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 |

Entité Client

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Attribut** | |  |  |  | | --- | |  | | |  | | --- | | **Type SQL** |  |  | | --- | |  | | |  | | --- | | **Utilité** |  |  | | --- | |  | | **Remarques** |
| client\_id   |  | | --- | |  | | SERIAL (PK) | |  | | --- | | Identifiant unique du client |  |  | | --- | |  | | |  | | --- | | Généré automatiquement |  |  | | --- | |  | |
| nom | |  | | --- | | VARCHAR(50) |  |  | | --- | |  | | |  | | --- | | Nom de famille du client |  |  | | --- | |  | | |  | | --- | | Obligatoire |  |  | | --- | |  | |
| prenom | |  | | --- | | VARCHAR(50) |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | Prénom du client |  |  | | --- | |  | |  |  | | --- | |  | | Obligatoire |
| adresse | TEXT | |  | | --- | | Adresse de livraison |  |  | | --- | |  | | Utilisée pour les livraisons |
| email | VARCHAR(120) UNIQUE | Identifiant de contact | Peut servir à se connecter à un compte |
| tel | VARCHAR(20) | Téléphone du client |  |
| solde | NUMERIC(8,2) | Montant en euros sur le compte | Décrémenté à chaque commande |
| pizzas\_payees\_compteur | SMALLINT | Nombre de pizzas payées | Sert pour la gratuité fidélité (toutes les 10 pizzas) |

Entité Pizza

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribut** | **Type SQL** | **Utilité** | **Remarques** |
| pizza\_id | SERIAL (PK) | Identifiant unique de la pizza |  |
| nom | VARCHAR(60) UNIQUE | Nom de la pizza | Exemple : "Reine", "Margherita" |
| prix\_base | NUMERIC(5,2) | Prix de base pour taille « humaine » | Les tailles ajustent ce prix avec un facteur multiplicatif |

Entité Ingrédient

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribut** | **Type SQL** | **Utilité** | **Remarques** |
| ingredient\_id | SERIAL (PK) | Identifiant unique de l’ingrédient |  |
| libelle | VARCHAR(60) UNIQUE | Nom de l’ingrédient |  |

Entité Taille

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribut** | **Type SQL** | **Utilité** | **Remarques** |
| taille\_code | CHAR(1) (PK) | Code court : N, H, O |  |
| libelle | VARCHAR(20) | Nom de la taille | "Naine", "Humaine", "Ogresse" |
| facteur\_prix | NUMERIC(4,3) | Multiplie le prix de base | Ex : 0.666, 1.000, 1.333 |

Entité Livreur

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribut** | **Type SQL** | **Utilité** | **Remarques** |
| livreur\_id | SERIAL (PK) | Identifiant unique |  |
| nom | VARCHAR(50) | Nom du livreur |  |
| prenom | VARCHAR(50) | Prénom du livreur |  |

Entité Vehicule

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribut** | **Type SQL** | **Utilité** | **Remarques** |
| vehicule\_id | SERIAL (PK) | Identifiant unique |  |
| immatriculation | VARCHAR(15) UNIQUE | Permet d’identifier le véhicule |  |
| type | VARCHAR(10) | Type de véhicule | Doit être 'AUTO' ou 'MOTO' |
| modele | VARCHAR(60) | Modèle du véhicule | Exemple : "Yamaha 125", "Fiat Panda" |

Entité Livraison

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribut** | **Type SQL** | **Utilité** | **Remarques** |
| livraison\_id | SERIAL (PK) | Identifiant de la livraison |  |
| client\_id | INT FK → CLIENT | Client concerné |  |
| pizza\_id | INT FK → PIZZA | Pizza livrée |  |
| taille\_code | CHAR(1) FK → TAILLE | Taille de la pizza |  |
| livreur\_id | INT FK → LIVREUR | Livreur assigné |  |
| vehicule\_id | INT FK → VEHICULE | Véhicule utilisé |  |
| date\_commande | TIMESTAMP | Quand la commande a été passée | Valeur par défaut = now() |
| date\_livraison | TIMESTAMP | Quand elle a été livrée | Sert à détecter les retards |
| prix\_facture | NUMERIC(6,2) | Prix réellement payé |  |
| gratuite | BOOLEAN |  |  |
| raison\_gratuite | VARCHAR(20) | Soit 'FIDELITE', 'RETARD' ou NULL |  |

