**Oracle的dbms和实例的安装**

下载11g的 安装包，解压后运行 setup.exe。

选择安装目录并设置实例名称和默认密码(SYS,SYSTEM)。

安装后，默认会将oracle的安装目录加入到环境变量path中。

可以运行sqlplus，输入用户名密码登录，验证数据库安装ok。

**安装后各个服务说明**

每个数据库实例都对应一套服务

OracleServiceXLORACLEDB 数据库服务(数据库实例)，Oracle核心服务，每个数据库实例对应1个服务

OracleOraDb11g\_home1TNSListener 监听器服务，支持数据库远程访问，每个监听端口对应一个服务

OracleOraDb11g\_home1ClrAgent Oracle数据库.NET扩展服务的一部分。 （可不启动）

OracleMTSRecoveryService 使数据库成为mts服务（可不启动）

OracleJobSchedulerXLORACLEDB Oracle作业调度（定时器）服务（可不启动）

OracleDBConsolexloracledb web管理端Enterprise Manager 需要的服务（可不启动）

OracleVssWriterXLORACLEDB Oracle卷映射拷贝写入服务（可不启动）

**Net Configuration Assitant 的使用**

监听程序配置

用于配置监听程序端口，使远程可以访问，配置的结果在listener.ora 文件中

此配置用在服务端，每配置1个端口，服务端会多一个监听服务

命名方法配置

服务命名的方法的搜索顺序，配置的结果在sqlnet.ora 文件中

默认为先搜索本地命名，再搜索轻松连接命名，不用修改

本地命名配置

用于连接服务器的配置，配置的结果在tnsnames.ora中

用在客户端，通过此配置来找到服务器

**NetManager**

同样可以配置 监听，命名方法和本地配置，和Net Configuration Assitant 功能一样

直接配置而不是向导配置，适合高级用户

**数据库实例的管理**

使用 Database Configuration Assistant，可以创建，配置数据库实例(oracle中的数据库实例 相当于sqlserver 中的数据库)

可以配置数据库示例对应的监听程序，这样在一台机器上可以有多个数据库实例

此时新建的实例会生成对应的windows服务

数据库实例对应的文件 在 oracle 安装目录下的oradata目录下，每个实例都会有单独的文件夹保存文件

其中包含 数据文件dbf， 控制文件ctl，日志文件log

**数据库远程登录配置**

使用 Net Configuration Assitant 配置 listener ，以便远程可访问

开listener配置的端口的防火墙(默认1521)

在远程服务器上 使用 sqlplus登录， 使用 sqlplus 用户名/密码@ip:端口/实例名

使用 pl/sql develop 远程登录：

下载 pl/sql develop (64位） 下载 intall client （64位）

解压intall client 到任意目录，在此目录下新建 network\admin目录

在此目录下新建 tnsnames.ora,并进行配置远程数据库实例

添加环境变量TNS\_ADMIN 指到 上一步的 network\admin目录

添加环境变量NLS\_LANG 到 SIMPLIFIED CHINESE\_CHINA.ZHS16GBK

配置 pl/sql develop ， preferences->connection 中

Oracle Home 指向intall client 解压的目录

OCI library 指向intall client 解压的目录下的 oci.dll

重启 pl/sql develop，在database中找到 tnsnames.ora 配置的连接实例

**常用命令**

显示当前登录的用户 show user

指定查询时1行显示的最长字符数 set linesize 100

显示当前1行显示的最长字符 show linesize

指定1页显示的行数 set pagesize 30

显示1页显示的行数 show pagesize

每页是否逐渐显示 set pause on/off

显示查询的前10行 select \* from emp where rownum < 10

解锁用户 alter user 用户名 account unlock;

设置用户名密码 password 用户名

给用户权限 grant 权限/角色 to 用户名

常用实例：grant connect to scott; grant resource to scott;

