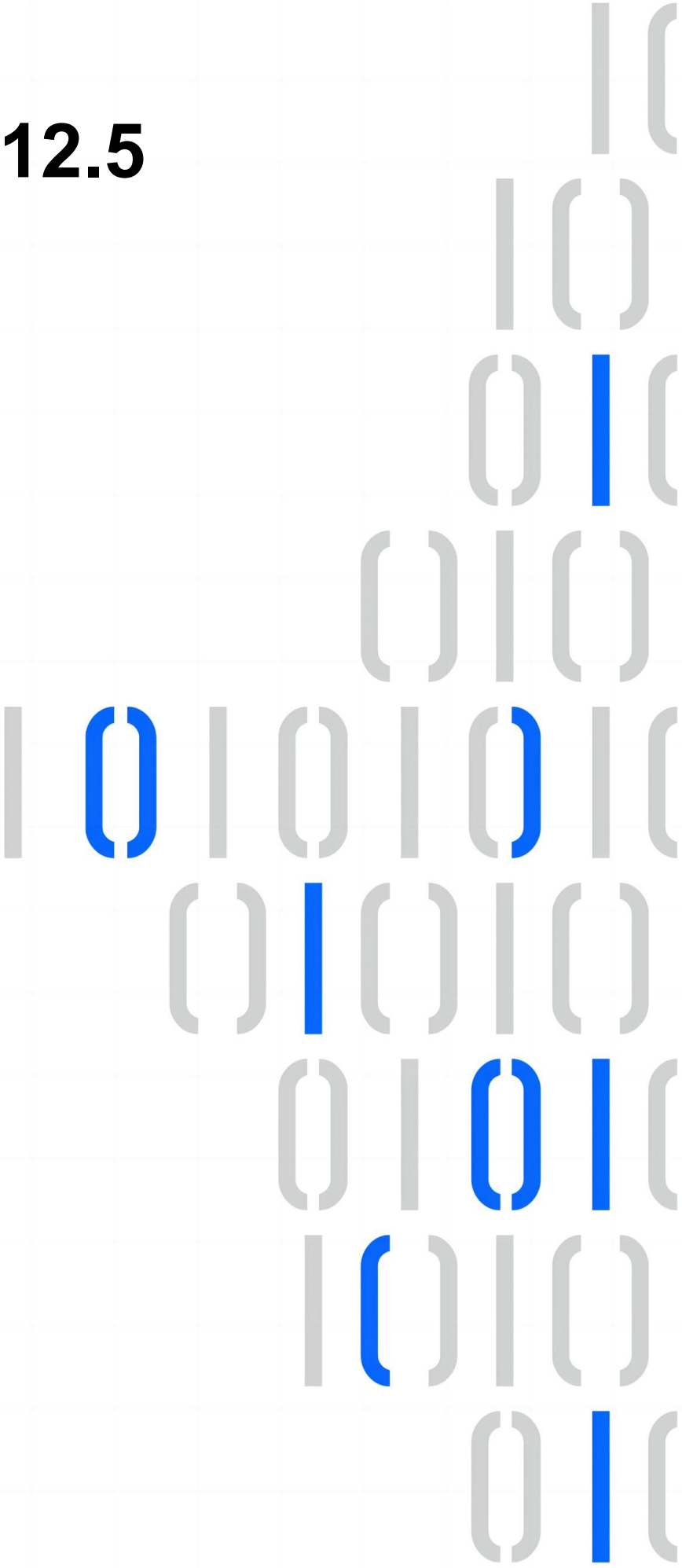
虚谷数据库 V12.5

安全管理指南

文档版本 01

发布日期 2024-07-30



虚谷数据库
XUGU DB

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

本文档主要介绍了数据库软件系统安全以及安全管理相关内容。

读者对象

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
⚠ 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	2024-07-30	第一次发布

目录

1 安全概述

数据库安全，包括软件系统安全、运行环境安全、安全管理规范等等。本文主要介绍数据库软件系统安全。

为了保障数据库安全，需要保护系统数据、用户数据等数据以及保证数据库安全运行，并应对各种威胁，如误操作、恶意攻击、服务失效等情况。

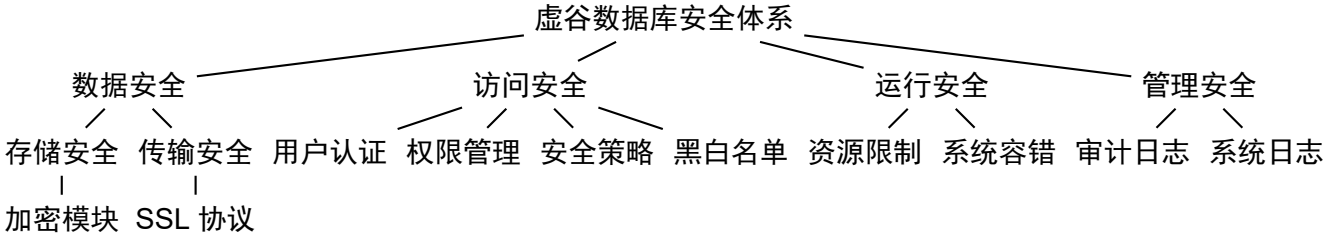
数据库采用各种机制、工具和手段，以保护数据和应对威胁。例如，数据库采用三权分治的模型，同时拥有四级权限，还通过强制访问控制、审计、加密等措施，保障数据库安全。

1.1 安全功能

虚谷数据库是一款自研的成熟商业化数据库产品，其核心组件包括管理、存储和计算，均自主研发，并未使用任何开源框架，完全避免了开源软件带来的安全风险和技术限制。虚谷数据库拥有一套完整的安全体系，包括身份验证、自主访问控制、强制访问控制、加密机制、安全审计等模块，保证用户使用产品过程中的数据安全、运行安全、访问安全以及管理安全。

1.2 安全体系

虚谷数据库的安全体系，从数据安全、访问安全、运行安全以及管理安全等维度保障数据库安全，降低使用风险。虚谷数据库的安全体系如下所示。



2 用户鉴权

2.1 用户策略

数据库用户鉴别相关参数包含在配置文件 xugu.ini 中，数据库管理员可通过管理配置文件以定义用户策略。参数设置详见数据库配置文件（xugu.ini），也可以由数据库管理员（SYSDBA）登入数据库后以命令方式查看和修改。

身份鉴别相关参数	参数含义	缺省值
min_pass_len	口令的最短长度	8
pass_mode	口令模式 1：任意字符或数字 2：必须含字母和数字 3：必须含字母和数字和特殊符号	3
conn_fail_cnt	禁止登录的失败次数	3

通过命令方式查看以及修改用户策略命令如下。

- 通过命令查看口令模式

```
SQL> SHOW PASS_MODE;

PASS_MODE |
----------------------------------------------------------------
1 |
```

- 通过命令修改口令模式

```
SQL> SET PASS_MODE TO 3;
```

⚠ 注意

- 只有数据库管理员有权限修改数据库参数。
- 通过命令方式修改用户口令策略直接生效，通过更改 xugu.ini 配置文件需重启生效。

鉴权时机

虚谷数据库在用户进行数据库登录时，除必要的连接信息如 IP、端口号、库名外，会根据输入的用户名、密钥等进行用户鉴别，同时还会判断登录信息是否通过黑白名单规则，未通过身份鉴别的用户无法登入数据库，通过身份鉴别的用户成功登录数据库。

鉴权方式

虚谷数据库鉴别用户身份的方式为数据库鉴别，即验证用户的口令，暂未提供其他的认证方式。

失败处理

对于用户登录失败的情况，虚谷数据库有相应的安全处理机制。默认以错误密码 3 次登录数据库后锁定该 IP 5 分钟，5 分钟内此 IP 任何用户即使输入正确密码也无法登录。数据库管理员可通过命令或者直接修改配置文件 xugu.ini 中的 conn_fail_cnt 参数自定义锁定 IP 的失败次数。通过系统表 SYS_ALL_FORBIDDEN_IPS 可查看相关失败情况。系统表记载详情请参见《系统字典参考》的访问控制章节的系统表 SYS_ALL_FORBIDDEN_IPS。

2.2 密钥存储

数据库用户密钥使用加密存储的方式存储在用户系统表 SYS_USERS 中，保证用户认证信息安全。

```
SQL> SELECT db_id,user_id,user_name,password FROM SYS_USERS;

DB_ID | USER_ID | USER_NAME | PASSWORD |
-----------------------------------------------------------
1 | 1 | SYSDBA| <BINARY> |
1 | 2 | SYSSSO| <BINARY> |
1 | 3 | SYSAUDITOR| <BINARY> |
1 | 4 | GUEST| <BINARY> |
```

