

# 虚谷数据库

## 管理工具 V11.2.2

### 用户指南

文档版本 01

发布日期 2024-04-19



版权所有 © 2024 成都虚谷伟业科技有限公司。

## 声明

未经本公司正式书面许可，任何企业和个人不得擅自摘抄、复制、使用本文档中的部分或全部内容，且不得以任何形式进行传播。否则，本公司将保留追究其法律责任的权利。

用户承诺在使用本文档时遵守所有适用的法律法规，并保证不以任何方式从事非法活动。不得利用本文档内容进行任何侵犯他人权益的行为。

## 商标声明



为成都虚谷伟业科技有限公司的注册商标。

本文档提及的其他商标或注册商标均非本公司所有。

## 注意事项

您购买的产品或服务应受本公司商业合同和条款的约束，本文档中描述的部分产品或服务可能不在您的购买或使用范围之内。由于产品版本升级或其他原因，本文档内容将不定期进行更新。

除非合同另有约定，本文档仅作为使用指导，所有内容均不构成任何声明或保证。

## 成都虚谷伟业科技有限公司

地址：四川省成都市锦江区锦盛路 138 号佳霖科创大厦 5 楼 3-14 号

邮编：610023

网址：[www.xugudb.com](http://www.xugudb.com)

# 前言

## 概述

本文档介绍了虚谷数据库管理工具的基本功能和管理操作。

## 读者对象

本文档主要适用于以下用户：

- 数据库开发人员
- 数据库维护人员
- 数据库管理员

## 符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 <b>注意</b>	用于传递设备或环境安全警示信息，若不避免，可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。
 <b>说明</b>	对正文中重点信息的补充说明。“说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

## 修改记录

文档版本	发布日期	修改说明
01	2024-04-19	第一次正式发布

# 目录

1	XuguManager 基本操作 .....	1
1.1	启动 XuguManager .....	1
1.2	退出 XuguManager .....	1
1.3	创建实例组 .....	1
2	虚谷数据库管理操作 .....	5
2.1	表操作 .....	5
2.1.1	创建表 .....	5
2.1.2	修改表 .....	11
2.1.3	修改表数据 .....	12
2.1.4	查询表 .....	13
2.2	视图操作 .....	15
2.3	序列值操作 .....	16
2.4	包操作 .....	17
2.5	过程和函数操作 .....	19
2.6	触发器操作 .....	21
2.7	索引操作 .....	23
2.8	同义词操作 .....	24
2.9	角色操作 .....	25
2.10	用户操作 .....	26
3	登录管理 .....	31
4	用户自定义数据类型操作 .....	32
4.0.1	数据库对象导出 .....	35
4.1	表注释添加 .....	36
4.1.1	创建表时添加表注释和列注释 .....	36
4.1.2	创建表完成后添加表注释或列注释 .....	37
5	SQL 编辑窗口 .....	41

---

6	数据库备份恢复 .....	48
---	---------------	----

# 1 XuguManager 基本操作

## 1.1 启动 XuguManager

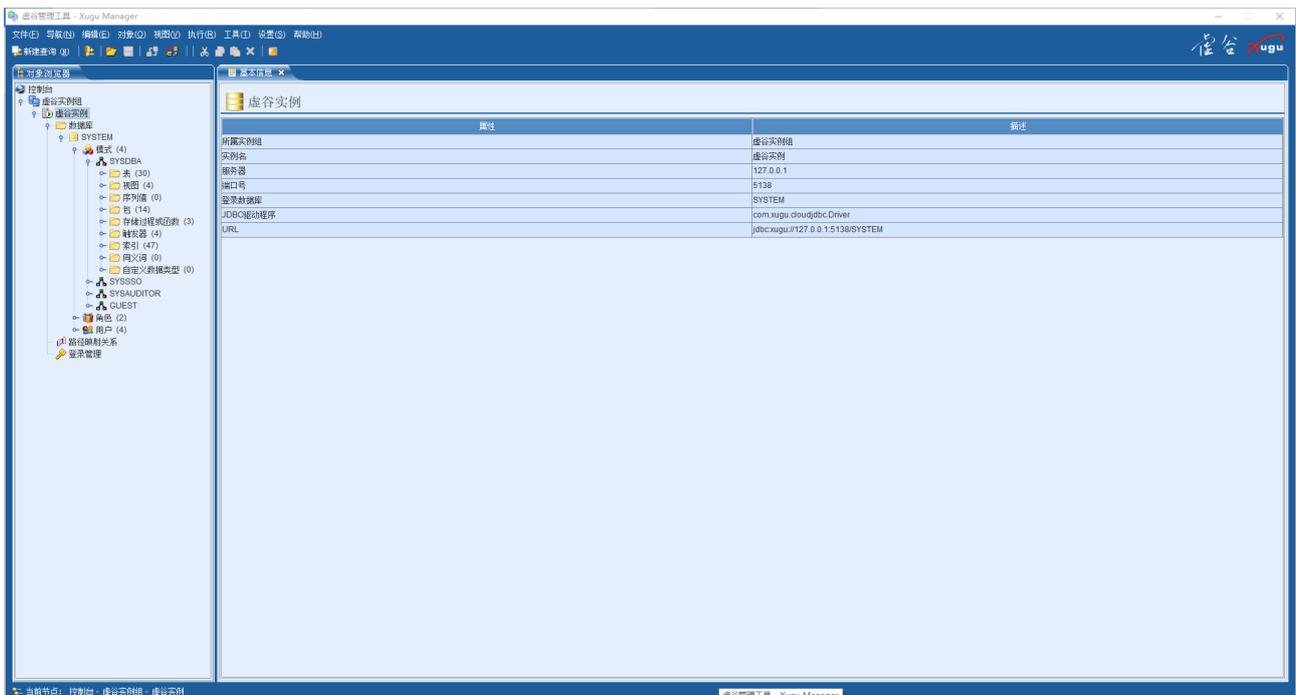
首先，确认虚谷数据库服务端已启动，客户机上所需环境已配置好，然后直接执行安装目录中“XuguManager”目录下面的“XuguManager.exe”来启动 XuguManager 程序。

### 说明

请确保客户机已安装 jre1.8 运行环境或“XuguManager.exe”文件父目录同级目录下有 jre1.8 运行环境，否则程序无法启动。

XuguManager 主界面如图 1-1 所示。

图 1-1 管理软件主界面



## 1.2 退出 XuguManager

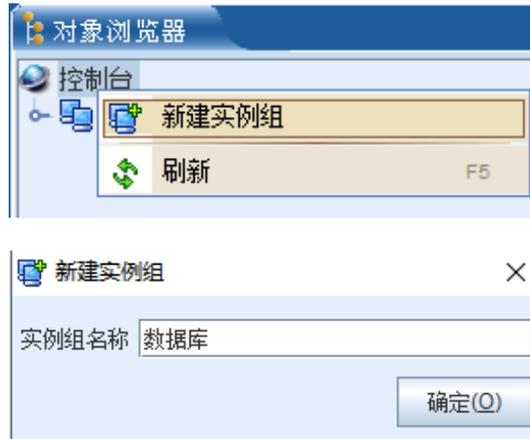
单击“文件”菜单下的“退出”按钮，或者单击右上角“关闭”按钮即可退出 XuguManager 程序。

## 1.3 创建实例组

### 操作步骤

1. 首先右键单击“控制台”。

弹出新建实例组对话框并创建一个“数据库”实例组。



2. 完成创建实例组后，可创建单个实例。

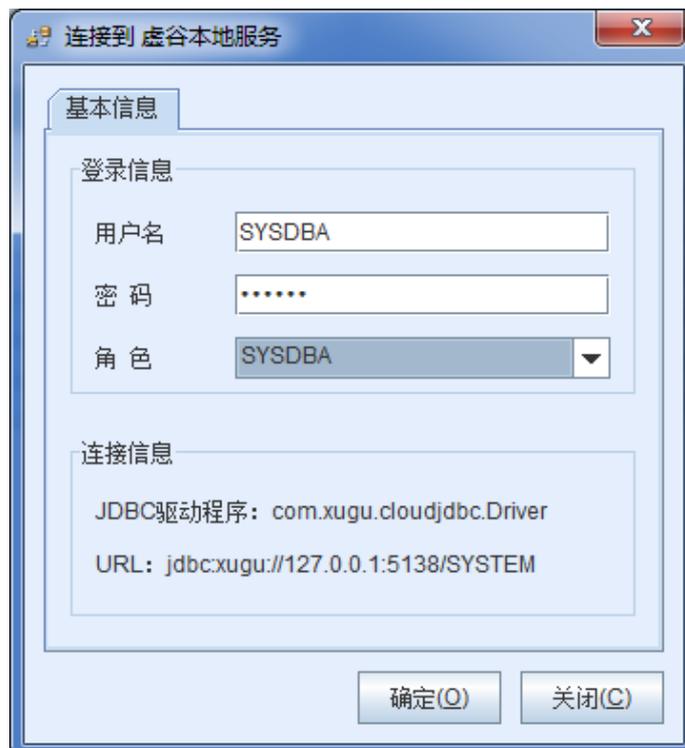
创建一个名为“本地服务”的实例，其中服务器信息指服务器 IP 地址（127.0.0.1 本机）和端口号（默认 5138），如图 2 所示。如果在安装虚谷数据库后，修改了端口，则“端口号”处填写修改后的端口号。虚谷数据库服务端的端口是在安装目录下面的“xugu.ini”文件中的“listen\_port = 5138; 侦听端口”项进行配置。





3. 实例建立之后，即可连接启动后的数据库。





连接数据库时需要对角色进行选择，SYSDBA、DBA、NORMAL 三种角色代表不同的身份。不同用户使用不同的角色，SYSDBA 角色仅在 SYSDBA 管理员登录时使用，能查询以“SYS\_”开头的系统表。DBA 角色针对具有 DBA 权限的用户登录使用，能查询以“DBA\_”开头的系统表。NORMAL 针对普通用户使用。登录时未选择正确的角色，会报“权限不够”错误。

# 2 虚谷数据库管理操作

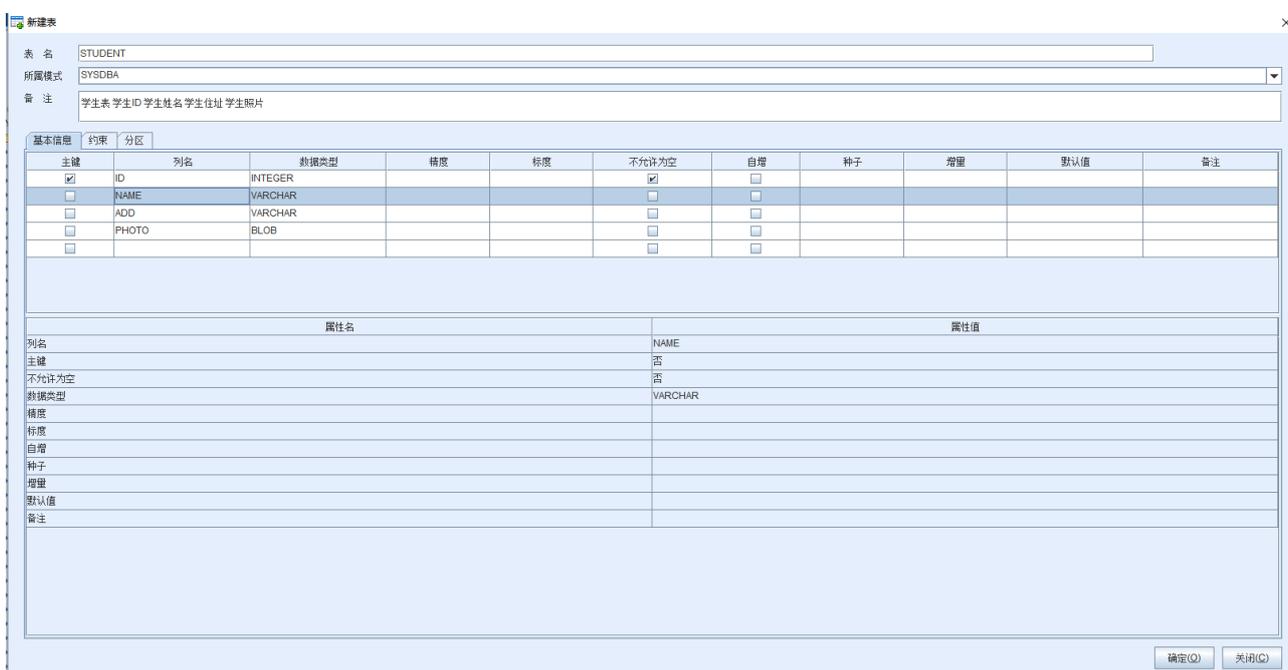
## 2.1 表操作

### 2.1.1 创建表

创建表分为基本信息、约束、物理特性和分区四个部分讲解。

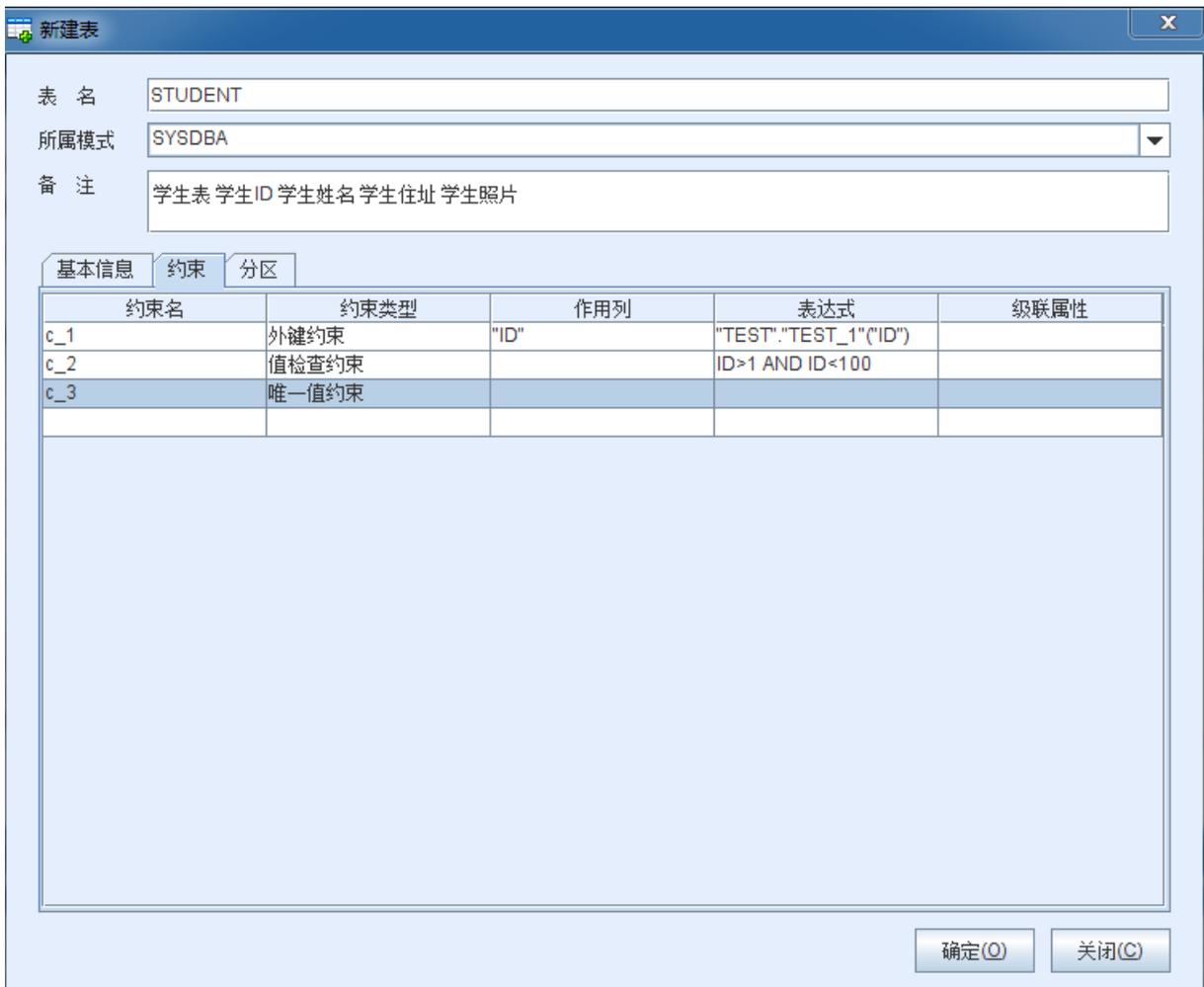
#### 基本信息

首先在基本信息中，现在给出的信息比以前的版本更加丰富，如种子、默认值等功能。



#### 约束

约束用于限定表中某些字段的取值，可以针对单个字段，也可以针对多个字段，它是保障整个数据库系统逻辑一致性的重要手段。



下面介绍几种约束的创建。

- 外键约束。

外键约束是用于与另一个表建立关联。通过与另一个表中的主键列（或者具有唯一值约束列）关联，这些列构成了另一个表的外键。



- 值检查约束。

值检查约束是用来指定某列的取值范围，它通过限制输入到列中的值来强制域的完整性。

值检查约束可以作用在多列上。值检查约束表达式的填写标准是一个关系表达式，如 `ID>10 AND NAME= 'ZHUFENG' AND ADDR LIKE 'CD'` 。

- 唯一值约束。

用来限制不受主键约束的列上的数据的唯一性，即表中任意两行在指定列上都不允许有相同的值。唯一值约束与主键约束的区别在于唯一值约束允许在该列上存在 NULL 值，而主键约束限制的更加严格，不但不允许有重复，而且也不允许有空值。



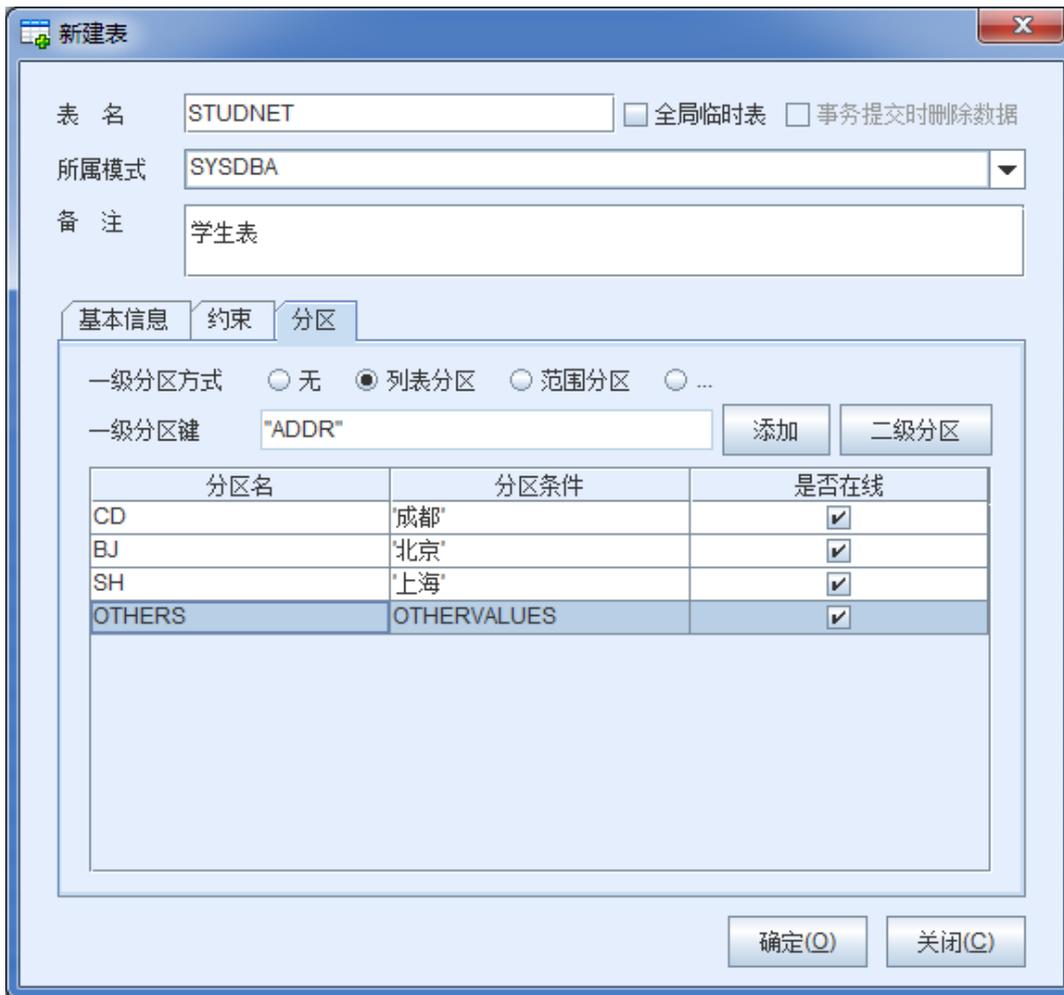
## 分区

表分区是指在创建表时，按照一定的划分条件，将表切分成若干个子表，在逻辑上，分区表与单一表具有相同的特性，对分区进行插入、更改、删除、选择等操作的命令与对单一表的操作命令完全一样，但分区表的各个分区拥有独立的存储实体，因此，分区表的各个分区可以存放在不同的存储空间上，从而实现表在物理上的分割。

