

TD4

Exercice 1

1. a) On utilise l'index sur formation (Diplome)

```
for id in I_farma_diplome.getRows('Master Info'):  
    f = lireTuple(id)  
    if f.region == 'IDF' and f.universite in ('Sorbonne', 'Dauphine'):  
        afficher f.NF, f.universite
```

op n°2 : access (diplome='Master Info')

op n°1 : filter (region='IDF' and universite in ('S', 'D'))

2. liste_id1 = I_Etu_NF.getRows('F1')

liste_id2 = I_Etu_Prestige.getRows('Paris')

IDS = liste_id1.intersect(liste_id2)

for id in IDS :

t = lireTuple(id)

if t.prenom like 'A%':

afficher t.prenom

← table access full

3. for e in Etudiants.types():

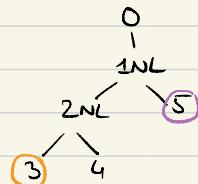
```
for id in I_farma_NF(e.NF)
```

f = lireTuple(id) ← table access by rowid

if f.diplome == 'licence'

affiche e.prenom, f.region

monique
get rowid



4. for r in $I_ferma_diplome$.getRawData('Master') :

$f = r$.tuple(r)

for r' in I_ETU_NF .getRawData(f.NF)

$e = tuple(r')$

affiche e.prenom, f.region

dans la table de hashage on

aura des données de la table

"étudiant", car c'est la table

Université

de gauche : / \

/ \

Université formation

5. $D = \{ \}$: clé : NF, valeur : liste de prénoms

for r in I_ETU_LESSON .getRawData('Aix') :

$e = tuple(r)$

if e.NF in D :

$D[e.NF].append(e.prenom)$

else :

$D[e.NF] = [e.prenom]$

for r in $I_FERMA_DISPLOME$.getRawData('M2') :

$f = tuple(r)$

for p in $D[f.NF]$:

afficher (p, f.université)

← on pourra aussi ajouter une condition si
f.NF est bien dans l'ensemble D, sinon on
aurait une erreur car il y a des étudiants de
M2 qui n'habitent pas à Aix

6. Projeter pour l'opération n°2 : jointure entre F et E

→ on garde f.université, e.prenom, e.NF

Projeter pour l'opération n°3 : f.université, f.NF

MJ

/ \
stage
/\
ferma Etudiant

Dictionnaires: D1 : clé : NF, valeur : université

D2 : clé : NF, valeur : (université, prenom)

} les valeurs sont celles de la
partie gauche du hashage

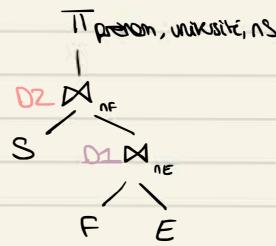
```

for f in Formation.tuples():
    D1[f.nF] = f.universite
for e in Etudiant.tuples():
    D2[e.nE] = (D1[e.nF], e.prenom)
for s in Stage.tuples():
    u, p = D2[s.nE]
    afficher (p, u, s.nS)

b) d'indice à droite:  $\Pi_{\text{prenom}, \text{universite}, nS} [$ 

```

b) d'indice à droite: $\Pi_{\text{prenom}, \text{universite}, nS} [$ Stage \bowtie (Formation \bowtie Etudiant)]
 (hash join reste à gauche)



D1: clé: nf, valeur: université

D2: clé: ne, valeur: liste de ns

car ne n'est pas unique dans stage,
 un étudiant peut faire plusieurs stages

```

for s in Stage.tuples():
    if s.ne in D2:
        D2[s.ne].append(s.ns)
    else:
        (D2[s.ne] = s.ns)

```

for f in Formation.tuples():

D1[f.nF] = f.universite

for e in Etudiant.tuples():

u = D1[e.nF]

for ns in D2[e.ne]:

afficher (e.prenom, u, ns)

D2.get(e.ne, [])

