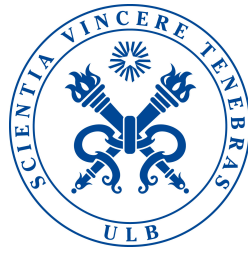


UNIVERSITÉ LIBRE DE BRUXELLES



INFO-H303 : PROJET BASES DE DONNÉES  
ANNUAIRE D'ÉTABLISSEMENTS HORECA

---

# Rapport

---

*Auteurs :*

Sacha MEDAER  
000393629

Nicolas FERON  
000363904

*Titulaire du cours :*

Esteban ZIMANYI

*Assistant :*

Michaël WAUMANS

13 mai 2016

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Hypothèses et choix de modélisation</b>	<b>2</b>
<b>3</b>	<b>Diagramme entité-association</b>	<b>3</b>
3.1	Remarques diagramme . . . . .	4
3.2	Contraintes d'intégrité . . . . .	4
<b>4</b>	<b>Modèle relationnel</b>	<b>5</b>
<b>5</b>	<b>Instructions d'installation de l'application</b>	<b>6</b>
<b>6</b>	<b>Requêtes</b>	<b>6</b>
6.1	Algèbre relationnelle . . . . .	7
6.2	SQL . . . . .	8
<b>7</b>	<b>Démonstration de l'application</b>	<b>8</b>
<b>8</b>	<b>Création de la base de données</b>	<b>9</b>
<b>9</b>	<b>Conclusion</b>	<b>11</b>

## 1 Introduction

Ce projet s'inscrit dans le cadre du cours de bases de données en deuxième année de Bachelier. Il est demandé de créer une base de données qui correspond à un annuaire en ligne pour les établissements Horeca de Bruxelles. L'utilisateur peut consulter la fiche d'un établissement contenant toutes ses propriétés, commentaires, note moyenne et labels. Cette interface offre également un moteur de recherche ainsi que l'accès à la fiche détaillée de chaque utilisateur. Une interface graphique permet de gérer la base de données. L'application est un site Web créé grâce à un serveur Apache, MySQL et le langage de script PHP.

## 2 Hypothèses et choix de modélisation

Cette section est destinée à préciser certains points de l'énoncé.

Un label est considéré comme un élément indépendant, c'est à dire que si un label a été apposé pour un seul établissement et que cet établissement est supprimé de la base de données, la label continue d'exister. Seul un administrateur a le droit de créer un établissement. Et un établissement ne peut être modifié ou supprimé que par l'administrateur qui l'a créé. Les commentaires et les labels sont considérés comme deux entités dérivant d'une seule entité description. Les hotels, cafés et restaurant sont considérés comme trois entités dérivant d'une seule entité établissement.

