



Tot plugin Denodo

Control de versiones	2
Modelo de integración	3
Extracción de metadatos	3
Muestreo de datos	4
Gobierno activo	4
Edición de objetos	4
Credenciales requeridas	5
Extracción de metadatos	5
Muestreo de datos	5
Gobierno activo	5
Edición de objetos	5
Autenticación y autorización con Ldap	5
Actualización de tags	5
Despliegue	6
Configuración	6
Queries case sensitive	7
ImAri disponibles	8
Comprobar los tags disponibles en Denodo	8

Control de versiones

Versión	Fecha de modificación	Responsable	Aprobador	Resumen de cambios
1.0	22/11/2023	Anjana Producto	Anjana Producto	Creación del documento. Compatibilidad con la v4.5 de todos los módulos de Anjana
1.1	05/12/2023	Anjana Producto	Anjana Producto	Añadido el detalle para la funcionalidad de la Edición de objetos de la integración
1.2	14/02/2024	Anjana Producto	Anjana Producto	Cambiada la ari para actualización de tags en los ejemplos de configuración. Añadida nota acerca de posibles errores con el muestreo de datos

Modelo de integración

Extracción de metadatos

Para la extracción de metadata de un objeto se utilizan los métodos que ofrece el driver com.denodo.vdp.jdbc.Driver mediante los cuales se accede a la definición de esquemas y tablas.

El plugin extrae los siguientes atributos que deben llamarse igual en la tabla attribute_definition, campo name para que aparezcan en la plantilla:

- **catalog** con el valor de catalog en la base de datos
- **schema** con el valor de schema en la base de datos
- **physicalName** y **name** con el mismo valor, el nombre de la tabla
- **path** con la concatenación de los valores de catalog, schema and table
- **infrastructure** con el valor seleccionado
- **technology** con el valor seleccionado
- **zone** con el valor seleccionado
- **tags** son las etiquetas a nivel de vista que tienen las tablas.

También envía los siguientes atributos relativos a los campos del recurso pedido:

- **name** y **physicalName** con el valor del campo
- **defaultValue** con el valor por defecto definido para el campo
- **fieldDataType** con el tipo de dato definido para el campo
- **length** con el tamaño del campo
- **incrementalField** indicando si es un campo incremental
- **position** posición que ocupa el campo
- **precision** con el valor de la precisión del campo
- **nullable** indicando si el campo es nullable
- **pk** indicando si el campo es una pk
- **description** con el valor para el campo
- **tags** son las etiquetas a nivel de columna que tienen las tablas.

Los atributos a crear en Anjana deben de tener los siguientes tipos:

Nombre de atributo	Tipo de atributo
Catalog	INPUT_TEXT
Schema	INPUT_TEXT
PhysicalName	INPUT_TEXT
Path	INPUT_TEXT
Infrastructure	SELECT
Technology	SELECT
Zone	SELECT

Tags	ARRAY_ALPHANUMERICAL
Name	INPUT_TEXT
DefaultValue	INPUT_TEXT
FieldDataType	INPUT_TEXT
Length	INPUT_NUMBER
IncrementalField	INPUT_CHECKBOX
Position	INPUT_NUMBER
Precision	INPUT_NUMBER
Nullable	INPUT_CHECKBOX
Pk	INPUT_CHECKBOX
Description	ENRICHED_TEXT_AREA_INTERNATIONAL

El plugin es capaz de realizar la extracción de metadatos de los siguientes tipos de elementos de Denodo:

- Vistas de tablas de base de datos
- Interfaces
- Derived

Muestreo de datos

Utilizando el driver `com.denodo.vdp.jdbc.Driver` de Java se ejecuta una query simple de SELECT para acceder a un número limitado de elementos de la tabla para recuperar una muestra de los datos almacenados. Adicionalmente se sustituyen los valores de los campos sensibles por asteriscos.

Si algún tipo de Vista está mal definida en Denodo no se podrá realizar el muestreo. De igual manera, si se modifican las implementaciones asociadas a las Vistas de tipo Interface, después de haber sido creadas, también fallará el muestreo.

Gobierno activo

Mediante queries se permite crear/borrar los roles de Denodo cuando se aprueba o expira un DSA y se administran los permisos necesarios a la vista de la tabla (Read, Write) cuando se añaden datasets o expiran en un DSA.

