



ORACLE 数据库

第 1 章

ORACLE 基础

传智播客.黑马程序员



一、ORACLE 简介

(一) 什么是 ORACLE

ORACLE 数据库系统是美国 ORACLE 公司 (甲骨文) 提供的以分布式数据库为核心的一组软件产品，是目前最流行的客户/服务器(CLIENT/SERVER)或 B/S 体系结构的数据库之一。

ORACLE 通常应用于大型系统的数据库产品。

ORACLE 数据库是目前世界上使用最为广泛的数据库管理系统，作为一个通用的数据库系统，它具有完整的数据管理功能；作为一个关系数据库，它是一个完备关系的产品；作为分布式数据库它实现了分布式处理功能。

ORACLE 数据库具有以下特点：

- (1) 支持多用户、大事务量的事务处理
- (2) 数据安全性和完整性控制
- (3) 支持分布式数据处理
- (4) 可移植性

(二) ORACLE 体系结构

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)。其实 Oracle 数据库的概念和其它数据库不一样，这里的数据库是一个操作系统只有一个库。可以看作是 Oracle 就只有一个



大数据库。

2.实例

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

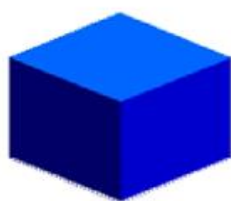
3.数据文件 (dbf)

数据文件是数据库的物理存储单位。数据库的数据是存储在表空间中的，真正是在某一个或者多个数据文件中。而一个表空间可以由一个或多个数据文件组成，一个数据文件只能属于一个表空间。一旦数据文件被加入到某个表空间后，就不能删除这个文件，如果要删除某个数据文件，只能删除其所属于的表空间才行。

4.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。一个数据库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。每个数据库至少有一个表空间(称之为system表空间)。

每个表空间由同一磁盘上的一个或多个文件组成，这些文件叫数据文件(datafile)。一个数据文件只能属于一个表空间。



段



区



数据块

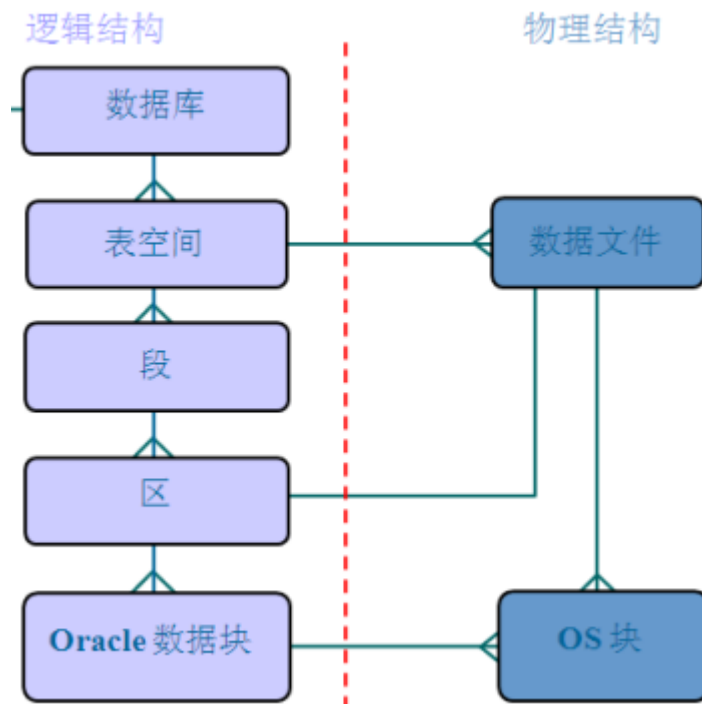


磁盘块

注：表的数据，是有用户放入某一个表空间的，而这个表空间会随机把这

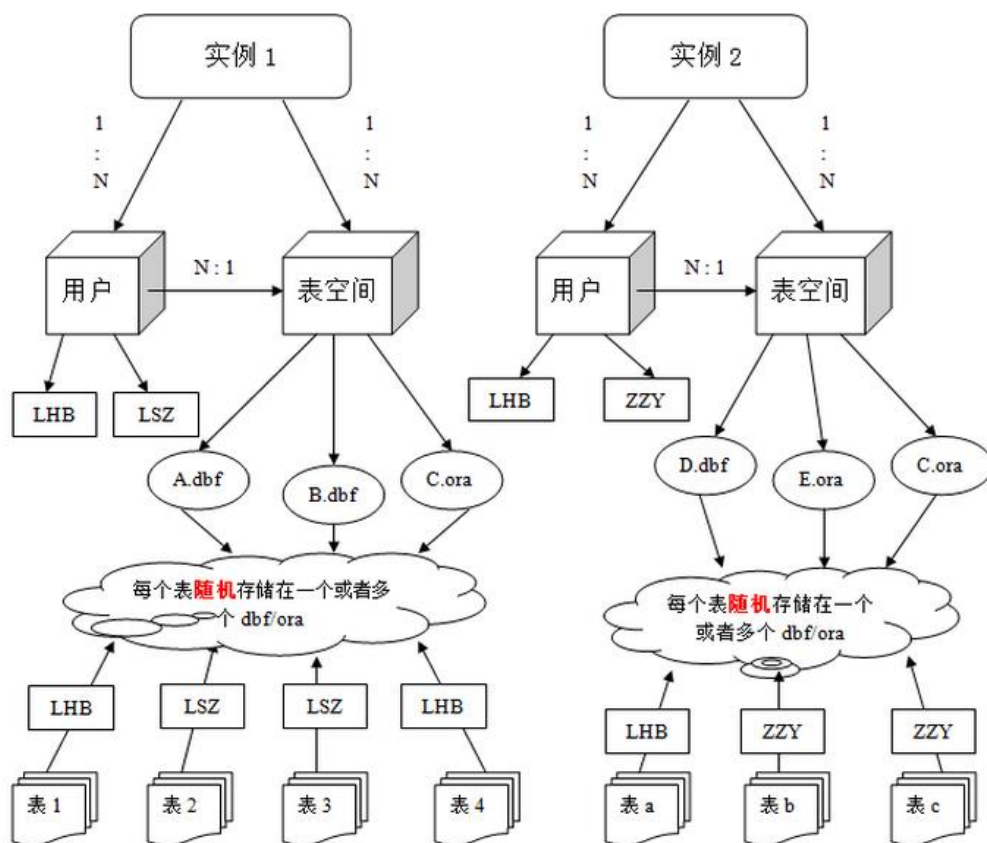
些表数据放到一个或者多个数据文件中。

由于 oracle 的数据库不是普通的概念，oracle 是有用户和表空间对数据进行管理和存放的。但是表不是有表空间去查询的，而是由用户去查的。因为不同用户可以在同一个表空间建立同一个名字的表！这里区分就是用户了！



