
Interrogation d'informatique

2^{ème} année - Octobre 2017-2018



Durée totale : 60mn
Documents autorisés : Aucun.

- Le barème est indicatif et sur 20 points.
- Le sujet est sur 6 pages - il y a deux exercices.

Proposition de corrigé - NE PAS DIFFUSER

Exercice 1 Requêtes SQL (12 pts)

On considère une base de données au sujet des associations étudiantes, de leurs membres et des événements qu'elles organisent. 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*).

- `etudiants(numero int(11), nom varchar(20), prenom varchar(20), ville varchar(20))`
Un étudiant est identifié de manière unique par un numéro. On mémorise son nom, son prénom et sa ville.
- `associations(nomAsso varchar(20), adresse varchar(40), contact int(11))`
L'attribut `contact` est une clé étrangère référençant l'attribut `numero` de la relation `etudiants`. Une association est identifiée par son nom. On mémorise aussi son adresse et le numéro de l'étudiant à contacter en cas de question (`contact`).
- `membres(numEtu int(11), nomAsso varchar(20))`
L'attribut `nomAsso` est une clé étrangère référençant la relation `associations`. `NumEtu` est une clé étrangère référençant l'attribut `numero` de la relation `etudiants`. Cette relation mémorise de quelle(s) association(s) les étudiants sont membres.
- `evenements(titreEvt varchar(20), asso varchar(20), datEvt date, cout int(7), paf int(4))`
L'attribut `asso` est une clé étrangère référençant l'attribut `nomAsso` de la relation `associations`. Un événement est identifié par un titre (`titreEvt`). Il est organisé par une association, à une date donnée (pour simplifier, un événement ne dure qu'un seul jour). On mémorise le coût lié à son organisation et la participation aux frais (`paf`) demandée à chaque participant.
- `participe(numEtu int(11), titreEvt varchar(20))`
L'attribut `titreEvt` est une clé étrangère référençant la relation `evenements` et `numEtu` est une clé étrangère référençant l'attribut `numero` de la relation `etudiants`. Cette relation mémorise à quels événements les étudiants ont participé jusqu'à maintenant.

Question : Écrivez les requêtes SQL correspondant aux recherches suivantes. Prenez soin d'éliminer les doublons lorsque le résultat peut en comporter. **Les questions ne sont pas classées par ordre de difficulté.**

1. On s'intéresse aux événements ayant eu lieu depuis le 12 septembre 2016 et de coût inférieur à 500 euros. Pour chacun, affichez le titre, la date à laquelle il a lieu, le nom de l'asso organisatrice, le coût, le nom et le prénom du contact de l'association. Remarque : la date sera indiquée par '12/09/2016' ; l'opérateur de comparaison est le même que pour les nombres.

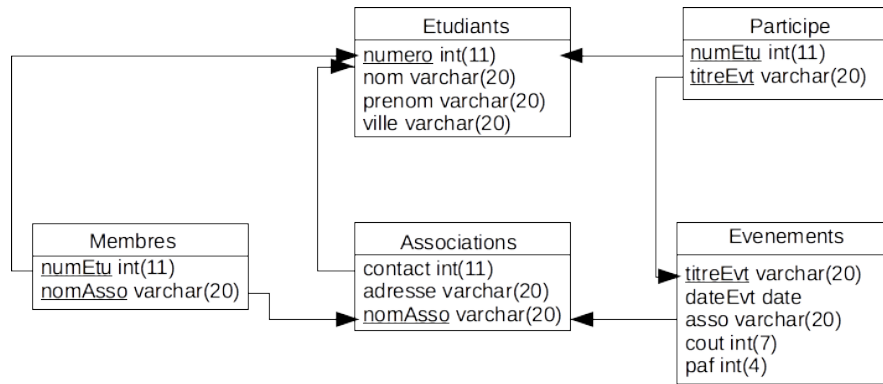


FIGURE 1 – Schéma relationnel de la base.

```
SELECT e.titreEvt, e.datEvt, e.asso, e.cout, t.nom, t.prenom
FROM evenements e, associations a, etudiants t
WHERE e.datEvt >= '12/09/2016'
AND e.asso = a.nomAsso
AND a.contact = t.numero
AND cout <= 500;
```

Barème : 1,5 pts - 0,25 par ligne, ce qui revient à mettre 0,75 sur la jointure (liste des relations+les 2 égalités)

