Partie 1

Base de données

Base de données

Introduction générale

1- Définition d'une base de données :

Une base de données est un ensemble de fichiers structuré et organisé permettant le stockage de grandes quantités d'informations afin d'en faciliter l'exploitation (ajout, mise à jour, recherche de données).

2- SGBD (Système de Gestion de Bases de Données)

Système de Gestion de Base de Données(SGBD)

C'est un outil qui permet de de **gérer** des nouvelles bases de données, c'est-à-dire :

1. Permettre l'accès aux données de façon simple.
2. Autoriser un accès aux informations à plusieurs utilisateurs,
3. Manipuler les données présentes dans la base de données (insertion, suppression, modification)
4. Assurer la sécurité des données

SGBD relationnel (SGBDR)

Objectifs du modèle relationnel

1. Indépendance entre les applications et la représentation interne des données.
2. Etablir une base de données solide pour traiter les problèmes de cohérence et de redondance des données.

Le SGBD qui gère une BDD relationnelle est appelé "SGBD relationnel", ce qui est souvent abrégé en SGBDR.

Dans le modèle relationnel, les données sont organisées sous forme de **tableaux de valeurs** appelés **tables.**

Table

Entité = tableau de valeurs = table

La table est une collection de données relatives à un sujet spécifique tel qu'un client ou un véhicule.

- *Les* tables organisent les données en colonnes et en lignes
- La Colonne (attribut ou champ) :
  - Catégorie d'informations précises
- La ligne (Enregistrement ou n-uplet):
  - Série d'informations concernant le même sujet
- **Schéma d'une table :** S'écrit sous la forme suivante :
  Nom-table (champ1, champ2, champ3,……)

Clée primaire

Est un champ qui permet d'identifier de façon unique chaque enregistrement de la table dont elle en fait partie et améliore les performances du serveur de bases de données.

Clé étrangère

La clé étrangère est **une clé primaire** qui a migré d'une table-mère à une table-fille.

Elle permet de **gérer des relations** entre plusieurs tables, et garantit **la cohérence des données.**

Notation : Par convention, la clé étrangère est suivi du caractère **#**

Relation

Les relations

Une relation est un lien entre deux tables d'une base de données à l'aide de deux **champs en commun** à ces deux tables. Ces deux champs sont dits **associés.**

Pour accéder simultanément à plusieurs tables, il faut définir une relation ; c.-à-d. créer un lien entre les tables pour que Microsoft Access puisse **fusionner les données de plusieurs tables**.

Relation un à un (1-1)

Signifie qu'un enregistrement de la première table ne peut correspondre qu'un seul enregistrement dans la deuxième table.

Exemple :

RG1 : Une personne possède un et un seul N° de carte d'identité

et

RG2 : Un N° de carte d'identité désigne une et une seule personne

Les types de relations

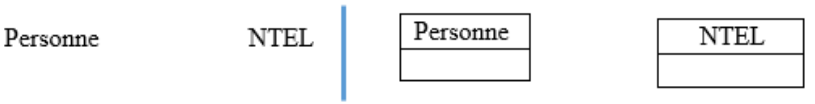
Relation un à plusieurs (1-N)

Signifie qu'un enregistrement de la première table contenant la clé primaire, peut être associé à plusieurs enregistrements dans la deuxième table.

Exemple :

RG1 : Une personne possède plusieurs N° de téléphone **et**

RG2 : Un N° de téléphone désigne une et une seule personne.



Relation plusieurs à plusieurs (M-N)

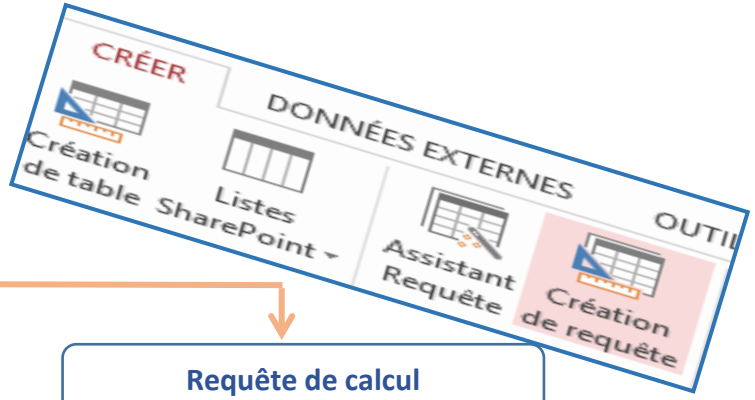
Signifie qu'un enregistrement de la première table peut être associé à plusieurs enregistrements dans la deuxième table et inversement.

Exemple :

RG1 : Un élève est enseigné par plusieurs enseignants et

RG2 : Un enseignant enseigne plusieurs élèves.

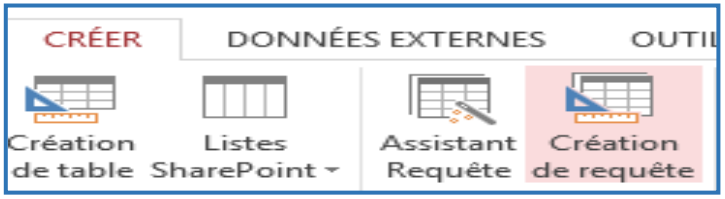
Requetes



Requête de sélection

- **Requête de sélection simple**
  - Utiliser les opérateurs relationnels
- **Requête paramétrée**
- **Requête de calcul**
  - Pour pouvoir calculer la somme, la moyenne,

Requête de mise à jour



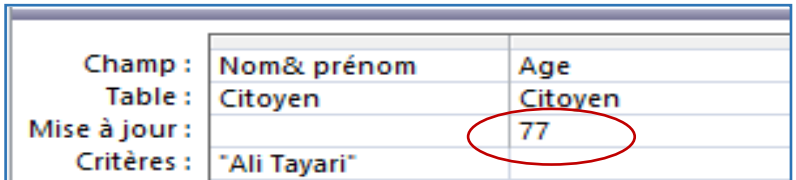
C'est une requête qui apporte des changements globaux à des enregistrements dans une ou plusieurs tables.

