

## 8. DDL – Création, suppression et modification de tables

Trouvez les requêtes répondant aux énoncés suivants. On vous demande de prendre le temps de bien comprendre les structures demandées afin de créer adéquatement les tables et leurs contraintes respectives.

### Exercice 8.1.

---

Créez la table suivante en respectant toutes les contraintes données :

- **pays :**
  - **id**
    - nombre de 3 chiffres sans décimal
    - clé primaire
  - **nom**
    - chaîne de caractères variable avec un maximum de 64 caractères
    - ne peut être nul

### Exercice 8.2.

---

Créez la table suivante en respectant toutes les contraintes données :

- **employe :**
  - **nas**
    - entier
    - clé primaire
    - la valeur doit être incluse entre 100000000 et 999999999
  - **nom**
    - chaîne de caractères variable avec un maximum de 64 caractères
    - ne peut être nul
  - **prenom**
    - chaîne de caractères variable avec un maximum de 64 caractères
    - ne peut être nul
  - **date\_embauche**
    - date
    - ne peut être nul
  - **date\_naissance**
    - date
  - **genre**
    - doit être f ou h ou x
    - ne peut être nul
  - **courriel\_entreprise**
    - chaîne de caractères variable avec un maximum de 64 caractères
    - ne peut être nul
  - **courriel\_personnel**
    - chaîne de caractères variable avec un maximum de 128 caractères
  - **salaires\_horaire**
    - valeur numérique pouvant avoir 2 décimales
    - ne peut être nul

- doit être supérieur ou égal à 20.00.
- `superviseur`
  - clé étrangère vers un autre employé

### Exercice 8.3.

En vous mettant dans le contexte d'une application gérant une école, on vous demande de rédiger un script créant les tables suivantes en respectant toutes les contraintes données :

- `etudiant` :
  - `id`
    - nombre de 4 chiffres sans décimal
    - clé primaire
  - `nom`
    - chaîne de caractères variable avec un maximum de 64 caractères
    - ne peut être nul
  - `prenom` (prénom)
    - chaîne de caractères variable avec un maximum de 64 caractères
    - ne peut être nul
  - `code_perm` (code permanent au ministère de l'éducation)
    - chaîne de caractères fixe de 12 caractères
    - ne peut être nul
    - doit être unique
  - `courriel`
    - chaîne de caractère variables avec un maximum de 128 caractères
    - doit être unique
  - `programme`
    - clé étrangère vers le champ `id` de la table `programme`
    - ne peut être nul
  - `date_inscr` (date d'inscription)
    - date
- `programme`
  - `id`
    - nombre de 2 chiffre sans décimal
    - clé primaire
  - `nom`
    - chaîne de caractères variable avec un maximum de 32 caractères
    - ne peut être nul
    - doit être unique
  - `sigle`
    - chaîne de caractères fixe de 3 caractères
    - ne peut être nul
    - doit être unique
  - `departement`
    - clé étrangère vers le champ `id` de la table `departement`
    - ne peut être nul
- `departement`
  - `id`

- nombre de 2 chiffre sans décimal
- clé primaire
- `nom`
  - chaîne de caractères variable avec un maximum de 32 caractères
  - ne peut être nul
  - doit être unique

#### Exercice 8.4.

On désire modifier la structure précédente. On vous demande de créer un **nouveau script complet** qui crée ce qui est spécifié au numéro précédent en incluant les ajouts précisés ici.

- La date d'inscription de l'étudiant doit être comprise dans l'intervalle [1<sup>er</sup> janvier 2010, 31 décembre 2019]
- Pour chaque programme, on désire ajouter un étudiant représentant tous les étudiants dudit programme. Pour chaque programme, un seul étudiant peut être nommé représentant. Néanmoins, il est possible qu'un programme n'ait pas de représentant.
- Pour chaque étudiant, on désire ajouter sa moyenne cumulative. Il est à noter qu'une moyenne doit être incluse dans l'intervalle [0%, 100%].
- On désire ajouter un champ de courriel supplémentaire pour accueillir celui donné par l'école. Ce nouveau champ obligatoire doit s'appeler `courriel_etu` et sera créé automatiquement par le SGBD lors de l'inscription d'un nouvel étudiant dans la base de données. Ce nouveau champ doit avoir la forme suivante: `nom_prenom_###@monecole.qc.ca` (où ### représente un nombre de trois chiffres). Vous n'avez pas à valider si le courriel est conforme au schéma donné (exercice complémentaire pour les étudiants désirant pratiquer les expressions régulières).

#### Exercice 8.5.

On désire modifier la structure du numéro précédent. Par contre, cette fois-ci on désire faire des modifications à la structure alors que cette dernière existe déjà et possède plusieurs données. On désire donc créer un **script qui modifie** la structure réalisée précédemment.

- On désire une nouvelle table possédant une adresse :
  - On doit inclure tous les champs habituels d'une adresse : `id`, `no_civique`, `no_appartement`, `rue`, `ville`, `province`, `pays` et `code_postal`.
  - La table doit s'appeler `adresse`.
  - Tous les champs sauf `no_appartement` sont obligatoires.
  - Il existe des valeurs par défaut à province et pays (Québec et Canada respectivement).
  - Une adresse ne peut exister qu'une seule fois dans la base de données.
  - Pour les données existantes, on désire ajouter :
    - Deux adresses optionnelles pour les étudiants :
      - adresse courante (`adresse_cour`);
      - adresse de référence qui sera une adresse fixe à long terme comme celle des parents (`adresse_ref`).
    - Une adresse **obligatoire** pour les départements (`adresse_dep`).
    - Pour tous les cas, si une adresse est supprimée du système, les références doivent être mise à nulle automatiquement.

- Finalement, on désire ajouter un champ pour le local où se trouve le secrétariat de chaque programme. Le champ doit être nommé `secretariat` et correspondre au schéma suivant : `x##-####` (une lettre majuscule indiquant le pavillon, deux chiffres pour l'étage, un tiret et quatre chiffres pour le local). On ne demande pas de valider si une chaîne de caractères est conforme au schéma donné (exercice complémentaire pour les étudiants désirant pratiquer les expressions régulières).