IUT de Paris - Rives de Seine
Université Paris Cité

Implémentation de la base de données Nil

projet SID

CHEN LAURENT
CHEN PATRICK
COADOU TRISTAN

2AFA VCOD 14/01/24

Contexte

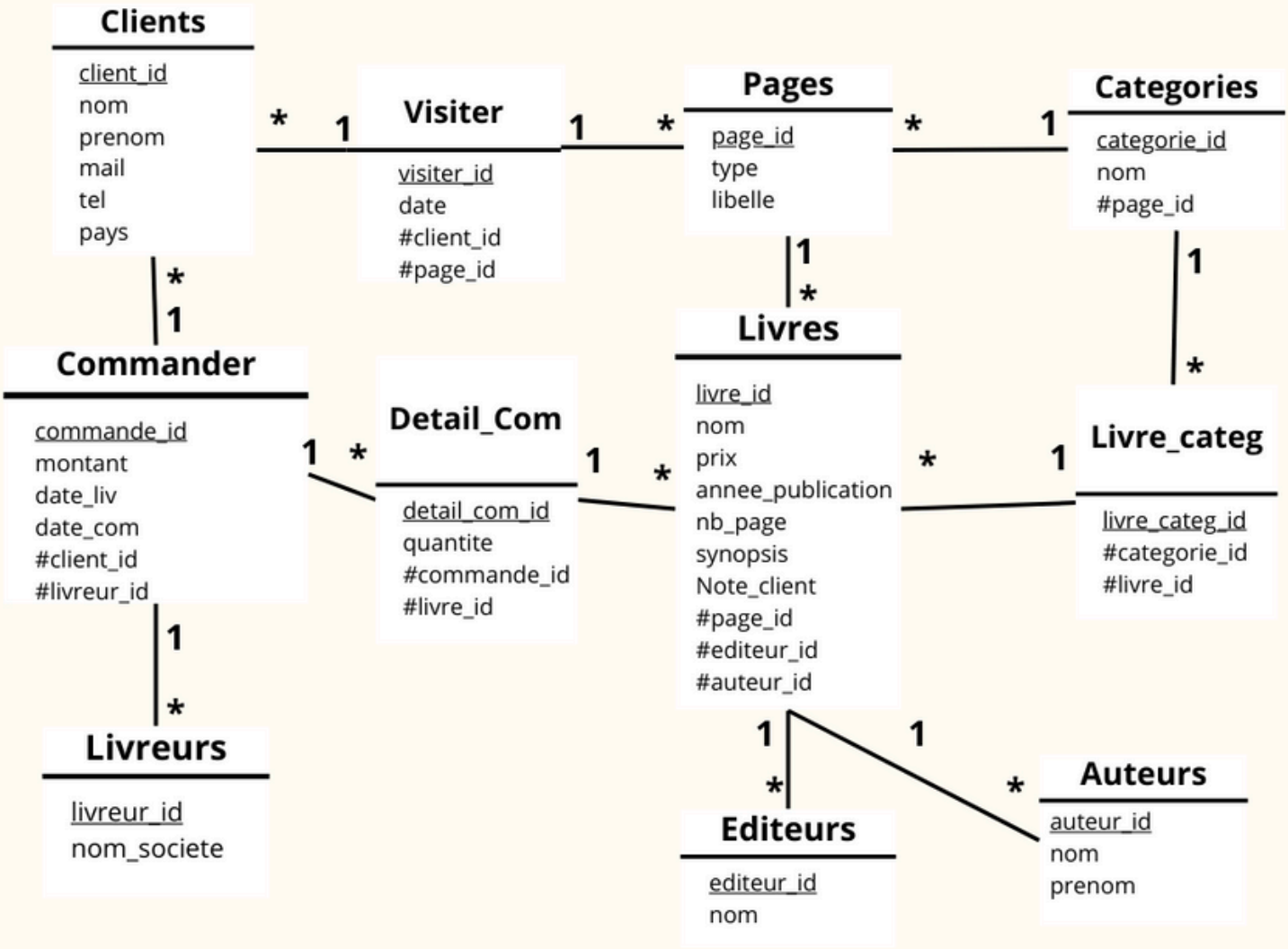
Nous sommes sollicités par l'entreprise française de e-commerce de livres Nil. L'objectif est de créer une base de données exploitable sous SQL, permettant de comprendre le comportement des utilisateurs sur la plateforme.

