

EXERCICE N°10

LES FONCTIONS

Jean-Pierre Duchesneau, DFC Cégep Ste-Foy

Automne 2018

Évaluation : formative

Travail de préférence individuel.

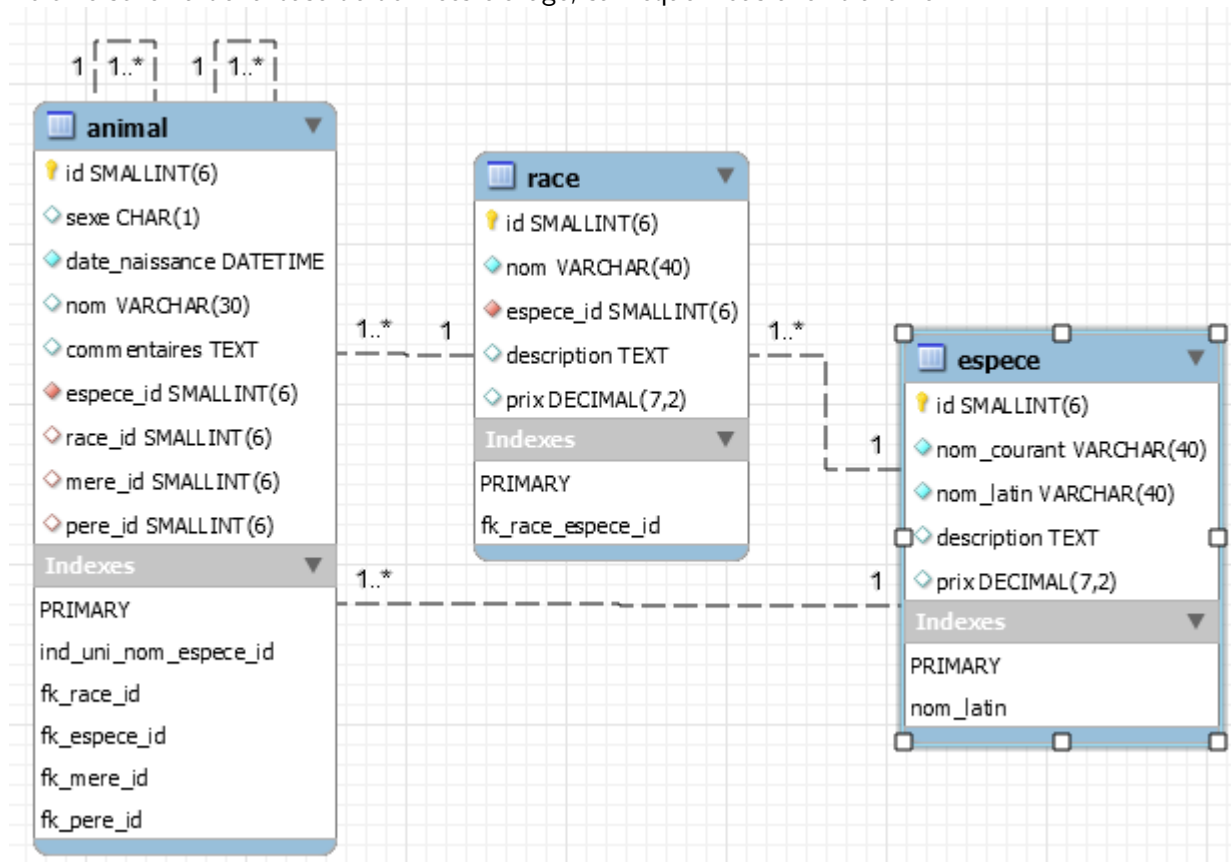
Durée : 2 heures

Environnement : MySQL Serveur 5.7

Fichier SQL : ElevageEx10.sql

1 Le schéma de la base de données

Voici le schéma de la base de données élevage, sur lequel vous allez travailler.



L'ensemble des exercices sont tirés du livre de Chantal Gribaumont¹ chapitres 20 à 23. Seule la mise en page fut modifiée pour nos besoins.

