COMMENT CHOISIR SA SOLUTION DÉCISIONNELLE

Partie 2 : Des modèles à l'analyse

Microsoft

GUSS

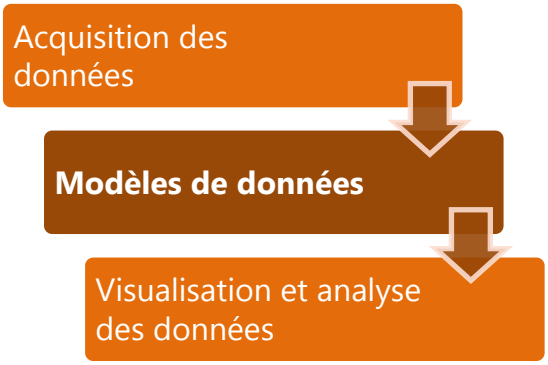
Copyright

Table des matières

Préambule

Un livre blanc en 3 parties

L'ambition de ce livre blanc est de traiter de l'ensemble des aspects d'une solution décisionnelle autour des outils ad hoc de l'offre Microsoft.



Il est découpé en 3 parties : l'acquisition des données, les modèles de données et enfin la visualisation et l'analyse des données.

Cette deuxième partie du livre blanc est intitulée « Des modèles à l'analyse » : Elle abordera toutes les problématiques liées à la création de la modélisation de données.

Audience

Ce livre blanc s'adresse particulièrement à des chefs ou directeurs de projets, architectes et responsables informatiques, ou des décideurs métiers qui souhaitent mettre en œuvre une solution décisionnelle.

Ce livre blanc est destiné à un public souhaitant avoir une vision complète des briques constituantes de la plate-forme décisionnelle de Microsoft. Si vous envisagez de démarrer ou de transformer un projet de Business Intelligence (BI), appuyez-vous sur lui pour vous aiguiller dans vos choix d'outils et d'architectures.

Comment lire ce livre blanc

Ce livre n'est ni un cours, ni une formation sur le décisionnel, ni un guide d'implémentation pratique. Il décrit les éléments constituants une solution décisionnelle, qu'ils soient obligatoires ou optionnels, les usages associés et des pistes pour leur mise en œuvre.

Ce document a été rédigé de manière collaborative par des experts des technologies Microsoft, spécialistes des solutions décisionnelles, appartenant à différentes sociétés de conseil et indépendantes de l'éditeur. Les rédacteurs sont des consultants expérimentés qui implémentent, conseillent et auditent quotidiennement des projets BI.

Toutefois, même si les sujets ont été débattus entre les rédacteurs, il subsiste forcément un biais propre à l'expérience de chaque auteur. Comme avec tout contenu éditorial, le lecteur sera juge et se fera son propre avis sur la base des éléments qui lui seront apportés par les auteurs.

Nota : l'actualité autour de la plate-forme décisionnelle chez Microsoft évolue très vite, les informations présentées ici correspondent à la situation au **31 mars 2015.**

Les auteurs

Jean-Pierre Riehl

Responsable Data & Business Intelligence chez AZEO. Architecte, consultant, expert, chef de projet, développeur, formateur, manageur, MVP, leader du GUSS, mais surtout passionné par les données et la Self-Service BI.

http://blog.djeepy1.net @djeepy1

Romain Casteres

Premier Field Engineer (PFE) SQL / BI chez Microsoft dans l'équipe Data Platform, Big Data et Biztalk. Romain était MVP SQL Server, il est certifié MCSE Data Platform et Business Intelligence ainsi que Hortonworks Data Plateform Developer.

http://pulsweb.fr @PulsWeb

Philippe Geiger

Consultant certifié (MCITP, MCSE), formateur certifié (MCT) et speaker (JSS, SQLSat, Techdays), Philippe travaille actuellement chez Neos-SDI en Alsace où il accompagne aussi les professionnels de l'IT que les développeurs, mais également les utilisateurs de la BI.

http://blog.pgeiger.net/ @PGeiger

Arnaud Voisin

Consultant expert pour le compte de DCube, spécialisé sur la partie décisionnelle, certifié (MCITP, MCSE), formateur certifié (MCT), Arnaud intervient aussi bien sur la partie audit, que conseil ou sur la réalisation.

http://arnaudvoisin.blogspot.fr @ArnaudVoisinSQL

La communauté SQL Server, ce sont tous les acteurs qui travaillent de près ou de loin avec SQL Server, qu'ils soient développeurs, formateurs, architectes, consultants, DBA, etc.

Le GUSS est une association loi de 1901, dirigée par le Board, composé de 9 personnes qui sont des professionnels reconnus sur SQL Server. Le GUSS fédère la communauté autour d'échanges réguliers et rassemble tous ceux qui souhaitent apprendre, partager ou tout simplement échanger sur SQL Server.

http://guss.pro @GUSS_FRANCE

C'est à ce titre que nous avons rédigé pour vous ce livre blanc.

Introduction

La modélisation des données permet aux professionnels de la Business Intelligence d'organiser des données disparates dans **un modèle analytique** prenant en charge de manière efficace les besoins de Reporting et d'analyse de l'entreprise.

Le passage de données brutes vers des données stratégiques se fait en plusieurs étapes. La première partie de ce livre blanc abordait **l'intégration des données**, dans la présente partie, c'est les modèles à des fins d'analyse qui sont détaillées : Les outils d'intégration permettent de récupérer les données où qu'elles soient dans le but d'alimenter **les modèles de données**. Enfin, la dernière partie du livre blanc sera consacrée à **la restitution des données**, issues desdits modèles, aux utilisateurs finaux.

Cette partie du livre blanc permettra au lecteur de mieux cerner la création des modèles d'analyse : Dans un premier temps, il est important de bien comprendre l'état de **l'art de la modélisation**. Ensuite, le lecteur pourra découvrir le **stockage des données** que ce soit sur les serveurs de l'entreprise ou via des serveurs dédiés (on-premise alors appelés « **appliances** » ou dans le cloud). Le **Big Data** est une plate-forme moderne permettant d'analyser les données en créant un modèle en mode différé. Le chapitre **BISM (Business Intelligence Semantic Model)** est consacré à la présentation des modèles de données permettant aux utilisateurs finaux de bénéficier d'une couche d'abstraction propre à leur métier. Enfin, de ces modèles, il est possible d'aller plus loin dans l'analyse de données dans le but de **prédire de nouvelles valeurs ou de classifier les données**.

État de l'art de la modélisation

La modélisation est la clé de voûte dans un système décisionnel. Elle permet aussi bien de garantir de bonnes **performances** que de résoudre des **problématiques fonctionnelles** telles que l'historisation ou la gestion des stocks par exemple.

Que la BI de l'entreprise soit composée d'un entrepôt de données ou directement connectée à des bases opérationnelles, la **qualité du modèle de données** sera primordiale aussi bien au niveau du stockage du moteur SQL qu'au niveau du modèle sémantique (BISM).

Alors, pourquoi modéliser ?

Dès l'invention[1] des bases de données relationnelles (ou **OLTP**), celles-ci ont présenté, entre autres, l'avantage de retrouver rapidement une donnée particulière grâce aux index. Toutefois, ces mêmes bases de données ne sont pas adaptées aux traitements de masse, nécessaires aux calculs des données de synthèse d'un système décisionnel, où l'objet est de ramener **un grand nombre de données et de les agréger** entre elles (somme, moyenne, etc.). Comme les systèmes de gestion de bases de données relationnelles ne sont pas adaptés, il était donc nécessaire de créer de nouveaux systèmes ad hoc : C'est ainsi que les systèmes **OLAP**[2] ont fait leur apparition.

Les systèmes OLAP pré-calculent (ou du moins facilitent largement les agrégations) toutes les grandeurs clefs (aussi appelées **mesures**) selon des axes d'analyse (ou **dimensions**). Pour en faciliter l'analyse, des outils comme les **tableaux croisés dynamiques** sous Excel[3] sont largement utilisés : Il est important que l'utilisateur qui accède ainsi aux données synthétiques, en comprenne le sens[4].

Le principe de cette démarche est de **créer un modèle**, c'est-à-dire une **représentation symbolique de la réalité** décrite par le système d'information en utilisant la terminologie conforme au vocabulaire habituel de l'entreprise et de ses utilisateurs.

