

RAPPORT PROJET BASE DE DONNÉES

Gestion d'une entreprise de pizzas à domicile



Alex FOULON, Thomas LE MAGNY,
Maxime BARDATet Erwan GAUTIER

25/06/2023
E3 FI - groupe 2

Tables des matières

Introduction.....	2
Conception.....	3
Vue administrateur.....	4
Vue livreur.....	5
Vue client.....	5
Fonctionnalités.....	6
Statistiques.....	7
Questions diverses.....	8
Intégrité des données.....	8
Conclusion.....	11
Guide utilisateur.....	12
Configuration de VS Code.....	12
Fichier de configuration.....	12
Installation des modules.....	13
Installation de la base de données.....	14

Introduction

Ce rapport présente le projet de gestion d'une entreprise de fabrication et de livraison de pizzas à domicile, réalisé dans le cadre du cours de Base de données à l'ESIEE Paris. Pour ce projet, nous avons utilisé les technologies JDBC, Java, JavaFX, MySQL et Looping pour concevoir et mettre en œuvre une solution de gestion complète pour l'entreprise RaPizz.

L'objectif de ce projet était de modéliser et de développer une base de données capable de gérer efficacement l'activité quotidienne de vente et de livraison de pizzas.

La conception de la base de données a suivi une approche en deux étapes : l'élaboration du modèle entité-association, suivi de la transformation vers le modèle relationnel. Nous avons également écrit des scripts pour la création des tables et l'insertion des données dans la base de données afin de la rendre opérationnelle.

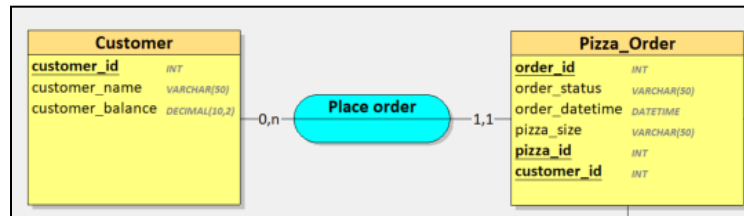
En utilisant JDBC et Java, nous avons développé une interface utilisateur permettant de passer des commandes, de gérer les comptes clients, de suivre les livraisons en temps réel, et d'accéder à des statistiques et des rapports sur l'activité de l'entreprise.

Conception

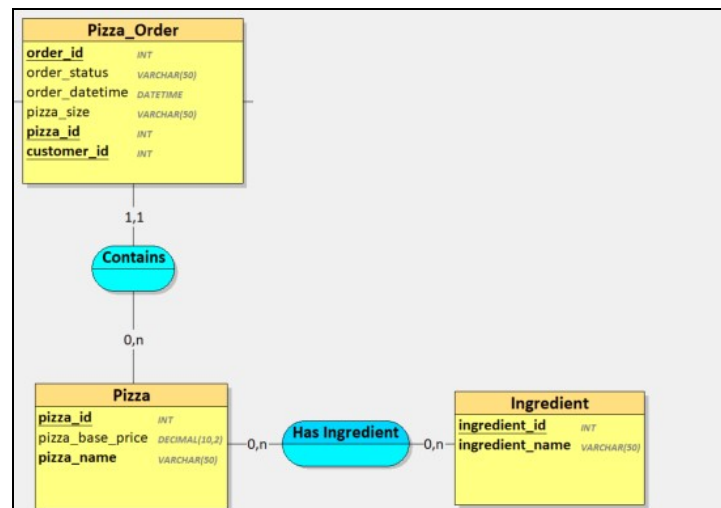
Pour commencer, nous avons dû modéliser l'architecture de la base de données avec Looping. Looping est un logiciel de modélisation de base de données qui permet d'organiser et de structurer les données en diagrammes et de générer les tables SQL en un seul clic.

Voici la conception que de notre base de donnée :

Nous avons un client, ce dernier peut passer de 0 à n commandes, et une commande appartient à un unique client.



Une commande contient une unique pizza, qui contient elle-même plusieurs ingrédients.



La commande peut avoir une livraison, qui est effectuée par un unique livreur auquel on assigne un unique véhicule. Elle dispose également d'un statut pour suivre son avancée.



