

TECHNIQUES D'OPTIMISATION SUR INFORMATICA POWER CENTER

Suivi du Document

Version	Date	Auteur	Modification
1.0	30/04/2021	KAWA Christian	Initialisation
1.1	11/07/2021	KAWA Christian	Mise à jour des technique optimisation (6 à 8)

TABLE DES MATIÈRES

TECHNIQUES D'OPTIMISATION.....	3
1. Optimisation des Sources et des Targets	5
2. Optimisation du Buffer Block Size (Mémoire tampon)	6
3. Push Down Optimization.....	6
4. Partitionnement des sessions (manuel).....	6
5. Partitionnement dynamique des sessions	7
6. Concurrent Workflow Exécution.....	8
7. Load Balancing for Workload Distribution on the Grid	10
8. Optimisation les traitements	11

9. Présentation des composants Informatica..... 12

- **Informatica Corporation** est un fournisseur indépendant qui propose des solutions d'intégration des données
- Les solutions Informatica permettent d'accéder, intégrer et fiabilité leur capital d'informations
- L'entreprise propose six produits principaux que sont :

- **Informatica PowerCenter**

- Informatica Data Quality

- Informatica MDM

- Informatica ILM

- Informatica B2B

- Informatica Cloud



Informatica PowerCenter

- Permet **d'accéder, découvrir et intégrer** les données de tous les systèmes métiers, quel que soit le format, et de les distribuer en temps voulu partout dans l'entreprise
- Est utilisé pour créer des mappings ETL entre les sources et les cibles

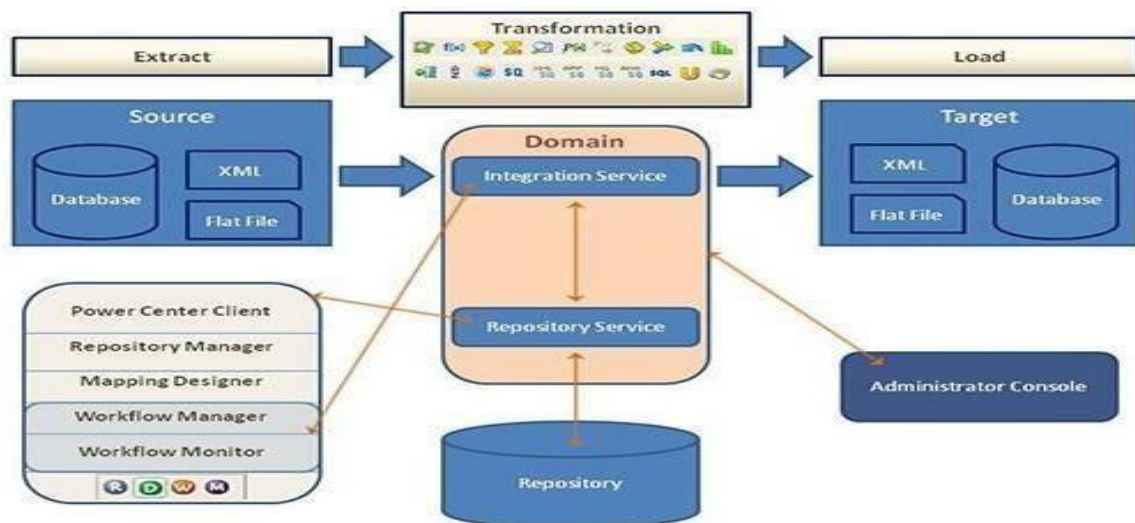
Informatica PowerCenter

Dispose de quatre services en particulier :

- Power Center Designer
- Power Center Workflow Manager
- Power Center Workflow Monitor
- Power Center Repository Manager

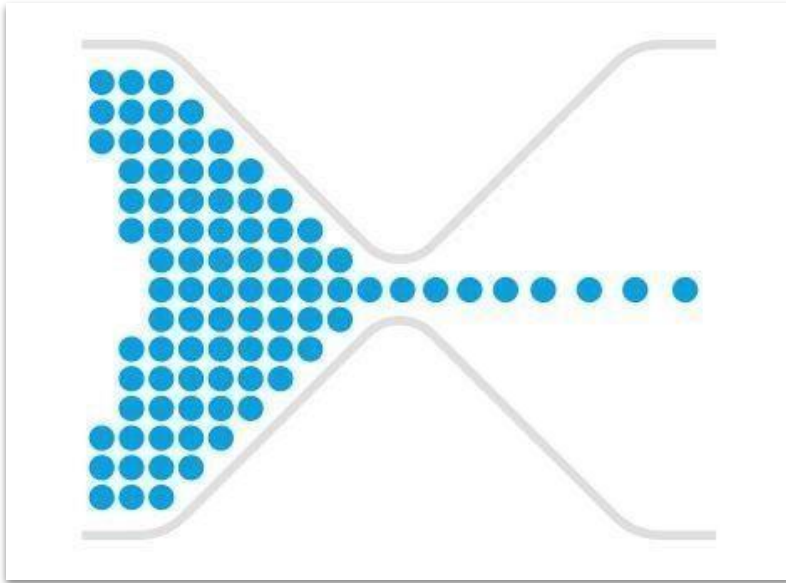


Informatica PowerCenter



TECHNIQUES D'OPTIMISATION

Bottlenecks : Freins de performance



Ne pas chercher à optimiser sans identifier les bottlenecks

- Ils peuvent se situer au niveau de la **source** et de la **cible**, du **Mapping**, de la **session** et du **système**.
- Pas de formule magique : Optimiser la performance globale → Itérer sur chaque bottleneck et améliorer l'un après l'autre
- Les **logs de session** aident à déceler les zones de frein. Les statistiques importantes sont
- Run Time
- Idle Time
- Busy Time
- Thread Work Time
- Le thread avec le plus haut busy percentage = bottleneck
- Comment configuration l'enregistrement des logs ?

1. Optimisation des Sources et des Targets

Bonnes performances → Pas de bottlenecks à la source et la cible

➤ **Equilibre de chargement (Load)**

Effectuer la lecture/écriture/tri/groupement/filtrage des données dans la base de données.

