



CHAPITRE 6 : LES FICHIERS

INTRODUCTION

Structures de données utilisées

Résident en mémoire centrale

→ Effacées si on termine le programme ou si l'ordinateur se plante

Nécessité de Fichiers

- Manipuler quantité de données ➤ capacité Mémoire Centrale
- Stocker des données au delà de la vie du programme
- Programme indépendant des données
- Partager les données entre programmes
- Limiter les saisies utilisateur

DÉFINITION DES FICHIERS

- Ensemble structuré de données de même type, nommé et stocké de manière linéaire dans une mémoire secondaire
 - sur un support externe (disque dur, disque optique, bande magnétique, ...).
- Peut contenir :
 - des caractères (fichiers textes), des programmes, des valeurs (fichiers de données)
- Taille du fichier est non connue à l'avance

ORGANISATION DES FICHIERS

Bloc 1	Bloc 2	Bloc 3			Bloc n
Enregistrement 0	Enregistrement 1	Enregistrement 2			(enregistrement n-1)

- Pour accéder ou modifier un bloc, on doit d'abord le localiser (connaitre son adresse ou numéro d'ordre)

Bloc 1	Bloc 2	Bloc 3			Bloc n
Enregistrement 0	Enregistrement 1	Enregistrement 2			(enregistrement n-1)

Pointeur

Placer le pointeur à l'adresse 0

La valeur du pointeur peut changer ... **chercher son numéro d'ordre**



LES TYPES DE FICHIERS

1.FICHER BINAIRE

2.FICHER TEXTE

LES FICHIERS TEXTES

- Stocke des données de type **caractères**, organisés en lignes
- Lisible et modifiable par **n'importe quel éditeur de texte**
- Chaque ligne se termine par une **marque de fin de ligne (EOL)**.
- Le fichier se termine par une **marque de fin de fichier (EOF)**.
- La lecture se fait **ligne par ligne**
- Un enregistrement se rajoute **uniquement à la fin du fichier**.

Déclaration d'un fichier texte

Var
F1, F2 :
Fichier



Les procédures prédéfinies:

**Lire, Lireln, Ecrire,
Ecrireln**

FICHIERS BINAIRES

On peut y stocker n'importe quoi !!!

- Suite de données (**octets**) écrites les unes après les autres, représentées sous forme **binaire**.
- Illisible à partir d'un éditeur de texte. Seul le programme avec lequel il a été édité peut l'Ouvrir => La structure des enregistrements est dépendante de l'interprétation du programme.
- Organisés en structures de données (enregistrements).
 - Les enregistrements peuvent être de longueur fixe, collés les uns après les autres sans retour à la ligne.
 - La lecture se fait enregistrement par enregistrement.

FICHIERS BINAIRES

Déclaration d'un fichier binaire

Type

t_personne = Enregistrement
 Nom, Prenom : chaine
 Adresse : t_adr
 Date_naissance: t_date

FinEnregistrement

Var

F : **Fichier d'**entiers
Fpersonne: **Fichier de** t_personne

ORGANISATION DES FICHIERS

- Les modes d'implémentation des enregistrements dans le fichier qui déterminent les propriétés d'accès.
- **3 types d'accès : séquentiel, direct et indexé**
- Chacune a ses particularités pour la création, les modifications ou les mises à jour des enregistrements
- Toutes les organisations ne sont pas possibles sur tous les supports (dépendent du support de stockage)
 - Ex: sur une bande magnétique, seule une organisation séquentielle est possible

1ÈRE PARTIE

FICHERS À ACCÈS SÉQUENTIEL

DÉFINITIONS ...

- L'organisation séquentielle convient pour **les 2 types** de fichiers: textes et binaires.
- Suite d'enregistrements disposés les uns à la suite des autres dans lequel on ne peut Lire un enregistrement qu'après avoir lu tous les enregistrements qui le précèdent.
 - Ex. Une bande audio est à accès séquentiel. On ne peut écouter une chanson qu'après avoir fait défiler les chansons précédentes.
- Enregistrements mémorisés consécutivement dans **l'ordre** de leur entrée et peuvent seulement être lus dans cet ordre
 - Accès aux enregistrements **dans leur ordre d'écriture**