5.用户

用户是在表空间下建立的。用户登陆后只能看到和操作自己的表，ORACLE 的用户与 MYSQL 的数据库类似，每建立一个应用需要创建一个用户。

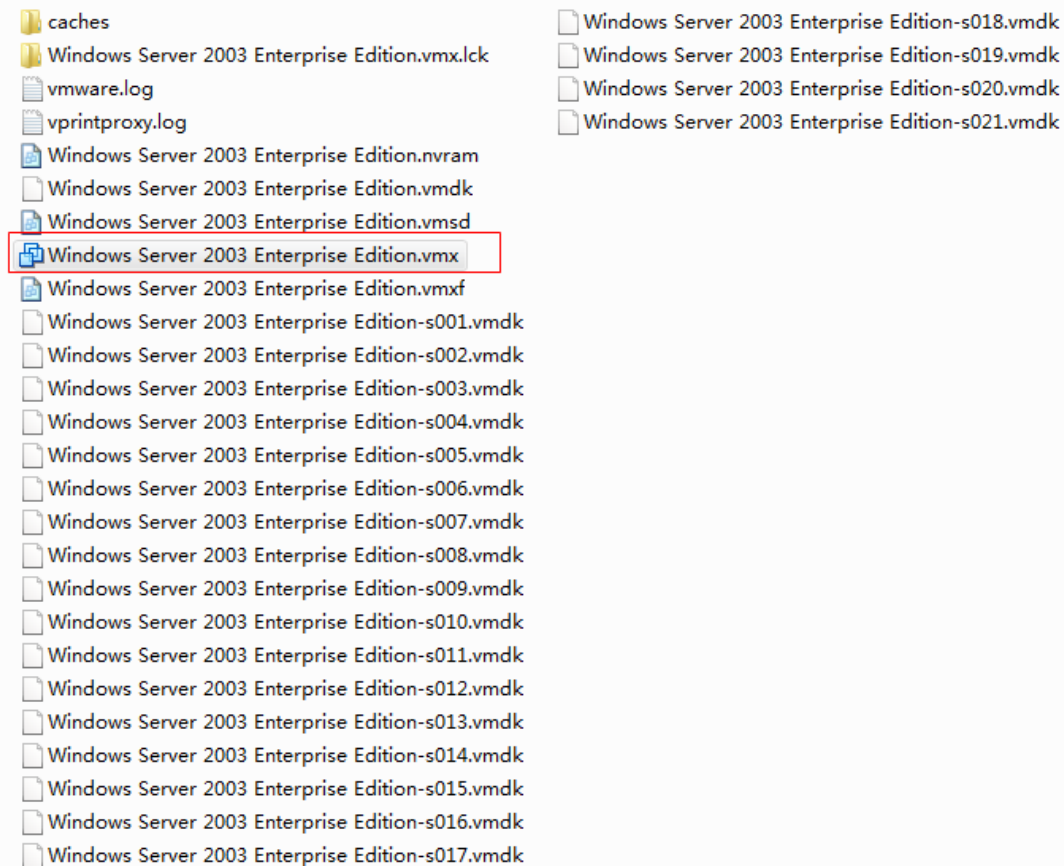


二、ORACLE 安装与配置

(一) VMware 挂载 windows server 2003

打开“资源”文件夹中的 windows2003 文件夹，双击扩展名为 vmx 的文件即可

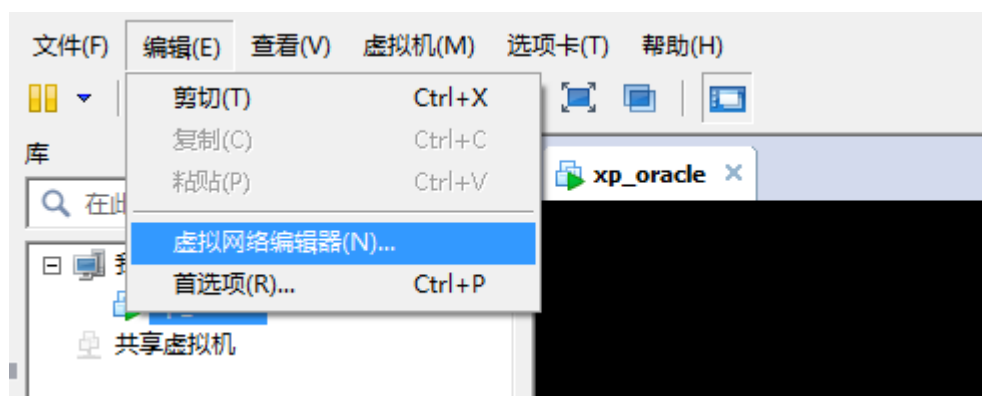
将 windows2003 系统挂载到 VMware 中



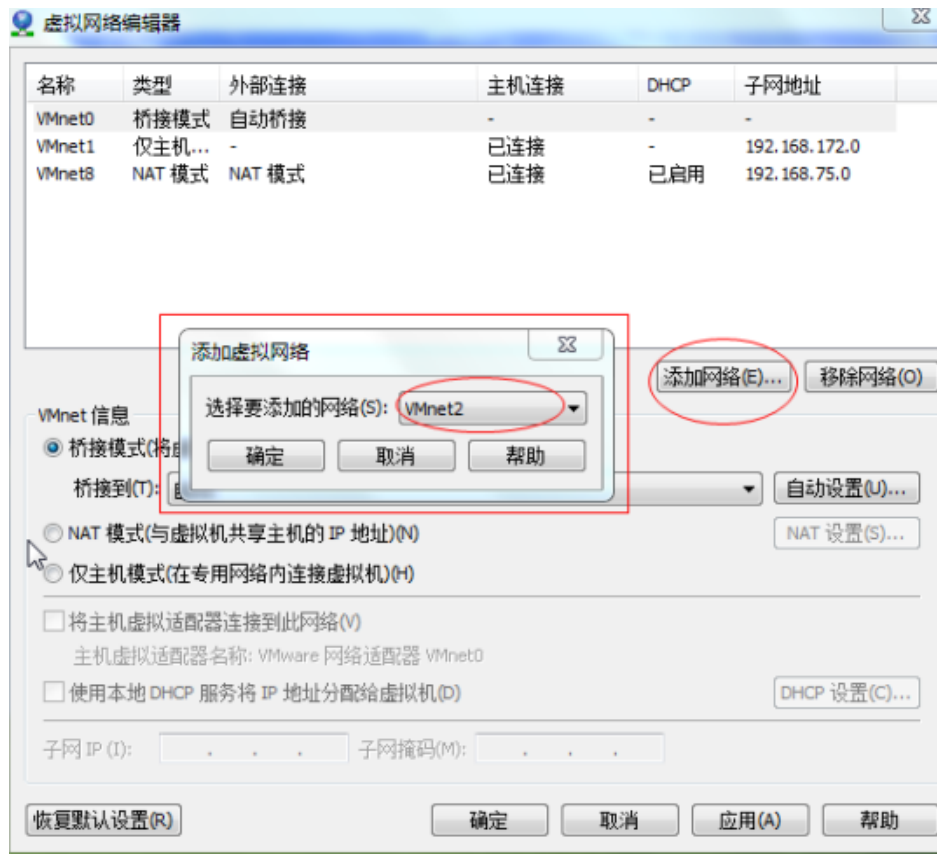
(二) 网络配置

1. 创建虚拟网卡

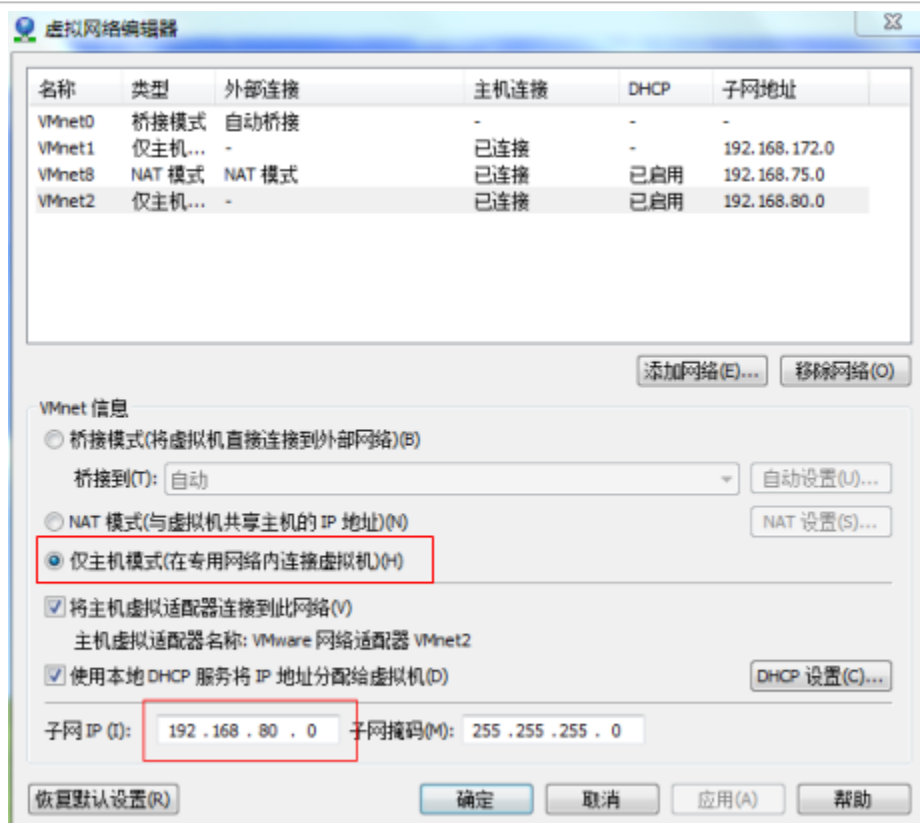
VMware 中选择菜单“编辑”--“虚拟网络编辑器”



