



Introduction à la SGBD

Base de données & SQL



Dr Pape Abdoulaye BARRO

Enseignant – Chercheur

Spécialiste en Télémétrie et Systèmes Intelligents

Plan

- Généralités
- Le modèle conceptuel
- Le modèle relationnel
- Le langage SQL
- Etude et réalisation d'une base de données
- La sécurité des données

Etude et réalisation

Objectif

Dans cette session, nous allons proposer une étude complète de la mise en place d'une base de données en partant de l'étude du cahier des charges jusqu'à l'interrogation de la base à l'aide du langage SQL. Nous allons faire:

- l'analyse du monde réel et la création du modèle entité-association ;
- le passage au modèle relationnel ;
- la création des tables en SQL avec intégration des contraintes de contenu ;

Etude et réalisation

Présentation de l'activité à modéliser

Nous cherchons à modéliser la gestion d'une entreprise de production et de livraison de repas à domicile: la société "**Sen BOUFF**" qui est une société qui propose des plats chauds de types sénégalaise. Le client doit choisir, dans le catalogue proposé, le plat qu'il souhaite commander. Les plats proposés sont:

➤ **pour le repas du midi:**

- Thiébou Dieun, Thiébou Yapp, Thiébou Guanar, Yassa Dieun, Yassa Yapp, Yassa Guanar, Mafé, Mborokhé, Soup Kandia, Domoda et Thiou Boulette.

➤ **Pour le repas du soir:**

- Spagetti Yapp, Spagetti Ganar, Frite Guanar, Frite Dieun, Dibi Yapp, Dibi Guanar, Ragout Yapp, Ragout Guanar, Ndambé Yapp, Ndambé Guanar, Vermicelle Yapp, Vermicelle Ganar, Dakhine, Mbakhale Saloum, Thiéré Siim, Thiéré Baassé, Couscous Yapp, Couscous Guanar.

Chaque plats est caractérisé par son nom, les ingrédients additionnels qui la composent et son prix de base. Il existe quatre type de plats: normale, large, normale amélioré (plus chère que le normale simple) ou large amélioré (plus chère que le large simple).

Etude et réalisation

Présentation de l'activité à modéliser

⊙ **Le mode de distribution**

Les plats sont livrés par des livreurs qui circulent en voiture ou à moto.

- On doit pouvoir effectuer le suivi des activités.

⊙ **Les modalités de vente**

Le mode de vente est du type prépayé: Avant toute commande, les clients doivent approvisionner leur compte. On vérifie le solde du compte avant de préparer et de livrer la commande.

- Il existe deux systèmes de bonification :
 - Un plat gratuit est offert au bout de 10 plats achetés.
 - Tout plat livré en plus d'une heure est gratuit.

Etude et réalisation

Présentation de l'activité à modéliser

⊙ **Les objectifs du système**

Le but de la mise en place de cette base de données est principalement de gérer l'activité quotidienne de vente et de livraison des plats de type sénégalaise:

- vérification du solde du compte et facturation aux clients ;
- suivi du chiffre d'affaires ;
- refus d'honorer les commandes pour lesquelles le solde du compte client est insuffisant ;
- non-facturation des plats gratuits (retard ou fidélité).

On veut également effectuer des statistiques diverses sur l'activité:

- identification du meilleur client ;
- identification du plus mauvais livreur (nombre de retards dans la livraison) et du véhicule utilisé ;
- identification du plat le plus ou le moins demandé ;

Etude et réalisation

Le modèle entité-association

Dans cette étape, nous allons essayer de déterminer les éléments qui permettront de constituer les futures tables de la base de données. Pour cela, nous procéderons en deux temps:

- la construction du graphe des entités reliées par des associations;
- la qualification des associations par leurs cardinalités.

◎ **Identification des entités et des associations**

- Nous procéderons d'abord au repérage des différentes entités avec des phrases simples qui identifient l'activité par rapport à l'énoncé en cherchant, en même temps, à identifier les objets concrets du monde réel qui semblent impliqués dans le système.
- Puis, pour chaque entité, il nous faut déterminer une clé parmi les attributs.
- Enfin, il faut caractériser les liens entre les entités par des associations.

