

GROUPE ASKIDA

# RAPPORT

## Analyse de Base de Données HydroSoft

---



---

## Introduction

L'objectif de ce document est de présenter l'analyse détaillée de la base de données d'Hydro Solution. Ce document est un effort fait afin de mieux comprendre l'application existante. Son squelette est une base de données SQL Server 2012© et utilise Great Plains©. Les problèmes existant sont, dans les grandes lignes, dues au manque de <<compartmentalisation>> de données. Tout réside dans le même schéma de Great Plains©, avec dédoublement et copie de tables, en plus des tables propres à l'application. Les agents externes causent beaucoup de transactions donc trop de pression sur la base de données.

## Serveur SQL 2012

L'analyse a été effectuée dans l'environnement de développement d'HydroSoft. L'environnement de production pourrait donner des résultats différents, mais nous sommes confiant que le résultat de cette analyse nous a donné les pistes suffisantes pour recommander et orienter les actions à prendre pour le développement de la nouvelle application.

## Schéma de base de données

Les nombreuses transactions entre les tables originales de Great Plains© et les tables HydroSoft (de l'application et en lien avec Great Plains©) causent la surutilisation de mémoire et d'espace disque. Il faudra découpler les nouveaux développements dans un autre Schéma de données (i.e. HydroSoft) pour éviter la saturation.

---

## Base de Données

### Espace de disque et mémoire

L'environnement de développement auquel nous avons accès, est assez léger car il consomme presque 900 Mo de mémoire mais prends 620 Go sur disque en plus de résider dans un fichier physique MDF , ceci n'est pas optimal.

### Contraintes

Décompte du nombre de contraintes par table, les tables ayant plus de 7 contraintes par table appartiennent a Great Plains©, donc les tables de HydroSoft moins lourdes.

### Utilisation d'objets

L'environnement de développement nous montre aussi que la base de données est centrée sur des procédures stockées plus que sur des fonctions.

### Nombre d'enregistrements pour chaque table

Il y a au-delà de 1000 tables qui n'ont pas d'enregistrements (non utilisées). On estime à 150 tables le nombre de tables de configuration ayant entre 1 et 10 enregistrements. Les table ayant plus de 1000 enregistrements sont les plus utilisées tandis que celles ayant de 100 à 1000 sont les plus liées à la logique d'affaires.

---

## Les fonctions

Plus de 300 fonctions sont présentes. La minorité sont “valuées” retournant un ensemble de données suite à une sélection. La plupart sont “scalaires” retournant une seule valeur. Les fonctions sont optimisées pour diminuer la quantité du flux de données.

### Les paramètres des fonctions

Toutes les fonctions reçoivent des valeurs en paramètre.

### Distribution du nombre de paramètres des fonctions

Dans la minorité des “valuées”, la plupart ont 4 paramètres mais moins que 10, ce qui vans dans le sens de la norme. Par contre dans la majorité des “scalaires”, la plupart ont 2, il faudrait investiguer les fonctions ayant de 10 à 20 paramètres pour des possibles optimisations.

## Les procédures stockées

Le système comporte au delà de 17 000 procédures stockées, dont la plupart n'utilisent pas de paramètres.

### Les paramètres des procédures stockées

Presque 8 000 procédures ont des paramètres.

### Distribution du nombre de paramètres des procédures stockées

Il faudrait vérifier celles ayant plus de 50 paramètres ainsi que le groupe de procédures comprises dans le pic de 40 paramètres.

---

## Les vues

Plus de 250 vues sont présentes. La moitié dédiée à Great Plains et l'autre moitié pour HydroSoft. La plupart sont "scalaires" retournant une seule valeur. Il faudrait profiler la charge de données sur les vues dans l'environnement de production pour connaître l'empreinte en mémoire et le ratio de lecture/ écriture sur disque.

## Les indexes

Les tables ayant moins de 5 index par table doivent être vérifiées.

## Les clefs principales

Il resterait à valider 20% des tables ayant moins de 2 clefs par table.

## Les clefs étrangères

Il resterait à valider les tables n'ayant pas de 2 clefs vers les principales tables.

