TURING 图灵程序设计丛书

Apress®

Pro Oracle SQL Second Edition

精通 Oracle SQL

（第2版）

Oracle ACE与OakTable团队专家合力打造，实用典范

[美] Karen Morton
Kerry Osborne
Robyn Sands
Riyaj Shamsudeen
Jared Still 著

朱浩波 译

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

Pro Oracle SQL Second Edition

精通 Oracle SQL

（第2版）

[美] Karen Morton
Kerry Osborne
Robyn Sands
Riyaj Shamsudeen
Jared Still 著

朱浩波 译

人民邮电出版社
北京

图书在版编目（CIP）数据

精通Oracle SQL : 第2版 / (美) 莫顿 (Morton, K.) 等著 ; 朱浩波译. -- 北京 : 人民邮电出版社, 2014.5
(图灵程序设计丛书)
ISBN 978-7-115-35166-1

Ⅰ. ①精… Ⅱ. ①莫… ②朱… Ⅲ. ①关系数据库系统 Ⅳ. ①TP311.138

中国版本图书馆CIP数据核字(2014)第062937号

内 容 提 要

本书语言精炼、风趣，所涵盖的内容涉及 SQL 核心、SQL 执行、分析函数、联结、测试与质量保证等，并提供大量实用性建议，且总结出方方面面的“技巧”，帮助读者在阅读过程中快速消化所看内容。新版针对 Oracle 12*c* 进行了大幅修订，以反映技术的最新发展动态。

本书旨在为 Oracle 数据库开发人员、DBA 和架构师提供参考。

◆ 著 [美] Karen Morton Kerry Osborne Robyn Sands Riyaj Shamsudeen Jared Still

译 朱浩波

责任编辑 丁晓昀

责任印制 焦志炜

◆ 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

北京鑫正大印刷有限公司印刷

◆ 开本：800×1000 1/16

印张：33.75

字数：798千字 2014年5月第1版

印数：1－4 000册 2014年5月北京第1次印刷

著作权合同登记号 图字：01-2014-2008号

定价：99.00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：京崇工商广字第 0021 号

版 权 声 明

致　　谢

我要感谢本书的另几位合著者以及Apress公司里那些了不起的员工。我还要感谢家人的支持和理解。谢谢你们所有人，有了你们，本书的升级版任务变得容易了许多。

——Karen Morton

目　录

第1章 SQL核心

不管你是刚开始写SQL语句还是已经写很多年了，学会写出“好的”SQL这个过程都需要具有很扎实的SQL核心语法和概念基础知识。本章对SQL语言的核心概念及其性能做了回顾，同时还描述了一些你应该已经很熟悉的常用SQL命令。对于那些以前曾经使用过SQL并且基础知识相当牢靠的读者来说，本章就是一个简要的复习，让你为后面更详细的SQL论述做好准备。如果你是一位SQL新人，你可以先阅读*Beginning Oracle SQL*这本书，以确保掌握了SQL的基础。不管是哪种情况，第1章的目的就是快速浏览5个核心SQL语句，衡量一下你的SQL水平。本章还概述了用来执行SQL语句的工具：SQL*Plus。

1.1 SQL 语言

SQL语言最早是IBM公司于20世纪70年代开发出来的，称为结构化英文查询语言，简称为SEQUEL。该语言是基于E. F. Codd在1969年提出的关系型数据库管理系统（RDBMS）的。后来因为商标的纠纷，其简称又进一步缩写为SQL。1986年和1987年，ANSI（美国国家标准化组织）和ISO（国际标准化组织）先后将SQL语言采纳为标准语言。而人们并不熟悉的是，ANSI官方曾将SQL语言的读音确定为“S-Q-L”。绝大多数人，包括我本人，都还在使用“sequel”的读音，只是因为这样读起来更顺口一些。

SQL的目的就是提供一个到数据库的接口，在本书指的是Oracle数据库。每一条SQL语句对于数据库来说就是一条命令或指令。SQL与其他编程语言（如C或Java）的区别就在于它是要处理数据集合而不是一行一行的数据。语言本身也不需要你提供如何导航到数据的指令——这是在后台透明地进行的。但你将在后面的章节中看到，如果想在Oracle中写出高效的SQL语句，了解数据及其在数据库中的存储方式与存储位置是很重要的。

