

Toad 快速入门

一、Toad 功能综述

在 Oracle 应用程序的开发过程中，访问数据库对象和编写 SQL 程序是一件乏味且耗费时间的工作，对数据库进行日常管理也是需要很多 SQL 脚本才能完成的。

Quest Software 为此提供了高效的 Oracle 应用开发工具-Toad (Tools of Oracle Application Developers)。在 Toad 的新版本中，还加入了 DBA 模块，可以帮助 DBA 完成许多日常管理工作。它最大的特点就是简单易用，访问速度快。使用 Toad，我们可以通过一个图形化的用户界面快速访问数据库，完成复杂的 SQL 和 PL/SQL 代码编辑和测试工作。Toad 由 Oracle 开发专家专门为开发人员而设计，是一个功能强大、结构紧凑的专业化 PL/SQL 开发环境。

Toad 主要具有如下特点：

模式浏览：

模式浏览功能可以让我们快速访问数据字典，浏览数据库中的表、索引、存储过程。Toad 提供对数据库的快速访问，使用极为方便，用户界面简洁，结构安排合理。当我们点击一个单独的数据库对象，Toad 立即显示此对象的详细信息。例如，当我们点一个数据库的表，所有和此表相关的索引、约束、存储过程、SQL 语句以及和其他表的相互引用关系都在同一界面显示出来。为了简化操作，用户可以在浏览窗口操作数据库对象。

SQL 编辑器：

SQL 编辑器的主要功能是编辑、运行和调整 SQL 语句。TOAD 的高级编辑窗口包括众多的特性来提高开发人员编写 SQL 语句的产品化程度。例如，简单地生成代码模板，在编写 SQL 前自动发现包的内容和列的名字等等。

SQL 编辑器包括一个编辑窗口和运行结果窗口，允许开发人员在编辑的过程中测试运行结果。SQL 编辑器中不仅包括标准的编辑命令，也包括一些增强的功能，如快速查询表中的字段、将 SQL 语句的内容格式化等等。这个窗口可以处理大到 4GB 的内容，对大的开发项目来说非常有用。便捷的书签可以让开发人员非常容易地找到相关位置。在运行结果窗口可提供用户定义的配置功能，支持 LONG 和 LONG RAW 列，可以将数据卸出到磁盘、打印数据、编辑数据等等。

存储过程编辑器：

存储过程编辑器的主要功能是编辑、编译、测试、调试存储过程和触发器。TOAD 提供语法标识、错误标识和其他很多易于使用的功能，如在弹出窗口显示表名、列名和 Oracle 函数。和其他的 PL/SQL 编辑工具不同，TOAD 允许在一个文件中操作多个数据库对象，可以编译一个对象、编译多个对象、编译到当前光标、从光标开始编译。在运行出现错误时，存储过程停止到有问题的语句。用户可以使用快捷方式或模板来快速编写 PL/SQL，也可以根据需要生成自己的模板。使用 Toad 可以非常方便地进行编辑工作，可如设置书签、取消注释、格式化 SQL 语句等等。

PL/SQL Debugger 选项：

Toad 提供简单易用的 PL/SQL 调试功能，可以节省开发人员在大型项目中用于开发和测试的宝贵时间，提高应用开发的质量。在存储过程开发的过程中，Toad 可以逐行编辑、调试和运行代码。运行时可以根据需要输入参数，观察相关参数的变化来检查存储过程的正确性。在调式过程中，Toad 可以通过窗口显示所有的断点、参数，调用堆栈和输出参数。使用 Toad，非常容易检测到存储过程的错误，开发人员可以一步一步运行 PL/SQL 语句来识别问题。调试会话可以和其他程序会话同时进行。

SQLab Xpert Option：

帮助开发人员优化 SQL，为他们提供各种优化模式下 SQL 执行计划，并且能够给出优化的建议，能够比较各种模式下实际的 SQL 运行结果，帮助开发人员真正高速地开发高效地代码。

Toad 还可以外挂一些别的产品，比如 PL/Formatter, RevealNet Knowledge Base，SQL Impact 等，这些都能够和 Toad 紧密集成，共同提供了一个完美的集成开发环境。新版本还新增加了 DBA 模块，更加拓广了 Toad 这个产品的适用范围。

