



达梦数据库7.1标准版及例题源码



达梦数据库

达梦

DAMENG  
DATABASE

数据库

SQL 指南

主 编 © 吴照林 戴剑伟

→ DM-SQL 函数语法及应用

→ DM SQL 语句语法及应用

→ DM SQL 程序设计

→ DM 嵌入式程序设计



中国工信出版集团



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>

# 达梦数据库 SQL 指南

主 编 吴照林 戴剑伟  
副主编 周 淳 朱明东 文 峰  
编撰人员 (按姓氏笔画排序)

王 强 文 峰 冯勤群 左青云  
朱明东 刘培磊 吴照林 张 胜  
张海粟 张守帅 周 淳 周英彪  
徐 飞 龚建华 曾昭文 戴剑伟



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书以达梦数据库管理系统 DM7.1 为蓝本, 系统介绍了 DM SQL 的功能、语法、应用及 DM SQL 程序设计方法。全书包括 9 章, 主要内容包括 DM SQL 函数、DM SQL 数据定义、数据查询、数据操作和事务处理、数据控制等 SQL 语句的语法及应用, 以及存储过程、存储函数、触发器、包、嵌入式 SQL 等数据库程序设计方法。附光盘 1 张, 内含达梦数据库管理系统 7.1 标准版及例题源码。

本书内容实用、操作性强, 语言通俗、格式规范, 可作为相关专业的教材, 也可作为工程技术人员的参考书。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有, 侵权必究。

## 图书在版编目 (CIP) 数据

达梦数据库 SQL 指南 / 吴照林, 戴剑伟主编. —北京: 电子工业出版社, 2016.11  
ISBN 978-7-121-30211-4

I. ①达… II. ①吴… ②戴… III. ①关系数据库系统—指南 IV. ①TP311.138-62

中国版本图书馆 CIP 数据核字 (2016) 第 257858 号

策划编辑: 李 敏

责任编辑: 郝黎明

印 刷: 三河市华成印务有限公司

装 订: 三河市华成印务有限公司

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: 787×1 092 1/16 印张: 20.25 字数: 468 千字

版 次: 2016 年 11 月第 1 版

印 次: 2016 年 11 月第 1 次印刷

定 价: 59.00 元 (含光盘 1 张)

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 [zltz@phei.com.cn](mailto:zltz@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

本书咨询联系方式: 010-88254753 或 [limin@phei.com.cn](mailto:limin@phei.com.cn)。

# 前 言

发展具有自主知识产权的国产数据库管理系统，打破国外数据库产品的垄断，为我国信息化建设提供安全可控的基础软件，是维护国家信息安全的重要手段。

达梦数据库管理系统作为国内最早推出的具有自主知识产权的数据库管理系统之一，是唯一获得国家自主原创产品认证的数据库产品，现已在公安、电力、铁路、航空、审计、通信、金融、海关、国土资源、电子政务等多个领域得到广泛应用，为国家机关、各级政府和企业信息化建设发挥了积极作用。

为了推动国产数据库管理系统的教学和人才培养，促进国产数据库的广泛应用，我们在总结数据库管理系统长期教学和科研实践经验的基础上，在达梦数据库有限公司的大力支持下，以达梦数据库 DM7.1 为蓝本，编写了《达梦数据库应用基础》和《达梦数据库 SQL 指南》两本教材。

《达梦数据库 SQL 指南》系统介绍了 DM SQL 的功能、语法、应用和 DM 数据库程序设计，全书共 9 章。内容包括 DM SQL 概述，DM SQL 函数，DM SQL 数据定义，DM SQL 数据查询，DM SQL 数据操作和事务处理，DM SQL 数据控制，DM SQL 程序设计基础，DM SQL 程序设计，DM 嵌入式 SQL 程序设计。

为了方便读者学习和体验操作，此书还附带了光盘。光盘内包含了达梦数据库管理系统 7.1 标准版和例题源码。

本书纲目由吴照林、戴剑伟拟制，最后统稿修改由吴照林、戴剑伟完成。具体分工为：第 1 章由吴照林、戴剑伟编写，第 2 章由文峰、戴剑伟编写，第 3 章由龚建华、冯勤群编写，第 4 章由曾昭文、张守帅编写，第 5 章由戴剑伟、周淳编写，第 6 章由朱明东、龚建华编写，第 7 章和第 8 章由戴剑伟、张胜编写，第 9 章由冯勤群、戴剑伟、张海粟编写。左青云、王强、刘培磊编写了书中例题，并对例题进行了测试验证。

