# Oracle

## 新用户创建表过程

新用户创建表过程：创建数据库-系统管理员-创建表空间-创建用户-授权用户-创建表

第0步 创建数据库(实例)在开始菜单Database进行操作

--查看实例（也就是当前全局数据库名）

select name from v$database

第一步 系统管理员账户进入数据库

命令 sqlplus 用户名/密码

第二步 创建表空间

永久性表空间：一般保存表、视图、过程和索引等的数据

临时性表空间：只用于保存系统中短期活动的数据（不指定一般都是创建的永久性表空间）

撤销表空间：用来帮助回退未提交的事务数据

-------------------------------------------------------------------------------------------------------------------------------------------------------------

创建表空间

CREATE TABLESPACE 表空间名

DATAFILE '数据文件路径' SIZE 大小 -- SIZE为初始表空间大小，单位为K或者M

[AUTOEXTEND ON] [NEXT 大小] -- AUTOEXTEND ON 是否自动扩展, NEXT为文件满了后扩展大小

[MAXSIZE 大小]; --MAXSIZE为文件最大大小，值为数值或UNLIMITED（表示不限大小）

CREATE TABLESPACE wwl

DATAFILE 'd:\oracle\_data\itcast01.dbf' SIZE 10M

AUTOEXTEND ON;

查询表空间

--查所有的表空间

select \* from v$tablespace

--管理员角色指定查看表空间详情

SELECT file\_name,tablespace\_name,bytes,autoextensible

FROM dba\_data\_files

WHERE tablespace\_name='WWL'; --注意这里必须是大写，虽然你创建表空间的名字是小写，但Oracle会转为大写

--查询表空间详细的文件在哪？

select file\_name,tablespace\_name from dba\_data\_files;

修改表空间

ALTER TABLESPACE wwl

ADD DATAFILE 'd:\oracle\_data\itcast02.DBF' SIZE 5M 修改为5m

AUTOEXTEND ON;

删除表空间

【语法】

DROP TABLESPACE 表空间名;

DROP TABLESPACE 表空间名 INCLUDING CONTENTS AND DATAFILES;

【说明】

第一个删除语句只删除表空间；第二个删除语句则删除表空间及数据文件

DROP TABLESPACE itcast\_ts;

DROP TABLESPACE itcast\_ts INCLUDING CONTENTS AND DATAFILES;

第三步 创建用户 (一个表空间可以有多个用户)

创建用户

CREATE USER 用户名 IDENTIFIED BY 密码

DEFAULT TABLESPACE 表空间;

CREATE USER wwl IDENTIFIED BY wwl --创建用户名密码

DEFAULT TABLESPACE itcast\_ts --指定永久性表空间

TEMPORARY TABLESPACE temp; --指定临时的表空间 一般不用写

查询系统所有用户

select \* from all\_users;

或

select \* from dba\_users; --更详细的用户信息

修改用户密码

ALTER USER 用户名 identified by 密码

ALTER USER itcast identified by it;

删除用户

DROP USER 用户名 CASCADE;

DROP USER itcast CASCADE;

第四步 授权用户 DCL数据控制语言

授予权限操作

第一种方式 【此表空间的全部表】

GRANT 角色权限（角色）[,角色权限] TO 用户;

--授予CONNECT和RESOURCE两个角色 connect：授权可连接，resource

GRANT connect, resource TO itcast;

使用如下语句可以查看resource角色下的权限

SELECT \* FROM DBA\_SYS\_PRIVS WHERE GRANTEE='RESOURCE'

第二种方式 【单表】

GRANT 操作 ON 模式.对象 TO 用户;

--允许用户查看、更新 EMP 表中的记录

GRANT select,update ON SCOTT.emp TO itcast;

--查看当前用户的系统权限

select \* from user\_sys\_privs;

--查看当前用户的对象权限

select \* from user\_tab\_privs;

--查看当前用户的所有角色

select \* from user\_role\_privs;

撤销权限操作

【此表空间的全部表】

REVOKE 角色权限（角色）[,角色权限] FROM 用户;

--撤销CONNECT和RESOURCE两个角色

REVOKE connect, resource FROM itcast;

【单表】

REVOKE 操作 ON 模式.对象 FROM 用户;

--撤销用户查看、更新 EMP 表中的记录的操作

REVOKE select,update ON SCOTT.emp FROM itcast;

第五步 创建表 （Oracle的回收站）

创建表

【语法】

CREATE TABLE <table\_name>(

column1 DATATYPE [NOT NULL] [PRIMARY KEY],

column2 DATATYPE [NOT NULL],

...

