

**Examen M2 Pro Info, M2 Bibs, M2 MIAGe**  
**Année 2003-2004**  
**Durée : 2h30**

**Module de Base de Données : Implémentation**

**Lire attentivement avant de commencer à rédiger**

Aucun document n'est autorisé pendant l'examen. Vous avez la possibilité d'utiliser les notes manuscrites (écrites manuellement) et préparées individuellement (pas de photocopie) sur 1 (une) feuille cadrillée 21/27, recto/verso, écrite une ligne sur deux.

Vous devez répondre aux questions dans l'ordre des exercices. Il est formellement interdit de recopier l'énoncé sur votre copie.

**Exercice 1 : Index**

Considérez l'instance ci-dessous de la relation Pilote(Id, Nom, T-Avion, Nb-vol) stockée dans un **fichier trié** sur les valeurs de l'attribut Nb-Vol. Hypothèse : une page du fichier permet de stocker 2 enregistrements de l'instance ; chaque enregistrement est identifié par son rid.

Pilote	Id	Nom	T-Avion	Nb-vol
	831	Bace	A230	542
	674	Yah	A230	670
	424	Arty	B740	675
	178	Mala	A400	854
	532	Cick	B740	875
	243	Eit	B350	875
	231	Doik	A300	987

1. Pour chacun des cas ci-dessous, donnez la liste des entrées d'un index dont la structure est celle d'un arbre B+ :
  - (a) entrées de type 3, dense, non groupant de clé T-Avion
  - (b) entrées de type 2, dense, groupant de clé T-Avion
  - (c) entrées de type 2, dense, non groupant de clé Nb-Vol
  - (d) entrées de type 2, non dense, groupant de clé Nb-Vol
  - (e) entrées de type 2, non dense, non groupant de clé Nom

**Correction** Le rid d'un enregistrement est un couple (numéro page, emplacement enregistrement dans la page). Les entrées de type 2 sont de la forme (valeur de clé, rid). Les entrées de type 3 sont de la forme (valeur de clé, liste de rids). Dans tous les cas, la liste des entrées d'un index sont triées sur la valeur de la clé de recherche.

- (a) (A230 , [(p1,1), (p1,2)]) (A300 , [(p4,1)]) (A400 , [(p2,2)]) (B350 , [(p3,2)]) (B740 , [(p2,1), (p3,1)])
- (b) cet index ne peut être construit car le fichier est trié sur Nb-Vol

(c) cet index ne peut pas être non groupant (c'est correct de donner la liste des entrées et de faire la remarque)

(d) (542, (p1,1)) (675, (p2,1)) (875, (p3,1)) (987, (p4,1)).

(e) cet index ne peut être construit car un index non dense doit être groupant.

2. Dans le but d'optimiser les requêtes les plus fréquentes effectuées sur la relation Pilote, l'administrateur propose de créer les index suivants:

- Index 1 = arbre B+, type 2, dense de clé T-Avion,

- Index 2 = hachage, type 2, dense de clé Nom

- Index 3 = arbre B+, type 2, dense de clé Id

Pour plus d'efficacité, l'administrateur doit choisir parmi ces index celui ou ceux qui sera (seront) groupant(s). Quel sera son choix ? expliquez.

**Correction** Aucun des index ne peut être groupant puisque le fichier est trié sur Nb-Vol et qu'aucun des index proposés n'a pour clé de recherche Nb-Vol.

## Exercice 2 : Tri et jointure multi-fusion

Supposez que l'instance de la relation R est stockée dans un fichier de taille 20000 en nombre de pages et que, pour le tri de R, l'espace du buffer alloué est de 6 en nombre de cellules.

1. Expliquez le fonctionnement de l'algorithme de tri-multifusion dans ce cas particulier en donnant le nombre d'étapes, le nombre et la taille des fragments triés produits à chaque étape. Quel est le coût total du tri (sans compter l'écriture du résultat sur disque) en terme de transferts.