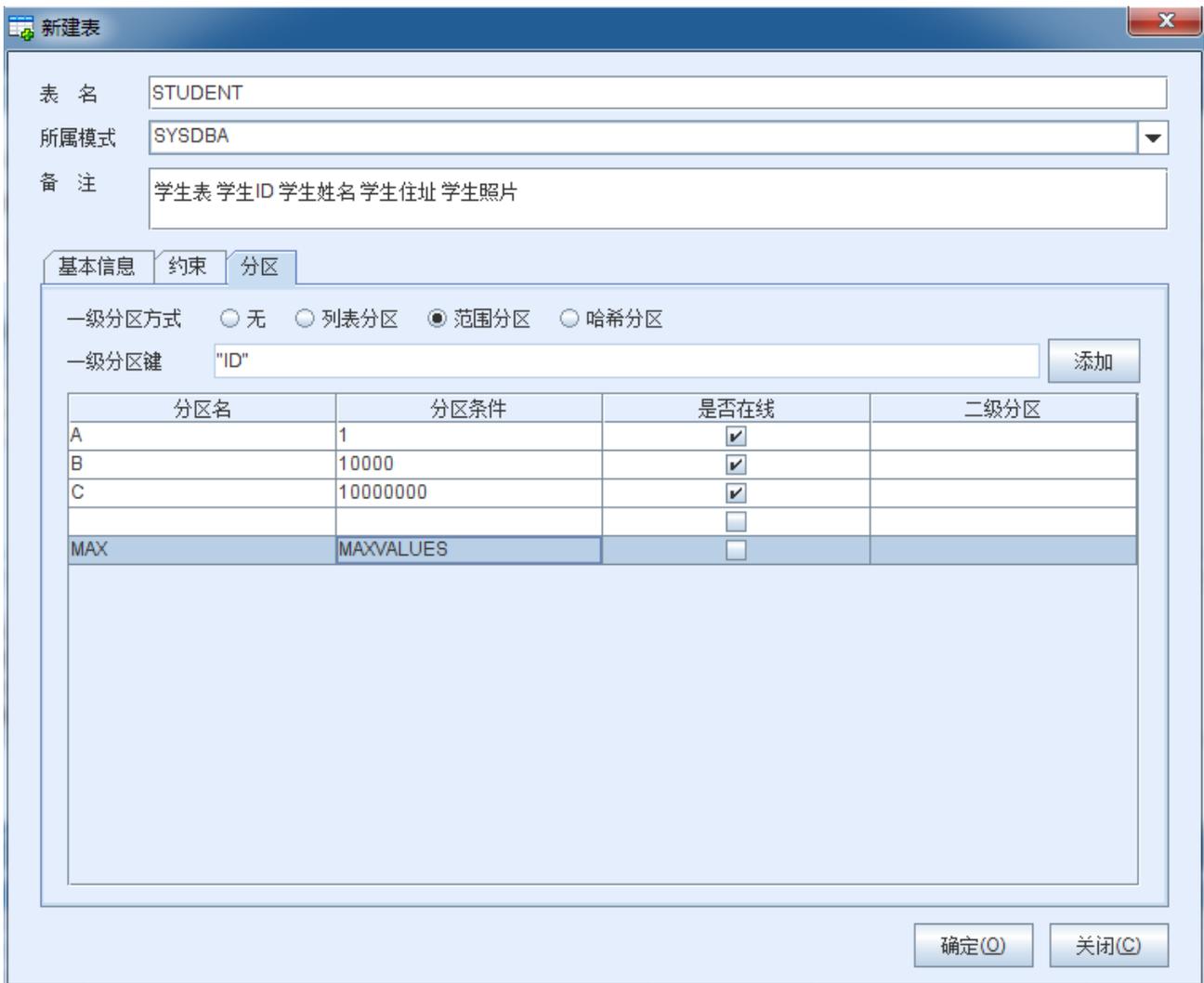
虚谷数据库提供了两级三种分区方式：

- 一级分区的方式有：列表分区、范围分区和哈希分区。
- 二级分区的方式有：列表分区、范围分区和哈希分区。

下面介绍的是列表分区。在这里要注意的是分区条件的选择，列表分区一般针对的是无规律数据，所以一般的列表分区的分区键是选择字符数据类型。在下面的例子中，选择 ADDR 字段作为分区键，二级分区不选就表示不使用二级分区。可以看到在下面还有一个 OTHERVALUES 分区，这是一个异常分区，表示在前面的分区中不符合分区条件的记录都被存放在 OTHERVALUES 分区中。

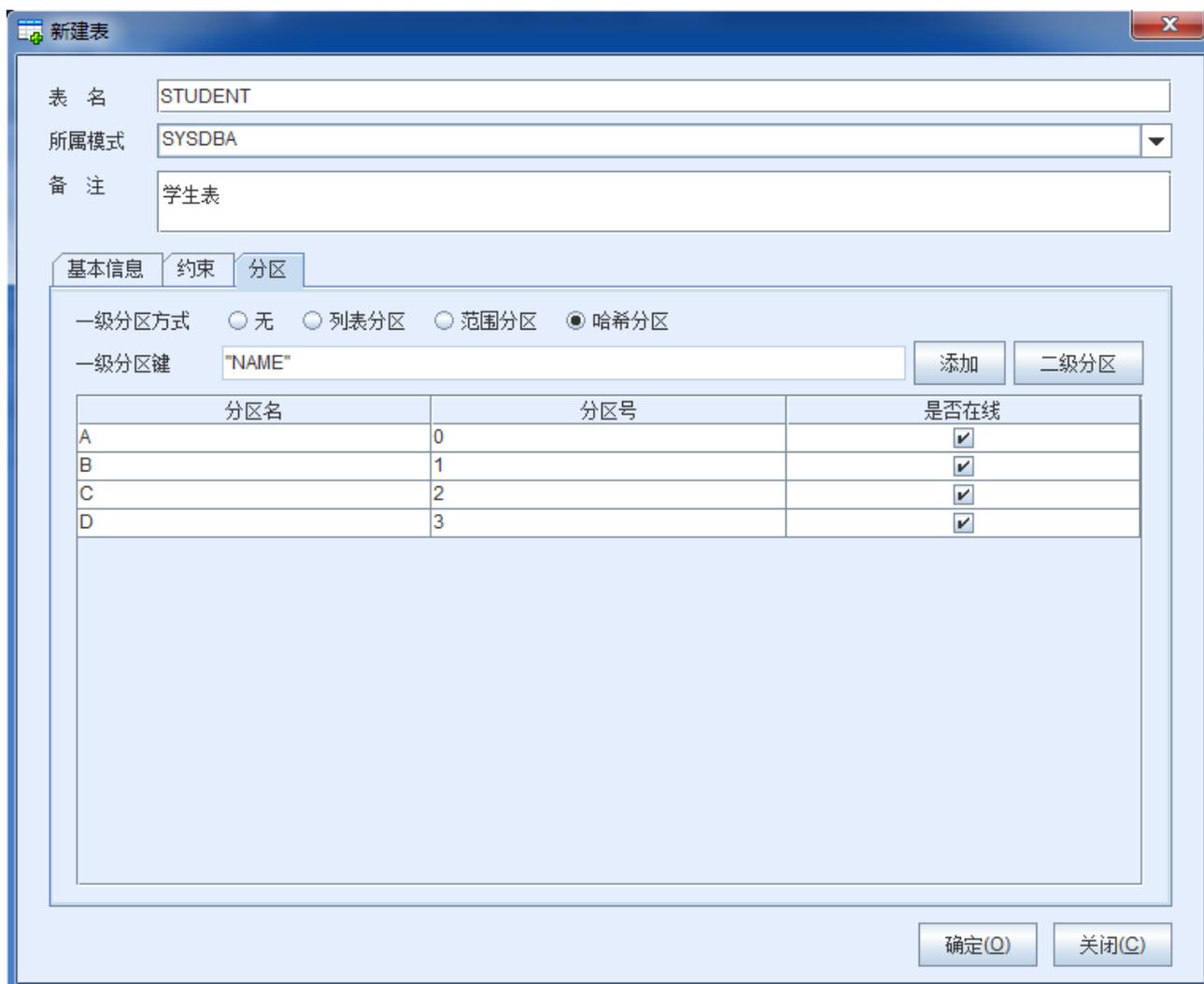


范围分区一般是针对数值数据类型的一种分区，如 1-100，101-200 这样的数据，一般选择范围分区。在范围分区中，可以看到有一个 MAXVALUES 分区条件，MAXVALUES 分区就相当于列表分区中的 OTHERVALUES 分区。在下图所示的例子中分区条件表示从负无穷到 0，1 到 9999，10000 到 9999999，10000000 到 MAXVALUES，这涵盖了所有的整数分区。



哈希分区则是按分区字段的散列值为键将记录分散到各个分区中，除非查询的条件表达式中含有与分区键字段相关的等式，否则查询不能被裁剪为只针对表的一个或少数几个分区，在此情况下，系统主要通过启用并行查询算法来提高查询的性能，因此，在以下情况下，可以考虑使用哈希分区：

- 不适合使用列表分区和范围分区。
- 作为分区条件的关键字段的取值繁多，其散列值分布较均匀。
- 查询的条件中与分区键字段相关的关系表达式是等式。



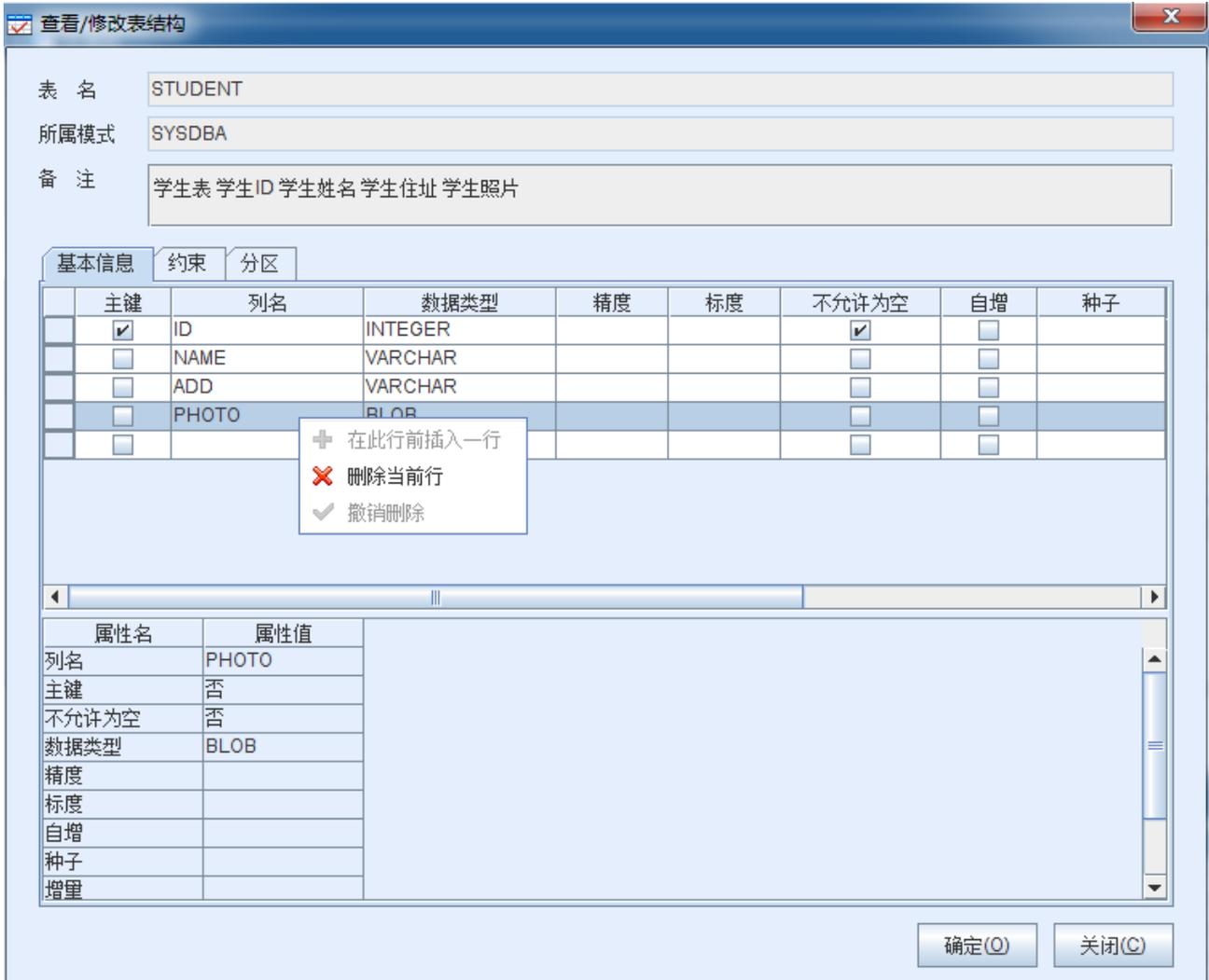
可以在列表分区当中建立二级分区:



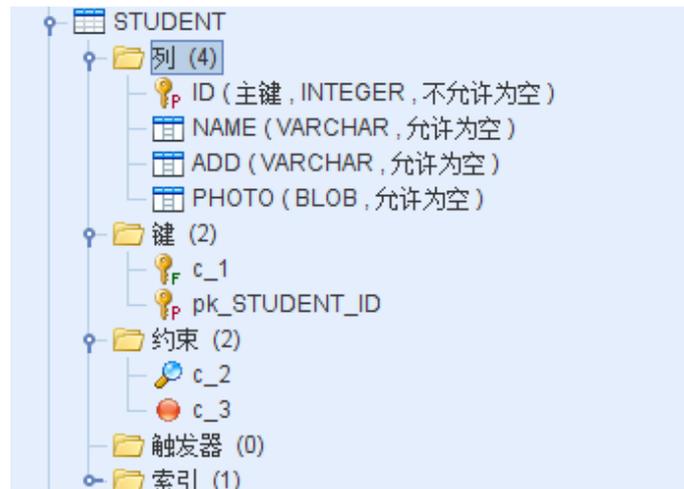
## 2.1.2 修改表

修改表就是对表名称、字段名、字段类型和约束等相关属性的修改。表的物理储存不可以修改，所以在设计表时需考虑字段的冗余。

下图是删除表中的 PHOTO 字段。



在表下面的节点中，同样可以对列做修改。只是在表结构修改界面中，是全面的修改，可以有列的增删改，也可以添加和删除约束。

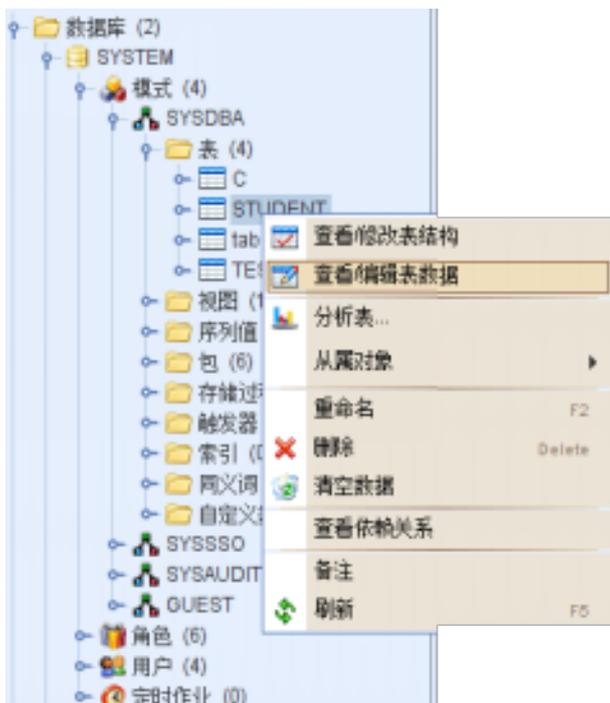


下图中添加了一个“c\_4”的约束。



### 2.1.3 修改表数据

可以通过在对象浏览器中右键点击对象对表数据进行编辑和查看。

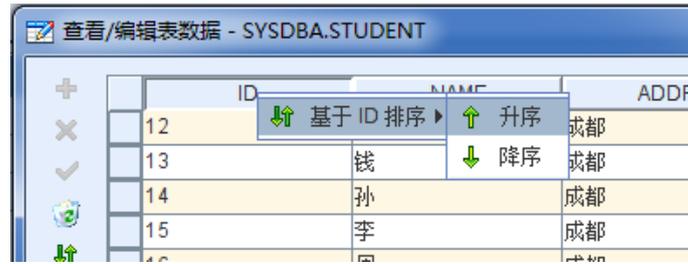


## 2.1.4 查询表

查询分为普通查询与过滤查询。

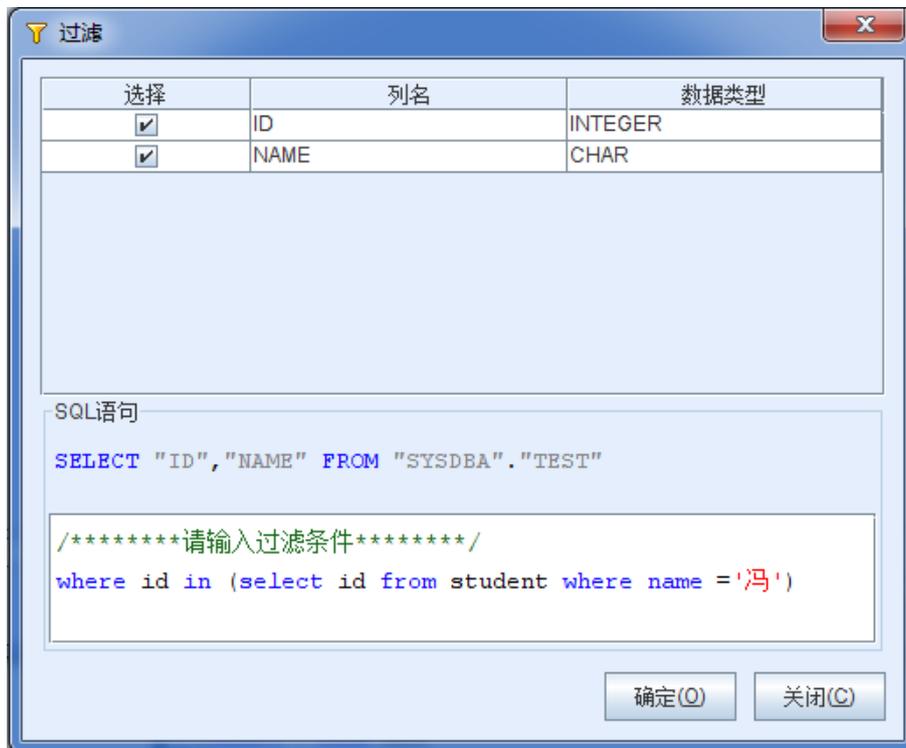
### 普通查询

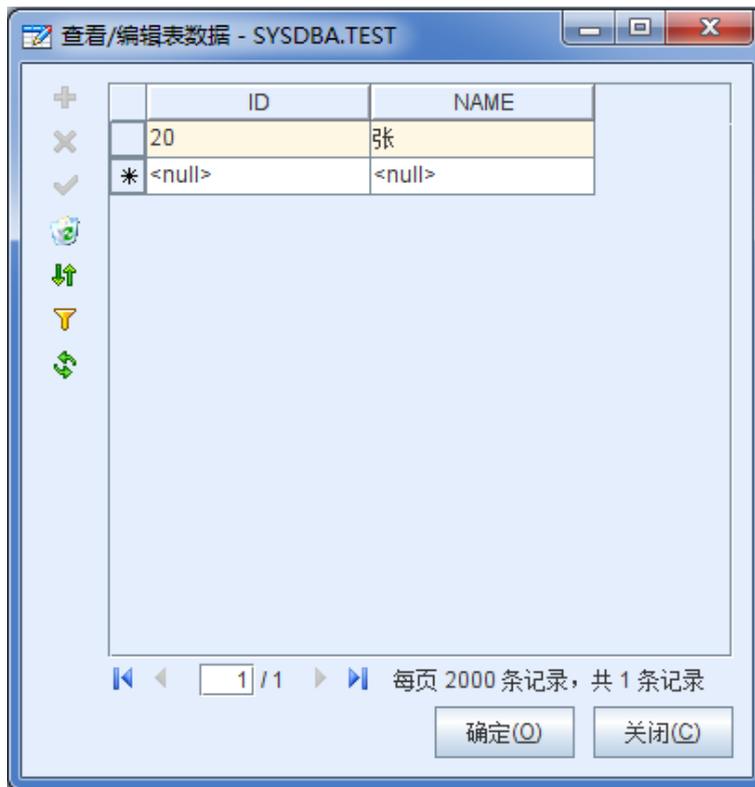
通过 **SELECT** 命令加上字段名称对表数据进行筛选，得到所有数据，并可以通过右键点击字段名称进行排序处理。



