

SOLUTIONS OCPIZZA

SPECIFICATIONS TECHNIQUE



1. TABLE DES MATIERES

1. TABLE DES MATIERES	1
Versions	3
Introduction	4
Diagramme de Classes UML	5
Modèle Physique de Données (MPD)	8
Fichier SQL de la Base de Données	11

VERSIONS

Auteur	Date	Description	Version
Younes ouasmi	11/04/2024		1.0

INTRODUCTION

Le présent document de spécifications techniques a été élaboré dans le but de fournir une vue détaillée et structurée de la conception du système informatique pour le projet OC Pizza. Il s'agit d'un ensemble de documents clés qui guideront les étapes de développement et d'implémentation nécessaires à la réalisation d'une solution complète et fonctionnelle.

À l'intérieur de ce document, vous trouverez les composants suivants :

- 1. Diagramme de Classes UML:** Ce diagramme représente la structure du domaine fonctionnel de notre système. Il est le fruit d'une analyse rigoureuse des besoins opérationnels et fonctionnels d'OC Pizza, capturant les relations entre les différentes entités du domaine d'affaires.
- 2. Modèle Physique de Données (MPD) :** Le MPD illustre la conception détaillée de la base de données. Il décrit la façon dont les données seront structurées, stockées, et interconnectées, garantissant une performance optimale et une intégrité des données.
- 3. Fichier SQL de la Base de Données :** Fournissant un ensemble de scripts SQL, ce fichier est essentiel pour créer la base de données initiale. Il assure que l'infrastructure de données peut être facilement reproduite ou réinitialisée sur n'importe quel environnement de production nécessaire.

Ces documents sont destinés à être utilisés comme une référence pour le développement, l'assurance qualité et la gestion du projet, assurant ainsi une compréhension uniforme et une exécution cohérente du plan de développement. Ils sont conçus pour être évolutifs et s'adapteront au fur et à mesure que le projet se développe et que de nouveaux besoins sont identifiés.

L'objectif final est de mettre en place une infrastructure technique fiable et efficace qui soutiendra la croissance rapide et les opérations quotidiennes d'OC Pizza, tout en améliorant l'expérience utilisateur pour les clients et le personnel.

DIAGRAMME DE CLASSES UML

Cette section présente le Diagramme de Classes UML, un composant crucial des spécifications techniques pour le projet OC Pizza. Le diagramme sert de représentation visuelle du domaine fonctionnel et détaille la structure et les relations entre les différentes entités du système. Il a été méticuleusement conçu pour modéliser la complexité des processus métiers d'OC Pizza, permettant ainsi une compréhension claire des interactions et des dépendances au sein de l'application.

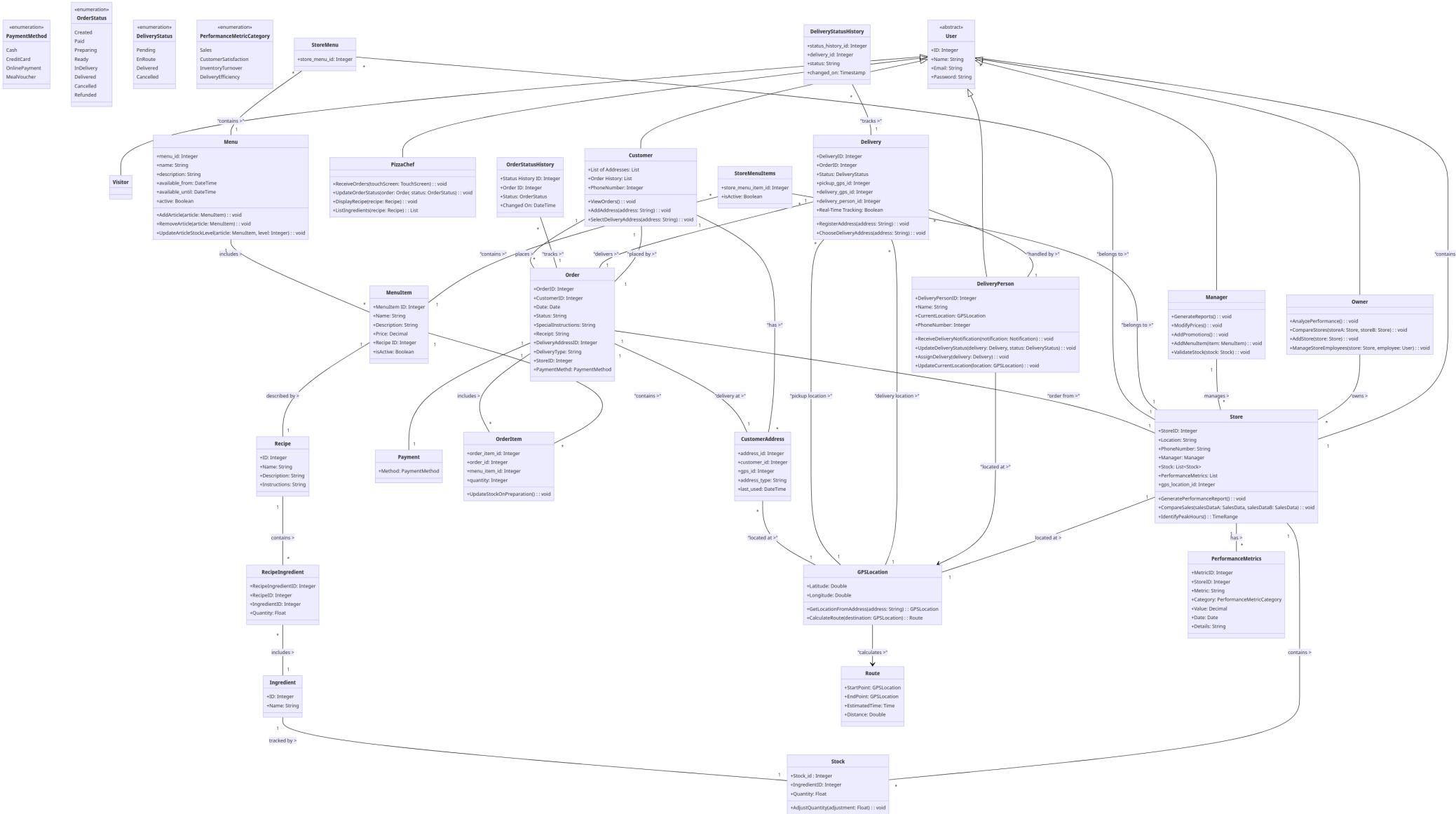
Le diagramme de classes est élaboré conformément aux exigences métiers et aux objectifs fonctionnels spécifiques du projet. Il comprend les éléments suivants :

