的元组对象 …… 316
8.5.1 行类型 …… 317
8.5.2 说明具有行类型的关系 …… 317
8.5.3 访问行类型的分量 …… 318

8.5.4 引用 …… 318
8.5.5 利用引用 …… 320
8.5.6 引用的作用域 …… 320
8.5.7 作为值的对象标识 …… 321
8.5.8 本节练习 …… 323
8.6 SQL3 的抽象数据类型 …… 324
8.6.1 ADT 的定义 …… 324
8.6.2 ADT 方法的定义 …… 327
8.6.3 外部函数 …… 329
8.6.4 本节练习 …… 329
8.7 ODL/OQL 和 SQL3 方法的比较 …… 330
8.8 本章总结 …… 331
8.9 本章参考文献 …… 332

前　　言

本书是从斯坦福大学的“数据库入门”(CS145)的课程笔记演变而来的。CS145是五门系列课程的第一门。[①] 由于Arthur Keller颇有创意的讲授,使这门课逐步发展成着重于数据库的设计和编程的课程,而这两方面内容对计算机科学专业的大多数学生来说是最有用的。这门课还包括一个内容广泛、不断滚动的课外工程项目,供学生设计并实现一个具体的数据库应用。与该工程项目相关的作业、其他课外作业、测验以及其他课程资料都可以从本书的主页上得到;请参阅“万维网(World Wide Web)上的支持”部分。

本书的使用

本书适用于讲授一学期的课程。如果像CS145这样按四分之一学年(译注:每学年分为四学期,这种制度的一学期)的课程来安排,就不得不省略或跳过书中的某些内容。最好由教师自己决定削减哪些内容,但以下内容显然是可以削减的:有关Datalog的部分,SQL编程的高级部分以及SQL3的细节部分。

如果课程中安排了不断滚动的工程项目,那么,提早讲授SQL语句部分是很重要的。可以推后讲授的内容包括:有关Datalog的部分,第5章和第6章的SQL3部分以及第3章的某些理论部分(但是,如果学生们在开始进行SQL编程之前,想设计出优秀的关系模式的话,他们就需要规范化的知识,或许还需要多值依赖的知识)。

预备知识

我们把本书定位于“夹层”水平,即高年级本科生和低年级研究生水平。这门课程正规的预备知识相当于大学二年级的水平:(1) 数据结构、算法和离散数学;(2)软件系统、软件工程和编程语言。学生们对以下内容至少要有初步的了解:代数的表达式和定律、逻辑、基本数据结构(如搜索树)、面向对象的编程概念以及编程环境。我们相信,按照典型的计算机科学专业的教学计划,到大学三年级结束时,学生们肯定会拥有充分的背景知识。

练　　习

本书包括多方面的练习,几乎每节都有。我们把比较难的练习或练习中比较难的部分

① 后面的四门是:数据库系统原理,数据库系统实现的工程训练,事务和分布式数据库,以及数据库理论。

用惊叹号(!)标出。最难的练习用双惊叹号(!!)标出。

有一些练习或练习的某些部分标有星号(*)。对于这些练习,我们将尽量通过本书的主页提供解答。这些解答是公开的,并可用于自我检测。注意:在某些情况下,练习B要求您对另一个练习A的解答进行修改或改进。如果A的某些特定部分有解答,那么B的相应部分也将有解答。

万维网上的支持

本书的主页是:

http://www-db.stanford.edu/~ullman/fcdb.html

这里有带星号的练习的解答,对已发现的书写或印刷错误的勘误表以及辅助教材。我们希望每一个像我们一样讲授CS145课程的人都能获得这些课程笔记,包括课外作业、解答和工程项目的作业。

致　　谢

特别感谢Bobbie Cochrane和Linda DeMichiel,感谢他们在SQL3标准方面给予的帮助。还有其他许多人帮助我们审校手稿,他们是:Donald Aingworth, Jonathan Becker, Larry Bonham, Christopher Chan, Oliver Duschka, Greg Fichtenholtz, Bart Fisher, Meredith Goldsmith, Steve Huntsberry, Leonard Jacobson, Thulasiraman Jeyaraman, dwight Joe, Seth Katz, Brian Kulman, Le-Wei Mo, Mark Mortensen, Ramprakash Narayanaswami, Torbjorn Norbye, Mehul Patel, Catherine Tornabene, Jonathan Ullman, Mayank Upadhyay, Vassilis Vassalos, Qiang Wang, Sundar Yamunachari和Takeshi Yokukawa。当然,剩下的错误由我们负责。

