NF26 P2017 **GUILLOT Pierre-Louis**
27/03/17 **LAFFON Paul**

TD n°1 : Établissement du Data Warehouse

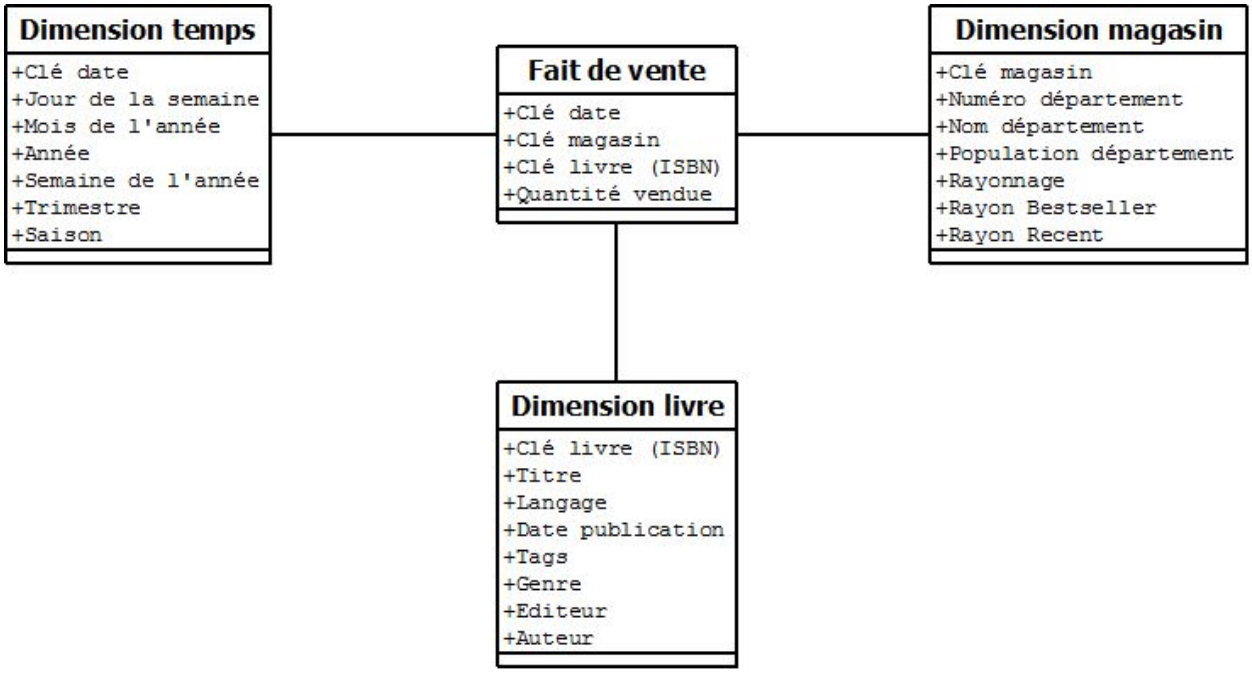
Nous allons dans ce rapport présenter le travail qui a été effectué lors de ce premier TD de NF26.

Celui-ci a consisté en la formation à partir de sources d'informations éparses d'un recueil de données de type ***data warehouse*** selon une organisation préalable en dimension -- que nous avons déjà évoqué dans le rapport intermédiaire.

Nous allons revenir sur les principaux choix techniques et conceptuels qui ont pu être fait lors de son élaboration. Nous reviendrons à chaque étape sur les différentes solutions possibles en insistant sur ce qui distingue l'option qui sera finalement implémentée dans notre projet des autres. Le but sera ainsi qu'à la fin de la lecture les principales décisions que nous avons effectuées soient rendues claires et justifiées.

1° Modèle conceptuel de données

Pour le modèle conceptuel de données, nous repartons de celui qui a été fait durant le premier TD qui peut se modéliser de la façon suivante



2° Alimentation de la dimension “livre”

Le problème des ISBN inconnus

Les principaux problèmes liés à l’implémentation de la dimension *Livre* sont liés à la façon dont on utilisera ses entrées dans la *table de fait* qu’on voudra produire.
En effet, celle-ci sera construite selon des jointures sur le champ *ISBN* et si l’on décide de faire une jointure *“classique”* de type *INNER JOIN* on va se retrouver avec une explosion du nombre d’entrées en sortie de la jointure.
Dans un tel cas tous les livres ayant des *ISBN* anormaux -- qu’on aura normalisés au préalable en une unique valeur de référence -- seront mis en relation avec toutes les ventes dans lesquelles l’*ISBN* associé est aussi inconnu.
Il nous a donc fallu répondre à ce problème et surtout mettre en revue les différentes solutions possibles, les avantages et les inconvénients de chacune.

