

GESTION DE CONTRATS ENTRE SOCIETES DU MARCHE TOURISTIQUE UTILISANT LE SGBD POSTGRESQL

Taras CHIEN

*Université nationale d'économie de Kharkiv Simon Kuznets, Ukraine, Université Lumière Lyon 2, France,
e-mail: chientaras@gmail.com*

On a considéré l'utilisation des systèmes de gestion de base de données pour le fonctionnement des GDS (système informatique de réservation). On a fait l'analyse comparative de deux SGBD – NoSQL et PostgreSQL. On a choisi le PostgreSQL – un système de gestion de base de données relationnelle et objet (SGBDRO). Et on a montré comment on l'a utilisé pour le projet de la société Soqima.

Mots-clés : *base de données, PostgreSQL, GDS, systèmes de réservation informatique.*

1. Introduction

Aujourd'hui le vol en avion est devenu une vraie nécessité pour toutes les classes sociales. C'est une façon de voyages très confortable, vite et agréable. Et c'est surtout populaire parmi les hommes d'affaires qui font souvent leurs voyages de mission de travail en avion.

Actuellement on peut remarquer une forte augmentation de la quantité des vols en avion et de la quantité des services fournis par les compagnies aériennes par rapport avec des années précédentes.

Le milieu du siècle dernier a été marqué par le grand progrès du tourisme grâce à l'apparition des nouveaux canaux de communication et d'échange d'informations. À cette époque-là l'un des plus importants fournisseurs de canaux de communication SITA qui fournit, y compris le gouvernement et les organismes de défense de plusieurs pays, a donné la première impulsion à l'intégration globale des bases de données internes et des systèmes comptables. Apparaissant dans ces années, l'un après l'autre, GDS (les systèmes de réservation informatiques), en fait, se sont développés grâce aux possibilités de SITA, en choisissant de se spécialiser dans l'industrie du tourisme (Figure1).

La création de tous les GDS connus a été initiée par les fournisseurs de services touristiques. Et c'est explicable puisque le tourisme est un seul domaine qui rencontre une forte nécessité de pouvoir vendre le même produit plusieurs fois (siège dans l'avion, chambre dans l'hôtel, cabine dans le bateau). Et donc ce fort besoin d'un contrôle de l'utilisation des capacités a créé les systèmes comptables internes, dont l'accès direct a été donné par les compagnies aériennes à leurs agences. C'est l'essentiel du principe du fonctionnement de GDS [1].

Un système informatique de réservation (SIR) est un outil employé pour stocker et récupérer des opérations d'information et de comportements liés au domaine touristique. À l'origine conçu et exploité par les compagnies aériennes, CRSes ont ensuite été étendues à l'utilisation des agences de voyage. Les grandes opérations de CRS que de réserver des billets et de vendre à plusieurs compagnies aériennes sont connus comme systèmes mondiaux de distribution (GDS) [1].



Figure 1. Le principe du fonctionnement de GDS

Les plus grands GDS sont les suivants :

- Amadeus (<http://www.amadeus.net/>) ;
- Galileo (<http://www.travelport.com/lob/gds/galileo.aspx>);
- Sabre (<http://www.sabre-holdings.com/>) ;
- Worldspan (<http://www.worldspan.com/>) ;
- Abacus (<http://www.abacus.com.sg/>) ;
- KIU (http://www.kiusys.com/site_en/) (Figure 2).



Figure 2. Les plus connus GDS systèmes

1. Le choix d'un meilleur SGBD

Pour le meilleur fonctionnement des systèmes de réservation informatiques il faut absolument utiliser les SGBD (Système de gestion de base de données) qui supportent les grandes charges et aussi qui supportent l'accès multiutilisateur à la même information en même temps.

Le choix d'une société SOQIMA variera entre deux SGBD : NoSQL et PostgreSQL. Pour choisir le meilleur il faut d'abord les comparer.

NoSQL (Not only SQL en anglais) désigne une catégorie de systèmes de gestion de base de données (SGBD) qui n'est plus fondée sur l'architecture classique des bases relationnelles. L'unité logique n'y est plus la table, et les données ne sont en général pas manipulées avec SQL. Les représentants principaux de bases de données NoSQL sont : Google, Facebook, Ebay etc. [4].