**Correction :** Le buffer ayant une capacité totale de 6 cellules, le tri-multifusion sera effectué en utilisant 5 cellules pour le chargement des pages du fichiers (B=5) et 1 cellule pour le résultat des fusions.

étape 1: Les 20000 pages sont traitées 5 par 5. Cela produit  $\lceil 20000/5 \rceil = 4000$  fragments de 5 pages chacun.

étape 2: Les 4000 fragments sont traités 5 par 5. Cela produit  $\lceil 4000/5 \rceil = 800$  fragments de 25 pages chacun.

étape 3: Les 800 fragments sont traités 5 par 5. Cela produit  $\lceil 800/5 \rceil = 160$  fragments de 125 pages chacun.

étape 4: Les 160 fragments sont traités 5 par 5. Cela produit  $\lceil 160/5 \rceil = 32$  fragments de 625 pages chacun.

étape 5: Les 32 fragments sont traités 5 par 5. Cela produit  $\lceil 32/5 \rceil = 7$  fragments dont 6 fragments de 3125 pages chacun et un fragment de 1250 pages.

étape 6: Les 7 fragments sont traités 5 par 5. Cela produit  $\lceil 7/5 \rceil = 2$  fragments dont 1 fragment de 15625 pages et un fragment de 4375 pages.

étape 7: Les 2 fragments sont fusionnés pour produire un fragment de 20000 pages triées ie le fichier trié.

Le coût d'une étape est la lecture et l'écriture de chaque page du fichier soit  $2 \times 20000$ .

coût du tri 5-fusion =  $6 \times (2 \times 20000) + 20000 = 260000$  E/S (sans compter l'écriture finale) ou  $7 \times (2 \times 20000) = 280000$  E/S (en comptant l'écriture finale).

2. Quel espace du buffer serait il nécessaire d'allouer pour trier le fichier stockant R en 2 étapes ? quel serait alors le coût du tri ?

**Correction :** Suivant la méthode précédente, pour que le tri multi-fusion des  $M=20000$  pages puisse se faire en 2 étapes il faut que le nombre  $B$  des cellules réservées au chargement des pages satisfasse les conditions suivantes :

étape 1: Les  $M$  pages sont traitées  $B$  par  $B$ , produisant  $\lceil M/B \rceil = F$  fragments de  $B$  pages chacun.

étape 2: Les  $F$  fragments sont traités  $B$  par  $B$ . Cela produit  $\lceil F/B \rceil = 1$  fragments de  $M = B \times B$  pages chacun.

Donc  $B = \lceil \sqrt{M} \rceil = 142$  et la capacité du buffer doit donc être de  $B + 1 = 143$  cellules.

Concernant le coût du tri fusion dans ce cas, il est suffisant de savoir que le nombre d'étapes est 2 (la valeur de  $B$  ne sert à rien):  $1 \times (2 \times 20000) + 20000 = 60000 E/S$  (sans compter l'écriture finale) ou  $2 \times (2 \times 20000) = 80000 E/S$  (en comptant l'écriture finale).

3. Considérez une relation  $S$  stockée dans un fichier de 150 pages. La jointure de  $R$  et  $S$  est effectuée par la méthode de tri-fusion. Quelle est le coût de cette jointure dans le cas non optimisé, dans le cas optimisé en terme de transferts (sans compter l'écriture du résultat de la jointure sur disque) ?

**Correction :** La jointure par tri-fusion consiste à trier chaque relation (par multi-fusion) puis à fusionner les relations triées. Il faut donc commencer par calculer le coût du tri 5-fusion de  $S$ . On peut reprendre le raisonnement fait pour  $R$ .

étape 1: Les 150 pages sont traitées 5 par 5. Cela produit  $\lceil 150/5 \rceil = 30$  fragments de 5 pages chacun.

étape 2: Les 30 fragments sont traités 5 par 5. Cela produit  $\lceil 30/5 \rceil = 6$  fragments de 25 pages chacun.

étape 3: Les 6 fragments sont traités 5 par 5. Cela produit  $\lceil 6/5 \rceil = 2$  fragments dont un de 125 pages et l'autre de 25 pages.

