

TALEND_ETL LEGNAIA_FURLA_TESI_BREGATA

Nome progetto	Talend Open Studio per Furla
Documento	Talend_ETL legnaia_FURLA_TESI_BREGATA v1.0
Oggetto	-
Status	Confidenziale
Versione	01.00.00
Data emissione	1/04/2019
Data revisione	-
Autore	Bregata
Destinatario	-

SOMMARIO

Obiettivo	3
Dati di partenza	3
SOLUZIONE	4
Data Ingestion	4
METADATI	5
DIM_PRODUCT	8
DIM_SHOP_ITALY	10
FACT_SALES	11
FACT_COUNTER	14
PALETTE UTILIZZATE	15
RUNNING – Caricamento TABELLE	18
Riassunto	19

OBIETTIVO

Furla è già un nostro cliente su effettuiamo attività legate al CPM e all'ETL per la parte delle Vendite.

Si vuole proporre una fase di ETL che riguarda l'utilizzo di strumenti di Advanced Analytics per ottenere una FACT_SALES più specifica con dati che seguono il sistema ACID, che attualmente il cliente non conosce e che potrebbero produrre un potenziale vantaggio economico e competitivo.

DIM_PRODUCT _PRODOTTO_ID_ int _PRODOTTO_CODICE_ varchar(100) _PRODOTTO_DESC_ varchar(100) _CODICE_LUNGO_ varchar(100) _MADE_IN_ID_ int _LOAD_ID_INS_ int _LOAD_ID_UPD_ int _DATA_AGG_ varchar(100) _MADE_IN_PIANIFICAZIONE_ID_ int _CODICE_EAN_ varchar(100) _LEAD_TIME_CUMULATIVO_ int _LEAD_TIME_PRODUZIONE_ int _MAGAZZINO_PREFERENZIALE_ID_ int _FORN_PREFERENZIALE_ID_ int _IMMAGINE_ varchar(100) _CATEGORIA_ID_ int _CATEGORIA_CODICE_ varchar(100) _CATEGORIA_DESC_ varchar(100) _MATERIALE_ID_ int _MATERIALE_CODICE_ varchar(100) _MATERIALE_DESC_ varchar(100) _COLORE_ID_ int _COLORE_CODICE_ varchar(100) _COLORE_DESC_ varchar(100) _TAGLIA_ID_ int _TAGLIA_CODICE_ varchar(100) _TAGLIA_DESC_ varchar(100) _MODELLO_ID_ int _MODELLO_CODICE_ varchar(100) _MODELLO_DESC_ varchar(100) _ALTEZZA_TACCO_ID_ int _ALTEZZA_TACCO_CODICE_ varchar(100) _ALTEZZA_TACCO_DESC_ varchar(100) _GENDER_ID_ int _GENDER_CODICE_ varchar(100) _GENDER_DESC_ varchar(100) _STAGIONE_ID_ int	FACT_SALES _FACT_SCONTRINO_ID_ bigint _NEGOZIO_ID_ int _ANNO_ID_ int _GIORNO_ varchar(100) _ORA_ int _NUMERO_SCONTRINO_ int _NUMERO_RIGA_ int _STAGIONE_ID_ int _CAUSALE_ID_ int _PRODOTTO_ID_ int _VALUTA_ID_ int _QUANTITA_SC_ int _SCONTO_SC_ varchar(100) _SCONTO_GENERICO_SC_ int _FATTURATO_U_SC_ varchar(100) _FATTURATO_MI_SC_ varchar(100) _LOAD_ID_INS_ int _LOAD_ID_UPD_ int _COSTO_VENDUTO_MI_SC_ varchar(100) _PIANO_RETTIFICHE_ID_ int _COSTO_MEDIO_SPA_ED_SC_ varchar(100) _ORARIO_ varchar(100) _NUMERO_UNIVOCO_SCONTRINO_ bigint	DIM_SHOP_ITALY _NEGOZIO_ID_ int _NEGOZIO_CODICE_ varchar(7) _NEGOZIO_DESC_ varchar(43) _INDIRIZZO_ varchar(46) _CAP_ varchar(7) _TELEFONO_ varchar(17) _CANALE_ID_ int _CANALE_CODICE_ varchar(4) _CANALE_DESC_ varchar(26) _NAZIONE_ID_ int _REGIONE_ID_ int _REGIONE_DESC_ varchar(21) _PROVINCIA_ID_ int _SUPERFICIE_ int _PARITA_ varchar(21) _LOAD_ID_INS_ int _LOAD_ID_UPD_ int	FACT_COUNTER _FACT_CONTAPERSONE_ID_ int _GIORNO_ varchar(10) _ORARIO_ varchar(7) _NEGOZIO_ID_ int _NUMERO_PERSONE_ int _LOAD_ID_INS_ int _LOAD_ID_UPD_ int
			DIM_CAUSALE _CAUSALE_ID_ int _CAUSALE_CODICE_ varchar(13) _CAUSALE_DESC_ varchar(47) _TIPO_CAUSALE_ID_ int _TIPO_CAUSALE_ROLL_INV_ varchar(4)
			DIM_MADE_IN _MADE_IN_ID_ int _MADE_IN_CODICE_ varchar(10) _MADE_IN_DESC_ varchar(100)
	DIM_MADE_IN_PIANIFICAZIONE _MADE_IN_PIANIFICAZIONE_ID_ int _MADE_IN_PIANIFICAZIONE_CODICE_ varchar(10) _MADE_IN_PIANIFICAZIONE_DESC_ varchar(100)		DIM_PROVINCIA _PROVINCIA_ID_ int _PROVINCIA_CODICE_ varchar(100) _PROVINCIA_DESC_ varchar(100)

DATI DI PARTENZA

Come dimensioni sono state create le seguenti tabelle ma prive di dati all'interno, perché andremo noi a riempirle tramite l'uso di Talend:

DATABASE TABLE	METADATA	NAME METADATA
DIM_CAUSALE	FILE: .csv	Causale
DIM_MADE_IN	FILE: .csv	Made_In
DIM_MADE_IN_PIANIFICAZIONE	FILE: .csv	Made_In_Pianificazione
DIM_PRODOTTO	FILE: .csv	Prodotto1, Prodotto2, Prodotto3,
DIM_SHOP	FILE: .excel	Negozio
FACT_SCONTRINO	FILE: .csv	Scontrino
DIM_PROVINCIA	FILE: .csv	Provincia
FACT_CONTAPERSONE	FILE: .csv	Contapersone

SOLUZIONE

DATA INGESTION

I dati sono stati estratti dai cubi TM1 del cliente ed importati su SQL server tramite esportazione/importazione di file csv, excel e nel caso della DIM_SCONTRINO direttamente dal database già creato in precedenza. Il database SQL Server è installato sulla macchina 192.168.2.14 e si trova sullo schema LEGNAIA.

