



恩墨学院
ENMOEDU

Oracle SQL

【SQL 简介】

SQL:结构化查询语言，Structured Query Language

1.SQL 的起源

关系型数据库，是指采用了关系模型来组织数据的数据库。简单来说，关系模型指的就是二维表格模型，而一个关系型数据库就是由二维表及其之间的联系所组成的一个数据组织。

20 世纪 70 年代，Codd 发布了《大型共享数据库的关系模型》，Codd 的模型被公认为关系型数据库管理系统（REBMS）的确定性模型。IBM 基于这个模式开发了 Structured English Query Language (SEQUEL)，后来成为了 SQL (Structured Query Language)。

1979:Oracle 发布了第一个商业版使用的 SQL。

2.SQL 的作用

查询数据

插入、更新、删除表中的数据。

创建、修改、删除对象。

控制对数据库及对象的访问。

SQL 用一种统一的语言了完成了所有上述任务。Oracle SQL 是 ANSI 标准的一种实现。Oracle SQL 支持许多标准 SQL 之外的扩展功能。

【环境准备】

【启动数据库实例】

使用 oracle 用户登录 linux 操作系统

查看环境变量

```
[oracle@enmoedu1 ~]$ echo $ORACLE_SID
```

```
[oracle@enmoedu1 ~]$ echo $ORACLE_SID  
PROD
```

使用 sqlplus 登录连接数据库

```
[oracle@enmoedu1 ~]$ sqlplus / as sysdba
```

```
[oracle@enmoedu1 ~]$ sqlplus / as sysdba  
SQL*Plus: Release 12.2.0.1.0 Production on Thu Jun 15 12:05:15 2017  
Copyright (c) 1982, 2016, Oracle. All rights reserved.  
Connected to an idle instance.
```

"idle instance" 说明数据库实例没有启动

startup 启动数据库实例

```
SYS@PROD> startup
```

```
SYS@PROD> startup  
ORACLE instance started.  
  
Total System Global Area 838860800 bytes  
Fixed Size 8626240 bytes  
Variable Size 633343936 bytes  
Database Buffers 188743680 bytes  
Redo Buffers 8146944 bytes  
Database mounted.  
Database opened.  
SYS@PROD>
```

查看数据库实例状态

```
SYS@PROD> select status from v$instance;
```

```
SYS@PROD> select status from v$instance;  
  
STATUS  
-----  
OPEN
```

关闭 linux 操作系统前，需正常关闭数据库实例！

需 sys 用户执行 **shutdown immediate** 命令

【SCOTT 用户方案】

从 Oracle 第一个商业化版本至今，oracle 的默认数据库里都少不了这个名字为 scott，密码为 tiger 的用户。

这要追溯到 Oracle 的创业阶段了，1977 年 6 月，Larry Ellison（拉里·埃里森）、Bob Miner 和 Ed Oates 在硅谷共同创办了一家名为软件开发实验室（Software Development Laboratories, SDL）的计算机公司，这个只有三个人的公司就是 ORACLE 公司的前身。当时埃里森合同的关系还在自己原来的公司，总要有写代码的人呀，公司的第一个程序员 Bruce Scott 出现了，他的猫的名字叫 tiger。

【重建 scott 用户方案方法】

如需重建 scott 用户方案，需退出 scott 用户，使用 sys 用户执行如下脚本

```
SYS@PROD>@?/rdbms/admin/utlsamp1.sql
```

如执行脚本后 scott 用户没有重建，可以重启数据库实例后再次执行。

【解锁用户、修改密码方法】

如果 scott 用户方案已安装但未解锁或未设置密码

```
[oracle@enmoedu1 ~]$ sqlplus / as sysdba
SYS@PROD>alter user scott identified by tiger account unlock;
SYS@PROD>conn scott/tiger
```


【设置 sqlplus 命令提示符 glogin.sql】

```
[oracle@enmoedu1 ~]$ vi $ORACLE_HOME/sqlplus/admin/glogin.sql
set linesize 120
set pagesize 999
define _editor=vi
set sqlprompt "_user'@'_connect_identifier> "
```

