

Examen Base de données (BD2)

(Durée 1 h 30)

Exercice 1 : CI & Vue (8pts)

On s'intéresse à la gestion des matchs de football de la coupe du monde. On propose, ci-joint, une portion du schéma relationnel de la base de données construite (on considère uniquement les matchs joués au premier tour):

Groupe (NumGp, VilleGp) : Les matchs du groupe (NumGp) se déroulent dans la ville VilleGp

Equipes_Tour1 (Pays, NumGp, NbButs) : NbButs : nombre de buts de Pays durant le 1^{er} tour

Joueur (NomJ, Pays, Position) : Position indique la position du joueur dans son équipe

Match (NumMatch, DateMatch, Pays1, Pays2, NbButP1, NbButP2):

Cette relation signifie qu'à la date (DateMatch), au match (NumMatch), Pays1 a marqué NbButP1 buts contre Pays2 qui a marqué NbButP2 buts. Le gagnant étant celui qui a marqué plus de but

But (NumMatch, NomJ, Minute) :

Au match (NumMatch), le joueur (NomJ) a marqué un but à l'instant (Minute).

1. Ecrire la commande de création de la table Match en précisant ses contraintes de clés primaire et étrangères.
2. Déterminer une contrainte d'intégrité intra-relation horizontale et écrire la commande permettant de la définir.
3. Créer une vue **But_Joueur** qui donne pour chaque joueur, le nombre de buts qu'il a marqué.
4. Utiliser la vue But_Joueur pour donner les joueurs ayant marqué tous les buts de leurs pays (cela signifie qu'un joueur x apparait dans le résultat s'il a marqué tous les buts de son pays)
5. Utiliser la vue But_Joueur pour donner le buteur de ce tour (le joueur ayant marqué le plus grand nombre de buts dans l'ensemble des matchs)
6. Donner les noms des joueurs n'ayant marqué aucun but.
7. Définir un trigger permettant de gérer automatiquement l'attribut NbButs de Equipes_Tour1 dans les cas insertion/suppression d'un Match

Exercice 2 : Accès concurrents (7pts)

Soient T1, T2 et T3 trois transactions, A et B deux éléments de la base de données

T1 : R1(A) : a1 ← A R1(B) : b1 ← B a1 ← a1 * b1 W1(A) : A ← a1	T2 : R2(B) : (b2 ← B) b2 ← b2 * 2 W2(A) : A ← b2	T3 : R3(B) : b3 ← B b3 ← b3 * 10 W3(A) : A ← 10 W3(B) : B ← b3	Où a1, b1, b2, b3 sont des variables locales aux transactions
---	--	---	--

1. Donner les résultats corrects possibles des exécutions simultanées des trois transactions pour A=10, B=0.

Soit l'ordonnancement suivant: **O : R1(A) R2(B) R1(B) R3(B) W3(A) W2(A) W3(B) W1(A)**

2. Donner le résultat de l'exécution de O pour A=10, B=0.
3. Donner le scénario d'exécution de l'ordonnancement O en appliquant le protocole du verrouillage à deux phases (V2P). Dessiner le graphe des attentes en cas d'interblocage.
4. Donner le scénario d'exécution de O en appliquant l'algorithme d'estampillage à 2 estampilles.
 On suppose qu'initialement : L(A)=0, E(A)=0, L(B)=0, E(B)=0, L(C)=0, E(C)=0
 Pour les estampilles des transactions, on suppose que E(T1)=1, E(T2)=2 et E(T3)=3

- Certains systèmes utilisant les techniques de verrouillage évitent les interblocage par la prévention. Parmi les stratégies de prévention des interblocages: la stratégie "Wound-Wait".

Description de la stratégie "Wound-Wait" (blesser-attendre)

Chaque transaction est estampillée avec l'heure de son lancement (ou un compteur).

Quand une transaction T_i demande un verrou sur un objet déjà verrouillé par une transaction T_j , Si T_i est plus vieille que T_j , alors la transaction T_i tue (Wound) T_j (T_j rejetée) et T_i s'empare de sa ressource (obtient le verrou), sinon T_i est mise en attente (Wait).

Réexécuter l'ordonnancement O en appliquant le protocole du verrouillage à deux phases mais en utilisant cette fois-ci, la stratégie Wound-Wait en cas de conflit entre transactions.

