



QianBase 备份恢复手册 1.6.6

2020 年 12 月

目录

修订历史.....	iv
前言.....	iv
目标读者.....	i
相关文档.....	i
批评与建议	iii
1. 概览.....	1
2. 设计原理.....	2
3. 限制条件.....	3
4. 备份恢复的前提	4
5. 数据库可用性	5
6. 备份操作类型和命令	6
备份操作类型	6
操作命令	6
访问规则	7
7. 使用策略.....	8
8. 备份，导出和标签重用	10
9. 预创建备份标记	17
10. 灾难恢复集群的导入，恢复	18
11. 非 trafodion 用户，导入备份到灾难恢复集群.....	22

12. 语法语义.....	23
备份和恢复	23
创建增量属性 schema 和 table	24
删除备份	24
查看备份	25
导出/导入备份	25
查看/清除对象锁	26
查看备份恢复进度	26
恢复到特定时间点	28
13. 备份还原操作的事务性影响	29
性能	29
存储	29
14. 事务日志.....	30

版权

© Copyright 2019-2020 贵州易鲸捷信息技术有限公司

公告

本文档包含的信息如有更改，恕不另行通知。

保留所有权利。除非版权法允许，否则在未经易鲸捷预先书面许可的情况下，严禁改编或翻译本手册的内容。易鲸捷对于本文中所包含的技术或编辑错误、遗漏概不负责。

易鲸捷产品和服务附带的正式担保声明中规定的担保是该产品和服务享有的唯一担保。本文中的任何信息均不构成额外的保修条款。

声明

Microsoft® 和 Windows® 是美国微软公司的注册商标。Java® 和 MySQL® 是 Oracle 及其子公司的注册商标。Bosun 是 Stack Exchange 的商标。Apache®、Hadoop®、HBase®、Hive®、openTSDB®、Sqoop® 和 Trafodion® 是 Apache 软件基金会的商标。Esgyn, EsgynDB 和 QianBase 是易鲸捷的商标。

修订历史

版本	日期	描述
1.6.6	2020 年 12 月	增加了使用的限制条件
1.5.4	2020 年 3 月	更新恢复到特定时间点的备份恢复
1.5.3	2020 年 2 月	更新查看/清除对象锁和查看进度语法
1.5.0	2020 年 1 月	新增查看/清除对象锁和查看进度语法
1.4.0	2019 年 11 月	新增基于时间点的恢复
1.1.0	2019 年 7 月	本指南解释了如何使用钱库的备份恢复技术，来规划数据库的灾难恢复。

前言

为保证关键应用程序依赖于 QianBase 的可用性及防止数据丢失，本指南解释了 QianBase 的灾难恢复(Disaster Recovery - DR)策略及可用的备份恢复技术支持。

目标读者

本指南的目标读者为 QianBase 系统管理员和用户。

相关文档

本指南为 QianBase 文档库的一部分，QianBase 文档库**包括但不限于**以下文档：

文档名称	说明
QianBase 安装部署指南	本文介绍安装 QianBase，包括安装前准备、安装 Hadoop 发行版、故障排除、配置、启用安全功能、提高安全性和卸载 QianBase 等。
易鲸捷 Designer 用户指南	本文介绍易鲸捷图形化数据库管理工具
易鲸捷迁移工具用户指南	本文介绍如何安装和使用易鲸捷迁移工具。
QianBase 技术白皮书	本文介绍 QianBase 技术架构，组件介绍，技术特点等。
QianBase 数据库规划文档	本文介绍节点数量规划、数据目录和安装部署目录规划、集群角色分配规划等。
QianBase 管理员手册	本文介绍 QianBase 的日常运维常用系统命令、常用检查 SQL，用户权限配置，连接设置等内容。
QianBase 常见问题提排查与解决	本文介绍如何排查和解决 QianBase 的常见问题。
QianBase 灾难恢复手册	本文介绍 QianBase 灾难恢复设计原理，方案

	建议以及使用手册。
QianBase 备份恢复手册	本文介绍 QianBase 备份恢复设计原理，方案建议以及使用手册。
QianBase 数据库扩容指南	本文介绍 QianBase 如何更换节点，增加节点，删除节点等操作。
QianBase 数据库参数调优建议	本文介绍如何进行数据模型优化，CQD 参数优化等。
QianBase 客户端安装手册	本文介绍 QianBase JDBC，ODBC 以及 Trafci 驱动安装。
QianBase JDBC 程序员参考指南	本文介绍 QianBase JDBC 驱动连接设置，开发人员指南。
QianBase ODBC 程序员参考指南	本文介绍 QianBase ODBC 驱动连接设置，开发人员指南。
QianBase SPSQL 存储过程用户手册	本文介绍 QianBase SPSQL 存储过程的使用。
Esgyn DBManager 用户手册	本文介绍图形化数据库监控运维工具 DB Manager 的使用。
QianBase 数据库迁移指南	本文介绍如何将常见关系型数据库（Oracle、MySQL、SQL server 等）迁移至 QianBase。
QianBase SQL 用户手册	本文是 QianBase 的 SQL 使用手册。

前言

批评与建议

我们支持您对本指南做出的任何批评与建议，并尽力提供符合您需求的文档。

若您发现任何错误、或有任何改进建议，请发邮件至 support@esgyn.cn。

1. 概览

1. 概览

备份和恢复策略的基本目的是保护数据，如果出现导致数据丢失的灾难，依靠以下策略，则能够恢复数据：

- 制定完整的数据备份策略，测试备份数据的完整性，并以安全的方式对备份数据进行归档
- 为定时备份配置数据库环境
- 定期向单独的系统上导入备份并恢复备份数据集，对恢复的数据集的正确性进行测试
- 将数据归档
- 在发生故障的情况下，通过使用指定为标签或时间点的备份数据集来还原主系统

QianBase 通过多种方式确保在发生灾难性故障时数据库数据集能够恢复。包括：在线备份整个数据库的能力，在线备份 schema(s)/table(s)的能力和在线增量备份 schema(s)/table(s)的能力。

schema(s)/table(s)在本文档中通常为“对象”。

非增量备份称为常规备份。全量常规备份会备份全部对象。

启用增量备份的对象，系统会捕获并储存其数据的每一次变更。执行增量备份时，它将记录先前常规备份的恢复点，并以此开始本次的增量备份。

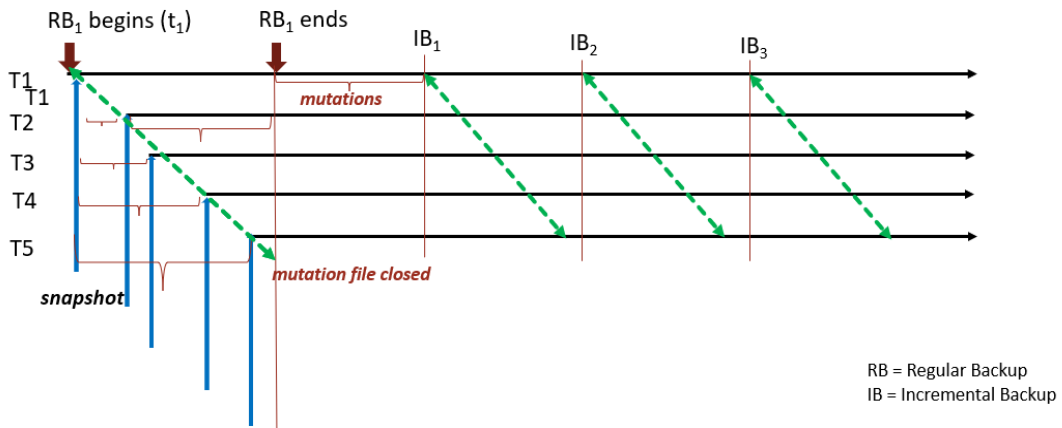
将增量备份导出到远程存储设备时，系统仅复制先前导出的备份的增量数据。这可以大大减少传输的数据量。

