T.P. I - Bases de données Premières requêtes

Les bases de données permettent de gérer les données numériques d'une entreprise. L'enjeu est d'organiser ces données efficacement de manière à pouvoir accéder rapidement à des informations et à pouvoir mettre à jour les données rapidement et sans erreurs.

I - Bases de données

I.1 - Introduction

Définition 1 - Base de données

Une *base de données* est un ensemble structuré d'informations qui permet de répondre à un besoin spécifique.

Nous allons prendre par la suite l'exemple d'une entreprise qui possède des antennes dans différentes villes de France et gère la formation de ses employés. On pourrait représenter les formations dans le tableau suivant :

Stagiaire	Ville	Formation	Date	Lieu	Formateur/Formatrice
M. Gagnot	Toulouse	Sécurité	04/01/23	Montpellier	Mme Frele
M. Gagnot	Toulouse	Accueil client	20/01/23	Nantes	Mme Gérard
Mme Gourot	Montpellier	Sécurité	04/01/23	Montpellier	Mme Frele
M. Gagnot	Toulouse	Communication	01/06/23	Montpellier	Mme Frele
⋮	⋮	⋮	⋮	⋮	⋮

On remarque que :

- la même information est présente sur différentes lignes, ce qui rend complexe les mises à jour (par exemple le changement de date d'une formation),
- beaucoup d'informations sont ici absentes et on aurait besoin de beaucoup plus de colonnes (par exemple le CV du formateur, les contacts téléphoniques,...).

On va donc utiliser un outil qui permet de manipuler les données plus aisément :

- le Système de Gestion de la Base de Données (SGBD) va représenter les données de manière structurées et facilement modifiables (nous ne nous préoccuperons pas de savoir comment ce système fonctionne),
- le langage Structured Query Language (SQL) va nous permettre d'interroger la base de données et de mettre à jour les données aisément (c'est ce que nous devons apprendre !)

Définition 2 - Requête

Une *requête* est une interrogation posée à la base de données. Les requêtes seront rédigées en langage SQL.

Exemple 1 - Exemple de requête

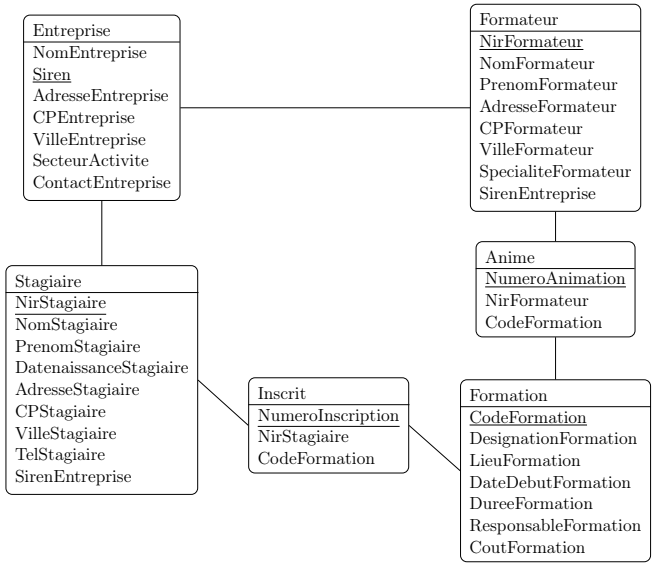
Quels employés sont en formation le 04/01/2023 ?

I.2 - Représentation

La structure des bases de données est souvent représentée à l'aide de rectangles reliés par des traits.
Dans notre exemple, on utilise différentes tables pour stocker les données de l'entreprise. Intuitivement, chaque table fonctionne comme un tableur où les colonnes sont appelées les attributs :

- une table `Entreprise` qui stocke les informations des entreprises qui bénéficient des formations,
- une table `Stagiaire` qui stocke les informations des personnels qui bénéficient des formations,
- une table `Formateur` qui stocke les informations concernant les personnels qui animent les formations,

* une table `Formation` qui stocke les informations concernant les différentes formations disponibles,
* une table `Anime` qui enregistre quel est le formateur qui anime quelle formation,
* une table `Inscrit` qui stocke les identifiants des personnels inscrits aux différentes formations.



