Instance**例程**：包含内存和进程结构的访问唯一开放数据库的例程

sys用户：sysdba（管理服务器例程+数据库）、sysoper（管理例程）

system用户：管理数据库

内存：**SGA**（系统全局区，例程启动分配）、**PGA**（程序全局区、会话启动分配）

SGA：共享池（库缓存(代码)、数据字典缓存(系统数据：数据字典(数据文件)/动态性能**v$**(内存+控制文件)视图)）、数据库高速缓存、Redo log重做日志缓存

RBO：基于规则优化器 CBO：基于开销优化器（依据统计数据）

回滚：对冲操作（delete—insect）

Restore：备份文件还原

Cache +Buffer：提高读+写性能（从cache中读写先写入buffer）

数据字典基表中的数据很难看懂。因此，很少人直接访问这些基表。取而代之的是数据字典视图。

USER\_\* 用户所拥有的对象信息

ALL\_\* 用户能访问的对象信息

DBA\_\* 整个数据库中的对象信息

数据字典视图：

dba\_objects

dba\_tables

dba\_users

dba\_tablespaces

动态性能视图：

还有一类数据字典视图：动态性能视图，动态性能视图只存在于运行的数据库中（都是虚拟表，关闭时丢失），只有数据库管理员可以查询，以v$为前缀

v$controlfile包含控制文件存储目录和文件名信息，

v$datafile包含数据库文件信息

v$database 数据库信息

v$fixed\_table视图包含了当前所有动态性能视图

v$log重做日志组, v$logfile重做日志文件, v$instance实例信息, v$session会话信息, v$parameter参数信息，v$tablespace表空间信息

进程：用户（通过服务器进程访问服务器）、服务器（用户进程访问代理）、守护（后台工作）

服务器进程：专用（便于管，效率高）、共享（支持连接数多）、连接池

Pmon：后台错误处理（清理内存，释放资源）

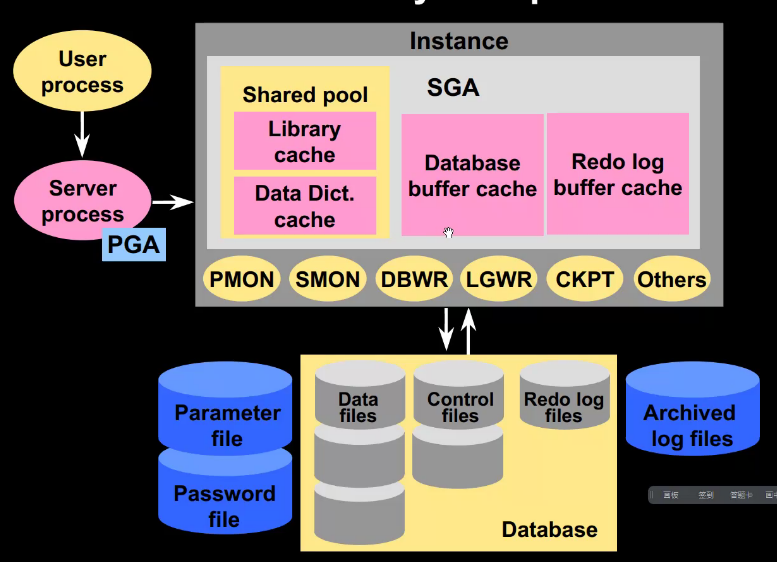
Smon：系统监视器，例程恢复

Dbwr：数据书写器

Lgwr：日志书写器

Ckpt：检查点进程（恢复时比较，同步数据/控制文件，**通知Dbwr写盘**）

日志体系：日志/归档模式



SQL语句：QL，**DML**(增删改)，DDL(create, alter, drop)，DCL(权限grant, revert)，TCL(事务)

语句：哈希值匹配执行（大小写哈希不同），exact/模糊（不可缩写）

NULL：未确定，不能与具体值相比，不可=，order by时在末尾

**TOP n功能**：rowlimited

前N个：fetch first N rows only/with ties(找出相同)