3 权限管理

3.1 权限简介

在虚谷数据库中，权限的最原始来源为天然权限，作为各级对象的拥有者自然具备该对象的天然权限，天然权限几乎等于该对象的一切权限（除数据库拥有者不能同时作为安全员及审计员外），也就是数据库的拥有者作为 DBO（Database Owner）自然拥有数据库的一切权限。即使未对该拥有者赋予任何权限，只要属主为该拥有者就可以对属主是他的对象进行一切操作。例如：新用户 A 未被赋予任何权限，如果用户 B 在用户 A 模式下创建了表对象，则用户 A 拥有对该表的一切操作权限，包括增删改查以及删除表。

通过 DBO 授权方能产生 DBA、审计员、安全管理员、可创建资源的用户等，DBA 或创建资源的用户可以创建表级资源，如：表、视图、存储过程、序列值等对象，创建者自然作为这些对象的拥有者，具有这些对象上的一切权限。DBA 也具有在一切表级对象上的全部权限，当创建对象的用户被删除后，该用户的数据库对象也一并删除。权限的授予者或 DBA 可以从被授权者收回全部或部分权限，若非 DBA，则只能收回其授出的权限，不能收回其他用户授出的权限。

在虚谷数据库系统中，权限继承主要针对用户与角色或角色与角色，用户继承其所属角色所有权限；用户之间不存在权限继承关系。

3.2 权限分类

在虚谷数据库中权限可以分为四类，分别是库级权限、模式级权限、对象级权限、列级权限。他们作用的范围如下：

- 库级权限：管理粒度为整个逻辑库，拥有库级权限的用户或角色可对该库下所有对象进行权限允许范围内的操作。
- 模式级权限：管理粒度为模式，拥有模式级权限的用户或角色可对该模式下所有对象进行权限允许范围内的操作。
- 对象级权限：管理粒度为对象，拥有对象级权限的用户或角色可对指定对象进行权限允许范围内的操作。
- 列级权限：管理粒度为表列或视图列，拥有列级权限的用户或角色可访问和操作权限允许

范围内的表列或视图列。

3.3 权限授予

3.3.1 概述

在数据库系统中除了用户或角色的天然权限无需进行权限授予外，其他任何权限均需通过权限授予的方式进行权限赋予。

语法格式

```
GrantStmt::=
GRANT privileges ON name_space TO grantee_list opt_with_grant
|   GRANT privileges ON TABLE name_space TO grantee_list
   opt_with_grant
|   GRANT privileges ON VIEW name_space TO grantee_list
   opt_with_grant
|   GRANT privileges ON PROCEDURE name_space TO grantee_list
   opt_with_grant
|   GRANT privileges ON SEQUENCE name_space TO grantee_list
   opt_with_grant
|   GRANT sys_privileges IN SCHEMA name TO grantee_list
   opt_with_grant
|   GRANT sys_privileges TO grantee_list opt_with_grant
|   GRANT operation_commalist ( name_list ) ON name_space TO
   grantee_list  opt_with_grant
|   GRANT operation_commalist ( name_list ) ON TABLE name_space TO
   grantee_list  opt_with_grant

privileges::=
ALL PRIVILEGES
|   ALL
|   operation [,operation]...

operation::=
{SELECT | INSERT | UPDATE | DELETE | EXECUTE | REFERENCES | ALTER
    | DROP | INDEX | TRIGGER | VACUUM}

grantee_list::=
{ROLE  UserId | UserId} [{,} {ROLE  UserId | UserId}]...

opt_with_grant::=
[WITH GRANT OPTION]

sys_privileges::=
sys_privilege [,sys_privilege]

sys_privilege::=
{DBA | SSO | AUDITOR | BACKUP DATABASE | BACKUP | RESTORE | RESTORE
    DATABASE | sys_operation obj_type}

sys_operation::=
```

```
{CREATE | CREATE ANY | ALTER ANY | DROP ANY | SELECT ANY | INSERT
   ANY | UPDATE ANY | DELETE ANY | EXECUTE ANY | REFERENCES ANY |
   VACUUM ANY | ENCRYPT ANY}

obj_type::=
{DATABASE | SCHEMA | TABLE | SEQUENCE | INDEX | VIEW | PROCEDURE |
   PACKAGE | TRIGGER | DATABASE LINK | REPLICATION | SYNONYM | USER
    | ROLE | JOB}

operation_commalist::=
operation [,operation]...
```

参数解释

- privileges：授予权限定义，可以是所有权限也可以是多个对象组合权限。
- grantee_list：被授予权限角色组或用户组，可同时将权限授予多个用户或角色，若权限授予角色则角色下的所有用户均自动获得该授予权限。
- opt_with_grant：权限可转授，权限的赋予和回收是关联的，如将 with grant option 用于对象授权时，被授予的用户也可把此对象权限授予其他用户或角色，权限回收时转授的权限会一同被收回，with grant option 只能在授予对象级和列级权限时使用。
- sys_privileges：库级权限定义。
- operation_commalist：可授予的操作类型，包含 DDL、DML 类型。
- operation：表示操作类型，包含 DDL、DML 类型，如：CREATE、ALTER、DROP、INSERT、DELETE、UPDATE、SELECT 等，若选择 ANY 参数则表示被授权对象可跨模式进行操作，否则只能在其所有模式进行相应操作。
- UserId：此处指被授权的用户名称或角色名称。
- obj_type：表示操作对象类型，包含 TABLE、VIEW、PROCEDURE 等。

3.3.2 授予库级权限

语法格式

```
|   GRANT sys_privileges TO grantee_list opt_with_grant
```

参数解释

- sys_privileges：库级权限定义。
- grantee_list：被授予权限角色组或用户组，可同时将权限授予多个用户或角色，若权限授予角色则角色下的所有用户均自动获得该授予权限。

- opt_with_grant：权限可转授，权限的赋予和回收是关联的，如将 with grant option 用于对象授权时，被授予的用户也可把此对象权限授予其他用户或角色，权限回收时转授的权限会一同被收回，with grant option 只能在授予对象级和列级权限时使用。

> **⚠ 注意**
>
> 授权操作类型与操作对象类型应保证匹配，例如针对 PROCEDURE 对象，可为其赋予 CREATE、ALTER、DROP、EXECUTE 等权限，但不能赋予 INSERT、UPDATE、DELETE、SELECT 等权限。

示例

```
GRANT CREATE ANY TABLE TO test_user;
```

该示例表示授予 test_user 在当前库下所有模式的创表权限，若无 ANY 关键字则表示，test_user 只能在属于他的模式下进行创表。

3.3.3 授予模式级权限

语法格式

```
|   GRANT sys_privileges IN SCHEMA name TO grantee_list
  opt_with_grant
```

参数解释

- sys_privileges：库级权限定义。
- name：操作模式名，即所授予的权限仅限于指定模式对象。
- grantee_list：被授权的用户名或角色名。
- opt_with_grant：权限可转授，权限的赋予和回收是关联的，如将 with grant option 用于对象授权时，被授予的用户也可把此对象权限授予其他用户或角色，权限回收时转授的权限会一同被收回，with grant option 只能在授予对象级和列级权限时使用。

示例

```
GRANT CREATE ANY TABLE IN SCHEMA SYSDBA TO role_1;
```

该示例表示授予角色 role_1 在模式 SYSDBA 下的创表权限。

3.3.3.1 授予对象级权限

语法格式

```
|   GRANT privileges ON name_space TO grantee_list opt_with_grant
|   GRANT privileges ON TABLE name_space TO grantee_list
  opt_with_grant
|   GRANT privileges ON VIEW name_space TO grantee_list
  opt_with_grant
|   GRANT privileges ON PROCEDURE name_space TO grantee_list
  opt_with_grant
|   GRANT privileges ON SEQUENCE name_space TO grantee_list
  opt_with_grant
```

参数解释

- privileges：授予权限操作类型，包括 SELECT、UPDATE、EXECUTE 等，若要授予对象的所有可操作权限，可使用 ALL 或 ALL PRIVILEGES 代替。
- name_space：此处为指定授予操作权限对象名，对象名与 privileges 操作类型必须匹配，如该对象为 TABLE，则不能授予 EXECUTE 权限。
- grantee_list：被授予权限的用户名或角色名。

示例

- 示例 1

```
CREATE TABLE test_permission(id int);
CREATE USER u1 IDENTIFIED BY 'test_123@';
GRANT INSERT ON test_permission TO u1;
```

该示例表示授予用户 U1 对于表 test_permission 的数据插入权限，则用户 U1 对表 test_permisssion 无删除、查询、更改、引用等权限。

- 示例 2

```
GRANT ALL ON test_permission TO u1;
```

该示例表示授予用户 U1 对于表 test_premission 所有数据库所允许的操作权限，包括：INSERT、UPDATE、DELETE、SELECT、REFERENCES 等。