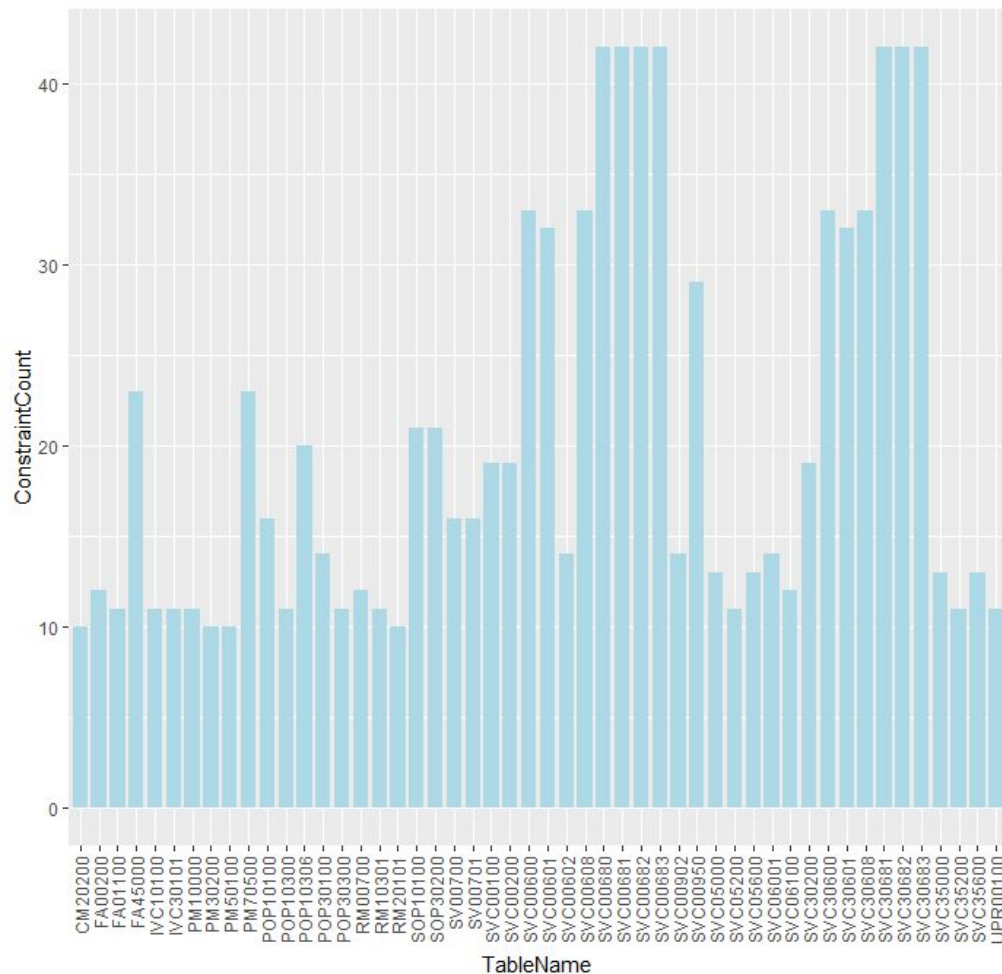
## Les tables

Ayant fait le survol des autres composantes, des indices clairs existent pour continuer l'exploration des tables et de les modifier. il faudra refaire l'exercice sur l'environnement en production pour la cueillette de données sur le comportement de la base de données.

Il faut développer la nouvelle application dans un nouveau schéma, valider les contraintes, investiguer les tables ayant de 100-1000 enregistrements, valider les accès disque des tables appelées le plus souvent par les procédures, par les fonctions et par les vues.

Il faudrait voir la possibilité d'indexer les colonnes les plus demandées, en priorité les tables avec peu de clefs principales. Il va falloir aussi ajouter de clés étrangères pour alléger les vues, procédures et fonctions. Il faudrait commencer par les tables utilisées par le code et par les workflow des scénarios testés en novembre.

