**2、连接Oracle数据库，然后探索它**

你可以仅通过一个客户端程序(如SQL Plus或者SQL Developer)连接到Oracle数据库

当连接到数据库时，你可以查看模式对象、查看Oracle数据库表的属性和数据，以及使用查询从Oracle数据库表检索数据。

通过客户端程序连接到Oracle数据库之后，输入并运行客户端程序中的命令。有关详细信息，请参阅文档

• 通过SQL \*Plus连接到Oracle数据库

SQL\*Plus是一个客户端程序，您可以从中访问Oracle数据库。这个主题展示了如何启动SQL\*Plus并连接到Oracle数据库。

• 通过SQL Developer连接到Oracle数据库

SQL Developer是一个客户端程序，您可以用它访问Oracle数据库。

•作为用户HR连接到Oracle数据库

要完成本文档中的教程和示例，必须作为用户HR连接到Oracle

•使用SQL\*Plus探索Oracle数据库

如果作为用户HR通过SQL\*Plus连接到Oracle数据库，则可以查看HR模式对象和EMPLOYEES表的属性。

•使用SQL Developer探索Oracle数据库

如果作为用户HR通过SQL Developer连接到Oracle数据库，则可以查看HR模式对象和EMPLOYEES表的属性。

•选择表数据

**2.1、通过SQL\*Plus连接到Oracle数据库**

SQL\*Plus是一个客户端程序，您可以从中访问Oracle数据库。本节演示如何启动SQL\*Plus并连接到Oracle数据库。

注：

对于以下过程的步骤3和4，需要用户名和密码。

从SQL\*Plus连接到Oracle数据库：

1、如果在Windows系统上，则显示Windows命令提示符。

2、在命令提示符处，键入sqlplus，然后按下键Enter。

3、在用户名提示符处，键入用户名，然后按Enter键。

4、在密码提示符下，键入密码，然后按Enter键。

注：

为了安全起见，您的密码在屏幕上不可见。

系统将您连接到Oracle数据库实例

您处于SQL\*Plus环境中。在SQL>提示符处，可以输入并运行

SQL\*Plus命令、SQL语句、PL/SQL语句和操作系统命令。

要退出SQL\*Plus，键入exit并按下Enter键。

注：

退出SQL\*Plus结束SQL\*Plus会话，但不关闭

Oracle数据库实例。

示例2-1启动SQL\*Plus，连接到Oracle数据库，运行SQL SELECT

语句，并退出SQL\*Plus。用户输入是粗体。

示例2-1，从SQL\*Plus连接到Oracle数据库

*> sqlplus  
SQL\*Plus: Release 12.1.0.1.0 Production on Thu Dec 27 07:43:41 2012  
Copyright (c) 1982, 2012, Oracle. All rights reserved.  
Enter user-name: your\_user\_name  
Enter password: your\_password  
Connected to:  
Oracle Database 12c Enterprise Edition Release - 12.1.0.1.0 64bit Production  
SQL> select count(\*) from employees;  
COUNT(\*)  
----------  
107*

*SQL>* ***exit****Disconnected from Oracle Database 12c Enterprise Edition Release - 12.1.0.1.0 64bit Production*

参见：

•“从SQL\*Plus作为用户HR连接到Oracle数据库”

•“关于SQL\*Plus”，用于简要描述SQL\*Plus

·SQL\*加上用户指南和参考，以获取关于启动的更多信息

SQL\*Plus并连接到Oracle数据库

**2.2、通过SQL Developer连接到Oracle数据库**

SQL Developer是一个客户端程序，您可以用它访问Oracle数据库。

推荐您使用当前可用的SQL Developer版本，您可以从：http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/进行安装。本节假设您的系统上已经安装了SQL Developer，并演示如何启动它并连接到Oracle数据库。如果系统上没有安装SQL Developer，那么请参阅Oracle SQL Developer用户指南以获得安装说明。

**注：**

对于下列程序：

•如果您使用的SQL Developer工具包不包括JDK，则第一次在系统上启动SQL Developer时，必须提供Java开发工具包的路径。

•当提示时，您需要输入用户名和密码。

从SQL Developer连接到Oracle数据库：

1、启动SQL Developer。

有关说明，请参阅Oracle SQL Developer用户指南。如果这是第一次在您的系统上启动SQL Developer，那么提示进入Java开发工具包(JDK)安装的路径(例如，C：\Program Files\java\jdk1.80\_65)。在提示符后键入路径或者浏览它，然后按下回车键。

2、在“连接”框中，单击图标“新建连接”。

3、在“新建/选择数据库连接”窗口中：

a.输入“连接名称”、“用户名”和“密码”。

为了安全起见，您输入的密码字符显示为星号。在“密码”字段附近是复选框“保存密码”。默认情况下，它是未选中的。Oracle建议接受默认设置。

b.如果Oracle窗格未显示，请单击Oracle选项

C.在Oracle窗格中，接受默认值。

(默认值是：Connection Type，Basic；Role，default，Hostname，localhost；Port，1521；SID option；SID，xe.)

d.单击Test按钮。

连接测试。如果连接成功，状态指示符从空白更改为“success”。

e.如果测试成功，单击Connect按钮。

关闭“新建/选择数据库连接”窗口。在Connections窗口显示您步骤3中输入的用户名。

您现在处于SQL Developer环境中。

要退出SQL Developer，请从“文件”菜单中选择“退出”。

**注：**

退出SQL Developer结束SQL Developer会话，但不关闭Oracle数据库实例。下次启动SQL Developer时，前面过程创建的连接仍然存在。SQL Developer提示您输入步骤3中提供的密码(除非选中“保存密码”复选框)。

**参见：**

•“从SQL Developer以用户HR身份连接到Oracle数据库”

