

虚谷数据库

控制台工具 V1.4.6

用户指南

文档版本 01

发布日期 2024-01-10



版权所有 © 2025 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

前言

概述



本文档对控制台工具的功能和安装使用进行了详细的介绍。

读者对象

- 数据库管理员
- 数据库用户

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
 注意	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。
 说明	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	2025-01-10	第一次正式发布。

目录

1	工具概述	1
2	安装部署	2
2.1	部署环境要求	2
2.1.1	Windows	2
2.1.2	Linux	2
2.2	启动控制台工具	2
2.2.1	Windows	2
2.2.2	Linux	3
2.3	退出控制台工具	4
2.3.1	Windows	4
2.3.2	Linux	5
3	功能介绍	6
3.1	切换用户	6
3.2	替换命令文本	6
3.3	主动重连数据库	7
3.4	设置控制台属性	7
3.4.1	B_PAUSE	7
3.4.2	DATE_FORMAT	7
3.4.3	ERRQUIT	8
3.4.4	HEADING	8
3.4.5	NULL	9
3.4.6	PAGESIZE	10
3.4.7	PRINT_FORMAT	10
3.4.8	SHOW_LINE_NUMBER	11
3.4.9	SHOW_TIME	11
3.4.10	SHOW_TYPE	12
3.4.11	SHOW_USE_TIME	12

3.4.12	SPUT_HEAD	13
3.5	重置控制台属性	13
3.5.1	HEADING	13
3.5.2	PAGESIZE	13
3.5.3	SHOW_LINE_NUMBER	14
3.5.4	SHOW_TIME	14
3.5.5	SHOW_TYPE	14
3.5.6	SHOW_USE_TIME	14
3.5.7	SPUT_HEAD	14
3.6	清理屏幕	15
3.7	清理输入缓存	15
3.8	重新建立连接	15
3.9	数据导入与导出	15
3.9.1	导入数据	15
3.9.2	导出数据	20
3.10	EDIT 编译命令	21
3.11	执行脚本	21
3.12	查询表信息	21

1 工具概述

虚谷数据库控制台工具是由虚谷数据库提供的连接数据库的客户端工具，类似于查询分析器，它自带驱动与数据库建立连接，和数据库进行会话。

控制台工具目前有 Windows 和 Linux 两种版本，比较常用，功能丰富，主要包括以下几点功能：

- 数据库的维护，如关闭数据库。
- 执行 SQL 语句和 PL/SQL 语句。
- 执行 SQL 脚本。
- 数据的导入和导出。
- 在脚本中使用。

说明

控制台的语句不区分大小写。

2 安装部署

2.1 部署环境要求

2.1.1 Windows

虚谷数据库控制台工具的 Windows 部署环境要求如表2-1所示。

表 2-1 Windows 部署环境

部署环境	环境要求
Windows	x86-64
操作系统	Windows10、Windows11
内存	建议 1GB 以上

2.1.2 Linux

虚谷数据库控制台工具的 Linux 部署环境要求如表2-2所示。

表 2-2 Linux 部署环境

部署环境	环境要求
Linux	X86-64、Arm-64
操作系统	主流的 Linux 发布版、RedHat、CentOS、Ubuntu 等
内存	建议 1GB 以上
ncurses5	如果存在高版本可通过软链接解决

2.2 启动控制台工具

2.2.1 Windows

在虚谷数据库产品包的虚谷数据库-客户端文件中解压“XuguConsole-win-x64”压缩包，然后找到 Windows 版控制台执行文件，默认文件名为“XuguConsole.exe”。

支持以下两种方式启动：

- 直接启动
- 参数启动

直接启动

双击 XGConsole.exe 运行，按照程序提示输入相应的信息，如果连接建立失败，则重新输入。

参数启动

在 CMD 中进入控制台所在的路径，执行以下命令。如果参数个数不正确或者连接建立失败，则会回到命令行界面。

```
C:\console>XGConsole.exe USESSL IP PORT DB_NAME USER_NAME USER_PASS  
[-pf]
```

参数说明：

- USESSL：是否使用加密，nssl 表示不使用加密。
- IP：数据库连接的 IP 地址。
- PORT：端口号。
- DB_NAME：目标库名。
- USER_NAME：目标用户名。
- USER_PASS：目标用户名的密码。
- -pf（可选）：指定结果集输出格式。
 - ALIGNED：表格化输出。
 - NOTHING：紧凑化输出。

