# QlikView : modélisation en étoile

par Yves Ducourneau (Page personnelle) (Blog)

Date de publication : 14 février 2012

Dernière mise à jour :

Si QlikView a l'avantage d'être un produit ancien, mûr et très performant, il a aussi l'inconvénient d'être très permissif en matière de modélisation, domaine où l'un des écueils est de copier-coller le schéma de la base transactionnelle. Nous proposons ici quelques guides pour implémenter le schéma dit « en étoile ».

## 1 - Un peu de vocabulaire

Dans le monde de l'informatique décisionnelle, on appelle **« dimension »** une table qui sert d'axe d'analyse (région, produit, équipe commerciale, etc.) et **« table de faits »** une table qui contient les données étudiées (commande, facture, dépense, etc.).
Dans QlikView, **le temps** est une dimension comme une autre.

## 2 - Pourquoi un schéma en étoile ?

Pour le comprendre, nous devons regarder comment QlikView *sélectionne* les données en réponse à un filtre. Imaginons une application comportant une table de faits, des commandes, reliée à différentes dimensions, parmi lesquelles le vendeur. Imaginons que l'utilisateur souhaite visualiser les commandes enregistrées par un vendeur donné. Lorsque l'utilisateur sélectionne le vendeur, QlikView répond en affichant les valeurs correspondantes : on les appelle les valeurs « possibles ».
(Selon les conventions en vigueur, QlikView affiche les valeurs sélectionnées en vert vif, les valeurs possibles en blanc et les impossibles en gris. La figure ci-dessous suit cette convention, sauf le gris qui devient bleu.)
Pour identifier ces valeurs « possibles », QlikView parcourt le schéma en partant de la sélection (le vendeur) et en progressant de lien en lien :

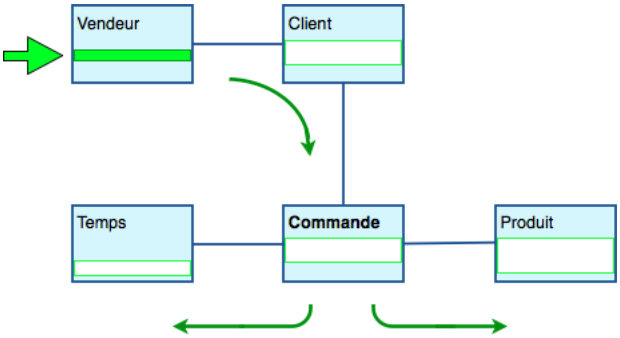


*Fig. 1 - Identification des possibles à partir de la sélection.*

Ajoutons maintenant une deuxième table de faits, les objectifs de vente, liée à deux dimensions existantes. Pas de chance : cette nouvelle table crée une *boucle* dans le schéma (autrement dit un « cycle »). Comment la sélection va-t-elle fonctionner ?

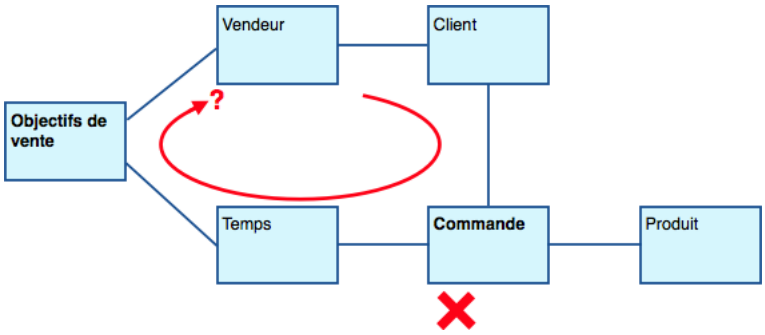


*Fig. 2 - Présence d'un cycle dans un schéma QlikView.*

Réponse : elle ne marche plus ! En présence d'un cycle, QlikView revient à son point de départ et peut alors trouver, dans la *même* table, des « possibles » différents de la « sélection ». Autrement dit, la valeur sélectionnée peut s'exclure elle-même des « possibles » ! Cela n'a pas de sens !
Pour autant, est-ce la « faute » de QlikView ? Pas forcément. En effet, le vendeur responsable de l'objectif n'est peut-être pas le même que celui qui est responsable du client ! Notre schéma a un cycle parce qu'il est *ambigu*.

**Il est important, dans QlikView, de n'avoir aucun cycle.** Un cycle signifie une ambiguïté. Si un cycle est présent, QlikView désactive des liens jusqu'à le faire disparaître.

## 3 - Les cycles : problème rare ou fréquent ?

Si notre problème précédent peut être résolu en créant une deuxième dimension 'Vendeur', plus les tables de faits seront nombreuses et plus la probabilité d'avoir des cycles sera grande, ne serait-ce que parce que les tables de faits sont en général aussi liées au temps.