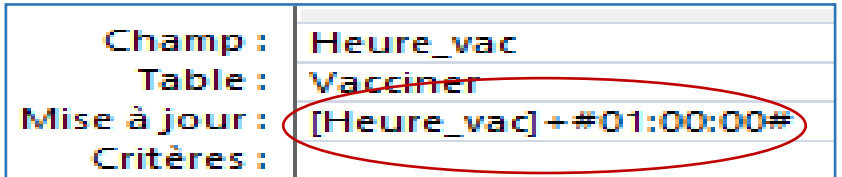
La nouvelle valeur ne dépend pas de l'ancienne valeur

Ecrire directement la nouvelle valeur dans l'emplacement adéquat.

La nouvelle valeur dépend de l'ancienne valeur

Ecrire le nom du champ entre crochets [ ] et l'utiliser dans une expression répondant à vos besoins



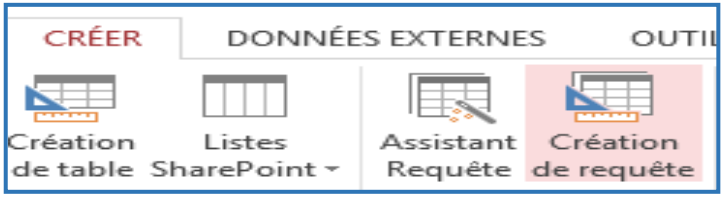


Requête d'ajout

Elle permet d'ajouter un groupe d'enregistrements d'une ou de plusieurs tables à la fin d'une ou de plusieurs tables

L'ajout des enregistrements se fait par défaut à la fin de la table.

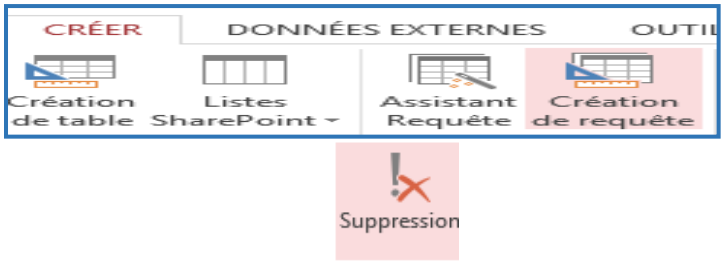
Si la table contient des contraintes d'ordre, l'ajout se fait dans la bonne position.



Champ :	N_ins	CIN	Nom& prénom	NumTEl	Maladie chronique	Age
Table :	Citoyen mineur	Citoyen mineur	Citoyen mineur	Citoyen mineur	Citoyen mineur	Citoyen mineur
Tri :						
Ajouter à :	N_ins	CIN	Nom& prénom	NumTEl	Maladie chronique	Age
Critères :						>=15
Ou :						

Requête de suppression

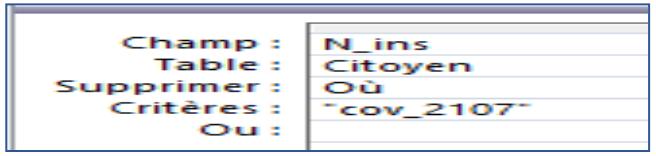
Les requêtes Suppression permettent de **supprimer rapidement les enregistrements qui répondent aux critères posés**.

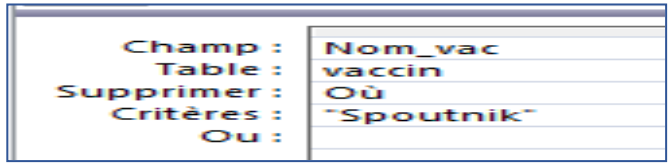


1 Créez tout d'abord une requête Sélection permettant de sélectionner les enregistrements concernés : seuls les champs sur lesquels portent les critères doivent être insérés.

2 Cliquez sur le bouton **Exécuter** pour vérifier la liste des enregistrements.

3 Cliquez sur le bouton **Suppression**



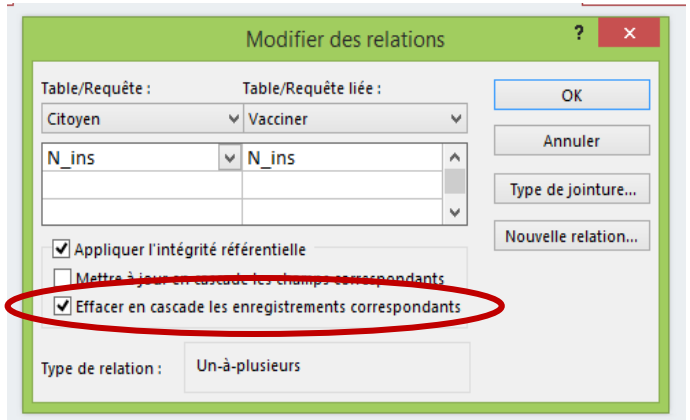


Important :

1. Utiliser des critères pour renvoyer uniquement les enregistrements que vous souhaitez supprimer.

Sinon, la requête Suppression supprime tous les enregistrements de la table.

2. Si les enregistrements résident du côté « un » et du coté « plusieurs » d'une relation un-à-plusieurs,



vous devrez peut-être modifier la relation avant d'exécuter la requête Suppression.

TP n°1 : Base de données

Soit la base de données suivante:

Client (num_clt, nom_clt, prenom_clt, adresse_clt, numtel_clt)

Fournisseur (num_fr, nom_fr, adresse_fr, numtel_fr)

Produit (code_pdt, desig_pdt, prix_pdt, qté_pdt)
Produit_livré (num_clt , num_fr, code_pdt, date, qté)

1. Lancer le logiciel MS Access puis créer une nouvelle base de données nommée **livraison** que vous devez enregistrer dans votre dossier de travail situé sous le chemin C:\bac2022.

2. Créer les tables **Client**, **Fournisseur**, **Produit** et **Produit_livré**, en tenant compte des indications ci-dessous, et les enregistrer respectivement sous les noms **Client**, **Fournisseur**, **Produit** et **Produit_livré**.

