

EsgynDB 备份与恢复指南 2.4.0

版权

© Copyright 2018 Esgyn

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 Esgyn 预先书面许可的情况下，严禁改编或翻译本手册的内容。Esgyn 对于本文中所包含的技术或编辑错误、遗漏概不负责。

Esgyn 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 和 EsgynDB 是 Esgyn 的商标。

目录

前言	iii
关于本指南.....	iii
目标读者.....	iii
相关文档.....	iii
修订历史.....	iii
1. 概览	1
2. 备份与恢复	2
2.1 备份类型.....	2
2.1.1 增量备份	3
2.2 功能.....	5
2.3 访问数据库.....	6
2.4 注意事项.....	7
2.5 使用策略.....	8
2.6 支持.....	9
3. 语法与语义	10
3.1 创建备份/恢复	10
3.1.1 语法	10
3.1.2 语义	11

3.2 创建 Schema/表, 指定增量备份属性.....	13
3.2.1 语法.....	13
3.3 删除备份.....	15
3.3.1 语法.....	15
3.3.2 语义.....	15
3.4 查看备份.....	16
3.4.1 语法.....	16
3.4.2 语义.....	16
3.5 导出/导入备份.....	17
3.5.1 语法.....	17
3.5.2 语义.....	17
4. 示例.....	19
5. 备份与恢复的影响.....	51
5.1 性能.....	51
5.2 存储.....	51
6. 事件日志.....	52

前言

关于本指南

本指南介绍 EsgynDB 如何备份和恢复数据库对象。

目标读者

本指南的目标读者为 EsgynDB 系统管理员和用户。

相关文档

《EsgynDB 产品指南》	提供 EsgynDB 产品概览，包括 EsgynDB 独有的概念、程序和参考信息。
《EsgynDB Manager 用户指南》	EsgynDB Manger 是一款基于 Web 的可视化工具，您可以在 EsgynDB 中使用备份和恢复功能。

修订历史

版本	日期
2.3.0	2018/01
2.2.0	2017/11
2.1.0	2016/11

1. 概览

为了确保在发生灾难性故障时能恢复数据库，EsgynDB 支持多种备份和恢复方式，例如：

- 全量数据库在线备份（EsgynDB 企业高级版支持）
- 在线备份和恢复 Schema 和表

从 2.4.0 开始，EsgynDB 支持增量备份数据库、指定 Schema 和表。

增量备份 (Incremental Backup)： 只备份增量变化。

全量备份 (Regular Backup)： 备份对象¹的所有数据，而不只是增量变化。



注意

全量备份既可以指**备份数据库中所有对象**，也可以指**备份数据库中某个特定对象**（例如，对一张表执行全量备份，即备份这张表；对一个 Schema 执行全量备份，即备份这个 Schema）。

EsgynDB 捕捉并保存增量备份中对象的变化。执行增量备份时，EsgynDB 为指定备份对象记录一个还原点，将前一次全量备份至该还原点之间的更改作为增量变化 (Incremental Delta)。将增量备份导出至远程文件夹时，EsgynDB 仅复制在前一次备份之后的数据更改 (增量变化)，因此能大幅减少传输的数据。

本指南提供备份/恢复操作的 SQL 命令，但您无需使用这些命令，只需使用 EsgynDB Manager 图形界面，便可简单快速地完成备份/恢复操作。更多信息，请参阅《EsgynDB Manager 用户指南》。

¹ 对象 (Object) 指的是 Schema 和表。

2. 备份与恢复

2.1 备份类型

您可以执行全量备份或增量备份。

如果未指定 Schema 或表，则 EsgynDB 将备份所有 Schema；如果数据库中所有 Schema 都包含在备份中，那么该备份为数据库的完整备份，即数据库的全量备份。您能从全量备份中恢复 Schema 或表的子集。

为了减少恢复次数和存储的增量数据，建议您定期执行全量备份，再多次执行增量备份。

备份类型	说明
全量备份 (Regular Backup)	<p>全量备份操作复制指定对象的所有数据。</p> <p>全量备份操作备份以下对象：</p> <ul style="list-style-type: none"> • 具有增量备份属性的对象。 • 不具有增量备份属性的对象。
增量备份 (Incremental Backup)	<p>增量备份操作仅复制在前一次备份之后的数据更改（增量变化）。</p> <p>增量备份操作备份以下对象：</p> <ul style="list-style-type: none"> • 具有增量备份属性的对象。 <p>仅复制对象的增量数据。增量备份操作为增量数据创建还原原点，不复制其它任何数据。</p> <ul style="list-style-type: none"> • 不具有增量备份属性的对象。 <p>复制对象的所有数据。</p>

2.1.1 增量备份

数据随时间的推移而增长，建议您在执行几次增量备份后执行全量备份。恢复增量备份时，EsgynDB 首先恢复全量备份，再恢复所有增量备份。因此，从增量备份中恢复对象所需时间与前一次全量备份后产生的数据增量成正比。

管理员需要考虑保留哪些备份，同时为了减少磁盘占用，也应考虑删除哪些备份。在删除旧备份前，建议您先导出备份，再删除本地备份。

例如，如果每天执行增量备份，且备份策略要求具备恢复一周内数据的能力，则您可以每周执行全量备份，每天执行增量备份。

	星期	备份操作	数据独立
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	↓

假设在第一周星期五需要恢复第一周星期三的数据，则恢复操作会先恢复第一周星期日的全量备份，再恢复第一周星期一至第一周星期三的增量备份。

假设在第二周星期三需要恢复第一周星期四的数据，则恢复操作会先恢复第一周星期日的全量备份，再恢复第一周星期一至第一周星期四的增量备份。

2.2 功能

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 EsgynDB 执行全量备份。 您可以执行全量、增量或系统备份 ² 。
恢复 (Restore)	从备份中恢复元数据/用户对象。 您可以从备份中恢复对象的子集。
查看备份 (Get Backup)	查看备份的信息。
删除备份 (Drop Backup)	删除备份。
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 EsgynDB。

² 更多关于系统备份的信息，请联系 Esgyn 技术支持。

2.3 访问数据库

事务 (Transaction) 可能涉及多个对象 (例如, Schema 和/或表), 为了保证事务一致性, 执行备份/恢复操作会在内部短暂地锁定某些被影响的元数据对象和用户表对象, 在执行备份/恢复操作期间:

- 如果查询访问的表/Schema 不受备份/恢复操作的影响, 则查询将正常执行。
- 如果正在执行的查询或新的查询访问的表/Schema 受备份/恢复操作的影响, 则系统将报错, 错误消息为备份/恢复操作正在进行。

2.4 注意事项

- 备份不支持非 HBase 或非事务性对象（例如，ORC 或 Hive 格式表）。
- 由于备份直接导入至 HBase，所以执行导入操作需要 Trafodion 用户具备 HBase sudo 权限。
- 仅支持 Red Hat 和 CentOS 操作系统。
- EsgynDB 2.4.0 无法兼容或访问在较低版本（EsgynDB 2.4.0 之前的版本）中执行的备份。如果您将数据库升级至 EsgynDB 2.4.0，建议您执行新的备份操作。

