

Integrix en Qlik Sense

Comment fonctionne la chaîne Integrix - QlikSense ?



1. Les bases de données de production des différents modules Integrix sont exécutées sur votre serveur D3/Universe (aka. Prod).
2. au sein, de ce système Prod, il y a l'outil ETL ¹⁾
 - ETL va convertir les données du format Multivalué (NF2) ²⁾ au format normalisé NF ³⁾ qui va permettre une exploitation avec les outils standards de la BI ⁴⁾.
 - Au cours de cette Extraction / transformation, les données cryptiques de Integrix sont rendues humainement lisible. ex: 1/0 → Oui/Non.
 - ETL va ensuite charger les données dans un système de base de données relationnelles de type SQL (mysql, oracle, db2, ...) (aka. DW ⁵⁾ et ce afin d'exposer les données au départ d'un point central.
 - En effet, de part sa configuration, le DW permet de regrouper les données de Prod issues de plusieurs environnements homogènes ou hétérogènes (plusieurs instance de Accountix, plusieurs instance de Stockix, ...) mais aussi il permet de regrouper plusieurs instances d'Integrix (plusieurs serveurs Integrix travaillant pour un groupe multi-société par exemple)!
 - selon une planification définie, ETL maintien à jour les données de Prod vers DW.
3. Le serveur QlikSense est planifié pour rafraîchir les données chargées dans les Qlikapp
 - une première phase (0 sql) va simplement lire les données dans le DW et les convertir au format Qlik de base
 - une deuxième phase (1 qvd) va d'une part simplifier le modèle en fusionnant certaines tables mais aussi enrichir le modèle en y ajoutant des **champs calculés**, des **calendriers**, et peut être des données issues de sources secondaires telles que fichiers Texte, fichier Excel, autres base de données , ...
 - une troisième phase (Qlikapp) charge les données enrichies dans la Qlikapp qui sera exploitée par l'utilisateur.
 - cette structure en trois phase permet de traiter les données de différents modèle (stockix, garagix, accountix, savix, ...) séparément et ensuite permet de fournir une Qlikapp personnalisée qui intègre des données de plusieurs modules distincts. exemple : la Qlikapp Garagix exploite les données de Garagix et de Stockix.
4. La/les Qlikapp sont dès lors mises à disposition des utilisateurs soit via QlikSenseServer soit via un export et un réimport en QlikSense Desktop.

Qlikapp Integrix

Comment est structurée une Qlikapp sur Integrix ?



XX représente le module applicatif au sein de Integrix (AC = Accountix, SX = Savix, ME = Medix, ...)

répertoire xx_init

xx_init est un répertoire de stockage situé dans le serveur QlikSense, répertoire Qvd/ xx_init.

fichier excel xx_ici.xlsx

On y trouvera un fichier Excel (xx_ici.xlsx) qui contiendra des valeurs propres aux sites d'exploitation.

Ces valeurs seront contenues dans des onglets excel et structurés en 'Tableau' avec entête.

L'onglet **Variables** contient les colonnes **Variable** et **Expression** ; ces valeurs seront chargées dans les qlikapp 0/1/app afin de fournir des variables propres au site. Une variable peut référencer une constante de valeur ou une expression de filtre, de sélection,...

Chaque module Integrix possède son xx_ici.xlsx et doit être considéré lors de la mise en oeuvre du site.

Cette méthode permet de mettre à disposition du site des Qlikapp générique propre au module integrix standard et de les

personnaliser.

qlikapp xx 0 sql

- **0 sql** représente la phase (0=initiale) et prend sa source (sql) dans le DW.
- cette qlikapp est interne au système et ne concerne pas les utilisateurs, elle permet de préparer les données normalisée vers le format Qlik.
- le résultat est sauvegardé au format QVD ⁶⁾ dans le serveur QlikSense, répertoire Qvd/xx_init

