Université Paris 13 — L2
Institut Galilée — Année 2014-2015
Licence 2 - 2eme semestre

Traitement de données appliqué à la finance
TP4 - SQL avec OOBase
Manipulation de données

Dans ce TP, vous utiliserez Open Office Base afin d'effectuer des requêtes sur une base de données et y extraire des données.

`http://goo.gl/0S2Zxd`

1 Base de données étudiants

Avant de commencer, ouvrez avec OpenOffice Base le fichier *BaseUniversiteUSA.odb* disponible ici : `http://goo.gl/0S2Zxd`. Regardez les trois tables composant cette base : l'un d'elles contient-elle des clefs étrangères ?

1.1 La commande SELECT

1.1.1 Les bases de SELECT

La commande SELECT permet de sélectionner et filtrer des données dans votre base. Cette commande ce ompose de trois parties : la partie SELECT où l'on explique les éléments que l'on souhaite afficher, la partie FROM où l'on explique où aller chercher les données (dans quelle table) et la partie WHERE où l'on peut spécifier des conditions de filtrage des données.

Pour afficher tous les noms d'étudiants possédant un score supérieur à 3.6, il vous faut tout d'abord aller dans le menu "requête" du programme OO Base. Ensuite, cliquez sur "Créer une requête en mode SQL", puis insérez la commande suivante :

```
SELECT "Nom" FROM "Etudiant" WHERE "Score" > 3.6
```

Sauvegardez la requête, fermez la fenêtre d'édition de code, puis double cliquez, dans la liste des requêtes, sur la requête que vous venez de créer. Une nouvelle fenêtre s'affiche avec les noms des étudiants ayant un score supérieur à 3.6.

OO Base est parfois capricieux si la requête possède une erreur de syntaxe (ici, la table Etudiants avec un "s" à la fin n'existe pas), et risque de s'arrêter brutalement si vous tentez d'exécuter de la sorte une requête erronée. Il est fortement conseillé de tester la syntaxe de sa requête avec le menu *Outils* > *SQL* avant de l'exécuter dans la partie *Requête* du logiciel.

Grâce au mot clef AND, modifiez la condition afin de n'afficher que les étudiants dont le score est entre 3.1 et 3.7.

1.1.2 Trier les données

Modifiez la commande précédente afin d'afficher le nom, le score ainsi que le numéro d'identité des étudiants dont le score est compris entre 3.1 et 3.7 (il faut ajouter des éléments après SELECT).

Pour trier les résultats obtenus par valeur croissante de score, vous devez ajouter à la fin de votre commande

```
ORDER BY "Score" ASC
```

Il est possible de trier les données selon plusieurs critères : par exemple, pour trier les données selon le score croissant puis selon le nom décroissant, on ferait :

```
ORDER BY "Score" ASC, "Nom" DESC
```

1.1.3 Effectuer des jointures

Il est, la plupart du temps, nécessaire de récupérer des données dans différentes tables. Pour ce faire, on effectue ce que l'on appelle une jointure. Le but de la jointure est de mettre en correspondance deux (ou plus) champs de deux tables différentes.

On va essayer d'afficher tous les noms des étudiants, leur score, leurs choix d'options lors de leur candidature ainsi que le résutat de la candidature. Est-ce que la table *Etudiant* et *Candidature* possèdent des "données communes" ? On va effectuer la jointure sur ces données.

Pour afficher le noms des étudiants, leur score, les universités auxquelles ils ont candidaté, l'option choisie et la décision, on fera :

```
SELECT "Nom", "Score", "NomUniv", "Option", "Decision"
FROM "Etudiant", "Candidature"
WHERE "ID" = "IDEtudiant"
```

Modifiez la requête précédente afin d'ajouter la taille de chaque université.

Modifiez la requête que vous venez de faire afin de n'afficher que les candidature retenues.