[constraint <约束名> 约束类型 (要约束的字段)

... ] );

【说明】

DATATYPE --是Oracle的数据类型

NUT NULL --可不可以允许资料有空的（尚未有资料填入）

PRIMARY KEY --是本表的主键

constraint --是对表里的字段添加约束.(约束类型有

Check,Unique,Primary key,not null,Foreign key);

create table t\_student(

s\_id number(8) PRIMARY KEY,

s\_name varchar2(20) not null,

s\_sex varchar2(8),

clsid number(8),

constraint u\_1 unique(s\_name),

constraint c\_1 check (s\_sex in ('MALE','FEMALE'))

);

修改表

表中添加新字段

ALTER TABLE <table\_name> ADD (字段1 类型 [NOT NULL],

字段2 类型 [NOT NULL] ... );

alter table t\_student add (s\_age number(3),s\_address varchar2(20));

修改表中字段

ALTER TABLE <table\_name> MODIFY(字段1 类型,字段2 类型 ... );

alter table t\_student modify(s\_name varchar2(50),s\_address varchar2(100));

删除表中字段

ALTER TABLE <table\_name> DROP(字段1,字段2... );

alter table t\_student drop(s\_age,s\_address);

修改表字段名称

ALTER TABLE <table\_name> RENAME COLUMN 原字段名称 TO 新字段名称;

alter table t\_student rename column s\_id to s\_no;

删除表

【语法1】

--删除表结构及数据（删除后可在回收站查看并恢复）

DROP TABLE <table\_name>;

--删除表结构及数据（删除后不可在回收站查看并恢复）

DROP TABLE <table\_name> PURGE;

回收站

--查看回收站

show recyclebin; 或 select \* from recyclebin;

--清空回收站

purge recyclebin;

## 数据类型

VARCHAR2(size)

可变长度的字符串,其最大长度为size个字节;size的最大值是4000,而最小值是1;你必须指定一个VARCHAR2的size;

NVARCHAR2(size)

可变长度的字符串,依据所选的国家字符集,其最大长度为size个字符或字节;size的最大值取决于储存每个字符所需的字节数,其上限为4000;你必须指定一个NVARCHAR2的size;

NUMBER(p,s)

精度为p并且数值范围为s的数值;精度p的范围从1到38;数值范围s的范围是从-84到127;

例如:NUMBER(5,2) 表示整数部分最大3位，小数部分为2位；NUMBER(5,-2) 表示数的整数部分最大为7其中对整数的倒数2位为0,前面的取整。NUMBER 表示使用默认值,即等同于NUMBER(5);

LONG

可变长度的字符数据,其长度可达2G个字节;

DATE

有效日期范围从公元前4712年1月1日到公元后9999年12月31日

RAW(size)

长度为size字节的原始二进制数据,size的最大值为2000字节；你必须为RAW指定一个size;

LONG RAW 可变长度的原始二进制数据，其最长可达2G字节;

CHAR(size)

固定长度的字符数据,其长度为size个字节;size的最大值是2000字节,而最小值和默认值是1;

NCHAR(size)

也是固定长度。根据Unicode标准定义

CLOB

一个字符大型对象,可容纳单字节的字符;不支持宽度不等的字符集;最大为4G字节

NCLOB

一个字符大型对象,可容纳单字节的字符;不支持宽度不等的字符集;最大为4G字节;储存国家字符集

BLOB

一个二进制大型对象;最大4G字节

BFILE

包含一个大型二进制文件的定位器,其储存在数据库的外面；使得可以以字符流I/O访问存在数据库服务器上的外部LOB;最大大小为4G字节.

