

Tifosi — Base de données MySQL

Ce projet a été réalisé dans le cadre d'un devoir de formation Développeur Web. Il consiste à concevoir, implémenter et tester une base de données relationnelle pour le site du restaurant *Tifosi*.

L'objectif est de traduire un modèle conceptuel de données (MCD) fourni en un schéma relationnel MySQL fonctionnel, respectant l'intégrité référentielle, puis de peupler la base avec des données de test et de vérifier son bon fonctionnement à l'aide de requêtes SQL.

Objectifs pédagogiques

- Concevoir un schéma de base de données relationnelle à partir d'un MCD
- Mettre en place les relations et contraintes d'intégrité
- Peupler la base avec des données de test
- Vérifier le bon fonctionnement via des requêtes SQL

Environnement de travail

- Visual Studio Code
- XAMPP (MySQL + phpMyAdmin)
- Git & GitHub

Modèle conceptuel de données (MCD)

Le modèle conceptuel de données fourni décrit les entités principales du restaurant *Tifosi* ainsi que leurs relations.

Les entités principales sont :

- **Focaccia**
- **Ingrédient**
- **Boisson**
- **Marque**
- **Menu**
- **Client**

Certaines relations du MCD sont de type plusieurs-à-plusieurs (N–N). Elles ont été implémentées à l'aide de tables d'association afin de respecter les règles de la modélisation relationnelle.

Relations et choix de modélisation

Focaccia — Ingrédient (relation N–N)

Une focaccia peut contenir plusieurs ingrédients, et un ingrédient peut être utilisé dans plusieurs focaccias.

Cette relation plusieurs-à-plusieurs a été implémentée à l'aide de la table d'association **focaccia_ingredient**, qui porte en plus l'attribut **quantite**, représentant la quantité d'un ingrédient dans une focaccia.

Boisson — Marque (relation N–1)

Chaque boisson est associée à une seule marque, tandis qu'une marque peut être associée à plusieurs boissons.

Cette relation a été implémentée à l'aide d'une clé étrangère `marque_id` dans la table **boisson**.

Menu — Focaccia (relation 1–N)

Un menu peut contenir plusieurs focaccias, tandis qu'une focaccia peut être associée à un menu ou exister indépendamment.

Cette relation est implémentée par une clé étrangère `menu_id` dans la table **focaccia**.

Menu — Boisson (relation N–N)

Un menu peut contenir plusieurs boissons, et une boisson peut appartenir à plusieurs menus.

Cette relation est implémentée par la table d'association **menu_boisson**.

Client — Menu (relation N–N)

Un client peut acheter plusieurs menus, et un menu peut être acheté par plusieurs clients.

Cette relation est implémentée par la table **achat**, qui porte l'attribut `date_achat`.

Structure du projet

Le dépôt est organisé de manière à séparer clairement les scripts SQL, les données de test et les documents de validation.

```
Devoir_BD_site_Tifosi_MySQL/
├── data/
│   └── fichiers sources (.xlsx)
├── sql/
│   ├── 00_create_database.sql
│   ├── 01_user_tifosi.sql
│   ├── 02_schema.sql
│   ├── 03_seed.sql
│   └── 04_tests.sql
├── tests/
│   └── preuves des requêtes et résultats commentés
└── README.md
```

Chaque script SQL a un rôle précis et doit être exécuté dans l'ordre indiqué par sa numérotation.

Installation et exécution de la base de données

L'installation de la base de données se fait en exécutant les scripts SQL dans l'ordre suivant à l'aide de phpMyAdmin (compte administrateur MySQL).

Étape 1 — Création de la base de données

Exécuter le script :

- `00_create_database.sql`

Ce script supprime la base *tifosi* si elle existe déjà, puis la recrée.

Étape 2 — Création de l'utilisateur MySQL

Exécuter le script :

- `01_user_tifosi.sql`

Ce script crée l'utilisateur MySQL **tifosi** et lui attribue les droits nécessaires sur la base de données.

Étape 3 — Création du schéma

Exécuter le script :

- `02_schema.sql`

Ce script crée l'ensemble des tables, des clés primaires et des clés étrangères conformément au modèle relationnel.

Étape 4 — Peuplement des données

Exécuter le script :

- `03_seed.sql`

Ce script insère les données de test issues des fichiers fournis.

Étape 5 — Exécution des tests SQL

Exécuter le script :

- `04_tests.sql`

Ce script contient les requêtes permettant de vérifier le bon fonctionnement de la base de données.



Tests SQL et validation

Afin de vérifier le bon fonctionnement de la base de données, une série de requêtes SQL a été réalisée.

Ces requêtes permettent notamment de :

- vérifier les relations entre les tables,
- contrôler la cohérence des données,
- valider les jointures et agrégations.

Le script `04_tests.sql` regroupe l'ensemble des requêtes demandées dans le cadre du devoir.

Les résultats obtenus ont été exportés et commentés dans le dossier `tests/`, sous forme de fichiers CSV et PDF, afin de fournir des preuves concrètes des exécutions réalisées.

Choix techniques et remarques

- Les clés primaires sont de type **BIGINT UNSIGNED** afin d'assurer une bonne évolutivité de la base de données.
- Les relations entre les tables sont sécurisées par des clés étrangères avec des règles adaptées (**CASCADE**, **RESTRICT**, **SET NULL**).
- Les relations plusieurs-à-plusieurs (N-N) ont été implémentées à l'aide de tables d'association conformes aux bonnes pratiques.
- Les scripts SQL sont divisés en plusieurs fichiers selon leur rôle (création de la base, schéma, peuplement, tests) et numérotés afin de garantir un ordre d'exécution clair et reproductible.
- Des données de test réalistes ont été utilisées afin de permettre des vérifications pertinentes via les requêtes SQL.

Évolutions possibles

Dans le cadre d'une application plus complète, plusieurs évolutions pourraient être envisagées :

- Mise en place d'un système d'utilisateurs applicatifs (clients, administrateurs, employés) avec gestion des rôles et des droits.
- Ajout de tables complémentaires pour gérer le cycle de commande (commandes, paiements, livraisons).
- Création de vues SQL afin de simplifier certaines requêtes complexes et faciliter leur réutilisation.
- Automatisation de l'installation de la base de données à l'aide d'un script d'exécution unique (approche étudiée à titre personnel).
- Renforcement des contraintes et des index afin d'améliorer la cohérence et les performances de la base de données.

Note personnelle

À titre d'exercice complémentaire et personnel, le schéma de la base de données sera également reproduit à l'aide de **MySQL Workbench**, dans le but de comparer la modélisation visuelle avec l'implémentation SQL réalisée manuellement.

Cette démarche n'entre pas dans le périmètre du devoir demandé, mais vise uniquement à approfondir la compréhension des concepts de modélisation relationnelle.