

04/12/2021  
INFOO304

# Gestion d'un Camping



---

*MTARFI Souhail*  
*ALLICIO mikael*  
*Dirigé par Mme*  
*BEAUJET Beatrice*

## **Sommaire**

Introduction.....	3
Analyse.....	4
Cahier des charges.....	4
Matrice de flu.....	6
Dictionnaires de donnés.....	7
Modèle conceptuel de données.....	13
Modèle logique de donnés.....	14
Modèle conceptuel de traitement.....	15
Modèle relationnelle.....	18
Normalisation.....	19
REQUETE SQL.....	20
CONCLUSION .....	24
ANNEX.....	25
Script de la base de donnees.....	25

## **Introduction**

*Dans le cadre de la gestion d'un camping à venir, nous avons été contactés pour créer une base de données pouvant stocker de nombreuses informations et les partager avec la communauté.*

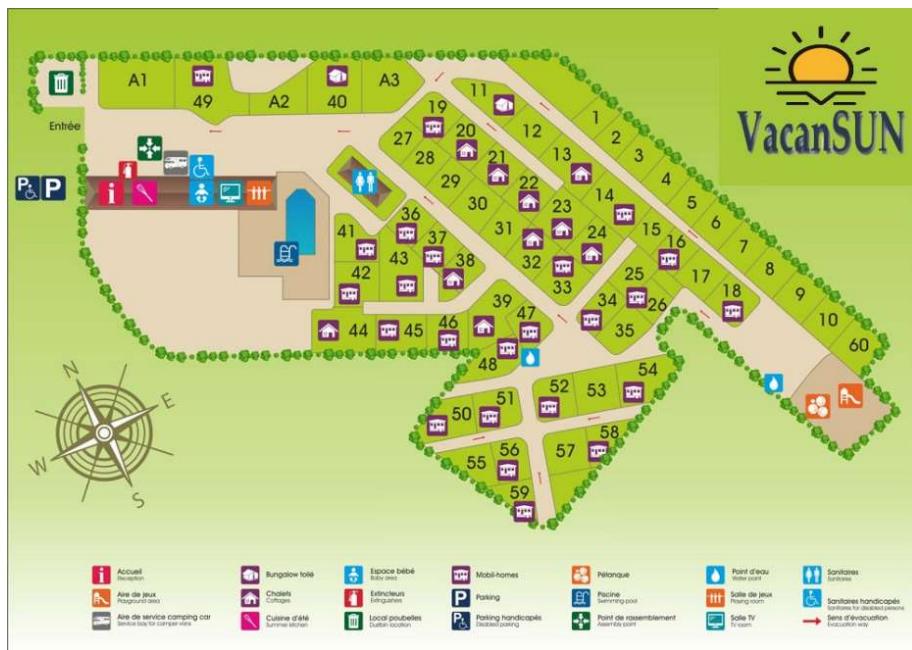
*De nos jours, de nombreuses industries utilisent ce système de base de données pour faciliter la gestion générale de leur entreprise.*

*En particulier, la complexité des campings et les nombreux métiers liés aux loisirs de plein air rendent l'utilisation des bases de données très utile pour façonner de tels réseaux et comprendre leur fonctionnement. Avant de créer la base de données, il a fallu observer le camp pendant longtemps. Nous avons dû aussi observer tous les employés et vérifier la structure du domaine.*



## ANALYSE

### Cahier des charges



- capacité nb emplacement : 60
- période d'ouverture : hiver (décembre à mars) printemps (avril à juin) / été (juillet à septembre)
- Nous devons donc pour le cas du camping VacanSun créer une base de données afin de gérer des plannings concernant des emplacements ,des clients ,des activités et des employés dans un système qui stocke des informations .De nos jours beaucoup de secteurs utilisent ce

système de base de données pour faciliter la gestion générale de leurs entreprises.

pour créer notre base de données nous avons dû relever tous les animateurs(employés) ainsi qu'examiner la structure de l'établissement .

Ce camping est composé par un bâtiment ou réside les employés , un local d'accueil ,une salle d'activité ,une aire de jeux pour enfants en plein air ,un terrain de pétanque , 2 Bâtiments sanitaires ,des espaces réservé au caravanes , 2 toilettes publics , 5 bacs à laver la vaisselle, 1 salle avec machine à laver le linge , Une aire de rangement pour décharger les camions, des espaces réservés au tentes ainsi que des mobile-home , des pavillons et chalets à louer. Il y a aussi une grande zone d'activité en plein air (sport)ainsi qu'un lac. Le camping fait 1,3 hectares.

TARIF Saison	
Emplacement <b>tentes</b> (1 personne)	<b>575 €</b>
Emplacement <b>caravane</b> (2 personne)	<b>988 €</b>
Emplacement <b>mobil-home</b> (3 personnes )	<b>1 133 €</b>
Chalets ( 4 personnes )	<b>1 545 €</b>
Personne supplémentaire <b>de 4 à 12 ans, forfait saison</b>	<b>50 €</b>
Personne supplémentaire <b>de 13 à 18 ans, forfait saison</b>	<b>75 €</b>
Personne supplémentaire <b>de + de 18ans, forfait saison</b>	<b>150 €</b>
Personne supplémentaire à partir de 4 ans, <b>à la nuitée</b>	<b>4,50 €</b>
Ordures ménagères, extra ménagères	<b>40 €</b>
Taxe de séjour par personne de <b>+ de 18 ans</b> et par jour	<b>0,20 €</b>
Forfait électricité	<b>225 €</b>
<b>Jeton de douche</b>	<b>50,00 €</b>
<b>Machine à laver</b>	<b>4,00 €</b>

TARIF journalier	
<b>Adulte</b>	<b>4,50 €</b>
<b>Enfant (4 à 13 ans)</b>	<b>3,50 €</b>
<b>Emplacement</b>	<b>6,00 €</b>
<b>Electricité</b>	<b>4,00 €</b>
<b>Taxe Ordures Ménagères</b>	<b>0,50 €</b>
<b>Taxe de séjour + de 18 ans (par jour, par personne)</b>	<b>0,20 €</b>
<b>Jeton de douche</b>	<b>1,00 €</b>
<b>Machine à laver</b>	<b>4,00 €</b>

## Matrice de flux

Une matrice de flux nous permet de structurer les échanges entre nos différents acteurs.  
Notre domaine d'étude étant un cabinet médical.

DE->VERS	Responsable du camping	Employe du camping	Client	Tente,cavarne...	Des équipements offerts	Branchement électrique	Connexion Internet	Seance	Activités	Animateur	Diplôme	Reservation	accompagnant
<b>Responsable du camping</b>		donne des ordres	accueillir	monter et apporter	gerer								
<b>Employe du camping</b>	obeir aux ordres		aider		apporte	installer			proposer				
<b>Client</b>	etre accueilli	etre aider par		dormir	utiliser	utiliser	utiliser	participer		suivre les conseils		acheter	accompagner
<b>Tente,cavarne...</b>		etre montée par	etre utilisée par			accueille							
<b>Des équipements offerts</b>	etre gerer	etre apporter	etre utilisée par										
<b>Branchement électrique</b>		etre utilisée par	etre utilisée par	existe									
<b>Connexion Internet</b>			etre utilisée par										
<b>Seance</b>			divertir							dont lequel est pratiquer	etre gerer		
<b>Activités</b>									Etre pratiquer				divertir
<b>Animateur</b>									encadrer				avoir
<b>Diplôme</b>											etre obtenu		
<b>Reservation</b>			acheter par										
<b>accompagnant</b>			etre accompagnier						participer				