Etude et réalisation

Le modèle entité-association

➤ Entités et attributs

Nous commencerons à identifier les mots clés dans la description générale: 'plat', 'ingrédient', 'client', 'livreur', 'véhicule'. Ces descripteurs représentent les objets familiers du monde réel. Nous avons d'autres mots clés comme 'taille', 'prix', 'compte', 'retard', 'nom' ... qui sont clairement plus des qualifiants que des objets.

C'est la notion de **commande** (objet abstrait) qui relie tous ces objets afin de constituer la représentation de l'activité.

Après ce constat, nous sommes en mesure de proposer quelques phrases permettant d'effectuer une première synthèse de l'activité.

- ☐ « Un **plat** est constitué de plusieurs **ingrédients**. »
- ☐ « Un **client** passe une **commande**. »
- ☐ « Une **commande** est livrée par un et un seul **livreur**. »
- ☐ « Une **commande** est livrée par un et un seul **véhicule**. »

Etude et réalisation

Le modèle entité-association

On peut en déduire les entités suivantes: commande, client, plat, livreur, véhicule, ingrédient.

Après un dialogue avec les différents acteurs de l'entreprise, on obtient les renseignements suivants :

- ❑ Un client **est caractérisé** par son **nom** et son **adresse**.
- ❑ Un livreur **est caractérisé** par son **nom** et son **numéro de téléphone**.
- ❑ Un véhicule **est caractérisé** par sa **marque**, son **type** et son **numéro d'immatriculation**.

Ces informations complémentaires permettent de déterminer quelques attributs des entités ainsi constituées. Les autres attributs se trouvent dans l'énoncé: par exemple « *un plat **est caractérisé** par son **nom** et par son **prix*** » ou « *un ingrédient **est caractérisé** par son **nom*** ».

L'affectation d'un attribut à une entité n'est pas toujours évidente. On va créer un attribut 'compte' qui contiendra les informations de solvabilité du client puisque l'on fonctionne en mode prépayé. Il faut se poser la question : « ***est-ce une propriété caractéristique du client ?*** » La réponse dans notre cas est aisée et l'attribut sera affecté à l'entité 'client'.

Etude et réalisation

Le modèle entité-association

➤ Entités et attributs

Comment prendre en compte le **type** de plat? Le type sert uniquement à pondérer le **prix de base** du **plat**: il n'est donc pas associé au plat, dont le prix est fixe; il ne constitue pas non plus une caractéristique d'un **client**. On l'associe logiquement à l'entité '**commande**'.

Par un raisonnement semblable, on détermine que le **retard** est associé à la commande alors que la **fidélisation** est associée au **client**.

Si l'on récapitule, on obtient les entités munies de leurs attributs suivants:

- ☐ commande (DateCom, Taille, Retard) ;
- ☐ client (NomClient, Adresse, Compte, PointsRaPlat) ;
- ☐ plat ((NomPlat, Prix) ;
- ☐ livreur (NomLivreur, Téléphone) ;
- ☐ véhicule (NumImmat, Marque, Type) ;
- ☐ ingrédient (NomIngrédient).

Etude et réalisation

Le modèle entité-association

➤ Choix de la clé

Les attributs étant identifiés, on doit maintenant choisir une clé pour chaque entité.

Si l'on prend l'entité '**client**', il est clair qu'aucun attribut ne peut convenir pour constituer une clé. De même, l'association de plusieurs attributs ne permet pas de créer une clé. On ajoute alors

classiquement un attribut identifiant qui sert de clé.

En utilisant ces mêmes arguments, on est amené à ajouter des attributs numériques comme clés pour les entités '**livreur**' et '**commande**'.

Pour l'entité '**plat**', on peut supposer que le nom du plat est significatif et qu'il peut donc servir de clé. **En pratique**, même si le nom est unique, on pourrait décider de ne pas l'utiliser car le contenu est **un peu long**.

L'entité '**véhicule**' dispose d'un champ identifiant: son numéro d'immatriculation qui est par définition unique.

Enfin, en ce qui concerne les **ingrédients**, il peut être moins évident que le nom seul de l'ingrédient suffise à l'identifier. Il est préférable de lui ajouter un numéro.