用户登录 connect 用户名/密码

断开连接 disconnect

收回权限 revoke权限名或者角色名from用户名

备份某个用户下的所有信息（需要在服务端运行） EXP SYSTEM/Xl711436 FILE=C:\FULL.DMP

还原 IMP SYSTEM/Xl711436 FILE=C:\FULL.DMP

备份某些表 EXP SYSTEM/Xl711436 tables=(aa) FILE=C:\aa.DMP

还原某些表 IMP SYSTEM/Xl711436 tables=(aa) FILE=C:\aa.DMP

表空间操作

建立表空间 create tablespace testspace datafile 'C:\oraclefolder\oradata\xloracledb\testspace.dbf' size 20M autoextend on maxsize 50M

可以设置 表空间名，保存的文件名 ，初始大小，是否自增长，并可设置增长的最大尺寸

查看已建立的表空间 select \* from dba\_data\_files select \* from dba\_tablespaces

表空间支持多文件 create tablespace testspace1 datafile 'C:\oraclefolder\oradata\xloracledb\testspace1.dbf'

size 10M,'C:\oraclefolder\oradata\xloracledb\testspace2.dbf' size 20M autoextend on

修改用户的默认表空间 每个用户都有默认表空间，当用户创建表且没有指定表空间时，则此表属于默认表空间 USERS

通过 select \* from dba\_users 可以查看到用户的默认表空间

通过alter database default tablespace 空间名 来修改默认表空间

表空间的重命名 alter tablespace user2 rename to user20 重命名并不会对数据造成影响，系统表空间无法重命名（system sysaux）

删除表空间 drop tablespace user20 只删除表空间的内容

Drop tablespace user20 including contents and datafiles 删除表空间的内容和数据文件

**表操作**

建立表 create table only\_test (id number,name varchar2(20))

当一般用户建立表时，表名会带上和用户名一样的schema信息

例如,scott用户使用以上语句建立表时， 表的名称是 scott.only\_test 默认在users表空间

当 system用户使用以上语句建立表时， 表的名称是 only\_test 默认在system表空间

描述表 describe only\_test （在命令窗口）

修改表结构 alter table only\_test add(remark varchar2(100)) 添加列

alter table only\_test rename column remark to remark2 重命名列

alter table only\_test modify(remark varchar2(15)) 修改列字段

alter table only\_test drop column remark 删除列

删除表 drop table only\_test

临时表 分会话型 和事务型 两种

会话型 临时表 当会话结束时，数据会被清空,但表还存在

事务型 临时表 当事务结束时，数据会被清空,但表还存在

临时表只有手工drop 才会被删除，不会自动删除

建立会话级临时表 create global temporary table session\_temp (id number,name varchar2(20)) on commit preserve rows

建立事务级临时表 create global temporary table trans\_temp (id number,name varchar2(20)) on commit delete rows

数据操作而言，临时表和一般表无区别

一般应用场景 大表分割 解决并行问题 作为数据缓存

dual表 是个虚表，用来作为from的源表，用于满足 sql的语法,不可以删除此表

获取当前日期 select sysdate from dual

进行数据计算 select 3\*4 as result from dual

**约束**

目的 用于保持数据的完整性

查看约束信息 select \* from user\_constraints where table\_name = ‘表名’ ,constraint\_type 字段为 键的类型

获取约束作用的列 select \* from user\_cons\_columns where constraint\_name = ‘约束名’

主键约束 主键可以有一列或多列，可以是任意类型，主键作为一个整体必须唯一

建立只有一列的主键 create table only\_test1 (id number primary key ,name varchar2(20))

显示命名的建立一个主键 create table only\_test2 (id number constraint pk\_testname primary key ,name varchar2(20))

在多列上建立主键 create table only\_test3 (id number ,name varchar2(20)，constraint pk\_testname primary key(id,name))

修改主键约束 alter table only\_test1 modify (id number primary key)

添加主键约束 alter table only\_test2 add constraint pk\_test3 primary key (id,name)

删除主键约束 alter table only\_test3 drop primary key

禁用主键约束 alter table only\_test2 disable primary key

启用主键约束 alter table only\_test2 enable primary key

重命名主键约束 alter table only\_test2 rename constraint 旧名称to 新名称

外键约束 描述表间的父子关系，

建立外键 alter table orders add constraint fk\_testname foreign key (custom\_id) references customers(customer\_id)