Les conditions **favorables** (ne créant PAS de cycle) :

- Une seule table de faits.
- Plusieurs tables de faits liées exactement aux mêmes dimensions (même « granularité »).
- Une table de faits principale, liée aux dimensions, les autres étant des sous-tables de cette table.

Les conditions **défavorables** :

- Nombreuses tables de faits de granularités différentes (ventes hebdomadaires par vendeur, historique des ventes mensuelles par magasin, objectifs de ventes annuels par équipe, prévisions de vente par région et famille de produit, budgets par équipe et semestre, dépenses par vendeur, etc., le tout joyeusement mélangé dans les rapports pour que le bonheur soit complet).

À ceci s'ajoute une autre problématique : lorsque deux tables sont liées par *plusieurs* clés, QlikView crée automatiquement une **clé synthétique**, combinant les différentes clés et s'intercalant entre les tables sous forme d'une troisième. Ce mécanisme sert en général à supprimer des cycles mais il peut parfois en créer. Dans l'exemple ci-dessous, un cycle est supprimé grâce à une clé synthétique :

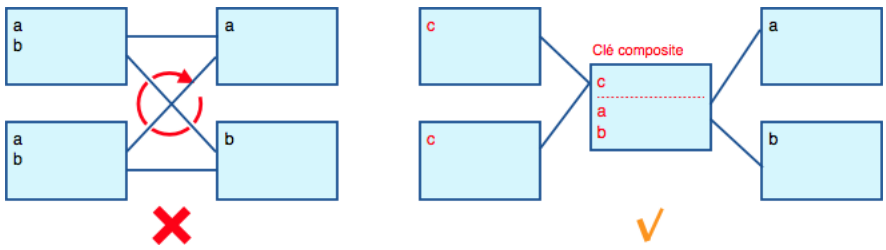


*Fig. 3 - Création d'une clé synthétique par QlikView, supprimant un cycle.*

Les clés synthétiques sont en général à **éviter**.
À nouveau, plus le schéma compte de tables de faits de granularités différentes et plus le risque est grand de voir les clés synthétiques se multiplier, et peut-être, créer des cycles. QlikView ne va pas s'en sortir.
Le schéma en étoile va résoudre tous ces problèmes.

## 4 - Schéma en étoile : méthode de la table creuse

Pour éviter que des cycles n'apparaissent dans le schéma, nous allons nous appuyer sur la facilité qu'ont les bases vectorielles à répéter les données à moindre coût. **Nous allons ranger tous les faits dans une seule table, autour de laquelle seront disposées les dimensions**. Nous laisserons vides les champs non concernés par un type de ligne donné.

Commençons par ajouter dans notre exemple une troisième table de faits, les actions commerciales. Un deuxième cycle apparaît :

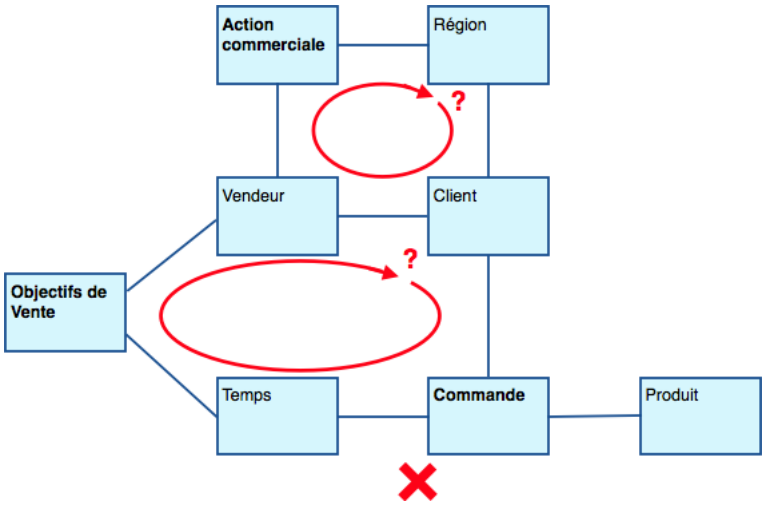


*Fig. 4 - Cycles dans un schéma QlikView comportant trois tables de faits.*

Réunissons maintenant nos tables de faits en une seule :

