

Introduction générale aux bases de données

Table des matières

I. Contexte	3
II. Utilisation de fichiers pour conserver des données	3
III. Exercice : Appliquez la notion	5
IV. Intérêt des bases de données	5
V. Exercice : Appliquez la notion	6
VI. Les principes des SGBD	8
VII. Exercice : Appliquez la notion	10
VIII. Différents SGBD	10
IX. Exercice : Appliquez la notion	13
X. Essentiel	14
XI. Auto-évaluation	14
A. Exercice final	14
B. Exercice : Défi	15
Solutions des exercices	16

I. Contexte

Durée : 45 min

Environnement de travail : Windows

Pré-requis : Aucun

Contexte

Les bases de données sont des outils très employés en informatique. L'objectif de ce cours est de présenter ce que sont ces outils, pourquoi est-ce qu'ils sont aussi populaires et quel est leur intérêt.

II. Utilisation de fichiers pour conserver des données

Objectifs

- Apprendre à stocker des données sans base de données
- Comprendre les avantages et les limites de cette méthode

Mise en situation

La plupart des applications ont besoin de conserver des données : un éditeur de texte, par exemple, n'aurait que peu d'intérêt s'il était impossible de conserver le texte écrit. De même, un navigateur Internet va conserver l'historique des sites visités afin de pouvoir les retrouver facilement.

On pourrait se dire que les applications les plus simples, comme une calculatrice, n'ont pas besoin de conserver des données. Pourtant, il n'est pas rare qu'elles le fassent quand même, ne serait-ce que pour retenir les préférences de l'utilisateur.

La façon la plus simple de conserver des données est de les écrire directement dans un fichier. C'est une méthode simple, mais efficace. Elle a cependant ses limites.

Méthode

Gestion des fichiers

Les fichiers sont gérés par le système d'exploitation. Toute application peut lire ou écrire dans des fichiers, à condition qu'elle soit autorisée à le faire. C'est une opération basique, supportée par la grande majorité des langages de programmation. Le code permettant de manipuler des fichiers est donc généralement très simple et rapide.

Exemple

Voici un exemple de code sous Python pour écrire dans un fichier :

```
1 with open("fichier.txt", "w") as fichier:
2     fichier.write("Quelques données")
```

La lecture n'est pas plus complexe :

```
1 with open("fichier.txt", "r") as fichier:
2     print(fichier.read())
```

Cette simplicité n'est pas spécifique à Python : en PHP, il est possible de réaliser la même chose en utilisant simplement les fonctions `file_put_contents` et `file_get_contents`.

Limites de l'utilisation des fichiers

En réalité, lire et écrire dans un fichier est simple seulement si les besoins sont simples.

Il est facile d'écrire dans un fichier vide ou de rajouter des données à la fin d'un fichier, mais si l'ordre des données a une importance, il est beaucoup plus complexe d'écrire au milieu d'un fichier sans altérer les données déjà présentes. De même, s'il est simple de récupérer l'intégralité d'un fichier, il est plus difficile de récupérer une information spécifique, en particulier si le fichier possède plusieurs millions de lignes.

Dans tous les cas, cela reste possible, mais nécessite beaucoup plus de code, pour des performances souvent mauvaises.

Autres considérations

L'accès aux données est également un point critique de l'utilisation de fichiers. En effet, il faut gérer les accès concurrents, c'est-à-dire les cas où plusieurs personnes veulent manipuler le même fichier en même temps. Une solution simple est de mettre en place une file d'attente : mais pour une application utilisée par plusieurs milliers d'utilisateurs, il faut une mettre en place des solutions plus intelligentes et performantes.

Le contrôle des données écrites est aussi un problème qu'il faut prendre en compte. Si un fichier contient des données utilisateurs, comme sa date de naissance, il faut s'assurer que cette information soit bien au format voulu, comme JJ/MM/AAAA par exemple. Pour stocker une quantité de produits en stock, il faut que cette valeur soit numérique. Or, les fichiers n'ont aucun contrôle natif des données qu'ils contiennent : n'importe qui pourrait y insérer le contenu qu'il souhaite.

Remarque

Il ne s'agit pas de dire que les fichiers ne doivent jamais être utilisés. Le stockage par fichier a de grandes qualités, parmi lesquelles la praticité d'utilisation et la simplicité d'échange. Pour envoyer des données à quelqu'un, il suffit de les écrire dans un fichier, puis d'envoyer ce fichier au destinataire. C'est une solution simple et efficace qui reste malgré tout utilisée dans de nombreux systèmes informatiques.

De plus, il existe de nombreux formats de fichiers permettant de structurer les données de manière plus efficace que du texte simple : CSV, XML, JSON... Ces formats ont des qualités qui leur sont propres et permettent une plus grande variété d'usages. Par exemple, il est possible d'écrire des règles dans un fichier XSD pour contrôler les données d'un fichier XML, ce qui pallie une des faiblesses des fichiers traditionnels.

Cependant, nous allons voir que, pour certains cas, il est plus judicieux d'utiliser une solution alternative aux fichiers.

Syntaxe À retenir

- L'utilisation de fichiers pour conserver des données a des avantages, notamment en termes de simplicité : il est parfaitement légitime d'utiliser des fichiers pour ce pour quoi ils ont été conçus.
- Cependant, lorsque le volume de données devient important, qu'une application est destinée à de nombreux utilisateurs, ou que l'on souhaite effectuer des opérations complexes sur les données, les fichiers ne conviennent plus. Il faut alors passer à d'autres outils plus adaptés.