- **Classes** : Chaque classe représente une entité significative du domaine d'affaires, telle que les clients, les commandes, et les stocks d'ingrédients.
- **Attributs** : Les attributs définissent les propriétés des classes, qui correspondent aux données que le système doit gérer.
- **Méthodes** : Les méthodes associées à chaque classe décrivent les actions ou comportements que les objets de la classe peuvent exécuter.
- **Associations** : Les lignes de connexion entre les classes représentent les associations et définissent comment les objets de différentes classes interagissent entre eux.

- **Multiplicité** : Les indicateurs de multiplicité aux extrémités des associations spécifient les règles de cardinalité entre les instances de classes.

Le Diagramme de Classes UML est un outil de référence pour les développeurs, architectes de données et testeurs. Il assure que tous les participants au projet partagent une vision commune de la structure du système. En outre, il fournit une base solide pour la conception de la base de données et le développement de l'application, en s'assurant que toutes les fonctionnalités répondent de manière cohérente aux besoins métiers identifiés.

La visualisation précise des relations et des dépendances dans le diagramme aide à anticiper et à prévenir les problèmes de conception potentiels, facilitant ainsi une approche de développement plus efficace et une maintenance plus aisée du système à long terme.



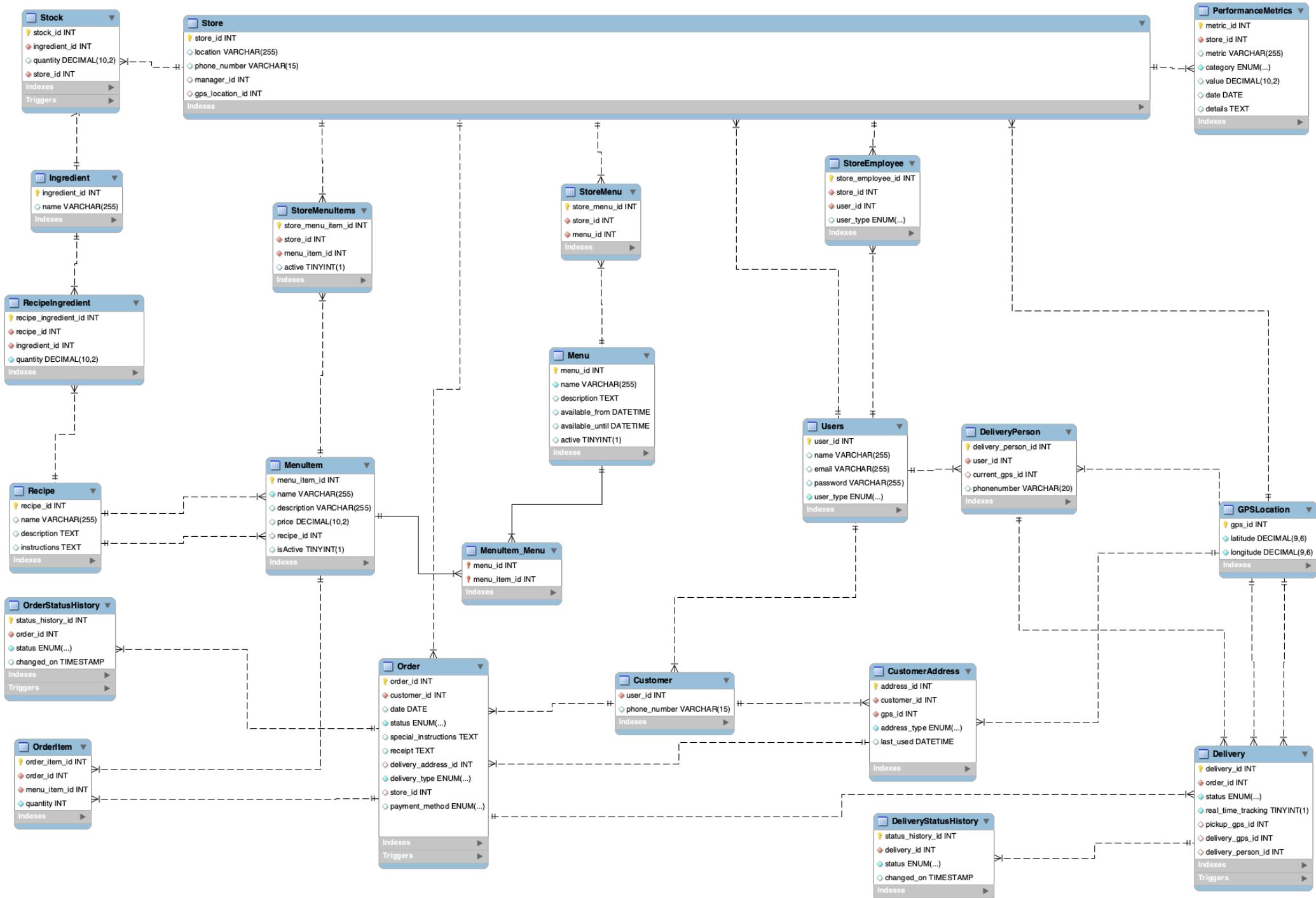
MODÈLE PHYSIQUE DE DONNÉES (MPD)

Le Modèle Physique de Données (MPD) est une représentation concrète et détaillée de la structure de la base de données qui sera utilisée pour stocker et gérer les informations dans le système d'information d'OC Pizza. Cette section du document technique expose la configuration des tables, des champs, des types de données, des clés primaires, des clés étrangères et des contraintes de la base de données qui seront déployées pour assurer la persistance des données du système. Le MPD est le résultat d'une traduction méticuleuse du Modèle Conceptuel de Données et du Diagramme de Classes UML en une structure de base de données optimisée pour les opérations et les performances réelles. Il inclut les éléments suivants :

- **Tables de Base de Données** : Chaque table correspond à une classe dans le Diagramme de Classes UML ou à une collection d'entités logiques, avec une structure définie pour soutenir les exigences de données de l'application.
- **Attributs et Types de Données** : Les colonnes de chaque table avec des types de données spécifiés garantissent que les informations sont stockées de manière appropriée, assurant l'intégrité et la qualité des données.
- **Relations et Intégrité Référentielle** : Les associations entre les tables sont formalisées par des relations, y compris des contraintes de clés étrangères, pour maintenir l'intégrité référentielle à travers la base de données.
- **Contraintes et Règles de Gestion** : Les contraintes supplémentaires, comme les clés uniques, les index et les règles de validation, sont définies pour assurer la cohérence des opérations et la validation des données au sein de la base de données.

