

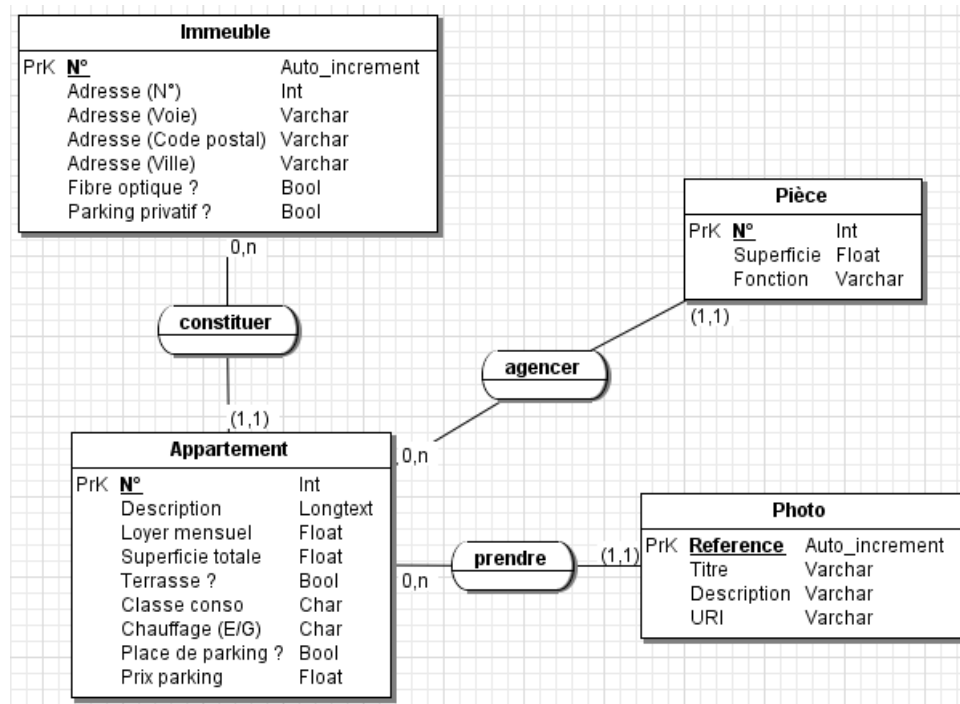
## TP N°5 : SQL et triggers

**Thème :** à la découverte des triggers

### Exercice 1 : parc immobilier (exemple)

Sujet :

Soit le modèle de données suivant :



**Immeuble**(id, adrNum, adrVoie, adrCodePostal, adrVille, fibreOptique, parkingPrivatif)

*Clef primaire :* id

**Appartement**(#immeuble, num, description, loyer, superficie, terrasse, classeConso, chauffage, placeParking, prixParking)

*Clef primaire :* immeuble, num

*Clef étrangère :* immeuble en référence à Immeuble(id)

**Pièce**(#(immeuble, appartement), num, superficie, fonction)

*Clef primaire :* immeuble, appartement, num

*Clefs étrangères :* (immeuble, appartement) en référence à Appartement(immeuble, num)

**Photo**(num, titre, description, uri, #(immeuble, appartement))

*Clef primaire :* num

*Clef étrangère :* (immeuble, appartement) en référence à Appartement(immeuble, num)