去除前N个开始：offset N rows

交互：

&，&&(自动define可复用)占位符，后接提示，字符串加引号并控制大小写

Define定义临时变量

函数：单行（影响每条记录）、组函数（多行返回一个结果）

CONCAT（连接字符串）、SUBSTR（子串0起,负数为从右开始）、INSTR（字符出现位置）

TRUNC（取整）、ROUND（四舍五入）、CEIL/FLOOR（向上/下取整）、MONTH\_BETWEEN

NLS：国际参数

日期参数：**nls**\_date\_format ‘yyyy-mm-dd hh24:mi:ss’

YY日期：当前世纪

RR日期：最接近当前日期

转换函数：to\_date/to\_char，参数填写格式

NVL(不空, 空)：设置NULL时显示策略（NVL2：第一项不空显示第二项，否则显示第三项）

NULLIF(exp1, exp2)：判断相等填空

COALESCE(exp1, exp2, …)找到第一个非空值，否则返回NULL(exp为相同类型)

**decode两种表示方式**：

decode(expression,value,result1,result2)

如果expression=value，则输出result1，否则输出result2

decode(expression,value1,result1,value2,result2,value3,result3......,default)

如果expression=value1，则输出result1，expression=value2，输出reslut2，expression=value3，输出result3，若expression不等于所列出的所有value，则输出为default

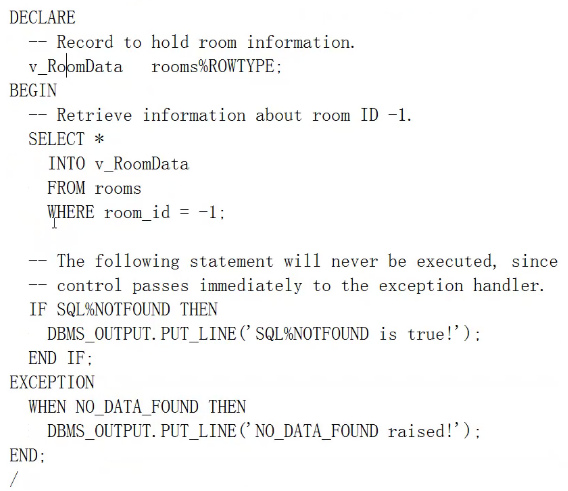
**group by rollup(a，b)**基于b相同的小计对a统计+总计

**group by cube(a, b)** 总计+对b统计（将a制空）+再对a统计

sql表内连接：后表做驱动，嵌套循环（使用小表驱动）

sql外连接：2 (+)不支持全外连接、3 left/right/full join

USING：指定匹配字段，不加表名



无条件循环：loop

条件循环：while

固定次数循环：for

创建存储过程：Create or declare procedure p as begin ... end

DDL：修改Oracle内部数据（表、视图、序列、索引、同义词）

表命名：字母开头、1-30字符、可用\_$#、同一用户下**相同名称空间**对象不重名、非Oracle关键字

Oracle17+：创建表时不分配空间，允许延时创建

帮其余用户建：grant create any table

伪列：rowid（物理地址：文件、块、行号）不依赖于表的表中列索引指针

Varchar/char：符号来自数据库字符集（快）

nvarchar：符号来自国家字符集（慢）

clob/blob：大型对象，最大4GB

raw：字符的二进制格式

修改表名：rename 原表名 更新后

修改列名：alter table t1 rename column 原列名 to 更新后列

修改类型/宽度：alter table t1 modify id1 类型(宽度)（改类型时列必须为空, char改不短）

增删字段：alter table t1 add 字段名 类型/ alter table t1 drop (字段名)

标记为未使用：alter...set unused column 然后删除（drop unused columns）(不删数据，快)

删除表：drop table buyers purge(彻底删除)

truncate/delete清空表记录：DDL语句自动提交无法回滚, 速度快, 高端水印下移(最大空间记录)释放空间/DML语句手动提交(复制到回滚段)，不释放空间

数据完整性：代码、触发器、约束（非空, unique, 主键, 外键reference, check）

约束：关键字、约束名、类型、字段名(表级别)（disable）

视图：收集某些字段，屏蔽部分信息，简化权限管理（replace不更改视图权限）

Create or replace force view as (select语句)

rownum：结果出现后编号，使用需嵌套子查询

序列：创建、调用（cache加入缓存）

例程管理：

关闭：shutdown

1. 正常N，不允许行连接
2. 事务性T，不等连接
3. 立即I，不等事务
4. 终止A，不做检查点

启动：startup

1. 例程，初始化参数文件，访问内存动态性能视图
2. 加载数据库Mount，加载控制文件，访问所有动态性能视图
3. 打开数据库，加载数据与日志文件，完整打开

配置：初始化参数（文本，二进制）、查看参数（show parameter <模糊参数名>）、修改动态参数（alter system set <参数名>=参数值 scope=memory/spfile/both）