在本书的编写过程中，参考了达梦数据库有限公司提供的技术资料，在此表示衷心的感谢。

由于作者水平有限，书中难免有些错误与不妥之处，敬请读者批评指正，欢迎读者通过达梦数据库技术支持联系方式（见附录 B）或电子邮件 [djw@sohu.com](mailto:djw@sohu.com) 与我们交流。

作 者

2016 年 10 月于武汉

# 目 录

<b>第 1 章 DM SQL 概述</b> .....	1
1.1 SQL 的发展.....	1
1.2 DM SQL 的特点.....	2
1.3 DM SQL 的主要功能.....	2
1.4 DM SQL 数据类型.....	3
1.4.1 数值型数据类型.....	3
1.4.2 字符型数据类型.....	3
1.4.3 日期型数据类型.....	4
1.4.4 日期时间间隔数据类型.....	5
1.4.5 二进制和多媒体数据类型.....	9
1.5 DM SQL 表达式.....	9
1.5.1 数值表达式.....	9
1.5.2 字符串表达式.....	10
1.5.3 时间值表达式.....	10
1.5.4 时间间隔值表达式.....	12
1.5.5 运算符的优先级.....	13
1.6 SQL 语法描述说明.....	13
1.7 示例数据库说明.....	14
<b>第 2 章 DM SQL 函数</b> .....	17
2.1 数值函数.....	17
2.2 字符串函数.....	29
2.3 日期时间函数.....	50
2.4 空值判断函数.....	66
2.5 类型转换函数.....	68
2.6 杂类函数.....	71

第 3 章	DM SQL 数据定义	73
3.1	数据库修改语句	73
3.2	表空间管理语句	75
3.2.1	创建表空间	75
3.2.2	修改表空间	77
3.2.3	删除表空间	78
3.2.4	创建大表空间	79
3.2.5	删除大表空间	80
3.3	用户管理语句	80
3.3.1	创建用户	80
3.3.2	修改用户	84
3.3.3	删除用户	85
3.4	模式管理语句	86
3.4.1	创建模式	86
3.4.2	设置当前模式	87
3.4.3	删除模式	88
3.5	表管理语句	88
3.5.1	管理数据库表	89
3.5.2	管理外部表	101
3.6	视图管理语句	103
3.6.1	创建视图	103
3.6.2	删除视图	106
3.6.3	创建物化视图	107
3.6.4	修改物化视图	111
3.6.5	删除物化视图	111
3.7	索引管理语句	111
3.7.1	创建常用索引	112
3.7.2	删除常用索引	114
3.7.3	创建位图连接索引	114
3.7.4	删除位图连接索引	115
3.7.5	创建全文索引	116
3.7.6	修改全文索引	117
3.7.7	删除全文索引	118

3.8	序列管理语句 .....	119
3.8.1	创建序列 .....	120
3.8.2	删除序列 .....	121
3.9	同义词管理语句 .....	122
3.9.1	创建同义词 .....	122
3.9.2	删除同义词 .....	123
<b>第 4 章</b>	<b>DM SQL 数据查询 .....</b>	<b>124</b>
4.1	单表查询 .....	124
4.1.1	简单查询 .....	125
4.1.2	带条件查询 .....	125
4.1.3	集函数 .....	127
4.1.4	分析函数 .....	129
4.1.5	CASE 表达式 .....	141
4.2	连接查询 .....	143
4.2.1	交叉连接 .....	143
4.2.2	自然连接 (NATURAL JOIN) .....	144
4.2.3	JOIN...USING .....	145
4.2.4	JOIN...ON .....	145
4.2.5	自连接 .....	145
4.2.6	内连接 (INNER JOIN) .....	146
4.2.7	外连接 (OUTER JOIN) .....	146
4.2.8	哈希连接 (HASH JOIN) .....	147
4.3	子查询 .....	148
4.3.1	标量子查询 .....	148
4.3.2	表子查询 .....	149
4.3.3	派生表子查询 .....	150
4.3.4	定量比较 .....	151
4.3.5	带 EXISTS 谓词的子查询 .....	152
4.3.6	多列表子查询 .....	153
4.4	公用表表达式 .....	154
4.4.1	公用表表达式的作用 .....	155
4.4.2	公用表表达式的使用 .....	155
4.5	合并查询结果 .....	156

