**第一讲**

1. sql:结构化查询语句
2. 数据库(DB)database:用于进行动态存储和处理用户数据的一系列类型的集合
3. 数据库系统（DBS）:由多个DB组成的一个集合
4. 数据库管理系统（DBMS）:针对用户而言的一套专业数据库管理软件，可以实现对

数据库的增删改查等一系列的操作。

1、oracle 数据库：属于甲骨文公司，该公司是全球企业级的软件服务公司，总部设立在美国加利福尼亚州的红木摊，其中oracle是全球为受欢迎的一款分布式关系型数据库。20世纪70年代的时候是为美国中央情报局所使用。目的主要是为了手机情报数据。

重点要考！！！

**2、oracle数据库与sql server数据库的区别**

1.oracle数据库对数据的处理性能以及内容量都要比sql server强。

2.oracle数据库的安全性比sql server的安全性要高；因为sql server 认证模式只有两种，而oracle 数据库是根据用户、角色权限来划分的。

3.oracle 数据库是以用户进行管理表的，而sql server是以库进行管理表的。

4.oracle 数据库是b/s和c/s架构的混合架构，而sql server是单一的c/s架构

5.oracle 数据库不仅具有图形化操作界面而同时还存在对应的文本命令模式，而sql server 只存在图形化操作界面。

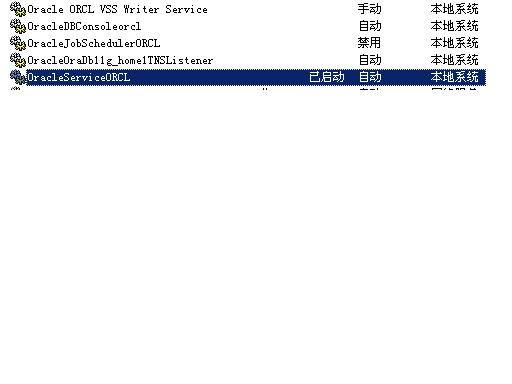
6.数据库认证工程师一般都是oracle数据库认证比较权威。

3、oracle版本标识

1.oracle 9i（internet 网络）表示的是oracle数据库嵌入了网络应用，即与JAVA应用程序融合。

2.oracle 10g/11g（gid 网格）表示的是oracle数据库实现了服务器集群的概念，主要包括独立式服务器集群、共享式服务器集群。

3.oracle 12c/14c/18c/19c（cloud 云）表示的是实现了云服务概念，实现了云数据处理。



ORACLE数据库五大服务：

1、Oracle VSS:版本写入控制服务，设计版本控制管理需要启动（正常情况不需要启动）

2、DBConsoleorcl：oracle的控制台服务，如果实现B/S（浏览器）访问服务器，可监控oracle服务运行情况（CPU、内存、实例），通过网址：http://oracle服务器IP地址:1158/em

3、JobschedulerORCL:工作台服务（可选）

4、TNSlistener:oracle的监听服务，必须启动，用于监听oracle的实例连接（如果未启动则无法建立实例连接，数据库无法访问）

5、ServiceORCL:oracle的实例服务，必须启动。如果不启动数据库无法访问。

名词解释：

oracle服务=实例+数据库

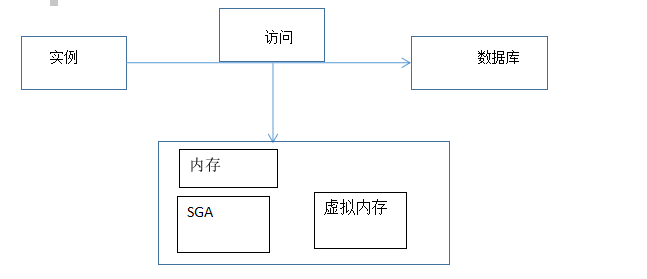
实例：访问数据库的通道

实例=内存+后台进程

数据库=数据库文件+日志文件+控制文件

RAM：随机访问存储器

ROM:只读存储器



SGA：系统全局区，指的是用于进行对数据库的所有内存存储的信息处理

常用账户

1. sys（系统管理员）： 对应的角色是sysdba，拥有一切权限，默认密码change\_install
2. system(系统操作员)：对应的角色是sysoper，拥有的权限和sys一样，但是缺少数据库操作权限（可以对数据库底下的表进行操作，增删改查）。默认密码：manager
3. Scott（普通用户）：拥有管理员所赋予的其他权限，默认密码是tiger