二、系统要求：

客户端：

- ü Windows 95 / 98 / NT/2000 中英文均可
- ü 完整安装的 SQL*Net 2.0 或者 Net8
- ü 磁盘：一般安装需要 20M 左右的空间，根据所选择模块的不同和安装方式的不同，最少也要 5M 的磁盘空间（从网络服务器读取）
- ü 内存：Windows 9x/Me:16M 内存；Windows NT/2000: Min 32M]
- ü Tuning 模块需要额外的 25M 空间

服务器端：

- ü Oracle 7.3 或者更高，在任何硬件平台上。
- ü 需要一个 DBA 用户帐号

三．安装：

1．三种安装方式的选择：

a) TOAD and Personal Configuration Files to PC

默认安装选择，完成一个完整的 Client 端的安装，把所有需要的文件拷贝到本地 Toad 目录。

绝大多数用户应该选择使用这个安装方式。

b) Personal Config Files to PC, read TOAD from Network Server

从网络服务器读取 Toad 程序，在本地计算机安装一些自定义的文件，并且创建到服务器文件的快捷方式，一般不推荐使用这个方式，而且要求网络服务器已经配置。

c) TOAD to Network Server

把 Toad 安装到网络服务器上，以便这样别的用户能够进行 Personal Config Files to PC, read TOAD from Network Server 这种方式的安装。

2. 根据需要使用到的特性，需要运行一些脚本(都在 Toad\temps 目录下)

ü 如果需要用 Toad 来查看执行计划，必须运行脚本 PrepToad.sql 或者 notoad.sql

n 如果希望专门建立一个 Toad 用户来存放 Toad 需要使用的对象的话，就运行 preptoad.sql。

n 如果希望在当前连接用户模式下建立 toad 需要使用的对象的话，就运行 notoad.sql。

n 注意，如果是选择了专门建立 toad 这个用户的话，需要先修改一下脚本，指定用户的默认表空间和临时表空间。

ü 需要使用 Oracle8i 的 Profile analyzer，必须运行 ToadProfiler.sql

ü 需要加强 Toad 的安全性，必须运行 ToadSecurity.sql

四、Toad 使用快速入门

1. Schema Browser :

Schema browser 是 Toad 的极为有特色的模块。在 schema browser 的帮助下，可以方便的浏览数据库内部各类数据库对象，比如表，索引，序列，存储过程等，而且可以方便的倒出 DDL 语句和进行各种修改，重建工作。定位到某一类对象上，这列对象允许的操作都会自动列出来。

特色：

1 支持 Oracle 数据库里面所有对象的新建、查看、修改，集成了几乎对所有数据库对象的管理所需要的功能。

1 按照 模式->对象类别->对象->对象可操作属性->对象本身的详细信息和对象相关的信息 来组织，非常有条理，容易定位对象，也能够看到所有你需要的对象的相关信息。

1 对于表：

i. 完备的建表向导，不用输入冗长的语法，更加全面地替你考虑。

ii. 可以修改表的各种逻辑和物理属性，修改列的属性，增加列，删除列(在 Oracle8i 里面)，修改表的物理存储属性，Truncate，Drop 等。

iii. 可以生成建表的完整的 DDL 语句，包括存储信息，权限信息，约束，索引和触发器等，而这些手工很难做到。

iv. 可以方便的对表进行分析，查看表地所有信息，可以有条件的倒出表里面的数据，并且保存为 sql 语句。

v. 可以方便的对表进行重组，可以完成 exp/imp 作不到的事情，而且更少地人工参与，更加安全。

1 对于视图，存储过程：

i. 可以保存建立视图、存储过程的语句为文本，

ii. 重新编译失效的视图，存储过程，修改视图的定义语句等。方便的查看存储过程的代码，相互依赖关系，权限信息等保存为文本等。

1 回滚段，

- i. 简洁明了的 online，offline 状态，方便地 online/offline 操作
- ii. 详细的回滚段物理信息和当前状态。
- iii. 可以方便的修改物理存储属性

1 表空间：

- i. Tablespace Map；直观的显示数据物理的在表空间上的分布
- ii. 直观的查看各个表空间的利用率、剩余空间、破碎情况等信息
- iii. 可以进行各种 alter 操作：online, offline, 增加数据文件，改变数据文件大小，改变物理存储属性等

