Antoine ROBIN
Abdoul RAFIO
Rachid SAHLI

02/02/2024
BUT SD 2 FA EMS

SAE Intégration

Madame Samia Benni (SID)

IUT de Paris - Rives de Seine
Université Paris Cité

SOMMAIRE

I - Introduction

L'entreprise française de e-commerce Nil (www.nil.fr) nous a sollicités afin d'analyser le comportement des utilisateurs sur sa plateforme. Pour cela, nous avons réalisé une modélisation relationnelle avec plusieurs tables de dimensions et de faits.
Notre objectif initial consiste à élaborer un modèle dimensionnel, offrant une structure cohérente pour les données et facilitant leur compréhension.

Pour pouvoir répondre à ce problème, nous allons nous appuyer sur ce que nous avons appris lors des cours de système d'informations décisionnels ce semestre.

Dans un premier temps, nous créerons un modèle relationnel pour organiser de manière cohérente les données, simplifiant ainsi leur compréhension. L'objectif est d'analyser de manière approfondie le comportement des utilisateurs sur la plateforme. En structurant les données selon des dimensions spécifiques, nous pourrons également optimiser l'efficacité des requêtes analytiques. Une conception judicieuse de ce modèle relationnel facilitera non seulement l'analyse du comportement des utilisateurs, mais également la création ultérieure d'un tableau de bord fonctionnel.
Dans un second temps, nous créerons un tableau de bord. Ce dernier nous offrira une visualisation synthétique des données. Il a pour objectif de fournir une vue d'ensemble rapide et facilement compréhensible des données.

Nous partons de 0, c'est-à-dire que nous allons décider des tables que nous allons inclure dans notre modèle. Mais aussi, quels attributs peuvent être intéressants à analyser, afin que l'entreprise Nil dispose de plus d'informations sur ces utilisateurs. Nous choisirons également, si nous optons pour un modèle relationnel en flocon ou un modèle relationnel en étoile. Enfin, nous peuplerons les tables du modèle à l'aide de données générées qui nous permettrons d'effectuer quelques analyses.

Puis nous réfléchirons, à la granularité pour les tables de fait. Mais aussi au type de gestion des changements dans les tables dimensionnelles. Nous préciserons également nos dimensions douteuses, dégénérées et causales.
Nous listerons ensuite deux mesures pour chaque niveau d'additivité.

Afin d'effectuer ce travail, nous utiliserons comme outils techniques, le langage SQL pour concevoir notre base de données. Lors de la réalisation de cette base de données, nous avons également utilisé le langage de programmation R pour résoudre des problèmes liés au format des données. Enfin, nous avons réalisé notre tableau de bord à l'aide de Power bi.

En outre, nous aborderons des notions importantes vu en cours, telles que la granularité des tables de fait et le type de gestion des changements dans les tables dimensionnelles. Nous définirons également avec précision les dimensions douteuses, dégénérées et causales, contribuant ainsi à une modélisation plus nuancée.

II – Tables

Nous avons consacré une réflexion approfondie à la sélection des tables, travaillant avec soin pour ne pas s'éloigner de la problématique de l'entreprise Nil. Nos objectifs étaient de préserver la pertinence de notre modèle relationnel en évitant l'inclusion de tables peu pertinentes pour répondre à notre question, tout en cherchant à obtenir un ensemble maximal d'informations pour analyser le comportement des utilisateurs sur la plateforme Nil.

Pour cela, nous avons opté pour un modèle relationnel en flocon. Ce dernier a pour objectif de réduire la redondance en normalisant les données. Nous utilisons deux tables de faits pour stocker les données mesurées et transactionnelles. Ce sont les tables **Navigation** et **Commande**. La première permet de mesurer le temps que l'utilisateur passe sur le site. Tandis que la seconde permet de mesurer les transactions, qui ici sont les commandes passées sur le site.
Ensuite, nous avons 7 tables dimensionnelles contenant des attributs sur les données de l'entreprise Nil. Cette structure forme donc un modèle en flocon avec des tables dimensionnelles communicant entres elles.