La méthodologie

La méthodologie décisionnelle a été théorisée par les 2 pères fondateurs de la BI moderne **Ralph Kimball et Bill Inmon**. Chacun d'eux à une approche différente :

- Celle de Ralph Kimball est plus agile : elle est dite « bottom-up ». Dans cette approche, il s'agit de construire des **petits entrepôts spécialisés** par besoins

[1] Pour donner une référence dans le temps, le langage SQL a été créé en 1974.
[2] Les systèmes OLAP ont été conceptualisés par Edgar Frank Codd en 1996.
3 Plus d'un milliard de licences existe dans le monde, soit autant d'utilisateurs.
4 Par exemple, les titres de colonnes dans les bases de données traditionnelles, où seuls les développeurs en comprennent le sens tant ils font l'usage d'abréviations, sont à proscrire.

fonctionnels : On les appelle datamart. Cette méthode a l'avantage de pouvoir livrer des lots séparés plus rapidement. Bien que les bases soient autonomes, elles peuvent malgré tout être le composant d'un ensemble plus large appelé entrepôt de données.

- L'approche d'Inmon se veut **le référentiel global de l'entreprise**. Ce type de projet permet d'avoir la modélisation la plus exhaustive et la moins redondante. Cette approche est plus adaptée au cycle en V.

Le point commun des deux approches est que l'on construit le modèle de donnée à partir du **besoin des utilisateurs** et non pas en fonction des données à disposition.

Ces deux méthodologies ont été créées pour traiter des **données structurées et relationnelles**. Avec ces méthodologies, il faut faire des choix lors de la conception (granularité, indicateurs, attributs, historisation, …). On ne stocke que ce dont on a besoin.

Ainsi, la BI, dans sa définition habituelle, est l'art **d'exploiter et d'aider à l'analyse des données** grâce à des indicateurs, parfois objectivés (KPI) et ventilés selon des axes d'analyses. En d'autres termes, en BI, on cherche à **mesurer des faits à partir de dimensions**. Des modélisations particulièrement adaptées à cet usage existent :

- **Le schéma en étoile** : Constitué de tables de fait connecté à des dimensions totalement dénormalisées[5]. L'avantage est d'apporter l'attribut au plus près de la donnée et d'économiser les jointures entre les tables. Ceci à un effet négatif sur le stockage, mais limité grâce à la compression et au coût sans cesse diminuant du stockage.

- **Le schéma en flocon** : Plus normalisé que le schéma en étoile. On cherche à sortir les attributs dans des dimensions satellites pour les attributs à faible cardinalité. Le stockage est plus optimisé, mais le modèle est plus complexe à appréhender et plus de jointures sont requises pour aller chercher la donnée.

Aujourd'hui, grâce à l'apparition de système de stockage distribué, comme dans Hadoop, il est possible d'aborder les choses différemment et de stocker les données (structurées ou non) et de repousser l'analyse à plus tard. La structure est créée au moment de l'analyse. Ce nouveau concept est appelé **Data Lake**.

La gestion de l'historisation

Le point le plus complexe dans un système décisionnel est la **gestion de l'historisation**.

[5] La dénormalisation est une technique de modélisation de bases de données relationnelles ayant pour but de faciliter l'analyse de données (agrégation par axe d'analyse). Ainsi contrairement aux formes normales, tous les attributs sont placés dans des dimensions sans tenir compte de la redondance et toutes les lignes sont répétées dans des tables de faits afin de pouvoir agréger facilement au niveau de la granularité escompté.

Ce qu'il faut comprendre par historisation, ce n'est pas uniquement l'historisation des indicateurs au fil du temps, mais surtout **l'historisation des référentiels ou des valeurs d'attributs** des dimensions dans le temps (exemple : historisation du statut d'un client au fil du temps).

L'historisation est complexe, car elle est génératrice de volumétrie et complexifie le chargement. Les modélisations possibles sont :

- **Les dimensions à variations lentes** (Slowly Changing Dimensions ou SCD) qui se déclinent en 3 types et s'appliquent au niveau attribut (un ou plusieurs SCD est applicable à la même dimension) :
    - **Le type 1** qui signifie « pas d'historisation ». La valeur courante s'applique à tout l'historique.
    - **Le type 2** qui historise tous les **changements pour un attribut**. Ce mode est à utiliser pour des dimensions à volumétrie limitée et à nombre de changements très limité. Si le nombre de changements devait être trop important, il faudrait envisager un autre mode d'historisation.
    - **Le type 3** qui permet d'historiser un nombre déterminé de changements grâce à un *stockage par colonne*.
- **L'historisation attribut à variations rapides** est plus complexe. Dans le cas d'un changement rapide, le SCD2 ne serait plus applicable. La technique pour gérer ce genre de cas est de sortir les attributs concernés dans une ou plusieurs **mini-dimensions** et de gérer l'historisation grâce à une table de fait.

Stockage

La modélisation des données est une couche logique d'organisation des données sur un **espace physique**. Le stockage et sa gestion sont liés à la modélisation. Pour de **bonnes performances**, il est important de maîtriser et optimiser l'accès aux données. De nombreuses fonctionnalités existent dans SQL Server pour optimiser le stockage, quelques-unes d'entre-elles seront détaillées ici.

Partitionnement

Le partitionnement permet d'optimiser un modèle de données (OLAP, Tabulaire et SQL). Le principe est simple, on **découpe les données** selon un critère (année, géographie, magasin, etc.) afin d'en faciliter la manipulation.

En SQL, le partitionnement offre une flexibilité permettant des usages très performants. Les cas d'usage sont nombreux :

- Répartition des données de tables et index sur **plusieurs axes disques** ;
- Affectation aux partitions des **options de compression** différentes en fonction des besoins ;
- Sécurisation des données en **verrouillant les données historiques** sur des partitions associées à d'anciens filegroups ;
- Suppression du contenu d'une partition afin d'éviter des opérations transactionnelles coûteuses comme le DELETE, grâce à une instruction à **journalisation minimum** (TRUNCATE) et à la **commutation de partition** (SWITCH partition).

Sur SSAS, les partitions constituent également un outil puissant et souple permettant de gérer les cubes, en particulier s'ils sont volumineux. Par exemple, un cube qui contient des informations sur les ventes peut réserver **une partition par mois**. Seule la partition de mois en cours devra être calculée si des informations sont ajoutées au cube, le traitement d'une plus **petite quantité de données réduit le temps de traitement** et améliore les performances. La création de cette partition peut être automatisée dans le cadre des procédures de chargement de l'entrepôt de données et de traitement du cube.

Le mode de stockage de chaque partition peut être **configuré indépendamment** des autres partitions. Les partitions peuvent être stockées différemment : emplacement, mode de stockage, mise en cache proactive et conception d'agrégation. Cette souplesse permet de créer des **stratégies de stockage de cube** appropriées à vos besoins.

Il est possible de modifier la stratégie de stockage des cubes multidimensionnels et tabulaires après leurs déploiements.

ColumnStore Index

A partir de SQL Server 2012, est apparu un nouveau mode d'indexation, **xVelocity**, qui

regroupe et stocke les données de **chaque colonne**, puis joint l'ensemble des colonnes pour remplir l'index tout entier. Cela diffère des index classiques qui regroupent et stockent les données de **chaque ligne**, puis joignent l'ensemble des lignes.

Les avantages :

- **Meilleure compression,** car les données d'une même colonne sont stockées au sein de la même page de donnée ;
- **Lecture plus rapide,** car moins d'octets à lire en raison de la compression ;
- Requêtes de type GROUP BY plus rapides, car toutes les données de la colonne sont proches et **l'index optimise les agrégations**.

Inconvénient sur SQL Server 2012 : les tables sous-jacentes sont en lecture seule.

Dans SQL Server 2014, une évolution du moteur de stockage permet de surmonter cette limitation avec l'arrivée des **Clustered ColumnStore Index**. Cet index permet la **mise à jour** de tables indexées en colonne grâce à une zone de stockage temporaire le « deltastore ». Il est à noter que le Clustered ColumnStore index n'autorise pas la création d'autres index sur la même table.

D'autres **limitations** existent :

- Une colonne calculée ne peut pas faire partie d'un index ColumnStore ;
- De nombreux types de colonnes ne sont pas supportés pour un index ColumnStore (VARCHAR(MAX), VARBINARY, XML, CLR, …) ;
- Un index ColumnStore ne peut pas être créé sur une vue indexée.

