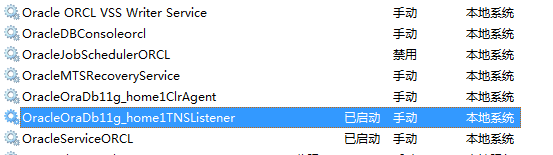
Oracle

Oracle 安装完成之后实际上会在系统中进行服务注册，在注册的服务中有以下两个服务必须启动，否则Oracle无法正常使用

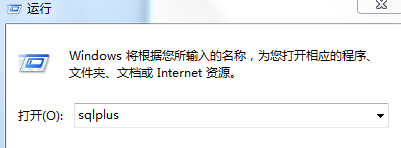


1. OracleOraDb10\_home1TNSListener：表示监听服务，如果客户端要想连接到数据库，此服务必须打开，在日后程序开发中此服务起作用
2. OracleServiceORCL表示数据库的主服务，命名规则：OracleService数据库名，此服务必须启动，否则Oracle无法使用

Oracle数据库基本操作

Oracle只有一个数据库ocle

1. 进入数据库中： sqlplus



1. 用户类型及我的密码

普通用户： 用户名：scott 密码： tiger

普通管理员： system 用户的密码：manager

超级管理员： sys用户的密码：851596

3.基本实验 select \* from emp;

设置页行：set linesize 300; set pagesize 20;

4.以系统管理员的身份登录

Sql> conn sys/851596 as sysdba

Sql> conn system/manger as sysdba

再次发出 select \* from emp;会提示出表或试图不存在，因为emp表时属于scott用户下的表，而sys用户本身没有。如果现在不同用户下想访问emp该表，则必须加上用户名，即：表的完整名称：“scott.emp”

5. 一些不常用的命令

Show user;显示当前正在连接的用户是哪一个

Select \* from tab;得到一个数据库中全部表的名称

Desc emp;查看表的结构

**oracle的scott用户下默认表**

EMP（雇员表）

NO 字段 类型 描述

1 EMPNO NUMBER(4) 雇员编号

2 ENAME VARCHAR2(10) 表示雇员姓名

3 JOB VARCHAR2(9) 表示工作职位

4 MGR NUMBER(4) 表示一个雇员的领导编号

5 HIREDATE DATE 表示雇佣日期

6 SAL NUMBER(7,2) 表示月薪，工资

7 COMM NUMBER(7,2) 表示奖金或佣金

8 DEPTNO NUMBER(2) 表示部门编号

部门表：表示一个部门的具体信息

DEPT（部门表）

NO 字段 类型 描述

1 DEPTNO NUMBER(2) 部门编号

2 DNAME VARCHAR2(14) 部门名称

3 LOC VARCHAR2(13) 部门位置

奖金表：表示一个雇员的工资及奖金。

BONUS（奖金表）

NO 字段 类型 描述

1 ENAME VARCHAR2(10) 雇员姓名

2 JOB VARCHAR2(9) 雇员工作

3 SAL NUMBER 雇员工资

4 COMM NUMBER 雇员奖金

一个公司是有等级制度，用此表表示一个工资的等级

SALGRADE（工资等级表）

NO 字段 类型 描述

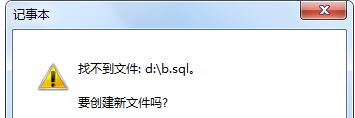
1 GRADE NUMBER 等级名称

2 LOSAL NUMBER 此等级的最低工资

3 HISAL NUMBER 此等级的最高工资

简单命令语句：

创建编辑文件：ed d:/a;关闭后执行



执行文件： @ d:/a.sql;

查看用户下所有的表：select \* from tab;

查看表结构： desc 表名;