2.5 使用策略

策略	说明
全量备份 (Regular Backup)	<p>满足以下任一条件时，建议使用全量备份：</p> <ul style="list-style-type: none">• 将被备份的表较小• 备份频率较低（例如，每周）• 需要在较短时间内恢复备份 <p>针对相同对象而言，恢复全量备份比恢复增量备份的耗时更少，因为增量备份是先恢复全量备份、再恢复增量变化</p>
增量备份 (Incremental Backup)	<p>满足以下任一条件时，建议使用增量备份：</p> <ul style="list-style-type: none">• 将被备份的表较大• 多个备份之间差异较小• 备份频率较高（例如，每日）

2.6 支持

EsgynDB 企业高级版 2.4.0 和更高版本

- 支持全量备份/恢复 Schema/表
- 支持增量备份/恢复 Schema/表
- 支持导出和导入备份

3. 语法与语义

您可以在 EsgynDB 命令行工具 (Traf Command Interface, TrafCI³) 中执行备份/恢复操作, 输入以下命令。

3.1 创建备份/恢复

3.1.1 语法

以下命令备份/恢复 EsgynDB 对象。

```
BACKUP TRAFODION {,br-option1, [br-option2,]...[br-  
optionN]}  
  
RESTORE TRAFODION {,br-option1, [br-option2,]...  
[br-optionN]}  
  
br-option is:  
  {TAG 'tag-name'}  
  [SCHEMA (schema-name)]  
  [SCHEMAS (schema-name1,...schma-nameN)]  
  [TABLE (table-name)]  
  [TABLES (table-name1,...table-nameN)]  
  [INCREMENTAL]  
  [SHOW OBJECTS]  
  [OVERRIDE]
```

³ 更多信息, 请参阅《EsgynDB 命令行工具指南》。

3.1.2 语义

语法项	语义/说明
<i>br-option</i>	您可以任意指定 <i>br-option</i> 中语法项的顺序，用逗号隔开多个语法项。
TAG ' <i>tag-name</i> '	指定识别备份的 TAG， <i>tag-name</i> 最多 32 个字符。
SCHEMA (<i>schema-name</i>)	备份/恢复 Schema <i>schema-name</i> 的所有对象。
SCHEMAS (<i>schema-name1, ...schma-nameN</i>)	备份/恢复多个 Schema 中的所有对象。
TABLE (<i>table-name</i>)	备份/恢复表 <i>table-name</i> 。
TABLES (<i>table-name1, ...table-nameN</i>)	备份/恢复多个表 (<i>table-name1, ...table-nameN</i>)。
INCREMENTAL	指定备份为增量备份。
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。
OVERWRITE	<p>使用现有 TAG 创建备份。</p> <hr/> <p> 注意</p> <ul style="list-style-type: none"> 该选项先在内部删除旧 TAG 备份，再创建同名的新 TAG 备份。所以，如需在执行增量备份时使用该选项，建议您先执行全量备份，再在之后的增量备份中使用该选项。如果没有执行全量备份，则在执行增量备份使用该选项时，EsgynDB 将不会捕捉增量数据，而是复制所有数据。 EsgynDB 为备份的 TAG 名称追加唯一时间戳标识，这有助于调度任务每次使用相同的 TAG 名称执行备份操作命令。但如果指定了 OVERWRITE 选项，则 TAG 名称不会包含唯一时间戳标识。 该语法项不适用于恢复操作。



注意

- 只有 TAG '*tag-name*' 是**必选项**，其它语法项是可选项。
- 您能在一条命令中指定多个 Schema 和表，EsgynDB 将备份/恢复该 Schema 和表中的所有对象。
- 如果备份时未指定 Schema 或表，则 EsgynDB 将执行全量备份，即备份所有 Schema 和表。
- 如果恢复指定 TAG 备份时未指定 Schema 或表，则 EsgynDB 将恢复该备份中的所有对象。

3.2 创建 Schema/表，指定增量备份属性

EsgynDB 对具备增量备份属性的对象执行增量备份，您能为每个 Schema 或表单独指定增量备份属性。



注意

- 如果 Schema 具备增量属性，则 Schema 中的表继承增量属性。
- EsgynDB 对不具备增量备份属性的对象执行全量备份。

3.2.1 语法

以下命令创建 Schema，并为 Schema 指定增量备份属性。

```
CREATE SCHEMA schema-clause4 INCREMENTAL BACKUP;  
  
CREATE [schema-class] SCHEMA [IF NOT EXISTS] schema-clause INCREMENTAL BACKUP;  
  
schema-class is:  
    [ PRIVATE | SHARED ]  
  
schema-clause is:  
    { schema-name [AUTHORIZATION authid] | AUTHORIZATION authid }
```

⁴ 更多关于 CREATE SCHEMA Statement 语法的信息，请参阅《EsgynDB SQL Reference Manual》。

以下命令创建表，并为表指定增量备份属性。

```
CREATE5 TABLE [IF NOT EXISTS] table-name
table-spec6 ATTRIBUTE INCREMENTAL BACKUP;

table-spec is:
    (table-element [, table-element]...)

table-element is:
    column-definition
    | [CONSTRAINT constraint-name] table-constraint

column-definition is:
    column data-type
    [DEFAULT default | NO DEFAULT]
    [[constraint constraint-name] column-constraint]...
```

⁵ 无法为 volatile 表指定增量属性。

⁶ 更多关于 CREATE TABLE Statement 语法的信息，请参阅《EsgynDB SQL Reference Manual》。

3.3 删除备份

DROP 命令删除本地备份，该命令中的 TAG 名称必须包含时间戳标识符。如需查看 TAG 名称，使用 GET 命令。更多关于 GET 命令的信息，请参阅 [3.4 查看备份](#)。



注意

- 如果在全量备份的基础上执行了增量备份，此时如需删除全量备份，则应先删除增量备份，再删除全量备份，否则系统将报错。
- 如果在执行增量备份前未执行过全量备份，则首次增量备份实际上等同于全量备份。

3.3.1 语法

以下命令删除匹配 `%match-string%` 条件的备份。

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

以下命令删除 TAG '`tag-name`' 备份。

```
DROP BACKUP TAG, TAG 'tag-name'
```

3.3.2 语义

语法项	语义
MATCH '%match string%'	指定匹配 ' <code>%match-string%</code> ' 条件的备份。
TAG ' <code>tag-name</code> '	备份的 TAG 名称（唯一标识符）。 如需查看 TAG 名称，使用 GET 命令。 更多关于 GET 命令的信息，请参阅 3.4 查看备份 。

3.4 查看备份

GET 命令显示 EsgynDB 所有备份的 TAG 名称和信息。

3.4.1 语法

以下命令显示所有备份的 TAG 名称和信息。

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

3.4.2 语义

语法项	语义
SHOW DETAILS	返回备份的所有信息，例如： <ul style="list-style-type: none"> • 时间戳 • 状态 <ul style="list-style-type: none"> ◦ 有效 (Valid) ◦ 无效 (Invalid) • 备份类型 <ul style="list-style-type: none"> ◦ 全量备份 ◦ 增量备份
MATCH '%match string%'	指定匹配 '%match string%' 条件的备份。

3.5 导出/导入备份

导出备份：将 EsgynD 的备份复制至 HDFS 备份文件夹。

导入备份：将 HDFS 备份文件夹的备份复制至 EsgynDB。

3.5.1 语法

以下命令将 EsgynDB 的 TAG '*tag-name*' 备份导出至指定 HDFS 备份文件夹。

```
EXPORT BACKUP TO LOCATION 'backup-root-location', TAG
'tag-name'
```