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**

Table CLIENT (

client\_id PK,

nom,

prenom,

adresse,

email,

tel,

solde,

pizzas\_payees\_compteur

)

Table PIZZA (

pizza\_id PK,

nom,

prix\_base

)

Table INGREDIENT (

ingredient\_id PK,

libelle

)

Table TAILLE (

taille\_code PK,

libelle,

facteur\_prix

)

Table LIVREUR (

livreur\_id PK,

nom,

prenom

)

Table VEHICULE (

vehicule\_id PK,

immatriculation,

type,

modele

)

Table LIVRAISON (

livraison\_id PK,

client\_id FK → CLIENT,

pizza\_id FK → PIZZA,

taille\_code FK → TAILLE,

livreur\_id FK → LIVREUR,

vehicule\_id FK → VEHICULE,

date\_commande,

date\_livraison,

prix\_facture,

gratuite,

raison\_gratuite

)

Table PIZZA\_INGREDIENT (

pizza\_id FK → PIZZA,

ingredient\_id FK → INGREDIENT,

PRIMARY KEY (pizza\_id, ingredient\_id)

)

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

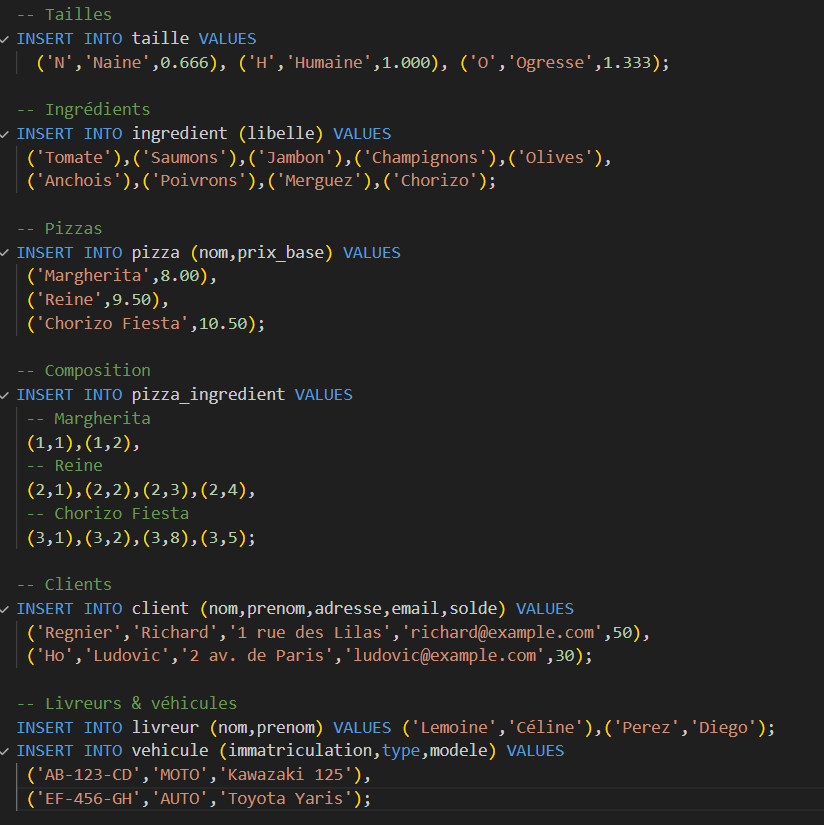
Une image contenant texte, capture d’écran, logiciel

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**



**6 . Requêtes demandées**

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.

**7 . Logique SQL : procédures & triggers**

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.