Developer SOP for Database Schema Changes  
11/08/2023

**Description:**  
During the development of GSRS, developers often need to make changes to the database schema. This can be dangerous if it is not handled carefully. For example, it may cause data loss from databases or break the functionalities of the application and then leave the database in an inconsistent state. Sufficient tests are recommended in development and test environments before moving the application up to production environment. This SOP describes the end products of the schema changes in a release and testing steps recommended to developers for Substances to make sure the changes can be applied to databases safely.

**Required items:**If there are database schema changes in a release, currently we will have the following end products in GitHub.

1. A full database schema SQL script.

This is a script that generates the entire GSRS schema for a blank database. This is useful for brand new GSRS installations.

1. A delta database schema SQL script.   
   This is a script that modifies the database from a previous version of GSRS to work with the new version. This is useful when upgrading an existing GSRS installation.
2. Other SQL scripts needed. We manage the database GSRS version ourselves for now. The script that inserts rows to the version table (IX\_CORE\_DB\_GSRS\_VERSION) is part of this. You can find the links to the examples of this script in the “How to generate the required items” section.
3. README describes how to use these scripts to update the database to test and deploy.

**Current Locations for different entities:**  
Substances: <https://github.com/ncats/gsrs-ci/tree/gsrs-example-deployment/substances/database/sql> with 4 database flavors: Oracle, PostgreSQL, MariaDB and MySQL.

**How to generate the required items:**  
1. After making your code changes, test locally with this setting in gsrs-ci/substances/application.conf for the first run

spring.jpa.hibernate.ddl-auto=update

Or

spring.jpa.hibernate.ddl-auto=create

2. Generate the full database schema SQL script  
In gsrs-ci/substances/application.conf, specify the database connection settings. Then generate the full schema sql scripts file with the following settings added to the config file:

spring.jpa.properties.javax.persistence.schema-generation.create-source=metadata

spring.jpa.properties.javax.persistence.schema-generation.scripts.action=create

spring.jpa.properties.javax.persistence.schema-generation.scripts.create-target=createSQL.sql

The file createSQL.sql will be generated in gsrs-ci/substances/ folder. Make sure to rename the file each time. Otherwise, the newly generated statements will be appended to the file.

3. Generate the delta database schema SQL script  
Use a diff editing tool to compare the newly generated full database SQL script with the full database SQL script of last version to get the delta database schema SQL script.

4. Add other scripts needed. Add the script that inserts records to version table (IX\_CORE\_DB\_GSRS\_VERSION). Update README file to explain the scripts and steps needed to deploy the application. Examples of this script for different database flavors can be found on GitHub:  
Oracle:

<https://github.com/ncats/gsrs-ci/blob/fda/substances/database/sql/Oracle/GSRS_3.1/InsertVersionInfo.sql>

PostgreSQL:

<https://github.com/ncats/gsrs-ci/blob/fda/substances/database/sql/PostgreSQL/GSRS_3.1/InsertVersionInfo.sql>   
MariaDB:  
[https://github.com/ncats/gsrs-ci/blob/fda/substances/database/sql/MariaDB/GSRS\_3.1/InsertVersionInfo.sql](https://github.com/ncats/gsrs-ci/blob/fda/substances/database/sql/MariaDB/GSRS_3.1/InsertVersionInfo.sql%20)   
MySQL:  
[https://github.com/ncats/gsrs-ci/blob/fda/substances/database/sql/MySQL/GSRS\_3.1/InsertVersionInfo.sql](https://github.com/ncats/gsrs-ci/blob/fda/substances/database/sql/MySQL/GSRS_3.1/InsertVersionInfo.sql%20)

**Test Steps:**  
These test steps are written with Substances in mind. Other entities might have different steps.   
**Tests full database schema SQL script with substances service:**   
1. Create a new database schema for the test.  
2. Apply the full database schema SQL script to the database schema.  
**Note: Oracle database has extra steps. Please check the readme file in Oracle folder.** [**https://github.com/ncats/gsrs-ci/tree/gsrs-example-deployment/substances/database/sql/Oracle/GSRS\_3.1**](https://github.com/ncats/gsrs-ci/tree/gsrs-example-deployment/substances/database/sql/Oracle/GSRS_3.1)

3. Start substances with the version that corresponding with the database schema

spring.jpa.hibernate.ddl-auto=none

4. Load data and test.   
  
**Tests the delta database schema SQL script with substances service:**

The delta database schema SQL script includes the database schema changes from the last published version. For example, GSRS\_3.1\_PostgreSQL\_DELTA.sql includes the database schema changes from GSRS3.0.3 to GSRS3.1 for PostgreSQL.

Below are the steps:

1. Check out the previous version from GitHub using tags of these repos and build them.

If you are testing to upgrade to GSRS 3.1, you need to check out the GSRS 3.0.3 version at this step to test update.

1. Create a new database schema for the test.

Either use the full database schema scripts to create the database schema objects including tables, sequences, indexes, etc., or set spring.jpa.hibernate.ddl-auto to create or update to get the database schema objects created automatically.

**Note: Oracle database has extra steps. Please check the readme file in Oracle folder. https://github.com/ncats/gsrs-ci/tree/gsrs-example-deployment/substances/database/sql/Oracle/GSRS\_3.1**

If you have customized schema changed, apply those changes in this step too.

1. Load data and test to make sure it runs successfully.

Now you have a working and running substances instance of the previous version.

1. Stop the substances service.
2. Apply the delta database schema SQL scripts to database.
3. Pull the latest version of codes which correspond to the database changes from GitHub and build them. If you are testing to upgrade to GSRS 3.1, now pull the GSRS 3.1 codes.
4. Update the setting to spring.jpa.hibernate.ddl-auto=none.
5. Start substance services.
6. Reindex and test.