Create a data location referencing a non empty schema (containing Semarchy data)

This process is required when a data location has been deleted (from the workbench/application builder side) without dropping data and the user wants to recreate a new data location referencing this same schema.

This could also be used in the process of restoring a data location schema dump on another environment.

If you try to achieve this without running the following steps, when you try to create the new data location the workbench will throw an error explaining that the schema isn't empty and already contains Semarchy tables.

Instructions

To workaround the error we'll need to remove some key tables and sequences that are used by Semarchy internal mechanisms to find out if the schema already holds data location details or not.

Key elements:

1. **Tables**: All tables that start with "DL_" (DL_DATA_LOC, DL_BATCH and others. The exact list varies with the xDM release version.)
2. **Sequences**:
   a. SEQ_MATCHGRP
   b. DL_SEQ_FILTER
   c. DL_SEQ_DUPS_OPERATION

Full procedure for Semarchy xDM v4 or v5

1. Make sure to backup the current data location schema before going any further
2. Drop the DL_DATA_LOC table

```
drop table DL_DATA_LOC;
```

3. Copy the content of other DL_ tables to temporary tables (do not try to rename existing tables or you'll end up with conflicting indexes later)

Run this SQL script to generate SQL statements

```
-- Oracle
select 'create table USR_' || table_name || ' as select * from '
|| table_name || ';'
from all_tables
where
  owner = '<your_schema_name>'
  and table_name like 'DL_%';

-- PostgreSQL
select 'create table usr_' || tablename || ' as select * from '  ||
tablename || ';'
from pg_tables
where
        schemaname = '<votre_schema_de_dataloc>'
        and tablename like 'dl_%';
```

4. Drop all DL_ tables (make sure the USR_DL tables have been created and contain data)

Run this SQL script to generate SQL statements

```
-- Oracle
select 'drop table ' || table_name || ';'
from all_tables
where
  owner = '<your_schema_name>'
  and table_name like 'DL_%';

-- PostgreSQL
select 'drop table ' || tablename || ' CASCADE;'
from pg_tables
where schemaname = '<votre_schema_de_dataloc>'
and tablename like 'dl_%';
```

5. Get current sequence values and keep them (we will delete these later and restore them with the same values)

```
--Oracle
select SEQ_MATCHGRP.currval from dual;
select DL_SEQ_FILTER.currval from dual;
select DL_SEQ_DUPS_OPERATION.currval from dual;
-- PostgreSQL
SELECT last_value FROM SEQ_MATCHGRP;
SELECT last_value FROM DL_SEQ_FILTER;
SELECT last_value FROM DL_SEQ_DUPS_OPERATION;
```

6. Drop the 3 sequences

```
drop sequence SEQ_MATCHGRP;
drop sequence DL_SEQ_FILTER;
drop sequence DL_SEQ_DUPS_OPERATION;
```

7. SQL Server only: delete both the scalar-valued user function `sem_split_tbl` type as well as the user_defined table type `sem_split_tb`
8. Create a new data location from the workbench, referencing this schema. This should now be allowed because DL_ tables and sequences do not exists anymore in the schema)
9. Once the data location installation is done, all missing DL_ tables and sequences should have been recreated
10. Stop Semarchy (to prevent anyone from accessing it)
11. Restore data to DL_ tables using USR_DL_ temporary backup tables

Run this SQL script to generate SQL statements

```
-- Oracle
select 'insert into ' || table_name || ' select * from USR_' ||
table_name || ';'
from all_tables
where
  owner = '<your_schema_name>'
  and table_name like 'DL_%'
  and table_name not like 'DL_DATA_LOC';

-- PostgreSQL
select 'insert into ' || tablename || ' select * from usr_' ||
tablename || ';'
from pg_tables
where schemaname = '<votre_schema_de_dataloc>'
and tablename like 'dl_%'
and tablename not like 'dl_data_loc';
```

12. Re-align sequences values

```
ALTER SEQUENCE SEQ_MATCHGRP INCREMENT BY
<old_sequence_value_from_step5>;

-- Oracle
select SEQ_MATCHGRP.nextval from dual;
-- PostgreSQL
select nextval('SEQ_MATCHGRP');

ALTER SEQUENCE SEQ_MATCHGRP INCREMENT BY 1;

ALTER SEQUENCE DL_SEQ_FILTER INCREMENT BY
<old_sequence_value_from_step5>;

-- Oracle
select DL_SEQ_FILTER.nextval from dual;
-- PostgreSQL
select nextval('DL_SEQ_FILTER');

ALTER SEQUENCE DL_SEQ_FILTER INCREMENT BY 1;

ALTER SEQUENCE DL_SEQ_DUPS_OPERATION INCREMENT BY
<old_sequence_value_from_step5>;

-- Oracle
select DL_SEQ_DUPS_OPERATION.nextval from dual;
-- PostgreSQL
select nextval('DL_SEQ_DUPS_OPERATION');

ALTER SEQUENCE DL_SEQ_DUPS_OPERATION INCREMENT BY 1;
```

13. Restart Semarchy
14. Test the application, everything should be running ok at this stage
15. Once you made sure everything works, drop temporary backup USR_DL_ tables to clean your schema

Run this SQL script to generate SQL statements

```
-- Oracle
select 'drop table ' || table_name || ';'
from all_tables
where
  owner = '<your_schema_name>'
  and table_name like 'USR_DL_%';

-- PostgreSQL
select 'drop table ' || tablename || ';'
from pg_tables
where
  schemaname = '<your_schema_name>'
  and tablename like 'usr_dl_%';
```

Important note

When using this procedure to synchronize data across environments, you'll have to pay attention to the current value of both SEQ_LOADID and SEQ_BATCHID (repository schema), or you could end up re-processing rows that were already been submitted with the same loadid or batchid on the source environment.

Also keep in mind that Continuous loads and Notification policies are defined at the data location level. There may be external processes that rely on these elements, so make sure you are able to recreate them before deleting the data location.