

Bases de données et Web

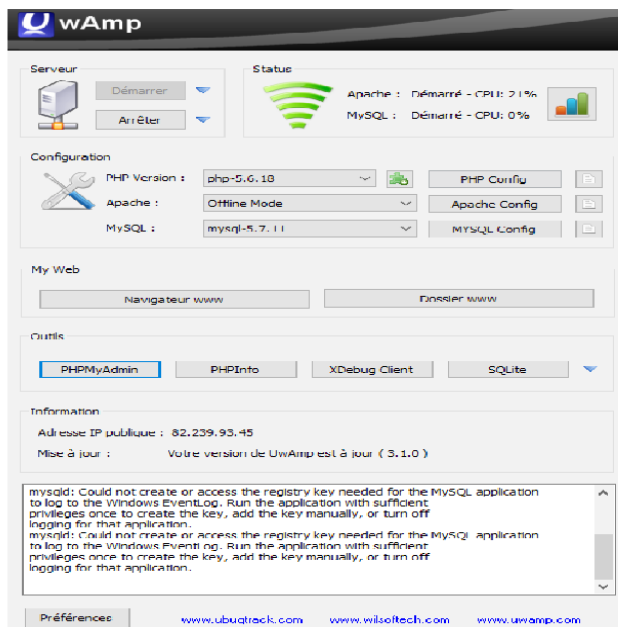
Correction TP3 : MySQL via PHPmyAdmin

Exercice 0

Si le UwAmp (ou un autre serveur web Apache, MySQL, PHP) n'est pas installé sur votre machine

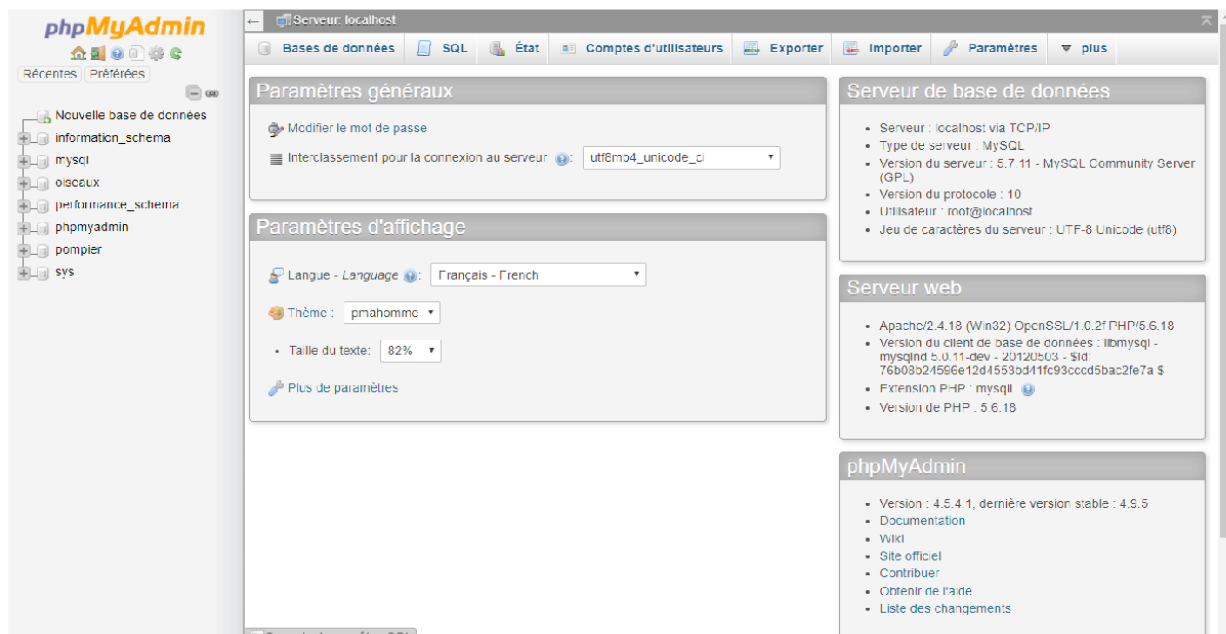
Télécharger UwAmp à partir du lien : http://cours.djoudi.online.fr/dwnld.php?pg=45&li=GNU_GPL
Double cliquer sur le fichier pour l'installation : par défaut, l'installation crée un dossier d'installation sur C:\UwAmp contenant un dossier **www** où déposer vos sites et scripts

Lancer UwAmp à partir du menu démarrer ou par un double cliquer sur son raccourci au bureau.