## Dictionnaire de données

### Les entités

- Client :

Attribut	Definition attribut	Type/Taille	Exemple de valeur
<b>Id_Client</b>	Identifiant du Client	INT	1
NomClient	Nom du Client	VARCHAR(50)	Albert
PrenomClient	Prenom du Client	VARCHAR(50)	Mark
DateNaissanceClient	Date de naissance du Client	DATE	15/01/1987
AdresseClient	Dernier adresse connu du Client	VARCHAR(50)	9 rue de vesle,Reims
EmailClient	Adresse Email de Client	VARCHAR(50)	Albert.Mark@gmail.com
NumTelClient	Numéro de téléphone de Client	INT	0685214212

Responsable :

Attribut	Definition attribut	Type/Taille	Exemple de valeur
<b>Id_Responsable</b>	Identifiant du Responsable	INT	1
NomResponsable	Nom de Responsable	VARCHAR(50)	Morceau
PrenomResponsable	Prenom de Responsable	VARCHAR(50)	Christophe
EmailResponsable	Adresse Email de Responsable	VARCHAR(50)	<a href="mailto:Morceau.1962@gmail.com">Morceau.1962@gmail.com</a>
NumTelResponsable	Numéro de téléphone de Responsable	INT	0662110101

- Employé :

Attribut	Définition attribut	Type/Taille	Exemple de valeur
<b>Id_Employe</b>	Identifiant de l'employé	INT	8

NomEmploye	Nom de l'employé	VARCHAR(50)	Silva
PrenomEmploye	Prénom de l'employé	VARCHAR(50)	David

- accompagnant :

Attribut	Définition attribut	Type/Taille	Exemple de valeur
<b>Id_accompagnant</b>	Identifiant de la mutuelle	INT	11
NomAcc	Nom de l'employé	VARCHAR(50)	lepen
PrenomAcc	Prénom de l'employé	VARCHAR(50)	Mary
NumTelAcc	Précise les « offres »	INT	0741526879

- EquipementsOfferts:

Attribut	Définition attribut	Type/Taille	Exemple de valeur
<b>Id_EquipementsOfferts</b>	Identifiant de l'équipement	INT	54
NomEquipements	Nom de l'équipement	VARCHAR(50)	Bélier

- Reservation

Attribut	Définition attribut	Type/Taille	Exemple de valeur
<b>Id_Reservation</b>	Identifiant de la resevation	INT	544
DateArrivee	La Date quand le client va arriver au camping	Date-Heure	12/05/2022 10 :00
DateDepart	La Date quand le client va partir du camping	Date-Heure	19/05/2022 10 :00
<b>PrixReservation</b>	Le prix de la reservation	DOUBLE	400€

- Animateur:

Attribut	Définition attribut	Type/Taille	Exemple de valeur
<b>Id_Animateur</b>	Identifiant du animateur	INT	5
NomAnimateur	Nom de l'animateur	VARCHAR(50)	Dupont
PrenomAnimateur	Prénom de l'animateur	VARCHAR(50)	Luis

- Emplacement:

Attribut	Définition attribut	Type/Taille	Exemple de valeur
<b>Id_Hebergement</b>	Identifiant de l'hebergement	INT	25
Taille	Taille de l'hebergement	VARCHAR(50)	3 m <sup>2</sup>
NumEmplacement	Numero de l hebergement	INT	12

- Branchement électrique:

Attribut	Définition attribut	Type/Taille	Exemple de valeur

<b>Id_Branchement électrique</b>	Identifiant de Branchement électrique	INT	25
----------------------------------	---------------------------------------	-----	----

- Diplome:

Attribut	Définition attribut	Type/Taille	Exemple de valeur
<b>Id_Diplome</b>	Identifiant de diplome	INT	5161
DateObtention	Date de l'obtention de diplome	VARCHAR(50)	10/07/2012

- Acitive:

Attribut	Définition attribut	Type/Taille	Exemple de valeur
<b>Id_Activite</b>	Identifiant de l'activite	INT	4
TypeActivite	Type de l'activité	VARCHAR(50)	Match de Foot

- Seance:

Attribut	Définition attribut	Type/Taille	Exemple de valeur
<b>Id_Seance</b>	Identifiant de la Seance	INT	4
DateSeance	La date de la seance	DATE-Heure	15/05/2022 08 :00
PrixSeance	Le prix pour participer à la seance	DOUBLE	8.25€

- ConnexionInternet:

Attribut	Définition attribut	Type/Taille	Exemple de valeur
<b>Id_connexionInternet</b>	Identifiant de l'hébergement	INT	56
Debit	Debit de la connexion	INT	25

## Les relations

- Donner des ordres :

Attributs	Etre la relation entre
-	Responsable et Employe

- apporter:

Attributs	Etre la relation entre
-	Employe et EquipementsOfferts

- aider:

Attributs	Etre la relation entre
-	Employe et Client

- installer:

Attributs	Etre la relation entre
-	Employe et Branchement électrique

- Faire l'entretien:

Attributs	Etre la relation entre
-	Employe et hébergement

- accompagner:

Attributs	Etre la relation entre
-	Client et accompagnant

- participer:

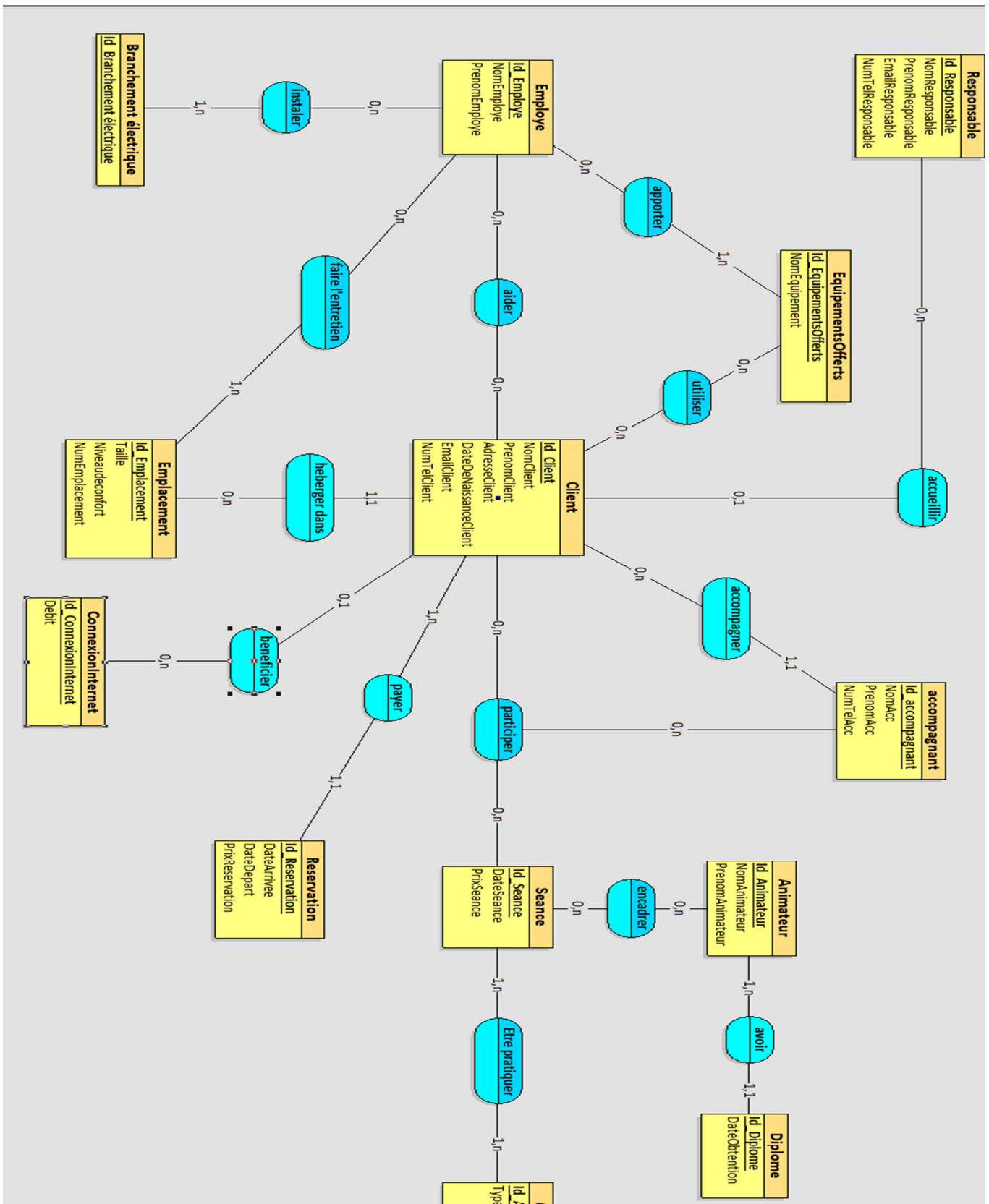
Attributs	Etre la relation entre
-	Client et Activite