Le MPD joue un rôle fondamental dans le processus de développement, offrant une base pour l'implémentation physique de la base de données et pour la préparation des scripts SQL nécessaires. Il est conçu pour être robuste, évolutif et facile à maintenir, avec l'objectif d'appuyer efficacement les processus métiers et les opérations quotidiennes d'OC Pizza.

En fin de compte, le MPD sert de pierre angulaire technique pour la construction, l'optimisation et le déploiement de la solution de base de données, garantissant que les solutions technologiques déployées sont en parfaite adéquation avec les modèles conceptuels et les exigences fonctionnelles du projet.



FICHIER SQL DE LA BASE DE DONNÉES

La section suivante du document de spécifications techniques présente les scripts SQL, un ensemble de commandes structurées qui serviront à la création et à la configuration de la base de données pour le système d'information d'OC Pizza. Ces scripts sont le fruit d'un processus de conception minutieux, visant à transcrire le Modèle Physique de Données (MPD) en une série d'instructions exécutables qui prépareront l'environnement de données pour une utilisation opérationnelle.

Les scripts SQL inclus ici couvrent une variété de fonctions nécessaires pour :

- **Création de Structures de Données:** Définir les tables, les colonnes, et les types de données spécifiques, en alignement direct avec notre MPD.
- **Définition des Relations :** Établir des relations précises entre les tables, y compris les contraintes de clés étrangères, pour maintenir l'intégrité référentielle au sein de la base de données.
- **Mise en Place de Contraintes :** Imposer des règles pour assurer l'exactitude et la validité des données à travers des contraintes de validation et d'unicité.
- **Insertion de Données Initiales :** Pré-remplir la base de données avec un ensemble de données de test ou de démarrage pour faciliter le développement initial et les tests fonctionnels.

Chaque script est soigneusement commenté pour assurer la compréhension et la maintenabilité, permettant aux développeurs et aux administrateurs de bases de données de suivre le flux logique de la construction de la base de données et de faciliter les éventuelles modifications ou extensions futures.

La mise à disposition de ces scripts SQL vise non seulement à simplifier le processus de déploiement initial mais aussi à offrir un mécanisme fiable pour répliquer ou restaurer la base de données selon les besoins, assurant une continuité d'activité et une robustesse système pour OC Pizza.

En résumé, cette section sert de guide opérationnel pour initialiser l'infrastructure de données qui soutiendra les processus critiques d'OC Pizza, et joue un rôle essentiel dans la concrétisation de notre solution informatique.


```

1 -- -----
2 -- Schema ocpizza2
3 --
4 CREATE SCHEMA IF NOT EXISTS `ocpizza2` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci ;
5 USE `ocpizza2` ;
6
7 --
8 -- Table `ocpizza2`.`Users`
9 --
10 CREATE TABLE IF NOT EXISTS `ocpizza2`.`Users` (
11   `user_id` INT NOT NULL AUTO_INCREMENT,
12   `name` VARCHAR(255) NULL DEFAULT NULL,
13   `email` VARCHAR(255) NULL DEFAULT NULL,
14   `password` VARCHAR(255) NULL DEFAULT NULL,
15   `user_type` ENUM('Visitor', 'Customer', 'Manager', 'PizzaChef', 'DeliveryPerson', 'Owner') NOT NULL,
16   PRIMARY KEY (`user_id`)
17 ENGINE = InnoDB
18 AUTO_INCREMENT = 21
19 DEFAULT CHARACTER SET = utf8mb4
20 COLLATE = utf8mb4_0900_ai_ci;
21
22 --
23 -- Table `ocpizza2`.`Customer`
24 --
25 CREATE TABLE IF NOT EXISTS `ocpizza2`.`Customer` (
26   `user_id` INT NOT NULL,
27   `phone_number` VARCHAR(15) NULL DEFAULT NULL,
28   INDEX `user_id` (`user_id` ASC) VISIBLE,
29   CONSTRAINT `customer_ibfk_1`
30     FOREIGN KEY (`user_id`)
31     REFERENCES `ocpizza2`.`Users` (`user_id`)
32     ON DELETE CASCADE)
33 ENGINE = InnoDB
34 DEFAULT CHARACTER SET = utf8mb4
35 COLLATE = utf8mb4_0900_ai_ci;
36
37

```

```

1 -- -----
2 -- Table `ocpizza2`.`GPSLocation`
3 --
4 CREATE TABLE IF NOT EXISTS `ocpizza2`.`GPSLocation` (
5   `gps_id` INT NOT NULL AUTO_INCREMENT,
6   `latitude` DECIMAL(9,6) NOT NULL,
7   `longitude` DECIMAL(9,6) NOT NULL,
8   PRIMARY KEY (`gps_id`)
9 ENGINE = InnoDB
10 AUTO_INCREMENT = 16
11 DEFAULT CHARACTER SET = utf8mb4
12 COLLATE = utf8mb4_0900_ai_ci;
13
14 --
15 -- Table `ocpizza2`.`CustomerAddress`
16 --
17 CREATE TABLE IF NOT EXISTS `ocpizza2`.`CustomerAddress` (
18   `address_id` INT NOT NULL AUTO_INCREMENT,
19   `customer_id` INT NOT NULL,
20   `gps_id` INT NOT NULL,
21   `address_type` ENUM('Home', 'Work', 'Other') NOT NULL,
22   `last_used` DATETIME NULL DEFAULT NULL,
23   PRIMARY KEY (`address_id`),
24   INDEX `customer_id_idx` (`customer_id` ASC) VISIBLE,
25   INDEX `gps_id_idx` (`gps_id` ASC) VISIBLE,
26   CONSTRAINT `customeraddress_ibfk_1`
27     FOREIGN KEY (`customer_id`)
28     REFERENCES `ocpizza2`.`Customer` (`user_id`)
29     ON DELETE CASCADE,
30   CONSTRAINT `customeraddress_ibfk_2`
31     FOREIGN KEY (`gps_id`)
32     REFERENCES `ocpizza2`.`GPSLocation` (`gps_id`)
33 ENGINE = InnoDB
34 AUTO_INCREMENT = 10
35 DEFAULT CHARACTER SET = utf8mb4
36 COLLATE = utf8mb4_0900_ai_ci;
37
38