由于不同的供应商（例如甲骨文、IBM和微软）实现SQL核心功能的机制相差无几，所以基于某一种数据库所学的技巧同样可以应用到其他类型的数据库上。你基本上可以利用同样的SQL语句来进行数据的查询、插入、更新和删除，以及创建、修改和删除对象，而不必管数据库的供应商是哪家。

尽管SQL是各种关系型数据库管理系统的标准语言，但实际上它并不一定是关系型的。在本书后面我将就这一点稍作扩展。如果想要了解更多的细节，我推荐大家阅读C. J. Date的*SQL and*

*Relational Theory*一书。需要铭记于心的一点是SQL语言并不总是严格遵守关系模型的——它根本就没有实现关系模型的某些要素，同时还不恰当地实现了一些要素。事实上，既然SQL是基于关系模型的，那么要想写出尽可能正确高效的SQL语句，你不仅必须要理解SQL语言，还要理解关系模型。

1.2　数据库的接口

多年以来人们开发出多种途径来传递SQL语句到数据库并获得结果。Oracle数据库的本地接口界面是Oracle调用界面（OCI）。OCI将由Oracle内核传送而来的查询语句发送到数据库。当使用某种Oracle工具如SQL*Plus或者SQL Developer时，你都在使用OCI。其他的Oracle工具如SQL*Loader、数据泵以及Real Application Testing（RAT）既可以使用OCI，也可以使用语言特定的接口，如Oracle JDBC-OCI、ODP.Net、Oracle预编译器、Oracle ODBC以及Oracle C++调用接口（OCCI）驱动。

当使用编程语言（如COBOL或C语言）时，你所写的语句被称为*嵌入式*的SQL语句，并且在应用程序编译之前会由SQL预处理器进行预处理。代码清单1-1是一段可以在C/C++程序块中使用的SQL语句的例子。

代码清单1-1　C/C++程序块中所嵌入的SQL语句

```
{
int a;
/* ... */
EXEC SQL SELECT salary INTO :a
    FROM hr.employees
   WHERE employee_id = 108;
/* ... */
printf("The salary is %d\n", a);
/* ... */
}
```

其他工具，例如SQL*Plus和SQL Developer，都是交互式的工具。你输入并执行命令，然后获得相应的输出。交互式工具并不需要在运行代码前先精确编译，你只需要输入想要执行的命令即可。代码清单1-2是一段使用SQL*Plus执行语句的例子。

代码清单1-2　使用SQL*Plus执行SQL语句

```
SQL> select  salary
  2  from    hr.employees
  3  where   employee_id = 108;

        SALARY
---------------
         12000
```

在本书中，为了保持一致性我们所用的示例代码清单都使用SQL*Plus工具，但需要记住的是，不管你是用什么方法或工具输入或执行SQL语句，所有的事情最后都要通过OCI来传递到数据库。

这里的主旨就是不管你所使用的是什么工具，其本地接口都是一样的。

1.3 SQL*Plus 回顾

SQL*Plus是一个不管采用哪个安装平台（Windows或Unix）都会提供的命令行工具。它是一个用来输入和执行SQL语句并显示输出结果的纯文本环境。用该工具可以直接输入、编辑命令，可以一条条地保存和执行命令或者通过脚本文件来进行，然后将输出结果以很精美格式的报表输出。要启动SQL*Plus你只需要在主机的命令提示符后敲入`sqlplus`即可。

1.3.1 连接到数据库

SQL*Plus连接数据库的方法有很多。然而在连接之前，你还需要在$ORACLE_HOME/network/admin/tnsnames.ora这个文件中登记想要连接的数据库。有两种常用的方法，或者如代码清单1-3所示那样在启动SQL*Plus时提供连接信息，或者如代码清单1-4所示那样在启动SQL*Plus以后使用`connect`命令。

