# 第一天

## Oralce简介

Oracle公司名称 甲骨文 常见产品 mysql 、jdk、virtualBox,oracle(数据库)

1. Oracle是大型数据库而Mysql是中小型数据库，
2. Mysql是开源免费的而Oracle价格非常高。
3. Oracle支持大并发，大访问量，是OLTP(处理数据实时性强)最好的工具。
4. Mysql安装完后才152M而Oracle有3G左且使用的时候Oracle占用特别大的内存空间和其他机器性能。

Oracle数据库应用工业、金融、商业、保险等领域

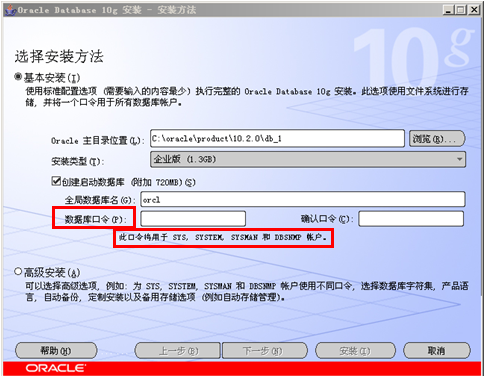
Redis mongodb hbase hive

## Oracle安装

1. Win7以上安装考虑兼容性
2. 虚拟机Winxp安装oracle（为了防止安装出错推荐这个方式）

见文档

此处的数据库口令是 sys,system等超级管理员的密码



## Oracle基本概念上

## Oracle基本概念下

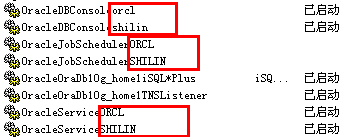
1. 数据库：

Oracle数据库是数据的物理存储，这就包括（数据文件ORA或者DBF、控制文件、联机日志、参数文件）。

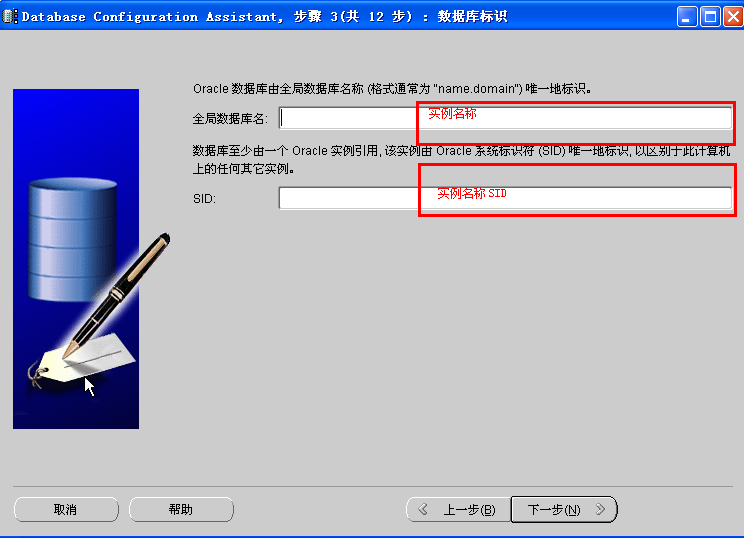
1. 实例：

一个Oracle实例（Oracle Instance）有一系列的后台进程（Backguound Processes)和内存结构（Memory Structures)组成。一个数据库可以有n个实例。

例如连个实例 orcl和 shilin



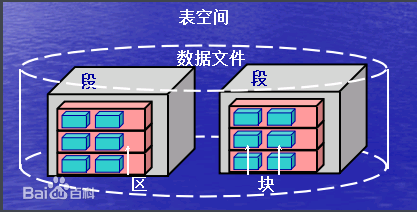
可以使用创建实例，安装过程中体现为全局数据库



1. 用户:相当于mysql的数据库，理解-每个用户有自己的表
2. 表空间：

表空间是Oracle对物理数据库上相关数据文件（ORA或者DBF文件）的逻辑映射。一个实例在逻辑上被划分成一到若干个表空间，每个表空间包含了在逻辑上相关联的一组结构。