文本模式中查看监听状态

注意：通常只有管理员用户才会有角色，普通用户没有角色

cmd命令行

进入监听程序：lsnrctl

查看服务是否正常启动：status

开启监听服务：start

停止监听服务：stop

重新加载：reload

退出监听服务状态：exit/quit

如果显示无监听服务则是没有启动服务

进入oracle中的sql状态命令

sqlplus /nolog

conn 用户名/密码 as 角色名

Sys system scott

关于oracle中的三个配置文件

配置文件路径：C:\app\Administrator\product\11.1.0\db\_1\NETWORK\ADMIN

1、sqlnet.ora:该配置文件是用于修改登录的认证模式

1. NTS：该模式下在本地服务器端登录表示密码可以随便写
2. NONE:该模式下表示在任何情况中，需要使用正确密码登录
3. listener.ora:用于修改监听程序的配置文件。
4. tnsnames.ora：用于修改实例的配置文件

Sql命令模式

查看当前登录用户名：show user;

解锁/锁定用户

Alter user 用户名 account unlock/lock;

修改用户密码

1. 使用管理员

alter user 用户名 identified by 新密码

1. 用户自己修改密码

使用该用户登录之后

Password

加载实例和卸载实例（只能管理员操作）

Startup open;

Shutdown immediate;

注意语句后面需要接分号结束。

一台计算机可以存在多个实例、一个实例可以对应多个数据库，但是一个数据库只能对应一个实例。

查询数据库名：Select name from v$database;

**第二讲**

用户、角色、权限

1、用户、角色、权限三者之间的关系

在oracle 数据库中主要使用用户进行管理表的，不同用户可以拥有对数据库所有操作对象的权限，用户可以拥有一个或多个角色，也可以直接拥有权限，角色的作用主要为了实现权限的划分和管理。

2、对用户的操作

1.创建用户

create user 用户名 identified by 密码

2.查询用户数据库中的所有用户名

select \* from dba\_users

where USERNAME='ZHANG3'这个是加了条件

3.查询当前用户中的表

查询数据库名：select name from v$database;

---管理员(所有表)：select table\_name from dba\_tables;

---普通用户（当前用户的表）：select table\_name from user\_tables;

4.查看表结构：

desc EMP

describe 表名

5.修改用户密码

alter user 用户名 identified by 密码

6.删除用户

drop user 用户名

drop user 用户名 cascade

注意当用户下存在表时，是无法直接删除，使用关键字cascade。

客户端连接服务器的语法（SQL命令下）

conn 用户名/密码 @服务器 IP地址：端口号/实例名 as 角色

conn sys/orcl @127.0.0.1:1521/orcl as sysdba;

如何检测和oracle服务器的实例通讯是否正常？

tnsping 服务器IP地址：端口号/实例名

注意该命令是在CMD环境下，执行结果返回的是OK说明实例正常

注意！！！端口号！！

mysql 3306

sql server 1433

orcale 1521

3、角色：主要分为系统角色和自定义角色

角色表示的是一组权限的集合，一般方便权限的划分和管理

系统角色中三大常用角色

1.dba角色：拥有最高权限

2.resource角色：

3.connect 角色：连接权限

a.查询数据库中的所有角色roles

select \* from dba\_roles

b.创建角色

create role 角色名

c.删除角色

drop role 角色名

d.查询用户拥有的角色

查询所有拥有的角色：select \* from dba\_role\_privs

查询当前用户拥有的角色：select \* from user\_role\_privs

4、权限

主要分为系统权限和实体权限（增删改查）

实体权限：对表内容操作的权限（增删改查）

1.查询用户/角色名拥有的权限

select \* from dba\_sys\_privs

2.查询当前用户拥有的权限

select \* from user\_sys\_privs

3.查询当前用户拥有的角色的权限

select \* from role\_sys\_privs

sys查询权限；role查询角色；

赋予权限 关键字grant

1、赋权语法

grant 权限 to 用户

grant 权限 to 角色

grant 角色 to 用户

2、撤权语法

revoke 权限 from 用户

revoke 权限 from 角色

revoke 角色 from 用户

实体权限：对表内容的操作，比如select/update/insert/delete

赋予实体权限

grant select/update/delete/insert to 用户名

grant select on scott.emp to xiaobai

on 用户名.表名

to 用户名

撤销权限

revoke select on scott.emp to xiaobai

on 用户名.表名

from 用户名

查询实体权限

select \* from dba\_tab\_privs

查询当前用户实体权限

select \* from user\_tab\_privs

A->B->C

赋予实体权限是，如果间接用户需要将实体权限赋予其他用户，就必须接关键字with grant option

当间接用户的被原表用户进行撤销权限后，那么间接用户所赋予其他用户的权限也相应会被撤销。

将命令行中的SQL语句结果直接写入到本地计算机中的文件

spool 路径名\文件名.sql

spool off

spool off 表示文件写入结束

**第三讲**

子表查询：查询emp表中3-5中的数据

第一种方式 重命名rownum隐藏字段为rn

select \* from

(select rownum rn,emp.\* from emp

where rownum<=5)a

where a.rn>=3

第二种方式 首先显示rownum隐藏字段，最后通过使用查询empno字段表达第二条件 小与3

select \* from

(select rownum,emp.\* from emp

where rownum<=5)a

where a.empno not in (

select empno from emp

where rownum <3

)