•“关于SQL Developer”，用于简要描述SQL Developer

·Oracle SQL Developer指南，以获取关于使用SQL Developer连接Oracle数据库的信息

**2.3作为用户HR连接到Oracle数据库**

要完成本文档中的教程和示例，您必须作为用户HR连接到Oracle数据库。HR用户拥有本文中的示例和教程使用的HR示例模式。

•解锁HR账户

在作为用户HR连接到Oracle数据库之前，必须解锁HR帐户并重新设置其密码。

·通过SQL\*Plus作为用户HR连接到Oracle数据库

您可以使用SQL\*Plus作为HR用户连接到Oracle数据库。

·从SQL Developer作为用户HR连接到Oracle数据库

您可以使用SQL Developer作为HR用户连接到Oracle数据库。

**2.3.1 解锁HR账户**

在以HR身份连接Oracle数据库之前，必须解锁HR账户并重新设置密码。默认情况下，在安装HR模式时，HR帐户被锁定，其密码过期。

**注：**

对于以下过程，您需要具有ALTER USER 系统特权的用户的用户名和密码来解锁HR帐户并重新设置其密码：

1、使用SQL\*Plus，作为具有ALTER USER系统特权的用户连接到Oracle数据库。

2、在SQL>提示符下，解锁HR帐户并重新设置其密码：

**警告：**

选择一个安全的密码。有关安全密码的指南，请参见Oracle数据库安全指南。

更改用户HR账号解锁密码；

系统响应：User altered.。

HR帐户现在是解锁的，它的密码就是你刚才设置的密码。现在可以使用新密码作为用户HR连接到Oracle数据库。

参见：

·Oracle Database 2 Day+ Security Guide，提供包括HR在内的有关预定义的信息和Oracle数据库提供的用户帐户。

·Oracle SQL Developer用户指南，用于访问相关SQL Developer中的SQL\*Plus信息

**2.3.2通过SQL\*Plus作为用户HR连接到Oracle数据库**

您可以使用SQL\*Plus作为HR用户连接到Oracle数据库。

**注：**

如果HR帐户被锁定，请参见“解锁HR帐户”，然后返回这部分。

通过SQL\*Plus作为用户HR连接到Oracle数据库

**注：78**

对于这个任务，您需要HR帐户的密码

1、如果连接到Oracle数据库，请关闭当前连接。

2。按照“从SQL\*Plus连接到Oracle数据库”中的说明，输入

步骤3中的用户名HR，步骤4中的HR帐户的密码。

您现在作为用户HR连接到Oracle数据库。

参见：

SQL\*Plus用户指南和参考示例创建HR连接

**2.3.3通过SQL Developer以HR角色连接到Oracle数据库**

您可以以HR用户使用SQL Developer连接到Oracle数据库。

**注：**

如果HR帐户被锁定，请参见“解锁HR帐户”，然后返回这部分。

通过SQL\*Plus作为用户HR连接到Oracle数据库

**注：**

对于这个任务，您需要HR帐户的密码

1、如果连接到Oracle数据库，请关闭当前连接。

2。按照“从SQL\*Plus连接到Oracle数据库”中的说明，输入

步骤3中的用户名HR，步骤4中的HR帐户的密码。

您现在作为用户HR连接到Oracle数据库。

参见：

SQL\*Plus用户指南和参考示例创建HR连接

遵循“Connecting to Oracle Database from SQL Developer ”中的说明

在步骤3输入以下值：

•对于连接名称，输入hr\_conn。

(您可以输入不同的名称，但是本文中的教程假定您命名了连接hr\_conn)。

•对于用户名，输入HR。

•对于密码，输入HR帐户的密码。

您现在作为用户HR连接到Oracle数据库。

**2.4使用SQL\*Plus开发Oracle数据库**

如果作为用户HR从SQL\*Plus连接到Oracle数据库，则可以查看HR模式对象和EMPLOYEES表的属性。

**注：**

如果您没有从SQL\*Plus作为用户HR连接到Oracle数据库，请参见“从SQL\*Plus作为用户HR连接到Oracle数据库”，然后返回这部分。

·使用SQL\*Plus查看HR模式对象

使用SQL\*Plus，可以通过查询静态视图USER\_OBJECTS。

·使用SQL\*Plus查看EMPLOYEES表属性和数据

您可以使用SQL\*Plus命令、SQLSELECT语句和静态数据

字典视图用于查看HR.EMPLOYEES表的属性和数据。

**2.4.1使用SQL\*Plus查看HR模式对象**

使用SQL\*Plus，可以通过查询查看属于HR模式的对象静态数据字典视图USER\_OBJECTS。

示例2-2显示了如何查看属于HR的对象的名称和数据类型以及计划。

示例2-2使用SQL\*Plus查看HR模式对象

COLUMN OBJECT\_NAME FORMAT A25  
COLUMN OBJECT\_TYPE FORMAT A25  
**SELECT OBJECT\_NAME, OBJECT\_TYPE FROM USER\_OBJECTS  
ORDER BY OBJECT\_TYPE, OBJECT\_NAME;**Result is similar to:  
OBJECT\_NAME OBJECT\_TYPE  
------------------------- -------------------------  
COUNTRY\_C\_ID\_PK INDEX  
DEPT\_ID\_PK INDEX  
DEPT\_LOCATION\_IX INDEX  
EMP\_DEPARTMENT\_IX INDEX  
EMP\_EMAIL\_UK INDEX  
EMP\_EMP\_ID\_PK INDEX  
EMP\_JOB\_IX INDEX  
EMP\_MANAGER\_IX INDEX

EMP\_NAME\_IX INDEX  
JHIST\_DEPARTMENT\_IX INDEX  
JHIST\_EMPLOYEE\_IX INDEX  
JHIST\_EMP\_ID\_ST\_DATE\_PK INDEX  
JHIST\_JOB\_IX INDEX  
JOB\_ID\_PK INDEX  
LOC\_CITY\_IX INDEX  
LOC\_COUNTRY\_IX INDEX  
LOC\_ID\_PK INDEX  
LOC\_STATE\_PROVINCE\_IX INDEX  
REG\_ID\_PK INDEX  
ADD\_JOB\_HISTORY PROCEDURE  
SECURE\_DML PROCEDURE  
DEPARTMENTS\_SEQ SEQUENCE  
EMPLOYEES\_SEQ SEQUENCE  
LOCATIONS\_SEQ SEQUENCE  
COUNTRIES TABLE  
DEPARTMENTS TABLE  
EMPLOYEES TABLE  
JOBS TABLE  
JOB\_HISTORY TABLE  
LOCATIONS TABLE  
REGIONS TABLE  
SECURE\_EMPLOYEES TRIGGER  
UPDATE\_JOB\_HISTORY TRIGGER  
EMP\_DETAILS\_VIEW VIEW  
34 rows selected.

参见：

•Oracle数据库参考，以获取有关USER\_OBJECTS的信息

•“选择表数据”以获取有关使用查询查看表的信息

数据

•“关于示例模式HR”，了解有关模式HR的一般信息

**2.4.2使用SQL \* Plus查看EMPLOYEES表属性和数据**

您可以使用SQL \* Plus命令，SQL SELECT语句和静态数据字典视图以查看HR.EMPLOYEES表的属性和数据。

您可以使用SQL \* Plus命令DESCRIBE来查看HR计划的EMPLOYEES表的列属性，使用SELECT语句来查看表中的数据。

要查看表的其他属性，请使用静态数据字典视图（for

例如，USER\_CONSTRAINTS，USER\_INDEXES和USER\_TRIGGERS）。

示例2-3显示了如何在HR中查看EMPLOYEES表的属性

示例2-3使用SQL \* Plus查看EMPLOYEES表属性

DESCRIBE EMPLOYEES

结果：

Name Null? Type  
---------------------------- -------- -------------  
EMPLOYEE\_ID NOT NULL NUMBER(6)  
FIRST\_NAME VARCHAR2(20)  
LAST\_NAME NOT NULL VARCHAR2(25)  
EMAIL NOT NULL VARCHAR2(25)  
PHONE\_NUMBER VARCHAR2(20)  
HIRE\_DATE NOT NULL DATE  
JOB\_ID NOT NULL VARCHAR2(10)  
SALARY NUMBER(8,2)  
COMMISSION\_PCT NUMBER(2,2)  
MANAGER\_ID NUMBER(6)  
DEPARTMENT\_ID NUMBER(4)

示例2-4显示了如何查看HR中EMPLOYEES表中的某些数据

示例2-4使用SQL \* Plus查看EMPLOYEES表的数据

COLUMN FIRST\_NAME FORMAT A20  
COLUMN LAST\_NAME FORMAT A25  
COLUMN PHONE\_NUMBER FORMAT A20  
**SELECT LAST\_NAME, FIRST\_NAME, PHONE\_NUMBER FROM EMPLOYEES  
ORDER BY LAST\_NAME;**

结果是：

LAST\_NAME FIRST\_NAME PHONE\_NUMBER  
------------------------- -------------------- --------------------  
Abel Ellen 011.44.1644.429267  
Ande Sundar 011.44.1346.629268  
Atkinson Mozhe 650.124.6234  
Austin David 590.423.4569  
Baer Hermann 515.123.8888  
Baida Shelli 515.127.4563  
Banda Amit 011.44.1346.729268  
Bates Elizabeth 011.44.1343.529268  
...  
Urman Jose Manuel 515.124.4469  
Vargas Peter 650.121.2004  
Vishney Clara 011.44.1346.129268  
Vollman Shanta 650.123.4234  
Walsh Alana 650.507.9811  
Weiss Matthew 650.123.1234  
Whalen Jennifer 515.123.4444  
Zlotkey Eleni 011.44.1344.429018  
107 rows selected.