示例：

```
C:\console>XGConsole.exe nssl 127.0.0.1 5138 SYSTEM SYSDBA SYSDBA  
C:\console>XGConsole.exe nssl 127.0.0.1 5138 SYSTEM SYSDBA SYSDBA -  
pf ALIGNED
```

2.2.2 Linux

将虚谷数据库产品包下虚谷数据库-客户端文件中的“XuguConsole-linux-x64”压缩包上传至 Linux 服务器，解压并找到 Linux 版本控制台文件，默认文件名为“xgconsole”，增加可执行权限。

支持以下两种方式启动：

- 直接启动
- 参数启动

直接启动

按照程序提示输入相应的信息，如果连接建立失败，则重新输入。

参数启动

进入控制台所在的路径，执行如下命令。如果参数个数不正确或者连接建立失败，则会回到命令行界面。

```
[root@localhose console]#./xgconsole USESSL IP PORT DB_NAME  
USER_NAME USER_PASS [-pf]
```

参数说明：

- USESSL：是否使用加密，nssl 表示不使用加密。
- IP：数据库连接的 IP 地址。
- PORT：端口号。
- DB_NAME：目标库名。
- USER_NAME：目标用户名。
- USER_PASS：目标用户名的密码。
- -pf（可选）：指定结果集输出格式。
 - ALIGNED：表格化输出。
 - NOTHING：紧凑化输出。

示例：

```
[root@localhose console]#./xgconsole nssl 127.0.0.1 5138 SYSTEM  
SYSDBA SYSDBA  
  
[root@localhose console]#./xgconsole nssl 127.0.0.1 5138 SYSTEM  
SYSDBA SYSDBA -pf ALIGNED
```

2.3 退出控制台工具

2.3.1 Windows

单击右上角关闭按钮即可退出控制台工具。

2.3.2 Linux

退出控制台工具。

```
SQL> EXIT;
```

3 功能介绍

3.1 切换用户

功能

实现在同一库中不同用户间进行切换。

关键字

\$

语法格式

```
SQL> $用户名
```

示例

```
SQL> $user_test
```

3.2 替换命令文本

功能

使用新字符串替换输入缓存区中的字符串。

关键字

c

语法格式

```
SQL> c/被替换的文本/替换文本
```

示例

```
SQL> select * from sys_table;  
Error: [E5021 L1 C15] 表或视图SYS_TABLE不存在  
SQL> c/sys_table/sys_tables;
```

说明

替换之后会自动执行缓存区命令。

3.3 主动重连数据库

功能

与数据库重新建立连接。

关键字

RECONNECT 或 RECONN

示例

```
SQL> RECONNECT;
```

3.4 设置控制台属性

3.4.1 B_PAUSE

功能

关闭或开启分页功能（与 PAGESIZE 配合作用）。

语法格式

```
SQL> set B\_PAUSE on|off
```

示例

```
SQL> set B_PAUSE on;
```

3.4.2 DATE_FORMAT

功能

控制 datetime 或 date 类型数据输出格式。

语法格式

```
SQL> set DATE\_FORMAT 任意年月日和中间分隔符的组合
```

示例

```
SQL> set DATE_FORMAT yyyy-mm-dd;
```

```
SQL> select sysdate();
```

```
EXPR1 |
```

```
-----  
2022-10-09 09:30:58.496 AD |
```

```
Total 1 records.
```

```
Use time:0 ms.
```

```
SQL> set DATE_FORMAT mm-YYYY-dd;
```

```
SQL> select sysdate();
```

```
EXPR1 |
```

```
-----  
10-2022-09 09:31:14.952 AD |
```

```
Total 1 records.
```

```
Use time:0 ms.
```

```
SQL> set DATE_FORMAT mm%YYYY%dd;
```

```
SQL> select sysdate();
```

```
EXPR1 |
```

```
-----  
10%2022%09 09:31:30.819 AD |
```

```
Total 1 records.
```

```
Use time:0 ms.
```

