

Exercice 01

Soit le schéma relationnel suivant :

Propriétaire(numPropriétaire, prénom, nom, adresse, numtél).

Propriété(numPropriété, ville, type, location, numPropriétaire)

Client (numClient, prénom, nom, adresse, typepréf, locMax)

Visite (numPropriété, numClient, datevisite)

- a. Exprimer en SQL la requête suivante : pour le compte de locataires qui ont visité des appartements, trouver tous les logements qui correspondent à leurs exigences et qui appartiennent au propriétaire de numéro CP93.
- b. Donner l'arbre algébrique qui correspond à la requête SQL.
- c. Optimiser la requête en utilisant les règles de transformation.

Exercice 01

Soit le schéma relationnel suivant :

Propriétaire(numPropriétaire, prénom, nom, adresse, numtél).

Propriété(numPropriété, ville, type, location, numPropriétaire)

Client (numClient, prénom, nom, adresse, typepréf, locMax)

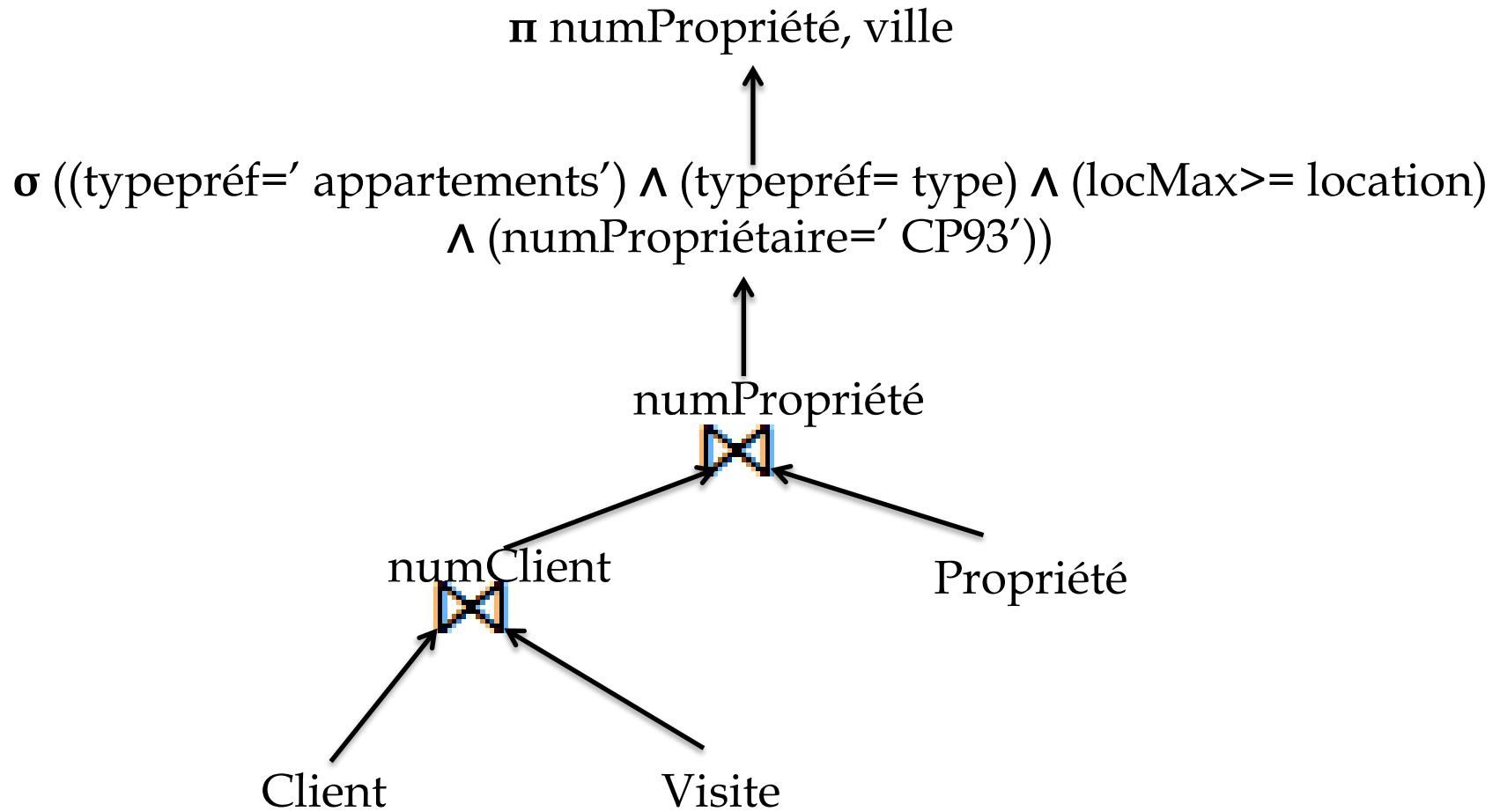
Visite (numPropriété, numClient, datevisite)

- a. Exprimer en SQL la requête suivante : pour le compte de locataires qui ont visité des **appartements**, trouver les logements qui correspondent à leurs exigences et qui appartiennent au propriétaire de numéro **CP93**.

```
SELECT P.numPropriété, P.ville
FROM Client C, Visite V, Propriété P
WHERE typepréf=' appartements'
AND typepréf= type
AND locMax>= location
AND numPropriétaire=' CP93'
AND C. numClient=V. numClient
AND V. numPropriété=P. numPropriété
```

