

GIS 2A 3^{ème} année

Devoir surveillé bases de données - documents autorisés

© Polytech Lille - Olivier Caron

Durée 2 heures

1 Conception UML (5 points)

Un site web désire recenser toutes les promotions sur des produits par différentes enseignes de commerce durant des périodes de promotion bien connues (exemples : "soldes d'été", "black friday", ...).

Une période de promotion est identifiée par un nom et se caractérise par les dates de début et fin. Les promotions proposées par différentes enseignes peuvent être de deux types : soit un pourcentage de réduction, soit un montant de réduction sur le prix initial. Les enseignes sont décrites par un nom et une adresse web. Les produits sont identifiés par un code se caractérise par un libellé et ont un prix hors promotion (prix initial).

Voici un extrait de données géré par ce site :

- Durant la période "Black Friday", le produit "console Switch" dont le prix hors promotion est de 290 Euros est proposé par :
 - l'enseigne "FNAC" avec une réduction de 20 Euros, soit au final 270 Euros.
 - l'enseigne "Cultura avec une réduction de 10%, soit au final 261 Euros.
- Durant la période "Soldes d'hiver", le produit "console Switch" dont le prix hors promotion est de 290 Euros est proposé par :
 - l'enseigne "FNAC" avec une réduction de 20%, soit au final 232 Euros.
 - l'enseigne "Darty avec une réduction de 10%, soit au final 261 Euros.

Question 1 : Proposez un schéma conceptuel décrit dans le formalisme UML qui permet de gérer l'ensemble de ces informations. Comme l'objectif est de créer une base de données relationnelle, précisez dans votre schéma UML à l'aide du stéréotype "<<Id>>", le ou les attributs UML qui joueront le rôle d'identifiant, ajoutez des attributs si nécessaire.

2 Du schéma conceptuel au schéma relationnel (4 points)

Une saga est une suite de films (épisode) ou interviennent des personnages récurrents. Le schéma conceptuel de la figure 1 permet de recenser ces personnages, de connaître les épisodes où ils interviennent, et de savoir dans quel épisode ils apparaissent pour la première fois.

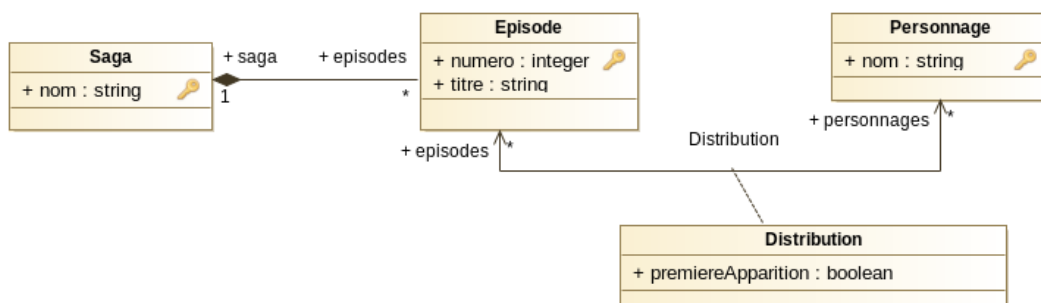


FIGURE 1 – Le schéma conceptuel

Question 2 : Ecrire un schéma relationnel dans le langage SQL conforme à ce schéma conceptuel. Précisez toute contrainte qui vous semble pertinente.

3 Conditionnement de colis (11 points)

Une entreprise de stockage conditionne les différents colis dans des caissons. Les caissons sont identifiés par un numéro et se distinguent par le nombre de colis qu'ils peuvent stocker (attribut capacité). Remarque : tous les colis ont exactement le même volume. Les colis identifiés eux-aussi par un numéro ont tous un type (ex : 'réfrigéré', 'médical', 'alimentaire', 'toxique', 'mobilier', 'jeux',...). Pour chaque type de colis, il existe une table de compatibilité de types de colis. Ainsi, dans la table "compatibilite", la ligne ('mobilier', 'jeux') signifie que tout colis de type "mobilier" peut être stocké dans un même caisson avec des colis de type "jeux". On considère que la table de compatibilité est correctement initialisée : s'il existe une ligne ('xxxx', 'yyyy'), il existe aussi une ligne ('yyyy', 'xxxx').

Voici un schéma conceptuel pour cette base de données :

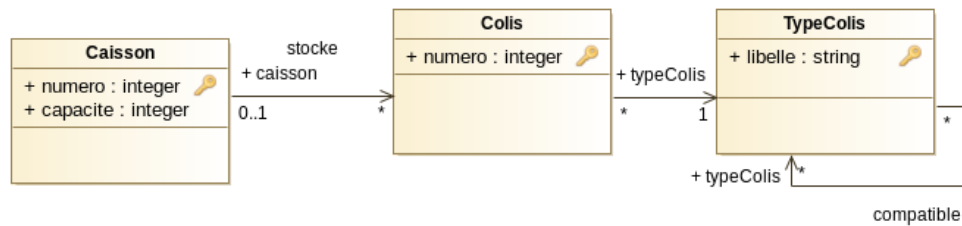


FIGURE 2 – Le schéma conceptuel

La base de données relationnelle correspondante est décrite ci-dessous :

```

create table Caisson (
    numero integer primary key,
    capacite integer not null check(capacite >= 1)
);
create table TypeColis (
    libelle varchar(80) primary key
);
create table Colis (
    numero integer primary key,
    refTypeColis varchar(80) not null references TypeColis,
    caisson integer references Caisson
);
create table compatibilite (
    refTypeColis1 varchar(80) references TypeColis,
    refTypeColis2 varchar(80) references TypeColis
    primary key(refColis1, refColis2)
);
  
```

Question 3 : Ecrire les requêtes SQL suivantes (3 points) :

- Quels sont les caissons non pleins ?
- Quels sont les caissons non vides ?
- Quels sont les colis compatibles avec le typeColis "alimentaire" ?

Question 4 : Ecrire (8 points) la fonction PHP suivante :

```
function affectationColis($base, $numColis, $caissonVide)
```

Le but de cette fonction est d'attribuer un caisson au colis de numéro \$numColis, ce colis existe déjà dans la base. L'attribution doit privilégier les caissons partiellement remplis. Si aucun de ces caissons n'est compatible (c'est à dire qu'il contient déjà au moins un colis non compatible avec le colis \$numColis), alors le caisson vide dont le numéro est passé en paramètre est sélectionné pour cette attribution.

Remarques utiles :

- La base de données distante est déjà connectée et est accessible via la variable \$base.
- Votre programme peut exploiter la fonction booléenne :

```
function iscompatible($base, $numColis1, $numColis2)
```

Cette fonction renvoie vraie si les 2 colis dont les numéros sont passés en paramètres sont de types compatibles.