## 其它

系统用户

sys 超级用户，主要用来维护系统信息和管理实例，以SYSDBA或SYSOPER角色登录。密码为在安装时设置的管理口令，如一般设置为：orcl

system 默认的系统管理员，拥有DBA权限，通常用来管理Oracle数据库的用户、权限和存储，以Normal方式登录。密码为在安装时设置的管理口令，如一般设置为：orcl

Oracle 11g数据库服务

查询系统用户

select \* from all\_users;

或

select \* from dba\_users; --更详细的用户信息

解锁用户

【语法】

ALTER USER 用户名 ACCOUNT UNLOCK;

【示例】解锁hr用户

alter user hr account unlock

Oracle 11g数据库服务

Oracle \* VSS Writer Service – Oracle

卷映射拷贝写入服务，VSS（Volume Shadow Copy Service）能够让存储基础设备（比如磁盘，阵列等）创建高保真的时间点映像，即映射拷贝（shadow copy）。它可以在多卷或者单个卷上创建映射拷贝，同时不会影响到系统的系统能。(非必须启动)

OracleDBConsole\*

-- Oracle数据库控制台服务；在运行Enterprise Manager(企业管理器EM)的时候，需要启动这个服务；此服务被默认设置为自动开机启动的(非必须启动)

OracleJobScheduler\*

-- Oracle作业调度服务。此服务被默认设置为禁用状态(非必须启动)

OracleMTSRecoveryService

-- 服务端控制。该服务允许数据库充当一个微软事务服务器MTS、COM/COM+对象和分布式环境下的事务的资源管理器。恢复、闪回需要开启该服务(非必须启动)

OracleOraDb11g\_home1ClrAgent

-- Oracle数据库.NET扩展服务的一部分。 (非必须启动)

OracleOraDb11g\_home1TNSListener

-- 监听器服务，服务只有在数据库需要远程访问或使用SQL Developer等工具的时候才需要，此服务被默认的设置为开机启动(非必须启动)

OracleService\*

-- 数据库服务，是Oracle核心服务该服务，是数据库启动的基础， 只有该服务启动，Oracle数据库才能正常操作。此服务被默认的设置为开机启动。(必须启动)

oracle原理（组成部分）

1、数据库是一系列物理文件的集合（数据文件，控制文件，联机日志，参数文件等）；Oracle数据库由操作系统文件组成，这些文件也称为数据库文件，为数据库信息提供实际物理存储区。Oracle数据库包括逻辑结构和物理结构。数据库的物理结构包含数据库中的一组操作系统文件。数据库的逻辑结构是指数据库创建之后形成的逻辑概念之间的关系，如表、视图、索引等对象。

2、实例则是一组Oracle后台进程/线程以及在服务器分配的共享内存区。

Oracle可以创建多个oracle数据库，一个oracle数据库将又由实例和数据库构成。如默认安装时创建的orcl数据库外还可再创建其它数据库。创建的数据库将在$oracleHome/oradata/数据库名 目录下以一个个的\*.DBF文件体现出来。

# SQLite3

非常简单的，直接打开软件视图sqlitestudio3.1.1 没有什么用户之类的

C#最简单的连接数据库操作

第一步 在工程目录下创建目录 Lib：把System.Data.SQLite.dll和System.Data.SQLite.xml放入

第二步 工程中导入引用 导入命名空间

第三步 往控件dataGridView显示数据库信息

控件的引用：工具箱—》数据---》dataGridView

第四步 一序列的连接数据库操作

private void Form1\_Load(object sender, EventArgs e) {

//操作数据库的类

//连接Connection

//命令Command

//适配器DataAdapter

//DataReader DataSet，DataTable

//从数据库表ManagerInfo中查询数据，

//开始 构造接收数据的集合

List<ManagerInfo> m = new List<ManagerInfo>();

//第一步 连接字符串

string connStr = @"data source=f:ItcastCater.db;version=3;";

//第二步 创建连接对象

using (SQLiteConnection conn = new SQLiteConnection(connStr)) {

//第三步 创建Command对象

SQLiteCommand cmd = new SQLiteCommand("select \* from ManagerInfo", conn);

//第四步 打开连接

conn.Open();

//第五步 发送sql语句 执行命令

SQLiteDataReader reader= cmd.ExecuteReader();

//第六步 读取

if(reader.HasRows) {

while(reader.Read()){

m.Add(new ManagerInfo(){

Mid=Convert.ToInt32(reader["mid"]),

Mname=reader["mname"].ToString(),

Mpwd=reader["mpwd"].ToString(),

Mtype=Convert.ToInt32(reader["mtype"])

}); } }}

//将显示到DataGridView上

dataGridView1.DataSource=m;