1. 数据文件：存储表中数据dbf、ora文件
2. 数据段：数据段是与数据库对象相对应，一般一个数据库对象对应一个数据段



## Oracle基本命令

登陆

Sqlplus 用户名/密码 [as sysdba]

如果是普通用户不需要as sysdba

show user

conn 用户名/密码 [as sysdba]

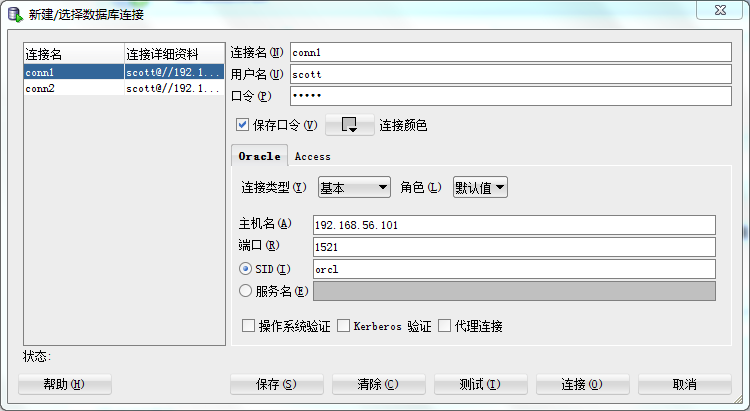
## Oracle的SQL简介

安装客户端：

Sql developer

PL/SQL

Sql developer 连接远程oracle数据库



常用数据类型

Number(4) 最大长度为4为的数值类型,number(5,2)

Varchar2(10) 最大长度为10的字符串，varchar2用于存储可变长度的字符串，.varchar2把所有字符都占两字节处理(一般情况下)，varchar只对汉字和全角等字符占两字节，数字，英文字符等都是一个字节，VARCHAR2把空串等同于null处理，而varchar仍按照空串处理；建议在oracle中使用varchar2

Number(7,2) 数值类型整数位占5位，小数位占2位，一共7位。

Date 时间类型

Mysql 与 oracle sql语句不互相兼容

**DML**(数据库操作语言): 其语句包括动词[INSERT](http://baike.baidu.com/view/270874.htm" \t "_blank)，[UPDATE](http://baike.baidu.com/view/1028155.htm" \t "_blank)和[DELETE](http://baike.baidu.com/view/4794.htm" \t "_blank)。它们分别用于添加，修改和删除表中的行。也称为动作查询语言。

**DDL**(数据库定义语言): 其语句包括动词CREATE和DROP。在数据库中创建新表或删除表（CREAT TABLE 或 DROP TABLE）；为表加入索引等。DDL包括许多与人[数据库目录](http://baike.baidu.com/view/4369490.htm" \t "_blank)中获得数据有关的保留字。它也是动作查询的一部分。

**DCL**(数据库控制语言):它的语句通过GRANT或REVOKE获得许可，确定单个用户和用户 组对[数据库对象](http://baike.baidu.com/view/50913.htm" \t "_blank)的访问。某些RDBMS可用GRANT或REVOKE控制对[表单](http://baike.baidu.com/view/296684.htm" \t "_blank)个列的访问。

## Oracle 的简单查询

|  |
| --- |
| select \* from emp;  select empno, ename, job from emp;  select empno as 员工编号, ename as 员工姓名 from emp;  select empno eno, ename en from emp; --常用  select distinct job from emp;  select distinct ename, job, deptno from emp;  --只有所有的列全相同才被认为是相同的行可以合并  select '员工编号是' || empno || '姓名是' || ename || '的工作是'|| job from emp;  -- Mysql字符串连接 concat('11','22','33') =>  ‘112233’  select ename, sal\*12 from emp; |

## Oracle限定查询上

|  |
| --- |
| select \* from emp where comm is not null;  select \* from emp where comm is null;  select \* from emp where sal <= 1500 and comm is null;  --相同  select \* from emp where not(sal > 1500 or comm is not null);  select \* from emp where sal between 1500 and 3000;  --相同  select \* from emp where sal >= 1500 and sal <= 3000;  select \* from emp where ename = 'SMITH';  --sql语句不区分大小写，但是表中的值是区分 |

