

GenialMarchéDB

Version 1 du 29 octobre 2021

Techniques avancées de gestion des données Administration de BD et XML

3^{ème} Bachelier en informatique de gestion

1. Introduction

Vous trouverez dans ce document le contexte général du système informatique qui devra être réalisé. On y détaille ensuite les différentes fonctionnalités demandées. Ce document peut donc être considéré comme votre "contrat de travail". Il permettra de mieux appréhender l'organisation du travail de l'année et sa planification au travers des différentes semaines.

Les modalités d'évaluation se trouvent à la fin de cet énoncé reprenant les explications quant à la construction de la cote finale. Il convient d'en prendre connaissance. Elles vous engagent pour toute la durée de l'année scolaire.

Les données sont purement fictionnelles.

2. Contexte

Vous êtes engagés pour le compte d'un lobby qui a pour projet de regrouper toute l'information concernant l'ensemble des supermarchés dans une même base de données, l'idée étant de pouvoir rationaliser les frais de gestion.

Pour vérifier la faisabilité du projet, on vous fait d'abord travailler sur une base de données fictive reprenant un faible échantillon de ce que la base de données contiendra par la suite.

2.1 Les ventes

La signalétique d'une vente comporte toutes les informations relatives à cette vente. Par exemple, la date, le client, la liste des articles Chaque vente est une ligne (un enregistrement) stocké dans un fichier texte appelé ventes.txt. On retrouve 11 champs par film expliqués dans le tableau page suivante

Le format d'un enregistrement de la table externe est le suivant :

Idvente ; IdClient ; NomClient ; PrénomClient ; EmailClient ; IdMagasin ; NomMagasin ; CodePostalMagasin ; [IdArticle.Description.Prix.Quantité]+ ; dataAchat ; URLTicket

Pour simplifier, aucun champ ne sera absent (il n'y a pas de valeur nulle, tous les champs sont renseignés).

Les séparateurs utilisés pour chaque enregistrement (ou ligne du fichier) texte sont les suivants :

- Le caractère ; est utilisé pour séparer les champs qui sont au niveau le plus haut de la hiérarchie (top-level).
- La liste des articles est reprise dans un champ représenté par []. Cette dénomination représente simplement le fait qu'il y a une liste de 1 ou plusieurs articles par liste. Les caractères [] **n'apparaissent pas** dans le fichier.
- Le caractère & est utilisé comme séparateur des éléments du tableau.
- Le caractère . est utilisé pour séparer les champs d'un élément dans un tableau. Il s'agit du simple caractère « point » et non le caractère Unicode appelé one dot leader dont le code est U+2024.

Pour la vente dont l'identifiant est 4, on a l'enregistrement ci-dessous dans le fichier texte ventes.txt :

4;5;Serrhini;Souad;souad.serrhini@hepl.be;3;Intermarché;3000;5.Fromage tranches.3,05.3&4.Bouteille d'eau.0,8.1&2.Pain.1,45.3;28/10/21;fichiers/2011/42/1319286512-dsc00041.jpg	en
--	----

Nom	Explications
Idvente	L'identifiant de la vente.
IdClient	L'identifiant du client ayant effectué la vente (ou plutôt, l'achat).
NomClient	Le nom de famille du client.
PrénomClient	Le prénom du client.
EmailClient	L'adresse e-mail du client.
IdMagasin	L'identifiant du magasin dans lequel la vente a eu lieu.
NomMagasin	Le nom du magasin.
CodePostalMagasin	Le code postal du magasin.
Articles	Un tableau contenant les différentes informations concernant les articles achetés lors de la vente. On y trouve le numéro d'article, le nom de l'article, son prix et la quantité achetée.
dataAchat	La date de la vente
URLTicket	Un chemin permettant de construire l'URL vers une image de type poster pour le film. Vous noterez que, par simplification, l'URL du ticket est toujours la même, mais cela ne change en rien la logique de votre programme.

3. Réalisations attendues

Vous utiliserez le fichier ventes.txt comme base pour vos données, et effectuerez les opérations suivantes pour obtenir la base de données souhaitée.

La liste peut vous paraître longue, mais certains items peuvent se réaliser avec très peu de lignes de code.

