



RAPPORT DE STAGE

BTS Services Informatique aux Organisations – Spécialité SLAM

Mai à Juin 2020

KANANKE ACHARIGE Sahan

Tuteur en entreprise et académique : Madame Béatrice CHATAING

Etablissement/Formation et entreprise d'accueil : Sup'Chassagnes

Sommaire

I - INTRODUCTION.....	4
II – Présentation de l’organisation.....	5
1 – Le projet.....	5
2 – Le thème de stage	6
III – Les missions	6
IV – Mission BO1.....	7
1 – Objectif	7
2 – Création d’un MCD	7
3 – Les utilisateurs	9
4 – Classes en PHP.....	9
5 – Jeux de tests.....	10
V – Mission BO2.....	11
1 – Objectif	11
2 – Premier volet.....	11
3 – Second volet.....	11
4 – Jeux de tests.....	12
VI – Mission BO3.....	14
1 – Objectif	14
2 – Changement dans la base de données	14
3 – PHP	15
4 – Erreur sur la base de données.....	15
VII – Problèmes rencontrés.....	Erreur ! Signet non défini.
VIII - Conclusion	16
IX – Annexes	18

Mes remerciements,

*Je tiens à remercier l'école supérieure Sup'Chassagnes et mes différents professeurs pour
m'avoir permis de faire ce stage.*

*De plus, je souhaiterais remercier tout particulièrement, Mme.CHATAING qui m'a suivi et
encouragé tout au long de mon année d'étude. Elle a réussi à mettre en place un stage
fictif en vue de la crise sanitaire que nous traversons et a permis de créer différentes
missions pour ce stage fictif.*

I - INTRODUCTION

Tout d'abord, je souhaiterais me présenter : Je m'appelle KANANKE ACHARIGE Sahan, j'ai 21 ans et je suis étudiant à l'école supérieur Sup'Chassagnes. Je suis actuellement en BTS SIO, Services Informatiques aux Organisations en spécialité SLAM, Solutions Logicielles et Applications Métiers.

Dans le cadre de la validation de ma première année de BTS SIO, j'ai effectué le stage obligatoire de fin d'année scolaire. De part la crise sanitaire que nous avons traversé du 17 Mars au 11 Mai 2020, trouver un lieu de stage durant cette période s'est avéré difficile. Notre professeure principale a donc mis en place un stage fictif à propos d'une organisation réelle afin de combler l'impossibilité, ou presque, d'obtenir un stage durant cette année, il y avait donc une mission à accomplir parmi 15 différentes afin de faire valider son stage. J'en ai donc effectué trois durant cette période stage.

II – Présentation de l'organisation

1 – Le projet

Le club « *Lyon Palme* » est une association sportive de nage avec palme : monopalme ou bi-palmes. Ce club est affilié à la Fédération Française d'Étude et de Sports Sous-Marin (FFESSM). Il compte une quarantaine d'adhérents, son siège se trouve à Saint Fons. Par ailleurs, l'association organise régulièrement des courses en eau libre, et ce partout en France. Leur but est de partager leur passion et de nager dans des endroits inédits. Il y a un calendrier visible sur leur site accessible par les adhérents afin de participer à des séances à plusieurs.

Leur bureau est composé d'un président, d'un président adjoint, d'un trésorier, d'un secrétaire et d'un secrétaire adjoint.

D'autres membres sont aussi très actifs :

- Le responsable communication qui s'occupe du site Internet, de la page Facebook et du compte Instagram ;
- Plusieurs coachs qui préparent les séances et encadrent les sorties ;
- Plusieurs entraîneurs qui animent les séances et encadrent les sorties ;
- Le responsable planning entraînement qui est chargé de prévoir deux initiateurs pour chaque séance piscine. Son rôle est de récolter les disponibilités des entraînements et élaborer en fonction un calendrier que les entraîneurs doivent respecter.

2 – Le thème de stage

Le but de mon stage est de mettre en place une base de données, ainsi que la création de ses utilisateurs. Par ailleurs, il fallait aussi que je crée quelques pages web et classes en PHP.

III – Les missions

Il y a deux types de missions :

- Front office : Création d'un site et des différents paramètres visibles par les utilisateurs (site vitrine, connexion, inscription, etc.)
- Back office : Création d'une base de données et des différents paramètres non visibles par les utilisateurs (base de données, tables, utilisateurs, mot de passes, etc.)

