

Compte rendu projet Base de données



Table des matières

Introduction	3
Besoins	3
Approche méthodologique	3
Conception de la base de données.....	4
Remplissage de la base de données	7
Génération des scripts.....	7
Conception du programme	8
Stack du programme	8
Infrastructure du programme	8
Principe du programme.....	8
Conclusion.....	9

Introduction

Ce projet a pour but de nous apprendre l'utilité ainsi que l'implémentation des bases de données. Pour cela, nous devons implémenter la propre base de données pour le système de télécommunication de l'entreprise ComPCOM.

Le système de l'entreprise ComPCOM permet à ses clients de diffuser des messages (messages d'urgence par exemple) ou des offres commerciales.

Afin d'implémenter notre base de données, nous avons d'abord modéliser les schémas de la base de données (MCD, MLD), puis nous la créons sous PostgreSQL. Par la suite, nous avons écrit les requêtes SQL permettant de répondre aux besoins du client.

Besoins

Les besoins de l'entreprise ComPCOM sont les suivants :

- Proposer une solution de structure de base de données
 - o Modèle conceptuel de données (MCD)
 - o Modèle logique de données (MLD)
- Implémenter la base de données sous PostgreSQL
 - o Générer un script permettant de mettre en place la base de données modéliser au préalable
- Ecrire les requêtes SQL permettant de répondre aux besoins des clients
- Implémenter un programme permettant de dialoguer avec la base

Approche méthodologique

Afin de répondre aux besoins énoncés ci-dessus, nous avons divisé notre travail en plusieurs parties :

- La modélisation du MCD (modèle conceptuel de données)
- La modélisation du MLD (modèle logique de données)
- La génération du script permettant de créer la base de données telle que nous l'avons modélisé dans le MCD
- L'écriture des requêtes SQL permettant de vérifier le contenu des tables
- L'écriture des requêtes SQL permettant de répondre aux besoins des clients (modifications, ajouts...)
- Implémenter un programme permettant de dialoguer avec la base de données

Ces différentes étapes nous permettent d’avoir un fil directeur dans l’implémentation de notre base de données. Elles nous permettent ainsi d’avoir une vue d’ensemble sur comment la base de données va être conçue :

- Le MCD nous permettra d’avoir une représentation graphique de comment nous souhaitons implémenter nos tables et comment ces dernières seront liées entre elles.
- Le MLD nous permettra d’avoir d’une représentation « écrite » de notre base de données nous permettant de visualiser sa structure. Ce modèle peut être généré à partir du MCD.
- Les étapes suivantes nous permettent d’implémenter notre base de données. C’est-à-dire sa création, ses interactions ainsi que son interface homme-machine.

Nous avons donc utilisé JMerise pour modéliser les modèles, puis PostgreSQL pour la création et le remplissage de la base de données.

Conception de la base de données

Afin d’avoir une idée claire et précise de la structure de notre base de données, nous avons repris les diagrammes créés lors de notre projet d’UML (voir figure suivante).