Utilisez Informatica pour la logique plus complexe, les jointures externes, l'intégration de données, les alimentations de sources multiples, etc.

➤ **Estimer la taille des données**

Connaître la taille des data sets d'entrée (nombre de lignes, nombre moyen d'octets par ligne) et de sortie.

➤ **Optimiser les requêtes**

Si une session joint plusieurs tables sources dans un seul Source Qualifier, on peut améliorer les performances en optimisant la requête à l'aide des Optimizer Hints.

➤ **Charger en masse (Bulk Loads)**

Utilisez la propriété de session Bulk Load qui insère une grande quantité de données dans une base de données Oracle, Microsoft SQL Server, etc → Contourne l'écriture dans le log → Accélère les performances

➤ **Supprimer les indices et clés primaires**

Définir les clés ralentit le loading des données dans les tables. Astuce : supprimer les indices et les contraintes de clés avant d'exécuter la session et reconstruire une fois la session terminée dans le Post SQL de la session.

➤ **Localisation**

Centraliser les procédures, fonctions, vues et séquences stockées dans la base de données source et évitez les synonymes. Si on lit à partir d'un fichier plat, c'est mieux de copier le fichier sur le serveur Informatica avant de lire les données du fichier.

- **Eviter les Level Sequences dans la BD**

- **Augmenter les Checkpoint Intervals de la BD**

A chaque checkpoint on perd de la performance. Astuce : augmenter les intervalles des checkpoints → diminuer le nombre de checkpoints.

2. Optimisation du Buffer Block Size (Mémoire tampon)

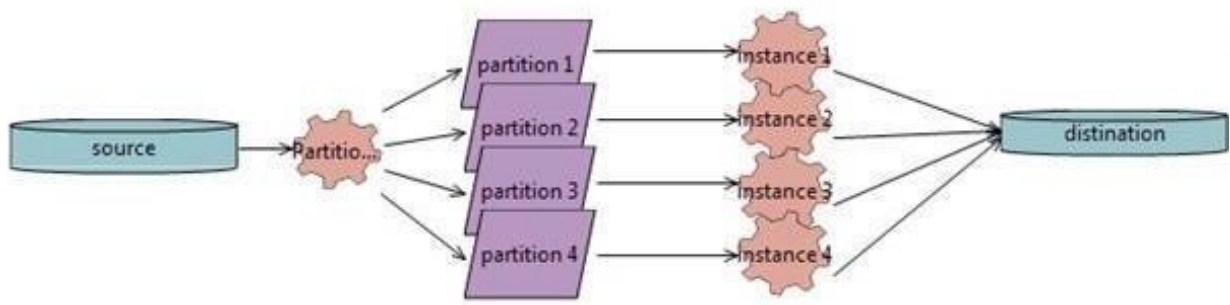
- Lorsque le service d'intégration initialise une session, il **alloue des blocs de mémoire pour contenir les données source et cible**. Les sessions qui utilisent un grand nombre de sources et de cibles peuvent nécessiter des blocs de mémoire supplémentaires.
- L'ajout de blocs de mémoire supplémentaires peut maintenir les threads occupés et améliorer les performances de la session. Vous pouvez y parvenir en **ajustant la taille des blocs de mémoire tampon** et la taille de la mémoire tampon DTM.
- Taille de bloc optimale de la mémoire tampon, additionnez la précision de chaque colonne de source et de cible → La plus grande précision = taille du bloc de mémoire tampon pour une ligne. Idéalement, un bloc tampon doit pouvoir contenir au moins 100 lignes à la fois.
- Augmentation de la mémoire tampon du DTM → Intégration Service crée plus de blocs tampons → Amélioration des performances.
- Mémoire tampon DTM optimale :

3. Push Down Optimization

- PDO : Transférer un job de traitement du serveur Informatica sur le serveur de la DataBase
- Améliore les performances globales des sessions (run time)
- 3 modes de Push Down : **Source**, **Target** et **Full**

4. Partitionnement des sessions (manuel)

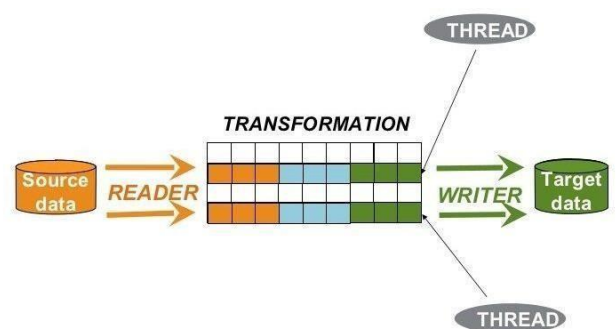
- Augmente les performances de PowerCenter grâce au traitement parallèle des données



- Nombre de partitions : Divise les data en plusieurs → Plus de threads → Augmentation de la performance de la session

5. Partitionnement dynamique des sessions

- Traitement des données en parallèle pour un processing plus rapide
- Power Center peut choisir dynamiquement le degré de parallélisme selon :
 - Le partitionnement de la source
 - Le nombre de CPUs
 - Le nombre de noeud dans la grid
 - Le nombre de partitions



Règles d'utilisation

- Ce partitionnement utilise la même connection pour chacune des partitions
- Ne fonctionne pas pour les sources et target XML
- Ne pas activer la « **dynamic partitioning** » si vous effectuez une partition manuelle.

Configuration :

Edit Tasks

General Properties Config Object Mapping Components Metadata Extensions

Select task: **s_m_T_FF**

Task type: Session

Config Name: default_session_config

Attribute	Value	UnOverride
Writer Commit Statistics Log Interval	0	
Error handling		
Stop on errors	1	Revert
Override tracing	None	
On Stored Procedure error	Stop	
On Pre-session command task error	Stop	
On Pre-Post SQL error	Stop	
Error Log Type	None	
Error Log DB Connection		
Error Log Table Name Prefix		
Error Log File Directory	\$PMBadFileDir\	
Error Log File Name	\$PMErorLog	
Log Row Data	<input checked="" type="checkbox"/>	
Log Source Row Data	<input type="checkbox"/>	
Partitioning Options		
Dynamic Partitioning	Disabled	
Number of Partitions	1	

Dynamic Partitioning

Enable dynamic partitioning and select whether the partition count is specified manually or determined automatically by the system.