sql将null转换为0用:is null

oracle将null转换为0用:nvl(字段名,0)

oracle中存在多个值与多个值比较，是无法直接比较，需要使用关键字all，all表示条件中的所有值（等价于条件中的最大值）。

使用关键字 any，表示条件中的任意值（等价于条件中的最小值）

UNION 表示将多表结果显示在一张表中，并且会剔除重复值

UNION ALL 表示将多表结果显示在一张表中，不会剔除重复值

INTERSET 表结果的交集

MINUS 表结果的差集（例如表A 数据中有1、2、3，表B数据中为1、2，结果为3）

|| 拼接：select '7369员工的工资是'||sal from emp

where empno=7369

结果：7369员工的工资是800

题目：

1、如何按照工资的从低到高的顺序显示雇员的信息？

select \* from emp order by sal+nvl(comm,0)

2、按照部门号升序而雇员工资降序

select \* from emp order by deptno asc,sal+nvl(comm,0) desc

3、按年薪排序，使用别名排序。

select (sal+nvl(comm,0))\*12 as 年薪,emp.\* from emp

order by 年薪 desc

4、显示所有员工的最高工资和最低工资

select max(sal),min(sal) from emp

5、显示每个部门的最高工资和最低工资

select deptno,avg(sal+nvl(comm,0)),max(sal+nvl(comm,0)) from emp

group by deptno

6、显示每个部门每种岗位的平均工资，最低工资

select deptno,avg(sal),min(sal) from emp

group by deptno,job

7.显示平均工资低于2000的部门号和平均工资

select deptno,avg(sal) from emp

group by deptno

having avg(sal)<2000

8、显示雇员名，雇员工资及所在部门的名字【笛卡尔集】

select \* from dept，emp

9、显示部门号为10的部门名、员工名和工资

select dept.dname,ename,sal from emp,dept

where emp.deptno=dept.deptno order by dname

10、显示各个员工工资等级

第一种方式：

select ename,sal,grade from emp,salgrade

where sal>=losal and sal <=hisal

第二种方式：

select ename as 姓名,sal,(

case when sal between 700 and 1200 then 1

when sal between 1200 and 1400 then 2

when sal between 1400 and 2000 then 3

when sal between 2000 and 3000 then 4

else 5 end)等级

from emp

11、查询jones的下属的下属各自的人数

select ename,mgr,count(\*) from emp

where mgr in(select empno from emp

where mgr=(select empno from emp where ename='JONES'))group by mgr,ename

12、如何显示工资比部门30的所有员工的工资高的员工姓名、工资和部门号？

select ename,sal,deptno from emp

where sal>all(

select max(sal+nvl(comm,0)) as max from emp

where deptno=30)

12、如何显示工资比部门30的任意员工的工资高的员工姓名、工资和部门号？

select \* from emp

where sal>any(select min(sal) from emp

where deptno=30)

13、如何显示高于自己部门平均工资的信息

第一种：

select \* from emp,

(select deptno,avg(sal) avg from emp

group by deptno) a

where emp.deptno=a.deptno and emp.sal>a.avg

第二种：

select \* from emp cross join

(select deptno,avg(sal+nvl(comm,0)) avg from emp

group by deptno) a

where emp.deptno=a.deptno and emp.sal>a.avg

第三种：

select \* from emp inner join

(select deptno,avg(sal+nvl(comm,0)) avg from emp

group by deptno) a

on emp.deptno=a.deptno and emp.sal>a.avg

14、对于emp中低于自己工资至少5人的员工，列出其部门号，姓名，工资，以及工资少于自己的人数；

select emp.\*,c.人数 from emp,(

select a.empno,count(\*)人数 from emp a,emp b

where a.empno<>b.empno

and a.sal>b.sal

group by a.empno

having count(\*)>=5)c

where emp.empno=c.empno

15、对于emp中工资高于本部门平均水平，人数多于1人的，列出部门号，人数，按部门排序

select \* from

(select deptno,count(\*)as 人数 from

(select emp.DEPTNO from emp inner join(select deptno,avg(sal) as av from emp

group by deptno)a

on emp.deptno=a.deptno and emp.sal>a.av)

group by deptno)b

where b.人数>1

order by deptno asc

16、各个部门中最高工资中最低的那个部门的最低工资是多少？

select min(sal)

from emp

where deptno = (

select deptno from emp

group by deptno

having max(sal)=(

select min(max(sal))

from emp

group by deptno))

having max（sal）这里指的是在emp表中deptno分组里最大的sal用来匹配子查询中的结果

**第四讲**

17、查询出职位不是clerk的员工且奖金为空的下属所对应的的各部门平均工资与每个部门的员工最高工资的结果集合

SELECT DEPTNO,AVG(SAL) FROM EMP

WHERE MGR IN(SELECT MGR FROM EMP

WHERE JOB<>'CLERK' AND COMM IS NULL)