On peut en effet décider de procéder de plusieurs façons au moment d'intégrer la dimension *Livre* à la *Table de fait* :

- Faire un LEFT JOIN / dédoublonner dans pentaho les livres de *Catalogue* : tout livre inconnu dans notre recueil de transactions sera alors associé arbitrairement à un des livres à l'*ISBN* inconnu. Ne gardant que la valeur de son *ISBN* dans l'agrégat final cela ne posera pas de problème puisque tous les *ISBN* des livres inconnus seront mis à la même valeur de référence.
- Supprimer tous les livres inconnus : on pourra alors effectuer un LEFT JOIN et garder la colonne *ISBN* associée au recueil des transactions du fichier *Fantastic*, l'autre étant automatiquement mise à *null*.
- Supprimer les livres inconnus mais ajouter un livre inconnu de référence : celui-ci servira à faire un INNER JOIN et à matérialiser dans la base de données l'existence d'autres livres inconnus

Nous avons décidé de choisir la première option : on a **gardé tous les livres ayant un *ISBN* inconnu** tout en effectuant dans pentaho un **dédoublonnement sur les livres ayant le même *ISBN*** avant la jointure.
L'avantage principal de cette façon de faire est qu'elle nous permet -- contrairement aux deux autres -- de garder en mémoire tous les livres qu'ils possèdent un *ISBN* ou non.
En effet, là où les entrées relatives à des livres dont on ne connaît pas l'*ISBN* n'auront pas d'intérêt pour les questions concernant les ventes -- puisqu'on devra les ignorer lors d'une jointure, elles pourront être utiles si jamais l'utilisateur veut effectuer une requête portant exclusivement sur les livres.
Dans le cas où un utilisateur voudrait par exemple savoir "Combien de livres ont été écrit par Agatha Christie ?" ces entrées trouveront toute leur utilité. Si on avait supprimé ces entrées nous n'aurions jamais pu garantir une réponse correcte à une telle question -- ne sachant pas si celle-ci met en oeuvre des livres ayant été supprimés.

Notons tout de même que cette façon de faire possède aussi son lot d'inconvénients puisqu'elle risque de compliquer par la suite les jointures qu'on pourra effectuer entre la dimension *Livre* et la *Table de fait*, dans lesquelles pourra se poser à nouveau le problème de l'explosion du nombre d'entrées en sortie. On devra donc être prudent et garder cela en tête à chaque fois qu'on fait de telle jointure : il faudra effectuer un *LEFT JOIN* ou alors trouver un moyen d'effectuer un dédoublonnage *a priori*.

La gestion des tags

Une colonne de la table *Catalogue* sur laquelle on s'est interrogé est celle associée aux tags.
En effet ceux-ci se distinguent par plusieurs aspects :

- Ils sont multiples, et ne semblent groupés selon aucun ordre logique
- Ils contiennent des informations très hétérogènes : on peut par exemple y lire indépendamment "*fantastique*", "*fantasy*", ou encore "*roman fantastique*". Certains tags

contiennent parfois des informations comme le nom des auteurs comme “*Brown”* pour le livre “*Anges et démons*” de “*Dan Brown”*.

- Ils contiennent beaucoup de bruit, on peut par exemple penser au livre ayant pour référence *796* qui contient dans la colonne “tag” la liste des valeurs de toutes ses autres colonnes.

On ajoutera aussi qu’il existe de nombreux ensembles de tags distincts, comme on a pu l’observer par la commande SQL suivante :

```
select count(*), count(distinct tags) from nf26prof2.catalogue;
1443 289
```

A noter que cela s’explique par le fait qu’on compte ici les tags et non pas les ensembles de tag.

On a aussi essayé d’observer la répartition des différents tags :

```
select count(ISBN) as nb_livres, tags from nf26prof2.catalogue group by tags order by nb_livres DESC;
```

Et les liens éventuels entre le *tag* et le *genre* :

```
select tags, genre from nf26prof2.catalogue;
```

On s’est alors rendu compte non-seulement que le *genre* était **très corrélé** avec le *tag* mais aussi que le *genre* associé formait souvent le seul *tag* dans la liste de *tags* qui semblait avoir une quelconque constance ou une quelconque pertinence..
On a donc vite considéré que le *tag* ne nous apportait pas grand chose et qu’il pouvait en termes décisionnels être considéré comme une version moins bien formalisée du *genre*.
Nous avons beaucoup hésité à le supprimer pour finalement décider de le garder sur la base que si celui-ci n’aura probablement aucune valeur discriminante sur les requêtes qui seront effectuées il pourrait avoir simplement pour vocation d’être affiché en résultat de certaines requêtes pour être lu par l’utilisateur final qui pourrait comprendre les quelques informations utiles qui s’y trouvent.

Normaliser les ISBN

Lorsqu'on a voulu normaliser les *ISBN* nous avons effectué quelques observations sur la colonne correspondante de la base *Catalogue*.
Il s'avère que les *ISBN* sont généralement présents sous 2 formats différents : soit un code a 13 chiffres qui correspond à la norme internationale d'identification d'un livre, soit -- souvent -- un code à seulement 3 chiffres.
Après quelques recherches sur internet il s'avère que l'ISBN peut être décomposé en plusieurs parties :

- Les 3 premiers chiffres sont toujours 978 ou plus récemment 979.
- Les 9 chiffres suivants sont le code ISBN lui même (les 4 premiers chiffres identifient le pays et l'éditeur)
- Le dernier chiffre correspond à une clé de contrôle

On serait donc tenter d'interpréter ces données à 3 chiffres, **très nombreuses** dans la base catalogue, comme constituant le début de la deuxième partie du code *ISBN*, et donc de transformer par exemple l'entrée "145" en 978145XXXXXXX, mais cela pose problème.
En effet le fait de les interpréter comme les premiers chiffres et pas les derniers par exemple peut sembler être un choix arbitraire. Il nous a semblé que ces deux types de valeurs étaient trop différents pour être ainsi réunies.
De plus, on se retrouve avec un grand nombre d'*ISBN* qui ne sont codés que par 3 chiffres.
Nous avons pris le parti de considérer cette plage de 1000 valeurs comme trop petite pour permettre une identification et avons donc remplacé tous les ISBN ne contenant pas l'ensemble des 13 caractères par la valeur 0.
On remarquera cependant que les ISBN compris entre 0 et 999 sont tous distincts dans la base *Catalogue*.