Vue administrateur

La vue administrateur offre des fonctionnalités et des outils avancés pour gérer, surveiller et contrôler efficacement le système.

Fonctionnalité :

- La gestion des livreurs
 - Tableau des livreurs avec l'id, le nom et le véhicule
 - Éditer leurs noms
 - Changer leurs véhicules
 - Supprimer le livreur
 - Ajouter un livreur
- La gestion des pizzas
 - Tableau des pizzas avec le nom, le prix, les ingrédients
 - Commander une pizza
 - Supprimer une pizza
 - Modifier une pizza
 - Ajouter une pizza
- La gestion des clients
 - Tableau des clients avec l'id, le nom et le crédit
 - Recharger le compte
 - Éditer le nom
 - Supprimer le client
 - Ajouter un client
- La gestion des commandes
 - Tableau des commandes avec l'id, nom client, nom pizza, taille pizza, prix, status commande, status livraison, date
 - Mise à jour du statut de la commande
 - Annuler la commande
 - Assigner un livreur
- La gestion des livraisons
 - Tableau des livraisons avec l'id, commande id, livreur id, livreur nom, véhicule id, type véhicule, status, date de livraison
 - Mettre à jour le statut
 - Annuler la livraison
 - Assigner un livreur
- La gestion des véhicules
 - Tableau des véhicules avec l'id, type, model
 - Éditer le véhicule
 - Supprimer le véhicule
- Les statistiques

Vue livreur

La vue livreur offre aux livreurs les informations nécessaires pour organiser et effectuer les livraisons de manière efficace et optimisée.

Fonctionnalité :

- L'assignation des commandes
 - Tableau des commandes avec l'id, nom client, pizza taille, status, date commande
 - S'assigner une commande
- Leurs livraisons
 - Tableau des pizzas avec l'id, commande id, status, date
 - Mettre à jour la livraison
 - Annuler la livraison
- La gestion du profil
 - Changer le nom
 - Changer de véhicule

Vue client

La vue client permet aux clients d'accéder facilement au catalogue des pizzas, de passer des commandes, de gérer leur compte et de suivre l'état de leurs livraisons.

Fonctionnalité :

- Catalogue des pizzas
 - Tableau des pizzas avec le nom, prix, ingrédients
 - Commander la pizza
 - Choisir la taille
- Leurs commandes
 - Tableau des pizzas avec l'id, nom pizza, pizza taille, prix, status commande, status livraison, date commande, annulé
- La gestion du profil
 - Changer le nom
 - Recharger le compte

Fonctionnalités

- La 10^e pizza commandée est gratuite (affichage dans l'interface order Pizza du nombre de pizza restantes avant la pizza gratuite, ainsi que notification quand la pizza est gratuite au bout de 10 commandes)
- Éditeur de pizza pour l'administrateur, permettant de concevoir une pizza
- Bouton "refresh" présent sur chaque scène, afin d'actualiser l'écran manuellement
- Les livreurs peuvent choisir un des véhicules disponible
- Interface de début qui permet de choisir si on veut se connecter en admin, livreur ou client.
- L'administrateur à le contrôle de la base de données
- Affichage des statistiques pour l'administrateur
- Association image - pizza possible en fonction du nom (Nous ne l'avons pas fait car nous n'avons pas d'images pour chaque pizza mais il suffirait de modifier l'initiation de l'image dans le constructeur de la classe Java pizza)

Statistiques

Pour obtenir diverses statistiques sur l'activité de l'entreprise, nous avons utilisé des procédures stockées, en effet cela permet de réduire considérablement le trafic réseau entre le serveur et le client.