# SQL Server2012

## 简单步骤

第一步 登录操作 SQL Server Management Studio文件名

windows身份验证 . 一个小点 代表本机意思 一般都不用此方式

默认实例名：MSSQLSERVER 没有特别指定就是进入默认实例名

./wwl 进入自己创建的实例库

sql server身份验证 就是用自己的账户和密码登录

服务器名称：就是ip地址 本机 127.0.0.1 然后账户密码登录

启用sa账户操作：windows身份验证登录—》安全性—》登录名—》sa—》修改密码—》强制实施密码策略关掉--》状态—》设置为 ：授予和已启用

强制实施密码策略：不能简单的密码，必须繁琐的密码

登录操作：服务器名称 127.0.0.1 身份验证 sql server身份验证 登录名 sa 密码 123456

第二步 创建数据库 数据库右击新建数据库

注意 初始大小最小5M 其它可以不变了

第三步 新建表 打开新建好数据库wwl 在’表’中右击创建

nvarchar(32): 两个字节 保存中文用，一般都用这个属性字符串

varchar(32): 1个字节

int 数字一般都用这个属性

设置主键：右击 –》列属性—》标识规范：是，标识增量 自动增长次数 标识种子 从第几个开始增长

注意 保存表可能遇到的问题：“阻止保存要求重新创建表的更改”的问题，解决办法： 工具→选项→设计器→阻止表的更改

远程连接我的数据库

计算机—》管理—》 服务和应用程序—》sql server配置管理器—》sql server 网络配置—》tcp/ip启用

## 字段类型

字符类型

有var的都是可变长度 节省空间 性能低

nvarchar(32): 两个字节 最多32个字节，保存中文用，一般都用这个属性字符串

varchar(32): 1个字节

无var的不可变，设定多少个，就多少个， 性能高 浪费空间

nchar(5) 固定5个字节，保存的值没有足够5个字节，自动补全

日期类型

datetime

## 其它

数据库迁移

第一种方式 ：分离 附加操作

第一步 ：选中数据库—》右击—》任务—分离—》删除连接选中—》完成

第二步 ：进入数据库目录复制两个文件 ：ww\_log.ldf和主文件wwl.mdf

找目录：新建数据库 ---》就可以看到目录位置了

第三步 迁移到新的服务器 右击数据库 –》附加—》添加—》选中—》wwl.mdf🡪确定即可

注意问题 如果出现问题一般都是权限问题，如是高版本要在低版本：建数据库设置成2005兼容 ：新建数据—选项

第二种方式 ：脱机 附加 可以说是复制一份给新机服务器。操作和上面一样

第三种方式： 备份 还原 操作 都和第一种方式操作差不多的

# sql语句优化

1。你必须选择记录条数最少的表作为基础表.

(from 是从前往后检索的，所以要最少记录的表放在最前面)

2。采用自下而上的顺序解析WHERE子句,根据这个原理,表之间的连接必须写在其他WHERE条件之前, 那些 可以过滤掉最大数量记录的条件必须写在WHERE子句的末尾。同时在链接的表中能过滤的就应该先进行过滤。

（where是从后往前检索，所以能过滤最多数据的条件应放到最后。）

3。SELECT子句中避免使用 ‘ \* ‘

4。尽量多使用COMMIT

5。计算记录条数时候，第一快：count(索引列)，第二快：cout(\*)

6。用Where子句替换HAVING子句

7。通过内部函数提高SQL效率

8。使用表的别名(Alias)

9。用EXISTS替代IN

10。用NOT EXISTS替代NOT IN

11。用表连接替换EXISTS

12。用索引提高效率

13。避免在索引列上使用计算（此条包括在select后面 where后面等任何地方，因为在索引列上计算会导致索引失效）

14。避免在索引列上使用NOT（在索引列使用not会导致索引失效）

15。用>=替代>

16。用UNION替换OR (适用于索引列)

17。用IN来替换OR

18。避免在索引列上使用IS NULL和IS NOT NULL

19。总是使用索引的第一个列

20。用UNION-ALL 替换UNION ( 如果有可能的话)

21。ORDER BY 子句只在两种严格的条件下使用索引.

