ORACLE

凯盛软件

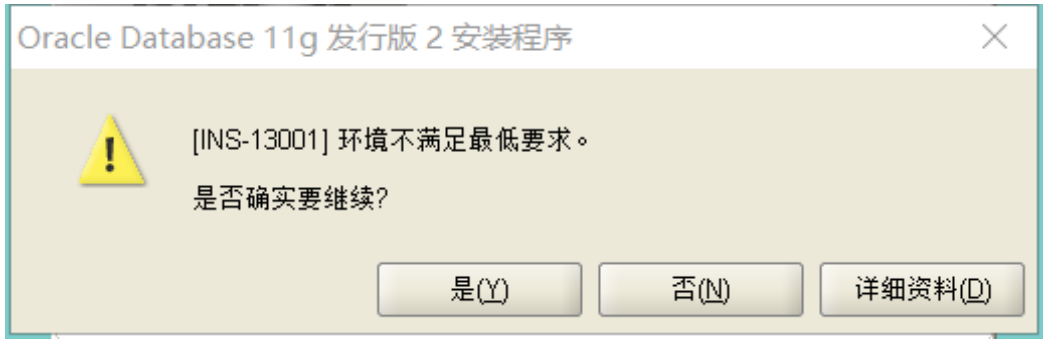
推荐书籍

- 官网
  - http://www.oracle.com
- 下载
  - http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html

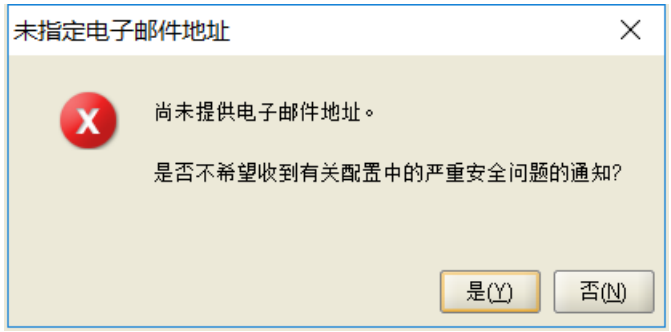
数据库安装

凯盛软件

- 下载Oracle11g的安装包file1、file2，合并file2包下的database\stage\Components的所有文件拷贝到file1对应的\database\stage\Components包下;
- 运行setup.exe,如果是win8、win10，请忽略如下异常：



- 配置安全更新：不填，并忽略异常提醒



数据库安装

凯盛软件

- 安装选项：创建和安装数据库
- 系统类：桌面类
- 安装典型：选择安装目录（不能含有中文和空格），企业版，填写数据库超级管理员的密码，密码一定要记住！
- 等待安装...
- 等到弹出Database Configuration Assistant窗口，单击口令窗口,解锁SCOTT用户，并设置密码；
- 安装成功

数据库卸载

凯盛软件

1. 暂停所有oracle服务
2. 开始菜单找到oracle，运行Universal Installer，单击全部展开，勾选除OraDb11g_home1外的其他项目并单击删除
3. Win+R输入regedit，打开注册表，删除如下oracle注册表：
   - 依次展开HKEY_LOCAL_MACHINE\SOFTWARE，找到oracle，删除
   - 依次展开HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services中，删除所有oracle开头的项
   - 依次展开HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application，删除所有oracle开头的项；
   - 在HKEY_CLASSES_ROOT，删除以ora开头的项
4. 重启电脑，删除oracle安装目录

Oracle服务

凯盛软件

- OracleServiceORCL：oracle的数据库启动的基础，只有该服务启动，数据库才可以使用
- OracleOraDb11g_home1TNSListener：该服务为客户端提供监听服务，只有该服务启动，客户端才能连接服务器

数据库配置向导

凯盛软件

- 启动Net Configuration Assistant，本地网络服务名配置；
- 以上配置写在Oracle HOME/product/11.2.0/dbhome_1/network/ADMIN/tnsnames.ora文件中，可以直接修改该文件

