

# ETL, les questions à se poser

par Yazid Grim ([Business Intelligen\(ce\)](#))

Date de publication :

Dernière mise à jour :

Nous savons tous maintenant ce qu'est un entrepôt de données et comment le modéliser (pour ceux qui ne savent pas, retour à la case départ). Intéressons nous maintenant à comment l'alimenter depuis les données sources. Cet article, sous forme d'une suite de questions, présente une check-list que le lecteur pourra utiliser avant, pendant et après la mise en oeuvre d'un ETL pour l'alimentation d'un entrepôt de données.

I - Introduction

II - Définitions

III - Conception d'un ETL

III-A - Comment procéder ?

III-B - Comment sont mes sources ?

III-C - Comment traiter les données ?

III-D - Comment charger les données dans l'entrepôt ?

III-E - Et les métas données dans tout ça !?

III-F - Et comment contrôler cet ETL ?

IV - Outils d'ETL

IV-A - Éléments à prendre en compte lors du choix de l'ETL

V - Conclusion

## I - Introduction

ETL, acronyme de *Extraction, Transformation, Loading*, est un système de chargement de données depuis les différentes sources d'information de l'entreprise (hétérogènes) jusqu'à l'entrepôt de données (modèles multidimensionnels). Ce système ne se contente pas de charger les données, il doit les faire passer par un tas de moulinettes pour les dé-normaliser, les nettoyer, les contextualiser, puis de les charger de la façon adéquate. Nous verrons par la suite ce que chaque mot veut dire.

Il est important de savoir que la réalisation de l'ETL constitue 70% d'un projet décisionnel en moyenne. Et ce n'est pas pour rien, ce système est complexe et ne doit rien laisser s'échapper, sous peine d'avoir une mauvaise information dans l'entrepôt, donc des données fausses, donc inutilisables.

## II - Définitions

Avant de commencer, visualisez le schéma d'un entrepôt et sa façon de fonctionner (gérer l'historique, dimensions, faits, etc.). Le but du jeu est de faire rentrer toutes les données de l'entreprise dans ce modèle, les données doivent donc être :

- **Dé-normalisées** : dans un DW (Data Warehouse), avoir des doublons n'est pas important, avoir un schéma en troisième forme normale est même déconseillé. Il faut que les données apparaissent là où elles doivent apparaître.
- **Nettoyées** : dans un système de production, les utilisateurs entrent les données. Les risques d'erreurs sont là : entrer la rue au lieu du pays, écrire Canoda au lieu de Canada. Ces erreurs ont des répercussions directes sur les analyses (les commandes avec Canoda comme pays ne feront pas partie des commandes faites au Canada). Il faut pouvoir détecter et corriger ces erreurs.
- **Contextualisées** : imaginez un système de production où les informations sur l'activité du personnel sont enregistrées, et un système de RH ou les informations personnelles, comptables des employés sont stockées. Un entrepôt de données possède une vision universelle, un employé est un employé, et il n'y aura qu'une seule dimension "Employé" avec toutes les informations le concernant.
- **Chargées en DW** : c'est l'étape la plus complexe, il s'agit ici d'ajouter les nouvelles lignes, voir si des lignes ont été modifiées et faire une gestion d'historique, voir si des lignes ont été supprimées et le mentionner dans l'entrepôt, tout en faisant attention de ne pas charger des données en double.

### III - Conception d'un ETL

Si vous cherchez des méthodes de conceptions d'ETL, et bien il n'y en a pas. Chaque entreprise possède ses propres systèmes, sa propre logique de fonctionnement, sa propre culture. Un ETL va essayer de prendre toutes les données de l'entreprise et les mettre dans un DW. Dans ce chapitre, nous essayerons plutôt de voir les questions à se poser pour bien cerner les spécificités de notre ETL.

#### III-A - Comment procéder ?

Deux cas sont à prendre en compte, le chargement initial et les chargements incrémentiels.

Le chargement initial est effectué au tout premier chargement de l'entrepôt et dans des cas spéciaux comme après la perte des données de l'entrepôt. Dans ce cas, on charge toutes les données de l'entreprise dans l'entrepôt.


