Interrogation d'informatique
2ème année - Novembre 2018-2019

Durée totale : *1h*
Documents autorisés : *Aucun.*

INSA

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

Exercice 1

On considère une base de données au sujet de l'organisation de courses de chevaux. 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*).

— `Cheval(numCheval int(11), nomCheval varchar(20), poids int(11), jockey varchar(20))`
Un cheval est référencé de manière unique par un identifiant. On mémorise son nom, son poids et son jockey.
— `Ecurie(nomEcurie varchar(20), contact int(11), adresse varchar(40))`
Une écurie est identifiée par son nom. On mémorise aussi son adresse et le numéro de téléphone à contacter en cas de question (`contact`).
— `Appartient(numCheval int(11), nomEcurie varchar(20))`
L'attribut `nomEcurie` est une clé étrangère référençant la clé de la relation `Ecurie`. `numCheval` est une clé étrangère référençant l'attribut `numCheval` de la relation `Cheval`. Cette relation mémorise à quelle écurie un cheval appartient.
— `Course(idCourse int(11), nomCourse varchar(20), dateCourse date, orga varchar(20), prix int(11))`
L'attribut `orga` est une clé étrangère référençant l'attribut `nomEcurie` de la relation `Ecurie`. Elle est organisée par une écurie, à une date donnée (pour simplifier, un événement ne dure qu'un seul jour). Les chevaux participant à la course reçoivent une récompense calculée à partir de leur classement et d'un prix (`prix`). Un même nom de course peut apparaitre plusieurs fois : il s'agit d'éditions différentes, ayant lieu à des dates différentes. On parle d'édition *dateCourse* de la course *nomCourse*.
— `Participe(idCourse int(11), numCheval int(11), classement int(11))`
Les attributs `idCourse` et `numCheval` sont des clés étrangères référençant, respectivement, la relation `Course` et la relation `Cheval`. Cette relation mémorise le classement des chevaux dans les courses auxquelles ils ont participé.

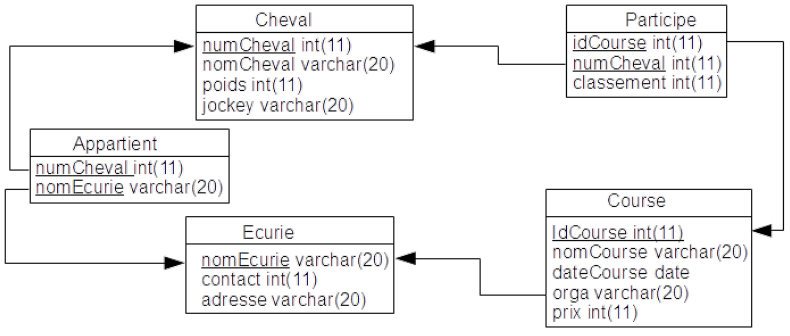

FIGURE 1 – Schéma relationnel de la base.

(Q1.1) Compréhension du schéma (4pt)

1. A combien d'écuries un cheval peut-il appartenir ? Combien de chevaux une écurie peut-elle posséder ? (1pt)
   La relation Appartient est constitué de 2 clés étrangères, donc l'association correspondante était de type 1..* (ou 0..*) de chaque côté ce qui signifie : un cheval peut appartenir à plusieurs écuries, une écurie peut posséder plusieurs chevaux.
2. Combien de jockeys différents peuvent courir avec un même cheval ? Avec combien de chevaux différents un même jockey peut-il courir ? (1pt)
   Jockey est un attribut (qui ne peut pas avoir la valeur NULL) de la relation Cheval donc un cheval ne peut être monté que par un et un seul Jockey. En revanche un jockey peut monter plusieurs chevaux.
3. Il serait logique qu'une écurie puisse posséder un nombre quelconque de chevaux (y compris aucun) et qu'un cheval n'appartienne qu'à une seule écurie. Modifiez, si nécessaire, le schéma pour que ce soit le cas. (2pt)
   Les cardinalités de l'association "Appartient" deviennent 1 du côté Ecurie et 1..* du côté Cheval. L'association "Appartient" ne devient donc pas une table et la clé primaire nomEcurie de l'entité Ecurie migre comme clé étrangère dans la table Cheval.

(justifiez rapidement vos réponses).