以下命令将 HDFS 备份文件夹的 TAG '*tag-name*' 备份导入至 EsgynDB。

```
IMPORT BACKUP FROM LOCATION 'backup-root-location', TAG
'tag-name'
```

3.5.2 语义

语法项	语义
<i>backup-root-location</i>	<p>HDFS 备份文件夹的绝对路径。</p> <p>该路径必须包含完全限定的主机名称，主机名称可以是本地或远程集群。</p> <p>HDFS 路径必须是一个完全限定的备份文件夹，该备份文件夹应是全量备份与增量备份共同使用的导出目录。如果更改了备份文件夹，则在导出增量备份时，EsgynDB 将重新导出所有数据（即全量备份）和所有增量备份。</p>

TAG ' <i>tag-name</i> '	备份的 TAG 名称 (唯一标识符)。 如需查看 TAG 名称, 使用 GET 命令。 更多关于 GET 命令的信息, 请参阅 3.4 查看备份 。
-------------------------	--

4. 示例

以下示例创建增量备份和全量备份、恢复增量备份和全量备份、导出备份和导入备份。

- 1、 创建 Schema *schbkp*，未指定备份属性，默认备份属性为全量备份。

```
>>CREATE SCHEMA schbkp;
--- SQL operation complete.
```

- 2、 在 Schema *schbkp* 中创建表 *schbkp.t1*，指定增量备份。

```
>>CREATE TABLE schbkp.t1 (a int, b int) ATTRIBUTE
INCREMENTAL BACKUP;
--- SQL operation complete.
```

- 3、 在表 *schbkp.t1* 上创建索引 *ti*。

```
>>CREATE INDEX ti ON schbkp.t1(a);
--- SQL operation complete.
```

- 4、 查看表 *schbkp.t1* 的信息。

```
>>SHOWDDL schbkp.t1;

CREATE TABLE TRAFODION.SCHBKP.T1
(
  A      INT DEFAULT NULL NOT SERIALIZED
, B      INT DEFAULT NULL NOT SERIALIZED
)
ATTRIBUTES ALIGNED FORMAT NAMESPACE 'TRAF_1500000'
INCREMENTAL BACKUP
;

CREATE INDEX TI ON TRAFODION.SCHBKP.T1
(
```


4. 示例

```
A ASC
)
ATTRIBUTES ALIGNED FORMAT
;

-- GRANT SELECT, INSERT, DELETE, UPDATE,
REFERENCES ON TRAFODION.SCHBKP.T1 TO DB__ROOT WITH
GRANT OPTION;

--- SQL operation complete.
```

5、 在 Schema *schbkp* 中创建表 *schbkp.t2*。

```
>>CREATE TABLE schbkp.t2 (a int, b int);
--- SQL operation complete.
```

6、 查看表 *schbkp.t2* 的信息。

```
>>SHOWDDL schbkp.t2;

CREATE TABLE TRAFODION.SCHBKP.T2
(
  A INT DEFAULT NULL NOT SERIALIZED
  , B INT DEFAULT NULL NOT SERIALIZED
)
ATTRIBUTES ALIGNED FORMAT NAMESPACE 'TRAF_1500000'
;

-- GRANT SELECT, INSERT, DELETE, UPDATE,
REFERENCES ON TRAFODION.SCHBKP.T2 TO DB__ROOT WITH
GRANT OPTION;

--- SQL operation complete.
```

- 7、 创建 Schema *schinc*，指定增量备份。

```
>>CREATE SCHEMA schinc INCREMENTAL BACKUP;
--- SQL operation complete.
```

- 8、 在 Schema *schinc* 中创建表 *schinc.t3*，未指定备份属性。

```
>>CREATE TABLE schinc.t3 (a int, b int);
--- SQL operation complete.
```

- 9、 查看表 *schinc.t3* 的信息。虽然未为表 *schinc.t3* 指定备份属性，但表 *schinc.t3* 继承 Schema *schinc* 的备份属性，即增量备份。

```
>>SHOWDDL schinc.t3;

CREATE TABLE TRAFODION.SCHINC.T3
(
  A INT DEFAULT NULL NOT SERIALIZED
, B INT DEFAULT NULL NOT SERIALIZED
)
ATTRIBUTES ALIGNED FORMAT NAMESPACE 'TRAF_1500000'
INCREMENTAL BACKUP
;

-- GRANT SELECT, INSERT, DELETE, UPDATE,
REFERENCES ON TRAFODION.SCHINC.T3 TO DB__ROOT WITH
GRANT OPTION;

--- SQL operation complete.
```

- 10、 在 Schema *schinc* 中创建表 *schinc.t4*，未指定备份属性。

```
>>CREATE TABLE schinc.t4 (a int, b int);
--- SQL operation complete.
```

- 11、查看表 *schinc.t4* 的信息。虽然未为表 *schinc.t4* 指定备份属性，但表 *schinc.t4* 继承 Schema *schinc* 的备份属性，即增量备份。

```
>>SHOWDDL schinc.t4;
```

```
CREATE TABLE TRAFODION.SCHINC.T4
(
  A INT DEFAULT NULL NOT SERIALIZED
, B INT DEFAULT NULL NOT SERIALIZED
)
ATTRIBUTES ALIGNED FORMAT NAMESPACE 'TRAF_1500000'
INCREMENTAL BACKUP
;

-- GRANT SELECT, INSERT, DELETE, UPDATE,
REFERENCES ON TRAFODION.SCHINC.T4 TO DB__ROOT WITH
GRANT OPTION;

--- SQL operation complete.
```

- 12、查看 Schema *schbkp* 的所有对象。

```
>>GET OBJECTS IN SCHEMA schbkp;
```

```
Tables in Schema TRAFODION.SCHBKP
```

```
=====
```

```
SB_HISTOGRAMS
```

```
SB_HISTOGRAM_INTERVALS
```

```
SB_PERSISTENT_SAMPLES
```

```
T1
```

```
T2
```

```
Indexes in Schema TRAFODION.SCHBKP
```

```
=====
```

```
TI
```

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

13、查看 Schema *schinc* 的所有对象。

```
>>GET OBJECTS IN SCHEMA schinc;
```

```
Tables in Schema TRAFODION.SCHINC
```

```
=====
```

```
SB_HISTOGRAMS
```

```
SB_HISTOGRAM_INTERVALS
```

```
SB_PERSISTENT_SAMPLES
```

```
T3
```

```
T4
```

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

14、向表 *schbkp.t1*、表 *schbkp.t2*、表 *schinc.t3* 和表 *schinc.t4* 插入数据。

```
>>INSERT INTO schbkp.t1 VALUES(10,10);
```

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

```
>>INSERT INTO schbkp.t2 VALUES(20,20);
```

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

```
>>INSERT INTO schinc.t3 VALUES(30,30);
```

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

```
>>INSERT INTO schinc.t4 VALUES(40,40);
```

