

TONNEAU Quentin

BARBIN Maël

Groupe 419

Projet Base de données



UNIVERSITÉ DE NANTES

Table des matières

1 Présentation – Description générale	3
1.1 Choix du sujet.....	3
1.2 Composition de notre base de donnée.....	3
1.3 Installation.....	4
1.4 Utilisation.....	4
1.4.1 Visiteur.....	4
1.4.2 Directeur de magasin et d'enseigne.....	4
1.4.3 Administrateur du site web.....	5
2 Description technique.....	5
2.1 Notes.....	5
2.2 Traitement des fichiers – Programme d'import.....	6
2.2.1 À la création.....	6
2.2.2 À l'utilisation.....	6
2.3 Principales requêtes.....	7
3 Bilan.....	7
3.1 Limites de notre travail.....	7
3.2 Conclusion.....	7
4 ANNEXE.....	9

1 Présentation – Description générale

Dans le cadre du module Base de Donnée 1, il nous a été demandé de réaliser un projet. Ce dernier a pour but la communication entre un utilisateur et une base de donnée. Cette communication s'effectue au travers d'une interface web. Nous vous présenterons rapidement notre choix de sujet, ainsi que la base de donnée associée, puis nous décrirons rapidement les fonctionnalités de notre site. Dans une seconde partie nous rentrerons dans les détails du code, avant de conclure sur l'état actuel du projet.

Les sources ainsi que les évolutions du projet sont visibles à l'adresse https://github.com/cocoto/projet_bdd_2011 , et tout signalement de bug, remarque est à effectuer sur cette plate-forme.

1.1 Choix du sujet

Nous souhaitons créer quelque chose d'utile. Nous sommes donc parti sur la réalisation d'un comparateur de prix. Très vite, nous avons du faire des choix de programmation importants. En effet, nous voulions vraiment coller à la réalité. Cependant comme il le sera rappelé plus tard dans ce rapport, le remplissage de notre base de donnée nous aurait pris trop de temps. De ce fait, les seuls choses qui appartiennent à la réalité sont les noms des produits et leur description ainsi que les types et les rayons. Tout le reste est soit généré aléatoirement, soit inspiré de la réalité.

1.2 Composition de notre base de donnée

Notre base de donnée comporte cinq tables :

- Enseigne
- Magasin
- Produit
- Tarif
- Type

Le schéma ainsi que les relations qui régissent notre base de données se trouve en annexe.

Dans la description qui suit, les clés primaires sont soulignées et les clés secondaires sont précédées du symbole #.

- Enseigne : chaque enseigne est identifiée par un identifiant unique (id_ens). Elles sont aussi caractérisées par un nom d'enseigne (nom_ens), par un nom de directeur (nom_dir) ainsi que par un mot de passe (mdp).
- Magasin : chaque magasin est identifié par un identifiant unique (id_mag). À chaque magasin, nous associons un nom (nom_m), un nom de responsable (nom_resp), une taille (taille), une ville (ville), l'identifiant de l'enseigne à laquelle le magasin est rattaché (#id_ens), ainsi qu'un mot de passe (mdp).

- Produit : chaque produit est identifié par un identifiant unique (id_p). Pour chaque produit, nous retrouvons une référence (ref), un nom (nom_p), un type (#type), et une description (description).
- Tarif : un tarif est composé de l'identifiant du produit (#id_p), de l'identifiant du magasin (#id_mag), d'un prix (prix), et de sa disponibilité (dispo).
- Type : Les deux attributs de cette table sont le type (type) et le rayon dans lequel nous pouvons trouver ce type (rayon). L'attribut type est unique.

La présence de la table type n'est pas obligatoire, à la place, nous aurions pu insérer un attribut rayon dans la table produit. Cependant, il est plus rapide dans le cas d'une modification de rayon de faire la modification sur une ligne de la base de donnée plutôt que sur l'ensemble des produits concernés.

