

Interrogation SGBD (COURS, 5 Pts)

1. Quelles sont les deux modes d'optimisation utilisés dans les SGBD en donnant leurs principes ?
2. Donner les entrées et les sorties de chaque mode.
3. Donner deux implémentations pour l'exécution d'une jointure. Donner le coût pour chaque implémentation.
4. **[APCI]** Quelles sont les stratégies d'exécution d'une jointure sur une BD distribuée.

Nom :

Prénom :

Matricule :

Groupe :

1. Quelles sont les deux modes d'optimisation utilisés dans les SGBD en donnant leurs principes ?

Mode RBO (Rule Based Optimization) : un ensemble de règles figées est à appliquer sur l'arbre algébrique pour générer un autre arbre. La transformation se base sur les propriétés des opérateurs algébriques.

Mode CBS (Cost Based Optimisization) : Un modèle de coût est utilisé pour évaluer le coût d'exécution de tous les plans d'exécution afin d'en choisir le moins couteux. Ce mode est basé sur un ensemble de statistiques collectées périodiquement et stockées dans la méta-base.

2. Donner les entrées et les sorties de chaque mode.

Modèle	Entrées	Sorties
RBO	Arbre algébrique (A) Ensemble de règles (R)	Un arbre algébrique optimisé résultat de l'application de R sur A.
CBO	Modèle de coût Statistiques Arbre algébrique	Un plan d'exécution caractérisé par le coût minimum.

3. Donner deux implémentations pour l'exécution d'une jointure. Donner le coût pour chaque implémentation.

- a. Jointure par boucles imbriquées : dans cette implémentation, le SGBD charge un bloc de la première table, et pour chaque bloc, il charge toute la deuxième table pour vérifier la correspondance.

Cost(R JOIN S)= |R|+|R|*|S| où |T| représente le nombre de page nécessaires pour stocker le table T.

- b. Jointure par hachage : le but de cette implémentation est de décomposer une large jointure en un ensemble de petites jointures qui peuvent se faire en mémoire. Une fonction de hachage est appliquée sur les deux tables ce qui les fragmente en plusieurs blocs. Le SGBD charge chaque couple de bloc et fait la sous-jointure en mémoire.

Cost(R JOIN S)= 3*(|R|+|S|)

5. **[APCI]** Quelles sont les stratégies d'exécution d'une jointure sur une BD distribuée.

- a. Pour exécuter une jointure (**R JOIN S**) dans une base de données distribuée, il faut que les deux tables soient sur le même nœud. Trois stratégies sont souvent possibles
 - i. Transférer les deux tables R et S sur le site où la requête est lancée, l'exécuter et rendre le résultat à l'utilisateur

- ii. Transférer la table R sur le site où se trouve la table S, l'exécuter sur ce site ensuite envoyer le résultat au site où la requête a été lancée.
- iii. Faire le contraire en transférant la table S sur le site où se trouve R.

Interrogation SGBD (TD 10 Pts)

Nom :

Prénom :

Matricule :

Groupe :

Soient les ordonnancements suivants :

O1 : R1(A)R3(B)W1(B)W4(A)W1(E)W5(A)R2(D)R3(D)W2(D)W5(D)R4(E)R5(E)

O2 : R1(A)R2(A)W2(A)R3(A)R4(D)W1(A)W3(B)W3(A)W1(B)W5(A)W1(D)W5(B)R5(D)

O3 : R1(A)R1(E)R3(D)R4(D)W2(D)R4(A)R3(A)W4(A)W4(B)W3(B)W1(B)W2(E)

Supposons qu'ils sont corrects :

1. Construire le graphe de précédence pour O1 et O2. Sont-ils sérialisables ?
2. Construire le graphe d'attente pour O2 et O3. Existe-t-il un deadlock ? Trouver une solution.
3. Dérouler l'algorithme d'estampillage pour O2. Existe-t-il un rejet ?

O1

Graphes de précédence

Sur A : R1(A) W4(A) W5(A)

T1 précède T4, T5

T4 " T5

Sur B : R3(B) W1(B)

T3 précède T1

Sur D :

R2(D) R3(D) W2(D) W5(D)

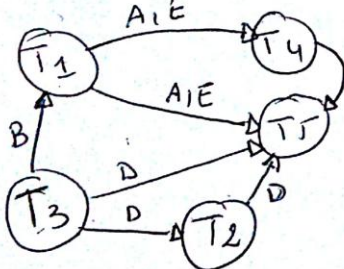
T2 précède T5

T3 " T2, T5

Sur E :

W1(E) R4(E) R5(E)

T1 précède T4, T5



- Absence de cycle
- Sérialisable

ordre : T3 T1 T4 T2 T5

O2

Sur A : R1(A) R2(A) W2(A) R3(A) W1(A) W3(A) W4(A)