2.1 – Le choix des tables

Pour pouvoir analyser le comportement des utilisateurs sur le site internet de l'entreprise Nil nous avons décidé d'avoir deux tables de faits. Il est important de bien prendre en compte, le fait qu'un utilisateur naviguant sur le site n'est pas forcément client. En conséquence, la table **Navigation** servira à étudier le comportement de tous les utilisateurs du site. Contrairement à la table commande, qui aura pour objectif d'analyser le comportement des clients ayant commandé un article sur la plateforme.

Ensuite, nous avons décidé d'avoir 7 tables de dimensions. Ce choix, nous semble correspondre aux données dont peut disposer une entreprise telle que Nil. Avec ces tables, il y a assez d'informations pour comprendre le comportement des utilisateurs :

- La Table **Livre** contient tous les livres présents et en vente sur le site Nil. Dedans, on trouve toutes les informations relatives à chaque livre.
- La Table **Auteur** contient l'ensemble des auteurs associés à chaque livre. Un livre peut appartenir qu'à 1 auteur, mais un auteur peut avoir écrit plusieurs livres.
- La Table **Catégorie** contient l'ensemble des catégories auxquelles peuvent appartenir les livres. Un livre ne peut appartenir qu'à une seule catégorie. Une catégorie peut contenir plusieurs livres.
- La Table **Page** regroupe l'ensemble des pages du site de Nil.
- La Table **Utilisateur** contient des informations relatives à l'utilisateur. Même si l'utilisateur ne passe pas de commande sur le site internet. Nous disposons d'informations sur ce dernier.
- La Table **Livraison** contient des données qui concernent la livraison des colis et la société qui s'en charge.
- La Table **Détail Commande** contient l'ensemble des informations essentielles pour chaque commande.

2.2 – Les spécifications de certaines tables

Dans la table **Utilisateur**, il est possible pour 1 ou N utilisateurs d'avoir des valeurs manquantes sur les données nominatives (nom, prénom, adresse, sexe…). Cela s'explique par le fait que tout utilisateur n'est pas forcément client. Lorsque l'individu est simplement utilisateur, nous avons par exemple son adresse IP ou encore son navigateur de connexion. Ces informations sont exploitables et nous permettent de comprendre un certain nombre de paramètres sur le comportement des utilisateurs sur la plateforme.
Cependant, chaque utilisateur qui passe une commande devient client et rentre ses informations personnelles lors du paiement.
Lorsqu'un client revient sur le site avec le même appareil qu'il a utilisé pour commander et qu'il ne s'identifie pas. On pourra toujours savoir de quel client il s'agit, car ses informations personnelles sont déjà associées à son adresse IP.