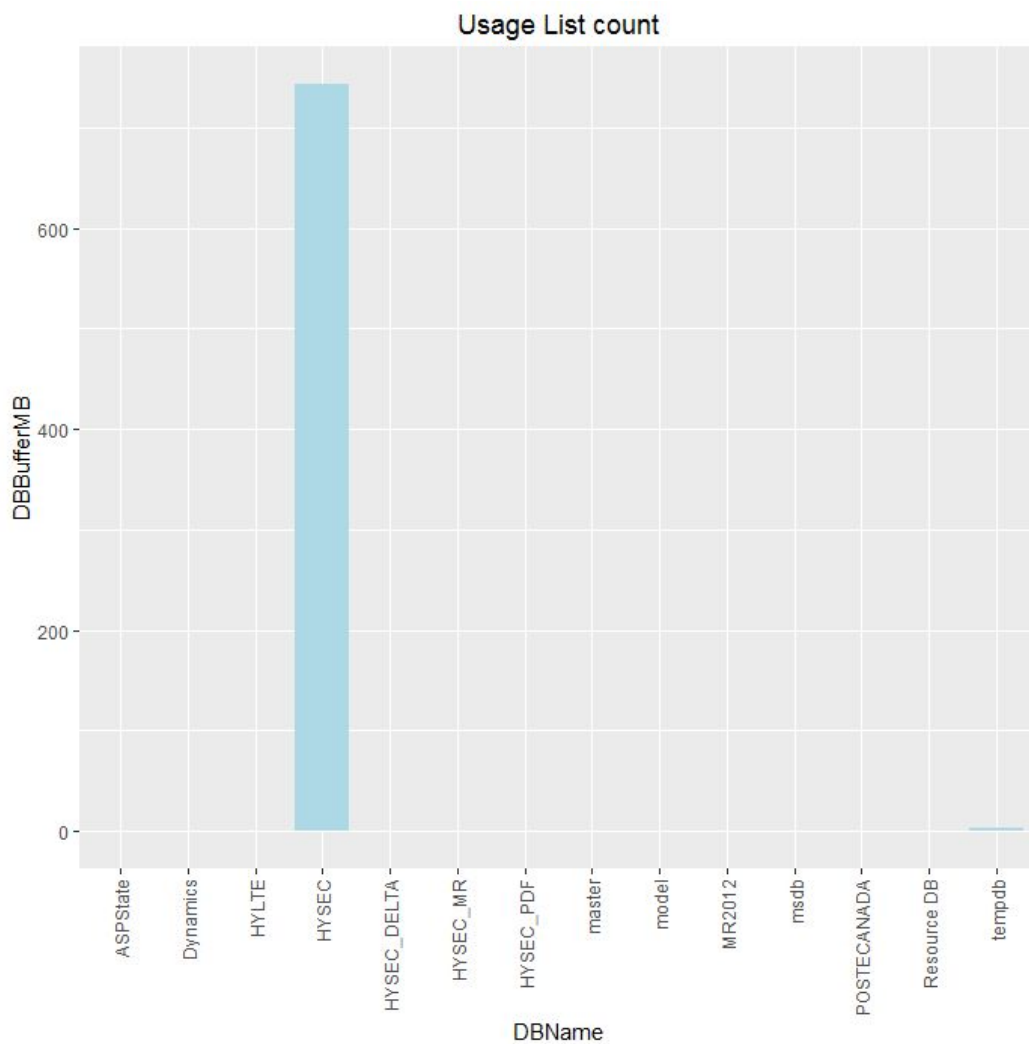
## Annexes



## Figure 1.- Analyse de base de données, les contraintes

Liste de tables sensible, ratio du nombre de contraintes pour chaque table de la base de données HySec.

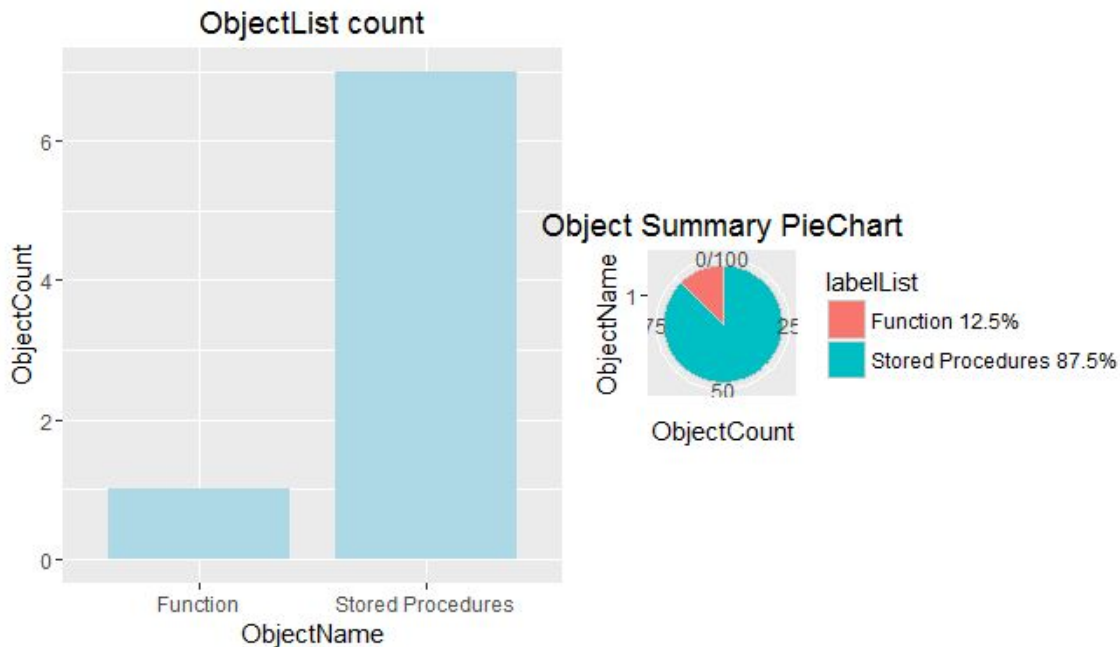
Veuillez noter que la plupart des tables ayant au delà de 10 contraintes appartiennent à Great Plains, il faut investiguer celle qui n'en appartiennent pas pour l'optimisation.



**Figure 2.- Analyse de base de données, la mémoire**

Figure montrant la consommation de mémoire de la liste de bases de données dans le système.

Veuillez noter que sur un environnement sans utilisation ni trafic comme celui de développement, la base de données de l'application HySec nécessite environ 750 Mo de mémoire de base.