1.1.4 Modifier l'affichage des colonnes

Il est possible, dans le résultat obtenu, de modifier le nom des colonnes affichées, afin de les rendre plus explicites, grâce au mot clef AS. Par exemple, si on reprend la requête précédente, on fera cela afin de modifier la colonne *NumUniv* en *Nom de l'Universite* :

```
SELECT "Nom", "Score", "NomUniv" AS "Nom de l'Universite",
"Taille", "Option", "Decision"
FROM "Etudiant", "Candidature", "Universite"
WHERE "ID" = "IDEtudiant"
AND "Universite"."NomUniv" = "Candidature"."NomUniv"
AND "Decision"='O'
```

Il est aussi possible de faire des calculs sur des éléments numériques. Par exemple, le score donné est sur 4 ; si on souhaite l'afficher comme une note sur 20 (dont, le multiplier par 5), on fera :

```
SELECT "Nom", "Score"*5, "NomUniv" AS "Nom de l'Universite",
"Taille", "Option", "Decision"
FROM "Etudiant", "Candidature", "Universite"
WHERE "ID" = "IDEtudiant"
AND "Universite"."NomUniv" = "Candidature"."NomUniv"
AND "Decision"='O'
```

1.1.5 Renommer les tableaux dans la partie FROM

Il est possible de renommer temporairement (juste le temps de la requête) les tables de données. Cela peut être très pratique dans certains cas que nous allons aborder...

Pour renommer les tableaux on fera, par exemple :

```
SELECT "Nom", "Score", "NomUniv", "Taille", "Option", "Decision"
FROM "Etudiant" "E", "Candidature" "C", "Universite" "U"
WHERE "ID" = "IDEtudiant" AND "U"."NomUniv" = "C"."NomUniv"
```

Qu'affiche cette requête ? Est-ce que, sans renommer les tableaux, nous aurions eu le même résultat ? Oui, tout à fait...

Alors, à quoi peut bien servir le fait de renommer les tableaux. Imaginons que l'on souhaite afficher le nom des étudiants qui possèdent le même score. Pour ce faire, on ferait :

```
SELECT "ID" AS "ID1", "Nom" AS "Nom1", "Score" AS "Score1",
"ID" AS "ID2", "Nom" AS "Nom2", "Score" AS "Score2"
FROM "Etudiant"
WHERE "Score" = "Score"
```

Si on lance la requête et que l'on regarde les résultats, est-ce que cela a marché ?

Malheureusement (ou heureusement ?) non : le système s'est contenté de recopier chaque ligne de nom d'étudiants, mais n'a pas vraiment cherché les étudiants ayant le même score...

En fait, pour réussir cette opération, il est nécessaire d'appeler deux fois le tableau Etudiant en changeant son nom :

```
SELECT "E1"."ID" AS "ID1", "E1"."Nom" AS "Nom1",
"E1"."Score" AS "Score1",
"E2"."ID" AS "ID2", "E2"."Nom" AS "Nom2",
"E2"."Score" AS "Score2"
FROM "Etudiant" "E1", "Etudiant" "E2"
WHERE "E1"."Score" = "E2"."Score"
```

Maintenant, la requête porte sur deux tableaux : le tableau *E1* (copie de *Etudiant*) et le tableau *E2* (une autre copie de *Etudiant*), et on effectue une jointure classique sur ces tableaux à travers le critère de score...

Modifiez cette requête afin d'éviter que, dans les résultats, chaque étudiant n'apparaisse avec lui-même (il peut être nécessaire de faire un test sur les numéros étudiants... Le symbole "différent" en SQL s'écrit <>).

Modifiez cette requête afin que chaque couple d'étudiant avec le même score n'apparaisse qu'une seul fois.

1.2 Les combinaisons de requêtes

Il est possible de combiner les résultats de plusieurs requêtes grâce à différents mots clefs.