1. Gribaumont, Chantal, "Administrez vous bases de données avec MySQL 2ième édition." Open Classrooms. Disponible sur LÉA.

2 Fonctions scalaires

On commence par du facile

- 1) Afficher une phrase donnant le prix de l'espèce, pour chaque espèce

Par exemple, afficher « Un chat coûte 100 euros. », ou une autre phrase du genre, et ce pour les cinq espèces enregistrées.

- 2) Afficher les chats dont la deuxième lettre du nom est un « a »

Puis on corse un peu

- 3) Afficher les noms des perroquets en remplaçant les « a » par « @ » et les « e » par « 3 » pour en faire des perroquets Kikoolol

Attention : il est tout à fait possible d'imbriquer plusieurs fonctions, ici c'est le cas avec la fonction REPLACE().

- 4) Afficher les chiens dont le nom a un nombre pair de lettres

2.1 Réponse

- 1) Afficher une phrase donnant le prix de l'espèce, pour chaque espèce

Par exemple, afficher « Un chat coûte 100 euros. », ou une autre phrase du genre, et ce pour les cinq espèces enregistrées.

```
SELECT CONCAT( 'Un(e) ', nom_courant, ' coûte ', prix, ' euros.'
) AS Solution
FROM Espece;

— OU

SELECT CONCAT_WS( ' ', 'Un(e)', nom_courant, 'coûte', prix, '
euros.') AS Solution
FROM Espece;
```

- 2) Afficher les chats dont la deuxième lettre du nom est un « a »

```
SELECT Animal.nom , Espece.nom_courant
FROM Animal
INNER JOIN Espece ON Animal.espece_id = Espece.id
WHERE Espece.nom_courant = 'Chat '
AND SUBSTRING(nom , 2, 1) = 'a';
```

- 3) Afficher les noms des perroquets en remplaçant les « a » par « @ » et les « e » par « 3 » pour en faire des perroquets Kikoolol

Attention : il est tout à fait possible d'imbriquer plusieurs fonctions, ici c'est le cas avec la fonction REPLACE().

```
SELECT REPLACE(REPLACE(nom , 'a' , '@') , 'e' , '3') AS Solution
FROM Animal
INNER JOIN Espece ON Animal.espece_id = Espece.id
WHERE Espece.nom_courant LIKE 'Perroquet%';
```

Une petite explication s'impose avant de vous laisser continuer. Comme vous voyez, il est tout à fait possible d'imbriquer plusieurs fonctions. Le tout est de le faire correctement, et pour cela, il faut procéder par étape. Ici, vous voulez faire deux remplacements successifs dans une chaîne de caractères (en l'occurrence, le nom des perroquets). Donc, vous effectuez un premier remplacement, en changeant les "a" par les "@" : `REPLACE(nom, 'a', '@')`. Ensuite, sur la chaîne résultant de ce premier remplacement, vous effectuez le second : `REPLACE(REPLACE(nom, 'a', '@'), 'e', '3')`. Logique, non ?

- 4) Afficher les chiens dont le nom a un nombre pair de lettres

```
SELECT nom , nom_courant
FROM Animal
INNER JOIN Espece ON Animal.espece_id = Espece.id
WHERE nom_courant = 'Chien '
AND CHAR_LENGTH(nom)%2 = 0;

— OU
SELECT nom , nom_courant
FROM Animal
INNER JOIN Espece ON Animal.espece_id = Espece.id
WHERE nom_courant = 'Chien '
AND CHAR_LENGTH(nom) MOD 2 = 0;

— OU
SELECT nom , nom_courant
FROM Animal
INNER JOIN Espece ON Animal.espece_id = Espece.id
WHERE nom_courant = 'Chien '
AND MOD(CHAR_LENGTH(nom),2) = 0;
```

Le nombre de lettres, c'était facile, il suffisait d'utiliser `CHAR_LENGTH()`. Pour savoir si un nombre est pair, il faut utiliser les modulus : lorsqu'un nombre est pair, le reste d'une division entière de ce nombre par 2 est 0, donc ce nombre modulo 2 vaut 0.

