Exercices sur l'algèbre relationnelle

Soit une base de données conçue par une agence immobilière lui permettant de gérer les appartements dont elle a la responsabilité :

Immeuble	NomImmeuble	Adresse	NBEtages	AnnéeConstruction	NomGerant
	Koudalou	3 rue Blanche	15	1975	Doug
	Barabas	2 Allée Nikos	2	1973	Ross

Appart	NomImmeuble	NoAppart	Superficie	Etage
	Koudalou	1	150	14
	Koudalou	34	50	15
	Koudalou	51	200	2
	Koudalou	52	50	5
	Barabas	1	250	1
	Barabas	2	250	2

Personne	**Nom**	**Age**	**Profession**
	Ross	51	Informaticien
	Alice	34	Cadre
	Rachel	23	Stagiaire
	William	52	Acteur
	Doug	34	Rentier

Occupant	**NomImmeuble**	**NoAppart**	**NomOccupant**	**AnneeArrivée**
	Koudalou	1	Rachel	2012
	Barabas	1	Doug	2014
	Barabas	2	Ross	2014
	Koudalou	51	William	2016
	Koudalou	34	Alice	2013

Pour simplifier, nous faisons les hypothèses suivantes :

- Deux personnes ont forcément des noms différents
- Deux immeubles ont forcément des noms différents
- Deux appartements situés dans le même immeuble ont forcément des numéros (NoAppart) différents.
- Un appart ne peut être occupé que par une personne
- Une personne ne peut occuper qu'un seul appartement

Un enregistrement dans la table Immeuble, Appart ou Personne décrit un immeuble, un appartement et une personne. Un enregistrement dans la table Occupant dit que telle personne (NomOccupant) occupe tel appartement (NomImmeuble et NoAppart). Cette même personne occupe cet appartement depuis une certaine Année (Année d'arrivée).

1. Pour chacune des requêtes suivantes exprimées en algèbre relationnelle, si elle est correcte alors donner son résultat, sinon expliquer pourquoi elle n'est pas correcte.
    a. $\pi_{NomImmeuble}(Appart)$
    b. $\pi_{NomImmeuble,AnnéeArrivée}(Occupant)$
    c. $\sigma_{NomImmeuble=Barabas}(Appart)$
    d. $Immeuble\ \times Occupant$
    e. $Immeuble\ \cap Appart$
    f. $Immeuble \bowtie Personne$
    g. $\pi_{Nom}(Personne) - \pi_{Nom}(\rho_{NomGerant \to Nom}(Immeuble))$
    h. $\pi_{NomOccupant}(\sigma_{NomImmeuble=Barabas}(Occupant))$
    i. $\sigma_{NomImmeuble=Barabas}(\pi_{NomOccupant}(Occupant))$
    j. $\pi_{NomOccupant}(Occupant \bowtie \sigma_{Superficie>100}(Appart))$