参见：

•有关DESCRIBE的信息，请参见SQL \* Plus用户指南和参考

•“Selecting Table Data”以获取有关使用查询查看表的信息

数据

•Oracle数据库参考有关静态数据字典的信息

**2.5使用SQL Developer探索Oracle数据库**

如果您从SQL Developer以用户HR身份连接到Oracle数据库，则可以查看HR模式对象和EMPLOYEES表的属性。

•教程：使用SQL Developer查看HR模式对象

本教程介绍如何使用SQL Developer查看属于HR模式的对象 - 即如何浏览HR schema。

•教程：使用SQL Developer查看EMPLOYEES表属性和数据

本教程介绍如何使用SQL Developer查看EMPLOYEES表的属性和数据

**2.5.1教程：使用SQL Developer查看HR模式对象**

本教程介绍如何使用SQL Developer查看属于该对象的对象

HR模式 - 即如何浏览HR模式。

注意：

如果您未从SQL Developer以用户HR身份连接到Oracle数据库，

请参阅“从SQL Developer连接到Oracle数据库作为用户HR”和

然后返回本教程。

要浏览HR模式：

1.在“连接”框中，在hr\_conn图标的左侧，单击加号（+）。如果未连接到数据库，则显示“连接信息”窗口打开。如果已连接到数据库，则会扩展hr\_conn信息（请参阅步骤2中“单击确定”后面的信息）。

2.如果“连接信息”窗口打开：

a、在“用户名”字段中，输入hr。

b、在“密码”字段中，输入用户HR的密码。

c、单击确定。

hr\_conn信息展开：加号变为减号（ - ）,在hr\_conn图标下，出现了一个模式对象类型列表 - Tables，Views，索引等。 （如果单击减号，hr\_conn信息将折叠：减号变为加号，列表消失。）

参见：

•“Oracle SQL Developer用户指南”，以获取有关SQL的更多信息

开发者用户界面

•“关于示例模式HR”，了解有关模式HR的一般信息

**2.5.2教程：使用SQL Developer查看EMPLOYEES表属性和数据**

本教程介绍如何使用SQL Developer查看HR模式中的EMPLOYEES表的属性和数据

注意：

如果您没有浏览HR模式，请参阅“教程：查看HR模式”使用SQL Developer的对象“然后返回本教程。

要查看EMPLOYEES表的属性和数据：

1.在“连接”框中，展开“表”。

在Tables下，将显示HR模式中的表列表。

2.选择EMPLOYEES表。

在Oracle SQL Developer窗口的右侧框架的“列”窗格中，一个包含所有表的信息的框会出现。每列的右边都是它的列名，数据类型等。 （要查看所有列属性，请移动水平滚动条向右。）

3.在右侧框架中，单击选项卡数据。

将出现“数据”窗格，其中显示此表中所有记录的编号列表。（要查看更多记录，向下移动垂直滚动条。要查看更多列记录，将水平滚动条向右移动。）

4.在右侧框架中，单击“约束”选项卡。

将出现“约束”窗格，其中显示了此表上所有约束的列表。到了

每个约束的权限是它的属性 - 名称，类型，搜索条件等

上。 （要查看所有约束属性，请将水平滚动条向右移动。）

5.单击相应的选项卡，浏览其他属性。

要查看用于创建EMPLOYEES表的SQL语句，请单击“SQL”选项卡。

SQL语句显示在名为EMPLOYEES的窗格中。要关闭此窗格，

单击名称EMPLOYEES右侧的x。

参见：

有关SQL的更多信息，请参见“Oracle SQL Developer用户指南”

开发者用户界面

2.6选择表数据

注意：

要执行本节中的教程和示例，您必须以HR用户的身份用SQL Developer连接到Oracle数据库。有关说明，请参阅“以HR用户从SQL Developer连接到Oracle数据库”。

•关于查询

查询或SQL SELECT语句从一个或多个表中选择数据

•在SQL Developer中运行查询

本节介绍如何使用Worksheet在SQL Developer中运行查询。

•教程：选择表的所有列

本教程介绍如何选择EMPLOYEES表的所有列。

•教程：选择表的特定列

本教程介绍如何仅选择FIRST\_NAME，LAST\_NAME，EMPLOYEES表的DEPARTMENT\_ID和DEPARTMENT\_ID。

•在新标题下显示选定的列

在显示的查询结果中，默认列标题是列名称。显示新标题下的列，紧跟在之后指定新标题（别名）列名称。别名在查询期间重命名列，但是不会在数据库中更改其名称。

•选择满足指定条件的数据

要仅选择与指定条件匹配的数据，在SELECT语句中包含WHERE子句

•对所选数据进行排序

显示查询结果时，展示的结果可以是任意的顺序，除非你指定要排序

•从多个表中选择数据

要从多个表中选择数据，请使用称为连接的查询。该联接中的表必须至少共享一个列名。

•在查询中使用运算符和函数

查询的select\_list可以包括SQL表达式，可以包括SQL运算符和SQL函数。这些运算符和函数可以具有表数据作为操作数和论点。将评估SQL表达式及其值出现在查询结果中。

**2.6.1关于查询**

查询或SQL SELECT语句从一个或多个表或视图中选择数据。

最简单的查询形式具有以下语法：

SELECT select\_list FROM source\_list

select\_list指定从中选择数据的列，以及source\_list指定具有这些列的表或视图。

嵌套在另一个SQL语句中的查询称为子查询。

在SQL \* Plus环境中，您可以在SQL>提示符之后输入查询（或任何其他SQL语句）

在SQL Developer环境中，您可以输入查询（或任何其他SQL）

注意：

显示查询结果时，记录可以按任何顺序排列，除非您使用ORDER BY子句指定其顺序。

有关更多信息，请参阅“对所选数据进行排序”。

参见：

•Oracle数据库SQL语言参考有关的更多信息

查询和子查询

•Oracle数据库SQL语言参考有关的更多信息

SELECT语句

•SQL \* Plus用户指南和参考有关的更多信息

SQL \* Plus命令行界面

•Oracle SQL Developer用户指南，了解有关使用的信息

SQL Developer中的工作表

2.6.2在SQL Developer中运行查询

本节介绍如何使用Worksheet在SQL Developer中运行查询。

注意：

工作表不仅限于查询;您可以使用它来运行任何SQL语句。

在SQL Developer中运行查询：

1.如果SQL Developer的右框显示hr\_conn窗格：

a.如果未显示Worksheet子窗格，请单击选项卡工作表。

b.转到第4步。

2.单击图标SQL Worksheet。

3.如果“选择连接”窗口打开：

a.如果“连接”字段没有值hr\_conn，请从菜单中选择该值

Sle3b.单击确定。

将出现一个窗格，其中包含标记为hr\_conn的选项卡和两个子窗格，Worksheet和查询生成器。在工作表中，您可以输入SQL语句。

4.在工作表中，键入查询（SELECT语句）。

5.单击图标Run Statement。

查询运行。在工作表下，将显示“查询结果”窗格，显示查询结果。

6.在hr\_conn选项卡下，单击图标Clear。

查询消失，您可以在中输入另一个SQL语句

工作表。当您运行另一个SQL语句时，其结果将显示在查询中

结果窗格，替换以前运行的SQL语句的结果。

参见：

有关使用工作表的信息，请参见“Oracle SQL Developer用户指南”