On obtient les entités suivantes munies de leurs attributs :

- ❑ commande (**NumCommande**, DateCom, Taille, Retard) ;
- ❑ client (**NumClient**, NomClient, Adresse, Compte, PointsRaPlat) ;
- ❑ plat (**NomPizza**, Prix) ;
- ❑ livreur (**CodeLivreur**, NomLivreur, Téléphone) ;
- ❑ véhicule (**NumImmat**, Marque, Type) ;
- ❑ ingrédient (**NumIngre**, NomIngrédient).

Etude et réalisation

Le modèle entité-association

➤ Associations et attributs

On peut, à partir des phrases qui ont permis de repérer les entités, en déduire les associations suivantes :

- ❑ '**Livre**' entre 'Livreur' et 'Commande' ;
- ❑ '**Transporte**' entre 'Véhicule' et 'Commande' ;
- ❑ '**Passe**' entre 'Client' et 'Commande' ;
- ❑ '**Constitue**' entre 'Plat' et 'Commande' ;
- ❑ '**Compose**' entre 'Plat' et 'Ingrédient'.

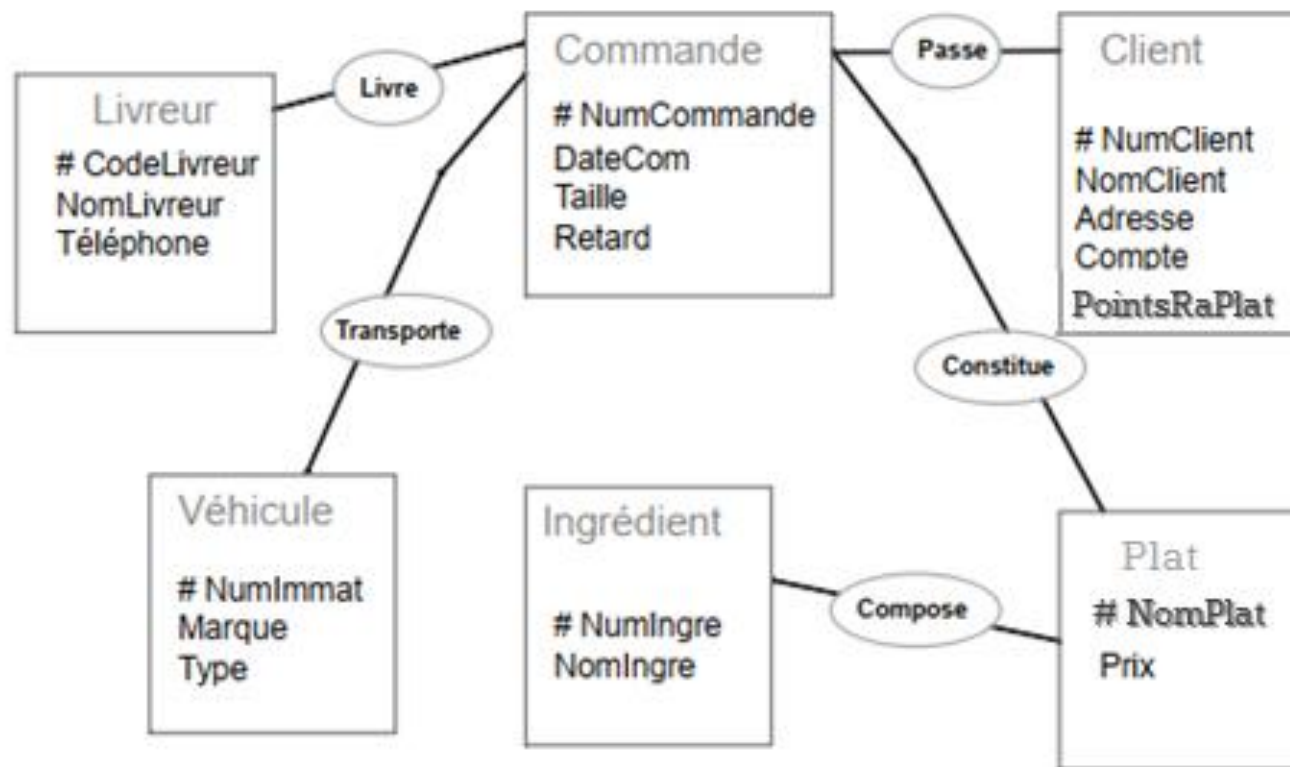
Il reste à déterminer les attributs éventuels des associations. Dans notre cas, le choix d'utiliser une entité 'commande' fait que l'on regroupe naturellement dans cette entité les attributs qui auraient pu se retrouver sur les associations.

