

Mémoire de fin d'études

Migration de bases de données

Julien Ivars

Master informatique et mobilité
UHA 4.0.5 - Promotion 2024 / 2025

Alternance réalisée chez **UNIT SOLUTIONS AG**

Tuteur professionnel
M. Cédric Martin

Tuteur pédagogique
M. Mounir ELBAZ



Remerciements

J'aimerais remercier monsieur le directeur d'Unit Solutions M. Thierry MOEBEL pour m'avoir donné l'opportunité de rejoindre l'entreprise et d'effectuer ma première année de master en alternance. Je le remercie également d'avoir pris en compte mes intérêts en me confiant un projet captivant, correspondant parfaitement aux attentes de mon année. De plus, je suis reconnaissant qu'il ait prolongé mon contrat pour l'année prochaine, me permettant ainsi de me lancer dans le monde du travail et de poursuivre mon évolution au sein de l'entreprise.

Je souhaite exprimer ma gratitude envers M. Cédric MARTIN, mon tuteur en entreprise, pour son accompagnement tout au long de l'année sur le projet. Sa transmission de connaissances techniques et ses explications sur l'architecture et le fonctionnement du projet ont été d'une grande aide pour moi.

Je remercie chaleureusement tous mes collègues chez Unit Solutions pour leur partage de connaissances, leur bonne humeur et leur soutien.

Je tiens à exprimer ma reconnaissance envers toute l'équipe pédagogique de l'UHA 4.0, notamment M. Mounir ELBAZ, M. Pierre-Alain MULLER, M. Florent BOURGEOIS, M. Daniel DA FONSECA, M. Pierre SCHULLER et Mme. Audrey, ainsi que les étudiants de l'UHA 4.0. Leur soutien, leur partage de connaissances, leur accompagnement et leurs conseils au long de l'année m'ont permis de mener à bien mon projet professionnel.

Enfin, je souhaite exprimer ma gratitude envers les relecteurs de ce rapport pour leurs précieux conseils, qui m'ont permis de mener à bien l'écriture de ce rapport.

Sommaire

| | |
|--|----|
| Introduction | 3 |
| 1. Contexte | 4 |
| 1.1. Présentation de la formation | 4 |
| 1.2. Mon parcours académique | 4 |
| 1.3. L'entreprise | 5 |
| 2. Etat de l'art | 6 |
| 2.1. PostgreSQL, un système open source | 6 |
| 2.2. Des outils pour simplifier les migrations | 8 |
| 2.3. Une étape trop coûteuse ? | 8 |
| 2.4. La solution parfaite | 10 |
| Conclusion | 11 |
| Glossaire | 12 |
| Liste des abréviations | 13 |
| Bibliographie et webographie | 14 |
| Annexes | 16 |

Introduction

Après avoir réalisé mon parcours de licence professionnelle « Développeur informatique » au sein de l'UHA et obtenu mon diplôme, j'ai souhaité approfondir mes connaissances rejoignant le cursus master proposé par l'UHA 4.0 qui fait suite à la licence.

Mon parcours de master a été réalisé au sein de l'entreprise Unit Solutions basée à Allschwil en Suisse, qui s'était déjà proposée de me suivre dans mon cursus universitaires pour les deux années précédentes. Mes contributions principales se sont orientées sur le projet InfSuite et l'environnement l'entourant. L'application pour laquelle j'ai pu apporter ma contribution a comme objectif premier de gérer le suivi et la maintenance d'état d'ouvrages d'art.

Dans ce mémoire je vous présenterais les détails du projet InfSuite et de ma contribution au projet. J'ai principalement eu pour objectif de planifier et de réaliser une migration de base de données. En effet, la base de données étant un point clef de l'application, une maintenance de cette dernière est nécessaire pour assurer une certaine pérennité de l'application. Cette étape de migration s'inscrit dans un projet de maintenir les technologies de l'application à jour et de permettre de palier à d'autres problèmes.

