BTS SIO – Option SLAM
Documentation d'épreuve

Mission 4 : Modélisation d'une base de données

Par Baptiste Grimaldi

Ce document est fourni en complément de la fiche E5 correspondante.

E4-Mission4-Modelisation-BDD-Baptiste-Grimaldi

Liens utiles:

Article Modélisation BDD

Check out the portfolio of Baptiste Grimaldi, a full-stack web developer specializing in creating dynamic and responsive websites. Explore his projects and contact him for your next web development project.

https://portfolio.baptistegrimaldi.info/projects/view/modelisation-de-base-de-donnee

Article de mon portfolio

SOMMAIRE

1. Introduction et présentation du projet

Le projet consiste à modéliser une base de données pour un site de recherche immobilière. L'objectif est de permettre aux utilisateurs du site de rechercher des biens immobiliers en fonction de critères précis tels que l'emplacement, le prix, la superficie, etc. La base de données doit être capable de stocker les informations relatives aux biens immobiliers ainsi que les informations associées aux utilisateurs (par exemple, leurs préférences de recherche).

La modélisation doit être réalisée de manière à garantir des performances optimales et une évolutivité facile de la solution.

> 👀 Certaines données sont confidentielles, les schéma ne seront donc par toujours entiers. Je vais tenter de rester le plus clair possible.

2. Analyse des besoins et expression fonctionnelle du besoin

- Immozia étant une image encore neuve, il est difficile de se représenter l’avenir et le future du projet. Dans une logique d’expansion et d’évolutivité massive potentielle du projet, immozia à besoin d’une base de donnée capable d’évoluer.
- L’entreprise annonce connaître l’environnement **SQL** et être familier avec les **bases de données relationnelles.**
- Il nous est précisé que la base de donnée devra sera **accédé par des millier d’utilisateur**s simultanées et doit être capable d’être **mise à l’échelle** et encaisser la charge.
- L’entreprise ne souhaite **pas payer de service externe ou de licences** pour sont début.
- Dans cette même optique, une préférence est à l’**open-source**.

Avec ces annonce de l’entreprise, un choix se pose:

- Le No-SQL
- La base de données relationnelle type MySQL

2.1. Choix de la base de données

Base de données No-SQL

avantages	inconvénients
Scalabilité horizontale facile	Pas de jointure
Permet une évolutivité facile	Pas de transaction
Facilité de lecture et d'écriture	Modèle de données moins structuré
Plus de flexibilité pour les données	Faible adoption sur le marché
Bonne performance pour les requêtes simples	Pas adapté aux requêtes complexes
	Pas adapté aux données structurées