NOTE : On a pu se permettre de prendre 0 comme valeur d'*ISBN* inconnu étant donné que les 3 premiers chiffres d'un *ISBN* correct sont nécessairement 978/979 et qu'il ne vaudra donc jamais 0

On notera tout de même que l'on aurait pu effectuer beaucoup plus de vérification sur un *ISBN*, comme vérifier qu'ils commencent bien tous par 978/979, que la clé de contrôle est valide, ...

Voici les résultats de l'intégration de cette dimension dans Pentaho :

	Nom étape	N°Copie	Lignes lues	Lignes écrites	Lignes en entrées	Lignes en sortie
1	Catalogue	0	0	1443	1443	0
2	Filtrage ISBN	0	1443	1443	0	0
3	Correction ISBN	0	295	295	0	0
4	Normalisation Date	0	1443	1443	0	0
5	Décomposition Autheur	0	1443	1443	0	0
6	Fill null	0	1443	1443	0	0
7	Date to Date Type	0	1443	1443	0	0
8	Publisher Inconnu	0	1443	1443	0	0
9	Insertion dans table	0	1443	1443	0	1443

2018/03/26 19:45:26 - Pentaho Data Integration - Using legacy execution engine
2018/03/26 19:45:26 - Pentaho Data Integration - Transformation ouverte.
2018/03/26 19:45:26 - Pentaho Data Integration - Chargement transformation [catalogue]...
2018/03/26 19:45:26 - Pentaho Data Integration - Exécution de la transformation démarrée.
2018/03/26 19:45:26 - catalogue - Distribution démarrée pour la tranformation [catalogue]
2018/03/26 19:45:26 - Insertion dans table.0 - Connected to database [NF26] (commit=1000)
2018/03/26 19:45:26 - Catalogue.0 - Finished reading query, closing connection.
2018/03/26 19:45:26 - Filtrage ISBN.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=1443, Ecrites=1443, Maj=0, Erreurs=0)
2018/03/26 19:45:26 - Catalogue.0 - Fin exécution étape (Entrées=1443, Sorties=0, Lues=0, Ecrites=1443, Maj=0, Erreurs=0)
2018/03/26 19:45:26 - Correction ISBN.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=295, Ecrites=295, Maj=0, Erreurs=0)
2018/03/26 19:45:26 - Normalisation Date.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=1443, Ecrites=1443, Maj=0, Erreurs=0)
2018/03/26 19:45:26 - Décomposition Autheur.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=1443, Ecrites=1443, Maj=0, Erreurs=0)
2018/03/26 19:45:26 - Fill null.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=1443, Ecrites=1443, Maj=0, Erreurs=0)
2018/03/26 19:45:26 - Date to Date Type.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=1443, Ecrites=1443, Maj=0, Erreurs=0)
2018/03/26 19:45:26 - Publisher Inconnu.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=1443, Ecrites=1443, Maj=0, Erreurs=0)
2018/03/26 19:45:26 - Insertion dans table.0 - Fin exécution étape (Entrées=0, Sorties=1443, Lues=1443, Ecrites=1443, Maj=0, Erreurs=0)
2018/03/26 19:45:26 - Pentaho Data Integration - L'exécution de la transformation a été achevée!

3° Alimentation de la dimension “magasin”

Afin de pouvoir alimenter la dimension magasin nous devons joindre les données des documents marketing contenant la liste des magasins et des départements français. Ces deux documents sont très propres et ne contiennent pas de valeurs manquantes ce qui en a facilité le traitement.

La colonne “Rayonnage” des magasins peut contenir une des 3 valeurs parmi Year, Author et Editor, nous avons choisi de conserver les noms complets plutôt que les abréviations (Y, A ou E). Ce choix est très peu coûteux en mémoire étant donné la faible taille de cette dimension, mais permet un compréhension bien meilleure de la signification de cette colonne.

Nous avons choisi d'ajouter une ligne dans cette dimension qui correspond à tous les magasins “inconnu”. Dans la table de faits, certaines lignes contiennent des magasins qui présentent un format erroné. Plutôt que de ne pas prendre en compte ces données, cela permet de les regrouper sous ce magasin inconnu afin de mieux pouvoir en estimer la taille et ses caractéristiques.

