

Nous allons démarrer gentiment notre projet « MiniSGBD »

Les actions à accomplir aujourd'hui (ou si besoin continuer la semaine prochaine) :

0. Organisez votre projet dans un répertoire `Projet_Nom1_Nom2...`

Dans l'idéal, votre répertoire devrait contenir un sous-répertoire `Code` et un autre sous-répertoire `DB`, ainsi qu'un script pour lancer l'exécution de votre SGBD. Cela implique en revanche la gestion des séparateurs de fichiers « multi-OS ». Plus de détails au cours du TD.

1. Créez un squelette de votre application console, qui devra prendre une série de commandes de la part de l'utilisateur, jusqu'à la commande « exit ».

2. Créez une structure / classe `RelSchema` qui contiendra le schéma d'une relation donnée :

- nom
- nombre de colonnes
- types des colonnes.

Pour les types, on se restreindra aux entiers, float et string de taille fixe `T` (notés « `stringT` »)

On suppose que pour une relation à `N` colonnes, les colonnes sont nommées `C1,...,CN`

3. Essayez de gérer, dans votre « appli console » la commande :

`create RelName NbCol TypeCol[1] TypeCol[2] ... TypeCol[NbCol]`
en créant une structure `RelSchema` correspondante.

Ex : `create R 3 int float string10`

4. Trouvez une représentation convenable pour un buffer - tableau (séquence contiguë) d'octets de taille fixe. Des arrays tout simples de `char` / `unsigned char` C ou `bytes` Java conviennent sans problème, mais rien ne vous empêche d'utiliser d'autres structures si elles vous semblent plus conviviales !

5. Essayez de vous familiariser avec la lecture et l'écriture dans les fichiers binaires suivant le langage de votre choix. En particulier, comprendre comment :

- ouvrir un fichier binaire en lecture et/ou écriture – et fermer le fichier aussi
- écrire et lire un buffer comme ci-dessus à un offset donné dans un fichier
- rajouter une ou plusieurs buffers vides (=tout à 0) à la fin du fichier

En C, vous pouvez par ex. jeter un coup d'œil à `fread/fwrite/fseek` etc ; en C++ à `fstream` ; en Java à `RandomAccessFile`.

6. Créez une structure / classe `PageId` qui contiendra un entier `FileId` et un entier `Idx`.