OK Annuler Appliquer Aide

Edit Tasks

General Properties Config Object Mapping Components Metadata Extensions

Select task: **s_m_T_FF**

Task type: Session

Config Name: default_session_config

Attribute	Value	UnOverride
Writer Commit Statistics Log Interval	0	
Error handling		
Stop on errors	1	Revert
Override tracing	None	
On Stored Procedure error	Stop	
On Pre-session command task error	Stop	
On Pre-Post SQL error	Stop	
Error Log Type	None	
Error Log DB Connection		
Error Log Table Name Prefix		
Error Log File Directory	\$PMBadFileDir\	
Error Log File Name	\$PMErorLog	
Log Row Data	<input checked="" type="checkbox"/>	
Log Source Row Data	<input type="checkbox"/>	
Partitioning Options		
Dynamic Partitioning	Based on number of partitions	Revert
Number of Partitions	\$DynamicPartitionCount	

Number of partitions to be created during runtime. Enter a positive integer or the session parameter \$DynamicPartitionCount.

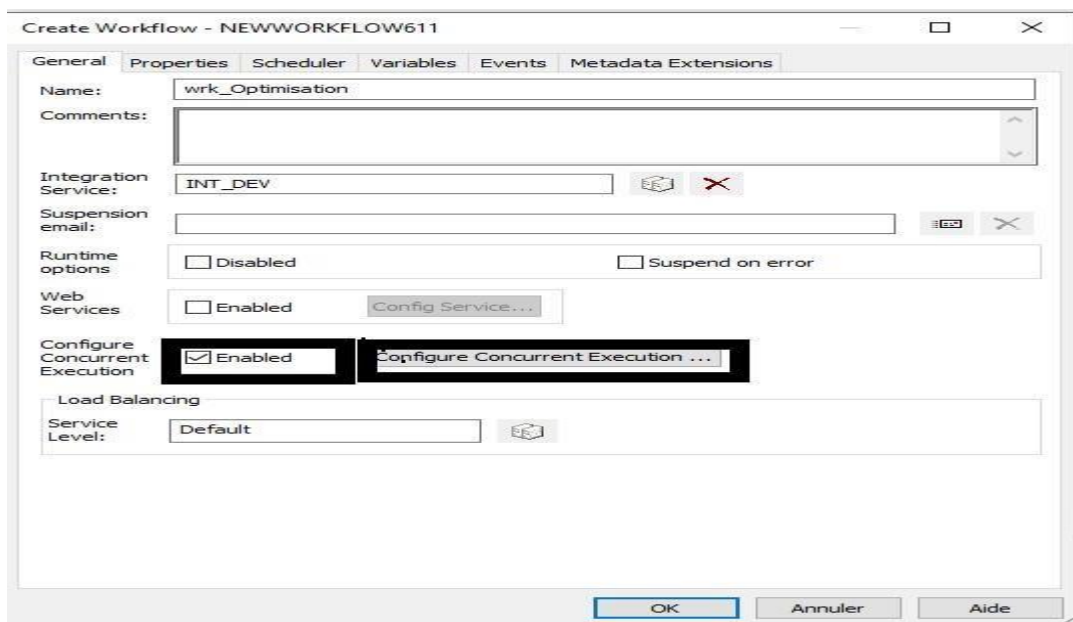
OK Annuler Appliquer Aide

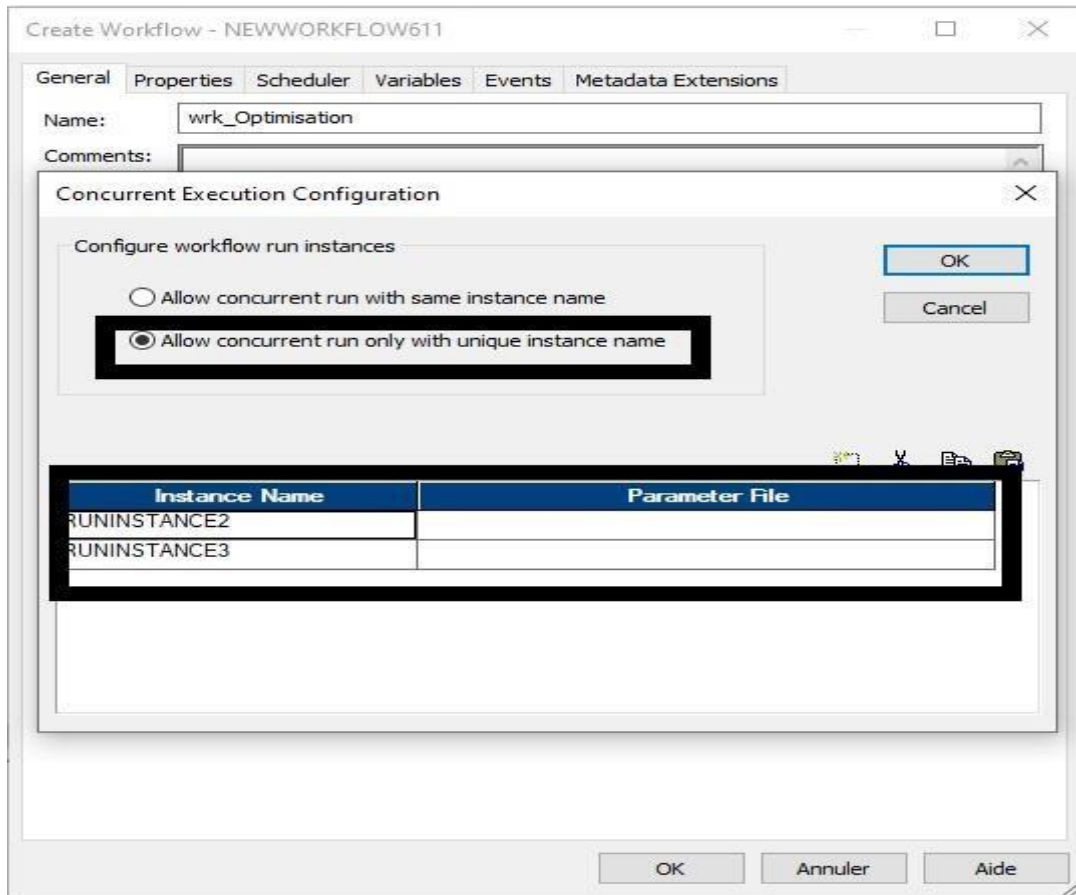
6. Concurrent Workflow Exécution

- Dans le cadre de projets à gros volume de données, il est possible de réduire le temps en exécutant plusieurs process ETL en **parallèle**