En voici le script de création de tables :

```

CREATE TABLE Immeuble(
  Id INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,
  adrNum VARCHAR(7) NOT NULL, adrVoie VARCHAR(100) NOT NULL,
  adrCodePostal VARCHAR(5) NOT NULL, adrVille VARCHAR(30) NOT NULL,
  fibreOptique TINYINT NOT NULL, parkingPrivatif TINYINT NOT NULL
);

CREATE TABLE Appartement(
  immeuble INT(11), num INT(3) NOT NULL, description LONGTEXT,
  loyer DOUBLE NOT NULL, superficie DOUBLE NOT NULL,
  terrasse TINYINT(1) NOT NULL, classeConso CHAR(1) NOT NULL, chauffage CHAR(1) NOT NULL,
  placeParking TINYINT(1) NOT NULL, prixParking DOUBLE,
  CONSTRAINT pk_appartement PRIMARY KEY (immeuble, num),
  CONSTRAINT fk_immeuble FOREIGN KEY (immeuble) REFERENCES Immeuble(id)
);

CREATE TABLE Photo(
  immeuble INT(11), appartement INT(3), reference INT(11) NOT NULL,
  titre VARCHAR(75), description VARCHAR(255), uri VARCHAR(120) NOT NULL,
  CONSTRAINT pk_photo PRIMARY KEY (immeuble, appartement, reference),
  CONSTRAINT fk_appartement_photo
    FOREIGN KEY (immeuble, appartement) REFERENCES Appartement(immeuble, num)
);

CREATE TABLE Piece(
  immeuble INT(11), appartement INT(3), num INT(2) NOT NULL,
  superficie DOUBLE ,
  fonction VARCHAR(30),
  CONSTRAINT pk_piece PRIMARY KEY (immeuble, appartement, num),
  CONSTRAINT fk_appartement_piece
    FOREIGN KEY (immeuble, appartement) REFERENCES Appartement(immeuble, num)
);

```

Créer ses tables dans une base de données sous MySQL.

Questions :

1. Rédiger le trigger permettant de vérifier la contrainte suivante : le prix de la place de parking d'un appartement peut et doit être NULL si l'appartement ne possède pas de place de parking. Tester le bon fonctionnement de votre trigger.
2. On souhaite que la contrainte suivante soit vérifiée : la superficie totale d'un appartement doit être égale à la somme de la superficie de chacune de ses pièces. Pour ce faire, créer le trigger qui permet de mettre à jour la superficie d'un appartement à l'insertion d'une pièce.
3. Adapter le trigger de la question 1 afin :
  - de vérifier la contrainte suivante : un appartement ne peut avoir de place de parking si l'immeuble n'a pas de parking privatif ;
  - d'initialiser la superficie de l'appartement à 0 à l'insertion d'un appartement ;

- d'empêcher la modification de la superficie d'un appartement en cas de mise à jour d'un appartement.

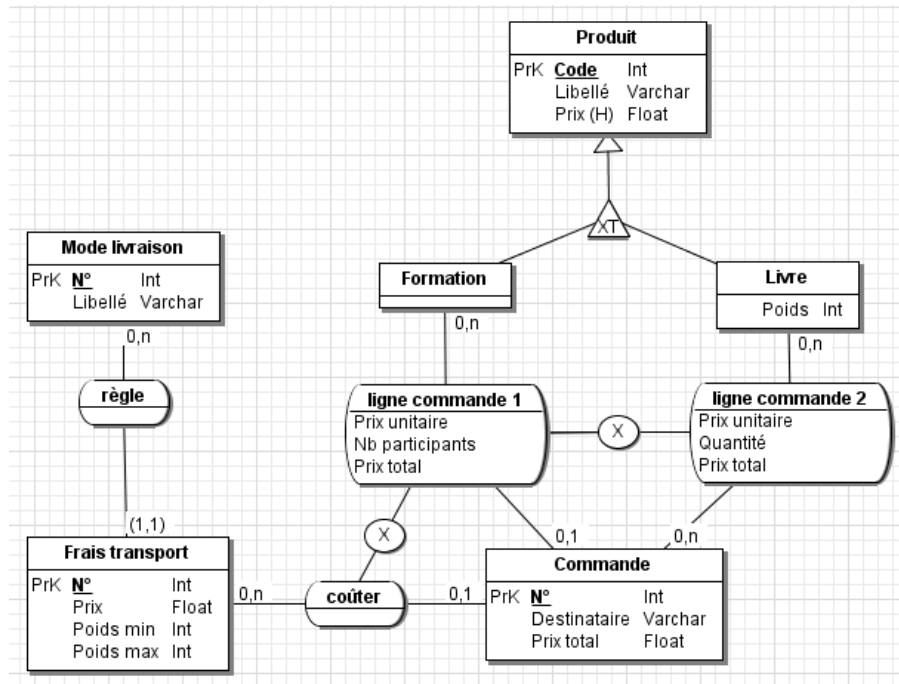
4. En vous inspirant du trigger de la question 2, rédiger celui qui permet de mettre à jour la superficie d'un appartement à la mise à jour d'une pièce. Rédiger également le trigger qui met à jour la superficie d'un appartement à la suppression d'une pièce.

