CUSTOMCARE VACCINS : BASE DE DONNEES POUR ENTREPRISE DE VACCINS SUR MESURE

Bases de données

2024

Andreadi Angeliki, Ibrahim Laila, Simao Bernardo

Table des matières

Contexte métier : *3*
Diagramme de classe : *4*
Modèle relationnel : *6*
Script MySQL: *6*
Requêtes SQL: 8
Visualisation Tableau Desktop : *10*
Utilisation de modèles d'IA : *11*
Répartition du travail : *11*
Conclusion : *12*
Sources : *13*

Contexte métier :

Contexte
L'entreprise que nous avons décidé de créer se situe dans le secteur de la santé et de la biotechnologie, deux domaines en constante évolution et cruciaux pour le bien-être de la population. Dans ce contexte, nous avons identifié une opportunité de répondre à une demande croissante de personnalisation des traitements médicaux, en particulier dans le domaine de la vaccination.

Problématique
Notre entreprise se spécialise dans la création de vaccins personnalisés, conçus spécifiquement pour chaque individu en fonction de son génome unique. Contrairement aux vaccins standards, qui sont conçus pour être efficaces pour une population générale, nos vaccins sont adaptés aux caractéristiques génétiques et médicales de chaque patient. Cette approche permet de répondre aux besoins spécifiques des personnes présentant des maladies génétiques ou des caractéristiques biologiques qui les rendent non réceptives aux vaccins standards. En utilisant des données génomiques et médicales, notre entreprise peut concevoir des vaccins sur mesure qui offrent une meilleure efficacité et une sécurité accrue pour ces populations. Nous utiliserons des données fictives générées pour mener à bien notre mission.

Publique cible

Notre public cible comprend les personnes ayant des conditions médicales les rendant non réceptifs aux vaccins standards. Ils peuvent être souffrant de maladies, d'allergies ou de conditions (comme la grossesse) qui font en sorte qu'ils doivent être très conscients des composants des traitements médicaux prescrits.

Diagramme de classe :

Client-Humain-Employé
L'entité principale de notre diagramme est la classe "Client", donc nous avons fait en sorte d'adapter les reste des classes et attributs selon elle. Sa première connexion est avec la classe "Humain" dont elle hérite les attributs, en dehors de l'attribut *spécialisation* qui ne concerne que les employés. La classe "Employe" hérite également de la classe "Humain" Le but derrière la mise en place de la classe "Humain" est d'éviter la redondance des attributs *nom* et *prénom* dans les deux autres classes mentionnées précédemment.

Employé-Jobs
De la classe "Employe" héritent ensuite les classes "Personnel de santé", "Technicien de laboratoire", "Prestataires de soins de santé" et "Chercheur". Elles ont chacune des identifiants de "Humain", de "Employé" de l'entreprise en générale et de leur domaine de travail spécifique. Le but derrière la création de ces classes est d'avoir un aperçu de notre personnel.

Client-Commande_client
La classe “Client” est aussi connectée à la classe “Commande_client” à travers une relation un-à-plusieurs. Cette nouvelle classe nous permet de garder un historique des commandes faites. Nous avons opté pour une relation un-à-plusieurs entre ces classes car un client peut commander plusieurs vaccins selon ses besoins et une commande est envoyée à un seul client.

Client-Dossier_médical
D’un autre côté, la classe “Client” est liée, à travers une relation un-à-un, à la classe “Dossier médical”. La relation est un-à-un car un client ne possède qu’un dossier médical et un dossier médical existant correspond forcément à une seule personne à (et non plusieurs car ce n’est pas possible).
Pour le stockage du génome, nous avons décidé d’utiliser le concept de Data Warehousing : le Warehouse avec lequel nous allons coopérer contiendra les séquences entières de chaque génome (le génome de chaque maladie et chaque allergie) et un identifiant qui facilitera leur utilisation. Notre base de données, au lieu de stocker les génomes, stockera ces identifiants, ce qui représente une économie d’espace très importante.
Pour les attributs *maladies*, *allergies* et leurs dérivés, nous avons créé deux énumérations pour mettre en place la liste des maladies et allergies qui impactent le plus sur la réception du vaccin, que le client cochera par la suite selon sa condition.