- Cette méthode permet de réduire considérablement le **load time**
- Un workflow simultané peut s'exécuter simultanément sous plusieurs instances.
 - Exécuter des workflows simultanés avec les mêmes noms d'instance
 - Exécuter plusieurs instances d'un workflow simultanément

Exemple : Considérons un scénario où nous devons charger des données de transactions quotidiennes à partir de différents sites





7. Load Balancing for Workload Distribution on the Grid

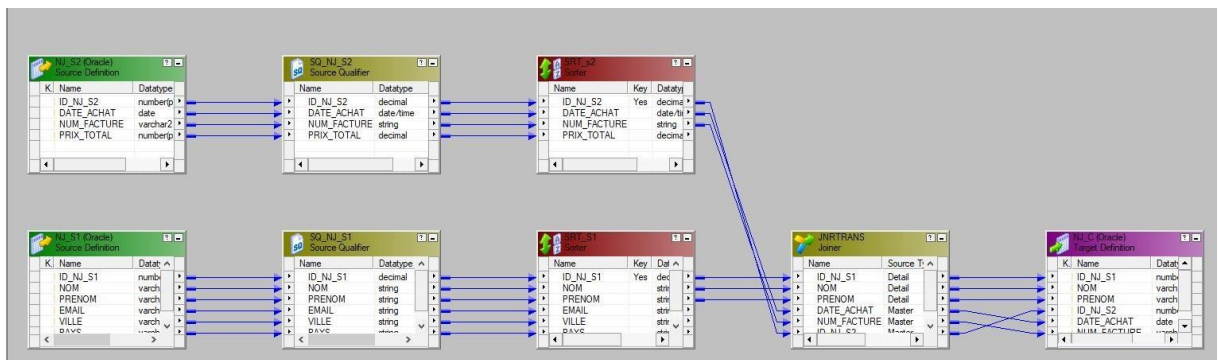
- **Informatica load balancing** est un mécanisme qui répartit les charges de travail entre les nœuds de la grille.
- Le Load Balancer fait correspondre les exigences des tâches avec la disponibilité des ressources pour identifier le meilleur nœud pour exécuter une tâche
- Les **Service levels** déterminent l'ordre dans lequel le répartiteur de charge répartit les tâches à partir de la file d'attente de répartition

8. Optimisation les traitements

➤ Optimiser un mapping avec des jointures

- On place devant le joiner un Sorter pour trier les données
- Et ensuite allé dans les parametre du joiner et cocher sorted Input

