

IBM\* DB2 Universal Database\*



# Tutoriel Visual Explain

*Version 8*



IBM\* DB2 Universal Database\*



# Tutoriel Visual Explain

*Version 8*

**Important**

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section *Remarques*.

**Première édition – juillet 2002**

LE PRESENT DOCUMENT EST LIVRE "EN L'ETAT". IBM DECLINE TOUTE RESPONSABILITE, EXPRESSE OU IMPLICITE, RELATIVE AUX INFORMATIONS QUI Y SONT CONTENUES, Y COMPRIS EN CE QUI CONCERNE LES GARANTIES DE QUALITE MARCHANDE OU D'ADAPTATION A VOS BESOINS. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. Les informations qui y sont fournies sont susceptibles d'être modifiées avant que les produits décrits ne deviennent eux-mêmes disponibles. En outre, il peut contenir des informations ou des références concernant certains produits, logiciels ou services non annoncés dans ce pays. Cela ne signifie cependant pas qu'ils y seront annoncés.

Pour plus de détails, pour toute demande d'ordre technique, ou pour obtenir des exemplaires de documents IBM, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial.

Vous pouvez également consulter les serveurs Internet suivants :

- <http://www.fr.ibm.com> (serveur IBM en France)
- <http://www.can.ibm.com> (serveur IBM au Canada)
- <http://www.ibm.com> (serveur IBM aux Etats-Unis)

*Compagnie IBM France  
Direction Qualité  
Tour Descartes  
92066 Paris-La Défense Cedex 50*

© Copyright IBM France 2002. Tous droits réservés.

© Copyright International Business Machines Corporation 2000 - 2002. All rights reserved.

---

# Table des matières

**Avis aux lecteurs canadiens . . . . .** v

**A propos de ce tutoriel. . . . .** vii

Informations spécifiques aux environnements  
utilisés . . . . . ix

## **Leçon 1. Création d'images instantanées**

**EXPLAIN . . . . .** 1

Création de tables EXPLAIN . . . . . 1

Utilisation d'images instantanées EXPLAIN . . . . . 2

Création d'images instantanées EXPLAIN  
pour des instructions SQL dynamiques . . . . . 4

Création d'images instantanées EXPLAIN  
pour des instructions SQL statiques . . . . . 5

Suite du tutoriel . . . . . 5

## **Leçon 2. Affichage et utilisation d'un graphique de plan d'accès . . . . .** 7

Visualisation d'un graphique de plan d'accès  
en sélectionnant une instruction dans une liste  
d'instructions précédemment explicitées . . . . . 7

Lecture des symboles dans un graphique de  
plan d'accès . . . . . 8

Utilisation du zoom pour augmenter la  
taille des certaines parties du graphique . . . . . 8

Affichage d'informations sur les objets d'un  
graphique . . . . . 9

Affichage de statistiques pour des tables,  
des index et des fonctions de table . . . . . 10

Affichage de détails sur les opérateurs  
d'un graphique . . . . . 10

Affichage de statistiques pour les fonctions 11

Affichage de statistiques pour les espaces  
table . . . . . 11

Affichage de statistiques pour les colonnes  
d'une instruction SQL. . . . . 11

Affichage d'informations sur les  
paramètres de configuration et les options  
de définition d'accès . . . . . 11

Modification de l'apparence d'un graphique 12

Suite du tutoriel. . . . . 12

## **Leçon 3. Amélioration d'un plan d'accès dans un environnement de base de données monopartition. . . . .** 13

Travail avec des graphiques de plans d'accès 13

Exécution d'une requête sans index ni  
statistiques . . . . . 14

Collecte des statistiques à jour pour les  
tables et les index à l'aide de la commande

runstats . . . . . 19

Création d'index sur les colonnes utilisées  
pour joindre des tables dans une requête . . . . . 24

Création d'index supplémentaires sur les  
colonnes de table . . . . . 29

Informations connexes . . . . . 31

## **Leçon 4. Amélioration d'un plan d'accès dans un environnement de base de données partitionnée . . . . .** 33

Travail avec des graphiques de plans d'accès 33

Exécution d'une requête sans index ni  
statistiques . . . . . 34

Collecte des statistiques à jour pour les  
tables et les index à l'aide de la commande

runstats . . . . . 38

Création d'index sur les colonnes utilisées  
pour joindre des tables dans une requête . . . . . 42

Création d'index supplémentaires sur les  
colonnes de table . . . . . 47

Informations connexes . . . . . 50

## **Annexe A. Concepts Visual Explain . . . . .** 51

Plan d'accès . . . . . 51

Graphique de plan d'accès . . . . . 52

Noeud d'un graphique de plan d'accès . . . . . 53

Classification. . . . . 53

Conteneur . . . . . 53

Coût . . . . . 54

Groupage de lignes . . . . . 54

Espace table DMS (Database-managed space) 55

SQL dynamique. . . . . 55

Image instantanée EXPLAIN . . . . . 56

Instruction explicitable . . . . . 56

Instruction explicitée . . . . . 57

Opérande . . . . . 57

Opérateur. . . . . 57

CMPEXP . . . . .	59	DELETE . . . . .	79
DELETE . . . . .	59	EISCAN . . . . .	79
EISCAN . . . . .	59	FETCH. . . . .	80
FETCH. . . . .	60	FILTER. . . . .	80
FILTER. . . . .	60	GENROW. . . . .	81
GENROW. . . . .	61	GRPBY. . . . .	81
GRPBY. . . . .	61	HSJOIN . . . . .	82
HSJOIN . . . . .	62	INSERT . . . . .	82
INSERT . . . . .	62	IXAND . . . . .	83
IXAND . . . . .	63	IXSCAN . . . . .	84
IXSCAN . . . . .	64	MSJOIN . . . . .	85
MSJOIN . . . . .	65	NLJOIN . . . . .	85
NLJOIN . . . . .	65	PIPE . . . . .	86
PIPE . . . . .	66	RETURN . . . . .	86
RETURN . . . . .	66	RIDSCN . . . . .	87
RIDSCN . . . . .	67	RQUERY . . . . .	87
RQUERY . . . . .	67	SORT . . . . .	87
SORT . . . . .	67	TBSCAN . . . . .	88
TBSCAN . . . . .	68	TEMP . . . . .	89
TEMP . . . . .	69	TQUEUE . . . . .	90
TQUEUE . . . . .	70	UNION . . . . .	90
UNION . . . . .	70	UNIQUE . . . . .	90
UNIQUE . . . . .	70	UPDATE . . . . .	91
UPDATE . . . . .	71		
Optimiseur . . . . .	71	<b>Annexe C. DB2 concepts . . . . .</b>	<b>93</b>
Module . . . . .	71	Bases de données . . . . .	93
Prédicat . . . . .	71	Schémas . . . . .	93
Classe d'optimisation de requêtes . . . . .	72	Tables . . . . .	94
Sélectivité des prédicats . . . . .	73		
Jointure en étoile . . . . .	74	<b>Annexe D. Remarques . . . . .</b>	<b>95</b>
SQL statique . . . . .	75	Marques . . . . .	98
Espace table SMS (System-managed space). . . . .	75		
Espace table . . . . .	76	<b>Index . . . . .</b>	<b>101</b>
Visual Explain . . . . .	76		
		<b>Contacteur IBM . . . . .</b>	<b>103</b>
<b>Annexe B. Liste alphabétique des</b>		Infos produit . . . . .	103
<b>opérateurs Visual Explain . . . . .</b>	<b>79</b>		
CMPEXP . . . . .	79		

---

## Avis aux lecteurs canadiens

Le présent document a été traduit en France. Voici les principales différences et particularités dont vous devez tenir compte.

### Illustrations

Les illustrations sont fournies à titre d'exemple. Certaines peuvent contenir des données propres à la France.

### Terminologie

La terminologie des titres IBM peut différer d'un pays à l'autre. Reportez-vous au tableau ci-dessous, au besoin.

IBM France	IBM Canada
ingénieur commercial	représentant
agence commerciale	succursale
ingénieur technico-commercial	informaticien
inspecteur	technicien du matériel

### Claviers

Les lettres sont disposées différemment : le clavier français est de type AZERTY, et le clavier français-canadien de type QWERTY.