1 对其他数据库对象也有完备的操作支持。

2 . SQL Editor:

a) 激活 SQL Editor 的三种方式：

- i. 连接到数据库之后，Toad 自动打开 SQL Editor 这个窗口。
- ii. 点击总工具栏上最左边的那个按钮
- iii. 从菜单栏：Database -> SQL Editor

b) 特点：

1 允许同时打开多个编辑窗口，同时编辑多个 SQL 语句

1 支持同时连接到多个数据库，同时对多个数据库进行操作

1 允许编辑 SQL，PLSQL，JAVA，HTML 和文本

1 使用书签功能，可以在大量 SQL 语句里，快速定位某个语句所在位置

1 SQL Editor 本身具有强大的编辑功能，非常方便编辑 SQL 语句。

1 强大的快捷键支持和工具栏快速按钮的支持，更加高效编写(很多快捷键可以自己定义，在 View ->options ->SQL Editor 里面)

1 能够根据前面的条件，智能推测需要输入的代码，可以做到某种程度的代码智能填充

1 把鼠标定位到表/视图/存储过程名称之上，按 F4，可以打开对象描述窗口，方便的查看表和视图的定义，存储过程的源代码，

1 非常容易对 SQL 语句的分析其执行计划：单击工具栏上的 按钮就可以看到 Explain Plan 的结果

1 支持部分 SQL*Plus 命令，可以方便的执行大量的为 SQL*Plus 编写的数据库脚本

1 可以通过 SQLLab xpert 来对当前 SQL 语句进行全面的分析，给出各种合理的调优意见，只需要单击工具栏上的 Tuning 按钮：

1 可以方便的保存，打开和运行 SQL 脚本

1 支持对查询结果进行修改，

1 可以把查询结果以多种格式保存起来，可以保存为 HTML，纯文本，XLS 等多种格式

1 很容易的得到 SQL 函数的列表和用法描述（Show SQL help Window 按钮）

1 可以查看 SQL 语句的执行情况：通过下方的 AutoTrace 窗口，看到 SQL 语句的实际执行结果和执行效率。

- 1 支持历史 SQL(F8)：可以方便的查询以前执行过的 SQL，不用反复重新输入。
- 1 可以执行匿名块，存储过程，从下方的窗口看到 DBMS_OUTPUT 的结果，如：

3. Stored Procedure Editor

a) 打开存储过程编写器的几个途径：

- 1 单击总菜单栏上的第三个按钮：
- 1 点击 Database - > Procedure Editor，
- 1 可以通过 Schema browser 来调用

b) 一些特点和优点：

- 1. 强大的自定义设置：从保留字的显示方式，到自动替换，到各种快捷键的定义等等，用得越熟越顺手。
 - 2.
 - 3. 自动用不同颜色显示 SQL 和 PL/SQL 关键字，语法清晰明了，可根据自己的习惯来具体定制各类关键字的具体显示方式。可以自定义存储过程的模板，在新建存储过程的时候，自动生成程序的框架可以方便的调用 Schema Browser，把鼠标定位于某个对象上，F4，Schema Browser 打开该对象的详细描述，帮助顺利快速开发程序。支持代码自动更正，如输入 ndf，自动替换成 NO_DATA_FOUND，输入 the 自动替换成 the，类似的自动替换还可以自己修改和添加。
 - 7. 方便的左边行号显示，单击左边行号就可以实现调试断点的增加/删除
 - 8. 对各种快捷键的支持，不但包括常见的编辑操作，还可以有自己定义的 PLSQL 块的支持（自己定义快捷方式）。如可以定义 CTRL+SHIFT+D:弹出所有 PLSQL Block 的类型，可以自己选择。
 - 9. 可以在一个集成开发环境里面，一边编码，一边调试，不用多处切换
 - 10. 方便的同时打开多个存储过程，通过页面快速切换
 - 11. 快速编译存储过程，快速定位错误所在，自己选择是否同时编译依赖的存储过程
 - 12. 内置对 PL/Formatter，可以用 PL/Formatter 对存储过程进行格式化，并且可以提供存储过程的概要分析和修改建议
- PL/Formatter 可以对现有的存储过程进行高速的格式化，使所有的存储过程都符合同样的编写标准，这样既有利于提高程序的可读性，避免不同人写的模块的风格上的很差异。
- PL/Formatter 还有助于提高程序的模块化，增强程序的结构清晰度，增强 SQL 的共用性。
- PL/Formatter 还可以帮助完成 PL/SQL 和 Oracle 版本之间的管理。