Exemple de mapping



Edit Transformations

Transformation | Ports | Properties | Condition | Metadata Extensions

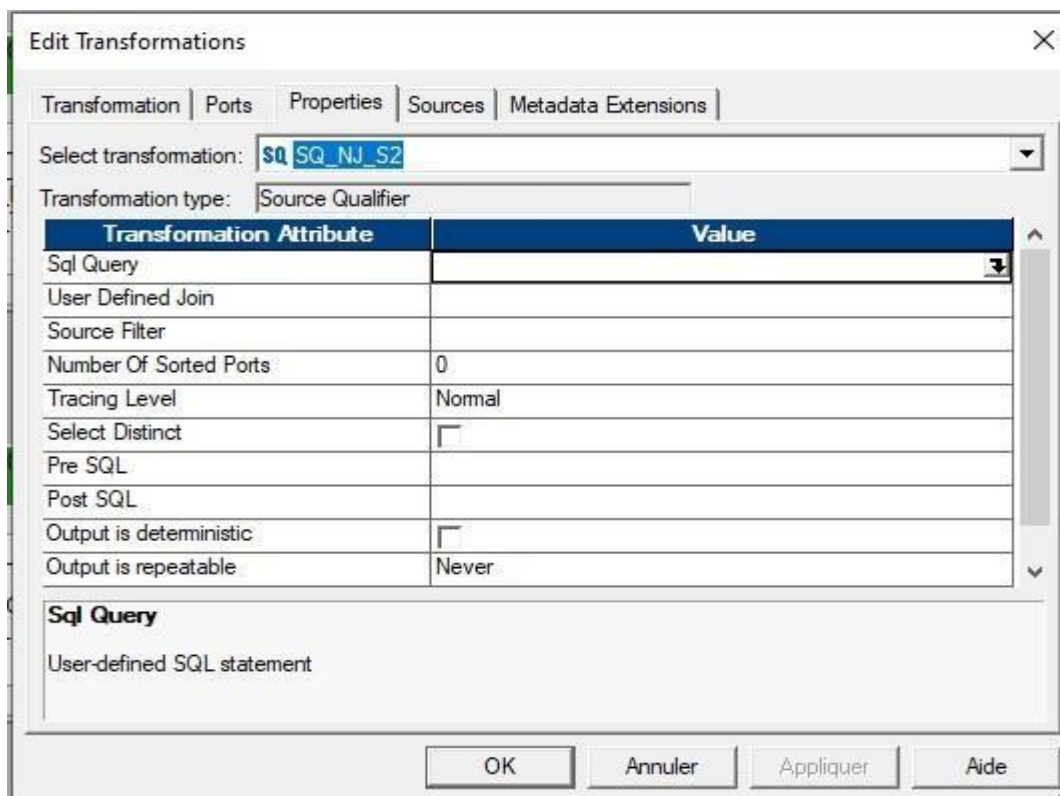
Select transformation: JNRTRANS


Transformation type: Joiner

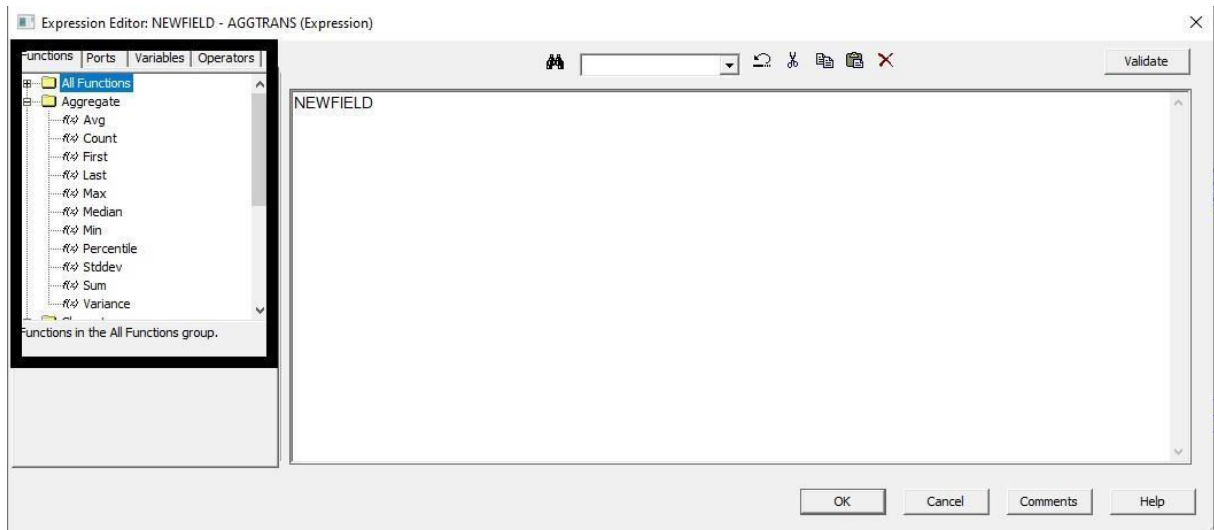
Transformation Attribute	Value
Cache Directory	\$PWCACHEDir
Join Condition	ID_NU_S2 = ID_NU_S1
Join Type	Normal Join
Null ordering in master	Null is Highest Value
Null ordering in detail	Null is Highest Value
Tracing Level	Normal
Joiner Data Cache Size	Auto
Joiner Index Cache Size	Auto
Sorted Input	<input checked="" type="checkbox"/>
Master Sort Order	Auto
Transformation Scope	All Input

9. Présentation des composants Informatica

- ✚ **Source Qualifier** : c'est une transformation active et connecté. Il convertie les types de données sources en types de données natives.

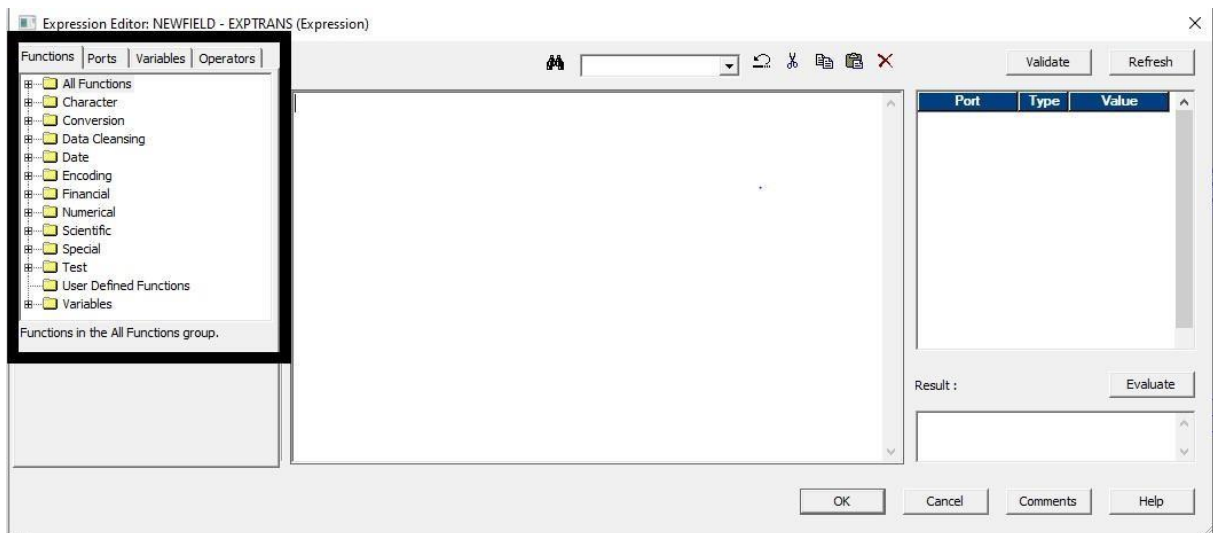


 **Aggregator** : pour faire des calculs et agréger les données. C'est une transformation qui permet de faire des calculs de type d'agrégat mais aussi le calcul des moyens, des minimums, des maximums, des comptages, des firsts pour récupérer la première ou des last pour récupérer la dernière valeur. C'est un peu comme des groupes by en SQL