2. 设计原理

QianBase 的备份功能旨在解决两个限制

- 在不影响应用程序工作负载的情况下随时启动备份：在备份运行时，允许任何应用程序对数据库进行事务性修改。
- 数据备份发生的时候，确保数据一致性。

通过捕获数据库随时间变化的状态，确保在恢复时，数据由 QianBase 数据重放引擎组装在一起，以提供一致的状态，从而解决了这些相互冲突的约束。如下图：



备份首先启用全量备份，获取数据库的快照。此全量备份构成后续增量备份的基础，后续增量备份将捕获自上次备份（全量备份或增量备份）以来的数据变更。

每个备份数据集包括一组存储数据更改的文件（称为 mutations）和一个记录安全权限，备份表（tables）列表，文件位置等信息的备份元数据。这样可以使备份数据集移至其他软件兼容群集以进行恢复。

尽管捕获了突变数据，但可以将其还原到启动增量备份操作时的状态（“按标签还原”），也可以还原到过去的特定时间点（“按时间戳还原”）。

QianBase Manager 中可以设置定期自动调用备份功能。备份完成后，备份数据集可以都自动导出到单独的 QianBase 实例（可以是灾备集群）。

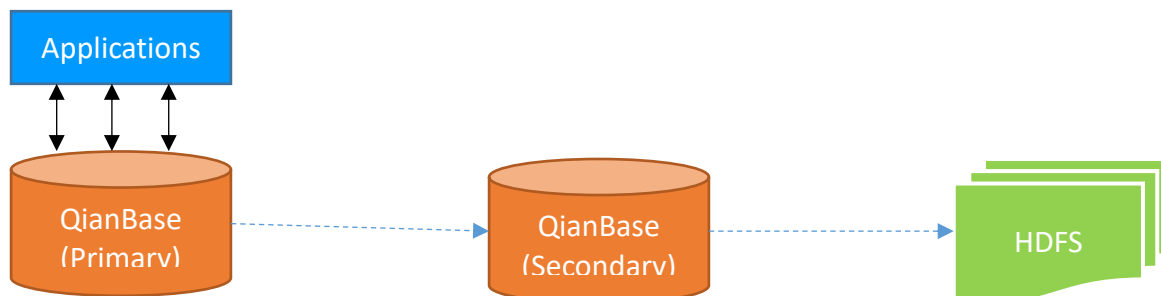
3. 限制条件

存在以下限制：

- 使用时间戳进行还原将还原整个数据库。单个 tables 或 schemas 无法还原。
- 按照现有的 BR 策略，半个小时一次增备并导出,DDL 之后并不需要触发全量备份，除非 CQD TM_SNAPSHOT_TABLE_CREATES 是 0。
- TM_SNAPSHOT_TABLE_CREATES 默认就是 1，这也就意味着，一张增量表，在后续做 DDL 更改的时候，会自动触发一次针对当前表的快照动作。
- mantis 14338：因为某些未知原因（无固定重现步骤），部分备份恢复操作会造成锁残留，并导致创建表/删除表等 DDL 操作报错。一种现象是 Mantis 14338 报告的，在 ZooKeeper 中残留了 Znode。因此过去遇到此问题，前端人员只能通过删除 ZooKeeper 的信息来进行恢复。
- 这种锁残留非常难彻底修复，目前的方案是：增强 get all locked objects 命令，可以返回锁相应的 userTag。然后可以利用 clean backup lock <userTag>命令进行清除。比用 ZooKeeper 的删除节点更加严谨。
- 使用 restore 恢复数据时，报 8102 错误，该错误是由于备份集群中用户创建的顺序如果与源集群不一致造成的，为了避免该错误，请按照固定脚本建立用户，确保备份集群和源集群建立用户的顺序一致。

4. 备份恢复的前提

QianBase 针对常规和增量备份的典型备份恢复策略如下



- 安装 QianBase 的主库和第二备份验证库，主库对第二备份验证库有写入权限。归档集群用于归档在第二备份验证库验证过的有效备份
- 查询访问主库。需要备份的表 (tables) 必须使用 DDL 操作进行标记。
- 全量常规备份首先在主钱库执行，然后定期进行增量备份。这些备份会立即传输到第二备份验证库。然后在第二备份验证库上使用最后的增量备份导入。
- 在第二备份验证库上测试备份数据集后，将有效备份存档在归档系统上。
- 如果主库发生灾难，并且在恢复群集后，检索相应的归档备份并导入主库，然后进行数据恢复。

注意：主库上定义的用户和角色必须存在于第二备份验证库。

5. 数据库可用性

事务可以在任意给定时间访问多个对象，例如 schema(s)和/或 table(s)。执行备份/还原涉及对特定受影响的元数据对象和用户表对象进行简要内部锁定，以实现事务一致性备份。在此操作期间

- 访问不受备份/恢复影响的 schema(s)/table(s)的查询将继续其运行不受影响.
- 只读类型的查询访问正在进行备份操作的 schema(s)/table(s)继续其运行不受影响。但是，这些查询在数据还原操作时会收到可重试的错误。
- SELECT, UPDATE 或 DELETE, 更新或删除正在进行备份操作的 schema(s)/table(s) 在内部重试三次，仍然失败再返回错误.

6. 备份操作类型和命令

备份操作类型

备份操作可以执行全量常规备份或增量备份。

常规或增量备份可以包括指定 schema(s)/table(s)

备份操作类型	说明
全量常规备份	保存特定 schema(s)/table(s)的完整副本
增量备份	仅保存自上次备份后更改的对象增量数据的副本

操作命令

支持备份和恢复的各种命令可以大致分类如下。

功能	说明
备份 (Backup)	将元数据/用户对象保存在系统中。 如果对象包括所有 Schema，则 QianBase 执行全量备份。 您可以执行全量、增量或系统备份。
恢复 (Restore)	从备份中恢复元数据/用户对象。还原可以是一个时间点（“还原到时间戳”），也可以是常规或增量备份（“还原到标记”）。 使用“还原到标记”操作时，仅恢复备份对象中的一个子集。 使用“还原到时间戳记”操作时，将还原整个数据库。单个 tables 或 schemas 无法还原。
查看备份 (Get Backup)	查看储存的备份信息。
删除备份 (灾难恢复 Drop Backup)	删除储存的备份。
导出备份 (Export Backup)	将备份导出至远程存储系统。 如果导出增量备份，则仅导出在前一次导出备份之后的更改（增量变化）。
导入备份 (Import Backup)	将远程存储系统中的备份导入至 QianBase。