建立本表的custom\_id 字段 和 customers 表的customer\_id 的外键关联

可以添加级联更新的选项 on delete cascade

可以添加级联删除的选项 deferrable initially deferred

唯一性约束 表示 一列的值 唯一，create table only\_test3 (id number ,name varchar2(20) unique)

检查约束 表示一类的值，需要满足的条件 create table only\_test4 (id number ,name varchar2(20) check(id<20))

默认值 insert 时可以不指定值，自动填充默认值 create table only\_test4 (id number ,name varchar2(20) default ‘aab’)

视图

函数和存储过程

游标

触发器

**序列**

目的 类似sqlserver 中的自增字段

创建序列 create sequence test\_seq

设置起始值 create sequence test\_seq start with 20

删除序列 drop sequence test\_seq

查询序列 select \* from user\_objects where lower(object\_name) = ’ test\_seq’

重要属性 currval 当前值 select test\_seq.currval from dual

nextval 下一个值 select test\_seq.nextval from dual

使用 create table only\_test5 (id number ,name varchar2(20))

Insert into only\_test5 (id,name) values (test\_seq.nextval,’a’)

id字段会自动增加

设置属性 设置最小值 alter sequence test\_seq minvalue 30

设置最大值 alter sequence test\_seq maxvalue 50

设置步长 alter sequence test\_seq increment by 5

**用户权限和角色**

查看已有的用户 select \* from dba\_users

建立用户 create user 用户名 identified by 密码 default tablespace users temporary tablespace temp quota 3m on users

schema 两个普通用户，无法互相看到对方的建立的表，哪怕是在同一个表空间中，因为建立的表分属不同的schema

权限 获取系统权限 select distinct privilege from dba\_sys\_privs

分配权限 grant 权限名 to 用户名

收回权限 revoke 权限名 from 用户名

示例： grant all on scott.emp to xiaohong 这样 用户xiaohong 可以操作用户scott建立的emp表了

角色 角色是权限的集合，方便分配

建立角色 create role role\_test

给角色添加权限 grant 权限 to 角色

继承角色权限 grant 基角色名 to 角色

几个默认角色 connect resource dba

CONNECT角色： --是授予最终用户的典型权利，最基本的

ALTER SESSION --修改会话

CREATE CLUSTER --建立聚簇

CREATE DATABASE LINK --建立数据库链接

CREATE SEQUENCE --建立序列

CREATE SESSION --建立会话

CREATE SYNONYM --建立同义词

CREATE VIEW --建立视图

RESOURCE 角色： --是授予开发人员的

CREATE CLUSTER --建立聚簇

CREATE PROCEDURE --建立过程

CREATE SEQUENCE --建立序列

CREATE TABLE --建表

CREATE TRIGGER --建立触发器

CREATE TYPE --建立类型

查看当前用户的角色 select \* from session\_roles

**基本数据类型**

数值型 number

日期型 date

大对象 clob 大型文本

blob 大型二进制

bfile 用单独文件存储二进制

特殊类型 rowid 伪列，系统生成

字符串 varchar2(10) 其中10为长度 ，varchar2 的最大长度是 32767

**内置函数**

字符串函数

lpad() 向左补全字符串 lpad(‘21’,6,’0’) 输出 000021

rpad() 向右补全

lower() 返回小写字符串

upper() 返回大写字符串

initcap() 首字母大写

length() 字符串长度

substr() 截取字符串 substr(‘123456789’,5,4) 返回 5678

instr() 查找字符串出现的位置 instr(‘abc’,’bc’) 返回2

ltrim() 去掉左侧空格

rtrim() 去掉右侧空格

to\_char() 将其他类型装换为字符串 to\_char(120,’999’) 返回字符串 120

chr() 将ascii 码转为字符串

translate() 字符串替换 translate(‘123’,’2312323123’,’abc’) 返回 ‘23abc23abc’