- encadrer:

Attributs	Etre la relation entre
-	activite et Animateur

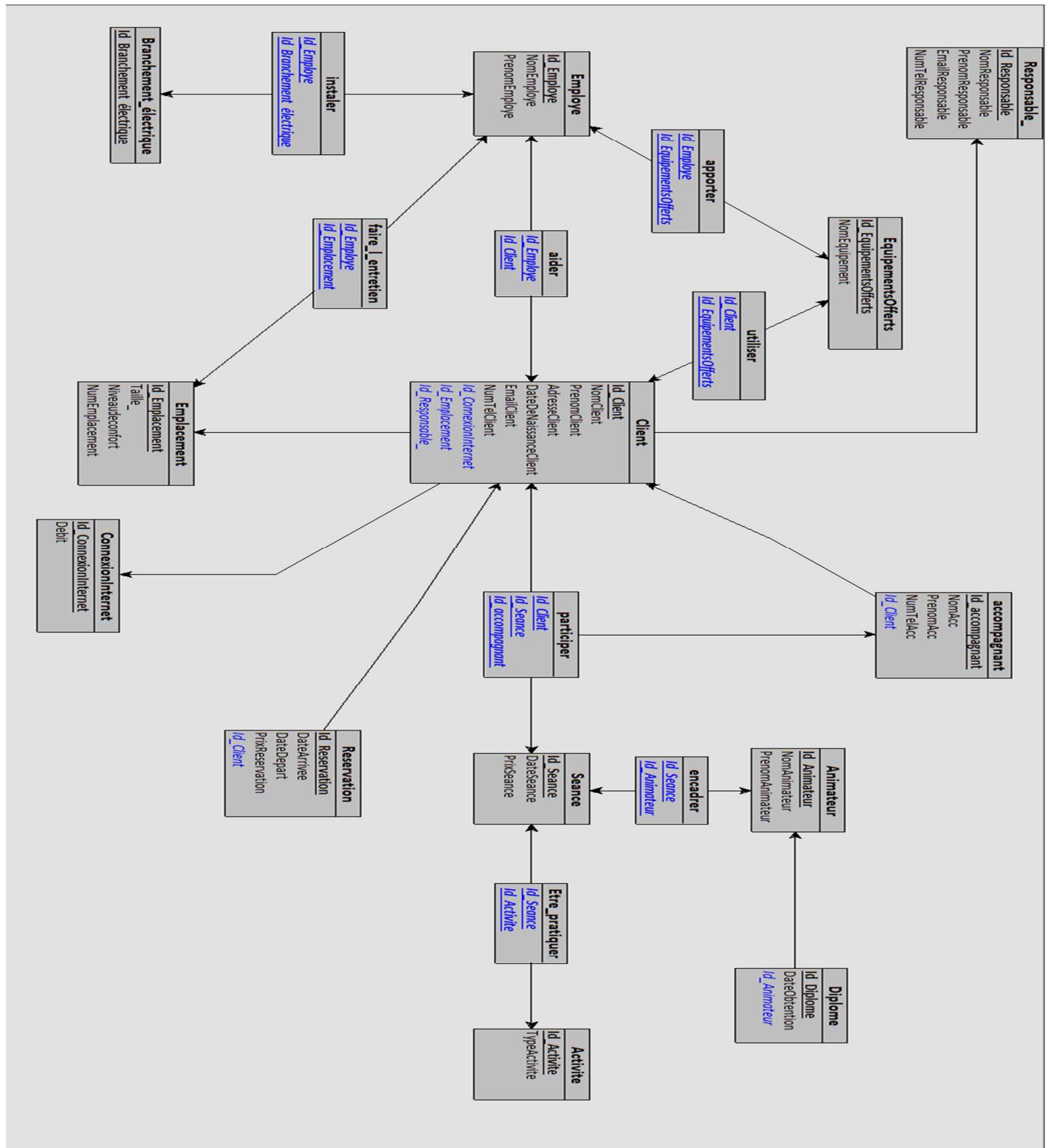
## Modèle conceptuel de données

Modèle conceptuel de données ou MCD est une représentation statique, structurée du système d'information d'une entreprise.