GROUP BY DEPTNO

UNION

SELECT DEPTNO,MAX(SAL)FROM EMP

GROUP BY DEPTNO

oracle 使用

事务：表示的是对一个事件的操作过程。在数据库中主要是对数据库的增删改查等操作构成。

事务提交 COMMIT

设置事务保存点：SAVAPOINT 保存点名（在不事务提交前使用，事务提交之后会取消所有保存点）

回滚事务：ROLLBACK TO 保存点名

如果用户不提交事务，则在用户退出之后，自动提交事务。

一个用户进行了增删改查操作之后需要进行事务提交（COMMIT）才会对该数据库操作成功提交生效，另一个用户才会更新到同一个数据库中的数据。

PL/SQL

procedure language structed query language

过程化结构查询语言

主要包括三大部分：1.声明定义部分：使用declare 关键字定义声明

1)声明变量v\_变量名

2）声明常亮c\_常量名

3）声明游标cursor\_游标名

4）声明例外部分e\_例外名 （exception）

2.执行部分（语句块部分）

begin

输出

dbms\_output.put\_line()

end

3.例外部分(属于执行部分，必须写在begin与end之间)

exception

如果在sql命令行中，语句块后面都必须接上分号；且最后结束时一定要使用/进行结束；

语句输出的时候需要开启回显

set serveroutput on

如果在客户端工具sql developer中编写PL/SQL语句时，语句结束需要接分号；

begin

dbms\_ouput.put\_line('helloworld');

end;

命令行操作：

begin

dbms\_ouput.put\_line('helloworld');

end;

/

给变量赋值时不需要使用关键字set且需要使用“:=”（双引号不用看）

例如： v\_a:=1;

第二种赋值方法：(在定义变量过程中直接赋值)

declare v\_a number:=1;

可以从控制台给变量赋值

DECLARE V\_A NUMBER(20):=&请输入第一个整数;

V\_B NUMBER(20):=&请输入第二个整数;

V\_C VARCHAR2(20):='&请输入逻辑运算符';

V\_SUM VARCHAR2(20);

BEGIN

V\_SUM:=V\_A||V\_C||V\_B;

dbms\_output.put\_line(V\_SUM);

END;

可以从表中给变量赋值

DECLARE V\_A NUMBER(10);

BEGIN

SELECT EMPNO INTO V\_A FROM EMP WHERE ENAME='ALLEN';

DBMS\_OUTPUT.PUT\_LINE(V\_A);

END;

编写一个PL/SQL语句中从控制台输入员工编号后，输出一句话 员工编号7499的工资是

DECLARE V\_A NUMBER(10):='&请输入员工编号';

V\_NAME VARCHAR2(20);

V\_SAL NUMBER;

BEGIN

SELECT SAL，ENAME INTO V\_SAL,V\_NAME FROM EMP WHERE EMPNO=V\_A;

DBMS\_OUTPUT.PUT\_LINE('员工姓名'||V\_NAME||'的员工编号是'||V\_A||'的工资是'||V\_SAL||'元');

END;

**记录类型**

相当于是一个数据

定义一个记录类型名

declare

type 记录类型名 is record(

变量名1 数据类型，

变量名2 数据类型

);

定义一个记录类型的变量名去接收记录类型中的变量

记录类型变量名 记录类类型名

begin

语句块;

end;

DECLARE

TYPE RECORD\_A IS RECORD(

V\_EMPNO NUMBER:='&请输入员工编号',

V\_SAL NUMBER

);

V\_RE RECORD\_A;

BEGIN

SELECT EMP.SAL INTO V\_RE.V\_SAL FROM EMP WHERE EMPNO=V\_RE.V\_EMPNO;

DBMS\_OUTPUT.PUT\_LINE(V\_RE.V\_SAL);

END;

**使用关键字%type 动态获取表中字段的数据类型**

**直接在定义变量的时候 数据类型改为 表名.字段名%TYPE**

DECLARE

TYPE RECORD\_A IS RECORD(

V\_EMPNO EMP.EMPNO%TYPE:='&请输入员工编号',

V\_SAL EMP.SAL%TYPE

);

