



Interopérabilité des systèmes d'informations

TD Alimentation Multidimensionnelle

Josselin ABEL
Donatien SEGARD

Aurélien BOULET

Description des jobs, dans l'ordre:

SITE :

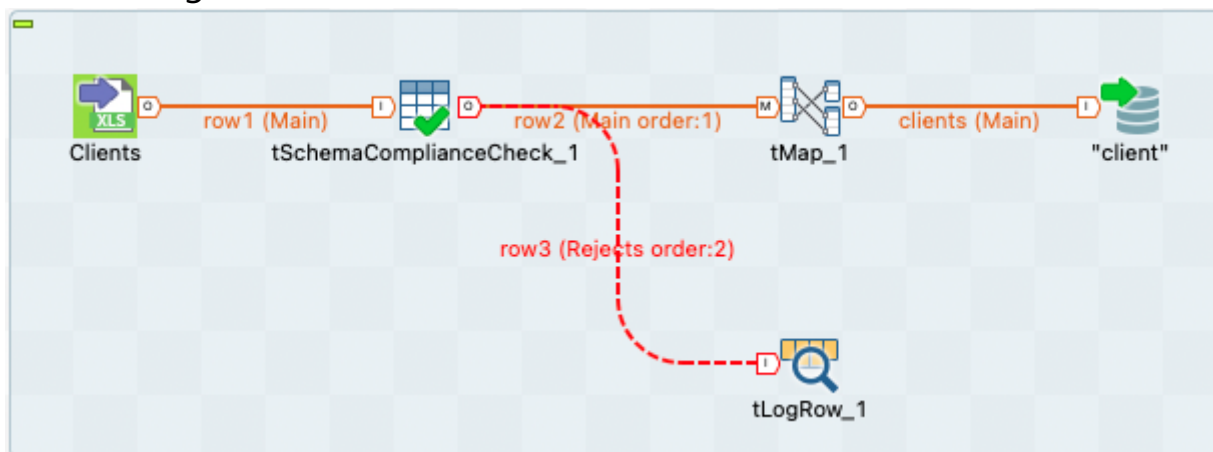
Les dossiers sont nommés en fonction des sites auxquels ils se rapportent. On parcourt donc les fichiers, afin de récupérer leur nom et donc ceux des sites (la méthode repose sur leur nom de dossier, c'est une méthode simple, mais qui ne nous assure pas une certaine adaptabilité):



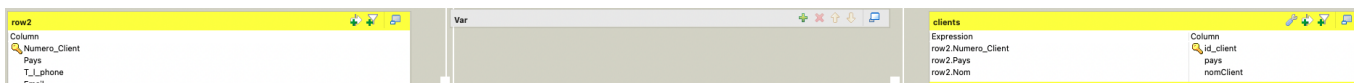
Ce job a pour vocation de mettre à jour la table des sites. On pourra le lancer de manière mensuelle, en pensant à vider la table avant de la recharger.

CLIENTS :

On parcourt les fichiers clients, on vérifie la validité des données (formats...), puis, on les charge dans la table client:



Grâce au mapping suivant, nous prenons seulement les informations dont nous avons besoin:



Ce job a pour but de mettre à jour la base client. Les clients pouvant évoluer plus souvent que les sites, ce job pourra être lancé de manière hebdomadaire.

CAMPAGNE :

On parcourt le fichier "2010" afin d'en extraire les informations concernant les campagnes, puis on les charge dans la table:

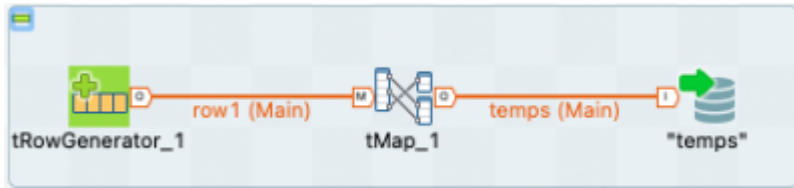
Encore une fois, grâce au mapping, nous ne prenons que les informations dont nous avons besoin:

La planification pourra être mensuelle. Les campagnes durant plusieurs mois, il n'y a pas besoin de faire tourner le job plus souvent.

On supprime les doublons avec un `tAggregateRow`:

La planification de ce job est la même que pour le job Campagne. On peut supposer qu'il n'y aura pas d'embauche ou de promotion toutes les semaines.

