第2章 数据查询

2.1 数据库查询语言SQL

2.1.1 SQL语言的特点和分类

SQL语言有以下的主要特点：

* SQL语言可以在Oracle数据库中创建、存储、更新、检索和维护数据，其中主要的功能是实现数据的查询和数据的插入、删除、修改等操作。

* SQL语言在书写上类似于英文，简洁清晰，易于理解。它由关键字、表名、字段名，表达式等部分构成。

* SQL语言属于非过程化的4GL(第四代语言)。

* SQL语言按功能可分为DDL语言、DML语言、DCL语言和数据库事务处理语言四个类别。

* SQL语言的主要关键字有：ALTER、DROP、REVOKE、AUDIT、GRANT、ROLLBACK、COMMIT、INSERT、SELECT、COMMENT 、 LOCK 、 UPDATE 、 CREATE 、 NOAUDIT 、 VALIDATE、DELETE、RENAME等。

按照SQL语言的不同功用，可以进一步对SQL语言进行划分。下表给出了SQL语言的分类和功能简介。

表2-1 SQL语言的分类

类 别	功 能	举 例
数据库控制语言(DCL)	控制对数据库的访问，启动和关闭等	对系统权限进行授权和回收的 GRANT、REVOKE 等语句
数据库定义语言(DDL)	用来创建、删除及修改数据库对象	创建表和索引的 CREATE TABLE、ALTER INDEX 等语句
数据库操纵语言(DML)	用来操纵数据库的内容，包括查询	查询、插入、删除、修改和锁定操作的 SELECT、INSERT、UPDATE、DELETE、LOCK TABLE 等语句
数据库事务处理	实现对数据的交易过程的完整控制	与数据库事物处理相关的 COMMIT、 ROLLBACK、SAVEPOINT、SET TRANSACTION 等语句

2.1.2 SQL的基本语法

SQL语言的语法比较简单，类似于书写英文的语句。其语句一般由主句和若干个从句组成，主句和从句都由关键字引导。主句表示该语句的主要功能，从句表示一些条件或限定，有些从句是可以省略的。在语句中会引用到列名、表名或表达式。另外还有如下一些说明：

* 关键字、字段名、表名等之间都要用空格或逗号等进行必要的分隔。

* 语句的大小写不敏感(查询的内容除外)。

* 语句可以写在一行或多行。

* 语句中的关键字不能略写和分开写在两行。

* 要在每条SQL语句的结束处添加“；”号。

* 为了提高可读性，可以使用缩进。

* 从句一般写在另一行的开始处。

查询语句是最常见的SQL语句，它从给定的表中，把满足条件的内容检索出来。以下是最基本的SELECT语句语法。

SELECT 字段名列表 FROM 表名 WHERE 条件;

SELECT为查询语句的关键字，后跟要查询的字段名列表，字段名列表用来指定检索特定的字段，该关键字不能省略。

字段名列表代表要查询的字段。

FROM 也是查询语句关键字，后面跟要查询的表名，该关键字不能省略。

WHERE条件限定检索特定的记录，满足“条件”的记录被显示出来，不满足条件的被过滤掉。

语句查询的结果往往是表的一部分行和列。如果字段名列表使用*，将检索全部的字段。如果省略WHERE条件，将检索全部的记录。

【训练1】 查询部门10的雇员。

输入并执行查询：

```
SELECT * FROM emp WHERE deptno=10;
```

结果略。

说明：该查询语句从emp表中检索出部门10 的雇员，条件由WHERE deptno=10 子句指定。

2.2 基本查询和排序

2.2.1 查询的基本用法

在Oracle数据库中，对象是属于模式的，每个账户对应一个模式，模式的名称就是账户名称。在表名前面要添加模式的名字，在表的模式名和表名之间用“.”分隔。我们以不同的账户登录数据库时，就进入了不同的模式，比如登录到STUDENT 账户，就进入了STUDENT模式。而在STUDENT模式要查询属于SCOTT模式的表，就需要写成：

```
SELECT * FROM SCOTT.EMP;
```

但如果登录用户访问属于用户模式本身的表，那么可以省略表名前面的模式名称。

SELECT * FROM emp;

1．指定检索字段

下面的练习，只显示表的指定字段。

【训练1】 显示DEPT表的指定字段的查询。

输入并执行查询：

SELECT deptno,dname FROM dept;

显示结果如下：

```
DEPTNO DNAME
-------------- -------------
    10  ACCOUNTING
    20  RESEARCH
    30  SALES
    40  OPERATIONS
```

说明：结果只包含2列deptno和dname。在语句中给出要显示的列名，列名之间用“，”分隔。表头的显示默认为全部大写。对于日期和数值型数据，右对齐显示，如deptno列。对于字符型数据，左对齐显示，如dname列。

【练习1】显示emp表的雇员名称和工资。

2．显示行号

每个表都有一个虚列ROWNUM，它用来显示结果中记录的行号。我们在查询中也可以显示这个列。

【训练2】 显示EMP表的行号。

输入并执行查询：

```
SELECT rownum,ename FROM emp;
```

结果如下：

```
    ROWNUM ENAME
---------- ----------
         1 SMITH
         2 ALLEN
         3 WARD
         4 JONES
```

注意：显示的行号是查询结果的行号，数据在数据库中是没有行号的。

3．显示计算列

在查询语句中可以有算术表达式，它将形成一个新列，用于显示计算的结果，通常称为计算列。表达式中可以包含列名、算术运算符和括号。括号用来改变运算的优先次序。常用的算术运算符包括：

* +：加法运算符。
* -：减法运算符。
* *：乘法运算符。
* /：除法运算符。

以下训练在查询中使用了计算列。

【训练3】 显示雇员工资上浮20%的结果。

输入并执行查询:

```
SELECT ename,sal,sal*(1+20/100) FROM emp;
```

显示结果为:

```
ENAME      SAL                    SAL*(1+20/100)
---------- ---------- ----------------------
SMITH      800                    960
ALLEN      1600                   1920
```

说明:结果中共显示了3列,第3列显示工资上浮20%的结果,它不是表中存在的列,而是计算产生的结果,称为计算列。

【练习2】显示EMP表的雇员名称以及工资和津贴的和。

4．使用别名

我们可以为表的列起一个别名，它的好处是，可以改变表头的显示。特别是对于计算列，可以为它起一个简单的列别名以代替计算表达式在表头的显示。

【训练4】 在查询中使用列别名。

输入并执行：

```
SELECT ename AS 名称, sal 工资 FROM emp;
```

显示结果为：

```
名称            工资
--------------- -----------------
SMITH           800
ALLEN           1600
```

说明：表头显示的是列别名，转换为汉字显示。在列名和别名之间要用AS分隔，如ename和它的别名“名称”之间用AS隔开。AS也可以省略，如sal和它的别名“工资”之间用空格分割。

注意：如果用空格分割，要区别好列名和别名，前面为列名，后面是别名。

别名如果含有空格或特殊字符或大小写敏感，需要使用双引号将它引起来。

【训练5】 在列别名上使用双引号。

输入并执行查询：

SELECT ename AS "Name", sal*12+5000 AS "年度工资(加年终奖)" FROM emp;

显示结果为：

```
Name       年度工资(加年终奖)
---------- --------------------------
SMITH                  14600
ALLEN                  24200
```

说明：其中别名“Name”有大小写的区别，别名“年度工资(加年终奖)”中出现括号，属于特殊符号，所以都需要使用双引号将别名引起。

【练习3】显示DEPT表的内容，使用别名将表头转换成中文显示。

5．连接运算符

在前面，我们使用到了包含数值运算的计算列，显示结果也是数值型的。我们也可以使用字符型的计算列，方法是在查询中使用连接运算。连接运算符是双竖线“||”。通过连接运算可以将两个字符串连接在一起。

【训练6】 在查询中使用连接运算。

输入并执行查询：

```
SELECT          ename||job AS "雇员和职务表" FROM emp;
```

输出结果为：

```
雇员和职务表
------------------
SMITHCLERK
ALLENSALESMAN
```

说明：在本例中，雇员名称和职务列被连接成为一个列显示。

在查询中可以使用字符和日期的常量，表示固定的字符串或固定日期。字符和日期的常量需要用单引号引起。下一个训练是作为上一个训练的改进。

【训练7】 在查询中使用字符串常量。

输入并执行查询：

```
SELECT        ename|| ' IS '||job AS "雇员和职务表" FROM emp;
```

输出结果为：

```
雇员和职务表
----------------------
SMITH IS CLERK
ALLEN IS SALESMAN
```

说明：本练习中将雇员名称、字符串常量“ IS ”和雇员职务3个部分连接在一起。

【练习4】显示DEPT表的内容，按以下的形式:

部门ACCOUNTING所在的城市为NEW YORK

6．消除重复行

如果在显示结果中存在重复行，可以使用的关键字DISTINCT消除重复显示。

【训练8】 使用DISTINCT消除重复行显示。

输入并执行查询：

```
SELECT DISTINCT job FROM emp;
```

结果为：

```
JOB
---------
ANALYST
CLERK
MANAGER
PRESIDENT
SALESMAN
```

说明：在本例中，如果不使用DISTINCT关键字，将重复显示雇员职务，DISTINCT关键字要紧跟在SELECT之后。请去掉DISTINCT关键字，重新执行，并观察显示结果的不同。

【练习5】显示EMP表中不同的部门编号。

2.2.2 查询结果的排序

如果要在查询的同时排序显示结果，可以使用如下的语句：

SELECT 字段列表 FROM 表名 WHERE 条件

ORDER BY 字段名1 [ASC|DESC][,字段名2 [ASC|DESC]...];

