



Projet DSID - BI

MIAGE CFA

Groupe 1:

- ANDRIAMAHADIMBY Henintsoa
- DAHOUMANE Ahcene
- KHOUBABA M'Hamed
- KUNDA Thierry

Introduction	3
Tâche préparatoire	3
Modélisation	3
SI Opérationnel	3
SI Décisionnel	3
DM1:	4
DM2:	5
Alimentation du SI OPE	5
Alimentation Datamart	8
Alimentation DM1:	8
Alimentation DM2:	10
Power BI	13
DM1:	13
STAT:	15
DM2:	16
Difficultés rencontrées	17
Conclusion	19
Matrice de répartition:	20

Introduction

Pour le projet DSID nous avons constitué un groupe de 4 personnes.

Le projet porte sur l'entreprise C&L qui est spécialisée dans la mise en relation entre les clients et restaurateurs, afin de permettre aux clients via l'application C&L de commander des menus, C&L assure la livraison via un réseau de livreurs indépendants.

Les objectifs sont de permettre aux dirigeants de suivre:

- Le montant total de la commande, le temps théorique de préparation en minutes ainsi que le temps réel de préparation en minutes.
- Le temps de livraison en minutes, le nombre d'articles livrés ainsi que la rémunération du livreur.

Tâche préparatoire

Tout d'abord nous avons décidé d'utiliser [GIT](#) pour la collaboration. Cela nous facilitera pour le travail en groupe ainsi que la sauvegarde et le versionning.

Pour avoir un retour rapide des étapes de réalisation nous sommes parti du principe qu'il était plus intéressant d'avant tout de préparer le SI OPE et d'enchaîner directement l'alimentation du DM1. Une fois toutes les opérations réalisées et vérifiées, nous avons répété les mêmes étapes pour le DM2.

Nous avons commencé par créer des users dans les différentes bases: Oracle, PostgreSQL, BigQuery ainsi que pour l'API openrouteservice. Ensuite, nous avons créé les fichiers source ainsi que le script d'insertion.

Modélisation

Pour l'étape de modélisation nous avons modélisé des tables du SI Opérationnel et du SI Décisionnel.

SI Opérationnel

Pour le SI Opérationnel nous avons suivi le MPD donné.

Pour le WRK nous avons préparé au préalable une vue pour chaque datamart pour alimenter la table de STAGING. On a en plus défini le modèle des tables ODS ayant comme les mêmes colonnes que celle du STAGING mais en y ajoutant les métriques nécessaires pour la suivi des dirigeants. Enfin la table REJET aura les mêmes colonnes que le STAGING mais avec la colonne message et le type d'erreur en plus.

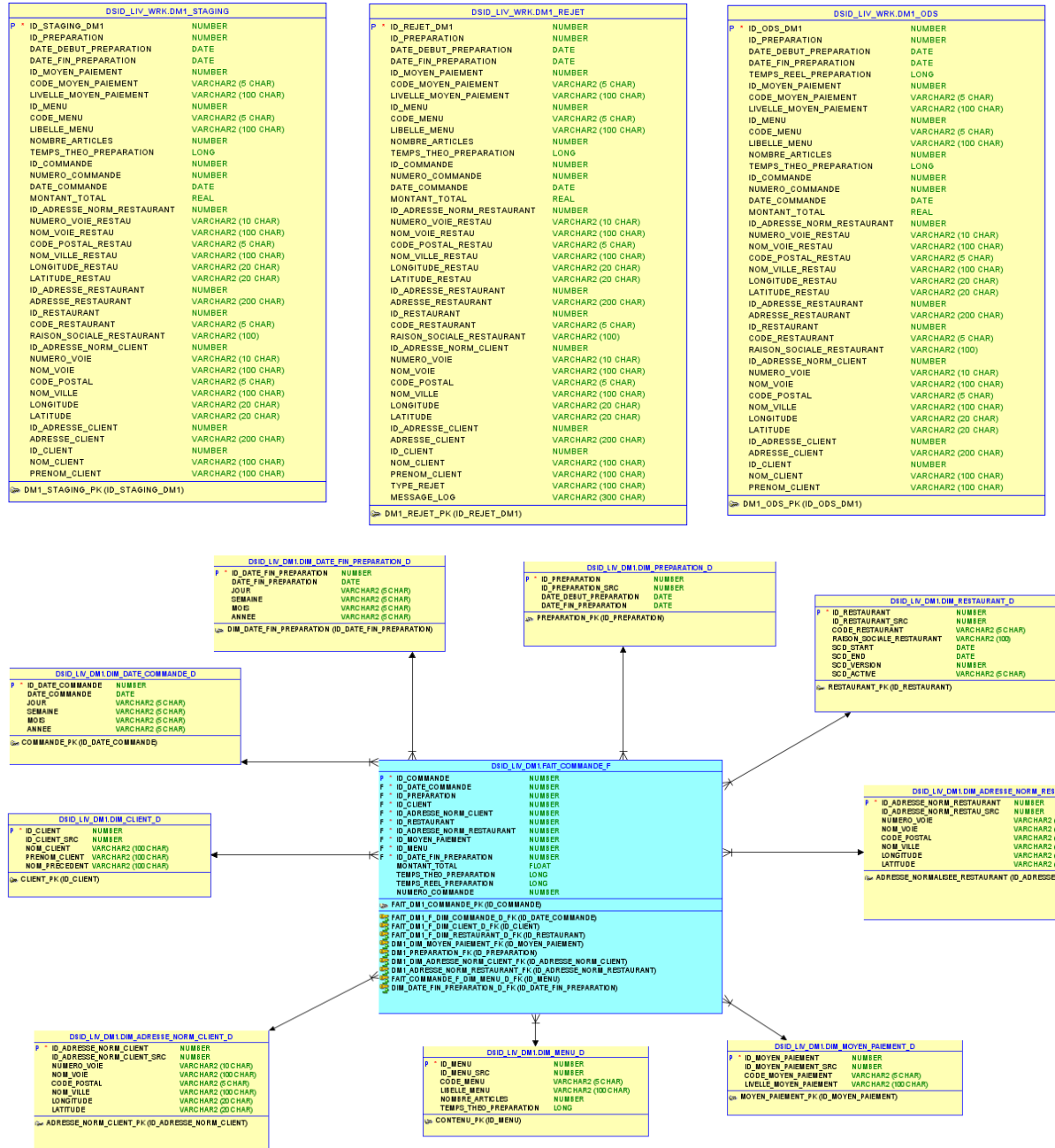
SI Décisionnel

Pour le SI Décisionnel:

Nous avons représenté les données dans un schéma en étoile. Les données du SI Opérationnel avec comme tables de fait les informations de commande pour le DM1 et les

informations de livraison pour le DM2. Nous avons éventuellement mis en table de dimension les données spécifiées dans les consignes, en plus des dates en table de dimension avec le jour, la semaine, mois et l'année.

DM1:

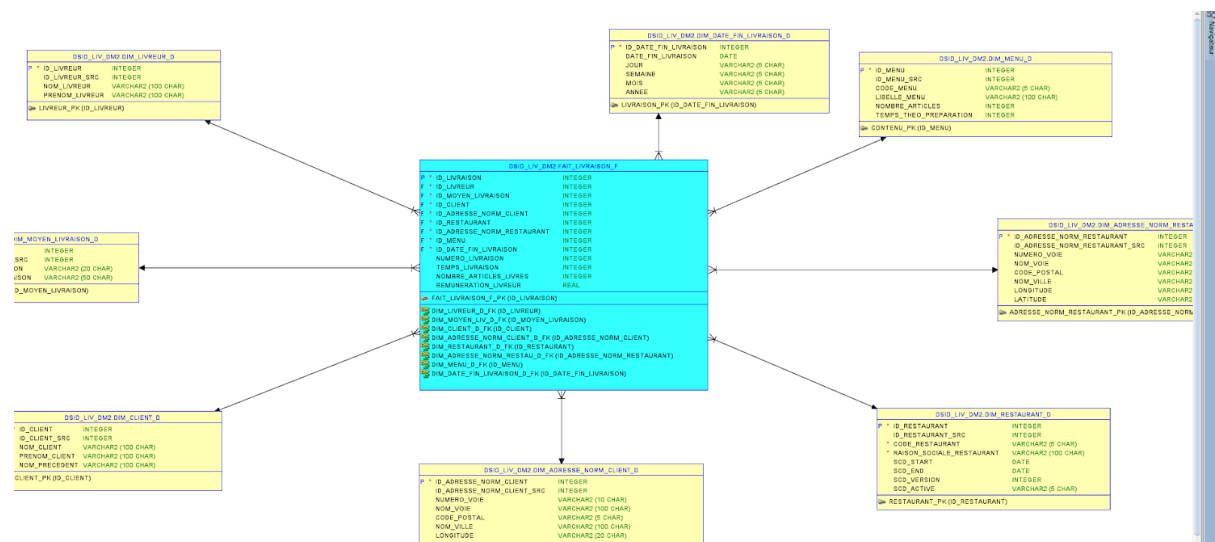


DM2:

DSID_LIV_WRK.DM2_STAGING	
P * ID_STAGING_DM2	INTEGER
ID_ADRESSE_NORM_CLIENT	INTEGER
NUMERO_VOIE	VARCHAR2 (10 CHAR)
NOM_VOIE	VARCHAR2 (100 CHAR)
CODE_POSTAL	VARCHAR2 (5 CHAR)
NOM_VILLE	VARCHAR2 (100 CHAR)
LONGITUDE	VARCHAR2 (20 CHAR)
LATITUDE	VARCHAR2 (20 CHAR)
ID_RESTAURANT	INTEGER
CODE_RESTAURANT	VARCHAR2 (5 CHAR)
RAISON_SOCIALE_RESTAURANT	VARCHAR2 (100 CHAR)
ID_ADRESSE_NORM_RESTAURANT	INTEGER
NUMERO_VOIE_RESTAU	VARCHAR2 (10 CHAR)
NOM_VOIE_RESTAU	VARCHAR2 (100 CHAR)
CODE_POSTAL_RESTAU	VARCHAR2 (5 CHAR)
NOM_VILLE_RESTAU	VARCHAR2 (100 CHAR)
LONGITUDE_RESTAU	VARCHAR2 (20 CHAR)
LATITUDE_RESTAU	VARCHAR2 (20 CHAR)
ID_CLIENT	INTEGER
ID_LIVRAISON	INTEGER
ID_COMMANDE	INTEGER
NUMERO_COMMANDE	INTEGER
DATE_COMMANDE	DATE
MONTANT_TOTAL	REAL
ID_MENU	INTEGER
CODE_MENU	VARCHAR2 (5 CHAR)
LIBELLE_MENU	VARCHAR2 (100 CHAR)
NOMBRE_ARTICLES	INTEGER
NUMBRE_ARTICLE_LIVRES	INTEGER
TEMPS_THEO_PREPARATION	INTEGER
NUMERO_LIVRAISON	INTEGER
DATE_DEBUT_LIVRAISON	DATE
DATE_FIN_LIVRAISON	DATE
ID_LIVREUR	INTEGER
NOM_LIVREUR	VARCHAR2 (100 CHAR)
PRENOM_LIVREUR	VARCHAR2 (100 CHAR)
ID_MOYEN_LIVRAISON	INTEGER
CODE_MOYEN_LIVRAISON	VARCHAR2 (20 CHAR)
LIBELLE_MOYEN_LIVRAISON	VARCHAR2 (20 CHAR)
NOM_CLIENT	VARCHAR2 (100 CHAR)
PRENOM_CLIENT	VARCHAR2 (100 CHAR)
ID_PREPARATION	INTEGER
DATE_DEBUT_PREPARATION	DATE
DATE_FIN_PREPARATION	DATE
DM2_STAGING_PK (ID_STAGING_DM2)	

DSID_LIV_WRK.DM2_REJET	
P * ID_REJET_DM2	INTEGER
ID_ADRESSE_NORM_CLIENT	INTEGER
NUMERO_VOIE	VARCHAR2 (10 CHAR)
NOM_VOIE	VARCHAR2 (100 CHAR)
CODE_POSTAL	VARCHAR2 (5 CHAR)
NOM_VILLE	VARCHAR2 (100 CHAR)
LONGITUDE	VARCHAR2 (20 CHAR)
LATITUDE	VARCHAR2 (20 CHAR)
ID_RESTAURANT	INTEGER
CODE_RESTAURANT	VARCHAR2 (5 CHAR)
RAISON_SOCIALE_RESTAURANT	VARCHAR2 (100 CHAR)
ID_ADRESSE_NORM_RESTAURANT	INTEGER
NUMERO_VOIE_RESTAU	VARCHAR2 (10 CHAR)
NOM_VOIE_RESTAU	VARCHAR2 (100 CHAR)
CODE_POSTAL_RESTAU	VARCHAR2 (5 CHAR)
NOM_VILLE_RESTAU	VARCHAR2 (100 CHAR)
LONGITUDE_RESTAU	VARCHAR2 (20 CHAR)
LATITUDE_RESTAU	VARCHAR2 (20 CHAR)
ID_CLIENT	INTEGER
ID_LIVRAISON	INTEGER
ID_COMMANDE	INTEGER
NUMERO_COMMANDE	INTEGER
DATE_COMMANDE	DATE
MONTANT_TOTAL	REAL
ID_MENU	INTEGER
CODE_MENU	VARCHAR2 (5 CHAR)
LIBELLE_MENU	VARCHAR2 (100 CHAR)
NOMBRE_ARTICLES	INTEGER
TEMPS_THEO_PREPARATION	INTEGER
NUMERO_LIVRAISON	INTEGER
DATE_DEBUT_LIVRAISON	DATE
DATE_FIN_LIVRAISON	DATE
ID_LIVREUR	INTEGER
NOM_LIVREUR	VARCHAR2 (100 CHAR)
PRENOM_LIVREUR	VARCHAR2 (100 CHAR)
CODE_MOYEN_LIVRAISON	VARCHAR2 (20 CHAR)
LIBELLE_MOYEN_LIVRAISON	VARCHAR2 (20 CHAR)
NOM_CLIENT	VARCHAR2 (100 CHAR)
PRENOM_CLIENT	VARCHAR2 (100 CHAR)
ID_PREPARATION	INTEGER
DATE_DEBUT_PREPARATION	DATE
DATE_FIN_PREPARATION	DATE
TYPE_REJET	VARCHAR2 (100 CHAR)
MESSAGE_LOG	VARCHAR2 (300 CHAR)
NOMBRE_ARTICLE_LIVRES	INTEGER
DM2_REJET_PK (ID_REJET_DM2)	

DSID_LIV_WRK.DM2_ODS	
P * ID_ODS_DM2	INTEGER
ID_ADRESSE_NORM_CLIENT	INTEGER
NUMERO_VOIE	VARCHAR2 (10 CHAR)
NOM_VOIE	VARCHAR2 (100 CHAR)
CODE_POSTAL	VARCHAR2 (5 CHAR)
NOM_VILLE	VARCHAR2 (100 CHAR)
LONGITUDE	VARCHAR2 (20 CHAR)
LATITUDE	VARCHAR2 (20 CHAR)
ID_RESTAURANT	INTEGER
CODE_RESTAURANT	VARCHAR2 (5 CHAR)
RAISON_SOCIALE_RESTAURANT	VARCHAR2 (100 CHAR)
ID_ADRESSE_NORM_RESTAURANT	INTEGER
NUMERO_VOIE_RESTAU	VARCHAR2 (10 CHAR)
NOM_VOIE_RESTAU	VARCHAR2 (100 CHAR)
CODE_POSTAL_RESTAU	VARCHAR2 (5 CHAR)
NOM_VILLE_RESTAU	VARCHAR2 (100 CHAR)
LONGITUDE_RESTAU	VARCHAR2 (20 CHAR)
LATITUDE_RESTAU	VARCHAR2 (20 CHAR)
ID_CLIENT	INTEGER
ID_LIVRAISON	INTEGER
ID_COMMANDE	INTEGER
NUMERO_COMMANDE	INTEGER
DATE_COMMANDE	DATE
MONTANT_TOTAL	REAL
ID_MENU	INTEGER
CODE_MENU	VARCHAR2 (5 CHAR)
LIBELLE_MENU	VARCHAR2 (100 CHAR)
NOMBRE_ARTICLES	INTEGER
TEMPS_THEO_PREPARATION	INTEGER
NUMERO_LIVRAISON	INTEGER
DATE_DEBUT_LIVRAISON	DATE
DATE_FIN_LIVRAISON	DATE
ID_LIVREUR	INTEGER
NOM_LIVREUR	VARCHAR2 (100 CHAR)
PRENOM_LIVREUR	VARCHAR2 (100 CHAR)
ID_MOYEN_LIVRAISON	INTEGER
CODE_MOYEN_LIVRAISON	VARCHAR2 (20 CHAR)
LIBELLE_MOYEN_LIVRAISON	VARCHAR2 (20 CHAR)
NOM_CLIENT	VARCHAR2 (100 CHAR)
PRENOM_CLIENT	VARCHAR2 (100 CHAR)
ID_PREPARATION	INTEGER
DATE_DEBUT_PREPARATION	DATE
DATE_FIN_PREPARATION	DATE
TEMPS_LIVRAISON	INTEGER
NOMBRE_ARTICLE_LIVRES	INTEGER
REMUNERATION_LIVREUR	REAL
DM2_ODS_PK (ID_ODS_DM2)	

