

TURING

图灵程序  
设计丛书

# SQL基础教程

第2版

[日] MICK / 著 孙淼 罗勇 / 译



人民邮电出版社  
北 京

## 图书在版编目 ( CIP ) 数据

SQL 基础教程 / (日) MICK 著; 孙淼, 罗勇译. -- 2  
版. -- 北京: 人民邮电出版社, 2017.6  
(图灵程序设计丛书)  
ISBN 978-7-115-45502-4

I. ①S… II. ①M… ②孙… ③罗… III. ①关系数据  
库系统—教材 IV. ①TP311.138

中国版本图书馆 CIP 数据核字 (2017) 第 087921 号

SQL 第 2 版 ゼロからはじめるデータベース操作  
(SQL dai2han Zero kara Hajimeru Database Sousa : 4445-0)  
Copyright © 2016 by Mick.

Original Japanese edition published by SHOEISHA Co., Ltd.

Simplified Chinese Character translation rights arranged with SHOEISHA Co., Ltd.  
through CREEK & RIVER Co., Ltd. and CREEK & RIVER SHANGHAI Co., Ltd.

Simplified Chinese Character translation copyright © 2017 by Posts & Telecom Press.

本书中文简体字版由 SHOEISHA Co., Ltd. 授权人民邮电出版社独家出版。未经出版者书面许可,  
不得以任何方式复制或抄袭本书内容。

版权所有, 侵权必究。

## 内 容 提 要

本书是畅销书《SQL 基础教程》的第 2 版, 介绍了关系数据库以及用来操作关系数据库的 SQL 语言的使用方法。书中通过丰富的图示、大量示例程序和详实的操作步骤说明, 让读者循序渐进地掌握 SQL 的基础知识和使用技巧, 切实提高编程能力。每章结尾设置有练习题, 帮助读者检验对各章内容的理解程度。另外, 本书还将重要知识点总结为“法则”, 方便读者随时查阅。第 2 版除了将示例程序更新为对应最新的 DB 的 SQL 之外, 还新增了一章, 介绍如何从应用程序执行 SQL。

本书适合数据库和 SQL 语言的初学者阅读, 也可作为大中专院校的教材及企业新人培训用书。

---

◆ 著 [日] MICK

译 孙 淼 罗 勇

责任编辑 杜晓静

执行编辑 刘香娣

责任印制 彭志环

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京 印刷

◆ 开本: 800×1000 1/16

印张: 21

字数: 455 千字 2017 年 6 月第 2 版

印数: 16 901-20 900 册 2017 年 6 月北京第 1 次印刷

著作权合同登记号 图字: 01-2016-6685 号

---

定价: 79.00 元

读者服务热线: (010)51095186 转 600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

# 前言

本书面向完全没有编程和系统开发经验的初学者，介绍了关系数据库以及用来操作关系数据库的 SQL 语言的使用方法。各个章节结合具体示例进行解说，并在每章的结尾安排了习题，用来检验读者对该章内容的理解程度。大家可以从第 1 章开始，亲自验证示例程序，循序渐进地掌握 SQL 的基础知识和技巧。另外，本书还将重要知识点总结为法则，方便读者在学习完本书之后随时查阅。

近年来，和其他系统领域一样，数据库领域也实现了飞速发展，应用范围不断扩大，不但出现了具有新功能的数据库，而且操作的数据量也大幅增长。

本书将要介绍的关系数据库是时下最流行的数据库，也是理解其他数据库的基础。在系统领域，通常所讲的数据库指的就是关系数据库，其重要性可见一斑。

估计很多读者今后都会慢慢积累各个领域、各种规模的系统开发经验（或者可能已经开始从事开发方面的工作了），到那时，所有的系统必定都需要使用数据库。它们使用的数据库，即便不是关系数据库，也一定是以关系数据库为基础的数据库。从这个意义上看，如果掌握了关系数据库和 SQL，就能成为任何系统开发都需要的数据库专家了。

现在距离本书初版问世已经 6 年了，在这 6 年间，数据库发挥了越来越重要的作用。以前就有专家使用数据库进行统计分析，后来数据库也开始逐渐被应用到大规模数据的处理上，并引发了商业领域的变革。象征着这一变化的“大数据”“数据科学”等用语，已经突破了系统的领域，蔓延到了整个社会之中。甚至有观点认为，统计分析将和人工智能并列成为决定社会未来走向的重要因素。

一方面，数据库的世界中也进行着技术的革新。如今，以 KVS 为代表的非关系数据库的使用已经不再稀奇。同时，为了追求更高的大规模数据处理的性能，内存数据库和面向列数据库的技术也取得了长足的进步，并逐渐投入到实际应用当中。

另一方面，关系数据库依然是当今的主流数据库，这一点没有变。从这个意义上来说，学习关系数据库和操作关系数据库的语言 SQL 语句，仍然是探究数据库世界的第一步，这一点也没有变，但这并不是说关系数据库和 SQL 语句一直在止步不前。大多数 DBMS 都支持窗口函数和 GROUPING 运算符（详见第 8 章），高效处理大规模数据的功能也更加完善。掌握了 SQL 语句，就可以自由自在地操作数据，构筑高效的系统。