DÉCLARATION D'UN FICHIER SÉQUENTIEL

Fichier texte

Var

F1, F2 : **Fichier séquentiel**

Fichier binaire

Type

t_personne = **Enregistrement**

Nom, Prenom : chaine

Adresse : t_adr

FinEnregistrement

Var

F_personne: **Fichier séquentiel de**

t_personne

F_entiers : **fichier séquentiel d'entiers**

F_reels : **fichier séquentiel de réels**

UTILISATION DE FICHIERS

1. Avant d'utiliser un fichier il faut **Associer** un nom logique à son nom physique

Associer (nom logique, nom physique)

- Si le fichier existe alors le fichier sera ouvert et son contenu sera effacé sinon il faut le créer

Créer (nom_logique)

- *A sa création, le fichier n'est pas vide, il possède la marque de EOF*

2. Ouvrir un fichier existant et définir le mode d'ouverture

Ouvrir (nom_logique, mode d'ouverture)

CRÉATION D'UN FICHIER BINAIRE À ACCÈS SÉQUENTIEL

Algorithme Fichier_seq_binaire

Type Tpersonne = enregistrementfin enregistrement

Var Fpersonne : fichier séquentiel de Tpersonne

NomFic : chaîne de caractères

Début

Ecrire "Donner l'adresse physique du fichier:"

Lire (NomFic)

Associer (Fpersonne, NomFic)

Si Existe (Fpersonne) Alors

Ecrire ("Création impossible", Nomf, "existe déjà.")

Sinon Créer (Fpersonne)

Finsi



.....

Fin

***Un fichier vide ne contient que la
marque EOF***

CRÉATION D'UN FICHIER TEXTE À ACCÈS SÉQUENTIEL

Algorithme Fichier_seq_texte

Var F1 : fichier séquentiel

NomFic : chaîne de caractères

Début

Ecrire "Donner le nom externe du fichier:"

Lire (NomFic)

Associer (F1, NomFic)

Si Existe(F1) Alors

Ecrire ("Création impossible", Nomf, "existe déjà.")

Sinon Créer (F1)

Finsi

.....

Fin

ÉTAPES D'UTILISATION DES FICHIERS

1. Ouvrir le fichier

Ouvrir (Nomlogique, Mode d'ouverture)

2. Selon le mode d'ouverture, Traiter
le contenu du fichier:

Le Lire, y écrire, toutes les opérations
souhaitées pour accéder et manipuler son
contenu

3. Fermer le fichier: quand tous les
traitements sont terminés.

MODE D'OUVERTURE D'UN FICHIER

1. **Lecture:** Consultation – l'accès aux données, sans modification possible du contenu. La lecture du contenu du fichier se fait à l'aide de l'instruction **Lire**.
2. **Ecriture:** Le contenu antérieur du fichier sera effacé; Ecrire de nouveaux enregistrements à la fin du fichier à l'aide de l'instruction **Ecrire**.
3. **Ajout:** Le contenu antérieur du fichier sera préservé; Ecrire de nouveaux enregistrements à la fin du fichier à l'aide de l'instruction **Ajouter**.

POSSIBLES ERREURS D'OUVERTURE D'UN FICHIER

- 1. Aucun nom externe de fichier n'est assigné au nom logique*
- 2. Un nom externe de fichier est assigné au nom logique mais le fichier n'existe pas*
- 3. Le fichier est déjà ouvert => il faut le reFermer d'abord car il est entrain de servir un autre fichier externe*

OUVERTURE DE FICHIERS TEXTES

Algorithme Traitement_Fichiers

Var

F1, F2, F3 : Fichier Sequentiel

Debut

Associer (F1, "C:/Carnet.txt")

Associer (F2, "C:/Temp.txt")

Associer (F3, "C:/Agenda.txt")

***Ouverture
en lecture***

{ Si Existe (F1) Alors Ouvrir (F1, Lecture)
Sinon Créer (F1)
Finsi

***Ouverture
en écriture***

{ Si Existe (F2) Alors Ouvrir (F2, Ecriture)
Sinon Créer (F2)
Finsi

***Ouverture en
mode ajout***

{ Si Existe (F3) Alors Ouvrir (F3, Ajout)
Sinon Créer (F3)
Finsi

LECTURE D'ENREGISTREMENTS

1. Ouverture du fichier en mode lecture

Ouvrir (Nom_fichier , Lecture)

Ouvre le fichier en laissant les données intactes et le curseur de parcours pointe sur le premier enregistrement.

2. Lire (Nom_fichier , Nom_variable)

- Copie le contenu de l'enregistrement courant pointé par le curseur de parcours dans la variable. Cette dernière doit avoir la même structure que les enregistrement du fichier. Ensuite le curseur se place sur l'enregistrement suivant.