Éventuellement, une date de commande pourrait être un attribut de l'association 'Passe' si elle était différente de la date de la commande.

Etude et réalisation

Le modèle entité-association

➤ Associations et attributs



Etude et réalisation

Le modèle entité-association

● Détermination des cardinalités

On se pose ensuite des questions sur la nature des liens entre les entités capables de déterminer les cardinalités qui seront utilisées **pour le passage au modèle relationnel**.

On doit trouver deux cardinalités, maximales et minimales, par entité associée, ce qui correspond à deux questions doubles par association.

➤ Association '**Compose**'

- ❑ **Ingrédient**: un ingrédient peut-il ne jamais être utilisé dans la composition d'un plat et peut-il être utilisé plusieurs fois ? On suppose que si un ingrédient est au catalogue, c'est qu'il est utilisé au moins une fois (1) et qu'il peut entrer dans la composition de plusieurs (n) plats. Les cardinalités associées sont de type '**1-n**'.
- ❑ **Plat**: un plat peut-il n'avoir aucun ingrédient et peut-il en avoir plusieurs ? On suppose qu'un plat est constitué d'au moins un (1) ingrédient et qu'il peut en avoir plusieurs (n). Les cardinalités associées sont de type '**1-n**'.

➤ Association '**Passe**'

- ❑ **Client**: un client peut-il n'avoir jamais passé de commandes et peut-il en avoir passé plusieurs ? Il peut y avoir une période pendant laquelle le client a approvisionné son compte mais n'a pas encore passé (0) de commande et il est évidemment encouragé à en passer plusieurs (n). Les cardinalités associées sont de type '**0-n**'.
- ❑ **Commande**: une commande peut-elle avoir été passée par aucun client ou par plusieurs ? Une commande donnée est passée par un (1) et un (1) seul client. Les cardinalités associées sont de type '**1-1**'.

Etude et réalisation

Le modèle entité-association

➤ Association '**Livre**'

- ❑ **Livreur**: un livreur peut-il n'avoir jamais livré de plat et peut-il en avoir livré plusieurs ? Un livreur a au moins effectué une (1) livraison, sinon il n'est pas considéré comme tel. On peut imaginer que l'on ne rentre les informations associées à un livreur qu'à partir du moment où il a réellement effectué une livraison. Il est supposé faire plusieurs (n) livraisons. Les cardinalités associées sont de type '**1-n**'.
- ❑ **Commande**: une commande peut-elle avoir été livrée par plusieurs livreurs ou par aucun ? Une commande est livrée par un (1) et un (1) seul livreur. Les cardinalités associées sont de type '**1-1**'.

