COMPTE RENDU

TER : Import Gedcom - MySQL

Davy Dequidt
Arnaud Federbe

Tuteur : J. Guizol

Sommaire

Compte rendu

Les Acteurs

Nom ***Téléphone*** ***E-mail***	DEQUIDT Davy _____ _____
Nom ***Téléphone*** ***E-mail***	FEDERBE Arnaud _____ _____

Tuteur : J. Guizol

Objectifs

Nous travaillons à la base sur un site existant développé en PHP (Genea4p). Ce site permet la gestion de généalogies en ligne dont les informations sont inscrites dans une base de données MySQL.
Un format de fichier standard pour partager une généalogie est déjà défini, le format Gedcom (la version 5.5 est la plus utilisée).

Le but est de récupérer auprès d'un client des informations dans un fichier au format Gedcom pour les transcrire dans la base de donnée MySQL.

Cependant la base de données ne supporte pas toutes les informations fournies par un fichier Gedcom. Il est donc nécessaire de structurer le script pour d'éventuelles modifications futures (suite à une modification de la base de données),

Spécifications Techniques

Le format Gedcom

Gedcom est un protocole destiné à transférer les données entre les logiciels de généalogie qui offrent les fonctions dites Gedcom. Le contenant intermédiaire est un fichier informatique. Il est écrit par le logiciel exportateur. Il sera lu par le logiciel importateur.
Le format fait appel au mode texte. Toutes les données sont représentées par des caractères alphanumériques. Ce fichier peut donc être lu/ouvert par un éditeur pratiquement sur toutes machines et tout systèmes d'exploitations.
Pour exploiter ces données elles doivent être organisées (structurées) selon une procédure connue des personnes qui les utilisent.

Structure du fichier
Chaque ensemble de caractères constituant une donnée doit occuper une ligne de longueur maximale de 255 caractères. Chaque ligne débute par un signet (tag) représentatif de la nature des informations. Elle a une place unique dans une structure arborescente dont les niveaux sont repérés par un numéro, croissant selon une hiérarchie décroissante.
La racine étant le fichier on trouve d'abord 10 types d'enregistrements repérés par le niveau 0. Ensuite chaque enregistrement reçoit une référence pour distinguer chaque utilisation. Par convention cette référence est encadrée par le signe @.
Voici les 10 types d'enregistrement avec le tag conventionnel et la nature des données:

- HEAD en tête du fichier avec les paramètres
- FAM enregistrement de données pour une famille
- INDI enregistrement de données pour un individu
- NOTE enregistrement de données pour une note
- SOUR enregistrement de données pour une source
- REPO enregistrement de données pour une archive
- OBJE enregistrement de données pour un document
- SUBM enregistrement de données pour un rédacteur
- SUBN enregistrement de données pour soumission
- TRLR marque de fin de fichier

Ce qui conduit à une première ligne d'un enregistrement par exemple
0 @xxxx@ INDI
0 @xxxxxxx@ SOUR

A la suite viendront des lignes de détail au niveau 1 puis à l'intérieur de chaque niveau 1 des compléments, affectés au niveau 2. Pour une personne on peut écrire
0 @xxxxx@ INDI (création d'un enregistrement d'individu)
1 NAME prénom/nom/ (indication du prénom et du nom)
1 BIRT (les données du niveau supérieur concernent la naissance)
2 DATE jj MMM aaaa (la date de naissance)

2 PLAC village (le lieu de naissance)
1 DEAT (les données du niveau supérieur concernent le décès)
...
0 @xxx@ INDI (une autre personne).
Ce schéma simple permet de décrire parfaitement l'ensemble des informations. Un ensemble de tags est défini dans les règles Gedcom pour couvrir les besoins. Cependant il ne permet pas de représenter des liens par exemple entre personnes.
Ainsi une famille est composée d'un homme et d'une femme, alors au lieu de reprendre les données de chaque personne il sera fait appel seulement à leur référence.
0 @fxxx@ FAM (création d'un enregistrement famille)
1 HUSB @ind215@ (l'époux est la personne décrite dans l'enregistrement @ind215@)
1 WIFE @ind12@ (l'épouse est la personne décrite dans l'enregistrement @ind12@)
1 MARR (éléments concernant le mariage)
2 DATE jj MMM aaaa (date)
2 plac un village (lieu)
1 .....