EX DE LECTURE FICHIER BINAIRE

Type t_employé = **Enregistrement**

Nom, Prenom,..... FinEnregistrement

Var

E : t_employé

Fpersonnel: Fichier Séquentiel de t_employé

Début

Associer (Fpersonnel,
« C:/Documents/Personnel.dat»)

Ouvrir (Fpersonnel, « Lecture »)

Tant que **Non EOF(Fpersonnel)** Faire

Lire (Fpersonnel, E)

Fintantque

Fermer (Fpersonne)

Fin

LECTURE FICHIER TEXTE

- Ouverture en mode Lecture : **Ouvrir (NomF, Lecture)**
- Lecture caractère par caractère : **Lire (NomF, V_Car)**
- Lecture ligne par ligne : **Lireln (NomF, Ligne)**
- Positionner au début de la ligne suivante : **Lireln (NomF)**
- Une lecture correcte doit tenir compte des marques de fin de ligne et de fichier sous peine de provoquer des erreurs.
 - EOLN (NomF) vrai si le **prochain** caractère à Lire est la marque de fin de ligne
 - EOF (NomF) vrai si le **prochain** caractère à Lire est la marque de fin de fichier

TRAITEMENTS DE LECTURE SUR FICHIER TEXTE

Lecture caractère par caractère

```
Tant que Non EOF( NomF )  
faire  
    Tant que Non EOL (NomF )  
    faire  
        Lire ( NomF, Var_car)  
        ...  
    (* traiter le caractère lue *)  
    Fintantque  
    Lireln (NomF, Ligne)  
Fintantque
```

Lecture ligne par ligne

```
Tant que Non EOF (NomF ) faire  
    Lireln ( F, Ligne)  
    ... (* traiter la chaîne lue *)  
Fintantque
```

AJOUTER UN ENREGISTREMENT

1. Ouverture du fichier en mode Ajout
 - **Ouvrir (NomFichier, Ajout)**
2. L'**ajout** d'un enregistrement se fait **à la fin du fichier**
 - *Ajouter (NomFichier, Nom_variable)*
 - La variable doit obligatoirement être de même structure que les enregistrement du fichier.
3. Fermer le fichier

ECRIRE UN ENREGISTREMENT

1. Ouverture du fichier en mode Ecriture
 - Ouvrir (NomFichier, Ecriture)
2. **Ecriture de nouveaux enregistrements : A la fin du fichier**
 - **Ecrire (*nom_fichier, nom_variable*)**
‘la variable est du type Enregistrement
 - Recopier le contenu d'une variable à la fin du fichier.
 - L'écriture se fait enregistrement par enregistrement => la variable écrite dans le fichier doit obligatoirement être du type des enregistrements.

EXERCICE

- On travaille avec le fichier du carnet d'adresses "C:/Documents/Adresses.txt" ayant chaque ligne la structure suivante :
 - Nom
 - Prénom
 - Tel
 - Courriel
- Ecrire un algorithme qui permet à l'utilisateur de saisir au clavier un nouvel individu qui sera ajouté à ce carnet d'adresses.
- Vous donnez les deux méthodes (ajouter et écrire)

AJOUT D'UN ENREGISTREMENT DANS UN FICHIER TEXTE

Algorithme Ajout Adresse

Var : Ligne , Nom, Prenom, Tel, Courriel : chaîne de caractères

FAdresse : **Fichier Séquentiel**

Début

Associer (FAdresse, "C:/Documents/Adresses.txt")

Ouvrir (F, Ajout)

Lire ("Entrez le nom", Nom)

Lire ("Entrez le prénom", Prenom)

Lire ("Entrez le numéro de téléphone", Tel)

Lire ("Entrez le courriel", Courriel)

Ligne ← Nom & Prenom & Tel & Courriel

Ajouter (FAdresse, Ligne)

Fermer (FAdresse)

Fin

ECRITURE D'ENREGISTREMENT DANS UN FICHIER TEXTE

Algorithme Ajout Adresse

Var : Ligne , Nom, Prenom, Tel, Courriel : chaîne de caractères

F : **fichier séquentiel**

Début

Associer (F, "C:/Documents/adresses.txt")

Ouvrir (F, Ecriture)

Lire ("Entrez le nom", Nom)

Lire ("Entrez le prénom", Prenom)

Lire ("Entrez le numéro de téléphone", Tel)