Client-Vaccin
La classe “Client” possède une dernière liaison avec la classe “Vaccin”, qui est une relation un-à-plusieurs. Nous avons opté pour ce type de relation car un client reçoit un ou plusieurs vaccins et un vaccin est fait pour un seul client. L’attribut dosage nous permet de connaître la quantité de la dose administrée au patient en question, puisque chaque patient reçoit une certaine dose selon sa condition. Nous avons aussi mis un attribut prix qui dépend de la condition du client.

Vaccin-Evaluation_pre_vaccin-Evaluation_post_vaccin
Pour tester l'efficacité du vaccin, nous avons créé deux classes-associations reliées à la liaison entre les classes “Client” et “Vaccin”. Tant que le vaccin n’a pas encore été conçu, ces classes ne sont pas utilisées. La première est la classe-association “Evaluation_pre_vaccin”. Le but de cette classe est de procéder à une première évaluation du vaccin pour savoir si le patient est éligible (s’il ne réagit pas dû à sa condition) ou non pour prendre le vaccin. Nous lui donnons à ce moment une dose plus petite que celle administrée s’il est éligible. La deuxième classe-association est la classe “Evaluation_pre_vaccin”. Celle-ci ne concerne que les patients éligibles pour prendre le vaccin. Dans cette dernière, nous traquons le nombre d’anticorps et les effets secondaires pour pouvoir améliorer la production par la suite. Pour les deux classes, l’attribut “date_eval” et “date_evaluation” nous permettent de rester à jour au niveau des évaluations que nous avons faites.

Vaccin-Contrôle technique
Puisque notre organisation fait partie de l’industrie de la santé, il est obligatoire que des contrôles techniques soient faits. Pour cela, nous avons créé une classe “Contrôle technique” reliée à la classe “Vaccin” par une liaison un-à-un (un vaccin est contrôlé par un contrôle

technique et un contrôle n'est fait que pour un vaccin). Pour le stockage du rapport fait, nous allons procéder de la même manière que nous avons fait pour stocker le génome dans la classe "Dossier médical". Le but est d'avoir accès au rapport mais sans devoir le stocker car les agents responsables du contrôle (identifiés par l'attribut nom_responsable) ne proviennent d'agences externes.

Vaccin-Composant
Le reste des connexions est fait à partir de la classe "Vaccin" puis de la classe "Composant". Ces deux classes sont connectées grâce à une liaison un-à-plusieurs, car un vaccin contient un ou plusieurs composants mais un composant ne se trouve que dans un vaccin, puisqu'ils sont personnalisés et très spécifiques. La classe "Composant" permet le stockage des formules des composants chimiques présents dans le laboratoire qui seront par la suite utilisés dans la production.

Composant-Fournisseur
La classe "Composant" fait ensuite une liaison un-à-plusieurs avec la classe "Fournisseur", desquelles la classe-association "Commande_composant" dépend. La classe fait référence à l'entreprise qui nous fournira les composants chimiques nécessaires pour la production, car notre entreprise n'est pas spécialisée dans la production des composants mêmes. Dans cette classe, nous stockons le nom et l'adresse de l'entreprise pour rester à jour dans nos commandes. La classe-association, quant à elle, permet de stocker les quantités commandées de chaque composant dans une commande particulière. Tant qu'il n'y a pas de commandes, aucune quantité n'est stockée.

Composant-Stock
De l'autre côté, la classe "Composant" est liée à la classe "Stock" grâce à une liaison un-à-plusieurs, car un composant peut avoir un ou plusieurs stocks, mais un stock ne correspond qu'à un seul composant. L'attribut stock permet de stocker la quantité actuellement disponible de l'élément.

Stock-Bâtiment
La classe "Stock" est ensuite liée à la classe "Bâtiment" par une liaison un-à-plusieurs. Nous avons opté pour ce type de liaison car le stock d'un composant peut être réparti sur un ou plusieurs bâtiments et un bâtiment peut contenir un ou plusieurs stocks différents. Nous avons l'attribut adresse pour avoir à disposition la localisation de nos bâtiments.