弹出的窗口中，点击“添加网络”按钮，名称为 VMnet2 ，确定



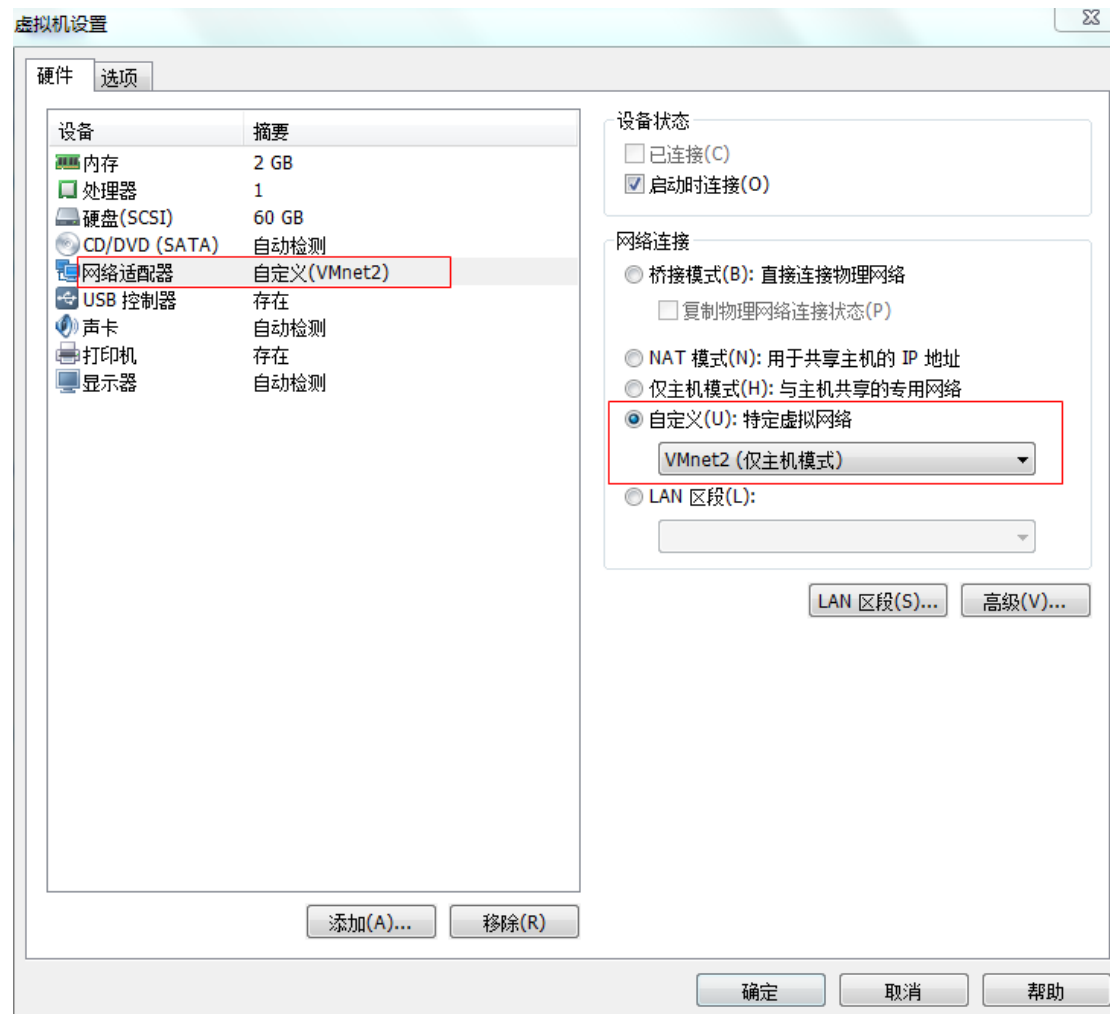
设置为仅主机方式，并设定子网 IP 为 192.168.80.0



2. 设定虚拟操作系统的网络网卡



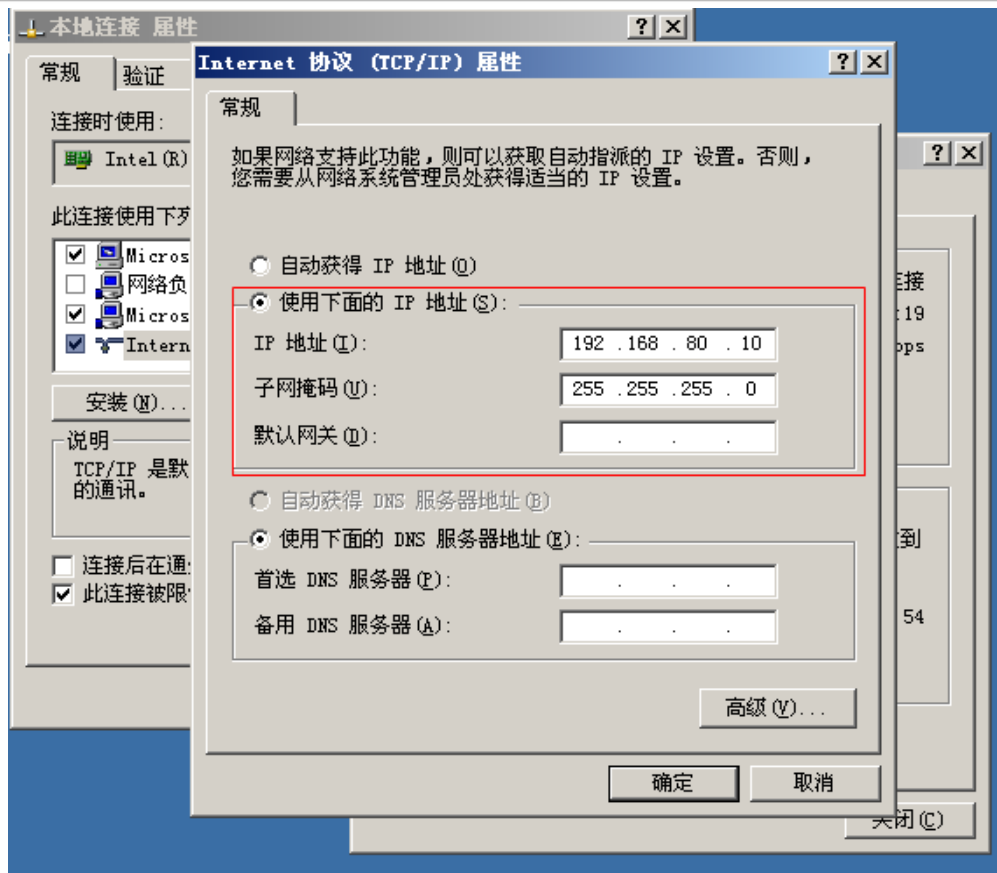
右键点击虚拟操作系统，选择“设置”菜单项，弹出以下窗口



点击网络适配器，选择自定义，VMnet2

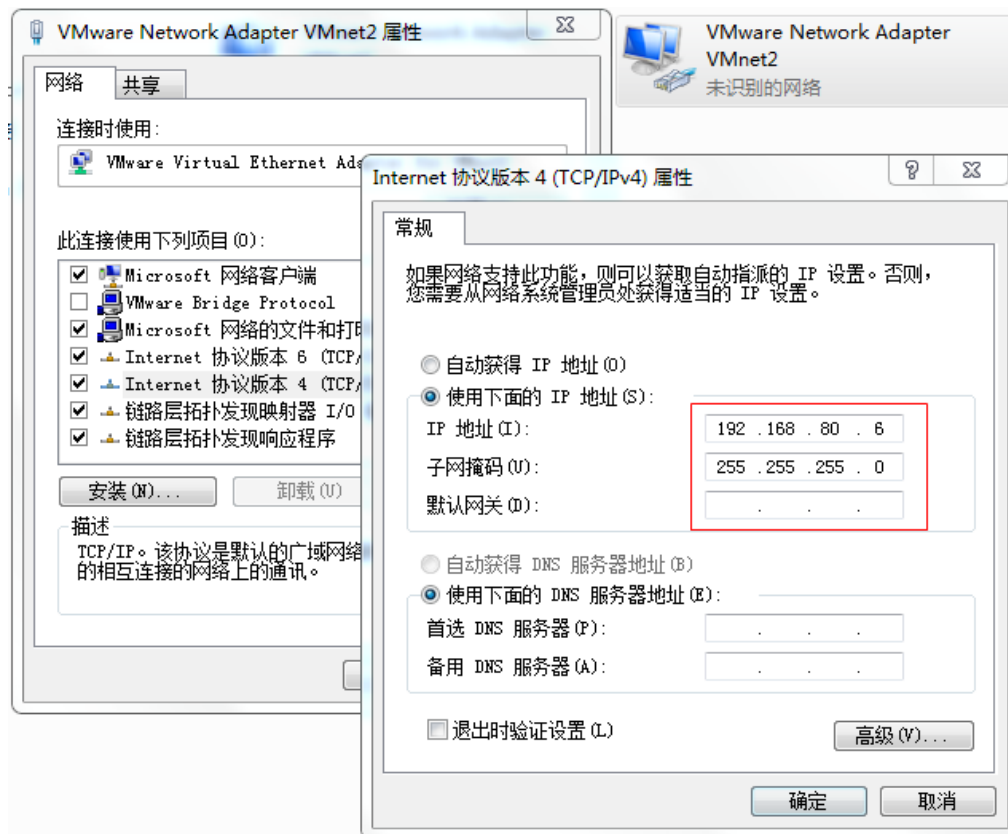
3. 设定虚拟操作系统的 IP 地址

在虚拟机的操作系统中设定 IP 地址为 192.168.80.10



设置本地的操作系统的虚拟网卡 VMnet 的 IP 为 192.168.80.6

(与虚拟机中的操作系统的 IP 地址处于同一 IP 网段)