Une autre solution à ce problème aurait été d'effectuer une “LEFT JOIN” durant la manipulation de la table de faits afin de pouvoir distinguer les magasins valide de ceux qui sont erronés. Cependant, contrairement au fait d’ajouter un magasin “inconnu”, cela ne permet pas de différencier les magasins ayant un format erroné des magasins ayant un format correct mais n’étant pas présent dans la dimension magasin (même si ce cas-là n’est pas présent dans notre table de fait)

Statistique de l’alimentation de la dimension magasin

	Nom étape	N°Copie	Lignes lues	Lignes écrites	Lignes en entrées	Lignes en sortie
1	Magasin inconnu	0	0	1	0	0
2	Extract dept	0	0	95	95	0
3	Marketing	0	0	152	152	0
4	Tri Dpt	0	95	95	0	0
5	Tri Mag	0	152	152	0	0
6	Jointure comparaison lignes	0	247	152	0	0
7	Select	0	152	152	0	0
8	Export NF26P025	0	153	153	0	153

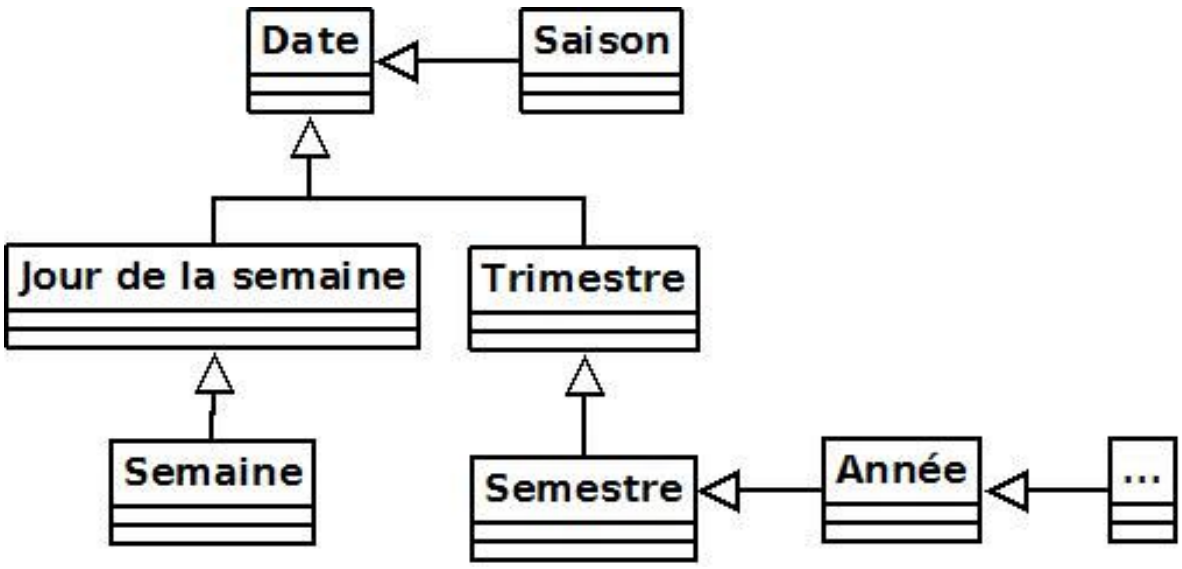
Trace d'exécution de l’alimentation de la dimension magasin

```
Pentaho Data Integration - Using legacy execution engine
Pentaho Data Integration - Transformation ouverte.
Pentaho Data Integration - Chargement transformation [catalogue]...
Pentaho Data Integration - Exécution de la transformation démarrée.
```

```
Pentaho Data Integration - L'exécution de la transformation a été achevée!
Pentaho Data Integration - Using legacy execution engine
Pentaho Data Integration - Transformation ouverte.
Pentaho Data Integration - Chargement transformation [magasin]...
Pentaho Data Integration - Exécution de la transformation démarrée.
magasin - Distribution démarrée pour la tranformation [magasin]
Export NF26P025.0 - Connected to database [NF26] (commit=1000)
Magasin inconnu.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=0, Ecrites=1, Maj=0, Erreurs=0)
Extract dept.0 - Fin exécution étape (Entrées=95, Sorties=0, Lues=0, Ecrites=95, Maj=0, Erreurs=0)
Tri Dpt.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=95, Ecrites=95, Maj=0, Erreurs=0)
Marketing.0 - Fin exécution étape (Entrées=152, Sorties=0, Lues=0, Ecrites=152, Maj=0, Erreurs=0)
Tri Mag.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=152, Ecrites=152, Maj=0, Erreurs=0)
Jointure comparaison lignes.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=247, Ecrites=152, Maj=0, Erreurs=0)
Select.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=152, Ecrites=152, Maj=0, Erreurs=0)
Export NF26P025.0 - Fin exécution étape (Entrées=0, Sorties=153, Lues=153, Ecrites=153, Maj=0, Erreurs=0)
Pentaho Data Integration - L'exécution de la transformation a été achevée!
```

4° Alimentation de la dimension “temps”

Nous avions donc décidé dans le préambule de ce TD de créer une dimension relative au temps, que nous avons modélisé sous la forme d’une hiérarchie multiple :



Cependant, pour modéliser celle-ci on s'est en fait assez rapidement rendu compte que deux façons complètement différentes de mettre en place cette dimension *temps* se dégageaient, qui correspondaient en fait à deux façons de la conceptualiser.