J. D. U.

J. W.

第 1 章　数据库系统的世界

通过本书,读者可以学会如何有效地使用数据库管理系统,包括数据库的设计和对数据库操作的编程。本章主要介绍各种重要的数据库概念。简短的历史回顾之后,我们将了解数据库系统与其他软件风格的区别。在这一章里,还将介绍支持数据库及其应用的数据库管理系统的实现背景。如果我们要正确评价为什么要设计各种各样的数据库、为什么要对数据库的操作有所限制,对其"内幕"的理解是非常重要的。最后,我们将回顾一些读者可能比较熟悉,但对后续章节而言又必不可少的思想,比如"面向对象的程序设计"等。

1.1　数据库系统的发展

数据库是什么呢? 实质上,数据库只不过是一些存在了很长时间——常常是许多年——的信息的聚集。通常意义下,"数据库"这个术语是指由数据库管理系统(database management system,简称为 DBMS,或称为数据库系统)管理的数据聚集。一个数据库系统应该是:

1. 允许用户用一种叫做数据定义语言(data definition language)的专用语言,建立新的数据库和指定它们的模式(schema)(数据的逻辑结构)。

2. 使用户能够用适当的语言查询数据("查询"(query)是一个数据库术语,指对数据的某种询问)和更新数据,所使用的语言通常称为"查询语言"或"数据操作语言"(data manipulation language)。

3. 支持存储大量的数据——G(吉,10^9)字节以上——经过很长一段时间以后,仍保证安全,使其免遭意外或非授权的使用,同时允许对数据库查询和更新的有效访问。

4. 控制多用户的同时访问,使得一个用户的访问不影响其他用户,保证同时访问不会损坏数据。

1.1.1　早期的数据库管理系统

第一批商用数据库管理系统出现在 20 世纪 60 年代后期。它们由文件系统演变而来。文件系统能满足以上第 3 项的要求——长时间地存储数据,并且能存储大量的数据。但是,如果没有备份的话,文件系统通常并不能保证数据不丢失;如果不知道数据所在的特定文件,文件系统也不能支持有效的数据访问。

另外,文件系统并不能直接满足第 2 项关于查询语言对数据的查询要求。它们对第 1 项——数据模式的支持也仅限于建立文件的目录结构。最后,文件系统不支持第 4 项。如果允许多个用户或进程对文件进行并发访问,由于文件系统一般不能避免两个用户几乎

同时修改同一文件的事件发生,因此,必然有一个用户的修改会失败,无法在文件中反映出来。

早期 DBMS 最重要的应用是在那些数据中包含很多小的数据项并且需要许多查询和更新的场合。下面是一些应用的实例。

飞机订票系统

在这里,每个数据包括如下数据项:

(1) 一个旅客对一次航班的座位预定,包括分配座位、进餐选择等信息;

(2) 航班信息——出发地和目的地,起飞和到达的时间,或飞机已经起飞等等;

(3) 机票信息——票价、要求、有无等。

典型的查询是在某一段时间内从某个给定的城市飞往另一个城市的航班,还有什么座位可供选择,以及机票价格。典型的数据更新包括为旅客登记航班,分配座位,选择餐饮。任何时刻,都会有许多代理来访问数据的某些部分。DBMS 必须允许这种"并发访问";还要避免一些诸如"两个代理同时分配了同一个座位"之类的问题;当系统突然崩溃时,DBMS 还要避免记录的丢失。

银行系统

数据项包括顾客的姓名、地址、帐号、存款、结余以及顾客与他们的帐号和存款之间的关系,比如,谁对那些帐号有签名权。对结余的查询固然不少,但更多的是针对一次存款或取款所进行的修改。

正如飞机订票系统一样,我们希望出纳员和顾客通过 ATM(Automated Teller Machine,自动出纳机)能同时查询、更新银行的数据。对同一帐号能同时访问,而不影响 ATM 业务,这一点是至关重要的。错误是不能容忍的。例如,一旦钱从 ATM 自动出纳机中弹出,银行必须记录这项支出,即使立刻掉电也不例外。正确处理这种操作远不像想象的那么简单,可以看作是 DBMS 系统结构的重大进展之一。

公司记录

许多早期应用都与公司记录有关,比如每次销售的记录,帐户的应付或应收信息,雇员信息——他们的姓名、地址、工资、津贴、税款等等。查询的内容包括打印帐户收入或雇员每周工资之类的记录。每次销售、采购,每张帐单、收据,雇员的雇用、解雇、提升等等,都将导致数据库的更新。

从文件系统发展而来的早期的 DBMS 希望用户把数据想象成很像它存放的样子。这些数据库系统使用几种不同的数据模型来描述数据库的信息结构,其中主要是基于树的层次模型和基于图的网状模型。后者在 60 年代末期通过 CODASYL[①](数据系统和语言协会,Committee on Data Systems and Languages)的一份报告进行了标准化。关于网状和层次模型,读者可以参考 2.7 节,尽管今天它们已经成了历史话题。

① CODASYL Data Base Task Group April 1971 Report, ACM, New York.

需要全本请访问 www.ertongbook.com