

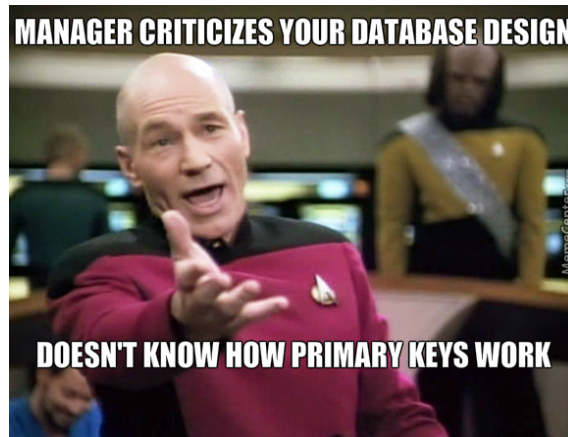
T. D. n° 6

Clés primaires et étrangères en MySQL

Résumé

Ce document est le TD n°6 du module **Base de données**. Il reprend rapidement des éléments du cours et propose une mise en pratique interactive des clés primaires et étrangères.

1 Clés primaire



source :Memecenter

Reprenez votre base de donnée. Cette base contient la table *Animal* doit être composée *a minima* de six colonnes : *id*, *espece*, *sexe*, *date_naissance*, *nom* et *commentaires*.

Les clés sont intimement liées aux index. Et tout comme NOT NULL et les index UNIQUE, les clés font partie de ce que l'on appelle les contraintes. Il existe deux types de clés :

- les clés primaires, qui ont déjà été survolées lors du chapitre sur la création d'une table, et qui servent à identifier une ligne de manière unique ;
- les clés étrangères, qui permettent de gérer des relations entre plusieurs tables, et garantissent la cohérence des données

1.1 Création et suppression d'une clé primaire

La clé primaire peut être créée en même temps que la table, ou par la suite. Lors de la création de la table on peut préciser PRIMARY KEY dans la description de la colonne

qui doit devenir la clé primaire (pas de clé composite, c'est à dire sur plusieurs colonnes, dans ce cas). Deux syntaxes sont possibles :

-- Syntaxe 1

```
CREATE TABLE [IF NOT EXISTS] Nom_table (  
    colonne1 description_colonne1 PRIMARY KEY [,  
    colonne2 description_colonne2 ,  
    colonne3 description_colonne3 ,  
    ...,]  
)  
  
[ENGINE=moteur];
```

-- Syntaxe 2

```
CREATE TABLE [IF NOT EXISTS] Nom_table (  
    colonne1 description_colonne1 [,  
    colonne2 description_colonne2 ,  
    colonne3 description_colonne3 ,  
    ...],  
    [CONSTRAINT symbole_contrainte] PRIMARY KEY (colonne_pk1 [,  
        colonne_pk2, ...])  
)  
  
[ENGINE=moteur];
```

Comme pour les index **UNIQUE**, **CONSTRAINT** est facultatif. Il permet de nommer la clé (ici `symbole_contrainte`), ce qui est vivement recommandé.

Pour déclarer une clé primaire après création de la table on utilise **ALTER TABLE**. Attention : si vous créez une clé primaire sur une table existante, assurez-vous que la ou les colonnes où vous souhaitez l'ajouter ne contiennent pas **NULL**.

```
ALTER TABLE nom_table  
  
ADD [CONSTRAINT [symbole_contrainte]] PRIMARY KEY (colonne_pk1 [,  
    colonne_pk2, ...]);
```

Pour supprimer la clé primaire, vous pouvez utiliser **DROP PRIMARY KEY**. Pas besoin de préciser de quelle clé il s'agit, puisqu'il ne peut y en avoir qu'une seule par table !

```
ALTER TABLE nom_table  
DROP PRIMARY KEY
```

À vous !

- a) Quelle est la différence entre les syntaxes 1 et 2 précédentes ?
- b) Quelle est la clef primaire de la table *Animal* ?
- c) Creez une table *Animal2* comprenant une colonne *id* (small int auto incrémenté jamais nul), *espece* (chaîne de caractères limité à 40), *sexe* (F ou M), *nom* (chaîne de caractères limité à 30) et *commentaires* (chaîne de caractères sans limite). Définissez l'*id* comme la clé primaire.
- d) Creez une table *Animal3* comprenant une colonne *id* (small int jamais nul), *espece* (chaîne de caractères limité à 40), *sexe* (F ou M), *nom* (chaîne de caractères limité à 30) et *commentaires* (chaîne de caractères sans limite).
- e) Définissez l'*id* comme la clé primaire de la table *Animal3* après la création de cette dernière.
- f) Supprimez la clé primaire de *Animal3*.
- g) Supprimez la table *Animal2* et la table *Animal3*.