13. 内置 RevealNet Knowledge Base，单击鼠标右键就可以打开，快速得到技术上的支持，解决技术难题

nPL/SQL:

- n 包含整个开发过程中可能遇到问题的答案

nCode Library:大量的严谨的 PL/SQL 代码例子和现成过程，完全源代码提供

- n 各种错误处理模块可供参考，各类字符日期函数

- n 包含 1400 多个主题

nAdmin:

- n 解答 DBA 日常工作中 80%的常见问题，覆盖 2400 多个主题

- n 从网络到性能优化，从备份恢复到 Error message,一应俱全

例子：遇到失效的存储过程怎么办？

4. 如何对存储过程进行 Debug？

- u 需要 Toad 单独的 Debug Option(Standard Version 没有这个选项)

- u 要用 Toad 对存储过程进行 debug，必须安装 Oracle 的系统包：dbms_debug，而且必须安装 Oracle Probe API v2.0 或者更高的版本。

- u 集成在 Stored Procedure Editor 里面，可以一边编写一边调试，大大提高开发效率

- u 打开 Procedure Editor 后，Debug 菜单激活，可以开始进行 Debug 工作。

- u 允许对存储过程进行逐行调试、断点调试,也支持对触发器进行调试。

- u 可以有断点支持，支持 Watch 变量，允许运行时改变变量的值

- u 允许对多层调用的 Debug 支持

调试步骤具体举例：

1. 点击菜单栏的第三个按钮：，打开 Procedure Editor

2. 点击 Procedure Editor 的菜单栏的 按钮，调出需要调试的存储过程

3. 单击工具栏上的按钮，输入需要传入的参数，如果不需要传入参数，可以直接单击工具栏上的按钮：，直接开始调试，如下图输入传入参数：

4. 单击菜单栏的按钮：，可以单步跟踪，也可以定义一个断点，直接运行到断点在单步执行，在存储过程运行的过程中，当前语句会高亮度显示，并且把鼠标指针定位到变量上，Toad 自动显示该变量的当前值。

5. 可以随时中止程序的执行，只要点 Debug ->Halt Execution，或者菜单栏上的按钮，就可以中止程序的执行

6. 当碰到一个存储过程调用另外一个存储过程的时候，可以选择跳过(Step Over)，这样就直接运行那个被调用的存储过程，返回调用结果，继续这个存储过程的调试，也可以选择 Trace Into，这样就打开被调用的存储过程，进一步深入调试那个被调用的子存储过程。如果保持 Debug 信息，在编译的时候应该选择 Compile Dependencies with Debug Information，就是 Procedure Editor 工具栏最右边

的那个按钮。

7. 支持对变量实时追踪：使用 Watch：

可以在运行前，或者运行过程中对某个变量进行实时跟踪：Add Watch，把鼠标定位到该变量上，然后点击 Debug ->Add Watch At Cursor，该变量就会出现在 Debug 状态窗口上，而且实时反应该变量的当前值。如上图中的变量 v_empname。可以运行时刻修改

8. 断点支持：

单击存储过程需要加断点的地方的行号，该行颜色变红，输入参数，单击执行按钮，可以直接运行到断点处，可以不必逐行追踪，更加高效率的进行调试。

Debug Option 还支持条件断点，只有当满足条件的时候才会在那个地方停止，否则就好像不存在这个断点一样继续运行。

?? 注意 DBMS_OUTPUT.PUT_LINE 的结果只有当程序执行完成以后才会出现，在程序执行过程中是没有结果的。

4. 如何用 SQLab Xpert 对存储过程进行调优？

我们知道，Oracle 数据库的优化，最主要的步骤就在于 SQL 的优化。而在没有专门的工具（如 Toad）的帮助下，进行 SQL 优化是一件非常困难和乏味的事情，并且需要对 Oracle 的深入理解和高超的 Oracle 调优技巧。使用 Toad 的 SQLab Xpert Tuning 模块，可以使我们普通开发人员也能够非常快速，精确的对 SQL 进行专家级的分析

在 SQL Editor 和 Procedure Editor 里面，选中需要调试的 SQL 语句（SQLab Xpert 只能调试 Select, Insert, Update, Delete 语句），复制到 SQL Editor 里面，然后点击工具栏上的调优按钮，Toad 会自动打开 SQLab xpert 窗口，捕获当前的 SQL 语句，并且把它转化为符合 explain plan 格式的语句。