## Modèle logique de données

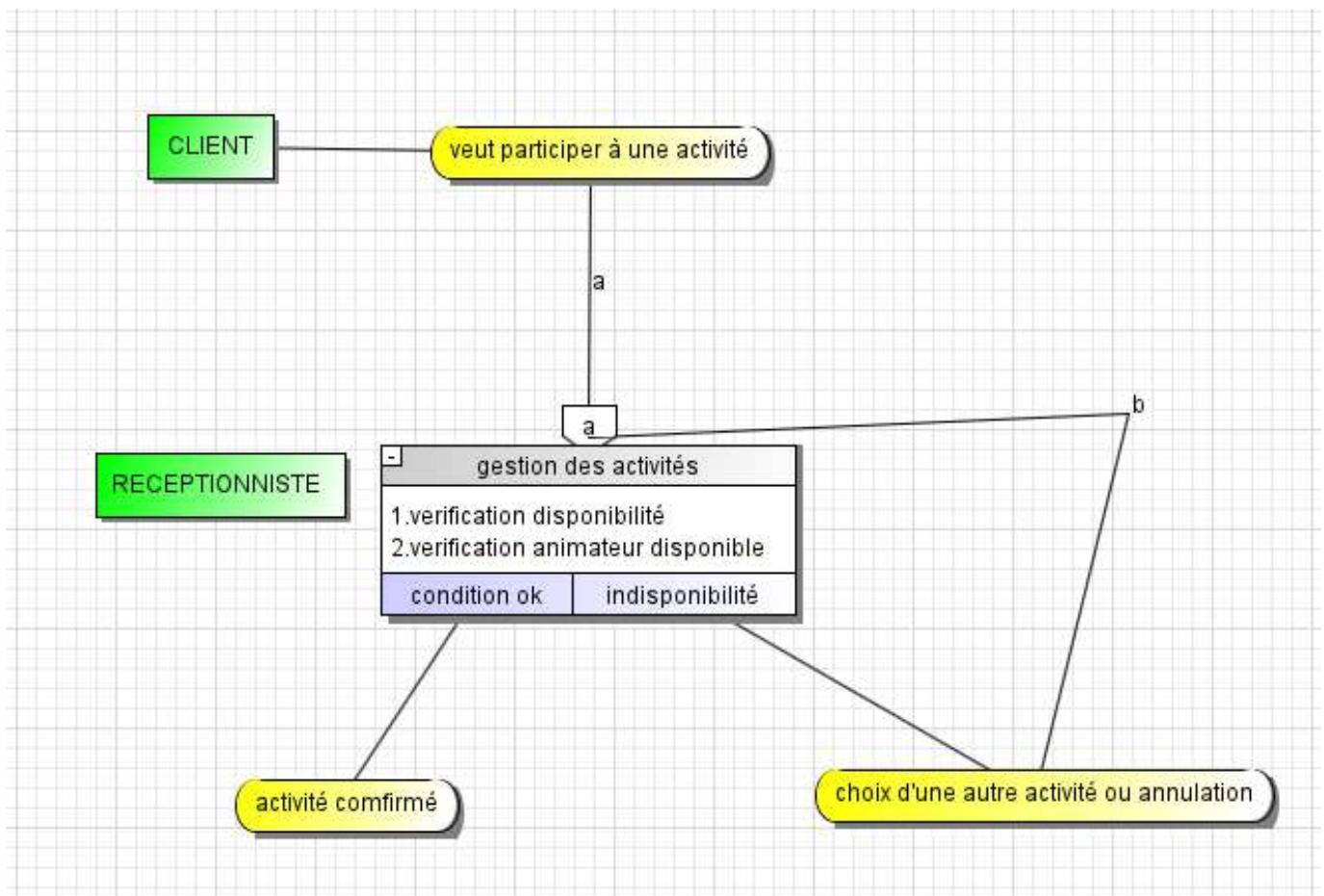
*Modèle logique de données ou MLD est la modélisation logique des données qui tient compte de l'organisation des données.*



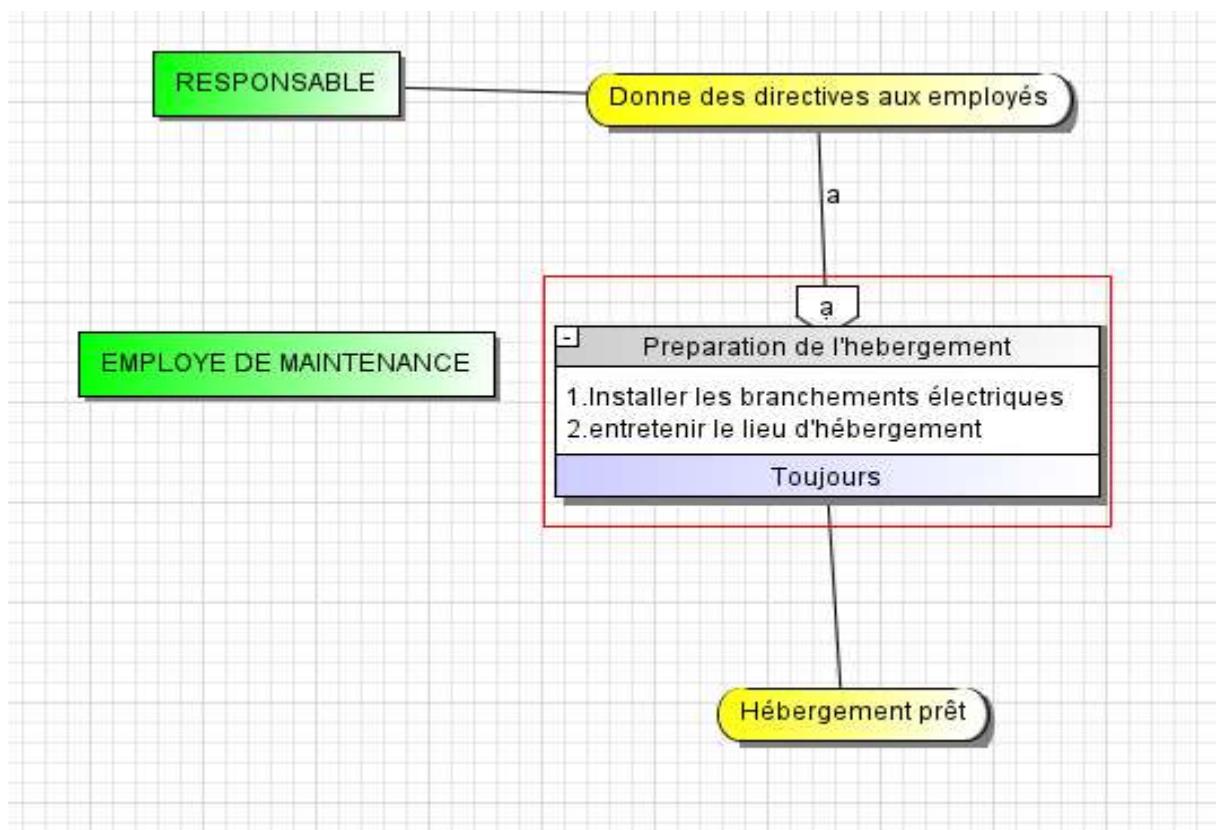
## Modele conceptuel de traitement

Modèle conceptuel de traitement ou MCT est la représentation dynamique du système d'information. Il permet de savoir quoi faire selon l'évènement en cours.