1.3 Installation

- Régler les paramètres de la base sql dans le fichier php/fonctions.php
- Remplir ou générer les fichiers 'donnees.csv', 'enseignes.csv', 'magasins.csv' selon le modèle exposé. Voir description technique pour plus de détails
- Importer le fichier Comparateur.sql via une interface de gestion (phpMyAdmin...)
- Exécuter le fichier 'php/parser.php' dans une fenêtre internet et attendre la fin d'exécution de celle-ci. En cas d'erreur signalée, merci d'en avertir l'équipe de développement (voir indications préambule)

1.4 Utilisation

Il est important de décrire la façon dont fonctionne notre travail. Dans un premier temps, il est nécessaire de comprendre que notre interface prend en considération trois types d'utilisateurs. Nous allons donc ici décrire les actions possibles pour chacun.

1.4.1 Visiteur

Voyons dans un premier temps, le visiteur c'est-à-dire une tierce personne qui cherche à comparer les prix d'un produit dans plusieurs magasins. Cet utilisateur peut effectuer une recherche de produit par référence, type, nom, ou accéder directement à l'ensemble des produits d'un rayon ou d'une catégorie. Une liste des occurrences correspondant à sa recherche est affichée (nom du produit, description, prix minimum constaté sur le site). L'utilisateur peut alors cliquer sur l'un de ces produits pour afficher la fiche complète de ce dernier. On y trouve en l'occurrence le type, le rayon, le nom, une description de l'article, ainsi que la liste des magasins qui le possèdent en stock, leur emplacement (ville), et leur prix.

1.4.2 Directeur de magasin et d'enseigne

Les différents **responsables de magasin** peuvent se connecter depuis l'ensemble des pages

du site, par le formulaire approprié. Une fois connectés, un nouveau menu apparaît, leur permettant d'accéder à la zone de gestion de leur magasin.

Cette page affiche les produits référencés sur le site (noms et catégories), ainsi que leur prix et la disponibilité depuis la dernière mise à jour. Un responsable de magasin peut modifier uniquement les informations relatives à son commerce.

Il peut modifier les informations les unes à la suite des autres, très utile en cas de modifications mineures (changement d'un seul tarif, promotion exceptionnelle, épuisement de stock). Une seconde méthode lui permet de modifier en masse tous les tarifs et disponibilités de son magasin. Il lui suffit de cliquer sur « télécharger votre grille tarifaire », d'ouvrir le fichier téléchargé à l'aide d'un tableur (Excel, OpenOffice Calc) ou de son logiciel de gestion interne, de mettre à jour les données, et de renvoyer le nouveau fichier via le formulaire prévu à cet effet. Les modifications sont immédiates et cette implémentation est très utile pour des modifications multiples (ajouts de nouveaux produits, soldes, changement de taxe, promotion étendue, arrivage massif).

De son côté, un **directeur d'enseigne**, après connexion, accède à une liste similaire des produits, mais cette fois-ci ce dernier est invité à ajouter ou modifier les attributs des produits, comme leurs noms, leurs descriptions, leurs catégories, leurs références. Un directeur d'enseigne ne peut changer les tarifs d'un magasin, car dans la plupart des systèmes commerciaux, les magasins sont indépendants de leurs enseignes. Les tarifs communs pourront être distribués par l'enseigne sur l'ensemble de ses magasins, qui à leur tour pourront mettre à jour leurs données. Ceci évite les incohérences dans les multi-gestions commerciales.

1.4.3 Administrateur du site web

L'administrateur du site web, en se connectant au site, a la possibilité de changer les produits de la même façon que le Directeur d'enseigne. Tout comme ce dernier, il ne peut accéder ni modifier les tarifs d'un magasin. Ceci évite à l'administrateur des menus chargés, et empêche toute remarque sur l'étiqette du site web (concernant la publicité ciblée, la corruption).