3.3.4 授予列级权限

语法格式

```
|   GRANT operation_commalist ( name_list ) ON name_space TO
  grantee_list  opt_with_grant
|   GRANT operation_commalist ( name_list ) ON TABLE name_space TO
  grantee_list  opt_with_grant
```

参数解释

- operation_commalist：表示操作类型，包含 DDL、DML 类型，如 CREATE、ALTER、DROP、INSERT、DELETE、UPDATE、SELECT 等。
- name_list：授予可操作的列名。
- name_space：此处指授予可操作的对象名，可为表名或视图名。
- grantee_list：授予权限的用户名或角色名。

示例

```
CREATE TABLE TEST_COL_PRE(ID INT,ID2 INT,ID3 INT);
CREATE USER TU1 IDENTIFIED BY 'test_123@';
GRANT SELECT(ID2) ON TEST_COL_PRE TO TU1;
GRANT UPDATE(ID) ON TEST_COL_PRE TO TU1;
```

该示例表示授予用户 TU1 针对表 TEST_COL_PRE 的列 ID2 的查询权限与列 ID 的变更权限。

> **⚠ 注意**
>
> 列级权限只包括 SELECT 与 UPDATE 操作权限，且必须指定可操作的对象名，对象类型只能是表或视图。

3.4 权限回收

3.4.1 概述

在数据库系统中，除了需要对于用户或角色进行权限授予，还需根据实际情况对其进行权限回收。权限回收的语法格式与权限授予一一匹配，在权限回收时仅需将权限授予的关键字 GRANT、TO 改为 REVOKE、FROM 即可。

同理，权限回收也可包括：库级权限回收、模式级权限回收、对象级权限回收与列级权限回收。

> **📖 说明**
>
> 权限回收在执行后立即生效。

语法格式

```
RevokeStmt::=
```

```
REVOKE privileges ON name_space FROM grantee_list
|   REVOKE privileges ON {TABLE | VIEW | PROCEDURE | PACKAGE |
   SEQUENCE} name_space FROM grantee_list
|   REVOKE GRANT OPTION FOR privileges ON name_space FROM
   grantee_list
|   REVOKE GRANT OPTION FOR sys_privileges FROM grantee_list
|   REVOKE GRANT OPTION FOR sys_privileges IN SCHEMA name FROM
   grantee_list
|   REVOKE GRANT OPTION FOR operation_commalist ( name_list ) ON
   name_space FROM grantee_list
|   REVOKE sys_privileges FROM grantee_list
|   REVOKE sys_privileges IN SCHEMA name FROM grantee_list
|   REVOKE operation_commalist ( name_list ) ON name_space FROM
   grantee_list
|   REVOKE operation_commalist ( name_list ) ON TABLE name_space
   FROM grantee_list
```

3.4.2 回收库级权限

语法格式

```
|   REVOKE GRANT OPTION FOR sys_privileges FROM grantee_list
|   REVOKE sys_privileges FROM grantee_list
```

参数解释

- REVOKE GRANT OPTION FOR：回收 GRANT 语句中指定的 WITH GRANT OPTION 的转授权限，用户仍然具有该权限，但是不能将该权限转授予其他用户。
- sys_privileges：回收权限操作类型，包含 DDL、DML 类型。
- grantee_list：被回收权限的用户名或角色名。

3.4.2.1 回收模式级权限

语法格式

```
|   REVOKE GRANT OPTION FOR sys_privileges IN SCHEMA name FROM
   grantee_list
|   REVOKE sys_privileges IN SCHEMA name FROM grantee_list
```

参数解释

- sys_privileges：回收权限操作类型，包含 DDL、DML 类型。
- name：操作模式名，即所回收的权限所属模式。
- grantee_list：被回收权限的用户名或角色名。

3.4.2.2 回收对象级权限

语法格式

```
|   REVOKE privileges ON name_space FROM grantee_list
|   REVOKE privileges ON {TABLE | VIEW | PROCEDURE | PACKAGE |
  SEQUENCE} name_space FROM grantee_list
|   REVOKE GRANT OPTION FOR privileges ON name_space FROM
  grantee_list
```

参数解释

- privileges：回收权限操作类型，包括 SELECT、UPDATE、EXECUTE 等，若要回收对象的所有可操作权限，可使用 ALL 或 ALL PRIVILEGES 代替。
- name_space：此处为回收操作权限的对象名。
- grantee_list：被回收权限的用户名或角色名。

3.4.2.3 回收列级权限

语法格式

```
|   REVOKE GRANT OPTION FOR operation_commalist ( name_list ) ON
  name_space FROM grantee_list
|   REVOKE operation_commalist ( name_list ) ON name_space FROM
  grantee_list
|   REVOKE operation_commalist ( name_list ) ON TABLE name_space
  FROM grantee_list
```

参数解释

- operation_commalist：回收权限操作类型信息。
- name_space：此处为回收权限的操作对象名。
- grantee_list：被回收权限的用户名或角色名。

3.5 管理方式

数据库所有用户和角色的权限信息通过系统表 SYS_ACLS 进行管理，系统表 SYS_ACLS 用于管理数据库用户访问控制信息。在系统表中拥有的权限 AUTHORITY 是由十进制数值体现，将 AUTHORITY 值转化为十六进制参照 AUTHORITY 解析即为具体的权限。

字段说明

字段名	字段类型	说明
DB_ID	INTEGER	库 ID
GRANTOR_ID	INTEGER	授权者 ID
GRANTEE_ID	INTEGER	被授权者 ID
OBJECT_ID	INTEGER	授权客体 ID
OBJECT_TYPE	INTEGER	授权客体类型
AUTHORITY	BIGINT	权限值
REGRANT	BIGINT	转授权
ORG_GRANTOR_ID	INTEGER	权限的原始拥有者
IS_SYS	BOOLEAN	是否系统内建

AUTHORITY 解析

权限级别	权限标识	权限值	说明
对象级	ACL_READ	0x1	读取权
对象级	ACL_UPDATE	0x2	更改权
对象级	ACL_INSERT	0x4	添加权
对象级	ACL_DELETE	0x8	删除权
对象级	ACL_REF	0x10	引用权
对象级	ACL_EXECUTE	0x20	执行权
对象级	ACL_INDEX	0x40	可创建索引
对象级	ACL_ALTER	0x80	修改对象结构权

接下页

权限级别	权限标识	权限值	说明
对象级	ACL_DROP	0x100	删除对象权
对象级	ACL_TRIG	0x200	可在对象上创建触发器
对象级	ACL_VACUUM	0x400	可清多余空间
对象级	ACL_TAB_ALL	0x7df	所有表 (字段) 级权
对象级	ACL_PROC_ALL	0x1a0	所有存储过程函数级权
对象级	ACL_VIEW_ALL	0x18f	所有视图级权
对象级	ACL_PACK_ALL	0x1a0	所有包级权
对象级	ACL_SEQ_ALL	0x183	所有序列值级权
对象级	ACL_UDT_ALL	0x1a0	所有自定义数据类型级权
库级	ACL_READ_ANY	0x1	对任何指定类型对象 (对象类型由 obj_type 确定) 拥有读取权
库级	ACL_UPDATE_ANY	0x2	对任何指定类型对象 (对象类型由 obj_type 确定) 拥有更改权
库级	ACL_INSERT_ANY	0x4	对全部表拥有插入权
库级	ACL_DELETE_ANY	0x8	对全部表拥有删除权
库级	ACL_REF_ANY	0x10	对全部表拥有引用权

接下页

权限级别	权限标识	权限值	说明
库级	ACL_EXECUTE_ANY	0x20	对全部过程拥有执行权
库级	ACL_INDEX_ANY	0x40	可创建索引
库级	ACL_CREATE	0x80	可在自身模式下创建任何指定类型对象 (对象类型由 obj_type 确定)
库级	ACL_CRE_ANY	0x100	可创建任何指定类型对象 (对象类型由 obj_type 确定)
库级	ACL_ALT_ANY	0x200	可修改任何指定类型对象 (对象类型由 obj_type 确定)
库级	ACL_DROP_ANY	0x400	可创删除何指定类型对象 (对象类型由 obj_type 确定)
库级	ACL_BACKUP_ANY	0x800	可备份所有对象 (无对象类型)
库级	ACL_RESTORE_ANY	0x1000	可恢复所有对象 (无对象类型)
库级	ACL_VACUUM_ANY	0x2000	可整理空余空间 (无对象类型)
库级	ACL_REPLICATION_ANY	0x4000	可做数据同步 (无对象类型)