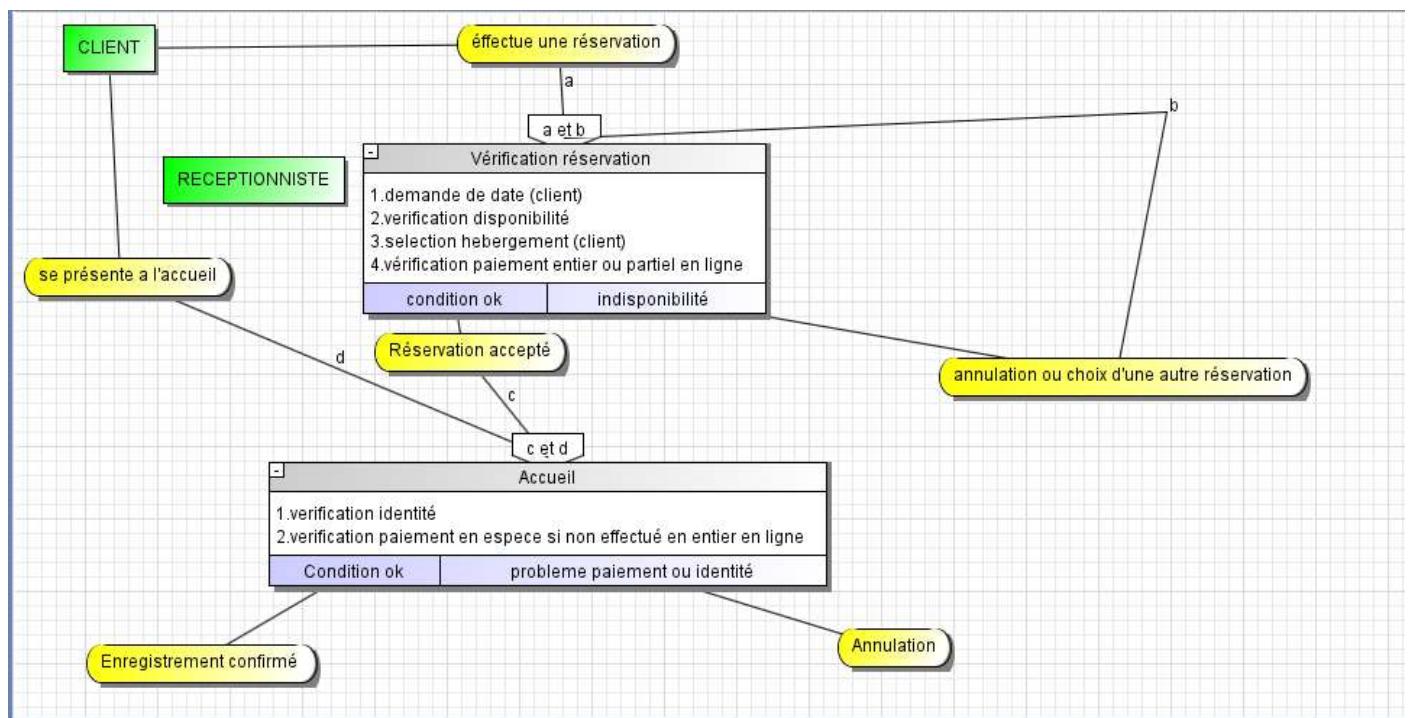
### Activité



## Hébergement



## Réervation



## Modèle relationnelle

*Modèle relationnelle ou MR est une façon de modéliser les différentes informations contenues dans une base de données.*

**Responsable** = (Id\_Responsable , NomResponsable, PrenomResponsable , Email , NumTelResponsable );

**Employe** = (Id\_Employe , NomEmploye , PrenomEmploye );

**hebergement** = (Id\_Hebergement , Taille, Niveaudeconfort, NumDemplacement );

**EquipementsOfferts** = (Id\_EquipementsOfferts , NomEquipement );

**Branchement\_électrique** = (Id\_Branchement\_électrique );

**ConnexionInternet** = (Id\_ConexionInternet , Debit );

**Activite** = (Id\_Activite , TypeActivite, );

**Seance** = (Id\_Seance , DateSeance, PrixSeance);

**Animateur** = (Id\_Animateur , NomAnimateur , PrenomAnimateur );

**Diplome** = (Id\_Diplome , DateObtention , #Id\_Animateur);

**TypedeReservation** = (Id\_Type );

**Client** = (Id\_Client , NomClient , PrenomClient , AdresseClient , DateDeNaissanceClient , Email , NumTelClient , #Id\_ConexionInternet, #Id\_Hebergement, #Id\_Responsable);

**Reservation** = (Id\_Reservation , DateArrivee, DateDepart, PrixReservation, #Id\_Client);

**accompagnant** = (Id\_accompagnant , NomAcc , PrenomAcc , NumTelAcc , #Id\_Client);

**aider** = (#Id\_Employe, #Id\_Client);

**apporter** = (#Id\_Employe, #Id\_EquipementsOfferts);

**instaler** = (#Id\_Employe, #Id\_Branchement\_électrique);

**encadrer** = (#Id\_Seance, #Id\_Animateur);

**faire\_L\_entretien** = (#Id\_Employe, #Id\_Hebergement);

**participer** = (#Id\_Client, #Id\_Seance ,#Id\_accompagnant);

utiliser = (#Id\_Client, #Id\_EquipementsOfferts);

Etre\_pratiquer = (#Id\_Seance, #Id\_Activite);

## Normalisation

La normalisation est une approche de conception de base de données utilisée dans les **bases de données relationnelles** pour éviter la redondance.

\*\*\*\*\*

**Client** (Id\_Client,NomClient, PrenomClient, DateDeNaissanceClient, EmailClient, NumTelClient, Id\_emplacement, Id\_connexioninternet, Id\_responsable)

F = { Id\_Client → NomClient 1  
Id\_Client → PrenomClient 2  
Id\_Client → DateDeNaissanceClient 3  
Id\_Client → EmailClient 4  
Id\_Client → NumTelClient 5  
Id\_Client → Id\_responsable 6  
Id\_Client → Id\_connexioninternet 7  
Id\_Client → Id\_emplacement 8  
id emplacement , id connexioninternet ,id responsable --> id client 9  
}

Calcul fermeture transitive de **Id\_Client**

(Id\_client)+= { Id\_Client, NomClient, PrenomClient, DateDeNaissanceClient, EmailClient, NumTelClient, Id\_emplacement, Id\_connexioninternet, Id\_responsable }

On retrouve bien tous les attributs de la relation donc **Id\_Client** est clé.

\*\*\*\*\*

**Responsable** (Id\_Responsable, NomResponsable, PrenomResponsable, EmailResponsable, NumTelResponsable)

F = { Id\_Responsable → NomResponsable 1  
Id\_Responsable → PrenomResponsable 2  
Id\_Responsable → EmailResponsable 3  
Id\_Responsable → NumTelResponsable }4

Calcul fermeture transitive de **Id\_Responsable**

(Id\_Responsable)+= { Id\_Responsable, NomResponsable, PrenomResponsable, EmailResponsable, NumTelResponsable }

On retrouve bien tous les attributs de la relation donc **Id\_Responsable** est clé.

## REQUETE SQL/Algebre

\*\*\*\*\* la liste des arrivées: \*\*\*\*\*

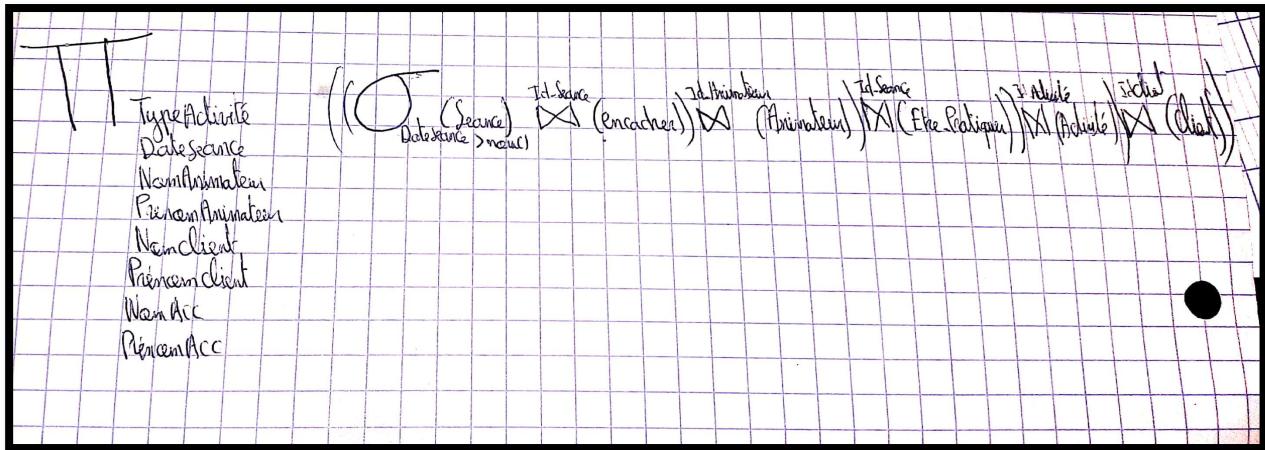
-- Je vais sélectionner la liste des arrivées du jour actuel.  
-- j'ai utilisé `CAST(CURRENT_TIMESTAMP AS DATE)` pour obtenir la journée actuelle  
-- J'ai également sélectionné le nom et le prénom de la personne responsable\_ qui l'accueillera pour faciliter la tâche d'accueil.

```
select Client.NomClient, Client.PrenomClient, reservation.DateArrivee,  
Emplacement.NumEmplacement,Responsable_.NomResponsable,  
Responsable_.PrenomResponsable,accompagnant.NomACC,accompagnant.PrenomAC  
C  
from Client  
inner join Emplacement on Client.Id_Emplacement = emplacement.Id_Emplacement  
inner join Reservation on Client.Id_Client = reservation.Id_Client  
inner join Responsable_ on Client.Id_Responsable_ = Responsable_.Id_Responsable_  
inner join accompagnant on Client.Id_Client = accompagnant.Id_Client  
where reservation.DateArrivee >= CAST(CURRENT_TIMESTAMP AS DATE)  
order by reservation.DateArrivee ASC;
```

