

# TDM de Bases de Données

## SQL EMBARQUÉ DANS LE LANGAGE C

La chaîne de production de programme C intégrant du SQL est la suivante :

### 1) Pré-processeur SQL

```
ecpg -o nom.c nom.sql -I /usr/include/postgresql/
```

Génère le fichier *nom.c* à partir du fichier *nom.sql*

Fichier source **.sql** :  
Langage C + SQL

### 1) PRE PROCESSEUR SQL

Fichier source **.c** :  
Langage C uniquement

### 2) Compilation

```
cc nom.c -o nom -I /usr/include/postgresql -lecpg
```

Le fichier *nom.c* est généré par le pré processeur SQL. Le développeur n'intervient donc jamais sur ce fichier.

En cas d'erreur aux étapes 1) ou 2), c'est uniquement dans le fichier *nom.sql* que le développeur effectue les corrections et retourne à l'étape 1).

### 2) COMPILATEUR C

Fichier objet :  
binaire uniquement

### 3) Exécution du programme

```
./nom ou bien ./nom | more
```

### 3) EXECUTION

### Documentation

Cours et "ECPG - Embedded SQL in C" du manuel postgresql.

(<http://www.postgresql.org/>)

**SECURITÉ:**

**AU DÉBUT DE CHAQUE SÉANCE, changer votre mot de passe habituel, avec la commande `passwd`, par un mot de passe "jetable" qui ne vous servira que pour la durée de la séance. En effet, votre mot de passe devra figurer en clair dans le fichier `nom.sql`.**

**À LA FIN DE CHAQUE SÉANCE, changer votre mot de passe "jetable" par votre mot de passe habituel.**

---

**Question 1**

**Avec votre base de données *BDlogin* contenant les tables et les données Tennis**

- Ecrire un programme C et SQL `consult_tennis.sql` qui affiche le nom et l'année de naissance des joueurs qui ont participé au tournoi de Roland Garros en 1989.
- Pour votre culture, examiner le fichier résultat du pré-processeur en le comparant au fichier `.sql` correspondant.

---

**Question 2**

Ouvrez deux fenêtres de terminal : dans l'une connectez vous à votre base de données ***BDlogin*** avec `psql` et dans l'autre, compilez et exécutez le programme décrit ci-dessous :

- Ecrire un programme C et SQL `update_tennis.sql` qui :
  - augment de 10 000 la prime des joueurs ayant gagné une rencontre à Roland Garros.
  - et ensuite affiche les noms et les nouvelles valeurs des primes des joueurs ayant gagné une rencontre à Roland Garros
- Exécutez votre programme puis, dans votre fenêtre `psql`, écrivez une requête qui affiche les noms et les nouvelles valeurs des primes des joueurs ayant gagné une rencontre à Roland Garros. Que constatez-vous ? Expliquez ce qu'il se passe.
- Modifier le programme précédent afin que les changements opérés par votre programme soient validés et visibles dans votre fenêtre `psql`.

**Question 3**

Ecrire un programme interactif en C et SQL `consult_bdtennis_dyn.sql` qui demande à l'utilisateur d'entrer 2 années de naissances, A1 et A2, et qui affiche le nom et l'année de naissance de chaque joueur dont l'année de naissance  $\in [A1 ; A2]$ , tel que

ci-contre :    `Annee de naissance min du joueur ? 1970`  
                  `Annee de naissance max du joueur ? 1980`

Chang	1972
Bruguera	1971
Courier	1970
(3 ligne(s))	

**Précisions techniques**

- La continuation d'une boucle, qui parcourt le résultat d'une requête, se fait tant que :

```
sqlca.sqlcode==0
```

- La connexion à votre base de données s'effectue de la façon suivante:

```
exec sql connect
to BDvotre_login@opale
user votre_login using mot_de_passe_jetable;
```

**NB :** Tous vos programmes doivent :

- se préoccuper des erreurs survenues lors de l'exécution des requêtes SQL en examinant la valeur de `sqlca.sqlcode` :

```
if (sqlca.sqlcode<0)
printf ("ERREUR XX: %s\n", sqlca.sqlerrm.sqlerrmc);
```

Remplacer XX par le nom de la clause SQL (`open`, `connect`, `disconnect`, `fetch`, etc) en cause

- inclure **OBLIGATOIREMENT** une déconnexion avec le serveur avant de se terminer !!