



Université
de Lomé

Ecole
Polytechnique de
Lomé

CAMPING -CERISAIE

INFI329-Projet Tuteuré I semestre 3

MEMBRES DU GROUPE

PISSANG Yindoutie Seth

ABBEY Anaté Victorien

PROF CHARGÉE DU COURS

Mr AKAKPO

SOMMAIRE

1. INTRODUCTION

2. ANALYSE DU BESOIN

3. DICTIONNAIRE DE DONNEES

4. ELABORATION DU MCD

5. ELABORATION DU MLD

6. FONCTIONNALITES ET STRUCTURES

7. CONCLUSION

1. INTRODUCTION

Chaque saison touristique représente un véritable défi organisationnel pour le camping La Cerisaie. Entre la gestion des emplacements, le suivi des séjours et la facturation des services annexes, une administration efficace devient indispensable.

Ce projet vise à concevoir un système de gestion structuré et fiable permettant d'automatiser le suivi des séjours des campeurs, l'attribution des emplacements et la prise en compte des activités sportives louées.

À travers cette étude, nous cherchons à proposer une solution informatisée capable d'améliorer la précision des données, de simplifier la facturation et d'optimiser la gestion globale du camping.

2. ANALYSE DU BESOIN

D'après l'énoncé, le camping *La Cerisaie* doit gérer :

- Des **Clients** (Vacanciers Ou Groupes, Identifiés Par Un Responsable),
- Des **Emplacements** De Différents Types (Tentes, Caravanes, Camping-Cars, Bungalows),
- Des **Séjours** Dont Il Faut Enregistrer L'arrivée Et Le Départ, • Des **Locations D'équipements Sportifs**,
- La **Facture** à éditer au départ.

La gestion porte donc sur :

- La réservation/affectation d'un emplacement,
- La tarification de cet emplacement selon son type, • Les activités sportives pratiquées durant le séjour, • Le calcul de la facture finale.

Le système doit centraliser toutes ces informations pour faciliter le travail de la direction.

Travail demandé : il nous est demandé de mettre en place à l'aide de nos compétences en génie logiciel une solution digitale pour gérer toutes ces données.

3. DICTIONNAIRE DE DONNES

| N° | Nom de l'attribut | Type | Description |
|----|-------------------|--------|------------------------------|
| 1 | idClient | Entier | Identifiant unique du client |
| N° | Nom de l'attribut | Type | Description |
| 2 | Nom | Chaîne | Nom du client |
| 3 | Prenoms | Chaîne | Prénoms du client |

| | | | |
|----|-------------------|--------|------------------------------------------|
| 4 | Adresse | Chaîne | Adresse du client |
| 5 | idSejour | Entier | Identifiant unique du séjour |
| 6 | DateDebut | Date | Date de début du séjour |
| 7 | DateFin | Date | Date de fin du séjour |
| 8 | NbrePersonne | Entier | Nombre de personnes séjournant |
| 9 | idClient | Entier | Référence du client effectuant le séjour |
| 10 | idEmplacement | Entier | Emplacement attribué au séjour |
| 11 | idEmplacement | Entier | Identifiant unique de l'emplacement |
| 12 | Surface | Réel | Surface de l'emplacement |
| 13 | MaxPersonne | Entier | Nombre maximal de personnes |
| 14 | idTypeEmplacement | Entier | Type auquel appartient l'emplacement |
| 15 | idTypeEmplacement | Entier | Identifiant unique du type d'emplacement |
| 16 | Libelle | Chaîne | Libellé du type d'emplacement |
| 17 | | | |

4. ELABORATION DE LA METHODE CONCEPTUELLE DE DONNEES

Le modèle conceptuel de données du projet La Cerisaie illustre les entités principales et leurs relations. Les cardinalités associées à chaque relation permettent de comprendre la logique de gestion : combien d'éléments d'une entité peuvent être liés à un élément d'une autre. Elles traduisent la réalité du camping en règles précises pour la base de données et, plus tard, pour l'implémentation en langage C.

