优炫数据库安全功能手册 2.1

UXSINO
优 炫 软 件

优炫数据库安全功能手册 2.1

法律声明

优炫数据库管理系统(简称：UXDB) 是由北京优炫软件股份有限公司开发并发布的一款商业性数据库管理系统。

注意

由于产品版本安装或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

北京优炫软件股份有限公司（总部）

- 地址：北京市海淀区学院南路62号中关村资本大厦11层（邮编：100081）
- 网址：http://www.uxsino.com
- 邮箱：<uxdb_support@uxsino.com>
- 电话：010-82886998
- 传真：010-82886338
- 服务热线：400-650-7837

目录

表格清单

前言

1. 文档目的

本文档主要介绍了UXDB数据库的安全特性。

2. 文档对象

- 维护工程师
- 技术支持工程师

3. 修改记录

修改记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

表 1. 文档更新记录

工具版本	发布日期	修改说明
2.1.1.5C	2023-01-10	第一次正式发布。

第 1 章 概述

1.1. 背景

数据库安全包含两层含义：第一层是指系统运行安全，系统运行安全通常受到的威胁如下，一些网络不法分子通过网络，局域网等途径通过入侵电脑使系统无法正常启动，或超负荷让机子运行大量算法，并关闭cpu风扇，使cpu过热烧坏等破坏性活动； 第二层是指系统信息安全，系统信息安全通常受到的威胁如下，黑客对数据库入侵，并盗取想要的资料。数据库系统的安全特性主要是针对数据而言的，包括数据独立性、数据安全性、数据完整性、并发控制、故障恢复等几个方面。

据Verizon2012年的数据泄露调查分析报告和对发生的信息安全事件技术分析，总结出信息泄露呈现两个趋势，如下所示。

1. 黑客通过B/S应用，以Web服务器为跳板，窃取数据库中数据；传统解决方案对应用访问和数据库访问协议没有任何控制能力，比如:SQL注入就是一个典型的数据库黑客攻击手段。

2. 数据泄露常常发生在内部，大量的运维人员直接接触敏感数据，传统以防外为主的网络安全解决方案失去了用武之地。

数据库在这些泄露事件成为了主角，这与我们在传统的安全建设中忽略了数据库安全问题有关，在传统的信息安全防护体系中数据库处于被保护的核心位置，不易被外部黑客攻击，同时数据库自身已经具备强大安全措施，表面上看足够安全，但这种传统安全防御的思路，存在致命的缺陷。

1.2. 数据库安全

数据库安全（DataBase Security）是指采取各种安全措施对数据库及其相关文件和数据进行保护。数据库系统的重要指标之一是确保系统安全，以各种防范措施防止非授权使用数据库。数据库系统中一般采用用户标识和鉴别、存取控制、视图以及密码存储等技术进行安全控制。

数据库安全的核心和关键是其数据安全。数据安全是指以保护措施确保数据的完整性、保密性、可用性、可控性和可审查性。由于数据库存储着大量的重要信息和机密数据，而且在数据库系统中大量数据集中存放，供多用户共享，因此，必须加强对数据库访问的控制和数据安全防护。

从系统与数据的关系上，也可将数据库安全分为数据库的系统安全和数据安全。

数据库系统安全主要利用在系统级控制数据库的存取和使用的机制，如下所示。

1. 系统的安全设置及管理，包括法律法规、政策制度、实体安全等。

2. 数据库的访问控制和权限管理。

3. 用户的资源限制，包括访问、使用、存取、维护与管理等。

4. 系统运行安全及用户可执行的系统操作。

5. 数据库审计有效性。

6. 用户对象可用的磁盘空间及数量。

数据安全是在对象级控制数据库的访问、存取、加密、使用、应急处理和审计等机制，包括用户可存取指定的模式对象及在对象上允许作具体操作类型等。

1.3. UXDB数据库

UXDB是新一代数据库管理系统，具备支持多种数据类型、高可用、高性能、高安全、数据即服务等核心能力，在不中断业务情况下，实现大规模并行处理、超级真正应用集群（Super RAC）与云数据库多种模式部署，应用于高频联机系统、地理信息、数据仓库、商业智能等多业务场景；产品支持在众多芯片、操作系统、中间件、应用软件的稳健运行，满足我国政府、军工、金融、能源、制造、医疗等各行业产业升级需求。

1.4. UXDB安全功能

UXDB数据库安全功能通过对登录进数据库系统的用户进行更为有效的身份鉴别， 权限限制，用户数据本地加密等措施来保证用户数据在访问、存储过程中的有效保护。

UXDB数据库的安全功能如下所示。

- 三权分立
- 审计

1.5. UXDB安全功能开关

安全功能开关用于控制安全功能的开启和关闭。安全功能开关作用于数据库实例初始化时，UXDB的安全功能是以插件化的形式实现的，所以当安全功能开关开启时，由于三权分立的策略影响，会改变UXDB的默认用户数量，带有安全功能的实例初始化后不可逆。当用户需要启用安全功能时，可以可以利用初始化工具initdb通过参数指定打开开关，具体使用方法和案例如下所示。

全库加密不能被安全功能开关控制，原因是这是一个普通集群也可以使用的功能，所以如果需要使用全库加密集群，需要在初始化时单独增加-M参数。目前在初始化时可以打开的开关包含模式开关和安全功能开关，分别使用参数--running mode和--security这两个开关相互没有影响。可以同时指定。

在使用initdb初始化数据库集群时，指定--security参数意为打开安全功能开关，此开关会记录在控制文件中，并且在初始化后不能够修改。在整个数据库生命周期内，提供安全功能，本阶段主要包括三权分立和安全审计功能。

结合安全功能开关和目前数据库集簇的几种模式的使用，数据库可以被初始化时指定为标准、标准安全、兼容、兼容安全，且每种模式下都可以选择是否开启全库加密。

开启安全功能的数据库实例，如下所示。

1. 以标准集群或者兼容集群为例。
   - 初始化标准集群且打开安全功能的数据库实例。

     ./initdb -W -D test_sec --security

   - 初始化兼容集群且打开安全功能的数据库实例

     ./initdb -W -D test_sec --running-mode=compatible --security

2. 启动此集群。

./ux_ctl -D test_sec start

3. 使用uxsql连接。

./uxsql -d uxdb -U uxsmo

> **注意**
>
> 连接时需要指定用户，三权分立中默认初始化有uxsmo、uxsso、uxsao三个用户。

开启安全功能并进行全库加密，如下所示。

1. 以标准集群或者兼容集群为例。
   - 初始化标准集群并进行全库加密。

     ./initdb -W -D test_sec --security -M

   - 初始化兼容集群并进行全库加密。

     ./initdb -W -D test_sec --running-mode=compatible --security -M

2. 启动此集群。

   ./ux_ctl -M -D test_sec start

3. 使用uxsql连接。

   ./uxsql -d uxdb -U uxsmo

UXDB提供系统函数cluster_option()返回数据库初始化时的模式(兼容/标准模式-是否安全)。

查看初始化数据库模式，如下所示。

select cluster_option();

```
UXDB=> select cluster_option();
   CLUSTER_OPTION
----------------------
 compatible-security
(1 row)

UXDB=>
```

若启用了全库加密功能，则系统表数据，索引表数据，用户自定义表数据都是加密的，Control文件全库加密开关也是打开的(其他模式一样)。

1. 查询表文件名称。

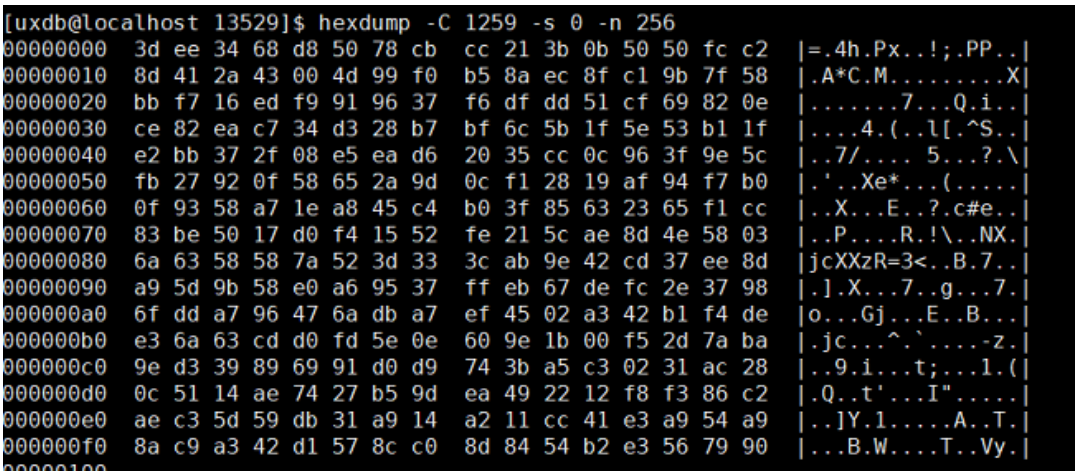
   select ux_relation_filepath('ux_class');
   select ux_relation_filepath('test');

2. 打开二进制表文件。

   - 方法一：使用hexedit命令打开文件。
   - 方法二：使用vim,然后输入:%!xxd命令转换为十六进制。
   - 方法三：hexdump命令。

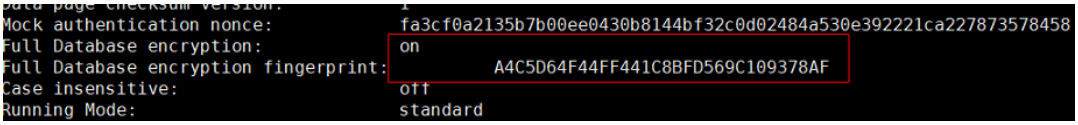
3. 验证。

   验证系统表ux_class为加密状态。



   验证control文件中全库加密为开启状态。

   ./ux_controldata -D test_sec



第 2 章 三权分立

2.1. 概述

用户是一个数据对象，是一系列数据库对象和权限的统称。用户所有的操作默认在可使用的模式下进行，模式是一个用户所拥有的数据库对象的集合，每个用户都有自己的默认模式，管理用户默认模式的名字与用户名相同，普通用户默认模式为public。UXDB数据库为了防止权限滥用问题，引入了三权分立的安全机制。默认提供了系统管理员（uxsmo）、安全管理员（uxsso）、审计管理员（uxsao），将DBA的权限拆解出来，按最小授权原则分别授予它们各自为完成自己承担任务所需的最小权限，并形成相互制约的关系。

- 系统管理员（uxsmo）

  系统管理员主要负责生成用户标识符和系统运行维护。

- 安全管理员（uxsso）

  安全管理员主要负责安全策略和标记的生成和维护。

- 审计管理员（uxsao）

  审计管理员主要负责审计相关参数配置，以及对系统审计日志进行审查分析。

注意

1. 管理员不能被删除，默认存在和三个管理员同名的模式，且不能被删除，管理员名和模式不能重命名（管理员创建的对象默认在私有模式下）。
2. 管理员不能是表、视图、列、序列、大对象GRANT语句的接收者，不能增加权限。
3. 管理员默认只能执行允许的GRANT语句，可以GRANT系统管理员对象的使用权，可以GRANT用户对象的操作权。
4. 其它角色不能是管理员的成员，管理员不能是其它角色的成员。
5. 授予权限时不能使用WITH GRANT OPTION等选项。
6. 任何角色不能使用DROP OWNER命令，但是可以使用命令REASSIGN OWNER/ALTER TABLE/... name OWNER TO new_owner，进行属主转移，但是属主转移只能发生在普通用户之间。
7. 所有管理员创建的数据库对象不能转OWNER。

2.2. 创建用户

数据库系统在运行的过程中，往往需要根据实际需求创建用户，然后为用户指定适当的权限。创建用户的操作只能由系统管理员uxsmo完成。

创建用户的命令是CREATE USER，具体用法参见《优炫数据库参考手册 V2.1》。其中用户名是代表用户账号的标识符，它的命名规则如下所示。

1. 必须以字符开始。

2. 从第二个字母开始，可以包括大小写字母、数字、_、$等字符。

2.3. 删除用户

一个用户不再访问数据库系统时，应该将这个用户及时地从数据库中删除，否则可能会有安全隐患。

删除用户的操作只能由uxsmo进行。删除用户的命令是drop user，语法格式如下所示。

DROP USER 用户名;

需要注意的是，如果一个用户的模式中已经包含一些数据库对象，那么这个用户是不能被直接删除的，在删除用户时系统将给出如下的错误信息。

role "u1" cannot be dropped because some objects depend on it

2.4. 权限管理

用户权限有两类，即数据库操作权限和数据库对象权限。数据库操作权限主要是指针对数据库对象的创建、删除、修改的权限，对数据库备份等权限。而对象权限主要是指对数据库对象中的数据的访问权限。数据库操作权限一般由uxsmo、uxsso、uxsao指定。对象权限一般由数据库对象的所有者授予用户。

2.4.1. 数据库权限的管理

数据库权限是与数据库安全有关的最重要的权限，这类权限一般是针对数据库管理员的。数据库权限的管理主要包括权限的分配、回收和查询等操作。以下列出与用户有关的最重要的几种数据库权限。

表 2.1. 数据库权限

数据库对象	权限
数据库	管理员可以创建并管理，其他用户只能使用
表空间	管理员可以创建并管理，其他用户只能使用
模式	管理员可以创建并管理，其他用户只能使用
表	在有使用权限的模式下创建、修改、删除、分配、使用
索引	在有使用权限的表下创建、修改、删除、使用
函数/存储过程	在有使用权限的模式下创建、修改、删除、分配、使用
触发器	在有权限的对象上创建、修改、删除、使用
用户	只有系统管理员可以创建并管理用户

对于表、视图、用户、触发器这些数据库对象，数据库权限包括创建、删除和修改他们的权限，相关的命令分别是CREATE、DROP和 ALTER。表、视图、触发器、存储过程等对象是与用户有关的，在默认情况下对这些对象的操作都是在当前用户自己的模式下进行的。要能够在其他用户的

模式下创建表，当前用户必须具有该模式的使用权限，模式的使用权限由管理员授权分配，语句如下所示。

GRANT 对象权限 1(列名),对象权限 2(列名).... ON 对象类型 对象名称 TO 用户 1，用户 2....

GRANT命令执行后，所有指定用户都将获得指定的权限。

2.4.2. 对象权限的管理

对象权限主要是对数据库对象中的数据的访问权限，这类权限主要是针对普通用户的。SELECT、INSERT、DELETE和UPDATE权限分别是针对数据库对象中的数据的查询、插入、删除和修改的权限。对于表和视图来说，删除操作是整行进行的，而查询、插入和修改却可以在一行的某个列上进行，所以在指定权限时，DELETE权限只要指定所要访问的表就可以了，而SELECT、INSERT和UPDATE权限还可以进一步指定是对哪个列的权限。EXECUTE权限是指可以执行存储函数、存储过程的权限。有了这个权限，一个用户就可以执行另一个用户的存储过程。

当一个用户获得另一个用户的某个对象的访问权限后，以“模式名.对象名”的形式访问这个数据库对象。一个用户所拥有的对象和可以访问的对象是不同的，这一点在数据字典视图中有所反映。在默认情况下用户可以直接访问自己创建的数据库对象，但是要访问其他用户所拥有的对象，就必须具有相应的对象权限。

对象权限的授予只能由对象的所有者完成。授予对象权限的命令是GRANT，回收权限的命令是REVOKE，与数据库模式授权的操作类似。

2.5. 安全功能限制说明

1. 系统初始化构建的默认的系统模式例如ux_catalog模式无法进行grant授权。
2. 兼容模式下，开启安全功能后使用工具createdb创建数据库时，需要指定模式名，否则会报错。

   ./createdb -e -U UXSMO uxdb0002 --running-mode=compatible

3. 在启用了安全功能的集群中，建议最多允许连接数不小于6，因为UXDB会预留默认连接给默认用户和审计进程，所以当设置最多允许同时连接5个客户端时，此时使用用户进行连接尝试，会发现相同用户的第二个会话会连接失败。

第 3 章 安全审计

UXDB安全数据有独立的安全审计系统，设置专门的安全审计员，定义与数据库安全相关的审计事件。主要涵盖了审计日志可读、操作记录和查询、审计策略设置以及日志安全。安全审计是和安全功能开关绑定的，当打开安全功能后，审计进程默认启动，可以通过参数配置修改审计范围。审计会记录系统级事件、用户行为、数据库会话连接、数据库配置参数修改、数据库对象访问等行为，通过查看审计信息，安全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。

3.1. 审计开关

UXDB数据库安全审计功能由审计开关（enable_audit）控制，该开关为GUC参数，有两种取值，如下所示。

- 打开审计：true（默认）、on、1。
- 关闭审计：false、off、0。

设置方式如下所示。

- 方式一：修改uxsinodb.conf文件中enable_audit = on/off。
- 方式二：执行alter syetem set enable_audit = off / on语句。

设置成功后通过以下方式使修改生效。

1. 重启实例，如：ux_ctl -D sec restart。
2. 加载配置文件，如：使用ux_ctl命令(ux_ctl -D sec reload) 或 使用sql语句(select ux_reload_conf();)。

> **注意**
>
> 1. 加载配置文件生效，只在实例启动时uxsinodb.conf中enable_audit = on情况下有效。
> 2. 如果实例启动时uxsinodb.conf中enable_audit = off，设置审计开关的值后，只有重启实例才可以生效。

3.2. 审计的设置与取消

数据库审计员指定被审计对象的活动称为审计设置，只有具有 AUDIT DATABASE 权限的审计员才能进行审计设置。UXDB提供审计设置系统过程来实现这种设置，被审计的对象可以是某类操作，也可以是某些用户在数据库中的全部行踪。只有预先设置的操作和用户才能被UXDB系统自动进行审计。

UXDB允许在三个级别上进行审计设置，如下表所示。

表 3.1. 审计级别说明

审计级别	说明
系统级	系统的启动与关闭，此级别的审计无法也无需由用户进行设置，只要审计开关打开就会自动生成对应审计记录。
语句级	导致影响特定类型数据库对象的特殊 SQL 或语句组的审计。如AUDIT TABLE将审计CREATE TABLE、ALTER TABLE和DROP TABLE等语句。
对象级	审计作用在特殊对象上的语句。如test表上的INSERT语句。

审计设置存放于ux_audit_stmt或ux_audit_object系统表中，进行一次审计设置就在ux_audit_stmt或ux_audit_object系统表中增加或修改一条对应的记录，取消审计则删除ux_audit_stmt或ux_audit_object系统表中相应的记录。

3.2.1. 语句级审计

语句级审计的动作是全局的，不对应具体的数据库对象。其审计选项如下表。

表 3.2. 语句级审计说明

审计选项	被审计操作	说明
ALL	所有的语句级审计选项	所有可审计操作
AGGREGATE	CREATE AGGREGATE	创建/修改/删除聚集函数
	ALTER AGGREGATE	
	DROP AGGREGATE	
COLLATION	CREATE COLLATION	创建/修改/删除排序规则
	ALTER COLLATION	
	DROP COLLATION	
CONVERSION	CREATE CONVERSION	创建/修改/删除编码转换
	ALTER CONVERSION	
	DROP CONVERSION	
DATABASE	CREATE DATABASE	创建/修改/删除数据库操作
	ALTER DATABASE	
	DROP DATABASE	
DOMAIN	CREATE DOMAIN	创建/修改/删除域
	ALTER DOMAIN	
	DROP DOMAIN	
EVENT TRIGGER	CREATE EVENT TRIGGER	创建/修改/删除时间触发器
	ALTER EVENT TRIGGER	
	DROP EVENT TRIGGER	
EXTENSION	CREATE EXTENSION	创建/修改/删除插件
	ALTER EXTENSION	
	DROP EXTENSION	

