
Interrogation d'informatique

2^{ème} année - Novembre 2019-2020



Durée totale : 60mn
Documents autorisés : Aucun.

- Le barème est indicatif et sur 20 points.
- Le sujet est sur 7 pages - il y a deux exercices.

Exercice 1 Requêtes SQL (13 pts)

On considère la base de données d'une entreprise qui exporte des produits français (type vins, fromages, etc) aux Etats-Unis. On considère les produits vendus, les entrepôts où ils sont stockés, et les transporteurs en charge de les faire transiter.

Dans le schéma relationnel correspondant, les attributs formant la clé sont soulignés. On suppose qu'il n'y a pas de valeurs indéfinies (aucune valeur à *NULL*). On considère que la base de données est cohérente et automatiquement mise à jour pour la position des produits.

- `produits(idProduit int(11), denomination varchar(20), type varchar(20), poidsUnitaire int(10))`
Un produit est identifié de manière unique par un identifiant. On mémorise son identifiant, sa dénomination (par exemple, "comté 18 mois AOC"), son type (vin, fromage, pâtisserie, etc), et son poids unitaire.
- `entrepots(idEntrepot varchar(20), adresse varchar(40), telephone varchar(20))`
L'entreprise dispose de plusieurs entrepôts. Lorsqu'un produit n'est pas en transit, il est stocké dans un entrepôt. Chaque entrepôt est identifié de manière unique par son identifiant. On mémorise également son adresse et son numéro de téléphone
- `stock(idProduit int(11), idEntrepot varchar(20), nombreUnites int(10))`
Cette relation mémorise dans quels entrepôts sont stockés quel(s) produits. L'attribut `idProduit` est une clé étrangère référençant la relation `produits`. `idEntrepot` est une clé étrangère référençant la relation `entrepots`. On mémorise également le nombre d'unités du produit présentes dans l'entrepôt..
- `transporteur(idTransporteur varchar(20), type varchar(40), paysSiegeSocial varchar(20))`
Lorsqu'un produit est en transit, il est entre les mains d'un transporteur, identifié de manière unique par un identifiant. On mémorise également le type du transporteur (aérien, ferroviaire, maritime, etc), ainsi que le pays du siège social du transporteur.
- `transit(idTransit int(11), idProduit int(11), idTransporteur varchar(20), idDepart varchar(20), idArrivee varchar(20), dateDeb date, nombreUnites int(10))`
Cette relation mémorise quels sont les produits en transit. Un transit est identifié de manière unique par un identifiant. L'attribut `idProduit` est une clé étrangère référençant la relation `produits`. `idTransporteur`, qui indique le transporteur en charge, est une clé étrangère référençant la relation `transporteur`. Les attributs `idDepart` et `idArrivee` sont des clés étrangères faisant référence à la relation `entrepots`, et qui désignent les entrepôts d'où partent et où arrivent le produit en transit. On mémorise également la date de début du transit, ainsi que le nombre d'unités de produit en transit. Lorsque le produit a atteint l'entrepôt d'arrivée, il n'est plus en transit et la ligne est supprimée de la table.

Question : Écrivez les requêtes SQL correspondant aux recherches suivantes. Prenez soin