## Oracle限定查询下

|  |
| --- |
| select \* from emp where empno = 7369 or empno = 7499 or empno = 7521;  --相同  select \* from emp where empno in (7369, 7499, 7521);  --在LIKE中主要使用以下两种通配符  --“%”：可以匹配任意长度的内容  --“\_”:可以匹配一个长度的内容  select \* from emp where ename like '\_M%';  select \* from emp where empno != 7369;  --相同  select \* from emp where empno <> 7369; |

## Oracle排序

|  |
| --- |
| select \* from emp order by sal ; --默认正序（从小到大）  select \* from emp order by sal asc; --正序  select \* from emp order by sal desc; --倒叙  select \* from emp order by sal desc, hiredate asc; --sal倒序，hiredate正序 |

## Oracle单行函数

|  |
| --- |
| --小写转大写  select upper('smith') from dual;  --大学转小写  select lower('SMITH') from dual;  --首字母大写  select initcap('smith') from dual;  --字符串连接  select concat('hello', 'world') from dual;  select 'hello' || 'world' from dual;--建议使用  --字符串截取  select substr('hello', 1, 3) from dual;--索引从1开始  select substr('hello', 0, 3) from dual;  --字符替换  select replace('hello', 'l', 'x') from dual;  --四舍五入  select round(12.48) from dual;  select round(12.487, 2) from dual;  --取整  select trunc(12.87) from dual;  select trunc(12.635, 2) from dual;  --取余  select mod(10, 3) from dual; |

## Oracle日期函数

日期加减时有一些规律

日期 – 数字 = 日期

日期 + 数字 = 日期

日期 – 日期 = 数字

数字指的是天数

|  |
| --- |
| select sysdate from dual; --获取系统时间  select ename, trunc((sysdate - hiredate)/7) from emp;  select ename, trunc(months\_between(sysdate, hiredate)) from emp;  select add\_months(sysdate, 6) from dual;  select next\_day(sysdate, '星期日') from dual;-- 系统地区为简体中文  select next\_day(sysdate, 'MONDAY') from dual; --系统地区为英文  select last\_day(sysdate) from dual; |

## Oracle转换函数

|  |
| --- |
| select ename,  to\_char(hiredate, 'yyyy') 年,  to\_char(hiredate, 'mm') 月,  to\_char(hiredate, 'dd') 日,  to\_char(hiredate, 'HH') 时,  to\_char(hiredate, 'mi') 分,  to\_char(hiredate, 'ss') 秒  from emp;  select to\_char(sysdate, 'yyyy-mm-dd HH:mi:ss') from dual;  select ename, to\_char(hiredate, 'yyyy-mm-dd') from emp;  select ename , to\_char(sal, 'l99,999') from emp;  select to\_number('10') + to\_number('10') from dual;  select to\_date('1985-04-22', 'yyyy-mm-dd') from dual; |

## Oracle通用函数

|  |
| --- |
| --nvl与mysql总的ifnull相同  select ename, sal, nvl(comm, 0) pcomm from emp;  --decode该函数类似if....else if...esle  select ename,  decode(job,  'CLERK','业务员',  'SALESMAN','销售',  'MANAGER','经理',  'ANALYST','分析员',  'PRESIDENT','总裁',  '无业') cjob  from emp;  select ename,  (case  when job = 'CLERK' then  '业务员'  when job = 'SALESMAN' then  '销售'  when job = 'MANAGER' then  '经理'  when job = 'ANALYST' then  '分析员'  when job = 'PRESIDENT' then  '总裁'  else  '无业'  end) cjob  rom emp; |

## Oracle多表联查上

|  |
| --- |
| ---查询出每一个员工的上级领导 mysql可用  select e.empno, e.ename, e1.empno, e1.ename  from emp e, emp e1  where e.mgr = e1.empno; |

