

## Ajout, suppression et modification des données

### 1 Valeurs NULL

En SQL, la valeur NULL est une constante particulière qui indique que l'information est manquante. Le comportement de NULL diffère un peu des autres valeurs.

- On teste si une colonne a la valeur NULL avec `col IS NULL`. On utilise `col IS NOT NULL` pour tester si une colonne n'est pas NULL.
  - Les valeurs NULL sont traitées différemment des autres valeurs par les opérateurs booléens, arithmétiques ou d'agrégations. Par exemple, `(NULL OR False)` ne renvoie ni vrai ni faux mais NULL.
1. Les requêtes `SELECT * FROM t` et `SELECT * FROM t WHERE col<0 or col>=0` renvoient-elles la même valeur ? Pourquoi ?
  2. Les requêtes `SELECT COUNT(capital) FROM world.country` et `SELECT COUNT(countrycode) FROM world.country` renvoient-elles la même valeur ?
  3. Quels sont les pays dont l'espérance de vie ou le PNB est inconnu ?
  4. Combien de pays n'ont pas de capitale ?
  5. Quels sont les pays dont toutes les villes ont moins de 1000 habitants ? Attention, il y a des pays qui n'ont pas de villes et qui, logiquement, respectent cette condition.
  6. Combien de clients de sakila ne vivent pas en France ? Attention, il se pourrait que certains clients n'aient pas spécifié leur adresse (champ `address_id` est NULL).
  7. Quels sont les films dont tous les exemplaires DVD n'ont jamais été retournés ?

### 2 INSERT INTO, UPDATE et DELETE

On rappelle que `INSERT INTO` permet d'insérer des données dans une table, `DELETE` de les supprimer et `UPDATE` de modifier certaines lignes.

Quel effet ont les requêtes suivantes sur la base de données ?

1. `INSERT INTO world.country (name_country, countrycode) VALUES ('Atlantide', 'ATL');`
2. `INSERT INTO sakila.country (country) VALUES (SELECT name_country FROM world.country);`
3. `DELETE FROM world.city;`
4. `DELETE FROM world.country WHERE countrycode != (SELECT countrycode FROM world.city WHERE id=capital);`
5. `UPDATE world.country SET lifeexpectancy = NULL WHERE population_country = 0;`
6. 

```
WITH LastRental AS
(SELECT customer_id as c, MAX(rental_date) as d FROM sakila.rental
GROUP BY customer_id)

DELETE FROM sakila.customer WHERE
EXTRACT(days FROM NOW())-(SELECT d FROM LastRental WHERE customer_id =
c)) > 365;
```

### 3 CREATE TABLE

```
CREATE TABLE table_name
(
column_name1 type,
column_name2 type,
...
);
```

créer une nouvelle table dans la base de données avec les colonnes spécifiées. Par exemple `CREATE TABLE livre(titre VARCHAR(200), auteur VARCHAR(100), resume TEXT, prix INT)`;

On peut remplir la table avec le résultat d'une requête au moment de son initialisation avec `CREATE TABLE name(col1 type,...,coln type) AS requete`. En cas d'erreur, on peut supprimer une table avec la commande `DROP TABLE name`.

1. Proposez une façon de copier une table dans une autre.

## 4 TP

Toutes les opérations modifiant la base de données nécessitent d'avoir les droits d'écritures sur la table ou le schéma concernés. Vous n'avez pas les droits d'écriture dans les schémas `sakila`, `world` etc. mais vous avez les droits d'écriture dans votre schéma personnel (votre identifiant ENT).

On va créer une table `CountryCapital` dans votre schéma qui doit pouvoir contenir le code `countrycode`, le nom `name_country`, le nom de la capitale `name_capital` et le nom de la plus grande ville `name_biggest_city` d'un pays.

1. Quel est le type de chacune des colonnes de `CountryCapital` ?
2. Remplissez-là à partir des informations du schéma `world`. Les pays n'ayant pas de villes doivent avoir les champs `name_capital` et `name_biggest_city` à NULL.
3. Modifier la table `CountryCapital` pour que tous les noms apparaissant dedans soient mis en majuscules. On utilisera la fonction `UPPER` qui prend une chaîne de caractères en argument et renvoie la même chaîne en majuscule.
4. Supprimez tous les pays de `CountryCapital` qui ont une population inférieure à 5000.
5. Créer une table `continent` avec les colonnes `continent_id` qui contient le numéro d'identification d'un continent donné par la table `world.code_cotinent` et les colonnes `name_continent`, `surface`, `population`, `lifeexpectancy` et `gnp` qui contient respectivement le nom, la surface et la population totale et l'espérance de vie et le PNB moyen par pays du continent.

## 5 DM

1. (a) Créez dans votre schéma, une *table* `moncontinent`, copie partielle de la *table* `world.country`. Votre copie contiendra les tuples correspondants aux pays du continent dont le code (donné par la table `world.code_continent`) est égal à `1 + (officialcode %7)` où `officialcode` est votre numéro d'inscription à Paris-Diderot (212xxxxx si vous vous êtes inscrit pour la première fois en 2012).  
(b) Insérer dans votre table, un tuple correspondant à un pays imaginaire.
2. Créer dans votre schéma, une *vue* `ville_pays`, de schéma (`pays`, `ville`, `est_capitale`, `pourcentage_pop`), où `pays` est le nom du pays, `est_capitale` vaut TRUE si l'attribut `ville` est le nom de la capitale du pays désigné par l'attribut `pays`, et FALSE sinon, et `pourcentage_pop` désigne le pourcentage de la population du pays résidant dans la ville.