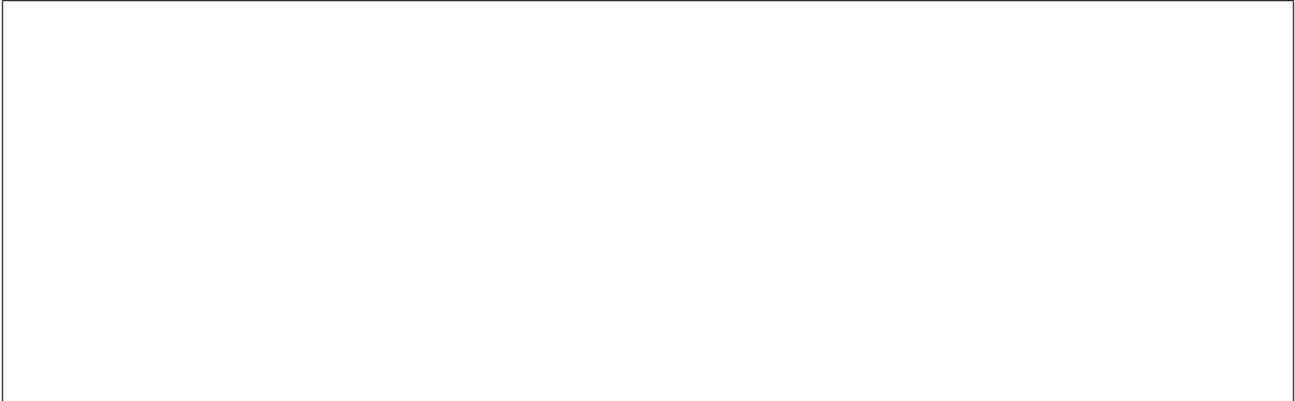


EXAMEN BD AVANCÉES
2019-2020
Durée : 1h30
1 feuille A4 RV autorisée
90 points

- Nom :
- Prénom :
- Groupe :

Exercice 1 (Définition). (9 points) Donner la définition des 4 formes normales.

Exercice 2 (Définition). (6 points) Expliquer ce qu'est un produit cartésien et donner un exemple significatif.



Exercice 3 (MCD). (25 points) Le BDE (Bureau Des Élèves) souhaite mettre en place une base de données pour organiser le WEI (Week-End d'Intégration).

L'objectif de la base de données est d'avoir les coordonnées des étudiants de 2A et de 1A qui participent pour pouvoir communiquer plus facilement. De savoir qui parraine qui et de quand à quand. De connaître les dates d'anniversaires pour organiser des surprises. De constituer des équipes pour des jeux lors du WEI, noter qu'un étudiant ne peut appartenir qu'à une seule équipe. Chaque équipe aura un nom. L'objectif est aussi de savoir qui est inscrit à quelle activité facultative pour pouvoir faire payer le bon prix à chacun.

Concevoir un MCD (10 points) en 4 FN, puis le MLD associé (5 points) et les commandes de créations des tables avec les contraintes correspondantes sans utiliser **ALTER** (10 points).

MCD (10 points)



MLD (5 points)