La base de données devra remplir certaines conditions de notre commanditaire pour un traçage optimal côté client avec le suivi des commandes : détails des commandes; pages consultées; des livraisons. Concernant le côté de l'entreprise Nil nous avons les informations suivantes : l'éditeur; le livreur; la page web.

Concernant le déroulé du projet, dans un premier temps, nous avons commencé par concevoir les tables de notre base de données, à savoir définir le nom et la fonction de chaque table, puis nous avons ajouté pour chaque table des attributs en fonction des attendus du commanditaire. Pour relier chaque table à l'aide de relation, nous avons pensé un modèle relationnel des données en attribuant des cardinalités pour chaque liaison. Nous avons créer les tables ainsi que nos relations dans le logiciel R à partir des librairies DBI, RSQLite. Pour la phase d'implémentation des données, nous avons préféré utiliser des bases existantes sur le web, nous avons utilisé des bases de clients (nom, prénom et numéro de téléphone) et de livres (titre, auteur, année de parution, etc...) sur le site kaggle. Puis, nous avons effectué des requêtes SQL pour tester notre base de données et nous assurer que tout soit opérationnel : vérifier la cohérence des données, des relations et les format des dates. Enfin, nous avons réalisé une maquette de dashboard sur les mesures listées.

Modélisation relationnelle

Schéma relationnel Nil



Les attributs soulignés représentent les clés primaires. Et les attributs précédés d'un # représentent les clés externes.

Tables de Fait

- **Visiter**
- **Commander**

Tables de Dimensions

- Clients
- Pages
- Categories
- Livre_categ
- Livres
- Detail_com
- editeurs
- auteurs
- Livreurs

Explication des choix

Les deux tables de fait 'Visiter' et 'Commander' ont été créé pour répertorié les actions des utilisateurs, ce sont donc des tables de fait. Tous les autres tables sont des tables de dimensions car ils représentent les entités.

Livre_categ a pour but de gérer les relations NN. En effet, un livre peut appartenir à plusieurs catégories. Et une catégorie a plusieurs livres. Detail_com a le même but parce que ici une commande peut comporter plusieurs livres, mais chaque livre peut appartenir à plusieurs commandes différentes, mais cette tables sert aussi à répertorier la quantité de livre commandé.

Pour la table Pages, le type représente soit accueil, soit catégorie, soit livre. Le libellé renvoie le détail du type.

Les tables de fait ont une granularité de transaction car une ligne représente une unité métier. Dans la table visiter, une ligne représente une visite. Dans la table commander, une ligne représente une commande.

Type de gestion des chargements des tables de dimension

Table	Type	Justification	Dimension
Clients	1	L'historique n'est pas important.	Douteuse Un client peut avoir plusieur compte.
Pages	1	L'historique n'est pas important.	/
Categories	1	L'historique n'est pas important.	/
Livre_categ	1	L'historique n'est pas important.	/
Livres	1	L'historique n'est pas important.	Douteuse Un livre peut être édité par plusieurs édition.
Detail_com	2	L'historique peut être intéressant à conserver pour le client. L'ajout de ligne est mieux car pas toutes les commandes risquent d'être modifié.	/
Editeurs	1	L'historique n'est pas important.	/
Auteurs	1	L'historique n'est pas important.	/
Livreurs	1	L'historique n'est pas important.	/

Création de la base de données

Les librairies nécessaires

```
library(DBI) # Connecte R à la base de données
library(RSQLite) # Permet de faire des rêquetes sql sur R
library(readxl) # Permet de lire les fichiers excel
```

Création du fichier sqlite

```
chemin = "C:/user/..../Nil_DataWarehouse.sqlite"
# Création du fichier sqlite
data = dbConnect(RSQLite::SQLite(), chemin)
```

