



Concours d'accès au Doctorat LMD Informatique 2015 – 2016

Module Bases de Données

Durée 1h30

Exercice 1 :

Soit le modèle relationnel suivant relatif à la gestion d'un grand tour de cyclisme.

EQUIPE(NumeroEquipe, NomEquipe, DirecteurSportif, Date-de-création, SiteWeb)
COUREUR(NumeroCoureur, NomCoureur, NumeroEquipe*, CodePays*)
PAYS(CodePays, NomPays)
TYPE_ETAPE(CodeType, LibelléType)
ETAPE(NuméroEtape, DateEtape, VilleDep, VilleArr, NbKm, CodeType*)
PARTICIPER(NumeroCoureur*, NumeroEtape*, TempsRéalisé)

*Remarque : les clés primaires sont soulignées et les clés étrangères sont marquées par *, le champ TempsRéalisé est donnée en secondes.*

Partie I (9 pt)

Supposons que toutes les tables ont été déjà créées dans la base de données par l'utilisateur USER 1 :

1. Exprimer en SQL les contraintes liées à la table « Participer ».
 - ALTER TABLE Participer ADD CONSTRAINT pk_participe PRIMARY KEY(NumeroCoureur, NumeroEtape) (0.5 pt)
 - ALTER TABLE Participer ADD CONSTRAINT fk_participe_coureur FOREIGN KEY(NumeroCoureur) REFERENCES Coureur(NumeroCoureur) (0.5 pt)
 - ALTER TABLE Participer ADD CONSTRAINT fk_participe_etape FOREIGN KEY(NumeroEtape) REFERENCES ETAPE(NumeroEtape) (0.5 pt)
2. La longueur en KM d'une étape doit être comprise entre 50 et 100 KM. Donner la requête SQL qui permet d'exprimer ce besoin.
 - Alter Table Etape ADD CONSTRAINT ch_etape_km CHECK(NbKm BETWEEN 50 AND 100) (0.5 pt)
3. Dans quelle partie de l'architecture du SGBD se trouve la liste de toutes ses tables de USER1?
 - Dans la métabase (catalogue, dictionnaire). (0.5 pt)
 - Donner la requête correspondante
 - SELECT * FROM USER_TABLES (0.5 pt)
4. L'administrateur veut créer un utilisateur USER2 qui peut lire toutes les tables de tous les schémas et modifier le contenu de la table Participer.
 - Donner les requêtes SQL correspondantes
 - CREATE USER USER2 IDENTIFIED BY xxxxx (0.5 pt)
 - GRANT SELECT ON ANY TABLE TO USER2 (0.5 pt)
 - GRANT UPDATE ON Participer TO USER2 (0.5 pt)
5. L'administrateur veut ajouter les caractéristiques suivantes à USER 2 :
 - Pour chaque session, 40 minutes d'inactivité maximum (**IDLE_TIME**) sont autorisées et 5 tentatives de connexion (**FAILED_LOGIN_ATTEMPTS**) sont permises avant blocage du compte.
 - Quelle solution proposez-vous ? Donner la requête SQL.

Pour chaque session, 40 minutes d'inactivité maximum (**IDLE_TIME**) sont autorisées et 5 tentatives de connexion (**FAILED_LOGIN_ATTEMPTS**) sont permises avant blocage du compte.

- Quelle solution proposez-vous ? Donner la requête SQL.

- Il faut créer un profil et l'attribuer à USER 2 (0.5 pt)

Profil (0.5 pt)

```
CREATE PROFILE Pr LIMIT
IDLE_TIME 40
FAILED_LOGIN_ATTEMPTS 5
```

Attribution du Profil

```
ALTER USER USER2 PROFIL Pr ; (0.5 pt)
```

Pour des besoins de Gestion, USER1 veut sauvegarder, dans la base de données, le total du temps réalisé pour chaque équipe.

