Manuel d’Exploitation et de recette de l’export des décisions

| Date | Version | Auteur | Objet de la mise à jour |
| --- | --- | --- | --- |
| 02/02/2018 | 1.0 | Anthony | Création |
| 08/03/2018 | 1.1 | Anthony | Mises à jour mineures, et ajout d’informations sur l’import dans SP |
| 13/06/2018 | 1.2 | Anthony | Mises à jour mineures de la doc pour ajouter des information sur les tâches planifiées, et pour détailler la modification faite pour résoudre bug sur les pdf texte |
| 15/04/2019 | 1.3 | Anthony | Maj d’une capture d’écran sur la tache planifiée d’import |
| 15/04/2019 | 1.4 | Anthony | Petite modification sur les tables de paramétrages |
| 15/04/2019 | 1.5 | Anthony | Ajout d’une information sur Active Directory |
| 16/04/2019 | 1.6 | Anthony | Ajout d’infos sur les différentes BDD de log |

# Sommaire

Table des matières

[1 Sommaire 2](#_Toc6325449)

[2 EXPLOITATION de l’export des décisions de la GED et du lancement de l’import des décisions dans SP 3](#_Toc6325450)

[2.1 Fonctionnement général 3](#_Toc6325451)

[2.2 Détails sur les tâches planifiées 4](#_Toc6325452)

[2.3 Dans le cas du traitement de masse 6](#_Toc6325453)

[2.4 En cas d’erreur technique 7](#_Toc6325454)

[2.4.1 Pour l’export 7](#_Toc6325455)

[2.4.2 Pour l’import 8](#_Toc6325456)

[2.5 Logs de l’export 8](#_Toc6325457)

[2.6 Autres tables de paramétrages 9](#_Toc6325458)

[2.7 Points particuliers 11](#_Toc6325459)

[3 RECETTE de l’export des décisions 12](#_Toc6325460)

[3.1 Recette des PDFs pour ELEC ancienne et nouvelle version 12](#_Toc6325461)

[3.1.1 Pour le document PDF issu de la GED (PDF Image) 12](#_Toc6325462)

[3.1.2 Pour le document PDF issu de l’application ELECTION (PDF Texte) 12](#_Toc6325463)

[3.2 Recette du fichier XML 12](#_Toc6325464)

[3.2.1 Pour ELEC nouvelle version 12](#_Toc6325465)

[3.2.2 Pour ELEC ancienne version 15](#_Toc6325466)

# EXPLOITATION de l’export des décisions de la GED et du lancement de l’import des décisions dans SP

## Fonctionnement général

L’exécution de **l’export** des décisions en mode manuel nécessite de lancer simplement le fichier z\_batch\_export\_shp\_quotidien.bat (dans le dossier D:\DEV\dev\_ia\2014\en recette et dans le dossier D:\Applic\_Elec\ en PROD).

En revanche en traitement planifié (exécuté tous les soirs à 21h sept jours sur sept) c’est un peu différent. La tâche planifiée va :

* Tout d’abord exécuter le fichier php ‘X\_moulinette\_outil\_consultation\_sharepoint\_batch\_quotidien.php’ sans utiliser de fichier batch
* Puis exécuter le fichier batch ‘z\_batch\_controle\_export\_quotidien.bat’

Dans les deux modes de lancement, le programme exporte des décisions, puis (en cas de succès) il va notifier dans un fichier que l’export s’est bien passé, cela servira au traitement d’import.

A noter : le fichier z\_batch\_export\_sans\_import\_shp\_quotidien.bat a été créé pour ne faire que la partie export (en cas de besoin)

Dans les deux modes de lancement, le premier fichier appelé est le fichier : X\_moulinette\_outil\_consultation\_sharepoint\_batch\_quotidien.php.

Dans ce dernier, se trouve le traitement d’export, ou plutôt l’appel au fichier principal d’export (à savoir ‘FONCTIONS\_decision\_texte\_depot\_EDOUARD\_07112017.php’) :

Cette ligne permet d’instancier la classe Outil\_consultation\_sp qui constitue à elle seule le fichier ‘FONCTIONS\_decision\_texte\_depot\_EDOUARD\_07112017.php’.

$Outil\_consultation\_sp = new Outil\_consultation\_sp();

Et cette ligne permet d’appeler le traitement d’export quotidien (ou de masse mais normalement on ne l’utilise plus) :

$texte\_mail = $Outil\_consultation\_sp->Traitement\_masse();  
//$texte\_mail = $Outil\_consultation\_sp->Traitement\_quotidien();

Ensuite, après l’export, le fichier X\_moulinette\_outil\_consultation\_sharepoint\_batch\_quotidien.php va notifier par mail l’équipe informatique pour l’informer que l’export s’est bien passé.

Puis le programme va, en cas de succès, notifier ‘OK’ dans un fichier ComExportImport.txt. Ce fichier sera ensuite copié sur la machine qui héberge la solution d’import, c’est à dire 192.168.6.25 dans le dossier C:\Import\ pour la recette, et sur 192.168.6.23 dans le dossier C:\ImportDecisions\ pour la PROD.