Au vu de mes difficultés en développement « Back Office », j'ai choisi de concentrer mes efforts sur ce type de mission. J'ai donc décidé de profiter de ce stage afin d'améliorer mes compétences, particulièrement dans le langage SQL et dans l'analyse des règles de gestion.

IV – Mission BO1

1 – Objectif

La première mission Back Office consiste à la création et à l'administration d'une base de données sous MySQL à l'aide de règles de gestion fournies.

2 – Création d'un MCD

La première chose à faire est la création d'un MCD, pour cela, je me suis référé à un document nommé « Règles de gestion » afin de récupérer les informations les plus importantes afin de créer un MCD. J'ai donc fait un résumé de toutes ces règles en retirant toutes les informations inutiles dans le but d'obtenir un document comportant toutes les règles de gestion au propre.

A partir de ce document, j'ai élaboré un premier MCD comportant des adhérents, des entraîneurs, des sorties, des compétitions et des séances.

3 – Les utilisateurs

Après avoir créé la table, j'ai décidé d'y implanter les différents utilisateurs :

- Administrateur
- Secrétaire
- Entraîneur
- Coach
- Matériel

Ces différents utilisateurs ont des droits attribués en fonction de leur rôle :

Utilisateurs	Droits
Administrateur	Tous les droits
Secrétaire	Aucun pour le moment
Entraîneur	Droits sur les tables « sortie » et « compétition »
Coach	Droits sur la table « entraînement »
Matériel	Droits sur les tables « calendrier », « prêt » et « matériel »

4 – Classes en PHP

Pour ne pas oublier d'aider les étudiants travaillant sur le Front Office, il m'est impératif de créer des classes en PHP afin de leur faciliter la tâche. Or, au vu de la nécessité d'utiliser l'héritage, il m'a fallu apprendre comment

cela fonctionne. Ainsi, j'ai pris connaissance de son utilisation en PHP grâce à des cours sur Internet. Par conséquent, j'ai pu créer mes différents utilisateurs.

5 – Jeux de tests

Pour finaliser cette première mission, j'ai dû tester la base de données et ainsi y insérer des données.

Par la suite, je me suis connecté en tant qu'administrateur afin d'essayer l'insertion de données, la récupération de données et différentes manipulations sur les tables qui ont toutes fonctionnées.

Ensuite, j'ai essayé avec tous les autres utilisateurs sauf le « *secrétaire* » pour vérifier si les différents privilèges accordés fonctionnaient : ceci fut concluant.

Enfin, j'ai créé une page PHP sans contenu afin d'afficher quelques informations provenant de la base de données SQL avec succès. C'est ainsi que s'achève ma première mission en Back Office.

V – Mission BO2

1 – Objectif

L'objectif de la mission Back Office 2 est le développement en PHP objet de l'application « *Inscription des adhérents* ».

Le projet se divise en 2 volets :

- Le secrétaire doit pouvoir administrer les inscriptions des adhérents.
- Les nageurs connectés doivent pouvoir afficher le trombinoscope des adhérents pour la saison en cours ainsi que l'annuaire et doit pouvoir modifier son compte.

2 – Premier volet

Il faut tout d'abord donner des privilèges à l'utilisateur « secrétaire » qui n'ont pas été donné plus tôt. Le secrétaire doit avoir l'accès à la table adhérent et toutes ses classes filles. A partir de là, le secrétaire peut d'ores et déjà modifier les adhérents, et en ajouter à sa guise.

3 – Second volet

Lors du second volet, on remarque déjà que les tables créées précédemment ne correspondent pas à la demande, il faut qu'il soit possible de voir les photos des autres adhérents en tant qu'adhérent. Il faut donc

ajouter une cellule dans la table « Adhérent » permettant de stocker l'image de l'adhérent. Par la suite, il faut modifier les droits pour avoir accès à la lecture de différentes informations à propos d'un adhérent :

- Nom
- Prénom
- Photo de profil
- Rôle (simple adhérent, secrétaire, entraîneur, etc.)

Il faut tout d'abord créer un nouvel utilisateur « Adherent » puis modifier ses privilèges afin de permettre la modification de son propre compte. A partir de la base de données il est difficile d'empêcher un utilisateur de modifier n'importe quelle donnée dans la table « Adhérent ». Or, il est possible de le manipuler en PHP.