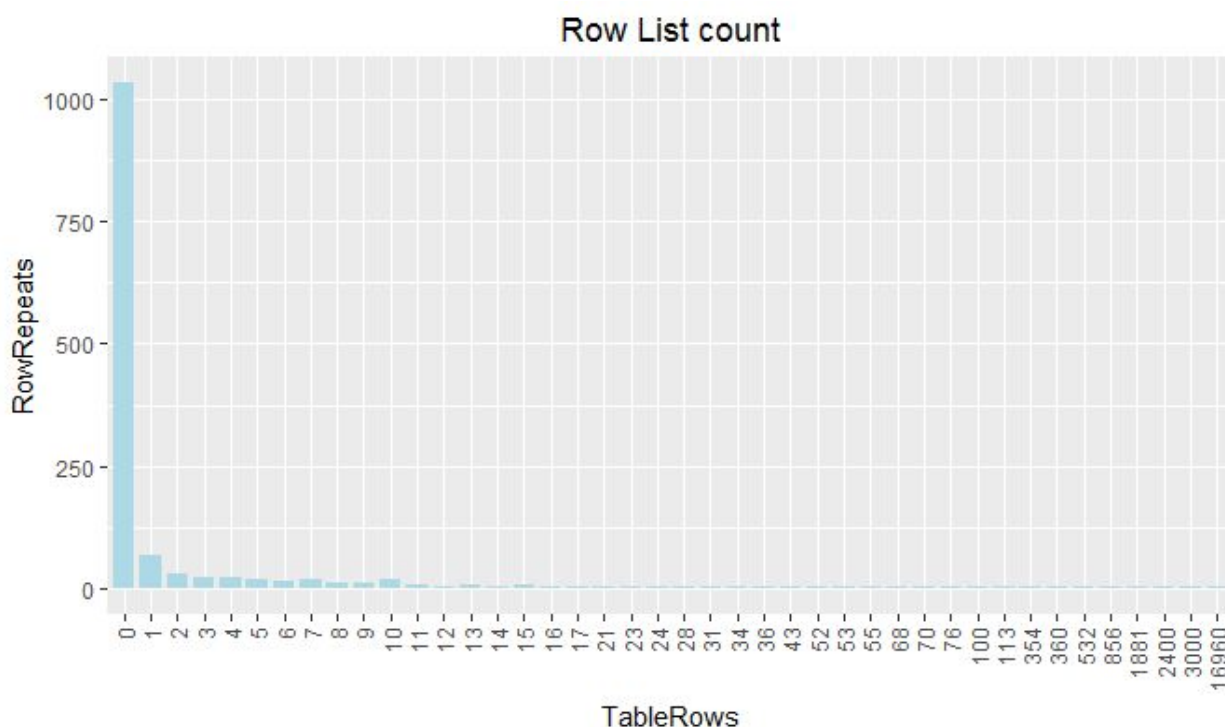


**Figure 3.- Analyse de base de données, les objets**

Liste de l'utilisation d'objets, dénombrement des différents types de fonction en un seul groupe versus le dénombrement du nombre de procédures stockées de la base de données HySec.

Veuillez noter que le nombre de procédures stockées est supérieur de 7 fois celui des fonctions. La performance d'exécution d'une procédure stockée étant supérieur à celui d'une fonction, les transactions globales des objets sont plus rapides par la prédominance de procédures stockées.





**Figure 4.- Analyse de base de données, les enregistrements des Tables**

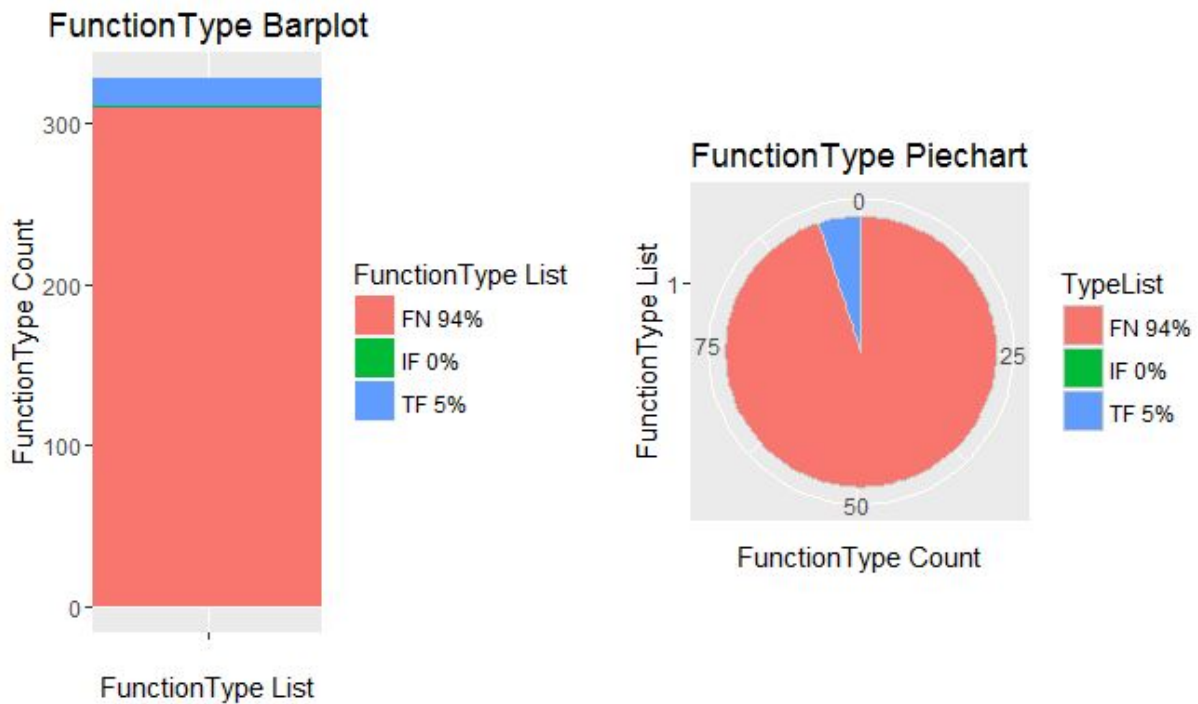
Tri décroissant sur le nombre d'enregistrements par table dans la base de données HySec.