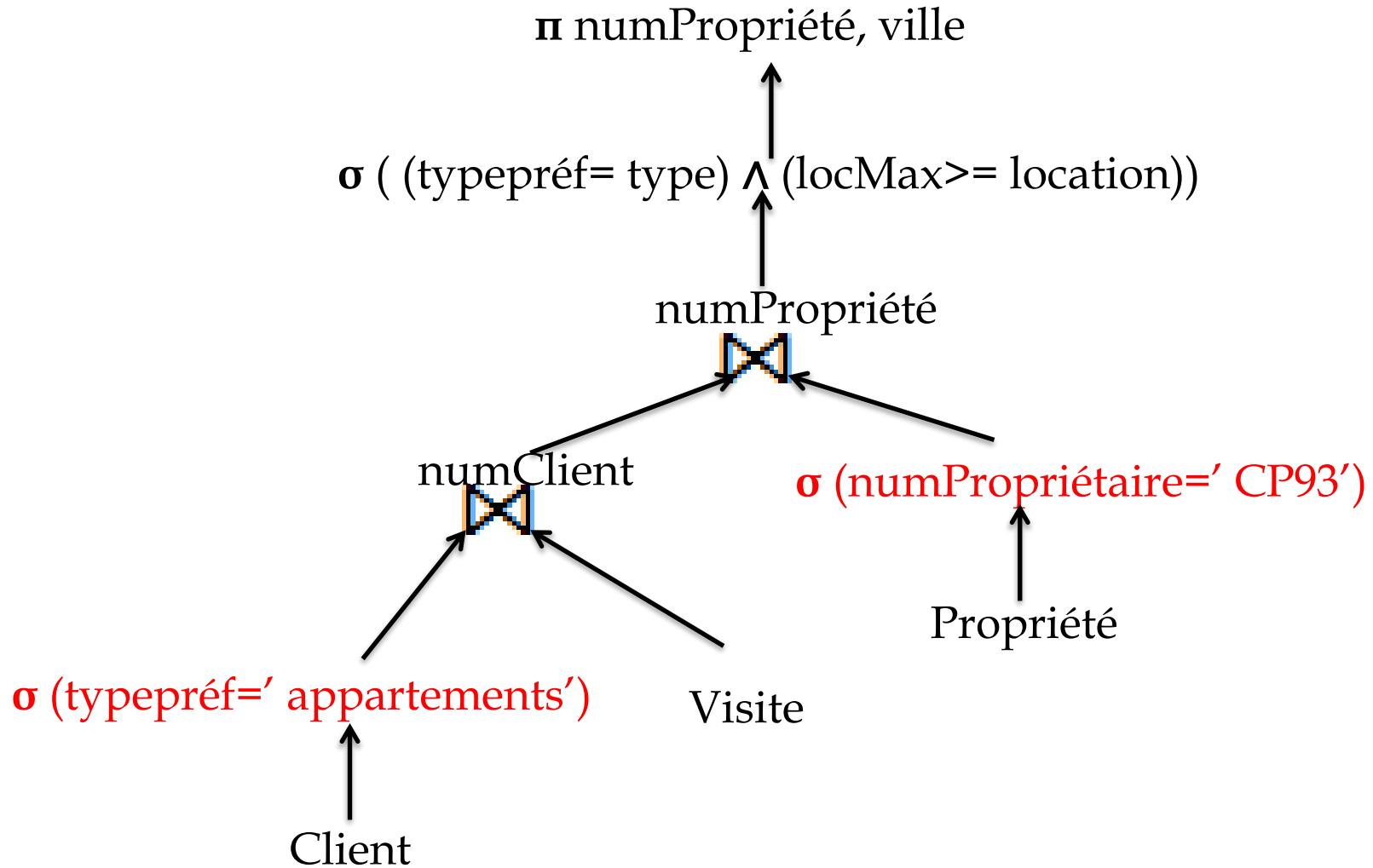
Exercice 01

- b. Donner l'arbre algébrique qui correspond à la requête SQL.