Lire ("Entrez le courriel", Courriel)

Ligne \leftarrow Nom & Prenom & Tel & Courriel

Ecrire (F, Ligne)

Fermer (F)

Fin

Algorithme trait_fich_employes

TYPE employe = enregistrement

num : ENTIER

nom, email : CHAINE

FinEnregistrement

VAR E : employe

Femploye : **fichier séquentiel de employe**

DEBUT

Associer (Femploye, «

C:/Documents/Personnel.dat»)

Ouvrir (Femploye, Ecriture)

Écrire ("entrez le numéro, le nom et l'adresse email")

Lire (E.num, E.nom, E.email)

Ecrire (Femploye, E)

Fermer (Femploye)

FIN

Var empl: employe

Début

empl.num ← 12435566

empl.nom ← « Dupont »

...

Ecrire (F, empl)

...

```

Algorithme trait_fich_personnes
Type Tpersonne = enregistrement
.....fin enregistrement
Var Fpersonne : fichier séquentiel de
    Tpersonne
    NomFic, continue : chaîne de
    caractères
    pers : Tpersonne
Début
    Ecrire « donner le nom externe du
    fichier »
    Lire (NomFic)
Associer (Fpersonne,
NomFic)
Ouvrir (Fpersonne, Ajout)
Lire (Continue)
Tant que continue = « Oui »

```

```

Lire (pers.Nom)
Si pers.Nom <> « » alors
    Lire (pers.Prénom)
    .....
    Lire (pers.Courriel)

```

Ajouter (Fpersonne, pers)

**Ecrire « voulez vous
continuer ? »**

Lire (continue)

Ftantque

Fermer (Fpersonne)

Fin

SUPPRESSION FICHER BINAIRE À ACCÈS SÉQUENTIEL

Type Tpersonne = enregistrementfin enregistrement

Var Fpersonne : fichier séquentiel de Tpersonne

NomFic : chaîne de caractères

Début

Ecrire "Donner le nom externe du fichier:"

Lire (NomFic)

Associer (Fpersonne, NomFic)

Si Non Existe(Fpersonne) Alors

Ecrire ("Suppression impossible", Nomf, "n'existe pas")

Sinon Supprimer (Fpersonne)

Finsi

SUPPRESSION FICHER TEXTE À ACCÈS SÉQUENTIEL

Var F1 : fichier séquentiel

NomFic : chaîne de caractères

Début

écrire « donner le nom externe du fichier »

Lire (NomFic)

Associer **(F1, NomFic)**

Si Non Existe (F1) Alors

Ecrire (« suppression impossible », Nomf, « n'existe pas »)

Sinon Supprimer (F1)

Finsi

MISE À JOUR D'UN FICHIER SÉQUENTIEL

- Il est impossible de:
 - modifier directement un enregistrement,
 - ajouter un nouvel enregistrement ailleurs qu'à la fin,
 - supprimer physiquement un enregistrement.
- Ces opérations de mise à jour ne peuvent se faire qu'en **réécrivant complètement le fichier**.
- **Pas de modification directe** : Pour modifier ou insérer un enregistrement, **il faudra tout réécrire**
- **La mise à jour directe n'est possible qu'avec les fichiers à accès direct.**

EXERCICE

- On voudrait saisir et enregistrer les notes des étudiants d'une classe dans un fichier de réels. Déclarer la structure de fichier adéquate
- Au lieu de sauvegarder les notes uniquement, on voudrait enregistrer aussi les fiches de renseignements de tous les étudiants dans un fichier appelé CARNET. Une fiche de renseignement comporte les informations suivantes :
 - Nom et Prénom
 - Tel, Adresse, Courriel
 - Date de naissance
 - Notes : note1, note2 et note3
- Déclarer le type T_renseignement et le type de fichier dans lequel vous allez enregistrer ces données
- Associez la variable logique CARNET au nom physique où se trouve réellement le fichier sur le disque dur (C:\Archives\fichier_etudiants.dat)
- On voudrait saisir des enregistrements de 100 étudiants et les sauvegarder dans le fichier CARNET
- On voudrait afficher tous les enregistrements du fichier
- On voudrait ajouter un nouvel enregistrement au fichier CARNET sans perdre son contenu initial.
- Même question, mais cette fois le carnet est supposé être trié par ordre alphabétique. L'étudiant doit être inséré au bon endroit dans le fichier.
- Ecrire une procédure qui trie les étudiants du fichier CARNET par ordre alphabétique
- On voudrait modifier le nom de famille d'un étudiant. Pour ce faire, nous demandons à l'utilisateur quel est le nom à modifier, puis quel est le nouveau nom, et mettre à jour le fichier.
- On désire supprimer les enregistrements dont le nom est « Ben Youssef ».
- On voudrait afficher le nombre de garçons présents dans le fichier sachant que le nombre de fiches de ce fichier est inconnu.