La visione su DATAGRIP è illustrata nella seguente immagine.

The screenshot shows the 'General' tab of a connection configuration window in Talend Open Studio. The 'Name' field is set to 'FASHION_RETAILER@192.168.2.14'. The 'Host' field is '192.168.2.14' and the 'Port' field is empty. The 'Instance' field is empty and the 'Database' field is 'FASHION_RETAILER'. There is an unchecked checkbox for 'Use Windows domain authentication'. The 'User' field is 'sa' and the 'Password' field is masked with '<hidden>'. A 'Remember password' checkbox is checked. The 'URL' field contains 'jdbc:sqlserver://192.168.2.14;database=FASHION_RETAILER' and a dropdown menu is set to 'default'. Below the URL field, it says 'Overrides settings above' and there is a 'Test Connection' button. At the bottom, the 'Driver' is listed as 'Microsoft SQL Server'. A 'Reset' button is located in the top right corner.

METADATI

Tramite le funzionalità del software ETL Talend Open Studio è stato possibile importare vari tipi di file per costruire un Nuovo database più Semplice, Ma utile per valutare tutto il reparto vendite di LEGNAIA.

Per usare il database su Talend ho bisogno di creare una connessione al database Legnaia dove poi costruirò le mie Dimensioni e la conseguente Fact table.

Per fare ciò, ho bisogno dei Metadati, ma cosa sono?

Tutte le informazioni associate a un documento possono essere organizzate in tre macrocategorie: metadati descrittivi, strutturali e gestionali.

I **metadati descrittivi** sono quelli che descrivono il contenuto del documento. Variano ovviamente in base al contenuto. Usare un metadato per il numero di protocollo e mantenerlo vuoto in tutti i documenti perché non protocollati non avrebbe senso. Questi metadati vanno adattati al contenuto che vogliamo descrivere. Per alcune tipologie di documenti fiscali alcuni metadati sono obbligatori, al fine di conservare le giuste informazioni per la mappatura del documento e per il suo immediato reperimento in caso di necessità. Una fattura emessa che non abbia il destinatario rimane un pezzo di carta inutile, e se questo destinatario c'è va mappato tramite apposita etichetta di descrizione.

I **metadati strutturali** si riferiscono invece alla persistenza fisica e logica del documento. Possono descrivere dove è stato localizzato in una cartella, o il suo codice identificativo interno, oppure contenere l'informazione dell'integrità nel tempo del contenuto. La loro mappatura permette di tenere sotto controllo dove fisicamente risiedono i file, e in che condizioni si trovano in quanto a struttura logica.

Infine i **metadati gestionali**, che come dice il nome sottendono alla gestione dei documenti e alla loro amministrazione. Possono contenere informazioni sui diritti d'accesso, su quando e come procedere allo scarto quando sarà il momento, su chi sia il custode di quei file.

Insomma, per essere molto sintetici possiamo dire che i metadati costituiscono un corredo di informazioni ai documenti informatici, e che servono alla loro descrizione e amministrazione. Un punto di partenza attraverso il quale potremo analizzare in seguito più approfonditamente quali sequenze di metadati sia possibile utilizzare, quali sono quelli obbligatori e per quali tipi di documenti vadano utilizzati.

- Connessione al DB

Aggiorna connessione database - Passo 2/2

Devi premere il bottone verifica per verificare l'impostazione del database

Tipo DB	Microsoft SQL Server	Tipo di DB
Versione Db	Open source JTDS	Versione del DB
Stringa di connessione	jdbc:jtds:sqlserver://192.168.2.14:1433/FASHION_RETAIL;sendStringParametersAsUnicode=false	Server, Porta, e nome del Database per avviare la connessione
Login	sa	
Password	
Server	192.168.2.14	
Porta	1433	
DataBase	FASHION_RETAIL	
Schema	Fashion_Retail	
Parametri aggiuntivi	sendStringParametersAsUnicode=false	
<input type="button" value="Test connection"/> <input type="button" value="v"/>		

- File .csv

Impostazioni file

Server: Localhost 127.0.0.1

File: C:/Users/Admin/Desktop/Mediamente consulting/TEST/Furla_luca/provincia.csv

Formato: UNIX

Visualizzatore file

PROVINCIA_ID	PROVINCIA_CODICE	PROVINCIA_DESC	LOAD_ID_INS	LOAD_ID_UPD	DATA_AGG	PROVINCIA_UKID
107638	EN	"Enna"	3901861	24959378	10-GIU-2016 13:52:10	59
72313	MB	"PROVINCIA DI MB"	4760927	17069046	13-NOV-2014 08:01:18	
72314	TR	"Terni"	4760927	24959378	10-GIU-2016 13:52:10	121
72315	LC	"Lecco"	4760927	24959378	10-GIU-2016 13:52:10	71
72316	CL	"Caltanissetta"	4760927	24959378	10-GIU-2016 13:52:10	52
72317	CN	"Cuneo"	4760927	24959378	10-GIU-2016 13:52:10	53
72318	MS	"Massa Carrara"	4760927	24959378	10-GIU-2016 13:52:10	82
72319	PT	"Pistoia"	4760927	24959378	10-GIU-2016 13:52:10	97
72320	TN	"Trento"	4760927	24959378	10-GIU-2016 13:52:10	118
103	ACT	"PROVINCIA DI ACT"	-1	17069046	13-NOV-2014 08:01:18	
104	VR	"Verona"	-1	24959378	10-GIU-2016 13:52:10	130
105	RA	"Ravenna"	-1	24959378	10-GIU-2016 13:52:10	101
106	AL	"Alessandria"	-1	24959378	10-GIU-2016 13:52:10	31

- File excel

Server: Localhost 127.0.0.1

File: C:/Users/Admin/Desktop/Mediamente consulting/TEST/Furla_luca/SHOP.xlsx

☒ Read excel2007 file format(xlsx)

Generation mode: Memory-consuming(User mode)

Impostazione visualizzatore file e fogli

Imposta parametri fogli: ☒ All sheets/DSelect sheet, ☒ Sheet1

Per favore scegli foglio (struttura foglio come guida dello schema): Sheet1

_NEGOZIO_ID_	_NEGOZIO_CODICE_	_NEGOZIO_DESC_	_INDIRIZZO_	_CAP_
73472.0	"IT305"	"ITALY - **BARI - V..."	"Via Argiro, 84"	"70121"
73438.0	"IT055"	"ITALY - **BARI - V..."	"via Principe Ame..."	"70121"
73802.0	"IT346"	"ITALY - **ROMA - ..."	"via Aniene, 1"	"00189"
73717.0	"IT045"	"ITALY - **MONZA"	"via Italia, 37/3"	"20052"
73584.0	"IT528"	"ITALY - **MILAN..."	"via Angelo"	"20122"
73516.0	"IT040"	"ITALY - **MILAN..."	"P.zza Liberty, 2"	"20100"
6096834.0	"ITD01"	"ITALY - **SAN M..."	"	"