Client		
Nom de champs	**Type de champs**	**Propriétés**
num_clt	Numérique	Entier
nom_clt	Texte	Taille =25
prenom_clt	Texte	Taille = 25
adresse_clt	Texte	Taille = 50
numtel_clt	Numérique	Entier long

Fournisseur		
Nom de champs	**Type de champs**	**Propriétés**
num_fr	Numérique	Entier
nom_fr	Texte	Taille =25
adresse_fr	Texte	Taille = 50
numtel_fr	Numérique	Entier long

Produit_livré		
Nom de champs	**Type de champs**	**Propriétés**
num_clt	Numérique	Entier
num_fr	Numérique	Entier
code_pdt	Texte	Taille = 10
date	Date/heure	Date, abrégé
qté	Numérique	Entier

3. Remplir les tables par les données suivantes :

Produit		
Nom de champs	**Type de champs**	**Propriétés**
code_pdt	Texte	Taille = 10
desig_pdt	Texte	Taille =30
prix_pdt	numérique	Entier
qté_pdt	numérique	Entier

Table Client				
num_clt	**nom_clt**	**prenom_clt**	**adresse_clt**	**numtel_clt**
100	Ben Saleh	Ali	Monastir	22365987
101	Abidi	Ahmed	Sousse	98600230
102	Aloulou	Med	Tunis	24500236
103	Abassi	Intissar	Sfax	99800632
120	Rayhane	Mohamed Ali	Tunis	23200456
121	Ben Ali	Miled	Monastir	73500236
125	Jallouli	Ines	Gabes	95236870

Table Fournisseur			
num_fr	**nom_fr**	**adresse_fr**	**numtel_fr**
200	SEFINFO	Moknine	73435555
210	Planet	Sousse	73220364
220	Global Info	Tunis	71200336
230	Sté Info	Sfax	74362214
250	Info net	Jammel	93654200
270	Web TIC	Sahline	22369100

Table Produit			
code_pdt	**desig_pdt**	**prix_pdt**	**qté_pdt**
A01	Ecran Plat 22x	288	15
A02	Ecran LCD	187	10
B10	Souris optique	12	100
B11	Clavier USB	8	66

Table Produit_livré				
num_clt	**num_fr**	**code_pdt**	**date**	**qté**
100	200	A01	22/06/2006	10
102	210	D55	06/06/2007	20
103	230	B11	10/10/2007	5
125	220	F12	06/02/2008	3
125	270	M01	11/01/2008	10

C12	Carte mère Acerpro	518	5
D55	Graveur externe 32*	148	23
F12	HDD externe 500 Go	489	10
F15	HDD 160 Go	155	10
M01	Clé USB 4Go	92	100

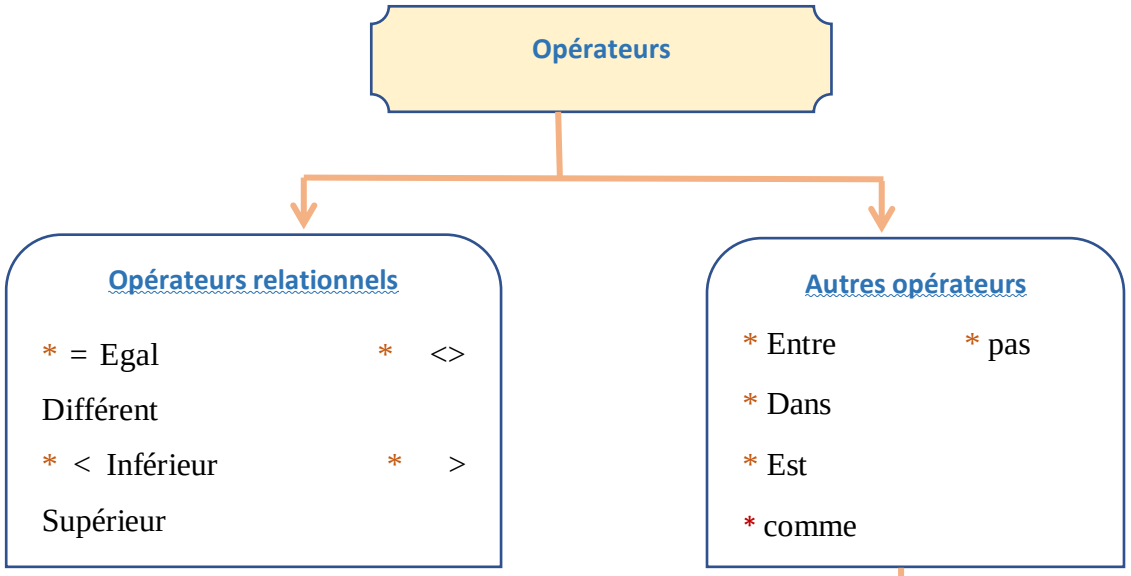
4. Identifier puis créer les relations entre ces tables.

TP n°2 : Base de données

Créer et enregistrer les requêtes suivantes :

- **R1 :** Afficher les numéros des fournisseurs et leurs noms dont la ville est Tunis.
- **R2 :** Lister les produits dont le prix < 100.
- **R3 :** Afficher le code et la désignation des produits dont le prix et compris entre 20 et 150.
- **R4 :** Afficher les fournisseurs de Sousse et Moknine.
- **R5 :** Afficher les produits en ordre décroissant selon leurs prix.
- **R6 :** Afficher les produits livrés durant l'année 2007.
- **R7 :** Afficher les fournisseurs provenant d'une ville donnée.
- **R8 :** Afficher la désignation, le prix et la quantité des produits livrés entre deux années données.
- **R9 :** Augmenter de 15% les prix des produits.
- **R10 :** Ajouter l'enregistrement suivante dans la table produit-livré

Num-clt : 102 num-fr : 250 code-pdt : F15



Entre

Sélectionne les enregistrements pour lesquels la valeur d'un champ est comprise dans un intervalle de valeurs.

Dans

Sélectionne les enregistrements pour lesquels la valeur d'un champ est comprise dans une liste.

Comme

Sélectionne les enregistrements contenant une donnée approximative.

