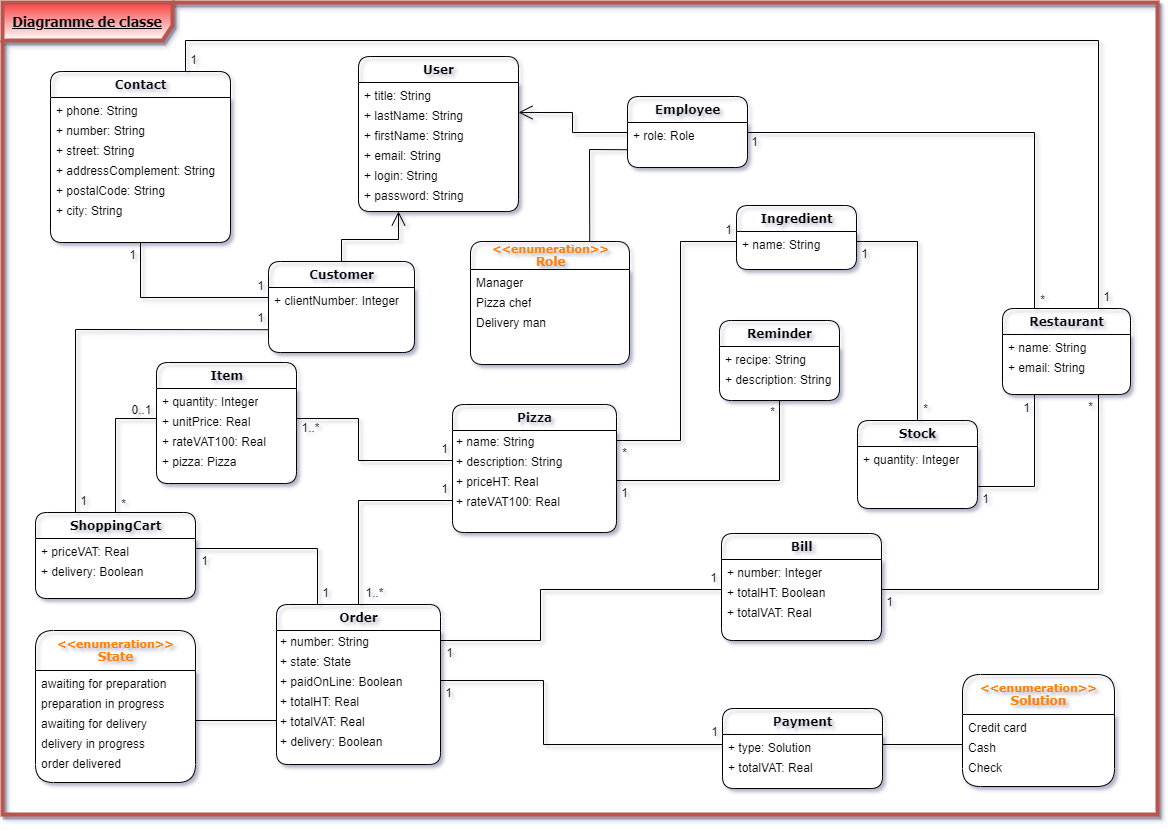
**Contexte**

« **OC Pizza** », jeune groupe de pizzeria en plein essor, est spécialisé dans la vente en livraison ou à emporter de pizza. Le groupe compte actuellement 5 points de vente et prévoit d’en ouvrir au moins 3 de plus d’ici la fin de l’année.

Dans le cas présent, il est demandé de mettre en place un nouveau système informatique qui sera déployé dans toutes les pizzerias du groupe. Ce système doit permettre notamment :

* D’améliorer l’efficacité de la gestion des commandes. Depuis leur réception à leur livraison, en passant bien entendu à leur préparation
* De suivre en temps réel les passées et en préparation
* De suivre en temps réel le stock d’ingrédients restants afin de connaitre les pizzas encore réalisables.
* De proposer un site internet afin qu’un client puisse :
* Passer une commande en plus de la prise de commande par téléphone ou directement dans un point de vente
* Payer en ligne s’il le souhaite, sinon la commande sera réglée lors de la livraison à domicile ou dans un point de vente
* Modifier ou annuler la commande tant que celle-ci n’a pas été préparée
* De proposer un aide-mémoire aux pizzaïolos indiquant la recette des produits proposés
* D’informer ou notifier les clients sur l’état de leur commande



* **Description du diagramme de classe**

Le diagramme ci-dessus représente les classes qui seront utilisées dans le cadre de notre solution technique.

