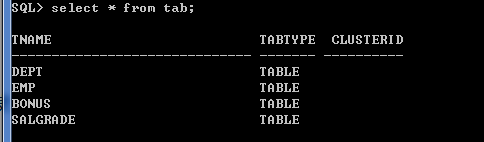
五、SQL语句基础:简单查询、限定查询

# scott用户的主要表结构

**查看当前用户所有的表：**

|  |
| --- |
| **SELECT \* FROM tab;** |

scott用户中的数据表有四张，如下图：

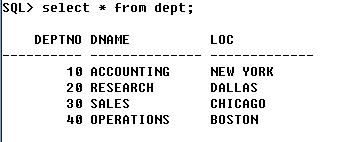


**查看一张表的结构：**

|  |
| --- |
| DESC 表名称； |

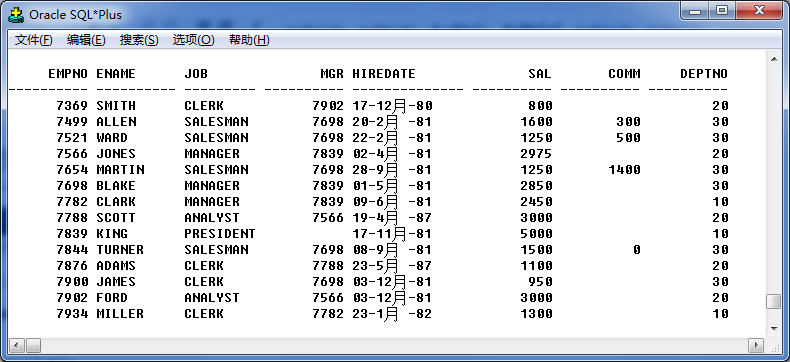
## DEPT表（部门表department）

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **名称** | **类型** | **描述** |
| 1 | DEPTNO | NOT NULL NUMBER(2) | 部门编号，有两位数字所组成 |
| 2 | DNAME | VARCHAR2(14) | 部门名称，最多由14个字符组成 |
| 3 | LOC | VARCHAR2(13) | 部门所在位置，最多由13个字符组成 |



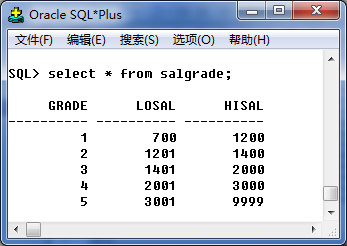
## 1.2 EMP表

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **名称** | **类型** | **描述** |
| 1 | EMPNO | NOT NULL  NUMBER(4) | 雇员编号（employee number） |
| 2 | ENAME | VARCHAR2(10) | 雇员姓名（employee name） |
| 3 | JOB | VARCHAR2(9) | 雇员职务 |
| 4 | MGR | NUMBER(4) | 所属领导（manager）编号 |
| 5 | HIREDATE | DATE | 雇用日期 |
| 6 | SAL | NUMBER(7,2) | 工资，最多有7位数字组成，其中有两位小数 |
| 7 | COMM | NUMBER(7,2) | 佣金（commission） |
| 8 | DEPTNO | NUMBER(2) | 所属部门编号（departmentnumber） |



## 1.3 SALGRADE表

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **名称** | **类型** | **描述** |
| 1 | GRADE | NUMBER | 工资等级 |
| 2 | LOSAL | NUMBER | 最低工资 |
| 3 | HISAL | NUMBER | 最高工资 |



## 1.4 BONUS表（奖金表）

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **名称** | **类型** | **描述** |
| 1 | ENAME | VARCHAR2 (10) | 雇员姓名 |
| 2 | JOB | VARCHAR2(9) | 职务 |
| 3 | SAL | NUMBER | 工资 |
| 4 | COMM | NUMBER | 佣金 |