Bâtiment-Machine
La dernière liaison de notre modèle conceptuel est une liaison un-à-plusieurs entre les classes "Bâtiment" et "Machine". Le rôle de la classe "Machine" est d'avoir un aperçu des machines que possède notre entreprise. La liaison est un-à-plusieurs car un bâtiment peut contenir une ou plusieurs machines mais une machine se trouve dans un seul bâtiment.

<u>Modèle relationnel :</u>

Notre modèle relationnel rejoint la structure du modèle conceptuel avec le simple ajout de clés étrangères à:

- Client: id_humain
- Employé: id_humain
- Technicien de laboratoire: id_employe
- Personnel de santé: id_employe
- Prestataires de soins de santé: id_employe
- Chercheur: id_employe
- Commande_client: id_client
- Dossier_medical: id_client, id_genome
- Vaccin: id_client
- Evaluation pre_vaccin: id_vaccin, id_client
- Evaluation post_vaccin: id_vaccin, id_client
- Composant: id_vaccin
- Commande_composant: id_composant, id_fournisseur
- Stock: id_composant, id_batiment
- Machine: id_batiment

Script MySQL:

Pour ce projet, nous avons créé une base de données nommée “fc_international” dans MySQL en utilisant une série de scripts SQL. Voici les étapes détaillées de ce processus :

1. **Suppression et création de la base de données :**

Nous avons commencé par supprimer toute base de données existante portant le même nom afin d'éviter les conflits, puis nous avons créé une nouvelle base de données “fc_international” avec le jeu de caractères `utf8` pour garantir une compatibilité internationale.

2. **Création des classes principales :**

- Classe “Humain”

Cette classe contient des informations de base sur les individus, avec l’identifiant unique “id_humain”, un nom et un prénom.

- Classe “Employe”

Cette classe dérive de “Humain” et inclut des informations spécifiques aux employés, telles que leur spécialisation. Chaque employé est identifié par un “id_employe”.

- Classe “Client”

Similaire à `humain`, cette classe stocke des informations sur les clients, y compris leur adresse, et utilise un identifiant unique “id_client”.

3. **Création des classes spécialisées pour les employés :**

Nous avons créé plusieurs classes pour différents types d'employés, héritant de la classe “Employe” :

- “personnel_sante” : Contient les employés travaillant dans le domaine de la santé.
- “technicien_labo”: Inclut les techniciens travaillant dans les laboratoires.

- “prestataire_soin_sante”: Regroupe les prestataires de soins de santé.
- “chercheur”: Enregistre les chercheurs avec leurs spécialisations.

4. **Création des classes supplémentaires :**

- Classe “batiment”

Cette classe enregistre les informations relatives aux bâtiments, avec des champs pour l'identifiant et l'adresse.

- Classe “commande_client”

Cette classe enregistre les commandes passées par les clients, en incluant des informations telles que l'identifiant de la commande, l'identifiant du client, l'adresse du client et la date de la commande. Une contrainte de clé étrangère relie chaque commande à un client spécifique.

- Classe “vaccin”

Cette classe contient des informations sur les vaccins, avec des champs pour le nom, le prix, le type, le dosage et une référence au client associé.

5. **Création des classe d'évaluation et de contrôle :**

- Classes “eval_pre_vaccin” et “eval_post_vaccin”

Ces classes enregistrent les évaluations pré- et post-vaccination, incluant des informations sur les résultats des évaluations et les effets secondaires.

- Classe “controle_technique”

Cette classe enregistre les contrôles techniques effectués sur les vaccins, avec des champs pour le nom du responsable et le rapport de contrôle.

6. **Création des classes de gestion des composants et du stock :**

- Classe “composant”

Contient des informations sur les composants des vaccins, incluant une référence aux vaccins associés.

- Classe “stock”

Enregistre les niveaux de stock des composants dans différents bâtiments.

7. **Création des classes pour les fournisseurs et les commandes de composants :**

- Classe “fournisseur”

Enregistre les informations des fournisseurs, avec des champs pour l'identifiant, le nom et l'adresse.

- Classe “commande_composant”

Cette classe enregistre les commandes de composants passées aux fournisseurs, incluant les quantités commandées.

8. **Création de la classe pour les machines :**

- Classe “machine”

Enregistre les machines présentes dans les bâtiments, incluant une référence aux bâtiments associés.