In-memory

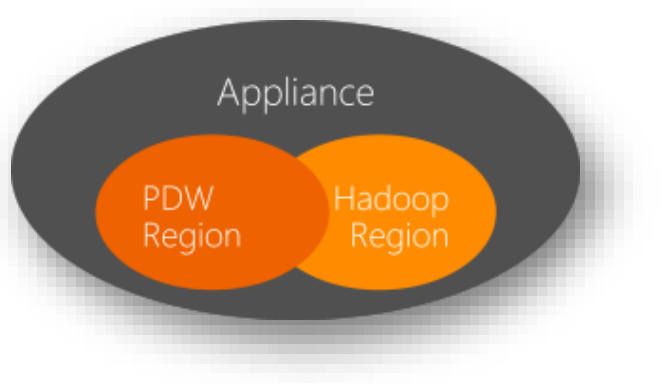
Microsoft propose via **SSAS Tabular** un moyen de stockage performant, permettant de bénéficier d'un nombre d'I/O accru et d'améliorer le temps d'accès aux données. Ce **mode tabulaire** utilise le moteur xVelocity, de manière similaire aux index ColumnStore : ce point sera détaillé plus loin dans le chapitre Big Data & NoSQL.

Avec SQL Server 2014, il est possible de stocker des tables en mémoire. Il faut noter que cette fonctionnalité appelée In-Memory OLTP n'est pas adapté aux applications de type décisionnelles.

.

Liste des outils utilisant xVelocity (in-memory)

Appliances



Analytics Platform System (APS) combine le meilleur de la base de données SQL Server et les technologies Hadoop[6] dans une Appliance.

L'Appliance contient à la fois Microsoft SQL Server 2012 Parallel Data Warehouse[7] (**PDW**) et **HDInsight**. Simple à déployer, APS est livrée préconfigurée avec les logiciels, les matériels et les composants réseaux dans un souci d'optimisation des performances. Il est aussi conçu pour évoluer suivant les besoins des utilisateurs.

APS est découpé **en régions** et en « **workloads** » : Une région est un conteneur logique permettant de cloisonner la charge de travail, la sécurité, les services. Un « workload » est un cluster de traitement de données.

La région **PDW** contient :

- L'infrastructure de l'appliance ;
- Le moteur de base de données distribué PDW ;
- Hadoop Data Integration (Polybase) ;
- Console de gestion.

La région **Hadoop** contient :

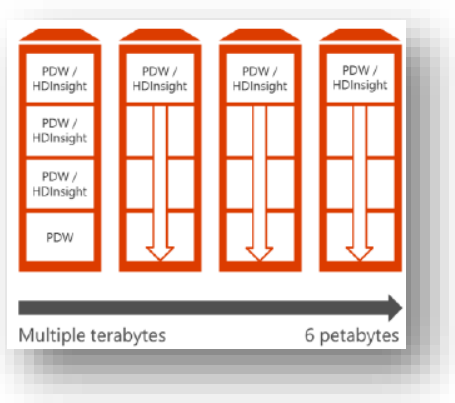
- HDInsight ;
- Un tableau de bord pour les développeurs.

Contrairement aux solutions SMP (Symmetric Multi-Processing) qui exécutent des requêtes sur un seul serveur partageant CPU, mémoire et les opérations sur disques nécessitant l'achat de serveur de plus en plus puissant, les infrastructures MPP (**Massively Parallel Processing**) tel que APS permettent de commencer avec un **petit rack** et de **l'agrandir** pour soutenir les charges de travail des entrepôts de données jusqu'à **6 péta-octets.**

[6] Les notions de Hadoop et de HDInsight citées dans ce chapitre seront développées dans le chapitre suivant relatif au Big Data.

[7] PDW est l'édition MPP de SQL Server, les requêtes s'exécutent en parallèle sur les différents nœuds (serveurs) de l'Appliance.

Parallel Data Warehouse (PDW) offre une puissance de traitements de données exceptionnelle basée sur une architecture dite MPP. La force de MPP est de pouvoir



distribuer la charge sur plusieurs unités de calcul en parallèle et de profiter des IO d'un très grand nombre de disques bon marché (JBOD ou « Just a Bunch Of Disks »). L'architecture MPP présente l'avantage de pouvoir faire du « Scale Out », donc d'améliorer les performances globales et la capacité de stockage par **l'ajout de simples unités de calcul supplémentaire**. Là où pour augmenter les capacités de traitement avec une architecture SMP, il fallait changer le hard (« Scale up »), le MPP apporte de la flexibilité et permet de commencer par un **investissement initial plus faible ajustable** par la suite aux besoins du projet.

La dernière édition arrive avec son lot de nouveautés :

- **Polybase** : C'est la fonctionnalité phare du produit. Elle permet d'interconnecter des données issues de HDFS (Hadoop) et de les présenter de manière transparente sous forme de table. Polybase permet de requêter/lier les données du moteur relationnel avec les données non structurées stockées dans Hadoop.
- **ColumnStore[8] updatable** : Sortie dans l'édition MPP de SQL Server avant même celle de l'édition classique, l'index permet plus de souplesse que la précédente version tout en garantissant un niveau de performance optimal. Cette version permet le tri physique de données (Clustered) et permet aussi la modification de données tout en gardant un haut niveau de compression des données.

Avant de se lancer dans l'aventure, il est important de bien **comprendre l'architecture** MPP. La migration d'une application taillée pour SMP ne se migre pas sans une réflexion préalable.

Voici quelques limitations :

- Pas d'index uniques ;
- Pas de vues ;
- La taille maximale d'une ligne 8060 Bytes ;
- dbo est le seul schéma supporté ;
- Certains types ne sont pas supportés (XML, text, CLR UDT, timestamp, …) ;
- Les identités et les contraintes default non supportées ;
- Limitation sur les collations ;

[8] Les Columstrore Index sont détaillés dans le chapitre relatif au stockage.

- etc.

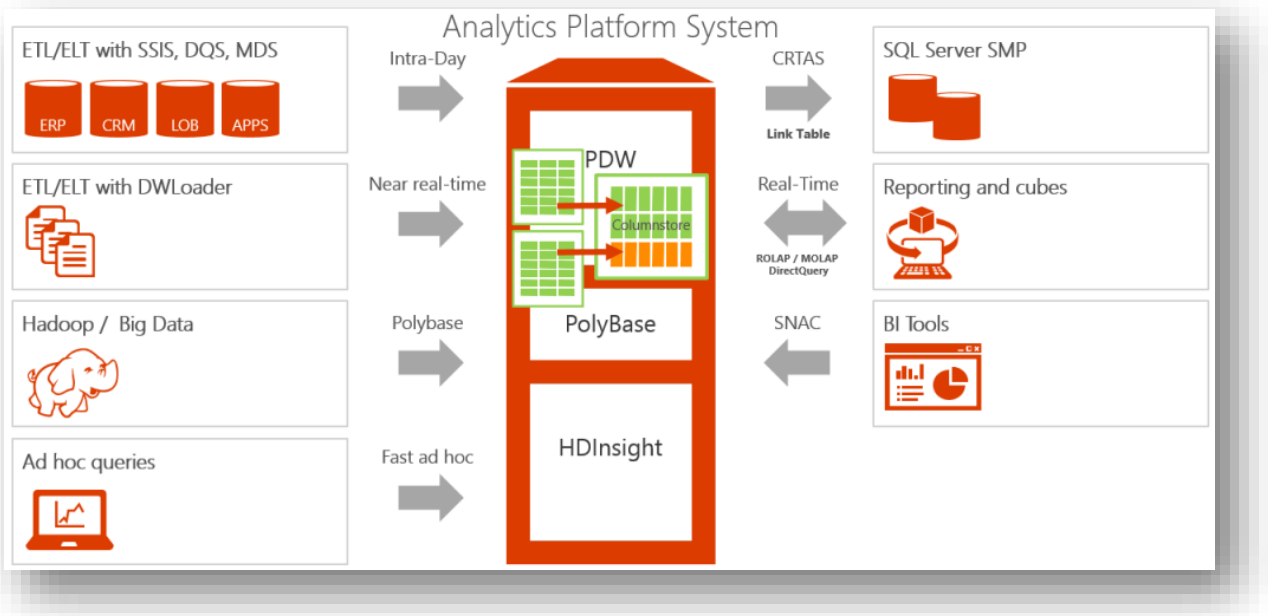
Toutes ces contraintes obligent à revoir le design des applications existantes, cela nécessitera de prévoir **une réécriture** de tout ou partie de l'entrepôt en prenant en compte les limitations et les avantages de l'architecture MPP.