6. 备份操作类型和命令

访问规则

备份和恢复

- 数据库 root 用户 (DB__ROOT) , 已被授予 DB__ROOTROLE 角色的任何人或已授予 MANAGE_LOAD 组件权限的任何人都可以备份或还原 schema/table。
- 如果您是对象的所有者, 或者您已获得对象的 SELECT 和 INSERT 权限, 则可以备份 schema/table。
- 如果您是对象的所有者, 或者在备份操作时为每个对象授予了 SELECT 和 INSERT 权限, 则可以还原 schema/table。

获取备份信息

- 任何人都可以执行获取备份语句。

导入, 导出和删除备份命令:

- 数据库 root 用户 (DB__ROOT) , 已被授予 DB__ROOTROLE 角色的任何人或已授予 MANAGE_LOAD 组件权限的任何人都可以执行导入, 导出或删除命令。

7. 使用策略

这是在主库上设置备份操作的典型工作流程。根据在给定时间间隔内累积的已更改数据量，我们建议在多次增量备份后进行一次全量常规备份。在第二备份验证库上执行增量备份的还原操作时，该对象的全量常规备份需首先被恢复，紧接着就是增量备份操作中捕获的所有增量更改将被回播。因此，从增量备份还原对象需要的时间与自上次对象的全量常规备份以来累积的已更改数据量成比例。

与恢复增量备份相比，恢复全量常规备份所需的时间要少得多。根据运行灾难恢复集群的时间，确定全量常规备份的频率非常重要。在全量常规备份间隔之间安排的增量备份可决定捕获更改数据的频率，此频率是为在灾难恢复群集上的最相近的恢复准备的。

对于备份恢复解决方案，备份策略可以是将灾难恢复集群同步与主集群紧密匹配的最新可用备份。这需要灾难恢复群集访问最新的备份，该备份可以是全量常规备份类型或增量备份类型。

以下操作序列定期在主群集上执行。

备份的导出操作可以与群集中的其他活动同时发生。



1. 全量常规备份，然后将备份导出到 HDFS 归档位置。
2. 增量备份，然后将增量备份导出到 HDFS 归档位置。
3. 在给定时间段及时间间隔定期重复步骤 2。
4. 几次增量备份后进行全量常规备份。将全量常规备份导出到灾难恢复 HDFS 归档位置。
5. 定期重复步骤 2 和 3。
6. 完成两个全量常规备份后，第三个全量常规备份及其后续增量备份集将开始替换第一个全量常规备份和增量备份集。步骤 6 非常重要，可以确保没有太多备份占用空间。在替换之前确认导出第一个备份也很重要。

当启动灾难恢复群集时，以下操作序列会开始执行。

1. 导入 HDFS 归档位置中可用的最新备份。
2. 恢复刚刚导入的备份。

在下面的示例中，第 1 周全量常规备份在星期日执行，然后在每个工作日执行增量备份。备份完成后，将立即被导出到远程灾难恢复集群。同样，对于第 2 周，执行全量常规备份，并执行每日增量备份。第 1 周备份（全量常规备份和增量备份）可以被删除，并被在主群集和灾难恢复群集上生成的第 3 周备份取代。

7. 使用策略

	备份时间 Day of week	备份操作种类 Backup Operation	数据依赖 Data Dependency
第一周	星期日	全量	
	星期一	增量	↓
	星期二	增量	↓
	星期三	增量	↓
	星期四	增量	↓
	星期五	增量	↓
	星期六	增量	↓
第二周	星期日	全量	
	星期一	增量	↓
	星期二	增量	↓
	星期三	增量	↓
	星期四	增量	↓
	星期五	增量	↓
	星期六	增量	↓

在第一周的星期五，如果灾难恢复群集需要使用最近的备份（第一周的星期四），则恢复主要涉及先恢复第一周星期日全量常规备份，然后是第1周星期一，星期二，星期三，星期四中捕获的所有更改数据。

管理员必须确定需要保留多少备份，并且必须删除旧备份以管理可用磁盘空间。在删除旧备份之前，必须确认这些备份是否已成功导出到灾难恢复群集。

接下来的几节将介绍用于预创建备份标记，执行备份命令，将备份导出到远程位置，删除主群集和灾难恢复群集上的旧备份以及最终在灾难恢复群集需要准备就绪时还原备份的实际步骤和命令。

8. 备份, 导出和标签重用

- 预创建备份标记之后, 首先进行全量常规备份(如在本例中, 它是第一周的周日), 然后是工作日的多个增量备份。假设一个名为“my_schema”的 schema 需要备份恢复支持。

```
>>create schema my_schema incremental backup;
```

```
--- SQL operation complete.
```

```
>>create table my_schema.t1 ( a int, b int) attribute
incremental backup;
```

```
--- SQL operation complete.
```

- Week1_Sunday 全量常规备份。覆盖关键字确保重复使用¹“week1_sunday”标记。

```
>> insert into my_schema.t1 values(1,1), (1,2);
```

```
--- 2 row(s) inserted.
```

```
>> backup trafodion, tag 'week1_sunday', schemas(my_schema),
override;
```

```
--- SQL operation complete.
```

- 列出系统上的当前备份。

```
>> get all backup snapshots;
```

```
BackupTag      BackupTime      BackupStatus    BackupOperation
=====
==
```

```
week1_sunday   2019-01-06:21:14:25  VALID           REGULAR
```

```
--- SQL operation complete.
```

- 导出备份到备份恢复系统

```
>> export backup to location
'hdfs://nap015.esgyn.local:8020/tmp/backups', tag 'week1_sunday';
```

```
--- SQL operation complete.
```

- 随后, 创建新的增量备份, 请注意使用 `incremental` 和 `override` 关键字来指示 schema 的增量备份。

```
>> insert into my_schema.t1 values(1,3), (1,4);
```

```
--- 2 row(s) inserted.
```

¹ 我们建议在有两组完整备份时重新使用该标记。常规备份上的覆盖会使所有增量备份无效, 因此, 除非有第二组增量备份, 否则将没有可用的备份。

8. 备份, 导出和标签重用

```
>> backup trafodion, tag 'week1_monday', schemas(my_schema),
incremental, override;
```

```
--- SQL operation complete.
```

```
>> get all backup snapshots;
```

```
BackupTag      BackupTime      BackupStatus    BackupOperation
=====
week1_sunday   2019-01-06:21:14:25  VALID          REGULAR
week1_monday   2019-01-06:21:17:10  VALID          INCREMENTAL
```

```
--- SQL operation complete.
```

- 将此备份导出到灾难恢复系统。注意每个导出的导出位置保持不变。

```
>> export backup to location
'hdfs://nap015.esgyn.local:8020/tmp/backups', tag 'week1_monday';
```

```
--- SQL operation complete.
```

- 对于剩余的工作日重复此步骤。随后, 第2周周日执行全量常规备份, 随后是第2周工作日执行增量备份。

```
>> insert into my_schema.t1 values(1,5), (1,6);
```

```
--- 2 row(s) inserted.
```

```
>> backup trafodion, tag 'week1_tuesday', schemas(my_schema),
incremental, override;
```

```
--- SQL operation complete.
```

```
>> get all backup snapshots;
```

```
BackupTag      BackupTime      BackupStatus    BackupOperation
=====
week1_sunday   2019-01-06:21:14:25  VALID          REGULAR
week1_monday   2019-01-06:21:17:10  VALID          INCREMENTAL
week1_tuesday  2019-01-06:21:19:46  VALID          INCREMENTAL
```

```
--- SQL operation complete.
```

```
>> export backup to location
'hdfs://nap015.esgyn.local:8020/tmp/backups', tag 'week1_tuesday';
```

```
--- SQL operation complete.
```