本书与时俱进地进行了版本升级。不但根据具有代表性的 DBMS 的新版本对 SQL 语言的支持情况更新了描述，还新增了第 9 章，介绍了通过应用程序来使用数据库的方法。

本书旨在把数据库领域的精彩展示给大家，衷心希望本书能为大家的进步提供一些帮助。

MICK

---

## 关于本书

本书是编程学习系列的 SQL 和关系数据库篇。该系列注重对初学者编程能力的培养，本书秉承了这一宗旨。本书不仅可以用于自学，也可以作为大学、专科学校和企业新人的培训用书。书中提供了大量的示例程序和详实的操作步骤说明，大家可以亲自动手解决具体的问题，切实提高自身的编程能力。

另外，在各章的结尾处还安排了习题来帮助大家复习该章的知识要点，习题的答案和讲解收录在附录中。

---

## 读者对象

- 不了解数据库和 SQL 知识的人
- 虽然自学了一些 SQL 知识，但仍希望进行系统学习的人
- 需要使用数据库，但不知道从何入手的人
- 在大学、专科学校和企业的教育部门等从事数据库和 SQL 教学的人
- 希望了解信息处理考试中 SQL 部分应试策略的人

---

## 学习本书前的预备知识

- 了解 Windows 的基本操作方法
- 能够使用 Windows 的资源管理器创建文件夹并复制文件
- 能够使用 Windows 的记事本（或者其他文本编辑器）创建文本文件

---

## 本书涉及的关系数据库

本书中使用的 SQL 语句全部都在下列关系数据库管理系统（RDBMS）中进行了验证。

- Oracle Database 12cR1
- SQL Server 2014
- DB2 10.5
- PostgreSQL 9.5.3
- MySQL 5.7

在这 5 种 RDBMS 之间存在差异的 SQL 语句，或者只能在某种特定的 RDBMS 中使用的 SQL 语句，本书都用下列图标进行标识，来提示执行 SQL 语句所使用的 RDBMS。



反之，在所有 RDBMS 中都能正常执行的 SQL 语句则不用图标标识。

---

## 本书的学习安排

首先，在第 1 章前半部分学习关系数据库和 SQL 的基础知识，然后结合具体的 SQL 示例程序进行循序渐进的学习。

在 SQL 的学习中，最重要的就是以下两点：

- 亲自编写 SQL 语句
- 通过执行 SQL 语句来学习和理解数据库操作

要提高学习效率，需尽量亲自执行并验证本书中的示例程序，逐步深入学习。

为了便于初学者操作，本书使用 PostgreSQL 作为 SQL 语句的学习环境。在开始学习之前，读者需要先在自已的电脑上安装 PostgreSQL，准备好 SQL 语句的执行环境。关于

PostgreSQL 的安装方法、SQL 语句的执行方法等详细内容，我们会在第 0 章介绍。

如果你已经安装了上述“本书涉及的关系数据库”中的数据库，也可以直接使用。

另外，如无特殊说明，本书中出现的 SQL 语句的执行结果，都是在 PostgreSQL 9.5 中执行的结果。

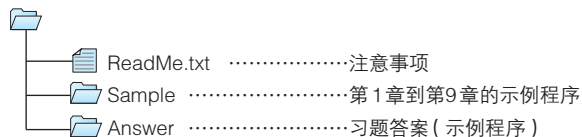
---

## 关于程序下载

本书中的示例程序都可以从下面的网站下载。

<http://www.ituring.com.cn/book/1880>

示例程序为压缩的 Zip 文件形式，解压后的文件结构如下所示。

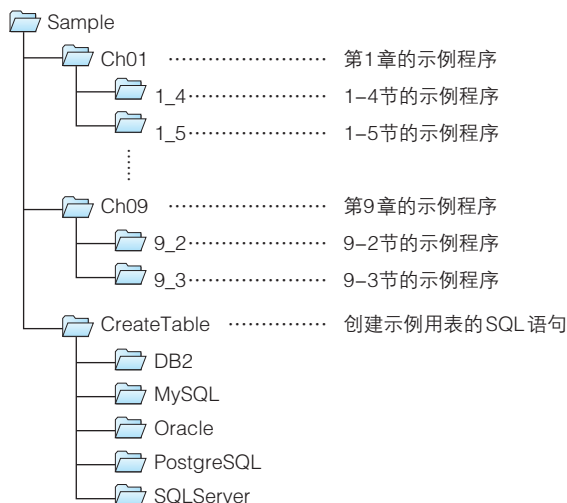


### ReadMe.txt 文件

介绍了示例程序的内容和注意事项，使用前请务必阅读该文件。

### Sample 文件夹

本书中所使用的示例程序分别保存在以章节为单位的文件夹中。在 Sample\CreateTable 文件夹中，按照 RDBMS 的不同，分别保存了用来创建示例用表的 SQL 语句。