V\_RE RECORD\_A;

BEGIN

SELECT EMP.SAL INTO V\_RE.V\_SAL FROM EMP WHERE EMPNO=V\_RE.V\_EMPNO;

DBMS\_OUTPUT.PUT\_LINE(V\_RE.V\_SAL);

END;

**EXCEPTION异常：**

表达方式：

BEGIN

EXCEPTION

WHEN NO\_DATA\_FOUND then 语句块;

END;

例如：

DECLARE

TYPE RECORD\_A IS RECORD(

V\_EMPNO EMP.EMPNO%TYPE:='&请输入员工编号',

V\_SAL EMP.SAL%TYPE

);

V\_RE RECORD\_A;

BEGIN

SELECT EMP.SAL INTO V\_RE.V\_SAL FROM EMP WHERE EMPNO=V\_RE.V\_EMPNO;

DBMS\_OUTPUT.PUT\_LINE(V\_RE.V\_SAL);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('该员工编号不存在');

END;

**声明变量并赋值的所有方法：**

1、 v\_a:=1;

2、(在定义变量过程中直接赋值)

declare v\_a number:=1;

3、可以从控制台给变量赋值

4、可以从表中给变量赋值

5、记录类型进行声明变量且赋值

6、使用关键字%type 动态获取表中字段的数据类型

**第五讲**

**第五讲**

使用关键字%rowtype 动态获取表中多个字段的数据类型。

v\_emp\_record emp%rowtype;

declare

v\_emp\_record emp%rowtype;

begin

select \* into v\_emp\_record from emp where empno='&请输入员工编号';

dbms\_output.put\_line('员工编号是'||v\_emp\_record.empno||'的工资'||v\_emp\_record.sal);

exception

when too\_many\_rows then dbms\_output.PUT\_LINE('返回数据行数太多！');

when no\_data\_found then dbms\_output.PUT\_LINE('输入的员工编号不存在！');

end;

exception异常2！关键字：too\_many\_rows

exception

when too\_many\_rows then dbms\_output.PUT\_LINE('返回数据行数太多！');

end;

**条件判断**

1、单条件判断

if 条件 then 语句块；

end if;

declare

v\_a number:='&请输入一个整数';

begin

if v\_a>100 then

dbms\_output.put\_line(v\_a||'是大于100的！');

end if;

end;

**2、双条件判断**

if 条件 then 语句块1；

else 语句块2；

end if;

declare

v\_a number:='&请输入一个整数';

begin

if v\_a>100 then

dbms\_output.put\_line(v\_a||'是大于100的！');

else dbms\_output.PUT\_LINE(v\_a||'是小于100的');

end if;

end;

**3、多条件判断**

if 条件1 then 语句块1；

else if 条件2 then 语句块2；

else 语句块3;

end if;

end if;

declare

v\_a number:='&请输入一个整数';

begin

if v\_a=100 then

dbms\_output.put\_line(v\_a||'是等于100的！');

else if v\_a>100 then dbms\_output.PUT\_LINE(v\_a||'是大于100的');

else dbms\_output.PUT\_LINE(v\_a||'是小于100的');

end if;

end if;

end;

**else if 是可以缩写，写成：elsif 后面不需要加任何end if 加了会报错**

declare

v\_a number:='&请输入一个整数';

begin

if v\_a=100 then

dbms\_output.put\_line(v\_a||'是等于100的！');

elsif v\_a>100 then dbms\_output.PUT\_LINE(v\_a||'是大于100的');

else dbms\_output.PUT\_LINE(v\_a||'是小于100的');

end if;

end;

题目：编写一个PL/SQL语句，从控制台输入一个员工编号，对比emp表中输出员工编号是多少，薪资是多少且输出该员工姓名的薪资是位于

多少级。

declare

v\_emp\_record emp%rowtype;

begin

select \* into v\_emp\_record from emp where empno='&请输入员工编号';

if v\_emp\_record.sal between 700 and 1200 then dbms\_output.PUT\_LINE('员工姓名为'||v\_emp\_record.ename||'的薪资是'||

v\_emp\_record.sal||',薪资等级为1');

elsif v\_emp\_record.sal between 1201 and 1400 then dbms\_output.PUT\_LINE('员工姓名为'||v\_emp\_record.ename||'的薪资是'||

v\_emp\_record.sal||',薪资等级为2');

elsif v\_emp\_record.sal between 1401 and 2000 then dbms\_output.PUT\_LINE('员工姓名为'||v\_emp\_record.ename||'的薪资是'||

v\_emp\_record.sal||',薪资等级为3');

elsif v\_emp\_record.sal between 2001 and 3000 then dbms\_output.PUT\_LINE('员工姓名为'||v\_emp\_record.ename||'的薪资

