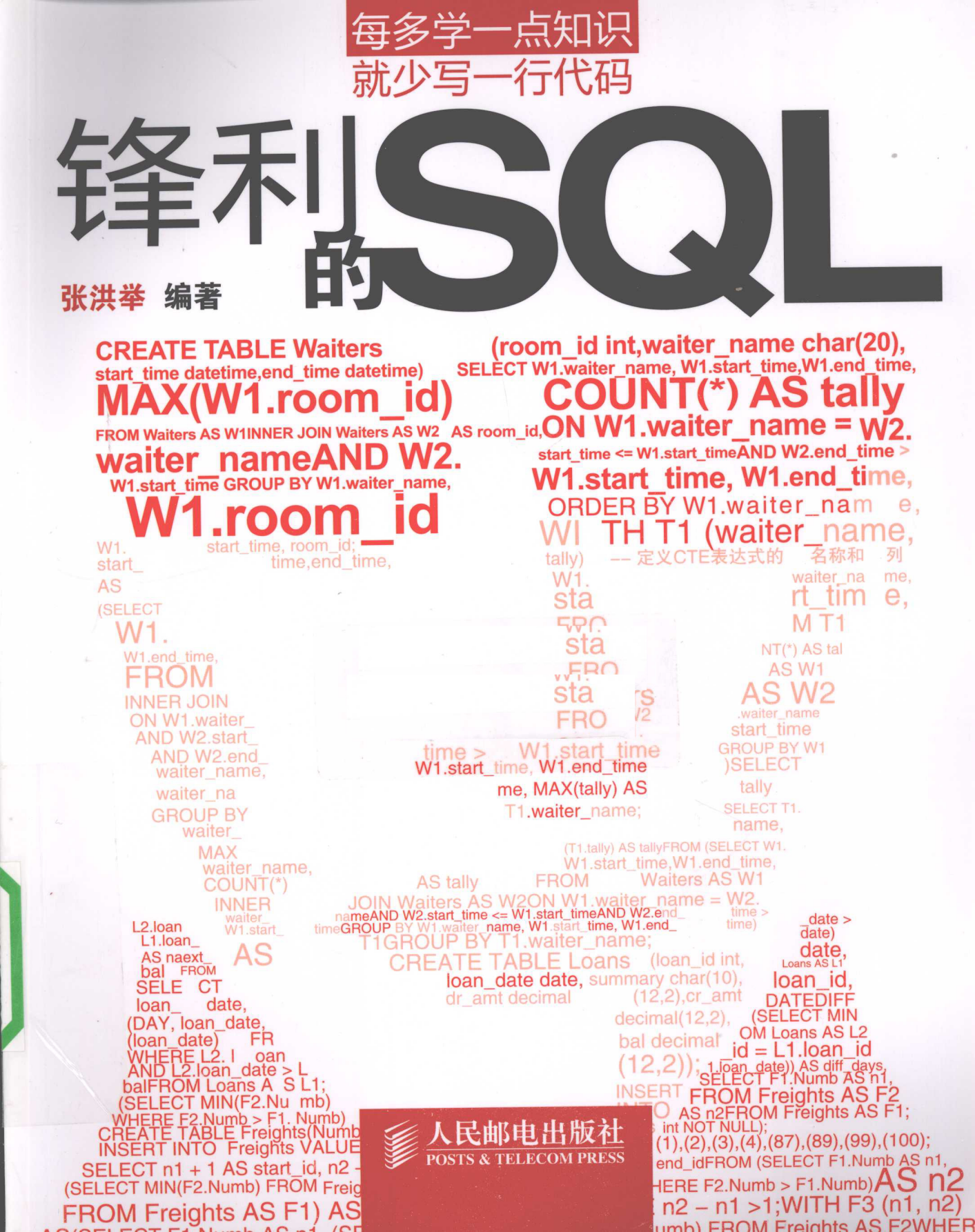
每多学一点知识
就少写一行代码

锋利的SQL

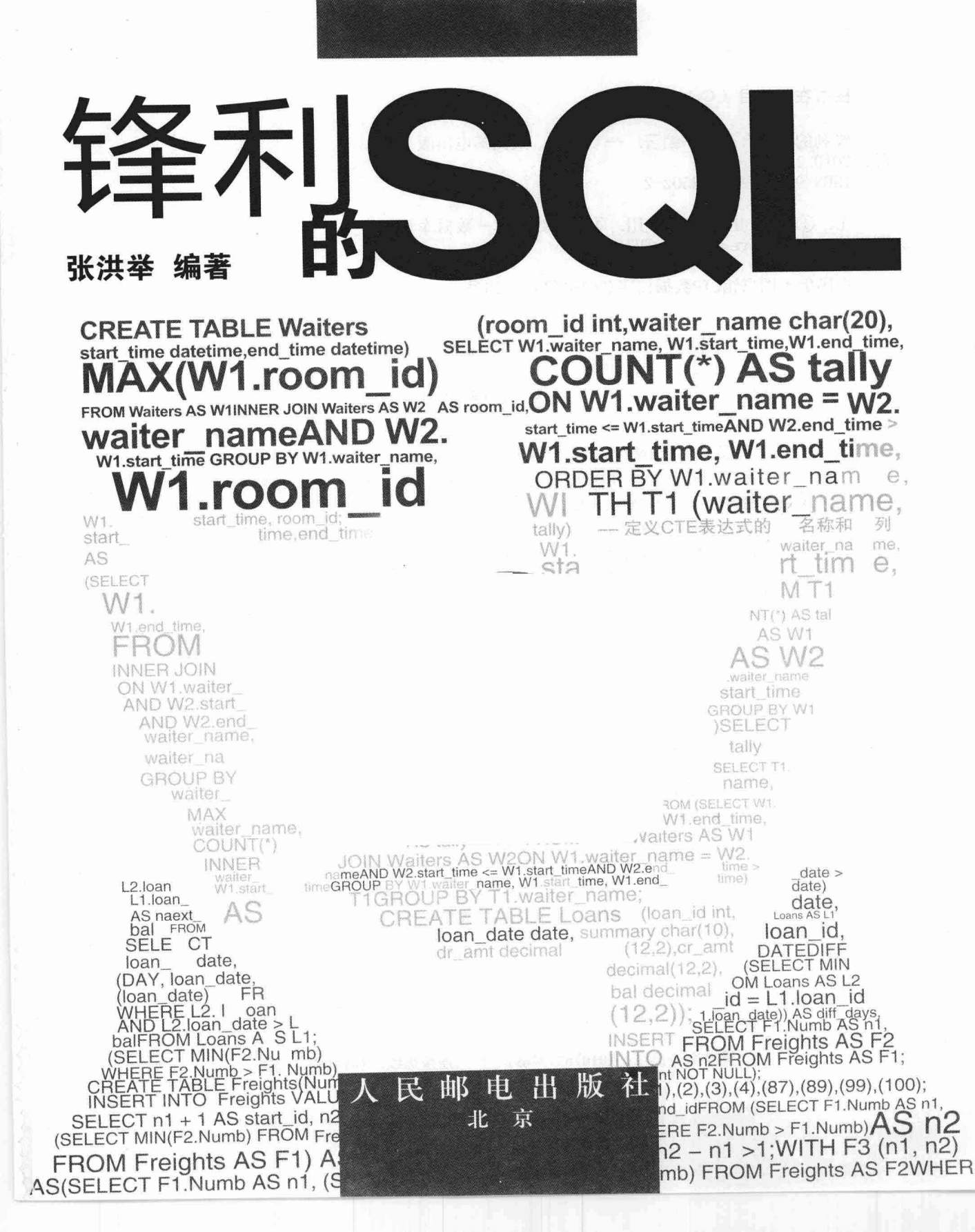
张洪举 编著



人民邮电出版社
POSTS & TELECOM PRESS

锋利的SQL

张洪举 编著



人民邮电出版社
北京

图书在版编目（CIP）数据

锋利的SQL / 张洪举编著. -- 北京 : 人民邮电出版社, 2010.11
ISBN 978-7-115-23502-2

Ⅰ. ①锋… Ⅱ. ①张… Ⅲ. ①关系数据库－数据库管理系统，SQL Server Ⅳ. ①TP311.138

中国版本图书馆CIP数据核字(2010)第160625号

内 容 提 要

本书从基础、开发、性能调整和实战4个方面介绍了SQL技术及其应用，包括数据库管理、表管理、索引管理、基本查询、子查询、联接和APPLY运算符、操作结果集、窗口计算和表旋转、数据修改、视图、游标、存储过程、触发器、用户自定义函数、事务处理、并发访问控制、查询的优化与执行等内容。