Complément

Fichiers informatiques¹

¹ https://fr.wikipedia.org/wiki/Fichier_informatique

III. Exercice : Appliquez la notion

Question

[solution n°1 p.17]

Tout système d'exploitation dispose de commandes permettant de lister les fichiers d'un dossier. Dans cet exercice, nous allons simplement sauvegarder cette liste dans un fichier en utilisant les outils du système d'exploitation : ce sont ces mêmes outils que les langages de programmation utilisent en interne.

Sous Windows, le terminal se nomme `cmd.exe`. Pour y accéder, il suffit d'ouvrir le menu **Démarrer** et saisir **cmd** ou **Invite de commande** dans la barre de recherche. La commande suivante permet de créer un fichier `liste.txt` dans votre dossier utilisateur et d'y mettre la liste des fichiers présents dans ce même dossier :

```
1 dir > liste.txt
```

Sous Linux et Mac, il faut rechercher **Terminal** dans le lanceur d'applications et saisir la commande suivante :

```
1 ls >> liste.txt
```

Exécuter la commande correspondant à votre système d'exploitation et vérifiez le contenu du fichier créé à l'aide d'un éditeur de texte.

IV. Intérêt des bases de données

Objectif

- Découvrir les bases de données et comprendre leur intérêt

Mise en situation

Les bases de données sont des systèmes dédiés au stockage et à la manipulation de données. Contrairement aux fichiers, les bases de données apportent également des outils facilitant la gestion des données. On parlera de « Système de Gestion de Bases de Données ».

Définition SGBD

Un système de gestion de bases de données (ou **SGBD**) est un logiciel servant à stocker, manipuler et gérer des données sous la forme de bases de données.

En plus de conserver les données, un SGBD offre donc de nombreuses fonctions permettant, entre autres, de rechercher des données, de les mettre à jour ou de les trier.

Pour communiquer avec un SGBD, il faut utiliser des requêtes SQL. Les requêtes SQL sont des instructions utilisant des mots-clés précis pour indiquer au SGBD ce que l'on attend de lui.

Remarque

Un SGBD conserve ses données dans des fichiers. Cependant, il ne s'agit pas de simples fichiers textes comme vu précédemment : l'intérêt d'un SGBD est d'apporter un grand nombre de fonctionnalités en plus de la gestion de fichiers. Ce sont ces fonctionnalités qui permettront de compenser les limites du simple stockage par fichier.

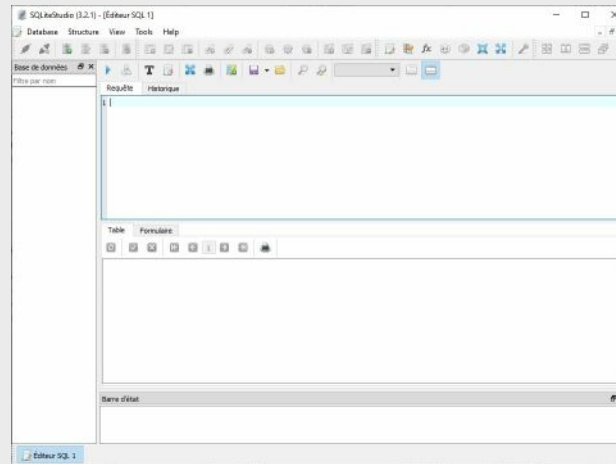
Le SGBD est donc un intermédiaire entre un programme et le stockage physique d'une machine.

Exemple SQLite

Il existe de nombreux SGBD, chacun ayant leurs spécificités. SQLite est un des SGBD les plus simples : il s'agit d'une base de données embarquée, c'est-à-dire d'un SGBD très petit et très léger, pouvant facilement être inclus dans n'importe quel autre programme.

Un bon moyen d'essayer cette base de données est d'utiliser la version portable du logiciel SQLiteStudio, qui incorpore SQLite et offre une interface graphique. Le logiciel est disponible librement en téléchargement sur le GitHub du projet¹.

Il suffit de télécharger la version portable, de décompresser l'archive et de lancer le programme.



Syntaxe À retenir

- Un système de gestion de bases de données (SGBD) permet de conserver des données sous forme de fichiers.
- L'intérêt d'un SGBD est d'apporter de nombreuses fonctions permettant de rechercher, manipuler et extraire les données.

Complément

SQLite²

V. Exercice : Appliquez la notion

Question 1

[solution n°2 p.17]

Dans cet exercice, il va falloir comparer deux manières de stocker les noms, prénoms et dates de naissance d'employés : sous forme de fichiers CSV (un format utilisé par les tableurs) et une base de données SQLite.

Dans un premier temps, ouvrez un éditeur de texte et saisissez-y les informations suivantes :

- 1 Dupond, Jean, 1970-02-24
- 2 Dupont, Jeanne, 1972-10-02
- 3 Dupré, Joseph, 1960-06-14

Sauvegardez ce fichier sous le nom `employees.csv`. Le type de fichier CSV, pour *Comma-Separated Values*, signifie que les valeurs sont séparées par des virgules.

Quel est le poids de ce fichier ?