Est

Sélectionne les enregistrements pour lesquels un champ est vide ou non

Dans **Comme "rue*"** **Est NULL** **Est pas NULL**

Caractères utilisés :

* : Remplace de 0 à n caractères

? : Remplace 1 seul caractères

Exemple : 2ème lettre B et se termine par e comme "?B*e"

Correction des requêtes

……

Champ :							
Table :							
Tri :							
Afficher :							
Critère :							
Ou :							

……

Champ :							
Table :							
Tri :							
Afficher :							
Critère :							

Ou :							

……

Champ :							
Table :							
Tri :							
Afficher :							
Critère :							
Ou :							

……

Champ :							
Table :							
Tri :							
Afficher :							
Critère :							
Ou :							

………

Champ :							
Table :							
Tri :							
Afficher :							
Critère :							
Ou :							

……

Champ :							
Table :							
Tri :							
Afficher :							
Critère :							
Ou :							

……

Champ :						
Table :						
Mise à jour						
Critères						
Ou :						

Application

Soit la base de données « **COMMERCIALE** » suivante :

Client (Code-client , Nom, Prénom, ville, Code-postale, Téléphone)

Commande (N°commande , Date commande ,adrsse-liv, Code-client)

Produit (code-Produit ,Des_Produit ,PU ,Qtstk)

Détail_ commande (N°commande , Code-Produit , Qte_commandé)

1. Préciser le type de données de chaque champ dans chaque table.

Nom table	
champ	*Type de données*

Nom table	
champ	*Type de données*

Nom table	
champ	*Type de données*

Nom table	
champ	*Type de données*

2. Souligner la clé primaire de chaque table.
3. Mettre « # » devant chaque clé étrangère
4. Schématiser les tables et établir les liens existants ente les tables.
5. Dans le dossier « bac2023 », créer un dossier intitulé « votre nom » dans lequel vous enregistrez votre travail.
6. Lancer le logiciel de création de bases de données.
7. Créer dans votre dossier de travail la base « commerciale ».
8. Créer les tables de cette base ainsi que les relations.
9. Remplir les tables par les données suivantes :

Table client

Code-client	nom	prénom	ville	Code-postale	téléphone
C120	Gurmazi	sami	Sfax	2134	74555444
C125	soltani	imen	Mahdia	5100	73200200
C420	tounsi	anis	tunis	3000	71555666

Table commande

N°commande	Date commande	Adresse-liv	Code-client
Cm1	12/03/2010	Métlaoui	C120
Cm2	15/03/2010	Mahdia	C125
Cm3	20/03/2010	Tunis	C420

Table produit

Code-produit	Des-produit	PU	Qtstk
GR	graveur	100	20
DD	Disque dur	120	30
IMPLZ	imprimante laser	125	15

Table Détail-commande

N°commande	Code-produit	Qté-commandé
Cm1	DD	2
Cm2	IMPLZ	3
Cm1	GR	5
Cm2	DD	4
Cm3	IMPLZ	2

10. Créer les requêtes permettant :

- **A.** d'afficher la liste des clients de « Sfax ».
- **B.** d'afficher la liste des clients possédant un nom contenant la lettre « s ».
- **C.** d'afficher la liste des clients dont leurs prénoms commencent par la lettre « a ».
- **D.** d'afficher la liste de produits (code-produit, Des-produit) ayant un prix unitaire >= 120.
- **E.** d'afficher la liste de clients (code-client, nom, prénom) qui ont passé une commande après le 14/03/2010.
- **F.** d'afficher le code, la désignation, le prix unitaire et la quantité commandée des produits ayant une quantité commandée >3.
- **G.** En donnant le nom d'un client, afficher son numéro de téléphone
- **H.** En donnant la désignation d'un produit, afficher sa quantité commandé
- **I.** d'afficher la liste des commandes pour une date commande saisie du clavier.
- **J.** d'augmenter le prix unitaire de produit de 8% de prix initiale.
- **K.** de modifier le code postal des clients de la ville de « Sfax » à 2136.

Correction des requêtes

……

Champ :							
Table :							
Tri :							
Afficher :							
Critère :							
Ou :							

……

Champ :							
Table :							
Tri :							
Afficher :							
Critère :							

Ou :							

……

Champ :							
Table :							
Tri :							
Afficher :							
Critère :							
Ou :							

……

Champ :							
Table :							
Tri :							
Afficher :							
Critère :							
Ou :							

……

Champ :							
Table :							
Tri :							
Afficher :							
Critère :							
Ou :							

……

Champ :							
Table :							
Tri :							
Afficher :							
Critère :							
Ou :							

……

Champ :							
Table :							
Tri :							
Afficher :							
Critère :							
Ou :							

……

Champ :							
Table :							
Tri :							
Afficher :							
Critère :							
Ou :							

……

Champ :						
Table :						
Mise à jour						
Critères						
Ou :						

……

Champ :						
Table :						

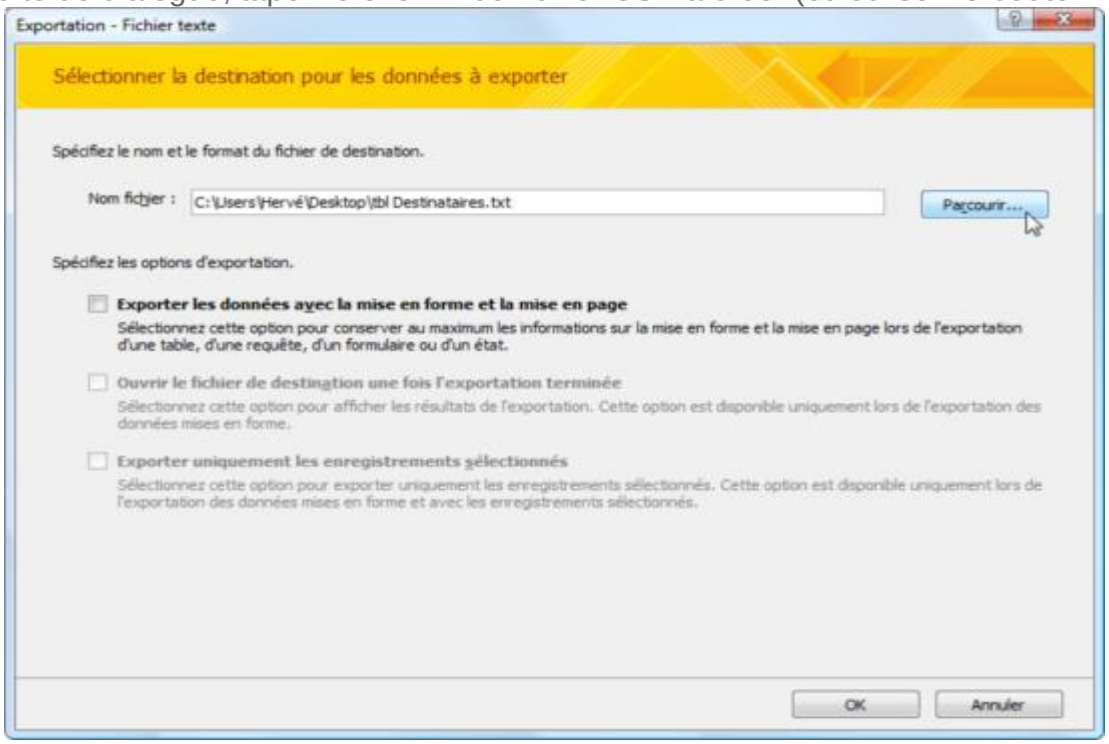
Mise à jour						
Critères						
Ou :						