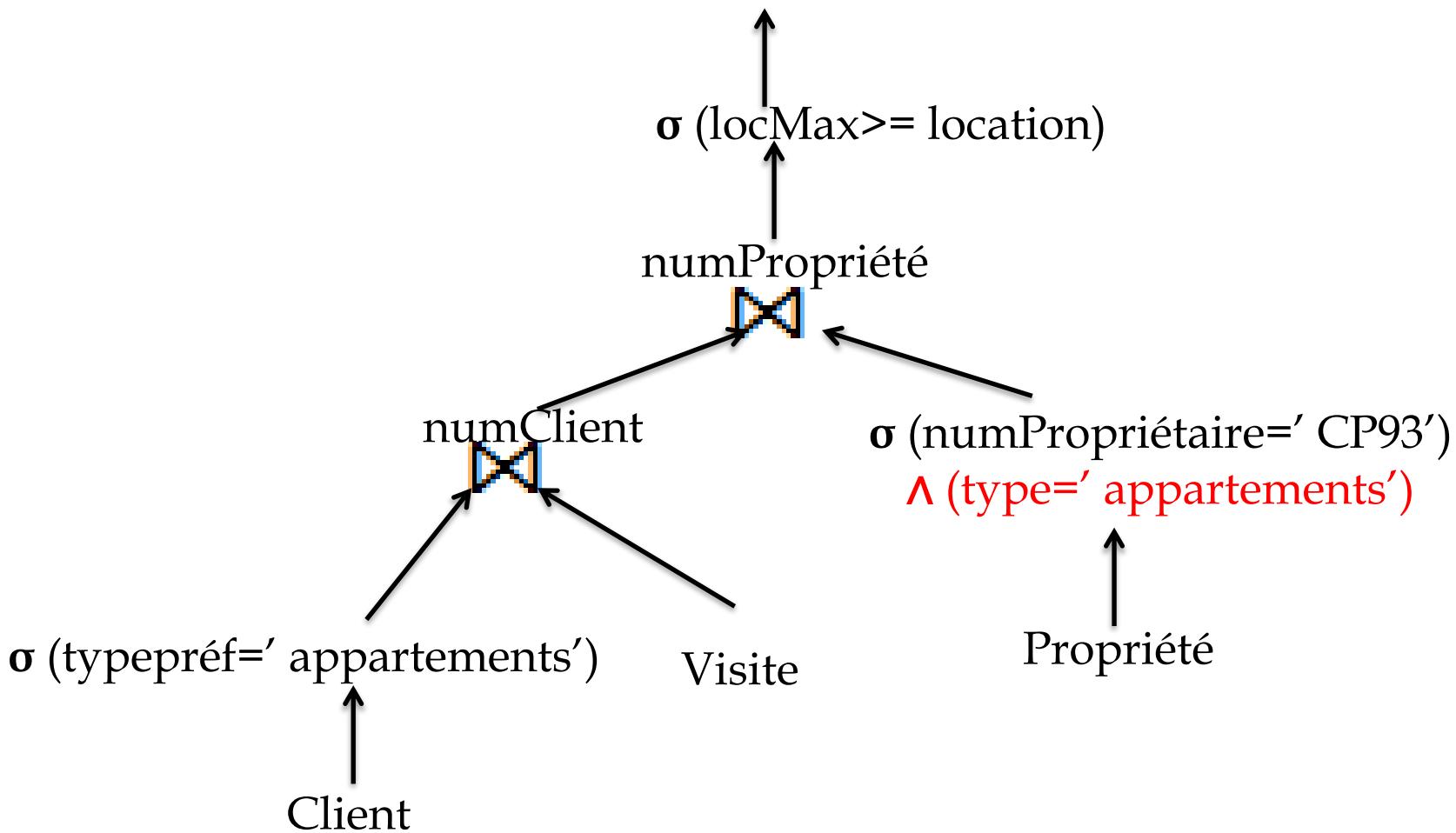
Exercice 01

- c. Optimiser la requête en utilisant les règles de transformation.
➤ Faire descendre les restrictions



Exercice 01

- c. Optimiser la requête en utilisant les règles de transformation.
- Remplacer σ ($\text{typepré} = \text{type}$) par σ ($\text{type} = 'appartements'$)
 $\pi \text{ numPropriété, ville}$



Exercice 01

- c. Optimiser la requête en utilisant les règles de transformation.
➤ Faire descendre les projections

$\pi \text{ numPropriété, ville}$

$\sigma (\text{locMax} \geq \text{location})$

numPropriété

numClient

$\pi \text{ numPropriété, location, ville}$

$\sigma (\text{numPropriétaire} = 'CP93') \wedge (\text{typepréf} = 'appartements')$

Propriété

$\pi \text{ numClient, locMax } \pi \text{ numClient, numPropriété}$

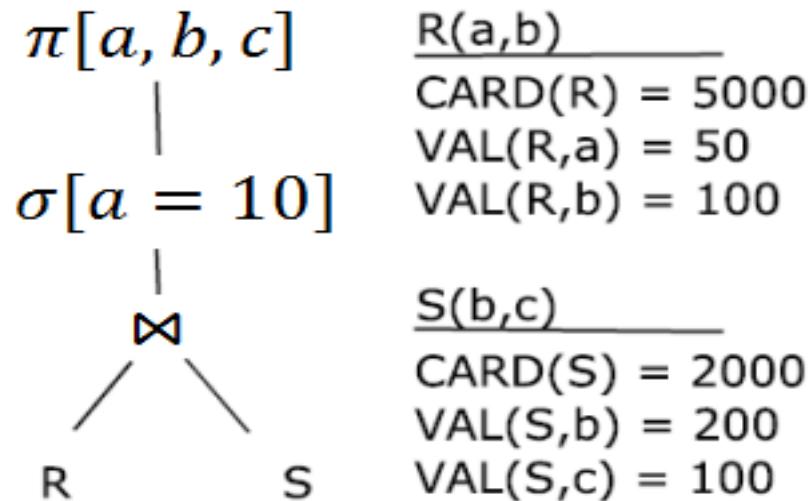
$\sigma (\text{typepréf} = 'appartements')$

Visite

Client

Exercice 02

Soit un arbre algébrique et des informations statistiques sur les tables qu'il manipule.