¹ <https://github.com/pawelsalawa/sqlitestudio/releases>

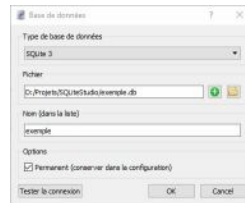
² <https://fr.wikipedia.org/wiki/SQLite>

Question 2

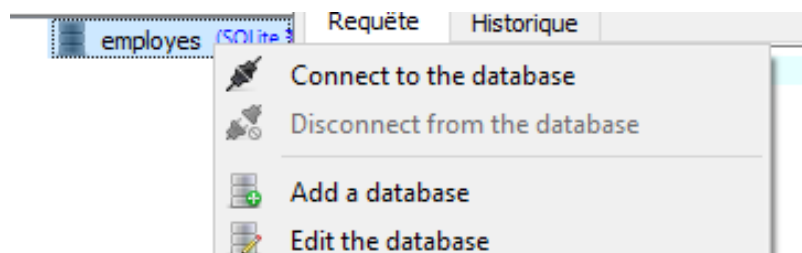
[solution n°3 p.17]

Créons maintenant une base de données avec SQLite. Pour cela, téléchargez SQLiteStudio sur le GitHub du projet¹ et lancez-le.

La première étape est de créer une base de données : allez dans Database > Add a Database, puis cliquez sur le + pour créer un nouveau fichier de base de données.



Appelez-la `exemple.db` et cliquez sur **OK**. La base de données apparaît dans le menu de gauche : faites un clic droit dessus, puis **Connect to the database** (le double clic fonctionne aussi).

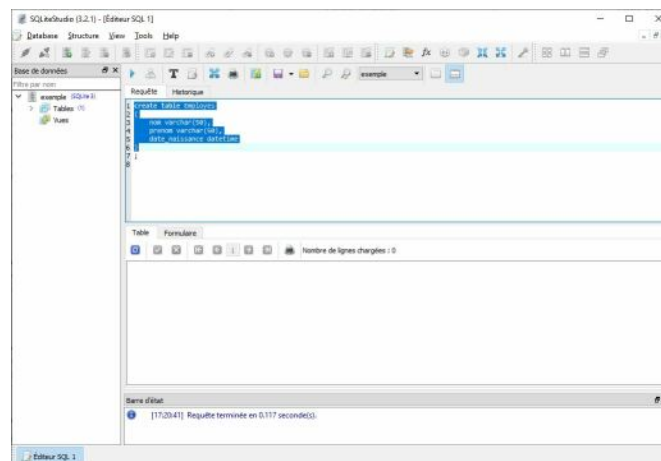


Enfin, allez dans **Tools > Open SQL Editor** et lancez les requêtes SQL suivantes, une par une, en les mettant dans le champ texte puis en cliquant sur la flèche bleue. Ces requêtes vont créer la structure de données, puis insérer les informations des employés.

```

1 create table Employes
2 (
3     nom varchar(50),
4     prenom varchar(50),
5     date_naissance datetime
6 )
7 ;

1 insert into Employes values ('Dupond', 'Jean', '1970-02-24');
1 insert into Employes values ('Dupont', 'Jeanne', '1972-10-02');
1 insert into Employes values ('Dupré', 'Joseph', '1960-06-14');
```



¹ <https://github.com/pawelsalawa/sqlitestudio/releases>

Une fois que ceci est fait, comparez la taille du fichier `employees.csv` avec celui de la base de données `exemple.db`.

VI. Les principes des SGBD

Objectif

- Découvrir les systèmes de gestion de bases de données

Mise en situation

Afin de pouvoir profiter pleinement des fonctionnalités d'un SGBD, les données qu'il contient doivent être structurées. Pour cela, il va falloir lui décrire précisément les différentes informations qu'il va devoir manipuler. Ainsi, un SGBD ne manipule pas seulement des données, mais aussi leur format.

Structuration des données

L'une des principales différences offertes par un SGBD par rapport à un simple fichier est la structuration des données. Cette structuration se fait en plusieurs niveaux.

Tout d'abord, un SGBD peut gérer plusieurs bases de données. Chaque base de données va ensuite être subdivisée en plusieurs entités, qui peuvent changer selon le type de SGBD. Le type le plus courant est le type relationnel, dans lequel les données sont réparties dans des tables. Une table est constituée de colonnes et de lignes : c'est une sorte de tableau.

Remarque

Bien que le fait de devoir structurer ses données puisse apparaître rigide et complexe, cela apporte d'importants bénéfices à l'utilisation. Cependant, il est important de se rappeler qu'un SGBD n'est qu'un outil parmi d'autres. Si les inconvénients dépassent les avantages, c'est qu'il ne s'agit pas du bon outil pour le problème à résoudre.

Exemple

Un fichier commercial peut être constitué d'informations de clients (comme leur nom, leur prénom et leur numéro de téléphone), ainsi que d'informations du commercial de chaque client (avec son nom, son prénom, son numéro de téléphone et le magasin auquel il est rattaché).

Un tel fichier peut avoir l'apparence suivante :

Client			Commercial			
Nom	Prénom	Téléphone	Nom	Prénom	Téléphone	Magasin
Dupond	Jean	0011223344	Martin	Alain	0011443322	Paris
Dupont	Jeanne	0011223355	Martin	Alain	0011443322	Paris
Dupré	Joseph	0011223366	Martin	Alain	0011443322	Paris

Voici une représentation en CSV de ces données :

```
1
2 Nom_Client,Prénom_Client,Téléphone_Client,Nom_Commercial,Prénom_Commercial,Téléphone_Commercial,Magasin_Commercial
3 Dupond,Jean,0011223344,Martin,Alain,0011443322,Paris
4 Dupont,Jeanne,0011223355,Martin,Alain,0011443322,Paris
5 Dupré,Joseph,0011223366,Martin,Alain,0011443322,Paris
```

On peut remarquer que ces trois clients ont chacun le même commercial : ses informations apparaissent donc en triple. Il serait plus pratique de séparer les informations des clients de celles des commerciaux dans deux tableaux séparés.