1.2.1 UNION

Pour concaténer les résultats de deux requêtes qui **produisent le même nombre de colonnes**, on utilisera le mot clef UNION. Par exemple, pour concaténer les résultats de la requête affichant tous les noms des étudiants avec les résultats de la requête affichant tous les noms d'université, on fera :

```
SELECT "Nom" FROM "Etudiant"
UNION
SELECT "NomUniv" FROM "Universite"
```

Si vous avez un message d'erreur à l'enregistrement, cliquez en haut de la fenêtre d'édition de code sur le bouton SQL, puis réessayez.

Quelles sont les deux différences majeures entre les résultats de la requête précédente est les résultats de cette requête :

```
SELECT "Nom" FROM "Etudiant"
UNION ALL
SELECT "NomUniv" FROM "Universite"
```

Comment pouvez-vous modifier la dernière requête afin d'afficher les noms triés de façon croissante ?

Quel résultat produit cette requête, et pourquoi ?

```
SELECT "ID", "Nom" FROM "Etudiant"
UNION ALL
SELECT "NomUniv" FROM "Universite"
```

Comment la modifier afin qu'elle fonctionne ?

1.2.2 INTERSECT

Le mot clef INTERSECT permet de trouver les éléments en commun dans deux requêtes. Si on souhaite afficher le numéro d'identité des étudiants ayant choisi l'option CS et l'option EE, nous ferons :

```
SELECT "IDEtudiant" FROM "Candidature" WHERE "Option"='CS'
INTERSECT
SELECT "IDEtudiant" FROM "Candidature" WHERE "Option"='EE'
```

Modifiez la requête précédente afin d'afficher aussi le nom des étudiants.

Pouvez-vous afficher aussi l'option choisie par les étudiants ?

En renommant les tableaux, affichez le même résultat (le numéro d'identité des étudiants ayant choisi l'option CS et EE) dans utiliser le mot clef INTERSECT.

Dans votre résultat, pourquoi l'un des étudiants apparaît plusieurs fois ? Essayez de vous débarrasser des répétitions en plaçant le mot clef DISTINCT après le mot clef SELECT.

1.2.3 EXCEPT

Le mot clef EXCEPT permet de choisir tous les résultats qui sont présents dans la première requête mais absents de la second requête. Pour obtenir tous les étudiants qui ont choisi l'option CS mais pas l'option EE, on fera :

```
SELECT "IDEtudiant" FROM "Candidature" WHERE "Option"='CS'
EXCEPT
SELECT "IDEtudiant" FROM "Candidature" WHERE "Option"='EE'
```

Pouvez-vous modifier cette requête afin d'afficher aussi le nom des étudiants ?

Comment faire si on souhaitait obtenir le même résultat sans utiliser le mot clef EXCEPT ? Que pensez-vous de cette requête ?

```
SELECT "C1"."IDEtudiant"
FROM "Candidature" "C1", "Candidature" "C2"
WHERE "C1"."Option"='CS' AND "C2"."Option"='EE'
AND "C1"."IDEtudiant" <> "C2"."IDEtudiant"
```

2 Base de notations de films

Ouvrez les trois fichiers CSV inclus dans le fichier zip, et importez-les dans une nouvelle base de données. Ces fichiers constituent une base de notes de films.

Trouvez les requêtes SQL répondant à ces questions :

1. Affichez tous les films dont le réalisateur est Steven Spielberg.
2. Trouvez tous les films ayant reçus une note de 5, et affichez-les sans répétition dans l'ordre alphabétique.
3. Affichez toutes les critiques avec le nom du reviewer, le titre du film, la note et la date de la critique.
4. Trouvez tous les films ayant reçus une note.
5. Trouvez tous les films n'ayant reçu aucune note.
6. Certains reviewers ont noté le même film plusieurs fois en leur attribuant une note différente : affichez, pour chacun de ces cas, le nom du reviewer et le titre du film.
7. Les scores de films sont sur 5 : affichez les en tant que pourcentage (sur 100).
8. Affichez tous les films dont le titre conteient "the" (le mot clef LIKE pourrait vous être utile).