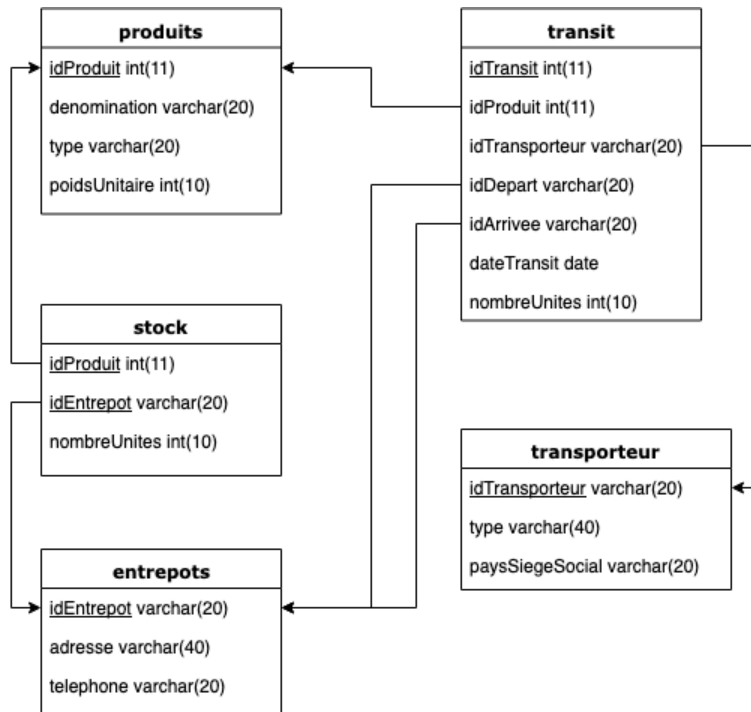


FIGURE 1 – Vue graphique du schéma relationnel de la base.

d'éliminer les doublons lorsque le résultat peut en comporter. **Les questions ne sont pas classées par ordre de difficulté.**

1. Quels sont les entrepôts dont l'adresse comprend le code postal 69100 ? Afficher tous les attributs. Indication : Utilisez une condition avec le mot-clé `like`, sachant que `%` désigne n'importe quelle chaîne.

```

select *
from entrepots
where adresse like '%69100%'
0,5 pts
  
```

2. Combien y a-t-il de transporteurs américains qui ont effectué au moins un transport ? Un transporteur est considéré comme américain si le pays de son siège social est 'USA'.

Attention, c'est plutôt "qui effectuent un transport"
usage du distinct indispensable car on veut compter les transporteurs et non les transports

```

select count(distinct i.idTransporteur)
from transit i, transporteur t
where i.idTransporteur = t.idTransporteur
and paysSiegeSocial = 'USA'
1pt
  
```

3. Donner le nombre total d'unités de chaque produit stocké dans l'entrepôt ayant l'identifiant 'MARSEILLE13'.

```

select idProduit, nombreUnites
from stock
where idEntrepot = 'MARSEILLE13'
0,5 pts
  
```

-
4. Quelles sont les dénominations des produits de type 'Vin' stockés dans l'entrepôt 'CHICAGO11' ?

```
select distinct denomination  
from stock s, produits p  
where s.idProduit = p.idProduit  
and idEntrepot = 'CHICAGO11'  
and type = 'Vin'  
1,5 pt
```

5. Quel est le poids total de l'ensemble des produits dénommés 'Camembert fermier' stockés dans l'entrepôt 'DALLAS02' ? Indication : prenez en compte le poids unitaire d'un produit et le nombre d'unités en stock.

```
select sum(poidsUnitaire*nombreUnites)  
from stock s, produits p  
where s.idProduit=p.idProduit  
and idEntrepot='DALLAS02'  
and denomination = 'Camembert fermier'  
1pt
```

6. Quels sont les produits dont le transit est assuré uniquement par des transporteurs dont le type est 'aerien' ? Afficher tous les attributs des produits concernés.

Le principal problème est d'assurer le "uniquement" : il faut s'assurer qu'un produit sélectionné n'appartient pas à l'ensemble des produits qui sont transportés au moins une fois par un transport autre que aerien.

```
select *  
from produits  
where idProduit not in (select distinct idProduit from transit t, transporteur o  
where t.transporteur=o.transporteur and type <> 'aerien')  
2 pts
```

7. Donner, par produit, le nombre moyen d'unités en stock.

```
select idProduit, avg(nombreUnites)  
from stock  
group by idProduit  
1 pt
```

8. Quels sont les produits de type fromage dont le nombre d'unités stockées est inférieur à la moyenne du nombre d'unités de fromage stockées. Cette moyenne est calculée pour les fromages, tous entrepôts confondus.

```
select s.idProduit, sum(nombreUnites)  
from produits p, stock s  
where p.idProduit = s.idProduit  
and type = 'fromage'  
group by s.idProduit  
having sum(nombreUnites)<(select avg(sum(nombreUnites))  
from produits p, stock s
```

*where p.idProduit = s.idProduit
and type = 'fromage'
group by s.idProduit)
2 pts*

9. Quels sont les fromages qui ont commencé leur transit à la même date qu'un fromage ayant la dénomination **Brebis basque** ? Il s'agit d'une auto-jointure.