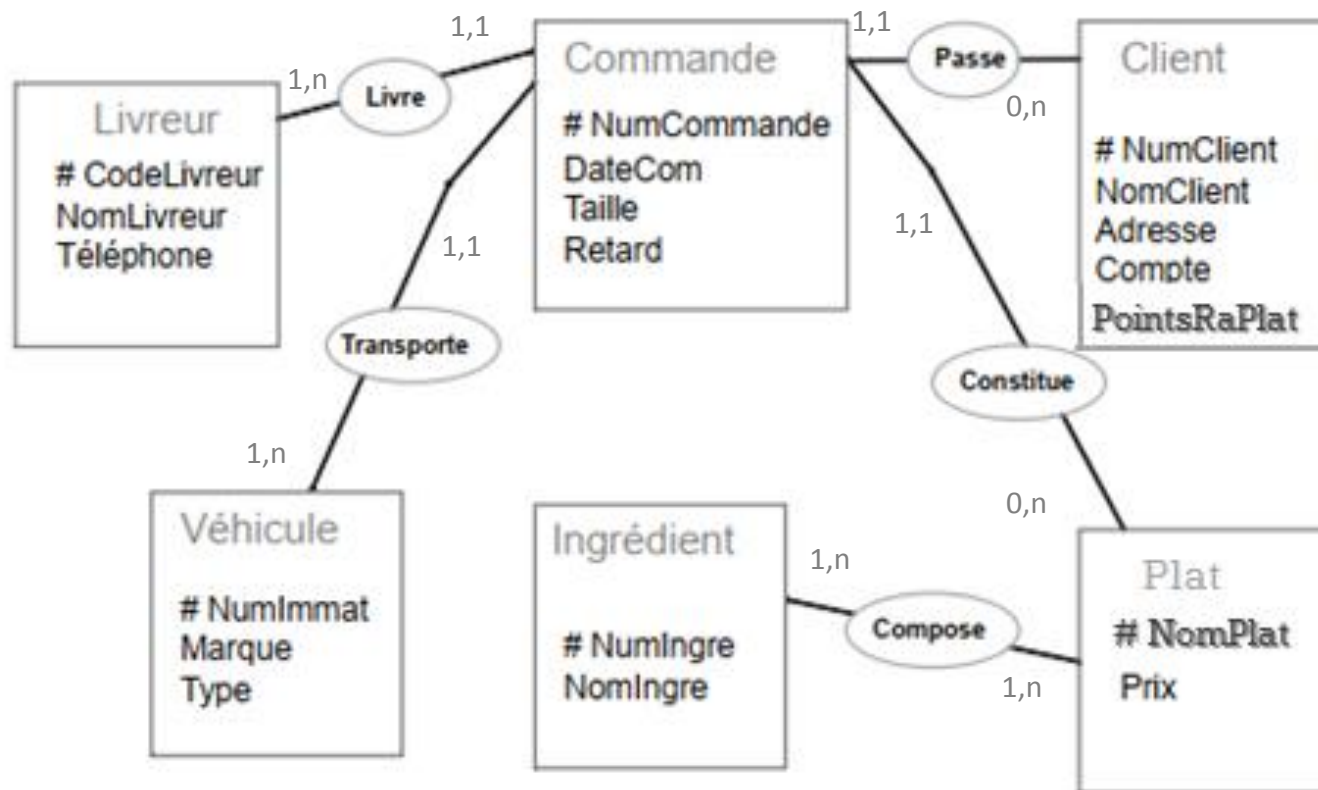
➤ Association '**Transporte**'

- ❑ **Véhicule**: un véhicule peut-il n'avoir jamais livré de plat et peut-il en avoir livré plusieurs ? Dans ce cas, la situation n'est pas tout à fait la même que pour les livreurs. On peut imaginer que les informations concernant un véhicule pourraient être insérées dans la base de données sans que le véhicule n'ait encore effectué une livraison (0). Si c'était le cas, on aurait des cardinalités de type '**0-n**'. On choisit ici des cardinalités associées de type '**1-n**'. Dans les deux cas, un véhicule est supposé être utilisé pour livrer plusieurs (n) commandes.
- ❑ **Commande**: une commande peut-elle avoir été livrée par plusieurs véhicules ou par aucun ? Une commande est livrée par un (1) et un (1) seul véhicule. Les cardinalités associées sont de type '**1-1**'.

Etude et réalisation

Le modèle entité-association

On obtient le modèle entité-association suivant :



Etude et réalisation

Passage au modèle relationnel

Dans cette étape, nous présenterons le travail préliminaire à effectué en vue du passage à *l'utilisation d'un SGBD*. Elle comprend plusieurs parties:

- la **transformation** du modèle entité-association en modèle relationnel ;
- la vérification de la **conformité** des relations créées par rapport à la définition des formes normales du modèle relationnel ;
- la discussion sur le **type des données** à adopter pour chaque champ ainsi que sur les contraintes d'intégrités à définir ;
- la **création des tables** en SQL.

● Transformation du modèle entité-association

Les deux règles générales de passage du modèle entité-association vers le modèle relationnel sont les suivantes :

- Une entité donne une relation de même clé que l'entité qui contient les mêmes attributs que l'entité.
- Une association donne une relation dont la clé est composée des deux clés des entités associées et des attributs de l'association.

Etude et réalisation

Passage au modèle relationnel

➤ Application de la règle générale

On obtient les relations suivantes à partir des entités:

- commande (**NumCommande**, DateCom, Taille, Retard) ;
- client (**NumClient**, NomClient, Adresse, Compte, PointsRaPlat) ;
- plat (**NomPlat**, Prix) ;
- livreur (**CodeLivreur**, NomLivreur, Telephone) ;
- véhicule (**NumImmat**, Marque, Type) ;
- ingrédient (**NumIngre**, NomIngredient).