# SQL语句基础

## 2.1 SQL语言简介

SQL(Structed Query Language)是一个功能强大的数据库查询语言，用以与数据库进行通信，是关系型数据库管理的标准语言。

SQL语言可以分为以下五类：

1. **DML（Data Manipulation Language，数据操作语言）**:用以更新数据（插入、删除、修改），用以检索数据；常用语句：INSERT, UPDATE, DELETE, MERGE;
2. **DDL(Data Definition Language，数据定义语言)**:用于定义数据结构，创建、修改或删除数据对象；常用语句：CREATE, ALTER, DROP, TRUNCATE
3. **DCL（Data Control Language，数据控制语言**）:用于管理数据库用户的权限；常用语句：GRANT, REVOKE;(对用户授权和收回权限)
4. **DQL（Data Query Language，数据查询语句）**：用于数据查询，即SELECT语句；
5. **事务控制语句**：用于控制事务操作，常用语句：COMMIT,ROLLBACK,SAVEPOINT;

## 2.2 运算符&基本操作：

### 2.2.1 算术运算符

* 算术运算符

**+ - \* /**

* 运算优先级
* 乘法和除法的优先级高于加法和减法;
* 同优先级运算的顺序是从左到右;
* 表达式中可使用小括号强行改变运算顺序;

示例：

|  |
| --- |
| SELECT empno, ename,sal, sal\*12 FROM emp;  SELECT empno, ename,sal, saL\*12 +1000 FROM emp;  SELECT empno, ename,sal, sal\*(12 +1000) FROM emp; |

### 2.2.2 连接运算符

连接运算符‘||’可以把列与字符、或其它表达式连接在一起，得到一个新的字符串，实现‘合成’列的功能。

示例：

|  |
| --- |
| SELECT ename || ' IS A ' || job FROM emp;  SELECT empno, ename || ' ''S ANNUAL SALARY IS ' || sal\*12 FROM emp; |

### 2.2.3使用字段别名

Oracle数据库中可以 重命名查询结果中的字段，以增强可读性，如果别名中使用特殊字符

或者是强制大小写敏感需使用双引号；

语法格式：

|  |
| --- |
| **SELECT <字段名> | <表达式> [ [AS] <字段别名>], ...**  **FROM <表名>;** |

示例：

|  |
| --- |
| SELECT empno AS 员工编号, ename 员工姓名, sal\*12 "年薪" FROM emp;  SELECT empno, ename "ENAME", sal\*12 "ANUAL SALARY" FROM emp; |

### 2.2.4空值

* 空值是无效的，未指定的，未知的或不可预知的值；
* 空值不等同于空格或者0或是长度为0的字符串；
* 在Oracle中，空值用null表示；
* 算术表达式中如果出现空值，则整个表达式结果为空；
* 连接表达式中出现的空值被当作一个空的（长度为零的）

字符串处理；

示例：

|  |
| --- |
| select empno, ename, sal, comm, ename||' - '||comm, sal+comm from emp; |
|  |

### 2.2.5 去除重复字段

使用DISTINCT关键字可从查询结果中清除重复行,DISTINCT的作用范围是后面所有字段的组合;

示例：

|  |
| --- |
| SELECT DISTINCT deptno, job FROM emp; |

### 2.2.6 比较运算符

|  |  |
| --- | --- |
| <> | 不等于 |
| <= | 小于等于 |
| < | 小于 |
| >= | 大于等于 |
| > | 大于 |
| = | 等于 |
| IS NULL | 进行空值判断 |
| LIKE | 模糊查询 |
| IN(set) | 出现在集合中 |
| BETWEEN ... AND ... | 界于两值之间（包括边界） |

**注：**对于常用的一些简单的数据操作，可以用数据库自带的方法或操作处理，但是如果是涉及到具体业务的数据操作要放到应用程序中去处理，这样可以降低数据库与应用程序之间的耦合程度；

### 2.2.7逻辑运算符

|  |  |
| --- | --- |
| NOT | 逻辑"非" |
| OR | 逻辑"或" |
| AND | 逻辑"与" |

示例：

|  |
| --- |
| SELECT \* FROM emp WHERE deptno=20 AND sal > 2000;  SELECT \* FROM emp WHERE deptno=20 OR sal > 2000;  SELECT \* FROM emp WHERE deptno NOT IN(20,30,40); |