DIM_PRODUCT

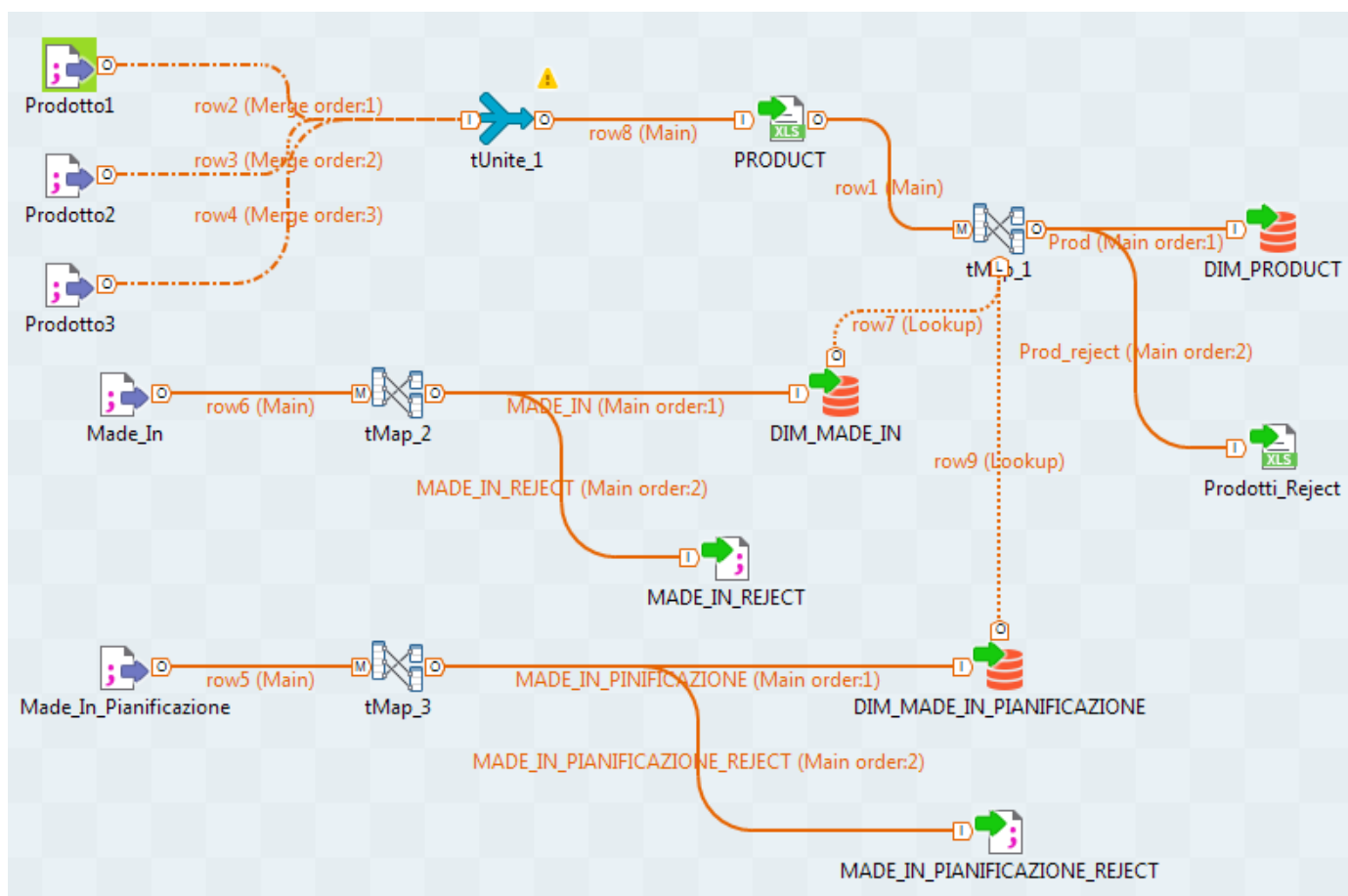
I Metadati, una volta importati, devono essere rielaborati per la creazione del database sul server. Concettualmente sono state effettuate le seguenti quattro operazioni:

1. Importazione file .csv o Excel.
2. Unione dei file tramite lo strumento tUnite, se neccessario.
3. Cambiamento e mappatura dei nomi, lunghezza o tipo degli attributi o unione di tabelle grazie alle chiavi primarie tramite lo strumento tMap.
4. Creazione della tabella in SQL SERVER tramite lo strumento tDBOutput(tMSSQLOutput)

Per svolgere le seguenti operazioni è necessario creare un nuovo JOB, che conterrà i vari metodi di importazione.

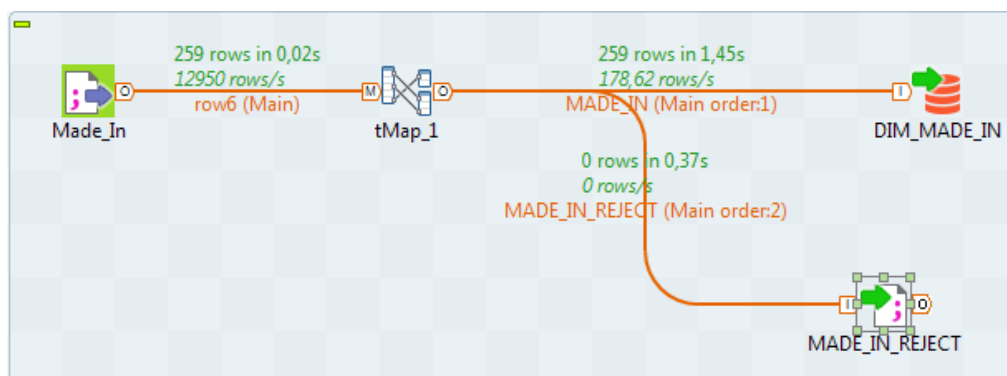
Di seguito vediamo un esempio completo descritto dalla creazione della DIM_PRODUCT, dopo aver fatto un Inner Join con le tabelle DIM_MADE_IN_PIANIFICAZIONE e DIM_MADE_IN create precedentemente.

Per ogni tabella creata si è creato un Output dove vengono caricati i file rigettati per contrastare eventuali Errori nel caricamento.

