

ECOLE PRATIQUE DES HAUTES ETUDES COMMERCIALES

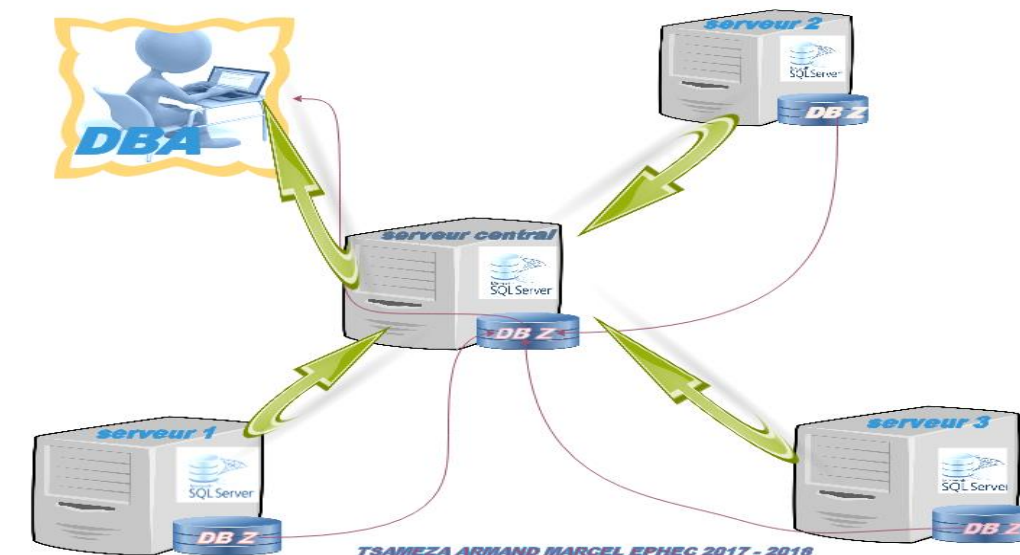


Avenue du Ciseau, 15

1348 Louvain-la-Neuve

RAPPORT DU TRAVAIL DE FIN D'ETUDE

Conception et implémentation d'un système de surveillance et d'alerte pour les serveurs Microsoft SQL



Armand TSAMEZA 3TL1

Rapporteur : Y. Delvigne

Année académique 2017-2018

Remerciements

Je n'aurai pas su écrire ce rapport sans l'aide de plusieurs m'ayant aidé lors de la rédaction et surtout ayant contribué à la bonne réalisation de mon projet.

Tout d'abord, j'adresse mes remerciements à **Mr Olivier Desmedt, et Mr Fabien Vrancken responsables du service DBA au sein de l'entreprise RADIONOMY**, pour le sujet fourni, le temps passé ensemble et le partage de leur expertise. Grâce à leur confiance j'ai pu m'accomplir dans certaines tâches. Ils ont été d'une aide précieuse dans les moments difficiles.

Je tiens à remercier vivement mes rapporteurs, **Mr Yves Delvigne, et Mr Laurent Schalkwijk de L'EPHEC** qui m'ont beaucoup orienté dans mes recherches. Leur écoute et leurs conseils m'ont permis de cibler et de bien orienter mes recherches, cela m'a permis de trouver des articles dont les sujets correspondaient à mon projet.

Je remercie également tout le personnel de l'EPHEC pour leur travail abattu pour nous durant tout ce cursus.

Enfin, je tiens à remercier toutes les personnes qui m'ont conseillé et qui ont relu lors ce rapport, ma famille, mes amis **et mes** camarades de promotion.

Table des matières

I.	Introduction générale.....	1
II.	Cahier de charge.....	2
III.	Etude théorique du sujet.....	3
III.1	Portée et importance du sujet	3
III.2	Difficultés liées au sujet.....	3
III.3	Etude comparative des outils de monitoring	3
III.4	Conclusion	4
IV.	Environnement de travail.....	5
V.	Mise en place de l'ossature du système d'alerte	6
V.1	Architecture générale du système de surveillance	6
V.1.1	Schémas.....	6
V.1.2	Description	6
VI.	Structure de la base de données.....	7
VI.1	Schémas.....	7
VI.1.1	Le modèle conceptuel de données(MCD)	7
VI.1.2	Le modèle logique de données(MLD).....	8
VI.1.3	Description	10
■	TABLE PROCESSLOG	12
VI.1.4	Notion de modularité	15
VI.1.5	Conclusion	16
VII.	Développement d'une solution de l'architecture du système mis en place.....	16
VII.1	Mise en place de la sécurité	16
VII.1.1	Gestion des droits d'accès aux objets	16
VII.1.2	Gestion de l'accès aux données	18
VII.2	Système de collecte de données	20
VII.2.1	Schémas.....	20
VII.2.2	Description	20
VII.3	Système d'agrégation des données.....	21
VII.3.1	Schémas.....	21
VII.3.2	Description	21
VII.4	Script générique d'installation	22
VII.5	Déploiement des données vers le serveur central.....	23
VIII.	Configuration de la gestion des mails	28
IX.	Les alertes.....	29

IX.1	Schéma	30
IX.2	Description	30
X.	Difficultés rencontrées et solutions apportées	30
XI.	Conclusion générale	32
XII.	Recommandations.....	33
	Outils	33
	Description du processus	33
	Instructions.....	33
	Ordre d'exécution	33
XIII.	Ressources.....	34
XIV.	Annexes	35
XIV.1	Outil d'administration	35
XIV.2	Les IDE (environnement de développement intégré)	35
	XIV.2.1 SSMS (SQL server management studio).	35
	XIV.2.2 Visual studio.	36
XIV.3	Les différents logiciels utilisés pour organiser cette tâche	36
	XIV.3.1 XMIND	36
	Pour l'organisation et la représentation de connaissances.	36
	XIV.3.2 Clickcharts.....	36
	XIV.3.3 Draw io conception de diagramme	37
XIV.4	Code source	37

I. Introduction générale

Dans le cadre de notre parcours à l'Ephec, il nous est demandé de développer un sujet représentant notre projet de fin d'étude. L'objectif ici étant de mener une mission concrète venant d'un besoin exprimé ou non par une entreprise. Ce projet devant nous permettre par la même occasion d'acquérir une formation professionnelle demandant de la rigueur, de la méthode et de l'organisation. Le but est de démontrer qu'avec nos acquis d'apprentissage accumulés, qu'on est capable non seulement de traiter un problème concret, c'est-à-dire faire l'analyse, de rechercher les solutions et d'en réaliser un cas pratique, mais aussi de prouver sa capacité d'insertion socioprofessionnelle par une réalisation autonome. Et, faisant mon stage dans une société de la place (www.radionomy.com), un sujet m'a été fourni dans ce cadre.

Il s'agit de la conception et l'implémentation d'un système de surveillance et d'alerte pour les serveurs Microsoft SQL.

L'informatique est toujours en constante évolution, et par conséquent très complexe quelques soit son secteur d'activité. Le monitoring, ayant comme simple définition la surveillance à l'aide d'un moteur (composant de contrôle), est une activité primordiale pour le service informatique. La grande partie des entreprises aujourd'hui pour fonctionner, font une acquisition et un traitement de données à travers un moteur de bases de données. Ces données, d'une grande sensibilité, sont primordiales et la disponibilité ainsi que la qualité de service conditionnent le bon fonctionnement de l'entreprise. Dès lors, comment mettre en place dans ces moteurs de bases de données, un système de surveillance qui permettra de détecter rapidement les processus défectueux ?

Dès lors, mon projet à consister à faire la mise en place d'un système qui aura pour but non seulement de garantir la disponibilité et le fonctionnement des services (SGBD) en cas de panne ou baisse de performances, mais aussi de tenter de faire une prévention des problèmes pouvant survenir et garantissant ainsi par la même occasion une remontée rapide de l'information et un temps d'intervention réduit.

Dans le cadre de cette mission, j'avais deux responsables au sein de l'Entreprise, à savoir Monsieur **Olivier Desmedt (Data Manager)** et Monsieur **Fabien Vrancken (DBA)** à qui je m'adressais pour des questions d'ordre technique. Ils sont responsables de la gestion et l'administration de toutes les bases de données que compose l'entreprise. En outre comme responsable au sein de l'établissement, j'avais comme rapporteur Monsieur **Yve Deligne** qui dans ce même cadre était disponible pour toute question éventuelle en rapport avec le sujet.

Tout en présentant le cahier de charges et la ligne de temps à suivre pour ce projet, ce rapport fera ressortir dans un premier temps une étude théorique du sujet, par la suite présentera l'architecture du système, puis à travers cette architecture décrira le système de collecte et d'agrégation des données qui seront par la suite rapatriées vers un serveur central aux fins d'analyses et de création des différentes alertes.

II. Cahier de charge

Cette partie formalise les différents besoins de l'entreprise. Il s'agit ici des directives à suivre durant la réalisation ce projet.

Le responsable DBA (administrateur des bases de données) de l'entreprise RADIONONMY désire une solution de surveillance des serveurs Microsoft SQL pour prévoir les pannes ou les éventuels problèmes liés à l'utilisation des ressources de ceux-ci. Ci-dessous sont reprises les différentes étapes par lesquelles est passé ce projet pour voir le jour.

- **Prise de contact :**
 - Pendant cette phase, il était question de prendre connaissance de l'environnement de travail, de l'environnement technique, et de la mission pour laquelle je devais par la suite réaliser une analyse.
- **Analyse :**
 - Plusieurs réunions avec diverses personnes ont été effectuées afin de déterminer les différents besoins ainsi que les aspects de la mission
 - Ces réunions avaient pour finalité la réalisation d'un document avec l'architecture et les techniques à utiliser pour mener à bien cette tâche.
 - Validation par le responsable.
- **Mise en place de l'ossature du système d'alerte :**
 - La première phase du développement fut le travail sur l'architecture de la solution, mise en place des différents éléments nécessaires à la mise en place des alertes
 - Validation par le client.
- **Mise en place d'une alerte de type technique :**
 - Création
 - Test
 - Validation par le responsable.
- **Mise en place d'une alerte de type fonctionnelle :**
 - Création
 - Test
 - Validation par le responsable.

III. Etude théorique du sujet

Afin de donner une orientation claire à mes recherches, il est important d'étudier la faisabilité du sujet à travers des concepts clés et des idées préexistantes en lien avec celui-ci. Cette partie présentera tout d'abord la portée et l'importance du sujet, étudiera les difficultés liées à celui-ci, établira une étude comparative des outils de surveillance sur le marché pour en finir avec l'environnement de travail.

III.1 Portée et importance du sujet

Dans son originalité, ce sujet est pertinent dans la mesure où non seulement il est réalisé à la demande d'une entreprise, mais aussi, me permettra d'intégrer plusieurs notions acquises durant ma formation. Il s'agit entre autres de :

- **Gestion de projet** : c'est une démarche visant à organiser de bout en bout le bon déroulement d'un projet. Une bonne gestion des facteurs opérationnels et de méthodologie font qu'un projet puisse aboutir dans un triangle représentant l'équilibre qualité-coût-délai (QCD).
- **Sécurité informatique** : discipline qui vise la protection de l'intégrité et la confidentialité des informations stockées dans un système informatique, en outre le contrôle des utilisateurs et des droits d'accès aux données.
- **Service réseau** : fonctionnalité assurée par un ou plusieurs serveurs ayant pour fonction la fourniture d'informations à d'autres ordinateurs via une connexion réseau normalisée.
- **SGBD** : système qui stocke et gère des données de façon organisée et cohérente. Un **SGBDR**, quant à lui, fera la même chose de manière relationnelle.

III.2 Difficultés liées au sujet

Ce sujet présente plusieurs difficultés car il enveloppe des technologies jamais abordées durant ma formation à l'Ephec. Il s'agit de Microsoft SQL server qui est un système de gestion de base de données développé et commercialisé par la société Microsoft. Aussi, des mots tels que surveillance et alertes auxquels il fallait trouver un sens en liaison avec le sujet.

Face à toutes ces difficultés, mes recherches ont été effectuées sur plusieurs solutions de monitoring dans la mesure de pouvoir à tout moment, en cas de blocage avec une solution, basculer vers une autre plus adéquate.

III.3 Etude comparative des outils de monitoring

Durant cette phase, plusieurs réunions ont été effectuées avec les responsables du service Data de l'entreprise aux fins de déterminer les différents aspects de la mission. Ces réunions m'ont surtout donné des pistes de recherche des technologies à utiliser pour réaliser cette tâche. En outre de déterminer certains problèmes auxquels ils sont confrontés quotidiennement.

Lors de l'analyse réflexive et les différentes recherches effectuées sur comment mener à bout cette mission, j'ai découvert trois catégories d'outils de surveillances à savoir :

- **Les outils open source**

On entend par open source un logiciel gratuit qu'on peut non seulement télécharger et installer gratuitement, mais aussi configurer selon ses propres besoins. Il en existe plusieurs dont ce document liste ceux ayant lu la documentation. Il s'agit de :

- ✓ **Zabbix** : logiciel permettant de surveiller l'état de divers services réseau, serveurs et autres matériels réseau et produisant des graphiques dynamiques de consommation des ressources. Il repose sur 3 composants à savoir :

Une partie serveur, cœur de la solution, qui prendra en charge le processus de collecte, d'agrégation, et de stockage de données.

Une partie agent qui, déployé sur les machines à superviser, sera non seulement interrogée par le serveur, mais aussi servira de transfert des données vers le serveur.

Une partie front-end qui met en forme à travers une interface web les données sous forme graphique pour une consultation des dernières mesures.

- ✓ **Léptide** : utilitaire gratuit permettant d'analyser les performances et la structure des serveurs SQL. Il offre une visibilité complète de l'espace de stockage et des performances. Il permet aux DBA d'identifier les problèmes potentiels avant qu'ils ne surviennent rapidement.

➤ **Les outils payants**

Comme les outils open sources, nous avons aussi plusieurs logiciels payants servant à faire du monitoring. Il s'agit de :

- ✓ **Apex SQL**

Il permet la surveillance des paramètres SQL Server, du système et des bases de données. Parmi ces paramètres on peut citer : espérance de vie des pages, taux d'accès au cache tampon, utilisation du processeur, mémoire disponible, taille de la base de données, croissance des journaux, taille disque physique.

- ✓ **OP5**

Il s'agit d'un logiciel qui offre de surveiller tous les types de serveur, y compris physique et virtuel. Par conséquent, cela peut être un bon choix pour surveiller un serveur SQL en tant qu'option la plus flexible. En outre, OP5 peut également gérer un appareil avec un gros volume. La métrique clé fournit un tableau de bord pour montrer les performances et la santé du serveur de disponibilité, les statistiques d'utilisation et l'identification.

➤ **Les outils natifs de SQL server**

Microsoft SQL Server est fourni avec un ensemble d'outils complet nécessaire à la surveillance des ressources du serveur et l'optimisation des bases de données physiques qui s'y trouvent. Le choix de l'outil dépend du type de surveillance ou de réglage à effectuer et des événements particuliers à surveiller. Ci-dessous quelques outils de surveillance et d'optimisation fournis avec SQL Server.

- ✓ **Data collector** : intégré dans SQL server agent, il s'agit d'un composant qui collecte différents ensembles de données. Cette collecte est exécutée en permanence ou selon un calendrier défini par l'utilisateur au final pour être stockée dans une base de données relationnelle appelée entrepôts de données de gestion aux fins d'analyse.
- ✓ **Extended EVENTS (événements étendus)** : système général de gestion d'événements pour les systèmes de serveur. Il prend en charge la corrélation des données provenant de SQL Server et, dans certaines conditions, la corrélation des données provenant du système d'exploitation et des applications de base de données.
- ✓ **[DMV \(Dynamic Management Views\)](#) et [DMF \(Dynamic Management Functions\)](#)** : il s'agit de vues système permettant de surveiller, diagnostiquer et régler les problèmes de performance de l'intégrité d'une instance SQL Server.

