Sous-requêtes

# Sous-requête renvoyant une valeur

**SELECT** id, sexe, nom, commentaires, espece\_id, race\_id

**FROM** Animal

**WHERE** race\_id =

(**SELECT** id **FROM** Race **WHERE** nom = 'Berger Allemand'); *-- la*

*sous-requête renvoie simplement 1*

Equivalent à

**SELECT** Animal.id, sexe, Animal.nom, commentaires, Animal.espece\_id,

race\_id

**FROM** Animal

**INNER JOIN** Race **ON** Race.id = Animal.race\_id

**WHERE** Race.nom = 'Berger Allemand';

Voici un exemple de requête avec sous-requête qu'il est impossible de faire avec une simple jointure :

**SELECT** id, nom, espece\_id

**FROM** Race

**WHERE** espece\_id = (

**SELECT MIN**(id) *-- Je rappelle que MIN() permet de récupérer*

*la plus petite valeur de la colonne parmi les lignes sélectionnées*

**FROM** Espece);

Autre exemple :

**SELECT** id, nom, espece\_id

**FROM** Race

**WHERE** espece\_id < (

**SELECT** id

**FROM** Espece

**WHERE** nom\_courant = 'Tortue d''Hermann');

# Conditions avec IN et NOT IN

**SELECT** id, nom, espece\_id

**FROM** Animal

**WHERE** espece\_id **IN** (

**SELECT** id

**FROM** Espece

**WHERE** nom\_courant **IN** ('Tortue d''Hermann', 'Perroquet amazone')

);

Fonctionne également avec NOT IN

# Conditions avec ANY, SOME et ALL

**ANY** : veut dire "au moins une des valeurs".

**SOME** : est un synonyme de **ANY**.

**ALL** : signifie "toutes les valeurs".

***ANY (ou SOME)***

La requête suivante signifie donc "Sélectionne les lignes de la table *Animal*, dont l'*espece\_id* est inférieur à **au moins une** des valeurs sélectionnées dans la sous-requête". C'est-à-dire inférieur à 3 **ou** à 4. Vous aurez donc dans les résultats toutes les lignes dont l'*espece\_id* vaut 1, 2 ou 3 (puisque 3 est inférieur à 4).

**SELECT** \*

**FROM** Animal

**WHERE** espece\_id < **ANY** (

**SELECT** id

**FROM** Espece

**WHERE** nom\_courant **IN** ('Tortue d''Hermann', 'Perroquet amazone')

);

***ALL***

Par contre, si vous utilisez **ALL** plutôt que **ANY**, cela signifiera "Sélectionne les lignes de la table *Animal*, dont l'*espece\_id* est inférieur à **toutes** les valeurs sélectionnées dans la sous-requête". Donc inférieur à 3 **et** à 4. Vous n'aurez donc plus que les lignes dont l'*espece\_id* vaut 1 ou 2.

**SELECT** \*

**FROM** Animal

**WHERE** espece\_id < **ALL** (

**SELECT** id

**FROM** Espece

**WHERE** nom\_courant **IN** ('Tortue d''Hermann', 'Perroquet amazone')

);

# Conditions avec EXISTS et NOT EXISTS

**SELECT** id, nom, espece\_id **FROM** Race

**WHERE EXISTS** (**SELECT** \* **FROM** Animal **WHERE** nom = 'Balou');

Se rapproche du if … then en Java ☺

**Exemple** : je veux sélectionner toutes les races dont on ne possède aucun animal.

**SELECT** \* **FROM** Race

**WHERE NOT EXISTS** (**SELECT** \* **FROM** Animal **WHERE** Animal.race\_id = Race.id);

Ici, la sous-requête est appelée sous-requête corrélée car elle est liée à la requête principale. En effet, elle fait appel à la table Race désignée dans le FROM de la requête principale.

# Modification de données

## Insertion

Syntaxe :

**INSERT INTO** nom\_table

[(colonne1, colonne2, ...)]

**SELECT** [colonne1, colonne2, ...]

**FROM** nom\_table2

[**WHERE** ...]

Exemple :

INSERT INTO Customers (CustomerName, City, Country)  
SELECT SupplierName, City, Country FROM Suppliers;

Autre exemple :

**INSERT INTO** Animal

(nom, sexe, date\_naissance, race\_id, espece\_id) *--*

*Je précise les colonnes puisque je ne donne pas une valeur pour*

*toutes.*

**SELECT** 'Yoda', 'M', '2010-11-09', id **AS** race\_id, espece\_id *--*

*Attention à l'ordre !*

**FROM** Race **WHERE** nom = 'Maine coon';

## Modification

### Pour la sélection

**UPDATE** Animal **SET** commentaires = 'Coco veut un gâteau !' **WHERE**

espece\_id =

(**SELECT** id **FROM** Espece **WHERE** nom\_courant **LIKE** 'Perroquet%');

### Pour l’élément à modifier

**UPDATE** Animal **SET** race\_id =

(**SELECT** id **FROM** Race **WHERE** nom = 'Nebelung' **AND** espece\_id = 2)

**WHERE** nom = 'Cawette';

**Important !** On ne peut pas utiliser dans la sous-requête la table qu’on est en train de modifier.

L’exemple suivant ne fonctionne pas :

**UPDATE** Animal **SET** race\_id =

(**SELECT** race\_id **FROM** Animal **WHERE** nom = 'Cawette' **AND** espece\_id

= 2)

**WHERE** nom = 'Callune';

## Suppression

**DELETE FROM** Animal

**WHERE** nom = 'Carabistouille' **AND** espece\_id =

(**SELECT** id **FROM** Espece **WHERE** nom\_courant = 'Chat');

**Même remarque** que pour la modification : on ne peut pas supprimer depuis une table si elle est utilisée dans la sous-requête.