- Proposer une solution. Donner les requêtes correspondantes
- **Solution :**
 - Ajouter un attribut Total_Time dans la table Equipe et un trigger Total_Time_Trigger qui mis à jour le trigger à chaque insertion dans la table Participer. (0.5 pt)
 - ALTER TABLE EQUIPE ADD Total_Time NUMBER; (0.25 pt)

Trigger (1 pt)

```
CREATE OR REPLACE TRIGGER Total_Time_Trigger
AFTER INSERT OR DELETE OR UPDATE ON PARTICIPER
FOR EACH ROW
BEGIN

IF INSERTING
UPDATE Equipe SET Total_Time=Total_Time + :NEW.TempRealise
WHERE NumeroEquipe IN
Select NumeroEquipe from Coureur Where NumeroCoureur=:NEW. NumeroCoureur;

IF DELETING
UPDATE Equipe SET Total_Time=Total_Time - :NEW.TempRealise
WHERE NumeroEquipe IN
Select NumeroEquipe from Coureur Where NumeroCoureur=:NEW. NumeroCoureur;

IF UPDATING
UPDATE Equipe SET Total_Time=Total_Time - :Old.TempRealise +:NEW.TempRealise
WHERE NumeroEquipe IN
Select NumeroEquipe from Coureur Where NumeroCoureur=:NEW. NumeroCoureur;
END;
```

6. USER 1 veut supprimer l'équipe 'FESTINA'. Quel est le problème rencontré ? Quelles sont les solutions possibles.

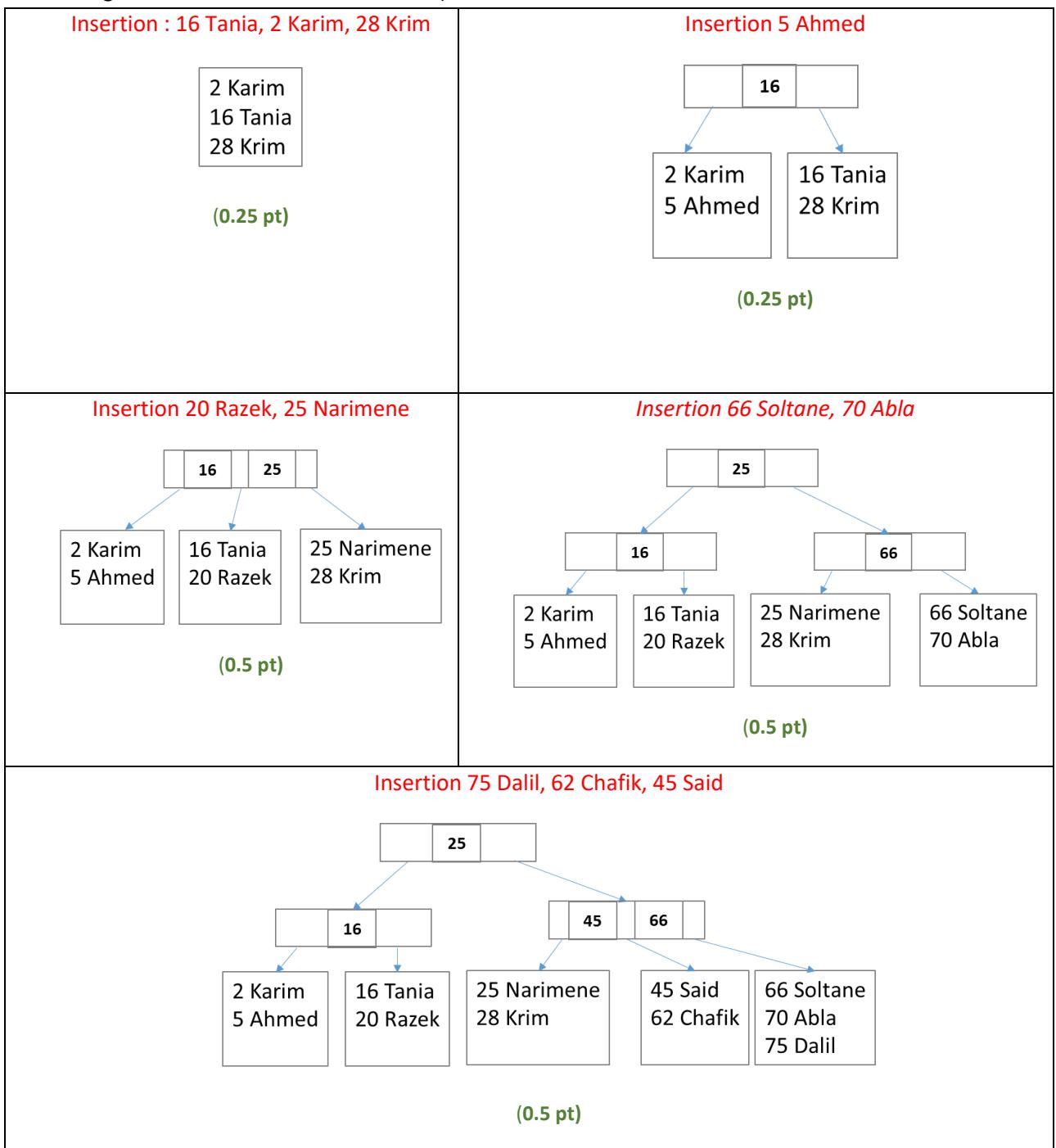
- **Problème :** Si des coureurs liés par une clé étrangère vers cette équipe existent déjà dans la BD, le système refuse de supprimer l'équipe. (0.5 pt)
- **Solutions :**
 - **Sol 1 :** Lors de la création de la clé étrangère, il faut ajouter l'option EN CASCADE (0.25 pt)
 - **Sol 2 :** Supprimer les coureurs de l'équipe Festina ou les attribuer à une autre Equipe ensuite supprimer l'équipe Festina. (0.25 pt)
 - **Sol 3 :** Désactiver la contrainte de clé étrangère. Supprimer l'équipe Festina ensuite décider pour les coureurs (suppression ou attribution à une autre équipe). (0.25 pt)