Le chargement incrémentiel est le fait d'ajouter des données à un entrepôt existant, c'est l'opération qui va se répéter dans le temps (chaque jour par exemple). Il faudra faire attention dans ce cas à ne charger que les informations nouvelles, et ne pas charger deux fois la même information.

#### III-B - Comment sont mes sources ?

Avant de faire un ETL, il faut bien étudier les sources de données. En effet, c'est d'après les sources que les stratégies de chargement vont se faire.

Il est à noter que le rapatriement des données peut se faire de trois façons différentes :

- **Push** : dans cette méthode, la logique de chargement est dans le système de production, il " pousse " les données vers le Staging quand il en a l'occasion. L'inconvénient est que si le système est occupé, il ne poussera jamais les données.
- **Pull** : au contraire de la méthode précédente, le Pull " tire " les données de la source vers le Staging. L'inconvénient de cette méthode est qu'elle peut surcharger le système s'il est en cours d'utilisation.
- **Push-Pull** : vous le devinez, c'est le mélange des deux méthodes. La source prépare les données à envoyer et prévient le Staging qu'elle est prête. Le Staging va récupérer les données. Si la source est occupée, le Staging fera une autre demande plus tard.

 *Staging est le terme désignant l'endroit où se fait l'ETL. C'est une machine dédiée à ce travail dans la plupart des cas. Considérez le Staging comme une zone tampon entre les sources de données et l'entrepôt.*

Une fois la bonne stratégie choisie (en fonction de vos spécificités en entreprise), il est temps de se poser les questions fondamentales qui dessineront les caractéristiques de votre système :

- Quelle est la disponibilité de mes sources de données ?
- Comment y accéder ?
- Comment faire des chargements incrémentiels ?
- Quel est le temps d'un chargement incrémentiel moyen, ai-je la possibilité de recharger des données dans le cas où mon processus de chargement échoue ?
- Quelle politique vais-je utiliser dans le cas d'échec de chargement ?

Ces questionnements vous aideront à établir une stratégie de chargement des données sources dans le *Staging*.

### III-C - Comment traiter les données ?

Maintenant que les données sont dans le Staging, va falloir nettoyer tout ça ! C'est l'opération la plus importante (et la plus casse-tête) du processus. En effet, une erreur dans un champ affecte forcément les analyses (exemple de Canada et Canonda). Voici les questions à se poser à cette étape :

- Quels sont les champs les plus sujets à erreurs ?
- Ai-je les moyens de corriger les erreurs automatiquement ?
- Comment permettre à un utilisateur de corriger les erreurs ?
- Quelle politique vais-je utiliser pour le traitement des erreurs (fichier log, table dans BD) ?
- Comment montrer à l'utilisateur final que des données n'ont pas été totalement chargées à cause d'erreurs ?

### III-D - Comment charger les données dans l'entrepôt ?

La dernière mission de l'ETL, charger les données dans le DW. Le point critique dans cette étape est qu'il faut avoir, à tout moment, un contrôle total sur les processus. Exemple : pendant un projet de construction d'entrepôt, vous commencez à automatiser les chargements incrémentiels. Mais un jour, la machine plante au beau milieu du chargement, c'est-à-dire qu'une partie des données a été chargée et une autre non (c'est du vécu). Que faire ??

Et bien si vous n'aviez pas prévu cela, pauvre de vous ! Vous n'avez qu'à vider la base et la recharger, avec toutes les pertes d'historique que cela implique, ou sinon prendre le temps et chercher une à une, les informations qui ont été chargées.

Voici les questions qu'il faut se poser pour cette étape :

- Que faire si un chargement échoue ?
- Ai-je les moyens de revenir à l'état avant le chargement ?
- Puis-je revenir dans le temps d'un chargement donné ?
- Comment valider mon chargement, comment détecter les erreurs ?

### III-E - Et les métas données dans tout ça !?

Et oui ! On n'en parle jamais assez ! Et quand on en parle, ben on ne sait pas forcément ce que ça veut dire ! C'est pourtant une des clés du succès de tout projet décisionnel.

Les métas donnés, en informatique décisionnelle, sont des informations décrivant notre environnement décisionnel. Je ne parle pas seulement des informations concernant le schéma des entrepôts ou la politique d'attribution de noms aux champs de l'entrepôt, mais de tout ce qui, de près ou de loin, peut ajouter de la compréhension aux chiffres présentés.