4 – Jeux de tests

Avant de tester la modification de son compte à un adhérent, il est important de tester s'il est possible au secrétaire de faire des modifications à propos des nouveaux adhérents.

Pour se faire, il faut se connecter simplement sur MySQL en tant que secrétaire, après une suite de tests (insertion de donnée, suppression de données, modification de données etc.), on en conclue que le secrétaire est seulement apte à ajouter ou supprimer des adhérents.

Pour l'utilisateur « Adhérent », il lui est autorisé de voir que certaines données : le nom, prénom, rôle et la photo de profil des autres adhérents. Il faut ensuite tester la possibilité des adhérents à modifier son propre mot de

passé. J'ai pour cela créé un site PHP très simple qui se connecte avec un utilisateur spécifique. Dans le code PHP, il est stipulé que le mot de passe du compte de l'adhérent connecté actuellement, deviendra « Test ». Ce test fût un succès. Ainsi, les limites des droits des adhérents ne seront pas définies dans la base de données MySQL mais directement en PHP en permettant que l'utilisateur connecté actuellement ne modifie que ses données personnelles.

VI – Mission BO3

1 – Objectif

L'objectif de la mission Back Office 2 est le développement en PHP objet de l'application « Planning des entraînements ».

Le but est de donner les droits à l'entraîneur responsable planning afin de lui permettre de créer des entraînements. Il faut aussi la possibilité aux entraîneurs de signaler une indisponibilité et de proposer un échange.

2 – Changement dans la base de données

Encore une fois, il est nécessaire de modifier la base de données. Il faut mettre en place un moyen de signaler son indisponibilité et de proposer un échange. J'ai rajouté dans la table « Entraînement », une cellule « Echange » et une autre cellule « Indisponibilité ».

De cette manière, il est possible, aux différents entraîneurs censés pouvoir visualiser le planning des entraînements, de communiquer avec l'entraîneur responsable planning.

3 – PHP

Après modification de la base de données, il faut se charger des classes métier, des DAO et du contrôleur en PHP.

4 – Erreur sur la base de données

Arrivé à ce niveau, je me suis rendu compte d'une erreur dans ma base de données SQL : Seulement un seul adhérent pouvait participer à une seule séance. J'ai ainsi modifié ma base de données afin de permettre à plus d'un adhérent de participer à une séance. J'ai donc dû supprimer la base de données et la reconstruire après modification.

VIII - Conclusion

Pour conclure, arrivé à la fin du stage, j'ai réussi à terminer 3 missions parmi celles données. Or, pour ma part, je ne trouve pas cela assez en fonction du temps qu'il m'a été accordé. J'ai trop souvent bloqué sur des étapes qui auraient pu me prendre moins de temps à finaliser. Tout d'abord, j'ai commis l'erreur de créer des utilisateurs avant de créer le MCD et la base de données. J'ai donc supprimé tous ces utilisateurs et opté pour la création d'un MCD. Ensuite, j'ai rencontré quelques problèmes au niveau des tables dites « Compromises ». Lors de l'attribution des différents privilèges sur les utilisateurs, il est apparu une erreur disant qu'une table nommée « tables_priv » située dans la database « mysql » était corrompue. Sans résoudre ce problème, il m'était impossible de donner des privilèges. Heureusement, j'ai trouvé des informations sur un forum m'apprenant la commande « repair » : J'ai dû me connecter en tant qu'administrateur sur MySQL afin de lancer une commande qui permet de réparer les erreurs d'une table endommagée. Par ailleurs, mes difficultés sur l'analyse des règles de gestion m'ont aussi retardé énormément, pendant les premières semaines de stage, la base de données changeait sans cesse. Cependant, ses erreurs m'ont permis de découvrir de nouvelles connaissances.

Pour la première fois, nous avons eu accès à des règles de gestion très complexes et détaillées d'une organisation réelle. Il était donc nécessaire d'extraire toutes les informations utiles, ce qui n'était pas le cas auparavant. Grâce à cela j'ai réussi à développer mes capacités d'analyse des règles de gestion mais cela reste à améliorer.