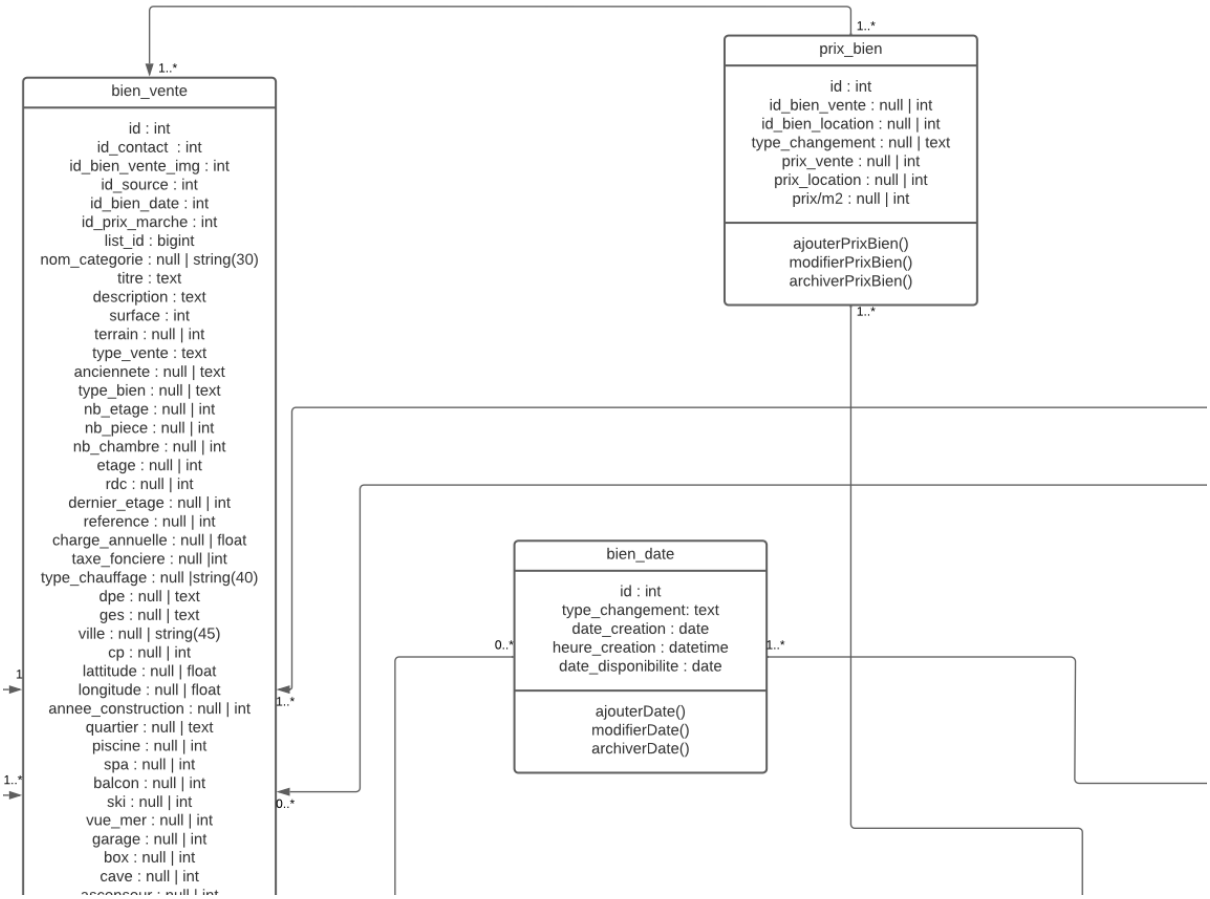
Base de données relationnelle type MySQL

avantages	inconvénients
Adapté aux données structurées	Difficilement évolutif
Adoption largement répandue sur le marché	Moins performant pour les requêtes simples
Jointures possibles entre les tables	Besoin de connaissances spécifiques pour la requête
Modèle de données structuré	Moins de flexibilité pour les données
Adapté aux requêtes complexes	Scalabilité verticale limitée

Après l'analyse des avantages et inconvénients de chaque type de base de données, nous avons choisi la base de données relationnelle type MySQL, plus précisément MariaDB, car l'entreprise a déjà de l'expérience avec cette technologie. De plus, bien que la scalabilité horizontale ne soit pas aussi facile qu'avec une base de données No-SQL, elle reste possible avec MariaDB. Enfin, MariaDB est open-source et ne nécessite pas de paiement de licence, ce qui répond aux exigences de l'entreprise.

💡 Comme outil l’interaction avec la base de donnée, deux outils sont principalement configurés:

1. PhpMyAdmin
2. les gestionnaires de BDD intégrés aux outils JetBrain.



Une partie du diagramme de classe pour la BDD.

3. Contraintes et gestion des droits d'accès

Budget

Le budget est dynamique et compte évoluer en fonction du besoin et de la dynamique d’ImmoZia. Pour commencer, la seule vraie ressource investie, est la ressource humaine.

En effet, le simple que je travail sur ce projet est une ressource en soit et je rentre donc dans le budget. Sur le projet ImmoZia, je ne suis pas tout seul, un collègue développeur m’accompagne pour cette aventure.

Contrainte technique

Quand on vend un bien immobilier, on veux que notre bien soit présenté à son avantage. L’on veux donc préciser le plus d’informations possibles.

> ❓ Cela rajoute une contrainte technique: Comment réduire la quantité de donnés stockées mais garder la pertinence de la multiple information?

La réponse est la reconstruction. Certaines informations n’ont pas besoins d’être stockés en entières où voir n’ont pas besoins d’être stockées tout court car elle dépendent d’autres valeurs.

Une astuce peux donc être d’utiliser ces valeurs utilisés connus et de reconstruire la data lors de l’affichage côté client.