2ÈME PARTIE

FICHIERS À ACCÈS DIRECT

- **PEUVENT ÊTRE DE TYPE TEXTE OU BINAIRE**
- **UTILISATION LIMITÉE POUR LES FICHIERS DE TEXTES**
- **ADAPTÉS PLUS POUR LES FICHIERS BINAIRES**

2- FICHIER À ACCÈS DIRECT

- Organisation : Chaque enregistrement porte un **numéro d'ordre** relatif à son emplacement dans le fichier
- **Accès direct** à n'importe quel enregistrement grâce à son numéro d'ordre (1 à tailleDe(f)).

■ **Il suffit de connaître le numéro d'ordre de l'enregistrement !!!!!**

=> sinon accès séquentiel

- A la création d'un fichier à accès direct, il est vide et ne possède pas la marque EOF.

AVANTAGES DES FICHIERS DIRECTS

- **Rapidité de la recherche** : Consultation immédiate à partir du numéro de l'enregistrement.
- **Accès rapide**: Lecture et écriture n'importe où dans le fichier
- **Améliore les fonctions de mise à jour** dans le fichier
 - Modifications de l'enregistrement
 - Ajout de nouveaux enregistrements
 - Suppression d'enregistrements
 - Création d'enregistrements dans un ordre quelconque

OUVERTURE D'UN FICHIER À ACCÈS DIRECT

- Contrairement aux fichiers séquentiels, l'ouverture d'un fichier à accès direct ne nécessite pas la spécification du mode d'ouverture quelque soit l'opération d'E/S

=> **Ouvrir (F1)**

- Pour Lire /Ecrire un enregistrement, il suffit de connaître **son numéro d'ordre**.

- **Lire (Fichier, Num, Enregistrement)**

- **Écrire (fichier, Num, Enregistrement)**

- Pour l'ajout d'un nouvel enregistrement, il sera ajouté en fin du fichier en leur attribuant **un numéro d'ordre**.

- **Ajouter (Fichier, Enregistrement)**

CRÉATION D'UN FICHIER DIRECT

Type Tdoc = enregistrement

référence, auteurs, titre : chaîne

FinEnregistrement

Var **Fdoc : Fichier Direct de Tdocument**

nomF: chaîne

Début

Ecrire « donner l'adresse physique du fichier: »

Lire (nomF)

Associer (Fd, nomF)

Si Existe (Fdoc) Alors

Ecrire (« création impossible », nomF, « existe déjà »)

Sinon **Créer (Fdoc)**

Finsi

Ouvrir (Fdoc)

.....

Fermer (Fdoc)

Fin

SUPPRESSION D'UN FICHIER À ACCÈS DIRECT

Type Tdoc = enregistrement

référence : chaîne

auteurs: chaîne

titre : chaîne

Fin enregistrement

Var **Fdoc : Fichier direct de Tdocument**

nomF: chaîne

Début

Ecrire « donner l'adresse physique du fichier: »

Lire (nomF)

Associer (Fdoc, nomF)

Si **Non Existe (Fdoc)** Alors

Ecrire (« suppression impossible », nomF, « n'existe pas »)

Sinon **Supprimer (Fdoc)**

Finsi

EXERCICE

Soit le fichier F1 de personnes (nom, prénom et age) à accès direct. Écrire l'algorithme qui permet de :

1. **Ajouter** en fin de fichier un nouvel enregistrement
2. **Modifier** un enregistrement existant
3. **Supprimer** logiquement un enregistrement existant (**** ou 00)
4. **Premier** : positionnement et affichage du premier enregistrement
5. **Dernier** : positionnement et affichage du dernier enregistrement
6. **Suivant** : passage à l'enregistrement suivant
7. **Précédent** : passage à l'enregistrement précédent
8. **Rechercher** : Rechercher un enregistrement dont on connaît le nom de la personne

ENREGISTREMENT

Type Tpers = Enregistrement

.....