### OS/2 et Windows - Paramètres canadiens

Au Canada, on utilise :

- les pages de codes 850 (multilingue) et 863 (français-canadien),
- le code pays 002,
- le code clavier CF.

## Nomenclature

Les touches présentées dans le tableau d'équivalence suivant sont libellées différemment selon qu'il s'agit du clavier de la France, du clavier du Canada ou du clavier des États-Unis. Reportez-vous à ce tableau pour faire correspondre les touches françaises figurant dans le présent document aux touches de votre clavier.

France	Canada	Etats-Unis
 (Pos1)		Home
Fin	Fin	End
 (PgAr)		PgUp
 (PgAv)		PgDn
Inser	Inser	Ins
Suppr	Suppr	Del
Echap	Echap	Esc
Attn	Intrp	Break
Impr écran	ImpEc	PrtSc
Verr num	Num	Num Lock
Arrêt défil	Défil	Scroll Lock
 (Verr maj)	FixMaj	Caps Lock
AltGr	AltCar	Alt (à droite)

## Brevets

Il est possible qu'IBM détienne des brevets ou qu'elle ait déposé des demandes de brevets portant sur certains sujets abordés dans ce document. Le fait qu'IBM vous fournisse le présent document ne signifie pas qu'elle vous accorde un permis d'utilisation de ces brevets. Vous pouvez envoyer, par écrit, vos demandes de renseignements relatives aux permis d'utilisation au directeur général des relations commerciales d'IBM, 3600 Steeles Avenue East, Markham, Ontario, L3R 9Z7.

## Assistance téléphonique

Si vous avez besoin d'assistance ou si vous voulez commander du matériel, des logiciels et des publications IBM, contactez IBM direct au 1 800 465-1234.

---

## A propos de ce tutoriel

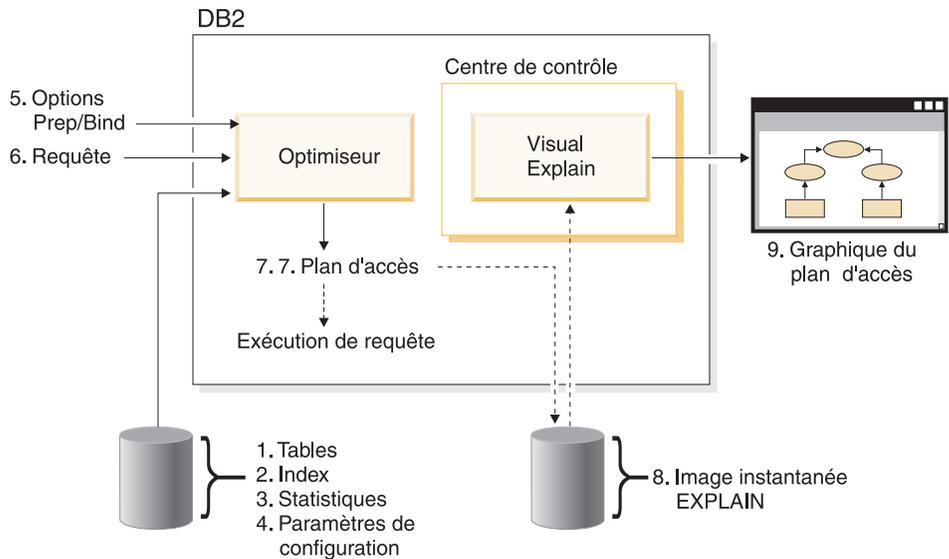
Ce tutoriel constitue un véritable tour d'horizon des fonctionnalités de DB2 Visual Explain. Grâce aux leçons de ce tutoriel, vous apprendrez comment Visual Explain permet de visualiser sous forme de graphique un plan d'accès associé à une instruction SQL. Vous apprendrez également à utiliser les informations fournies par le graphique pour adapter vos instructions SQL à des fins d'optimisation des performances.

Grâce à son optimiseur, DB2 examine vos requêtes SQL et détermine le meilleur accès à vos données. Ce chemin d'accès aux données est appelé plan d'accès. DB2 vous permet de voir ce que l'optimiseur a effectué en visualisant un plan d'accès sélectionné pour traiter une requête SQL en particulier. Vous pouvez utiliser Visual Explain pour afficher un plan d'accès sous forme de graphique. Ce graphique est une représentation visuelle des objets de base de données impliqués dans une requête (par exemple, des tables et des index). Il présente également les opérations effectuées sur ces objets (par exemple, des balayages et des tris) et montre le flot de données.

Vous pouvez améliorer l'accès d'une requête à des données en effectuant une des opérations d'optimisation suivantes :

1. Modifiez la conception de votre table et réorganisez les données qu'elle contient.
2. Créez des index adéquats.
3. Utilisez la commande **runstats** pour que l'optimiseur dispose de statistiques à jour.
4. Choisissez des paramètres de configuration adéquats.
5. Sélectionnez des options adéquates de définition d'accès.
6. Concevez des requêtes de sorte qu'elles ne récupèrent que les données requises.
7. Travaillez avec un plan d'accès.
8. Créez des images instantanées EXPLAIN.
9. Utilisez un graphique de plan d'accès pour améliorer un plan d'accès.

Ces opérations d'optimisation des performances correspondent à celles qui sont indiquées dans l'illustration suivante (les lignes en pointillé indiquent des actions obligatoires pour Visual Explain).



Ce tutoriel comprend leçons relatives à :

- La création d'images instantanées EXPLAIN. Celles-ci sont obligatoires pour pouvoir afficher des graphiques de plan d'accès.
- L'affichage et la manipulation d'un graphique de plan d'accès.
- L'exécution d'opérations d'optimisation et leur analyse pour savoir comment celle-ci améliorent votre plan d'accès.

**Remarque :** L'optimisation des performances est traitée dans deux leçons distinctes : une concernant les environnements de bases de données monopartition, et un pour les environnement de bases de données partitionnées.

Vous utiliserez la base de données exemple DB2 pour ce tutoriel.

Reportez-vous au manuel *Administration Guide* si vous n'avez pas encore créé de base de données exemple.

---

## Informations spécifiques aux environnements utilisés



Les informations introduites par cette icône concernent uniquement les environnements de base de données monopartition.



Les informations introduites par cette icône concernent uniquement les environnements de base de données partitionnée.



---

## Leçon 1. Création d'images instantanées EXPLAIN

Dans cette leçon, vous apprendrez à créer des images instantanées EXPLAIN. SQL EXPLAIN est utilisé pour la capture d'informations concernant l'environnement dans lequel une instruction SQL statique ou dynamique est compilée. Les informations capturées vous permettent de comprendre la structure et les performances potentielles de vos instructions SQL. Une image instantanée EXPLAIN se compose de données comprimées, recueillies lors de l'explicitation d'une instruction SQL. L'image instantanée est stockée sous la forme d'un objet BLOB (Binary Large Object) dans la table EXPLAIN\_STATEMENT et contient les informations suivantes :

- représentation interne du plan d'accès, y compris les opérateurs, ainsi que les tables et les index vers lesquels un accès est établi ;
- critères utilisés par l'optimiseur pour l'établissement du plan d'accès, y compris les statistiques sur les objets de base de données et le coût cumulé de chaque opération.

Pour pouvoir afficher un graphique de plan d'accès, Visual Explain requiert les informations contenues dans une image instantanée EXPLAIN.

---

### Création de tables EXPLAIN

Pour créer des images instantanées EXPLAIN, vérifiez que les tables EXPLAIN suivantes existent pour votre ID utilisateur :

- EXPLAIN\_INSTANCE
- EXPLAIN\_STATEMENT

Pour ce faire, lancez la commande DB2 **LIST TABLES**. Si ces tables n'existent pas, vous devez les créer à l'aide des instructions suivantes :

1. Si DB2 n'a pas encore été lancé, entrez la commande **db2start**.
2. A partir de l'invite de l'interpréteur de commandes DB2, connectez-vous à la base de données que vous voulez utiliser. Pour ce tutoriel, connectez-vous à la base de données exemple en lançant la commande **CONNECT TO SAMPLE**.
3. Créez les tables EXPLAIN à l'aide du modèle de fichier de commandes EXPLAIN.DDL qui se trouve dans le répertoire sqllib\misc. Pour exécuter ce fichier, passez dans ce répertoire et entrez la commande suivante : **db2 -tf EXPLAIN.DDL**. Le nom des tables créées par ce fichier de commandes commence par l'ID de l'utilisateur connecté. Cet ID doit disposer du privilège CREATETAB sur la base de données, ou des droits SYSADM ou DBADM.