Veuillez noter que 1000 tables n'ont pas de données, donc des tables non utilisées, il faut nettoyer celles qui n'appartiennent pas a Great Plains.

Une centaine ont une donnée, une trentaine ont deux données, une vingtaine ont de trois à dix données, certainement sont des tables de configuration.

Les tables ayant de 100 à 856 données devraient être liées aux transactions et fonctionnalités les plus utilisées.

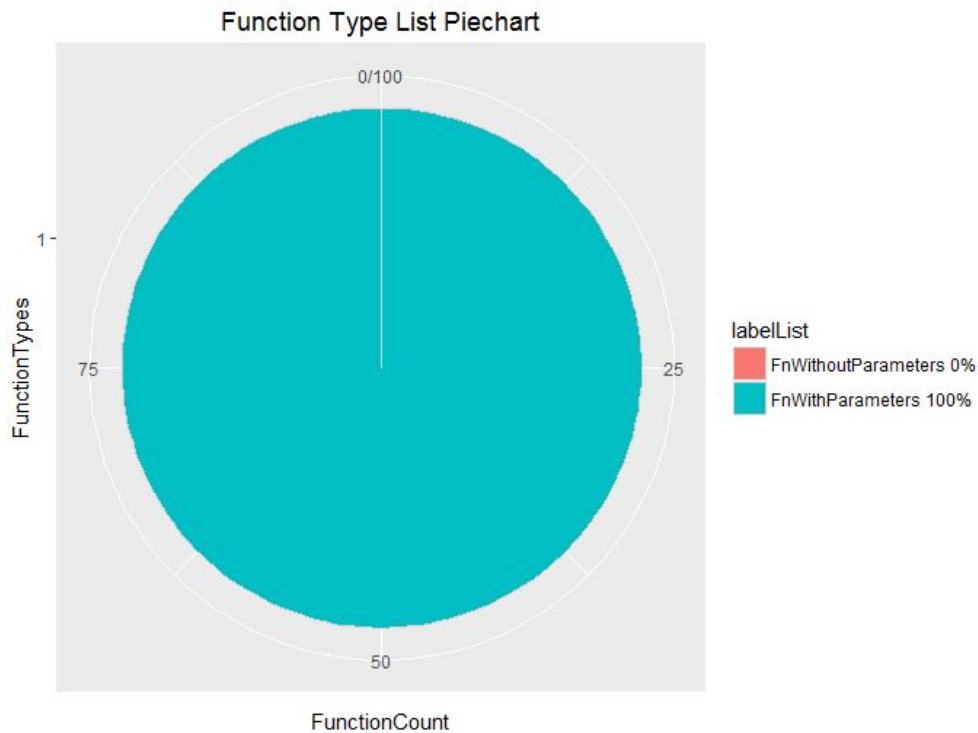
Les tables ayant de 1881 à 16960 données devraient être liées aux transactions et fonctionnalités centrales.



**Figure 5.- Analyse de base de données, les fonctions**

Décompte des trois types de fonctions dans la base de données HySec, prédominance de fonctions scalaires de la base de données HySec.

Veuillez noter qu'une seule est plus rapide, de type *inline*, que 5% sont de type *table valued*, plus lentes, et que le reste sont de type *scalar*. Les fonctions de la base de données HySec sont optimisées à 95%.