Dans la première façon de faire, on considérera que la dimension *temps* est constituée des différentes dates **auxquelles ont pu être effectuées des ventes**. Une entrée y sera enregistrée si et seulement si la date qui correspond apparaît au moins une fois dans le fichier *fantastic* -- registre qui sert à emmagasiner les différentes transactions.

Dans la deuxième on considérera au contraire que la dimension *temps* concerne l'écoulement du temps **indépendamment des ventes** qui ont pu y être effectuées. Elle possède des avantages non seulement techniques mais aussi conceptuels puisqu'elle est cohérente avec la façon dont nous avons défini les dimensions *Livres* et *Magasin*.
En effet, on y avait par exemple décidé d'ajouter les quelques livres appartenant au catalogue mais jamais vendu.
On avait retrouvé ceux-ci par la commande :

```
select dim_livre.ISBN, dim_livre.title FROM faits RIGHT JOIN dim_livre on dim_livre.ISBN = faits.ISBN WHERE faits.ISBN is null;
```

ISBN	***Title***
9782207259160	*La Forêt De Cristal: Roman*
9782265037755	*L'Abominable Postulat*

Dans tous les cas, le choix entre ces deux façons de faire a constitué un des choix les plus importants de notre modélisation et après avoir chacun de notre côté essayé d'implémenter les deux on en a résumé les différents avantages et inconvénients.

On peut par exemple noter que si l'on considère uniquement les dates auxquelles des ventes ont eu lieu on s'affranchit du choix relativement arbitraire de la plage de valeurs de temps que l'on voudra considérer. En effet, si le grain minimal d'une journée est plutôt évident, on peut se demander quelles sont la date la plus petite et la plus grande qu'on voudra prendre en considération. Choisir comme date la plus petite celle de la transaction la plus ancienne pourrait convenir *a priori* si jamais les transactions passées n'était jamais amenées à être rectifiées ou ajoutées. En situation réelle on aurait probablement cherché à répondre plus précisément à cette question en allant par exemple voir le moment où l'entreprise fantastique a commencé à vendre des livres.
Dans notre cas on aura toujours à rajouter une marge sur les dates considérées. Celle-ci sera ajoutée avant la date de la première vente et après la date de la dernière / après la date courante, de manière à ce que les mises à jour futures des transactions et donc de la *table de fait* ne sortent pas de la plage de valeurs pré-établie.

La valeur de cette marge aura donc nécessairement un caractère arbitraire : il faudra déterminer une date jusqu'à laquelle on voudra être capable de mettre à jour la base de fait sans avoir à modifier à nouveau la dimension *temps*.

Un autre avantage de cette façon de faire est qu'elle permettrait de relativement facilement se rendre compte des jours au cours desquels des ventes ont été effectuées. Une requête portant sur la présence d'achats relativement à la seule dimension *temps* -- par exemple "quelle saison de 2018 a eu le plus de jours avec au moins une vente ?" -- ne demandera pas de faire de jointure avec la *table de fait* contrairement au cas où on aurait décidé d'enregistrer toutes les dates dans une plage de valeurs.

Enfin le dernier avantage est que cela nous permet d'enregistrer beaucoup moins d'éléments dans la table : on se contente du nombre d'éléments strictement nécessaires pour répondre à toutes les questions typiques par le biais de jointures.
Cela peut se montrer avantageux au cas où on est amené à en effectuer plusieurs, ce qui nous permettrait de diminuer pour chacune d'entre elles le nombre d'opérations à effectuer.
Cependant on sera forcé de constater que la table correspondant à la dimension *Temps* sera de toute manière relativement petite, et que le problème que l'on cherche à résoudre fait de toute façon appel à un nombre relativement restreint de données.

Pour ce qui est de la méthode "*naïve"* qui consiste à enregistrer toutes les dates dans un intervalle donné, elle présente elle même plusieurs avantages. Elle permet notamment -- si on choisit une plage de valeur assez large -- de ne pas avoir à mettre à jour la dimension *temps* à chaque fois qu'une transaction est effectuée.
En effet si on a dit plus tôt qu'un des désavantages de la méthode était que pour avoir à effectuer cette mise à jour le moins souvent possible il fallait choisir une marge assez grande mais arbitraire, il ne faut pas oublier que dans le cas où on enregistre seulement les dates au cours desquelles des transactions ont été effectuées on doit modifier la dimension *temps* **à chaque fois que des transactions ayant lieu à une nouvelle date sont enregistrées**. La méthode "*naïve"* est donc de toute façon supérieure à cette dernière pour ce qui est de la facilité de maintenance : celle-ci ne devra de toute façon être mise à jour que très rarement en comparaison !
On notera qu'on aurait pu penser une version hybride dans laquelle à chaque nouvel ensemble de transaction ajouté à la *table de fait*, on ajuste la plage de valeur si et seulement si on a enregistré des ventes à une date en dehors de la plage.