4.6	查询子句	158
4.6.1	GROUP BY 子句的使用	158
4.6.2	ROLLUP 的使用	159
4.6.3	CUBE 的使用	161
4.6.4	GROUPING 的使用	162
4.6.5	GROUPING SETS 的使用	163
4.6.6	HAVING 子句的使用	164
4.6.7	ORDER BY 子句	165
4.6.8	FOR UPDATE 子句	166
4.6.9	TOP 子句	167
4.6.10	LIMIT 子句	168
4.7	全文检索	169
4.7.1	全文检索的使用	169
4.7.2	自定义全文检索词库	171
4.8	层次查询	175
4.8.1	层次查询子句	175
4.8.2	层次查询相关伪列	176
4.8.3	层次查询相关操作符	176
4.8.4	层次查询相关函数	176
4.8.5	层次查询层内排序	176
4.8.6	层次查询的限制	177
4.9	伪列的使用	182
4.9.1	ROWNUM	182
4.9.2	ROW	183
4.9.3	ROWID	183
4.9.4	UID 及 USER	184
4.9.5	TRXID	184
<b>第 5 章 DM SQL 数据操作和事务处理</b>		<b>185</b>
5.1	数据插入语句	185
5.2	数据修改语句	186
5.3	MERGE INTO 语句	187
5.4	数据删除语句	189
5.5	事务相关语句	190
5.5.1	事务简介	190



5.5.2	事务提交	192
5.5.3	事务回滚	193
5.5.4	事务锁定	195
5.5.5	多版本	198
5.5.6	事务隔离级	199
5.5.7	闪回	202
<b>第 6 章</b>	<b>DM SQL 数据控制</b>	<b>210</b>
6.1	权限管理	210
6.1.1	权限分类	210
6.1.2	授予权限	212
6.1.3	回收权限	216
6.2	角色管理	218
6.2.1	创建角色	219
6.2.2	管理角色权限	220
6.2.3	分配与回收角色	220
6.2.4	启用与停用角色	222
6.2.5	删除角色	223
<b>第 7 章</b>	<b>DM SQL 程序设计基础</b>	<b>224</b>
7.1	DM SQL 程序的特点	224
7.2	DM SQL 程序块结构	225
7.3	DM SQL 程序代码编写规则	225
7.3.1	变量命名规范	225
7.3.2	大小写规则	226
7.3.3	注释	226
7.4	DM SQL 程序变量声明、赋值及操作符	227
7.4.1	变量声明及初始化	227
7.4.2	变量赋值	228
7.4.3	操作符	228
7.5	变量类型	229
7.5.1	标量数据类型	229
7.5.2	大对象数据类型	230
7.5.3	%TYPE 类型	231
7.5.4	%ROWTYPE 类型	231

7.5.5	记录类型	231
7.5.6	数组类型	232
7.5.7	集合类型	235
7.5.8	类类型	241
7.6	DM SQL 程序控制结构	245
7.6.1	条件控制 IF 语句	245
7.6.2	循环语句	247
7.6.3	CASE 语句	250
7.6.4	其他控制语句	251
7.7	异常处理	253
7.7.1	异常处理语法	254
7.7.2	用户自定义异常	255
7.7.3	异常处理函数	257
7.8	游标	258
7.8.1	游标控制	259
7.8.2	游标变量	263
7.8.3	游标更新数据、删除数据	264
<b>第 8 章</b>	<b>DM SQL 程序设计</b>	<b>266</b>
8.1	存储过程	266
8.1.1	存储过程的定义和调用	267
8.1.2	存储过程应用实例	269
8.1.3	存储过程编译	270
8.1.4	存储过程删除	270
8.1.5	C 语法的 DM SQL 程序	271
8.2	存储函数	272
8.2.1	存储函数的定义和调用	272
8.2.2	存储函数编译	274
8.2.3	存储函数删除	274
8.2.4	C 外部函数	274
8.2.5	Java 外部函数	278
8.3	触发器	280
8.3.1	触发器概述	280
8.3.2	触发器创建	281
8.3.3	触发器管理	287