第一次打开 Xpert 的时候，Toad 会提示你，需要安装一些 Server 端的表、视图和包等对象，可以把表存储到某个指定的表空间(如 Tools)里面。可以简单的按照提示一步一步做就可以了。如果系统设定不允许 Job 的话

（Job_queue_process=0），Toad 提示有些部件不会现在安装，可以忽略。安装完成以后，可以选择让某些非 DBA 用户也可以使用 Toad 的 SQLab Xpert 调优功能。

安装完成以后，Toad 会提示你一些关于优化目标方面的问题，你可以选择每次优化都可以重新设定，或者一次设定，以后一直使用这个设定。当然如果需要，还是可以更改的，在 SQLab Option xpert 的窗口的 Administration 和 Preference 下面重新设定。

例子：

下面这个 SQL 语句涉及到多表之间的关联，有一定的复杂性，我们可以借助 Toad 的 SQLab Xpert Option 来帮助分析：

```
select a.name, b.name, c.emp_name
from emp a, emp b, emp c
where a.emp_no=c.EMP_NO
and b.emp_id=c.emp_id
```

and c.emp_no=3

我们把这个 SQL 复制到 SQL Editor 里面，单击上面的调优按钮，Toad 打开 SQLab Xpert Option 窗口。

如下图：

从上面的图中我们可以方面的看到：在各种优化模式下，该 SQL 语句的执行计划分别是什么样的，Toad 给出了每一种执行计划的具体每一个步骤，每个步骤的成本，该表涉及的相关的数据等，如果 SQL 存取的表还没有经过分析，Toad 会以红色警告显示，并且很方便的帮助你对该表进行分析，收集统计数据，重建索引等，以有助于优化器作出最好的选择。可以把当前的分析结果保存到它的资料库里边，以备以后再次调用之需。

我们还可以求助与 SQLab 的自动优化功能：点击 Original SQL 下面的 auto tune 按钮，Toad 会对执行结果按照 tkprof 进行分析（当然，这需要你在 initsid.ora 里面设置好 utl_file_dir，如果服务器是在 Unix 机器上的话，还需要通过 FTP 方式得到 Trace 文件，有一些小小的设置）。

自动优化非常高效的对所有执行计划进行实际运行比较，并且给出实际执行的时间的对比，如下图：

我们还可以让 Toad 给出实际的优化建议：点击 Original SQL,单击下面的 Advice 按钮，Toad 根据对 SQL 的分析和执行结果给出合理的建议：

如：

1. 把表和索引分别放到两个不同的表空间，因为我们在创建表，加上约束的时候没有指定 USING INDEX 子句，所以表和索引都建立在了用户的默认表空间上了。
2. 对表 EMP, DEP 和 MANAGER 进行分析，收集最新的统计数据，有利于 CBO 得到最优的执行计划
3. 在合适的列上创建索引：EMP 的 manag_id 和 dep_no。当表中存在大量的数据的时候，这样可以大大减少由于全表扫描带来的成本。

5. SQL Modeler :

SQL Modeler 可以帮助开发人员很容易的写出各种复杂的 SQL 语句，只需通过简单的拖动鼠标，就能够写出复杂的 SQL 语句来。

打开 SQL Modeler 的几个途径：

- i. 菜单栏 Database -> SQL Modeler ，打开窗口
- ii. 单击工具栏第四个按钮 ，打开 SQL Modeler 窗口

窗口由三个主要部分组成：

- n 表选择区域：最右边的那个部分，用来选择 From 子句中用到的表。拖动到左边的区域或者双击就选择了该表。
- n 表模型区域：用来图形化的显示选择的表和表之间的关联关系。
- n 生成 SQL 区域：这里显示用图形化方式创建的查询对应的 SQL 语句，

在 SQL Modeler 里面，还支持对生成的 SQL 进行 Explain Plan，调试 SQL 的性能，运行 SQL，看是否得到了希望的结果，保存生成的语句，或者把语句粘贴到 SQL Editor 里面。