```

```

1  --
2  -- Table `ocpizza2`.`Store`
3  --
4  CREATE TABLE IF NOT EXISTS `ocpizza2`.`Store` (
5      `store_id` INT NOT NULL AUTO_INCREMENT,
6      `location` VARCHAR(255) NULL DEFAULT NULL,
7      `phone_number` VARCHAR(15) NULL DEFAULT NULL,
8      `manager_id` INT NULL DEFAULT NULL,
9      `gps_location_id` INT NULL DEFAULT NULL,
10     PRIMARY KEY (`store_id`),
11     INDEX `manager_id` (`manager_id` ASC) VISIBLE,
12     INDEX `fk_store_gps_location` (`gps_location_id` ASC) VISIBLE,
13     CONSTRAINT `fk_store_gps_location`
14         FOREIGN KEY (`gps_location_id`)
15             REFERENCES `ocpizza2`.`GPSLocation` (`gps_id`),
16     CONSTRAINT `store_ibfk_1`
17         FOREIGN KEY (`manager_id`)
18             REFERENCES `ocpizza2`.`Users` (`user_id`)
19         ON DELETE SET NULL)
20 ENGINE = InnoDB
21 AUTO_INCREMENT = 6
22 DEFAULT CHARACTER SET = utf8mb4
23 COLLATE = utf8mb4_0900_ai_ci;
24
25
26 -- -----
27 -- Table `ocpizza2`.`Order`
28 --
29 CREATE TABLE IF NOT EXISTS `ocpizza2`.`Order` (
30     `order_id` INT NOT NULL AUTO_INCREMENT,
31     `customer_id` INT NOT NULL,
32     `date` DATE NULL DEFAULT NULL,
33     `status` ENUM('Created', 'Preparing', 'Ready', 'Paid', 'InDelivery', 'Delivered', 'Cancelled', 'Refund') NOT NULL,
34     `special_instructions` TEXT NULL DEFAULT NULL,
35     `receipt` TEXT NULL DEFAULT NULL,
36     `delivery_address_id` INT NULL DEFAULT NULL,
37     `delivery_type` ENUM('Delivery', 'Takeaway') NOT NULL DEFAULT 'Takeaway',
38     `store_id` INT NULL DEFAULT NULL,
39     `payment_method` ENUM('Cash', 'CreditCard', 'OnlinePayment', 'MealVoucher') NULL DEFAULT NULL,
40     PRIMARY KEY (`order_id`),
41     INDEX `customer_id` (`customer_id` ASC) VISIBLE,
42     INDEX `fk_order_delivery_address` (`delivery_address_id` ASC) VISIBLE,
43     INDEX `fk_order_store` (`store_id` ASC) VISIBLE,
44     CONSTRAINT `fk_order_delivery_address`
45         FOREIGN KEY (`delivery_address_id`)
46             REFERENCES `ocpizza2`.`CustomerAddress` (`address_id`),
47     CONSTRAINT `fk_order_store`
48         FOREIGN KEY (`store_id`)
49             REFERENCES `ocpizza2`.`Store` (`store_id`),
50     CONSTRAINT `order_ibfk_1`
51         FOREIGN KEY (`customer_id`)
52             REFERENCES `ocpizza2`.`Customer` (`user_id`)
53         ON DELETE CASCADE)
54 ENGINE = InnoDB
55 AUTO_INCREMENT = 19
56 DEFAULT CHARACTER SET = utf8mb4
57 COLLATE = utf8mb4_0900_ai_ci;
58

```

```

1  --
2  -- Table `ocpizza2`.`DeliveryPerson`
3  --
4  CREATE TABLE IF NOT EXISTS `ocpizza2`.`DeliveryPerson` (
5      `delivery_person_id` INT NOT NULL AUTO_INCREMENT,
6      `user_id` INT NOT NULL,
7      `current_gps_id` INT NULL DEFAULT NULL,
8      `phonenumber` VARCHAR(20) NULL DEFAULT NULL,
9      PRIMARY KEY (`delivery_person_id`),
10     INDEX `user_id` (`user_id` ASC) VISIBLE,
11     INDEX `current_gps_id` (`current_gps_id` ASC) VISIBLE,
12     CONSTRAINT `deliveryperson_ibfk_1`
13         FOREIGN KEY (`user_id`)
14             REFERENCES `ocpizza2`.`Users` (`user_id`)
15         ON DELETE CASCADE,
16     CONSTRAINT `deliveryperson_ibfk_2`
17         FOREIGN KEY (`current_gps_id`)
18             REFERENCES `ocpizza2`.`GPSLocation` (`gps_id`)
19 ENGINE = InnoDB
20 AUTO_INCREMENT = 5
21 DEFAULT CHARACTER SET = utf8mb4
22 COLLATE = utf8mb4_0900_ai_ci;
23
24
25 -- -----
26 -- Table `ocpizza2`.`Delivery`
27 --
28 CREATE TABLE IF NOT EXISTS `ocpizza2`.`Delivery` (
29     `delivery_id` INT NOT NULL AUTO_INCREMENT,
30     `order_id` INT NOT NULL,
31     `status` ENUM('Pending', 'EnRoute', 'Delivered', 'Cancelled') NOT NULL,
32     `real_time_tracking` TINYINT(1) NOT NULL DEFAULT '0',
33     `pickup_gps_id` INT NULL DEFAULT NULL,
34     `delivery_gps_id` INT NULL DEFAULT NULL,
35     `delivery_person_id` INT NULL DEFAULT NULL,
36     PRIMARY KEY (`delivery_id`),
37     INDEX `order_id` (`order_id` ASC) VISIBLE,
38     INDEX `fk_delivery_pickup_gps` (`pickup_gps_id` ASC) VISIBLE,
39     INDEX `fk_delivery_delivery_gps` (`delivery_gps_id` ASC) VISIBLE,
40     INDEX `fk_delivery_deliveryperson` (`delivery_person_id` ASC) VISIBLE,
41     CONSTRAINT `delivery_ibfk_1`
42         FOREIGN KEY (`order_id`)
43             REFERENCES `ocpizza2`.`Order` (`order_id`)
44         ON DELETE CASCADE,
45     CONSTRAINT `fk_delivery_delivery_gps`
46         FOREIGN KEY (`delivery_gps_id`)
47             REFERENCES `ocpizza2`.`GPSLocation` (`gps_id`),
48     CONSTRAINT `fk_delivery_deliveryperson`
49         FOREIGN KEY (`delivery_person_id`)
50             REFERENCES `ocpizza2`.`DeliveryPerson` (`delivery_person_id`),
51     CONSTRAINT `fk_delivery_pickup_gps`
52         FOREIGN KEY (`pickup_gps_id`)
53             REFERENCES `ocpizza2`.`GPSLocation` (`gps_id`)
54 ENGINE = InnoDB
55 AUTO_INCREMENT = 2
56 DEFAULT CHARACTER SET = utf8mb4
57 COLLATE = utf8mb4_0900_ai_ci;