3 Fonctions d'agrégation

- 1) Combien de races avons-nous dans la table Race ?
- 2) De combien de chiens connaissons-nous le père ?
- 3) Quelle est la date de naissance de notre plus jeune femelle ?

- 4) En moyenne, quel est le prix d'un chien ou d'un chat de race, par espèce, et en général ?
- 5) Combien avons-nous de perroquets mâles et femelles, et quels sont leurs noms (en une seule requête bien sûr) ?
- 6) Quelles sont les races dont nous ne possédons aucun individu ?
- 7) Quelles sont les espèces (triées par ordre alphabétique du nom latin) dont nous possédons moins de cinq mâles ?
- 8) Combien de mâles et de femelles de chaque race avons-nous, avec un compte total intermédiaire pour les races (mâles et femelles confondues) et pour les espèces ? Afficher le nom de la race, et le nom courant de l'espèce.
- 9) Quel serait le coût, par espèce et au total, de l'adoption de Parlotte, Spoutnik, Caribou, Cartouche, Cali, Canaille, Yoda, Zambo et Lulla ? Petit indice, pour avoir le prix d'un animal selon que sa race soit définie ou non, vous pouvez utiliser une fonction que nous venons de voir au chapitre précédent.

3.1 Réponses

- 1) Combien de races avons-nous dans la table Race ?

```
SELECT COUNT (*)  
FROM Race ;
```

- 2) De combien de chiens connaissons-nous le père ?

```
SELECT COUNT(pere_id)  
FROM Animal  
INNER JOIN Espece ON Espece.id = Animal.espece_id  
WHERE Espece.nom_courant = 'Chien ' ;
```

L'astuce ici était de ne pas oublier de donner la colonne `pere_id` en paramètre à `COUNT()`, pour ne compter que les lignes où `pere_id` est non NULL. Si vous avez fait directement `WHERE espece_id = 1` au lieu d'utiliser une jointure pour sélectionner les chiens, ce n'est pas bien grave.

- 3) Quelle est la date de naissance de notre plus jeune femelle ?

```
SELECT MAX(date_naissance)  
FROM Animal  
WHERE sexe = 'F' ;
```

- 4) En moyenne, quel est le prix d'un chien ou d'un chat de race, par espèce, et en général ?

```
SELECT nom_courant AS Espece , AVG(Race.prix) AS prix_moyen
FROM Race
    INNER JOIN Espece ON Race.espece_id = Espece.id
WHERE Espece.nom_courant IN ( 'Chat ' , 'Chien ' )
GROUP BY Espece.nom_courant WITH ROLLUP;
```

Ne pas oublier WITH ROLLUP pour avoir le résultat général.

- 5) Combien avons-nous de perroquets mâles et femelles, et quels sont leurs noms (en une seule requête bien sûr) ?

```
SELECT sexe , COUNT (*) , GROUP_CONCAT(nom SEPARATOR ' , ')
FROM Animal
    INNER JOIN Espece ON Animal.espece_id = Espece.id
WHERE nom_courant = 'Perroquet amazone '
GROUP BY sexe;
```

Il suffisait de se souvenir de la méthode GROUP_CONCAT() pour pouvoir réaliser simplement cette requête. Peut-être avez-vous groupé sur l'espèce aussi (avec nom_courant ou autre). Ce n'était pas nécessaire puisqu'on avait restreint à une seule espèce avec la clause WHERE. Cependant, cela n'influe pas sur le résultat, mais sur la rapidité de la requête.

- 6) Quelles sont les races dont nous ne possédons aucun individu ?

```
SELECT Race.nom , COUNT(Animal.race_id) AS nombre
FROM Race
    LEFT JOIN Animal ON Animal.race_id = Race.id
GROUP BY Race.nom
HAVING nombre = 0;
```

Il fallait ici ne pas oublier de faire une jointure externe (LEFT ou RIGHT, selon votre requête), ainsi que de mettre la colonne Animal.race_id (ou Animal.id, ou Animal.espece_id mais c'est moins intuitif) en paramètre de la fonction COUNT().