1 基本 SELECT 语句



1.1 【学习目标】

 云和恩墨
ENMOTECH

课程目标

学完本课后，应能完成下列工作：

- 列举 SQL SELECT 语句的功能
- 执行基本 SELECT 语句

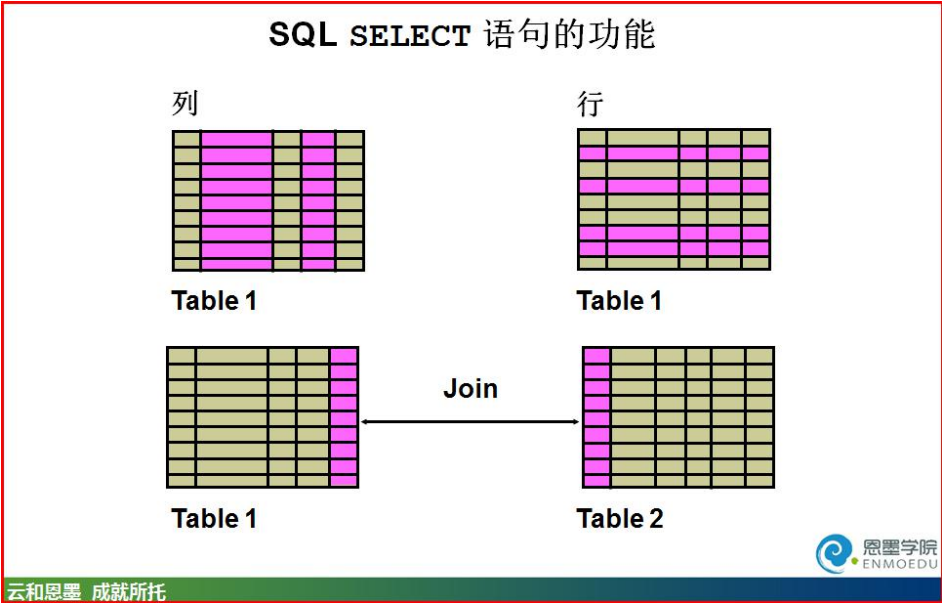
主要知识点：

- 1.select 语句的基本功能
- 2.数学表达式在 select 语句中的应用
- 3.select 语句中 null 值的处理
- 4.定义列的别名、使用连接运算符
- 5.在 select 语句中消除重复行

1.2 【select 语句基本功能】

select 语句可以从数据库表中检索信息。

选择、投影、连接



选择：查询符合条件的行，过滤掉不符合条件的行

投影：查询符合条件的列，不显示不符合条件的列

连接：从多个表中查询数据。

连接是关系型数据库的核心，把数据放在不同的表里，通过关联获得完整信息。

表是由行和列组成的，列构成表的结构，行组成表的数据。

select 语句并不改变数据库中的数据，只是将数据从数据库中提取显示出来。

1.3 【select 语句基本语法】

云和恩墨
ENMOTech

基本 SELECT 语句

```
SELECT *|{ [DISTINCT] column|expression [alias],...}  
FROM      table;
```

- SELECT 标识选择哪些列。
- FROM 标识从哪个表中选择。

[] 代表可选； * 代表所有列

语句：一个语句的结束以；为标志。

子句：一个语句可以由多个子句组成

最基本的 select 语句包含两个子句：select 子句和 from 子句。

1.4 【SQL 语句的书写规范】



编写 SQL 语句规范

- SQL 语言大小写不敏感。
- SQL 可以写在一行或者多行
- 关键字不能被缩写也不能分行
- 各子句一般要分行写。
- 使用缩进提高语句的可读性。
- SQL 语句在 SQL Developer 中，可以以一个分号(;)终止，当执行多个 SQL 语句，结束分号是必需的。
- 在 SQL * Plus 中，用分号(;)结束每条 SQL 语句。