```

```

1 --- -----
2 --- Table `ocpizza2`.`DeliveryStatusHistory` -----
3 ---
4 CREATE TABLE IF NOT EXISTS `ocpizza2`.`DeliveryStatusHistory` (
5   `status_history_id` INT NOT NULL AUTO_INCREMENT,
6   `delivery_id` INT NOT NULL,
7   `status` ENUM('Pending', 'EnRoute', 'Delivered', 'Cancelled') NOT NULL,
8   `changed_on` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
9   PRIMARY KEY (`status_history_id`),
10  INDEX `delivery_id` (`delivery_id` ASC) VISIBLE,
11  CONSTRAINT `deliverystatushistory_ibfk_1`
12    FOREIGN KEY (`delivery_id`)
13      REFERENCES `ocpizza2`.`Delivery` (`delivery_id`)
14 ENGINE = InnoDB
15 DEFAULT CHARACTER SET = utf8mb4
16 COLLATE = utf8mb4_0900_ai_ci;
17
18
19 --- -----
20 --- Table `ocpizza2`.`Ingredient` -----
21 ---
22 CREATE TABLE IF NOT EXISTS `ocpizza2`.`Ingredient` (
23   `ingredient_id` INT NOT NULL AUTO_INCREMENT,
24   `name` VARCHAR(255) NULL DEFAULT NULL,
25   PRIMARY KEY (`ingredient_id`))
26 ENGINE = InnoDB
27 AUTO_INCREMENT = 25
28 DEFAULT CHARACTER SET = utf8mb4
29 COLLATE = utf8mb4_0900_ai_ci;
30
31
32 --- -----
33 --- Table `ocpizza2`.`Menu` -----
34 ---
35 CREATE TABLE IF NOT EXISTS `ocpizza2`.`Menu` (
36   `menu_id` INT NOT NULL AUTO_INCREMENT,
37   `name` VARCHAR(255) NOT NULL,
38   `description` TEXT NULL DEFAULT NULL,
39   `available_from` DATETIME NULL DEFAULT NULL,
40   `available_until` DATETIME NULL DEFAULT NULL,
41   `active` TINYINT(1) NULL DEFAULT '1',
42   PRIMARY KEY (`menu_id`))
43 ENGINE = InnoDB
44 AUTO_INCREMENT = 5
45 DEFAULT CHARACTER SET = utf8mb4
46 COLLATE = utf8mb4_0900_ai_ci;

```

```

1 --- -----
2 --- Table `ocpizza2`.`Recipe` -----
3 ---
4 CREATE TABLE IF NOT EXISTS `ocpizza2`.`Recipe` (
5   `recipe_id` INT NOT NULL AUTO_INCREMENT,
6   `name` VARCHAR(255) NULL DEFAULT NULL,
7   `description` TEXT NULL DEFAULT NULL,
8   `instructions` TEXT NULL DEFAULT NULL,
9   PRIMARY KEY (`recipe_id`))
10 ENGINE = InnoDB
11 AUTO_INCREMENT = 10
12 DEFAULT CHARACTER SET = utf8mb4
13 COLLATE = utf8mb4_0900_ai_ci;
14
15
16 --- -----
17 --- Table `ocpizza2`.`MenuItem` -----
18 ---
19 CREATE TABLE IF NOT EXISTS `ocpizza2`.`MenuItem` (
20   `menu_item_id` INT NOT NULL AUTO_INCREMENT,
21   `name` VARCHAR(255) NOT NULL,
22   `description` VARCHAR(255) NULL DEFAULT NULL,
23   `price` DECIMAL(10,2) NULL DEFAULT NULL,
24   `recipe_id` INT NULL DEFAULT NULL,
25   `isActive` TINYINT(1) NULL DEFAULT '1',
26   PRIMARY KEY (`menu_item_id`),
27   INDEX `fk_menuitem_recipe` (`recipe_id` ASC) VISIBLE,
28   CONSTRAINT `fk_menuitem_recipe`
29     FOREIGN KEY (`recipe_id`)
30       REFERENCES `ocpizza2`.`Recipe` (`recipe_id`)
31       ON DELETE SET NULL,
32   CONSTRAINT `menuitem_ibfk_1`
33     FOREIGN KEY (`recipe_id`)
34       REFERENCES `ocpizza2`.`Recipe` (`recipe_id`)
35       ON DELETE SET NULL)
36 ENGINE = InnoDB
37 AUTO_INCREMENT = 10
38 DEFAULT CHARACTER SET = utf8mb4
39 COLLATE = utf8mb4_0900_ai_ci;
40
41
42 --- -----
43 --- Table `ocpizza2`.`MenuItem_Menu` -----
44 ---
45 CREATE TABLE IF NOT EXISTS `ocpizza2`.`MenuItem_Menu` (
46   `menu_id` INT NOT NULL,
47   `menu_item_id` INT NOT NULL,
48   PRIMARY KEY (`menu_id`, `menu_item_id`),
49   INDEX `menu_item_id` (`menu_item_id` ASC) VISIBLE,
50   CONSTRAINT `menuitem_menu_ibfk_1`
51     FOREIGN KEY (`menu_id`)
52       REFERENCES `ocpizza2`.`Menu` (`menu_id`)
53       ON DELETE CASCADE,
54   CONSTRAINT `menuitem_menu_ibfk_2`
55     FOREIGN KEY (`menu_item_id`)
56       REFERENCES `ocpizza2`.`MenuItem` (`menu_item_id`)
57       ON DELETE CASCADE)
58 ENGINE = InnoDB
59 DEFAULT CHARACTER SET = utf8mb4
60 COLLATE = utf8mb4_0900_ai_ci;

