


INFOH303 – Projet Internet Food DataBase

Hac Le, Théo Charlier, Albane Keraudren-Riguidel, Keyla Kamara
Mai 2024

A dark blue diagonal gradient bar that starts from the bottom left corner and extends towards the top right corner, covering the lower half of the slide.

Structure et méthode d'extraction des données



Framework, languages utilisés

FRONTEND:

- HTML
- CSS

BACKEND:

- Python avec le framework Flask
- Module: SQLAlchemy

Système de gestion de bases de données
relationnelles SQL:

- MySQL

Nos choix et hypothèses

Hypothèses de la phase 1:

- Un Client n'a pas de ClientID parce que ID de Actor est son primary key
- Moderator a un ModID pour le différencier du Client
- Un Owner ne peut pas modifier des Reviews pour son propre Restaurant
- Un Restaurant peut avoir plusieurs Menus
- 2 Actors peuvent avoir le même attribut AddressID

Les nouvelles hypothèses:

- Owner, Client et Moderator ont tous leurs propres ID qui dérivent de ID dans Actors

Implémentation de la base de données

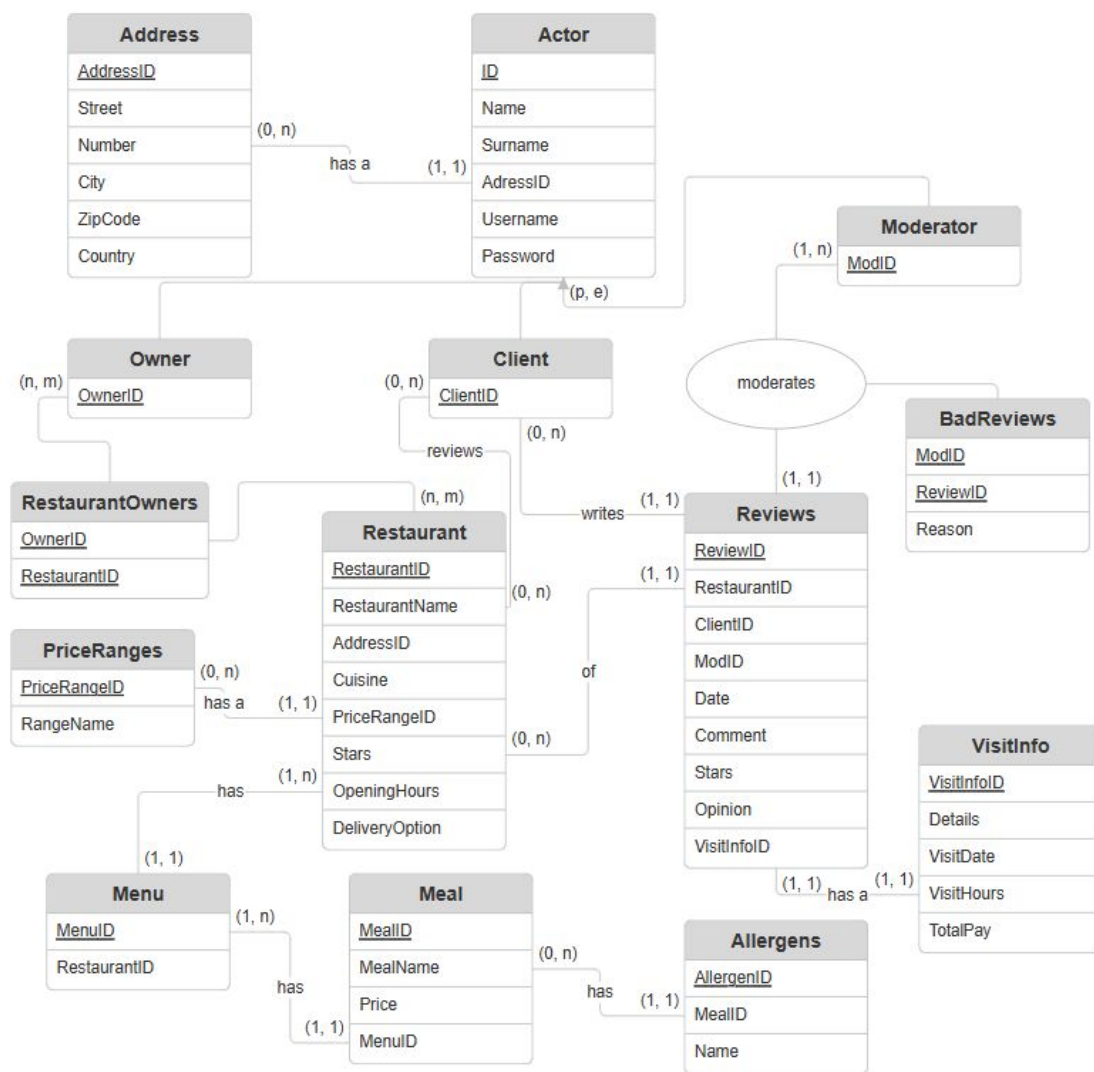
Création de la base de données:

```
1 DROP DATABASE IF EXISTS `internet_food_db`;
2 CREATE DATABASE `internet_food_db`;
3 USE `internet_food_db`;
4
5 CREATE TABLE Addresses (
6     AddressID INT NOT NULL AUTO_INCREMENT,
7     Street VARCHAR(255) NOT NULL,
8     StreetNumber INT NOT NULL,
9     ZipCode VARCHAR(255) NOT NULL,
10    City VARCHAR(255) NOT NULL,
11    Country VARCHAR(255) NOT NULL,
12    PRIMARY KEY (AddressID)
13 );
```