2 Clé étrangère

Une clé étrangère est un peu plus complexe à créer qu'un index ou une clé primaire, puisqu'il faut deux éléments :

- la ou les colonnes sur lesquelles on crée la clé : on utilise **FOREIGN KEY** ;
- la ou les colonnes qui vont servir de référence : on utilise **REFERENCES**.

```
CREATE TABLE [IF NOT EXISTS] Nom_table (  
  
    colonne1 description_colonne1,  
  
    [colonne2 description_colonne2,  
  
    colonne3 description_colonne3,  
  
    ...,]  
  
    [ [CONSTRAINT [symbole_contrainte]] FOREIGN KEY (colonne(s)_clé_étrangère) REFERENCES table_référence (colonne(s)_référence)]  
  
)  
  
[ENGINE=moteur];
```

Tout comme pour les clés primaires, pour créer une clé étrangère après création de la table, il faut utiliser **ALTER TABLE**.

Par exemple :

```
ALTER TABLE Commande
```

```
ADD CONSTRAINT fk_client_numero FOREIGN KEY (client) REFERENCES
  Client(numero);
```

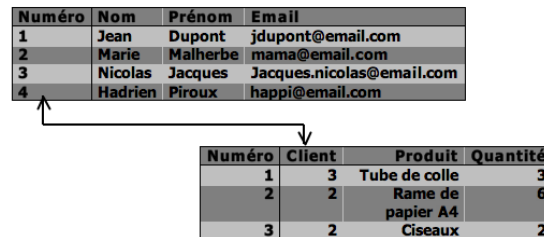


FIGURE 1 – Exemple d’une bdd de gestion de commandes

Il peut y avoir plusieurs clés étrangères par table. Par conséquent, lors d’une suppression, il faut identifier la clé à détruire. Cela se fait grâce au symbole de la contrainte.

```
ALTER TABLE nom_table
```

```
DROP FOREIGN KEY symbole_contrainte
```

À vous !

- Quel index est automatiquement créé lors de la création d’une clé étrangère ?
- Quel index doit préalablement exister sur la table de référence lors de la création d’une clé étrangère ?
- Créez une table *Client* avec les conditions suivantes
 - numéro, de type integer jamais négatif qui se génère automatiquement (et défini comme la clé primaire)
 - nom, de type caractère (limité à 40)
 - prénom, de type caractère (limité à 40)
 - email, de type caractère (limité à 40)
 - commentaire, de type texte
 - une moteur de stockage innnoDB
- Créez une table *Commande* avec les conditions suivantes
 - numéro, de type integer jamais négatif qui se génère automatiquement (et défini comme la clé primaire)
 - client, de type integer jamais négatif et jamais null
 - produit, de type caractère (limité à 40)
 - quantité, de type integer (limité à 4 octets)
 - une clé étrangère (nommée *fk_client_numero*) sur la colonne *client*, faisant référence à la colonne *numéro* de la table *Client*
 - une moteur de stockage innnoDB

- e) Essayer de créer une clé étrangère sur la colonne *commentaire* de la table *Client*, faisant référence à la colonne *numéro* de la table *Commande*
- f) Que pouvez vous conclure sur les règles de typage pour les clés étrangères ?
- g) Remplir vos tables en suivant la figure 1.

3 Retour à la nature

Il est temps de modifier votre base et ajouter quelques tables afin de préparer le prochain chapitre, qui portera sur les jointures.

Pour l'instant, la seule information dont nous disposions était le nom courant de l'espèce d'un animal. Nous voudrions maintenant stocker aussi son nom latin, ainsi qu'une petite description. Ajouter deux colonnes à notre table *nom_latin_espece* et *description_espece* est une très mauvaise solution. En effet, cela obligerait à stocker la même description pour chaque chien, chaque chat, etc. Ainsi que le même nom latin. Nous le faisons déjà avec le nom courant, et cela aurait déjà pu poser problème. Imaginez que, pour un animal, vous fassiez une faute d'orthographe au nom de l'espèce, "chein" au lieu de "chien", par exemple. L'animal en question n'apparaîtrait jamais lorsque vous feriez une recherche par espèce.

Il faudrait donc un système qui nous permet de ne pas répéter la même information plusieurs fois, et qui limite les erreurs que l'on pourrait faire.