本书既覆盖了改善效率和性能的普通SQL技术，也深入探讨了SQL新技术，更包含一些实用的查询解决方案，希望本书能够成为引领读者进入SQL查询殿堂的捷径。

锋利的SQL

◆ 编　　著　张洪举
责任编辑　杜　洁
◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街14号
邮编　100061　　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京艺辉印刷有限公司印刷
◆ 开本：800×1000　1/16
印张：24.5
字数：595千字　　2010年11月第1版
印数：1－4 000册　　2010年11月北京第1次印刷

ISBN 978-7-115-23502-2

定价：49.00元

读者服务热线：(010)67132692　印装质量热线：(010)67129223
反盗版热线：(010)67171154

前　言

从2009年年初开始动笔到最终定稿，本书的编写花费了我一年多的时间，大大超出了先前预期的计划。当时，书刚刚开了个头，我便被安排参加了单位的一个流程优化项目，其中包含5个新系统的开发和1个旧系统的改造。同时，我自己还主持着一个管理信息系统的开发工作。项目目标定义、业务需求分析、系统架构设计，环环相扣，整天忙得不可开交，写作成了真正忙里偷闲的事。

当系统一个个瓜熟蒂落，能够自己支配的时间才稍微充裕了一些，这本书的写作才算步入正轨。好在写作只是我的一种业余爱好，没有其他方面的压力，唯一的目的就是希望能够创作一些精品图书。

之所以要写作这本图书，主要出于两方面的原因：一是伴随着各种数据库技术日新月异的发展，无论是哪种数据库产品，想用有限的篇幅去描述它的全貌，几乎是不可能完成的任务。所以我就在考虑能否抽取出各种数据库产品中一些大家共同关心的内容，进行深入细致的挖掘，而SQL无疑是这方面的首选。二是在与一些开发公司的合作中，发现公司间的SQL应用情况也差异很大，一些公司出于产品的可移植性考虑，拒绝使用一些新的SQL技术，甚至尽量避免在服务器上部署存储过程。所以，我希望在深入地讨论一些常用SQL技术的同时，也尽可能多地介绍一些SQL的新技术，从而解除大家对新技术的恐惧感，对新技术的推广能够起到一定的推波助澜作用。

本书特点

本书既覆盖了改善效率和性能的普通SQL技术，也深入探讨了SQL新技术，更包含一些实用的查询解决方案，希望本书能够成为引领读者进入SQL查询殿堂的捷径。

本书的内容是基于SQL Server数据库产品进行讨论的。不过，无论是哪种数据库产品的SQL，由于大家都在遵循ANSI-SQL标准，所以差别并不大。数据库开发人员在跨越不同的数据库产品时，一般不会遇到什么障碍。

本书在介绍各种查询语法时，更注重对查询逻辑思维方式的引导和介绍。只有这样，才可以帮助读者在阅读之后举一反三，提升自己动手解决实际问题的能力。

本书适用读者

本书是按照由浅入深、循序渐进方式对SQL进行介绍的，既包含了入门知识，也包含了深层次技术的讨论。即使是最基本的查询语句，我们也会尽力为读者提供解决深层次问题的能力。也就是说，同样一个问题，开发人员可以写几十甚至上百行的SQL语句来解决问题，也有可能仅通过一条SELECT语句就可以解决问题。作为SQL而言，虽然代码最简化并不一定是性能最优化，但至少是对思维能力的一种提升。

从这个角度而言，本书可以作为SQL入门书籍，也可以作为SQL程序员、DBA的参考书籍。

本书内容与结构

本书共 19 章，可大致分为基础篇、开发篇、性能调整篇和实战篇，共 4 部分。

基础篇包括第 1 章至第 10 章。其中，第 1 章是对查询工具、书写规范等基本内容的介绍，第 2 章至第 5 章是对数据库、表和索引的介绍，第 5 章至第 10 章则介绍了使用 SELECT 进行数据查询和使用 INSERT、UPDATE、DELETE 进行数据修改的各个方面。