在本地操作系统打开命令行，用 ping 命令测试网络是否连接成功

```
C:\Users\Administrator>ping 192.168.80.10

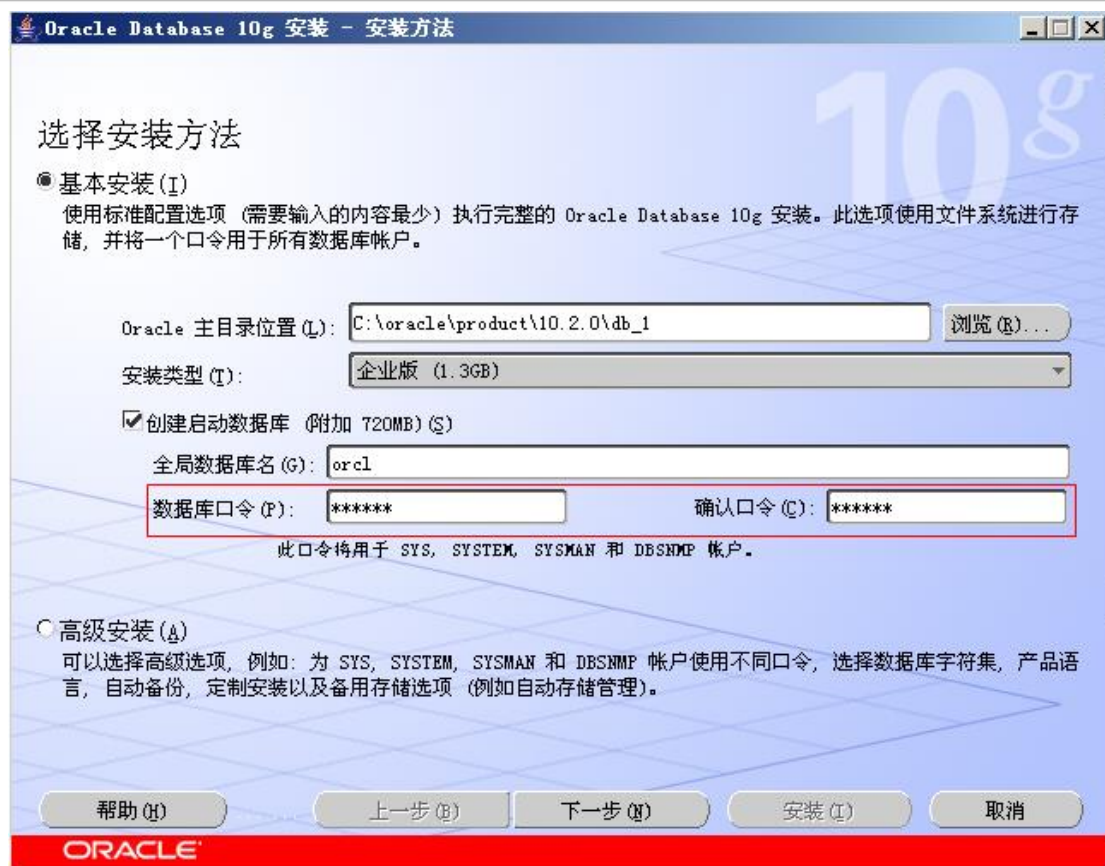
正在 Ping 192.168.80.10 具有 32 字节的数据:
来自 192.168.80.10 的回复: 字节=32 时间=1ms TTL=128
来自 192.168.80.10 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.80.10 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.80.10 的回复: 字节=32 时间<1ms TTL=128

192.168.80.10 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 0ms, 最长 = 1ms, 平均 = 0ms
```