Client			Commercial			
Nom	Prénom	Téléphone	Code	Nom	Prénom	Téléphone
Dupond	Jean	0011223344	AMARTIN	Martin	Alain	0011443322
Dupont	Jeanne	0011223355	AMARTIN			
Dupré	Joseph	0011223366	AMARTIN			

On peut noter que l'on a rajouté une colonne *Code* au commercial afin de pouvoir l'identifier pour chaque client sans avoir à répéter toutes les autres informations.

Dans un SGBD, ce modèle serait traduit en deux tables : une table appelée `clients` et une table `commerciaux`. Cet arrangement présente de nombreux avantages :

- Les données identiques ne sont plus démultipliées
- Il est facile de modifier l'information d'un commercial (par exemple, son numéro de téléphone), car elle n'apparaît qu'à un endroit
- Il est facile de retrouver tous les clients associés à un commercial : il suffit de les trier par code
- Il est facile d'affecter un autre commercial à un client : il suffit de changer le code
- Il est possible de forcer la vérification du code commercial, pour être certain que le code associé existe bel et bien

Définition Manipulation de données

Un SGBD possède les fonctions de base suivantes :

- **Insertion** de données : permet d'ajouter une ou plusieurs informations à la base,
- **Lecture** de données : permet d'extraire des informations contenues dans la base,
- **Mise à jour** de données : permet de modifier des données dans la base,
- **Suppression** de données : permet d'effacer des données dans la base.

Le terme anglais correspondant est **CRUD** pour *Create, Read, Update, Delete*.

Remarque

Toutes ces fonctions ne sont pas exclusives aux bases de données : elles sont également réalisables avec les fichiers traditionnels. L'intérêt du SGBD est de fournir ces fonctions prêtes à l'emploi, avec une grande souplesse d'utilisation.

Syntaxe À retenir

- Un système de gestion de bases de données (SGBD) offre une structure permettant de conserver des données. Il propose également les opérations de base suivantes : écriture, lecture, modification, suppression.

Complément

Définition du CRUD¹

¹ <https://fr.wikipedia.org/wiki/CRUD>

VII. Exercice : Appliquez la notion

Question 1

[solution n°4 p.17]

Cet exercice reprend la base SQLite utilisée dans l'exercice précédent. Voici les instructions à utiliser pour la recréer si nécessaire : crée une base de données en cliquant sur **Add a database**, connectez-vous et lancez les requêtes suivantes, une par une :

```
1 create table Employes
2 (
3     nom varchar(50),
4     prenom varchar(50),
5     date_naissance datetime
6 )
7 ;

1 insert into Employes values ('Dupond', 'Jean', '1970-02-24');
1 insert into Employes values ('Dupont', 'Jeanne', '1972-10-02');
1 insert into Employes values ('Dupré', 'Joseph', '1960-06-14');
```

Les requêtes commençant par `INSERT` permettent d'insérer des données dans la base de données. C'est la première étape du CRUD : la Création de données. Le but de cet exercice va être d'utiliser les autres fonctionnalités d'un SGBD.

L'instruction permettant de lire des données (étape Read du CRUD) est `SELECT`. Lancez la requête suivante dans SQLiteStudio :

```
1 select * from Employes;
```

Quel est le résultat ?

Question 2

[solution n°5 p.18]

La troisième étape du CRUD est la mise à jour des données : Update. L'instruction permettant de réaliser cette opération porte le même nom. Voici ensuite une requête permettant de modifier le prénom d'un des employés :

```
1 update Employes set prenom = 'Gérard' where prenom = 'Joseph';
```

Lancez cette requête, puis relancez la même requête `SELECT` que précédemment. Qu'est-ce qui a changé ?

Question 3

[solution n°6 p.18]

La dernière étape est la suppression de données, c'est-à-dire la partie Delete du CRUD. Comme pour l'Update, l'instruction SQL permettant de le faire porte le même nom. Voici ensuite une requête permettant de supprimer un des employés :

```
1 delete from Employes where nom = 'Dupond';
```

Lancez cette requête, puis relancez le `SELECT`. Qu'est-ce qui a changé ?

VIII. Différents SGBD

Objectif

- Voir les différents types de SGBD

Mise en situation

Il existe plusieurs types de SGBD, chacun ayant leur propre logique pour structurer les données qu'ils doivent gérer. Jusqu'à présent, nous avons vu l'exemple de structure la plus utilisée au travers des SGBD relationnels, mais cette solution n'est pas toujours adaptée. Dans ce cours, nous allons revenir sur ce type de SGBD et le comparer aux autres méthodes existantes.

Définition SGBD relationnel

Dans un SGBD relationnel, les données sont structurées sous la forme de tables. Chaque table contient des colonnes qui décrivent le type de données, et des lignes qui forment les enregistrements.

Le croisement d'une colonne et d'une ligne représente une valeur.