Les droits d’accès

Pour répondre aux besoins d'Immozia en matière de sécurité, nous avons mis en place des restrictions d'accès aux utilisateurs de la base de données. Seuls les utilisateurs autorisés ont accès aux bases de données et aux informations stockées dans celles-ci. De plus, chaque utilisateur a un niveau d'autorisation spécifique déterminé par l'administrateur de la base de données. Les niveaux d'autorisation peuvent être ajustés en fonction des besoins de l'entreprise et du niveau d'accès requis pour chaque utilisateur.

En outre, nous avons implémenté des mesures de sécurité pour protéger les données stockées dans la base de données contre les attaques malveillantes. Cela comprend l'utilisation de mots de passe forts, le cryptage des données sensibles et la surveillance régulière de l'activité de la base de données pour détecter toute activité suspecte.

Enfin, nous avons mis en place des procédures de sauvegarde régulières pour assurer la récupération des données en cas de panne du système ou de perte de données. Ces sauvegardes sont stockées dans un emplacement sécurisé et peuvent être récupérées en cas de besoin.

Ainsi, la communication avec la base de données est limitée aux utilisateurs autorisés et les informations stockées sont protégées contre les menaces potentielles grâce à des mesures de sécurité et des procédures de sauvegarde régulières.

4. Description des environnements techniques nécessaires

Docker

Docker est une plate-forme de conteneurisation qui permet de créer, de déployer et de gérer des applications dans des conteneurs. Les conteneurs sont des environnements d'exécution isolés qui contiennent toutes les dépendances et les bibliothèques nécessaires à l'exécution de l'application. Cela permet de garantir la portabilité de l'application, car elle peut être exécutée sur n'importe quel système qui prend en charge Docker.

Nous utiliserons Docker pour créer un environnement de développement isolé pour la base de données. Cela nous permettra de travailler sur la base de données localement sans avoir à installer MariaDB ou PhpMyAdmin directement sur notre machine.

MariaDB

MariaDB est une base de données relationnelle open-source qui est largement utilisée dans le développement web. C'est une alternative à MySQL qui offre des performances, une sécurité et une stabilité améliorées. Nous avons choisi MariaDB car elle est open-source, gratuite et largement utilisée dans l'entreprise.

PhpMyAdmin

PhpMyAdmin est un outil de gestion de base de données open-source qui permet de gérer des bases de données MySQL et MariaDB via une interface web conviviale. Nous utiliserons PhpMyAdmin pour administrer notre base de données MariaDB. Cela nous permettra de créer et de gérer facilement les tables, les vues, les procédures stockées, etc.

Récapitulatif

- Docker pour créer un environnement de développement isolé

- MariaDB pour la base de données
- PhpMyAdmin pour administrer la base de données

5. Méthodologie, versioning et gestion de projet

Versioning

Nous avons mis en place un système de versioning pour la base de données en utilisant des backups Docker. Nous créons un backup de la base de données à chaque étape importante du développement et utilisons ces backups pour restaurer la base de données à un état antérieur si nécessaire. Nous avons également mis en place des tests de validation pour chaque backup afin de nous assurer que la base de données est fonctionnelle.

En utilisant cette méthode de versioning, nous pouvons facilement revenir à des versions antérieures de la base de données en cas d'erreur ou de problème de performance. Nous pouvons également suivre l'évolution de la base de données au fil du temps et vérifier que les modifications apportées sont cohérentes avec les besoins de l'entreprise.

Enfin, cette méthode de versioning nous permet de travailler en toute sécurité et de minimiser les risques de perte de données ou de corruption de la base de données.

Suivit sur le long terme

L'évolutivité verticale de MariaDB peut effectivement poser un problème à long terme.

Cela est dû au fait qu'il est plus difficile de mettre à l'échelle verticalement une base de données relationnelle qu'une base de données No-SQL.

Cependant, cette limitation peut être contournée en utilisant des techniques telles que la réplication et la partitionnement de la base de données.

Il est également important d'optimiser les requêtes et le schéma de la base de données pour minimiser la charge sur le serveur.

En fin de compte, il est important de surveiller régulièrement les performances de la base de données et de mettre à jour l'infrastructure en conséquence pour garantir des performances optimales et une évolutivité facile de la solution.