8.4	包	288
8.4.1	创建包语法	288
8.4.2	删除包	289
8.4.3	包应用举例	290
<b>第9章</b>	<b>DM 嵌入式 SQL 程序设计</b>	<b>293</b>
9.1	嵌入式 SQL 程序组成及编译过程	293
9.1.1	嵌入式 SQL 程序组成	293
9.1.2	嵌入式 SQL 程序编译过程	294
9.2	嵌入式 SQL 常用语法	295
9.2.1	SQL 前缀和终结符	295
9.2.2	宿主变量	296
9.2.3	输入和输出变量	296
9.2.4	指示符变量	297
9.2.5	服务器登录与退出	297
9.2.6	单元组查询语句	298
9.3	动态 SQL	299
9.3.1	EXECUTE IMMEDIATE 立即执行语句	300
9.3.2	PREPARE 准备语句	300
9.3.3	EXECUTE 执行语句	301
9.4	嵌入式程序的异常处理	301
9.4.1	嵌入的异常声明语句	301
9.4.2	异常处理语句使用实例	302
附录 A	关键字和保留字	304
附录 B	达梦数据库技术支持	311

# 1

## 第 1 章

# DM SQL 概述

---

结构化查询语言 (Structured Query Language, SQL) 是一种数据库查询和程序设计语言, 用于存储数据以及查询、更新和管理关系数据库系统。SQL 是高级的非过程化编程语言, 其语法接近英语的语句结构, 方便简洁、使用灵活、功能强大, 倍受用户及计算机工业界的欢迎, 被众多计算机公司和数据库厂商所采用, 经过各公司的不断修改、扩充和完善, SQL 语言发展成为关系数据库的标准语言。DM SQL 遵循 SQL 标准, 并对标准 SQL 进行了扩充, 对其他数据库管理系统如 ORACLE、SQL SERVER 具有较强的兼容性。本章主要介绍 DM SQL 的特点、功能和基本语法。

### 1.1 SQL 的发展

1970 年, IBM 研究中心的 E.F.Codd 提出关系模型; 1972 年 IBM 开始研制实验型关系数据库管理系统 System R, 为其开发的查询语言称为 SQUARE (Specifying QUeries As Relation Expression) 语言。1974 年, 来自同一实验室的 Boyce 和 Chamberlain 把 SQUARE 修改为 SEQUEL (Structured English QUery Language) 语言, 1980 年改名为 SQL; Oracle 公司于 1979 年首先提供了商用的 SQL。1986 年, 美国国家标准化协会 (American National Standards Institute, ANSI) 宣布将 SQL 作为关系数据库语言的美国标准, 1987 年 6 月, 国际化标准化组织 (International Standardization Organization, ISO) 将 SQL 采纳为国际标准, 称为 SQL-86; 1989 年, ANSI 用 SQL-89 取代了 SQL-86, 以后通过对 SQL-86 的不断修改和完善, 于 1989 年第二次公布了 SQL 标准 ISO/IEC9075-1989(E), 即 SQL-89, 1992 年又公布了 SQL 标准 ISO/IEC 9075-1992, 即 SQL-92。1999 年发布了 ISO/IEC 9075-1999 《信息技术—数据库语言 SQL》标准, 即 SQL-99。目前, 主流的数据库管理系统均支持 SQL 语言, 如 ORACLE、SQL SERVER、DB2 等。

## 1.2 DM SQL 的特点

DM SQL 遵循结构化查询语言 SQL 标准，并对标准 SQL 进行了扩充。DM SQL 集数据定义、数据查询、数据操纵和数据控制于一体，是一种一体化的、综合的关系数据库语言。DM SQL 语言具有如下特点。

(1) 便捷的多媒体数据管理和处理能力。DM SQL 对多媒体数据类型的管理和处理，与常规数据类型一样，一体化定义、一体化存储、一体化检索、一体化处理，最大限度地提高了 DM 数据库管理系统处理多媒体的能力和速度。