Par ailleurs, ce stage va m'aider dans mes futurs projets : il m'a appris le travail en entreprise et connaître leur organisation. Pour finir, avoir choisi de travailler dans un domaine où je suis très peu à l'aise, tel que dans le SQL et l'élaboration d'un MCD, m'ont aidé à me sentir plus à l'aise avec cela.


```
ADHFonctionBureau VARCHAR(50) UNIQUE,  
ADHAutreFonction VARCHAR(50) UNIQUE,  
ADHId VARCHAR(50) UNIQUE,  
ADHMdp VARCHAR(50) UNIQUE,  
PRIMARY KEY(ADHNum)  
);
```

```
CREATE TABLE COMPETITION(  
    COMPNum VARCHAR(50),  
    COMPLibelle VARCHAR(50),  
    COMPFinInscription VARCHAR(50),  
    PRIMARY KEY(COMPNum)  
);
```

```
CREATE TABLE MATERIEL(  
    MATNum VARCHAR(50),  
    MATLibelle VARCHAR(100),  
    MATMarque VARCHAR(50),  
    MATTaille DECIMAL(15,2),  
    PRIMARY KEY(MATNum)  
);
```

```
CREATE TABLE ENTRAINEUR(  
    ADHNum VARCHAR(50),  
    PRIMARY KEY(ADHNum),  
    FOREIGN KEY(ADHNum) REFERENCES ADHERENT(ADHNum)  
);
```

```
CREATE TABLE COACH(  
    ADHNum VARCHAR(50),  
    PRIMARY KEY(ADHNum),  
    FOREIGN KEY(ADHNum) REFERENCES ADHERENT(ADHNum)  
);
```

```
ADHNum VARCHAR(50),  
PRIMARY KEY(ADHNum),  
FOREIGN KEY(ADHNum) REFERENCES ENTRAINEUR(ADHNum)  
);
```

```
CREATE TABLE CALENDRIER(  
    DatePretMateriel DATE,  
    PRIMARY KEY(DatePretMateriel)  
);
```

```
CREATE TABLE MODALITE(  
    MODNum VARCHAR(50),  
    MODLibelle VARCHAR(50),  
    PRIMARY KEY(MODNum)  
);
```

```
CREATE TABLE DISTANCE(  
    METRES_DISTANCE INT,  
    PRIMARY KEY(METRES_DISTANCE)  
);
```

```
CREATE TABLE SORTIE(  
    SORTIENum VARCHAR(50),  
    SORTIEHeureDebut TIME,  
    SORTIELieu VARCHAR(50),  
    SORTIEPlage VARCHAR(50),  
    SORTIENiveauPublic VARCHAR(50),  
    SORTIEMentionObligatoire VARCHAR(50),  
    SORTIEMentionHiver VARCHAR(50),
```

```
ADHNum VARCHAR(50) NOT NULL,  
PRIMARY KEY(SORTIENum),  
FOREIGN KEY(ADHNum) REFERENCES ENTRAINEUR(ADHNum)  
);
```

```
CREATE TABLE ENTRAINEMENT(  
    ENTRAINEMENTNum VARCHAR(50),  
    ENTRAINEMENTDate DATE,  
    ADHNum VARCHAR(50) NOT NULL,  
    PRIMARY KEY(ENTRAINEMENTNum),  
    FOREIGN KEY(ADHNum) REFERENCES COACH(ADHNum)  
);
```

```
CREATE TABLE SEANCE(  
    SEANCENum VARCHAR(50),  
    SEANCEDate VARCHAR(50),  
    SEANCEContenu VARCHAR(50),  
    ENTRAINEMENTNum VARCHAR(50) NOT NULL,  
    PRIMARY KEY(SEANCENum),  
    FOREIGN KEY(ENTRAINEMENTNum) REFERENCES  
ENTRAINEMENT(ENTRAINEMENTNum)  
);
```

```
CREATE TABLE S_INSCRIRE(  
    ADHNum VARCHAR(50),  
    SORTIENum VARCHAR(50),  
    PRIMARY KEY(ADHNum, SORTIENum),  
    FOREIGN KEY(ADHNum) REFERENCES ADHERENT(ADHNum),  
    FOREIGN KEY(SORTIENum) REFERENCES SORTIE(SORTIENum)
```

);