Les SGBD traditionnels se sont soumis aux exigences ACID: atomiques, cohérentes, isolées, et durables, alors que dans NoSQL à la place d'ACID, on peut considérer un ensemble de propriétés BASE:

- disponibilité principale- chaque requête s'achève toujours (avec succès ou non) ;
- état souple –l'état du système peut changer avec le temps, même sans nouvelles données, pour s'assurer de l'harmonisation des données ;
- cohérence en fin compte - les données peuvent être mal aligné un certain temps , mais parvenir à un accord après un certain temps [4].

En général, les bases de données NoSQL comme MongoDB ou CouchDB concourent les unes avec les autres, mais maintenant il existe une base de données relationnelle – PostgreSQL – qui se positionne comme un acteur NoSQL (Figure 3).

PostgreSQL est un système de gestion de base de données relationnelle et objet (SGBDRO). C'est un outil libre disponible selon les termes d'une licence de type BSD[4].

Data Model	Performance	Scalability	Flexibility	Complexity	Functionality
Key-value Stores	high	high	high	none	variable (none)
Column Store	high	high	moderate	low	minimal
Document Store	high	variable (high)	high	low	variable (low)
Graph Database	variable	variable	high	high	graph theory
Relational Database	variable	variable	low	moderate	relational algebra

Figure 3. Comparaison d'opportunités de base de données

Ce système est concurrent d'autres systèmes de gestion de base de données, qu'ils soient libres (comme MariaDB, MySQL et Firebird), ou propriétaires (comme Oracle, Sybase, DB2, Informix et Microsoft SQL Server). Comme les projets libres Apache et Linux, PostgreSQL n'est pas contrôlé par une seule entreprise, mais est fondé sur une communauté mondiale de développeurs et d'entreprises [4].

Le projet PostgreSQL, qui est soutenu par EnterpriseDB, a ajouté NoSQL style de traitement – JSON des caractéristiques qui datent de 2012. Maintenant, la société encourage la poursuite des travaux autour de cet ensemble de fonctionnalités en fournissant un kit de développement pour le rendre plus facile pour les programmeurs de tirer parti des fonctions de JSON de PostgreSQL et de créer des applications autour d'eux [2].

Grâce à JSON on peut « obtenir » NoSQL dans le PostgreSQL. Le type de données json peut être utilisé pour stocker des données au format JSON (JavaScript Object Notation), dont la spécification est disponible sur RFC 4627. Ce type de données peut aussi être stocké dans une colonne de type texte mais le type de données json a l'avantage de vérifier que chaque valeur stockée est une valeur JSON valide. Il existe aussi des fonctions de support, voir Section 9.15, « Fonctions et opérateurs JSON ».

PostgreSQL permet un seul encodage serveur par base de données. Du coup, il n'est pas possible pour JSON de se conformer rigidement à la spécification sauf si l'encodage serveur est de l'UTF-8. Les tentatives pour inclure des caractères qui ne peuvent pas être représentés dans l'encodage serveur échoueront ; au contraire, des

caractères qui peuvent être représentés dans l'encodage du serveur mais pas en UTF-8 seront autorisés. Les échappements `\uXXXX` sont autorisés quelque soit l'encodage du serveur et sont seulement vérifiés syntaxiquement.

Il existe deux types de données JSON: `json` et `jsonb`. Ils acceptent les ensembles presque identiques de valeurs en entrée. La différence pratique importante est l'un d'efficacité. Le type de données JSON stocke une copie exacte de la saisie de texte, fonctions de traitement qui doivent « repasser » à chaque exécution; tandis que les données `jsonb` sont stockées dans un format binaire décomposé qui le rend légèrement plus lent à l'entrée en raison de surdébit de conversion supplémentaire, mais beaucoup plus rapide à traiter, car aucune ré-analyse est nécessaire. `jsonb` supporte également l'indexation, qui peut être un avantage significatif.