在 SQL*Plus 中，在 SQL 提示符下输入 SQL 语句，从第二行起有行号。
通常用 **分号** 作为 SQL 语句的终止符。

1.5 【选择所有列】



选择全部列

```
SELECT *  
FROM dept;
```

* 代表所有列

写*比较方便，但是要注意：

1. 如果不需要查询所有列，用*会把所有数据都查出来，增加 oracle 处理的 **代价**，**性能** 就会降低。
2. 写*时，oracle 需要查询数据字典，转换为具体列名称。
3. sql 开发建议写具体字段名。

1.5.1 【熟悉 scott 用户方案】

用 scott 用户登录数据库

user_tables 存储的是当前用户所有表的信息——数据字典表（视图）

```
select table_name from user_tables;
```

```
SCOTT@PROD> select table_name from user_tables;

TABLE_NAME
-----
DEPT
EMP
BONUS
SALGRADE
PLAN_TABLE
EMP2
EMP1

7 rows selected.
```

```
select * from emp;
```

```
select * from dept;
```

```
SCOTT@PROD> select * from emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

```
14 rows selected.
```

```
SCOTT@PROD> select * from dept;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

1.6 【选择特定列】

```
select empno,ename from emp;
```



```
SCOTT@PROD> select empno,ename from emp;
```

EMPNO	ENAME
7369	SMITH
7499	ALLEN
7521	WARD
7566	JONES
7654	MARTIN
7698	BLAKE
7782	CLARK
7788	SCOTT
7839	KING
7844	TURNER
7876	ADAMS
7900	JAMES
7902	FORD
7934	MILLER

1.6.1 【伪列：rownum】

select 子句中还可以选择**伪列**。

伪列从功能上看像是表中的一列，实际上并不存储在实际的表中。

常见伪列：rownum,rowid(对象号,文件号,块号,行号),currval,nextval

伪列上不能进行插入和删除操作。

官方文档：伪列

SQL Language Reference -> 2 Pseudocolumns

rownum 伪列反映的是每一行在当前查询的**结果集**中的位置。

```
select rownum,deptno,dname,loc from dept;
```

```
SCOTT@PROD>select rownum,deptno,dname,loc from dept;
```

ROWNUM	DEPTNO	DNAME	LOC
1	10	ACCOUNTING	NEW YORK
2	20	RESEARCH	DALLAS
3	30	SALES	CHICAGO
4	40	OPERATIONS	BOSTON

1.7 【描述表的结构】describe

不知道具体列名时，可使用 describe 命令查看表结构。

```
desc dept
```

```
SCOTT@ PROD>desc dept;
Name                                         Null?    Type
-----
DEPTNO                                       NOT NULL NUMBER(2)
DNAME                                       VARCHAR2(14)
LOC                                         VARCHAR2(13)
```

DESCRIBE :SQL*Plus 命令

官方文档: SQL*Plus 命令

SQL*Plus User's Guide and Reference -> 12 SQL*Plus Command Reference

1.8 【列标题的缺省格式】



列标题默认属性

- SQL Developer:
 - 默认标题对齐: 左对齐
 - 默认标题显示: 大写
- SQL*Plus:
 - 字符和日期类型的列左对齐
 - 数值类型的列右对齐
 - 默认字段显示方式: 大写

```
select * from emp;
```

```
SCOTT@PROD> select * from emp;

  EMPNO  ENAME      JOB              MGR  HIREDATE          SAL          COMM          DEPTNO
-----
   7369  SMITH      CLERK              7902  17-DEC-80           800              0             20
   7499  ALLEN     SALESMAN           7698  20-FEB-81          1600             300             30
   7521  WARD      SALESMAN           7698  22-FEB-81          1250             500             30
   7566  JONES     MANAGER            7839  02-APR-81          2975              0             20
   7654  MARTIN    SALESMAN           7698  28-SEP-81          1250            1400             30
   7698  BLAKE     MANAGER            7839  01-MAY-81          2850              0             30
   7782  CLARK     MANAGER            7839  09-JUN-81          2450              0             10
   7788  SCOTT     ANALYST            7566  19-APR-87          3000              0             20
   7839  KING      PRESIDENT          17-NOV-81          5000              0             10
   7844  TURNER    SALESMAN           7698  08-SEP-81          1500              0             30
   7876  ADAMS     CLERK              7788  23-MAY-87          1100              0             20
   7900  JAMES     CLERK              7698  03-DEC-81           950              0             30
   7902  FORD      ANALYST            7566  03-DEC-81          3000              0             20
   7934  MILLER    CLERK              7782  23-JAN-82          1300              0             10

14 rows selected.
```