Var Fpers : fichier Direct de
Tpers

NomFic, cont : chaîne de
caractères

P : Tpers

Début

Lire (NomFic)

Associer (Fpers, NomFic)

Ouvrir (Fpers)

Cont ← « Oui »

Tant que Cont = « Oui » faire

Avec P faire

Lire (Nom, Prénom, adr.rue,
adr.cp, adr.ville,
date_nais.jour,
date_nais.mois,
date_nais.année)

Ajouter (Fpers, P)

**Ecrire « Voulez vous
rajouter ? »**

Lire (Cont)

Fintantque

Fermer (Fpers)

Fin

Lire (NumE)

Si (NumE < 1 ou NumE
> TailleDe(Fdoc))

MODIFIER UN ENREGISTREMENT

enregistrement.....

Var **Fdoc : fichier
direct de Tdoc**

nomF: chaine

cont : chaine de caractères

E : tdoc

NumE : entier

Début

Ecrire « donner l'adresse
physique du fichier: »

Lire (nomF)

Associer (Fdoc, nomF)

Ouvrir (Fdoc)

Cont ← « Oui »

Alors

Ecrire « numero inexistant »

Sinon

Lire (Fdoc, NumE, E)

Ecrire (E. Nom) //afficher

.....

Lire (E.Nom) // saisie nouvelle
valeur

.....

Ecrire (Fdoc, NumE, E)

Finsi

Lire (cont)

Ftantque

Fermer (Fdoc)

Fin

RECHERCHE D'UN ENREGISTREMENT

Procédure recherche(E f: fichier de Personne, E nNom : chaîne)

Parcours séquentiel

Tant que non eof(fichier) faire
Lire(fichier,E)
si (E.nom=nNom) ...

Accès direct

Lire(fichier, num[?]E, E)

3ÈME PARTIE

CES FICHIERS SONT UNIQUEMENT DE TYPE BINAIRE

DÉFINITIONS

- Il combine la rapidité de l'accès direct et la simplicité de l'accès séquentiel.
- Implique deux structures de données : **Données + Index**.
 - **Index** ou clé d'accès = Le champ (ou les champs) par lequel les enregistrements sont référencés
 - Pour accéder à un enregistrement, il faut préciser la valeur de la clé.
- Un fichier indexé doit avoir **au moins une clé (clé primaire)**.
 - Il peut avoir d'autre clés (secondaire, tertiaire,...).
 - La clé d'accès peut être unique ou non.
- Les enregistrements sont identifiés par un index (id) ou une clé qui permet l'accès direct.

FICHER SÉQUENTIEL À ACCÈS INDEXÉ

- On associe au fichier de données un **fichier indexé**, construit à partir de certains champs des enregistrements du fichier de données. **Le fichier indexé est toujours trié.**
- L'accès aux données se fait de 2 manières :
 1. **Séquentiellement** sur le fichier de données lui-même
 2. **En mode direct** par l'intermédiaire du fichier indexé.
- Le principe :
 - Les enregistrements sont triés ou enregistrés selon l'ordre croissant de la clé primaire
 - Un index * pour chaque clé.

OUVRIR (FICHIER, MODE, ACCÈS)

- Mode d'ouverture
 - **Création** : Ouverture du fichier en création, si celui ci existe au préalable il sera réinitialisé (tous les enregistrements seront supprimés)
 - **Lecture** : Ouverture du fichier en lecture ; ce qui va permettre de transférer les enregistrements du fichier vers le programme par la procédure Lire()
 - **Écriture** : Ouverture du fichier en écriture ; ce qui va permettre de transférer les enregistrements du programme vers le fichier par la procédure écrire()
- Accès au fichier (*séquentiel, direct, alterné*)
 - alterné : simultanément direct et séquentiel.

Accès standard

Accès Direct

- Ouvrir le fichier en mode de création quel que soit le type d'accès.

Début

Ouvrir (FICH, Création)

Fermer (FICH)

Fin

Accès séquentiel

Début

Ouvrir (FICH, Création, Séquentiel)

Fermer (FICH)

Fin

Début

Ouvrir (FICH, Création, Direct)

Fermer (FICH)

Fin

Accès alterné

Début

Ouvrir (FICH, Création, Alterné)

Fermer (FICH)

Fin

OPÉRATIONS SUR LES FICHIERS INDEXÉS

- Lecture d'un enregistrement:
 - **Lire (FICHER, Enregistrement)** // Mode séquentiel
 - **Lire (FICHER, Enregistrement, Index)** // Mode Direct