```

```

1  -- Table `ocpizza2`.`OrderItem`
2  --
3  --
4  CREATE TABLE IF NOT EXISTS `ocpizza2`.`OrderItem` (
5    `order_item_id` INT NOT NULL AUTO_INCREMENT,
6    `order_id` INT NOT NULL,
7    `menu_item_id` INT NOT NULL,
8    `quantity` INT NOT NULL,
9    PRIMARY KEY (`order_item_id`),
10   INDEX `order_id` (`order_id` ASC) VISIBLE,
11   INDEX `menu_item_id` (`menu_item_id` ASC) VISIBLE,
12   CONSTRAINT `orderitem_ibfk_1`
13     FOREIGN KEY (`order_id`)
14       REFERENCES `ocpizza2`.`Order` (`order_id`)
15     ON DELETE CASCADE,
16   CONSTRAINT `orderitem_ibfk_2`
17     FOREIGN KEY (`menu_item_id`)
18       REFERENCES `ocpizza2`.`MenuItem` (`menu_item_id`)
19     ON DELETE CASCADE)
20 ENGINE = InnoDB
21 AUTO_INCREMENT = 11
22 DEFAULT CHARACTER SET = utf8mb4
23 COLLATE = utf8mb4_0900_ai_ci;
24
25
26  --
27  -- Table `ocpizza2`.`OrderStatusHistory`
28  --
29 CREATE TABLE IF NOT EXISTS `ocpizza2`.`OrderStatusHistory` (
30   `status_history_id` INT NOT NULL AUTO_INCREMENT,
31   `order_id` INT NOT NULL,
32   `status` ENUM('Created', 'Preparing', 'Ready', 'Paid', 'InDelivery', 'Delivered', 'Cancelled', 'Refund') NOT NULL,
33   `changed_on` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
34   PRIMARY KEY (`status_history_id`),
35   INDEX `order_id` (`order_id` ASC) VISIBLE,
36   CONSTRAINT `orderstatushistory_ibfk_1`
37     FOREIGN KEY (`order_id`)
38       REFERENCES `ocpizza2`.`Order` (`order_id`)
39     ON DELETE CASCADE)
40 ENGINE = InnoDB
41 AUTO_INCREMENT = 19
42 DEFAULT CHARACTER SET = utf8mb4
43 COLLATE = utf8mb4_0900_ai_ci;
44
45
46  --
47  -- Table `ocpizza2`.`PerformanceMetrics`
48  --
49 CREATE TABLE IF NOT EXISTS `ocpizza2`.`PerformanceMetrics` (
50   `metric_id` INT NOT NULL AUTO_INCREMENT,
51   `store_id` INT NOT NULL,
52   `metric` VARCHAR(255) NULL DEFAULT NULL,
53   `category` ENUM('Sales', 'CustomerSatisfaction', 'InventoryTurnover', 'DeliveryEfficiency') NOT NULL,
54   `value` DECIMAL(10,2) NULL DEFAULT NULL,
55   `date` DATE NULL DEFAULT NULL,
56   `details` TEXT NULL DEFAULT NULL,
57   PRIMARY KEY (`metric_id`),
58   INDEX `store_id` (`store_id` ASC) VISIBLE,
59   CONSTRAINT `performancemetrics_ibfk_1`
60     FOREIGN KEY (`store_id`)
61       REFERENCES `ocpizza2`.`Store` (`store_id`)
62     ON DELETE CASCADE)
63 ENGINE = InnoDB
64 DEFAULT CHARACTER SET = utf8mb4
65 COLLATE = utf8mb4_0900_ai_ci;

```

```

1  -- Table `ocpizza2`.`RecipeIngredient`
2  --
3  --
4  CREATE TABLE IF NOT EXISTS `ocpizza2`.`RecipeIngredient` (
5    `recipe_ingredient_id` INT NOT NULL AUTO_INCREMENT,
6    `recipe_id` INT NOT NULL,
7    `ingredient_id` INT NOT NULL,
8    `quantity` DECIMAL(10,2) NOT NULL,
9    PRIMARY KEY (`recipe_ingredient_id`),
10   INDEX `recipe_id` (`recipe_id` ASC) VISIBLE,
11   INDEX `ingredient_id` (`ingredient_id` ASC) VISIBLE,
12   CONSTRAINT `recipeingredient_ibfk_1`
13     FOREIGN KEY (`recipe_id`)
14       REFERENCES `ocpizza2`.`Recipe` (`recipe_id`)
15     ON DELETE CASCADE,
16   CONSTRAINT `recipeingredient_ibfk_2`
17     FOREIGN KEY (`ingredient_id`)
18       REFERENCES `ocpizza2`.`Ingredient` (`ingredient_id`)
19     ON DELETE CASCADE)
20 ENGINE = InnoDB
21 AUTO_INCREMENT = 33
22 DEFAULT CHARACTER SET = utf8mb4
23 COLLATE = utf8mb4_0900_ai_ci;
24
25
26  --
27  -- Table `ocpizza2`.`Stock`
28  --
29 CREATE TABLE IF NOT EXISTS `ocpizza2`.`Stock` (
30   `stock_id` INT NOT NULL AUTO_INCREMENT,
31   `ingredient_id` INT NOT NULL,
32   `quantity` DECIMAL(10,2) NULL DEFAULT NULL,
33   `store_id` INT NOT NULL,
34   PRIMARY KEY (`stock_id`),
35   INDEX `ingredient_id` (`ingredient_id` ASC) VISIBLE,
36   INDEX `store_id` (`store_id` ASC) VISIBLE,
37   CONSTRAINT `stock_ibfk_1`
38     FOREIGN KEY (`ingredient_id`)
39       REFERENCES `ocpizza2`.`Ingredient` (`ingredient_id`)
40     ON DELETE CASCADE,
41   CONSTRAINT `stock_ibfk_2`
42     FOREIGN KEY (`store_id`)
43       REFERENCES `ocpizza2`.`Store` (`store_id`)
44     ON DELETE CASCADE)
45 ENGINE = InnoDB
46 AUTO_INCREMENT = 121
47 DEFAULT CHARACTER SET = utf8mb4
48 COLLATE = utf8mb4_0900_ai_ci;
49