## Oracle多表联查下

|  |
| --- |
| --查询出每个员工编号，姓名，部门名称，工资等级和他的上级领导的姓名，工资等级  select e.empno,  e.ename,  d.dname,  decode(s.grade,  1,'一级',  2,'二级',  3,'三级',  4,'四级',  5,'五级',  '无级') egrade,  e1.ename,  decode(s1.grade,  1,'一级',  2,'二级',  3,'三级',  4,'四级',  5,'五级',  '无级') e1grade  from emp e, dept d, salgrade s, emp e1, salgrade s1  where e.deptno = d.deptno  and e.sal between s.losal and s.hisal  and e.mgr = e1.empno  and e1.sal between s1.losal and s1.hisal |
|  |

## Oracle外连接上

Sql1992写法mysql不支持

|  |
| --- |
| --做外连接查询的时候全量表在左端就是左连接 左表全量  select \* from dept d, emp e where d.deptno = e.deptno(+);  --做外连接查询的时候全量表在右端就是右连接 右表全量  select \* from emp e, dept d where e.deptno(+) = d.deptno; |

## Oracle外链接下

|  |
| --- |
| --查询员工的编号和姓名和他的上级领导的编号和姓名  select e.empno, e.ename, e1.empno, e1.ename  from emp e, emp e1  where e.mgr = e1.empno(+); |

# 第二天

## Oracle的SQL1999语法

Sql1999 mysql oracle 都支持

交叉连接

|  |
| --- |
| select \* from emp e cross join dept d;  --相同  select \* from emp , dept; |

自然连接（自动匹配条件）

|  |
| --- |
| select \* from emp natural join dept;  --类似  select \* from emp e, dept d where e.deptno = d.deptno; |

USING

|  |
| --- |
| select \* from emp join dept using(deptno) where deptno = 20;  --相同  select \* from emp e, dept d where e.deptno = d.deptno and d.deptno = 20; |

ON

|  |
| --- |
| select \* from emp e join dept d on e.deptno = d.deptno ;  --相同  select \* from emp e , dept d where e.deptno = d.deptno ; |

Sql1999和sql1992

|  |
| --- |
| select \* from dept d left join emp e on d.deptno = e.deptno;  --oracle可以互换使用  select \* from dept d, emp e where d.deptno = e.deptno(+);  select \* from emp e right join dept d on e.deptno = d.deptno;  --oracle可以互换使用  select \* from emp e, dept d where e.deptno(+) = d.deptno; |

## Oracle分组函数

|  |
| --- |
| select \* from emp;  select count(\*) from emp;  select count(1) from emp;  select count(empno) empnum from emp; --建议使用  select min(sal) minsal from emp;  select max(sal) maxsal from emp;  select avg(sal) avgsal from emp;  select sum(sal) sumsal20 from emp where deptno = 20; |

## Oracle分组统计上

|  |
| --- |
| select count(empno), deptno from emp group by deptno;  deptno 有几种不同的值分几组，deptno值相同的数据为一组数据 |

## Oracle分组统计下

|  |
| --- |
| select count(\*), deptno from emp ; deptno匹配存在分歧，出错  select count(\*), d.deptno, d.dname, d.loc  from emp e, dept d  where e.deptno = d.deptno  group by d.deptno,d.dname, d.loc; |

## Oracle分组统计条件查询

Having对分组后的数据过滤

|  |
| --- |
| select sum(e.sal), e.job  from emp e  where e.job <> 'SALESMAN'  group by e.job  having sum(e.sal) > 5000 --对分组后的数据过滤  order by sum(e.sal) asc --对分组后的数据排序 |

## Oracle子查询

中间表的别名作为伪列

|  |
| --- |
| select t.\*, d.dname  from emp t,  (select min(e.sal) minsal, e.deptno from emp e group by e.deptno) a,  dept d  where t.sal = a.minsal  and t.deptno = a.deptno  and t.deptno = d.deptno; |

## Oracle的exits的使用

Where根据真假boolean作为判断条件

|  |
| --- |
| --用exists来实现查询出有员工的部门有哪些？  select \* from dept t where exists (select \* from emp e where e.deptno = t.deptno);  --反则  select \* from dept t where not exists (select \* from emp e where e.deptno = t.deptno); |