接下页

权限级别	权限标识	权限值	说明
库级	ACL_REFRESH_ANY	0x8000	可刷新数据 (无对象类型)
库级	ACL_GRANT_ANY	0x10000	可授权 (无对象类型)
库级	ACL_ENCRYPT_ANY	0x20000	可加密任何表 (无对象类型)
库级	ACL_CRE_JOB	0x40000	可创建定时任务 (无对象类型)
库级	ACL_TRACE	0x80000	可调试任何连接
库级	ACL_AUDITOR	0x10000000	审计设置权 (无对象类型)
库级	ACL_AUDIT_ADMIN	0x30000000	审计管理权 (无对象类型)
库级	ACL_SSO	0x40000000	安全标记权 (无对象类型)
库级	ACL_SS_ADMIN	0xc0000000	安全管理权 (无对象类型)
库级	ACL_DBA	0x7fffff	所有库级权 (除审计权，安全标记权外)
库级	ACL_DBO	0xffffff	DBO 权 (与 DBA 相比，多一个对 DBA 权限用户的收授权)

3.5.1 权限清单

级别	权限选项	说明
库级-库	CREATE ANY DATABASE	创建任何数据库
库级-用户表	CREATE ANY TABLE	创建任何表
	ALTER ANY TABLE	修改任何表
	DROP ANY TABLE	删除任何表
	REFERENCES ANY TABLE	引用任何表
	SELECT ANY TABLE	查询任何表
	INSERT ANY TABLE	可插入任何表
	DELETE ANY TABLE	可删除任何表记录
	UPDATE ANY TABLE	可更新任何表记录
	VACUUM ANY TABLE	可清空任何表
	ENCRYPT ANY TABLE	可加密任何表
	CREATE ANY INDEX	可创建任何索引在表上
	ALL PRIVILEGES	某个表对象的所有操作
库级-用户视图	CREATE ANY VIEW	创建任何视图
	ALTER ANY VIEW	修改任何视图
	DROP ANY VIEW	删除任何视图
	SELECT ANY VIEW	查询任何视图
	INSERT ANY VIEW	插入任何视图
	DELETE ANY VIEW	删除任何视图记录

接下页

级别	权限选项	说明
	UPDATE ANY VIEW	更改任何视图记录
	ALL ON VIEW	某个视图对象的所有操作
库级-索引	CREATE ANY INDEX	创建任何索引
	ALTER ANY INDEX	修改任何索引
	DROP ANY INDEX	删除任何索引
库级-模式	CREATE ANY SCHEMA	创建任何模式
	ALTER ANY SCHEMA	修改任何模式
	DROP ANY SCHEMA	删除任何模式
库级-自定义数据类型	CREATE ANY OBJECT	创建任何自定义数据类型
	ALTER ANY OBJECT	修改任何自定义数据类型
	DROP ANY OBJECT	删除任何自定义数据类型
库级-序列值	SELECT ANY SEQUENCE	查询任何序列值
	CREATE ANY SEQUENCE	创建任何序列值
	ALTER ANY SEQUENCE	修改任何序列值
	DROP ANY SEQUENCE	删除任何序列值
	ALL PRIVILEGES	对某个序列值对象的所有操作
库级-同义词	CREATE ANY SYNONYM	创建任何同义词

接下页

级别	权限选项	说明
	DROP ANY SYNONYM	删除任何同义词
库级-包	CREATE ANY PACKAGE	创建任何包
	DROP ANY PACKAGE	删除任何包
	ALTER ANY PACKAGE	修改任何包
	EXEC ANY PACKAGE	执行任何包
	ALL PRIVILEGES	对某个包对象的所有操作
库级-触发器	CREATE ANY TRIGGER	创建任何触发器
	DROP ANY TRIGGER	删除任何触发器
	ALTER ANY TRIGGER	修改任何触发器
库级-存储过程/函数	CREATE ANY PROCEDURE	创建任何存储过程/函数
	DROP ANY PROCEDURE	删除任何存储过程/函数
	ALTER ANY PROCEDURE	修改任何存储过程/函数
	EXEC ANY PROCEDURE	执行任何存储过程/函数
	ALL PRIVILEGES	对某个存储过程/函数的所有操作
库级-作业	CREATE ANY JOB	创建任何作业
	DROP ANY JOB	删除任何作业

接下页

级别	权限选项	说明
	ALTER ANY JOB	修改任何作业
库级-角色	CREATE ANY ROLE	创建任何角色
库级-用户	CREATE ANY USER	创建任何用户
	ALTER ANY USER	修改任何用户
	DROP ANY USER	删除任何用户
库级-用户权限	DBA	数据库管理员
	SSO	数据库安全员
	AUDIT	数据库审计员
	BACKUP DATABASE	备份数据库
	RESTORE DATABASE	恢复数据库
	BACKUP	备份对象
	RESTORE	恢复对象
库级-DBLINK	CREATE ANY DATABASE LINK	创建任何数据库连接
模式级-表	CREATE ANY TABLE IN SCHEMA	在指定模式下创建任何表
	ALTER ANY TABLE IN SCHEMA	在指定模式下修改任何表
	DROP ANY TABLE IN SCHEMA	在指定模式下删除任何表
	REFERENCES ANY TABLE IN SCHEMA	在指定模式下引用任何表

接下页

级别	权限选项	说明
	SELECT ANY TABLE IN SCHEMA	在指定模式下查询任何表
	DELETE ANY TABLE IN SCHEMA	在指定模式下删除任何表记录
	UPDATE ANY TABLE IN SCHEMA	在指定模式下更新任何表记录
	VACUUM ANY TABLE IN SCHEMA	在指定模式下清空任何表记录
模式级-视图	CREATE ANY ANY VIEW IN SCHEMA	在指定模式下创建任何视图
	DROP ANY ANY VIEW IN SCHEMA	在指定模式下删除任何视图
	SELECT ANY ANY VIEW IN SCHEMA	在指定模式下查询任何视图
	DELETE ANY ANY VIEW IN SCHEMA	在指定模式下删除任何视图记录
	UPDATE ANY ANY VIEW IN SCHEMA	在指定模式下更新任何视图记录
	INSERT ANY ANY VIEW IN SCHEMA	在指定模式下插入任何视图
模式级-序列值	CREATE ANY SEQUENCE IN SCHEMA	在指定模式下创建任何序列值
	ALTER ANY SEQUENCE IN SCHEMA	在指定模式下修改任何序列值

接下页

级别	权限选项	说明
	DROP ANY SEQUENCE IN SCHEMA	在指定模式下删除任何序列值
	SELECT ANY SEQUENCE IN SCHEMA	在指定模式下查询任何序列值
模式级-包	CREATE ANY PACKAGE IN SCHEMA	在指定模式下创建任何包
	ALTER ANY PACKAGE IN SCHEMA	在指定模式下修改任何包
	DROP ANY PACKAGE IN SCHEMA	在指定模式下删除任何包
	EXECUTE ANY PACKAGE IN SCHEMA	在指定模式下执行任何包
模式级-存储过程/函数	CREATE ANY PROCEDURE IN SCHEMA	在指定模式下创建任何存储过程/函数
	ALTER ANY PROCEDURE IN SCHEMA	在指定模式下修改任何存储过程/函数
	DROP ANY PROCEDURE IN SCHEMA	在指定模式下删除任何存储过程/函数
	EXECUTE ANY PROCEDURE IN SCHEMA	在指定模式下执行任何存储过程/函数
模式级-触发器	CREATE ANY TRIGGER IN SCHEMA	在指定模式下创建任何触发器
	ALTER ANY TRIGGER IN SCHEMA	在指定模式下修改任何触发器

接下页

级别	权限选项	说明
	DROP ANY TRIGGER IN SCHEMA	在指定模式下删除任何触发器
模式级-索引	CREATE ANY INDEX IN SCHEMA	在指定模式下创建任何索引
	ALTER ANY INDEX IN SCHEMA	在指定模式下修改任何索引
	DROP ANY INDEX IN SCHEMA	在指定模式下删除任何索引
模式级-同义词	CREATE ANY SYNONYM IN SCHEMA	在指定模式下创建任何同义词
	DROP ANY SYNONYM IN SCHEMA	在指定模式下删除任何同义词
模式级-自定义数据类型	CREATE ANY OBJECT IN SCHEMA	在指定模式下创建任何自定义数据类型
	ALTER ANY OBJECT IN SCHEMA	在指定模式下修改任何自定义数据类型
	DROP ANY OBJECT IN SCHEMA	在指定模式下删除任何自定义数据类型
	ALL ON TABLE	指定表的所有操作权限
	ALTER ON TABLE	指定表的修改权限
	DROP ON TABLE	指定表的删除权限
	INSERT ON TABLE	指定表的插入权限
对象级-表		接下页