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

15、创建全量备份 TAG 'fulldb', 查看该备份的所有对象。

```
>>BACKUP TRAFODION, TAG 'fulldb', SHOW OBJECTS;
```

```
MetaData objects to be backed up
```

```
=====
```

```
TRAFODION._BACKUP_fulldb_.COLUMNS
```

```
TRAFODION._BACKUP_fulldb_.INDEXES
```

```
TRAFODION._BACKUP_fulldb_.KEYS
```

```
TRAFODION._BACKUP_fulldb_.OBJECTS
```

```
TRAFODION._BACKUP_fulldb_.REF_CONSTRAINTS
```

```
TRAFODION._BACKUP_fulldb_.SEQ_GEN
```

```
TRAFODION._BACKUP_fulldb_.TABLES
```

```
TRAFODION._BACKUP_fulldb_.TABLE_CONSTRAINTS
```

```
TRAFODION._BACKUP_fulldb_.TEXT
```

```
TRAFODION._BACKUP_fulldb_.UNIQUE_REF_CONSTR_USAGE
```

```
TRAFODION._BACKUP_fulldb_.VIEWS
```

```
TRAFODION._BACKUP_fulldb_.VIEWS_USAGE
```

```
TRAFODION._BACKUP_fulldb_.COLUMN_PRIVILEGES
```

```
TRAFODION._BACKUP_fulldb_.OBJECT_PRIVILEGES
```

```
TRAFODION._BACKUP_fulldb_.SCHEMA_PRIVILEGES
```

```
TRAFODION._BACKUP_fulldb_.SB_HISTOGRAMS
```

```
TRAFODION._BACKUP_fulldb_.SB_HISTOGRAM_INTERVALS
```

```
TRAFODION._BACKUP_fulldb_.SB_PERSISTENT_SAMPLES
```

```
User objects to be backed up
```

```
=====
```

```
TRAFODION.SCH.SB_HISTOGRAMS
```

```
TRAFODION.SCH.SB_HISTOGRAM_INTERVALS
```

```
TRAFODION.SCH.SB_PERSISTENT_SAMPLES
```

```
TRAFODION.SCHBKP.SB_HISTOGRAMS
```

```
TRAFODION.SCHBKP.SB_HISTOGRAM_INTERVALS
```

```

TRAFODION.SCHBKP.SB_PERSISTENT_SAMPLES
TRAFODION.SCHBKP.T1
TRAFODION.SCHBKP.T2
TRAFODION.SCHBKP.TI
TRAFODION.SCHINC.SB_HISTOGRAMS
TRAFODION.SCHINC.SB_HISTOGRAM_INTERVALS
TRAFODION.SCHINC.SB_PERSISTENT_SAMPLES
TRAFODION.SCHINC.T3
TRAFODION.SCHINC.T4
TRAFODION.SEABASE.SB_HISTOGRAMS
TRAFODION.SEABASE.SB_HISTOGRAM_INTERVALS
TRAFODION.SEABASE.SB_PERSISTENT_SAMPLES

--- SQL operation complete.

```

16、执行全量备份 TAG 'fulldb'。

```

>>BACKUP TRAFODION, TAG 'fulldb';
--- SQL operation complete.

```

17、查看所有备份的信息。

```

>>GET ALL BACKUP TAGS, SHOW DETAILS;

```

```

BackupTag          BackupTime
BackupStatus      BackupOperation
=====
=====
fulldb_00212393035662759229  2018-05-14:05:27:42
VALID              REGULAR

```

```

--- SQL operation complete.

```

18、删除表 *schbkp.t1*。

```
>>DROP TABLE schbkp.t1 CASCADE;  
--- SQL operation complete.
```

19、查看从全量备份 TAG '*fulldb_00212393035662759229*' 中恢复表 *schbkp.t1* 时，
将被恢复的所有对象。

```
>>RESTORE TRAFODION, TAG  
'fulldb_00212393035662759229', TABLES(schbkp.t1),  
SHOW OBJECTS;
```

```
MetaData objects to be restored  
=====
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_fulldb_0021239303566275  
9229_.COLUMNS
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_fulldb_0021239303566275  
9229_.COLUMN_PRIVILEGES
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_fulldb_0021239303566275  
9229_.INDEXES
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_fulldb_0021239303566275  
9229_.KEYS
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_fulldb_0021239303566275  
9229_.OBJECTS
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_fulldb_0021239303566275  
9229_.OBJECT_PRIVILEGES
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_fulldb_0021239303566275  
9229_.REF_CONSTRAINTS
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_fulldb_0021239303566275  
9229_.SB_HISTOGRAMS
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_fulldb_0021239303566275  
9229_.SB_HISTOGRAM_INTERVALS
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_fulldb_0021239303566275  
9229_.SB_PERSISTENT_SAMPLES
```

```

TRAF_RSRVD_3:TRAFODION._BACKUP_fullldb_0021239303566275
9229_.SCHEMA_PRIVILEGES
TRAF_RSRVD_3:TRAFODION._BACKUP_fullldb_0021239303566275
9229_.SEQ_GEN
TRAF_RSRVD_3:TRAFODION._BACKUP_fullldb_0021239303566275
9229_.TABLES
TRAF_RSRVD_3:TRAFODION._BACKUP_fullldb_0021239303566275
9229_.TABLE_CONSTRAINTS
TRAF_RSRVD_3:TRAFODION._BACKUP_fullldb_0021239303566275
9229_.TEXT
TRAF_RSRVD_3:TRAFODION._BACKUP_fullldb_0021239303566275
9229_.UNIQUE_REF_CONSTR_USAGE
TRAF_RSRVD_3:TRAFODION._BACKUP_fullldb_0021239303566275
9229_.VIEWS
TRAF_RSRVD_3:TRAFODION._BACKUP_fullldb_0021239303566275
9229_.VIEWS_USAGE

```

```
User objects to be restored
```

```
=====
```

```

TRAF_1500000:TRAFODION.SCHBKP.T1
TRAF_1500000:TRAFODION.SCHBKP.TI

```

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

20、从全量备份 TAG 'fullldb_00212393035662759229' 中恢复表 *schbkp.t1*。

```

>>RESTORE TRAFODION, TAG
'fullldb_00212393035662759229', TABLES(schbkp.t1);

```

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


21、查看恢复结果。

(1) 查看表 *schbkp.t1* 的信息。

```
>>SHOWDDL schbkp.t1;
```

```
CREATE TABLE TRAFODION.SCHBKP.T1
(
  A INT DEFAULT NULL NOT SERIALIZED
, B INT DEFAULT NULL NOT SERIALIZED
)
ATTRIBUTES ALIGNED FORMAT NAMESPACE 'TRAF_1500000'
INCREMENTAL BACKUP
;
```

```
CREATE INDEX TI ON TRAFODION.SCHBKP.T1
(
  A ASC
)
ATTRIBUTES ALIGNED FORMAT
;
```

```
-- GRANT SELECT, INSERT, DELETE, UPDATE,
REFERENCES ON TRAFODION.SCHBKP.T1 TO DB__ROOT WITH
GRANT OPTION;
--- SQL operation complete.
```

(2) 查看表 *schbkp.t1* 的数据。

```
>>SELECT * FROM schbkp.t1;
```

```
A          B
-----
          10          10
```

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

22、向表 *schbkp.t1*、表 *schbkp.t2*、表 *schinc.t3* 和表 *schinc.t4* 插入数据（增量变化）。

```
>>INSERT INTO schbkp.t1 VALUES (11,11);  
--- 1 row(s) inserted.
```

```
>>INSERT INTO schbkp.t2 VALUES (21,21);  
--- 1 row(s) inserted.
```

```
>>INSERT INTO schinc.t3 VALUES (31,31);  
--- 1 row(s) inserted.
```

```
>>INSERT INTO schinc.t4 VALUES (41,41);  
--- 1 row(s) inserted.
```

