Oracle学习笔记 - KT野人 - 博客园

oracle 命令:
1.登录
sqlplus system/yeren
2.切换用户(连接命令)
conn sys/yeren@oracle as sysdba(即切换到了超级管理员)
3.修改密码
passw
4.断开连接
disc
5.显示当前用户
show user
6.退出
exit
7.用@或start执行脚本文件
@ d:\tab1.txt
8.将控制台上的内容输出到文件
spool d:\out.txt
select * from tab1;
spool off
9.修改表结构
添加一列
alter table tablename add(clumnname,type);
修改某列
alter table tablename modify(clumnname,type);
10.事务
设置保存点
savepoint name;
回滚至保存点
rollback to name;
提交
commit;
设置只读事务
set transaction read only;
11.字符函数
select upper(substr(ename,1,1))||lower(substr(ename,2,length(ename)-1)) from emp;
而mssql 是这样实现的
select upper(substring(id,1,1))+lower(substring(id,2,len(id)-1)) from person
---
select replace(ename,'A','我是A') from emp; mssql无区别
---
to_date('1999/8/8','yyyy-mm-dd')进行自定义格式时间转换 默认是dd-mm月-yyyy
12.数学函数
四舍五入
round(dataclumn,位数);round(34.45)=34; round(34.45,1)=34.5
直接舍弃
trunc(dataclumn) 截取到整数 trunc(55.99)=55; trunc(55.49,1)=55.4;trunc(55.99,-1)=50;
向下(小)方向取整 即原值>=新值
floor(11.99)=11;floor(-11.99)=-12;
向上(大)方向取整 即原值<=新值
ceil(11.01)=12;ceil(-11.01)=-10;
取模函数
mod(10,2)=0;mod(5,3)=2;
abs(-1)=1; acos(n);asin(n);atan(n);cos(n);sin(n);exp(n);log(m,n);power(m,n);
13.日期函数
当前系统时间
sysdate
add_months(clumn,n);在clumn基础上增或者减去相应的月数
last_day(hiredate)取得该日期月份的最后一天

14.转化函数
to_char(data,exp)转换为指定格式的字符串
select to_char(hiredate,'yyyy-mm-dd hh24:mi:ss') from emp;
select to_char(sal,'L99,999.99') from emp; L本地货币 $ 美元 C本地国际货币符
15.系统函数
sys_context
select sys_context('userenv','db_name') from dual;
select sys_context('userenv','current_schema') from dual;
select sys_context('userenv','SESSION_USER') from dual;
select sys_context('userenv','NLS_DATE_FORMAT') from dual;
16.导出表
exp userid=system/yeren@oracle tables=(scott.emp) file=d:\emp.dmp //导出表及数据
exp userid=system/yeren@oracle tables=(scott.emp) file=d:\emp.dmp rows=n //只导出表结构(间接)
exp userid=system/yeren@oracle tables=(scott.emp) file=d:\emp.dmp direct=y //只导出表结构(直接)
17.导出方案
exp userid=system/yeren@oracle owner=scott file=d:\scott.dmp //导出方案
18.导出数据库
exp userid=system/yeren@oracle full=y inctype=complete file=d:\oracle.dmp
19.导入表
只要把exp替换成imp,userid=scott/yeren@oracle
但要注意由于导出是由system创建的,所以要暂时把scott设置成dba角色权限
20.导入数据
imp userid=scott/yeren@oracle tables=(emp) file=d:\emp.dmp ignore=y
21.导入方案
导入自己创建的方案
imp userid=scott/yeren@oracle file=d:\scott.dmp
导入别人创建的方案
imp userid=system/yeren@oracle file=d:\scott.dmp fromuser=system touser=scott;
22.导入数据库
imp userid=system/yeren@oracle full=y file=d:\oracle.dmp
23.查询当前用户创建的所有表
select * from user_tables;
24.查询当前用户能访问的所有表
select * from all_tables;
25.查询dba拥有的表(需具有dba权限)
select * from dba_tables;
26.查询当前所有用户信息(需具有dba权限)
select * from dba_users;
27.查询所有的角色
SELECT * FROM DBA_ROLES;
28.查询角色包含的权限
系统权限
select * from dba_sys_privs where grantee='CONNECT';
对象权限
select * from dba_tab_privs where grantee='DBA';
29.查询用户包含的权限
系统权限
select * from dba_sys_privs where grantee='SCOTT';
对象权限
select * from dba_tab_privs where grantee='SCOTT';
30.查询所有的系统权限
select * from dba_sys_privs;
31.查询所有的对象权限
select * from dba_tab_privs;
32.查询某个用户具有的角色
select * from dba_role_privs where grantee='SCOTT';
33.显示当前数据库的全称
select * from global_name;
34.显示当前用户可以访问的数据字典视图
select * from dict where comments like '%grant%';
35.创建表空间
create tablespace sp1 datafile 'd:\sp1001.dbf' size 20m

```
   uniform size 128k;--uniform:区大小
 ALTER TABLESPACE tbs_03
     ADD DATAFILE 'tbs_f04.dbf'
     SIZE 50K
     AUTOEXTEND ON
     NEXT 10K
     MAXSIZE 100K;
```