Ensuite c’est **l’import**, planifié pour tourner à 2h du matin tous les jours sept jours sur sept, qui va se lancer. Le fichier appelé est z\_batch\_import.bat.

Ce fichier va vérifier qu’il y a bien le fichier ComExportImport.txt avec OK dedans. Si oui, il lance l’import en exécutant ce fichier développé par Infeeny : ImportDecisions.ps1. Lui-même communique avec les fichiers Populate-Items.ps1 et Utils.ps1 et LogFiles.xml. C’est le package fourni par Infeeny.

* Si l’import s’est bien passé, le batch supprime sur le réseau les fichiers pdf xml et csv qui ont été importés (le fichier de log reste pour laisser quelques informations même en cas de succès de l’import). Ensuite le batch supprime le fichier ComExportImport.txt, et envoie un mail (en appelant EnvoiMailOkImport.ps1) pour notifier l’équipe informatique que l’import s’est bien passé.
* Si l’import a échoué, le batch écrit *ERREUR LORS DE L IMPORT* dans le fichier SiErreurImport.txt, et le traitement s’arrête sans supprimer les fichiers pdf xml et csv bien évidemment et sans envoyer de mail.

A noter pour l’export : Les fichiers z\_batch\_export\_shp.bat, X\_moulinette\_outil\_consultation\_sharepoint.php et X\_moulinette\_outil\_consultation\_sharepoint\_batch.php ont été utilisés (et donc ne sont plus utilisés) pour l’export de décisions à partir d’un fichier csv.

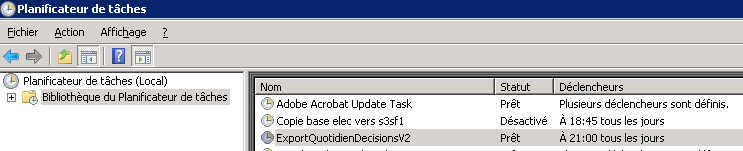
A noter pour l’export :

const *PROCESS\_TYPE* = ''; // Activation du traitement csv (en mettant 'csv') OU Désactivation du traitement csv (en mettant '')  
private $PDF\_FILES = true; // à commenter pour ne pas créer de fichier PDFs

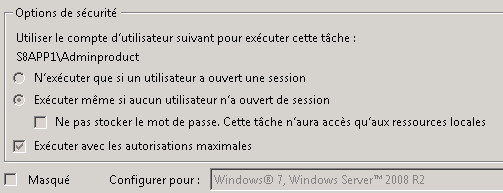
Dans le fichier ‘FONCTIONS\_decision\_texte\_depot\_EDOUARD\_07112017.php’, la première variable doit être renseigner à ‘csv’ si on a lancé le traitement d’export à partir d’un fichier csv (cf. z\_batch\_export\_shp.bat). La deuxième variable permet de ne pas créer les fichiers PDF (en cas de besoin).

## Détails sur les tâches planifiées

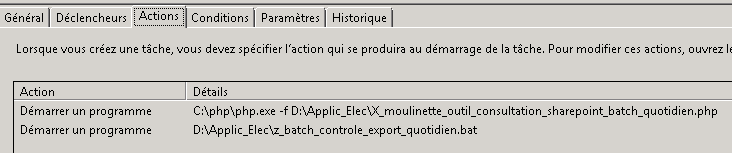
La première tâche planifiée, celle qui gère **l’export** tous les jours à 21h, se nomme ‘ExportQuotidienDecisionsV2’ :



Sur l’onglet Général, voici le paramétrage de cette tâche :

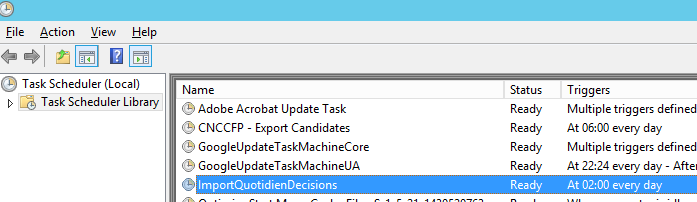


Sur l’onglet Actions, voici la configuration :

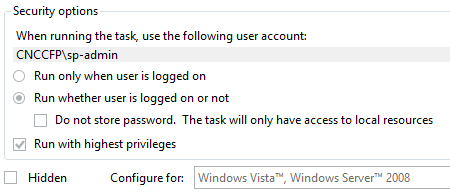


Le reste étant logique, les autres onglets ne sont pas détaillés.