ORDER BY从句后跟要排序的列。ORDER BY 从句出现在SELECT语句的最后。

排序有升序和降序之分，ASC表示升序排序，DESC表示降序排序。如果不指明排序顺序，默认的排序顺序为升序。如果要降序，必须书写DESC关键字。

1．升序排序

【训练1】 查询雇员姓名和工资，并按工资从小到大排序。

输入并执行查询：

```
SELECT ename, sal FROM emp ORDER BY sal;
```

执行结果为：

```
ENAME          SAL
---------- --------------------
SMITH          800
JAMES          950
```

注意：若省略ASC和DESC，则默认为ASC，即升序排序。

2．降序排序

【训练2】 查询雇员姓名和雇佣日期，并按雇佣日期排序，后雇佣的先显示。

输入并执行查询：

```
SELECT ename,hiredate FROM emp ORDER BY hiredate DESC;
```

结果如下：

```
ENAME      HIREDATE
---------- --------------------
ADAMS      23-5月 -87
SCOTT      19-4月 -87
```

MILLER 23-1月 -82

JAMES 03-12月-81

FORD 03-12月-81

注意： DESC表示降序排序，不能省略。

3．多列排序

可以按多列进行排序，先按第一列，然后按第二列、第三列.....。

【训练3】 查询雇员信息，先按部门从小到大排序，再按雇佣时间的先后排序。

输入并执行查询：

SELECT ename,deptno,hiredate FROM emp ORDER BY deptno,hiredate;

结果如下：

```
ENAME      DEPTNO HIREDATE
---------- ---------- ---------
CLARK              10 09-6月 -81
KING               10 17-11月-81
MILLER             10 23-1月 -82
SMITH              20 17-12月-80
JONES              20 02-4月 -81
FORD               20 03-12月-81
SCOTT              20 19-4月 -87
```

说明：该排序是先按部门升序排序，部门相同的情况下，再按雇佣时间升序排序。

4．在排序中使用别名

如果要对计算列排序，可以为计算列指定别名，然后按别名排序。

【训练4】 按工资和工作月份的乘积排序。

输入并执行查询：

```
SELECT empno, ename, sal*Months_between(sysdate,hiredate)
AS total FROM emp
ORDER BY total;
```

执行结果为：

```
 EMPNO ENAME      TOTAL
---------- ---------- --------------------
7876       ADAMS          221526.006
7369       SMITH          222864.661
7900       JAMES          253680.817
 7654      MARTIN         336532.484
```

说明：求得雇员工作月份的函数Months_between将在后面介绍。sysdate表示当前日期。

【练习1】将部门表中的部门名称按字母顺序显示。

2.3 条件查询

2.3.1 简单条件查询

要对显示的行进行限定，可在FROM从句后使用WHERE从句，在WHERE从句中给出限定的条件，因为限定条件是一个表达式，所以称为条件表达式。条件表达式中可以包含比较运算，表达式的值为真的记录将被显示。常用的比较运算符列于表2-2中。

表2-2 比 较 运 算 符

运算符	功 能	实 例
>,<	大于，小于	Select * from emp where sal>2000
>=.<=	大于等于，小于等于	Select * from emp where sal>=2000
=	等于	Select * from emp where deptno=10
!=,<>,^=	不等于	Select * from emp where deptno!=10

【训练1】 显示职务为“SALESMAN”的雇员的姓名、职务和工资。

输入并执行查询：

```
SELECT ename,job,sal FROM emp WHERE job='SALESMAN';
```

执行结果为：

```
ENAME          JOB           SAL
---------- --------------------- --------------
ALLEN          SALESMAN      1600
WARD           SALESMAN      1250
MARTIN         SALESMAN      1250
TURNER         SALESMAN      1500
```

说明：结果只显示职务为“SALESMAN”的雇员。字符串和日期型数据的值是包含在单引号中的，如SALESMAN，需要用单引号引起。字符的值对大小写敏感，在emp表中存放的职务字符串全部是大写。

注意：在本练习中，如果SALESMAN写成小写或大小写混合，将不会有查询结果输出。

【训练2】 显示工资大于等于3000的雇员姓名、职务和工资。

输入并执行查询：

```
SELECT ename, job,sal FROM emp WHERE sal>=3000;
```

执行结果为：

```
ENAME          JOB              SAL
---------- ------------------ ----------
SCOTT          ANALYST          3000
KING           PRESIDENT        5000
FORD           ANALYST          3000
```

说明：结果只显示工资大于等于3000的雇员。

缺省中文日期格式为DD-MM月-YY，如2003年1月10日应该表示为“10-1月-03”。

【训练3】 显示1982年以后雇佣的雇员姓名和雇佣时间。

输入并执行查询：

```
SELECT ename,hiredate FROM emp WHERE hiredate>='1-1月-82';
```

执行结果为：

```
ENAME      HIREDATE
---------- --------------------
```

SCOTT 19-4月 -87
ADAMS 23-5月 -87
MILLER 23-1月 -82

说明：检查hiredate字段的内容，都在82年以后。

【练习1】显示部门编号为10的雇员姓名和雇佣时间。

2.3.2 复合条件查询

可以用逻辑运算符构成复合的条件查询，即把两个或多个条件，用逻辑运算符连接成一个条件。有3个逻辑运算符，如表2-3所示。

表2-3 逻辑运算符

运算符	说 明	实 例
AND	逻辑与，表示两个条件必须同时满足	Select * from emp where sal>1000 and sal<2000
OR	逻辑或，表示两个条件中有一个条件满足即可	Select * from emp where deptno=10 or deptno=20
NOT	逻辑非，返回与某条件相反的结果	Select * from emp where not job='MANAGER'

运算的优先顺序是NOT，AND，OR。如果要改变优先顺序，可以使用括号。

下面是使用逻辑与运算的练习。

1．使用逻辑与

【训练1】 显示工资在1000～2000之间(不包括1000和2000)的雇员信息。

输入并执行查询：

SELECT ename, job,sal FROM emp WHERE sal>1000 AND sal<2000;

执行结果为：

```
ENAME      JOB              SAL
---------- ------------------ ----------
ALLEN      SALESMAN         1600
WARD       SALESMAN         1250
MARTIN     SALESMAN         1250
TURNER     SALESMAN         1500
ADAMS      CLERK            1100
MILLER     CLERK            1300
```

说明：两个条件需要同时满足，所以必须使用AND运算。

注意：条件sal>1000 AND sal<2000不能写成sal>1000 AND <2000。

【练习1】显示部门10中工资大于1500的雇员。

2．使用逻辑或

下面是使用逻辑或运算的练习。

【训练2】 显示职务为CLERK或MANAGER的雇员信息。

输入并执行查询：

```
SELECT * FROM emp WHERE job='CLERK' OR job='MANAGER';
```

执行结果从略。

说明：检索职务为'CLERK'或'MANAGER'的雇员，需要使用OR运算，请自行察看结果。

注意：条件job='CLERK' OR job='MANAGER'不能写成job='CLERK' OR 'MANAGER'。

3．使用逻辑非

下面是使用逻辑非运算的练习。

【训练3】 显示部门10以外的其他部门的雇员。

输入并执行查询：

```
SELECT * FROM emp WHERE NOT deptno=10;
```

执行结果从略。

说明：执行结果包含部门编号不等于10的其他部门的雇员，请自行察看结果。

4．使用逻辑或和逻辑与

下面是同时使用逻辑或和逻辑与的复合练习。

【训练4】 显示部门10和部门20中工资小于1500的雇员。

输入并执行查询

```
SELECT * FROM emp WHERE (deptno=10 OR deptno=20)
AND sal<1500;
```

执行结果为：

EMPNO ENAME JOB MGR HIREDATE SAL
COMM DEPTNO

---------- ---------- --------- ------- --------------- ---------- -----
--------- ----------

7369 SMITH CLERK 7902 17-12月-80 800 20

7876 ADAMS CLERK 7788 23-5月 -87 1100 20

7934 MILLER CLERK 7782 23-1月 -82 1300 10

注意：该练习中的括号是不可省的。如果省略，意义有所不同。

【练习2】请说明在如上练习中如果省略括号，该语句所代表的含义和查询的结果。

2.3.3 条件特殊表示法

使用如表2-4所示的特殊运算表示法，可使语句更为直观，易于理解。

表2-4 特殊运算符

运　算	功　能	实　例
[NOT] BETWEEN…AND…	用于测试是否在范围内	Select * from emp Where sal between 1000 and 2000
[NOT] IN (…)	用于测试是否在列表中	Select*from emp Where job in('CLERK','SALESMAN','ANYLYST')
[NOT] LIKE	用于进行模式匹配	Select * from emp Where ename like '%A%'
IS [NOT] NULL	用于测试是否为空值	Select * from emp Where comm is not null
ANY SOME	同列表或查询中的每一个值进行比较，测试是否有一个满足，前面必须使用的运算符包括=、!=、>=、<=、>、< 等	Select * from emp Where sal<any(select sal from emp where deptno=10)

表2-4 特殊运算符

ALL	同列表或查询中的每一个值进行比较，测试是否所有的值都满足，前面必须使用的运算符包括=、!=、>=、<=、>、<等	Select*from emp Where sal<all(1000,1500,2000)
[NOT] EXISTS	测试是否子查询至少返回一行	Select '存在雇员 SCOTT' from dual where exists(select*from emp where ename='SCOTT');

1．BETWEEN的用法

对于数值型或日期型数据，表示范围时可以用以下的特殊运算表示方法：

[NOT] BETWEEN... AND...

【训练1】 显示工资在1000～2000之间的雇员信息。

输入并执行查询：

```
SELECT * FROM emp WHERE sal BETWEEN 1000 AND 2000;
```

执行结果从略。

注意：下限在前，上限在后，不能颠倒。查询范围中包含上下限的值，因此在本例中，查询工资包含1000和2000在内。请自行执行并察看结果。

2．IN的用法

使用以下运算形式，可以显示值满足特定集合的结果：

[NOT] IN (...)