### 2.2.8运算符优先级

9 OR

8 AND

7 NOT

6 [NOT] BETWEEN...AND

5 IS [NOT] NULL LIKE [NOT] IN

4 = > >= < <= <>

3 ||

2 + -

1 \* /

**注：**可使用小括号强行改变运算顺序

## 2.3 简单查询

语法

|  |
| --- |
| **SELECT [DISTNCT] \* |字段[别名] [字段[别名]…]**  **FROM 表名称[别名]；** |

**注1：**SQL语句规范，语法关键字部分用大写，表名、字段名等涉及数据名称的变量用小写；

**注 2：**查询语句中使用字符串和日期:

* 字符串和日期值要用单引号扩起来
* 字符串大小写敏感
* 日期值格式敏感，缺省的日期格式是'DD-MON-RR'
* 获取当前缺省日期格式:SYSDATE;

示例：

|  |
| --- |
| SELECT \* FROM emp WHERE ename = 'SMITH';  SELECT \* FROM emp WHERE hiredate = '02-4月-81';  SELECT SYSDATE FROM dual; |

**例1.**查询dept表中的全部记录；（查询全部记录）

|  |
| --- |
|  |

**例2.**查询emp表中雇员编号、姓名、职务、工资记录；（指定字段）

|  |
| --- |
|  |

**注：**若表中所查询字段有重复记录，用DISTINCT关键字，可以去掉重复的记录；

**例3.**查询emp表中职务，并去掉重复记录；（去掉重复记录）

|  |
| --- |
|  |

**例4.**查询emp表中雇员编号、姓名以及年薪；（字段的算术运算）

|  |
| --- |
|  |

**注：**虽然查询出年薪但字段显示为“SAL\*12”，不利于理解，故可以为之起一个别名。

**例5.**为上例中年薪起一个别名；（为字段其别名）

|  |
| --- |
|  |

**注：**规范，别名不应该是中文🡪设计到程序设计时应该避免使用中文字符；

**例6．**公司每月发200元食补和100元车补，查询其年薪；（字段的算术运算）

|  |
| --- |
|  |

**例7.** 若公司除有食补和车不外，每年多发一个月基本工资，查询其年薪；（字段的算术运算）

|  |
| --- |
|  |

**例8.**在字段间也可以使用“||”符号进行连接，不过显示查询数据时字段之间没有间隔，需要

自己加；（“||”符号的使用）

|  |
| --- |
|  |

**注：**SQL语句中的字符串用单引号括住字符表示。

**例9.** 查询emp表，按如下格式显示记录：

雇员编号是：。。。 ，雇员姓名是：。。。，雇员职务是：。。。，雇员工资是：。。。

|  |
| --- |
|  |

**注：**只有select子句中的别名可以用单引号。

## 2.4限定查询（条件查询）

所谓限定查询即为在限定的条件下，查询符合条件的数据，条件可以用WHERE子句引出。

**SQL语句规范：**在查询时，如果有多个查询条件，则尽可能把过滤数据量最大的条件放在最前面，这样可以减少查询次数，提高效率；

语法

|  |
| --- |
| **SELECT [DISTNCT] \*| 字段[别名] [字段[别名]…]**  **FROM 表名称[别名]**  **[WHERE 条件（s）];** |
| **SELECT \* | { [DISTINCT] <字段名> | <表达式> [<别名>] , ...}**  **FROM <表名>**  **[WHERE <查询条件>];** |

**注**：WHERE子句后可以添加多个条件，常见条件有：

比较运算：>,>=,<,<=,<>（!=）;

关系运算：AND , OR , NOT;

范围操作符：BETWEEN…AND…, LIKE , IN ,IS (NOT) NULL;