23、创建增量备份 TAG '*sunday*'（备份 Schema *schbkp* 和表 *schinc.t3*），查看该备份的所有对象。

```
>>BACKUP TRAFODION, TAG 'sunday', SCHEMAS(schbkp),  
TABLES(schinc.t3), INCREMENTAL, SHOW OBJECTS;
```

```
MetaData objects to be backed up
```

```
=====
```

```
TRAFODION._BACKUP_sunday_.COLUMNS
```

```
TRAFODION._BACKUP_sunday_.INDEXES
```

```
TRAFODION._BACKUP_sunday_.KEYS
```

```
TRAFODION._BACKUP_sunday_.OBJECTS
```

```
TRAFODION._BACKUP_sunday_.REF_CONSTRAINTS
```

```
TRAFODION._BACKUP_sunday_.SEQ_GEN
```

```
TRAFODION._BACKUP_sunday_.TABLES
```

```
TRAFODION._BACKUP_sunday_.TABLE_CONSTRAINTS
```

```
TRAFODION._BACKUP_sunday_.TEXT
```

```
TRAFODION._BACKUP_sunday_.UNIQUE_REF_CONSTR_USAGE
```

```
TRAFODION._BACKUP_sunday_.VIEWS
```

4. 示例

```
TRAFODION._BACKUP_sunday_.VIEWS_USAGE
TRAFODION._BACKUP_sunday_.COLUMN_PRIVILEGES
TRAFODION._BACKUP_sunday_.OBJECT_PRIVILEGES
TRAFODION._BACKUP_sunday_.SCHEMA_PRIVILEGES
TRAFODION._BACKUP_sunday_.SB_HISTOGRAMS
TRAFODION._BACKUP_sunday_.SB_HISTOGRAM_INTERVALS
TRAFODION._BACKUP_sunday_.SB_PERSISTENT_SAMPLES
```

```
User objects to be backed up
=====
```

```
TRAFODION.SCHBKP.SB_HISTOGRAMS
TRAFODION.SCHBKP.SB_HISTOGRAM_INTERVALS
TRAFODION.SCHBKP.SB_PERSISTENT_SAMPLES
TRAFODION.SCHBKP.T1
TRAFODION.SCHBKP.T2
TRAFODION.SCHBKP.TI
TRAFODION.SCHINC.T3
```

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

24、执行增量备份 TAG 'sunday'。

```
>>backup trafodion, tag 'sunday', schemas(schbkp),
tables(schinc.t3), incremental;
```

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

25、查看所有备份的信息。

```
>>GET ALL BACKUP TAGS, SHOW DETAILS;
```

```
BackupTag          BackupTime
BackupStatus  BackupOperation
```

```
=====
```

```

=====

fulldb_00212393035662759229 2018-05-14:05:27:42
VALID          REGULAR
sunday_00212393036364859822 2018-05-14:05:39:24
VALID          INCREMENTAL

--- SQL operation complete.

```

- 26、将增量备份 TAG '*sunday_00212393036364859822*' 导出至远程 HDFS 文件夹。

```

>>EXPORT BACKUP TO LOCATION
'hdfs://localhost:9000/user/trafodion/backups',
tag 'sunday_00212393036364859822';

--- SQL operation complete.

```

- 27、向表 *schinc.t3* 插入数据 (增量变化)。

```

>>INSERT INTO schinc.t3 VALUES(32,32);
--- 1 row(s) inserted.

```

- 28、创建全量备份 TAG '*monday*' (备份表 *schinc.t3*)，查看该备份的所有对象。

```

>>BACKUP TRAFODION, TAG 'monday', TABLE(schinc.t3),
SHOW OBJECTS;

MetaData objects to be backed up
=====

TRAFODION._BACKUP_monday_.COLUMNS
TRAFODION._BACKUP_monday_.INDEXES
TRAFODION._BACKUP_monday_.KEYS
TRAFODION._BACKUP_monday_.OBJECTS
TRAFODION._BACKUP_monday_.REF_CONSTRAINTS

```

```

TRAFODION._BACKUP_monday_.SEQ_GEN
TRAFODION._BACKUP_monday_.TABLES
TRAFODION._BACKUP_monday_.TABLE_CONSTRAINTS
TRAFODION._BACKUP_monday_.TEXT
TRAFODION._BACKUP_monday_.UNIQUE_REF_CONSTR_USAGE
TRAFODION._BACKUP_monday_.VIEWS
TRAFODION._BACKUP_monday_.VIEWS_USAGE
TRAFODION._BACKUP_monday_.COLUMN_PRIVILEGES
TRAFODION._BACKUP_monday_.OBJECT_PRIVILEGES
TRAFODION._BACKUP_monday_.SCHEMA_PRIVILEGES
TRAFODION._BACKUP_monday_.SB_HISTOGRAMS
TRAFODION._BACKUP_monday_.SB_HISTOGRAM_INTERVALS
TRAFODION._BACKUP_monday_.SB_PERSISTENT_SAMPLES

```

```
User objects to be backed up
```

```
=====
```

```
TRAFODION.SCHINC.T3
```

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

29、执行全量备份 TAG 'monday'。

```
>>backup trafodion, tag 'monday', tables(schinc.t3);
```

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

30、查看所有备份的信息。

```
>>GET ALL BACKUP TAGS, SHOW DETAILS;
```

```
BackupTag          BackupTime
```

```
BackupStatus  BackupOperation
```

```
=====
```

```
=====
```

4. 示例

```
fulldb_00212393035662759229 2018-05-14:05:27:42
VALID          REGULAR
monday_00212393038085889128 2018-05-14:06:08:05
VALID          REGULAR
sunday_00212393036364859822 2018-05-14:05:39:24
VALID          INCREMENTAL

--- SQL operation complete.
```

31、删除表 *schinc.t3*。

```
>>DROP TABLE schinc.t3;

--- SQL operation complete.
```

32、删除增量备份 TAG '*sunday_00212393036364859822*'。

```
>>DROP BACKUP TAG, TAG 'sunday_00212393036364859822';

--- SQL operation complete.
```

33、查看所有备份的信息。

```
>>GET ALL BACKUP TAGS, SHOW DETAILS;

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

fulldb_00212393035662759229 2018-05-14:05:27:42
VALID          REGULAR
monday_00212393038085889128 2018-05-14:06:08:05
VALID          REGULAR

--- SQL operation complete.
```

34、导入增量备份 TAG '*sunday_00212393036364859822*'。

```
>>IMPORT BACKUP FROM LOCATION
'hdfs://localhost:9000/user/trafodion/backups',
tag 'sunday_00212393036364859822';

--- SQL operation complete.
```

35、查看所有备份的信息。

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

fulldb_00212393035662759229  2018-05-14:05:27:42
VALID                REGULAR
monday_00212393038085889128  2018-05-14:06:08:05
VALID                REGULAR
sunday_00212393036364859822  2018-05-14:05:39:24
NO_METADATA        INCREMENTAL (IMPORTED)

--- SQL operation complete.
```

36、查看增量备份 TAG '*sunday_00212393036364859822*'能恢复的对象。

```
>>RESTORE TRAFODION, TAG
'sunday_00212393036364859822', SHOW OBJECTS;

MetaData objects to be restored
=====

TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.COLUMNS
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.COLUMN_PRIVILEGES
```

4. 示例

