



tDB2TemporalDeleteELT

Purpose

This component carries out the delete procedure for Temporal Tables in the IBM DB2.

Temporal Tables are introduced with DB2 v10 and provides mechanism to keep historical data in two time possible time dimensions:

1. Business Time period is the time range within the data a valid from a business perspective. The business time period is optional.

2. System Time period is the time range, which reflects the technical creation and modification of the data. The system time period will be automatically defined by timestamps of the database management system and is mandatory.

If a table use both time periods it's called bi-temporal table.

To get a better idea about this concept please refer to this web resource:

<http://www.ibm.com/developerworks/data/library/techarticle/dm-1204db2temporaldata/>

Talend-Integration

This component can be found in the palette under Database -> DB2 and Business Intelligence -> DB SCD.

Parameters

Connection configuration:

Property	Content
Property Type	Choose the database connection from the Metadata or use the build-in mode to setup individual configurations. (Only if use want to establish a dedicated connection for this component)
Use existing connections	True: choose an existing connection component in your job False: configure its own connection.
Host	Host (IP address or hostname) of your DB2 server. Required
Port	Port where the instance is listening. Default is 50000.
Additional JDBC Parameters	Set here semicolon separated list of key=value pairs with JDBC parameters. The default pair in this component is: "retrieveMessagesFromServerOnGetMessage=true" which cause in case of errors a readable error message instead of getting only the SQLCODE.
Database	The database you want to work with
Reference DB Schema	The database schema containing the reference table.
Username	User name
Password	Password of the user

Reference configuration

Property	Content
Reference table	Name (without schema) of the source table. The name will be set automatically if you drag and drop the component from a metadata table.
Where condition to select the reference data sets	<p>The condition to select the reference datasets. The keyword <i>where</i> is not needed here (it will be added in the SQL code generation in case of this attribute is not empty).</p> <p>Without the where condition all existing entities in the target table, which does not exist in the reference table will be temporal deleted in the target table.</p> <p>With this where condition you can specify which records have to be taken into account as reference. E.g. you can load multiple file contents (which contains always the full load and you will be able to iterate through these different file datasets and delete for each file in the target table).</p> <p>By the help of the where condition you can select additional entities, which have to be temporal deleted also if they exist in the reference.</p>
Apply the where condition to select the reference data to target	If the reference consists of data realm and the delete operation should only be applied to one of these data realm, use the where condition above to select the data realm as well as in the target table.
Where condition to additional mark the records to delete	If you reference-records contain delete marks, use this condition to select these records. They will also be temporal deleted in the target.
Schema from input table	Talend Schema chooser for the input table.
Use self defined source key	The component can take the keys from the input schema. In case of there are no keys defined in this schema (e.g. the source table is probably a view) it is possible to define the keys which are part of the source key. It is necessary to have at least one input field declared as source key. Source key field will not update and will be used to find a single unique dataset.
Columns	<p>Per source column you can specify:</p> <p><i>Is source key</i>: Check all columns, which identifies a unique source dataset.</p> <p>Visible only if option "Use self defined source key" is enabled.</p>

Target configuration

Property	Content
Target DB Schema	Database schema for the target table.
Target table	Name (without schema) of the target table.
Business Timeperiod is of data type Date	If switched on the business time period columns will work with days and ignores the time (set the time to 00:00:00)
Value Start BUS time	<p>The value (as literal or a context variable) which defines the start of the business time period (timestamp or day).</p> <p>The value has to be of a data type inheriting java.util.Date.</p>
Value End BUS time	<p>The value (as literal or a context variable), which defines the end of the business, time period (timestamp or day) in case of insert a new data set.</p> <p>The value has to be of a data type inheriting java.util.Date.</p> <p>In case of days, this has typically 2 different values depending the situation:</p> <p>The value should point to a date in the far future (like a SCD end value to specify the current valid dataset).</p>
Switch On Versioning at the Start	The component work only if the table in the versioning mode. With this option you can check that and if necessary it will be done.
Switch Off Versioning at the End	Especially in data ware house applications it is a common task to update foreign key fields in the target table or converting / transforming special values. To avoid creating historical datasets by this post processing it is possible to switch off the versioning feature at the end. Every new run of this component will switch on the versioning in case of it is switched off.

Alternative History Table Name	This table name must known to allow switch on the version mode if necessary. The default name is <reference-table-name>_HIST. If the history table as a alternative name, set it here.
--------------------------------	--

Return values

Return value	Content
ERROR_MESSAGE	Last error message
COUNT_DELETED	Number of datasets actually temporal deleted by the component

Scenario: Bi-temporal processing of the EMPLOYEE table from the DB2 examples

The component creates its own database connection and carries out a bi-temporal versioning.

In the first place it is quite enough to use only this component. It is a typical scenario to use this component in conjunction with tDB2TemporalMergeELT. At first proceed with the new and changed records and at the end use the temporal delete to proceed with the records not exists in the reference (or source).

The if-condition typically checks if the source table contains a full load or only incremental updates. In the second case the temporal delete is not necessary.



Define the connection. Because of the ELT nature of this component reference and target must be in the same database.

The reference is pretty much the same as the source in the temporal merge.

The reference is the table, which contains the full amount of records (actually the current “through”).

You can select a part of the records for using as reference data sets (e.g. multiple loads in a stage, which can be separated by criteria. In this example it is the attribute SEX. You can limit the delete operation also to the same data range in the target table with the Apply condition also to the target – option.

If the reference data also contains a delete flag, you can additionally use it to temporal delete in the target table. In this example it is the missing value for the JOB attribute.

This component does not execute any data modelling statements (except the switch on/off of the versioning). Also it depends on the usage of a business time period. If you only need the technical time stamps, it would be enough to run simple delete statements against the temporal table.

Target Configuration	
Target DB Schema	"DWH_ODS" * Target Table "EMPLOYEE" *
<input checked="" type="checkbox"/> Business time-period is of Date type	
Value Start BUS Time	context.currentCutoffDate *
Value End BUS Time	"9999-12-30" *
<input checked="" type="checkbox"/> Check and switch ON versioning at start <input checked="" type="checkbox"/> Switch OFF versioning at the end	
Alternative History Table	"EMPLOYEE_HIST"

Statements created by the component

Check if the target table is already in the temporal mode, if not switch versioning on

```
alter table DWH_ODS.EMPLOYEE add versioning use history table DWH_ODS.EMPLOYEE_HIST;
```

Delete statement

```
-- execute delete
delete from DWH_ODS.EMPLOYEE
  for portion of business_time
    from '2014-09-01' -- here can also appear an ? as prepared parameter
    to '9999-12-30' t
where not exists (
  select * from DB2INST2.EMPLOYEE s
  where (SEX='M') and not (s.JOB is null) and
    t.EMPNO = s.EMPNO
) and (SEX='M');
```

After the completing the processing (after the delete statement) and if the option to switch off the versioning at the end is on:

```
-- switch off versioning
alter table DWH_ODS.EMPLOYEE drop versioning;
```