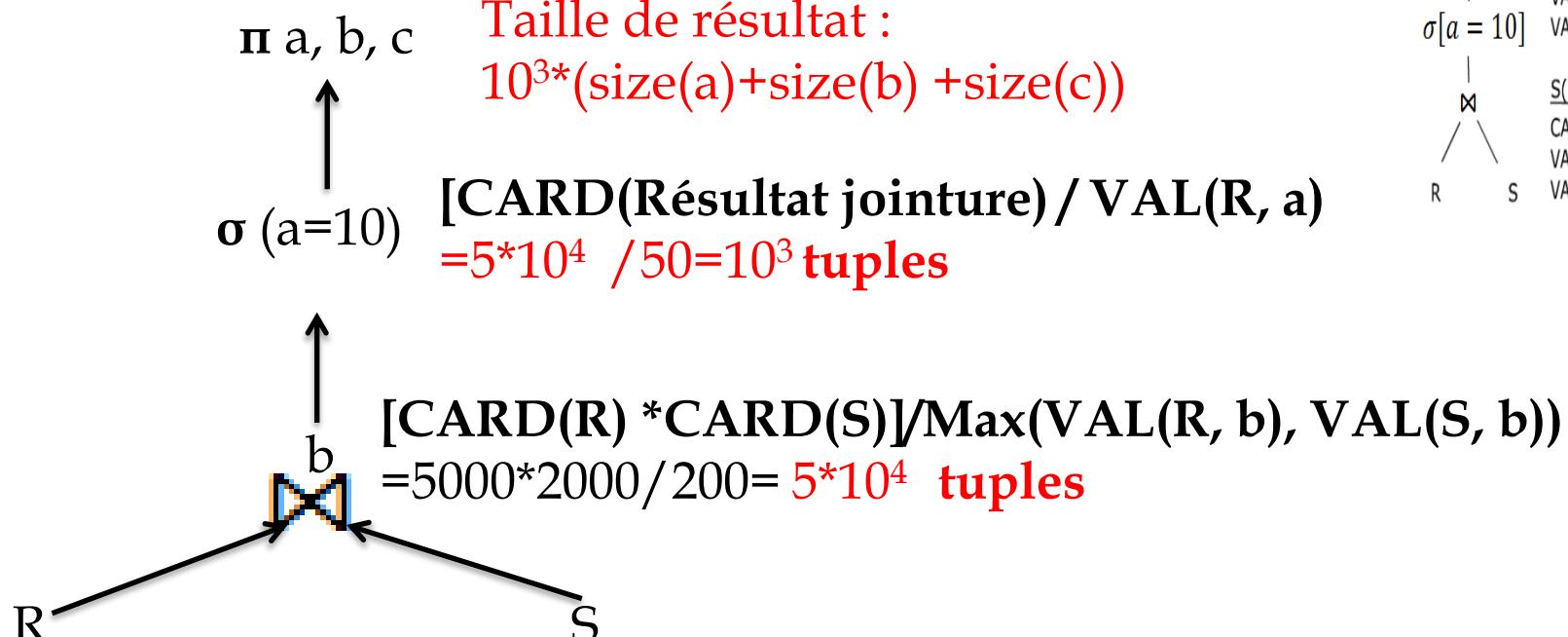


Indiquez les exécutions possibles et évaluez leur coût d'exécution.

Exercice 02

Indiquez les exécutions possibles et évaluez leur coût d'exécution.

➤ Plan 1:



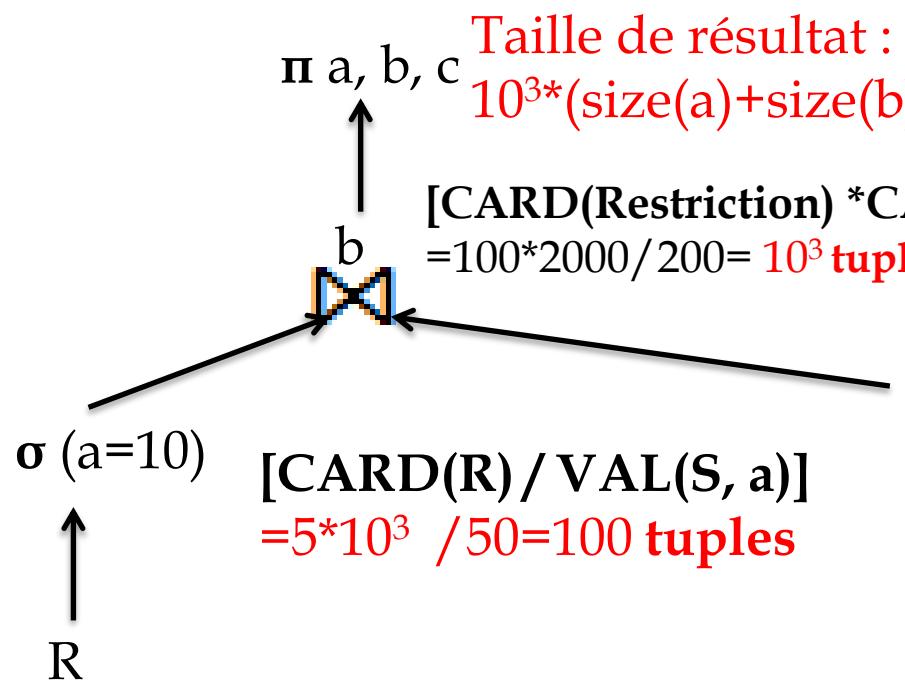
| | |
|-----------------------|---|
| $\pi[a, b, c]$ | $\frac{R(a,b)}{\text{CARD}(R) = 5000}$ |
| σ _[a = 10] | $\frac{\text{VAL}(R,a) = 50}{\text{VAL}(R,b) = 100}$ |
| | \bowtie |
| | $\frac{S(b,c)}{\text{CARD}(S) = 2000}$ |
| R S | $\frac{\text{VAL}(S,b) = 200}{\text{VAL}(S,c) = 100}$ |

Coût logique de plan 1: $5 * 10^4 + 10^3 = 51 * 10^3$ tuples

Exercice 02

Indiquez les exécutions possibles et évaluez leur coût d'exécution.

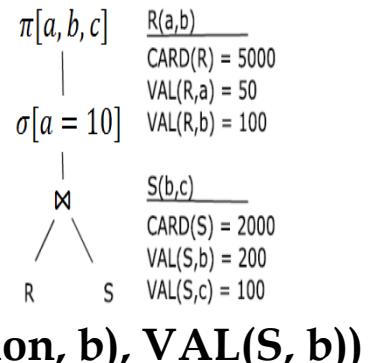
➤ Plan 2:



Taille de résultat :
 $10^3 * (\text{size}(a) + \text{size}(b) + \text{size}(c))$

$$[\text{CARD}(\text{Restriction}) * \text{CARD}(S)] / \text{Max}(\text{VAL}(\text{Restriction}, b), \text{VAL}(S, b))$$
$$= 100 * 2000 / 200 = 10^3 \text{ tuples}$$

$$[\text{CARD}(R) / \text{VAL}(S, a)]$$
$$= 5 * 10^3 / 50 = 100 \text{ tuples}$$



Coût logique de plan 2: $100 + 10^3 = 11 * 10^2$ tuples

Exercice 02

Indiquez les exécutions possibles et évaluez leur coût d'exécution.

❑ L'optimisation logique consiste à réduire le nombre de tuples générés à chaque étape.

➤ Plan 1: **$51 \cdot 10^3$ tuples**

➤ Plan 2: **$11 \cdot 10^2$ tuples .**

→ Le plan optimal est le plan 2

Exercice 03

❑ Étant donné le modèle relationnel et la requête SQL suivants:

Étudiant (ide, nom, âge, adresse)

Livre (idl, titre, auteur)

Emprunter (ide*, idl*, date)

❑ Requête : SELECT e.nom

```
FROM etudiant e, livre l, emprunter em  
WHERE e.ide = em.ide  
AND l.idl = em.idl  
AND l.auteur = 'DATE'  
AND e âge = 20
```

❑ Et en supposant:

- Il y a 10000 enregistrements de la table étudiants stockés sur 1000 pages.
- Il y a 50000 enregistrements de la table livres stockés sur 5000 pages.
- Il y a 300000 enregistrements de la table emprunter stockés sur 15000 pages.
- Il y a 500 auteurs différents.
- Les âges des étudiants vont de 10 à 24 ans.