CLIENT – SEJOUR (Effectuer : 1,n ↔ 1,1)

Un client peut effectuer plusieurs séjours, mais chaque séjour est lié à un seul client. Cela reflète la réalité où un vacancier peut revenir plusieurs fois au camping.

SEJOUR – EMPLACEMENT (Occupe : 1,1 ↔ 0,n)

Chaque séjour occupe un seul emplacement, mais un emplacement peut être occupé par plusieurs séjours successifs (dans le temps). Cela traduit la rotation des vacanciers sur un même emplacement.

EMPLACEMENT – TYPE_EMPLACEMENT (Appartient : 1,1 ↔ 1,n)

Chaque emplacement appartient à un seul type, mais un type peut regrouper plusieurs emplacements. Par exemple, plusieurs emplacements peuvent être des tentes ou des bungalows.

SEJOUR – LOCATION_SPORT (Donne_lieu_à : 1,n ↔ 1,1)

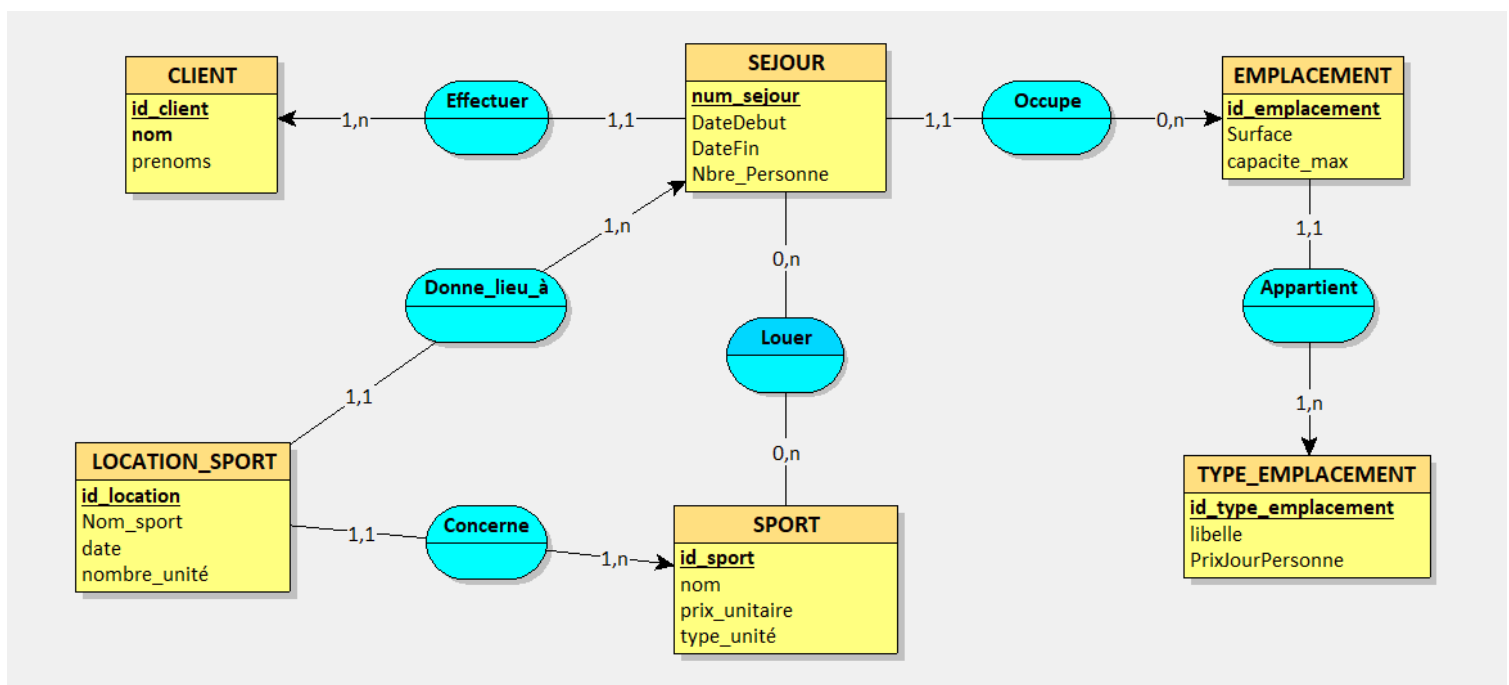
Un séjour peut donner lieu à plusieurs locations sportives, mais chaque location est rattachée à un seul séjour. Cela permet de suivre les activités pratiquées par les vacanciers durant leur séjour.