Au-delà des contraintes, PDW ouvre la porte à de nouveaux cas d'usage de BI temps réel :

- **Architecture multidimensionnelle ROLAP :** Ce mode permet de profiter de la puissance du ColumnStore Index et de la puissance de la distribution, ce qui s'avère une réponse performante aux problématiques des groupes de mesures DistinctCount[9].
- **Architecture tabulaire DirectQuery :** ce mode offre de très bons temps de réponse pour des volumétries dépassant les capacités en mémoire de la machine. Comme pour l'architecture multidimensionnelle ROLAP, DirectQuery pour Tabular permet de déporter la charge sur PDW et donc de profiter des performances exceptionnelles en lecture.

Remarque : Prévoir un serveur SSAS au sein du même réseau Infinyband pour éviter les goulets d'étranglement.

En utilisant les index en mémoire ColumnStore Clustered pour stocker des données sur le disque, PDW atteint des taux élevés de **compression** de données qui permettent **d'économiser les coûts** de stockage et **d'améliorer les performances** des requêtes.



[9] Pour plus de détail, le lecteur pourra se reporter à ce livre blanc : http://bit.ly/1eT6cpB.

Big Data & NoSQL

Le **volume de données** à traiter ne cesse de croître, les types de données générés par les applications et les réseaux sociaux entre autres sont **de moins en moins structurés**. De plus, les bases de données relationnelles peuvent atteindre **leurs limites** : la volumétrie, la fréquence avec laquelle les données sont générées et/ou accédées, …

Les bases de données NoSQL (« **non uniquement SQL** ») telles que HBase permettent d'outrepasser ces limites en se basant sur un **système distribué** comme celui d'Hadoop. Elles stockent généralement les données sous forme de clef/valeur qui sont **non-relationnelle, distribuées, horizontalement évolutive et sans schéma**.

La complexité d'une base de données relationnelle limite l'évolutivité du stockage de données, mais il est très facile d'interroger les données par le biais d'un moteur SQL. Les systèmes de base de données NoSQL ont les caractéristiques opposées : une **évolutivité illimitée,** mais des **restrictions au niveau du requêtage**. Le défi du Big Data est de faciliter l'interrogation de données.

Microsoft offre les solutions de Big Data suivantes :

- **Azure HDInsight** permet de déployer et d'approvisionner des clusters Apache Hadoop dans le cloud, en fournissant une infrastructure logicielle conçue pour gérer et analyser les données.
- **L'Appliance APS** (voir le paragraphe précédent) offre la possibilité d'interroger et de combiner des données relationnelles et non relationnelles avec Polybase.
- **Excel** permet d'analyser des données provenant d'Hadoop en toute simplicité.
- **Power BI** combine des données internes et externes et répond facilement à de nouveaux types de questions.

Le Big Data et la modélisation ?

Contrairement aux SGBD traditionnels, dans Hadoop, **aucun schéma** n'est appliqué lorsque les données sont intégrées. Le schéma sera défini lors de **l'exploitation des données**. L'idée est d'intégrer les données dans leurs formats d'origine avec le maximum d'information sans se préoccuper de leurs tailles ni de leurs formats (principe du **Data Lake**).

En Big Data, il est généralement préconisé de dénormaliser les données pour pallier aux diminutions de performance dues aux **opérations de jointures**.

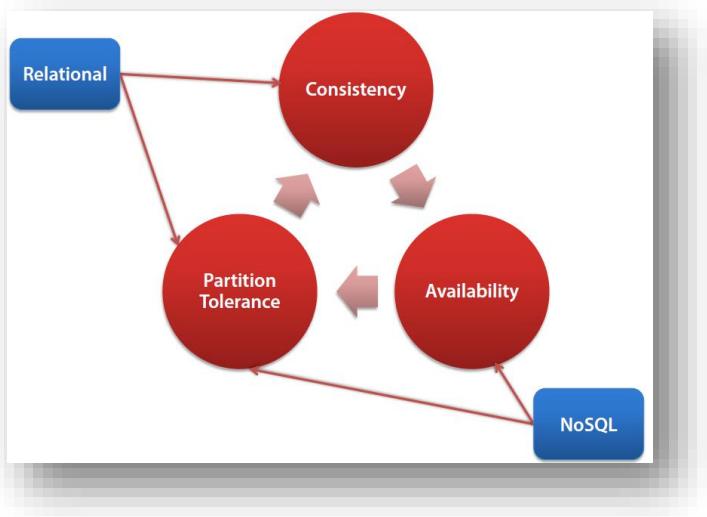
D'autre part, du fait que les enregistrements sont stockés sous forme de Clé–Valeur, ceux-ci peuvent être **partitionnés sur plusieurs serveurs**. Il est donc impératif de bien sélectionner cette clé afin que la distribution des valeurs permette une **parallélisation des traitements**. À noter qu'il n'y a pas de bonnes ou de mauvaises techniques, il faut d'abord réfléchir à quelles sont les informations que l'on va vouloir récupérer.

En conclusion, la modélisation Big Data doit être guidée avant tout par une très **bonne analyse des informations à traiter** et doit être corrélée avec les besoins d'analyses qui seront effectués.

NoSQL

SQL est l'acronyme de Structured Query Language. Il est connu depuis longtemps de tous les acteurs de l'informatique qui ont eu à **rechercher rapidement des informations** dans de bases de données relationnelles (SGBD).

Pourtant, un certain nombre de **limitations importantes** sont apparues au fil des années. Les premiers acteurs à les rencontrer sont les fournisseurs de services en ligne les plus populaires, tels que Yahoo, Google ou plus récemment les acteurs du web social comme Facebook, Twitter ou LinkedIn. Le constat était simple : les SGBD relationnels ne sont pas adaptés aux **environnements distribués** requis par les **volumes gigantesques** de données et par les **trafics tout aussi gigantesques** générés par ces acteurs.



Dans un contexte centralisé, les contraintes ACID[10] sont plutôt aisées à garantir. Dans le cas de systèmes distribués, les traitements de données sont répartis entre les différents serveurs. Il devient alors difficile de maintenir **les contraintes ACID** à l'échelle du système distribué entier tout en maintenant des performances correctes.

Le **théorème de CAP** (Consistency Availibility Partition Tolerance) énonce trois grandes propriétés pour les systèmes distribués :

- C comme **Cohérence** : Tous les nœuds du système voient exactement les mêmes données au même moment.
- A comme **Disponibilité** : La perte de nœuds n'empêche pas les survivants de continuer à fonctionner correctement.
- P comme Résistance **au partitionnement** : Aucune panne moins importante qu'une coupure totale du réseau ne doit empêcher le système de répondre correctement.

[10] Les propriétés ACID (atomicité, cohérence, isolation et durabilité) sont un ensemble de propriétés qui garantissent qu'une transaction liée à des données est exécutée de façon fiable.

Le théorème de CAP stipule qu'il est impossible d'obtenir ces trois propriétés en même temps dans un système distribué et qu'il faut donc en **choisir deux** parmi les trois en fonction de vos exigences.

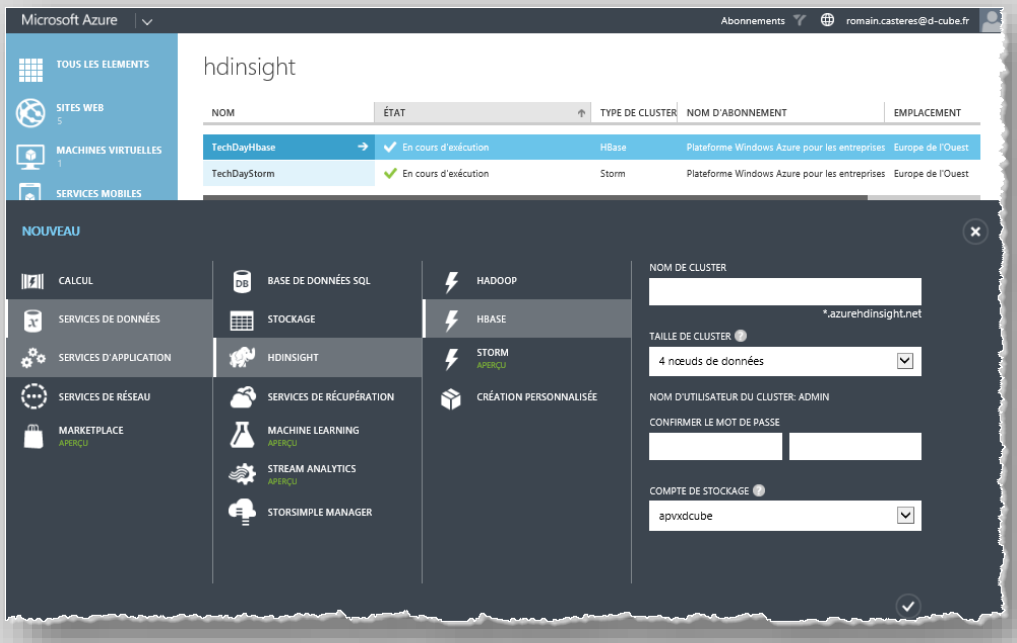
En vue de ces caractéristiques, les moteurs NoSQL **sont adaptés** pour les problématiques suivantes :

- Vélocité des données (croissance rapide) ;
- Nombre de colonnes croissant ;
- Données non structurées ;
- Données hiérarchisées et graphiques.