6. Modélisation de la base de données

Fonctionnement

La table bien_reçu est un élément essentiel de notre système. Elle contient l'identifiant du bien, qui est un identifiant unique assigné par chaque site web analysé. Elle contient également un JSON contenant les données à traiter, qui peuvent être de différentes natures. Ce JSON peut contenir des informations sur le bien lui-même, telles que sa taille, son emplacement ou son état, ainsi que des informations comme sa surface, son terrain, son nombre de chambre, etc… . L'indicateur de traitement est un autre élément important de la table bien_reçu, car il indique si le bien a été traité ou non. Les dates importantes sont également stockées dans cette table, afin que nous puissions suivre les différentes étapes de traitement.

La table bien_recu est liée à une table stockant les images. Cette table contient toutes les images associées à chaque bien stocké dans la table bien_reçu. Les images sont essentielles pour les clients qui cherchent à acheter ou à louer un bien immobilier, car elles leur permettent de visualiser le bien et de se faire une idée plus précise de ce qu'il offre même si un lien redirigant directement sur le site d’ou vient le bien sera disponible à la consultation de celui-ci.

Une fois que les biens sont stockés dans la table bien_recu, ils sont ensuite traités et calculés avant d'être stockés dans les tables bien_vente et location. Ces tables sont utilisées pour stocker les informations sur les biens qui sont disponibles à la vente ou à la location. Les informations stockées dans ces tables sont utilisées pour alimenter le site immobilier et aider les clients à trouver le bien qui correspond le mieux à leurs besoins.

Les tables bien_vente et location sont également liées à d'autres tables qui contiennent des informations supplémentaires sur les biens, telles que les différents prix trouvés, les différentes date de publication, etc… . Toutes ces informations sont essentielles pour aider les clients à prendre des décisions éclairées lorsqu'ils cherchent à acheter ou à louer un bien immobilier. En fin de compte, notre système repose sur la qualité des données stockées dans ces tables et sur notre capacité à les rendre accessibles et utilisables pour nos clients.

Choix techniques

Stockage des données

La base de données du site de recherche immobilière utilise MySQL, qui est actuellement utilisée et maîtrisée dans l'entreprise. Cette base de données est capable de stocker une grande quantité d'informations, organisées de manière structurée afin de faciliter la recherche et l'analyse.

MySQL est l'un des systèmes de gestion de base de données les plus populaires au monde, offrant de nombreuses fonctionnalités avancées telles que la réplication, la sauvegarde et la récupération de données. En utilisant MySQL, l'entreprise peut bénéficier d'une grande communauté de développeurs et d'utilisateurs qui peuvent fournir un support technique et des conseils, ainsi que des mises à jour régulières pour améliorer la performance et la sécurité de la base de données.

Notifications

L'application offre une fonctionnalité très utile qui permet à l'utilisateur de recevoir des notifications en temps réel sur les modifications apportées aux biens enregistrés dans les favoris, ainsi que sur les biens qui répondent aux critères de recherche spécifiés.

Cette fonctionnalité est particulièrement avantageuse pour les personnes qui cherchent constamment de nouveaux biens à acquérir, car elle leur permet de rester informées sans avoir à consulter l'application en permanence. De plus, grâce à cette fonctionnalité, les utilisateurs peuvent facilement suivre les changements de prix, les nouvelles propriétés disponibles sur le marché, les offres spéciales et les promotions, ainsi que toute autre information pertinente liée aux biens immobiliers.

En bref, l'application offre une expérience utilisateur améliorée et une plus grande efficacité dans la recherche de biens immobiliers.

Recherche API BDD

L'API de recherche Développer est un outil puissant pour les investisseurs immobiliers, car elle permet de traiter efficacement les données et de fournir des calculs précis concernant les rendements et les coûts des biens immobiliers. Cette API est également très utile pour les particuliers qui cherchent à acheter une propriété pour y vivre.

En utilisant cette API, les utilisateurs peuvent trouver facilement la propriété idéale en suivant son historique de prix, le nombre de fois où elle a été publiée sur différents sites web, ainsi que le nombre d'agences immobilières qui l'ont proposée. Cette fonctionnalité est essentielle pour simplifier l'expérience utilisateur et pour permettre aux utilisateurs de trouver rapidement les biens qui les intéressent.