3.4.3 ERRQUIT

功能

SQL 执行失败下，直接退出控制台工具，并将错误信息重定向到 stderr 中。

说明

仅在 Linux 版本控制台中支持。

语法格式

```
SQL> set ERRQUIT on|off
```

示例

```
SQL> set ERRQUIT on;
```

3.4.4 HEADING

功能

控制查询结果集是否输出列头信息。

语法格式

```
SQL> set HEADING on|off
```

示例

```
SQL> select 1 from dual;

EXPR1 |
-----
1 |

Total 1 records.
Use time:1 ms.

SQL> set heading off;
HEADING OFF

SQL> select 1 from dual;
1 |

Total 1 records.
Use time:0 ms.
```

3.4.5 NULL

功能

控制空列值输出内容。

语法格式

```
SQL> set NULL 空值输出内容
```

示例

```
SQL> select null from dual;

EXPR1 |
-----
<NULL> |

Total 1 records.
Use time:0 ms.

SQL> set NULL NULL;
NULL will be display as NULL

SQL> select null from dual;

EXPR1 |
-----
NULL |

Total 1 records.
Use time:0 ms.
```

3.4.6 PAGESIZE

功能

开启结果集分页功能并设置结果集分页尺寸。

语法格式

```
SQL> set PAGESIZE XXX
```

示例

```
SQL> set pagesize 10;
pagesize 10

SQL> select ACTIVITY_CODE from txn_proc_result limit 100;

ACTIVITY_CODE |
-----
013022 |
013022 |
013022 |
013022 |
013022 |
013022 |
013022 |
013022 |
013022 |
013022 |
013022 |
```

📖 说明

开启分页功能时可以使用 Q 或者 ESC 退出结果集展示。

3.4.7 PRINT_FORMAT

功能

控制查询结果输出格式。

语法格式

```
SQL> set PRINT_FORMAT ALIGNED|NOTHING
```

参数说明

- NOTHING: 紧凑式输出（默认）
- ALIGNED: 表格式输出

示例

```
SQL> SELECT 1 COL1,2 COL2 FROM DUAL;
```

```
COL1 | COL2 |
```

```
-----  
1 | 2 |
```

```
SQL> SET PRINT_FORMAT ALIGNED  
PRINT_FORMAT SET ALIGNED
```

```
SQL> SELECT 1 COL1,2 COL2 FROM DUAL;
```

```
+-----+-----+  
| col1 | col2 |  
+-----+-----+  
| 1    | 2    |  
+-----+-----+  
(1 row)
```

📖 说明

开启分页功能时可以使用 Q 或者 ESC 退出结果集展示。

3.4.8 SHOW_LINE_NUMBER

功能

控制输入多行 SQL 的时候，是否在每行开头展示当前行数。

语法格式

```
SQL> set SHOW\_LINE\_NUMBER on|off
```

示例

```
SQL> set show_line_number on;  
SHOW_LINE_NUMBER ON
```

```
SQL> select 1  
2      from dual;
```

3.4.9 SHOW_TIME

功能

控制输入命令时是否展示当前时间。

语法格式

```
SQL> set SHOW\_TIME on|off
```


示例

```
SQL> set SHOW_TIME ON;
SHOW_TIME ON

09:23:30 SQL>
```

3.4.10 SHOW_TYPE

功能

控制结果集列头是否展示列的数据类型和精度（需要先开启列头信息 heading）。

语法格式

```
SQL> set SHOW\_TYPE on|off
```

示例

```
SQL> set show_type on;
SHOW_TYPE ON

SQL> select 1 from dual;

EXPR1 (TINYINT) |
-----
1 |

Total 1 records.
Use time:0 ms.
```

3.4.11 SHOW_USE_TIME

功能

控制是否输出语句执行时间。

语法格式

```
SQL> set SHOW\_USE\_TIME on|off
```

示例

```
SQL> select 1 from dual;

EXPR1 |
-----
1 |

Total 1 records.
Use time:0 ms.

SQL> set SHOW_USE_TIME Off;
SHOW_USE_TIME OFF
```

```
SQL> select 1 from dual;

EXPR1 |
-----
1 |

Total 1 records.
```

3.4.12 SPUT_HEAD

功能

控制将结果输入到文件时是否输入列名信息。

语法格式

```
SQL> set SPUT\_HEAD on|off
```

示例

```
SQL> set sput_head on;
SPUT_HEAD ON

SQL> select * from t_var; >$ d:/test.csv;

Total:15 record
Use time:5 ms.
```