Nous disposons maintenant d'un moyen pour écrire toutes les relations dans nos généalogies en imbriquant ce mécanisme de pointage dans l'arbre des données.

Pour plus d'informations se reporter au document : « THE GEDCOM STANDARD – Release 5.5 » disponible à l'adresse :
http://genea4p.espace.fr.to/gedcom55.pdf

La base de données

La base de données est gérée par un moteur Innodb ce qui permet la gestion de relation, la base est structurée par 8 tables principales et 16 autres tables liant celle-ci entre elles, voir l'annexe pour plus d'informations et de détails.

Contraintes

Le script ayant pour but d'être utilisé par un grand nombre, il doit être implantable chez la plupart des hébergeurs (support PHP, MySQL Innodb). Les hébergeurs limitent le temps d'exécution d'un script entre 30 et 180 secondes.

Les fichiers Gedcom peuvent être d'une taille importante ce qui peut poser problème à l'exécution du script dans cette limite de temps.

Solutions techniques

Langage utilisé

Pour ce projet nous avons deux solutions:

- exécuter le programme sur le serveur : l'accès à la BD est rapide cependant l'exécution d'un script est limité en temps
- exécuter le programme sur la machine cliente : pas de limitation de ressource cependant cette solution nécessite l'envoi de requête au serveur (taille plus importante qu'un fichier gedcom, risque de faille de sécurité).

La première solution est plus avantageuse puisque l'inconvénient majeur de la limite de temps est maîtrisable. En effet, le format gedcom permet une analyse segmentée.
Par conséquent, le site étant codé en PHP, nous utilisons ce langage.

Méthode utilisée

Bufferisation des requêtes SQL

Afin de limiter le nombre de transferts entre le serveur et la BD, les requêtes sont stockées dans un buffer et envoyées régulièrement (lorsque celui-ci atteint une limite fixée).

Découpage

Comme expliqué ci-dessus dans les spécifications techniques, le format Gedcom est structuré en blocs. L'analyse du fichier est effectuée par bloc de niveau 0 ainsi on peut arrêter le script entre 2 blocs si le temps d'exécution est supérieur à la limite fixée.
Cette limite correspond au temps maximum d'exécution du serveur avec une marge de sécurité de 5 secondes permettant vider le buffer des requêtes SQL et de finir le script « proprement ».
L'analyse du fichier reprend ensuite après l'action de l'utilisateur (lien « Continuer l'analyse ») à la ligne du fichier où le script s'est interrompu précédemment.

Gestion des références : 2 cas

Références XREF du Gedcom – identifiants des tables BD

Il est possible d'importer deux fichiers Gedcom différents, cependant, par exemple, il se peut qu'un individu de chaque fichier ait le même identifiant. Il est donc impossible d'insérer directement des identifiants XREF comme clé primaire des tables. De plus, la base de données utilise des identifiants (clé primaire) de type entier, alors que le format Gedcom autorise une chaîne de caractères alphanumériques.
Nous devons donc générer des identifiants pour la BD correspondant aux références XREF.
Le tableau de correspondance est conservé dans une variable de session.

Liens entre les tables de la BD

La BD contient des tables représentant les liens entre les différents blocs, ces tables contiennent des clés étrangères, il faut donc que les tables principales soient remplies pour insérer ces liens. L'insertion dans la BD est effectuée en deux étapes : le remplissage des tables principales puis celles des liens (Voir Annexe Base de données pour visualiser les dépendances entre les tables).
Les liens sont conservés dans des fichiers, chacun d'eux correspondant à une table.

