

---

# Interrogation d'informatique

## 2<sup>ème</sup> année - Décembre 2020-2021



**Durée totale :** 30mn  
**Documents autorisés :** Toutes notes personnelles ou du cours.

---

- Le barème est sur 20 points.
- Le sujet est sur 4 pages - il y a un exercice.

### Exercice 1 Attestation sur l'honneur

Sur papier libre, écrivez de manière manuscrite le texte suivant :

« **J'atteste sur l'honneur que je ne tricherai pas, notamment en respectant les consignes données dans le sujet et en n'ayant aucun échange avec une tierce personne autre que l'enseignant qui accompagne l'évaluation** ».

Signez ce document.

### Exercice 2 Requêtes SQL

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, les transports en cours et leur cargaison, les transporteurs assurant les transports.

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*). La base de données est cohérente et automatiquement mise à jour pour la position des produits.

- `produit(idProduit int(11), denomination varchar(20), typeProd 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.
- `entrepot(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 quels produits. L'attribut `idProduit` est une clé étrangère référençant la relation `produit`. `idEntrepot` est une clé étrangère référençant la relation `entrepot`. On mémorise également le nombre d'unités du produit présentes dans l'entrepôt.
- `transporteur(idTransporteur varchar(20), typeTrans varchar(40), paysSiegeSocial varchar(20))`  
Chaque transporteur est identifié de manière unique par un identifiant. On mémorise également son type (aérien, ferroviaire, maritime, etc), et le pays de son siège social.
- `transport(idTransport int(11), idTransporteur varchar(20), dateDeb date, entrOrigine varchar(20), entrDestination varchar(20))`  
Cette relation mémorise les transports en cours. Un transport est identifié de manière unique par un identifiant. L'attribut `idTransporteur`, qui indique le transporteur en charge, est une clé étrangère référençant la relation `transporteur`. Les attributs `entrOrigine` et `entrDestination` sont des clés étrangères faisant référence à la relation `entrepot`. Ils

désignent respectivement l'entrepôt d'où part et celui où arrive le transport. On mémorise également la date de début du transport. Lorsque le transport a atteint l'entrepôt d'arrivée, la ligne le concernant est supprimée de la table.

- `cargaison(idTransport int(11), idProduit int(11), nombreUnites int(10))`  
 Cette relation mémorise le contenu des cargaisons. On y mémorise l'identifiant du transport concerné et l'identifiant des produits transportés, avec le nombre d'unités de chacun. `idTransport` est une clé étrangère référençant `transport` et `idProduit` est une clé étrangère référençant `produit`.

Le schéma relationnel est représenté sous forme graphique sur la figure 1.

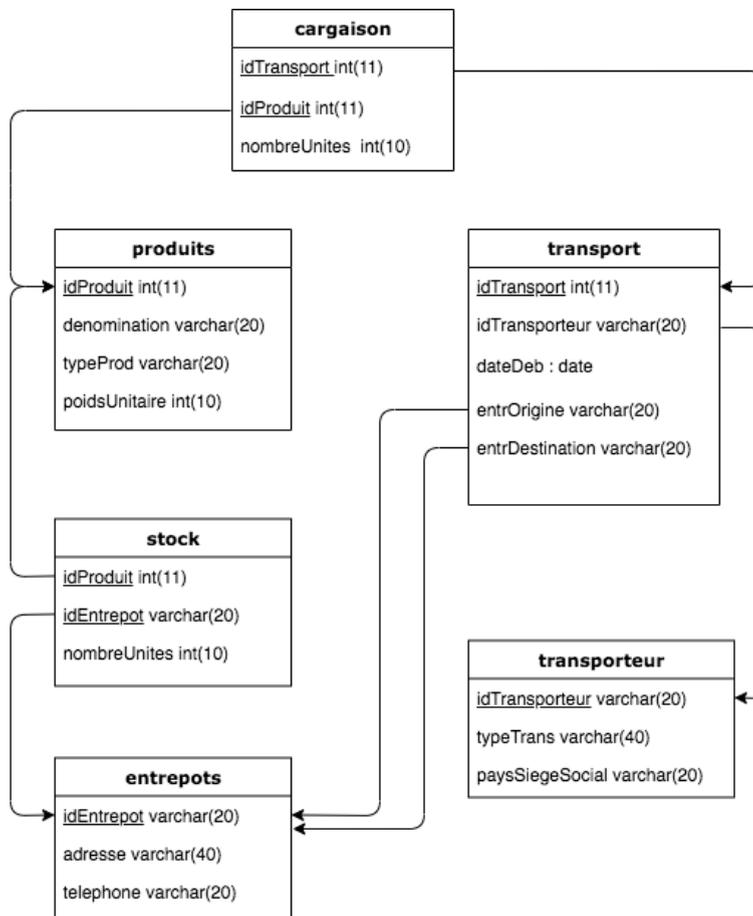


FIGURE 1 – Vue graphique du schéma relationnel de la base.