Sur un plan conceptuel, un des autres avantages de cette façon de définir les dates est que cela nous permet d'en donner une définition plus générale. En effet, le fait que celle-ci traite la dimension *temps* comme le déroulement du temps indépendamment des ventes nous ouvre des possibilités quant à son utilisation. Celle-ci pourrait servir non seulement à faire des jointures sur la *table de fait* relativement aux dates des transactions correspondantes mais aussi éventuellement à répondre **ponctuellement** à des interrogations effectuées sur les dates de publication par exemple. Nous nuancerons cependant cela par le fait que la dimension

Temps n'a sur le plan conceptuel pas pour vocation à être ainsi mise en lien avec la dimension *Livre*, car cela formerait un cycle dans la hiérarchie.
Cependant cette généralisation possède un autre intérêt : elle nous permet de prendre en considération des requêtes relatives aux dates qui n'ont pas de ventes. Notre structure sera ainsi armée pour répondre correctement à des interrogations telles que "A quelles dates le magasin MXX n'a-t-il vendu aucun livre ?".
De telles requêtes auraient en effet été plus compliquées à mettre en place dans le cas où on n'aurait considéré que les dates apparaissant dans au moins une transaction, puisqu'on aurait dû à un moment ou un autre faire une sorte de reconstitution de l'ensemble des dates pour y répondre.

Nous avons finalement résumé les avantages de chacune des deux méthodes dans le tableau suivant :

Dates exhaustives sur un intervalle	**Dates correspondant aux transactions**
Traite la dimension *temps* comme le déroulement du temps en dehors de toute vente.	Traite la dimension *temps* comme le déroulement des périodes dans lesquelles sont effectuées des ventes.
Peut donner une cohérence entre les dates autres que celles de ventes (date publi)	Permet de visualiser plus facilement à quelles périodes de temps des ventes ont été effectuées
Facilite les interrogations sur les dates où rien n'a été vendu	Évite d'avoir à choisir arbitrairement une plage de valeurs
Beaucoup moins de mises à jour	

Et avons finalement décidé de partir sur la méthode *exhaustive* et d'enregistrer toutes les dates sur un intervalle de temps donné, étant principalement convaincu par l'argument selon lequel les mises à jour seraient alors bien plus simples..
Les dates de notre recueil de transactions -- contenues dans le fichier *fantastic* -- étant toutes situées dans l'année 2014 on a décidé de prendre en guise de plage de valeur les dates allant du 31/12/2013 au 30/12/2014 inclus.
Nous noterons aussi que, ayant enregistré les dates au format date dans nos tables SQL, nous avons été obligé de choisir une date de référence pour les dates mal renseignées et avons décidé de prendre la valeur de la date epoch UNIX : le 1er janvier 1970. Celle-ci fait en effet office selon nous de convention assez généralement comprise de valeur nulle et est de plus assez éloigné des dates réellement présentes dans les données fournies.

On a donc rajouté cette date nulle dans la table de la dimension *Temps*, de manière à ce que des jointures puissent être effectuées sur les dates à laquelle la dimension est nulle.
Voici les résultats de l'intégration de cette dimension dans Pentaho :

	Nom étape	N°Copie	Lignes lues	Lignes écrites	Lignes en entrées	Lignes en sortie	Lignes maj
1	365 days	0	0	365	0	0	0
2	Days_since	0	365	365	0	0	0
3	Calc Date	0	365	365	0	0	0
4	Ajout date inconnue	0	0	1	0	0	0
5	Select values	0	365	365	0	0	0
6	Insertion dans table	0	366	366	0	366	0

2018/03/26 19:44:41 - Pentaho Data Integration - Using legacy execution engine
2018/03/26 19:44:41 - Pentaho Data Integration - Transformation ouverte.
2018/03/26 19:44:41 - Pentaho Data Integration - Chargement transformation [date]...
2018/03/26 19:44:41 - Pentaho Data Integration - Exécution de la transformation démarrée.
2018/03/26 19:44:41 - date - Distribution démarrée pour la tranformation [date]
2018/03/26 19:44:42 - Insertion dans table.0 - Connected to database [NF26] (commit=1000)
2018/03/26 19:44:42 - Ajout date inconnue.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=0, Ecrites=1, Maj=0, Erreurs=0)
2018/03/26 19:44:42 - 365 days.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=0, Ecrites=365, Maj=0, Erreurs=0)
2018/03/26 19:44:42 - Days_since.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=365, Ecrites=365, Maj=0, Erreurs=0)
2018/03/26 19:44:42 - Calc Date.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=365, Ecrites=365, Maj=0, Erreurs=0)
2018/03/26 19:44:42 - Select values.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=365, Ecrites=365, Maj=0, Erreurs=0)
2018/03/26 19:44:42 - Insertion dans table.0 - Fin exécution étape (Entrées=0, Sorties=366, Lues=366, Ecrites=366, Maj=0, Erreurs=0)
2018/03/26 19:44:42 - Pentaho Data Integration - L'exécution de la transformation a été achevée!

5° Alimentation du Data Warehouse : table de faits

Pour l'alimentation de la table de fait du data warehouse, nous avons appliqué les mêmes corrections que décrites dans les alimentations des dimensions temps, magasin et livre pour l'ISBN, l'identifiant du magasin et la date. Nous n'avons pas effectué de corrections sur les numéros de tickets car une fois les données agrégées, ce numéro ne sera plus présent dans la table de fait.

Afin de s'assurer que les données étaient en accord avec celles des dimensions, des jointures sont faites avec celle-ci, de type "INNER JOIN" les valeurs inconnues sont présentes dans les

dimensions. Une étape nécessaire a été de dédoublonner les livres inconnus de la dimension livre afin de ne pas multiplier les données inutilement.