### 2.4.1关系运算

**例10.**查询emp表中所有薪水大于1500的员工记录；

|  |
| --- |
|  |

**例11.**查询所有职务为clerk的员工记录；（未选定行）

|  |
| --- |
|  |

**注：**在oracle中，数据是区分大小写的，在表中，clerk为大写的，故无法查出来；应修改如下：

|  |
| --- |
|  |

**注：**当查询时出现“未选定行”时，说明未查出相关条件数据，一般有两个原因：

（1）表中没有数据；（2）查询条件有问题，例如：大小写问题；

**例12.**查询emp表，查出薪水在1500-3000的雇员的记录；（多条件查询，条件用AND或OR 连接）

|  |
| --- |
|  |

**例13.**查询emp表，查出其中职务为“clerk”或“salesman”的员工记录；(AND)

|  |
| --- |
|  |

**例14.**查询emp表，查出职务为“clerk”或“salesman”且薪水大于1500的员工记录；(OR,AND)

|  |
| --- |
|  |

**例15.**查询emp表，查出职务不是“clerk”的所有员工记录；(NOT,<>,!=)

|  |
| --- |
|  |
|  |
|  |

### 2.4.2范围判断

**例16．**查询emp表，查出薪水介于1500到3000之间（即sal>=1500 AND sal<=3000）的员工记录；

(BETWEEN 最小值 AND 最大值)

|  |
| --- |
|  |

**注**：BETWEEN…AND…不光可以对数字操作，也可以对日期操作，且可以用NOT 取反；

### 2.4.3 判断是否为空（IS (NOT) NULL）

**例17．**查询emp表，查出雇用日期在“1981年1月1日”到“1981年12月31日”的员工记录；

|  |
| --- |
|  |

**例18.**查询emp表，查出所有获得奖金的员工记录；(IS NOT NULL)

|  |
| --- |
|  |
|  |

**注：**NULL的概念不等同与0或空字符串，它表示未标定值；

### 2.4.4 IN操作符：判断是否在指定范围

**例19.**查询emp表，查出雇员编号为7369,7566,7799的员工记录；

|  |
| --- |
|  |
|  |

**注：**NOT IN表示判断不再指定范围；IN操作符的范围中允许出现null值，但NOT IN的范围中

不允许出项null值；

|  |
| --- |
|  |
|  |

### 2.4.5 LIKE子句：模糊查询

模糊查询某些字段，有两个匹配符

1. 匹配单个字符：\_
2. 匹配任意多个字符：%
3. 对于特殊符号可使用ESCAPE ‘转义符’声明转义符，然后运用声明好的转义符进行转义，Oracle中的转义字符可以随意声明，并不限定某个字符；

示例：

|  |
| --- |
| SELECT \* FROM emp WHERE ename LIKE '%\\_%' ESCAPE '\'; |

1. 单引号转义：对于字符串中有单引号的情形，可以用两个单引号进行转义；

示例：

|  |
| --- |
| INSERT INTO emp(empno,ename,sal) VALUES(8000,'Tom''cd',2200);  SELECT \* FROM emp WHERE ename LIKE'%''%'; |
|  |

**例20．**查询emp表，查出雇员姓名以A打头的全部记录。

|  |
| --- |
|  |

**例21.** 查询emp表，查出雇员姓名中第二个字母为A的记录；

|  |
| --- |
|  |

**例22.** 查询emp表，查出雇员姓名中带有字母A的记录；

|  |
| --- |
|  |

**注：**LIKE前也可加NOT 关键字表示取反；LIKE子句不光可以操作字符串，还可以操作其他类型

数据；

**例23.** 查询emp表，查出雇员编号或雇用日期或工资中带“1”的记录；

|  |
| --- |
|  |

**注：**若查询关键字没有做限定，则表示查询全部数据；

**例24.** 查询全部数据；

|  |
| --- |
|  |

## 2.5数据排序

默认情况下，查询数据按照主键排序；但也可以使用ORDER BY 子句指明排序方式。

语法

|  |
| --- |
| **SELECT [DISTNCT] \*| 字段[别名] [字段[别名]]**  **FROM 表名称[别名]**  **[WHERE 条件（s）]**  **ORDER BY 字段 [ASC|DESC] [,字段 [ASC|DESC]…];** |

**注**：ORDER BY 子句要放到SQL语句的最后边；可以指定按照多个字段进行排序；排序有两种

方式：

（1）ASC:升序排序，由小到大，默认；

（2）DESC：降序排序，由大到小；

**例25.** 查询emp表中的所有记录，并按工资排序（默认排序方式）；

|  |
| --- |
|  |
|  |

若按降序为记录排序，则如下：

|  |
| --- |
|  |

**例26.** 查询emp表中的所有记录，并按工资排序，若工资相同则按雇佣日期排序；

|  |
| --- |
|  |