En appliquant la règle générale, on obtient les relations suivantes à partir des associations:

- livre (**CodeLivreur**, **NumCommande**) ;
- transporte (**NumImmat**, **NumCommande**) ;
- passe (**NumClient**, **NumCommande**) ;
- constitue (**NomPlat**, **NumCommande**) ;
- compose (**NumIngre**, **NomPlat**).

Etude et réalisation

Passage au modèle relationnel

➤ Cas particuliers des associations de cardinalité 1-1

Un cas particulier existe lorsque l'une des cardinalités d'une association est de type '1-1'. La relation représentant l'association disparaît et fusionne avec la relation représentant l'entité associée avec la cardinalité '1-1'. On peut dire qu'il y a « aspiration » de l'association par l'entité.

Nous aurons finalement les relations suivantes:

- client (**NumClient**, NomClient, Adresse, Compte, PointsRaPlat) ;
- plat (**NomPlat**, Prix) ;
- livreur (**CodeLivreur**, NomLivreur, Telephone) ;
- véhicule (**NumImmat**, Marque, Type) ;
- ingrédient (**NumIngre**, NomIngrédient) ;
- commande(**NumCommande**, DateCom, Taille, Retard, CodeLivreur, NumImmat, NumClient, NomPlat) ;
- compose (**NumIngre**, **NomPlat**).

Etude et réalisation

Passage au modèle relationnel

● **Vérification de la conformité aux formes normales**

On doit vérifier que l'ensemble de relations créées à l'étape précédente est bien en **conformité** avec les **formes normales**. On peut se limiter en général aux trois premières formes normales.

➤ **Première forme normale**

Chaque champ doit posséder une valeur « atomique ». Ici, aucune des relations ne dispose de champs à plusieurs valeurs. On peut affirmer avec certitude qu'elles sont en première forme normale.

➤ **Deuxième forme normale**

La recherche de non-conformité à la deuxième forme normale n'a de sens que si la clé est composée de multiples champs. Dans notre cas, la seule relation qui possède une clé « composite » est la relation '**compose**'. De plus, cette relation ne contient aucun attribut qui ne fasse pas partie de la clé. La relation est en deuxième forme normale. Toutes les relations sont en deuxième forme normale.

Etude et réalisation

Passage au modèle relationnel

➤ **Troisième forme normale**

La troisième forme normale concerne les liens qui peuvent exister entre des champs qui ne font pas partie de la clé d'une relation.

Si l'on considère la relation '**client**', les champs ne faisant pas partie de la clé sont les suivants : '**NomClient**', '**Adresse**', '**Compte**', '**PointsRaPlat**'. Il est clair que le solde du compte ne permet de déterminer ni le nom du client, ni son adresse, ni son nombre de points de fidélité. Pareil pour les autres aussi. La question pourrait éventuellement se poser pour le champ '**Adresse**', susceptible dans certains cas, d'établir le nom du client si l'on suppose que deux clients n'habitent pas à la même adresse.

Suivant le même raisonnement, sur la relation '**vehicule**', on peut être amené à identifier une relation de dépendance entre le **type** qui est unique pour un véhicule. Mais sa décomposition risque de devenir complexe. On gagnera à accepter une redondance relativement faible.

On ne trouve de relations de dépendance entre aucun des champs de la relation '**commande**' qui ne font pas partie de la clé, c'est-à-dire les champs '**DateCom**', '**Taille**', '**Retard**', '**CodeLivreur**', '**NumImmat**', '**NumClient**' et '**Nomlat**'.

Toutes les relations ne sont pas strictement en troisième forme normale, mais on considère les redondances résiduelles comme acceptables.

Etude et réalisation

Passage au modèle relationnel

● Types de données et contraintes

➤ Types de données

Un type des données a pour but de faire une première restriction sur les valeurs que peut prendre un champ et surtout de spécifier les opérations et fonctions qu'il sera possible de lui appliquer.