(2) 强大的混合编程能力。DM SQL 语言既是自含式语言，又是嵌入式语言。作为自含式语言，它能独立运行于联机交互方式。作为嵌入式语言，DM SQL 语句能够嵌入到 C、C++、Java 语言程序中，将高级语言（也称主语言）灵活的表达能力、强大的计算功能与 DM SQL 语言的数据处理功能相结合，实现各种复杂的事务处理。而在这两种不同的使用方式中，DM SQL 语言的语法结构是一致的，从而为用户使用提供了极大的方便性和灵活性。

(3) 高度的封装性。DM SQL 语言对 SQL 命令进行了高度封装。用户只需指出“做什么”，而无须指出“怎么做”，对数据存取路径的选择及 DM SQL 语句功能的实现均由系统自动完成，与用户编制的应用程序和具体的机器及关系 DBMS 的实现细节无关，从而降低应用程序开发的复杂性，提高应用程序开发效率，也增强了数据的独立性和应用系统的可移植性。

(4) 高效的数据操作方式。DM SQL 语言采用了集合操作方式。不仅查询结果可以是元组的集合，而且一次插入、删除、修改操作的对象也可以是元组的集合，相对于面向记录的数据语言（一次只能操作一条记录）来说，DM SQL 语言简化了用户的处理，提高了应用程序的运行效率。

(5) 简洁易学的语法结构。DM SQL 语言集数据库的定义、查询、更新、控制、维护、恢复、安全等一系列操作于一体，每一项操作都只需一种操作符表示，格式规范，风格一致，简单方便，易学易用。

## 1.3 DM SQL 的主要功能

DM SQL 语言的功能主要包括数据定义、查询、操纵和控制四个方面，通过各种不同的 SQL 语句来实现。DM SQL 的主要功能包括：

(1) 数据定义功能。数据定义功能用于定义、撤销和修改数据模式。例如，用户、模式、基表、视图、索引、序列、全文索引、存储过程和触发器的定义和删除语句，基表、视图、全文索引的修改语句，对象的更名语句等。

(2) 数据查询功能。数据查询功能用于进行数据的查询。

(3) 数据操纵功能。数据操纵功能用于增加、删除和修改数据。

(4) 数据控制功能。数据控制功能用于对数据访问权限的控制、完整性描述、事务控制等。

## 1.4 DM SQL 数据类型

DM 数据库系统具有 SQL-92 的绝大部分数据类型，以及部分 SQL-99 数据类型。

### 1.4.1 数值型数据类型

数值型数据类型语法及说明如表 1-1 所示。

表 1-1 数值型数据类型语法及说明

语 法	说 明
NUMERIC[(精度[,标度])] DEC[(精度[,标度])] DECIMAL[(精度[,标度])]	NUMERIC 数据类型用于存储零、正负定点数。其中：精度是一个无符号整数，定义了总的数字数，精度范围为 1~38，标度定义了小数点右边的数字位数，定义时如果省略精度，则默认是 16；如果省略标度，则默认是 0。一个数的标度不应大于其精度。所有 NUMERIC 数据类型，如果其值超过精度，达梦数据库返回一个出错信息，如果超过标度，则多余的位截断。NUMERIC(4,1)定义了小数点前面 3 位和小数点后面 1 位，共 4 位的数字，范围为-999.9~999.9
BIT	BIT 类型用于存储整数数据 1、0 或 NULL，只有 0 才转换为假，其他非空、非 0 值都会自动转换为真，可以用来支持 ODBC 和 JDBC 的布尔数据类型。DM 的 BIT 类型与 SQL Server 2000 的 BIT 数据类型相似。功能与 ODBC 和 JDBC 的 BOOL 相同
INTEGER INT PLS_INTEGER	用于存储有符号整数，精度为 10，标度为 0。取值范围为： $-2^{31}$ ~ $+2^{31}-1$
BIGINT	用于存储有符号整数，精度为 19，标度为 0。取值范围为： $-2^{63}$ ~ $2^{63}-1$
BYTE	与 TINYINT 相似，精度为 3，标度为 0
SMALLINT	用于存储有符号整数，精度为 5，标度为 0
REAL	REAL 是带二进制的浮点数，但它不能由用户指定使用的精度，系统指定其二进制精度为 24，十进制精度为 7。取值范围为 $-3.4E+38$ ~ $3.4E+38$
FLOAT[(精度)]	FLOAT 是带二进制精度的浮点数，精度最大不超过 53，如果省略精度，则二进制精度为 53，十进制精度为 15。取值范围为 $-1.7E+308$ ~ $1.7E+308$
DOUBLE[(精度)]	同 FLOAT 相似，精度最大不超过 53
DOUBLE PRECISION	类型指明双精度浮点数，其二进制精度为 53，十进制精度为 15。取值范围为 $-1.7E+308$ ~ $1.7E+308$

