**Logica di Aggregazione delle VIE**

La correlazione delle VIE avviene secondo i seguenti passi

Caricamento in Unico delle vie di riferimento ed aggiornamento(associazione) delle vie collegate in Totale

Caricamento del viarioUnico da UNICO

Caricamento dei dati della TOTALE da dover aggregare

Per ogni dato vengono valutati in sequenza i criteri

A0 (Codice Via Fornito dall'ente)

A1 (Associazione tra due vie di fonti diverse)

A2 (Uguaglianza descrizione)

A6/A7 (Similitudine descrizione con uguale o differente prefisso)

Se viene attivato il criterio “A00” i criteri 30,40,50,60,70,80 non sono valutati per le fonti dato con codifica toponomastica totale (**S2-UC-CONF#2.2**).

**Logica di Correlazione SOGGETTI:**

La correlazione dei Soggetti avviene secondo i seguenti passi per ogni fonte

* Caricamento da tabella totale dei record da dover associare, ovvero con

**RATING is null AND FK\_SOGGETTO is null AND REL\_DESCR is null AND ANOMALIA is null**

* Elaborazione della fonte di riferimento
* Elaborazione dei criteri di associazione per ogni fonte partendo dal criterio più forte a quello più debole. All'atto dell'associazione impostare il codice (Rating) del criterio
* Dopo aver associato un soggetto controllare i soggetti precedentemente associati e sostituire la chiave di associazione con quella nuova se il dato possiede un criterio di associazione più basso rispetto all'elemento corrente
* Viene anche valorizzato un campo ATTENDIBILITA nella SIT\_SOGGETTO\_TOTALE ad ogni nuova associazione, impostando come valore il risultato di

**1/(Rating Criterio Max Valutabile per la fonte – Rating Criterio Valutato )**

Il Rating Criterio Max Valutabile per la fonte è impostato su di un properties e quindi configurabile a seconda delle varie fonti

Il Rating Criterio Valutato è invece impostato a seconda del criterio che di volta in volta viene eseguito

**NB:** ci sarà un ulteriore criterio detto LASCO (solo nome e cognome) che sarà possibile attivare o meno da configurazione (criterio.lasco.disattivato)

#### Gestione delle Fonti dato

Le fonti dato che vengono caricate da DWH sono state suddivise per categoria :

SOGGETTO, OGGETTO, VIA, CIVICO, FABBRICATO

Tali fonti, inserite in opportuni packages verranno instanziate in automatico da una Factory, che prenderà in input la lista delle fonti da creare.

Ogni classe Fonte, estenderà una classe astratta DatoDWH e implementerà un'interfaccia relativa della tipologia della fonte (Soggetto, Via, Civico, Oggetto, Fabbricato)

La classe dato DWH definirà una serie di metodi da dover implementare nel caricatore:

**getFkEnteSorgente():** che restituisce il codice relativo alla fonte caricata

**getProgEs():** che restituisce il codice Prog\_es della fonte

**getSql(String pID)**: nella quale viene inserita la query di lettura dei dati da DWH

**getTable():** restituisce il nome della tabella interessata nel caricamento. Verrà usata per la costruzione di alcune query

**existProcessId():** restituirà valore true/false a seconda che la fonte preveda o no la gestione con processID

**queryWithParamCodEnte():** parametro true/false che serve ad indicare se la query di caricamento da DWH contiene dei parametri di input (indicati con un ? Nella query)

**prepareSaveDato:** metodo nel quale viene effettuato il mapping dei dati, la normalizzazione e dal quale viene poi invocato il salvataggio nel DB chiamando il metodo di DatoDWH **super**.saveSitViaTotale

**prepareUpdateDato:** metodo nel quale viene effettuato il mapping dei dati, la normalizzazione e dal quale viene poi invocato l'update nel DB chiamando il metodo di DatoDWH **super**.updateSitViaTotale. Tale metodo può rimanere vuoto per le fonti che hanno gestione senza ProcessID

**getDeleteSQL():** metodo che restituisce la query di cancellazione (letta da properties) nella tabella \_TOTALE relativa alla fonte instanziata. Basta semplicemente richiamare il metodo

**this**.getProperty("<nome>");specificando la properties da leggere relativa alla query

**getInsertSQL():** metodo che restituisce la query di inserimento (letta da properties) nella tabella \_TOTALE relativa alla fonte instanziata. Basta semplicemente richiamare il metodo