代码清单1-3 通过窗口命令提示符连接SQL*Plus

```
$ sqlplus hr@ora12c

SQL*Plus: Release 12.1.0.1.0 Production on Tue May 7 12:32:36 2013

Copyright (c) 1982, 2013, Oracle.  All rights reserved.
Enter password:

Last Successful login time: Tue May 07 2013 12:29:09 -04:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

SQL>
```

代码清单1-4 通过SQL>提示符连接SQL*Plus并登录数据库

```
$ sqlplus /nolog

SQL*Plus: Release 12.1.0.1.0 Production on Tue May 7 12:34:21 2013

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> connect hr@ora12c
Enter password:
Connected.
SQL>
```

如果你使用的是SQL*Plus 12*c*，默认会显示最后一次登录时间。如果想禁用此功能，可以使用`-no logintime`选项。

如果想要启动SQL*Plus而又不显示登录到数据库后的提示，可以在启动SQL*Plus时使用/nolog选项。

1.3.2　配置SQL*Plus环境

SQL*Plus有很多的命令可以定制工作环境和显示选项。SQL*Plus的help数据必须由DBA安装后方可使用。$ORACLE_HOME/SQL PLUS/AD MIN/HELP/中存有3个help脚本（HLPBLD.SQL、HELPDROP.SQL和HELPUS.SQL），可用于创建、删除和移动help表。代码清单1-5所示是在SQL>提示符下输入help index命令后显示出来的可用的命令。

代码清单1-5　SQL*Plus命令列表

```
SQL> help index

Enter Help [topic] for help.
```

@	COPY	PAUSE	SHUTDOWN
@@	DEFINE	PRINT	SPOOL
/	DEL	PROMPT	SQLPLUS
ACCEPT	DESCRIBE	QUIT	START
APPEND	DISCONNECT	RECOVER	STARTUP
ARCHIVE LOG	EDIT	REMARK	STORE
ATTRIBUTE	EXECUTE	REPFOOTER	TIMING
BREAK	EXIT	REPHEADER	TTITLE
BTITLE	GET	RESERVED WORDS (SQL)	UNDEFINE
CHANGE	HELP	RESERVED WORDS (PL/SQL)	VARIABLE
CLEAR	HOST	RUN	WHENEVER OSERROR
COLUMN	INPUT	SAVE	WHENEVER SQLERROR
COMPUTE	LIST	SET	XQUERY
CONNECT	PASSWORD	SHOW	