Exercice 03

□ Étant donné le modèle relationnel et la requête SQL suivants:

Étudiant (ide, nom, âge, adresse)

Livre (idl, titre, auteur)

Emprunter (ide*, idl*, date)

□ **Requête :** SELECT e.nom

FROM etudiant e, livre l, emprunter em

WHERE e.ide = em.ide

AND l.idl = em.idl

AND l.auteur = 'DATE'

AND e.âge=20

Exercice 03

□ Et en supposant:

- Il y a 10000 enregistrements de la table étudiants stockés sur 1000 pages.

→ CARD(étudiant)= 10000 tuples. Nombre de tuples par page=CARD(étudiant)/nombre de pages(étudiant)= $10000/1000 = 10$ tuples .

- Il y a 50000 enregistrements de la table livres stockés sur 5000 pages.

→ CARD(livre)= 50000 tuples. Nombre de tuples par page=CARD(livre)/nombre de pages(livre)= $50000 / 5000 = 10$ tuples.

- Il y a 300000 enregistrements de la table emprunter stockés sur 15000 pages.

→ CARD(emprunter)= 300000 tuples. Nombre de tuples par page=CARD(emprunter)/nombre de pages(emprunter)= $300000 / 15000 = 20$ tuples

- Il y a 500 auteurs différents.

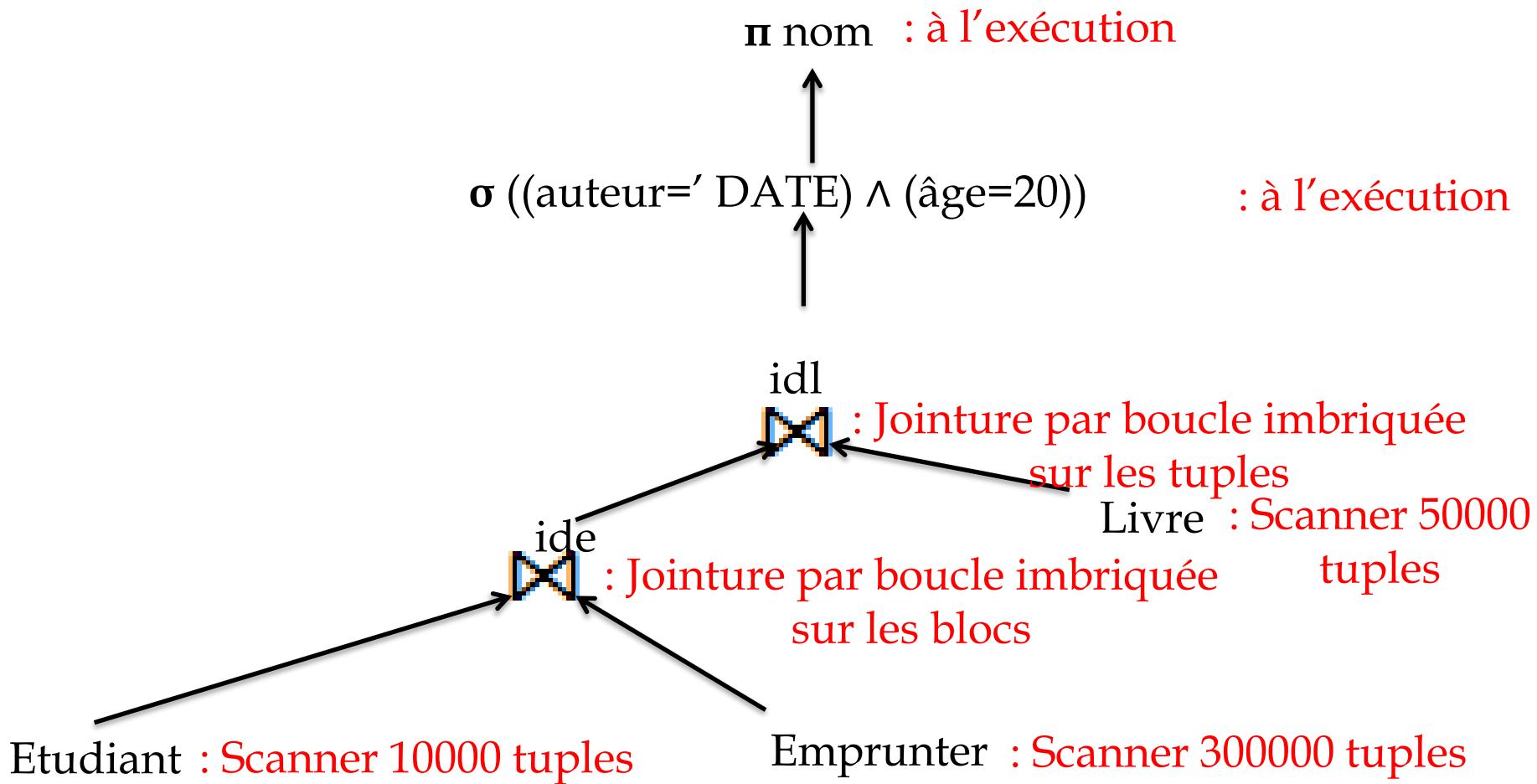
→ Val(auteur, livre)=500.

- Les âges des étudiants vont de 10 à 24 ans.

→ val(âge, étudiant)= $24-10+1=15$, max(âge)=24 et min(âge)=10.

Exercice 03

- a. Donner un plan d'exécution pour cette requête, en supposant qu'il n'y a pas d'index et que les données ne sont pas triées sur n'importe quel attribut.



Exercice 03

b. Calculez le coût de ce plan et la cardinalité du résultat.