9. **Création de la classe des dossiers médicaux :**

- Classe “dossier_medical”

Cette classe enregistre des informations détaillées sur les dossiers médicaux des clients, y compris les maladies, les allergies et les antécédents familiaux.

Nous avons ensuite généré des données en utilisant des algorithmes d'intelligence artificielle, plus précisément des modèles de langages. Ces modèles sont entraînés sur de vastes ensembles de données pour apprendre à générer du texte qui ressemble à celui qu'ils ont appris. Dans ce cas, nous avons utilisé un modèle pour générer des requêtes SQL correspondant à la structure de nos classes et à des données factices pour chaque classe.

Les requêtes d'insertion de données suivent la structure suivante, que nous avons utilisé pour chacune de nos tables:

INSERT INTO table_name (col1, col2, ...) VALUES (val1, val2, ...)

INSERT INTO table_name: Cette partie de la requête spécifie dans quelle table les données doivent être insérées.

(col1, col2, ...): Ici, nous spécifions les colonnes dans lesquelles les données doivent être insérées.

VALUES: Cela indique que les valeurs des colonnes vont suivre.

(val1, val2, ...): Ici, nous fournissons les valeurs réelles qui seront insérées dans les colonnes spécifiées. Chaque tuple de valeurs correspond à une ligne de données dans la table.

Requêtes SQL:

Dans cette section, nous allons aborder l'utilisation de requêtes SQL permettant d’extraire de l’information de notre base de données et leurs résultats. Chaque explication fournira un aperçu clair des critères de sélection utilisés et des données récupérées pour répondre à des besoins spécifiques en matière de gestion des soins de santé, de production de vaccins et de planification des ressources.

1. **Sélection des clients éligibles pour un type spécifique de vaccin**

Cette requête sélectionne les clients éligibles pour un vaccin spécifique: elle récupère l’identifiant du client, son nom, son prénom et indique s’ils sont ou non à ce vaccin. Dans la commande “WHERE d.maladies = “Immunodéficience combinée sévère”, le critère de sélection est la présence de la maladie “Immunodéficience combinée sévère” dans le dossier du patient.

Cette information est cruciale pour cibler les personnes à vacciner en priorité contre des maladies spécifiques.

2. **Sélection des commandes de composants pour la production de vaccins**

Cette requête récupère les identifiants des composants et les quantités commandées, ou la quantité commandée est supérieure ou égale à 200 unités. Le critère de sélection est la quantité minimale requise pour la production de vaccins. Le rendu de cette requête est importante pour multiples raisons:

- Optimisation des ressources: La vérification des stocks permet de planifier la production de manière proactive, en s'assurant que tous les composants nécessaires sont disponibles. Cela minimise les temps d'arrêt et optimise l'utilisation des ressources tout en prévenant les pénuries.
- Réduction des coûts: Une bonne gestion des stocks peut réduire les coûts liés au stockage excessif ou à l'achat urgent de composants à des prix plus élevés.

3. **Sélection des employés avec leur spécialisation:**

Cette requête extrait les identifiants, nom, prénoms et spécialisations de tous les employés de l'entreprise. Même si c'est une requête élémentaire, la liste complète de personnes qu'elle fournit pourrait avoir des utilisations diverses et multiples comme attribuer les tâches et responsabilités, gérer les salaires, etc.

4. **Sélection des rapport des contrôles techniques des vaccins:**

Cette requête récupère les identifiants des contrôles techniques, les noms des responsables et les rapports associés.

5. **Nombre de vaccins administrés par mois**

Cette requête compte le nombre de commandes de vaccins effectuées chaque mois. Elle fournit une vue d'ensemble de l'activité de vaccination au fil du temps, ce qui est pertinent pour suivre la progression de la vaccination et planifier les ressources nécessaires. Plus précisément elle sert à:

- Suivre la demande : Cette métrique permet de suivre la demande pour les vaccins au fil du temps, d'identifier les tendances saisonnières ou les pics de demande. Cela aide à planifier la production et à gérer les stocks de manière plus efficace.
- Évaluer de la performance : Analyser le nombre de vaccins administrés chaque mois permet d'évaluer la performance de l'entreprise et de ses campagnes de vaccination, ajustant les stratégies en fonction des résultats obtenus.
- Prévoir des besoins futurs : Les données historiques sur les vaccinations peuvent être utilisées pour prévoir les besoins futurs, optimisant ainsi la planification des ressources et des approvisionnements.

