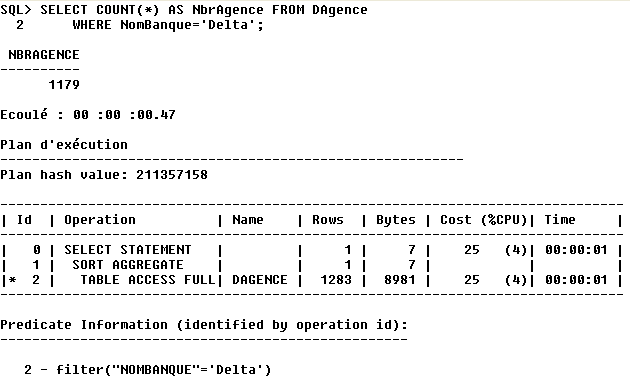
***Rapport TP8***

TP8 : *Optimisation par index bitmap et fragmentation horizontale.*

**b14d35 3117**

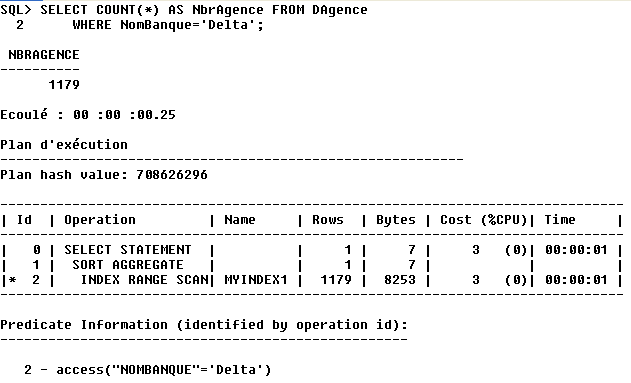
1. Requête **R1** (nombre d’agences d’une banque donnée).



1. Index **b-arbre** sur la table ***DAgence*** (***NomBanque***).

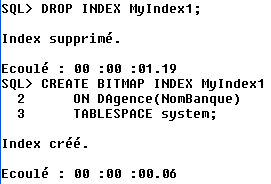


1. Réexécution de **R1**.

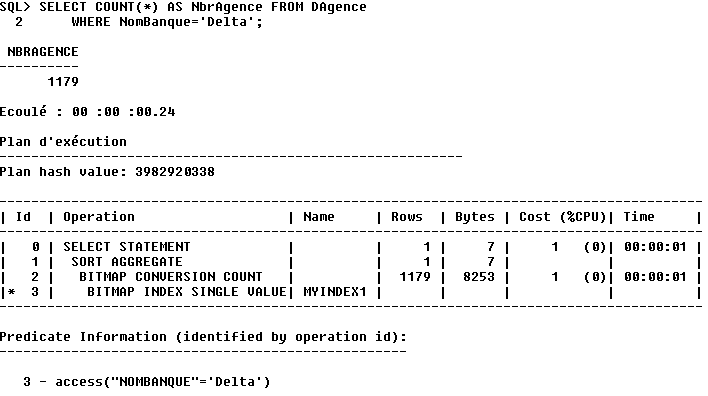


***Comparaison :*** On voit que le temps de la requête a diminué lorsqu’on rajoute l’index **b-arbre**. Avec l’index moins de lignes ont été chargées donc moins de de données a chargé ce qu’il fait qu’elle s’exécute plus rapidement.

1. Suppression de l’index **b-arbre** en le remplaçant avec un index **bitmap**.

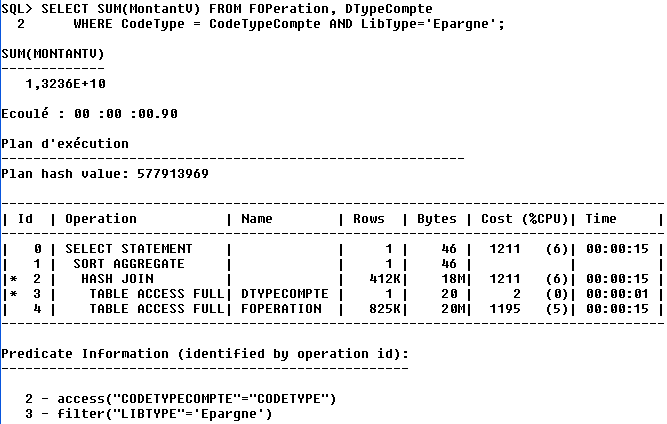


1. Réexécution de **R1**.

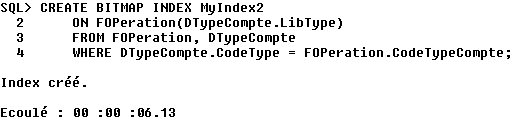


***Comparaison :*** On changeant l’index vers un de type **bitmap** le temps est resté presque le même comparé à la 2ème exécution, mais on voit que le CPU est moins utilisé que cette dernière. La requête est plus optimisée lorsqu’un index **bitmap** est utilisé dans notre cas.