L'accès au serveur MySQL pour la création et la manipulation des bases de données se fait via le bouton PhPMyAdmin et la connexion avec les données :

- Utilisateur : root
- Mot de passe : root



Exercice 1 : Création de la base de données et des tables

Sur le serveur WAMP et via l'interface PHPMyAdmin, créer la base de données DBoiseau contenant les tables suivantes : Utilisateur, Oiseau, Amis et Oiseau_Utilisateur.

La table Utilisateur

```
create table Utilisateur
(
    pseudo varchar(100) primary key,
    pass varchar(100),
    nom varchar(100) not null,
    prenom varchar(100) not null,
    dateNaissance date not null,
    photo varchar(500)
);
```

La table Oiseau

```
create table Oiseau
(
    nomScientifique varchar(100) primary key,
    nomCommun varchar(50) not null,
    photo varchar(500)
);
```

La table Amis

```
create table Amis
(
    utilisateur1 varchar(100) not null,
    utilisateur2 varchar(100) not null,
    foreign key (utilisateur1) references Utilisateur(pseudo),
    foreign key (utilisateur2) references Utilisateur(pseudo),
    primary key (utilisateur1, utilisateur2)
);
```

La table Oiseaux_Utilisateur

```
create table Oiseau_Utilisateur
(
    idUtilisateur varchar(100) not null,
```

```

idOiseau varchar(100) not null,
foreign key (idUtilisateur) references Utilisateur(pseudo),
foreign key (idOiseau) references Oiseau(nomScientifique),
primary key (idUtilisateur, idOiseau)
);

```

Exercice 2 : Peuplement des tables

Peuplez vos tables afin d'avoir les données suivantes.

Données de la table Utilisateur

```

INSERT INTO Utilisateur (pseudo,pass,nom,prenom,dateNaissance, photo)
VALUES ("Jordan","UNW52ADR5YV","Strickland","Doris","10/07/03", "jordan.png"),
("Prescott","BGW03WTC2PD","Buckner","Xandra","04/02/03", "prescott.png"),
("Drew","FGI08JBD4GV","Holt","Erasmus","08/11/06", "drew.png"),
("Nomlanga","NCN51BLZ0JJ","Bowen","Martin","06/11/01", NULL),
("Graiden","WEP15UPI0PW","Branch","Kitra","05/11/05", "graiden.png"),
("Irene","VHY25OSM1LQ","Barber","Meredith","05/08/04", "irene.png"),
("Lamar","EWL09XNL9IP","Jones","Amy","10/09/08", NULL),
("Jameson","PHN78GPO2RJ","Kirby","Flavia","04/03/01", NULL),
("Leonard","TDL30WTO3XO","Cash","Damon","07/02/03", NULL),
("Wylie","JEA41ZH03KF","Adkins","Colorado","03/06/04", "wylie.jpg");

```

Données de la table Oiseaux

```

INSERT INTO Oiseau (nomScientifique, nomCommun, photo)
VALUES ("Vultur gryphus", "condors des andes", "condor_des_andes.jpg"),
("Cathartes burrovianus", "urubu à tête jaune", "urubu_tete_jaune.jpg"),
("Coragyps atratus", "urubu à tête noire", "urubu_tete_noire.jpg"),
("Terathopius ecaudatus", "aigle bateleur des savanes", "aigle_bateleur_savane.jpg"),
("Lophaetus occipitalis", "aigle huppard", "agle_huppard.jpg"),
("Melierax canorus", "vautour chanteur", "vautour_chanteur.jpg"),
("Aquila nipalensis", "aigle des steppes", "aigle_steppes.jpg"),
("Busarrelus nigricollis", "buse à tête blanche", "buse_pecheuse.jpg"),
("Milvus migrans", "milan noir", "milan_noir.jpg"),
("Sarcoramphus papa", "vautour pape", "vautour_pape.jpg");

```

Données de la table Amis

```

INSERT INTO Amis (utilisateur1, utilisateur2)
VALUES ("Jordan", "Prescott"),
("Jordan", "Drew"),

```

```
("Jordan", "Irene"),
("Irene", "Lamar"),
("Irene", "Wylie"),
("Graidon", "Prescott"),
("Graidon", "Jameson"),
("Graidon", "Leonard"),
("Wylie", "Jordan"),
("Wylie", "Jameson");
```