(Q1.2) Requêtes SQL (10 pts)

1. On désire obtenir la liste des noms et numéros des chevaux ayant participé à au moins une course en 2017. On veut le résultat trié par ordre alphabétique de leurs noms et sans doublon (1pt).
   select distinct numCheval, nomCheval from Cheval, Participe, Course where Cheval.numCheval = Participe.numCheval and Participe.idCourse = Course.idCourse and dateCourse >= '01-01-2017' and dateCourse <= '31-12-2017' order by nomCheval
2. On désire obtenir la liste des noms et numéros des chevaux ayant gagné au moins une course en 2017. On veut le résultat trié par ordre alphabétique des noms et sans doublon. (un cheval gagne une course si son classement vaut 1) (1pt).
   select distinct numCheval, nomCheval from Cheval, Participe where Cheval.numCheval = Participe.numCheval and Course.idCourse = Participe.idCourse and dateCourse >= '01-01-2017' and dateCourse <= '31-12-2017' and classement = 1 order by nomCheval
3. On veut obtenir la liste des noms des courses associées avec à chaque fois le nom de l'écurie organisatrice et le numéro de contact correspondant. On ne veut pas savoir si une écurie a organisé plusieurs courses de même nom (1pt).
   select distinct nomCourse, orga, contact from Course, Ecurie where Course.orga = Ecurie.nomEcurie
4. On souhaite savoir pour chaque nom de course, combien d'éditions chaque écurie a organisé. Affichez le nom de la course, le nom de l'écurie organisatrice et le nombre d'éditions correspondantes. (1pt).
   select nomCourse, orga, count(orga) as "nombre de fois" from Course group by nomCourse, orga
5. On veut connaître le nombre de chevaux ayant participé à au moins une course organisée par les écuries de Lyon (ie pour lesquelles "Lyon" fait partie de l'adresse, le test s'écrit attribut LIKE "%Lyon%") (2pts).
   select count(numCheval) from Participe, Course, Ecurie where Participe.idCourse = Course.idCourse and Course.orga = Ecurie.nomEcurie and Ecurie.adresse like "%Lyon%"

6. A chaque course à laquelle il participe, un cheval reçoit le prix de la course divisé par son classement. L'écurie "INSA" veut savoir combien elle a donné à chaque cheval. On veut donc la liste des noms des chevaux ayant participé à une course organisée par l'écurie INSA et le total de ce qu'il a reçu dans ces courses. (2 pts)
   select Cheval.nomCheval, SUM(Course.prix/Participe.classement) as total from Cheval, Participe, Course where Cheval.numCheval = Participe.numCheval and Course.idCourse = Participe.idCourse and Course.orga = "INSA" group by Cheval.nomCheval
7. On veut la liste sans doublon des noms des courses ayant été organisées par au moins deux écuries différentes au cours du temps (2 pts)
   select distinct c1.nomCourse from Course c1, Course c2 where c1.nomCourse = c2.nomCourse and c1.orga <> c2.orga order by c1.nomCourse
   Autre possibilité :
   select distinct nomCourse from Course group by nomCourse having count(distinct orga)>=2

(Q1.3) Rétroconception (4 pts)

Proposez un modèle conceptuel des données utilisant le formalisme UML qui, une fois transformé en modèle relationnel par les règles utilisées en cours-TD-TP, correspond exactement au schéma relationnel donné dans l'énoncé.

NE PAS reporter le type des attributs ; leur nom suffit. Pour rappel, il convient d'identifier les types d'entités et les types d'associations avec leur cardinalités.

*** Correction Schéma Conceptuel ***
Pour les cardinalités multiples, (0..*) et (1..*) sont acceptés car l'énoncé n'était pas très explicite sur ce point.

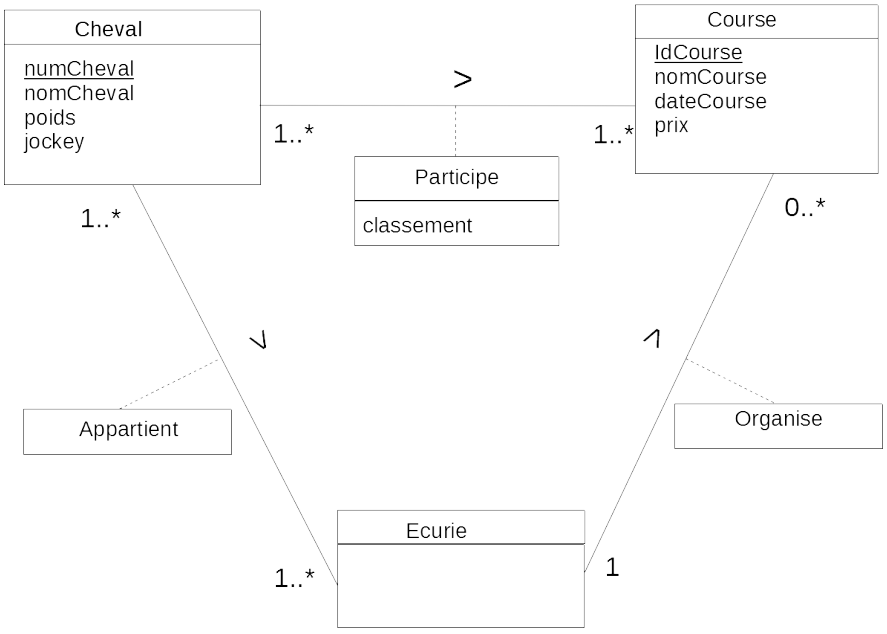

FIGURE 2 – Schéma conceptuel (figure 2).