有了 SQL Modeler 的帮助，即使是初级的程序员也能编写出复杂而又高效的 SQL 查询来。

比如下面的例子；

生成步骤：

1. 选出表 emp, dep, manager
2. 用鼠标拖住 emp 的 emp_name 和 emp_no, dep 的 name, manager 的 name
3. 双击 criteria 窗口的 emp_no condition 列，弹出窗口，输入查询条件，比如 emp_no=3
4. 由于结果里面不希望 emp_no 出现，双击 emp_no 列的 visible，设置为 false
5. 这样就出来了需要的查询，现在可以运行查询得到结果，对查询 SQL 进行分析，或者在 SQL Editor 里面打开这个 SQL，或者保存为 SQL Model 文件，或者再进一步处理，比如加排序，汇总等。

DBA 方面的功能：

Toad 在 DBA 日常管理方面的能力是非常强大的，可以帮助 DBA 安全快速地完成许多日常工作，还有 Toad 特有的一些小工具能完成一些普通方法很难做到的功能。下面大概讲述了各个常用模块的主要功能：

a) Database Monitor

提供直观的 Oracle 数据库运行状况，包括下面这些详细信息：

可以看到逻辑 IO，物理 IO，各个状态的 session 数目，SGA 的利用率，共享池的使用状况，各类事件的等待状况，以及各类事件随着时间的变化情况

b) Instance Monitor

i. 可以用来检查本地 tnsnames.ora 文件里面定义的所有节点、Listener 和数据库的运行状态和版本等相关信息。

ii. 可以远程启动、关闭数据库（如果设置了 password 验证方式）

c) Unix Monitor：

i. 可以监控 Unix 机器的运行状况比如详细的机器启动时间，CPU 利用率，主要的进程以及这些进程占用的资源，主要的磁盘 IO 等。

d) Trace/Kill session

i. 方便的查看所有的连接用户相关信息：比如终端，程序，连接时间等

ii. 对某个用户进行跟踪，查看用户执行的 SQL 语句，查看语句的执行计划，

iii. 方便的查阅系统的 Lock 对象，有没有什么等待别的事务释放锁的情况，并能够轻易查出等待什么用户的什么事件结束，那个用户执行的是什么 SQL。

iv. 查看回滚段的使用状态，什么用户，什么语句，使用的什么回滚段，多少读写量等

e) Oracle Parameter

i. 可以方便的查看 Oracle 系统的参数：是否默认值，参数作用的简要描述，是否可

以用 alter session/alter system 来动态改变，对于那些可以动态改变的参数，可以双击鼠标来进行修改

f) NLS_PARAMETER:

- i. 方便的查看 session，instance，database 一级的 NLS 参数值，
- ii. 可以动态修改 session 的 NLS 参数值

g) Extents

- i. 可以查看某个用户下的表和索引的详细的存储情况，
- ii. 发现那些占用大量空间的表和索引，可以考虑使用相应的对策，比如分区
- iii. 发现那些即将达到 max_extents 的表和索引并可以及时纠正
- iv. 发现那些索引和表处于同一个表空间的情况，并可以及时纠正。

h) Tablespace

- i. 直观的查看每个表空间的利用率，总空间，已使用空间，剩余可用空间以及默认存储参数，和表空间是否需要整理等信息
- ii. 查看每个表空间的状态，组成的数据文件，以及每个数据文件的详细信息
- iii. 查看每个表空间里面存在的对象和这些对象的详细存储特性

i) Tablespace Map：

- i. 图形化的显示表空间里面各个表/索引所占用的空间，各个 segment 的大小，segment 是否连续等信息，非常直观
- ii.

j) Server Statistics:

- i. 可以非常直观的看到系统总体运行情况，迅速定位系统瓶颈所在
- ii. 可以查看系统范围内各类等待事件的状况，查看每个连接会话的状况，

k) Pinned Code：

- i. 我们知道，频繁执行的大的存储过程最好 Pin 起来，这样可以避免被 Oracle 在内存不足的情况下换出内存，避免动态加载，也避免 flush shared pool 的时候被清出内存。
- ii. 根据每个模块的大小，执行次数，决定是否要把某个存储过程 PIN 起来。一个按钮就搞定了。

l) Control Files:

可以查看当前系统有哪几个控制文件，以及控制文件内部记录的详细信息。

(实际上就是 V\$CONTROLFILE, V\$CONTROLFILE_RECORD_SECTION)

m) Redo Log Manager:

- i. 可以方便的查看 LOG 的组，当前日志，每个组里面的成员名字，大小，方便的增加、删除 log 组，切换当前日志，对日志归档进行具体的控制。
- ii. 可以在这个界面里完成几乎所有的对 redo log 的操作。

n) Repair chained rows:

- i. Chained rows 影响系统性能，处于系统性能的考虑，需要把 chained rows 修复成普通的表的行。使用 repair chained rows,就可以比较容易的做到这一点。选择需要分析的表，然后分析一下，看有没有 chained rows，如果有，点击 repair，Toad 就

帮你把那些 chained rows 给修复了。

o) Identify Space Deficits

如果系统出现 1659,1653 错误，就是一个不合格的 DBA。但是要即时预见这些错误，还是有一定的难度的。有了 Toad 的这个工具之后，一点某个表空间，就知道这个表空间里面的剩余自由空间是否允许任何表能够扩展到下一个 extent。并能够在发现问题之后，直接改正问题。

p) Estimate Table/Index Size

i. 数据库的物理规划中，表/索引的存储子句的定义是非常重要的，影响系统。但是没有什么工具的话，要为表定义合理的存储子句也不容易。

ii. 在 Toad 里面，专门有工具帮助你合理定义表/索引的存储情况：

1. 自己可以定义将来表中需要存储的数据的数目

2. 多种估计方式：

a) 根据对已有的数据的分析来定义表的大小（最精确，但是比较慢）

b) 根据表定义数据类型来定义表的大小

c) 根据数据字典中以前分析过的数据来估计大小（需要近期分析过该表）

q) Analyze All Tables

i. 基于 CBO 的 SQL 需要 CBO 提供最优化的执行路径来提高程序的执行效率，但是假如对表进行大量的 DML 操作之后，不及时更新表的统计数据，这时候 CBO 就不能选择最为有效的执行路径，导致程序执行效率低。

ii. 手工分析表比较麻烦，一个一个做。

iii. 用 Toad 的 Analyze Alltables 这个工具，可以一次分析整个模式下面的所有表，有丰富的控制选项，包括是采用估计还是完全，是否同时分析索引，估计的话，采用多大的百分比等。

r) Rebuild Table

i. 当表中的数据被反复大量的 DML 之后，而且 PCTFREE 和 PCTUSED 定义的不合理的情况下，表可能会变得非常破碎，占用大量的 extent，索引也会相当破碎，导致效率相当低下。

ii. 传统的方式是通过 exp/imp 来解决这个问题，但是步骤实在比较麻烦，手工操作也有一定的危险性。

iii. 使用 Rebuild Table 工具，轻松搞定，而且可以手工修改一些通过 EXP/IMP 无法修改的参数，比如 freelists ,freelists group, tablespace , initial 等，还能够更改列的名字。

s) Relocate Index

i. 我们知道，表和索引一般都需要分开存放到不同的表空间，这样比较便于管理，性能上也会有一定的提高，因为可以对表和索引同时存取。

ii. 但是实际上很多时候，由于这样那样的原因，常常忽略了索引单独存放这个基本概念，比如建立主键约束的时候没有加 USING INDEX 子句，后来 disable/enable 了主键约束/唯一性约束等，都会导致把索引直接存放到和表同一个表空间里面。

iii. 手工发现这些索引，再逐个移动非常麻烦，容易出错，在 Toad 里面，就很容易根据用户/表空间来发现那些不恰当存放的索引，可以批量移动这些索引，并且允

许重新定义索引的存储子句。

t) Generate Schema Scripts :

- i. 可以用来生成整个用户下所有/部分对象的 DDL 语句。可以把结果保存为 sql 语句或者保存为 TOAD 的自定义格式。
- ii. 可以用来复制表结构，（EXP/IMP 有非常严重的缺陷，就是从一个大的数据库倒出的表结构，无法导入到一个小的数据库里面，因为里面包含了原来表的存储信息，即使选择 rows=N，也要在导入数据库里面占用相当大的空间）

u) Compare Schema :

- i. 在开发的时候，有时候同时在几个开发库上开发，但是最后两个库的结构不完全一样，很难比较
 - ii. 用 TOAD 的这个功能，可以对两个数据库的模式（用户）下面的所有对象（包括存储过程）进行比较，可以选择具体的比较内容。
 - iii. 可以通过直接打开两个到各自数据库的连接，或者根据 Toad generate schame scripts 生成的文件进行对比。
 - iv. 可以直接生成同步两个数据库的 sql 脚本，给出汇总的区别简要和详细报告等。
-

几乎是学习 TOAD 的唯一中文使用说明，我再也找不到比这个更好的了，特跟大家分享。