### 过滤查询

所谓过滤就是比排序更复杂的查询，如 ‘SELECT ID,NAME FROM STUDENT WHERE ID>15 AND NAME =’ 赵’ ; ‘ SELECT ID,NAME FROM TEST WHERE ID IN(SELECT ID FROM STUDENT WHERE NAME =’ 冯’ );’ 。



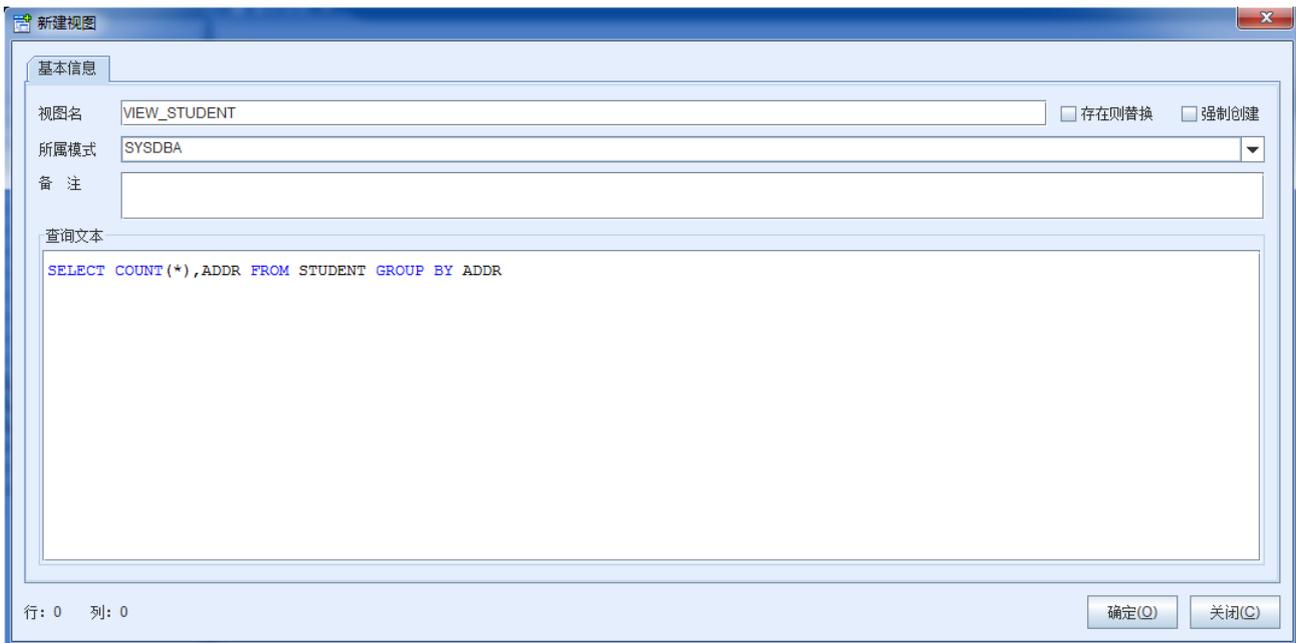


## 2.2 视图操作

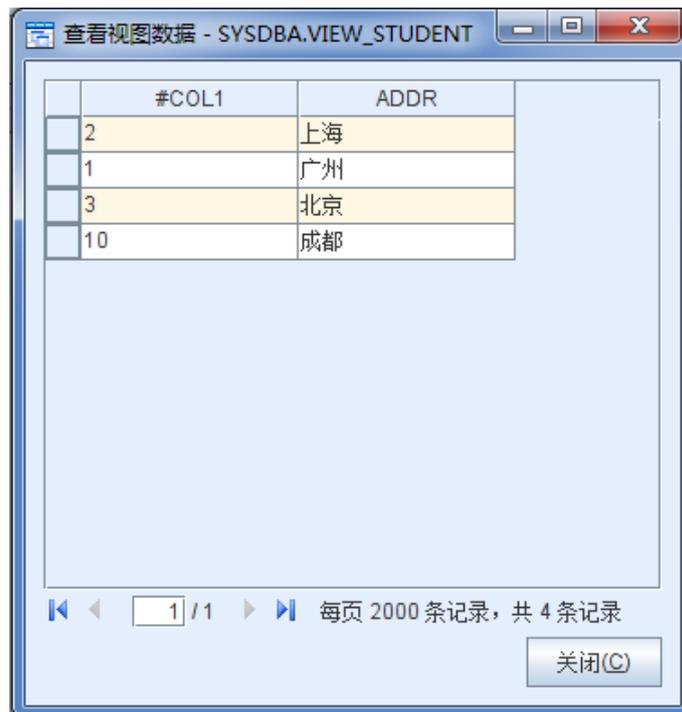
视图也称逻辑表，是建立在查询基础上的非物理存在的表，其基表可以是一个或多个物理表，视图中的数据随基表中数据变化而变化，视图一旦被定义后，在查询中其地位与物理表相当，虚谷数据库允许视图作为其它视图的基表，同时也允许针对单表视图进行记录的插入、修改与删除，多表连接视图则不能进行数据变更。

视图的创建由用户自己来定义 SQL 语句，这样主要是提高其灵活性。

下面是创建在 student 表上的一个分组统计视图。



创建视图后，对视图进行查看。

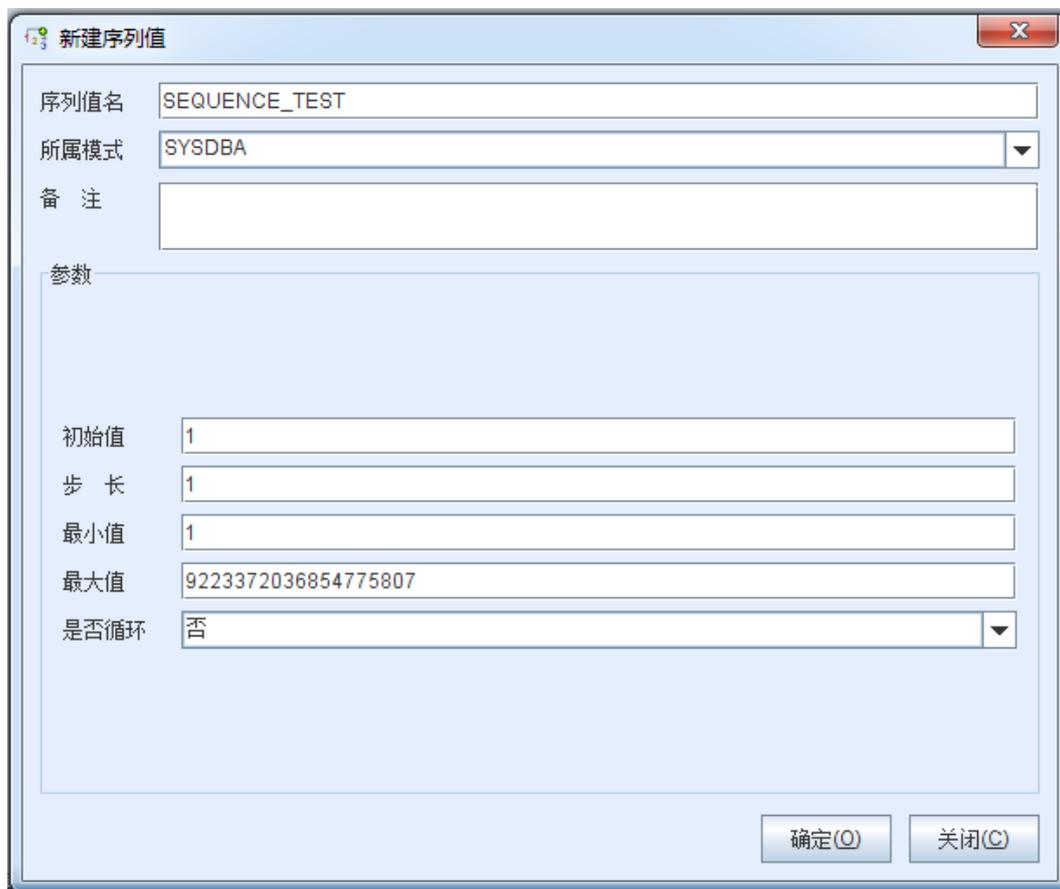


## 2.3 序列值操作

序列值发生器用于产生整数序列值，其产生的整数按一定规律递增或递减，并保证在多用户并行调用时其产生的整数不出现重复。序列值发生器的主要用途是：在多用户环境中，当各个用户或客户端程序需要用一个数值唯一标识某个记录或其它对象时，用户或客户端程序自身很难生成一个有序的且与其它用户或客户端程序生成的值不相同的整数，因为，任何客户或应用程序皆无法预测别的用户或客户端程序将要产生的数值，只有集中产生的数据才能保证不

出现重复，数据库应用程序的集中点显然是数据库服务端，因而，由数据库服务端产生唯一值数值是理所当然的。

创建序列值如下图所示：



下面重点讲解一下序列值的使用：

当创建序列值发生器之后，就可以在 DML 语句中使用它了，使用的方式为，将 SEQUENCE\_NAME.NEXTVAL 作为一个值表达式运用到语句的各个表达式中，SEQUENCE\_NAME.NEXTVAL 用于取得名为 SEQUENCE\_NAME 的序列值的当前值，在取值的同时，将序列值的当前值修改为下一个产生值，整个过程在封锁下进行，是原子性的。INSERT INTO TEST VALUES(SEQ\_1.NEXTVAL)，这样就自动在表中插入一个连续自动增长的值。（对于 SEQUENCE 使用的函数在函数解析部分说明）。

## 2.4 包操作

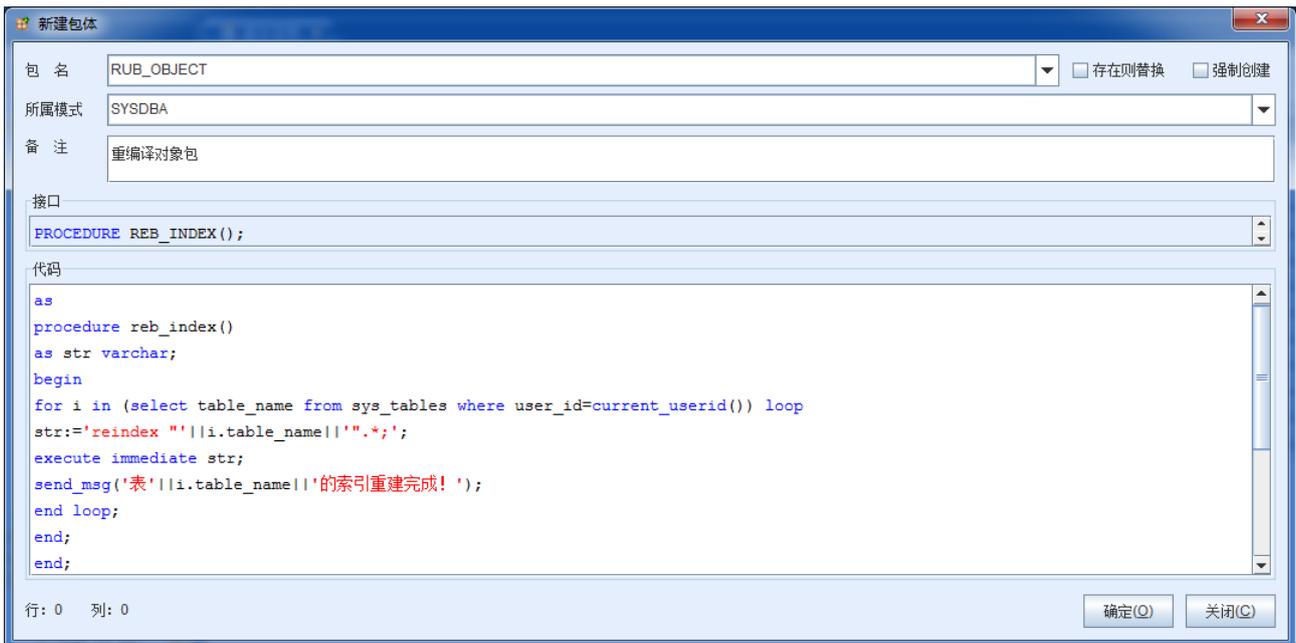
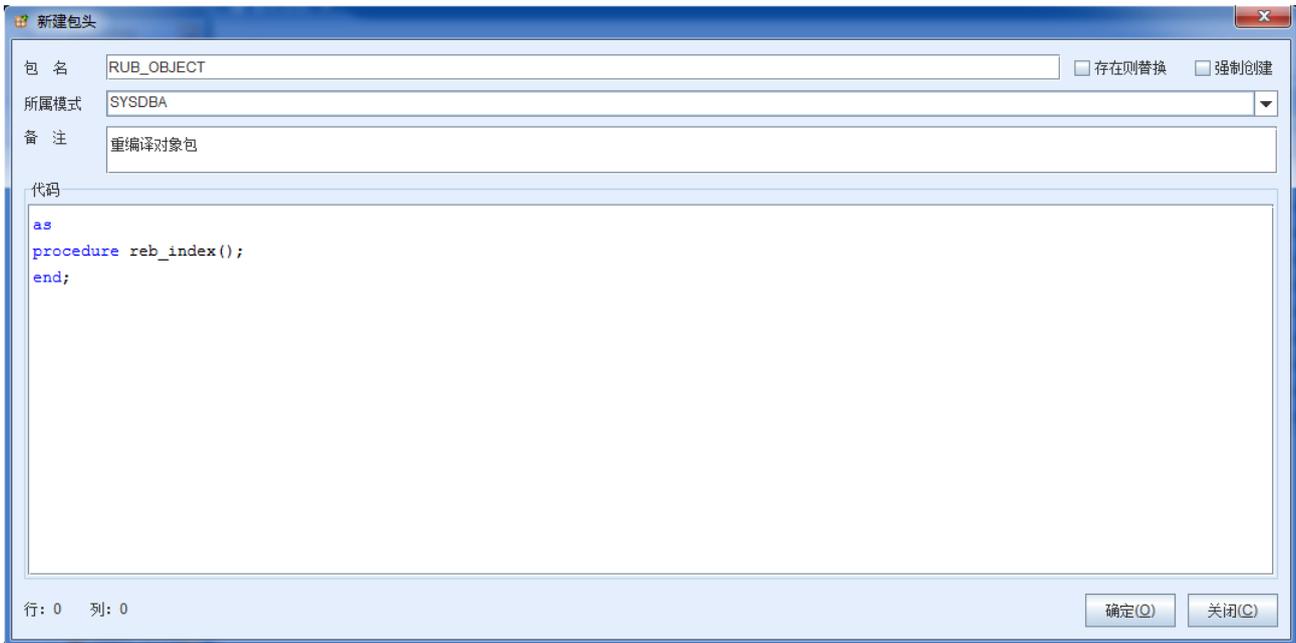
包，就是将一部分相关的 PL/SQL 类型、项目和子程序等有计划的组织起来封装在一起。包通常包含两个部分：规范和主体。有时候包的主体部分并不是必须存在。

包的规范又叫包头，是应用程序的接口部分 (interface)。它通常包含类型、记录、变量、常

量、异常、游标和子程序等的声明。任何可以在块中声明的语句都可以在包中使用。这些声明对于应用程序是可见的，应用程序可以调用它们。包体部分完整的定义了包的规范部分声明的游标和子程序，从而实现包的规范。包体内的内容对于外部的应用程序来讲是不可见的，包体可以说是包头的代码部分。

可以把包的规范部分看成是一个操作或应用的接口部分，而包体被封装在一个黑匣子里，它的作用是实现包的规范，可以调试、改进和替换包体而无须改变包的规范部分。

下面创建一个 RUB\_OBJECT 包，包中是一个重建表上索引的存储过程。包的使用是“包名称.过程或者函数名称”。



📖 说明

包的修改只是在包体中进行，包头不可以进行修改。一般在创建一个比较大的包时，包头的函数或者过程必须写完全，否则在“REPLACE”（界面上的“存在则替换”）时包体将找不到包头，从而丢掉包体，造成数据的丢失。

在控制台执行包的内部过程“rub\_object.reb\_index()”。

```
Connect to 127.0.0.1:5138 SYSTEM SYSDBA
Connect ok.

SQL> execute rub_object.reb_index();
表TEST的索引重建完成!
表DS的索引重建完成!
表SYNONYM_TEST1的索引重建完成!
表TY的索引重建完成!
表STUDENT的索引重建完成!
表ORDERS的索引重建完成!
表GOODS的索引重建完成!

Execute successful.
Use time:2 ms.

SQL>
```

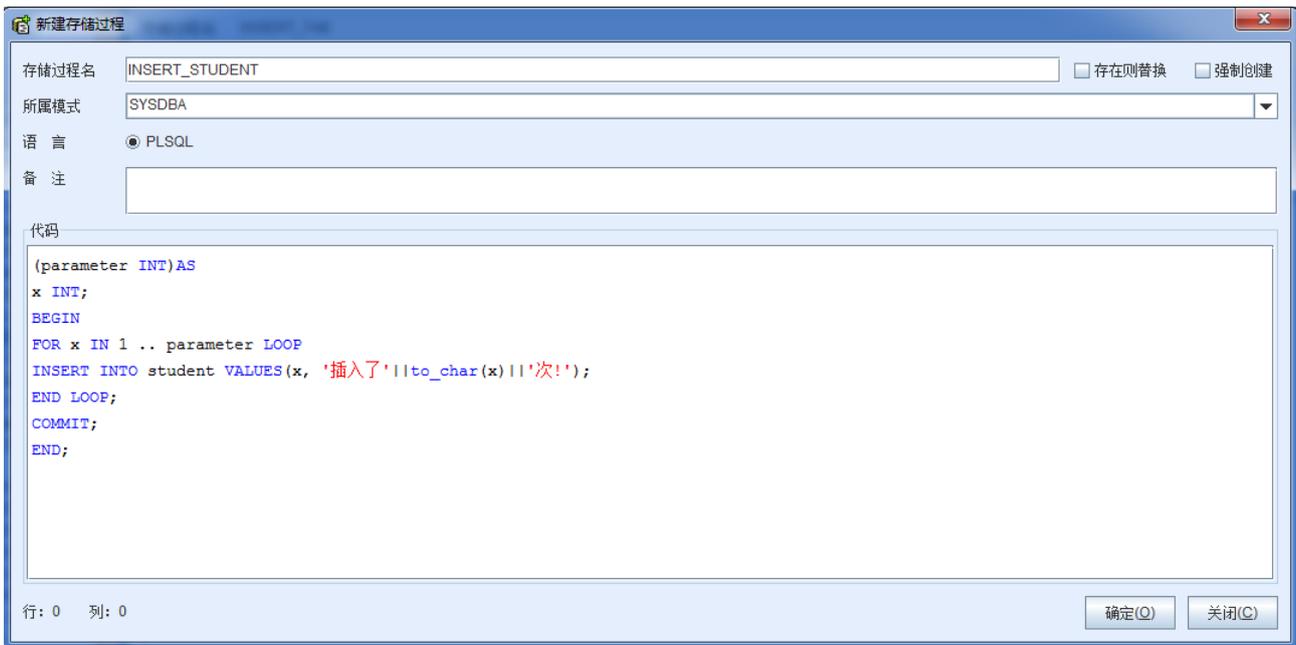
## 2.5 过程和函数操作

一般说过程是为了执行一定的任务而组合在一起的 SQL 或 PL/SQL 语句集。通常过程包含声明部分、执行部分和异常处理部分。其中执行部分是必须的，声明部分和异常处理部分要根据实际情况而定。