3.5 重置控制台属性

3.5.1 HEADING

默认值

ON

语法格式

```
SQL> reset HEADING
```

3.5.2 PAGESIZE

默认值

10

语法格式

```
SQL> reset PAGESIZE
```

3.5.3 SHOW_LINE_NUMBER

默认值

OFF

语法格式

```
SQL> reset SHOW_LINE_NUMBER
```

3.5.4 SHOW_TIME

默认值

OFF

语法格式

```
SQL> reset SHOW_TIME
```

3.5.5 SHOW_TYPE

默认值

OFF

语法格式

```
SQL> reset SHOW_TYPE
```

3.5.6 SHOW_USE_TIME

默认值

ON

语法格式

```
SQL> reset SHOW_USE_TIME
```

3.5.7 SPUT_HEAD

默认值

OFF

语法格式

```
SQL> reset SPUT_HEAD
```

3.6 清理屏幕

功能

清空屏幕输出。

关键字

CLS

示例

```
SQL> CLS;
```

3.7 清理输入缓存

功能

清空输入缓存区里面的命令。

关键字

CLBUFF

示例

```
SQL> CLBUFF;
```

3.8 重新建立连接

功能

重新输入信息连接新的数据库。

关键字

LOGIN

示例

```
SQL> LOGIN;
```

3.9 数据导入与导出

3.9.1 导入数据

3.9.1.1 SQLLDR

功能

SQLLDR 是文件导入命令，可以将控制台工具或者其他工具导出的结果集文件导入到数据库的表中，需要注意的是文件的列数要与表的列数一致，并且是顺序对应的关系。

语法说明

```
SQLLDR TABLE=[schema_name.]tab_name DATAFILE=filepath FT=ft RT=rt [
  LOG=logpath] [ERRLOG=errlogpath] [MODE=replace|append] [IMP_TYPE=
  block|insert|auto|store] [CSV_DEAL=true|false] [LAST_FT=true|
  false]
```

参数说明

- SQLLDR：导入命令标识符。
- TABLE：导入的表。
 - schema_name.：可选，模式名。
 - tab_name：目标表名。
- DATAFILE：数据文件的路径 + 文件名。
- FT：列分隔符，如 |、\、等。
- RT：行结尾标识符，可设为 X'0a' 或 X'0d0a'。
 - X'0a'：行结尾换行为\n。
 - X'0d0a'：行结尾换行为\r\n。
- LOG：可选，SQLLDR 运行日志记载路径。
- ERRLOG：可选，SQLLDR 错误信息日志记载路径。
- MODE：可选，导入方式，可设为 append 或 replace，默认为 append。
 - append：追加导入。
 - replace：覆盖导入。
- IMP_TYPE：可选，导入类型，可设为 block、insert、auto 或 store。
 - block：按块导入。
 - insert：插入导入。
 - auto：自动分配，V12 及以上版本的数据库不支持此功能。
 - store：存储导入，V12 及以上版本的数据库不支持此功能。
- CSV_DEAL：可选，格式化 CSV 文件，去除单元格开头双引号，可设为 true 或 false，默

认为 false。若数据文件不为 CSV 格式时，无需指定该参数。

- LAST_FT: 可选，每行最后一列数据是否自带列分标志位，可设为 true 或 false，默认为 false。示例：数据表有 A、B、C 共 3 个列，此时文件数据为 1、2、3，则 LAST_FT=true；若文件数据为 1、2，则 LAST_FT=false。

📖 说明

控制台 V1.4.6 版本开始支持 LAST_FT 参数。

示例

将 F:\tpcc100w\warehouse.txt 文件中的数据导入到 sysdba.warehouse 数据表中。

```
SQL> sqlldr table=sysdba.warehouse datafile=F:\tpcc100w\warehouse.txt ft='|' rt=X'0d0a' log=./load.log errlog=./loaderr.log mode=replace imp_type=block
```

3.9.1.2 TABLDR

功能

TABLDR 是文件导入命令，和 SQLLDR 一样可以将控制台工具或者其他工具导出的结果集文件导入到数据库的表中，TABLDR 导入的效率高于 SQLLDR。

语法格式

```
TABLDR TABLE=[schema_name.]tab_name DATAFILE=filepath FT=ft RT=rt [LOG=logpath] [ERRLOG=errlogpath] [PARSE_PARE=parse_num] [INS_PARE=ins_num] [MODE=replace|append] [CSV_DEAL=true|false] [LAST_FT=true|false]
```