Listes des procédures stockées permettant de répondre aux statistiques demandés pour le projet :

- Identification du meilleur client, celui ayant le plus de commande 'COMPLETED'
 - GetDeliveriesDetailsOfDeliveryPerson(IN delivery_person_id INT)
- Identification du plus mauvais livreur, celui ayant le plus de livraison 'LATE' avec le véhicule utilisé
 - GetWorstDeliveryPerson()
- Identification du meilleur livreur, celui ayant le plus de livraison 'COMPLETED' avec le véhicule utilisé
 - GetBestDeliveryPerson()
- Identification de la pizza la plus vendue
 - GetMostOrderedPizza()
- Identification de l'ingrédient favori, celui étant le plus dans les pizzas commandées
 - GetMostPopularIngredient()

Questions diverses

Les méthodes suivantes nous permettent de répondre à ces questions via un simple appel de procédure. Certaines sont utilisées dans l'interface et d'autres non, bien qu'elles existent dans la base de données.

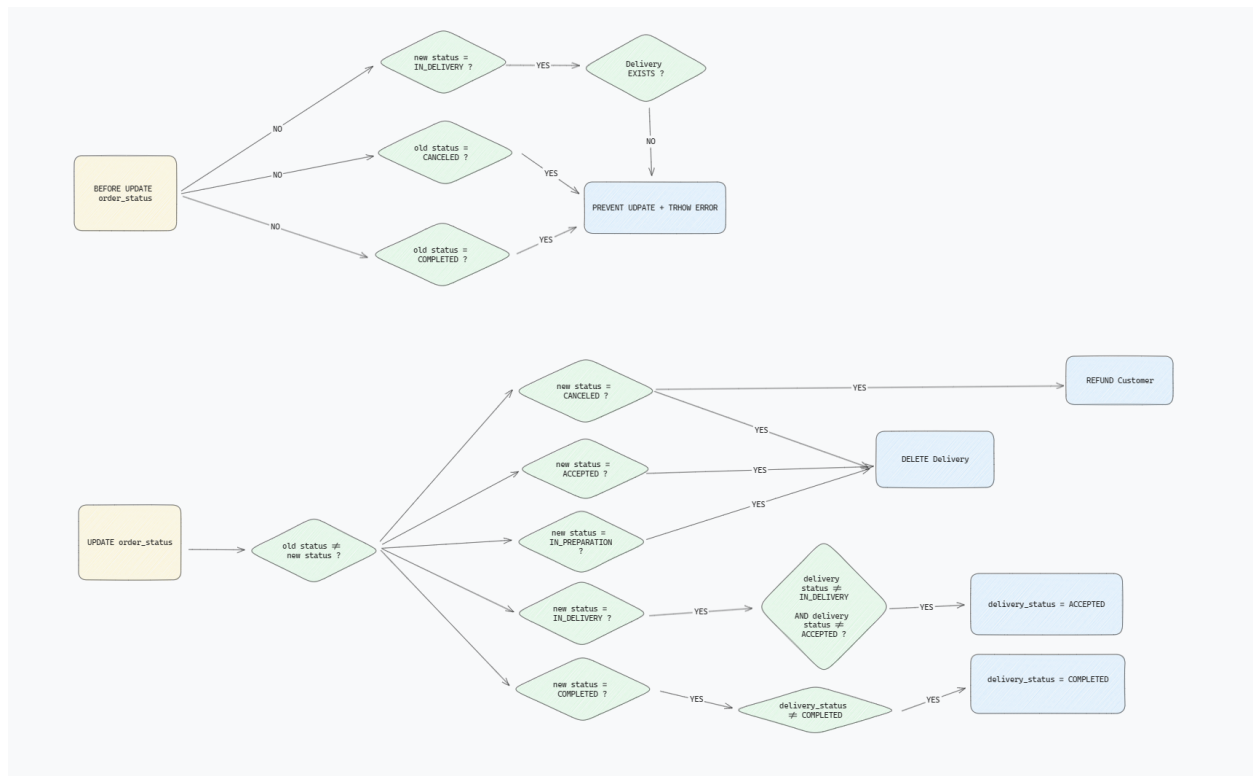
- Quels sont les véhicules n'ayant jamais servi ?
 - FindUnusedVehicles()
- Quel est le temps moyen de livraison d'une commande ?
 - CalculateAverageDeliveryTime()
- Quel est le prix moyen des commandes ?
 - CalculateAverageOrderPrice()
- Quel est le chiffre d'affaires total et mensuel ?
 - CalculateTotalRevenue()
 - CalculateMonthlyRevenue()
- Quels sont les clients ayant commandé plus que la moyenne ?
 - GetCustomersAboveAverage()
- Combien de commandes a passé chaque client ?
 - CalculateOrderCountByCustomer()
- Comment se répartissent les ventes en fonction de la taille des pizzas ?
 - GeneratesSalesDistributionByPizzaSize()

