

GBA 4^{ème} année - T.P. Base de données

Sujet : Étudiants partant en stage à l'étranger

© Polytech Lille

1 Objectif et indications

Ce TP a pour objectif de comparer l'utilisation des tableurs (e.g. Libreoffice Calc, Excel) aux bases de données relationnels. Vous devez prendre note des avantages, difficultés et différences des approches (e.g. stockage des données, facilité d'assurer la cohérence, facilité d'utilisation, travailler avec des grandes quantités de données, ...).

Nous vous fournissons deux copies des données : une dans un fichier `.ods` que vous pouvez ouvrir avec *Libreoffice Calc*, et la deuxième copie dans des fichiers `.sql` que vous devez utiliser pour créer la base de données. Notez par exemple la structuration des données dans les deux approches.

On vous demande de répondre deux fois à chaque question, une fois avec le tableur et l'autre avec une requête SQL. Faites une question à la fois, en passant d'une technologie à l'autre. Vous êtes libre d'utiliser des fonctions avancés de Calc (fonctions, formules, filtres, ...).

2 Initialisation de la partie Tableur (Libreoffice Calc / Excel)

Récupérer le fichier `BDExcel.ods` qui contient les données "*Étudiants partants à l'étranger*" avec la commande :

```
cp ~wrudamet/public/GBA/BDExcel.ods .
```

3 Initialisation de la partie SQL (Postgres)

La définition des relations et des données "*Étudiants partants à l'étranger*" se trouvent respectivement dans les fichiers `relationsEtudiants.sql` et `donneesEtudiants.sql`. Exécutez les commandes suivantes en remplaçant **VotreLogin** par votre compte postgres et **xx** par vos initiales.

Pour copier les fichiers dans le répertoire courant, utilisez la commande `cp` :

```
cp ~wrudamet/public/GBA/relationsEtudiants.sql .
cp ~wrudamet/public/GBA/donneesEtudiants.sql .
```

Positionnez le variable d'environnement qui spécifie le serveur postgres à utiliser :

```
export PGHOST=serveur-etu.polytech-lille.fr
```

Configurez l'affichage de requêtes dans postgres pour permettre de visualiser beaucoup de données :

```
export PAGER='less -SFMX'
```

Créez la base avec le nom `xxetudiants` où `xx` sont vos initiales (par exemple pour Blaise Pascal ce serait `bpetudiants`):

```
createdb -U VotreLogin xxetudiants
Password: postgres
```

Ensuite connectez vous à votre base de données étudiants sur le serveur houplin :

```
psql -U VotreLogin xxetudiants
Password: postgres
```

Vous pouvez maintenant créer les relations (*c.a.d.* les tables) et insérez les données en utilisant nos fichiers `sql` préalablement copiés :

```
\i relationsEtudiants.sql
\i donneesEtudiants.sql
```

Finalement vous pouvez lister les relations avec la commande :

```
\d
```

4 Requêtes SQL

Votre base de données est prête, maintenant vous pouvez interagir avec vos données. Par exemple, pour lister tous les tuples de la table `ETUDIANT`, utilisez la requête suivante :

```
SELECT * FROM etudiant ;
```

Ensuite, pour vous donner une idée du contenu de la base essayez une requête plus complexe (vous pouvez copier/coller les requêtes suivantes). Attention aux conditions de jointure, aux clés primaires composés de plus d'un attribut, et aux attributs de même nom qui demandent d'enlever les ambiguïtés (e.g. l'attribut `dept` se trouve dans deux relations différents). Notez bien quelles sont les tables listés dans chaque requête. Naviguez avec les flèches.

Requête qui affiche 6 des 9 tables présents dans la base

```
SELECT *
FROM etudiant, formation, effectue, stage, pays, entreprise
WHERE stage.etudiant=numEt AND entreprise=numEn AND pays=numPa
AND stage.formation=numFo AND stage.dept=formation.dept
AND effectue.stage=numSt AND effectue.etudiant=numEt ;
```

Requête qui affiche tous les tables sauf Département (8/9 tables). La projection (partie du SELECT) reproduit la feuille Excel du TP1 : regardez les colonnes choisis, les calculs de durée, et l'ordonnement des résultats par numéro d'étudiant.

```
SELECT stage.dept, numfo, CONCAT(anneedeb || '-' || anneefin) as annee, numEt,
Etudiant.civilite, nomEt, prenomEt, datedeb, datefin,
(datefin::timestamp - datedeb::timestamp) as duree_en_jours,
job, fax as fax_signe_le, convention as convention_signee_le,
nomPa, nomEn, Contact.civilite, Contact.fonction, ville,
sujet, remuneration
FROM etudiant, formation, effectue, stage, pays, entreprise, suit, contact
WHERE stage.etudiant=numEt AND stage.entreprise=numEn AND pays=numPa
AND stage.formation=numFo AND stage.dept=formation.dept
AND effectue.stage=numSt AND effectue.etudiant=numEt
AND formation.numFo=suit.formation AND formation.dept=suit.dept
AND suit.etudiant=numEt
AND contact.entreprise=numEn AND contact.stage=stage.numSt
ORDER BY numet ;
```

