

TRS 命令语言

TRSL 6.5

用户手册

北京拓尔思信息技术股份有限公司

Beijing TRS Information Technology Co., Ltd.

版权说明

本手册由北京拓尔思信息技术股份有限公司（以下简称 TRS 公司）出版，版权属于 TRS 公司所有。未经出版者正式书面许可，不得以任何方式复制本文档的部分或全部内容。

©北京拓尔思信息技术股份有限公司 版权所有。保留所有权利。

TRS 是北京拓尔思信息技术股份有限公司的注册商标。

关于本手册

命令语言是独立于 TRS 管理员以外的一个仿命令行工具，用于输入和执行 TRS 命令语句。它的操作方法和 MS-DOS 的命令行工具基本一致，但增加了一些辅助的界面功能（如检索历史、操作日志、表达式生成器等）以方便用户的使用。

读者对象

本手册的主要读者为 TRS 全文数据库的系统管理员，但普通用户也可以使用。只要基本掌握 Windows 95/NT 操作系统的使用方法就可以很快地学会用 TRS 命令语言处理工具完成各种 TRS 系统操作。

用户反馈

TRS 公司感谢您使用 TRS 产品。如果您发现本手册中有错误或者产品运行不正确，或者您对本手册有任何意见和建议，请及时与 TRS 公司联系。您的意见将是我们做版本修订时的重要依据。

目 录

第 1 章 总体概述	1
1.1 TRS全文数据库概述	1
1.2 TRS命令语言概述	2
1.3 新增特性	3
1.3.1 采用全新的标签页式多文档管理及浮动窗口的主界面	3
1.3.2 新的系统选项对话框，采用属性表方式显示选项，每一项均有概要提示	4
1.3.3 语句中不同类型词法符号的彩色显示	4
1.3.4 语句输入过程提供多种提示功能	5
1.3.5 修改装入记录等语句的进度显示方式，可实现进度百分数等信息不断刷新显示	6
1.3.6 DROP、UPDATE、DELETE等语句增加了提示确认的功能，防止误操作	7
1.3.7 支持打开多个脚本文件进行编辑，同时还提供了若干新功能	8
1.4 启动和退出	8
1.5 登录和断开	10
第 2 章 界面操作	11
2.1 编辑和执行命令行	11
2.2 各窗口介绍	12
2.2.1 输出窗口	12
2.2.2 命令窗口	14
2.2.3 检索历史窗口	17
2.2.4 操作日志窗口	18
2.2.5 用户变量窗口	18
2.2.6 脚本编辑窗口	19
2.3 定制界面	22
2.3.1 输出窗口的显示状态	22
2.3.2 清除屏幕	22
2.3.3 修改显示字体	23
2.3.4 设置窗口背景	24
2.3.5 自动折行	24
2.4 使用辅助工具	25
2.4.1 定时运行脚本	25
2.4.2 表达式生成器	25
2.4.3 执行选定脚本	28
2.4.4 制表位	28
2.4.5 查找	28

2.5 处理执行结果	29
2.5.1 保存执行结果	29
2.5.2 发送执行结果	31
2.5.3 打印执行结果	31
2.6 修改设置参数	34
2.6.1 缺省登录	34
2.6.2 集成环境	35
2.6.3 检索选项	37
2.6.4 语法颜色	38
2.7 获取帮助	39
2.7.1 界面使用	39
2.7.2 语法参考	40
2.7.3 获取更多信息	40
第3章 语法详解	41
3.1 语法概述	41
3.2 语句列表	42
3.3 语法规则	43
3.3.1 ALTER 修改对象属性和结构	43
3.3.2 BACKUP 备份数据库	54
3.3.3 BROWSE 浏览检索结果	56
3.3.4 CLASSIFY 字段值分类	58
3.3.5 CLEAR 清除屏幕内容	60
3.3.6 COPY 复制数据库记录	61
3.3.7 CREATE 创建新对象	63
3.3.8 DELETE 删除数据	75
3.3.9 DROP 删除对象	77
3.3.10 ECHO 显示控制	79
3.3.11 EDIT 编辑检索结果	80
3.3.12 EXIT 退出命令语言处理工具	82
3.3.13 GRANT 给用户授权	83
3.3.14 HELP 显示帮助信息	85
3.3.15 INSERT 插入新记录	86
3.3.16 LOAD 把记录装进数据库	88
3.3.17 LOCK 锁定数据库	90
3.3.18 LOGIN 登录到服务器	92
3.3.19 LOGOUT 注销登录	94
3.3.20 MAIL 发送检索结果	95
3.3.21 MODIFY 修改记录	97

3.3.22	NOTEPAD 编辑文件.....	99
3.3.23	OPTIMIZE 优化数据库.....	100
3.3.24	PRINT 打印检索结果.....	102
3.3.25	QUIT 退出并清除操作日志.....	104
3.3.26	RECOVER 恢复数据库.....	105
3.3.27	REM 注释语句.....	107
3.3.28	RENAME 为对象改名.....	108
3.3.29	REVOKE 收回用户的权限.....	110
3.3.30	RUN 运行脚本文件.....	112
3.3.31	SAVE 保存检索结果.....	113
3.3.32	SELECT 对数据库/视图进行检索.....	115
3.3.33	SET 设置系统参数.....	118
3.3.34	SHOW 显示信息.....	125
3.3.35	SORT 对检索结果重新排序.....	130
3.3.36	SPOOL 输出到文件.....	131
3.3.37	UNLOCK 解锁数据库.....	132
3.3.38	UPDATE 更新数据库记录.....	133
3.3.39	VALIDATE 验证索引.....	135
附录A	语句列表.....	136
附录B	保留字.....	147

第1章 总体概述

1.1 TRS 全文数据库概述

TRS 全文数据库是由北京拓尔思信息技术股份有限公司开发的一个体系结构先进、功能强大的面向文本数据的数据库管理系统，广泛地用于各种企业信息数据库、以及企业信息门户的建设。TRS 全文数据库的核心技术为全文检索，它为各种格式文档的存储、管理和检索提供动力。它的主要功能和性能如下：

1. “零”空间膨胀率（-10%至 20%）。
2. 基于成本优化的查询算法（索引分区技术，多线程并行运算技术，面向全文检索的 Cache 技术），使得 G 级数据库查询速度达到亚秒级，并支持大量并发用户的同时访问。
3. 多线程设计，支持 SMP 体系结构。
4. 支持跨平台计算：支持 Windows, Linux, Sun Solaris, IBM AIX, HP-UX。
5. 除了提供规范的 C/C++ API 外，还提供 ADO 和 JavaBeans 组件，可以使用符合工业标准的开发工具进行快速开发（ASP 和 JSP）。
6. 采用开放的数据存贮管理机制，通过 TRS for RDBMS Gateway，可以透明地连接 oracle、db2、sql server、sybase 和 informix 五个主流的数据库管理系统。它是电子商务应用中全文检索引擎的最佳解决方案。
7. 支持 Unicode(WCS2)。
8. 领先的中文信息处理技术：内嵌汉语自动分词系统，支持按词索引、按字索引、按关键词索引、二元组索引，大大提高检索的准确性和响应时间。
9. 检索信息快、准而且全：允许使用文中的任意字、词、句和片段进行检索，提供了基于文献内容而不仅仅是文献外部特征的全文检索手段。TRS 所提供的按词和按用户自定义关键词进行索引和检索，以及基于知识词典的扩展检索功能，满足了特殊应用领域的高查准率和高查全率的要求。
10. 检索功能强大：全方位检索手段，提供了多达 48 种检索运算符。包括外部特征与正文内容的各种逻辑组合检索、位置检索、二次检索、渐进检索、历史检索、词根检索、大小写敏感检索、概念检索、对检索结果按与检索表达式的相关性和重要性程度排序等。
11. 安全性好：具有系统级、数据库级、记录级和字段级四级安全控制机制。
12. 提供功能强大的 TRS 管理员工具：数据库维护、数据库的安全性控制、数据的完整性和一致性、数据库备份和恢复、用户管理等功能。

1.2 TRS 命令语言概述

命令语言是独立于 TRS 管理员以外的一个仿命令行工具，用于输入和执行 TRS 命令语句。它的操作方法和 MS-DOS 的命令行工具基本一致，但增加了一些辅助的界面功能以方便用户的使用。有关命令语言的详细语法请参阅语法参考部分。

新版本的命令语言工具在布局上做了比较大的调整(如下图所示)，主界面采用了分割式窗体，将输入的语句命令和输出的界面分开。上方输出窗口为语句执行结果的显示窗口，包括语句执行的一般结果，检索结果的概览和细览都将在这个窗口中显示。下方命令窗口，用户输入的所有 TRS 命令语句都在这个窗口中显示。

界面上方除输出窗口外，还以标签页方式排列文件编辑窗口，可打开多个脚本文件和文本文件进行查看编辑。

下方除了命令窗口以外，还设置了检索历史、操作日志和用户变量三个子窗口。分别显示最近执行过的检索的历史表，最近执行过的命令行及执行结果，和当前定义的用户变量。这些内容以窗口列表的形式列出，将更加直观，用户在进行检索的同时，可以随时进行查看。

在开始菜单中选择“TRS 全文数据库”程序组中的“TRS 命令语言”菜单项，或以资源管理器、命令行或其它方式运行 TRSL.exe 程序，即可启动 TRS 命令语言。TRS 命令语言的界面如下：

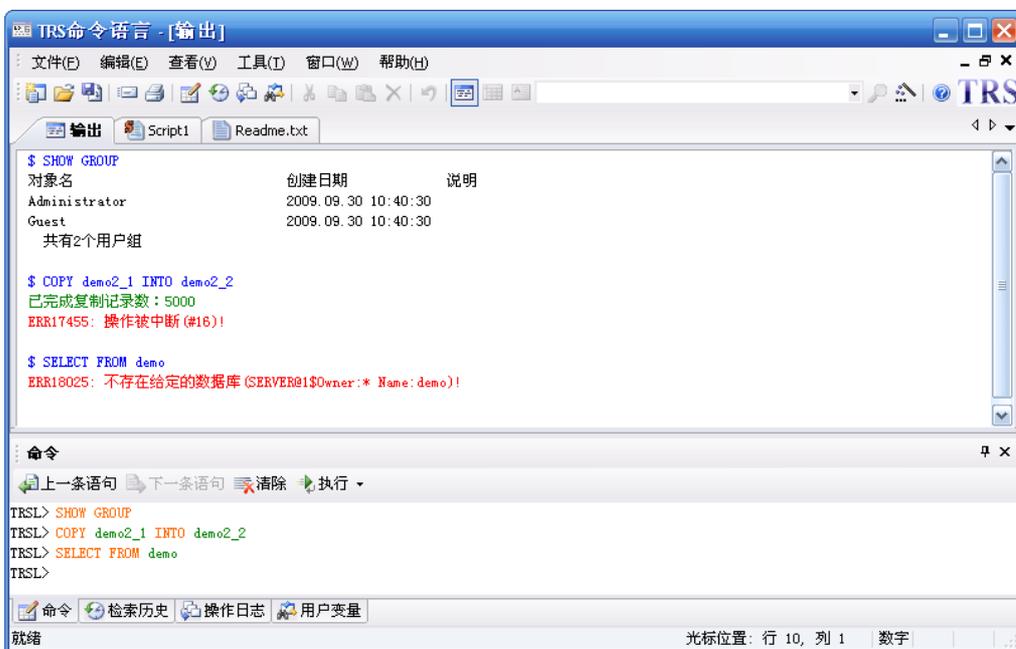


图 1—1 命令语言主界面

1.3 新增特性

在上一版本的基础上，TRSL 命令语言作出了以下几个方面的重大改进：

- ◆ 采用全新的标签页式多文档管理及浮动窗口的主界面
- ◆ 新的系统选项对话框，采用属性表方式显示选项，每一项均有概要提示
- ◆ 语句中不同类型词法符号的彩色显示
- ◆ 语句输入过程提供多种提示功能
- ◆ 修改装入记录等语句的进度显示方式，可实现进度百分数等信息不断刷新显示
- ◆ DROP、UPDATE、DELETE 等语句增加了提示确认的功能，防止误操作
- ◆ 支持打开多个脚本文件进行编辑，同时还提供了若干新功能

1.3.1 采用全新的标签页式多文档管理及浮动窗口的主界面

主界面采用了分割式窗体，将输入的语句命令和输出的界面分开。上方输出窗口为语句执行结果的显示窗口，包括语句执行的一般结果，检索结果的概览和细览都将在这个窗口中显示。除输出窗口外，还以标签页方式排列文件编辑窗口，可打开多个脚本文件和文本文件进行查看编辑。

下方设置了命令窗口、检索历史、操作日志和用户变量四个浮动窗口。用法与 TRS 管理员中的属性窗口和导航窗口一致。

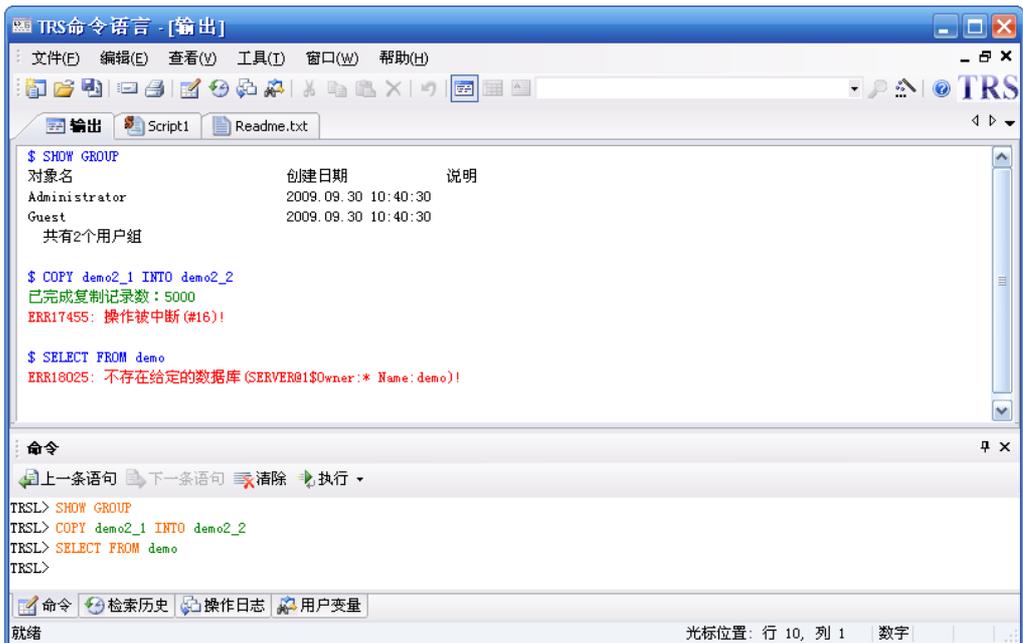


图 1—2 TRS 命令语言的主界面

1.3.2 新的系统选项对话框，采用属性表方式显示选项，每一项均有概要提示

系统选择对话框也采用了新的属性框。单击对话框左部图片框中的图标可分别查看各页面的具体选项的设置。点击每一个属性项时，下方的注释框显示其相应的注释。

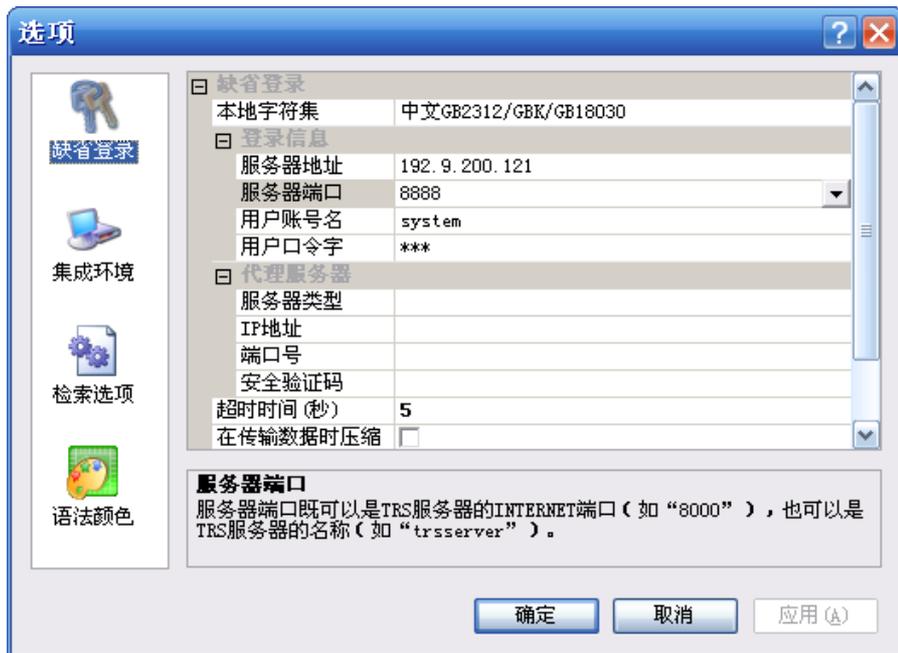


图 1—3 选项对话框

1.3.3 语句中不同类型词法符号的彩色显示

新版本增加了关键字着色功能，各种不同的关键字采用不同的颜色显示，便于区分。每种关键字显示的颜色可以在选项对话框的语法颜色中进行更改。命令窗口和脚本编辑窗口都有此功能。

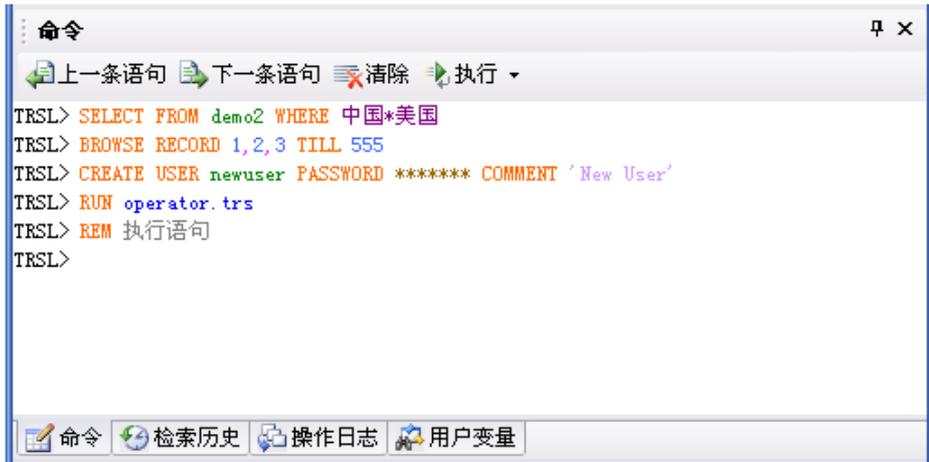


图 1—4 不同类型词法符号的彩色显示

1.3.4 语句输入过程提供多种提示功能

新版本还增加了命令语句的提示功能，使语句的输入更加容易。这些提示功能包括语法提示对话框、语法选项列表对话框和语法选项的语法提示框三种。输入语句的过程中，按左右键(· ·)后，将会弹出语法提示对话框显示该语句的完整语法，当前输入的某个参数将会以粗体显示；当输入到某个参数，并点击空格键以后，将会弹出语法选项列表对话框，提示此参数的可能取值；当语法选项列表中的值被选中，点击上下键(· ·)选择列表中的参数时，在列表框的旁边会出现一个语法提示框，显示此参数的具体语法。

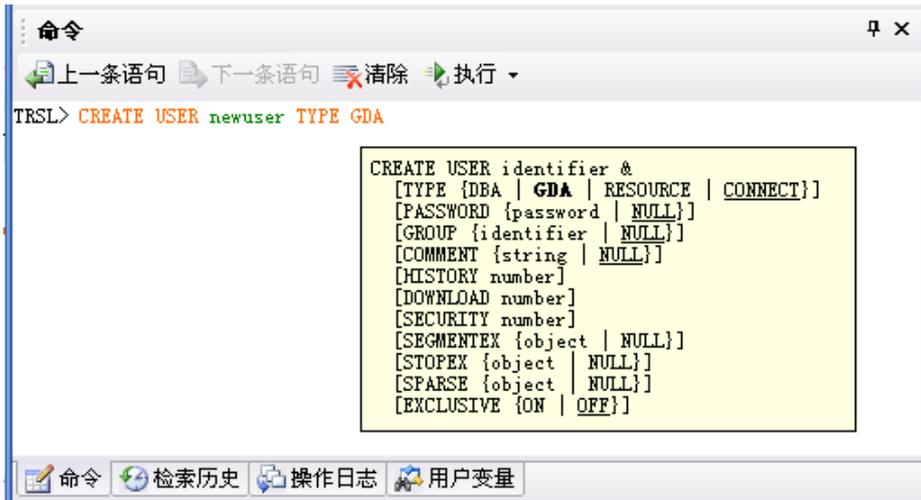


图 1—5 语法提示对话框

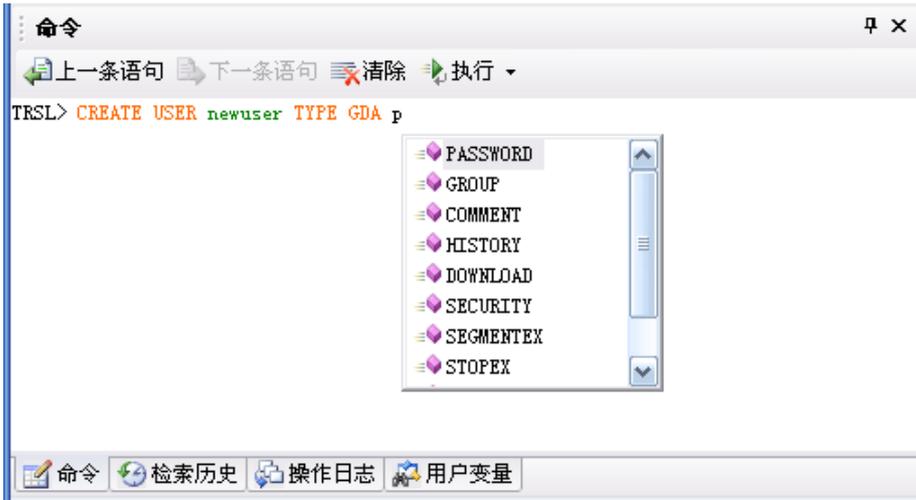


图 1—6 语法选项列表对话框



图 1—7 语法选项的语法提示框

1.3.5 修改装入记录等语句的进度显示方式，可实现进度百分数等信息不断刷新显示

如下图所示，在进行复制优化等耗时操作时，在输出窗口中会显示进度百分数，使操作更加直观。

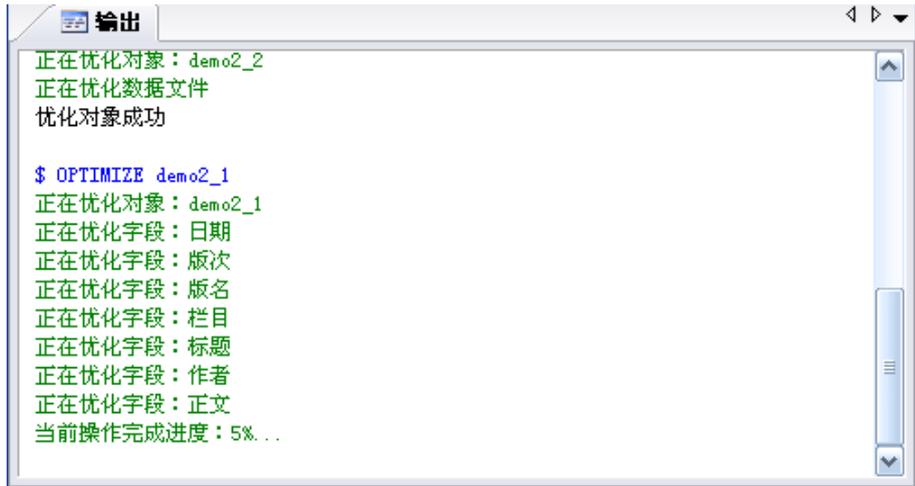


图 1—8 装入记录的进度显示

1.3.6 DROP、UPDATE、DELETE 等语句增加了提示确认的功能，防止误操作

如下图所示，对于 DROP、UPDATE、DELETE 等不可逆的语句，在操作时会提示确认，以防止误操作。



图 1—9 删除操作确认

1.3.7 支持打开多个脚本文件进行编辑，同时还提供了若干新功能

新的 TRS 命令语言以标签页方式排列文件编辑窗口，可打开多个脚本文件和文本文件进行查看编辑。其中脚本文件也具有前面已经提到过的关键字着色和命令语句的提示功能。详细叙述见 2.2.6 节脚本编辑窗口。

脚本运行过程支持 **Ctrl+Break** 中断。在运行比较长的脚本时，增加了中断功能，只要按下 **Ctrl+Break** 键就能中断当前脚本的执行，这样更有利于用户的操作。同时，**SHOW** 语句输出结果列表过程中增加暂停和中断输出功能。

此外还增加了定时执行脚本功能。可以预先设置好需要运行的脚本和时间，当系统时间和某一项设置的运行时间一致时，系统会自动运行设置的脚本文件。

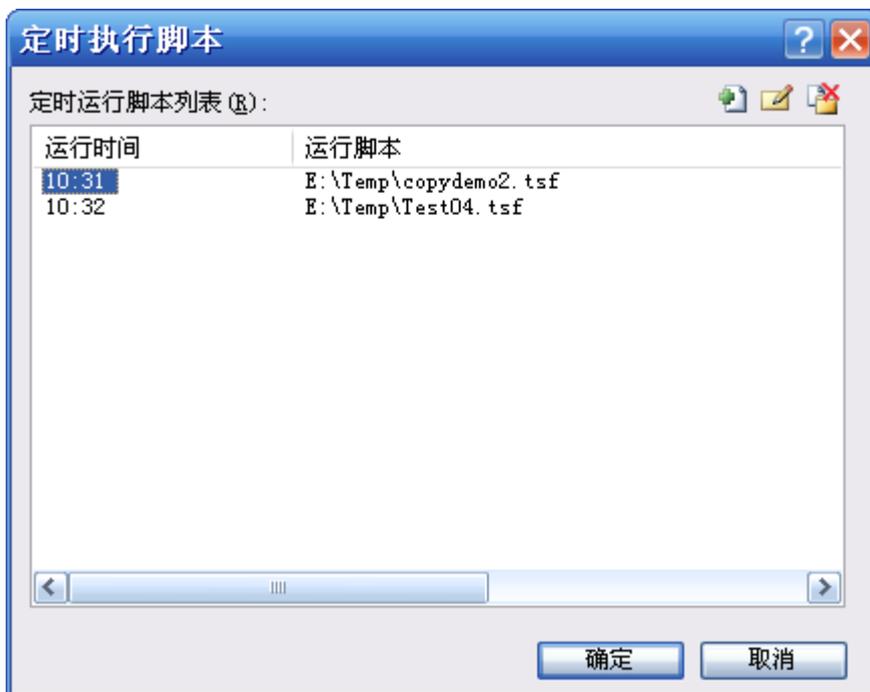


图 1—10 定时运行脚本对话框

1.4 启动和退出

TRS 命令语言运行于中文 Windows XP/2003 或以上版本平台。如要登录到 TRS 服务器，必须保证该服务器的指定端口上安装了 TRS 全文数据库版本，并且该 TRS 服务进程处于启动状态。

在开始菜单中选择“TRS 全文数据库”程序组中的“TRS 命令语言”菜单项，或以资源管理器、命令行或其它方式运行 TRSL.exe 程序，即可启动 TRS 命令语言。单击 TRS 管理员状态栏右下角的按钮或者在 TRS 管理员中选择“查看|命令语言”命令也可打开命令语言工具。如果 TRS 命令语言未自动登录，将弹出如下“登录”对话框：



图 1—11 登录对话框

对话框中各控制的说明如下：

控制名称	功能简介
服务器地址	服务器所在的主机，既可以是 TRS 服务器在网络中的 IP 地址（如“127.0.0.1”），也可以是服务器所在主机的名称（如“ALPHA”）。如果输入的是服务器主机的名称则要求该主机名是当前所使用的网络驱动系统的数据库文件 hosts 的一个入口项（如“127.0.0.1 alpha”）。服务器地址可以从 TRS 的系统管理员处获得。
服务器端口	服务器端口既可以是 TRS 服务器的 Internet 端口（如“8888”），也可以是 TRS 服务器的名称（如“trsserver”）。如果输入的是服务器的名称，则要求该名称是当前所使用的网络驱动系统的数据库文件 services 的一个 TCP 协议入口项（如“trsserver 8888/tcp”）。服务器端口可以从 TRS 的系统管理员处获得。
用户帐号名	用户的帐号名，必须是指定服务器中已经存在的用户名。用户的帐号由 SYSTEM 用户创建。
用户口令字	用户的密码，区分大小写。为防止泄密，输入字符将用“*”号显示。用户的密码可由用户自己修改。
保存密码	是否保存所输入的密码标志，密码会加密保存。

在登录组内依次填入服务器地址、服务器端口、用户帐号名和用户口令字，然后按确定，TRS 命令语言会尝试与指定服务器建立连接。如连接成功，输入的登录信息将会成为下一次运行时的缺省配置。

选择了“文件|退出”命令（或直接用鼠标单击关闭按钮、按下 Alt+F4）后，TRS 命令语言将切断与当前登录的 TRS 服务器的连接，最后关闭主窗口。

1.5 登录和断开

如想登录到其它服务器，或以其它用户的身份重新登录，可选择“工具|登录”命令，TRS 管理员将弹出“登录”对话框。除没有“保存密码”一项以外，其它均和启动时显示的“登录”对话框一致。依次填入服务器地址、服务器端口、用户帐号名和用户口令字，然后按确定，即可与指定服务器建立连接。但输入的登录信息不会成为下一次运行时的缺省配置。

当不再需要在当前登录的服务器上执行 TRS 操作时，选择“工具|断开”命令，系统将弹出一个提问对话框请求确认“是否决定断开连接?”。选择[是(Y)]，将切断与当前 TRS 服务器的连接；选择[否(N)]，将继续保持当前连接。退出登录后，如要再执行 TRS 操作，必须重新登录到服务器。

第2章 界面操作

2.1 编辑和执行命令行

当进入命令语言工具的主界面以后，下方命令窗口将出现 **TRSL>** 提示符，此时可直接输入 **TRSL** 语句，按回车后该语句即可被执行，执行结果上方输出窗口中显示。一行中可同时有多条 **TRSL** 语句，语句之间以半角分号分隔。如果要重复多次执行同一组语句，可把这些语句写到一个脚本文件里，以后用 **RUN** 语句运行该脚本文件即可。