审计选项	被审计操作	说明
FOREIGN DATA WRAPPER	CREATE FOREIGN DATA WRAPPER	创建/修改/删除外部数据包装器
	DROP FOREIGN DATA WRAPPER	
	ALTER FOREIGN DATA WRAPPER	
FOREIGN TABLE	CREATE FOREIGN TABLE	创建/修改/删除外部表/从一个外部服务器导入表定义
	IMPORT FOREIGN SCHEMA	
	ALTER FOREIGN TABLE	
	DROP FOREIGN TABLE	
FUNCTION	CREATE FUNCTION	创建/修改/删除函数
	ALTER FUNCTION	
	DROP FUNCTION	
INDEX	CREATE INDEX	创建/修改 / 删除/重建索引操作
	ALTER INDEX	
	DROP INDEX	
	REINDEX	
LANGUAGE	CREATE LANGUAGE	创建/修改/删除语言
	ALTER LANGUAGE	
	DROP LANGUAGE	
MATERIALIZED VIEW	CREATE MATERIALIZED VIEW	创建/修改/删除/替换物化视图
	ALTER MATERIALIZED VIEW	
	DROP MATERIALIZED VIEW	
	REFRESH MATERIALIZED VIEW	
OPERATOR	CREATE OPERATOR	创建/修改/删除操作符
	ALTER OPERATOR	
	DROP OPERATOR	
OPERATOR CLASS	CREATE OPERATOR CLASS	创建/修改/删除操作符类
	ALTER OPERATOR CLASS	
	DROP OPERATOR CLASS	
OPERATOR FAMILY	CREATE OPERATOR FAMILY	创建/修改/删除族
	ALTER OPERATOR FAMILY	
	DROP OPERATOR FAMILY	
POLICY	CREATE POLICY	创建/修改/删除行级安全策略
	ALTER POLICY	
	DROP POLICY	
PROCEDURE	CREATE PROCEDURE	创建/修改/删除/调用存储模块操作
	ALTER PROCEDURE	
	DROP PROCEDURE	
	CALL	
PUBLICATION	CREATE PUBLICATION	创建/修改/删除发布

审计选项	被审计操作	说明
	ALTER PUBLICATION	
	DROP PUBLICATION	
ROLE	CREATE ROLE	创建/修改/删除角色/用户/组操作
	ALTER ROLE	
	DROP ROLE	
	CREATE USER	
	ALTER USER	
	DROP USER	
	CREATE GROUP	
	ALTER GROUP	
	DROP GROUP	
RULE	CREATE RULE	创建/修改/删除规则
	ALTER RULE	
	DROP RULE	
SCHEMA	CREATE SCHEMA	创建/删除/设置当前模式操作
	ALTER SCHEMA	
	DROP SCHEMA	
SEQUENCE	CREATE SEQUENCE	创建/修改/删除序列操作
	ALTER SEQUENCE	
	DROP SEQUENCE	
SERVER	CREATE SERVER	创建/修改/删除外部服务器
	ALTER SERVER	
	DROP SERVER	
STATISTICS	CREATE STATISTICS	创建/修改/删除扩展统计
	ALTER STATISTICS	
	DROP STATISTICS	
SUBSCRIPTION	CREATE SUBSCRIPTION	创建/修改/删除订阅
	ALTER SUBSCRIPTION	
	DROP SUBSCRIPTION	
TABLE	CREATE TABLE	创建/修改/删除/锁定/清空基表/从一个查询的结果创建一个新表/从一个查询的结果定义一个新表
	ALTER TABLE	
	DROP TABLE	
	LOCK	
	TRUNCATE	
	CREATE TABLE AS	
	SELECT INTO	
TABLESPACE	CREATE TABLESPACE	创建/修改/删除表空间操作
	ALTER TABLESPACE	

审计选项	被审计操作	说明
	DROP TABLESPACE	
TEXT SEARCH CONFIGURATION	CREATE TEXT SEARCH CONFIGURATION	创建/修改/删除文本搜索配置
	ALTER TEXT SEARCH CONFIGURATION	
	DROP TEXT SEARCH CONFIGURATION	
TEXT SEARCH DICTIONARY	CREATE TEXT SEARCH DICTIONARY	创建/修改/删除文本搜索字典
	ALTER TEXT SEARCH DICTIONARY	
	DROP TEXT SEARCH DICTIONARY	
TEXT SEARCH PARSER	CREATE TEXT SEARCH PARSER	创建/修改/删除文本搜索解析器
	ALTER TEXT SEARCH PARSER	
	DROP TEXT SEARCH PARSER	
TEXT SEARCH TEMPLATE	CREATE TEXT SEARCH TEMPLATE	创建/修改/删除文本搜索模板
	ALTER TEXT SEARCH TEMPLATE	
	DROP TEXT SEARCH TEMPLATE	
TRIGGER	CREATE TRIGGER	创建/修改/删除触发器操作
	ALTER TRIGGER	
	DROP TRIGGER	
TYPE	CREATE TYPE	创建/修改/删除类型
	ALTER TYPE	
	DROP TYPE	
USER MAPPING	CREATE USER MAPPING	创建/修改/删除用户映射器
	ALTER USER MAPPING	
	DROP USER MAPPING	
VIEW	CREATE VIEW	创建/修改/删除视图操作
	ALTER VIEW	
	DROP VIEW	
ACCESS METHOD	CREATE ACCESS METHOD	创建/删除访问方法
	DROP ACCESS METHOD	
CAST	CREATE CAST	创建/删除造型
	DROP CAST	
TRANSFORM	CREATE TRANSFORM	创建/删除转换
	DROP TRANSFORM	
SYNONYM	CREATE SYNONYM	创建/删除同义词
	DROP SYNONYM	
ROUTINE	ALTER ROUTINE	创建/删除例程
	DROP ROUTINE	

审计选项	被审计操作	说明
PORTAL	DECLARE	定义一个游标
	FETCH	使用游标从查询中检索行
	MOVE (T_FetchStmt)	定位一个游标
	CLOSE	关闭一个游标
SENTENCE	PREPARE	为执行准备一个语句
	EXECUTE	执行一个预备语句
	DEALLOCATE	释放一个预备语句
TRANS	PREPARE TRANSACTION	为两阶段提交准备当前事务
	ROLLBACK PREPARED	取消一个之前为两阶段提交准备好的事务
	COMMIT PREPARED	提交一个早前为两阶段提交预备的事务
	BEGIN	事务开始
	START TRANSACTION	开始一个事务块
	COMMIT	提交操作
	SAVEPOINT	保存点
	ROLLBACK	回滚操作
	ABORT	事务中断
	END	提交当前事务
	RELEASE SAVEPOINT	销毁一个之前定义的保存点
	ROLLBACK TO SAVEPOINT	回滚到一个保存点
OBJPROPERTIES	REASSIGN OWNED	更改一个数据库角色拥有的数据库对象的拥有关系
	DROP OWNED	移除一个数据库角色拥有的数据库对象
	COMMENT	定义或者更改一个对象的注释
	SECURITY LABEL	定义或更改应用到一个对象的安全标签
PRIVILEGES	GRANT	授予权限操作
	REVOKE	撤销权限操作
	ALTER DEFAULT PRIVILEGES	定义默认访问特权
FILE	COPY	在一个文件和一个表之间复制数据
	LOAD	载入一个共享库文件
NOTIFY	NOTIFY	生成一个通知
	UNLISTEN	停止监听一个通知
	LISTEN	监听一个通知
PARAMETER	SET	更改一个运行时参数
	ALTER SYSTEM	更改一个服务器配置参数

审计选项	被审计操作	说明
	RESET	把一个运行时参数的值恢复到默认值
	SHOW	显示一个运行时参数的值
	SET TRANSACTION	设置事务的读写属性和隔离级别
	SET ROLE	设置当前会话的当前用户标识符
	SET SESSION AUTHORIZATION	设置当前会话的会话用户标识符和当前用户标识符
SESSION	DISCARD	抛弃会话状态
	SET CONSTRAINTS	为当前事务设置约束检查时机
DO	DO	执行一个匿名代码块
ANALYZE	ANALYZE	收集有关一个数据库的统计信息
VACUUM	VACUUM	垃圾收集并根据需要分析一个数据库
CHECKPOINT	CHECKPOINT	检查点（checkpoint）
LARGE OBJECT	ALTER LARGE OBJECT	更改一个大对象的定义
EXPLAIN	EXPLAIN	显示执行计划
CLUSTER	CLUSTER	根据一个索引聚簇一个表
INSERT	INSERT	表上的插入操作
UPDATE	UPDATE	表上的修改操作
DELETE	DELETE	表上的删除操作
SELECT	SELECT	表上的查询操作

3.2.1.1. 语句级审计设置

- 函数

```
bool set_audit_stmt(
settype  text,
username text,
whenever int
)
```

- 参数

表 3.3. set_audit_stmt 参数

参数	说明
settype	审计选项，必须是支持的审计选择。审计选项请参见表 3.2 “语句级审计说明”。
username	用户名，用户必须已存在，除非指定’all’审计所有用户。

参数	说明
whenever	审计时机。0为操作失败时，1为操作成功时，2为所有。

- 示例
  - 设置uxsmo用户创建/修改/删除表等语句审计。

    select uxaudit.set_audit_stmt('TABLE', 'uxsmo', '1');

  - 设置uxsmo用户创建/修改/删除数据库语句审计。

    select uxaudit.set_audit_stmt('DATABASE', 'uxsmo', '1');

  - 设置uxsmo用户创建/修改/删除函数语句审计。

    select uxaudit.set_audit_stmt('FUNCTION', 'uxsmo', '1');

3.2.1.2. 语句级审计取消

- 函数

```
bool unset_audit_stmt(
settype    text,
username  text,
whenever  int
)
```

- 参数

表 3.4. unset_audit_stmt 参数

参数	说明
settype	审计选项，必须是支持的审计选择。审计选项请参见表 3.2 “语句级审计说明”。
username	用户名，用户必须已存在，除非指定’all’审计所有用户。
whenever	审计时机。0为操作失败时，1为操作成功时，2为所有。

> **注意**
>
> 取消审计语句和设置审计语句进行匹配，只有完全匹配的才可以取消审计，否则无法取消审计。

- 示例
  - 取消uxsmo用户创建/修改/删除表等语句审计。

select uxaudit.unset_audit_stmt('TABLE', 'uxsmo', '1');

- 取消uxsmo用户创建/修改/删除数据库语句审计。

select uxaudit.unset_audit_stmt('DATABASE', 'uxsmo', '1');

- 设置uxsmo用户创建/修改/删除函数语句审计。

select uxaudit.unset_audit_stmt('FUNCTION', 'uxsmo', '1');

3.2.1.3. 语句级审计查看

- 查询系统表

select * from ux_audit_stmt;

- 查询接口

select * from uxaudit.audit_stmt_record();

3.2.2. 对象级审计

对象级审计发生在具体的对象上，需要指定模式名以及对象名。其审计选项如下表。

表 3.5. 对象级审计参数

审计选项	TABLE	VIEW	PROCEDURE FUNCTION	TRIGGER
INSERT	√	√		
UPDATE	√	√		
DELETE	√	√		
SELECT	√	√		
EXECUTE			√	√
ALL	√	√	√	√

> **注意**
>
> 不支持事件触发器，使用set设置对象级别审计，对象为事件触发器时报错该触发器不存在，是正常的。

3.2.2.1. 对象级审计设置

- 函数

```
bool set_audit_object (
setobjtype text,
```

settype text,
username text,
schemaname text,
objname text,
whenever int
)

- 参数

表 3.6. set_audit_object 参数

参数	说明
setobjtype	对象类型，必须是支持的对象类型,TABLE/VIEW/FUNCTION/TRIGGER。
settype	审计选项，必须是支持的操作类型，INSERT/UPDATE/SELECT/DELETE/EXECUTE/ALL。
username	用户名，用户必须已存在，除非指定'all'审计所有用户。
schemaname	模式名，模式必须已存在，除非指定'all'审计所有模式。
objname	对象名，对象必须已存在，除非指定'all'审计所有对象。
whenever	审计时机。0为操作失败时，1为操作成功时，2为所有。

- 示例

  - 对uxsmo用户test表的update所有操作进行审计。

    select uxaudit.set_audit_object('table', 'update', 'uxsmo', 'uxsmo', 'test',2);

  - 对uxsmo用户test表的update成功操作进行审计。

    select uxaudit.set_audit_object('table', 'update', 'uxsmo', 'uxsmo', 'test',1);

3.2.2.2. 对象级审计取消

- 函数

bool unset_audit_object(
setobjtype text,
settype text,
username text,
schemaname text,
objname text,
whenever int
)

- 参数

表 3.7. unset_audit_object 参数

参数	说明
setobjtype	对象类型，必须是支持的对象类型，TABLE/VIEW/FUNCTION/TRIGGER。
settype	审计选项，必须是支持的操作类型，INSERT/UPDATE/SELECT/DELETE/EXECUTE/ALL。
username	用户名，用户必须已存在，除非指定'all'审计所有用户。
schemaname	模式名，模式必须已存在，除非指定'all'审计所有模式。
objname	对象名，对象必须已存在，除非指定'all'审计所有对象。
whenever	审计时机。0为操作失败时，1为操作成功时，2为所有。

> **注意**
>
> 取消审计语句和设置审计语句进行匹配，只有完全匹配的才可以取消审计，否则无法取消审计。

- 示例
  - 对uxsmo用户test表的update所有操作取消审计。

    select uxaudit.unset_audit_object('table', 'update', 'uxsmo', 'uxsmo', 'test',2);

  - 对uxsmo用户test表的update成功操作进行取消审计。

    select uxaudit.unset_audit_object('table', 'update', 'uxsmo', 'uxsmo', 'test',1);

3.2.2.3. 对象级审计查看

- 查询系统表

  select * from ux_audit_object;

- 查询接口

  select * from uxaudit.audit_object_record();

3.3. 审计文件管理

uxdb审计信息存储在审计文件中。审计文件存放在数据库的auditlog_directory指定的路径，未设置则走默认值，即数据库所在路径。审计文件命名格式为“创建时间_ID”，其中“ID”为uxdb 给定的一个唯一值。审计文件的大小可以通uxdb 的 guc 参数 auditlog_rotation_size指定。当单个审计文件超过指定大小时，系统会自动切换审计文件，自动创建新的审计文件，审计

记录将写入新的审计文件中。auditlog_rotation_size 缺省值为 10M，DBA用户可在uxmaster启动时通过配置文件uxsinodb.conf进行配置，有效值范围为 0~2047M。

随着系统的运行，审计记录将会不断增加，审计文件需要更多的磁盘空间。在极限情况下，审计记录可能会因为磁盘空间不足而无法写入审计文件，最终导致系统无法正常运行。对这种情况的处理，通过设置guc参数auditlog_cleanup_time和 auditlog_retention_days进行配置。 auditlog_retention_days为审计数据保留的天数，（缺省值0，表示保留所有数据）。auditlog_cleanup_time为审计数据清理的开始时间，（缺省值’02:00:00’，凌晨两点）将删除过期的的审计文件。

3.4. 审计信息查阅

当使用UXDB提供的审计机制进行了审计设置后，审计信息都记录在log_evt表中中，审计类型用户可以通过直接查看审计表和调用函数两种方式查看，log_evt表结构如下表所示。

表 3.8. log_evt表结构

序号	列名	数据类型	说明
1	audit_no	bigint	审计记录的序号，递增
2	logtime	timestamp without time zone	时间
3	username	text	用户名
4	dbname	text	数据库名
5	pid	integer	进程PID
6	hostport	text	客户端主机:端口
7	sessionid	text	会话ID
8	sessionlinenumber	bigint	每个会话的行号
9	psdisplay	text	命令标签
10	sessionstarttime	timestamp with time zone	会话开始时间
11	vtid	text	虚拟事务ID
12	tid	text	普通事务ID
13	elevel	integer	错误严重性
14	sqlcode	integer	SQLSTATE 代码
15	errmessage	text	错误消息
16	errdetail	text	错误消息详情
17	errhint	text	提示
18	internalquery	text	导致错误的内部查询
19	internalpos	integer	错误位置所在的字符计数
20	errcontext	text	错误上下文
21	userquery	text	导致错误的用户查询
22	querypos	integer	错误位置所在的字符计数
23	fileerrorlocation	text	源代码中错误的位置

序号	列名	数据类型	说明
24	applicationname	text	应用名
25	userid	oid	用户oid标识
26	databaseid	oid	数据库oid标识
27	audit_type	text	审计类别
28	statement_id	text	命令ID
29	substatement_id	text	命令子ID
30	class	text	事件类型
31	command	text	命令标签
32	object_type	text	对象类型
33	object_name	text	对象名称
34	query	text	命令
35	parameter	text	命令参数

3.4.1. 查询所有事件

- 功能

  审计用户通过审计视图（vw_log_event）查询所有事件。

- 函数

  select * from uxaudit.vw_log_event;

> **注意**
>
> vw_log_event视图与log_evt表结构相同。

3.4.2. 查询成功的操作

3.4.2.1. 查询函数

- 函数

```
uxaudit.event_process(
limit_page    int,
where_cmd    text,
sub_list      int
)
```

- 参数

表 3.9. event_process 参数

参数	说明
limit_page	表示查询第几页数据，默认查询第一页。
where_cmd	表示查询条件，text类型,若有单引号需要转义，具体参见第 3.4.2.2.3 节 “指定查询”。

参数	说明
sub_list	表示子表索引,1表示按时间倒序第一个子表，找不到则报错，默认0为查询主表。

> **注意**
>
> 1. para3的值不能小于0。
> 2. 根据子表索引查询，如果没有找到子表则报错（无对应子表）。

3.4.2.2. 示例

3.4.2.2.1. 默认查询

不传入参数表示查询主表第一页，无筛选条件。

select * from uxaudit.event_process();

```
uxdb=> select * from uxaudit.event_process();
   sessionid     |         logtime         | audit_type | class | object_type | object_name |   command    | state
ment_id | substatement_id |                query
----------------+-------------------------+------------+-------+-------------+-------------+--------------+------
--------+-----------------+-----------------------------------
 636ddde6.1215c | 2022-11-11 13:30:32.138 | SESSION    | MISC  |             |             | SHOW         | 5
        | 1               | show enable_audit ;
 636ddde6.1215c | 2022-11-11 13:30:14.27  | SESSION    | MISC  |             |             | SHOW         | 4
        | 1               | show ux_security.login_idle_timeout
 636ddde6.1215c | 2022-11-11 13:30:14.27  | SESSION    | MISC  |             |             | SHOW         | 3
        | 1               | show running_security
 636ddde6.1215c | 2022-11-11 13:30:14.27  | SESSION    | MISC  |             |             | SHOW         | 2
        | 1               | show running_mode
 636ddde6.1215c | 2022-11-11 13:30:14.27  | SESSION    | MISC  |             |             | SHOW         | 1
        | 1               | show server_version_prompt
 636c9eb2.e3cc  | 2022-11-10 14:49:06.667 | SESSION    | DDL   | TABLE       | uxsao.t     | CREATE TABLE | 6
        | 1               | create table t(id int);
 636c9eb2.e3cc  | 2022-11-10 14:48:33.55  | SESSION    | DDL   |             |             | ALTER SYSTEM | 5
        | 1               | alter system set enable_audit = off;
 636c9eb2.e3cc  | 2022-11-10 14:48:18.448 | SESSION    | MISC  |             |             | SHOW         | 4
        | 1               | show ux_security.login_idle_timeout
 636c9eb2.e3cc  | 2022-11-10 14:48:18.448 | SESSION    | MISC  |             |             | SHOW         | 3
        | 1               | show running_security
 636c9eb2.e3cc  | 2022-11-10 14:48:18.448 | SESSION    | MISC  |             |             | SHOW         | 2
        | 1               | show running_mode
 636c9eb2.e3cc  | 2022-11-10 14:48:18.448 | SESSION    | MISC  |             |             | SHOW         | 1
        | 1               | show server_version_prompt
 636c90d0.e0a2  | 2022-11-10 13:51:17.254 | SESSION    | DDL   |             |             | ALTER SYSTEM | 6
        | 1               | alter system set enable_audit = off;
 636c90d0.e0a2  | 2022-11-10 13:49:20.052 | SESSION    | MISC  |             |             | SHOW         | 5
        | 1               | show enable_audit ;
 636c90d0.e0a2  | 2022-11-10 13:49:04.712 | SESSION    | MISC  |             |             | SHOW         | 4
        | 1               | show ux_security.login_idle_timeout
 636c90d0.e0a2  | 2022-11-10 13:49:04.711 | SESSION    | MISC  |             |             | SHOW         | 3
```

