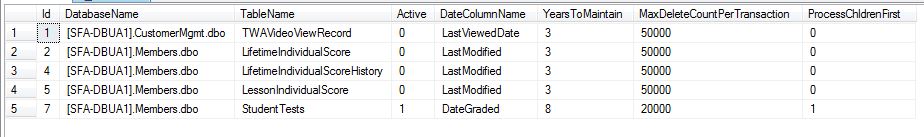
Archiver

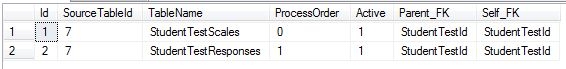
The Archiver is a set of SQL procedures and tables built to archive data from a defined set of database tables into a separate set of tables within the Archiver database.

Setup:

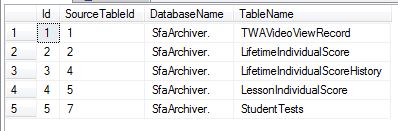
1. SourceTable – Defines where the data is coming from
   * DatabaseName sets what Db to look in
   * TableName sets the table to archive
     1. The parts are put together ‘[SFA-DBUA1].Members.dbo.StudentTests’
   * Active tells the process to either archive the table or not
   * DateColumnName tells the process which column to use when determining what rows to archive
   * YesraToMaintain tells the process how many years to keep in the source database
   * MaxDeleteCountPerTransaction tells the process how big the transaction chunks should be. Instead of processing millions of rows at once and jamming up the transaction log we chunk it up into more manageable sections
   * ProcessChildrenFirst if the table has FK dependencies then archive off the other tables before this defined table



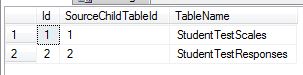
1. SourceChildTables – Defiens where the data is coming from, child tables would be relational data often referenced by a FK
   * SourceTableId is the Id of the parent table (SourceTable)
   * TableName is the name of the table to be archived, it’s implied that the database is the same as the parent
   * ProcessOrder is the order in which the tables are archived
   * Active tells the process to either archive the table or not
   * Parent\_FK tells the process which column to use in the parent table when selecting which rows to archive
   * Self\_FK tells the process which column to match on from the parent table to the child table



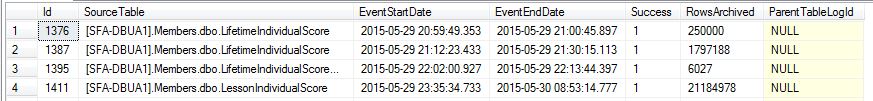
1. TargetTable – Defines where the data is going to
   * SourceTableID used to match the SourceTable to the TargetTable
   * DatabaseName tells the process what database is being used to store the archived data
   * TableName tells the process what table to store archived data in



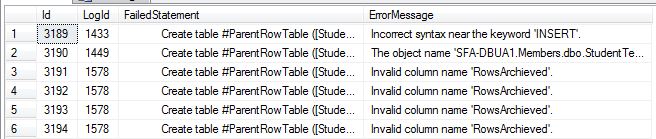
1. TargetChildTable – Defines where the data is going to, child tables would be relational data often referenced by a FK
   * SourceChildTableId used to link the SourceChildTable to the TargetChildTable
   * TableName tells the process what table to store the archived data in



1. Log – stores execution times and archive counts



1. ErrorTransactions – Stores any generated errors from the process



1. Stored Procedures
   * \_ProcessArchive – This runs the whole process.
     1. Loops through active items in the SourceTable
     2. Dynamic scripts are generated and processed item by item
        1. Creates a log of the transactions
        2. Collects how many rows are needing to be archived based on the YearsTomaintain column of the SourceTable. Processes MaxDeleteCountPerTransaction until there are no records left to process
        3. Updates log for rows archived
   * ColumnFetcher
     1. Used in the generation of dynamic SQL
   * CreateMultiTableScript
     1. Used when there are child tables to be archived before the parent can be archived
   * CreateSingleTableScript
     1. Used on single table archives.

Dynamic Script Example – CreateMultiTableScript – this is the output from the stored procedure. This gets executed in the \_ProcessArchive stored procedure

CREATE TABLE #ParentRowTable

(

[StudentTestId] INT,

[SchoolTestID] INT,

[StudentID] INT,

[Score] DECIMAL(6, 2),

[NumberCorrect] DECIMAL(6, 2),

[TargetGrowth] DECIMAL(6, 2),

[TargetScore] DECIMAL(6, 2),

[OriginalScore] VARCHAR(30),

[Notes] VARCHAR(30),

[ScoreTypeID] INT,

[TestResultsSourceId] INT,

[Graded] BIT,

[DateGraded] DATETIME,

[CustomerContactId] INT,

[LastModified] DATETIME

)

INSERT #ParentRowTable

SELECT TOP 10000 [StudentTestId],

[SchoolTestID],

[StudentID],

[Score],

[NumberCorrect],

[TargetGrowth],

[TargetScore],

[OriginalScore],

[Notes],

[ScoreTypeID],

[TestResultsSourceId],

[Graded],

[DateGraded],

[CustomerContactId],

[LastModified]

FROM [SFA-DBUA1].Members.dbo.StudentTests

WHERE DateGraded < (SELECT Dateadd(yy, -3, Getdate()))

SET IDENTITY\_INSERT SfaArchiver..StudentTestScales ON;

BEGIN TRY

DECLARE @StudentTestScales\_Log INT = 1

DELETE FROM [SFA-DBUA1].Members.dbo.StudentTestScales

OUTPUT DELETED.[StudentTestScaleId],

DELETED.[StudentTestId],

DELETED.[TestVersionScaleId],

DELETED.[TestVersionSubScaleId],

DELETED.[Score],

DELETED.[NumberCorrect],

DELETED.[SchoolId]