```
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.INDEXES
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.KEYS
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.OBJECTS
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.OBJECT_PRIVILEGES
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.REF_CONSTRAINTS
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.SB_HISTOGRAMS
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.SB_HISTOGRAM_INTERVALS
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.SB_PERSISTENT_SAMPLES
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.SCHEMA_PRIVILEGES
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.SEQ_GEN
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.TABLES
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.TABLE_CONSTRAINTS
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.TEXT
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.UNIQUE_REF_CONSTR_USAGE
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.VIEWS
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.VIEWS_USAGE
```

```
User objects to be restored
```



```

=====

TRAF_1500000:TRAFODION.SCHBKP.SB_HISTOGRAMS
TRAF_1500000:TRAFODION.SCHBKP.SB_HISTOGRAM_INTERVALS
TRAF_1500000:TRAFODION.SCHBKP.SB_PERSISTENT_SAMPLES
TRAF_1500000:TRAFODION.SCHBKP.T1
TRAF_1500000:TRAFODION.SCHBKP.T2
TRAF_1500000:TRAFODION.SCHBKP.TI
TRAF_1500000:TRAFODION.SCHINC.T3

--- SQL operation complete.

```

- 37、查看从增量备份 TAG '*sunday_0021239303636485982*' 中恢复表 *schinc.t3* 时，
将被恢复的所有对象。

```

>>RESTORE TRAFODION, TAG
'sunday_00212393036364859822', TABLES(schinc.t3), SHOW
OBJECTS;

```

```

MetaData objects to be restored

```

```

=====

TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.COLUMNS
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.COLUMN_PRIVILEGES
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.INDEXES
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.KEYS
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.OBJECTS
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.OBJECT_PRIVILEGES

```

4. 示例

```
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.REF_CONSTRAINTS
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.SB_HISTOGRAMS
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.SB_HISTOGRAM_INTERVALS
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.SB_PERSISTENT_SAMPLES
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.SCHEMA_PRIVILEGES
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.SEQ_GEN
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.TABLES
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.TABLE_CONSTRAINTS
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.TEXT
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.UNIQUE_REF_CONSTR_USAGE
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.VIEWS
TRAF_RSRVD_3:TRAFODION._BACKUP_sunday_0021239303636485
9822_.VIEWS_USAGE
```

```
User objects to be restored
```

```
=====
```

```
TRAF_1500000:TRAFODION.SCHINC.T3
```

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

38、从增量备份 TAG 'sunday_00212393036364859822' 中恢复表 *schinc.t3*。

```
>>RESTORE TRAFODION, TAG 'sunday_00212393036364859822',  
TABLES (schinc.t3);
```

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

39、查看表 *schinc.t3* 的数据。

```
>>SELECT * FROM schinc.t3;
```

```
A          B  
-----  
          30          30  
          31          31
```

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

40、查看所有备份的信息。

```
>>GET ALL BACKUP TAGS, SHOW DETAILS;
```

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

```
fulldb_00212393035662759229  2018-05-14:05:27:42  
VALID          REGULAR  
monday_00212393038085889128  2018-05-14:06:08:05  
VALID          REGULAR  
sunday_00212393036364859822  2018-05-14:05:39:24  
VALID INCREMENTAL (IMPORTED)
```

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

- 41、 查看恢复全量备份 TAG '*monday_00212393038085889128*' 时, 将被恢复的所有对象。

```
>>RESTORE TRAFODION, TAG
'monday_00212393038085889128', SHOW OBJECTS;

MetaData objects to be restored
=====

TRAF_RSRVD_3:TRAFODION._BACKUP_monday_0021239303808588
9128_.COLUMNS
TRAF_RSRVD_3:TRAFODION._BACKUP_monday_0021239303808588
9128_.COLUMN_PRIVILEGES
TRAF_RSRVD_3:TRAFODION._BACKUP_monday_0021239303808588
9128_.INDEXES
TRAF_RSRVD_3:TRAFODION._BACKUP_monday_0021239303808588
9128_.KEYS
TRAF_RSRVD_3:TRAFODION._BACKUP_monday_0021239303808588
9128_.OBJECTS
TRAF_RSRVD_3:TRAFODION._BACKUP_monday_0021239303808588
9128_.OBJECT_PRIVILEGES
TRAF_RSRVD_3:TRAFODION._BACKUP_monday_0021239303808588
9128_.REF_CONSTRAINTS
TRAF_RSRVD_3:TRAFODION._BACKUP_monday_0021239303808588
9128_.SB_HISTOGRAMS
TRAF_RSRVD_3:TRAFODION._BACKUP_monday_0021239303808588
9128_.SB_HISTOGRAM_INTERVALS
TRAF_RSRVD_3:TRAFODION._BACKUP_monday_0021239303808588
9128_.SB_PERSISTENT_SAMPLES
TRAF_RSRVD_3:TRAFODION._BACKUP_monday_0021239303808588
9128_.SCHEMA_PRIVILEGES
TRAF_RSRVD_3:TRAFODION._BACKUP_monday_0021239303808588
9128_.SEQ_GEN
TRAF_RSRVD_3:TRAFODION._BACKUP_monday_0021239303808588
```

```

9128_.TABLES
TRAF_RSRVD_3:TRAFODION._BACKUP_monday_0021239303808588
9128_.TABLE_CONSTRAINTS
TRAF_RSRVD_3:TRAFODION._BACKUP_monday_0021239303808588
9128_.TEXT
TRAF_RSRVD_3:TRAFODION._BACKUP_monday_0021239303808588
9128_.UNIQUE_REF_CONSTR_USAGE
TRAF_RSRVD_3:TRAFODION._BACKUP_monday_0021239303808588
9128_.VIEWS
TRAF_RSRVD_3:TRAFODION._BACKUP_monday_0021239303808588
9128_.VIEWS_USAGE

```

```
User objects to be restored
```

```
=====
```

```
TRAF_1500000:TRAFODION.SCHINC.T3
```

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

42、恢复全量备份 TAG 'monday_00212393038085889128'。

```
>>RESTORE TRAFODION, TAG
'monday_00212393038085889128';
```

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

43、查看表 *schinc.t3* 的数据。

```
>>select * from schinc.t3;
```

```

A          B
-----
          32          32
          30          30
          31          31
--- 3 row(s) selected.

```

44、向表 *schbkp.t1*、表 *schbkp.t2*、表 *schinc.t3* 和表 *schinc.t4* 插入数据 (增量变化)。

```
>>INSERT INTO schbkp.t1 VALUES (13,13);  
--- 1 row(s) inserted.
```

```
>>INSERT INTO schbkp.t2 VALUES (23,23);  
--- 1 row(s) inserted.
```

```
>>INSERT INTO schinc.t3 VALUES (33,33);  
--- 1 row(s) inserted.
```

```
>>INSERT INTO schinc.t4 VALUES (43,43);  
--- 1 row(s) inserted.
```

45、创建增量备份 TAG '*tuesday*' (备份 Schema *schbkp*, 指定 *override*), 查看该备份的所有对象。