set命令是用来定制工作环境的最基本的命令。代码清单1-6为set命令的帮助文本。

代码清单1-6　SQL*Plus的SET命令

```
SQL> help set

 SET
 ---

 Sets a system variable to alter the SQL*Plus environment settings
 for your current session. For example, to:
     -   set the display width for data
     -   customize HTML formatting
```

```
    -   enable or disable printing of column headings
    -   set the number of lines per page

 SET system_variable value
 where system_variable and value represent one of the following clauses:

APPI[NFO]{OFF|ON|text}
ARRAY[SIZE] {15|n}
AUTO[COMMIT] {OFF|ON|IMM[EDIATE]|n}
AUTOP[RINT] {OFF|ON}
AUTORECOVERY {OFF|ON}
AUTOT[RACE] {OFF|ON|TRACE[ONLY]}
   [EXP[LAIN]] [STAT[ISTICS]]
BLO[CKTERMINATOR] {.|c|ON|OFF}
CMDS[EP] {;|c|OFF|ON}
COLSEP {_|text}
CON[CAT] {.|c|ON|OFF}
COPYC[OMMIT] {0|n}
COPYTYPECHECK {ON|OFF}
DEF[INE] {&|c|ON|OFF}
DESCRIBE [DEPTH {1|n|ALL}]
   [LINENUM {OFF|ON}] [INDENT {OFF|ON}]
ECHO {OFF|ON}
EDITF[ILE] file_name[.ext]
EMB[EDDED] {OFF|ON}
ERRORL[OGGING] {ON|OFF}
   [TABLE [schema.]tablename]
   [TRUNCATE] [IDENTIFIER identifier]

ESC[APE] {\|c|OFF|ON}
ESCCHAR {@|?|%|$|OFF}
EXITC[OMMIT] {ON|OFF}
FEED[BACK] {6|n|ON|OFF}
FLAGGER {OFF|ENTRY|INTERMED[IATE]|FULL}
FLU[SH] {ON|OFF}
HEA[DING] {ON|OFF}
HEADS[EP] {||c|ON|OFF}
INSTANCE [instance_path|LOCAL]
LIN[ESIZE] {80|n}
LOBOF[FSET] {1|n}
LOGSOURCE [pathname]
LONG {80|n}
LONGC[HUNKSIZE] {80|n}
MARK[UP] HTML [OFF|ON]
   [HEAD text] [BODY text] [TABLE text]
   [ENTMAP {ON|OFF}]
   [SPOOL {OFF|ON}]
   [PRE[FORMAT] {OFF|ON}]
SQL>

NEWP[AGE] {1|n|NONE}
       NULL text
NUMF[ORMAT] format
NUM[WIDTH] {10|n}
PAGES[IZE] {14|n}
       PAU[SE] {OFF|ON|text}
RECSEP {WR[APPED]|EA[CH]|OFF}
       RECSEPCHAR {_|c}
SERVEROUT[PUT] {ON|OFF}
                [SIZE {n | UNLIMITED}]
        [FOR[MAT]  {WRA[PPED] |
                WOR[D_WRAPPED] |
         TRU[NCATED]}]
SHIFT[INOUT] {VIS[IBLE] |
                        INV[ISIBLE]}
SHOW[MODE] {OFF|ON}
       SQLBL[ANKLINES] {OFF|ON}
SQLC[ASE] {MIX[ED] |
        LO[WER] | UP[PER]}
       SQLCO[NTINUE] {> | text}
SQLN[UMBER] {ON|OFF}
SQLPLUSCOMPAT[IBILITY]
                {x.y[.z]}
SQLPRE[FIX] {#|c}
SQLP[ROMPT] {SQL>|text}
SQLT[ERMINATOR] {;|c|ON|OFF}
       SUF[FIX] {SQL|text}
TAB {ON|OFF}
       TERM[OUT] {ON|OFF}
       TI[ME] {OFF|ON}
TIMI[NG] {OFF|ON}
       TRIM[OUT] {ON|OFF}
       TRIMS[POOL] {OFF|ON}
       UND[ERLINE] {-|c|ON|OFF}
VER[IFY] {ON|OFF}
                WRA[P] {ON|OFF}
XQUERY {BASEURI text|
       ORDERING{UNORDERED|
       ORDERED|DEFAULT}|
       NODE{BYVALUE|BYREFERENCE|
       DEFAULT}|
       CONTEXT text}
```

有了上面这些可用命令，你就能够很轻松地定制最适合你的运行环境了。但有一点要铭记于心的就是，当你退出或关闭SQL*Plus的时候，这些设置命令不会被保留。为了避免每次使用SQL*Plus时都重新敲入一遍这些设置命令，你可以创建一个login.sql文件。事实上每次启动

SQL*Plus的时候它都会默认去读两个文件。第一个是$ORACLE_HOME/sqlplus/admin目录下的glogin.sql文件。如果找到了这个文件，它就会被读进来，文件中的命令语句也会被执行。这样就可以把那些定制你的会话体验的SQL*Plus命令和SQL语句保存起来。

在读取glogin.sql文件以后，SQL*Plus会进一步寻找login.sql文件。这个文件必须在SQL*Plus的启动文件夹中或者包含在环境变量SQLPATH所指向的文件夹路径中。在login.sql文件中的所有命令优先级都比glogin.sql文件中的命令高。从10*g*开始，Oracle在每次启动SQL*Plus或者从SQL*Plus里执行connect命令时都会同时读取glogin.sql和login.sql这两个文件。在Oracle 10*g*之前，login.sql脚本文件只有在SQL*Plus启动时才会执行。代码清单1-7是一个常见的login.sql文件内容。