3.4.2.2.2. 部分指定查询

查询主表第二页，无筛选条件，不足20条的有多少条显示多少条数据。

select * from uxaudit.event_process(2);

```
uxdb=> select * from uxaudit.event_process(2);
   sessionid    |         logtime         | audit_type | class | object_type | object_name |   command    | statem
ent_id | substatement_id |                query
--------------+-------------------------+------------+-------+-------------+-------------+--------------+-------
-------+-----------------+---------------------------------------
 636c9eb2.e3cc | 2022-11-10 14:48:33.55  | SESSION    | DDL   |             |             | ALTER SYSTEM | 5
       | 1               | alter system set enable_audit = off;
 636c9eb2.e3cc | 2022-11-10 14:48:18.448 | SESSION    | MISC  |             |             | SHOW         | 4
       | 1               | show ux_security.login_idle_timeout
 636c9eb2.e3cc | 2022-11-10 14:48:18.448 | SESSION    | MISC  |             |             | SHOW         | 3
       | 1               | show running_security
 636c9eb2.e3cc | 2022-11-10 14:48:18.448 | SESSION    | MISC  |             |             | SHOW         | 2
       | 1               | show running_mode
 636c9eb2.e3cc | 2022-11-10 14:48:18.448 | SESSION    | MISC  |             |             | SHOW         | 1
       | 1               | show server_version_prompt
 636c90d0.e0a2 | 2022-11-10 13:51:17.254 | SESSION    | DDL   |             |             | ALTER SYSTEM | 6
       | 1               | alter system set enable_audit = off;
 636c90d0.e0a2 | 2022-11-10 13:49:20.052 | SESSION    | MISC  |             |             | SHOW         | 5
       | 1               | show enable_audit ;
 636c90d0.e0a2 | 2022-11-10 13:49:04.712 | SESSION    | MISC  |             |             | SHOW         | 4
       | 1               | show ux_security.login_idle_timeout
 636c90d0.e0a2 | 2022-11-10 13:49:04.711 | SESSION    | MISC  |             |             | SHOW         | 3
```

3.4.2.2.3. 指定查询

查询主表筛选条件为object_type= ’TABLE’第一页数据。

select * from uxaudit.event_process(1,'object_type= ''TABLE''',0);

```
uxdb=> select * from uxaudit.event_process(1,'object_type= ''TABLE''',0);
   sessionid    |         logtime         | audit_type | class | object_type | object_name |   command    | state
ment_id | substatement_id |           query
---------------+-------------------------+------------+-------+-------------+-------------+--------------+------
--------+-----------------+-------------------------
 636ddde6.1215c | 2022-11-11 13:41:16.524 | SESSION    | WRITE | TABLE       | uxsao.t     | INSERT       | 19
       | 1               | insert into t values(4);
 636ddde6.1215c | 2022-11-11 13:41:14.514 | SESSION    | WRITE | TABLE       | uxsao.t     | INSERT       | 18
       | 1               | insert into t values(3);
 636ddde6.1215c | 2022-11-11 13:41:14.512 | SESSION    | WRITE | TABLE       | uxsao.t     | INSERT       | 17
       | 1               | insert into t values(2);
 636ddde6.1215c | 2022-11-11 13:41:14.509 | SESSION    | WRITE | TABLE       | uxsao.t     | INSERT       | 16
       | 1               | insert into t values(1);
 636ddde6.1215c | 2022-11-11 13:40:31.907 | SESSION    | WRITE | TABLE       | uxsao.t3    | INSERT       | 15
       | 1               | insert into t3 values(4);
 636ddde6.1215c | 2022-11-11 13:40:30.632 | SESSION    | WRITE | TABLE       | uxsao.t3    | INSERT       | 14
       | 1               | insert into t3 values(3);
 636ddde6.1215c | 2022-11-11 13:40:30.631 | SESSION    | WRITE | TABLE       | uxsao.t3    | INSERT       | 13
       | 1               | insert into t3 values(2);
 636ddde6.1215c | 2022-11-11 13:40:30.629 | SESSION    | WRITE | TABLE       | uxsao.t3    | INSERT       | 12
       | 1               | insert into t3 values(1);
 636ddde6.1215c | 2022-11-11 13:40:30.627 | SESSION    | WRITE | TABLE       | uxsao.t2    | INSERT       | 11
       | 1               | insert into t2 values(4);
 636ddde6.1215c | 2022-11-11 13:40:30.625 | SESSION    | WRITE | TABLE       | uxsao.t2    | INSERT       | 10
       | 1               | insert into t2 values(3);
 636ddde6.1215c | 2022-11-11 13:40:30.623 | SESSION    | WRITE | TABLE       | uxsao.t2    | INSERT       | 9
       | 1               | insert into t2 values(2);
 636ddde6.1215c | 2022-11-11 13:40:30.621 | SESSION    | WRITE | TABLE       | uxsao.t2    | INSERT       | 8
       | 1               | insert into t2 values(1);
```

3.4.2.2.4. 无效查询

select * from uxaudit.event_process(1,'',-1);
select * from uxaudit.event_process(1,'',3);

```
uxdb=> select * from uxaudit.event_process(1,'',-1);
ERROR:  the index of subtable is is invalid!
uxdb=> select * from uxaudit.event_process(1,'',3);
ERROR:  No corresponding subtable!
uxdb=>
```

3.4.3. 查询失败的操作

3.4.3.1. 查询函数

- 函数

  uxaudit.event_exception (
  limit_page int ,
  where_cmd text ,
  sub_list int,
)

- 参数

  表 3.10. event_exception 参数

参数	说明
limit_page	表示查询第几页数据，默认查询第一页。
where_cmd	表示查询条件，text类型,若有单引号需要转义，具体参见第 3.4.3.2.3 节 “指定查询”
sub_list	表示查询子表，1表示按时间倒序第一个子表，找不到则报错，默认0为查询主表。

注意

1. 该函数查询除成功外的操作信息，包含LOG、失败等。
2. 页数（para1）不能小于1，否则报错。
3. 子表索引（para3）不能小于0，否则报错。
4. 根据子表索引查询，如果没有找到子表则报错（无对应子表）。

3.4.3.2. 示例

3.4.3.2.1. 默认查询

查询主表第一页，无筛选条件。

select * from uxaudit.event_exception();

```
uxdb=> select * from uxaudit.event_exception();
   sessionid    |         logtime          |    command     | elevel |                                  message
                                    | errdetail | errhint |          query
----------------+--------------------------+----------------+--------+-----------------------------------------
---------------------------------------+-----------+---------+--------------------------
 636ddde6.1215c | 2022-11-11 13:40:30.621 | INSERT         | ERROR  | relation "t1" does not exist
                                    |           |         | insert into t1 values(4);
 636ddde6.1215c | 2022-11-11 13:40:30.62  | INSERT         | ERROR  | relation "t1" does not exist
                                    |           |         | insert into t1 values(3);
 636ddde6.1215c | 2022-11-11 13:40:30.62  | INSERT         | ERROR  | relation "t1" does not exist
                                    |           |         | insert into t1 values(2);
 636ddde6.1215c | 2022-11-11 13:40:30.62  | INSERT         | ERROR  | relation "t1" does not exist
                                    |           |         | insert into t1 values(1);
 636ddde6.1215c | 2022-11-11 13:30:14.268 | authentication | LOG    | connection authorized: user=uxsao database=
uxdb application_name=uxsql        |           |         |
 636c9167.e117  | 2022-11-10 14:49:42.324 |                | LOG    | received SIGHUP, reloading configuration fi
les                                |           |         |
 636c9eb2.e3cc  | 2022-11-10 14:49:33.811 | idle           | LOG    | SYSTEM BOOT: connection broken,process exit
: Process 58316 exit.              |           |         |
 636c9eb2.e3cc  | 2022-11-10 14:48:18.446 | authentication | LOG    | connection authorized: user=uxsao database=
uxdb application_name=uxsql        |           |         |
 636c9e92.e3c7  | 2022-11-10 14:47:46.411 | authentication | LOG    | connection authorized: user=uxsao database=
uxdb application_name=audit collector |           |         |
 636c90b2.e08c  | 2022-11-10 13:51:35.649 |                | LOG    | database system is shut down
                                    |           |         |
 636c90b2.e08c  | 2022-11-10 13:51:35.637 |                | LOG    | received fast shutdown request
                                    |           |         |
 636c90d0.e0a2  | 2022-11-10 13:51:30.357 | idle           | LOG    | SYSTEM BOOT: connection broken,process exit
: Process 57506 exit.              |           |         |
```

3.4.3.2.2. 部分指定查询

查询主表第二页，无筛选条件，不足一页数据有多少条显示多少条。

select * from uxaudit.event_exception(2);

```
uxdb=> select * from uxaudit.event_exception(2);
   sessionid    |         logtime          |    command     | elevel |                                  message
                                    | errdetail | errhint |          query
----------------+--------------------------+----------------+--------+-----------------------------------------
---------------------------------------+-----------+---------+--------------------------
 636ddde6.1215c | 2022-11-11 13:40:30.62  | INSERT         | ERROR  | relation "t1" does not exist
                                    |           |         | insert into t1 values(1);
 636ddde6.1215c | 2022-11-11 13:30:14.268 | authentication | LOG    | connection authorized: user=uxsao database=
uxdb application_name=uxsql        |           |         |
 636c9167.e117  | 2022-11-10 14:49:42.324 |                | LOG    | received SIGHUP, reloading configuration fi
les                                |           |         |
 636c9eb2.e3cc  | 2022-11-10 14:49:33.811 | idle           | LOG    | SYSTEM BOOT: connection broken,process exit
: Process 58316 exit.              |           |         |
 636c9eb2.e3cc  | 2022-11-10 14:48:18.446 | authentication | LOG    | connection authorized: user=uxsao database=
uxdb application_name=uxsql        |           |         |
 636c9e92.e3c7  | 2022-11-10 14:47:46.411 | authentication | LOG    | connection authorized: user=uxsao database=
uxdb application_name=audit collector |           |         |
 636c90b2.e08c  | 2022-11-10 13:51:35.649 |                | LOG    | database system is shut down
                                    |           |         |
 636c90b2.e08c  | 2022-11-10 13:51:35.637 |                | LOG    | received fast shutdown request
                                    |           |         |
 636c90d0.e0a2  | 2022-11-10 13:51:30.357 | idle           | LOG    | SYSTEM BOOT: connection broken,process exit
: Process 57506 exit.              |           |         |
```

3.4.3.2.3. 指定查询

查询主表第1页，审计级别为LOG的信息。

select * from uxaudit.event_exception(1,'elevel=15',0);

```
uxdb=> select * from uxaudit.event_exception(1,'elevel=15',0);
   sessionid    |         logtime         |    command     | elevel |                                  message
                                    | errdetail | errhint | query
----------------+-------------------------+----------------+--------+-------------------------------------------
-------------------------------------+-----------+---------+-------
 636de29e.12255 | 2022-11-11 13:50:22.893 | authentication | LOG    | connection authorized: user=uxsao database=
uxdb application_name=uxsql          |           |         |
 636ddde6.1215c | 2022-11-11 13:30:14.268 | authentication | LOG    | connection authorized: user=uxsao database=
uxdb application_name=uxsql          |           |         |
 636c9167.e117  | 2022-11-10 14:49:42.324 |                | LOG    | received SIGHUP, reloading configuration fi
les                                  |           |         |
 636c9eb2.e3cc  | 2022-11-10 14:49:33.811 | idle           | LOG    | SYSTEM BOOT: connection broken,process exit
: Process 58316 exit.                |           |         |
 636c9eb2.e3cc  | 2022-11-10 14:48:18.446 | authentication | LOG    | connection authorized: user=uxsao database=
uxdb application_name=uxsql          |           |         |
 636c9e92.e3c7  | 2022-11-10 14:47:46.411 | authentication | LOG    | connection authorized: user=uxsao database=
uxdb application_name=audit collector |           |         |
 636c90b2.e08c  | 2022-11-10 13:51:35.649 |                | LOG    | database system is shut down
                                     |           |         |
 636c90b2.e08c  | 2022-11-10 13:51:35.637 |                | LOG    | received fast shutdown request
                                     |           |         |
 636c90d0.e0a2  | 2022-11-10 13:51:30.357 | idle           | LOG    | SYSTEM BOOT: connection broken,process exit
: Process 57506 exit.                |           |         |
 636c90d0.e0a2  | 2022-11-10 13:49:04.71  | authentication | LOG    | connection authorized: user=uxsao database=
uxdb application_name=uxsql          |           |         |
```

表　3.11.　审计级别对应列表

类型	级别
LOG	15
INFO	17
NOTICE	18
WARNING	19
ERROR	20
FATAL	21
PANIC	22

3.4.3.2.4.　无效查询

select * from uxaudit.event_exception(1,'',-1);
select * from uxaudit.event_exception(1,'',3);
select * from uxaudit.event_exception(0,'',3);

```
uxdb=> select * from uxaudit.event_exception(1,'',-1);
ERROR:  the index of subtable is is invalid!
uxdb=> select * from uxaudit.event_exception(1,'',3);
ERROR:  No corresponding subtable!
uxdb=> select * from uxaudit.event_exception(0,'',3);
ERROR:  the page number is invalid!
uxdb=>
```

3.4.4.　查询会话信息

3.4.4.1.　查询函数

- 函数

uxaudit.session (
sessionIdIn text,
sub_list   int
)

> **注意**
>
> 一次成功的连接对应一条记录。

- 参数

表 3.12. session 参数

参数	说明
sessionIdIn	表示sessionid，默认为空表示查询全部会话信息。
sub_list	表示查询子表,1表示按时间倒序第一个子表，找不到则报错，默认0为查询主表。

> **注意**
>
> 1. para2不能小于0，否则报错。
> 2. 根据子表索引查询，如果没有找到子表则报错（无对应子表）。

3.4.4.2. 示例

3.4.4.2.1. 默认查询

查询主表所有会话信息。

select * from uxaudit.session();

```
uxdb=> select * from uxaudit.session();
   sessionid    | username | dbname | applicationname | hostport |    sessionstarttime     |  pid
----------------+----------+--------+-----------------+----------+-------------------------+-------
 636de29e.12255 | uxsao    | uxdb   | uxsql           | [local]  | 2022-11-12 03:50:22+08 | 74325
 636ddde6.1215c | uxsao    | uxdb   | uxsql           | [local]  | 2022-11-12 03:30:14+08 | 74076
 636c9eb2.e3cc  | uxsao    | uxdb   | uxsql           | [local]  | 2022-11-11 04:48:18+08 | 58316
 636c90d0.e0a2  | uxsao    | uxdb   | uxsql           | [local]  | 2022-11-11 03:49:04+08 | 57506
```

3.4.4.2.2. 部分指定查询

select * from uxaudit.session('61b9a17c.12500');

```
uxdb=> select * from uxaudit.session('636de29e.12255');
   sessionid    | username | dbname | applicationname | hostport |    sessionstarttime     |  pid
----------------+----------+--------+-----------------+----------+-------------------------+-------
 636de29e.12255 | uxsao    | uxdb   | uxsql           | [local]  | 2022-11-12 03:50:22+08 | 74325
(1 row)

uxdb=>
```

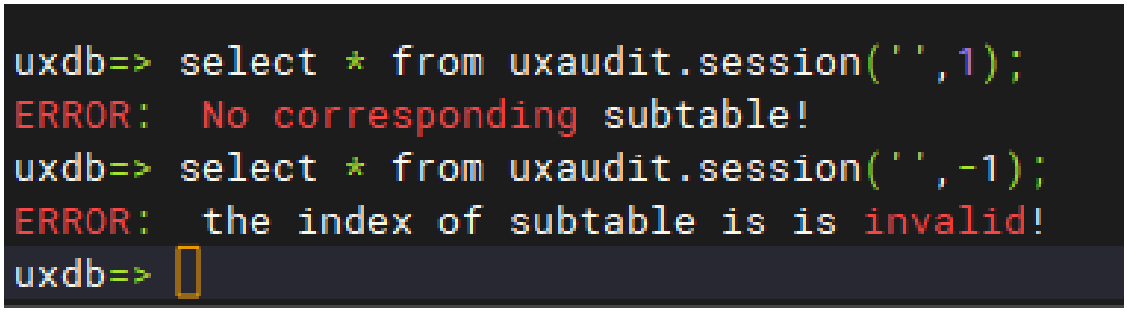
3.4.4.2.3. 指定查询

参见第 3.4.2.2.3 节 “指定查询”和第 3.4.3.2.3 节 “指定查询”。

3.4.4.2.4. 无效查询

子表索引无效或找不到报错。

select * from uxaudit.session('',1);
select * from uxaudit.session('',-1);



3.4.5. 查询登录信息

查询用户登录信息，包含成功与失败的登录。

3.4.5.1. 查询函数

- 函数

  uxaudit.login (
  usernameIn text,
  sub_list int
)

- 参数

表 3.13. login 参数

参数	说明
usernameIn	表示用户名，默认为空表示查询全部用户登录信息。
sub_list	表示查询子表,1表示按时间倒序第一个子表，找不到则报错，默认0为查询主表。

注意

1. para2不能小于0，否则报错。
2. 根据子表索引查询，如果没有找到子表则报错（无对应子表）。

3.4.5.2. 示例

3.4.5.2.1. 默认查询

默认查询所有用户登录信息，包含成功与失败的。

select * from uxaudit.login();

```
uxdb=> select * from uxaudit.login();
   sessionid    | username | dbname |    hostport     | result  |         logtime         |
             message                                          | errdetail | errhint
----------------+----------+--------+-----------------+---------+-------------------------+---------------------
-------------------------------------------------------------+-----------+---------
 636de29e.12255 | uxsao    | uxdb   | [local]         | SUCCESS | 2022-11-11 13:50:22.893 | connection authorized
: user=uxsao database=uxdb application_name=uxsql            |           |
 636ddde6.1215c | uxsao    | uxdb   | [local]         | SUCCESS | 2022-11-11 13:30:14.268 | connection authorized
: user=uxsao database=uxdb application_name=uxsql            |           |
 636c9eb2.e3cc  | uxsao    | uxdb   | [local]         | SUCCESS | 2022-11-10 14:48:18.446 | connection authorized
: user=uxsao database=uxdb application_name=uxsql            |           |
 636c9e92.e3c7  | uxsao    | uxdb   | 127.0.0.1:54810 | SUCCESS | 2022-11-10 14:47:46.411 | connection authorized
: user=uxsao database=uxdb application_name=audit collector  |           |
 636c90d0.e0a2  | uxsao    | uxdb   | [local]         | SUCCESS | 2022-11-10 13:49:04.71  | connection authorized
: user=uxsao database=uxdb application_name=uxsql            |           |
 636c90b2.e09f  | uxsao    | uxdb   | 127.0.0.1:54804 | SUCCESS | 2022-11-10 13:48:34.797 | connection authorized
: user=uxsao database=uxdb application_name=audit collector  |           |
```

3.4.5.2.2. 部分指定查询

select * from uxaudit.login('u1');

```
uxdb=> select * from uxaudit.login('uxsao');
   sessionid    | username | dbname |    hostport     | result  |         logtime         |
             message                                          | errdetail | errhint
----------------+----------+--------+-----------------+---------+-------------------------+---------------------
-------------------------------------------------------------+-----------+---------
 636de29e.12255 | uxsao    | uxdb   | [local]         | SUCCESS | 2022-11-11 13:50:22.893 | connection authorized
: user=uxsao database=uxdb application_name=uxsql            |           |
 636ddde6.1215c | uxsao    | uxdb   | [local]         | SUCCESS | 2022-11-11 13:30:14.268 | connection authorized
: user=uxsao database=uxdb application_name=uxsql            |           |
 636c9eb2.e3cc  | uxsao    | uxdb   | [local]         | SUCCESS | 2022-11-10 14:48:18.446 | connection authorized
: user=uxsao database=uxdb application_name=uxsql            |           |
 636c9e92.e3c7  | uxsao    | uxdb   | 127.0.0.1:54810 | SUCCESS | 2022-11-10 14:47:46.411 | connection authorized
: user=uxsao database=uxdb application_name=audit collector  |           |
 636c90d0.e0a2  | uxsao    | uxdb   | [local]         | SUCCESS | 2022-11-10 13:49:04.71  | connection authorized
: user=uxsao database=uxdb application_name=uxsql            |           |
 636c90b2.e09f  | uxsao    | uxdb   | 127.0.0.1:54804 | SUCCESS | 2022-11-10 13:48:34.797 | connection authorized
: user=uxsao database=uxdb application_name=audit collector  |           |
```