(三) 安装 ORACLE 数据库

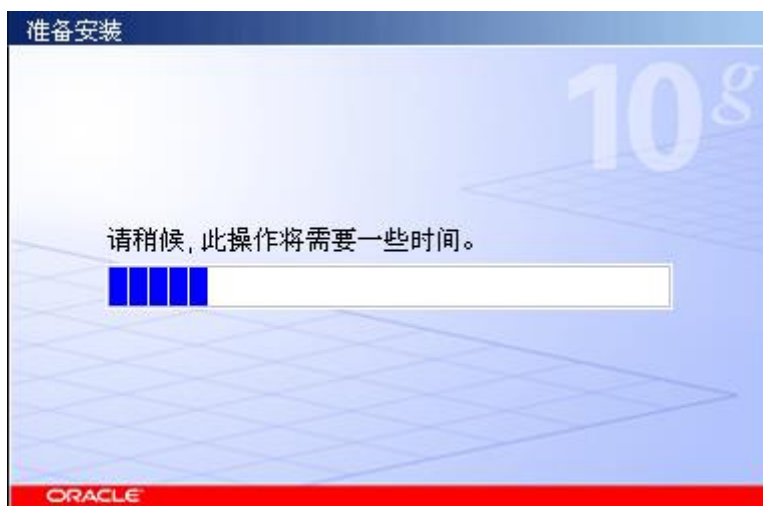
将“资源”文件夹 ORACLE 安装包解压拷贝到虚拟机的系统中

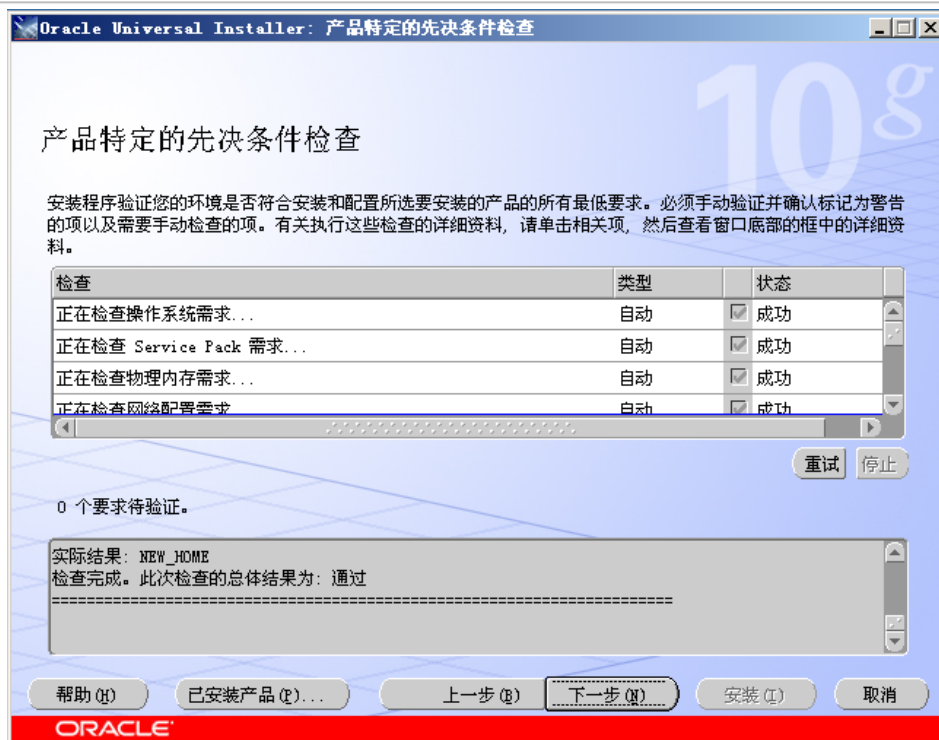
并双击解压目录下的 setup.exe，出现安装界面，如下：



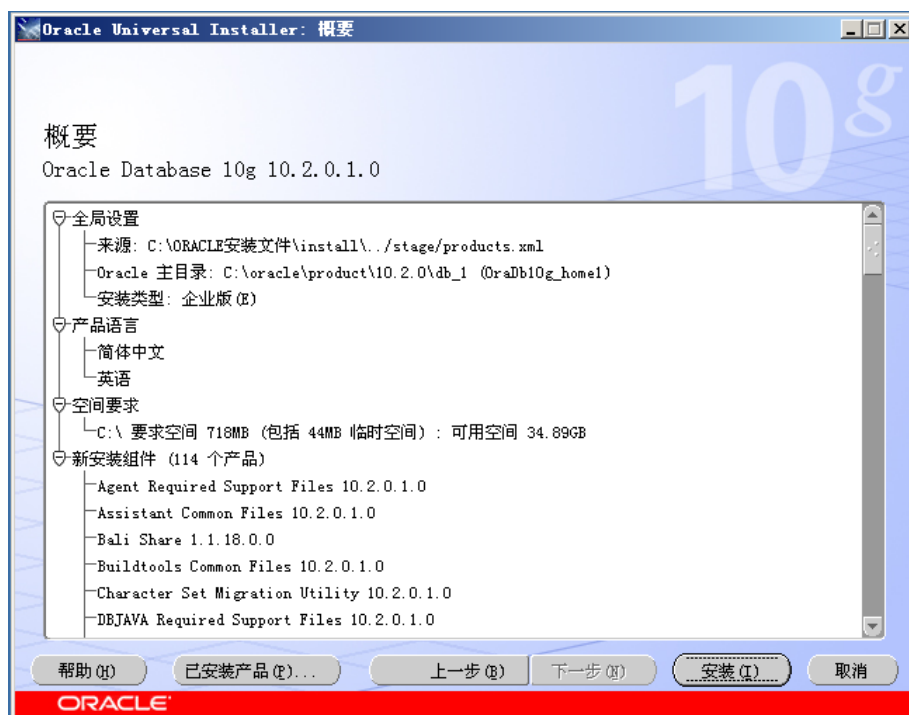
输入口令和确认口令，如：itcast，点击下一步，出现如下进度条，

注：此口令即是管理员密码。

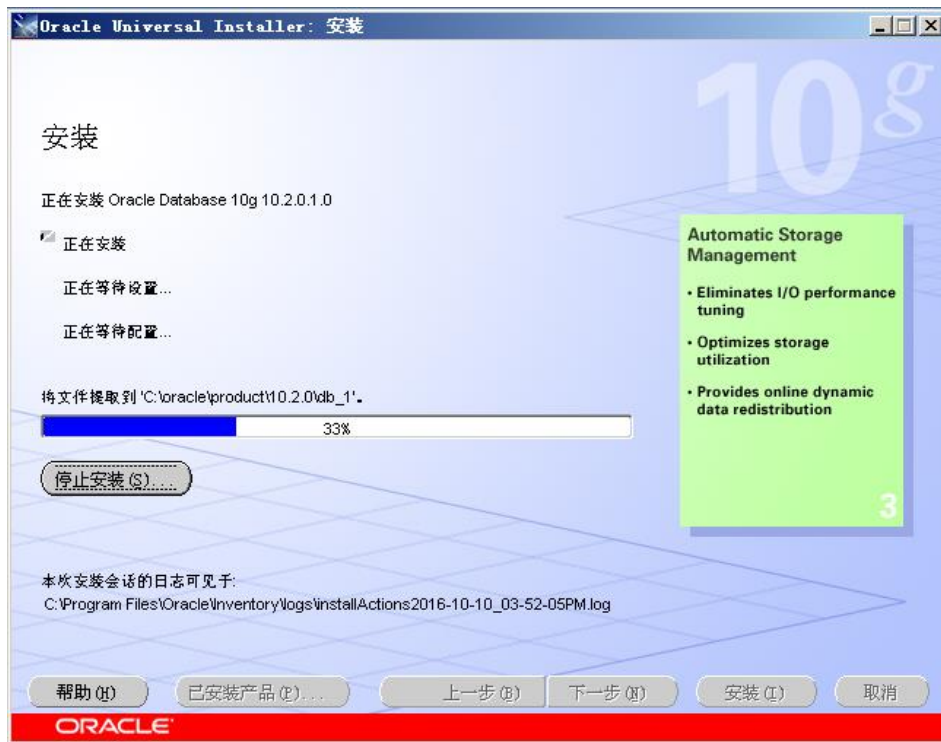




点击“下一步”，出现“概要”界面，点击“安装”。



出现安装进度条，等待安装完成，如下图：



安装完成后，自动运行配置向导，如下图，等待其完成：



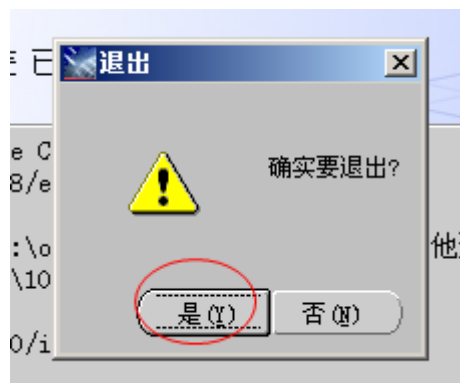
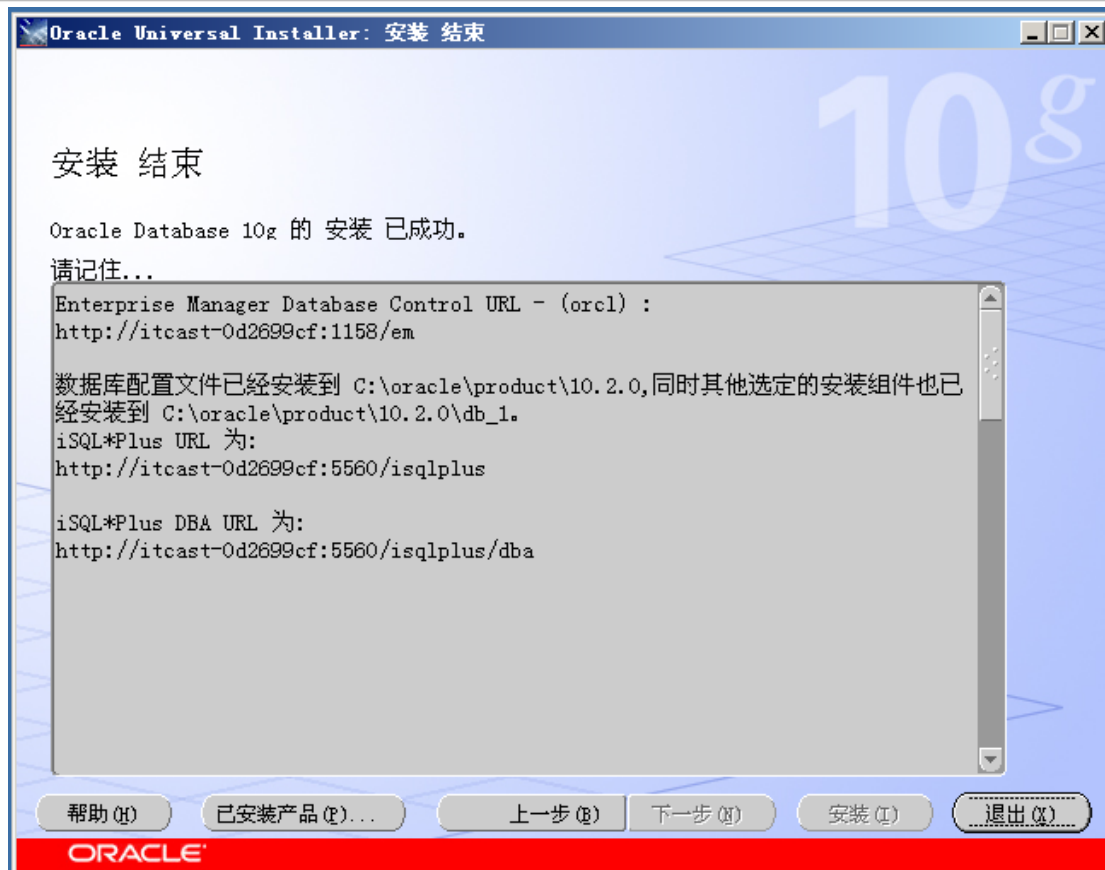


完成后，出现“口令管理”界面，如下图：

点击“确定”，如下图：



安装结束，点击“退出”。



此时可以在命令提示符下进行测试安装结果

输入:sqlplus system/itcast

itcast 为你安装时输入的密码



```
C:\> 命令提示符 - sqlplus system/itcast
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>sqlplus system/itcast

SQL*Plus: Release 10.2.0.1.0 - Production on 星期日 5月 29 10:48:28 2016

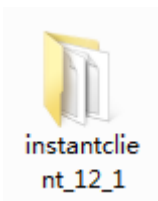
Copyright (c) 1982, 2005, Oracle. All rights reserved.

连接到:
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> _
```

(四) SQLPlus 远程连接 ORACLE 数据库

将“资源”文件夹中的 instantclient_12_1 拷贝到 D 盘根目录



进入命令提示符，进入该目录，输入如下命令连接远程的 ORACLE

```
C:\Users\Administrator>d:
D:\>cd instantclient_12_1
D:\instantclient_12_1>sqlplus system/itcast@192.168.80.10:1521/orcl

SQL*Plus: Release 12.1.0.1.0 Production on Sat Oct 8 22:10:35 2016

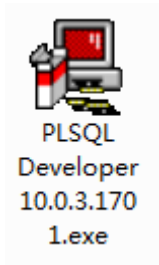
Copyright (c) 1982, 2013, Oracle. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production
With the Partitioning, OLAP and Data Mining options

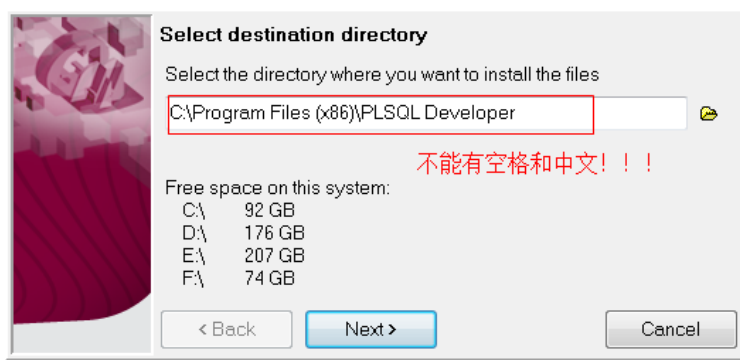
SQL>
```