Analyse

Encodage des caractères

Le format gedcom supporte trois types d'encodages :

- 8-Bit ANSEL
- ASCII (USA Version)
- UNICODE (ISO 10646)

Or, les données de la base sont encodées en UTF-8, Une conversion est donc nécessaire.
La fonction utf8_encode de PHP permet cette conversion sauf pour certains caractères de l'encodage ANSEL. Nous avons utilisée une table de correspondance pour la conversion (Exemple : âe <=> é).

Contrôle des tags

Nous avons fait une liste de tous les tags supportés par le format Gedcom. De plus nous sélectionnons seulement les tags utilisés par la base de données, les autres étant ignorés.

Contrôle des champs

Le format Gedcom définit les types de champs par leur taille et leur format (entier, chaîne de caractères ...). Certains types font références à d'autres.

Pour contrôler ces types, nous avons créé une table qui contient :

- la taille
- les références vers un ou plusieurs autres types
- les valeurs fixe (Ex : EVENT_TYPE_INDIVIDUAL := ADOP | BIRT | BAPM...)

Une fonction générique (data.php : type_check()) utilise ces données ainsi que des expressions régulières pour contrôler ces types de champs.

Analyse des blocs

Deux types de blocs sont définis par le format Gedcom :

- les « Record structures » (blocs)
- les « substructures » (sous-blocs)

L'analyse d'un fichier Gedcom correspond à la structure de la grammaire du format.
C'est à dire, chaque bloc et sous-blocs sont analysés par une fonction les caractérisant.

Chaque ligne contient les informations suivantes :

- un niveau
- un tag
- une description

Selon le niveau, le format du bloc requiert un tag.
Selon le tag lu, soit le type de la description est contrôlé, soit une autre fonction de bloc est appelée.

Exemple d'une ligne du format gedcom :

```
1     NAME    Henri/DEQUIDT/
-     ----   ---------------
|       |           |-> Description
|       |-> Tag
|
|-> Niveau
```

Gestion des lieux

Ordre des informations

Les lieux sont caractérisés par un ensemble d'informations (commune, région, pays ...). Ils sont représentés dans le fichier Gedcom par une seule ligne, chaque information étant séparée de la suivante par une virgule.
Il n'existe pas de norme spécifiant l'ordre de ces informations (cela dépend des logiciels utilisés pour générer le fichier Gedcom).
L'utilisateur doit donc spécifier cet ordre manuellement.
Pour cela, on affiche au maximum les 15 premiers lieux différents rencontrés ce qui facilite le choix de cet ordre. (Voir annexe p.6)

Économie de la mémoire

Étant donnée la taille importante des chaînes de caractères représentants les lieux, il est préférable de ne pas utiliser la même méthode que les tableaux de correspondant XREF – Identificateur. Par souci d'économie de mémoire, nous utilisons une fonction de hachage (md5) qui réduit la chaîne en une suite de 32 caractères hexadécimaux. Ce haché est enregistré dans la table de correspondance avec l'identificateur de ce lieux dans la BD.

Récupération des informations

La lecture du fichier est linéaire et effectuée en une seule passe. Il y a deux types d'informations à récupérer, d'une part les données correspondant à un bloc (et leurs sous-blocs), d'autre part, les liens entre ces différents blocs.

Récupération des données

- des sous-blocs

Chaque donnée correspondant à un tag est enregistrée dans un même tableau, celui-ci étant retourné comme le résultat de la fonction.

- des blocs

Chaque donnée correspondant à un tag est enregistrée dans un même tableau. Les données récupérées par les sous-blocs sont aussi insérées dans ce tableau.
A la fin de l'analyse du bloc, les données du tableau sont formatées en une requête SQL et insérées dans le buffer.
Le sous-bloc EVENT_DETAIL (informations sur un évènement) est considéré comme un bloc par la base de données. En effet, il existe une table d'évènements (genea_events).

Récupération des liens entre les blocs

Si un tag fait référence à un autre bloc (XREF), on ajoute le lien dans un fichier correspondant à son type (ex : indi_event).
Si un tag fait appel à un sous-bloc EVENT_DETAIL, un lien est ajouté de la même façon.