ENAME 是字符类型, HIREDATE 是日期类型:左对齐;
 EMPNO、MGR、SAL 等是数值类型: 右对齐。
 数据是区分大小写的。

1.9 【格式化列的宽度】column .. format

```
column hiredate format a10
col mgr for 9999
select * from emp;
```

```
SCOTT@PROD> column hiredate format a10
SCOTT@PROD> col mgr for 9999
SCOTT@PROD> select * from emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30

```
SCOTT@PROD> col mgr for 99
SCOTT@PROD> select * from emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	###	17-DEC-80	800		20
7499	ALLEN	SALESMAN	###	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	###	22-FEB-81	1250	500	30
7566	JONES	MANAGER	###	02-APR-81	2975		20
7654	MARTIN	SALESMAN	###	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	###	01-MAY-81	2850		30
7782	CLARK	MANAGER	###	09-JUN-81	2450		10
7788	SCOTT	ANALYST	###	19-APR-87	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	###	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	###	23-MAY-87	1100		20
7900	JAMES	CLERK	###	03-DEC-81	950		30
7902	FORD	ANALYST	###	03-DEC-81	3000		20
7934	MILLER	CLERK	###	23-JAN-82	1300		10

14 rows selected.

【补充说明】

如果列宽度设置不够，会出现###。

1.10 【数学表达式在 select 中】

select 语句功能强大，可以在 SELECT 语句中用数学表达式对列值进行计算。

算数表达式

使用算术运算符可以创建包含数字和日期数据的表达式

操作符	描述
+	Add 加
-	Subtract 减
*	Multiply 乘
/	Divide 除

需求：涨工资，查询每个员工工资增加 300 以后的值

```
select ename,sal,sal+300 from emp;
```

```
SCOTT@PROD> select ename,sal,sal+300 from emp;
```

ENAME	SAL	SAL+300
SMITH	800	1100
ALLEN	1600	1900
WARD	1250	1550
JONES	2975	3275
MARTIN	1250	1550
BLAKE	2850	3150
CLARK	2450	2750
SCOTT	3000	3300
KING	5000	5300
TURNER	1500	1800
ADAMS	1100	1400
JAMES	950	1250
FORD	3000	3300
MILLER	1300	1600

```
14 rows selected.
```

1.10.1 【表达式的优先级】

表达式的优先级



乘和除优先级高于加、减

相同优先级的操作符自左向右进行计算

圆括号 () 用来重新组合表达式的优先级

1.10.2 【圆括号改变优先级】

需求：查询员工全年的收入（年终奖 100/每月奖励 100）

```
select ename,sal,12*sal+100,12*(sal+100) from emp;
```

```
SCOTT@PROD> select ename,sal,12*sal+100,12*(sal+100) from emp;
```

ENAME	SAL	12*SAL+100	12*(SAL+100)
SMITH	800	9700	10800
ALLEN	1600	19300	20400
WARD	1250	15100	16200
JONES	2975	35800	36900
MARTIN	1250	15100	16200
BLAKE	2850	34300	35400
CLARK	2450	29500	30600
SCOTT	3000	36100	37200
KING	5000	60100	61200
TURNER	1500	18100	19200
ADAMS	1100	13300	14400
JAMES	950	11500	12600
FORD	3000	36100	37200
MILLER	1300	15700	16800

```
14 rows selected.
```

1.11 【空值的定义和处理】

需求：查询员工工资+奖金总收入

```
select ename,sal,comm,sal+comm from emp;
```

```
SCOTT@PROD> select ename,sal,comm,sal+comm from emp;
```