级别	权限选项	说明
	UPDATE ON TABLE	指定表的更新数据权限
	DELETE ON TABLE	指定表的删除数据权限
	SELECT ON TABLE	指定表的查询权限
	VACUUM ON TABLE	指定表的清空数据权限
	TRIGGER ON TABLE	指定表上创建触发器权限
	INDEX ON TABLE	指定表上创建索引权限
	REFERENCES ON TABLE	指定表的引用权限
对象级-视图	ALL ON VIEW	指定视图的所有操作权限
	ALTER ON VIEW	指定视图的修改权限
	DROP ON VIEW	指定视图的删除权限
	SELECT ON VIEW	指定视图的查询权限
	INSERT ON VIEW	指定视图的插入权限
	UPDATE ON VIEW	指定视图的更新数据权限
	DELETE ON VIEW	指定视图的删除数据权限

接下页

级别	权限选项	说明
对象级-序列值	ALL ON SEQUENCE	指定序列值的所有操作权限
	SELECT ON SEQUENCE	指定序列值的查询权限
	ALTER ON SEQUENCE	指定序列值的修改权限
	DROP ON SEQUENCE	指定序列值的删除权限
对象级-包	ALL ON PACKAGE	指定包的所有操作权限
	ALTER ON PACKAGE	指定包的修改权限
	DROP ON PACKAGE	指定包的删除权限
	EXECUTE ON PACKAGE	指定包的执行权限
对象级-存储过程/函数	ALL ON PROCEDURE	指定存储过程/函数的所有操作权限
	ALTER ON PROCEDURE	指定存储过程/函数的修改权限
	DROP ON PROCEDURE	指定存储过程/函数的删除权限
	EXECUTE ON PROCEDURE	指定存储过程/函数的执行权限
对象级-自定义数据类型	ALL ON OBJECT	指定自定义数据类型的所有操作权限

接下页

级别	权限选项	说明
	ALTER ON OBJECT	指定自定义数据类型的修改权限
	DROP ON OBJECT	指定自定义数据类型的删除权限
	EXECUTE ON OBJECT	指定自定义数据类型的执行权限
对象级-触发器	ALL ON TRIGGER	指定触发器的所有操作权限
	ALTER ON TRIGGER	指定触发器的修改权限
	DROP ON TRIGGER	指定触发器的删除权限
对象级-列查询	SELECT(COLUMNLIST) ON TABLE_NAME	查询指定表对应列
对象级-列更新	UPDATE(COLUMNLIST) ON TABLE_NAME	更新指定表指定列

4 安全策略

4.1 策略简介

数据库通过安全策略实现强制访问控制。安全策略是一组预定义的标记，根据标记的值来确定主体（用户）对客体（表）是否拥有某种权限，用于实现细粒度的访问控制。

安全策略中，包括等级和范畴两种组件。

安全策略由数据库的安全管理员（SYSSSO）或拥有安全管理员权限的用户管理：

- 创建、更改和删除安全策略。更改安全策略，包括更改安全策略名、给安全策略添加或删除组件（等级和范畴）、更改安全策略中组件的名称或值。
- 给指定的主体或客体添加、更改或删除安全策略。

> **说明**
>
> 数据库最多支持 47 个策略。

安全等级

等级由名称和值两部分组成，其中值为从 0 到 30000 的整数。

等级用于读访问控制时，主体（用户）的等级必须大于客体（表）的等级；用于写访问控制时，主体（用户）的等级必须小于等于客体（表）的等级。如果不满足相应条件，则读或写会被拒绝。

安全范畴

范畴是集合类型，集合中每个元素都是一个名称。不同的范畴之间没有等级高低之分，但可以进行比较（采用集合间的包含关系）。

范畴用于读写访问控制时，主体（用户）的标记必须包含客体（表）的所有范畴。如果不满足相应条件，则读或写会被拒绝。

4.2 策略管理

4.2.1 创建安全策略

语法格式

```
CREATE POLICY policy_name
  [
    ADD LEVEL level_name AS level_number
    [ , ADD CATEGORY category_name [ ADD ... ] ]
  ];
```

参数解释

- policy_name: 安全策略名称。
- level_name: 级别名称。
- level_number: 级别的值，取值为整数。
- category_name: 范畴名称。

示例

- 创建不带等级和范畴的安全策略。

```
    SQL> CREATE POLICY policy_1;
```

- 创建带等级的安全策略。

```
SQL> CREATE POLICY policy_2 ADD LEVEL level_1 AS 1;
```

- 创建带范畴的安全策略。

```
SQL> CREATE POLICY policy_3 ADD CATEGORY category_1;
```

- 创建带等级和范畴的安全策略。

```
SQL> CREATE POLICY policy_4 ADD LEVEL level_1 AS 1,ADD LEVEL
   level_2 AS 2,
ADD CATEGORY category_1,ADD CATEGORY category_2;
```

4.2.2 修改安全策略

语法格式

```
ALTER POLICY policy_name RENAME TO new_policy_name;

ALTER POLICY policy_name ALTER LEVEL level_name RENAME TO
   new_level_name;

ALTER POLICY policy_name ALTER CATEGORY category_name RENAME TO
   new_category_name;

ALTER POLICY policy_name
  ADD LEVEL level_name AS level_number
  [, ADD CATEGORY category_name] ;
```

参数解释

- policy_name: 安全策略名称。
- new_policy_name: 安全策略要修改为的名称。
- level_name: 级别名称。
- new_level_name: 等级要修改为的名称。
- category_name: 范畴名称。
- new_category_name: 范畴要修改为的名称。
- level_number: 级别的值，取值为整数。

示例

- 更改安全策略名。

```
SQL> ALTER POLICY policy_1 RENAME TO policy_001;
```

- 更改等级名。

```
SQL> ALTER POLICY policy_2 ALTER LEVEL level_1 RENAME TO level_2;
```

- 删除范畴。

```
SQL> ALTER POLICY policy_3 DROP CATEGORY category_1;
```

- 增加多个等级的同时删除一个范畴。

```
SQL> ALTER POLICY policy_4 ADD LEVEL level_3 AS 3, ADD LEVEL
   level_4 AS 4,
DROP LEVEL level_1,DROP CATEGORY category_1;
```

4.2.3 删除安全策略

语法格式

```
DROP POLICY policy_name ;
```

参数解释

policy_name: 安全策略名称。

示例

```
SQL> DROP POLICY policy_4;
```

4.2.4 为主体（用户）添加、更改、删除安全策略

语法格式

```
ALTER USER POLICY user_name ADD policy_name LEVEL level_name
   CATETORY category_name;

ALTER USER POLICY user_name DROP policy_name;
```

参数解释

- user_name: 用户名。
- policy_name: 安全策略名称。
- level_name: 级别名称。
- category_name: 范畴名称。

示例

- 为用户添加安全策略、等级、范畴。

```
SQL> ALTER USER POLICY usr_1 ADD policy_4 LEVEL level_3 CATEGORY
   category_2;
```

- 为用户删除安全策略。

```
SQL> ALTER USER POLICY usr_1 DROP policy_4;
```

4.2.5 为客体（表）添加、更改、删除安全策略

语法格式

```
ALTER TABLE POLICY user_name.table_name ADD policy_name
  COLUMN column_name [NOT] HIDE LABEL 'lable_name';

ALTER TABLE POLICY user_name.table_name DROP policy_name;
```

参数解释

- user_name: 用户名。
- table_name: 表名。
- policy_name: 安全策略名称。
- column_name: 要添加安全策略的表的列名；不能是表中已存在的列名。
- lable_name: 标签名，由若干等级和范畴名组成，不同名称间以英文冒号分隔。

示例

- 为表添加安全策略、带等级和范畴的列。

```
SQL> ALTER TABLE POLICY usr_1.tab_test ADD policy_4 COLUMN col_4
   NOT HIDE LABEL 'level_3:category_2';
```

- 为表删除安全策略。

```
SQL> ALTER TABLE POLICY usr_1.tab_test DROP policy_4;
```

4.2.6 综合示例

- SYSDBA 用户登录。

```
SQL> CREATE USER usr_1 IDENTIFIED BY 'QWEasd12#@';

SQL> GRANT DBA TO usr_1;

SQL> CREATE TABLE usr_1.tab_test_1(c1 INT, c2 VARCHAR);

SQL> CREATE TABLE usr_1.tab_test_2(c1 INT, c2 VARCHAR);

SQL> INSERT INTO usr_1.tab_test_1 VALUES(1, 'a');

SQL> INSERT INTO usr_1.tab_test_1 VALUES(2, 'b');

SQL> INSERT INTO usr_1.tab_test_2 VALUES(1, 'alpha');

SQL> INSERT INTO usr_1.tab_test_2 VALUES(2, 'beta');
```

- SYSSSO 用户登录。

```
SQL> CREATE POLICY policy_1 ADD LEVEL level_1 AS 1,ADD LEVEL
   level_2 AS 3,ADD LEVEL level_3 AS  5;

SQL> ALTER USER POLICY usr_1 ADD policy_1 LEVEL level_2;

SQL> ALTER TABLE POLICY usr_1.tab_test_1 ADD policy_1 COLUMN c3
   NOT HIDE LABEL 'level_1:';

SQL> ALTER TABLE POLICY usr_1.tab_test_2 ADD policy_1 COLUMN c3
   NOT HIDE LABEL 'level_3:';
```