La base de données du site de recherche immobilière stocke toutes les informations nécessaires ainsi que les résultats des calculs de rendement pour permettre à l'API de recherche de fonctionner efficacement. Cela garantit que les informations sont stockées de manière organisée et facilement accessibles.

7. Mise en œuvre et déploiement de la solution

Planification

Les différentes étapes de la construction de la base de données seront organisées en fonction des dates de livraison prévues et des dépendances entre les différentes tâches. Les étapes suivantes seront planifiées :

- Analyse des besoins et expression fonctionnelle du besoin
- Choix de la base de données
- Contraintes et gestion des droits d'accès
- Description des environnements techniques nécessaires
- Méthodologie, versioning et gestion de projet
- Modélisation de la base de données
- Mise en œuvre et déploiement de la solution
- Gestion de la maintenance (corrective / évolutive)

Gantt

Le diagramme de Gantt suivant représente la planification des différentes étapes de la construction de la base de données.

Tâches	Durée
Analyse des besoins et expression fonctionnelle du besoin	1 semaine
Choix de la base de données	2 jours
Contraintes et gestion des droits d'accès	1 semaine
Description des environnements techniques nécessaires	2 jours
Méthodologie, versioning et gestion de projet	1 semaine
Modélisation de la base de données	2 semaines

Mise en œuvre et déploiement de la solution	2 semaines
Gestion de la maintenance (corrective / évolutive)	1 semaine

Ressources

Les ressources nécessaires pour la construction de la base de données sont les suivantes :

- Un serveur de déploiement (ici une docker dans un VPS de production).
- Un architecte de base de données.
- Un développeur pour intégrer la base de données.

L’architecte et le développeur seront la même personne: moi-même. L’entreprise me demande de conceptualiser et d’implémenter la future base de données.

Suivi et évaluation

Le chef de projet sera responsable du suivi et de l'évaluation de la construction de la base de données. Il assurera que les différentes étapes sont réalisées dans les délais impartis et que la qualité de la solution est conforme aux exigences. Des tests seront réalisés à chaque étape pour garantir la qualité de la solution et identifier les axes d'amélioration.

Le chef de projet sera également responsable de l'élaboration de la documentation technique et fonctionnelle pour assurer la pérennité de la solution. Cette documentation sera mise à jour tout au long du projet pour refléter les différentes étapes de la construction de la base de données et les changements apportés à la solution.

Enfin, le chef de projet sera responsable du déploiement de la solution sur les environnements de production et de la gestion de la maintenance (corrective / évolutive) de la base de données.

- Élaboration de la documentation technique et fonctionnelle pour assurer la pérennité de la solution
- Déploiement de la solution sur les environnements de production

8. Gestion de la maintenance (corrective / évolutive)

Évaluation de la qualité de la solution et identification des axes d'amélioration

Il est important de surveiller régulièrement le système pour s'assurer qu'il fonctionne correctement et répond aux besoins de l'entreprise et des utilisateurs. Cela comprend la surveillance des performances de la base de données, de l'utilisation des ressources et de la qualité des données stockées. Nous avons mis en place des outils de surveillance pour surveiller en permanence l'état de la base de données et alerter l'administrateur en cas de problème.

En outre, nous avons mis en place un processus d'identification des axes d'amélioration pour améliorer constamment la qualité de la solution. Cela comprend la collecte de commentaires et de suggestions des utilisateurs, ainsi que la surveillance des tendances du marché et de l'évolution de la technologie. En utilisant ces informations, nous pouvons apporter des améliorations à la base de données pour répondre aux besoins changeants de l'entreprise et des utilisateurs.

Procédure de correction d'un dysfonctionnement de la solution

En cas de dysfonctionnement de la solution, il est important de réagir rapidement pour minimiser les interruptions de service et les impacts sur les utilisateurs. Nous avons mis en place une procédure de correction des dysfonctionnements qui comprend les étapes suivantes :

1. Identification du problème : nous utilisons des outils de surveillance pour détecter les problèmes dès qu'ils surviennent.
2. Évaluation de l'impact : nous évaluons l'impact du dysfonctionnement sur les utilisateurs et sur l'entreprise pour déterminer la priorité de la résolution du problème.
3. Correction du problème : une fois le problème identifié et son impact évalué, nous travaillons à sa résolution en collaboration avec l'équipe de développement.
4. Vérification de la solution : une fois le problème résolu, nous effectuons des tests pour nous assurer que la solution est fonctionnelle et qu'elle répond aux besoins de l'entreprise et des utilisateurs.
5. Communication avec les utilisateurs : nous communiquons avec les utilisateurs pour les informer de la résolution du problème et des mesures prises pour prévenir les futurs dysfonctionnements.