Es necesario tener desplegado el plugin de Ldap para permitir crear los grupos.

Edición de objetos

Cuando se activa o desactiva una entidad no nativa, mediante queries, se dan o eliminan los permisos de las tablas correspondientes para los roles de Denodo.

Credenciales requeridas

Extracción de metadatos

Usuario con privilegios necesarios para hacer SELECT sobre las vistas de las tablas que se quieran gobernar (Connect, Metadata y Execute).

Muestreo de datos

Usuario con privilegios necesarios para hacer SELECT sobre las vistas de las tablas que se quieran gobernar (Connect, Metadata y Execute).

Gobierno activo

Para realizar las operaciones de gobierno activo son necesarios los siguientes roles en el usuario de conexión que se indique en el yml:

- create_role** -> Necesario para crear roles.
- assignprivileges** -> Necesario para realizar el grant/revoke.
- serveradmin** -> Necesario para borrar los roles.

Edición de objetos

Cuando en Anjana se active o desactive una entidad no nativa, el plugin dará o eliminará los permisos en las tablas correspondientes. Para realizar estas acciones el usuario de conexión que se indique en el yml debe tener el permiso:

- assignprivileges** -> Necesario para realizar el grant/revoke.

Autenticación y autorización con Ldap

Para que el plugin de Denodo funcione es necesario activar en Denodo la autenticación y autorización con Ldap, para ello hay que seguir estos pasos

https://community.denodo.com/docs/html/browse/8.0/en/vdp/administration/server_configuration/server_authentication/ldap_authentication/ldap_authentication

NOTA:

- Para que funcione tanto la autenticación como la autorización los usuarios no deben de existir en Denodo, solo en Ldap, así de esa manera se conecta a Ldap para autenticar y autorizar.

Actualización de tags

Es posible configurar el plugin para que actualice los tags modificados en Anjana automáticamente en Denodo a partir de todos aquellos datasets que estén gobernados y estén aprobados o deprecados. Esta configuración se hace por medio de un comando crontab que define la periodicidad en la que se quiere ejecutar la acción.

Para realizar las operaciones de actualización de tags, son necesarios los siguientes roles en el usuario de conexión que se indique en el yml:

- **manage_tags** -> Necesario para crear los tags.
- **assign_tags** -> Necesario para asignar tags a las columnas.

El siguiente ejemplo el proceso se ejecuta de lunes a viernes cada hora desde la 7:00 a las 22:00:

```
totplugin:
  batch:
    tags:
      cron: 0 0 7-22 * * MON-FRI
```

Además se tendrá que configurar la ARI de tripleta para especificar sobre qué objetos de Anjana se actuará.

```
totplugin:
  aris:
    - ari: "anja:totplugin:updateTags:/jdbc/denodo/denodo/"
```

También existe la posibilidad de configurar el separador de Anjana que indica un atributo con una lista de valores. Normalmente no hay que cambiar este valor pero si hubiera que hacerlo se podría modificar la siguiente configuración:

```
totplugin:
  anjana:
    attributeSeparator: "_-"
```

NOTA:

- El atributo tags de los datasets y dataset fields debe tener el nombre 'tags' en la tabla attribute_definition y ser del tipo ARRAY_ALPHANUMERICAL.

Despliegue

Se ha de seguir el manual genérico del despliegue de plugins.

Doc: Anjana Data x.x - DOC - Tot despliegue de plugins

Configuración

Aquí se incluye el detalle de la configuración específica del plugin.