Définition 3 - Table, Attribut, Domaine, Enregistrements, Relation

* Une *table* (ou *relation*) est un tableau (on pourra se le représenter mentalement comme une page d'un tableur).
* Un *enregistrement* est une composante de la table.
* Un *attribut* est le nom d'une colonne d'une table.
* Le *domaine* d'un attribut est le type de données de cet attribut.
* Les tables sont reliées par des *relations*.

Exemple 2 - Vocabulaire des bases de données

* Les tables de notre bdd sont : `Entreprise`, `Stagiaire`, `Inscrit`, `Formation`, `Anime`, `Formateur`.
* Dans la table entreprise, les attributs sont : `NomEntreprise`, `Siren`, `AdresseEntreprise`, `CPEntreprise`, `VilleEntreprise`, `SecteurActivite`, `ContactEntreprise`.
* Un enregistrement de la table `Entreprise` pourra être par exemple
  (L'Ozenne, 193100476, 9 rue Merly, 31070, Toulouse, Formation, 0310047h@ac-toulouse.fr)
* Les attributs `NomEntreprise` ou `NomStagiaire` sont du texte. Leur domaine est TEXT.
* Le domaine de `DureeFormation` est INTEGER (nombre entier) car il compte un nombre de jours de formations.
* Le domaine de `CoutFormation` est FLOAT (réel) car il représente le coût de la formation en euros.

Définition 4 - Clé primaire / étrangère

Un attribut qui permet d'identifier de manière unique un enregistrement est nommé *clé primaire*.
Un attribut dans une table qui est clé primaire d'une autre table est une *clé étrangère*.

Exemple 3 - Clés primaires

* Le numéro `Siren` permet d'identifier de manière unique une entreprise. C'est la clé primaire de la table `Entreprise`.
* Le numéro `NirStagiaire` permet d'identifier de manière unique un stagiaire via son numéro de sécurité sociale. C'est la clé primaire de la table `Stagiaire`.
* L'attribut `NomStagiaire` ne peut pas être une clé primaire car des personnels différents peuvent avoir le même nom

de famille.
* Le numéro `NumeroInscription` permet d'identifier de manière unique l'inscription d'un personnel à une formation. Il s'agit d'un numéro attribué automatiquement.
* La clé `SirenEntreprise` est une clé étrangère de la table `Stagiaire`.
* Les clés `NirStagiaire` et `CodeFormation` sont des clés étrangères de la table `Inscrit`.

Exercice 1. La base de données d'une société de remontées mécaniques contient les tables suivantes :

* `Clients (Cl_num:INTEGER, Cl_nom:TEXT, Cl_rue:TEXT, Cl_ville:TEXT)`
* `Badge (Ba_num:INTEGER, Do_num:INTEGER, TP_num:INTEGER, TF_num:INTEGER, Cl_num:INTEGER, Ba_date:TEXT)`
* `Domaine_Skiable (Do_num:INTEGER, Do_nom:TEXT)`
* `Type_Forfait (TF_num:INTEGER, TF_intitule:TEXT)`
* `Type_Public (TP_num:INTEGER, TP_nom:TEXT, TP_conditions:TEXT)`
* `Saison (Sa_num:INTEGER, Sa_intitule:TEXT)`
* `Tarif_Forfait (Do_num:INTEGER, TF_num:INTEGER, TP_num:INTEGER, Sa_num:INTEGER, Mt_forfait:FLOAT)`

1. Souligner les clés primaires et entourer les clés étrangères.

2. Proposer un enregistrement pour la table `Clients` puis un enregistrement pour la table `Tarif_Forfait`.

II - Sélection d'attributs

II.1 - Requêtes simples

Définition 5 - SELECT ... FROM ...