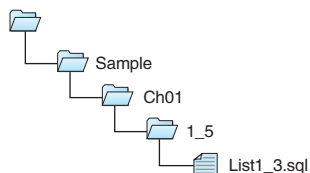


## Answer文件夹

各章末习题的答案（示例程序），分别保存在以章为单位的目录中。

## 关于示例程序

示例程序的文件名与书中的代码清单编号相对应。例如，1-5 节的代码清单 1-3 的示例程序，保存的位置和文件名如下所示。



另外，像如下代码清单这样，在不同的 RDBMS 中存在差异的 SQL 语句，会在其文件名的末尾加上 RDBMS 的名称。

代码清单 1-4 添加一列可以存储 100 位可变长度字符串的 **product\_name\_pinyin** 列

DB2 PostgreSQL MySQL

```
ALTER TABLE Product ADD COLUMN product_name_pinyin VARCHAR(100);
```

**Oracle**

```
ALTER TABLE Product ADD (product_name_pinyin VARCHAR(100));
```

**SQL Server**

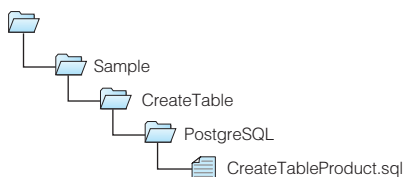
```
ALTER TABLE Product ADD product_name_pinyin VARCHAR(100);
```

这种情况下，示例程序的文件名如下所示。

- List1\_4\_DB2\_PostgreSQL\_MySQL.sql
- List1\_4\_Oracle.sql
- List1\_4\_SQL Server.sql

### 创建示例用表的 SQL 语句

用于创建示例用表的 SQL 文件保存在 Sample\CreateTable 文件夹中，文件名为“CreateTable 表名 .sql”。例如，PostgreSQL 用到的表 Product 保存在下述目录中。



保存在 Sample 文件夹中的示例程序文件，可以使用 Windows 的记事本（或者其他文本编辑器）打开。

---

## 声明

翔泳社

本书中的示例程序已经经过编辑部确认，在正常使用时不会出现任何问题。对于执行程序所造成的任何损失，本书作者、软件开发人员和翔泳社概不承担相关责任。

Sample 文件夹中所收录的文件的著作权归本书作者所有。读者可以出于个人目的，根据需要自行使用和修改其中的程序。

对于个别环境相关的问题，以及在超出本书内容范围的环境中进行设置时的问题，本社概不负责解答。



# 目 录

第0章 绪论——搭建 SQL 的学习环境	1
0-1 PostgreSQL 的安装和连接设置	3
安装步骤	3
修改设置文件	7
0-2 通过 PostgreSQL 执行 SQL 语句	9
连接 PostgreSQL ( 登录 )	9
执行 SQL 语句	10
创建学习用的数据库	11
连接学习用的数据库 ( 登录 )	12
第1章 数据库和 SQL	13
1-1 数据库是什么	15
我们身边的数据库	15
为什么 DBMS 那么重要	16
DBMS 的种类	18
1-2 数据库的结构	20
RDBMS 的常见系统结构	20
表的结构	22
1-3 SQL 概要	25
标准 SQL	25
SQL 语句及其种类	26
SQL 的基本书写规则	27
1-4 表的创建	30
表的内容的创建	30
数据库的创建 ( CREATE DATABASE 语句 )	31

表的创建 ( CREATE TABLE 语句 )	31
命名规则	33
数据类型的指定	34
约束的设置	35
1-5 表的删除和更新	37
表的删除 ( DROP TABLE 语句 )	37
表定义的更新 ( ALTER TABLE 语句 )	38
向 Product 表中插入数据	39
练习题	42

第 2 章 查询基础 43

2-1 SELECT 语句基础	45
列的查询	45
查询出表中所有的列	47
为列设定别名	48
常数的查询	50
从结果中删除重复行	50
根据 WHERE 语句来选择记录	53
注释的书写方法	55
2-2 算术运算符和比较运算符	57
算术运算符	57
需要注意 NULL	58
比较运算符	60
对字符串使用不等号时的注意事项	62
不能对 NULL 使用比较运算符	65
2-3 逻辑运算符	68
NOT 运算符	68

AND 运算符和 OR 运算符	70
通过括号强化处理	72
逻辑运算符和真值	74
含有 NULL 时的真值	76
练习题	78

### 第3章 聚合与排序 79

3-1 对表进行聚合查询	81
聚合函数	81
计算表中数据的行数	82
计算 NULL 之外的数据的行数	83
计算合计值	84
计算平均值	86
计算最大值和最小值	87
使用聚合函数删除重复值 ( 关键字 DISTINCT )	89
3-2 对表进行分组	91
GROUP BY 子句	91
聚合键中包含 NULL 的情况	93
使用 WHERE 子句时 GROUP BY 的执行结果	94
与聚合函数和 GROUP BY 子句有关的常见错误	96
3-3 为聚合结果指定条件	101
HAVING 子句	101
HAVING 子句的构成要素	104
相对于 HAVING 子句, 更适合写在 WHERE 子句中的条件	105
3-4 对查询结果进行排序	108
ORDER BY 子句	108
指定升序或降序	110