36.使表空间脱机或联机

```
 alter tablespace name offline;
 alter tablespace name online;
```

37.使表空间只读或可读写

```
 alter tablespace name read only;
 alter tablespace name read write;
```

38.查询表空间下的所有表

```
 select * from all_tables where TABLESPACE_NAME='SP1';
```

39.查询某表相关信息

```
 select * from USER_tables where TABLE_NAME='EMP';
```

40.删除表空间(包含所有表内容)

```
 drop tablespace name including contents and datafiles;
```

41.扩展表空间

a.增加数据文件

```
 alter tablespace sp1
 add datafile 'd:\sp1002.dbf'
 size 20m
 autoextend on
 next 10m
 maxsize 200m;
```

b.增加数据文件的大小(报错)

```
 alter tablespace sp1 datafile 'd:\sp1001.dbf' resize 40m;
```

c.设置文件的自增长(报错)

```
 alter tablespace sp1 datafile 'd:\sp1001.dbf' autoextend on next 10m maxsize 500m;
```

42.蠕虫式添加数据

```
 insert into myorz select * from myorz;
```

43.查询所有数据文件

```
 select * from dba_data_files;
```

44.移动数据库文件

```
 cmd下 move 原路径 新路径;
```

45.修改表空间文件路径

```
 alter tablespace sp1 rename datafile 原路径 to 新路径;
```

46.建表

```
SQL> create table goods(
  2  goodsId char(8) primary key,
  3  goodsName varchar2(30),
  4  unitprice number(10,2) check(unitprice>0),
  5  category varchar2(8),
  6  provider varchar2(30)
  7  );
Table created
SQL> create table customer(
  2  customerId char(8) primary key,
  3  name varchar2(50) not null,
  4  address varchar2(50),
  5  email varchar2(50) unique,
  6  sex char(2) default '男' check(sex in ('男','女')),
  7  cardId char(18)
  8  );
Table created
SQL> create table purchase(
  2  customerId char(8) references customer(customerid),
  3  goodsId char(8) references goods(goodsid),
  4  nums number(10) check(nums between 1 and 30));
```

Table created
47.添加非空约束
```
alter table goods modify goodsName not null;
```
48.添加唯一约束
```
alter table customer add constraint unq_carid unique(cardId);
```
49.添加检查约束
```
alter table customer add constraint ch_address check(address in('东城','西城','海淀','朝阳','通州','崇文'));
```
50.添加外键
```
alter table PURCHASE add foreign key (GOODSID) references GOODS (GOODSID);
```
51.删除约束
```
alter table 表名 drop constraint 约束名;
alter table 表名 drop primary key cascade;
```
52.查询约束信息
显示约束信息
```
select * from user_constraints where table_name='PURCHASE';
```
显示约束列
```
select *  from user_cons_columns where table_name='PURCHASE';
```
53.列级定义约束(如:非空)
```
 create table customer(
    customerId char(8) constraint pk_customer primary key,
    name varchar2(50) not null,
     address varchar2(50),
    email varchar2(50) unique,
     sex char(2) default '男' check(sex in ('男','女')),
     cardId char(18)
   );
```
54.表级定义约束
```
create table customer(
    customerId char(8),
    name varchar2(50) not null,
     address varchar2(50),
    email varchar2(50),
     sex char(2) default '男',
     cardId char(18) ,
 constraint pk_customer primary key,
 constraint uq_email unique(email),
 constraint ck_sex check (address in('男','女'));
   );
```
55.新建索引index
单列索引
```
create index indexname on customer(name);
```
复合索引
```
create index indexname on customer(name,cardId);
```
56.查询索引信息
显示索引信息
```
select * from user_indexes where table_name='CUSTOMER';
```
显示索引列信息
```
select * from user_ind_columns where table_name='CUSTOMER';
```
57.创建用户
```
create user name identified by password
```
58.授予用户系统权限
```
grant create session,create table to user [with admin option];
with admin option --表示权限可继续授权
```
59.回收系统权限
```
revoke create session from user;(系统权限不是级联回收)
```
60.对象权限(用户访问其他方案对象的权利)
```
grant select on emp to user with grant option;
grant all on emp to user with grant option;
grant update on emp(sal) to user;
with grant option --可以将对象权限继续授权(带上该属性后user不能是角色)
```
61.回收对象权限