Intégrité des données

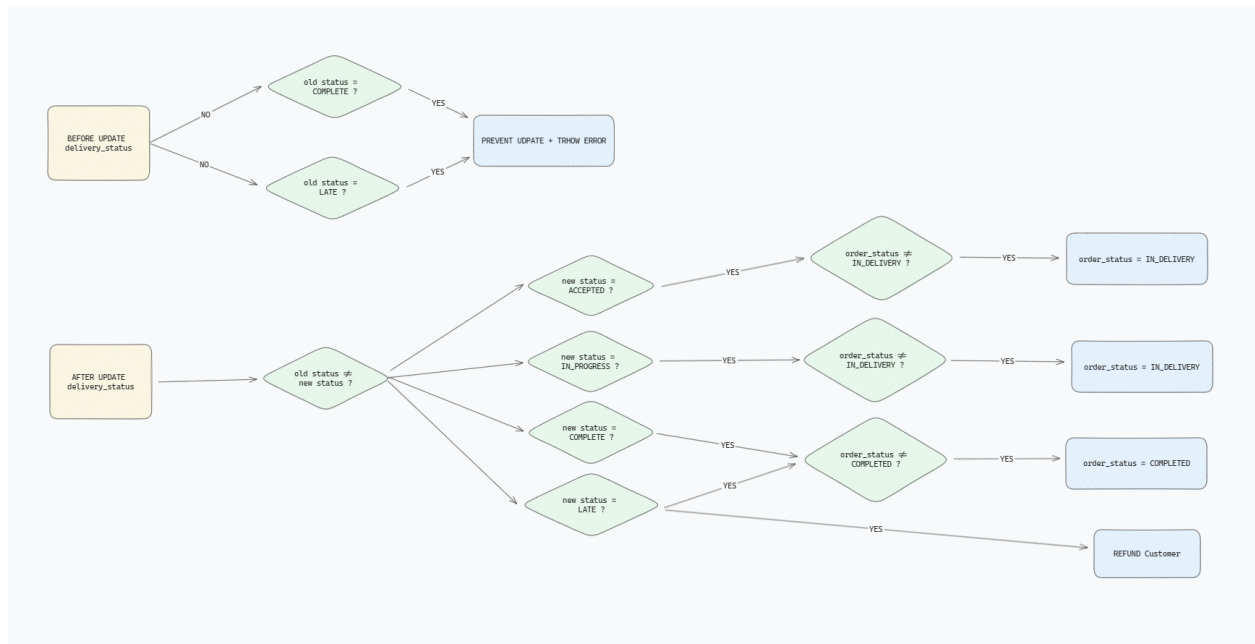
Pour s'assurer de l'intégrité des données lors des mises à jour des statuts des commandes et des livraisons, nous utilisons des procédures stockées pour toutes les modifications sur ceux-ci. De sorte, la cohérence entre le statut de la commande et celui de la livraison est quasi-infaillible (sauf insertion en dur via un UPDATE statement).

Au départ, nous avons conçu un système de trigger pour maintenir l'intégrité. En voici le diagramme de la première version du système :

Réaction au changement de statut dans Pizza Order:



Réaction au changement de statut dans Delivery :



Ce modèle a plus tard été renforcé pour prendre en compte davantage de paramètres (annulation / suppression de livraison) puis converti en une série de procédures stockées qui sont utilisées à chaque mise à jour de statut.

Conclusion

En conclusion, ce projet de gestion d'une base de données relationnelle, réalisé dans le cadre du cours de Base de données, nous a permis de concevoir une base de données et de mettre en place des fonctionnalités essentielles pour la gestion quotidienne de l'activité de l'entreprise.