指定多个排序键.....	111
NULL 的顺序.....	111
在排序键中使用显示用的别名.....	112
ORDER BY 子句中可以使用的列.....	114
不要使用列编号.....	114
练习题.....	116

第 4 章 数据更新 117

4-1 数据的插入 ( INSERT 语句的使用方法 ).....	119
什么是 INSERT.....	119
INSERT 语句的基本语法.....	120
列清单的省略.....	123
插入 NULL.....	123
插入默认值.....	124
从其他表中复制数据.....	126
4-2 数据的删除 ( DELETE 语句的使用方法 ).....	129
DROP TABLE 语句和 DELETE 语句.....	129
DELETE 语句的基本语法.....	129
指定删除对象的 DELETE 语句 ( 搜索型 DELETE ).....	130
4-3 数据的更新 ( UPDATE 语句的使用方法 ).....	133
UPDATE 语句的基本语法.....	133
指定条件的 UPDATE 语句 ( 搜索型 UPDATE ).....	134
使用 NULL 进行更新.....	135
多列更新.....	136
4-4 事务.....	138
什么是事务.....	138
创建事务.....	139

ACID 特性	144
练习题	145
<b>第 5 章 复杂查询</b>	<b>147</b>
5-1 视图	149
视图和表	149
创建视图的方法	151
视图的限制①——定义视图时不能使用 ORDER BY 子句	154
视图的限制②——对视图进行更新	155
删除视图	159
5-2 子查询	160
子查询和视图	160
子查询的名称	163
标量子查询	163
标量子查询的书写位置	166
使用标量子查询时的注意事项	167
5-3 关联子查询	168
普通的子查询和关联子查询的区别	168
关联子查询也是用来对集合进行切分的	171
结合条件一定要写在子查询中	172
练习题	173
<b>第 6 章 函数、谓词、CASE 表达式</b>	<b>175</b>
6-1 各种各样的函数	177
函数的种类	177
算术函数	178
字符串函数	182

日期函数.....	190
转换函数.....	194
6-2 谓词.....	198
什么是谓词.....	198
LIKE 谓词——字符串的部分一致查询.....	198
BETWEEN 谓词——范围查询.....	202
IS NULL、IS NOT NULL——判断是否为NULL.....	203
IN 谓词——OR的简用法.....	204
使用子查询作为 IN 谓词的参数.....	205
EXIST 谓词.....	210
6-3 CASE 表达式.....	214
什么是CASE表达式.....	214
CASE 表达式的语法.....	214
CASE 表达式的使用方法.....	215
练习题.....	221

第7章 集合运算 223

7-1 表的加减法.....	225
什么是集合运算.....	225
表的加法——UNION.....	225
集合运算的注意事项.....	228
包含重复行的集合运算——ALL 选项.....	229
选取表中公共部分——INTERSECT.....	230
记录的减法——EXCEPT.....	231
7-2 联结(以列为单位对表进行联结).....	234
什么是联结.....	234
内联结——INNER JOIN.....	235

外联结——OUTER JOIN·····	240
3张以上的表的联结·····	243
交叉联结——CROSS JOIN·····	246
联结的特定语法和过时语法·····	249
练习题·····	254

## 第8章 SQL 高级处理 255

8-1 窗口函数·····	257
什么是窗口函数·····	257
窗口函数的语法·····	258
语法的基本使用方法——使用RANK函数·····	258
无需指定PARTITION BY·····	261
专用窗口函数的种类·····	262
窗口函数的适用范围·····	263
作为窗口函数使用的聚合函数·····	264
计算移动平均·····	266
两个ORDER BY·····	269
8-2 GROUPING运算符·····	271
同时得到合计行·····	271
ROLLUP——同时得出合计和小计·····	272
GROUPING函数——让NULL更容易分辨·····	277
CUBE——用数据来搭积木·····	279
GROUPING SETS——取得期望的积木·····	281
练习题·····	282

## 第9章 通过应用程序连接数据库 283

9-1 数据库世界和应用程序世界的连接·····	285
数据库和应用程序之间的关系·····	285



驱动——两个世界之间的桥梁.....	286
驱动的种类.....	287
9-2 Java 基础知识.....	289
第一个程序 Hello, World.....	289
编译和程序执行.....	291
常见错误.....	294
9-3 通过 Java 连接 PostgreSQL.....	298
执行 SQL 语句的 Java 程序.....	298
Java 是如何从数据库中获取数据的呢.....	299
执行连接数据库的程序.....	301
选取表中的数据.....	302
更新表中的数据.....	305
小结.....	307
练习题.....	307

附录 练习题答案	308
----------	-----



## 第 2 章 查询基础

SELECT 语句基础  
算术运算符和比较运算符  
逻辑运算符



SQL

## 本章重点

---

本章将会和大家一起学习查询前一章创建的 Product 表中数据的 SQL 语句。这里使用的 SELECT 语句是 SQL 最基本也是最重要的语句。请大家在实际运行书中的 SELECT 语句时，亲身体验一下其书写方法和执行结果。