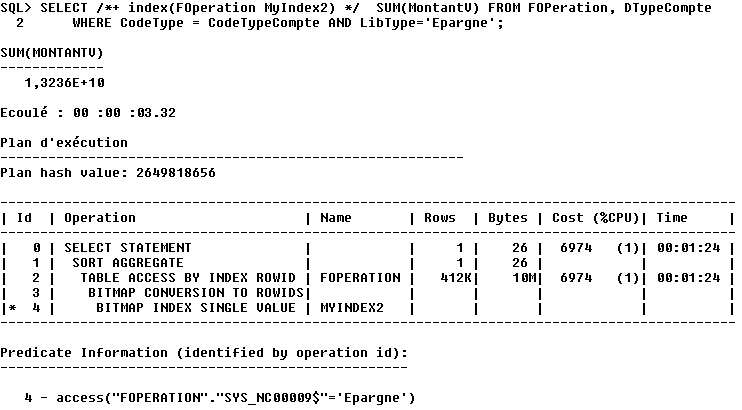
1. Requête **R2** (montant versé global dans des comptes d’épargne).



1. Index **bitmap** de **jointure** entre ***FOperation*** et ***DTypeCompte*** sur ***LibType***.

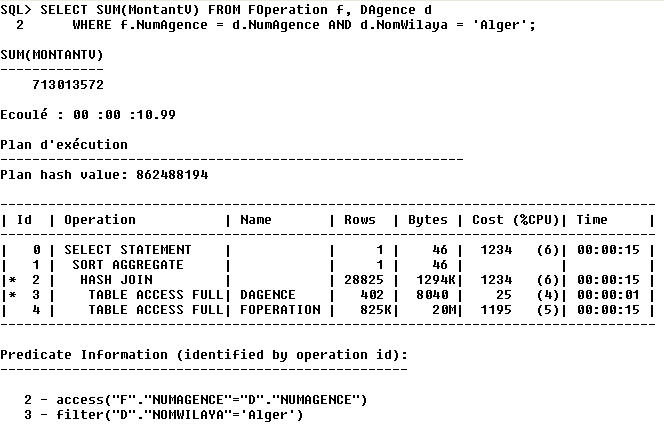


1. Réexécution de **R2**.

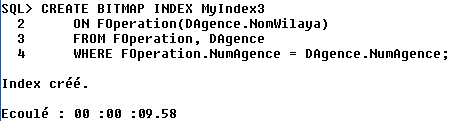


***Comparaison :*** On remarque que sans l’index de jointure, les deux tables ont été chargées pour faire une jointure tandis qu’avec l’index on a accès directe aux lignes nécessaires pour notre requête (comme si l’attribut ***LibType*** existe dans la table ***FOperation*** et on restreint selon cette attribut, plus besoin de la table ***DTypeCompte***).

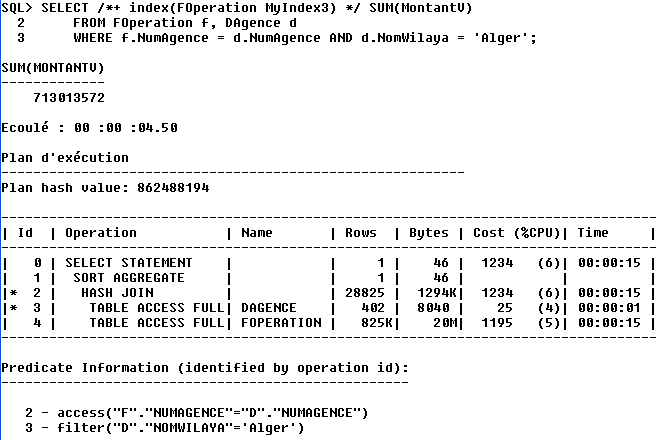
1. Requête **R3** (montant versé global dans la wilaya d’Alger).



