

# LABORATOIRE DE BASES DE DONNEES

## "GESTION DES JEUX" : APPLICATION FICHIERS EN LANGAGE C

---

### I Description :

On considère l'application relative à la gestion des jeux d'un particulier passionné de jeux de sociétés dont l'énoncé global est donné ci-après (*point I.1*).

Le travail demandé consiste en l'écriture d'un programme en langage C répondant aux différentes impositions décrites ci-dessous. Pour la programmation, on ne tiendra compte que d'une partie du Modèle Conceptuel des Données correspondant à la solution globale.

Ce Modèle Conceptuel "réduit" est présenté au point I.2.

### I.1. Présentation générale

#### 1. Enoncé : Gestion de jeux

---

Un passionné de jeux souhaite informatiser ses acquisitions et les séances qu'il organise chez lui.

Pour chaque jeu qu'il possède, il souhaite conserver un certain nombre de caractéristiques. A savoir le nom de ce jeu, si on peut y jouer à l'intérieur et/ou à l'extérieur, quel est l'âge minimum conseillé ainsi que l'âge maximum, le nombre minimum et le nombre maximum de joueurs.

Il souhaite cependant ne pas en rester là et souhaite conserver également des renseignements à propos de l'éditeur des jeux : son nom, son adresse complète, une adresse mail et l'adresse internet correspondante.

De plus, afin de pouvoir éventuellement faire valoir une garantie sur un jeu qui présenterait des défauts, notre futur utilisateur du projet souhaite garder des données à propos du fournisseur de chaque jeu; à savoir, son nom, son adresse complète, un numéro de téléphone, une adresse électronique et un site internet. La date d'achat sera donc aussi une donnée essentielle pour savoir si un jeu est encore couvert ou non par une garantie.

Vu le grand nombre de jeux qu'il possède, il souhaite également pouvoir identifier rapidement l'armoire dans laquelle se trouve chacun de ceux-ci. Les armoires étant réparties dans plusieurs pièces, le nom de cette pièce, une description de l'armoire et un identifiant seront nécessaires.

Enfin, il faut encore conserver des informations concernant les parties de jeux qu'il organise.

Pour ce faire, il faut tout d'abord conserver des informations concernant les joueurs qui sont venus jouer au moins une fois chez lui. Les données nécessaires sont le nom, le prénom, la date de naissance, l'adresse complète, le numéro de téléphone et l'adresse mail.

Pour chaque partie de jeu, il faudra savoir le jeu auquel les joueurs ont joué, qui a participé, combien de points chaque joueur a obtenu. Il se peut que les mêmes joueurs fassent plusieurs parties du même jeu le même jour, il faut évidemment pouvoir conserver l'information à propos de chacune des parties.

Pour déterminer le nombre de points obtenus par chaque joueur, on procèdera comme suit : celui qui a gagné obtient un nombre de points égal au nombre de joueurs, le deuxième reçoit un point de moins et ainsi de suite. S'il y a deux ex-æquo, ils ont les mêmes points et le suivant a deux points de moins.

Chaque joueur obtient donc toujours au moins un point pour chaque partie qu'il joue.

Exemple :

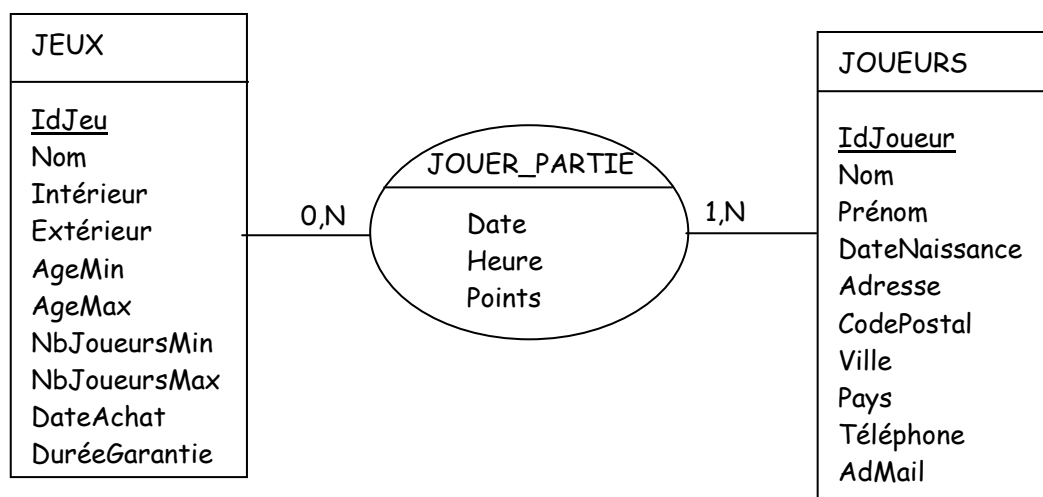
Jean, Luc, Pierre et Philippe jouent une partie de Monopoly de Liège. Philippe est ruiné le premier et sera donc le dernier de la partie. Ensuite, pendant le même tour, Luc et Jean perdent tous leurs avoirs. C'est donc Pierre qui gagne.