2. Pour chacune des requêtes correctes ci-dessus, donner un sens à son résultat (expliquer que représente son résultat).

3. Exprimer chacune des requêtes suivantes en utilisant l'algèbre relationnelle :

    a. Afficher le nom du gérant de l'immeuble qui s'appelle Koudalou
    Principe : On sélectionne les lignes (et il n'y aura qu'une car il n'y a qu'un immeuble qui peut s'appeler comme ça. Cf. les hypothèses qu'on a considérées) de la table Immeuble qui correspondent à Koudalou puis on projette le résultat sur NomGérant
    $$\pi_{NomGerant}\left(\sigma_{NomImmeuble=Koudalou}(Immeuble)\right)$$
    b. Afficher la profession du gérant de Koudalou.
    Principe : la profession est enregistrée dans la table Personne ⇒ On doit utiliser cette table. Pour savoir qui est le gérant de Koudalou, on doit utiliser Immeuble. Conclusion : on doit combiner/composer ces deux tables. On doit donc utiliser soit un produit soit une jointure. On note qu'il n'y a pas d'attribut commun à ces deux tables. Donc jointure ou produit, le résultat sera le même.
    $$Immeuble\ \times Personne$$
    Une fois qu'on a effectué le produit, on aura chaque personne qui sera associée à tous les immeubles (et inversement). Pour ne garder que les bonnes associations : chaque personne n'est associée qu'aux immeubles qu'elle gère (noter qu'on ne dit pas « l'immeuble qu'elle gère » car rien n'interdit qu'une personne puisse gérer plusieurs immeubles), il faut éliminer les associations où le nom de la personne est différent du nom du gérant
    $$\sigma_{Nom=NomGerant}(Immeuble\ \times Personne)$$
    Chaque immeuble est maintenant associé à la personne qui le gère. Noter que du coup, les personnes qui ne gèrent aucun immeuble ne figurent plus dans le résultat. Nous ne sommes intéressés que par Koudalou. On ajoute donc une sélection
    $$\sigma_{NomImmeuble=Koudalou}(\sigma_{Nom=NomGerant}(Immeuble\ \times Personne))$$
    Noter que l'expression ci-dessus est équivalente à
    $$\sigma_{Nom=NomGerant\ ET\ NomImmeuble=Koudalou}(Immeuble\ \times Personne))$$
    On se retrouve avec un enregistrement où l'on a tous les attributs qui décrivent Koudalou (NomImmeuble, AnnéeConstruction …) ainsi que ce qui décrit la personne qui le gère (Nom, âge, Profession). Il ne reste plus qu'à projeter cet enregistrement sur la profession
    $$\pi_{Profession}(\sigma_{NomImmeuble=Koudalou}(\sigma_{Nom=NomGerant}(Immeuble\ \times Personne)))$$

c. Afficher la profession du gérant de l'immeuble où Rachel habite.
Principe : Pour savoir où Rachel habite, on a besoin de Occupant. Pour savoir qui gère quoi, on a besoin de la table Immeuble. Pour savoir la profession de quelqu'un, on a besoin de la table Personne ⇒ On a besoin de combiner les 3 tables. Noter qu'il n'existe pas d'opérateur ternaire. On aura donc forcément besoin de deux opérations pour combiner les 3 tables. Pour s'assurer que chaque occupant soit associé à l'immeuble où il/elle habite, il suffit de réaliser la jointure

$$Occupant \bowtie Immeuble$$

Cette jointure se fait sur la base de l'attribut commun NomImmeuble. Le résultat de cette jointure n'a aucun attribut commun avec la table Personne. Donc on peut utiliser soit le produit soit la jointure

$$Personne \times (Occupant \bowtie Immeuble)$$

Ce faisant, chaque personne est associée à toutes les lignes résultant de la jointure. Il faut supprimer les « mauvaises associations »

$$\sigma_{Nom=NomGerant}(Personne \times (Occupant \bowtie Immeuble))$$

Maintenant, chaque occupant est associé à son immeuble lequel est associé à la personne qui le gère. On est intéressé par juste l'occupant Rachel, donc sélection

$$\sigma_{Nom=NomGerant\ ET\ NomOccupant=Rachel}(Personne \times (Occupant \bowtie Immeuble))$$

Il ne reste plus qu'à projeter sur Profession

$$\pi_{Profession}(\sigma_{Nom=NomGerant\ ET\ NomOccupant=Rachel}(Personne \times (Occupant \bowtie Immeuble)))$$

Noter que dès le départ, on aurait pu se « focaliser » sur Rachel et non pas tous les occupants. Ca aurait donné la requête

$$\pi_{Profession}(\sigma_{Nom=NomGerant}\ (Personne \times (\sigma_{NomOccupant=Rachel}(Occupant) \bowtie Immeuble)))$$

Les deux requêtes sont équivalentes.