### 3 Diagramme entité-association

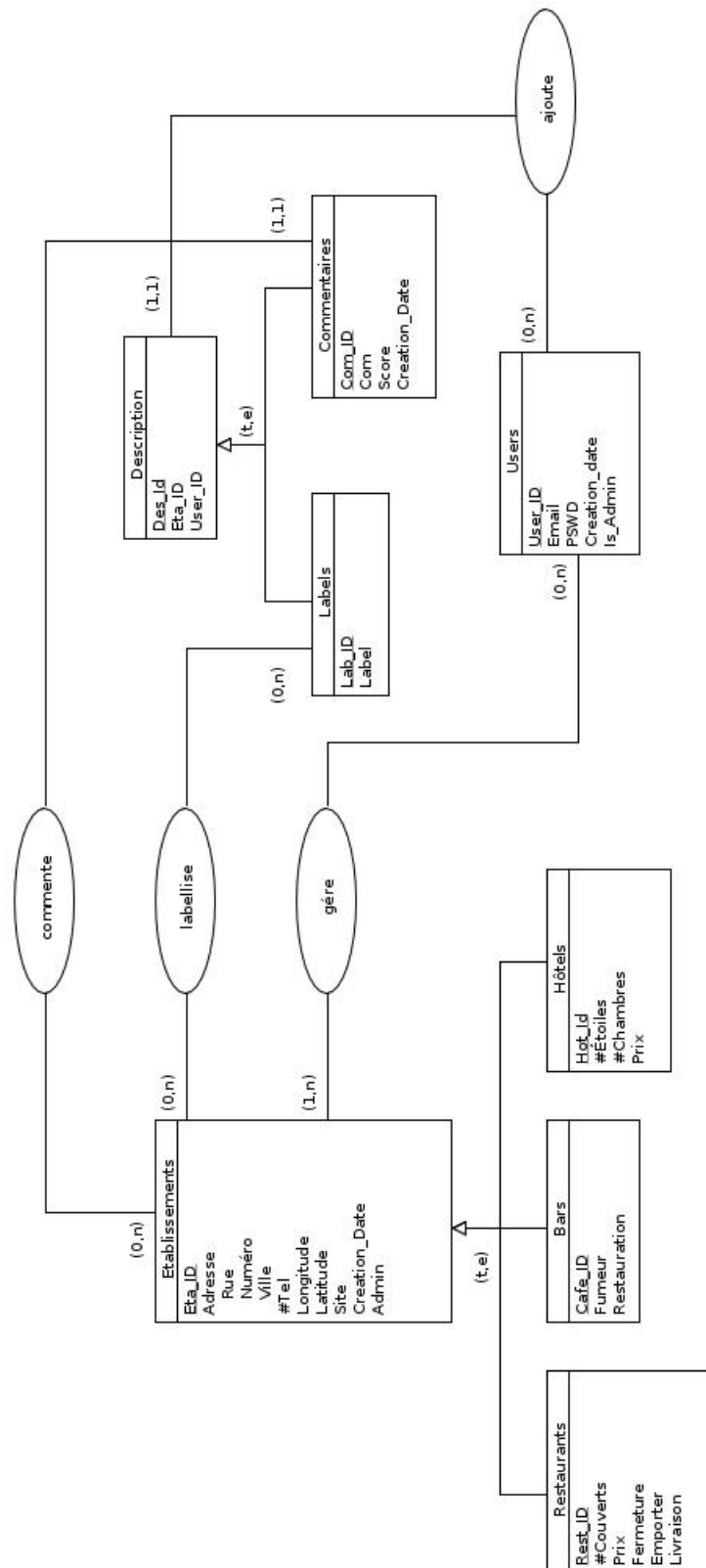


FIGURE 1 – Diagramme entité-relation de la base de donnée

### 3.1 Remarques diagramme

- L'entité Descriptions a été ajoutée car des attributs communs sont partagés entre les Commentaires et les Labels (exemple : établissement concerné).
- Un label peut ne décrire aucun établissement en particulier (dans le cas où le seul établissement qu'il décrit est retiré de la BDD).

### 3.2 Contraintes d'intégrité

- Un utilisateur ne peut pas apposer le même label plusieurs fois sur le même établissement.
- Le nombre de couvert(s) d'un restaurant doit être strictement supérieur à 0.
- La date d'enregistrement d'un utilisateur doit être inférieure à la date d'une description ajoutée par cette utilisateur.
- La date d'enregistrement d'un admin doit être inférieure à la date de création d'un nouvel établissement.
- Le nombre d'étoiles d'un hôtel doit être compris entre 0 et 5 (inclus).
- Le score entré par un utilisateur lors d'un commentaire doit être compris entre 0 et 5 (inclus).
- Le nombre de chambre(s) d'un hôtel doit être strictement supérieur à 0.
- Un Etablissement.Eta\_ID correspond à exactement soit un Restaurants.Res\_ID, soit un Bars.Bar\_ID, soit un Hotels.Hot\_ID.
- Un Descriptions.Des\_ID correspond à exactement soit un Labels.Lab\_ID, soit un Commentaires.Com\_ID.

