



VOS RECHARGEMENTS QLIK SONT TROP LONGS?

Voici **5** astuces concrètes pour
réduire le temps de chargement..
sans sacrifier la qualité des
données



AU PROGRAMME

Astuce 1 : Réduire le volume dès la source

Astuce 2 : Utiliser le chargement incrémental

Astuce 3 : Préférer l'ApplyMap() au JOIN)

Astuce 4 : Nettoyer du script

Astuce 5 : Utiliser les QVD ou les fichiers Parquet



Astuce 1 : Réduire le volume dès la source

Principe

Moins de colonnes + moins de lignes = moins de temps de chargement.

À faire tout de suite :

- Charger **uniquement les champs utiles** (commenter les champs inutiles et éviter les LOAD*)
- **Filtrer tôt** (WHERE sur la source si possible)
- **Compacter les clés composites avec Autonumber()**

Exemple : clé compacte + drop champs lourds

```
Fact:
LOAD
    OrderID,

    AutonumberHash128 (CustomerID,
    ProductID) as %Key_CustProd,

    OrderDate,
    Amount
FROM [lib://QVD/Fact_Src.qvd]
(qvd);

/* Si la clé composite ne sert
qu'au lien, on peut alléger :/*
DROP FIELDS CustomerID,
ProductID;
```



Astuce 2 : Chargement incrémental par "date repère"

Principe

Mémoriser la dernière date chargée, ne **prendre que les nouveautés** à chaque rechargement.

À faire tout de suite :

Stockez la date maximale dans un fichier QVD ou une variable, **puis utilisez-la** comme filtre **au prochain chargement**.

💡 Variante efficace : Utilisez un ID croissant (clé technique) si la date n'est pas fiable ou inexistante.

Exemple : chargement incrémental

```
// Récupérer la dernière date de  
l'historique QVD  
TmpMax:  
LOAD max(OrderDate) as MaxDate  
FROM [lib://QVD/Fact.qvd] (qvd);  
LET vLastDate = Date(Peek('MaxDate', 0,  
'TmpMax'), 'YYYY-MM-DD');  
DROP TABLE TmpMax;  
  
// Charger uniquement les nouvelles lignes  
NewRows:  
SQL SELECT OrderID, CustomerID, OrderDate,  
Amount  
FROM Sales  
WHERE OrderDate >= '$(vLastDate)';  
  
// Assembler & mettre à jour le QVD  
Fact:  
LOAD * FROM [lib://QVD/Fact.qvd] (qvd);  
CONCATENATE (Fact) LOAD * RESIDENT NewRows;  
DROP TABLE NewRows;
```



Astuce 3 : Moins de JOIN, plus d'ApplyMap()

Principe

Les JOIN lourds rallongent le reload et gonflent la mémoire.

À faire tout de suite :

Pour améliorer les performances dans Qlik Sense, **simplifiez le modèle en réduisant le nombre de tables** : intégrez les tables à 2 champs à d'autres **via la fonction de mapping, plus rapide qu'une jointure**.

Exemple :

```
// Petite dimension -- mapping
MapCustomerName:
MAPPING
LOAD CustomerID, CustomerName
FROM [lib://QVD/DimCustomer.qvd] (qvd);

Fact:
LOAD
    OrderID,
    CustomerID,
    ApplyMap('MapCustomerName', CustomerID,
'Inconnu') as CustomerName,
    OrderDate, Amount
FROM [lib://QVD/Fact.qvd] (qvd);
```





Astuce 4 : Nettoyer le script

Principe

- **Supprimez les tables temporaires** dès qu'elles ne servent plus (DROP TABLE)
- **Évitez** les Resident en cascade
- Résoudre les **clés synthétiques**
- **Chainez avec Preceding Load** quand c'est plus lisible/évite des tables intermédiaires
- **Limitez les calculs coûteux au strict nécessaire** par exemple des fonctions if()

Exemple de Preceding Load:

```
Tablename OrderDetails:  
Preceding LOAD LOAD *,  
LinePreDiscount * (1 - Discount)  
AS LineSalesAmount;  
  
LOAD  
OrderID,  
LineNo,  
ProductID,  
Quantity,  
UnitPrice,  
Discount,  
  
Transformed Date UnitPrice * Quantity  
AS LinePreDiscount  
  
Connection FROM [lib:/<connection>/<file>];
```

💡 Les clés synthétique peuvent vous faire perdre beaucoup en performance, supprimez les en priorités.



Astuce 5 : Ne pas dupliquer le chargement de sources identiques entre les applications

Principe

Augmenter la vitesse de chargement, réduire la taille du stockage et diminuer la charge sur les serveurs de base de données en utilisant les **fichiers QVD** (natifs Qlik) et les **fichiers Parquet** (format standard open source).

💡 Gardez les renommages et transformations dans la couche Transform et chargez la couche Load sans modification.

Exemple : QVD et Parquet

```
// Création d'un QVD
STORE Fact_Sales INTO
[lib://QVD/Fact_Sales.qvd] (qvd);

// Lecture optimisée
Fact_Sales:
LOAD * FROM [lib://QVD/Fact_Sales.qvd]
(qvd);

// Création d'un fichier Parquet
STORE Fact_Sales INTO
[lib://DataLake/Fact_Sales_2025.parquet]
(parquet);

//Lecture d'un fichier Parquet
Fact_Sales:
LOAD *
FROM [lib://DataLake/Fact_Sales.parquet]
(parquet);
```



Check-list pour accélérer vos rechargements

- Charger uniquement les champs et lignes utiles
 - Commenter les champs inutiles
 - Eviter les LOAD*
 - Utiliser le Where
- Utiliser l'Autonumber() pour les clés composites
- Supprimer les tables et champs dès que possible
- Utiliser le chargement incrémental (date ou clé)
- Résoudre les clés synthétiques
- Préférer l'applyMap() à la jointure
- Chaîner le script avec des Preceding Load
- Utiliser les fichiers les fichiers QVD ou Parquet



Prêt à transformer vos performances Qlik ?

💡 Quelle astuce allez-vous mettre en pratique dès aujourd'hui ?
Partagez votre choix en commentaire !

🚀 Passez au niveau supérieur : **inscrivez-vous à notre formation Qlik Sense Developer** et maîtrisez le scripting ainsi que l'optimisation de vos modèles de données.

📌 La formation incontournable pour devenir un expert Qlik.

👉 Contactez-nous : contact@altiview.fr | [LinkedIn Altiview](#)



Pour être informé(e) des nouveautés, abonnez-vous à notre page LinkedIn



Pour plus d'informations contactez nos experts