在SQL Developer中

2.6.3教程：选择表的所有列

本教程介绍如何选择EMPLOYEES表的所有列。

要选择EMPLOYEES表的所有列：

1.如果存在带有选项卡hr\_conn的窗格，请选择它。否则，请单击图标

SQL Worksheet，如“在SQL Developer中运行查询”。

2.在工作表中，输入以下查询：

SELECT \* FROM EMPLOYEES;

3.单击“运行语句”图标。

查询运行。在工作表下，将显示“查询结果”窗格，显示

EMPLOYEES表的所有列。

警告：

在包含存储列的表上使用SELECT \*时要非常小心敏感数据，如密码或信用卡信息。

参见：

“教程：使用SQL Developer查看EMPLOYEES表属性和数据”获取有关使用SQL查看表数据的另一种方法的信息。

**2.6.4教程：选择表的特定列**

本教程介绍了如何仅选择EMPLOYEES表的FIRST\_NAME，LAST\_NAME和DEPARTMENT\_ID列。

要仅选择FIRST\_NAME，LAST\_NAME和DEPARTMENT\_ID：

1.如果存在带有选项卡hr\_conn的窗格，请选择它。 否则，请单击图标SQL Worksheet，如“在SQL Developer中运行查询”。

2.如果“Worksheet”窗格包含查询，请单击“清除”图标清除查询。

3.在工作表中，输入以下查询：

SELECT FIRST\_NAME, LAST\_NAME, DEPARTMENT\_ID FROM EMPLOYEES;

单击图标Run Statement。

查询运行。 在工作表下，将显示“查询结果”窗格，显示

查询的结果，类似于：

FIRST\_NAME LAST\_NAME DEPARTMENT\_ID  
-------------------- ------------------------- -------------  
Donald OConnell 50  
Douglas Grant 50  
Jennifer Whalen 10  
Michael Hartstein 20  
Pat Fay 20  
Susan Mavris 40  
Hermann Baer 70  
Shelley Higgins 110  
William Gietz 110  
Steven King 90  
Neena Kochhar 90  
FIRST\_NAME LAST\_NAME DEPARTMENT\_ID  
-------------------- ------------------------- -------------  
Lex De Haan 90  
...  
Kevin Feeney 50  
107 rows selected.

**2.6.5在新标题下显示所选列**

在显示的查询结果中，默认列标题是列名称。要显示一个在新标题下的列，在紧接着之后指定新列名（别名）。 别名在查询期间重命名列，但不会在数据库中更改其名称。

例2-5中的查询选择与例子中的查询相同的列：

选择表的特定列，但它为它们指定了别名。因为别名不包含在双引号中，所以它们显示为大写字母。

如果将列别名括在双引号中，则保留大小写，并且别名可以包含空格，如例2-6所示。

参见：

有关的更多信息，请参阅Oracle数据库SQL语言参考

SELECT语句，包括列别名（c\_alias）

示例2-5在新标题下显示所选列

SELECT FIRST\_NAME **First**, LAST\_NAME **last**, DEPARTMENT\_ID **DepT**FROM EMPLOYEES;

结果类似于：

FIRST LAST DEPT  
-------------------- ------------------------- ----------  
Donald OConnell 50  
Douglas Grant 50  
Jennifer Whalen 10  
Michael Hartstein 20  
Pat Fay 20  
Susan Mavris 40  
Hermann Baer 70  
Shelley Higgins 110  
William Gietz 110  
Steven King 90  
Neena Kochhar 90  
FIRST LAST DEPT  
-------------------- ------------------------- ----------  
Lex De Haan 90  
...  
Kevin Feeney 50  
107 rows selected

例2-6在列别名中保留大小写和包含空格

SELECT FIRST\_NAME **"Given Name"**, LAST\_NAME **"Family Name"**FROM EMPLOYEES;  
结果类似于：

Given Name Family Name  
-------------------- -------------------------  
Donald OConnell  
Douglas Grant  
Jennifer Whalen  
Michael Hartstein  
Pat Fay  
Susan Mavris  
Hermann Baer  
Shelley Higgins  
William Gietz  
Steven King  
Neena Kochhar  
Given Name Family Name  
-------------------- -------------------------  
Lex De Haan  
...  
Kevin Feeney  
107 rows selected.

**2.6.6选择满足特定条件的数据**

要仅选择与指定条件匹配的数据，请在其中包含WHERE子句SELECT语句。WHERE子句中的条件可以是任何SQL条件（有关SQL条件的信息，请参阅Oracle数据库SQL语言参考）。

例2-7中的查询仅为90部门中的员工选择数据。

仅要为部门100,110和120中的员工选择数据，请使用此WHERE

条款：WHERE DEPARTMENT\_ID IN (100, 110, 120);

例2-8中的查询姓氏开头仅为“ma”的员工数据。

仅要查询为姓氏包含“ma”的员工的数据，请使用此WHERE条件：

WHERE LAST\_NAME LIKE '%ma%';

例2-9中的查询测试了两个条件 - 工资是否至少11000，以及佣金百分比是否为空。

参见：

•Oracle数据库SQL语言参考有关的更多信息

SELECT语句，包括WHERE子句

•Oracle数据库SQL语言参考有关的更多信息

SQL条件

示例2-7从一个部门选择数据

SELECT FIRST\_NAME，LAST\_NAME，DEPARTMENT\_ID FROM employee WHERE DEPARTMENT\_ID = 90;

结果类似于：

FIRST\_NAME LAST\_NAME DEPARTMENT\_ID

FIRST\_NAME LAST\_NAME DEPARTMENT\_ID  
-------------------- ------------------------- -------------  
Steven King 90  
Neena Kochhar 90  
Lex De Haan 90  
3 rows selected.

示例2-8选择以相同子字符串开头的姓氏的数据

SELECT FIRST\_NAME, LAST\_NAME

FROM EMPLOYEES

WHERE LAST\_NAME LIKE 'Ma%';

结果类似于：

FIRST\_NAME LAST\_NAME  
-------------------- -------------------------  
Jason Mallin  
Steven Markle  
James Marlow  
Mattea Marvins  
Randall Matos  
Susan Mavris  
6 rows selected.

例2-9选择满足两个条件的数据

SELECT FIRST\_NAME, LAST\_NAME, SALARY, COMMISSION\_PCT "%"  
FROM EMPLOYEES  
**WHERE (SALARY >= 11000) AND (COMMISSION\_PCT IS NOT NULL)**;

结果类似于：

SELECT FIRST\_NAME, LAST\_NAME, SALARY, COMMISSION\_PCT "%"  
FROM EMPLOYEES  
**WHERE (SALARY >= 11000) AND (COMMISSION\_PCT IS NOT NULL)**;

结果类似于：

FIRST\_NAME LAST\_NAME SALARY %  
-------------------- ------------------------- ---------- ----------  
John Russell 14000 .4  
Karen Partners 13500 .3  
Alberto Errazuriz 12000 .3  
Gerald Cambrault 11000 .3  
Lisa Ozer 11500 .25  
Ellen Abel 11000 .3  
6 rows selected.