ENAME	SAL	COMM	SAL+COMM
SMITH	800		
ALLEN	1600	300	1900
WARD	1250	500	1750
JONES	2975		
MARTIN	1250	1400	2650
BLAKE	2850		
CLARK	2450		
SCOTT	3000		
KING	5000		
TURNER	1500	0	1500
ADAMS	1100		
JAMES	950		
FORD	3000		
MILLER	1300		



定义空值

- NULL是无效的、未指定的、未知的或不可预知的值。
- NULL不是0，也不是空格。
- 包含空值的数学表达式的值都为空值

1.12 【定义列别名 as、空格、双引号】

需求：查询员工编号、姓名、工资。

未使用别名时的显示：

```
select empno,ename,sal from emp;
```

```
SCOTT@PROD> select empno,ename,sal from emp;
```

EMPNO	ENAME	SAL
7369	SMITH	800
7499	ALLEN	1600
7521	WARD	1250
7566	JONES	2975
7654	MARTIN	1250
7698	BLAKE	2850
7782	CLARK	2450
7788	SCOTT	3000
7839	KING	5000
7844	TURNER	1500
7876	ADAMS	1100
7900	JAMES	950
7902	FORD	3000
7934	MILLER	1300

【使用别名目的】为了以友好的方式显示，可以定义列的别名。

```
select empno,ename,sal,comm,sal+nvl(comm,0) new_sal from emp;
```

```
SCOTT@PROD> select empno,ename,sal,comm,sal+nvl(comm,0) new_sal from emp;
```

EMPNO	ENAME	SAL	COMM	NEW_SAL
7369	SMITH	800		800
7499	ALLEN	1600	300	1900
7521	WARD	1250	500	1750
7566	JONES	2975		2975
7654	MARTIN	1250	1400	2650
7698	BLAKE	2850		2850
7782	CLARK	2450		2450
7788	SCOTT	3000		3000
7839	KING	5000		5000
7844	TURNER	1500	0	1500
7876	ADAMS	1100		1100
7900	JAMES	950		950
7902	FORD	3000		3000
7934	MILLER	1300		1300

当别名中有特殊字符（如空格、/ 等）、别名需要区分大小写时，要用双引号。

```
select empno as ID,ename Name,sal "Salary" from emp;
```

```
SCOTT@PROD>select empno as ID,ename Name,sal "Salary" from emp;
```

ID	NAME	Salary
7369	SMITH	800
7499	ALLEN	1600
7521	WARD	1250
7566	JONES	2975
7654	MARTIN	1250

【使用别名注意事项】

列的别名不能在数学表达式中运算。

```
select sal salary ,salary + 300 from emp;
```

```
SCOTT@PROD>select sal salary ,salary + 300 from emp;
select sal salary ,salary + 300 from emp
*
ERROR at line 1:
ORA-00904: "SALARY": invalid identifier
```

1.13 【字符串、连接符||、引用运算符@】

1.13.1 字符串



字符串

- 字符串可以是 **SELECT** 列表中的一个字符,数字,日期。
- 日期和字符只能在单引号中出现。
- 每当返回一行时，字符串被输出一次。

```
select 'ename' from emp;
```

```
SCOTT@PROD> select 'ename' from emp;

' ENAM
-----
ename
ename
ename
ename
ename
ename
ename
ename
ename
ename
ename
ename
ename
ename
```

1.13.2 连接符



连接符

连接符：

- 把列与列，列与字符连接在一起。
- 用 ‘||’ 表示。
- 可以用来‘合成’列。

Oracle 使用单引号作为封装或包含其他字符的特殊符号。

需求：SMITH is a CLERK

```
select ename,job,ename||' is a '||job from emp;
```

```
SCOTT@PROD> select ename,job,ename||' is a '||job from emp;
```