Toutefois, les moteurs NoSQL sont **moins adaptés** aux besoins suivants :

- OLTP nécessaire (On-Line Transaction Processing) ;
- ACID (Atomicity, Consistency, Isolation, Durability) ;
- Relation complexe entre les données ;
- Requêtes complexes.

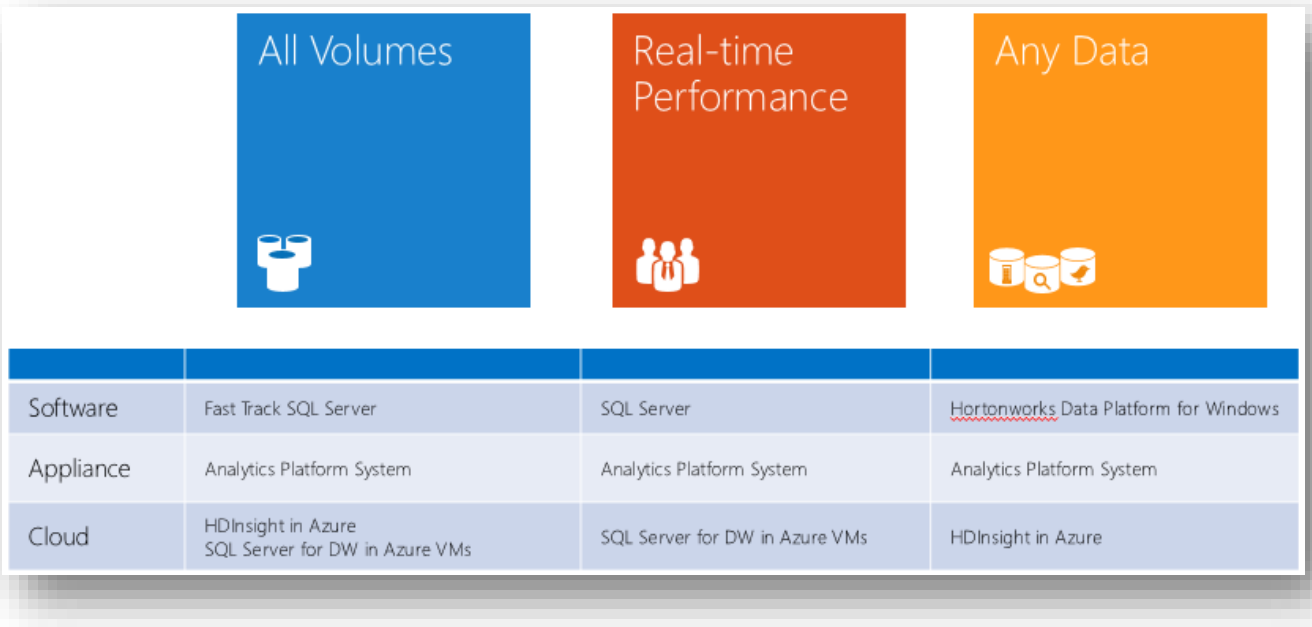
HDInsight inclut Apache **HBase**, une base de données NoSQL en colonnes qui s'exécute sur le système de fichiers distribués Hadoop (HDFS). Les données sont stockées dans un Blob Storage Azure (par défaut) ce qui fournit une faible latence et une élasticité (performances/coût).



HBase est dit **« SPARSE »**, les colonnes NULL ne sont pas stockées et n'occupent aucune place (identique à SQL Server avec l'option « SPARSE » sur les colonnes). De plus, toutes les données sont stockées dans **un tableau d'octets**.

L'implémentation d'un langage comme SQL rend l'utilisation d'outils comme **Hive**[11] plus aisée. Il en est de même pour **Phoenix**, il fournit une couche SQL au-dessus de **HBase** tout en apportant de nouvelles fonctionnalités. À noter qu'il est possible d'utiliser des clients JDBC standards comme **SQuirreL** pour se connecter à Phoenix et ainsi interroger les données HBase ou encore via **SQLLine**, un utilitaire Java en mode console.

Quelle solution en fonction du besoin ?



Software	Fast Track SQL Server	SQL Server	Hortonworks Data Platform for Windows
Appliance	Analytics Platform System	Analytics Platform System	Analytics Platform System
Cloud	HDInsight in Azure SQL Server for DW in Azure VMs	SQL Server for DW in Azure VMs	HDInsight in Azure

11 Apache Hive est une infrastructure d'entrepôt de données intégré au-dessus d'Hadoop. Pour plus de détails sur Hive, le lecteur pourra se reporter première partie de ce livre blanc, où Hive y est détaillé.

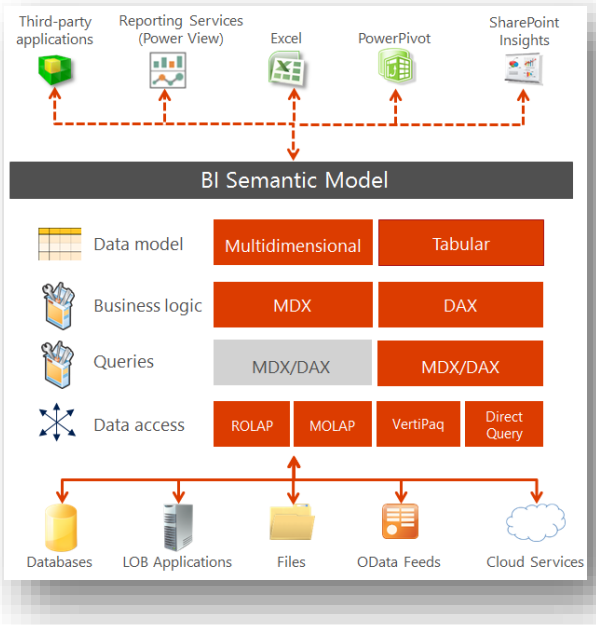
Modèle sémantique (BISM)

SQL Server 2012 a introduit le **modèle sémantique BI (BISM)**, ce modèle offre une vue unifiée des données et permet de **masquer la complexité sous-jacente** aux utilisateurs finaux, indépendamment du mode de stockage utilisé.

Le modèle de données représente la façon dont les différentes tables de faits et de dimensions sont corrélées, mais des fonctionnalités supplémentaires de modélisation peuvent enrichir l'expérience utilisateur :

- **Les perspectives** permettent de définir un sous-ensemble d'un modèle de données pour simplifier l'exploration par les utilisateurs finaux.
- Dans les modèles multidimensionnels, **les traductions** permettent d'afficher la dimension, l'attribut, la mesure, le membre calculé, ainsi que d'autres noms d'objets et valeurs de membres de dimension, dans la langue spécifiée par les paramètres régionaux de l'ordinateur de l'utilisateur. Ainsi, ce dernier affichera les informations du modèle **dans sa langue**.
- Le **Drill Down** permet de naviguer dans les hiérarchies et d'afficher les résultats des agrégats à différents niveaux de détail.
- Dans les modèles multidimensionnels, **les actions** permettent aux utilisateurs finaux d'effectuer **des Drill-Through**[12], d'appeler **un rapport** Reporting Services, d'accéder à **une URL**, ou encore d'initier **une opération externe** en fonction du contexte de la cellule où l'action se produit.
- Dans les modèles multidimensionnels, il est possible d'autoriser les utilisateurs à **écrire des données dans le cube** (« writeback »).