L'administrateur du site est en mesure, via les formulaires de la page "admin.php", d'ajouter, modifier, supprimer toutes les enseignes, magasins, rayons, types de produit. Via un menu déroulant (ascenseur), l'administrateur sélectionne l'élément à modifier, et change les paramètres via différents éléments de formulaires (Boutons, checkbox, ascenseurs, champs de saisie). On notera que toute modification d'information relative à un magasin ou une enseigne entraîne un changement de mot de passe, que l'administrateur doit communiquer à l'unité visée. Ceci évite encore une fois toute modification de l'administrateur sans en avertir la cellule concernée.

On voit donc que derrière une utilisation basique du site (recherche, affichage), l'administration de ce dernier est au cœur de notre projet. Les droits de chacun des utilisateurs sont clairement définis. Le choix pour une Enseigne de ne pouvoir ajouter ou modifier un magasin vient du fait que ce genre de changement est rare dans la vie du site, et que l'importance de ces modifications entraîne une vérification systématique de l'administrateur.

2 Description technique

2.1 Notes

- Les mots de passe du site sont cryptés à l'aide de l'algorithme "SHA 1", et stockés dans la base comme des chaînes de caractères.
- L'utilisation de guillemets dans l'ensemble des champs est déconseillé, malgré les systèmes de protections mis en place.

- En cas d'utilisation sur une base de donnée autre (Oracle, Sql....), il suffit de modifier le fichier fonctions.php
- L'ensemble du site est valide W3C (html et css)
- Les sources sont publiés sous les mêmes conditions que le document ici-présent (CreativCommons)

2.2 Traitement des fichiers – Programme d'import

2.2.1 À la création

Le projet présenté possède deux programmes d'import, le premier, inclus dans le "parser" et est utilisé pour remplir la base de donnée à partir de trois fichiers :

- enseignes.csv → description des enseignes
- magasins.csv → descriptions des magasins
- donnees.csv → liste des produits à ajouter

L'ensemble des fichiers de données qui transitent sur le site possèdent l'extension '.csv' (Comma-separated values), sous la forme : donnée1|donnée2|donnée3| par ligne d'ajout.

Ce système autorise l'ensemble des accentuations et caractères spéciaux, mais l'utilisation des caractères " et ' est déconseillé. La sécurisation contre les failles de type XSS (injection de code par le biais d'un formulaire) est de mise dans la plupart des scripts.

Le parser ouvre en premier le fichier d'enseignes (fopen 'r'), puis parcourt ligne par ligne l'ensemble du fichier.

Pour chaque ligne il explose chaque élément dans un tableau [preg_split('#\\#', \$ligne)], et insère un nouveau tuple dans la table (mot de passe = nom de l'enseigne).

Il fait de même avec le fichier magasin, en attribuant aléatoirement à chaque magasin une enseigne et un mot de passe. Le mot de passe sera composé de 12 caractères alphanumériques minuscules (36 caractères au total) sélectionnés aléatoirement.

L'ajout de produit fonctionne sur le même principe, mais commence par ajouter dans la table le type et le rayon du produit (si inexistants), et ajoute le produit après avoir généré sa référence aléatoirement (même principe mais majuscule).

Enfin, un script va ajouter aléatoirement \$taille_echantillon_prix prix dans la base (sélection aléatoire d'un produit, sélection aléatoire d'un magasin, attribution d'un prix aléatoire entre 0 et \$prix_max). Cette partie du code est très lourde et dans le cadre d'une implémentation réelle, ne doit pas être exécutée.

Pour toute information complémentaire sur le remplissage de la base de donnée à la création, se référer au code php du fichier parser.php, dont les commentaires expliquent clairement toute la démarche.

2.2.2 À l'utilisation

Le second programme d'import est situé dans la partie administration de magasin. Cette page permet à un magasin de modifier les prix pratiqués et la disponibilité des produits.