d. Afficher la superficie de l'appartement occupé par Rachel ainsi que la profession du gérant de l'immeuble où elle habite.
La différence par rapport à la question précédente c'est qu'ici nous avons en plus besoin de la superficie qui se trouve dans la table Appart ⇒ On l'ajoute à l'ensemble des tables à composer. La requête sera donc

$$\pi_{Profession,Superficie}(\sigma_{Nom=NomGerant}\ (Appart \bowtie (Personne \times (\sigma_{NomOccupant=Rachel}(Occupant) \bowtie Immeuble))))$$

La jointure qu'on vient d'ajouter portera à la fois sur l'attribut NomImmeuble et NoAppart

e. Afficher le numéro et le nom d'immeuble des appartements occupés.
Il suffit de projeter la table Occupant sur ces deux attributs

f. Afficher le numéro et le nom d'immeuble des appartements inoccupés.
Principe : C'est l'ensemble de TOUS les apparts MOINS ceux qui sont occupés. Projeter la table Appart sur NomImmeuble et NoAppart pour obtenir l'ensemble de tous les appartements décrits juste par ces deux attributs. Puis faire la différence avec la requête de la question f

$$\pi_{NomImmeuble,NoAppart}(Appart) - \pi_{NomImmeuble,NoAppart}(Occupant)$$

g. Afficher le nom de ou des immeubles qui ont tous leurs appartements occupés (sur la base qui est donnée, c'est clair que c'est Barabas).
Principe : c'est l'ensemble de tous les immeubles ayant des apparts en location MOINS ceux qui ont au moins un appart de libre. Pour obtenir ces derniers, il suffit de projeter le résultat de la requête f ci-dessus sur le nom de l'immeuble.

$$\pi_{NomImmeuble}(\pi_{NomImmeuble,NoAppart}(Appart) - \pi_{NomImmeuble,NoAppart}(Occupant))$$

Maintenant il ne reste plus qu'à exclure cet ensemble d'immeuble de l'ensemble de tous les immeubles.

$$\pi_{NomImmeuble}(Appart) - \pi_{NomImmeuble}(\pi_{NomImmeuble,NoAppart}(Appart) - \pi_{NomImmeuble,NoAppart}(Occupant))$$

A priori, on pourrait projeter Immeuble et non pas Appart. C'est subtil car on peut avoir des Immeubles qui n'ont aucun appart. Du coup, ces derniers apparaitraient dans le résultat. Ca pourrait prêter à confusion car certes AUCUN de leurs appart est inoccupé mais ils n'en ont pas d'apparts. C'était une parenthèse pour montrer le besoin d'être ultra précis quand on pose un exercice où on demande d'exprimer une requête. Le texte en français peut facilement ne pas être suffisamment précis pour qu'il puisse se prêter à différentes interprétations. Laissons ça pour les juristes et législateurs qui sont les spécialistes du domaine.

h. Donner le nom des occupants qui sont arrivés après Alice (exprimer cette requête sans utiliser la constante 2013 qui est l'année d'arrivée d'Alice).
Principe : Nous avons besoin de comparer la date d'arrivée de chaque occupant à celle d'Alice. Trouvons d'abord la date d'Alice

$$\pi_{AnneeArrivee}\left(\sigma_{NomOccupant=Alice}(Occupant)\right)$$

Cette année d'arrivée on voudrait la mettre à côté de CHAQUE occupant afin de comparer les 2 années. Pour ce faire, on peut penser au produit cartésien avec Occupant sauf que ce dernier n'est pas possible car AnnéeArrivée est présent dans les deux. On pourrait penser à la jointure mais à ce moment seuls les occupants qui sont arrivé la même année qu4alice seront préservés et en plus on se retrouve avec un seul attribut qui contient une année d'arrivée. La solution est donc de renommer pour pouvoir réaliser le produit

$$\rho_{AnneeArrivee \to AnneeAlice}(\pi_{AnneeArrivee}\left(\sigma_{NomOccupant=Alice}(Occupant)\right))$$