代码清单1-7　一个常见的login.sql文件

```
SET LINES 3000
--Sets width of display line (default 80 characters)
SET PAGES 1000
-Sets number of lines per page (default 14 lines)
SET TIMING ON
--Sets display of elapsed time (default OFF)
SET NULL <null>
--Sets display of nulls to show <null> (default empty)
SET SQLPROMPT '&_user@&_connect_identifier> '
--Sets the prompt to show connected user and instance
```

注意这里在SET SQLPROMPT中使用的变量_user和_connect_identifier。它们是预定义变量的两个示例。你可以在login.sql文件中或者任何你创建的脚本文件中使用下面这些预定义变量：

- _connect_identifier（用于指定数据库连接的连接标识符）
- _date（当前日期）
- _editor（这个变量指定了当你使用EDIT命令时启动哪个编辑器）
- _o_version（当前Oracle数据库的版本）
- _o_release（当前Oracle数据库的完整版本号）
- _privilege（当前连接的权限级别）
- _sqlplus_release（当前安装的SQL*Plus模块的完整版本号）
- _user（当前连接使用的用户名）

1.3.3　执行命令

有两种命令可以在SQL*Plus中执行：SQL语句和SQL*Plus命令。代码清单1-5和代码清单1-6中列出的SQL*Plus命令对于SQL*Plus来说是特有的命令，可以用来定制运行环境并且可以运行SQL*Plus特有的命令，例如DESCRIBE和CONNECT。要想执行SQL*Plus命令，你只需在命令提示符后输入该命令然后敲回车，命令就会自动执行。另一方面，如果要执行SQL语句，就必须使用一个特定字符来表明你想要执行输入的语句，分号（;）或者斜线（/）都可以。使用分号可以直接放在输入命令的后面或者放在接下来的空行中，而斜线则必须放在接下来的空行中才可以识别。

代码清单1-8展示了如何使用这两种符号。

代码清单1-8 执行字符的用法

```
SQL>select empno, deptno from scott.emp where ename = 'SMITH' ;
     EMPNO     DEPTNO
---------- ----------
      7369         20
SQL>select empno, deptno from scott.emp where ename = 'SMITH'
  2  ;
     EMPNO     DEPTNO
---------- ----------
      7369         20
SQL>select empno, deptno from scott.emp where ename = 'SMITH'
  2  /
     EMPNO     DEPTNO
---------- ----------
      7369         20
SQL>select empno, deptno from scott.emp where ename = 'SMITH'
  2
SQL>/
     EMPNO     DEPTNO
---------- ----------
      7369         20
SQL>select empno, deptno from scott.emp where ename = 'SMITH'/
  2
SQL>l
  1* select empno, deptno from scott.emp where ename = 'SMITH'/
SQL>/
select empno, deptno from scott.emp where ename = 'SMITH'/
                                                        *
ERROR at line 1:
ORA-00936: missing expression
```

注意第5个在语句最后面加了一个斜线（/）的例子。光标移动到了下一行而不是立即执行语句命令。接下来，如果你再按一下回车键，语句就会放入SQL*Plus的缓冲器中，但是也不执行。如果想要查看SQL*Plus缓冲器中的内容，可以使用list命令（也可以简写为l）。接下来如果你想在缓冲器中使用斜线（/）来执行语句（尽管斜线（/）命令本来就是这样来用的）在这里也将会返回一个错误。这是因为你最初在SQL语句的结尾敲入了一个斜线（/），而斜线（/）并不是有效的SQL命令，从而在语句执行的时候报错。

另外一种执行命令的方法是把命令放到一个文件中。你可以在SQL*Plus之外直接用文本编辑器生成这些文件，也可以在SQL*Plus中使用EDIT命令来直接调用编辑器。如果已经有了一个文件，EDIT命令可以打开这个文件，如果没有就会创建新的文件。文件必须放在默认文件夹中，否则必须指定文件的全路径。想要设定选择的编辑器，你只需要利用命令define_ editor='/<full path>/myeditor.exe'来设置预定义变量_editor即可。具有.sql扩展名的文件在执行的时候不必敲入扩展名，使用@或START命令都可以执行。代码清单1-9中展示了这两个命令的用法。