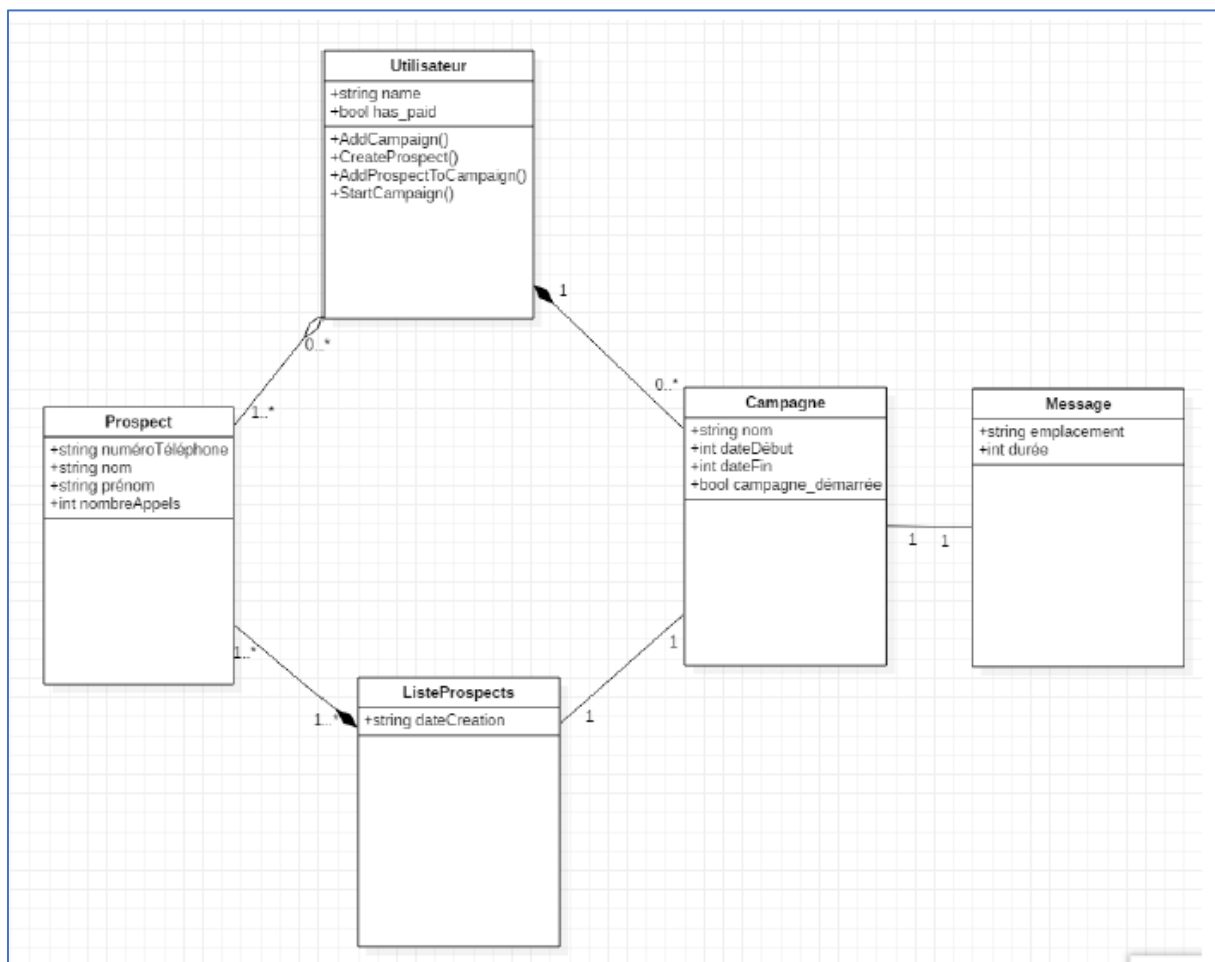
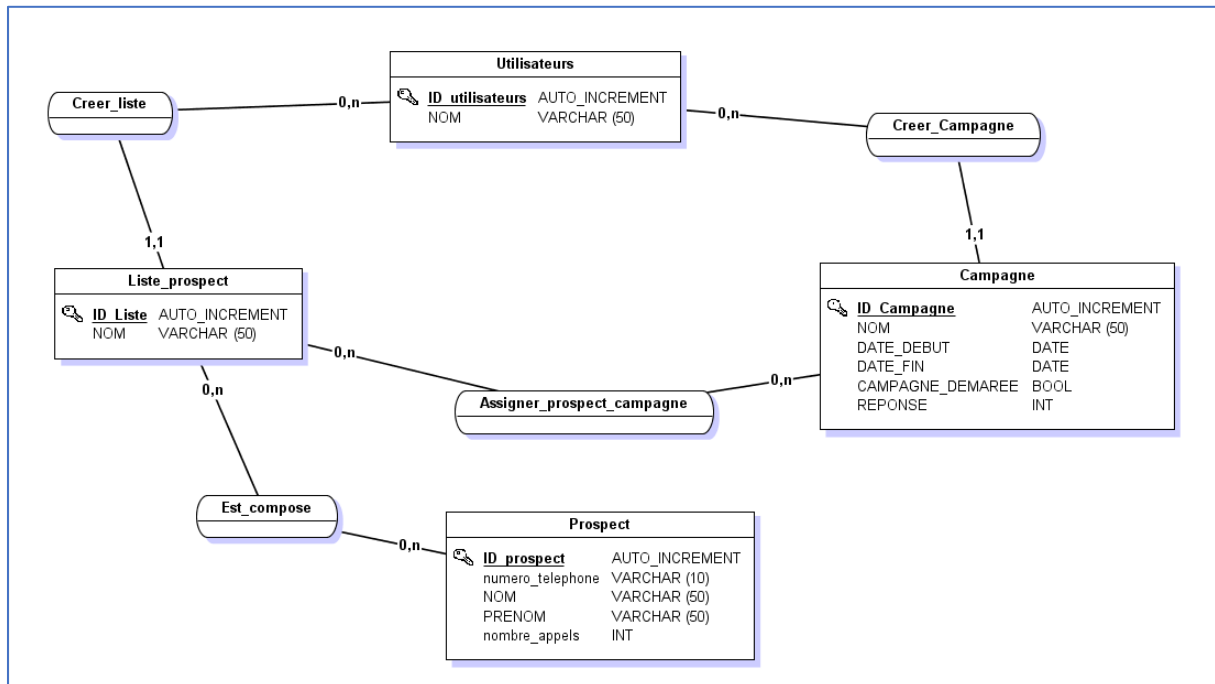