Exportation csv et xlsx

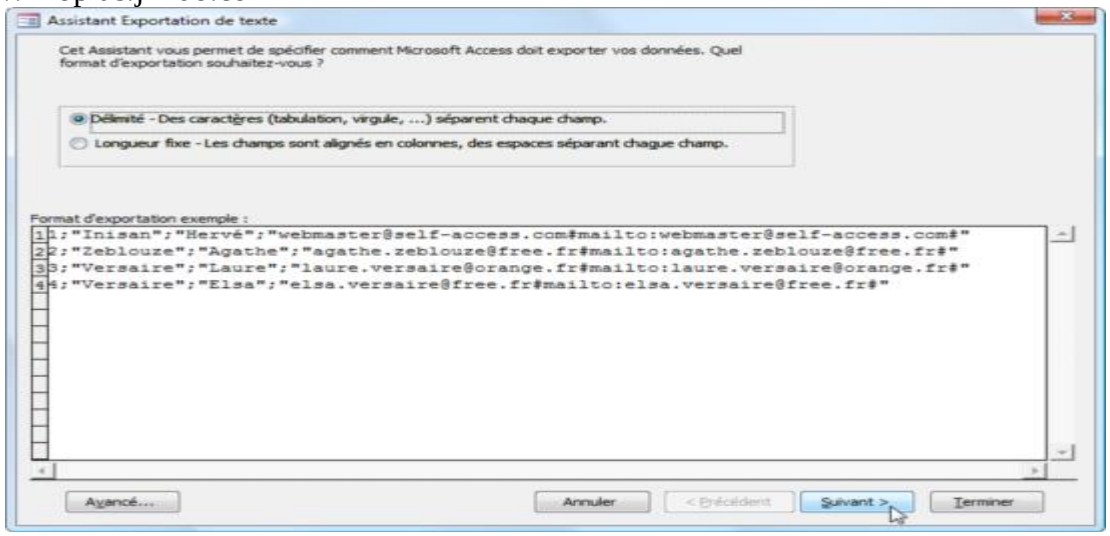
1. Sélectionnez la table ou requête à exporter.

2. Activez l'onglet **Données externes** du ruban.
3. Cliquez sur l'icône *Fichier texte* du groupe *Exporter* (à ne pas confondre avec le groupe *Importer et lier*, qui ferait l'inverse !).

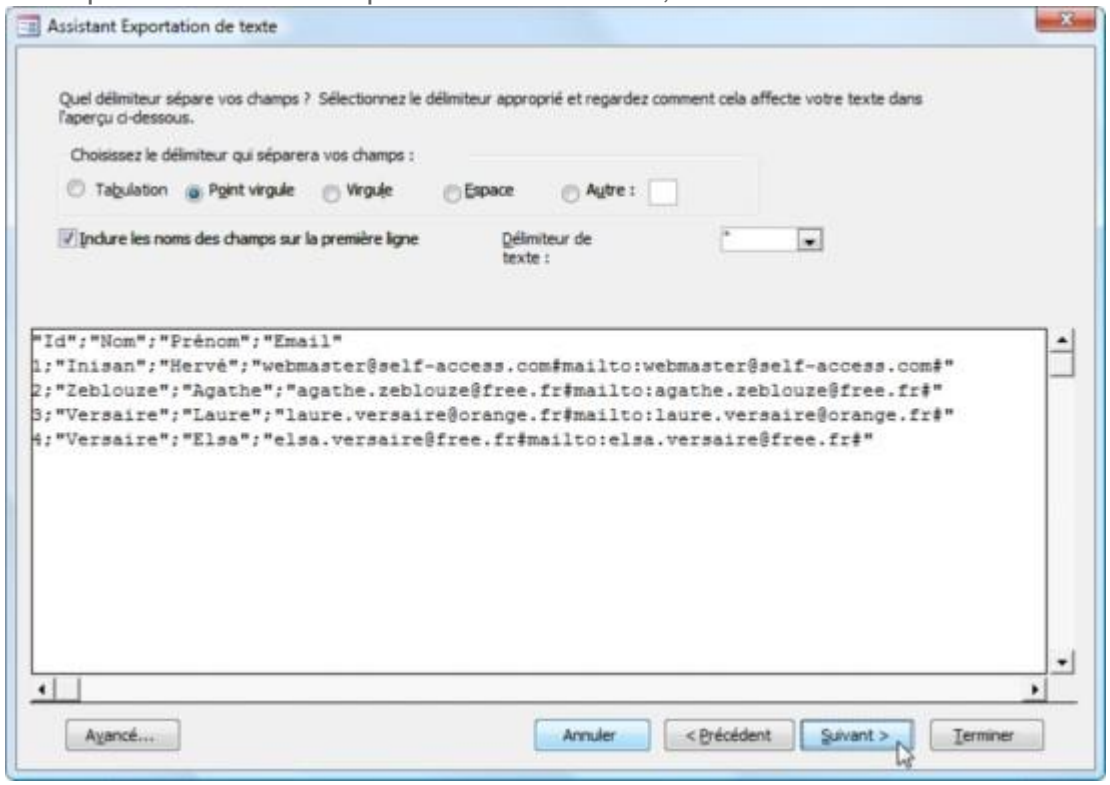
4. Dans la boîte de dialogue, tapez le chemin du fichier CSV à créer (ou utilisez le bouton **Parcourir**, c'est sans doute plus pratique).



