

INFO642-INFO633 : SQL et Programmation

CC1 (1h, documents autorisés)

Consignes

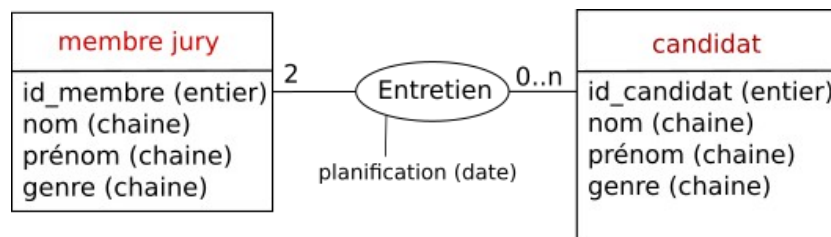
Le sujet sera disponible dans l'espace de cours Moodle. Vous utiliserez obligatoirement un ordinateur de la salle informatique. Vous travaillerez et testerez vos réponses sur le serveur de base de données de l'école <https://tp-epua.univ-savoie.fr/phpMyAdmin/index.php>.

En fin de séance, vous déposerez un fichier au format PDF dans l'espace prévu à cet effet sur l'espace Moodle. Dans ce fichier, vous devez reporter pour chaque question : le code SQL, le code de(s) test(s) effectué(s) ainsi qu'une capture d'écran des résultats obtenus.

Votre fichier devra être nommé selon le modèle suivant *votrenom_votreprenom_votregroupedtp*.

Cas « Gestion des entretiens »

Le modèle entité-association ci-après est un modèle conceptuel des données relationnelles sur lesquelles vous allez travailler. Ce modèle permet la gestion d'entretiens de candidats par un jury composé de deux membres respectant la parité homme/femme :



Travail à faire

Après connexion au serveur de base de données et avoir exécuté les 2 scripts [SchemaBaseEntretiens.sql](#) et [DonneesEntretiens.sql](#), il vous est demandé d'écrire et de tester des requêtes et programmes pour répondre aux questions ci-dessous.

Vous devez reporter à chaque fois le code SQL, le code de(s) test(s) effectué(s) ainsi qu'une capture d'écran des résultats obtenus :

- Ajouter dans la base la candidate Mme *ABORT Roberte* dont l'entretien aura lieu le 01 avril 2022 à 15h00 avec comme membre du jury Mme *JAMES Chloé* et M. *GARNIER Isaac* (déjà présents dans la base également). Vérifier que cet ajout a bien été effectué.
 - L'entretien de M. *REED Justin* (déjà présent dans la base) a été planifié. Il aura lieu le 02 avril 2022 à 17h30 avec Mme *ROUSSEAU Maélie* et M. *COOPER Adam* comme membres du jury (déjà présents dans la base également). Proposer la requête SQL permettant de mettre à jour la base avec ces informations.
- Créer une vue *liste_entretien_candidat* qui affiche selon l'ordre chronologique descendant des entretiens et par ordre alphabétique des nom et prénom des candidats : les nom et prénom des candidats, la date de l'entretien, les nom et prénom des membres du jury, pour l'ensemble des entretiens.
Conseil : écrire et tester la requête avant de l'intégrer dans la vue.

3. Définir et tester une fonction *nombre_entretiens_membre_jury* qui retourne le nombre d'entretiens prévus pour un membre dont l'identifiant *id_membre* est passé en paramètre (utiliser la fonction *COUNT*). Tester votre fonction pour déterminer le nombre d'entretiens prévus pour Mme *SMITH Sophie* dont l'identifiant est 1.
Conseil : écrire et tester la requête avant de l'intégrer dans la fonction.
4. Définir et tester un déclencheur *verification_jury_f* qui, avant chaque modification dans la table candidat, vérifie que l'identifiant du jury féminin correspond bien à un membre féminin.
Dans le trigger, vous pouvez utiliser la commande ci-dessous pour mettre fin à la demande de modification si nécessaire.

```
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Le membre du jury doit être une femme';
```
5. Écrire et tester une procédure *binome_entretien* qui, pour un membre de jury dont les nom et prénom sont donnés en argument, crée une table *liste_binomes* dans laquelle seront stockés les nom et prénoms des membres avec lesquels il a été en binôme lors des entretiens.
Conseil : écrire et tester la requête avant de l'intégrer dans la procédure.
Question subsidiaire : ajouter une colonne à la table *liste_binomes* contenant le nombre d'entretiens réalisés avec chaque binôme.

Rappel de commandes utiles :

- Pour lister les contraintes :

```
SELECT *  
FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS  
[WHERE TABLE_NAME = 'nom-table'];
```
- Pour lister les fonctions et procédures :

```
SELECT *  
FROM INFORMATION_SCHEMA.ROUTINES;
```
- Pour lister les déclencheurs (triggers) :

```
SELECT *  
FROM INFORMATION_SCHEMA.TRIGGERS;
```
- Pour lister l'état des procédures :

```
SHOW PROCEDURE STATUS  
[WHERE DB = 'nom_base'];
```