```

```

1 -- -----
2 -- Table `ocpizza2`.`StoreEmployee`
3 --
4 CREATE TABLE IF NOT EXISTS `ocpizza2`.`StoreEmployee` (
5   `store_employee_id` INT NOT NULL AUTO_INCREMENT,
6   `store_id` INT NOT NULL,
7   `user_id` INT NOT NULL,
8   `user_type` ENUM('Visitor', 'Customer', 'Manager', 'PizzaChef', 'DeliveryPerson', 'Owner') NULL DEFAULT NULL,
9   PRIMARY KEY (`store_employee_id`),
10  INDEX `store_id` (`store_id` ASC) VISIBLE,
11  INDEX `user_id` (`user_id` ASC) VISIBLE,
12  CONSTRAINT `storeemployee_ibfk_1`
13    FOREIGN KEY (`store_id`)
14      REFERENCES `ocpizza2`.`Store` (`store_id`)
15    ON DELETE CASCADE,
16  CONSTRAINT `storeemployee_ibfk_2`
17    FOREIGN KEY (`user_id`)
18      REFERENCES `ocpizza2`.`Users` (`user_id`)
19    ON DELETE CASCADE)
20 ENGINE = InnoDB
21 AUTO_INCREMENT = 4
22 DEFAULT CHARACTER SET = utf8mb4
23 COLLATE = utf8mb4_0900_ai_ci;
24
25
26 -- -----
27 -- Table `ocpizza2`.`StoreMenu`
28 --
29 CREATE TABLE IF NOT EXISTS `ocpizza2`.`StoreMenu` (
30   `store_menu_id` INT NOT NULL AUTO_INCREMENT,
31   `store_id` INT NOT NULL,
32   `menu_id` INT NOT NULL,
33   PRIMARY KEY (`store_menu_id`),
34   INDEX `store_id` (`store_id` ASC) VISIBLE,
35   INDEX `menu_id` (`menu_id` ASC) VISIBLE,
36   CONSTRAINT `storemenu_ibfk_1`
37     FOREIGN KEY (`store_id`)
38       REFERENCES `ocpizza2`.`Store` (`store_id`)
39     ON DELETE CASCADE,
40   CONSTRAINT `storemenu_ibfk_2`
41     FOREIGN KEY (`menu_id`)
42       REFERENCES `ocpizza2`.`Menu` (`menu_id`)
43     ON DELETE CASCADE)
44 ENGINE = InnoDB
45 AUTO_INCREMENT = 21
46 DEFAULT CHARACTER SET = utf8mb4
47 COLLATE = utf8mb4_0900_ai_ci;
48
49
50 -- -----
51 -- Table `ocpizza2`.`StoreMenuItems`
52 --
53 CREATE TABLE IF NOT EXISTS `ocpizza2`.`StoreMenuItems` (
54   `store_menu_item_id` INT NOT NULL AUTO_INCREMENT,
55   `store_id` INT NOT NULL,
56   `menu_item_id` INT NOT NULL,
57   `active` TINYINT(1) NULL DEFAULT '1',
58   PRIMARY KEY (`store_menu_item_id`),
59   INDEX `store_id` (`store_id` ASC) VISIBLE,
60   INDEX `menu_item_id` (`menu_item_id` ASC) VISIBLE,
61   CONSTRAINT `storemenuitems_ibfk_1`
62     FOREIGN KEY (`store_id`)
63       REFERENCES `ocpizza2`.`Store` (`store_id`)
64     ON DELETE CASCADE,
65   CONSTRAINT `storemenuitems_ibfk_2`
66     FOREIGN KEY (`menu_item_id`)
67       REFERENCES `ocpizza2`.`MenuItem` (`menu_item_id`)
68     ON DELETE CASCADE)
69 ENGINE = InnoDB
70 AUTO_INCREMENT = 68
71 DEFAULT CHARACTER SET = utf8mb4
72 COLLATE = utf8mb4_0900_ai_ci;
73
74 USE `ocpizza2` ;

```

```

1 -- -----
2 -- procedure FetchIngredientsNeeded
3 --
4
5 DELIMITER $$$
6 USE `ocpizza2`$$
7 CREATE DEFINER=`root`@`localhost` PROCEDURE `FetchIngredientsNeeded`(orderId INT)
8 BEGIN
9   DECLARE ingredient_id INT;
10  DECLARE needed_quantity DECIMAL(10,2);
11  DECLARE done INT DEFAULT 0;
12
13  DECLARE recipe_cursor CURSOR FOR
14    SELECT ri.ingredient_id, SUM(ri.quantity * oi.quantity) AS needed
15    FROM OrderItem oi
16    JOIN MenuItem mi ON oi.menu_item_id = mi.menu_item_id
17    JOIN RecipeIngredient ri ON mi.recipe_id = ri.recipe_id
18    WHERE oi.order_id = orderId
19    GROUP BY ri.ingredient_id;
20
21  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
22
23  OPEN recipe_cursor;
24
25  fetch_loop: LOOP
26    FETCH recipe_cursor INTO ingredient_id, needed_quantity;
27    IF done THEN
28      LEAVE fetch_loop;
29    END IF;
30    -- Pour voir les résultats, vous pouvez choisir de les afficher, par exemple:
31    SELECT ingredient_id, needed_quantity;
32  END LOOP;
33
34  CLOSE recipe_cursor;
35 END$$
36
37 DELIMITER ;
38
39 SET SQL_MODE=@OLD_SQL_MODE;
40 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
41 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
42 USE `ocpizza2`;