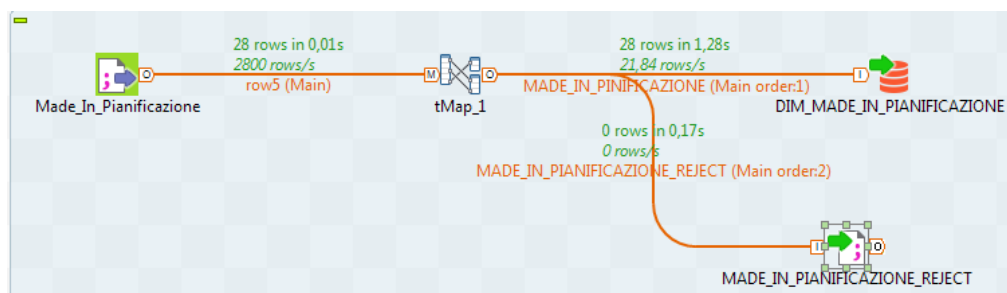


Un'altra specifica usata per controllare e prevenire gli errori possibili nel caricamento nell'ETL, è separare ogni Dimensione in un job differente.

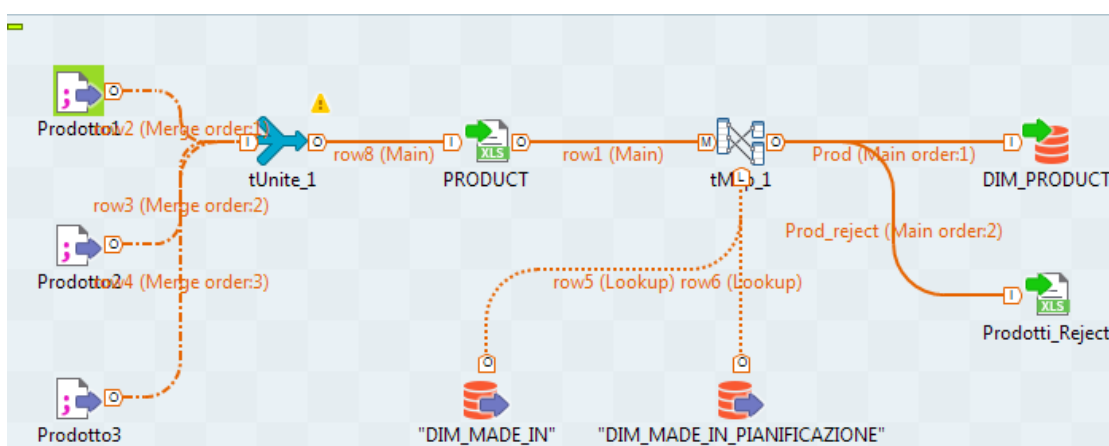
DIM_MADE_IN:



DIM_MADE_IN_PIANIFICAZIONE:

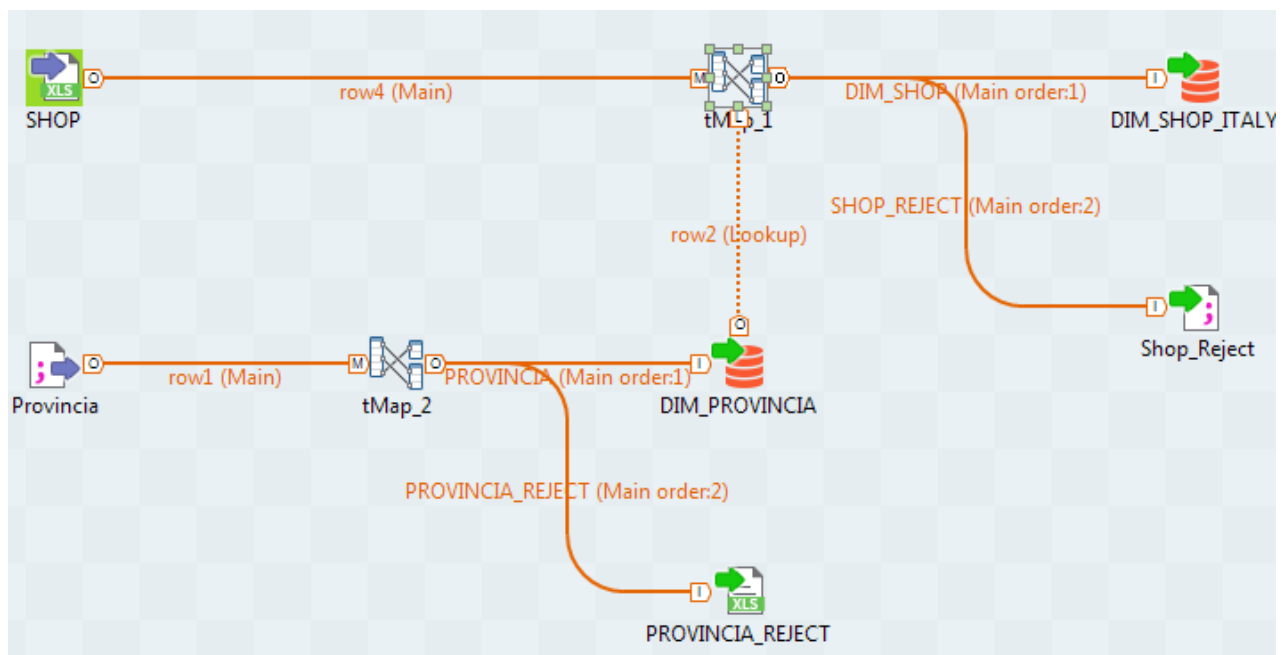


Per poi unirle Tramite L'import dei metadati direttamente dal DB sottoforma di Input.



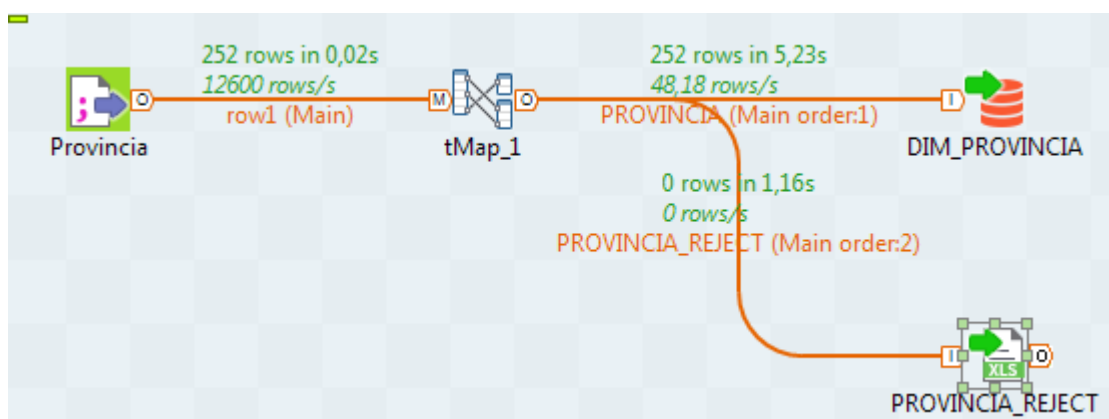
DIM_SHOP_ITALY

Per la creazione della DIM_SHOP, si eseguirà lo stesso principio elencato in precedenza, facendo un Inner Join con la tabella DIM_PROVINCIA creata precedentemente. Come in precedenza, per ogni tabella si è creata un Output dove vengono caricati i file rigettati per contrastare eventuali Errori nel caricamento.

