

Sujet d'examen

Les documents papier sont autorisés. Les appareils électriques sont interdits.

PARTIE I – Modèle relationnel et SQL (14pts)

Une application pour les randonneurs et les gérants d'un parc naturel régional utilise une base de données relationnelle pour stocker entre autres les données sur les lieux, les sentiers, les randonneurs et leurs déplacements. Le schéma de la base de données est présenté ci-dessous. Les clés primaires sont indiquées en gras souligné. Tous les attributs préfixés par ID sont de type entier.

La table LIEUX contient les différents endroits du parc (ex., des points de départ et d'arrivée des sentiers). La table SERVICE indique les services proposés (ex., restauration, toilettes, couchage, poste de secours, zone ludiques, etc.) dans les lieux du parc (voir aussi la table PROPOSE_SERVICE). Les tables SENTIER et RANDONNEURS stockent les données des sentiers et des randonneurs respectivement. Dans SENTIER, *IDlieu1* et *IDlieu2* indiquent les identifiants des lieux représentant les deux extrémités du sentier. La table TRAJET indique les trajets des randonneurs sur les sentiers. Les attributs *DateDepart*/*DateArrivee* sont de type date et incluent l'heure.

```
LIEUX(IDlieu, NomL, Description, Altitude)
SERVICE(IDservice, Type, Description, Telephone, SiteWeb)
PROPOSE_SERVICE(IDlieu, IDservice)
SENTIER(IDsentier, NomS, Description, Difficulte, Longueur,
        IDlieu1, IDlieu2)
RANDONNEUR(Email, Age, Sexe, Niveau)
TRAJET(Email, IDsentier, DateDepart, IDlieuDepart, DateArrivee)
```

Question 1.1 (1pt) : Indiquez les clés étrangères dans ces relations, pour que le modèle soit cohérent. Utilisez la notation TableA.CléEtrangère → TableB.AttributRéféréncé

Question 1.2 (2pts) : Donnez le schéma dans le format entité-association de cette base de telle sorte que le schéma relationnel qui se déduit par la méthode vue en cours de votre schéma E/A est bien celui donné plus haut. Indiquez dans le schéma E/A les clés primaires et les cardinalités des associations.

Question 1.3 (1pt) : Donnez la requête SQL permettant de créer la table SENTIER et ses contraintes d'intégrités (c.à.d., clés primaire et étrangères).

Dans les questions 2.x on demande de donner la formulation SQL des requêtes permettant de répondre aux questions.

Question 2.1 (1pt) : Afficher les sentiers qui partent de « Lac des cygnes » et arrivent à « Grotte de l'ours ».

Question 2.2 (1pt) : Trouver les lieux situés à plus de 1500 mètres d'altitude qui proposent des services de type restauration et couchage.

Question 2.3 (1pt) : Quels sont les lieux que personne n'a visités (si l'on considère les trajets) pendant la journée du 1 janvier 2019.

Question 2.4 (1pt) : Créer la vue *Randonneurs actifs* qui donne les randonneurs les plus mobiles (ayant fait le maximum de trajets) par jour. Afficher l'email du randonneur, le nombre de trajet et la date.

Question 2.5 (1.5pts) : Trouver les sentiers qui partent du lieu nommé la « Motte aux marmottes » et arrivent à la « Prairie des cerfs » soit directs soit passant par un seul lieu intermédiaire. Afficher le

IN507 - Bases de Données

nom des sentiers directs ou le couple de noms pour les trajets composés triés sur la difficulté des sentiers.

Question 2.6 (1.5pts) : Trouver les randonneurs qui ont visité tous les lieux (par rapport aux trajets effectués) dans la période du 24 au 31 décembre 2018.

Question 2.7 (1.5pts) : Afficher les couples de randonneurs qui se sont croisés sur les sentiers (en les parcourant dans des directions opposées) dans la journée du 10 janvier 2019. Pour extraire l'heure à partir d'un attribut de type date utiliser la fonction *Time(attribut_date)*.

Question 2.8 (1.5pts) : Créer la vue *Stats_randonneurs* qui calcule pour chaque randonneur le nombre total de sentiers parcourus, la distance totale et le dénivelé total. Le dénivelé d'un sentier est obtenu en faisant la différence entre l'altitude des points de départ et d'arrivée.

PARTIE II – HTML/PHP et droits d'accès (6pts)

On souhaite construire un site web permettant de visualiser et administrer en ligne les informations stockées dans la base de données de la partie I. On suppose que le site web tourne sur la même machine que la base de données, que le nom de la base est « parc_vert », le login de la base est « user » et le mot de passe « toto ». On suppose que la base de données est déjà remplie.

a) (1pt) Donner les requêtes SQL permettant de créer l'utilisateur « user » et de lui donner l'accès en lecture seule sur les tables LIEUX et SERVICE.

b) (1pts) Donner le code d'une page de recherche contenant un titre « Recherche parcours » et un formulaire de recherche contenant les champs de type texte 'Lieu départ' et 'Lieu arrivée' indiquant les endroits de départ et d'arrivée du parcours, et une liste déroulante permettant d'indiquer le type de parcours désiré : facile (qui évite les sentiers de niveau difficile), rapide (qui prend les sentiers de longueur minimum), lent (qui prend les sentiers de longueur maximum), loisir (qui passe par un maximum de lieux sur le trajet), etc.. Le formulaire sera situé dans une table ayant deux colonnes correspondant aux étiquettes et aux champs du formulaire. À la soumission du formulaire les données seront transmises en utilisant la méthode POST à la page 'affiche_parcours.php'.

c) (4pts) Donner le code PHP de la page 'affiche_parcours.php' pour afficher les résultats de la recherche d'un parcours triés sur le type de parcours choisi (ex., si 'rapide', les parcours sont triés du plus rapide au moins rapide). Vous pouvez considérer que tous les champs du formulaire ont été bien renseignés. Il faut bien évidemment se situer dans le cas général dans lequel le parcours est composé de plusieurs sentiers connectés entre le point de départ et d'arrivée. Pour chaque parcours du résultat, afficher d'abord des informations globales (ex., longueur totale, nombre de sentiers du parcours) et ensuite le détail de chaque sentier dans le parcours. On considère que deux lieux sont connectés si le chemin entre les deux points correspondants couvre au maximum cinq sentiers. Donner le code PHP qui calcule le parcours pour un seul type au choix entre : facile/lent/rapide/loisir.