

## 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 ?

**Solution:** Si la colonne `col` peut prendre la valeur NULL (peut ne pas être renseignée), les résultats peuvent être différents. La seconde requête est équivalente à `SELECT * FROM t WHERE col IS NOT NULL`.

2. Les requêtes `SELECT COUNT(capital) FROM world.country` et `SELECT COUNT(countrycode) FROM world.country` renvoient-elles la même valeur ?

**Solution:** Non. La première renvoie 232, la seconde renvoie 239. L'agrégation `COUNT` ne compte que les valeurs non nulles. On obtient le même résultat que la première requête avec `SELECT COUNT(countrycode) FILTER (WHERE capital IS NOT NULL) FROM world.country`.

3. Quels sont les pays dont l'espérance de vie ou le PNB est inconnu ?

**Solution:** Les 17 pays peuvent être listés par `SELECT countrycode FROM world.country WHERE lifeexpectancy IS NULL OR gnp IS NULL`. Ce sont en fait les pays où `lifeexpectancy` n'est pas renseigné et où la colonne `gnp` vaut 0.

4. Combien de pays n'ont pas de capitale ?

**Solution:** Nous avons déjà répondu à la question :  $239 - 232 = 7$ . Par exemple avec : `SELECT COUNT(countrycode) FILTER (WHERE capital IS NULL) FROM world.country`

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.

**Solution:** En calcul relationnel cela donne

$$\left\{ \text{nom} : \exists co \quad \text{country}(co) \wedge co.\text{name\_country} = \text{nom} \wedge \neg \left( \exists ci \quad \text{city}(ci) \wedge ci.\text{countrycode} = co.\text{countrycode} \wedge ci.\text{population\_city} > 1000 \right) \right\}$$

La requête suivante ne répond pas à la question

```
SELECT name_country
FROM world.country NATURAL JOIN world.city ci
WHERE ci.population_city < 1000 ;
```

La requête suivant répond à la question

```
SELECT name_country
FROM world.country co
WHERE NOT EXISTS (
  SELECT *
  FROM world.city ci
  WHERE ci.countrycode=co.countrycode AND ci.population_city>1000) ;
```

(15 pays)

C'est la traduction directe de la requête exprimée en calcul relationnel.

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).

**Solution:** En fait le schéma complet de `customer` spécifie que `address_id` ne peut être NULL. On peut donc répondre par

```
SELECT COUNT(cu.customer_id)
FROM sakila.customer cu NATURAL JOIN sakila.address NATURAL JOIN
      sakila.city NATURAL JOIN sakila.country co
WHERE co.country <> 'France' ;
```

(595)

7. Quels sont les films dont tous les exemplaires DVD n'ont jamais été retournés ?

**Solution:** Cette question est de même forme que la question 5. On peut la traduire en quels sont les films dont il n'existe pas d'exemplaire DVD retourné. Cela comprend les films qui n'ont jamais été loués.

```
SELECT title
FROM sakila.film fi
WHERE NOT EXISTS (
  SELECT *
  FROM sakila.inventory inv NATURAL JOIN sakila.rental re
  WHERE inv.film_id = fi.film_id AND re.return_date IS NOT NULL
) ;
```

Les 42 films listés sont en fait ceux pour lesquels il n'existe pas de DVD dans l'inventaire.

## 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 ?

Alex: on les fait créer une copie de `world.country` dans leur schéma pour qu'il puissent tester ? "create table as select \* from... " ça ne va pas conserver les contraintes ? Si ?

1. `INSERT INTO world.country (name_country , countrycode) VALUES ('Atlantide' , 'ATL');`

**Solution:**

Ajoute le pays dont le nom est Atlantide et le code est ATL. Les colonnes dont la valeur n'a pas été spécifiée prennent la valeur NULL (ou possiblement une valeur par défaut dépendant de la table).

En fait, il ne se passe rien, car de nombreuses colonnes de `world.country` sont NOT NULL. Il se passe quelque chose si on travaille sur une copie de `country` où les contraintes NOT NULL n'ont pas été spécifiées.

2. `INSERT INTO sakila.country (country) VALUES (SELECT name_country FROM world.country);`

**Solution:** Insère les noms pays de la table `world.country` dans la table `sakila.country`.

Les colonnes `country_id` et `lastupdate_country` sont renseignées automatiquement.

3. `DELETE FROM world.city;`

**Solution:** Il ne se passe rien, car une suppression violerait les contraintes référentielles entre les tables `city` et `country` (une ville doit se situer dans un pays).

Alex: ce n'est pas plutôt des pays ont une capital qui doit être dans city ??  
ou alors la requête est plutôt "delete from world.country"

4. `DELETE FROM world.country WHERE countrycode != (SELECT countrycode FROM world.city WHERE id=capital);`

**Solution:** Supprimerait les pays de `world.country` dont la capitale indiquée n'est pas dans ce pays.

Alex: ce qui me réconforte dans ma remarque précédente

5. `UPDATE world.country SET lifeexpectancy = NULL WHERE population_country = 0;`

**Solution:** Met le champ `lifeexpectancy` à NULL si la population du pays est 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;`

**Solution:**

Supprime les clients inactifs depuis plus d'un an.

### 3 CREATE TABLE

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

crée 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.

**Solution:** Pour constituer une copie de `world.country`

```
CREATE TABLE entid.country AS  
SELECT * FROM world.country ;
```

### 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` ?

**Solution:** Pour `CountryCapital`, `character(3)`, `text`, `text`, `text`.

Pour lister les types des colonnes d'une table :

```
SELECT column_name, data_type  
FROM information_schema.columns  
WHERE table_name='country' AND table_schema='world' ;
```

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`.

**Solution:**

```
CREATE TABLE entid.countrycapital AS  
SELECT co.countrycode, co.name_country, ci.name_city as name_capital ,  
  entid.city(co.name_country) name_biggest_city  
FROM world.country co LEFT OUTER JOIN world.city ci ON  
  (co.capital=ci.id) ;
```

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.

**Solution:**

```
UPDATE entid.countrycapital
SET name_country = upper(name_country),
    name_capital = upper(name_capital)
    biggest_city = upper(biggest_city);
```

4. Supprimez tous les pays de `CountryCapital` qui ont une population inférieure à 5000.

**Solution:**

```
DELETE FROM entid.countrycapital cc
USING world.country co
WHERE co.countrycode=cc.countrycode AND co.population_country < 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.

**Solution:**

```
CREATE TABLE entid.continent AS
SELECT codecontinent code_continent, continent as name_continent,
    sum(surfacearea) surface, sum(population_country) population, avg(
    lifeexpectancy) lifeexpectancy, avg(gnp) gnp
FROM world.code_continent ct NATURAL JOIN world.country co
GROUP BY codecontinent, continent ;
```

NB : Faire la moyenne des espérances de vie et des gnp sans pondérer par les populations respectives est aberrant.

```
CREATE TABLE entid.continent AS
SELECT code_continent, continent as name_continent, sum(surface),
    sum(population_country),
    sum(population_country*lifeexpectancy)/sum(population_country),
    sum(population_country*gnp)/sum(population_country)
FROM world.code_continent ct NATURAL JOIN world.country co
GROUP BY code_continent, continent ;
```

## 5 DM

Alex: on fait qqch du coup ?

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 (214xxxxx si vous vous êtes inscrit pour la première fois en 2014).

(b) Insérer dans votre table, un tuple correspondant à un pays imaginaire.

2. TODO