---

## Utilisation d'images instantanées EXPLAIN

Quatre images instantanées exemples sont fournies pour que vous puissiez vous familiariser avec Visual Explain. Des informations sur la création de vos images instantanées sont fournies dans les sections suivantes, mais vous n'avez pas besoin de créer vos propres images instantanées pour progresser dans ce tutoriel.

- Création d'images instantanées EXPLAIN pour des instructions SQL dynamiques
- Création d'images instantanées EXPLAIN pour des instructions SQL statiques

La requête utilisée pour les images instantanées exemples répertorie le nom, le service et la rémunération de tous les employés non-cadres qui gagnent plus de 90 % du salaire du cadre le mieux payé.

```
SELECT S.ID,S.NAME,O.DEPTNAME,SALARY+COMM
FROM ORG O, STAFF S
WHERE
  O.DEPTNUMB = S.DEPT AND
  S.JOB <> 'Mgr' AND
  S.SALARY+S.COMM > ALL( SELECT ST.SALARY*.9
                        FROM STAFF ST
                        WHERE ST.JOB='Mgr' )
ORDER BY S.NAME
```

La requête est constituée de deux parties :

1. La sous-requête (entre parenthèses) fournit des lignes de données qui présentent les 90 % du salaire de chaque cadre. Comme la sous-requête est qualifiée par ALL, seule la valeur la plus importante de cette table est récupérée.
2. La requête principale joint toutes les lignes des tables ORG et STAFF où les numéros de services sont identiques. JOB n'équivaut pas à 'Mgr' et le salaire plus les commissions est plus grand que la valeur renvoyée par la sous-requête.

La requête principale contient les trois prédicats suivants (comparaisons):

1. O.DEPTNUMB = S.DEPT
2. S.JOB <> 'Mgr'
3. S.SALARY+S.COMM > ALL ( SELECT ST.SALARY\*.9
 FROM STAFF ST
 WHERE ST.JOB='Mgr' )

Ces prédicats représentent respectivement :

1. un prédicat de jointure, qui joint les lignes de données des tables ORG et STAFF où les numéros de services sont identiques ;
2. un prédicat local dans la colonne JOB de la table STAFF ;
3. un prédicat local dans les colonnes SALARY et COMM de la table STAFF qui utilise le résultat de la sous-requête.

Pour charger les images instantanées exemples :

1. Si DB2 n'a pas encore été lancé, entrez la commande **db2start**.
2. Assurez-vous que les tables EXPLAIN existent dans votre base de données. Pour ce faire, suivez les instructions de la section Création de tables EXPLAIN.
3. Connectez-vous à la base de données que vous souhaitez utiliser. Dans ce tutoriel, vous vous connecterez à la base de données exemple. Pour ce faire, lancez la commande **connect to sample** à l'invite CLP DB2.

Si la base de données exemple n'existe pas, reportez-vous à la section du manuel décrivant son installation (*Administration Guide*).

4. Pour importer les images instantanées prédéfinies, exécutez le fichier de commandes DB2 VESAMPL.DDL.

-  Ce fichier se trouve dans le répertoire sqllib\samples\ve.
-  Ce fichier se trouve dans le répertoire sqllib\samples\ve\inter.

Pour exécuter ce fichier, passez dans le répertoire adéquat et entrez la commande suivante : **db2 -tf vesampl.ddl**.

- Ce fichier de commandes doit être exécuté à l'aide de l'ID utilisateur utilisé pour la création des tables EXPLAIN.
- Ce fichier de commandes permet d'importer uniquement les images instantanées prédéfinies. Il ne crée pas de tables ou de données. Les opérations d'optimisation décrites ultérieurement (CREATE INDEX et runstats par exemple), seront exécutées sur les tables et les données de la base de données exemple.

Vous êtes désormais prêt à afficher et utiliser les graphiques du plan d'accès.

## Création d'images instantanées EXPLAIN pour des instructions SQL dynamiques

**Remarque :** Les informations relatives à la création d'images instantanées EXPLAIN contenues dans cette section sont fournies à titre de référence. Comme vous disposez d'images instantanées exemples EXPLAIN, il n'est pas nécessaire d'effectuer cette opération pour progresser dans ce tutoriel.

Suivez les étapes suivantes pour créer une image instantanée EXPLAIN pour une instruction SQL dynamique :

1. Si DB2 n'a pas encore été lancé, entrez la commande **db2start**.
2. Assurez-vous que les tables EXPLAIN existent dans votre base de données. Pour ce faire, suivez les instructions de la section Création de tables EXPLAIN.
3. A partir de l'invite de l'interpréteur de commandes DB2, connectez-vous à la base de données que vous voulez utiliser. Par exemple, pour vous connecter à la base de données exemple, lancez la commande **CONNECT TO SAMPLE**.

Pour créer la base de données exemple, reportez-vous à la section du manuel décrivant son installation (*Administration Guide*).

4. Créez une image instantanée pour une instruction SQL dynamique, à l'aide des commandes suivantes à partir de l'invite CLP de DB2 :
  - Pour créer une image instantanée EXPLAIN sans exécuter d'instruction SQL, lancez la commande **set current explain snapshot=explain**.
  - Pour créer une image instantanée EXPLAIN et exécuter une instruction SQL, lancez la commande **set current explain snapshot=yes**.

Cette commande permet de définir le registre spécial EXPLAIN. Une fois le registre défini, toutes les instructions SQL suivantes seront affectées. Pour plus d'informations, reportez-vous aux sections relatives aux images instantanées actuelles (*SQL Reference*).

5. Soumettez vos instructions SQL à partir de l'invite CLP de DB2.
6. Pour voir le graphique du plan d'accès pour l'image instantanée, actualisez la fenêtre Historique des instructions explicitées (disponible à partir du Centre de contrôle), et double-cliquez sur l'image instantanée.
7. Facultatif. Pour désactiver la fonction d'image instantanée, lancez la commande **set current explain snapshot=no** une fois que vos instructions SQL ont été soumises.

## Création d'images instantanées EXPLAIN pour des instructions SQL statiques

**Remarque :** Les informations relatives à la création d'images instantanées EXPLAIN contenues dans cette section sont fournies à titre de référence. Comme vous disposez d'images instantanées exemples EXPLAIN, il n'est pas nécessaire d'effectuer cette opération pour progresser dans ce tutoriel.

Effectuez les étapes suivantes pour créer une image instantanée EXPLAIN pour une instruction SQL statique :

1. Si DB2 n'a pas encore été lancé, entrez la commande **db2start**.
2. Assurez-vous que les tables EXPLAIN existent dans votre base de données. Pour ce faire, suivez les instructions de la section Création de tables EXPLAIN.
3. A partir de l'invite de l'interpréteur de commandes DB2, connectez-vous à la base de données que vous voulez utiliser. Par exemple, pour vous connecter à la base de données exemple, lancez la commande **CONNECT TO SAMPLE**.
4. Créez une image instantanée pour une instruction SQL statique à l'aide de l'option EXPLSNAP lorsque vous liez ou préparez votre application. Par exemple, lancez la commande **bind votre fichier explsnap yes**.
5. Facultatif. Pour voir le graphique du plan d'accès pour l'image instantanée en question, actualisez la fenêtre Historique des instructions explicitées (disponible à partir du Centre de contrôle), et double-cliquez sur l'image instantanée.

Pour plus d'informations sur l'utilisation de l'option EXPLSNAP pour des API équivalentes, reportez-vous aux sections relatives à chacune d'entre elles (*Application Development Guide*).

---

### Suite du tutoriel

Dans la «Leçon 2. Affichage et utilisation d'un graphique de plan d'accès» à la page 7, vous apprendrez comment visualiser un graphique de plan d'accès pour mieux en saisir le contenu.