étape 4: Les 2 fragments sont fusionnés pour produire le fichier trié.

Le coût du tri fusion de  $S$  est de  $3 \times (2 \times 150) + 150 = 1050 E/S$  (sans compter l'écriture finale) ou  $4 \times (2 \times 150) = 1200 E/S$  (en comptant l'écriture finale).

Dans le cas "non optimisé" de la jointure par tri-fusion, le coût est la somme des coûts du tri de  $R$  et de  $S$  (en comptant l'écriture finale) et du coût de la fusion de  $R$  et de  $S$  (sans compter l'écriture du résultat comme d'habitude) :  $280000 + 1200 + 20000 + 150 = 301350 E/S$ .

Dans le cas "optimisé" de la jointure par tri-fusion, l'algorithme essaie (si c'est possible) d'économiser une étape du tri de  $R$  et une étape du tri de  $S$ . On remarque ici qu'après la 6ème étape du tri-fusion de  $R$ , on obtient 2 fragments et que de même après la 3ème étape du tri-fusion de  $S$ , on obtient 2 fragments. L'algorithme ne va donc pas trier  $R$  jusqu'au bout (de même pour  $S$ ) et va directement fusionner les 2 fragments de  $R$  avec les 2 fragments de  $S$ . Le coût de la jointure est dans ce cas égal à la somme des coûts des 6 étapes de tri fusion de  $R$  et des 3 étapes de tri fusion de  $S$  et du coût de la fusion de  $R$  et de  $S$  (sans compter l'écriture du résultat comme d'habitude) :  $240000 + 900 + 20000 + 150 = 260150 E/S$

**Commentaires :** Cet exercice peut (et doit) être fait sans utilisation de logarithme.

## Exercice 3 : Optimiseur

Considérez les schémas Acteur(ac-id, nom, cachet), Agence(ag-id, agence), Tournage(ag-id, ac-id, film) dont les clés respectives sont ac-id, ag-id et (ac-id, film). Considérez la requête SQL ci-dessous qui extrait les acteurs dont le cachet est supérieur à 100 euros/heure et qui ont été placés par l'agence "Univers" pour le tournage d'un film intitulé "retour" :

```

SELECT A.nom FROM Acteur A, Agence AG, Tournage T
WHERE A.cachet ≥ "100" AND AG.agence="Univers" AND T.film = "retour"
AND AG.ag-id = T.ag-id AND A.ac-id = T.ac-id ;

```

Les index suivants sont disponibles :

index	relation	clé	type entrée	groupant	structure	hauteur
IND1	Acteur	ac-id	2	non	B+	$h_1 = 2$
IND2	Acteur	cachet	2	oui	B+	$h_2 = 2$
IND3	Agence	(ag-id, agence)	1	oui	hachage	
IND4	Tournage	ag-id	2	oui	B+	$h_4 = 3$
IND5	Tournage	ac-id	2	non	B+	$h_5 = 3$
IND6	Tournage	film	2	non	hachage	

Les informations suivantes sont disponibles concernant les fichiers stockant les relations Acteur, Agence et Tournage:

Relations	page	fichier
Acteur	$n_1 = 25$	$M_1 = 10^3$
Agence	$n_2 = 40$	$M_2 = 50$
Tournage	$n_3 = 20$	$M_3 = 10^5$

page   :	taille d'une page en nombre de n-uplets
fichier   :	taille d'un fichier en nombre de pages

Un acteur sur 100 a un cachet supérieur à 100. Une centaine de films environ portent le titre "retour". Le nombre d'agences ayant un même nom est en moyenne d'une dizaine. On estime à 150 le nombre d'acteurs qui interviennent pendant le tournage d'un film.