En la Guía de Configuración técnica se explica la configuración común.

```
server:
  port: 15018

totplugin:
```

```
server:
  urls:
    - http://totserver:15000/tot/

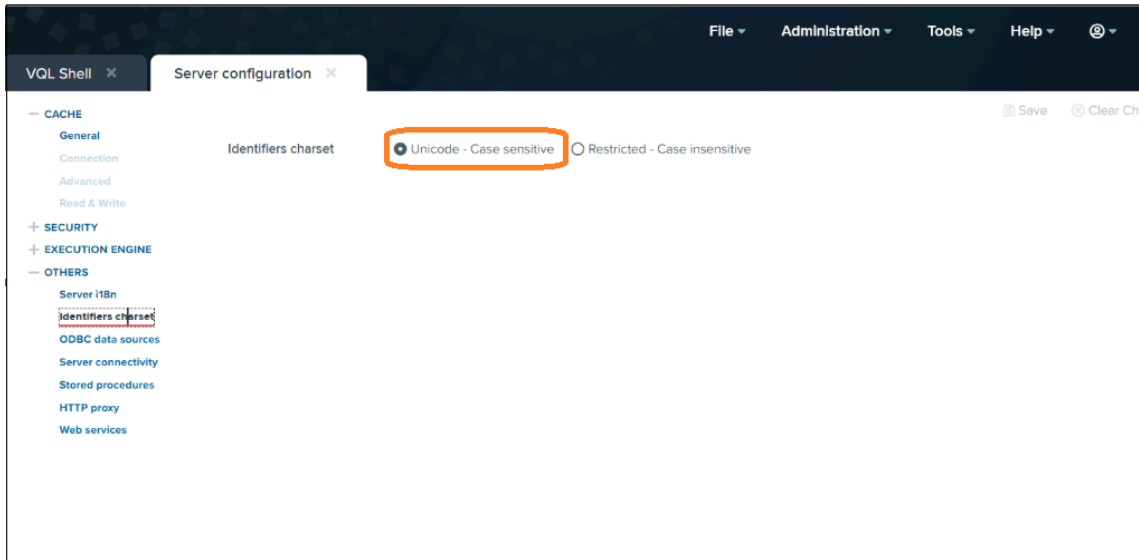
connection:
  driver: com.denodo.vdp.jdbc.Driver
  url:
jdbc:vdb://denodo.anjanadata.org:9999/dev?user=userdenodo&password=
sdenodo?ssl=false
  user: <user>
  password: <pwd>
  sampleRows: 15
  path-separator: "/"
  obfuscation-string: "*****"
aris:
  - ari: "anja:totplugin:updateTags:/jdbc/denodo/denodo/"
  - ari: "anja:totplugin:sample:/jdbc/denodo/denodo/"
  - ari: "anja:totplugin:extract:/jdbc/denodo/denodo/"
  - ari: "anja:totplugin:im:/jdbc/denodo/denodo/"
  - ari: "anja:totplugin:edit:/jdbc/denodo/devQA/"
  imAri: "anja:totplugin:im:/ldap/ldap/ldap/"
anjana:
  attributeSeparator: "_-"
  groupPrefix: Dsa_

eureka:
  client:
    serviceUrl:
      defaultZone: http://totserver:15000/tot/eureka
```

- groupPrefix indica el prefijo que tiene el nombre del grupo.
- En totplugin.connection.url hay que incluir ssl=false para que funcione la conexión con Denodo.
- No hace falta incluir el usuario y contraseña en las propiedades totplugin.user y totplugin.password si se han incluido en la url.

Queries case sensitive

El plugin de Denodo lanza las queries incluyendo los nombres de los tags, roles, schemas y datasources entre comillas dobles para que sean case sensitive pero también es necesario configurar la siguiente propiedad en el portal de Denodo:



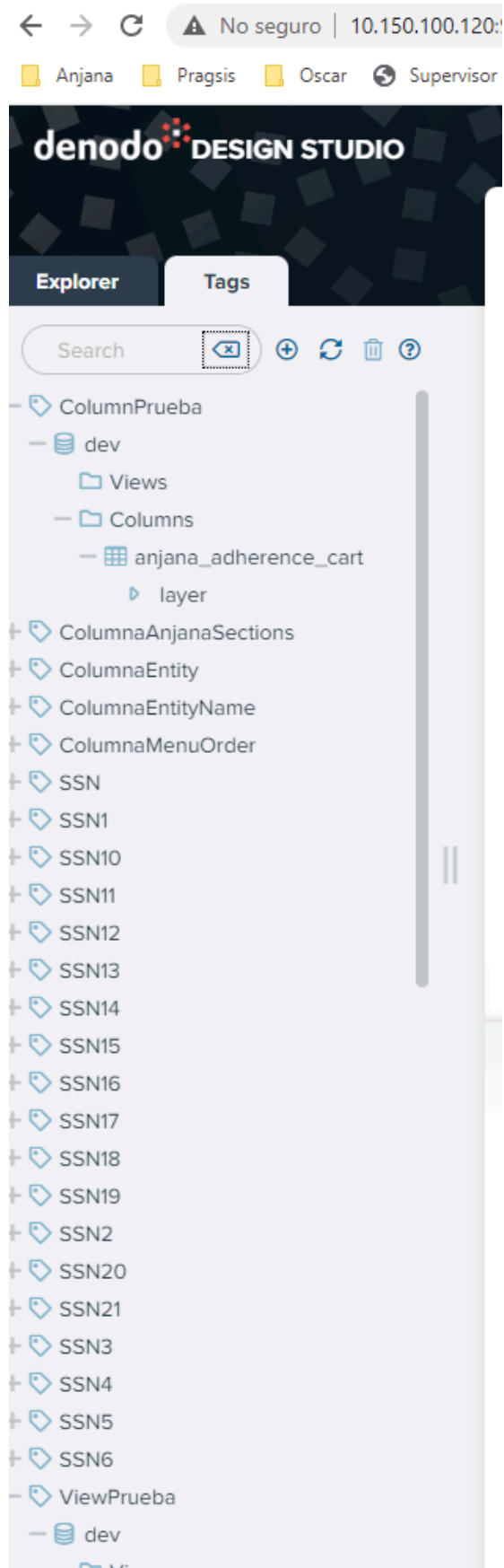
ImAri disponibles

- Ldap

Comprobar los tags disponibles en Denodo

Existen dos opciones:

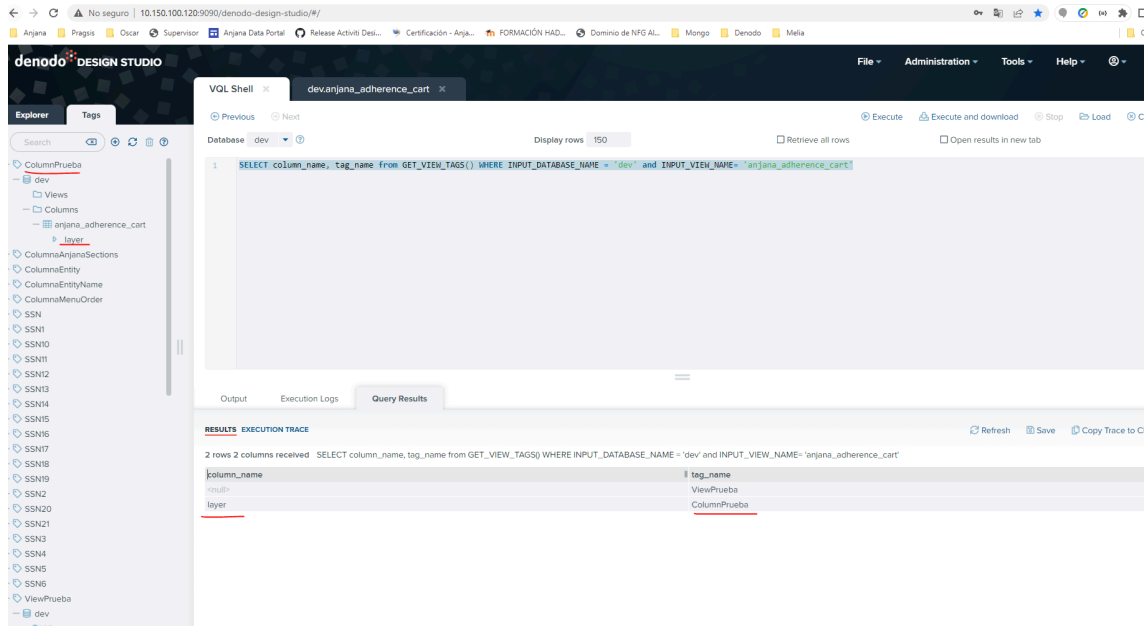
- En el apartado tags de Denodo



- Lanzando esta query

```
SELECT column_name, tag_name from GET_VIEW_TAGS() WHERE  
INPUT_DATABASE_NAME = 'dev' and INPUT_VIEW_NAME=  
'anjana_adherence_cart'
```

Aparecen así:



The screenshot shows the denodo DESIGN STUDIO interface. The VQL Shell window contains the following query:

```
1 SELECT column_name, tag_name from GET_VIEW_TAGS() WHERE INPUT_DATABASE_NAME = 'dev' and INPUT_VIEW_NAME = 'anjana_adherence_cart'
```

The Query Results tab shows the following output:

column_name	tag_name
column	ViewPrueba
layer	ColumnPrueba