Partie II (4 pt)

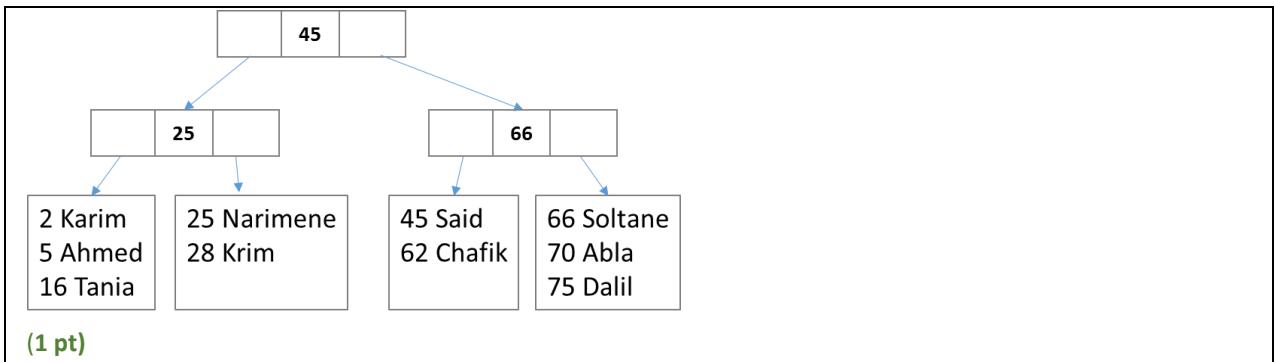
Soit un b-arbre d'ordre 3 défini sur l'attribut *numéroCoureur* de la table « Coureur ». Nous supposons que chaque page feuille du b-arbre peut contenir au maximum 3 enregistrements. Soit l'ordre d'arrivée suivant des enregistrements (pour chaque enregistrement, nous représentons seulement le numéro du coureur et le nom du coureur) :

16 Tania, 2 Karim, 28 Krim, 5 Ahmed, 20 Razek, 25 Narimane, 66 Soltane, 70 Abla, 75 Dalil, 62 Chafik, 45 Said

- Construire le b-arbre correspondant étape par étape (montrer seulement les étapes où il y a un changement sur la structure du b-arbre)



- Supprimer : 20 Razek



3. Soit la requête R suivante : *Select * from Coureur where NomCoureur like "K%"*

L'administrateur constate que la requête R est fréquente dont l'exécution prend beaucoup de temps.
Proposer une solution pour réduire son temps d'exécution.