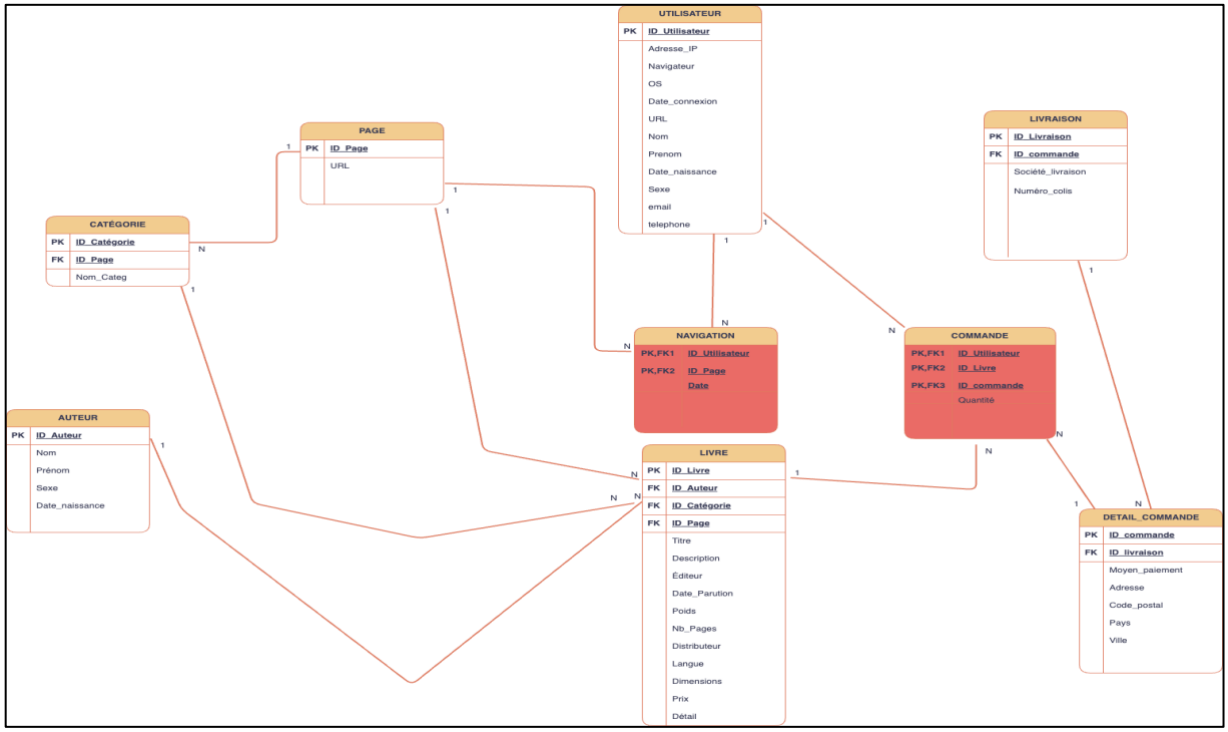
Nous avons créé une table intitulée **Page** qui comporte un identifiant page (ID_page) ainsi que l'URL de chaque page de notre site. Cela a pour objectif de savoir quelle page chaque utilisateur à consulter.

La table **Navigation** est une table représentant les actions des utilisateurs sur le site, cela va donc modéliser le changement ou le rafraîchissement d'une page via un click ou une touche sur le clavier. Cette table comporte également un attribut **Date** qui nous permet de suivre le parcours des utilisateurs dans la durée.

La table **Commande** peut contenir plusieurs fois le même numéro de commande, avec cependant un contenu de commande différents entre les observations. C'est dans la table **detail_commande** que toutes les commandes sont regroupées par leur identifiant unique.

2.3 – Modèle Relationnel

Table de dimensions : Livre, Auteur, Catégorie, Utilisateurs, Livraison, Detail_Commande, Page
Tables de Fait : Commande et Navigation



Nous avons organisé notre modèle en flocon, de sorte à avoir un côté plutôt axé sur les commandes et leur gestion. C'est la partie droite du graphique avec les tables **commande, detail_commande et livraison.** Ces tables ne nous permettent pas de répondre directement à notre problématique, mais elles restent très importantes dans la structure du site et de l'entreprise.
La partie gauche, nous donne beaucoup plus d'informations sur le site en lui-même, mais aussi sur le comportement des utilisateurs sur le site de Nil. On y trouve des tables avec énormément d'informations telles que la table **Livre, Utilisateur, Auteur…**

Représenté, notre modèle de cette manière nous a permis d'avoir une représentation simplifiée et synthétique de nos données. On constate sur le modèle également, les nombreuses relations entre les tables de dimensions qui sont l'essence même d'un modèle en flocon. Elles nous permettent ici, de pouvoir obtenir un maximum d'informations sur le comportement des utilisateurs. Cela facilite grandement les analyses dans la partie ultérieure.

