

Numérique et Sciences InformatiquesDevoir N°3*Calculatrice non autorisée*Durée : 1 heure**Exercice 1:** (5 points)

Une société de location de véhicules dispose d'informations rangées dans un tableur concernant les différents véhicules empruntés et disponibles.

Par exemple, on voit que M. X (dont on a l'adresse mail) a loué un Citroën Berlingo (dont on a les caractéristiques : puissance, année de mise en circulation) immatriculée AA111AB du 7 au 10 octobre 2020.

On voit aussi que M. Y a loué une Audi A4 immatriculée EB100DE du 14 au 22 octobre 2020 (on constate donc qu'elle n'est pas disponible actuellement comme nous sommes le 15).

Construire un schéma relationnel décrivant 4 relations : *Voiture* (pour l'ensemble du parc de véhicules dont dispose l'agence), *Modèle* (pour les descriptif de chaque modèle de véhicules - l'agence ayant parfois plusieurs voitures d'un même modèle), *Location* et *Client*.

Dans ces relations, devront apparaître les clés primaires et secondaires et les associations.

Exercice 2: (5 points)

Un concours pour chiens est organisé, chacun pouvant se présenter dans plusieurs catégories. On souhaite construire une base de donnée relationnelle pour gérer les participants. On propose la création de quatre relations :

Proprietaire (idProprio Int, Nom String, Adresse String)

Race (idRace Int, intitule String, description String)

Chien (idChien Int, proprio Int, race Int, nom String, dateNaissance Date, sexe String, dateAcquis Date)

Resultat (NumInscription Int, idChien Int NOT NULL, NomCatégorie String, classement Int)

On rappelle que les clés primaires de ces relations sont soulignées d'un trait plein et que les clés secondaires sont soulignées par des tirets.

1. Écrire les commandes SQL permettant de créer les tables *Race* et *Chien* dans un SGBD (prendre des types judicieux, autres que TEXT pour les chaînes de caractères).
2. On donne ci-dessous le début de la table *Resultats* :

NumInscription	idChien	NomCatégorie	classement
156		'Agility'	3
'X12'	211	'Jumping'	
157	19	'Agility'	'1er'
157	86	'Jumping'	3

Chercher toutes les anomalies présentes dans cette table.

3. Après avoir corrigé les éventuelles erreurs, écrire les requêtes permettant d'ajouter les deux premières lignes de cette table.

Exercice 3: (4 points)

Une agence bancaire crée une base de données pour gérer les rendez-vous de ses conseillers avec ses clients. On définit les relations :

Client (IdClient Int, Nom String, Prenom String, Adresse String)

Conseiller (IdConseil String, Nom String, Prenom String)

RendezVous (Num Int, Client Int, Conseiller String, Date Date, Heure Time, Duree Int, NouveauClient Boolean)

La durée s'exprime en minutes. Voici une ligne de la relation *RendezVous* :

Num	Client	Conseiller	Date	Heure	Duree	NouveauClient
123	1234512345	'C6356'	'2020-10-15'	'09:15:00'	20	False

Écrire les requêtes en SQL qui permettent d'obtenir ce qui est demandé :

1. Les rendez-vous client-conseiller présents dans la relation ;
2. l'ensemble des rendez-vous (avec toutes les caractéristiques) du conseiller d'identifiant 'C6356' ;
3. la liste des conseillers qui ont eu au moins un rendez-vous avec un client le 7 octobre 2020 ;
4. les dates, heures et durées de tous les rendez-vous du conseiller d'identifiant 'A2341' à partir du 20 octobre 2020.