---

## Leçon 2. Affichage et utilisation d'un graphique de plan d'accès

Dans cette leçon, vous utiliserez la fenêtre Graphique du plan d'accès pour afficher et utiliser un graphique de plan d'accès. Un graphique de plan d'accès est une représentation graphique d'un plan d'accès. A partir de ce graphique, vous pouvez visualiser tous les détails :

- des tables (et colonnes correspondantes) et des index,
- des opérateurs (tels que les balayages, les tris et les jointures de tables),
- des espaces table et des fonctions.

Vous pouvez visualiser un graphique de plan d'accès en :

- sélectionnant une instruction dans une liste d'instructions précédemment explicitées ;
- sélectionnant une instruction dans une liste d'instructions explicitables dans un ensemble ;
- en explicitant dynamiquement une instruction SQL.

Comme vous allez travailler avec les graphiques de plans d'accès pour les images instantanées exemples que vous avez chargées dans la leçon 1, vous sélectionnerez une instruction dans une liste d'instructions précédemment explicitées. Pour plus d'informations sur les autres méthodes de visualisation de graphiques de plans d'accès, reportez-vous à l'aide de Visual Explain.

---

### Visualisation d'un graphique de plan d'accès en sélectionnant une instruction dans une liste d'instructions précédemment explicitées

Pour afficher un graphique de plan d'accès en sélectionnant une instruction dans une liste d'instructions précédemment explicitées:

1. Dans le Centre de contrôle, développez l'arborescence jusqu'à ce que vous trouviez la base de données exemple.
2. Cliquez avec le bouton droit de votre souris sur la base de données et sélectionnez **Affichage de l'historique des instructions explicitées** dans le menu qui s'affiche. La fenêtre Historique des instructions explicitées s'ouvre.
3. Vous ne pouvez visualiser qu'un seul graphique de plan d'accès pour une instruction disposant d'une image instantanée EXPLAIN. OUI apparaît dans la colonne **Image instantanée EXPLAIN** pour les instructions qualifiées. Double-cliquez sur l'entrée appelée Numéro de requête 1 (il se

peut que vous ayez à faire défiler la fenêtre vers la droite pour voir la colonne **Numéro de requête**). La fenêtre Graphique du plan d'accès pour l'instruction s'ouvre.

**Remarque :** Le graphique se lit de bas en haut. La première étape de la requête est répertoriée à la fin du graphique et la dernière en haut du graphique.

### Lecture des symboles dans un graphique de plan d'accès

Le graphique de plan d'accès indique la structure d'un plan d'accès sous forme d'arborescence. Les *noeuds* de l'arborescence représentent :

- des tables, représentées par des rectangles ;
- des index, représentés par des losanges ;
- des opérateurs, représentés par des octogones. Les opérateurs TQUEUE sont représentés par des parallélogrammes ;
- des fonctions de table, représentées par des hexagones.

Pour les opérateurs, le nombre entre parenthèses apparaissant à droite du type d'opérateur est un identificateur unique pour chaque noeud. Le nombre situé en-dessous du type d'opérateur représente le coût cumulatif.

### Utilisation du zoom pour augmenter la taille des certaines parties du graphique

Lorsque vous affichez un graphique de plan d'accès, vous accédez à l'ensemble du graphique. Par conséquent, il se peut que vous ne soyez pas en mesure de voir les détails permettant de distinguer chaque noeud.

Dans la fenêtre Graphique du plan d'accès, utilisez le **zoom** pour augmenter la taille de différentes parties d'un graphique :

1. Placez le pointeur de la souris sur le curseur de la barre de zoom dans la partie gauche du graphique.
2. Cliquez sur le zoom et faites glisser le curseur jusqu'à ce que la partie du graphique soit à la taille voulue.

Pour visualiser différentes parties du graphique, utilisez la barre de défilement.

Pour visualiser un graphique de plan d'accès large et complexe, utilisez la fenêtre Vue générale du graphique. Vous pouvez utiliser cette fenêtre pour connaître la partie du graphique vous visualisez, pour augmenter la taille d'une partie de graphique ou pour le parcourir. Le zoom est indiqué dans le plan d'accès.



## Affichage de statistiques pour des tables, des index et des fonctions de table

Pour visualiser les statistiques de catalogue pour une seule table (rectangle), un seul index (losange) ou une seule fonction de table (hexagone) d'un graphique, double-cliquez sur le noeud approprié. Une fenêtre Statistiques s'ouvre pour les objets sélectionnés. Elle affiche des informations relatives aux statistiques en vigueur au moment où l'image instantanée a été créée, ainsi que des informations relatives aux statistiques qui existent actuellement dans les tables de catalogue système.

Pour afficher les statistiques catalogue pour *plusieurs* tables, index ou fonctions de table d'un graphique, sélectionnez-les en cliquant dessus (s'ils sont mis en évidence). Sélectionnez ensuite **Noeud**→**Affichage des statistiques**. Une fenêtre Statistiques s'ouvre pour chacun des objets sélectionnés (les fenêtres peuvent se chevaucher et vous aurez peut-être à les faire glisser pour pouvoir toutes les voir.)

Si l'entrée **STATS\_TIME** de la colonne **Explicitée** indique **Statistiques non mises à jour**, aucune statistique n'existait lorsque l'optimiseur a créé le plan d'accès. Par conséquent, si l'optimiseur avait besoin de certaines statistiques pour créer un plan d'accès, il a utilisé les valeurs par défaut. Si des statistiques par défaut ont été utilisées par l'optimiseur, elles sont identifiées par (**par défaut**) dans la colonne Explicitée.

## Affichage de détails sur les opérateurs d'un graphique

Pour voir les statistiques catalogue d'un opérateur unique (octogone), double-cliquez sur le noeud approprié. Une fenêtre Détails de l'opérateur s'ouvre pour l'opérateur sélectionné et indique des informations telles que :

- le coût cumulé estimé (E-S, instructions CPU et coût total) ;
- la cardinalité (c'est-à-dire, le nombre approximatif de lignes recherchées) ;
- les tables accédées et jointes dans le plan ;
- les colonnes de ces tables ;
- les prédicats qui ont été appliqués, ainsi que leur sélectivité sélectionnée ;
- les arguments d'entrée pour chaque opérateur.

Pour voir les détails de *plusieurs* opérateurs, sélectionnez-les en cliquant dessus (s'ils sont mis en évidence). Sélectionnez ensuite **Noeud**→**Afficher les détails**. Une fenêtre Statistiques s'ouvre pour chacun des objets sélectionnés (les fenêtres peuvent se chevaucher et vous aurez peut-être à les faire glisser pour pouvoir toutes les voir.)

## Affichage de statistiques pour les fonctions

Pour voir les statistiques catalogue de fonctions intégrées et de fonction définies par l'utilisateur, sélectionnez **Instruction**→**Affichage des statistiques**→**Fonctions**. Sélectionnez une ou plusieurs entrées dans la liste qui s'affiche dans la fenêtre Fonctions, puis cliquez sur **OK**. Une fenêtre Statistiques sur fonction s'ouvre pour chacune des fonctions sélectionnées.

## Affichage de statistiques pour les espaces table

Pour voir les statistiques catalogue des espaces table, sélectionnez **Instruction**→**Affichage des statistiques**→**Espaces table**. Sélectionnez une ou plusieurs entrées dans la liste qui s'affiche dans la fenêtre Espaces table, puis cliquez sur **OK**. Une fenêtre Statistiques sur espace table s'ouvre pour chacun des espaces table sélectionnés.

## Affichage de statistiques pour les colonnes d'une instruction SQL

Pour voir des statistiques des colonnes référencées dans une instruction SQL :

1. Double-cliquez sur la table dans le graphique de plan d'accès. La fenêtre Statistiques sur table s'affiche.
2. Cliquez sur le bouton **Colonnes référencées**. La fenêtre Colonnes référencées s'ouvre, répertoriant les colonnes de la table.
3. Sélectionnez une ou plusieurs colonnes dans la liste, et cliquez sur **OK**. Une fenêtre Statistiques sur colonne référencée s'ouvre pour chaque colonne sélectionnée.