- 7) Quelles sont les espèces (triées par ordre alphabétique du nom latin) dont nous possédons moins de cinq mâles ?

```
SELECT Espece.nom_latin , COUNT(espece_id) AS nombre
FROM Espece
    LEFT JOIN Animal ON Animal.espece_id = Espece.id
WHERE sexe = 'M' OR Animal.id IS NULL
GROUP BY Espece.nom_latin
HAVING nombre < 5;
```

À nouveau, une jointure externe et `espece_id` en argument de `COUNT()`, mais il y avait ici une petite subtilité en plus. Puisqu'on demandait des informations sur les mâles uniquement, il fallait une condition `WHERE sexe = 'M'`. Mais cette condition fait que les lignes de la jointure provenant de la table `Espece` n'ayant aucune correspondance dans la table `Animal` sont éliminées également (puisque forcément, toutes les colonnes de la table `Animal`, dont `sexe`, seront à `NULL` pour ces lignes). Par conséquent, il fallait ajouter une condition permettant de garder ces fameuses lignes (les espèces pour lesquelles on n'a aucun individu, donc aucun mâle). Il fallait donc ajouter `OR Animal.id IS NULL`, ou faire cette condition sur toute autre colonne d'`Animal` ayant la contrainte `NOT NULL`, et qui donc ne sera `NULL` que lors d'une jointure externe, en cas de noncorrespondance avec l'autre table. Il n'y a plus alors qu'à ajouter la clause `HAVING` pour sélectionner les espèces ayant moins de cinq mâles.

- 8) Combien de mâles et de femelles de chaque race avons-nous, avec un compte total intermédiaire pour les races (mâles et femelles confondus) et pour les espèces? Afficher le nom de la race, et le nom courant de l'espèce.

```
SELECT Animal.sexe , Race.nom , Espece.nom_courant , COUNT (*) AS
nombre
FROM Animal
  INNER JOIN Espece ON Animal.espece_id = Espece.id
  INNER JOIN Race ON Animal.race_id = Race.id
WHERE Animal.sexe IS NOT NULL
GROUP BY Espece.nom_courant , Race.nom , sexe WITH ROLLUP;
```

Deux jointures sont nécessaires pour pouvoir afficher les noms des races et des espèces. Il suffit alors de ne pas oublier l'option `WITH ROLLUP` et de mettre les critères de regroupement dans le bon ordre pour avoir les super-agrégats voulus.

- 9) Quel serait le coût, par espèce et au total, de l'adoption de Parlotte, Spoutnik, Caribou, Cartouche, Cali, Canaille, Yoda, Zambo et Lulla?

```
SELECT Espece.nom_courant , SUM(COALESCE(Race.prix , Espece.prix)
) AS somme
FROM Animal
  INNER JOIN Espece ON Espece.id = Animal.espece_id
  LEFT JOIN Race ON Race.id = Animal.race_id
WHERE Animal.nom IN ( 'Parlotte ' , 'Spoutnik ' , 'Caribou ' , '
Cartouche ' , 'Cali ' , 'Canaille ' , 'Yoda ' , 'Zambo ' , 'Lulla ' )
GROUP BY Espece.nom_courant WITH ROLLUP;
```

C'est ici la fonction `SUM()` qu'il fallait utiliser, puisqu'on veut le prix total par groupe. Sans oublier le `WITH ROLLUP` pour avoir également le prix total tous groupes confondus. Quant au prix de chaque animal, c'est typiquement une situation où l'on peut utiliser `COALESCE()`!

Fin de l'exercice 10.

Sommaire

1	Le schéma de la base de données	1
2	Fonctions scalaires	2
2.1	Réponse	2
3	Fonctions d'agrégation	3
3.1	Réponses	4
	Sommaire	7

Ce document a été écrit avec LaTeX.

Cette oeuvre, création, site ou texte est sous licence Creative Commons Attribution - Pas d'Utilisation commerciale - Partage dans les Mêmes Conditions 4.0 International. Pour accéder à une copie de cette licence, merci de vous rendre à l'adresse suivante <http://creativecommons.org/licenses/by-nc-sa/4.0/> ou envoyez un courrier à Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.