Parce que le type JSON stocke une copie exacte de la saisie de texte, il permettra de préserver l'espace blanc sémantiquement négligeable entre les jetons, ainsi que l'ordre des clés dans des objets JSON. En outre, si un objet JSON dans la valeur contient la même touche plus d'une fois, toutes les paires clé / valeur sont conservés. (Les fonctions de traitement considèrent la dernière valeur que celle de fonctionnement.) En revanche, `jsonb` ne préserve pas l'espace blanc, ne conserve pas l'ordre des clés de l'objet, et qui ne garde pas les clés de l'objet en double. Si un double des clés sont spécifiés dans l'entrée, seule la dernière valeur est maintenue.

PostgreSQL fournit un riche ensemble d'outils pour les développeurs pour gérer l'accès simultané aux données. En interne, la cohérence des données est maintenue en utilisant un modèle multiversion (Multiversion Concurrency contrôle, MVCC). Cela signifie que lors de l'interrogation d'une base de données de chaque transaction voit une image de données (une version de base de données) comme il était il ya quelque temps, indépendamment de l'état actuel des données sous-jacentes. Cela protège la transaction de données incohérentes, qui pourrait être causé par les (autres) mises à jour de transaction concurrentes sur les mêmes lignes de données, offrant une isolation de transaction pour chaque session de base de données. MVCC, en renonçant à des méthodes de verrouillage des systèmes de bases de données traditionnelles, minimise les conflits de verrouillage afin de permettre des performances raisonnables dans des environnements multi-utilisateurs.

Le principal avantage de l'utilisation du modèle MVCC de contrôle de concurrence plutôt que de verrouillage est que dans les écluses MVCC acquis pour l'interrogation (lecture) des données n'entrent pas en conflit avec les verrous acquis pour l'écriture de données, et ainsi de lire ne bloque jamais l'écriture et écrire ne bloque jamais la lecture. PostgreSQL maintient cette garantie même lors de la fourniture la plus stricte niveau d'isolation de transaction par le biais d'un plan de l'innovation Serializable Snapshot Isolation (SSI).

Céramiques installations de verrouillage de niveau ligne sont également disponibles dans PostgreSQL pour les applications qui ne sont pas généralement besoin d'isolement complet des transactions et préfèrent gérer explicitement des points particuliers du conflit. Cependant, le bon usage de MVCC fournira généralement de meilleures performances que les verrous. En outre, les verrous consultatifs définis par l'application fournissent un mécanisme pour acquérir un verrou qui n'est pas lié à une seule transaction.

The PGXDK (Postgres Extended Datatype Developer Kit) est conçu pour permettre aux développeurs d'utiliser Postgres pour les types d'applications qui, jusqu'à récemment avaient requis le seule NoSQL solution spécialisée [2].

Un ensemble des test de performance a montré PostgreSQL comme plus rapide et moins « gourmand » en espace disque que MongoDB pour le même ensemble de données JSON.

Donc, il est évident que PostgreSQL sera plus simple à utiliser puisqu'au moins il donne la possibilité de rester sur les bases des données relationnelles. Depuis l'année 2004 sa popularité extrêmement augmente (Figure 4) [2].

MongoDB est un système de gestion de base de données orientée documents, répartitionnable sur un nombre quelconque d'ordinateurs et ne nécessitant pas de schéma prédéfini des données. Il est écrit en C++ et distribué sous licence AGPL. Il fait partie de la mouvance NoSQL.

MongoDB permet de manipuler des objets structurés au format BSON (JSON binaire), sans schéma prédéterminé. En d'autres termes, des clés peuvent être ajoutées à tout moment "à la volée", sans reconfiguration de la base.