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

```
>> insert into my_schema.t1 values(2,1), (2,2);
```

8. 备份, 导出和标签重用

--- 2 row(s) inserted.

```
>> backup trafodion, tag 'week2_sunday', schemas(my_schema),
override;
```

--- SQL operation complete.

```
>> get all backup snapshots;
```

BackupTag	BackupTime	BackupStatus	BackupOperation
week1_sunday	2019-01-06:21:14:25	VALID	REGULAR
week1_monday	2019-01-06:21:17:10	VALID	INCREMENTAL
week1_tuesday	2019-01-06:21:19:46	VALID	INCREMENTAL
week1_wednesday	2019-01-06:21:22:16	VALID	INCREMENTAL
week1_thursday	2019-01-06:21:24:46	VALID	INCREMENTAL
week1_friday	2019-01-06:21:27:22	VALID	INCREMENTAL
week1_saturday	2019-01-06:21:29:55	VALID	INCREMENTAL
week2_sunday	2019-01-06:21:32:28	VALID	REGULAR

--- SQL operation complete.

```
>> export backup to location
'hdfs://nap015.esgyn.local:8020/tmp/backups', tag 'week2_sunday';
```

--- SQL operation complete.

- 一旦导出第2周备份后, 就要重新使用第1周标记了, 并且会删除灾难恢复群集上的主群集和备份位置上的第1周备份。

```
>> get all backup snapshots;
```

BackupTag	BackupTime	BackupStatus	BackupOperation
week1_sunday	2019-01-06:21:14:25	VALID	REGULAR
week1_monday	2019-01-06:21:17:10	VALID	INCREMENTAL
week1_tuesday	2019-01-06:21:19:46	VALID	INCREMENTAL
week1_wednesday	2019-01-06:21:22:16	VALID	INCREMENTAL
week1_thursday	2019-01-06:21:24:46	VALID	INCREMENTAL
week1_friday	2019-01-06:21:27:22	VALID	INCREMENTAL
week1_saturday	2019-01-06:21:29:55	VALID	INCREMENTAL
week2_sunday	2019-01-06:21:32:28	VALID	REGULAR
week2_monday	2019-01-06:21:35:19	VALID	INCREMENTAL
week2_tuesday	2019-01-06:21:37:52	VALID	INCREMENTAL
week2_wednesday	2019-01-06:21:40:31	VALID	INCREMENTAL
week2_thursday	2019-01-06:21:43:07	VALID	INCREMENTAL
week2_friday	2019-01-06:21:45:34	VALID	INCREMENTAL
week2_saturday	2019-01-06:21:48:04	VALID	INCREMENTAL

--- SQL operation complete.

```
>>drop all backup snapshots, match 'week1%';
```

--- SQL operation complete.

```
>>get all backup snapshots;
```

8. 备份, 导出和标签重用

```
BackupTag          BackupTime          BackupStatus      BackupOperation
=====
=====
week2_sunday       2019-01-06:21:32:28  VALID             REGULAR
week2_monday       2019-01-06:21:35:19  VALID             INCREMENTAL
week2_tuesday      2019-01-06:21:37:52  VALID             INCREMENTAL
week2_wednesday    2019-01-06:21:40:31  VALID             INCREMENTAL
week2_thursday     2019-01-06:21:43:07  VALID             INCREMENTAL
week2_friday       2019-01-06:21:45:34  VALID             INCREMENTAL
week2_saturday     2019-01-06:21:48:04  VALID             INCREMENTAL
```

--- SQL operation complete.

```
>>get all backup tags, show details;
```

```
BackupTag          BackupTime          BackupStatus      BackupOperation
=====
=====
week1_friday       2019-01-06:21:26:52  NO_SNAPSHOT      UNKNOWN
week1_monday       2019-01-06:21:16:39  NO_SNAPSHOT      UNKNOWN
week1_saturday     2019-01-06:21:29:29  NO_SNAPSHOT      UNKNOWN
week1_sunday       2019-01-06:21:13:51  NO_SNAPSHOT      UNKNOWN
week1_thursday     2019-01-06:21:24:18  NO_SNAPSHOT      UNKNOWN
week1_tuesday      2019-01-06:21:19:15  NO_SNAPSHOT      UNKNOWN
week1_wednesday    2019-01-06:21:21:44  NO_SNAPSHOT      UNKNOWN
week2_friday       2019-01-06:21:45:06  VALID             INCREMENTAL
week2_monday       2019-01-06:21:34:49  VALID             INCREMENTAL
week2_saturday     2019-01-06:21:47:30  VALID             INCREMENTAL
week2_sunday       2019-01-06:21:31:56  VALID             REGULAR
week2_thursday     2019-01-06:21:42:37  VALID             INCREMENTAL
week2_tuesday      2019-01-06:21:37:22  VALID             INCREMENTAL
week2_wednesday    2019-01-06:21:40:01  VALID             INCREMENTAL
```

--- SQL operation complete.

- 在灾难恢复群集上, 第1周备份过时。这需要从HDFS位置手动删除。

```
>> hdfs dfs -rm -r -skipTrash
/user/trafodion/backups/week1_*
$ hdfs dfs -ls hdfs://nap015.esgyn.local:8020/tmp/backups
Found 15 items
drwxr-xr-x - centos supergroup          0 2019-01-06 21:50
hdfs://nap015.esgyn.local:8020/tmp/backups/TRAF_1500000_TRAFODION.MY_
SCHEMA.T1
drwxr-xr-x - centos supergroup          0 2019-01-06 21:29
hdfs://nap015.esgyn.local:8020/tmp/backups/week1_friday
drwxr-xr-x - centos supergroup          0 2019-01-06 21:19
hdfs://nap015.esgyn.local:8020/tmp/backups/week1_monday
drwxr-xr-x - centos supergroup          0 2019-01-06 21:32
hdfs://nap015.esgyn.local:8020/tmp/backups/week1_saturday
drwxr-xr-x - centos supergroup          0 2019-01-06 21:17
hdfs://nap015.esgyn.local:8020/tmp/backups/week1_sunday
drwxr-xr-x - centos supergroup          0 2019-01-06 21:27
hdfs://nap015.esgyn.local:8020/tmp/backups/week1_thursday
drwxr-xr-x - centos supergroup          0 2019-01-06 21:22
hdfs://nap015.esgyn.local:8020/tmp/backups/week1_tuesday
drwxr-xr-x - centos supergroup          0 2019-01-06 21:24
hdfs://nap015.esgyn.local:8020/tmp/backups/week1_wednesday
drwxr-xr-x - centos supergroup          0 2019-01-06 21:48
```

