RESTORE特性仅支持 [企业版](https://www.cockroachlabs.com/product/cockroachdb/)用户，非企业版用户请参阅 [Restore Data](http://doc.cockroachchina.baidu.com/#develop/sql-statements/RESTORE/restore-data.html)

The RESTORE [语句](http://doc.cockroachchina.baidu.com/#develop/sql-statements/overview/) 通过存储服务如（AWS S3, 谷歌云 存储, NFS 或者 HTTP 存储）上的 [企业版 备份](http://doc.cockroachchina.baidu.com/#develop/sql-statements/BACKUP/)来修复集群的结构和数据。

因为CockroachDB是高容错设计, RESTORE功能主要是为了灾难恢复设计的, 例如重启一个已经丢失大多数节点副本的集群，而网络分区问题 (例如小规模节点的丢失或中断) 则不需要干预。

功能细节

Restore 目标

通过备份可以修复整张表 (包括索引) 或者 [视图](http://doc.cockroachchina.baidu.com/#develop/views/)，这个过程使用备份里存储的数据去创建 [目标数据库](http://doc.cockroachchina.baidu.com/#develop/sql-statements/RESTORE/#target-database)中完全新的表或者视图。

restoring a database的简单概念为恢复该库中的所有表数据和视图，但是不会创建该数据库。详情参阅[Target Database](http://doc.cockroachchina.baidu.com/#develop/sql-statements/RESTORE/#target-database)

RESTORE 只提供表级别粒度; 它 *不支持* 恢复表的子集

因为这个过程是为了灾难恢复设计的，CockroachDB希望[目标数据库](http://doc.cockroachchina.baidu.com/#develop/sql-statements/RESTORE/#target-database)中目前不存在这些表。这意味着目标数据库中不存在和需要恢复的表或视图同名的存在。若恢复目标的名称已经在使用了，可以如下：

* [DROP TABLE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/DROP-TABLE/), [DROP VIEW](http://doc.cockroachchina.baidu.com/#develop/sql-statements/DROP-VIEW/), or [DROP SEQUENCE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/DROP-SEQUENCE/) 然后恢复它们。 需要注意的是，当一个sequence被用在一个列中作为默认表达式时是无法drop掉的，因此在drop sequence之前需要先把表达式drop掉，然后在sequence被创建后再创建。 The setval [函数](http://doc.cockroachchina.baidu.com/#develop/functions-and-operators/#sequence-functions) 可以用来设置sequence为以前的值 。
* [修复不同数据库中的表和视图](http://doc.cockroachchina.baidu.com/#develop/sql-statements/RESTORE/#into_db).

对象依赖关系

被对象依赖的关系对象必须在对象被修复的同一时间修复。

| **对象** | **依赖于** |
| --- | --- |
| 有 [外键](http://doc.cockroachchina.baidu.com/#develop/constraints/foreign-key/) 约束的表 | 表存在 REFERENCES (然而, 这种关系可以[在修复中移除](http://doc.cockroachchina.baidu.com/#develop/sql-statements/RESTORE/#skip_missing_foreign_keys)). |
| 拥有 [sequence](http://doc.cockroachchina.baidu.com/#develop/sql-statements/CREATE-SEQUENCE/)的表 | sequence. |
| [视图](http://doc.cockroachchina.baidu.com/#develop/views/) | 视图中 SELECT 语句中使用的表 . |
| [交错表](http://doc.cockroachchina.baidu.com/#develop/performance-optimization/interleaved-tables/) | [交错表](http://doc.cockroachchina.baidu.com/#develop/performance-optimization/interleaved-tables/#interleaved-hierarchy)的父表. |

目标数据库

通常，库中表和视图都会通过我们对指定库的备份修复，需要注意的是：

* 你可以改变[目标数据库](http://doc.cockroachchina.baidu.com/#develop/sql-statements/RESTORE/#into_db).
* 如果不存在该数据库，需要[创建该数据库](http://doc.cockroachchina.baidu.com/#develop/sql-statements/CREATE-DATABASE/).

目标数据库中不能存在和需要修复的表和视图同名的存在。

用户和权限

表和视图的用户 权限 是不会被修复的。表和视图的的权限仅会从被修复的数据库继承。

然而，每一个备份都包括system.users，所以你可以[修复用户名和密码](http://doc.cockroachchina.baidu.com/#develop/sql-statements/RESTORE/#restoring-users-from-system-users-backup).