III.4 Conclusion

Au terme de cette petite description, il devient évident de comprendre que si vous utilisez une version de SQL server, vous avez à votre disposition tous ses outils natifs pour administrer votre

environnement selon vos besoins. Le tableau ci-dessous présente quelques avantages et inconvénients des deux premières catégories.

CATEGORIES	OUTILS	AVANTAGES	INCONVENIENTS
OPEN SOURCE	Zabbix	Gratuit, facilité d'installation pour les débutants	Documentation et interface graphique complexe
	Lepide	Suivre, gérer et planifier efficacement l'utilisation de l'espace disque sur vos serveurs SQL	
PAYANTE	Apex SQL	Propose diverses solutions clé en main. Rien ne doit être configuré.	Le prix
	OP5	Tableau de bord entièrement personnalisable et interactif	Le prix

IV. Environnement de travail

Pour sa facilité d'installation et de configuration, mon choix était Zabbix que j'avais commencé à installer et configurer certains composants. Mais à la sortie d'une réunion avec le client, j'ai réalisé que l'entreprise travaillait exclusivement sur l'environnement Windows contrairement à Zabbix qui lui est utilisable dans un environnement linux. Comme tout produit, il se doit d'être développé aux fins d'utilisation par l'entreprise. Après plusieurs échanges avec le client, nous avons opté à la sortie de cette réunion l'utilisation des outils natifs de SQL server. En outre ce choix s'est fait tout en respectant le cahier de charges de l'entreprise ceci dans l'optique de ne pas développer un produit avec des technologies non maîtrisées par des utilisateurs finaux et qui par la suite deviennent inutiles. Ainsi les différents éléments mis en place sont :

➤ OS (système d'exploitation)

L'entreprise travaille principalement sur des serveurs Windows de la version 2012 et supérieur. Ils regroupent des systèmes d'exploitation orientés **serveur** de Microsoft

➤ Outil d'administration

Le moteur de base données est SQL server version 2014 et plus. Concrètement, il s'agit d'un outil qui possède toutes les caractéristiques pour accompagner l'utilisateur dans l'administration, la manipulation, le contrôle, le tri, et la mise à jour des bases de données.

➤ Les IDE (environnement de développement intégré)

✚ **SSMS** (SQL server management studio). *Il s'agit d'un environnement intégré permettant de gérer une infrastructure SQL Server. Il fournit une interface utilisateur et un groupe d'outils avec des éditeurs de scripts enrichis qui interagissent avec SQL Server.*

✚ **Visual studio**. C'est un ensemble complet d'outils de développement permettant de générer des applications. Aussi utilisé pour les projets décisionnels, il a permis de créer le package SSIS pour le déploiement des données vers le serveur central.

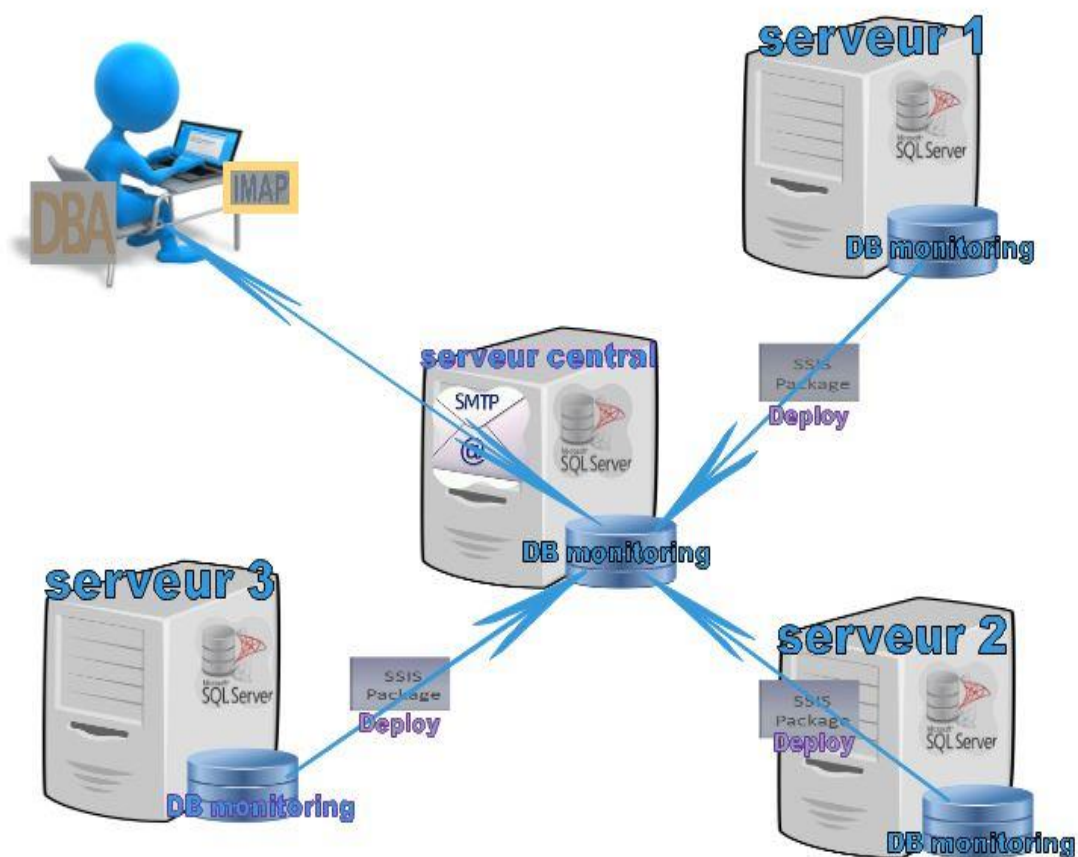
- **Les différents logiciels** utilisés pour organiser cette tâche sont :
 - ✓ **XMIND** pour l'organisation et la représentation de connaissances.
 - ✓ **Clickcharts** logiciel convivial de création d'organigramme et de diagramme.

V. Mise en place de l'ossature du système d'alerte

En rapport avec le cahier de charge, il s'agit d'un système devant fonctionner aussi bien sur plusieurs serveurs que sur plusieurs bases de données. Dès lors, cette partie présentera dans un premier temps l'architecture générale du système, décrira par la suite la structure de la base de données de surveillance, pour finir avec une explication sur la notion de modularité.

V.1 Architecture générale du système de surveillance

V.1.1 Schémas



TSAMEZA ARMAND MARCEL EPHEC 2017 - 2018

Figure 1 : architecture du système d'alerte

V.1.2 Description

Un système d'information est aujourd'hui un élément central du fonctionnement d'une entreprise. On le définit comme un ensemble de ressources (données, processus, équipement informatique) permettant la collecte, le stockage, la structuration, la gestion, le transport, la manipulation, l'échange, l'analyse et la diffusion des informations (textes, images, sons, vidéo) au sein d'une entreprise. Parmi ces ressources figurent les bases de données et système de gestion de bases de données (SGBD). Cette définition laisse entrevoir la complexité du système

d'information, dont les déclinaisons vont s'exprimer à l'aide d'une architecture. Une architecture désigne la structure inhérente à un système informatique et des relations entre les éléments. Cette structure fait suite à un ensemble de décisions stratégiques prises durant la conception du système.

En rapport avec le schéma de la figure 1, le système présente plusieurs serveurs à monitorer contenant chacun une instance SQL server. On trouve dans ces instances une multitude de bases de données qui feront elles-mêmes partie du système de surveillance. Ainsi, pour recueillir les ressources hardware et software du serveur ainsi que les caractéristiques des différentes bases de données se trouvant dans une instance, il est judicieux de mettre en place dans chaque instance une base de données de surveillance (data base nommée "monitoring") dont la structure sera présentée plus tard.

Le système devant être centralisé (rassemblement des ressources nécessaires à un traitement sur un hôte central), il est mis en place un serveur central qui recevra dans sa base de données de surveillance (DB monitoring) ses données de surveillances ainsi que les données de surveillances de tous les autres servers aux fins d'analyses. Ce système est avantageux dans la mesure où, en cas de rupture d'un lien entre le serveur central et les autres serveurs, les informations continueront à être collectées sur chaque serveur localement de telle manière qu'après rétablissement du lien, ces informations pourront être déployées vers le serveur central. Ce système, très pratique pour une entreprise, présente toutefois quelques inconvénients : dans la mesure où le serveur central tombe en panne, le système ne fonctionne plus. Aussi, un problème de saturation peut survenir s'il y'a beaucoup de connexions simultanées ponctuellement non prévues. Mais pour réduire au minimum ces problèmes, la partie sécurité expliquera brièvement les éléments à mettre en place.

VI. Structure de la base de données

Une fois l'architecture générale du système mis en place vient le volet de la structure de la base de données de surveillance (DB monitoring). Comme définition, une base de données est un ensemble structuré et organisé permettant le stockage de grandes quantités d'informations afin de faciliter l'exploitation (ajout, mise à jour, et recherche de données). Ces informations peuvent être en rapport avec une activité donnée et peuvent être utilisées par des utilisateurs communs. Le système faisant une collecte de données aux fins de prévention des problèmes ultérieurs, il était question de mettre en place une base de données de réception des différentes informations rassemblées.

VI.1 Schémas

Deux sortes de modèle schématisent la structure de la base de données. Il s'agit du modèle conceptuel et du modèle logique.

VI.1.1 Le modèle conceptuel de données(MCD)


Permet de représenter de façon formelle les flux de données qui seront utilisées par le système d'information. Il s'agit donc d'une représentation des données, facilement compréhensible, permettant de décrire le système d'information à l'aide d'entités (représentation d'un élément matériel ou immatériel ayant un rôle dans le système que l'on désire décrire) et de relations entre ces dernières.

Permet de modéliser la structure selon laquelle les données seront stockées dans la base de données. Il utilise le formalisme graphique merise (méthode d'analyse et de conception d'un système d'information) et donnera une clarification sur la cardinalité des données.

```

classDiagram
    class processType {
        processTypeid
        processTypeName
    }
    class processGroupe {
        processGroupeld
        processGroupeName
    }
    class processLog {
        processLogId
        processLogName
        lastExecuteLog
        isProcessError
        ErrorMessage
    }
    class process {
        processId
        processName
        processdescription
        processType
        defaultWait
        monitoringLength
        lastDuration
        joblevel
        retention
        processRetention
        auteur
        createDate
    }
    class planification {
        planificationId
        parameter
        serverName
        wait
        startTime
        endTime
        startDate
        endDate
    }
    class metrics {
        metricsId
        serverName
        value1
        value2
        value3
        value4
        unitMeasure
        dbName
        createDate
        isSync
    }
    class express {
        expressId
        serverName
        dbName
        tableName
        no_recompute
        is_incremental
        data_compression_desc
        isSync
    }
    class processAggregation {
        processAggregationId
        processAggregationName
        frequency
    }
    class aggregation_by_day {
        aggregationDiskId
        serverName
        currentValue
        minimum
        maximum
        croissance
        volumePoint
        tableRecort
        unit
        description
        dbName
        createDate
        isSync
    }
    class processAggregationLog {
        aggregationLogId
        aggregationLogLogName
        lastExecuteLogOfAggregate
        isProcessError
        ErrorMessage
    }

    processType "1,1" -- "1,N" process : comprend
    processType "1,N" -- "1,1" process : représente
    processGroupe "1,1" -- "1,N" process : possède
    processLog "1,N" -- "1,1" process : possède
    processLog "0,1" -- "1,1" process : génère
    process "1,1" -- "0,N" planification : possède
    process "1,1" -- "0,N" metrics : possède
    process "1,1" -- "0,1" processAggregation : possède
    process "1,1" -- "1,N" express : possède
    process "1,1" -- "1,1" processAggregation : possède
    process "1,1" -- "0,N" aggregation_by_day : possède
    processAggregation "1,1" -- "1,N" processAggregationLog : possède
  
```



➤ MLD

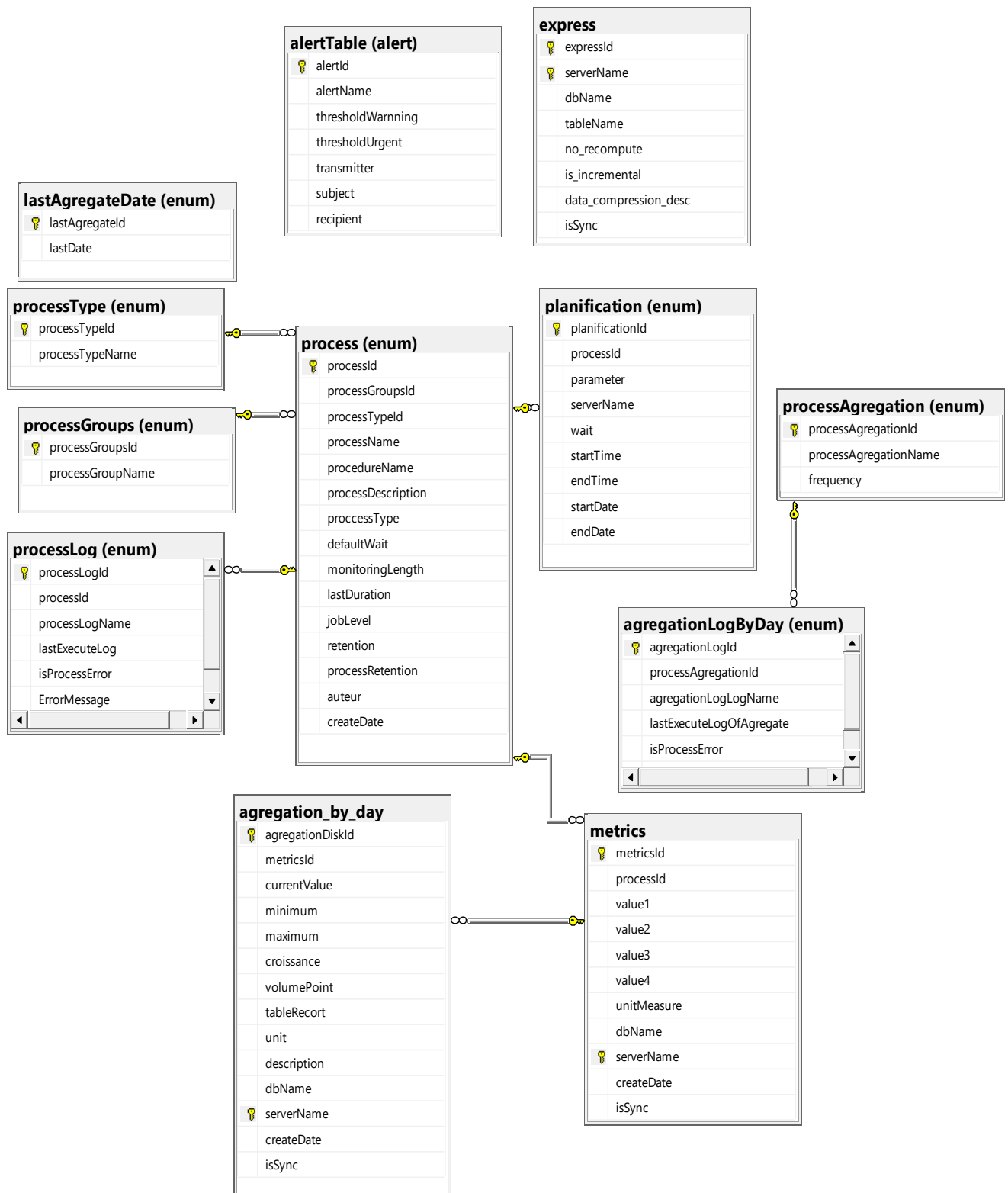


Figure 3 : modèle logique de données

VI.1.3 Description

Au travers de ce schéma logique, on constate bien que la base de données contient plusieurs tables dont cette partie prendra le soin de décrire la fonctionnalité de chaque table et le type d'informations qu'elle recevra. Ainsi, pour s'y faire, les tableaux ci-dessous permettront de comprendre au mieux le contenu de ces tables.