(五) PLSQL Developer 安装与配置

(1) 安装 资源文件夹 PLSQL+Developer10.0.3.1701

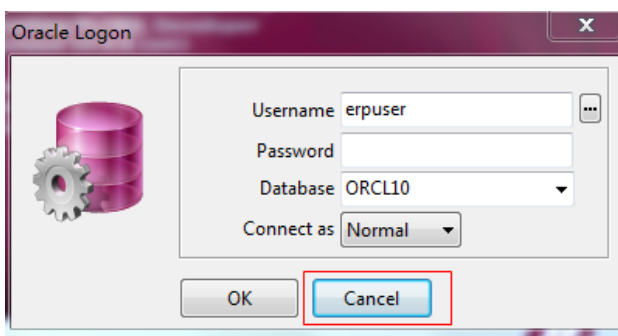


注意事项：安装目录不能有中文和空格，否则连接不上

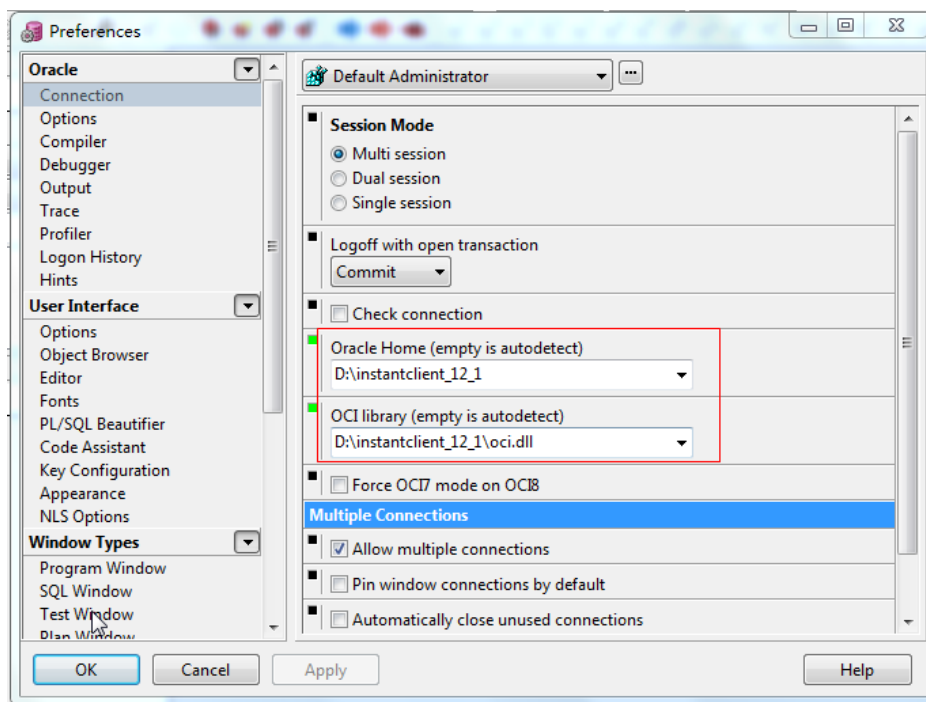


将上边的目录改为 D:\PLSQLDeveloper

(2) 配置客户端路径

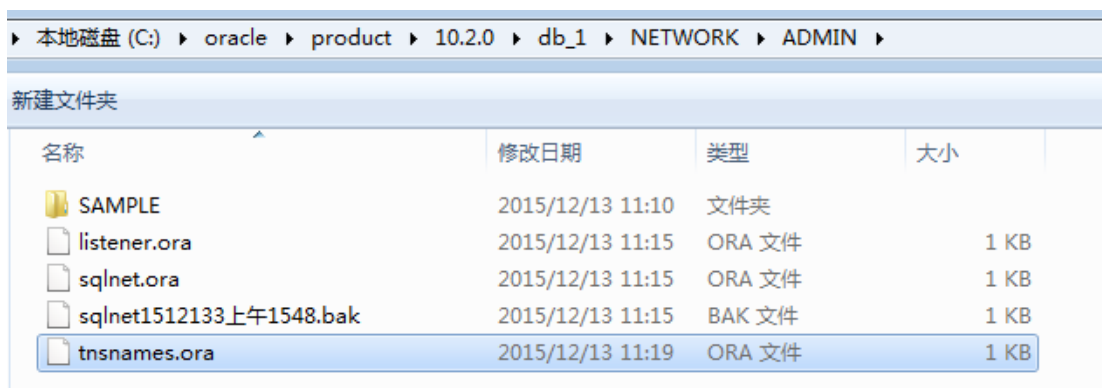


在弹出的登陆窗口中，点取消，Tools->>Preferences



(3) 编辑 ORA 文件

在 ORACLE 的以下安装目录中找到 tnsnames.ora 文件 ,拷贝到本地电脑的 D 盘根目录。



打开 tnsnames.ora 文件编辑，修改下图红色方框处

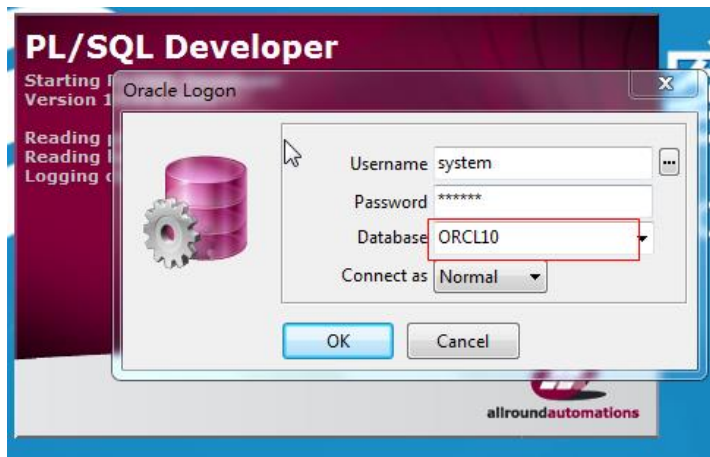
```
ORCL10 =  
(DESCRIPTION =  
  (ADDRESS = (PROTOCOL = TCP) (HOST = 192.168.80.10) (PORT = 1521))  
  (CONNECT_DATA =  
    (SERVER = DEDICATED)  
    (SERVICE_NAME = orcl)  
  )  
)
```



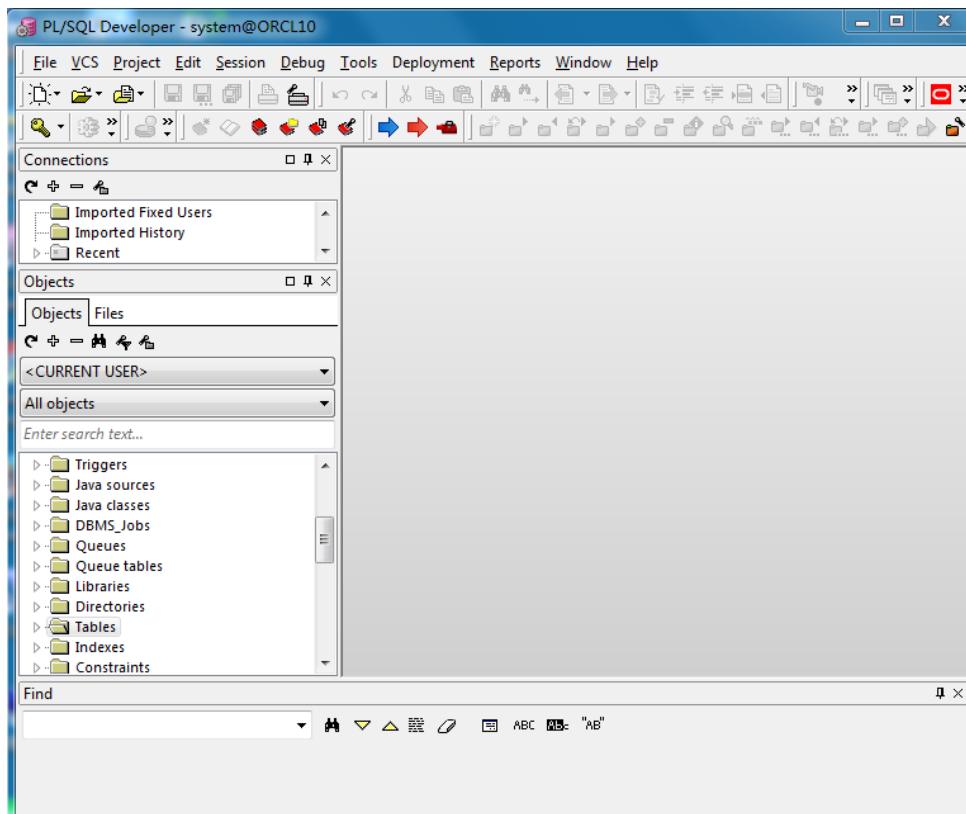
(4) 设置环境变量 TNS_ADMIN 为 D 盘根目录(tnsnames.ora 所在目录)



(5) 进入系统



输入用户名 system 密码 itcast 即可进入到软件主界面。



(6) 中文编码设置

查看服务器端编码 SQL:

```
select userenv('language') from dual
```

我实际查到的结果为:AMERICAN_AMERICA.ZHS16GBK

计算机->属性->高级系统设置->环境变量->新建

设置变量名:NLS_LANG,变量值:第 1 步查到的值, 我的是

AMERICAN_AMERICA.ZHS16GBK



三、项目案例：《自来水公司收费系统》

（一）项目介绍与需求分析

XXX 市自来水公司为更好地对自来水收费进行规范化管理，决定委托传智播客·黑马程序员开发《自来水公司收费系统》。考虑到自来水业务数量庞大，数据并发量高，决定数据库采用 ORACLE 数据库。主要功能包括：

1、基础信息管理：

（1）业主类型设置

（2）价格设置

（3）区域设置

（4）收费员设置

（5）地址设置

2、业主信息管理：

（1）业主信息维护

（2）业主信息查询

3、收费管理：

（1）抄表登记

（2）收费登记

（3）收费记录查询

（4）欠费用户清单

4、统计分析：



(1) 收费日报单

(2) 收费月报表

.....

(二) 表结构设计

1. 业主类型表 (T_OWNERTYPE)

字段名	类型(位数)	是否必填	说明
ID	NUMBER	是	主键
NAME	VARCHAR2(30)	是	类型名称

2. 价格表 (T_PRICETABLE)

字段名	类型(位数)	是否必填	说明
ID	NUMBER	是	主键
PRICE	NUMBER(10,2)	是	价格
OWNERTYPEID	NUMBER	是	业主类型 ID
MINNUM	NUMBER(10,2)	是	区间数开始值
MAXNUM	NUMBER(10,2)	是	区间数截止值

3. 区域表 (T_AREA)

字段名	类型(位数)	是否必填	说明
ID	NUMBER	是	主键
NAME	VARCHAR2(30)	是	区域名称

4. 收费员表 (T_OPERATOR)



字段名	类型(位数)	是否必填	说明
ID	NUMBER	是	主键
NAME	VARCHAR2(30)	是	操作员名称

5.地址表 (T_ADDRESS)

字段名	类型(位数)	是否必填	说明
ID	NUMBER	是	主键
NAME	VARCHAR2(30)	是	地址名称
AREAID	NUMBER	是	区域 ID
OPERATORID	NUMBER	是	操作员 ID

6.业主表 (T_OWNERS)