3.4.5.2.3. 指定查询

参见第 3.4.2.2.3 节 “指定查询”和第 3.4.3.2.3 节 “指定查询”。

3.4.5.2.4. 无效查询

子表索引无效或找不到报错。

select * from uxaudit.login('u1',1);
select * from uxaudit.login('u1',-1);

```
uxdb=> select * from uxaudit.login('u1',1);
ERROR:  No corresponding subtable!
uxdb=> select * from uxaudit.login('u1',-1);
ERROR:  the index of subtable is is invalid!
uxdb=>
```

3.5. 审计说明

3.5.1. 使用说明

1. 只要审计功能被启用，系统级的审计记录就会产生。
2. 语句级审计不针对特定的对象，只针对用户。
3. 对象级审计针对指定的用户与指定的对象进行审计。
4. 在设置审计时，审计选项不区分包含关系，都可以设置。
5. 在设置审计时，审计时机区分包含关系，会更新前一条设置。
6. 如果用户执行的一条语句与设置的若干审计项都匹配，只会在审计文件中生成一条审计记录。
7. 如果针对某个对象设置规则，但是执行语句时涉及多个对象操作，且中途报错的场景，只能审计到报错点之前的对象，并匹配规则决定能否留下审计信息。
8. 被设置审计操作的用户或对象被删除后，会有无效的审计设置记录，需要调用clear_invalid_record()清空无效记录。

3.5.2. 特殊场景说明

1. 对于某些函数，需要设置select才能审计。

   根据原理，审计函数的实现是利用object_access_hook，在SELECT语句执行阶段(ExecutorStart_hook后)且调用 InvokeFunctionExecuteHook 时，会有FUNCTION的审计记录，而且不会系统函数审计。由于不是所有的函数在执行时都严格按照这种方式，所以不能保证对所有函数的审计。

2. 对表的merge操作.

   merge操作属于dml操作，审计dml操作是根据acl权限位判断的，而merge操作检查的是mergeWhenClause的commandType的权限，因此设置insert/update/delete时，根据情况会审计merge某些操作。

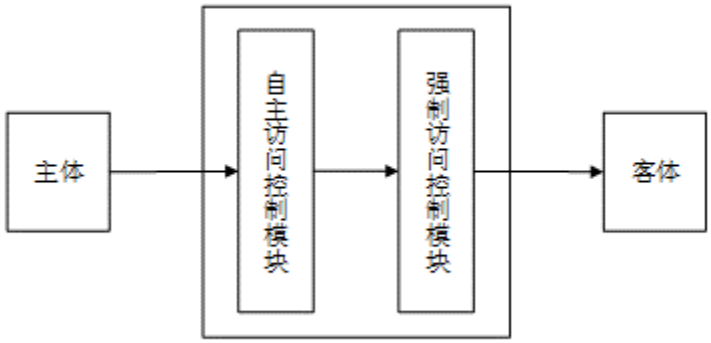
3. 某些情况，执行失败的语句也会有成功的审计记录。

   uxaudit是在执行开始时候就审计了，执行过程中的错误就会出现这种现象，多了审计信息（执行分start、run、end过程）。

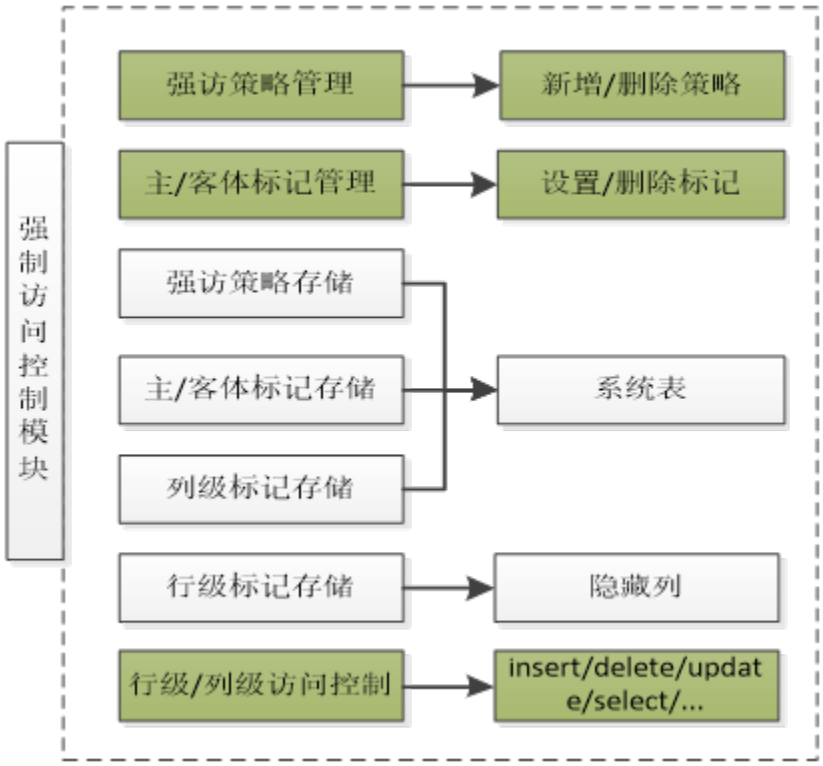
第 4 章 强制访问控制

4.1. 强制访问控制模型

UXDB将强制访问控制部署在自主访问控制模块之后，也就是说，当用户发出访问某个客体的请求时，系统首先根据访问控制列表判断用户是否具有访问客体的权限，如果有权访问，进入强制访问控制模块，获取主客体的敏感度标记，根据请求访问的类型对主客体的标记进行比较。如果比较结果是通过，那么准许用户此次访问，否则，提示用户无权访问，终止连接。



访问控制模块



4.2. 标记简介

强制访问控制的基础是敏感度标记。敏感度标记包含两个基本要素：等级和范围。

等级是一线性有序的名称序列，序列中存在着一定的级别关系。比如，绝密、秘密、机密、普通之间满足大小关系：绝密>秘密>机密>普通。而范围是一个集合，由一个或者多个名称组成，各个名称之间彼此独立无序。采用形式化描述如下：

用L=(l1,l2,...,lp)来表示等级，其中li(1≤i≤p)表示第i个等级，我们称其为第i个等级元素；用集合C={c1,c2,...,cm}及其子集来表示范围，其中每一个元素都是一个名称。标记S={S1,S2,...,Sn}是由Si(1≤i≤n)组成的集合。Si是等级元素与范围组成的二元组Si=(li,Ci),li∈L,Ci∈C。

标记之间的偏序关系定义为：对S中的任意两个元素Si=(li,Ci)和Sj=(lj,Cj)，当且仅当li≤lj且Ci是Ci的子集时，有Si≤Sj。

注意

1. 等级是必须的，且一个标记中只能有一个等级。
2. 范围是可选部分，可以指定0个或者多个范围。因此，一个标记在没有指定任何范围时，系统认为包含策略中的所有范围。当明确指定多个范围时，不同范围间用英文逗号分隔。注意，UXDB在同一个策略中最多支持31个范围。
3. 等级和范围1,…N来自同一个已存在的策略。因此，应当在使用标记之前创建策略，确定所有合法的等级和范围。UXDB对策略名字长度也作出规定，不能超过64。

4.2.1. 主体标记

数据库中的主体有五个标记：最大读标记，最大写标记，最小写标记，默认会话标记和默认写标记。安全员为用户设置标记时，一次性指定这五个标记的内容。

表 4.1. 主体标记类型

标记类型	含义
最大读标记	用户在进行读写操作时，所能达到的最高安全级别。
最大写标记	用户在进行写操作时，所能达到的最高安全级别。
最小写标记	用户在进行写操作时，所能达到的最低安全级别。用户的最小写标记最大写标记。
默认会话标记	保留。默认会话标记最大读标记。
默认写标记	用户插入元组时，元组的默认标记。最小写标记默认写标记，默认写标记最大写标记支配。

4.2.2. 标记比较

标记比较关系运算定义：如果(policyid1,level1,scope1)>=(policyid2,level2,scope2)成立，那么policyid1必须与policyid2相等(没有标记时，policyid为0)，level1数值大于等于level2(没有标记是，level数值为0)，scope1包含scope2（没有标记时scope为空）。Scope采用位编码，类型是OID(unsigned int)方便集合运算。

- 行级保护比较

select：max_read >= row_label
insert：直接插入，标记为default_write
update：max_write >= row_label >= min_write
delete：max_write >= row_label

- 列级保护比较

select：max_read >= column_label
insert：max_write >= column_label >= min_write
update：max_write >= column_label >= min_write
delete：max_write >= column_label

4.3. 访问控制规则

UXDB的强制访问控制在BLP模型的基础上作了一定的改进，采用“向下读，区间写”的原则，实现主客体的访问控制。

4.3.1. 读访问控制

读访问控制规则较为简单。UXDB首先获取客体标记，然后试图获得主体在同一策略上的最大读标记。下面两种情况下允许主体读客体数据：

- 客体没有标记。
- 客体有标记，主体在同一策略上有标记且满足主体最大读标记大于客体标记。

否则，禁止主体对客体数据的访问。

4.3.2. 写访问控制

写访问控制的规则中，UXDB首先获取客体标记，然后试图获得主体在同一策略上的最大写标记和最小写标记。下面两种情况下允许主体写客体数据：

- 客体没有标记。
- 客体有标记，主体在同一策略上有标记，且满足主体的最小写标记大于客体标记的最大写标记。

否则，禁止主体对客体数据的访问。

4.4. 强访相关函数、系统表和列

4.4.1. 相关系统表和列

UXDB安全数据库通过系统表来存储强访相关元数据，如下所示。

表 4.2. 系统表

表名	描述
ux_mac_policy	存储策略定义
ux_label_level	存储等级信息

表名	描述
ux_label_scope	存储范围信息
ux_user_label	存储主体标记
ux_class	reloptions字段存储策略id
XX设置强访策略的表	plcol，plcol_level，plcol_scope存储行标记（隐藏列）
ux_attribute	列attoptions存储列标记

1. ux_mac_policy，ux_label_level，ux_label_scope用于存储策略定义和内部编码。

表 4.3. 表ux_mac_policy列说明

名称	类型	描述
oid	oid	策略的oid
policyname	name	策略的名称

表 4.4. 表ux_label_level列说明

名称	类型	描述
policyid	oid	策略的oid
levelid	oid	策略对应等级的oid
levelname	name	策略对应等级的名称

表 4.5. 表ux_label_scope列说明

名称	类型	描述
policyid	oid	策略的oid
scopeid	oid	策略对应范围的oid
scopename	name	策略对应范围的名称

2. ux_user_label存储主体标记，主体标记可以有多个。

表 4.6. 表ux_user_label列说明

名称	类型	引用	描述
useid	oid	ux_authid.oid	策略的oid
policyid	oid		策略oid
max_read_level	oid		最大读等级
max_read_scope	oid		最大读范围
max_write_level	oid		最大写等级
max_write_scope	oid		最大写范围
min_write_level	oid		最小写等级
min_write_scope	oid		最小写范围
default_session_level	oid		默认会话等级

名称	类型	引用	描述
default_session_scope	oid		默认会话范围
default_write_level	oid		默认写等级
default_write_scope	oid		默认写范围

3. 在ux_class的reloptions选项中mac_policy_id存储策略id。

```
Expanded display is on.
uxdb=> select * from ux_class where reloptions!='{}';
-[ RECORD 1 ]-------+---------------------------
oid                 | 16385
relname             | table_copy
relnamespace        | 2200
reltype             | 16387
reloftype           | 0
relowner            | 6019
relam               | 2
relfilenode         | 16385
reltablespace       | 0
relpages            | 0
reltuples           | 0
relallvisible       | 0
reltoastrelid       | 16388
relhasindex         | f
relisshared         | f
relpersistence      | p
relkind             | r
relnatts            | 2
relchecks           | 0
relhasrules         | f
relhastriggers      | f
relhassubclass      | f
relrowsecurity      | t
relforcerowsecurity | t
relispopulated      | t
relreplident        | d
relispartition      | f
relrewrite          | 0
relfrozenxid        | 554
relminmxid          | 1
relacl              |
reloptions          | {mac_policy_id=16384}
relpartbound        |
-[ RECORD 2 ]-------+---------------------------
```

4. 在ux_attribute的attoptions选项中mac_policy_id，mac_label_level和mac_label_scope存储列标记。

```
uxdb=> select * from ux_attribute where attoptions!='{}';
-[ RECORD 1 ]-+-----------------------------------------------------------
attrelid      | 16401
attname       | b
atttypid      | 23
attstattarget | -1
attlen        | 4
attnum        | 2
attndims      | 0
attcacheoff   | -1
atttypmod     | -1
attbyval      | t
attstorage    | p
attalign      | i
attnotnull    | f
atthasdef     | f
atthasmissing | f
attidentity   |
attgenerated  |
attisdropped  | f
attislocal    | t
attinhcount   | 0
attcollation  | 0
attacl        |
attoptions    | {mac_policy_id=16400,mac_level_id=3,encoded_mac_scope=7}
attfdwoptions |
attmissingval |
```

5. 新增隐藏列plcol，plcol_level，plcol_scope存储行的标记。

4.4.2. 相关函数

表 4.7. 函数表

函数名	描述
mac_create_policy	创建策略
mac_set_user_label	设置主体标记
mac_apply_row_policy	设置客体行级标记
mac_set_column_label	设置客体标记
mac_drop_policy	删除策略
mac_drop_user_label	删除主体标记
mac_drop_row_policy	删除客体行级标记
mac_drop_column_label	删除客体列级标记

4.4.2.1. 创建策略

- 函数

  mac_create_policy('policyname','level_list','scope_list')

- 参数

表 4.8. mac_create_policy参数说明

参数	说明
policyname	策略名称。
level_list	等级列表，从高到低。
scope_list	范围列表，范围列表须小于32。

4.4.2.2. 设置主体标记

- 函数

mac_set_user_label('username','policyname','max_read','max_write','min_write','default_session', 'default_write')

- 参数

表 4.9. mac_set_user_label参数说明

参数	说明
username	主体名（用户名）。
policyname	策略名称。
max_read	最大读标记。
max_write	最大写标记。
min_write	最小写标记。
default_session	默认会话标记。
default_write	默认写标记。

- 标记规则

max_read>=max_write>=default_write>=min_write

max_read>=default_session

4.4.2.3. 设置客体行级标记

- 函数

mac_apply_row_policy('schema_name','table_name','policyid')

- 参数

表 4.10. mac_apply_row_policy参数说明

参数	说明
schema_name	需要标记的表所在的模式名。
table_name	需要标记的表名。
policyid	标记使用的策略名。

> **注意**
>
> 只能在普通表上创建客体行级标记。

4.4.2.4. 设置客体列级标记

- 函数

  mac_set_column_label('schema_name','table_name','column_name','label')

- 参数

  表 4.11. mac_set_column_label参数说明

参数	说明
schema_name	需要标记的表所在的模式名。
table_name	需要标记的表名。
column_name	需要标记的列名。
label	使用的标记。

> **注意**
>
> 只能在普通表上创建客体行级标记。

4.4.2.5. 删除策略

- 函数

  mac_drop_policy('policyname')

- 参数

  表 4.12. mac_drop_policy参数说明

参数	说明
policyname	删除策略的名称。

> **注意**
>
> 删除策略时该策略不能被应用于任何主体或客体，否则会报错。

4.4.2.6. 删除主体标记

- 函数

  mac_drop_user_label('username',' policyname')

- 参数

表 4.13. mac_drop_user_label参数说明

参数	说明
username	用户名称。
policyname	策略的名称。

4.4.2.7. 删除客体行级标记

- 函数

  mac_drop_row_policy('schema_name','table_name')

- 参数

表 4.14. mac_drop_row_policy参数说明

参数	说明
schema_name	带有标记的表所在模式名称。
table_name	带有标记的表的表名。

> **注意**
>
> 删除客体行标记说明接触对表的强制访问，此时所有用户都可以查看/操作表中数据；若再次在该表应用其他策略，则需要重新设置主体标记操作，之前的数据对所有用户可见。

4.4.2.8. 删除客体列级标记

- 函数

  mac_drop_column_label('schema_name','table_name','column_name')

- 参数

表 4.15. mac_drop_column_label参数说明

参数	说明
schema_name	带有标记的表所在模式名称。
table_name	带有标记的表的表名。
column_name	带有标记的列名。

4.5. 强访实例

4.5.1. 行级访问控制

1. uxsmo创建测试用户。

   ```
   uxdb=> \c - uxsmo
   You are now connected to database "uxdb" as user "uxsmo".
   ```

```
uxdb=> create user u_row1 password '1qaz!QAZ';
CREATE ROLE
uxdb=> create user u_row2 password '1qaz!QAZ';
CREATE ROLE
```

2. uxsso创建策略，并设置主体标记。

```
uxdb=> \c - uxsso
Password for user uxsso:
You are now connected to database "uxdb" as user "uxsso".
uxdb=> select mac_create_policy('p_row1','1,2,3,4','s1,s2,s3,s4');
 mac_create_policy
-------------------

(1 row)
```

该标记的等级依次为1，2，3，4。

该标记的范围为1~15（2^0 + 2^1 + 2^2 + 2^3）。

分别对两个用户设置最大读标记、最大写标记、最小写标记、默认会话标记和默认写标记。

```
uxdb=> select mac_set_user_label('u_row1', 'p_row1', '3:s1,s2,s3', '3:s1,s3',
 '2:s1','1:s1','3:s1,s3');
 mac_set_user_label
---------------------

(1 row)
uxdb=> select mac_set_user_label('u_row2', 'p_row1', '4:s1,s2,s3', '4:s1,s2',
 '4:s1,s2','2:s1','4:s1,s2');
 mac_set_user_label
---------------------

(1 row)
```

> **注意**
>
> u_row1的最大读标记'3:s1,s2,s3'中冒号前面的3表示等级为3，冒号后面的s1,s2,s3表示范围为7（2^0+2^1 +2^2 ）；最大写标记'3:s1,s3'中冒号前面的3表示等级为3，冒号后面的's1,s3'表示范围为5（2^0 +2^2 ）。其他标记和范围也如此计算。结果如下所示。
>
> 表 4.16. 计算表
>
测试用户	最大读	最大写	最小写	默认会话	默认写
u_row1	3:7	3:5	2:1	1:1	3:5
u_row2	4:7	4:3	4:3	2:1	4:3

3. uxsso创建测试表并设置客体行标记。

```
uxdb=> create table public.table_row (i int, k text);
CREATE TABLE
uxdb=> grant all on table_row to public;
```

```
GRANT
uxdb=> select mac_apply_row_policy('public','table_row','p_row1');
 mac_apply_row_policy
------------------------

(1 row)
```

4. u_row1和u_row2用户分别插入数据，并查看标记。

```
uxdb=> \c - u_row1
Password for user u_row1:
You are now connected to database "uxdb" as user "u_row1".
uxdb=> insert into table_row values(1,'u_row1');
INSERT 0 1
uxdb=> select plcol,plcol_level,plcol_scope,* from table_row;
 plcol | plcol_level | plcol_scope | i | k
-------+-------------+-------------+---+------
 16386 |           3 |           5 | 1 | u_row1
(1 row)
uxdb=> \c - u_row2
Password for user u_row2:
You are now connected to database "uxdb" as user "u_row2".
uxdb=> insert into table_row values(2,'u_row2');
INSERT 0 1
uxdb=> select plcol,plcol_level,plcol_scope,* from table_row;
 plcol | plcol_level | plcol_scope | i | k
-------+-------------+-------------+---+------
 16386 |           3 |           5 | 1 | u_row1
 16386 |           4 |           3 | 2 | u_row2
(2 rows)
```

5. u_row1再次查看数据。

   预期结果：u_row1用户无权访问u_row2插入的数据。

```
uxdb=> \c - u_row1
Password for user u_row1:
You are now connected to database "uxdb" as user "u_row1".
uxdb=> select plcol,plcol_level,plcol_scope,* from table_row;
 plcol | plcol_level | plcol_scope | i | k
-------+-------------+-------------+---+------
 16386 |           3 |           5 | 1 | u_row1
(1 row)
```

6. u_row1用户更新数据，分别用两个用户查询数据。

   预期结果：u_row1用户无权修改u_row2用户的数据。

```
uxdb=> \c - u_row1
Password for user u_row1:
You are now connected to database "uxdb" as user "u_row1".
uxdb=> select * from table_row;
 i | k
---+-----
 1 | u_row1
```

```
(1 row)
uxdb=> update table_row set k ='u1';
UPDATE 1
uxdb=> select * from table_row;
 i | k
---+-----
 1 | u1
(1 row)

uxdb=> \c - u_row2
Password for user u_row2:
You are now connected to database "uxdb" as user "u_row2".
uxdb=> select * from table_row;
 i | k
---+-----
 2 | u_row2
 1 | u1
(2 rows)
```