■ Table processGroup

COLUMN	DATATYPE	REPRESENTATION	EXEMPLE	DESCRIPTION
ProcessGroupId	Int (not null)	Nombre	Incrémentation par pas de 1	PK
ProcessGroupeName	Varchar (100) null	Texte	CPU, disk, mémoire, data base etc....	<ul style="list-style-type: none"> Permet de regrouper les processus par groupe de monitoring

■ Table processType

COLUMN	DATATYPE	REPRESENTATION	EXEMPLE	DESCRIPTION
ProcessTypeId	Int (not null)	Nombre	Incrémentation par pas de 1	PK
ProcessTypeName	Varchar (100)	Texte	Procedure, job, Extended events etc....	<ul style="list-style-type: none"> Cette colonne indique l'action utilisée pour exécuter le processus

■ TABLE PROCESS

COLUMN	DATATYPE	REPRESENTATION	EXEMPLE	DESCRIPTION
ProcessId	Int	Nombre	Incrémentation par pas de 1	PK
ProcessGroupId	Int	Nombre		FK permet une jointure de la table processgroup à la table process de manière à associer chaque processus à l'identifiant de son groupe
ProcessTypeId	Int	Nombre		FK permet une jointure de la table processType à la table process de manière à associer chaque processus à l'identifiant de son type
ProcedureName	Varchar (100)	Texte	Taille physique disk Taille physique mémoire	Cette colonne désigne de manière unique tous les processus susceptibles d'être exécuté (processus qui sont des procédures stockées)

Paramètre	Varchar (100)	Texte	(C ; K ; E ;)	Permet d'indiquer si la procédure Name contient des paramètres ou pas
ProcessDescription	Varchar (1000)	Texte		Explication de la procédure Name
ProcessType	Varchar (100)	Texte	Procédures, XE, Job	Indiquera de manière informative le type de process
DefaultWait	Int	Minutes	5 : toutes les 5 min. 10080 : une fois par semaine	Donnée temporelle permettant de savoir à quelle fréquence un process doit être lancé.
MonitoringLength	Int	Jours	-1 : illimité 10 : 10 jours	Durée de planification monitoring. Exemple, on ne veut lancer ce monitoring sur une durée de 10 jours.
LastExecute	Datetime2(0)	Date	20180101 00 :00:00	Date du premier monitoring. Cela servira à : <ul style="list-style-type: none"> • Avoir une info dans les rapports • Calculer la fin d'un monitoring avec la colonne MonitoringLength.
LastDuration	Int	Secondes	10 : durée de 10sec.	Temps total qu'a duré le dernier lancement de ce process. Cela servira à calculer : <ul style="list-style-type: none"> • Dans quel job (Job Type) doit être rangé le lancement de ce futur process • Quelles sont les process qui prennent trop de temps
JobLevel	Int		1 : process < 1 secondes 2 : process < 6 secondes et >= 1 seconde 3 : process >= 6 secondes	Ceci afin d'éviter que d'éventuels process durant très longtemps bloquent les petits process qui doivent se lancer à des faibles intervalles.
Rétention	Datetime2(0)	Date	Tous les 6 mois	Indique ici une période à laquelle les métriques pourront être nettoyées de la base de données
ProcessRetention	Datetime2(0)	Date		Indique ici la période pendant laquelle les process pourront être supprimé de la base de données

■ Table PLANIFICATION

COLUMN	DATATYPE	REPRESENTATION	EXEMPLE	DESCRIPTION
--------	----------	----------------	---------	-------------

PlanificationId	INT	Nombre	Incrémentation par pas de 1	PK
ProcessId	INT	Nombre		<ul style="list-style-type: none"> FK permettant la jointure entre la table des processus et celle de planification Dans un point de vu d'associer une planification à un process
ServerName	Varchar (100) Default null	Texte	Armand	<ul style="list-style-type: none"> Cette colonne permettra à tout moment de savoir sur quel serveur a été exécuté le process
Wait	INT	Minutes	5 : toutes les 5 min. 10080 : une fois par semaine	Donnée temporelle permettant de savoir l'intervalle d'exécution d'un process.
StartTime	Time (0) Default null	Heure	13 :00 :00 démarre à 03 :00 :00 démarre à	<ul style="list-style-type: none"> Donnée temporelle permettant de savoir l'heure de démarrage d'un processus
EndTime	Time (0) Default null	Heure	11 :00 :00 s'arrête à 03 :00 :00 s'arrête à	Donnée temporelle permettant de savoir l'heure à laquelle le process sera stoppé
StartDate	Date Default null	Jours	Tous les jours Tous les 3 jours	Donnée temporelle permettant de savoir la date de démarrage d'un processus
EndDate	Date Default null	Jours	Après un jour Après 5 jours	Donnée temporelle permettant de savoir la date d'arrêt d'un processus

■ TABLE PROCESSLOG

VCOLUMN	DATATYPE	REPRESENTATION	EXEMPLE	DESCRIPTION
ProcessLogId	Int not null	Nombre	Incrémentation par pas de 1	PK
ProcessId	Int not null	Nombre		FK permet d'associer chaque process à ses logs
ProcessLogName	Varchar (100)	Texte		Cette <i>colonne</i> enregistre tous les processus exécutés
LastExecuteLog	Datetime2(0)			Date à laquelle le processLog a été exécuté
IsProcessError	Bit		0 = process exécuté sans erreur 1 = erreur	Indique si l'exécution du process a réussi ou pas
ErrorMessage	Varchar(max)	Texte		Enregistre les raisons pour lesquelles un process ne s'est pas exécuté

■ Table metrics

COLUMN	DATATYPE	REPRESENTATION	EXEMPLE	DESCRIPTION
MetricId	INT	Nombre	Incrémentation par pas de 1	PK
ProcessId	INT	Nombre		FK permet d'associer chaque process à ses metrics
Value1	Int	Nombre	-456789 Mo (espace disk)	Représente la valeur de la métrique recherchée
Value2	Int	Nombre		
Value3	Varchar (100) Default nulle	Texte		Stocke les différents paramètres provenant de la colonne paramètre de la table des planifications
Value4	Varchar (100) Default null	Texte		
UnitMeasure	Varchar (10) Default null	Texte	Mo, %, ms	Stocke l'unité de mesure des valeurs (value1) des différentes métriques
DbName	Varchar (100) Default null	Texte		Stocke le nom de la base de données pour laquelle la métrique est recherchée
ServerName	Varchar (100) Default null	Texte		Cette colonne permettra à tout moment de savoir sur quel serveur a été exécuté le processus
CreateDate	Datetime2(0)	Date		Stocke à tout moment la date de création de la métrique

■ TABLE processAggregation

COLUMN	DATATYPE	REPRESENTATION	EXEMPLE	DESCRIPTION
ProcessAggregationId	Int (not null)	Nombre	Incrémentation par pas de 1	PK
ProcessAggregationName	Varchar (100)	Texte		<ul style="list-style-type: none"> Indique les procédures qui seront agrégées
Fréquence	Varchar (100)	Texte	Day, hour etc....	<ul style="list-style-type: none"> Indique le moment d'agrégation

■ TABLE AGRegationLog

COLUMN	DATATYPE	REPRÉSENTATI ON	EXEMPLE	DESCRIPTION
--------	----------	--------------------	---------	-------------

AgregationLogId	Int	Nombre	Incrémentation par pas de 1	Pk
ProcessAgregationId	Int	Nombre		Fk
AgregationLogName	Varchar	Texte		Indique le nom de la procédure
LastExecuteLogOfAgregate	Vachar	Texte		Indique la date d'agrégation
IsProcessError	Bit	O, 1		1 en cas d'échec, 0 en cas de réussite
ErrorMessage	Vachar (max)	Texte		Enregistre les raisons pour lesquelles une agrégation ne s'est pas effectuée

■ TABLE AGREGGATION

COLUMN	DATATYPE	REPRÉSENTATION	EXEMPLE	DESCRIPTION
AgregationId	Int	Nombre	Incrémentation par pas de 1	PK
CurrentValue	Int	Nombre	20000	Parmi le set de données qui seront agrégés, Indique la dernière valeur insérée dans la table des métriques.
Minimum	Int	Nombre	44444	Parmi le set de données qui seront agrégés, Indique la valeur maximale de ce set
Maximum	Int	Nombre	88888	Parmi le set de données qui seront agrégés, Indique la valeur minimale de ce set
Croissance	Int	Nombre	3, 4, 8, 10	Parmi le set de données qui seront agrégés, Indique la différence entre la currentValue et
VolumePoint	Varchar	Texte	C, D	
Unit	Varchar	Texte	%, Mo, QTY	Unité de mesure de l'agrégation
Description	Varchar	Texte	Taille du disque	Description de l'agrégation
DbName	Varchar	Texte	Monitoring	Nom de la base de données
Server Name	Varchar	Texte	115	Nom du serveur
CreateDate	Datetime2(0)	Date	20182205	Date de création
isSync	Int	Nombre	1 ou NULL	Colonne qui indiquera si une a été déployée vers le serveur central ou pas

■ TABLE lastAgregationDate

COLUMN	DATATYPE	REPRESENTATION	EXEMPLE	DESCRIPTION
LastAgregateId	Int	Nombre	Incrémentation par pas de 1	PK
LastDate	Datetime2(0)	Texte	Procedure, job, Extended events etc....	<ul style="list-style-type: none"> Après agrégation, récupère la dernière date qui sera une

				référence la prochaine agrégation
--	--	--	--	-----------------------------------

■ TABLE express

COLUMN	DATATYPE	REPRÉSENTATION	EXEMPLE	DESCRIPTION
ExpressId	Int	Nombre	Incrémentation par pas de 1	PK
No_recompute	Bit	Nombre	0 1	Parmi le set de données qui seront agrégés, Indique la dernière valeur insérée dans la table des métriques.
Is_incremental	Bit	Nombre	0 1	Indique si une table a l'option statistique incrémentale
Data_compression-desc	Varchar	Texte	Page, none,	Indique le type de compression d'une table
TableName	Varchar	Texte	Express	Nom de la table
DbName	Varchar	Texte	Monitoring	Nom de la base de données
Server Name	Varchar	Texte	110	Nom du serveur
isSync	Int	Nombre	1 ou NULL	Colonne qui indiquera si une donnée a été déployée vers le serveur central ou pas

■ TABLE Alerte

COLUMN	DATATYPE	REPRÉSENTATION	EXEMPLE	DESCRIPTION
alertId	Int	Nombre	Incrémentation par pas de 1	PK
alertName	Varchar	Texte	Nombre de partition insuffisant à la fin	Indique le nom de l'alerte qui sera exécutée
thresholdWarning	Int	Nombre	100	Indique le seuil d'avertissement pour un KPI
thresholdUrgent	Int	Nombre	70	Indique le seuil d'urgent pour un KPI
transmitter	Varchar	Texte	marceltsameza@gmail.com	Indique l'émetteur du message
subject	Varchar	Texte	Alerte partition	Contenu du message
recipient	Varchar	Texte	marceltsameza@gmail.com	Indique la personne à notifier

VI.1.4 Notion de modularité

Une architecture est dite modulaire lorsque la conception de son système est formée d'éléments qui peuvent être assemblés puis séparément modifiés, retirés, ou ajoutés sans interférer avec le

fonctionnement des autres éléments. Ainsi, notre architecture se voulant modulaire, elle est conçue de telle manière que, à tout moment, on peut ajouter ou retirer des serveurs, ajouter, modifier, ou retirer des bases de données, créer, supprimer, modifier des objets dans une base de données. Ceci, sans pour autant interférer avec le fonctionnement du système de surveillance.

VI.1.5 Conclusion

Une fois l'architecture générale du système mise en place, la structure de la base donnée créée, tout ceci validé par le client, venait l'étape du développement d'une solution du système mis en place.

VII. Développement d'une solution de l'architecture du système mis en place

Au travers du système mis en place précédemment, cette partie permettra de récolter les différentes informations pour enrichir notre système. Pour s'y faire, nous passerons dans un premier temps au niveau sécurité dans l'optique de limiter les droits d'accès à l'information, par la suite on procédera à la collecte de données nécessaires au système. Ensuite, dans une optique de faciliter l'analyse prévisionnelle plus tard, viendra l'étape d'agrégation des données collectées. Une fois Ceci fait, il faudra créer un script générique d'installation sur différents serveurs pour en finir avec la phase de rapatriement des données vers le serveur central aux fins d'analyse.

VII.1 Mise en place de la sécurité

VII.1.1 Gestion des droits d'accès aux objets

La **sécurité informatique** est l'ensemble des moyens techniques nécessaires à la mise en place de moyens visant à empêcher l'utilisation non-autorisée, le mauvais usage, la modification ou le détournement du système d'information (ensemble organisé de ressources qui permet de collecter, stocker, traiter et distribuer de l'information, en général grâce à un ordinateur). Aujourd'hui, la sécurité est un enjeu majeur pour les entreprises ainsi que pour l'ensemble des acteurs qui l'entourent. Avec la sécurité mise en place au niveau du domaine de l'entreprise, un compte était créé par l'administrateur des serveurs au niveau du domaine. Ainsi grâce à ce compte, il était possible de se connecter aux serveurs SQL en utilisant l'authentification Windows.



Figure 4 : connexion au server

Une fois connecté à ceux-ci, une partie de la sécurité interne se trouvait au niveau de la gestion des droits d'accès à la base de données de surveillance. Pour cela, grâce à ce compte du domaine, une connexion est créée au niveau du serveur SQL effectuant par la même occasion un mapping de ce login vers l'utilisateur crée au niveau de la base donnée.

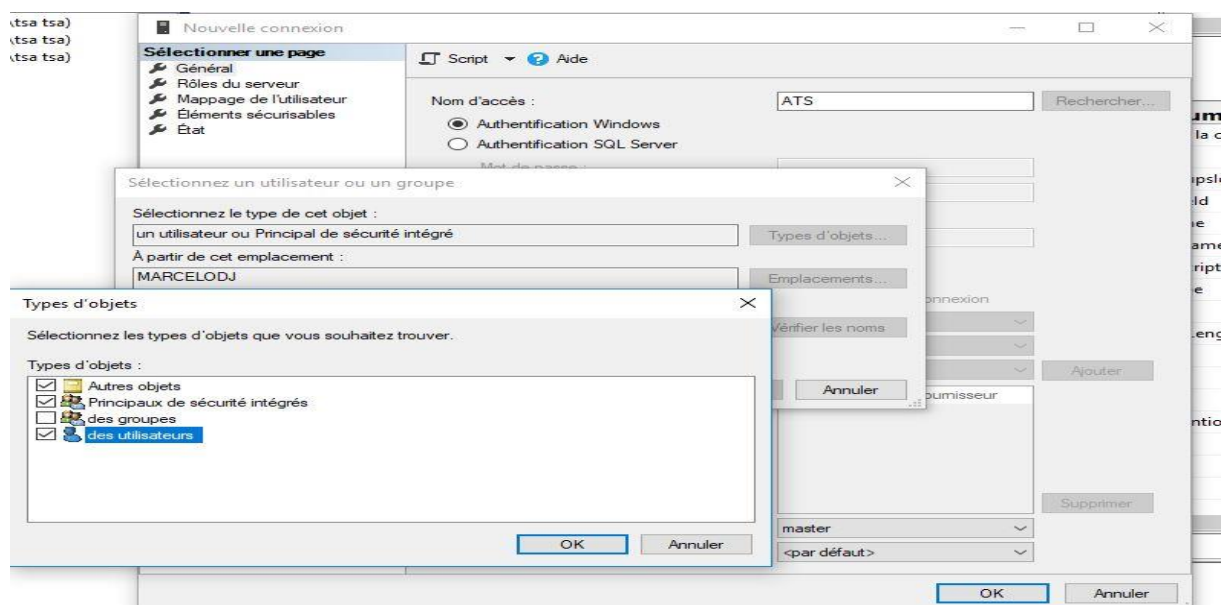


Figure 5 : authentification au domaine

De ce fait, on peut décider des différentes actions que cet utilisateur peut effectuer au niveau des bases de données.