**Oracle: 数据类型：**

**主要的数据类型：**

**number(4)表示数字，长度为4**

**Varchar2（10）表示的是字符串，只能容纳10个长度**

**Date；表示日期**

**Number(7,2) 表示数字，其中小数位占2位，整数位占5位**

1. 数字函数   
   ABS(abs) 取绝对值 (power) 乘方 ln 10为底数取0   
   sqrt 平方根 EXP e的n次乘方 LOG(m,n) m为底数n取0   
   数学运算函数:ACOS ATAN ATAN2 COS COSH SIGN SIN SINH TAN TANH   
   CEIL 大于或等于取整数   
   FLOOR 小于或等于取整数   
   mod 取余数   
   ROUND(n,m) 按m的位数取四舍五入值如果round(日期): 中午12以后将是明天的日期.       
   round(sysdate,'Y')是年的第一天   
   TRUNC(n,m) 按m的位数取小数点后的数值如果trunc(日期), 确省的是去掉时间
2. 字符函数

CHR 按数据库的字符集由数字返回字符   
CONCAT(c1,c2) 把两个字符c1,c2组合成一个字符, 和 || 相同   
REPLACE(c,s,r) 把字符c里出现s的字符替换成r, 返回新字符   
SUBSTR(c,m,n) m大于0,字符c从前面m处开始取n位字符,m等于0和1一样,   
m小与0,字符c从后面m处开始取n位字符   
TRANSLATE(c,f1,t1) 字符c按f1到t1的规则转换成新的字符串   
INITCAP 字符首字母大写,其它字符小写   
LOWER 字符全部小写   
UPPER 字符全部大写   
LTRIM(c1,c2) 去掉字符c1左边出现的字符c2   
RTRIM(c1,c2)   
TRIM(c1,c2) 去掉字符c1左右两边的字符c2   
LPAD(c1,n,c2) 字符c1按制定的位数n显示不足的位数用c2字符串替换左边的空位   
RPAD(c1,n,c2)

1. 日期函数

ADD\_MONTHS(d,n) 日期值加n月   
LAST\_DAY  返回当月的最后一天的日期   
MONTHS\_BETWEEN(d1,d2) 两个日期值间的月份,d1<d2 返回负数   
NEXT\_DAY  返回日期值下一天的日期   
SYSDATE 当前的系统时间

DUAL是SYS用户下一个空表，它只有一个字段dummy

1. 转换函数(1)

**TO\_CHAR(date,'日期显示格式') to\_char(sysdate,’yyyy-mm-dd hh24:mi:ss’)**

**所有转换后是字符串，不在date格式的数据  
TO\_CHAR(number) 用于显示或报表的格式对齐**

**To\_char(123456789,’999,999,999’);  
TO\_DATE(char,'日期显示格式')**   
TO\_LOB 把long字段转换成lob字段   
TO\_NUMBER(char) 用于计算或者比较大小   
4. 转换函数(2)   
to\_date里日期显示格式   
YYYY 年 YEAR YYY YY Y   
Q 季度   
MM 月 MONTH MON   
W 星期 (week of month) WW, IW (week of year)   
(说明：周计是按ISO标准,从1月1日的星期数到后面七天为一周,不一定是从周一到周日)   
DD 日 DAY DY   
HH24 小时 HH12 HH   
MI 分钟   
SS 秒   
如果想固定日期的显示格式可以在数据库的参数文件initorasid.ora里新写一行参数        
NLS\_DATE\_FORMAT=yyyy-mm-dd   
hh24:mi:ss可以在UNIX环境变量或者NT的注册表里的设置 NLS\_DATE\_FORMAT=yyyy-mm-dd        
  
4. 转换函数(3)

如果想固定日期的显示格式可以用alter session命令改变   
SQL> alter session set nls\_date\_format='yyyy-mm-dd hh24:mi:ss';   
它的作用顺序如下:   
initialization parameter   
Environment variable   
ALTER SESSION command   
4. 转换函数 (4)   
to\_char(number)里数字显示格式   
9 数字位   
0 数字前面补0 to\_char(-1200,'00000.00')   
. 小数点的位置   
, 标记位置的逗号 用在数字显示格式的左边   
L 根据数据库字符集加货币符号 to\_char(-1200,'L9999.99')   
B 把数字0显示为空格,用在数字显示格式的右边   
MI 右边的负数标记 to\_char(-1200,'9999.99MI')   
PR 括起来的负数 to\_char(-1200,'9999.99PR')   
EEEE 用指数方式表示 to\_char(-1200,'9999.99EEEE')

