

INFO-H-303 : Bases de données

Projet : Annuaire d'établissements horeca

Professeur : Esteban Zimányi
<http://cs.ulb.ac.be/public/teaching/infoh303>
Année académique 2015-2016

Étude de cas

On vous demande de créer un annuaire en ligne pour les établissements horeca de Bruxelles.

Un établissement est soit un restaurant, soit un bar, soit un hôtel. Il est caractérisé par son nom, son adresse (rue, numéro, code postal, localité), ses coordonnées GPS (longitude et latitude), son numéro de téléphone et éventuellement un lien vers son site web officiel. Pour un restaurant, on désire retenir la fourchette de prix d'un plat, le nombre de places maximum pour un banquet, si l'on peut emporter des plats, s'il dispose d'un service de livraison et ses demi-journées de fermeture. Pour un bar on enregistre s'il est fumeur et s'il fait de la petite restauration. Pour un hôtel, on veut connaître le nombre d'étoiles, le nombre de chambres et un indice de prix pour une nuit en chambre double.

Pour chaque utilisateur, on retient son identifiant, son adresse email, son mot de passe et sa date d'enregistrement. Après s'être enregistrés et connectés, les **utilisateurs** peuvent **faire des commentaires** sur un établissement. Un commentaire est composé d'un **score sur 5** et d'un **texte**. Les utilisateurs peuvent **commenter plusieurs fois** le même établissement à des dates différentes. Les utilisateurs peuvent également **labelliser (*tagger*)** un établissement ("terrasse", "cuisine française", "bon rapport qualité prix", "parking", etc). Pour ce faire, ils peuvent choisir dans la **liste des labels (*tags*)** déjà utilisés ou **créer un nouveau label**. Un utilisateur **ne peut apposer le même label plusieurs fois sur le même établissement**. Un utilisateur peut être **administrateur** et dans ce cas, il a le pouvoir de **créer, modifier et supprimer un établissement**. On désire **retenir à quelle date et par quel administrateur un établissement a été créé**.

L'interface doit permettre de consulter la fiche d'un établissement contenant toutes ses propriétés, les commentaires associés, sa note moyenne ainsi que ses labels. Pour chaque label, on doit pouvoir différencier son poids (le nombre de fois que ce label a été apposé sur un établissement). L'interface doit également **fournir un moteur de recherche basé sur le type, le nom, la commune et les labels**. On devra également avoir accès à la fiche de chaque utilisateur pour connaître la liste des établissements qu'il a commenté et les labels qu'il a apposés.

Les informations de départ vous seront fournies au format XML suivant le modèle ci-dessous.

```
<Restaurant creationDate="2/10/2008" nickname="fred">
  <Information>
    <Name>Grenier d'Elvire</Name>
    <Address>
      <Street> Chaussée de Boondael</Street>
      <Num>339A</Num>
      <Zip> 1050 </Zip>
      <City>Ixelles</City>
      <Longitude>4.38384</Longitude>
      <Latitude>50.818766</Latitude>
    </Address>
    <Site link="http://www2.resto.be/grenierdelvire/">
```

```

<Tel>02/648 43 48</Tel>
<Closed>
  <!-- am = fermé le midi, pm = fermé le soir -->
  <!-- day 0 = lundi, ..., day 6 = dimanche -->
  <On day="0" hour="am"/>
  <On day="1" hour="am"/>
  <On day="6" hour="am"/>
</Closed>
<TakeAway/>
<Delivery/>
<PriceRange>20</PriceRange>
<Banquet capacity="25"/>
</Information>
<Comments>
  <Comment nickname="boris" date="2/10/2008" score="4">texte</Comment>
  ...
</Comments>
<Tags>
  <Tag name="Bon rapport qualité/prix">
    <User nickname = "fred"/>
    ...
  </Tag>
  ...
</Tags>
</Restaurant>

```

Les cafés sont modélisés de la même manière. Leur élément principal est `<Cafe>`, ils possèdent l'élément `<Smoking/>` s'ils sont fumeurs et l'élément `<Snack/>` s'ils proposent de la petite restauration. Les données des hôtels ne sont pas fournies.

Ces données devront être importées dans votre base de données.

Déroulement du projet

Première partie

Pour cette partie, on vous demande de modéliser le problème à l'aide du formalisme entité-association et de préciser les contraintes d'intégrité nécessaires. Ces contraintes doivent être exprimées en français et utiliser les mêmes noms d'entités, d'associations ou d'attributs que dans votre modèle conceptuel. Vous pouvez également exprimer et justifier des hypothèses sur votre modèle. Ces hypothèses peuvent résulter par exemple d'incomplétudes ou d'ambiguïtés dans l'énoncé du problème.