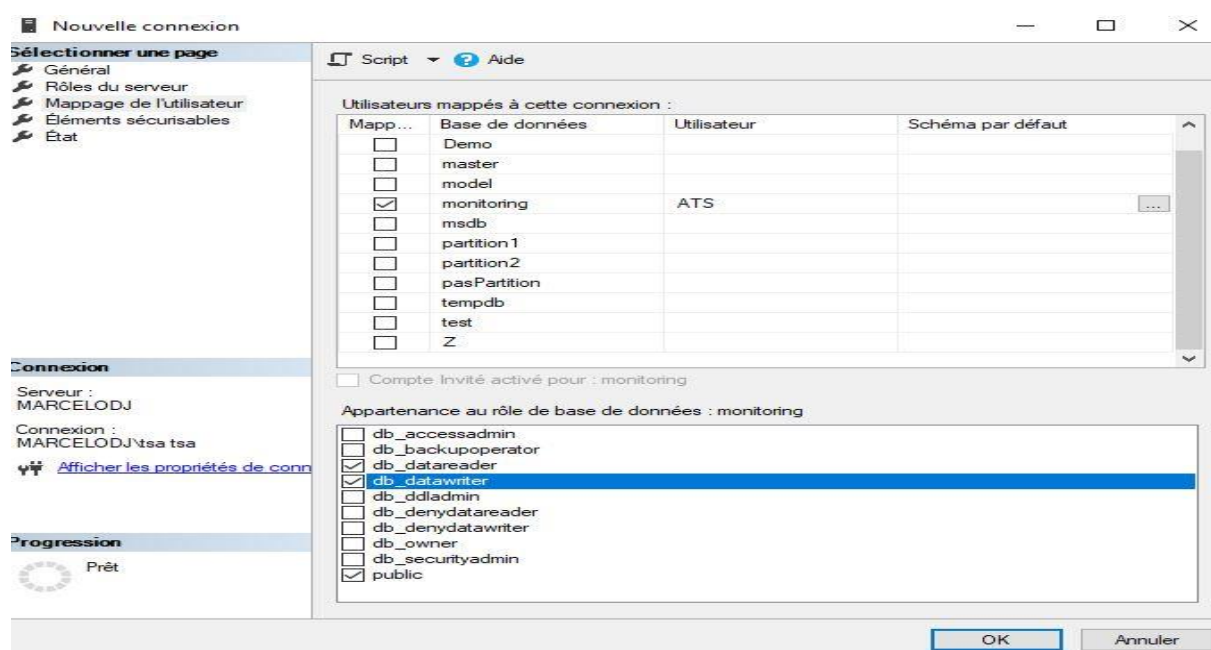


Figure 6 attribution des rôles

VII.1.2 Gestion de l'accès aux données

Outre la sécurité au niveau de la base données, il y'a aussi une possibilité de crypter les données dans les fichiers physiques de la base de données. Sans le certificat de cryptage d'origine et la clé principale, les données ne peuvent pas être lues lors de l'accès au lecteur, du vol du support physique ou en cas d'usurpation. SQL server est fourni avec plusieurs types d'algorithmes de chiffrement. Celui utilisé ici est le chiffrement de données transparent (TDE) donc le schéma ci-dessous décrit la hiérarchie et la réalisation.

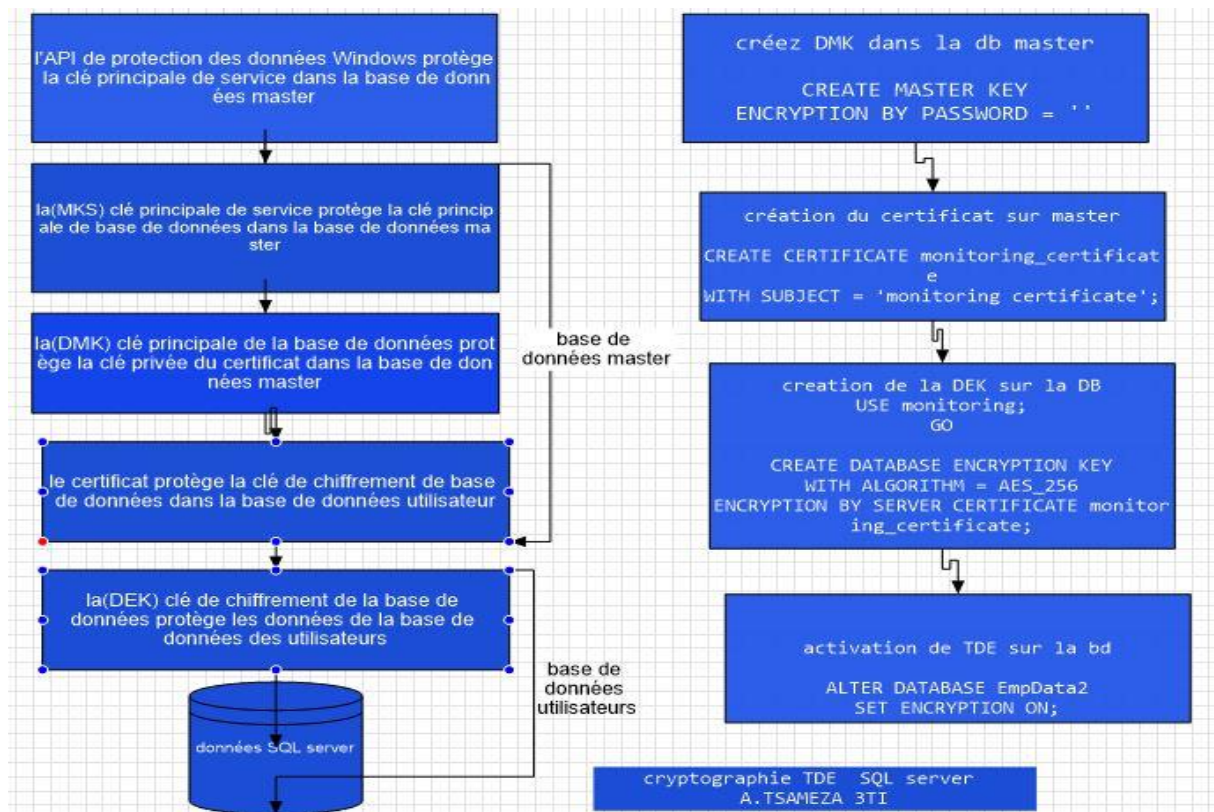


Figure 7 : chiffrement

En ce qui concerne l'image ci-dessous, on voit bien grâce à cette commande de test que la base de données monitoring affiche "3" comme état de chiffrement ce qui indique bien que les données sont chiffrées. Cela dit, il est très important par la suite de faire une sauvegarde du certificat et des différentes clés de chiffrements bien en sécurité.

```

56
57 --commande de test
58 SELECT DB_NAME(database_id) DbName,
59        encryption_state EncryptState,
60        key_algorithm KeyAlgorithm,
61        key_length KeyLength,
62        encryptor_type EncryptType
63 FROM sys.dm_database_encryption_keys;
64
65
  
```

	DbName	EncryptState	KeyAlgorithm	KeyLength	EncryptType
1	tempdb	3	AES	256	ASYMMETRIC KEY
2	monitoring	3	AES	256	CERTIFICATE
3	Demo	3	AES	256	CERTIFICATE

Figure 8 : vérification du chiffrement

VII.2 Système de collecte de données

VII.2.1 Schémas

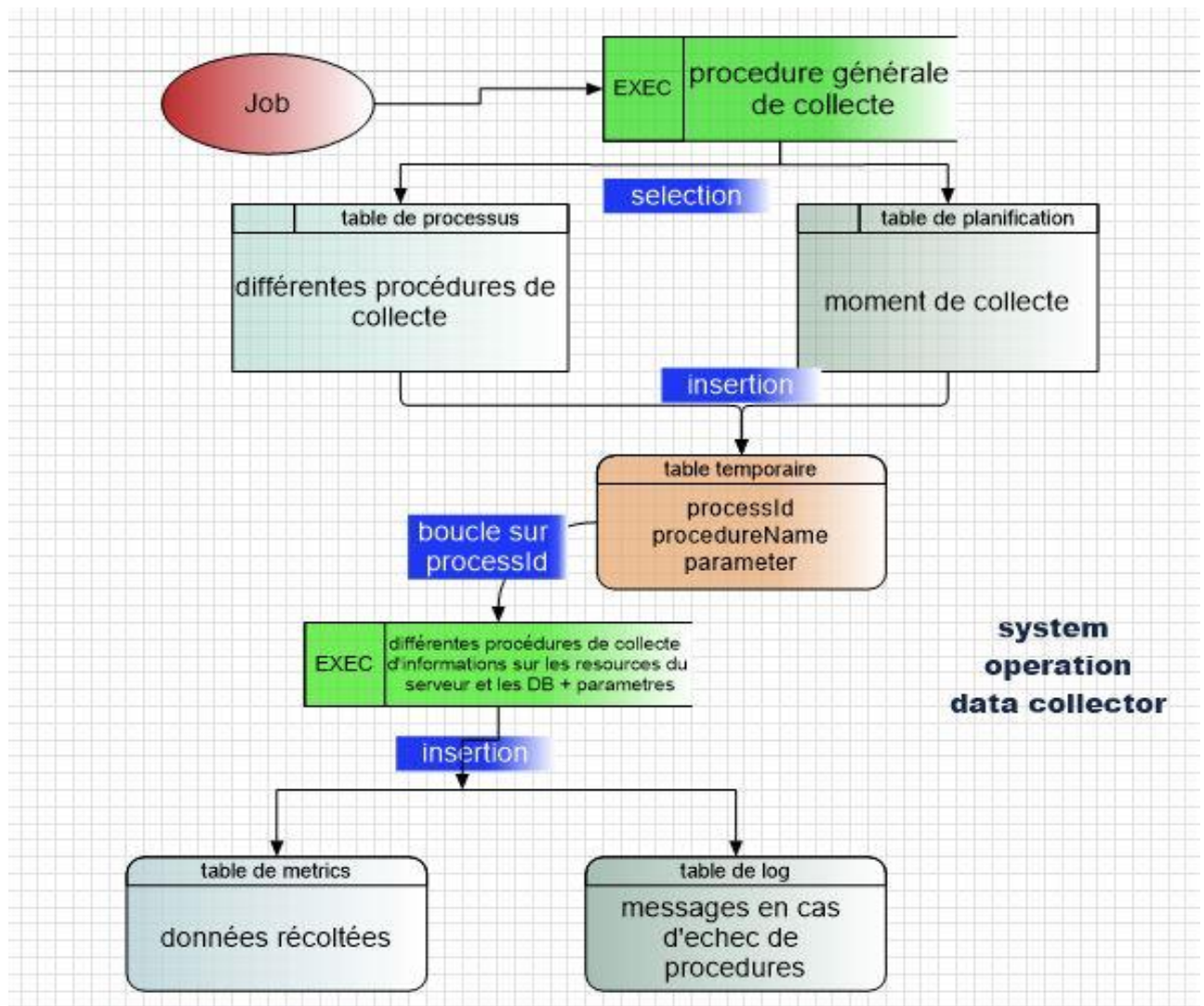


Figure 9 : système de collecte de données

VII.2.2 Description

Un serveur abrite non seulement les composants comme le processeur, la mémoire et les disques, mais aussi les services arrière-plan comme le stockage des informations dans les bases de données. Et, le système devant faire une surveillance des paramètres de ces composants et services, il est donc crucial d'obtenir les informations à propos de ceux-ci. Ces informations aussi appelées métriques sont un type de mesure quantifiable de la performance d'un appareil. Ainsi, le schéma présenté ci-dessus permet de décrire notre système de collecte de métriques en ce qui concerne les différentes ressources et bases de données des serveurs SQL de Microsoft. Pour s'y faire, nous avons dans le système un travail (série spécifiée d'opérations exécutées de manière séquentielle par l'Agent SQL Server) qui exécute une procédure générale. Cette procédure fait une sélection dans la table de processus toutes les procédures de collecte de métriques ainsi que leurs planifications et paramètres correspondant provenant de la table de planification des processus pour insertion dans une table temporaire. La planification ici indique la fréquence d'exécution du processus. Une fois les données insérées dans la table temporaire, une boucle est effectuée sur les différentes lignes de la table pour exécuter les procédures qui y

sont stockées. Ces procédures ayant comme but final de recherche des informations sur composant et services cités plus haut. Une fois les procédures exécutées, les données récoltées sont insérées dans la table des métriques aux fins d'analyse et d'agrégation. Parallèlement, il y'a une table journal qui est mis en place pour recevoir les messages d'erreurs lorsqu'une procédure ne s'est pas exécutée. En ce qui concerne ces procédures, elles ne sont pas toutes exécutées à la même fréquence car il y dans la table des processus une colonne date de création qui indique à quel moment le processus a été créé. Ainsi une procédure ne sera exécutée que si la date de création additionnée à la planification (wait) est inférieure ou égale à la date courante. Dans cette mesure, une fois la procédure exécutée, la date de création du processus est mise à jour pour changer la date de création avec la date courante d'exécution (GETDATE).