2.4 – Explications des relations pour chaque table

Livre
La table Livre est essentielle, car elle est liée à 4 autres tables (**Auteur, Catégorie, Page et Commande**).
Un livre a été écrit par un auteur. Cependant, un auteur peut avoir écrit plusieurs livres.
Un livre appartient obligatoirement à une catégorie. Une catégorie peut contenir plusieurs livres.
Un livre possède une page de description. Une page peut contenir plusieurs livres, par exemple une page d'accueil de catégorie.
Un livre peut être dans plusieurs commandes. Cependant, une ligne commande dans la table commande peut contenir qu'un livre.

Catégorie
Une catégorie peut contenir plusieurs livres. Un livre appartient obligatoirement à une catégorie.
Une catégorie ne peut avoir qu'une page d'informations. Plusieurs pages, notamment de description livre peuvent appartenir à une catégorie.

Page
Une page peut contenir plusieurs livres, par exemple une page d'accueil de catégorie. Un livre possède une page de description.
Une page peut être associée à plusieurs navigations. Cependant, la table navigation qui correspond à l'action d'un utilisateur sur une page ne peut être que d'une page à la fois.

Utilisateur
Un utilisateur peut avoir différentes navigations, s'il navigue dans plusieurs pages du site. Cependant, une navigation qui correspond à une action sur le site ne peut être fait que par un utilisateur respectif.
Un utilisateur peut avoir différentes commandes sur le site Nil. Cependant, une commande ne peut appartenir qu'à un seul utilisateur.

Commande
Une commande ne peut appartenir qu'à un seul utilisateur. Cependant, un utilisateur peut avoir différentes commandes sur le site.
Il ne peut y avoir qu'un numéro de commande unique dans la table detail_commande. Cependant, la table commande peut contenir plusieurs identifiant commande.

Livraison
La table livraison peut contenir plusieurs identifiants commandes de la table detail_commande. Mais une livraison ne peut être associée qu'à une commande.

Auteur
Un auteur peut avoir écrit plusieurs livres. Mais un livre n'appartient qu'à un auteur.

Navigation
Une navigation peut être fait par un utilisateur. Un utilisateur peut faire N navigation.

Detail_commande
Il peut y avoir N identifiant commande dans la table commande. Mais un seul identifiant commande unique dans la table detail_commande.

Ces liens entre les tables sont très importants pour la cohérence et l'analyse des données.

III – Granularité

Les tables de faits vont se rapporter à une granularité définie par les besoins. La granularité est de détail nécessaire à l'analyse du fait, choisie en amont. On distingue alors trois grandes familles de granularité :

- Granularité de **transaction** : mesure unitaire d'une action métier.
- Granularité de **l'instantané périodique** : mesure des actions métier dans un intervalle fixe (semaine, mois, ...).
- Granularité de **l'instantané récapitulatif** : mesure des actions métier sur une période non prédéterminée

Le choix de la granularité des tables de faits dépend des besoins spécifiques du système d'informations relationnels. Chaque niveau de granularité comporte des avantages et des inconvénients.

En ce qui concerne notre table **Commande**, nous disposons d'une ligne par commande, le chargement de la table de faits se fait par insertion. Ensuite, nous n'avons pas de mise à jour de cette table. La dimension date est représentée par la date de la transaction. Enfin, sa période dans le temps est limitée par un point fixe.
La granularité de notre table commande est donc transactionnel.
On peut expliquer cela par le fait que l'entreprise, doit être au courante dès qu'il y a une commande afin de pouvoir respecter son délai de livraison et livrer dans les meilleurs délais.

Pour la table **Navigation**, chaque observation de cette table représente une transaction individuelle. Chaque ligne contient des détails spécifiques sur une interaction ou un événement à un moment précis entre l'utilisateur et le site internent. Par exemple, le moment où ce dernier à visiter une page. La granularité est également transactionnelle. Nous avons une ligne par événement, il n'y a pas de mise à jour des faits. Nous mesurons la dimension date à l'aide de la date de l'interaction avec le site.