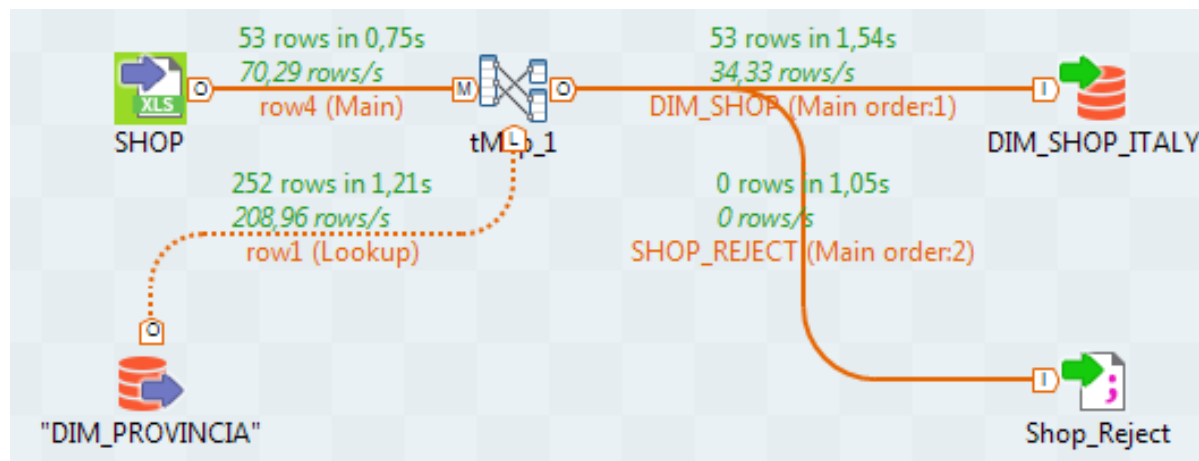


Separiamo ogni Dimensione in un job differente.

DIM_PROVINCIA:



Per poi unirle Tramite L'import dei metadati direttamente dal DB sottoforma di Input.

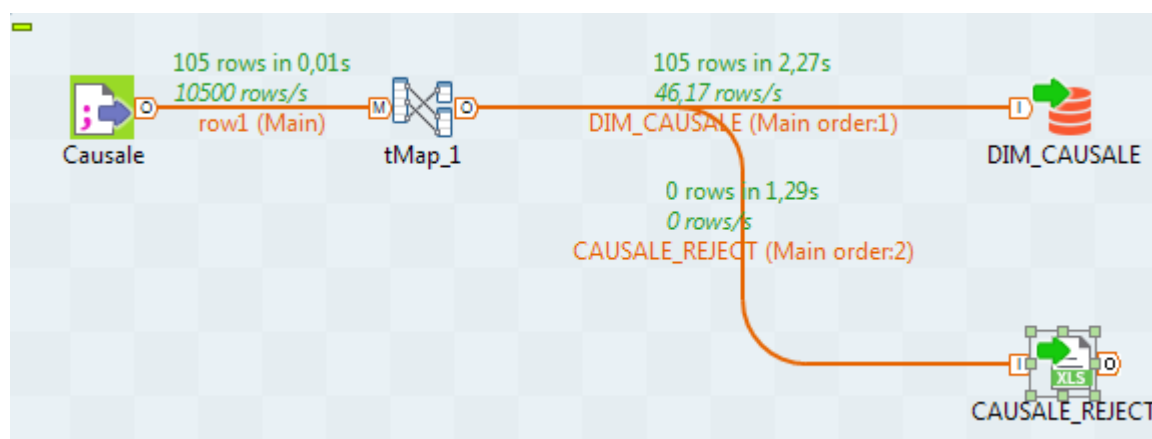


FACT_SALES

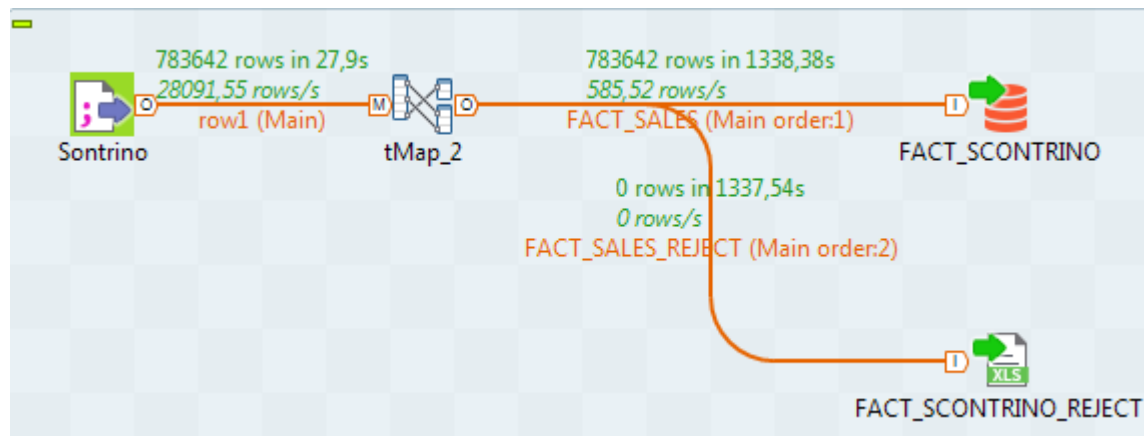
Dopo aver creato la DIM_SHOP e la DIM_PRODUCT, abbiamo tutto il necessario per costruirci la nostra FACT_SALES, facendo un Inner Join tra la tabella DIM_CAUSALE, derivante dai metadati del file csv e la DIM_SCONTRINO, anch'essa da file delimitato.

Creiamo la tabella nel Database MSSQL e la colleghiamo tramite il tMap alla FACT_SALES, che caricheremo nel DB FASHION_RETAILER.

