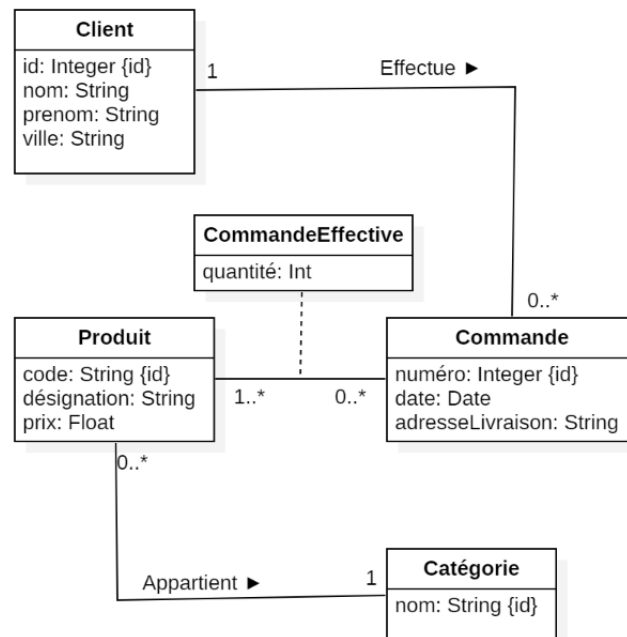


## Examen INF403 - Mai 2022

Durée : 2 Heures / Document autorisé : une feuille A4, recto-verso manuscrite

### 1 UML - Relationnel

Le diagramme UML ci-dessous modélise une première version du système d'information pour gérer un site d'e-commerce. L'état actuel de la BD est fourni en annexe.



Un début de traduction en relationnel est le suivant :

**Categories** (nom\_categorie)

**Produits** (code\_produit, designation\_produit, prix\_produit, nom\_categorie)

**Clients** (id\_client, nom\_client, prenom\_client, ville\_client)

**Commandes** (numero\_commande, date\_commande, adresse\_livraison\_commande, id\_client)

**CommandesEffectives** (numero\_commande, code\_produit, quantite\_commande\_effective)

Les contraintes d'intégrité référentielle :

Produits[nom\_categorie]  $\subseteq$  Categories[nom\_categorie]

Commandes[id\_client]  $\subseteq$  Clients[id\_client]

CommandesEffectives[numero\_commande] = Commandes[numero\_commande]

CommandesEffectives[code\_produit]  $\subseteq$  Produits[code\_produit]

## 2 Compréhension de modèles

### Question 1 (1,5 points) :

Dire si les affirmations suivantes sont vraies ou fausses en s'appuyant sur le modèle UML fourni :

1. On ne peut pas avoir deux clients avec le même nom et prénom (FAUX)
2. Un client effectue forcément des commandes (FAUX)
3. Une commande est liée à un seul client (VRAIS)
4. Le prix d'un produit peut changer en fonction de la commande (FAUX)
5. Un produit est forcément lié à une catégorie (VRAIS)
6. Un produit peut être commandé plusieurs fois (VRAIS)

### Question 2 (1.5 points) :

Expliquer pourquoi chacune de ces propositions n'est pas appropriée par rapport au modèle UML.

1. **Clients** (id\_client, nom\_client, prenom\_client, ville\_client, numero\_commande)  
*Inapproprié, car ici, un client peut juste faire une commande. Par exemple, le client 1, peut apparaitre une seule fois dans la table car id\_client est la clé.*
2. **CommandesEffectives** (numero\_commande, code\_produit, quantite\_commande\_effective)  
*Inapproprié, car le fait de définir deux clés (au lieu d'une clé composée) fait qu'une commande puisse être liée à un seul produit, et vice-versa, un produit peut être lié à qu'une seule commande.*

## 3 Compréhension de requêtes en SQL

### Question 3 (3 points) :

En considérant les relations fournies en annexe, donner le **résultat retourné** par chacune des requêtes ci-dessous (considérer le système Oracle vu en TP) ainsi qu'une **phrase qui décrit le but de la requête** (si elle est correcte). Il peut s'agir d'une erreur, ou de n-uplets affichés sous forme de tableau.