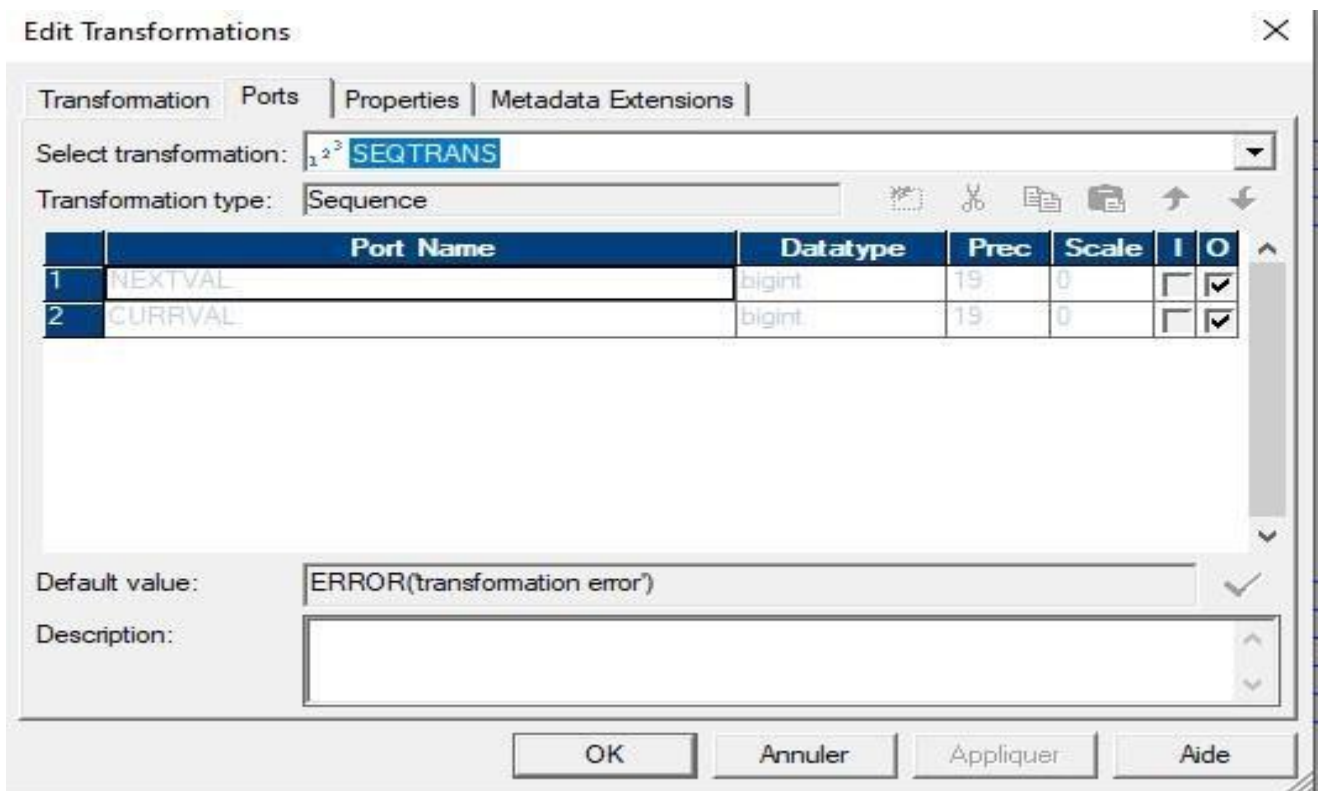


Document de note sur Informatica PowerCenter

- ✚ **Expression** : il permet d'effectuer toutes les transformations simples avant insertion dans la cible (transformation des chaînes de caractère, conversion, etc...)

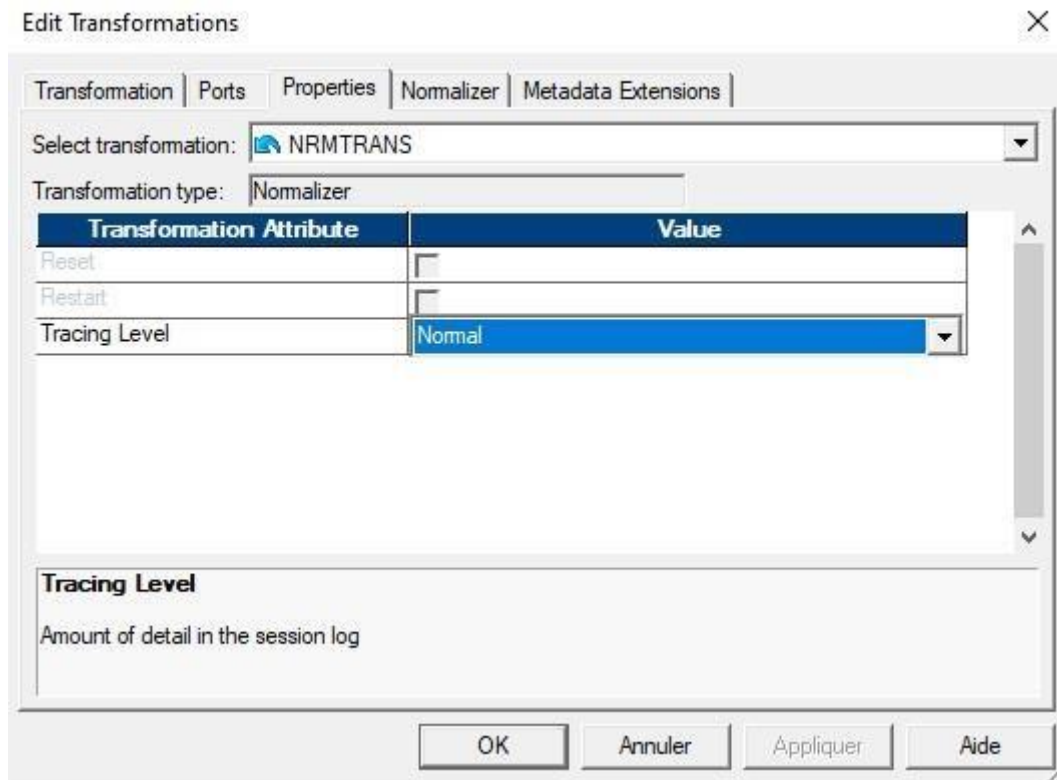


- ✚ **Séquence generator** : c'est une transformation passive qui génère les valeurs génériques. Il est utilisé pour la création des valeurs de clé primaire unique