CREATE TABLE PRETER(

MATNum VARCHAR(50),

DatePretMateriel DATE,

PRETERDate DATE,

ADHNum VARCHAR(50) NOT NULL,

PRIMARY KEY(MATNum, DatePretMateriel),

FOREIGN KEY(MATNum) REFERENCES MATERIEL(MATNum),

FOREIGN KEY(DatePretMateriel) REFERENCES CALENDRIER(DatePretMateriel),

FOREIGN KEY(ADHNum) REFERENCES ADHERENT(ADHNum)

);

CREATE TABLE PARTICIPERCOMP(

ADHNum VARCHAR(50),

COMPNum VARCHAR(50),

PARTModaliteChoisie VARCHAR(50),

PARTDistanceChoisie INT,

PARTBesoinCovoiturage VARCHAR(50),

PARTProposerCovoiturage VARCHAR(50),

PARTNbPlacesVoiture INT,

PARTHebergement VARCHAR(50),

PARTNbPersonneAccompagnee VARCHAR(50),

PRIMARY KEY(ADHNum, COMPNum),

FOREIGN KEY(ADHNum) REFERENCES ADHERENT(ADHNum),

FOREIGN KEY(COMPNum) REFERENCES COMPETITION(COMPNum)

);

CREATE TABLE FAIRE(

```

ADHNum VARCHAR(50),
SEANNum VARCHAR(50),
FAIREIndispo VARCHAR(50),
FAIREProposerEchanger VARCHAR(50),
PRIMARY KEY(ADHNum, SEANNum),
FOREIGN KEY(ADHNum) REFERENCES ENTRAINEUR(ADHNum),
FOREIGN KEY(SEANNum) REFERENCES SEANCE(SEANNum)
);

```

```

CREATE TABLE PROPOSERMOD(
    COMPNum VARCHAR(50),
    MODNum VARCHAR(50),
    PRIMARY KEY(COMPNum, MODNum),
    FOREIGN KEY(COMPNum) REFERENCES COMPETITION(COMPNum),
    FOREIGN KEY(MODNum) REFERENCES MODALITE(MODNum)
);

```

```

CREATE TABLE PROPOSERDIST(
    COMPNum VARCHAR(50),
    METRES_DISTANCE INT,
    PRIMARY KEY(COMPNum, METRES_DISTANCE),
    FOREIGN KEY(COMPNum) REFERENCES COMPETITION(COMPNum),
    FOREIGN KEY(METRES_DISTANCE) REFERENCES DISTANCE(METRES_DISTANCE)
);

```

BO1 : Création des utilisateurs et don des privilèges

```
CREATE USER 'Administrateur'@'localhost' IDENTIFIED BY 'admin';  
GRANT ALL PRIVILEGES ON lyonpalme.* TO 'Administrateur'@'localhost';
```

```
CREATE USER 'Secetaire'@'localhost' IDENTIFIED BY 'secret';
```

```
CREATE USER 'Entraîneur'@'localhost' IDENTIFIED BY 'entraîne';  
GRANT ALL PRIVILEGES ON lyonpalme.sortie TO 'Entraîneur'@'localhost';  
GRANT ALL PRIVILEGES ON lyonpalme.competition TO 'Entraîneur'@'localhost';
```

```
CREATE USER 'Coach'@'localhost' IDENTIFIED BY 'coach';  
GRANT ALL PRIVILEGES ON lyonpalme.entrainement TO 'Coach'@'localhost';
```

```
CREATE USER 'Materiel'@'localhost' IDENTIFIED BY 'mate';  
GRANT ALL PRIVILEGES ON lyonpalme.materiel TO 'Materiel'@'localhost';  
GRANT ALL PRIVILEGES ON lyonpalme.calendrier TO 'Materiel'@'localhost';  
GRANT ALL PRIVILEGES ON lyonpalme.pret TO 'Materiel'@'localhost';
```