Ils obtiendront donc les points suivants pour cette partie :

Pierre	4
Jean	3
Luc	3
Philippe	1

Par simplification, on supposera que le commanditaire de l'application ne possède qu'un seul exemplaire de chaque jeu (un seul Monopoly classique, un seul Monopoly junior, un seul scrabble ...).

## I.2. Solution réduite (modèle conceptuel des données) à considérer



## II Définition des structures à utiliser :

(certains champs nécessaires à la mise en œuvre de l'organisation de fichier demandée peuvent bien sûr être ajoutés)

- **Table Jeux**

```
struct Jeux {  
    long      IdJeu ;  
    char      Nom [25] ;  
    char      Intérieur ;  
    char      Extérieur ;  
    short     AgeMin;  
    short     AgeMax;  
    short     NbJoueursMin;  
    short     NbJoueursMax;  
    Date      DateAchat;  
    short     DuréeGarantie;  
};
```

- **Table Joueurs**

```
struct Joueurs {  
    long      IdJoueur ;  
    char      Nom [25] ;  
    char      Prenom [25] ;  
    Date      DateNaissance;  
    char      Adresse [50] ;  
    char      CodePostal[5];  
    char      Ville[25];  
    char      Pays [25];  
    char      Telephone [13];  
    char      AdMail [65];  
};
```

- **Table Jouer Partie**

```
struct Jouer_Partie {  
    long      IdJoueur ;  
    long      IdJeu;  
    Date      DatePartie;  
    Heure     HeurePartie;  
    short     Points;  
};
```

Dans cette structure, Date et Heure sont deux structures composées respectivement de 3 et 2 éléments de type short : jour, mois, année pour la date et heure, minute pour l'heure

### III Organisation physique des fichiers :

Le programme comporte trois fichiers (Jeux, Joueurs, PartiesJouees). Aucun de ces fichiers ne pourra être chargé entièrement en mémoire pendant l'exécution du programme.

Veillez à programmer les différentes organisations suivant la théorie vue au cours d'"Organisation et Exploitation des Données" de Mme Serrhini.

1. Joueurs :

**Organisation indexée directe (full indexed)** : l'index sera créé au début de l'exécution du programme. Cet index devra contenir au minimum le numéro identifiant le joueur ainsi que son nom.

2. Jeux :

**Organisation indexée direct (full indexed)** : l'index sera créé au début de l'exécution du programme.

3. PartiesJouees :

**Organisation séquentielle physique**

### IV Fonctionnalités du programme :

- *Création des fichiers (par bidonnage des fichiers Jeux et Joueurs).*

On admettra les tailles suivantes pour chaque fichier :

Nom du fichier	Nombre maximum d'enregistrements
Jeux	50
Joueurs	30

La taille du fichier *PartiesJouees* dépendra du nombre d'enregistrements qu'il contient.

- *Dans le fichier Joueurs :*

- Ajout d'un nouveau joueur (mise à jour du fichier et de l'index).
- Modification des données d'un joueur (tous les attributs sont susceptibles de modification, mais ne sont pas forcément tous changés).
- suppression "logique" d'un joueur; la suppression d'un joueur pour lequel des parties ont été enregistrées ne pourra se faire qu'en tenant compte de l'intégrité référentielle. L'étudiant aura le choix de supprimer le joueur ainsi que toutes les parties qu'il a jouées ou de refuser cette suppression.
- Listage des joueurs classés par ordre alphabétique croissant sur leur nom.