- Combien d'événements chaque association a-t-elle organisé depuis le 12 septembre 2016 ? Afficher le nom de l'asso et le nombre d'événements.

```
SELECT asso, count(titreEvt)
FROM evenements
WHERE datEvt >= '12/09/2016'
GROUP BY asso;
```

Barème : 1,5 pts - 1 pour count+group by ; 0,5 pour le reste

- Afficher le titre et la date des événements organisés par l'association 'Théâtre en folie', du plus récent au moins récent.

```
SELECT titreEvt, datEvt
FROM evenements
WHERE asso = 'Théâtre en folie'
order by datEvt desc;
```

Barème : 1 pts - 0,25 par ligne

- Quels sont les événements dont le coût est le plus élevé parmi tous les événements ? Afficher le titre et la date de ces événements, en commençant par les plus récents.

```
SELECT titreEvt, datEvt
FROM evenements
WHERE cout = (select max(cout) from evenements)
order by datEvt desc;
```

Barème : 1,5 pts - 1 pt pour la ligne du where avec requête imbriquée ; 0,25 order by ; 0,25 le reste

- Etant donné les événements auxquels elle a participé, combien l'étudiante de numéro 000007 a-t-elle dépensé en participation aux frais (paf) ?

```
SELECT sum(paf)
FROM evenements e, participe p
```

WHERE p.numEtu='000007'

AND e.titreEvt=p.titreEvt ; *Barème : 1,5 pts - 0,75 pour le sum (ne mettre que 0,25 si il y a un group by) ; 0,5 pour la jointure ; 0,25 pour le reste*

6. Affichez le numéro des étudiant-e-s qui ont participé à TOUS les événements ayant eu lieu depuis le 12 septembre 2016. Pensez à compter les événements.

SELECT p.numEtu

FROM evenements e, participe p

WHERE datEvt>='12/09/2016'

AND e.titreEvt=p.titreEvt

GROUP BY p.numEtu

HAVING count(titreEvt) = (select count(*) from evenements where datEvt>='12/09/2016');

Barème : 2 pts - 0,75 pour la requête principale sans la partie having ; 0,25 pour HAVING count(titreEvt) ; 1 pt pour requête imbriquée

7. Affichez le nom et le prénom des étudiant-e-s qui ne sont membres d'aucune asso, en classant le résultat suivant le nom et le prénom

SELECT nom, prenom

FROM etudiants

WHERE numero not in (select distinct numEtu from membres) order by nom, prenom ;

Barème : 1,5 pts - 1 pt pour la ligne du where ; 0,5 pour le reste

8. Quels sont les événements qui ont eu lieu à la même date que l'événement de titre 'Plantons des arbres'? Afficher le titre, l'association, le coût et la participation aux frais (paf) de chacun. On souhaite ne pas afficher le titre 'Plantons des arbres'.

SELECT e1.titreEvt, e1.asso, e1.cout, e1.paf

FROM evenements e1, evenements e2

WHERE e1.datEvt=e2.datEvt

AND e2.titreEvt = 'Plantons des arbres'

AND e1.titreEvt<>e2.titreEvt ;

on peut aussi faire une requête imbriquée.

Barème : 1,5 pts - 0,25 pour les lignes 1,3,4 et 5 ; 0,5 pour la ligne 2 (celle du from)

Exercice 2 Modèle conceptuel et modèle relationnel (8 pts)

On considère le modèle conceptuel de la figure 2. Il représente les données gérées par un site qui propose différents forums. Les forums ont un titre, une thématique et une date de mise en fonction. Les internautes sont mémorisé-e-s après inscription à au moins un forum ; on leur attribue un identifiant et on mémorise leur pseudo, leur mot de passe (crypté) et leur mél. Après inscription, dont on mémorise la dernière date, tout internaute peut ajouter un message dans un des sujets des différents forums. Tout internaute peut aussi créer un sujet dans un forum donné, sujet auquel on attribue un identifiant et un résumé établi en fonction du contenu des différents messages déposés pour ce sujet.

Pour cet exercice, on ne considère pas les types des propriétés.

(Q2.1) Compréhension du modèle conceptuel

1. Expliquez les cardinalités entre les types d'entité Message et Sujet.

Un message est obligatoirement posté en lien avec un sujet ; un sujet est constitué de 0 ou plusieurs messages.