\*\*\*\*\* la liste des départs: \*\*\*\*\*

-- je vais appliquer meme principe que celle j'ai appliqué avec la liste des arrivées  
-- ici j'ai selectionnée en plus le prix de réservation car ils veulent faire l'état des lieux et la facturation

```
select Client.NomClient, Client.PrenomClient, Reservation.DateDepart,  
Reservation.PrixReservation,  
Emplacement.NumEmplacement,accompagnant.NomACC,accompagnant.PrenomACC  
from Client  
inner join Emplacement on Client.Id_Emplacement = emplacement.Id_Emplacement  
inner join Reservation on Client.Id_Client = reservation.Id_Client  
inner join accompagnant on Client.Id_Client = accompagnant.Id_Client  
where reservation.DateArrivee >= CAST(CURRENT_TIMESTAMP AS DATE)  
order by reservation.DateDepart ASC;
```



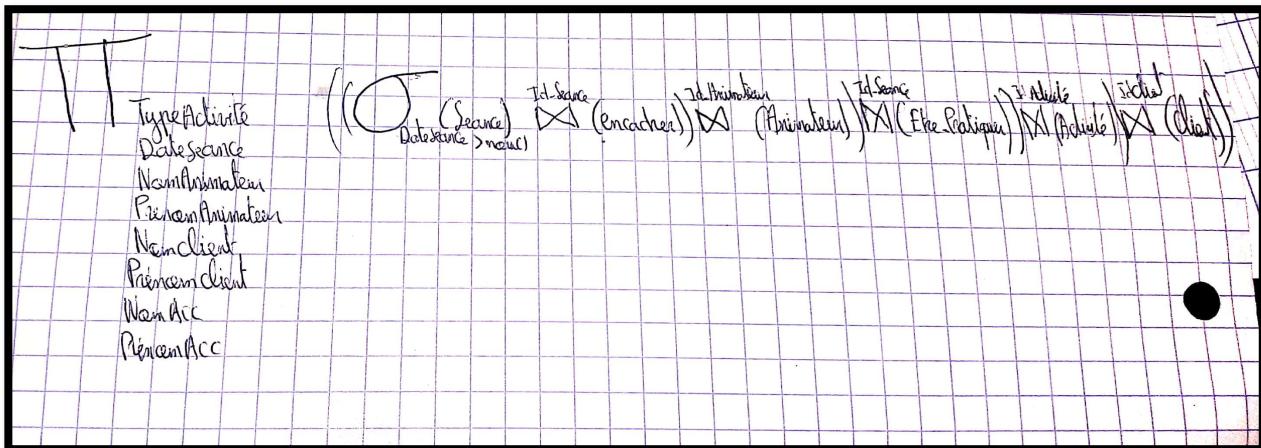
### \*\*\*\*\* le planning des activités: \*\*\*\*\*

-- je vais selectionner toute les activités ordonnées par le temps d'une façon ascendante pour faciliter les décisions

```

select Activite.TypeActivite, Seance.DateSeance, Animateur.NomAnimateur,
       Animateur.PrenomAnimateur, Client.NomClient,
       Client.PrenomClient, accompagnant.NomACC, accompagnant.PrenomACC
  from Seance
 inner join Etre_pratiquer on Seance.Id_Seance = Etre_pratiquer.Id_Seance
 inner join activite on Activite.Id_Activite = Etre_pratiquer.Id_Activite
 inner join encadrer on Seance.Id_Seance = encadrer.Id_Seance
 inner join Animateur on Animateur.Id_Animateur = encadrer.Id_Animateur
 inner join participer on Seance.Id_Seance = participer.Id_Seance
 inner join Client on Client.Id_Client = participer.Id_Client
 inner join accompagnant on Client.Id_Client = accompagnant.Id_Client
 WHERE Seance.DateSeance >= CAST(CURRENT_TIMESTAMP AS DATE)
 Order By Seance.DateSeance ASC;

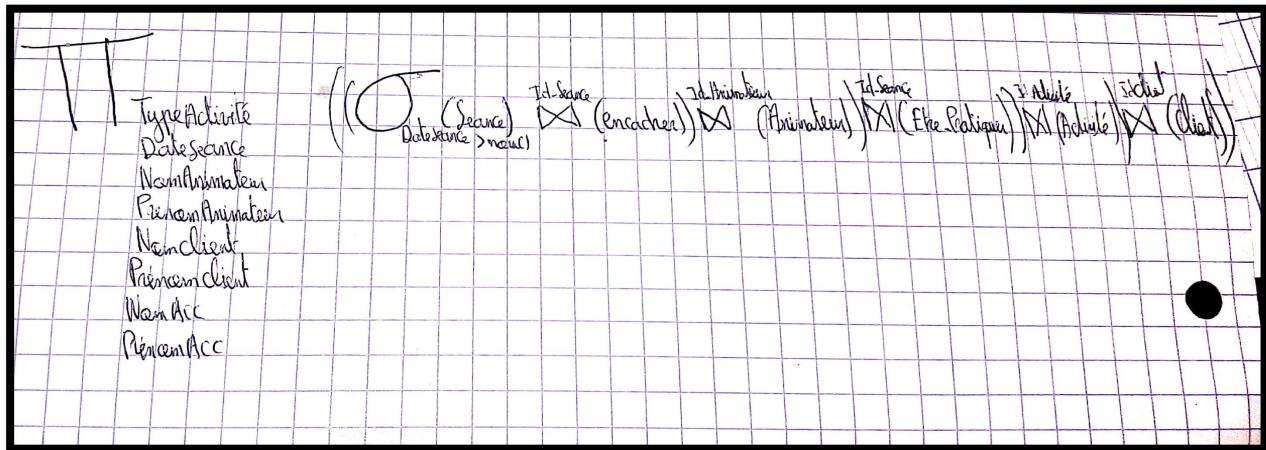
```



## \*\*\*\*\* planning de chaque animateur \*\*\*\*\*

-- je vais lister toute les activité avec leur type et son animateur ordonnées par le temps d'une façon ascendante

```
select Animateur.NomAnimateur, Animateur.PrenomAnimateur, Activite.TypeActivite,  
Seance.DateSeance  
from Animateur  
inner join encadrer on Animateur.Id_Animateur = encadrer.Id_Animateur  
inner join Seance on Seance.Id_Seance = encadrer.Id_Seance  
inner join Etre_pratiquer on Seance.Id_Seance = Etre_pratiquer.Id_Seance  
inner join Activite on Activite.Id_Activite = Etre_pratiquer.Id_Activite  
WHERE Seance.DateSeance >= CAST(CURRENT_TIMESTAMP AS DATE)  
Order by Seance.DateSeance ASC;
```