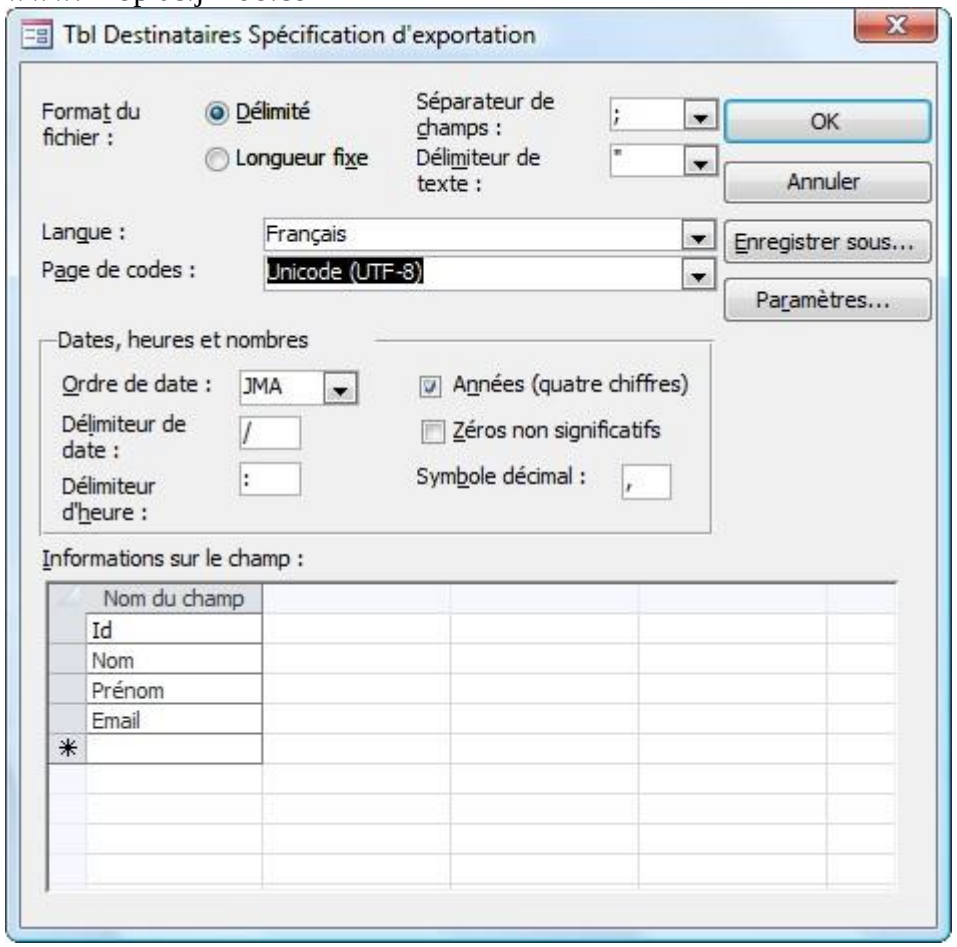
5. Cliquez sur **OK** lorsque le chemin est correct. Vous obtenez un aperçu des données, et le choix du type d'exportation (conservez *Délimité* pour un fichier CSV).


Assistant Exportation de texte
Cet Assistant vous permet de spécifier comment Microsoft Access doit exporter vos données. Quel format d'exportation souhaitez-vous ?
Délimité - Des caractères (tabulation, virgule, ...) séparent chaque champ.
Longueur fixe - Les champs sont alignés en colonnes, des espaces séparant chague champ.
Format d'exportation exemple :
1;"Inisan";"Hervé";"webmaster@self-access.com#mailto:webmaster@self-access.com#"
2;"Zeblouze";"Agathe";"agathe.zeblouze@free.fr#mailto:agathe.zeblouze@free.fr#"
3;"Versaire";"Laure";"laure.versaire@orange.fr#mailto:laure.versaire@orange.fr#"
4;"Versaire";"Elsa";"elsa.versaire@free.fr#mailto:elsa.versaire@free.fr#"
Avancé...
Annuler
< Précédent
Suivant >
Terminer

6. Vous pouvez maintenant préciser le délimiteur, c'est-à-dire le caractère utilisé pour séparer les données dans le fichier CSV.



7. Cliquez encore sur le bouton **Suivant**. Vous pouvez encore modifier le nom du fichier de destination, si nécessaire, avant de cliquer sur le bouton **Terminer**.
8. Il est possible ensuite de mémoriser toute votre séquence pour la « rejouer » ultérieurement. Dans ce cas, vous devrez nommer cette séquence, et lui donner une description.


Tbl Destinataires Spécification d'exportation
Format du fichier :
Délimité
Longueur fixe
Séparateur de champs :
;
Délimiteur de texte :
"
OK
Annuler
Langue :
Français
Page de codes :
Unicode (UTF-8)
Enregistrer sous...
Paramètres...
Dates, heures et nombres
Ordre de date :
JMA
Années (quatre chiffres)
Délimiteur de date :
/
Zéros non significatifs
Délimiteur d'heure :
:
Symbole décimal :
,
Informations sur le champ :
Nom du champ
Id
Nom
Prénom
Email

Partie 2
Analyse de données

Python : Manipulation des données avec Pandas

Pandas est une librairie Python spécialisée dans l'analyse des données. Dans ce support, nous nous intéresserons surtout aux fonctionnalités de manipulations de données qu'elle propose. Un objet de type "data frame" permetderéaliserdenombreusesopérationsdefiltrage,prétraitements,etc.,préalablesàlamodélisationstatistique.

1. Librairie Pandas

```
#Première étape : il faut charger la librairie Pandas
import pandas
```

2. Structure DataFrame

Concernant notre fichier Excel **« heart.xlsx »**: il s'agit d'un **Tableau en Excel**, la première**ligne** correspond aux noms des colonnes (des variables) ; à partir de la seconde ligne, nous disposons des valeurs pour chaque**colonne**(variable).

	A	B	C	D	E	F	G	H
1	age	sexe	type douleur	sucre	taux max	angine	depression	coeur
2	70	masculin	D	A	109	non	24	presence
3	67	feminin	C	A	160	non	16	absence
4	57	masculin	B	A	141	non	3	presence
5	64	masculin	D	A	105	oui	2	absence
6	74	feminin	B	A	121	oui	2	absence

Un Tableau **DataFrame** correspond à un **Tableau en Python** où les **lignes** correspondent à des observations (valeurs, calculs,..) les noms de **colonnes** correspondent à des noms décrivant les observations.

3. Chargement d'un Tableau Excel dans un DataFrame

Dans ce qui suit, nous chargeons **le fichier Excel « heart.xlsx »** dans un **DataFrame « df »** cet nous procédons à quelques vérifications.