En résumé, les deux tables, **Commande** et **Navigation**, présentent une granularité transactionnelle, reflétant chacune des interactions ou des transactions spécifiques au sein de leur domaine respectifs. Ici, la granularité transactionnelle va offrir une analyse détaillée, permettant une compréhension fine du comportement des utilisateurs et des détails spécifiques de chaque commande. Cette approche va favoriser la réactivité opérationnelle, car elle informera immédiatement l'entreprise Nil de chaque événement, tel qu'une commande passée, contribuant ainsi à respecter les délais de livraison et à optimiser la satisfaction client.

Cependant, il est important de noter que la granularité transactionnelle entraînera probablement un volume de données important. La gestion de ce volume pourra devenir complexe à terme, affectant potentiellement les performances du stockage et des requêtes. Les requêtes sur des données

transactionnelles pourront également devenir plus complexes en raison du grand nombre d'enregistrements, nécessitant une attention particulière pour optimiser les performances du système.

IV – Type de gestion des changements

Les valeurs des dimensions peuvent être amenées à évoluer. Par exemple, dans notre cas un livre peut changer d'édition, de nom… Une considération importante de ce problème est importante afin de pouvoir gérer ces changements. Il y a donc plusieurs solutions pour gérer ce type de situation.
Il y a le :

- Type 1 : Écrasement de l'ancienne valeur par la nouvelle
- Type 2 : Ajout d'une ligne pour intégrer la modification
- Type 3 : Ajout d'une colonne de dimension

Voici le tableau avec le type de gestion des changements des valeurs dans les tables de notre modèle :

Nom de la table	Type de gestion de changements
Utilisateur	Type 1
Livre	Type 1
Auteur	Type 1
Catégorie	Type 1
Page	Type 2
Navigation	Type 2
Commande	Type 2
Detail_Commande	Type 2
Livraison	Type 1

Pour choisir le type de gestion des changements approprié à chaque table, nous nous sommes efforcés de rester étroitement liés à notre sujet afin de déterminer la solution la plus adaptée.

Nous avons choisi le type 1 qui correspond à l'écrasement de l'ancienne valeur par la nouvelle pour les tables suivantes :

- **Utilisateur** : Nous procédons au remplacement d'informations périmées, par exemple, si un client modifie ses données nominatives (nom, prénom, sex …). Il est superflu de conserver les anciennes données, nous les substituons par les plus récentes.
- **Livre** : Il est inutile de conserver des informations obsolètes dans cette table. Par exemple, si un livre n'est plus produit par la même édition, nous le remplaçons par celui produit par la nouvelle édition.
- **Auteur** : Similaire à la table **Utilisateur**, si un auteur modifie ses données nominatives, nous effectuons le remplacement de l'ancien par le nouveau afin de maintenir des données actuelles.
- **Catégorie** : En principe, les noms des catégories ne devraient que rarement changer. Toutefois, en cas de modification, nous substituons l'ancienne par la nouvelle pour garantir la cohérence des données, car un livre appartient obligatoirement à une catégorie.

Ensuite, les tables suivantes sont du type 2 correspondant à l'ajout d'une ligne pour intégrer la dimension :

- **Navigation** : Dans la table navigation, chaque ligne correspond à l'interaction d'un utilisateur avec le site. Le type de changement se fait obligatoirement par l'ajout d'une ligne, car la ligne correspondant à l'interaction ne peut être modifiée.
- **Commande** : Dans cette table, le type de changement se fait par l'ajout d'une ligne. Lorsque nous avons plusieurs commandes avec le même identifiant mais un livre différent, nous

ajoutons une ligne. Car une fois la commande validée, il est impossible de modifier son contenu en écrasant l'ancien contenu.