Insertion dans la base

Allocation de l'espace

Dans un fichier Gedcom, des liens peuvent faire référence à des blocs présents dans la suite du fichier. Les identificateurs des ces futurs blocs doivent donc être fixées avant que ces blocs soient insérés dans la base de données. Par conséquent, pour éviter tout conflit d'identificateur (clé primaire), il est nécessaire de réserver des lignes dans la BD.

Pour régler ce problème, nous effectuons les deux étapes suivantes :

- Comptage du nombre de blocs

Lors d'une première lecture du fichier, on distingue chaque bloc par ligne de niveau 0. La fonction loadNbBlocs() incrémente le nombre de blocs correspondant au type de tag lu (INDI,FAM...).
Il existe deux cas supplémentaires :
- Les évènements correspondant à une multitude de tags (ex : EVEN, BIRT, BAPM ...) sont aussi comptés.
- Les lieux (tag PLAC) sont également comptés. (cf « Gestion des lieux »)

- Réservation de l'espace

Les clés primaires des tables de la base de données utilisent un auto-incrément géré par le moteur MySQL – InnoDB. Chaque nouvelle ligne insérée, utilise comme clé primaire la valeur de l'auto-incrément ; celui-ci étant incrémenté juste après.
Pour réserver un intervalle d'identificateur (correspondant au nombre de blocs comptés), il suffit donc de changer la valeur de cet auto-incrément (en l'augmentant du nombre de blocs). Les lignes seront insérés dans cet intervalle.

Insertion dans l'ordre de dépendance

Dans la base de données, certaines tables utilisent des clés étrangères nécessitant l'existence d'une même clé en tant que clé primaire d'une autre table. Ceci impose donc un ordre dans l'insertion des données dans la base de données (cf Annexes « Base de données ») :

1) Insertion des lieux

Le fichier Gedcom est lu avant son analyse pour extraire les informations concernant les lieux (cf « Gestion des lieux »). Les données de chaque nouveau lieu sont formatées en requête SQL puis inséré dans le buffer.

2) Insertion des blocs

Pour chaque bloc, on insère les données récupérées durant l'analyse de celui-ci.
Certains blocs sont traités différemment pour satisfaire les dépendances :

- les familles : toutes les données sont conservées dans un fichier texte pour être insérées ultérieurement.
- les sources : la clé étrangère « repo_id » (référence à un « repository ») est sauvegardée dans un fichier (avec l'identificateur de la source). Le reste est inséré dans la base de données. Le champs « repo_id » sera mis à jour ultérieurement.

3) Insertion des familles

On insère les données des familles sauvegardées précédemment dans le fichier texte.