## Affichage d'informations sur les paramètres de configuration et les options de définition d'accès

Pour voir des informations sur les paramètres de configuration et sur les options de définition d'accès (paramètres d'optimisation), sélectionnez **Instruction**→**Affichage des paramètres d'optimisation** dans la fenêtre Graphique du plan d'accès. La fenêtre Paramètres d'optimisation s'ouvre et affiche des informations concernant les valeurs de paramètres en vigueur au moment où l'image instantanée a été créée, ainsi que des informations sur les valeurs actuelles.

---

## Modification de l'apparence d'un graphique

Pour modifier les diverses caractéristiques d'un graphique, procédez comme suit :

1. Dans la fenêtre Graphique de plan d'accès, sélectionnez **Afficher les paramètres**. Le bloc-notes Plan d'accès - Paramètres du graphique s'affiche.
2. Pour modifier la couleur d'arrière-plan, sélectionnez l'onglet Graphique.
3. Pour modifier la couleur des différents opérateurs, utilisez les onglets Base, Etendu, Mise à jour, et Divers.
4. Pour modifier la couleur d'une table, d'un index ou de noeuds de fonction de tables, sélectionnez l'onglet Opérande.
5. Pour spécifier le type d'informations indiqué dans les noeuds d'opérateurs (type de coût ou cardinalité, qui correspond au nombre de lignes approximatif), choisissez l'onglet Opérateur.
6. Pour spécifier les noms de schéma ou les ID utilisateur devant être indiqués dans les noeuds de table, sélectionnez l'onglet Opérande.
7. Pour spécifier si les noeuds doivent apparaître en deux ou en trois dimensions, sélectionnez l'onglet Noeud.
8. Pour mettre à jour le graphique avec les options choisies et sauvegarder la configuration, cliquez sur **Appliquer**.

---

## Suite du tutoriel

Si vous travaillez dans un environnement de base de données monopartition, reportez-vous à la section «Leçon 3. Amélioration d'un plan d'accès dans un environnement de base de données monopartition» à la page 13, où vous apprendrez comment les différentes opérations d'optimisation des performances peuvent modifier et améliorer un plan d'accès.

Si vous travaillez dans un environnement de base de données partitionné, reportez-vous à la section «Exécution d'une requête sans index ni statistiques» à la page 14, où vous apprendrez comment les différentes opérations d'optimisation des performances peuvent modifier et améliorer un plan d'accès.

---

## Leçon 3. Amélioration d'un plan d'accès dans un environnement de base de données monopartition

Dans cette leçon, vous apprendrez comment un plan d'accès et les fenêtres associées à la requête de base sont modifiés lorsque vous optimisez les performances. À l'aide d'une série d'exemples et d'illustrations, vous apprendrez comment le coût total estimé du plan d'accès, même pour une simple requête, peut être amélioré en utilisant la commande **runstats** et en ajoutant des index.

Au fur et à mesure que vous vous familiariserez avec Visual Explain, vous découvrirez d'autres façons d'optimiser les requêtes.

---

### Travail avec des graphiques de plans d'accès

À l'aide de quatre images instantanées EXPLAIN exemples, vous apprendrez comment l'optimisation joue un rôle essentiel quant aux performances de la base de données.

Les requêtes associées aux images instantanées EXPLAIN sont numérotées de 1 à 4. Chaque requête utilise la même instruction SQL (décrite dans la Leçon 1) :

```
SELECT S.ID,S.NAME,O.DEPTNAME,SALARY+COMM
FROM ORG O, STAFF S
WHERE
  O.DEPTNUMB = S.DEPT AND
  S.JOB <> 'Mgr' AND
  S.SALARY+S.COMM > ALL( SELECT ST.SALARY*.9
                        FROM STAFF ST
                        WHERE ST.JOB='Mgr' )
ORDER BY S.NAME
```

Chaque itération de la requête utilise davantage de techniques d'optimisation que la dernière exécution. Par exemple, la requête 1 n'a pas été optimisée, alors que la requête 4 a été optimisée au maximum. Les différences entre les requêtes apparaissent ci-dessous :

#### Requête 1

Exécution d'une requête sans index ni statistiques

#### Requête 2

Collecte des statistiques à jour pour les tables et les index dans une requête

### Requête 3

Création d'index sur les colonnes utilisées pour joindre des tables dans une requête

### Requête 4

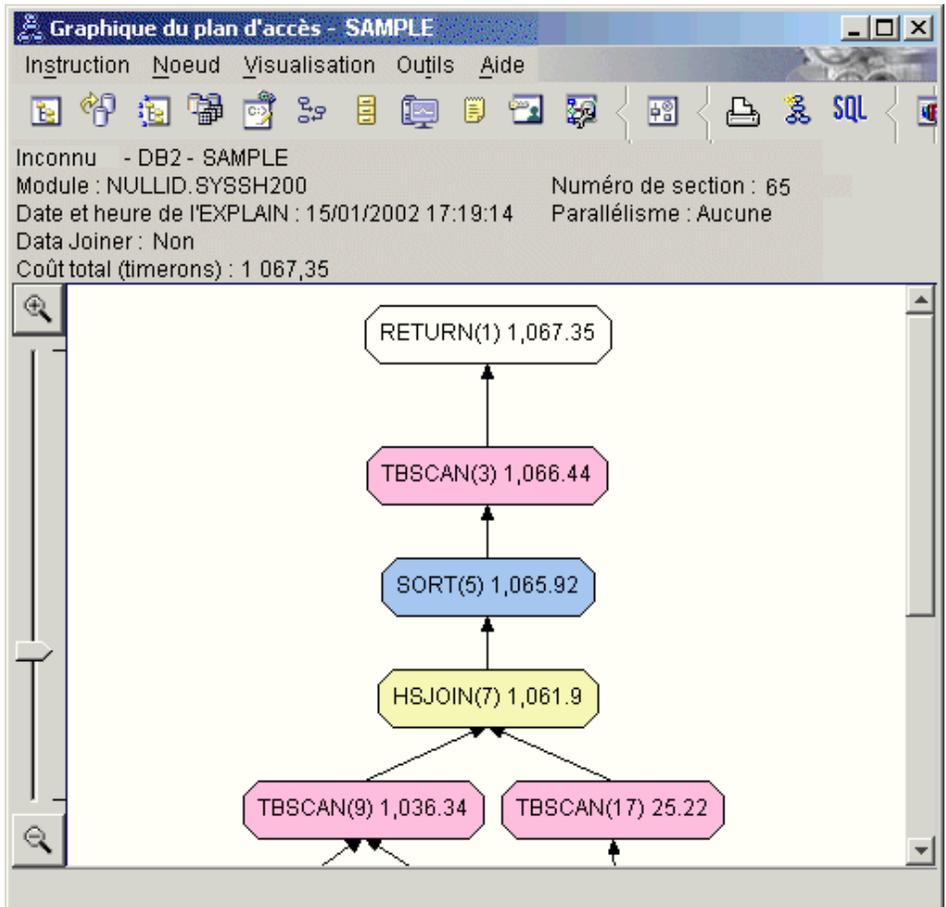
Création d'index supplémentaires sur les colonnes de table