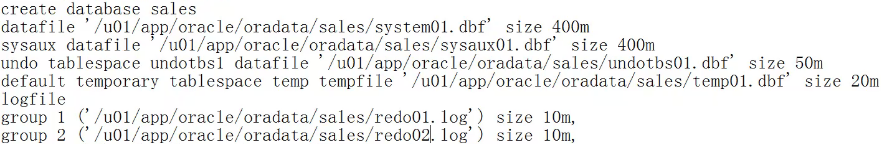
基本初始化参数：数据库名(启动)、控制文件(正常打开)

修复错误初始化参数：根据spfile创建pfile，更改响应参数，根据pfile创建spfile

参数文件选择顺序：spfile<sid>.ora、spfile.ora、init<sid>.ora

手工创建数据库：创建/编辑初始化参数文件、目录结构、启动例程、执行建库语句（创建表、数据文件、sysaux、撤销表空间、默认临时表空间、多组日志文件）





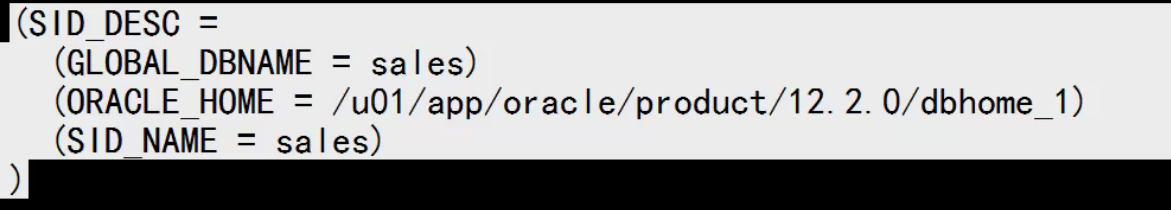
创建数据库字典：运行脚本@?/rdbms/admin/catalog

创建口令验证：orapwd file=orapwsales password=admin1#3

创建Oracle内部包：运行脚本@?/rdbms/admin/catproc

创建scott方案、加载产品用户概要文件

配置监听器（服务器端）和服务名（用户端）



配置EM Express

管理控制文件：查看（oradebug dump写入文本文件）、复用（alter多个初始化参数文件）

另存控制文件，按照命令重新创建，执行其中命令：



管理日志文件：查看组（v$log）、成员信息（v$logfile）、添加组/成员（添加镜像）、删除（切换为不使用，不删除物理文件，drop）、OMF（设置参数位置，自动管理）



清除日志：clear logfile group N（解决损坏日志）

数据库模式：mount下，alter database archivelog改为**归档**模式

手工归档：设置归档日志位置，设置归档模式，手工生成归档日志







移动/重命名日志：关闭，解除日志使用，移动日志（开至mount），更改控制文件标记位置



日志丢失：非当前，alter database clear logfile group N、当前，recover database until cancel;

