

Suivi des décès

Base de données et page web

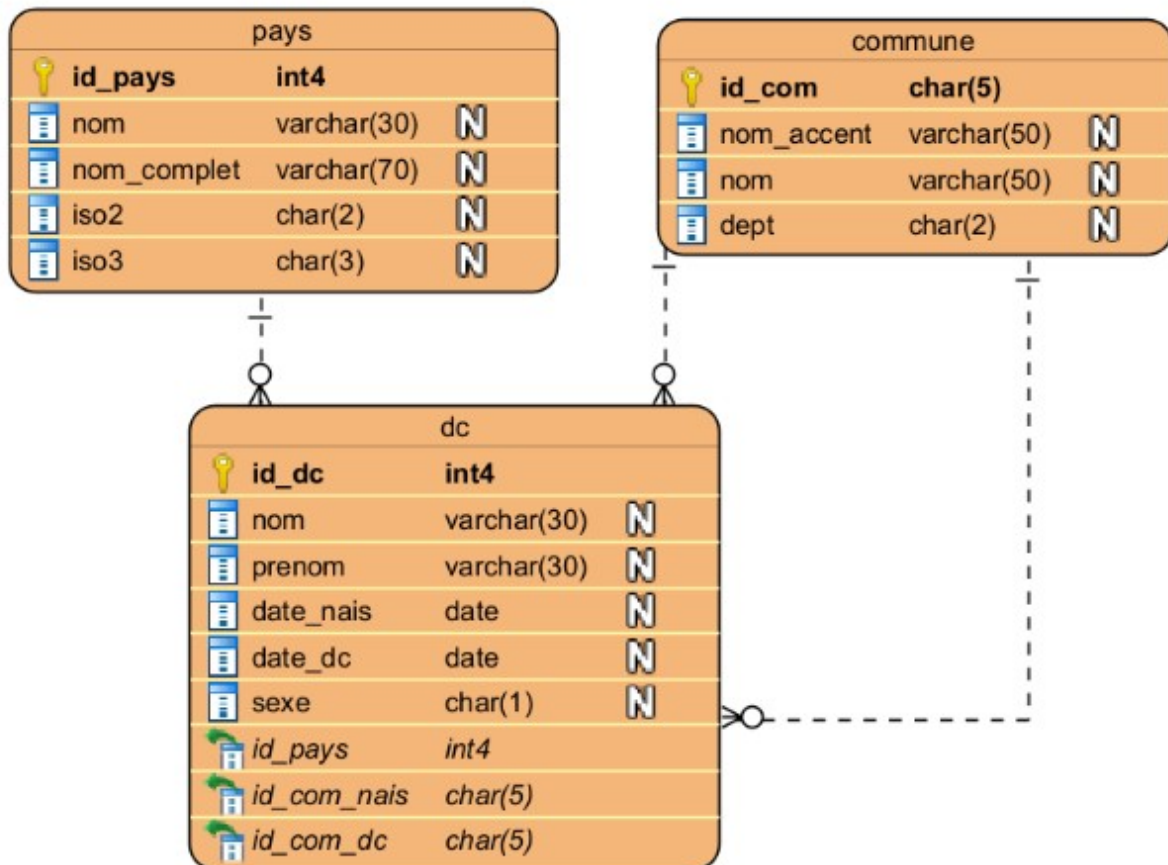
Suivi des décès

Objectif.....	3
I. Base de données :.....	3
Script de création de la base :.....	3
Scripts de création de tables :.....	3
Scripts de migration communes et pays :.....	3
Script de migration des décès :.....	3
Migration complète vers la base :.....	3
Vérification du remplissage des tables :.....	3
II. Développement web – Accès à la base.....	3
Page web.....	3

Objectif

L'objectif était de créer une base de données (PostgreSQL) pouvant être consultée depuis une page web sous PHP.

Cette base de données est composée de 3 tables (communes, pays et dc) :



Ces tables seront remplies par les données des fichiers .csv créés par le programme Fix2csv :

- deces-2021-m01.csv
- deces-2021-m02.csv
- deces-2021-m03.csv
- deces-2021-m04.csv
- deces-2021-m05.csv
- deces-2021-m06.csv
- deces-2021-m07.csv

I. Base de données :

dc_admin.sql

```
-- Script de creation de la base de donnees
create database dc;
create role dc_admin password 'admin' login;
grant all on database dc to dc_admin;
```