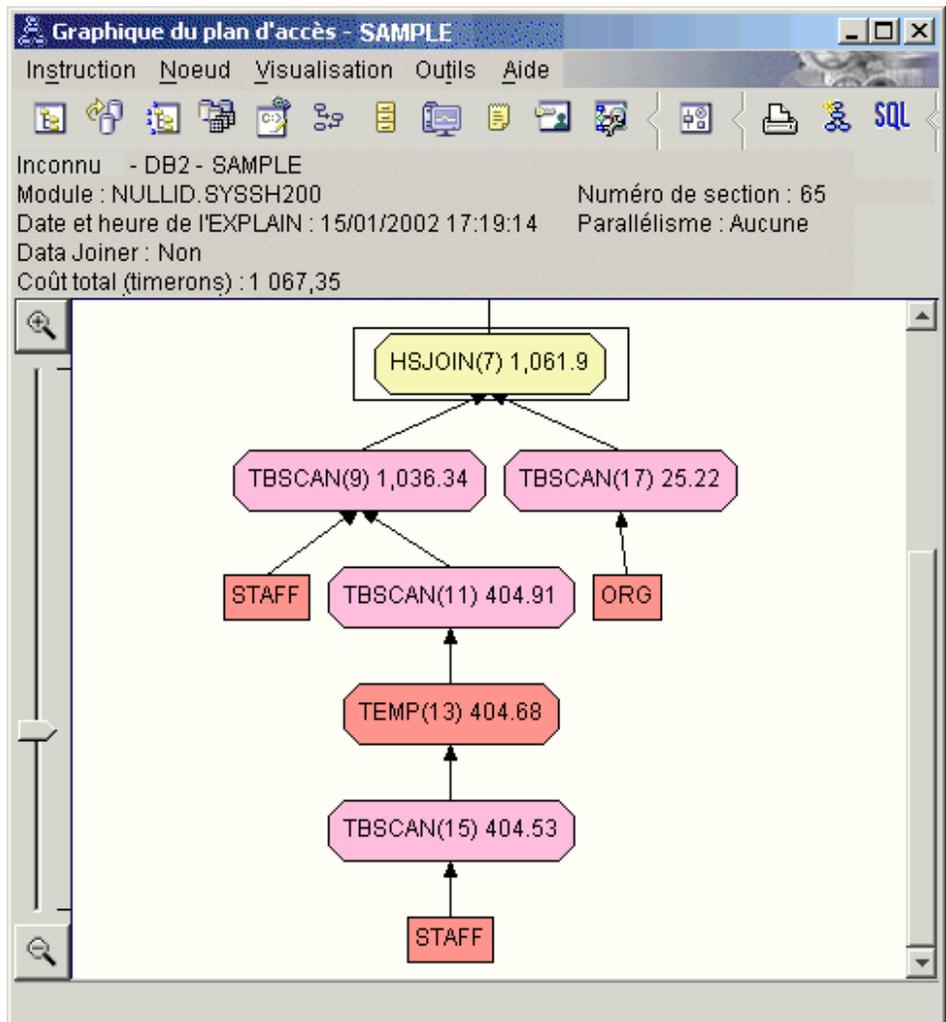
## Exécution d'une requête sans index ni statistiques

Dans cet exemple, le plan d'accès a été créé pour la requête SQL sans index ni statistiques.

Pour visualiser le graphique de plan d'accès pour cette requête (Requête 1), procédez comme suit :

1. Dans le Centre de contrôle, développez l'arborescence jusqu'à ce que vous trouviez la base de données exemple.
2. Cliquez avec le bouton droit de votre souris sur la base de données et sélectionnez **Affichage de l'historique des instructions explicitées** dans le menu qui s'affiche. La fenêtre Historique des instructions explicitées s'ouvre.
3. Double-cliquez sur l'entrée appelée Numéro de requête 1 (il se peut que vous ayez à faire défiler la fenêtre vers la droite pour voir la colonne **Numéro de requête**). La fenêtre Graphique du plan d'accès pour l'instruction s'ouvre.





En répondant aux questions suivantes, vous comprendrez mieux comment améliorer une requête.

1. Les statistiques à jour existent-elles pour chaque table de la requête ?

Pour vérifier si les statistiques à jour existent pour chaque table de la requête, double-cliquez sur chaque noeud de table dans le graphique du plan d'accès. Dans la fenêtre Statistiques sur table qui s'ouvre, la ligne **STATS\_TIME** sous la colonne **Explicitée** contient la phrase "Statistiques non mises à jour" si aucune statistique n'a été collectée au moment où l'image instantanée a été créée.

Si des statistiques à jour n'existent pas, l'optimiseur utilise les statistiques par défaut, qui peuvent différer des vraies statistiques. Les statistiques par

défaut sont identifiées par les mots "par défaut" en dessous de la colonne **Explicitée** de la fenêtre Statistiques sur table.

En fonction des informations de la fenêtre Statistiques sur table pour la table ORG, l'optimiseur a utilisé les statistiques par défaut (comme indiqué à côté des valeurs explicitées). Les statistiques par défaut ont été utilisées parce que les statistiques à jour n'étaient pas disponibles au moment où l'image instantanée a été créée (comme indiqué à la ligne **STATS\_TIME**).

Statistiques	Valeurs explicitées	Valeurs en cours
CREATE_TIME	03/26/2002 1:35:42	03/26/2002 1:35:42
STATS_TIME	04/03/2002 5:24:55	04/03/2002 5:24:55
CARD	4	8
NPAGES	1	2
FPAGES	1	2
COLCOUNT	5	5
OVERFLOW	0	0
TABLESPACE	USERSPACE1	USERSPACE1
INDEX_TABLESPACE		
LONG_TABLESPACE		
VOLATILE	Non	Non

2. Ce plan d'accès utilise-t-il les méthodes les plus efficaces pour l'accès aux données ?

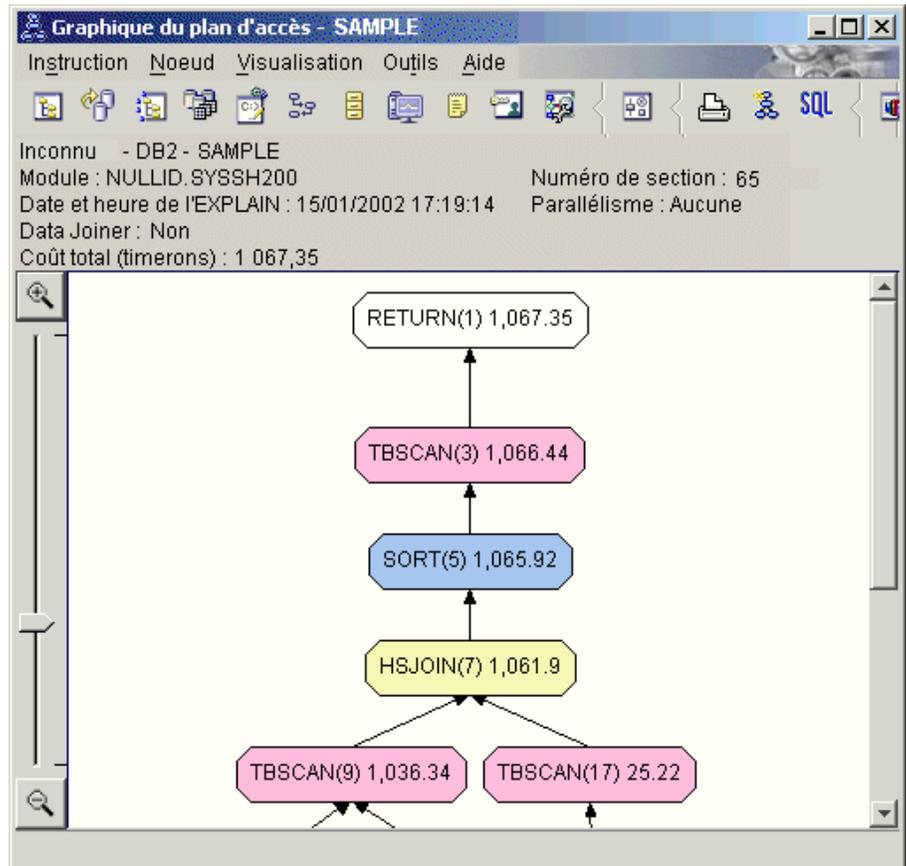
Ce plan d'accès contient des balayages de tables, et non des balayages d'index. Les balayages de table sont indiqués par des octogones appelés TBSCAN. Si des balayages d'index avaient été utilisés, ils seraient apparus sous la forme de losanges appelés IXSCAN. L'utilisation d'un index créé pour une table est plus efficace d'un point de vue qualité/prix qu'un balayage de table si de petits volumes de données sont extraits.

3. Ce plan d'accès est-il vraiment efficace ?

Vous ne pouvez connaître l'efficacité d'un plan d'accès que s'il est basé sur des statistiques à jour. Comme l'optimiseur a utilisé les statistiques par défaut dans le plan d'accès, vous ne pouvez pas déterminer l'efficacité du plan.