La requête `SELECT attributs FROM table` permet d'afficher les `attributs` de `table`.

À Savoir

La casse des caractères (majuscule ou minuscule) n'a pas d'importance pour les mots-clés, les noms de tables et d'attributs. La convention veut que les mots-clés soient écrits en majuscules.

* Affiche tout le contenu de la table `Entreprise` :

```
SELECT *
FROM Entreprise
```

* Affiche tous les noms d'entreprises de la table `Entreprise` :

```
SELECT NomEntreprise
FROM Entreprise
```

* Affiche les dates de début de formation de la table `Formation` :

```
SELECT DateDebutFormation
FROM Formation
```

* Affiche les noms, prénoms et téléphones des stagiaires présents dans la table `Stagiaire` :

```
SELECT NomStagiaire, PrenomStagiaire, TelStagiaire
FROM Stagiaire
```

* Affiche le nom de l'entreprise et son numéro SIREN pour les entreprises présentes dans la table `Entreprise` :

```
SELECT NomEntreprise, Siren
FROM Entreprise
```

Exercice 2. On reprend l'exemple de la société de remontées mécaniques. Écrire les requêtes suivantes.

1. Afficher le nom des clients.

2. Afficher le nom et la ville des clients.

3. Afficher le nom des domaines skiables.

4. Afficher le nom des différents types de forfaits.

5. Afficher le montant des différents types de forfaits.

II.2 - Requêtes conditionnelles

Il est parfois utile de ne rechercher que les données qui satisfont certaines conditions.

Définition 6 - SELECT ... FROM ... WHERE ...

La requête `SELECT attributs FROM table WHERE conditions` permet d'afficher les `attributs` de `table` qui satisfont `conditions`.

Définition 7 - Comparateurs & Connecteurs

* Pour comparer les entiers, dates,... on pourra utiliser les *opérateurs de comparaison* :
  =, <> (différent), <, <=, >, >=.
* Pour agréger des conditions on pourra utiliser les *connecteurs logiques* :
  `AND`, `OR`, `NOT`.

* Nom des entreprises de la table `Entreprise` domiciliées à Toulouse :

```
SELECT NomEntreprise
FROM Entreprise
WHERE VilleEntreprise = "Toulouse"
```

* Nom des entreprises de la table `Entreprise` domiciliées en dehors de Toulouse :

```
SELECT NomEntreprise
FROM Entreprise
WHERE VilleEntreprise <> "Toulouse"
```

* Lieux et durées des formations débutant le `2023-01-24` :

```
SELECT LieuFormation, DureeFormation
FROM Formation
WHERE DateDebutFormation = 2023-01-24
```

* Nom des stagiaires nés avant le `1994-01-01` :

```
SELECT NomStagiaire
FROM Stagiaire
WHERE DatenaissanceStagiaire < 1994-01-01
```

* Nom des responsables, libellés et coûts des formations dont le coût est compris entre 500 et 1000 euros.

```
SELECT ResponsableFormation, DesignationFormation,
CoutFormation
WHERE CoutFormation >= 500
      AND CoutFormation <= 1000
```

Exercice 3.

1. En utilisant la base des formations, écrire des requêtes permettant d'afficher :

a) les spécialités des formateurs domiciliés à Toulouse.

b) le nom et le contact des entreprises exerçant dans le secteur de l'éducation.

c) les numéros de sécurité sociale des formateurs animant la formation numéro 2034.

2. En reprenant la base de donnée gérant les remontées mécaniques, écrire des requêtes permettant d'afficher :

a) les noms des clients domiciliés à Toulouse.

b) les numéros de domaines skiables correspondants à des forfaits dont le montant est compris entre 30 et 50 euros.

c) les numéros de domaines skiables correspondants à des forfaits dont le montant est compris entre 30 et 50 euros en basse saison (saison numérotée 0).