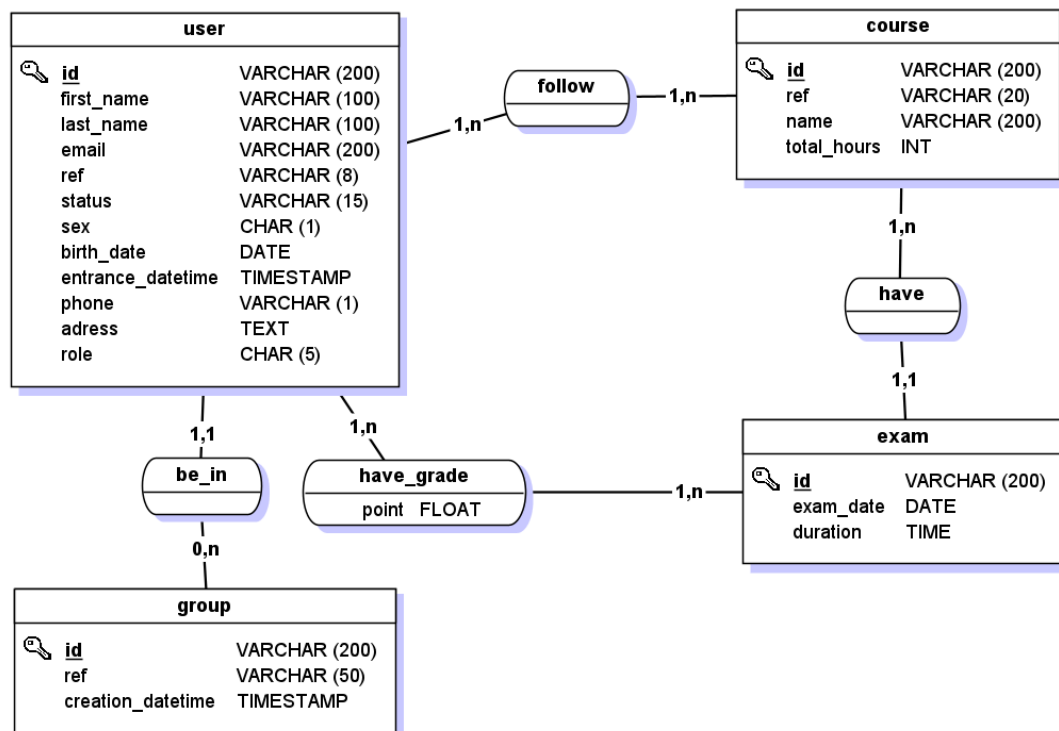


Examen théorique, 7 Avril 2023, Intelligence externe interdite, machine et internet autorisés.

### A. Langage de Définition des Données (LDD - 10 points)

Vous parlez tranquillement avec Herilala (oui, le même Herilala qui est m'assiste dans vos TDs), et d'un bond de curiosité, vous lui demandez sur quel projet il travaille. Il vous parle d'un logiciel de gestion d'école appelé... HEI Admin, et de comment ils sont actuellement en train de galérer pour implémenter un sprint sur la gestion des notes aussi vite que possible. Sans trop réfléchir, vous dites vouloir participer aussi ! D'un coup de mail, bien cordial comme il faut, Herilala vous envoie le schéma MERISE de la base de données à implémenter.



- 1) Au travail ! écrivez dans un fichier **create\_db.sql** ou **create\_db.txt** toutes les instructions nécessaires pour implémenter ce schéma MERISE dans une base de données nommée **hei\_admin**. À vous d'ajouter toutes les contraintes que vous jugerez nécessaires. Se référer au dictionnaire à la fin du sujet pour plus d'informations sur les colonnes (7 points).
- 2) Juste ciel ! Nous avons oublié le nombre de crédits de chaque **Course** ! Décrivez dans un fichier **migration\_v1.sql** le(s) requête(s) SQL pour ajouter la colonne **credits** de type entier à la table **Course**, et aussi, faites en sorte qu'elle n'accepte **que les valeurs > 0** (2 points);
- 3) Toujours dans **migration\_v1.sql**, corrigez le type de **phone** qui doit être un varchar(18); (1 point)

### B. Langage de Manipulation de Données (LMD - 10 points)

Vous avez soumis votre base de données, Herilala vous fournit des données fictives pour chaque table (dites “données mock”) générées avec l’outil FakerJS. Dans un fichier, **requests.sql** ou **requests.txt**, écrivez les requêtes SQL pour faire les tâches suivantes :

1. Vous essayez de voir comment attribuer une note : écrire la requête qui permet de donner la note de 20 à l’**user (id = 1)**, à un examen (**exam.id = 2**). Pas de panique, on suppose que l’étudiant et l’examen en question sont déjà enregistrés dans la base de données; (1 point)
2. Misère ! On apprend que l’élève à qui on vient de donner 20/20 a triché ce jour -là : changez sa note en 0,5 à ce fameux examen; (1 point)
3. Voyons qui ont besoin d’assistance sur ce même examen: listez tous les étudiants, qui n’ont pas eu la moyenne (note < 10), sur l’**exam (id = 2)** en affichant le nom, le prénom, la ref (de l’étudiant) et sa note; (2 points)
4. Mais c’était qui ce tricheur ?! il paraît qu’il y a “Tobi” dans son nom ou prénom : écrire la requête affichant tout sur les “**User**” qui ont fait l’**exam (id = 2)** et dont le nom ou prénom contient “Tobi”(insensible à la casse) + la ref de son groupe; (2 points)
5. Suite à beaucoup de questions sur le **cours (id = 1, ref = ‘LV1’)**, on souhaite savoir à l’aide d’une requête SQL qui sont les élèves (**user.role = ‘S’**) qui ne suivent PAS ce cours; (2 points)
6. Bon, on veut assurer les assistances de PROG1, en invitant les élèves avec les 10 meilleures notes de **PROG1 (course.id = 3)** tout examen confondu : écrire la requête qui permet de le faire en affichant l’email, le nom, le prénom de l’élève; sa note; la date de l’examen. NB : Ce n’est pas grave si un même élève apparaît plusieurs fois dans ce top 10, ça dit qu’il est très doué et qu’il faut qu’il assiste (2 points)

Dictionnaire de données (seulement sur les champs difficiles à comprendre)		
Nom	Description	Exemple
user.ref	Numéro de référence unique de l'utilisateur	'STD22001', 'TCR21004'
user.role	Le rôle de l'utilisateur : soit 'S' pour STUDENT, 'T' pour TEACHER ou 'M' pour MANAGER	'S', 'T', 'M'
user.status	Le statut de l'utilisateur s'il est actif ou non	'actif(ve)', 'décroché(e)'
has_grade.point	La note de l'élève à un examen donné entre 0 et 20	17, 8.5, 0, 15.75
course.ref	La référence du cours qui doit être unique	'PROG1', 'WEB1', 'LV1'
course.name	Le nom du cours	'Algorithmique', ...
group.ref	La référence du groupe qui doit être unique	'H1', 'G1', 'G3'...