En général, vous devez noter le coût total approximatif pour le plan d'accès pour une comparaison ultérieure avec les plans d'accès révisés. Le coût répertorié dans chaque noeud est cumulatif, des premières étapes de votre requête jusqu'au noeud inclus.

Dans la fenêtre Graphique du plan d'accès, le coût total est d'environ 1067 timerons (indiqué dans **RETOUR (1)** en haut du graphique). Le coût total approximatif est également indiqué dans la zone supérieure de la fenêtre.



#### 4. Suite du tutoriel

La Requête 2 ressemble à un plan d'accès pour la requête de base une fois que la commande **runstats** a été lancée. La commande **runstats** permet de fournir à l'optimiseur les statistiques à jour sur les tables accédées par la requête.

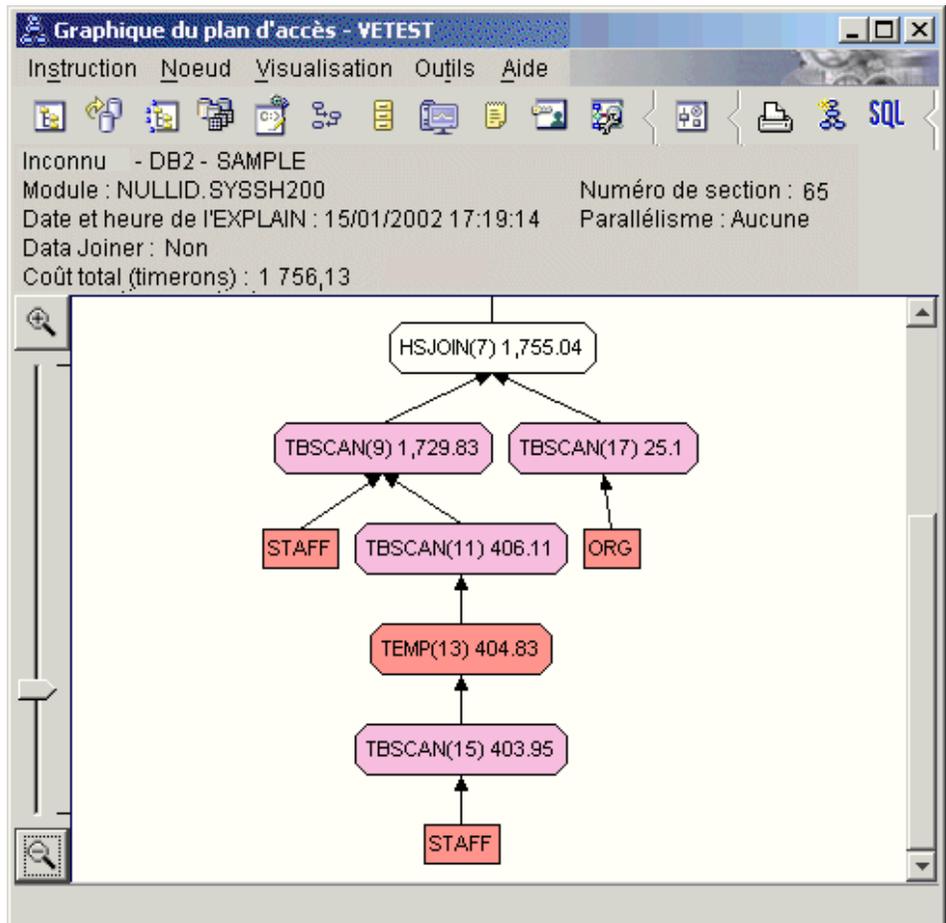
## Collecte des statistiques à jour pour les tables et les index à l'aide de la commande `runstats`

Cet exemple s'appuie sur le plan d'accès décrit dans la Requête 1 en réunissant des statistiques à jour à l'aide de la commande `runstats`.

Il est vivement recommandé de relancer la commande `RUNSTATS` pour collecter des statistiques à jour sur les tables et les index, notamment si des mises à jour importantes ont été effectuées ou si des index ont été créés depuis la dernière exécution de cette commande. L'optimiseur dispose ainsi des informations les plus récentes lui permettant de définir le meilleur plan d'accès. Si les statistiques disponibles dans le catalogue système ne sont pas à jour, l'optimiseur utilise des valeurs par défaut et risque d'établir un plan d'accès peu performant.

Lorsque vous modifiez une table, veillez à utiliser la commande `RUNSTATS` après cette mise à jour ; sinon, l'optimiseur risque de considérer que la table est vide. Pour vérifier que la commande `RUNSTATS` a bien été lancée, consultez la valeur de cardinalité affichée dans la fenêtre Détails opérateur. Si elle est égale à zéro, terminez les mises à jour de la table, exécutez de nouveau `RUNSTATS`, et recréez les images instantanées `EXPLAIN` correspondant aux tables concernées.

Pour visualiser le graphique de plan d'accès pour cette requête (Requête 2), procédez comme suit : dans la fenêtre Historique des instructions explicitées, double-cliquez sur l'entrée appelée Requête 2. La fenêtre Graphique de plan d'accès pour l'exécution de l'instruction s'ouvre.



En répondant aux questions suivantes, vous comprendrez mieux comment améliorer une requête.

1. Les statistiques à jour existent-elles pour chaque table de la requête ?

La fenêtre Statistiques sur table pour la table ORG indique que l'optimiseur a utilisé les statistiques à jour (la valeur **STATS\_TIME** correspond à l'heure réelle de collecte des statistiques). La précision des statistiques dépend de l'impact des modifications sur le contenu des tables, une fois la commande **runstats** exécutée.

Statistiques sur la table - ORG

Inconnu - DB2 SAMPLE

Table: CKMWONG.ORG

Date et heure en cours : 04/03/2002 5:25:03

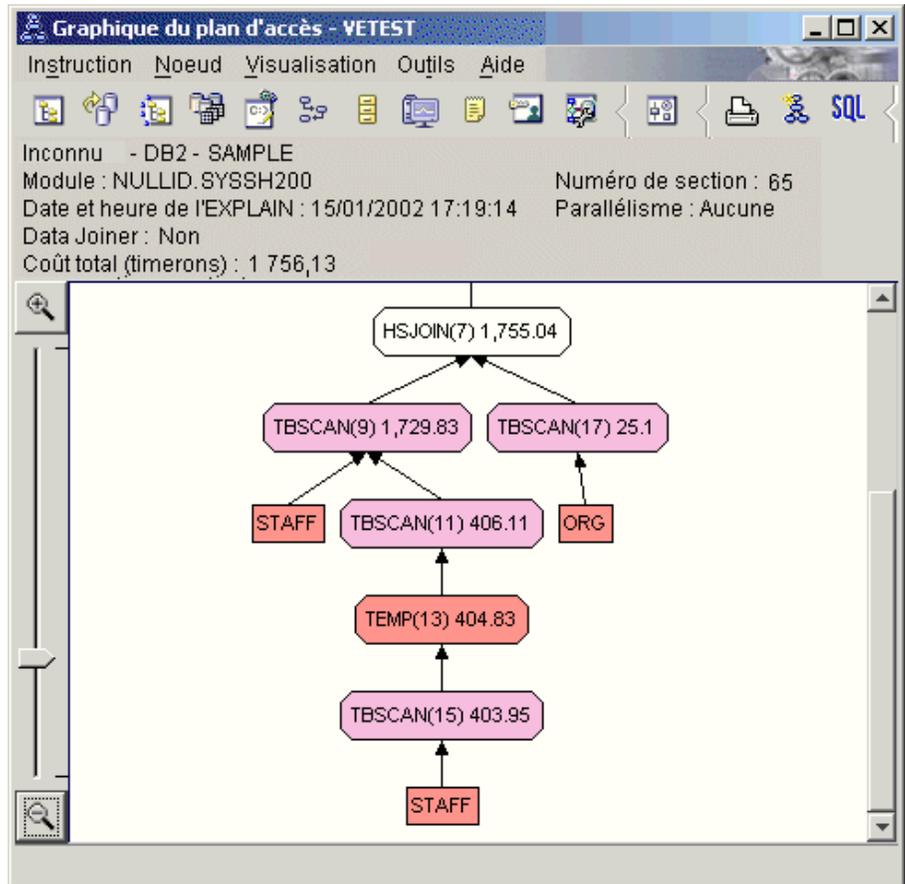
Date et heure de l'EXPLAIN : 04/03/2002 5:26:21