可以用以下编辑键编辑当前命令行（不能编辑当前命令行以外的文本）：

· **移动和选择键：**

- 插入点左移一个字符
- Ctrl+· 插入点左移一个单词
- 插入点右移一个字符
- Ctrl+· 插入点右移一个单词
- Home 插入点移到行首
- End 插入点移到行尾
- Shift+移动键 选择文本

· **行编辑键：**

- Esc 清除整行
- Enter 换行，把当前行加到操作日志，并执行该命令
- Delete 删除插入点后的字符或所选内容
- Ctrl+Delete 删除插入点后的单词
- Back 删除插入点前的字符
- Ctrl+Back 删除插入点前的单词

· **操作日志键：**

- 复制操作日志里的上一条命令
- 复制操作日志里的下一条命令
- PageUp 在命令窗口中表示复制操作日志里向上的第十条命令，在输出窗口中表示屏幕上滚一屏
- PageDown 在命令窗口中表示复制操作日志里向下的第十条命令，在输出窗口中表示屏幕下滚一屏

· **屏幕滚动键：**

- Ctrl+Up 在命令窗口中屏幕上滚一行，但插入点位置不变
- Ctrl+Down 在命令窗口中屏幕下滚一行，但插入点位置不变
- Ctrl+PageUp 在命令窗口中屏幕上滚一屏，但插入点位置不变

Ctrl+PageDown	在命令窗口中屏幕下滚一屏，但插入点位置不变
Ctrl+Home	在命令窗口和输出窗口中屏幕滚动到最顶端，但插入点位置不变
Ctrl+End	在命令窗口和输出窗口中屏幕滚动到最底端，但插入点位置不变

命令语言处理工具可用的快捷键如下：

Ctrl+S	“文件 另存为”命令，把窗口里的文本保存到文件里
Ctrl+M	“文件 发送”命令，把窗口里的文本内容发送给其他用户
Ctrl+P	“文件 打印”命令，直接打印当前窗口里的文本内容
Ctrl+Z	“编辑 撤消”命令，撤消上一次操作
Ctrl+X	“编辑 剪切”命令，删除所选内容并把它放到剪贴板上
Ctrl+C	“编辑 复制”命令，复制所选内容到剪贴板上
Ctrl+V	“编辑 粘贴”命令，插入剪贴板内容
Del	“编辑 删除”命令，删除所选内容
Ctrl+A	“编辑 全部选定”命令，选定所有内容
F1	显示当前所在窗口、菜单、控制的帮助信息
Ctrl+Tab	程序主窗口下方命令、检索历史、操作日志和用户变量这四个浮动窗口相互切换的快捷键；程序主窗口上方输出窗口和多个脚本窗口或者文本窗口相互切换的快捷键
Esc	按照细览视图、概览视图、执行结果视图、命令窗口的顺序，按 Esc 键焦点依次到达
Shift+Tab	程序主窗口上方输出窗口、脚本窗口、文本窗口与下方四个浮动窗口切换的快捷键
Ctrl+Q	输出窗口切换到执行结果视图
Ctrl+W	输出窗口切换到概览视图
Ctrl+E	输出窗口切换到细览视图

执行时当输出的结果多于一屏时，窗口里的内容会自动向上滚动。当窗口里的内容超过一定范围的时候，最早输出的部分会自动从窗口里清除。如要清除窗口里的所有内容，请选择“查看|清屏”菜单或执行 **CLEAR** 语句，此时窗口将回到初始状态，并重新显示提示符。滚动过程中可以按 **Pause** 键暂停执行过程，暂停后按任意键即可继续执行。如果要终止执行过程，可按 **Ctrl+Break** 键。

2.2 各窗口介绍

2.2.1 输出窗口

输出窗口用来显示 TRSL 语句执行的结果，检索结果的概览和细览。因显示的结果分为三类，输出窗口的显示状态也分为三种不同的视图。

- ◆ **执行结果视图**：用于显示 TRSL 语句执行的结果，是输出窗口最常用的状态，也是登录以后输出窗口显示的状态。当下方的命令窗口运行各种指令后，会有

一些运行结果需要显示，这些结果将会显示于执行结果视图中，与下面的用户的指令分开，这样更加直观。



图 2—1 输出窗口的执行结果视图

其中，本次所执行的语句以符号“\$”开头并以蓝色显示，后面即为语句的执行结果。黑色代表一般的信息，绿色代表执行过程的信息，红色代表错误信息。当输出窗口显示的是检索结果的细览或者概览时，点击工具栏上的按钮即可返回执行结果视图。

- ◆ **概览视图：**用于显示检索结果的概览。进行检索后，输入“BROWSE”指令，输出窗口将转换为一个列表框，进行检索结果的概览显示，即概览视图。概览视图里面显示检索结果的概览字段值，与 Admin 中的概览窗口类似。



图 2—2 输出窗口的概览视图

若执行结果视图显示的是检索结果的细览或者语句执行结果时，点击工具栏上的按钮即可返回概览视图。

- ◆ **细览视图：**用于显示检索结果的细览。在显示的概览结果中，双击其中一条记录，窗口即转换为如下图所示的第三种视图，即细览视图。可直接显示记录所

带的图片，其他类型的附件在默认状态下以链接形式显示，单击带下划线的文字以打开或保存该文件或连接。此窗口与 Admin 的细览窗口功能类似。



图 2—3 输出窗口的细览视图

若输出窗口显示的是检索结果的概览或者语句执行结果时, 点击工具栏上的  按钮即可返回细览视图。

2.2.2 命令窗口

命令窗口是一个可输入窗口, 用户在里面输入需要执行的 TRS 命令语句。它和检索历史窗口、操作日志窗口、用户变量窗口以浮动面板的形式出现, 刚登录时都停靠在界面的下方。若命令窗口被关闭或者隐藏时, 点击主工具栏上的  按钮或者点击菜单“查看|其他窗口”下的“命令”菜单项即可打开命令窗口。

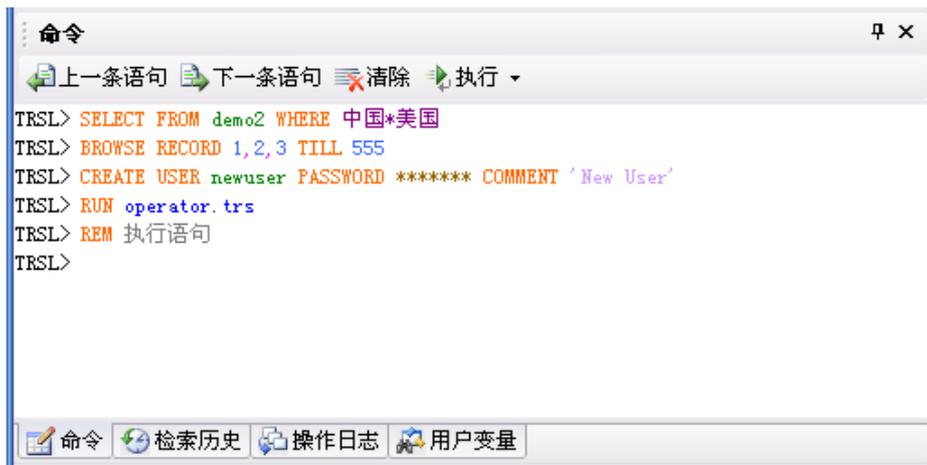
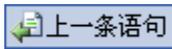
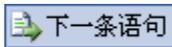


图 2—4 命令窗口

如上图所示，每行的“TRSL> ”后输入的便是需要执行的 TRS 命令语句。新版本增加了关键字着色功能，各种不同的关键字采用不同的颜色显示，便于区分。每种关键字显示的颜色可以在选项的语法颜色属性页中进行更改。点击工具栏上的



按钮将复制操作日志里的上一条命令到命令窗口中，点击



按钮将复制操作日志里的下一条命令到命令窗口中，点击



按钮

清除命令窗口中的所有语句，点击



按钮执行当前输入的语句，点击此按钮

旁边的  可以选择回车键作为换行使用还是作为执行语句的快捷键，此外，Ctrl+Enter 执行语句，Shift+Enter 换行。

另外，新版本还增加了命令语句的提示功能，使语句的输入更加容易。

- ◆ **语法提示对话框：**对整个语句的语法进行提示，当输入一条指令，按左右键(· ·)后，将会弹出一个提示框显示该语句的完整语法，当前输入的某个参数将会以粗体显示。

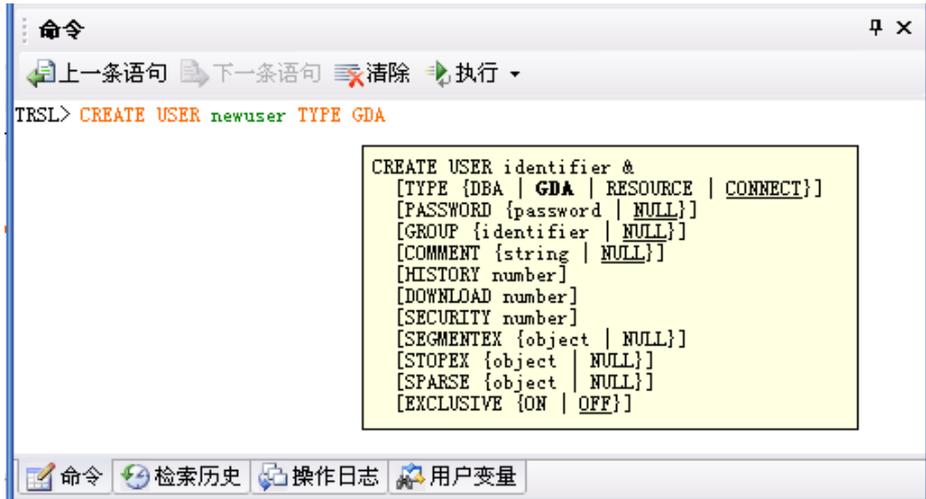


图 2—5 命令窗口的语法提示框

- ◆ **语法选项列表对话框:** 当输入到某个参数，并点击空格键以后，将会弹出一个提示框，提示此参数的可能取值。用户可以对照提示框直接输入可能的取值，当输入的值与列表中的值有匹配时，列表中的值会被选中，此时点击上下键(↕)可以上下选择列表中的参数，点击空格或者回车键后，被选中的参数将自动写入语句后面，列表框消失。同时也可以直接在提示框中双击所需要的值，点击的值将会自动写入语句后面，列表框消失。

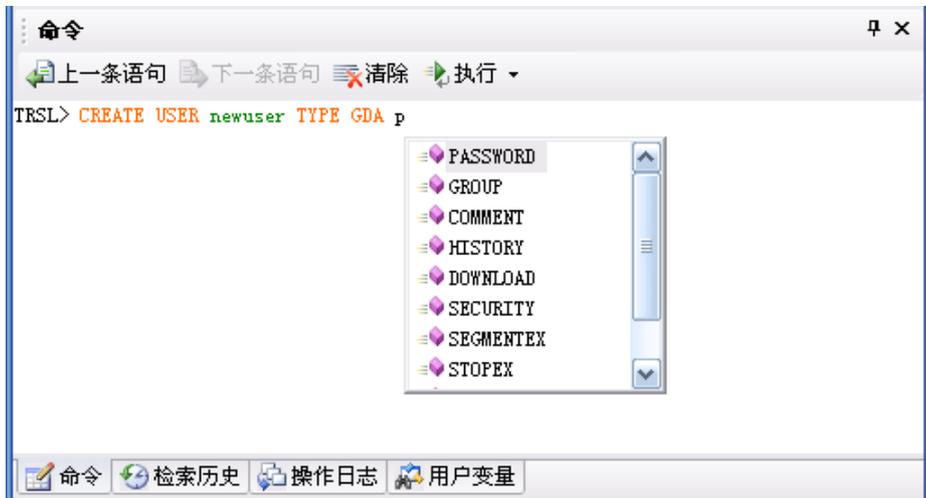


图 2—6 命令窗口的语法选项列表对话框

- ◆ **语法选项的语法提示框:** 当语法选项列表中的值被选中，点击上下键(↑↓)选择列表中的参数时，在列表框的旁边会出现一个语法提示框，显示此参数的具

体语法，便于选择时查看。



图 2—7 命令窗口的语法选项的语法提示框

2.2.3 检索历史窗口

检索历史窗口用来显示最近执行过的检索的历史表，为只读窗口。列表中列出里最近每次检索的表达式、数据库/视图名、命中记录数、命中点数等信息。若检索历史窗口被关闭或者隐藏时，点击主工具栏上的按钮或者点击菜单“查看|其他窗口”下的“检索历史”命令即可打开检索历史窗口。



图 2—8 检索历史窗口

点击按钮可清除所有的检索历史；双击日志中的某一系列项或者点击按钮可把当前选中的检索表达式复制到命令窗口中的插入点位置。

2.2.4 操作日志窗口

操作日志窗口用来查看最近执行过的命令行及执行结果，列表中列出里最近执行过的每条命令及执行的时间和结果。若操作日志窗口被关闭或者隐藏时，点击主工具栏上的按钮或者点击菜单“查看|其他窗口”下的“操作日志”命令即可打开操作日志窗口。



图 2—9 操作日志窗口

如图所示，点击按钮可清除所有的操作日志；双击日志中的某一列表项或者点击按钮可把当前选中的命令复制到命令窗口的插入点位置，以便于执行。

2.2.5 用户变量窗口

用户变量窗口用来显示 TRSL 中定义的各个用户变量的名称及对应的变量值，并可对变量进行增加、删除或者修改。若用户变量窗口被关闭或者隐藏时，点击主工具栏上的按钮或者点击菜单“查看|其他窗口”下的“用户变量”命令即可打开用户变量窗口。

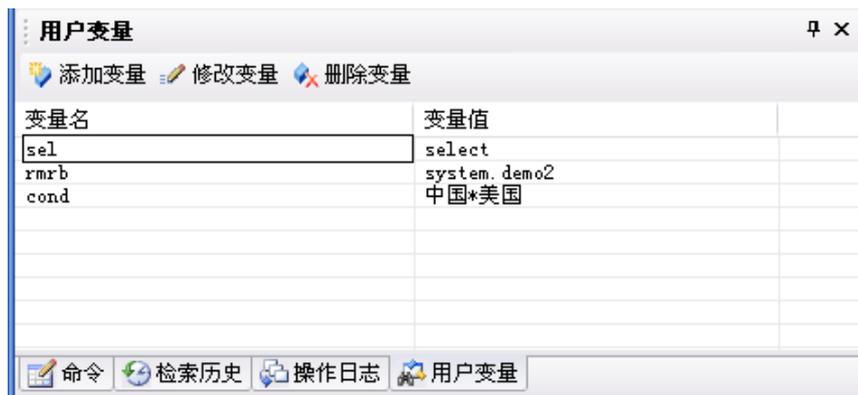


图 2—10 用户变量窗口

如上图所示，可以直接点击列表，在出现的编辑框里面进行变量名和变量值的编辑修改，编辑完成以后回车即可保存。点击 **添加变量** 按钮，将增加一行列表项，在对应的编辑框里面输入新增的变量名和变量值即可，点击 **修改变量** 按钮对已存在的变量进行修改，点击 **删除变量** 按钮删除当前变量。

2.2.6 脚本编辑窗口

脚本编辑窗口用于创建编辑文本文件或者脚本文件。它的用法与 Windows 的记事本类似，功能有所增强，并能编辑大于 64K 的文件。以下是打开了一个“.tsf”和一个“.txt”文件的编辑窗口：

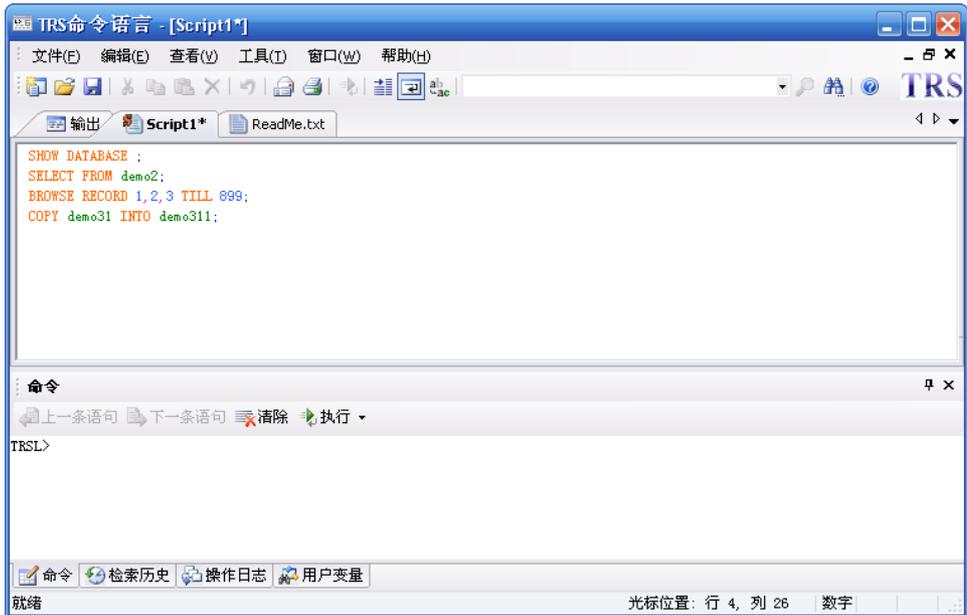


图 2—11 脚本编辑窗口

如图所示，脚本编辑窗口打开时以标签页的形式出现，可以同时打开多个脚本文件和文本文件。

脚本文件是包含许多 TRS 命令语句的文本文件，以便于在执行语句“RUN”时，可以一次性执行多条语句。新版本中，在编辑脚本文件时，增加了关键字着色功能，各种不同的关键字采用不同的颜色显示，便于区分。每种关键字显示的颜色可以在选项的语法颜色属性项中进行更改。此外还增加了命令语句的提示功能，使语句的输入更加容易。

- ◆ **语法提示对话框：**对整个语句的语法进行提示，当输入一条指令，按左右键(••)后，将会弹出一个提示框显示该语句的完整语法，当前输入的某个参数将会以粗体显示。

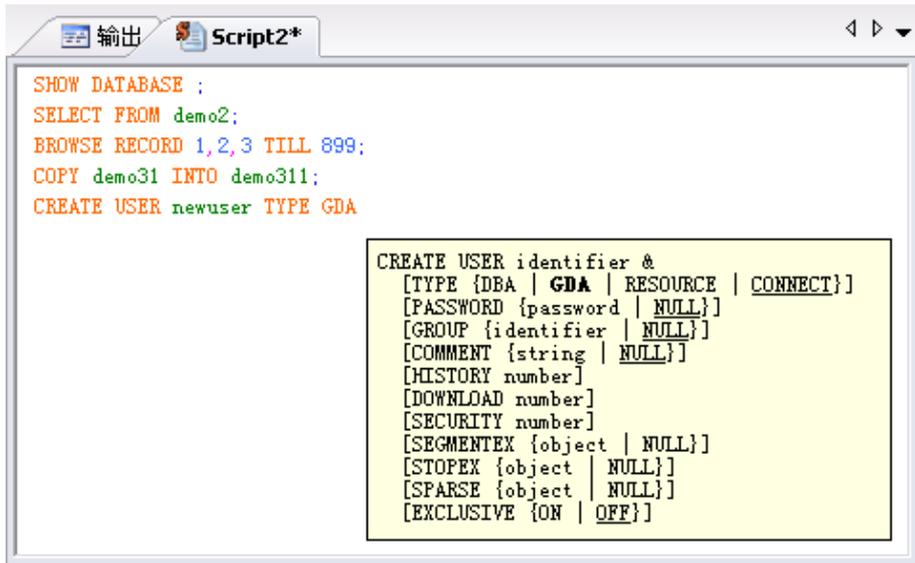


图 2—12 脚本编辑窗口中的语法提示框

- ◆ **语法选项列表对话框：**当输入到某个参数，并点击空格键以后，将会弹出一个提示框，提示此参数的可能取值。用户可以对照提示框直接输入可能的取值，当输入的值与列表中的值有匹配时，列表中的值会被选中，此时点击上下键(••)可以上下选择列表中的参数，点击空格或者回车键，被选中的参数将自动写入语句后面，列表框消失。同时也可以直接在提示框中双击所需要的值，点击的值将会自动写入语句后面，列表框消失。

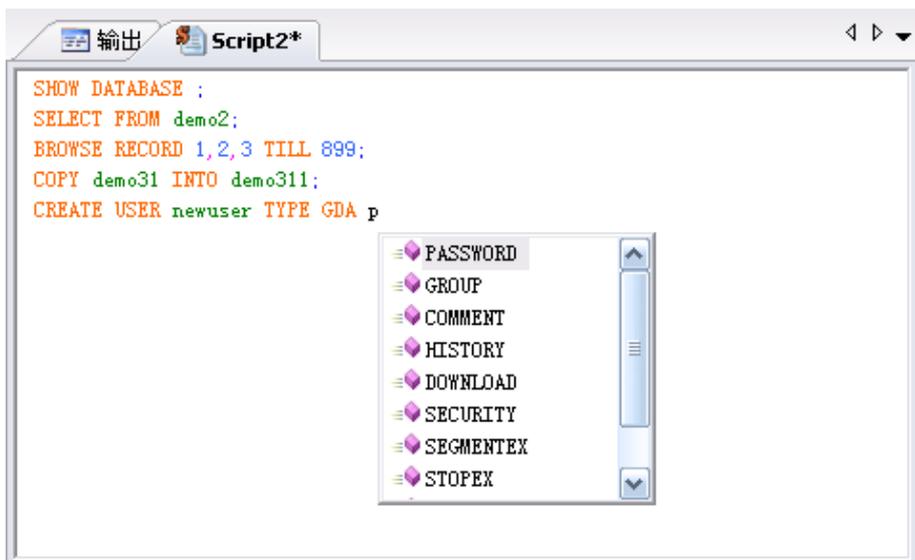


图 2—13 脚本编辑窗口中的语法选项列表对话框

- ◆ **语法选项的语法提示框：**当语法选项列表中的值被选中，点击上下键(••)选择列表中的参数时，在列表框的旁边会出现一个语法提示框，显示此参数的具体语法，便于选择时查看。

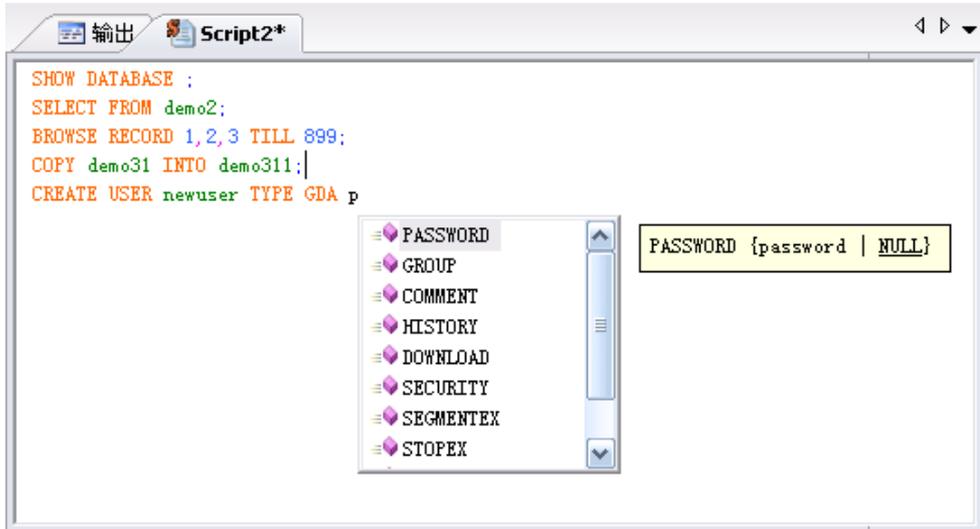


图 2—14 脚本编辑窗口中的语法选项的语法提示框

2.3 定制界面

2.3.1 输出窗口的显示状态

输出窗口因显示的结果分为三类，显示状态也分为三种不同的视图，即执行结果视图，细览视图和概览视图。在工具栏上分别用以  这三个图标表示，同时对应菜单中的“查看|执行结果视图”、“查看|概览视图”和“查看|细览视图”命令。输出窗口显示相应视图时，工具栏上其对应的图标即为选中状态。

当进行检索并输入“BROWSE”指令后，概览视图和细览视图激活，这时可以点击以上三个图标进行三种视图的转换，以便于不同结果的查看。

2.3.2 清除屏幕

当窗口中输出的内容太多的时候，可以选择“查看|清除”命令清除窗口中的所有内容，并重新显示命令行提示符。在命令行中执行 CLEAR 语句也能起到同样作用。

2.3.3 修改显示字体

如想改变窗口显示的字体和颜色，可选择“查看|字体”命令，TRS 命令语言将弹出如下对话框：



图 2—15 修改字体对话框

对话框中各控制的说明如下：

控制名称	功能简介
字体	列出和选择可用的字体，只列出等宽的字体。字体名前有  标志的为 TrueType 字体。
字形	列出和选择指定字体可用的字形。
大小	列出和选择指定字体可用的字号。
效果	指定显示文字的颜色和风格。
加删除线	选中表示在字体的中间加删除线。
加下划线	选中表示在字体的下面加下划线。
颜色	选择字体的显示颜色。
示例	显示所选字体的范例。
字符集	选择显示字体的语言。

选择好显示字体及其颜色、效果、语言等属性以后，按[确定]，将关闭对话框并以新字体显示整个命令语言窗口的内容，所设字体将保存到注册表里，对以后启动的 TRS 命令语言同样有效。按[取消]则关闭对话框但不改变显示字体。

2.3.4 设置窗口背景

缺省情况下，窗口的背景为白色。如要改变窗口的背景颜色，可选择“查看|背景颜色”命令，TRS 命令语言弹出如下对话框：



图 2—16 设置背景颜色对话框

对话框中各控制的说明如下：

控制名称	功能简介
基本颜色	显示可用的基本颜色，单击可选择为背景颜色。
自定义颜色	显示已定义的自定义颜色，单击可选择为背景颜色。如要增加自定义颜色，请按[规定自定义颜色]按钮，并在颜色阵列中选择需要的颜色，然后按[添加到自定义颜色]按钮。

选择好字体颜色以后按[确定]，将关闭对话框并以新颜色作为命令语言窗口的背景，所设颜色将保存到注册表里，并在以后启动的 TRS 命令语言中同样有效。按[取消]则关闭对话框但不改变背景颜色。

2.3.5 自动折行

如果希望窗口内容超过一行所能容纳的长度能自动折行，可选中“查看|自动折行”选项。所设自动折行标志将保存到注册表里，并对以后启动的 TRS 命令语言同样有效。

2.4 使用辅助工具

2.4.1 定时运行脚本

设置定时运行脚本的时间。

选择“工具|定时运行脚本”命令后，TRS 命令语言弹出如下对话框：

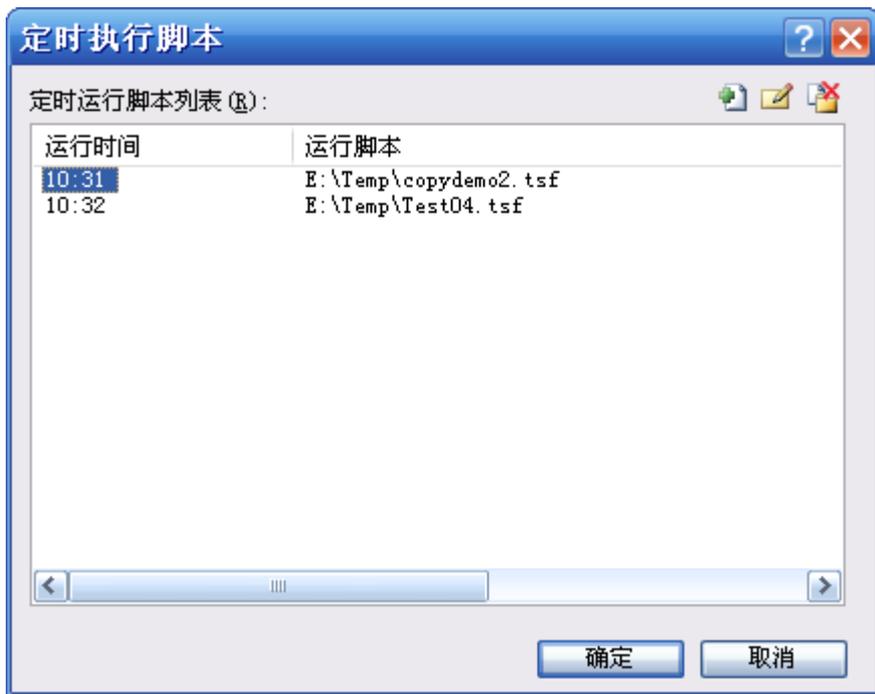


图 2—17 定时运行脚本对话框

点击  添加一项新的定时运行时间和脚本路径，点击  修改当前选中项的运行时间和脚本路径，点击  删除当前选中项。运行列表编辑完成后点击[确定]将保存列表中的各项，并保存到注册表，当前系统时间和某一项设置的运行时间一致时，系统会自动运行设置的脚本文件。

2.4.2 表达式生成器

命令语言工具提供了一个表达式生成器，用于帮助用户生成检索表达式，对初学者尤为有用。选择“工具|表达式生成器”命令后，TRS 命令语言将弹出如下对话框：

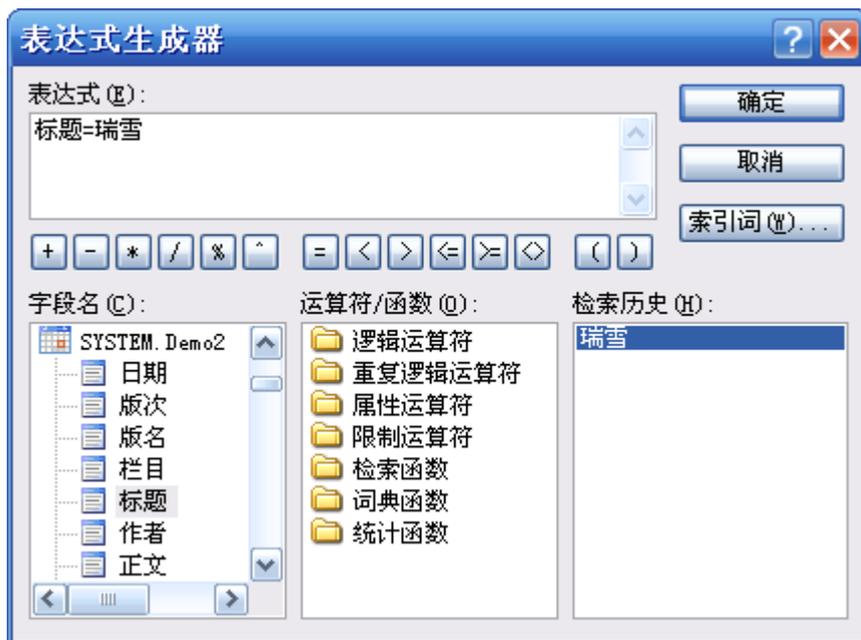


图 2—18 表达式生成器

对话框中各控制的说明如下：

控制名称	功能简介
表达式	生成的检索表达式。
运算符按钮	从 到 为常用的运算符按钮，按下按钮会把相应运算符复制到表达式编辑框里。
字段名	所有数据库/视图的字段列表，双击数据库/视图名可显示出该对象的所有字段名，双击字段名可把字段名复制到表达式编辑框里。
运算符/函数	TRS 的运算符和函数列表。运算符分为算术运算符、比较运算符、逻辑运算符、重复逻辑运算符、属性运算符、限制运算符和加权运算符，函数分为统计函数、检索函数和词典函数。双击文件夹可显示该项包含的所有运算符/函数列表，再双击某一项则可把该运算符/函数复制到表达式编辑框里。
检索历史	检索历史列表，列出最近用到的 20 个检索表达式，双击可把检索历史项复制到表达式编辑框里。
索引词	打开“查询索引词”对话框，对数据库或者视图的索引词进行查询。