Par exemple, la table `Client` ci-dessous contient les colonnes *Identifiant*, *Nom*, *Prénom* et *Téléphone* et les lignes correspondant à Jean Dupont, Jeanne Dupont et Joseph Dupré :

Client			
Identifiant	Nom	Prénom	Téléphone
1 Dupont	Jean		0011223344
2 Dupont	Jeanne		0011223355
3 Dupré	Joseph		0011223366

Remarque

Les SGBD relationnels sont parmi les plus utilisés. Ils disposent d'une structuration forte et offrent de nombreuses garanties en termes de contrôle et de fiabilité des données. Ils sont adaptés à la plupart des usages.

Exemple

Voici quelques SGBD relationnels parmi les plus courants :

- Open source : MySQL, MariaDB, PostgreSQL, SQLite
- Propriétaires : Microsoft SQL Server, Oracle BD, IBM DB2

Définition SGBD orienté colonne

Dans un SGBD orienté colonne, les données sont conservées par colonne grâce à un système **clé-valeur**, plutôt que par ligne, comme c'est le cas avec un SGBD relationnel. Dans l'exemple de la table des clients ci-dessus, un SGBD orienté colonne va ainsi regrouper tous les noms à part, puis tous les prénoms, puis tous les numéros de téléphone, plutôt que de regrouper les données par client.

Identifiant	Nom	Identifiant	Prénom	Identifiant	Téléphone
1 Dupont		1 Jean		1	0011223344
2 Dupont		2 Jeanne		2	0011223355
3 Dupré		3 Joseph		3	0011223366

Cette structure est plus limitée qu'un SGBD relationnel : il est notamment impossible de faire des liens entre plusieurs tables. De plus, de par sa structure basée sur les colonnes, un SGBD orienté colonne est beaucoup moins performant pour rassembler toutes les données d'un seul client. En revanche, il a l'avantage de fournir des performances très élevées pour manipuler de très gros volumes de données et ne récupérer que certaines informations précises.

Exemple

Voici quelques exemples de SGBD orientés colonne :

- BigTable, qui est le SGBD utilisé par Google
- HBase, qui est notamment utilisé par Facebook
- CassandraDB, qui est utilisé par Netflix et Discord

Certains SGBD relationnels proposent également un moteur orienté colonne :

- La version `MariaDB AX` de MariaDB
- L'extension `cstore_fdw` de PostgreSQL

Définition SGBD document

Dans un SGBD orienté document, les données sont réparties dans des documents indépendants les uns des autres. Contrairement à un SGBD relationnel où chaque donnée d'une table doit respecter la structure de la table, les documents peuvent être d'une structure différente les uns des autres. Ainsi, certaines valeurs peuvent être présentes dans un document et absentes dans un autre.

Le principal avantage d'un SGBD orienté document est la flexibilité qu'il offre.

Les SGBD orientés document présentent souvent des performances élevées pour la lecture de documents, mais n'offrent que peu de possibilités de croiser des données.

Client		
1	Nom	Dupond
	Prénoms	Jean
	Téléphone	0011223344
2	Nom	Dupont
	Prénoms	Jeanne
	Téléphone	0011223355
	Code postal	99 000
3	Nom	Dupré
	Prénoms	Joseph
	Fonction	Manager
	Service	Logistique

Exemple

Voici quelques SGBD orientés document :

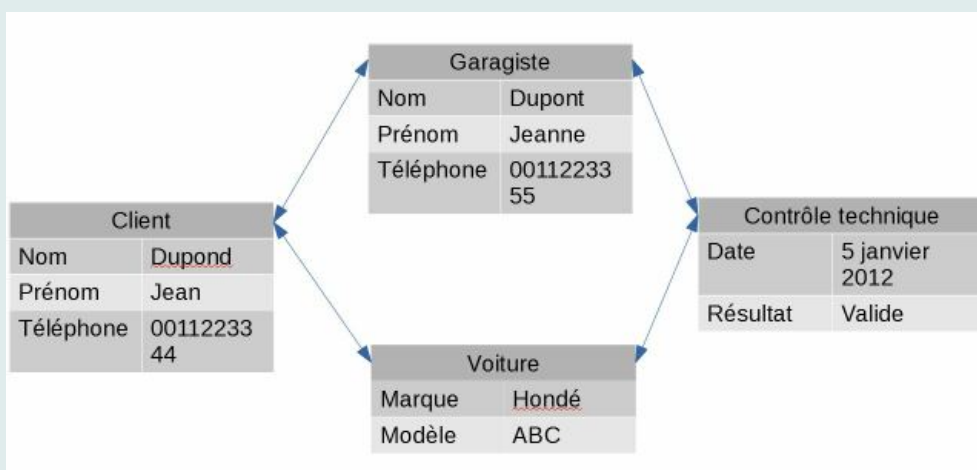
- MongoDB
- CouchDB
- Redis

Définition SGBD graphe

Dans un SGBD orienté graphe, les données sont réparties selon les relations qu'elles ont entre elles.

Les données sont appelées des **nœuds** et les relations sont appelées des **arcs**. Les arcs relient donc des nœuds entre eux.

Les bases de données graphes sont particulièrement adaptées aux tâches nécessitant d'analyser les relations entre données.



Exemple

Voici quelques SGBD orientés graphe :

- AllegroGraph
- FlockDB
- Neo4J

Syntaxe **À retenir**

- Il existe de nombreux SGBD différents, chacun ayant ses avantages et inconvénients en termes de flexibilité, de puissance, de performances...
- Il est nécessaire de choisir un SGBD adapté à l'usage requis.

