Data Sync - Tracking DSS data changes

Last updated by | Keith Elmore | Apr 5, 2021 at 7:57 AM PDT

Contents

Data Sync - Tracking DSS data changes

Scenario

Customers ask for the capability of tracking the Azure SQL Data Sync Service data changes. Since we do not have built-in feature in DSS to provide this option, customers need to build their own solutions.

To help, we verified that If your DSS login is unique, using the trigger/auditing table could provide the functionality of tracking the DSS data changes.

Below is a sample script for your reference to help build your own DSS data tracking solutions. The idea is to create the trigger/auditing table for each of the tables in the sync group schema, on the hub and member databases after the initial provisioning. When the sync data task is executed, the triggers will be fired, and the auditing tables updated. You can check the auditing table to track the data changes by the unique DSS login ID after the task, and compare the auditing tables between the hub and member databases.

```
-- table
CREATE TABLE [dbo].[TestEMP]
(
    [EmpID] [int] NOT NULL PRIMARY KEY,
    [LastName] [varchar](255) NULL,
    [FirstName] [varchar](255) NULL,
    [Address] [varchar](255) NULL
);
go

-- Auditing table
create table TestEMP_Audit
(
    EmpID int,
    [LastName] [varchar](255) NULL,
    [FirstName] [varchar](255) NULL,
    [Address] [varchar](255) NULL,
    Activity varchar(20),
    DoneBy varchar(50),
    Date_Time datetime NOT NULL DEFAULT GETDATE()
);
Go

-- Trigger
create trigger TestEMP_trigger on TestEMP
after UPDATE, INSERT, DELETE as
declare @user varchar(20), @activity varchar(20);
SET @user = SYSTEM_USER;
if exists(SELECT * from inserted) and exists (SELECT * from deleted)
begin
    SET @activity = 'UPDATE';
    INSERT TestEMP_Audit
        (EmpID,LastName,FirstName, Address, Activity, DoneBy)
    SELECT EmpID, LastName, FirstName, Address, @activity, @user FROM Deleted
    INSERT TestEMP_Audit
        (EmpID,LastName,FirstName,Address, Activity, DoneBy)
    SELECT EmpID, LastName, FirstName, Address, @activity, @user FROM inserted
end
If exists (Select * from inserted) and not exists(select * from deleted)
begin
    SET @activity = 'INSERT';
    INSERT TestEMP_Audit
        (EmpID,LastName,FirstName,Address, Activity, DoneBy)
    SELECT EmpID, LastName, FirstName, Address, @activity, @user FROM inserted
end
If exists(select * from deleted) and not exists(select * from inserted)
begin
    SET @activity = 'DELETE';
    INSERT TestEMP_Audit
        (EmpID,LastName,FirstName,Address, Activity, DoneBy)
    SELECT EmpID, LastName, FirstName, Address, @activity, @user FROM Deleted
End
```

Tracking knowledge changes

```
CREATE TABLE [DataSync].[scope_info_dss_AUDIT](
  [scope_local_id] [int],
  [scope_id] [uniqueidentifier],
  [sync_scope_name] [nvarchar](100),
  [scope_sync_knowledge] [varbinary](max),
  [scope_tombstone_cleanup_knowledge] [varbinary](max),
  [scope_config_id] [uniqueidentifier],
  [scope_restore_count] [int],
  [scope_user_comment] [nvarchar](max),
  Activity varchar(20),
  DoneBy varchar(50),
  Date_Time datetime NOT NULL DEFAULT GETDATE()
)
GO

CREATE TRIGGER [DataSync].[scope_info_trigger] on [DataSync].[scope_info_dss]
AFTER UPDATE, INSERT, DELETE AS
DECLARE @user varchar(20), @activity varchar(20);
SET @user = SYSTEM_USER;
IF exists(SELECT * from inserted) and exists (SELECT * from deleted)
BEGIN
    SET @activity = 'UPDATE FROM';
    INSERT [DataSync].[scope_info_dss_AUDIT]
        ([scope_local_id],[scope_id],[sync_scope_name],[scope_sync_knowledge],[scope_tombstone_cleanup_knowled
    SELECT [scope_local_id],[scope_id],[sync_scope_name],[scope_sync_knowledge],[scope_tombstone_cleanup_knowl
    SET @activity = 'UPDATE TO';
INSERT [DataSync].[scope_info_dss_AUDIT]
        ([scope_local_id],[scope_id],[sync_scope_name],[scope_sync_knowledge],[scope_tombstone_cleanup_knowled
    SELECT [scope_local_id],[scope_id],[sync_scope_name],[scope_sync_knowledge],[scope_tombstone_cleanup_knowl
END
IF exists (Select * from inserted) and not exists(select * from deleted)
BEGIN
    SET @activity = 'INSERT';
    INSERT [DataSync].[scope_info_dss_AUDIT]
        ([scope_local_id],[scope_id],[sync_scope_name],[scope_sync_knowledge],[scope_tombstone_cleanup_knowled
    SELECT [scope_local_id],[scope_id],[sync_scope_name],[scope_sync_knowledge],[scope_tombstone_cleanup_knowl
END
IF exists(select * from deleted) and not exists(select * from inserted)
BEGIN
    SET @activity = 'DELETE';
    INSERT [DataSync].[scope_info_dss_AUDIT]
        ([scope_local_id],[scope_id],[sync_scope_name],[scope_sync_knowledge],[scope_tombstone_cleanup_knowled
    SELECT [scope_local_id],[scope_id],[sync_scope_name],[scope_sync_knowledge],[scope_tombstone_cleanup_knowl
END
```

How good have you found this content?