*select distinct t1.idProduit
from transit t1, transit t2, produits p1, produits p2
where t1.idProduit=p1.idProduit
and p1.type = 'fromage'
and t2.idProduit = p2.idProduit
and p2.type = 'fromage' /*ce test n'est pas indispensable si on suppose que la d
énomination 'brebis basque' ne peut concerner que le type fromage */
and p2.denomination = 'Brebis basque'
and t1.dateDeb=t2.dateDeb
2 pts*

10. Combien chaque transporteur assure-t-il de transit de chaque produit ? Afficher l'identifiant de transporteur, le produit et le nombre de transits correspondants.

select idTransporteur, idProduit, count()
from transit
group by idTransporteur, idProduit
1,5 pt*

Exercice 2 Modèle conceptuel et modèle relationnel (7 pts)

(Q2.1) Retro-conception

Question : Proposez un modèle entité-association, utilisant la syntaxe UML, qui, transformé par les règles utilisées en cours et en TD, produit le modèle relationnel précédent.

3 pts + bonus 0,25 - Voir proposition de la figure 2.

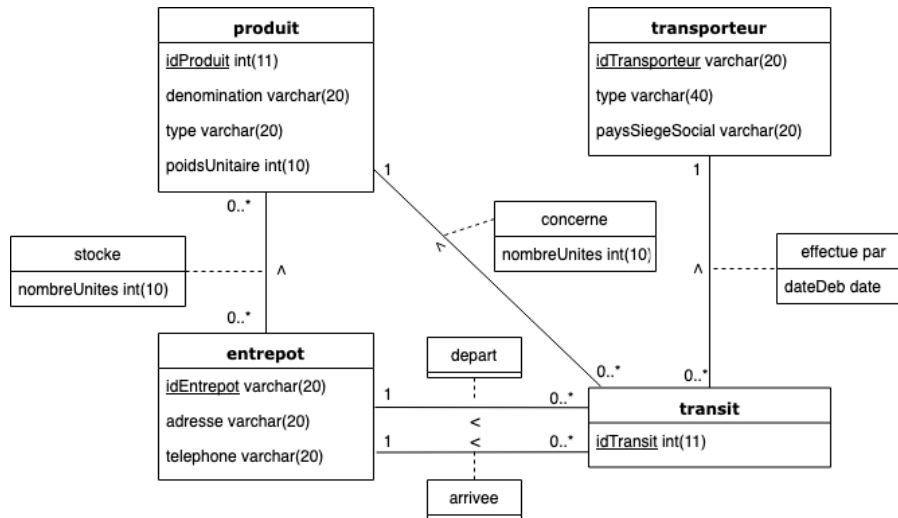


FIGURE 2 – Modèle entité/association, syntaxe UML, compatible avec le modèle relationnel de l'exercice 1

Les cardinalités 1 indiquées sont obligatoires

Les cardinalités 0.. proches de transit sont aussi obligatoires, car transit ne mémorise que les transit en cours. Il y a donc des produits, transporteurs et entrepôts qui peuvent n'être concernés par aucun transit à moment donné.*

(Q2.2) Extension du modèle entité-association - partie 1

On souhaite étendre la base de données pour mémoriser les organismes qui certifient les produits et quel organisme a certifié quel produit. L'entreprise mémorise tous les organismes certificateurs existant, qu'ils aient certifié des produits de l'entreprise ou non. Les informations mémorisées sont : le numéro siren de l'organisme, son nom, l'adresse de son siège social, et sa nationalité. Sa politique impose qu'un produit est mémorisé dans la base uniquement s'il est certifié par au moins deux organismes certificateurs.

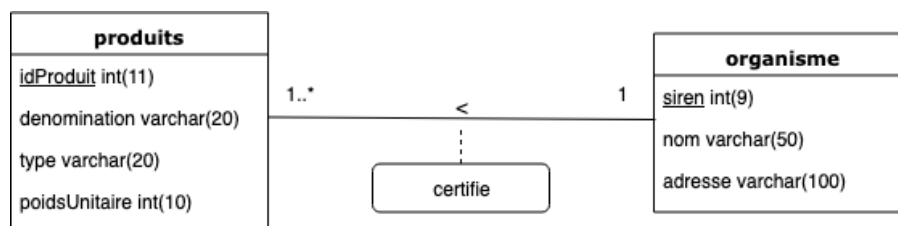


FIGURE 3 – Modèle entité/association, syntaxe UML, partie sur la certification

Question : Selon vous, le schéma de la figure 3 est-il cohérent avec les informations précédentes ?