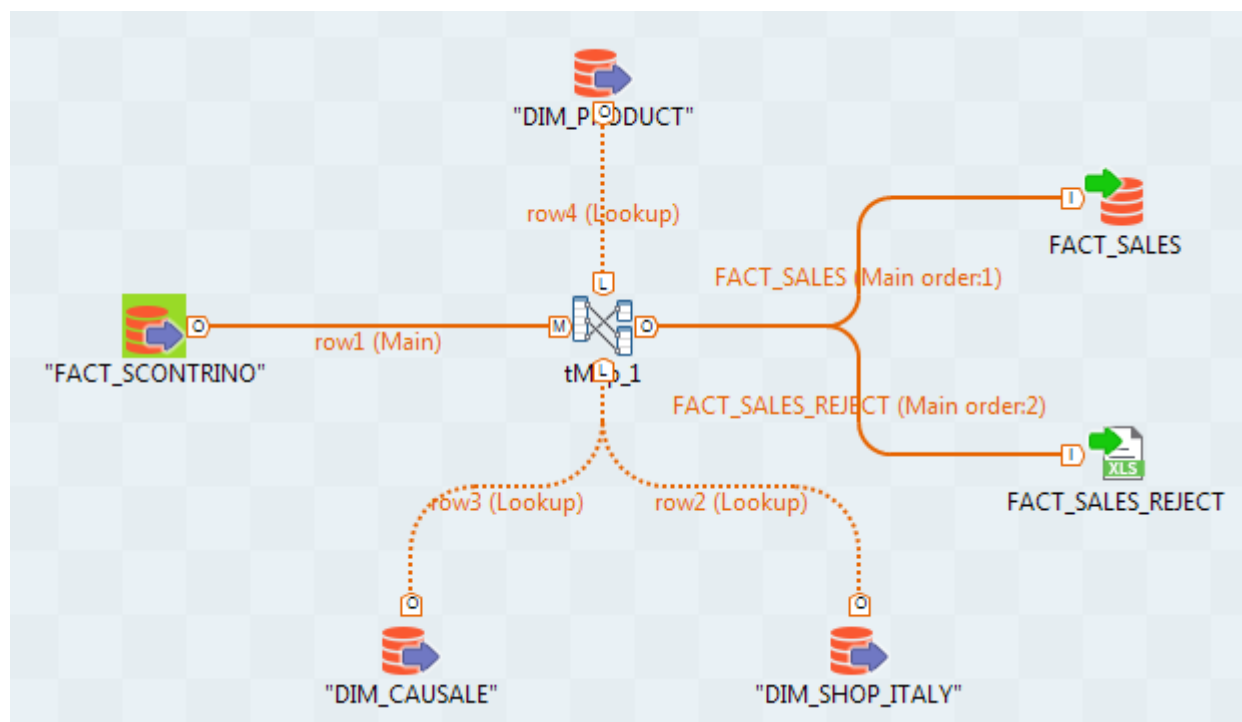
DIM_CAUSALE:



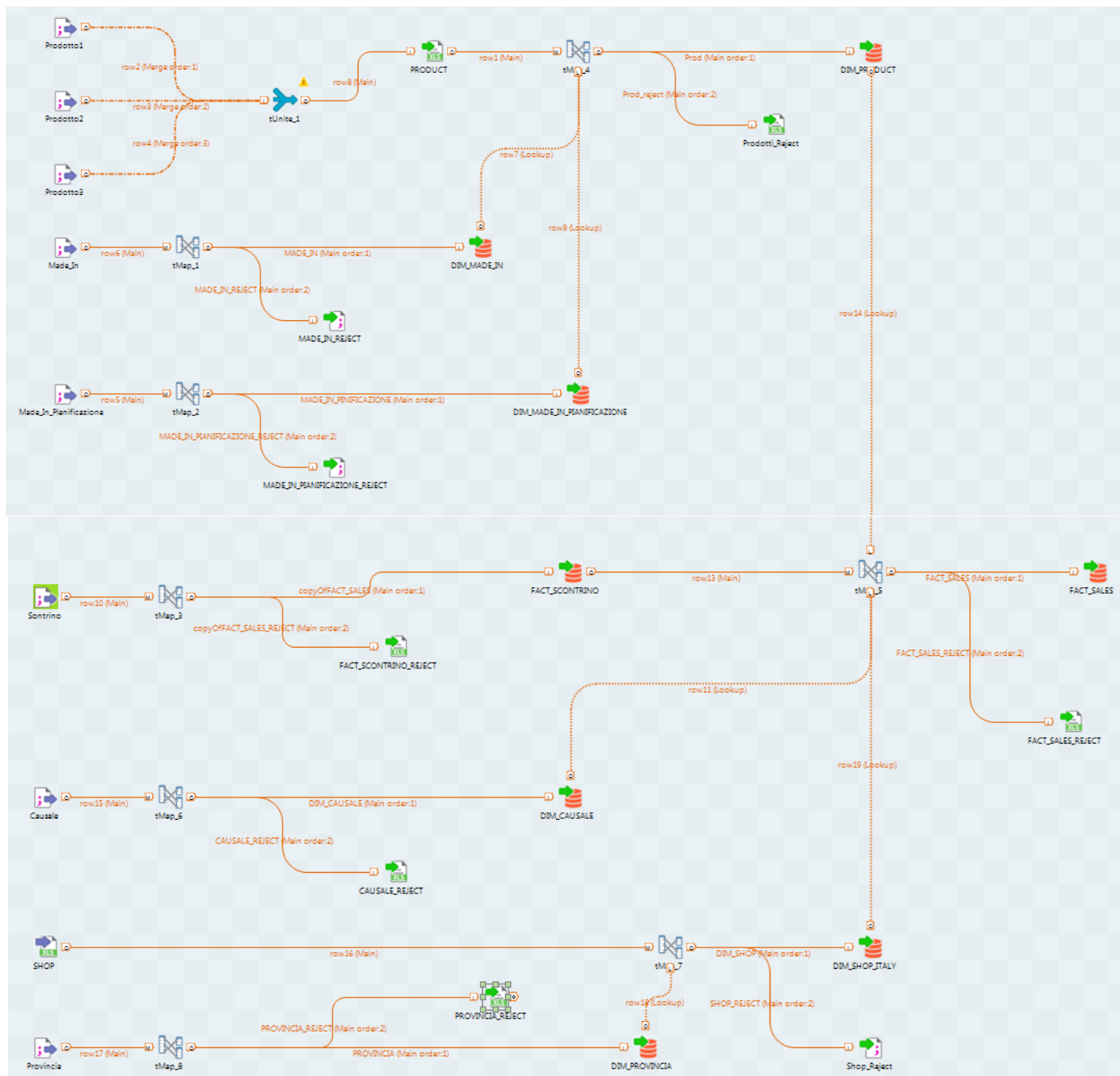
FACT_SCONTRINO:



FACT_SALES:

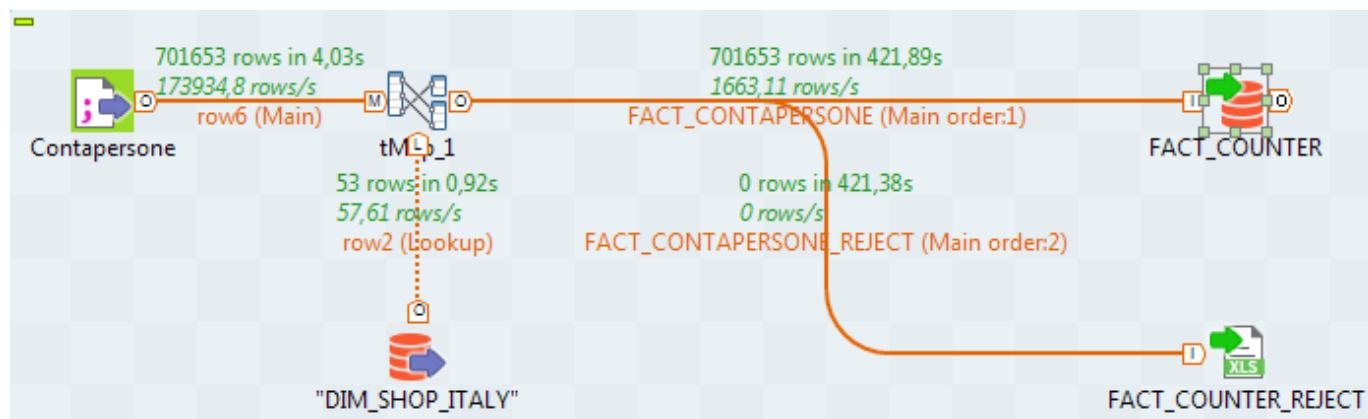


LO SCHEMA GENERALE DELLA FACT_SALES OTTENUTA È IL SEGUENTE:

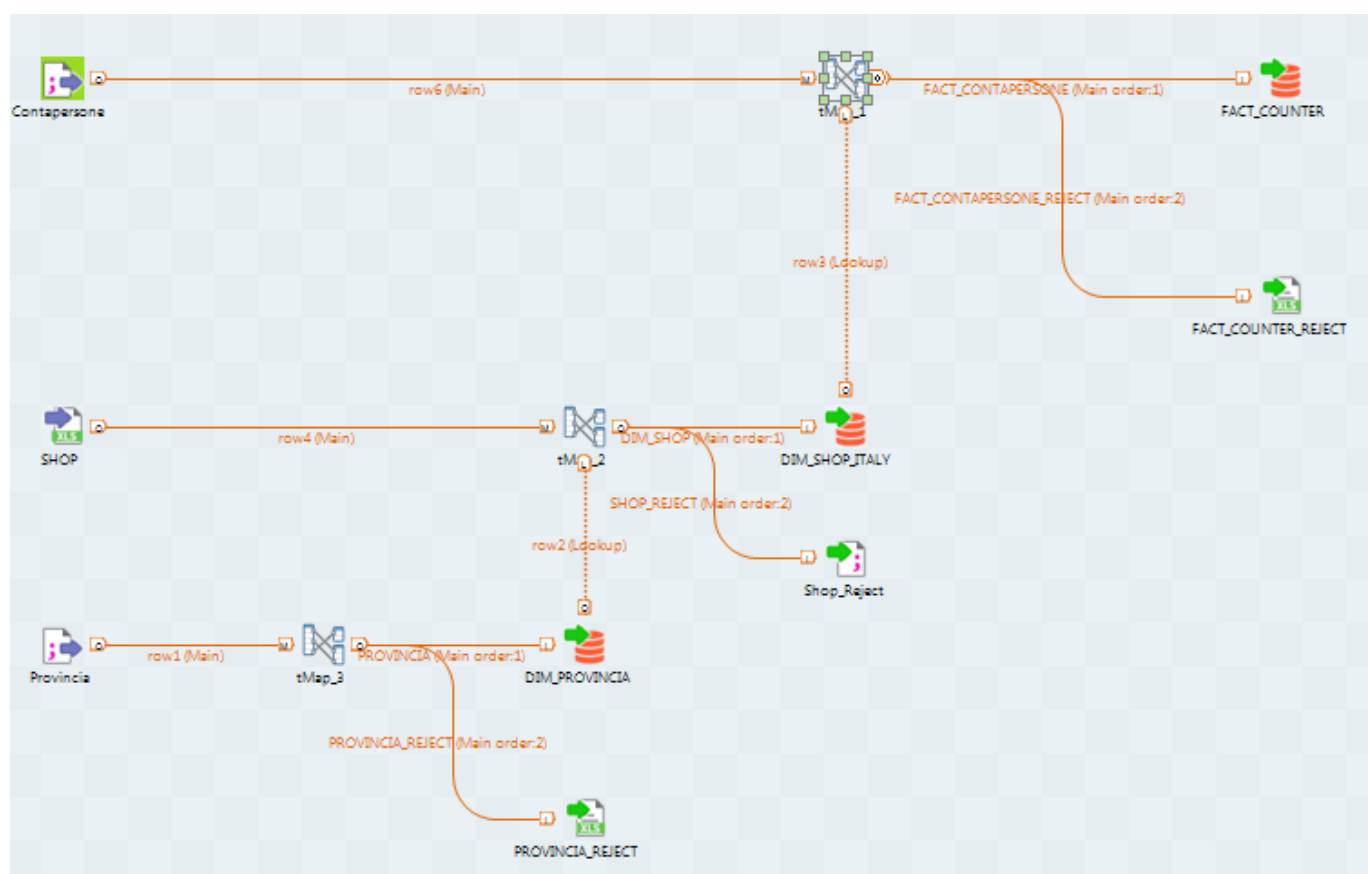


FACT_COUNTER

L'ULTIMA TABELLA DA Creare e caricare nel Database FASHION_RETAILER è la FACT counter, facendo un Inner Join tra la tabella DIM_SHOP già creata, e i record derivante dai metadati del file csv Contapersone.



Lo schema generale è il seguente:



PALETTE UTILIZZATE

- tMap



Function	tMap is an advanced component, which integrates itself as plugin to <i>Talend Studio</i> .
Purpose	tMap transforms and routes data from single or multiple sources to single or multiple destinations.