- usr_1 用户登录。

```
SQL> UPDATE tab_test_1 SET c2 = 'c' WHERE c1 = 2;

[E18028] 更改操作违反强制安全控制策略

SQL> SELECT * FROM tab_test_1;

C1 | C2 | C3 |
-----------------------------------------------------------------
1 | a| 281474976710656 |
2 | b| 281474976710656 |
```

```
SQL> SELECT * FROM tab_test_2;

C1 | C2 | C3 |
-----------------------------------------------------------------
```

4.3 安全标记

当把安全策略应用于客体（表）时，相应的主客体便获得了相应的安全标记。一个安全标记由等级和范畴名组成。

此外，添加或更改安全标记时，可指定标记是否被隐藏。

语法格式

```
ALTER TABLE POLICY user_name.table_name
    ADD policy_name COLUMN column_name
    [ NOT ] HIDE LABEL 'NameExpr' ;
```

参数解释

- user_name: 用户名。
- table_name: 表名。
- policy_name: 安全策略名称。
- column_name: 表列的名称，不能是表中已存在的列名。
- level_name: 级别名称。
- category_name: 范畴名称。
- NameExpr: 标签的名称，由等级或范畴名组成；如果有多个等级和范畴名，不同名称间用英文冒号分隔。

示例

```
SQL> ALTER TABLE POLICY usr_1.tab_test
     ADD policy_4 COLUMN col_4
     NOT HIDE LABEL 'level_3:category_2';
```

4.4 相关数据字典

跟安全策略相关的系统表有以下三个：SYS_POLICIES、SYS_LEVELS 和 SYS_CATEGORIES，它们分别记载安全策略、等级、范畴相关的信息。

SYS_POLICIES

用于管理数据库安全策略信息。

字段名	字段类型	说明
DB_ID	OID_TYPE	库 ID
POLICY_ID	OID_TYPE	ID
POLICY_NAME	VARCHAR	策略名
COMMENTS	VARCHAR	注释信息
RESERVED1	VARCHAR	保留字段
RESERVED2	VARCHAR	保留字段

SYS_LEVELS

用于管理数据库安全策略对应的安全等级信息。

字段名	字段类型	说明
DB_ID	OID_TYPE	库 ID
PID	OID_TYPE	安全策略 ID
LID	INTEGER	安全级编号
NAME	VARCHAR	安全级别名
COMMENTS	VARCHAR	注释信息
RESERVED1	VARCHAR	保留字段
RESERVED2	VARCHAR	保留字段

SYS_CATEGORIES

用于管理数据库安全策略对应的安全范畴信息。

字段名	字段类型	说明
DB_ID	OID_TYPE	库 ID
PID	OID_TYPE	策略 ID
CID	INTEGER	范围编号
NAME	VARCHAR	范畴名
COMMENTS	VARCHAR	注释信息
RESERVED1	VARCHAR	保留字段
RESERVED2	VARCHAR	保留字段

5 审计管理

审计机制是数据库管理系统安全管理的重要组成部分之一。数据库除了提供数据安全保护措施外，还提供对日常事件的事后审计监督。可以通过它来记录系统级事件、个别用户的行为以及对数据库对象的访问。通过考察、跟踪审计信息，数据库审计员可以查看用户访问的形式以及曾试图对该系统进行的操作，从而采取积极、有效的应对措施。详细信息请参见手册《审计管理指南》。

6 加密

虚谷数据库包含一套完整的加密体系，提供多种加密方式对数据库中表数据、存储文件、传输(SSL)、备份文件等进行加密，确保用户数据的安全性。详细信息请参见《数据加密指南》。

7 资源限制

7.1 系统容错

当主备集群某个节点死亡或者多节点集群只剩下主控节点，则数据库集群会降级。数据库集群降级是为了预防在脑裂情况下造成数据不一致的情况发生，即将集群变为只读来预防。但是数据库节点宕机也有可能产生降级。所以必须了解降级产生的条件和恢复方式。如果是因为节点死亡导致的降级，可以通过命令进行恢复。

产生原因

- 集群节点数等于 2：脑裂或某个节点死亡，副 Master 节点降级，主 Master 不降级。
- 集群节点数大于 2：如果 M 角色的节点只剩下自身一个节点，则降级。例如 5 个节点的集群，由于网络故障，主副 Master 分裂成两个集群，主 Master 和 3 个工作节点形成 1 个 4 节点的集群，此时这个集群不降级；副 Master 只剩自身 1 个节点的集群，此时这个集群要降级。另外，如果是 5 个节点的集群，其中 4 个节点依次死亡，只剩下主 Master 一个节点时，此时这个集群也会降级。

降级后的影响

- 不能新建、删除和 truncate 存储
- M 角色的节点只读（表现和试用版到期一样）
- 集群不再接受节点加入（新增节点或故障节点加入都不接受）

7.2 配额限制

为避免某个用户或某条命令占用大量的系统资源，需要为用户或者命令指定最大配额，配额的指定可以禁止用户的对象使用过多的系统资源，保证数据库的稳定运行。

限制内容	相关配置	默认值	取值范围
临时表空间的最大尺度 (M)	max_temp_space_size	-1	[-1, 2097152]
最大 prepared 语句数	max_prepare_num	100	[100, 2097152]
最大闲置时间 (秒)	max_idle_time	3600	[0, 86400]
禁止登录的失败次数	conn_fail_cnt	3	[2, 100]
用户默认可用连接数	session_per_user	1000	[1, 10000]
最大活动事务连接数	max_act_conn_num	0	[0,1024]
系统最大连接数	max_conn_num	1000	[1, 10000]
最大单任务并行度	max_parallel	1	[1, 600]
存储过程最大循环次数	g_max_loop_num	100000	[100, 2097152]
最大 cursor 数	max_cursor_num	100	[0, 10000]
单个事务最大允许变更行数	max_trans_modify	10000	[0, 1073741824]
允许单个 hash 节点使用的最大内存量 (M)	max_hash_mem	32	[32, 65536]
最大 Hash 表槽数	max_hash_size	3000000	[3000, 2147483647]
最大单次分配内存块大小	max_malloc_once	512	[64, 1024]

接下页

限制内容	相关配置	默认值	取值范围
任务线程运行最大内存	max_task_mem	0	[1, 1048576]
允许记载的最大大对象大小 (M)	max_allow_lob_len	10	[1, 2048]

8 黑白名单

8.1 黑白名单定义

数据库通过配置信任 IP、信任用户和指定数据库来限制用户登录，即黑白名单。其中不允许或不可信的配置项被称为黑名单；允许或可信的项被称为白名单。

黑白名单可通过两种方式进行配置，一种是修改 trust.ini 配置文件，另一种是执行数据库命令。

8.2 黑名单管理

8.2.1 概述

黑名单中记载了被禁止访问数据库的项。每一项记载了指定库的指定用户，不能从指定的 IP 地址（或地址范围）登录。

利用黑名单，可以：

- 限制/取消限制指定用户登录数据库
- 限制/取消限制从指定 IP 地址登录数据库
- 限制/取消限制从指定 IP 段登录数据库
- 限制/取消限制对指定（逻辑）库的登录
- 限制/取消限制指定数据库节点的登录

8.2.2 修改配置文件调整黑名单

语法格式

```
untrust { db_name | everydb } { user_name | everyone } from ip1 [
   to ip2 ]
```

参数解释

- untrust: 该配置项为不可信策略，即不允许访问数据库。
- db_name: 该配置项影响的（逻辑）库名。
- everydb: 该配置项影响所有（逻辑）库。

- user_name: 该配置项影响的用户名。
- everyone: 该配置项影响所有用户。
- ip1: 该配置项影响的 IP 地址。
- ip2: 如果指定了该可选项，该配置项影响从 ip1 到 ip2 的地址段。

> ⚠ 注意
>
> 通过修改配置文件的方式调整黑名单，仅对当前节点有效；且修改完后，需要重启数据库新的修改才生效。此方式需要调整各节点上的 trust.ini 文件，并在调整完后重启数据库。

示例

```
untrust system sysdba from 192.168.1.100
```

示例表示当前节点不允许 sysdba 用户从 192.168.1.100 地址登录 system 库。

8.2.3 执行数据库命令调整黑名单

语法格式

```
ALTER CONNECT POLICY ADD
  DISABLE { user_name | EVERYONE }
  LOGIN { db_name | EVERYDB }
  FROM { ip1 | ANYWHERE } [ TO ip2 ]
  [ ON ALL NODE ] ;
```

