

```
--table poste
create table poste(
    idPoste int(5) AUTO_INCREMENT,
    emplacementX VARCHAR(100),
    emplacementY VARCHAR(100),
    description VARCHAR(200),
    titre VARCHAR(100),
    datePoste DATE,
    etat VARCHAR(10),
    PRIMARY KEY(idPoste)
)
```

```
--table Utilisateur
create table utilisateur(
    login VARCHAR(50) PRIMARY KEY,
    email VARCHAR(200),
    motDePasse VARCHAR(250),
    photoDeProfil VARCHAR(250)
)
```

```
create table listeAffichable(
    idliste int(5) AUTO_INCREMENT,
    nomListe VARCHAR(50),
    descriptionListe VARCHAR(200),
    PRIMARY KEY (idListe)
)
```

```
--table Photo
create table photo(
    idPhoto int(5) AUTO_INCREMENT,
    url VARCHAR(250),
    PRIMARY KEY(idPhoto)
```

```

)

--table Partage

create table partage(
loginUtilisateur VARCHAR(50),
idPoste int(5),
PRIMARY KEY (loginUtilisateur,idPoste),
CONSTRAINT login_utilisateur_foreign_key FOREIGN KEY (loginUtilisateur) REFERENCES Utilisateur
(login),
CONSTRAINT idPoste_Foreign_key FOREIGN KEY (idPoste) REFERENCES poste(idPoste)
)

--table AVotePour

create table AVotePour(
loginUtilisateur VARCHAR(50),
idPoste int(5),
PRIMARY KEY (loginUtilisateur,idPoste),
CONSTRAINT loginuser_vote_foreign_key FOREIGN KEY (loginUtilisateur) REFERENCES Utilisateur
(login),
CONSTRAINT idpostvote_Foreign_key FOREIGN KEY (idPoste) REFERENCES poste(idPoste)
)

--table Contient

create table Contient(
idListe int(5),
idPoste int(5),
PRIMARY KEY (idListe,idPoste),
CONSTRAINT idListContient_foreign_key FOREIGN KEY (idListe) REFERENCES listeAffichable(idListe),
CONSTRAINT idpostContient_Foreign_key FOREIGN KEY (idPoste) REFERENCES poste(idPoste)
)

--table listeEnregistrées

create table ListeEnregistrées(
idListe int(5),
loginUtilisateur VARCHAR(50),
PRIMARY KEY (idListe,loginUtilisateur),

```

```

CONSTRAINT idListSave_foreign_key FOREIGN KEY (idListe) REFERENCES listeAffichable(idListe),
CONSTRAINT loginuserListSave_foreign_key FOREIGN KEY (loginUtilisateur) REFERENCES Utilisateur
(login)
)
create table administrateur(
login VARCHAR(50),
email VARCHAR(200),
motDePasse VARCHAR(250),
primary key(login)
);
--Insert (test)

--Postes

insert into Poste values('1','10','10','Photo de la tour de pise','Tour de Pise',
str_to_date('28,12,2021','%d,%m,%Y'),'Invalide');

insert into Poste values('2','87','32','Gare MontParnasse','Gare MontParnasse',
str_to_date('28,12,2021','%d,%m,%Y'),'Valide');

--Photo

insert into Photo values('1','Lien image tour de pise');
insert into Photo values('2','Lien imageGare MontParnasse');

--Utilisateur

insert into utilisateur values('PierreDu78000','pierre@gmail.com','1234','Lien photo de profil');
insert into utilisateur values('JacobFromTheStreet','JacobFromTheStreet@gmail.com','azerty','Lien
photo de profil');

--Liste affichable

insert into listeAffichable values('1','Les beaux monuments','Liste contenant de beaux monuments');

```

--AVotéPour

```
insert into avotepour values('PierreDu78000','2');
```

--Partage

```
insert into partage values('PierreDu78000','1');
```

--listeEnregistrées

```
insert into LISTEENREGISTRÉES values('1','JacobFromTheStreet');
```

--Contient

```
insert into Contient VALUES('1','1');
```

```
insert into Contient VALUES('1','2');
```

### Quelques requêtes utiles pour notre projet :

#### **Comment voir les liste enregistrées par un utilisateur ?**

```
select * from utilisateur inner join LISTEENREGISTRÉES on  
utilisateur.login=LISTEENREGISTRÉES.LOGINUTILISATEUR where utilisateur.login like  
'JacobFromTheStreet';
```

#### **Vérifier l'état valide ou non d'un poste**

```
select * from poste where etat like 'Valide';
```

```
select * from poste where etat like 'Invalide';
```

OU

```
select * from poste where etat = 'Valide';
```

```
select * from poste where etat = 'Invalide';
```

#### **Récupérer les poste dans une liste(donnée par son id)**

```
select * from poste inner join contient on poste.idposte = contient.idposte where idliste=1;
```

### **Voir les postes liké par un utilisateur**

```
select * from poste inner join AVotePour on poste.idPoste = AVotePour.idPoste inner join utilisateur  
on avotepour.loginUtilisateur=utilisateur.login where utilisateur.login like 'PierreDu78000';
```

### **Faire en sorte qu'un administrateur valide un poste (surement avec un bouton)**

```
UPDATE poste SET poste.etat = 'Valide' WHERE idPoste = 2;
```

### **INSERER UN POSTE**

-- Lors de l'insertion d'un poste, suite à sa création:

```
INSERT INTO `poste`(`emplacementX`, `emplacementY`, `description`, `titre`, `datePoste`, `etat`)  
VALUES ('10','10','Voici un chateau', 'Le chateau très  
beau',str_to_date('11/01/2022','%d,%m,%Y'),'Invalide')
```