qlikapp xx 1 qvd

- **1 qvd** représente la phase d'enrichissement du modèle qlik.
- cette qlikapp est interne au système et ne concerne pas les utilisateurs, elle prend sa source dans le xx_init préparé par **0 sql**
- elle permet d'enrichir la Qlikapp. Principalement, on y fera :
 - la fusion des paramètres vers les signalétiques
 - la fusion des signalétiques vers les transactions
 - l'élimination des petites tables fusionnées
 - le calcul de valeurs intermédiaires : Exemple
 - $[\text{€ S/Total}] = [\text{Prix Unitaire}] \times [\text{Qte}]$
 - $[\text{€ Marge Unitaire}] = [\text{Prix Unitaire}] - [\text{Prix de revient}]$
 - $[\text{€ Marge Total}] = ([\text{Prix Unitaire}] - [\text{Prix de revient}]) \times [\text{Qte}]$
 - $[\text{\% Marge}] = ([\text{Prix Unitaire}] - [\text{Prix de revient}]) / [\text{Prix Unitaire}]$
 - ...
 - le calcul de Flags servant aux filtres rapides : Exemple :
 - [is Tiers NewBiz] ; un flag qui défini si une ligne de transaction est de type 'New Business' ... *si elle est dans l'année de première facturation d'un client.*
 - [is Marge Ok] ; un flag qui défini si la Marge est supérieure à 10%
 - ...
 - le calcul des [Calendriers](#)
- le résultat est sauvegardé au format QVD ⁷⁾ dans le serveur QlikSense, répertoire Qvd/xx_qvd

qlikapp xx Dft/Site

- **Dft/site** représente la qlikapp finale mise à disposition des utilisateurs
- elle se nourrit des données préparées par **1 qvd** et xx_ici.xlsx
- elle charge les calendriers secondaires, identiques aux calendriers calculés, mais un peu moins performant.

convention

Les conventions de nomination des champs.

Afin de comprendre les données à disposition pour créer les vues dans qlikSense, nous avons adopté une convention de nomage des champs.

Les champs peuvent contenir des espaces et des caractères spéciaux.

On utilisera le codage 'CamelCase' : une majuscule veut dire un nouveau mot.

les préfixes

Les préfixes permettent de comprendre la nature du champs :

- **#** représente des unités (qte) (ex : [# Qte], [# Stock], [# Heures], ...)
- **€** représente des valeurs monétaires (ex : [€ Prix Unitaire], [€ Base], [€ Marge])
- **%** représente des coefficients multiplicateurs (ex : [% TauxTVA], [% Marge], [% Coefficient],...)
- **is** représente un flag de valeur Vraie{1} ou Fausse{0} ou autre {2,3,...} selon le flag (ex : [is Tiers NewBiz], [is Cloture], [is in year to Date], ...)
- une lettre suivie d'un _ permet de simplifier les noms des tables courantes (c_ client, d_ document, sav_ sav, ...) chaque qlikapp documente son modèle et ses simplifications de nom

les constantes

- **xxxxx_Id** représente un champ à valeur unique identifiant une clé dans une table ex: c_Id (client Id) , sav_Id, ... (INFODATA_DD, INFODATA_EFV,) (INFODATA_12456, INFODATA_45789)
- **xxxxx_Code** représente un champs de valeur 'Code' sans référence unique ex: c_Code (client Code) , sav_Code, ... (DD, EFV,) (12456, 45789)
- **xxxx_D_xxxx** représente une valeur du [Calendrier](#) dérivée d'un champ de date (ex : [dateIn_D_Year], [DateIn_D_Month],...)
- *Date de* faire référence à un champ Date (ex : [Date de commande], [Date de Saisie], [Date de cloture], ...)

les variables

Pour distinguer les variables nous adoptons la convention suivante :

- variable de *set attribute* préfixée de **with_table_field_nom-de-filtre** ex : \$(with_doc_type_FSNV)
- variable constante préfixée de **_** pour string ex : \$(_FS)
- variable d'input préfixée de **v** ex : vNom

calendrier

Les détail de l'extension 'Calendar' - qui porte la déclinaison des Dates en différentes valeurs.

Il est important de bien comprendre que :

- les données qui sont des dates sont fixes (date de la facture, date d'entrée du SAV, date de clôture du SAV, date de livraison, ...)
- étant donnée que TOUTES les dates du calendrier (01/01 → 31/12) n'existent pas dans nos données, nous allons fabriquer une table 'Calendrier'
 - dont la plus petite date sera celle de la plus ancienne valeur connue dans le champ date de référence.
 - et la plus grande date sera celle de la plus récente valeur connue dans les données dans le champs date de référence.
 - (une qlikapp peut porter plusieurs Calendrier, chacun référençant un champ date particulier)
 - le Calendrier permet de sélectionner des dates qui n'existent pas dans nos données, on pourra dire : le dimanche dd/mm/yyyy il y a zéro vente; tout en montrant la date dd/mm/yyyy.
 - Les noms des déclinaisons sont toujours les mêmes et en anglais pour ne pas confondre avec des données du modèle applicatif.
- chaque date du calendrier va être
 - décomposées selon différents formats
 - Année, Trimestre, Mois, Semaine, AnnéeTrimestre, annéeMois, AnnéeSemaine
 - évaluée par rapport à 'today()' ; fonction Qlik qui donne la date du jour **! du dernier rafraîchissement !** (la date du dernier rafraîchissement et donc du today() dans la qlikapp est lisible avec la fonction RefreshTime())
 - xxxx**Ago** : QuantitéRéelle décalage en année, Trimestre, Mois Semaine par rapport à Today()
 - xxxx**RelNo** : QuantitéRelative : calcul du nombre d'unité temporelle indépendamment de l'année
 - xxxx**is_In?** : Est-ce que la date est dans la période ?