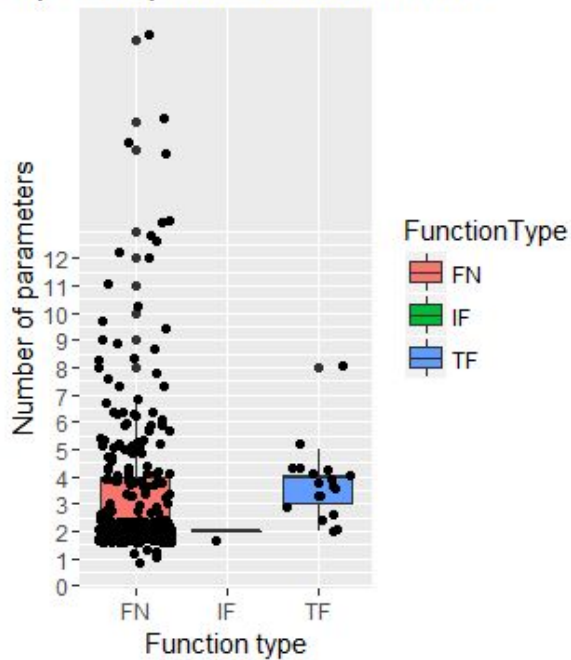


**Figure 6a.- Analyse de base de données, les paramètres des fonctions**

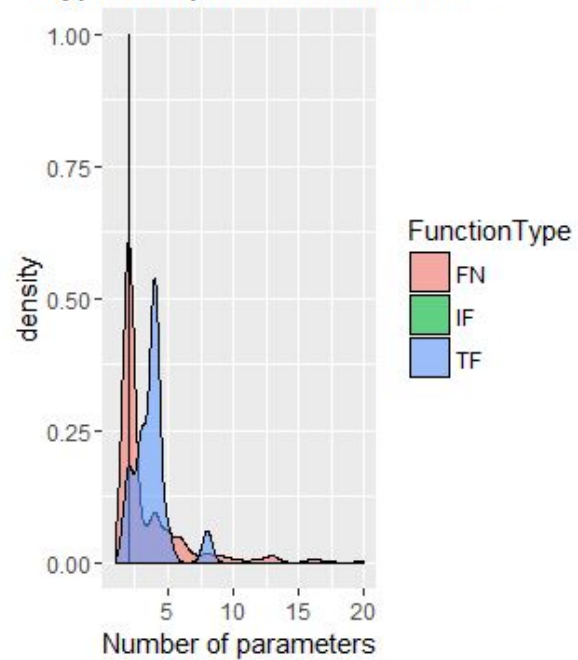
Ratio du nombre de fonctions ayant des paramètres versus le nombre de fonctions n'ayant pas de paramètres de la base de données HySec.

Veuillez noter que tous les différents types de fonctions utilisent des paramètres. Si elles sont appelées souvent, elles resteront longtemps en mémoire donc seront plus rapides. L'environnement de développement n'est pas utilisé, nous ne pouvons pas extrapoler pour l'environnement de production.

Boxplot for parameter distribution



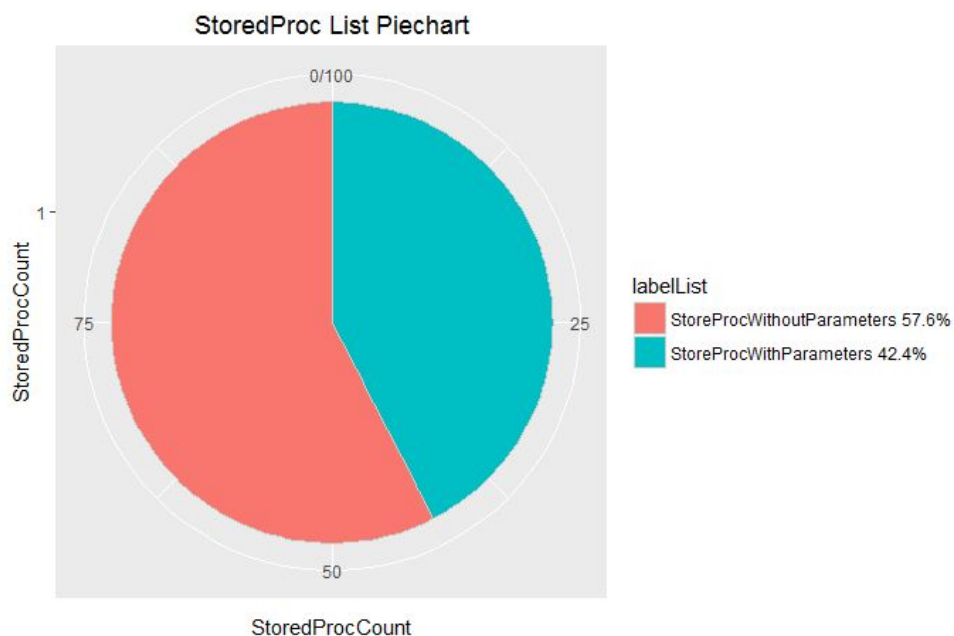
Densityplot for parameter distribution



**Figure 6b.- Analyse de base de données, la distribution des paramètres des fonctions**

Distribution du nombre de paramètres par type de fonction de la base de données HySec.

Veuillez noter que, dans les 5% de fonctions plus lentes *table valued*, il y en a un certain nombre ayant plus de 4 paramètres à valider. Il est intéressant de noter que pour les fonctions de type *scalar*, la moitié ont plus de 3 paramètres, il faudrait les valider.