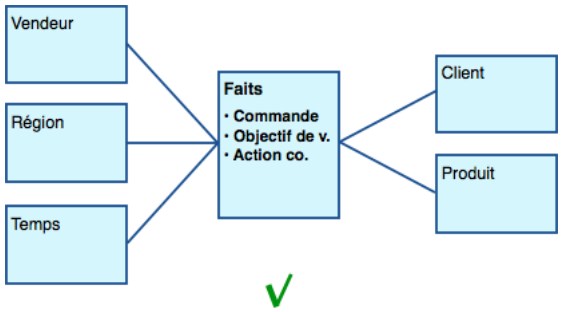


*Fig. 5 - Schéma en étoile avec table creuse, dans QlikView.*

Les cycles ont disparu.
Les clés dont nous avons besoin sont uniquement celles des dimensions :

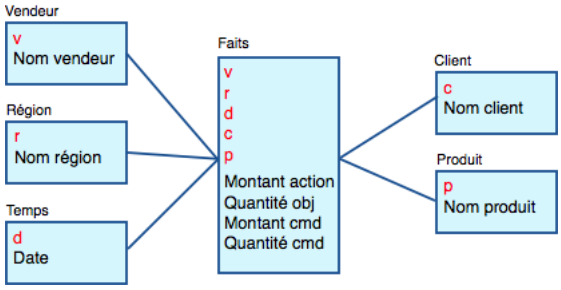


*Fig. 6 - Clés dans un schéma en étoile avec table creuse.*

On dit que la table centrale est « creuse » en raison du nombre de champs vides. Détail :

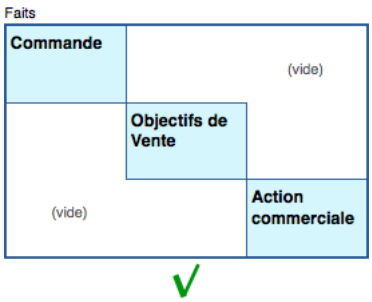


*Fig. 7 - Détail d'une table creuse.*

Explication : pour une ligne de type 'Commande', les champs « Montant action » et « Quantité obj » sont vides (valeur null()), de même que les éventuelles clés non utilisées.

Si la table centrale réunit trois tables, les 2/3 sont vides. Si elle en réunit six, les 5/6 sont vides. Et ainsi de suite. Plus on ajoute de tables et plus la table creuse est vide, mais dans une base vectorielle cela importe peu ! Cela peut surprendre lorsqu'on a l'habitude de la modélisation normalisée (c'est même une hérésie…) mais ce schéma est réellement très efficace avec QlikView !

Pour réunir plusieurs tables en une seule, il n'y a pas de mot-clé dans QlikView ; il suffit de charger exactement les mêmes colonnes. Par exemple :

```
Faits :
LOAD
   v,
   r,
   null()    as d,
   null()    as c,
   null()    as p,
   montant   as "Montant action",
   null()    as "Quantité obj",
   null()    as "Montant cmd";
   null()    as "Quantité cmd";
SELECT v, r, montant from action_commerciale;

LOAD
   v,
   null()    as r,
   d,
   null()    as c,
   null()    as p,
   null()    as "Montant action",
   quantite  as "Quantité obj",
   null()    as "Montant cmd";
   null()    as "Quantité cmd";
SELECT v, d, quantite from objectif_de_vente;

LOAD
   // etc.
```

Noter que les colonnes renvoyées par les deux instructions LOAD sont *parfaitement* identiques.
Les champs non concernés doivent être fournis avec une valeur nulle (par exemple, « r » dans le deuxième LOAD puisque les objectifs de vente ne sont pas liés à la région).
Attention, il faut faire la réunion des tables dans QlikView et non en SQL (avec union), car cela chargerait inutilement le réseau.

## 5 - Schéma en étoile : méthode de la table des clés

Un autre type de schéma en étoile, plus proche de la modélisation normalisée, consiste à mettre au centre une table des clés connectant toutes les tables.

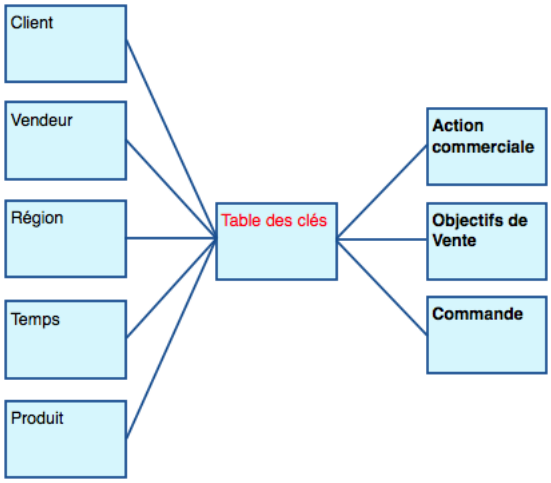


*Fig. 8 - Principe du schéma en étoile avec une table des clés.*

(Par simple convention, nous choisissons de mettre les dimensions à gauche et les faits, à droite ; cela n'a aucune conséquence sur le fonctionnement.)
La table centrale comporte à la fois des clés de dimension et des clés « composites » :

