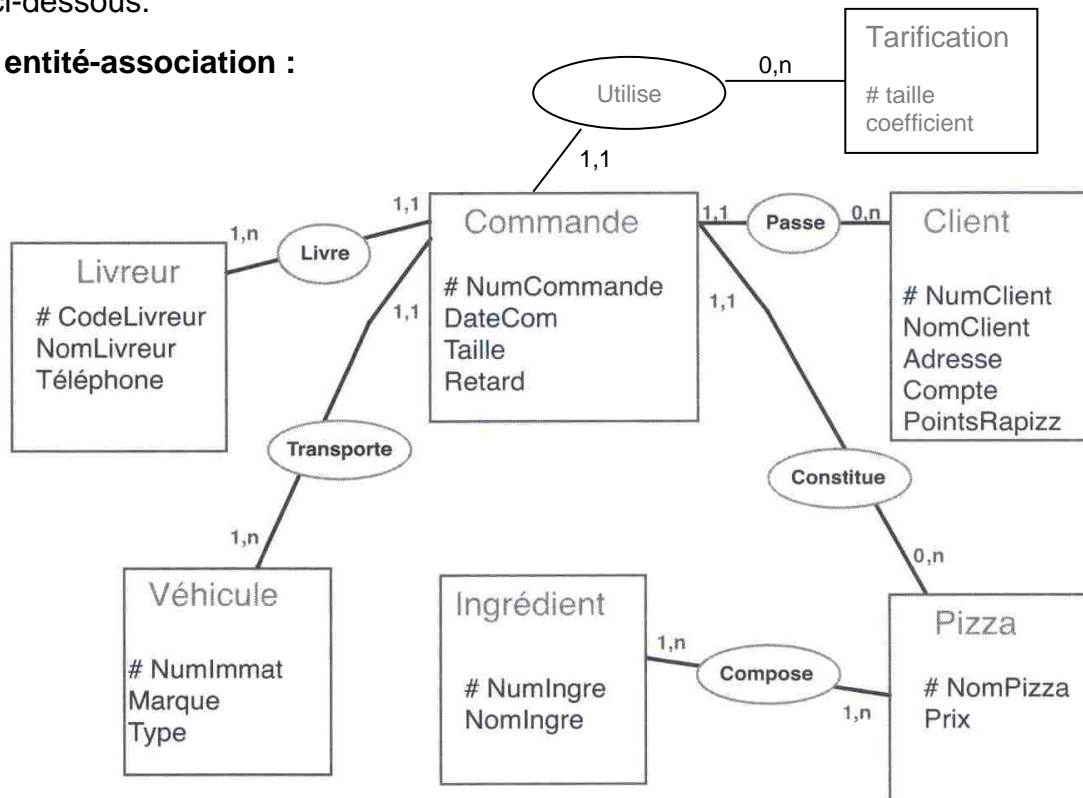


Présentation :

Soit la base de données gestion_pizza dont le modèle entité-association et modèle relationnel sont décrits ci-dessous.

Modèle entité-association :



Modèle relationnel :

client (NumClient, NomClient, Adresse, Compte, PointsRaPizz) ;
pizza (NomPizza, Prix) ;
livreur (CodeLivreur, Nom Livreur, Telephone) ;
véhicule (NumImmat, Marque, Type) ;
ingrédient (NumIngre, NomIngrédient) ;
commande (NumCommande, DateCom, Taille, Retard, CodeLivreur, NumImmat, NumClient, NomPizza) ;
compose (NumIngre, NomPizza) ;
Tarification (Taille, coefficient).

On distingue trois grandes catégories d'utilisateurs :

- Les clients : Ils peuvent consulter les informations les concernant ainsi que le catalogue des pizzas disponibles avec leur prix et leur composition.
- Les employés : Ils sont concernés par l'activité courante, qui par conséquent va générer les mises à jour les plus fréquentes, est la commande de pizzas. Ainsi ils mettent à jour les données de la table 'commande', mais également celles de la table 'client' pour créer de nouveaux clients le cas échéant. Ils ont accès en lecture à toutes les données des autres tables.
- Les gestionnaires : Ils doivent pouvoir mettre à jour toutes les informations de gestion: en particulier, celles sur les pizzas, les clients, les livreurs et les voitures utilisées. En revanche, on peut considérer qu'ils n'ont pas besoin de modifier les informations concernant les commandes, mais qu'ils doivent pouvoir y accéder en lecture.

Question 1.

Utiliser le modèle ci-dessous pour faire la synthèse des droits par catégories de la BD pizza

table	commande	pizza	compose	ingrédient	Tarification	client	livreur	véhicule
utilisateur								
client	X							
employé	Lec/Ecr							
gestionnaire	Lec							

Question 2.

Cette entreprise gère une centaine de clients {c₁.. c₁₀₀}, les employés sont {e₁.. e₁₀} et le gestionnaire sont {g₁.. g₅}

A/ Donner les requêtes SQL permettant de créer les comptes précédents.

B/ Donner les requêtes SQL permettant de réaliser les droits d'accès étudiés dans la question 1.

Indication : Utiliser la notion de rôle (voir annexe) :

```
create role client, ... , grant select on pizza to client, ..., grant client
to cl, ... ;
```

Question 3.

De nouveaux utilisateurs aux besoins plus spécifiques vont utiliser la BD pizza :

- Le service de comptabilité {Michel, Laurent } doit rémunérer les livreurs, il a besoin le nombre de commande livrée.
- Le service du personnel { Thiry, Yves, Armelle } effectue un suivi des livreurs, il a besoin le nombre de retard de chacun et également le nombre de commande livrés.
- Un laboratoire de recherche { Jacques, Micheline } en sociologie effectue des études sur la clientèle, (relation entre les ingrédients et l'adresse des clients)

Ces utilisateurs ne doivent accéder qu'aux champs dont ils ont strictement besoin. Etudier une solution.

Indication : Utiliser les vues, par exemple :

```
creat view retard_livreur as ...
grant select on retrad_livreur to Thiry
```

Pour grouper l'attribution des privilèges on a vu que c'est préférable d'utiliser les rôles.

Annexe

Il peut devenir problématique s'il faut donner plusieurs privilèges à plusieurs utilisateurs. La notion de `role` peut faire face à ce problème. La commande pour créer un rôle est :

```
▪ CREATE ROLE nomderole;
```

Une fois ce rôle créé, il vous suffit de lui donner des privilèges (sur objets ou bien systèmes) :

- ```
▪ GRANT SELECT,INSERT,UPDATE ON table1 TO nomderole;
▪ GRANT nomderole TO user1, user2, user3;

▪ GRANT CREATE TABLE TO nomderole;
▪ CREATE USER nomuser IDENTIFIED BY passé ROLE nomderole;
▪ ALTER USER nomuser DEFAULT ROLE nomderole;
```