参数解释

- DISABLE: 该策略为黑名单策略，即不允许访问数据库。
- user_name: 该策略影响的用户名。
- EVERYONE: 该策略影响所有用户。
- db_name: 该策略影响的（逻辑）库名。
- EVERYDB: 该策略影响所有（逻辑）库。
- ip1: 该策略影响的 IP 地址。
- ANYWHERE: 该策略影响所有 IP 地址。
- ip2: 如果指定了该可选项，该策略影响从 ip1 到 ip2 的地址段。
- ON ALL NODE: 该策略对所有节点生效。

> ⚠ 注意
>
> 通过执行数据库命令的方式调整黑名单，既可仅对当前节点有效，也可对所有节点有效；且命令执行完后立即生效。

示例

```
SQL> ALTER CONNECT POLICY ADD
     DISABLE 'u2' LOGIN 'test'
     FROM '192.168.3.107' TO '192.168.3.110'
     ON ALL NODE;
```

示例表示增加一个黑名单，不允许用户 u2 从 192.168.3.107 至 192.168.3.110 的地址段访问数据库 test。

8.3 白名单

8.3.1 概述

白名单中记载了被允许访问数据库的项。每一项记载了指定库的指定用户，可以从指定的 IP 地址（或地址范围）登录。

利用白名单，可以：

- 允许/取消允许指定用户登录数据库
- 允许/取消允许从指定 IP 地址登录数据库
- 允许/取消允许从指定 IP 段登录数据库
- 允许/取消允许对指定（逻辑）库的登录
- 允许/取消允许指定数据库节点的登录

8.3.2 修改配置文件调整白名单

语法格式

```
trust { db_name | everydb } { user_name | everyone } from ip1 [ to
   ip2 ]
```

参数解释

- trust: 该配置项为可信策略，即允许访问数据库。
- db_name: 该配置项影响的（逻辑）库名。

- everydb: 该配置项影响所有（逻辑）库。
- user_name: 该配置项影响的用户名。
- everyone: 该配置项影响所有用户。
- ip1: 该配置项影响的 IP 地址。
- ip2: 如果指定了该可选项，该配置项影响从 ip1 到 ip2 的地址段。

> ⚠ 注意
>
> 通过修改配置文件的方式调整白名单，仅对当前节点有效；且修改完后，需要重启数据库新的修改才生效。此方式需要调整各节点上的 trust.ini 文件，并在调整完后重启数据库。

示例

```
trust system sysdba from 192.168.1.100
```

示例表示当前节点允许 sysdba 用户从 192.168.1.100 地址登录 system 库。

8.3.3 执行数据库命令调整白名单

语法格式

```
ALTER CONNECT POLICY ADD
  ENABLE { user_name | EVERYONE }
  LOGIN { db_name | EVERYDB }
  FROM { ip1 | ANYWHERE } [ TO ip2 ]
  [ ON ALL NODE ] ;
```

参数解释

- ENABLE: 该策略为白名单策略，即允许访问数据库。
- user_name: 该策略影响的用户名。
- EVERYONE: 该策略影响所有用户。
- db_name: 该策略影响的（逻辑）库名。
- EVERYDB: 该策略影响所有（逻辑）库。
- ip1: 该策略影响的 IP 地址。
- ANYWHERE: 该策略影响所有 IP 地址。
- ip2: 如果指定了该可选项，该策略影响从 ip1 到 ip2 的地址段。

- ON ALL NODE: 该策略对所有节点生效。

> **⚠ 注意**
>
> 通过执行数据库命令的方式调整白名单，既可仅对当前节点有效，也可对所有节点有效；且命令执行完后立即生效。

示例

```
SQL> ALTER CONNECT POLICY ADD
     ENABLE 'u2' LOGIN 'test'
     FROM '192.168.3.107' TO '192.168.3.110'
     ON ALL NODE;
```

示例表示增加一个白名单，允许用户 u2 从 192.168.3.107 至 192.168.3.110 的地址段访问数据库 test。

8.4 相关数据字典

系统表 SYS_BLACK_WHITE_LIST 记载了当前数据库中存在的黑白名单规则。

字段说明

字段名	字段类型	说明
NODEID	INTEGER	节点 ID
LIST_TYPE	CHAR(128)	名单类型
DB_NAME	CHAR(128)	库名
USER_NAME	CHAR(128)	用户名
IP_RANGE	CHAR(128)	IP 范围

9 客体重用

9.1 存储重用

虚谷数据库支持存储重用，相同存储空间在释放后可以重复利用，减少存储浪费。为保证存储的使用安全，存储重用机制如下：

- 存储被分配给数据库对象后，存储被对象独占，其他对象无法使用已分配存储。
- 在数据库对象被删除时，相应的存储空间被释放，并且在被重用前，数据库会清除该地址中的内容，以防止先前的数据被后面的重用者访问。

9.2 缓存重用

虚谷数据库支持内存重用，即内存释放后可以重新分配使用。为了确保内存重用的安全性，在对内存进行重用前，会对该地址中的内容进行清除，以防止先前的数据被泄露。

10 会话管理

10.1 概述

数据库会话（Session）是通信双方（客户端与服务端）从开始通信到结束通信期间的一个上下文（Context）。此上下文位于服务端（数据库）的内存：记录了本次连接的所有相关状态和运行数据。数据库会话管理是针对数据库会话连接属性、数据库会话状态、数据库会话传输安全等开展的一系列会话控制行为。

10.2 会话建立

当前数据库支持 TCP/IP 方式与 SSL 方式的数据库会话，SSL 会话是在 TCP/IP 连接基础上，进行传输通道加密，保证传输安全。数据库会话的建立应满足数据库黑白名单的可信规则，否则无法建立有效的数据库会话，黑白名单可信规则参照第8章黑白名单。数据库会话建立方式见下表。

序号	连接方式	默认是否开启	支持系统	仅支持本机	开启方式	参数配置
1	TCP/IP	是	所有系统	否	默认开启	访问端口：listen_port 绑定地址：-bind -address
2	SSL	是	所有系统（基于 TCP/IP）	否	-ssl=ssl/nssl	开启/关闭：-ssl=* options

10.3 会话查询

数据库管理员 (SYSDBA) 具备数据库连接会话查询和管理的能力，通过系统表能够查看到当前所有的会话信息，管理员可以分析会话的状态来管理会话。数据库会话包含会话属性 (会话变

量) 和全局变量。

- 会话变量：当客户端连接到数据库后，数据库会复制全局变量以自动生成会话变量。会话变量的修改只对当前会话生效。
- 全局变量：数据库实例共享全局变量。这意味着不同用户共享这些全局变量，且数据库会保存对全局变量做出的更改，断开连接并再次进入数据库时，更改依旧有效。

数据库会话的连接信息查询 (系统表)：

会话系统表	作用
DBA_SESSIONS	查看连接会话所在节点的所有数据库连接会话
SYS_ALL_SESSIONS	查看数据库系统内所有数据库连接会话
SYS_ALL_THD_SESSION	用于管理数据库当前任务线程信息
SYS_ALL_THD_STATUS	用于管理数据库当前节点线程状态信息

数据库会话的会话变量查询可通过 SHOW 命令进行查看，具体的会话变量及查看命令请参见《SQL 语法参考指南》的会话变量章节。

数据库会话的全局变量可通过 SYS_VARS 命令进行查看。

执行命令

查看数据库会话全局变量。

```
SQL> SHOW SYS_VARS;
VAR_NAME | IS_GLOBAL | ACCESS | DESCRI |
-----------------------------------------------------------
listen_port| T | R/W| 侦听端口|
nio_timeout| T | R/W| 网络读写超时(单位:秒)|
login_timeout| T | R/W| 登录超时时间(单位:秒)|
use_std_nio| T | R/W| 是否使用标准网络监听器(若为假,则使用依赖于操
    作系统的效率更高的网络侦听器)|
max_idle_time| T | R/W| 最大闲置时间(单位:秒,在此时间内若无请求,则
    断开连接)|
min_pass_len| T | R/W| 口令的最短长度|
pass_mode| T | R/W| 口令模式 1:任意字符或数字 2:必须含字母和数字 3:
    必须含字母和数字和特殊符号|
def_timezone| T | R/W| 默认客户端时区|
def_timefmt| T | R/W| 默认客户端时间格式|
def_charset| T | R/W| 默认客户端字符集|
send_warning| T | R/W| 是否发送警告信息|
conn_fail_cnt| T | R/W| 禁止登陆的失败次数|
max_act_conn_num| T | R/W| 最大活动事务连接数|
max_conn_num| T | R/W| 系统最大连接数|
session_per_user| T | R/W| 用户默认可用连接数|
```

```
. . . . . .
```

参数解释

- VAR_NAME：全局变量名称。
- IS_GLOBAL：是否数据库全局变量，T：是，F：否。
- ACCESS：是否允许修改，R：只读，R/W：读写。
- DESCRI：全局变量描述。