- Ecriture d'un enregistrement
 - **Ecrire (FICHER, Enregistrement)** // Mode séquentiel
 - **Ecrire (FICHER, Enregistrement, Index)** // Mode Direct

OPÉRATIONS SUR LES FICHIERS INDEXÉS

- Suppression d'un enregistrement dans le fichier physique.
 - **Supprimer (FICHIER, Enregistrement)** // Mode séquentiel
 - **Supprimer (FICHIER, Enregistrement, Index)** // Mode Direct
- **Positionner (FICHIER, clé)** // mode d'accès direct sur un fichier de type séquentiel indexé, qui va permettre de positionner le curseur d'accès au fichier en fonction d'une condition.
 - Après le positionnement il faudra Lire l'enregistrement

OPÉRATIONS SUR LES FICHIERS INDEXÉS

1. Création d'un fichier
 - Définition des enregistrements.
 - Définition des clés.
2. Lecture
3. Accès à un enregistrement en fonction d'une clé et d'une valeur de clé.
4. Parcours séquentiel du fichier selon la clé primaire, éventuellement selon les autres clés.
5. Ecriture
6. Ajout d'un enregistrement.
7. Modification d'un enregistrement désigné par une clé et une valeur de clé.

CLÉ PRIMAIRE ET CLÉ ÉTRANGÈRE

■ Dans un fichier séquentiel indexé, les valeurs des clés

ELEVE

CodeEleve est
une clé primaire

PROFESSEUR

Prenom

Nom

CodeProf

Monique	Anne	Paul
EGAI	TIZE	SEBASTO
P01	P02	P03

MATIERE

CodeProf

Coef

Libellé

CodeMat

P02	P03	P01	P03
1	2	2,5	1,5
Archi	Logiciel	Analyse	Algo
MAT1	MAT2	MAT3	MAT4

CodeProf est
une clé
étrangère

CodeEleve	Nom	Prénom	Datnais
EL001	HAMME	Eric	01/02/1985
EL002	TITE	Annick	05/08/1984
EL003	HAMME	Elie	09/10/1986
EL004	CROQUE	Odile	15/02/1985
EL005	M'NAIE	Monique	05/12/1986
EL008	INTER	Paul	25/06/1985
EL009	HUILDA	Rachid	17/09/1986

EXEMPLE

une matière est enseignée par un enseignant. Les élèves qui suivent cette matière vont obtenir une moyenne dans cette matière.

PROFESSEUR

MATIERE

ELEVE

En **bleu** les clés primaires
En **rouge** les clés étrangères

Prenom	Monique	Anne	Paul
Nom	EGAI	TIZE	SEBASTO
CodeProf	P01	P02	P03

CodeProf	P02	P03	P01	P03
Coef	1	2	2,5	1,5
Libellé	Archi	Logiciel	Analyse	Algo
CodeMat	MAT1	MAT2	MAT3	MAT4

MOYENNE

CodeEleve	Nom	Prénom	Datnais
EL001	HAMME	Eric	01/02/1985
EL002	TITE	Annick	05/08/1984
EL003	HAMME	Elie	09/10/1986
EL004	CROQUE	Odile	15/02/1985
EL005	M'NAIE	Monique	05/12/1986
EL008	INTER	Paul	25/06/1985
EL009	HUILDA	Rachid	17/09/1986

	MAT1	MAT2	MAT3	MAT4
EL001	11,5	10,0	12,0	10,0
EL002	6,5	9,0	10,0	12,5
EL003	3,5	8,5	6,0	4,5
EL004	17,5	18,0	17,5	16,0
EL005	10,0	11,0	12,0	7,0
EL008	1,5	15,0	10,0	10,0
EL009	12,0	11,0	10,0	10,0

CLÉ PRIMAIRE COMPOSÉE

- Dans le fichier MOYENNE, un même élève peut avoir plusieurs moyennes mais une seule par matière. Pour une même matière on peut avoir plusieurs moyennes, mais une seule par élève.
- **CodeElev** est une clé étrangère et **CodeMat** est une clé étrangère du fichier MOYENNE
- **(CodeElev, CodeMat)** est une **clé primaire composée** du fichier MOYENNE

MOYENNE	MAT1	MAT2	MAT3	MAT4
EL001	11,5	10,0	12,0	10,0
EL002	6,5	9,0	10,0	12,5
EL003	3,5	8,5	6,0	4,5
EL004	17,5	18,0	17,5	16,0
EL005	10,0	11,0	12,0	7,0
EL008	1,5	15,0	10,0	10,0
EL009	12,0	11,0	10,0	10,0