Complément

Bases de données NoSQL¹

IX. Exercice : Appliquez la notion

Question

[solution n°7 p.18]

Un chef d'entreprise gérant un site de vente à distance a besoin d'un logiciel permettant à ses clients de laisser des avis sur les produits qu'ils ont achetés.

Chaque avis comporte trois données :

- L'identifiant de la personne (masqué)
- Le score donné (de 1 à 5)
- Un espace de texte permettant au client de rédiger son avis

Toutes ces informations sont obligatoires, y compris l'avis rédigé. Le responsable ne veut pas qu'une note soit donnée sans commentaire explicitant la note en question.

L'entreprise est une petite entreprise régionale vendant à des clients du secteur.

Plusieurs SGBD ont été vus dans ce chapitre. Pour cet exercice, seuls les trois types de SGBD suivants seront retenus :

- SGBD relationnel
- SGBD colonne
- SGBD document

Ces trois types peuvent faire l'affaire. Cependant, lequel paraît le plus adapté, et pourquoi ?

Indice :

Le premier élément permettant d'éliminer un candidat est le fait que toutes les informations soient obligatoires : les données sont fortement structurées, donc il faut privilégier la facilité de manipulation à la flexibilité.

Indice :

Le deuxième élément important est la taille du projet : c'est une petite entreprise régionale, qui ne devra donc pas manipuler énormément de données.

¹ <https://fr.wikipedia.org/wiki/NoSQL>

X. Essentiel

XI. Auto-évaluation

A. Exercice final

Exercice 1

[solution n°8 p.18]

Exercice

Quel est l'avantage des fichiers pour conserver des données ?

- ☐ Une grande simplicité d'utilisation
- ☐ De nombreuses possibilités de manipulation des données
- ☐ La facilité de contrôler les lectures et écritures dans le fichier

Exercice

Qu'est-ce qu'un SGBD ?

- ☐ Une base de données
- ☐ Un ensemble de logiciels permettant de conserver et de manipuler des données
- ☐ Un outil permettant de trier des données

Exercice

Où est-ce qu'un SGBD conserve ses données ?

- ☐ En mémoire
- ☐ Sous forme de fichiers sur un stockage physique
- ☐ Dans le Cloud

Exercice

Les opérations de base d'un SGBD sont...

- ☐ Le filtrage et le tri des données
- ☐ La conservation des données et leur vérification
- ☐ L'écriture, la lecture, la mise à jour et la suppression

Exercice

Comment est-ce qu'un SGBD relationnel conserve ses données ?

- ☐ Sous forme de valeurs en colonnes
- ☐ Sous forme de relations entre données
- ☐ Sous forme de tables formées de lignes et de colonnes

Exercice

Quel est le principal avantage d'un SGBD orienté colonne ?

- ☐ Sa facilité d'utilisation
- ☐ Sa performance pour manipuler de gros volumes de données
- ☐ Sa facilité à retrouver les données associées à d'autres données

Exercice

Dans un SGBD document...

- ☐ La structure de chaque document est libre
- ☐ La structure de chaque document doit être définie à l'avance et peut être contrôlée
- ☐ La structure de chaque document est définie par des lignes et des colonnes

Exercice

Dans un SGBD graphe, que représente un arc ?

- ☐ Une relation entre deux données
- ☐ Une donnée
- ☐ Un groupe de données

Exercice

Les différents SGBD sont...

- ☐ Substituables : on peut facilement passer de l'un à l'autre
- ☐ Spécialisés : chacun a ses avantages et ses inconvénients
- ☐ Classés du meilleur au moins bon : il est préférable de prendre le meilleur

Exercice

Il est préférable d'utiliser un SGBD plutôt qu'un simple stockage par fichier.

- ☐ Vrai, car un SGBD est un stockage de fichiers en mieux
- ☐ Vrai, car un SGBD apporte de nombreuses fonctions
- ☐ Faux, tout dépend du besoin : parfois un stockage par fichier sera plus adapté qu'un SGBD

B. Exercice : Défi

Une enseigne commerciale disposant de nombreux points de vente en France a besoin d'un site Internet pour développer sa vente en ligne.

Le responsable de l'enseigne présente les besoins suivants :

1. Le site doit pouvoir gérer une grande quantité de visiteurs, en particulier lors des événements comme Noël ou la rentrée des classes.
2. Le site doit pouvoir gérer tout l'aspect « facturation ». Les factures doivent répondre à des critères légaux stricts : leur format est donc déterminé par l'équipe comptable du groupe.
3. Le site doit disposer d'un espace libre pour les clients, leur permettant de laisser des avis, commentaires, conseils... Cette partie du site doit être facile d'emploi et peu restrictive.
4. Le responsable souhaite que les gestionnaires puissent analyser les différents schémas de vente. Par exemple, ils doivent savoir à tout moment quels articles sont fréquemment achetés ensemble.

Le but de cet exercice va être d'essayer de déterminer quel type de SGBD est le plus adapté à cette situation.

Question 1

[solution n°9 p.21]

Chacun des besoins soulevés par le responsable de l'enseigne correspond à un des quatre grands types de SGBD qui ont été vus dans le cours. Faites le lien entre chaque point et le type correspondant.