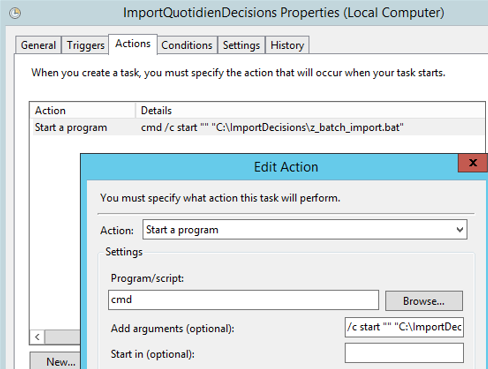
La deuxième tâche planifiée, celle qui gère **l’import** tous les jours à 2h du matin, se nomme ‘ImportQuotidienDecisions’ :



Pour l’onglet Général, voici le paramétrage :



Pour l’onglet Actions, voici la configuration :



Les autres onglets sont paramétrés dans la même logique que la tache planifiée d’export.

## Dans le cas du traitement de masse

L’export de masse est très long surtout pour la nouvelle application ELEC. L’export sur 2015 dure plus de 2 jours à titre d’exemple.

Donc généralement on filtre par année, voir par morceau d’année.

Ce filtrage s’effectue en dur dans le code PHP, dans le fichier ‘FONCTIONS\_decision\_texte\_depot\_EDOUARD\_07112017.php’, dans la fonction Traitement\_masse() :

$this->document\_param['id\_lettre']="";  
$this->document\_param['type\_decision']="";  
$this->document\_param['annee']="2017";  
$this->document\_param['DATE\_SCAN\_SQL']= "";  
$this->document\_param['DATE\_SCAN\_DEBUT\_SQL']="20171101";  
$this->document\_param['DATE\_SCAN\_FIN\_SQL']="20171231";  
$this->document\_param['type\_process']="";  
$this->document\_param['dossier']="";  
$this->document\_param['chk\_grief\_financier']="";  
$this->document\_param['id\_candidat'] = "";

Dans cet exemple on filtre sur l’année 2017 sur les deux derniers mois de l’année.

Attention au filtrage sur un candidat (cf. $this->document\_param['id\_candidat'), car la première requête sur la GED risque de ne rien ramener et on n’aura rien exportée. En effet, un certain nombre de données ne sont pas renseignées dans les tables de la GED comme le NUMCAND.

En conséquence si on a une erreur sur un document, et qu’on veut le rejouer en « pas à pas » (sous phpstorm par exemple) en local sur son poste, le filtrage sur le NUMCAND risque de ne pas fonctionner. Il faut ruser avec des filtrages sur les dates, ou forcer des valeurs en dur de manière temporaire. Attention au points d’arrêts conditionnels sous phpstorm, les valeurs sont « faussées ».

Aussi quand on debug en local, il faut avoir tout le dossier complet (plusieurs Go) bien évidemment. Mais surtout c’est la création des pdf qui est très difficile à faire fonctionner (cf. Imagick). D’où une condition sur « ENVIRONNEMENT != LOCAL » dans le code php avant de créer le pdf. Mais ce n’est pas indispensable de créer les pdf en local pour tester en pas à pas, tout dépend de ce que l’on veut tester.

## En cas d’erreur technique

### Pour l’export

Il n’y a pas de mécanisme de reprise après incident. Si l’export quotidien a échoué, et n’est pas allé au bout pour une raison technique, il faut :

1. S’assurer que la ressource qui était indisponible est à nouveau disponible (serveur ou BDD)
2. Relancer le traitement à la main. Si le traitement échoue une veille de jour chômé, il sera de toute façon relancé 24h plus tard de manière automatique (cf. planificateur des tâches), et il sera capable d’analyser en base qu’il faut exporter 2 jours ou plus.

A noter : Comme le programme PHP supprime tous les fichiers de tous les dossiers du DDRT (sauf pour le dossier ‘A\_Anonymiser’), et ce dès le début du traitement, il n’y aura pas de problème de doublon dans la fabrication des fichiers pdf et xml et csv si le programme a échoué la veille.

Pour rappel : L’équipe informatique reçoit un mail en cas de succès de l’export. Donc si on n’a pas reçu d’email en arrivant le matin, il faut investiguer.

### Pour l’import

Il n’y a pas de mécanisme de reprise après incident. Si l’import (quotidien) a échoué, et n’est pas allé au bout pour une raison technique, il faut :

1. S’assurer que la ressource qui était indisponible est à nouveau disponible (serveur ou BDD)
2. Relancer le traitement à la main dès J (J étant le jour de l’import à 2h du matin) car sinon l’export de J (qui copie les fichiers pdf et xml et csv dans un autre dossier pour l’import) va écraser les fichiers (enfin les xml et csv) et il y aura une perte de données

A noter : Le traitement d’import gère les doublons dans SP, c’est-à-dire que s’il reçoit un document déjà intégré, il l’ignore sans arrêter le traitement, et un warning est logué dans le fichier en \*.log

Pour rappel : L’équipe informatique reçoit un deuxième mail en cas de succès de l’import. Donc si on n’a pas reçu cet email en arrivant le matin, il faut investiguer et résoudre le problème dans la journée.