开发篇包括第 11 章至第 15 章。如果将 SELECT、INSERT、UPDATE 和 DELETE 作为基本查询语句，则 IF...ELSE、WHILE 和 TRY...CATCH 构造等则可以看作是 SQL 编程语句。在存储过程、触发器等对象中可以通过这些语句实现一些复杂的逻辑处理。如果你曾经是一位使用 VB 或 VC 程序员，在学习 C/S 或 B/S 编程时，你应当掌握这种服务器端的编程工具，从而将业务逻辑计算合理地分布到服务器和客户端。

性能调整篇包括第 16 章至第 18 章。第 16 章和第 17 章介绍的是事务处理机制和并发访问控制。其实，无论是微软还是 Oracle、IBM，其数据库产品的核心功能都是一样的，即在保证数据完整性的前提下提供最大的并发支持。数据库系统是通过“锁机制”来实现的，数据库引擎都提供有多种粒度的锁定模式，从而允许用户可以根据需要将资源锁定在适当的级别，尽量减少锁定开销。第 18 章则是讨论了查询优化器的工作原理，重用查询计划，可以减少额外的编译开销，提高服务器性能。

实战篇仅包含第 19 章的内容，提供了同一时间范围内并发数统计、时间段天数统计、数字范围统计、地域范围内最大数统计等较为常用问题的解决方案。

本书声明

本书实例中使用的操作系统是 Windows 7，使用的数据库是 SQL Server 2008 开发者版本，开发工具是 Visual Studio 2008。

感谢

在本书的完成过程中，得到了诸多 SQL Server 技术专家和爱好者的支持和帮助。他们无私和热情的参与，使本书的内容更加实用和更具指导性，在此一并表示感谢。他们是王向东、秦广、魏兰花、凌亚东、王亚羽、陈雨薇、王光辉、高存亭、桑晓红、王新河、张宪国、李联国、韩燕军。

由于时间仓促，加之水平有限，书中不足之处在所难免，敬请读者批评指正。

编　者

2010 年 9 月

目 录

基 础 篇

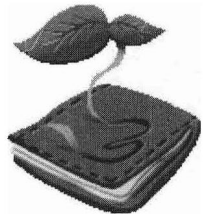
开 发 篇

性能调整篇

实 战 篇

1

第 1 部分 基础篇



第1章 SQL简介

SQL的全称是结构化查询语言（Structured Query Language），这是一种非常易读的语言，只要稍微有一点英语基础，一些简单的数据查询、操作语句几乎都可以理解。但是，要想精通SQL，也并不是一件很容易的事情，因为在数据处理方面有许许多多的个案。要处理好这些个案，除了具有缜密的逻辑思维，还需要多练习和实践，从而增强记忆。从本章开始，打好坚实的基础，并在此基础上不断提升自己的理论知识体系，当感受某个成功喜悦的时候，或许就会发现自己已经站在了SQL的巅峰。

在本章我们将讲述一些最基本的SQL内容，如SQL的历史起源、ANSI是什么，以及SQL的语法元素和执行SQL的工具等。对于基本知识，学习起来可能比较枯燥。但是，如果你是初学SQL，这些基本知识对于学习好本书的后续内容，将起着至关重要的作用。

1.1 SQL的历史起源

在20世纪60年代，网状数据库系统（如CODASYL）和分层数据库系统（如IMS™）是用于自动化银行业务、记账和订单处理系统的一流技术，这些系统是由于商业大型计算机的引入才启用的。而SQL是在70年代创建的一种基于关系数据库管理系统（Relational Database Management System，RDBMS）模型的数据查询、操作语言。

1.1.1 CODASYL

CODASYL是美国数据系统语言协会（Conference on Data System Language）的英文缩写，该协会成立于1957年，主要目的是为了开发一种用于创建商业应用的通用语言。1959年5月28日该协会召开了首次会议，就语言开发进行讨论。这个语言实际上就是Cobol语言。