客户端

凯盛软件

- cmd：sqlplus 安装成功使用,展示用户show user ; 退出exit;

  sqlplus/nolog 不登录的情况下连接到服务器，然后使用conn 用户名/密码；

- 官方客户端工具：sqldeveloper
- 第三客户端方工具：plsqlDeveloer

用户管理

凯盛软件

- Oracle中不轻易创建数据库实例
- 使用不同的用户区分访问表的权限
- 创建用户：CREATE USER 用户名 IDENTIFIED BY 密码 ACCOUNT LOCK|UNLOCK
  - 只有切换管理员账户才能新建账户
  - 被锁定账户不能登录数据库
- 修改用户：
  - 修改用户密码：ALTER USER 用户名 IDENTIFIED BY 新密码
  - 修改用户的锁定状态(管理员)：ALTER USER 用户名 ACCOUNT LOCK|UNLOCK

用户权限

凯盛软件

- 系统权限： create session; create table……
  - GRANT 权限 TO 用户;
  - REVOKE 权限 FROM 用户;
- 数据库对象权限(对数据库表的CRUD操作)，授权某用户某表的相关权限
  - alert：允许使用者更改表或序列的定义
  - execute：允许使用者执行存储过程
  - index：允许使用者在表上建立索引
  - insert：允许使用者插入数据
  - references：允许使用者建立外键
  - select：允许使用者查询表、视图、序列等信息
  - update：允许使用者修改数据

  - 示例：授权：grant select on scott.emp to tom;

    撤销授权：revoke select on scott.emp from tom;

用户角色

凯盛软件

角色：若干个系统权限的集合

- CONNECT：主要应用在临时用户，不需要建表的用户，该角色能够连接数据库服务器
- RESOURCE：该角色可以创建表、序列、存储过程、触发器、索引等
- DBA：该角色拥有数据库的所有权限
- *一般情况下，普通用户拥有connect和resource角色即可

用户授权角色：

- GRANT 角色 TO 用户;

用户取消授权：

- REVOKE 角色 FROM 用户;

数据导入导出

在cmd中执行：（退出sqlplus登录）

- 导出：exp
- 导入：imp

Sql语句组成

凯盛软件

- DDL：数据库定义语言
  - create
  - alter
  - drop
- DML：数据操纵语言
  - insert
  - update
  - delete
  - select
- TCL：事务控制语言
  - commit;
  - savepoint;
  - rollback
- DCL：数据控制语言
  - grant
  - revoke

Oracle数据类型

凯盛软件

- 文本型
  - varchar2(n)：存放变长的字符串，长度为n字节，n最大可以到4000字节
  - nvarchar2(n)：存放变长的Unicode字符串，长度为n字节，最大为4000字节
  - char(n)：存放固定长度的字符串，长度为n字节，最大为2000字节
  - nchar(n)：存放固定长度的Unicode字符串，长度为n字节，最大为2000字节
- 数值型
  - number(n,[m])：可存放数值型数据，总长度为n位数（默认38位）；n最多38为数；此外number可以指定m位小数。
- 日期时间型
  - date：存放日期时间型，长度为7个字节
  - sysdate系统函数可以获取系统当前日期和时间:

```
select sysdate from dual
select 1+1 from dual
```

创建表

- 创建表语句（没有自动增长）

```
create table t_student(
        stuid number(10) not null,
        stuname nvarchar2(20) not null,
        age number(2) not null,
        address nvarchar2(200),
        codenum nvarchar2(18)
);
create table t_score(
        scoreid number(10) primary key,
        score number(3,1) not null,
        stuid number(10)
);
```

表约束

主键约束：alter table t_student add constraint PK_t_student primary key(stuid);

检查约束：alter table t_student add constraint ck_t_student_age check (age>18 and age<30);

默认约束：alter table t_student modify (address nvarchar2(200) default 'China');

唯一约束：alter table t_student add constraint un_t_student_codenum unique(codenum);

外键约束：alter table t_score add constraint fk_t_score_t_student foreign key(stuid) references t_student(stuid);

删除约束：alter table t_score drop constraint fk_t_score_t_student;

表修改

添加列：alter table t_student add (sex nvarchar2(2));

修改列：alter table t_student modify (codenum nvarchar2(20));

删除列：alter table t_student drop column sex;

重命名列：alter table t_student rename column age to nianling;