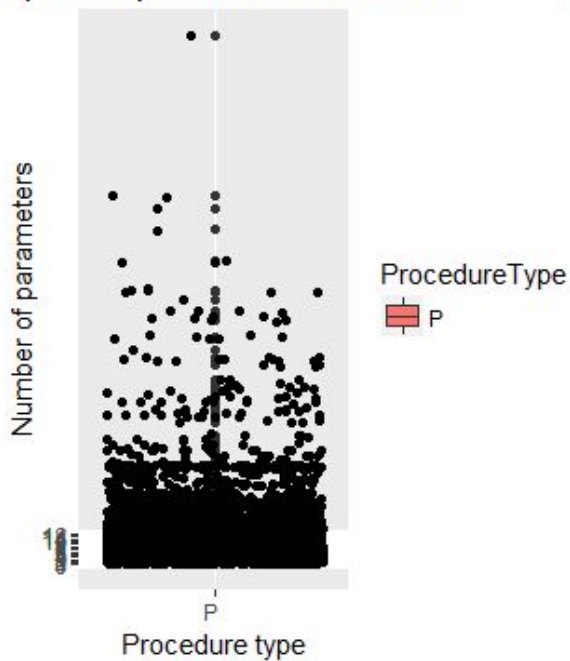


**Figure 7a.- Analyse de base de données, les paramètres de procédures stockées**

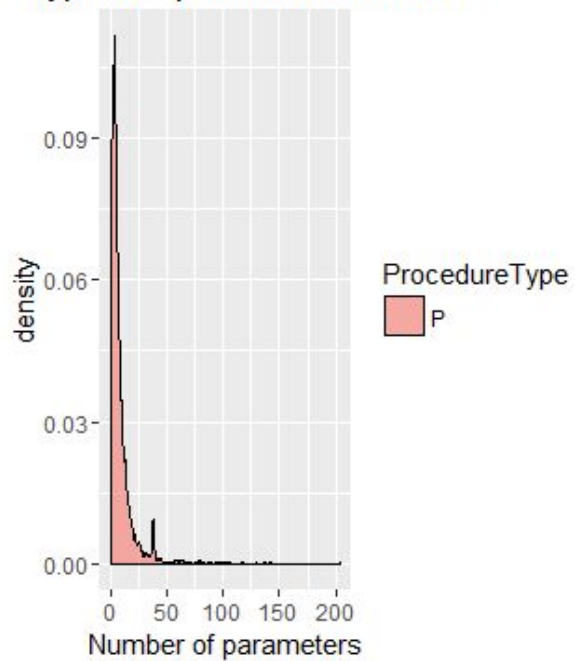
Ratio de 45/55 sur le nombre de procédures stockées ayant paramètres versus le nombre de procédures stockées n'ayant pas de paramètres de la base de données HySec.

Veuillez noter que presque la moitié d'entre elle ont des paramètres, malgré la rapidité des procédures stockées, il faudrait valider celles ayant des paramètres.

oxplot for parameter distribution



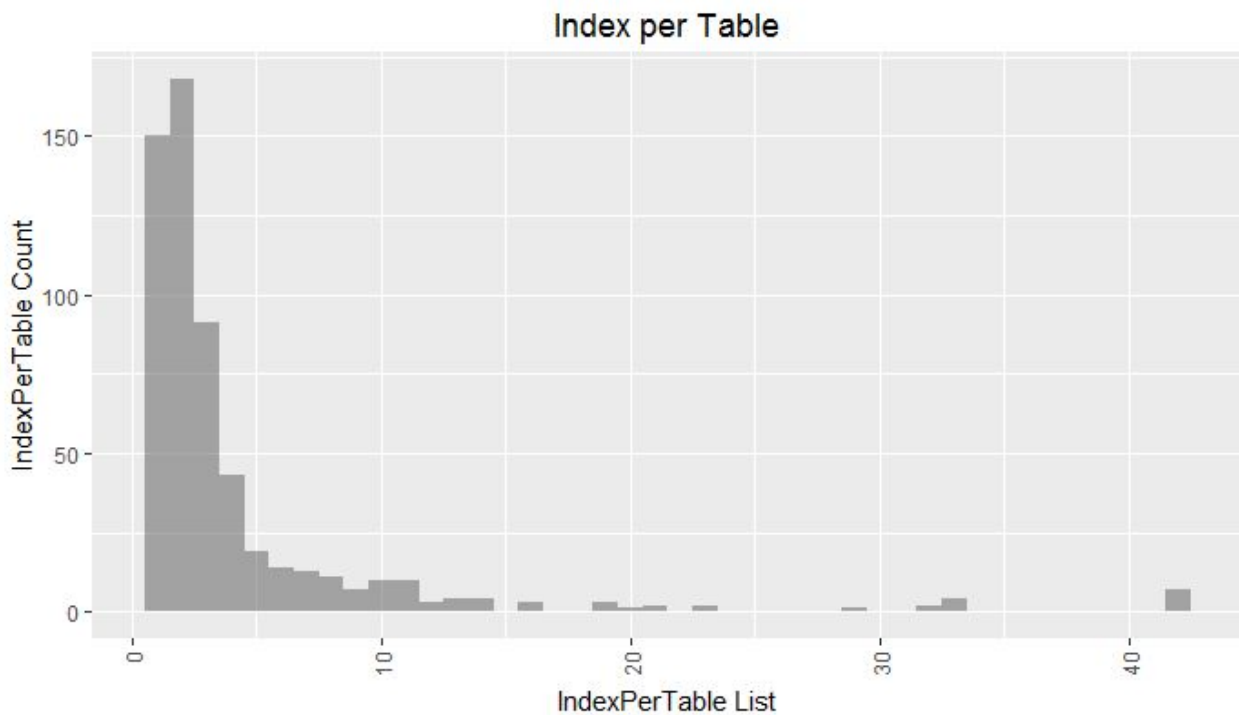
Densityplot for parameter distribution



**Figure 7b.- Analyse de base de données, la distribution des paramètres des procédures stockées**

Le nombre de procédures stockées ayant paramètres dépasse 20 avec un pic à 40 de la base de données HySec.