生成合适的检索表达式后，按[确定]，将关闭“表达式生成器”对话框并把生成的检索表达式复制到命令行的当前位置。按[取消]则关闭对话框。

在“表达式生成器”对话框里，单击窗口里的[索引词]按钮将出现如下对话框，用于查询数据库或者视图里的索引词：



图 2—19 查询索引词对话框

该对话框中各控制的说明如下：

控制名称	功能简介
查询条件	查询表达式。可为“目标词”或“函数名(目标词, ...)”形式，其中“目标词”既可是个确定的词汇或字段值，也可是一个符合 TRS 规范的模糊词汇或模糊字段值。按右边的 按钮进行索引词查询，并把结果显示在结果列表里。
显示个数	显示结果的个数，缺省为不设置。
当前页面	当前显示的页数。在编辑框中输入想要跳转的页数以后点击回车键，将会跳转到指定的页数。
结果列表	查询结果的索引词列表，每页显示个数可在“选项[集成环境]”里设定，缺省为 100 个。每个索引词显示其词语、命中数、词频数和它相对于主题词的深度。要看其它页的索引词信息，可用 和 按钮进行前后翻页。
对象名	待查询的目标数据库名或者视图名。改变查询对象将重新检索索引词并刷新检索列表。
字段名	待查询的字段名，为空表示在缺省全文字段上查询。改变查询字段将重新检索索引词并刷新检索列表。
复制	把结果列表里当前选中的检索词复制到“表达式生成器”对话框的检索条件编辑框里。
关闭	关闭“查询索引词”对话框。

查询完毕后，按[复制]或直接双击结果列表的索引词项将把指定索引词复制到“表达式生成器”对话框的检索条件编辑框里。按[关闭]则关闭对话框。

2.4.3 执行选定脚本

执行脚本编辑窗口被选中的语句。

当脚本编辑窗口中有语句被选中时，点击“查看|执行选定脚本”菜单项或者点击工具栏上的按钮，TRS 命令语言会将所选中的语句复制到命令窗口中，并执行。

2.4.4 制表位

修改编辑控制的制表符字符数。

在脚本编辑窗口中，点击“查看|制表位”菜单项后，在弹出的窗口中输入在 1-256 之间的某一个数，缺省值为 8。所设制表位字符数将对以后打开的所有脚本编辑窗口有效。

2.4.5 查找

在当前显示文本中查找指定的字符串。在脚本编辑窗口中，选择“查看|查找”命令后，TRS 命令语言弹出如下对话框：



图 2—20 查找对话框

对话框中各控制的说明如下：

控制名称	功能简介
查找目标	要查找的文字。
全字匹配	选中表示只查找整个单词，而不查找包含此单词的长字符串。
区分大小写	选中表示只查找与输入文字大小写一致的字符串。

输入要查找的文字，设置好查找选项，按[查找下一个]，TRS 命令语言将从脚本编辑窗口的当前位置开始查找键入文字下一次出现的位置。由于查找对话框是一个无模式对话框，所以可以执行多次查找而不关闭对话框，并可切换到脚本编辑窗口进行文本编辑。如要关闭对话框，请按[取消]。

2.5 处理执行结果

2.5.1 保存执行结果

把输出窗口里的文本保存到文件里或者把当前检索结果的全部或部分保存为文本文件。

因输出窗口显示的结果分为三类，所以选择“文件|另存为”命令后，TRS 命令语言弹出的对话框也分为两种。

- ◆ **保存输出窗口的文本：**当输出窗口显示的是语句执行结果时，TRSL 弹出的对话框如下图所示：

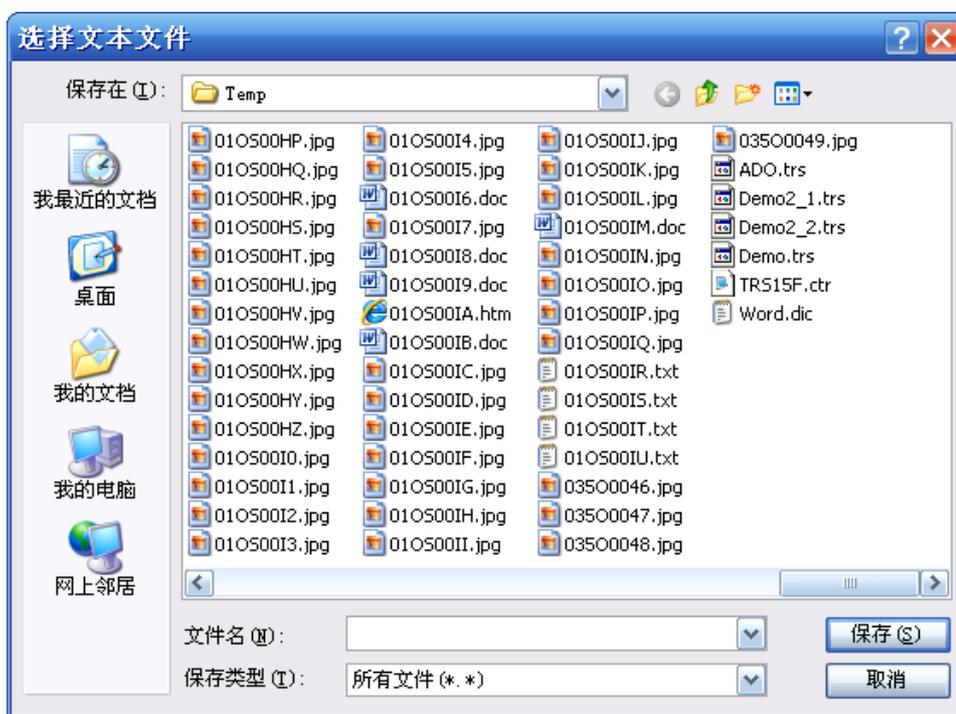


图 2—21 选择保存文件路径对话框

在保存在组合框里选择保存文件的路径，然后在文件列表框里选择文件或直接在文件名编辑框里输入文件名，最后按[保存]按钮即可把输出窗口里的文本保存到文件里，供以后查看或编辑。

- ◆ **保存当前检索结果：**当输出窗口显示的是检索结果的概览或者细览时，TRSL 弹出的对话框如下图所示：



图 2—22 保存检索结果对话框

对话框中各控制的说明如下：

控制名称		功能简介
目标文件名		保存的目标文件名，可按右边的  按钮来选择目标文件名。
字段列表		需要保存的字段列表。可按右边的  按钮来选择字段列表。
记录范围	当前记录	只保存当前记录。
	所有记录	保存检索结果里的所有记录。为缺省选项。
	指定范围	保存指定范围的记录，可用“起始记录号-终止记录号”的方式表示一定范围的记录，或仅用单个数字表示一条记录，多个分段之间以“,”分隔。
	每文件存放条数	设置每个文件存放记录的条数，清空或为零表示不限制。
保存格式	保存格式	结果保存的格式，对索引词典无效，缺省为TRS装库数据格式。
	记录文件	以记录数据格式保存检索结果，一般只需浏览检索结果时选择此格式。
	装库文件	以TRS装库数据格式保存检索结果，所存文件可以用来重新进行装库。
	ISO2709文件	以ISO2709国际标准格式保存检索结果。
	XML文件	以XML格式保存检索结果。

	传输装库文件	以传输格式保存检索结果。传输格式是数据加载时用于传输一批记录的内部格式文件。这种格式的一个最大优点是能够将文本与非文本数据打包在一个文件里，避免大量外挂文件可能引起的性能问题。
	保存记录ID	保存记录时保存记录的ID。
	添加记录前缀	记录前加引导前缀\$\$TRSTAG\$\$，可与以上格式组合，但只对装库数据格式和镜像装库格式有效。
文件选项	保存附件方式	保存记录里包含的源文件的模式。
	附件存放目录	指定记录所含附件存放的目录。
	每组容纳记录	指记录附件存放的目录中，每组容纳多少条记录的附件。若为0或者清空表示不限制，即所有附件全部直接存入指定的文件中，若指定了数值比如为10，则生成若干个文件分别存放附件，每个文件中存储10条记录的附件。
	最大文件长度	设置保存的目标文件的最大长度，清空或为零时取缺省值2G。若当前所要保存的记录大小超过此数值，将自动分为N个大小为此数值的目标文件进行保存，如Demo2_1.tris、Demo2_2.tris...

输入目标文件名，设置保存字段、记录范围、保存格式等选项以后，按[确定]，TRSL 将开始保存检索结果到目标文件里。按[取消]则放弃保存操作并关闭对话框。由于保存检索结果是一个耗时过程，因此保存过程中有“保存进度”对话框显示目前已保存的记录数和保存完成的百分比。如要终止保存过程，可按[终止]按钮。

2.5.2 发送执行结果

如要把命令语言窗口里的内容发送到其他用户，可选择“文件|发送”命令，或按工具栏上的按钮，TRSL 命令语言将调用相应的邮件处理软件（如 Microsoft Outlook Express 等），用户可以利用该软件把命令语言窗口里的内容发送到其他用户手里。

2.5.3 打印执行结果

选择“文件|打印”命令，或按工具栏上的按钮，可打印命令语言窗口里的文本。TRSL 命令语言将弹出如下对话框：



图 2—23 打印选项对话框

在对话框中选择目标打印机，输入打印份数后，按[确定]键后，TRS 命令语言将开始打印命令语言窗口里的内容，按[取消]则放弃打印操作。

打印纸张及打印的页面参数由“文件|页面设置”命令设置，对话框如下：

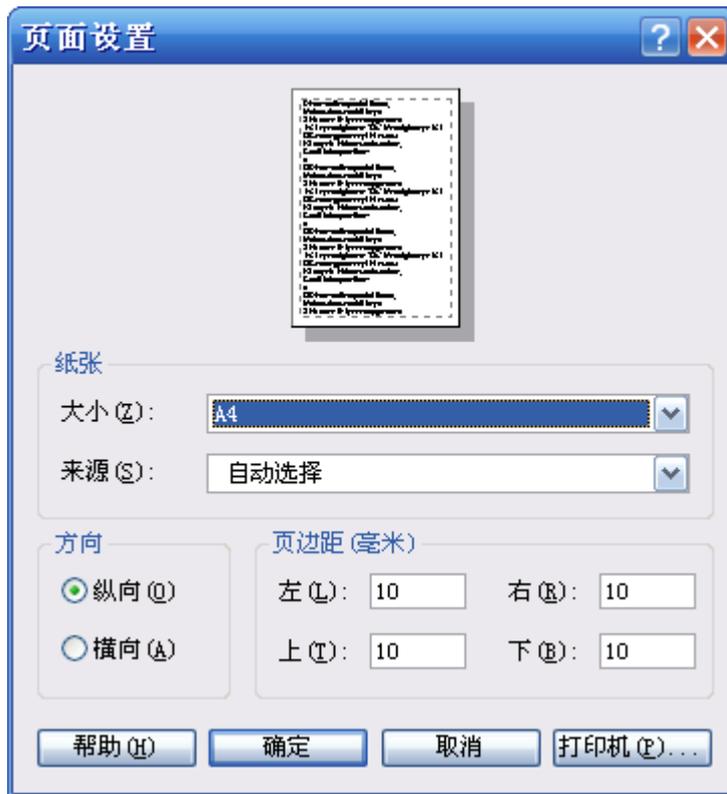


图 2—24 页面设置对话框

对话框中各控制的说明如下：

控制名称	功能简介
纸张	指定打印纸张的大小和来源。
大小	指定要使用的纸张或信封大小。
来源	指定纸张在打印机上的位置。不同打印机有不同的纸张来源，如上层纸盒、信封送纸器和手动送纸器等。
方向	显示整个文档在整个页面上的位置。
纵向	纵向打印文件。
横向	横向打印文件。
页边距	设置页面的打印区域。此处设置的边距不在打印范围内。
左	设置页面的左边距，缺省为 10mm。
右	设置页面的右边距，缺省为 10mm。
上	设置页面上边距，缺省为 10mm。
下	设置页面下边距，缺省为 10mm。

每改变一个选项，对话框上部的预览画面都会相应地作出调整。按[确定]，所作设置将作为打印时的页面设置；按[取消]将放弃所作的设置并关闭对话框；按[打

印机]将选择打印的目标打印机。

2.6 修改设置参数

TRS 命令语言有一些参数可由用户自己设置。选择“工具|选项”命令后，TRS 命令语言弹出“选项”对话框。该对话框共有缺省登录、集成环境和用户变量三页。单击对话框上部属性页的标签可分别查看各页面的具体选项的设置。如对选项作了修改，按[确定]按钮可保存所作的修改并关闭对话框，按[应用]按钮可保存所作的修改而不关闭对话框，按[取消]按钮则放弃所作的修改并关闭对话框。

2.6.1 缺省登录

“缺省登录”属性页如下所示：

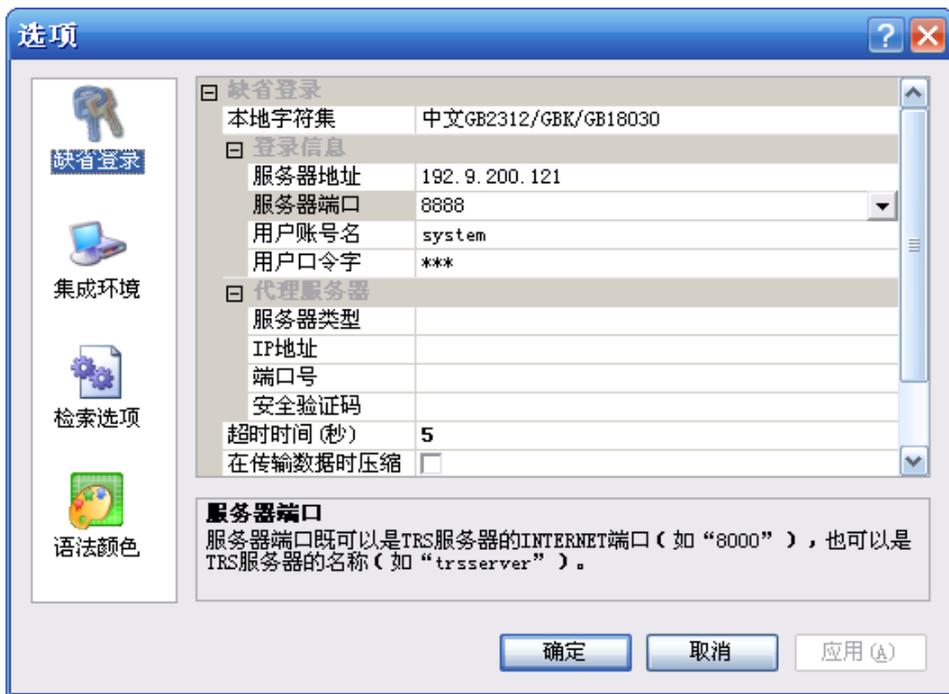


图 2—25 选项的缺省登录属性页

属性页中各选项的含义如下：

控制名称	功能简介
本地字符集	本地所用的字符集，可为“中文 GB2312/GBK/GB18030”、“中文 BIG5”、“统一码 UTF8”或“不设置（与原来兼容）”，缺省为第一个选项。
登录信息	启动时缺省的登录信息，包括服务器地址和端口、用户帐号和口令等。输入的登录信息将会保存到注册表中，作为下一次启动的缺省设置。

服务器地址	服务器所在的主机，既可以是 TRS 服务器在网络中的 IP 地址（如“199.9.200.80”），也可以是服务器所在主机的名称（如“ALPHA”）。如果输入的是服务器主机的名称则要求该主机名是当前所使用的网络驱动系统的数据库文件 hosts 的一个入口项（如“192.9.200.80 alpha”）。服务器地址可以从 TRS 的系统管理员处获得。
服务器端口	服务器端口既可以是 TRS 服务器的 Internet 端口（如“8000”），也可以是 TRS 服务器的名称（如“trsserver”）。如果输入的是服务器的名称，则要求该名称是当前所使用的网络驱动系统的数据库文件 services 的一个 TCP 协议入口项（如“trsserver 8000/tcp”）。服务器端口可以从 TRS 的系统管理员处获得。
用户帐号名	用户的帐号名，必须是指定服务器中已经存在的用户名。用户的帐号由 SYSTEM 用户创建。
用户口令字	用户的密码，区分大小写。为防止泄密，输入字符将用‘*’号显示。用户的密码可由用户自己修改。
代理服务器	代理服务器的相关选项。
服务器类型	不为空表示将通过代理连接到目的服务器，目前仅支持“TRS”的代理服务器。
IP 地址	如果服务器类型不为空，则此属性必须指定代理服务器的有效 IP 地址。
端口号	如果服务器类型不为空，则此属性必须指定代理服务器的有效端口号。
安全验证码	如果服务器类型不为空，则此属性可以指定一个代理服务器所需要的安全验证码。
超时时间	设置登录时的超时时间，以秒为单位，超过指定时间仍未连接上则报错。为零则超时时间为网络系统的缺省值，一般为 75 秒到几分钟不等。
在传输数据时压缩	选中表示在传输数据时进行压缩，压缩算法由内部确定，对于本地客户端无效。
在传输数据时加密	选中表示在传输数据时进行加密，加密算法由内部确定，对于本地客户端无效。
自动登录	启动 TRS 命令语言时是否自动用缺省登录信息进行登录选项，缺省为设置。

2.6.2 集成环境

“集成环境”属性页如下所示：

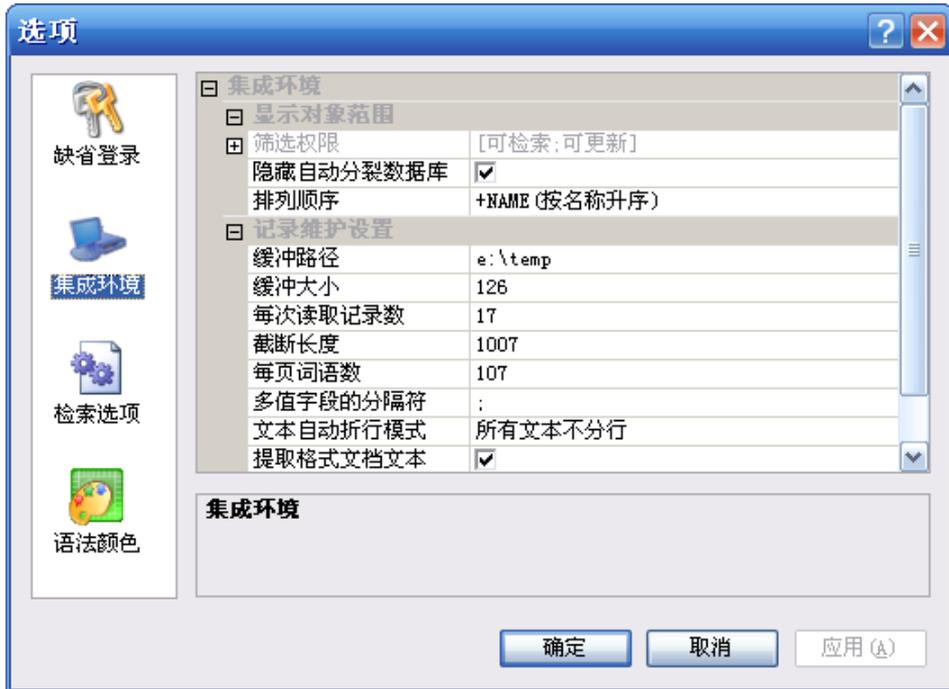


图 2—26 选项的集成环境属性页

属性页中各选项的含义如下：

控制名称	功能简介
显示对象范围	当对象个数太多，不便于查看时，可设置对象的显示范围。
筛选条件	显示对象的筛选条件，可使用通配符：用'?'匹配单个字符，用'*'匹配任何字符。缺省为“*.*”，表示所有对象。
筛选权限	只显示具有选定权限的对象。
可检索的	如选中表示只显示可以检索的对象，对仅对数据库、视图和词典有效。
可更新的	如选中表示只显示可以更新的对象，对仅对数据库和词典有效。
可索引的	如选中表示只显示可以建立索引的对象，对仅对数据库有效。
可修改的	如选中表示只显示可以修改结构的对象，对仅对数据库和视图有效。
可删除的	如选中表示只显示可以删除的对象，对仅对数据库、视图和词典有效。
隐藏自动分裂数据库	如选中表示不显示自动分裂视图对应的物理数据库。
排列顺序	显示对象的排列顺序。
记录维护设置	检索结果浏览/维护窗口的选项设置。
缓冲路径	存放检索结果源文件的临时路径，按右边的  按钮可选择缓冲文件存放的目录，缺省为 TRS 管理员安装路径下的 Temp 子目录。
缓冲大小	检索结果缓冲区大小。缓冲区大小以记录为单位，缺省为 128。
每次读取记录数	读取检索结果时每次读取的记录数，仅对取概览内容时有效，缺省为 16。
截断长度	PHRASE、DOCUMENT 和 BIT 字段的截断长度，为零表示不进行截

	断。缺省为 1024。
每页词语数	查询数据库的索引词和词典的时候，每页显示的词语数，缺省为 100。
多值字段的分隔符	多值字段的分隔符，仅对 DATE、NUMBER 和 CHAR 类型字段有效。为空串表示取系统缺省值‘;’。
文本自动折行模式	只对 DOCUMENT 和 PHRASE 字段有效，其中“所有文本不分行”选项一般用于细览字段以便于控制其显示格式，“段落文本不分行”一般用于细览字段以便于自动换行。
提取格式文档文本	只对 DOCUMENT 字段有效，表示只存储了单个格式化文档字段值的字段，尽量提取文本并给出命中点。
对 URL 输出对应文件	选中表示对 URL(统一资源定位)输出其对应的文件(如 HTML 等)，而不是输出 URL 本身。缺省值为清除。
命令执行	命令执行的相关选项。
重要操作提示确认	选中表示在执行 DROP、UNLOCK、RECOVER、UPDATE、MODIFY 和 DELETE 这些操作前提示确认，缺省值为选中。

2.6.3 检索选项

“检索选项”属性页如下所示：

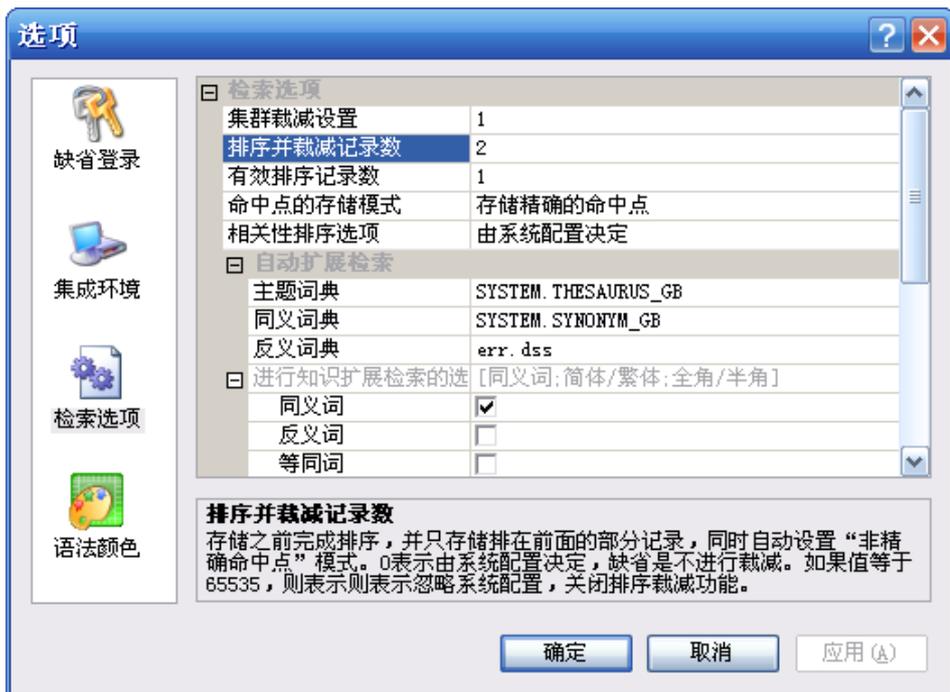


图 2—27 选项的检索选项属性页

属性页中各选项的含义如下：

控制名称	功能简介
集群裁减设置	集群对后台检索结果的裁减设置。0 表示不作限制；1~9 表示集群至少需要等待相应数量的后台返回结果；10~99 表示集群至少需要等待相应比例的后台返回结果；100 及以上表示至少需要返回的结果记录数。

排序并裁减记录数	存储之前完成排序，并只存储排在前面的部分记录，同时自动设置“非精确命中点”模式。0 表示由系统配置决定，缺省是不进行裁减。如果值等于 65535，则表示则表示忽略系统配置，关闭排序裁减功能。
有效排序记录数	即部分排序，只保证前一部分记录是有序的。0 表示由系统配置决定，缺省是全部排序。如果值等于 65535，则表示则表示忽略系统配置，关闭部分排序功能。只有在排序裁减记录数为 0 的情况下才可能有效。
命中点的存储模式	不同的命中点存储模式，不但影响检索速度，还会影响相关性排序结果。
相关性排序选项	不同的相关性的计算模式和 IDF 模式组合下会有不同的排序结果。分为 7 种情况：由系统配置决定；忽略词的位置和使用基于单个数据库范围的 IDF 值；忽略词的位置和词 IDF 值不参与相关性计算；考虑词的位置和使用基于单个数据库范围的 IDF 值；考虑词的位置和词 IDF 值不参与相关性计算；用一个记录中命中词的单位向量长度（每个命中词只计算一次）作为记录的相关；用一个记录中命中词的单位向量长度，以及命中词的词频和作为记录的相关度。
自动扩展检索	自动扩展检索的相关选项。
主题词典	利用知识辅助词典自动进行知识扩展检索时的主题词典。
同义词典	利用知识辅助词典自动进行知识扩展检索时的同义词典。
反义词典	利用知识辅助词典自动进行知识扩展检索时的反义词典。
进行知识扩展检索的选项	进行知识扩展检索的各个选项。
同义词	选中表示自动进行“同义词”扩展检索。
反义词	选中表示自动进行“反义词”扩展检索。
等同词	选中表示自动进行“等同词”扩展检索。
替代词	选中表示自动进行“替代词”扩展检索。
缩略词	选中表示自动进行“缩略词”扩展检索。
被缩略词	选中表示自动进行“被缩略词”扩展检索。
相关词	选中表示自动进行“相关词”扩展检索。
相关词	选中表示自动进行“相关词”扩展检索。
上位词	选中表示自动进行“上位词”扩展检索。
下位词	选中表示自动进行“下位词”扩展检索。
所有上位词	选中表示自动进行“所有上位词”扩展检索。
所有下位词	选中表示自动进行“所有下位词”扩展检索。
外文等同词	选中表示自动进行“外文等同词”扩展检索。
简体/繁体	选中表示自动进行“简体/繁体”扩展检索。
全角/半角	选中表示自动进行“全角/半角”扩展检索。

2.6.4 语法颜色

“语法颜色”属性页如下所示：



图 2—28 选项的语法颜色属性页

本属性页主要用于设置语句分析时各种不同类型语法符号的颜色，包括命令窗口和文件编辑窗口中的脚本文件编辑，各选项的含义如下：

控制名称	功能简介
语句结束符	语句结束符的颜色。
未知符号	语句中无法分辨的未知符号的颜色。
注释	语句注释部分的颜色。
保留字	语句中保留字的颜色。
标识符	语句中标识符的颜色。
字符串	语句中字符串的颜色。
数值	语句中数值的颜色。
密码	语句中密码的颜色。
文件路径	语句中文件路径的颜色。
表达式	语句中表达式的颜色。
操作符	语句中操作符的颜色。

2.7 获取帮助

2.7.1 界面使用

TRS 命令语言工具有详尽的参考说明，选择“帮助|帮助主题”命令即可打开帮

助系统。TRS 命令语言的帮助信息和 TRS 管理员的帮助信息包含在同一个文件里，可互相参考。同时，TRS 命令语言的每个对话框里的每一个控制都有相应的说明，只需单击对话框右上角的  按钮，然后再点击相应的控制即可查看该控制的功能说明。

注意：由于 TRS 管理员的帮助系统采用 HTML 方式，因此，如要浏览帮助，必须确保当前的操作系统上安装了 Microsoft Internet Explorer 6.0 或以上版本。

2.7.2 语法参考

TRS 命令语言的帮助文件里同样包含详尽的语法规则说明。除了可以通过选择“帮助|帮助主题”命令打开帮助以外，还可以通过 **HELP** 命令直接查看具体语句的规则说明。如 **HELP** 命令后没有指定具体语句，则显示命令语言的帮助主题。

2.7.3 获取更多信息

选择“帮助|TRS 主页”命令，或单击工具栏右上方的  标志可打开 TRS 公司的主页。在该站点上，你可以了解到 TRS 公司的简要介绍、最新动态、产品系列、解决方案等内容，并可获得最新的版本更新和技术支持信息。

第3章 语法详解

3.1 语法概述

TRS 命令语言简称 TRSL，是 TRS 全文数据库对外提供的一套 SQL 语句集合，用于快速执行各种 TRS 操作，尤其适合于批处理方式，能大大地简化操作步骤。TRSL 共包含 35 条语句，对每条 TRSL 语句，按目的、前提、语法、参数、要点、举例和参考的顺序描述。下面首先介绍 TRSL 的基本语法描述的约定：

大写粗体英文词

TRSL 保留字。如 **SELECT * FROM** *newsbase* 语句中的 **SELECT** 和 **FROM** 均为保留字。但用户在输入时大小写无关。

小写英文词或汉字

用户输入的信息。如 **CREATE DATABASE** *newsbase(txt DOCUMENT)* 语句中的 *newsbase* 和 *txt* 分别为数据库名和字段名。

.

