



Projet Bases de données Relationnelles

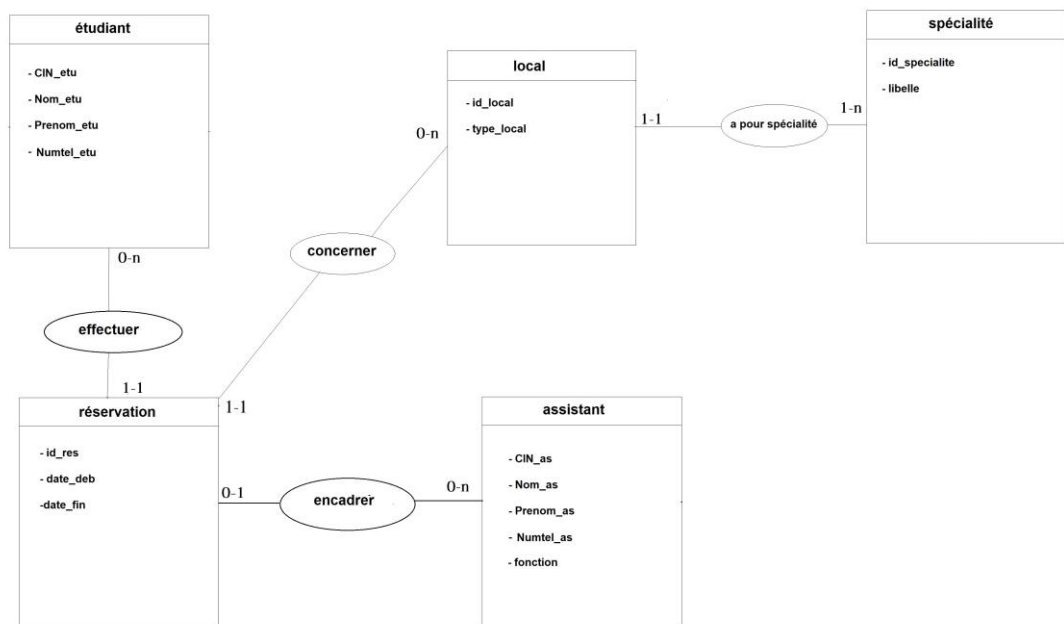
Bouhlal el mahdi

Aseds ine1

+ Descriptions des besoins :

Pour faciliter la vie étudiante dans l'INPT, le bureau des étudiant en partenariat avec l'administration veulent automatiser la réservation des terrains de sport et des salles de révision. Chaque réservation est identifiée par un numéro unique, Une date de début et une date de fin. Une réservation concerne un et un seul local qui est caractérisé par un numéro unique et un type (un terrain ou une salle) et qui peut être réservé plusieurs fois dans le temps, ce local a une spécialité c.-à-d. qu'il peut être un terrain de foot ou une salle de cours etc.. . Une réservation peut être effectuer par un et un seul étudiant qui est lui-même identifié par son CIN, nom, prénom, numéro de téléphone et qui peut effectuer plusieurs réservations. Lorsque l'étudiant effectue une réservation il peut demander un assistant qui peut être un coach ou un professeur, ça dépend de la réservation s'elle concerne un terrain ou une salle, un assistant est identifié par son CIN, nom, prénom, numéro de téléphone et une fonction (coach ou prof).

+ Modèle conceptuel de données(MCD) :



Modèle Logique de données(MLD) :

- Etudiant (CIN-etu , Nom_etu, Prenom_etu , Numtel_etu)
- Réservation (id_res, date_deb, date_fin, #CIN_etu, #id-local, #CIN_assitant)
- Assistant(CIN_as , Nom_as, Prenom_as , Numtel_as, fonction)
- Local(id_local, type_local, #id_specialite)
- Spécialité(id_specialite, libelle)

Code SQL utilisé :

Création de la base de données :

Pour créer la base de données :

- Create database **inptreservation** ;

Pour créer les tables :

- create table etudiant(CIN_etu varchar(15) not null PRIMARY KEY , Nom_etu varchar(30) not null , Prenom_etu varchar(30) not null , Numtel_etu varchar(20)) ;

- create table assistant(CIN_as varchar(15) not null PRIMARY KEY ,
Nom_as varchar(30) not null , Prenom_as varchar(30) not null ,
Numtel_as varchar(20), fonction varchar(20)) ;
- create table specialite(id_specialite int not null PRIMARY KEY , libelle
varchar(30)) ;
- create table local(id_local varchar(10) not null PRIMARY KEY ,
type_local varchar(30) not null , id_specialite int not null , foreign
key(id_specialite) references specialite(id_specialite)) ;
- create table reservation(id_res int not null PRIMARY KEY
AUTO_INCREMENT , date_deb datetime not null , date_fin
datetime not null , CIN_etu varchar(15) not null, CIN_as varchar(15),
id_local varchar(10) not null , foreign key(CIN_etu) references
etudiant(CIN_etu) , foreign key(CIN_as) references assistant(CIN_as)
, foreign key(id_local) references local(id_local)) ;

Peuplement de base de données :