LOCATION_SPORT – SPORT (Concerne : 1,1 ↔ 1,n)

Chaque location concerne un seul sport, mais un sport peut être loué dans plusieurs locations. Par exemple, le tennis peut apparaître dans de nombreuses locations sportives.

SEJOUR – SPORT (Louer : 0,n ↔ 0,n)

Un séjour peut être associé à plusieurs sports, et un sport peut être pratiqué dans plusieurs séjours. Cette relation traduit la flexibilité des activités offertes aux vacanciers.



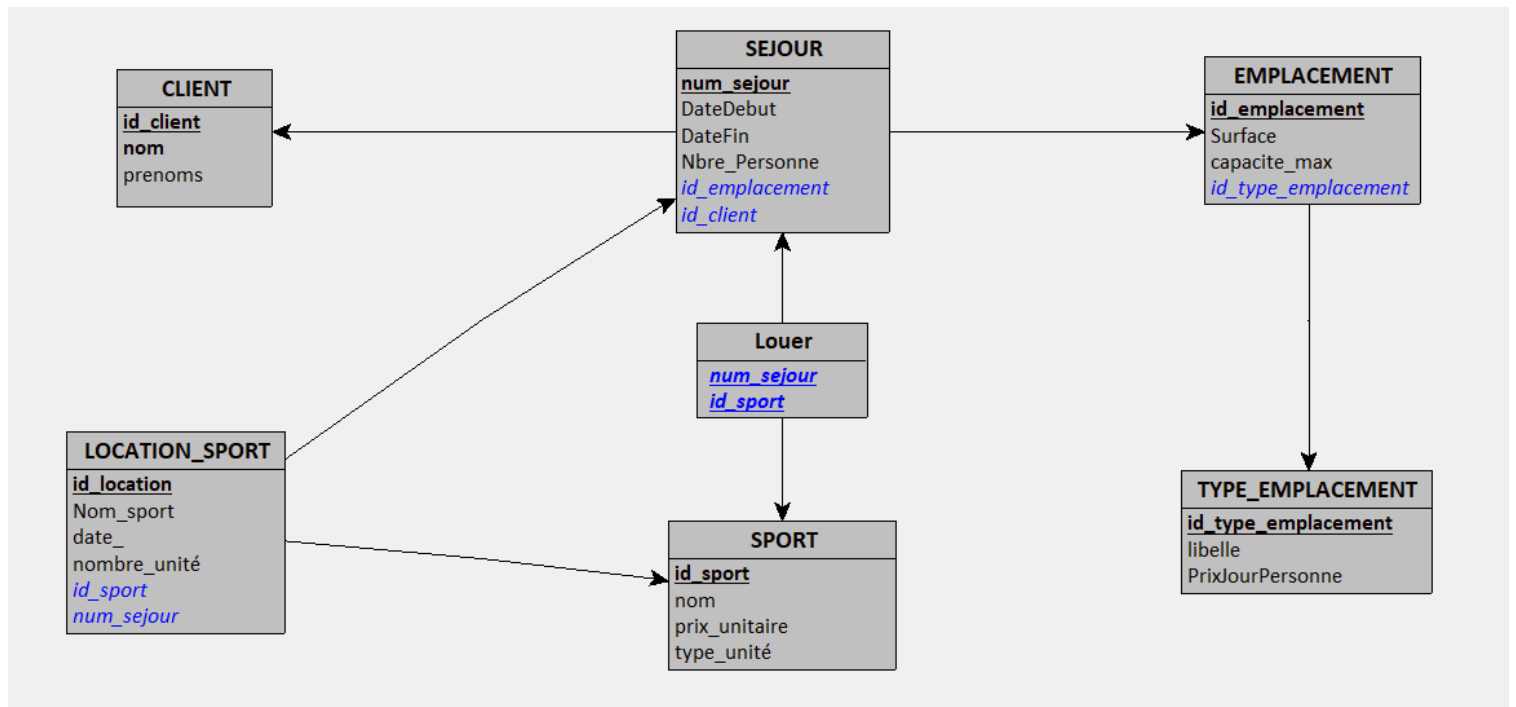
5. ELABORATION DU MODELE LOGIQUE DE DONNEES