声明部分包含常量、变量、异常、游标和子程序等的声明，声明部分所有的声明都是局部声明，当过程结束后，这些声明就终止，其存在的周期也就结束。执行部分包含给声明部分的变量赋值、为达到一定目的操作和执行 SQL 或 PL/SQL 语句。异常处理部分包括对存储过程执行中出现的异常进行操作以及对自定义异常的一些处理。

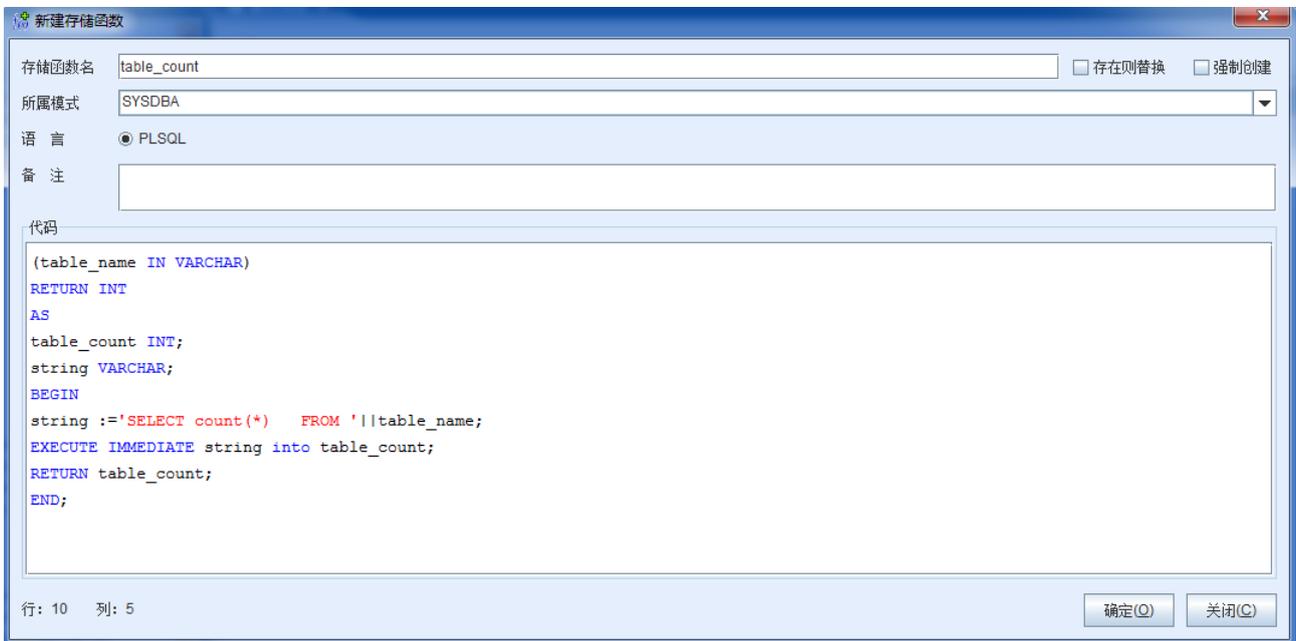
虚谷数据库使用的 PL/SQL 语言与 Oracle 使用的 PL/SQL 语言比较接近，但也有许多不同的地方。在虚谷数据库中，PL/SQL 除了作为过程及函数编程语言外，同时也是编写触发器的过程体所使用的语言。

下面创建一个生成数据的存储过程。生成记录的条数作为参数由用户在执行过程时传入。



函数和过程类似。也可以包含声明部分、执行部分和异常处理部分。和过程不同的是，过程是为了执行一定的任务而组合在一起的 SQL 或 PL/SQL 语句集，而函数是返回一个值。函数的调用是表达式的一部分，而过程的调用本身就是一条 PL/SQL 语句。

下面创建一个通过表名称取得表记录数的函数。表名称作为参数传递进去，返回表的记录条数。



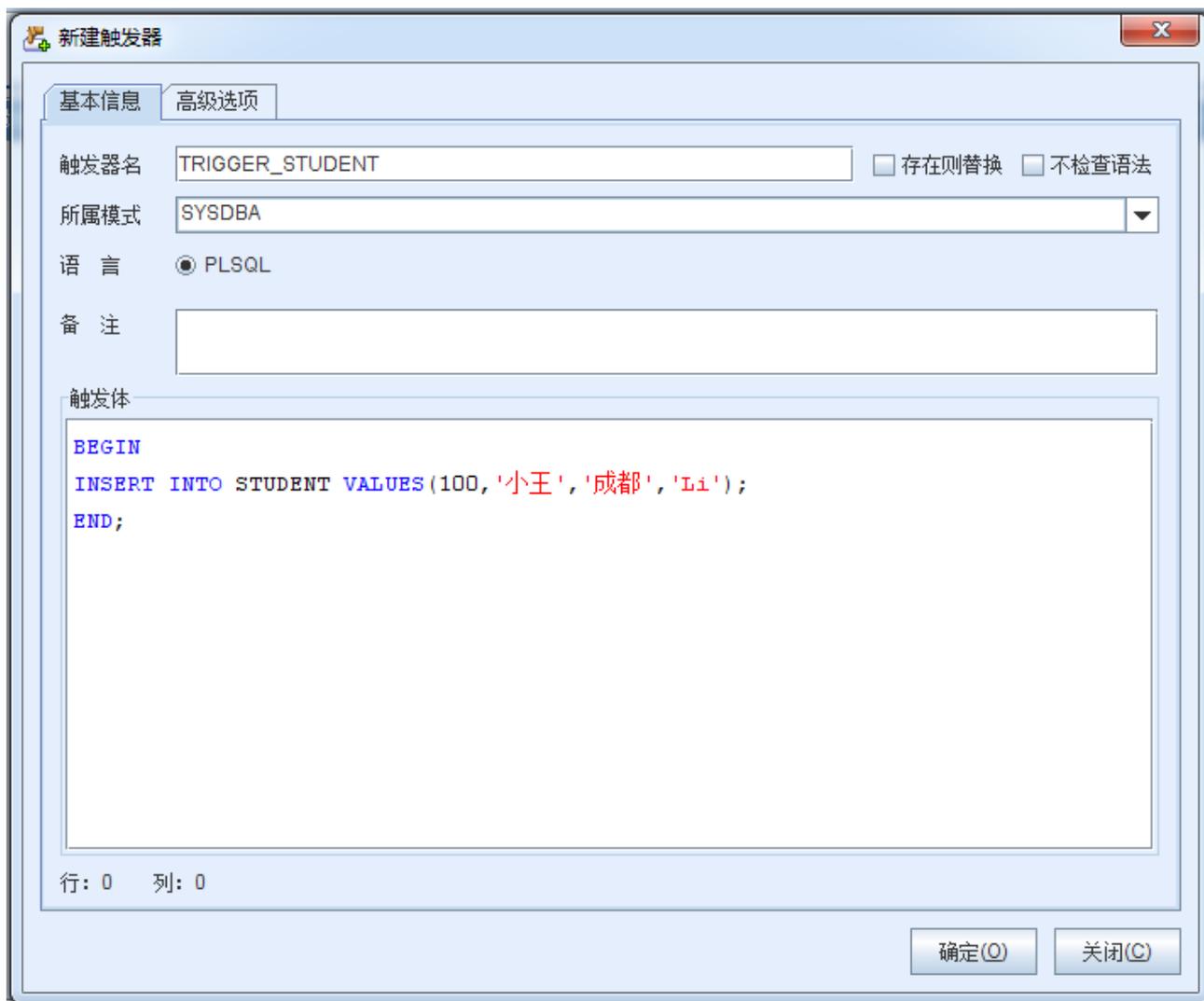
控制台执行查询语句，嵌入函数。

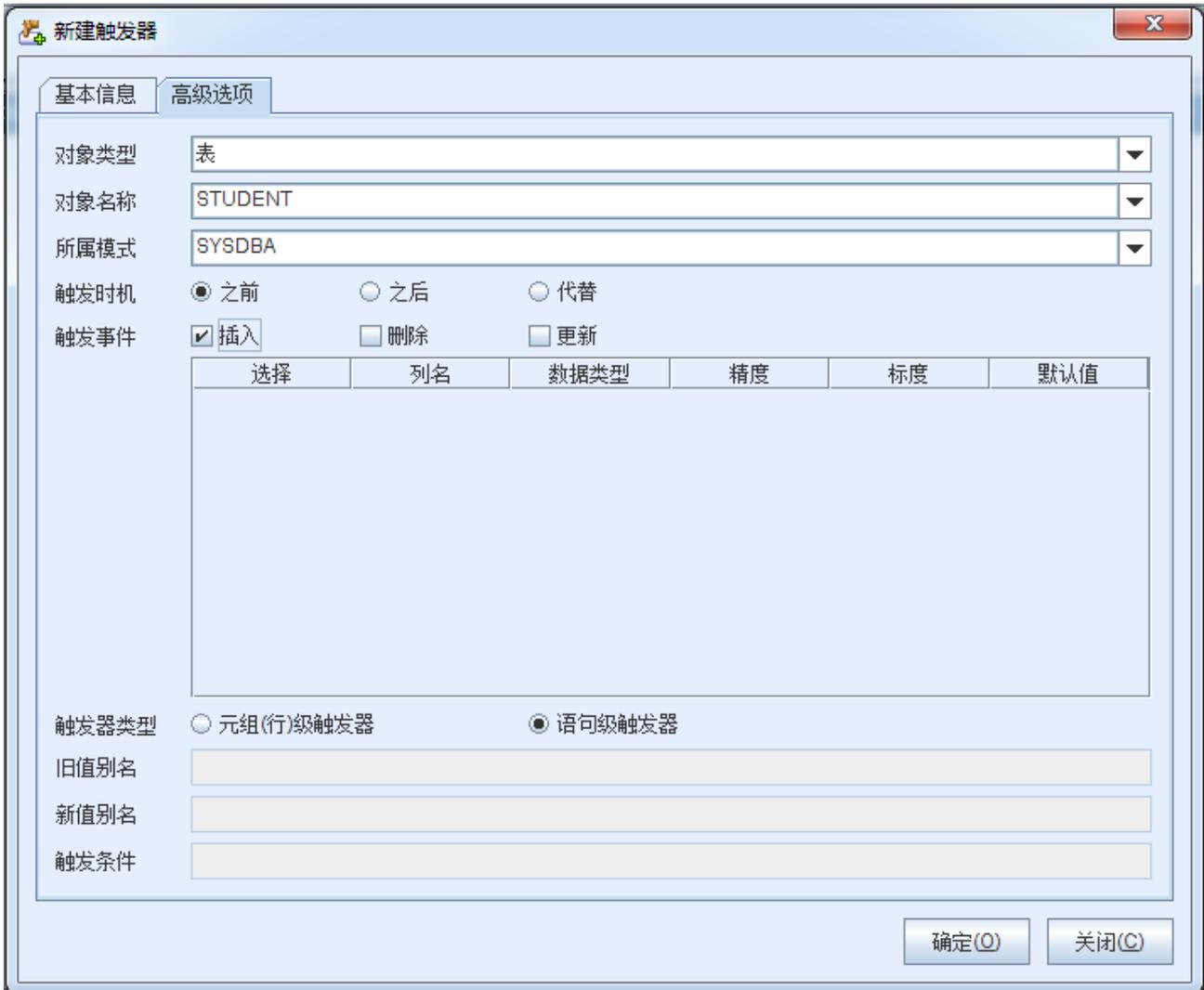
```
SQL>select table_count('student');  
  
EXPR1 :  
-----  
110000 :  
  
Total 1 records.  
  
Use time:10 ms.  
  
SQL>_
```

## 2.6 触发器操作

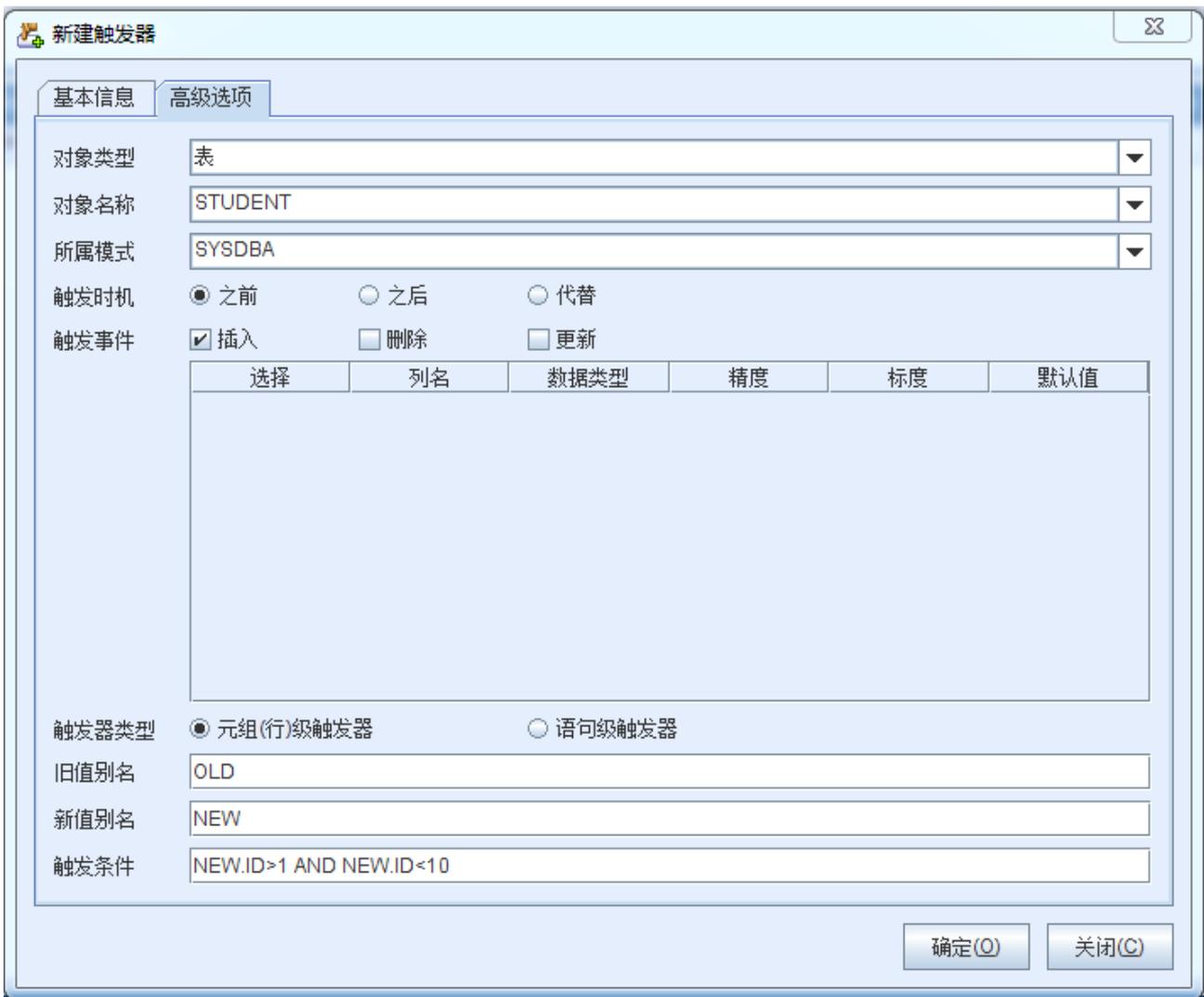
触发器是一种特殊类型的存储过程，它在插入、删除或修改特定表中数据时起作用。通过触发器维持不同表中逻辑上相关数据的一致性，保持数据的相关完整性。

触发器创建如下图所示：





元组级触发器，即针对记录级触发。NEW 和 OLD 分别针对新的记录和存在的记录。在新值、旧值别名处可以使用默认值 NEW 和 OLD 作为别名，也可以用户自己定义新的别名，如新别名用 sen 表示，旧别名用 den 表示，那么在后面的记录定位就是 sen.id 或者 den.id，系统不会再认 NEW 和 OLD。



## 2.7 索引操作

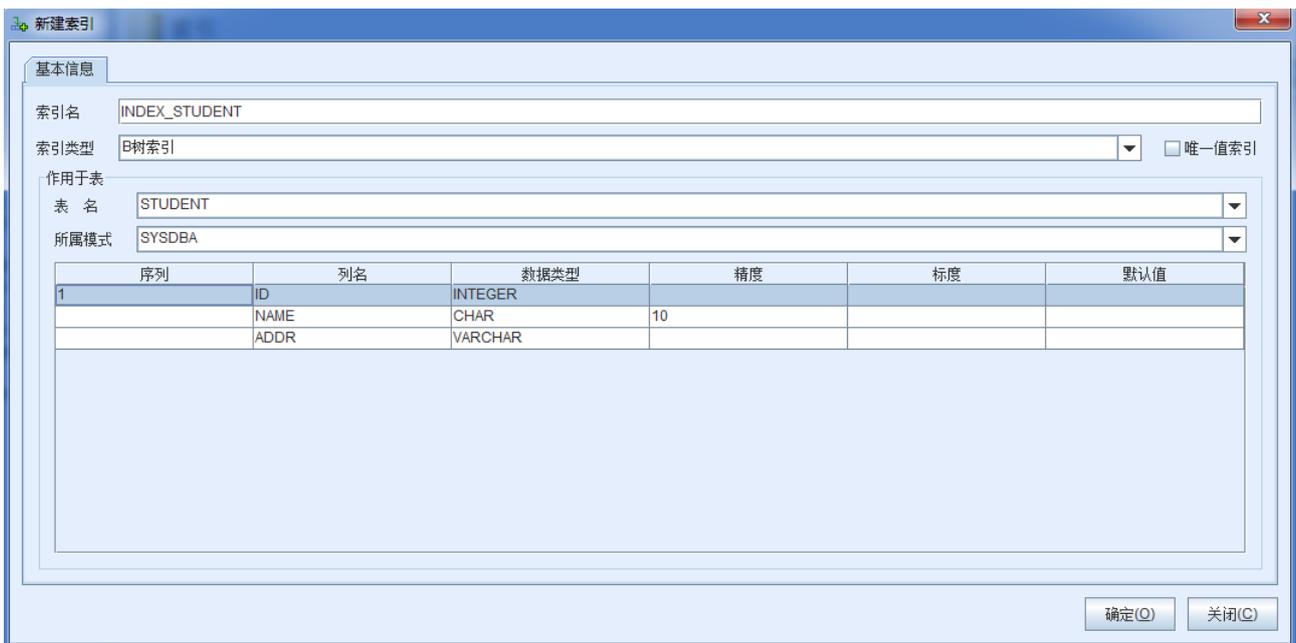
在数据库中包含了一个用于对表中记录按需排序，从而可以优化查询的特殊对象，它就是索引。索引和常见的书籍的目录用途相似，目录可以帮助读者快速地实现查找功能。同样，数据库中的索引也可以帮助用户在表中快速的查询满足条件的记录。

为了加快记录的定位速度，可以在表上创建必要的索引。一旦表上创建有索引，所有的 DML 语句都将试图利用索引提高其执行速度，但索引不能提高所有 DML 语句的执行速度，只有条件表达式中含有索引键，且表达式适合索引扫描才能发挥索引的作用。若该索引偶尔提高查询效率（使用索引键频率低），或根本不能提高效率，则索引的创建除了给系统平添麻烦而外，别无任何好处。

虚谷数据库提供了 B 树、全文索引等索引方式，其中 B 树索引用于在可比较大小的字段上创建索引，能享受索引好处的表达式形式应是基于关系操作符 '<' '>' '<=' '>=' '=' 的关系表达式，以及非通配符开始的 LIKE 表达式（其形式为 column LIKE 'xxx%'）。B 树索引

允许创建复合索引，即索引键字段数可以超过 1 个，复合索引发挥作用的条件为：索引键字段中的最前面的一个或连续多个字段的表达式应作为“与”式的因子出现在条件表达式中。例如：假定在表的字段 f1、f2、f3 上创建了复合索引，则语句...WHERE f1=? AND f2=? 可以享受索引的好处，而语句...WHERE f2=? AND f3=? 则不能享受索引带来的好处，原因是最前面的 f1 字段未出现在条件表达式中。

全文索引只针对文本类数据类型，如 CHAR、VARCHAR，以及大对象中的文本等，若没有全文索引，则这类数据只能使用 B 树索引。但在对文本进行检索时，通常会以文本中包含有哪些关键词作为检索的条件，传统关系表达式中只有形式为 column LIKE '%xxx%' 的关系表达式能表达查询的意图，并能得到预想的结果。但以该形式的表达式为条件表达式的查询只能使用顺序扫描，即全表扫描。在扫描过程中还要对相关文本字段关键词进行匹配，涉及的数据量和运算强度都非常大。对于巨型表而言，经常性地全文搜索查询会使系统不堪重负。针对以上应用情况，虚谷数据库采用全文索引解决查询性能问题。