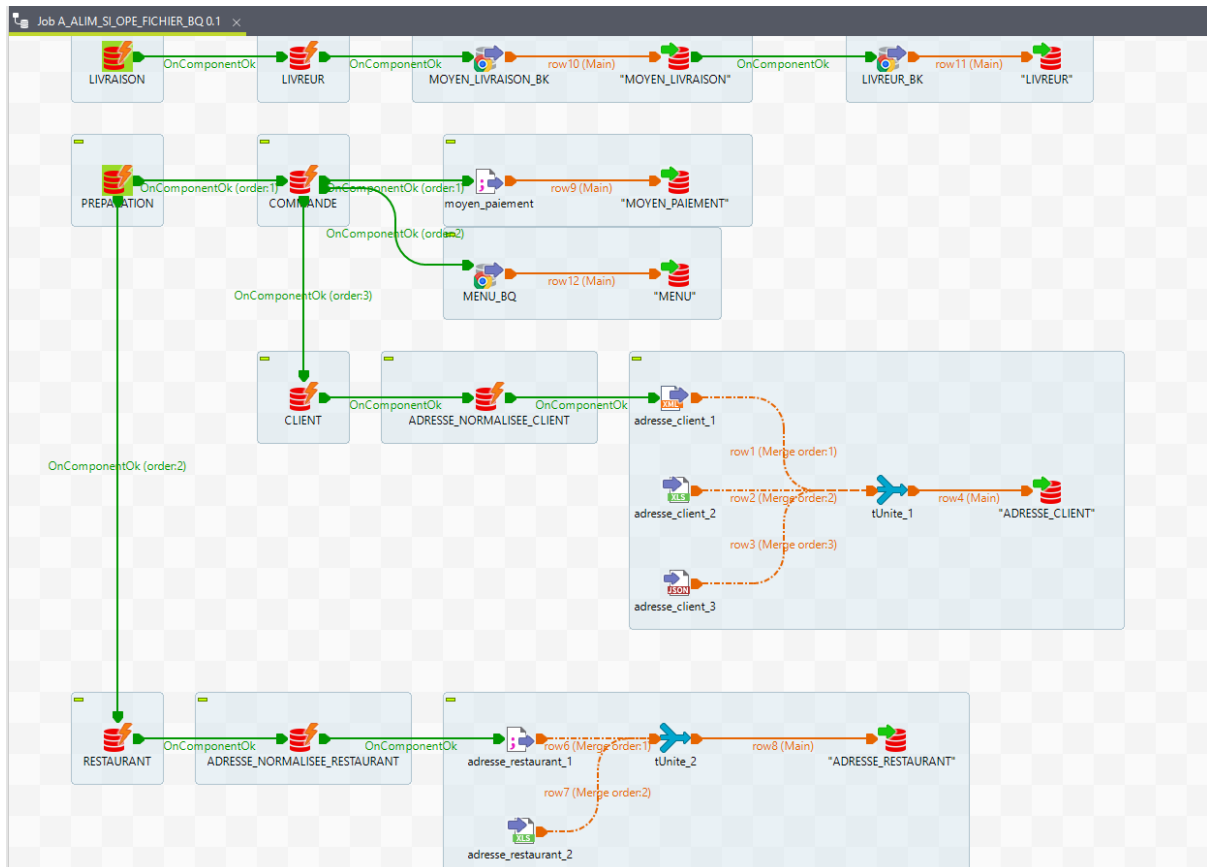


Pour la gestion des SCD, nous avons g r  la table restaurant avec le SCD de type 2, et la colonne client.nom avec le SCD de type 3. Les colonnes restantes ont  t  en SCD de type 0 ou 1 selon nos choix arbitraire (exemple: client.prenom type 1 et restaurant.code_restaurant type 0)

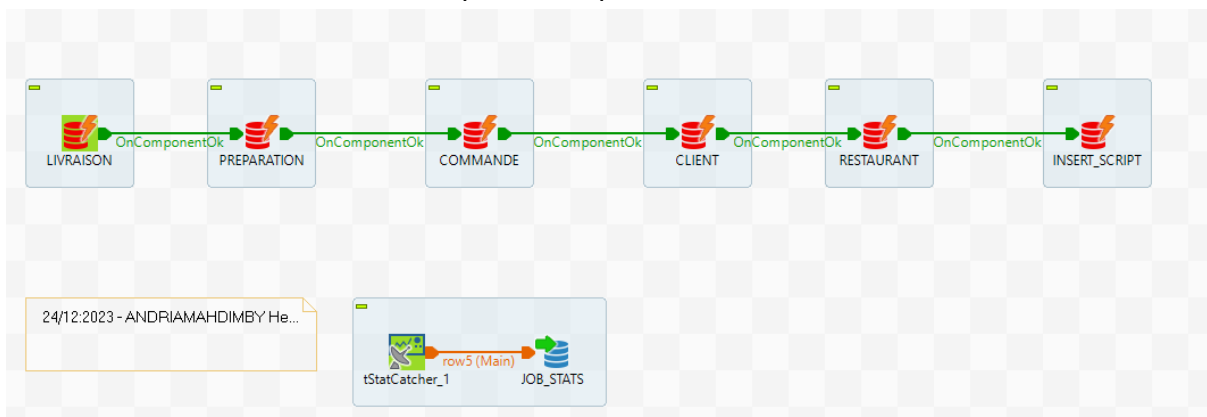
Alimentation du SI OPE

Pour l'alimentation des tables du SI Op rationnel nous avons g n r  un jeu de donn es enregistr  dans des fichiers .xml , .xls , .json , .csv .

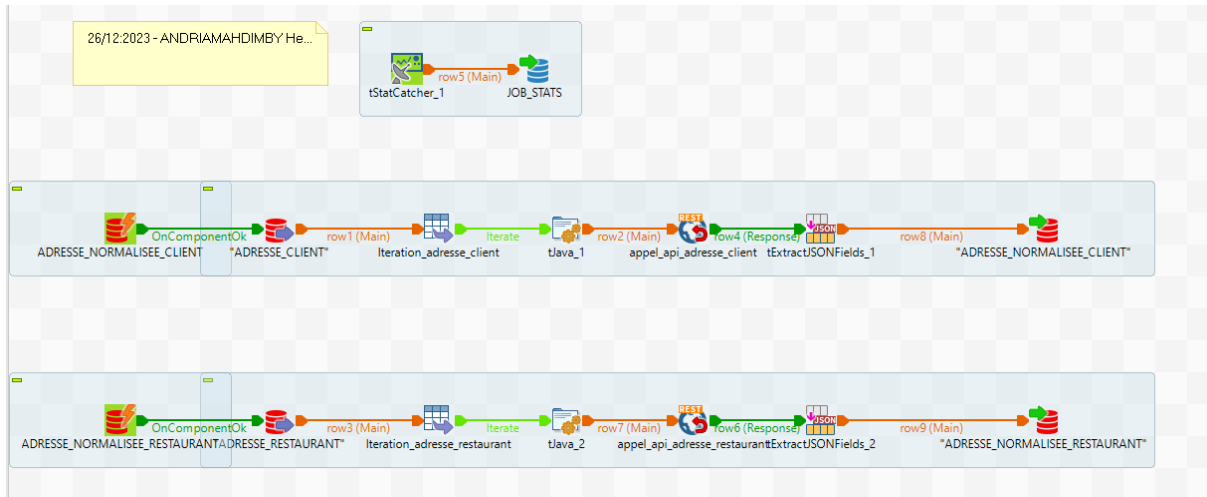
- En premier lieu nous avons alimenté les tables Moyen paiement, Adresse client, Adresse restaurant avec les fichiers sources.
- En deuxième lieu nous avons alimenté les tables Livreur, Moyen livraison et Menu via Bigquery.



