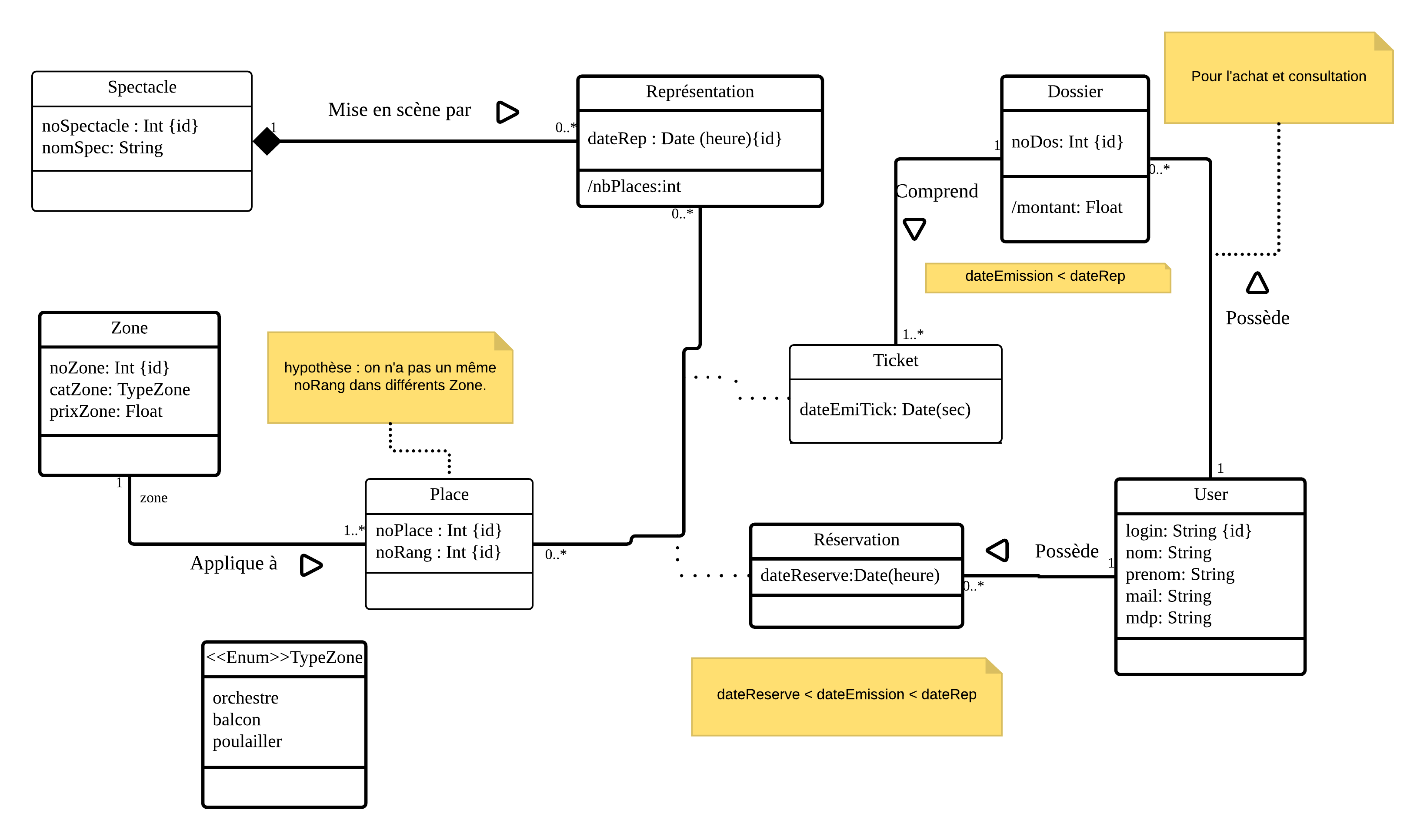
Pour finir les deuxs fonctionnalités de la partie 4, on va aller changer le code SQL DDL ainsi le diagramme de classe. Ici on va juste s’attaquer à « Achats en ligne » et à « Réservations ».

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)

/\* *< l , n, p ,m , w> ∈ LesUsers ⇔ 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)

/\* *< s, d, p, r, e, l > ∈ LesReservations ⇔ la réservation correspondant à la place <p, r> pour la représentation <s,d>, a été fait à la date e par l’utilisateur login u. \*/*

Le schéma relationnel (changé) :

LesDossiers\_base( noDos, login)

/\* *<d, l> ∈ LesDossiers base ⇐⇒ 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 LesRepresentations\_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;