1963年6月10日，加利福尼亚州的系统开发公司（System Development Corporation）举办了一个题为“基于计算机的数据库开发和管理”（Development and Management of a Computer-centered Data Base）的研讨会，首次提出并定义了数据库（Database）术语，即：一组文件（表）的集合，其中文件是数据项（行）的有序集合，而每个数据项由数据以及一个或多个键组成。

1965年，CODASYL成立了“列表处理任务组”（List Processing Task Force），后更名为“数据

CODASYL 专家。

此时的 E. F. Codd 博士在 IBM 的 San Jose 研究中心（即现在的 Almaden 研究中心，http://www.almaden.ibm.com/）工作，1970 年 6 月，他发表了里程碑性的论文《大型共享数据库数据的关系模型》，确立了关系数据库的概念。但是，由于 IBM 正在从事 IMS 的开发，这种思想对 IBM 本身产品造成了威胁，所以公司内部最初持压制态度。当然这也与 Codd 采用了大量的数学方法，不容易理解有关。

1973 年，IBM 在外部竞争压力下，开始加强在关系数据库方面的投入。Chamberlin 被调到 San Jose 研究中心，加入新成立的项目 System R。System R 基于 Codd 提出的关系数据库管理系统模型。

System R 项目包括研究高层的关系数据系统（Relational Data System，RDS）和研究底层的存储系统（Research Storage System，RSS）两个小组，Chamberlin 担任 RDS 组的经理。RDS 实际上就是一个数据库语言编译器，由于 Codd 提出的关系代数和关系演算过于数学化，影响了易用性。于是 Chamberlin 选择了自然语言作为研究方向，其结果就是诞生了结构化英语查询语言（Structured English Query Language，SEQUEL）。后来，由于商标之争，SEQUEL 更名为 SQL。

System R 是一个具有开创性意义的项目。它第一次实现了结构化查询语言，并具以成为标准的关系数据查询语言。同时，它也是第一个证明了关系数据库管理系统可以提供良好事务处理性能的系统。System R 系统中的设计决策，以及一些基本算法选择（如查询优化中的动态编程算法）对以后的关系系统都产生了积极影响。

System R 本身作为原型虽然并未问世，但鉴于其影响，计算机协会（Association for Computing Machinery，ACM，http://www.acm.org/）还是把 1988 年的“软件系统奖”授予了 System R 开发小组。

1.1.4 ANSI 和 SQL 方言

ANSI 是美国国家标准学会（American National Standards Institute）的英文简称，成立于 1918 年。当时，美国的许多企业和专业技术团体，已开始了标准化工作，但因彼此间没有协调，存在不少矛盾和问题。为了进一步提高效率，数百个科技学会、协会组织和团体，均认为有必要成立一个专门的标准化机构，并制定统一的通用标准。1918 年，美国材料试验协会（ASTM）、美国机械工程师协会（ASME）、美国矿业与冶金工程师协会（ASMME）、美国土木工程师协会（ASCE）和美国电气工程师协会（AIEE）等组织，共同成立了美国工程标准委员会（AESC）。美国政府的商务部、陆军部和海军部也参与了该委员会的筹备工作。1928 年，美国工程标准委员会改组为美国标准协会（ASA）。为致力于国际标准化事业和消费品方面的标准化，1966 年 8 月，又改组为美利坚合众国标准学会（USASI）。1969 年 10 月 6 日改成现名：美国国家标准学会（ANSI）。

虽然 IBM 首创了关系数据库理论，但 Oracle 却是第一家在市场上推出了这套技术的公司。随着时间的推移，SQL 的简洁、直观，在市场上获得了不错的反响，从而引起了 ANSI 的关注，分别在 1986 年、1989 年、1992 年、1999 年及 2003 年发布了 SQL 标准。SQL Server 2000 遵循 ANSI

SQL:1992 标准，而 SQL Server 2005 和 2008 还实现了 ANSI SQL:1999 和 ANSI SQL:2003 中的一些重要特性。