**2.6.7对数据进行排序**

显示查询结果时，记录可以按任何顺序排列，除非您使用ORDER BY子句指定它们的排列顺序。

示例2-10中的查询结果按LAST\_NAME按升序排序（默认）。

或者，在SQL Developer中，可以省略ORDER BY子句并双击要排序的列的名称。

排序标准不必包含在选择列表中，如例2-11所示。

参见：

有关的更多信息，请参阅Oracle数据库SQL语言参考

SELECT语句，包括ORDER BY子句

示例2-10按LAST\_NAME对所选数据进行排序

SELECT FIRST\_NAME, LAST\_NAME, HIRE\_DATE  
FROM EMPLOYEES  
**ORDER BY LAST\_NAME**;

结果：

FIRST\_NAME LAST\_NAME HIRE\_DATE  
-------------------- ------------------------- ---------  
Ellen Abel 11-MAY-04  
Sundar Ande 24-MAR-08  
Mozhe Atkinson 30-OCT-05  
David Austin 25-JUN-05  
Hermann Baer 07-JUN-02  
Shelli Baida 24-DEC-05  
Amit Banda 21-APR-08  
Elizabeth Bates 24-MAR-07  
...  
FIRST\_NAME LAST\_NAME HIRE\_DATE  
-------------------- ------------------------- ---------  
Jose Manuel Urman 07-MAR-06  
Peter Vargas 09-JUL-06  
Clara Vishney 11-NOV-05  
Shanta Vollman 10-OCT-05  
Alana Walsh 24-APR-06  
Matthew Weiss 18-JUL-04  
Jennifer Whalen 17-SEP-03  
Eleni Zlotkey 29-JAN-08  
107 rows selected

示例2-11按未选择的列对选定数据进行排序

**SELECT FIRST\_NAME, HIRE\_DATE**FROM EMPLOYEES  
**ORDER BY LAST\_NAME**;

结果：

FIRST\_NAME HIRE\_DATE  
-------------------- ---------  
Ellen 11-MAY-04  
Sundar 24-MAR-08  
Mozhe 30-OCT-05  
David 25-JUN-05  
Hermann 07-JUN-02  
Shelli 24-DEC-05  
Amit 21-APR-08  
Elizabeth 24-MAR-07  
...  
FIRST\_NAME HIRE\_DATE  
-------------------- ---------  
Jose Manuel 07-MAR-06  
Peter 09-JUL-06  
Clara 11-NOV-05  
Shanta 10-OCT-05  
Alana 24-APR-06  
Matthew 18-JUL-04  
Jennifer 17-SEP-03  
Eleni 29-JAN-08  
107 rows selected.

2.6.8从多个表中选择数据

要从多个表中选择数据，请使用称为连接查询。表的连接中必须至少共享一个列名。

假设您要选择每位员工的FIRST\_NAME，LAST\_NAME和DEPARTMENT\_NAME，FIRST\_NAME和LAST\_NAME在EMPLOYEES表，DEPARTMENT\_NAME位于DEPARTMENTS表中，两个表都有DEPARTMENT\_ID。 您可以使用示例2-12中的查询。

对于仅出现在的一个表中的列名，表名限定符是可选的连接，但两个表中出现的列名称都是必需的。以下查询等效于例2-12中的查询：

SELECT **FIRST\_NAME** "First",  
**LAST\_NAME** "Last",  
**DEPARTMENT\_NAME** "Dept. Name"  
FROM EMPLOYEES, DEPARTMENTS  
**WHERE EMPLOYEES.DEPARTMENT\_ID = DEPARTMENTS.DEPARTMENT\_ID**ORDER BY DEPARTMENT\_NAME, LAST\_NAME;

虽然您在FROM子句中创建别名，但您可以在之前使用它们

查询，如下例所示：

SELECT **e.FIRST\_NAME** "First",  
**e.LAST\_NAME** "Last",  
**d.DEPARTMENT\_NAME** "Dept. Name"  
FROM EMPLOYEES e, DEPARTMENTS d  
WHERE e.DEPARTMENT\_ID = d.DEPARTMENT\_ID  
ORDER BY d.DEPARTMENT\_NAME, e.LAST\_NAME;

参见：

有关联接的详细信息，请参见Oracle数据库SQL语言参考

例2-12从两个表中选择数据（连接两个表）

SELECT EMPLOYEES.FIRST\_NAME "First",  
EMPLOYEES.LAST\_NAME "Last",  
DEPARTMENTS.DEPARTMENT\_NAME "Dept. Name"  
FROM EMPLOYEES, DEPARTMENTS  
**WHERE EMPLOYEES.DEPARTMENT\_ID = DEPARTMENTS.DEPARTMENT\_ID**ORDER BY DEPARTMENTS.DEPARTMENT\_NAME, EMPLOYEES.LAST\_NAME;  
Result:  
 First Last Dept Name  
----------- --------------- ---------------------

William Gietz Accounting  
Shelley Higgins Accounting  
Jennifer Whalen Administration  
Lex De Haan Executive  
Steven King Executive  
Neena Kochhar Executive  
John Chen Finance  
...  
Jose Manuel Urman Finance  
Susan Mavris Human Resources  
David Austin IT  
...  
Valli Pataballa IT  
Pat Fay Marketing  
Michael Hartstein Marketing  
Hermann Baer Public Relations  
Shelli Baida Purchasing  
...  
Sigal Tobias Purchasing  
Ellen Abel Sales  
...  
Eleni Zlotkey Sales  
Mozhe Atkinson Shipping  
...  
Matthew Weiss Shipping  
106 rows selected

**2.6.9在查询中使用运算符和函数**

查询的select\_list可以包括SQL表达式，可以包括SQL运算符和SQL函数。这些运算符和函数可以具有表数据，操作数和参数。将SQL表达式及其值展示在查询结果中。

•在查询中使用算术运算符

基本算术运算符 - +（加法）， - （减法），\*（乘法），

和/（除法） - 对列值进行操作。

•在查询中使用数字函数

数字函数接受数字输入并返回数值。每个数字

函数为每个计算的行返回一个值。

•在查询中使用连接运算符

连接运算符（||）将两个字符串组合成一个字符串，将第二个字符串附加到第一个字符串。例如，'a'||'b'='ab'。您可以使用此运算符用于组合来自两列或表达式查询结果的列的信息。

•在查询中使用字符函数

字符函数接受字符输入。大多数返回字符值，但是一些返回数值。对于每一行，每个字符函数都返回一个值。

•在查询中使用日期函数

日期函数用于操作DATA，time stamp和interval上。日期函数为每行返回一个值。

•在查询中使用转换函数

转换函数将一种数据类型转换为另一种。

•在查询中使用聚合函数

聚合函数采用一组行并返回单个结果行。该行可以是整个表或视图。

•在查询中使用NULL相关函数

与NULL相关的函数有助于处理NULL值。

•在查询中使用CASE表达式

CASE表达式允许您在SQL语句中使用IF ... THEN ... ELSE逻辑

而不调用子程序。有两种CASE表达式，simple和search。

•在查询中使用DECODE函数

DECODE函数将表达式与多个搜索值进行比较。只要表达式的值与搜索值匹配，DECODE就会返回与该搜索值相关联的结果。如果DECODE找不到匹配，那么它返回默认值（如果指定）或NULL（如果未指定默认值）。