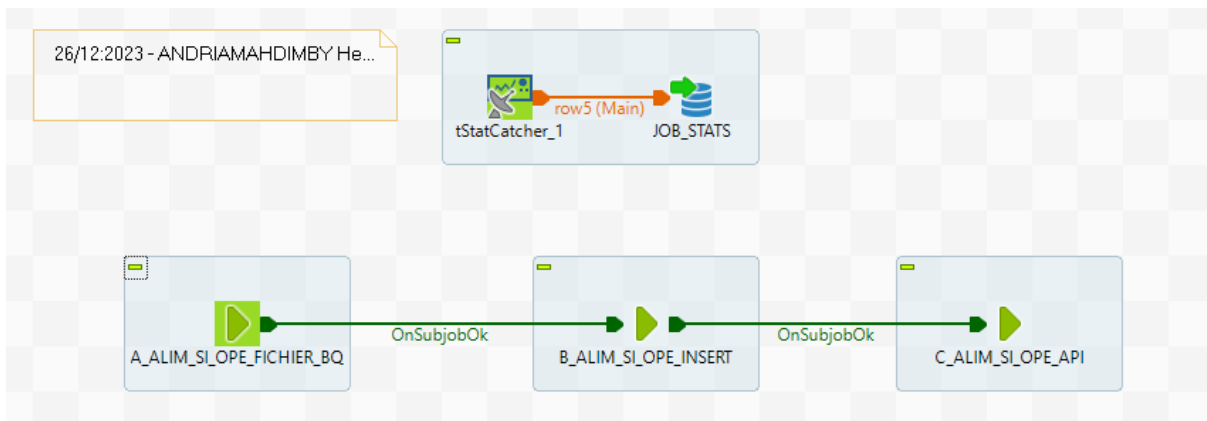
- Ensuite, nous avons alimenté les tables livraison, préparation, commande, client, restaurant en mettant le script insert sql dans un TDBRow.



- Nous avons, par la suite, alimenté les tables adresses normalisées client et adresse normalisée restaurant.



- Après avoir créé tous les jobs d'alimentation on a mis en place l'alimentation globale composée des toutes les alimentations.



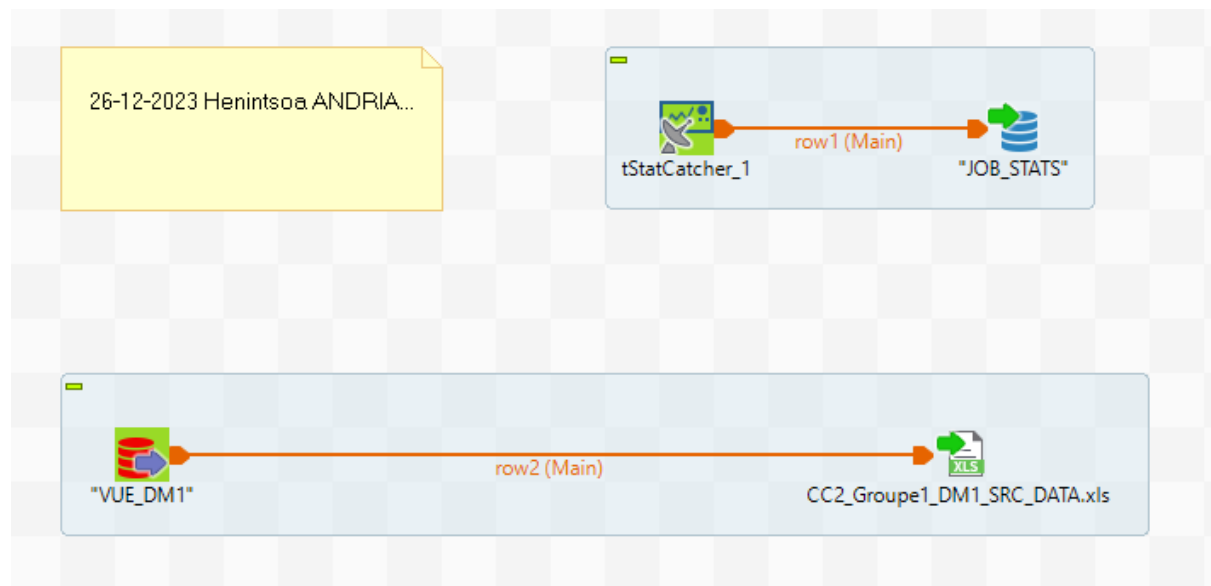
Remarque pour les alimentations du SI OPE:

Nous avons fait en sorte d'alimenter des tables référencées par les clés étrangères d'abord puis les tables qui les référencent. Pour vider les tables nous avons fait le sens inverse. Cela nous a permis de gérer les violations de clés étrangères.

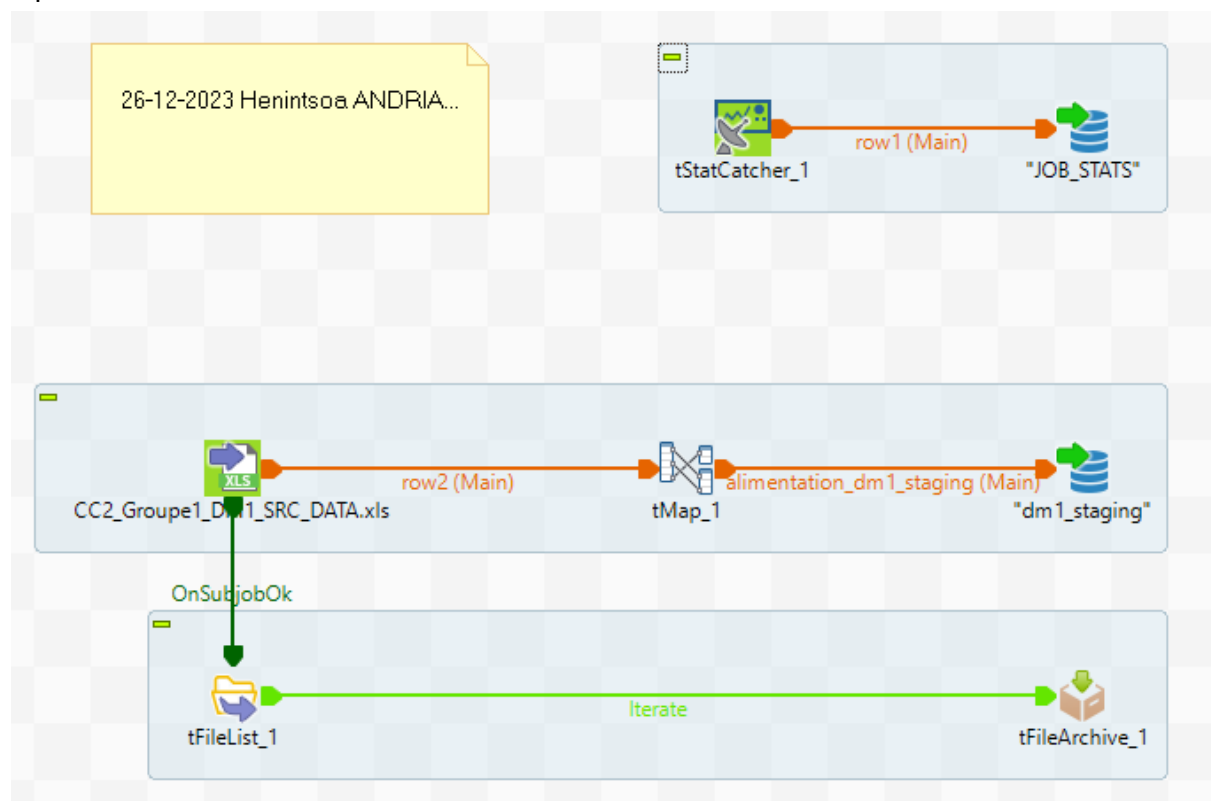
Alimentation Datamart

Alimentation DM1:

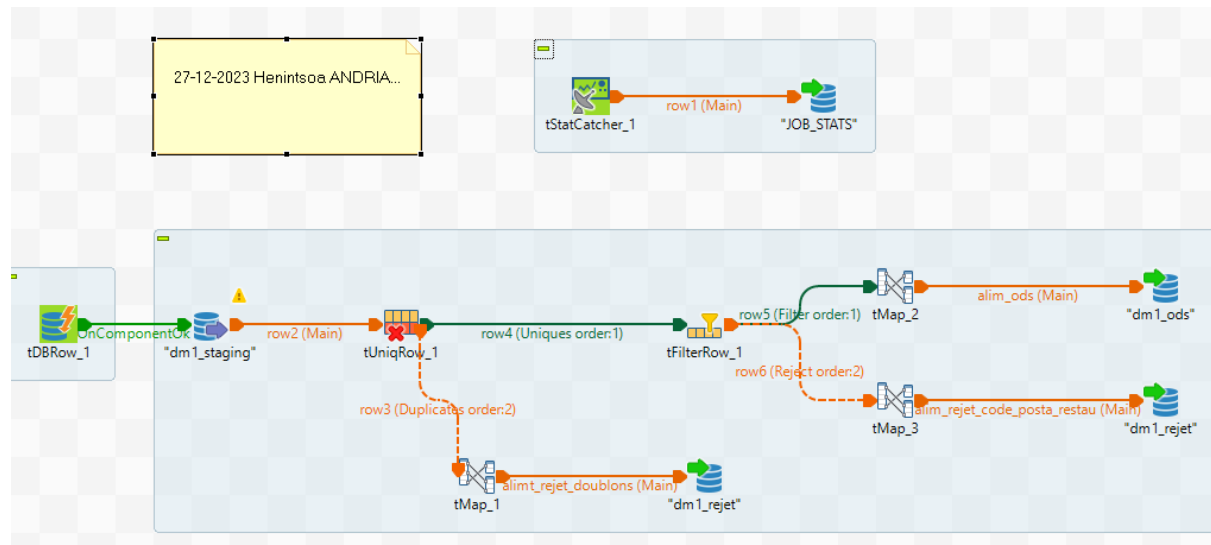
Tout d'abord nous avons généré des fichiers d'archive Excel des données de la VUE du DM1



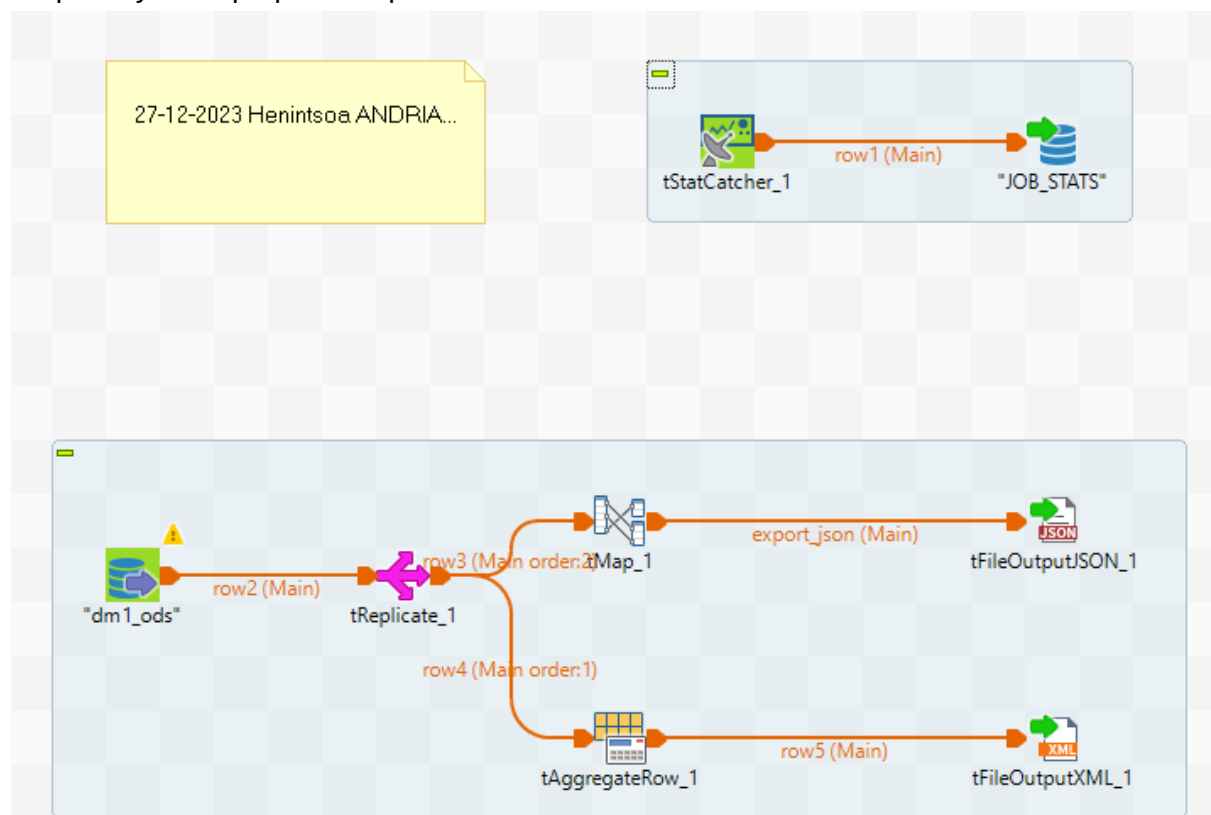
A partir de ces fichiers d'archives nous avons alimenté le STAGING.



Pour l'ODS Rejet nous avons rejeté les doublons, les adresses normalisées (client et restaurant) qui ont comme numéro de voie , nom voie et client NULL.



Nous exportons en fichiers plats les données de l'ODS et l'agrégation du montant total et du temps moyen de préparation par client et date de commande.

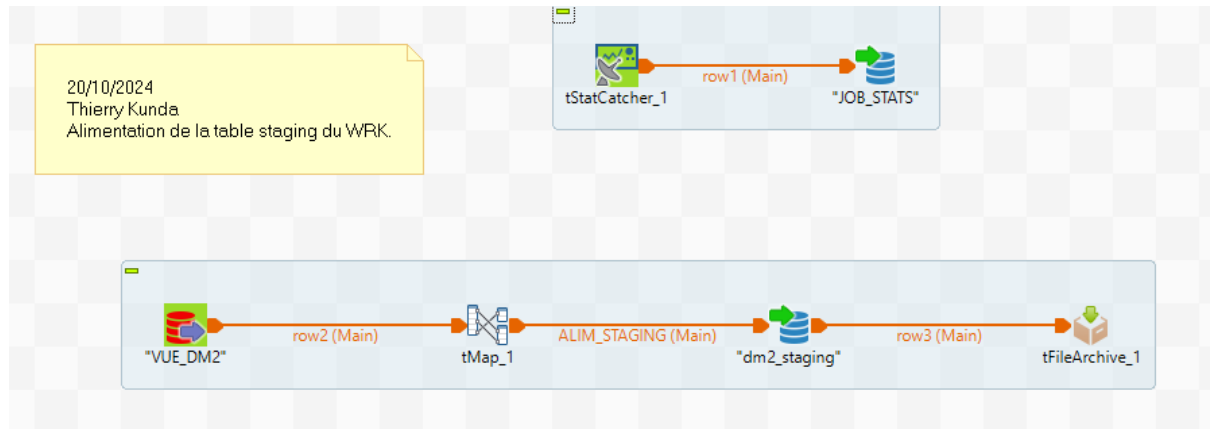


Remarques:

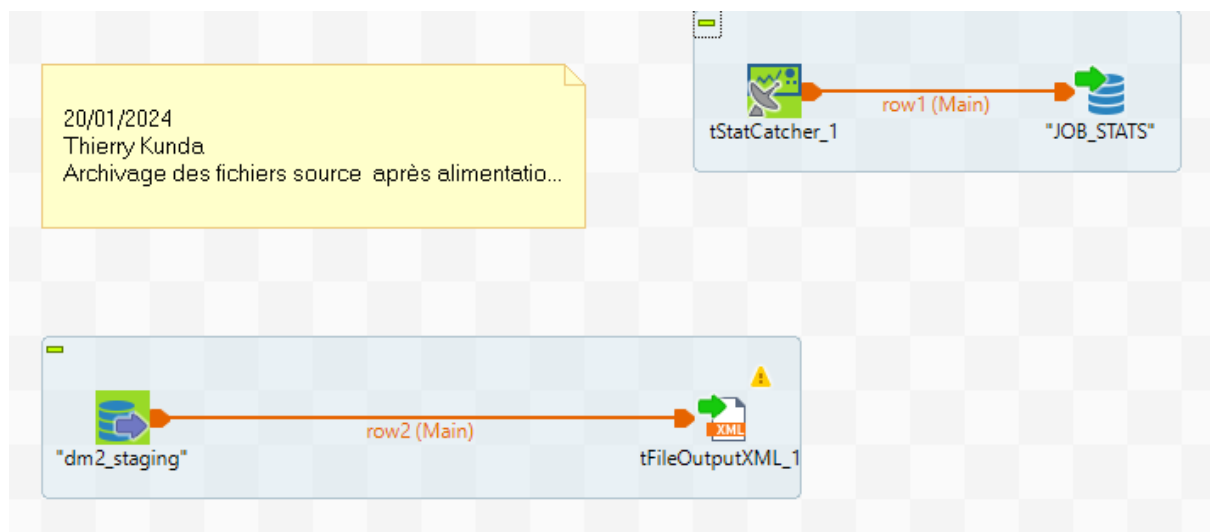
- . On utilise un composant tReplicate pour avoir 2 sorties de données à partir du DM1 ODS.
- . On a utilisé tous les types de SCD pour les tables de dimensions.