- *Dans le fichier Jeux :*

- Ajout d'un nouveau jeu,
- Affichage, au choix, de tous les jeux (l'affichage se fera par ordre alphabétique sur le nom) ou uniquement des jeux qui sont toujours sous garantie.
- **Bonus** : suppression d'un jeu que l'on recherchera sur base de son nom.

- *Dans le fichier PartiesJouees :*
  - Ajout de joueurs ayant participé à une partie. Choisir un jeu et encoder tous les joueurs ayant participé à la partie correspondante. Un même joueur peut jouer plusieurs fois au même jeu, pour autant, évidemment, que les parties ne débutent pas le même jour à la même heure.
  - Affichage séquentiel du fichier.
- Un menu général des diverses fonctionnalités sera affiché afin que l'utilisateur puisse faire son choix d'action et lui offrira aussi la possibilité de quitter l'application.

## V Contraintes à respecter :

Lors de la conception du programme, l'étudiant veillera à faire respecter par l'application les trois contraintes d'intégrité :

- L'intégrité d'entité,
- L'intégrité de référence,
- L'intégrité de domaine : on distinguera ici :

### ***Contraintes obligatoires :***

- Toutes les dates apparaissant dans les différentes structures sont des champs date encodés sous la forme JJ/MM/AAAA. A l'étudiant d'évaluer si chacune des dates est obligatoire ou non en se basant sur l'énoncé et sur le rôle de chaque champ date. Il faut également décider si ces dates peuvent ou doivent être inférieures ou supérieures à la date du jour, à nouveau en se référant à l'énoncé.
- Les champs intérieur et extérieur de la table JEUX sont des booléens (OUI/NON). Ces champs sont obligatoires.
- Les champs AgeMin et AgeMax sont des champs obligatoires positifs ou nuls (pour les jeux adaptés aux enfants de moins d'un an) pour lesquels on vérifiera que AgeMin est inférieur à AgeMax.
- Les champs NbJoueursMin et NbJoueursMax seront également obligatoires, strictement positifs et on vérifiera aussi que NbJoueursMin est inférieur ou égal à NbJoueursMax.
- Le champ DuréeGarantie est lui aussi un champ obligatoire qui doit avoir une valeur positive.
- Le code postal est un champ obligatoire de 4 chiffres.
- Le champ AdMail est un champ facultatif, mais lorsque celui-ci reçoit une valeur, celle-ci doit être une adresse valide !

### ***Contraintes diverses :***

D'autres contraintes peuvent être envisagées et sont laissées à l'appréciation de l'étudiant. Il en sera tenu compte dans la note d'évaluation finale de l'application.

## VI Consignes du travail :

- Travail individuel, toute ressemblance (programme ou procédure avec le travail d'un condisciple) donnera lieu à une note de 0.
- L'étudiant veillera à fournir sur ses supports électroniques des fichiers tests dans lesquels il aura inséré, les données fournies en annexe.
- La cotation tiendra compte de la présentation (en ce compris l'ORTHOGRAPHE) et de la convivialité : le programme doit ainsi être conçu de manière "professionnelle" et donc rendu accessible à tout utilisateur,
- La bonne mise en pratique de l'organisation des fichiers imposée ainsi que l'exécution correcte de toutes les fonctionnalités décrites plus haut font partie des critères d'évaluation. Veuillez revoir les organisations vues au cours d'organisation et exploitation des données (Informatique de gestion) ou au cours de systèmes de gestion de bases de données (informatique et systèmes).
- Le dossier doit être rendu suivant les consignes données par votre titulaire de laboratoire et comporter au minimum :
  - Une page de garde avec Nom, prénom, groupe, année d'étude, intitulé du cours et le nom du professeur titulaire
  - Cet énoncé,
  - Le code source du programme imprimé
  - 2 supports électroniques (l'un servant de backup) contenant le code source et des fichiers tests décrits dans les contraintes obligatoires. Veillez à utiliser des supports compatibles avec les machines du laboratoire de l'école.
- Le dossier complet doit être remis en main propre à votre titulaire le Mercredi 8 Mai 2019 avant 15h.
- Tout dossier incomplet, par exemple non accompagné des fichiers sur 2 supports électroniques, comme précisé ci-dessus, recevra la cote de 0 sur 20.
- Ce dossier fait l'objet d'une évaluation orale, tout étudiant qui remet le dossier mais est absent pour la défense orale, recevra la cote de 0 sur 20.
- **ATTENTION** : vous serez évalués sur un **PC des laboratoires** et non sur votre portable. Le compilateur utilisé sera donc CodeBlocks tel qu'utilisé dans les laboratoires depuis le début de l'année scolaire.