是'||v\_emp\_record.sal||',薪资等级为4');

else dbms\_output.PUT\_LINE('员工姓名为'||v\_emp\_record.ename||'的薪资是'||v\_emp\_record.sal||',薪资等级为5');

end if;

end;

**循环语句**

1、while 条件 loop

语句块；

end loop;

输出1-10

declare v\_a number:=1;

begin

while v\_a<=10 loop

dbms\_output.PUT\_LINE(v\_a);

v\_a:=v\_a+1;

end loop;

end;

**输出1-10之和**

declare v\_a number:=1;

v\_sum number:=0;

begin

while v\_a<=10 loop

v\_sum:=v\_a+v\_sum;

v\_a:=v\_a+1;

end loop;

dbms\_output.PUT\_LINE(v\_sum);

end;

**输出1-10奇数之和与1-10偶数之和**

declare v\_a number:=1;

v\_jsum number:=0;

v\_osum number:=0;

begin

while v\_a<=10 loop

if mod(v\_a,2)=1 then

v\_jsum:=v\_a+v\_jsum;

else v\_osum:=v\_a+v\_osum;

end if;

v\_a:=v\_a+1;

end loop;

dbms\_output.PUT\_LINE('奇数之和是'||v\_jsum||'偶数之和是'||v\_osum);

end;

**输出9\*9乘法表**

declare v\_i number:=1;

v\_j number;

v\_sum number:=0;

v\_str varchar2(2000);

begin

while v\_i<=9 loop

v\_j:=1;

v\_str:='';

while v\_j<=v\_i loop

v\_sum:=v\_i\*v\_j;

v\_str:=v\_str||v\_i||'\*'||v\_j||'='||v\_sum||' ';

v\_j:=v\_j+1;

end loop;

dbms\_output.put\_line(v\_str);

v\_i:=v\_i+1;

end loop;

end;

for语句：

declare

v\_sum number:=0;

v\_str varchar2(2000);

begin

for v\_i in 1..9 loop

v\_str:='';

for v\_j in 1..v\_i loop

v\_sum:=v\_i\*v\_j;

v\_str:=v\_str||v\_i||'\*'||v\_j||'='||v\_sum||' ';

end loop;

dbms\_output.put\_line(v\_str);

end loop;

end;

loop语句：

declare v\_i number:=1;

v\_j number;

v\_sum number:=0;

v\_str varchar2(2000);

begin

loop

v\_j:=1;

v\_str:='';

loop

v\_sum:=v\_i\*v\_j;

v\_str:=v\_str||v\_i||'\*'||v\_j||'='||v\_sum||' ';

v\_j:=v\_j+1;

exit when v\_j>v\_i;

end loop;

dbms\_output.put\_line(v\_str);

v\_i:=v\_i+1;

exit when v\_i>9;

end loop;

end;

**字符串和数字能够直接拼接，不需要数据转换。**

输出

\*

\*\*

\*\*\*

declare v\_a number:=1;

v\_b number;

v\_str varchar2(200);

v\_x varchar2(2):='\*';

begin

while v\_a<=10 loop

v\_b:=1;

v\_str:='';

while v\_b<=v\_a loop

v\_str:=v\_str||v\_x;

v\_b:=v\_b+1;

end loop;

v\_a:=v\_a+1;

dbms\_output.put\_line(v\_str);

end loop;

end;

**2、loop循环**

loop

语句块；

exit when 条件;

end loop;

declare v\_a number:=1;

begin

loop

dbms\_output.put\_line(v\_a);

v\_a:=v\_a+1;

exit when v\_a>10;

end loop;

end;

**3、for语句**

默认值只有递增 +1

for 变量名 in 初始值..结束值 loop

语句块；

end loop;

begin

for v\_a in 1..10 loop

dbms\_output.PUT\_LINE(v\_a);

end loop;

end;

**return 结束循环体**

goto 标号

**标记标号是 <<标号>>**

begin

for v\_a in 1..10 loop

dbms\_output.PUT\_LINE(v\_a);

if v\_a =6 then goto x; end if;

end loop;

<<x>>

end;

题目：编写一个PL/SQL语句将emp表中的所有信息输出到屏幕上，显示效果如下：

员工编号：7499 员工姓名： ALLEN 员工工资：1600

……

declare

cursor cursor\_emp is select EMPNO,ename,sal from emp;

type record\_a is record(

v\_num emp.EMPNO%type,

v\_name emp.ename%type,

v\_sal emp.SAL%type

);

v\_re record\_a;

begin

open cursor\_emp;

fetch cursor\_emp into v\_re.v\_num,v\_re.v\_name,v\_re.v\_sal;

while cursor\_emp%found loop

dbms\_output.put\_line('员工编号：'||v\_re.v\_num||' 员工姓名：'||v\_re.v\_name||' 员工工资：'||v\_re.v\_sal||'元');

fetch cursor\_emp into v\_re.v\_num,v\_re.v\_name,v\_re.v\_sal;

end loop;

close cursor\_emp;

end;