VII.3 Système d'agrégation des données

VII.3.1 Schémas

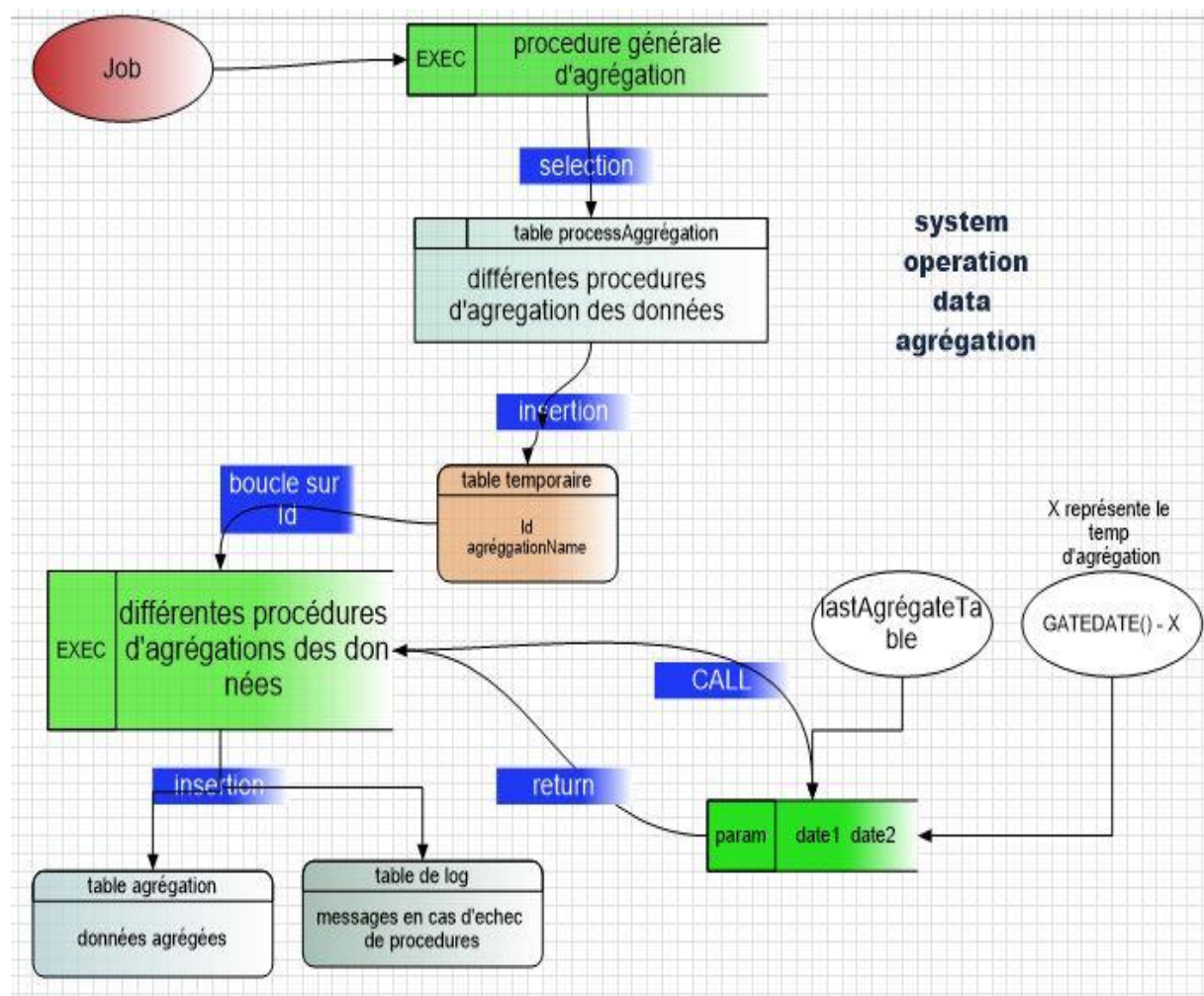


Figure 10 : système d'agrégation de données

VII.3.2 Description

En programmation informatique, l'agrégation permet de définir une entité liée à plusieurs entités de classe différentes. Ainsi, le schéma présenté ci-dessus permet de décrire notre système d'agrégation de données collectées. Pour s'y faire, nous avons dans le système un travail qui exécute une procédure générale. Cette procédure fait une sélection dans la table de processus d'agrégation toutes les procédures d'agrégation ainsi que leurs fréquences correspondantes

pour insertion dans une table temporaire. La fréquence ici indique le moment d'exécution du processus d'agrégation (heure, jour, mois, année). Une fois les données insérées dans la table temporaire, une boucle est effectuée sur les différentes lignes de la table pour exécuter les procédures qui y sont stockées. Ces procédures ayant comme but final d'agréger les données stockées dans la table des métriques. Une fois les procédures exécutées, les données récoltées sont insérées dans la table d'agrégation aux fins de générations des rapports¹. Parallèlement, il y'a une table journal qui est mise en place pour recevoir les messages d'erreurs lorsqu'une procédure ne s'est pas exécutée. En ce qui concerne ces procédures, elles s'exécutent à un moment bien précis et prend par conséquent deux paramètres. Le premier concerne la dernière date d'agrégation. Cette date provient d'une table stockant par une mise à jour la date maximale de la dernière agrégation. Le deuxième concerne le moment d'agrégation qui est la date courante (GETDATE) à laquelle on retire le moment d'agrégation (wait). Ainsi une agrégation est faite par évaluation à une date supérieure au premier paramètre et inférieure au second paramètre.

VII.4 Script générique d'installation

En programmation, la généricité (ou programmation générique), consiste à définir des algorithmes identiques opérant sur des données de types différents. On définit de cette façon des procédures ou des types entiers génériques. Il s'agit ici d'une forme de polymorphisme consistant à fournir une interface unique à des entités pouvant avoir différentes configurations. Ainsi le système se voulant modulaire, il est donc question de mettre en place un script d'installation pouvant être exécuté sur n'importe quel serveur. Pour s'y faire, les différentes étapes du script d'installation sont :

- Création de la base de données : il s'agit de la base de données qui contiendra toutes les informations relatives au système de surveillance.
- Création des différents schémas pour la base de données : un schéma confère à la structure de la base de données une certaine modularité et tout objet de cette base ne saurait se passer de lui. Il s'agit entre autres des tables, vue, procédures, et fonctions
- Création des différentes tables : il s'agit d'un ensemble organisé dont les colonnes permettront de stocker les catégories d'informations récoltées.
- Création des différentes procédures : il s'agit d'un ensemble d'instructions SQL précompilées, stockées dans une base de données et exécutées sur demande par le SGBD qui manipule la base de données.
- Insertion des données : le système étant automatique, il y'a des procédures stockées qui récoltent les informations et les inserts de manière automatique dans les différentes tables. De cette manière, plus besoin d'encoder manuellement les données dans différentes tables car le système se charge de le faire.
- Création des Job : l'agent SQL server possède un onglet JOB (travail) permettant d'automatiser plusieurs tâches de manières planifiées. Ainsi toutes les procédures générales seront planifiées dans plusieurs jobs de l'agent SQL.

¹ Cette notion de rapport ne fera pas partir de ce projet

VII.5 Déploiement des données vers le serveur central

Une fois le système mis en place dans les différents serveurs avec le script de déploiement, vient la phase de rapatriement des informations des différents serveurs vers un serveur central pour gestion centralisée du système à partir d'un serveur. Pour s'y faire, il est utilisé un outil qu'offre la pile SQL server. Il s'agit de SSDT². Il intègre dans sa pile plusieurs services parmi lesquelles SSIS qui dans un projet décisionnel est le premier service à entrer en action car il gère l'extraction et l'enregistrement des données en dehors de l'environnement de production. Il joue donc un rôle d'ETL³. Les données extraites peuvent être alors stockées dans Datawarehouse (entrepôt de données) qui pour moi ici est la base de données se trouvant dans le serveur central. Les bases de données étant identiques, il n'y aura donc pas de transformation de données à faire. Par conséquent juste l'extraction et le chargement. Ainsi pour venir à bout de ceci, les différentes étapes ci-dessous expliqueront le procédé d'extraction et chargement.

Après son installation, l'accès à la pile SSDT se fait à travers l'IDE Visual studio. Une fois devant l'interface, on procède à la création d'un nouveau projet SSIS

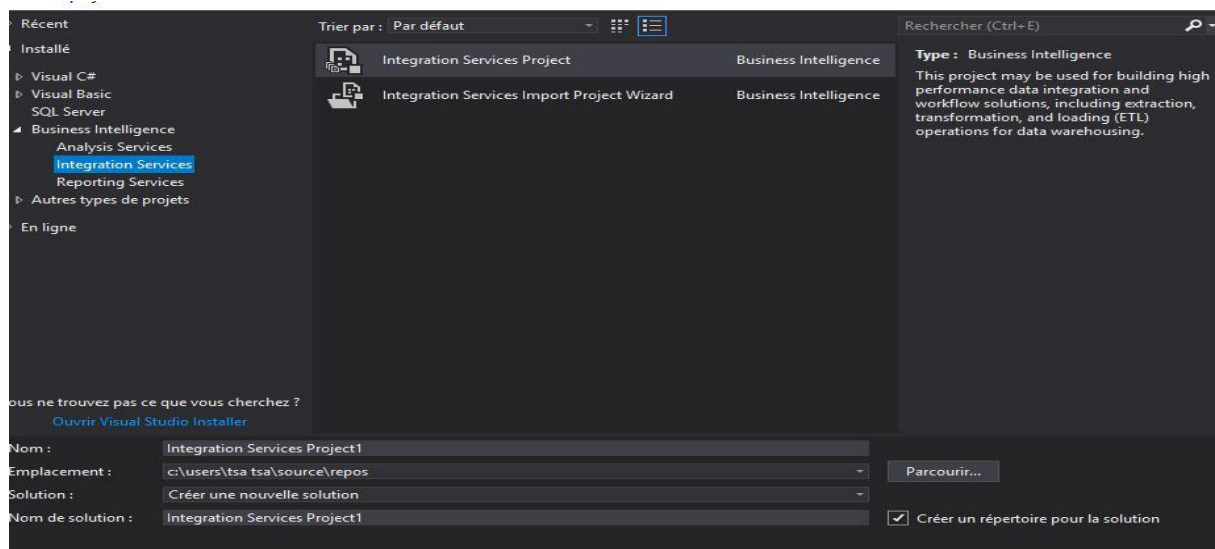


Figure 11 : création d'un projet SSIS

L'interface de SSIS présente 4 rubriques principales pour la création d'un package (variété d'élément qui définissent le flux de travail) dont celles utilisées sont :

- Le flux de contrôle permettant de définir une tâche (unité de travail atomique qui effectue une action). Il y'en a une douzaine dont celles utilisées sont :
 - ✚ La tâche de flux de données qui encapsule le moteur de flux de données chargé de déplacer des données entre les sources et les destinations et permet de modifier les données au cours de leur déplacement. Comme le montre la figure ci-dessous, le package présente plusieurs tâches de flux de données car plusieurs tables doivent être déplacé d'une base de données source à destination d'une base données ayant les mêmes structures de tables.
 - ✚ La tâche d'exécution de requêtes SQL qui exécute des instructions ou procédures SQL à partir d'un package. Elle est utilisée ici pour mettre à jour les colonnes de synchronisation des tables (passage de NULL à la valeur 1) de telle

² SQL Server Data Tools

³ Extract transform and load

manière qu'à la prochaine exécution, un filtre est effectué pour ne sélectionner que des nouvelles données.

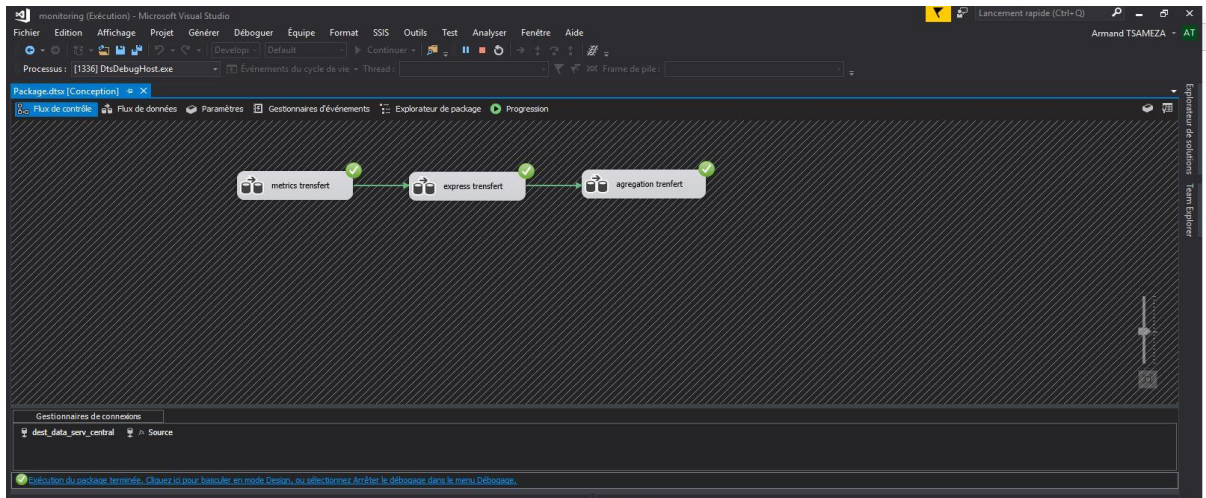


Figure 12 : mise en place d'un flux de contrôle

- Le flux de données. SSIS fournit trois types de flux de données à savoir :
 - La source utilisée pour extraire les données de la base de données sources
 - Les transformations modifient les données lors de leur extraction.
 - La destination charge les données vers la base de données de destination.

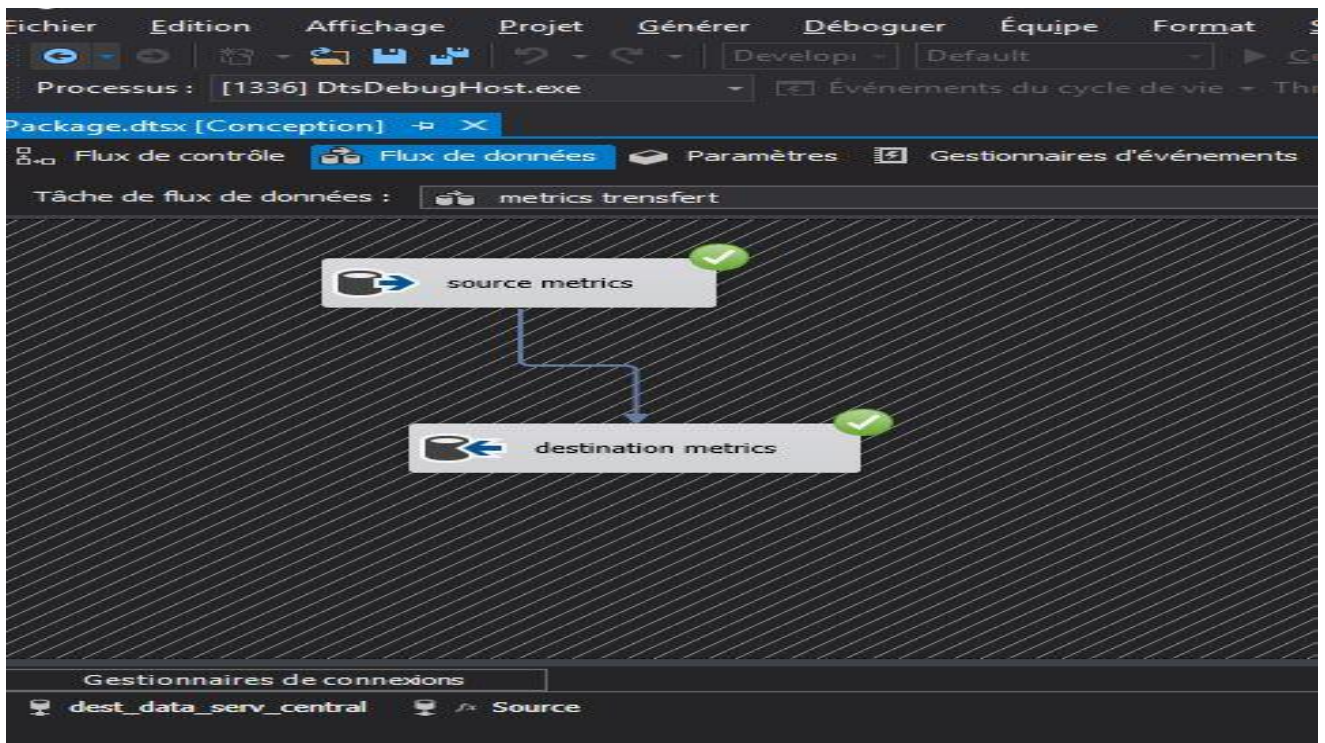


Figure 14 : mise en place des flux de données

Il y'a au sein du package un gestionnaire de connexion qui permet d'établir la connexion entre la source et la destination. Dans ce cas, ayant plusieurs sources et une seule destination, dans le but d'éviter la création d'un package pour chaque serveur, on procède par la création de deux

paramètres au niveau du package. Dans cette mesure, on peut lier à la connexion source ces paramètres.