## 4 Modèle relationnel

Etablissements(Eta\_ID, AdRue, AdNuméro, AdVille, Tel, Latitude, Longitude, Site, Admin)

Admin référence Users.User\_ID

Restaurants(Res\_ID, Couverts, Prix, Fermeture, Emporter, Livraison)

Restaurant.Res\_ID référence Etablissements.Eta\_ID

Bars(Bar\_ID, Fumeur, Restauration)

Bars.Bar\_ID référence Etablissements.Eta\_ID

Hotels(Hot\_ID, Étoiles, Chambres, Prix)

Hotels.Hot\_ID référence Établissement.Eta\_ID

Users(User\_ID, Email, PSWD, Creation\_Date, Is\_Admin)

Ajoute(Des\_ID, User\_ID)

Des\_ID référence Descriptions.Des\_ID

User\_ID référence Users.User\_ID

Gère(Eta\_ID, User\_ID)

Eta\_ID référence Etablissements.Eta\_ID

User\_ID référence Users.User\_ID

Descriptions(Des\_ID, Eta\_ID, User\_ID)

Descriptions.Eta\_ID référence Etablissements.Eta\_ID

Descriptions.User\_ID référence Users.User\_ID

Labels(Lab\_ID, Label)

Labels.Lab\_ID référence Descriptions.Des\_ID

Labellise(Eta\_ID, Labels)

Eta\_ID référence Établissements.Eta\_ID

Labels référence Labels.SSN

Commentaires(Com\_ID, Com, Score, Creation\_Date, SID)

Commentaires.Com\_ID référence Descriptions.Des\_ID

Commente(Eta\_ID, Commentaires)

Eta\_ID référence Établissements.Eta\_ID

Commentaires référence Commentaires.SSN

## 5 Instructions d'installation de l'application

L'application est un site Web. Il a été développé sur base de logiciels libres. Une version temporaire en ligne est disponible à l'adresse suivante : <http://51.255.39.146/info/www/> . Les deux administrateurs sont Fred et Boris. Leurs identifiants et adresses mail pour accéder à leur compte respectif sont Fred@gmail.com et Boris@gmail.com. De même, leurs mots de passe sont Fredpswd et Borispswd.

Les informations qui suivent permettent une installation locale. Cette installation requiert au minimum quatre composants. Ci-dessous sont listés le(s) composant(s) qui ont été utilisé(s) pour l'implémentation de l'application.

- système d'exploitation : Windows ou distribution Unix
- un serveur web : Apache
- un système de gestion de bases de données : mySQL
- un langage permettant la génération des pages web et la communication avec le système de gestion de bases de données : PHP

Ci-dessous sont décrites les différentes étapes à suivre. Des exemples sont ensuite donnés pour chaque étape pour le système d'exploitation Linux et les composants cités ci-dessus.

- Créer la base de données à partir du fichier *DB\_creation.txt* dans le répertoire *db* ou en important le fichier *horeca.sql* dans mySQL.
- Copier le dossier *www/* dans le répertoire localhost du système d'exploitation.
- Ouvrir un navigateur et accéder au répertoire localhost dans la bar de recherche.

1. `shell> cat DB_creation.txt | mysql -u<username> -p<pswd> ou shell> mysql -u<username> -p<pswd> db_name < horeca.sql`
2. `shell> cp -r Horeca/ /var/www/`
3. localhost  
127.0.0.1  
<pc\_name>

## 6 Requêtes

Les requêtes suivantes concernent la base de données Horeca. Les quatre premières requêtes sont traduites en algèbre relationnelle. De plus, toutes les requêtes sont traduites en SQL.

Il est à noter que un utilisateur qui apprécie un établissement est un utilisateur qui a accordé un score d'au moins quatre à cet établissement dans son commentaire.

- 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.