【训练2】 显示职务为“SALESMAN'，“CLERK”和“MANAGER”的雇员信息。

输入并执行查询：

```
SELECT * FROM emp WHERE job IN ('SALESMAN','CLERK','MANAGER');
```

执行结果从略。

注意：如果在IN前面增加NOT，将显示职务不在集合列表中的雇员。以上用法同样适用于数值型集合，请自行执行并察看结果。

【练习1】显示部门10和20的雇员信息。

3．LIKE的用法

使用LIKE操作符可完成按通配符查找字符串的查询操作，该操作符适合于对数据进行模糊查询。其语句法为：

[NOT] LIKE 匹配模式

匹配模式中除了可以包含固定的字符之外，还可以包含以下的通配符：

%：代表0个或多个任意字符。

_：代表一个任意字符。

【训练3】 显示姓名以“S”开头的雇员信息。

输入并执行查询：

```
SELECT * FROM emp WHERE ename LIKE 'S%';
```

执行结果为：

```
 EMPNO ENAME      JOB              MGR HIREDATE      SAL
COMM     DEPTNO
---------- ---------- --------------------- --------- ---------------- ---
-------- ----------- -----------------
7369 SMITH    CLERK 7902 17-12月-80       800     20
7788 SCOTT    ANALYST          7566 19-4月 -87   3000          20
```

说明：SMITH和SCOTT名字均以S开头，名字后边的字符和长度任意。

【训练4】显示姓名第二个字符为“A”的雇员信息。

执行查询：

```
SELECT * FROM emp WHERE ename LIKE '_A%';
```

执行结果从略，请自行执行并察看结果。

说明：“_”代表第一个字符任意，第二个字符必须为“A”，“%”代表第二个字符后面的字符为任意字符，个数任意。

【练习2】显示姓名中包含字符“A”的雇员信息。

4．判断空值NULL

在表中，字段值可以是空，表示该字段没有内容。如果不填写，或设置为空则我们说该字段的内容为NULL。NULL没有数据类型，也没有具体的值，但是使用特定运算可以判断出来。这个运算就是：

IS [NOT] NULL

【训练5】 显示经理编号没有填写的雇员。

输入并执行查询：

```
SELECT  ename, mgr FROM emp WHERE mgr IS NULL;
```

执行结果为：

```
ENAME      MGR
---------- ----------
KING
```

注意：以下用法是错误的。

```
SELECT  ename, mgr FROM emp WHERE mgr=NULL;
```

2.4 函数

2.4.1 数值型函数

常用的数值型函数如表2-5所示。

表2-5 数值型函数

函 数	功 能	实 例	结 果
abs	求绝对值函数	abs(−5)	5
sqrt	求平方根函数	sqrt(2)	1.41421356
power	求幂函数	power(2,3)	8
cos	求余弦三角函数	cos(3.14159)	−1
mod	求除法余数	mod(1600, 300)	100
ceil	求大于等于某数的最小整数	ceil(2.35)	3
floor	求小于等于某数的最大整数	floor(2.35)	2
round	按指定精度对十进制数四舍五入	round(45.923, 1) round(45.923, 0) round(45.923, −1)	45.9 46 50
trunc	按指定精度截断十进制数	trunc(45.923, 1) trunc(45.923) trunc(45.923, −1)	45.9 45 40

【训练1】 使用数值型函数练习。

步骤1：使用求绝对值函数abs。

```
SELECT abs(-5) FROM dual;
```

执行结果：

```
 ABS(-5)
--------------
          5
```

说明：求-5的绝对值，结果为5。

步骤2：使用求平方根函数sqrt。

```
SELECT sqrt(2) FROM dual;
```

执行结果：

```
 SQRT(2)
--------------
1.41421356
```

说明：2的平方根为1.41421356。

步骤3：使用ceil函数。

```
SELECT ceil(2.35) FROM dual;
```

执行结果：

```
CEIL(2.35)
----------
         3
```

说明：该函数求得大于等于2.35的最小整数，结果为3。

步骤4：使用floor函数。

```
SELECT floor(2.35) FROM dual;
```

执行结果：

```
FLOOR(2.35)
---------------
          2
```

说明：该函数求得小于等于2.35的最大整数，结果为2。

步骤5：使用四舍五入函数round。

SELECT round(45.923,2), round(45.923,0), round(45.923,-1) FROM dual;

执行结果：

```
ROUND(45.923,2) ROUND(45.923,0) ROUND(45.923, -1)
--------------- --------------- ----------------
          45.92              46               50
```

说明：该函数按照第二个参数指定的位置对第一个数进行四舍五入。2代表对小数点后第三位进行四舍五入，0 代表对小数位进行四舍五入，-1代表对个位进行四舍五入。

步骤6：使用截断函数trunc。

SELECT trunc(45.923,2), trunc(45.923),trunc(45.923, -1) FROM dual;

执行结果：

TRUNC(45.923,2) TRUNC(45.923) TRUNC(45.923, -1)

---------- ---------- ----------

45.92 45 40

说明：该函数按照第二个参数指定的位置对第一个数进行截断。2代表对小数点后第三位进行截断，0 代表对小数位进行截断，-1代表对个位进行截断。

步骤7：使用求余数函数mod。

```
SELECT mod(1600, 300) FROM dual;
```

执行结果：

```
MOD(1600,300)
-------------
100
```

说明：1600除以300的商为5，余数为100。

步骤8：使用cos函数。

```
SELECT cos(3.14159) FROM dual;
```

执行结果：

```
COS(3.14159)
----------------
          -1
```

说明：cos函数的输入参数应为弧度，3.14159的cos值为-1。

函数可以嵌套使用，看如下例子。

【训练2】 求|sin(230°)|的值，保留一位小数。

步骤1：执行查询。

```
SELECT sin(230*3.14159/180) FROM dual;
```

结果为：

```
SIN(230*3.14159/180)
---------------------------
                -.76604226
```

说明：先将230°转换成为弧度，然后进行计算求值。

步骤2：求绝对值。

```
SELECT abs(sin(230*3.14159/180)) FROM dual;
```

结果为：

```
ABS(SIN(230*3.14159/180))
---------------------------------
                .766042264
```

说明：本步骤求绝对值。

步骤3：保留一位小数。

SELECT round(abs(sin(230*3.14159/180)),1) FROM dual;

结果为：

ROUND(ABS(SIN(230*3.14159/180)),1)

.8

说明：本步骤进行四舍五入，保留小数点后1位。

【练习1】求23/2，四舍五入，保留一位小数。

2.4.2 字符型函数

字符型函数包括大小写转换和字符串操作函数。大小写转换函数有3个。常用的字符型函数如表2-6所示。

表2-6 字符函数

函数名称	功　　能	实　例	结　果
ascii	获得字符的 ASCII 码	Ascii('A')	65
chr	返回与 ASCII 码相应的字符	Chr(65)	A
lower	将字符串转换成小写	lower ('SQL Course')	sql course
upper	将字符串转换成大写	upper('SQL Course')	SQL COURSE
initcap	将字符串转换成每个单词以大写开头	initcap('SQL course')	Sql Course
concat	连接两个字符串	concat('SQL', ' Course')	SQL Course
substr	给出起始位置和长度，返回子字符串	substr('String',1,3)	Str
length	求字符串的长度	length('Wellcom')	7
instr	给出起始位置和出现的次数，求子字符串在字符串中出现的位置	instr('String', 'r',1,1)	3
lpad	用字符填充字符串左侧到指定长度	lpad('Hi',10,'-')	--------Hi
rpad	用字符填充字符串右侧到指定长度	rpad('Hi',10,'-')	Hi--------
trim	在一个字符串中去除另一个字符串	trim('S' FROM 'SSMITH')	MITH
replace	用一个字符串替换另一个字符串中的子字符串	replace('ABC', 'B', 'D')	ADC

【训练1】 如果不知道表的字段内容是大写还是小写，可以转换后比较。

输入并执行查询：

```
SELECT  empno, ename, deptno    FROM emp
WHERE  lower(ename) ='blake';
```

结果为：

```
 EMPNO ENAME         DEPTNO
-------------- --------- ----------------------
 7698 BLAKE             30
```

说明：该查询将表中的雇员名转换成小写，与小写的blake进行比较。

【训练2】 显示雇员名称和职务列表。

输入并执行查询：

```
SELECT  concat(rpad(ename,15,'.'),job) as 职务列表 FROM  emp;
```

结果为：

```
职务列表
-----------------------------------
SMITH.....CLERK
ALLEN.....SALESMAN
WARD.....SALESMAN
```

说明：rpad函数向字符串的右侧添加字符，以达到指定宽度。该例中雇员名称右侧连接若干个“.”，凑足15位，然后与雇员职务连接成列表。本例中使用了函数的嵌套。

【训练3】 显示名称以“W”开头的雇员，并将名称转换成以大写开头。

输入并执行查询：

```
SELECT empno,initcap(ename),job FROM emp
WHERE substr(ename,1,1)='W';
```

结果为：

```
  EMPNO INITCAP(EN JOB
--------- ---------- ---------
 7521       Ward      SALESMAN
```

说明：本例在字段列表和查询条件中分别应用了函数initcap和substr。函数initcap将雇员名称转换成以大写开头。函数substr返回ename从第一个字符位置开始，长度为1的字符串，即第一个字符，然后同大写W比较。

【训练4】 显示雇员名称中包含“S”的雇员名称及名称长度。

输入并执行查询：

```
SELECT empno,ename,length(ename) FROM emp
        WHERE instr(ename, 'S', 1, 1)>0;
```