用户名和数据库对象名的分隔符。如 **SELECT * FROM** *system.newsbase* 语句中的用户名 *system* 和数据库名 *newsbase* 之间的圆点。

[]

表示可选项。如 **NOTEPAD** [*file_name*] 语句中的方括号内的 *file_name* 表示可选项。

{ }

表示至少出现一个元素。如 **CREATE DATABASE** [*user.*]*database_name*(*{...}*) 语句中的大括号表示每个数据库至少有一个字段。

()

系统语法要求，必须输入。

<>

为可选圆括号（语句格式符“()”）的简写，即等价于“[()]”，用于消除歧义。表示圆括号可以出现，也可以不出现。

字符串

规定用户在输入字符串时必须带单引号，字段值或全文检索词则既可以带也可以不带单引号。如：**CREATE DATABASE** *newsbase(txt DOCUMENT DISPLAY '全文')*; **SELECT * FROM** *newsbase* **WHERE** *中国*;

;

TRSL 语句结束符(回车也是 TRSL 语句结束符)。多个 TRSL 语句之间必须以分号隔开，

脚本文件中的 TRSL 语句之间必须带分号。

“|”所连接的项最多只能出现一个。如 **BROWSE FORMAT [RECORD|TRS]** 语句中的方括号内的 **RECORD** 和 **TRS** 表示可选项只能有一个。

多个对象之间的分隔符。如 **SELECT newsbase, hlm WHERE 宝玉** 语句中数据库 *newsbase* 和数据库 *hlm* 之间的逗号。

下划线表示缺省值。

删除线表示当与语句不使用时的缺省值，与下划线不同。

在输入 TRSL 语句时，系统对大小写是不敏感的，你可以输入大写或小写，甚至大小写混合。但要记住，不能使用 TRSL 保留字作为数据库对象名(如数据库名、字段名等)。TRSL 语法的另一个特点是各子句的顺序无关，如：

```
TRSL> SELECT * FROM newsbase WHERE 中国 */sen 美国;
```

也可以写为：

```
TRSL> SELECT * WHERE 中国 */sen 美国 FROM newsbase;
```

TRS 脚本文件中可以使用注释，单行注释以//开头，从//开始到本行结束均为注释内容。也可以使用/*和*/把其间的一段文字标明为注释内容。

TRSL 里可以使用变量，变量以\$开头，变量名和 TRS 对象的命名规则一致。

3.2 语句列表

<u>ALTER</u>	修改对象	<u>MODIFY</u>	修改记录
<u>BACKUP</u>	备份数据库	<u>NOTEPAD</u>	编辑文件
<u>BROWSE</u>	浏览检索结果	<u>OPTIMIZE</u>	优化数据库
<u>CLASSIFY</u>	字段值分类	<u>PRINT</u>	打印检索结果
<u>CLEAR</u>	清除屏幕内容	<u>QUIT</u>	退出并清除操作日志
<u>COPY</u>	复制记录	<u>RECOVER</u>	恢复数据库
<u>CREATE</u>	创建新对象	<u>REM</u>	注释语句
<u>DELETE</u>	删除数据	<u>RENAME</u>	为对象改名
<u>DROP</u>	删除对象	<u>REVOKE</u>	回收用户的权限
<u>ECHO</u>	显示控制	<u>RUN</u>	运行脚本文件
<u>EDIT</u>	编辑检索结果	<u>SAVE</u>	保存检索结果
<u>EXIT</u>	退出命令语言	<u>SELECT</u>	执行检索
<u>GRANT</u>	给用户授权	<u>SET</u>	设置系统参数
<u>HELP</u>	显示帮助信息权	<u>SHOW</u>	显示信息
<u>INSERT</u>	插入新记录	<u>SORT</u>	检索结果重新排序
<u>LOAD</u>	把记录装进数据库	<u>SPOOL</u>	输出到文件
<u>LOCK</u>	锁定数据库	<u>UNLOCK</u>	解锁数据库

LOGIN	登录到服务器	UPDATE	更新记录
LOGOUT	注销登录	VALIDATE	验证索引
MAIL	发送检索结果		

3.3 语法规则

3.3.1 ALTER 修改对象属性和结构

【目的】

修改数据库、视图、词典、用户组、用户、数据格式的属性和结构。

【前提】

只有 SYSTEM 用户才有权修改用户组、用户和数据格式的属性；修改数据库、视图、词典的属性和结构时，操作用户必须对指定对象有 ALTER 权限。

【语法】

1. 修改数据库/词典的属性和结构

ALTER DATABASE database_name

[**ADD**(((**<basecol_element>**[, ...]))|**AS**{[[owner.]source_database_name ((**<column_name** [, ...]))}] [, ...])}]

[**MODIFY** ((**<mod_basecol_element>** [, ...])]

[**DROP** {**column_name** [, ...]}]

[**COMMENT** {**comment** | **NULL**}]

[**CHARSET** {**DEFAULT** | **GB18030** | **BIG5** | **ENGLISH** | **UTF8**}]

[**ALIAS** {**alias** [, ...] | **NULL**}]

[**DEFCOL** **column_name**]

[**PATH** **data_path**]

[**HOST** {**data_host** | **NULL**}]

[**SEGMENT** {[owner.]segment_dic | **DEFAULT** | **NULL**}]

[**SEGMENTEX** {[owner.]segmentex_dic | **DEFAULT** | **NULL**}]

[**STOP** {[owner.]stop_dic | **DEFAULT** | **NULL**}]

[**STOPEX** {[owner.]stopex_dic | **DEFAULT** | **NULL**}]

[**SPARSE** {[owner.]sparse_dic | **DEFAULT** | **NULL**}]

[**STEM** {**ON** | **OFF**}]

[**CASE** {**ON** | **OFF**}]

[NUMBER {ON | OFF}]
[BIGRAM {ON | OFF}]
[COMPRESS {ON | OFF}]
[BANSTOP {ON | OFF}]
[PUBLIC {{ALTER, DROP, INDEX, SELECT, UPDATE} | NULL}]

database_name:

待修改的数据库名。

ADD:

增加数据库字段，basecol_element 的见以下的说明。

MODIFY:

修改数据库字段的属性。mod_basecol_element 的定义见以下的说明。

DROP:

删除的数据库字段。

basecol_element:

数据库字段及其属性，其语法定义如下：

column_name

{DATE | NUMBER[(length)] | CHAR[(length)] | PHRASE | DOCUMENT | BIT}
[COMMENT {comment | NULL}]
[ALIAS {{alias [, ...]} | NULL}]
[FORMAT {{format | @filename} | NULL}]
[VALUES [NULL | MUST] [MULTI | SINGLE] [UNIQUE | ITERATIVE]]
[RANGE {value1 [,...] | @filename | NULL}]
[DEFAULT {NULL | default_vaue | @filename }]
[PATH {index_path | NULL}]
[INDEX {OFF | [ON | index_attr | @filename] [NORMAL | REDUCHAR |
REDUFULL | REDUHALF | REDUZERO] [ALLIDEO | NOTIDEO]}]
[LANGUAGE {stem_language | NULL}]
[DISPLAY {title | NULL}]
[WIDTH width]
[HEIGHT height]
[CACHE {ON | OFF}]
[RELEVANCE {ON | OFF}]
[DUMMY {ON | OFF | DEFCOL}]
[PUBLIC {SELECT | NULL}]

column_name:

数据库字段名。

DATE:

日期型字段，用来存储“年月日时分秒”这类表示日期的数据。

NUMBER:

数值型字段，用来存储整数或实数。对于实数，可以用 *length* 限制其小数位的长度，在数据库加载时将忽略多余的小数位。整数或实数不能超出机器的单精度浮点数所能表示的范围，可以使用科学计数法表示这些数据。

CHAR:

字符串型字段，可以用来存储任何形式的字符串。字符串的最大长度可以用 *length* 指定，但不能超过 255 个字符，在数据库加载时将截断超过规定长度的字符串，而只保留字符串的前部。

PHRASE:

短语型字段，可以用来存储任何形式、任意长度的文本数据，最适合作为“标题”这样的字段。短语型字段与字符串型字段的最大差别在于短语型字段支持自动分词，而字符串型字段不支持自动分词，其字段值整体是一个索引键值。

DOCUMENT:

全文型字段，既可以用来存储任何形式、任意长度的文本数据，也可以用来存储其它格式类型的多媒体数据。

BIT:

二进制型字段，可以用来存储任何格式类型的多媒体数据。

COMMENT:

数据库字段的注释。缺省值为空。

ALIAS:

数据库字段的别名列表，一个数据库字段可以有多个别名。缺省值为空。

FORMAT:

数据库字段的输入输出格式。

对于 CHAR 型字段，其格式规定为：

- A 在相应位置上必须出现一个英文字母；
- 9 在相应位置上必须出现一个数字；
- 空格 表示在相应位置上必须出现一个空格；
- X 在相应位置上可出现任意字符。

对于 DATE 和 NUMBER 型字段，则为字段值的输出限定格式，并且格式由用户确定，如“Y.MM.DD HH:MM:SS”、“L.DP”等。

如果限定格式第一个字符为‘@’，则表示其中存储的是限定格式所在文件的文件名。缺省值为空，表示字段的值没有格式限制。

VALUES:

字段的索引标志。**MUST** 表示必须有值，缺省为 **NULL**；**MULTI** 为缺省值，表示允许多值(**PHRASE** 和 **DOCUMENT** 字段除外)；**SINGLE** 表示不允许多值；**UNIQUE** 表示取值不允许重复，仅对 **DATE**、**NUMBER** 和 **CHAR** 型字段有效，缺省为 **ITERATIVE**。

RANGE:

DATE、NUMBER 或 CHAR 型字段的取值范围。取值范围的定义格式包括：

- 枚举定义：格式为 V1,V2,V3,Vn，即枚举该字段上所有可能取值；
- 区间定义：格式为 V1,...,V2，表示该字段上所有可能取值必须在最小值 V1 和最大值 V2 之间；
- 混合定义：格式为 V1,V2,...,V3,Vn，是以上两种方式的混合。

以上各格式中的逗号和点号都必须是半角符号。CHAR 型字段的取值范围所给出的每个值串首尾不允许空格。DATE 型字段的取值范围所给出的每个日期中只支持绝对（完全）年份，而不支持缺省世纪数的短格式年份，即总是认为缺省世纪数为 0。只有 DATE 型字段的取值范围支持模糊格式，即允许日期格式中缺省一些成份，但对于区间定义定义的范围，其中的 V1 和 V2 “前模糊”的成份必须是相同的，即在“年月日时分秒”的顺序下，V1 和 V2 的第一个确定的日期成份的位置必须是相同的。如果第一个字符为‘@’，则表示其中存储的是取值范围所在文件的文件名。缺省值为空，表示字段的值没有范围限制。

DEFAULT:

DATE、NUMBER 或 CHAR 型字段的缺省值。如果第一个字符为‘@’，则表示其中存储的是缺省值所在文件的文件名。缺省值为空，表示没有缺省值。

PATH:

数据库字段索引的存放路径，必须是系统配置中的数据和索引路径之一。缺省值为空，表示字段索引路径和数据库存放路径一致。

INDEX:

字段的索引属性。index_attr 为 DOCUMENT 型字段的倒排索引属性名，为字符串类型。多个属性名之间用‘;’分隔，如“SEG;SEN;POS”等。对 DOCUMENT 字段缺省值为“POS”，对其它类型的字段缺省值为空。对 DOCUMENT 和 PHRASE 字段进行字词混合索引时，混合索引的模式有 **NORMAL**（不进行字词混合索引）、**REDUCHAR**（禁止按词索引）、**REDUFULL**（全冗余模式）、**REDUHALF**（半冗余模式）、**REDUZERO**（零冗余模式）五种；**ALLIDEO** 表示允许进行单字检索，缺省值为 **NOTIDEO**。

LANGUAGE:

文本型字段索引进行词根处理时的语种。

DISPLAY:

数据库字段的显示标题，为字符串类型，只供界面使用。缺省值为字段名。

WIDTH:

数据库字段的显示宽度，只供界面使用，缺省值为零。

HEIGHT:

数据库字段的显示高度，只供界面使用，缺省值为零。

CACHE:

允许进行检索缓冲标志。

RELEVANCE:

是否参与相关性排序标志。

DUMMY:

是否为虚拟逻辑字段/缺省虚拟逻辑字段标志。

PUBLIC:

数据库字段的公共访问标志，只能为 **SELECT** 标志。

注意：除字段名和字段类型以外，其它属性都可以不指定，没有指定的属性值自动保持原来的值。

COMMENT:

数据库的注释，为字符串类型。

CHARSET:

数据库所用的语言，可为 **GB18030**、**BIG5**、**ENGLISH**、**UTF8** 之一，缺省 **DEFAULT** 为系统配置中的缺省语言环境。

ALIAS:

数据库的别名，一个数据库可有多个别名。

DEFCOL:

进行全文检索的缺省全文字段名。缺省为最后一个 **DOCUMENT** 或者 **PHRASE** 字段，**DOCUMENT** 字段优先。

PATH:

数据库存放的路径，必须是系统配置中的数据 and 索引路径之一。

HOST:

数据库存放的位置。如果不为空，表明实际数据集中存储在一个数据宿主服务器中，并指向存储服务器的有关信息的字符串，如“ORACLE”。

SEGMENT:

数据库所用的主分词词典名。

SEGMENTEX:

数据库所用的附加分词词典名。

STOP:

数据库所用的主停用词典名。

STOPEX:

数据库所用的附加主停用词典名。

SPARSE:

数据库所用的稀疏词典名。

STEM:

为 **ON** 表示只建立词根索引。

CASE:

为 **ON** 表示建立索引时区分英文词的大小写。

NUMBER:

为 **OFF** 表示建立索引时忽略数字串。

BIGRAM:

为 **ON** 表示建立二元组索引。

COMPRESS:

为 **ON** 表示压缩索引。

BANSTOP:

为 **ON** 表示创建索引时禁止启用停用词典。

PUBLIC:

数据库的公共访问标志，对于用户自定义数据库可为 **SELECT**、**UPDATE**、**INDEX**、**ALTER**、**DROP** 之一，对于词典可为 **SELECT**、**UPDATE**、**DROP** 之一。使用 **NULL** 可把公共访问标志设为空。为了提高效率，若设置的权限对象不支持时不会报错，只是当请求发送到 **SERVER** 端后不支持的权限标志将会被忽略。

修改数据库属性和结构时要注意以下事项：

- 除数据库名外，其它参数都可以不指定，不指定时保持数据库原来的属性值。
- 本语句不能修改字段名，要修改字段名请用 **RENAME** 语句。
- 当数据库中已有数据时，数据库的 **LANGUAGE**、**FIRSTCOL**、**PATH** 和 **HOST** 属性不能修改。当数据库中已有数据且已有字段建立了索引时，数据库的所有索引标志和使用词典属性均不能被修改。修改字段属性时，若某字段已经有了数据，则此字段的 **TYPE**、**INDEX**、**MULTI**、**UNIQUE**、**NULL**、**PATH**、**ATTR** 等属性不能修改，字段长度只能增加，不能减小。由以上要点可以看出，当数据库中已经存在数据时，修改数据库结构的操作有很多限制，定义数据库时，应预先作比较周到的考虑，尽量不要在数据已经入库后再修改数据库的属性。

2. 修改视图的属性和结构

ALTER VIEW view_name

```
[ADD{{{<viewcol_element>[,...]}|[column_name[,...]|ALL}}FROM  
{[owner.]database_name [, ...]}}  
[MODIFY {{{<viewcol_element> [, ...]} | [{column_name [, ...]} | ALL] {INSERT |  
DELETE}} {[owner.]database_name [, ...]}}}  
[DROP {column_name [, ...]}}  
[WHERE {condition | NULL}}  
[COMMENT {comment | NULL}}  
[ALIAS {alias [, ...] | NULL}}  
[DEFCOL default_column]  
[PUBLIC {{{ALTER, DROP, SELECT} | NULL}}]
```

view_name:

待修改的视图名。

ADD:

增加视图字段，viewcol_element 的定义请参见下面的说明。

MODIFY:

修改视图字段的属性。viewcol_element 的定义请参见下面的说明。

DROP:

删除的视图字段。

viewcol_element:

视图字段及其属性，视图字段的语法定义如下：

```
column_name
    [({[owner.]source_database.column [, ...]})]
    [COMMENT {comment | NULL}]
    [ALIAS {{alias [, ...]} | NULL}]
    [DISPLAY {title | NULL}]
    [WIDTH width]
    [HEIGHT height]
    [PUBLIC {SELECT | NULL}]
```

column_name:

视图字段名。

source_database.column:

视图字段对应的源数据库字段名，字段所在的数据库由[owner.]source_database 指定。一个视图字段可对应多个数据库字段，但这些字段的类型必须一致。

COMMENT:

视图字段的注释，为字符串类型，缺省值为空。

ALIAS:

视图字段的别名列表，一个视图字段可以有多个别名。缺省值为空。

DISPLAY:

视图字段的显示标题，只供界面使用。缺省值为字段名。

WIDTH:

视图字段的显示宽度，只供界面使用，缺省值为零。

HEIGHT:

视图字段的显示高度，只供界面使用，缺省值为零。

PUBLIC:

视图字段的公共访问标志，只能为 **SELECT**。

注意：除字段名和源数据库字段列表以外，其它属性都可以不指定，此时会自动保持原来的值。

WHERE:

视图的创建条件。

COMMENT:

视图的注释。

ALIAS:

视图的别名，一个视图可有多个别名。

DEFCOL:

进行全文检索的缺省全文字段名。缺省为最后一个 **DOCUMENT** 或者 **PHRASE** 字段，**DOCUMENT** 字段优先。

PUBLIC:

视图的公共访问标志，可为 **SELECT**、**ALTER**、**DROP** 之一，若为自动模式视图还包含 **UPDATE** 和 **INDEX** 权限。使用 **NULL** 可把公共访问标志设为空。为了提高效率，若设置的权限对象不支持时不会报错，只是当请求发送到 **SERVER** 端后不支持的权限标志将会被忽略。

修改数据库属性和结构时要注意以下事项：

- 除视图名外，其它参数都可以不指定，不指定时保持视图原来的属性值。
- 本语句不能修改字段名，要修改字段名请用 **RENAME** 语句。
- 视图涉及的数据库的结构可以不同。

3. 修改用户组的属性

```
ALTER GROUP group_name  
    [COMMENT {comment | NULL}]
```

group_name:

待修改创建的用户组名。

COMMENT:

用户组的注释，缺省值为空。

注意：除用户组名外，其它参数都可以不指定，不指定时保持用户组原来的属性值。

4. 修改用户的属性

```
ALTER USER user_name  
    [TYPE {DBA | GDA | RESOURCE | CONNECT}]  
    [PASSWORD {password | NULL}]  
    [GROUP {group_name | NULL}]  
    [COMMENT {comment | NULL}]  
    [HISTORY history]  
    [DOWNLOAD download]  
    [SECURITY security]
```

[**SEGMENTEX** {[owner.]segmentex_dic[**NULL**]}]
[**STOPEX** {[owner.]stopex_dic[**NULL**]}]
[**SPARSE** {[owner.]sparse_dic[**NULL**]}]
[**EXCLUSIVE** {**ON** | **OFF**}]

user_name:

待修改的用户名。

TYPE:

用户的类型，可为 **DBA**、**GDA**、**RESOURCE** 或 **CONNECT** 之一。

PASSWORD:

用户的新密码，为 **NULL** 表示没有密码。

GROUP:

用户所在的组名，为 **NULL** 表示它不属于任何特定的组。

COMMENT:

用户的注释。

HISTORY:

用户的最大检索历史步数，不设置或者为零时取缺省值，即系统配置中设定的最大检索历史步数。取值范围为 0 到 6553。

DOWNLOAD:

用户的最大下载记录数，不设置或者为零时取缺省值，即系统配置中设定的最大下载记录数。取值范围为 0 到 0xFFFFFFFF。

SECURITY:

用户的安全控制级别，用户只能访问安全级别不小于自己的安全控制级别的记录。取值范围为 0 到 0xFFFFFFFF，缺省值为零，表示该用户不能查看任何用密级加密的记录。

SEGMENTEX:

用户缺省的附加分词词典名。创建该用户的数据库时，如不指定附加分词词典，则自动取此词典作为数据库的附加分词词典。

STOPEX:

用户缺省的附加停用词典名。创建该用户的数据库时，如不指定附加停用词典，则自动取此词典作为数据库的附加停用分词词典。

SPARSE:

用户缺省的稀疏词典名。创建该用户的数据库时，如不指定稀疏词典，则自动取此词典作为数据库的稀疏词典。

EXCLUSIVE:

为 **ON** 表示不允许多个 IP 地址上的 Client 用户同时用该用户名登录。

修改用户属性时要注意以下事项：

- 除用户名外，其它参数都可以不指定，不指定时保持用户原来的属性值。
- 任何用户可修改“自己”的某些属性，这些属性包括：**PASSWORD**、**HISTORY**、**SEGMENTEX**、**STOPEX**、**SPARSE**、**COMMENT**。其它项只能由 **SYSTEM** 用户

管理。

- 用户的口令只能由用户自己修改，但 **SYSTEM** 用户能删除任何用户的口令。

5. 修改数据格式的屬性

```
ALTER FORMAT format_name  
    [CLASS {class | NULL}]  
    [SUFFIX {{suffix [, ...]} | NULL}]  
    [COMMENT {comment | NULL}]  
    [COMPRESS {ON | OFF}]
```

format_name:

待修改的数据格式名。

CLASS:

数据格式的分类名。

SUFFIX:

数据格式所对应的文件的后缀名列表，为字符串类型，内容需加上单引号，一种数据格式可有多个文件后缀名，多个文件后缀名之间以分号或者逗号分隔，如'HTM;HTML'。

COMMENT:

数据格式的注释。

COMPRESS:

为 **ON** 表示进行无损压缩，为 **OFF** 表示不进行压缩。

注意：除数据格式名以外，其它参数都可以不指定，不指定时保持数据格式原来的属性值。

【举例】

1. 修改数据库结构和属性

```
TRSL> ALTER DATABASE trs_group  
    ADD (address CHAR(20) INDEX)  
    MODIFY (photo MULTI)  
    DELETE department, office  
    PUBLIC SELECT
```

这条语句的作用是为数据库 *trs_group* 增加一个新的字段 *address*，类型为 **CHAR** 型，定长 20 个字符，且此字段为索引字段，同时把 *photo* 字段改为 **MULTI** 字段，删除 *department* 和 *office* 字段，并把该数据库设为公共检索数据库。

2. 修改视图结构和属性

```
TRSL> ALTER VIEW vnews
      ADD (标题(demo1.标题, demo2.题目), 作者(demo1.作者, demo2.author))
      DELETE (版次, 栏目)
      WHERE 中国*美国
```

这条语句的作用是向视图 *vnews* 里增加 *标题*和 *作者*两个字段，同时删除 *版次*和 *栏目*两个字段，并把创建条件改为‘*中国*美国*’。

3. 修改用户组属性

```
TRSL> ALTER GROUP product COMMENT 'Product department'
```

这条语句的作用是把 *Product* 用户组的说明信息改为 *Product department*。

4. 修改用户的属性

```
TRSL> ALTER USER bill TYPE CONNECT GROUP product
```

这条语句的作用是把 *bill* 用户改为 CONNECT 类型，并把该用户的所在组改为 *product*。

5 修改数据格式的的属性

```
TRSL> ALTER FORMAT html SUFFIX 'htm,html,htx' COMPRESS ON
```

这条语句的作用是把数据格式 *html* 的后缀名列表改为 *htm*、*html*、*htx*，对这种文件进行压缩处理。

【参考】

[CREATE](#) 创建新对象

[RENAME](#) 为对象改名

[DROP](#) 删除对象

3.3.2 BACKUP 备份数据库

【目的】

把一个或多个 TRS 数据库备份到服务器端的文件或设备里。

【前提】

操作用户必须是 DBA 用户，或是数据库所有者所在组的 GDA 用户，或者是所有待备份的数据库的拥有者。

【语法】

```
BACKUP [ALL] | {[owner.]database_name [, ...]} | [owner.]auto_view_name]
      TO backup_file
```

【参数】

ALL:

如果当前操作者是 DBA 用户，则备份所有的数据库；否则备份当前操作用户的所有数据库。此选项是缺省情况。

owner:

数据库的拥有者，可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。

database_name/auto_view_name:

需要备份的数据库名，可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。

backup_file:

存放数据库备份集的目标设备名，可以是服务器端的文件名或磁带名。不指定目标设备名时自动备份到系统缺省备份设备。

【要点】

1. 除了数据库中的数据 and 索引被备份外，下列信息也将被备份到备份集中：数据库的属性、字段的属性、索引的属性等数据词典信息，但权限信息不进行备份。
2. 数据库被备份时，处于 UPDATE 锁定状态，系统中的 TRS 用户不能修改数据库中的内容，但可以进行查询操作。
3. 如果用户希望从数据库中下载(Download)一部分数据，可以使用检索并存储检索结果的功能，而不使用 BACKUP 功能。
4. 备份集中的数据是格式化文件，只能用于数据库恢复。由于不同平台的文件格式可能存在差异，TRS 并不总能保证在一种平台上备份的数据库可以在其它平台上进行数据库恢复，一个数据库从一种平台上移到另外一种平台的最可靠方法是把数据库记录以 TRS 格式文件存储，从另一个台装库。

【举例】

TRSL> **BACKUP demo TO /trshome/temp/demo.bak**

这条语句的作用是把当前用户的 *demo* 数据库备份到服务器的 */trshome/temp/demo.bak* 文件里。

TRSL> **BACKUP ALL TO /dev/rst0;**

这条语句的作用是把当前用户的所有数据库备份到服务器的 */dev/rst0* 设备下。

【参考】

[RECOVER 恢复数据库](#)

3.3.3 BROWSE 浏览检索结果

【目的】

对当前检索结果进行浏览。

【前提】

当前存在检索结果记录集，并且操作用户对所浏览的字段有 **SELECT** 权限。

【语法】

```
BROWSE [RECORD {ALL | {{n, [ TILL m }} [, ...]]] [DETAIL]  
      [COLUMN {ALL | {column_name [, ...]}]}
```

【参数】

RECORD:

选择需要浏览的记录号(或记录号集合)，用 **TILL** 关键字连接两个记录号可指定一定范围的记录。取值 **ALL** 表示浏览所有的命中记录，为缺省值。

DETAIL:

表示直接显示检索结果的细览。

COLUMN:

选择显示的字段名(或字段名集合)，取值 **ALL** 表示显示所有的字段，为缺省值。

【要点】

1. 当不指定 **DATABASE**、**COLUMN**、**RECORD** 参数时，浏览所有的命中结果。
2. *column_name* 为需显示的字段名，当 **BROWSE** 后不带字段选择参数时，将显示所有字段。
3. 当指定需要浏览的记录时，所指定的记录号是命中的记录号，不是记录的内部顺序号，当需要按内部记录号进行查询时，可以使用以下 TRSL 语句：

```
TRSL> SELECT FROM 库名 WHERE DOCID=docid
```

4. 本命令只能浏览文本数据，不能浏览记录中的多媒体数据（如图象）、格式化文件（如 WPS 文件）、复合文档（MS WORD 文件）等。

【举例】

```
TRSL> BROWSE
```

这条不带任何参数的 **BROWSE** 语句表示显示所有的查询结果。

```
TRSL> BROWSE COLUMN name RECORD 1,3 TILL 5,9
```

表示浏览字段 *name*，浏览的命中记录为 1, 3, 4, 5, 9 共 5 个记录。

【参考】

[SELECT](#) 数据库检索

[EDIT](#) 编辑检索结果

[SAVE](#) 存贮检索结果

[PRINT](#) 打印检索结果

[MAIL](#) 邮寄检索结果

3.3.4 CLASSIFY 字段值分类

【目的】

按字段值对检索结果进行分类统计。

【前提】

当前存在检索结果记录集。

【语法】

```
CLASSIFY column_name  
    [AS [{<regular_expression> [, ...]} | VALUES {<focus_value> [, ...]}] [NOCASE]  
    [ORDER BY [{RELEVANCE | {column_name [ASC | DESC}}] [, ...]] [LIFO]  
    [{PRUNE | VALID} record_number] [MIXED] [MISCOL] [NOCASE]  
    [CUTSIZE class_cutsize]  
    [LIST BY [NAME | COUNT] [ASC | DESC]]
```

【参数】

column_name:

指定按值分类的字段。

AS:

指定分类时关注的字段值。

ORDER BY:

记录的排列顺序。对每个 column_name 可用 **ASC** 或 **DESC** 分别指定按升序/降序方式排列。字段名也可为 **RELEVANCE**，表示按相关性排序，但只能按升序排列。也可指定为 **LIFO**，表示按后进先出顺序排列。对于参数 record_number，若前面选择 **PRUNE** 时则代表排序并裁减保存的结果记录数，同时自动设置“非精确命中点”模式；选择 **VALID** 时则代表有效排序结果记录数，即部分排序，按排序表达式排序的结果只保证前一部分记录是有序的。**MIXED** 表示对检索结果进行混合统一排序。**MISCOL** 则表示对排序表达式中不存在的字段，按字段无值时情况处理。**NOCASE** 表示排序表达式中的字段值，其字符是大小写无关的。

CUTSIZE:

每个类中保留结果记录的个数。

LIST BY:

显示结果的排列方式。**NAME** 指按类别名排序，**COUNT** 值按每个类别的记录数排序。排序方式分为 **ASC** 升序和 **DESC** 降序排列两种。

【要点】

1. 必须首先进行对数据库/视图进行检索，并取得有效的检索句柄。
2. 重新排序后，分类统计将失效，检索结果恢复为原来的记录集。
3. 当设置了**CUTSIZE**后，**LIST BY**排序将不起作用。

【举例】

TRSL> **CLASSIFY** 版次 **LIST BY COUNT DESC**

表示按字段“版次”分类，显示结果按每个类别的记录数降序排列。

【参考】

[SELECT 数据库检索](#)