22。避免改变索引列的类型

23。需要当心的WHERE子句

24。避免使用耗费资源的操作（带有DISTINCT,UNION,MINUS,INTERSECT,ORDER BY）

# mariadb

systemctl start mariadb #启动MariaDB

systemctl stop mariadb #停止MariaDB

systemctl restart mariadb #重启MariaDB

systemctl enable mariadb #设置开机启动

# mysql

## 备份常用操作基本命令

1、备份命令mysqldump格式

格式：mysqldump -h主机名 -P端口 -u用户名 -p密码 –database 数据库名 > 文件名.sql

2、备份MySQL数据库为带删除表的格式

备份MySQL数据库为带删除表的格式，能够让该备份覆盖已有数据库而不需要手动删除原有数据库。

mysqldump --add-drop-table -uusername -ppassword -database databasename > backupfile.sql

3、直接将MySQL数据库压缩备份

mysqldump -hhostname -uusername -ppassword -database databasename | gzip > backupfile.sql.gz

4、备份MySQL数据库某个(些)表

mysqldump -hhostname -uusername -ppassword databasename specific\_table1 specific\_table2 > backupfile.sql

5、同时备份多个MySQL数据库

mysqldump -hhostname -uusername -ppassword –databases databasename1 databasename2 databasename3 > multibackupfile.sql

6、仅备份份数据库结构

mysqldump –no-data –databases databasename1 databasename2 databasename3 > structurebackupfile.sql

7、备份服务器上所有数据库

mysqldump –all-databases > allbackupfile.sql

8、还原MySQL数据库的命令

mysql -hhostname -uusername -ppassword databasename < backupfile.sql

9、还原压缩的MySQL数据库

gunzip < backupfile.sql.gz | mysql -uusername -ppassword databasename

10、将数据库转移到新服务器

mysqldump -uusername -ppassword databasename | mysql –host=\*.\*.\*.\* -C databasename

11、--master-data 和--single-transaction

在mysqldump中使用--master-data=2，会记录binlog文件和position的信息 。--single-transaction会将隔离级别设置成repeatable-commited

12、导入数据库

常用source命令，用use进入到某个数据库，mysql>source d:\test.sql，后面的参数为脚本文件。

13、查看binlog日志

查看binlog日志可用用命令 mysqlbinlog binlog日志名称|more

14、general\_log

General\_log记录数据库的任何操作，查看general\_log 的状态和位置可以用命令show variables like "general\_log%" ,开启general\_log可以用命令set global general\_log=on

## 增量备份

小量的数据库可以每天进行完整备份，因为这也用不了多少时间，但当数据库很大时，就不太可能每天进行一次完整备份了，这时候就可以使用增量备份。增量备份的原理就是使用了mysql的binlog志。

1、首先做一次完整备份：

mysqldump -h10.6.208.183 -utest2 -p123 -P3310 --single-transaction --master-data=2 test>test.sql这时候就会得到一个全备文件test.sql

在sql文件中我们会看到：

-- CHANGE MASTER TO MASTER\_LOG\_FILE='bin-log.000002', MASTER\_LOG\_POS=107;是指备份后所有的更改将会保存到bin-log.000002二进制文件中。

2、在test库的t\_student表中增加两条记录，然后执行flush logs命令。

这时将会产生一个新的二进制日志文件bin-log.000003，bin-log.000002则保存了全备过后的所有更改，既增加记录的操作也保存在了bin-log.00002中。

3、再在test库中的a表中增加两条记录，然后误删除t\_student表和a表。

a中增加记录的操作和删除表a和t\_student的操作都记录在bin-log.000003中。

## 恢复

1、首先导入全备数据

mysql -h10.6.208.183 -utest2 -p123 -P3310 < test.sql，也可以直接在mysql命令行下面用source导入

2、恢复bin-log.000002

mysqlbinlog bin-log.000002 |mysql -h10.6.208.183 -utest2 -p123 -P3310

3、恢复部分 bin-log.000003

在general\_log中找到误删除的时间点，然后更加对应的时间点到bin-log.000003中找到相应的position点，需要恢复到误删除的前面一个position点。

可以用如下参数来控制binlog的区间

--start-position 开始点 --stop-position 结束点

--start-date 开始时间 --stop-date 结束时间

找到恢复点后，既可以开始恢复。

mysqlbinlog mysql-bin.000003 --stop-position=208 |mysql -h10.6.208.183 -utest2 -p123 -P3310