6. **Regroupement des clients par maladies/allergies communes**

Cette requête compte le nombre total de clients pour chaque combinaison de la maladie et d'allergie répertoriée dans la base de données. Elle fournit des informations sur la prévalence des maladies et des allergies parmi la clientèle, ce qui peut aider à adapter les programmes de vaccination et les soins de santé en fonction des besoins spécifiques des patients.

- Optimisation des processus R&D : Ces regroupements fournissent des informations précieuses pour la recherche et le développement, permettant de cibler les efforts sur les besoins les plus fréquents ou les plus critiques.
- Communication ciblée : Les campagnes de communication et d'éducation peuvent être mieux ciblées en fonction des besoins spécifiques de chaque groupe, améliorant ainsi l'engagement et la satisfaction des clients.

Visualisation Tableau Desktop :

Pour ces visualisations, nous avons décidé d'analyser plusieurs domaines différents. Dans un premier temps nous avons visualiser la récurrence d'effets secondaires que nos vaccins ont procuré à certains de nos clients. Ensuite nous avons observé qu'elles étaient les maladies et les allergies qui sont les plus fréquentes chez nos clients. Pour conclure nous avons analysé le prix par type de vaccin, vu que chaque vaccin est unique c'est intéressant de voir la différence de prix entre les vaccins.

Pour cette première visualisation (fig. 1), nous avons fait un diagramme circulaire afin de voir la récurrence d'effets secondaires causés par nos vaccins.
Sur ce diagramme nous pouvons voir qu' un tiers des clients n'a rien subi ce qui est très positif pour l'avenir de notre entreprise, cependant nous avons 10 clients ont subi quelques symptômes, ils sont assez variés. Ces informations sont très utiles surtout pour nos chercheurs, afin d'améliorer les facultés de nos vaccins.

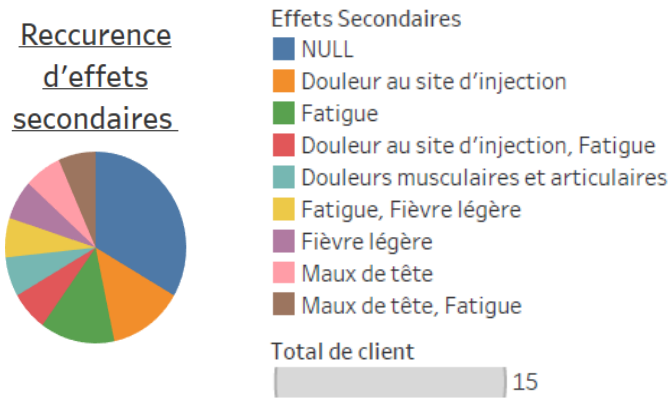

Figure 1: Récurrence d'effets secondaires

Pour cette deuxième (fig. 2), afin d'étudier le nombre de clients par allergie et maladie. Nous avons fait des diagrammes verticaux pour avoir la possibilité de voir les maladies et allergies qui sont les plus fréquentes.
Nous pouvons voir que les maladies et allergies sont assez variées, mais nous sommes en mesure de voir que certains clients ont des maladies et allergies en commun.
Ces visualisations sont à nouveau très pertinentes car nous devions anticiper lors des commandes de composants ces diagrammes auraient un rôle crucial afin d'acheter les composants qui seraient les plus utilisés.