Données de la table Oiseaux_Utilisateur

```
INSERT INTO Oiseau_Utilisateur (idUtilisateur, idOiseau)
VALUES ("Jordan", "Vultur gryphus"),
("Jordan", "Cathartes burrovianus"),
("Jordan", "Terathopius ecaudatus"),
("Irene", "Aquila nipalensis"),
("Irene", "Lophaetus occipitalis"),
("Graidon", "Sarcoramphus papa"),
("Graidon", "Aquila nipalensis"),
("Wylie", "Milvus migrans"),
("Wylie", "Cathartes burrovianus"),
("Wylie", "Terathopius ecaudatus");
```

Exercice 3 : Interrogation de la base de données

Toutes les requêtes sont à écrire dans un fichier unique que vous nommerez "requetes.sql".

```
-- 1) Liste des utilisateurs
-- En langage SQL, écrivez une requête qui permet de retourner la liste des
utilisateurs (pseudo, nom, prenom et photo).
SELECT `pseudo`, `nom`, `prenom`, `photo`
FROM `utilisateur`
```

```
-- 2) Liste des oiseaux
-- En langage SQL, écrivez une requête qui permet de retourner la liste des oiseaux
(nomScientifique et nom commun et photo).
SELECT `nomScientifique`, `nomCommun`, `photo`
FROM `oiseau`
```

OU

```
SELECT *
FROM oiseau;
```

```
-- 3) Liste des oiseaux commençant par la lettre A
-- En langage SQL, écrivez une requête qui permet de retourner la liste des oiseaux
dont les noms scientifiques commencent par la lettre 'A'

SELECT nomScientifique, nomCommun, photo
FROM oiseau
WHERE nomScientifique LIKE 'A%';
```

OU

```
SELECT *
FROM oiseau
WHERE nomScientifique LIKE 'A%';
```

```
-- 4) Liste des utilisateurs qui ont une photo
-- En langage SQL, écrivez une requête qui permet de retourner la liste des
utilisateurs qui ont une photo.

SELECT pseudo, nom, prenom
FROM utilisateur
WHERE photo IS NOT NULL;
```

```
-- 5) Liste des oiseaux d'Irene
-- En langage SQL, écrivez une requête qui permet de retourner la liste des oiseaux
de l'utilisateur dont le pseudo est "Irene".

SELECT `nomScientifique` `photo`
FROM `oiseau`, `oiseau_utilisateur`, `utilisateur`
WHERE `oiseau`.`nomScientifique` = `oiseau_utilisateur`.`idOiseau`
AND `oiseau_utilisateur`.`idUtilisateur` = `utilisateur`.`pseudo`
AND `utilisateur`.`pseudo` = 'Irene'
```

OU

```
SELECT nomScientifique, photo
FROM oiseau_utilisateur JOIN oiseau ON oiseau_utilisateur.idOiseau =
oiseau.nomScientifique
WHERE idUtilisateur = 'Irene';
```

OU

```
SELECT nomScientifique, photo
FROM oiseau
WHERE nomScientifique
IN (
```

```
SELECT idOiseau
FROM oiseau_utilisateur
WHERE idUtilisateur= 'Irene'
);
```

```
-- 6) Liste des utilisateurs qui ont un vautour pape
-- En langage SQL, écrivez une requête qui permet de retourner la liste des
utilisateurs (pseudo, nom, prenom et photo) qui possèdent un vautour pape.
SELECT `pseudo`, `nom`, `prenom`, `utilisateur`.`photo`
FROM `utilisateur`, `oiseau_utilisateur`, `oiseau`
WHERE `utilisateur`.`pseudo`= `oiseau_utilisateur`.`idUtilisateur`
AND `oiseau`.`nomScientifique`= `oiseau_utilisateur`.`idOiseau`
AND `oiseau`.`nomCommun`= 'vautour pape';
```

OU

```
SELECT U.pseudo, U.nom, U.prenom, U.photo
FROM utilisateur U JOIN oiseau_utilisateur OU ON (OU.idUtilisateur = U.pseudo) JOIN
oiseau O ON (Ou.idOiseau = O.nomScientifique)
WHERE O.nomCommun = 'vautour pape';
```