En cas d’erreur liée à une décision rattachée à un chargé de mission qui n’est plus dans les murs : L’import jettera une erreur dans ce cas. Pour y remédier, il faut chercher le CM qui reste dans un sous-dossier de l’AD. Ensuite, si on le remonte d’un niveau dans l’AD, on peut relancer l’import, qui sera en succès.

## Logs de l’export

La table z\_depot\_traitement contient les informations globales sur les traitements d’export. On peut voir si le traitement en cours est terminé, et si oui son statut, et les 13 compteurs.

Il y a 3 types d’erreurs : techniques, fonctionnelles, warnings :

* Les erreurs techniques sont logiquement suivi d’un arrêt du programme, sauf certaines où on va loguer une erreur technique et continuer l’exécution du programme en espérant que le problème de ressource (serveur, ou BDD) peut se résoudre et donc ne pas bloquer tout le traitement
* Les erreurs fonctionnelles, ne sont pas bloquantes et constitue souvent une incohérence des données de production dans BDD. Il peut y avoir plusieurs erreurs de ce type par document.
* Le traitement peut générer aussi quelques warnings (erreurs peu importante) essentiellement pour les considérants manquants.

La table z\_depot\_traitement\_message contient le détail des 3 types d’erreur. Elle contient donc évidemment une clé étrangère sur la table z\_depot\_traitement.

Il faut noter que le programme d’export logue dans une base différente suivant si on est en RECETTE, en PRE-PROD, ou en PROD. Ceci grâce à la lecture du fichier ‘ENV\_PourPrgmExport.txt’ qui est différent dans chaque environnent.

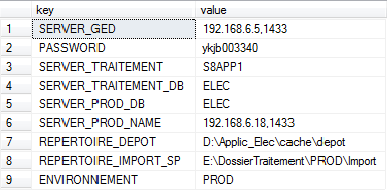
La base sera :

* DEVEA en DEV
* DEVTJ en PREPROD
* S8APP1 en PROD

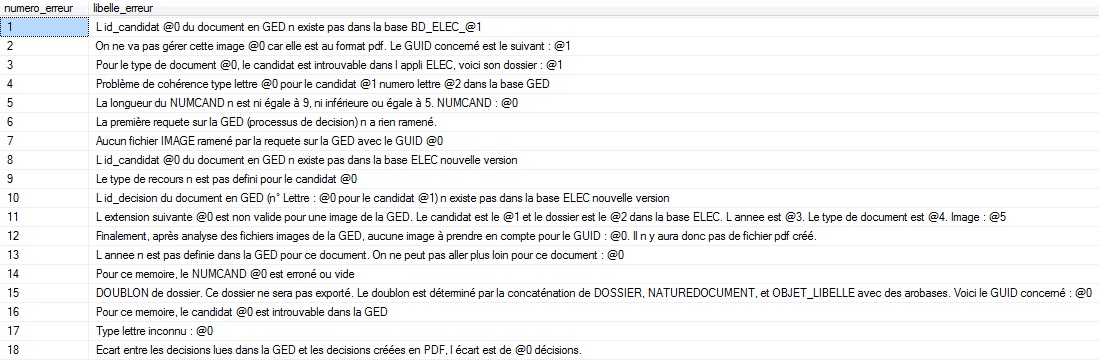
## Autres tables de paramétrages

Il y a 3 autres tables dédiées au traitement d’export, voici leur utilité :

z\_depot\_parametrages contient des key/value qui sont toutes lues au début du traitement. Cela permet de ne pas toucher au code PHP en, cas de modification d’un serveur, d’un mot de pass, d’un répertoire. Cela permet surtout de changer facilement les valeurs qui évoluent suivant les environnements (LOCAL ou DEV, RECETTE, PROD).



z\_param\_erreurs\_fonctionnelles héberge tous les codes d’erreurs fonctionnelles. Attention dans la colonne libelle\_erreur se trouve des @0, @1, @2…etc, qui permettront au code PHP de savoir où placer les valeurs des variables à ajouter au texte statique.



z\_depot\_traitment\_param a servi à l’export à partir d’un fichier csv.



## Points particuliers

* Attention, dans le code PHP, la création des PDF ne se lancera que si on n’est pas en LOCAL, car Imagick ne fonctionnait pas sur mon poste :

if($this->environnement !== 'LOCAL')

* La métadonnée qui gère le critère ‘remboursable’ diffère suivant les environnements :

|  |  |  |  |
| --- | --- | --- | --- |
|  | LOCAL | RECETTE | PROD |
| Décisions initiales | DCSRemboursable | DCSRemboursable | DCSRemboursable |
| Décisions recours | DCSRemboursable | Remboursable | DCSRemboursable |
| Décisions modificatives | DCSRemboursable | Remboursable | DCSRemboursable |
| Jugements | DCSRemboursable | Remboursable | DCSRemboursable |
| Mémoires | DCSRemboursable | Remboursable | DCSRemboursable |
| Requêtes | DCSRemboursable | Remboursable | DCSRemboursable |

Mais le code PHP s’adapte aux différents environnements et change le nom de la balise, pour que l’import fonctionne.