3.3.5 CLEAR 清除屏幕内容

【目的】

清屏。

【前提】

无，任何 TRS 用户均可操作。

【语法】

CLEAR

【参数】

无。

【要点】

CLEAR 操作将把当前屏幕中的所有内容清除，并在屏幕的最后一行显示命令行提示符 TRSL> 。

【举例】

TRSL> **CLEAR**

【参考】

无。

3.3.6 COPY 复制数据库记录

【目的】

把符合条件的记录从一个数据库/视图复制到目标数据库中。

【前提】

操作用户必须对目标数据库有 UPDATE 权限，并且对源数据库/视图有 SELECT 权限。

【语法】

```
COPY {[owner.]source_database_name | [owner.]source_auto_view_name} [(column_name  
[, ...])] [RAPIDLY]  
      INTO {[owner.]target_database_name | [owner.]target_auto_view_name}  
      [WHERE expression]  
      [ORDER BY [{RELEVANCE | {column_name [ASC | DESC]} [, ...]}] [LIFO]  
      [{PRUNE | VALID} record_number] [NOCASE]]  
      [RELEVANT {NULL | NORMAL | GLOBAL | NORMPOS | GLOBPOS | WORDS |  
      WFREQ}]
```

【参数】

owner:

待复制的数据库/视图的拥有者。

source_database_name/source_auto_view_name:

待复制的数据库/视图名。

column_name:

待复制的字段名，必须同时存在于目标数据库和源数据库/视图中，并且具有相同的字段类型，不指定时表示复制所有字段。字段名可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。

RAPIDLY:

如果复制记录时不需要同步创建索引，则可使用 RAPIDLY 选项。

INTO:

owner 表示目标数据库的拥有者。target_database_name 和 target_auto_view_name 分别表示目标数据库/视图名。

WHERE:

复制记录的条件，只有符合条件的记录才会被复制到目标数据库中。不指定时表示复制所有记录。

ORDER BY:

记录的排列顺序。对每个 `column_name` 可用 **ASC** 或 **DESC** 分别指定按升序/降序方式排列。字段名也可用 **RELEVANCE**，表示按相关性排序，但只能按升序排列。也可指定为 **LIFO**，表示按后进先出顺序排列。对于参数 `record_number`，若前面选择 **PRUNE** 时则代表排序并裁减保存的结果记录数，同时自动设置“非精确命中点”模式；选择 **VALID** 时则代表有效排序结果记录数，即部分排序，按排序表达式排序的结果只保证前一部分记录是有序的。**NOCASE** 表示不区分大小写。

RELEVANT:

以相关性模式和 **IDF** 模式组合的相关性排序选项。**NULL** 表示由系统配置决定；**NORMAL** 表示忽略词的位置和使用基于单个数据库范围的 **IDF** 值；**GLOBAL** 表示忽略词的位置和词 **IDF** 值不参与相关性计算；**NORMPOS** 表示考虑词的位置和使用基于单个数据库范围的 **IDF** 值；**GLOBPOS** 考虑词的位置和词 **IDF** 值不参与相关性计算；**WORDS** 表示用一个记录中命中词的单位向量长度（每个命中词只计算一次）作为记录的相关；**WFRE** 表示用一个记录中命中词的单位向量长度，以及命中词的词频和作为记录的相关度。

【要点】

无。

【举例】

TRSL> **COPY TO demo2 FROM demo1(日期,标题,正文) WHERE 中国*北京**这条语句的作用是把 `demo1` 数据库里包含“中国”和“北京”的记录复制到 `demo2` 数据库里，只复制 `日期`、`标题`和 `正文`三个字段。

【参考】

[LOAD 把记录装进数据库](#)

[INSERT 插入新记录](#)

[DELETE 删除数据库记录](#)

[UPDATE 更新数据库记录](#)

3.3.7 CREATE 创建新对象

【目的】

创建新的数据库、视图、词典、用户组、用户、数据格式和索引。

【前提】

只有 SYSTEM 用户才有权创建用户组、用户和数据格式；创建数据库、视图、词典时，操作用户必须具有 Resource 权限；创建索引时，当前操作者必须对数据库拥有 INDEX 权限。

【语法】

1. 创建数据库/词典

```
CREATE DATABASE database_name
    {{{<basecol_element> [, ...]}} | AS {{{[owner.]source_database_name {{{column_name
[, ...]}}}} [, ...]}}}
    [TYPE {NORMAL | SEGMENT | SEGMENTEX | STOP | STOPEX | SPARSE |
THESAURUS | SYNONYM | ANTONYM}]
    [COMMENT {comment | NULL}]
    [CHARSET {DEFAULT | GB18030 | BIG5 | ENGLISH | UTF8}]
    [ALIAS {{{alias [, ...]}} | NULL}]
    [DEFCOL default_column]
    [PATH data_path]
    [HOST {data_host | NULL}]
    [SEGMENT {[owner.]segment_dic | DEFAULT | NULL}]
    [SEGMENTEX {[owner.]segmentex_dic | DEFAULT | NULL}]
    [STOP {[owner.]stop_dic | DEFAULT | NULL}]
    [STOPEX {[owner.]stopex_dic | DEFAULT | NULL}]
    [SPARSE {[owner.]sparse_dic | DEFAULT | NULL}]
    [STEM {ON | OFF}]
    [CASE {ON | OFF}]
    [NUMBER {ON | OFF}]
    [BIGRAM {ON | OFF}]
    [COMPRESS {ON | OFF}]
    [BANSTOP {ON | OFF}]
    [PUBLIC {{{ALTER, DROP, INDEX, SELECT, UPDATE}} | NULL}]
```

database_name:

待创建的新数据库名。

basecol_element:

数据库字段及其属性，其语法定义如下：

column_name

```
{DATE | NUMBER[(length)] | CHAR[(length)] | PHRASE | DOCUMENT | BIT}
  [COMMENT {comment | NULL}]
  [ALIAS {{alias [, ...]} | NULL}]
  [FORMAT {{format | @filename} | NULL}]
  [VALUES [NULL | MUST] [MULTI | SINGLE] [UNIQUE | ITERATIVE]]
  [RANGE {value1 [,...] | @filename | NULL}]
  [DEFAULT {NULL | default_vaue | @filename } ]
  [PATH {index_path | NULL}]
  [INDEX {OFF | [ON | index_attr | @filename] [NORMAL | REDUCHAR |
REDUFULL | REDUHALF | REDUZERO] [ALLIDEO | NOTIDEO]}]
  [LANGUAGE {stem_language | NULL}]
  [DISPLAY {title | NULL}]
  [WIDTH width]
  [HEIGHT height]
  [CACHE {ON | OFF}]
  [RELEVANCE {ON | OFF}]
  [DUMMY {ON | OFF | DEFCOL}]
  [PUBLIC {SELECT | NULL}]
```

column_name:

数据库字段名。

DATE:

日期型字段，用来存储“年月日时分秒”这类表示日期的数据。

NUMBER:

数值型字段，用来存储整数或实数。对于实数，可以用 *length* 限制其小数位的长度，在数据库加载时将忽略多余的小数位数。整数或实数不能超出机器的单精度浮点数所能表示的范围，可以使用科学计数法表示这些数据。

CHAR:

字符串型字段，可以用来存储任何形式的字符串。字符串的最大长度可以用 *length* 指定，但不能超过 255 个字符，在数据库加载时将截断超过规定长度的字符串，而只保留字符串的前部。

PHRASE:

短语型字段，可以用来存储任何形式、任意长度的文本数据，最适合作为“标题”

这样的字段。短语型字段与字符串型字段的最大差别在于短语型字段支持自动分词，而字符串型字段不支持自动分词，其字段值整体是一个索引键值。

DOCUMENT:

全文型字段，既可以用来存储任何形式、任意长度的文本数据，也可以用来存储其它格式类型的多媒体数据。

BIT:

二进制型字段，可以用来存储任何格式类型的多媒体数据。

COMMENT:

数据库字段的注释。缺省值为空。

ALIAS:

数据库字段的别名列表，一个数据库字段可以有多个别名。缺省值为空。

FORMAT:

数据库字段的输入输出格式。

对于 CHAR 型字段，其格式规定为：

- A 在相应位置上必须出现一个英文字母；
- 9 在相应位置上必须出现一个数字；
- 空格 表示在相应位置上必须出现一个空格；
- X 在相应位置上可出现任意字符。

对于 DATE 和 NUMBER 型字段，则为字段值的输出限定格式，并且格式由用户确定，如“Y.MM.DD HH:MM:SS”、“L.DP”等。

如果限定格式第一个字符为‘@’，则表示其中存储的是限定格式所在文件的文件名。缺省值为空，表示字段的值没有格式限制。

VALUES:

字段的索引标志。**MUST** 表示必须有值，缺省为 **NULL**；**MULTI** 为缺省值，表示允许多值（**PHRASE** 和 **DOCUMENT** 字段除外）；**SINGLE** 表示不允许多值；**UNIQUE** 表示取值不允许重复，仅对 **DATE**、**NUMBER** 和 **CHAR** 型字段有效，缺省为 **ITERATIVE**。

RANGE:

DATE、**NUMBER** 或 **CHAR** 型字段的取值范围。取值范围的定义格式包括：

- 枚举定义：格式为 V1,V2,V3,Vn，即枚举该字段上所有可能取值；
- 区间定义：格式为 V1,...,V2，表示该字段上所有可能取值必须在最小值 V1 和最大值 V2 之间；
- 混合定义：格式为 V1,V2,...,V3,Vn，是以上两种方式的混合。

以上各格式中的逗号和点号都必须是半角符号。**CHAR** 型字段的取值范围所给出的每个值串首尾不允许空格。**DATE** 型字段的取值范围所给出的每个日期中只支持绝对（完全）年份，而不支持缺省世纪数的短格式年份，即总是认为缺省世纪数为 0。只有 **DATE** 型字段的取值范围支持模糊格式，即允许日期格式中缺省一些成份，但对于区间定义定义的范围，其中的 V1 和 V2 “前模糊”的成份必须是相同的，即在“年月日时分秒”的顺序下，V1 和 V2 的第一个确定的日期成份的位置必须是相同

的。如果第一个字符为‘@’，则表示其中存储的是取值范围所在文件的文件名。缺省值为空，表示字段的值没有范围限制。

DEFAULT:

DATE、NUMBER 或 CHAR 型字段的缺省值。如果第一个字符为‘@’，则表示其中存储的是缺省值所在文件的文件名。缺省值为空，表示没有缺省值。

PATH:

数据库字段索引的存放路径，必须是系统配置中的数据和索引路径之一。缺省值为空，表示字段索引路径和数据库存放路径一致。

INDEX:

字段的索引属性。index_attr 为 DOCUMENT 型字段的倒排索引属性名，为字符串类型。多个属性名之间用‘;’分隔，如“SEG;SEN;POS”等。对 DOCUMENT 字段缺省值为“POS”，对其它类型的字段缺省值为空。对 DOCUMENT 和 PHRASE 字段进行字词混合索引时，混合索引的模式有 **NORMAL**（不进行字词混合索引）、**REDUCHAR**（禁止按词索引）、**REDUFULL**（全冗余模式）、**REDUHALF**（半冗余模式）、**REDUZERO**（零冗余模式）五种；**ALLIDEO** 表示允许进行单字检索，缺省值为 **NOTIDEO**。

LANGUAGE:

文本型字段索引进行词根处理时的语种。

DISPLAY:

数据库字段的显示标题，为字符串类型，只供界面使用。缺省值为字段名。

WIDTH:

数据库字段的显示宽度，只供界面使用，缺省值为零。

HEIGHT:

数据库字段的显示高度，只供界面使用，缺省值为零。

CACHE:

允许进行检索缓冲标志。

RELEVANCE:

是否参与相关性排序标志。

DUMMY:

是否为虚拟逻辑字段/缺省虚拟逻辑字段标志。

PUBLIC:

数据库字段的公共访问标志，只能为 **SELECT** 标志。

注意：除字段名和字段类型以外，其它属性都可以不指定，没有指定的属性值自动保持原来的值。

AS:

从一个已存在的数据库中选择部分字段产生一个新的数据库。字段可属于一个数据库，

也可分别属于不同的数据库。但如果指定了 `source_database_name`, 则所有的 `column_name` 都默认为 `source_database_name` 的字段, 此时不能为字段指定数据库。

TYPE:

数据库或词典的类型, 可为 **NORMAL**、**SEGMENT**、**SEGMENTEX**、**STOP**、**STOPEX**、**SPARSE**、**THESAURUS**、**SYNONYM**、**ANTONYM** 之一, 缺省为 **NORMAL**。

COMMENT:

数据库的注释, 为字符串类型, 缺省值为空。

CHARSET:

数据库所用的语言, 可为 **GB18030**、**BIG5**、**ENGLISH** 和 **UTF8** 之一。缺省 **DEFAULT** 为系统配置中的缺省语言环境。

ALIAS:

数据库的别名, 一个数据库可有多个别名。缺省值为空。

DEFCOL:

进行全文检索的缺省全文字段名, 缺省为最后一个 **DOCUMENT** 或者 **PHRASE** 字段, **DOCUMENT** 字段优先。

PATH:

数据库存放的路径, 必须是系统配置中的数据 and 索引路径之一, 缺省为系统配置中的第一个数据路径。

HOST:

数据库存放的位置。如果不为空, 表明实际数据集中存储在一个数据宿主服务器中, 并指向存储服务器的有关信息的字符串, 如 “**ORACLE**”。缺省值为空。

SEGMENT:

数据库所用的主分词词典名。缺省为系统配置中相应语言的缺省主分词词典名。

SEGMENTEX:

数据库所用的附加分词词典名。缺省为数据库拥有者的缺省附加分词词典名。

STOP:

数据库所用的主停用词典名。缺省为系统配置中相应语言的缺省主停用词典名。

STOPEX:

数据库所用的附加主分词词典名。缺省为数据库拥有者的缺省附加停用词典名。

SPARSE:

数据库所用的稀疏词典名。缺省为数据库拥有者的缺省稀疏词典名。

STEM:

为 **ON** 表示只建立词根索引, 缺省为系统配置中设定的只建立词根索引标志。

CASE:

为 **ON** 表示建立索引时区分英文词的大小写, 缺省为系统配置中设定的大小写敏感标志。

NUMBER:

为 **ON** 表示建立索引时不忽略数字串, 缺省为系统配置中设定的忽略数字串标志。

BIGRAM:

为 **ON** 表示建立二元组索引, 缺省为 **OFF**。

COMPRESS:

为 **ON** 表示压缩索引，缺省为 **OFF**。

BANSTOP:

为 **ON** 表示创建索引时禁止启用停用词典。

PUBLIC:

数据库的公共访问标志，对于用户自定义数据库可为 **SELECT**、**UPDATE**、**INDEX**、**ALTER**、**DROP** 之一，对于词典可为 **SELECT**、**UPDATE**、**DROP** 之一。不指定时所有操作都不是公共的。为了提高效率，若设置的权限对象不支持时不会报错，只是当请求发送到 **SERVER** 端后不支持的权限标志将会被忽略。

创建数据库时要注意以下几点：

- 除数据库名外，其它参数都可以不指定，不指定时自动取缺省值（定义方式下）或源数据库的值（选择方式下）。
- 由于词典是一种特殊的数据库，所以此语句也能创建词典，但只能用定义方式，用 **TYPE** 子句指定词典的类型，并且不能指定字段。词典的属性只有 **ALIAS**、**LANGUAGE**、**PATH**、**HOST**、**COMMENT** 和 **PUBLIC** 有效。
- 以选择方式建库时只建立数据库的结构，并不从源数据库里复制记录。要复制记录可使用 **COPY** 语句。

2. 创建视图

CREATE VIEW view_name

```
{({<viewcol_element> [, ...]}) | AS {[owner.]database_name [, ...]} [SELECT
{column_name [, ...]}] | SPLIT [owner.]database_name}
[WHERE {condition | NULL}]
[COMMENT {comment | NULL}]
[ALIAS {{alias [, ...]} | NULL}]
[DEFCOL default_column]
[PUBLIC {{ALTER, DROP, SELECT} | NULL}]
```

view_name:

待创建的新视图名。

viewcol_element:

视图字段及其属性，视图字段的语法定义如下：

```
column_name
[[({[owner.]source_database.column [, ...]})]
[COMMENT {comment | NULL}]
[ALIAS {{alias [, ...]} | NULL}]
[DISPLAY {title | NULL}]
[WIDTH width]
[HEIGHT height]
```

[PUBLIC {SELECT | NULL}]

column_name:

视图字段名。

source_database.column:

视图字段对应的源数据库字段名，字段所在的数据库由[owner.]source_database 指定。一个视图字段可对应多个数据库字段，但这些字段的类型必须一致。

COMMENT:

视图字段的注释，为字符串类型，缺省值为空。

ALIAS:

视图字段的别名列表，一个视图字段可以有多个别名。缺省值为空。

DISPLAY:

视图字段的显示标题，只供界面使用。缺省值为字段名。

WIDTH:

视图字段的显示宽度，只供界面使用，缺省值为零。

HEIGHT:

视图字段的显示高度，只供界面使用，缺省值为零。

PUBLIC:

视图字段的公共访问标志，只能为 **SELECT**。

注意：除字段名和源数据库字段列表以外，其它属性都可以不指定，此时会自动保持原来的值。

AS:

从几个已存在的数据库中选择部分字段产生一个新的视图。字段可属于一个数据库，也可分别属于不同的数据库。**SELECT** 表示指定某个 database_name 中或者某几个 database_name 中具有相同名称和类型的 column_name 产生一个新的视图 字段。如果不使用 **SELECT** 子句，则自动从所指定的 database_name 中选择名称类型相同的 column_name 生成新字段。

SPLIT:

表示将一个数据库转换成自动模式视图。

WHERE:

视图的创建条件。缺省值为空。

COMMENT:

视图的注释，缺省值为空。

ALIAS:

视图的别名，一个视图可有多个别名。缺省值为空。

DEFCOL:

进行全文检索的缺省全文字段名，缺省为最后一个 DOCUMENT 或者 PHRASE 字段，DOCUMENT 字段优先。

PUBLIC:

数据库的公共访问标志，可为 **SELECT**、**ALTER**、**DROP** 之一，不指定时所有操作都不是公共的。为了提高效率，若设置的权限对象不支持时不会报错，只是当请求发送到 **SERVER** 端后不支持的权限标志将会被忽略。

创建视图时要注意以下几点：

- 除视图名外，其它参数都可以不指定，不指定时自动取缺省值。
- 在检索系统中，一个重要的查询功能是实现跨数据库查找，**TRS** 提供的 **CREATE VIEW** 功能为这种检索提供了方便的措施，一旦定义了视图，就不再需要每次检索时选择多个数据库，只要对视图进行检索就可以了。比如：要同时查询多个库的某一个字段，则可创建一个视图，定义其字段取自多个库的某个字段。
- 视图的功能之一是实现记录一级的安全控制，通过 **WHERE** 条件，可限制对数据库中某些数据的访问，如在人事档案数据库中普通职员不能查阅部门经理信息等需求。
- 视图涉及的数据库的结构可以不同。

3. 创建用户组

```
CREATE GROUP group_name  
    [COMMENT {comment | NULL}]
```

group_name:

待创建的用户组名。

COMMENT:

用户组的注释，缺省值为空。

注意：除用户组名外，其它参数都可以不指定，不指定时自动取缺省值。

4. 创建用户

```
CREATE USER user_name  
    [TYPE {DBA | GDA | RESOURCE | CONNECT}]  
    [PASSWORD {password | NULL}]  
    [GROUP {group_name | NULL}]  
    [COMMENT {comment | NULL}]  
    [HISTORY history]  
    [DOWNLOAD download]  
    [SECURITY security]  
    [SEGMENTEX {[owner.]segmentex_dic | NULL}]  
    [STOPEX {[owner.]stopex_dic | NULL}]
```

[**SPARSE** {[owner.]sparse_dic | **NULL**}]
[**EXCLUSIVE** {**ON** | **OFF**}]

user_name:

待创建的新用户名。

TYPE:

用户的类型，可为 **DBA**、**GDA**、**RESOURCE** 或 **CONNECT** 之一，缺省为 **CONNECT**。

PASSWORD:

用户的密码。

GROUP:

用户所在的组名。

COMMENT:

用户的注释。缺省值为空。

HISTORY:

用户的最大检索历史步数，不设置或者为零时取缺省值，即系统配置中设定的最大检索历史步数。取值范围为 0 到 6553。

DOWNLOAD:

用户的最大下载记录数，不设置或者为零时取缺省值，即系统配置中设定的最大下载记录数。取值范围为 0 到 0xFFFFFFFF。

SECURITY:

用户的安全控制级别，用户只能访问安全级别不小于自己的安全控制级别的记录。取值范围为 0 到 0xFFFFFFFF，缺省值为零，表示该用户不能查看任何用密级加密的记录。

SEGMENTEX:

用户缺省的附加分词词典名。创建该用户的数据库时，如不指定附加分词词典，则自动取此词典作为数据库的附加分词词典。缺省值为 **NULL**。

STOPEX:

用户缺省的附加停用词典名。创建该用户的数据库时，如不指定附加停用词典，则自动取此词典作为数据库的附加停用分词词典。缺省值为 **NULL**。

SPARSE:

用户缺省的稀疏词典名。创建该用户的数据库时，如不指定稀疏词典，则自动取此词典作为数据库的稀疏词典。缺省值为 **NULL**。

EXCLUSIVE:

为 **ON** 表示不允许多个 IP 地址上的 Client 用户同时用该用户名登录。缺省值为 **OFF**。

注意：除用户名外，其它参数都可以不指定，不指定时自动取缺省值。

5. 创建数据格式

CREATE FORMAT format_name

[**CLASS** {class | **NULL**}]
[**SUFFIX** {{suffix [, ...]} | **NULL**}]
[**COMMENT** {comment | **NULL**}]
[**COMPRESS** {**ON** | **OFF**}]

format_name:

待创建的数据格式名。

CLASS:

数据格式的分类名。缺省值为空。

SUFFIX:

数据格式所对应的文件的后缀名列表，为字符串类型，内容需加上单引号，多个文件后缀名之间以分号或者逗号分隔，如'HTM;HTML'。缺省值为空。

COMMENT:

数据格式的注释，缺省值为空。

COMPRESS:

为 **ON** 表示进行无损压缩，为 **OFF** 表示不进行压缩，缺省值为 **ON**。

注意：除数据格式名以外，其它参数都可以不指定，不指定时自动取缺省值。

6. 创建索引

```
CREATE INDEX {[owner.]database_name | [owner.]auto_view_name} [(column_name  
[, ...])]  
[TILL record]
```

database_name/auto_view_name:

创建索引时，指定索引字段所在的数据库名，可由 *user* 指定数据库的拥有者。

column_name:

创建索引时，指定索引所在的字段名。可使用通配符，用 '?' 匹配单个字符，用 '*' 匹配任何字符。

record:

创建索引的终止记录号，不指定时表示创建所有记录的索引。

创建索引时要注意以下两点：

- 一次可为多个字段创建索引。
- 一般情况下，索引由系统自动维护，只有该字段上的索引被删除，或不需要在该字段上进行检索时才有必要手动创建索引。

【举例】

1、以定义方式创建新数据库

```
TRSL> CREATE DATABASE newsbase(  
    日期 DATE,  
    版次 NUMBER,  
    版面 CHAR(10),  
    标题 PHRASE,  
    作者 CHAR(10) MULTI,  
    正文 DOCUMENT,  
    图象 BIT MULTI )  
    PATH /trshome/ibase  
    SEGMENT system.syssegdic  
    STOP system.sysstopdic  
    NUMBER OFF  
    COMMENT '新闻资料数据库'
```

这条语句的功能是：创建一个名为 *newsbase* 的数据库，共有 7 个字段，数据存放在服务器上的 */trshome/ibase* 目录下，此数据库使用的分词词典为用户 *system.syssegdic*，停用词典为 *system.sysstopdic*，记录中的数字不建索引，因此不能用数字作为检索条件，备注中说明此数据库为 '新闻资料数据库'。该语句中没有定义其它数据库属性，这些属性由 TRS 数据库服务器根据系统配置自动确定。

2、以选择方式创建新数据库

```
TRSL> CREATE DATABASE rmr1999  
    AS rmr(日期,标题,作者,正文,图象)  
    COMMENT '人民日报1999年'
```

这条语句的功能是：以 *rmr* 数据库的 *日期*、*标题*、*作者*、*正文*、*图象* 字段作为模板，创建一个名为 *rmr1999* 的新数据库，新数据库的注释为 '人民日报1999年'，其它属性取 *rmr* 数据库的属性值。

3、创建一部同义词典

```
TRSL> CREATE DATABASE mysynodic TYPE SYNONYM PUBLIC SELECT,  
    UPDATE
```

这条语句的功能是：创建一个名为 *mysynodic* 的新同义词典，所有用户均可对此词典进行检索和记录维护操作。

4、以定义方式创建新视图

```
TRSL> CREATE VIEW vnews(  
    新闻(news90.正文, news91.正文, news92.正文, news93.正文))
```

WHERE 版面=摄影

PUBLIC SELECT

这条语句的作用是把 news90、news91、news92、news93 四个数据库的正文字段定义为一个视图，视图的字段名为“新闻”，创建条件为‘版面=摄影’。四个数据库跨库查询可直接在视图 *vnews* 上进行，并且所有用户都可对该视图进行检索。

5、以选择方式创建新视图

TRSL> **CREATE VIEW** *engineer*

AS *personnel* **SELECT** 标题

这条语句的作用是对数据库 *personnel* 创建一个视图，并取“标题”字段作为新视图的字段。

6、创建新用户组

TRSL> **CREATE GROUP** *product*

这条语句的作用是创建名为 *product* 的新用户组。

7、创建新用户

TRSL> **CREATE USER** *newuser* **TYPE RESOURCE PASSWORD** *trs40* **SEGMENTEX**
mysegdic

这条语句的作用是创建一个名为 *newuser* 的新资源用户，密码为 *trs40*，并把该用户创建的数据库的缺省附加分词词典设为 *mysegdic*。

8、创建新数据格式

TRSL> **CREATE FORMAT** *mp3* **CLASS** *audio* **SUFFIX** '*mp1,mp2,mp3*' **COMPRESS**
OFF

这条语句的作用是创建一种名为 *mp3* 的新的 *audio* 数据格式，文件后缀名可为 *mp1*、*mp2* 或 *mp3*，对这种文件不进行压缩。

9、为数据库字段建立索引

TRSL> **CREATE INDEX** *personnel(resume)*

这条语句的作用是对数据库 *personnel* 的 *resume* 字段，为所有的记录建立索引。

【参考】

[ALTER](#) 修改对象

[RENAME](#) 为对象改名

[DROP](#) 删除对象

3.3.8 DELETE 删除数据

【目的】

从数据库中删除记录。

【前提】

操作用户对数据库有 UPDATE 权限。

【语法】

```
DELETE FROM {[owner.]database_name | [owner.]auto_view_name} [FINALLY]
           [WHERE condition]
```

【参数】

owner:

目标数据库的拥有者。

database_name/auto_view_name:

目标数据库/自动模式视图名。

FINALLY:

彻底删除数据与索引。

WHERE:

待删除记录的条件，只有满足条件的记录才会被删除。也可用 **DOCID=id** 表示物理记录号为 *id* 的记录。除此之外，本子句还可以指定删除词语。

【要点】

1. 如果不指定删除条件，则默认为删除数据库中的所有记录。
2. 本语句同样可以删除词典中的词语，对于主题词典、同义词典、反义词典用法和数据库一致，对其它词典则只需在 **WHERE** 子句中指定待删除的词语列表就行了。

【举例】

```
TRSL> DELETE FROM newsbase WHERE 苏联 COMPLETE
```

这条语句的作用是完全删除数据库 *newsbase* 中所有含有词“苏联”的记录。

```
TRSL> DELETE FROM mysegdic WHERE '苏联,民主德国'
```

这条语句的作用是删除词典 *mysegdic* 中指定的词语。

【参考】

[LOAD](#) 把记录装进数据库
[INSERT](#) 插入新记录
[UPDATE](#) 更新数据库记录

3.3.9 DROP 删除对象

【目的】

删除数据库、视图、词典、用户组或用户对象，或删除字段上的索引。

【前提】

删除用户组或用户时，当前操作者必须是 **SYSTEM** 用户；删除数据库/视图时，当前操作者必须对该对象有 **DROP** 权限；删除索引时，当前操作者必须对数据库拥有 **INDEX** 权限。

【语法】

1. 删除对象

```
DROP [DATABASE | VIEW | GROUP | USER]  
      [owner.]object_name, [[owner.]object_name]...
```

2. 删除索引

```
DROP INDEX [owner.]database_name  
            [(column_name [,column_name]...)]  
            [FROM record]
```

【参数】

DATABASE:

指明为删除数据库，为缺省值。由于词典是一种特殊的数据库，此参数也用于能删除词典。

VIEW:

指明为删除视图。

GROUP:

指明为删除用户组。

USER:

指明为删除用户。

INDEX:

指明为删除字段上的索引。

object_name:

待删除的对象名，删除数据库/视图时可由 **owner** 指定对象的拥有者。可使用通配符，用 '?' 匹配单个字符，用 '*' 匹配任何字符。

database_name:

删除索引时，指定索引字段所在的数据库名，可由 **user** 指定数据库的拥有者。

column_name:

删除索引时，指定索引所在的字段名。可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。

record:

删除索引的起始记录号，不指定时表示删除所有记录的索引。

【要点】

1. 本语句不能删除字段和数据格式。要删除字段请用 ALTER 语句，数据格式不能被删除。
2. 一次可以删除多个对象，或删除多个字段上的索引。
3. 数据库/视图被删除后，数据库/视图上所有的数据文件、索引文件和权限信息都被删除，和数据库有关的视图也被删除。
4. 删除一个用户组以后，该组里的所有用户的所在组名将变为空。
5. 字段索引被删除后，在此字段上不能再进行检索。一般情况下，索引由系统自动维护，只在如下几种情况下才需要手动删除索引：
 - 在此字段上不需按条件进行查询；
 - 为了进行空间整理，临时删除字段上的索引
 - 字段索引文件被破坏。

【举例】

TRSL> **DROP DATABASE** *pub.newsbase*

这条语句的作用是删除数据库 *pub.newsbase*。

TRSL> **DROP USER** *tom, jenny*

这条语句的作用是删除用户 *tom* 和 *jenny*。

TRSL> **DROP INDEX** *patent(标题, 摘要)*

这条语句的作用是删除数据库 *patent* 的 *标题*和*摘要*字段上的索引。

【参考】

[CREATE 创建对象](#)

[ALTER 修改对象](#)

[RENAME 为对象改名](#)

3.3.10 ECHO 显示控制

【目的】

对 TRSL 语句执行结果信息的显示进行控制。

【前提】

无。

【语法】

ECHO { ON• OFF }

【参数】

ON:

显示执行过程和结果。

OFF:

不显示执行过程和结果。

【要点】

1. 当指定为 **ON** 时，TRSL 语句执行结果的提示显示在屏幕上，如建立数据库，建立成功或失败均有显示，为 **OFF** 则不显示。缺省值是 **ON**。
2. 本语句单独在 TRSL 窗口中使用没有什么意义，但在运行 TRSL 脚本文件时，如果 **ECHO** 语句和 **SPOOL** 语句配合使用，可以不显示脚本文件的运行过程，但执行的过程和结果全部输出到外部文件中，适合于长时间的脚本文件运行操作。

【举例】

以下是一个简单的 TRS 脚本文件的内容：

```
ECHO OFF;  
SPOOL batch.log;  
SELECT * FROM newsbase WHERE 检索;  
SELECT * FROM newsbase WHERE 经济 * 日期>=1993;  
SPOOL OFF;  
ECHO ON;
```

这个脚本文件的作用是关闭屏幕显示，并把其后两次检索的结果写到 *batch.log* 文件里，最后恢复屏幕显示。

【参考】

[SPOOL 输出到文件](#)

3.3.11 EDIT 编辑检索结果

【目的】

对检索结果进行编辑。

【前提】

当前存在检索结果记录集，并且操作用户对所编辑的字段有 **SELECT** 权限。

【语法】

```
EDIT [RECORD {ALL | {{n [, ...]} [TILL m]} [, ...]]  
      [COLUMN {ALL | {column_name [, ...]}]}  
      [FORMAT [NORMAL | ISO2709 | XML] [DOCID] [PREFIX prefix]]
```

【参数】

RECORD:

选择需要编辑的记录号(或记录号集合)，用 **TILL** 关键字连接两个记录号可指定一定范围的记录。取值 **ALL** 表示编辑所有的命中记录，为缺省值。

COLUMN:

选择编辑的字段名(或字段名集合)，取值 **ALL** 表示编辑所有的字段，为缺省值。

FORMAT:

指明以何种格式编辑数据库记录。**NORMAL** 是缺省的编辑方式，如果要以 **TRS** 格式文件中 **TRS** 记录格式来编辑，为默认格式。如果要以 **ISO2709** 或 **XML** 格式文件进行发送，取值为 **ISO2709/XML**。**DOCID** 表示加上记录的物理记录号，以支持恢复数据镜像。

PREFIX 表示添加记录前缀。

【要点】

1. 当不指定 **COLUMN**、**RECORD** 参数时，编辑所有的命中结果。
2. *column_name* 为需编辑的字段名，当 **EDIT** 后不带字段选择参数时，将编辑所有字段。
3. 当指定需要编辑的记录时，所指定的记录号是命中的记录号，不是记录的内部序号，当需要按内部记录号进行查询时，可以使用以下 **TRSL** 语句：

```
TRSL> SELECT FROM 库名 WHERE DOCID=docid
```

4. **EDIT** 语句使用的缺省编辑器是 Windows 的写字器(WordPad)，文件名为 *default.doc*，如果数据中有非 ASCII 数据(多媒体数据或复合文档)，则不处理这些数据。如果用户需要使用其它的编辑器，则先用 **SAVE** 语句将结果存于文件中，然后再进行编辑。

【举例】

TRSL 6.5

```
TRSL> SELECT FROM newsbase WHERE 中国 */sen 经济改革
```

```
TRSL> EDIT RECORD 1,2,3,4
```

上面第一条语句的作用是对 *newsbase* 数据库进行查询，第二条语句的作用是编辑前四个命中记录。

```
TRSL> SELECT FROM newsbase,hlm WHERE 宝玉
```

```
TRSL> EDIT COLUMN tx RECORD 1
```

上面第一条语句的作用是在两个数据库 *newsbase* 和 *hlm* 上进行查询，第二条语句的作用是编辑命中的第一条记录的 *tx* 字段。

【参考】

[SELECT 数据库检索](#)

[BROWSE 浏览检索结果](#)

[SAVE 存贮检索结果](#)

[PRINT 打印检索结果](#)

[MAIL 邮寄检索结果](#)

3.3.12 EXIT 退出命令语言处理工具

【目的】

退出 TRS 命令语言处理工具。

【前提】

无。

【语法】

EXIT

【参数】

无。

【要点】

1. 用户使用 **EXIT** 从 TRSL 中退出后，操作日志和检索历史将被保留。
2. TRSL 的 **QUIT** 语句也可以退出命令语言处理工具，**EXIT** 语句和 **QUIT** 语句的区别在于 **EXIT** 语句保留操作日志，而 **QUIT** 语句将删除操作日志。
3. 退出 TRSL 时会自动清除当前的检索结果，并断开与当前服务器的连接。

【举例】

```
TRSL> EXIT
```

【参考】

[QUIT 退出](#)

3.3.13 GRANT 给用户授权

【目的】

把数据库、视图、词典、字段的操作权限授予用户组/用户。

【前提】

当前操作用户必须是 DBA 用户，或是指定对象所有者所在组的 GDA 用户，或是指定对象的拥有者。

【语法】

```
GRANT [ALL] | {ALTER, DROP, INDEX, SELECT, UPDATE}]  
    TO {{user_name | group_name} [, ...]}  
    ON [owner.]object_name [({column_name [, ...]})]
```

【参数】

ALL/ALTER/DROP/INDEX/SELECT/UPDATE:

授予的权限类型，对不同的对象有不同的取值范围。对数据库可为 **SELECT**、**UPDATE**、**INDEX**、**ALTER**、**DROP**，对视图可为 **SELECT**、**ALTER**、**DROP**，对词典可为 **SELECT**、**UPDATE**、**DROP**，对字段只能为 **SELECT**。指定为 **ALL** 表示所有权限，为缺省设置。为了提高效率，若授予的权限对象不支持，将不会报错，只是当请求发送到 **SERVER** 端后不支持的权限标志将会被忽略。

user_name/group_name:

待授予权限的用户/用户组。用户组和用户名均可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。其中“*”表示所有用户组，“用户组.*”表示该用户组里的所有用户，“*. *”表示所有用户。

object_name:

授予权限的目标对象，可用 *user* 指定对象的拥有者。

column_name:

待授予权限的字段名，不指定时表示对对象授权。可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。

【要点】

1. 一次可以给多个用户组/用户授权，授权的目标数据库、视图、词典只能有一个，但可同时针对多个字段授予权限。
2. 对 DBA 用户、对象所有者所在组的 GDA 用户、或指定对象的拥有者授权没有实际意义。

【举例】

```
TRSL> GRANT SELECT, UPDATE TO tom, product.* ON newsbase
```

这条语句的作用是把数据库 *newsbase* 的 **SELECT** 和 **UPDATE** 权限授予用户 *tom* 和 *product* 组里的所有用户。

```
TRSL> GRANT TO *.* ON personnel(resume)
```

这条语句的作用是把数据库 *personnel* 的字段 *resume* 上的所有权限授予所有用户。

【参考】

[REVOKE 收回用户的权限](#)

3.3.14 HELP 显示帮助信息

【目的】

显示 TRSL 的帮助信息。

【前提】

无。

【语法】

HELP [keyword]

【参数】

keyword

需要查看帮助信息的语句，如不指定则显示语法概述和语句列表。

【要点】

无。

【举例】

TRSL> **HELP** *create*

这条语句的作用是显示 *create* 语句的帮助信息。

【参考】

无。

3.3.15 INSERT 插入新记录

【目的】

向数据库中插入新的记录。

【前提】

操作用户对数据库有 UPDATE 权限。

【语法】

```
INSERT INTO {[owner.]database_name | [owner.]auto_view_name} [(column_name [, ...]  
VALUES ({<value> [, ...]})  
[SEPARATOR separator]  
[PASSWORD {password | NULL}]  
[GROUP {group_name | NULL}]  
[SECURITY security_code]  
[CONTROL control_file]
```

【参数】

owner:

目标数据库的拥有者。

database_name/auto_view_name:

目标数据库/自动模式视图名。

column_name:

字段名。如果不指定字段名列表，则表示对词典进行操作。

VALUES:

字段值，与字段名列表里指定的字段一一对应。

SEPARATOR:

多值字段的分隔符。为字符串类型，仅对 DATE、NUMBER 和 CHAR 类型字段有效，缺省为';'。

PASSWORD:

记录的访问密码，取该记录内容时必须提供正确的密码。

GROUP:

记录对外开放的用户组。

SECURITY:

赋予记录的安全级别，访问这些记录时操作者的安全级别必须不大于此安全级别。取值范围为 0 到 0xFFFFFFFF，缺省为零，表示访问此记录无须经过安全级别检查。

CONTROL:

加载控制的文件名。

【要点】

1. 本语句每次只能插入一个记录，要批量加载记录请使用 **LOAD** 语句。
2. 插入新记录时，如果字段上已经建立了索引，则索引会自动调整，但如果数据库字段上没有建立索引，则不调整索引。这和 **LOAD** 语句不一样，对 **LOAD** 语句来说，如果数据库的同步索引控制开关为 **ON**，则自动调整数据库字段索引，无论用户是否已经在数据库字段上创建了索引。
3. 使用本语句同样可以向词典中插入词语。对于主题词典、同义词典和反义词典，用法和数据库一致；对于其它词典，则直接在 **VALUES** 后面指定多个要添加的词语就可以了，无须且不能指定字段名。

【举例】

```
TRSL> INSERT INTO personnel(name,age,resume) VALUES('Scott',27,'@scott.res')
```

这条语句的作用是向数据库 *personnel* 中插装入一条记录。符号@表示从文件中获得字段数据。

```
TRSL> INSERT INTO newbase(正文) VALUES('@test.doc|test.txt^WORD') PASSWORD  
test
```

这条语句的作用是在数据库 *newbase* 中插入一条记录，并把该记录的密码设为 *test*。字段值对应 *newbase* 的头一个字段，符号@表示从文件中获得数据，原始数据文件名为 *test.doc*，它是一 MS WORD 格式的文件，*test.txt* 为从 *test.doc* 转换后的 ASCII 文本文件。

```
TRSL> INSERT INTO mysegdic VALUES(中关村,亚运村)
```

这条语句的作用是向词典 *mysegdic* 中插装入两条词语。

【参考】

[LOAD 把记录装进数据库](#)

[DELETE 删除数据库记录](#)

[UPDATE 更新数据库记录](#)

3.3.16 LOAD 把记录装进数据库

【目的】

把 TRS 装库文件里的数据装到目标数据库中。

【前提】

操作用户必须对指定数据库有 UPDATE 权限。

【语法】

```
LOAD [BATCH] source_file
      [THROUGHT] INTO {[owner.]database_name | [owner.]auto_view_name}
      [FORMAT format]
      [CONTROL control_file]
```

【参数】

source_file:

装库数据文件或批装库文件名，文件的类型默认为单 TRS 装库数据文件，若指明为 BATCH 则表示 批 TRS 装库文件。

THROUGHT:

允许以“直通模式”加载数据。如果数据加载所涉及到所有原始数据文件、批处理文件、控制文件，服务器的启动帐号都有读取权限，则可使用直通模式进行数据加载。由于直通模式下服务器直接读取原始数据文件，从而减少了中间环节，所以效率会更高。

owner:

装库的目标数据库的拥有者。

database_name/owner.]auto_view_name:

装库的目标数据库\自动模式视图名。

FORMAT:

装库文件格式。

CONTROL:

装库控制文件名，不指定时由系统决定装库参数的值。

【要点】

1. load_file 为被装库的数据文件名，当选用 TRS 模式装库时，该文件的内容为一条或多条 TRS 记录；当选用 BATCH 模式装库时，该文件中的每一行为一个文件名，其内容是一条或多条 TRS 记录。文件名可以带路径，也可以在装库控制文件中由 DATA_PATH_DEFAULT 参数来指定。如果没有指定装库控制文件，则由系统决定装库参数的缺省值。

2. 本语句除了可以向数据库中加载记录以外，还可以向词典里加载记录。每次可装入一条记录或一批记录。

3. 有关 TRS 装库数据文件和装库控制文件的格式请参见《数据库加载格式规范》。

【举例】

```
TRSL> LOAD e:\demo.trsl INTO demo1
```

这条语句的作用是把一个 TRS 装库数据文件 demo.trsl 装入数据库 demo1 中。

```
TRSL> LOAD BATCH e:\temp\trsl_gp.tbl INTO trsl_group CFILE trsl_gp.ctr
```

这条语句的作用是把一个 TRS 批装库文件 trsl_gp.tbl 里包含的所有 TRS 数据文件装入数据库 trsl_group 中，装库参数由 trsl_gp.ctr 控制文件指定。

【参考】

[COPY 复制数据库记录](#)

[INSERT 插入新记录](#)

[DELETE 删除数据库记录](#)

[UPDATE 更新数据库记录](#)

3.3.17 LOCK 锁定数据库

【目的】

对数据库进行锁定操作。

【前提】

当前操作用户必须是 DBA 用户，或是数据库所有者所在组的 GDA 用户，或是数据库的拥有者。

【语法】

```
LOCK {[owner.]database_name [, ...]}  
      IN {EXCLUSIVE | UPDATE} MODE
```

【参数】

owner:

数据库的拥有者，可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。

database_name:

指明要锁定的数据库名，可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。

EXCLUSIVE / UPDATE:

加锁的模式。分别为访问互斥锁和修改互斥锁

【要点】

1. 锁定操作是由系统自动管理的，通常情况下不需要干预。只有当数据库由于某种原因处于不正确锁定状态时，才需要使用本语句把数据库重新置为共享状态。
2. TRS 中数据库的锁定模式有下列几种：
 - ACCESS_EXCL: 访问互斥锁，当前用户独占数据库，其它用户不能对此库进行任何操作。
 - UPDATE_EXCL: 修改锁，不允许其它用户在数据库上作维护操作，但允许进行查询操作。
 - UPDATE_WAIT: 修改等待锁，一直等待直到能给数据库加上修改锁为止，访问规则同修改锁。
 - ACCESS_DENY: 这种锁表明数据库处于禁止访问状态，其它用户不能进行任何操作，也不能使用其它锁对数据库进行锁定。
3. IN SHARE MODE 子句可以省略，效果跟提供时是一样的，

【举例】

TRSL> **LOCK demo2 IN UPDATE MODE**

这条语句的作用是把 demo2 数据库设置为修改互斥状态。

【参考】

[SELECT 执行检索](#)

[INSERT 插入新记录](#)

[DELETE 删除数据库记录](#)

[UPDATE 更新数据库记录](#)

[UNLOCK 解锁数据库](#)

3.3.18 LOGIN 登录到服务器

【目的】

登录到 TRS 服务器。

【前提】

无。

【语法】

```
LOGIN [address:port]
      [USER user_name]
      [PASSWORD password]
```

【参数】

address:

服务器所在的主机，既可以是 TRS 服务器在网络中的 IP 地址（如“199.9.200.80”），也可以是服务器所在主机的名称（如“ALPHA”）。如果输入的是服务器主机的名称则要求该主机名是当前所使用的网络驱动系统的数据库文件 hosts 的一个入口项（如“192.9.200.80 alpha”）。

port:

服务器端口既可以是 TRS 服务器的 Internet 端口（如“8000”），也可以是 TRS 服务器的名称（如“trserver”）。如果输入的是服务器的名称，则要求该名称是当前所使用的网络驱动系统的数据库文件 services 的一个 TCP 协议入口项（如“trserver 8000/tcp”）。

USER:

用户的帐号名，必须是指定服务器中已经存在的用户名。

PASSWORD:

用户的口令字，区分大小写。

【要点】

1. 如果没有指定参数，则命令语言处理工具会弹出对话框提示输入服务器地址、服务器端口、用户名和口令字，并进行登录。如只指定了用户账号而没有给出服务器地址和端口，则表示用不同的用户和密码重新登录当前服务器，此时如没有连接到服务器则会报错。
2. 必须首先登录到服务器，然后才能执行其它 TRS 操作。
3. 退出命令语言处理工具时会自动断开与 TRS 服务器的连接。

【举例】

```
TRSL> LOGIN alpha:8888 ACCOUNT system PASSWORD manager  
TRSL> LOGIN
```

第一条语句直接用 *system* 用户名登录到 *alpha:8888* 服务器上，第二条语句则打开对话框进行登录操作。

【参考】

[LOGOUT](#) 注销登录。

3.3.19 LOGOUT 注销登录

【目的】

注销登录。

【前提】

必须首先登录到 TRS 服务器。

【语法】

LOGOUT

【参数】

无。

【要点】

连接终止后，不能再执行除登录和网络操作外的任何 TRS 操作。

【举例】

TRSL> **LOGOUT**

【参考】

[LOGIN 登录到服务器。](#)

3.3.20 MAIL 发送检索结果

【目的】

把当前检索结果发送给其他用户。

【前提】

当前存在检索结果记录集，并且操作用户对所发送的字段有 **SELECT** 权限。

【语法】

```
MAIL [RECORD {ALL | {{n [, ...]} [ TILL m ]} [, ...]]  
      [COLUMN {ALL | {column_name [, ...]} }]  
      [FORMAT {PLAIN | NORMAL | ISO2709 | XML} [DOCID] [PREFIX prefix]]  
      [SUBJECT mail_subject]  
      [TO {{user_name | mail_box} [, ...]}]  
      [CC {{user_name | mail_box} [, ...]}]
```

【参数】

RECORD:

选择需要发送的记录号(或记录号集合), 用 **TILL** 关键字连接两个记录号可指定一定范围的记录。取值 **ALL** 表示发送所有的命中记录, 为缺省值。

COLUMN:

选择发送的字段名(或字段名集合), 取值 **ALL** 表示发送所有的字段, 为缺省值。

FORMAT:

指明以何种格式发送数据库记录。**PLAIN** 表示记录文件格式; **NORMAL** 表示 TRS 装库文件格式, 为默认格式; 如果要以 **ISO2709** 或 **XML** 格式文件进行发送, 取值为 **ISO2709/XML**。**DOCID** 表示加上记录的物理记录号, 以支持恢复数据镜像。**PREFIX** 表示添加记录前缀。

SUBJECT:

选邮件的主题。

TO:

邮件接收者。

CC:

邮件的抄送者。

【要点】

1. 当不指定 **COLUMN**、**RECORD** 参数时, 发送所有的命中结果。

2. *column_name* 为需发送的字段名，当 **MAIL** 后不带字段选择参数时，将发送所有字段。
3. 当指定需要发送的记录时，所指定的记录号是命中的记录号，不是记录的内部顺序号，当需要按内部记录号进行查询时，可以使用以下 TRSL 语句：

```
TRSL> SELECT FROM 库名 WHERE DOCID=docid
```

4. 发送检索结果时，记录中的多媒体数据（如图象）、格式化文件（如 WPS 文件）、复合文档（MS WORD 文件）等将被滤掉。

5. MAIL 语句执行结束后，检索结果会以附件的形式放入邮件中等待发送。

【举例】

```
TRSL> SELECT FROM newsbase WHERE 中国 */sen 经济改革
```

```
TRSL> MAIL RECORD 1,2,3,4
```

上面第一条语句的作用是进行查询，第二条语句的作用是把命中记录集中的前四个记录发送到 *tom@263.net*。

```
TRSL> SELECT FROM newsbase,hlm WHERE 宝玉
```

```
TRSL> MAIL DATABASE tx RECORD 1 FORMAT ISO2709
```

上面第一条语句的作用是在两个数据库 *newsbase* 和 *hlm* 上进行查询，第二条语句的作用是把数据库 *newsbase* 中命中的第一条记录的 *tx* 字段内容发送给其他用户（由用户在邮件发送程序里指定），发送格式为 ISO2709 格式。

【参考】

[SELECT 数据库检索](#)
[BROWSE 浏览检索结果](#)
[EDIT 编辑检索结果](#)
[SAVE 保存检索结果](#)
[PRINT 打印检索结果](#)

3.3.21 MODIFY 修改记录

【目的】

以数据文件方式更新记录。

【前提】

当前操作者必须对目标数据库/自动模式视图/词典有 UPDATE 权。

【语法】

```
MODIFY [INCREMENT] {[owner.]database_name | [owner.]auto_view_name}
[({column_name [, ...])}]
  [THROUGHT] JOIN source_file [BY {DOCID | column_name}]
  [SEPARATOR separator]
  [PASSWORD {password | NULL}]
  [GROUP {group_name | NULL}]
  [SECURITY security_code]
  [FORMAT format]
  [CONTROL control_file]
```

【参数】

INCREMENT:

表示批量更新记录时，允许源记录格式文件中出现的新记录。

owner:

目标数据库/自动模式视图/词典的拥有者。

database_name/auto_view_name:

目标数据库/自动模式视图/词典名。

THROUGHT:

允许以“直通模式”加载数据。如果数据加载所涉及所有原始数据文件、批处理文件、控制文件，服务器的启动帐号都有读取权限，则可使用直通模式进行数据加载。由于直通模式下服务器直接读取原始数据文件，从而减少了中间环节，所以效率会更高。注意，此处用到的 `source_file` 在生成时必须指定生成格式为 `DOCID`。

JOIN:

更新源数据库文件及更新条件。`source_file` 表示更新的源数据库文件。`DOCID` 表示以指定物理记录号列表的方式更新，`column_name` 为指定的字段名。

BY:

用于指定作为记录映射的字段名。

SEPARATOR:

多值字段的分隔符。为字符串类型，仅对 DATE、NUMBER 和 CHAR 类型字段有效，缺省为';'。

PASSWORD:

记录的访问密码。

GROUP:

记录对外开放的用户组。

SECURITY:

赋予记录的安全级别，访问这些记录时操作者的安全级别必须不大于此安全级别。取值范围为 0 到 0xFFFFFFFF，缺省为零，表示访问此记录无须经过安全级别检查。

FORMAT:

更新源数据库文件的格式。

CONTROL:

加载控制文件名。

【要点】

无。

【举例】

```
TRSL> MODIFY demo2 JOIN C:\\DEMO.DAT
```

上面第一条语句的作用是以数据文件 DEMO.DAT 的内容更新数据库 demo2 中的记录。

【参考】

无。

3.3.22 NOTEPAD 编辑文件

【目的】

调用 Windows 的记事本 NOTEPAD 编辑文件。

【前提】

Windows 下有 NOTEPAD 应用程序。

【语法】

NOTEPAD [file_name]

【参数】

file_name:

用户需要编辑的文件名，不带路径时表示为当前目录。如果用户没有指定文件名，则新建一个空文件。

【要点】

1. **NOTEPAD** 语句一般用来编辑 TRS 脚本文件、TRS 装库控制文件等。
2. 用户可以在其它 WINDOWS 编辑器中编辑文件，但 TRSL 的 **RUN** 语句可执行的必须是 ASCII 文本文件。

【举例】

```
TRSL> NOTEPAD
```

```
TRSL> NOTEPAD test.trs
```

第一条 TRSL 语句没有带文件名，Windows 将打开一个空的记事本；第二条语句是编辑文件 *test.trs*。

【参考】

无。

3.3.23 OPTIMIZE 优化数据库

【目的】

对数据库进行优化，提高查询速度并整理空间。

【前提】

操作用户必须对指定数据库有 INDEX 权限。

【语法】

```
OPTIMIZE {[owner.]database_name | [owner.]auto_view_name} [({column_name [, ...])]  
[TEMP temp_path]
```

【参数】

owner:

待优化的数据库的拥有者。

database_name/auto_view_name:

待优化的数据库名。

column_name:

待优化的字段名，可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。不指定字段列表时表示优化所有字段。

TEMP:

优化时使用的临时工作空间路径，必须是服务器配置中包含的数据路径。不指定时自动取系统缺省的数据路径。

【要点】

1. 数据库优化时处于 UPDATE 锁定状态，其他 TRS 用户不能对此数据库进行维护操作，但可以进行查询。
2. 优化操作需要较多的磁盘空间，因此指定临时目录时，应该确认服务器上有足够的磁盘空间。
3. 由于优化操作需要较长的时间和占有较多的机器资源，建议在作业较少时进行数据库优化操作。
4. 系统能够自动检测数据库的数据文件和索引文件是否需要优化，并直接跳过那些不需要优化的数据和索引文件。

【举例】

TRSL> **OPTIMIZE** *newsbase*(作者, 正文) **ON** /trshome/ibase

这条语句的作用是在/trshome/ibase 目录下对数据库 *newsbase* 的“作者”和“正文”字段进行优化。

【参考】

[CREATE](#) 创建对象

[DROP](#) 删除对象

[VALIDATE](#) 验证索引

3.3.24 PRINT 打印检索结果

【目的】

打印当前检索结果。

【前提】

当前存在检索结果记录集，并且操作用户对所打印的字段有 **SELECT** 权限。

【语法】

```
PRINT [RECORD {ALL | {{n [, ...]} [TILL m]} [, ...]]  
      [COLUMN {ALL | {column_name [, ...]} }]  
      [FORMAT [PLAIN | NORMAL | ISO2709 | XML] [DOCID] [PREFIX prefix]]
```

【参数】

RECORD:

选择需要打印的记录号(或记录号集合),用 **TILL** 关键字连接两个记录号可指定一定范围的记录。取值 **ALL** 表示打印所有的命中记录,为缺省值。

COLUMN:

选择打印的字段名(或字段名集合),取值 **ALL** 表示打印所有的字段,为缺省值。

FORMAT:

指明以何种格式打印数据库记录。**PLAIN** 表示记录文件格式;**NORMAL** 表示 TRS 装库文件格式,为默认格式;如果要以 **ISO2709** 或 **XML** 格式文件进行发送,取值为 **ISO2709/XML**。**DOCID** 表示加上记录的物理记录号,以支持恢复数据镜像。**PREFIX** 表示添加记录前缀。

【要点】

1. 当不指定 **DATABASE**、**COLUMN**、**RECORD** 参数时,打印所有的命中结果。
2. *column_name* 为需打印的字段名,当 **BROWSE** 后不带字段选择参数时,将打印所有字段。
3. 当指定需要打印的记录时,所指定的记录号是命中的记录号,不是记录的内部顺序号,当需要按内部记录号进行查询时,可以使用以下 TRSL 语句:

```
TRSL> SELECT FROM 库名 WHERE DOCID=docid
```
4. 当前必须配置了可用的打印机,且打印机处于准备状态。
5. 在打印检索结果时,记录中的多媒体数据(如图象)、格式化文件(如 **WPS** 文件)、复

合文档（MS WORD 文件）等将不被打印。

【举例】

```
TRSL> SELECT FROM newsbase WHERE 中国 */sen 经济改革
```

```
TRSL> PRINT RECORD 1,2,3,4
```

上面第一条语句的作用是对 *newsbase* 数据库进行查询，第二条语句的作用是打印命中记录集中的前四条记录。

```
TRSL> SELECT FROM newsbase, hlm WHERE 宝玉
```

```
TRSL> PRINT DATABASE tx RECORD 1
```

上面第一条语句的作用是在两个数据库 *newsbase* 和 *hlm* 上进行查询，第二条语句的作用是打印在命中的第一条记录的 *tx* 字段内容。

【参考】

[SELECT 数据库检索](#)

[BROWSE 浏览检索结果](#)

[EDIT 编辑检索结果](#)

[SAVE 保存检索结果](#)

[MAIL 邮寄检索结果](#)

3.3.25 QUIT 退出并清除操作日志

【目的】

退出命令语言处理工具，并清除操作日志。

【前提】

无。

【语法】

QUIT

【参数】

无。

【要点】

1. 用户使用 **QUIT** 从 TRSL 中退出后，操作日志将被删除，检索历史将保留。
2. TRSL 的 **EXIT** 语句也可以从 TRSL 窗口中退出，**QUIT** 语句和 **EXIT** 语句的区别在于 **EXIT** 语句保留操作日志，而 **QUIT** 语句将删除操作日志。
3. 退出 TRSL 时会自动清除当前的检索结果，并断开与当前服务器的连接。

【举例】

```
TRSL> QUIT
```

【参考】

[EXIT 退出](#)

3.3.26 RECOVER 恢复数据库

【目的】

从指定文件或设备恢复数据库。

【前提】

当前用户必须具备 Resource 权限。

【语法】

```
RECOVER [ALL | {[owner.]database_name [, ...]} | [owner.]auto_view_name]
FROM backup_file
[TO data_path]
```

【参数】

ALL:

表示恢复指定备份包里的所有数据库，为缺省设置。当只恢复某个或某些数据库时，应列出具体的数据库名，一次可以恢复多个数据库。

owner:

数据库的拥有者，可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。

database_name/auto_view_name:

待恢复的数据库名，可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。

FROM:

备份包所在的设备名，可以是服务器端的文件名或磁带名。不指定时表示从系统缺省备份设备恢复。

TO:

数据库恢复后的路径，必须是服务器配置中包含的数据路径，不指定时表示恢复到系统缺省的数据路径。

【要点】

1. 如果被恢复的数据库在系统中不存在，系统将把这个库加到 TRS 数据库中；如果被恢复的数据库在系统中已存在，则自动更换库名（更换后的库名由原来的库名后加一编号构成）。如果要使用原来的数据库名，则需先删除或重命名原有的数据库，并把恢复后的数据库改为原来的名字。
2. 由于权限信息不进行备份，因此数据库恢复后也不存在授权信息（但数据库所有者除外）。

【举例】

TRSL> **RECOVER** newsbase **FROM** /dev/rst0 **TO** /trshome/ibase

这条语句的作用是把备份集/dev/rst0 中的 newsbase 数据库恢复到/trshome/ibase 目录下。

TRSL> **RECOVER ALL FROM** /trshome/temp/demo.bak

这条语句的作用是把/trshome/temp/demo.bak 备份包里的所有数据库恢复到系统缺省的数据路径下。

【参考】

[BACKUP 备份数据库](#)

3.3.27 REM 注释语句

【目的】

注释信息。

【前提】

无。

【语法】

REM remark

【参数】

remark:

为注释信息，内容可为除分号和单引号外的任何字符（包括空格），在一行中书写，长度无限制。字符串可用单引号括起来，也可以直接书写。

【要点】

REM 语句一般用于 TRSL 脚本文件中作为提示，执行脚本文件时仅显示，不作任何处理。

【举例】

以下是一个简单的 TRS 脚本文件的内容：

```
REM Select operation will start right now:
```

```
SELECT FROM rmrb WHERE 中国;
```

这个脚本文件的作用是先显示一条开始检索的信息，然后执行检索操作。

【参考】

无。

3.3.28 RENAME 为对象改名

【目的】

为数据库、视图、词典、用户组、用户、数据格式和字段改名。

【前提】

修改用户组、用户或数据格式名时，当前操作者必须是 **SYSTEM** 用户；为数据库/视图改名时，当前操作者必须对该对象有 **ALTER** 权限；修改字段名时，当前操作者必须对字段所在的数据库/视图有 **ALTER** 权限。