BISM offre plus qu'une couche d'abstraction, il s'articule sur **3 briques différentes** : un modèle de données, une logique fonctionnelle et un accès aux données.



Modèle de données

BISM rassemble les modèles de données **tabulaires** et **multidimensionnels** dans une même plate-forme. Ces deux modèles ont le même objectif : fournir une **couche sémantique** au-dessus des données sous-jacentes, permettant aux utilisateurs finaux d'explorer les données. Ces deux modèles font partie d'Analysis Services, mais ce

[12] La technique des Drill-Through correspond à lister toutes les données détaillées qui participent au résultat de l'agrégation.

sont **deux produits bien différents**. Ils peuvent coexister sur la même machine, mais ils nécessitent tous deux leurs propres instances.

Le modèle multidimensionnel n'est plus à présenter, basé sur un **schéma en étoile ou en flocon** ; il reste le modèle le plus adéquat pour des cubes à **relations complexes** et de **grosses volumétries**.

Le modèle tabulaire permet quant à lui de créer plus rapidement un cube de complexité moins importante. Ce sont des bases de données **en mémoire** utilisant des algorithmes de **compression** et des traitements des **requêtes multithread** (moteur xVelocity[13]). Sur des données de **granularité fine**, le modèle tabulaire répond mieux que le multidimensionnel.

Voici un petit comparatif des deux modèles :

	Modèle multidimensionnel	Modèle tabulaire
Technologie	Technologie mature	Technologie d'avenir
Volume	Technologie évolutive capable de gérer de très grands volumes de données	Limité à la RAM disponible (= projet de taille moyenne)
Modélisation	Capable de faire face modélisation complexe et aux calculs exigeants	Manque quelques calculs avancés disponibles en MDX
Mise en œuvre	Complexe	Simple
Modélisation internationale (langue, devise)	Oui	Non
Many-to-many	Oui	Non (sauf avec DAX)
Outil de développement	SQL Server Data Tools (SSDT-BI)	SQL Server Data Tools (SSDT-BI) / Power Pivot pour Excel
Langages	MDX et DMX	DAX pour les calculs et MDX pour les requêtes
Write-back	Oui	Non
What If	Oui	Non
Sécurité	Permission par rôle/par cellule	Permission par rôle/par ligne
Compression	X3	X10
Granularité fine	+	++
Actions	Oui	Non

[13] Pour plus de détail sur le moteur xVelocity : http://bit.ly/1xCwLDU

Pour résumer, la modélisation tabulaire propose une expérience de **modélisation simplifiée**, susceptible de satisfaire **la plupart des besoins** de Reporting et d'analyse peu exigeants. Cependant, si le besoin est d'embarquer une **logique-métier**, **une sécurité complexe**, ou **de grosses volumétries**, la modélisation multidimensionnelle est probablement mieux adaptée à ces besoins.

Logique métier

Pour enrichir le modèle, il est possible d'utiliser les **langages MDX ou DAX**, d'encapsuler des règles métier, d'ajouter des hiérarchies, des indicateurs de performance (KPI) et des perspectives offrants des vues simplifiées du modèle.

- **Le MDX (Multidimensional Expressions)** a été introduit avec Analysis Services et est devenu le standard de l'industrie BI pour les logiques multidimensionnelles. Le MDX est basé sur les concepts OLAP : des dimensions, des faits et leurs mesures.
- **Le DAX (Data Analysis Expression)** est un langage d'expression basé sur des formules Excel qui a été introduit avec Power Pivot et est basé sur les concepts tabulaires : des tables, des colonnes et des relations. C'est un langage conçu pour les utilisateurs finaux (voire avancés).

Remarque :

- Un cube tabulaire peut être requêté en DAX et en MDX.
- Les tableaux croisés dynamiques Excel génèrent des requêtes MDX.
- Power View génère des requêtes DAX.

Les résultats des requêtes DAX ne sont jamais enregistrés dans le cache, ce qui signifie que les requêtes DAX prendront toujours le même temps d'exécution.

Depuis la mise à jour cumulative SQL Server 2012 Service Pack 1 CU 4[14] ou Service Pack 2, l'ensemble des outils de restitutions peuvent se connecter aux cubes multidimensionnels et tabulaires. Cependant, **Power View dans Excel 2013** ne peut pas encore se connecter à un cube multidimensionnel au moment de l'écriture du livre, seul le **Power View** intégré à **SharePoint** le permet.

Accès aux données et stockage

La couche d'accès aux données est la couche rassemblant des données provenant de sources multiples telles que les bases de données, fichiers plats, OData, RSS et autres. Il existe deux **familles d'accès aux données** : « **Cached** » (pré-agregé) et « **Passthrough** » (Real Time) :

- En mode « **cached** », les données sont lues, traitée et stockée en mode compressé

[14] http://bit.ly/1vASgki

et optimisé, cela permet d'atteindre de très bonnes performances.
- En mode « **Passthrough** », les données proviennent des sources de données sous-jacentes, le système exploite les capacités du système source tout en évitant de dupliquer les données dans le modèle.

Bien qu'ils fassent partie de la même famille d'accès/requêtes aux données, c'est-à-dire « Cached », les modes MOLAP et In-Memory restent bien différents. En effet, le **mode MOLAP** stocke sur disque des données pré-agrégées et de détails alors que le **mode In-Memory** stocke les données de détail en colonne et en mémoire.

Récapitulatif des différents modes de stockage et d'accès aux données :

Modèle multidimensionnel	
MOLAP	Les requêtes sont effectuées sur les agrégations sauvegardées dans la base de données multidimensionnelle, les données de détail sont copiées.
HOLAP	Les agrégations de la partition sont stockées dans une structure multidimensionnelle et les données de détails sont requêtées sur les sources sous-jacentes.
ROLAP	Les agrégations de la partition sont stockées dans des vues indexées dans la base de données relationnelle, aucune copie des données n'est effectuée.

Modèle tabulaire	
In-Memory	Les requêtes sont uniquement effectuées sur le cache.
In-Memory with DirectQuery	Les requêtes sont effectuées sur le cache, sauf indication contraire dans la chaîne de connexion du client.
DirectQuery with In-Memory	Les requêtes sont effectuées sur la source relationnelle, sauf indication contraire dans la chaîne de connexion du client.
DirectQuery	Toutes les requêtes utilisent la source de données relationnelle sous-jacente.

Pour vulgariser, le **modèle tabulaire** lit les données directement à partir de la mémoire cache et tire parti de l'accélération de la requête obtenue à partir des **ColumnStore Index**, tandis que le **modèle multidimensionnel** lit les données pré-agrégées ou des données atomiques à partir du disque selon les **agrégations existantes définies** lors de la conception du cube.

Par défaut, les modes de stockage et de requêtes sont les modes de mise en cache (MOLAP, In-Memory), cependant il est possible de les ajuster en fonction des problématiques :

Problématiques	Modèle multidimensionnel	Modèle tabulaire
Temps réel	ROLAP + CSI[15]	DirectQuery + CSI
Grosse volumétrie	ROLAP / HOLAP	DirectQuery
Sécurité avec exigence de cryptage HIPPA / Besoin d'Audit des requêtes sources	ROLAP	DirectQuery
Source unique et < SQL Server 2005	ROLAP / HOLAP / MOLAP	In-Memory
Performance	MOLAP	In-Memory
Stockage uniquement des métadonnées	ROLAP	DirectQuery
Stockage des données et métadonnée	MOLAP	In-Memory

Le mode DirectQuery introduit **quelques limitations** :

- Seul le DAX est pris en charge donc il n'est pas possible de naviguer à partir d'un tableau croisé dynamique Excel.
- Les calculs de temps ne sont pas pris en charge.
- Les colonnes calculées ne sont pas prises en charge.

[15] CSI : ColumnStore Index.

Datamining et analyse prédictive

L'exploration de données, aussi connue sous les noms de fouille de données, datamining (forage de données) ou encore Extraction de Connaissances à partir de Données (ECD en français, KDD en Anglais), a pour objet l'**extraction d'un savoir ou d'une connaissance** à partir de grandes quantités de données.

Le datamining ne nécessite pas que l'on établisse une hypothèse de départ qu'il s'agira de vérifier. C'est à partir des données que **les corrélations intéressantes** sont déduites, le datamining se situe à la croisée **des statistiques, de l'intelligence artificielle et des bases de données**.