表级别的权限必须在修复完成后[给用户授权](http://doc.cockroachchina.baidu.com/#develop/sql-statements/GRANT-%3Cprivileges%3E/)

修复类型

可以选择通过全量备份恢复也可以选择通过全量备份的增量备份恢复，基于你拥有的备份文件。

| **修复类型** | **参数** |
| --- | --- |
| **Full backup** | 全量备份的路径 |
| **Full backup + incremental backups** | 全量备份的路径为第一个参数，后续参数为增量备份的路径. |

基于时间点的修复 v2.0版本的新特性

如果全量或者增量备份包含[历史版本信息](http://doc.cockroachchina.baidu.com/#develop/sql-statements/BACKUP/#backups-with-revision-history-new-in-v2-0),就可以基于指定的历史版本去修复数据。 如果没有指定时间点，数据将会修复到备份时的时间点。

性能

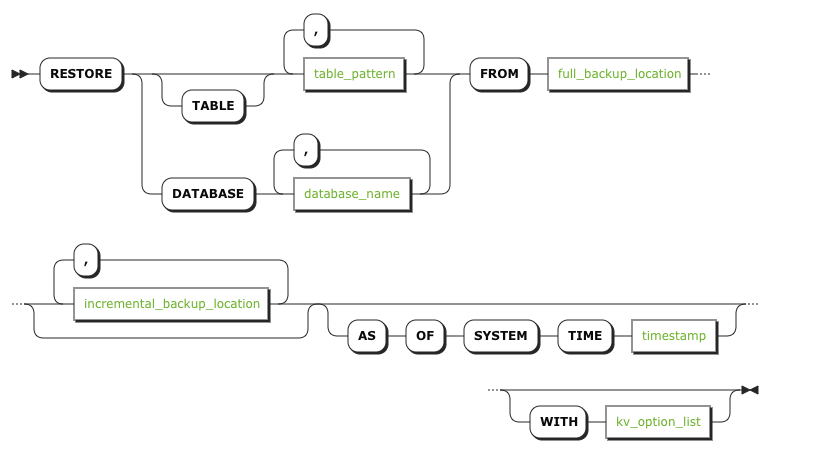
restore过程通过把工作分配给集群所有的节点，将对集群性能的影响降到最低。一旦所修复数据的range在节点中修复出来后，后续将会在节点中开始复制，然后随机选择节点分布。

当一个 修复 任务失败或者取消, 部分修复的数据将会清理，这可能对集群性能产生轻微的临时影响

查看和控制 修复任务

在CockroachDB成功开始一个修复后，会将其注册为一个任务job。可以通过[SHOW JOBS](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SHOW-JOBS/)查看细节。 在恢复任务开始后，你可以通过[PAUSE JOB](http://doc.cockroachchina.baidu.com/#develop/sql-statements/PAUSE-JOB/), [RESUME JOB](http://doc.cockroachchina.baidu.com/#develop/sql-statements/RESUME-JOB/), and [CANCEL JOB](http://doc.cockroachchina.baidu.com/#develop/sql-statements/CANCEL-JOB/)来控制。

概要



RESTORE 语句不能通过[事务](http://doc.cockroachchina.baidu.com/#develop/sql-statements/RESTORE/transactions.html)来执行

所需权限

只有root 用户可以使用RESTORE

参数

| **参数** | **描述** |
| --- | --- |
| table\_pattern | 你想要恢复的 数据表 或 [视图](http://doc.cockroachchina.baidu.com/#develop/views/) |
| database\_name | 你想要恢复的数据库名称 |
| full\_backup\_location | 全量备份的URL路径.  想要了解更多URL的结构信息，可以参考[Backup File URLs](http://doc.cockroachchina.baidu.com/#develop/sql-statements/RESTORE/#backup-file-urls). |
| incremental\_backup\_location | 增量备份的URL路径.  增量备份的列表必须按照旧到新排序。最新的增量备份timestamp必须在表的垃圾回收时间内。  关于URL的结构，详情参考 [Backup File URLs](http://doc.cockroachchina.baidu.com/#develop/sql-statements/RESTORE/#backup-file-urls).  关于垃圾回收, 详情参考 [Configure Replication Zones](http://doc.cockroachchina.baidu.com/#deploy/configure-replication-zones/#replication-zone-format). |
| AS OF SYSTEM TIME timestamp | v2.0版本新特性: 根据[timestamp](http://doc.cockroachchina.baidu.com/#develop/sql-statements/as-of-system-time/)修复数据. 根据全量或增量备份的 [历史版本](http://doc.cockroachchina.baidu.com/#develop/sql-statements/BACKUP/#backups-with-revision-history-new-in-v2-0)来基于时间点恢复。 |
| kv\_option\_list | 通过这些[options](http://doc.cockroachchina.baidu.com/#develop/sql-statements/RESTORE/#restore-option-list)控制修复任务 |

备份文件的URLs

备份路径的URL必须使用以下格式 :

