

Bases de Données

Éléments de correction de l'examen final de Bases de Données – L2 info, L3 ts – 7 janvier 2016 - Session 1

Michel SOTO

Université Paris Descartes

AUCUN DOCUMENT AUTORISÉ

Durée: 1h30

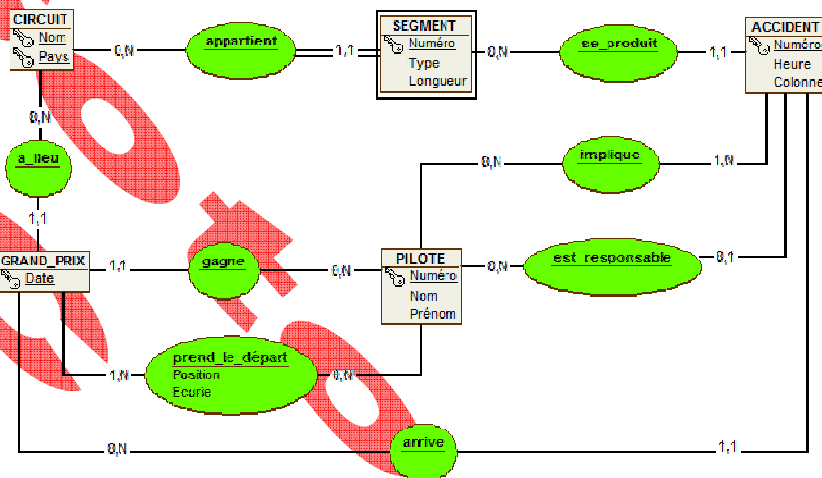
La concision de vos réponses et la propreté de vos copies seront prises en compte.

PARTIE I : CONCEPTION D'UNE BASE DE DONNÉES

L'objectif est de réaliser une base de données permettant de gérer les championnats du monde de Formule 1 (F1) qui ont lieu tous les ans. Chaque course de ce championnat est appelé *Grand Prix* et se déroule sur un circuit automobile. Chaque circuit du championnat est décrit par son nom et son pays. Un grand prix est défini par une date (jour, mois, an) et le circuit où il se déroule. Un pilote de F1 est décrit par un n° unique et ses nom et prénom. Un circuit est considéré comme une suite de segments numérotés de 1 à n. Chaque segment est numéroté relativement au circuit auquel il appartient. Un segment est caractérisé par un type (droite, virage à gauche, virage à droite, chicane) et une longueur en mètre. Pour chaque segment, la BD doit mémoriser les accidents éventuels qui s'y sont produits. Chaque accident est caractérisé par un n° unique, une heure et la météo du moment (soleil, pluie, etc.). Un même accident peut impliquer plusieurs pilotes. La BD doit permettre de retrouver les dates des accidents éventuels de chaque grand prix et le pilote déclaré responsable. Certains accidents n'ont pas de responsable. La position de chaque pilote (1^{ère} position, 2^{ème} position, etc.) par rapport à la ligne de départ de chacun des grands prix doit être mémorisée dans la BD. Un pilote dispute plusieurs grands prix. La BD doit également mémoriser l'unique vainqueur de chacun des grands prix. Chaque pilote qui dispute un grand prix appartient à une seule écurie désignée par un nom (Ferrari, Mercedes, Red Bull, etc.). Un pilote n'appartient qu'à une seule écurie quand il dispute un grand prix mais peut changer d'écurie au cours de sa carrière. Une écurie possède un nom unique.

Question 1 (5 points)

Proposez une représentation graphique de ce système de gestion des championnats du monde de Formule 1 dans le modèle entité-association. Faites figurer les cardinalités sur toutes les associations. Précisez, **au besoin uniquement**, toutes les hypothèses que vous faites dans votre représentation.



NB: la relation **arrive** est nécessaire afin de pouvoir retrouver la date de l'accident

Question 2 (2 points)

En utilisant les règles de passage vues en cours, effectuez le passage de la représentation entité-association de ce système de gestion des championnats du monde de Formule 1 vers un schéma de base de données relationnel.

Vous devez mettre en évidence le résultat de l'application de chaque règle utilisée.