Barème : 0,5 pts

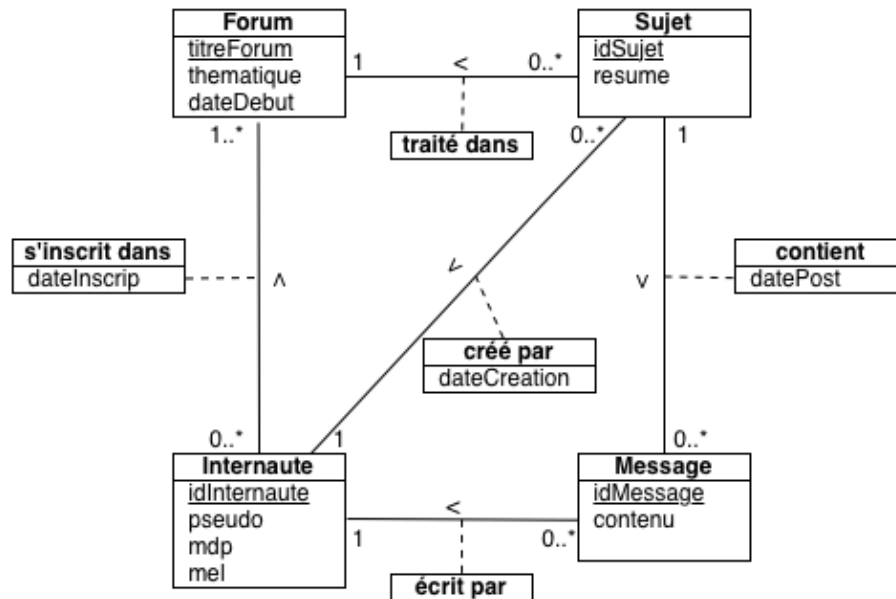


FIGURE 2 – Modèle conceptuel (syntaxe UML).

2. Une personne internaute peut-elle être mémorisée sans avoir jamais ni écrit un message ni créé de sujet ? Expliquez.

Oui, pour cela il suffit qu'elle se soit inscrite au forum (condition nécessaire pour être mémorisée) sans avoir jamais créé de sujet ou posté de message.

Barème : 0,5 pts

(Q2.2) Transformation conceptuel vers relationnel

En utilisant les règles de transformation travaillées en TD-TP, proposez le modèle relationnel correspondant à ce schéma conceptuel : nom des tables, attributs, clés, clés étrangères. Le type des attributs n'est pas demandé.

Les attributs faisant partie de la clé primaire sont soulignés.

— forums(titreForum, theme, dateDeb)

— internautes(idInternaute, pseudo, mdp, mel)

— sujets(idSujet, resume, titreForum, idInternaute, dateCreation)
 titreForum : clé étrangère référençant titreForum de la table forums
 idInternaute : clé étrangère référençant idInternaute de la table internautes

— messages(idMessage, contenu, idAuteur, idSujet, datePost)
 idAuteur : clé étrangère référençant idInternaute de la table internautes
 idSujet : clé étrangère référençant idSujet de la table sujets

— inscriptions(titreForum, idInternaute, dateInscription)
 titreForum : clé étrangère référençant titreForum de la table forums
 idInternaute : clé étrangère référençant idInternaute de la table internautes
 pas utile de souligner dateInscription car seule la date de la dernière inscription est mémorisée.

Barème 3 pts :

0,5 pt pour forums et internautes (avec clé primaire indiquée) ;

0,75 pour sujets (dont 0,5 pour titreForum-idInternaute-dateCreation avec clés étrangères) ;

0,75 pour messages (dont 0,5 pour idAuteur-idsujet-datePost, avec mention des clés étrangères) ;

1 pt pour inscriptions dont 0,25 pour clé primaire et 0,25 pour clés étrangères.
(en gros ça fait 1 pour l'application de chaque règle.)

(Q2.3) Extension du modèle

On souhaite modifier le schéma conceptuel pour mémoriser les informations suivantes : il y a plusieurs modérateurs (qui ne sont pas considérés comme des internautes) dont on mémorise le pseudo, le nom, le prénom et le tél. Chaque message est vérifié par un modérateur et un seul, qui le valide ou non (tout message posté est mémorisé, même les non validés). Un modérateur peut envoyer un avertissement à tout internaute à une certaine date. Certains internautes peuvent ne jamais recevoir d'avertissement, d'autres plusieurs avertissements. Le message d'avertissement étant généré automatiquement, il n'est pas mémorisé dans la base.