(Q1.4) Extension du modèle (2 pts)

On désire conserver plus de données sur les jockeys : leur nom, leur numero de licence, leur adresse. Un jockey cours avec un cheval entre une date de début et une date de fin. Il peut

courir avec plusieurs chevaux et un cheval peut courir avec plusieurs jockey.

1. Proposez une extension de votre modèle conceptuel précédent qui permette de prendre en compte cette extension du cahier des charges (1pt).
   *** Correction Schéma Conceptuel ***
   Pour les cardinalités multiples, (0..*) et (1..*) sont acceptés car l'énoncé n'était pas très explicite sur ce point.

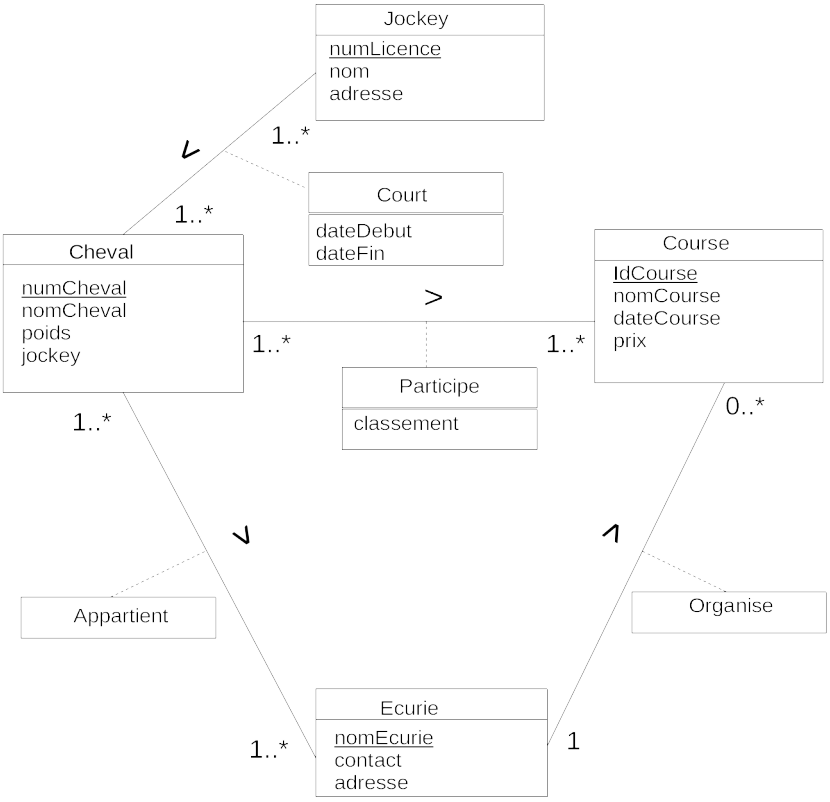

FIGURE 3 – Schéma conceptuel Bis.

2. Donnez le schéma relationnel correspondant (uniquement les relations modifiées ou ajoutées) (1pt).
   *** Correction Schéma Relationnel ***

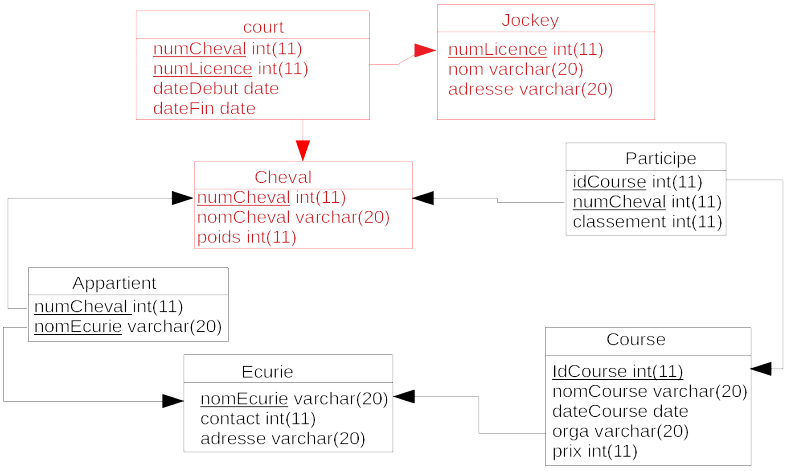

FIGURE 4 – Schéma relationnel bis.