Connection de la base de données à l'application:

```
7 # Create the database connection
8 engine = create_engine('mysql+mysqldb://root:1234@127.0.0.1/internet_food_db')
9 conn = engine.connect()

16 statement_check_menu = text("SELECT MenuID FROM Menus WHERE RestaurantID = :restaurantID")
17 statement_insert_menu = text("INSERT INTO Menus (RestaurantID) VALUES (:restaurantID)")
```



Address

AddressID

Street

Number

City

ZipCode

Country

Actor

ID

Name

Surname

AddressID

Username

Password

(0, n)

Has a

(1, 1)

Owner

OwnerID

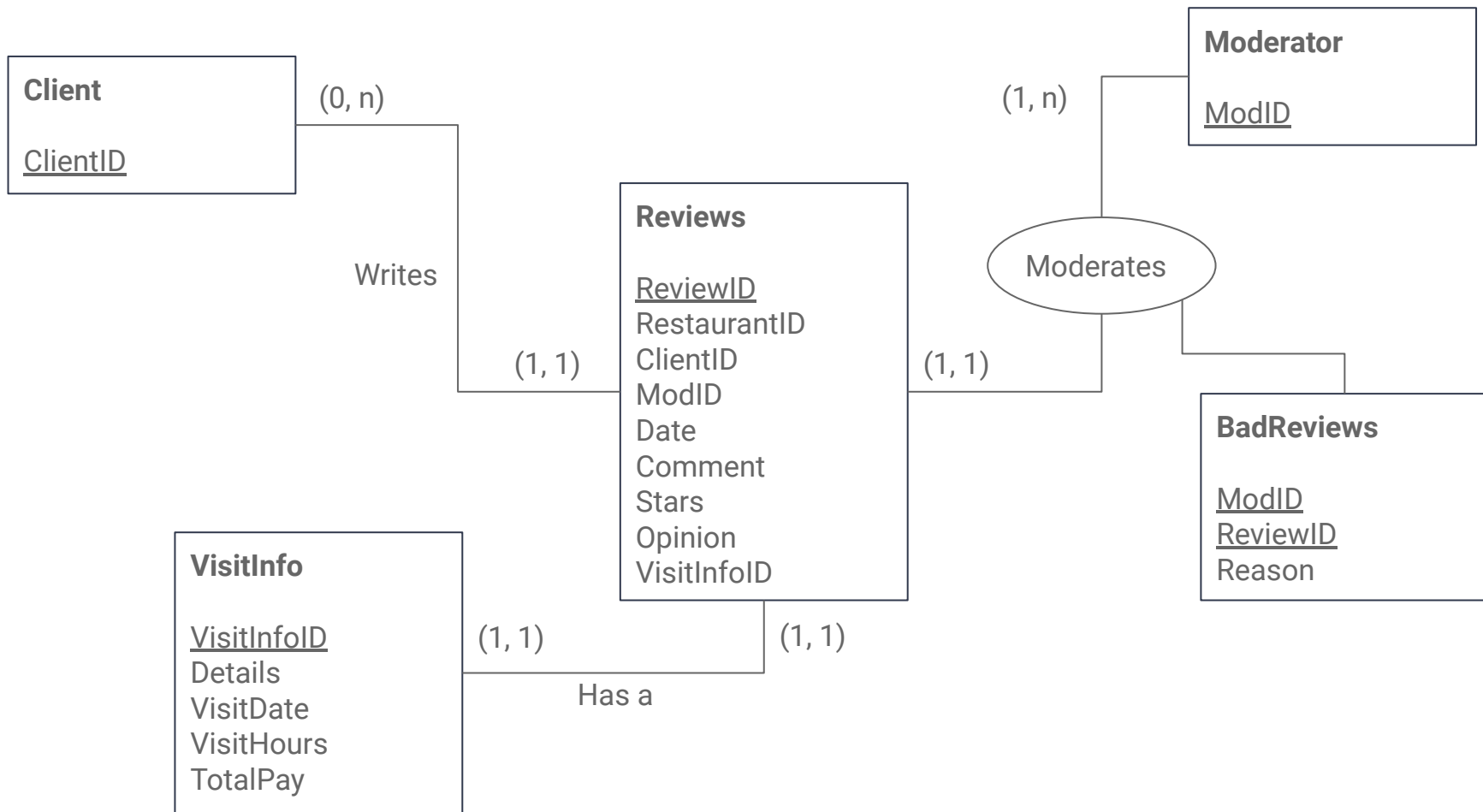
Client

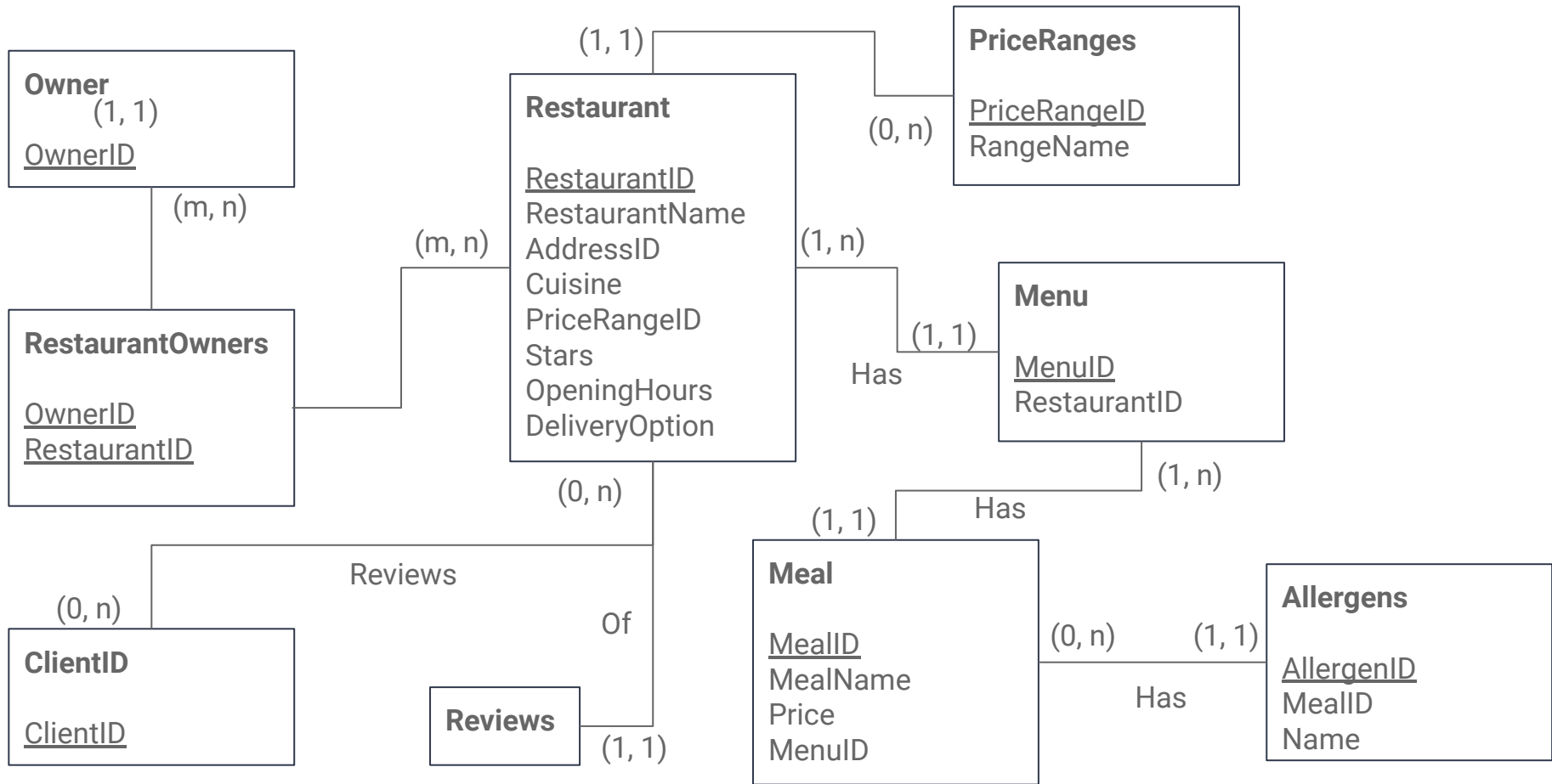
ClientID

Moderator

ModID

(p, e)





Requêtes demandées



Règles appliquées