V – Dimension douteuse, causale, dégénéré

Dans l'objectif, de mieux appréhender la dynamique complexe qui sous-tend la gestion de nos données, et permettant des analyses plus précises et de la compréhension du comportement des utilisateurs. Les dimensions jouent un rôle fondamental en structurant les données, apportant ainsi une cohérence et une signification cruciale à l'ensemble du système d'information décisionnel.
Parmi les diverses catégories de dimensions, nous nous penchons sur trois types spécifiques afin de saisir pleinement leur impact sur la modélisation de nos données :

- **Dimension douteuse** : Dimension pour laquelle le même attribut peut apparaître plusieurs fois (doublons)
- **Dimension causale** : Dimension qui provoque le fait
- **Dimension dégénérée** : Dimension sans attribut (donc sans table). La dimension est intégrée à la table de fait

Les dimensions douteuses dans un modèle de données peuvent survenir lorsque le même attribut peut apparaître plusieurs fois, introduisant potentiellement des doublons ou des ambigüités. Dans notre modèle, les dimensions douteuses présentes sont :

- **Utilisateur** : Si deux utilisateurs possèdent les mêmes données nominatives, malgré qu'ils aient deux identifiants uniques distincts. Nous pouvons les confondre et penser que c'est la même personne.
- **Livraison** : Si un utilisateur modifie son adresse de livraison fréquemment, chaque mise à jour écrasera l'ancienne adresse, mais plusieurs enregistrements pour la même adresse de livraison pourraient persister.
- **Auteur** : Comme pour la table **Utilisateur,** dans le cas ou deux auteurs possèderaient les mêmes données nominatives, malgré qu'ils aient deux identifiants uniques distincts. Nous pouvons les confondre et penser que ce sont les mêmes auteurs.
- **Livre** : Chaque livre dispose d'un identifiant unique, cela n'empêche pas qu'il puisse y avoir des ambiguïtés ou des doublons. Il peut avoir y avoir plusieurs livres qui ont le même titre, qui sont parus à la même date…En somme, le doute peut être présent dans ce genre de situation.

Dans notre modèle relationnel, nous n'avons pas de dimension dégénérée.

VI – Développement de l'entrepôt de données

Nous avons réalisé ce projet en plusieurs phases. A travers, les étapes vues précédemment nous avons conçu notre modèle relationnel. Ce dernier est très important, car il nous sert de base solide pour la suite du travail. Ce travail a été réalisé sans outils techniques particuliers, mais plutôt avec beaucoup d'échange et de réflexion. Il était également très important de ne pas s'éloigner de l'objectif final, qui est de permettre à l'entreprise Nil d'analyser le comportement des utilisateurs sur sa plateforme.

Dans la deuxième partie, nous avons mis en production le modèle relationnel réfléchi précédemment. Plus concrètement, nous avons conçu notre base de données au format .sqlite. Ce format permet de travailler un faible besoin en mémoire et requiert peu de performances nécessaires pour le serveur.
La base a été construite sur un logiciel de programmation Rstudio qui nous a servis d'éditeur de texte.

Nous avons utilisé le package **RSQLite** et **DBI**. Le premier, nous a permis d'interagir avec notre base de données dans R tout en utilisant le langage SQL. Ceci a été très pratique car SQL permet de communiquer très simplement avec une base de données. Tandis que le second package, nous a permis

de disposer d'une interface générique pour interagir avec notre base de données. Les packages ont été complémentaires pour notre travail.

Une fois la base construite, il nous a fallu peupler notre table. Pour cela, nous avons créé un fichier Excel avec plusieurs feuilles correspondant à chaque dimension. Nous avons généré des données en utilisant des listes de livres trouvées sur internet par exemple ou encore en utilisant CHATGPT qui nous a permis de gagner du temps.
Cette étape réalisée, nous avons ensuite importé nos données dans notre base de données. A la suite de ça, nous avons résolu un certain nombre de problèmes sur R qui étaient essentiellement des problèmes liés au format de la date dans notre base.

Afin d'être sûr de la qualité de nos données, nous avons regardé si l'importation était bonne. C'est-à-dire, qu'il ne manque rien à notre base, qu'il n'y est pas de valeur manquante, que le format des variables soit respecté…

Enfin, nous avons pu construite notre Dashboard et exécuté nos requêtes.