* J’ai dû modifié le powershell d’Infeeny ImportDecisions.ps1 pour changer des chemins relatifs en absolu, et surtout pour gérer le problème d’accès à [\\S8APP1\Import\](file://S8APP1/Import/) résolu en montant un lecteur réseau dans le powershell (net use I: \\S8APP1\Import AdminProd1 /user:CNCCFP\adminproduct /PERSISTENT:YES)

# RECETTE de l’export des décisions

## Recette des PDFs pour ELEC ancienne et nouvelle version

Pour chaque pdf fabriqué, il faut effectuer les vérifications suivantes :

### Pour le document PDF issu de la GED (PDF Image)

* Vérifier que ces informations sont les mêmes dans le titre du document et en BDD : NUMCAND, N° de lettre, N° de dossier.
* Vérifier en BDD que la date de décision se trouvant en première page du document est correcte

Pour ces dernières vérifications, voici une requête SQL qui questionne la BDD de la GED :

SELECT GUID, FD\_E7787B05 AS ANNEE, FD\_09A9C5FE AS NUMCAND, FD\_3A5E7E76 AS NOLETTRE, FD\_CD52E931 AS CHRONO, CONVERT(VARCHAR,CAST(FD\_Documents.CreatedOn AS DATETIME),112) AS DATE\_CREATEDON, PageNo AS PAGE

FROM [FD\_C66DBCEA].[dbo].[FD\_Documents]

LEFT JOIN [FD\_C66DBCEA].[dbo].[FD\_Images] ON (GUID = DocGUID)

WHERE (FD\_C6004355 = '**N**' OR FD\_C6004355= '**X**')

AND FD\_E7787B05= **'2016'**

AND ActRevision = RevNo and Deleted <> '1'

AND PageNo=1

AND FD\_09A9C5FE = **'201600204'**

Evidement la requête est à adapter suivant le type de document, l’année, le NUMCAND (cf. champs en gras dans le code SQL).

### Pour le document PDF issu de l’application ELECTION (PDF Texte)

* Vérifier l’iso conformité du document avec la version scannée : nom du document et contenu du document (Cela ne concerne que les décisions)

## Recette du fichier XML

### Pour ELEC nouvelle version

La requête ci-dessous permet de vérifier sur la BDD de l’application ELEC un grand nombre de métadonnées :

SELECT

C.id\_candidat AS DCSCandidatId, C.nom\_cand AS DCSCandidatNom1, C.prenom\_cand AS DCSCandidatPrenom1,

(CASE WHEN C2.nom\_cand is null THEN '' ELSE C2.nom\_cand END) AS DCSCandidatNom2,

(CASE WHEN C2.prenom\_cand is null THEN '' ELSE C2.prenom\_cand END) AS DCSCandidatPrenom2,

S.id\_scrutin AS DCSScrutin, nom\_circons AS DCSCirconscription, S.id\_dpt AS DCSDepartement, D.id\_region AS DCSRegion,

E.libelle\_elec + ' ' + (CASE WHEN abrev\_type\_elec = 'G' THEN 'Générale' WHEN abrev\_type\_elec = 'P' THEN 'Partielle' ELSE 'Erreur' END) AS DCSElection,

(CASE WHEN abrev\_type\_elec = 'G' THEN 'Générale' WHEN abrev\_type\_elec = 'P' THEN 'Partielle' ELSE 'Erreur' END) AS DCSElectionType, C.annee AS DCSElectionAnnee, (CASE WHEN chk\_ctx = '0' THEN 'False' WHEN chk\_ctx = '1' THEN 'True' ELSE 'Erreur' END) AS DCSScrutinContentieux, concat(U\_RAP.prenom\_util, ' ', U\_RAP.nom\_util) AS DCSRapporteur, U\_RAP.id\_util AS DCSRapporteurId, concat(U\_RG.prenom\_util, ' ', U\_RG.nom\_util) AS DCSMembreCommission, concat(U\_CM.prenom\_util, ' ', U\_CM.nom\_util) AS DCSSuiviPar, C.etiquette AS DCSEtiquette, C.nuance AS DCSNuance,

(CASE WHEN PARTI.nom\_parti is null THEN '' ELSE PARTI.nom\_parti END) AS DCSParti,

(CASE WHEN PARTI.id\_parti is NULL THEN '0' ELSE 'A voir' END) AS DCSPartiId,

(CASE WHEN EXPERT.id\_expert = 0 THEN '' ELSE EXPERT.id\_expert END) AS DCSExpertComptableId,

(CASE WHEN EXPERT.nom\_cie\_expert is null THEN '' ELSE EXPERT.nom\_cie\_expert END) AS DCSExpertComptable,

(CASE WHEN C.chk\_elu = '0' THEN 'False' WHEN C.chk\_elu = '1' THEN 'True' ELSE 'Erreur' END) AS DCSElu,