片

Restore Option 列表

可以在kv\_option\_list 中通过以下option参数作为键值来控制修复的过程

into\_db

* **描述**:要将表或视图恢复到与其原来不一样的数据库中，可以[更改目标数据库](http://doc.cockroachchina.baidu.com/#develop/sql-statements/RESTORE/#restore-into-a-different-database)。如果您想恢复当前存在的表，但又不想因为原库中存在而要删除它，那么这是非常有用的。
* **Key**: into\_db
* **Value**: 想要用的数据库名称
* **示例**: WITH into\_db = 'newdb'

skip\_missing\_foreign\_keys

* **描述**: 如果想用外键恢复表，但不想恢复表引用的表，可以从表中删除外键约束，然后恢复它。
* **Key**: skip\_missing\_foreign\_keys
* **Value**: 无
* **示例**: WITH skip\_missing\_foreign\_keys

skip\_missing\_sequences

v2.0版本新特性

* **描述**: 如果要恢复依赖于序列的表，但不想恢复它引用的序列，可以从表中删除序列依赖项，恢复完后将其还原。
* **Key**: skip\_missing\_sequences
* **Value**: *No value*
* **示例**: WITH skip\_missing\_sequences

示例

修复一个单独的表

**RESTORE** bank.customers **FROM** 'gs://acme-co-backup/database-bank-2017-03-27-weekly';

修复多个表

**RESTORE** bank.customers, bank.accounts **FROM** 'gs://acme-co-backup/database-bank-2017-03-27-weekly';

修复整个数据库

**RESTORE** **DATABASE** bank **FROM** 'gs://acme-co-backup/database-bank-2017-03-27-weekly';

使用RESTORE DATABASE 需要先备份整个数据库。

根据时间点修复

**RESTORE** bank.customers **FROM** 'gs://acme-co-backup/database-bank-2017-03-27-weekly' \

**AS** **OF** SYSTEM **TIME** '2017-02-26 10:00:00';

通过增量备份修复

**RESTORE** bank.customers \

**FROM** 'gs://acme-co-backup/database-bank-2017-03-27-weekly', 'gs://acme-co-backup/database-bank-2017-03-28-nightly', 'gs://acme-co-backup/database-bank-2017-03-29-nightly';

通过增量备份基于时间点修复New in v2.0

**RESTORE** bank.customers \

**FROM** 'gs://acme-co-backup/database-bank-2017-03-27-weekly', 'gs://acme-co-backup/database-bank-2017-03-28-nightly', 'gs://acme-co-backup/database-bank-2017-03-29-nightly' \

**AS** **OF** SYSTEM **TIME** '2017-02-28 10:00:00';

在不同数据库中修复

默认情况下，表和视图被还原到它们最初所属的数据库。然而，使用[' into\_db '](http://doc.cockroachchina.baidu.com/#develop/sql-statements/RESTORE/#into_db)选项，您可以控制目标数据库。

**RESTORE** bank.customers \

**FROM** 'gs://acme-co-backup/database-bank-2017-03-27-weekly' \

**WITH** into\_db = 'newdb';

修复前移除外键

默认情况下，带有[Foreign Key](http://doc.cockroachchina.baidu.com/#develop/constraints/foreign-key/)约束的表必须与它们引用的表同时恢复。然而，使用[' skip\_missing\_foreign\_keys '](http://doc.cockroachchina.baidu.com/#develop/sql-statements/RESTORE/#skip_missing_foreign_keys)选项，您可以从表中删除外键约束，然后恢复它。

**RESTORE** bank.accounts \

**FROM** 'gs://acme-co-backup/database-bank-2017-03-27-weekly' \

**WITH** skip\_missing\_foreign\_keys;

从system.users 备份中修复用户名数据

每个完整的备份都包含system.users表，您可以使用它恢复集群的用户名和密码。然而，要恢复它们，您必须恢复system.users 表在一个新的数据库中，因为您不能删除现有的system.users表。

恢复到新的数据库后，可以将恢复后的users表数据写入集群的现有system.users表中。

**RESTORE** system.users \

**FROM** 'azure://acme-co-backup/table-users-2017-03-27-full?AZURE\_ACCOUNT\_KEY=hash&AZURE\_ACCOUNT\_NAME=acme-co' \

**WITH** into\_db = 'newdb';

**INSERT** **INTO** system.users **SELECT** \* **FROM** newdb.users;

**DROP** **TABLE** newdb.users;

其他

* [BACKUP](http://doc.cockroachchina.baidu.com/#develop/sql-statements/BACKUP/)
* [Configure Replication Zones](http://doc.cockroachchina.baidu.com/#deploy/configure-replication-zones/)