5. 输入字符,返回数字的函数

instr(c1,c2) 字符c2出现在c1的位置, 不出现, 返回0, 常用于模糊查询   
length  按数据库的字符集,算出字符c的长度,跟数据库的字符集有关, 一个汉字长度为1   
6. 有逻辑比较的函数NVL(EXPR1, EXPR2)函数   
解释: IF EXPR1=NULL RETURN EXPR2   
ELSE RETURN EXPR1   
DECODE(AA0V10R10V20R2....)函数   
解释: IF AA=V1 THEN RETURN R1   
IF AA=V2 THEN RETURN R2   
..…   
ELSE   
RETURN NULL   
举例: decode(id,1,'dept sale',2,'dept tech')

查询语句：

SELECT {DISTINCT}\*|查询列1 别名1，查询列2 别名2……

FORM 表名称`1 别名1,表名称2 别名2,……

{WHERE 条件表达式}

{ORDER BY 排序字段 ASC|DESC,排序字段 ASC|DESC,……}

[可有可无] ， <可变的>

1. 查询语句关键字：

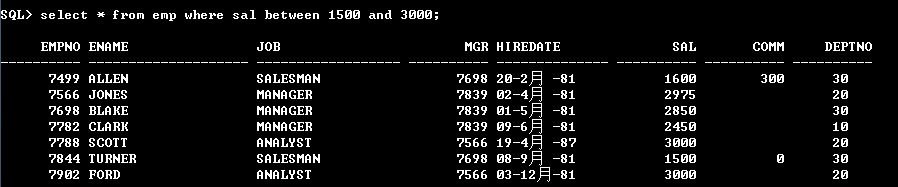
From \* from table;

1.确定范围： between and ，not between and

**|| 字符串连接符：**



**Between** : between 1500 and 300;(1500<=sal<=300)



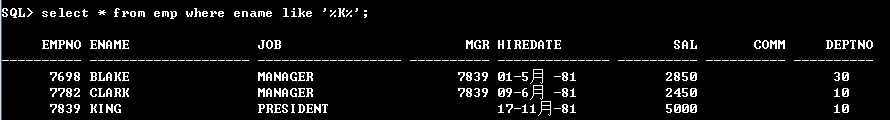
**相反的是： not between 1500 and 3000;**

**相同： sal>=1500 and sal<=3000;**

2.字符匹配：like ， not like；

通配符：用%代表任意长度（可以为0）的字符串，

用\_ 代表任意单个字符

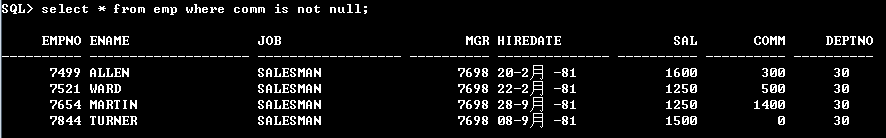


当查询t\_user类型，就不能用 like ‘\_\_%’，需要是\_是，用’\\_’来取消通配符，

2.比较大小： > < = <= >= != !< !>

4.空值：is null，is not null

当查询为是否null的时候，select \*from emp where conn is null;不能用”=”;等不到值



5.确定集合 in not in

**排序：**

**Order by sal asc ,ename desc;**

**当第一列如果有相等，则进下一列的排序**

**Select \* from emp order by sal ,ename desc;**

**聚集函数：**

Count([distinct |all]\*) 统计个数(count(\*))

Count([distinct |all]<列名>)统计一列中值的个数（count(sal)）

Sum() 计算一列值的总和

Avg() max min

Select count(\*) from emp group by sal;

关于表：

Create table t\_user(id int , varchar2(23),password varchar2(23));

复制表：

Create table myemp as select \* from emp;

dual系统中的一张表：

select sysdate from dual;获取当前系统时间

删除表，drop table myemp;

插入：

Insert into <表名> [(<属性列1>……)] value(<常量1>……)

Insert into t\_users(username,password) values(‘tom’,’1234’);

Insert into t\_users values(1,’tom’,’12345’);

下面一种必须对应创建表的字段循序