Diagramme des classes du projet d’UML

A partir de ce diagramme, nous avons effectué le MCD suivant :



Modèle conceptuel de données

Ce modèle a été modélisé avec JMerise. Nous avons repris à peu près la même structure que sur notre diagramme des classes. Cependant nous avons apporté des modifications car dans le cas d'une base de données. En effet, de nouvelles tables peuvent être créées en fonction des relations entre les tables et des cardinalités. De ce fait, nous avons dû implémenter notre base de façon différente :

- Table « **Utilisateurs** » : la table utilisateur concerne la liste des clients de ComPCOM qui utilise le service de l'entreprise. Nous avons décidé d'uniquement lui assigner un « **Nom** » ainsi que « **ID_utilisateurs** » (clé primaire) car d'autres attributs seraient hors-périmètre.
- Relation « **Créer liste** » : la relation « créer_liste » est une requête permettant d'ajouter un élément dans la table « **Liste_prospect** ». Dû aux cardinalités de la relation (0,n – 1,1), il ne s'agit pas d'une nouvelle table.
- Table « **Liste_prospect** » : nous avons décidé d'uniquement mettre les attributs « **Nom** » et « **ID_Liste** » (clé primaire). Comme pour la table utilisateurs, d'autres attributs seraient hors-périmètre.
- Table « **Prospect** » : la table « Prospect » possède plus d'attributs : « **ID_prospect** », « **numero_telephone** », « **Nom** », « **Prenom** », « **nombre_appels** ». Il s'agit de toutes les informations utiles afin de prospecter une personne.
- Relation « **Est_compose** » : la relation « Est_compose » est une relation permettant « d'ajouter » un prospect à une liste de prospects. Dû aux cardinalités (0,n – 0,n), la relation « Est_compose » va créer une jointure de table entre « **Prospect** » et « **Liste_prospect** ». Cette table sera composée des attributs « **ID_prospect** » et « **ID_Liste** ». De ce fait elle se présentera de la façon suivante :

ID_Liste	ID_Prospect
4	3

Cela se traduit par « le prospect possédant l'ID_prospect = 3 est présent dans la liste de prospect possédant l'ID_Liste = 4 »

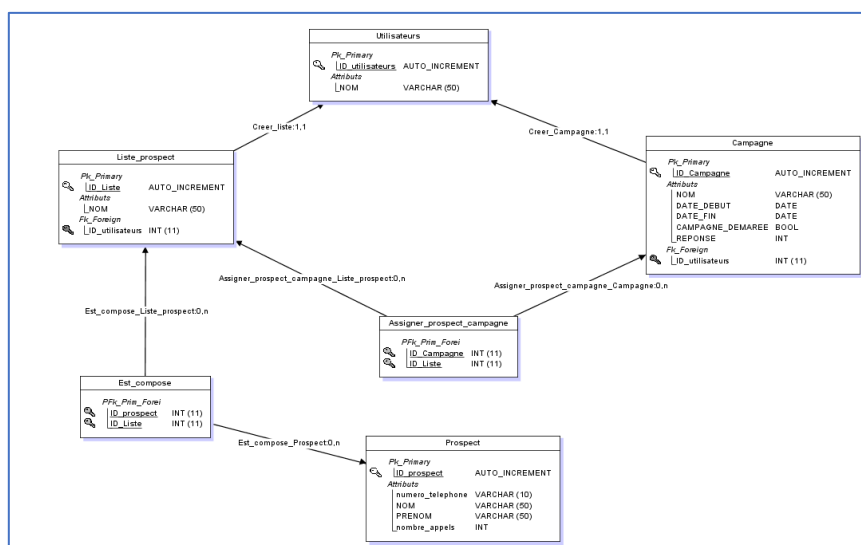
- Table « **Campagne** » : cette table correspond aux données relatives aux informations des messages à diffuser. Les attributs que nous avons mis dedans sont les suivants : « **ID_campagne** » (clé primaire), « **NOM** », « **Date_debut** », « **Date_fin** », « **Campagne_demaree** », « **Reponse** ». Les attributs « Date_debut » et « Date_fin » nous semblaient utiles afin de savoir combien de temps dure la diffusion du message. L'attribut « Campagne_demaree » nous semblait utile afin de savoir si le projet de diffusion n'a pas été abandonné et l'attribut « Réponse » correspond à la réponse d'un prospect à une potentielle question. Il s'agit d'un entier car nous considérons que la réponse est saisie au clavier numérique et qu'il n'y a pas de relation directe entre le client de ComPcom et les prospects.
- Relation « **Assigner_prospect_campagne** » : cette relation permet d'assigner une campagne à une liste de prospect. C'est-à-dire savoir à quelle liste, le message de campagne va être diffusé. Dû aux cardinalités (0,n – 0,n), une jointure de table va se créer. Cette nouvelle table contiendra les attributs « **ID_Liste** » de la table « **Liste_prospect** » et « **ID_Campagne** » de la table « **Campagne** ». La table se présentera de la façon suivante :