## \*\*\* la liste de personnel avec l'emplacement pour faire l'entretien et le ménage\*\*\*

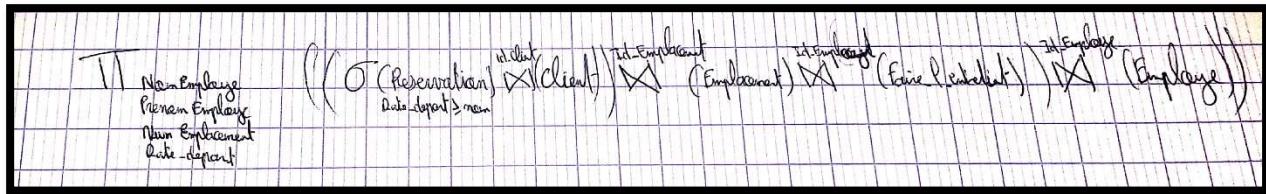
-- le personnel chargé de l'entretien

-- Je posterai la liste des sites qui seront libérés aujourd'hui avec son employé de maintenance pour les remettre en état.

```

select Employe.NomEmploye,
Employe.PrenomEmploye,Emplacement.NumEmplacement,Reservation.DateDepart
from Employe
inner join faire_1_entretien on Employe.Id_Employe = faire_1_entretien.Id_Employe
inner join Emplacement on Emplacement.Id_Emplacement =
faire_1_entretien.Id_Emplacement
inner join Client on Emplacement.Id_Emplacement = Client.Id_Emplacement
inner join Reservation on Reservation.Id_Client = Client.Id_Client
where Reservation.DateDepart >= CAST(CURRENT_TIMESTAMP AS DATE)
order by Reservation.DateDepart ASC;

```

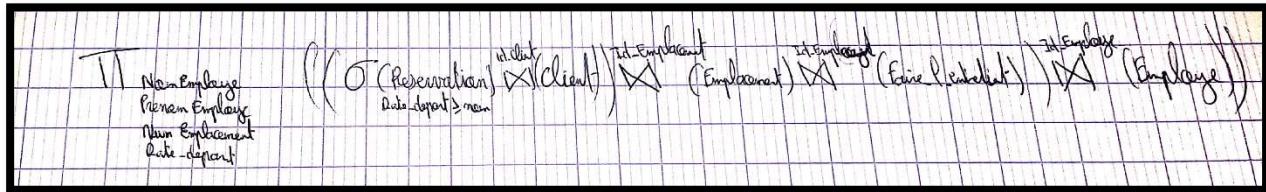


-- le même principe avec la liste de personnel qui vont faire le ménage (aide les clients)  
-- liste des arrivées avec le personnel responsable\_ pour faire préparer leur  
emplACEMENT

```

select Employe.NomEmploye,
Employe.PrenomEmploye,Emplacement.NumEmplacement,Reservation.DateArrivee
from Employe
inner join aider on Employe.Id_Employe = aider.Id_Employe
inner join Client on client.Id_Client = aider.Id_Client
inner join Emplacement on Emplacement.Id_Emplacement = Client.Id_Emplacement
inner join Reservation on Reservation.Id_Client = Client.Id_Client
where Reservation.DateArrivee >= CAST(CURRENT_TIMESTAMP AS DATE)
order by Reservation.DateDepart ASC;

```



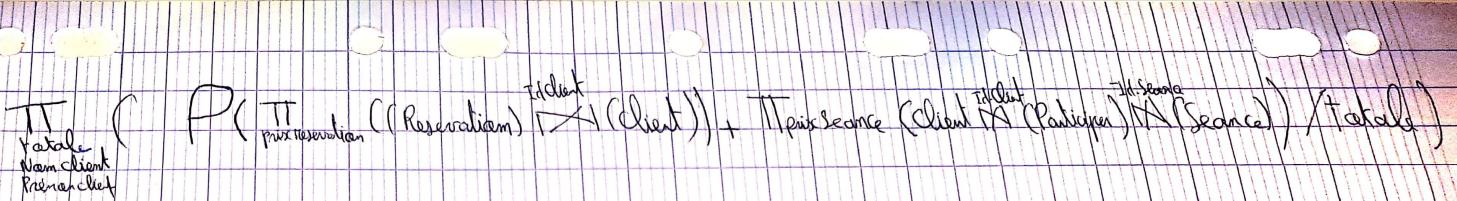
## \*\*\*\*\* la facturation de séjours \*\*\*\*\*

-- il s'agit de sélectionner tous les clients avec les prix de réservation de chaque enregistrement de ce client

```
select Client.NomClient, Client.PrenomClient, Reservation.PrixReservation,
ifnull(activite.TypeActivite,"") as activite, ifnull(seance.PrixSeance,0) as prix_activite
from Client
inner join Reservation on Client.Id_Client = Reservation.Id_Client
left join participer on client.Id_Client = participer.Id_Client
left join accompagnant on client.Id_Client = accompagnant.Id_Client
left JOIN seance on seance.Id_Seance = participer.Id_Seance
LEFT JOIN etre_pratiquer on seance.Id_Seance = etre_pratiquer.Id_Seance
left join activite on activite.Id_Activite = etre_pratiquer.Id_Activite
Group by Client.NomClient, Client.PrenomClient;
```

--Somme :

```
select Client.NomClient, Client.PrenomClient, Reservation.PrixReservation +
ifnull(Sum(seance.PrixSeance),0) as somme
from Client
inner join Reservation on Client.Id_Client = Reservation.Id_Client
left join participer on client.Id_Client = participer.Id_Client
left join accompagnant on client.Id_Client = accompagnant.Id_Client
left JOIN seance on seance.Id_Seance = participer.Id_Seance
LEFT JOIN etre_pratiquer on seance.Id_Seance = etre_pratiquer.Id_Seance
left join activite on activite.Id_Activite = etre_pratiquer.Id_Activite
Group by Client.NomClient, Client.PrenomClient;
```



## Conclusion

Tout au long de la préparation de notre projet de base de données, nous avons essayé de mettre en pratique les connaissances acquises au cours de ce semestre universitaire afin de Créer une base de données complète et entièrement fonctionnelle .Nous avons donc notre base de données permettant la gestion d'un camping, que nous avons développé grâce à la méthode merise.

Nous avons donc un MCD Il y a aussi les requêtes SQL comme concernant les planning pour les activités et les animateurs .

Nous avons donc notre base de données permettant la gestion d'un camping, que nous avons développé grâce à la méthode merise.

il y a un point en ce qui concerne la sécurité. Les informations des clients contient des informations confidentielles, l'administrateur et les responsables ont la vision sur tout ces informations.

vous trouvez ci-joint tous les fichiers (MCD,MLD,La base de données,Matrice de flux...)