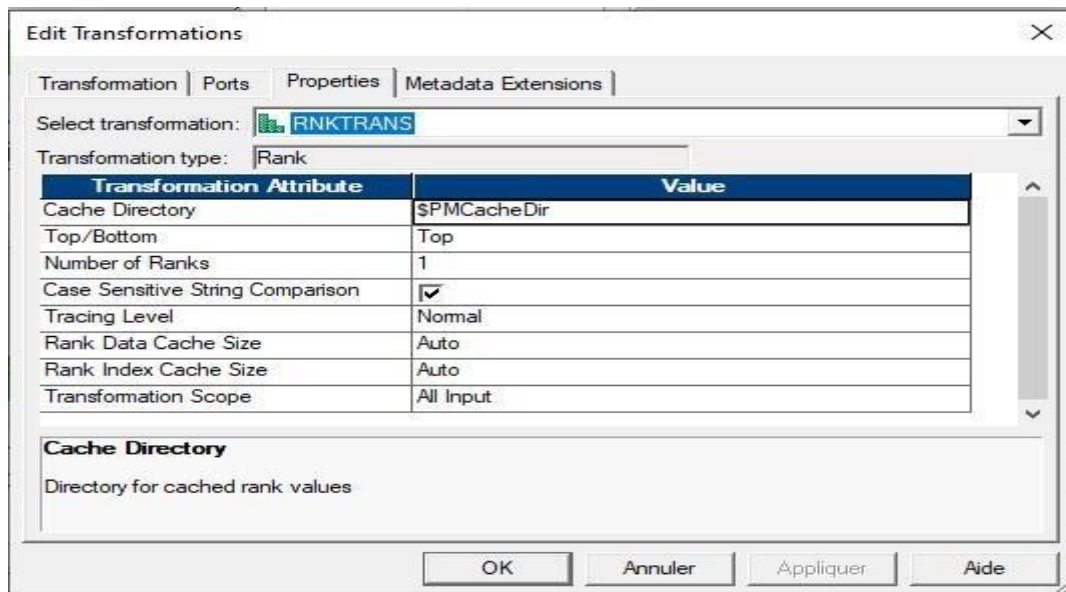
Document de note sur Informatica PowerCenter

- ✚ **Normalizer** : c'est une transformation active car elle permet de modifier le nombre de ligne. Elle transforme une ligne entrante en plusieurs ligne sortante

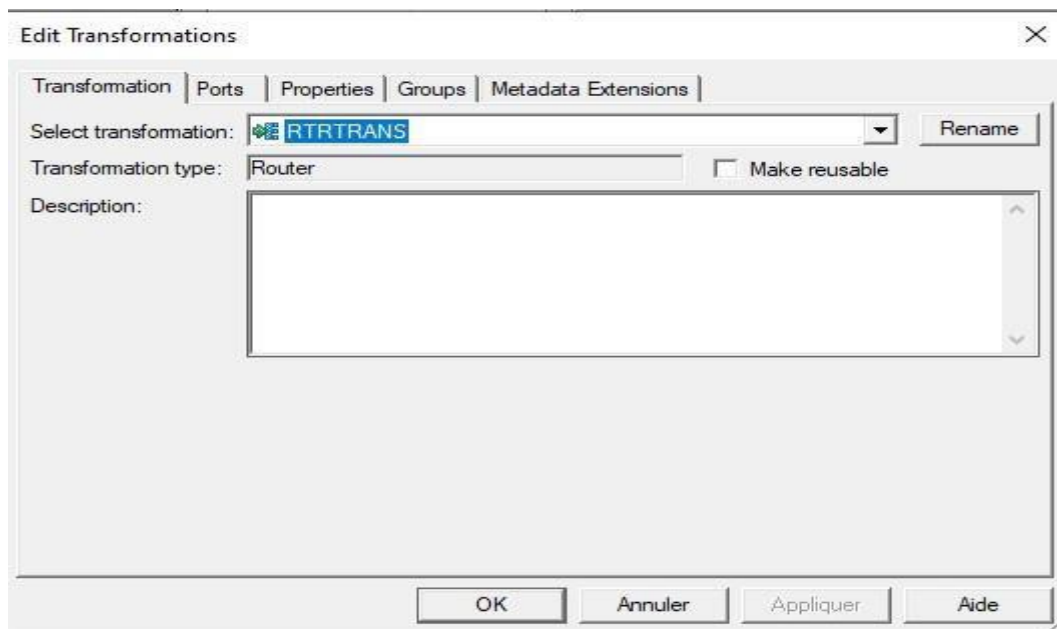


Document de note sur Informatica PowerCenter

- ✚ **Rank** : c'est une transformation qui permet de sélectionner le premier ou le dernier d'un classement sur un port