Abréviation : **ET** : Etudiant, **EM** : Emprunter, **L** : livre, **B** : Bloc

| Opération | Coût | Cardinalité | Commentaire |
|---|---|---|---|
| ET  EM | $\begin{aligned} & B(ET) + B(ET) * B(EM) \\ & = 1000 + 1000 * 15000 \\ & = 15001000 \text{ opérations} \end{aligned}$ | -300000 tuples(jointure sur clé étrangère (ide) : $(300000 * 10000) / 10000$). | -On utilise la boucle imbriquée basée sur les pages, chaque itération on charge une page étudiant et pour chaque page on charge les pages de la table emprunter, on suppose que les tuples résultants restent en mémoire centrale (il y a assez d'espace). Le tuple résultant sera utilisé pour la deuxième jointure. |

Exercice 03

b. Calculez le coût de ce plan et la cardinalité du résultat.

Abréviation : **ET** : Etudiant, **EM** : Emprunter, **L** : livre, **B** : Bloc

| Opération | Coût | Cardinalité | Commentaire |
|---|---|---|---|
| ET  EM | $B(ET) + B(ET) * B(EM)$ $= 1000 + 1000 * 15000$ $= 15001000 \text{ opérations}$ | -300000 tuples(jointure sur clé étrangère (ide) : $(300000 * 10000) / 10000$). | -On utilise la boucle imbriquée basée sur les pages, chaque itération on charge une page étudiant et pour chaque page on charge les pages de la table emprunter, on suppose que les tuples résultants restent en mémoire centrale (il y a assez d'espace). Le tuple résultant sera utilisé pour la deuxième jointure. |
| (ET  EM)  L $= C \bowtie L$ | $C * B(L)$ $= 300000 * 5000$ $= 1500000000 \text{ opérations}$ | -300000 tuples(jointure sur clé étrangère (idl) : $(300000 * 5000) / 5000$). | -Les 300000 tuples résultants qui existent au niveau de la mémoire centrale vont être utilisés directement pour la deuxième jointure, on les nomme C. Pour chaque tuple C on charge tous les blocs de la table livre. |

Exercice 03

b. Calculez le coût de ce plan et la cardinalité du résultat.

Abréviation : **ET** : Etudiant, **EM** : Emprunter, **L** : livre, **B** : Bloc

| Opération | Coût | Cardinalité | Commentaire |
|---|---|---|---|
| ET  EM | $B(ET) + B(ET) * B(EM)$ =1000+1000*15000 =15001000 opérations | -300000 tuples(jointure sur clé étrangère (idet) : (300000*10000)/ 10000). | -On utilise la boucle imbriquée basée sur les pages, chaque itération on charge une page étudiant et pour chaque page on charge les pages de la table emprunter, on suppose que les tuples résultants restent en mémoire centrale (il y a assez d'espace). Le tuple résultant sera utilisé pour la deuxième jointure. |
| (ET  EM)  L $= C \bowtie L$ | $C*B(L)$ =300000*5000 =1500000000 opérations | -300000 tuples(jointure sur clé étrangère (idl) : (300000*5000)/ 5000). | -Les 300000 tuples résultants qui existent au niveau de la mémoire centrale vont être utilisés directement pour la deuxième jointure, on les nomme C. Pour chaque tuple C on charge tous les blocs de la table livre. |
| σ et π | A l'exécution | $-300000 * \sigma_{auteur} * \sigma_{age}$ $-300000 * (1/500) * (1/15)$ $-40 * \text{size (nom)}$ | -On suppose qu'on a une distribution uniforme sur l'âge et auteur. |

Exercice 03

b. Calculez le coût de ce plan et la cardinalité du résultat.

Abréviation : **ET** : Etudiant, **EM** : Emprunter, **L** : livre, **B** : Bloc

| Opération | Coût | Cardinalité | Commentaire |
|---|---|---|---|
| ET \bowtie EM | $B(ET) + B(ET) * B(EM)$ = $1000 + 1000 * 15000$ = 15001000 opérations | -300000 tuples(jointure sur clé étrangère (idet) : $(300000 * 10000) / 10000$). | -On utilise la boucle imbriquée basée sur les pages, chaque itération on charge une page étudiant et pour chaque page on charge les pages de la table emprunter, on suppose que les tuples résultants restent en mémoire centrale (il y a assez d'espace). Le tuple résultant sera utilisé pour la deuxième jointure. |
| (ET \bowtie EM) \bowtie L = C \bowtie L | $C * B(L)$ = $300000 * 5000$ = 1500000000 opérations | -300000 tuples(jointure sur clé étrangère (idl) : $(300000 * 5000) / 5000$). | -Les 300000 tuples résultants qui existent au niveau de la mémoire centrale vont être utilisés directement pour la deuxième jointure, on les nomme C. Pour chaque tuple C on charge tous les blocs de la table livre. |
| σ et π | A l'exécution | $-300000 * \sigma_{auteur} * \sigma_{age}$ $-300000 * (1/500) * (1/15)$ $-40 * \text{size (nom)}$ | -On suppose qu'on a une distribution uniforme sur l'âge et auteur. |
| Total | 1515001000 opérations | 40 * size (nom) octets | |

Exercice 03

c. Suggérer deux index et un autre plan pour cette requête.