On fait le produit de ce résultat avec Occupant et on ne retient que ceux où AnnéeArrivée > AnnéeAlice

$$\sigma_{AnneeArrivee>AnneeAlice}(Occupant \times \rho_{AnneeArrivee \to AnneeAlice}(\pi_{AnneeArrivee}\left(\sigma_{NomOccupant=Alice}(Occupant)\right)))$$

Il ne reste plus qu'à projeter sur le nom de l'occupant

i. Donner les paires de noms d'occupants qui correspondent à des personnes habitant le même immeuble. La paire (Alice, William) en est un exemple car les deux habitent Koudalou.
Remarque : essayer d'éviter de retourner des paires inutiles, ex : (Alice, Alice) est inutile car toute personne habite forcément le même immeuble qu'elle-même et (ii) ne pas retourner à la fois (Alice, William) et (William, Alice) car il s'agit en fait de la même paire. Indication : On peut utiliser le comparateur < entre des chaines de caractère. Dans ce cas, c'est l'ordre lexicographique qui est pris en compte. Par exemple, Alice < William mais pas l'inverse.
Principe : Cette requête ressemble à la précédente dans le sens où la table Occupant devra être utilisée deux fois. La raison est qu'on a besoin de comparer des occupants à des occupants (dans h., on comparait des occupants à un occupant particulier qui est Alice). Dans un premier temps on fait en sorte à associer chaque occupant à TOUS les occupants qui habitent le MEME immeuble. Ceci revient à joindre la table Occupant avec elle-même mais l'attribut de jointure doit être juste l'attribut NomImmeuble. Pour ce faire, on change le nom des attributs NomOccupant, NoAppart et AnnéeArrivée dans Occupant puis on fait une jointure avec Occupant

$$\rho_{NomOccupant \to O, NoAppart \to N, AnneeArrivee \to A}(Occupant) \bowtie Occupant$$

Deux occupants se retrouvent dans le même enregistrement si et seulement ils habitent le même immeuble. Noter que le nom du premier est O et celui du second est NomOccupant. Si

on projette tout de suite sur O et NomOccupant, on aura des paires de noms. Entre autres, on aura la paire (Alice, Alice) car Alice habite bien le même immeuble qu’Alice. Pour éviter ce type de paires, il suffit d’ajouter une sélection avec O <> NomOccupant. Ceci va certes éliminer les paires où l’on a un seul nom mais ça va garder, entre autres les paires (Doug, Ross) et (Ross, Doug). Noter qu’il s’agit de la même paire même s’il s’agit d’enregistrements distincts : il ne faut pas compter sur le fait que l’algèbre élimine les doublons pour que l’une des deux soit enlevée. Pour ne garder que l’une des deux, il suffit d’ajouter la condition O< NomOccupant (ou bien O>NomOccupant). Ainsi, (Doug,Ross) sera gardée car Doug < Ross selon l’ordre lexicographique mais pas (Ross, Doug).
Donc

$$\pi_{O,NomOccupant}(\sigma_{O<NomOccupant}(\rho_{NomOccupant\rightarrow O,NoAppart\rightarrow N,AnneeArrivee\rightarrow A}(Occupant) \bowtie Occupant))$$

C’est normal de trouver que les requêtes g et i (h en moindre mesure) soient difficiles à trouver.

4. Soient R1(A,B) et R2(B,C) deux tables avec R1 qui contient m enregistrements et R2 qui en contient n. Quels sont les nombres minimal et maximal d’enregistrement que l’on va obtenir dans R1 ⋈ R2.
Min = 0 car aucune tuple de R1 n’a une valeur de B qui se trouve dans R2.
Max = n * m. Si dans R1 on a une seule valeur de B et cette valeur est l’unique valeur présente dans le B de R2. Dans ce cas, chaque enregistrement de R1 est joignable avec tous les enregistrements de R2.

5. Soient R1(A,B) et R2(A,B) deux relations. Montrer que R1 ∩ R2 = R1 ⋈ R2
Standard ☺