Exemple type d'une création de table

```
# Création de la table Visiter
dbExecute(data, "CREATE TABLE Visiter (
    Visiter_id BIGINT,
    Client_id BIGINT,
    Page_id BIGINT,
    Date TIMESTAMP,
    CONSTRAINT pk_vid PRIMARY KEY (Visiter_id),
    CONSTRAINT fk_clid FOREIGN KEY (Client_id) REFERENCES Clients (Client_id),
    CONSTRAINT fk_pid FOREIGN KEY (Page_id) REFERENCES Pages (Page_id)
);")

# Vérification
dbListFields(data, "Visiter")

#implémentation des données
Visiter <- read_excel("C:/Users/.../SID_database.xlsx", sheet = "Visiter")
dbWriteTable(data, name = "Visiter", value = Visiter, append = TRUE)

#Visualisation des données
dbGetQuery(data, "SELECT * FROM Visiter")
```

Les tables Clients et Pages ont été créé en amont avant la table Visiter.
En effet, il faut toujours créer la table des clés externes en premier.

Implémentation de données

Dans la phase d'implémentation, nous avons choisi de réaliser un fichier Excel contenant une feuille pour chaque table qui sera implémenté par la suite. Cette solution était pour nous la plus efficace puisqu'elle nous évitait de rentrer les informations des tables ligne par ligne. Nous avons donc pris comme base 2 base de données issus du site web Kaggle : un fichier client et un fichier livre.

Nous avons par la suite arrangé les fichiers de sorte à ce qu'elles soient cohérentes dans notre jeu de données, comme l'ajout des ID et d'autres colonnes spécifique à chaque table.
Exemple table livre :
Fichier d'origine :

Livre_id	Nom	Prix	Nb_page	Note_client	Annee_publication
200000	Iron Flame (The Empyrean, 2)	18,42	107	4,1	2023
200001	The Woman in Me	20,93	245	4,5	2023
200002	My Name Is Barbra	31,50	169	4,5	2023
200003	Friends, Lovers, and the Big Terrible Thing: A Memoir	23,99	191	4,4	2023
200004	How to Catch a Turkey	5,65	110	4,8	2018
200005	Fourth Wing (The Empyrean, 1)	16,99	188	4,8	2023
200006	No Brainer (Diary of a Wimpy Kid Book 18)	8,55	185	4,8	2023
200007	Killers of the Flower Moon: The Osage Murders and the Birth of the FBI	9,86	156	4,4	2017
200008	All the Light We Cannot See: A Novel	11,98	224	4,5	2014
200009	The Ballad of Songbirds and Snakes (A Hunger Games Novel) (The Hunger Games)	10,99	154	4,6	2020

Fichier final :

Livre_id	Nom	Prix	Nb_page	Synopsis	Note_client	Editeur_id	Auteur_id	Page_id	Annee_publication
200000	Iron Flame (The Empyrean, 2)	18,42	107		4,1	400003	300000	600000	2023
200001	The Woman in Me	20,93	245	zef	4,5	400000	300001	600001	2023
200002	My Name Is Barbra	31,50	169	fzefzefz	4,5	400001	300002	600002	2023
200003	Friends, Lovers, and the Big Terrible Thing: A Memoir	23,99	191	zef	4,4	400003	300003	600003	2023
200004	How to Catch a Turkey	5,65	110		4,8	400003	300004	600004	2018
200005	Fourth Wing (The Empyrean, 1)	16,99	188		4,8	400000	300000	600005	2023
200006	No Brainer (Diary of a Wimpy Kid Book 18)	8,55	185	fzefzefz	4,8	400000	300005	600006	2023
200007	Killers of the Flower Moon: The Osage Murders and the Birth of the FBI	9,86	156	zfzzef	4,4	400002	300006	600007	2017
200008	All the Light We Cannot See: A Novel	11,98	224		4,5	400002	300007	600008	2014
200009	The Ballad of Songbirds and Snakes (A Hunger Games Novel) (The Hunger Games)	10,99	154	fzefzefz	4,6	400001	300008	600009	2020