7. u_row2用户更新数据，分别用两个用户查询数据。

   预期结果：u_row2用户无权修改u_row1用户的数据。

   只修改u_row2这条数据，如下所示。

```
uxdb=> \c - u_row2
You are now connected to database "uxdb" as user "u_row2".
uxdb=> select * from table_row;
 i | k
---+-----
 2 | u_row2
 1 | u1
(2 rows)
uxdb=> update table_row set k ='u2';
UPDATE 1
uxdb=> select * from table_row;
 i | k
---+-----
 1 | u1
 2 | u2
(2 rows)
```

   u_row1的数据u1未被改变，如下所示。

```
uxdb=> \c - u_row1
Password for user u_row1:
You are now connected to database "uxdb" as user "u_row1".
uxdb=> select * from table_row;
 i | k
---+-----
 1 | u1
(1 row)
```

4.5.2. 列级访问控制

1. 创建测试用户，并创建策略。

```
uxdb=> \c - uxsmo
You are now connected to database "uxdb" as user "uxsmo".
uxdb=> create user u_col1 password '1qaz!QAZ';
CREATE ROLE
uxdb=> \c - uxsso
Password for user uxsso:
You are now connected to database "uxdb" as user "uxsso".
uxdb=> select mac_create_policy('p_col1','1,2,3,4','s1,s2,s3,s4');
 mac_create_policy
--------------------

(1 row)
```

2. 设置主体标记。

```
uxdb=> select mac_set_user_label('u_col1', 'p_col1', '3:s1,s2,s3', '3:s1,s3',
 '2:s1','1:s1','3:s1,s3');
 mac_set_user_label
--------------------

(1 row)
```

> **注意**
>
> u_col1用户最大读标记（3:7）、最大写标记（3:5）、默认写（3:5）。

3. 切换u_col1用户，创建测试表和测试数据。

```
uxdb=> \c - u_col1
Password for user u_col1;
You are now connected to database "uxdb" as user "u_col1".
uxdb=> create table public.table_col(a int ,b int,c int);
CREATE TABLE
uxdb=> insert into table_col values(1,2,3);
INSERT 0 1
uxdb=> insert into table_col values(10,20,30);
INSERT 0 1
uxdb=> select * from table_col;
 a | b | c
-----+-----+-----
  1 |  2 |  3
 10 | 20 | 30
(2 rows)
```

4. 创建客体table_col表列b标记。

```
uxdb=> \c - uxsso
Password for user uxsso;
You are now connected to database "uxdb" as user "uxsso".
uxdb=> select mac_set_column_label('public','table_col','b','p_col1','3:s1,s2,s3');
```

```
 mac_set_column_label
----------------------
```

(1 row)

b列标记（3:7）。

5. 查询表table_col数据。

   预期结果：u_col1用户可查看table_col表列b数据。

```
uxdb=> \c - u_col1
Password for user u_col1;
You are now connected to database "uxdb" as user "u_col1".
uxdb=> select * from table_col;
 a | b | c
---+---+---
  1|  2|  3
 10| 20| 30
(2 rows)
```

6. 增删改table_col数据。

   预期结果：u_col1用户无权修改table_col表列b数据。

```
uxdb=> update table_col set b=22;
ERROR:  access denied for violation of MAC rules
uxdb=> delete from table_col where b=2;
ERROR:  access denied for violation of MAC rules
uxdb=> insert into table_col values(100,200,300);
ERROR:  access denied for violation of MAC rules
uxdb=> insert into table_col(a,c) values(100,300);
INSERT 0 1
uxdb=> select * from table_col;
 a | b | c
---+---+---
  1|  2|  3
 10| 20| 30
100|   |300
(3 rows)
```

4.5.3. with plcols语法

4.5.3.1. copy to with plcols语法

1. 创建强访策略、设置uxsmo用户主体标记。

```
uxdb=> \c - uxsso
You are now connected to database "uxdb" as user "uxsso".
uxdb=> select mac_create_policy('p_copy','1,2,3,4','s1,s2,s3,s4');
 mac_create_policy
--------------------
```

(1 row)

```
uxdb=> select mac_set_user_label('uxsmo', 'p_copy', '3:s1,s2,s3', '3:s1,s3',
 '2:s1','1:s1','3:s1,s3');
 mac_set_column_label
----------------------

(1 row)
```

2. uxsmo创建测试表，并设置客体行标记。

```
uxdb=> \c - uxsmo
You are now connected to database "uxdb" as user "uxsmo".
uxdb=> create table public.table_copy (i int, k text);
CREATE TABLE
uxdb=> grant all on table_copy to public;
GRANT
uxdb=> \c - uxsso
Password for user uxsso:
You are now connected to database "uxdb" as user "uxsso".
uxdb=> select mac_apply_row_policy('public','table_copy','p_copy');
 mac_apply_row_policy
----------------------

(1 row)
```

3. 插入测试数据，并查询。

```
uxdb=> \c - uxsmo
Password for user uxsmo:
You are now connected to database "uxdb" as user "uxsmo".
uxdb=> insert into public.table_copy values(1,'11111');
INSERT 0 1
uxdb=> insert into public.table_copy values(2,'22222');
INSERT 0 1
uxdb=> select plcol,plcol_level,plcol_scope,* from table_copy;
 plcol | plcol_level | plcol_scope | i | k
-------+-------------+-------------+---+------
 16404 |           3 |           5 | 1 | 11111
 16404 |           3 |           5 | 2 | 22222
(2 rows)
```

4. copy数据到文件，并查看文件内容。

预期结果：使用with plcols的copy to语法，可将包含plcol、plcol_level、plcol_scope隐藏列表数据准确copy至文件。

```
uxdb=> copy table_copy to '/home/uxdb/table_copy.txt' with plcols;
COPY 2

[uxdb@localhost ~]$ cat /home/uxdb/table_copy.txt
p_copy:3:s1,s3  1    11111
p_copy:3:s1,s3  2    22222
```

5. copy二进制数据到文件。

预期结果：使用with plcols的copy binary to语法，不支持此语法，报错。

```
uxdb=> copy binary table_copy to '/home/uxdb/table_copy.binary' with plcols;
ERROR: copy plcols in binary mode is not in support at present
```

4.5.3.2. copy from with plcols语法

1. 清理table_copy表数据。

```
uxdb=> truncate table_copy;
TRUNCATE TABLE
uxdb=> select plcol,plcol_level,plcol_scope,* from table_copy;
 plcol | plcol_level | plcol_scope | i | k
-------+-------------+-------------+---+------
(0 rows)
```

2. 从文件copy数据到表，并查询表内容。

   预期结果：使用with plcols的copy from语法，可将包含plcol、plcol_level、plcol_scope隐藏列的文件数据准备的copy至表中。

```
uxdb=> copy table_copy from '/home/uxdb/table_copy.txt' with plcols;
COPY 2
uxdb=> select plcol,plcol_level,plcol_scope,* from table_copy;
 plcol | plcol_level | plcol_scope | i | k
-------+-------------+-------------+---+------
 16404 |           3 |           5 | 1 | 11111
 16404 |           3 |           5 | 2 | 22222
(2 rows)
```

3. 从二进制文件copy数据到表。

   预期结果：使用with plcols的copy binary from语法，不支持此语法，报错。

```
uxdb=> copy table_copy to '/home/uxdb/table_copy.binary';
COPY 2
uxdb=> copy binary table_copy from '/home/uxdb/table_copy.binary' with plcols;
ERROR: copy plcols in binary mode is not in support at present
```

4.5.4. 批量插入性能优化开关enable_mac_cache

由于强访功能，批量插入数据时性能会下降，因此添加enable_mac_cache开关。

4.5.4.1. 开关关闭

1. 查看开关状态。

```
uxdb=> show enable_mac_cache;
 enable_mac_cache
--------------------
 off
(1 row)
```

2. uxsmo创建测试表。

```
uxdb=> create table public.test(id int,info text);
```

CREATE TABLE

3. uxsso创建策略设置主体及客体标记。

```
uxdb=> select mac_create_policy('p_insert','1,2,3,4','s1,s2,s3,s4');
 mac_create_policy
-------------------

(1 row)
uxdb=> select mac_set_user_label('uxsmo', 'p_insert', '3:s1,s2,s3', '3:s1,s3',
 '2:s1','1:s1','3:s1,s3');
 mac_set_user_label
--------------------

(1 row)
uxdb=> select mac_apply_row_policy('public','test','p_insert');
 mac_apply_row_policy
----------------------

(1 row)
```

4. uxsmo插入数据并查看时间。

```
uxdb=> \c - uxsmo
Password for user uxsmo;
You are now connected to database "uxdb" as user  "uxsmo".
uxdb=> \timing
Timing is on.
uxdb=> insert into public.test select generate_series(1,100000), md5(random()::text);
INSERT 0 100000
Timing: 347.036 ms
uxdb=> insert into public.test select generate_series(1,100000), md5(random()::text);
INSERT 0 100000
Timing: 348.437 ms
uxdb=> insert into public.test select generate_series(1,100000), md5(random()::text);
INSERT 0 100000
Timing: 331.090 ms
uxdb=> insert into public.test select generate_series(1,100000), md5(random()::text);
INSERT 0 100000
Timing: 350.557 ms
uxdb=> insert into public.test select generate_series(1,100000), md5(random()::text);
INSERT 0 100000
Timing: 338.984 ms
```

4.5.4.2. 开关打开

1. uxsso设置开关打开并重启数据库。

```
uxdb=> \c - uxsmo
You are now connected to database "uxdb" as user  "uxsso".
uxdb=> alter system set enable_mac_cache =on;
ALTER SYSTEM
uxdb=> \q
[uxdb@localhost bin]$ ./ux_ctl -D testdb1 -l logfile restart
waiting for server to shut down.... done
```

server stopped
waiting for server to start....uxmaster status ready
 done
server started

2. uxsmo插入数据测试。

uxdb=> **\c - uxsmo**
You are now connected to database "uxdb" as user "uxsmo".
uxdb=> **\timing**
Timing is on.
uxdb=> **insert into public.test select generate_series(1,100000), md5(random()::text);**
INSERT 0 100000
Timing: 285.952 ms
uxdb=> **insert into public.test select generate_series(1,100000), md5(random()::text);**
INSERT 0 100000
Timing: 268.499 ms
uxdb=> **insert into public.test select generate_series(1,100000), md5(random()::text);**
INSERT 0 100000
Timing: 288.667 ms
uxdb=> **insert into public.test select generate_series(1,100000), md5(random()::text);**
INSERT 0 100000
Timing: 271.712 ms
uxdb=> **insert into public.test select generate_series(1,100000), md5(random()::text);**
INSERT 0 100000
Timing: 267.189 ms

关闭开关平均运行时间为343.221 ms，打开开关平均运行时间为276.404 ms 。由此可见，打开enable_mac_cache，批量插入性能有明显的提升。

第 5 章 数据保密性

5.1. 概述

作为应用最广泛的信息存储和处理系统，数据库中存在大量敏感数据，如何防止数据被窃取和篡改是重中之重。加密技术是提高数据库安全的一个重要手段。

UXDB提供了几个不同级别的加密，目前包括全库加密、列加密和表空间加密。方便用户应对不同场景的需要。保护数据不会因为数据库服务器偷窃、不道德的管理员、不安全网络等因素而泄漏。全库加密和列加密使用不冲突，但是不建议同时开启。

5.2. 全库加密

全库加密即对数据库中所有的系统表、数据表、索引、视图和存储过程等进行加密处理。这种加密方法简单快捷，只需对相应数据库文件进行加密处理即可，对于企业或者用户简单的备份整个数据库，可以采取这种操作更加方便。但是数据库中的数据共享性高，会同时被多个用户和应用访问使用，因此，全库加密会对系统性能会产生一定的影响。

5.2.1. 全库加密方式

只需在初始化的时候加上-M参数就可以进行全库加密。初始化带-M，启动时也需要加-M参数。

全库加密在标准模式、兼容模式、标准安全模式、兼容安全模式下都适用。

5.2.2. 示例

这里以标准安全模式为例。

1. 初始化并启动数据库。

```
./initdb -M -D testdb -W –security
./ux_ctl -M -D testdb -l logfile start
```

```
[uxdb@localhost bin]$ ./initdb -M -D testdb -W --security
The files belonging to this database system will be owned by user "uxdb".
This user must also own the server process.

The databse cluster user will be set to "uxdb".
The database cluster will be initialized with locale "en_US.UTF_8".
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".

Enter new database encryption key:
Enter new column encryption key(len is 1~16):
Data page checksums are enabled.
Using uxcrypto for full database encryption.
Enter new System Manage officer user's password:
Enter it again:
Enter new System Security officer user's password:
Enter it again:
Enter new System Audit officer user's password:
Enter it again:

creating directory testdb ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... posix
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting default time zone ... Asia/Shanghai
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

Success. You can now start the database server using:

    ./uxdb -M -D testdb
or
    ./ux_ctl -M -D testdb -l logfile start
and
start client using:

    ./uxsql -d uxdb -U uxsmo
```

```
[uxdb@localhost bin]$ ./ux_ctl -M -D testdb -l logfile start
Enter database encryption key:
waiting for server to start....uxmaster status ready
 done
server started
```

2. 登录控制台，创建表并插入数据。

./uxsql -d uxdb -U uxsmo
uxdb=> **create table public.t1(id int ,name char(10));**
CREATE TABLE
uxdb=> **insert into t1 values (1,'hhhhhh');**
INSERT 0 1
uxdb=> **insert into t1 values (1,'hhhhhh');**

```
INSERT 0 1
uxdb=> checkpoint;
CHECKPOINT
```

3. 查询表文件的路径。

```
uxdb=> select ux_relation_filepath('t1');
 ux_relation_filepath
----------------------
 base/13592/16384
(1 row)
```

4. 进入集群目录查看表文件。

```
[uxdb@localhost bin]$ vi testdb/base/13592/16384

  1 0000000: d0a9 0868 7acd edc0 e4b6 93b6 c6c3 da99  ...hz...........
  2 0000010: a78e 5dba 006f f2c4 6303 8aed e3dd c7dc  ..]..o..c.......
  3 0000020: 0c3b 0ecd 6dfb 470a 4674 53a8 b890 7510  .;..m.G.FtS...u.
  4 0000030: abfd 5612 0302 aa2e 0312 8ef5 afea 6727  ..V...........g'
  5 0000040: 1845 1be0 2b9d 7253 704b 98f1 6c3c 79dd  .E..+.rSpK..l<y.
  6 0000050: 0ca2 d456 0a21 1de5 caa3 c68a 09c5 f69d  ...V.!..........
  7 0000060: 2c60 e47f 608e f256 0287 bcf1 5f97 087f  ,`..`..V...._...
  8 0000070: 5b04 1a6e 44ac b749 0e6e a13c 0d5a bcf7  [..nD..I.n.<.Z..
  9 0000080: 619a ae45 3f9c f8f9 5e11 ebd1 8ab1 c491  a..E?...^.......
 10 0000090: c5ed f0c5 947a bf8f 966a 8b4a 3848 2f92  .....z...j.J8H/.
 11 00000a0: d7de d4fa efb4 42be 28ce 2ff1 4e3f 9ba8  ......B.(./.N?..
 12 00000b0: 6889 8763 0658 feca 8469 a88b d9aa bc60  h..c.X...i.....`
 13 00000c0: 2157 bf00 96d9 be2d 9e51 34c4 5c46 8656  !W.....-.Q4.\F.V
 14 00000d0: 4c52 6a9a 1f68 clad 2e48 15ee d698 3a3e  LRj..h...H....:>
 15 00000e0: ced7 28cd 5ee9 5664 4b57 99bb e12f e09d  ..(.^.VdKW.../..
 16 00000f0: 9f94 ca65 9d79 16e1 4669 d6a6 2adb f2e7  ...e.y..Fi..*...
 17 0000100: e384 7da5 7250 6184 27e3 6e74 3585 22a3  ..}.rPa.'.nt5.".
 18 0000110: 1544 2c99 cda2 7238 0ba6 1943 75ba e263  .D,...r8...Cu..c
 19 0000120: fc2b 7e73 2acf 8daf 1b70 c55e 95f8 df3f  .+~s*....p.^...?
 20 0000130: ca0f 8189 3f59 b0bb f32f 9b06 5bbc c872  ....?Y.../..[..r
 21 0000140: 8978 8f24 043c c8d3 07c5 7246 dc93 0c0b  .x.$.<....rF....
```

打开二进制表文件的三种方法，如下所示。

- 方法一：使用hexedit命令打开文件。
- 方法二：使用vim，然后输入:%!xxd命令转换为十六进制。
- 方法三：hexdump命令。

5.3. 列加密

列加密功能必须在带有安全功能的集群中使用，安全功能集群创建方法请参见第 1.5 节 “UXDB安全功能开关”。

5.3.1. 钱包

5.3.1.1. 概述

钱包是一个加密容器，用于存储加密功能的主密钥。钱包对密钥进行管理，使密钥与数据文件分开，从而即使得到了数据文件，没有钱包文件，数据也是无法解密的，大大提高了数据的安全性。

> **提示**
>
> 钱包需要经常备份，防止钱包因系统损坏或硬件设备导致的丢失钱包的风险。

5.3.1.2. 钱包操作

5.3.1.2.1. 生成钱包文件

5.3.1.2.1.1. 制作密钥

切换目录并制作密钥，如下所示。

cd uxdbinstall/thirdparty/gmssl
./ux_gmssl ecparam -genkey -name SM2 -out myPri.key

选项说明如下所示。

-genkey

表示生成一个 EC 密钥。

-name

标识算法名称，目前仅支持SM2算法。

-out

表示密钥的输出文件。

5.3.1.2.1.2. 生成公钥证书

用生成的密钥生成公钥证书。

./ux_gmssl req -x509 -sm3 -days 365 -key myPri.key -out myCert.pem -nodes -subj /C="CN"/ ST="SHAANXI"/L="XiAn"/O="uxsino"/OU="uxdb"/CN="uxdb"

选项说明如下所示。

ux_gmssl req

表示制作证书签名申请。

-x509

表示制作自签证书。

-sm3

表示使用 SM3 加密算法进行签名，目前公钥证书仅支持sm3算法。

-days

表示证书的有效期，单位是天。默认30天。

-key

表示签名密钥的输入文件。

-out

表示证书签名申请的输出文件。

-nodes

表示不加密输出密钥。

-subj

表示一次性写入请求信息。

-subj /C="CN"/ST="SHAANXI"/L="XiAn"/O="uxsino"/OU="uxdb"/CN="uxdb

以上表示：设置国家名称 => 设置省份名称 => 设置城市名称 => 设置组织机构名称 => 设置组织单元名称 => 设置标识名称。

5.3.1.2.1.3. 生成钱包文件

使用密钥和证书生成一个pkcs12的钱包文件，钱包中存放前面生成的公私钥。

./ux_gmssl pkcs12 -export -out uxdb_wallet.p12 -inkey myPri.key -in myCert.pem -passout pass:<password>

选项说明如下所示。

openssl pkcs12

表示制作PKCS12证书。

-export

表示导出PKCS12证书。

-out

表示PKCS12证书的输出文件；亦可以指定绝对路径。

-inkey

表示密钥的输入文件。

-in

表示签名证书的输入文件。

-passwout

表示生成的pkcs12文件密码。

示例如下所示。

./ux_gmssl pkcs12 -export -out uxdb_wallet.p12 -inkey myPri.key -in myCert.pem -passout pass:'1qaz!QAZ'

> **注意**
>
> 1. 生成的在钱包文件默认保存在ux_gmssl同目录下。
> 2. 基于安全考虑，建议不要把钱包文件存储于数据库安装目录uxdbinstall下。

5.3.1.2.1.4. 配置钱包文件

钱包相关的GUC参数为wallet_location，该参数表示钱包的全路径（包含钱包文件名），默认为/home/uxdb/wallet/uxdb_wallet.p12。

钱包生效的方式有两种：拷贝钱包文件到默认路径；修改配置参数为钱包全路径。

1. 拷贝钱包文件到默认路径。
   a. 创建默认目录(存在就跳过)。

      mkdir /home/uxdb/wallet

   b. 拷贝钱包文件到默认目录。

      cp uxdb_wallet.p12 /home/uxdb/wallet

2. 修改配置参数为钱包全路径。
   a. 修改配置文件uxsinodb.conf中参数值，或者使用alter system命令，如下所示。

      alter system set WALLET_LOCATION to '/home/uxdb/wallet/uxdb_wallet.p12'

   b. 修改之后重新启动集群使其生效。

5.3.1.2.2. 打开钱包

使用uxsql连接服务端，执行\wallet on命令，如下所示。

```
uxdb=> \wallet on
Please enter the Uxsino wallet password:
信息：The Uxsino wallet opened successfully!
wallet is on.
```

> **注意**
>
> 此处输入密码必须与生成钱包文件时输入密码保持一致。

5.3.1.2.3. 关闭钱包

使用uxsql连接服务端，执行\wallet off命令，如下所示。

```
uxdb=> \wallet on
Please enter the Uxsino wallet password:
信息：The Uxsino wallet is already closed.
wallet is off.
```

> **注意**
>
> 此处输入密码必须与生成钱包文件时输入密码保持一致。

5.3.1.2.4. 替换钱包

> **注意**
>
> 进行主密钥替换操作时，一定要进行原wallet备份，以防原wallet出现问题，导致集群不可恢复。
>
> 主密钥替换一旦失败，wallet进入admin状态(通过\wallet show查看)，在该状态下必须重新启动集群，最好在业务停止状态下去进行操作

1. 根据生成钱包文件第 5.3.1.2.1.1 节 “制作密钥”至第 5.3.1.2.1.3 节 “生成钱包文件”步骤，重新生成钱包文件，如下所示。

   [uxdb]$**./ux_gmssl ecparam -genkey -name SM2 -out myPri.key**
   [uxdb]$**./ux_gmssl req -x509 -sm3 -days 3650 -key myPri.key -out myCert.pem -nodes -subj / C="CN"/ST="SHAANXI"/L="XiAn"/O="uxsino"/OU="uxdb"/CN="uxdb"**
   [uxdb]$**./ux_gmssl pkcs12 -export -out uxdb_wallet_new.p12 -inkey myPri.key -in myCert.pem -passout pass:123456**

2. uxsql连接服务端（使用UXSMO用户）。

   ./uxsql -d uxdb -U uxsmo

3. 执行\wallet on打开wallet开关。

   uxdb=> **\wallet on**
   Please enter the Uxsino wallet password:
   INFO: The Uxsino wallet opened successfully!
   wallet is on.