执行结果为：

```
EMPNO ENAME    LENGTH(ENAME)
------------ ------------ -------------------------
     7369 SMITH         5
     7566 JONES                   5
```

【训练4】 显示雇员名称中包含“S”的雇员名称及名称长度。

输入并执行查询：

```
SELECT empno,ename,length(ename) FROM emp
        WHERE instr(ename, 'S', 1, 1)>0;
```

执行结果为：

```
 EMPNO ENAME    LENGTH(ENAME)
------------ ------------ ------------------------
        7369 SMITH                    5
7566 JONES                    5
```

说明：本例在字段列表和查询条件中分别应用了函数length和instr。Length函数返回ename的长度。instr(ename,'S'1,1)函数返回ename中从第一个字符位置开始，字符串“S”第一次出现的位置。如果函数返回0，则说明ename中不包含字符串“S”；如果函数返回值大于0，则说明ename中包含字符串“S”。

【练习1】显示部门表中部门和所在城市列表，中间以下划线“_”连接，城市名转换成以大写字母开头。

2.4.3 日期型函数

Oracle使用内部数字格式来保存时间和日期，包括世纪、年、月、日、小时、分、秒。缺省日期格式为 DD-MON-YY，如“08-05月-03”代表2003年5月8日。

SYSDATE是返回系统日期和时间的虚列函数。

使用日期的加减运算，可以实现如下功能：

* 对日期的值加减一个天数，得到新的日期。

* 对两个日期相减，得到相隔天数。

* 通过加小时来增加天数，24小时为一天，如12小时可以写成12/24(或0.5)。

还有如表2-7所示的日期函数可以使用。

表2-7 日期函数

函 数	功 能	实 例	结 果
months_between	返回两个日期间的月份	months_between ('04-11 月-05','11-1 月-01')	57.7741935
add_months	返回把月份数加到日期上的新日期	add_months('06-2 月-03',1) add_months('06-2 月-03',-1)	06-3 月-03 06-1 月-03
next_day	返回指定日期后的星期对应的新日期	next_day('06-2 月-03','星期一')	10-2 月-03
last_day	返回指定日期所在的月的最后一天	last_day('06-2 月-03')	28-2 月-03
round	按指定格式对日期进行四舍五入	round(to_date('13-2 月-03'),'YEAR') round(to_date('13-2 月-03'),'MONTH') round(to_date('13-2 月-03'),'DAY')	01-1 月-03 01-2 月-03 16-2 月-03 (按周四舍五入)
trunc	对日期按指定方式进行截断	trunc(to_date('06-2 月-03'),'YEAR') trunc(to_date('06-2 月-03'),'MONTH') trunc(to_date('06-2 月-03'),'DAY')	01-1 月-03 01-2 月-03 02-2 月-03 (按周截断)

【训练1】 返回系统的当前日期。

输入并执行查询：

```
SELECT sysdate FROM dual;
```

返回结果为：

```
SYSDATE
--------------
06-2月-03
```

说明：该查询返回执行该查询时的数据库服务器的系统当前时间，日期显示格式为默认格式，如“06-2月-03”表示03年2月6日。

【训练2】 返回2003年2月的最后一天。

输入并执行查询：

```
SELECT last_day('08-2月-03') FROM dual;
```

返回结果为：

```
LAST_DAY('
---------------
28-2月-03
```

说明：该函数给定参数为某月份的任意一天，返回时间为该月份的最后一天。本例中，参数为03年2月8号，返回日期为03年2月28日，是该月的最后一天。

【训练3】 假定当前的系统日期是2003年2月6日，求再过1000天的日期。

输入并执行查询：

```
SELECT sysdate+1000 AS "NEW DATE" FROM dual;
```

返回结果为：

```
NEW DATE
---------------
04-11月-05
```

说明：该查询使用到了日期的加法运算，求经过一定天数后的新日期。

【训练4】 假定当前的系统日期是2003年2月6日，显示部门10雇员的雇佣天数。

输入并执行查询：

```
SELECT ename, round(sysdate-hiredate) DAYS
FROM   emp
WHERE  deptno = 10;
```

返回结果为：

```
ENAME          DAYS
-------------- --------------------
CLARK              7913
KING               7752
MILLER             7685
```

说明：该查询使用日期的减法运算求两个日期的相差天数。用round函数对天数进行四舍五入。

【练习1】显示雇员名称和雇佣的星期数。

【练习2】显示从本年1月1日开始到现在经过的天数(当前时间取SYSDATE的值)。

2.4.4 转换函数

Oracle的类型转换分为自动类型转换和强制类型转换。常用的类型转换函数有TO_CHAR、TO_DATE或TO_NUMBER，如表2-8所示。

表2-8 类型转换函数

函 数	功 能	实 例	结 果
To_char	转换成字符串类型	To_char(1234.5, '$9999.9')	$1234.5
To_date	转换成日期类型	To_date('1980-01-01', 'yyyy-mm-dd')	01-1 月-80
To_number	转换成数值类型	To_number('1234.5')	1234.5

1. 自动类型转换

Oracle可以自动根据具体情况进行如下的转换：

* 字符串到数值。
* 字符串到日期。
* 数值到字符串。
* 日期到字符串。

以下是自动转换的训练。

【训练1】 自动转换字符型数据到数值型。

输入并执行查询：

```
SELECT '12.5'+11 FROM dual;
```

执行结果为：

```
'12.5'+11
-----------
      23.5
```

说明：在本训练中，因为出现+运算符，说明进行的是算术运算，所以字符串'12.5'被自动转换成数值12.5，然后参加运算。

【训练2】 自动转换数值型数据到字符型。

执行以下查询：

```
SELECT '12.5'||11 FROM dual;
```

结果为：

```
'12.5'
------
12.511
```

说明：在本训练中，因为出现||运算符，说明进行的是字符串连接运算，数值11被自动转换成字符串'11'，然后参加运算。

2．日期类型转换

将日期型转换成字符串时，可以按新的格式显示。

如格式YYYY-MM-DD HH24:MI:SS表示“年-月-日 小时:分钟:秒”。Oracle的日期类型是包含时间在内的。

主要的日期格式字符的含义如表2-9所示。

表2-9 日期转换格式字符

代　码	代表的格式	例　子
AM、PM	上午、下午	08 AM
D	数字表示的星期(1～7)	1,2,3,4,5,6,7
DD	数字表示月中的日期(1～31)	1,2,3,…,31
MM	两位数的月份	01,02,…,12
Y、YY、YYY、YYYY	年份的后几位	3,03,003,2003
RR	解决 Y2K 问题的年度转换	
DY	简写的星期名	MON,TUE,FRI,…
DAY	全拼的星期名	MONDAY,TUESDAY,…

表2-9 日期转换格式字符

MON	简写的月份名	JAN,FEB,MAR,…
MONTH	全拼的月份名	JANUARY,FEBRUARY,…
HH、HH12	12 小时制的小时(1～12)	1,2,3,…,12
HH24	24 小时制的小时(0～23)	0,1,2,…,23
MI	分(0～59)	0,1,2,…,59
SS	秒(0～59)	0,1,2,…,59
,./-;:	原样显示的标点符号	
'TEXT'	引号中的文本原样显示	TEXT

【训练3】 将日期转换成带时间和星期的字符串并显示。

执行以下查询：

```
SELECT    TO_CHAR(sysdate,'YYYY-MM-DD    HH24:MI:SS
AM DY') FROM dual;
```

结果为：

```
TO_CHAR(SYSDATE,'YYYY-MM-DD HH24
----------------------------------------------------
2004-02-07 15:44:48 下午 星期六
```

说明：该语句中的第一个参数表示要转换的日期，第二个参数是格式字符串，表示转换后的格式，结果类型为字符串。“YYYY”为4位的年份，“MM”为两位的月份，“DD”为两位的日期，“HH24”表示显示24小时制的小时,“MI”表示显示分钟，“SS”表示显示秒，“AM”表示显示上午或下午(本例中为下午)，“DY”表示显示星期。“-”、“:”和空格原样显示，用于分割日期和时间。转换出来的系统时间为：2004年2月7日(星期六)下午15点44分48秒。

还可以按其他的格式显示。以下查询中插入中文的年月日，其中原样显示部分区别于外层的单引号，需要用双引号引起。

【训练4】 将日期显示转换成中文的年月日。

输入并执行查询：

```
SELECT  TO_CHAR(sysdate,'YYYY" 年 "MM" 月 "DD" 日 "')
FROM dual;
```

执行结果为：

```
TO_CHAR(SYSDAT
----------------------
2003年11月18日
```

说明：双引号中的中文字“年”、“月”、“日”原样显示，单引号为字符串的界定标记，区别于双引号，不能混淆。

【训练5】 将雇佣日期转换成字符串并按新格式显示。

输入并执行查询：

```
SELECT ename, to_char(hiredate, 'DD Month YYYY') HIREDATE
FROM emp;
```

执行结果为：

```
ENAME      HIREDATE
---------- --------------------
SMITH      17 12月 1980
ALLEN      20 2月  1981
```

说明：Month表示月份的特殊格式，如“12月”。年度用4位显示。

对于数字型的日期格式，可以用数字或全拼格式显示，即在格式字符后面添加TH或SP。TH代表序列，SP代表全拼。

【训练6】 以全拼和序列显示时间。

执行以下查询：

```
SELECT
SYSDATE,to_char(SYSDATE,'yyyysp'),to_char(SYSDATE,'mmspth'),
to_char(SYSDATE,'ddth') FROM dual;
```

执行结果为：

```
SYSDATE   TO_CHAR(SYSDATE,'YYYYSP')
TO_CHAR( TO_C
------------ ---------------------------------------------------- ----------
---- --------
```