参见：

•Oracle数据库SQL语言参考有关的更多信息

SQL运算符

•Oracle数据库SQL语言参考有关的更多信息

SQL函数

**2.6.9.1在查询中使用算术运算符**

基本算术运算符 - +（加法）， - （减法），\*（乘法）和/

（除法） - 对列值进行操作。

示例2-13中的查询显示LAST\_NAME，SALARY（月工资）和90部门中每位员工的年薪，并按照SALARY进行降序排列。

例2-13在查询中使用算术表达式

SELECT LAST\_NAME,  
SALARY "Monthly Pay",  
**SALARY \* 12** "Annual Pay"  
FROM EMPLOYEES  
WHERE DEPARTMENT\_ID = 90  
ORDER BY SALARY DESC;  
结果：  
LAST\_NAME Monthly Pay Annual Pay  
------------------------- ----------- ----------  
King 24000 288000  
De Haan 17000 204000  
Kochhar 17000 204000

2.6.9.2在查询中使用数字函数

数字函数接受数字输入并返回数值。 每个数字函数为每个计算的行返回一个值。

SQL数据库中列出并描述了SQL支持的数字函数

SQL语言参考。

例2-14中的查询使用数字函数ROUND来显示部门100中的每个员工的每日工资，并且四舍五入到最接近的分数。

例2-15中的查询使用数字函数TRUNC来显示部门100中的每个雇员的每日工资，截断到最接近的美元。

参见：

有关SQL的更多信息，请参见Oracle数据库SQL语言参考

数字函数

例2-14舍入数字数据

SELECT LAST\_NAME,  
**ROUND (((SALARY \* 12)/365), 2)** "Daily Pay"  
FROM EMPLOYEES  
WHERE DEPARTMENT\_ID = 100  
ORDER BY LAST\_NAME;

结果：

LAST\_NAME Daily Pay  
------------------------- ----------

Chen 269.59  
Faviet 295.89  
Greenberg 394.52  
Popp 226.85  
Sciarra 253.15  
Urman 256.44  
6 rows selected.

例2-15截断数值数据

SELECT LAST\_NAME,  
**TRUNC ((SALARY \* 12)/365)** "Daily Pay"  
FROM EMPLOYEES  
WHERE DEPARTMENT\_ID = 100  
ORDER BY LAST\_NAME;  
结果：  
LAST\_NAME Daily Pay  
------------------------- ----------  
Chen 269  
Faviet 295  
Greenberg 394  
Popp 226  
Sciarra 253  
Urman 256  
6 rows selected.

**2.6.9.3在查询中使用连接运算符**

连接运算符（||）通过追加将两个字符串组合成一个字符串

第二个字符串到第一个。 例如，'a'||'b'='ab'。 您可以使用此运算符组合查询的同一列中的两列或表达式的结果。

例2-16中的查询连接了每个选定的员工的名字，空格和姓氏

参见：

有关的更多信息，请参阅Oracle数据库SQL语言参考

连接运算符

例2-16连接字符数据

SELECT **FIRST\_NAME || ' ' || LAST\_NAME** "Name"  
FROM EMPLOYEES  
WHERE DEPARTMENT\_ID = 100  
ORDER BY LAST\_NAME;  
Result:  
Name  
----------------------------------------------  
John Chen

Daniel Faviet  
Nancy Greenberg  
Luis Popp  
Ismael Sciarra  
Jose Manuel Urman  
6 rows selected.

**2.6.9.4在查询中使用字符函数**

字符函数接受字符输入。大多数返回字符值，但有些函数返回数值。每个字符函数为每行返回一个值。

SQL支持的字符函数在Oracle Database SQL Language列出和描述。

UPPER，INITCAP和LOWER函数分别以大写，每个单词首字母大写，其余字母小写和小写显示其字符参数。

示例2-17中的查询以大写形式显示LAST\_NAME，其中FIRST\_NAME第一个字符为大写，所有其他字符为小写，EMAIL为小写。

参见：

有关SQL的更多信息，请参见Oracle数据库SQL语言参考

角色功能

例2-17更改字符数据的大小写

SELECT **UPPER(LAST\_NAME)** "Last",  
**INITCAP(FIRST\_NAME)** "First",  
**LOWER(EMAIL)** "E-Mail"  
FROM EMPLOYEES  
WHERE DEPARTMENT\_ID = 100  
ORDER BY EMAIL;  
结果  
 Last First E-Mail  
---------- ----------------- ---------------------  
FAVIET Daniel dfaviet  
SCIARRA Ismael isciarra  
CHEN John jchen  
URMAN Jose Manuel jmurman  
POPP Luis lpopp  
GREENBERG Nancy ngreenbe  
6 rows selected.

**2.6.9.5在查询中使用日期时间函数**

日期函数用于操作DATA，time stamp和interval上。日期函数为每行返回一个值。

Oracle Database SQL Language Reference 对SQL支持的日期时间函数进行了罗列及描述。

对于每个DATE和时间戳值，Oracle数据库存储以下信息：

• 年

•月份

•日期

•小时

•分钟

•秒

对于每个时间戳值，Oracle数据库还存储了部分秒，你可以指定其精度。要同时存储时区，请使用TIMESTAMP WITH TIME ZONE或TIMESTAMP WITH LOCAL TIME。

有关DATE数据类型的更多信息，请参阅Oracle Database SQL Language Reference。

有关TIMESTAMP数据类型的更多信息，请参阅Oracle Database SQL Language Reference。

有关其他时间戳数据类型和间隔数据类型的信息，请参阅

Oracle Database SQL Language Reference。

例2-18中的查询使用EXTRACT和SYSDATE函数来显示多年来100部门一共雇佣了多少员工。SYSDATE function将系统时钟的当前日期作为DATE值返回。

更多有关SYSDATE函数的信息，请参阅Oracle Database SQL Language Reference

有关EXTRACT函数的信息，Oracle Database SQL Language Reference

例2-19中的查询使用SYSTIMESTAMP函数显示当前系统日期和时间。 SYSTIMESTAMP函数返回TIMESTAMP值。

有关SYSTIMESTAMP函数的信息，请参阅Oracle Database SQL Language Reference

查询中的FROM子句中的DUAL表是Oracle的单行表，是与数据库和数据字典一起自动创建的。当您想使用SELECT语句计算常量表达式时从DUAL中选择。

因为DUAL只有一行，所以常量只返回一次。

更多有关从DUAL中选择的信息，请参阅Oracle Database SQL Language Reference

参见：

有关SQL的更多信息，请参阅Oracle Database SQL Language Reference

日期时间功能

示例2-18显示日期之间的年数

SELECT LAST\_NAME,  
**(EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM HIRE\_DATE))** "Years Employed"  
FROM EMPLOYEES  
WHERE DEPARTMENT\_ID = 100  
ORDER BY "Years Employed";  
结果  
LAST\_NAME Years Employed  
------------------------- --------------  
Popp 5  
Urman 6  
Chen 7  
Sciarra 7  
Greenberg 10  
Faviet 10  
6 rows selected.

