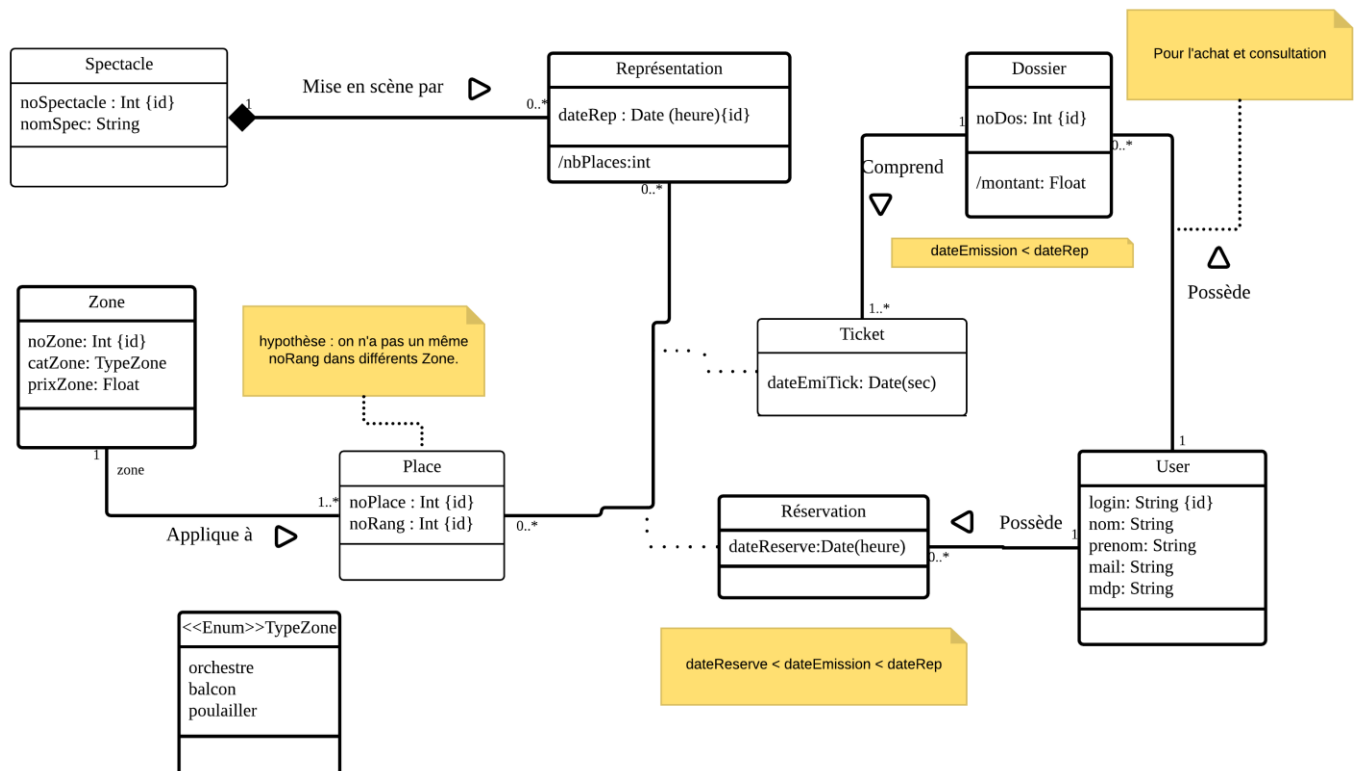


Pour finir les deux fonctionnalités de la partie 4, on va aller changer le code SQL DDL ainsi le diagramme de classe. Ici on va juste s'attaquer à « Achats en ligne » et à « Réservations ».

Dans la partie 4 « gérer des réservations », on a fait juste la consultation, l'addition et la suppression de réservations.

UML :



Le schéma relationnel (ajouté) (servir à la partie 4 uniquement):

LesUsers (login, nom, prenom, mail, mdp)

$\forall \langle l, n, p, m, w \rangle \in LesUsers \Leftrightarrow$ Le nom n et le prénom p d'un utilisateur est identifié par son login l , son adresse mail est m et le mot de passe de ce login est w . */

LesReservations(noSpec, dateRep, noPlace, noRang, dateReserve, login)

$\forall \langle s, d, p, r, e, l \rangle \in LesReservations \Leftrightarrow$ la réservation correspondant à la place $\langle p, r \rangle$ pour la représentation $\langle s, d \rangle$, a été fait à la date e par l'utilisateur login u . */

Le schéma relationnel (changé) :

LesDossiers_base(noDos, login)

$\forall \langle d, l \rangle \in LesDossiers_base \Leftrightarrow$ le dossier identifié par un numéro de dossier d concerne

*(appartient à) l'utilisateur l. */*

Les vues :

LesDossiers(noDos, login, montant)

*/ * <d, m, l> ∈ LesDossiers ⇔ m est la somme des prix des places des tickets associées au dossier d qui concerne l'utilisateur l. */*

- LesDossiers[login] = LesUsers[login]
- LesReservations[login] = LesUsers[login]
- LesReservations[noSpec,dateRep] ⊆ LesReprésentations_base[noSpec, dateRep]
- LesReservations[noPlace,noRang] ⊆ LesPlaces[noSpec, dateRep]

SQL DDL :

Nouveau:

```
create table LesUsers(  
    login varchar (50) not null,  
    nom varchar (50) not null,  
    prenom varchar (50) not null,  
    mail varchar (50) not null,  
    mdp varchar (50) not null,  
    constraint pk_User_log primary key (login)  
);  
  
create table LesReservations(  
    noSpec number (4,0),  
    dateRep date,  
    noPlace number (3,0),  
    noRang number (3,0),  
    dateReserve date,  
    login varchar (50) not null,  
    constraint pk_resr_place_rep unique (noSpec,dateRep,noPlace,noRang),  
    constraint fk_resr_numS_dateR foreign key (noSpec, dateRep) references  
LesReprésentations_base(noSpec, dateRep),  
    constraint fk_resr_noP_noR foreign key (noPlace, noRang) references LesPlaces  
(noPlace,noRang),  
    constraint fk_resr_login foreign key (login) references LesUsers (login),  
    constraint ck_dates check (dateReserve < dateRep)  
);
```

Changement:

```
create table LesDossiers_base (  
    noDos number (3,0),  
    login varchar (50) not null,
```

```
constraint pk_dos_noD primary key (noDos),
constraint fk_dos_login foreign key (login) references LesUsers(login)
);
create view LesDossiers (noDos,login, montant) as
  with X as (select noDos, prixZone from lesZones natural join LesPlaces natural join
LesTickets natural join LesDossiers_base)
  select noDos, login,sum(prixZone) as montant from X group by noDos;
```