Solution :

Créer un index secondaire sur l'attribut NomCoureur de la table Coureur (0.5 pt)

CREATE INDEX Nom_C_idx ON Coureur(NomCoureur); (0.5 pt)

Exercice 2 : (7 pt)

Soient les transactions T1, T2 et T3 :

T1	T2	T3
R1(a1←A)	R2(a2←A)	R3(c3←C)
R1(b1← B)	a2←a2*2	R3(b3← B)
W1(B←a1)	W2(A←a2)	W3(B←0)
W1(A←b1)	R2(b2←B)	c3←c3+b3
	b2←b2+10	W3(C←c3)
	W2(B←b2)	

- 1- Etant donnés A=10, B=20, C=0, quels sont les résultats corrects des exécutions simultanées de ces trois transactions ? (1 pt)

	A	B	C
T1T2T3	40	0	20
T1T3T2	40	10	10
T2T1T3	30	0	20
T2T3T1	0	20	30
T3T1T2	0	20	20
T3T2T1	10	20	20

- 2- Soit l'ordonnancement O suivant :

R2(A)W2(A)R2(B)R3(C)R1(A)W2(B)R1(B)W1(B)R3(B)W1(A)W3(B)W3(C)

- a. Donnez le résultat de O pour les valeurs initiales A=10, B=20, C=0.

A=30 , B=0,C=20 (1 pt)

- b. O est-il serialisable ? Justifiez.

Permutation des opérations (1 pt)

R2(A)W2(A)R2(B)R3(C)R1(A)W2(B)R1(B)W1(B)R3(B)W1(A)W3(B)W3(C)

R2(A)W2(A)R2(B)W2(B)R3(C)R1(A)R1(B)W1(B)R3(B)W1(A)W3(B)W3(C)

R2(A)W2(A)R2(B)W2(B)R1(A)R1(B)W1(B)R3(C)R3(B)W1(A)W3(B)W3(C)

R2(A)W2(A)R2(B)W2(B)R1(A)R1(B)W1(B)W1(A)R3(C)R3(B)W3(B)W3(C)

T2 T1 T3

DONC SERIALISABLE (0.5 pt)

3- Donnez le scénario d'exécution de O en appliquant le protocole du verrouillage à 2 phases. Y-a-t-il un deadlock ?, (1.5 pt)

si oui comment le résoudre ? (0.5 pt)

R2(A) : T2 demande S(A) , A libre ok

W2(A): T2 demande X(A), A verrouillé S par T2, mise à niveau du verrou

R2(B): T2 demande S(B) , B libre ok

R3(C): T3 demande S(C) , C libre ok

R1(A): T1 demande S(A) , A verrouillé X par T2, T3 attend T2

W2(B): T2 demande X(B), B verrouillé S par T2, mise à niveau du verrou

T2 termine et libère ses verrous

R1(A): T1 réveillée et obtient S(A)

R1(B): T1 demande S(B), B libre, ok

W1(B): T1 demande X(B), B verrouillé S par T1, mise à niveau du verrou

R3(B) : T3 demande S(B) , B verrouillé X par T1, T3 attend T1

W1(A) : T1 demande X(A), A verrouillé S par T1, mise à niveau du verrou

T1 termine et libère ses verrous

R3(B): T3 réveillée et obtient S(B)

W3(B) : T3 demande X(B), B verrouillé S par T3, mise à niveau du verrou

W3(C) : T3 demande X(C), C verrouillé S par T3, mise à niveau du verrou

T3 termine et libère ses verrous.

Conclusion : Aucun Deadlock

Remarque : Nous acceptons ceux qui terminent l'exécution de T3 ensuite reviennent sur T1. Dans les TD on leur dit qu'on peut soit réveiller soit continuer à exécuter et réveiller à la fin de l'exécution.

4- Déroulez le scénario d'exécution de O en appliquant le protocole d'estampillage à 2 estampilles, pour les valeurs initiales. (1.5 pt)

	ER(A)	EW(A)	ER(B)	EW(B)	ER(C)	EW(C)
	0	0	0	0	0	0
R2(A)	2	0	0	0	0	0
W2(A)	2	2	0	0	0	0
R2(B)	2	2	2	0	0	0
R3(C)	2	2	2	0	3	0
R1(A)	T1 annulée					
W2(B)	2	2	2	0	3	0
R1(B)						
W1(B)						
R3(B)	2	2	3	0	3	0
W1(A)						
W3(B)	2	2	3	3	3	0
W3(C)	2	2	3	3	3	3
Reprise de T1 avec l'estampille 4						
R4(A)	4	2	3	3	3	3
R4(B)	4	2	4	3	3	3
W4(B)	4	2	4	4	3	3
W4(A)	4	4	4	4	3	3