Question 2

[solution n°10 p.21]

Bien que les besoins exprimés correspondent à des types de SGBD différents, il est rarement pratique de faire fonctionner plusieurs SGBD en même temps pour un seul site : il faut faire un choix. Bien entendu, il ne sera pas forcément le meilleur choix pour un besoin précis, mais il doit pouvoir couvrir tous les cas.

D'après vous, quel type de SGBD choisir pour ce projet ?

Indice :

Il est important de définir quel est le besoin le plus important du projet pour définir le SGBD à utiliser. Ensuite, il sera généralement possible de trouver des solutions dans le code pour pallier ses faiblesses dans les autres aspects.

Solutions des exercices

p. 5 Solution n°1

Le fichier `liste.txt`, situé dans votre dossier utilisateur, doit contenir la liste des fichiers présents dans ce même dossier.

Les commandes `dir` et `ls` permettent, respectivement sous Window et Linux ou Mac, de lister les répertoires et fichiers présents dans le répertoire en cours. L'utilisation de l'opérateur `>` sous Windows et `>>` sous Linux ou Mac permet de réaliser une redirection du flux de sortie de la commande. Cela signifie que le retour de la commande `dir` ou `ls` est redirigé vers le fichier `liste.txt`.

La manipulation des fichiers est quelque chose de natif et de très simple pour un système d'exploitation : une simple commande permet de conserver des données dans un fichier. Notons toutefois que, dans le cas présent, quelques difficultés se présenteront dans l'utilisation de ces données, notamment en raison du formatage et de la présence d'entêtes.

p. 6 Solution n°2

Le fichier CSV est très léger : 1 Ko.

p. 7 Solution n°3

Voici les deux fichiers générés :

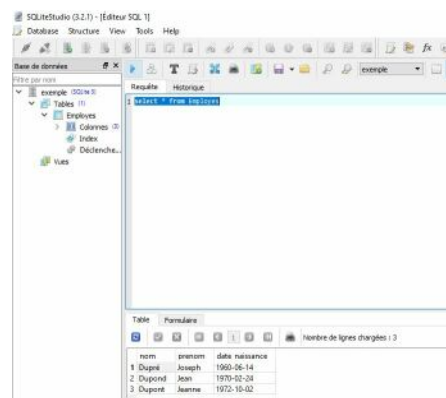


On peut constater que le fichier `.csv` est beaucoup plus petit. De plus, en essayant d'ouvrir le fichier `.db` avec un éditeur, on constate la présence de nombreux caractères non alphabétiques.

La base de données ne contient pas seulement les données, mais beaucoup d'autres informations utiles au SGBD, qui ne peuvent pas être affichées telles quelles.

p. 10 Solution n°4

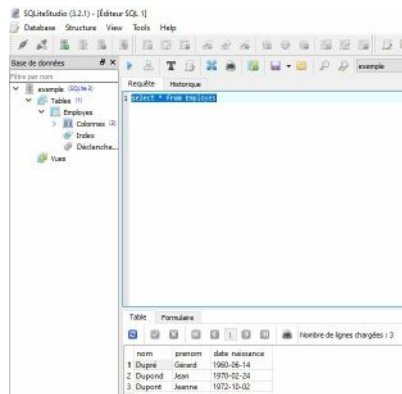
Le `SELECT` retourne les trois employés qui ont été insérés : Jean DUPOND, Jeanne DUPONT et Joseph DUPRÉ :



L'ordre des lignes peut être différent que celui indiqué, mais l'important est de retrouver les informations des employés.

p. 10 Solution n°5

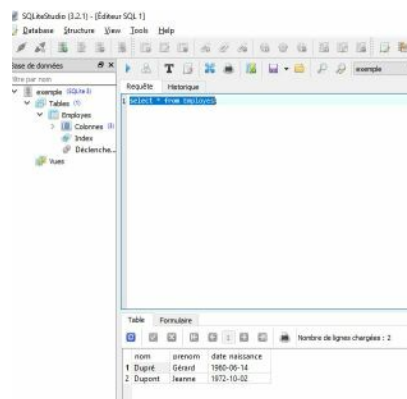
Voici maintenant le résultat du `SELECT` après l'Update :



Le prénom de Joseph DUPRÉ a été modifié : il s'appelle maintenant **Gérard** DUPRÉ.

p. 10 Solution n°6

Voici le résultat du `SELECT` après le Delete :



Il n'y a plus que deux lignes au lieu de trois : Jean Dupond a été supprimé. À noter que la ligne supprimée l'est définitivement : il n'y a pas d'option de restauration.

p. 13 Solution n°7

L'analyse du problème permet de mettre en évidence les points suivants :


- Les données sont fortement structurées puisque tous les champs sont obligatoires : nous n'avons donc pas besoin de la flexibilité offerte par un SGBD orienté document.
- Le volume de données restera probablement faible, puisqu'il s'agit d'une petite entreprise régionale : il ne sera donc pas possible de profiter des avantages d'un SGBD orienté colonne.

Un SGBD relationnel est probablement le plus adapté.

Exercice p. 14 Solution n°8


Exercice

Quel est l'avantage des fichiers pour conserver des données ?

- ☒ Une grande simplicité d'utilisation
- ☐ De nombreuses possibilités de manipulation des données
- ☐ La facilité de contrôler les lectures et écritures dans le fichier
-  Les fichiers sont simples à utiliser, mais n'offrent pas la puissance et la sécurité que proposent des outils plus complexes.