Le MLD est la traduction opérationnelle du MCD. Il exprime les entités sous forme de tables relationnelles, avec leurs clés primaires et étrangères, et précise les liens entre elles. Les cardinalités du MCD se matérialisent ici par des relations de type 1-N ou N-N, gérées par des clés étrangères ou des tables d'association.

Commentaire du MLD

- **CLIENT (id_client, nom, prenom)** La table CLIENT conserve les informations personnelles des vacanciers. La clé primaire est id_client. → Relation : chaque client peut être lié à plusieurs séjours (1-N).
- **SEJOUR (num_sejour, DateDebut, DateFin, Nbre_Personne, id_emplacement, id_client)** La table SEJOUR enregistre les séjours effectués. La clé primaire est num_sejour. → id_client est une clé étrangère vers CLIENT. → id_emplacement est une clé étrangère vers EMPLACEMENT. → Chaque séjour est associé à un seul client et un seul emplacement, mais un client peut avoir plusieurs séjours et un emplacement peut être occupé par plusieurs séjours successifs.
- **EMPLACEMENT (id_emplacement, Surface, capacite_max, id_type_emplacement)** La table EMPLACEMENT décrit les caractéristiques physiques des emplacements. La clé primaire est id_emplacement. → id_type_emplacement est une clé étrangère vers TYPE_EMPLACEMENT. → Chaque emplacement appartient à un seul type, mais un type peut regrouper plusieurs emplacements (relation 1-N).
- **TYPE_EMPLACEMENT (id_type_emplacement, libelle, PrixJourPersonne)** La table TYPE_EMPLACEMENT définit les catégories tarifaires. La clé primaire est id_type_emplacement. → Relation : un type peut concerner plusieurs emplacements.
- **SPORT (id_sport, nom, prix_unitaire, type_unité)** La table SPORT recense les activités sportives proposées. La clé primaire est id_sport. → Relation : un sport peut être loué dans plusieurs séjours.
- **LOCATION_SPORT (id_location, Nom_sport, date_, nombre_unité, id_sport, num_sejour)** La table LOCATION_SPORT enregistre les locations d'activités sportives. La clé primaire est id_location. → id_sport est une clé étrangère vers SPORT. → num_sejour est une clé étrangère vers SEJOUR. → Chaque location est liée à un seul séjour et à un seul sport, mais un séjour peut comporter plusieurs locations et un sport peut être loué plusieurs fois.

- **LOUER (num_sejour, id_sport)** Cette table d'association gère la relation N-N entre SEJOUR et SPORT. → La clé primaire est composée de (num_sejour, id_sport). → Elle traduit le fait qu'un séjour peut inclure plusieurs sports et qu'un sport peut être pratiqué dans plusieurs séjours.



6. Fonctionnalités nécessaires à la gestion des emplacements

■ Structures fondamentales

Dans le cadre du projet *La Cerisaie*, la gestion des emplacements constitue une étape cruciale. En effet, chaque vacancier ou groupe de vacanciers doit être associé à un emplacement précis, lequel appartient à une catégorie tarifaire définie. Pour traduire cette réalité dans un programme informatique, il est indispensable de définir des **structures fondamentales** en langage C. Ces structures permettent de représenter les entités du modèle conceptuel de données (en lien avec la gestion des emplacements) directement en mémoire, de les manipuler efficacement et de garantir la cohérence avec les fichiers binaires utilisés pour la persistance.