重命名表：rename t_student to t_xuesheng;

查看表结构

Desc table;

函数

每种数据库的函数都不尽相同

字符函数

LOWER：将字符转换为小写

UPPER：将字符转换为大写

INITCAP：首字符大写，其他字符小写

CONCAT：连接字符串 select contact(“ab”,loc) from dept 或者 ||

SUBSTR (column,startIndex,num) column：列名 startIndex：开始的索引 num：从索引起，获取几个字符

获取前三个字符：select ename,substr(ename,0,3) from emp

获取后两个字符：select ename,substr(ename,-2,2) from emp;

LENGTH：获取字符串长度

TRIM：去掉字符右边或左边的空格

LTRIM：去掉字符左边的空格

RTRIM：去掉字符右边的空格

select trim(ename),ltrim(ename),rtrim(ename) from emp;

数值函数

ROUND（num，n）：将列或表达式所表示的数值四舍五入到小数点后第n位

```
select round('49.255',1),round('49.255',2),round('49.255',-1) from dual;
```

TRUNC（num，n）：将列或表达式的数值截取到小数点后第n位

```
select trunc('49.255',1),trunc('49.255',2),trunc('49.255',-1) from dual;
```

MOD(m,n)：m除以n以后的余数

```
select mod(3,2) from dual;
```

日期函数

sysdate：获取当前系统日期及时间

select to_char(sysdate,'yyyy-MM-dd hh:mi:ss') from dual;

用户入职到现在的星期数:

select ename,round((sysdate-hiredate)/7,0) as weeks from emp ;

MONTHS_BETWEEN（date1，date2）：返回date1和date2之间的月份数量，结果为正数或负数

select months_between(sysdate,to_date('2016-01-01','yyyy-MM-dd')) from dual;

add_MONTHS(date,n)：向date加上n个月，n必须是整数

select add_months(sysdate,2) from dual;

next_day(date,char)：求出date之后一周内某天char的日期，char是一个有效表示星期几的数字或字符串

select next_day(sysdate,7) from dual;

last_day(date)：求出date所在月的最后一天

select last_day(sysdate) from dual;

数据类型转换为字符串

凯盛软件

to_char(date|number,[fmt])：把日期或数值型按照模式fmt转换为变长字符串

转换日期：select to_char(sysdate,'yyyy-MM-dd HH:mi:ss day d q') from dual;

yyyy：年 MM：月 dd：日 HH：小时 mm：分钟 ss：秒

day：星期几 d：本星期第几天 q：季度

转换数值： select to_char('89234334.239','L99,999,999.99') from dual;

L：本地货币符号 9：一个有效位 ,：千分位 .：小数点

获取月份：select to_char(sysdate,'MM') from dual;

获取天：select to_char(sysdate,'dd') from dual;

获取1981年12月的记录：select * from emp where to_char(hiredate,'MM') = '12' and
to_char(hiredate,'yyyy') = '1981';

数据类型转换为数值

to_number(char)：把一个由数字组成的字符串转换为数值

```
select to_number('12312') from dual;
```

数据类型转换为日期

模式fmt转换为日期

```
select * from emp where hiredate = to_date('1982/1/23','yyyy/MM/dd');
```

其他函数

nvl：显示当一个值为null时的默认值

```
select ename,nvl(mgr,'0') from emp;
```

聚合函数

注意count（*）和count（列）的区别

- avg
- count
- max
- min
- sum

分组函数

group by:

统计每个部门的平均销售额：

```
select avg(sal),deptno from emp where deptno is not null group by deptno;
```

显示平均销售额在2100以上的部门：

```
select avg(sal),deptno from emp where deptno is not null group by deptno having
avg(sal) > 2100;
```

** where 和 having区别？

数据库操作

凯盛软件

- 查询
- Where条件查询：
  - 逻辑运算符
    - and
    - or
    - not
  - 比较运算符：
    - =：等于
    - >：大于
    - >=：大于等于
    - <：小于
    - <=：小于等于
    - <>或!=：不等于
    - between... and...
    - in
    - Like % _
    - is null
- order by：
    - Desc降序排列
    - Asc 升序排列（默认）
- 例子： select * from emp where empno= 10 or empname= ‘james’ order by hiredate desc;