5 Exercices

Répondez aux questions suivantes en utilisant *Libreoffice Calc* et en écrivant des requêtes SQL :

1. Quels étudiants sont partis à l'étranger pour leur stage découverte en entreprise ?
2. Combien d'étudiants sont partis en Chine en 2012 ?
3. Quel est le nom et le prénom des étudiants partis plusieurs fois à l'étranger ?
4. Combien d'étudiants sont partis plusieurs fois à l'étranger ?
5. Où a été Remi Carton ?
6. Quels sont les organismes américains qui ont reçu des étudiants de Polytech ?
7. Quel est le nom et le pays des entreprises qui ont reçu plusieurs étudiants ?
8. Quelle est la rémunération moyenne des étudiants ?
9. Quels sont les étudiants qui sont partis en pays anglophones (c'est-à-dire États Unis, Australie, Irlande, Royaume Uni, Canada).
10. Quel est le département qui a envoyé le plus d'étudiants à l'étranger ?
11. Quel est le sujet de stage Charlotte Capelle ?
12. Quels sont les contacts de l'INRS ?
13. Sur quelle période, Justine Vanneste a-t-elle effectué son stage ?
14. Quel a été le lieu et le sujet de stage des étudiants partis en Pologne ?
15. Quel étudiant a été le plus rémunéré en stage à l'étranger ?

```

1: --Effacer les tables existants (erreur pas important si
   elles n'existent pas)
2: DROP TABLE effectuee;
3: DROP TABLE suit;
4: DROP TABLE Contact;
5: DROP TABLE Stage;
6: DROP TABLE Formation;
7: DROP TABLE Departement;
8: DROP TABLE Entreprise;
9: DROP TABLE Pays;
10: DROP TABLE Etudiant;
11:
12:
13: --Creer les tables
14: CREATE TABLE Etudiant (
15:   numEt INTEGER ,
16:   nomEt TEXT ,
17:   prenomEt TEXT ,
18:   civilite TEXT ,
19:   PRIMARY KEY (numEt)
20: ) ;
21:
22: CREATE TABLE Pays (
23:   numPa INTEGER,
24:   nomPa TEXT ,
25:   PRIMARY KEY (numPa)
26: ) ;
27:
28: CREATE TABLE Entreprise (
29:   numEn INTEGER ,
30:   nomEn TEXT ,
31:   PRIMARY KEY (numEn)
32: ) ;
33:
34: CREATE TABLE Departement (
35:   nomCourt TEXT ,
36:   nomLong TEXT ,
37:   PRIMARY KEY (nomCourt)
38: ) ;
39:
40: CREATE TABLE Formation (
41:   numFo INTEGER ,
42:   dept TEXT REFERENCES Departement,
43:   PRIMARY KEY (numFo, dept)
44: ) ;

```

```

45: CREATE TABLE Stage (
46:   numSt INTEGER ,
47:   sujet TEXT ,
48:   job BOOLEAN ,
49:   remuneration FLOAT ,
50:   convention DATE ,
51:   fax DATE ,
52:   ville TEXT ,
53:   pays INTEGER REFERENCES Pays,
54:   entreprise INTEGER REFERENCES Entreprise,
55:   etudiant INTEGER REFERENCES Etudiant,
56:   formation INTEGER ,
57:   dept TEXT ,
58:   dateDeb DATE ,
59:   dateFin DATE ,
60:   FOREIGN KEY (formation, dept) REFERENCES Formation
   (numFo, dept),
61:   PRIMARY KEY (numSt)
62: ) ;
63:
64: CREATE TABLE Contact (
65:   numCo INTEGER ,
66:   civilite TEXT ,
67:   fonction TEXT ,
68:   entreprise INTEGER REFERENCES Entreprise,
69:   stage INTEGER REFERENCES Stage,
70:   PRIMARY KEY (numCo)
71: ) ;
72:
73: CREATE TABLE suit (
74:   formation INTEGER ,
75:   dept TEXT ,
76:   etudiant INTEGER REFERENCES Etudiant,
77:   anneeDeb INTEGER ,
78:   anneeFin INTEGER ,
79:   PRIMARY KEY(formation, dept, etudiant),
80:   FOREIGN KEY (formation, dept) REFERENCES Formation
   (numFo, dept)
81: ) ;
82:
83: CREATE TABLE effectue (
84:   stage INTEGER REFERENCES Stage,
85:   etudiant INTEGER REFERENCES Etudiant,
86:   PRIMARY KEY(stage, etudiant)
87: ) ;

```