Dans ce document je commencerais par présenter ce qui m'a amené à rejoindre le cursus master et les compétences acquises durant ma formation, j'aborderais par la suite les enjeux, une analyse et le plan d'action de la migration, puis j'expliquerais la réalisation et les problèmes rencontrés et enfin je pourrais conclure ce document.

1. Contexte

Dans cette première section, je vais remettre en contexte le concept de formation, détailler mon parcours académique et présenter l'entreprise qui m'accueille en alternance.

1.1. Présentation de la formation

L'UHA 4.0 propose des formats de formation légèrement différents des cursus traditionnels pour favoriser une immersion professionnelle tout en poursuivant des études universitaires. La formation inclut un stage obligatoire d'une durée minimale de 6 mois en complément de la période de formation. Cette immersion professionnelle fournit aux étudiants les outils essentiels pour intégrer le monde du travail. La première année introduit le développement et fournit les outils fondamentaux pour comprendre la logique de la programmation. La deuxième année approfondit ces connaissances en abordant la programmation orientée objet avec des langages tels que Java ou C#. La troisième année de Licence ajoute une méthodologie approfondie de gestion de projet en mode agile.

Après ces trois années, les étudiants peuvent, sur admission, rejoindre le parcours master de l'UHA 4.0. Ce parcours aborde les fondements de la recherche et des thématiques variées telles que la cybersécurité et l'algorithmie. Le master, également en alternance, impose une période en entreprise plus soutenue, passant de 6 à 9 mois, tout en réalisant les projets donnés par les encadrants de l'UHA 4.0.

1.2. Mon parcours académique

Après l'obtention de mon baccalauréat scientifique, j'ai intégré l'UHA 4.0 pour y suivre trois années de formation. Ces années m'ont permis de réaliser deux stages de 6 mois dans deux entreprises et une troisième année en alternance chez Unit Solutions AG. Après l'obtention de ma Licence, j'ai décidé de poursuivre en master à l'UHA 4.0 pour compléter mes acquis et acquérir de nouvelles compétences. Cette formation en alternance privilégie le temps en entreprise, passant de 6 à 9 mois.

Durant ma première année en DU 4.0.4, j'ai eu l'opportunité d'aborder des sujets complexes d'algorithmie, d'intelligence artificielle, de fouille de données et quelques notions de sécurité informatique. Pour mettre en pratique nos acquis lors des topos nous réalisons, en groupe de 3 élèves, un fil-rouge regroupant les connaissances acquises dans un sujet concret. Durant cette première année, nous avons pu mettre en place un système de suivi et d'alerte de l'état d'une plante par analyse d'./images, relevé de données environnementales, algorithmes de prédiction, ...

1.3. L'entreprise

Pour ma seconde année de master, j'ai eu l'opportunité de poursuivre mon alternance au sein de la même entreprise qui a souhaité m'accompagner dans mon projet d'études. J'ai continué à apprendre les technologies employées, ce qui m'a permis de progresser en C# et d'utiliser le framework Angular. Travailler avec des développeurs expérimentés m'a permis d'acquérir de nombreuses connaissances en gestion de projet et en développement.. Unit Solutions est une entreprise suisse basée à Allschwil, fondée en 1986 sous le nom de CADRZ et dirigée par M. Thierry MOEBEL. Elle est initialement dédiée à la création d'un cadastre numérique pour la ville d'Allschwil. La philosophie de l'entreprise évolue pour se concentrer sur la création et la maintenance de ses propres logiciels. L'entreprise compte actuellement une vingtaine d'employés pour le développement, le support et l'administratif. Les développements de l'entreprise reposent sur quatre projets principaux : Langsam Verkehr pour la gestion des sentiers de mobilité douce en Suisse, Kuba et InfKuba pour la gestion de données territoriales, et Observo pour la gestion de petits ouvrages et de mobilier urbain. Pendant mon alternance, j'ai intégré l'équipe chargée des développements pour la suite logicielle InfSuite, ce qui m'a permis de développer mes connaissances sur le projet et les technologies utilisées.