(CASE WHEN C.chk\_sortant = '1' THEN 'True' ELSE 'False' END) AS DCSSortant,

C.id\_mandataire

FROM dbo.compte\_seance\_defaut AS SEANCE\_DEFAUT, dbo.candidat AS C

LEFT JOIN dbo.candidat AS C2 ON (C.id\_candidat\_associe = C2.id\_candidat)

LEFT JOIN dbo.parti AS PARTI ON (C.id\_parti = PARTI.id\_parti)

LEFT JOIN dbo.civilite AS CIVCAND ON (C.id\_civ\_cand = CIVCAND.id\_civilite)

LEFT JOIN dbo.compte AS CPTE ON (C.id\_candidat = CPTE.id\_compte)

LEFT JOIN dbo.rapport AS R ON (CPTE.id\_compte = R.id\_compte)

LEFT JOIN dbo.fiche\_synthese AS FS ON (CPTE.id\_compte = FS.id\_compte)

LEFT JOIN dbo.parametre AS PARAM ON (CPTE.id\_param = PARAM.id\_param)

LEFT JOIN dbo.utilisateur AS U\_RG ON (CPTE.id\_util\_rg = U\_RG.id\_util)

LEFT JOIN dbo.civilite AS CIVRG ON (U\_RG.id\_civ\_util = CIVRG.id\_civilite)

LEFT JOIN dbo.utilisateur AS U\_CM ON (CPTE.id\_util\_cm = U\_CM.id\_util)

LEFT JOIN dbo.civilite AS CIVCM ON (U\_CM.id\_civ\_util = CIVCM.id\_civilite)

LEFT JOIN dbo.utilisateur AS U\_CMA ON (CPTE.id\_util\_cma = U\_CMA.id\_util)

LEFT JOIN dbo.civilite AS CIVCMA ON (U\_CMA.id\_civ\_util = CIVCMA.id\_civilite)

LEFT JOIN dbo.groupe AS GR ON (CPTE.id\_groupe\_rapporteur = GR.id\_groupe)

LEFT JOIN dbo.utilisateur AS U\_RAP ON (GR.id\_util = U\_RAP.id\_util)

LEFT JOIN dbo.civilite AS CIVRAP ON (U\_RAP.id\_civ\_util = CIVRAP.id\_civilite)

LEFT JOIN dbo.scrutin AS S ON (C.id\_scrutin = S.id\_scrutin)

LEFT JOIN dbo.prefecture AS PREF ON (S.id\_dpt = PREF.id\_dpt)

LEFT JOIN dbo.prefecture\_type AS PREF\_TYPE ON (PREF.id\_type\_pref = PREF\_TYPE.id\_type\_pref)

LEFT JOIN dbo.juridiction AS JUR ON (S.id\_juridiction = JUR.id\_juridiction)

LEFT JOIN dbo.juridiction\_type AS JUR\_TYPE ON (JUR.id\_juridiction\_type = JUR\_TYPE.id\_juridiction\_type)

LEFT JOIN dbo.election AS E ON (S.id\_election = E.id\_election)

LEFT JOIN dbo.election\_ref AS E\_REF ON (E.id\_election\_ref = E\_REF.id\_election\_ref)

LEFT JOIN dbo.departement AS D ON (S.id\_dpt = D.id\_dpt)

LEFT JOIN dbo.region AS REG ON (D.id\_region = REG.id\_region)

LEFT JOIN dbo.mandataire AS MAND ON (C.id\_mandataire = MAND.id\_mandataire)

LEFT JOIN dbo.civilite AS CIVMAND ON (MAND.id\_civ\_mf = CIVMAND.id\_civilite)

LEFT JOIN dbo.expert AS EXPERT ON (C.id\_expert = EXPERT.id\_expert)

LEFT JOIN dbo.civilite AS CIVEXPERT ON (EXPERT.id\_civ\_expert = CIVEXPERT.id\_civilite)

LEFT JOIN dbo.suppleant AS SUPPL ON (C.id\_suppleant = SUPPL.id\_suppleant)

LEFT JOIN dbo.civilite AS CIVSUPPL ON (SUPPL.id\_civ\_suppleant = CIVSUPPL.id\_civilite)

LEFT JOIN dbo.suppleant AS SUPPL\_ASSOC ON (C2.id\_suppleant = SUPPL\_ASSOC.id\_suppleant)

LEFT JOIN dbo.civilite AS CIVSUPPL\_ASSOC ON (SUPPL\_ASSOC.id\_civ\_suppleant = CIVSUPPL\_ASSOC.id\_civilite)

WHERE C.id\_candidat = **'201511871'**

AND SEANCE\_DEFAUT.date\_debut\_defaut\_seance\_def < GETDATE()

AND (

SEANCE\_DEFAUT.date\_fin\_defaut\_seance\_def IS NULL OR SEANCE\_DEFAUT.date\_fin\_defaut\_seance\_def < GETDATE()