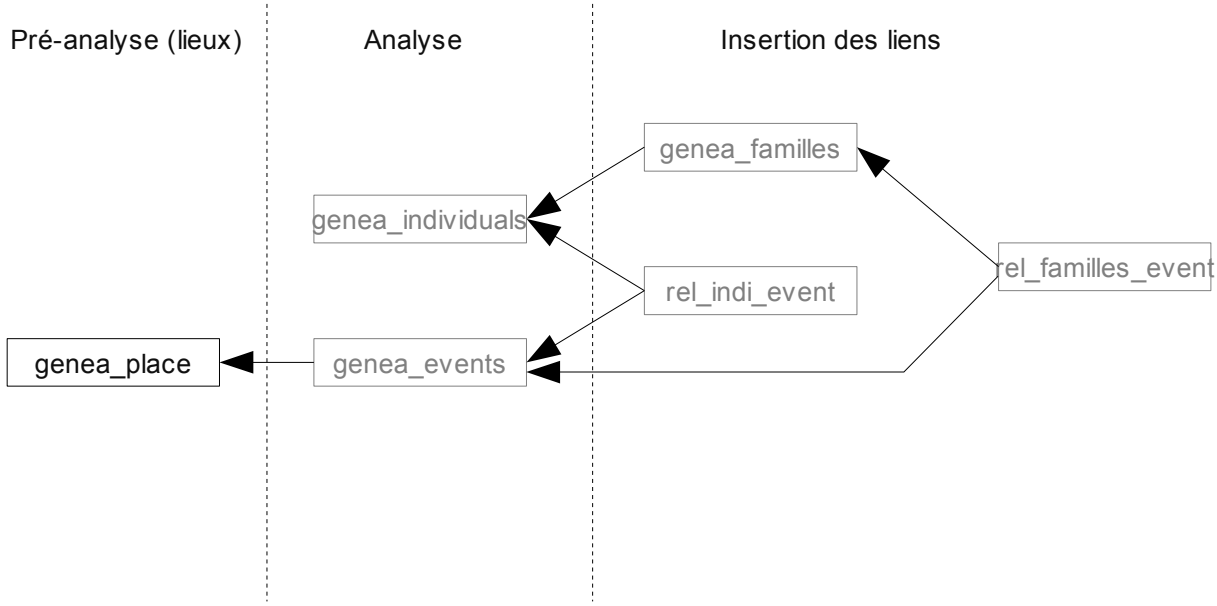
4) Insertion des liens dans l'ordre

Les liens sauvegardés pendant l'analyse du fichier Gedcom sont insérés selon leur dépendances mutuelles.

5) Mise à jour du lien source.repo_id

On met à jour les enregistrements de la table genea_sources pour insérer les liens présents dans le fichier correspondant.

Exemple d'une partie de l'insertion des données dans l'orde de dépendance.



Gestion des erreurs

Nous avons divisé les erreurs en deux catégories, d'une part les erreurs critiques qui interrompent le script et annulent les actions effectuées auparavant, d'autre part les erreurs minimes qui peuvent avoir lieu pendant l'analyse ; ces dernières sont signalées à l'utilisateur.

Les erreurs critiques

Certaines erreurs peuvent être constatées avant l'analyse, notamment les droits d'accès sur les fichiers (droit d'écriture sur les fichiers utilisés pour conserver les liens, droit de lecture sur le fichier Gedcom).
Les autres erreurs peuvent provenir des requêtes SQL qui sont inattendues. C'est à dire des erreurs indépendantes des éventuelles incohérences du fichier Gedcom.

Les erreurs minimes

Ces erreurs proviennent des incohérences du fichier Gedcom ou du non respect de son standard. Nous les référençons dans un journal d'erreurs pour en informer l'utilisateur à la fin de l'exécution du script (voir annexe p.7). Dans ce journal, les évènements sont de trois types :

- les erreurs : standard Gedcom
- les avertissements : ligne(s) ignoré(s) suites à une erreur ou tag non standard
- debug : affiche des informations qui permettent de localiser des éventuelles erreurs produites durant le développement.

Le fichier Gedcom peut contenir des liens vers des blocs inexistants. Ainsi lors de l'insertion des liens, si certaines contraintes (clés étrangères) ne sont pas satisfaites, ceci génère alors des erreurs SQL, celles-ci sont seulement affichées.
Pour le moment ces erreurs SQL ne peuvent pas être parfaitement traitées, il faut attendre la version 5 de SQL innodb stable et approuvée par les développeurs, celle-ci permettra entre autre la gestion des triggers.

Interruption du script

Elle provient d'une coupure de connexion, d'une perte de la session ou d'une interruption due de l'utilisateur. Dans ce cas, le script reprend à son état initial tout en supprimant les données précédemment insérées.

Conclusion

Ce projet nous a apporté la possibilité d'acquérir de nouvelles connaissances dans le langage php4 et une certaine maîtrise du format standard Gedcom 5.5.
Nous avons mis en pratique les connaissances acquises durant cette année ainsi que les précédentes, notamment nous avons utilisés la technique du hachage ainsi que nos compétences en base de données. De plus la formation de génie logiciel nous a permis de nous organisé afin de concilier ce projet, les révisions pour les examens ainsi que nos activités professionnelles et extra-universitaires.
Enfin le projet a apporté au site des possibilités plus importante qui permettront de satisfaire les utilisateurs en attente d'une fonction d'importation de fichier gedcom dans la base de données du site.