Collego gli attributi del file con gli attributi che creerò nella tabella del DB. La visualizzazione completa di essi è presente nella parte bassa della rappresentazione

INNER JOIN tramite Chiave primaria

Creazione e Output contenet e i record rigettati

Mapping

- tMSSQLOutput

FACT_COUNTER(tDBOutput_2)(Microsoft SQL Server)

Impostazioni base

Database: Microsoft SQL Server [Apply]

Tipo proprietà: Integrato [Icona]

☐ Usa una connessione esistente

JDBC Provider: Open source JTDS

Host: "192.168.2.14" Porta: "1433" Schema: "Fashion_Retail"

Database: "FASHION_RETAIL"

Username: "sa" Password: "*****"

Tabella: "FACT_COUNTER"

Azione tabella: Tronca tabella ☐ Abilita inserimento identity Azioni nei dati: Inserisci

Schema: Integrato [Edit schema] [Sync columns]

Data source
This option only applies when deploying and running in the Talend Runtime

☐ Specify a data source alias

☐ Interrompi se rilevato errore

Dati della connessione riferita al DB (MSSQL Server) Legnaia facendo riferimento allo schema corretto dove è presente la tabella vuota nel DB da riempire. Usare una proprietà di tipo Integrato per avere uno schema sempre aggiornato

Selezionare il nome della Tabella appartenente allo schema scelto su cui scriverò i dati.
ATTENZIONE! È molto importante scegliere l'azione Tronca Tabella per velocizzare il processo rimuovendo tutte le righe da una tabella o da partizioni specificate di una tabella senza registrare le eliminazioni delle singole righe

FACT_COUNTER(tDBOutput_2)(Microsoft SQL Server)

Advanced settings

Parametri aggiuntivi JDBC: ""

Committa ogni: 800000

Colonna Aggiuntiva

Nome	Tipo di dato	Espressione SQL	Posizione	Colonna referenziata

[+][X][Up][Down][Icona][Icona]

☐ Usa campi opzioni

☐ Ignore Date validation

☐ Enable debug mode

☐ Support null in "SQL WHERE" statement

☒ Use Batch Size grandezza lotto: 2000000

☐ Statistiche tStatCatcher

Le impostazioni base dell'Output sono identiche a quelle di Default. Abbiamo però le Impostazioni Avanzate che cambiano. Infatti, dopo vari tentativi, siamo arrivati a stabilire come Batch Size una grandezza del lotto a 2000000 che committa ogni 800000 record, arrivando alla massima efficienza per quanto riguarda il nostro DB.

- tFileInputExcel

SHOP(tFileInputExcel_1)

Impostazioni base

Tipo proprietà: Repository EXCEL:SHOP

☒ Read excel2007 file format(xlsx)

Nome file: "C:/Users/Admin/Desktop/Mediamente consulting/TESI/Furla_Luca/SHOP.xlsx"

☒ Tutte i fogli

Intestazione: 1 Piè di pagina: 0 Limite:

☐ Affect each sheet(header&footer)

☐ Interrompi se rilevato errore

Prima colonna: 1 Ultima colonna:

Schema: Repository EXCEL:SHOP - metadata Edit schema

Usare la REPOSITORY così nel caso di un cambiamento nei metadati il programma si aggiorna automaticamente.

- tMSSQLInput

"DIM_SHOP_ITALY"(tDBInput_1)(Microsoft SQL Server)

Impostazioni base

Database: Microsoft SQL Server Apply

☐ Usa una connessione esistente

Tipo proprietà: Integrato

JDBC Provider: Open source JTDS

Host: "192.168.2.14" Porta: "1433" Schema: "Fashion_Retail"

Database: "FASHION_RETAIL"

Username: "sa" Password: "*****"

Schema: Integrato Edit schema

Nome Tabella: "DIM_SHOP_ITALY"

Tipo query: Integrato Guess Query Guess schema

Query: "SELECT Fashion_Retail.DIM_SHOP_ITALY_NEGOZIO_ID_,
Fashion_Retail.DIM_SHOP_ITALY_NEGOZIO_CODICE_,
Fashion_Retail.DIM_SHOP_ITALY_NEGOZIO_DESC_,
Fashion_Retail.DIM_SHOP_ITALY_INDIRIZZO_,
Fashion_Retail.DIM_SHOP_ITALY_CAP_,
Fashion_Retail.DIM_SHOP_ITALY_TELEFONO_,
Fashion_Retail.DIM_SHOP_ITALY_CANALE_ID_,
Fashion_Retail.DIM_SHOP_ITALY_CANALE_CODICE_,"

- tFileInputDelimited

Contapersone(tFileInputDelimited_2)

Impostazioni base

Tipo proprietà: Repository DELIM:Contapersone

Schema: Repository DELIM:Contapersone - metadata

"When the input source is a stream or a zip file, footer and random shouldn't be bigger than 0."

Nome file: "C:/Users/Admin/Desktop/Mediamente consulting/TESI/Furla_luca/Contapersone.csv"

Separatore riga: "\n" Separatore di campo: "\t"

☐ Opzioni CSV

Intestazione: 1 Piè di pagina: 0 Limite:

☐ Tralascia righe vuote ☐ Non compresso come file zip ☐ Interrompi se rilevato errore

Usare la REPOSITORY così nel caso di un cambiamento nei metadati il programma si aggiorna automaticamente.

Fare attenzione ai separatori di campo dei dati che aiutano la separazione degli attributi e la creazione delle tabelle

RUNNING – CARICAMENTO TABELLE

Job TABELLE_DIM_LEGNAIA

Esecuzione base

☒ Statistics ☒ Salvare il job prima di eseguire

☐ Exec time ☒ Clear before run

Impostazioni avanzate

JVM Setting

Job Run VM arguments

☒ Utilizza argomenti specifici JVM

Argument
-Xms256M
-Xmx3072M

New... Remove

Job TABELLE_DIM_LEGNAIA

Esecuzione base

Execution

Esegui Kill Pulisci

Usa 3 GB di RAM per aver migliori prestazioni durante l'esecuzione del job, inizializzata tramite il pulsante esegui nella sezione "Esecuzione base"

A fine esecuzione si vedrà il seguente schema con le relative statistiche per ogni passaggio.

RIASSUNTO

CAMPO	DESCRIZIONE
ARGOMENTO	ETL, METADATI, CARICAMENTO DA FILE A DB, DATA INGESTION
STRUMENTO	TALEND OPEN STUDIO, DATAGRIP
HOST	LOCALHOST
SERVER	192.168.2.14
PORTA	1433
JDBC PROVIDER	OPEN SOURCE JTDS
TIPO DB	MICROSOFT SQL SERVER
DATABASE	FASHION_RETAILER
SCHEMA	Fashion_Retailer
QUERY TOPICS	WHERE CLAUSE
COMPONENTI USATE	TMSSQLCONNECT, TMAP, TFILEINPUTEXCEL, TFILEINPUTDELIMITED, TMSSQLINPUT, TMSSQLOUTPUT, TMEORIZEROWS, TJavaFLEX, TLOGROWS

TABELLA	SCHEMA	DESCRIZIONE
DIM_CAUSALE	Fashion_Retailer	CAUSALE
DIM_PROVINCIA	Fashion_Retailer	PROVINCIA
DIM_MADE_IN	Fashion_Retailer	LUOGO DI FABBRICAZIONE
DIM_MADE_IN_PIANIFICAZIONE	Fashion_Retailer	SEDE DI FABBRICAZIONE
DIM_SHOP_ITALY	Fashion_Retailer	NEGOZIO
DIM_PRODUCT	Fashion_Retailer	PRODOTTO
FACT_SALES	Fashion_Retailer	VENDITE OTTIMIZZATE
FACT_COUNTER	Fashion_Retailer	CONTAPERSONE