07-2月 -04 two thousand four second 07th

说明：“two thousand four”为全拼表示的2004年；“second”为全拼序列表示的2月；“07th”为用序列表示的7号。

在格式字符中，前两个字符代表显示结果的大小写。如果格式中的前两个字符是大写，则输出结果的全拼也为大写。如果格式中的前两个字符是小写，则输出结果的全拼也为小写。如果格式中的前两个字符的第一个字符是大写，第二个字符是小写，则输出结果的全拼也为大写开头，后面为字符小写。

【训练7】 时间显示的大小写。

步骤1：执行以下查询：

```
SELECT SYSDATE,to_char(SYSDATE,'yyyysp') FROM dual;
```

结果为：

```
SYSDATE   TO_CHAR(SYSDATE,'YYYYSP')
---------- ----------------------------------------
07-2月 -04  two thousand four
```

步骤2：执行以下查询：

```
SELECT to_char(SYSDATE,'Yyyysp') FROM dual;
```

结果为：

```
SYSDATE    TO_CHAR(SYSDATE,'YYYYSP')
---------- ----------------------------------------
Two Thousand Four
```

步骤3：执行以下查询：

```
SELECT SYSDATE,to_char(SYSDATE,'YYyysp') FROM dual;
```

结果为：

```
SYSDATE    TO_CHAR(SYSDATE,'YYYYSP')
---------- ----------------------------------------
TWO THOUSAND FOUR
```

说明：步骤1输出全拼小写的年度，步骤2输出全拼的以大写开头的年度，步骤3输出全拼大写的年度。

【练习1】显示2008年的8月8日为星期几。

3．数字类型转换

将数字型转换成字符串时，也可以按新的格式显示。格式字符含义如表2-10所示。

表2-10 数值转换符

代　码	代表的格式	例　子
9	代表一位数字，如果是正数，前面是空格，如果是负数，前面是-号	9999
0	代表一位数字，在相应的位置上如果没有数字则出现 0	0000
,	逗号，用作组分隔符	99,999
.	小数点，分隔整数和小数	999.9
$	$货币符号	$999.9
L	本地货币符号	L999.99
FM	去掉前后的空格	FM999.99
EEEE	科学计数法	9.9EEEE
S	负数符号–放在开头	S999.9

【训练8】 将数值转换成字符串并按新格式显示。

执行以下查询：

```
SELECT TO_CHAR(123.45,'0000.00'),
TO_CHAR(12345,'L9.9EEEE') FROM dual;
```

结果为：

```
TO_CHAR( TO_CHAR(12345,'L9.9
---------- ------------------------------
0123.45    RMB1.2E+04
```

说明：格式字符串中“0”表示一位数字，转换结果中相应的位置上没有数字则添加0。“.”表示在相应的位置上显示小数点。“L”将以本地货币符号显示于数字前，在本例中本地货币符号为“RMB”。“EEEE”将显示转换为科学计数法。

【训练9】 将数值转换成字符串并按新格式显示。

执行以下查询：

```
SELECT TO_CHAR(sal,'$99,999') SALARY FROM emp
WHERE ename = 'SCOTT';
```

结果为：

```
SALARY
-----------
 $4,000
```

说明：格式字符串中“$”表示转换结果前面添加$。“9”表示一位数字，“99，999”表示结果可以显示为5位的数字。“,”表示在相应的位置上添加逗号。如果实际数值位数不足5位，则只显示实际位数，如4000实际位数为4位，则只显示4位。如果实际位数超过5位，则会填充为#号。

2.4.5 其他函数

Oracle还有一些函数，如decode和nvl，这些函数也很有用，归纳如表2-11所示。

表2-11 其他常用函数

函 数	功 能	实 例	结 果
nvl	空值转换函数	nvl(null, '空')	空
decode	实现分支功能	decode(1,1,'男',2,'女')	男
userenv	返回环境信息	userenv('LANGUAGE')	SIMPLIFIED CHINESE_CHINA.ZHS16GBK
greatest	返回参数的最大值	greatest(20,35,18,9)	35
least	返回参数的最小值	least(20,35,18,9)	9

1．空值的转换

如果对空值NULL不能很好的处理，就会在查询中出现一些问题。在一个空值上进行算术运算的结果都是NULL。最典型的例子是，在查询雇员表时，将工资sal字段和津贴字段comm进行相加，如果津贴为空，则相加结果也为空，这样容易引起误解。

使用nvl函数，可以转换NULL为实际值。该函数判断字段的内容，如果不为空，返回原值；为空，则返回给定的值。

如下3个函数，分别用新内容代替字段的空值：

nvl(comm, 0)：用0代替空的Comm值。

nvl(hiredate, '01-1月-97')：用1997年1月1日代替空的雇佣日期。

nvl(job, '无')：用“无”代替空的职务。

【训练1】 使用nvl函数转换空值。

执行以下查询：

```
SELECT ename,nvl(job,'无'),nvl(hiredate,'01-1月-97'),nvl(comm,0) FROM emp;
```

结果为：

```
ENAME      NVL(JOB,'N NVL(HIREDA NVL(COMM,0)
---------- ---------- ---------- ----------
SMITH      CLERK      17-12月-80          0
ALLEN      SALESMAN   20-2月 -81        300
```

说明：本例中，空日期将显示为“01-1月-97”，空职务显示为“无”，空津贴将显示为0。

2．decode函数

decode函数可以通过比较进行内容的转换，完成的功能相当于分支语句。该函数的第一个参数为要进行转换的表达式，以后的参数成对出现，最后一个参数可以单独出现。如果第一个参数的值与第二个表达式的值相等，则返回第三个表达式的值；如果不等则继续比较，如果它的值与第四个表达式的值相等，则返回第五个表达式的值，以此类推。在参数的最后位置上可以存在单独的参数，如果以上比较过程没有找到匹配值，则返回该参数的值，如果不存在该参数，则返回NULL。

【训练2】 将职务转换成中文显示。

执行以下查询：

```
SELECT ename,decode(job, 'MANAGER', '经理',
'CLERK','职员', 'SALESMAN','推销员', 'ANALYST','系统分析员','
未知') FROM emp;
```

结果为：

```
ENAME      DECODE(JOB
---------- ----------------------
SMITH      职员
ALLEN      推销员
WARD       推销员
JONES      经理
```

MARTIN	推销员
BLAKE	经理
CLARK	经理
SCOTT	系统分析员
KING	未知
TURNER	推销员
ADAMS	职员
JAMES	职员
FORD	系统分析员
MILLER	职员

已选择14行。

说明：在以上训练中，如果job字段的内容为“MANAGER”则返回“经理”，如果是“CLERK”则返回“职员”，以此类推。如果不是“MANAGER”、“CLERK”、“SALESMAN”和“ANALYST”之一，则返回“未知”，如KING的职务“PRESIDENT”不在上述范围，返回“未知”。

【练习1】对部门表的部门名称和城市名进行转换。

3．userenv函数

函数userenv返回用户环境信息字符串，该函数只有一个字符串类型的参数，参数的内容为如下之一的字符串，可以不区分大小写：

* ISDBA：判断会话用户的角色是否为SYSDBA，是则返回TRUE。

* INSTANCE：返回会话连接的INSTANCE标识符。

* LANGUAGE：返回语言、地区、数据库字符集信息。

* LANG：返回会话语言的ISO简称。

* TERMINAL：返回正在会话的终端或计算机的标识符。

【训练3】 返回用户终端或系统标识信息。

执行以下查询：

```
SELECT        userenv('TERMINAL') FROM dual;
```

结果为：

```
ORASERVER
```

说明：根据用户使用的机器不同返回的信息不同，在本例中机器标识符ORASERVER为主机的名称。

【训练4】 返回语言、地区、数据库字符集信息。

执行以下查询：

```
SELECT        userenv('LANGUAGE') FROM dual;
```

结果为：

```
SIMPLIFIED CHINESE_CHINA.ZHS16GBK
```

说明：显示当前用户的语言为简体中文(SIMPLIFIED CHINESE)，地区为中国(CHINA)，字符集为ZHS16GBK。

【练习2】判断用户的角色是否为SYSDBA。

4．最大、最小值函数

greatest返回参数列表中的最大值，least返回参数列表中的最小值。

这两个函数的参数是一个表达式列表，按表达式列表中的第一个表达式的类型对求值后的表达式求得最大或最小值。对字符的比较按ASCII码的顺序进行。如果表达式中有NULL，则返回NULL。

【训练5】 比较字符串的大小，返回最大值。

执行以下查询：

```
SELECT          greatest('ABC','ABD','abc', 'abd') FROM dual;
```

执行结果为：

```
GRE
------
abd
```

说明：在上述四个字符串中，大小关系为abd>abc>ABD>ABC。在ASCII码表中，排在后边的字符大，小写字母排在大写字母之后。字符串的比较原则是，先比较第一位，如果相同，则继续比较第二位，依此类推，直到出现大小关系。

2.5 高级查询

2.5.1 多表联合查询

通过连接可以建立多表查询，多表查询的数据可以来自多个表，但是表之间必须有适当的连接条件。为了从多张表中查询，必须识别连接多张表的公共列。一般是在WHERE子句中用比较运算符指明连接的条件。

忘记说明表的连接条件是常见的一种错误，这时查询将会产生表连接的笛卡尔积(即一个表中的每条记录与另一个表中的每条记录作连接产生的结果)。一般N个表进行连接，需要至少N-1个连接条件，才能够正确连接。两个表连接是最常见的情况，只需要说明一个连接条件。