## Oracle并集union和union all的使用

|  |
| --- |
| --union把两个集合做并集的时候会把重复的数据去掉留一份  select \* from emp t where t.sal > 1000  union  select \* from emp t where t.sal > 2000;  --union all 把两个集合做并集的时候会不去重  select \* from emp t where t.sal > 1000  union all  select \* from emp t where t.sal > 2000;  --并集时要合并的列类型，列数必须一致，列名可以不一致, |

## Oracle的增删改

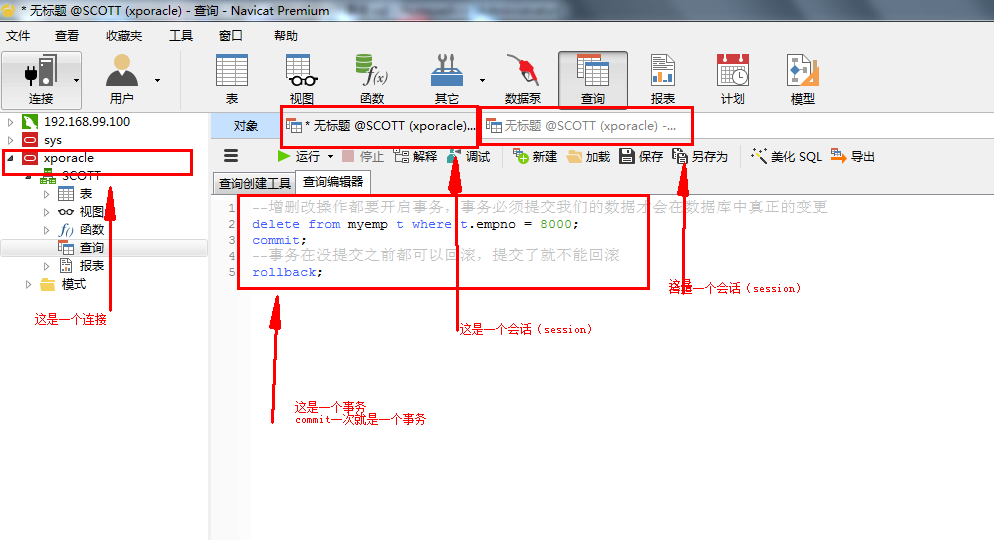
|  |
| --- |
| create table myemp select \* from emp  insert into myemp  (empno, ename, job, mgr, hiredate, sal, comm, deptno)  values  (8000,  'kobe',  'sport',  7369,  to\_date('1978-12-11', 'yyyy-mm-dd'),  1000,  100,  10);  update myemp t set t.ename = 'smith', t.sal = 1000 where t.empno = 7369;  insert into myemp select \* from emp |

## Oracle数据库事务

查询操作不存在事务问题

|  |
| --- |
| --增删改操作都要开启事务，事务必须提交我们的数据才会在数据库中真正的变更  delete from myemp t where t.empno = 8000;  commit;  --事务在没提交之前都可以回滚，提交了就不能回滚  rollback;  select \* from myemp;  --执行update时没有提交之前事务是挂起状态，这时这条数据被锁住  update myemp t set t.sal = t.sal + 100 where t.empno = 7369; |

**连接，会话（session），事务的关系**



项目中

连接池中的连接就是图中的连接

Session会话就是图中的会话

Spring事务管理的事务就是指事务

## Oracle表管理

常用的数据类型

|  |  |
| --- | --- |
| Varchar， varchar2 | 表示一个字符串 |
| NUMBER | NUMBER(n)表示一个整数，长度是n |
| NUMBER(m,n):表示一个小数，总长度是m，小数是n，整数是m-n |
| DATA | 表示日期类型 |

|  |
| --- |
| --建表  create table person(  person\_id number(4),  pname varchar2(10),  age number(3),  birthday date  );  insert into person  (person\_id, pname, age, birthday, address)  values  (1000, '张三', 20, to\_date('1985-04-22', 'yyyy-mm-dd'), 'beijing');  select \* from person;  --删表  drop table person;  --添加列  alter table person add(address varchar2(50));  alter table person modify(address varchar2(5)); |