## ANNEXE

### Script de la création de la base de données

#### **@ TABLE Responsable**

```
CREATE TABLE Responsable_(
    Id_Responsable_ INT AUTO_INCREMENT,
    NomResponsable VARCHAR(50) ,
    PrenomResponsable VARCHAR(50) ,
    EmailResponsable VARCHAR(50) ,
    NumTelResponsable INT,
    PRIMARY KEY(Id_Responsable_)
);
```

#### **@ TABLE Employe**

```
CREATE TABLE Employe(
    Id_Employe INT AUTO_INCREMENT,
    NomEmploye VARCHAR(50) ,
    PrenomEmploye VARCHAR(50) ,
    PRIMARY KEY(Id_Employe)
);
```

## **@ TABLE emplacement**

```
CREATE TABLE Emplacement(
    Id_EmplACEMENT INT AUTO_INCREMENT,
    Taille_ VARCHAR(50) ,
    Niveau de confort VARCHAR(50) ,
    NumEmplacement INT,
    PRIMARY KEY(Id_EmplACEMENT)
);
```

## **@ TABLE EquipementsOfferts**

```
CREATE TABLE EquipementsOfferts(
    Id_EquipementsOfferts INT AUTO_INCREMENT,
    NomEquipement VARCHAR(50) ,
    PRIMARY KEY(Id_EquipementsOfferts)
);
```

## **@ TABLE Branchement\_electrique**

```
CREATE TABLE Branchement_electrique(
    Id_Branchement_electrique INT AUTO_INCREMENT,
    PRIMARY KEY(Id_Branchement_electrique)
);
```

## **@ TABLE ConnexionInternet**

```
CREATE TABLE ConnexionInternet(
    Id_ConexionInternet INT AUTO_INCREMENT,
    Debit INT,
    PRIMARY KEY(Id_ConexionInternet)
);
```

## **@ TABLE Seance**

```
CREATE TABLE Seance(
    Id_Seance INT AUTO_INCREMENT,
    DateSeance DATETIME,
    PrixSeance DOUBLE,
```

```
    PRIMARY KEY(Id_Seance)
);
```

## @ TABLE Activite

```
CREATE TABLE Activite(
    Id_Activite INT AUTO_INCREMENT,
    TypeActivite VARCHAR(50) ,
    PRIMARY KEY(Id_Activite)
);
```

## @ TABLE Animateur

```
CREATE TABLE Animateur(
    Id_Animateur INT AUTO_INCREMENT,
    NomAnimateur VARCHAR(50) ,
    PrenomAnimateur VARCHAR(50) ,
    PRIMARY KEY(Id_Animateur)
);
```

## @ TABLE Diplome

```
CREATE TABLE Diplome(
    Id_Diplome INT AUTO_INCREMENT,
    DateObtention DATE,
    Id_Animateur INT NOT NULL,
    PRIMARY KEY(Id_Diplome),
    FOREIGN KEY(Id_Animateur) REFERENCES Animateur(Id_Animateur)
);
```

## @ TABLE Client

```
CREATE TABLE Client(
    Id_Client INT AUTO_INCREMENT,
    NomClient VARCHAR(50) ,
    PrenomClient VARCHAR(50) ,
    AdresseClient VARCHAR(50) ,
    DateDeNaissanceClient VARCHAR(50) ,
    EmailClient VARCHAR(100) ,
    NumTelClient INT,
    Id_ConexionInternet INT,
```

```

Id_EmplACEMENT INT NOT NULL,
Id_Responsable_ INT,
PRIMARY KEY(Id_Client),
FOREIGN KEY(Id_ConexionInternet) REFERENCES
ConnexionInternet(Id_ConexionInternet),
FOREIGN KEY(Id_EmplACEMENT) REFERENCES Emplacement(Id_EmplACEMENT),
FOREIGN KEY(Id_Responsable_) REFERENCES Responsable_(Id_Responsable_)
);

```

## @ TABLE Reservation

```

CREATE TABLE Reservation(
Id_Reservation INT AUTO_INCREMENT,
DateArrivee DATETIME,
DateDepart DATETIME,
PrixReservation DOUBLE,
Id_Client INT NOT NULL,
PRIMARY KEY(Id_Reservation),
FOREIGN KEY(Id_Client) REFERENCES Client(Id_Client)
);

```

## @ TABLE accompagnant

```

CREATE TABLE accompagnant(
Id_accompagnant INT AUTO_INCREMENT,
NomAcc VARCHAR(50) ,
PrenomAcc VARCHAR(50) ,
NumTelAcc INT,
Id_Client INT NOT NULL,
PRIMARY KEY(Id_accompagnant),
FOREIGN KEY(Id_Client) REFERENCES Client(Id_Client)
);

```

## @ TABLE aider

```

CREATE TABLE aider(
Id_Employe INT,
Id_Client INT,
PRIMARY KEY(Id_Employe, Id_Client),
FOREIGN KEY(Id_Employe) REFERENCES Employe(Id_Employe),
FOREIGN KEY(Id_Client) REFERENCES Client(Id_Client)
);

```

);

## @ TABLE apporter

```
CREATE TABLE apporter(
    Id_Employe INT,
    Id_EquipementsOfferts INT,
    PRIMARY KEY(Id_Employe, Id_EquipementsOfferts),
    FOREIGN KEY(Id_Employe) REFERENCES Employe(Id_Employe),
    FOREIGN KEY(Id_EquipementsOfferts) REFERENCES
    EquipementsOfferts(Id_EquipementsOfferts)
);
```

## @ TABLE installer

```
CREATE TABLE instaler(
    Id_Employe INT,
    Id_Branchement_électrique INT,
    PRIMARY KEY(Id_Employe, Id_Branchement_électrique),
    FOREIGN KEY(Id_Employe) REFERENCES Employe(Id_Employe),
    FOREIGN KEY(Id_Branchement_électrique) REFERENCES
    Branchement_électrique(Id_Branchement_électrique)
);
```

## @ TABLE encadrer

```
CREATE TABLE encadrer(
    Id_Seance INT,
    Id_Animateur INT,
    PRIMARY KEY(Id_Seance, Id_Animateur),
    FOREIGN KEY(Id_Seance) REFERENCES Seance(Id_Seance),
    FOREIGN KEY(Id_Animateur) REFERENCES Animateur(Id_Animateur)
);
```

## @ TABLE faire\_l\_entretien

```
CREATE TABLE faire_l_entretien(
    Id_Employe INT,
    Id_EmplACEMENT INT,
    PRIMARY KEY(Id_Employe, Id_EmplACEMENT),
    FOREIGN KEY(Id_Employe) REFERENCES Employe(Id_Employe),
    FOREIGN KEY(Id_EmplACEMENT) REFERENCES Emplacement(Id_EmplACEMENT)
);
```

## **@ TABLE participer**

```
CREATE TABLE participer(
    Id_Client INT,
    Id_Seance INT,
    Id_accompagnant INT,
    PRIMARY KEY(Id_Client, Id_Seance, Id_accompagnant),
    FOREIGN KEY(Id_Client) REFERENCES Client(Id_Client),
    FOREIGN KEY(Id_Seance) REFERENCES Seance(Id_Seance),
    FOREIGN KEY(Id_accompagnant) REFERENCES accompagnant(Id_accompagnant)
);
```

## **@ TABLE utiliser**

```
CREATE TABLE utiliser(
    Id_Client INT,
    Id_EquipementsOfferts INT,
    PRIMARY KEY(Id_Client, Id_EquipementsOfferts),
    FOREIGN KEY(Id_Client) REFERENCES Client(Id_Client),
    FOREIGN KEY(Id_EquipementsOfferts) REFERENCES
    EquipementsOfferts(Id_EquipementsOfferts)
);
```

## **@ TABLE Etre\_pratiquer**

```
CREATE TABLE Etre_pratiquer(
    Id_Seance INT,
    Id_Activite INT,
    PRIMARY KEY(Id_Seance, Id_Activite),
    FOREIGN KEY(Id_Seance) REFERENCES Seance(Id_Seance),
    FOREIGN KEY(Id_Activite) REFERENCES Activite(Id_Activite)
);
```