revoke select on emp from user;(对象权限是级联回收)
62.创建角色
create role 角色名 not identified;
create role 角色名 identified by 密码;
63.给角色赋权限和给用户赋权几乎没有区别
64.把角色赋给用户
grant 角色名 to 用户名;
65.删除角色
drop role 角色名;
66.查询角色所具有的系统权限
select * from role_sys_privs where role='角色名'
67.查询角色所具有的对象权限
select * from dba_tab_privs where role='角色名'
68.查询所有角色
select * from dba_roles;
69.显示错误信息
show error;
70.代码块
```
SQL> set serveroutput on
SQL> begin
  2  dbms_output.put_line('hello');
  3  end;
  4  /
```
71.创建存储过程
A:无参的
```
create or replace procedure proc_emp_insert is
 begin
  insert into emp values('12','afsdfa');
  end;
  /
```
B:有参的
只有输入参数的
```
create or replace procedure proc_emp_insert(empid in varchar2,empname in varchar2) is
 begin
  insert into emp values(empid,empname);
  end;
/
```
带输出参数的
```
create or replace procedure proc_emp_selectnamebyId(empid in varchar2,v_empname out  varchar2) is
 begin
  select empname into v_empname from emp where empno=empid;
  end;
/
```
返回结果集的
```
create or replace package testpackage as type testcursor is ref cursor;
end testpackage;
create or replace procedure sp_pro(spno in number,p_cursor out  testpackage.testcursor) is
begin
 open p_cursor for select * from emp where empno=spno;
end;
```
只有输入参数的执行
```
exec proc_emp_insert('67','ayiueyria');
```
带输出参数的执行
```
declare name varchar2;
exec proc_emp_insert('67',name out);
```
72.调用存储过程
exec 存储过程名(参数1，参数2...);
call 存储过程名(参数1，参数2...);(报错)
73.创建函数
```
create or replace function func_emp_empname(v_empid varchar2)
 return varchar is v_empname varchar2(20);
 begin
```

```
       select empname into v_empname from emp where empid=v_empid;
       return v_empname;
 end;

SQL> var ab varchar2(20);
SQL> call func_emp_empname('8888') into:ab;
```
74.创建包
```
 create package sp_package1 is
 procedure 存储过程声明;
 function 函数声明;
 end;
```
75.实现包体
```
 create or replace package body sp_package1 is
 存储过程代码
 函数代码
 end;
```
76.变量常量的定义赋值
```
 set serveroutput on;
 declare c_rate number(3,2):=0.03;
 v_ename emp.ename%type;
 v_sal emp.sal%type;
 v_tax_sal number(7,2);
 begin
  select ename,sal into v_ename,v_sal from emp where empno=&no;
  v_tax_sal:=v_sal*c_rate;
  dbms_output.put_line('姓名是'||v_ename||'工资是'||v_sal||'所得税'||v_tax_sal);
 end;
```
77.复合类型