En effet, il est peut être pertinent pour notre ami analyste de savoir que la colonne prix qu'il est en train d'analyser provient des archives et non des données courantes. Il est peut être utile aussi de savoir que les chiffres devant nos yeux sont issus d'un chargement qui a échoué mais qu'on a réussi à recharger correctement. Il est important pour le grand patron d'une entreprise d'avoir une petite info bulle qui lui indique que les données de son tableau de bord sont ceux de l'avant-veille car le chargement ne s'est pas bien déroulé. Imaginez la catastrophe si le décideur prenait des décisions sur des données erronées !!

Il est très important, dans un environnement décisionnel, de non seulement documenter tout le projet (des processus d'ETL aux logs de chargements) mais de rendre aussi disponible toutes ces métas données aux utilisateurs de l'environnement pour générer encore plus de connaissance. Car n'oubliez pas que c'est le but finalement, créer de la connaissance.

### III-F - Et comment contrôler cet ETL ?

Dans tout ce que vous ferez dans la vie, le contrôle est la clé du succès. Le non contrôle amenant inévitablement le risque et le risque entraînant l'erreur. Si vous ne savez pas d'où provient une erreur, il est fort probable qu'elle soit du côté d'un élément dont vous n'avez pas le contrôle. Le meilleur système étant celui qui laisse le moins de place au risque.

La partie philosophie à la matrix étant dite. Intéressons nous à comment contrôler un ETL, quels sont les points clés à surveiller et, surtout, que faire lorsqu'un élément ne fait pas son travail correctement.

Les ETL sont, malheureusement, la plus grande faiblesse des environnements décisionnels. Tout est critique et sujet à contrôle dans un ETL. Sans oublier que le contrôle sans action ne nous est en rien utile !

Comment retourner en arrière, c'est la principale question que je me pose et à laquelle j'essaye de répondre quand je conçois un ETL. Comment retourner en arrière si un chargement s'arrête brusquement, comment revenir à l'état initial si les données semblent incohérentes, comment valider mes transformations#

Le but de tous ces questionnements est de préserver l'intégrité et la " vérité " de l'entrepôt de données. Car c'est le seul point de défaillance de l'environnement (à par la phase de chargement, tout le système est en lecture seule).