执行查询操作时可以指定想要查询数据的条件（查询条件）。查询时可以指定一个或多个查询条件，例如“某一列等于这个值”“某一列计算之后的值大于这个值”等。

### 2-1 SELECT 语句基础

- 列的查询
- 查询出表中所有的列
- 为列设定别名
- 常数的查询
- 从结果中删除重复行
- 根据 WHERE 语句来选择记录
- 注释的书写方法

### 2-2 算术运算符和比较运算符

- 算术运算符
- 需要注意 NULL
- 比较运算符
- 对字符串使用不等号时的注意事项
- 不能对 NULL 使用比较运算符

### 2-3 逻辑运算符

- NOT 运算符
- AND 运算符和 OR 运算符
- 使用括号强化处理
- 逻辑运算符和真值
- 含有 NULL 时的真值

## 2-1

## 第2章 查询基础

## SELECT 语句基础

## 学习重点

- 使用 SELECT 语句从表中选取数据。
- 为列设定显示用的别名。
- SELECT 语句中可以使用常数或者表达式。
- 通过指定 DISTINCT 可以删除重复的行。
- SQL 语句中可以使用注释。
- 可以通过 WHERE 语句从表中选取出符合查询条件的数据。

## KEYWORD

- SELECT 语句
- 匹配查询
- 查询

## 列的查询

从表中选取数据时需要使用 **SELECT 语句**，也就是只从表中选出 (SELECT) 必要数据的意思。通过 SELECT 语句查询并选取出必要数据的过程称为**匹配查询**或**查询** (query)。

SELECT 语句是 SQL 语句中使用最多的最基本的 SQL 语句。掌握了 SELECT 语句，距离掌握 SQL 语句就不远了。

SELECT 语句的基本语法如下所示。

## 语法2-1 基本的 SELECT 语句

```
SELECT <列名>, .....  
FROM <表名>;
```

## KEYWORD

- 子句

该 SELECT 语句包含了 SELECT 和 FROM 两个**子句** (clause)。子句是 SQL 语句的组成要素，是以 SELECT 或者 FROM 等作为起始的短语。

SELECT 子句中列举了希望从表中查询出的列的名称，而 FROM 子句则指定了选取出数据的表的名称。

接下来，我们尝试从第1章创建出的 Product（商品）表中，查询出图 2-1 所示的 product\_id（商品编号）列、product\_name（商品名称）列和 purchase\_price（进货单价）列。

图2-1 查询出 Product 表中的列

product_id (商品编号)	product_name (商品名称)	product_type (商品种类)	sale_price (销售单价)	purchase_price (进货单价)	regist_date (登记日期)
0001	T恤衫	衣服	1000	500	2009-09-20
0002	打孔器	办公用品	500	320	2009-09-11
0003	运动T恤	衣服	4000	2800	
0004	菜刀	厨房用具	3000	2800	2009-09-20
0005	高压锅	厨房用具	6800	5000	2009-01-15
0006	叉子	厨房用具	500		2009-09-20
0007	擦菜板	厨房用具	880	790	2008-04-28
0008	圆珠笔	办公用品	100		2009-11-11

输出这3列

对应的 SELECT 语句请参见代码清单 2-1，该语句正常执行的结果如执行结果所示<sup>❶</sup>。

注❶

结果的显示方式根据RDBMS的客户端的不同略有不同（数据的内容都是相同的）。如无特殊说明，本书中显示的都是PostgreSQL 9.5的执行结果。

代码清单2-1 从 Product 表中输出3列

```
SELECT product_id, product_name, purchase_price
FROM Product;
```

执行结果

product_id	product_name	purchase_price
0001	T恤衫	500
0002	打孔器	320
0003	运动T恤	2800
0004	菜刀	2800
0005	高压锅	5000
0006	叉子	
0007	擦菜板	790
0008	圆珠笔	

SELECT 语句第一行的 SELECT product\_id, product\_name, purchase\_price 就是 SELECT 子句。查询出的列的顺序可以任意指

**注②**

行的顺序也可能存在与上述执行结果不同的情况。如果用户不设定 **SELECT** 语句执行结果中行的顺序, 就可能会发生上述情况。行的排序方法将在第3章进行学习。

**KEYWORD**

● 星号 (\*)

定。查询多列时, 需要使用逗号进行分隔。查询结果中列的顺序和 **SELECT** 子句中的顺序相同<sup>②</sup>。

## 查询出表中所有的列

想要查询出全部列时, 可以使用代表所有列的星号 (\*)。

### 语法2-2 查询全部的列

```
SELECT *  
FROM <表名>;
```

例如, 查询 **Product** 表中全部列的语句如代码清单 2-2 所示。

### 代码清单2-2 输出 **Product** 表中全部的列

```
SELECT *  
FROM Product;
```

得到的结果和代码清单 2-3 中的 **SELECT** 语句的结果相同。

### 代码清单2-3 与代码清单2-2具有相同含义的 **SELECT** 语句

```
SELECT product_id, product_name, product_type, sale_price,  
       purchase_price, regist_date  
FROM Product;
```