Table client

NumClient : entier
NomClient : chaîne de caractères (obligatoire)
Adresse : chaîne de caractères
Compte : réel
PointsRaPlat: entier

Table livreur

CodeLivreur: entier
NomLivreur : chaîne de caractères (obligatoire)
Telephone: chaîne de caractères (obligatoire)

Table ingredient

NumIngred: entier
NomIngred: chaîne de caractères (obligatoire)

Table compose

NumIngre : entier
NomPlat : chaîne de caractères (obligatoire)

Table plat

NomPlat: chaîne de caractères
Prix: réel (obligatoire)

Table vehicule

NumImmat : chaîne de caractères
Marque : chaîne de caractères (obligatoire)
Type : chaîne de caractères (obligatoire)

Table commande

NumCommande: entier
DateCom : date (obligatoire)
Taille: chaîne de caractères
retard: chaîne de caractères
CodeLivreur : entier
NumImmat : chaîne de caractères
NumClient : chaîne de caractères
NomPlat : chaîne de caractères

Etude et réalisation

Passage au modèle relationnel

➤ Contraintes d'intégrité

Les contraintes permettent de décrire de manière plus précise les ensembles auxquels appartiennent les champs.

- Le champ '**Prix**' de la table '**plat**' peut être limité à l'intervalle **500 .. 2500**.
- On pourrait définir un intervalle de validité pour les dates de commande (champ '**DateCom**' de la table '**commande**'), par exemple la date de commande doit être supérieure ou égale à la date du jour.
- Le champ '**Retard**' de la table '**commande**' doit contenir les valeurs '**O**' ou '**N**'.
- Le champ '**Taille**' de la table '**commande**' doit contenir les valeurs « **normale** », « **large** », « **normale amélioré** » ou « **large amélioré** ».
- Les champs '**NomPlat**' dans les autres champs constituent une contrainte de référence faisant référence au champ '**NomPlat**' de la table '**plat**'.
- '**CodeLivreur**' de la relation '**commande**' fait référence au champ '**CodeLivreur**' de la relation '**livreur**'.
- '**NumImmat**' de la relation '**commande**' fait référence au champ '**NumImmat**' de la relation '**vehicule**'.
- '**NumClient**' de la relation '**commande**' fait référence au champ '**NumClient**' de la relation '**client**'.

Etude et réalisation

Passage au modèle relationnel

● Création des tables

À la création des tables, on spécifie le nom des champs, leur type, les contraintes d'intégrité et la définition des clés.

Table 'client'

```
CREATE TABLE client (  
  NumClient INT PRIMARY KEY,  
  NomClient VARCHAR(20) NOT NULL,  
  Adresse VARCHAR(150) ,  
  Compte REAL,  
  PointsRaPlat INT  
);
```

Table 'livreur'

```
CREATE TABLE livreur (  
  CodeLivreur INT PRIMARY KEY,  
  NomLivreur CHAR(30) NOT NULL,  
  TeleLivreur CHAR(30) NOT NULL  
);
```

Table 'plat'

```
CREATE TABLE plat(  
  NomPlat CHAR(30) PRIMARY KEY,  
  Prix CHAR(30) NOT NULL,  
  CHECK (Prix BETWEEN 500 AND 2500)  
);
```

Table 'ingrédient'

```
CREATE TABLE ingredient(  
  NumIngréd INT PRIMARY KEY ,  
  NomIngréd CHAR(30) NOT NULL  
);
```

Table 'vehicule'

```
CREATE TABLE vehicule(  
  NumImmat CHAR(30) PRIMARY KEY,  
  Marque CHAR(30) NOT NULL,  
  Type CHAR(30) NOT NULL  
);
```


Etude et réalisation

Passage au modèle relationnel

Table 'compose'

```
CREATE TABLE compose (  
  NomPlat CHAR(30) REFERENCES plat(NomPlat),  
  NumIngred INT REFERENCES ingredient(NumIngred),  
  PRIMARY KEY (NomPlat, NumIngred)  
);
```

Table 'commande'

```
CREATE TABLE commande (  
  NumCommande INT PRIMARY KEY,  
  DateCom DATE,  
  Taille CHAR(30) NOT NULL,  
  Retard CHAR(1) NOT NULL,  
  NumClient INT REFERENCES client(NumClient),  
  NomPlat CHAR(30) REFERENCES plat(NomPlat),  
  CodeLivreur INT REFERENCES livreur(CodeLivreur),  
  NumImmat CHAR(30) REFERENCES vehicule(NumImmat),  
  CHECK Taille in ('normale','large','normale ameliore','large ameliore'),  
  CHECK Retard in ('O','N')  
);
```

FIN

Feedback: pape.abdoutaye.barro@gmail.com