Règle 1, Règle 2

Circuit (Nom, Pays)

Segment (N°Segment, Longueur, Type, #(Nom_circuit, Pays))

Accident (N°Accident, Météo, Heure, #Date_grand_prix, #(N°Segment, #(Nom_circuit, Pays)))

Grandd_prix (Date, # (Nom_circuit, Pays), #N°pilote_vainqueur)

Pilote (N°Pilote, Nom, Prénom)

Règle 3

prend_le_départ (#Date_grand_prix, #N°Pilote, Position, Ecurie)

implique (#N°Accident, #N°Pilote)

Règle 4

- Cas du **zéro rare**

L'association **est_responsable** est implémentée en l'intégrant dans la relation **accident** qui devient:

Accident (N°Accident, Météo, Heure, #Date_grand_prix, #(N°Segment, #(Nom_circuit, Pays)), #N°pilote_responsable)

- Cas du **zéro fréquent**

L'association **est_responsable** est implémentée en créant une nouvelle relation:

est_responsable (#N°Accident, #N°Pilote)

!! NB : Afin d'accélérer la rédaction et la correction de votre copie, veuillez rédiger les questions 1 et 2 de sorte que vos réponses à ces deux questions soient visibles simultanément. !!

PARTIE II : CONNAISSANCE ET COMPREHENSION DU COURS

Question 3 (3 points)

Répondez aux affirmations suivantes **uniquement** par "VRAI", ou "FAUX" ou "NE SAIS PAS".

BAREME : réponse **exacte** : +1 point, réponse **fausse** : -0,5 point **sur la copie**, "ne sais pas" : 0 point

- La relation **AGENCE**, du schéma de base de données fourni en annexe 2, possède 2 clés primaires. **FAUX**
- Un **update** effectué à partir 'un programme C nécessite la déclaration d'un curseur. **FAUX**
- Une dépendance fonctionnelle exprime une contrainte d'intégrité. **VRAI**

Question 4 (2 points) (avec le schéma fourni en annexe 2)

Réécrivez **uniquement** les relations du schéma de la BD, fourni en annexe 1, qui possèdent des clés étrangères. Vous utiliserez la notation diésée (#) pour signaler les clés étrangères. Justifiez votre réponse **une seule fois** pour une même clé étrangère.

COLIS (N°, poids, #N°Expéditeur, #N°Destinataire, #(N°AgenceDépart, #NomPlateFormeDépart), #(N°AgenceArrivée, #NomPlateFormeArrivée))

- L'ensemble {N°Expéditeur} référence l'ensemble {N°} de EXPEDITEUR où il est clé primaire
- L'ensemble {N°Destinataire} référence l'ensemble {N°} de DESTINATAIRE où il est clé primaire
- L'ensemble {NomPlateFormeDépart} référence l'ensemble {Nom} de PLATEFORME où il est clé primaire
- L'ensemble {N°AgenceDépart, NomPlateFormeDépart} référence l'ensemble {N°, NomPlateForme} de AGENCE où il est clé primaire

COLIS_REEXPEDIE (N°, Motif, #N°ColisInitial, #(N°AgenceArrivée, #NomPlateFormeArrivée))

- L'ensemble {N°ColisInitial} référence l'ensemble {N°} de COLIS où il est clé primaire

AGENCE (N°, Non, Adresse, #NomPlateforme)

PARTIE III : SQL (avec le schéma fourni en annexe 1)

Ecrivez en SQL les requêtes suivantes:

Question 5 (8 points)

- a) Création de la table participe. Chaque attribut sera déclaré du type qui vous paraît le plus approprié. Vous considérerez que toutes les tables nécessaires ont déjà été créées

```
create table participe (  
    Date_concours date,  
    Nom_type_peche char(20),  
    Code_pecheur integer,  
    core interger,  
    primary key (Date_concours, Nom_type_peche, Code_pecheur),  
    foreign key (Code-pecheur) references Peche (Code),  
    foreign key (Nom_type_peche) references Type_peche (Nom),  
    foreign key (Date_concours, Nom_type_peche)  
        references Concours (Date, Nom_type_peche)  
);
```

- b) Nombre de pêcheurs qui pratiquent la pêche de type "lancer" et de type "mouche" ?

```
select count(code) from pecheur, participe P1, participe P2  
where P1.Nom_type_peche = 'Lancer'  
and P2.Nom_type_peche = 'Au ver'  
and code = P1.Code_pecheur  
and code = P2.Code_pecheur;
```