Lors de toutes communication, les logs de maridb sont enregistrés sur un volume persistant.

Planification et mise en place des tests pour garantir la qualité des mises à jour

Il est important de tester les mises à jour pour s'assurer qu'elles ne causent pas de dysfonctionnements ou d'interruptions de service. Nous avons mis en place une procédure de test des mises à jour qui comprend les étapes suivantes :

1. Planification des tests : nous planifions les tests en fonction des besoins de l'entreprise et des utilisateurs, en nous assurant que les tests sont effectués dans un environnement de test isolé.
2. Préparation de l'environnement de test : nous préparons l'environnement de test en installant les dernières versions des logiciels et en configurant l'environnement de manière à reproduire l'environnement de production.
3. Exécution des tests : nous exécutons des tests manuels et automatisés pour vérifier la fonctionnalité de la mise à jour et son impact sur le système.
4. Vérification de la solution : une fois les tests terminés, nous vérifions que la solution est fonctionnelle et qu'elle répond aux besoins de l'entreprise et des utilisateurs.
5. Communication avec les utilisateurs : nous communiquons avec les utilisateurs pour les informer de la mise à jour et de ses avantages, ainsi que des éventuels impacts sur le système.

En utilisant cette procédure de test, nous pouvons garantir la qualité des mises à jour et minimiser les risques de dysfonctionnement ou d'interruption de service.

9. Bilan du projet et compétences acquises

Conclusion

La base de données du site de recherche immobilière est conçue pour stocker des informations les biens reçus, les biens en vente et en location, les images associées à ces biens, les contacts, les dates de publication, les prix, les sources, les favoris et les données de localisation.

Les tables principales sont interconnectées pour permettre aux utilisateurs d’ajouter les biens à leurs favoris et de les consulter facilement. Cette fonctionnalité est essentielle pour simplifier l'expérience utilisateur et pour permettre aux utilisateurs de trouver rapidement les biens qui les intéressent.

L'architecture de base de données est essentielle pour permettre au site de fonctionner de manière efficace et pour fournir aux utilisateurs des informations précises et à jour sur les biens immobiliers disponibles. La conception de la base de données assure que les informations sont stockées de manière organisée et facilement accessibles.

L'application permet également de recevoir des notifications sur les modifications apportées aux biens enregistrés dans les favoris et aux biens répondant aux critères de recherche de l'utilisateur. En outre, l'interface utilisateur est personnalisable pour répondre aux préférences de chaque utilisateur, et il y a des options pour filtrer les résultats de recherche en fonction de divers critères tels que la proximité, le prix et les commodités.

En résumé, la base de données est un élément clé du site de recherche immobilière, permettant aux utilisateurs de trouver facilement des biens immobiliers et de les ajouter à leurs favoris. L'architecture de la base de données assure que les données sont stockées de manière organisée et facilement accessibles, garantissant une expérience utilisateur fluide et efficace.

Axes d’amélioration

Prévoir des évènements imprévisibles fait parti du travail de développeur. Pour cela, il se doit d’être irréprochable sur sont approche de sont organisation.

Il aurait peux être était mieux lors du premier meeting pour l’architecture de proposer des idées et de faire un brainstorm afin d’anticiper de futurs changement car telle pratique est mieux qu’une autre ou que tels nouveaux types de biens sont apparus.

Compétences acquises

- Analyse des besoins d’une entreprise
- Conception de la base de données d’un site web
- Choix de la base de données adaptée aux besoins de l’entreprise
- Utilisation de MySQL pour le stockage de données
- Mise en place de la documentation technique et fonctionnelle

- Suivi et évaluation de la qualité de la solution
- Gestion de la maintenance (corrective / évolutive) d’une base de données
- Planification et mise en place des tests pour garantir la qualité de la solution

Avis de droits d'auteur

Informations légales sur les droits d'auteur de cette documentation

Par Baptiste Grimaldi