)

AND id\_compte\_seance = '4'

Cette autre requête ci-dessous permet de vérifier, toujours sur la BDD de l’application ELEC, d’autres métadonnées :

SELECT

date\_pass\_com AS DCSDate,

'' DCSAssociation,

(CASE WHEN chk\_signalee is null THEN 'False' ELSE 'True' END) AS DCSSignalee,

'False' AS DCSJurisprudence,

'False' AS DCSExJurisprudence,

(CASE WHEN chk\_doctrine = '1' THEN 'True' ELSE 'False' END) AS DCSDoctrine,

'False' AS DCSExDoctrine,

(CASE WHEN chk\_cas\_d\_espece is null THEN 'False' ELSE 'True' END) AS DCSCasEspece,

'' AS DCSMotsClesAuto,

'' AS DCSMotsClesIndex,

(CASE WHEN commentaire\_signalement is null THEN '' ELSE commentaire\_signalement END) AS DCSCommentaires,

'' AS DCSRemiseEnCause,

(CASE WHEN chk\_anonymiser is null THEN 'Non' ELSE 'Oui' END) AS DCSAnonymisation,

(CASE WHEN chk\_completer\_theme is null THEN 'False' ELSE 'True' END) AS DCSAReviser

FROM decision

WHERE id\_compte = **'201513373'**

AND id\_type\_decision = '3'

Cette dernière requête permet de recetter sur la BDD de la GED d’autres métadonnées :

----- POUR DECISIONS

select top 1

FD\_3A5E7E76 AS DCSLettreId,

[FD\_CD52E931] AS DCSGEDChrono,

(CASE WHEN CONVERT(VARCHAR,CAST(FD\_34CFCD24 AS DATETIME),110) is not null THEN FD\_34CFCD24 ELSE CONVERT(VARCHAR,CAST(FD\_Documents.CreatedOn AS DATETIME),110) END) AS DCSDateJour

from [FD\_C66DBCEA].[dbo].[FD\_Documents]

LEFT JOIN [FD\_C66DBCEA].[dbo].[FD\_Images] ON (GUID = DocGUID)

where (FD\_C6004355 = '**X**')

AND FD\_E7787B05= **'2016'**

AND ActRevision = RevNo and Deleted <> '1'

and PageNo=1

and FD\_09A9C5FE = **'201513373'**

order by [FD\_CD52E931] asc

----- POUR Process Mutualisés

select FD\_7EDEFFC4 AS DCSAffaireId,

[FD\_CD52E931] AS DCSGEDChrono,

(CASE WHEN CONVERT(VARCHAR,CAST(FD\_34CFCD24 AS DATETIME),110) is not null THEN FD\_34CFCD24 ELSE CONVERT(VARCHAR,CAST(FD\_Documents.CreatedOn AS DATETIME),110) END) AS DCSDateJour

from dbo.FD\_Documents

LEFT JOIN dbo.FD\_Images ON (GUID = DocGUID)

where substring(CONVERT(VARCHAR,CAST(FD\_Documents.CreatedOn AS DATETIME),112), 1, 4) = '2016' AND FD\_E8554E3E like **'Jugements/avis/décisions'** AND ActRevision = RevNo and FD\_5188A29D = 'Elections' and Deleted <> '1' and PageNo='1'

and FD\_09A9C5FE = **'201518796'**

ORDER BY GUID, FD\_3A5E7E76, FD\_34CFCD24, PageNo desc

### Pour ELEC ancienne version

Cette requête sur la BDD de l’application ELEC permet de ramener un grand nombre de métadonnées :

SELECT top 1

TCR.NumCand AS DCSCandidatId

, TCR.NomCand AS DCSCandidatNom1

, TCR.PrenomCand AS DCSCandidatPrenom1

, '' AS DCSCandidatNom2

, '' AS DCSCandidatPrenom2

, NomCircons AS DCSCirconscription

, DEP.CodeDepart AS DCSDepartement

, DEP.CodeRegion AS DCSRegion

, (case when TELEC.LibelleElections = 'cantonale' then 'departementale' else TELEC.LibelleElections end) + ' ' + TTELEC.LibelleTypeElection AS DCSElection

, TTELEC.LibelleTypeElection AS DCSElectionType

, TS.AnElection AS DCSElectionAnnee

, (CASE WHEN CtxInit = '1' THEN 'False' WHEN CtxInit = '0' THEN 'True' ELSE 'Erreur' END) AS DCSScrutinContentieux

, (RAP.PrenomRap + ' ' + RAP.NomRap) AS DCSRapporteur

, RAP.CodeRap as DCSRapporteurId

, (RAPG.PrenomRap + ' ' + RAPG.NomRap) AS DCSMembreCommission

, (FIL.prenom + ' ' + FIL.nom) AS DCSSuiviPar

, TCR.NoScrutin AS DCSLettreId

, CONVERT(VARCHAR,CAST(TS.DatePassComm AS DATETIME),110) as DCSDate

, '' As DCSEtiquette

, NuanceCand AS DCSNuance

, (CASE WHEN PP.nom\_parti is null THEN '' ELSE PP.nom\_parti END) AS DCSParti

, TCR.PartiPol As DCSPartiId

, (CASE WHEN ASS.NomAsso is not null THEN 'Association de financement électoral' WHEN ASS.NomAsso is null THEN 'Personne physique' ELSE 'Erreur' END) AS DCSMandataireType