数据库生产商在遵循 ANSI 标准的同时，也会根据自己产品的特点对 SQL 进行了一些改进和增强，于是也就有了 SQL Server 的 Transact-SQL、Oracle 的 PL/SQL 等语言，我们称之为 SQL 方言。在本书中，我们将以 Transact-SQL 为基础进行 SQL 语言的介绍。实际上，在学习过程中，大家也没有必要刻意关心哪些语句或关键字是 SQL 标准，哪些是 Transact-SQL 的扩展。其实常见的数据库操作，在绝大多数支持 SQL 语言的数据库中差别并不大，所以数据库开发人员在跨越不同的数据库产品时，一般不会遇到什么障碍。但是对于数据库管理员来说，则需要面对很多挑战，不同数据库产品在管理、维护和性能调整方面区别很大。

1.2 Transact-SQL 语言的类型

在介绍了 SQL 的起源后，来看一下 Transact-SQL 包括哪些语言类型。首先，为了遵循 ANSI SQL 标准，Transact-SQL 提供了数据定义语言（Data Definition language，DDL）语句和数据操作语言（Data Manipulation Language，DML）语句；其次，为了增强灵活性，Transact-SQL 还提供了用于编程的流控制语句和其他语句。

对于语言类型，读者仅作大致了解就可以。在实际应用中，就像没必要区分哪些是 SQL 标准，哪些是 SQL 扩展一样，也没有必要区分 DDL 和 DML，它们是一个协同工作的整体。

1.2.1 DDL 语句

DDL 语句用于创建数据库对象，如表、视图、索引等，表 1-1 中列出了一些常用的 DDL 语句。

表 1-1 DDL 常用语句

语　句	作　用
ALTER DATABASE	修改一个数据库或与该数据库关联的文件和文件组
ALTER FUNCTION	更改用户定义函数
ALTER LOGIN	更改 Microsoft Windows 或数据库服务器登录账户的属性
ALTER PROCEDURE	修改存储过程
ALTER SCHEMA	更改架构所有者
ALTER TABLE	通过修改、添加或删除列和约束来修改表定义
ALTER TRIGGER	更改 DML 或 DDL 触发器的定义
ALTER VIEW	修改先前创建的视图
CREATE DATABASE	创建新的数据库和用来存储数据库的文件
CREATE FUNCTION	创建用户定义函数
CREATE INDEX	为指定表或视图创建关系索引，或为指定表创建 XML 索引

续表

语　句	作　用
CREATE LOGIN	创建新的 Microsoft Windows 或数据库服务器登录账户
CREATE PROCEDURE	创建存储过程
CREATE RULE	创建规则对象
CREATE SCHEMA	在当前数据库中创建架构
CREATE TABLE	创建新表
CREATE TRIGGER	创建 DML 或 DDL 触发器
CREATE VIEW	创建一个虚拟表（视图），该表以一种备用方式提供一个或多个表中的数据
DROP DATABASE	从数据库服务器实例中删除一个或多个数据库或数据库快照
DROP FUNCTION	从当前数据库中删除一个或多个用户定义函数
DROP INDEX	从当前数据库中删除索引
DROP LOGIN	删除 Microsoft Windows 或数据库服务器登录账户
DROP RULE	从当前数据库中删除一个或多个用户定义的规则
DROP SCHEMA	从数据库中删除架构
DROP TABLE	删除表的表定义和所有数据、索引、约束和权限规范
DROP TRIGGER	从当前数据库中删除一个或多个 DML 或 DDL 触发器
DROP VIEW	从当前数据库中删除一个或多个视图

1.2.2　DML 语句

DML 语句用来检索和修改数据库的内容，表 1-2 中列出了一些常用的 DML 语句。

表 1-2　　DML 常用语句

语　句	作　用
BEGIN TRANSACTION	显式开始一个本地事务
CLOSE	释放当前结果集，然后解除定位游标的行上的游标锁定，从而关闭一个开放的游标
COMMIT	结束事务成功并提交
DELETE	从表或视图中删除行
INSERT	添加新行到表或视图中
READTEXT	从 text、ntext 或 image 列读取 text、ntext 或 image 值，从指定的偏移量开始读取指定的字节数
ROLLBACK	回滚事务
SAVE TRANSACTION	在事务内设置保存点
SELECT	从数据库中检索行，并允许从一个或多个表中选择一个或多个行或列