Une page "generateur_csv.php" récupère les informations tarifaires du magasin en question (nécessite d'être connecté), et construit un fichier .csv contenant le nom, le type, le prix et la disponibilité des produits. Ce fichier est facilement éditable via un tableur (Excel, OpenOffice Calc). Une fois modifié (à la main ou par un programme interne au magasin), un simple envoi du fichier via la page "liste_produits.php", mettant à jour toutes les données en un seul clic. À nouveau nous vous invitons à vous rendre dans le fichier "generateur_csv.php" et "liste_produits.php" pour plus

de renseignements.

2.3 Principales requêtes

- `DELETE FROM Tarif WHERE Tarif.id_p in (SELECT id_p FROM Produit WHERE type in (SELECT type FROM Type WHERE rayon="$_POST['nom_rayon']."))';`

Cette requête, présente dans l'administration principale, permet de supprimer tous les tarifs des produits dont la catégorie correspond au rayon passé en paramètre.

- `SELECT id_p ident,nom_p,description,prix FROM Produit NATURAL JOIN Tarif WHERE 0 '$desc.$nom.$ref.' HAVING Prix <= ALL (SELECT prix FROM Tarif WHERE id_p=ident)'`

Ceci est la principale requête du site, elle permet la recherche de tous les produits correspondants à la demande de l'utilisateur. Les variables \$desc, \$nom et \$ref représentent les conditions de sélection (a or b or c) à partir de l'expression régulière : REGEXP "[[:<:]]'.\$recherche.'[[:>:]]" où \$recherche est la liste des mots rentré par l'utilisateur. Le code se situe dans le fichier 'resultat.php'.

- `SELECT Produit.id_p, type, nom_p,dispo,prix FROM Produit LEFT OUTER JOIN (SELECT id_p,prix,dispo FROM Tarif WHERE id_mag="$_SESSION['id_mag'].") test ON Produit.id_p = test.id_p ORDER BY dispo DESC,type,nom_p`

Cette requête, utilisée dans la liste des produits d'un magasin (édition des tarifs), permet de récupérer la liste des produits du site (id_p,nom_p,type, prix et dispo), qu'ils soient ou non dans le magasin passé en paramètre \$_SESSION['id_mag'], et de les trier selon leur disponibilité, leur type et leur nom.

3 Bilan

3.1 Limites de notre travail

Lors de ce projet, le temps à jouer en notre défaveur. Le sujet que nous avons choisi permet de faire quelque chose d'utile et nous avons manqué de temps pour construire quelque chose de vraiment professionnel. Ainsi, il est possible d'améliorer cette ébauche. Dans les améliorations possibles, nous pouvons citer qu'un graphisme plus attractif serait le bienvenu. De même, nous pourrions afficher une photo du produit en plus de sa description. Pour chaque magasins, nous aurions pu afficher le moyen de s'y rendre via une google map. D'autre part avec un peu plus de temps, nous aurions pu nous pencher sur une meilleur méthode pour récupérer les produit dans nos tables. En effet, même si le produit existe réellement, son prix reste fictif. Il en est de même pour les entreprises, même si les enseignes et magasins qui y sont attachés nous font penser à des commerces qui existent vraiment, la présence du produit où non dans tel ou tel magasin n'est pas vérifié. Les responsables des enseignes ne sont pas des personnes réelles.

3.2 Conclusion

Ce projet nous a permis de mettre en pratique l'ensemble des acquis de CM/TD/TP de base de données. Nous pris mesure de la quantité de travail à fournir pour aboutir à un site utilisable, ainsi que l'organisation et le travail d'équipe nécessaire à la conception d'un comparateur de prix complet et viable. Nous avons pu expérimenter des outils de gestion de version (Git et la plateforme github) et de management de projet (Google Wave) pour accélérer notre efficacité. Nous regrettons de n'avoir pu implémenter un système de récupération de données sur des sites réels, ou d'utiliser des bases de données existantes. Il aurait été intéressant d'étudier dans le cadre de ce module la sécurité internet dans les échanges de données (magasin, base de donnée), ainsi qu'une utilisation dynamique de nos tables via JavaScript (Ajax).

4 ANNEXE