T1 précède T2, T3, T5

T2 " T1, T3, T5

T3 " T1, T5

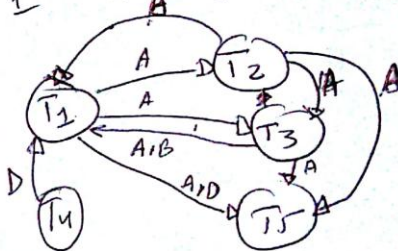
Sur B : W3(B) W1(B)

T3 précède T1

Sur D : R4(D) W1(D) R5(D)

T4 précède T1

T1 " T5

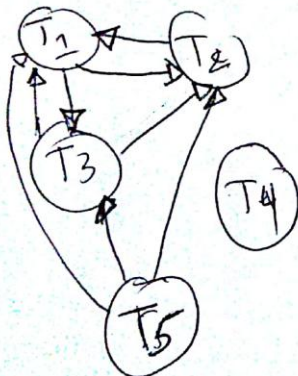


- Deux cycles
- Non sérialisable

Graphes d'attente

02

Action	Demande	Rép	attente
R ₂ (A)	S ₂ (A)	OK	
R ₃ (A)	S ₂ (A)	OK	
W ₂ (A)	X ₂ (A)	Now	T ₂ att T ₁
R ₃ (A)	Fin T₂		
R ₄ (D)	S ₃ (A)	OK	
R ₄ (D)	S ₄ (D)	OK	U(D)
W ₁ (A)	Fin T₄		
W ₁ (A)	S ₁ (A)	Now	T ₁ att T ₂ , T ₃
W ₃ (B)	X ₃ (B)	OK	
W ₃ (A)	X ₃ (A)	Now	T ₃ att T ₁ , T ₂
W ₁ (B)	Fin T₃		
W ₁ (A)	X ₁ (A)	Now	attente
W ₁ (D)	Fin T₁		
W ₅ (B)			T ₅ att T ₂ , T ₃ , T ₄
R ₅ (D)	Fin T₅		



3 cycles T₁ T₂ T₃ : C₁
 T₁ T₂ : C₂ → Deadlock
 T₁ T₃ : C₃

$$C_1 \wedge C_2 \wedge C_3 = T_1$$

→ annuler T₁, Relancer T₁
 ordre : T₄ T₂ T₃ T₅ T₁

03

Action	Demande	Rép	Attale
R ₁ (A)	S ₁ (A)	OK	
R ₁ (E)	S ₁ (E)	OK	
R ₃ (D)	S ₃ (D)	OK	
R ₄ (D)	S ₄ (D)	OK	
W ₂ (D)	W ₂ (D)	Now	T ₂ att T ₃ , T ₄
R ₄ (A)	S ₄ (A)	OK	
R ₃ (A)	S ₃ (A)	OK	
W ₄ (A)	X ₄ (A)	Now	T ₄ att T ₂ , T ₃
W ₄ (B)			attente
W ₃ (B)	Fin T₄		
W ₁ (B)	Fin T₃		
W ₁ (B)	Fin T₁		
W ₂ (E)	Fin T₂		

Passé 2: Réveiller T₄
 (T₂ non réveillée car elle attend T₃ et T₄)

W₄(A) — OK

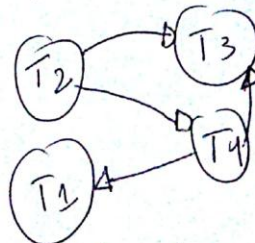
W₄(B) — OK Fin T₄ U(A, B, D)

W₂(D) —

Passé 3: Réveiller T₂

W₂(D) — OK

W₂(E) — OK



pas de cycle → sérialisable.

Estampillage 02

Action	EL(A)	EE(A)	EL(B)	EE(B)	EL(D)	EE(D)
R ₁ (A)	0	0	0	0	0	0
R ₂ (A)	1	0	0	0	0	0
R ₂ (A)	2	0	0	0	0	0
W ₂ (A)	2	2	0	0	0	0
R ₃ (A)	3	2	0	0	0	0
R ₄ (D)	3	2	0	0	4	0
W ₁ (A)	Rejeter w ₁ (A), annuler T ₁ , Relancer T ₁					
W ₃ (B)	3	2	0	3	4	0
W ₃ (A)	3	3	0	3	4	0
W ₁ (B)						
W ₅ (A)	3	5	0	3	4	0
W ₁ (D)						
W ₅ (B)	3	5	0	5	4	0
R ₅ (D)	3	5	0	5	5	0

Oni T₁ a été Rejetée.

R ₆ (A)	6	5	0	5	5	0
W ₆ (A)	6	6	0	5	5	0
W ₆ (B)	6	6	0	6	5	0
W ₆ (D)	6	6	0	6	<u>5</u>	6