On utilise un `tGeneratorRow` pour générer les plages de temps dont nous avons besoins:

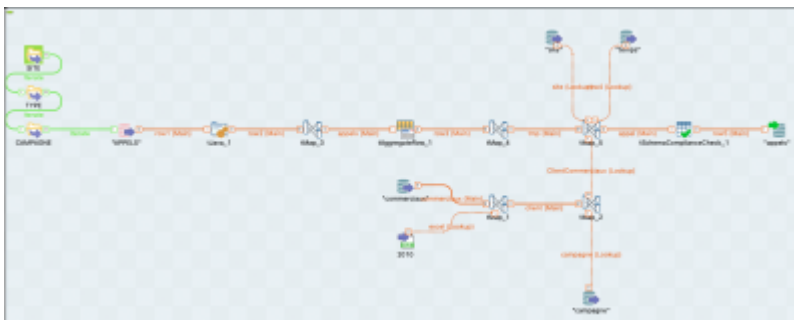


Pour les lignes générées, on utilise la fonction `AddDate` qui va nous permettre d'incrémenter d'une heure la date pour chaque ligne:



Ce job peut tourner tous les ans. En le modifiant, on peut le faire tourner une fois par ans pour ajouter les nouvelles tranches horaires pour la nouvelle année.

APPELS :



On parcourt les dossiers concernant les sites. Pour chacun de ces dossiers, on regarde les sous-dossiers qui eux concernent les campagnes. Et pour chacun de ces sous-dossiers, on parcourt les fichiers ".mdb", des fichiers access qui comportent les détails de chaque campagne:



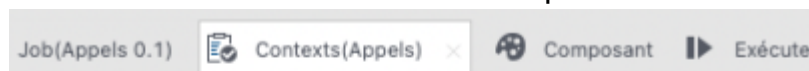
On change dynamiquement la base de données appelée grâce à un tFileList:

```
Base de données ((String)globalMap.get("tFileList_3_CURRENT_FILEPATH"))
```

Dans un tJava, on extraie les données concernant le fichier access actuel à partir des tFileList. On a donc la campagne et le site à un instant T:

```
String campagne = ((String)globalMap.get("tFileList_3_CURRENT_FILE"));
context.CAMPAGNE = campagne.substring(0,3);
context.SITE = ((String)globalMap.get("tFileList_1_CURRENT_FILE"));
System.out.println(context.CAMPAGNE + " || " + context.SITE);
```

Ici, les variables sont définies au préalable:



	Name	Type	Comment	Default	
				Value	
1	SITE	String			<input type="checkbox"/>
2	CAMPAGNE	String			<input type="checkbox"/>

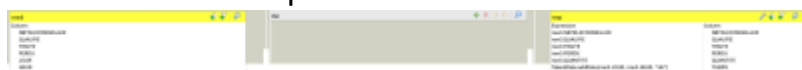
On utilise un premier tMap qui nous permet d'écarter les données superflues, et de formater la colonne heure sur un entier:



Le formatage de l'heure nous permet, par la suite, d'utiliser un tAggregateRow et d'effectuer un GroupBy sur le jour et l'heure, qui nous permet aussi, de faire un "count" sur les téléconseillers et faire une somme sur les trois autres colonnes (à préciser):

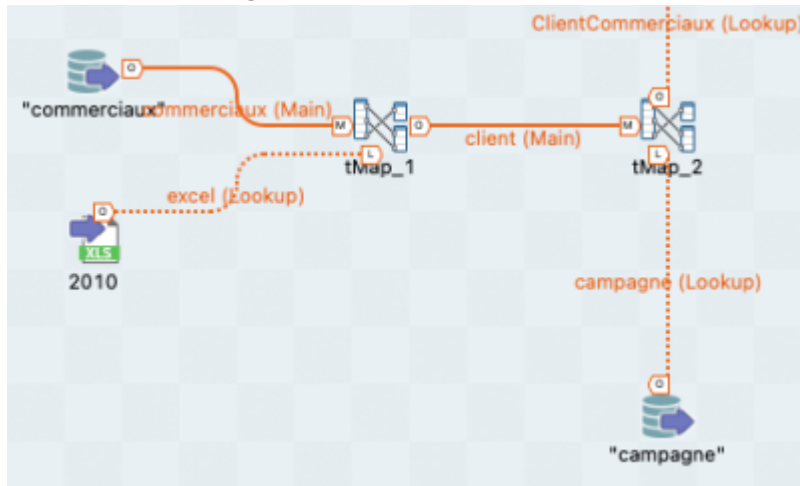
Group by	Colonne de sortie		Position de la colonne d'entrée	
	JOUR		JOUR	
	HEURE		HEURE	
Opérations				
	Colonne de sortie		Position de la colonne d'entrée	
		Fonction		
	QUALITE	somme	QUALITE	
	TRAITE	somme	TRAITE	
	PERDU	somme	PERDU	
	TELECONSEILLER	count	TELECONSEILLER	
	QUANTITE	somme	QUANTITE	

Le tMap suivant, lui, permet de fusionner les colonnes JOUR et HEURE en une colonne TEMPS qui rassemble les informations des deux colonnes pour faire le lien avec la table temps:

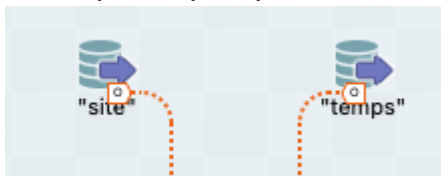


Pour avoir nos dimensions COMMERCIAUX, CLIENT et CAMPAGNE, on effectue une jointure entre la table COMMERCIAUX, CAMPAGNE et le fichier 2010 nous permettant d'obtenir l'idCampagne, l'idCommerciaux et l'idClient, on garde aussi

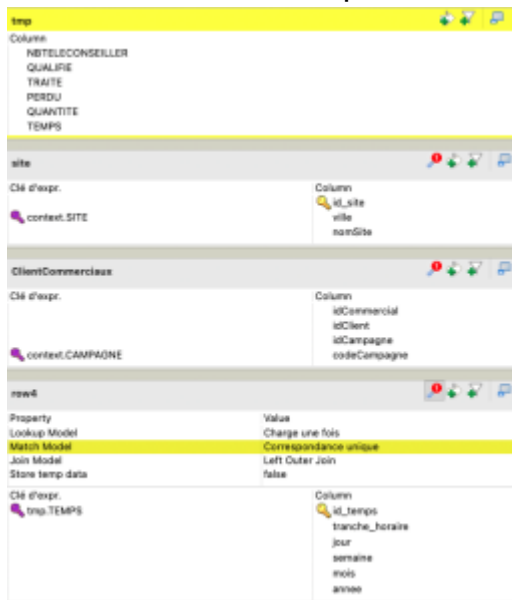
le codeCampagne pour plus tard:



Nous faisons aussi appel à la table SITE et à la table TEMPS afin d'avoir tous les champs requis pour la table cible:



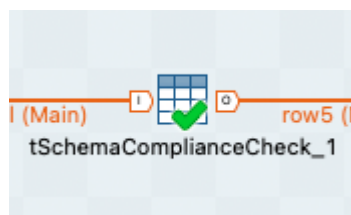
Dans le tMap suivant, les variables context sont utilisés pour filtrer le site et le flux contenant les champs suivant: idCommercial, idClient et idCampagne (on peut filtrer ce dernier grâce au codeCampagne géré précédemment). Pour s'assurer de la validité de l'idTemps, on fait une jointure avec la colonne temps:



Pour faire le mapping, on utilise un tMap:

appel	
Expression	Column
ClientCommerciaux.idCommercial	commerciaux_idcommerciaux
site.id_site	site_id_site
row4.id_temps	temps_id_temps
ClientCommerciaux.idCampagne	campagne_id_campagne
ClientCommerciaux.idClient	client_id_client
tmp.QUANTITE	quantite
tmp.QUALIFIE	qualife
tmp.TRAITE	traite
tmp.PERDU	perdu
tmp.NBTELECONSEILLER	nb_teleconseiller

Ensuite, on utilise un tSchemaCompliance pour vérifier s'il y a une différence entre notre flux d'informations et la table cible:



Enfin, on envoi les données dans la table:

Capture d'écran de la table appels:

commerciaux_idcommerciaux	site_id_site	temps_id_temps	campagne_id_campagne	client_id_client	quantite	qualife	traite	perdu	nb_teleconseiller
5	2	2009-11-16 10:00:00	10	3	273	0	1	1	1
5	2	2009-11-16 20:00:00	10	3	184	0	0	1	1
5	2	2009-11-19 16:00:00	10	3	484	1	0	0	1
5	2	2009-11-20 16:00:00	10	3	485	0	0	0	1
5	2	2009-12-02 03:00:00	10	3	542	1	1	1	1
5	2	2009-12-02 10:00:00	10	3	277	0	1	0	1
5	2	2009-12-03 01:00:00	10	3	458	1	0	1	1
5	2	2009-12-03 13:00:00	10	3	238	1	1	0	1
5	2	2009-12-05 22:00:00	10	3	334	0	0	1	1
5	2	2009-12-09 00:00:00	10	3	282	0	1	1	1
5	2	2009-12-16 15:00:00	10	3	370	0	1	1	1
5	2	2009-12-18 15:00:00	10	3	343	0	1	0	1

Le job Appels permet de charger les appels dans la base. Etant donné que ce ne sont pas les données précises des appels mais une vision globale des données agrégées par tranche horaire. On peut donc supposer que c'est avant tout du reporting et non du suivi en temps réel. On peut donc faire tourner le job de manière hebdomadaire, en ignorant les fichiers déjà chargés.

Ordre :

Les jobs chargeant les tables dimension n'ont pas de précédence entre-eux, ils peuvent donc être chargés dans l'ordre qu'on veut. Etant donné que nous ne leur avons pas donné la même périodicité de chargement, ça tombe bien. Pour le job Appel, il faudra attendre que chaque job hebdomadaire soit chargé avant de le lancer pour éviter des problèmes de FK. Enfin, il faudra faire tourner les jobs mensuels en début de mois dans la même optique.