

Contrôle terminal de BDW1

UCBL - Département Informatique de Lyon 1 – automne 2016

Aucun document ni calculatrice ne sont autorisés. Anonymisez votre copie. Durée : 1h30.

Félicitations, vous venez d'être recruté.e pour travailler sur la nouvelle version d'un jeu éducatif animalier !

Exercice 1 Modélisation (7 points)

Dans la première version du jeu, les animaux sont modélisés en tant que proie et/ou prédateur. Les moyens de défense sont par exemple la fuite, le venin, le camouflage. Pour un animal donné, le jeu permet de connaître sa classification scientifique en partant de son taxon le plus précis (e.g., eucaryotes, primates, vertébrés). Chaque taxon est rattaché à un rang (e.g., règne, famille, espèce) de la classification.

1. En présence d'une base de données existante, une étape de rétro-ingénierie est nécessaire : dessinez un diagramme E/A (formalisme Merise) dont la transformation permettrait d'obtenir le schéma relationnel ci-dessous (prévoir de la place pour étendre votre diagramme, voir question suivante).

RANGS (nomR)
TAXONS (idT, nomT, #nomR, #idTParent)
ANIMAUX (idA, nomA, #idT)
MANGER (#idAproie, #idApred)
DEFENSES (typeD, descD)
DISPOSER (#idA, #typeD)

2. Complétez votre diagramme E/A avec les nouvelles spécifications suivantes. Dans la version étendue du jeu, les animaux se divisent en trois catégories selon leur milieu : terrestre, marin, volant. On considère qu'un animal est forcément associé à l'une des catégories, et qu'il ne peut pas être rattaché à plusieurs. Pour les animaux volants, on stockera le nombre d'ailes. Les animaux terrestres possèdent aussi un ou plusieurs modes de déplacement (un type, par exemple marcher ou ramper, et une description). Les animaux évoluent dans des zones géographiques, qui se caractérisent par un identifiant, un nom, et des coordonnées (latitude, longitude). On distingue les zones où habitent les animaux pendant une ou plusieurs saisons, et les zones où se reproduisent les animaux pendant une ou plusieurs saisons.
3. Si l'on souhaite prendre en compte la contrainte suivante "certains animaux (super-prédateurs) ne sont jamais des proies", que faut-il modifier dans le diagramme ? Justifiez.

Exercice 2 Manipulation de la base (7 points)

Écrivez les requêtes suivantes dans le langage indiqué, en utilisant uniquement la base de données existante (schéma relationnel de l'exercice 1).

1. En CRVT : le nom des taxons et leur rang, pour les taxons avec un identifiant strictement inclus entre 10 et 20
2. En CRVT : le nom des animaux et le type de défense pour les animaux qui sont les seuls à posséder ce type de défense
3. En AR : le nom des animaux qui possèdent le même taxon que le *loup*
4. En SQL : le nom sur les animaux dont l'identifiant de taxon est supérieur à 100 ou qui possèdent un type de défense contenant 'jet de'
5. En SQL : le type et la description des moyens de défense qu'aucun animal ne possède. Le résultat sera ordonné par type décroissant

- En SQL : le nom de tous les animaux, et le nombre de moyens de défense de chaque animal
- En SQL : le nom des animaux qui sont prédateurs de plus de 10 animaux et qui sont proies d'un animal au maximum

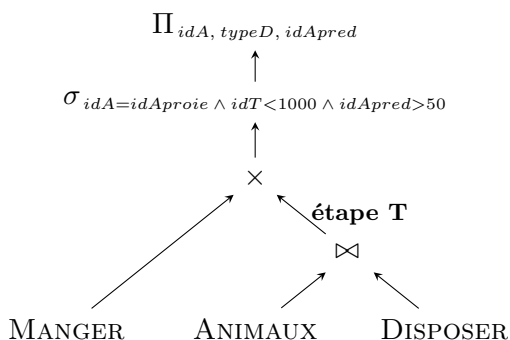
Exercice 3 Mise à jour du site web (3 points)

Vous interagissez maintenant avec l'équipe en charge du site web. Vos collègues vous interrogent sur certains aspects techniques. Cochez les affirmations qui vous semblent correctes.

- Quels sont les avantages ou limitations des méthodes de transmission des paramètres d'un formulaire ?
 - Les paramètres `get` sont facilement modifiables
 - Les paramètres `get` n'acceptent que des caractères ASCII
 - Les paramètres `post` sont limités en taille
 - Les paramètres `post` peuvent être récupérés en HTML
- La technologie Flexbox est un modèle de boîtes qui...
 - Agence des éléments HTML selon trois axes
 - Rend la page adaptative à différents dispositifs
 - S'applique uniquement aux composants "enfant" d'un container
 - Nécessite l'extension *flex*
- Quand on utilise PHP pour interagir avec MySQL ou MariaDB, ...
 - On doit utiliser `mysqli_connect` avant toute exécution de requête
 - La fonction `mysqli_query` retourne toujours un booléen
 - Une requête préparée permet de calculer le résultat en avance
 - La fonction `mysqli_fetch_assoc` stocke le résultat d'une requête dans un tableau associatif

Exercice 4 Optimisation de requête (3 points)

En testant l'application, vous constatez qu'une de vos requêtes met beaucoup trop de temps à s'exécuter. Ci-dessous, le plan d'exécution de cette requête fourni par le SGBD, et un tableau de statistiques sur vos tables (les tailles sont en octets).



	ANIMAUX	MANGER	DISPOSER
Nombre de tuples	100000	150000	50000
Taille d'un tuple	40	10	20
Taille des clés primaires	5	10	20

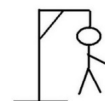
- Quelles sont les données manipulées à l'étape T ? Donnez le nombre de tuples et la taille de ces tuples.
- Dessinez un plan d'exécution optimal en remplacement du plan fourni par le SGBD.

Exercice 5 Questions bonus (1 point)

- Parmi ces animaux, quels sont ceux classifiés comme *poisson* (i.e., de classe *chondrichthyens* ou de super-classe *ostéichthyens*) ?
 - Narval
 - Hippocampe
 - Requin-baleine
 - Ornithorynque
 - Poisson chauve-souris rouge à lèvres

- Un indice pour ce jeu du pendu : *algorithme d'optimisation pour les jointures*.

H _ _ H _ _ O _ N



Lettres déjà proposées :
Y - B - T - U - M