执行结果如下所示。

### 执行结果

product_id	product_name	product_type	sale_price	purchase_price	regist_date
0001	T恤衫	衣服	1000	500	2009-09-20
0002	打孔器	办公用品	500	320	2009-09-11
0003	运动T恤	衣服	4000	2800	
0004	菜刀	厨房用具	3000	2800	2009-09-20
0005	高压锅	厨房用具	6800	5000	2009-01-15
0006	叉子	厨房用具	500		2009-09-20
0007	擦菜板	厨房用具	880	790	2008-04-28
0008	圆珠笔	办公用品	100		2009-11-11

**法则 2-1**

星号（\*）代表全部列的意思。

但是，如果使用星号的话，就无法设定列的显示顺序了。这时就会按照 CREATE TABLE 语句的定义对列进行排序。

**专 栏****随意使用换行符**

SQL 语句使用换行符或者半角空格来分隔单词，在任何位置进行分隔都可以，即使像下面这样通篇都是换行符也不会影响 SELECT 语句的执行。但是这样可能会由于看不清楚而出错。原则上希望大家能够以子句为单位进行换行（子句过长时，为方便起见可以换行）。

```
SELECT
*
FROM
Product
;
```

另外，像下面这样插入空行（无任何字符的行）会造成执行错误，请特别注意。

```
SELECT *

FROM Product;
```

## 为列设定别名

**KEYWORD**

- AS 关键字
- 别名

SQL 语句可以使用 **AS 关键字** 为列设定 **别名**。请参见代码清单 2-4。

代码清单 2-4 为列设定别名

```
SELECT product_id      AS id,
       product_name    AS name,
       purchase_price  AS price
FROM Product;
```

执行结果

id	name	price
0001	T恤衫	500
0002	打孔器	320
0003	运动T恤	2800
0004	菜刀	2800
0005	高压锅	5000
0006	叉子	
0007	擦菜板	790
0008	圆珠笔	

KEYWORD

● 双引号 ( " )

注 ①

使用双引号可以设定包含空格 ( 空白 ) 的别名。但是如果忘记使用双引号就可能出错, 因此并不推荐。大家可以像product\_list这样使用下划线 ( \_ ) 来代替空白。

别名可以使用中文, 使用中文时需要用双引号 ( " ) 括起来<sup>①</sup>。请注意不是单引号 ( ' )。设定中文别名的 SELECT 语句请参见代码清单 2-5。


代码清单 2-5 设定中文别名

```
SELECT product_id      AS "商品编号",
       product_name    AS "商品名称",
       purchase_price AS "进货单价"
FROM Product;
```

执行结果

商品编号	商品名称	进货单价
0001	T恤衫	500
0002	打孔器	320
0003	运动T恤	2800
0004	菜刀	2800
0005	高压锅	5000
0006	叉子	
0007	擦菜板	790
0008	圆珠笔	

通过执行结果来理解就更加容易了。像这样使用别名可以让 SELECT 语句的执行结果更加容易理解和操作。

 法则 2-2

设定汉语别名时需要使用双引号 ( " ) 括起来。

KEYWORD

- 字符串常数
- 数字常数
- 日期常数

注 ①

在 SQL 语句中使用字符串或者日期常数时，必须使用单引号(')将其括起来。

常数的查询

SELECT 子句中不仅可以书写列名，还可以书写常数。代码清单 2-6 中的 SELECT 子句中的第一列 '商品' 是字符串常数，第 2 列 38 是数字常数，第 3 列 '2009-02-24' 是日期常数，它们将与 product\_id 列和 product\_name 列一起被查询出来。<sup>①</sup>

代码清单2-6 查询常数

```
SELECT '商品' AS string, 38 AS number, '2009-02-24' AS date,
       product_id, product_name
FROM Product;
```

执行结果

string	number	date	product_id	product_name
商品	38	2009-02-24	0001	T恤衫
商品	38	2009-02-24	0002	打孔器
商品	38	2009-02-24	0003	运动T恤
商品	38	2009-02-24	0004	菜刀
商品	38	2009-02-24	0005	高压锅
商品	38	2009-02-24	0006	叉子
商品	38	2009-02-24	0007	擦菜板
商品	38	2009-02-24	0008	圆珠笔

如上述执行结果所示，所有的行中都显示出了 SELECT 子句中的常数。此外，SELECT 子句中除了书写常数，还可以书写计算式。我们将在下一节中学习如何书写计算式。

从结果中删除重复行

想知道 Product 表中保存了哪些商品种类 (product\_type) 时，如果能像图 2-2 那样删除重复的数据该有多好啊。



图2-2 除去重复数据后的商品种类

product_id (商品编号)	product_name (商品名称)	product_type (商品种类)	sale_price (销售单价)	purchase_price (进货单价)	regist_date (登记日期)
0001	T恤衫	衣服	1000	500	2009-09-20
0002	打孔器	办公用品	500	320	2009-09-11
0003	运动T恤	衣服	4000	2800	
0004	菜刀	厨房用具	3000	2800	2009-09-20
0005	高压锅	厨房用具	6800	5000	2009-01-15
0006	叉子	厨房用具	500		2009-09-20
0007	擦菜板	厨房用具	880	790	2008-04-28
0008	圆珠笔	办公用品	100		2009-11-11