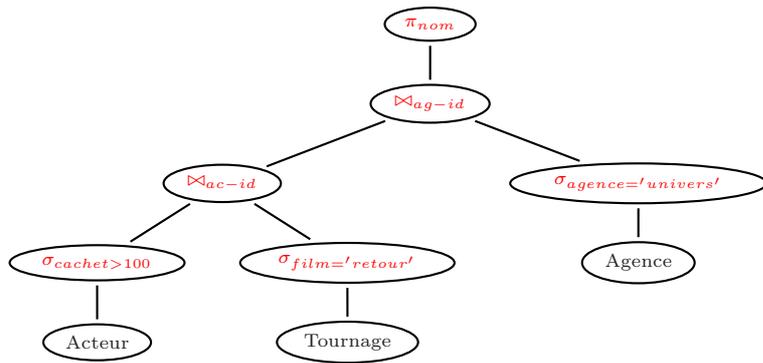
Pour les besoins de l'exercice vous pouvez si nécessaire précisez certaines informations statistiques qui manqueraient et posez des hypothèses (raisonnables) statistiques sur les données comme par exemple leur répartition uniforme, ...

Vous supposerez que les méthodes d'évaluation de la jointure connues du système sont la jointure par itération par bloc (le buffer est de taille 7) et la jointure par itération avec index.

**Attention :** décrivez (dessinez) avec soin en apportant tous les détails nécessaires les plans d'exécution demandés; pour l'estimation des coûts des plans, donnez les formules qui vous ont permis d'obtenir les résultats numériques (les résultats exclusivement numériques ne seront pas pris en compte).

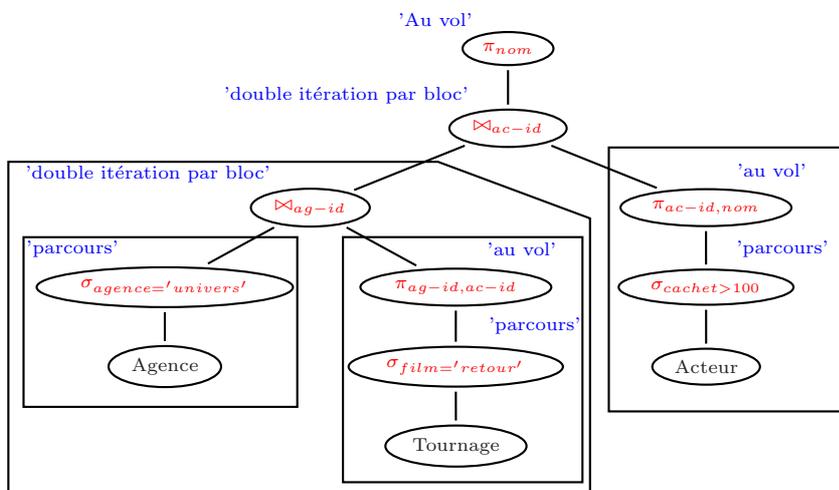
1. Donnez un arbre de requête associé à la requête ci-dessus.

**Correction :** Je donne un arbre possible; il y a bien sûr beaucoup d'autres solutions possibles.



2. Donnez un plan d'exécution de la requête n'utilisant aucun index et qui vous paraît optimal. Justifiez et donnez une estimation du coût de ce plan.

**Correction :**



Ce qui est "encadré" correspond à la matérialisation des résultats de sous requêtes et cette matérialisation engendre un coût d'écriture.

Analyse du coût du plan d'exécution : Cela nécessite d'estimer la taille des résultats intermédiaires (en particulier ceux qui sont matérialisés).

- (a) Estimation du coût  $C_1$  de la sous-requête  $E_1 = \sigma_{agence='Univers'}[Agence]$  et de sa matérialisation (i.e. estimation de la taille  $P_1$  du résultat de  $E_1$ ) :
- $C_1 = M_2 E/S$  (nombre de pages du fichier stockant la relation Agence)  
 Il y a 10 agences portant le nom 'Univers' donc  $P_1 = 1$  page.
- (b) Estimation du coût  $C_2$  de la sous-requête  $E_2 = \pi_{ag-id,ac-id}(\sigma_{film='retour'}[Tournage])$  et de sa matérialisation (i.e. estimation de la taille  $P_2$  du résultat de  $E_2$ ) :
- $C_2 = M_3 E/S$  (nombre de pages du fichier stockant la relation Tournage)

100 films portent le titre 'retour' et 150 acteurs participent (en moyenne) au tournage de chaque film donc l'évaluation de  $E_2$  produit environ 15000 n-uplets dont la taille est (approximativement) 2/3 de la taille des n-uplets de la relation Tournage (on va supposer qu'une page peut contenir 30 n-uplets résultat). Donc,  $P_2 = 500$  pages.