BO1 : Insertion des données pour le jeu de tests

```
use lyonpalme;  
  
insert into adherent  
values('ADH0001','Pierre','Jean','1992/05/20','2019/01/01',NULL,NULL,'pierrejean','pierre  
jeanmdp'),  
  
('ADH0002','Boam','Jérôme','1940/12/25','2006/04/02',NULL,NULL,'jboam','jboampwd'),
```



```
('ADH0003','Bonnet','Mélanie','1993/02/20','2015/09/30',NULL,NULL,'mbonnet','mbonne  
tpwd');
```

```
insert into entraineur (ADHNum)
```

```
values
```

```
('ADH0001'),
```

```
('ADH0002');
```

```
insert into coach (ADHNum)
```

```
values('ADH0002');
```

```
insert into competition (COMPNum, COMPLibelle, COMPFinInscription)
```

```
values
```

```
('COMP0001','Demi-finale Rhône 2020','2020/06/30'),
```

```
('COMP0002','Amicale inter-région 2020','2020/09/30');
```

```
insert into entrainement
```

```
values
```

```
('ENT0001','2020/06/30','ADH0002'),
```

```
('ENT0002','2020/07/01','ADH0002');
```

```
insert into distance (metres_distance)
```

```
values
```

```
('1000 m'),
```

```
('3000 m');
```

```
insert into modalite (MODnum, modlibelle)
```

```
values
```

```
('MOD0001','relais'),
```

```
('MOD0002','mono-palme'),  
('MOD0003','bi-palme');
```

```
insert into proposermod (COMPNum, MODNum)  
values('COMP0001','MOD0001'),  
('COMP0001','MOD0002');
```

```
insert into proposerdist (COMPNum, METRES_DISTANCE)  
values  
('COMP0001','1000'),  
('COMP0001','3000');
```

```
insert into participercomp (ADHNum, COMPNum, PARTModaliteChoisie,  
PARTDistanceChoisie, PARTBesoinCovoiturage, PARTProposerCovoiturage,  
PARTNbPlacesVoiture, PARTHebergement, PARTNbPersonneAccompagnee)  
values  
('ADH0002','COMP0001','MOD0003','DIST0002',0,1,4,0,0),  
('ADH0003','COMP0002','MOD0002','DIST0001',0,1,4,1,0);
```

```
insert into materiel (MATNum, MATLibelle, MATMarque, MATTaille)  
values('MAT0001','Monopalme souple','Palmes',43),  
('MAT0002','Monopalme','Palmes',43);
```

```
insert into sortie (SORTIENum, SORTIEHeureDebut, SORTIELieu, SORTIEPlage,  
SORTIENiveauPublic, SORTIEMentionObligatoire, SORTIEMentionHiver, ADHNum)  
values  
('SOR0001','10:00','Lac de Miribel','Plage ouest','Confirmé','Bonnet de couleur et  
bouée de signalisation obligatoire.',NULL,'ADH0001');
```

```
insert into seance (SEANCENum, SEANCEDate, SEANCEContenu, ENTRAINEMENTNum)
```

values

```
('SEA0001','2020/06/08','Séance de nage','ENT0001'),  
('SEA0002','2020/06/10','Séance d\'apnée','ENT0002');
```

insert into faire (ADHNum, SEANCENum, FAIREIndispo, FAIREProposerEchanger)

values

```
('ADH0001','SEA0001',NULL,NULL),  
('ADH0002','SEA0001',NULL,NULL);
```

insert into s_inscrire (ADHNum, SORTIENum)

values

```
('ADH0001','SOR0001'),  
('ADH0002','SOR0001');
```

insert into calendrier (DatePretMateriel)

```
values ('2020/01/01'),  
('2020/02/01');
```

insert into preter (MATNum, DatePretMateriel, PRETERDate, ADHNum)

values

```
('MAT0001','2020/01/01',NULL,'ADH0003'),  
('MAT0002','2020/02/01','2020/06/04','ADH0002');
```

BO1 : DBInterface

```
<?php
```

```
// MARIA DB
```

```
abstract class Modele {
```

```
    private $bdd;
```

```
    protected function executerRequete($sql, $params = null) {
```

```
        if ($params == null) {
```

```
            $resultat = $this->getBdd()->query($sql);
```

```
        }
```

```
        else {
```

```
            $resultat = $this->getBdd()->prepare($sql);
```

```
            $resultat->execute($params);
```

```
        }
```

```
        return $resultat;
```

```
    }
```

```
    private function getBdd() {
```

```
        if ($this->bdd == null) {
```

```
            $this->bdd = new PDO('mysql:host=localhost;dbname=lyonpalme;charset=utf8',
```

```
                'Administrateur', 'admin',
```

```
                array(PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION));
```

```
        }
```

```
        return $this->bdd; }}
```