1. Index **bitmap** de **jointure** pour améliorer le temps de la requête.

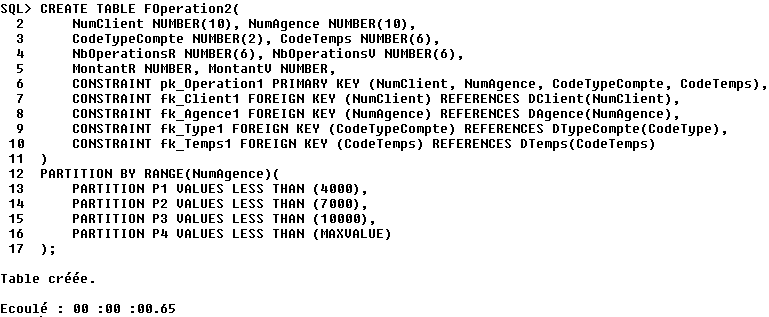


1. Réexécution de **R3**.

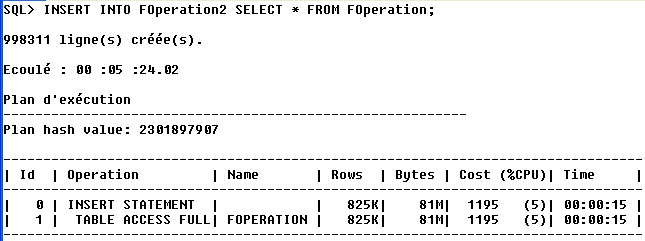


***Comparaison :*** On voit qu’il n’y a aucune différence entre l’ajout d’un index de jointure on non, même si l’on force le système à utiliser l’index, il l’ignore.

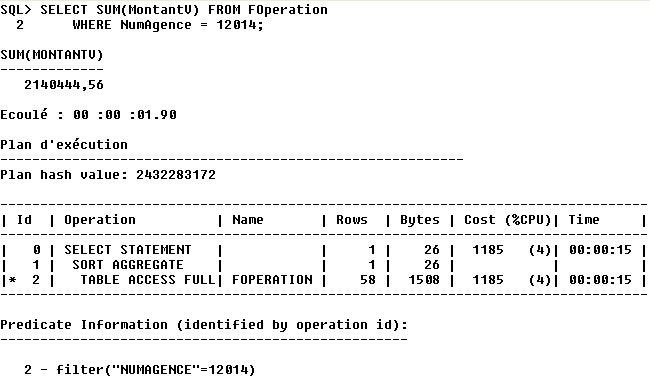
1. Fragmentation de ***FOperation*** en 4 parties selon ***NumAgence***.



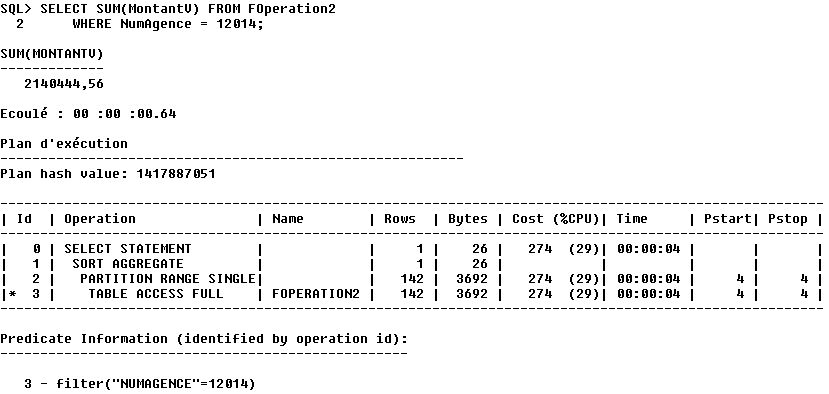
1. Remplissage de ***FOperation2*** avec le même contenu de ***FOperation***.



1. Requête **R4** (montant versé global dans l’agence *12014*) en utilisant ***FOperation***.



1. Modification de **R4** pour utiliser ***FOperation2***.



***Comparaison :*** On voit que dans la 2ème exécution, le système accède à la 4ème partition où il y’a le ***NumAgence*** ce qui permet de diminuer le nombre de tuples à accéder (les tuples de P1, P2 et P3 seront ignorés) ainsi l’optimisation de l’exécution de la requête.

***PS :*** Avant toute exécution les commandes ***ALTER SYSTEM FLUSH SHARED\_POOL*** ***et ALTER SYSTEM FLUSH BUFFER\_CACHE*** one été exécutées.