- c) Montant moyen des prix par catégorie pour les montants moyens > 500 ?

```
select Nom_categorie, avg(montant) from Prix  
group by Nom_categorie  
having avg(montant) > 500;
```

- d) Noms des pêcheurs qui pratiquent tous les types de pêche sauf la pêche de type "au ver" ?

```
select Nom from Pecheur  
where code not in (  
    select Code_pecheur from Participe  
    where Nom_type-peche = 'Au ver'  
)  
and code in (  
    select Code from Participe  
    group by Code  
    having count(*) = (select count (*) from Type_peche  
        where Nom <> 'Au ver')  
);
```

- e) Liste des codes de pêcheurs qui participent à des concours ne correspondant pas à leur spécialité ?

```
select distinct P.Code_pecheur from Participe P, est_specialiste S  
where P.Code_pêcheur = S.Code_pêcheur  
and P.Type_peche <> S.Type_peche;
```

- f) Pourquoi peut-on affirmer que la requête ci-dessous contient une sous requête non paramétrée ?

```
select nom from type_pêche  
where nom not in (select nom_type_pêche from concours where lieu = 'Millau');  
Cette sous requête est non paramétrée car elle n'utilise aucun attribut de la relation Type_pêche mentionnée dans le from de la requête englobante
```

Réécrivez la requête ci-dessus en utilisant cette fois une sous requête paramétrée.

```
select nom from type_pêche  
where not exists (select * from concours where lieu = 'Millau'  
    and nom_type_pêche = nom);
```

- g) On décide de supprimer la table Type_pêche. Quelles sont les opérations qui devront être effectuées sur le schéma et dans quel ordre. Justifiez votre réponse.

Il faudra dans l'ordre suivant:

1. Supprimer les clés étrangères #Nom_type_pêche des tables Concours, est_specialiste, participe.
2. Supprimer la table Type_pêche.

C'est le seul ordre possible car si on tente de supprimer d'abord la table Type_pêche, le SGBD refuse de réaliser cette suppression. En effet, si cette suppression était réalisée, elle romprait la contrainte d'existence de toutes les valeurs de l'attribut Nom_type_pêches contenues dans les tables, Concours, est_specialiste et participe.

h) Ecrivez en SQL les opérations énoncées en g)

```
1. alter table concours drop constraint concours_type_peche_fk;
   alter table est_spécialiste drop constraint est_specialiste_type_peche_fk;
   alter table participe drop constraint participe_type_peche_fk;
```

En faisant l'hypothèse que la contrainte de clé étrangère #Nom_type_pêche est nommée respectivement: concours_type_peche_fk, est_specialiste_type_peche_fk et participe_type_peche_fk dans les tables Concours, est_specialiste et participe.

```
2. drop table Type_peche;
```

ANNEXE 1

Schéma d'une base de données utilisée pour la gestion de concours de pêche.

Type_pêche (Nom)

Pêcheur (Code, Nom, Prénom, Adresse, Téléphone, #Nom_catégorie)

Catégorie (Nom) // benjamin, junior, senior, retraité

Poisson (Nom)

Concours (Date, lieu, #Nom_type_pêche, Montant_prix_toutes_catégories)

Prix (Rang, Montant, #Nom_catégorie)

est_spécialiste (#Code_pêcheur, #Nom_type_pêche)

participe (#(Date_concours, #Nom_type_pêche), #Code_pêcheur, Score)

compte_pour (#Nom_poisson, (#(Date_concours, Nom_type_pêche), Coefficient)

ANNEXE 2

Schéma d'une base de données utilisée pour la gestion d'expédition de colis.

EXPEDITEUR (N°, Nom, Adresse)

DESTINATAIRE (N°, Nom, Adresse)

COLIS (N°, poids, N°Expéditeur, N°Destinataire, N°AgenceDépart, N°AgenceArrivée, NomPlateFormeDépart, NomPlateFormeArrivée)

COLIS_REEXPEDIE (N°, Motif, N°ColisInitial, N°AgenceArrivée, NomPlateFormeArrivée)

AGENCE (N°, Nom, Adresse, NomPlateforme)

PLATEFORME (Nom, Adresse, Capacité)