=> permet de ne pas avoir le pb de
 cle inexistant

Exercice 8

$$1. SF_1 = \frac{5}{20} = 0,25$$

$$SF_2 = \frac{4}{100} = 0,04$$

$$SF_3 = \frac{1}{1000} \times \frac{1}{8} = \frac{1}{8000}$$

$$SF_4 = \underbrace{(duree = 6 \text{ ou duree} = 12)}_{2 \text{ valeurs parmi } 4} \text{ ou } \underbrace{note > 5}_{15 \text{ valeurs parmi } 20}$$

$$= \frac{2}{4} + \frac{15}{20} - \frac{2}{4} \times \frac{15}{20}$$

$$= \frac{7}{8}$$

$$2. \text{ Card}(R_1) = SF(\sigma_{\text{mention} = rs}(\text{Etudiant})) \times SF(\sigma_{\text{region} = \text{def}}(\text{Formation})) \times \text{Card}(E \setminus \text{Formation})$$

$$= \frac{1}{4} \times \frac{1}{5} \times \text{Card}(\text{Etudiant})$$

$$= \frac{1}{20} \times 1000$$

$$= 50$$

autre méthode : $R_1 = T_1 \Delta T_2$

$$\text{avec } T_1 = \sigma_{\text{mention} = rs}(\text{Etudiant})$$

$$\text{Card}(T_1) = \frac{1}{4} \times 1000 = 250$$

$$\text{et } T_2 = \sigma_{\text{region} = \text{def}}(\text{Formation})$$

$$\text{Card}(T_2) = \frac{1}{5} \times 50 = 10$$

$$\begin{aligned} \text{Card}(T_1 \Delta T_2) &= \text{Card}(T_1) \times \text{Card}(T_2) \times \frac{1}{\text{nombre de nf}} \\ &= 250 \times 10 \times \frac{1}{50} \\ &= 50 \end{aligned}$$

$$\text{Card}(R_2) = \text{Card}(S \setminus F) = 2000$$

ou

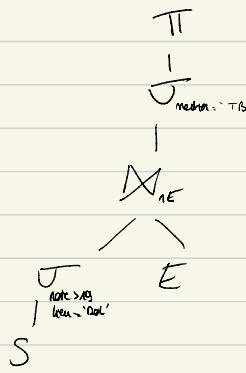
$$\begin{aligned} \text{Card}(R_2) &= \text{Card}((E \setminus S) \setminus_{nf} F) \\ &= \underline{\text{Card}(E \setminus S)} \times \text{Card}(F) \times \frac{1}{\text{nb de nf}} \\ &= \underline{\text{Card}(E) \times \text{Card}(S)} \times \frac{1}{\text{nbne}} \times \underline{\text{Card}(F) \times \text{Card}(F)} \\ &= \text{Card}(S) \\ &= 2000 \end{aligned}$$

(cardans E \ S c'est card(S) qui prime et dans S \ F c'est card(S) aussi mais à revoir plus en détail)

$$\begin{aligned}
 R3: \text{card}(N_3) &= \text{card}(\text{Storage}) \times \text{card}(\text{Prochaine ligne}) \times \frac{1}{\text{nb lignes}} \\
 &= 2000 \times 200 \times \frac{1}{1000} \\
 &= 600
 \end{aligned}$$

$$\begin{aligned}
 3. \text{Coût}(E \times_F F) &= \text{Coût}(E) + \text{card}(E) \times \text{card}(\sigma(F)) \\
 &= \text{P}(E) + \text{card}(E) \times \text{card}(F) \times \frac{1}{\text{nb lignes}} \\
 &= \text{card}(E)/10 + " \\
 &= 1000/10 + 1000 \times 50 \times \frac{1}{50} \\
 &= 100 + 1000
 \end{aligned}$$

4. a) P1:



$$\text{Coût}(T1) =$$

$$\text{Coût}(T1) = \frac{1}{20} \times 2000 + \frac{1}{100} \times 2000$$

$$\text{Coût}(\sigma_{\text{lignes} = \text{lignes}} \text{Storage}) = \text{Coût}(...)$$

$$= \text{card}(S) \times \frac{1}{\text{nb lignes}}$$

$$= 2000 \times \frac{1}{100}$$

$$= 20$$

$$\text{Coût}(T1) = 20 \times \frac{1}{20} = 1$$

$$\text{Coût}(T1 \times_E E) = \text{Coût}(T1) + \text{card}(T1) \times 1$$

$$= 1 + ...$$

Réponse 4:

$\text{Coût}(P1) = 21$
$\text{Coût}(P2) = 600$
$\text{Coût}(P3) = 120$