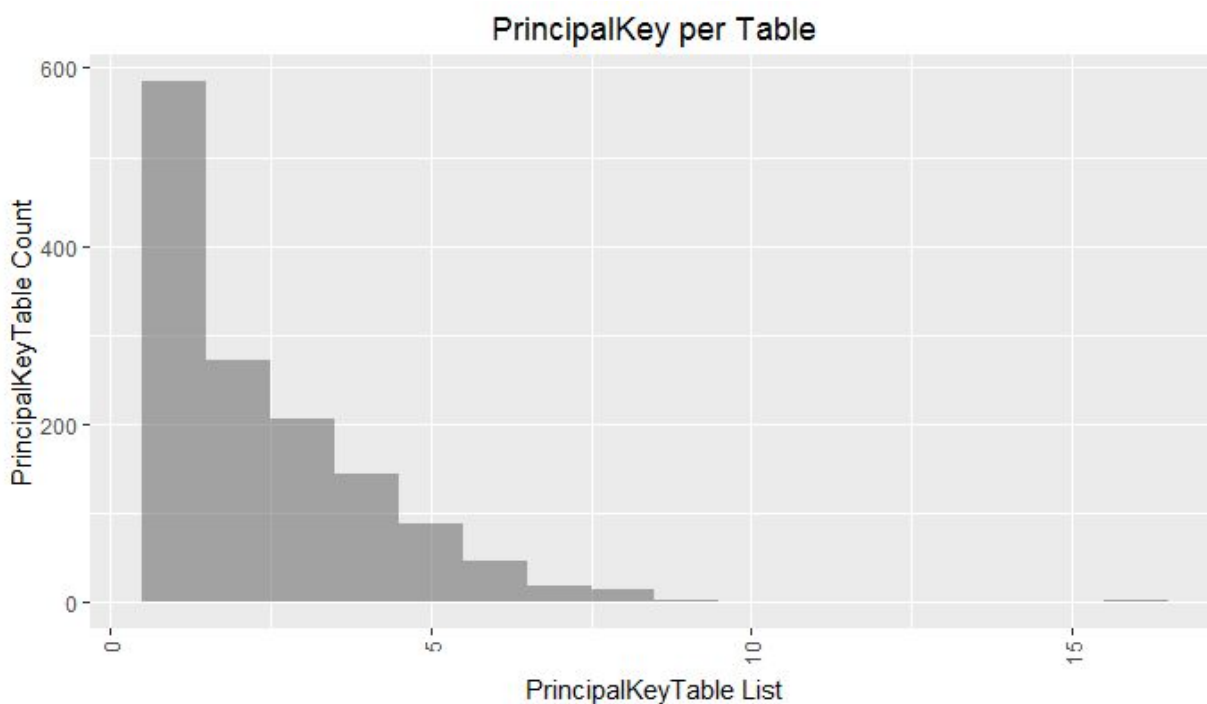
Veuillez noter qu'il faut valider les procédures stockées ayant plus de 20 paramètres (20%) pour en faire l'optimisation. Il n'est pas normal d'en avoir avec 200 paramètres.



**Figure 8.- Analyse de base de données, les indexes**

Histogramme du décompte du nombre d'index pour chaque table de la base de données HySec.

Veuillez noter que certaines tables ont au delà de 100 indexes, certaines tables ont en deçà de 10 indexes, il faut les valider pour en optimiser.



**Figure 9.- Analyse de base de données, les clefs principales**

Histogramme du décompte du nombre de clefs principales pour chaque table, 20% ont moins de 3 clefs de la base de données HySec.

Veuillez noter que certaines tables ont au delà de 200 indexes, certaines tables ont en deçà de 20 indexes, il faut les valider pour en optimiser.



**Tableau 1.- Analyse de base de données, les clefs étrangères**

TableName	ColumnName	ForeignKey	ForeignKeyID	ReferenceTableName	ReferenceColumnName
GMP_AUTO_SCRIPT	ScriptGroupId	FK_ScriptGroup	1513041538	HS_ScriptGroup	ID
GMP_ETAT_COMPTE_Detail	SID	FK_GMP_ETAT_CO__SID__5472C43C	1416807484	GMP_ETAT_COMPTE_Sommaire	SID
HS_Client_HS_Portail Message	HS_PortailMessageID	FK_HS_Client_HS_PortailMessage_HS_PortailMessage	1912809251	HS_PortailMessage	ID
HS_Client_HS_Portail Questions	HS_PortailQuestionsID	FK_HS_Client_HS_PortailQuestions_HS_PortailQuestions	2008809593	HS_PortailQuestions	ID
HS_COMMUNICATION_USAGERS	UsagerID	FK_HS_COMMUNICATION_USAGERS_USAGERS	347068593	GMP_USAGERS	UsagerID
HS_PortailModifLog	HS_PortailTypeModifID	FK_HS_PortailModifLog_HS_PortailTypeModif	1688808453	HS_PortailTypeModif	ID
RM00105	CPRCSTNM	RM_NationalAccounts_MSTR_FKC	1129053987	RM00101	CUSTNMBR

Tableau indiquant la liste de clefs étrangères de la base de données HySec.

Veuillez noter le petit nombre de clés, uniquement 7 sont répertoriées.