Le tableau ci-dessous illustre les valeurs obtenues selon les déclinaisons évoqués.

Les explications

Suffixe	Description	Exemple de valeur humaine
Year	le format Année	2018, 2015, 1992
Quarter	le format Trimestre	Q1, Q2, Q3, Q4
Month	le format Mois	jan,...,juil,...
Week	le format Semaine	W1, W5,...W52
Year+Periode		
YearQuarter	le format Année avec le format Trimestre	2015-Q1, 2018-Q3, ...
YearMonth	le format Année avec le format Mois	2015-jan., 2018-juil., ...
YearWeek	le format Année avec le format Semaine	2015-Q1, 2018-Q3, ...
xxxAGO	calcule le nombre absolu de ? entre la date et Today()	
YearsAgo	le nombre d'année entière qui séparent today() de la date de référence	0,1,2,3, ... (ex : 2018-2016→2)
QuartersAgo	le nombre de trimestre qui séparent today() de la date de référence	-1,-2,-999 ,0,1,2,3,...,999
MonthsAgo	le nombre de mois qui séparent today() de la date de référence	-1,-2,-999 ,0,1,2,3,...,999
WeeksAgo	le nombre de semaine qui séparent today() de la date de référence	-1,-2,-999 ,0,1,2,3,...,999
xxxRelNo	calcule le nombre relatif de ? entre la date et Today() ; ne tient pas compte de l'année dans le calcul	
QuarterRelNo	le nombre de trimestre relatif qui séparent today() et la date de référence	-3,-2,-1,0,1,2,3
MonthRelNo	le nombre de mois relatif qui séparent today() et la date de référence	-11,...,0,...11
WeekRelNo	le nombre de semaine relatives qui séparent today() et la date de référence	-51,...,0,...,51
is _ In?TD	flag Vrai/Faux si la date de référence est dans ? et = à today()	
is_InYTD	is In Year To Date est vai (1) si le numéro du jour (1..352) de la date de référence est < = au numéro du jour de today() dans tenir compte de l'année	1/0

Suffixe	Description	Exemple de valeur humaine
is_InQTD	is In Quarter To Date est vrai (1) si le trimestre (1..4) de la date de référence est < = au trimestre de Today() sans tenir compte de l'année.	1/0
is_InMTD	is In Month To Date est vrai (1) si le mois (1..12) de la date de référence est < = au mois de Today() sans tenir compte de l'année.	1/0
is_InWTD	is In week To Date est vrai (1) si la semaine (1..53) de la date de référence est < = à la semaine de Today() sans tenir compte de l'année.	1/0
is_InCurr?TD	flag Vrai/Faux si la date de référenc est dans ? courante ; courant est défini par today()	
is_InCurrYTD	is In current Year to Date est vrai (1) si la date de référence est dans la même année que today() et < = today() en tenant compte de l'année	1/0
is_InCurrMTD	is In current Year to Date est vrai (1) si la date de référence est dans le même mois que today() et < = today() en tenant compte de l'année et du mois	1/0
is_InPrev?TD	flag Vrai/Faux si la date de référenc est dans ? précédente; précédente est défini par today()	
is_InPrevYTD	is In current Year to Date est vrai (1) si la date de référence est dans l'année précédente à today() et < = today() en tenant compte de l'année précédente	1/0
is_InPrevMTD	is In current Year to Date est vrai (1) si la date de référence est dans le même mois que today() dans l'année précédente à today() et < = today() en tenant compte de l'année précédente et du mois	1/0

Les exemples

1)

ETL : Extract, Transform, Load <https://fr.wikipedia.org/wiki/Extract-transform-load>

2)

NF2 : Non first Normal Form : https://en.wikipedia.org/wiki/Unnormalized_form

3)

NF : Normal Form : [https://fr.wikipedia.org/wiki/Forme_normale_\(bases_de_donn%C3%A9es_relationnelles\)](https://fr.wikipedia.org/wiki/Forme_normale_(bases_de_donn%C3%A9es_relationnelles))

4)

BI : Business Intelligence : https://en.wikipedia.org/wiki/Business_intelligence

5)

DW : Datawarehouse : https://en.wikipedia.org/wiki/Data_warehouse

6) 7)

QVD : Qlik View Data