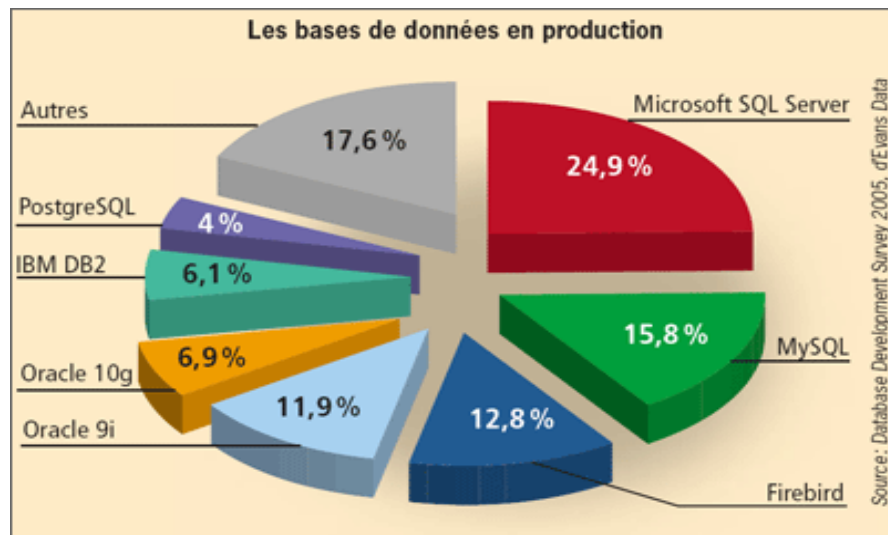


Figure 4. Comparaison des SGBD

2. Le projet PostgreSQL

3.1. L'histoire et les principales caractéristiques de PostgreSQL

Le projet POSTGRES™, mené par le professeur Michael Stonebraker, était sponsorisé par le DARPA (acronyme de *Defense Advanced Research Projects Agency*), l'ARO (acronyme de *Army Research Office*), laNSF (acronyme de *National Science Foundation*) et ESL, Inc. Le développement de POSTGRES™ a débuté en 1986. Les concepts initiaux du système ont été présentés dans Stonebraker and Rowe, 1986 et la définition du modèle de données initial apparut dans Rowe and Stonebraker, 1987. Le système de règles fût décrit dans Stonebraker, Hanson, Hong, 1987, l'architecture du gestionnaire de stockage dans Stonebraker, 1987 [2].

Depuis, plusieurs versions majeures de POSTGRES™ ont vu le jour. La première « démo » devint opérationnelle en 1987 et fut présentée en 1988 lors de la conférence ACM-SIGMOD. La version 1, décrite dans Stonebraker, Rowe, Hirohama, 1990, fut livrée à quelques utilisateurs externes en juin 1989. Suite à la critique du premier mécanisme de règles (Stonebraker et al, 1989), celui-ci fut réécrit (Stonebraker et al, ACM, 1990) pour la version 2, présentée en juin 1990. La version 3 apparut en 1991. Elle apporta le support de plusieurs gestionnaires de stockage, un exécuteur de requêtes amélioré et une réécriture du gestionnaire de règles. La plupart des versions qui suivirent, jusqu'à Postgres95™ (voir plus loin), portèrent sur la portabilité et la fiabilité [2].

PostgreSQL™ est un descendant libre du code original de Berkeley. Il supporte une grande partie du standard SQL tout en offrant de nombreuses fonctionnalités modernes :

- requêtes complexes ;
- clés étrangères ;
- triggers ;
- vues modifiables ;
- intégrité transactionnelle ;
- contrôle des versions concurrentes (MVCC, acronyme de « MultiVersion Concurrency Control »).

PostgreSQL est un système de base de données open source puissant objet-relationnel. Il a plus de 15 années de développement actif et une architecture éprouvée qui lui a valu une solide réputation de fiabilité, de l'intégrité des données, et de l'exactitude. Il fonctionne sur tous les principaux systèmes d'exploitation, y compris Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), et Windows. Il est entièrement compatible ACID, possède un support complet pour les clés étrangères, les jointures, les vues, les déclencheurs et les procédures stockées (en plusieurs langues). Il comprend la plupart des SQL types de données, y compris ENTIER, numérique, booléen, CHAR, VARCHAR date, la fréquence, et TIMESTAMP. Il prend également en charge le stockage de grands objets binaires, y compris des images, des sons ou de la vidéo. Il dispose d'interfaces de programmation natives pour C / C ++, Java, Net, Perl, Python, Ruby, Tcl, ODBC, entre autres, et de la documentation exceptionnelle [2].