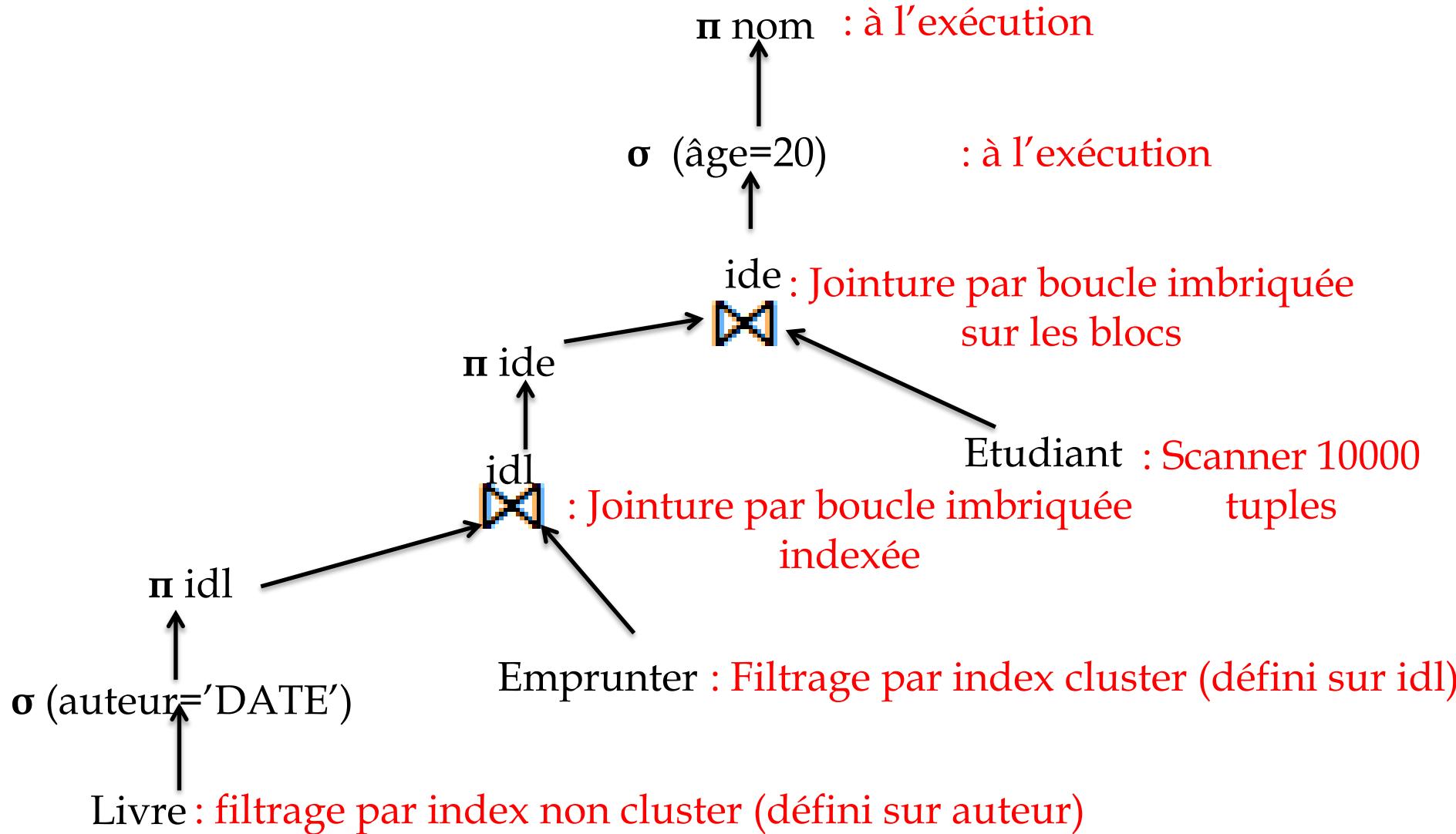
➤ On peut avoir plusieurs solutions, pour illustrer, on suppose d'avoir un index de *type b-arbre n'est pas organisé en cluster* pour la table **Livre** défini sur l'attribut **auteur** et un index de *type b-arbre organisé en cluster* pour la table **Emprunter** sur l'attribut **idl**.

Il existe de nombreux types d'indexes, dont les plus connus et utilisés sont l'index en cluster (clustered index) et l'index non-cluster (non-clustered index). Et la différence entre les 2 est la suivante :

- ❑ **L'index en cluster (ou ordonné) (primaire)** est un index au sein duquel les enregistrements sont **physiquement triés**, et ses pages de données au niveau feuille et les numéros d'enregistrements au niveau intermédiaire. C'est ainsi qu'une table ne peut avoir **qu'un index en cluster** à la fois dans la mesure où les lignes ne peuvent être triées qu'en un seul ordre physique.
- ❑ **L'index non-cluster (ou non-ordonné)(secondaire)** est un index au sein duquel **le tri logique** des enregistrements (i.e., tri des pages d'indexes) ne correspond pas au **tri physique des enregistrements** (sur le disque). De plus, au niveau structurel, seuls **les adresses d'accès aux enregistrements** sont placés au niveau feuille, ce qui permet donc, à une table, de pouvoir disposer de **plusieurs indexes non-clusters**.

Exercice 03

- c. Suggérer deux index et un autre plan pour cette requête.



Exercice 03

c. Calculez le coût de nouveau plan.

| Opération | Coût | Cardinalité | Commentaire |
|--|---|--------------|---|
| σ (auteur=' DATE') : Filtrage par un index sur la table Livre | $Card(L) * (1 / Val(auteur, L))$ $= 50000 * 1 / 500$ $= 100$ opérations | - 100 tuples | - on suppose que toutes les pages index sont en mémoire centrale. L'index qu'on a défini et organisé en non cluster, la page index contient la clé (nom auteur) et l'idl (identifiant de livre), pour la jointure on a besoin que de idl d'où on n'a pas besoin de faire une entrée/sortie. |

Exercice 03

c. Calculez le coût de nouveau plan.

| Opération | Coût | Cardinalité | Commentaire |
|---|---|---|---|
| σ (auteur=' DATE') : Filtrage par un index sur la table Livre | $Card(L) * (1 / Val(auteur, L))$ $= 50000 * 1 / 500$ $= 100$ opérations | - 100 tuples | - on suppose que toutes les pages index sont en mémoire centrale. L'index qu'on a défini et organisé en non cluster, la page index contient la clé (nom auteur) et l'idl (identifiant de livre), pour la jointure on a besoin que de idl d'où on n'a pas besoin de faire une entrée/sortie. |
| $\pi_{Id_L} [L \bowtie EM]$ | $100 * ((card(EM) / Val(idl, EM)) / N(EM))$ $100 * ((300000 / 50000) / 20)$ $= 100$ opérations NB: $((300000 / 50000) / 20)$: chargés dans une page | $-(100 * 300000) / Max(100, Val(idl, Emprunter))$ $= 600$ tuples = C | - Avec l'index organisé en cluster défini sur la table Emprunter (idl) on aura $(300000 / 50000) / 20 = 6$ tuples seront chargés en une page, pour chaque tuple livre résultant (les 100 tuples résultants de l'étape précédente), on va charger cette page. Le résultat des 600 tuples vont rester en mémoire centrale. - $N(EM)$: nombre de tuple emprunter dans une page |