Si ce n'est pas le cas, expliquez pourquoi et proposez une modification.

1 pt + bonus 0,25

Les éléments qui posent pb :

- Il manque l'attribut nationalite à l'entité organisme*
- La cardinalité côté produit n'est pas bonne, car l'entreprise mémorise tous les organismes, qu'ils aient ou non certifié des produits de l'entreprise. Il se peut donc qu'un organisme n'ait certifié aucun produit de l'entreprise. Donc cardinalité 0..* :*

- La cardinalité côté organisme n'est pas correcte car un produit est mémorisé s'il est certifié par au moins 2 organismes. Le cardinalité min est donc 2 ; rien n'empêche un produit d'être certifié par plus de 2 organismes. Donc 2..* :*

(Q2.3) Extension du modèle entité-association - partie 2

On souhaite étendre la base de données pour mémoriser les achats de produits effectués par l'entreprise. Certains des employés de l'entreprise sont spécialisés dans l'achat des produits et nommés acheteurs. Un achat est identifié par un numéro d'achat. Il est effectué par un acheteur et un seul, à une date donnée. Un acheteur peut effectuer plusieurs achats. Un achat est effectué auprès d'un fournisseur et un seul, chez qui l'entreprise peut se fournir plusieurs fois. Lors d'un achat, le fournisseur et l'acheteur se mettent d'accord sur les produits achetés, avec, pour chaque produit, un prix unitaire d'achat et un nombre d'unités achetées. Un achat peut concerner des produits différents.

Question : Proposez un modèle entité-association (syntaxe UML), qui prend en compte ces informations. Expliquez brièvement le choix de vos cardinalités, et indiquez si nécessaire les hypothèses faites pour définir les cardinalités (et qui ne sont pas en contradiction avec ce qui est écrit dans le texte). **Il n'est pas utile de recopier l'ensemble du modèle de la question précédente. Recopiez uniquement l'entité produit.**

2pts - Voir la proposition de la figure 4

Les cardinalités 1.. de comprend ne peuvent être 0..* : un achat comprend au moins un produit et un produit est forcément acheté au moins une fois.*

Les cardinalités 1 ne peuvent être remplacées par 0..1 ni une cardinalité multiple.

(Q2.4) Passage au modèle relationnel

Question : Transformez votre modèle entité/association de la question précédente (Q2.3) en modèle relationnel. Ne transformez pas la partie sur la certification.

1pt - En plus de la relation produit :

- fournisseur(idFournisseur int(11), nomFournisseur varchar(50))*
- acheteur(idAcheteur int(11), nomAcheteur varchar(60), tel varchar(10))*
- achat(numAchat int(11), idFournisseur int(11), idAcheteur int(11)) où idFournisseur est une clé étrangère référençant fournisseur et idAcheteur est une clé étrangère référençant Acheteur.*

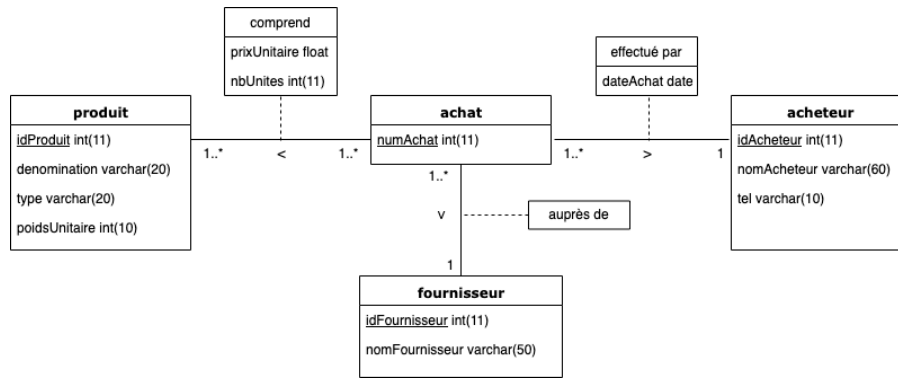


FIGURE 4 – Modèle entité/association, syntaxe UML, extension aux achats et aux fournisseurs.

— *compoAchat(numAchat int(11), idProduit int(11), prixUnitaire float, nbUnites int(11))*
 où *numAchat* est une clé étrangère référençant *achat* et *idProduit* est une clé étrangère référençant *produit*.