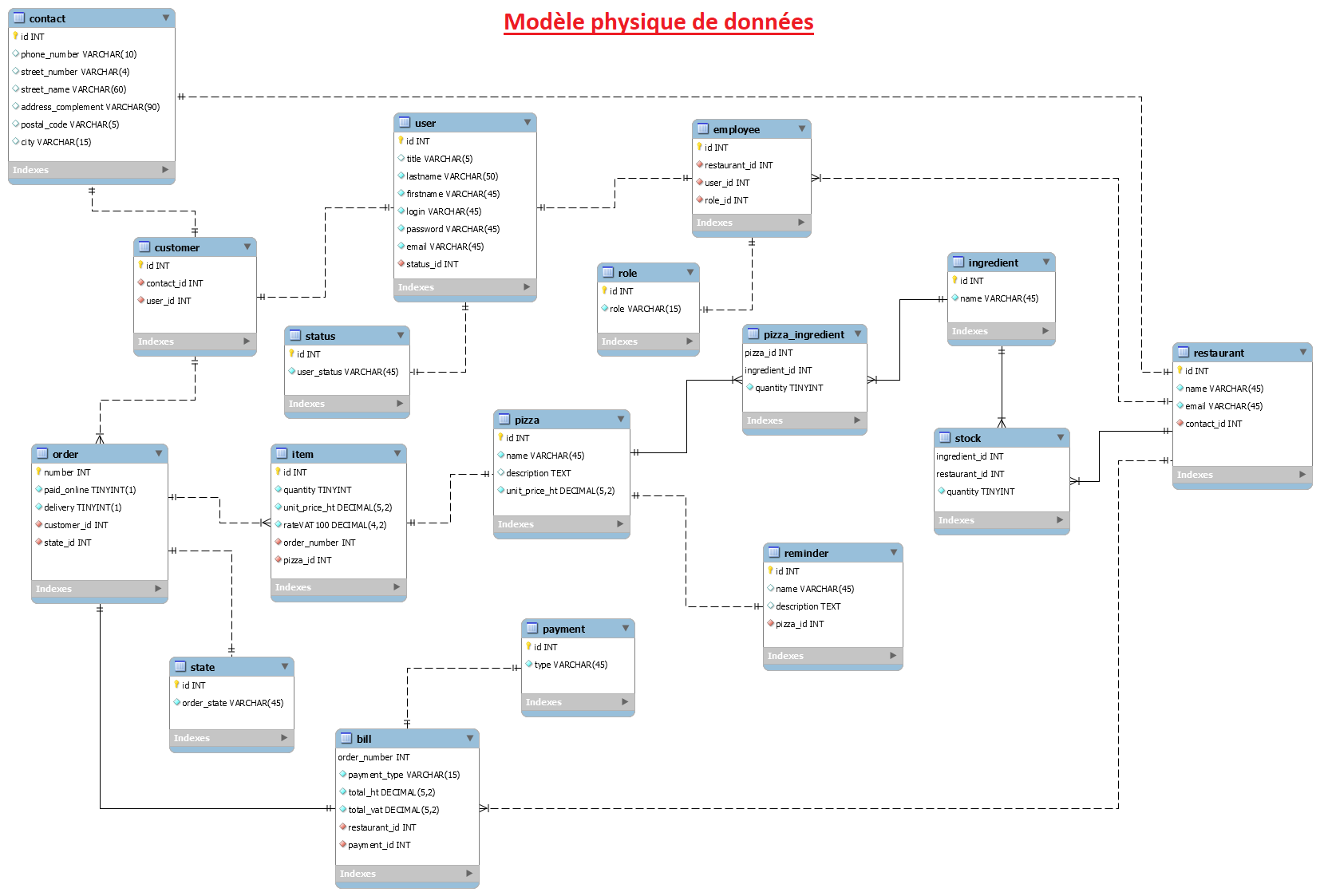
Il peut être divisé en 3 parties distinctes, une première qui concernera la « **Gestion des utilisateurs** », une seconde pour la « **Gestion des commandes** » et une dernière pour le « **Fonctionnement du restaurant** ».