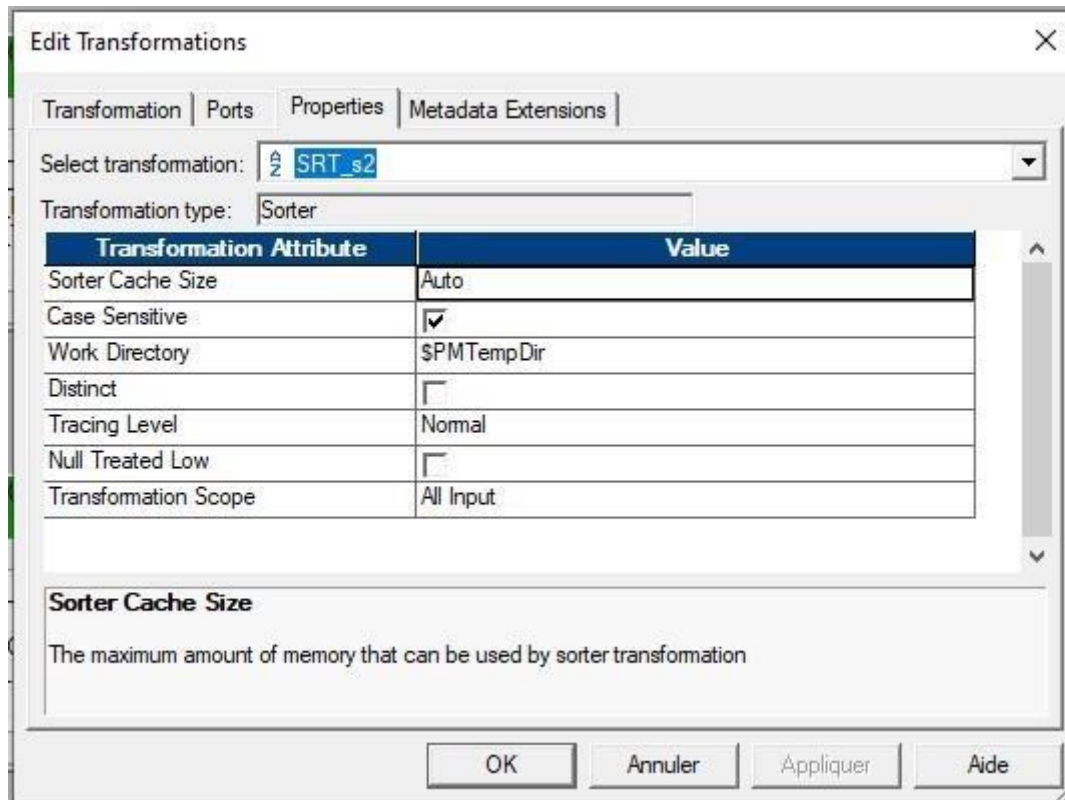


- ✚ **Router** : c'est un séparateur de flux selon les besoins

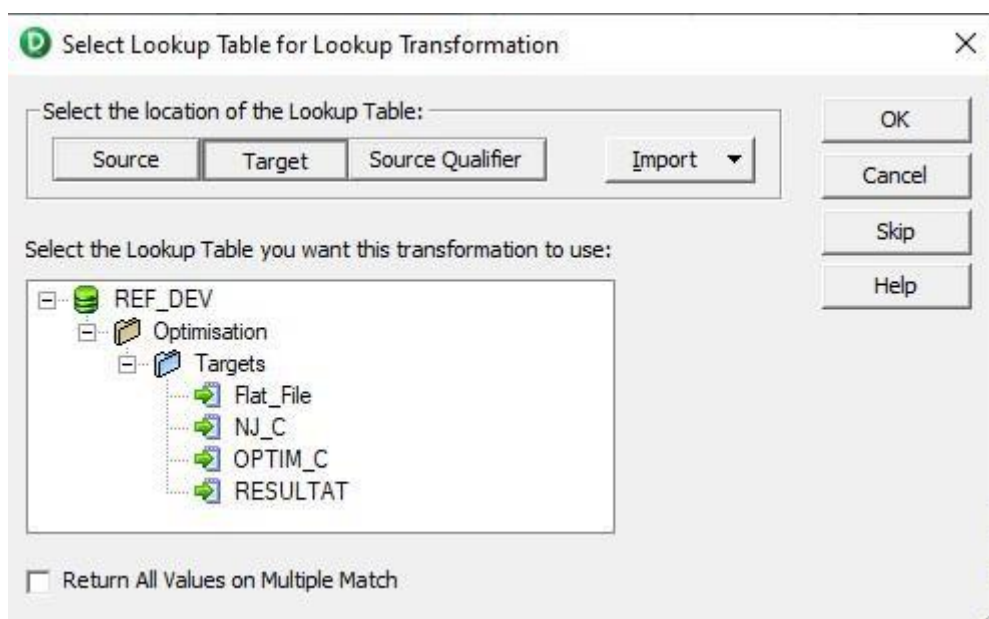


- ✚ **Le trier** : permet d'effectuer un tri sur les données, pour déboullonner les données. Il est généralement placé devant un aggregator ou un joiner

Document de note sur Informatica PowerCenter

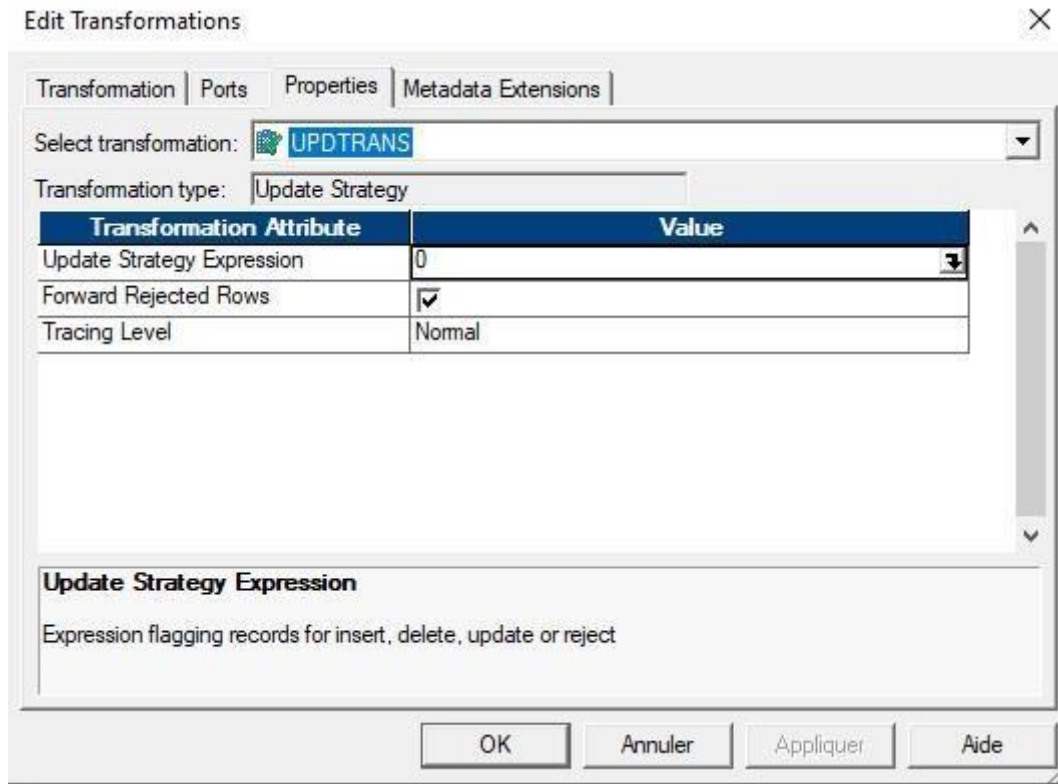


- ✚ **Look-up** : permet de faire de la recherche dans une table. Il permet de faire la comparaison entre les données en entré avec les données de la table interrogé, il permet aussi d'améliorer un calcul




Document de note sur Informatica PowerCenter

- ✚ **Update Strategy** : permet d'élaborer une stratégie d'alimentation et de mise à jour de la table cible en fonction des règles définir dans le mapping



Document de note sur Informatica PowerCenter

 **Union** : c'est une transformation active utilisé pour faire la fusion des données de plusieurs sources pour alimenter une seule cible.