Pour les estampilles des transactions, on suppose que $E(T_1) = 1$, $E(T_2) = 2$ et $E(T_3) = 3$

Exercice 3: Index (5 Pts):

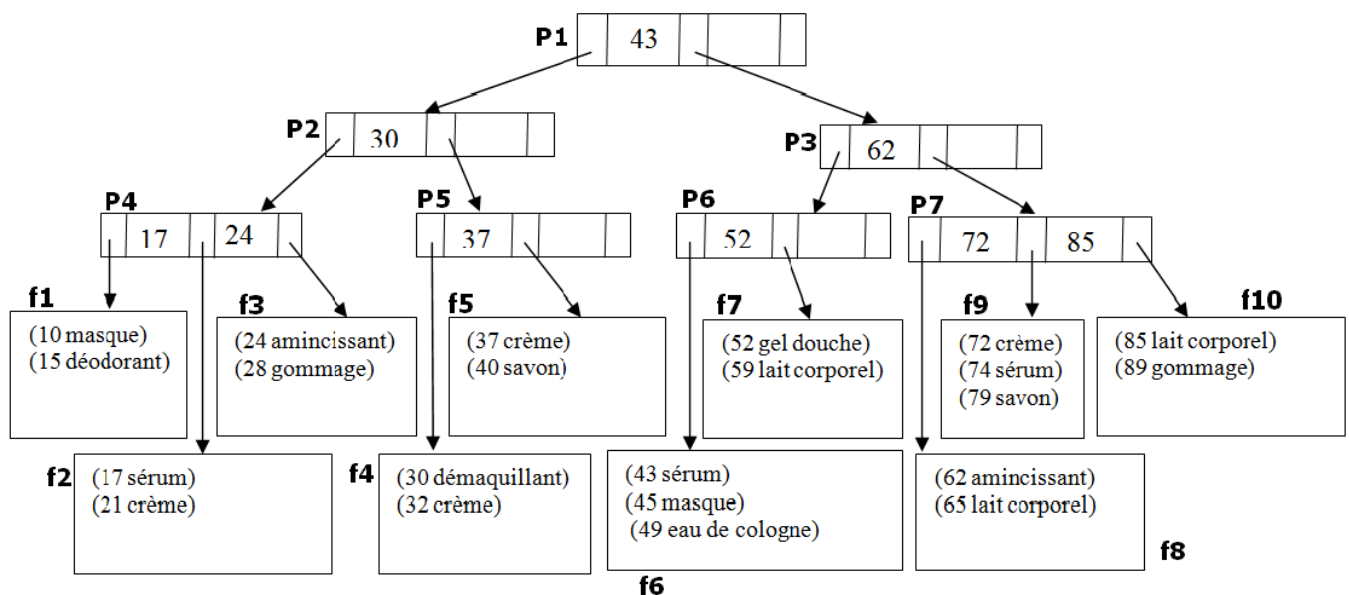
Soit la relation Produits sur laquelle est défini un index primaire (**Idx_P**) sur le numéro produit (NumP) en utilisant un B-arbre d'ordre 3 dont les feuilles pouvant contenir 3 tuples au max.

Produits (**NumP**, **TypeP**, **NomP**, **PrixUV**, **QtéStock**)

NumP est un nombre correspondant au numéro du produit permettant de l'identifier (clé primaire).

TypeP est le type du produit pouvant prendre comme valeur : (amincissant, crème, déodorant, démaquillant, eau de cologne, gel douche, gommage, lait corporel, masque, savon, sérum)

Seuls NumP et TypeP sont représentés dans les tuples, les autres attributs ne sont pas mentionnés par souci de clarté du dessin.



- Insérer les tuples (51 crème) puis (73 masque) (**expliquer**)
- Supprimer (17 sérum) puis (40 savon) (**expliquer**)
(Pour les questions 1 et 2, dessiner uniquement les parties de l'arbre modifiées)
- Pour quel type de requêtes, le temps d'exécution est amélioré par l'utilisation de l'index Idx_P?
- Si on veut créer un index pour la relation Produit sur le type du produit (TypeP), quel est l'index (parmi ceux vus en cours) le plus approprié? Justifier.

Bon courage