Ajout des colonnes : Synopsis - Editeur_id - Auteur_id - Page_id

Pour le reste des tables, nous les avons créé sous Excel en ajoutant les colonnes établies dans notre modèle relationnel. Puis nous avons entré des valeurs associées à ces colonnes

Requêtes SQL

On souhaite vérifier que notre base de données soit fonctionnelle. Pour cela, on réalisera 6 requêtes SQL fractionnées en fonction de chaque niveau d'additivité. Concernant les mesures, nous avons choisi :

Pour les faits additifs :

Le chiffre d'affaire est un fait additif car il est additif :

- En lieu : on peut additionner les montants payés en fonction des pays
- En temps : on peut additionner les chiffres d'affaires mensuels entre un mois A et un mois B
- En produit : on peut additionner les chiffres d'affaires en produit en fonction des livres

Nombre de ventes est un fait additif car il est additif :

- En lieu : on peut additionner le nombre de ventes entre pays
- En temps : on peut compter le nombre de ventes d'un jour A et B et en faire la somme
- En produit : on peut additionner le nombre de ventes en produit, en fonction du livre

Pour les faits semi additifs :

Nombre de clients est un fait semi additif :

- En lieu : on peut additionner le nombre de clients de chacun pays
- En temps : on ne peut pas additionner le nombre de clients en temps car il peut y avoir des doublons
- En produit : on ne peut pas additionner le nombre de clients en produit, car un client peut acheter plusieurs livres et pourrait être comptabilisé plusieurs fois

Nombre de colis restant à livrer à un instant donnée est semi additif :

- Le nombre de colis restant à livrer à un instant donnée n'est pas additif sur le temps mais l'est en lieu. On ne peut pas additionner le nombre de colis restants d'un jour à l'autre.

Pour les faits non additif :

Les ratio/moyenne ne sont pas additif en produit, lieu et temps. Cela comprend :