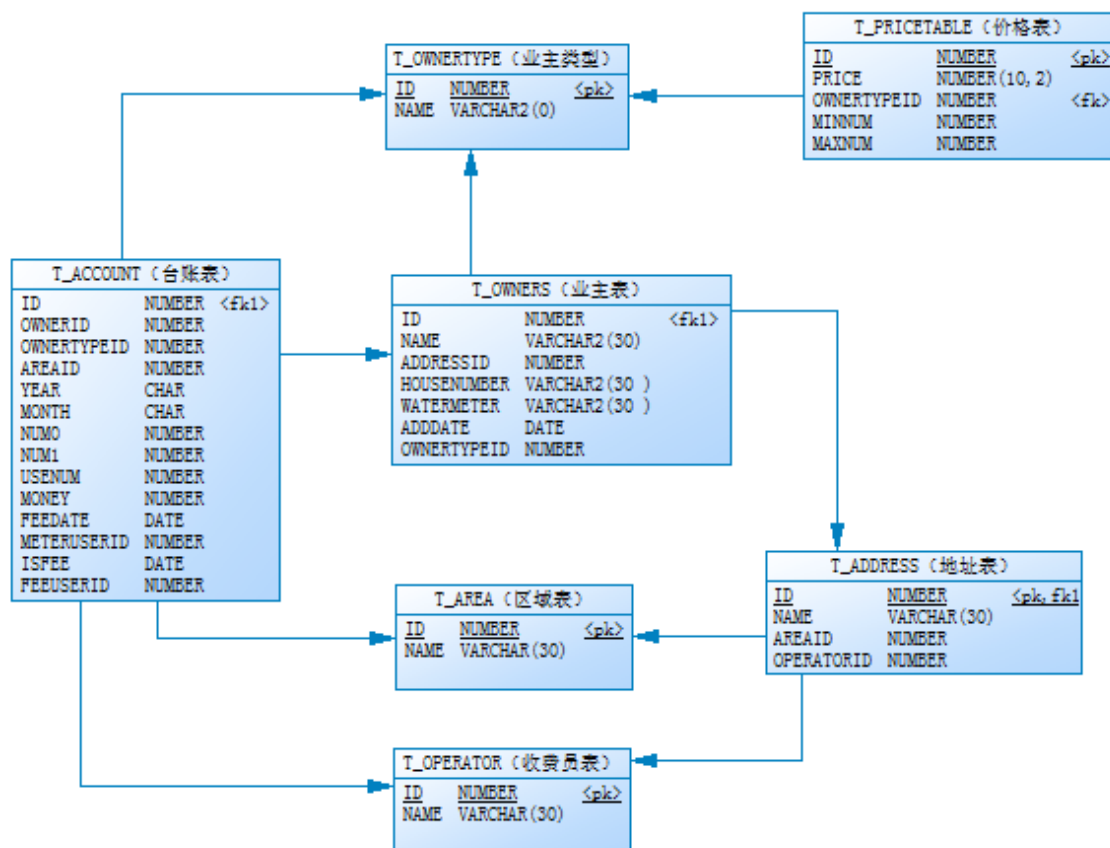
字段名	类型(位数)	是否必填	说明
ID	NUMBER	是	主键
NAME	VARCHAR2(30)	是	业主名称
ADDRESSID	NUMBER	是	地址 ID
HOUSENUMBER	NUMBER	是	门牌号
WATERMETER	VARCHAR2(30)	是	水表编号
ADDDATE	DATE	是	登记日期
OWNERTYPEID	NUMBER	是	业主类型 ID

7.收费台账 (T_ACCOUNT)



字段名	类型(位数)	是否必填	说明
ID	NUMBER	是	主键
OWNERID	NUMBER	是	业主编号
OWNERTYPEID	NUMBER	是	业主类型
AREAID	NUMBER	是	所在区域
YEAR	CHAR(4)	是	账务年份
MONTH	CHAR(2)	是	账务月份
NUM0	NUMBER		上月累计数
NUM1	NUMBER		本月累计数
USENUM	NUMBER		本月使用数
METERUSERID	NUMBER		抄表员
METERDATE	DATE		抄表日期
MONEY	NUMBER(10,2)		应缴金额
ISFEE	CHAR(1)	是	是否缴费
FEEDATE	DATE		缴费日期
FEEUSERID	NUMBER		收费员

上述 7 张表的物理模型如下：



(三) 创建表空间

```

create tablespace waterboss
datafile 'c:\waterboss.dbf'
size 100m
autoextend on
next 10m
    
```

解释：

waterboss 为表空间名称

datafile 用于设置物理文件名称

size 用于设置表空间的初始大小

autoextend on 用于设置自动增长，如果存储量超过初始大小，则开始自动扩容

next 用于设置扩容的空间大小



（四）创建用户

```
create user wateruser  
identified by itcast  
default tablespace waterboss
```

wateruser 为创建的用户名

identified by 用于设置用户的密码

default tablespace 用于指定默认表空间名称

（五）用户赋权

```
grant dba to wateruser
```

给用户 wateruser 赋予 DBA 权限后即可登陆

四、表的创建、修改与删除

（一）创建表

语法：

```
CREATE TABLE 表名称(  
  
    字段名 类型(长度) primary key,  
  
    字段名 类型(长度),  
  
    .....  
);
```

数据类型：

1. 字符型

（1）CHAR：固定长度的字符类型，最多存储 2000 个字节

（2）VARCHAR2：可变长度的字符类型，最多存储 4000 个字节



(3) LONG：大文本类型。最大可以存储 2 个 G

2.数值型

NUMBER：数值类型

例如：NUMBER(5) 最大可以存的数为 99999

NUMBER(5,2) 最大可以存的数为 999.99

3.日期型

(1) DATE：日期时间型，精确到秒

(2) TIMESTAMP：精确到秒的小数点后 9 位

4.二进制型（大数据类型）

(1) CLOB：存储字符,最大可以存 4 个 G

(2) BLOB：存储图像、声音、视频等二进制数据,最多可以存 4 个 G

实例：

创建业主表

```
create table t_owners
(
  id number primary key,
  name varchar2(30),
  addressid number,
  housenumber varchar2(30),
  watermeter varchar2(30),
  adddate date,
  ownertypeid number
);
```

其它表的创建见资料“自来水收费系统建表语句.txt”



(二) 修改表

1. 增加字段语法:

```
ALTER TABLE 表名称 ADD(列名 1 类型 [DEFAULT 默认值], 列名 1 类型  
[DEFAULT 默认值]...)
```

为业主表增加两个字段，语句：

```
--追加字段  
ALTER TABLE T_OWNERS ADD  
(  
    REMARK VARCHAR2(20),  
    OUTDATE DATE  
)
```

2. 修改字段语法：

```
ALTER TABLE 表名称 MODIFY(列名 1 类型 [DEFAULT 默认值], 列名 1 类型  
[DEFAULT 默认值]...)
```

修改两个字段的类型，语句：

```
--修改字段  
ALTER TABLE T_OWNERS MODIFY  
(  
    REMARK CHAR(20),  
    OUTDATE TIMESTAMP  
)
```

3. 修改字段名语法：

```
ALTER TABLE 表名称 RENAME COLUMN 原列名 TO 新列名
```



语句：

```
ALTER TABLE T_OWNERS RENAME COLUMN OUTDATE TO EXITDATE
```

4. 删除字段名

--删除一个字段

```
ALTER TABLE 表名称 DROP COLUMN 列名
```

--删除多个字段

```
ALTER TABLE 表名称 DROP (列名 1,列名 2...)
```

语句：

--删除字段

```
ALTER TABLE T_OWNERS DROP COLUMN REMARK
```

(三) 删除表

语法：

```
DROP TABLE 表名称
```

五、数据增删改

(一) 插入数据

语法：

```
INSERT INTO 表名[(列名 1 , 列名 2 , ...)]VALUES(值 1 , 值 2 , ...)
```



执行 INSERT 后一定要再执行 commit 提交事务

向业主表插入数据：

```
insert into T_OWNERS VALUES (1, '张三丰', 1, '2-2', '5678', sysdate, 1);
```

语句中的 sysdate 是系统变量用于获取当前日期，点击齿轮的图标后，再点击下图的绿色图标，此图标为 commit



我们再次录入一条数据，语句如下：

```
insert into T_OWNERS VALUES (2, '赵大侃', 1, '2-3', '9876', sysdate, 1);  
commit;
```

（二）修改数据

语法：

```
UPDATE 表名 SET 列名 1=值 1, 列名 2=值 2, ....WHERE 修改条件;
```

执行 UPDATE 后一定要再执行 commit 提交事务

需求：将 ID 为 1 的业主的登记日期更改为三天前的日期

```
update T_OWNERS set adddate=adddate-3 where id=1;  
commit;
```

（三）删除数据

语法 1：

DELETE FROM 表名 WHERE 删除条件;