索引！！！

**创建索引**

create index 索引名称 on 表名(列名) [tablespace<tablespace\_name>];

**重置索引**

alter index 索引名称 rebuild;

**删除索引**

drop index 索引名称

**游标指的是一组sql语句（select）的结果集合**

**1、先声明游标**

declare

cursor 游标名 is select \* from 表名;

**2、打开游标**

open 游标名

**3、提取游标数据**

fetch 游标名 into 变量名

**4、对游标进行判断是否存在下一条记录，需要使用循环**

while 游标名%found loop

fetch 游标名 into 变量名；

end loop;

**5、关闭游标**

close 游标名

题目：打印30号部门所有员工的工资

某某员工工资是多少？

declare

cursor cursor\_emp is select ename,sal from emp where deptno=30;

v\_name emp.ename%type;

v\_sal emp.sal%type;

begin

open cursor\_emp;

fetch cursor\_emp into v\_name,v\_sal;

while cursor\_emp%found loop

dbms\_output.put\_line(v\_name||'员工的工资是'||v\_sal);

fetch cursor\_emp into v\_name,v\_sal;

end loop;

close cursor\_emp;

end;

**游标：**

**使用for循环进行**

declare

cursor cursor\_emp is select \* from emp where deptno=30;

v\_emp\_record emp%rowtype;

begin

for v\_emp\_record in cursor\_emp loop

dbms\_output.put\_line(v\_emp\_record.ename||'员工的工资是'||v\_emp\_record.sal);

end loop;

end;

**使用while并且使用记录类型进行**

declare

cursor cursor\_emp is select empno,sal from emp where deptno=30;

type emp\_record is record(

v\_empno emp.empno%type,

v\_sal emp.sal%type

);

v\_re emp\_record;

begin

open cursor\_emp;

fetch cursor\_emp into v\_re;

while cursor\_emp%found loop

dbms\_output.put\_line(v\_re.v\_empno||'员工的工资是'||v\_re.v\_sal);

fetch cursor\_emp into v\_re;

end loop;

close cursor\_emp;

end;

**使用loop循环并且使用记录类型进行**

declare

cursor cursor\_emp is select empno,sal from emp where deptno=30;

type emp\_record is record(

v\_empno emp.empno%type,

v\_sal emp.sal%type

);

v\_re emp\_record;

begin

open cursor\_emp;

fetch cursor\_emp into v\_re;

loop

dbms\_output.put\_line(v\_re.v\_empno||'员工的工资是'||v\_re.v\_sal);

fetch cursor\_emp into v\_re;

exit when cursor\_emp%notfound;

end loop;

close cursor\_emp;

end;

**隐式游标：NOTFOUND**

题目：从控制台输入员工编号,如果该员工的工资小于2000则加100否则输出工资大于或等于2000并实现数据异常错误报错分两种情况:

Too\_many\_rows 和 no\_data\_found

declare

v\_empno empa.empno%type:='&请输入员工编号！';

v\_sal empa.sal%type;

begin

select sal into v\_sal from empa where empno=v\_empno;

if(v\_sal<2000)then

dbms\_output.put\_line('旧工资是'||v\_sal);

update empa set sal=sal+100 where empno=v\_empno;

select sal into v\_sal from empa where empno=v\_empno;

dbms\_output.put\_line('新工资是'||v\_sal);

else dbms\_output.put\_line('工资大于或等于2000');

end if;

exception

when no\_data\_found then DBMS\_OUTPUT.PUT\_LINE('该员工编号不存在');

when too\_many\_rows then DBMS\_OUTPUT.PUT\_LINE('赋值时输入的行数太多');

end;

**第六讲**

**存储过程：**

含义以及作用和sql server一样

关键字procedure

1. 创建存储过程

create or replace procedure 存储过程名

as /is

begin

语句块;

end;

1. **删除存储过程**

Drop procedure 存储过程名

1. 调用存储过程
2. exec 存储过程名； 只能在sql命令行中使用
3. Call 存储过程名；
4. Begin 存储过程名； end；

存储过程带参数时，不能指定其数据类型的长度

输入参数使用关键字in,默认就是输入，可以省略。

输出参数使用关键字out，

即使是输出参数也是输入参数 关键字 in out

输入实例：

create or replace procedure pro\_2(v\_a in number,v\_b in number)

as

begin

declare v\_sum number(10);

begin

v\_sum:=v\_a+v\_b;

dbms\_output.put\_line(v\_sum);

end;

end;