Alimentation DM2:

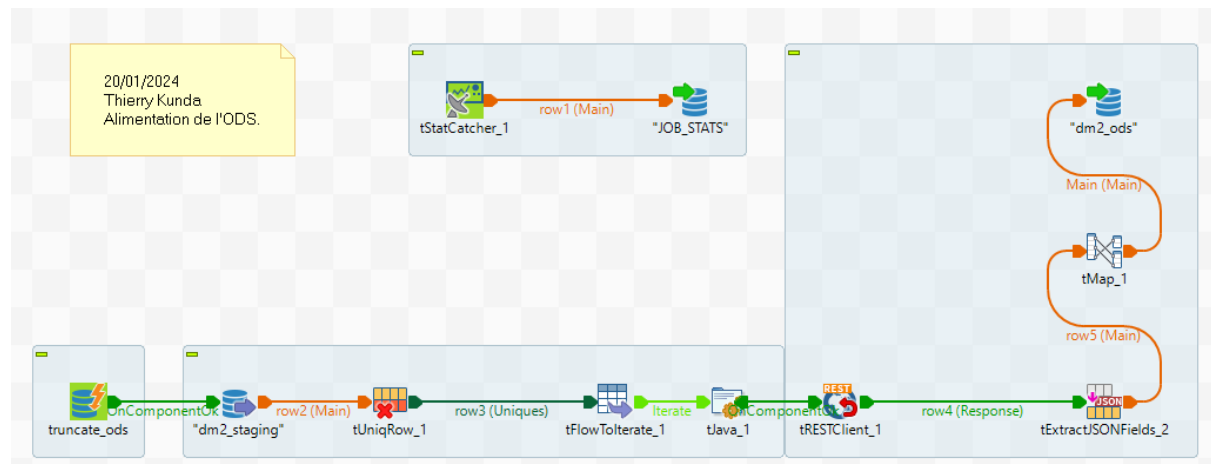
Tout d'abord nous alimentons la table staging WRK à partir de la vue DM2 du SI Opérationnel et que l'on archive dans un fichier compressé.



Exportation de l'archive.