CRUD

注意事务的提交：

- INSERT
- UPDATE
- DELETE

做完增删改，必须commit;

select

凯盛软件

- 等值连接查询

  输出每人所在的部门及部门所在地：

  ```
  select ename,dname,loc from emp,dept where emp.deptno = dept.deptno;
  ```

  查询deptno为10的员工及部门所在地：

  ```
  select ename,dname,loc from emp,dept where emp.deptno = dept.deptno
  and dept.deptno = 10;
  ```

- 标准连接查询

  内连接：select ename,dname from emp inner join dept on emp.deptno = dept.deptno;

  左外连接： select ename,dname from emp left join dept on emp.deptno = dept.deptno;

  右外连接：select ename,dname from emp right join dept on emp.deptno = dept.deptno;

- 非等值查询

  根据员工的销售额来查询工资等级：

  ```
  select ename,grade,sal from emp,salgrade where emp.sal
  between salgrade.losal and salgrade.hisal;
  ```

- 外部连接

显示员工对应的部门，并显示所有部门:(右连接)

```
select ename,dept.dname,emp.empno,dept.deptno from emp,dept
        where emp.deptno(+) = dept.deptno;
```

显示员工对应的部门，并显示所有的员工：(左连接)

```
select ename,dept.dname,emp.empno,dept.deptno from emp,dept
        where emp.deptno = dept.deptno(+);
```

外部连接运算符(+)只能放在连接条件表达式的一侧

- 子查询

显示比id为7566销售额高的记录：

```
select * from emp where sal > (select sal from emp where empno = 7566);
```

显示工资最低的员工信息：

```
select * from emp where sal = (select min(sal) from emp);
```

子查询使用原则：

1.子查询要使用括号括起来

2.将子查询放在比较运算符的右边

3.不要在子查询中使用order by字句，select语句中只能有一个order by字句，并且它只能是主select语句的最后一个字句

- 替代变量

select * from emp where deptno = &deptno;

字符串替代变量需要使用单引号引起来：

select * from emp where job = '&job';

替代变量可以出现SQL语句的地方

WHERE条件

order by字句

列表达式

完整的select语句

示例：select ename,&column from emp where &where;

select ename,&column from emp where &where order by ℴ

&sql;

视图

视图的优点

1.限制对数据库的访问，因为视图可以有选择性的显示数据库的某一部分

2.通过简单的查询便能从复杂的查询的到结果

3.维护数据的独立性，尤其对于用户和应用程序来说，视图可以从多个表中检索数据

4.根据用户群组的不同要求，为他们提供数据访问

创建视图

赋予创建视图的权限：grant create view to scott;

创建视图：create view myview1 as select * from emp where deptno = 10;

定义视图的查询可以使用更复杂的select语句，包含连接，分组，子查询

定义视图的查询中不能包含order by语句,mysql允许，查询视图也是用order by，覆盖视图的order by

在视图的内部使用列的别名：

```
create view myview as select empno as id,ename as name,job from emp;
```

在创建视图时指定列的别名：

```
create view myview(maxsal,minsal,avgsal) as select max(sal),min(sal),avg(sal) from emp;
```

删除视图：


```
drop view myview1;
```

查询视图的定义


```
select view_name , text from user_views;
```

重建视图

create or replace view myview as select * from emp;

优点是不用删除现有视图。

序列

凯盛软件

序列产生器可以为表中的记录自动产生序列值

序列是由用户创建数据库对象并且可以被多个用户共享

序列的典型应用是产生主键值，用于标示记录的唯一性

序列产生及存储和表无关，因此相同的序列可以用于多个表

创建、删除序列

凯盛软件

创建序列：
```
create sequence id_seq
       increment by 1
       start with 1
       maxvalue 999999999
       nocache
       nocycle;
```

删除序列：
```
drop sequence id_seq;
```

获取序列信息

凯盛软件

```
select sequence_name,max_value,min_value,increment_by,last_number from user_sequences;
```

sequence_name：序列名称

max_value：最大的值

min_value：最小的值

increment_by：增长量

last_number：使用的或缓存的最后一个序列号，一般大于缓冲区中的最后一个值

nextval和currval伪列

凯盛软件

nextval用于获取序列中下一个有效的序列值：select id_seq.nextval from dual;

currval用于获取序列中当前的有效序列值，在使用之前必须先使用nextval获取序列值：

```
select id_seq.currval from dual;
```