Les données traitées peuvent être issues de **sources multiples et hétérogènes, plus ou moins structurées**. Cela impose de disposer de systèmes performants de préparation ou de manipulation des données.

Les solutions Microsoft

SQL Server Analysis Services offre les fonctionnalités suivantes permettant l'analyse prédictive et la construction de modèles analytiques de recommandation et d'exploration :

- **Sources de données :** Il n'est pas nécessaire de disposer d'un **entrepôt de données** pour faire du datamining. En effet, les données à analyser peuvent provenir de **sources externes** comme des fichiers texte, Excel, etc.
- **Outils clients :** En plus des outils de développement et de conception fournis avec SQL Server, il est possible d'utiliser le complément **Excel** Datamining pour créer, interroger, et analyser les modèles.
- **Algorithmes personnalisables :** En plus de fournir des algorithmes tels que le clustering, les réseaux de neurones, les décisions des arbres, la plate-forme prend en charge l'intégration de vos propres algorithmes et plug-ins.
- **Langages :** Tous les objets d'exploration de données sont entièrement programmables. Le **scripting** est possible grâce aux langages **MDX et XMLA**, la librairie **AMO** ou encore des extensions **PowerShell** pour Analysis Services. Le langage **DMX** (Data Mining Extensions) permet de créer et d'utiliser des modèles d'exploration de données dans Microsoft SQL Server Analysis Services.
- **Sécurité et déploiement :** La sécurité est basée sur les **rôles** d'Analysis Services.

Remarque : seul le **modèle multidimensionnel** supporte la génération d'algorithme de Datamining.

Microsoft SQL Server Analysis Services fournit de base **plusieurs algorithmes** d'exploration de données. Ils peuvent être personnalisés et sont entièrement programmables en utilisant les API fournies, ou les composants d'exploration de données

dans SQL Server Integration Services :

- **Microsoft Naive Bayes :** permet de classifier facilement et rapidement. Utilisation : catégoriser les bons et mauvais clients.
- **Microsoft Decision Trees :** c'est un algorithme de classification et de régression utilisé pour la modélisation prédictive d'attributs discrets et continus. L'algorithme est facile à appréhender. Utilisation : identifier les caractéristiques d'un client.
- **Microsoft Linear Regression :** c'est une variante de l'algorithme Microsoft Decision Trees permettant de trouver des relations entre attributs. Utilisation : déterminer une relation entre deux colonnes continues.
- **Microsoft Neural Network :** l'algorithme associe chaque état possible de l'attribut d'entrée avec chaque état possible de l'attribut prévisible, et il utilise les données d'apprentissage pour calculer les probabilités. Utilisation : prédiction des stocks.
- **Microsoft Logistic Regression :** la régression logistique est une technique statistique connue utilisée pour modéliser les résultats binaires. Utilisation : Exploration et évaluation des facteurs qui contribuent à un résultat comme par exemple rechercher les facteurs qui influencent les clients à se rendre plusieurs fois dans un magasin.
- **Microsoft Clustering :** permet de regrouper des faits similaires ensemble. Utilisation : regroupement de clients par typologie.
- **Microsoft Sequence Clustering :** Trouve les séquences les plus courantes. Utilisation : identifier les tendances de navigation sur les sites internet.
- **Microsoft Time Series :** l'algorithme prédit des valeurs futures à partir de valeurs continues dans le temps. Utilisation : prévision des ventes.
- **Microsoft Association Rules :** l'algorithme génère des recommandations. Utilisation : mettre en relation des produits achetés : par exemple, il existe une corrélation[16] d'achat de couches pour bébés et de bière le samedi après-midi.

Les 9 algorithmes peuvent être catégorisés suivant leurs utilités :

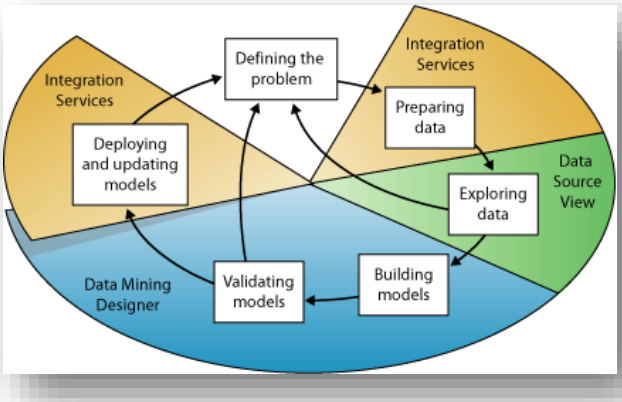
Classification	Estimation	Regroupement	Prévision temporelle	Association
Decision Trees	Decision Trees	Clustering	Time Series	Association
Logistic Regression	Linear Regression			Decision Trees
Naive Bayes	Logistic Regression			
Neural Network	Neural Network			

[16] http://www.zdnet.fr/actualites/datamining-39600371.htm

Construction du modèle

Plusieurs phases sont nécessaires à la construction d'un modèle d'exploration, voici les **six étapes** :

1. Définir le problème
2. Préparation des données
3. Exploration des données
4. Construction du modèle
5. Exploration et validation des modèles
6. Déploiement et mise à jour des modèles



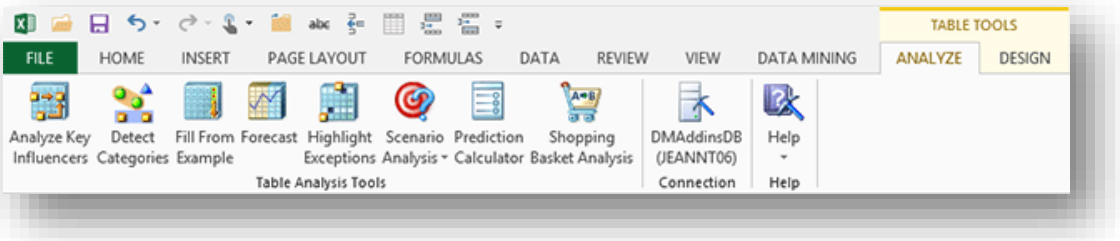
Le processus illustré est cyclique, ce qui signifie que la création d'un modèle d'exploration de données est un **processus dynamique et itératif**. Après avoir exploré les données et constaté que les données sont insuffisantes pour créer les modèles d'exploration appropriés, il faut alors chercher d'autres sources de données. Alternativement, il est possible de construire plusieurs modèles et se rendre compte que **les modèles ne répondent pas de manière adéquate au problème défini**, il est nécessaire alors de redéfinir le problème. Chaque étape du processus peut être répétée plusieurs fois afin de créer un bon modèle.

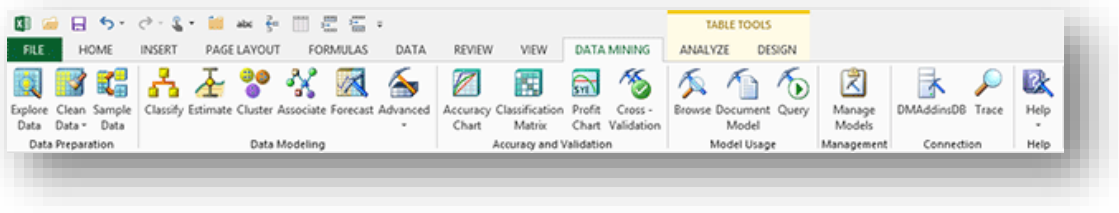
Complément Excel

Le complément SQL Server 2012 Data Mining pour Office est un ensemble d'outils permettant la construction de modèles analytiques pour la prédiction, recommandation, ou l'exploration de données.

Pour le télécharger : http://bit.ly/1KNzyyY

Lors de l'installation du complément d'exploration pour Office 2013, les **barres d'outils** et les modèles suivants seront ajoutés :





Outils d'analyse de table pour Excel : Les outils d'analyse de tables constituent un moyen simple et familier d'accéder à la **puissance de l'exploration de données** Microsoft SQL Server Analysis Services et de générer des **rapports conviviaux et utilisables**, exploitables par des experts ou par des débutants en matière d'exploration de données. Excel n'effectue pas l'exploration de données, celle-ci est **exécutée sur le serveur** et les résultats sont retournés dans votre feuille de calcul Excel.