删除重复数据

product_type (商品种类)
衣服
办公用品
厨房用具

KEYWORD

● DISTINCT 关键字


如上所示，想要删除重复行时，可以通过在 SELECT 子句中使用 **DISTINCT** 来实现（代码清单 2-7）。

代码清单 2-7 使用 **DISTINCT** 删除 **product\_type** 列中重复的数据

```
SELECT DISTINCT product_type
FROM Product;
```

执行结果

```
product_type
-----
厨房用具
衣服
办公用品
```

 法则 2-3

在 SELECT 语句中使用 **DISTINCT** 可以删除重复行。

在使用 **DISTINCT** 时，**NULL** 也被视为一类数据。**NULL** 存在于多行中时，也会被合并为一条 **NULL** 数据。对含有 **NULL** 数据的 **purchase\_**

price（进货单价）列使用 DISTINCT 的 SELECT 语句请参见代码清单 2-8。除了两条 2800 的数据外，两条 NULL 的数据也被合并为一条。

代码清单2-8 对含有NULL数据的列使用DISTINCT关键字

```
SELECT DISTINCT purchase_price  
FROM Product;
```

执行结果

```
purchase_price  
-----  
5000  
790  
500  
2800  
320
```

NULL数据被保留了下来

DISTINCT 也可以像代码清单 2-9 那样在多列之前使用。此时，会将多个列的数据进行组合，将重复的数据合并为一条。代码清单 2-9 中的 SELECT 语句，对 product\_type（商品种类）列和 regist\_date（登记日期）列的数据进行组合，将重复的数据合并为一条。

代码清单2-9 在多列之前使用DISTINCT

```
SELECT DISTINCT product_type, regist_date  
FROM Product;
```

执行结果

```
product_type |regist_date  
-----+-----  
衣服         | 2009-09-20  
办公用品     | 2009-09-11  
办公用品     | 2009-11-11  
衣服         |  
厨房用具     | 2009-09-20  
厨房用具     | 2009-01-15  
厨房用具     | 2008-04-28
```

如上述执行结果所示，product\_type 列为 '厨房用具'，同时 regist\_date 列为 '2009-09-20' 的两条数据被合并成了一条。

DISTINCT 关键字只能用在第一个列名之前。因此，请大家注意不能写成 regist\_date, DISTINCT product\_type。

根据WHERE语句来选择记录

前面的例子都是将表中存储的数据全都选取出来，但实际上并不是每次都需要选取全部数据，大部分情况都是要选取满足“商品种类为衣服”“销售单价在 1000 日元以上”等某些条件的数据。

KEYWORD

WHERE子句

注

这和Excel中根据过滤条件对行进行过滤的功能是相同的。

SELECT 语句通过 WHERE 子句来指定查询数据的条件。在 WHERE 子句中可以指定“某一列的值和这个字符串相等”或者“某一列的值大于这个数字”等条件。执行含有这些条件的 SELECT 语句，就可以查询出只符合该条件的记录了。

在 SELECT 语句中使用 WHERE 子句的语法如下所示。

语法2-3 SELECT语句中的WHERE子句

```
SELECT <列名>, .....  
FROM <表名>  
WHERE <条件表达式>;
```

图 2-3 显示了从 Product 表中选取商品种类 (product\_type) 为 '衣服' 的记录。

图2-3 选取商品种类为'衣服'的记录

product_id (商品编号)	product_name (商品名称)	product_type (商品种类)	sale_price (销售单价)	purchase_price (进货单价)	regist_date (登记日期)
0001	T恤衫	衣服	1000	500	2009-09-20
0002	打孔器	办公用品	500	320	2009-09-11
0003	运动T恤	衣服	4000	2800	
0004	菜刀	厨房用具	3000	2800	2009-09-20
0005	高压锅	厨房用具	6800	5000	2009-01-15
0006	叉子	厨房用具	500		2009-09-20
0007	擦菜板	厨房用具	880	790	2008-04-28
0008	圆珠笔	办公用品	100		2009-11-11

选取product\_type列为'衣服'的记录

从被选取的记录中还可以查询出想要的列。为了更加容易理解，我们在查询 product\_type 列的同时，把 product\_name 列也读取出来。SELECT 语句请参见代码清单 2-10。

代码清单2-10 用来选取product\_type列为'衣服'的记录的SELECT语句

```
SELECT product_name, product_type
FROM Product
WHERE product_type = '衣服';
```