ID_Liste	ID_Campagne
7	5

Cela se traduit par « La campagne possédant l'ID_campagne = 5 sera diffusée à la liste de prospect possédant l'ID_prospect = 7 ».

- Relation « **Créer_Campagne** » : cette relation permet de savoir quel utilisateur a créé une campagne de diffusion. Dû aux cardinalités (0,n – 1,1), aucune nouvelle table sera créée.

Grâce à l'application JMerise, nous pouvons générer automatiquement le MLD (modèle logique de données). Pour cela, nous avons vérifié notre conception puis nous l'avons converti. Nous obtenons le MLD suivant :



Aperçu du script de remplissage de la BDD

Conception du programme

Une fois la base de données faite et les requêtes SQL identifiées, nous devons réaliser un programme qui puisse faire l'interface entre la base de données et l'utilisateur.

Stack du programme

Back end en Node.js avec API qui récupère les informations depuis PostgreSQL.

Front end en React qui call les API (Framework CSS/JS Material-ui).

Infrastructure du programme

L'infrastructure est montée avec Docker. Le script est écrit avec la syntaxe docker-compose.yml qui exécute deux containers

- Node.js qui héberge le code front repacké avec webpack
- PostgreSQL qui est initialisé avec les scripts d'exécution que l'on a écrit au préalable

Principe du programme

Une interface simple permettant de :

- Remplir la base de données avec de multiples données que l'on peut remplir avec un formulaire

The screenshot displays the PostgreSQL Console interface. At the top, there is a blue header bar with the text 'PostgreSQL Console' on the left and '+1 FILL DB' and 'QUERY DB' on the right. Below the header, there are three main form panels. The first panel, titled 'Add campaign', contains fields for 'Age' (a dropdown menu with 'Ten' selected), 'Name' (a text input with 'Cat in the Hat'), 'Beginning date' (a date input with 'August 19th' and a calendar icon), 'Ending date' (a date input with 'August 19th' and a calendar icon), and two more 'Name' text inputs, each with 'Cat in the Hat'. A pink 'ADD' button is at the bottom. The second panel, titled 'Add prospect list', contains an 'Age' dropdown menu, a 'Name' text input with 'Cat in the Hat', and a pink 'ADD' button. The third panel, titled 'Add prospect', contains an 'Age' dropdown menu and three 'Name' text inputs, each with 'Cat in the Hat'. A pink 'ADD' button is at the bottom.

Figure 1 – Application en développement

- Effectuer des requêtes dans une console virtuelle
- Afficher des charts de statistique sur la base de données

Conclusion

Ce projet de mise en place de base de données nous a permis d'apprendre de nouvelles choses telles que :

- Les bases concernant les bases de données
- Le langage SQL permettant de communiquer avec la base de données
- L'écriture de script permettant de ne pas passer par l'invite de commandes
- L'utilisation du logiciel JMerise

Il nous a également permis d'aiguiser certains points tels que :

- Le travail d'équipe avec la répartition des tâches
- La conception d'un projet en fonction des besoins du client
- L'organisation des idées en fonction des demandes

Afin de voir plus en détails notre travail, veuillez trouver ci-dessous le lien de notre Github :

https://github.com/CamilleLeou/projet_BDD.git