两个以上的表也可以进行连接，在这里不做专门介绍。

两个表的连接有四种连接方式：

* 相等连接。

* 不等连接。

* 外连接。

* 自连接。

1．相等连接

通过两个表具有相同意义的列，可以建立相等连接条件。使用相等连接进行两个表的查询时，只有连接列上在两个表中都出现且值相等的行才会出现在查询结果中。

在下面的练习中，将显示雇员姓名、所在的部门编号和名称，在雇员表emp中是没有雇员的部门名称信息的，只有雇员所在部门的编号，部门的信息在另外的部门表dept中，两个表具有相同的部门编号列deptno，可以通过该列建立相等连接。

【训练1】 显示雇员的名称和所在的部门的编号和名称。

执行以下查询：

```
SELECT emp.ename,emp.deptno,dept.dname FROM emp,dept
WHERE emp.deptno=dept.deptno;
```

执行结果如下：

```
ENAME      DEPTNO DNAME
---------- ---------------------- - ----------
SMITH          20          RESEARCH
ALLEN          30          SALES
```

说明：相等连接语句的格式要求是，在FROM从句中依次列出两个表的名称，在表的每个列前需要添加表名，用“.”分隔，表示列属于不同的表。在WHERE条件中要指明进行相等连接的列。

以上训练中，不在两个表中同时出现的列，前面的表名前缀可以省略。所以以上例子可以简化为如下的表示：

```
SELECT ename,emp.deptno,dname FROM emp,dept
WHERE emp.deptno=dept.deptno;
```

【练习1】省略表前缀，察看执行结果。

【练习2】执行以下查询(省略表连接条件)并察看执行结果中共有多少记录产生。

SELECT ename,emp.deptno,dname FROM emp,dept

如果表名很长，可以为表起一个别名，进行简化，别名跟在表名之后，用空格分隔。

【训练2】 使用表别名。

执行以下查询：

SELECT ename,e.deptno,dname FROM emp e,dept d

WHERE e.deptno=d.deptno;

执行结果同上。

说明：emp表的别名为e，dept表的别名为d。

相等连接还可以附加其他的限定条件。

【训练3】 显示工资大于3000的雇员的名称、工资和所在的部门名称。

执行以下查询：

```
SELECT ename,sal,dname FROM emp,dept
WHERE emp.deptno=dept.deptno AND sal>3000;
```

显示结果为：

```
ENAME          SAL DNAME
-------------- ---------- --------------
KING           5000 ACCOUNTING
```

说明：只显示工资大于3000的雇员的名称、工资和部门名称。在相等连接的条件下增加了工资大于3000的条件。增加的条件用AND连接。

2．外连接

在以上的例子中，相等连接有一个问题：如果某个雇员的部门还没有填写，即保留为空，那么该雇员在查询中就不会出现；或者某个部门还没有雇员，该部门在查询中也不会出现。

为了解决这个问题可以用外连，即除了显示满足相等连接条件的记录外，还显示那些不满足连接条件的行，不满足连接条件的行将显示在最后。外连操作符为(+)，它可以出现在相等连接条件的左侧或右侧。出现在左侧或右侧的含义不同，这里用如下的例子予以说明。

【训练4】 使用外连显示不满足相等条件的记录。

步骤1：显示雇员的名称、工资和所在的部门名称及没有任何雇员的部门。

执行以下查询：

```
SELECT ename,sal,dname FROM emp,dept
WHERE emp.deptno(+)=dept.deptno;
```

执行结果为：

```
ENAME          SAL DNAME
---------- ---------- ----------
```

```
SELECT ename,sal,dname FROM emp
right outer join dept on emp.deptno = dept.deptno;
```

CLARK	2450 ACCOUNTING
KING	5000 ACCOUNTING
MILLER	1300 ACCOUNTING
SMITH	800 RESEARCH
ADAMS	1100 RESEARCH
FORD	3000 RESEARCH
SCOTT	3000 RESEARCH
JONES	2975 RESEARCH
ALLEN	1600 SALES

BLAKE 2850 SALES

MARTIN 1250 SALES

JAMES 950 SALES

TURNER 1500 SALES

WARD 1250 SALES

OPERATIONS

步骤2：显示雇员的名称、工资和所在的部门名称及没有属于任何部门的雇员。

执行以下查询：

```
SELECT ename,sal,dname FROM emp,dept
WHERE emp.deptno=dept.deptno(+);
```

结果从略，请自行执行观察结果。

说明：部门OPERATION没有任何雇员。查询结果通过外连显示出该部门。

```
SELECT ename,sal,dname FROM emp
 left outer join dept on emp.deptno = dept.deptno;
```

3．不等连接

还可以进行不等的连接。以下是一个训练实例，其中用到的salgrade表的结构如下：

DESC salgrade

名称	是否为空	类型
GRADE		NUMBER
LOSAL		NUMBER
HISAL		NUMBER

Grade 表示工资等级，losal和hisal分别表示某等级工资的下限和上限。

表的内容为：

SELECT * FROM salgrade;

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

【训练5】 显示雇员名称，工资和所属工资等级。

执行以下查询：

```
SELECT e.ename, e.sal, s.grade FROM emp e,salgrade s
WHERE          e.sal BETWEEN s.losal AND s.hisal;
```

执行结果为：

```
ENAME          SAL      GRADE
---------- ---------- ----------
```

SMITH	800	1
ADAMS	1100	1
JAMES	950	1
WARD	1250	2
MARTIN	1250	2
MILLER	1300	2
ALLEN	1600	3
TURNER	1500	3

JONES	2975	4	
BLAKE	2850	4	
CLARK	2450	4	
SCOTT	3000	4	
FORD	3000	4	
KING	5000	5	

说明：通过将雇员工资与不同的工资上下限范围相比较，取得工资的等级，并在查询结果中显示出雇员的工资等级。

4. 自连接

最后是一个自连接的训练实例，自连接就是一个表，同本身进行连接。对于自连接可以想像存在两个相同的表(表和表的副本)，可以通过不同的别名区别两个相同的表。

【训练6】 显示雇员名称和雇员的经理名称。

执行以下查询：

```
SELECT worker.ename||' 的经理是 '||manager.ename AS 雇员经理
FROM emp worker, emp manager
WHERE worker.mgr = manager.empno;
```

执行结果为：

```
雇员经理
```

SMITH 的经理是 FORD

ALLEN 的经理是 BLAKE

WARD 的经理是 BLAKE

说明：为EMP表分别起了两个别名worker和manager，可以想像，第一个表是雇员表，第二个表是经理表，因为经理也是雇员。然后通过worker表的mgr(经理编号)字段同manager表的empno(雇员编号)字段建立连接，这样就可以显示雇员的经理名称了。

注意：经理编号mgr是雇员编号empno之一，所以经理编号可以同雇员编号建立连接。

2.5.2 统计查询

通常需要对数据进行统计，汇总出数据库的统计信息。比如，我们可能想了解公司的总人数和总工资额，或各个部门的人数和工资额，这个功能可以由统计查询完成。

Oracle提供了一些函数来完成统计工作，这些函数称为组函数，组函数不同于前面介绍和使用的函数(单行函数)。组函数可以对分组的数据进行求和、求平均值等运算。组函数只能应用于SELECT子句、HAVING子句或ORDER BY子句中。组函数也可以称为统计函数。

查询公司的总人数需要对整个表应用组函数；查询各个部门的人数，需要对数据进行分组，然后应用组函数进行运算。

常用的组函数如表2-12所示。

表2-12 常用的组函数

函 数	说 明
AVG	求平均值
COUNT	求计数值，返回非空行数，*表示返回所有行
MAX	求最大值
MIN	求最小值
SUM	求和
STDDEV	求标准偏差，是根据差的平方根得到的
VARIANCE	求统计方差

分组函数中SUM和AVG只应用于数值型的列，MAX、MIN和COUNT可以应用于字符、数值和日期类型的列。组函数忽略列的空值。

使用GROUP BY 从句可以对数据进行分组。所谓分组，就是按照列的相同内容，将记录划分成组，对组可以应用组函数。

如果不使用分组，将对整个表或满足条件的记录应用组函数。

在组函数中可使用DISTINCT或ALL关键字。ALL表示对所有非NULL值(可重复)进行运算(COUNT除外)。DISTINCT 表示对每一个非NULL值，如果存在重复值，则组函数只运算一次。如果不指明上述关键字，默认为ALL。

1．统计查询

【训练1】 求雇员总人数。

执行以下查询：

```
SELECT COUNT(*) FROM emp;
```

返回结果为：

```
COUNT(*)
----------
        14
```

说明：该实例中，因为没有WHERE条件，所以对emp表的全部记录应用组函数。使用组函数COUNT统计记录个数，即雇员人数，返回结果为14，代表有14个记录。

注意：*代表返回所有行数，否则返回非NULL行数。

【训练2】 求有佣金的雇员人数。

执行以下查询：

```
SELECT COUNT(comm) FROM emp;
```

返回结果为：

```
COUNT(COMM)
-------------------
                  4
```

说明：在本例中，没有返回全部雇员，只返回佣金非空的雇员，只有4个人。

【训练3】 求部门10的雇员的平均工资。

执行以下查询：

```
SELECT AVG(sal) FROM emp WHERE deptno=10;
```

返回结果为：

```
  AVG(SAL)
----------
2916.66667
```

说明：增加了WHERE条件，WHERE条件先执行，结果只对部门10的雇员使用组函数AVG求平均工资。

最大值和最小值函数可以应用于日期型数据，以下是训练实例。

【训练4】 求最晚和最早雇佣的雇员的雇佣日期。

执行以下查询：

```
SELECT MAX(hiredate),MIN(hiredate) FROM emp;
```

返回结果为：

```
MAX(HIREDA MIN(HIREDA
---------- ----------
23-5月     -87 17-12月-80
```