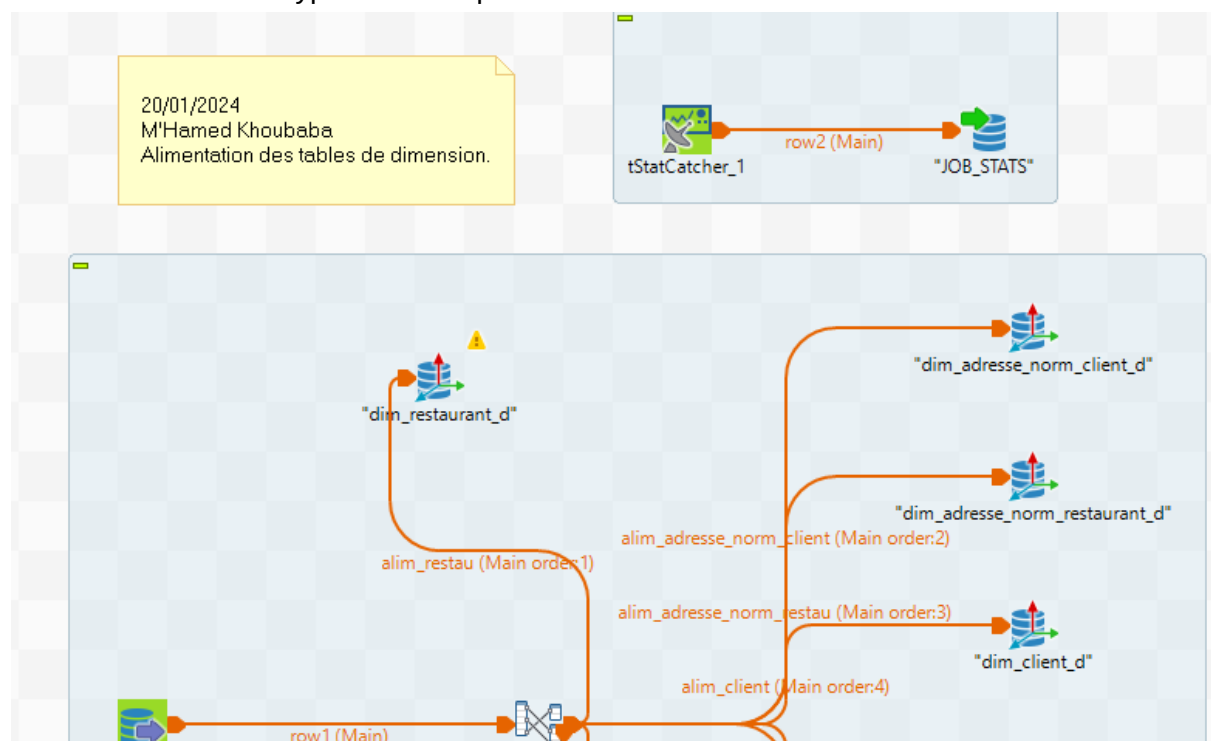


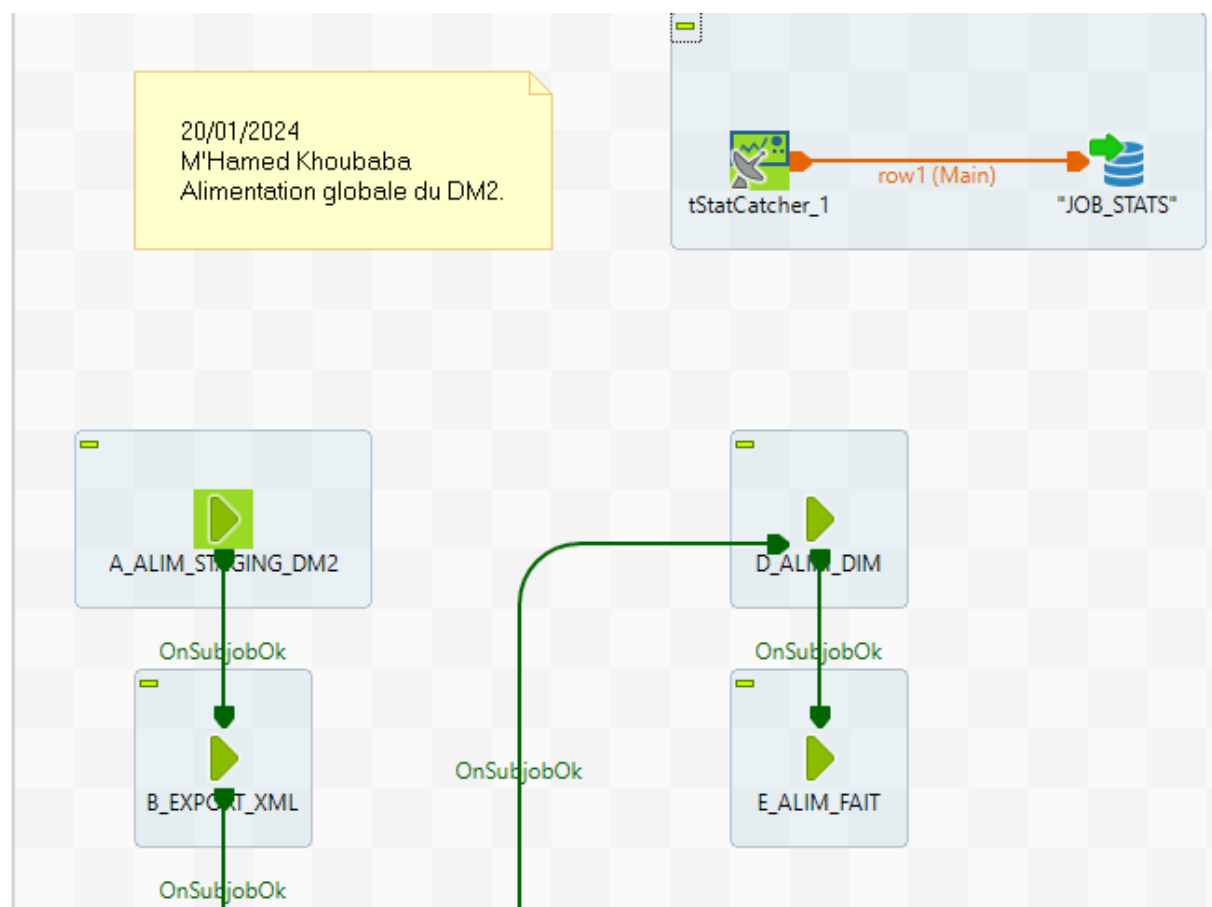
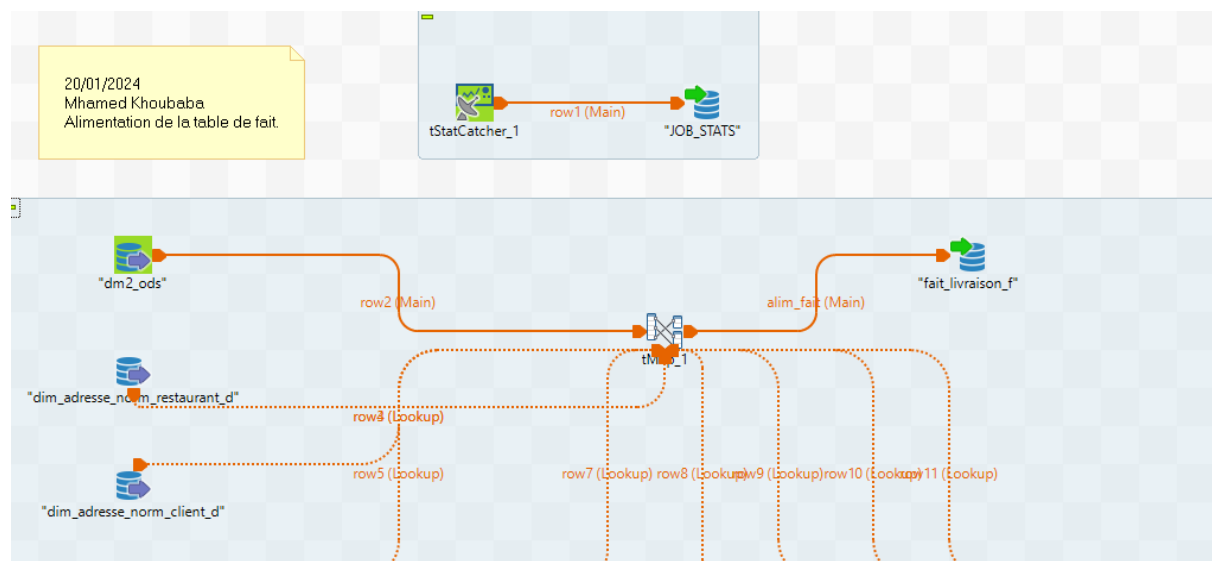
Pour l'alimentation de l'ODS nous la vidons au préalable. Il suffit ensuite de reprendre les données du STAGING, en plus de la distance et du temps de livraison (pour calculer la rémunération du livreur) calculée à partir des appels API.



Remarques:

. On a utilisé tous les types de SCD pour les tables de dimensions.





L'alimentation des tables de dimensions, de la table de fait ainsi que l'alimentation globale est la même que celle du DM1.

Power BI

DM1:

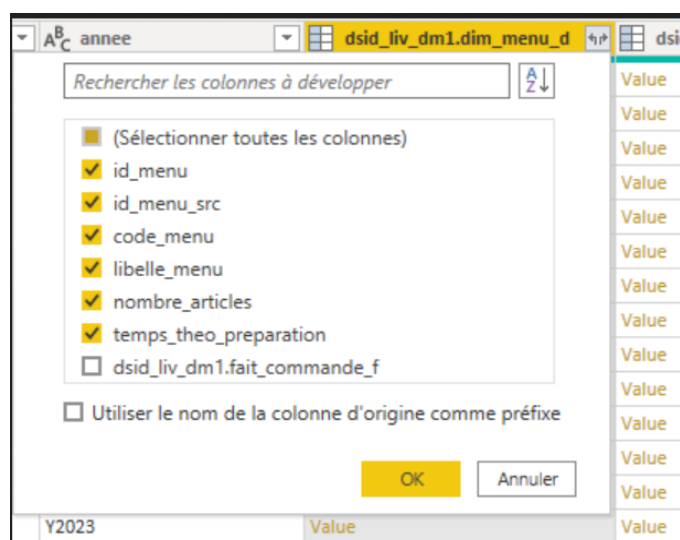
Introduction:

Ce rapport, présenté sous forme de fichier .pbix, propose une série de visualisations de données et de graphiques. Il se concentre sur l'analyse comparative de plusieurs variables, notamment en lien avec les commandes et leur temps de préparation, selon divers critères. Cette approche permet d'identifier les tendances, les corrélations et d'optimiser les processus de commande.

Traitement PowerQuery:

La stratégie choisie consiste à travailler uniquement sur la table de fait pour la construction des rapports en réponse à la surabondance de colonnes présentes dans les tables de fait et de dimension .

Il convient alors de développer dans un premier temps cette table pour ajouter les différentes colonnes des tables de dimensions associées.



Ensuite dans la colonne moyen de paiements on doit remplacer les variables “Chèque” et “Espèces” pour avoir ce résultat :

livelle_moyen_paiement
Carte bleue
Chèque
Carte bleue
Chèque
Tickets restaurant
Espèces
Carte bleue
Chèque
Tickets restaurant
Espèces
Espèces
Carte bleue
Espèces
Carte bleue
Chèque
Carte bleue
Chèque
Tickets restaurant
Espèces
Carte bleue
Tickets restaurant
Espèces
Tickets restaurant

On obtient a la fin la table de fait qui contient 23 lignes et ces colonnes ainsi que l'ensemble des colonnes 72 qui provient des tables de dimension.

Table.ExpandRecordColumn(#"dsid_liv_dm1.dim_adresse_norm_client_d développé", "dsid_liv_dm1.dim_adresse_norm_restaurant_d", ("id_adresse_norm_restaurant",								
id_commande	id_date_commande	id_preparation	id_client	id_adresse_norm_client	id_restaurant	id_adresse_norm_restaurant	id	
1	1	1	1	1	1	1	1	
2	2	2	2	2	2	2	2	
3	3	3	3	3	3	3	3	
4	4	4	4	4	4	4	4	
5	5	5	5	5	5	5	5	
6	6	6	6	6	6	6	6	
7	7	7	7	7	7	7	7	
8	8	8	8	8	8	8	8	
9	9	9	9	9	9	9	9	
10	10	10	10	10	10	10	10	
11	11	11	11	11	11	11	11	
12	12	12	12	12	12	12	12	
13	13	13	13	13	13	13	13	
14	14	14	14	14	14	14	14	
15	15	15	15	15	15	15	15	
16	16	16	16	16	16	16	16	
17	17	17	17	17	17	17	17	
18	18	18	18	18	18	18	18	
19	19	19	19	19	19	19	19	
20	20	20	20	20	20	20	20	
21	21	21	21	21	21	21	21	
22	22	22	22	22	22	22	22	
23	23	23	23	23	23	23	23	

On a aussi rajouté une colonne qui contient le nom du jour à partir des dates de commande.