- D'après le théorème de Codd: "Une requête est exprimable en calcul relationnel si et seulement si elle peut être exprimée par une expression de l'algèbre relationnelle."
- Le tri et la limite du nombre de résultats ne sont pas supportés par l'algèbre linéaire et le calcul relationnel.
- Nous n'utilisons pas la fonction agrégative COUNT() pour l'algèbre linéaire et le calcul relationnel.

1. Les restaurants ayant un avis moyen de 3 ou plus

SQL:

```
SELECT Restaurants.RestaurantName  
FROM Restaurants  
WHERE Restaurants.Stars >= 3;
```

Algèbre relationnelle:

$$\pi_{\text{RestaurantName}}(\sigma_{\text{Stars} \geq 3}(\text{Restaurants}))$$

Calcul tuple:

$$\{\text{name} \mid \text{Restaurants}(\text{RestaurantName} : \text{name}, \text{Stars} : \geq 3)\}$$

2. Le restaurant avec le plat le plus cher

SQL:

```
SELECT Restaurants.RestaurantName, Meals.Price  
FROM Restaurants  
JOIN Menus ON Restaurants.RestaurantID = Menus.RestaurantID  
JOIN Meals ON Menus.MenuID = Meals.MenuID  
WHERE Meals.Price = (SELECT MAX(Meals.Price) FROM Meals);
```

Algèbre relationnelle:

$$\pi_{\text{RestaurantName, Price}}(\sigma_{\text{Meals.Price} = (\sigma_{\text{max(Meals.Price)})(\text{Meals})}}(\text{Restaurants} \bowtie \text{Menus} \bowtie \text{Meals}))$$

Calcul tuple:

$$\{r.\text{RestaurantName}, \text{Meals.Price} \mid \text{Restaurants}(r) \wedge \exists m \in \text{Meals} (\exists n \in \text{Menus} (r.\text{RestaurantID} = n.\text{RestaurantID} \wedge n.\text{MenuID} = m.\text{MenuID} \wedge m.\text{Price} = (\forall m2 \in \text{Meals} (m2.\text{Price} \leq m.\text{Price}))))\}$$