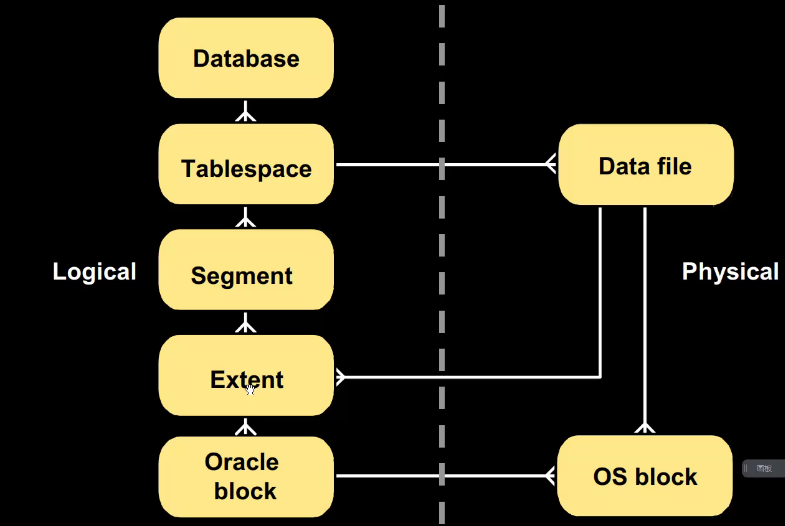
表空间：对应唯一数据库，包含数据文件

簇表：多张小表放在一起，需同时使用和存储提高效率（先创建簇，创建表时放入簇）

段：存储结构

区：数据文件最小空间分配单位

块：最小IO读写单位（OS块整数倍）



本地管理表空间：位图记录是否使用、统一大小

创建表空间，更改默认表空间，移动表所在的表空间，表空间放入多个数据文件，查看表空间文件







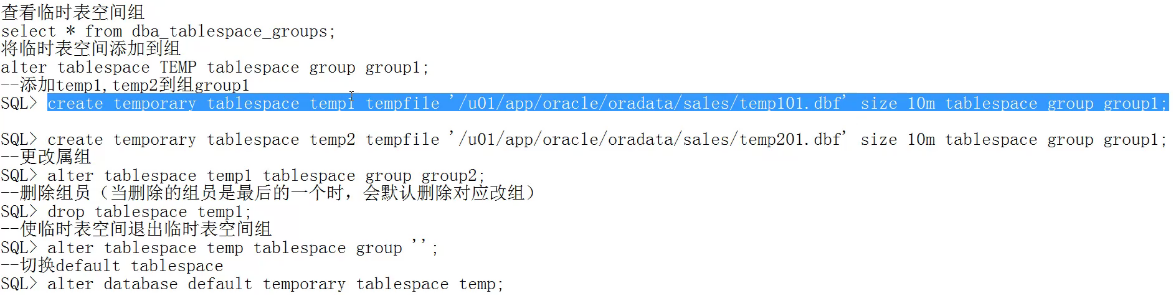




大文件表空间：文件空间无限，顺序访问适用



表空间类型：常规、撤销（undo）、临时（排序时做虚拟内存）、临时表空间组



表空间状态：只读（表空间数据文件只读, 数据字典可增定义）、联机可读写、脱机（暂时不许访问，system不可, sysaux(辅助系统表空间)可脱机但失去自优化，当前撤销表空间不可，）

扩展表空间：修改数据文件大小（自动（后台进程，每次加5m） /手工）、添加新数据文件





12c前：保证文件不被使用，用系统命令移动，调整

12c后移动文件：



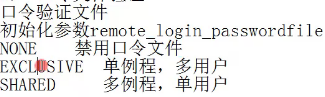
脱机损坏文件，正常使用数据库，通过日志修复，重新联机

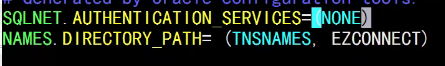
安全3A：验证、授权、审核

验证：OS验证（信任系统用户,优先）、口令文件（禁用OS验证）

创建口令验证文件，初始化参数

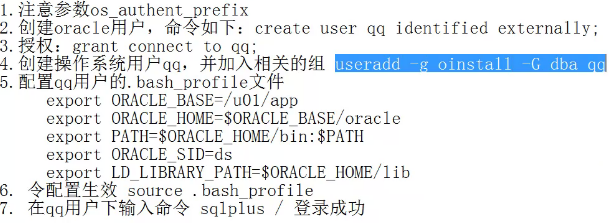
(sys)

禁用OS验证：更改/u01/app/oracle/product/12.2.0/dbhome\_1/network/admin/sqlnet.ora



用户加入v$pwfile\_user，grant相应身份

普通用户操作系统验证



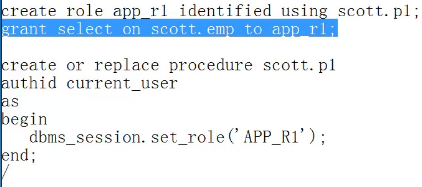
权限：系统（create..）、对象（查表..）；权力（全局，用户）、权限（局部，资源）