2. Etat de l'art

2.1. PostgreSQL, un système open source

2.1.1. Présentation de PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle (SGBDR¹). Le projet est initié en 1986 par Michael Stonebraker et Andrew Yu à l'Université de Californie à Berkley.

L'une des force majeur de ce système est d'être OpenSource, ce qui signifie qu'il est développé et maintenu par la communauté en plus des développements apportés par la société mère PostgreSQL.

PostgreSQL tient sa réputation de sa fiabilité, sa robustesse et sa richesse fonctionnelle que je détaillerais juste après.

2.1.2. Les principes de base

Comme dit précédemment, PostgreSQL est un SGBDR¹. Il utilise le langage SQL² pour chercher ou manipuler les données stockées. Le système met à disposition une serie de fonctions pour permettre ces interactions, à savoir:

- Les transactions: un ensemble d'une ou de plusieurs opérations regroupées en une seule opération atomique.
- Les vues: table virtuelle qui sélectionne et affiche des données à partir d'une ou plusieurs tables réelles.
- Les contraintes d'intégrité: règles qui garantissent la validité et la cohérence des données dans une base de données.
- Les procédures stockées: programme écrit en SQL qui est stocké dans une base de données et peut être exécuté à la demande.
- Les triggers: procédure stockée qui est automatiquement exécutée en réponse à un événement spécifique sur une table.
- Les fonctions utilisateurs: procédure stockée qui renvoie une valeur et peut être utilisée dans une requête SQL comme une fonction intégrée.

PostgreSQL à également l'avantage d'être multiplateforme. Il peut ainsi fonctionner sur des environnements variés avec des systèmes d'exploitations différents, comme par exemple Windows, Linux, Mac,... L'une des forces de ce système de gestion de base de données réside dans sa capacité à gérer des volumes importants de données allant jusqu'à plusieurs Terraoctets. Cette gestion passe par différents points clefs, à savoir:

- L'indexation

¹Système de gestion de base de données relationnelle

²Structured Query Language

- Le partitionnement
- La gestion du cache
- Des notions de concurrence et d'isolation
- De la réplication et du sharding³.

2.1.3. Les avantages de PostgreSQL

PostgreSQL est un SGBDR¹ très populaire pour plusieurs raisons:

- Il est open source, ce qui signifie qu'il est gratuit et que son code source est disponible pour tous. Cela permet à la communauté de développeurs de contribuer à son amélioration et de créer des extensions pour ajouter des fonctionnalités supplémentaires.
- Il est très fiable et robuste, ce qui en fait un choix idéal pour les applications critiques et les environnements de production.
- Comme vu précédemment il est très performance, grâce à son moteur de stockage et son optimiseur de requêtes. Il est capable de gérer de gros volumes de données et de supporter des charges de travail élevées.
- Le système au complet est très flexible, grâce à son architecture modulaire et à son support des extensions. Il peut s'adapter à de nombreux types d'applications et de besoins, notamment pour des applications géographiques avec des besoins plus complets.
- De par sa nature open source, il est compatible avec de nombreux langages de programmation, tels que Python, Java, C++, Ruby, PHP, etc.

Il est également important de noter que PostgreSQL tient sa popularité, au delà de ses performances et fonctionnalités déjà complètes, de par sa capacité à gérer des types de données bien plus complexes. Il propose la gestion de modèles de données complexes tel que des données géographiques et des données attributaires, mais permet surtout de gérer les relations entre ces données.

Cette gestion de données complexe permet une ouverture sur d'autres systèmes, notamment QGIS, un système d'informations géographiques, et ainsi d'étendre les fonctionnalités proposées par ce système.

En type de fichiers volumineux, on peut par exemple citer les fichiers MAJICS, RPG, référentiels vecteurs, ...