执行结果

product_name	product_type
T恤衫	衣服
运动T恤	衣服

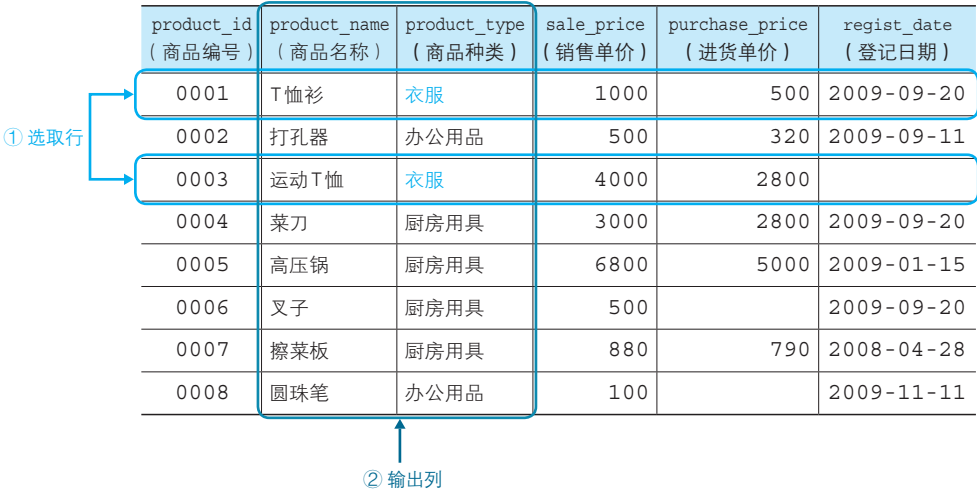
KEYWORD

● 条件表达式

WHERE 子句中的“product\_type = '衣服'”就是用来表示查询条件的表达式（[条件表达式](#)）。等号是比较两边的内容是否相等的符号，上述条件就是将 product\_type 列的值和 '衣服' 进行比较，判断是否相等。Product 表的所有记录都会被进行比较。

接下来会从查询出的记录中选取 SELECT 语句指定的 product\_name 列和 product\_type 列，如执行结果所示，也就是[首先通过 WHERE 子句查询出符合指定条件的记录，然后再选取 SELECT 语句指定的列](#)（图 2-4）。

图2-4 选取行之后，再输出列



代码清单 2-10 中的语句为了确认选取出的数据是否正确，通过 SELECT 子句把作为查询条件的 product\_type 列也选取出来了，其实这并不是必须的。如果只想知道商品名称的话，可以像代码清单 2-11 那

样只选取出 product\_name 列。

代码清单 2-11 也可以不选取出作为查询条件的列

```
SELECT product_name
FROM Product
WHERE product_type = '衣服';
```

执行结果

```
product_name
-----
T恤衫
运动T恤
```

SQL 中子句的书写顺序是固定的，不能随意更改。WHERE 子句必须紧跟在 FROM 子句之后，书写顺序发生改变的话会造成执行错误（代码清单 2-12）。

代码清单 2-12 随意改变子句的书写顺序会造成错误

```
SELECT product_name, product_type
WHERE product_type = '衣服'
FROM Product;
```

执行结果 (PostgreSQL)

```
ERROR:  "FROM"或者其前后有语法错误
第3行: FROM Product;
      ^
```



#### 法则 2-4

WHERE 子句要紧跟在 FROM 子句之后。

## 注释的书写方法

最后给大家介绍一下注释的书写方法。注释是 SQL 语句中用来标识说明或者注意事项的部分。

注释对 SQL 的执行没有任何影响。因此，无论是英文字母还是汉字都可以随意使用。

注释的书写方法有如下两种。

### KEYWORD

● 注释

**KEYWORD**

- 1行注释

- --

注 ①

MySQL中需要在“--”之后加入半角空格(如果不加的话就不会被认为是注释)。

**KEYWORD**

- 多行注释

- /\*

- \*/

- 1行注释

书写在“--”之后,只能写在同一行。<sup>①</sup>

- 多行注释

书写在“/\*”和“\*/”之间,可以跨多行。

实际的示例请参见代码清单 2-13 和代码清单 2-14。

#### 代码清单 2-13 1行注释的使用示例

```
-- 本SELECT语句会从结果中删除重复行。
SELECT DISTINCT product_id, purchase_price
FROM Product;
```

#### 代码清单 2-14 多行注释的使用示例

```
/* 本SELECT语句,
   会从结果中删除重复行。*/
SELECT DISTINCT product_id, purchase_price
FROM Product;
```

任何注释都可以插在 SQL 语句中(代码清单 2-15、代码清单 2-16)。

#### 代码清单 2-15 在 SQL 语句中插入 1 行注释

```
SELECT DISTINCT product_id, purchase_price
-- 本SELECT语句会从结果中删除重复行。
FROM Product;
```

#### 代码清单 2-16 在 SQL 语句中插入多行注释

```
SELECT DISTINCT product_id, purchase_price
/* 本SELECT语句,
   会从结果中删除重复行。*/
FROM Product;
```

这些 SELECT 语句的执行结果与没有使用注释时完全一样。注释能够帮助阅读者更好地理解 SQL 语句,特别是在书写复杂的 SQL 语句时,希望大家能够尽量多加简明易懂的注释。注释不仅可以写在 SELECT 语句中,而且可以写在任何 SQL 语句当中,写多少都可以。



#### 法则 2-5

注释是 SQL 语句中用来标识说明或者注意事项的部分。  
分为 1 行注释和多行注释两种。