## IV - Outils d'ETL

### IV-A - Éléments à prendre en compte lors du choix de l'ETL

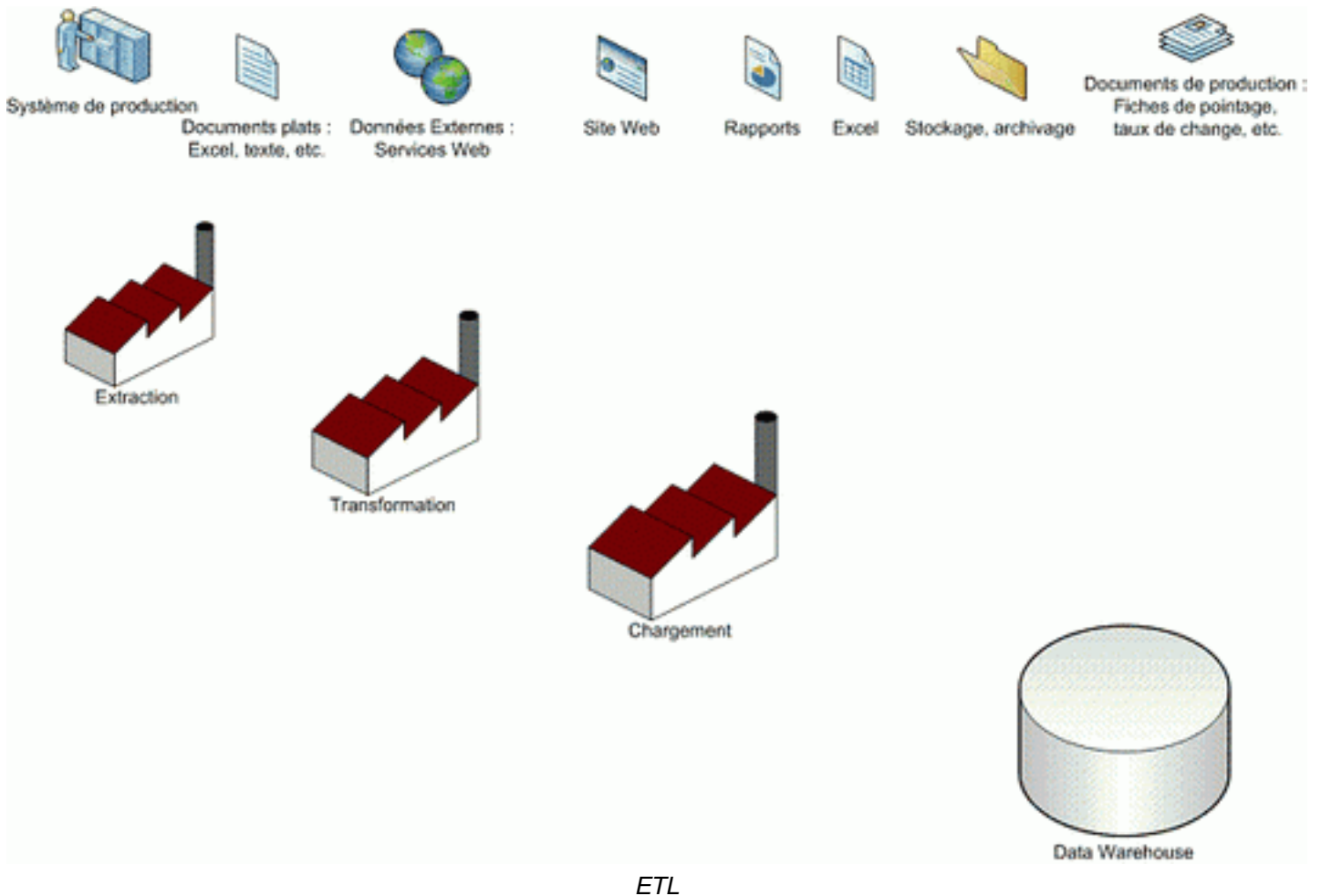
Nous avons la chance d'évoluer, à l'heure où j'écris ces mots, dans un domaine en pleine effervescence, les solutions d'ETL existent, sont nombreuses et répondent à toutes les demandes de performance et de portefeuille (c'est important !).

Cependant, devant un choix si diversifié, on se retrouve un peu perdu : Open Source ou payant, solution intégrée ou indépendante, sous traitance ou développement. Les éléments à prendre en compte dans le choix de votre ETL sont les suivants :

- **Taille de l'entreprise** : j'entends par là taille des structures. S'il s'agit d'une multinationale avec des milliers de succursales à travers le monde, on ira plus pour une solution complète et, en général, très coûteuse. Si on est une PME, on optera plutôt pour des solutions payantes (comme Microsoft Integration Services) assurant un certain niveau de confort sans impliquer des mois de développement.
- **Taille de la structure informatique** : une entreprise avec une grosse structure informatique pourra se permettre d'opter pour une solution Open Source et la personnaliser selon les besoins de l'entreprise. Une PME ne pourra sûrement pas faire cela.
- **Culture d'entreprise** : évidemment, si une entreprise à une culture de l'Open Source très prononcée, l'application d'une solution payante risquera fortement de subir un phénomène de rejet.
- **Maturité des solutions** : il existe des solutions bien rodées, qui fonctionnent bien et qui bénéficient d'un bon retour d'expérience, c'est en général les plus chères (Business Objects, Oracle, SAP). Il existe d'autres solutions, moins matures, bénéficiant d'un " effet de mode " et qui semble offrir de très bonnes performances (Microsoft). Enfin, il existe des solutions Open Source qui, de part leur jeunesse, n'offrent pas autant de flexibilité et de facilité de mise en #uvre que les solutions précédemment citées. Il faudra compter avec le temps pour que ces solutions émergent et arrivent à un niveau de maturité acceptable#

## V - Conclusion

Une image valant mille mots, je tiens à terminer cet article par un petit schéma montrant la position et un résumé des fonctionnalités d'un ETL dans un environnement décisionnel.



### ETL

Le secret d'un bon ETL réside dans sa complétude et dans son exhaustivité dans la prise en charge des données depuis les sources de données jusqu'à l'entrepôt.