2.1.4. Les inconvénients de PostgreSQL

PostgreSQL présente également quelques inconvénients qu'il faut prendre en compte:

- Il peut être plus complexe à installer et à configurer que d'autres SGBDR¹, tels que MySQL ou SQLite.
- Il peut nécessiter plus de ressources matérielles (mémoire, CPU, espace disque) que d'autres SGBDR¹ pour fonctionner de manière optimale.
- Il peut être moins performant que d'autres SGBDR¹ pour certaines tâches spécifiques, telles que les requêtes de type OLAP (Online Analytical Processing).

³Alié à la réplication il permet de répartir la charge sur plusieurs instances d'un même serveur.

« PostgreSQL 12 - Guide de l'administrateur » de Guillaume Lelarge et Stéphane Schildknecht, éditions Eyrolles, 2020.

« PostgreSQL - Maîtrisez les fondamentaux du SGBD open source » de Régis Montoya, éditions ENI, 2019.

« PostgreSQL - Le guide complet de l'administrateur et du développeur » de Joshua D. Drake et Peter Eisentraut, éditions Pearson, 2018.

2.1.5. Conclusion

PostgreSQL est un SGBDR¹ open source très populaire, grâce à sa fiabilité, sa robustesse, sa richesse fonctionnelle et sa flexibilité. Il est utilisé dans de nombreux domaines, tels que la finance, la santé, l'éducation, le gouvernement, etc. Il est également compatible avec de nombreux langages de programmation et de nombreux systèmes d'exploitation. Cependant, il peut être plus complexe à installer et à configurer que d'autres SGBDR¹ et nécessiter plus de ressources matérielles. Malgré ces inconvénients, PostgreSQL reste un choix idéal pour de nombreuses applications critiques et environnements complexes.

2.2. Des outils pour simplifier les migrations

2.2.1. Le but d'une telle migration

2.2.2. Les points clefs

2.2.3. Les avantages

2.2.4. Les inconvénients

2.2.5. En conclusion

2.3. Une étape trop couteuse ?

Effectuer une migration de base de données n'est pas une tâche anodine. Cela demande du travail en amont, il faut analyser les différents scénarios possibles, estimer un budget pour une telle tâche, réaliser de potentiels développements, s'assurer de la fiabilité avant de mettre tout ça en pratique et enfin la réalisation de l'étape cruciale sur les environnements sensibles. Pour faire une migration de base de données il existe de nombreuses solutions, certaines plus couteuse, d'autre plus fiables, encore d'autres des plus spécialisées, ... Il faut donc dans un premier temps trouver différents outils pour comparer les avantages et leur faiblesses.

2.3.1. Les outils

On peut dans un premier temps penser à effectuer cette migration grâce à des outils spécialisés qui se chargent de faire la migration automatiquement et de s'assurer de la pérennité entre les données de l'ancienne base et les données insérées dans la nouvelle.

En prenant pour exemple l'outil Oracle Data Dump fourni par l'entreprise Oracle, ce logiciel permet de sauvegarder et restaurer des données et des métadonnées pour les bases Oracle.

Il est rapide et fiable, on peut ainsi lui fournir un fichier dump de la base source et l'outil va se charger, grâce à ce fichier, d'intégrer les données dans la base cible.

Microsoft propose sa solution alternative SSMA pour importer les types de bases Access, DB2, MySQL, Oracle, SAP ASE vers leurs différents SGBDR¹ propriétaires (à savoir les suites SQL Server).

Sur le même principe les outils MySQL workbench, AWS Database Migration Service (DMS) et PgAdmin permettent de réaliser le même type de migration vers leurs systèmes propriétaires à savoir respectivement MySQL, AWS et PostgreSQL.

Les principaux désavantages de ce genre de solutions sont:

- L'import des données reste assez stricte et ne permet pas de flexibilité, si les données ne sont pas compatibles, il faut passer du temps à travailler le modèle de données pour essayer de palier aux incompatibilités.
- Ils peuvent également rendre les migrations onéreuses avec certaines solutions qui coûtent jusqu'à plusieurs centaines d'euros pour les entreprises.
- Les logiciels proposés peuvent parfois être complexes et demander un certain temps d'adaptation avant de réellement pouvoir effectuer la tâche de migration.