OU

```
SELECT pseudo, nom, prenom, photo
FROM utilisateur
WHERE pseudo
    IN (
        SELECT idUtilisateur
        FROM oiseau_utilisateur
        WHERE idOiseau
            IN (
                SELECT nomScientifique
                FROM oiseau
                WHERE nomCommun = 'vautour pape'
            )
    );
```

```
-- 7) Liste des amis de chaque utilisateur
-- En langage SQL, écrivez une requête qui permet de retourner la liste des amis
(nom, prenom et photo) de chaque utilisateur (nom, prenom et photo).
SELECT DISTINCT U1.nom, U1.prenom, U1.photo, U2.nom, U2.prenom, U2.photo
FROM utilisateur U1 JOIN Amis A1 ON (U1.pseudo = A1.utilisateur1) JOIN utilisateur U2
ON (U2.pseudo = A1.utilisateur2)
UNION
```

```
SELECT DISTINCT U3.nom, U3.prenom, U3.photo, U4.nom, U4.prenom, U4.photo
FROM utilisateur U3 JOIN Amis A2 ON (U3.pseudo = A2.utilisateur2) JOIN utilisateur U4
ON (U4.pseudo = A2.utilisateur1);
```

```
-- 8) Liste des amis d'Irene
-- En langage SQL, écrivez une requête qui permet de retourner la liste des amis
(pseudo, nom et prenom) de l'utilisateur dont le pseudo est "Irene".
```

```
SELECT pseudo, nom, prenom, photo
FROM utilisateur JOIN amis on utilisateur.pseudo=amis.utilisateur1
WHERE utilisateur2="Irene"
UNION
SELECT pseudo, nom, prenom, photo
FROM utilisateur JOIN amis on utilisateur.pseudo=amis.utilisateur2
WHERE utilisateur1="Irene" ;
```

OU

```
SELECT pseudo, nom, prenom, photo FROM utilisateur WHERE pseudo
    IN(
        SELECT utilisateur1
        FROM amis
        WHERE utilisateur1<>'Irene'
        AND utilisateur2='Irene'
    )
UNION
SELECT pseudo, nom, prenom, photo
FROM utilisateur
WHERE pseudo
    IN (
        SELECT utilisateur2
        FROM amis
        WHERE utilisateur1='Irene'
        AND utilisateur2<>'Irene'
    );
```

OU

```
SELECT U2.nom,U2.prenom,U2.photo,U2.pseudo
FROM Utilisateur U1 JOIN Amis A1 on(U1.pseudo=A1.utilisateur1) JOIN Utilisateur U2 on
(A1.utilisateur2=U2.pseudo)
WHERE u1.pseudo = "Irene"
UNION
SELECT U4.nom,U4.prenom,U4.photo,U4.pseudo
```

```
FROM Utilisateur U3 JOIN Amis A2 on(U3.pseudo=A2.utilisateur2) JOIN Utilisateur U4 on
(A2.utilisateur1=U4.pseudo)
WHERE u3.pseudo = "Irene";
```

```
-- 9) Liste des utilisateurs qui ont des amis
-- En langage SQL, écrivez une requête qui permet de retourner la liste des
utilisateurs (pseudo, nom et prenom) qui ont des amis.
SELECT pseudo, nom, prenom, photo
FROM utilisateur JOIN amis on utilisateur.pseudo=amis.utilisateur1
UNION
SELECT pseudo, nom, prenom, photo
FROM utilisateur JOIN amis on utilisateur.pseudo=amis.utilisateur2 ;
```

OU

```
SELECT pseudo,nom, prenom, photo
FROM utilisateur
WHERE pseudo
    IN (
        SELECT utilisateur2
        FROM amis
    )
UNION
SELECT pseudo, nom, prenom, photo
FROM utilisateur
WHERE pseudo
    IN (
        SELECT utilisateur1
        FROM amis
    );
```

```
-- 10) Liste des utilisateurs qui n'ont pas d'oiseaux
-- En langage SQL, écrivez une requête qui permet de retourner la liste des
utilisateurs (pseudo, nom et prénom) qui n'ont pas d'oiseaux.
SELECT pseudo, nom, prenom, photo
FROM utilisateur
WHERE pseudo
    NOT IN (
        SELECT idUtilisateur
        FROM oiseau_utilisateur
    );
```

```
-- 11) Liste des utilisateurs qui n'ont pas d'amis
-- En langage SQL, écrivez une requête qui permet de retourner la liste des
```


utilisateurs (pseudo, nom et prénom) qui n'ont pas d'amis.

```
SELECT pseudo,nom,prenom, photo
FROM Utilisateur
WHERE pseudo
      NOT IN(
          SELECT utilisateur1
          FROM amis
      )
AND pseudo
      NOT IN (
          SELECT utilisateur2
          FROM amis
      );
```

```
SELECT U.pseudo, U.nom, U.prenom, U.photo
FROM utilisateur U
WHERE U.pseudo
      NOT IN(
          SELECT U.pseudo
          FROM utilisateur U JOIN Amis A ON U.pseudo = A.utilisateur1
          UNION
          SELECT U.pseudo
          FROM utilisateur U JOIN Amis A ON U.pseudo = A.utilisateur2);
```