1. 

```
SELECT MIN(quantite_commande_effective)
FROM CommandesEffectives
WHERE numero_commande = 300;
```

*Pour la commande numéro 300 donner la quantité effective minimum.*

```
MIN(quantite_commande_effective)
```

```
-----
1
```

2. 

```
SELECT code_produit, designation_produit
FROM Produits
JOIN CommandesEffectives
USING (code_produit)
WHERE nom_categorie = "alimentaire";
```

*Donner les produits alimentaires (code et désignation) qui ont été vendus au moins une fois. Manque un DISTINCT pour faire la différence entre les deux achats de AS273 (on aurait pu aussi afficher la quantité afin de faire la différence).*

```
code_produit | designation_produit
```

```
-----
```

```
AF824      | Jus d'orange 1l
AS273      | Figues sèches 100g
AS273      | Figues sèches 100g
AS729      | Pâtes fraîches 500g
```

3. 

```
WITH R AS (
  SELECT code_produit,
         SUM(quantite_commande_effective) AS x
  FROM CommandesEffectives
  GROUP BY code_produit)
SELECT code_produit
FROM R
GROUP BY code_produit
HAVING x IN
      (SELECT MAX(x) FROM R);
```

*Donner le code du produit qui a été le plus vendu (cela prend la quantité en compte).*

```
code_produit
```

```
-----
```

```
AF824
```

## 4 Expression de requêtes en SQL

### Question 4 (6 points) :

Exprimer en SQL les requêtes ci-dessous. Les requêtes devront construire des résultats sans répétition de valeurs, la clause `DISTINCT` ne sera utilisée que lorsque nécessaire.

1. Donner le code, désignation et prix des livres qui n'ont jamais été vendus.

```
SELECT code_produit, designation_produit, prix_produit
FROM Produits
WHERE nom_categorie = 'livres' AND
      code_produit NOT IN (SELECT code_produit
                          FROM CommandesEffectives);
```

2. Donner le code et la désignation des produits avec un prix entre 2 et 5 (les deux valeurs incluses) et qui ont été vendus au moins deux fois, pas forcément sur commandes différentes.

```
SELECT code_produit, designation_produit
FROM Produits JOIN CommandesEffectives USING (code_produit)
WHERE prix_produit >= 2 AND prix_produit <= 5
GROUP BY code_produit, designation_produit
HAVING SUM(quantite_commande_effective) > 1;
```

3. Donner les couples de clients (`id_client1` et `id_client2`) qui ont acheté le même produit et habitent dans la même ville.

```
WITH ClientsCommandes AS (
  SELECT id_client,
         code_produit,
         ville_client
  FROM Commandes JOIN CommandesEffectives USING (numero_commande)
              JOIN Clients USING (id_client)
)
SELECT DISTINCT C1.id_client AS id_client1, C2.id_client AS id_client2
FROM ClientsCommandes C1 JOIN ClientsCommandes C2 ON
  (C1.code_produit = C2.code_produit
   AND C1.id_client < C2.id_client
   AND C1.ville_client = C2.ville_client);
```

4. Donner le code du produit le plus vendu par catégorie. Schéma attendu : (`nom_categorie`, `code_produit`)

```
WITH ProduitsQuantites AS (
  SELECT code_produit, nom_categorie, SUM(quantite_commande_effective) AS qte
  FROM CommandesEffectives JOIN Produits USING (code_produit)
  GROUP BY code_produit, nom_categorie),
MaxCategorie AS (
  SELECT nom_categorie, MAX(qte) AS maxQte
  FROM ProduitsQuantites
  GROUP BY nom_categorie
)
SELECT P.nom_categorie, P.code_produit
FROM ProduitsQuantites P JOIN MaxCategorie M ON (
  P.nom_categorie = M.nom_categorie AND P.qte = M.maxQte);
```

**Question 5 (1 point) :**

Donner le code SQL permettant d'éliminer le produit AS273. On suppose que toutes les foreign keys ont été codées en comportement RESTRICT.

*Le comportement restrict ne permet pas de faire un DELETE directement si le produit est dans une commande. Il faut donc éliminer d'abord le produit AS273 de la table CommandeEffectives. Comme dans les commandes où il se trouve le produit concerné il y a d'autres produits (100 et 300), alors on a pas besoin d'éliminer ces dernières afin de toujours respecter les contraintes d'intégrité.*

```
DELETE FROM CommandesEffectives WHERE code_produit = 'AS273';
DELETE FROM Produits WHERE code_produit = 'AS273';
```

## 5 Création d'une vue

**Question 6 (3 points) :**

Proposer le code SQL de la vue *CommandesAvecPrixTotal*. Cette dernière fournit les informations relatives aux commandes avec son prix total (qui se calcule à partir des prix de chaque produit et la quantité choisie). **Fournir le résultat en forme de tableau et Expliquer l'intérêt d'une vue.**

```
CREATE VIEW CommandesAvecPrixTotal (
    numero_commande,
    date_commande,
    adresse_livraison_commande,
    id_client,
    prix_total_commande) AS
SELECT numero_commande,
    date_commande,
    adresse_livraison_commande,
    id_client,
    SUM(quantite_commande_effective * prix_produit) AS prix_total_commande
FROM Commandes JOIN CommandesEffectives USING (numero_commande)
    JOIN Produits USING (code_produit)
GROUP BY numero_commande, date_commande, adresse_livraison_commande, id_client;
```