- (c) Estimation du coût  $C_3$  de la sous-requête  $E_3 = \Pi_{ac-id,Nom}(\sigma_{cachet>100}[Acteur])$  et de sa matérialisation (i.e. estimation de la taille  $P_3$  du résultat de  $E_3$ ) :

$C_3 = M_1 E/S$  (nombre de pages du fichier stockant la relation Acteur)

Un acteur sur 100 soit 250 acteurs au total ont un cachet > 100. L'évaluation de  $E_3$  produit 250 n-uplets dont la taille est (approximativement) 2/3 de la taille des n-uplets de la relation Acteur (on va supposer qu'une page peut contenir 37 n-uplets résultat). Donc  $P_3 = 7$  pages.

- (d) Estimation du coût  $C_4$  de la jointure de la sous-requête  $E_4 = E_1 \bowtie E_2$  et de sa matérialisation (i.e. estimation de la taille  $P_4$  du résultat de  $E_4$ ) :

La jointure est optimale ici parce que  $P_1 = 1$ . Donc  $C_4 = (P_1 + P_2) E/S$ .

Sachant qu'il y a au total 2000 agences dont 10 ayant le nom 'univers', sous hypothèse de distribution uniforme, l'évaluation de  $E_4$  va produire (environ)  $15000 \times 10/2000 = 75$  n-uplets. On va estimer qu'une page peut contenir 30 n-uplets et donc  $P_4 = 3$  pages.

- (e) Estimation du coût  $C_5$  de la jointure de la sous-requête  $E_5 = E_4 \bowtie E_3$  :

La jointure par bloc est optimale (on va charger les 3 pages résultats de  $E_4$  et effectuer la jointure de ces 3 pages avec chacune des pages du résultat de  $E_3$ ). Donc  $C_5 = P_4 + P_3$ .

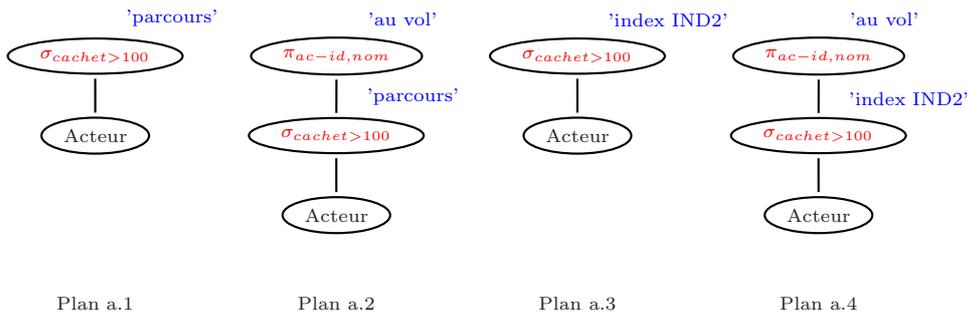
Le coût global se décompose comme suit:  $C_1 + P_1 + C_2 + P_2 + C_4 + P_4 + C_3 + P_3 + C_5$ .

3. A partir de maintenant, vous supposerez de plus que l'enchaînement des opérateurs doit impérativement être "pipelined" ou "au vol". Donnez un exemple de plan (différent de celui de la question 2.) qui ne sera jamais considéré par l'optimiseur.

**Correction :** Il suffisait de prendre l'arbre de requête de la question 1. par exemple et d'ajouter des méthodes pour chaque opérateur. Les sélections "à droite" des jointures imposent la matérialisation des résultats de sous requêtes.