**this**.getProperty("<nome>");specificando la properties da leggere relativa alla query

**getUpdateSQL():** metodo che restituisce la query di aggiornamento (letta da properties) nella tabella \_TOTALE relativa alla fonte instanziata. Basta semplicemente richiamare il metodo **this**.getProperty("<nome>");specificando la properties da leggere relativa alla query

**getSearchSQL() :** metodo che restituisce la query di ricerca (letta da properties) nella tabella \_TOTALE relativa alla fonte instanziata. Basta semplicemente richiamare il metodo **this**.getProperty("<nome>");specificando la properties da leggere relativa alla query

**dwhIsDrop(Connection conn):** restituisce un flag che indica se all'atto del caricamento nel Dwh la tabella è stata o no droppata. Tale valore non viene preso in considerazione per la gestione con ProcessID

**getQuerySQLSaveProcessId():** metodo che restituisce la query di inserimento (letta da properties) nella tabella SIT\_CORRELAZIONE\_PROCESSID relativa alla fonte instanziata. Basta semplicemente richiamare il metodo **this**.getProperty("<nome>");specificando la properties da leggere relativa alla query. Tale Stringa può essere messa a null nel caso di una fonte che non ha Gestione del ProcessID

**getQuerySQLUpdateProcessId():** metodo che restituisce la query di aggiornamento (letta da properties) nella tabella SIT\_CORRELAZIONE\_PROCESSID relativa alla fonte instanziata. Basta semplicemente richiamare il metodo **this**.getProperty("<nome>");specificando la properties da leggere relativa alla query. Tale Stringa può essere messa a null nel caso di una fonte che non ha Gestione del ProcessID

**getQuerySQLNewProcessId():** metodo che restituisce la query di inserimento (letta da properties) nella tabella SIT\_CORRELAZIONE\_PROCESSID relativa alla fonte instanziata. Basta semplicemente richiamare il metodo **this**.getProperty("<nome>");specificando la properties da leggere relativa alla query. Tale Stringa può essere messa a null nel caso di una fonte che non ha Gestione del ProcessID

**getQuerySQLgetProcessId():** metodo che restituisce la query di ricerca (letta da properties) nella tabella SIT\_CORRELAZIONE\_PROCESSID relativa alla fonte instanziata. Basta semplicemente richiamare il metodo **this**.getProperty("<nome>"); specificando la properties da leggere relativa alla query. Tale Stringa può essere messa a null nel caso di una fonte che non ha Gestione del ProcessID

**getQuerySQLDeleteProcessId():** metodo che restituisce la query di cancellazione (letta da properties) nella tabella SIT\_CORRELAZIONE\_PROCESSID relativa alla fonte instanziata. Basta semplicemente richiamare il metodo **this**.getProperty("<nome>");specificando la properties da leggere relativa alla query. Tale Stringa può essere messa a null nel caso di una fonte che non ha Gestione del ProcessID

La classe DatoDwh conterrà inoltre l'implementazione dei metodi che serviranno per la gestione delle tebelle \_TOTALE.

Le interfacce relative alle 5 categorie di dati (soggetti, civici, vie, oggetti e fabbricati) serviranno per il controllo delle fonti instanziate e per la gestione di metodi relativi alla categoria di dato gestita. In particolare le interfacce conterranno un metodo da implementare che restituirà un flag (true/false) stante ad indicare se nel caricamento e successiva aggregazione dei dati dovrà essere preso in considerazione il codice originario delle vie, dei civici e dei soggetti

**Aggiornamento dei dati dell’indice**

**N.B. :**

Per la gestione dei caricamenti delle tabelle \_TOTALE, si è distinta la casistica di fonti aventi PROCESSID e fonti senza tale codice.

Nel caso di gestione con PROCESSID, viene utilizzata una tabella del DB chiamata

SIT\_CORRELAZIONE\_PROCESSID

nella quale viene mantenuto uno storico dei processId per le varie fonti caricate. Questo permette di caricare in successive operazioni solamente dati corrispondenti a nuovi caricamenti. Tale tabella avrà come chiave la terna di valori

ProcessID, fk\_ente\_sorgente, prog\_Es

Nel caso di gestione senza ProcessID si è invece scelto di caricare ad ogni operazione solamente i nuovi dati non presenti nel DB, evitando di cancellare ogni volte le tabelle \_TOTALE e \_UNICO

Per evitare problemi con l'aggiornamento dei dati si sono dovuti togliere dalla chiavi delle tabelle SIT\_SOGGETTO\_TOTALE e SIT\_VIA\_TOTALE il campo CTR\_HASH