自定义记录类型(实际上相当于一种数据结构)
```
declare type emp_type is record(empno emp.empno%type,ename emp.ename%type);
```
自定义表类型(实际上相当于数组)
```
declare type emp_type is table of ename emp.ename%type index by binary_integer;
```
游标变量
```
declare type emp_cursor is ref cursor;
test_cursor emp_cursor;
v_ename emp.ename%type;
v_sal emp.sal%type;
v_empno emp.empno%type;
begin
open test_cursor for select ename,sal,empno from emp where deptno=&no;
loop
    fetch test_cursor into v_ename,v_sal,v_empno;
    if v_sal<1000
    then
         update emp set sal=sal+100 where empno=v_empno;
    end if;
    exit when test_cursor%notfound;
    dbms_output.put_line(v_ename||':'||v_sal);
end loop;
end;
```
78.条件判断语句
```
if 条件表达式 then 语句 end if;
if 条件表达式 then 语句 else 语句 end if;
if 条件表达式 then 语句 elsif 条件表达式 then 语句 else 语句 end if;
case
when 条件表达式 then
语句;
when 条件表达式 then
语句;
else
语句;
```

```
end case;
```
79.循环语句
```
loop 语句 end loop;(相当于do-while)
while  条件表达式 loop 语句 end loop;(相当于while)
for循环
for i in reverse 1..10 loop
 insert into users values(i,"xx");
end loop;
```
80.顺序控制 goto
```
begin
 loop
  语句
  if 退出循环条件表达式 then
   goto end_loop;
  end if;
 end loop;
 <>
end;
```
null;相当于什么都不做
81.分页存储过程
```
create or replace package testpackage as type testcursor is ref cursor;
 end testpackage;
create or replace procedure fenye
(tablename in varchar2,pagesize in number,pageindex in number,descclumn in varchar2, totalrows out
number,pagecount out number,p_cursor out testpackage.testcursor) is
v_sql varchar2(1000);
v_begin number:=(pageindex-1)*pagesize+1;
v_end number:=pageindex*pagesize;
begin
 v_sql:='select * from (select rownum rn,t1.* from (select * from '||
tablename||' order by '||descclumn||' desc) t1 where rownum<='||v_end||') where rn >= '||v_begin;
 open p_cursor for v_sql;
 v_sql:='select count(*) from '||tablename;
 execute immediate v_sql into totalrows;
 if mod(totalrows,pagesize)=0 then
 pagecount:=totalrows/pagesize;
 else
 pagecount:=totalrows/pagesize+1;
 end if;
 dbms_output.put_line(v_sql);
end;
```
82.三种分页
```
--rowid分页
select * from (select rownum rn,t2.* from (select t1.*,rowid rid from temp t1) t2 where rownum <= 1000)
where rn >=800;
--rownum分页
select * from (select rownum rn,t1.* from (select * from student) t1 where rownum<=10) where rn >= 5;
--row_number函数分页
select  *  from
(select  t.* ,row_number() over (order  by xh  desc ) rk   from  student  t)
 where  rk <=10  and  rk  >=5;
--三种方式比较
select rownum rn,t1.* from (select * from temp) t1 where rownum<=1000 minus select rownum rn,t1.* from
(select * from temp) t1 where rownum<=800;
set timing on;
select * from (select rownum rn,t1.* from (select * from temp) t1 where rownum<=1000) where rn >= 800;
select * from (select rownum rn,t2.* from (select t1.*,rowid rid from temp t1) t2 where rownum <= 1000)
where rn >=800;
```
83.预定义例外
```
exception when no_data_found then dbms_output.put_line('异常消息');
case_not_found case分支语句错误
dup_val_on_index 在唯一索引列上插入重复值
```

invaild_cursor 试图在不合法的游标上操作或是游标没打开或是关闭没有打开的游标
invaild_number 输入数字错误
too_many_rows 返回超过一行
zero_divide 除以0
value_error 定义的变量长度不足以接受返回值
84.自定义例外

```
create or replace procedure extest(spno number) is
myex exception;
begin
 update emp set sal=sal+1000 where empno=spno;
 if sql%notfound then
 raise myex;
 end if;
 exception when myex then dbms_output.put_line('异常消息');
end;
```

85.视图

```
create view 试图名 as select语句;
```

86.触发器

```
create or replace trigger tri_banjibefore before update on ban_ji
for each row
begin
insert into ban_log values(:old.banid);
end;
```

87.权限相关字典：

DBA_USERS 所有用户
DBA_ROLES 所有角色
SYSTEM_PRIVILEGE_MAP 所有权限
ROLE_SYS_PRIVS 角色拥有的系统权限
ROLE_TAB_PRIVS 角色拥有的对象权限
USER_TAB_PRIVS_MADE 查询授出去的对象权限(通常是属主自己查)
USER_TAB_PRIVS_RECD 用户拥有的对象权限
USER_COL_PRIVS_MADE 用户分配出去的列的对象权限
USER_COL_PRIVS_RECD 用户拥有的关于列的对象权限
USER_SYS_PRIVS 用户拥有的系统权限
USER_TAB_PRIVS 用户拥有的对象权限
USER_ROLE_PRIVS 用户拥有的角色

88.同义词

如果要创建一个远程数据库上的某张表的同义词，需要先创建一个dblink来拓展访问，然后使用如下语句创建数据库同义词：

```
create synonym table_alias_name for tablename@dblink ;
```