8. 备份, 导出和标签重用

```
hdfs://nap015.esgyn.local:8020/tmp/backups/week2_friday
drwxr-xr-x - centos supergroup 0 2019-01-06 21:37
hdfs://nap015.esgyn.local:8020/tmp/backups/week2_monday
drwxr-xr-x - centos supergroup 0 2019-01-06 21:50
hdfs://nap015.esgyn.local:8020/tmp/backups/week2_saturday
drwxr-xr-x - centos supergroup 0 2019-01-06 21:35
hdfs://nap015.esgyn.local:8020/tmp/backups/week2_sunday
drwxr-xr-x - centos supergroup 0 2019-01-06 21:45
hdfs://nap015.esgyn.local:8020/tmp/backups/week2_thursday
drwxr-xr-x - centos supergroup 0 2019-01-06 21:40
hdfs://nap015.esgyn.local:8020/tmp/backups/week2_tuesday
drwxr-xr-x - centos supergroup 0 2019-01-06 21:43
hdfs://nap015.esgyn.local:8020/tmp/backups/week2_wednesday
```

```
$hdfs dfs -rm -r -skipTrash
hdfs://nap015.esgyn.local:8020/tmp/backups/week1*
Deleted hdfs://nap015.esgyn.local:8020/tmp/backups/week1_friday
Deleted hdfs://nap015.esgyn.local:8020/tmp/backups/week1_monday
Deleted hdfs://nap015.esgyn.local:8020/tmp/backups/week1_saturday
Deleted hdfs://nap015.esgyn.local:8020/tmp/backups/week1_sunday
Deleted hdfs://nap015.esgyn.local:8020/tmp/backups/week1_thursday
Deleted hdfs://nap015.esgyn.local:8020/tmp/backups/week1_tuesday
Deleted hdfs://nap015.esgyn.local:8020/tmp/backups/week1_wednesday
```

- 现在重新使用 week1 标记为下一组备份。

```
>>insert into my_schema.t1 values(1,15),(1,16);
```

```
--- 2 row(s) inserted.
```

```
>> backup trafodion, tag 'week1_sunday', schemas(my_schema),
override;
```

```
--- SQL operation complete.
```

```
>>get all backup snapshots;
```

BackupTag	BackupTime	BackupStatus	BackupOperation
week2_sunday	2019-01-06:21:32:28	VALID	REGULAR
week2_monday	2019-01-06:21:35:19	VALID	INCREMENTAL
week2_tuesday	2019-01-06:21:37:52	VALID	INCREMENTAL
week2_wednesday	2019-01-06:21:40:31	VALID	INCREMENTAL
week2_thursday	2019-01-06:21:43:07	VALID	INCREMENTAL
week2_friday	2019-01-06:21:45:34	VALID	INCREMENTAL
week2_saturday	2019-01-06:21:48:04	VALID	INCREMENTAL
week1_sunday	2019-01-07:02:02:44	VALID	REGULAR

```
--- SQL operation complete.
```

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

```
>> insert into my_schema.t1 values(1,27),(1,28);
```

8. 备份, 导出和标签重用

--- 2 row(s) inserted.

```
>> backup trafodion, tag 'week1_saturday', schemas(my_schema),
incremental, override;
```

--- SQL operation complete.

```
>> get all backup snapshots;
```

BackupTag	BackupTime	BackupStatus	BackupOperation
week2_sunday	2019-01-06:21:32:28	VALID	REGULAR
week2_monday	2019-01-06:21:35:19	VALID	INCREMENTAL
week2_tuesday	2019-01-06:21:37:52	VALID	INCREMENTAL
week2_wednesday	2019-01-06:21:40:31	VALID	INCREMENTAL
week2_thursday	2019-01-06:21:43:07	VALID	INCREMENTAL
week2_friday	2019-01-06:21:45:34	VALID	INCREMENTAL
week2_saturday	2019-01-06:21:48:04	VALID	INCREMENTAL
week1_sunday	2019-01-07:02:02:44	VALID	REGULAR
week1_monday	2019-01-07:02:05:29	VALID	INCREMENTAL
week1_tuesday	2019-01-07:02:08:11	VALID	INCREMENTAL
week1_wednesday	2019-01-07:02:10:41	VALID	INCREMENTAL
week1_thursday	2019-01-07:02:13:08	VALID	INCREMENTAL
week1_friday	2019-01-07:02:15:41	VALID	INCREMENTAL
week1_saturday	2019-01-07:02:18:20	VALID	INCREMENTAL

--- SQL operation complete.

```
>> export backup to location
'hdfs://nap015.esgyn.local:8020/tmp/backups', tag 'week1_saturday';
```

--- SQL operation complete.

```
>>
```

```
>>get all backup tags, show details;
```

BackupTag	BackupTime	BackupStatus	BackupOperation
week1_friday	2019-01-07:02:15:17	VALID	INCREMENTAL
week1_monday	2019-01-07:02:05:04	VALID	INCREMENTAL
week1_saturday	2019-01-07:02:17:59	VALID	INCREMENTAL
week1_sunday	2019-01-07:02:02:11	VALID	REGULAR
week1_thursday	2019-01-07:02:12:45	VALID	INCREMENTAL
week1_tuesday	2019-01-07:02:07:48	VALID	INCREMENTAL
week1_wednesday	2019-01-07:02:10:18	VALID	INCREMENTAL
week2_friday	2019-01-06:21:45:06	VALID	INCREMENTAL
week2_monday	2019-01-06:21:34:49	VALID	INCREMENTAL
week2_saturday	2019-01-06:21:47:30	VALID	INCREMENTAL
week2_sunday	2019-01-06:21:31:56	VALID	REGULAR
week2_thursday	2019-01-06:21:42:37	VALID	INCREMENTAL
week2_tuesday	2019-01-06:21:37:22	VALID	INCREMENTAL
week2_wednesday	2019-01-06:21:40:01	VALID	INCREMENTAL

--- SQL operation complete.

```
>>
```

8. 备份, 导出和标签重用

- 一旦重用与 week2 相关的标签, 重复与第 1 周相同的步骤, 以删除主群集上的第 2 周备份和与灾难恢复群集上的第 2 周标签相关的旧副本。

9. 预创建备份标记

续使用“[使用策略](#)”部分中概述的示例，预先创建以下备份标记以唯一地表示备份。这些标记在主群集上创建。

第2周的备份完成后，将重复使用 Week1 标记。Week1 标签的备份完成后，将重复使用 Week2 标签。

Week1_Sunday

Week1_Monday

...

...

Week2_Saturday

```
>>backup trafodion, create tags(
+>'week1_sunday', 'week1_monday', 'week1_tuesday',
+> 'week1_wednesday', 'week1_thursday', 'week1_friday',
+>'week1_saturday', 'week2_sunday', 'week2_monday',
'week2_tuesday',
+>'week2_wednesday', 'week2_thursday', 'week2_friday',
+>'week2_saturday');
```

--- SQL operation complete.

>>

```
>>get all backup tags;
```

```
BackupTag          IsPrefix  NumBackups
=====
```

week1_friday	NO	1
week1_monday	NO	1
week1_saturday	NO	1
week1_sunday	NO	1
week1_thursday	NO	1
week1_tuesday	NO	1
week1_wednesday	NO	1
week2_friday	NO	1
week2_monday	NO	1
week2_saturday	NO	1
week2_sunday	NO	1
week2_thursday	NO	1
week2_tuesday	NO	1
week2_wednesday	NO	1

--- SQL operation complete.