参数说明

- TABLDR: 导入命令标识符。
- TABLE: 导入的表。
 - schema_name.: 可选，模式名。
 - tab_name: 目标表名。
- DATAFILE: 数据文件的路径 + 文件名。
- FT: 列分隔符，如 |、,、等。
- RT: 行结尾标识符，可设为 X'0a' 或 X'0d0a'。

- X'0a': 行结尾换行为\n。
- X'0d0a': 行结尾换行为\r\n。
- LOG: 可选, TABLDR 运行日志记载路径。
- ERRLOG: 可选, TABLDR 错误信息日志记载路径。
- PARSE_PARE: 可选, 服务端解析数据线程数, 从 1 开始, 默认为 1。
- INS_PARE: 可选, 服务端插入数据线程数, 从 1 开始, 默认为 1。
- MODE: 可选, 导入方式, 可设为 append 或 replace, 默认为 append。
 - append: 追加导入。
 - replace: 覆盖导入。
- CSV_DEAL: 可选, 格式化 CSV 文件, 去除单元格开头双引号, 可设为 true 或 false, 默认为 false。若数据文件不为 CSV 格式时, 无需指定该参数。
- LAST_FT: 可选, 每行最后一列数据是否自带列分标志位, 可设为 true 或 false, 默认为 false。示例: 数据表有 A、B、C 共 3 个列, 此时文件数据为 1、2、3, 则 LAST_FT=true; 若文件数据为 1、2, 则 LAST_FT=false。

说明

控制台 V1.4.6 版本开始支持 LAST_FT 参数。

示例

先将查询结果集导出成文件, 再将文件导入到数据表中。

```
SQL> tabldr table=NATION datafile=D:/NATION.TXT ft='|' rt=X'0d0a'  
log=./load.log errlog=./loaderr.log mode=APPEND parse_para=8  
ins_para=8
```

异常处理

数据库新增参数 stream_import_error 用于控制 TABLDR 导入异常数据时的处理模式, 用户可根据个人需求自行设置该参数。

- 0: 严格报错, 数据库解析时出现异常连接断开, 控制台工具退出停止导入
- 1: 跳过错误行并记载日志, 数据库解析时出现异常则忽略异常数据并将异常数据记录至 TRACE.LOG 中并继续导入后续数据

- 2: 跳过错误行不记载日志，数据库解析时出现异常则忽略异常数据且不记载日志继续导入后续数据

 **注意**

日志记载有长度限制则可能导致原始数据不会被完全记载。

支持的行数据异常：

- 分隔符不一致
- 字段数据中包含分隔符
- 类型转换错误

3.9.1.3 COLLDLDR

功能

COLLDLDR 是文件导入命令，不同于 TABLLDR，COLLDLDR 使用 INSERT 语句导入数据，可以实现部分列的导入功能。

语法格式

```
COLLDLDR TABLE=tab_name DATAFILE=filepath FT=ft RT=rt COLHEAD=colhead
```

参数说明

- COLLDLDR：导入命令标识符。
- TABLE：目标表名。
- DATAFILE：数据文件的路径 + 文件名。
- FT：列分隔符，如 |、, 等。
- RT：行结尾标识符，可设为 X'0a' 或 X'0d0a'。
 - X'0a'：行结尾换行为\n。
 - X'0d0a'：行结尾换行为\r\n。
- COLHEAD：导入文件的列名，列名之间使用逗号, 分隔。

📖 说明

不设置 `colhead` 参数表示列名在文件的第一行中存在。

示例

先将查询结果集导出成文件，再将文件导入到数据表中。

```
SQL> collldr table=t_var datafile=d:/test.csv ft=',' rt=X'0d0a'  
colhead=id,var
```

3.9.1.4 导入大对象

功能

使用控制台可以将大对象导入到表中。

示例

```
SQL> CREATE TABLE test(B BLOB, C CLOB);  
  
-- 导入图片  
SQL> INSERT INTO test(B) VALUES (?); <# D:\t1.png;  
  
-- 导入文本  
SQL> INSERT INTO test(C) VALUES (?); <% D:\t3.txt;
```