Representation graphique effective du fichier moyenne

CLEMOY

CLEMOY

CodeEleve	CodeMat	Moy
EL001	MAT1	11,5
EL001	MAT2	10,0
EL001	MAT3	12,0
EL001	MAT4	10,0
EL002	MAT1	6,5
EL002	MAT2	9,0
EL002	MAT3	10,0
EL002	MAT4	12,0
EL003	MAT1	3,5
EL003	MAT2	8,5
EL003	MAT3	6,0
EL003	MAT4	4,5
EL004	MAT1	17,5
EL004	MAT2	18,0
EL004	MAT3	17,5
EL004	MAT4	16,0

CodeEleve	CodeMat	Moy
EL005	MAT1	10,0
EL005	MAT2	11,0
EL005	MAT3	12,0
EL005	MAT4	7,0
EL008	MAT1	1,5
EL008	MAT2	15,0
EL008	MAT3	10,0
EL008	MAT4	10,0
EL009	MAT1	12,0
EL009	MAT2	11,0
EL009	MAT3	10,0
EL009	MAT4	10,0

DÉCLARATION DES TYPES DE DONNÉES

Type TypDATE = enregistrement

Jour :entier

Mois :entier

An : entier

Fin enregistrement

TypPROF = enregistrement

CodeProf :chaîne

Nom :chaîne

Prenom :chaîne

Fin enregistrement

TypELEV = enregistrement

CodeEleve :chaîne

Nom: chaîne

Prenom: chaîne

DatNais: TypDATE

Fin enregistrement

TypMAT= enregistrement

CodeMat : chaîne

Libelle :chaîne

Coef :réel

CodeProf :chaîne

Fin enregistrement

TypMOY = enregistrement

{une clé primaire composée}

CLEMOY = enregistrement

CodeEleve :chaîne

CodeMat :chaîne

Fin enregistrement

Moy :réel

fin enregistrement

DECLARATION DES FICHIERS

Var

FICHPROF : fichier séquentiel indexé de TypPROF

clé primaire CodeProf

ENRPROF :TypProf

FICHELEV : fichier séquentiel indexé de TypELEV

clé primaire CodeEleve

ENRELEV :TypELEV

FICHMAT : fichier séquentiel indexé de TypMAT

clé primaire CodeMat

clé étrangère CodeProf de FICHPROF

ENRMAT : TypMAT

FICHMOY : fichier séquentiel indexé de TypMOY

clé primaire CLEMOY

clé étrangère CodeEleve de FICHELEV

clé étrangère CodeMat de FICHMAT

ENRMOY :TypMOY

EXERCICE

1. On veut ajouter des professeurs dans l'ordre des codes professeurs
2. On veut ajouter des professeurs dans un ordre quelconque des codes professeurs
3. On veut afficher l'ensemble des informations des enregistrements du fichier des professeurs
4. Rechercher le code d'un professeur dont on connaît le nom et le prénom
5. Rechercher le code et le nom des professeurs correspondants à un prénom donné
6. Rechercher le nom et le prénom d'un professeur dont on connaît le code.
7. Modifier le nom et le prénom d'un professeur dont on connaît le code
8. Suppression du nom et prénom d'un professeur dont on connaît le code.

9. Afficher les matières associées au nom et prénom du professeur enseignant cette matière.

EXERCICE

10. Ajouter des enregistrements dans le fichier FICHMAT sachant que la matière est enseignée par un professeur du fichier FICHPROF
11. Déterminer les matières d'un professeur dont on connaît le code, on affichera le nom et le prénom du professeur puis la liste des matières (code, libellé, coefficient).
12. Saisie de la moyenne d'un élève dont on connaît son code et le code de la matière.
13. Afficher toutes les moyennes d'un élève à partir de la saisie du code de l'élève.

EXERCICE

14. Ecrire une procédure qui retourne le code correspondant au libellé d'une matière.
15. Ecrire une fonction qui compte le nombre des professeurs enregistrés dans le fichier FICHPROF.
16. Écrire une procédure qui modifie la moyenne d'un élève
17. Écrire une fonction retourne le coef d'une matière dont on connaît le libellé.
18. Ecrire la fonction somme des coefficients des matières assurées par un professeur dont on connaît le nom et le prénom
19. Ecrire la procédure de calcul de la moyenne coefficientée d'un élève dont on fournira le nom et le prénom