Data Mining Client pour Excel : Permet une analyse complète, et la prévision de résultats en utilisant soit des données **de feuille de calcul ou de données externes** accessibles via votre base de données Analysis Services.

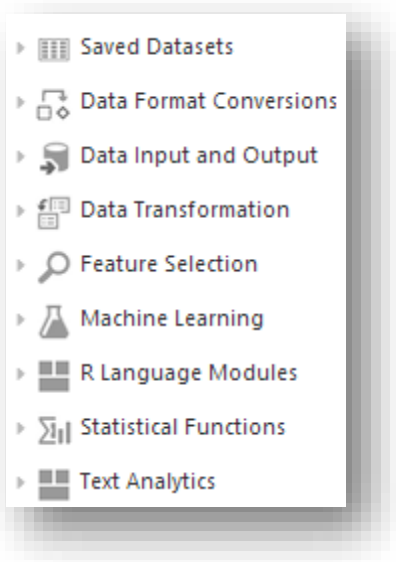
Azure Machine Learning

Azure Machine Learning (ML) est un service permettant de réaliser des **analyses prédictives**. En tirant parti du cloud, Azure ML permet à un large public d'accéder à **l'apprentissage automatique depuis un navigateur web**. En effet, l'apprentissage automatique requiert généralement des logiciels complexes, des ordinateurs ultra-performants et des « Data Scientist ». Pour de nombreuses start-ups et même pour de grandes entreprises, cette technologie reste trop compliquée et trop chère.

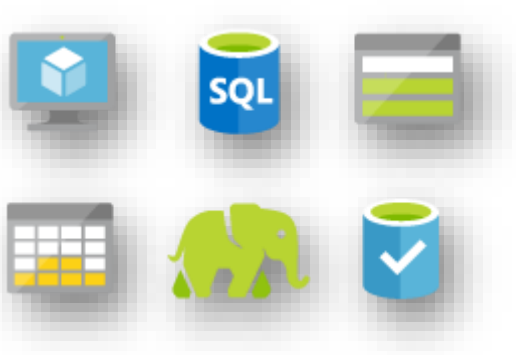
Après avoir créé un **espace de travail**, il est possible de charger des données, créer des expérimentations, publier des Web Services…

Voici à quoi ressemble la fenêtre de création d'expérimentation :

Un certain **nombre de modules** permettent de formater, corréler, transformer les données… Ils sont organisés par types d'action :



Les **données d'entrée** peuvent aujourd'hui être récupérées des **sources** suivantes :



- Azure Blob Storage;
- Azure Table Storage;
- HTTP;
- SQL Azure;
- Hive Query;
- Power Query.

Il est possible **d'enregistrer** les résultats d'une expérimentation dans les **destinations** suivantes :

- Azure Blob Storage ;
- Azure Table Storage ;
- Azure SQL Database ;
- Hive Query.

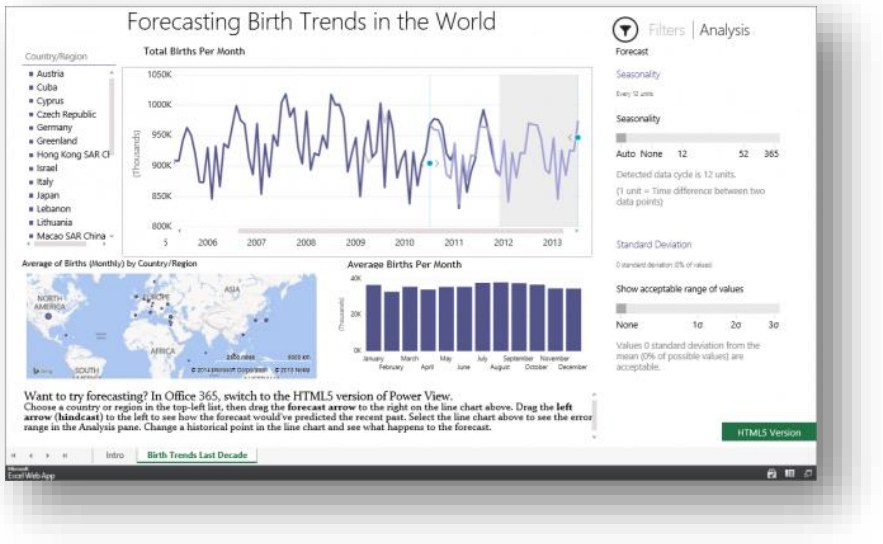
Power View dans Office 365

Depuis peu, **Power View**[17] dans sa version online **Office 365** intègre la possibilité d'effectuer les opérations suivantes :

- De la **prédiction sur les données** : Pour se baser sur un jeu de données afin de prévoir des données futures ;

[17] Il est à noter que depuis décembre 2014, ce service n'est plus disponible car celui-ci est en maintenance : http://bit.ly/1K8eK8J.

- Du **backtesting ou tests rétroactifs** de validité qui consiste à tester la pertinence d'un modèle ou d'une stratégie en s'appuyant sur des données historiques ;
- Du **réajustement** : Pour corriger ou supprimer une valeur très particulière sur les données pour avoir une visualisation plus réaliste des prévisions.



Remarque : ces nouvelles options ne sont disponibles qu'en **version HTML 5**.

Et demain ?

Dans la modélisation, la conception et l'architecture ne sont pas figées. Les possibilités s'accroissent en fonction **des améliorations et enrichissements** de la stack BI Microsoft.

Des innovations remettent en question la manière de gérer **la sécurité** par exemple. La **sécurité au niveau ligne** existera dès la base de données alors qu'elle n'était gérée que sur BISM. Les possibilités de stockage seront également augmentées. Les **données JSON** seront nativement reconnues comme données relationnelles.

Les tendances du marché allant clairement vers le **traitement de volume de masse** et vers **le cloud**, Microsoft proposera dans la prochaine version de nombreuses innovations dans ce sens. Tout d'abord, la généralisation de **Polybase à l'édition SMP** de SQL Server permettra une interopérabilité des systèmes relationnels et Big Data permettant d'en **démocratiser l'usage**. Ensuite, les **scénarii hybrides**, mélangeant données on-premise et dans le cloud, seront facilités. Les **bases de données Stretch** feront leur apparition et offriront la possibilité de séparer données critiques, des données qui ne le sont pas ou des données courantes de celles historiques. Les scénarii de **hautes disponibilités hybrides** seront également améliorés. Le **cloud** est clairement une priorité et bien qu'il ne se substituera sans doute jamais complétement aux systèmes **on-premise**, la part d'utilisation du cloud augmentera à l'avenir…

La modélisation dans les environnements décisionnels n'échappe pas à ces tendances. En effet, du côté du stockage dans Azure, Microsoft devrait sortir de nouvelles architectures comme **Azure Datawarehouse**, **Azure Data Lake** ou encore **SQL Database Elastic**. Coté moteur d'analyse, des nouveautés très attendues vont faire leur apparition comme la gestion du « **many-to-many** » dans SSAS Tabular et la levée de certaines limitations existant actuellement[18] citons par exemple **Power View pour Excel 2016**, qui saura se connecter aux modèles multidimensionnels.

[18] Régulièrement au long cette partie du livre blanc, des limitations ont été détaillées, libre au lecteur de s'y reporter.

En savoir plus

Conférence de Satya Nadella sur la culture de la donnée vue par Microsoft : http://bit.ly/1oCFmnd

Microsoft dans le carré des leaders du cloud : http://bit.ly/1FgqtPm et http://bit.ly/1e3oS5s

Choosing a Tabular or Multidimensional Modeling Experience in SQL Server 2012 Analysis Services : http://bit.ly/1vPKG6n

Comparing Tabular and Multidimensional Solutions (SSAS) : http://bit.ly/1zZCXJp

Pour plus d'informations sur le mode de stockage DirectQuery : http://bit.ly/1Bd15av

Choosing Microsoft Business Intelligence (BI) Tools for Analysis : http://bit.ly/1BqAxk8

APS aux Journées SQL Server 2014 : http://bit.ly/1E7rEvj

Microsoft Analytics Platform System Delivers Best TCO-to-Performance : http://bit.ly/1K8ffzs

Présentation de Hadoop dans HDInsight : http://bit.ly/1AZiu46

Phoenix Overview : http://bit.ly/1EHlsvD

SQuirreL informations : http://bit.ly/1IAE6vh

Vos commentaires nous aideront à améliorer la qualité de nos livres blancs.

Envoyez vos commentaires.