

Master 1 Humanités numériques – Bases de données

TD 4 : Création, mise à jour et interrogation d'une base de données avec SQL

Cécile Favre – <https://eric.univ-lyon2.fr>

Préambule

Soit le modèle physique de la base de données RANDO, qui permet de gérer des randonnées. Chaque randonnée correspond à un type. Par ailleurs, chaque randonnée peut permettre de voir certains animaux.



Données des tables de RANDO

animal		type_rando		faune	
id_animal	nom_animal	id_type_rando	nom_type_rando	id_rando	id_animal
1	Chamois	1	Randonnée fraîcheur	1	2
2	Marmotte	2	Beaux panoramas	1	4
3	Libellule	3	Randonnée nature	2	4
4	Renard	4	Vestiges et patrimoine	2	7
5	Héron pourpré	5	Lacs et baignades	3	1
6	Grenouille			3	2
7	Epervier			7	3
				7	6

rando

id_rando	nom_rando	lieu_rando	distance_rando	duree_rando	note_difficulte_rando	id_type_rando
1	Gorges du Bruyant	Eydoche	12.6	04:15:00	2	1
2	Côteau de Serrières	Morestel	10.5	03:00:00	1	4
3	Croix de Cassini	Alpe-d'Huez	10	04:00:00	2	2
4	Pierre percée	Pierre-Chatel	8	01:45:00	1	3
5	Sept Laux	Allevard	10.2	07:00:00	3	5
6	Gorges du Bruyant	Engins	5	02:30:00	1	1
7	Étang des Grenouilles	Charantonay	10.2	02:30:00	1	1
8	Aiguille de Chalais	Chalais	4	01:30:00	1	2
9	Le Chatel	Mens	8.3	03:00:00	1	2
10	Le Charmant Som	La Diat	7.6	05:30:00	2	2

LDD – Création des tables

1. Créer la base de données RANDO soit via phpMyAdmin, soit avec la requête SQL suivante : CREATE DATABASE RANDO;
2. Créer chacune des tables (animal, type_rando, rando, faune). Ne pas oublier de spécifier les clés primaires et étrangères.

LMD – Mise à jour des données

1. Importer les données ci-dessus à partir du fichier https://eric.univ-lyon2.fr/jdarmont/docs/rando_data.sql.
2. Ajouter à l'aide d'une requête SQL un n-uplet (ligne) à chaque table de la base de données RANDO.
3. Modifier le *nom_rando* correspondant à l'*id_rando* 5 en "7 Laux".
4. Supprimer à l'aide d'une requête SQL les n-uplets créés à la question 2. Un commentaire ?

LMD – Interrogation des données

Formuler à l'aide de requêtes SQL les interrogations suivantes.

1. Affichage du contenu de la table rando.
2. Toutes les informations concernant les randonnées dont le lieu est « Allevard ».
3. Nom et lieu des randonnées dont le niveau de difficulté est 1.
4. Randonnées lors desquelles il est possible de voir des chamois, des marmottes ou des renards.
5. Durée minimale des randonnées par niveau de difficulté.
6. Nom, lieu et distance des randonnées ayant une difficulté égale à 2, selon un tri décroissant de la distance.

7. Durée maximale des randonnées.
8. Nombre de randonnées dont la distance est strictement supérieure à 10 km (la distance étant exprimée dans la base par un nombre réel qui représente le nombre de kilomètres), par type de randonnée et en indiquant le nom du type de randonnée.
9. Ensemble des informations sur les randonnées et leur type, pour les randonnées qui ont une distance comprise entre 5 km et 10 km ou ont un niveau de difficulté égal à 2.