Voici les classes que ces différentes parties contiennent :

* **Gestion des utilisateurs :**
* **User** : Classe mère utilisée pour identifier tous les utilisateurs, que ce soit les clients (classe héritée « **Customer** ») ou les employés (classe héritée « **Employee** »). La classe « **Employee** » possède un attribut « **role** » permettant d’identifier le rôle d’un employé au sein de la société. Cet attribut est une instance de l’énumération « **Role** ».
* **Contact** : Classe qui est utilisée pour stocker les coordonnées des clients, mais également des différents points de vente (cinq actuellement, huit avant la fin de l’année).
* **Gestion des commandes :**
* **Item** : Dans cette partie, une première classe « **Item** » (liée également à la classe « **Pizza** ») est utilisée lorsque le client ajoute un produit dans son panier. On retrouve dans les attributs de cette classe, notamment, la quantité du produit ajouté, le prix à l’unité, le taux TVA ainsi que le produit lui-même (instance de la classe « **Pizza** »).
* **ShoppingCart** : Cette classe représente le panier du client et contient les produits ajoutés par le client (instance(s) « **Item** »). Dans cette classe, un attribut booléen « **delivery** » aura une valeur « t**rue** » ou « **false** » selon que le client choisisse de se faire livrer la commande ou pas.
* **Order** : Cette classe est instanciée lorsque la commande est validée par le client. Elle contient un numéro unique et les différentes informations de la commande passée. En plus de l’attribut « **delivery** » qui concerne la livraison de la commande, un autre booléen apparait ici, « **paidOnLine** ». Cet attribut varie selon que la commande soit déjà payée (en ligne) ou pas, notamment dans le cadre d’une livraison ou de la récupération d’une commande dans un point de vente. Un autre attribut nommé « **state** » varie selon l’état d’avancement de la commande. Les différents états sont contenus dans l’énumération « **State** ».
* **Bill / Payment** : Ces deux classes concernent la facturation de la commande au client. La classe « **Payment** » est utilisée lors du paiement de la commande et contient comme attribut « **type** » qui est la solution de paiement choisie par le client (*carte de crédit, espèce, chèque*). Les différentes solutions sont contenues dans une énumération nommée « **Solution** ».

Une autre classe nommée « **Bill** » concerne la facture de la commande. Elle permet de garder un historique des commandes traitée, elle est également liée au point de vente qui a traité la commande.

* **Gestion des produits** :
* **Pizza** : Cette classe est commune aux différentes parties énoncées. Elle est instanciée notamment lors de l’ajout de produit dans le panier et contient les informations des produits.
* **Ingredient / Stock** : Ces deux classes concernent l’inventaire des ingrédients nécessaires à la confection des pizzas. La classe « Stock » est, quant à elle, liée à un point de vente en particulier.
* **Reminder** : La classe « **Reminder** » est utilisée comme aide-mémoire et contient les recettes des différentes pizzas proposées par le groupe. Elle est commune aux différents points de vente.
* **Restaurant** : Cette classe est utilisée pour identifier les différents points de vente. Elle est également liée à la classe « **Contact** » pour obtenir ici les coordonnées des différents points de vente.



* **Description du modèle physique de données**

Le diagramme ci-dessus (**modèle physique de données**) représente l’architecture de la base de données et également les relations qu’il existe entre les différentes tables.