A ^B C date_commande_jour
dimanche
lundi
jeudi
vendredi
samedi
dimanche
lundi
mardi
mercredi
jeudi
lundi
mardi
vendredi
samedi
dimanche
mercredi
jeudi
vendredi
samedi
dimanche
mardi
mercredi
samedi

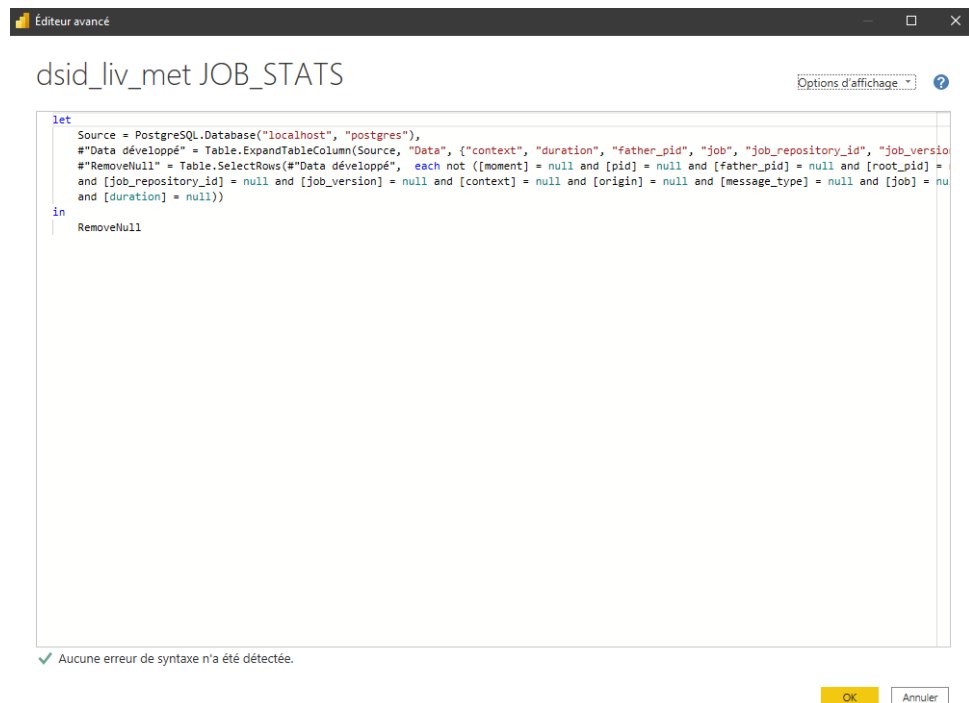
STAT:

Introduction:

Ce fichier .pbix, présente une série de visualisations dédiées à analyser les statistiques de performance des jobs exécutés par Talend.

Traitement PowerQuery:

Dans le cadre du traitement avec PowerQuery, nous avons préparé les données de la table stat issue de Talend en créant une copie sans les valeurs nulles. Cette démarche a pour but de faciliter l'analyse des jobs Talend selon leur état d'achèvement. Une requête M personnalisée, nommée "RemoveNull", a été élaborée pour filtrer les lignes en fonction de critères spécifiques, permettant ainsi de distinguer les jobs terminés de ceux qui ne le sont pas.



DM2:

Introduction:

Ce fichier .pbix, propose des visualisations et des indicateurs de performance clés (KPIs) pour analyser et comparer diverses variables liées aux livraisons et à la rémunération, en se basant sur des critères spécifiques.

Traitement PowerQuery:

Sur ce datamart pas de traitements ou de transformation à cause de la familiarisation avec les données ainsi que la granularité fine avec une structure élargie de données.

Construction des rapports :

Le rapport contient des pages de qui sont divisés en fonction du thème :

- La première page est une page de garde .
- Pour les DM1 et DM2 la deuxième page représente des cartes pour ces données
- Les autres pages contiennent des KPIs ou des visualisations, pour les graphes les plus compliqués une explication est fournie .

On peut naviguer entre les pages en cliquant sur la touche Ctrl+le bouton vers la page .

Difficultés rencontrées

Nous avons rencontrés plusieurs problèmes:

- Pour la modélisation nous avons eu beaucoup d'erreurs lorsque les noms des tables, des colonnes, et des contraintes dépassent 30 caractères.
- Pour la gestion des clés étrangères, il a fallu trouver le bon enchaînement des composants et le vidage des tables.
- Pour l'appel API, la difficulté était notamment sur le fait de trouver le bon JSONPATH et la récupération de ID_ADRESSE_CLIENT et ID_ADRESSE_RESTAURANT comme foreign key.

Lu par Version de l'API

Champ JSON

Requête de boucle JSONPath

Mapping

Column	Requête JSON
ID_ADRESSE_NORM_CLIENT	
ID_ADRESSE_CLIENT	
NUMERO_VOIE	"properties.housenumber"
NOM_VOIE	"properties.name"
CODE_POSTAL	"properties.postcode"
NOM_VILLE	"properties.city"
LONGITUDE	"geometry.coordinates.[0]"
LATITUDE	"geometry.coordinates.[1]"

"ADRESSE_NORMALISEE_CLIENT" (tDBOutput_3)(Oracle)

Paramètres simples

Paramètres avancés

Paramètres dynamiques

Vue

Documentation

Paramètres supplémentaires JDBC ""

*Note: Example for Additional JDBC Parameters: "parameterName1=value1¶meterName2=value2"

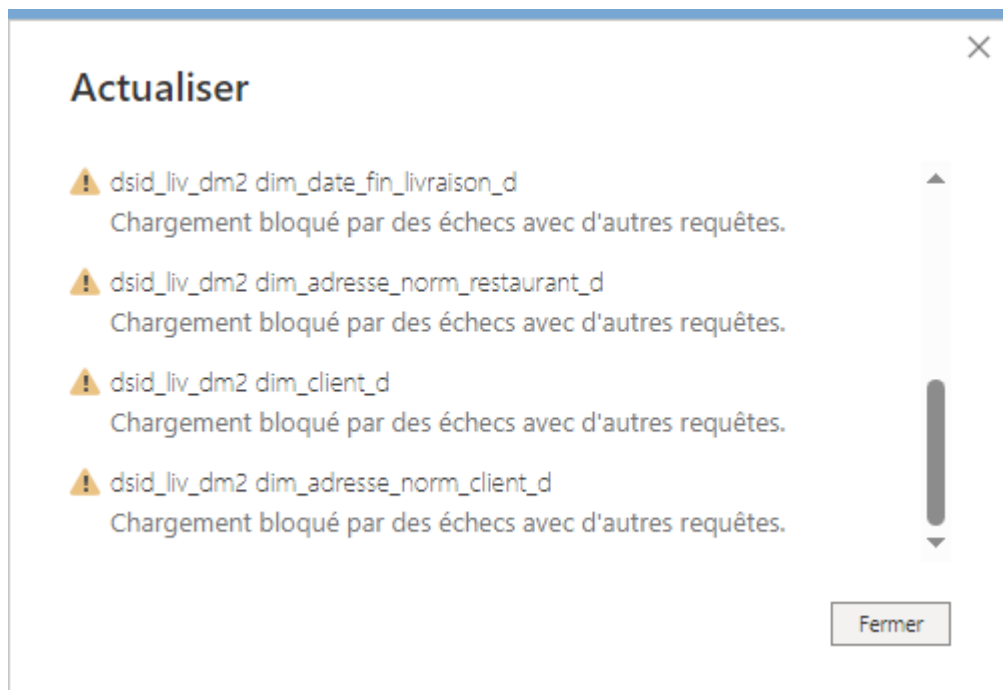
Commiter tous les

Colonnes supplémentaires

Nom	Expression SQL	Position	Colonne de référence
"ID_ADRESSE_NORM_CLIENT"	"dsid_liv_ope.seq_id_adresse_norm_client.nextval"	Remplacer	ID_ADRESSE_NORM_C
"ID_ADRESSE_CLIENT"	(((BigDecimal)globalMap.get("row1.ID_ADRESSE_CLIENT"))	Remplacer	ID_ADRESSE_CLIENT

- Pour l'appel API du DM2, nous avons perdu beaucoup de temps pour les paramètres start et end qui était latitude,longitude dans l'énoncé alors que c'est l'invers dans la documentation de l'API
- Pareil pour le code_moyen_paiement: driving-car au lieu de driving_car comme dans l'enoncé
- Les différents traitements en JAVA nous ont aussi posés pas mal de difficulté, notamment pour les conversions

- Pour la partie BI nous avons rencontrés quelques problèmes par rapport à la connexion à la source de données du DM2



Conclusion

Le projet nous a permis de monter en compétences sur les différents outils utilisés, à savoir DataModeler, Talend et PowerBI. En plus, cela nous a permis de renforcer notre pratique des différentes étapes de la BI en plus du TP vu en classe.

Lorsque nous avons des problèmes et/ou rencontré des erreurs nous avons pu trouver des solutions en consultant des forums de Microsoft PowerBI et StackOverflow.

Lors des erreurs sur Talend nous avons aussi pu essayer différentes façons de résoudre un problème, par exemple en utilisant des composants.

De cette façon, nous en savons un peu plus aujourd'hui et sommes plus autonomes et aptes à exercer dans le domaine professionnel.

Matrice de répartition:

Etape	Exécutant(s)
Modélisation SI OPE	Henintsoa, Ahcene
Modélisation DM1	Henintsoa, Ahcene
Modélisation DM2	Thierry, M'Hamed
Alimentation SI OPE	Henintsoa
Alimentation DM1	Henintsoa
Alimentation DM2	Thierry, M'Hamed
Power BI	Ahcene
Rapport	M'Hamed, Thierry