字符串拼接 使用 ||

like通配符 使用 %

数学函数

abs() 返回绝对值

round() 返回某个数值的四舍五入值

ceil() 向上取整

floor() 向下取整

mod() 取模操作

sign() 返回数的正负

sqrt() 返回平方根

power() 乘方运算

trunc() 截取数字

vsize() 返回数据占用的空间

to\_number() 将字符串转换为数值

日期函数

to\_date() 将字符串转换为日期

add\_months() 为日期加上月份

last\_day()) 返回日期所在月的最后1天

months\_between() 返回两个日期相差的月份

current\_date() 返回当前日期

current\_timestamp() 返回当前会话时区的时间戳

extract() 返回日期的某个区域

聚合函数

max() 取最大值

min() 取最小值

avg() 取平均值

sum() 求和

count() 取记录数

其他函数

decode()

nvl() 处理空字符串

cast() 强制转换数据类型

运算表达式

数学运算 + - \* /

逻辑运算 > >= < <= = <> != Not And Or

位运算 bitand

特殊判式 between in like isnull exists all any some

分析函数和窗口函数

排名 rank() dense\_rank() row\_number()

分区 partition by

窗口子句 rows range

分析函数 first\_value() last\_value() lead() lag()

控制语句

条件 if else case when

循环 无条件循环 while循环 for循环

查询语句

基本查询 select where distinct order by group by having

子查询

联合查询 union union all intersect minus

联接查询 自然联接 内联接 左联接 右联接 完全联接

层次化查询

更新语句

插入 普通插入 利用子查询批量插入

修改 修改单列 修改多列

删除 truncate 删除

索引和约束

使用索引提高查询速度

使用约束保证数据静态完整性

一致性，隔离级别

使用事务保证数据一致性 commit roll bac savepoint roll back to set transaction set constraint

事务处理的特性 原子性 一致性 隔离性 持久性

并发和锁

表级锁 和行级锁

死锁的处理方式

悲观锁

乐观锁

锁定转换

并发控制的其他方法

正则表达式

regexp\_like()

regexp\_instr()

regexp\_substr()

regxp\_replace()

Java相关

使用java程序操作

**C#相关**

使用OracleHelper操作

整理OracleHelper

连接字符串示例connectString = "Data Source=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=192.168.11.175)(PORT=1521))(CONNECT\_DATA=(SERVICE\_NAME=oraclexl)));Persist Security Info=True;User ID=usera;Password=Xl711436;";

增删查改代码示例

object tempResult = OracleHelper\_Lite.ExecuteScalar(connectString, CommandType.Text, "select count(\*) from TEST\_COURSE ");

tempResult = OracleHelper\_Lite.ExecuteScalar(connectString, CommandType.Text, @"insert into test\_student (name) values('dd')");

tempResult = OracleHelper\_Lite.ExecuteScalar(connectString, CommandType.Text, @"update test\_student set name = name || '1' where name like '%cc%'");

tempResult = OracleHelper\_Lite.ExecuteScalar(connectString, CommandType.Text, @"delete from test\_student where id = (select max(id) from usera.test\_student )");

使用EF操作

下载Oracle Data Provider for .NET ，下载 Oracle Developer Tools for Visual Studio 2017

新建实体数据模型，在向导中新建连接字符串时选择oracle驱动

编辑对应的tnsnames.ora 文件，

后续操作和sqlserver类似，使用code first 或者modelfirst,写linq操作语句

**自增列的实现**

建表 create table test\_student( id number primary key,name varchar2(20))

建立序列 create sequence test\_sutdent\_seq increment by 1 start with 2 minvalue 2 maxvalue 9999999999999 nocache order;

建立触发器

create or replace trigger test\_sutdent\_trigger

before insert on test\_student

for each row

begin

select test\_sutdent\_seq.Nextval into:new.id from sys.dual ;

end;

插入数据 insert into test\_student (name) values('bb');

数据操作

Update 当在命令行中进行数据修改时，需要输入commit 命令进行提交后才会在服务端生效

Pl/sql develop的操作

备份 还原

建立数据表

修改表结构

删除表

对表中数据进行导入导出

建立各种约束

提交和回滚

复杂查询

聚合

联合查询

分页

子查询

多表查询

联合查询