权限传递规则：sys-a-b; 对象权限（grant option连带收回），系统权限（admin不连带）

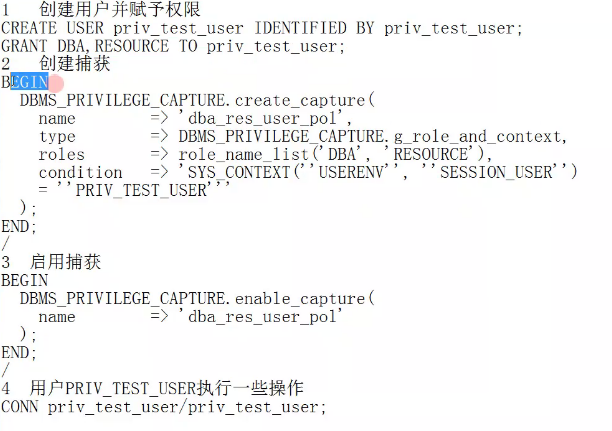
自定义角色权限：权限的集合、非默认用户赋予用户需激活（set role r1）、动态控权限

系统预定义角色：connect(包含create session)、dba、resource（创建对象/开发）、scheduler（自动化作业）、select catalog(查看数据字典)

应用程序角色：通过应用激活、执行应用(存储过程)时激活（create role r1 identified using <应用>; 未指派权限，运行程序时集成临时角色）



最小权限：权限分析（创建捕获，监视操作，权限报告视图，创建需使用权限角色）、完成一件事所需基本



通过sqldeveloper.sh更改产生**概要文件**，解析语句所需资源，拒绝执行

CPU会话/调用，连接/空闲时间，并行会话

create profile p1 limit <限制参数> 限制值

alter user <用户> profile “pro1”; 指派profile

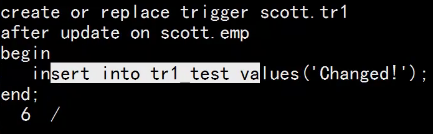
审核：强制（默认）、标准数据库（[extend]详细, xml应用, none, os,放入文件 db放入表）

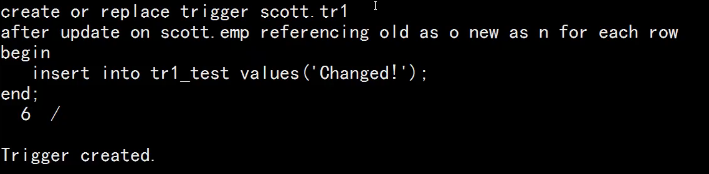
审核选项：用户（监控用户权限执行）、对象（监控对象操作）、语句（监控行为）

audit <审核条件> by session（取消为noaudit）

基于值审核：触发器实现（自动调用）dml，for each row逐行触发





记录历史值：

create or replace trigger scott.val

after update on scott.emp referencing old as o new as n for each row

begin

insert into scott.sal\_change values(:o.ename,:o.sal,:n.sal);

end;

/



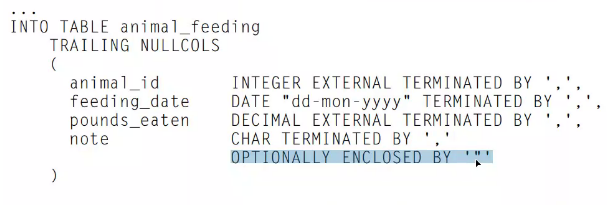
系统触发器：由系统事件触发（需要系统权限）

grant administer database trigger to scott;

从文件导入数据：load, 指定数据源，目标表，数据格式（记录、字段、名称、类型）

position无分隔时截取字段

optionally enclosed可选分节符”



交互式导出导入表：EXP\_FULL\_DATABASE导出表角色权限（**系统数据不允许导出**）

exp/imp服务器+客户端均可用

expdp/impdp服务器端（并行机制 大任务，资源充足时快）：

创建表空间directory，赋予读写权限



（ignore忽略已经存在）

部分导出：



使用临时表空间

授予读写权限，grant read,write on directory  dump\_dir1 to scott



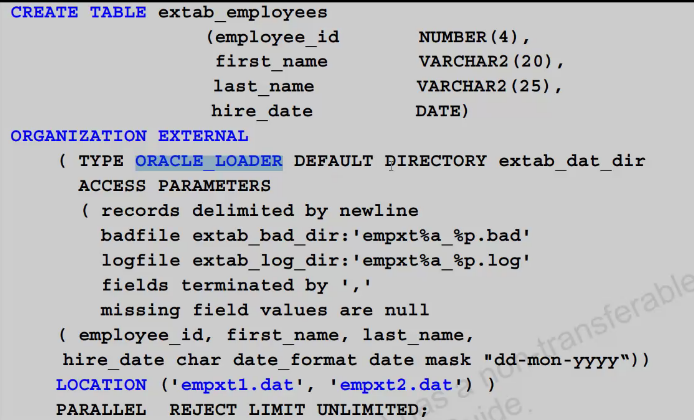
Impdp：remap\_table emp:emp1/\_schema/\_tablespace换表名