Déduisez ensuite de ce modèle conceptuel le modèle relationnel correspondant ainsi que ses contraintes. Vos choix de modélisation doivent être justifiés.

Ces modèles doivent être suffisamment riches pour pouvoir servir de support à l'application décrite ci-dessous ainsi que pour pouvoir répondre aux requêtes de la deuxième partie.

Deuxième partie

Création

On vous demande tout d'abord de déduire de votre modèle relationnel un script SQL DDL de création de la base de données et de ses différentes tables ainsi que de créer cette base de données.

Initialisation

On vous demande d'écrire un script permettant d'importer les fichiers XML représentant les établissements dans votre base de données. Ces données devront être présentes dans votre base de données lors de la défense. Les fichiers XML seront disponibles le 7 novembre sur la page web du projet.

Application

Nous vous demandons ensuite de développer une interface graphique pour votre base de données permettant au minimum de réaliser les opérations ci-dessous :

- enregistrer et connecter des utilisateurs
- ajouter des commentaires sur les établissements
- ajouter et créer des labels sur des établissements
- consulter les fiches des établissements et des utilisateurs
- rechercher des établissements
- ajouter et modifier des établissements
- supprimer des établissements
- visualiser le résultat des requêtes décrites ci-dessous à l'endroit de votre application où cela vous semble pertinent

Votre application devra bien entendu veiller à ce que la base de données reste cohérente.

Vous pouvez bien sûr ajouter des fonctionnalités à votre application comme par exemple la gestion des photos des utilisateurs ou établissements, un système de vote sur les commentaires, la localisation des restaurants sur une carte, l'affichage des labels avec une taille différente en fonction de leur poids, l'internationalisation, etc. Ces apports personnels seront valorisés.

Requêtes

Nous vous demandons d'écrire en algèbre relationnelle et calcul relationnel tuple les requêtes R1, R2, R3 et R4 ainsi que toutes les requêtes en SQL. Si une requête vous semble imprécise, indiquez dans votre rapport les hypothèses que vous avez faites pour lever ces imprécisions.

Dans les requêtes suivantes, on dit qu'un utilisateur apprécie un établissement s'il lui a attribué un score de 4 ou 5 dans au moins un de ses commentaires.

- R1 : Tous les utilisateurs qui apprécient au moins 3 établissements que l'utilisateur "Brenda" apprécie.
- R2 : Tous les établissements qu'apprécie au moins un utilisateur qui apprécie tous les établissements que "Brenda" apprécie.
- R3 : Tous les établissements pour lesquels il y a au plus un commentaire.
- R4 : La liste des administrateurs n'ayant pas commenté tous les établissements qu'ils ont créés.
- R5 : La liste des établissements ayant au minimum trois commentaires, classée selon la moyenne des scores attribués.
- R6 : La liste des labels étant appliqués à au moins 5 établissements, classée selon la moyenne des scores des établissements ayant ce label.

Rapports

Pour la première partie, on vous demande les documents suivants :

- un diagramme entité-association modélisant le projet ainsi que ses contraintes
- une traduction relationnelle de ce diagramme et ses contraintes
- vos hypothèses et la justification de vos choix de modélisation

Le formalisme utilisé doit être un de ceux vus au cours ou aux TPs.

Pour la deuxième partie, vous rendrez un rapport contenant :

- les documents de la première partie tenant compte des remarques des assistants
- le script SQL DDL de création de la base de données
- les requêtes demandées

- les instructions d'installation de votre application
- un scénario de démonstration de votre application
- les explications et justifications de vos choix et hypothèses

Le code source de votre script d'insertion et de votre application devra être rendu par mail à mwaumans@ulb.ac.be avant votre défense.

Informations pratiques

- Le projet se fera obligatoirement par groupe de deux.
- La première partie devra être rendue par email à mwaumans@ulb.ac.be pour le Lundi 11 Avril.
- Le rapport de la seconde partie devra être envoyé avec une archive contenant tous les codes sources à l'adresse mwaumans@ulb.ac.be pour le Vendredi 13 Mai. La défense du projet la semaine suivante selon un horaire à déterminer. On vous demandera de présenter votre application pendant 10 minutes et de répondre à quelques questions.
- Le projet comptera pour 25% de la note finale du cours.
- Les apports personnels au projet seront valorisés à hauteur de 4 points sur 20. En d'autres mots, un projet parfait sans apport personnel aura une valeur de 16 points sur 20.
- Sauf mention explicite, vous pouvez utiliser les langages et outils de votre choix (MySQL, PostgreSQL, PHP, Java, Python, ...).
- Vous pouvez développer sur votre propre machine et présenter vos projets sur un ordinateur portable que vous apporterez lors de la défense. Au besoin, nous mettrons à votre disposition une base de données MySQL ainsi qu'un espace web PHP sur un serveur de l'Université.

Bon travail !