执行 DELETE 后一定要再执行 commit 提交事务

需求：删除业主 ID 为 2 的业主信息

```
delete from T_OWNERS where id=2;  
commit;
```

语法 2：

TRUNCATE TABLE 表名称

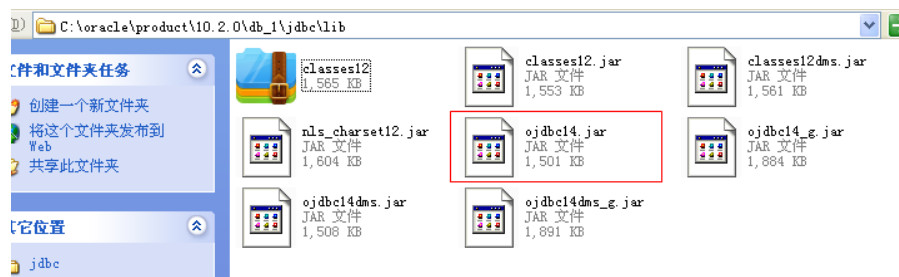
比较 truncat 与 delete 实现数据删除？

1. delete 删除的数据可以 rollback
2. delete 删除可能产生碎片，并且不释放空间
3. truncate 是先摧毁表结构，再重构表结构

六、JDBC 连接 ORACLE

(一) 创建工程，引入驱动包

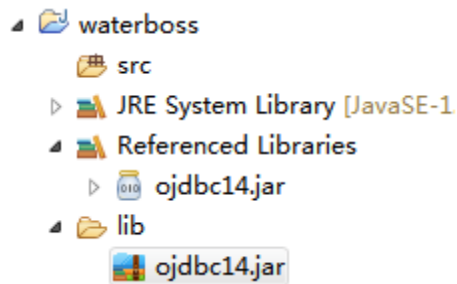
在下图目录中可以找到驱动包 ORACLE 的 JDBC 驱动包，拷贝到工程即可使用



创建 java 工程 waterboss，建立 lib 文件夹，将 ojdbc.jar 拷贝到此文件夹，然后 add



build path



(二) BaseDao

我们通常编写 BaseDao 负责加载驱动，获取数据库连接，关闭资源，代码如下：

```
package cn.itcast.waterboss.dao;
import java.sql.SQLException;
/**
 * 基本数据访问类
 * @author Administrator
 *
 */
public class BaseDao {

    //加载驱动
    static{
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }

    /**
     * 获取数据库连接
     * @return
     * @throws SQLException
     */
    public static java.sql.Connection getConnection() throws
    SQLException{

        return java.sql.DriverManager.getConnection(
```



```
        "jdbc:oracle:thin:@192.168.80.10:1521:orcl",
        "wateruser", "itcast");

    }

    /**
     * 关闭资源
     * @param rs
     * @param stmt
     * @param conn
     */
    public static void closeAll(java.sql.ResultSet rs,
    java.sql.Statement stmt, java.sql.Connection conn)
    {
        //关闭结果集
        if(rs!=null){
            try {
                rs.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        //关闭执行对象
        if(stmt!=null){
            try {
                stmt.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        //关闭执行对象
        if(conn!=null){
            try {
                conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}
```



JDBC 驱动为：

oracle.jdbc.OracleDriver

连接字符串(瘦连接)：

jdbc:oracle:thin:@虚拟机的 IP:1521:orcl

(三) 业主增删改代码编写

1.创建实体类

```
package cn.itcast.waterboss.entity;
import java.util.Date;

/**
 * 业主实体类
 * @author Administrator
 *
 */
public class Owners {

    private Long id;//编号
    private String name;//业主名称
    private Long addressid;//地址编号
    private String housenumber;//门牌号
    private String watermeter;//水表编号
    private Date adddate;//登记日期
    private Long ownertypeid;//业主类型 ID

    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
```



```
        this.name = name;
    }
    public Long getAddressid() {
        return addressid;
    }
    public void setAddressid(Long addressid) {
        this.addressid = addressid;
    }
    public String getHousenumber() {
        return housenumber;
    }
    public void setHousenumber(String housenumber) {
        this.housenumber = housenumber;
    }
    public String getWatermeter() {
        return watermeter;
    }
    public void setWatermeter(String watermeter) {
        this.watermeter = watermeter;
    }
    public Date getAdddate() {
        return adddate;
    }
    public void setAdddate(Date adddate) {
        this.adddate = adddate;
    }
    public Long getOwnertypeid() {
        return ownertypeid;
    }
    public void setOwnertypeid(Long ownertypeid) {
        this.ownertypeid = ownertypeid;
    }
}
```

2.创建 Dao 类实现增删改

```
package cn.itcast.waterboss.dao;

import java.sql.SQLException;
```



```
import cn.itcast.waterboss.entity.Owners;

/**
 * 业主数据访问类
 * @author Administrator
 *
 */
public class OwnersDao {

    /**
     * 新增业主
     * @param owners
     */
    public static void add(Owners owners){

        java.sql.Connection conn=null;
        java.sql.PreparedStatement stmt=null;

        try {
            conn=BaseDao.getConnection();
            stmt=conn.prepareStatement("insert into T_OWNERS
values(?,?,?,?,?,?,?,?)");
            stmt.setLong(1, owners.getId());
            stmt.setString(2, owners.getName());
            stmt.setLong(3, owners.getAddressid());
            stmt.setString(4, owners.getHousenumber());
            stmt.setString(5, owners.getWatermeter());
            stmt.setDate(6, new
java.sql.Date(owners.getAdddate().getTime()));
            stmt.setLong(7, owners.getOwnertypeid());
            stmt.execute();
        } catch (SQLException e) {

            e.printStackTrace();
        }finally {
            BaseDao.closeAll(null, stmt, conn);
        }
    }

    /**
     * 修改业主
     * @param owners
     */
}
```



```
*/  
public static void update(Owners owners){  
  
    java.sql.Connection conn=null;  
    java.sql.PreparedStatement stmt=null;  
  
    try {  
        conn=BaseDao.getConnection();  
        stmt=conn.prepareStatement("update T_OWNERS set  
name=?,addressid=?,houenumber=?,"  
        + "watermeter=?,adddate=?, ownertypeid=? where  
id=?");  
  
        stmt.setString(1, owners.getName());  
        stmt.setLong(2, owners.getAddressid());  
        stmt.setString(3, owners.getHousenumber());  
        stmt.setString(4, owners.getWatermeter());  
        stmt.setDate(5, new  
java.sql.Date(owners.getAdddate().getTime()));  
        stmt.setLong(6, owners.getOwnertypeid());  
        stmt.setLong(7, owners.getId());  
        stmt.execute();  
    } catch (SQLException e) {  
  
        e.printStackTrace();  
    }finally {  
        BaseDao.closeAll(null, stmt, conn);  
    }  
}  
  
/**  
 * 删除业主  
 * @param owners  
 */  
public static void delete(Long id){  
  
    java.sql.Connection conn=null;  
    java.sql.PreparedStatement stmt=null;  
  
    try {  
        conn=BaseDao.getConnection();
```



```
        stmt=conn.prepareStatement("delete from T_OWNERS where  
id=?");  
        stmt.setLong(1, id);  
        stmt.execute();  
    } catch (SQLException e) {  
  
        e.printStackTrace();  
    }finally {  
        BaseDao.closeAll(null, stmt, conn);  
    }  
}
```

七、数据导出与导入

当我们使用一个数据库时，总希望数据库的内容是可靠的、正确的，但由于计算机系统的故障（硬件故障、软件故障、网络故障、进程故障和系统故障）影响数据库系统的操作，影响数据库中数据的正确性，甚至破坏数据库，使数据库中全部或部分数据丢失。因此当发生上述故障后，希望能重构这个完整的数据库该处理称为数据库恢复，而要进行数据库的恢复必须要有数据库的备份工作。

（一）整库导出与导入

整库导出命令

```
exp system/itcast full=y
```

添加参数 full=y 就是整库导出



```
连接到: Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production
With the Partitioning, OLAP and Data Mining options
已导出 ZHS16GBK 字符集和 AL16UTF16 NCHAR 字符集

即将导出整个数据库...
. 正在导出表空间定义
. 正在导出概要文件
. 正在导出用户定义
. 正在导出角色
. 正在导出资源成本
. 正在导出回退段定义
. 正在导出数据库链接
. 正在导出序号
. 正在导出目录别名
. 正在导出上下文名称空间
. 正在导出外部函数库名
. 导出 PUBLIC 类型同义词
. 正在导出专用类型同义词
. 正在导出对象类型定义
. 正在导出系统过程对象和操作
```

执行命令后会在当前目录下生成一个叫 EXPDAT.DMP，此文件为备份文件。

如果想指定备份文件的名称，则添加 file 参数即可，命令如下

```
exp system/itcast file=文件名 full=y
```

整库导入命令

```
imp system/itcast full=y
```

此命令如果不指定 file 参数，则默认用备份文件 EXPDAT.DMP 进行导入

如果指定 file 参数，则按照 file 指定的备份文件进行恢复

```
imp system/itcast full=y file=water.dmp
```

（二）按用户导出与导入

按用户导出

```
exp system/itcast owner=wateruser file=wateruser.dmp
```

按用户导入

```
imp system/itcast file=wateruser.dmp fromuser=wateruser
```




(三) 按表导出与导入

按表导出

```
exp wateruser/itcast file=a.dmp tables=t_account,a_area
```

用 tables 参数指定需要导出的表，如果有多个表用逗号分割即可

按表导入

```
imp wateruser/itcast file=a.dmp tables=t_account,a_area
```

八、总结

(一) 知识点总结

(二) 上机任务布置

1. 在虚拟机软件中挂载 win7 (模拟作为服务器使用) , 并配置网络 , 网络要 ping 通
2. 在服务器中安装 ORACLE.
3. 在本机安装 ORACLE 客户端软件 PL/SQL Developer , 并配置
4. 创建表空间、用户、赋权
5. DDL 语句练习 (创建表、修改表、删除表语句)
6. DML 语句练习 (insert 、 update 、 delete)
7. 编写 JDBC 连接 ORACLE 的代码
8. 数据导入导出练习