```

```

1  DELIMITER $$ 
2  USE `ocpizza2`$$
3  CREATE
4  DEFINER='root'@'localhost'
5  TRIGGER `ocpizza2`.`RecordOrderStatusChange`
6  AFTER UPDATE ON `ocpizza2`.`Order`
7  FOR EACH ROW
8  BEGIN
9    -- Vérifier si le statut a changé
10   IF NEW.status <> OLD.status THEN
11     -- Insérer l'enregistrement du changement de statut dans OrderStatusHistory
12     INSERT INTO OrderStatusHistory (order_id, status, changed_on)
13       VALUES (NEW.order_id, NEW.status, NOW());
14   END IF;
15 END$$
16
17 USE `ocpizza2`$$
18 CREATE
19 DEFINER='root'@'localhost'
20 TRIGGER `ocpizza2`.`after_order_ready`
21 AFTER UPDATE ON `ocpizza2` .`Order`
22 FOR EACH ROW
23 BEGIN
24   -- Déclarer les variables avant toute opération
25   DECLARE pickupGpsId INT;
26   DECLARE deliveryGpsId INT;
27
28   -- Vérifier le changement de statut vers 'Ready'
29   IF NEW.status = 'Ready' AND OLD.status != 'Ready' THEN
30     -- Récupérer l'ID GPS du magasin
31     SELECT gps_location_id INTO pickupGpsId FROM Store WHERE store_id = NEW.store_id;
32     -- Récupérer l'ID GPS de l'adresse de livraison
33     SELECT gps_id INTO deliveryGpsId FROM CustomerAddress WHERE address_id = NEW.delivery_address_id;
34
35   -- Insérer dans la table Delivery
36   INSERT INTO Delivery (order_id, pickup_gps_id, delivery_gps_id, status, real_time_tracking)
37     VALUES (NEW.order_id, pickupGpsId, deliveryGpsId, 'Pending', 0);
38 END IF;
39 END$$
40

```

```

1  USE `ocpizza2`$$
2  CREATE
3  DEFINER='root'@'localhost'
4  TRIGGER `ocpizza2`.`after_delivery_update`
5  AFTER UPDATE ON `ocpizza2`.Delivery
6  FOR EACH ROW
7  BEGIN
8    IF NEW.status <> OLD.status THEN
9      INSERT INTO `DeliveryStatusHistory` (delivery_id, status)
10        VALUES (NEW.delivery_id, NEW.status);
11    END IF;
12 END$$
13
14 USE `ocpizza2`$$
15 CREATE
16 DEFINER='root'@'localhost'
17 TRIGGER `ocpizza2`.`UpdateStockOnPreparing`
18 AFTER UPDATE ON `ocpizza2`.`OrderStatusHistory`
19 FOR EACH ROW
20 BEGIN
21   -- Vérifier si le statut a changé pour 'Preparing'
22   IF NEW.status = 'Preparing' AND OLD.status <> 'Preparing' THEN
23     -- Mettre à jour le stock basé sur les items commandés
24     UPDATE Stock s
25     JOIN (
26       SELECT
27         ri.ingredient_id,
28         SUM(ri.quantity * oi.quantity) AS total_needed,
29         o.store_id
30       FROM `Order` o
31       JOIN OrderItem oi ON o.order_id = NEW.order_id
32       JOIN MenuItem mi ON oi.menu_item_id = mi.menu_item_id
33       JOIN RecipeIngredient ri ON mi.recipe_id = ri.recipe_id
34       GROUP BY ri.ingredient_id, o.store_id
35     ) AS needed ON s.ingredient_id = needed.ingredient_id AND s.store_id = needed.store_id
36     SET s.quantity = s.quantity - needed.total_needed
37     WHERE s.quantity >= needed.total_needed;
38   END IF;
39 END$$
40
41 USE `ocpizza2`$$
42 CREATE
43 DEFINER='root'@'localhost'
44 TRIGGER `ocpizza2`.`DisableStoreMenuItemsWhenStockDepleted`
45 AFTER UPDATE ON `ocpizza2`.Stock
46 FOR EACH ROW
47 BEGIN
48   IF NEW.quantity = 0 THEN
49     -- Mettre à jour uniquement les entrées de StoreMenuItems pour le magasin spécifique où le stock est épuisé
50     UPDATE StoreMenuItems smi
51     JOIN MenuItem mi ON smi.menu_item_id = mi.menu_item_id
52     JOIN RecipeIngredient ri ON mi.recipe_id = ri.recipe_id
53     JOIN Stock s ON s.ingredient_id = ri.ingredient_id AND s.store_id = smi.store_id
54     SET smi.active = FALSE
55     WHERE ri.ingredient_id = NEW.ingredient_id AND s.store_id = NEW.store_id;
56   END IF;
57 END$$
58
59 USE `ocpizza2`$$
60 CREATE
61 DEFINER='root'@'localhost'
62 TRIGGER `ocpizza2`.`ReactivateStoreMenuItemsWhenStockReplenished`
63 AFTER UPDATE ON `ocpizza2`.Stock
64 FOR EACH ROW
65 BEGIN
66   IF OLD.quantity = 0 AND NEW.quantity > 0 THEN
67     -- Réactiver les entrées de StoreMenuItems pour le magasin spécifique où le stock est réapprovisionné
68     UPDATE StoreMenuItems smi
69     JOIN MenuItem mi ON smi.menu_item_id = mi.menu_item_id
70     JOIN RecipeIngredient ri ON mi.recipe_id = ri.recipe_id
71     JOIN Stock s ON s.ingredient_id = ri.ingredient_id AND s.store_id = smi.store_id
72     SET smi.active = TRUE
73     WHERE ri.ingredient_id = NEW.ingredient_id AND s.store_id = NEW.store_id;
74   END IF;
75 END$$
76
77
78 DELIMITER ;
79

```