Exercice 03

c. Calculez le coût de nouveau plan.

| Opération | Coût | Cardinalité | Commentaire |
|---|---|---|--|
| σ (auteur=' DATE') : Filtrage par un index sur la table Livre | $Card(L) * (1 / Val(auteur, L))$ $= 50000 * 1 / 500$ $= 100$ opérations | - 100 tuples | - on suppose que toutes les pages index sont en mémoire centrale. L'index qu'on a défini et organisé en non cluster, la page index contient la clé (nom auteur) et l'idl (identifiant de livre), pour la jointure on a besoin que de idl d'où on n'a pas besoin de faire une entrée/sortie. |
| $\pi_{Id_e}(L \bowtie EM)$ | $100 * ((card(EM) / Val(idl, EM)) / N(EM))$ $100 * ((300000 / 50000) / 20)$ $= 100$ opérations NB: $((300000 / 50000) / 20)$: chargés dans une page | $-(100 * 300000) / Max(100, Val(idl, Emprunter))$ $= 600$ tuples = C | - Avec l'index organisé en cluster défini sur la table Emprunter (idl) on aura $(300000 / 50000) / 20 = 6$ tuples seront chargés en une page, pour chaque tuple livre résultant (les 100 tuples résultants de l'étape précédente), on va charger cette page. Le résultat des 600 tuples vont rester en mémoire centrale. |
| C \bowtie ET | B(ET) = 1000 opérations | $-(10000 * 600) / Max(600, Val(ide, Etudiant))$ $= 600$ tuples | - Vu que les 600 tuples sont en mémoire centrale donc on a besoin de scanner que la table étudiant. |

Exercice 03

c. Calculez le coût de nouveau plan.

| Opération | Coût | Cardinalité | Commentaire |
|---|---|---|--|
| σ (auteur=' DATE') : Filtrage par un index sur la table Livre | $Card(L) * (1 / Val(auteur, L))$ $= 50000 * 1 / 500$ $= 100$ opérations | - 100 tuples | - on suppose que toutes les pages index sont en mémoire centrale. L'index qu'on a défini et organisé en non cluster, la page index contient la clé (nom auteur) et l'idl (identifiant de livre), pour la jointure on a besoin que de idl d'où on n'a pas besoin de faire une entrée/sortie. |
| π_{Id} $L \bowtie EM$ | $100 * ((card(EM) / Val(idl, EM)) / N(EM))$ $100 * ((300000 / 50000) / 20)$ $= 100$ opérations NB: $((300000 / 50000) / 20)$: chargés dans une page | $-(100 * 300000) / Max(100, Val(idl, Emprunter))$ $= 600$ tuples = C | - Avec l'index organisé en cluster défini sur la table Emprunter (idl) on aura $(300000 / 50000) / 20 = 6$ tuples seront chargés en une page, pour chaque tuple livre résultant (les 100 tuples résultants de l'étape précédente), on va charger cette page. Le résultat des 600 tuples vont rester en mémoire centrale. |
| C \bowtie ET | B(ET) = 1000 opérations | $-(10000 * 600) / Max(600, Val(ide, Etudiant))$ $= 600$ tuples | - Vu que les 600 tuples sont en mémoire centrale donc on a besoin de scanner que la table étudiant. |
| σ et π | A l'exécution | $-600 * \sigma_{age}$ $600 * (1 / 15)$ $40 * size(nom)$ | - On suppose qu'on a une distribution uniforme sur l'âge. |

Exercice 03

c. Calculez le coût de nouveau plan.

| Opération | Coût | Cardinalité | Commentaire |
|---|---|---|--|
| σ (auteur=' DATE') : Filtrage par un index sur la table Livre | $Card(L) * (1 / Val(auteur, L))$ $= 50000 * 1 / 500$ $= 100$ opérations | - 100 tuples | - on suppose que toutes les pages index sont en mémoire centrale. L'index qu'on a défini et organisé en non cluster, la page index contient la clé (nom auteur) et l'idl (identifiant de livre), pour la jointure on a besoin que de idl d'où on n'a pas besoin de faire une entrée/sortie. |
| $\pi_{Id\acute{e}}(L \bowtie EM)$ | $100 * ((card(EM) / Val(idl, EM)) / N(EM))$ $100 * ((300000 / 50000) / 20)$ $= 100$ opérations NB: $((300000 / 50000) / 20)$: chargés dans une page | $-(100 * 300000) / Max(100, Val(idl, Emprunter))$ $= 600$ tuples = C | - Avec l'index organisé en cluster défini sur la table Emprunter (idl) on aura $(300000 / 50000) / 20 = 6$ tuples seront chargés en une page, pour chaque tuple livre résultant (les 100 tuples résultants de l'étape précédente), on va charger cette page. Le résultat des 600 tuples vont rester en mémoire centrale. |
| C \bowtie ET | B(ET) = 1000 opérations | $-(10000 * 600) / Max(600, Val(ide, Etudiant))$ $= 600$ tuples | - Vu que les 600 tuples sont en mémoire centrale donc on a besoin de scanner que la table étudiant. |
| σ et π | A l'exécution | $-600 * \sigma_{age}$ $600 * (1 / 15)$ $40 * size(nom)$ | - On suppose qu'on a une distribution uniforme sur l'âge. |
| Total | 1200 opérations | $40 * size(nom)$ octets | |