### 1.4.2 字符型数据类型

字符型数据类型语法及说明如表 1-2 所示。

表 1-2 字符型数据类型语法及说明

语 法	说 明
CHAR[(长度)]	CHAR 数据类型指定定长字符串。在表中，定义 CHAR 类型的列时，其最大存储长度由数据库页面大小决定，可以指定一个不超过其最大存储长度的正整数作为字符长度，如 CHAR(100)。如果未指定长度，默认为 1。DM 确保存储在该列的所有值都具有这一长度。CHAR 数据类型最大存储长度和页面大小的对应关系见表 1-3。但是，在表达式计算中，该类型的长度上限不受页面大小限制，为 32767
VARCHAR[(长度)] CHARACTER[(长度)] VARCHAR2	VARCHAR 数据类型指定变长字符串，用法类似 CHAR 数据类型，可以指定一个不超过 8188 的正整数作为字符长度，如 VARCHAR(100)。如果未指定长度，默认为 8188。但是，在表中，插入 VARCHAR 数据类型的实际最大存储长度由数据库页面大小决定，具体最大长度算法见表 1-3。但 VARCHAR 类型在表达式计算中的长度上限不受页面大小限制，为 32767。CHAR 同 VARCHAR 的区别在于前者长度不足时，系统自动填充空格，而后者只占用实际的字节空间。另外，实际插入表中的列长度要受到记录长度的约束，每条记录总长度不能大于页面大小的一半。VARCHAR2 类型和 VARCHAR 类型用法相同

表 1-3 CHAR 数据类型最大存储长度和页面大小的对应关系

数据库页面大小	实际最大长度
4K	1900
8K	3900
16K	8000
32K	8188

### 1.4.3 日期型数据类型

日期型数据类型语法及说明如表 1-4 所示。

表 1-4 日期型数据类型语法及说明

语 法	说 明	举 例
DATE	日期类型，包括年、月、日信息，定义了'0001-01-01'和'9999-12-31'之间任何一个有效的格里高利日期。 DM 支持 SQL-92 标准或 SQL Server 的 DATE 字值。例如，DATE'1999-10-01'、'1999/10/01'或'1999.10.01'都是有效的 DATE 值，且彼此等价。年月日中第一个非 0 数值前的 0 亦可省略，如'0001-01-01'等价于'1-1-1'	创建数据表 t1，其中字段 c1 为 DATE 类型，然后插入记录： CREATE TABLE t1(c1 DATE); INSERT INTO t1 VALUES (DATE '2002-12-12');
TIME	包括时、分、秒信息，定义了一个在'00:00:00.000000'和'23:59:59.999999'之间的有效时间。TIME 类型的小数秒精度规定了秒字段中小数点后面的位数，取值范围为 0~6，如果未定义，默认精度为 0。 DM 支持 SQL-92 标准或 SQL Server 的 TIME 字值，如 TIME'09:10:21'、'09:10:21'或'9:10:21'都是有效的 TIME 值，且彼此等价	创建数据表 t2，其中字段 c1 为 TIME 类型，然后插入记录： CREATE TABLE t2(c1 TIME); INSERT INTO t2 VALUES (TIME '09:10:21');

(续表)