ENAME	JOB	ENAME 'ISA' JOB
SMITH	CLERK	SMITH is a CLERK
ALLEN	SALESMAN	ALLEN is a SALESMAN
WARD	SALESMAN	WARD is a SALESMAN
JONES	MANAGER	JONES is a MANAGER
MARTIN	SALESMAN	MARTIN is a SALESMAN
BLAKE	MANAGER	BLAKE is a MANAGER
CLARK	MANAGER	CLARK is a MANAGER
SCOTT	ANALYST	SCOTT is a ANALYST
KING	PRESIDENT	KING is a PRESIDENT
TURNER	SALESMAN	TURNER is a SALESMAN
ADAMS	CLERK	ADAMS is a CLERK
JAMES	CLERK	JAMES is a CLERK
FORD	ANALYST	FORD is a ANALYST
MILLER	CLERK	MILLER is a CLERK

【特殊情况】

需要显示单引号时，通常在字符串中应该出现的单引号前面再加一个单引号进行转义。

需求：SMITH's job is CLERK

【错误用法】

```
select ename ||''s job is '|| job from emp;
```

```
SCOTT@PROD> select ename ||''s job is '|| job from emp;  
ERROR:  
ORA-01756: quoted string not properly terminated
```

【正确用法】

```
select ename ||'|'s job is '|| job from emp;
```

此处 ' 符号为转移符


```
SCOTT@PROD> select ename || '''s job is ''' job from emp;

ENAME||'''SJOBIS'||JOB
-----
SMITH's job is CLERK
ALLEN's job is SALESMAN
WARD's job is SALESMAN
JONES's job is MANAGER
MARTIN's job is SALESMAN
BLAKE's job is MANAGER
CLARK's job is MANAGER
SCOTT's job is ANALYST
KING's job is PRESIDENT
TURNER's job is SALESMAN
ADAMS's job is CLERK
JAMES's job is CLERK
FORD's job is ANALYST
MILLER's job is CLERK
```

1.13.3 引用运算符



Alternative Quote (q) Operator 引用操作符

- 指定您自己的引号分隔符。
- 选择任何分隔符。
- 增加可读性和可用性。

```
select ename||q'<'s job is >'||job from emp;
select ename||q'\ 's job is \'||job from emp;
```

```
SCOTT@PROD> select ename||q'<'s job is >'||job from emp;

ENAME||Q'<'SJOBIS>'||JOB
-----
SMITH's job is CLERK
ALLEN's job is SALESMAN
WARD's job is SALESMAN
JONES's job is MANAGER
MARTIN's job is SALESMAN
BLAKE's job is MANAGER
CLARK's job is MANAGER
SCOTT's job is ANALYST
KING's job is PRESIDENT
TURNER's job is SALESMAN
ADAMS's job is CLERK
JAMES's job is CLERK
FORD's job is ANALYST
MILLER's job is CLERK
```

{ } [] \ < > 等符号内部的字符作为普通字符正常显示。

【扩展应用】

连接运算符在生成批量脚本时非常有用,提高工作效率:

如构造描述 scott 用户所有表结构的 sql 语句

```
desc emp
select table_name from user_tables;
```

```
select 'desc '|| table_name from user_tables;
```

```
desc DEPT
desc EMP
desc BONUS
desc SALGRADE
desc PLAN_TABLE
desc EMP2
desc EMP1
```

7 rows selected.

1.14 【消除重复行】(distinct)

需求: 查询 emp 表中员工工作在哪个部门

先查看原始表情况:

```
select * from emp;
select deptno from emp;
```

使用 distinct 关键字来消除重复行

```
select distinct deptno from emp;
```

```
SCOTT@ PROD>select distinct deptno from emp;

DEPTNO
-----
      30
      20
      10
```

1.14.1 对多个列去重

DISTINCT 关键字可以从结果集中删除重复行,可以对多个列进行去重。行是唯一还是重复,取决于 distinct 关键字之后指定的内容。

```
select distinct deptno,ename from emp;
```



```
SCOTT@PROD> select distinct deptno,ename from emp;
```