Grâce à notre modèle entité-association et au passage au modèle relationnel, nous avons pu structurer efficacement les données liées aux produits, aux livreurs, aux clients et aux commandes de pizzas. L'utilisation de JavaFX nous a permis d'implémenter rapidement des interfaces utilisateur, JDBC et JAVA ont permis la connexion entre notre application et la base de données.

Nous avons mis en œuvre des fonctionnalités telles que la gestion des comptes clients, la préparation et la livraison des pizzas, le suivi des livreurs et des véhicules, ainsi que des statistiques sur l'activité de l'entreprise. Ces fonctionnalités ont permis d'automatiser plusieurs processus et d'améliorer l'efficacité de l'entreprise.

Malgré les défis rencontrés lors de la conception et de l'implémentation, nous avons réussi à atteindre nos objectifs et à livrer un système entièrement fonctionnel. Cependant, il reste encore des pistes d'amélioration, notamment en ce qui concerne l'optimisation des requêtes et l'ajout de fonctionnalités supplémentaires pour prendre en compte la gestion des commandes comprenant plusieurs pizzas. La partie architecturale de l'interface JavaFX mérite également une mise à jour, avec un découpage par onglet plutôt que par fenêtre.

Dans l'ensemble, ce projet nous a permis d'approfondir nos connaissances en bases de données et de mettre en pratique les concepts appris en cours tout en découvrant de nouvelles technologies telles que la bibliothèque JavaFX.

Guide utilisateur

Le guide suivant permet de configurer le projet pour un lancement via VS Code. Libre à vous de l'interpréter pour les autres IDE / Éditeurs.

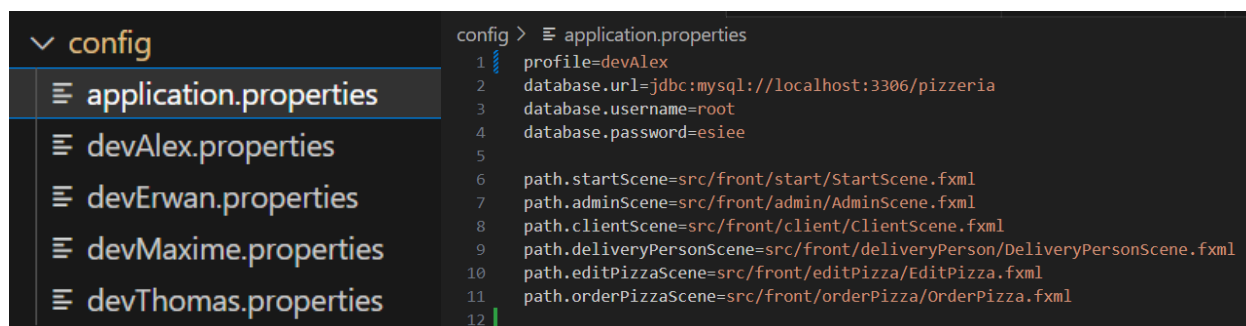
Configuration de VS Code

Pour une interaction fluide avec le code, il est recommandé d'installer l'extension "Extension Pack for Java".

Fichier de configuration

Les fichiers de configuration (.properties) ont été utilisés pour nous permettre d'accéder à des variables globales, définissant notamment des chemins vers des ressources ou des identifiants pour la base de données.

Nous avons également créé un système de profil, de sorte à ce que toute constante écrite dans un profil de développement écrase celle du fichier de configuration initial. Cela nous a été très utile pour y stocker les informations de connexion propres à chacune de nos bases de données. Pour changer de profil, il suffit d'inscrire le nom du fichier correspondant dans le champ "profil" de application.properties.



```
config > application.properties
1 profile=devAlex
2 database.url=jdbc:mysql://localhost:3306/pizzeria
3 database.username=root
4 database.password=esiee
5
6 path.startScene=src/front/start/StartScene.fxml
7 path.adminScene=src/front/admin/AdminScene.fxml
8 path.clientScene=src/front/client/ClientScene.fxml
9 path.deliveryPersonScene=src/front/deliveryPerson/DeliveryPersonScene.fxml
10 path.editPizzaScene=src/front/editPizza/EditPizza.fxml
11 path.orderPizzaScene=src/front/orderPizza/OrderPizza.fxml
12
```

Vous devez donc remplir le fichier application.properties comme il se doit pour que la connexion avec votre base de données se fasse correctement (url, username, password).