可用使用的情况

1.select语句的select列表中，不包含子查询

2.insert语句的values子句中

3.update的set子句中

序列的使用：

```
insert into t_student(stuid,stuname,nianling,address)
        values(id_seq.nextval,'tom','23','china');
```

序列的修改

凯盛软件

```
alter sequence id_seq
        increment by 1
        maxvalue 999999999
        nocache
        nocycle;
```

注意：1.修改序列不能修改start with项
2.maxvalue不能比原有的值小

索引

1.使用索引可以加快对数据的查询

2.索引可以提供对表中记录直接快速的访问，索引的目的就是通过指针快速定位数据的方法有效的减少磁盘的IO操作。

3.Oracle服务器自动使用和维护索引，一旦创建了索引就不需要人工干预

创建索引：

自动：当在表中定义了primary key和unique约束，Oracle则会自动建立索引，索引名称和约束名称相同。

手动创建：用户可以在列上创建非唯一索引（用户可以创建唯一索引，但不推荐，唯一索引应该由unique约束来自动创建）

创建索引

适合创建索引的情况

1.经常用于WHERE子句或作为连接条件的列

2.所含数据值范围比较大的列

3.含有大量空值的列

4.经常同时用于一个where子句或连接条件的两个或多个列

5.绝大多数情况下只查询出总记录的2%~4%的表

不适合创建索引的情况

1.表数据量很小

2.列很少在查询中作为条件

3.频繁更新的表

4.绝大多数情况下查询的数据量大于总记录的2%~4%的表

创建索引

自动：当在表中定义了primary key和unique约束，Oracle则会自动建立索引，索引名称和约束名称相同。

手动创建：用户可以在列上创建非唯一索引（用户可以创建唯一索引，但不推荐，唯一索引应该由unique约束来自动创建）

```
create index emp_ename_index on emp(ename);
```

emp_ename_index：索引名称

emp：表名

ename：列名

索引操作

查询索引：select * from user_indexes;

删除索引：drop index emp_ename_index;

触发器

Insert触发器：

```
create or replace trigger save_before_student
after insert on t_student
for each row
begin
dbms_output.put_line('刚插入的id为：' || to_char(:new.stuid));
end;
```

打印之前需要设置：set serveroutput on

delete触发器:

```
create or replace trigger del_student_id
after delete on t_class
for each row
begin
delete from t_student where cid = :old.id;
end;
```

update触发器：

```
create or replace trigger update_before_class
after update on class
for each row
begin
update student set cid = :new.id where cid = :old.id;
end;
```

PL/SQL

PL/SQL块是SQL的一种增强，类似于mysql的存储过程，是一种程序语言，叫做过程化SQL语言（Procedural Language/SQL），在普通sql的基础上增加编程语言的特点，实现数据库语言和过程语言的联合，通过逻辑判断、循环等操作实现复杂的功能或者计算。Oracle独有。

PL/SQL块的组成部分

声明部分：该部分包含类变量和常量的定义，以及变量和常量的初始值定义，这部分由关键字declare开始，如果PL/SQL块中不需要声明变量或常量，该部分可以省略

执行部分：执行部分是PL/SQL的执行指令部分，由关键字begin开始，关键字end结尾。所有可执行的SQL语句都放在这一部分，该部分不能省略，end关键字后面使用分号结尾

异常处理部分：该部分是可选的，该部分使用exception关键字将可执行部分分为两个部分，一部分是正常运行的程序，一旦出现异常就跳转到异常部分执行

PL/SQL中符号的声明

凯盛软件

赋值运算符	:=						
字符串连接	||						
单行注释	--						
多行注释	/**/						
范围操作符	.. （1..5表示从1到5）						
算术运算符	+	-	*	/	** (幂操作 比如3**2=9)		
关系运算符	>	<	<=	>=	=	<>	!=
逻辑运算符	and	or	not				