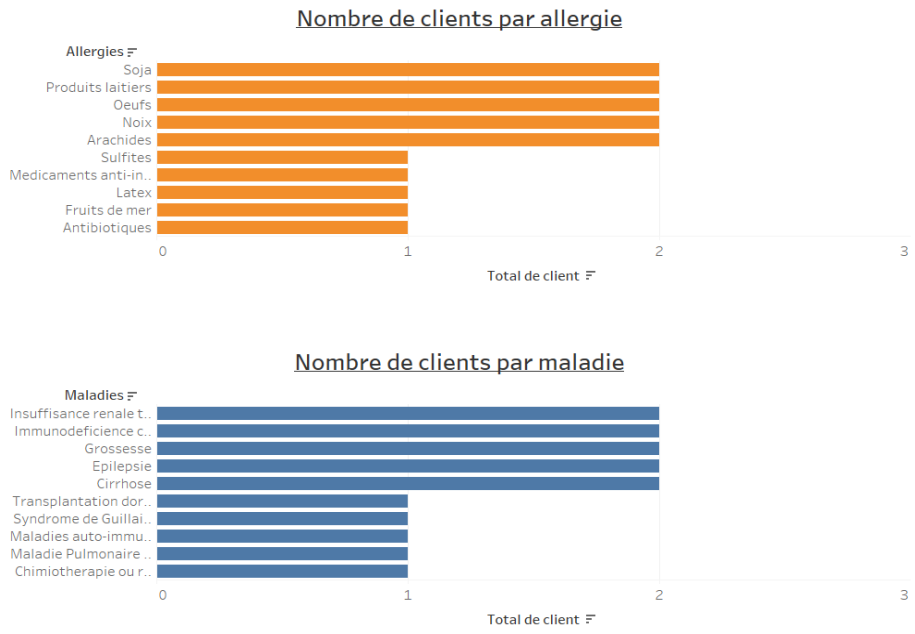

Figure 2: Nombre de clients par allergie

Enfin, pour la dernière visualisation (fig. 3) pour comparer le prix avec le type de vaccin. Nous avons conceptualisé un diagramme vertical. Cette visualisation nous montre combien coûte chaque vaccin. Nous pouvons voir que chaque vaccin a un prix différent, ce qui est totalement normal, car chaque vaccin est unique. Chaque vaccin est personnalisé pour un seul client, chaque client a des besoins différents. C'est possible que plusieurs clients aient la même maladie, cependant ils ont des génomes, âge, sexe différent, donc même si ils ont le même besoin médical le prix peut varier selon la personne. Ce diagramme démontre bien cela.

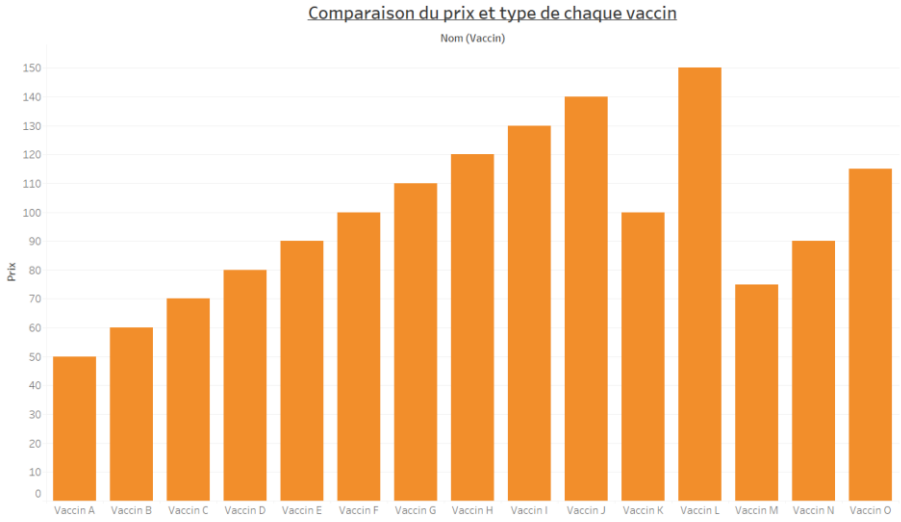

Figure 3:Comparaison du prix et type de chaque vaccin

Utilisation de modèles d'IA :

Les algorithmes utilisés durant notre projet ont été ChatGPT et Copilote pour les fonctions suivantes, avec leurs points pour et contre :

Utilisation	**Pour**	**Contre**
Source d'inspiration pour les classes initiales	Offre de la perspective sur un sujet qu'on connait pas (la production de vaccins) rapidement et en condensé	Nécessite du fact-check
Débogage de code SQL	Rapide, explique clairement	N'offre pas toujours la solution la plus intelligente ou la plus efficace
Création de données fictives	Gain de temps	Comprends rarement nos besoins du premier coup, nécessite beaucoup de "fine-tuning"
Reformulation de texte	Corrige orthographe, améliore syntaxe et grammaire	Phrases robotiques et "trop parfaites" qui elles aussi nécessitent de la réécriture