语 法	说 明	举 例
TIMESTAMP	时间戳型, 包括年、月、日、时、分、秒信息, 定义了一个在 '0001-01-01 00:00:00.000000' 和 '9999-12-31 23:59:59.999999' 之间的有效格里高利日期时间。TIMESTAMP 类型的小数秒精度规定了秒字段中小数点后面的位数, 取值范围为 0~6, 如果未定义, 默认精度为 6。DM 支持 SQL-92 标准或 SQL Server 的 TIMESTAMP 字值, 如 TIMESTAMP'2002-12-12 09:10:21' 或 '2002-12-12 9:10:21' 或 '2002/12/12 09:10:21' 或 '2002.12.12 09:10:21' 都是有效的 TIMESTAMP 值, 且彼此等价	创建数据表 t3, 其中字段 c1 为 TIMESTAMP 类型, 然后插入记录: CREATE TABLE t3(c1 TIMESTAMP); INSERT INTO t3 VALUES (TIMESTAMP'1999-07-13 10:11:22');
TIME [(小数秒精度)] WITH TIME ZONE	描述一个带时区的 TIME 值, 其定义是在 TIME 类型的后面加上时区信息	创建数据表 t4, 其中字段 c1 为 TIME WITH TIME ZONE 类型, 然后插入记录: CREATE TABLE t4(c1 TIME (2) WITH TIME ZONE); INSERT INTO t4 VALUES (TIME '09:10:21 +8:00');
TIMESTAMP [(小数秒精度)] WITH TIME ZONE	描述一个带时区的 TIMESTAMP 值, 其定义是在 TIMESTAMP 类型的后面加上时区信息。时区部分的实质是 INTERVAL HOUR TO MINUTE 类型, 其值应在 -12:59~+13:00 之间。一个合适的例子是: '2009-10-11 19:03:05.0000 -02:10'。TIMESTAMP 类型的小数秒精度规定了秒字段中小数点后面的位数, 取值范围为 0~6, 如果未定义, 默认精度为 6。DM 支持 SQL-92 标准或 SQL Server 的 TIMESTAMP 字值, 如 TIMESTAMP'2002-12-12 09:10:21 +8:00' 或 '2002-12-12 9:10:21 +8:00' 或 '2002/12/12 09:10:21 +8:00' 或 '2002.12.12 09:10:21 +8:00' 都是有效的 TIMESTAMP WITH TIME ZONE 值, 且彼此等价。语法中, TIMESTAMP 也可以写为 DATETIME	创建数据表 t5, 其中字段 c1 为 TIMESTAMP WITH TIME ZONE 类型, 然后插入记录: CREATE TABLE t5(c1 TIMESTAMP (2) WITH TIME ZONE); INSERT INTO t5 VALUES (TIMESTAMP '2002-12-12 09:10:21 +8:00');
TIMESTAMP [(小数秒精度)] WITH LOCAL TIME ZONE	描述一个本地时区的 TIMESTAMP 值, 能够将标准时区类型 TIMESTAMP WITH TIME ZONE 转化为本地时区类型, 如果插入的值没有指定时区, 则默认为本地时区。时区部分的实质是 INTERVAL HOUR TO MINUTE 类型, 其值应为 -12:59~+13:00 之间。一个合适的例子是: '2009-10-11 19:03:05.0000 -02:10'。TIMESTAMP 类型的小数秒精度规定了秒字段中小数点后面的位数, 取值范围为 0~6, 如果未定义, 默认精度为 6。 DM 支持 SQL-92 标准或 SQL Server 的 TIMESTAMP 值, 如 TIMESTAMP'2002-12-12 09:10:21 +8:00' 或 '2002-12-12 9:10:21 +8:00' 或 '2002/12/12 09:10:21 +8:00' 或 '2002.12.12 09:10:21 +8:00' 都是有效的 TIMESTAMP WITH LOCAL TIME ZONE 值, 且彼此等价	创建数据表 t6, 其中字段 c1 为 TIMESTAMP WITH LOCAL TIME ZONE 类型, 然后插入记录: CREATE TABLE t6 (c1 TIMESTAMP (3) WITH LOCAL TIME ZONE, c2 TIMESTAMP(3) WITH LOCAL TIME ZONE); INSERT INTO t6 VALUES (TIMESTAMP '2002-12-12 09:10:21 +8:00', TIMESTAMP '2002-12-12 09:10:21');