- **Annexe**

Jeux

IdJeu	Nom	Intérieur	Extérieur	AgeMin	AgeMax	NbJoueursMin	NbJoueursMax	DateAchat	DuréeGarantie
1201	Monopoly Electronique	Oui	Non	8	99	2	6	10/12/2008	2
1301	Scrabble	Oui	Oui	10	99	2	4	05/07/1995	1
1505	Trivial poursuite	Oui	Oui	10	99	2	6	01/12/2002	1
1302	Scrabble Junior	Oui	Oui	5	10	2	4	04/04/1990	1
1100	Le Verger	Oui	Oui	3	12	2	8	12/12/1990	1
1402	Petanque	Non	Oui	6	99	2	8	10/07/2005	0
1508	Trivial poursuite Junior	Oui	Oui	8	12	2	6	15/12/2005	1
1408	Cricket	Non	Oui	6	99	2	6	15/07/2000	0
1214	Monopoly de Liege	oui	Non	8	99	2	8	31/08/1995	1

Joueurs

IdJoueur	Nom	Prénom	DateNaissance	Adresse	CodeP ostal	Ville	Pays	Téléphone	AdMail
110	Dubrun	Luc	12/04/1997	Rue de Fétine, 2	4000	Liège	Belgique	04/312.14.14	<a href="mailto:ldubrun@skynet.be">ldubrun@skynet.be</a>
120	Joskin	Jeanine	24/11/1973	Rue de Cerisiers, 26	4680	Oupeye	Belgique	04/323.25.52	<a href="mailto:jjoskin6230@belgacom.net">jjoskin6230@belgacom.net</a>
150	Duchateau	Franck	12/06/1980	Voie Tombe, 17	4537	Verlaine	Belgique	04/259.23.98	<a href="mailto:franck.duchateau@hotmail.com">franck.duchateau@hotmail.com</a>
130	Baltus	Pierre	13/03/2006	Rue Sartage, 3	4540	Ampsin	Belgique	085/31.39.58	<a href="mailto:pb@baltus.be">pb@baltus.be</a>
170	Verstraeten	Alexandre	25/10/1983	Quai de Rome, 15	4000	Liège	Belgique	04/331.45.47	<a href="mailto:alex.verstr@gmail.com">alex.verstr@gmail.com</a>
180	Deuster	Marie	31/07/1998	Av. H. Monjoie, 25	4300	Waremmme	Belgique	019/33.99.55	<a href="mailto:marie.d@skynet.be">marie.d@skynet.be</a>
140	Mirguet	Léon	16/12/1999	Sur le Bois 65	4601	Argenteau	Belgique	04/374.51.59	<a href="mailto:leon.mirguet@provincedeliege.be">leon.mirguet@provincedeliege.be</a>
160	Palmans	Solange	24/08/1994	Rue Mont Falise, 115	4500	Huy	Belgique	085/21.23.77	<a href="mailto:solpal@hotmail.com">solpal@hotmail.com</a>



PartiesJouees

IdJeu	IdJoueur	DatePartie	HeurePartie	Points
1408	150	01/05/2009	14:00	1
1408	140	01/05/2009	14:00	3
1408	120	01/05/2009	14:00	2
1100	140	05/05/2009	16:00	2
1100	130	05/05/2009	16:00	2
1301	180	01/05/2009	15:30	3
1301	150	01/05/2009	15:30	2
1301	170	01/05/2009	15:30	1
1505	160	03/05/2009	20:00	1
1505	110	03/05/2009	20:00	2