Installation des modules

Les modules de JavaFX doivent être référencés dans le fichier “.vscode/settings.json.”
Voici un exemple de configuration correcte. Les modules liés à JDBC et jackson se situent dans le dossier “Java/modules”. Ceux liés à JavaFX se trouvent dans “javafx-sdk-20.0.1/lib”. Vous devez adapter le chemin pour qu’il corresponde à celui de votre machine.

```
{
  "java.dependency.packagePresentation": "hierarchical",
  "java.project.sourcePaths": [
    ".",
    "modules",
    "scenes"
  ],
  "java.project.referencedLibraries": [
    "modules/Driver.jar",
    "modules/jackson-annotations-2.15.1.jar",
    "modules/jackson-core-2.15.1.jar",
    "modules/jackson-databind-2.15.1.jar",
    "c:\\Users\\foulo\\OneDrive\\Projets\\Projets ESIEE\\Projet-bdd-pizzeria\\javafx-sdk-20.0.1\\lib\\javafx.base.jar",
    "c:\\Users\\foulo\\OneDrive\\Projets\\Projets ESIEE\\Projet-bdd-pizzeria\\javafx-sdk-20.0.1\\lib\\javafx.controls.jar",
    "c:\\Users\\foulo\\OneDrive\\Projets\\Projets ESIEE\\Projet-bdd-pizzeria\\javafx-sdk-20.0.1\\lib\\javafx.fxml.jar",
    "c:\\Users\\foulo\\OneDrive\\Projets\\Projets ESIEE\\Projet-bdd-pizzeria\\javafx-sdk-20.0.1\\lib\\javafx.graphics.jar",
    "c:\\Users\\foulo\\OneDrive\\Projets\\Projets ESIEE\\Projet-bdd-pizzeria\\javafx-sdk-20.0.1\\lib\\javafx.media.jar",
    "c:\\Users\\foulo\\OneDrive\\Projets\\Projets ESIEE\\Projet-bdd-pizzeria\\javafx-sdk-20.0.1\\lib\\javafx.swing.jar",
    "c:\\Users\\foulo\\OneDrive\\Projets\\Projets ESIEE\\Projet-bdd-pizzeria\\javafx-sdk-20.0.1\\lib\\javafx.web.jar",
    "c:\\Users\\foulo\\OneDrive\\Projets\\Projets ESIEE\\Projet-bdd-pizzeria\\javafx-sdk-20.0.1\\lib\\javafx.swt.jar"
  ]
}
```

Il est également important d’ajouter la ligne “vmArgs” comme suit dans le fichier “vscode/launch.json”. Le chemin doit correspondre à celui de votre machine.

```
{
  // Use IntelliSense to learn about possible attributes.
  // Hover to view descriptions of existing attributes.
  // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
  "version": "0.2.0",
  "configurations": [
    {
      "type": "java",
      "name": "PizzeriaApp",
      "request": "launch",
      "mainClass": "src.PizzeriaApp",
      "projectName": "Java_fd8534a8",
      "vmArgs": "--module-path \"C:/Users/foulo/OneDrive/Projets/Projets ESIEE/Projet-bdd-pizzeria/javafx-sdk-20.0.1/lib\" --add-modules javafx.controls,javafx.fxml"
    },
    {
      "type": "java",
      "name": "Current File",
      "request": "launch",
      "mainClass": "${file}"
    }
  ]
}
```

Installation de la base de données

La base de données contient énormément de procédures stockées. Pour la mettre en place rapidement il suffit d'exécuter le script "SQL/SetupDatabase.sql" qui contient une copie de tous les autres fichiers SQL pour accélérer le processus.

Des données de test y sont déjà insérées. Vous n'avez plus qu'à lancer le programme et tester toutes les fonctionnalités.

Conseil: Lancez trois instances du programme et connectez-vous à toutes les interfaces simultanément pour voir les données réagir en temps réel. N'oubliez pas d'appuyer sur le bouton "refresh" quand nécessaire.