10. 灾难恢复集群的导入, 恢复

使用最新备份启动灾难恢复群集, 这涉及从备份位置导入最新备份, 然后还原备份。在灾难恢复群集上, 执行以下操作。

假设灾难恢复群集需要准备好最新的备份标记 “week2_thursday”。

- 从灾难恢复群集上的 hdfs shell 列出导出位置中可用的备份标记。

```
$hdfs dfs -ls hdfs://nap015.esgyn.local:8020/tmp/backups
Found 15 items
drwxr-xr-x - centos supergroup          0 2019-01-07 02:21
hdfs://nap015.esgyn.local:8020/tmp/backups/TRAF_1500000_TRAFODION.MY_
SCHEMA.T1
drwxr-xr-x - centos supergroup          0 2019-01-07 02:18
hdfs://nap015.esgyn.local:8020/tmp/backups/week1_friday
drwxr-xr-x - centos supergroup          0 2019-01-07 02:08
hdfs://nap015.esgyn.local:8020/tmp/backups/week1_monday
drwxr-xr-x - centos supergroup          0 2019-01-07 02:20
hdfs://nap015.esgyn.local:8020/tmp/backups/week1_saturday
drwxr-xr-x - centos supergroup          0 2019-01-07 02:05
hdfs://nap015.esgyn.local:8020/tmp/backups/week1_sunday
drwxr-xr-x - centos supergroup          0 2019-01-07 02:15
hdfs://nap015.esgyn.local:8020/tmp/backups/week1_thursday
drwxr-xr-x - centos supergroup          0 2019-01-07 02:10
hdfs://nap015.esgyn.local:8020/tmp/backups/week1_tuesday
drwxr-xr-x - centos supergroup          0 2019-01-07 02:13
hdfs://nap015.esgyn.local:8020/tmp/backups/week1_wednesday
drwxr-xr-x - centos supergroup          0 2019-01-06 21:48
hdfs://nap015.esgyn.local:8020/tmp/backups/week2_friday
drwxr-xr-x - centos supergroup          0 2019-01-06 21:37
hdfs://nap015.esgyn.local:8020/tmp/backups/week2_monday
drwxr-xr-x - centos supergroup          0 2019-01-06 21:50
hdfs://nap015.esgyn.local:8020/tmp/backups/week2_saturday
drwxr-xr-x - centos supergroup          0 2019-01-06 21:35
hdfs://nap015.esgyn.local:8020/tmp/backups/week2_sunday
drwxr-xr-x - centos supergroup          0 2019-01-06 21:45
hdfs://nap015.esgyn.local:8020/tmp/backups/week2_thursday
drwxr-xr-x - centos supergroup          0 2019-01-06 21:40
hdfs://nap015.esgyn.local:8020/tmp/backups/week2_tuesday
drwxr-xr-x - centos supergroup          0 2019-01-06 21:43
hdfs://nap015.esgyn.local:8020/tmp/backups/week2_wednesday
[centos@vasudevp4 scripts]$ sqlci
```

- 导入 “week2_thursday” 备份标记

```
>>get all backup tags, show details;

--- SQL operation complete.

>>import backup from location
'hdfs://nap015.esgyn.local:8020/tmp/backups', tag 'week2_thursday';

--- SQL operation complete.

>>get all backup tags, show details;
```

10. 灾难恢复集群的导入, 恢复

```
BackupTag          BackupTime          BackupStatus  BackupOperation
=====
===

week2_thursday          NO_METADATA
INCREMENTAL (IMPORTED)

--- SQL operation complete.

>>get all backup snapshots;

BackupTag          BackupTime          BackupStatus  BackupOperation
=====
====

week2_thursday    2019-01-06:21:43:07  VALID
INCREMENTAL (IMPORTED)

--- SQL operation complete.
```

如果 trafodion 用户 ID 被授予 hbase sudo 访问权限, 那么来自 trafci 提示的上述 import 命令是适用的。但是, 如果无法访问 sudo, 则指定的灾难恢复群集升级用户 (具有 hbase sudo 访问权限的升级用户或 hbase 用户) 可以执行相同的导入操作。有关非 trafodion 用户的等效导入命令, 请参阅第 13 节。

- 从 trafci 执行恢复备份。使用 show objects 显示备份对象关键字, 去掉 show objects, 真正执行还原操作。

```
>>restore trafodion, tag 'week2_thursday', show objects;

MetaData objects to be restored
=====

TRAF_RSRVD_3:TRAFODION._BACKUP_week2_thursday_.COLUMNS
TRAF_RSRVD_3:TRAFODION._BACKUP_week2_thursday_.COLUMN_PRIVILEGES
TRAF_RSRVD_3:TRAFODION._BACKUP_week2_thursday_.INDEXES
TRAF_RSRVD_3:TRAFODION._BACKUP_week2_thursday_.KEYS
TRAF_RSRVD_3:TRAFODION._BACKUP_week2_thursday_.OBJECTS
TRAF_RSRVD_3:TRAFODION._BACKUP_week2_thursday_.OBJECT_PRIVILEGES
TRAF_RSRVD_3:TRAFODION._BACKUP_week2_thursday_.REF_CONSTRAINTS
TRAF_RSRVD_3:TRAFODION._BACKUP_week2_thursday_.SB_HISTOGRAMS
TRAF_RSRVD_3:TRAFODION._BACKUP_week2_thursday_.SB_HISTOGRAM_INTERVALS
TRAF_RSRVD_3:TRAFODION._BACKUP_week2_thursday_.SB_PERSISTENT_SAMPLES
TRAF_RSRVD_3:TRAFODION._BACKUP_week2_thursday_.SCHEMA_PRIVILEGES
TRAF_RSRVD_3:TRAFODION._BACKUP_week2_thursday_.SEQ_GEN
TRAF_RSRVD_3:TRAFODION._BACKUP_week2_thursday_.TABLES
TRAF_RSRVD_3:TRAFODION._BACKUP_week2_thursday_.TABLE_CONSTRAINTS
TRAF_RSRVD_3:TRAFODION._BACKUP_week2_thursday_.TEXT
TRAF_RSRVD_3:TRAFODION._BACKUP_week2_thursday_.UNIQUE_REF_CONSTR_USAG
E
TRAF_RSRVD_3:TRAFODION._BACKUP_week2_thursday_.VIEWS
TRAF_RSRVD_3:TRAFODION._BACKUP_week2_thursday_.VIEWS_USAGE

User objects to be restored
=====

TRAF_1500000:TRAFODION.MY_SCHEMA.SB_HISTOGRAMS
TRAF_1500000:TRAFODION.MY_SCHEMA.SB_HISTOGRAM_INTERVALS
```

10. 灾难恢复集群的导入, 恢复

```
TRAF_1500000:TRAFODION.MY_SCHEMA.SB_PERSISTENT_SAMPLES
TRAF_1500000:TRAFODION.MY_SCHEMA.T1
```

```
--- SQL operation complete.
```

```
>>restore trafodion, tag 'week2_thursday';
```

```
--- SQL operation complete.
```

```
>>get schemas;
```

```
Schemas in Catalog TRAFODION
=====
```

```
MY_SCHEMA
SCH
SEABASE
_BACKUP_week2_thursday_
_LIBMGR_
_MD_
_PRIVMGR_MD_
_REPOS_
_TENANT_MD_
```

```
--- SQL operation complete.
```

```
>>set schema my_schema;
```

```
--- SQL operation complete.
```

```
>>get tables;
```

```
Tables in Schema TRAFODION.MY_SCHEMA
=====
```

```
SB_HISTOGRAMS
SB_HISTOGRAM_INTERVALS
SB_PERSISTENT_SAMPLES
T1
```

```
--- SQL operation complete.
```

```
>>select * from t1;
```

```
A          B
-----
1          1
1          2
1          3
1          4
1          5
1          6
1          7
1          8
1          9
1         10
1         11
1         12
1         13
1         14
```

10. 灾难恢复集群的导入, 恢复

2	1
2	2
2	3
2	4
2	5
2	6
2	7
2	8
2	9
2	10

--- 24 row(s) selected.