Correction

-- LDD - Création des tables

```
CREATE TABLE animal (  
  id_animal int(11),  
  nom_animal text,  
  CONSTRAINT PRIMARY KEY (id_animal)  
);
```

```
CREATE TABLE type_rando (  
  id_type_rando int(11),  
  nom_type_rando text,  
  CONSTRAINT PRIMARY KEY (id_type_rando)  
);
```

```
CREATE TABLE rando (  
  id_rando int(11),  
  nom_rando text,  
  lieu_rando text,  
  distance_rando float,  
  duree_rando time,  
  note_difficulte_rando int(11),  
  id_type_rando int(11),  
  CONSTRAINT PRIMARY KEY (id_rando),  
  CONSTRAINT rando_cleetr_type FOREIGN KEY (id_type_rando)  
    REFERENCES type_rando(id_type_rando)  
);
```

```
CREATE TABLE faune (  
  id_rando int(11),  
  id_animal int(11),  
  CONSTRAINT PRIMARY KEY (id_rando, id_animal),  
  CONSTRAINT faune_cleetr_rando FOREIGN KEY (id_rando)  
    REFERENCES rando(id_rando),  
  CONSTRAINT faune_cleetr_animal FOREIGN KEY (id_animal)  
    REFERENCES animal(id_animal)  
);
```

-- LMD - Mise à jour des données

```
INSERT INTO type_rando VALUES (6, 'Randonnée à la cool');  
INSERT INTO animal VALUES (8, 'Lapin blanc') ;  
INSERT INTO rando VALUES (11, 'Mini rando', 'Station', 1, '00:10:00', 1, 6) ;  
INSERT INTO faune VALUES (11, 8);
```

```
UPDATE rando SET nom_rando = '7 Laux' WHERE id_rando = 5;
```

```
DELETE FROM faune WHERE id_rando = 11 AND id_animal = 8;  
DELETE FROM rando WHERE id_rando = 11;  
DELETE FROM animal WHERE id_animal = 8;  
DELETE FROM type_rando WHERE id_type_rando = 6 ;
```

```

-- LMD - Interrogation des données

-- 1
SELECT *
FROM rando;
-- 2
SELECT *
FROM rando
WHERE lieu_rando = 'Allevard';
-- 3
SELECT nom_rando, lieu_rando
FROM rando
WHERE note_difficulte_rando = 1;
-- 4.1
SELECT DISTINCT nom_rando
FROM rando, faune, animal
WHERE rando.id_rando = faune.id_rando
AND faune.id_animal = animal.id_animal
AND nom_animal IN ('Chamois', 'Marmotte', 'Renard');
-- 4.2
SELECT DISTINCT nom_rando
FROM (rando INNER JOIN faune ON rando.id_rando = faune.id_rando)
INNER JOIN animal ON faune.id_animal = animal.id_animal
WHERE nom_animal IN ('Chamois', 'Marmotte', 'Renard');
-- 5
SELECT note_difficulte_rando, min(duree_rando)
FROM rando
GROUP BY note_difficulte_rando;
-- 6
SELECT nom_rando, lieu_rando, distance_rando
FROM rando
WHERE note_difficulte_rando = 2
ORDER BY distance_rando DESC;
-- 7
SELECT max(duree_rando)
FROM rando;
-- 8.1
SELECT nom_type_rando, COUNT(id_rando)
FROM rando, type_rando
WHERE rando.id_type_rando = type_rando.id_type_rando
AND distance_rando > 10
GROUP BY nom_type_rando ;
-- 8.2
SELECT nom_type_rando, COUNT(id_rando)
FROM rando
INNER JOIN type_rando ON rando.id_type_rando = type_rando.id_type_rando
WHERE distance_rando > 10
GROUP BY nom_type_rando;

```

```
-- 9.1
SELECT *
FROM rando, type_rando
WHERE rando.id_type_rando = type_rando.id_type_rando
AND distance_rando BETWEEN 5 AND 10
OR note_difficulte_rando = 2;
-- 9.2
SELECT *
FROM rando
INNER JOIN type_rando ON rando.id_type_rando = type_rando.id_type_rando
WHERE distance_rando BETWEEN 5 AND 10
OR note_difficulte_rando = 2;
```