```
>>BACKUP TRAFODION, TAG 'tuesday', SCHEMA(schbkp),  
INCREMENTAL, OVERRIDE, SHOW OBJECTS;
```

```
MetaData objects to be backed up
```

```
=====
```

```
TRAFODION._BACKUP_tuesday_.COLUMNS  
TRAFODION._BACKUP_tuesday_.INDEXES  
TRAFODION._BACKUP_tuesday_.KEYS  
TRAFODION._BACKUP_tuesday_.OBJECTS  
TRAFODION._BACKUP_tuesday_.REF_CONSTRAINTS  
TRAFODION._BACKUP_tuesday_.SEQ_GEN  
TRAFODION._BACKUP_tuesday_.TABLES  
TRAFODION._BACKUP_tuesday_.TABLE_CONSTRAINTS  
TRAFODION._BACKUP_tuesday_.TEXT  
TRAFODION._BACKUP_tuesday_.UNIQUE_REF_CONSTR_USAGE  
TRAFODION._BACKUP_tuesday_.VIEWS
```

```

TRAFODION._BACKUP_tuesday_.VIEWS_USAGE
TRAFODION._BACKUP_tuesday_.COLUMN_PRIVILEGES
TRAFODION._BACKUP_tuesday_.OBJECT_PRIVILEGES
TRAFODION._BACKUP_tuesday_.SCHEMA_PRIVILEGES
TRAFODION._BACKUP_tuesday_.SB_HISTOGRAMS
TRAFODION._BACKUP_tuesday_.SB_HISTOGRAM_INTERVALS
TRAFODION._BACKUP_tuesday_.SB_PERSISTENT_SAMPLES

```

```
User objects to be backed up
```

```
=====
```

```

TRAFODION.SCHBKP.SB_HISTOGRAMS
TRAFODION.SCHBKP.SB_HISTOGRAM_INTERVALS
TRAFODION.SCHBKP.SB_PERSISTENT_SAMPLES
TRAFODION.SCHBKP.T1
TRAFODION.SCHBKP.T2
TRAFODION.SCHBKP.TI

```

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

46、执行增量备份 TAG 'tuesday'。

```
>>BACKUP TRAFODION, TAG 'tuesday', SCHEMA(schbkp),
INCREMENTAL, OVERRIDE;
```

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

47、查看所有备份的信息。



注意

由于指定了 `override`，所以 TAG 'tuesday' 不包括时间戳。

4. 示例

```
>>GET ALL BACKUP TAGS, SHOW DETAILS;
```

```
BackupTag          BackupTime
BackupStatus      BackupOperation
=====
=====
tuesday           2018-05-14:06:27:03
VALID             INCREMENTAL
fulldb_00212393035662759229  2018-05-14:05:27:42
VALID             REGULAR
monday_00212393038085889128  2018-05-14:06:08:05
VALID             REGULAR
sunday_00212393036364859822  2018-05-14:05:39:24
VALID INCREMENTAL (IMPORTED)

--- SQL operation complete.
```

48、查看恢复增量备份 TAG 'tuesday' 时，将被恢复的所有对象。

```
>>RESTORE TRAFODION, TAG 'tuesday', SHOW OBJECTS;
```

```
MetaData objects to be restored
=====
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.COLUMNS
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.COLUMN_PRIVILEGES
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.INDEXES
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.KEYS
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.OBJECTS
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.OBJECT_PRIVILEGES
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.REF_CONSTRAINTS
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.SB_HISTOGRAMS
```


4. 示例

```
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.SB_HISTOGRAM_I
INTERVALS
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.SB_PERSISTENT
SAMPLES
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.SCHEMA_PRIVILE
GES
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.SEQ_GEN
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.TABLES
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.TABLE_CONSTRAI
NTS
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.TEXT
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.UNIQUE_REF_CON
STR_USAGE
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.VIEWS
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.VIEWS_USAGE
```

```
User objects to be restored
```

```
=====
```

```
TRAF_1500000:TRAFODION.SCHBKP.SB_HISTOGRAMS
TRAF_1500000:TRAFODION.SCHBKP.SB_HISTOGRAM_INTERVALS
TRAF_1500000:TRAFODION.SCHBKP.SB_PERSISTENT_SAMPLES
TRAF_1500000:TRAFODION.SCHBKP.T1
TRAF_1500000:TRAFODION.SCHBKP.T2
TRAF_1500000:TRAFODION.SCHBKP.TI
```

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

49、恢复增量备份 TAG 'tuesday'。

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

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

50、查看表 *schbkp.t1* 和表 *schbkp.t2* 的数据。

```
>>SELECT * FROM schbkp.t1;
```

A	B
13	13
10	10
11	11

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

```
>>SELECT * FROM schbkp.t2;
```

A	B
23	23
20	20
21	21

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

51、向表 *schbkp.t1* 和表 *schbkp.t2* 插入数据 (增量变化)。

```
>>INSERT INTO schbkp.t1 VALUES (14,14);
```

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

```
>>INSERT INTO schbkp.t2 VALUES (24,24);
```

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

52、创建增量备份 TAG 'tuesday' (备份 Schema *schbkp*, 指定 *override*), 查看该备份的对象。



注意

由于指定了 *override* 且 TAG 'tuesday' 已存在, 所以 EsgynDB 将删除旧增量备份 TAG 'tuesday', 创建新增量备份 TAG 'tuesday'。

```
>>BACKUP TRAFODION, TAG 'tuesday', SCHEMA(schbkp),
INCREMENTAL, OVERRIDE, SHOW OBJECTS;
```

```
MetaData objects to be backed up
```

```
=====
```

```
TRAFODION._BACKUP_tuesday_.COLUMNS
TRAFODION._BACKUP_tuesday_.INDEXES
TRAFODION._BACKUP_tuesday_.KEYS
TRAFODION._BACKUP_tuesday_.OBJECTS
TRAFODION._BACKUP_tuesday_.REF_CONSTRAINTS
TRAFODION._BACKUP_tuesday_.SEQ_GEN
TRAFODION._BACKUP_tuesday_.TABLES
TRAFODION._BACKUP_tuesday_.TABLE_CONSTRAINTS
TRAFODION._BACKUP_tuesday_.TEXT
TRAFODION._BACKUP_tuesday_.UNIQUE_REF_CONSTR_USAGE
TRAFODION._BACKUP_tuesday_.VIEWS
TRAFODION._BACKUP_tuesday_.VIEWS_USAGE
TRAFODION._BACKUP_tuesday_.COLUMN_PRIVILEGES
TRAFODION._BACKUP_tuesday_.OBJECT_PRIVILEGES
TRAFODION._BACKUP_tuesday_.SCHEMA_PRIVILEGES
TRAFODION._BACKUP_tuesday_.SB_HISTOGRAMS
TRAFODION._BACKUP_tuesday_.SB_HISTOGRAM_INTERVALS
TRAFODION._BACKUP_tuesday_.SB_PERSISTENT_SAMPLES
```

```
User objects to be backed up
```

4. 示例

```
=====
TRAFODION.SCHBKP.SB_HISTOGRAMS
TRAFODION.SCHBKP.SB_HISTOGRAM_INTERVALS
TRAFODION.SCHBKP.SB_PERSISTENT_SAMPLES
TRAFODION.SCHBKP.T1
TRAFODION.SCHBKP.T2
TRAFODION.SCHBKP.TI

--- SQL operation complete.
```

53、执行增量备份 TAG 'tuesday'。

```
>>BACKUP TRAFODION, TAG 'tuesday', SCHEMA(schbkp),
INCREMENTAL, OVERRIDE;
--- SQL operation complete.
```

54、查看所有备份的信息。

```
>>GET ALL BACKUP TAGS, SHOW DETAILS;