Script de création de la base :

```
-- Debut de la transaction
BEGIN TRANSACTION;

-- Création de la table pays
CREATE TABLE pays
(
    id_pays    SERIAL PRIMARY KEY,
    nom        varchar(30),
    nom_complet varchar(70),
    iso2       char(2),
    iso3       char(3)
);

-- Création de la table commune
CREATE TABLE commune
(
    id_com     char(5) PRIMARY KEY,
    nom_accent varchar(50),
    nom        varchar(50),
    dept       char(2)
);

-- Création de la table dc
CREATE TABLE dc
(
    id_dc      SERIAL PRIMARY KEY,
    nom        varchar(30),
    prenom     varchar(30),
    date_nais  date,
    date_dc    date,
    sexe       char(1),
    id_pays    integer REFERENCES pays (id_pays),
    id_com_nais char(5) REFERENCES commune (id_com),
    id_com_dc   char(5) REFERENCES commune (id_com)
);

-- Fin de la transaction
COMMIT;
```

Scripts de création de tables :

Connexion en tant que dc_admin :

```
dc=> \i public_html/dc/suivi/migration_base/dc_create.sql
BEGIN
CREATE TABLE
CREATE TABLE
CREATE TABLE
COMMIT
```

Scripts de migration communes et pays :

Pour ces 2 scripts, nous allons créer une table « tempo » contenant les champs du fichier à importer puis nous allons effectuer une requête permettant d'alimenter la table voulue.

Pour lancer ces scripts, il faut à la base « dc » en tant que DBA :

```
postgres=# \c dc
You are now connected to database "dc" as user "postgres".
dc=# █
```

Migration_commune.sql

```
begin transaction;

-- Crée la table tempo (table temporaire)
create table tempo
(
  typecom varchar(50),
  com varchar(50),
  reg varchar(50),
  dep varchar(50),
  arr varchar(50),
  tncc varchar(50),
  ncc varchar(500),
  nccenr varchar(500),
  libelle varchar(500),
  can varchar(50),
  comparent varchar(50)
);

-- Copiage des données du fichier communes vers la table tempo
copy tempo from '/home/cha/public_html/suivi_dc/fic/communes2020.csv' with csv delimiter ',' NULL '' encoding 'utf8';

-- Insertion dans la table commune des données copiées dans la table tempo
insert into commune(id_com, nom, nom_accnt, dept)
select com, ncc, libelle, dep::char(2)
  from tempo
  where typecom <> 'COMD';

-- Efface la table tempo
drop table tempo;

commit;
vacuum full;
```

Migration_pays.sql

```
begin transaction;

-- Crée la table tempo (table temporaire)
create table tempo
(
  cog varchar(50),
  actual varchar(50),
  capay varchar(50),
  crpay varchar(50),
  ani varchar(50),
  libcog varchar(50),
  libenr varchar(500),
  ancnom varchar(50),
  codeiso2 varchar(50),
  codeiso3 varchar(50),
  codenum3 varchar(50)
);

-- Copiage des données du fichier des pays vers la table tempo
copy tempo from '/home/cha/public_html/suivi_dc/fic/pays2020.csv' with csv delimiter ',' NULL '' encoding 'utf8';

-- Insertion dans la table pays des données copiées de la table tempo
insert into pays(nom, nom_complet, iso2, iso3)
select libcog::varchar(30), libenr, codeiso2::char(2), codeiso3::char(3)
  from tempo;

-- Efface la table tempo
drop table tempo;
commit;
vacuum full;
```

Difficultés rencontrées :

- Taille de certaines colonnes de la table « tempo » supérieures à la table « commune » ou « pays » :
 - Conversion des colonnes : dep::char(2) / libcog::varchar(30)
- Doublons de clé primaire de <commune> :
 - Ecarter les lignes qui ont comme type DCOM

Script de migration des décès :

Migration_dc.sql

```
begin transaction;

-- Crée la table tempo (table temporaire)
create table tempo
(
    nom varchar(100),
    prenom varchar(100),
    sexe varchar(100),
    date_nais varchar(100),
    code_com_nais varchar(100),
    pays_nais varchar(100),
    date_dc varchar(100),
    code_com_dc varchar(100)
);

-- Copiage des données des fichiers CSV dans la table tempo
copy tempo from '/home/cha/public_html/suivi_dc/fic/deces-2021-m01.csv' with csv delimiter ';' NULL '' encoding 'utf8';
copy tempo from '/home/cha/public_html/suivi_dc/fic/deces-2021-m02.csv' with csv delimiter ';' NULL '' encoding 'utf8';
copy tempo from '/home/cha/public_html/suivi_dc/fic/deces-2021-m03.csv' with csv delimiter ';' NULL '' encoding 'utf8';
copy tempo from '/home/cha/public_html/suivi_dc/fic/deces-2021-m04.csv' with csv delimiter ';' NULL '' encoding 'utf8';
copy tempo from '/home/cha/public_html/suivi_dc/fic/deces-2021-m05.csv' with csv delimiter ';' NULL '' encoding 'utf8';
copy tempo from '/home/cha/public_html/suivi_dc/fic/deces-2021-m06.csv' with csv delimiter ';' NULL '' encoding 'utf8';
copy tempo from '/home/cha/public_html/suivi_dc/fic/deces-2021-m07.csv' with csv delimiter ';' NULL '' encoding 'utf8';
copy tempo from '/home/cha/public_html/suivi_dc/fic/deces-2021-m08.csv' with csv delimiter ';' NULL '' encoding 'utf8';
copy tempo from '/home/cha/public_html/suivi_dc/fic/deces-2021-m09.csv' with csv delimiter ';' NULL '' encoding 'utf8';

-- Insertion dans la table dc des données copiées de la table tempo
insert into dc(nom,prenom,date_nais,date_dc,sexe,id_pays,id_com_nais,id_com_dc)
select tempo.nom::varchar(30),prenom::varchar(30),date_nais::date,date_dc::date,sexe,id_pays,c1.id_com,c2.id_com
from tempo
left join pays on pays.nom = tempo.pays_nais
left join commune c1 on c1.id_com = code_com_nais
left join commune c2 on c2.id_com = code_com_dc;

-- Efface la table tempo
drop table tempo;

commit;
vacuum full;
```