例2-19显示系统日期和时间

SELECT EXTRACT(HOUR FROM SYSTIMESTAMP) || ':' ||  
EXTRACT(MINUTE FROM SYSTIMESTAMP) || ':' ||  
ROUND(EXTRACT(SECOND FROM SYSTIMESTAMP), 0) || ', ' ||  
EXTRACT(MONTH FROM SYSTIMESTAMP) || '/' ||  
EXTRACT(DAY FROM SYSTIMESTAMP) || '/' ||  
EXTRACT(YEAR FROM SYSTIMESTAMP) "System Time and Date"  
FROM DUAL;

结果取决于当前的SYSTIMESTAMP值，但具有以下格式：

System Time and Date  
-------------------------------------------------------------------  
18:17:53, 12/27/2012

2.6.9.6在查询中使用转换函数

转换函数将一种数据类型转换为另一种。

SQL支持的转换函数在Oracle中列出和描述

数据库SQL语言参考。

例2-20中的查询使用TO\_CHAR函数转换HIRE\_DATE值（具有DATE类型）到具有FMMonth DD格式的字符值YYYY。 FM从月份名称中删除前导和尾随空白。 FMMonth DD YYYY是日期时间格式模型的示例。 有关日期时间格式的信息模型，请参阅Oracle数据库SQL语言参考。

例2-21中的查询使用TO\_NUMBER函数进行转换

POSTAL\_CODE值（类型为VARCHAR2）到NUMBER类型的值，

它在计算中使用。

参见：

•Oracle数据库SQL语言参考有关的更多信息

SQL转换函数

•“关于NLS\_DATE\_FORMAT参数”

示例2-20使用格式模板将日期转换为字符

SELECT LAST\_NAME,  
HIRE\_DATE,  
**TO\_CHAR(HIRE\_DATE, 'FMMonth DD YYYY')** "Date Started"  
FROM EMPLOYEES  
WHERE DEPARTMENT\_ID = 100  
ORDER BY LAST\_NAME;  
Result:  
LAST\_NAME HIRE\_DATE Date Started  
------------------------- --------- -----------------  
Chen 28-SEP-05 September 28 2005  
Faviet 16-AUG-02 August 16 2002  
Greenberg 17-AUG-02 August 17 2002  
Popp 07-DEC-07 December 7 2007  
Sciarra 30-SEP-05 September 30 2005  
Urman 07-MAR-06 March 7 2006  
6 rows selected.

示例2-21将字符转换为数字

SELECT CITY,  
POSTAL\_CODE "Old Code",  
**TO\_NUMBER(POSTAL\_CODE) + 1** "New Code"  
FROM LOCATIONS  
WHERE COUNTRY\_ID = 'US'  
ORDER BY POSTAL\_CODE;  
结果：  
CITY Old Code New Code  
------------------------------ ------------ ----------  
Southlake 26192 26193  
South Brunswick 50090 50091  
Seattle 98199 98200  
South San Francisco 99236 99237  
4 rows selected.

2.6.9.7在查询中使用聚合函数

聚合函数采用一组行并返回单个结果行。该行组可以是整个表或视图。

SQL支持的聚合函数在Oracle Database SQL Language Reference

中列出和描述。

与GROUP BY子句一起使用时，聚合函数特别强大，它按一个或多个列对查询结果进行分组，并显示每个组的结果。

例2-22中的查询使用COUNT函数和GROUP BY子句

显示有多少人向每位经理报告。 通配符\*表示整个记录。

示例2-22显示一名员工未向经理报告。

下列query选择该员工的名字，姓氏和职位：

COLUMN FIRST\_NAME FORMAT A10;  
COLUMN LAST\_NAME FORMAT A10;  
COLUMN JOB\_TITLE FORMAT A10;  
SELECT e.FIRST\_NAME,  
e.LAST\_NAME,  
j.JOB\_TITLE  
FROM EMPLOYEES e, JOBS j  
WHERE e.JOB\_ID = j.JOB\_ID  
AND **MANAGER\_ID IS NULL**;  
结果：  
FIRST\_NAME LAST\_NAME JOB\_TITLE  
---------- ---------- ----------  
Steven King President

要使查询仅返回聚合值满足指定条件的行，在聚合函数中使用HAVING子句。

例2-23中的查询显示了每个部门每年花费的工资金额，但仅适用于金额超过1,000,000美元的部门。

例2-24中的查询使用几个聚合函数来显示统计信息每个JOB\_ID的工资。

参见：

有关SQL的更多信息，请参见Oracle Database SQL Language Reference

集合函数

示例2-22计算每个组中的行数

SELECT MANAGER\_ID "Manager",  
**COUNT(\*)** "Number of Reports"  
FROM EMPLOYEES  
**GROUP BY MANAGER\_ID**ORDER BY MANAGER\_ID;

结果：

Manager Number of Reports  
---------- -----------------  
100 14  
101 5  
102 1  
103 4  
108 5  
114 5  
120 8  
121 8  
122 8  
123 8  
124 8  
145 6  
146 6  
147 6  
148 6  
149 6  
201 1  
205 1  
1  
19 rows selected.

示例2-23将聚合函数限制为满足条件的行

SELECT DEPARTMENT\_ID "Department",  
SUM(SALARY\*12) "All Salaries"  
FROM EMPLOYEES  
HAVING SUM(SALARY \* 12) >= 1000000GROUP BY DEPARTMENT\_ID;  
Result:  
Department All Salaries  
---------- ------------  
50 1876800  
80 3654000

示例2-24使用统计信息的聚合函数

SELECT JOB\_ID,  
COUNT(\*) "#",  
MIN(SALARY) "Minimum",  
ROUND(AVG(SALARY), 0) "Average",  
MEDIAN(SALARY) "Median",  
MAX(SALARY) "Maximum",  
ROUND(STDDEV(SALARY)) "Std Dev"  
FROM EMPLOYEES  
GROUP BY JOB\_ID  
ORDER BY JOB\_ID;  
Result:  
JOB\_ID # Minimum Average Median Maximum Std Dev  
---------- ---------- ---------- ---------- ---------- ---------- ----------  
AC\_ACCOUNT 1 8300 8300 8300 8300 0  
AC\_MGR 1 12008 12008 12008 12008 0  
AD\_ASST 1 4400 4400 4400 4400 0  
AD\_PRES 1 24000 24000 24000 24000 0

AD\_VP 2 17000 17000 17000 17000 0  
FI\_ACCOUNT 5 6900 7920 7800 9000 766  
FI\_MGR 1 12008 12008 12008 12008 0  
HR\_REP 1 6500 6500 6500 6500 0  
IT\_PROG 5 4200 5760 4800 9000 1926  
MK\_MAN 1 13000 13000 13000 13000 0  
MK\_REP 1 6000 6000 6000 6000 0  
PR\_REP 1 10000 10000 10000 10000 0  
PU\_CLERK 5 2500 2780 2800 3100 239  
PU\_MAN 1 11000 11000 11000 11000 0  
SA\_MAN 5 10500 12200 12000 14000 1525  
SA\_REP 30 6100 8350 8200 11500 1524  
SH\_CLERK 20 2500 3215 3100 4200 548  
ST\_CLERK 20 2100 2785 2700 3600 453  
ST\_MAN 5 5800 7280 7900 8200 1066  
19 rows selected.