输出实例：

create or replace procedure pro\_4(v\_sum out number)

as

begin

declare v\_a number:=1;

v\_b number:=2;

begin

v\_sum:=v\_a+v\_b;

end;

end;

调用存储过程：

declare v\_c number;

begin

pro\_4(v\_c);

dbms\_output.put\_line(v\_c);

end;

即使是输入参数也是输出参数的实例：

create or replace procedure pro\_5(v\_empnosal in out number)

as

begin

select sal into v\_empnosal from emp where empno=v\_empnosal;

dbms\_output.put\_line('该员工工资是'||v\_empnosal);

end;

调用存储过程：

declare v\_a number:='&123';

begin

pro\_5(v\_a);

end;

编写一个存储过程，从控制台输入一个员工编号从emp表中查询该员工的姓名

create or replace procedure pro\_3(v\_empno in emp.EMPNO%type)

as

begin

declare v\_name emp.ENAME%type;

begin

select ename into v\_name from emp where empno=v\_empno;

dbms\_output.put\_line(v\_name);

exception

when no\_data\_found then dbms\_output.put\_line('没有该员工编号');

end;

end;

调用存储过程

call pro\_3('&请输入员工编号')

Oracle中可以使用万能表dual，在不知道表是否存在的情况下，都可以使用该表，进行一切的查询。

**Oracle 函数 关键字 funciton**

1. 创建函数

Create or replace function fun\_函数名（参数 或不带参数）

Return 返回值的数据类型

As/is

Begin

Return 返回值；

End；

在创建函数时不能出现输出语句

实例：

create or replace function fun\_hello

return varchar2

as

begin

return 'Helloworld';

end;

1. 调用函数
2. Select 函数名（） from dual
3. 使用PL/SQL语句块的方式

调用函数方式：

select fun\_hello from dual

第二种调用函数方式：

declare v\_b varchar2(200);

begin

v\_b:=fun\_hello;

dbms\_output.put\_line(v\_b);

end;

实例创建不带参数函数：

create or replace function f\_date

return date

as

v\_a date;

begin

select sysdate into v\_a from dual;

return v\_a;

end;

select f\_date from dual

实例:创建带参数函数：

create or replace function f\_3(v\_a number,v\_b number)

return number

as

begin

return v\_a+v\_b;

end;

select f\_3(100,200) from dual

实例创建带参数函数并且带输出参数：

create or replace function f\_3(v\_a number,v\_b number,v\_sum out number)

return number

as

begin

v\_sum:=v\_a+v\_b;

return v\_sum;

end;

declare v\_a number;

begin

dbms\_output.PUT\_LINE(f\_3(500,700,v\_a));

end;

Floor(3.2)返回小于等于3.2的最大整数值 3

Ceil（4.8）返回大于等于4.8的最小整数数值 4

Sysdate 返回系统时间

Last\_day(sysdate)返回该月最后一天

select last\_day('1-6月-1997') from dual

Concat（）连接字符串

select concat('a','b') from dual

Inttcap()将每个字符串的首个字符改为大写

Upper()将整行字符串改为大写

Lower()将整行字符串改为小写

触发器：

是一个特殊事件的存储过程，触发器的调用不会实现系统的自动调用或者手动调用，需要通过某一个时间的事件去触发才能执行。

一般事件的触发只能在操作表的时候发生，比如insert/delete/update

1、创建触发器关键字：Trigger

Create or replace trigger 触发器名

Before/after insert/delete/update

On 表名

For each row 行级触发（删除了多少数据就提示多少次）

Begin

语句块；

End；

2、删除触发器

drop trigger 触发器名

触发器例子：

create or replace trigger tr\_test after update or delete on scott.TEST1 for each row

begin

if Updating then

insert into test\_update values(:new.id);

else if deleting then

insert into test\_del values(:new.id);

end if;

end if;

end;

如果使用了after关键字，则会在更新或者删除或者插入之前就执行触发器。

如果使用了before关键字，则会在更新之后的数据才执行触发器。例如更新，将表中的1值更新为2，则触发器会在上例子中插入2的值到对应的表中。

查询用户下的所有对象

select \* from user\_objects

Select \* from dba\_objects

备份还原方法关键字：exp

重要 !!! 导入用户的表和导出用户的表！！！

**索引的存在的作用是：如果ID列为索引，多个表中存在相同ID列，则修改根ID列，则其他会自动修改**

**视图的存在的作用：使数据简单化、表设计结构的补充、增加安全性**