#### 1.4.4 日期时间间隔数据类型

DM 支持两类 13 种时间间隔类型: 年-月间隔类和日-时间隔类, 它们通过时间间隔限定符区分, 前者结合了日期字段年和月, 后者结合了时间字段日、时、分、秒。由时间间



隔数据类型所描述的值总是有符号的，如表 1-5 所示。

需要说明的是，使用时间间隔数据类型时，如果使用了其引导精度的默认精度，要注意保持精度匹配，否则会出现错误。如果不指定精度，那么将使用默认精度 6。

表 1-5 日期时间间隔数据类型语法及说明

数据类型	语 法	说 明	举 例
年-月间隔	INTERVAL YEAR [(引导精度)] TO MONTH	描述一个若干年若干月的间隔，引导精度规定了年的取值范围。引导精度取值范围为 0~9，如果未定义，默认精度为 2。月的取值范围为 0~11。例如：INTERVAL YEAR(4) TO MONTH，其中 YEAR(4)表示年的精度为 4，表示范围为负 9999 年 12 月到正 9999 年 12 月，如时间间隔值 INTERVAL'1-5' YEAR TO MONTH 表示 1 年 5 个月	创建数据表 t7，其中字段 c1 为 INTERVAL YEAR(3) TO MONTH 类型，然后插入记录： CREATE TABLE t7(c1 INTERVAL YEAR(3) TO MONTH); INSERT INTO t7 VALUES (INTERVAL '1-5' YEAR TO MONTH); INSERT INTO t7 VALUES (INTERVAL '15' MONTH);
	INTERVAL YEAR [(引导精度)]	描述一个若干年的间隔，引导精度规定了年的取值范围。引导精度取值范围为 1~9，如果未定义，默认精度为 2。例如，INTERVAL YEAR(4)，其中 YEAR(4)表示年的精度为 4，表示范围为负 9999 年到正 9999 年，如时间间隔值 INTERVAL'0025' YEAR 表示 25 年	创建数据表 t8，其中字段 c1 为 INTERVAL YEAR(2) 类型，然后插入记录： CREATE TABLE t8(c1 INTERVAL YEAR(2)); INSERT INTO t8 VALUES(INTERVAL '2' YEAR );
	INTERVAL MONTH [(引导精度)]	描述一个若干月的间隔，引导精度规定了月的取值范围。引导精度取值范围为 0~9，如果未定义，默认精度为 2。例如，INTERVALMONTH(3)，其中 MONTH(3)表示月的精度为 3，表示范围为负 999 月到正 999 月，如时间间隔值 INTERVAL'140' MONTH 表示 140 个月	创建数据表 t9，其中字段 c1 为 INTERVAL MONTH(3)类型，然后插入记录： CREATE TABLE t9(c1 INTERVAL MONTH(3)); INSERT INTO t9 VALUES(INTERVAL '120' MONTH );
日-时间隔	INTERVAL DAY [(引导精度)]	描述一个若干日的间隔，引导精度规定了日的取值范围。引导精度取值范围为 0~9，如果未定义，默认精度为 2。例如，INTERVALDAY(3)，其中 DAY(3)表示日的精度为 3，表示范围为负 999 日到正 999 日，如时间间隔值 INTERVAL'210' DAY 表示 210 天	创建数据表 t10，其中字段 c1 为 INTERVAL DAY(2)类型，然后插入记录： CREATE TABLE t10(c1 INTERVAL DAY(3)); INSERT INTO t10 VALUES (INTERVAL '210' DAY);
	INTERVAL DAY[(引 导精度)] TO HOUR	描述一个若干日若干小时的间隔，引导精度规定了日的取值范围。引导精度取值范围为 0~9，如果未定义，默认精度为 2。而时的取值范围为 0~23。例如，INTERVAL DAY(1) TO HOUR，其中 DAY(1)表示日的精度为 1，表示范围为负 9 日 23 小时到正 9 日 23 小时，如时间间隔值 INTERVAL'7 420' DAY TO HOUR 表示 7 天 420 小时	创建数据表 t11，其中字段 c1 为 INTERVAL DAY(1) TO HOUR 类型，然后插入记录： CREATE TABLE t11(c1 INTERVAL DAY(1) TO HOUR); INSERT INTO t11 VALUES(INTERVAL '7 20' DAY TO HOUR );