Projet « RaPizz »

Richard HO, Ludovic REGNIER et Tom RAFNER

**Table des matières**

1. Introduction et hypothèses
2. Modèle conceptuel (Entité‐Association)
3. Passage au modèle relationnel
4. Script SQL : création des tables
5. Script SQL : jeu d’essai (insertions)
6. Requêtes demandées
7. Programmation : logique SQL (procédures, triggers)
8. Application Java Swing (structure, exemples de code)
9. Instructions de build & de test
10. Pistes d’évolutions

**1. Introduction & hypothèses**

* **Unité d’opération** : le cahier des charges précise qu’« on considère que l’opération de base à modéliser est la *vente d’une unique pizza* ». Une *livraison* = 1 pizza.
* **Trois tailles** : naine (–1/3 du prix), humaine (prix de base), ogresse (+1/3 du prix).
* **Système prépayé** : le solde du client est débité à la création de la livraison, sauf si la pizza est gratuite.
* **Gratuités** :
  + **Fidélité** : une pizza gratuite est offerte toutes les 10 achetées.
  + **Retard > 30 min** : la pizza est gratuite et le client est remboursé.
* **Technologies cibles** : PostgreSQL 16 (SQL/PSM standard, PL/pgSQL), Java 21, Maven, JDBC, Swing.

**2 . Modèle conceptuel E‑A**

Une image contenant texte, capture d’écran, diagramme, Police

Le contenu généré par l’IA peut être incorrect.

Entités et attributs

|  |  |  |
| --- | --- | --- |
| **Entité** | **PK** | **Attributs supplémentaires** |
| **CLIENT** | client\_id | nom, prenom, adresse, email, tel, solde, pizzas\_payées\_compteur |
| **PIZZA** | pizza\_id | nom, prix\_base |
| **INGREDIENT** | ingredient\_id | libelle |
| **LIVREUR** | livreur\_id | nom, prenom |
| **VEHICULE** | vehicule\_id | immatriculation (UNIQUE), type ('AUTO', 'MOTO'), modele |
| **TAILLE** | taille\_code | libelle, facteur\_prix |
| **LIVRAISON** | livraison\_id | client\_id FK, pizza\_id FK, taille\_code FK, livreur\_id FK, vehicule\_id FK, date\_commande, date\_livraison, prix\_facture, gratuite BOOL, raison\_gratuite |

Associations détaillées

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Association | Entités reliées | Cardinalité | Clé étrangère située dans | Attributs portés par la relation |
| passe | CLIENT — LIVRAISON | CLIENT (1,n) — LIVRAISON (1,1) | LIVRAISON.client\_id |  |
| concerne | LIVRAISON — PIZZA | LIVRAISON (1,1) — PIZZA (1,1) | LIVRAISON.pizza\_id |  |
| compose | PIZZA — INGREDIENT | PIZZA (1,n) — INGREDIENT (1,n) | Table associative PIZZA\_INGREDIENT |  |
| utilise | LIVRAISON — LIVREUR | LIVRAISON (1,1) — LIVREUR (1,n) | LIVRAISON.livreur\_id |  |
| roule | LIVREUR — VEHICULE | LIVREUR (1,n) — VEHICULE (1,n) | Aucun (association séparée) |  |
| est\_de\_taille | LIVRAISON — TAILLE | LIVRAISON (1,1) — TAILLE (1,n) | LIVRAISON.taille\_code |  |

**3 . Modèle relationnel**

**4 . Script SQL : création des tables (PostgreSQL‑16)**

Une image contenant texte, capture d’écran, logiciel

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, capture d’écran, logiciel

Le contenu généré par l’IA peut être incorrect.

**5 . Script SQL : jeu d’essai minimal**