Statistique de l'alimentation de la table de faits

	Nom étape	N°Copie	Lignes lues	Lignes écrites	Lignes en entrées	Lignes en sortie	Lignes maj
1	Fantastic	0	0	200000	200000	0	1
2	Filtrage Date	0	200000	200000	0	0	0
3	Correction date	0	200000	200000	0	0	0
4	Livre	0	0	1443	1443	0	0
5	To date	0	200000	200000	0	0	0
6	Magasin	0	0	153	153	0	0
7	Filtrage ISBN	0	200000	200000	0	0	0
8	Tri ISBN livre	0	1443	1443	0	0	0
9	Correction ISBN	0	44918	44918	0	0	0
10	Temps	0	0	366	366	0	0
11	Filtrage Mag	0	200000	200000	0	0	0
12	Correction Magasin	0	6540	6540	0	0	0
13	Tri ISBN	0	200000	200000	0	0	0
14	Tri Temps	0	366	366	0	0	0
15	Dédoublonnage	0	1443	1149	0	0	0
16	Join ISBN	0	201149	200000	0	0	0
17	Select	0	200000	200000	0	0	0
18	Tri Date	0	200000	200000	0	0	0
19	Tri Dim Mag	0	153	153	0	0	0
20	Join Date	0	200366	200000	0	0	0
21	Select 2	0	200000	200000	0	0	0
22	Tri Mag	0	200000	200000	0	0	0
23	Join Mag	0	200153	200000	0	0	0
24	Tri lignes	0	200000	200000	0	0	0
25	Agrégation valeurs	0	200000	81985	0	0	0
26	Table de faits	0	81985	81985	0	81985	0

Trace exécution de l'alimentation de la table de faits

```
2018/03/26 19:42:12 - Pentaho Data Integration - Using legacy execution engine
2018/03/26 19:42:12 - Pentaho Data Integration - Transformation ouverte.
2018/03/26 19:42:12 - Pentaho Data Integration - Chargement transformation [fanstatic]...
2018/03/26 19:42:12 - Pentaho Data Integration - Exécution de la transformation démarrée.
2018/03/26 19:42:12 - fanstatic - Distribution démarrée pour la tranformation [fanstatic]
2018/03/26 19:42:12 - Table de faits.0 - Connected to database [NF26] (commit=1000)
2018/03/26 19:42:12 - Fantastic.0 - Opening file: file:///volsme/users/nf26p025/Documents/TD01/Fantastic.txt
2018/03/26 19:42:13 - Magasin.0 - Finished reading query, closing connection.
2018/03/26 19:42:13 - Tri Dim Mag.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=153, Ecrites=153, Maj=0, Erreurs=0)
2018/03/26 19:42:13 - Magasin.0 - Fin exécution étape (Entrées=153, Sorties=0, Lues=0, Ecrites=153, Maj=0, Erreurs=0)
2018/03/26 19:42:13 - Temps.0 - Finished reading query, closing connection.
2018/03/26 19:42:13 - Tri Temps.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=366, Ecrites=366, Maj=0, Erreurs=0)
2018/03/26 19:42:13 - Temps.0 - Fin exécution étape (Entrées=366, Sorties=0, Lues=0, Ecrites=366, Maj=0, Erreurs=0)
2018/03/26 19:42:13 - Livre.0 - Finished reading query, closing connection.
2018/03/26 19:42:13 - Tri ISBN livre.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=1443, Ecrites=1443, Maj=0, Erreurs=0)
2018/03/26 19:42:13 - Livre.0 - Fin exécution étape (Entrées=1443, Sorties=0, Lues=0, Ecrites=1443, Maj=0, Erreurs=0)
2018/03/26 19:42:13 - Dédoublonnage.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=1443, Ecrites=1149, Maj=0, Erreurs=0)
```