4. Présentez tous les sous-plans d'exécution de la requête impliquant 1 relation et pour chacun donnez une estimation du coût d'exécution en terme de transferts de page (lecture/écriture) en ne prenant pas en compte le coût de l'écriture du résultat final produit par le sous-plan. Donnez une estimation de la taille du résultat final produit par chaque sous-plan.

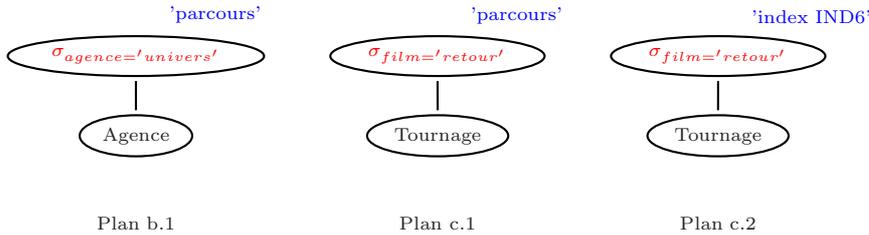
**Correction :** Voici tout d'abord tous les plans d'exécution impliquant la relation Acteur.



Les plans a.1 et a.2 ont même coût soit  $M_1 E/S$ . La taille du résultat de  $\sigma_{cachet>100}[Acteur]$  est de 250 n-uplets (voir discussion précédente). Donc le plan a.1 produit 10 pages et le plan a.2 produit 7 pages.

Les plans a.2 et a.3 ont même coût soit  $h_2 + 10 E/S$  (rappel : Ind2 est groupant et il y a 10 pages de n-uplets réponses). Evidemment, le plan a.3 produit 10 pages et le plan a.4 produit 7 pages.

Comme les plans avec projection ont un coût identique à ceux sans projection, nous les omettons par la suite pour les relations Agence et Tournage.



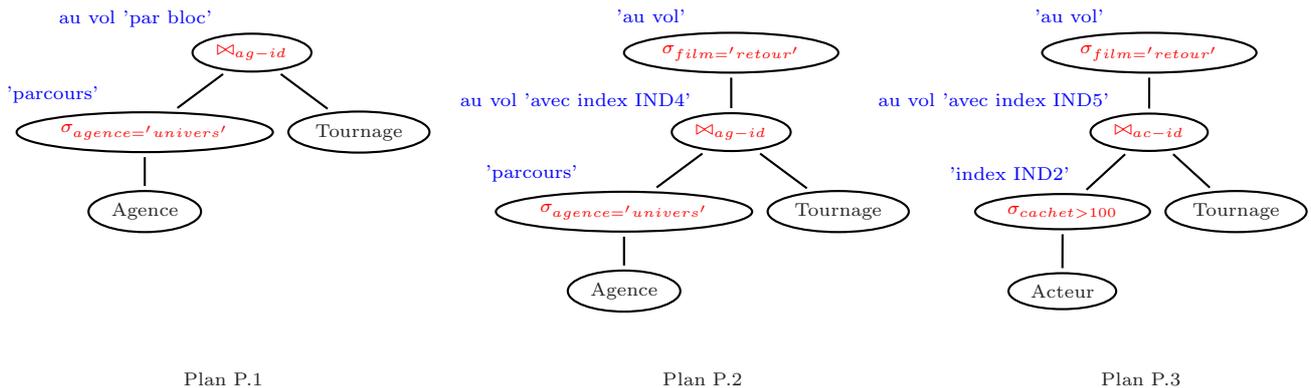
Le coût du plan b.1 est  $M_2 E/S$ . Le nombre de n-uplets produits est 10 (voir discussion précédente). Attention, il est impossible d'utiliser l'index IND3 pour cette sélection.

Le coût du plan c.1 est  $M_3 E/S$ . Le nombre de n-uplets produits est  $100 \times 150$  (voir discussion précédente).

Le coût du plan c.2 est  $(1, 2 + 15000)E/S$  puisque l'index IND6 est non groupant et que 15000 sont à extraire.