此时, 灾难恢复集群就可以启用了。

11. 非 trafodion 用户，导入备份到灾难恢复集群

- 切换到 hbase 用户并更新 CLASSPATH 作为初始设置

```
>sudo su hbase // switch to hbase user
```

```
>export
```

```
CLASSPATH=/opt/cloudera/parcels/ESGYNDB_TRX/lib/5.7/hbase  
-trx-cdh5_7-2.4.9.jar:`hbase classpath`
```

基本上，包括 trx jar。其他所有内容都包含在 hbase 类路径中。

请注意`hbase classpath`周围的引号。它是向前引号，而不是常规引号。

- 用 hbase 用户导入备份标记“week2_thursday”。

```
>>java
```

```
org.apache.hadoop.hbase.pit.Import import week2_thursd  
ay hdfs://<DR IP>/user/trafodion/backups imp_hbase 4
```

where

```
org.apache.hadoop.hbase.pit.Import //类名
```

```
import //导入命令
```

```
week2_thursday //包名
```

```
hdfs:// //存储备份位置的 hdfs url
```

```
imp_hbase //表示以 hbase 用户身份运行此命令。
```

如果从具有 sudo 访问权限的任何用户运行此命令，则使用的选项是“imp_sudoibase”

4 is indicating number of parallel threads. //这通常是群集中的数据节点数

12. 语法语义

这些命令通过钱库命令行界面 trafci 启动。

备份和恢复

备份和恢复 QianBase 数据库对象的基本操作。

BACKUP TRAFODION { <br-options> }

RESTORE TRAFODION { <br-options> }

br-options: [,br-option1 ,br-option]

下面讨论的 br-options 可以按任何顺序指定。选项需要用逗号 (',') 分隔。有些是必需的, 有些是非必需的。

语法项 br-options	语义/说明
TAG '<tag>'	必需的识别备份的 TAG, <i>tag-name</i> 最多 32 个字符。
SCHEMA (<sch>)	备份/恢复 Schema <i>schema-name</i> 的所有对象。
TO TIMESTAMP '<ts>'	仅适用于还原操作。使用格式 YYYY-MM-DD HH:MM:SS 指定时间戳
SCHEMAS(<sch1>,, <schN>)	备份/恢复指定 Schema 中的所有对象。
TABLE (<table>)	备份/恢复表 <i>table-name</i> 。
TABLES(<table1>,<tableN>)	备份/恢复多个表 (<i>table-name1, ...table-nameN</i>) 。
INCREMENTAL	对具有 incremental 属性的指定对象执行增量 (INCREMENTAL) 备份。否则, 默认情况下它是定期 (REGULAR) 备份
SHOW OBJECTS	仅显示将被备份/恢复的对象, 实际上并不执行备份/恢复操作。
OVERRIDE	重用现有标签。删除具有相同标记的先前备份。标签未附加时间戳, 因此可以重复使用标签。这不适用于还原操作。

可以在一个命令中指定多个 schema/table。将备份或还原该 schema/table 中的所有对象。

如果 backup 命令中未指定 schema/table, 则执行数据库全量备份。这包括数

12. 语法语义

数据库中的所有 schema/table。

在还原期间，如果未指定 schema/table，则还原使用指定标记备份的所有对象。

必须谨慎使用 **OVERRIDE** 选项。此选项在内部删除具有指定标记的现有备份，然后创建具有相同标记的新备份。与增量备份一起使用时，建议始终执行定期全量常规备份操作作为初始备份。随后可以执行使用覆盖选项的增量备份。如果未执行全量常规备份操作，则使用覆盖选项的增量备份操作将永远不会捕获增量数据。取而代之，捕获的是一份完整的 table。

为备份指定的标记始终附加唯一的时间戳标识符。这有助于调用备份命令的调度程序作业每次都使用相同的标记名称。但是，使用覆盖选项，将覆盖附加唯一时间戳标识符的功能。

创建增量属性 schema 和 table

对具有“**INCREMENTAL**”属性的对象执行增量备份。此属性在 schema 或 table 上设置。

在具有 incremental 属性的 schema 中创建的 tables 会自动继承 incremental 属性。

```
CREATE TABLE ... ATTRIBUTE INCREMENTAL BACKUP;
```

```
CREATE SCHEMA ... INCREMENTAL BACKUP;
```

没有 incremental 属性的对象会被当作 regular 对象处理。

删除备份

Drop 命令用于从本地系统中删除备份。这些命令中使用的标记必须是具有附加时间戳标识符的完全限定标记。请使用 **Get** 命令列出备份标记。

在增量备份对之前执行的全量常规备份非常重要，作为增量备份的一部分捕获的对象的更改数据只有在增量备份之前有相同对象的全量常规备份时才有效。如果没有全量常规备份，则对象的第一个增量备份会被系统视为全量常规备份。

如果存在依赖于全量常规备份的增量备份，删除该全量常规备份会操作失败并报错。必须先删除相关增量备份，然后才能删除全量常规备份。如果增量备份不是初始备份，则可以将其删除。

```
DROP ALL BACKUP TAGS [ , MATCH '%match string%']
```

将会删除当前系统所有备份。

12. 语法语义

可选的 MATCH 子句仅适用于匹配的通配符标记字符串。

使用 DROP BACKUP TAG, TAG 'tag'

将从本地系统删除具有指定“标记”的备份。

查看备份

此命令用于查看系统上记录的所有可用备份的状态。

GET ALL BACKUP TAGS [, SHOW DETAILS] [, MATCH '%match string%']

将返回有关系统上所有备份标记的信息。

可选的 SHOW DETAILS 子句将返回与备份时间戳，有效性和备份类型（如全量常规备份或增量备份）相关的其他信息。

可选的 MATCH 子句将结果过滤到匹配的通配符标记字符串。

导出/导入备份

此命令可以将备份数据在数据库系统和 HDFS 位置复制并相互转移。此 HDFS 位置必须是主机名限定的 HDFS 目录。主机名可以是本地群集或任何远程群集。

EXPORT BACKUP TO LOCATION '<backup root location>', TAG '<tag>'
将 TAG 标识的备份复制到指定的 HDFS 备份根位置（backup root location）。

IMPORT BACKUP FROM LOCATION '<backup root location>', TAG '<tag>'
将 TAG 标识的备份从 HDFS 位置复制到本地存储系统。

EXPORT BACKUP TO LOCATION '<backup root location>', TAG '<tag>',
override;