INTO SfaArchiver..StudentTestScales ( [StudentTestScaleId], [StudentTestId], [TestVersionScaleId], [TestVersionSubScaleId], [Score], [NumberCorrect], [SchoolId])

WHERE StudentTestId IN (SELECT StudentTestId

FROM #ParentRowTable);

WITH LogUpdater

AS (SELECT RowsArchived

FROM [Log]

WHERE Id = @StudentTestScales\_Log)

UPDATE LogUpdater

SET RowsArchived = RowsArchived + @@ROWCOUNT

END TRY

BEGIN Catch

INSERT ErrorTransactions

(LogId,

FailedStatement,

ErrorMessage)

VALUES(@StudentTestScales\_Log,

'OUTPUT DELETED.[StudentTestScaleId], DELETED.[StudentTestId], DELETED.[TestVersionScaleId], DELETED.[TestVersionSubScaleId], DELETED.[Score], DELETED.[NumberCorrect], DELETED.[SchoolId] INTO SfaArchiver. (StudentTestScales)',

Error\_message());

END Catch

SET IDENTITY\_INSERT SfaArchiver..StudentTestScales OFF;

SET IDENTITY\_INSERT SfaArchiver..StudentTestResponses ON;

BEGIN TRY

DECLARE @StudentTestResponses\_Log INT = 1

DELETE FROM [SFA-DBUA1].Members.dbo.StudentTestResponses

OUTPUT DELETED.[StudentTestResponseId],

DELETED.[SchoolTestId],

DELETED.[StudentId],

DELETED.[TestVersionQuestionId],

DELETED.[TestQuestionTypeValueId],

DELETED.[QuestionTypeValue],

DELETED.[SchoolId],

DELETED.[TestVersionId],

DELETED.[StudentTestId]

INTO SfaArchiver..StudentTestResponses ( [StudentTestResponseId], [SchoolTestId], [StudentId], [TestVersionQuestionId], [TestQuestionTypeValueId], [QuestionTypeValue], [SchoolId], [TestVersionId], [StudentTestId])

WHERE StudentTestId IN (SELECT StudentTestId

FROM #ParentRowTable);

WITH LogUpdater

AS (SELECT RowsArchived

FROM [Log]

WHERE Id = @StudentTestResponses\_Log)

UPDATE LogUpdater

SET RowsArchived = RowsArchived + @@ROWCOUNT

END TRY

BEGIN Catch

INSERT ErrorTransactions

(LogId,

FailedStatement,

ErrorMessage)

VALUES(@StudentTestResponses\_Log,

'OUTPUT DELETED.[StudentTestResponseId], DELETED.[SchoolTestId], DELETED.[StudentId], DELETED.[TestVersionQuestionId], DELETED.[TestQuestionTypeValueId], DELETED.[QuestionTypeValue], DELETED.[SchoolId], DELETED.[TestVersionId], DELETED.[StudentTestId] INTO SfaArchiver. (StudentTestResponses)',

Error\_message());

END Catch

SET IDENTITY\_INSERT SfaArchiver..StudentTestResponses OFF;

SET IDENTITY\_INSERT SfaArchiver..StudentTests ON;

BEGIN TRY

DELETE FROM [SFA-DBUA1].Members.dbo.StudentTests

OUTPUT DELETED.[StudentTestId],

DELETED.[SchoolTestID],

DELETED.[StudentID],

DELETED.[Score],

DELETED.[NumberCorrect],

DELETED.[TargetGrowth],

DELETED.[TargetScore],

DELETED.[OriginalScore],

DELETED.[Notes],

DELETED.[ScoreTypeID],

DELETED.[TestResultsSourceId],

DELETED.[Graded],

DELETED.[DateGraded],

DELETED.[CustomerContactId],

DELETED.[LastModified]

INTO SfaArchiver..StudentTests ([StudentTestId], [SchoolTestID], [StudentID], [Score], [NumberCorrect], [TargetGrowth], [TargetScore], [OriginalScore], [Notes], [ScoreTypeID], [TestResultsSourceId], [Graded], [DateGraded], [CustomerContactId], [LastModified])

WHERE StudentTestId IN (SELECT StudentTestId

FROM #ParentRowTable);

WITH LogUpdater

AS (SELECT RowsArchived

FROM [Log]

WHERE Id = 1)

UPDATE LogUpdater

SET RowsArchived = RowsArchived + @@ROWCOUNT

END TRY

BEGIN Catch

INSERT ErrorTransactions

(LogId,

FailedStatement,

ErrorMessage)

VALUES(1,

'OUTPUT DELETED.[StudentTestId], DELETED.[SchoolTestID], DELETED.[StudentID], DELETED.[Score], DELETED.[NumberCorrect], DELETED.[TargetGrowth], DELETED.[TargetScore], DELETED.[OriginalScore], DELETED.[Notes], DELETED.[ScoreTypeID], DELETED.[TestResultsSourceId], DELETED.[Graded], DELETED.[DateGraded], DELETED.[CustomerContactId], DELETED.[LastModified] INTO SfaArchiver..StudentTests ([StudentTestId], [SchoolTestID], [StudentID], [Score], [NumberCorrect], [TargetGrowth], [TargetScore], [OriginalScore], [Notes], [ScoreTypeID], [TestResultsSourceId], [Graded], [DateGraded], [CustomerContactId], [LastModified])',

Error\_message());

END Catch

SET IDENTITY\_INSERT SfaArchiver..StudentTests OFF;