2

```
#Chargement du fichier dans l'objet de type Data Frame nommé <df>
df = pandas.read_excel("heart.xlsx")
#Afficher le type de df
print ( type(df) )
```

```
<class 'pandas.core.frame.DataFrame'>
```

4. La structure DataFrame

3

```
#Afficher la structure d'un DataFrame: nombre de lignes et nombre de colonnes
print(df.shape)
```

```
(270, 8)
```

4

```
#Afficher les premières lignes du DataFrame
print(df.head() )
```

	Age	sexe	typedouleur	sucre	tauxmax	angine	depression	coeur
1	70	masculin	D	A	109	non	24	presence
2	67	feminin	C	A	160	non	16	absence
3	57	masculin	B	A	141	non	3	presence
4	64	masculin	D	A	105	oui	2	absence
5	74	feminin	B	A	121	oui	2	absence

5

```
#Afficher les dernières lignes du DataFrame
print(df.tail() )
```

6

```
#Afficher les colonnes
print(df.columns)
```

```
Index(['age', 'sexe', 'typedouleur', 'sucre', 'tauxmax', 'angine', 'depression', 'coeur'],dtype='object')
```

7

```
#Afficher le type de chaque colonne
print(df.dtypes)
```

```
age                int64
sexe               object
typedouleur        object sucre     object
tauxmax            int64
angine             object
depression         int64
coeur              object dtype:object
```

8

```
#Afficher des informations sur les données
print(df.info() )
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex:    270 entries, 0 to 269 Data columns (total 8columns):
age            270 non-nullint64
sexe270 non-null object
typedouleur270 non-nullobject
sucre270 non-nullobject
tauxmax270 non-null int64
angine         270 non-null object
depression     270 non-null int64
coeur270 non-null object
dtypes: int64(3),object(5)
memory usage:  17.0+ KB None
```

9

```
#Description des données
print(df.describe() )
```

	age	sexe	typedouleur	sucre	tauxmax	angine	depression
\							
count	270.000000	270	270	270	270.000000	270	270.0
unique	NaN	2	4	2	NaN	2	NaN
top	NaN	masculin	D	A	NaN	non	NaN
freq	NaN	183	129	230	NaN	181	NaN
mean	54.433333	NaN	NaN	NaN	149.677778	NaN	10.5
...	...	...	...	...	...	...	...

min	29.000000	NaN	NaN	NaN	71.000000	NaN	0.0
25%	48.000000	NaN	NaN	NaN	133.000000	NaN	0.0
50%	55.000000	NaN	NaN	NaN	153.500000	NaN	8.0
75%	61.000000	NaN	NaN	NaN	166.000000	NaN	16.0
max	77.000000	NaN	NaN	NaN	202.000000	NaN	62.0

	coeur		
count	270		
unique	2		
top	absence		
freq	150		
mean	NaN		
...	...		
min	NaN		
25%	NaN		
50%	NaN		
75%	NaN		
max	NaN		

[11 rows x 8 columns]

Certains indicateurs statistiques ne sont valables que pour les variables numériques (ex. moyenne, min, etc. pour age, tauxmax,...), et inversement pour les non-numériques (ex. top, freq, etc. pour sexe, typedouleur, ...), d'où les **NaN** dans certaines situations.

5. Manipulation des données

10

```
#Afficher les données d'une colonne (sexe par exemple)
print(df [ 'sexe' ])
```

```
1        masculin
2          feminin
3        masculin
4        masculin
5          feminin
           ...
265      masculin
266      masculin
267        feminin
268      masculin
269      masculin
```

```
Name: sexe, dtype: object
```

11

```
#Afficher les données d'une colonne:autre manière avec « . »
print(df.sexe)
```

```
1        masculin
2          feminin
3        masculin
4        masculin
5          feminin
           ...
265      masculin
266      masculin
267        feminin
268      masculin
269      masculin
Name: sexe, dtype: object
```

13

```
#accéder à un ensemble de colonnes
print(df[['sexe','sucre' ]] )
```

```
            sexe   sucre
0        masculin       A
1         feminin       A
2        masculin       A
3        masculin       A
4         feminin       A
..            ...     ...
269      masculin       A


[270 rows x 2 columns]
```

13

```
#Une colonne est un Tableau, pour afficher les premières valeurs de la colonne age
print(df['age'].head( ))
```

```
0       70
1       67
2       57
3       64
4       74
Name: age, dtype: int64
```

14
```
#Afichage des dernières valeurs de la colonne age
print(df['age'].tail( ))
```

```
265       52
266       44
267       56
268       57
269       67
Name: age, dtype: int64
```

15
```
#Afficher les statistiques descriptives.
print(df['age'].describe())
```

```
count       270.000000
mean         54.433333
std           9.109067
min          29.000000
25%          48.000000
50%          55.000000
75%          61.000000
max          77.000000
Name:   age, dtype: float64
```

16
```
#calculer la somme (somme des tauxmax )
print(df['tauxmax'].sum())
```

```
636
```

17
```
#calculer la moyenne (moyenne de l'age)
print(df['age'].mean())
```

```
54.433333
```

18
```
#calculer le minimum (minimum de l'age)
print(df['age'].min() )
```

```
29.000000
```

19
```
#calculer le maximum (maximum de l'age)
print(df['age'].max())
```

```
77.000000
```

20
```
#Calculer le nombre de valeurs
print(df['typedouleur'].value_counts( ) )
```

```
D     129
C      79
B      42
A      20
Name: typedouleur, dtype: int64
```

21
```
#df['age']est un tableau d'ages, il est possible d'afficher  la première valeur de la manière suivante
print(df['age'][0])
```

```
70
```

22
```
#Afficher les 3 premières valeurs
print(df['age'][0:3])
#ou bien aussi
print(df.age[0:3])
```

```
0    70
1    67
2    57
Name: age, dtype: int64
```

6. Trier les données d'un DataFrame

trier consiste à ordonner les données unsous-ensembledelignesquivérifieune**condition**

```
#trier les ages de manière croissante
print(df['age'].sort_values())

#trier les ages de manière décroissante
print(df['age'].sort_values(ascending=False))

#ou bien aussi,trier tous les données du  dataFrame de manière croissante selon l'age
print(df.sort_values(by='age'))
```

23

```
214      29
174      34
138      34
224      35
81       35
         ..
15       71
255      71
4        74
73       76
199      77
Name:   age, dtype: int64
```

La plus petite valeur est 29, elle correspond à l'observation n°214.