说明：最晚雇员雇佣的时间为87年5月23日，最早雇员雇佣的时间为80年12月17日。

【训练5】 求雇员表中不同职务的个数。

执行以下查询：

```
SELECT COUNT( DISTINCT job) FROM emp;
```

返回结果为：

```
COUNT(DISTINCT JOB)
-------------------
                  5
```

说明：该查询返回雇员表中不同职务的个数。如果不加DISTINCT，则返回的是职务非空的雇员个数。

【练习1】求部门10中工资大于1500的雇员人数。

2．分组统计

通过下面的训练，我们来了解分组的用法。

【训练6】 按职务统计工资总和。

步骤1：执行以下查询：

```
SELECT SUM(sal) FROM emp GROUP BY job;
```

执行结果为：

```
    SUM(SAL)
----------
      6000
      4150
      8275
      5000
      5600
```

步骤2：执行以下查询：

```
SELECT job,SUM(sal) FROM emp GROUP BY job;
```

执行结果为：

```
JOB       SUM(SAL)
--------- ---------
ANALYST       6000
CLERK         4150
MANAGER       8275
PRESIDENT     5000
SALESMAN      5600
```

说明：步骤1按职务对雇员进行分组，有多少种职务就会返回多少行结果，相同职务的工资被汇总到一起，其中使用到了SUM函数对分组后的工资进行求和。以上查询结果没有显示分组后的职务。分组查询允许在查询列表中包含分组列，对以上实例，因为是按职务job分组的，所以在查询列中可以包含job字段，使统计结果很清楚，如步骤2所示。

职务为ANALYST的雇员的总工资为6000，职务为CLERK的雇员的总工资为4150，依此类推。

注意：在查询列中，不能使用分组列以外的其他列，否则会产生错误信息。

【练习2】查看以下查询的显示结果，并解释原因。

SELECT ename,job,SUM(sal) FROM emp GROUP BY job;

3．多列分组统计

可以按多列进行分组，以下是按两列进行分组的例子。

【训练7】 按部门和职务分组统计工资总和。

执行以下查询：

```
SELECT   deptno, job, sum(sal) FROM emp
GROUP BY deptno, job;
```

执行结果为：

```
   DEPTNO JOB         SUM(SAL)
---------- --------- ----------
```

10	CLERK	1300
10	MANAGER	2450
10	PRESIDENT	5000
20	ANALYST	6000
20	CLERK	1900
20	MANAGER	2975
30	CLERK	950
30	MANAGER	2850
30	SALESMAN	5600

说明：该查询统计每个部门中每种职务的总工资。

4．分组统计结果限定

对分组查询的结果进行过滤，要使用HAVING从句。HAVING从句过滤分组后的结果，它只能出现在GROUP BY从句之后，而WHERE从句要出现在GROUP BY从句之前。

【训练8】 统计各部门的最高工资，排除最高工资小于3000的部门。

执行以下查询：

```
SELECT   deptno, max(sal) FROM emp
GROUP BY deptno
HAVING   max(sal)>=3000;
```

执行结果为：

```
DEPTNO   MAX(SAL)
---------- ----------
        10       5000
        20       3000
```

说明：结果中排除了部门30，因部门30的总工资小于3000。

注意：HAVING从句的限定条件中要出现组函数。如果同时使用WHERE条件，则WHERE条件在分组之前执行，HAVING条件在分组后执行。

【练习3】统计人数小于4的部门的平均工资。

5．分组统计结果排序

可以使用ORDER BY从句对统计的结果进行排序，ORDER BY从句要出现在语句的最后。

【训练9】 按职务统计工资总和并排序。

执行以下查询：

```
SELECT job 职务, SUM(sal) 工资总和 FROM emp
GROUP BY job
ORDER BY SUM(sal);
```

执行结果为：

职务　　工资总和

---------- ----------

CLERK	4150
PRESIDENT	5000
SALESMAN	5600
ANALYST	6000
MANAGER	8275

注意：排序使用的是计算列SUM(sal)，也可以使用别名，写成：

```
SELECT job 职务, SUM(sal) 工资总和 FROM emp
GROUP BY job
ORDER BY 工资总和;
```

【练习4】统计各部门的人数，按平均工资排序。

6．组函数的嵌套使用

在如下训练中，使用了组函数的嵌套。

【训练10】 求各部门平均工资的最高值。

执行以下查询：

```
SELECT max(avg(sal)) FROM emp GROUP BY deptno;
```

执行结果为：

```
MAX(AVG(SAL))
---------------------
2916.66667
```

说明：该查询先统计各部门的平均工资，然后求得其中的最大值。

注意：虽然在查询中有分组列，但在查询字段中不能出现分组列。如下的查询是错误的：

SELECT deptno,max(avg(sal)) FROM emp GROUP BY deptno;

因为各部门平均工资的最高值不应该属于某个部门。

【练习5】求每种职务总工资的最低值。

2.5.3 子查询

我们可能会提出这样的问题，在雇员中谁的工资最高，或者谁的工资比SCOTT高。通过把一个查询的结果作为另一个查询的一部分，可以实现这样的查询功能。具体的讲：要查询工资高于SCOTT的雇员的名字和工资，必须通过两个步骤来完成，第一步查询雇员SCOTT的工资，第二步查询工资高于SCOTT的雇员。第一个查询可以作为第二个查询的一部分出现在第二个查询的条件中，这就是子查询。出现在其他查询中的查询称为子查询，包含其他查询的查询称为主查询。

子查询一般出现在SELECT语句的WHERE子句中，Oracle也支持在FROM或HAVING子句中出现子查询。子查询比主查询先执行，结果作为主查询的条件，在书写上要用圆括号扩起来，并放在比较运算符的右侧。子查询可以嵌套使用，最里层的查询最先执行。子查询可以在SELECT、INSERT、UPDATE、DELETE等语句中使用。

子查询按照返回数据的类型可以分为单行子查询、多行子查询和多列子查询。

1．单行子查询

【训练1】 查询比SCOTT工资高的雇员名字和工资。

执行以下查询：

```
SELECT ename,sal FROM emp
WHERE sal>(SELECT sal FROM emp WHERE empno=7788);
```

执行结果为：

```
ENAME          SAL
---------- ----------
KING           5000
```

说明：在该子查询中查询SCOTT的工资时使用的是他的雇员号，这是因为雇员号在表中是惟一的，而雇员的姓名有可能相重。SCOTT的雇员号为7788。

下面的训练实例包含两个子查询。

【训练2】 查询和SCOTT同一部门且比他工资低的雇员名字和工资。

执行以下查询：

```
SELECT ename,sal FROM emp
WHERE sal<(SELECT sal FROM emp WHERE
  empno=7788)
AND deptno=(SELECT deptno FROM emp WHERE
  empno=7788);
```

执行结果为：

```
ENAME          SAL
---------- ----------
SMITH           800
JONES          2975
ADAMS          1100
```

说明：两个子查询出现在两个条件中，用AND连接表示需要同时满足。在子查询中也可以使用组函数。

【训练3】 查询工资高于平均工资的雇员名字和工资。

执行以下查询：

```
SELECT ename,sal FROM emp
WHERE sal>(SELECT AVG(sal) FROM emp);
```

执行结果为：

```
ENAME          SAL
```

---------- ----------

JONES 2975

BLAKE 2850

CLARK 2450

SCOTT 3000

KING 5000

FORD 3000

说明：在子查询中出现了组函数。由执行结果可知，在14个雇员中，大于平均工资的有6个。

【练习1】查询工资最高的雇员名字和工资。

2．多行子查询

如果子查询返回多行的结果，则我们称它为多行子查询。多行子查询要使用不同的比较运算符号，它们是IN、ANY和ALL。

【训练4】 查询工资低于任何一个“CLERK”的工资的雇员信息。

执行以下查询：

```
SELECT  empno, ename, job,sal FROM emp
WHERE    sal < ANY (SELECT sal FROM emp WHERE job = 'CLERK')
AND job <> 'CLERK';
```

执行结果为：

```
EMPNO ENAME      JOB              SAL
---------- ---------- --------------- ---------------------
 7521 WARD       SALESMAN         1250
                7654 MARTIN     SALESMAN         1250
```

说明：在emp表的雇员中有4个职务为“CLERK”，他们的工资分别是800、1100、950、1300。满足工资小于任何一个“CLERK”的工资的记录有2个，在这里使用了ANY运算符表示小于子查询中的任何一个工资。

注意：条件job <> 'CLERK'排除了职务是CLERK的雇员本身。

【训练5】 查询工资比所有的“SALESMAN”都高的雇员的编号、名字和工资。

执行以下查询：

```
SELECT  empno, ename,sal FROM emp
WHERE sal > ALL(SELECT sal FROM emp WHERE job= 'SALESMAN');
```

执行结果为：

```
 EMPNO ENAME          SAL
-------------- ------------ --------------------
7566 JONES                2975
7698 BLAKE                2850
7782 CLARK                2450
7788 SCOTT                3000
7839 KING                 5000
7902 FORD                 3000
```

说明：在emp表的雇员中有4个职务为“SALESMAN”，他们的工资分别是1600、1250、1250、1500。在这里使用了ALL运算符，表示大于查询中所有的工资。

【训练6】 查询部门20中职务同部门10的雇员一样的雇员信息。

执行以下查询：

```
SELECT  empno, ename, job FROM emp
WHERE    job  IN  (SELECT  job  FROM  emp  WHERE
  deptno=10)
AND deptno =20;
```

执行结果为：

```
     EMPNO ENAME      JOB
---------- ---------- ---------
      7369 SMITH      CLERK
      7876 ADAMS           CLERK
      7566 JONES      MANAGER
```