Afin que les requêtes soient claires et lisibles, les différentes tables sont identifiées comme suit :

<i>Users</i>	U	<i>Etablissements</i>	E	<i>Restaurants</i>	R
<i>Cafes</i>	CA	<i>Descriptions</i>	D	<i>Commentaires</i>	CO
<i>Labels</i>	L				

N.B. : Le "au moins" dans la requête numéro 2 est ambigu. Tous les établissements de tous les utilisateurs qui apprécie tous les établissements que "Brenda" apprécie sont listés par la requête.

## 6.1 Algèbre relationnelle

- R1 :
 
$$\begin{aligned} \text{AllBrendaLike} &\leftarrow \pi_{Eta\_ID} (D \bowtie_{Des\_ID=Com\_ID \wedge Score>3 \wedge User\_ID='Brenda'} CO) \\ \text{Result} &\leftarrow \pi_{User\_ID} \left( \sigma_{User\_ID=User\_ID_3 \wedge Eta\_ID \neq Eta\_ID_2 \wedge Eta\_ID_2 \neq Eta\_ID_3} \right. \\ &\quad \left. (\alpha_{User\_ID:User\_ID_3, Eta\_ID:Eta\_ID_3} (\text{AllBrendaLike}) * \right. \\ &\quad \left. \alpha_{Eta\_ID \neq Eta\_ID_2 \wedge Id=Id_2} (\text{AllBrendaLike} * \alpha_{User\_ID:User\_ID_2, Eta\_ID:Eta\_ID_2} (\text{AllBrendaLike})) \right) \end{aligned}$$
- R2 :
 
$$\begin{aligned} \text{AllUserLike} &\leftarrow \pi_{User\_ID, Eta\_ID} (D \bowtie_{Des\_ID=Com\_ID \wedge Score>3 \wedge User\_ID \neq 'Brenda'} CO) \\ \text{AllBrendaLike} &\leftarrow \pi_{Eta\_ID} (D \bowtie_{Des\_ID=Com\_ID \wedge Score>3 \wedge User\_ID='Brenda'} CO) \\ \text{UserMatch} &\leftarrow \text{AllUserLike} / \text{AllBrendaLike} \\ \text{LikedEtab} &\leftarrow D \bowtie_{Des\_ID=Com\_ID \wedge Score>3} CO \\ \text{EtabAndUserMatch} &\leftarrow \text{LikedEtab} \bowtie_{User\_ID=User\_ID} \text{EtabAndUserMatch} \\ \text{Result} &\leftarrow \pi_{Eta\_ID} (\text{EtabAndUserMatch}) \end{aligned}$$
- R3 :
 
$$\begin{aligned} \text{AllEtab} &\leftarrow \pi_{Eta\_ID} (E) \\ \text{COM} &\leftarrow D \bowtie_{Des\_ID=Com\_ID} CO \\ \text{More1Com} &\leftarrow \sigma_{User\_ID \neq User\_ID_2 \vee Creation\_Date \neq Creation\_Date_2} \left( \pi_{User\_ID, Eta\_ID, Creation\_date} (COM) \right. \\ &\quad \left. * \alpha_{User\_ID:User\_ID_2, Creation\_Date:Creation\_Date_2} (\pi_{User\_ID, Eta\_ID, Creation\_date} (COM)) \right) \\ \text{EtabMore1Com} &\leftarrow \pi_{Eta\_ID} (\text{More1Com}) \\ \text{Result} &\leftarrow \text{AllEtab} - \text{EtabMore1Com} \end{aligned}$$
- R4 :
 
$$\begin{aligned} \text{AllCom} &\leftarrow \pi_{User\_ID, Eta\_ID} (D) \\ \text{AllEtab} &\leftarrow \pi_{Admin, Eta\_ID} (E) \\ \text{AdminNoComment} &\leftarrow \text{AllEtab} - \text{AllCom} \\ \text{Result} &\leftarrow \pi_{User\_ID} (\text{AdminNoComment}) \end{aligned}$$



## 6.2 SQL

Par manque de temps, toutes les requêtes n'ont pas pû être traduites en SQL.