【语法】

1. 修改数据库或字段名

```
RENAME DATABASE [old_owner.]database_old_name  
    TO [new_owner.]database_new_name
```

```
RENAME DATABASE [owner.]database_name (column_old_name)  
    TO column_new_name
```

2. 修改数据格式名

```
RENAME FORMAT format_old_name  
    TO format_new_name
```

3. 修改用户组名

```
RENAME GROUP group_old_name  
    TO group_new_name
```

4. 修改用户名

```
RENAME USER user_old_name  
    TO user_new_name
```

5. 修改视图或字段名

```
RENAME VIEW [old_owner.]view_old_name  
    TO [new_owner.]view_new_name
```

```
RENAME VIEW [owner.]view_name (column_old_name)  
    TO column_new_name
```

【参数】

DATABASE:

指明为数据库或者数据库字段改名。由于词典是一种特殊的数据库，此参数也用于能修改词典名。old_owner 指数据库原来的拥有者，new_owner 指数据库修改后的拥有者；

`database_old_name` 指数据库的旧名字, `database_new_name` 指数据库的新名字。
`database_name.column_old_name` 指旧的字段名, `column_new_name` 指新的字段名。

VIEW:

指明为视图或者视图字段改名。`old_owner` 指原来的视图的拥有者, `new_owner` 指修改后的视图的拥有者; `view_old_name` 指视图的旧名字, `view_new_name` 指视图的新名字。
`view_name.column_old_name` 指旧的字段名, `column_new_name` 指新的字段名。

GROUP:

指明为用户组改名。`group_old_name` 指数据格式的旧名字, `group_new_name` 指数据格式的新名字。

USER:

指明为用户改名。`user_old_name` 指数据格式的旧名字, `user_new_name` 指数据格式的新名字。

FORMAT:

指明为数据格式改名。`format_old_name` 指数据格式的旧名字, `format_new_name` 指数据格式的新名字。

【要点】

对象被改名后, 和它有关的对象也进行修改。如数据库名修改后, 和数据库有关的视图等信息全部要修改, 以保证数据词典信息的一致性。

【举例】

TRSL> **RENAME** system.新闻资料数据库 **TO** pub.newsbase

这条语句的作用是把数据库新闻资料数据库改名为 newsbase, 并把此数据库的拥有者从 system 改为 pub。

TRSL> **RENAME B** abstract(摘要) **TO** science

这条语句的作用是把数据库 abstract 的字段摘要改名为 science。

TRSL> **RENAME USER** administrator **TO** admin

这条语句的作用是把用户 administrator 改名为 admin。

【参考】

- [CREATE 创建对象](#)
- [ALTER 修改对象](#)
- [DROP 删除对象](#)

3.3.29 REVOKE 收回用户的权限

【目的】

收回用户组/用户对数据库、视图、词典、字段上的操作权限。

【前提】

当前操作用户必须是 DBA 用户，或是指定对象所有者所在组的 GDA 用户，或是指定对象的拥有者。

【语法】

```
REVOKE [ALL | {ALTER, DROP, INDEX, SELECT, UPDATE}]  
FROM {{user_name | group_name} [, ...]}  
ON [owner.]object_name [({column_name [, ...])}]
```

【参数】

ALL /ALTER/DROP/INDEX/ SELECT/UPDAT:

回收的权限类型，对不同的对象有不同的取值范围。对数据库可为 **SELECT**、**UPDATE**、**INDEX**、**ALTER**、**DROP**，对视图可为 **SELECT**、**ALTER**、**DROP**，对词典可为 **SELECT**、**UPDATE**、**DROP**，对字段只能为 **SELECT**。指定为 **ALL** 表示所有权限，为缺省设置。为了提高效率，若收回的权限对象不支持，将不会报错，只是当请求发送到 **SERVER** 端后不支持的权限标志将会被忽略。

user_name/group_name:

待回收权限的用户/用户组。用户组和用户名均可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。其中“*”表示所有用户组，“用户组.*”表示该用户组里的所有用户，“*.*”表示所有用户。

object_name:

回收权限的目标对象，可用 *user* 指定对象的拥有者。

column_name:

待回收权限的字段名，不指定时表示收回对象的权限。可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。

【要点】

1. 一次可以回收多个用户组/用户的权限，回收权限的目标数据库、视图、词典只能有一个，但可同时针对多个字段回收权限。
2. DBA 用户、对象所有者所在组的 GDA 用户、和指定对象的拥有者对对象的操作权限不能被收回。

【举例】

```
TRSL> REVOKE SELECT, ALTER FROM tom ON newsbase
```

这条语句的作用是把数据库 *newsbase* 的 **SELECT** 和 **ALTER** 权限从用户 *tom* 收回。

```
TRSL> REVOKE FROM * ON personnel(resume)
```

这条语句的作用是把数据库 *personnel* 的字段 *resume* 上的所有权限从所有用户组收回。

【参考】

[GRANT 收回用户的权限](#)

3.3.30 RUN 运行脚本文件

【目的】

批处理运行 TRS 语句。

【前提】

指定的脚本文件存在。

【语法】

RUN [script_file]

【参数】

script_file:

脚本文件名，不带路径时表示为当前目录。用户不指定脚本文件名时，命令语言处理工具会弹出对话框让用户选择文件。

【要点】

1. TRS 中用户可以以一个文件的形式编辑要执行的 TRSL 语句，文件中可以存放数量不限的 TRSL 语句（包括可执行语句如 **SELECT**、**INSERT** 等及非执行语句如 **REM**、**SPOOL** 等），每个 TRSL 语句必须以单字节分号(;)结束，最后一个 TRSL 语句可以没有分号结束符，这个文件在 TRS 中被称为脚本文件。脚本文件提供了以批方式操纵 TRS 数据库的灵活手段。
2. 脚本文件为客户端的 DOS 文件名，可以带路径。

【举例】

下面是一个 TRSL 脚本文件 demo.trsl 的例子，其内容为：

```
SELECT * FROM newsbase WHERE 中国 */sen 经济 */sen 改革;  
SAVE RECORD 1 TILL 10 TO myresult1.dat;  
SELECT * FROM hlm WHERE 黛玉 * 葬花;  
SAVE TO myresult2.dat;
```

运行上述 TRSL 脚本文件的语句为：

```
TRSL> RUN demo.trsl
```

【参考】

[NOTEPAD 编辑文件](#)

3.3.31 SAVE 保存检索结果

【目的】

把当前的检索结果保存到文件里。

【前提】

当前存在检索结果记录集，并且操作用户对所保存的字段有 SELECT 权限。

【语法】

```
SAVE [RECORD {ALL | {{n [, ...]} [ TILL m ]} [, ...]]  
      [COLUMN {ALL | {column_name [, ...]}]}  
      [FORMAT [PLAIN | NORMAL | ISO2709 | XML | BINARY] [DOCID] [PREFIX  
prefix]]  
      [AFFIX [source_path [, file_count]] [IN {NULL | DEFAULT | ALONE | CLASS | LINK  
| DUMB}]]  
      [DIVIDE max_count [, max_size]]  
      [BATCH batch_count]  
      [TO file_name]
```

【参数】

RECORD:

选择需要保存的记录号(或记录号集合)，用 **TILL** 关键字连接两个记录号可指定一定范围的记录。取值 **ALL** 表示保存所有的命中记录，为缺省值。

COLUMN:

选择保存的字段名(或字段名集合)，取值 **ALL** 表示保存所有的字段，为缺省值。

FORMAT:

指明以何种格式保存数据库记录。**PLAIN** 表示记录文件格式；**NORMAL** 表示 TRS 装库文件格式，为默认格式；如果要以 **ISO2709** 或 **XML** 格式文件进行发送，取值为 **ISO2709/XML**。**DOCID** 表示加上记录的物理记录号，以支持恢复数据镜像。**PREFIX** 表示添加记录前缀。

AFFIX:

附件目录和每组存放文件数。**source_path** 指定记录所含附件存放的目录。**file_count** 指定每组存放的文件数，0 或者不设置表示不限制，最大值为 FFFFFFFD。

IN:

保存附件的方式。**DEFAULT** 表示按缺省方式保存源文件；**ALONE** 表示以独立的文件形式存储；**CLASS** 表示按字段类型分类存储；**LINK** 表示存储文件名所指定的 URL；**DUMB**

表示只存储文件信息，不存储数据。

DIVIDE:

每个文件允许存放的最大记录数和尺寸。**max_count** 表示记录附件存放的目录中，每组容纳多少条记录的附件，最大值为 32768；若为 0 或者不指定数值则表示不限制，即所有附件全部直接存入指定的文件中；若指定了数值比如为 10，则生成若干个文件分别存放附件，每个文件中存储 10 条记录的附件。**max_size** 指设置保存的目标文件的最大长度，不设置或为零时取缺省值 2G。若当前所要保存的记录大小超过此数值，将自动分为 N 个大小为此数值的目标文件进行保存，如 Demo2_1.trs、Demo2_2.trs...

BATCH:

指定每批读取的记录数。

TO:

结果文件名，为客户端的文件名，可以带路径，不指定文件名时将报错。

【要点】

1. 当不指定 **COLUMN**、**RECORD** 参数时，保存所有的命中结果。
2. *column_name* 为需保存的字段名，当 **SAVE** 后不带字段选择参数时，将保存所有字段。
3. 当指定需要保存的记录时，所指定的记录号是命中的记录号，不是记录的内部序号，当需要按内部记录号进行查询时，可以使用以下 TRSL 语句：

```
TRSL> SELECT FROM 库名 WHERE DOCID=docid
```

4. 对于有 **BIT** 字段的记录在存储时，**BIT** 字段将单独作为文件存放，在记录文件中的 **BIT** 字段处写入该字段存放的文件名。
5. 如果您发现无法存贮检索结果，这可能是没有足够的磁盘空间。

【举例】

```
TRSL> SAVE TO tmp.dat
```

不带任何选项的这条语句表示存贮所有的检索结果到文件 *tmp.dat* 中。

```
TRSL> SAVE COLUMN name RECORD 1,3 TILL 5,9 TO tmp.dat
```

表示存贮检索结果到 *tmp.dat* 文件中，存贮字段为 *name*，存贮的命中记录为 1, 3, 4, 5, 9 共 5 个记录。

【参考】

- [SELECT 数据库检索](#)
- [BROWSE 浏览检索结果](#)
- [EDIT 编辑检索结果](#)
- [PRINT 打印检索结果](#)
- [MAIL 邮寄检索结果](#)

3.3.32 SELECT 对数据库/视图进行检索

【目的】

对数据库/视图进行检索，为 TRS 用户最常用的操作。

【前提】

当前操作者对被检索的数据库/视图有 SELECT 权限。

【语法】

```
SELECT [* | COUNT(*) | {stat_expression [, ...]}]
      FROM {[owner.]object_name [, ...]}
      [WHERE search_expression [LOWFREQ] [BANSTOP] [NONWORD]]
      [ORDER BY [{RELEVANCE | {column_name [ASC | DESC}}] [, ...] [LIFO]
      [{PRUNE | VALID} record_number] [MIXED] [MISCOL] [NOCASE] [ALWAYS]
      [RELEVANT {NULL | NORMAL | GLOBAL | NORMPOS | GLOBPOS | WORDS |
WFREQ}]
      [HITPOINT {NULL | EXACT | IGNORE | INEXACT}]
      [DEFCOL {column_name [, ...]} [[AS] SINGLE] [[IN] {AND | OR | NOT | XOR}]
      [QUICKEND quickend_prune]
      [RETRYCOL {NULL | DEFAULT | DUMMY | column_name}]
      [KNOWASS {NULL | {ST, AT, PT, UF, AB, ABF, RT, BT, NT, ALLBT, ALLNT, LE,
TXS, FXH}}]
      [THESAURUS {thesaurus_dic | NULL}]
      [SYNONYM {synonym_dic | NULL}]
      [ANTONYM {antonym_dic | NULL}]
```

【参数】

stat_expression:

统计结果表达式，多个表达式之间以逗号分隔，可省略。在 TRS 中，选择输出的字段在结果处理语句中定义，因此这里不能使用字段名。当只有 **COUNT**(*)这一个统计表达式时，**COUNT**(*)将被忽略，当与其他统计表达一起出现时，**COUNT**(*)才有效。

owner:

检索目标数据库/视图的拥有者。可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。

object_name:

检索目标数据库/视图名。可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。

search_expression:

检索条件表达式，它是由检索条件通过 TRSL 操作符和 TRSL 函数适当联结构成的。

ORDER BY:

记录的排列顺序。对每个 column_name 可用 **ASC** 或 **DESC** 分别指定按升序/降序方式排列。字段名也可作为 **RELEVANCE**，表示按相关性排序，但只能按升序排列。也可指定为 **LIFO**，表示按后进先出顺序排列。对于参数 record_number，若前面选择 **PRUNE** 时则代表排序并裁减保存的结果记录数，同时自动设置“非精确命中点”模式；选择 **VALID** 时则代表有效排序结果记录数，即部分排序，按排序表达式排序的结果只保证前一部分记录是有序的。**MIXED** 表示对检索结果进行混合统一排序。**MISCOL** 则表示对排序表达式中不存在的字段，按字段无值时情况处理。**NOCASE** 表示排序表达式中的字段值，其字符是大小写无关的。**ALWAYS** 表示在命中记录数不超过系统配置的最高限值的情况下，“总是”允许排序检索结果。

RELEVANT:

相关性的计算模式和 IDF 模式选项。

HITPOINT:

命中点的存储模式选项。**NULL** 表示由系统配置决定；**EXACT** 表示存储精确的命中点；**IGNORE** 表示检索过程中忽略命中点；**INEXACT** 表示存储非精确的命中点。

DEFCOL:

检索条件表达式中缺省字段名及逻辑。column_name 表示检索条件表达式中缺省字段的子检索表达式的目标字段名集，为 ‘*’ 表示在所有 DOCUMENT 字段和所有 PHRASE 字段上同时进行检索。**AS** 表示缺省字段检索的字段逻辑优先于检索表达式中的逻辑。多个字段之间的关系可为 **AND**、**OR**、**NOT** 或 **XOR**。

QUICKEND:

集群对后台检索结果的裁减选项。高 16 位为选项。低 16 位的意义是：0 表示不作限制；1~9 表示集群至少需要等待相应数量的后台返回结果；10~99 表示集群至少需要等待相应比例的后台返回结果；100 及以上表示至少需要返回的结果记录数。

RETRYCOL:

对于检索表达式中不存在的字段的处理方式。

KNOWASS:

进行知识扩展检索时的主题词典。后面为利用知识辅助词典自动进行知识扩展检索的选项，可为下表中的值之一，多个值之间以逗号分隔。为 **NULL** 则表示取消所有自动扩展功能。

extend_type	说明
ST	自动进行“同义词”扩展检索
AT	自动进行“反义词”扩展检索
PT	自动进行“等同词”扩展检索
UF	自动进行“替代词”扩展检索
AB	自动进行“缩略词”扩展检索
ABF	自动进行“被缩略词”扩展检索

RT	自动进行“相关词”扩展检索
BT	自动进行“上位词”扩展检索
NT	自动进行“下位词”扩展检索
ALLBT	自动进行“所有上位词”扩展检索
ALLNT	自动进行“所有下位词”扩展检索
ALLKAX	自动进行所有扩展检索
ECX	自动进行“半角/全角字符”扩展检索

THESAURUS:

进行知识扩展检索时的同义词典。

SYNONYM:

进行知识扩展检索时的反义词典。

ANTONYM:

进行知识扩展检索的选项。

【要点】

1. SELECT 语句只显示检索的命中记录数、命中点数和统计结果等信息，要浏览检索结果请用 **BROWSE** 或 **EDIT** 等检索结果处理函数。
2. 本次检索结果会一直保存到重新进行检索，或重新进行登录，或退出命令语言为止。

【举例】

TRSL> **SELECT FROM** *newsbase* **WHERE** 中国 **AND/SEG** 美国

这条语句的作用是在数据库 *newsbase* 中查找有关“中国”和“美国”在同一段中的资料。

TRSL> **SELECT FROM** *newsbase*, *rnr* **WHERE** 红学研究 **ORDER BY** 日期

MIXSORT

这条语句的作用是在数据库 *newsbase* 和数据库 *rnr* 中查找有关“红学研究”的资料，检索结果按日期对两个数据库的命中记录进行统一排序。

TRSL> **SELECT AVG**(*column1*)/10 **FROM** *Dbase* **WHERE** *column2* = '开发'

这条语句的作用是对数据库 *Dbase* 上所有 *column2* 的值等于“开发”的那些记录求字段 *column1* 上的平均值的十分之一。

【参考】

- [SORT 对检索结果重新排序](#)
- [BROWSE 浏览检索结果](#)
- [EDIT 编辑检索结果](#)
- [SAVE 保存检索结果](#)
- [PRINT 打印检索结果](#)
- [MAIL 发送检索结果](#)

3.3.33 SET 设置系统参数

【目的】

设置当前服务器的系统配置、命令语言的环境变量或用户自定义变量。

【前提】

只有 SYSTEM 用户能修改系统配置，对环境变量的修改没有限制。

【语法】

1. 维护系统配置

SET SYSTEM

```
[KNOWASS {<knowass_element> [, ...]]  
[PATH {data_path [, ...]]  
[TEMP temp_path]  
[BACKUP backup_path]  
[CHARSET {GB18030 | BIG5 | ENGLISH | UTF8}]  
[OFFLINE [max_time] [, min_time]]  
[IDLE [start_time] [, end_time]]  
[MEMORY memory_size]  
[HISTORY history]  
[DOWNLOAD download]  
[TIMEOUT search_time]  
[MAXSORT sort_record]  
[STEM {ON | OFF}]  
[CASE {ON | OFF}]  
[NUMBER {ON | OFF}]
```

knowass_element:

维护知识词典选项，其语法定义如下：

CHARSET {GB18030 | BIG5 | ENGLISH | UTF8}

```
[SEGMENT {segment_dic | NULL}]  
[STOP {stop_dic | NULL}]  
[THESAURUS {thesaurus_dic | NULL}]  
[SYNONYM {synonym_dic | NULL}]  
[ANTONYM {antonym_dic | NULL}]
```

其中的各参数说明如下：

LANGUAGE:

需要修改的知识词典集的语言，可为 **GB18030**、**BIG5**、**ENGLISH**、**UTF8** 之一。

SEGMENT:

缺省的主分词词典名，建库时作为缺省的数据库主分词词典，可为 **NULL**。

STOP:

缺省的主停用词典名，建库时作为缺省的数据库主停用词典，可为 **NULL**。

THESAURUS:

缺省的主题词典名，检索时作为缺省的自动扩展主题词典，可为 **NULL**。

SYNONYM:

缺省的同义词典名，检索时作为缺省的自动扩展同义词典，可为 **NULL**。

ANTONYM:

缺省的反义词典名，检索时作为缺省的自动扩展反义词典，可为 **NULL**。

除知识词典集的语言必须提供以外，其它词典可以不指定，此时对词典集中相应词典的配置将不作修改。

PATH:

系统允许存放数据库的目录集。目录名之间可用单字节的分号';'、逗号','、空格或制表符分隔。系统缺省值为系统安装路径下的 **ibase** 和 **dict**，不允许为空。在修改数据库的目录集时，其格式如下：

- 如果某目录名有单字节的前缀 '+'（加号），则表示在目录集中增加这个目录。
- 如果某目录名有单字节的前缀 '-'（减号），则表示从目录集中删除这个目录。
- 如果某目录名是由单字节的中缀 '*'（星号）连接的两个目录，则表示用 '*' 后面的目录替换 '*' 前面的目录。

其中，前缀和中缀符号都必须紧跟目录名，不能间隔任何目录名之间的分隔符。系统对不带任何前缀或中缀符号的目录，不做任何处理（如果是目录集中的目录，则保持不变，否则忽略该目录）。

TEMP:

系统缺省的临时文件路径。缺省为 **TRS** 安装目录下的 **temp** 目录。

BACKUP:

系统缺省的备份设备，可以是磁带名或文件名。缺省为 **TRS** 安装目录下的 **temp** 目录。

CHARSET:

系统缺省的语言环境，可为 **GB18030**、**BIG5**、**ENGLISH**、**UTF8** 之一。

max_time:

允许一个用户实例最长的空闲时间，以分计，超时将自动 **Logout**。缺省值为零，表示不限制。

TRSL 6.5

min_time:

一个用户应该保持的最小有效时间（除非 Logout），以秒计，在此期间内不允许自动 Logout。缺省值为零，表示不限制。

start_time:

当系统空闲时，允许自动进行系统或数据库维护的起始时间，以小时计，在 0 到 23 之间。24 表示禁止系统进行自动维护，缺省值为 23。

end_time:

允许自动进行系统或数据库维护的截止时间，以小时计，在 0 到 23 之间。缺省值为 24，表示不限制。

MEMORY:

服务器的物理内存总量，以兆字节计算。缺省值为 64。

HISTORY:

系统缺省的最大检索历史步数，缺省为 20，取值范围为 0 到 6553。创建用户时作为用户最大检索历史步数的缺省值。

DOWNLOAD:

系统缺省的一次最大下载记录数，缺省值为 1000，取值范围为 0 到 0xFFFFFFFF。创建用户时作为用户最大下载记录数的缺省值。

TIMEOUT:

一次检索操作的最长执行时间，以秒计算，零表示不限制，缺省值为 10。

MAXSORT:

对检索结果进行排序的最大记录数，为零表示不限制，缺省值为 10000。

STEM:

系统缺省的只建立词根索引标志，缺省为 **OFF**。

CASE:

系统缺省的大小写敏感标志，缺省为 **ON**。

NUMBER:

系统缺省的对数字进行索引标志，缺省为 **ON**。

2. 设置控制文件变量

SET CONTROL filename

```
[ADD]VARIABLE{RECORD_FILE_FORMAT=<value>,SYNC_CREATE_INDEX=<value>, ...}
```

filename:

控制文件名。

VARIABLE:

装库控制参数和值。**ADD** 表示增加变量。

3. 设置运行环境参数

SET LOCALE

[TEMP temp_path]
[BUFFER buffer_size]
[BATCH batch_count]
[SEPARATOR {separator | NULL}]
[DRAWTEXT {ON | OFF}]
[AUTOWRAP {OFF | ALL | SEGMENT}]
[URLFILE {ON | OFF}]
[QUICKEND quickend_prene]
[SORTSAVE [sort_prune] [, sort_valid]]
[RELEVANT {NULL | NORMAL | GLOBAL | NORMPOS | GLOBPOS | WORDS |
WFREQ}]
[HITPOINT {NULL | EXACT | IGNORE | INEXACT}]
[THESAURUS {thesaurus_dic | NULL}]
[SYNONYM {synonym_dic | NULL}]
[ANTONYM {antonym_dic | NULL}]
[KNOWASS {NULL | {ST, AT, PT, UF, AB, ABF, RT, BT, NT, ALLBT, ALLNT, LE,
TXS, FXH}}]
[LIST BY {NULL | {NAME | OWNER | TYPE | CREATE | COUNT | COMMENT}
[ASC | DESC}}]
[RIGHT {NULL | {ALTER, DROP, INDEX, SELECT, UPDATE}}]
[HIDESPLIT {ON | OFF}]
[LOGIN {{address:port, user_name, password} | NULL}]
[PROXY {{name, address:port, quiz} | NULL}]
[TIMEOUT connect_time]
[COMPRESS {ON | OFF}]
[ENCRYPT {ON | OFF}]

TEMP:

读取检索结果时临时文件所存放的目录，缺省为操作系统的临时文件目录。为了提高效率，设置路径时不会检查有效性，只是在底层产生临时文件时才会报错。如果没有设置这个参数，会使用集成环境里的缓冲路径设置。

BUFFER:

检索结果缓冲区大小，以记录为单位，缺省为 128。

BATCH:

读取检索结果时每次取记录数，缺省为 16。

SEPARATOR:

多值字段的分隔符，仅对 DATE、NUMBER 和 CHAR 类型字段有效，缺省为‘;’。

DRAWTEXT:

对格式化文档尽量提取文本。

AUTOWRAP:

浏览文本自动折行模式。**OFF** 表示不自动折行；**ALL** 表示所有文本不分行；**SEGMENT** 表示段落文本不分行。

URLTOFILE:

为 **ON** 表示对 URL(统一资源定位)输出其对应的文件(如 HTML 等)，而不是输出 URL 本身。缺省值为 **OFF**。

QUICKEND:

集群对后台检索结果的裁减。

SORTSAVE:

排序裁减和有效排序结果记录数。

RELEVANT:

以相关性模式和 **IDF** 模式组合的相关性排序选项。**NULL** 表示由系统配置决定；**NORMAL** 表示忽略词的位置和使用基于单个数据库范围的 **IDF** 值；**GLOBAL** 表示忽略词的位置和词 **IDF** 值不参与相关性计算；**NORMPOS** 表示考虑词的位置和使用基于单个数据库范围的 **IDF** 值；**GLOBPOS** 考虑词的位置和词 **IDF** 值不参与相关性计算；**WORDS** 表示用一个记录中命中词的单位向量长度（每个命中词只计算一次）作为记录的相关；**WFRE** 表示用一个记录中命中词的单位向量长度，以及命中词的词频和作为记录的相关度。

HITPOINT:

命中点的存储模式。

THESAURUS:

利用知识辅助词典自动进行知识扩展检索时的主题词典。不指定时使用系统配置的缺省主题词典。

SYNONYM:

利用知识辅助词典自动进行知识扩展检索时的同义词典。不指定时使用系统配置的缺省同义词典。

ANTONYM:

利用知识辅助词典自动进行知识扩展检索时的反义词典。不指定时使用系统配置的缺省反义词典。

KNOWASS:

利用知识辅助词典自动进行知识扩展检索的选项，可为下表中的值之一，多个值之间以逗号分隔。为 **NULL** 则表示取消所有自动扩展功能。

extend_type	说明
ST	自动进行“同义词”扩展检索
AT	自动进行“反义词”扩展检索
PT	自动进行“等同词”扩展检索
UF	自动进行“替代词”扩展检索

AB	自动进行“缩略词”扩展检索
ABF	自动进行“被缩略词”扩展检索
RT	自动进行“相关词”扩展检索
BT	自动进行“上位词”扩展检索
NT	自动进行“下位词”扩展检索
ALLBT	自动进行“所有上位词”扩展检索
ALLNT	自动进行“所有下位词”扩展检索
ALLKAX	自动进行所有扩展检索
ECX	自动进行“半角/全角字符”扩展检索

LIST BY:

显示对象的排列顺序。排列的顺序包括对**NAME**(名称)、**OWNER**(拥有者)、**TYPE**(类型)、**CREATE**(创建日期)、**COUNT**(字段数)、**COMMENT**(说明信息)的升序**ASC**和降序**DESC**排列。

RIGHT:

显示对象列表时的权限条件。可为 **SELECT**、**UPDATE**、**INDEX**、**ALTER**、**DROP** 之一，使用 **NULL** 可把该标志设为零，表示显示所有对象。

HIDESPLIT:

为 **ON** 表示隐藏自动分裂数据库。

LOGIN:

集群对后台检索结果的裁减。

PROXY:

代理服务器设置。name、address、port、quiz 分别表示代理服务器的服务器的类型、IP 地址、端口号和安全验证码。

TIMEOUT:

建立网络连接过程的超时时间。

COMPRESS:

为 **ON** 表示在传输数据时进行压缩。

ENCRYPT:

为 **ON** 表示在传输数据时进行加密。

4. 设置用户自定义变量

SET [VARIABLE] {{NAME=<value>} [...]}

VARIABLE:

用户自定义变量名和值，以 **NAME=<value>** 的形式出现。

【举例】

```
TRSL> SET CONFIG PATH '+C:\TRS4\user1' IDLE 20,23 NUMBER OFF
```

这条语句的作用是向增加数据库目录集里添加一个新的目录'C:\TRS4\user1',并把自动维护起始时间设为 20, 自动维护终止时间设为 23, 数字索引时间设为 OFF。

```
TRSL> SET SETTINGS BUFFERPATH D:\Temp URLTOFILE ON
```

这条语句的作用是把存放检索结果文件的目录设为 D:\Temp,并把 URL 自动转换为文件。

```
TRSL> SET SETTINGS THESAURUS mythes FUNCTION RT, BT, NT
```

这条语句的作用是把自动扩展检索的主题词典名设为 mythes, 并自动进行 RT、BT、NT 扩展检索。

```
TRSL> SET PARAMETER basename=system.demo, cond='中国*美国'
```

这条语句的作用是 basename 变量设为 system.demo, 把 cond 变量设为'中国*美国'。

【参考】

[SHOW 显示信息](#)

3.3.34 SHOW 显示信息

【目的】

显示系统配置、对象列表和其它信息。

【前提】

合法的 TRS 用户，已经登录到 TRS 服务器中。

【语法】

1. 显示当前的日期和时间

SHOW TIME

2. 显示当前服务器的系统配置

SHOW SYSTEM [DETAIL]

服务器的配置信息包括系统路径、知识词典和其它配置三大类。如果当前操作者是 SYSTEM 用户，可以用 SET 语句修改系统设置的值。DETAIL 选项表示显示系统资源以及版本信息等。

3. 显示运行环境参数

SHOW LOCALE [DETAIL]

TRSL 环境变量是 TRSL 内部保存的一组运行状态的值，包括临时文件路径、自动扩展检索、读取结果设置、记录维护选项等。每次 TRSL 启动时自动复位，任何用户均可用 SET 语句修改这些环境变量的值。DETAIL 选项表示显示运行环境资源以及版本信息等。

4. 显示当前的用户自定义变量值

SHOW VARIABLE

TRSL 的用户自定义变量是由用户自己设置的一些参数值，在执行 TRSL 时可以代替具体的对象名、表达式等用户输入的信息。变量的命名规则和对象名一致，但使用时要在前面加上 '\$' 符号。每次退出 TRSL 时自动清除，任何用户均可用 SET 语句修改这些变量的值。

5. 显示数据库信息

SHOW DATABASE [[owner.]database_name [{(column_name [, ...])}] [DETAIL]

```
[INDEX] [SIZE]
  [TYPE {NORMAL, {(SEGMENT, SEGMENTEX, STOP, STOPEX, SPARSE,
  THESAURUS, SYNONYM, ANTONYM} | KNOWASS)}}]
  [LIST BY {NULL | {NAME | OWNER | TYPE | CREATE | COUNT | COMMENT}
  [ASC | DESC]]]
  [RIGHT {NULL | {ALTER, DROP, INDEX, SELECT, UPDATE}}]
```