1. Dessinez la partie de schéma conceptuel concernée avec les modifications apportées.

Barème : 2 pts ;

présence du type d'entité Modérateur avec les bonnes propriétés et identifiant souligné : 0,5pts

type d'association avertit avec cardinalités et attribut date : 0,75 pt

type d'association verifie avec cardinalités et attribut valide : 0,75 pt

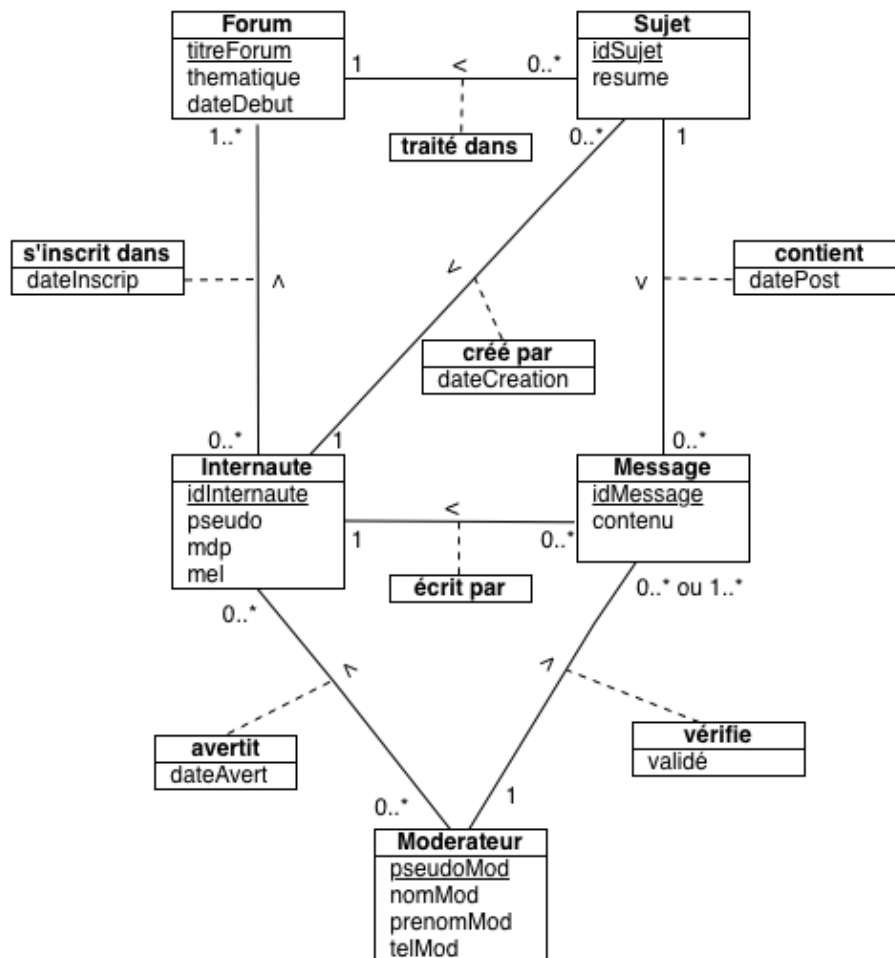


FIGURE 3 – Modèle conceptuel (syntaxe UML).

2. Expliquez rapidement vos choix de cardinalités.

Verifie : 1 côté Modérateur car message vérifié par 1 et un seul modérateur. Pour l'autre côté 0.. ou 1..* car un modérateur vérifie plusieurs messages.*

avertit : un internaute peut avoir été averti 0 ou plusieurs fois (même modérateur ou pas) ; un modérateur peut avertir un ou plusieurs internautes (ou plusieurs fois le même)
Barème : 0,5 pt

3. Indiquer les modifications à apporter au modèle relationnel. * Ajouter les tables :

modérateurs(pseudoMod, nomMod, prenomMod, telMod)

avertissements(pseudoMod, idInternaute, dateAvert)

il est nécessaire que la date fasse partie de la clé pour mémoriser tous les avertissements.

* Ajouter deux attributs à la table messages : pseudoMod (clé étrangère référençant Modérateurs) ainsi qu'un attribut booléen indiquant si le message a été ou non validé.

Barème : 1,5 pts - modérateurs 0,5 pt; avertissements : 0,5 pt; ajout de pseudoMod et boolean dans Messages : 0,5 pt.