## Oracle主键约束和非空约束

|  |
| --- |
| --谨慎使用 无法回滚  truncate table person; |

主键约束

|  |
| --- |
| create table person(  person\_id number(4) primary key,--主键是唯一的，不可为空  pname varchar2(10),  age number(3),  birthday date  ); |

非空约束

|  |
| --- |
| create table person(  person\_id number(4) ,--主键是唯一的，不可为空  pname varchar2(10) not null,  age number(3),  birthday date,  constraint person\_person\_id\_pk primary key(person\_id)  ); |

## Oracle唯一性约束

唯一约束

|  |
| --- |
| create table person(  person\_id number(4) ,--主键是唯一的，不可为空  pname varchar2(10) unique,  age number(3),  birthday date  ); |

## Oracle检查约束

检查约束

|  |
| --- |
| create table person(  person\_id number(4) ,  pname varchar2(10),  gender number(1) check (gender in(1, 2)),  birthday date  ); |

## Oracle外键约束

外键约束

|  |
| --- |
| create table orders(  order\_id number(4),  totalPrice number(7,2),  order\_time date,  constraint orders\_order\_id\_pk primary key(order\_id)  ); |

## Oracle的rownum和分页

Rownum 不支持大于号

|  |
| --- |
| 显示rownum  select rownum, t.\* from emp t  分页模板 rownum作为中间表的一列支持大于号  select \* from (select rownum rm, a.\* from (  select \* from emp –自己写  ) a where rownum <  11 –结束行号  ) b where b.rm >  5 –起始行号 |

开发中 pageNo ,pageSize 已知 计算出

startNum = (pageNo - 1)\*pageSize;

endNum = pageNo\*pageSize + 1;

## Oracle视图

应用：

经常用到的查询，或较复杂的联合查询应当创立视图，能够优化性能的  
还有就是涉及到权限管理方面，比如某表中的部分字段含有机密信息，不应当让低权限的用户访问到的情况，这时候给这些用户提供一个适合他们权限的视图，供他们阅读自己的数据。

修改视图表的字段值也会被修改。一般不会去修改视图，设置视图为只读。

|  |
| --- |
| --创建只读的视图  create or replace view view\_d20 as select \* from myemp t where t.deptno = 20 with read only; |

## Oracle序列

作用：生成主键

|  |
| --- |
| --序列虽然是给某个表使用，但是序列并没有绑定字某一张表，任何一张表使用这个序列都可以  create sequence seqpersonid;  --查询序列的下一个值(重点) 生成主键  select seqpersonid.nextval from dual;  --查看当前值是多少  select seqpersonid.currval from dual;  insert into person  (person\_id, pname, gender, birthday)  values  (seqpersonid.nextval, '张三', 1, sysdate);    select \* from person; |

## Oracle索引

理解：一本书的目录

创建规则：简单（单例）粗暴

查询规则严格符合索引创建顺序才能使用到索引

|  |
| --- |
| --创建单例索引  create index pname\_index on person(pname);  --索引不需要主动使用，根据索引列查询时就自动的使用了索引  select \* from person t where t.pname = '张三';  --复合索引,查询数据使用的时候是有索引顺序的  create index pg\_index on person(pname, gender);  select \* from person t where t.pname = '张三' and t.gender = 1;  --下面这sql就不会使用pg\_index  select \* from person t where t.gender = 1 and t.pname = '张三'; |

# 第三天

## Oracle高级PLSQL简介

MySQL在5.0以前并不支持存储过程

## Oracle高级PLSQL变量

语法固定

declare

说明部分 （变量说明，游标申明，例外说明 〕

begin

语句序列 （DML语句〕…

exception（可选）

例外处理语句

End;