24
```
#Afficher les indices des valeurs triées
print( df ['age' ].argsort( ) )
```

```
0        214
1        174
2        138
3        224
4         81
       ...
Name:age,   dtype: int64
```

214 est le numéro de l'individu portant la plus petite valeur de la variable age,

25
```
#le résultat du tri est aussi un DataFrame
tri= df['age'].sort_values()
print(tri.head())
```

	age	sexe	typedouleur	sucre	tauxmax	angine	depression	coeur
214	29	masculin	B	A	202	non	0	absence
174	34	masculin	A	A	174	non	0	absence
138	34	feminin	B	A	192	non	7	absence
224	35	feminin	D	A	182	non	14	absence
81	35	masculin	D	A	130	oui	16	presence

7. Filtrer les données d'un DataFrame

Filtrer consiste à sélectionner unsous-ensembledelignesquivérifieune**condition**

26
```
#Afficher les lignes qui représentent des hommes
hommes = df [df.sexe=="Masculin"]
print(hommes.head() )
```

	age	sexe	typedouleursucre	tauxmaxangine		depression	coeur
0	70	masculin	D	A	109	non	24 presence
1	57	masculin	B	A	141	non	3 presence
2	64	masculin	D	A	105	oui	2 absence

27
```
#Afficher les lignes qui représententdes hommes âgés de 64 ans
hommes= df[(df.age== 64)&(df.sexe=="Masculin")]
print(homme.head() )
```

	age	sexe	typedouleursucre	tauxmaxangine		depression	coeur
0	64	masculin	D	A	105	oui	2 absence

8. Accès indicé aux données d'un DataFrame

On peut accéder aux valeurs du DataFrame via des indices ou plages d'indice. La structure se comporte alors comme une matrice. La cellule en haut et à gauche est de coordonnées (0,0).Il y a différentes manières de le faire, l'utilisation de **.iloc[,]** constitue une des solutions les plus simples. N'oublions pas que Shape permet d'obtenir les dimensions (lignes et colonnes) du DataFrame.

28
```
#Afficher la première ligne
print( df.iloc [ 0 ] )
```

```
age              70
sexe           masculin
typedouleur    D sucre      A
tauxmax        109
angine         non
depression     24
coeur          presence
Name=0, dtype: object
```

29
```
#Afficher la valeur située en (ligne 0, colonne 0)
print( df.iloc [ 0,0 ] )
```

```
70
```

30
```
#Valeur située en dernière ligne (-1), première colonne (0)
print( df.iloc [-1,0 ] )
```

```
67
```

31
```
#Afficher les 5 premières lignes
print(df.iloc[0:5])
```

	age	sexe	typedouleur	sucre	tauxmax	angine	dépression		coeur
1	70	masculin	D	A	109	non	24	presence	
2	67	feminin	C	A	160	non	16	absence	
3	57	masculin	B	A	141	non	3	presence	
4	64	masculin	D	A	105	oui	2	absence	
5	74	feminin	B	A	121	oui	2	absence	

32
```
#Afficher les 5 premières lignes, des deux premières colonnes
print( df.iloc [0:5,0:2 ] )
```

	age	sexe
1	70	masculin
2	67	feminin
3	57	masculin
4	64	masculin
5	74	feminin

33
```
#Afficher les 5 premières lignes et colonnes 0, 1 et 4
print(df.iloc[0:5,[0,2,4]])
```

```
   agetypedouleur  tauxmax 070   D              109
1              67         C  160
2              57         B  141
3              64         D  105
4              74         B  121
```

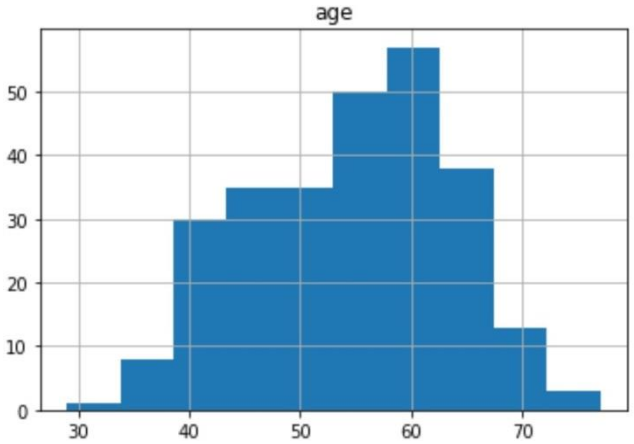
9. Graphiques

Passer par **matplotlib** permet de réaliser des graphiques ~~performants~~ (http://matplotlib.org/). Mais il faut connaître les procédures de la librairie, ce qui nécessite un apprentissage supplémentaire qui n'est pas toujours évident.

Heureusement, Pandas propose des commandes simples qui encapsulent l'appel à ces procédures et nous simplifie grandement la vie. Il faut importer **matplotlib** pour que l'ensemble fonctionne correctement.

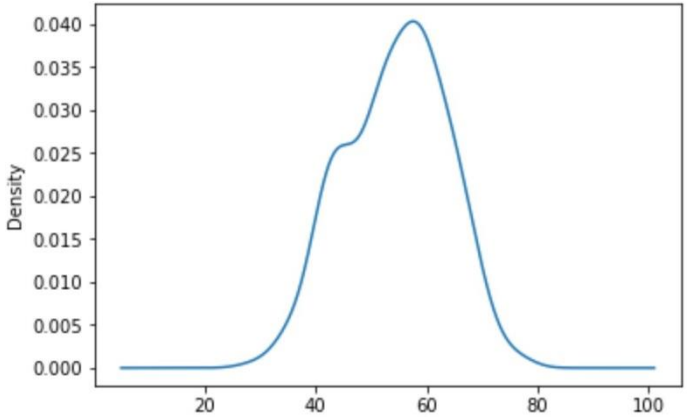
38
```
#importation de la librairie des courbes
Import matplotlib.pyplot as plt
```

39
```
#histogramme del'âge
df.hist(column='age')
```



40

```
#Courbe de densité
df['age'].plot.kde()
```



41

```
#histogrammes de l'âge selon le sexe
df.hist(column='age',by='sexe')
```