Exercice

Qu'est-ce qu'un SGBD ?

- ☐ Une base de données
- ☒ Un ensemble de logiciels permettant de conserver et de manipuler des données
- ☐ Un outil permettant de trier des données
-  Un SGBD permet de gérer des bases de données et offre de nombreux outils pour manipuler les données.


Exercice

Où est-ce qu'un SGBD conserve ses données ?

- ☐ En mémoire
- ☒ Sous forme de fichiers sur un stockage physique
- ☐ Dans le Cloud
-  Un SGBD utilise des fichiers pour conserver ses données. Cela ne l'empêche pas d'utiliser également la mémoire sous la forme de cache temporaire. Quand au Cloud, il s'agit avant tout de serveurs disposant eux-mêmes de stockages physiques. Une donnée conservée « dans le Cloud » est donc elle-même enregistrée dans un fichier sur le stockage d'un serveur distant.

Exercice

Les opérations de base d'un SGBD sont...

- ☐ Le filtrage et le tri des données
- ☐ La conservation des données et leur vérification
- ☒ L'écriture, la lecture, la mise à jour et la suppression
-  Ce sont les quatre fonctions regroupées sous le sigle anglais CRUD (*Create, Read, Update, Delete*).

Exercice

Comment est-ce qu'un SGBD relationnel conserve ses données ?

- ☐ Sous forme de valeurs en colonnes
- ☐ Sous forme de relations entre données
- ☒ Sous forme de tables formées de lignes et de colonnes

- Q Les données d'un SGBD relationnel sont conservées sous forme de tables. Ce sont les tables qui ont des relations entre elles, pas seulement les données.

Exercice

Quel est le principal avantage d'un SGBD orienté colonne ?

- ☐ Sa facilité d'utilisation
- ☒ Sa performance pour manipuler de gros volumes de données
- ☐ Sa facilité à retrouver les données associées à d'autres données

- Q Les SGBD orientés colonne sont principalement utilisés pour gérer les très gros volumes de données, nécessitant donc des performances importantes.

Exercice

Dans un SGBD document...

- ☒ La structure de chaque document est libre
 - ☐ La structure de chaque document doit être définie à l'avance et peut être contrôlée
 - ☐ La structure de chaque document est définie par des lignes et des colonnes
- Q Les SGBD documents ont pour principal avantage une grande flexibilité, grâce à leur structure libre.

Exercice

Dans un SGBD graphe, que représente un arc ?

- ☒ Une relation entre deux données
- ☐ Une donnée
- ☐ Un groupe de données

- Q Un arc est une relation qui relie deux nœuds (données) entre eux. Bien qu'au sens strict cet arc soit lui-même une information, donc quelque part une donnée, il s'agit avant tout d'une relation.

Exercice

Les différents SGBD sont...

- ☐ Substituables : on peut facilement passer de l'un à l'autre
 - ☒ Spécialisés : chacun a ses avantages et ses inconvénients
 - ☐ Classés du meilleur au moins bon : il est préférable de prendre le meilleur
- Q Il n'y a pas forcément de bon ou de mauvais SGBD. Chacun a ses cas d'usage privilégiés.

Exercice

Il est préférable d'utiliser un SGBD plutôt qu'un simple stockage par fichier.

- ☐ Vrai, car un SGBD est un stockage de fichiers en mieux
- ☐ Vrai, car un SGBD apporte de nombreuses fonctions
- ☒ Faux, tout dépend du besoin : parfois un stockage par fichier sera plus adapté qu'un SGBD
- ☐ Il n'y a pas de meilleure solution, il y a une solution mieux adaptée. Les fichiers sont notamment privilégiés pour échanger des données par Internet.

p. 16 Solution n°9

1. Pour gérer un grand volume de données (grande quantité de visiteurs), les SGBD colonnes sont les plus performants.
2. Pour la fiabilité et la rigueur de la structure, les SGBD relationnels sont les plus indiqués
3. Pour une partie plus libre, où l'on ne sait pas à l'avance ce que vont mettre les utilisateurs, la flexibilité des SGBD documents est un avantage.
4. Les SGBD graphes sont les plus adaptés pour analyser les relations entre les données, et donc pour pouvoir faire le lien entre les achats de deux articles.

p. 16 Solution n°10

Ce problème est très courant en informatique. Les besoins exprimés sont très différents, mais il n'est pas forcément possible d'utiliser des outils différents pour résoudre chacun de ces besoins. Le choix le plus fréquent dans ce cas est le **SGBD relationnel**. En effet :

- Le besoin de fiabilité et de structure forte pour l'aspect « facture » est primordial, car il s'agit ici d'un aspect légal. L'entreprise prendrait un gros risque à ne pas pouvoir respecter cette contrainte.
- Un SGBD relationnel n'est pas le plus rapide des SGBD, mais il reste quand même un outil performant malgré tout.
- L'aspect « facile d'emploi » de l'espace client dépend principalement de l'ergonomie du site, plus que du choix du SGBD. Ce sera peut être un peu plus complexe à programmer, mais pas plus complexe à l'utilisation.
- L'aspect « analyse » peut être complexe et plus lent à établir avec un SGBD qui n'est pas un graphe, mais reste possible à faire avec un SGBD relationnel. De plus, ces analyses sont un besoin interne de l'entreprise et n'ont aucun impact sur la vitesse d'affichage du site pour les utilisateurs : un amoindrissement des performances n'est donc pas très important.