### 说明

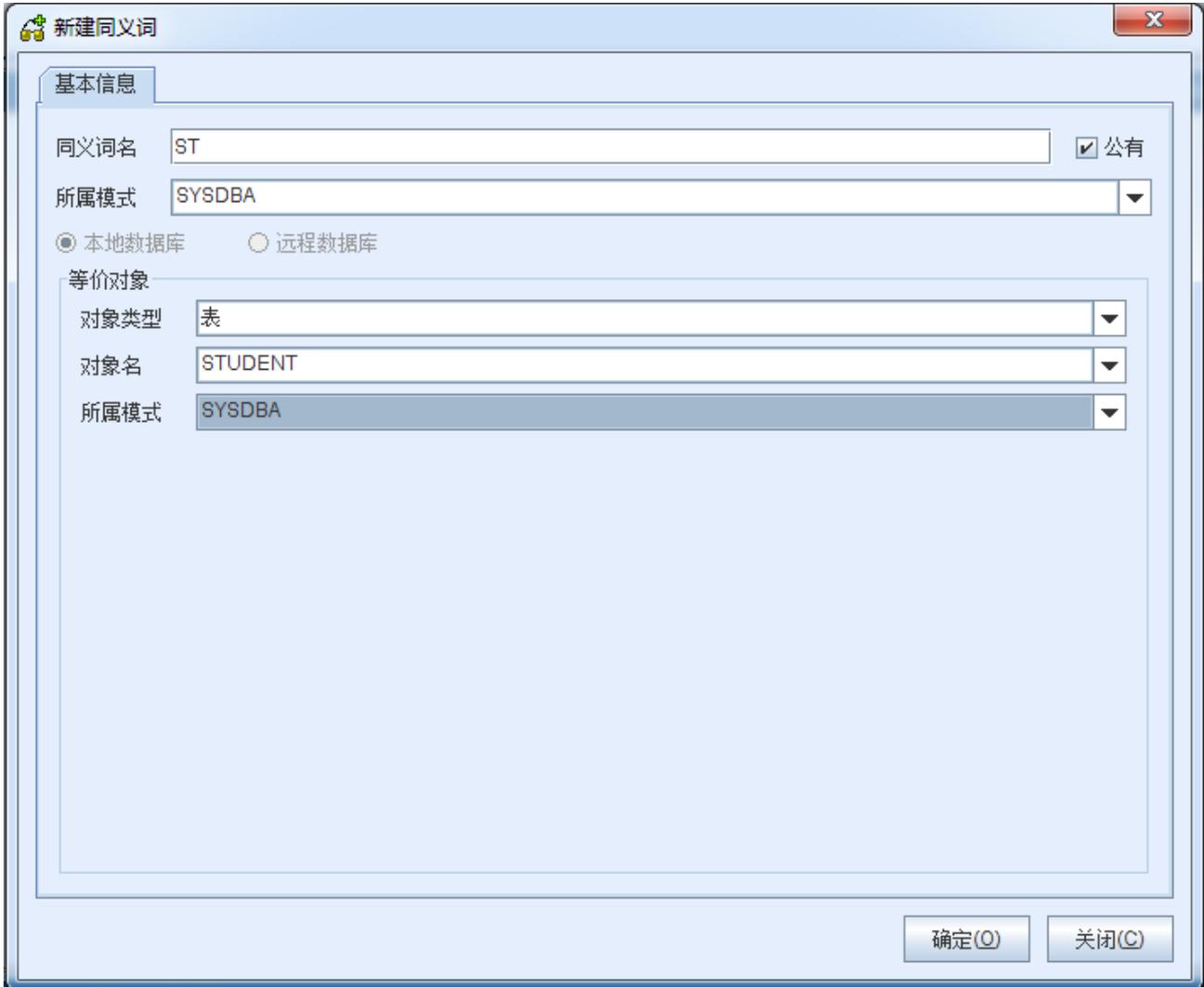
B 树索引和全文索引它们所对应的数据类型不相同。B 树索引用于在可比较大小的字段上创建索引，如 integer 等数值数据类型。B 树索引允许创建复合索引，即索引键字段数可以超过 1 个。全文索引只针对文本类数据类型如：CHAR、VARCHAR，以及大对象中的文本等。

## 2.8 同义词操作

在数据库中，一个对象名可能含有模式名以及连接名等部分，如：表名 mystudentable，也可写做 sysdba.mystudentable，sysdba 为表所在的模式名。由于此名较长，在它构成 SQL 语

句时很不方便，若将该名用简单名代替，则使用起来将方便很多。虚谷数据库同义词解决了上述问题，只要为一个复杂对象名创建了同义词，以后在使用该对象时就可以用同义词替代该对象名，从而简化命令，并使命令更加直观。

同义词在一个模式下的名称是绝对唯一的。



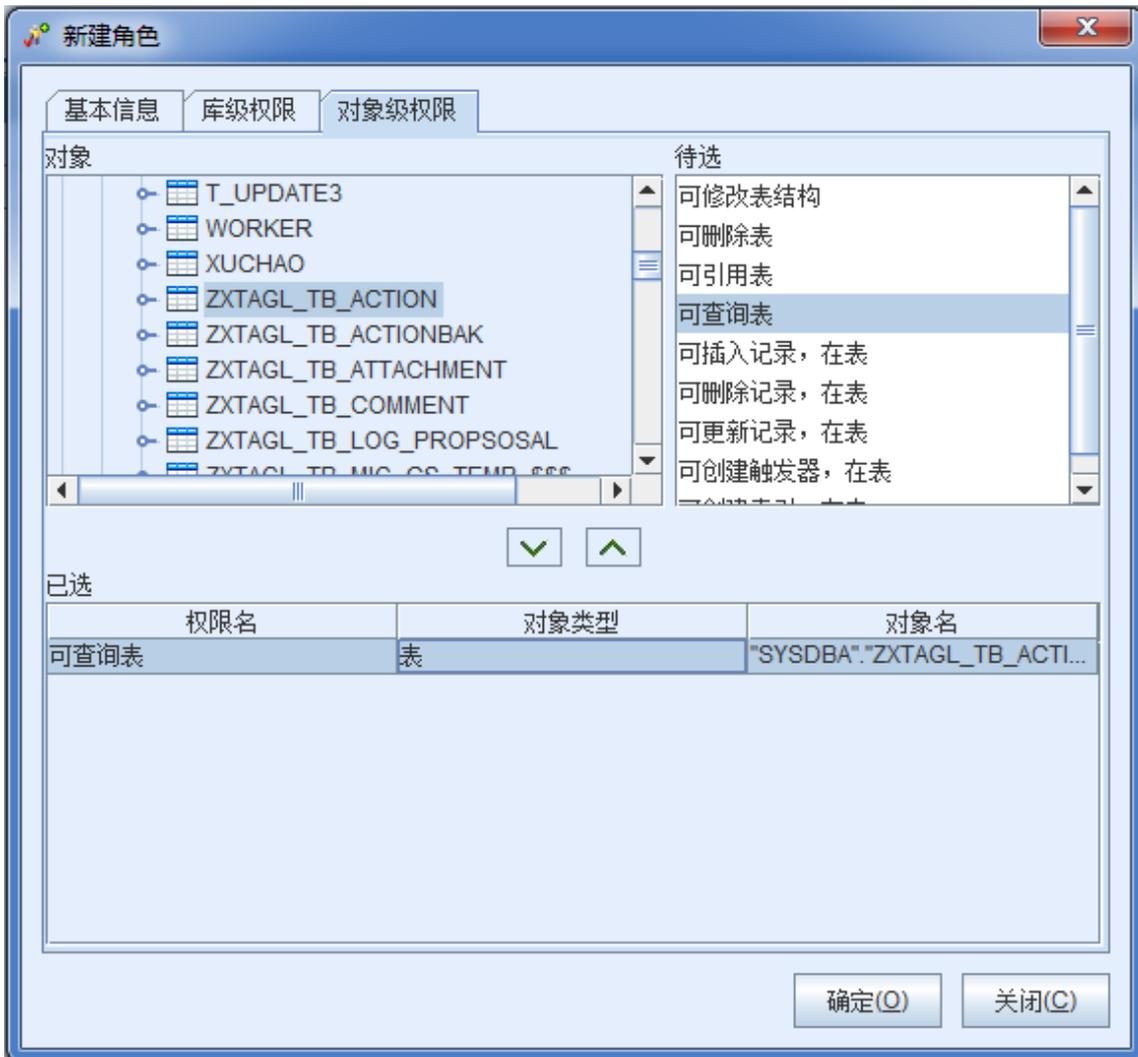
## 2.9 角色操作

虚谷数据库设立角色机制就是为了方便权限管理，在虚谷数据库中，可以创建多个角色，一个用户担任一个或多个角色，角色本身也可加入到别的角色中，从而继承所加入的角色组的权限，用户最终拥有的权限是他自己拥有的权限以及加入到的一个或多个角色具有的权限的合成权限。

比如在一个大型的人员管理系统中，要给 500 个人授予读写某张表的权限，给 400 个人授予只读某张表权限，给 300 个人授予只写某张表权限，假如是逐个授予权限，那么这将浪费很多时间，如果使用角色这个权限管理机制，只需要把这些权限授予给对应的用户就可以，不需要花

费大量的时间，同时批量修改用户的权限也是很方便。

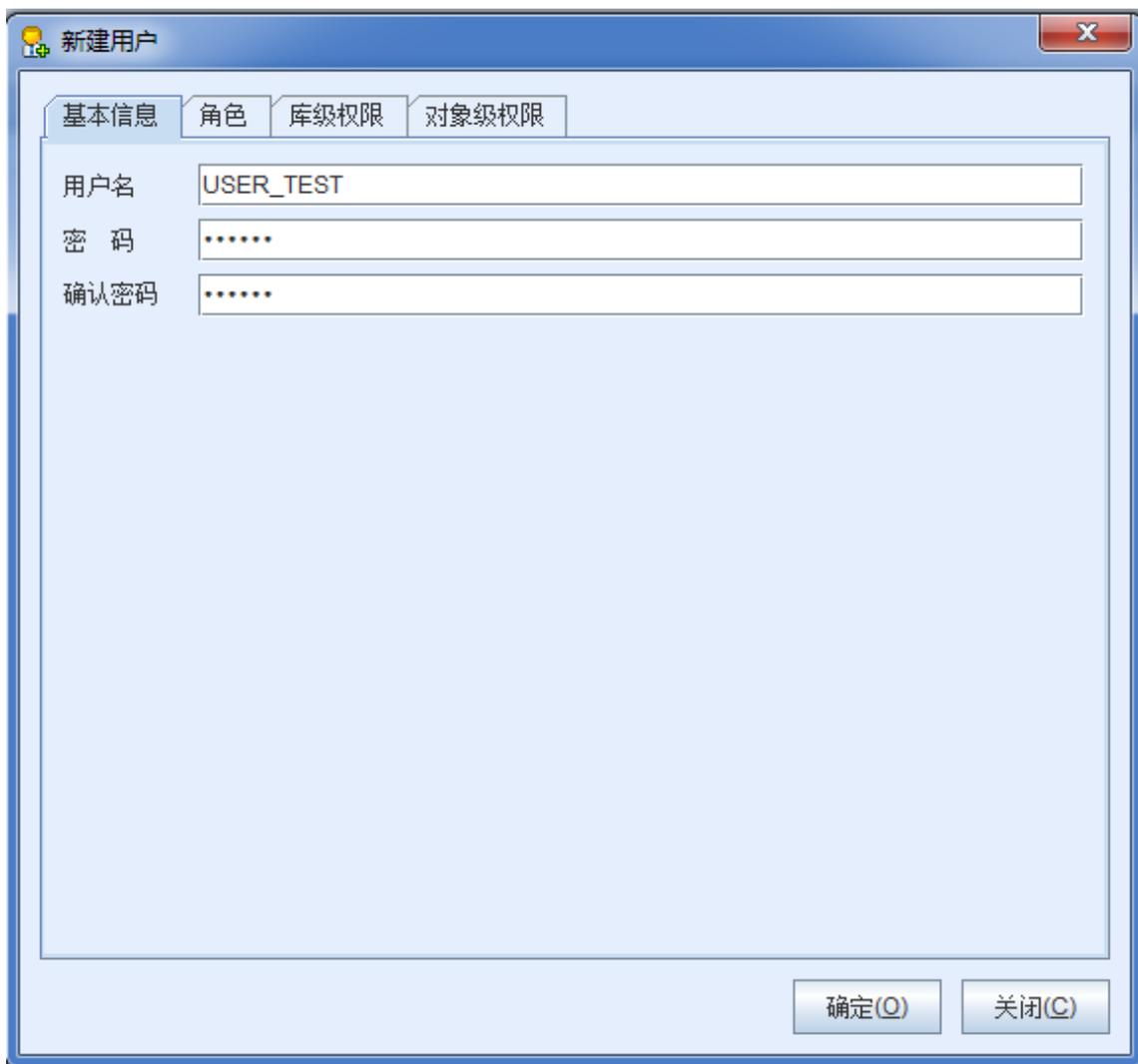
创建一个可以读某张表的角色。



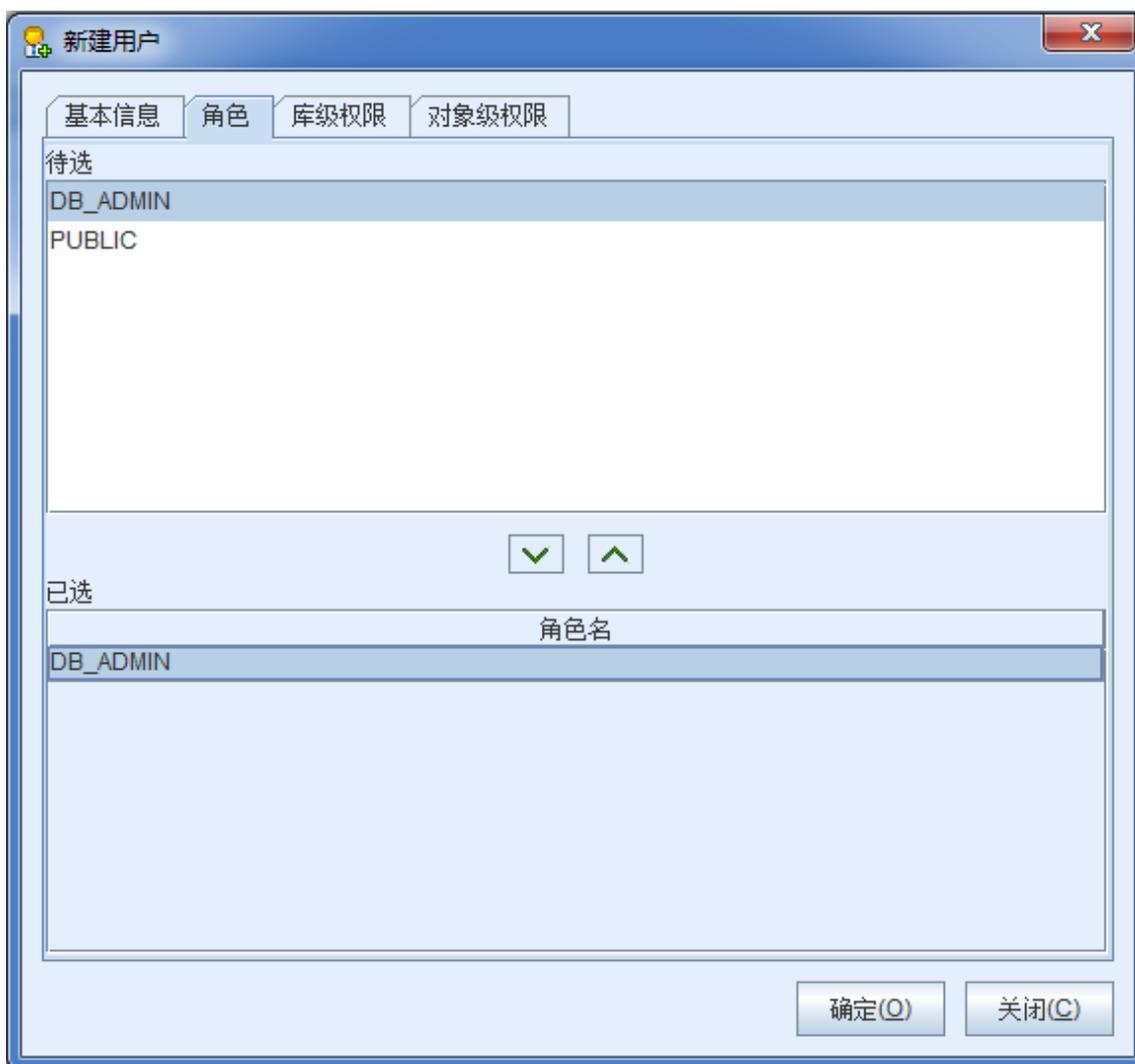
系统包含的角色有 DB\_ADMIN、PUBLIC。角色是同样可以嵌套在别的角色中，但这样的使用方法不值得推荐。库级权限和对象级权限包含了所有的权限，创建者在这里选择创建角色所需要的权限，直接添加即可。