La moyenne du prix des catalogues

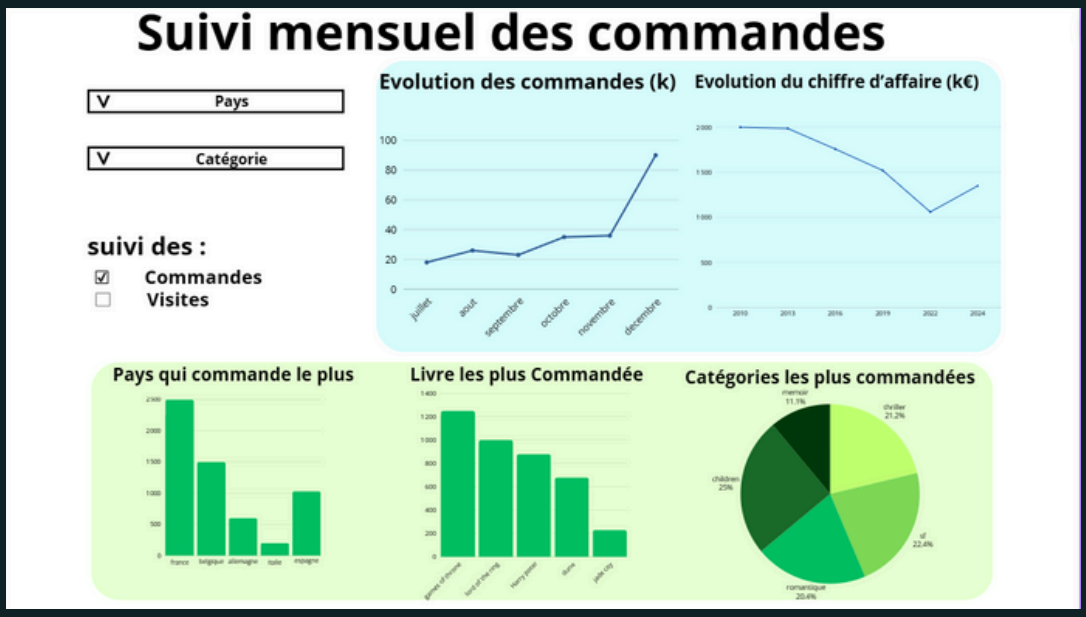
moyenne des délais de livraison

Requêtes SQL

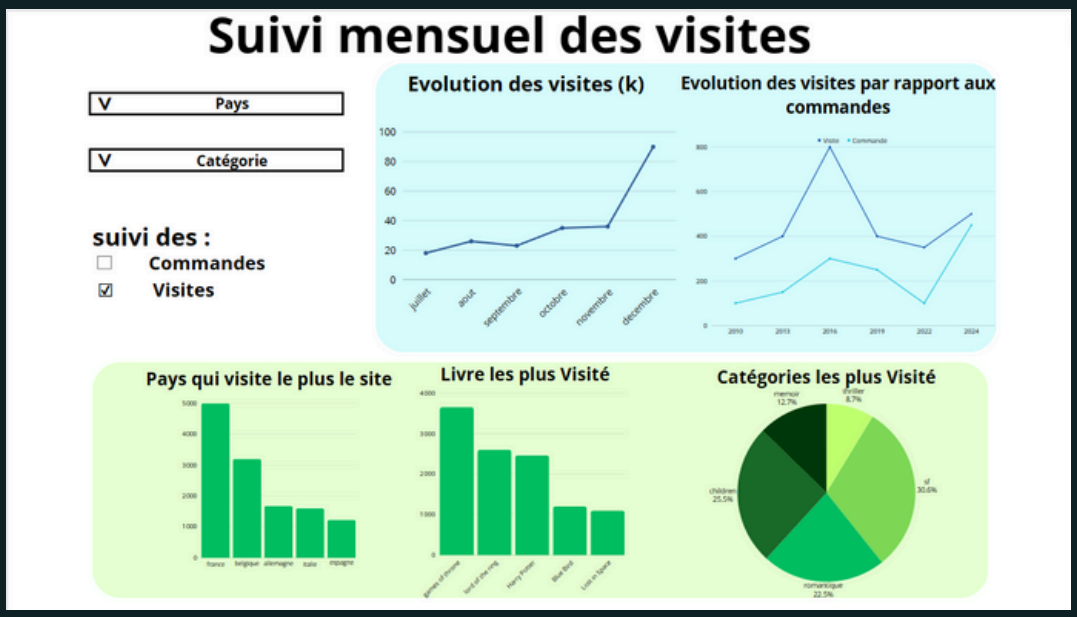
```
#additif
# Chiffre d affaire :
dbGetQuery(data, "SELECT sum(Montant) from Commander;")
# Nombre de ventes :
dbGetQuery(data, "SELECT count(commande_id) from Commander;")


#semi additif
# Nombre de commandes restantes à partir du 01/12/2024 :
dbGetQuery(data, "SELECT * from Commander where date_liv > '2023-12-01 00:00'")
# Nombre clients :
dbGetQuery(data, "SELECT count(Client_id) from Clients;")


#non additif
# Moyenne du prix des livres du catalogue :
dbGetQuery(data, "SELECT ROUND(AVG(Prix),2) From Livres;")
# Moyenne des delai :
dbGetQuery(data, "SELECT ROUND(AVG(Date_liv - Date_com), 2) from Commander;")
```

Maquettes de dashboard :

- **Dashboard des commandes :**



- **Dashboard des visites :**



Conclusion

Afin de comprendre le comportement des utilisateurs sur la plateforme Nil, nous avons réalisé la conception relationnelle des données, la création des tables sur R avec les requêtes SQL, et l'implémentation des données préparée sous Excel. Bien évidemment la structure de notre base permettra la mise en place de mise à jour, à savoir l'insertion de nouvelles données comme la création de profil de nouveau clients, de produits ou encore de commandes.
Avec cette base de données, cela nous permettra de centraliser, organiser et sécuriser les données, facilitant ainsi leur accès, leur manipulation et leur gestion. Concernant le suivi de l'activité de la plateforme Nil, nous pouvons ressortir un grand nombre d'information, comme le chiffre d'affaires, le nombre de ventes, le nombre de clients, etc. Enfin, pour que les directeurs de Nil puissent avoir un aperçu de l'évolution de leur business, et pour la prise de décisions, nous avons proposé des maquettes de Dashboard.