

Examen Conception et Administration de Bases de Données Avancées

Janvier 2019. Durée : 3h, notes de cours et polycopiés autorisés.

Les parties 1 et 2 sont à rendre sur copies séparées.

Partie 1 : Utilisation des SGBD

Exercice 1.1 : Conception de base de données

Soit l'application *Gamellon* souhaitant stocker des recettes de cuisine définies de la façon suivante :

Une recette, identifiée par un numéro de recette, est décrite par son nom, une catégorie de plat (*soupe, entrée, plat principal, etc.*), un temps de préparation et un temps de cuisson (pouvant être *zéro*).

Une recette nécessite un ou plusieurs ingrédients (*carottes, viande de bœuf, poivre, etc.*) et requiert un ou plusieurs ustensiles (*casserole, bol, fouet, friteuse, four, etc.*). Pour chaque ingrédient, identifié par son nom (unique dans la recette), on précisera sa quantité et son unité de mesure (*une pincée de sel, 100 grammes de sucre, 3 œufs, etc.*) De plus certains ingrédients ne sont pas disponibles toute l'année (*asperges fraîches en mai, par exemple*), on précisera donc à quelle saison ils sont disponibles. On précisera enfin le coût des ingrédients et leur valeur énergétique.

Une recette se déroule en plusieurs étapes successives, chacune de ces étapes est identifiée par son numéro d'ordre dans la recette et décrite par une action qui porte soit sur un ingrédient (par exemple : *hacher les oignons*), soit sur un ustensile (par exemple : *mettre le four à préchauffer*). Une action peut avoir également une destination qui sera nécessairement un ustensile, par exemple : *verser le lait dans une casserole* ou *mettre le plat dans le four*. Chaque action peut également avoir une durée (par exemple : *cuire 20 minutes*).

Les ustensiles, identifiés par leur nom sont indépendants des recettes.

Question 1.1.1 Analysez la description de l'application *Gamellon* pour identifier les propriétés élémentaires et les contraintes (dépendances fonctionnelles, contraintes de valeur, contraintes de multiplicité et autres) s'appliquant sur ces propriétés élémentaires.

Question 1.1.2 Proposez un schéma Entités-Associations (noté en UML) représentant les données de l'application. Précisez les contraintes non représentées dans le schéma et explicitez si besoin les points difficiles.

Exercice 1.2 : Algèbre relationnelle et SQL

EMDb : Ensimag Movie Database

Nous allons travailler sur une base de données cinématographique made in Ensimag, dont l'objectif affiché est de concurrencer sérieusement IMDb. Vous avez du boulot.

Afin d'assurer le stockage persistant des informations, vous décidez d'utiliser une base de données relationnelles. Très bon choix. Cela vous permettra en plus d'appliquer vos connaissances acquises cette année dans votre formidable école. Après une analyse et une conception scrupuleuses, vous créez donc le schéma relationnel (extrêmement simplifié) suivant.

Personne(*idPersonne*, *nomPersonne*, *prénomPersonne*, *annéeDeNaissance*). Cette relation représente la liste des personnes (réalisateurs ou acteurs) de la base de données. Chaque personne est identifiée par un identifiant *idPersonne*, et nous en connaissons le nom, le prénom et l'année de naissance.

Film(*idFilm*, *titre*, *réalisateur*, *annéeDeSortie*). Cette relation représente la liste des films. Chaque film possède un identifiant unique *idFilm*, un *titre*, un *réalisateur* et une année de sortie. L'attribut *réalisateur* est une clef étrangère référençant **Personne**(*idPersonne*).

Joue(*acteur*, *film*). Cette relation représente l'ensemble des individus jouant dans les films. *acteur* est une clef étrangère référençant **Personne**(*idPersonne*), et *film* est une clef étrangère référençant **Film**(*idFilm*)

Question 1.2.1

Traduisez les requêtes qui suivent en algèbre relationnelle et en SQL, sauf celles qui portent la mention « *En SQL uniquement* » qui sont les seules requêtes non traduisibles en algèbre.

Vous ne devez utiliser que les concepts vus en cours. En particulier en SQL2, vous ne devez pas :

- mettre de requête imbriquée dans la clause *FROM*
- utiliser le mot-clef *LIMIT* pour ne renvoyer que les n premières lignes d'un résultat

Toute réponse ne satisfaisant pas ces deux contraintes **sera considérée comme fausse**, même si elle s'exécute correctement sur un SGBD. Vous êtes prévenus !