10.4 管理策略

数据库会话会消耗数据库相应的资源，对于会话的申请、使用、释放，都需要相应的管理策略，以便有效的管理数据库会话资源。数据库会话的管理包括配置参数管理、会话变量的管理以及相应的资源清除策略，可灵活管理数据库会话资源，实现数据库会话的安全使用。数据库会话配置参数见下表。

配置内容	配置参数	默认值	限定范围	配置说明
侦听端口	listen_port	5138	操作系统未占用的 TCP 端口	用于控制数据库会话端口，结合操作系统防火墙进行安全控制
最大闲置时间(单位：秒)	max_idle_time	3600	[0, 86400]	用于控制会话连接空闲时长，超过空闲时长时，连接断开

接下页

配置内容	配置参数	默认值	限定范围	配置说明
会话建立最大失败次数	conn_fail_cnt	3	[2, 100]	建立数据库会话，密钥验证最大失败次数，超过配置阈值时，登录账户锁定 5 分钟
用户默认可用连接数	session_per_user	1000	[1, 10000]	单一用户默认可以创建的数据库会话连接最大值
系统最大连接数	max_conn_num	1000	[1, 10000]	数据库内整个系统默认可以创建的数据库会话连接最大值
最大活动事务连接数	max_act_conn_num	0	[0, 1024]	数据库内同时允许的最大活动事务连接数，用于限制活动事务量，防止资源被占满

查看/设置数据库会话变量命令

- 查看数据库连接会话属性命令。

```
SQL> SHOW { 会 话 变 量 };
```

- 修改数据库连接会话属性命令。

```
SQL> SET { 会 话 变 量 } TO { 值 };
```

数据库会话操作管理依赖系统管理员操作包 DBMS_DBA，此系统包主要用于辅助管理员进行系统维护操作，如：数据库会话、数据库事务的维护。通过管理员操作包可以针对一些导致系

统资源消耗较大的操作、存在安全风险的 SQL 操作行为进行系统维护或终止。DBMS_DBA 系统包的介绍与使用请参见《系统包参考》的DBMS_DBA章节。

接口/方法	参数名称	参数类型	参数解释	接口功能
KILL_TRANS	•NODEID •TRANID	INTEGER	• 事务所在的节点编号 • 全局唯一的事务号	终止事务的运行
KILL_SESSION	•NODEID •SESSID	INTEGER	• 会话所在的节点编号 • 节点内唯一的数据库连接会话号	终止数据库连接会话
KILL_SESSION_TRANS	•NODEID •SESSID	INTEGER	• 会话所在的节点编号 • 节点内唯一的数据库连接会话号	终止会话上的事务

⚠ 注意

终止数据库会话会导致会话上正在执行的所有操作都一并中断，属于较为重的操作。一般情况建议用户尽量采用小而轻的维护操作，仅针对单一事务或 SQL 操作进行维护。

示例

- 查看当前数据库连接会话时区。

```
SQL> SHOW DEF_TIMEZONE;
DEF_TIMEZONE |
-------------------------------------------------------------
GMT+08:00|
```

- 查看当前数据库连接会话自动提交属性。

```
SQL> SHOW AUTO_COMMIT
```

```
AUTO_COMMIT |
------------------------------------------------------------
T |
```

- 设置当前数据库连接会话默认时区属性为 GMT+00:00。

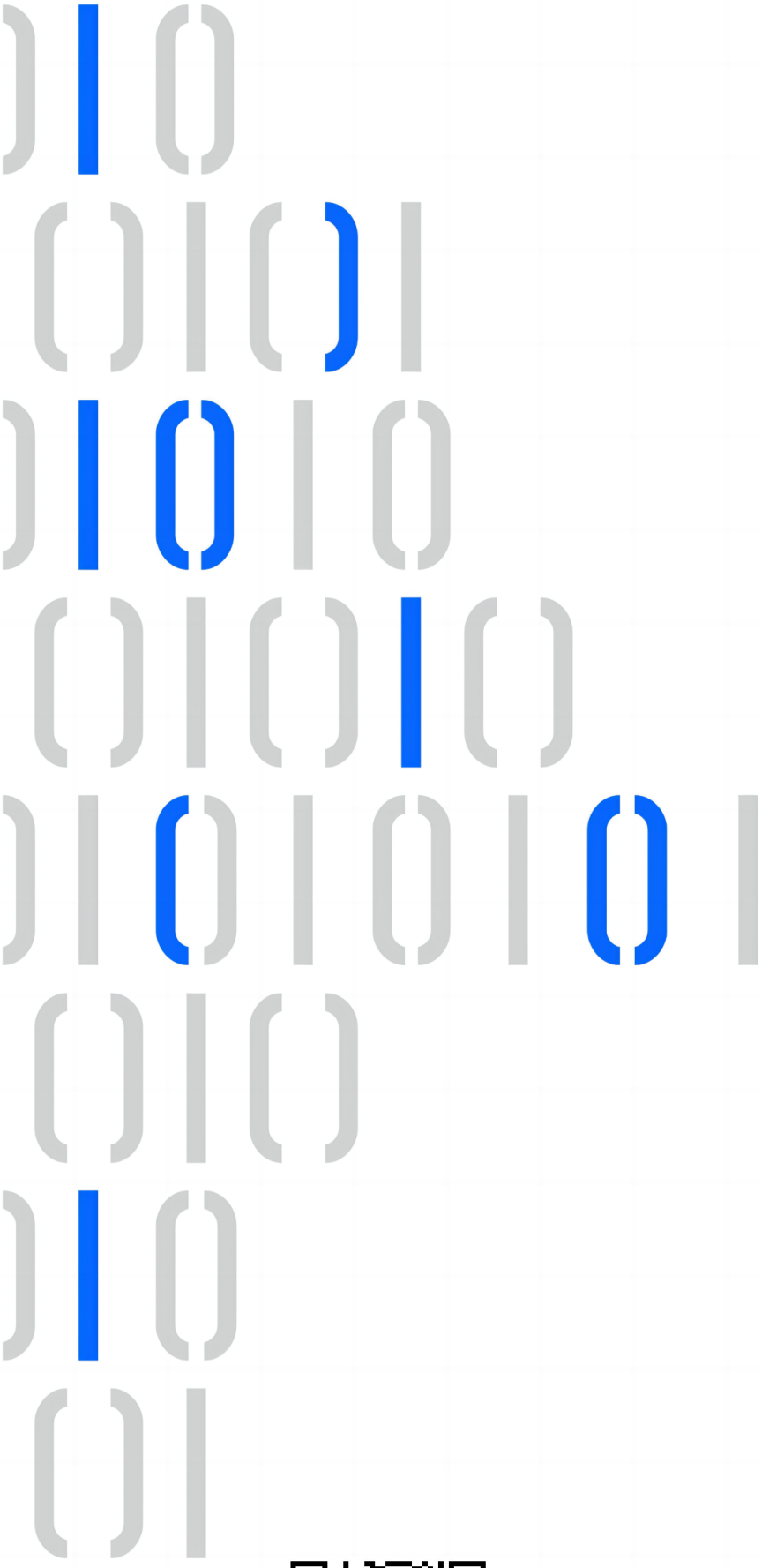
```
SQL> SET DEF_TIMEZONE TO 'GMT+00:00';

Execute successful.Use time:1 ms.

SQL> SHOW DEF_TIMEZONE

DEF_TIMEZONE |
------------------------------------------------------------
GMT+00:00|
```

- 查看数据库连接会话信息。

```
SQL> SELECT nodeid,session_id,ip,user_name,db_name,start_t,status
   ,curr_tid,auto_commit,trans_start_t FROM sys_sessions;

NODEID | SESSION_ID | IP | USER_NAME | DB_NAME | START_T | STATUS
    | CURR_TID | AUTO_COMMIT | TRANS_START_T |
------------------------------------------------------------------
1 | 230493 | 127.0.0.1| SYSDBA| SYSTEM| 2015-05-13 17:26:52.000
   AD | 114 | 717722 | T | 2022-05-13 17:46:23.000 AD |
1 | 230494 | 127.0.0.1| SYSDBA| SYSTEM| 2015-05-13 17:44:29.000
   AD | 112 | 717719 | F | 2022-05-13 17:45:35.000 AD |
1 | 230495 | 127.0.0.1| SYSDBA| SYSTEM| 2015-05-13 17:44:47.000
   AD | 112 | 717720 | F | 2022-05-13 17:45:46.000 AD |

Total 3 records.

Use time:0 ms.
```

- 使用 DBMS_DBA.KILL_TRANS 接口终止数据库连接会话正在执行的事务。

```
SQL> EXEC DBMS_DBA.KILL_TRANS(1,717720);
```



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