PostgreSQL a beaucoup d'avantages. Par exemple il n'y a pas de coût de licence associé pour ce logiciel. Ceci présente plusieurs avantages supplémentaires :

- des modèles d'affaires plus rentables avec le déploiement à grande échelle ;
- pas de possibilité d'être audité pour la conformité des licences à tout moment ;
- flexibilité pour faire de la recherche de concept et déploiements d'essai sans avoir à inclure les frais de licence supplémentaires [2].

PostgreSQL a été conçu et créé pour avoir des exigences d'entretien et de réglage beaucoup plus bas que les grandes bases de données propriétaires, mais tout en conservant toutes les caractéristiques, la stabilité et les performances.

PostgreSQL est disponible pour presque toutes les marques d'Unix (34 plates-formes avec la dernière version stable), et la compatibilité avec Windows est disponible via le cadre Cygwin. La compatibilité native de Windows est également disponible avec la version 8.0 et au-dessus [2].

De plus, PostgreSQL peut être étendu par l'utilisateur de multiples façons, en ajoutant, par exemple :

- de nouveaux types de données ;
- de nouvelles fonctions ;
- de nouveaux opérateurs ;
- de nouvelles fonctions d'agrégat ;
- de nouvelles méthodes d'indexage ;
- de nouveaux langages de procédure [2].

3.2. L'administration de la base de données PostgreSQL

Pour l'administration des bases de données PostgreSQL propose deux types d'utilités: celle avec l'utilisation du navigateur - phpPgAdmin et l'autre plus confortable, pgAdmin III.

pgAdmin III est la plateforme d'administration et de développement libre la plus populaire et la plus riche pour PostgreSQL, le serveur de base de données libre le plus riche en fonctionnalités. L'application est utilisable sur les plateformes Linux, FreeBSD, OpenSUSE, Solaris, Mac OSX et Windows pour gérer un serveur PostgreSQL, version 7.3 et ultérieures, exécuté sur toute plateforme, ainsi que sur les versions commerciales de PostgreSQL comme Postgres Plus Advanced Server et Greenplum.

pgAdmin est conçu pour répondre aux besoins de tous les utilisateurs, de l'écriture de requêtes SQL simples aux développements de bases de données complexes. L'interface graphique supporte toutes les fonctionnalités de PostgreSQL et simplifie l'administration. L'application inclut aussi un éditeur SQL avec coloration syntaxique, un éditeur de code côté serveur, un agent de planification de job SQL/batch/shell, le support du moteur de réplication Slony-I et bien plus encore. La connexion serveur peut se faire en utilisant TCP/IP ou les sockets de domaine Unix (sur les plateformes *nix), et pourrait être chiffrée via SSL pour la sécurité. Aucun pilote supplémentaire n'est requis pour communiquer avec le serveur de bases de données.

pgAdmin est développé par une communauté d'experts de PostgreSQL répartis tout autour du monde et est disponible dans plus d'une douzaine de langues. C'est un logiciel libre proposé sous la license PostgreSQL.