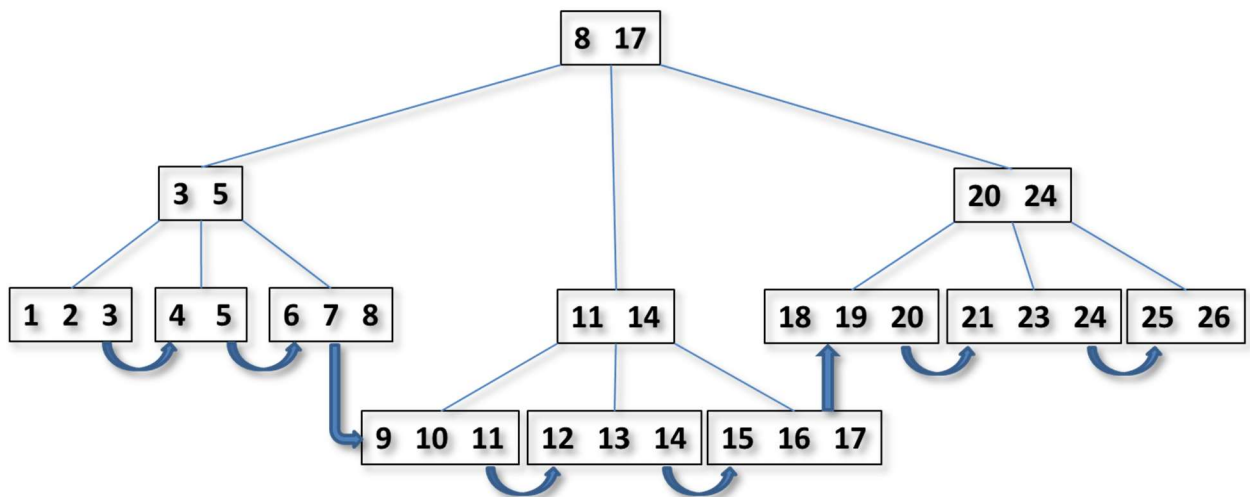
- a) Y a-t-il des films dont l'année de sortie est antérieure à l'année de naissance du réalisateur ? (Si c'est le cas, vous avez de sérieux problèmes de cohérence dans votre base de données...)
- b) Chuck Norris est, comme chacun sait, l'homme le plus fort du monde, et tous les réalisateurs se l'arrachent (d'ailleurs, *ce n'est pas Chuck Norris qui est dans un film, c'est le film qui est Chuck Norris*). Y a-t-il des réalisateurs qui n'ont **jamais** fait jouer Chuck Norris dans aucun de leurs films ?
- c) *En SQL uniquement*. Donnez l'ensemble des années de sortie des films de la base de données, avec, pour chaque année, le nombre de films sortis. Ordonnez le tout par ordre décroissant de nombre de films.
- d) Donnez la liste des couples d'acteurs (nom, prénom) ayant joué ensemble sur le même film (pensez à éliminer les couples constitués de deux acteurs identiques).

Partie 2 : Fonctionnement des SGBD

A rendre sur copie séparée.

Exercice 2.1 : Arbres B+

Soit l'arbre B+ d'ordre 1 suivant :



Question 2.1.1

Que devient l'arbre après suppression de la clef 5 ? Justifiez.

Question 2.1.2

Que devient l'arbre précédent après l'insertion de la clef 22 ? Justifiez.

Exercice 2.2 : Evaluation de requêtes

Soit la base de données suivante :

Employés (NSS, Nom, Prénom, DDN)
 Départements (DeptNo, DNom, NSSChef)
 Projets (PNo, PNom, DeptNo, Ville)
 TravailleSur (NSS, PNo, NbHeures)

Le schéma physique mis en place pour cette BD est le suivant :

- Index primaire trié sur Employés.NSS
- Index secondaire trié sur Employés.DeptNo
- Index secondaire trié sur Départements.DNom
- Index secondaire trié sur Départements.Ville
- Index primaire trié sur Projets.PNo
- Index secondaire trié sur Projets.PNom
- Index primaire trié sur TravailleSur.NSS

Les statistiques associées :

Relation	Cardinalité
Employés	150
Départements	5
Projets	200
TravailleSur	450

Index	Cardinalité
Employés.NSS	150

Employés.DNom	5
Départements.DeptNo	5
Projets.PNo	200
Projets.PNom	150
Projets.Ville	10
TravailleSur.NSS	150

Question 2.2.1

Déterminez l'arbre algébrique optimisé et la taille des résultats intermédiaires pour la requête suivante :

```
SELECT                E.Nom,                                E.Prénom
FROM  Employés E,  TravailleSur T,  Projets P,  Départements D
WHERE E.NSS = T.NSS AND  T.PNo = P.PNo AND P.DeptNo = D.DeptNo
      AND P.Ville = 'Grenoble' AND D.DNom = 'Recherche' ;
```

Question 2.2.2

Construisez le plan d'exécution (opérateurs physiques vus en cours) utilisant les chemins d'accès et algorithmes suivants :

- Employés : Index sur NSS
- TravailleSur : Index sur NSS
- Projets : Index sur PNo
- Départements : Index sur DNom
- Jointure entre Employés et TravailleSur : tri-fusion
- Jointure entre TravailleSur et Projets : Tri-fusion
- Jointure entre Projets et Départements : Boucles imbriquées avec Départements en boucle externe et matérialisation pour la boucle interne.