*N.B. : si NEW permet de manipuler en lecture/écriture la ligne nouvellement insérée ou modifiée, OLD permet de manipuler la ligne avant modification ou suppression.*

**Exercice 2 : des produits au prix variant au fil du temps**

Sujet :

Soit le modèle de données suivant :



**Produit**(code, libelle, prix, formation, poids)

Clef primaire : code

Commentaire : le champ discrimination formation prend la valeur 1 (vrai) ou 0 (faux)

**Mode\_Livraison**(num, libelle)

Clef primaire : num

**Frais\_Transport**(#livraison, num, prix, poidsMin, poidsMax)

Clef primaire : livraison, num

Clefs étrangères : livraison en référence à Mode\_Livraison(num)

**Commande**(num, destinataire, prixTotal, #(livraison, transport))

Clef primaire : num

Clef étrangère : (livraison, transport) en référence à Frais\_Transport(livraison, num)

**Ligne\_Commande**(#commande, #produit, prixu, quantite, prix)

Clef primaire : commande, produit

Clef étrangères :

- commande en référence à Commande(num)

- produit en référence à Produit(code)

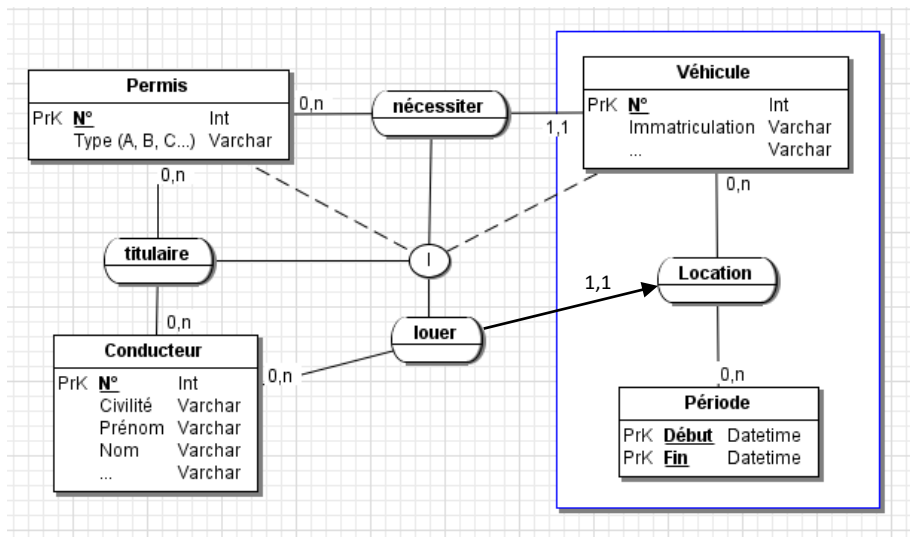
Questions :

1. Rédiger le script de création de table relatif à la base de données ci-avant décrite.
2. Rédiger le trigger permettant de vérifier la contrainte suivante : si une commande porte sur une formation, alors elle ne peut porter sur un ou plusieurs livres.
3. Rédiger le trigger permettant de vérifier la contrainte réciproque : si une commande porte sur un ou plusieurs livres, alors elle ne peut porter sur une formation.
4. Adapter le trigger de la question 2 ou 3 afin de vérifier la contrainte suivante : si la commande porte sur une formation, alors il ne doit pas y avoir de frais de transport.
5. Adapter le trigger de la question 2 ou 3 afin que le prix total d'une commande soit automatiquement mis à jour à lorsqu'une ligne de commande est insérée.
6. Préciser les cas que nous n'avons pas traités au travers des questions 1 à 5.

### Exercice 3 : location de véhicules

Sujet :

Soit le modèle de données suivant :



Questions :

1. Créer sous MySQL une base de données correspondant au modèle de données ci-dessus (fournir le script de création de tables).
2. Rédiger le trigger qui permet, à l'insertion et à la modification d'une Location (INSERT ou UPDATE sur la table « Location »), de vérifier que le conducteur spécifié correspond à un conducteur ayant parmi ses permis celui requis pour conduire le véhicule.