3.1 Intégration du fichier en tant que table externe

Le fichier ventes.txt sera pointé par une table externe, ce qui permettra l'accès aisée de ces données à l'intérieur d'une base de données oracle

3.2 Normalisation des tables

En suivant la méthodologie vue au cours, vous normaliserez la table externe en un ensemble de tables (internes, cette fois), en vous assurant que chaque table respecte bien la BCNF.

3.3 Gestion des blobs

Dans les tables internes, l'image pointée par l'URL dans le document externe sera récupérée sur le site <http://image.noelshack.com/> et en concaténant l'adresse du site web avec l'URL donnée dans le fichier.

Pour être très clair, chaque image provient donc en fait de l'URL
<http://image.noelshack.com/fichiers/2011/42/1319286512-dsc00041.jpg>

3.4 Fragmentation dans une base de données distribuée

Votre base de données sera en réalité distribuée en deux bases de données distinctes. Chaque base de données contiendra le même schéma, mais les données seront répliquées et/ou fragmentées.

Les informations concernant les articles et les clients seront répliquées sur chacune des deux bases de données. On considérera que cette information ne sera pas soumise à changement (pas de nouveaux clients, pas de nouveaux articles ou de retraits d'articles).

Les informations sur les ventes et les magasins seront fragmentées par rapport au code postal du magasin. La première base de données contiendra uniquement les informations concernant les ventes s'étant effectuées dans un magasin dont le code postal est compris entre 0 et 4999, alors que la deuxième base de données contiendra les informations des magasins avec un code postal entre 5000 et 9999.

3.5 Service ORDS pour visualiser et modifier les données

Votre base de données publiera un service REST à l'aide d'ORDS.

Deux modules devront être opérationnels :

- 1) Un module GET, permettant d'accéder à toutes les informations d'une vente (id, date, liste des articles, etc). Pour des questions de performances, vous chercherez l'information depuis les tables internes, et non pas depuis la table externe.

- 2) Un module POST permettant de modifier le code postal d'un magasin. Le module prendra donc, en entrée dans le body de la requête, l'ID du magasin ainsi que son nouveau code postal. Pensez que le schéma de fragmentation de votre base de données distribuée doit rester cohérent, même après utilisation de votre procédure REST. La valeur de retour sera simplement les nouvelles informations du magasin modifié (ID, Nom, code postal). Vous pouvez créer un petit module GET pour ces informations, et rediriger la réponse du module POST vers ce module si cela vous semble plus simple.

3.6 Job quotidien dans une optique de sauvegarde dans un entrepôt de données

Votre base de données contiendra un job quotidien, s'exécutant tous les jours à 3h du matin, et ajoutant les statistiques du jour dans un entrepôt de données.

Pour simplifier, on imaginera simplement ajouter une ligne dans l'entrepôt avec la date du jour et le total des ventes, tout clients et articles confondus, pour ce jour. L'entrepôt sera simplement une (autre) base de données contenant une seule table. L'update sera fait à travers un DBLink.

A des fins de testing, il vous est loisible d'ajouter des données (potentiellement aléatoires), et lors de la démonstration, de modifier l'heure du job pour qu'elle colle à l'heure actuelle.

4. Evaluation

Ce projet est à réaliser par binôme. Il sera à présenter pendant l'année, au plus tard pendant la dernière séance de labo (semaine 15). Vous pouvez toutefois présenter plus tôt si vous êtes prêts. Dans tous les cas, pour pouvoir présenter le projet, il faut l'avoir envoyé, par mail à samuel.hiard@hepl.be **au plus tard le dimanche précédent le jour du labo** (avec 23h59 comme heure limite).

Il n'y a pas de mention « au minimum », « pour les pros » et « pour les experts », mais certains modules sont indépendants alors que d'autres sont liés. On peut dès lors considérer que le minimum consiste à développer les modules indispensables, et que les pros et les experts développent les modules indépendants. Dans tous les cas, voici la répartition des points (sur un total de 15 points) :

- Table externe	2 points	
- Normalisation	3 points	(Dépendance : Table externe)
- Blobs	2 points	
- Fragmentation	3 points	(Dépendance : Normalisation)
- ORDS	3 points	(Dépendance : Fragmentation)
- Job	2 points	

Bon courage et, surtout, amusez-vous bien 😊