4. 执行主密钥替换系统函数。

   该操作即让生成的新wallet生效。

   uxdb=> **select update_wallet_with(0, '/home/uxdb/wallet/uxdb_wallet_new.p12', '123456');**

   参数说明如下所示。

   - 参数1：为新钱包类型,0表示PKCS12文件格式；目前不支持其他类型。
   - 参数2：为新钱包的全路径。
   - 参数3：为新钱包密码。

   > **注意**
   >
   > a. 钱包open状态下进行。
   >
   > b. 仅支持uxsmo进行操作。

c. 函数执行成功后，wallet_location设置为新钱包文件路径。

5. 异常处理。

   对于操作员来说，当主密钥替换操作发生异常时，首先检查uxsinodb.auto.conf中是否存在wallet_location的配置信息。

   若不存在或者为原wallet路径的值，使用\wallet show 查看钱包状态，若非admin状态则列加密功能可正常使用；若为admin状态需重新启动集群。

   若存在且非原wallet路径的值，即已经发生了改动，则修改配置参数为原wallet全路径，并重启集群，列加密功能恢复正常使用。

5.3.1.2.5. 示例

1. 初始化安全集群。

   ./initdb -D data --security -W

2. 启动集群。

   ./ux_ctl -D data/ start

3. sql登录终端。

   使用uxsmo登录终端。

   ./uxsql -d uxdb -U uxsmo

4. 打开钱包开关。

   uxdb=> **\wallet on**
   Please enter the Uxsino wallet password:
   信息: The Uxsino wallet opened successfully!
   wallet is on.

> **注意**
>
> a. 如果钱包打开失败，请检查是否生成钱包文件，生成钱包文件后在执行钱包打开操作。
>
> b. 此处输入密码必须与生成钱包文件时输入密码保持一致。

5.3.2. 列加密集群

5.3.2.1. 密钥管理

5.3.2.1.1. 数据加密密钥

系统表ux_col_key，存储列加密密钥信息。

表 5.1. ux_col_key 参数说明

名称	类型	描述
reloid	Oid	加密表Oid
datakey	text	根密钥加密后的数据密钥
algorithm	text	加密算法
create_time	timestamp_tz	密钥创建时间
cert_fingerprint	text	主密钥指纹

注意

1. 表的owner为uxsmo，所有用户仅有select权限。
2. 当用户需要用到密钥时，通过根密钥解密得到数据加密密钥。

5.3.2.1.2. 打开钱包

当用户使用列加密功能时，首先必须打开钱包。打开钱包仅支持uxsmo用户进行操作。

uxsmo客户端键入\wallet ，提示用户输入钱包口令(即生成钱包文件时输入的口令)。

- 钱包打开（密码校验成功）

  \wallet

  \wallet on

  \wallet 1

  \wallet true

  \wallet yes

- 钱包关闭（密码校验成功）

  \wallet

  \wallet off

  \wallet 0

  \wallet false

  \wallet no

- 查看当前wallet状态

  \wallet show

\wallet s

- 强制进行口令交互认证

\wallet w

> **注意**
>
> 1. 若口令校验失败，wallet为之前状态。
> 2. 若非uxsmo用户执行\wallet失败，wallet状态保持原状。

5.3.2.1.3. ACL权限位

通过判断该用户有无 ux_col_key表的encrypt_decrypt权限，决定对加密列有无权限。该权限位用于密钥的管理，仅支持uxsmo管理员进行授权。语法如下所示。

- 赋权

  GRANT ENCRYPT_DECRYPT ON TABLE ux_col_key TO u1;

- 回收权限

  REVOKE ENCRYPT_DECRYPT ON TABLE ux_col_key FROM u1;

> **注意**
>
> 授权之前，必须成功打开钱包。

5.3.2.2. 列加密说明

- 表类型

  1. 支持表类型包含：普通表、unlogged表、toast表、索引、分区表、继承表、物化视图、复制表（create table as/like）。
  2. 不支持表类型包括：临时表，local，global。不支持的表类型创建加密列时报错。

- 数据类型

  UXDB支持一组可用于TDE列加密的特定数据类型。目前支持的数据类型如下所示。

  - CHAR
  - VARCHAR(定长/变长)
  - NUMERIC
  - INT2
  - INT
  - INT8
  - TEXT

- 算法

  目前列加密支持算法包括SM4、AES128、AES192、AES256、DES64、DES128、DES192。

- 多个加密列

  如果表中有多个加密列，则所有这些列都必须使用相同的加密算法对。

- salt加盐

  salt在列级别指定。这意味着表中的加密列可以选择不使用 salt，而不管表中的其他加密列是否使用salt。int类型不支持加盐。

- 加mac

  salt在列级别指定。指定mac时必须指定salt。

- 当操作对象存在加密列，而用户无列加解密权限时，操作是否被允许

  表 5.2. 无加解密权限时，操作是否被允许

操作	是否允许
CREATE	NO
DROP	YES
INSERT	NO
UPDATE	NO
DELETE	YES
TRUNCATE	YES

5.3.2.3. 创建表时指定加密列

5.3.2.3.1. 语法

```
create table table_name (column_name data_type [encrypted][salt | no salt] [mac | no mac]) [encrypted
 using[algorithm]];
```

5.3.2.3.2. 参数

表 5.3. 加密列参数说明

名称	描述
encrypted	设置列为加密列。
salt|no salt	是否将盐添加到明文中。添加盐会使攻击者更难通过蛮力攻击窃取数据。默认为no salt。
algorithm	加密算法，如果未指定算法的情况下加密表列，则该列将使用该SM4算法进行加密。
mac|no mac	是否添加mac， 添加mac防止数据篡改。默认为no mac。

5.3.2.4. 向现有表添加加密列

- 语法

ALTER TABLE <table name> ADD COLUMN <column name> <column type> ENCRYPTED [SALT|NO SALT] [MAC|NO MAC]

- 说明

  1. 向加密表中添加加密列，该列以原表的加密算法加密。
  2. 向明文表中添加加密列，该表成为加密表，默认以sm4算法加密。
  3. 禁止向临时表和global表中添加加密列。
  4. 加密表中最大列只有32列；默认100个表，可以由smo通过修改GUC参数ux_tde.max_entries，进行配置修改。
  5. 默认新增的加密列不带salt和mac，可指定。
  6. 增加的加密列类型为不支持加密列类型时报错。
  7. 一次性增加多列可以是普通列也可以是加密列。

5.3.2.5. 修改列的加密属性（迁移及脱密）

5.3.2.5.1. 迁移

- 语法

ALTER TABLE <table name> ALTER COLUMN <column name> [TYPE <type>] [[ENCRYPTED [SALT|NO SALT] [MAC|NO MAC]]]

- 说明

  1. 迁移操作会重写表文件。
  2. 迁移成功前提是必须是表的owner，列加密开关必须打开，操作用户需要有列加密权限。
  3. 不支持的表类型迁移报错（如临时表，global表）。
  4. 不支持的列类型迁移报错（如oid/timestamp类型迁移）。
  5. 不支持在一条sql中对同一列的加密属性多次修改。
  6. 已经加密的列再次迁移的条件：迁移前后不能完全一致，即列加密属性必须有所改变。(如encrypted只能迁移为encrypted+salt/encrypted+salt+mac，不能迁移为encrypted。)
  7. 迁移和脱密对于一个alter语句不能同时执行，语法层报错。
  8. int列迁移不能加盐，不能加mac。
  9. 如果同时存在多次修改同一列的加密属性且类型修改类型不支持：第一个alter column 若校验失败则报对应错，否则报错不能多次修改。

5.3.2.5.2. 脱密

- 语法

ALTER TABLE <table name> ALTER COLUMN <column name> [DECRYPTED];

- 说明

  1. 脱密操作会重写表文件。
  2. 脱密成功前提是必须是表的owner，列加密开关必须打开，操作用户需要有列加密权限。
  3. 明文列或已经脱密的列不能脱密。
  4. 如果同时存在多次修改同一列的加密属性：第一个alter column 脱密明文，则报错明文不能脱密，否则报错不能多次修改（alter table test alter column id encrypted,alter column id decrypted ,若id列为非加密列，则报错明文不能脱密，否则报错不能多次修改同一列）。

> **注意**
>
> 迁移脱密导致的明文表与密文表的转换，日志记录也对应记录明文或密文。

5.3.2.6. 修改列类型

- 语法

  ALTER TABLE <table name> ALTER COLUMN <column name> [TYPE <type>] [[ENCRYPTED [SALT|NO SALT] [MAC|NO MAC]]]

- 说明

  1. 修改加密表的列类型需要表的owner权限，列加密开关打开及用户列加密权限。
  2. 只修改列的类型，原有列的加密属性保持不变。
  3. 加密表和明文表的限制。
     - 无权限用户可以修改明文表的列类型。
     - 无权限用户不能修改加密表的列类型（包含明文列和加密列）。
     - 有权限用户可以修改明文表的列类型。
     - 有权限用户可以修改加密表的列类型（修改加密列时要做类型限制）。

5.3.2.7. 修改密钥及算法

- 语法

  ALTER TABLE <table name> REKEY; --只修改密钥
  ALTER TABLE <table name> REKEY USING <algo_name>; --同时修改密钥和加密算法

- 说明

  1. 数据密钥是表级密钥。
  2. 前置条件：是表的owner、列加密开关打开、有列加密权限、是加密表。
  3. 加密列的加密属性不会有所变化。
  4. rekey操作的表现为密文变化，密钥表ux_col_key中对应的记录被更新。

5. 修改的算法必须为列加密支持的算法，修改的算法可以和之前保持一致。

5.3.2.8. 创建物化视图时指定加密列

create materialized view mvname(col1 [encrypted][salt | no salt][mac | no mac], col2[…],……) [encrypted using 'algo'] as ……

> **注意**
>
> 1. 不指定加密列时，默认创建的列不加密。
> 2. 不支持使用“ALTER MATERIALIZED VIEW”改变列加密属性、算法等内容。
> 3. 只支持refresh操作，不支持修改列类型，insert等操作。

5.3.2.9. create table as/like时指定加密列

5.3.2.9.1. create table as创建新表时指定加密列和加密算法

- 语法

  create table tbname(col1 [encrypted][salt | no salt][mac | no mac], col2[…],……) [encrypted using 'algo'] as ……

- 参数

  表 5.4. create table as 参数说明

参数	描述
encrypted	指定列加密
salt	指定加密时加盐
mac	指定加密时带mac
encrypted using 'algo'	指定加密算法

- 注意

  1. 新表创建后，支持列加密的其他操作，如迁移脱密rekey等。
  2. 不支持创建临时加密表。

5.3.2.9.2. create table like指定不拷贝或拷贝加密属性

- 语法

  create table tbname_bak(like tbname [excluding/including encrypted]);

- 注意

  1. 新表创建后，支持列加密的其他操作，如迁移脱密rekey等。
  2. 不支持创建临时加密表。

- 示例

create table t1_bak(like t1);--指定不拷贝加密属性
create table t1_bak(like t1 including encrypted);--指定拷贝加密属性
create table t1_bak(like t1 excluding encrypted);--指定不拷贝加密属性
create table t1_bak(like t1 including all);--指定拷贝所有属性

5.3.2.10. 回显

1. 当用户无权限加密列出现在子句时直接报错。
2. 当用户无权限加密列出现在where/having等从句时直接报错。

> **注意**
>
> 列加密时对于特殊字符的处理说明，对于不同类型的null、''、单个空格和多个空格入库时，null和''不做加密处理，单个空格和多个空格均进行加密。
>
> 有mac必须有salt列加密加密属性若要指定mac，则salt必须存在。

5.3.3. 示例

5.3.3.1. 生成钱包

1. 切换目录。

   cd uxdbinstall/thirdparty/gmssl

2. 制作密钥。

   ./ux_gmssl ecparam -genkey -name SM2 -out myPri.key

3. 生成公钥证书。

   ./ux_gmssl req -x509 -sm3 -days 365 -key myPri.key -out myCert.pem -nodes -subj /C="CN"/ ST="SHAANXI"/L="XiAn"/O="uxsino"/OU="uxdb"/CN="uxdb"

4. 生成钱包文件。

   ./ux_gmssl pkcs12 -export -out uxdb_wallet.p12 -inkey myPri.key -in myCert.pem -passout pass:123456

5.3.3.2. 集群配置

1. 初始化安全集群带--security。

   ./initdb -D data --security -W

2. 使用ux_ctl start启动集群。

./ux_ctl -D data/ start

3. 使用uxsmo登录终端。

./uxsql -d uxdb -U uxsmo

4. 打开钱包。

uxdb=>**\wallet on**
Please enter the Uxsino wallet password:
信息：The Uxsino wallet opened successfully!
wallet is on.

> **注意**
>
> 1. 如果钱包打开失败，请检查是否生成钱包文件，生成钱包文件后在执行钱包打开操作。
> 2. 此处输入密码必须与生成钱包文件时输入密码保持一致。

5.3.3.3. 列加密操作

5.3.3.3.1. 创建列加密表并查看数据

1. 创建u1、u2用户。

create user u1 with password '1qaz!QAZ';
create user u2 with password '1qaz!QAZ';

2. 授予u1加解密权限，u2无(要进行赋权必须已经过校验)。

grant ENCRYPT_DECRYPT on ux_col_key TO u1;

3. u1用户创建加密表，并初始化数据。

create table t1 (id int, name varchar(16) encrypted);
insert into t1 values (1,'zhangsan');
insert into t1 values (2,'lisi');
insert into t1 values (3,'wangwu');
insert into t1 values (4,'zhaoliu');
insert into t1 values (5,'tianqi');

4. 授予u2用户t1的基础权限。

grant ALL on t1 TO u2;

5. u2用户登录进行查看（u2无加解密权限）。

uxdb=> **\c - u2**
Password for user u2;

```
You are now connected to database "uxdb" as user "u2".
uxdb=> select * from t1;
Error:  The current user has no column encryption and decryption permission
uxdb=> show encryptcol_display_mode;
 encryptcol_display_mode
-------------------------
 encryptcol_error
(1 row)
```

因回显默认为encryptcol_error，所以无权限时，直接报错。

修改回显为encryptcol_null(conf文件中设置encryptcol_display_mode = encryptcol_null) 重启集群生效。

```
uxdb=> \c - u2
Password for user u2;
You are now connected to database "uxdb" as user "u2".
uxdb=> show encryptcol_display_mode;
 encryptcol_display_mode
-------------------------
 encryptcol_null
(1 row)
uxdb=> select * from t1;
 id |  name
----+----------
  1 |
  2 |
  3 |
  4 |
  5 |
(5 rows)
```

此时加密列数据为NULL。

5.3.3.3.2. 向现有表添加加密列

1. 创建列加密表并插入数据。

```
\c - uxsmo
CREATE TABLE alter_t_test1(ID INTEGER ENCRYPTED);
INSERT INTO alter_t_test1 VALUES(1), (2), (3);
```

2. 添加加密列。

```
ALTER TABLE alter_t_test1 ADD COLUMN  STR VARCHAR(32) ENCRYPTED SALT
 MAC;
INSERT INTO alter_t_test1 VALUES(4, 'aaa'), (5, 'bbbb'), (6, 'ccccc');

ALTER TABLE alter_t_test1 ADD COLUMN NUM NUMERIC(20,10) ENCRYPTED;
 INSERT INTO alter_t_test1 VALUES(7, 'ddddd', 12345.6789), (8, 'eeeeee', 3.141592653);
```

3. 查看列属性。

```
SELECT a.attname, a.attcolencryptenum FROM ux_class c, ux_attribute a WHERE
 c.relname = 'alter_t_test1' AND a.attrelid = c.oid AND a.attnum>0;
```

4. 删除表并关闭钱包。

```
DROP TABLE alter_t_test1;
```

5.3.3.3.3. 修改列的加密属性（迁移及脱密）

- 迁移

```
\c - uxsmo
CREATE TABLE alter_encrypt_test1(id int,name text encrypted);
INSERT INTO alter_encrypt_test1 VALUES(1, 'aaa'), (2, 'bbbb'), (3, 'ccccc');
SELECT * FROM alter_encrypt_test1;

ALTER TABLE alter_encrypt_test1 alter COLUMN id encrypted;
ALTER TABLE alter_encrypt_test1 alter COLUMN id type oid encrypted; --error类型不支持
ALTER TABLE alter_encrypt_test1 alter COLUMN name encrypted ;--error已经是加密列
ALTER TABLE alter_encrypt_test1 alter COLUMN name encrypted salt mac;
ALTER TABLE alter_encrypt_test1 alter COLUMN name encrypted , alter COLUMN name
 decrypted;--error不允许多次修改同一列的加密属性

DROP TABLE alter_encrypt_test1;
```

- 脱密

```
\c - uxsmo
\wallet on
CREATE TABLE alter_decrypt_test2(id int encrypted, name text encrypted);
INSERT INTO alter_decrypt_test2 VALUES(1, 'aaa'), (2, 'bbbb'), (3, 'ccccc');
select reloptions from ux_class where relname = 'alter_decrypt_test2';

vacuum full alter_decrypt_test2;
ALTER TABLE alter_decrypt_test2 alter COLUMN id encrypted, alter COLUMN id
 decrypted;--第一个alter colum加密失败（这种多次修改加密属性的报错情况，先看第一个alter
 colum是否成功，成功则报错不能多次修改，否则报错对应的错误）
ALTER TABLE alter_decrypt_test2 alter COLUMN id decrypted, alter COLUMN id
 encrypted;--不能多次修改同一列的加密属性
ALTER TABLE alter_decrypt_test2 alter COLUMN id decrypted;
ALTER TABLE alter_decrypt_test2 alter COLUMN id decrypted;--error明文列不能被脱密
ALTER TABLE alter_decrypt_test2 alter COLUMN name decrypted;
select reloptions from ux_class where relname = 'alter_decrypt_test2';--已经脱密为明文表
SELECT * FROM alter_decrypt_test2;

drop table alter_decrypt_test2;
```