5. Présentez au moins 3 sous-plans d'exécution de la requête impliquant 2 relations et pour chacun donnez une estimation du coût d'exécution en terme de transferts de page (lecture/écriture) en ne prenant pas en compte le coût de l'écriture du résultat final produit par le sous-plan. Donnez une estimation de la taille du résultat final produit par chaque sous-plan.

**Correction**



Le plan P.1 est construit à partir du plan b.1 qui produit 10 n-uplets (soit une page) et donc la jointure par bloc effectue simplement le chargement des pages de la relation tournage. Le coût global du plan est donc:  $(M_2 + M_3) E/S$ .

Le plan P.2 est également construit à partir du plan b.1; pour chacun des 10 n-uplets produits, l'index IND4 est utilisé pour rechercher les n-uplets de Tournage "joignable". Par hypothèse de distribution uniforme, on peut estimer que chaque n-uplet du résultat de la sélection est joignable avec  $2 \cdot 10^6 / 2 \cdot 10^3 = 1000$  n-uplets de Tournage. L'index IND4 est groupant donc on sait que ces 1000 n-uplets sont dans  $1000/20 = 50$  pages consécutives.

Donc le coût global du plan est :  $(M_2 + (10 \times (h_4 + 50))) E/S$ .

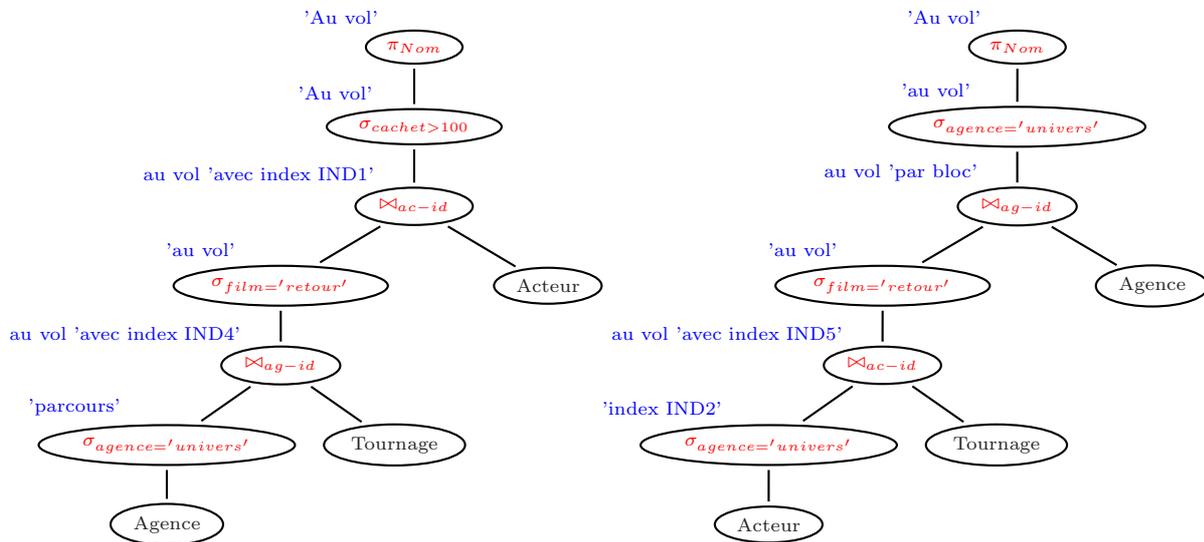
Le plan P.1 produit  $10 \times 10^3 = 10^4$  n-uplets.

Le plan P.2 "effectue" au vol une sélection supplémentaire avec la condition  $film = 'univers'$ . Pour calculer le nombre de n-uplets produits par cette sélection à partir des  $10^4$  n-uplets résultats de la jointure, on peut faire le raisonnement suivant: (i) ces n-uplets ont été obtenus à partir de tous les films; (ii) combien y a-t-il de films ?  $2.10^6/150 = 13334$  (150 étant le nbre d'acteurs par film et  $2.10^6$  le nombre de n-uplets de Tournage); (iii) l'énoncé dit qu'il y a 100 films portant le nom de 'retour'. Donc le plan P.2 produit  $10^4 \times (100/13334) = 75$  n-uplets.