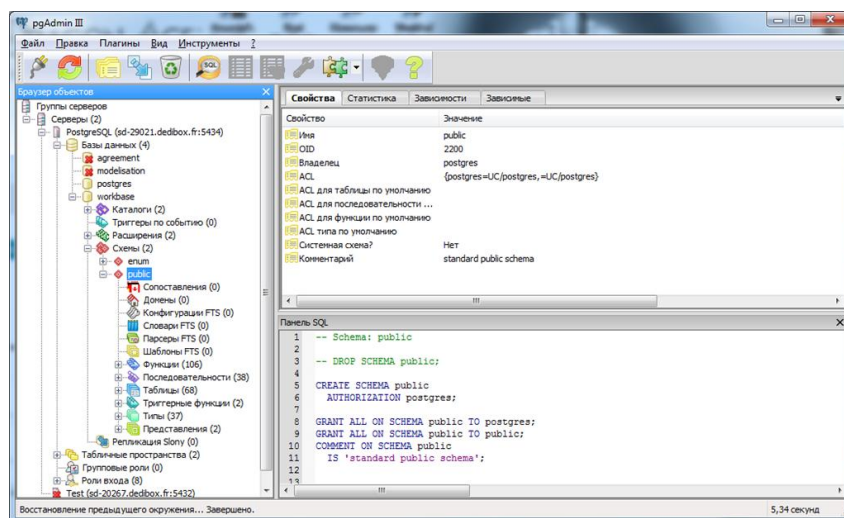


Figure 5. L'interface de pgAdmin III

1. Le projet de SOQIMA

Le projet « Gestion des stocks aériens » en cours de développement offre aux agences de voyage une application unifiée pour la saisie et la gestion des contrats de stocks de vols auprès des compagnies aérienne (Figure 6).

Tableau de bord vendeur

The screenshot displays a search interface for flight stocks. At the top, there are filters for 'Trajet' (Route: A to PAR to EV), 'Départs' (16-07-2014), 'Voyageurs' (Number: 3, Ages: 2,10), and 'Stock' (options: Disponible, Engage, A pousser). A 'Recherche' button is on the right. Below the filters is a table of flight results:

Cie	A	PAR -> EV	Pro/Dispo	Actions
AF	A: mer 16/07/2014	PAR -> EV - Economy Direct	6385 €	Reserver Detail
AF	A: mer 16/07/2014	PAR -> EV - Economy Direct	380 €	Reserver Detail
AF	A: mer 16/07/2014	PAR -> EV - Economy Direct	1290,3 €	Reserver Detail
AF	A: mer 16/07/2014	CDG -> EV - Economy Direct	380 €	Reserver Detail
AF	A: mer 16/07/2014	CDG -> EV - Economy Direct	6385 €	Reserver Detail

At the bottom, there are navigation buttons: 'Previous', '1', '2', and 'Next'.

Figure 6. Gestion des stocks aériens

Une interface permet de saisir les stocks, informations tarifaires, conditions pour les différents contrats. Une interface et des web services permettent de consommer ce stock (effectuer des réservations), et de gérer les réservations (annuler, compléter etc.).

Inspiré par les outils de gestion de projet et les places de marché en ligne, l'interface permet également de gérer des tâches à effectuer sur l'ensemble des stocks, et d'envoyer des informations aux fournisseurs comme au client.

Ce produit de l'entreprise Soqima est très grand et très important.. L'équipe du projet inclut trois personnes :

- 1) Chefduprojet
- 2) DéveloppeurAngularJS
- 3) Développeurdesbasesdedonnées

Dans ce projet la base de données est un fondement et il est essentiel de choisir un outil solide et pratique pour créer cette base et la gérer efficacement. Après une longue considération on a pris la décision d'utiliser le SGBD gratuit et pourtant puissant – PostgreSQL qui permet de répondre à de nombreuses problématiques métiers.

1. Utilisation de PostgreSQL pour le projet « Gestion des stocks aériens »

Pour une start-up avec un budget limité PostgreSQL prouvé un excellent choix, pendant les premiers stades de développement de produits quand la simplicité et la rapidité de déploiement étaient d'une importance cruciale, et au fil du temps quand les SIG fonctionnalité, la réplication et le point-intime récupération ont devenus des priorités. Ensuite PostgreSQL a du répondre à la hausse des attentes et des exigences puisque le produit de la société est devenu plus complexe.

Comme avec la plupart des entreprises en démarrage, SOQIMA a voulu développer et lancer son produit le plus rapidement possible.

PostgreSQL a une bonne documentation en ligne, offre les fonctionnalités d'une plate-forme de base de données de pointe, et suit les normes ANSI-SQL largement acceptés.

Parce que ce projet de SOQIMA affiche des informations de millions de vols aériens dans le monde, requêtes qui joignent million de tables et de lignes doit être très performantes. Tout retard dans la récupération des résultats de requête affecte directement chargement de la page – une affaire sérieuse pour un site Web axé sur le consommateur.

Il est très important d'éviter les erreurs pendant la recherche. Comme ce projet est conçu pour les entreprises (et pas pour les consommateurs - particuliers) il est essentiel qu'il soit capable d'effectuer plusieurs opérations en même temps pour le même utilisateur et qu'il soit vite et très fiable.