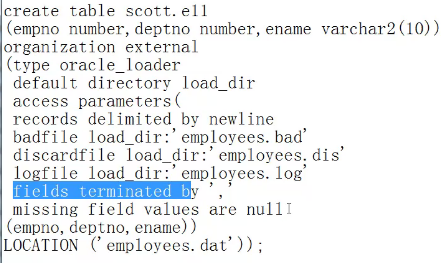
导入导出并更换表空间：expdp导出（使用临时空间）、create table from emp定义表、impdp导入并更换



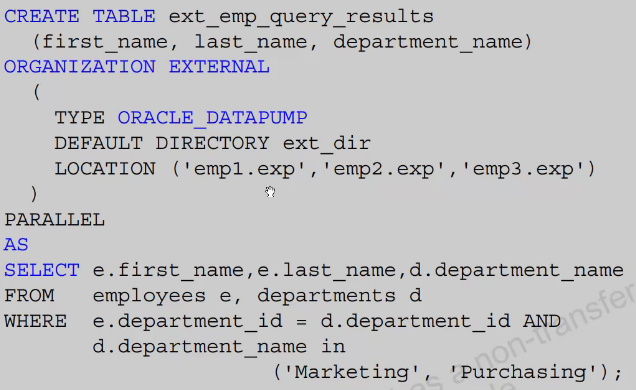
\_

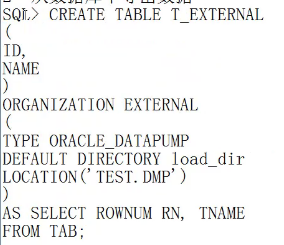
外部表：定义表（organization external）、驱动与目录、访问参数、字段顺序、数据文件

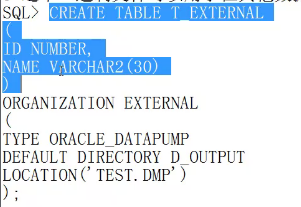




访问**二进制文件**：外部表通过select卸出二进制数据文件、使用dmp文件恢复









用户误操作：闪回（使用logmnr查找日志）

Logmnr：启用数据库补充日志、产生数据字典文件（可读形式）、开始事务、添加日志、查找









备份：冷（未开启）/热（联机）、用户/服务器（recover manager）管理备份、全/增量备份

**用户管理的**：

冷：shutdown状态，产生文件列表（初始化参数，控制，数据, 日志文件），创建测试表，备份整库，插入新记录（重备份控制与日志），恢复数据库(复制初始化参数，创建目录结构，启动至nomount，复制控制文件，加载数据库，复制日志+数据文件，recover database

show parameter spfile;

select name from v$controlfile;

select name from v$datafile;

select member from v$logfile;

select name from v$tempfile;

导入spfile，创建目录结构

startup nomount;

导入控制文件

alter database mount;

导入数据+日志文件

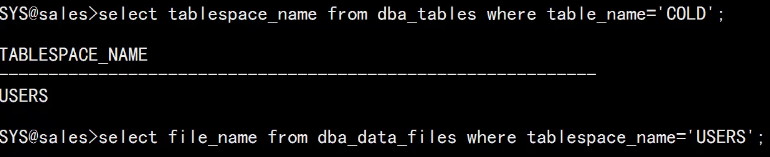
recover database;

打开数据库alter database open;



热备份：归档模式，表空间备份模式，备份数据文件，（更改表），结束备份，脱机，恢复

查找表的表空间与文件





热备份：（部分表空间）

热备份整库：alter database begin backup

**服务器管理的**：RMAN（recover manage）

mount---冷，open---热

连接时指定类型：catalog/target/auxiliary（备份目标/控制目录/辅助数据库）

根据控制文件备份，利用辅助数据库导出导入数据

restore tablespace users

recover tablespace users

repair failure

使用自动提醒