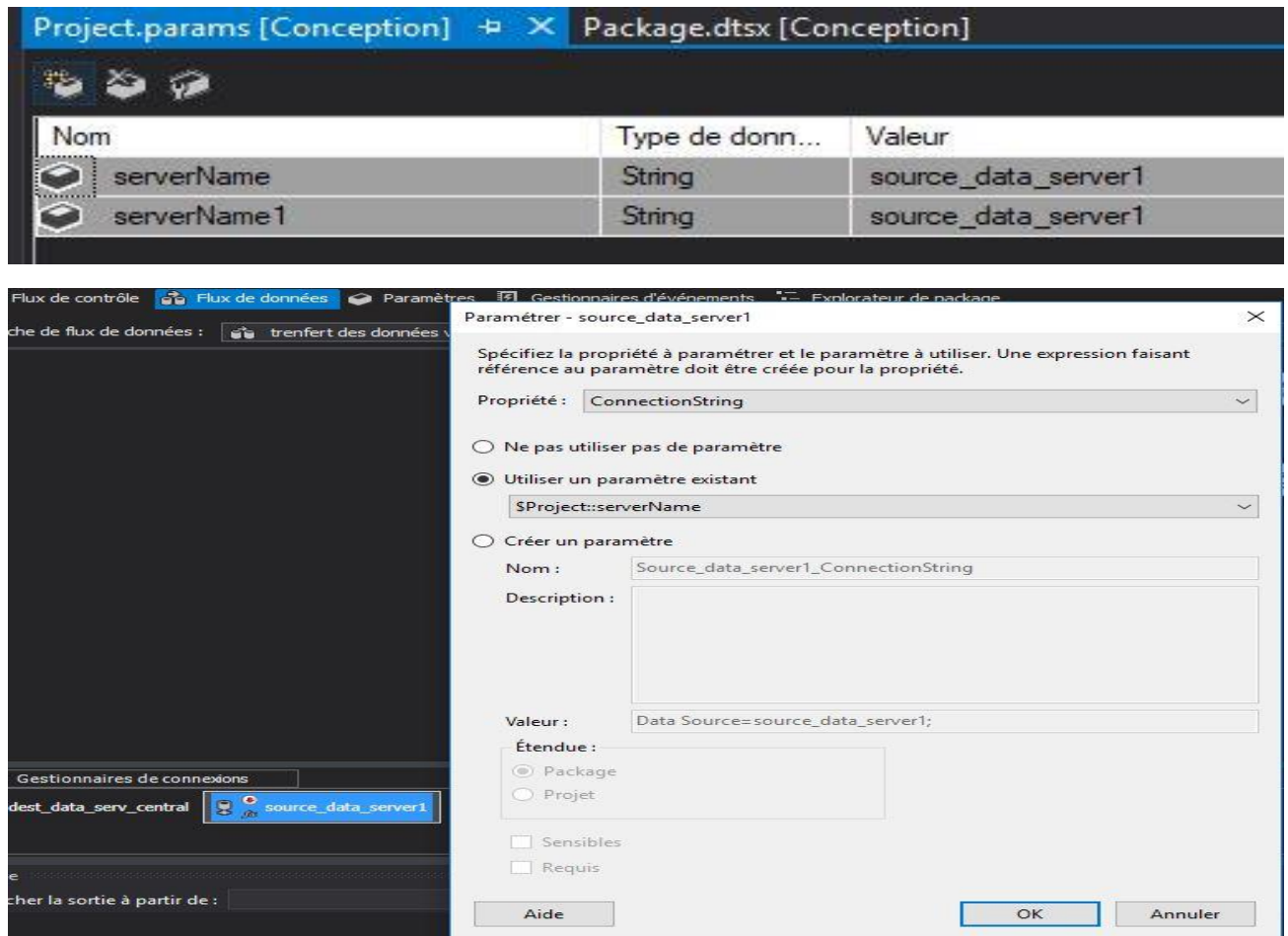


Figure 15 : configuration des paramètres de travail

Ainsi on peut accéder au paramètre du gestionnaire de connexion de la source et paramétrer de telle manière qu'il prenne en considération les différents paramètres lors de l'exécution du package.

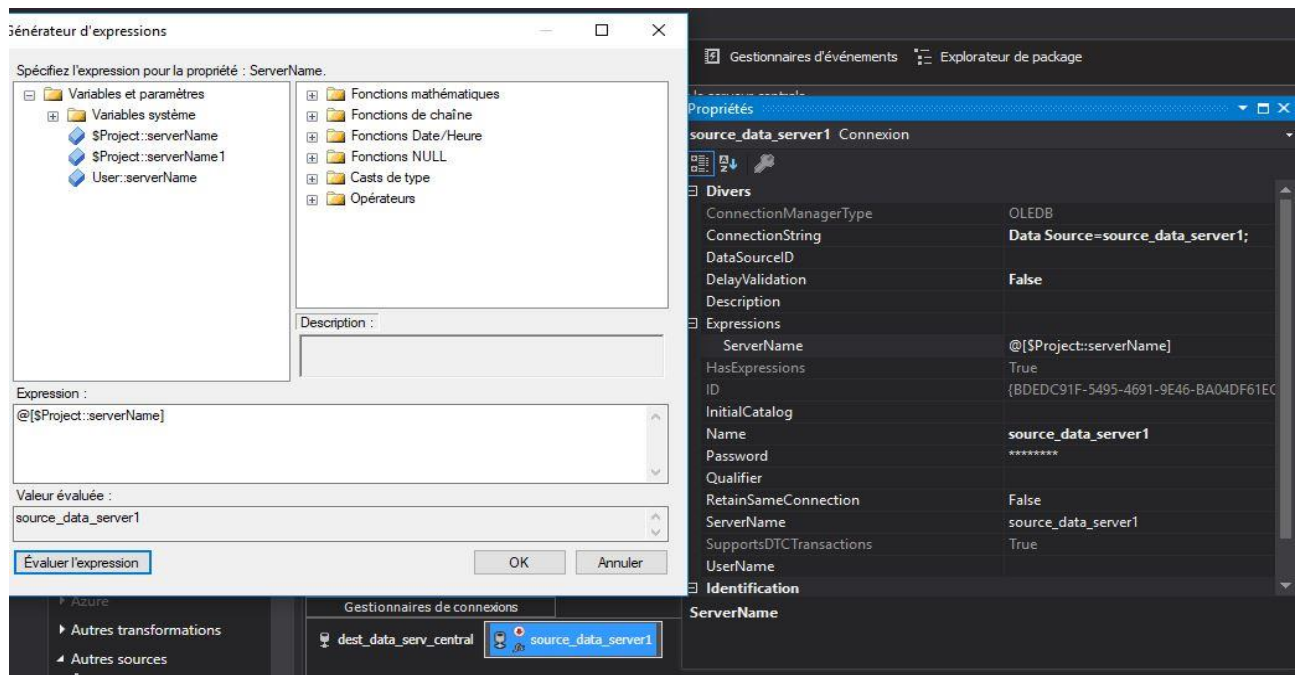


Figure 16 : liaison des paramètres au gestionnaire de connexion source

Une fois ceci fait, on procède au déploiement du package vers SQL server. Mais avant ce déploiement, il faut créer sur SQL server un catalogue de package SSIS qui recevra le package. Ainsi après le déploiement on peut vérifier bien que le package est arrivé sans problème comme le présente la figure ci-dessous.

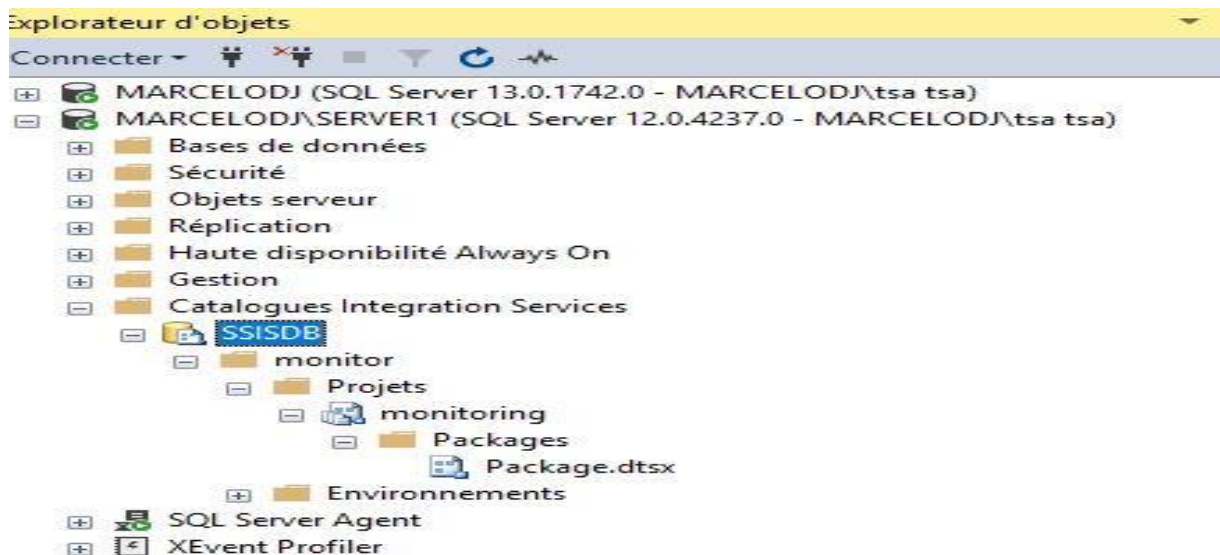


Figure 17 : catalogue SSIS sur SQL server pour déploiement du package SSIS

Cela dit, il faut mapper les paramètres dans SQL server. Pour cela SQL server offre un onglet permettant de créer les environnements. Pour s'y faire, on crée trois environnements dans lesquels seront paramétrés les différentes variables pour y associer les paramètres du package et, associer à la variable une valeur et la mapper vers le paramètre.

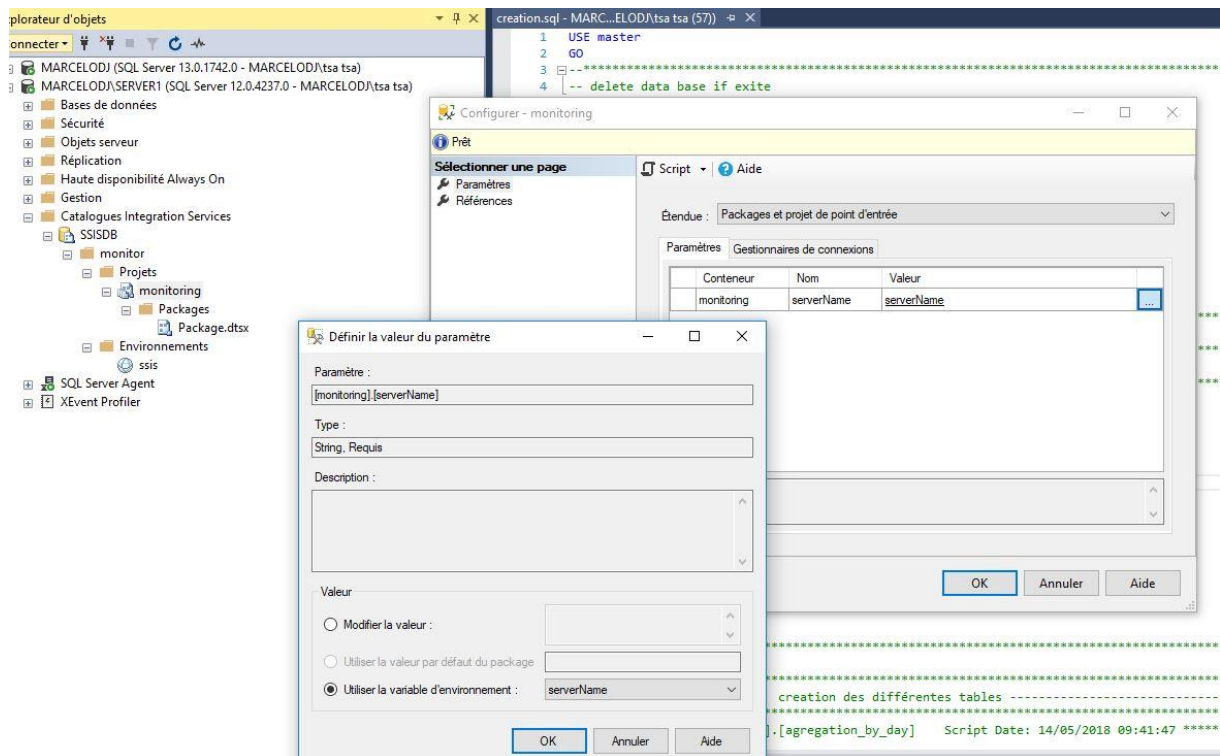


Figure 17 : création des variables SSIS sur SQL server

Dans cette optique avant d'exécuter un package on pourra choisir l'environnement dans lequel l'exécuter et aussitôt la valeur sera mappée en conséquence.

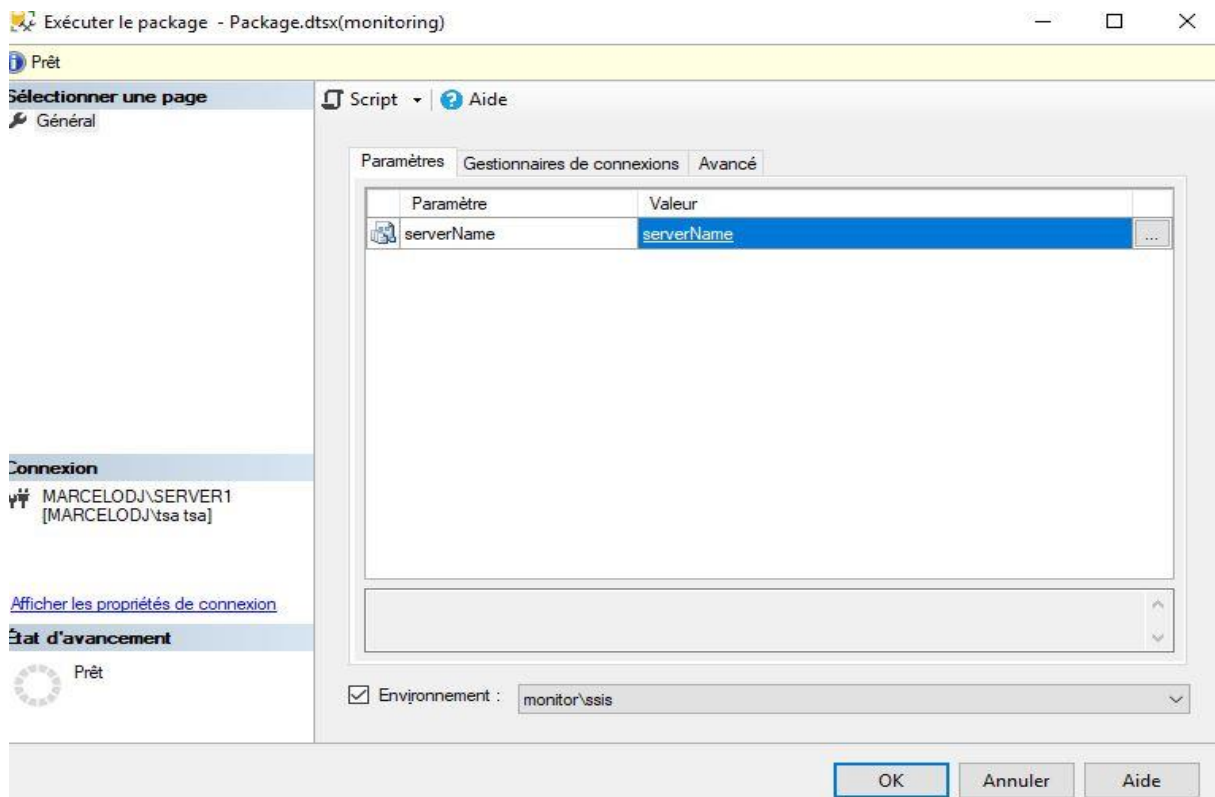


Figure 18 : attribution des valeurs aux paramètres

VIII. Configuration de la gestion des mails

SQL server offre dans son environnement un outil de configuration de la messagerie. Ce dernier permet de créer un profil auquel on peut associer une ou plusieurs adresses email. Ainsi, les emails inscrits pourront recevoir des notifications en fonction des exigences fixées. Ceci pour autant que vous possédiez un serveur SMTP (simple mail transfert Protocol) dans votre environnement de production. Dans l'alternative, il est possible d'utiliser le serveur SMTP de providers mails comme Gmail ou Yahoo. Mais, il faut au préalable changer certains paramètres de configuration au niveau du client mail.