De la même manière que le **diagramme de classe**, nous pouvons ici le diviser en trois parties distinctes, dont voici les tables principales :

* **Gestion des utilisateurs :**
* **User** : La table « **User** » est utilisée pour stocker les informations d’identification des différents utilisateurs présents dans la base de données. Cette table contient notamment les clients, mais également les employés du groupe. Afin de différencier ces deux types d’utilisateur, il existe un champ « **status\_id** » lié à la table « **Status** ». Ce champ représente les différents « **types** » d’utilisateur énumérés dans la table « **Status** ».
* **Customer**: Les champs de la table « **Customer** » sont liés aux utilisateurs « **clients** » de la table « **User** ». C’est-à-dire, ceux qui ont un « **status\_id** » équivalent à « **Customer** » (voir les explications de la table « **User** »). La table « **Customer** » est également liée à la table « **Contact** » qui contient les coordonnées des clients.
* **Employee**: De la même façon que la table « **Customer** », la table « **Employee** » est liée aux utilisateurs « **employés** » de la table « **User** », et donc, ceux qui ont un « **status\_id** » équivalent à la valeur « **Employee** ». Un champ « **role\_id** » est également présent dans cette table, et lié à la table « **Role** » qui énumère les différentes fonctions au sein du groupe. Un champ « **restaurant\_id** » est également présent pour identifier l’établissement d’accueil de l’employé.
* **Gestion des commandes :**
* **Order** : La table « **Order** » contient les informations des commandes passées par les clients. Elles sont identifiées et liées à un client par le champ « **customer\_id** ». Un autre champ « **state\_id** » représente l’état actuel de la commande. Il est lié à la table « **State** » qui énumère les différents états d’une commande (en cours de préparation, livrée etc..).
* **Item**: La table « **Item** » contient les différents éléments qui constituent une commande. Pour imager, il s’agit des différentes lignes de la commande. Elle contient notamment, le « **pizza\_id** » lié à la table « **Pizza** » et qui représente donc un produit d’une commande. On retrouve ensuite les informations d’un produit comme le prix hors-taxe, la TVA à appliquer et la quantité. Un champ « **order\_number** » lié à la table « **Order** » permet d’identifier la commande à laquelle appartient l’élément.
* **Bill** : Cette table contient les informations de facturation des différentes commandes. Un champ « **order\_number** » contient le numéro de commande lié à la facture. Le champ « **payment\_id** » lié à la table « **payment\_type** » permet d’identifier le type de paiement utilisé (CB, espèce, chèque) énuméré dans cette même table. Le champ « **restaurant\_id** » permet d’identifier l’établissement d’origine de la facture.
* **Gestion des produits :**
* **Pizza** : Cette table contient les différents produits proposés par le groupe. Outre le nom et la description du produit, on retrouve également son prix hors-taxe à l’unité.
* **Ingredient**: La table « **Ingredient** » contient tous les ingrédients utilisés pour faire les différentes pizzas proposées. On y trouve un « **id** » et le nom de l’ingrédient. Elle est liée à la table « **Pizza** » par l’intermédiaire d’une autre table nommée « **Pizza\_Ingredient** » permettant notamment d’identifier les ingrédients utilisés pour chaque pizza et en quelle quantité.
* **Stock**: Cette table liée à « **Ingredient** » permet de connaitre la disponibilité des différents ingrédients ainsi que la quantité restante. De manière à identifier les pizzas encore réalisables à n’importe quel moment du service. En plus de « **ingredient\_id** », la clé primaire composée contient également « **restaurant\_id** » qui permet d’associer un ingrédient à un point de vente en particulier. Nous pourrons donc avoir par exemple :

|  |  |  |
| --- | --- | --- |
| **Table Stock** | | |
| **Ingredient\_id** | **Restaurant\_id** | **quantité** |
| 1 | 1 | 187 |
| 1 | 2 | 329 |
| 1 | 3 | 67 |

Dans cet exemple, un même ingrédient apparait dans trois points de vente différents et donc en quantité liée à un point de vente.

* **Restaurant**: Cette table permet d’identifier les différents points de vente. Elle est également liée à la table « **Contact** » par l’intermédiaire du champ « **contact\_id** » pour retrouver les coordonnées des différents points de vente.
* **Reminder**: « **Reminder** » est la table utilisée pour l’aide-mémoire, contenant les recettes des produits proposés. Elle est destinée à être utilisée (si nécessaire) par les pizzaïolos lors de la réalisation des pizzas.