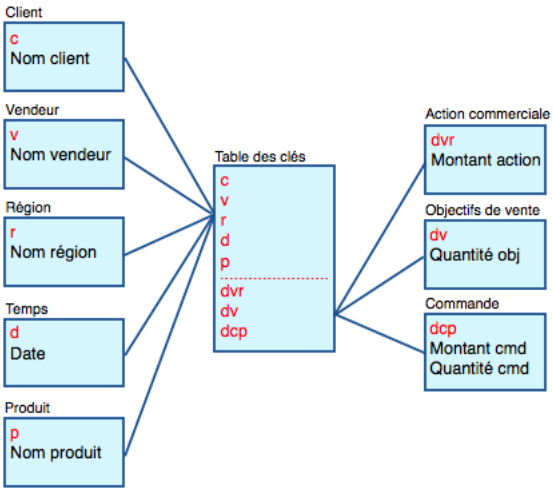


*Fig. 9 - Détail des clés dans un schéma en étoile avec table des clés.*

Les clés composites servent aux tables de faits.
À chaque clé composite correspond une granularité.
Une clé « composite » est une association de clés (élémentaires) de dimension.
La clé « dvr », par exemple, associe le temps (d), un vendeur (v) et la région (r). Elle est partagée par toutes les tables de faits s'exprimant par vendeur, région et date.

## 6 - Comparaison des deux méthodes

Voici les performances des deux méthodes, en taille et en vitesse de chargement, mesurées dans un cas réel :

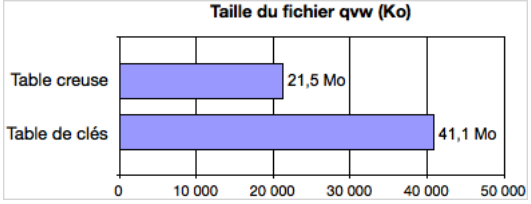


*Fig. 10 - Taille du fichier qvw pour deux méthodes de schéma en étoile.*

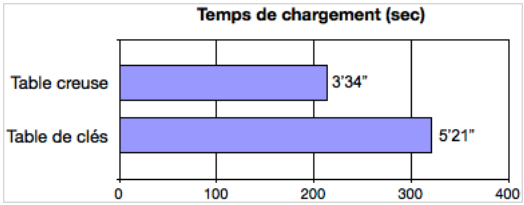


*Fig. 11 - Durée de chargement pour deux méthodes de schéma en étoile.*

La table creuse semble donc une excellente solution, ce qui confirme que, dans une base vectorielle, répéter les données ne nuit pas !

*Je dis « semble » car je n'ai, malheureusement, pas été en mesure d'effectuer des mesures de performances à l'utilisation de la méthode de la table creuse, domaine où la table des clés peut, éventuellement, prendre sa revanche !*

## 7 - Dernières remarques

Ces remarques sont valables quelle que soit la méthode choisie.

- Pour les faits comme pour les dimensions, des sous-tables peuvent bien entendu exister en périphérie (ça ne nuit pas). On parle de modèle « en flocon ».
- La table centrale peut accueillir des lignes ne comportant que des clés de dimensions, si l'on souhaite matérialiser certains liens entre dimensions. Par exemple, la matérialisation de la combinaison « temps * client » rend possible le comptage des clients à n'importe quelle date. Garder cependant un œil sur la volumétrie.
- On veillera, dans la table centrale, à toujours accompagner les clés des dimensions « fines » (le client par exemple) par les clés des dimensions « supérieures » (la région du client par exemple). Ne pas insérer, par exemple, un client sans sa région, sans quoi les sélections se feraient imparfaitement. Ajouter les clés manquantes dans QlikView grâce à un JOIN, plutôt qu'en SQL car cela augmenterait le trafic réseau. QlikView fait cela très rapidement.

## 8 - Conclusion

Dans une application mélangeant différentes tables de faits de granularités différentes, un schéma en étoile est une assurance contre l'apparition de cycles. Une méthode simple et efficace consiste à réunir tous les faits dans une table centrale unique, autour de laquelle sont disposées les tables de dimension.

P.-S. Contrairement à d'autres systèmes, QlikView ne fait pas la distinction entre dimension et table de faits. Cette distinction n'existe que dans l'esprit du concepteur, pour faciliter la modélisation.

## 9 - Remerciements

Je tiens à remercier **TomDuBouchon** pour la mise en forme de cet article et **ClaudeLELOUP** pour sa relecture.