## 2.10 用户操作

下面创建一个用户。用户名：**USER\_TEST**，密码：ab@123。

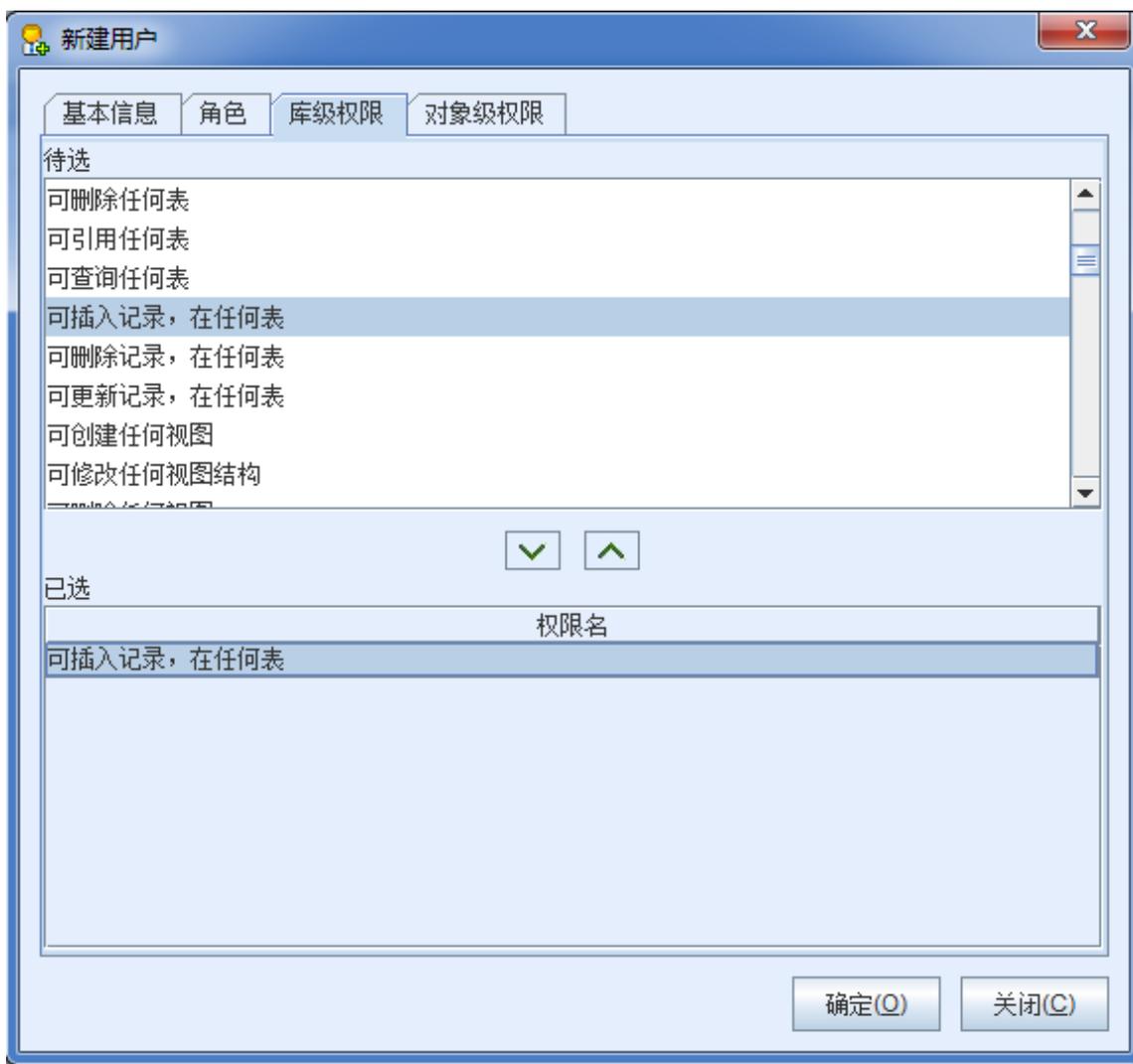


然后是用户角色的添加。目前系统角色有 DB\_ADMIN、PUBLIC，系统用户创建的角色也是可以添加。下面是添加一个名为 DB\_ADMIN 的角色。

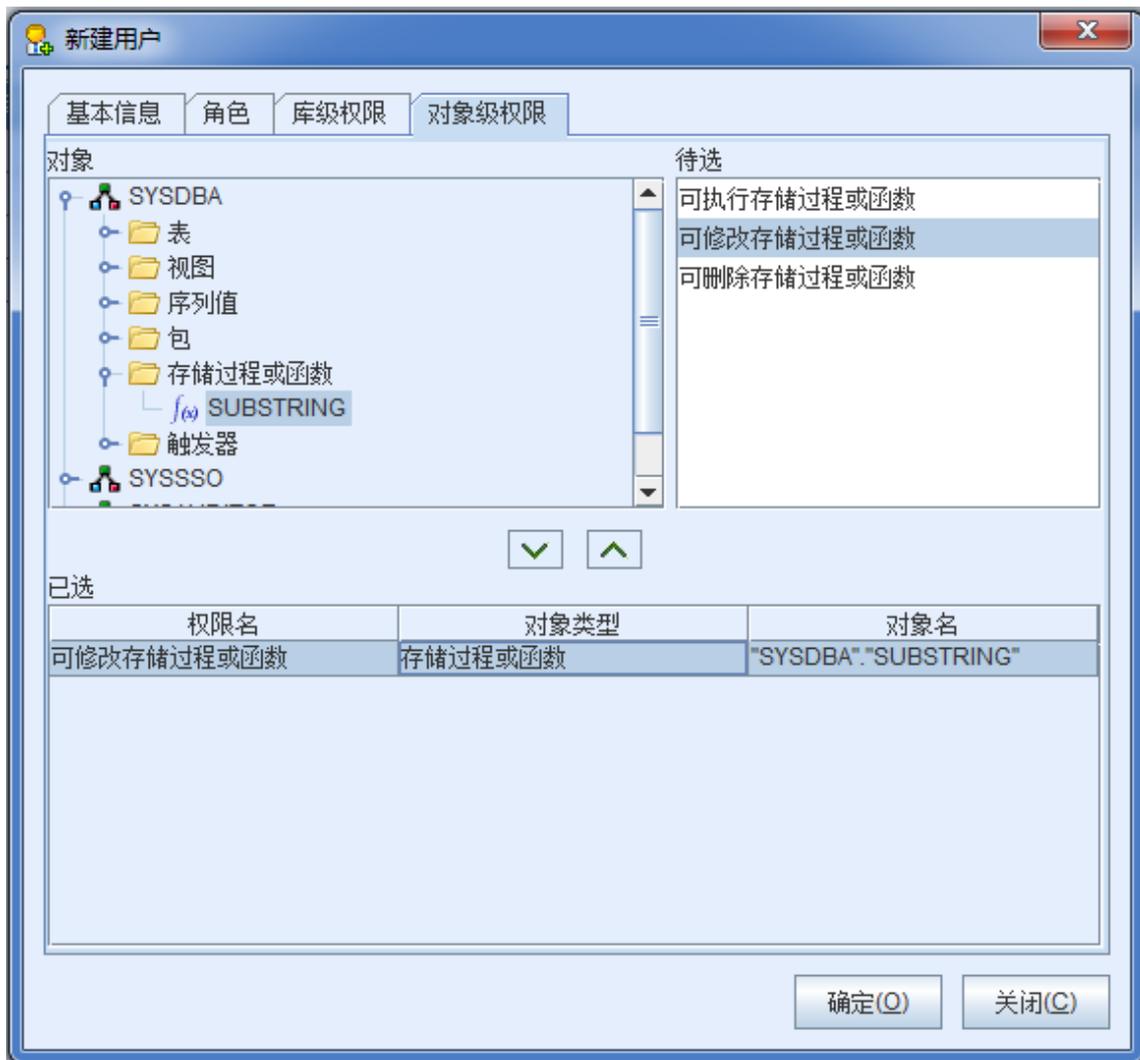


在 XuguManager 界面上，把权限大致的分为两部分：库级权限和对象级权限。

实际上库级权限就是模式级权限、对象级权限和列级权限的合集。这里便于管理，就将它们放在一起。



这里对象级权限所指的对象只是包括表、视图、序列值、包、存储过程和函数，不包含索引和触发器。其中触发器和索引的对象权限是指是否可以在表上创建索引或者触发器，并不是直接可以创建、修改、删除触发器或者索引的权限。



# 3 登录管理

登录管理这里是修改当前登录用户的密码。在多数数据库情况下建议用户一个实例对应一个数据库。



## 说明

此时修改的是当前登录者的密码。

密码修改时旧密码指当前登录时的密码，新密码由用户自己输入。虚谷数据库密码的规则在 xugu.ini 文件中的“pass\_mode”项设置。

# 4 用户自定义数据类型操作

面向对象编程思想在越来越多的语言中得以实现，如 JAVA、.NET，在这些语言中都允许自定义类，虚谷数据库也可以使用对象了。

虚谷数据库提供了内置的数据类型供用户选择使用：如 INT，VARCHAR，DATE；用户还可以自己定义需要的数据类型，即：自定义数据类型，也叫虚谷数据库对象。像其他面向对象语言一样，虚谷数据库对象也包括属性和方法。

使用语法：

```
CREATE TYPE type_name AS OBJECT (  
a INT,  
b VARCHAR,  
MEMBER FUNCTION c RETURN DATE);
```

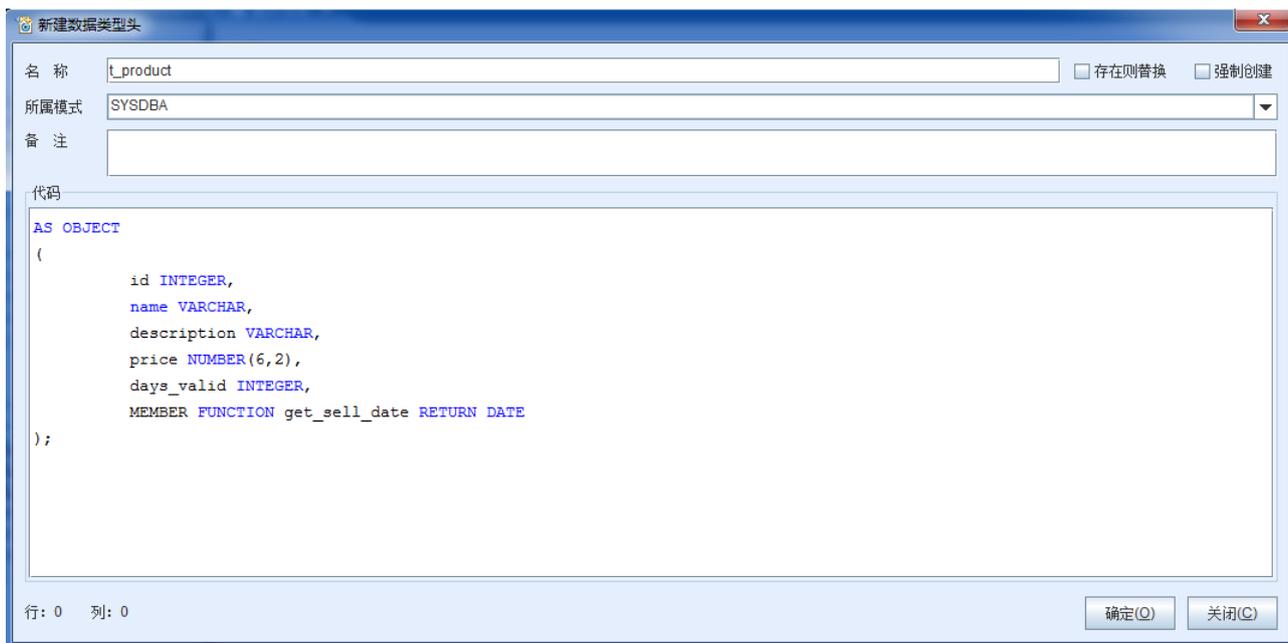
例：定义这样一个对象，T\_ADDRESS 表示地址，包括四个属性：

```
CREATE TYPE t_address AS OBJECT (  
street VARCHAR,  
city VARCHAR,  
state VARCHAR,  
country VARCHAR);
```

例：定义这样一个对象，T\_PRODUCT 不仅包括属性，还包括一个方法：

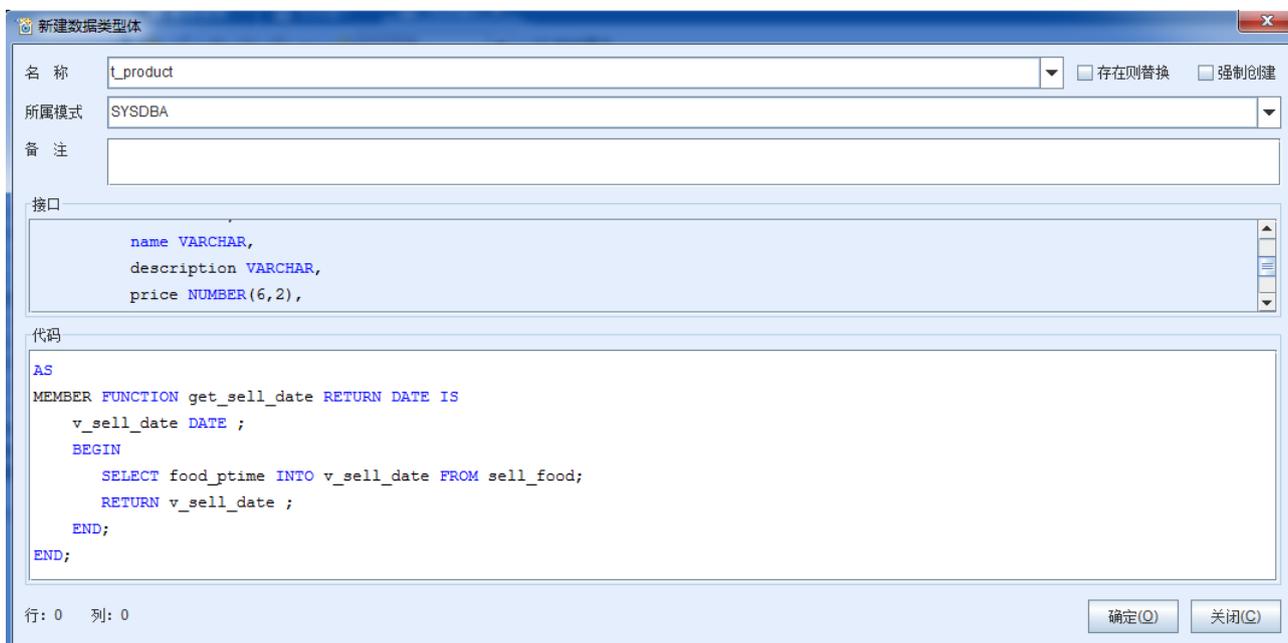
```
CREATE TYPE t_product AS OBJECT (  
id INTEGER,  
name VARCHAR,  
description VARCHAR,  
price NUMBER(6,2),  
days_valid INTEGER,  
MEMBER FUNCTION get_sell_date RETURN DATE);
```

如下图所示：



### 创建类型体

```
CREATE TYPE BODY t_product AS  
MEMBER FUNCTION get_sell_date RETURN DATE IS  
v_sell_date DATE ;  
BEGIN  
SELECT food_ptime INTO v_sell_date FROM sell_food;  
RETURN v_sell_date ;  
END;  
END;
```



构造函数：像其他对象语言一样，虚谷数据库对象也有构造函数，用来初始化自身的属性默认值。每个类对象都有自己的构造函数，相应的构造函数会在创建类对象时自动创建，也可以自己定义虚谷数据库对象的构造函数，按照使用者的想法对对象属性赋初始值。

例：

```
CREATE TYPE t_person AS OBJECT(  
p_id INTEGER,  
p_name VARCHAR2(20),  
phone VARCHAR2(12),  
CONSTRUCTOR FUNCTION t_person(p_parameter int) RETURN SELF AS  
RESULT);  
  
CREATE TYPE BODY t_person AS  
CONSTRUCTOR function t_person(p_parameter int) RETURN SELF AS  
RESULT  
IS  
BEGIN  
    (处理部分);  
END;  
END;
```

根据构造函数 `p_parameter` 个数和类型的不同，可以定义多个自定义构造函数。虽然构造函数的名字相同，但因参数的个数或类型不同，所以并不发生冲突。这是面向对象编程中“多态”的一种体现。

另外，以上创建的对象类型属性的数据类型都是虚谷数据库内部提供的，也可以将自定义的对象类型作为一种数据类型在创建对象类型时使用。如：

```
CREATE TYPE t_person1 AS OBJECT(  
name VARCHAR,  
phone VARCHAR,  
address t_address);
```

此处用的 `t_address` 就是在上面创建的自定义对象类型 `t_address`。

对象类型在数据库中的使用：

可以像使用虚谷数据库内置的数据类型一样使用自定义数据库对象类型，在创建表时，可以将其作为列的一个定义，如：使用刚才创建的对象 `t_product` 创建表 `products`。

```
CREATE TABLE products (  
product t_product,  
quantity INTEGER);
```

向表中插入数据：

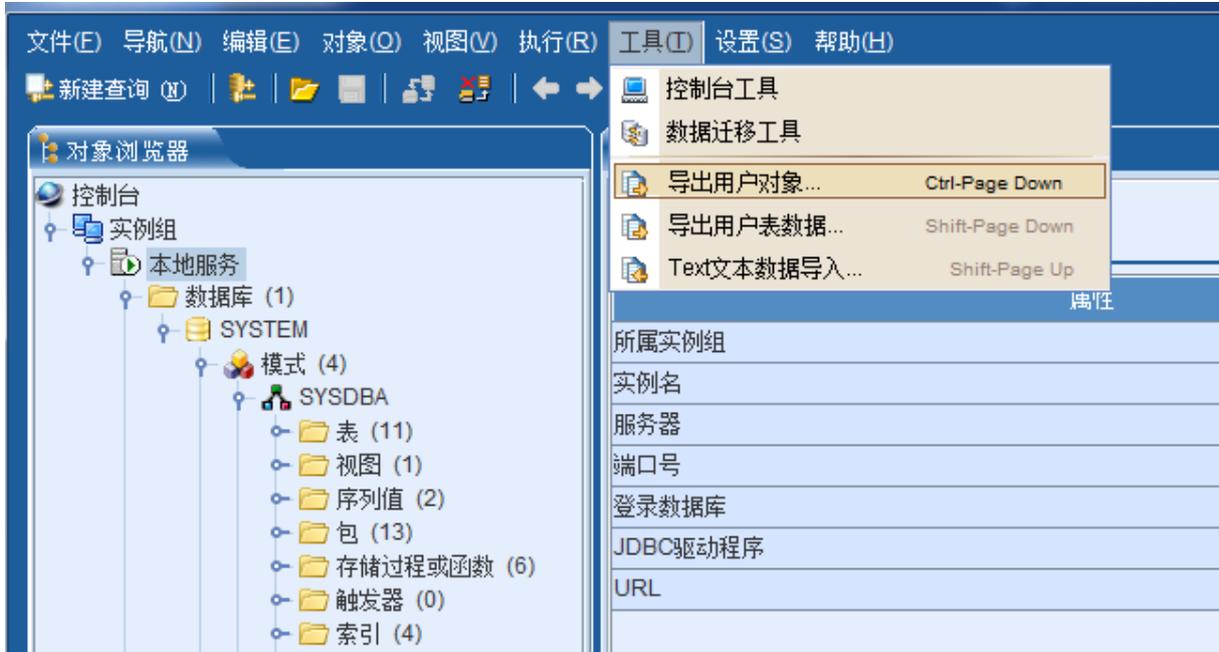
```
INSERT INTO products VALUES (t_product (1, 'name of goods', '  
description of goods', 3.2, 10), 20);
```

```
DECLARE  
a t_product;  
BEGIN  
a:=t_product(2, 'name of goods', 'description of goods', 3.0, 10);  
INSERT INTO products VALUES (a, 20);  
END;
```

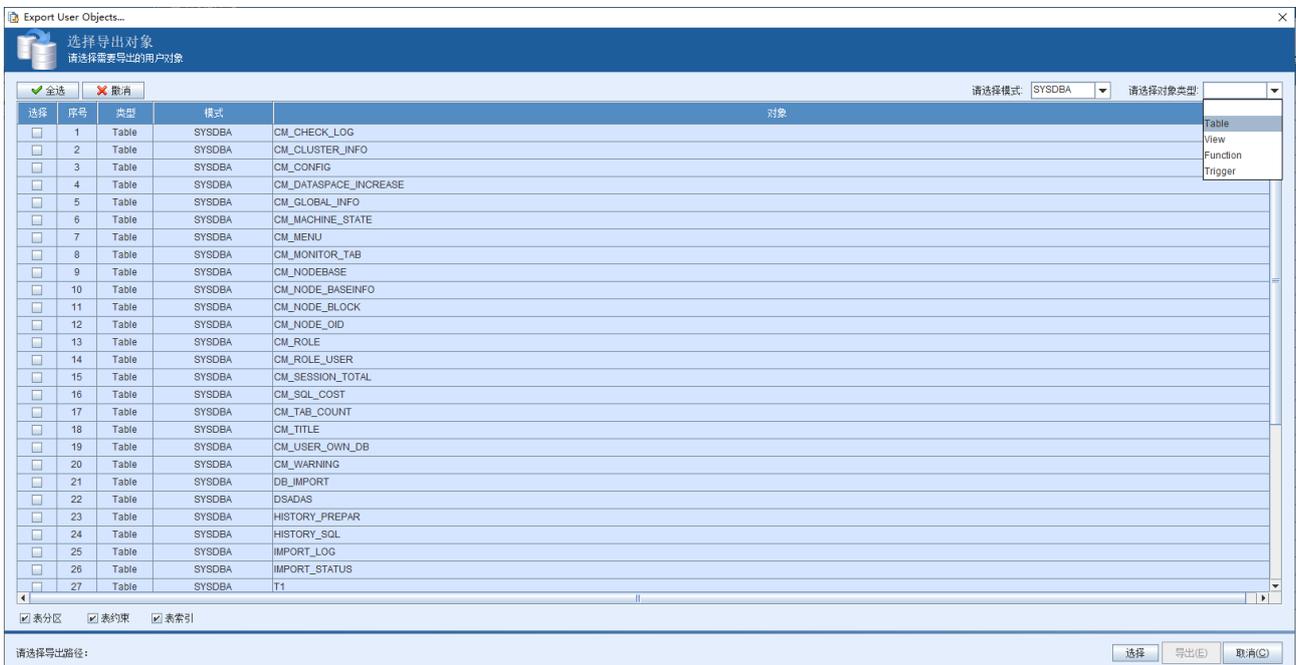
```
SELECT * FROM products p WHERE p.product.id=1 ;
```

### 4.0.1 数据库对象导出

虚谷数据库管理软件提供导出数据库对象的功能。



导出对象可以涵盖表、视图、序列值、存储过程、存储函数和包等数据库对象：



数据库对象导出结果，可以在文本中查看：

```
-- Create Table --
create table SYSDBA.C(
I integer);
comment on table SYSDBA.C is 'test table';

-- Create Table --
create table SYSDBA.STUDENT(
ID integer,
NAME varchar,
ADDR varchar);
comment on table SYSDBA.STUDENT is '学生表 ';

-- Create Table --
create table SYSDBA.TEST(
ID integer,
NAME varchar);
comment on table SYSDBA.TEST is ' test table';
```