*Dans ce cas, l'intérêt de la vue est d'étendre une relation de base (Commandes) avec les attributs calculés (prix\_total\_commande) à partir d'autres attributs de la base. Les valeurs calculés seront mis à jour au moment où on interroge la vue via des requêtes.*

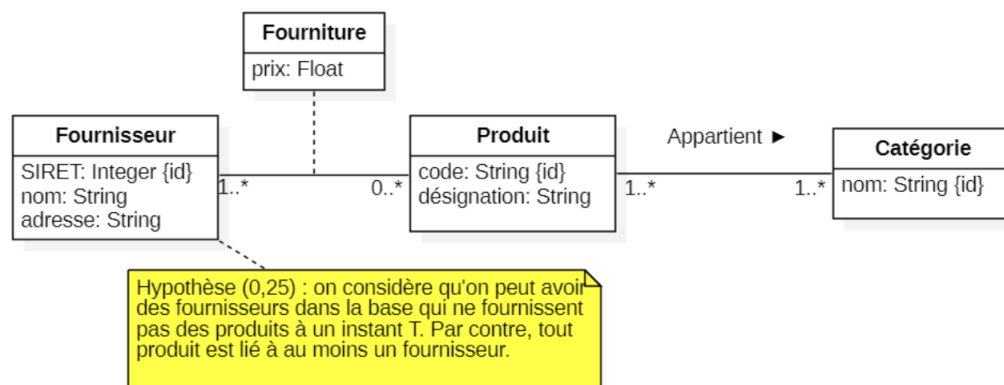
numero_commande	date_commande	adresse_livraison_commande	id_client	prix_total_commande
Filtre	Filtre	Filtre	Filtre	Filtre
100	2022-01-26	1 cours Gaston Planté, Marseille	1	9.5
200	2022-03-20	49 avenue Enrico Fermi, Nantes	2	25
300	2022-03-20	32 boulevard Pavel Tcherenkov, Lille	2	18.7

## 6 Modélisation

**Question 7 (2 points) :**

Proposer une extension du modèle UML avec les caractéristiques suivantes. Modéliser seulement les parties concernées par l'extension.

- Un produit peut appartenir à plusieurs catégories (au moins une) et une catégorie doit être liée à au moins un produit.
- On introduit la notion de fournisseur, identifié par un numéro SIRET. On garde le nom et une adresse. Un fournisseur fournit plusieurs produits et un produit peut être fourni par plusieurs fournisseurs.
- Le code et la désignation du produit ne changent pas par rapport au fournisseur. Par contre, le prix dépend de ce dernier.



### Question 8 (2 points) :

Proposer une traduction en relationnel à partir du modèle UML proposé dans la question précédente. Donner toutes les possibles contraintes sauf celles des domaines.

**Produits** (code\_produit, designation\_produit)

**ProduitAppartientCategories** (nom\_categorie, code\_produit) /\* clé composé, pas deux clés \*/

**Fournisseurs** (siret\_fournisseur, nom\_fournisseur, adresse\_fournisseur)

**Fournitures** (code\_produit, siret\_fournisseur, prix\_fourniture) /\* clé composé aussi \*/

Les contraintes d'intégrité référentielle :

ProduitAppartientCategories[code\_produit] = Produits[code\_produit] /\* égalité à cause du (1..\*) \*/

ProduitAppartientCategories[nom\_categorie] = Categories[nom\_categorie] /\* égalité à cause du (1..\*) \*/

Fournitures[code\_produit] = Produits[code\_produit] /\* égalité à cause du (1..\*) \*/

Fournitures[siret\_fournisseur]  $\subseteq$  Fournisseurs[siret\_fournisseur]

## 7 Annexe

Clients				Categories
id_client	nom_client	prenom_client	ville_client	nom_categorie
1	Laporte	François	Nantes	alimentaire
2	Kai	Eva	Paris	livres
3	Fuentes	Noemi	Lille	jeux

Produits			
code_produit	designation_produit	prix_produit	nom_categorie
AS729	Pâtes fraîches 500g	1.2	alimentaire
AS273	Figues sèches 100g	2.0	alimentaire
AF824	Jus d'orange 1l	1.5	alimentaire
AS829	Pois chiches 250g	0.9	alimentaire
LS938	L'Attrape-Coeur	10.0	livres
LC328	Le Rouge et le Noir	15.0	livres
JK023	Jeu d'échecs en bois	13.5	jeux

CommandesEffectives		
numero_commande	code_produit	quantite_commande_effective
100	AS273	1
100	AF824	5
200	LC328	1
200	LS938	1
300	JK023	1
300	AS729	1
300	AS273	2

Commandes			
numero_commmmande	date_commande	adresse_livraison_commande	id_client
100	2022-01-26	1 cours Gaston Planté, Marseille	1
200	2022-03-20	49 avenue Enrico Fermi, Nantes	2
300	2022-03-20	32 boulevard Pavel Tcherenkov, Lille	2