D'abord j'ai créé deux utilisateur le premier 'admin' et le deuxième 'etudiant' :

- create user 'admin'@'localhost' identified with mysql_native_password
by 'admin12' ;
- create user 'etudiant'@'localhost' identified with
mysql_native_password by 'etudiant12' ;

Après j'ai donné à l'utilisateur 'admin' des privilèges comme suit :

- GRANT CREATE, ALTER, DROP, INSERT, UPDATE, DELETE, SELECT, REFERENCES, RELOAD on *.* TO 'admin'@'localhost' WITH GRANT OPTION;

Pour peupler la base de données je vais se connecter en tant qu'un admin en utilisant :

- SYSTEM mysql -u admin -p

Les requêtes pour insérer des données de test :

- insert into etudiant values ('wa11224','bouhlal','el mahdi','0611223344'), ('wa11556','nabli','yassin','0611223355'), ('wa11223','belghiti','oussama','0611223399') ;
- insert into assistant values ('t1112','badri','khalid','0611221100', 'coach'), ('t1142','benhamou','said','0611554411', 'professeur') ;
- insert into specialite values (1, 'terrain de foot') , (2, 'terrain de basketball') , (3, 'terrain de volleyball'),(4, 'salle de cours') , (5, 'salle de tp') , (6, 'amphi') ;
- insert into local values ('F1', 'Terrain' , 1) , ('F2', 'Terrain' , 1) , ('V1', 'Terrain' , 3) , ('B119', 'Salle' , 4),('Ba1', 'Terrain' ,2) , ('E1', 'Salle' , 6), ('D23', 'Salle', 5) ;

Avant d'ajouter les réservations je vais ajouter des triggers pour contrôler l'ajout des réservations :

- Trigger pour vérifier si le local est disponible :

```
DELIMITER //  
CREATE TRIGGER validate_date  
BEFORE INSERT  
ON reservation  
FOR EACH ROW  
IF (select count(*) from reservation where id_local=New.id_local and  
((date_deb<= NEW.date_deb and NEW.date_deb<=date_fin) or  
(date_deb<=NEW.date_fin and NEW.date_fin <= date_fin)) ) >=1  
THEN  
SIGNAL SQLSTATE '45000'  
SET MESSAGE_TEXT= 'le local est déjà réservé pour cette date.' ;  
END IF //  
DELIMITER ;
```

- **Trigger pour vérifier si l'assistant est disponible :**

```
DELIMITER //  
CREATE TRIGGER validate_assistant  
BEFORE INSERT  
ON reservation  
FOR EACH ROW  
IF (select count(*) from reservation where CIN_as=New.CIN_as and  
((date_deb<=NEW.date_deb and NEW.date_deb<=date_fin) or  
(date_deb<=NEW.date_fin and NEW.date_fin <= date_fin)) ) >=1  
THEN  
SIGNAL SQLSTATE '45000'  
SET MESSAGE_TEXT= 'l'assistant n'est pas disponible pour cette  
date.' ;  
END IF //  
DELIMITER ;
```

- **Trigger pour vérifier si l'étudiant existe dans la base de données :**

```
DELIMITER //  
CREATE TRIGGER validate_etudiant  
BEFORE INSERT  
ON reservation  
FOR EACH ROW  
IF (select count(*) from etudiant where CIN_etu=NEW.CIN_etu) =0  
THEN  
SIGNAL SQLSTATE '45000'  
SET MESSAGE_TEXT= 'pas d'informations sur l'étudiant concerné !  
Veuillez contactez l'administration.' ;  
END IF //  
DELIMITER ;
```

- **Trigger pour vérifier si le local existe dans la base de données :**

```
DELIMITER //  
CREATE TRIGGER validate_local  
BEFORE INSERT  
ON reservation  
FOR EACH ROW  
IF (select count(*) from local where id_local=NEW.id_local) =0  
THEN  
SIGNAL SQLSTATE '45000'  
SET MESSAGE_TEXT= 'ce local n'existe pas !' ;  
END IF //  
DELIMITER ;
```

- **Trigger pour vérifier si l'assistant existe dans la base de données :**

```
DELIMITER //  
CREATE TRIGGER validate_assistantExiste  
BEFORE INSERT  
ON reservation
```

```
FOR EACH ROW
IF (select count(*) from assistant where CIN_as=NEW.CIN_as) =0
THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT= 'cet assistant n'existe pas !';
END IF //
DELIMITER ;
```

**Maintenant je vais donner à l'utilisateur 'etudiant' le droit d'ajouter ,
afficher et supprimer des réservations :**

- grant insert , select , delete on inptreservation.reservation to
'etudiant'@'localhost' ;

**Après je vais se connecter au tant qu'un étudiant pour ajouter des
réservations :**

- insert into reservation(date_deb,date_fin,CIN_etu,CIN_as,id_local)
values ('2022-11-15 15:00:00' , '2022-11-15
17:00:00','wa11224','t1112','V1') ;
- insert into reservation(date_deb,date_fin,CIN_etu,CIN_as,id_local)
values ('2022-11-15 16:00:00' , '2022-11-15
18:00:00','wa11223','t1142','F1') ;