2018/03/26 19:42:13 - Fantastic.0 - linenr 50000
2018/03/26 19:42:13 - Filtrage Date.0 - N°Ligne 50000
2018/03/26 19:42:15 - To date.0 - N°Ligne 50000
2018/03/26 19:42:17 - Filtrage ISBN.0 - N°Ligne 50000
2018/03/26 19:42:17 - Filtrage Mag.0 - N°Ligne 50000
2018/03/26 19:42:17 - Fantastic.0 - linenr 100000
2018/03/26 19:42:17 - Tri ISBN.0 - Linenr 50000
2018/03/26 19:42:18 - Filtrage Date.0 - N°Ligne 100000
2018/03/26 19:42:20 - To date.0 - N°Ligne 100000
2018/03/26 19:42:22 - Filtrage ISBN.0 - N°Ligne 100000
2018/03/26 19:42:25 - Filtrage Mag.0 - N°Ligne 100000
2018/03/26 19:42:25 - Fantastic.0 - linenr 150000
2018/03/26 19:42:25 - Tri ISBN.0 - Linenr 100000
2018/03/26 19:42:27 - Filtrage Date.0 - N°Ligne 150000
2018/03/26 19:42:30 - To date.0 - N°Ligne 150000
2018/03/26 19:42:32 - Filtrage ISBN.0 - N°Ligne 150000
2018/03/26 19:42:34 - Filtrage Mag.0 - N°Ligne 150000
2018/03/26 19:42:34 - Fantastic.0 - Fin exécution étape (Entrées=200000, Sorties=0, Lues=0, Ecrites=200000, Maj=1, Erreurs=0)
2018/03/26 19:42:34 - Tri ISBN.0 - Linenr 150000
2018/03/26 19:42:36 - Filtrage Date.0 - N°Ligne 200000
2018/03/26 19:42:36 - Filtrage Date.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=200000, Ecrites=200000, Maj=0, Erreurs=0)
2018/03/26 19:42:38 - Correction date.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=200000, Ecrites=200000, Maj=0, Erreurs=0)
2018/03/26 19:42:39 - To date.0 - N°Ligne 200000
2018/03/26 19:42:39 - To date.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=200000, Ecrites=200000, Maj=0, Erreurs=0)
2018/03/26 19:42:40 - Filtrage ISBN.0 - N°Ligne 200000
2018/03/26 19:42:40 - Filtrage ISBN.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=200000, Ecrites=200000, Maj=0, Erreurs=0)
2018/03/26 19:42:40 - Correction ISBN.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=44918, Ecrites=44918, Maj=0, Erreurs=0)
2018/03/26 19:42:40 - Filtrage Mag.0 - N°Ligne 200000
2018/03/26 19:42:40 - Filtrage Mag.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=200000, Ecrites=200000, Maj=0, Erreurs=0)
2018/03/26 19:42:41 - Correction Magasin.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=6540, Ecrites=6540, Maj=0, Erreurs=0)
2018/03/26 19:42:41 - Tri ISBN.0 - Linenr 200000
2018/03/26 19:42:41 - Select.0 - N°Ligne 50000
2018/03/26 19:42:41 - Tri Date.0 - Linenr 50000
2018/03/26 19:42:41 - Select.0 - N°Ligne 100000
2018/03/26 19:42:41 - Tri Date.0 - Linenr 100000
2018/03/26 19:42:41 - Select.0 - N°Ligne 150000
2018/03/26 19:42:41 - Tri Date.0 - Linenr 150000
2018/03/26 19:42:42 - Tri ISBN.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=200000, Ecrites=200000, Maj=0, Erreurs=0)
2018/03/26 19:42:42 - Join ISBN.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=201149, Ecrites=200000, Maj=0, Erreurs=0)
2018/03/26 19:42:42 - Select.0 - N°Ligne 200000
2018/03/26 19:42:42 - Select.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=200000, Ecrites=200000, Maj=0, Erreurs=0)
2018/03/26 19:42:42 - Tri Date.0 - Linenr 200000
2018/03/26 19:42:42 - Tri Date.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=200000, Ecrites=200000, Maj=0, Erreurs=0)
2018/03/26 19:42:44 - Select 2.0 - N°Ligne 50000
2018/03/26 19:42:44 - Tri Mag.0 - Linenr 50000
2018/03/26 19:42:45 - Select 2.0 - N°Ligne 100000
2018/03/26 19:42:45 - Tri Mag.0 - Linenr 100000
2018/03/26 19:42:46 - Select 2.0 - N°Ligne 150000
2018/03/26 19:42:46 - Tri Mag.0 - Linenr 150000
2018/03/26 19:42:46 - Select 2.0 - N°Ligne 200000
2018/03/26 19:42:46 - Join Date.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=200366, Ecrites=200000, Maj=0, Erreurs=0)

2018/03/26 19:42:46 - Select 2.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=200000, Ecrites=200000, Maj=0, Erreurs=0)
2018/03/26 19:42:46 - Tri Mag.0 - Linenr 200000
2018/03/26 19:42:46 - Tri lignes.0 - Linenr 50000
2018/03/26 19:42:46 - Tri lignes.0 - Linenr 100000
2018/03/26 19:42:46 - Tri lignes.0 - Linenr 150000
2018/03/26 19:42:46 - Tri Mag.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=200000, Ecrites=200000, Maj=0, Erreurs=0)
2018/03/26 19:42:47 - Join Mag.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=200153, Ecrites=200000, Maj=0, Erreurs=0)
2018/03/26 19:42:47 - Tri lignes.0 - Linenr 200000
2018/03/26 19:42:47 - Agrégation valeurs.0 - N°ligne 50000
2018/03/26 19:42:47 - Agrégation valeurs.0 - N°ligne 100000
2018/03/26 19:42:47 - Agrégation valeurs.0 - N°ligne 150000
2018/03/26 19:42:47 - Table de faits.0 - linenr 50000
2018/03/26 19:42:48 - Tri lignes.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=200000, Ecrites=200000, Maj=0, Erreurs=0)
2018/03/26 19:42:48 - Agrégation valeurs.0 - N°ligne 200000
2018/03/26 19:42:48 - Agrégation valeurs.0 - Fin exécution étape (Entrées=0, Sorties=0, Lues=200000, Ecrites=81985, Maj=0, Erreurs=0)
2018/03/26 19:42:48 - Table de faits.0 - Fin exécution étape (Entrées=0, Sorties=81985, Lues=81985, Ecrites=81985, Maj=0, Erreurs=0)
2018/03/26 19:42:48 - Pentaho Data Integration - L'exécution de la transformation a été achevée!