DEPTNO	ENAME
20	JONES
30	WARD
20	SCOTT
10	KING
30	JAMES
30	ALLEN
30	MARTIN
30	BLAKE
20	FORD
20	SMITH
20	ADAMS
10	MILLER
10	CLARK
30	TURNER

```
14 rows selected.
```

```
select distinct deptno,job from emp;
```

```
SCOTT@PROD> select distinct deptno,job from emp;
```

DEPTNO	JOB
20	CLERK
30	SALESMAN
20	MANAGER
30	CLERK
10	PRESIDENT
30	MANAGER
10	CLERK
10	MANAGER
20	ANALYST

```
9 rows selected.
```

1.14.2 distinct 的误用

distinct 关键字前不能查询其他列；

```
select empno,distinct deptno from emp;
```

一个 select 子句中不能出现多个 distinct

```
select distinct empno,distinct deptno from emp;
```

```
SCOTT@PROD>select empno,distinct deptno from emp;  
select empno,distinct deptno from emp  
*
```

```
ERROR at line 1:  
ORA-00936: missing expression
```

```
SCOTT@PROD>select distinct empno,distinct deptno from emp;  
select distinct empno,distinct deptno from emp  
*
```

```
ERROR at line 1:  
ORA-00936: missing expression
```

1.15 【save 保存查询语句】

执行查询语句

```
select distinct ename,deptno from emp;
```

查看历史命令

list 或者 **l**

保存此 sql 语句到操作系统中 /home/oracle/p1_01.sql

save /home/oracle/p1_01.sql

在 sqlplus 中查看保存在操作系统中文件

get /home/oracle/p1_01.sql 或 **!cat** /home/oracle/p1_01.sql

sqlplus 中执行操作系统中的 sql 脚本/home/oracle/p1_01.sql

SCOTT@PROD> @/home/oracle/p1_01.sql

```
SCOTT@PROD>save /home/oracle/p1_01.sql
Created file /home/oracle/p1_01.sql
SCOTT@PROD>
SCOTT@PROD>
SCOTT@PROD>! cat /home/oracle/p1_01.sql
select distinct ename,deptno from emp
/

SCOTT@PROD>
SCOTT@PROD>@/home/oracle/p1_01.sql

ENAME                DEPTNO
-----
SMITH                  20
JONES                  20
FORD                   20
CLARK                  10
SCOTT                  20
TURNER                 30
JAMES                  30
KING                   10
ADAMS                  20
BLAKE                  30
MILLER                 10
ALLEN                  30
WARD                   30
MARTIN                 30

14 rows selected.
```

【总结】



小结

在本课中，您应该已经学会如何：

- 写一个 SELECT 语句：
 - 从表中返回所有行和列
 - 从表中返回指定的列
 - 使用列的别名，显示更具描述性的列标题

```
SELECT *|[DISTINCT] column|expression [alias],...|  
FROM table;
```



云和恩墨 成就所托

```
select empno as ID,ename Name,comm,(nvl(comm,0)+sal)*12 "Total Salary"  
  from emp;  
select distinct deptno,empno from emp;  
describe dept  
column hiredate format a20  
select ename || '''s job is '|| job from emp;
```

【课后练习】

1.描述 `scott.emp` 表的结构，查询 `emp` 表中所有数据。

```
desc emp;  
select * from emp;
```

2.找出下面语句中的错误并修正。

```
select empno ename,job sal x 12 New Salary from emp;  
select empno ename,job,sal*12 "New Salary" from emp;
```

3.查询 `scott.emp` 表中，所有员工的姓名、工资、工资与奖金之和，列标题显示为如下格式：NAME SALARY Total_Salary。

```
select ename as name,sal salary,sal+nvl(comm,0) "Total_Salary" from emp;
```

4.查看最后一次执行的查询语句，将此语句保存在操作系统中：`/home/oracle/les01_01.sql`，在 `sqlplus` 中查看并执行这个脚本。

```
list  
save /home/oracle/les01_01.sql  
get /home/oracle/les01_01.sql  
@/home/oracle/les01_01.sql
```