5.3.3.3.4. 修改列类型

```
\c - uxsmo
```

```
create user test_altertype_u1 password '1qaz!QAZ';
create user test_altertype_u2 password '1qaz!QAZ';
grant ENCRYPT_DECRYPT on ux_col_key TO test_altertype_u1;

\c - test_altertype_u2
grant test_altertype_u2 to test_altertype_u1;

\c - test_altertype_u1
create table test_altertype_table1(id int ,name varchar);
create table test_altertype_table2(id int encrypted, name varchar);
insert into test_altertype_table1 values(1, 'aaaaa'), (2, 'bbbbb'), (3, 'ccccc');
insert into test_altertype_table2 values(1, 'aaaaa'), (2, 'bbbbb'), (3, 'ccccc');
alter table test_altertype_table1 owner to test_altertype_u2;
alter table test_altertype_table2 owner to test_altertype_u2;

\c - test_altertype_u2

alter table test_altertype_table1 alter column id type varchar; --无加解密权限可以修改明文表列类型
alter table test_altertype_table1 alter column id type text; --无加解密权限可以修改明文表列类型
alter table test_altertype_table2 alter column id type varchar; --error无加解密权限禁止修改密文表列类型
alter table test_altertype_table2 alter column name type int; --error无加解密权限禁止修改密文表列类型

vacuum full test_altertype_table1;
vacuum full test_altertype_table2;
select * from test_altertype_table1;
select * from test_altertype_table2; --error

 \c - test_altertype_u1
select * from test_altertype_table1;
select * from test_altertype_table2;

alter table test_altertype_table1 alter column id type varchar; --有加解密权限可以修改明文表列类型

alter table test_altertype_table1 alter column id type int USING id::integer;--有加解密权限可以修改明文表列类型

alter table test_altertype_table1 alter column id type numeric; --有加解密权限可以修改密文表密文列类型
alter table test_altertype_table1 alter column name type char(16);--有加解密权限可以修改密文表密文列类型
alter table test_altertype_table2 alter column id type varchar2;--error有加解密权限可以修改密文表列类型，类型不支持报错
alter table test_altertype_table2 alter column id type timestamp using timestamp with time zone
 'epoch' + id * interval '1 second';--error
alter table test_altertype_table2 alter column id type varchar;
alter table test_altertype_table2 alter column name type varchar2;
alter table test_altertype_table2 alter column name type char(16);

vacuum full test_altertype_table1;
vacuum full test_altertype_table2;
```

```
select * from test_altertype_table1;
select * from test_altertype_table2;

\d test_altertype_table1
\d test_altertype_table2
drop table test_altertype_table1;
drop table test_altertype_table2;
```

5.3.3.3.5. 修改密钥及算法

```
\c - uxsmo
\wallet on
create table test_alterrekey_table1(id int encrypted, name varchar encrypted salt mac, sex int);
insert into test_alterrekey_table1 values(1, 'aaaaa',0), (2, 'bbbbb',0), (3, 'ccccc',1);
SELECT a.attname, a.attcolencryptenum FROM ux_class c, ux_attribute a WHERE c.relname =
 'test_alterrekey_table1' AND a.attrelid = c.oid AND a.attnum>0;
select reloptions from ux_class where relname = 'test_alterrekey_table1';

alter table test_alterrekey_table1 rekey;
select * from test_alterrekey_table1;
select reloptions from ux_class where relname = 'test_alterrekey_table1';

vacuum full test_alterrekey_table1;
select * from test_alterrekey_table1;
drop table test_alterrekey_table1;

create table test_alterrekey_table2(id int encrypted, name varchar encrypted salt mac, sex int);
insert into test_alterrekey_table2 values(1, 'aaaaa',0), (2, 'bbbbb',0), (3, 'ccccc',1);
SELECT a.attname, a.attcolencryptenum FROM ux_class c, ux_attribute a WHERE c.relname =
 'test_alterrekey_table2' AND a.attrelid = c.oid AND a.attnum>0;
select reloptions from ux_class where relname = 'test_alterrekey_table2';

alter table test_alterrekey_table2 rekey using 'noexist'; --error
alter table test_alterrekey_table2 rekey using 'DES128';
select * from test_alterrekey_table2;
select reloptions from ux_class where relname = 'test_alterrekey_table2';
select * from test_alterrekey_table2;

vacuum full test_alterrekey_table2;
select * from test_alterrekey_table2;
drop table test_alterrekey_table2;

CREATE TABLE test_alterrekey_table3(id char encrypted, name text encrypted salt, newname
 text, sex varchar encrypted salt mac);
INSERT INTO test_alterrekey_table3 VALUES(1, 'aaa', 'AAA','female'), (2, 'bbbb',
 'BBBB','male'), (3, 'ccccc', 'CCCCC','male');
SELECT a.attname, a.attcolencryptenum FROM ux_class c, ux_attribute a WHERE c.relname =
 'test_alterrekey_table3' AND a.attrelid = c.oid AND a.attnum>0;
select reloptions from ux_class where relname = 'test_alterrekey_table3';
select * from test_alterrekey_table3;

alter table test_alterrekey_table3 alter column id decrypted, alter column name encrypted salt
 mac;
```

```
alter table test_alterrekey_table3 alter column id type varchar;
alter table test_alterrekey_table3 rekey using 'DES128';
alter table test_alterrekey_table3 rekey using 'AES256';

select * from test_alterrekey_table3;
vacuum full test_alterrekey_table3;
select * from test_alterrekey_table3;
drop table test_alterrekey_table3;
\wallet off
```

5.3.3.3.6. 创建物化视图时设置加密属性

1. 创建物化视图。

```
uxdb=⇒ create materialized view test_maview02(id_bak,name_bak encrypt,sex_bak encrypt salt mac) as select * from test_maview_table1;
SELECT 3
uxdb=⇒ select reloptions from ux_class where relname = 'test_maview02';
         reloptions
----------------------------
 {col_encrypt_algorithm=SM4}
(1 row)

uxdb=⇒ select * from test_maview02;
 id_bak | name_bak | sex_bak
--------+----------+---------
      1 | aaaaa    | male
      2 | bbbbb    | male
      3 | ccccc    | female
(3 rows)

uxdb=⇒ \d test_maview02
                  Materialized view "uxsmo.test_maview02"
  Column  |       Type        | Colencryptenum | Collation | Nullable | Default
----------+-------------------+----------------+-----------+----------+---------
 id_bak   | integer           | 0              |           |          |
 name_bak | character varying | 1              |           |          |
 sex_bak  | text              | 7              |           |          |
```

2. 查看test_maview02磁盘文件。

```
[uxdb@localhost bin]$ hexdump -C test/base/16727/16753
00000000  00 00 00 00 38 46 d8 01  00 00 00 00 24 00 c8 7e  |....8F......$..~|
00000010  00 80 04 80 00 00 00 00  98 ff ca 00 30 ff ca 00  |............0...|
00000020  c8 fe ce 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00007ec0  00 00 00 00 00 00 00 00  33 02 00 00 00 00 00 00  |........3.......|
00007ed0  07 00 00 00 00 00 00 00  03 00 03 08 02 09 28 00  |..............(.|
00007ee0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00007ef0  03 00 00 00 23 22 65 43  77 e9 43 2a 5e 8c c9 eb  |....#"eCw.C*^...|
00007f00  c7 7f 52 cc 8d 63 60 4b  eb 2f 1d 05 c1 12 be fd  |..R..c`K./......|
00007f10  28 e7 bc 9c 9d 0a 77 39  b8 fa de 8b d2 15 28 93  |(.....w9......(.|
00007f20  00 7f 9e b1 e9 cf 60 97  f7 76 9b b8 59 fb 51 f8  |......`..v..Y.Q.|
00007f30  38 b0 fb 91 c1 3c 00 00  07 00 00 00 00 00 00 00  |8....<..........|
00007f40  02 00 03 08 02 09 28 00  00 00 00 00 00 00 00 00  |......(.........|
00007f50  00 00 00 00 00 00 00 00  02 00 00 00 23 bf 0e b1  |............#...|
00007f60  2f 0b fc ba 4a 3c 28 28  cf 5b ee 60 d5 63 b8 19  |/...J<((.[.`.c..|
00007f70  7a b1 4d a2 c1 ca 7d c8  e1 be 7e da aa b1 d6 2b  |z.M...}...~....+|
00007f80  c0 b2 29 ec 51 3f e0 99  bb 03 03 79 64 39 d7 ae  |..).Q?.....yd9..|
00007f90  df df 0b d6 0e e8 39 a9  16 ba 16 88 af 48 00 00  |......9......H..|
00007fa0  07 00 00 00 00 00 00 00  01 00 03 08 02 09 28 00  |..............(.|
00007fb0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00007fc0  01 00 00 00 23 2b 66 a7  32 03 95 e3 d0 0e bc b8  |....#+f.2.......|
00007fd0  da 23 c9 e3 bd 63 11 e4  16 3a 67 87 42 d7 28 63  |.#...c...:g.B.(c|
00007fe0  3b 33 43 13 26 94 cd 40  c8 ba 15 8b 35 9c 5e a2  |;3C.&..@....5.^.|
00007ff0  4f 1a 36 34 26 21 68 1f  19 57 16 4d 28 15 a3 4e  |O.64&!h..W.M(..N|
00008000
[uxdb@localhost bin]$
```

5.3.3.3.7. create table as/like时设置加密属性

- create table as 创建表。

```
uxdb=⇒ create table test_createtableas_table1(id int encrypt, name varchar encrypt salt mac, sex text);
CREATE TABLE
uxdb=⇒ insert into test_createtableas_table1 values(1, 'aaaaa','male'), (2, 'bbbbb','male'), (3, 'ccccc','female');
INSERT 0 3
uxdb=⇒ \d test_createtableas_table1
                Table "uxsmo.test_createtableas_table1"
 Column |       Type        | Colencryptenum | Collation | Nullable | Default
--------+-------------------+----------------+-----------+----------+---------
 id     | integer           | 1              |           |          |
 name   | character varying | 7              |           |          |
 sex    | text              | 0              |           |          |

uxdb=⇒ create table test_createtableas_table1_1 as select * from test_createtableas_table1;
SELECT 3
uxdb=⇒ \d test_createtableas_table1_1
                Table "uxsmo.test_createtableas_table1_1"
 Column |       Type        | Colencryptenum | Collation | Nullable | Default
--------+-------------------+----------------+-----------+----------+---------
 id     | integer           | 0              |           |          |
 name   | character varying | 0              |           |          |
 sex    | text              | 0              |           |          |

uxdb=⇒ create table test_createtableas_table1_3(id_bak, name_bak encrypt salt mac, sex_bak encrypt) as select * from test_createtableas_table1;
SELECT 3
uxdb=⇒ \d test_createtableas_table1_3
                Table "uxsmo.test_createtableas_table1_3"
  Column  |       Type        | Colencryptenum | Collation | Nullable | Default
----------+-------------------+----------------+-----------+----------+---------
 id_bak   | integer           | 0              |           |          |
 name_bak | character varying | 7              |           |          |
 sex_bak  | text              | 1              |           |          |
```

```
uxdb=⇒ select * from ux_class where relname='test_createtableas_table1_3';
  oid  |           relname           | relnamespace | reltype | reloftype | relowner | relam | relfilenode | reltablespace | relpages | reltuples
 | relallvisible | reltoastrelid | relhasindex | relisshared | relpersistence | relkind | relnatts | relchecks | relhasrules | relhastriggers | r
elhassubclass | relrowsecurity | relforcerowsecurity | relispopulated | relreplident | relispartition | relrewrite | relfrozenxid | relminmxid |
relacl |         reloptions          | relpartbound
-------+-----------------------------+--------------+---------+-----------+----------+-------+-------------+---------------+----------+----------
-+---------------+---------------+-------------+-------------+----------------+---------+----------+-----------+-------------+----------------+--
-------------+----------------+---------------------+----------------+--------------+----------------+------------+--------------+------------+-
-------+-----------------------------+-------------
 16746 | test_createtableas_table1_3 |        16671 |   16748 |         0 |     6015 |     2 |       16746 |             0 |        0 |         0
 |             0 |          16749 | f             | f             | p              | r       |         3 |         0 | f             | f              | f
              | f              | f                    | t              | d            | f              |          0 |          564 |          1 |
       | {col_encrypt_algorithm=SM4} |
(1 row)

uxdb=⇒ create table test_createtableas_table1_4(name_bak encrypt salt, sex_bak encrypt salt) encrypt using 'DES192' as select name ,sex from test
_createtableas_table1;--error
select reloptions from ux_class where relname = 'test_createtableas_table1_4';
\d test_createtableas_table1_4SELECT 3
uxdb=⇒ select reloptions from ux_class where relname = 'test_createtableas_table1_4';
          reloptions
-------------------------------
 {col_encrypt_algorithm=DES192}
(1 row)

uxdb=⇒ \d test_createtableas_table1_4
                Table "uxsmo.test_createtableas_table1_4"
  Column  |       Type        | Colencryptenum | Collation | Nullable | Default
----------+-------------------+----------------+-----------+----------+---------
 name_bak | character varying | 3              |           |          |
 sex_bak  | text              | 3              |           |          |

uxdb=⇒ select * from ux_relation_filepath('test_createtableas_table1_4');
 ux_relation_filepath
----------------------
 base/16727/16752
(1 row)
```

查看base/16727/16752文件。

```
00000000  00 00 00 00 58 f2 ce 01  00 00 00 00 24 00 c8 7e  |....X.......$..~|
00000010  00 80 04 80 00 00 00 00  98 ff c2 00 30 ff c2 00  |............0...|
00000020  c8 fe c6 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00007ec0  00 00 00 00 00 00 00 00  2a 02 00 00 00 00 00 00  |........*.......|
00007ed0  05 00 00 00 00 00 00 00  03 00 02 08 02 08 28 00  |..............(.|
00007ee0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00007ef0  23 49 c0 6e 53 ec 37 02  a4 35 b5 4d d3 1e f2 af  |#I.nS.7..5.M....|
00007f00  32 23 00 07 b5 bf 0b b0  fd ac c0 15 3d e5 5f f4  |2#..........=._.|
00007f10  82 e8 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00007f20  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00007f30  2a 02 00 00 00 00 00 00  05 00 00 00 00 00 00 00  |*...............|
00007f40  02 00 02 08 02 08 28 00  00 00 00 00 00 00 00 00  |......(.........|
00007f50  00 00 00 00 00 00 00 00  23 56 14 3d 42 7a 23 a3  |........#V•=Bz#.|
00007f60  ed a8 6d 00 4d 40 12 d0  c3 23 d7 5c 1e 5a c0 fb  |..m.M@...#.\.Z..|
00007f70  cf 64 39 c2 e9 70 8d 83  36 c3 00 00 00 00 00 00  |.d9..p..6.......|
00007f80  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00007f90  00 00 00 00 00 00 00 00  2a 02 00 00 00 00 00 00  |........*.......|
00007fa0  05 00 00 00 00 00 00 00  01 00 02 08 02 08 28 00  |..............(.|
00007fb0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00007fc0  23 4c bf f7 b6 5b 89 ae  fb 55 00 3e 30 cd af 57  |#L...[...U.>0..W|
00007fd0  6f 23 2e a6 70 29 23 61  72 f7 39 c2 e9 70 8d 83  |o#..p)#ar.9..p..|
00007fe0  36 c3 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |6...............|
00007ff0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00008000
```

- create table like 创建表。

```
uxdb=⇒ create table test_tablelike_table01(id int encrypt, name varchar encrypt salt mac, sex text) encrypt using 'AES192';
CREATE TABLE
uxdb=⇒ insert into test_tablelike_table01 values(1, 'aaaaa','male'), (2, 'bbbbb','male'), (3, 'ccccc','female');
INSERT 0 3
uxdb=⇒
uxdb=⇒ \d test_tablelike_table01
                    Table "uxsmo.test_tablelike_table01"
 Column |       Type        | Colencryptenum | Collation | Nullable | Default
--------+-------------------+----------------+-----------+----------+---------
 id     | integer           | 1              |           |          |
 name   | character varying | 7              |           |          |
 sex    | text              | 0              |           |          |

uxdb=⇒ create table test_tablelike_table01_3(like test_tablelike_table01 including indexes including encrypt);
CREATE TABLE
uxdb=⇒ create table test_tablelike_table01_5(like test_tablelike_table01 including all);
create table test_tablelike_table01_6(like test_tablelike_table01 including all excluding encrypt);
create table test_tablelike_table01_7(like test_tablelike_table01 excluding encrypt);CREATE TABLE
uxdb=⇒ create table test_tablelike_table01_6(like test_tablelike_table01 including all excluding encrypt);
CREATE TABLE
uxdb=⇒ create table test_tablelike_table01_7(like test_tablelike_table01 excluding encrypt);
CREATE TABLE
uxdb=⇒ \d test_tablelike_table01_3
                    Table "uxsmo.test_tablelike_table01_3"
 Column |       Type        | Colencryptenum | Collation | Nullable | Default
--------+-------------------+----------------+-----------+----------+---------
 id     | integer           | 1              |           |          |
 name   | character varying | 7              |           |          |
 sex    | text              | 0              |           |          |

uxdb=⇒ \d test_tablelike_table01_5
                    Table "uxsmo.test_tablelike_table01_5"
 Column |       Type        | Colencryptenum | Collation | Nullable | Default
--------+-------------------+----------------+-----------+----------+---------
 id     | integer           | 1              |           |          |
 name   | character varying | 7              |           |          |
 sex    | text              | 0              |           |          |
```

```
uxdb=⇒ \d test_tablelike_table01_6
                    Table "uxsmo.test_tablelike_table01_6"
 Column |       Type        | Colencryptenum | Collation | Nullable | Default
--------+-------------------+----------------+-----------+----------+---------
 id     | integer           | 0              |           |          |
 name   | character varying | 0              |           |          |
 sex    | text              | 0              |           |          |

uxdb=⇒ \d test_tablelike_table01_7
                    Table "uxsmo.test_tablelike_table01_7"
 Column |       Type        | Colencryptenum | Collation | Nullable | Default
--------+-------------------+----------------+-----------+----------+---------
 id     | integer           | 0              |           |          |
 name   | character varying | 0              |           |          |
 sex    | text              | 0              |           |          |
```

5.4. 表空间加密

5.4.1. 概述

表空间设置为加密表空间后，则对表空间中的数据进行加密处理。表空间加密只有在带安全功能的集群中支持。

5.4.2. 密钥管理

与列加密使用相同的根秘钥及二层密钥管理模型（参见第 5.3 节 “列加密”），表空间加密密钥由数据库后台生成（调用GM接口）。每个加密表或者加密表空间密钥都不相同。

> **注意**
>
> 表空间工作密钥暂不支持更新。

5.4.3. 加密表空间操作

5.4.3.1. 创建加密表空间

CREATE TABLESPACE tb1 LOCATION ‘/home/uxdb/uxdbinstall/dbsql/bin/tb1’ [ENCRYPTED];

> **注意**
>
> 1. 创建加密表空间时，必须是带有安全功能的数据库且钱包为打开转态，负责报错。
> 2. 表空间创建后不能修改加密设置。
> 3. 表空间加密算法固定为SM4，密钥为系统生成，经过钱包根密钥加密后写入系统表。
> 4. 加密表空间支持表移入/移出表空间。
> 5. 表空间加密与列加密互相独立，可以组合使用，再加密表空间中的含有加密列的表的数据，会对加密列的数据加密后，再使用表空间加密再次加密。
> 6. 不支持对表空间密钥的修改。
> 7. 加密表空间只能由系统管理员（uxsmo）创建，并且管理员的对象不能转授权。
> 8. 加密表空间中支持对表的其他操作，如：创建/删除表、视图、索引等。

5.4.3.2. 加密表空间中表的移入移出

alter table <tablename> set tablespace <tablespacename>

5.4.4. 示例

5.4.4.1. 创建/删除加密表空间

1. 打开钱包。

   uxdb=> **\wallet on**
   INFO: The Uxsino wallet opened successfully!
   wallet is on.