虚谷数据库管理软件还提供表数据的导出和文本数据的导入导出。

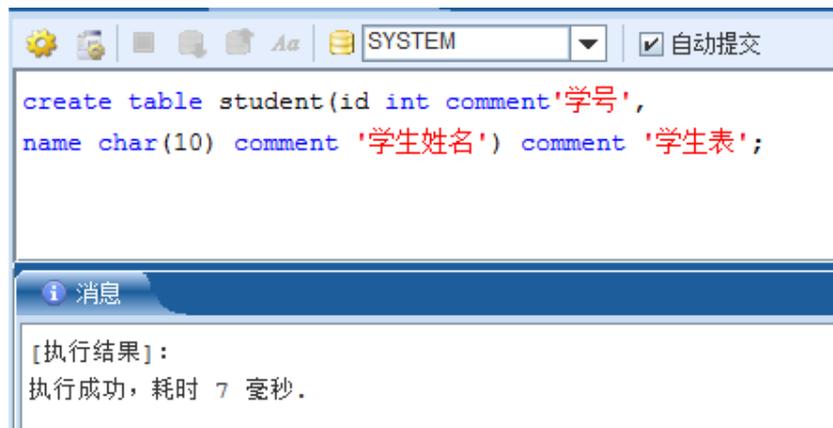
## 4.1 表注释添加

### 4.1.1 创建表时添加表注释和列注释

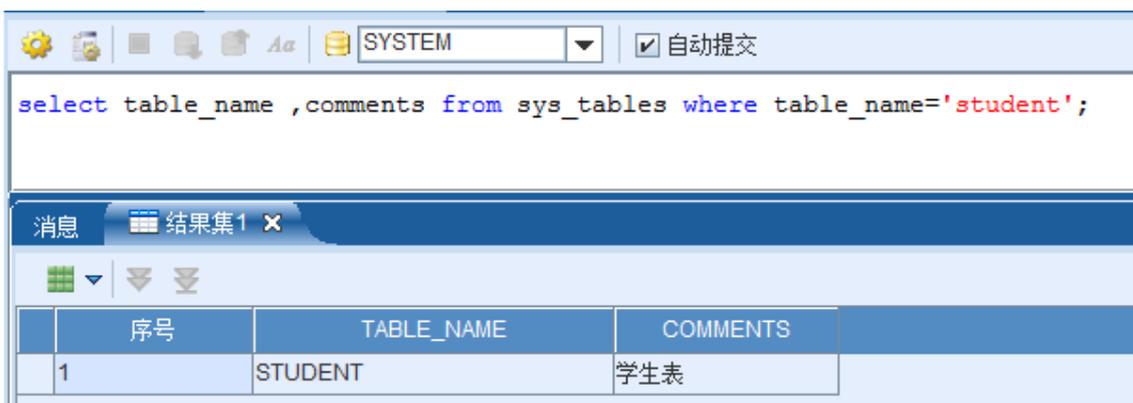
语法格式：

```
Create table table_name (Column_name Datatype [COMMENT ' 列注释内容' ][,])COMMENT ' 表注释内容' ;`
```

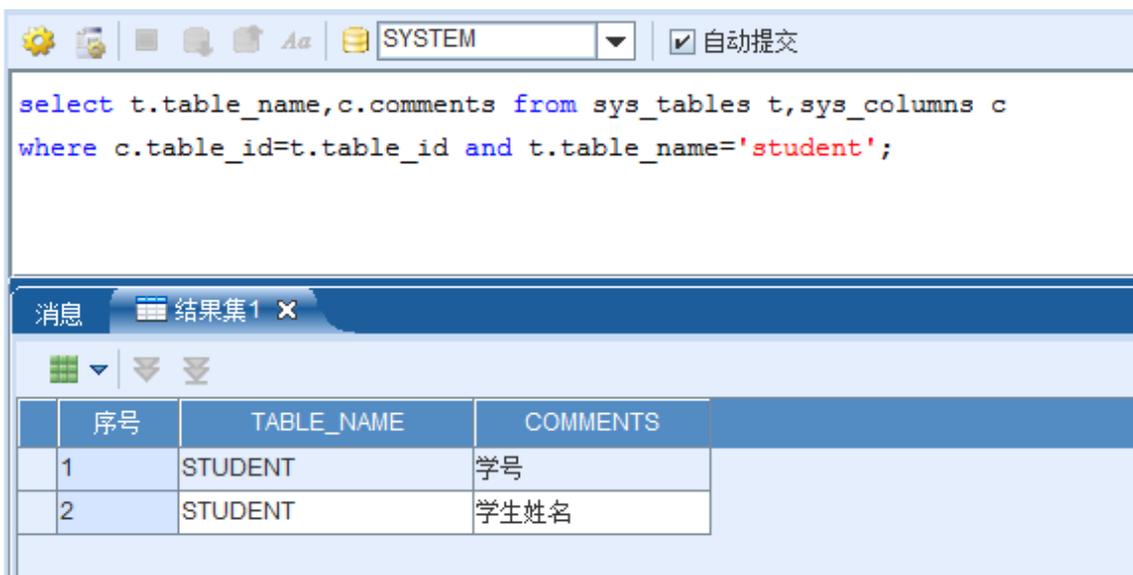
- 创建表，在创建表的同时加上对表和列的注释。



- 查询表注释。



- 查询列注释。



### 📖 说明

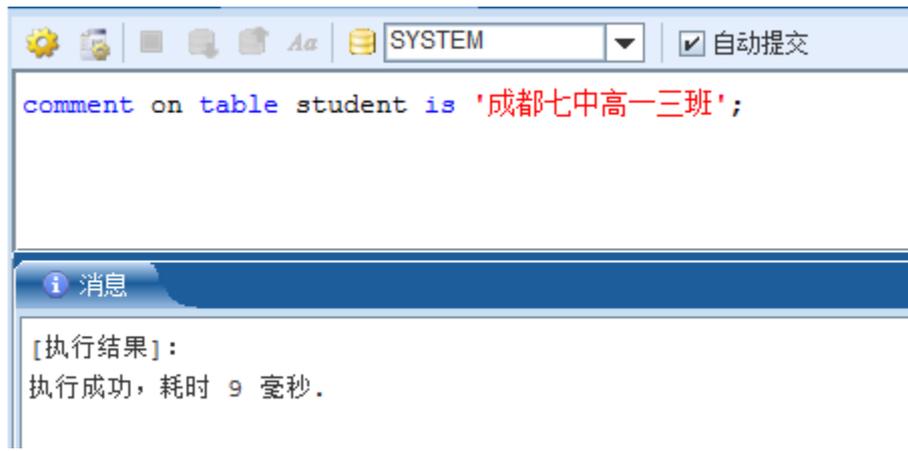
上面所添加的表注释和列注释可以单独添加。

## 4.1.2 创建表完成后添加表注释或列注释

### 添加注释

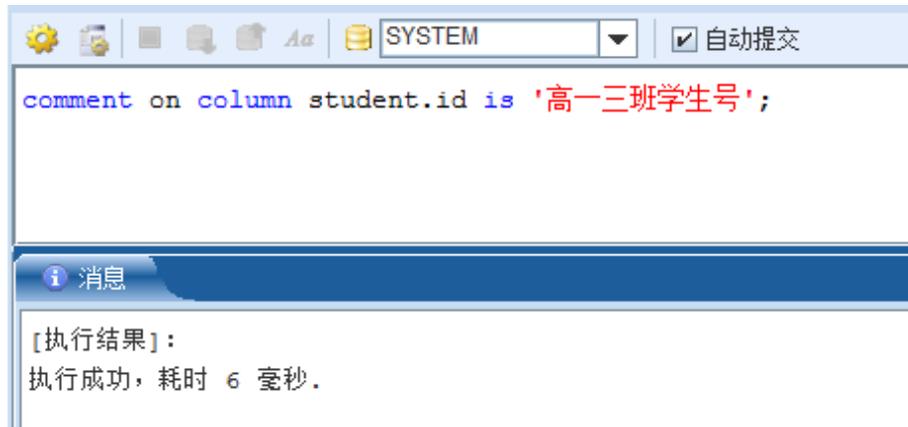
- 添加表注释。

```
Comment on table table_name is ' 注释内容' ;`
```



- 添加列注释。

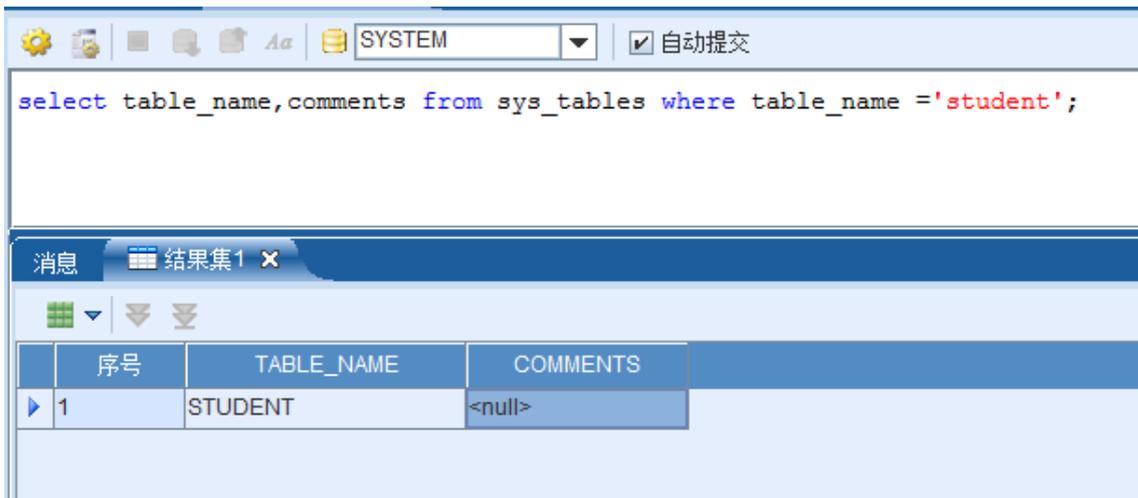
```
Comment on column Table_name.Column_name is ' 注释内容' ;
```



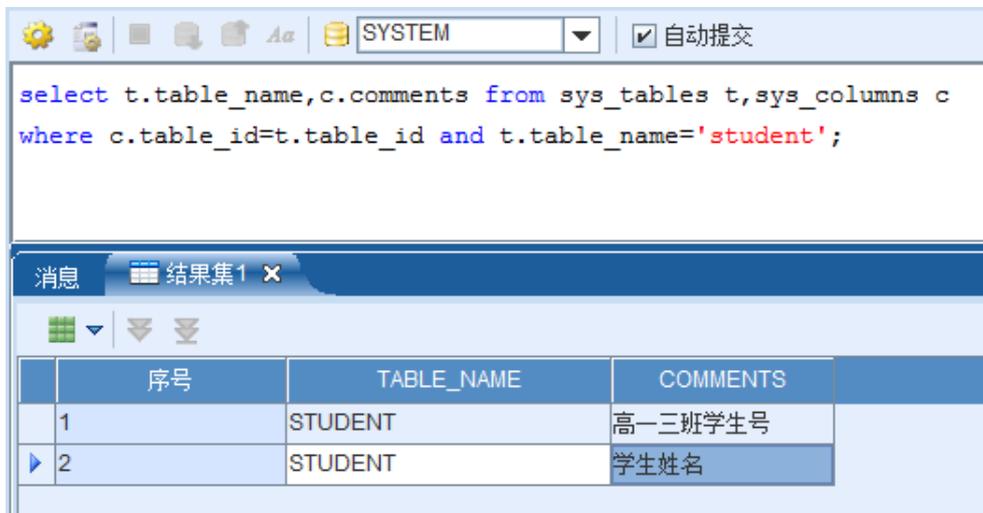
### 查看注释

- 查看表注释。





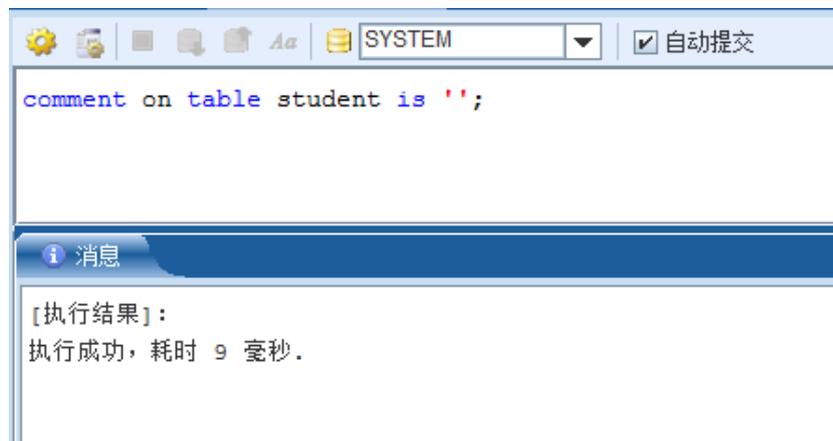
- 查看列注释。



## 删除注释

- 删除表注释。

```
Comment on table table_name is '';
```



- 删除列注释。

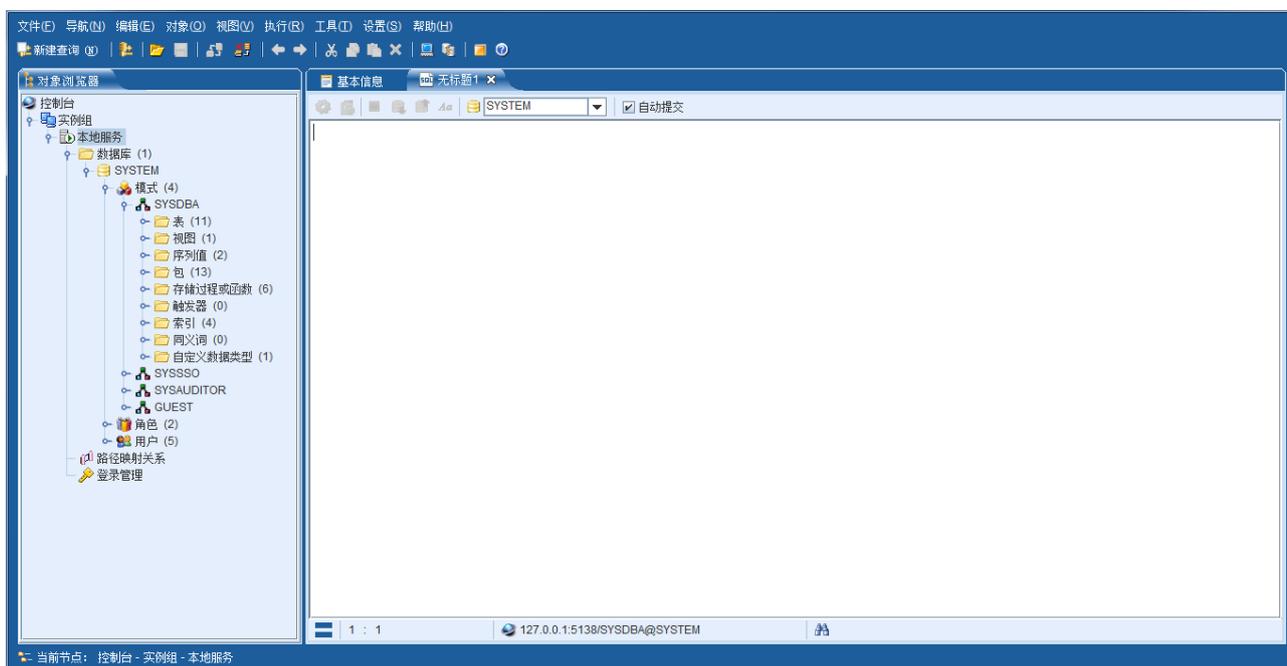
```
Comment on column Table_name.Column_name is '';
```

如 ‘comment on column student.id is ” ;’，删除 student 表 ID 列的注释。

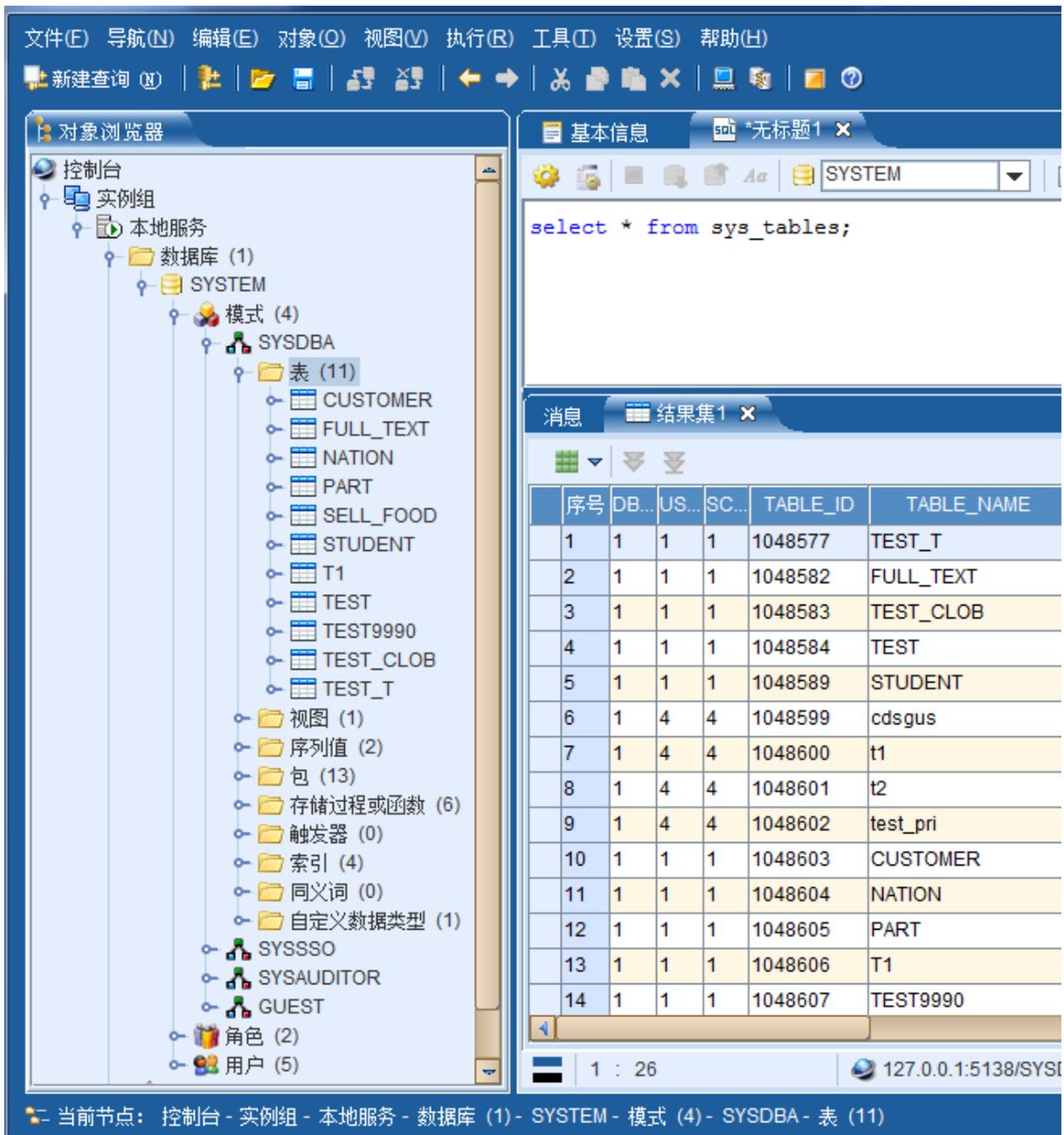
# 5 SQL 编辑窗口

XGSQL 查询器是虚谷数据库管理系统的 SQL 语句交互工具，支持用户以交互的方式完成对虚谷数据库的操作。下面以 Windows 平台为例，介绍 SQL 查询器的使用方法。

首先，确认虚谷数据库服务端已启动，客户机上 Windows 工作正常且所需环境和工具已配置好。这时候可以在 Windows 的“开始”菜单中选取“程序”下的“虚谷数据库管理软件 XuguManager”，启动 XuguManager 后，确保建立连接到指定数据库之后点击“文件-新建查询”，弹出用户界面如下：