-- 12) Liste des oiseaux communs d'Irene et de Graiden

-- En langage SQL, écrivez une requête qui permet de retourner la liste des oiseaux communs (pseudo, nom, prenom, photo) à Irene et Graiden

```
SELECT nomScientifique, nomCommun, photo
FROM oiseau JOIN oiseau_utilisateur ON nomScientifique = idOiseau
WHERE idUtilisateur = "Irene"
AND idOiseau
      IN(
          SELECT idOiseau
          FROM oiseau_utilisateur
          WHERE idUtilisateur = "Graiden"
      );
```

-- 13) Liste des amis communs d'Irene de Drew

-- En langage SQL, écrivez une requête qui permet de retourner la liste des amis communs (pseudo, nom, prenom, photo) à Irene et Drew.

```
SELECT pseudo, nom, prenom, photo
FROM utilisateur U1 JOIN amis A1 on (U1.pseudo=A1.utilisateur2)
```

```

WHERE A1.utilisateur1 = "Irene"
AND A1.utilisateur2
    IN (
        SELECT U2.pseudo
        FROM utilisateur U2 JOIN amis A2 ON U2.pseudo=A2.utilisateur2
        WHERE A2.utilisateur1 = "Drew"
        UNION
        SELECT U3.pseudo
        FROM utilisateur U3 JOIN amis A3 ON U3.pseudo=A3.utilisateur1
        WHERE A3.utilisateur2 = "Drew"
    )
UNION
SELECT pseudo, nom, prenom, photo
FROM utilisateur U1 JOIN amis A1 on (U1.pseudo=A1.utilisateur1)
WHERE A1.utilisateur2 = "Irene"
AND A1.utilisateur1
    IN (
        SELECT U2.pseudo
        FROM utilisateur U2 JOIN amis A2 ON U2.pseudo=A2.utilisateur2
        WHERE A2.utilisateur1 = "Drew"
        UNION
        SELECT U3.pseudo
        FROM utilisateur U3 JOIN amis A3 ON U3.pseudo=A3.utilisateur1
        WHERE A3.utilisateur2 = "Drew"
    );

```

OU

```

SELECT U.pseudo, U.nom, U.prenom, U.photo
FROM utilisateur U JOIN Amis A ON (U.pseudo = A.utilisateur2)
WHERE utilisateur1 = "Irene"
AND U.pseudo
    IN(
        SELECT U.pseudo
        FROM utilisateur U JOIN Amis A ON (U.pseudo = A.utilisateur2)
        WHERE utilisateur1 = "Drew"
        UNION
        SELECT U.pseudo
        FROM utilisateur U JOIN Amis A ON (U.pseudo = A.utilisateur1)
        WHERE utilisateur2 = "Drew"
    )
UNION
SELECT U.pseudo, U.nom, U.prenom, U.photo

```

```

FROM utilisateur U JOIN Amis A ON (U.pseudo = A.utilisateur1)
WHERE utilisateur2 = "Irene"
AND U.pseudo
    IN(
        SELECT U.pseudo
        FROM utilisateur U JOIN Amis A ON (U.pseudo = A.utilisateur2)
        WHERE utilisateur1 = "Drew"
        UNION
        SELECT U.pseudo
        FROM utilisateur U JOIN Amis A ON (U.pseudo = A.utilisateur1)
        WHERE utilisateur2 = "Drew"
    );

```

```

-- 14) Pour chaque utilisateur qui ont des amis, le nombre de ses amis
-- En langage SQL, écrivez une requête qui permet de retourner pour chaque
utilisateur (pseudo, nom, prenom, photo) le nombre de ses amis
SELECT U1.pseudo, U1.nom, U1.prenom, COUNT(A1.utilisateur2)
FROM Utilisateur U1 JOIN Amis A1 ON U1.pseudo=A1.utilisateur1
GROUP BY U1.pseudo
UNION
SELECT U2.pseudo, U2.nom, U2.prenom, COUNT(A2.utilisateur1)
FROM Utilisateur U2 JOIN Amis A2 ON U2.pseudo = A2.utilisateur2
GROUP BY U2.pseudo

```