Override，当第一次导出失败时，再次导出时，可覆盖上次导出的部分文件，否则需要手动清理。

备份根位置（backup root location）：所有导出备份的指向基本目录（base directory）的 HDFS 位置。

tag：备份标识符。请使用 Get 命令获取备份标识符。

HDFS 位置是完全合格的备份目录。此备份目录必须是导出多个备份的基本目录（base directory）。导出增量备份要求与之前导出的增量备份相同的基本目录。如果更改了基本目录，则导出增量备份有关从备份开始和所有后续增量备份所有数据。

12. 语法规义

查看/清除对象锁

此命令用于查看系统上的被锁的对象列表并清除对象上的锁。

```
GET ALL LOCKED OBJECTS;
```

获取数据库中被锁对象列表

```
CLEANUP BACKUP LOCK,TAG '<tag>';
```

```
CLEANUP SCHEMA "_BACKUP_<tag>_";
```

清除数据库中因备份还原产生的备份锁

注意:

在备份失败遇到“lock”时，也可以进行“cleanup backup,tag '<user-tag>';”但是命令中的 tag <user-tag>实际上会被删除。这意味着为了生成新备份，将删除历史备份。

但在恢复失败产生的“lock”时，建议使用 CLEANUP BACKUP LOCK, TAG '<tag>'和 CLEANUP SCHEMA "_BACKUP_<tag>_",来清除 lock，以防止需要使用的备份被删除。

查看备份恢复进度

此命令用于查看系统上正在执行的备份恢复或导入导出的进度状态。

```
GET PROGRESS STATUS FOR CURRENT <operation_type>
```

operation_type 为以下其中一项：

BACKUP --查看备份进度状态

EXPORT --查看导出进度状态

RESTORE --查看恢复进度状态

IMPORT --查看导入进度状态

例子

```
14:56:24 SQL>get progress status for current backup;
```

```
BACKUP Progress Status for tag "m13544_0103_00212444772745376894"
```

```
=====
```

Step	State	ET	Progress
=====	=====	=====	=====
Initialize	Completed	00:00:01	100%
CreateMetadata	Completed	00:02:02	100% (22/22)
SetupMetadata	Completed	00:00:13	100% (4/4)
BackupObjects	Completed	00:00:31	100% (35/35)
Finalize	Completed	00:00:01	100%

12. 语法语义

BACKUP End Status for tag "m13544_0103_00212444772745376894"
=====

StartTime: 2020-01-03 08:52:38 (local time)
EndTime: 2020-01-03 08:55:27 (local time)
ET: 00:02:49 (hh:mm:ss)

--- SQL operation complete.

14:56:45 SQL>get progress status for current restore;

RESTORE Progress Status for tag "m13554_00212444563266465523"
=====

Step	State	ET	Progress
=====	=====	=====	=====
Initialize	Completed	00:00:00	100%
CreateMetadata	Completed	00:02:46	100% (22/22)
RestoreObjects	Completed	00:02:42	100% (35/35)
Finalize	Completed	00:00:05	100%

RESTORE End Status for tag "m13554_00212444563266465523"
=====

StartTime: 2019-12-31 23:23:17 (local time)
EndTime: 2019-12-31 23:28:52 (local time)
ET: 00:05:35 (hh:mm:ss)

--- SQL operation complete.

14:57:15 SQL>get progress status for current import;

IMPORT Progress Status for tag "m13554_00212444563266465523"
=====

Step	State	ET	Progress
=====	=====	=====	=====
Initialize	Completed	00:00:00	100%
ImportObjects	Initiated	00:00:00	0%

--- SQL operation complete.

14:57:19 SQL>get progress status for current export;

EXPORT Progress Status for tag "m13544_0103_00212444772745376894"
=====

Step	State	ET	Progress
=====	=====	=====	=====

12. 语法语义

```
Initialize          Completed    00:00:00  100%
ExportObjects      Completed    00:13:08  100%(35/35)
Finalize           Completed    00:00:00  100%
```

```
EXPORT End Status for tag "m13544_0103_00212444772745376894"
=====
```

```
StartTime: 2020-01-03 08:55:47 (local time)
EndTime:    2020-01-03 09:08:57 (local time)
ET:        00:13:09 (hh:mm:ss)
```

```
--- SQL operation complete.
```

恢复到特定时间点

此命令用于恢复数据库到某一特定时间点:

```
RESTORE TRAFODION,TO TIMESTAMP <TIMESTAMP>;
```

例子:

```
create schema pittest incremental backup;
set schema pittest;
create table tabl (a int,b varchar(20));
insert into pittest.tabl values (1,'insert_c1');
insert into pittest.tabl values (2,'insert_c1');
backup trafodion,tag 'pit1';
insert into pittest.tabl values (3,'insert_c2'); $timestamp 2019-
11-20 16:05:00
insert into pittest.tabl values (4,'insert_c2');

restore trafodion,to timestamp '2019-11-20 16:05:00';

--expect:(1,'insert_c1') (2,'insert_c1') (3,'insert_c2')
>>select * from TRAFODION.PITTEST.TABL ;
A          B
-----
          1  insert_c1
          2  insert_c1
          3  insert_c2

--- 3 row(s) selected.
```

注意事项:

需要特别说明,指定的时间点是需要用 wck 工具在增量文件(即 mutation 文件)中查看到的时间点,而不是我们实际看到的服务器的时间点。

因为服务器的时间点只能精确到秒,在毫秒级差异的情况下会引起恢复的数据不是用户想要的时间点的数据的情况。

13. 备份还原操作的事务性影响

性能

- 由于在备份操作期间使用了额外的磁盘和 CPU 资源，因此建议在维护期间或低活动期间启动它们。请注意，数据库不需要脱机（还原期间除外）
- 备份数据集
性能受实例上存储的快照数量的影响。设置系统定期删除快照。
- 还原备份数据的时间主要取决于还原中涉及的对象数。

存储

备份需要系统上的额外存储空间。数据库管理员应定期监视存储使用情况，并根据其数据保留策略定期删除旧备份。

14. 事务日志

备份还原的事件记录在 `trafodion.sql.java.log` 中。您可以通过 QianBase Manager 监控信息事件。

15. 附录 A 控制查询默认 (CQD) 属性

本附录描述用于覆盖系统默认设置的 CQD。

QianBase 环境 CQD

本小节用来描述备份, 恢复, 导入, 导出的并发数以及备份锁类型设置

1) TRAF_EXPORT_IMPORT_PARALLEL

类别	QianBase
描述	用来控制导入导出并发数
数值	在系统环境承受能力内, 数值越大, 导入导出越快

2) TRAF_BACKUP_PARALLEL

类别	QianBase
描述	用来控制备份并发数
数值	建议不使用, 默认时并发数, 为数据库节点数, 综合性能最优

3) TRAF_BR_LOCKED_OBJECTS_QUERY_TYPE

类别	QianBase
描述	用来控制在备份过程是否阻塞业务
数值	指定以下值之一: <ul style="list-style-type: none"> • 0: 在备份过程中不阻塞备份对象 select, 阻塞 insert/update/delete 和 DDL 操作, 支持对不是备份对象的所有操作 • 1: 在备份过程中不阻塞备份对象 select/insert/update/delete, 阻塞 DDL 操作, 支持对不是备份对象的所有操作