2.6.9.8在查询中使用NULL相关函数

与NULL相关的函数有助于处理NULL值。

SQL支持的与NULL相关的函数在Oracle Database SQL Language Reference中列出并描述。

例2-25中的查询返回员工的姓氏和佣金

其姓氏以'B'开头。 如果员工没有收到任何佣金（即，如果

COMMISSION\_PCT为NULL），NVL函数将“Not Applicable”替换为NULL。

例2-26中的查询返回的姓氏，工资和收入，姓氏以'B'开头的员工，使用NVL2功能：如果COMMISSION\_PCT不是NULL，收入是工资加佣金; 如果COMMISSION\_PCT为NULL，收入只含有工资。

参见：

•Oracle Database SQL Language Reference参考有关的更多信息

NVL功能

•Oracle Database SQL Language Reference参考有关的更多信息

NVL2功能

示例2-25将字符串替换为NULL值

SELECT LAST\_NAME,  
NVL(TO\_CHAR(COMMISSION\_PCT), 'Not Applicable')"COMMISSION"  
FROM EMPLOYEES  
WHERE LAST\_NAME LIKE 'B%'  
ORDER BY LAST\_NAME;  
Result:  
LAST\_NAME COMMISSION  
--------------- ----------------------------------------  
Baer Not Applicable  
Baida Not Applicable  
Banda . 1

Bates .15  
Bell Not Applicable  
Bernstein .25  
Bissot Not Applicable  
Bloom .2  
Bull Not Applicable  
9 rows selected.

Example 2-26 Specifying Different Expressions for NULL and Not NULL Values

SELECT LAST\_NAME, SALARY,  
NVL2(COMMISSION\_PCT, SALARY + (SALARY \* COMMISSION\_PCT), SALARY) INCOMEFROM EMPLOYEES WHERE LAST\_NAME LIKE 'B%'  
ORDER BY LAST\_NAME;  
Result:  
LAST\_NAME SALARY INCOME  
------------------------- ---------- ----------  
Baer 10000 10000  
Baida 2900 2900  
Banda 6200 6820  
Bates 7300 8395  
Bell 4000 4000  
Bernstein 9500 11875  
Bissot 3300 3300  
Bloom 10000 12000  
Bull 4100 4100  
9 rows selected.

**2.6.9.9在查询中使用CASE表达式**

CASE表达式允许您在SQL语句中使用IF ... THEN ... ELSE逻辑而不用调用子程序。 有两种CASE表达式，simple和searched。

例2-27中的查询使用简单的CASE表达式来显示国家/地区名称和每个国家代码。

例2-28中的查询使用搜索的CASE表达式来显示建议

根据与长度相关的日期范围，加薪（15％，10％，5％或0％）

服务。

参见:• Oracle Database SQL Language Reference for more information about  
CASE expressions  
• Oracle Database PL/SQL Language Reference for more information about  
CASE expressions  
• "Using the DECODE Function in Queries"  
• "Using the CASE Statement"

示例2-27在查询中使用简单CASE表达式

SELECT UNIQUE COUNTRY\_ID ID,

CASE COUNTRY\_ID  
WHEN 'AU' THEN 'Australia'  
WHEN 'BR' THEN 'Brazil'  
WHEN 'CA' THEN 'Canada'  
WHEN 'CH' THEN 'Switzerland'  
WHEN 'CN' THEN 'China'  
WHEN 'DE' THEN 'Germany'  
WHEN 'IN' THEN 'India'  
WHEN 'IT' THEN 'Italy'  
WHEN 'JP' THEN 'Japan'  
WHEN 'MX' THEN 'Mexico'  
WHEN 'NL' THEN 'Netherlands'  
WHEN 'SG' THEN 'Singapore'  
WHEN 'UK' THEN 'United Kingdom'  
WHEN 'US' THEN 'United States'  
ELSE 'Unknown'  
END COUNTRY  
FROM LOCATIONS  
ORDER BY COUNTRY\_ID;  
Result:  
ID COUNTRY  
-- --------------  
AU Australia  
BR Brazil  
CA Canada  
CH Switzerland  
CN China  
DE Germany  
IN India  
IT Italy  
JP Japan  
MX Mexico  
NL Netherlands  
SG Singapore  
UK United Kingdom  
US United States  
14 rows selected

示例2-28在查询中使用搜索的CASE表达式

SELECT LAST\_NAME "Name",  
HIRE\_DATE "Started",  
SALARY "Salary",  
CASE  
WHEN HIRE\_DATE < TO\_DATE('01-Jan-03', 'dd-mon-yy')  
THEN TRUNC(SALARY\*1.15, 0)  
WHEN HIRE\_DATE >= TO\_DATE('01-Jan-03', 'dd-mon-yy') AND  
HIRE\_DATE < TO\_DATE('01-Jan-06', 'dd-mon-yy')  
THEN TRUNC(SALARY\*1.10, 0)  
WHEN HIRE\_DATE >= TO\_DATE('01-Jan-06', 'dd-mon-yy') AND  
HIRE\_DATE < TO\_DATE('01-Jan-07', 'dd-mon-yy')  
THEN TRUNC(SALARY\*1.05, 0)  
ELSE SALARY  
END "Proposed Salary"  
FROM EMPLOYEES

WHERE DEPARTMENT\_ID = 100  
ORDER BY HIRE\_DATE;  
Result:  
Name Started Salary Proposed Salary  
------------------------- --------- ---------- ---------------  
Faviet 16-AUG-02 9000 10350  
Greenberg 17-AUG-02 12008 13809  
Chen 28-SEP-05 8200 9020  
Sciarra 30-SEP-05 7700 8470  
Urman 07-MAR-06 7800 8190  
Popp 07-DEC-07 6900 6900  
6 rows selected.

2.6.9.10在查询中使用DECODE函数

DECODE函数将表达式与多个搜索值进行比较。每当表达式的值与搜索值匹配，DECODE返回结果与该搜索值相关联。如果DECODE找不到匹配项，则返回默认值（如果指定）或NULL（如果未指定默认值）。

例2-29中的查询使用DECODE函数来显示工资

增加三个不同的任务，表达式为JOB\_ID;搜索值是'PU\_CLERK'，'SH\_CLERK'和'ST\_CLERK';默认为SALARY。

注意：

DECODE函数的参数可以是任何SQL数字或字符类型。 Oracle会自动转换表达式和每次搜索比较前第一个搜索值的数据类型的值。自动将返回值转换为与第一个结果相同的数据类型。

如果第一个结果的数据类型为CHAR，或者第一个结果为NULL，那么

Oracle将返回值转换为数据类型VARCHAR2。

参见：

Oracle Database SQL Language Reference，以获取有关的信息

DECODE功能，“在查询中使用CASE表达式”

示例2-29在查询中使用DECODE函数

SELECT LAST\_NAME, JOB\_ID, SALARY,  
DECODE(JOB\_ID,  
'PU\_CLERK', SALARY \* 1.10,  
'SH\_CLERK', SALARY \* 1.15,  
'ST\_CLERK', SALARY \* 1.20,  
SALARY) "Proposed Salary"  
FROM EMPLOYEES  
WHERE JOB\_ID LIKE '%\_CLERK

AND LAST\_NAME < 'E'  
ORDER BY LAST\_NAME;  
Result:  
LAST\_NAME JOB\_ID SALARY Proposed Salary  
------------------------- ---------- ---------- ---------------  
Atkinson ST\_CLERK 2800 3360  
Baida PU\_CLERK 2900 3190  
Bell SH\_CLERK 4000 4600  
Bissot ST\_CLERK 3300 3960  
Bull SH\_CLERK 4100 4715  
Cabrio SH\_CLERK 3000 3450  
Chung SH\_CLERK 3800 4370  
Colmenares PU\_CLERK 2500 2750  
Davies ST\_CLERK 3100 3720  
Dellinger SH\_CLERK 3400 3910  
Dilly SH\_CLERK 3600 4140  
11 rows selected