3.9.2 导出数据

3.9.2.1 导出结果集

功能

使用控制台可以将 SELECT 查询的结果导出到文本文件中，导出的数据默认以 “|” 作为分隔符（csv 文件以 “,” 作为分隔符）。导出的结果集中不能包含大对象，大对象需要单独导出。

示例

```
SQL> SELECT*FROM test; >$ D:\test_select.txt;
```

3.9.2.2 导出大对象

功能

使用控制台可以将 SELECT 查询的大对象导出中。

示例

```
SQL> CREATE TABLE test (B BLOB, C CLOB);  
  
-- 导出图片  
SQL> SELECT B FROM test; ># D:\t2.png;  
  
-- 导出文本  
SQL> SELECT C FROM test; >% D:\t4.txt;
```

3.10 EDIT 编译命令

功能

控制台会将上一次执行过的 SQL 命令（包括 SQL 语句和 PL/SQL 语句，一行或多行）信息保存在缓冲区，输入“EDIT”进行编辑，编辑后通过“/”命令执行修改后的缓冲区中的命令。

示例

```
SQL> EDIT  
SQL> /
```

3.11 执行脚本

操作步骤

1. 准备脚本。

在脚本中用 EXECBEGIN/EXECEND 包裹 SQL 语句，格式如下：

```
EXECBEGIN  
... ..  
多条 SQL 命令  
... ..  
EXECEND
```

2. 执行脚本。

```
SQL> EXECFILE C:\test_execfile.sql  
  
SQL> EXECFILE2 C:\test_execfile.sql
```

说明

若使用 EXECFILE2 执行脚本文件，则脚本内的 SQL 语句不需要用 EXECBEGIN/EXECEND 包裹，直接列出即可。

3.12 查询表信息

功能

查询表的列信息，包括列名、是否为空和列类型。

关键字

DESC

语法格式

```
DESC 表名
```

示例

```
SQL> DESC sys_tables;
```

Column_Name	NULL	
Data_Type		
DB_ID		
INTEGER		
USER_ID		
INTEGER		
SCHEMA_ID		
INTEGER		
TABLE_ID		
INTEGER		
TABLE_NAME		CHAR
(128)		
TABLE_TYPE		
INTEGER		
TEMP_TYPE		
INTEGER		
FIELD_NUM		
INTEGER		
PARTI_TYPE		
INTEGER		
PARTI_NUM		
INTEGER		
PARTI_KEY		CHAR
(1024)		
AUTO_PARTI_TYPE		
INTEGER		
AUTO_PARTI_SPAN		
INTEGER		
SUBPARTI_TYPE		
INTEGER		
SUBPARTI_NUM		
INTEGER		
SUBPARTI_KEY		CHAR
(1024)		
GSTO_NO		
INTEGER		
COPY_NUM		
INTEGER		

BLOCK_SIZE			
INTEGER			
CHUNK_SIZE			
INTEGER			
RECORD_NUM			
BIGINT			
PCTFREE			
INTEGER			
HOTSPOT_NUM			
INTEGER			
USE_CACHE			
BOOLEAN			
ONLINE			
BOOLEAN			
IS_SYS			
BOOLEAN			
IS_ENCRY			
BOOLEAN			
SLOW_MODIFY			
BOOLEAN			
XLS_PID			
INTEGER			
XLS_COL_NO			
INTEGER			
XLS_COL_OPT			
INTEGER			
ON_COMMIT_DEL			
BOOLEAN			
ENA_TRANS			
BOOLEAN			
ENA_LOGGING			
BOOLEAN			
REG_MODIFY			
BOOLEAN			
VALID			
BOOLEAN			
ACL_MASK			
INTEGER			
AUTO_PARTI_NO			
INTEGER			
CREATE_TIME			
DATETIME			
COMMENTS			
(128)			CHAR
ANA_POLICY			
INTEGER			
ANA_PERIOD			
INTEGER			
ANA_THRESHOLD			
INTEGER			
ANA_LEVEL			
INTEGER			
ENCRY_ID			
INTEGER			
STO_ZONE			

INTEGER		
CACHE_HASH_KEY (1024)		CHAR
COMPRESS_LEVEL		
INTEGER		
RESERVED5 (128)		CHAR



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