, (CASE WHEN MAND.NomMand is null THEN '' ELSE MAND.NomMand END) AS DCSMandataire

, (CASE WHEN ASS.NomAsso is null THEN '' ELSE ASS.NomAsso END) AS DCSAssociation

, (CASE WHEN EXPR.NoExpert is null THEN '' WHEN EXPR.NoExpert = '0' THEN '' ELSE EXPR.NoExpert END) AS DCSExpertComptableId

, (CASE WHEN EXPR.RSExpert is null THEN '' ELSE EXPR.RSExpert END) AS DCSExpertComptable

, (CASE WHEN Elu = '1' THEN 'False' WHEN Elu = '0' THEN 'True' ELSE 'Erreur' END) AS DCSElu

, 'False' AS DCSSortant

, DECTYPE.AbregeTypeDecision AS DCSSens

, 'False' AS DCSSignalee

, 'False' AS DCSJurisprudence

, 'False' AS DCSExJurisprudence

, 'False' AS DCSDoctrine

, 'False' AS DCSExDoctrine

, 'False' AS DCSCasEspece

, '' AS DCSMotsClesIndex

, '' AS DCSCommentaires

, '' AS DCSRemiseEnCause

, 'Non' AS DCSAnonymisation

, 'False' AS DCSAReviser

FROM BD\_ELEC\_2014.dbo.Table\_Candidat\_Rectifie AS TCR

LEFT JOIN WEB\_CONST.dbo.partis\_politiques\_N AS PP ON (PP.id\_spp = TCR.PartiPol AND TCR.PartiPol <> 0 )

LEFT JOIN BD\_ELEC\_2014.dbo.Table\_Expert\_Rectifie AS EXPR ON (EXPR.NumCand = TCR.NumCand)

LEFT JOIN BD\_ELEC\_2014.dbo.Table\_Mandataire\_Rectifie AS MAND ON (MAND.NumCand=TCR.NumCand)

LEFT JOIN BD\_ELEC\_2014.dbo.Table\_Association\_Rectifie AS ASS ON (ASS.NumCand=TCR.NumCand)

LEFT JOIN BD\_ELEC\_2014.dbo.Table\_Scrutin AS TS ON (TCR.NoScrutin = TS.NoScrutin)

LEFT JOIN WRAPP.dbo.Table\_Rapporteur AS RAPG ON (TS.RappG = RAPG.CodeRap)

LEFT JOIN WRAPP.dbo.Table\_Rapporteur AS RAP ON (TS.Rapp = RAP.CodeRap)

LEFT JOIN BD\_ELEC\_2014.dbo.Table\_Rapport\_RappG AS TRAPPG ON (TCR.NumCand = TRAPPG.NoCandidat)

LEFT JOIN WEB\_CONST.dbo.Table\_TypeDecision AS DECTYPE ON (TRAPPG.CodeDecision = DECTYPE.CodeTypeDecision)

LEFT JOIN BD\_ELEC\_2014.dbo.comptereserve AS TRES ON (TCR.NumCand = TRES.numcand AND TRES.recours = '0')

LEFT JOIN BD\_ELEC\_2014.dbo.nouveaux\_recours AS R ON (TRES.id\_id = R.id\_recours)

LEFT JOIN WEB\_CONST.dbo.Table\_Departement AS DEP ON (TS.NoDepart = DEP.NoDep)

LEFT JOIN WEB\_CONST.dbo.Table\_Elections AS TELEC ON (TS.Election = TELEC.NoElections)

LEFT JOIN WEB\_CONST.dbo.Table\_TypeElections AS TTELEC ON (TS.TypeElection = TTELEC.TypeElection)

LEFT JOIN WRAPP.dbo.filieres AS FIL ON

(TS.RappG = FIL.num\_filiere and FIL.date\_debut <(CASE WHEN LEN(TS.datepasscomm) = 10 THEN

CONVERT(VARCHAR,CAST(TS.DatePassComm AS DATETIME),112)

ELSE

CASE WHEN LEN(TCR.DatePassCCFP) = 10 THEN

CONVERT(VARCHAR,CAST(TCR.DatePassCCFP AS DATETIME),112)

ELSE

CONVERT(VARCHAR,CAST(TRES.datepasscomm AS DATETIME),112)

END

END))

WHERE TRAPPG.CodeDecision >= '0'

AND TCR.NumCand IS NOT NULL

AND TCR.NumCand = **'3851'**

ORDER BY TCR.NumCand,FIL.date\_debut DESC