3. Les 10 clients ayant consommé le plus de plats mexicains

```
SELECT Clients.Name  
FROM Clients  
JOIN Reviews ON Reviews.ClientID = Clients.ClientID  
JOIN Restaurants ON Reviews.RestaurantID =  
Restaurants.RestaurantID  
JOIN Menus ON Restaurants.RestaurantID =  
Menus.RestaurantID  
JOIN Meals ON Menus.MenuID = Meals.MenuID  
WHERE Restaurants.Cuisine = "mexicain"  
GROUP BY Clients.ClientID  
ORDER BY count(distinct Meals.MealID) desc  
LIMIT 10;
```

4. Le restaurant non-asiatique proposant le plus de plats qui sont généralement proposés dans des restaurants asiatiques

```
SELECT r1.RestaurantName, Meals.MealName
FROM Restaurants r1
JOIN Menus ON r1.RestaurantID = Menus.RestaurantID
JOIN Meals ON Menus.MenuID = Meals.MenuID
JOIN Restaurants r2 ON r2.Cuisine = "asiatique"
WHERE not r1.Cuisine = "asiatique"
GROUP BY r1.RestaurantName, Meals.MealName
ORDER BY count(distinct Meals.MealID) desc
LIMIT 1;
```

5. Le code postal de la ville dans laquelle les restaurants sont les moins bien notés en moyenne

SQL:

```
SELECT Addresses.ZipCode, Restaurants.Stars
FROM Addresses
JOIN Restaurants ON Restaurants.AddressID = Addresses.AddressID
GROUP BY Addresses.ZipCode, Restaurants.Stars
ORDER BY AVG(DISTINCT Restaurants.Stars) ASC
LIMIT 1;
```

Algèbre relationnelle:

$\pi_{\text{Addresses.ZipCode, Restaurants.Stars}}(\sigma_{\text{Restaurants.Stars}=\min(\text{Restaurants.Stars})}(\text{Addresses} \bowtie \text{Restaurants}))$

Calcul tuple:

$\{a.\text{ZipCode}, r.\text{Stars} \mid \text{Addresses}(a) \wedge \exists r \in \text{Restaurants} (r.\text{AddressID} = a.\text{AddressID} \wedge r.\text{Price} = (\forall r2 \in \text{Restaurants} (r.\text{Stars} \leq r2.\text{Stars})))\}$

6. Pour chaque tranche de score moyen de restaurant, le type de nourriture le plus représenté

SQL:

```
SELECT CASE
    WHEN AVG(Stars) >= 0 AND AVG(Stars) <= 1 THEN "1/5"
    WHEN AVG(Stars) >= 1 AND AVG(Stars) <= 2 THEN "2/5"
    WHEN AVG(Stars) >= 2 AND AVG(Stars) <= 3 THEN "3/5"
    WHEN AVG(Stars) >= 3 AND AVG(Stars) <= 4 THEN "4/5"
    WHEN AVG(Stars) >= 4 AND AVG(Stars) <= 5 THEN "5/5"
    END
    AND Restaurants.Cuisine
FROM Restaurants
GROUP BY Cuisine
ORDER BY Rating_Category ASC;
```

Démonstration

Ressources:

Conversion du SQL en Algèbre Relationnelle et Calcul Relationnel:

[sql2alg_eng.pdf \(ulb.ac.be\)](#)

<http://www.cpgebank.com/public/documents/1475147354434.pdf>

Implémentation de la base de données en python et création de l'application web:

<https://codeshack.io/login-system-python-flask-mysql/>

<https://youtu.be/yBDHkveJUf4?si=9yyUxmhs3Sx2RYt2>

https://youtu.be/Z1RJmh_OqeA?si=O1ADurE4b9jEGglO

<https://www.w3schools.com/css/>

<https://www.w3schools.com/html/default.asp>