2. 创建表空间。

   uxdb=> **CREATE TABLESPACE tblspace_encrypt LOCATION '/home/uxdb/uxdbinstall/dbsql/bin/tblspace_encrypt' ENCRYPTED;**

3. 查看表空间是否加密属性。

   uxdb=> **SELECT spcname, encrypted from ux_tablespace WHERE spcname='tblspace_encrypt';**
   spcname | encrypted
   ----------+-----------
   tblspace_encrypt | t
   (1 行记录)

4. 删除表空间。

   uxdb=> **drop tablespace tblspace_encrypt ;**

> 注意
>
> 创建加密表空间在钱包打开之后，若钱包处于关闭状态，则无法创建加密表空间，删除加密表空间不受钱包打开或关闭的影响。

5.4.4.2. 加密表空间中表的移入移出

1. 打开钱包。

   uxdb=> **\wallet on**
   INFO: The Uxsino wallet opened successfully!
   wallet is on.

2. 创建表空间。

   uxdb=> **CREATE TABLESPACE ts_encrypted LOCATION '/home/uxdb/uxdbinstall/dbsql/bin/tblspace_encrypt' ENCRYPTED;**
   uxdb=> **CREATE TABLESPACE ts LOCATION '/home/uxdb/uxdbinstall/dbsql/bin/tblspace' ENCRYPTED;**

3. 创建表并插入数据。

   uxdb=> **CREATE TABLE t1 (id int encrypted, name varchar(32)) TABLESPACE ts_encrypted;**
   uxdb=> **CREATE TABLE t2 (id int encrypted, name varchar(32)) TABLESPACE ts;**
   uxdb=> **insert into t1 values(100, 't1name1'), (101, 't1name2');**

```
uxdb=> insert into t2 values(200, 't2name1'), (201, 't2name2');
```

4. 关闭钱包并移动表。

```
uxdb=> \wallet off
INFO: The Uxsino wallet is already closed.
wallet is off.
uxdb=> ALTER TABLE t1 SET TABLESPACE ts;
ERROR: Encryption wallet is not open.
DETAIL: Please contact your administrator.
uxdb=> ALTER TABLE t2 SET TABLESPACE ts_encrypted;
ERROR: Encryption wallet is not open.
DETAIL: Please contact your administrator.
```

5. 打开钱包并移动表。

```
uxdb=> \wallet on
INFO: The Uxsino wallet opened successfully!
wallet is on.
uxdb=> ALTER TABLE t1 SET TABLESPACE ts;
uxdb=> ALTER TABLE t2 SET TABLESPACE ts_encrypted;
uxdb=> insert into t1 values(102, 't1name3'), (103, 't1name4');
uxdb=> insert into t2 values(202, 't2name3'), (203, 't2name4');
uxdb=> SELECT * FROM t1;
 id  |  name
-----+---------
 100 | t1name1
 101 | t1name2
 102 | t1name3
 103 | t1name4
(4 行记录)
uxdb=> SELECT * FROM t2;
 id  |  name
-----+---------
 200 | t2name1
 201 | t2name2
 202 | t2name3
 203 | t2name4
(4 行记录)
uxdb=> \wallet off
INFO: The Uxsino wallet is already closed.
wallet is off.
uxdb=> DROP TABLE t1;
uxdb=> DROP TABLE t2;
uxdb=> DROP TABLESPACE ts;
uxdb=> DROP TABLESPACE ts_decrypted;
```

第 6 章 用户标识与鉴别

目前仅支持Linux平台。

6.1. 用户登录信息提示

6.1.1. 概述

用户在使用uxsql工具连接登录安全数据库时，连接登录成功后，将在终端打印用户登录信息。提示信息如下所示。

- 当前登录信息

  登录用户名、登录ip和登录时间。

- 上次登录成功信息

  登录用户名、登录ip和登录时间。

- 上次登录失败信息

  登录用户名、登录ip和登录时间。

- 上次登录成功和本次登录成功之间的登录失败次数

- 当前用户登录失败总数

- 当前用户密码的有效期

6.1.2. 支持模式

仅支持安全模式。

6.1.3. 用法示例

前提条件：安装UXDB数据库。

1. 初始化集群，并启动集群。

   ./initdb -W -D security -a
   ./ux_ctl -D security/ start

2. 登录数据库，并查看登录信息。

   [uxdb@uxdb57 bin]$ **./uxsql -d uxdb -U uxsmo**
   用户 uxsmo 的口令：
   uxsql:错误:无法连接到服务器:致命错误:用户"uxsmo" Password认证失败
   [uxdb@uxdb57 bin]$ **./uxsql -U uxsmo -d uxdb**
   uxsql (2.1.1.5C)
   Type "help" for help.

```
Welcome to 2.1.1.5C uxsql Security, the UXDB interactive terminal.
This time login information:
User Name: uxsmo
Host: [local]
Login Time: 2022-11-18 17:23:12+08
Last time login information:
User Name: uxsmo
Host:
Login Time:
Last time failed login information:
User Name: uxsmo
Host:
Login Time:

The number of failures between the last successful login and this successful login of current user is
 0

The total login failure times of current user is 0

The valid period of the current user's password cannot be set

Type: \h for help on syntax of SQL commands
\? for help information
\g or terminate with semicolon to execute query
\q to quit
```

6.2. 用户有效期限

6.2.1. 概述

增加一个用户属性“用户有效期限”，当用户有效期到期后，该用户将不能登录，尝试登录则会报错。该属性默认为无限期。仅安全管理员UXSSO可以设置除管理员用户(uxsso/uxsmo/uxsao)以外的其他用户的用户有效期。

6.2.2. 原理

1. 增加语法ROLE VALID UNTIL。
2. 增加用户属性rolrolevaliduntil。
3. 修改时检测，只有安全管理员可以修改，管理员不可被修改。
4. 用户登录时检测，过期则报错。

6.2.3. 支持模式

仅支持安全模式。

6.2.4. 用法示例

语法描述

ALTER ROLE role_name [WITH] ROLE VALID UNTIL ‘timestamp’ [...]

本功能默认是关闭的（所有用户默认永远有效）。

如需要修改用户的有效期限，通过alter语法设置。只有uxsso用户可以修改。

1. 使用uxsmo用户连接数据库。

   ./uxsql -d uxdb -U uxsmo

2. 创建一个用户。

   uxdb=> **create user u3;**
   CREATE ROLE

3. 切换到uxsso用户。

   uxdb=> **\c - uxsso**
   用户 uxsso 的口令：
   您现在已经连接到数据库 "uxdb",用户 "uxsso".

4. 执行sql，设置有效期限。

   uxdb=> **alter user u3 role valid until '20220202';**
   ALTER ROLE

5. 设置后，可以查看设置的结果。

   uxdb=> **select rolname,rolrolevaliduntil from ux_authid where rolname = 'u3';**
    rolname | rolrolevaliduntil
   ---------+------------------------
    u3 | 2022-02-02 00:00:00+08

6. 如果用户有效期过期，该用户登录会报错，无法登录。

   uxdb=> **\c - u3**
   致命错误：连接中需要一个有效的客户端登录有效时间
   保留上一次连接
   uxdb=> **\q**
   uxdb@LAPTOP-J7019II4:~/uxdbinstall/dbsql/bin$ **./uxsql -d uxdb -U u3**
   uxsql: 错误：无法连接到服务器：致命错误：连接中需要一个有效的客户端登录有效时间

6.3. 限定时间登录

6.3.1. 概述

限制某一ip客户端只能在指定时间段连接数据库，否则报错。用户创建集群后，可以通过修改ux_hba.conf配置文件的方式实现限值，配置留空则表示无限制。时间段单位精确到分钟。如果是多时间段，可以通过设置多条规则解决。

6.3.2. 支持模式

仅支持安全模式。

6.3.3. 用法示例

1. 用户创建新集群后，可以通过修改ux_hba.conf配置文件的方式，限制某一ip客户端只能在指定时间段连接数据库。如下所示。

   # TYPE DATABASE USER ADDRESS METHOD TIME(e.g. 00:00-24:00)
   local all all md5 22:00-24:00

2. 修改后，执行下面命令重新加载配置。

   ./ux_ctl -D security/ reload 或 **./ux_ctl -D security/ restart**

3. 由于登录时间不在合法时段，这时尝试登录则会报错，如下所示。

   uxdb@LAPTOP-J7019II4:~/uxdbinstall/dbsql/bin$ **./uxsql -d uxdb -U uxsso**
   uxsql: 错误: 无法连接到服务器：致命错误: 没有用于主机 "[local]", 用户 "uxsso", 数据库 "uxdb", SSL 关闭 的 ux_hba.conf 记录

4. 通过date命令查看当前系统时间。

   uxdb@LAPTOP-J7019II4:~/uxdbinstall/dbsql/bin$ **date**
   2022年 05月 26日 星期四 17:59:28 CST

5. 这时可以修改合法时间段，如下所示。

   # TYPE DATABASE USER ADDRESS METHOD TIME(e.g. 00:00-24:00)
   local all all md5 12:00-24:00

6. 也可以增加一个时间段，如下所示。

   # TYPE DATABASE USER ADDRESS METHOD TIME(e.g. 00:00-24:00)
   local all all md5 22:00-24:00
   local all all md5 12:00-22:00

7. 修改完成后，执行命令./ux_ctl -D security/ reload 或 ./ux_ctl -D security/ restart 重新加载配置，然后可以成功登录。

   uxdb@LAPTOP-J7019II4:~/uxdbinstall/dbsql/bin$ **./uxsql -d uxdb -U uxsso**
   用户 uxsso 的口令：
   uxsql (2.1.1.5A)
   输入 "help" 来获取帮助信息.

8. 如果时间段格式非法，如下所示。

   # TYPE DATABASE USER ADDRESS METHOD TIME(e.g. 00:00-24:00)
   local all all md5 22:00-25:00

9. 如果执行命令./ux_ctl -D security/ reload或 ./ux_ctl -D security/ restart，会将报错写到日志。

   cd security/log/

   查看对应日志文件，可以看到对应报错。

   2022-05-26 18:04:07.684 CST [13419] 日志：时间段格式错误！

2022-05-26 18:04:07.684 CST [13419] 上下文：配置文件"/home/uxdb/uxdbinstall/dbsql/bin/security/ux_hba.conf"的第80行
2022-05-26 18:04:07.684 CST [13419] 致命错误：无法加载ux_hba.conf
2022-05-26 18:04:07.688 CST [13419] 日志：数据库系统已关闭

6.4. 登录口令失败次数限制

6.4.1. 概述

安全集群下，限制用户登录时口令失败次数，可通过UXSMO设置GUC参数login_error_lock_times限制，参数值范围为0-5次，默认为0代表不锁定，登录失败超过设置次数，终端退出登录过程，服务器将锁定用户，在一段时间内拒绝该用户连接认证，锁定时间可由UXSMO设置参数login_error_lock_duration来设置，默认为30分钟，当用户解锁后，服务器才重新接受该用户的认证请求。

解锁方式有以下两种：

- 等待锁定时间结束自动解锁。
- uxsso手动执行select set_deblocking_user('被锁定的用户名') 解锁指定用户。

  因为需要uxsso手动解锁其它用户，所以uxsso自身不受失败次数限制，不会被锁定。

6.4.2. 支持模式

支持安全模式和安全兼容模式。

6.4.3. 用法示例

1. uxsmo连接数据库。

   ./uxsql -U uxsmo -d uxdb

2. 设置口令失败次数为3次，锁定时间为30分钟。

   uxdb=> **alter system set ux_security.login_error_lock_times = 3;**
   ALTER SYSTEM
   uxdb=> **alter system set ux_security.login_error_lock_duration = 30;**
   ALTER SYSTEM

3. 设置后退出，重启数据库。

   ux_ctl -D test restart

4. 尝试登录失败3次，用户被锁定。

   ./uxsql -U uxsmo -d uxdb
   用户 uxsmo 的口令：
   uxsql: 错误: 无法连接到服务器：致命错误: 用户uxsmo暂时被锁定，剩余时间为30分钟

5. 解锁。

   - 自动解锁

等待锁定时间结束。

- 手动解锁

  uxsso连接数据库。

  ./uxsql -U uxsso -d uxdb
  uxdb=> **select set_deblocking_user('uxsmo');**
  set_deblocking_user
  ---
  t
  (1 行记录)

6.5. 强化口令鉴别

6.5.1. 概述

安全集群下，对所有用户口令都有相应的复杂度，拒绝用户使用过于简单的密码。默认是：

1. 长度必须在8-99个字符之间（包括8、99）。
2. 至少包含一个大写字母。
3. 至少包含一个小写字母。
4. 至少包含一个数字。
5. 至少包含一个特殊字符。
6. 首尾任意一处不能带有空格。

使用场景：控制台设置用户密码指令、initdb时设置密码时、createuser工具设置密码时，都遵循此复杂度要求。

6.5.2. 支持模式

支持安全模式和安全兼容模式。

6.5.3. 用法示例

- 控制台指令

  连接数据库，创建用户u1并设置密码。

  ./uxsql -U uxsmo -d uxdb
  Create user u1 with password ‘1qaz!QAZ’

- Initdb

  初始化安全数据库目录，并输入管理员密码。

  ./initdb -W -D test -a

输入管理员密码：**1qaz!QAZ**

- createuser工具

./createuser u1 -P -U uxsmo
设置u1密码：**1qaz!QAZ**

6.6. 口令有效期

6.6.1. 概述

口令有效期是用户密码的有效使用周期，默认为无限期，可以通过ALTER ROLE rolename valid Until‘有效期时间’来修改；口令有效期到期可以登录，但只能先通过密码重置有效期后，才能进行其它操作。

安全集群下，通过设置guc参数role_password_lifetime，来控制口令有效期的上限（当前时间+参数值（单位：天）），默认为0代表是无限期，在创建用户时、修改用户口令有效期时、修改用户密码时，上限对设置、使用口令有效期进行了一些限制。

1. 创建用户时，口令有效期是口令有效期上限时间。
2. 修改用户口令有效期时，不能设置为超过上限时间。
3. 修改用户密码时，自动重置用户的密码有效期为上限时间。

6.6.2. 支持模式

支持安全模式和安全兼容模式。

6.6.3. 用法示例

假设当前时间为2022-3-1，设置步骤如下所示。

1. uxsmo连接数据库并设置口令有效期。

   ./uxsql -U uxsmo -d uxdb
   uxdb=> **alter system set ux_security.role_password_lifetime = 7;**

2. uxsmo创建用户，并修改密码。

   uxdb=> **create user u1;** --口令有效期=2022-3-8
   uxdb=> **alter user u1 password 'xxx';** --口令有效期=2022-3-8

3. uxsso设置口令有效期。

   uxdb=> **alter user u1 valid Until '2000-3-1';** --口令有效期=2000-3-1
   uxdb=> **alter user u1 valid Until '2022-3-8';** --口令有效期=2022-3-8
   uxdb=> **alter user u1 valid Until '2022-5-1';** --error

4. 有效期到期登录。

   uxdb=> **alter user u1 valid Until** ’2000-1-1’;
   uxdb=> **Create table t1** （id int）; --error

uxdb=> **Alter user u1 password 'xxx';** --口令有效期=2022-3-8
uxdb=> **Create table t1** （id int）;

6.7. 用户空闲登出

6.7.1. 概述

当客户使用uxsql客户端访问db时，如果一段时间没有操作，即空闲时间超出设置的超时退出时间，则会自动触发客户端登出。用户可以设置guc参数ux_security.login_idle_timeout，来修改用户连接超时退出时间，默认为无限长。

6.7.2. 支持模式

仅支持安全模式。

6.7.3. 用法示例

1. 本功能默认是关闭的（当ux_security.login_idle_timeout = 0 时不进行用户空闲登出检测），如需要开启则修改配置文件uxsinodb.conf，在文件最后新增一行，如下所示。

   ux_security.login_idle_timeout = 10

   > **注意**
   >
   > guc单位是秒，如果单纯写数字则认为使用默认单位。当然也可以指明单位，如6min、7d。合法的时间单位有"s", "min", "h", "d"，设置"us"或"ms"单位将无效，值依然是0，因为guc的精度是秒。
   >
   > 或者在终端使用set命令直接设置，这种方式不能指定单位，时间单位固定为秒。如下所示。
   >
   > set ux_security.login_idle_timeout = 10;

2. 设置后，如果uxsql客户端idle状态超过设定时间（即用户空闲无输入），则会自动登出，如下所示。

   uxdb=> **set ux_security.login_idle_timeout = 10;**
   SET
   uxdb=>
   客户端空闲了 10s, 连接已经主动断开
   uxdb=> **set ux_security.login_idle_timeout = 60;**
   SET
   uxdb=>
   客户端空闲了 1min, 连接已经主动断开

6.8. 限制用户连接数

6.8.1. 概述

限制用户成功连接服务端的会话个数，默认容许创建3个。

> **注意**
>
> 1. uxsmo、uxsso、uxsao不受连接个数限制，也不能被修改。
> 2. 只在安全模式集群中限制。
> 3. 连接个数包括用户所有形式的成功连接。
> 4. 只有uxsmo有权限修改连接个数。

6.8.2. 语法

- 创建时指定连个数

  create user username connection limit n; -- n为整数

- 修改连接个数

  alter user username connection limit n; -- n为整数

6.8.3. 支持模式

仅支持安全模式。

6.8.4. 查看限制限制用户连接个数

select rolname,rolconnlimit from ux_authid;

6.8.5. 用法示例

1. 创建用户时不指定连接数。

   create user u1 with password '1qaz!QAZ';

2. 查看连接数。

   ```
   uxdb=> select rolname,rolconnlimit from ux_authid where rolname='u1';
    rolname | rolconnlimit
   ---------+--------------
    u1      |            3
   (1 row)
   ```

3. 创建用户时指定连接数。

   create user u2 with password '1qaz!QAZ' connection limit 5;

4. 查看默认连接数。

   ```
   uxdb=> select rolname,rolconnlimit from ux_authid where rolname='u2';
    rolname | rolconnlimit
   ```

```
---------+--------------
 u2      |           5
(1 row)
```

5. 修改连接数。

   alter user u1 connection limit 5;

6. 查看默认连接数。

   uxdb=> **select rolname,rolconnlimit from ux_authid where rolname='u1';**
   ```
    rolname | rolconnlimit
   ---------+--------------
    u1      |           5
   (1 row)
   ```

6.9. 用户锁定与解锁

6.9.1. 概述

用户锁定为限制被锁定用户连接数据库，用户解锁为取消对用户限制连接的限制。只有uxsso可以设置用户锁定与解锁状态，安全员可以根据用户使用数据库的情况，如果发现用户操作会破坏数据库的安全，可以对用户进行锁定；当判定已经锁定的用户可以被解锁时，通过解锁操作使用户恢复正常。安全员解锁用户可以分两种情况：一是密码多次尝试登录失败被锁定，另一种是安全员主动锁定用户，解锁时会同时解除以上任意两种锁定情况。

> **注意**
>
> 1. uxsmo、uxsso、uxsao不能为被锁定或解锁的对象。
> 2. uxsmo、uxsao 如果密码多次登录失败被锁定，使用函数set_blocking_user(‘用户名’)来解锁。另外，该函数不能解锁其他用户。

6.9.2. 语法

- 用户锁定

  ALTER USER username ACCOUNT LOCK;

- 用户解锁

  ALTER USER username ACCOUNT UNLOCK;

6.9.3. 支持模式

仅支持安全模式。

6.9.4. 支持平台

Linux和Windows。

6.9.5. 用户转态查询

- 函数

  ux_user_status()

- 返回值

  用户名，自动锁定状态，主动锁定状态。

  - 自动锁定指用户密码多次认证失败被锁定。
  - 主动锁定指uxsso通过语法主动锁定一个用户。
  - 锁定状态以数字表示，0代表未锁定，1代表锁定。

6.9.6. 用法示例

1. 初始化标准安全集群的数据库实例、启动集群、连接数据库。

   ```
   [uxdb@localhost bin]$ ./initdb -W -D test_sec --security
   [uxdb@localhost bin]$ ./ux_ctl -D test_sec start
   [uxdb@localhost bin]$ ./uxsql -d uxdb -u uxsmo
   ```

2. 锁定用户后用户无法登录。

   ```
   uxdb=> alter user u1 account lock;
   ALTER ROLE
   uxdb=> \c - u1
   Password for user u1:
   致命错误：user u1 is locked by uxsso, please contact uxsso.
   Previous connection kept
   ```

3. 解锁用户后用户可以登录。

   ```
   uxdb=> alter user u1 account unlock;
   ALTER ROLE
   uxdb=> \c - u1
   Password for user u1:
   You are now connected to database "uxdb" as user "u1".
   ```