En considérant la quantité de données et la complexité des relations en cause, SOQIMA savait qu'il aurait besoin d'une avancée, système de base de données sophistiquée.

PostgreSQL a prouvé son efficacité à SOQIMA développeurs quand les requêtes de grandes tables qui comprenaient multiples joins complexes ont atteint ou dépassé les attentes de performance. Dans plusieurs cas, PostgreSQL dispose d'indices fonctionnels et conditionnelles révélée très précieuse pour maximiser le temps de chargement.

Dans plusieurs cas, PostgreSQL caractéristiques comme les index fonctionnels et conditionnels ont été précieuses pour maximiser le temps de chargement de page.

Afin d'assurer la disponibilité 24x7, Soqima a choisi une stratégie de basculement qui assurerait le programme de rester en vie pendant de graves problèmes du système d'exploitation. L'entreprise était capable de mettre pleinement en œuvre un standby serveur en moins de 30 heures de la planification et de l'exécution, ce qui a conduit à une amélioration dans l'architecture des systèmes et la stabilité à un prix très modeste.

PostgreSQL joue un rôle majeur dans le stockage des toutes méta-données, des associations et des relations qui font le projet de SOQIMA ce qu'il est. PostgreSQL "a permis" à SOQIMA de construire un GDS puissant et de devenir très fort dans ce domaine. Il s'agit d'une solution qui s'est également avérée évolutive.

Après des centaines de lignes de code et après plusieurs mois de développement, « Gestion des stocks aériens » a parcouru un long chemin. PostgreSQL a permis à SOQIMA d'atteindre ses objectifs de l'efficacité et de la fiabilité dans le marché concurrentiel.

PostgreSQL a inauguré une nouvelle ère de la base de données relationnelle en introduisant une plate-forme de base de données open source multi-plateforme qui bat facilement les produits établis sur la matrice des fonctionnalités-performance-coût. Il s'agit d'une technologie éprouvée et solide avec une communauté active et solidaire, grande base de développeurs, et une bibliothèque impressionnante de modules et de produits tiers pour le sauvegarder.

2. Conclusions

On a pris une bonne décision d'utiliser le système de gestion de base de données PostgreSQL et finalement pendant la création et la gestion du projet il a montré une grande efficacité et fiabilité. Cependant avant de faire ce choix il fallait faire une analyse des différents SGBD et choisir les deux meilleurs pour faire la comparaison et pour ensuite choisir entre eux. Ces deux SGBD étaient – NoSQL et PostgreSQL. PostgreSQL est la seconde base de données Open Source la plus utilisée du marché, derrière MySQL. Elle contient quelques fonctionnalités de NoSQL et en même temps elle est relationnelle donc compréhensible et familière aux spécialistes des bases des données.

PostgreSQL a beaucoup d'avantages ; comme par exemple : possibilité d'utilisation, de modification et de distribution dans un cadre privé, public et académique grâce à sa licence libérale; communauté active et importante ; le déploiement nécessite peu de compétences techniques ; la possibilité d'étendre PostgreSQL via l'écriture de fonctions (UDF, user-defined functions) ; vues matérialisées.

Du fait de sa gratuité, PostgreSQL convient aux PME-PMI souhaitant se doter d'un SGBD puissant et performant sans pour autant engager de budget.

Références

1. GDS, Global Distribution System, https://en.wikipedia.org/wiki/Computer_reservations_system
2. PostgreSQL, All about PostgreSQL. <http://www.postgresql.org/about/>
3. PostgreSQL, Advantages of PostgreSQL. <http://www.postgresql.org/about/advantages/>
4. NoSQL, All about NoSQL. <http://en.wikipedia.org/wiki/NoSQL>

Sous la supervision de (Під керівництвом):

Jean-Hugues Chauchat (PhD, HdR, professeur, ERIC lab)

Золотарьова І.О. (к.е.н., професор кафедри інформаційних систем)

Керівник з іноземної мови ст. викладач кафедри іноземних мов та перекладу Безугла І.В.