变量声明

语法：变量名 数据类型[:=初始值]

声明变量：

```
declare name varchar(20) := 'alex';
--声明一个名称为name数据类型为varchar的变量，默认值为alex
begin
dbms_output.put_line(name);
--dbms_output.put_line()用于将变量输出到控制台上，
如果没有输出可以通过set serveroutput on设置输出
end;
```

变量的赋值


```
declare p_v1 int;
begin
p_v1 := 10;
select 123 into p_v1 from dual;
dbms_output.put_line(p_v1);
end;
```

通过:=操作符或者 通过select ....into 变量名 from..来赋值

注意：查询的结果只能是一行，如果多行或没有行，则会引发异常

常量声明


```
declare c_pi constant float := 3.14;
r int := 4;
area float;
begin
area := c_pi * r * r;
dbms_output.put_line('AREA:' || area);
end;
/
```

通过constant关键字来声明常量

特殊数据类型

凯盛软件

%ROWTYPE

该数据类型表示一条记录，相当于Java中的一个对象，可以共通过"."点操作符来访问记录中的属性

```
declare emprow emp%ROWTYPE;
begin
select * into emprow from emp where empno=7934;
dbms_output.put_line(emprow.ename);
end;
/
```

%TYPE

引用某个变量或数据库的列作为数据类型来声明一个变量

```
declare myname emp.ename%TYPE;  /*使用emp表中的ename列的数据类型*/
begin
select ename into myname from emp where empno = 7934;
dbms_output.put_line(myname);
end;
```

条件控制和循环结构

条件分支语句


```
declare
myval number(10) := 10;
begin
if myval > 10 then
  dbms_output.put_line('>10');
elsif myval = 10 then
  dbms_output.put_line('=10');
else
  dbms_output.put_line('<10');
end if;
end;
```

循环结构

凯盛软件

loop ... end loop循环

```
declare temp number(3) := 0;
 total number(5) := 0;
 begin
  loop
   temp := temp + 1;
   total := total + temp;
   if temp >= 100 then
    exit;
   end if;
    /*或者采用exit when temp >= 100;*/
  end loop;
  dbms_output.put_line('total:'||to_char(total));
end;
```

注意：使用exit退出循环

while循环

```
declare temp number(3) := 0;
total number(5) := 0;
begin
  while temp<100
   loop
      temp := temp +1;
      total := total +temp;
   end loop;
   dbms_output.put_line('total' ||to_char(total));
end;
```

for循环

```
declare temp number(3) := 0;
total number(5) := 0;
 begin
   for temp in 1..100 loop
         total := total + temp;
 end loop;
 dbms_output.put_line('total:' || to_char(total));
end;
```

存储过程

普通存储过程

```
create or replace procedure my_proc9
is
begin
dbms_output.put_line('hello,Oracle');
end;
```

带参的存储过程

```
create or replace procedure proc2(temp_id in emp.empno%TYPE)
is
name emp.ename%TYPE;
begin
select ename into name from emp where emp.empno = temp_id;
dbms_output.put_line(name);
end;
```

带返回值得存储过程

```
create or replace procedure proc3(temp_id in emp.empno%TYPE,temp_name out emp.ename%TYPE)
is
begin
select ename into temp_name from emp where emp.empno = temp_id;
end;

--调用--
declare name emp.ename%TYPE;
begin
proc3(7934,name);
dbms_output.put_line(name);
end;
```

操作存储过程

凯盛软件

执行存储过程：execute 存储过程名称(参数列表):

```
execute my_proc;
exec my_proc;
```

删除存储过程：

```
drop procedure 存储过程名称;
```

Oracle and Maven

凯盛软件

手动添加jar到本地仓库：

```
mvn install:install-file -DgroupId=com.oracle -DartifactId=ojdbc14 -Dversion=10.2.0.4.0 -Dpackaging=jar -Dfile=E:\ojdbc14-10.2.0.4.0.jar
```

Oracle and java

凯盛软件

驱动：

```
oracle.jdbc.driver.OracleDriver
```

连接字符串：

```
jdbc:oracle:thin:@127.0.0.1:1521:ORCL
```

SQL中使用序列：

```
String sql = "select * from emp";
```