Journal fields technically don't store data in the table that the Journal field is on, but instead, create entries in the Journal Entries [sys\_journal\_field]table. These journal entries, along with other values corresponding to changes to audited fields from the Sys Audits [sys\_audit], History [sys\_history\_line], and Record History[sys\_history\_set]tables, are displayed in the **Activity formatter**.

In order to update audit or journal entries through the ‘sys\_history\_line’ table, you need to open up security a bit so that you can change the ‘New’ field value. This can be easily accomplished by creating a new ‘write’ ACL on the ‘History [sys\_history\_line]’ table and the ‘New’ field. This ACL should limit the ‘write’ operation to the ‘admin’ role using the related list at the bottom of the ACL form.

# Incident Activity data Import R&D

In order to test if the related data on Incident record present in Remedyforce can be imported and attached to the incident records imported in ServiceNow, following steps were followed:

1. Importing Incidents
   1. Create a field ‘Remedyforce ID’ on Incident table in ServiceNow to store the unique ID of the incident which will be imported from Remedyforce.
   2. Create a staging table and load the incident data in ServiceNow.
   3. Select Incident as target table.
   4. Map the related fields in source and target tables.

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

* 1. Make Remedyforce ID a coalesce field.
  2. Create an onAfter Transform Script to import and map the created and updated fields of incident record.

Code-

(function runTransformScript(source, map, log, target /\*undefined onStart\*/ ) {

// Add your code here

var sys = target.sys\_id;

var date = source.u\_createddate;

var date1 = date.replace('T',' ');

var date2 = date1.split('.');

var gdt = new GlideDateTime(date2[0]);

gs.info(gdt)

gdt.addSeconds(-36000);

var gr = new GlideRecord('incident');

gr.addQuery('sys\_id',sys);

gr.query();

if(gr.next()){

gr.sys\_created\_on = gdt.getDisplayValue()

gr.update();

}

})(source, map, log, target);

* 1. Run transform.

1. Importing Activity Data
   1. Create a staging table and load the activity data in ServiceNow.
   2. Select Incident as target table.
   3. Map the Incident ID field from source table to Remedyforce ID field on Incident table.

Graphical user interface, application

Description automatically generated

Table

Description automatically generated with medium confidence

* 1. Make Remedyforce ID a coalesce field.
  2. Create an onAfter Transform Script to concatenate the related fields on source and map them to ‘work notes’ field on incident record.

Code –

(function runTransformScript(source, map, log, target /\*undefined onStart\*/ ) {

// Add your code here

var sys = target.sys\_id;

//target.comments\_and\_work\_notes = 'source.u\_bmcservicede\_description\_\_c';

var gr = new GlideRecord('incident');

gr.addQuery('sys\_id',sys);

gr.query();

if(gr.next()){

gr.work\_notes = source.u\_bmcservicede\_description\_\_c+" "+source.u\_bmcservicedesk\_\_actionid\_\_c+" "+source.u\_bmcservicedesk\_\_userid\_\_c;

gr.update();

}

//target.work\_notes = 'source.u\_bmcservicede\_description\_\_c';

gs.info("and here1 "+target.work\_notes);

})(source, map, log, target);

* 1. Run transform.