BO2 : Ajout d'une photo pour les adhérents

```
ALTER TABLE adherent  
ADD ADHPhoto VARCHAR(255);
```

BO2 : Don des privileges

```
CREATE USER 'Adherent'@'localhost' IDENTIFIED BY 'adh';  
GRANT select, update ON lyonpalme.ADHERENT TO 'Adherent'@'localhost';  
GRANT ALL PRIVILEGES ON lyonpalme.adherent TO 'Secrétaire'@'localhost';  
GRANT ALL PRIVILEGES ON lyonpalme.entraineur TO 'Secrétaire'@'localhost';  
GRANT ALL PRIVILEGES ON lyonpalme.coach TO 'Secrétaire'@'localhost';
```

BO2 : DAOAdherent

```
public class Adherent {  
    private string num;  
    private string nom;  
    private string prenom;  
    private date datenaissance;  
    private date datecertmedical;  
    private string fonctionbureau;  
    private string autrefonction;  
    private string id;  
    private string mdp;  
  
    public Adherent(){}
```

```

    public Adherent(string num, string nom, string prenom, date datenaissance, date
datecertmedical, string fonctionbureau, string autrefonction, string id, string mdp){

        this.num = num;

        this.nom = nom;

        this.prenom = prenom;

        this.datenaissance = datenaissance;

        this.datecertmedical = datecertmedical;

        this.fonctionbureau = fonctionbureau;

        this.autrefonction = autrefonction;

        this.id = id;

        this.mdp = mdp;

    }


    public long getNum() {

        return num;

    }


    public void setNum(string num) {

        this.num = num;

    }


    public string getNom() {

        return nom;

    }


    public void setNom(string nom) {

        this.nom = nom;

    }

```

```
public string getPrenom() {  
    return prenom;  
}
```

```
public void setPrenom(string prenom) {  
    this.prenom = prenom;  
}
```

```
public date getDateNaissance() {  
    return datenaissance;  
}
```

```
public void setDateNaissance(date datenaissance) {  
    this.datenaissance = datenaissance;  
}
```

```
public date getDateCertMedical() {  
    return datecertmedical;  
}
```

```
public void setDateCertMedical(date datecertmedical) {  
    this.datecertmedical = datecertmedical;  
}
```

```
public string getFonctionBureau() {  
    return fonctionbureau;  
}
```

```
public void setFonctionBureau(string fonctionbureau) {
```

```
        this.fonctionbureau = fonctionbureau;
    }

    public String getAutreFonction() {
        return autrefonction;
    }

    public void setAutreFonction(String autrefonction) {
        this.autrefonction = autrefonction;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getMdp() {
        return mdp;
    }

    public void setMdp(String mdp) {
        this.mdp = mdp;
    }
}
```


BO2 : Adherent en PHP

```
class Adherent
{
    private string $_num;
    private string $_nom;
    private string $_prenom;
    private date $_datenaissance;
    private date $_datecertificatmedical;
    private string $_id;
    private string $_mdp;

    public function __get($property) {
        if (property_exists($this, $property)) {
            return $this->$property;
        }
    }

    public function __set($property, $value) {
        if (property_exists($this, $property)) {
            $this->$property = $value;
        }
    }
}

class Entraîneur extends Adherent
{
    private string $_bureau;
```

```

private string $_autre;

public function __get($property) {
    if (property_exists($this, $property)) {
        return $this->$property;
    }
}

public function __set($property, $value) {
    if (property_exists($this, $property)) {
        $this->$property = $value;
    }
}

class Coach extends Entraîneur
{

}

```

BO3 : Classe Coach

```

class Coach extends Adherent
{
    private string $_num;

    public function __get($property) {
        if (property_exists($this, $property)) {

```

```

        return $this->$property;
    }
}

public function __set($property, $value) {
    if (property_exists($this, $property)) {
        $this->$property = $value;
    }
}

```

BO3 : Classe entraîneur

```

class Entraîneur extends Coach
{
    private string $_num;

    public function __get($property) {
        if (property_exists($this, $property)) {
            return $this->$property;
        }
    }

    public function __set($property, $value) {
        if (property_exists($this, $property)) {
            $this->$property = $value;
        }
    }
}

```