**Question :** Écrivez les requêtes SQL correspondant aux recherches suivantes. Prenez soin d'éliminer les doublons lorsque le résultat peut en comporter.

1. Quels sont les transports dont la date de début est le '14-11-2019' et qui partent de l'entrepôt *Lyon012* ? Afficher l'identifiant du transport et l'identifiant de l'entrepôt d'arrivée.

*1pt*

*SELECT idTransport, entrDestination*

*FROM transport*

*Where dateDeb = '14-11-2019' and entrOrigine = 'Lyon012';*

2. Quels sont les différents types de produits de la cargaison du transport d'identifiant *1411201924* ?

*1,25pts*

---

```
SELECT DISTINCT typeProd
FROM Produit, Cargaison
Where Produit.idProduit=Cargaison.idProduit and idTransport=1411201924
```

3. Quels sont les entrepôts (identifiant et adresse) où il n'y a pas de produit de type 'Vin' ?

*1,75pts*

```
SELECT idEntrepot, adresse FROM entrepot
Where idEntrepot not IN (
SELECT idEntrepot
from stock,Produit
Where stock.idProduit = Produit.idProduit
and typeProd= 'Vin' )
```

4. Combien de transports sont-ils partis chaque jour de l'entrepôt d'identifiant 'Lyon011' ? Afficher la date et le nombre.

Remarque : comme on ne mémorise que les transports en cours, il s'agit d'afficher le nombre de transports partis et non encore arrivés, ceci pour chaque date.

*1,5pts*

```
SELECT dateDeb, count(*)
FROM transport
Where entrOrigine = 'Lyon011'
Group By dateDeb
```

5. Quels sont les entrepôts destinataires d'un transport dont la cargaison contient des produits de type 'fromage' ? Afficher l'identifiant et l'adresse.

*2pts*

```
SELECT idEntrepot, adresse
From entrepot, transport, cargaison, produit
Where produit.idProduit=cargaison.idProduit
And transport.idTransport= cargaison.idTransport
and Entrepot.idEntrepot = transport.entrDestination
and typeProd='Fromage'
```

6. Afficher, pour chaque transport, le poids total de sa cargaison. Indication : le poids total d'un produit de la cargaison est égal aux nombre d'unités multiplié par le poids unitaire.

*2pts*

```
SELECT idTransport,SUM(nombreUnites*poidsUnitaire)
FROM cargaison, produit
Where Cargaison.idProduit= produit.idProduit
GROUP BY idTransport
```

7. Combien y a t-il de transports dont la cargaison comporte du fromage ?

*2pts*

```
SELECT count(Distinct idTransport) as 'nb transport'
from cargaison, produit
Where cargaison.idProduit=produit.idProduit
and typeProd= 'Fromage'
```

8. Pour chacun des produits actuellement transportés, on souhaite connaître le nombre total d'unités transportées lorsque ce nombre dépasse 1000 unités. Afficher la dénomination du produit et le nombre total d'unités (ie tous transports confondus).

*2pts*

```
SELECT denomination, sum(nombreUnites)
```

---

*from produit, cargaison  
where produit.idProduit =cargaison.idProduit  
Group by idProduit  
Having sum(nombreUnites) > 1000*

9. Afficher l'identifiant des transports et le type de transporteur pour les transports qui sont partis à la même date, du même entrepôt de départ et vers le même entrepôt d'arrivée que le transport d'identifiant "1311201901". Indication : il s'agit d'une auto-jointure.

*2,25pts*

*SELECT t1.idTransport, typeTrans  
FROM transport t1, transport t2, transporteur  
Where t2.idTransport = 1311201901 and t1.dateDeb= t2.dateDeb  
and t1.entrOrigine=t2.entrOrigine and t1.entrDestination = t2.entrDestination  
and transporteur.idTransporteur = t1.idTransporteur and t1.idTransport <> t2.idTransport*

10. On s'intéresse au nombre de transports entre les entrepôts. Afficher chaque couple (entrOrigine, entrDestination) présent dans la relation transport avec le nombre de transports ayant lieu depuis l'entrepôt d'identifiant entrOrigine vers l'entrepôt d'identifiant entrDestination.

*1,75pts*

*select entrOrigine, entrDestination, count(idTransport)  
from transport  
group by entrOrigine, entrDestination*

11. Pour les transports qui ont lieu depuis l'entrepôt d'identifiant "Paris-007" à destination de l'entrepôt d'identifiant "New-York-001", quels sont les transporteurs qui assurent plus de 20 % de ces transports ? Pour chacun de ces transporteurs, afficher son identifiant et le nombre de transports qu'il assure.

*2,5pts*

*select idTransporteur, count(idTransport)  
from transport  
where entrOrigine='Paris-007' and entrDestination = 'New-York-001'  
group by idTransporteur  
having count(idTransport) >= (select 0.2\*count(\*)  
from transport  
where entrOrigine='Paris-007' and entrDestination = 'New-York-001');*