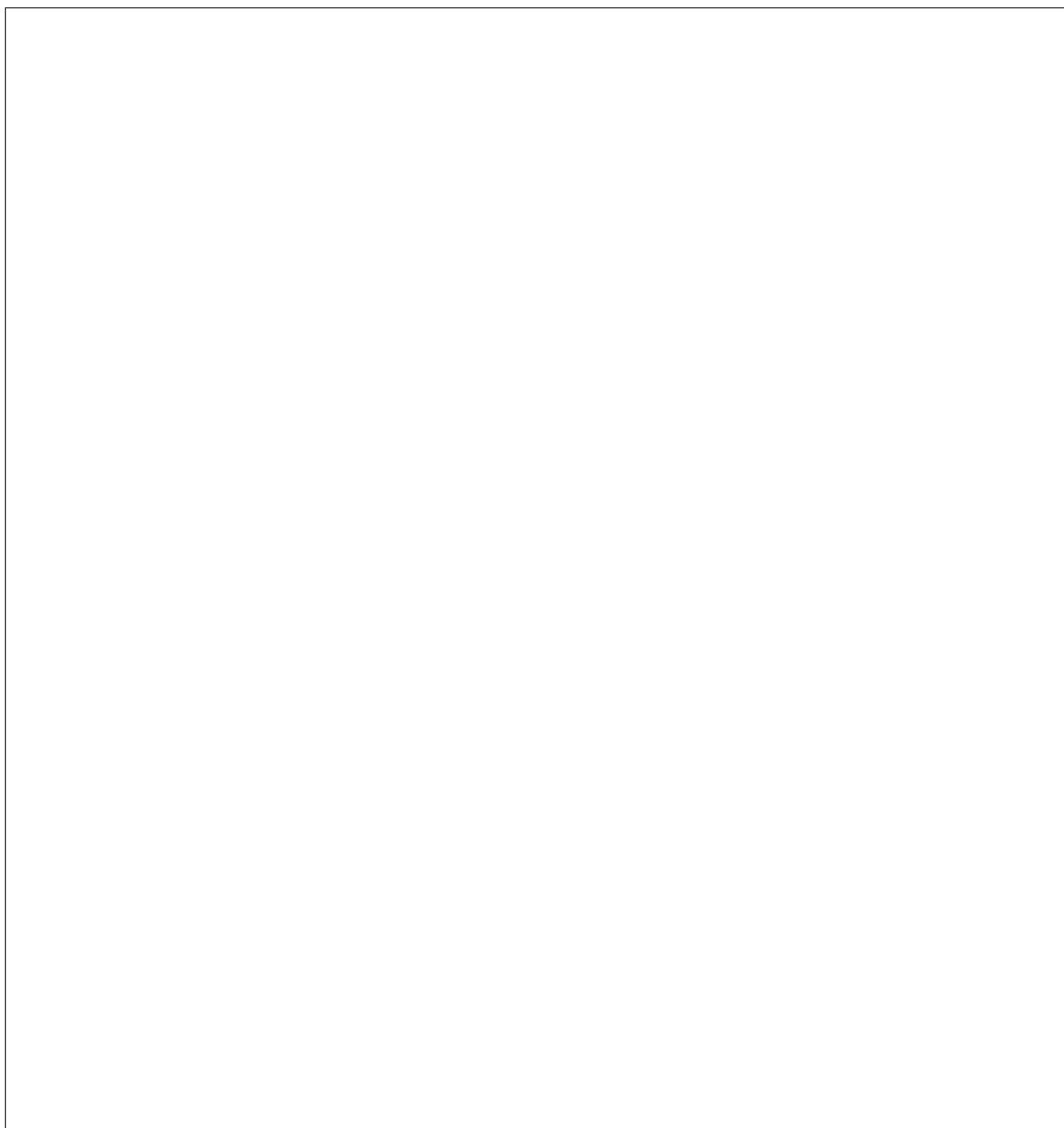
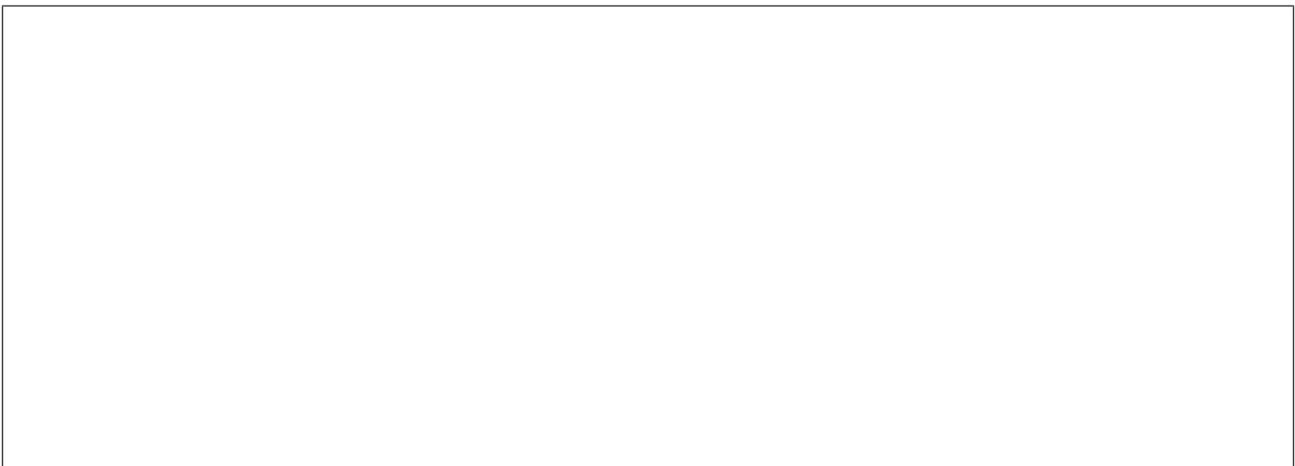


TABLE sans ALTER (10 points)



Exercice 4 (Séquence). (10 points) À partir des tables créées précédemment.

1. (2 points) Donner la requête permettant de trouver l'identifiant d'étudiant le plus élevé.
2. (5 points) La requête précédente donne 41, créer la séquence qui sera utilisée pour définir l'identifiant de la table etudiant.
3. (3 points) Donner la commande permettant d'ajouter un nouvel étudiant de votre choix en faisant attention que les clefs étrangères ne soient pas violées.



Exercice 5 (Trigger). (20 points) En considérant la table `trace`(`d DATE, designation VARCHAR2(100)`); et la table `employe`(`id number, primary key(id), nom VARCHAR2(30), prenom VARCHAR2(30), sexe CHAR(1) check (sexe in ('F','H'))`, `date_naissance DATE, chef number, duree_affectation DATE`); écrire un trigger permettant, en cas de suppression d'un employé, de garder dans la table `trace` les informations suivantes : Le chef `XX` a perdu `NOM PRENOM` collaborateur `YY` et Le chef `XX` `nomme NOM PRENOM` est `supprime`. Attention, il y a toujours un PDG dans une entreprise. Écrire un exemple de commandes pour la table `trace` et la table `employe` permettant de tester votre trigger.

Exercice 6 (Fonction). (10 points)

1. (8 points) Écrire la fonction `proportion`, qui permet de calculer la proportion du nombre de femmes ou d'hommes dans les employés en fonction d'un paramètre d'entrée 'H' ou 'F'.
2. (2 points) Utiliser cette fonction pour connaître la proportion de femmes parmi les employés. Le résultat doit être "XX est le pourcentage de femmes".

Exercice 7 (Optimisation). (10 points) En reprenant la table **employe**, écrire les requêtes suivantes :

1. (2 points) Trouver la liste des employées **nom**, **prenom**, les plus âgées de sexe féminin, triées par ordre alphabétique des prénoms.
2. (5 points) Proposer deux optimisations différentes, donner leurs codes et expliquer comment ces mécanismes fonctionnent.
3. (3 points) Malgré l'optimisation, la requête suivante est très longue, pourquoi ?

```
SELECT distinct nom, prenom
FROM employe chef, employe f
WHERE (f.sexe='F' OR chef.sexe='F');
AND chef.id= f.chef;
```