VII – Caractéristiques des faits

L'additivité se réfère généralement à la propriété d'ajouter ou de combiner différentes quantités ou éléments pour obtenir une somme totale. L'additivité est une propriété importante dans la conception et l'interprétation des systèmes d'informations décisionnels, car elle permet de simplifier les calculs et de faciliter la compréhension des données agrégées.
Il y a plusieurs propriétés d'additivité :

- Fait additif : additionnable suivant toutes les dimensions
- Fait semi additif : additionnable seulement suivant certaines dimensions
- Fait non additif : Non additionnable, peu importe la dimension

Notre modèle relationnel, contient ces trois propriétés d'additivité.

Fait additif

La première mesure additive ici, est le montant total des commandes réalisées par les clients sur le site de Nil.
Cela, car la somme du montant total des livres commandées est égale à la somme du montant total reçu par l'entreprise. Cette mesure est additive notamment sur la dimension des livres, car en additionnant cette mesure nous ne perdons pas de sens.

La seconde mesure additive est le nombre total de livres commandé. La somme des livres commandés par les utilisateurs, est égale au nombre total de livre commander à l'entreprise. Nous pouvons additionner cette mesure, sans prendre en compte la date et cela restera assez pertinent. On peut agréger le nombre total de livres commandés sur différentes dimensions comme le client, la catégorie de livres, etc., sans perdre de précision ni de pertinence.

Ces mesures additives nous offrent une flexibilité dans l'analyse, car nous obtenons des informations agrégées à différents niveaux de granularité.

Fait semi-additif

Dans la catégorie des mesures semi-additives, il y a le nombre total de pages distinct vu par utilisateur. Cette mesure est semi-additive, parce que l'on peut agréger selon la dimension utilisateur, mais pas par la dimension auteur par exemple. Le nombre de pages distinctes associé aux auteurs n'est pas égal au nombre total de page. Dans ce cas, la mesure ne sera pas pertinente.

La seconde mesure est le nombre total d'auteurs. Cette mesure est semi-additive, car elle n'est tout simplement pas additive par la dimension catégorie. La somme du nombre d'auteurs n'est pas égale à la somme du nombre total d'auteurs par catégorie étant donné qu'un auteur peut appartenir à plusieurs catégories s'il a écrit un roman et une bd par exemple.

Fait Non-additif

Enfin, dans cette catégorie de mesures, nous avons le nombre moyen d'articles par commande. La mesure n'est tout simplement additive, car nous ne pouvons pas sommer des moyennes. Cela conduirait à des résultats trompeurs.

Une autre mesure également non additive est la marge bénéficiaire par catégorie réalisée par l'entreprise. La marge bénéficiaire correspond au ratio du bénéfice. La somme n'est donc pas possible. Il peut également y avoir le pourcentage de livre de la catégorie fantastique pour la même raison que la mesure précédente.

Enfin, nous avons écrit les requêtes SQL, permettant d'obtenir ces mesures sur notre base. Elles sont dans notre code commenté.

VIII – Conclusion

Grâce à notre modèle relationnel, nous avons résolu notre problématique initiale qui consistait à analyser le comportement des utilisateurs sur le site de Nil. Notre base de données regorge d'informations riches, offrant une compréhension approfondie du comportement des utilisateurs. En allant au-delà, nous avons également des données détaillées sur les clients et leurs commandes. La structure adoptée ici se distingue par son efficacité, facilitant une analyse approfondie et pertinente.

Le sujet du projet était très intéressant. Il nous a vraiment permis de mettre en pratique tous que nous avons appris en cours de système d'informations décisionnelles. Cela nous a énormément plu. La phase de construction du modèle a exigé un investissement significatif. Le choix des tables ainsi que leurs liens nous ont pris beaucoup de temps. C'était réellement le cœur du problème et de la demande de l'entreprise Nil. La réalisation de la base a été plus facile et plus rapide. Lors de la réalisation du dashboard nous étions très contents lorsque nous avons pu avoir des représentations graphiques de nos données. Bien qu'elles soient fictives.
Nous avons beaucoup aimé travailler ensemble. La collaboration et l'échange nous ont permis d'apprendre des autres.