当**DATABASE**后面不带参数，显示符合指定条件的数据库的列表；如果指定了database_name，但后面没有其它参数，则显示符合条件的数据库列表；如果后面跟**DETAIL**参数，则显示由database_name指定的数据库的信息；如果后面跟**SIZE**参数，则显示该数据库占用的空间大小；如数据库名后跟**INDEX**参数，则显示该数据库包含的字段列表；如数据库名后跟字段列表，则显示这些指定数据库字段的属性。owner、database_name、column_name均可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。可用**TYPE**子句限定显示的数据库类型，分别可为**NORMAL**、**SEGMENT**、**SEGMENTEX**、**STOP**、**STOPEX**、**SPARSE**、**THESAURUS**、**SYNONYM**、**ANTONYM**、**KNOWASS**(表示所有词典)之一，多种词典类型之间以逗号分隔。**LIST BY**是指显示对象以**NAME**(名称)、**OWNER**(拥有者)、**TYPE**(类型)、**CREATE**(创建日期)、**COUNT**(字段数)、**COMMENT**(说明信息)的升序**ASC**或者降序**DESC**之一排列。**RIGHT**表示显示对象列表时的权限条件。

6. 显示视图信息

```
SHOW VIEW [[owner.]view_name [{column_name [, ...]}] [DETAIL]
  [LIST BY {NULL | {NAME | OWNER | CREATE | COMMENT} [ASC | DESC]
  [RIGHT {NULL | {ALTER, DROP, SELECT}}]
```

当**VIEW**后面不带参数，显示符合指定条件的视图的列表；如果指定了view_name，但后面没有其它参数，则显示符合条件的视图列表；如果后面跟**DETAIL**参数，则显示由view_name指定的视图的信息；如视图名后跟字段列表，则显示这些指定视图字段的属性。owner、view_name、column_name均可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。**LIST BY**是指显示对象以**NAME**(名称)、**OWNER**(拥有者)、**CREATE**(创建日期)、**COMMENT**(说明信息)的升序**ASC**或者降序**DESC**之一排列。**RIGHT**表示显示对象列表时的权限条件。

7. 显示用户组信息

```
SHOW GROUP [group_name [DETAIL]]
  [LIST BY {NULL | {NAME | CREATE | COMMENT} [ASC | DESC]]]
```

如果**GROUP**后不带参数，显示所有用户组的列表；如果指定了group_name，但没有跟**DETAIL**参数，则显示符合指定条件的用户组的列表；否则显示指定用户组的详细属性

信息。group_name 里可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。**LIST BY** 是指显示对象以 **NAME**(名称)、**CREATE**(创建日期)、**COMMENT**(说明信息)的升序 **ASC** 或者降序 **DESC** 之一排列。

8. 显示用户信息

```
SHOW USER [[group.]user_name [DETAIL]]  
    [LIST BY {NULL | {NAME | GROUP | TYPE | CREATE | COMMENT} [ASC |  
DESC]]]
```

如果 **USER** 后不带参数，显示所有用户的列表；如果指定了 group.user_name，但没有跟 **DETAIL** 参数，则显示符合指定条件的用户的列表；否则显示指定用户的详细属性信息。group 和 user_name 里均可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。**LIST BY** 是指显示对象以 **NAME**(名称)、**GROUP**(拥有者)、**TYPE**(类型)、**CREATE**(创建日期)、**COMMENT**(说明信息)的升序 **ASC** 或者降序 **DESC** 之一排列。

9. 显示数据格式信息

```
SHOW FORMAT [[class.]format_name [DETAIL]]  
    [LIST BY {NULL | {NAME | CLASS | SUFFIX | COMMENT} [ASC | DESC]]]
```

如果 **FORMAT** 后不带参数，显示所有数据格式的列表；如果指定了 class.format_name，但没有跟 **DETAIL** 参数，则显示符合指定条件的数据格式的列表；否则显示指定数据格式的详细属性信息。class 和 format_name 里均可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。**LIST BY** 是指显示对象以 **NAME**(名称)、**CLASS**(所属分类)、**SUFFIX**(后缀列表)、**COMMENT**(说明信息)的升序 **ASC** 或者降序 **DESC** 之一排列。

10. 显示访问权限信息

```
SHOW RIGHT [{group_name | user_name} [, ...]]  
    ON [owner.]object_name [{column_name [, ...]}]
```

group_name/user_name 为用户组/用户名，如不指定则表示所有用户组 and 用户。owner 为数据库/视图的拥有者，object_name 为数据库/视图名。column_name 为字段名，如不指定字段名表示要显示数据库级权限信息。其中，group_user 和 column_name 可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。执行该语句将显示指定用户组/用户在数据库/视图/字段上的访问权限信息。

11. 显示索引词信息

```
SHOW WORDS {[owner.]database_name | [owner.]auto_view_name} [(column_name)]  
    [WHERE condition]  
    [LIST BY {NULL | {NAME | RECORD | WFREQ} [ASC | DESC]]]
```

[**PRUNE** prune_count]

database_name/auto_view_name表示查询的数据库/自动模式视图名。column_name指定查询的字段，如没有给出column_name则在缺省字段上进行查询。。condition为检索的目标条件，为字符串类型，可以使用通配符，以'?'匹配单个字符，用'%'匹配任何字符，缺省时表示所有词语。user.database_name指定查询的数据库，column_name指定查询的字段，如没有给出column_name则在缺省字段上进行查询。**LIST BY**是指显示对象以**NAME**(名称)、**RECORD**(字段值/词的命中记录数)、**WFREQ**(字段值/词的词频数)的升序**ASC**或者降序**DESC**之一排列。**PRUNE**表示显示结果词语的个数。

12. 显示控制文件设置

SET CONTROL filename

filename 为控制文件名。

【举例】

TRSL> **SHOW TIME**

这条语句的作用是显示当前的日期和时间。

TRSL> **SHOW LOCALE**

这条语句的作用是显示当前所有的 TRS 环境变量的值。

TRSL> **SHOW DATABASE** system.*

这条语句显示 TRS 中的 *SYSTEM* 用户的所有数据库。

TRSL> **SHOW DATABASE** newsbase **STRUCTURE**

这条语句的作用是显示数据库 *newsbase* 的结构。

TRSL> **SHOW VIEW** demo_view (正文, 图象)

这条语句的作用是显示视图 *demo_view* 的“正文”和“图象”字段的属性。

TRSL> **SHOW GROUP** EPROBITI

这条语句的作用是显示 *EPROBITI* 用户组的属性。

TRSL> **SHOW USER** PRODUCT.*

这条语句的作用是显示 *PRODUCT* 组里所有用户的列表。

TRSL> **SHOW RIGHT** EPROBITI.* **ON** newsbase(正文)

这条语句的作用是显示 *EPROBITI* 组里所有用户对 *newsbase* 数据库的“正文”字段的访

问权限。

TRSL> **SHOW WORDS** '中国%' **ON** *newsbase*

这条语句的作用是在数据库 *newsbase* 的缺省字段上显示所有以“中国”开始的检索词的命中数和词频数。

【参考】

[SET 设置系统参数](#)

3.3.35 SORT 对检索结果重新排序

【目的】

对当前的检索结果重新进行排序。

【前提】

当前存在检索结果记录集。

【语法】

```
SORT BY {RELEVANCE | {column_name [ASC | DESC]} [, ...] | LIFO} [{PRUNE | VALID} record_number] [MIXED] [MISCOL] [NOCASE]
```

【参数】

sort_column:

指定检索结果排序的字段名，必须在源数据库/视图里存在。对每个排序字段可用 **ASC** 或 **DESC** 分别指定按升序/降序方式排列。字段名也可为 **RELEVANCE**，表示按相关性排序，但只能按升序排列。也可指定为 **LIFO**，表示按后进先出规则排序。

MIXED:

表示对检索结果进行混合统一排序。

MISCOL:

表示对排序表达式中不存在的字段，按字段无值时情况处理。

NOCASE:

表示排序表达式中的字段值，其字符是大小写无关的。

【要点】

1. 只有在成功地进行了 **SELECT** 操作以后，才能对检索结果进行重新排序。
2. 本次排序的结果在重新进行检索或排序、或检索结果被清除前一直有效。

【举例】

```
TRSL> SELECT FROM demo1, demo2 WHERE 中国
```

```
TRSL> SORT BY 日期 DESC, 版次 MIXED
```

上一条语句在 *demo1* 和 *demo2* 数据库里查询含有“中国”的记录，下一条语句对检索结果先按 *日期* 的降序、再按 *版次* 的升序重新排列，对两个数据库进行统一排序。

【参考】

[SELECT 执行检索](#)

3.3.36 SPOOL 输出到文件

【目的】

把 TRSL 的执行过程和结果输出到文件中。

【前提】

无。

【语法】

```
SPOOL { file_name | OFF }
```

【参数】

file_name:

输出文件名，为客户端的 DOS 文件名，可以带路径。

OFF:

取消输出设置，关闭上次 **SPOOL** 语句中设置的输出文件。

【要点】

使用该命令可以将执行的过程和结果以文件方式保留起来，可供以后查阅。缺省值是 **OFF**。

【举例】

```
TRSL> SPOOL result.doc  
TRSL> SELECT * FROM rmr  
TRSL> SPOOL OFF
```

上面三条语句的功能是把把在 *rmr* 数据库上进行检索的结果信息保存到 *result.doc* 文件里。

【参考】

[ECHO 显示控制](#)

3.3.37 UNLOCK 解锁数据库

【目的】

解除数据库的锁定。

【前提】

当前操作用户必须是 DBA 用户，或是数据库所有者所在组的 GDA 用户，或是数据库的拥有者。

【语法】

```
UNLOCK {[owner.]database_name [...]}
```

【参数】

owner:

数据库的拥有者，可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。

database_name:

指明要解锁的数据库名，可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。

【要点】

无。

【举例】

```
TRSL> UNLOCK demo2
```

这条语句的作用是解除demo2的锁定。

【参考】

[LOCK 锁定数据库](#)

3.3.38 UPDATE 更新数据库记录

【目的】

修改数据库记录。

【前提】

操作用户对数据库有 UPDATE 权限。

【语法】

```
UPDATE {[owner.]database_name | [owner.]auto_view_name}
      [SET {{column_name=<value>} [, ...]]]
      [WHERE condition]
      [SEPARATOR separator]
      [PASSWORD {password | NULL}]
      [GROUP {group_name | NULL}]
      [SECURITY security_code]
      [CONTROL control_file] /
```

【参数】

owner:

目标数据库/自动模式视图的拥有者。

database_name/auto_view_name:

目标数据库/自动模式视图名。

SET:

设置新的字段名和新值。**column_name** 为待修改的字段名，可同时修改多个字段；**value** 表示修改后的字段值，与 **column_name** 一一对应。如果不指定字段名，则表示对词典进行操作。

WHERE:

待更新记录的条件，只有满足条件的记录才会被更新。也可用 **DOCID=id** 表示物理记录号为 **id** 的记录。除此之外，本子句还可以指定删除词语。

SEPARATOR:

多值字段的分隔符。为字符串类型，仅对 DATE、NUMBER 和 CHAR 类型字段有效，缺省为';'。

PASSWORD:

记录的访问密码，取该记录内容时必须提供正确的密码。

GROUP:

记录对外开放的用户。

SECURITY:

赋予记录的安全级别，访问这些记录时操作者的安全级别必须不大于此安全级别。取值范围为 0 到 0xFFFFFFFF，缺省为零，表示访问此记录无须经过安全级别检查。

CONTROL:

加载的控制文件名。

【要点】

1. 本语句可一次可以更新多条记录。
2. 如果不指定修改条件时，默认为修改数据库中的所有记录。
3. 本语句不能更新词典中的数据。

【举例】

```
TRSL> UPDATE personnel SET name= '张三', resume='@zhang.res' WHERE  
name='ZHANG SAN'
```

这条语句的作用是把数据库 *personnel* 中满足条件 *name='ZHANG SAN'* 的记录修改为 *name='张三'*, *resume='@zhang.res'*。

```
TRSL> UPDATE personnel SET salary= 3000 WHERE position='engineer'
```

这条语句的作用是将那些职务(字段 *position*)为工程师(*engineer*)的员工的工资修改为 3000。

【参考】

[INSERT 插入新记录](#)

[DELETE 删除数据库记录](#)

3.3.39 VALIDATE 验证索引

【目的】

验证数据库文件的索引文件的正确性。

【前提】

操作用户必须对指定数据库有 INDEX 权限。

【语法】

```
VALIDATE INDEX {[owner.]database_name | [owner.]auto_view_name} [(column_name  
[, ...])]
```

【参数】

owner:

待验证索引的数据库的拥有者。

database_name/auto_view_name:

待验证索引的数据库/自动模式视图名。

column_name:

待验证的字段名，可使用通配符，用‘?’匹配单个字符，用‘*’匹配任何字符。不指定字段列表时表示验证所有字段。

【要点】

1. 索引的出错信息登记在文件 TRSValididx.log 中。
2. 如果验证的结果表明索引不正确，用户可用 **DROP INDEX** 和 **CREATE INDEX** 语句删除索引和重新创建索引。

【举例】

```
TRSL> VALIDATE INDEX zibase(c1, c2)
```

这条语句的作用是验证数据库 *zibase* 中 *c1*、*c2* 字段上索引文件的正确性。

【参考】

[CREATE 创建对象](#)

[DROP 删除对象](#)

[OPTIMIZE 优化数据库](#)

附录 A 语句列表

● ALTER 修改对象属性和结构

1. 修改数据库/词典的属性和结构

ALTER DATABASE *database_name*

```
[ADD ({basecol_element [,basecol_element]...})]
[MODIFY ({basecol_element [,basecol_element]...})]
[DELETE {column_name [,column_name]...}]
[COMMENT comment]
[LANGUAGE {GB|GB2312|GBK|GB18030|BIG|BIG5|ENGLISH|UNICODE|UTF8}]
[ALIAS {alias, [,alias]...}]
[FIRSTCOL first_column]
[PATH data_path]
[HOST data_host]
[SEGMENT {[user.]segment_dic|NULL}]
[SEGMENTEX {[user.]segmentex_dic|NULL}]
[STOP {[user.]stop_dic|NULL}]
[STOPEX {[user.]stopex_dic|NULL}]
[SPARSE {[user.]sparse_dic|NULL}]
[STEM {ON|OFF}]
[CASE {ON|OFF}]
[NUMBER {ON|OFF}]
[MIXINDEX {ON|OFF}]
[PACKINDEX {ON|OFF}]
[PUBLIC {priv_type [,priv_type]...}]
```

2. 修改视图的属性和结构

ALTER VIEW *view_name*

```
[ADD ({viewcol_element [,viewcol_element]...})]
[MODIFY ({viewcol_element [,viewcol_element]...})]
[DELETE {column_name [,column_name]...}]
[FROM {[user.]source_name [, [user.]source_name]...}]
[WHERE condition]
[COMMENT comment]
[ALIAS {alias, [,alias]...}]
[FIRSTCOL first_column]
[PUBLIC {priv_type [,priv_type]...}]
```

3. 修改用户组的属性

```
ALTER GROUP group_name
    [COMMENT comment]
```

4. 修改用户的属性

```
ALTER USER user_name
    [TYPE {DBA|GDA|RESOURCE|CONNECT}]
    [GROUP {group|NULL}]
    [COMMENT comment]
    [HISTORY history]
    [DOWNLOAD download]
    [SECURITY security]
    [SEGMENTEX {[user.]segmentex_dic|NULL}]
    [STOPEX {[user.]stopex_dic|NULL}]
    [SPARSE {[user.]sparse_dic|NULL}]
    [LOGINEXCL {ON|OFF}]
```

5. 修改数据格式的属性

```
ALTER FORMAT format_name
    [CLASS class]
    [SUFFIX {suffix [,suffix]...}]
    [COMMENT comment]
    [COMPRESS {ON|OFF}]
```

● **BACKUP 备份数据库**

```
BACKUP [ALL] {[user.]database_name [, [user.]database_name]...}
    [TO device]
```

● **BROWSE 浏览检索结果**

```
BROWSE [DATABASE { ALL | {[user.]object_name, [[user.]object_name]...}
    [COLUMN { ALL | {column_name [,column_name]...}]]
    [RECORD { ALL | {{n1 [,n2]...}[ TILL n ][,m]...}}]
    [ORDER BY { sort_column [ASC|DESC] [, sort_column [ASC|DESC]...} [MIXSORT]]
    [FORMAT { RECORD | TRS | ISO2709 | XML}]
```

● **CLEAR 清除屏幕内容**

```
CLEAR
```

● CONNECT 登录到服务器

CONNECT [*server:port*] [ACCOUNT *username* [PASSWORD *password*]]

● COPY 复制数据库记录

COPY TO [*database_user.*]*database_name*

FROM [*source_user.*]*source_name*

[(*column_name* [,*column_name*]...)]

[WHERE *expression*]

[ORDER BY { *sort_column* [ASC|DESC] [,*sort_column* [ASC|DESC]]...}]

[INDEX { ON | OFF }]

● CREATE 创建新对象

1. 创建数据库/词典

A. 定义方式:

CREATE DATABASE *database_name*

[(*basecol_element* [,*basecol_element*]...)]

[TYPE {USERDEF|SEGMENT|SEGMENTEX|STOP|STOPEX|SPARSE|
THESAURUS|SYNONYM|ANTONYM}]

[COMMENT *comment*]

[LANGUAGE {GB|GB2312|GBK|GB18030|BIG|BIG5|ENGLISH|UNICODE|UTF8}]

[ALIAS {*alias*, [,*alias*]...}]

[FIRSTCOL *first_column*]

[PATH *data_path*]

[HOST *data_host*]

[SEGMENT {[*user.*]*segment_dic*|NULL}]

[SEGMENTEX {[*user.*]*segmentex_dic*|NULL}]

[STOP {[*user.*]*stop_dic*|NULL}]

[STOPEX {[*user.*]*stopex_dic*|NULL}]

[SPARSE {[*user.*]*sparse_dic*|NULL}]

[STEM {ON|OFF}]

[CASE {ON|OFF}]

[NUMBER {ON|OFF}]

[MIXINDEX {ON|OFF}]

[PACKINDEX {ON|OFF}]

[PUBLIC {*priv_type* [,*priv_type*]...}]

B.选择方式:

CREATE DATABASE *database_name*

AS SELECT [*•] {[*user.*]*source_name*].*source_column*...}]

[**FROM** [*user.*]*source_name*]
[**COMMENT** *comment*]
[**LANGUAGE** {**GB**|**GB2312**|**GBK**|**GB18030**|**BIG**|**BIG5**|**ENGLISH**|**UNICODE**|**UTF8**}]
[**ALIAS** {*alias*, [,*alias*]...}]
[**FIRSTCOL** *first_column*]
[**PATH** *data_path*]
[**HOST** *data_host*]
[**SEGMENT** {[*user.*]*segment_dic*|**NULL**}]
[**SEGMENTEX** {[*user.*]*segmentex_dic*|**NULL**}]
[**STOP** {[*user.*]*stop_dic*|**NULL**}]
[**STOPEX** {[*user.*]*stopex_dic*|**NULL**}]
[**SPARSE** {[*user.*]*sparse_dic*|**NULL**}]
[**STEM** {**ON**|**OFF**}]
[**CASE** {**ON**|**OFF**}]
[**NUMBER** {**ON**|**OFF**}]
[**MIXINDEX** {**ON**|**OFF**}]
[**PACKINDEX** {**ON**|**OFF**}]
[**PUBLIC** {*priv_type* [,*priv_type*]...}]

2. 创建视图

A. 定义方式:

CREATE VIEW *view_name*
[({*viewcol_element* [,*viewcol_element*]...})]
[**WHERE** *condition*]
[**COMMENT** *comment*]
[**ALIAS** {*alias*, [,*alias*]...}]
[**FIRSTCOL** *first_column*]
[**PUBLIC** {*priv_type* [,*priv_type*]...}]

B. 选择方式:

CREATE VIEW *view_name*
AS SELECT [*• {[[*user.*]*source_database.*]*source_column*...}]
[**FROM** [*user.*]*source_name*]
[**WHERE** *condition*]
[**COMMENT** *comment*]
[**ALIAS** {*alias*, [,*alias*]...}]
[**FIRSTCOL** *first_column*]
[**PUBLIC** {*priv_type* [,*priv_type*]...}]

3. 创建用户组

```
CREATE GROUP group_name  
    [COMMENT comment]
```

4. 创建用户

```
CREATE USER user_name  
    [TYPE {DBA|GDA|RESOURCE|CONNECT}]  
    [PASSWORD {password|NULL}]  
    [GROUP {group|NULL}]  
    [COMMENT comment]  
    [HISTORY history]  
    [DOWNLOAD download]  
    [SECURITY security]  
    [SEGMENTEX {[user.]segmentex_dic|NULL}]  
    [STOPEX {[user.]stopex_dic|NULL}]  
    [SPARSE {[user.]sparse_dic|NULL}]  
    [LOGINEXCL {ON|OFF}]
```

5. 创建数据格式

```
CREATE FORMAT format_name  
    [CLASS class_name]  
    [SUFFIX {suffix [,suffix]...}]  
    [COMMENT comment]  
    [COMPRESS {ON|OFF}]
```

6. 创建索引

```
CREATE INDEX ON [user.]database_name  
    [({column_name [,column_name]...})]  
    [STOP record]
```

● DELETE 删除数据库记录

```
DELETE [RECORD {ALL• {n1 [,n2]...}[TILL n ][,m]...}]  
    [FROM [user.]database_name]  
    [WHERE condition]  
    [FAST | COMPLETE]
```

● DROP 删除对象

1. 删除对象

```
DROP [DATABASE | VIEW | GROUP | USER]
```

[*user.*]*object_name*, [[*user.*]*object_name*]...

2. 删除索引

DROP INDEX ON [*user.*]*database_name*
 [(*column_name* [,*column_name*]...)]
 [**START** *record*]

● ECHO 显示控制

ECHO { **ON** • **OFF** }

● EDIT 编辑检索结果

EDIT [**DATABASE** { **ALL** | {[*user.*]*object_name*, [[*user.*]*object_name*]...}]
 [**COLUMN** { **ALL** | {*column_name* [,*column_name*]...}]
 [**RECORD** { **ALL** | {{*n1* [,*n2*]...}[**TILL** *n*][,*m*]...}]
 [**ORDER BY** { *sort_column* [**ASC**|**DESC**] [, *sort_column* [**ASC**|**DESC**]...}] [**MIXSORT**]
 [**FORMAT** { **RECORD** | **TRS** | **ISO2709** | **XML**}]

● EXIT 退出命令语言处理工具

EXIT

● GRANT 给用户授权

GRANT [**ALL** | { *priv_type* [, *priv_type*]...}]
 TO {*user_name* [,*user_name*]...}
 ON [*user.*]*object_name*
 [({*column_name* [,*column_name*]...})]

● HELP 显示帮助信息

HELP [*keyword*]

● INSERT 插入新记录

INSERT INTO [*user.*]*database_name*
 [({*column_name* [,*column_name*]... })]
 VALUES ({*value* [,*value*]... })
 [**PASSWORD** *password*]

● LOAD 把记录装进数据库

LOAD [**TRS** | **BATCH**] *load_file*
 INTO [*user.*]*database_name*
 [**CFILE** *control_file*]

● **LOCK 为数据库解锁**

LOCK {[user.]database_name [, [user.]database_name]...}
[IN SHARE MODE]

● **MAIL 发送检索结果**

MAIL [DATABASE { ALL | {[user.]object_name, [[user.]object_name]]...}]
[COLUMN { ALL | {column_name [,column_name]...}]
[RECORD { ALL | {{n1 [,n2]...} [TILL n] [,m]...}]
[ORDER BY { sort_column [ASC|DESC] [, sort_column [ASC|DESC]...}] [MIXSORT]]
[FORMAT { RECORD | TRS | ISO2709 | XML}]

● **NOTEPAD 编辑文件**

NOTEPAD [file_name]

● **OPTIMIZE 优化数据库**

OPTIMIZE [user.]database_name
[(column_name [,column_name]...)]
[ON device]

● **PRINT 打印检索结果**

PRINT [DATABASE { ALL | {[user.]object_name, [[user.]object_name]]...}]
[COLUMN { ALL | {column_name [,column_name]...}]
[RECORD { ALL | {{n1 [,n2]...} [TILL n] [,m]...}]
[ORDER BY { sort_column [ASC|DESC] [, sort_column [ASC|DESC]...}] [MIXSORT]]
[FORMAT { RECORD | TRS | ISO2709 | XML}]

● **QUIT 退出并清除操作日志**

QUIT

● **RECOVER 恢复数据库**

RECOVER [ALL | {[user.]database_name [, [user.]database_name]...}]
[FROM device]
[TO directory]

● **REM 注释语句**

REM string

● **RENAME 为对象改名**

RENAME [DATABASE • VIEW • GROUP | USER | FORMAT | BASECOL | VIEWCOL]

[*user.*]*old_name* **TO** *new_name*
[**ON** [*user.*]*object_name*]

● REVOKE 收回用户的权限

REVOKE [**ALL** | { *priv_type* [, *priv_type*]...}]
FROM {*user_name* [,*user_name*]...}
ON [*user.*]*object_name*
[({*column_name* [,*column_name*]...})]

● RUN 运行脚本文件

RUN [*script_file*]

● SAVE 保存检索结果

SAVE [**DATABASE** { **ALL** | { [*user.*]*object_name*, [[*user.*]*object_name*]...}]
[**COLUMN** { **ALL** | { *column_name* [,*column_name*]...}}]
[**RECORD** { **ALL** | {{*n1* [,*n2*]...} [**TILL** *n*] [,*m*]...}}]
[**ORDER BY** { *sort_column* [**ASC**|**DESC**] [, *sort_column* [**ASC**|**DESC**]...}] [**MIXSORT**]
[**FORMAT** { **RECORD** | **TRS** | **ISO2709** | **XML**}]
[**TO** *file_name*]

● SELECT 对数据库/视图进行检索

SELECT [{*stat_expression* [,*stat_expression*]...}]
[**FROM** { [*user.*]*object_name* [, [*user.*]*object_name*]...}]
[**WHERE** *search_expression*]
[**ORDER BY** { *sort_column* [**ASC**|**DESC**] [, *sort_column* [**ASC**|**DESC**]...}] [**MIXSORT**]
[**DEFAULT** ({ *column_name* [, *column_name*]...)] [**AND**|**OR**|**NOT**|**XOR**]
[**SEGMENT** { **ON** | **OFF** }]

● SET 设置系统参数

1. 维护系统配置

SET CONFIG

[**MODIFY** {*knowass_element* [,*knowass_element*]...}]
[**PATH** {*data_path* [,*data_path*]...}]
[**TEMP** *temp_path*]
[**DEVICE** *backup_device*]
[**LANGUAGE** { **GB**|**GB2312**|**GBK**|**GB18030**|**BIG**|**BIG5**|**ENGLISH**|**UNICODE**|**UTF8**}]
[**LOGIN** *max_login*, *min_login*]
[**IDLE** *idle_start*, *idle_over*]
[**MEMORY** *memory_size*]

[**HISTORY** *history*]
[**DOWNLOAD** *download*]
[**SEARCH** *search_time*]
[**SORT** *sort_record*]
[**STEM** {**ON**|**OFF**}]
[**CASE** {**ON**|**OFF**}]
[**NUMBER** {**ON**|**OFF**}]

2. 修改环境变量

SET SETTINGS

[**BUFFERPATH** *buffer_path*]
[**BUFFERSIZE** *buffer_size*]
[**READCOUNT** *read_count*]
[**SEPARATOR** *separator*]
[**USERGROUP** *user_group*]
[**SECURITY** *security_code*]
[**URLTOFILE** {**ON**|**OFF**}]
[**THESAURUS** [*user.*]*thesaurus_dic*]
[**SYNONYM** [*user.*]*synonym_dic*]
[**ANTONYM** [*user.*]*antonym_dic*]
[**FUNCTION** {*extend_type* [*extend_type*]...}]
[**PUBLIC** {*priv_type* [*priv_type*]...}]

3. 设置用户自定义变量

SET PARAMETER

{ *variable= value* [,*variable= value*]... }

● SHOW 显示信息

1. 显示当前的日期和时间

SHOW TIME

2. 显示当前服务器的系统配置

SHOW CONFIG

3. 显示当前的 TRSL 环境变量值

SHOW SETTINGS

4. 显示当前的用户自定义变量值

SHOW PARAMETER

5. 显示数据库信息

SHOW DATABASE

```
[[user.]database_name [STRUCTURE | INFORMATION | SIZE |  
    {(column_name [,column_name]...)}  
    [TYPE {dic_type [,dic_type]...}]
```

6. 显示视图信息

SHOW VIEW

```
[[user.]view_name [STRUCTURE | INFORMATION | {(column_name  
[,column_name]...)}]
```

7. 显示用户组信息

SHOW GROUP [group_name [INFORMATION]]

8. 显示用户信息

SHOW USER [[group.]user_name [INFORMATION]]

9. 显示数据格式信息

SHOW FORMAT [[class.]format_name [INFORMATION]]

10. 显示访问权限信息

```
SHOW RIGHT [group_user [,group_user]...]  
    ON [user.]object_name [(column_name [,column_name]...)]
```

11. 显示数据库的索引词

SHOW WORDS [string] ON [user.]database_name [(column_name)]

● **SORT 对检索结果重新排序**

```
SORT BY { sort_column [ASC|DESC ] [, sort_column [ASC|DESC]...}  
    [MIXSORT]
```

● **SPOOL 输出到文件**

SPOOL { file_name | OFF }

● **UPDATE 更新数据库记录**

```
UPDATE [user.]database_name  
    SET {column_name= value [,column_name= value]...}  
    [RECORD {ALL | {n1 [,n2]...} [TILL n ] [, m]...}}  
    [WHERE condition]
```

[**PASSWORD** *password*]

● **VALIDATE 验证索引**

VALIDATE INDEX ON [*user.*]*database_name*

[(*column_name* [,*column_name*]...)]

附录 B 保留字

AB	ABF	ADJ	ALL	ALLBT
ALLNT	AMB	AND	AT	AVG
BETWEEN	BEYOND	BT		
CASE	COUNT			
DATA	DOCID			
EQU	EXACT			
INCLUDE				
LE	LIFO	LIKE	LOWER	
MAX	MIN			
NEAR	NOT	NT		
OR				
PRE	PT			
RANGE	RELEVANCE	RT		
SPELL	ST	SUM		
TO	TOASCII	TOCHINESE		
UF	UPPER			
XOR				