2.3.2. Les types de migrations

Il est possible, comme vu précédemment, d'effectuer une migration à **partir d'outils automatiques**. L'avantage est de déléguer la majorité de la complexité à une application qui va se charger d'effectuer la tâche cruciale, de réduire les risques d'erreurs pour de grosses applications mais peut être coûteuses, insuffisamment flexible si le besoin se présente et nécessite une expertise pour la configuration des outils.

Une alternative à prendre en compte est la migration **manuelle**. Cette solution est intéressante pour les petites bases de données sans trop de complexité. Le but est d'exporter un fichier dump de la base source et de l'importer dans la base cible si possible, ou d'exporter les données sous un autre format pour l'importer simplement. Les coûts de cette technique sont faibles voir nuls et permet une flexibilité lors de la migration.

Pour palier au problème de prix, une autre solution envisageable est l'export d'un fichier de backup importé par la suite sur la nouvelle base. De la sorte, pas de coût supplémentaire, pas de temps dépensé pour effectuer la migration en elle-même et cette méthode rentre dans le cas des bonnes pratiques de posséder un backup régulier d'une base de données pour s'assurer de ne pas perdre les données en cas de problème.

Un problème majeur de cette solution est qu'elle ne peut pas garantir que les données soient compatibles, ne permet pas de résoudre les erreurs si il y'en a, il n'y a aucune garantie de succès.

Une solution possible intermédiaire permettant de mitiger problèmes de coûts et problèmes de pérennité, serait de développer en interne de l'entreprise une application de migration de base de données. Même si les coûts ne sont pas moindres, la solution s'adapte au besoin précis à l'instant de la migration, il permet de gérer l'intégralité des actions réalisées et permet également de gérer les cas spéciaux si il y'en a.

En prenant pour exemple une ancienne version de PostgreSQL qui ne supportait pas les identifiants uniques universels (GUID) et il est maintenant question de faire une migration vers une version qui supporte ce type de données. En développant un logiciel de migration on peut par exemple

2.3.3. Les ressources disponibles

2.3.4. La philosophie de l'entreprise

2.4. La solution parfaite

Conclusion

Conclusion

Glossaire

- **Dump** : Un fichier « dump » est un fichier de sauvegarde qui contient une copie de toutes les données et métadonnées d'une base de données à un instant T.
- **Sample** : Sample.

Liste des abréviations

- **sample** : sample

Bibliographie et webographie

Table des matières

| | |
|--|----|
| Introduction | 3 |
| 1. Contexte | 4 |
| 1.1. Présentation de la formation | 4 |
| 1.2. Mon parcours académique | 4 |
| 1.3. L'entreprise | 5 |
| 2. Etat de l'art | 6 |
| 2.1. PostgreSQL, un système open source | 6 |
| 2.1.1. Présentation de PostgreSQL | 6 |
| 2.1.2. Les principes de base | 6 |
| 2.1.3. Les avantages de PostgreSQL | 7 |
| 2.1.4. Les inconvénients de PostgreSQL | 7 |
| 2.1.5. Conclusion | 8 |
| 2.2. Des outils pour simplifier les migrations | 8 |
| 2.2.1. Le but d'une telle migration | 8 |
| 2.2.2. Les points clefs | 8 |
| 2.2.3. Les avantages | 8 |
| 2.2.4. Les inconvénients | 8 |
| 2.2.5. En conclusion | 8 |
| 2.3. Une étape trop coûteuse ? | 8 |
| 2.3.1. Les outils | 8 |
| 2.3.2. Les types de migrations | 9 |
| 2.3.3. Les ressources disponibles | 10 |
| 2.3.4. La philosophie de l'entreprise | 10 |
| 2.4. La solution parfaite | 10 |
| Conclusion | 11 |
| Glossaire | 12 |
| Liste des abréviations | 13 |
| Bibliographie et webographie | 14 |
| Annexes | 16 |

Annexes

Résumé

Abstract

Mots-clés

-
-