说明：在该训练中，使用IN运算符表示职务是子查询结果中的任何一个。部门10中有3种职务：MANAGER、PRESIDENT和CLERK，以上查询得到的是部门20中是这3种职务的雇员。

【训练7】 查询职务和SCOTT相同，比SCOTT雇佣时间早的雇员信息。

执行以下查询：

```
SELECT  empno, ename, job FROM emp
WHERE job =(SELECT job FROM emp WHERE empno=7788)
AND hiredate < (SELECT hiredate FROM emp WHERE
  empno=7788);
```

执行结果为：

```
    EMPNO ENAME      JOB
---------- ---------- ---------
      7902 FORD       ANALYST
```

说明：在查询中用到了时间的比较。

【练习2】查询工资比SCOTT高或者雇佣时间比SCOTT早的雇员的编号和名字。

3．多列子查询

如果子查询返回多列，则对应的比较条件中也应该出现多列，这种查询称为多列子查询。以下是多列子查询的训练实例。

【训练8】 查询职务和部门与SCOTT相同的雇员的信息。

执行以下查询：

```
SELECT  empno, ename, sal FROM emp
WHERE (job,deptno) =(SELECT job,deptno FROM emp
WHERE empno=7788);
```

执行结果为：

```
EMPNO ENAME      JOB
---------- ---------- ---------
      7902 FORD       ANALYST
```

说明：在该例的子查询中返回两列，查询条件中也要出现两列，表示雇员的职务和部门应该和SCOTT的职务和部门相同。

4．在FROM从句中使用子查询

在FROM从句中也可以使用子查询，在原理上这与在WHERE条件中使用子查询类似。有的时候我们可能要求从雇员表中按照雇员出现的位置来检索雇员，很容易想到的是使用rownum虚列。比如我们要求显示雇员表中6～9位置上的雇员，可以用以下方法。

【训练9】 查询雇员表中排在第6～9位置上的雇员。

执行以下查询：

```
SELECT ename,sal FROM (SELECT rownum as num,ename,sal
FROM emp WHERE rownum<=9 )
WHERE num>=6;
```

执行结果为：

```
ENAME          SAL
-------------- --------------------
BLAKE          2850
CLARK          2450
SCOTT          3000
KING           5000
```

说明：子查询出现在FROM从句中，检索出行号小于等于9的雇员，并生成num编号列。在主查询中检索行号大于等于6的雇员。

注意：以下用法不会有查询结果，请自行分析原因。

```
SELECT ename,sal FROM emp
WHERE rownum>=6 AND rownum<=9;
```

【练习3】查询雇员表中的第6个雇员。

2.5.4 集合运算*

多个查询语句的结果可以做集合运算，结果集的字段类型、数量和顺序应该一样。

Oracle共有4个集合操作，如表2-13所示。

表2-13 集合运算操作

操　作	描　　述
UNION	并集，合并两个操作的结果，去掉重复的部分
UNION ALL	并集，合并两个操作的结果，保留重复的部分
MINUS	差集，从前面的操作结果中去掉与后面操作结果相同的部分
INTERSECT	交集，取两个操作结果中相同的部分

1．使用集合的并运算

【训练1】 查询部门10和部门20的所有职务。

执行以下查询：

```
SELECT  job FROM emp WHERE deptno=10
UNION
SELECT  job FROM emp WHERE deptno=20;
```

执行结果为：

```
JOB
---------
ANALYST
CLERK
MANAGER
PRESIDENT
```

说明：部门10的职务有PRESIDENT、MANAGER、CLERK；部门20的职务有MANAGER、CLERK、ANALYST。所以两个部门的所有职务(相同职务只算一个)共有4个：ANALYST、CLERK、MANAGER和PRESIDENT。可以将UNION改为UNION ALL查看一下结果。

2．使用集合的交运算

【训练2】 查询部门10和20中是否有相同的职务和工资。

执行以下查询：

```
SELECT  job,sal FROM emp WHERE deptno=10
INTERSECT
SELECT  job,sal FROM emp WHERE deptno=20;
```

执行结果为：

未选定行

说明：部门10的职务有PRESIDENT、MANAGER、CLERK；部门20的职务有MANAGER、CLERK、ANALYST。所以两个部门的相同职务为：CLERK和MANAGER。但是职务和工资都相同的雇员没有，所以没有结果。

3．使用集合的差运算

【训练3】 查询只在部门表中出现，但没有在雇员表中出现的部门编号。

执行以下查询：

```
SELECT  deptno FROM dept
MINUS
SELECT  deptno FROM emp ;
```

执行结果为：

```
   DEPTNO
----------
        40
```

说明：部门表中的部门编号有10、20、30和40。雇员表中的部门编号有10、20和30。差集的结果为40。

【练习1】查询具有职务CLERK和SALESMAN的所有部门编号。

【练习2】试求部门10和20中不相同的职务(即部门10中有、部门20中没有和部门20中有、部门10中没有的职务)。

2.6 阶段训练

【训练1】 显示人数最多的部门名称。

输入并执行以下查询：

SELECT DECODE(dname,'SALES','销售部','ACCOUNTING','财务部','RESEARCH','研发部','未知')

部门名

FROM emp,dept

```
WHERE emp.deptno=dept.deptno
GROUP BY dname
HAVING COUNT(*)=(SELECT MAX(COUNT(*)) FROM emp GROUP BY deptno);
```

执行结果：

```
部门名
---------
销售部
```

说明：本训练使用了分组统计、相等连接和子查询，使用了DECODE函数进行部门名称转换。

【训练2】 显示各部门的平均工资、最高工资、最低工资和总工资列表，并按平均工资高低顺序排序。

输入并执行以下查询：

```
SELECT dname 部门,AVG(sal) 平均工资,MAX(sal) 最高工资,MIN(sal) 最低工资,SUM(sal) 总工资
FROM emp,dept
WHERE emp.deptno=dept.deptno
GROUP BY dname
ORDER BY AVG(sal) DESC;
```

执行结果为

部门　　平均工资　最高工资　最低工资　总工资

---------- ---------- ---------- ---------- ----------

ACCOUNTING　2916.66667 5000 1300　　8750

RESEARCH　2175　　3000　800　　10875

SALES　　1566.66667　2850　950　9400

说明：本训练使用了分组统计、相等连接和排序，使用相等连接可以通过部门编号获取部门名称。

2.7 练习

1．SQL语言中用来创建、删除及修改数据库对象的部分被称为：

A. 数据库控制语言(DCL)

B. 数据库定义语言(DDL)

C. 数据库操纵语言(DML)

D. 数据库事务处理语言

2．执行以下查询，表头的显示为：

SELECT sal "Employee Salary" FROM emp

A. EMPLOYEE SALARY　　B. employee salary

C. Employee Salary　　D. "Employee Salary“

3．执行如下两个查询，结果为：

SELECT ename name,sal salary FROM emp order by salary;

SELECT ename name,sal "SALARY" FROM emp order by sal ASC;

A. 两个查询结果完全相同

B. 两个查询结果不相同

C. 第一个查询正确，第二个查询错误

D. 第二个查询正确，第一个查询错误

4．参考本章的emp表的内容执行下列查询语句，出现在第一行上的人是：

SELECT ename FROM emp WHERE deptno=10 ORDER BY sal DESC;

A. SMITH　　B. KING

C. MILLER　　D. CLARK

5．哪个函数与||运算有相同的功能：

A. LTRIM B. CONCAT

C. SUBSTR D. INSTR

6．执行以下语句后，正确的结论是：

SELECT empno,ename FROM emp WHERE hiredate<to_date('04-11月-1980')-100

A. 显示给定日期后100天以内雇佣的雇员信息

B. 显示给定日期前100天以内雇佣的雇员信息

C. 显示给定日期100天以后雇佣的雇员信息

D. 显示给定日期100天以前雇佣的雇员信息

7. 执行以下语句出错的行是：

```
SELECT deptno,max(sal) FROM emp
WHERE job IN('CLERK','SALEMAN','ANALYST')
GROUP BY deptno
HAVING sal>1500;
```

A. 第一行　　B. 第二行

C. 第三行　　D. 第四行

8．执行以下语句出错的行是：

```
SELECT deptno,max(avg(sal))
FROM emp
WHERE sal>1000
Group by deptno;
```

A. 第一行　　B. 第二行

C. 第三行　　D. 第四行

9．执行以下语句出错的行是：

```
SELECT deptno,dname,ename,sal
FROM emp,dept
WHERE emp.deptno=dept.deptno
AND sal>1000;
```

A. 第一行　　B. 第二行

C. 第三行　　D. 第四行

10. 以下语句出错，哪种改动能够正确执行：

```
SELECT   deptno, max(sal)
FROM emp
GROUP BY deptno
WHERE   max(sal)>2500;
```

A. 将WHERE和GROUP BY 语句顺序调换一下

B. 将WHERE max(sal)>2500语句改成HAVING max(sal)>2500

C. 将WHERE max(sal)>2500语句改成WHERE sal>2500

D. 将WHERE max(sal)>2500语句改成HAVING sal>2500

11. 以下语句的作用是：

```
SELECT ename,sal FROM emp
WHERE sal<(SELECT min(sal) FROM emp)+1000;
```

A. 显示工资低于1000元的雇员信息

B. 将雇员工资小于1000元的工资增加1000后显示

C. 显示超过最低工资1000元的雇员信息

D. 显示不超过最低工资1000元的雇员信息

12. 以下语句的作用是：

```
SELECT  job FROM emp WHERE deptno=10
MINUS
SELECT  job FROM emp WHERE deptno=20;
```

A. 显示部门10的雇员职务和20的雇员职务

B. 显示部门10和部门20共同的雇员职务

C. 显示部门10和部门20不同的雇员职务

D. 显示在部门10中出现，在部门20中不出现的雇员职务