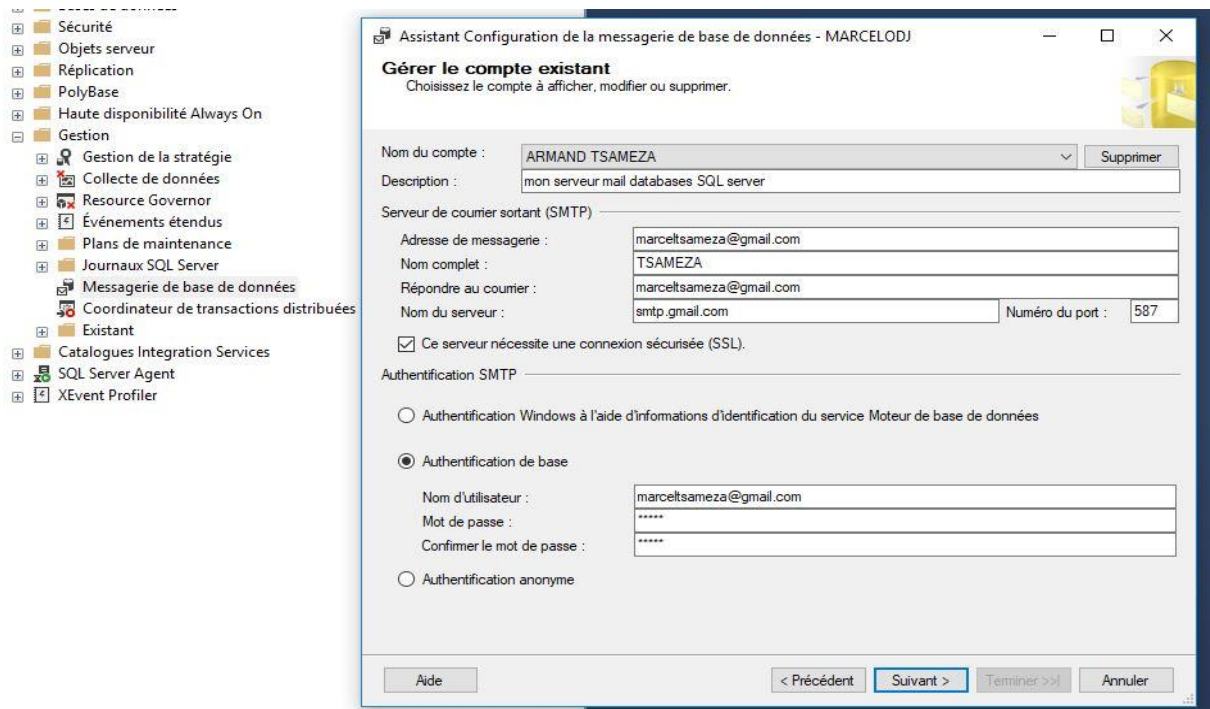


Figure 19 : création du profil et compte email

En ce qui concerne Gmail, ci-dessous l'image présente les paramètres à modifier.

Choose the **Forwarding and POP/IMAP** tab at the top of the page:



In the **IMAP Access** section, make sure that **Enable IMAP** option is selected and click the [Configuration instructions](#) link at the bottom:



Figure 20 : paramétrage du client mail

Une fois la messagerie de bases de données mise en place, il est possible d'être notifié en cas d'échec ou de la réussite d'un job.

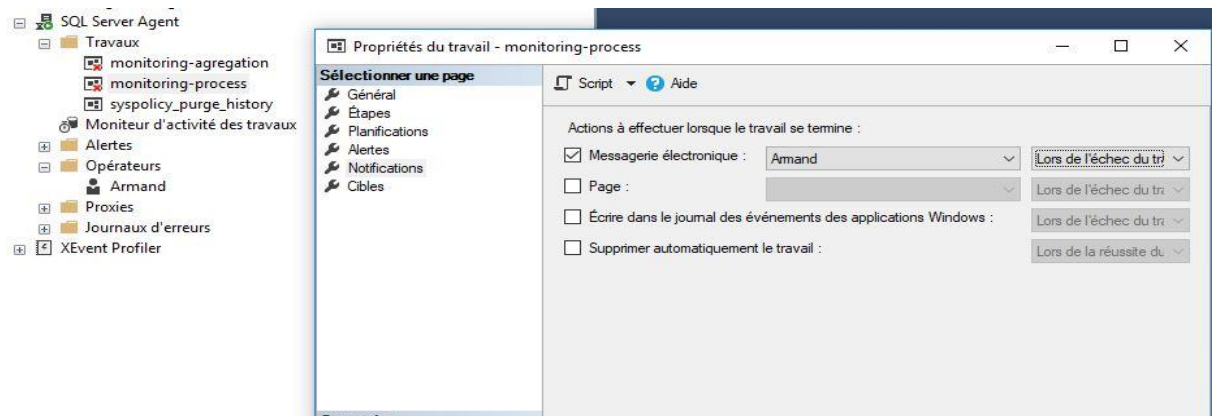


Figure 21 : création d'un opérateur

IX. Les alertes

Dès lors que les différents sous systèmes permettant d'enrichir le système général sont mis en place et, l'espace de gestion des emails configuré, vient la phase du système d'alerte. Il s'agit d'un Message généré et envoyé automatiquement à un ou plusieurs destinataires lorsque certains critères sont réunis. Pour ce faire, on passe par une phase d'analyse afin de fixer en fonction du processus des KPI (indicateurs de performances clés) qui seront mis en relation avec les valeurs métriques, express, et d'agrégation correspondante. Ainsi, le schéma ci-dessous prend le soin de décrire le scénario.

IX.1 Schéma

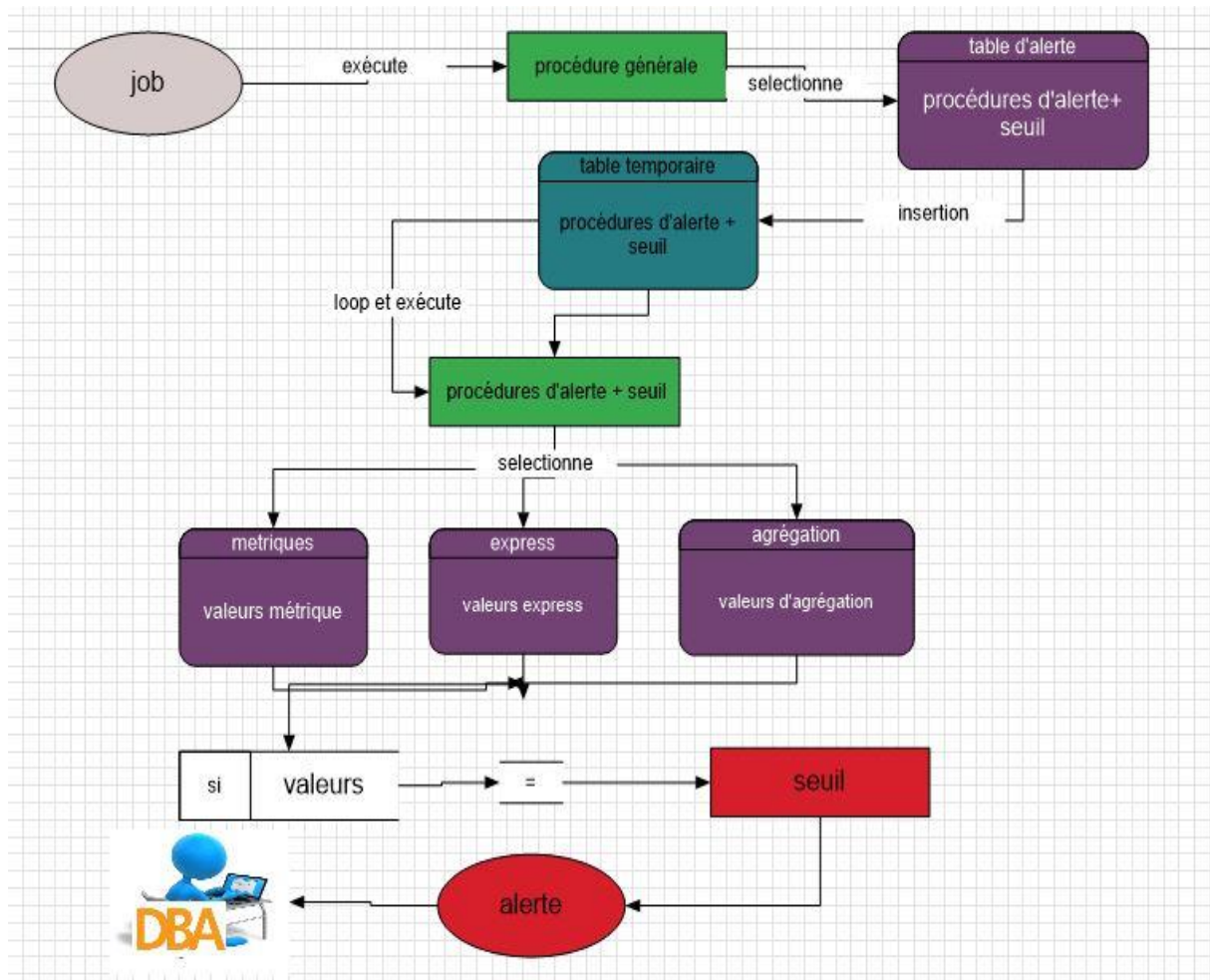


Figure 22 : système de gestion des alertes

IX.2 Description

Au sein du système, il existe un travail de l'agent SQL server qui exécute une procédure générale. Aux fins d'insertion dans une table temporaire, cette procédure fait une sélection dans la table d'alertes sous procédures qui s'y trouvent ainsi que les KPIs correspondants. Une fois dans la table temporaire, elle effectue une boucle sur les différentes lignes de cette table pour exécuter les ces sous procédures qui leur tour, font une sélection dans les tables de métriques, express et d'agrégations pour obtenir les valeurs numériques que contiennent ces tables aux fins de comparaison avec les KPIs passés en paramètre. Ainsi une alerte sera envoyée à destination d'un DBA si ces valeurs correspondent au seuil KPIs passés en paramètre.

X. Difficultés rencontrées et solutions apportées

Durant la réalisation de ce projet, j'ai été confronté à de nombreuses difficultés parmi lesquelles :

- La recherche de l'information dans un environnement que je ne maîtrisais pas et dont le vocabulaire technique m'était inconnu.
- La longue période d'adaptation à certains outils comme Visual studio
- L'exécution des travaux de types SSIS

- La manipulation du chiffrement de base de données qui impliquait une connaissance approfondie des permissions Microsoft

Aux fins d'avoir une idée claire sur ces différents problèmes, une des solutions entreprise était non seulement la rencontre des responsables du projet (client), mais aussi avec mon promoteur. Ces rencontres ont été d'une aide très précieuse.

XI. Conclusion générale

Le sujet portait sur la conception et l'implémentation d'un système de surveillance et d'alerte pour les serveurs Microsoft SQL. Ceci permettant de réduire le temps d'intervention lors de certaines défaillances systèmes. A travers les outils de Microsoft SQL et d'autres outils tiers, ce thème a permis de mettre en place un système d'information(SI) dont les différentes phases sont :

- + La structure de la base de données permettant de recueillir les différentes informations nécessaires au SI ;
- + La mise en place d'un système permettant de collecter les données en direction de la base données précédemment construite
- + La mise en place d'un système d'agrégation des données aux fins d'analyses prévisionnelle ;
- + La mise en place d'un système permettant de sécuriser les données se trouvant dans cette base de données.

Dans l'optique de notifier les personnes chargées de la gestion des serveurs de bases données, un système de mailing a été mis en place permettant en cas d'alerte en rapport avec une défaillance système d'avertir ceux-ci.

En définitive, bien que ceci réponde à la ligne directrice fixée en introduction, il reste encore à ce jour un grand travail à effectuer au niveau du système d'alerte car à présent, un seul processus d'alerte est fonctionnel dans le système.

Aussi, je tiens à exprimer ma gratitude envers les personnes qui m'ont aidé. J'éprouve de même la satisfaction d'avoir pu réaliser un tel projet car, bien qu'avec l'aide des personnes citées plus haut, c'est l'une des premières fois pour moi de suivre un tel projet seul de bout en bout.

XII. Recommandations

Outils

Ces scripts peuvent être exécutés sur n'importe quel environnement (Windows, linux, mac os), pour autant que ces environnements incorporent non seulement le moteur de base de données SQL server avec une ou plusieurs instances, mais aussi les IDS listés plus haut nécessaires à manipulation de ce moteur.

Description du processus

Dans une clé USB se trouve un dossier contenant tous les scripts d'installation du système. Copier ce dossier et coller le dans votre environnement de travail. A l'aide SSMS et de vos informations de login, connectez-vous aux différentes instances. A l'aide de Visual studio, ouvrir le package « **monitoring** » se trouvant dans le dossier que vous avez copié.

Instructions

En cas d'instances multiples, exemple : server1, server2, server3, server4, choisir une instance comme serveur centrale et les autres constitueront les serveurs annexes à surveiller.

Ordre d'exécution

Dans le dossier que vous avez copié, il y'a deux scripts « **création et insertion** ». Sur toutes les instances, exécutez dans un premier temps le script de création, en suite le script d'insertion. Cela permettra de créer non seulement la base de données de données de surveillance ainsi que tous les objets permettant de piloter cette base de données (schémas, tables, procédures, fonctions, droits d'accès), mais aussi par la même occasion insérer les données de manière automatique grâce aux procédures dans les différentes tables.

Dans le serveur central, créer au niveau de l'onglet catalogue intégration service un catalogue de réception du package SSIS. Déployez-le package précédemment ouvert dans Visual studio vers ce catalogue. Exécutez les scripts « **SSIS environnement** », et « **JOB SSIS** ». Cela permettra de créer les environnements d'exécution du package pour les trois instances ainsi qu'un travail planifié de ploiement des données vers le serveur central.

Exécutez sur le serveur central le script « **gestions des mails** » pour la création des profils et comptes mails, ayant au préalable modifié mes informations par les vôtres.

Exécutez dans le serveur central le script « **gestion des alertes** ». Cela mettra en place le processus d'alerte.

Cela dit, vérifiez que tous fonctionnent en interrogeant les tables métriques, express et agregation_by_day. Dans la même foulé, regardez votre boîte mail si vous recevez d'éventuelle alertes en rapport avec les informations fournis par vos servers et bases de données.

NB : la base de données étant chiffrée, penser à faire une sauvegarde de vos clés de chiffrements dans un endroit sécurisé. Servez-vous du script « backup and restore keys » se trouvant dans le dossier fourni.