常量

married boolean:=true

age number:=20 赋值用 :=

变量

基本变量

(建表时字段的变量)char, varchar2, date, number, boolean, long

引用变量

Myname emp.ename%type; Myname变量的类型和emp表ename的类型相同

记录型变量

Emprec emp%rowtype；Myname变量的类型和emp表一条记录的各个字段类型相同

## Oracle高级PLSQL的if

## Oracle高级PLSQL的循环loop

## Oracle高级PLSQL的游标上

## Oracle高级PLSQL的游标下

## Oracle高级PLSQL例外

## Oracle高级PLSQL存储过程

## Oracle高级PLSQL程序调用存储过程

## Oracle高级PLSQL触发器上

## Oracle高级PLSQL触发器下

## Oracle数据库建模单表

## Oracle数据库建模表一对多和多对一

## Oracle数据库建模一对一

## Oracle数据库建模多对多

## Oracle数据库模型和表的转换

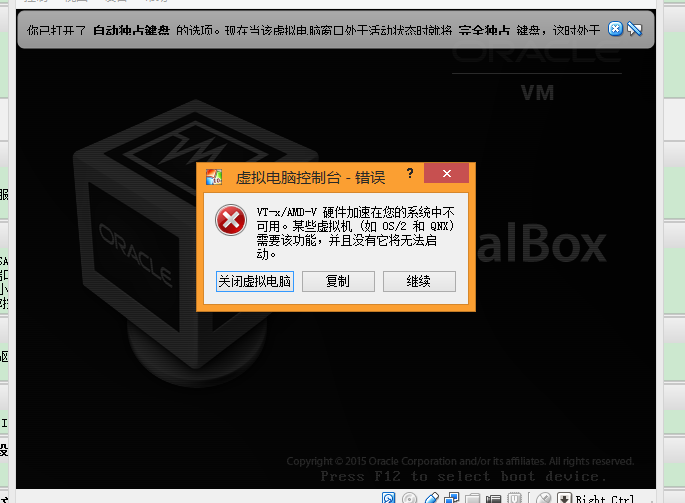
## Oracle数据库表数据的导出

## Oracle数据库用户的导入导出

# 常见问题汇总

## Xp系统无法启动

1. virtualbox虚拟机提示 “VT-x/AMD-V 硬件加速在的系统中不可用。类似这样的提示



解决方法 使cpu支持虚拟化技术Intel VT技术的。

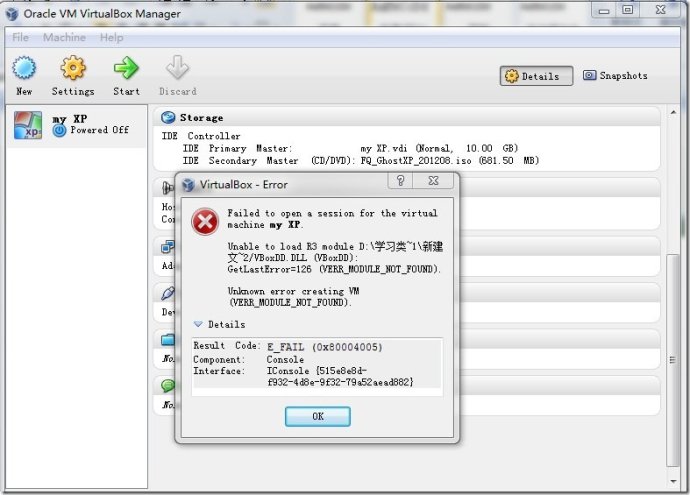
 开机进入BIOS选项 ，依次选Config->CPU->Intel Virtualization Technology，里面有个Intel VT-d Feature ，改成Enabled ，保存退出，关机，然后启动机器。

虚拟机的配置，是在：

系统->硬件加速，中的 硬件虚拟，记得选上：

启用VT-x/AMD-V

1. 出现如下错误



原因：安装的64位WIN7系统主题被破解（ghost系统通常主题配破解），

windows\system32下themeservice.dll   themeui.dll   uxtheme.dll这个三个文件不是原版，改回原版

## Xp系统重启oracle无法连接到

1、ORA-27101: shared memory realm does not exist

提示ORA-01034: ORACLE not available   
    ORA-27101: **shared** **memory** **realm** does not exist

解决办法：  
用CMD进入命令行  
sqlplus /nolog 登录进了SQLPLUS,此时并没有登录数据库

conn / as sysdba 连接上数据库  
startup

“/”的解释:本身只是作为用户名和口令的一个分隔符，现在以操作系统认证的方式登录，不需要用户名和口令，但是这个分隔符不能少。