La bonne solution est de créer une seconde table : la table *Espec*. Cette table aura 4 colonnes : le nom courant, le nom latin, une description, et un numéro d'identification (qui sera la clé primaire de cette table).

À vous !

- a) Créer la table *Espec* en choisissant judicieusement les types de variables possible. Ces variables seront appelées *id*, *nom_courant*, *nom_latin* et *description*.
- b) Si ce n'est pas déjà fait, créer une clé primaire sur l'attribut qui vous semble le plus pertinent.
- c) L'utilisateur de la base de données souhaite être sûr que l'on ne rentrera pas deux fois la même espèce. Selon vous, faut-il mettre une contrainte sur le nom latin ou sur le nom courant (Justifiez).
- d) modifiez votre table pour répondre au souhait de l'utilisateur
- e) Ajouter les animaux suivants dans votre base de données
 - un chien nommé *Canis canis*, avec le commentaire : 'Bestiole à quatre pattes qui aime les caresses et tire souvent la langue',
 - un chat nommé '*Felis silvestris*', avec le commentaire 'Bestiole à quatre pattes qui saute très haut et grimpe aux arbres',
 - une tortue d'Hermann', nommée '*Testudo hermanni*', avec le commentaire 'Bestiole avec une carapace très dure',

- un Perroquet amazone nommé *Alipiopsitta xanthops*, avec le commentaire Joli oiseau parleur vert et jaune'
- f) Modifier notre table *Animal* en ajoutant une colonne *espece_id*, qui contiendra l'id de l'espèce à laquelle appartient l'animal,
- g) Remplir cette colonne.
- h) Supprimer la colonne *espece*, de votre table *animaux*.
- i) définissez la colonne *espece_id* sera une clé étrangère ayant pour référence la colonne *id* de la table *Espece*.
- j) Ajouter un animal dont l'*espece_id* n'existe donc pas dans la table *Espece*. Quel message apparaît ?
- k) Ajouter une contrainte : "jamais nul" sur l'*espece_id*
- l) Ajouter un index **UNIQUE**

4 Plus de tables !

L'utilisateur de votre base de données souhaite pouvoir reconnaître un chien d'un chat, mais reconnaître également un berger allemand d'un teckel, non ?

Par conséquent, nous allons encore ajouter deux choses à notre base. D'une part, nous allons ajouter une table *Race*, basée sur le même schéma que la table *Espece*. Il faudra également ajouter une colonne à la table *Animal*, qui contiendra l'id de la race de l'animal. Contrairement à la colonne *espece_id*, celle-ci pourra être **NULL**. Il n'est pas impossible que nous accueillions des bâtards, ou que certaines espèces que nous élevons ne soient pas classées en plusieurs races différentes. Ensuite, nous allons garder une trace du pedigree de nos animaux. Pour ce faire, il faut pouvoir connaître ses parents. Donc, nous ajouterons deux colonnes à la table *Animal* : *pere_id* et *mere_id*, qui contiendront respectivement l'id du père et l'id de la mère de l'animal.

nom	espece_id	description
Berger allemand	1	Chien sportif et élégant au pelage dense, noir-marron-fauve, noir ou gris.
Berger blanc suisse	1	Petit chien au corps compact, avec des pattes courtes mais bien proportionnées et au pelage tricolore ou bicolore
Boxer	1	Chien de taille moyenne, au poil ras de couleur fauve ou bringé avec quelques marques blanches.
Bleu russe	2	Chat aux yeux verts et à la robe épaisse et argentée.
Maine coon	2	Chat de grande taille, à poils mi-longs.
Singapura	2	Chat de petite taille aux grands yeux en amandes.
Sphynx	2	Chat sans poils.

TABLE 1 – Races possibles

À vous !

- a) Créer la table Race. Cette dernière doit comprendre un id (sa clef primaire), un nom, un espece_id, une description. Vous ajouterez également une clef étrangère fk_race_espece_id qui sera associée à l'attribut id de la table Espece pour assurer l'intégrité de la référence.
- b) Insérer les races listées dans le tableau 1.
- c) Ajouter la colonne race_id à la table Animal.
- d) Ajouter la clef étrangère fk_race_id sur l'attribut race_id de la table Animal et qui fera référence à l'attribut id de la table Race.
- e) Renseigner la race_id de vos animaux dans la table Animal.
- f) Ajouter les colonnes mere_id et pere_id à la table Animal. Définir des clés étrangères (fk_mere_id et fk_pere_id) sur ces colonnes, faisant référence à l'attribut id de la table Animal.
- g) Définir un père et une mère pour au moins 4 animaux de votre table.