Répartition du travail :

Personne	Travail fait
Laila Ibrahim	- Dictionnaire de données - Modèle conceptuel ½ - SQL ½ - Rédaction rapport ¼
Angeliki Andreadi	- Modèle relationnel - SQL ½ - Rédaction rapport ½
Bernardo Simao	- Modèle conceptuel ½ - Requêtes SQL - Visualisations Tableau

Conclusion :

La création d'une base de données complète a été un processus très itératif et parfois frustrant. Au fur et à mesure de notre progression dans le cours, de notre acquisition d'expérience et de notre découverte de nouveaux cas d'utilisation, nous découvrions dans notre projet de nouveaux problèmes à résoudre et des nouvelles améliorations à apporter. Chaque étape nous a permis de peaufiner notre compréhension et d'affiner nos compétences en gestion et manipulation de données.

Un aspect particulièrement laborieux a été la nécessité de mettre à jour tous nos documents à chaque modification d'un modèle ou d'un morceau de code. Cette tâche répétitive nous a appris l'importance de la documentation et de la gestion de version, soulignant l'importance de maintenir une cohérence et une exactitude dans nos fichiers de projet. Ces efforts constants pour maintenir une documentation précise nous ont préparés à gérer des projets de plus grande envergure à l'avenir, où la clarté et la précision sont cruciales pour le succès.

Ce projet a également été très enrichissant d'un point de vue pédagogique. En tant qu'étudiants en Science des Données, il nous a offert des bases solides pour l'avenir. Nous avons non seulement acquis des compétences techniques essentielles, telles que la modélisation de données et la rédaction de scripts SQL, mais aussi une meilleure compréhension des défis pratiques liés au développement de bases de données.

L'expérience nous a également sensibilisés à l'importance de la flexibilité et de l'adaptabilité dans le développement de projets technologiques. À mesure que nous avancions, nous avons souvent dû ajuster nos plans et stratégies pour tenir compte de nouvelles informations ou de

changements dans les exigences du projet. Cette capacité à s'adapter rapidement est une compétence précieuse qui nous servira bien dans nos futures carrières.

En conclusion, ce projet a été un parcours riche en apprentissages, consolidant nos connaissances théoriques par une application pratique rigoureuse. Il a solidifié notre confiance en tant que futurs professionnels du domaine de la Science des Données et nous a préparés à affronter les défis professionnels avec compétence et assurance.

Sources :

Contexte métier:
Baset, Selena. “Introduction”. Faculté des Sciences Economiques, Unine, 2024, PowerPoint

Modèle conceptuel:
Baset, Selena. “Modélisation II: Niveau Conceptuel”. Faculté des Sciences Economiques, Unine, 2024, PowerPoint
Rodina, Dusan . “UML Enumeration (in UML Class Diagram”, Software Ideas Modeler, 2024
https://www.softwareideas.net/class-diagram-enum
“The ENUM Type”, MySQL Documentation, 2024
https://dev.mysql.com/doc/refman/8.0/en/enum.html#enum-using
Modéliser une énumération
https://www.pm-consultant.fr/blog_des_ti/uml/faq_ea/modeliser_une_enumeration.html

Modèle relationnel:
Baset, Selena. “Modélisation III: Niveau Relationnel”. Faculté des Sciences Economiques, Unine, 2024, PowerPoint

SQL:
Baset, Selena. “SQL (I). Faculté des Sciences Economiques, Unine, 2024, PowerPoint
Baset, Selena. “SQL (II)”. Faculté des Sciences Economiques, Unine, 2024, PowerPoint
Baset, Selena. “SQL (III)”. Faculté des Sciences Economiques, Unine, 2024, PowerPoint

Visualisations:
Macko, Vladimir. “Tableau 1”. Faculté des Sciences Economiques, Unine, 2023, PowerPoint
Macko, Vladimir. “Tableau 2”. Faculté des Sciences Economiques, Unine, 2023, PowerPoint