XIII. Ressources

<https://www.technig.com/20-best-sql-server-monitoring-tools/>

https://www.dynatrace.com/availability-and-performance/sql-server-monitoring/?utm_source=google&utm_medium=cpc&utm_term=%2Bmonitoring%20%2Bsql%20%2Bserver&utm_campaign=benelux-database-monitoring&gclid=Cj0KCQiA5t7UBRDaARIsAOReQtj8YIUmYTGUgenf7YRMx9BOPq6gBYzvQTZMlzC7-4_cw8c8xHkfK14aAopqEALw_wcB

[https://technet.microsoft.com/en-us/library/bb630354\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/bb630354(v=sql.105).aspx)

<https://docs.microsoft.com/fr-fr/sql/relational-databases/security/securing-sql-server>

<https://dataedo.com/blog/creating-database-er-diagram-with-microsoft-sql-server-management-studio-ssms>

<https://www.red-gate.com/simple-talk/sql/sql-development/encrypting-sql-server-transparent-data-encryption-tde/>

<https://www.red-gate.com/simple-talk/sql/sql-development/encrypting-sql-server-using-encryption-hierarchy-protect-column-data/>

<https://dataedo.com/blog/creating-database-er-diagram-with-microsoft-sql-server-management-studio-ssms>

<https://support.office.com/fr-fr/article/cr%C3%A9er-une-table-des-mati%C3%A8res-dans-word-882e8564-0edb-435e-84b5-1d8552ccf0c>

<https://www.sqlshack.com/configure-database-mail-sql-server>

<https://www.codeproject.com/Tips/803195/Foreach-File-Enumerator-in-SSIS>

<https://www.sqlshack.com/parameterizing-database-connection-sql-server-integration-services>

<https://www.mssqltips.com/sqlservertip/4810/setup-environment-variables-in-sql-server-integration-services-2016>

XIV. Annexes

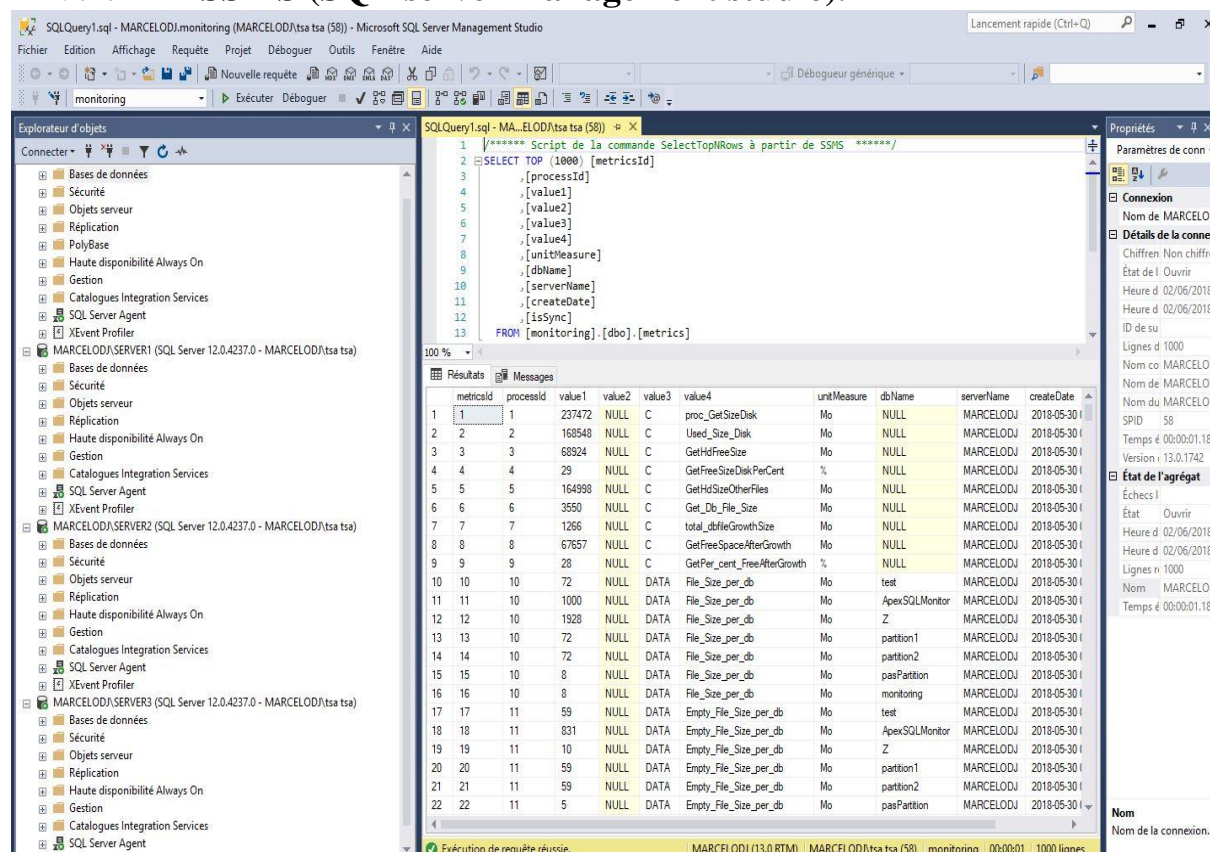
Comme mentionné plus haut, cette partie reprend l'ensemble des images des logiciels utilisés pour le système d'information ainsi que le code source nécessaire à son installation.

XIV.1 Outil d'administration



XIV.2 Les IDE (environnement de développement intégré)

XIV.2.1 SSMS (SQL server management studio).



The screenshot displays the Microsoft SQL Server Management Studio (SSMS) interface. The left pane shows the 'Explorateur d'objets' (Object Explorer) with a tree view of the database structure. The central pane shows a SQL query being executed, and the bottom pane displays the results of the query in a table format.

SQL Query:

```
1 /***** Script de la commande SelectTopNRows à partir de SSMS *****/
2 SELECT TOP (1000) [metricsId]
3     ,[processId]
4     ,[value1]
5     ,[value2]
6     ,[value3]
7     ,[value4]
8     ,[unitMeasure]
9     ,[dbName]
10    ,[serverName]
11    ,[createDate]
12    ,[isSync]
13 FROM [monitoring].[dbo].[metrics]
```

Query Results Table:

metricsId	processId	value1	value2	value3	value4	unitMeasure	dbName	serverName	createDate
1	1	237472	NULL	C	proc_GetSizeDisk	Mo	NULL	MARCELODJ	2018-05-30
2	2	168548	NULL	C	Used_Size_Disk	Mo	NULL	MARCELODJ	2018-05-30
3	3	68924	NULL	C	GetHdFreeSize	Mo	NULL	MARCELODJ	2018-05-30
4	4	29	NULL	C	GetFreeSizeDiskPerCent	%	NULL	MARCELODJ	2018-05-30
5	5	164998	NULL	C	GetHdSizeOtherFiles	Mo	NULL	MARCELODJ	2018-05-30
6	6	3550	NULL	C	Get_Db_File_Size	Mo	NULL	MARCELODJ	2018-05-30
7	7	1266	NULL	C	total_dbfileGrowthSize	Mo	NULL	MARCELODJ	2018-05-30
8	8	67657	NULL	C	GetFreeSpaceAfterGrowth	Mo	NULL	MARCELODJ	2018-05-30
9	9	28	NULL	C	GetPer_cent_FreeAfterGrowth	%	NULL	MARCELODJ	2018-05-30
10	10	72	NULL	DATA	File_Size_per_db	Mo	test	MARCELODJ	2018-05-30
11	11	1000	NULL	DATA	File_Size_per_db	Mo	ApexSQLMonitor	MARCELODJ	2018-05-30
12	12	1928	NULL	DATA	File_Size_per_db	Mo	Z	MARCELODJ	2018-05-30
13	13	72	NULL	DATA	File_Size_per_db	Mo	partition1	MARCELODJ	2018-05-30
14	14	72	NULL	DATA	File_Size_per_db	Mo	partition2	MARCELODJ	2018-05-30
15	15	8	NULL	DATA	File_Size_per_db	Mo	pasPartition	MARCELODJ	2018-05-30
16	16	8	NULL	DATA	File_Size_per_db	Mo	monitoring	MARCELODJ	2018-05-30
17	17	59	NULL	DATA	Empty_File_Size_per_db	Mo	test	MARCELODJ	2018-05-30
18	18	831	NULL	DATA	Empty_File_Size_per_db	Mo	ApexSQLMonitor	MARCELODJ	2018-05-30
19	19	11	NULL	DATA	Empty_File_Size_per_db	Mo	Z	MARCELODJ	2018-05-30
20	20	11	NULL	DATA	Empty_File_Size_per_db	Mo	partition1	MARCELODJ	2018-05-30
21	21	59	NULL	DATA	Empty_File_Size_per_db	Mo	partition2	MARCELODJ	2018-05-30
22	22	11	5	NULL	DATA	Empty_File_Size_per_db	pasPartition	MARCELODJ	2018-05-30

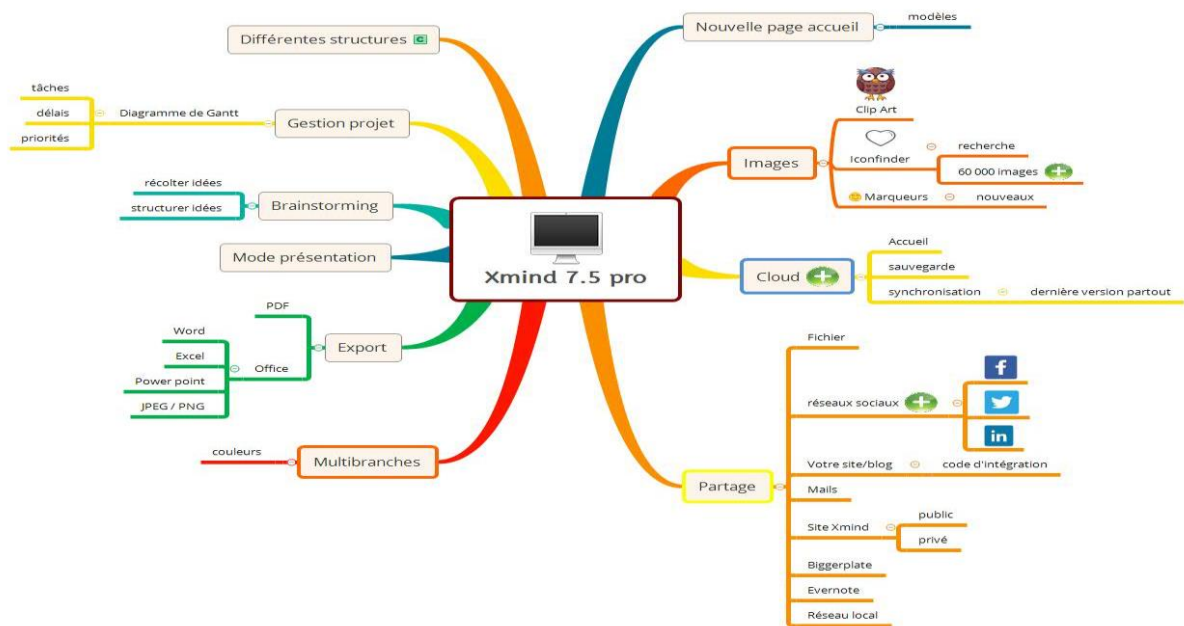
XIV.2.2 Visual studio.



XIV.3 Les différents logiciels utilisés pour organiser cette tâche

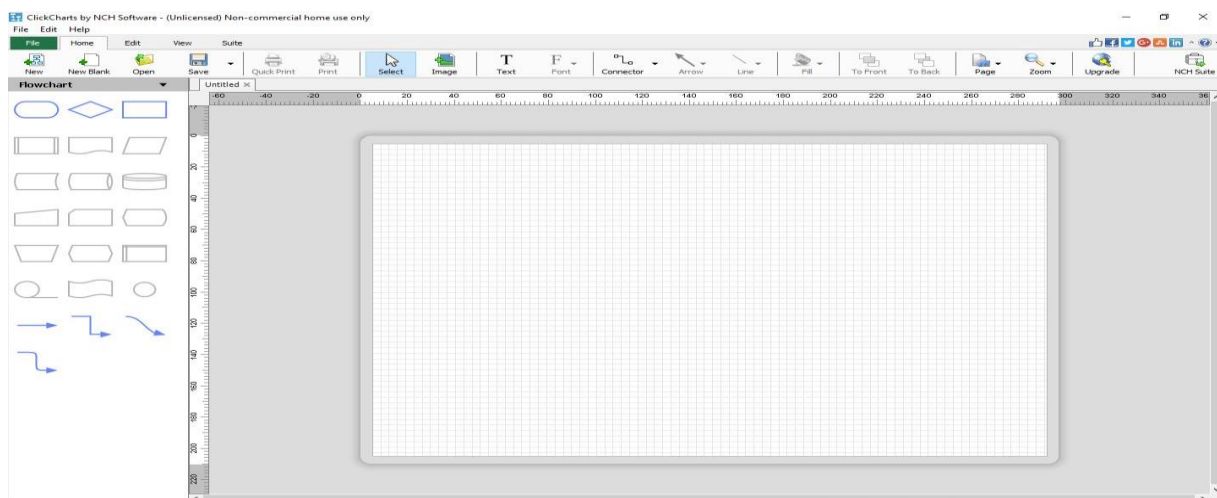
XIV.3.1 XMIND

Pour l'organisation et la représentation de connaissances.

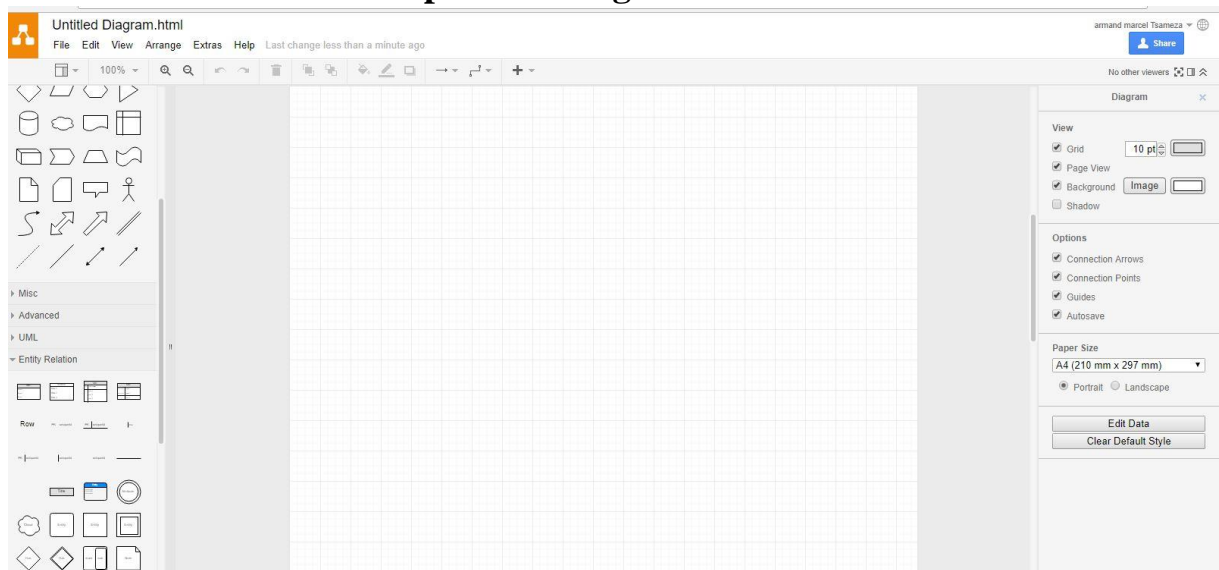


XIV.3.2 Clickcharts

Logiciel convivial de création d'organigramme et de diagramme



XIV.3.3 Draw io conception de diagramme



XIV.4 Code source

Cette partie reprend les différents scripts nécessaires pour le système d’alerte. Il est soumis dans une clé USB.