Difficultés rencontrées :

- code commune des fichiers décès qui ne sont pas dans la table communes
- idem pour les noms de pays

Résolutions :

- ✓ Si code commune n'existe pas, indiquer une clé étrangère nulle
- ✓ idem pour les pays

II. Développement web – Accès à la base

Page web

RECHERCHE D'UN DÉCÈS

 ?

Formulaire

```
<h1>RECHERCHE D'UN DÉCÈS</h1>
<form action="cherche_p2.php" method="get">
  <input type="text" name="mot" value="">
  <input type="submit" value="?">
</form>
```

Récupération du mot, affichage de l'entête de la table et connexion à la base

```
if(isset($_GET['mot']))
{
  // Mot à chercher
  $cherche = strtoupper($_GET['mot']);

  echo '<table border="1">';
  echo '<tr>';
  echo '<th>Num</th>';
  echo '<th>Nom</th>';
  echo '<th>Prénom</th>';
  echo '<th>Pays</th>';
  echo '</tr>';
  echo '<br>';

  $conn = pg_connect('host=127.0.0.1 dbname=dc user=dc admin password=admin');
```

Paramétrage de la pagination

```
// Mise en place des paramètres pour la pagination
if(!isset($_GET['offset']))
{
  // Chaîne qui contient la requête comptabilisant le nombre de ligne du résultat
  $string_req = "select count(*) from dc left join pays on pays.id_pays = dc.id_pays where dc.nom like '%".$cherche."%";
  // lancement de la requête
  $req = pg_query($string_req);
  // Récupération de la première colonne de la première ligne
  $taille = pg_fetch_row($req)[0];
  // Première utilisation de l'offset, celui-ci vaudra 0
  $soff=0;
}
else
{
  // Il y a déjà un offset, on le récupère
  $soff = $_GET['offset'];
  // La taille est passée en paramètre pour éviter de lancer la requête
  // de calcul à chaque appel de la page.
  $taille = $_GET['taille'];
}
```


Requête SQL et affichage du tableau contenant les données

```
$string_req = "select dc.nom, prenom, pays.nom as pays from dc left join pays on pays.id_pays = dc.id_pays";
$string_req .= " where dc.nom like '%.$cherche.'" order by dc.nom, prenom";
// On modifie la requête pour tenir compte de l'offset
$string_req .= ' offset '.$soff.' limit 20';
$req = pg_query($string_req);
$stab = pg_fetch_assoc($req);
// Pour permettre la numérotation des lignes
$inc = 1;
while($stab)
{
    echo '<tr>';
    echo '<td>'.($soff+$inc++).'</td>';
    echo '<td>'. $stab['nom'].'</td>';
    echo '<td>'. $stab['prenom'].'</td>';
    echo '<td>'. $stab['pays'].'</td>';
    echo '</tr>';
    $stab = pg_fetch_assoc($req);
}
pg_close($conn);
echo "</table>\n";
```

Dans cette requête, l'offset permet de définir à partir de quelle ligne on récupère le résultat, la limite définit le nombre de lignes à retourner. Il suffira donc d'ajouter 20 à la variable \$soff afin d'afficher les 20 autres décès sur la page suivante.

Zone de déplacement

```
// Zone de déplacement dans les enregistrements

// affichage des pages
for($i=0; $i<=$taille; $i=$i+20)
{
    $string_page = 'cherche_p2.php?mot='.$cherche.'&offset='.$i.'&taille='.$taille;
    echo '&nbsp;<a href="'.$string_page.'">'.(int)($i/20).'</a>&nbsp;';
}
}
```

« offset » correspond au nombre de réponses contenues sur l'ensemble des pages précédentes

« taille » correspond au nombre total de décès correspondant au mot passé en paramètre