Le plan P.3 est construit à partir du plan a.3 qui produit 250 n-uplets; pour chacun de ces 250 n-uplets, l'index IND5 est utilisé pour rechercher les n-uplets de Tournage "joignables". L'hypothèse de distribution uniforme permet de supposer que chaque n-uplet du résultat de la sélection est joignable avec  $2.10^6/25.10^3 = 80$  n-uplets de Tournage. L'index IND5 n'est pas groupant. Donc le coût global du plan est :  $(M_1 + (250 \times (h_5 + 80))) E/S$ .

La jointure du plan P.3 produit donc  $250 \times 80 = 2.10^4$  n-uplets. Pour obtenir une estimation du nombre de n-uplets obtenus par le plan P.3, on procède à un raisonnement similaire au cas de P.2: (i) les  $2.10^4$  n-uplets ont été obtenus à partir de tous les films; (ii) combien y a-t-il de films ?  $2.10^6/150 = 13334$  (150 étant le nbre d'acteurs par film et  $2.10^6$  le nombre de n-uplets de Tournage); (iii) l'énoncé dit qu'il y a 100 films portant le nom de 'retour'. Donc le plan P.3 produit  $2.10^4 \times (100/13334) = 150$  n-uplets.

6. Présentez 2 plans d'exécution de la requête (différents de celui de la question 2.) et pour chacun donnez une estimation du coût d'exécution en terme de transferts de page (lecture/écriture) en ne prenant pas en compte le coût de l'écriture du résultat final produit par le sous-plan.



Pan Final.1

Pan Final.2

Le plan Final.1 est obtenu à partir du plan P.2. Son coût est donc égal à la somme du coût du plan P.2 (déjà calculé) et du coût de la jointure  $\bowtie_{ac,d}$ . Le coût de la jointure par double itération avec index IND1 est simple à calculer puisque  $ac - id$  est la clé de la relation Acteur et donc pour chaque n-uplet produit par le sous plan P.2, il a exactement un n-uplet de Acteur joignable.

Le coût de la jointure est donc :  $75 \times (h_1 + 1) E/S$ .

Le plan Final.2 est obtenu à partir du plan P.3. Son coût est donc égal à la somme du coût du plan P.3 (déjà calculé) et du coût de la jointure  $\bowtie_{ag,d}$  évalué par double itération par bloc. En supposant qu'une page peut contenir 10 n-uplets résultat du plan P.3, le nombre de pages de résultat générées

par P.3 est estimé à 15. L'algorithme de jointure va donc joindre 5 par 5 les pages de résultats de P.3 avec le fichier stockant Agence. Le coût de la jointure est donc :  $3 \times M_2$ .

## Exercice 4 : Bonus

Attention les réponses à cet exercice ne seront prises en compte que si vous avez obtenu 5 points à l'exercice précédent.

- (a) La relation  $R(A,B,C)$  est stockée dans un fichier tas muni d'un index de type 2 non groupant dont la clé est A,B. Quels sont les types de sélections qui peuvent être optimisées en utilisant cet index.

**Correction :** Toute sélection avec une condition d'égalité portant sur A et sur B pourra être évalué grâce à l'index et sans accès au fichier (donc sans les inconvénients d'un index non groupant).

- (b) Avec les mêmes hypothèses que précédemment, on suppose en plus que l'index est un arbre B+. Donnez une requête pouvant être exécutée par consultation exclusive de l'index ?

**Correction :** Toutes les sélections de (a) mais aussi les sélections ne portant que sur A comme la recherche de la plus grande valeur de A.

- (c) Avec les mêmes hypothèses que précédemment, on suppose en plus que l'index est du hachage. Donnez une requête pouvant être exécutée par consultation exclusive de l'index ?

**Correction :** Toute sélection avec une condition d'égalité portant sur A et sur B pourra être évalué grâce à l'index et sans accès au fichier (donc sans les inconvénients d'un index non groupant). On ne pourra pas utiliser le hachage pour une sélection dont la condition ne porte que sur A.