BackupTag                               BackupTime
BackupStatus  BackupOperation
=====
=====
tuesday                               2018-05-14:06:32:02
VALID                                INCREMENTAL
fulldb_00212393035662759229  2018-05-14:05:27:42
VALID                                REGULAR
monday_00212393038085889128  2018-05-14:06:08:05
VALID                                REGULAR
sunday_00212393036364859822  2018-05-14:05:39:24
VALID  INCREMENTAL (IMPORTED)

--- SQL operation complete.
```

55、查看恢复增量备份 TAG 'tuesday' 时, 将被恢复的所有对象。

```
>>RESTORE TRAFODION, TAG 'tuesday', SHOW OBJECTS;
```

```
MetaData objects to be restored
```

```
=====
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.COLUMNS
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.COLUMN_PRIVILEGES
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.INDEXES
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.KEYS
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.OBJECTS
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.OBJECT_PRIVILEGES
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.REF_CONSTRAINTS
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.SB_HISTOGRAMS
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.SB_HISTOGRAM_INTERVALS
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.SB_PERSISTENT_SAMPLES
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.SCHEMA_PRIVILEGES
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.SEQ_GEN
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.TABLES
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.TABLE_CONSTRAINTS
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.TEXT
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.UNIQUE_REF_CONSTR_USAGE
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.VIEWS
```

```
TRAF_RSRVD_3:TRAFODION._BACKUP_tuesday_.VIEWS_USAGE
```

```

User objects to be restored
=====

TRAF_1500000:TRAFODION.SCHBKP.SB_HISTOGRAMS
TRAF_1500000:TRAFODION.SCHBKP.SB_HISTOGRAM_INTERVALS
TRAF_1500000:TRAFODION.SCHBKP.SB_PERSISTENT_SAMPLES
TRAF_1500000:TRAFODION.SCHBKP.T1
TRAF_1500000:TRAFODION.SCHBKP.T2
TRAF_1500000:TRAFODION.SCHBKP.TI

--- SQL operation complete.

```

56、恢复增量备份 TAG 'tuesday'。

```

>>RESTORE TRAFODION, TAG 'tuesday';
--- SQL operation complete.

```

57、查看表 *schbkp.t1* 和表 *schbkp.t2* 的数据。

```

>>SELECT * FROM schbkp.t1;

A          B
-----
          13          13
          14          14
          10          10
          11          11

--- 4 row(s) selected.

```

4. 示例

```
>>SELECT * FROM schbkp.t2;
```

```
A          B  
-----  
  
          23          23  
          24          24  
          20          20  
          21          21
```

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

5. 备份与恢复的影响

5.1 性能

- 在执行备份/恢复操作期间，所有受影响的对象将在内部暂时被锁定。根据备份/恢复（全量或子集）操作涉及的对象的数量，锁定时间为几秒或几分钟。在锁定间，内存中的数据将写入至磁盘，任何修改数据库的事务操作将被暂时中止，应用程序可能超时并报错。

由于备份操作占用更多磁盘和 CPU 资源，因此建议在数据库维护期间或数据库负载较低时执行备份操作。注意，此时数据库无需离线（除了执行全量恢复操作）。

- 备份数据集
存储在实例上的快照数量会影响 EsgynDB 的性能。如果性能下降，可以使用 DROP 命令减少备份的数量。
- 恢复备份操作涉及的对象数量决定恢复备份操作的耗时。

5.2 存储

备份操作使用额外的系统存储空间，因此数据库管理员应定期检查存储使用情况，并根据数据保留政策删除旧备份。

6. 事件日志

与备份恢复相关的事件记录在 `trafodion.sql.java.log`，您可以通过 EsgynDB Manager 监测信息事件。

示例

备份日志

```
2017-11-16 11:07:30,702 INFO pit.BackupRestoreClient:  
ENTER doSnapshot Backup Tag: fullDB Thread ID 539  
TableName: TRAFODION.MYSCH1.MYT1 SnapshotPath:  
3f3d8709cf8a1a91db9bef710ce744cc SnapshotName:  
TRAFODION.MYSCH1.MYT1_SNAPSHOT_fullDB_1584190346868838
```

```
2017-11-16 11:07:31,027 INFO pit.BackupRestoreClient:  
EXIT doSnapshot thread 539 TableName  
TRAFODION.MYSCH1.MYT1
```

```
2017-11-16 11:07:31,028 INFO pit.BackupRestoreClient:  
ENTER doSnapshot Backup Tag: fullDB Thread ID 539  
TableName: TRAFODION.MYSCH1.MYT2 SnapshotPath:  
e85c32d70349147539642aab9b04744d SnapshotName:  
TRAFODION.MYSCH1.MYT2_SNAPSHOT_fullDB_1584190346868838
```

```
2017-11-16 11:07:31,354 INFO pit.BackupRestoreClient:  
EXIT doSnapshot thread 539 TableName  
TRAFODION.MYSCH1.MYT2
```

```
2017-11-16 11:07:31,354 INFO pit.BackupRestoreClient:  
ENTER doSnapshot Backup Tag: fullDB Thread ID 539  
TableName: TRAFODION.MYSCH1.MYT3 SnapshotPath:  
12b6693d35bd55914fc9c104f0ac4843 SnapshotName:  
TRAFODION.MYSCH1.MYT3_SNAPSHOT_fullDB_1584190346868838
```

```
2017-11-16 11:07:31,682 INFO pit.BackupRestoreClient:  
EXIT doSnapshot thread 539 TableName  
TRAFODION.MYSCH1.MYT3
```

恢复日志

2017-11-16 16:55:13,861 INFO pit.BackupRestoreClient:
Restore User tag: abc Backup Type: SUBSET SnapshotName:
d01d73b7bbdc6d3b6c5582fb099760da TableName:
TRAFODION.SCHBKP.SB_HISTOGRAMS

2017-11-16 16:55:13,861 INFO pit.BackupRestoreClient:
Restore User tag: abc Backup Type: SUBSET SnapshotName:
af581139af09c13d2f076a3995a5a311 TableName:
TRAFODION.SCHBKP.SB_HISTOGRAM_INTERVALS

2017-11-16 16:55:13,861 INFO pit.BackupRestoreClient:
Restore User tag: abc Backup Type: SUBSET SnapshotName:
561bb7fee46bc4d664989facd74ee341 TableName:
TRAFODION.SCHBKP.SB_PERSISTENT_SAMPLES

2017-11-16 16:55:13,861 INFO pit.BackupRestoreClient:
Restore User tag: abc Backup Type: SUBSET SnapshotName:
1d8983a6ddc8071c07d308d59dcf8940 TableName:
TRAFODION.SCHBKP.T

2017-11-16 16:55:13,861 INFO pit.BackupRestoreClient:
Restore User tag: abc Backup Type: SUBSET SnapshotName:
a01902a798520d33ac723191eab3b59e TableName:
TRAFODION.SCHBKP.T1

2017-11-16 16:55:13,861 INFO pit.BackupRestoreClient:
Restore User tag: abc Backup Type: SUBSET SnapshotName:
e1d510f6af08526b8c4bff387dd54b09 TableName:
TRAFODION.SCHBKP.T2

2017-11-16 16:55:13,861 INFO pit.BackupRestoreClient:
Restore User tag: abc Backup Type: SUBSET SnapshotName:
125e190c1c1cf2d503a5b31f83524532 TableName:
TRAFODION.SCHBKP.