Les structures retenues sont :

```

typedef struct {
    int id_type;
    char libelle[30]; // Exemple : "tente", "caravane", "bungalow"
    float prix_jour_par_personne;
} TypeEmplacement;
  
```

i TypeEmplacement

La structure TypeEmplacement représente la catégorie tarifaire d'un emplacement et permet de stocker le libellé (nom du type) et le prix par jour et par personne.

Charger en mémoire la liste des types disponibles au démarrage du programme va permettre la **recherche**, l'**ajout**, la **modification**, ou la **suppression de types**. Elle servira aussi de référence pour les emplacements concrets (via `id_type`).

ii Emplacement

```
typedef struct {
    int id_emplacement;
    float surface;
    int capacite_max;
    int id_type; // lien vers TypeEmplacement
} Emplacement;
```

La structure `Emplacement` représente un emplacement concret dans le camping. Elle contient ses caractéristiques physiques (surface, capacité maximale) et est reliée à un type d'emplacement pour déterminer sa catégorie tarifaire.

Elle permettra de :

- ☐ Permettre la recherche et l'affichage des emplacements disponibles.
- ☐ Stocker et manipuler les données dans un fichier binaire (`emplacements.dat`).
- ☐ Gérer l'attribution des emplacements aux vacanciers lors des séjours.

b Fonctionnalités nécessaires à la gestion des emplacements

Une fois les structures fondamentales définies, il est indispensable de préciser les **fonctionnalités logicielles** qui permettront de manipuler ces données. Ces fonctionnalités traduisent les besoins réels de la gestion des emplacements dans le camping *La Cerisaie* et assurent la cohérence entre la modélisation conceptuelle et l'implémentation en langage C.

Fonctionnalités liées à `TypeEmplacement`

Cette structure représente les catégories tarifaires des emplacements. Les fonctionnalités associées doivent permettre une gestion complète de ces types :

1. **Chargement initial** : lire le fichier binaire `types_emplacements.dat` et charger tous les types dans un tableau en mémoire.
2. **Ajout d'un type** : insérer un nouveau type avec son libellé et son prix.
3. **Modification d'un type** : mettre à jour le prix ou le libellé d'un type existant.
4. **Suppression d'un type** : retirer un type du tableau et du fichier binaire.
5. **Recherche** : retrouver un type par son identifiant ou par son libellé.
6. **Affichage** : lister tous les types disponibles avec leurs caractéristiques.
7. **Déchargement à la fermeture** : sauvegarder le tableau en mémoire vers le fichier binaire avant de quitter le programme.

Fonctionnalités liées à `Emplacement`

Cette structure représente les emplacements concrets attribués aux vacanciers. Les fonctionnalités doivent permettre de gérer leur cycle de vie :

1. **Chargement initial** : lire le fichier binaire emplacements.dat et charger tous les emplacements en mémoire.
2. **Ajout d'un emplacement** : créer un nouvel emplacement avec sa surface, sa capacité et son type associé.
3. **Modification d'un emplacement** : mettre à jour les caractéristiques (surface, capacité, type).
4. **Suppression d'un emplacement** : retirer un emplacement du tableau et du fichier binaire.
5. **Recherche** : retrouver un emplacement par son identifiant ou par type.
6. **Affichage** : lister tous les emplacements disponibles, avec leurs caractéristiques et leur type.
7. **Déchargement à la fermeture** : sauvegarder les emplacements en mémoire vers le fichier binaire avant de quitter.

7. CONCLUSION

La modélisation du projet La Cerisaie a permis d'identifier les entités clés et leurs relations, traduites dans le MCD puis dans le MLD. Les cardinalités assurent une représentation fidèle de la réalité du camping, notamment la gestion des clients, des séjours, des emplacements et des activités sportives.

La définition des structures fondamentales en C, ainsi que la liste des fonctionnalités associées, établit une base claire pour l'implémentation. Cette démarche méthodologique garantit une solution cohérente et adaptée aux besoins du projet.