-R3:

```
(SELECT eta_id
FROM Descriptions, Commentaires
WHERE com_id = des_id
GROUP BY eta_id
HAVING COUNT(com_id) < 2)
UNION
(SELECT eta_id
FROM etablissements E
WHERE NOT EXISTS (SELECT *
                  FROM Descriptions D, Commentaires
                  WHERE com_id = des_id AND E.eta_id = D.eta_id))
```

-R5:

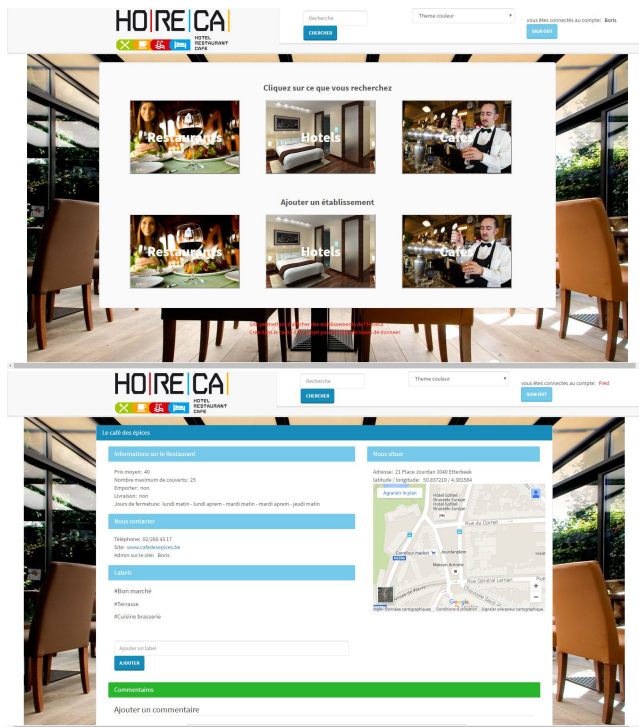
```
SELECT eta_id, AVG(score) AS note
FROM Descriptions D, Commentaires C
WHERE D.des_id = C.com_id
GROUP BY eta_id
HAVING COUNT(eta_id) > 2
ORDER BY note DESC
```

-R6:

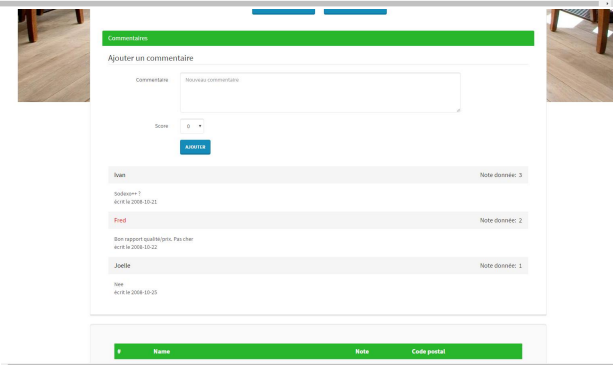
```
SELECT label
FROM (SELECT DISTINCT eta_id, label
      FROM Labels, Descriptions
      WHERE des_id = lab_id) L
GROUP BY label
HAVING COUNT(label) > 4
```

## 7 Démonstration de l'application

Voici un scénario de démonstration de l'application. Considérant que l'utilisateur s'est connecté en tant que administrateur. Celui-ci est amené sur la page de garde où il peut soit créer un nouvel établissement, soit consulter les établissements déjà existant. Après avoir choisi quel type d'établissement il souhaite consulter, il peut choisir quel établissement il veut voir. En cliquant sur le nom de celui-ci, il peut consulter sa fiche détaillée, ses commentaires et ses labels.