默认创建一个会话连接。登录新建查询之后，用户便可以在 XGSQL 中通过 SQL 语句对数据库进行操作。



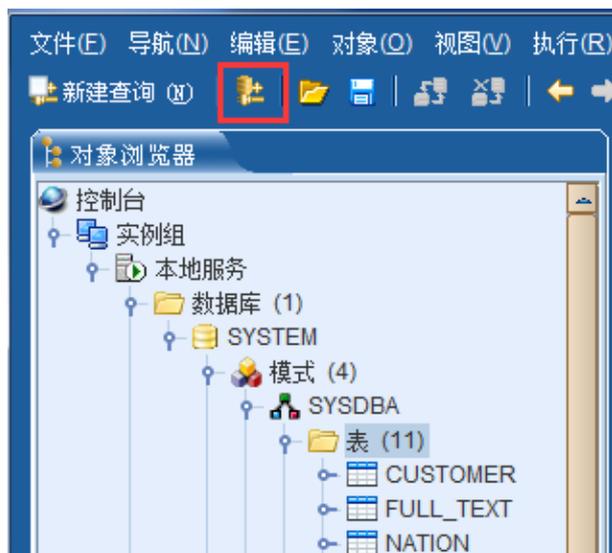
XGSQL 的功能主要通过菜单项体现，常用的菜单项在按钮图标区有相应按钮，当鼠标指向按钮时，会有该按钮的功能、热键提示。具体按钮功能请参考对应菜单项介绍。

### 新建查询

“新建查询”是在当前登录 XuguManager 连接的基础上，建立一个新的查询窗口。

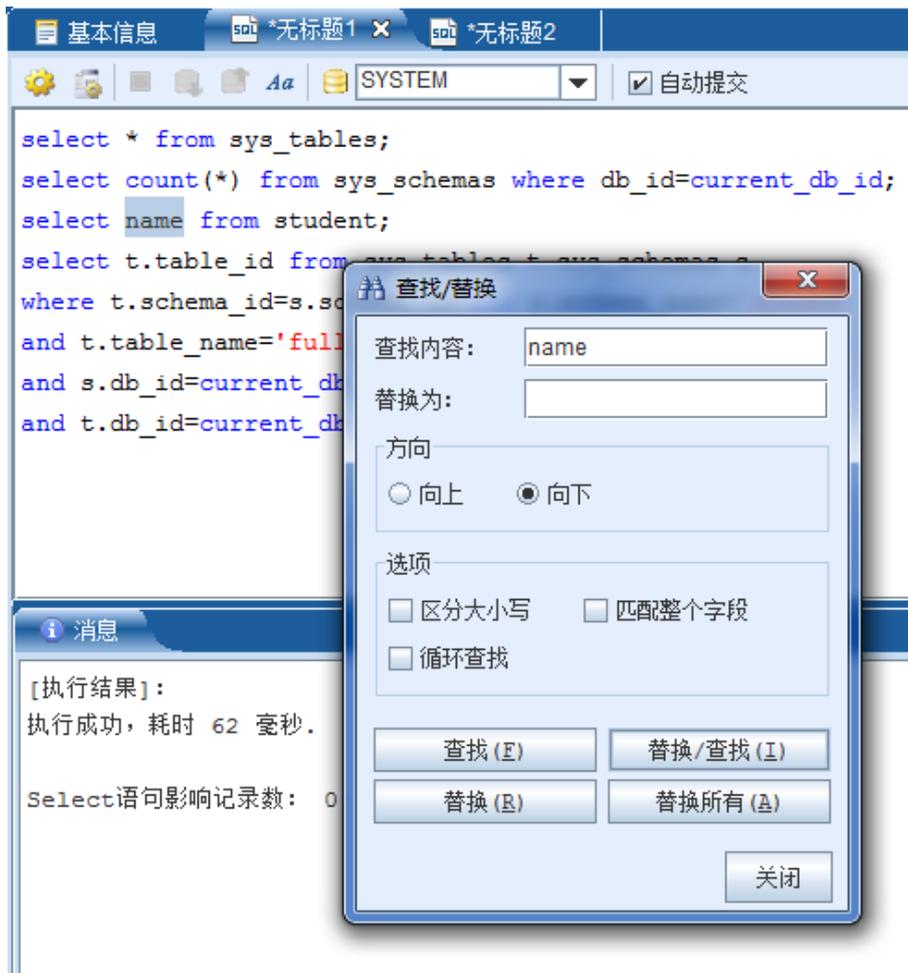
### 新建数据库引擎查询

新建数据库引擎查询就是换一个 session 进行连接查询，用户可更换用户名或者更换数据库进行重新连接。

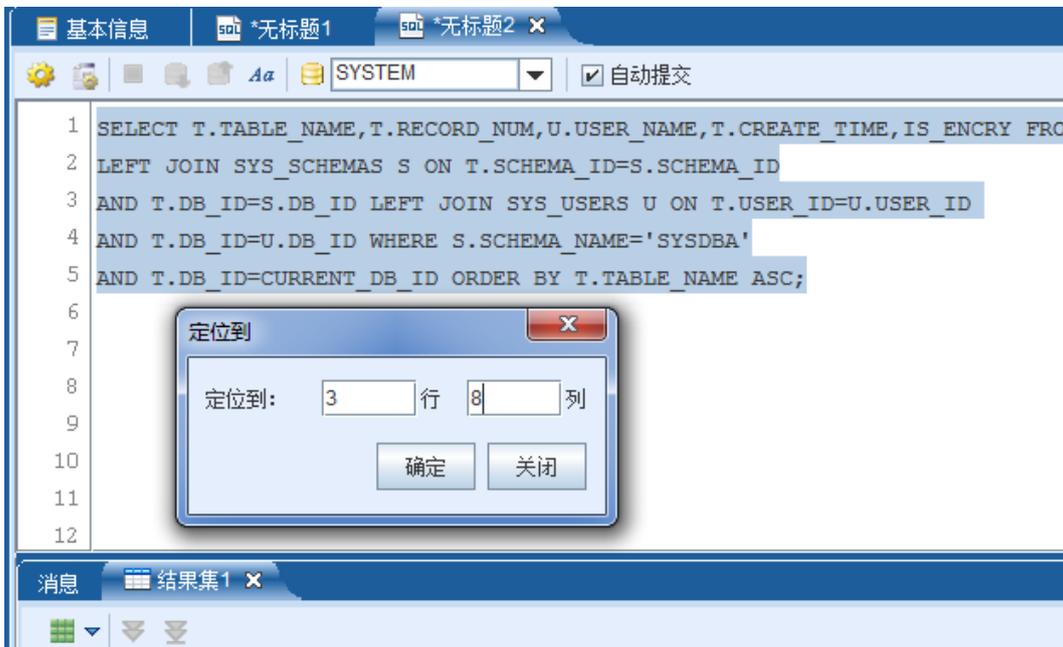


## 查找和定位

查找替换。



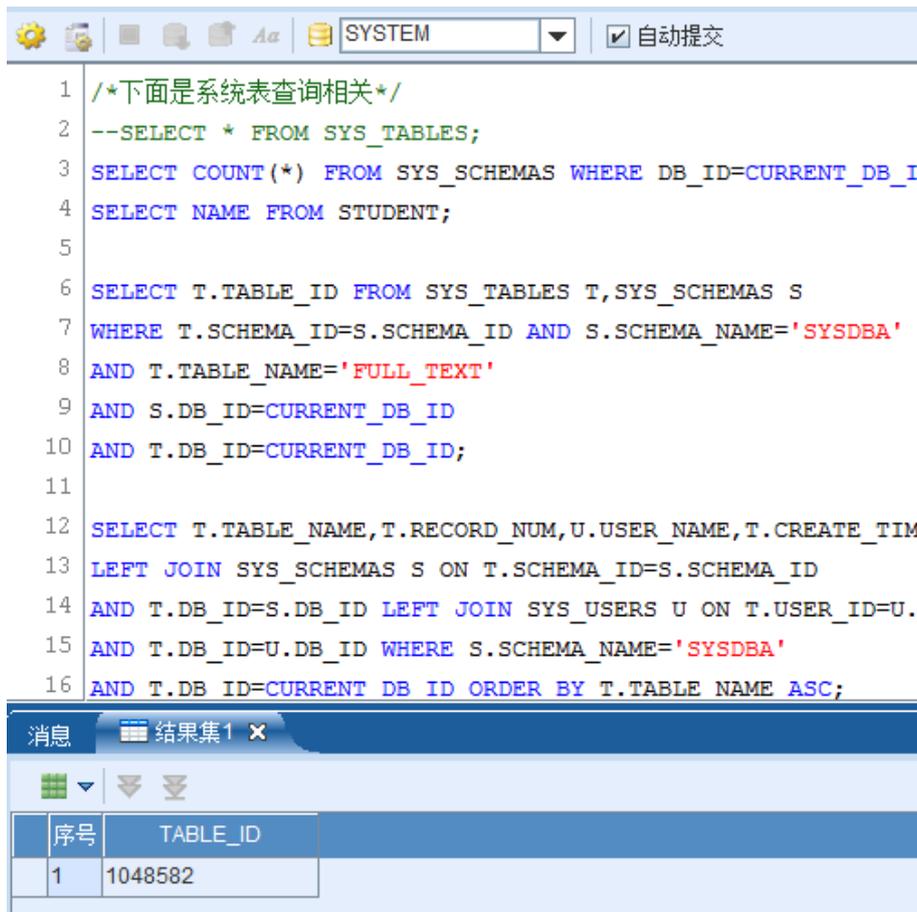
“定位到...”菜单项。在编辑区中让光标定位到指定语句的行上。



虚谷数据库注释。虚谷数据库单行注释使用“-”，多行注释使用“/SELECT NAME...../”。

### 说明

区别于数据库注释。

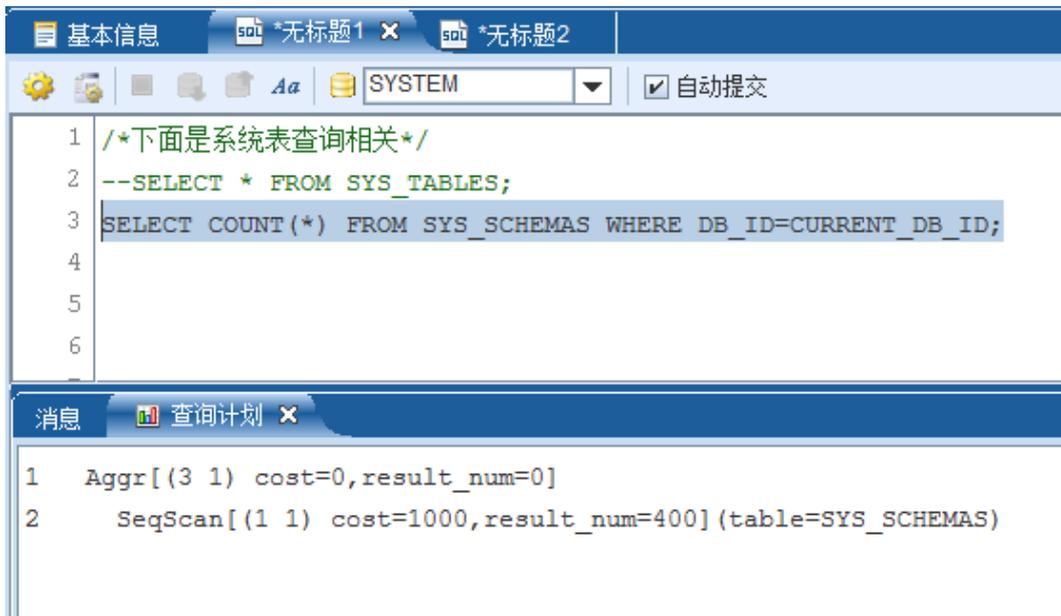


### 执行

“执行”菜单项。执行编辑区域内的所有 SQL 语句。

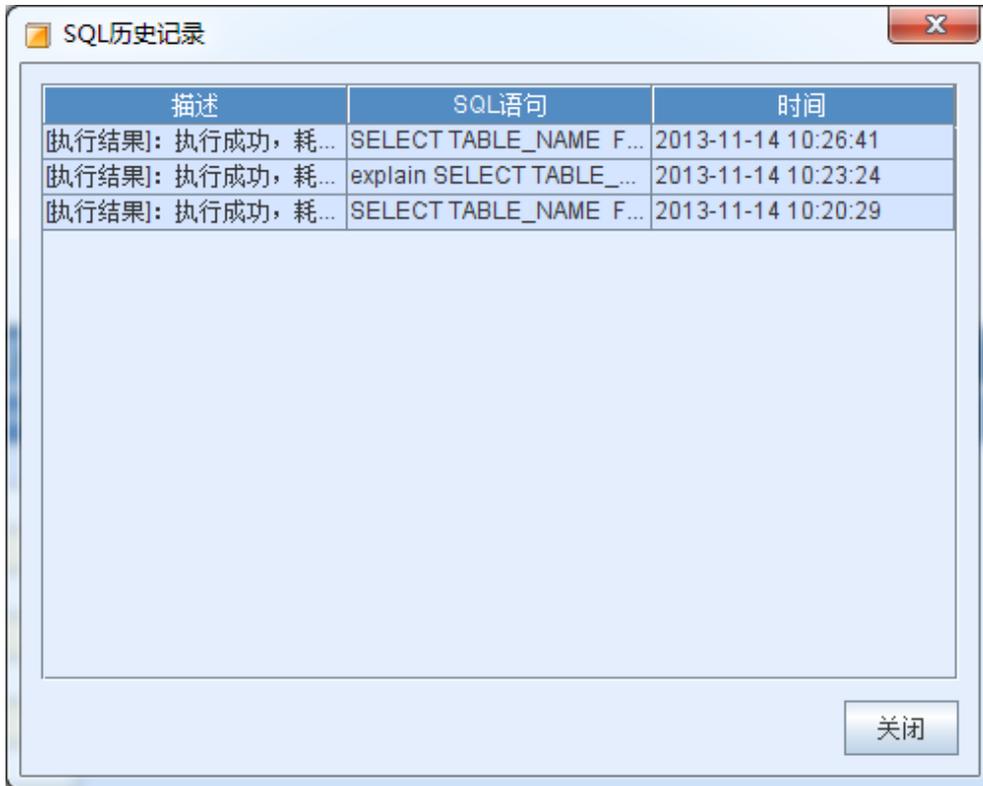
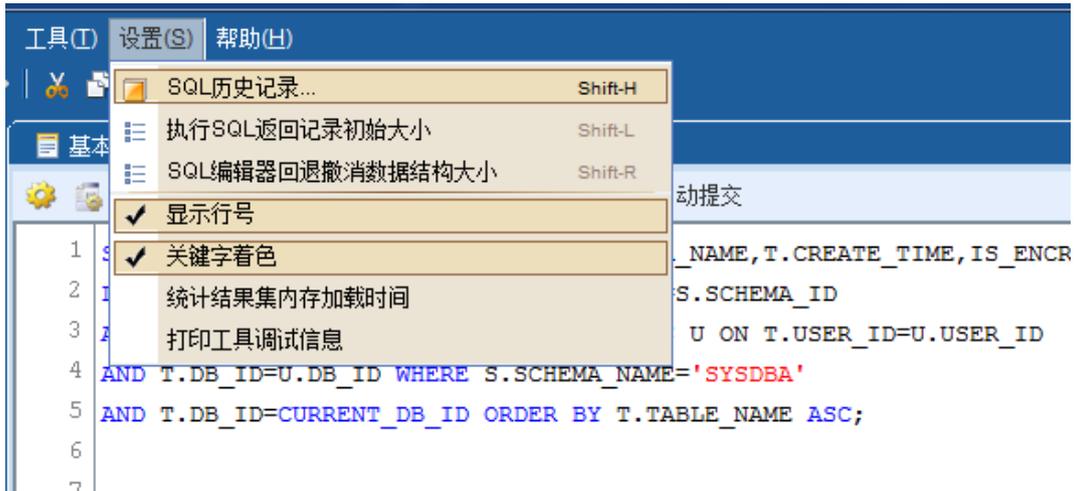
选中需要执行的 SQL 语句后，选择“执行”，则执行用户所选中的语句，否则执行编辑区域内所有的 SQL 语句。

“显示查询计划”菜单项。显示当前编辑区域内的 SQL 语句的查询计划。



## 设置

“设置”菜单项在界面上只是对已经执行过的 SQL 语句做记载。



在编辑区域内执行后的语句是可以从” SQL 历史记录” 里面查看。

## 消息窗口



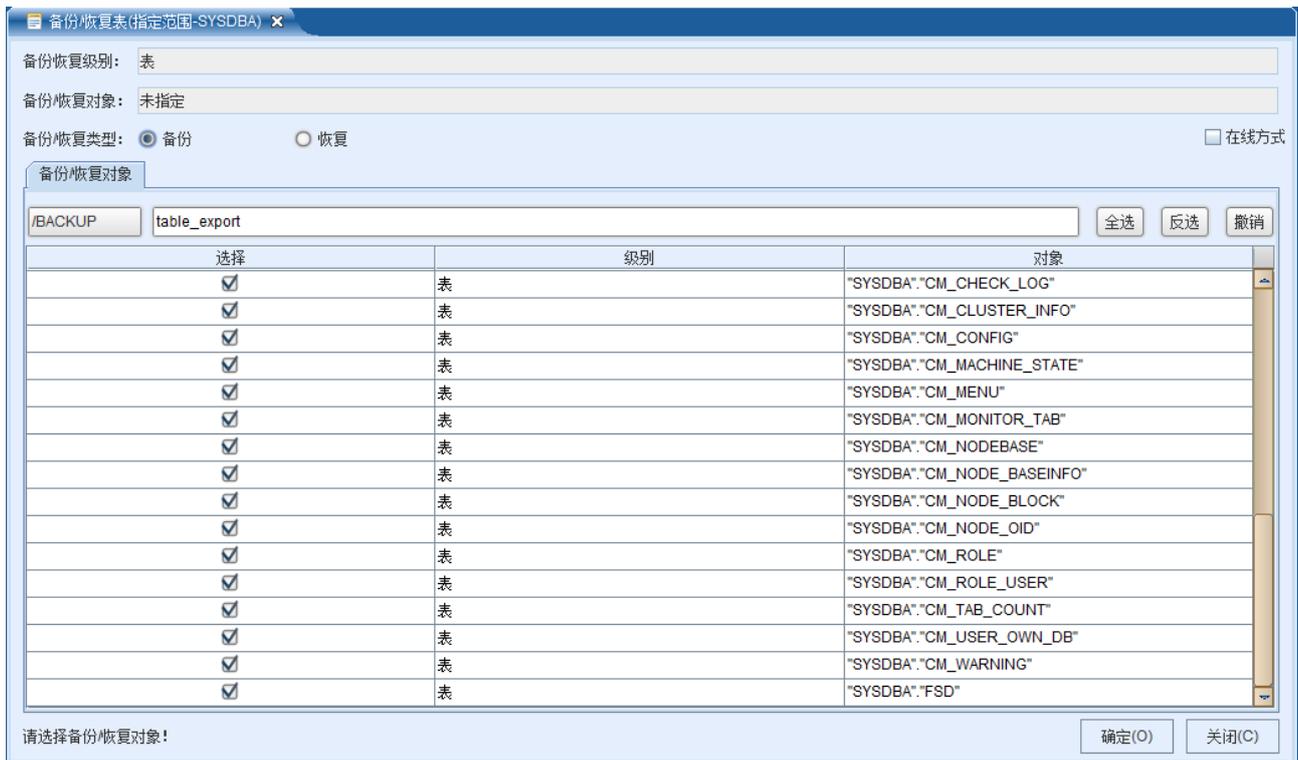
消息返回窗口用于显示结果集。用户可以通过单击“查看数据...”查看结果集，或者通过单击“导出结果集”把想要的结果集保存在一个文件中，用户再次查看结果集时，打开保存的文件即可。

# 6 数据库备份恢复

随着计算机的普及和信息技术的进步，特别是计算机网络的飞速发展，信息安全的重要性日趋明显。但是作为信息安全的一个重要部分——数据备份却往往被人们所忽视。只要发生数据传输、数据存储和数据交换就有可能产生数据丢失，这时如果没有进行数据备份，就容易导致数据的丢失，造成的损失是无法弥补与估量的。

虚谷数据库除了多副本保障数据安全外，还提供数据库备份和恢复功能。目前仅在安全版提供数据备份恢复功能。





备份默认路径是在安装目录“/XHOME/BACKUP”，用户可以自行设置需要备份的路径，路径映射需要在 datafile.ini 和 mount.ini 两个文件中进行配置，方法详见《虚谷数据库安装指南》的“数据库文件存储路径配置”章节。

### 备份和恢复注意事项

集群模式下只能进行表备份，不能进行全库备份，只有在单节点下能进行全库备份，多节点下虚谷数据库有多副本数据安全机制。表备份只能把表数据备份到工具连接所在节点，恢复也只能在该节点上进行数据恢复。在系统库下能进行全库备份，包含系统库和各个用户库的数据，也可以对系统库下的表进行备份恢复。在用户库下只能对当前库和当前库下的表进行备份恢复。



成都虚谷伟业科技有限公司

联系电话：400-8886236

官方网站：[www.xugudb.com](http://www.xugudb.com)