Estrattore SQL

(dalla versione **4.6.10** di geowebframework, issue #1076)

L'*Estrattore SQL* è una particolare UI, impostata come *wizard*, che permette di configurare **modelli di estrazione** SQL.

Un modello di estrazione si riferisce sempre ad un particolare **dataset** (che rappresenta la fonte dei dati) e ne determina le modalità con le quali i dati ne vengono estratti.

Una volta definito un modello di estrazione, è possibile utilizzarlo per lanciare l'estrazione SQL vera e propria.

Esistono anche delle API js che permettono, di integrare ulteriori funzionalità dell'estrattore come azioni di dettaglio in lista.

Fasi del wizard:

1. creazione nuovo modello di estrazione/modifica modello di estrazione esistente
2. scelta campi da estrarre, con eventuale configurazione di alias per le intestazioni di colonna
3. scelta dei criteri di filtro
4. effettiva creazione nuovo modello/salvataggio modello esistente

Operazioni preliminari

1) Lanciare SQL script

Postgres

Creare le tabelle necessarie eseguendo gli script di base sulla **schema dei dati**.

```
-- DROP TABLE data_schema.gw_dataset;

CREATE TABLE IF NOT EXISTS data_schema.gw_dataset
(
    pk_dataset INTEGER NOT NULL,
    name_dataset CHARACTER VARYING(100) COLLATE pg_catalog."default" NOT
NULL,
    description CHARACTER VARYING(400) COLLATE pg_catalog."default",
    class_name_dataset CHARACTER VARYING(100) COLLATE pg_catalog."default",
    cod_dataset CHARACTER VARYING(100) COLLATE pg_catalog."default" NOT
NULL,
    CONSTRAINT gw_dataset_pkey PRIMARY KEY (pk_dataset),
    CONSTRAINT cod_ds_uk UNIQUE (cod_dataset)
)
WITH (
    OIDS = FALSE
```

```
)
TABLESPACE pg_default;

ALTER TABLE data_schema.gw_dataset
    OWNER TO data_schema;
```

2) Importare le classi di base

Scaricare, estrarre dall'archivio ed importare le classi di base **gw_dataset** e **gw_extraction_model**. Assegnare i permessi necessari alle classi e relative azioni.

3) Creazione Dataset

Creare una tabella/view/vista materializzata 'piatta' con tutti i campi *in chiaro* (non codificati, senza bisogno di join con tabelle esterne) cosi come vanno esportati. Creare la relativa classe in geoweb. la quale:

- necessita degli attributi, con attenzione in particolare alla correttezza del data type
- non ha bisogno di aggiungere widget in lista: se non per poter esplicitarne la **width** da utilizzare nel caso l'utente li scelta per l'estrazione (Utile quando ci sono tantissimi campi scelti, per evitare che la grid assegni la width di default, a campi come le Note, per esempio, che farebbero incrementare l'altezza della riga
- non ha bisogno di aggiungere widget nella sezione filtro: anche se si possono aggiungere determinati attributi e configurarli per poterne determinare la modalità di scelta del valore (supportati: **scelta da lista**, **scelta da finestra**, **scelta statica**. Ovviamente è esculsa la decodifica, e i vari valori in visualizzazione label, fieldToShow etc saranno i medesimi di quelli nel tracciato del db)
- non ha bisogno di configurare l'xml dei widget, anche se si può configurare un formato specifico per i widget date/time, che verrà utilizzato per l'export
- gli attributi **vanno obbligatoriamente associati** a dei gruppi attributi, con la convenzione nel nome

  ```
  'ext_'+[n_order]+'_'+[label gruppo campi]
  ```

  , dove **'ext_'** è fisso, **n_order** è un intero che determina l'ordine fra i gruppi, e **label gruppo campi** è il nome del gruppo effettivamente visualizzato.

Esempio di classe pma_tri_tax_datamart.

4) Configurare il menu della Function

Nell'xml di progetto, aggiungere i menu di terzo livello:

```
<leafItem name="gw_extraction_model" label="Modelli estrazione dati"
image="fa-solid fa-file-export" type="leafItemListFilters">
```

```
    <parameter name="attributeToFilter" value="cod_dataset"
hideToClient="false"/>
    <parameter name="className" value="gw_extraction_model"
hideToClient="false"/>
</leafItem>
```

ed opzionalmente:

```
<leafItem name="gw_dataset" label="Dataset" image="fa-solid fa-database"
type="gwClassList">
    <parameter name="className" value="gw_dataset" hideToClient="false"/>
</leafItem>
```

WIZARD

Pagina dedicata Estrattore SQL WIZARD.

API JS

Lista signature api js.

```
function openGwExtractionWizardStart(/*String*/cod_dataset, /*String*/
cod_extraction_model, /*Boolean*/ isEdit, /*String*/ stateId)

function openGwExtractionWizardChooseFields(
    cod_dataset, //String
    cod_extraction_model, //String
    stateId, //String
    selectedAttributes //Object[], {name: '', field: ''}
)

function openGwExtractionWizardPreview(
    cod_dataset, //String
    cod_extraction_model, //String
    gwState, //Object, optional
    listedAttributesNames, //String, optional, array, string comma separated
    stateId, //String, optional (used when cod_extraction_model is null)
    isOnlyPreview //Boolean
)

function createGwExtractionXLSX(
    cod_extraction_model, //optional, alternative to grid
    grid //optional, alternative to cod_extraction_model
)
```