#	Nom	Note	Code postal
1	Au Grand Brasseur	5	1050
2	Bistro	3,3	1050
3	Chez Theo Soudanville L.L.B	0	1050
4	Comme Chez Soi	5	1050
5	Grain de sel	3,2	1050
6	Grolier et Dubois	3,4	1050
7	Indochine	4,7	1050
8	L'horloge du sud	3,5	1050
9	La Boutique	2	1050
10	Le Jardin qu'on	3,7	1050
11	Le Café des églises	0	1040
12	Le Café de la Gare	3,5	1050
13	Maison à Paris	4,5	1050
14	Musée	4	1050
15	1	0	0
16	Hotel	0	5



## 8 Création de la base de données

Cette section donne le code SQL DDL de création de la base de données.

```
DROP DATABASE IF EXISTS horeca;
CREATE DATABASE horeca CHARACTER SET 'utf8';
use horeca;
```

```
CREATE TABLE Users (
  User_ID VARCHAR(30) NOT NULL UNIQUE,
  Email VARCHAR(40),
  PSWD VARCHAR(20),
  Creation_date DATE,
  Is_Admin BOOLEAN,
  PRIMARY KEY (User_ID)
);
```

```
CREATE TABLE Etablissements (
  Eta_ID VARCHAR(30) NOT NULL UNIQUE,
  AdRue VARCHAR(50),
  AdNumero SMALLINT UNSIGNED,
  AdCodePostal SMALLINT UNSIGNED,
  AdCity VARCHAR(30),
  Longitude DOUBLE UNSIGNED,
  Latitude DOUBLE UNSIGNED,
```

```
Tel CHAR(12),
Site VARCHAR(50),
Creation_date DATE,
Admin VARCHAR(30),
PRIMARY KEY (Eta_ID),
FOREIGN KEY (Admin) REFERENCES Users(User_ID)
);
```

```
CREATE TABLE Restaurants (
Rest_ID VARCHAR(30) NOT NULL REFERENCES Etablissements (Eta_ID),
Prix SMALLINT UNSIGNED,
Couverts SMALLINT UNSIGNED,
Emporter BOOLEAN,
Livraison BOOLEAN,
Fermeture CHAR(27),
PRIMARY KEY (Rest_ID)
);
```

```
CREATE TABLE Cafes (
Cafe_ID VARCHAR(30) NOT NULL REFERENCES Etablissements (Eta_ID),
Fumeur BOOLEAN,
Restauration BOOLEAN,
PRIMARY KEY (Cafe_ID)
);
```

```
CREATE TABLE Hotels (
Hot_ID VARCHAR(30) NOT NULL REFERENCES Etablissements (Eta_ID),
Prix SMALLINT UNSIGNED,
Chambres SMALLINT UNSIGNED,
Etoiles SMALLINT UNSIGNED,
PRIMARY KEY (Hot_ID)
);
```

```
CREATE TABLE Descriptions (
Des_ID INTEGER UNSIGNED UNIQUE AUTO_INCREMENT,
User_ID VARCHAR(30),
Eta_ID VARCHAR(30),
PRIMARY KEY (Des_ID),
FOREIGN KEY (User_ID) REFERENCES Users(User_ID),
FOREIGN KEY (Eta_ID) REFERENCES Etablissements(Eta_ID)
);
```

```
CREATE TABLE Commentaires (  
  Com_ID INTEGER UNSIGNED REFERENCES Descriptions (Des_ID),  
  Com TEXT,  
  Creation_date DATE,  
  Score SMALLINT UNSIGNED,  
  PRIMARY KEY (Com_ID)  
);
```

```
CREATE TABLE Labels (  
  Lab_ID INTEGER UNSIGNED REFERENCES Descriptions (Des_ID),  
  Label VARCHAR(40),  
  PRIMARY KEY (Lab_ID)  
);
```

## 9 Conclusion

Ce projet a permis de se familiariser avec un gestionnaire de base de données (MySQL), un serveur web (Apache) et un langage de script (PHP).