Statistiques	Valeurs explicitées	Valeurs en cours
CREATE_TIME	03/26/2002 1:35:42	03/26/2002 1:35:42
STATS_TIME	04/03/2002 5:24:55	04/03/2002 5:24:55
CARD	4	8
NPAGES	1	2
FPAGES	1	2
COLCOUNT	5	5
OVERFLOW	0	0
TABLESPACE	USERSPACE1	USERSPACE1
INDEX_TABLESPACE		
LONG_TABLESPACE		
VOLATILE	Non	Non

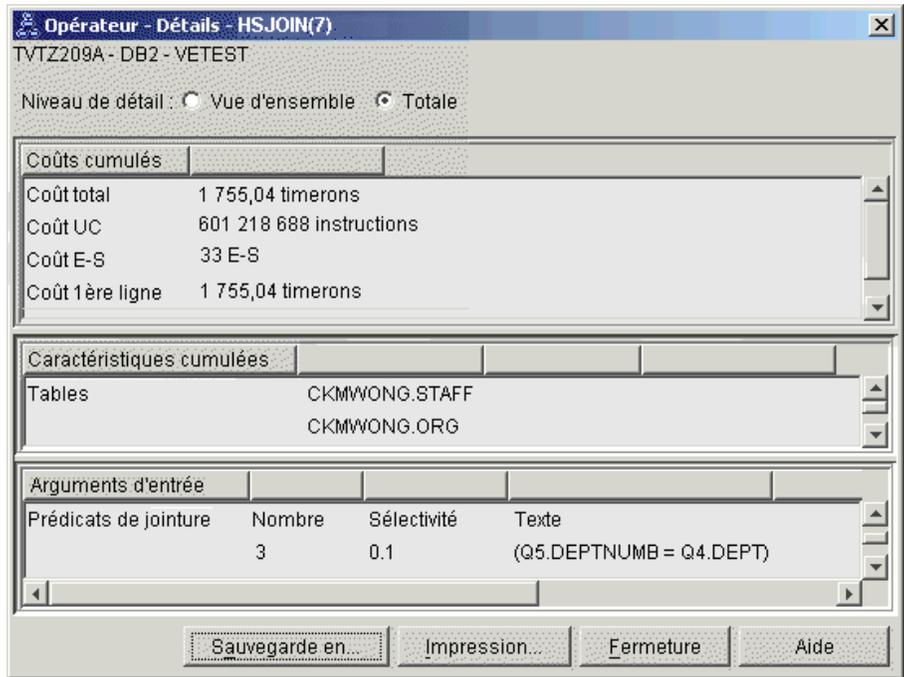
Colonnes référencées | Groupes de colonnes | Index | Sauvegarde en... | Impression... | Fermeture | Aide

2. Ce plan d'accès utilise-t-il les méthodes les plus efficaces pour l'accès aux données ?

Comme pour la Requête 1, le plan d'accès de la Requête 2 utilise les balayages de table (TBSCAN) et non des balayages d'index (IXSCAN). Même si des statistiques à jour existent, un balayage d'index ne peut pas être effectué parce qu'il n'y a aucun index sur les colonnes utilisées par la requête. Une façon d'améliorer la requête serait de fournir à l'optimiseur des index sur les colonnes utilisées pour joindre les tables (c'est-à-dire, sur les colonnes utilisées pour joindre les prédicats). Dans cet exemple, il s'agit de la première jointure de balayage : HSJOIN (7).



Dans la fenêtre Opérateur pour l'opérateur HSJOIN (7), regardez la section **Joindre les prédicats** sous **Arguments d'entrée**. Les colonnes utilisées pour cette opération de jointure sont répertoriées sous la colonne **Texte**. Dans cet exemple, ces colonnes sont DEPTNUMB et DEPT.



### 3. Ce plan d'accès est-il vraiment efficace ?

Les plans d'accès basés sur des statistiques à jour produisent toujours un coût approximatif réaliste (mesuré en timerons). Comme le coût approximatif de la Requête 1 a été basé sur des statistiques par défaut, il est impossible de comparer le coût de deux graphiques de plans d'accès pour déterminer lequel est le plus efficace. Que le coût soit plus élevé ou plus bas n'est pas significatif. Vous devez comparer le coût de plans d'accès basés sur des statistiques à jour pour obtenir une image réaliste de l'efficacité de chacun d'eux.

### 4. Suite du tutoriel

La Requête 3 répercute les effets de l'ajout d'index sur les colonnes DEPTNUMB et DEPT. L'ajout d'index sur les colonnes utilisées dans les prédicats de jointure peuvent améliorer les performances.

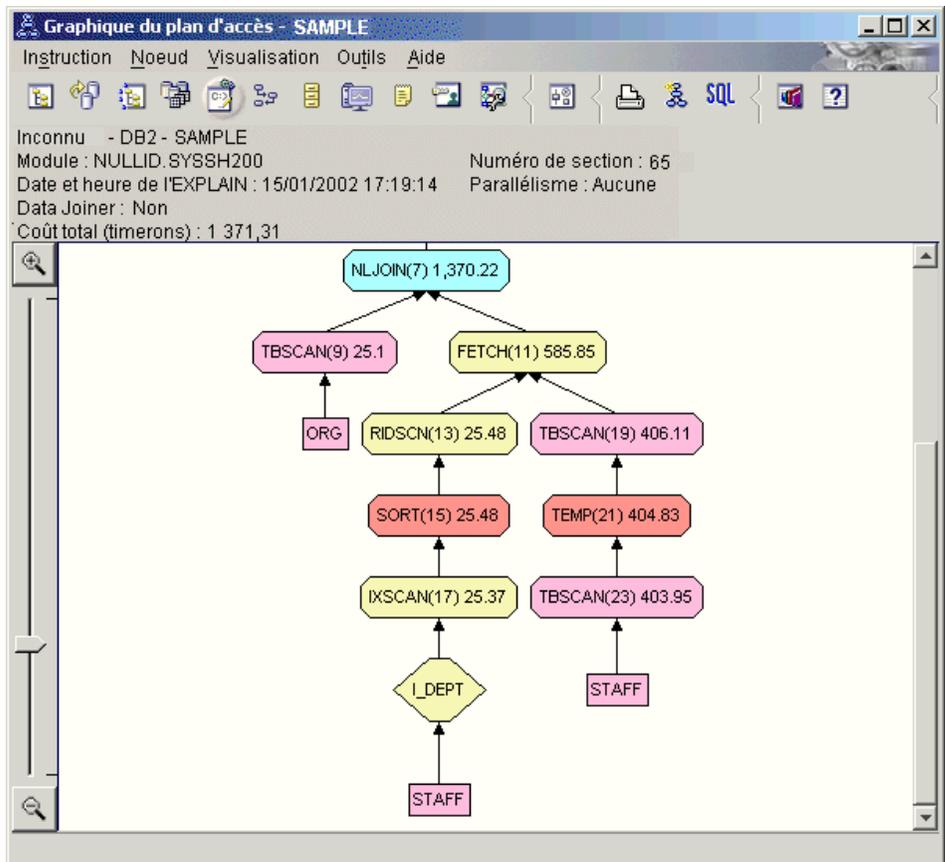
## Création d'index sur les colonnes utilisées pour joindre des tables dans une requête

Cet exemple s'appuie sur le plan d'accès décrit dans la Requête 2 pour la création d'index sur la colonne DEPT de la table STAFF et sur la colonne DEPTNUMB de la table ORG.

**Remarque :** Dans la version 8, des index recommandés peuvent être créés à l'aide de l'assistant d'amélioration des performances de la charge de travail.

Pour visualiser le graphique de plan d'accès pour cette requête (Requête 3), procédez comme suit : dans la fenêtre Historique des instructions explicitées, double-cliquez sur l'entrée appelée Requête 3. La fenêtre Graphique de plan d'accès pour l'exécution de l'instruction s'ouvre.

**Remarque :** Même si un index a été créé pour DEPTNUM, l'optimiseur n'a pas pu l'utiliser.

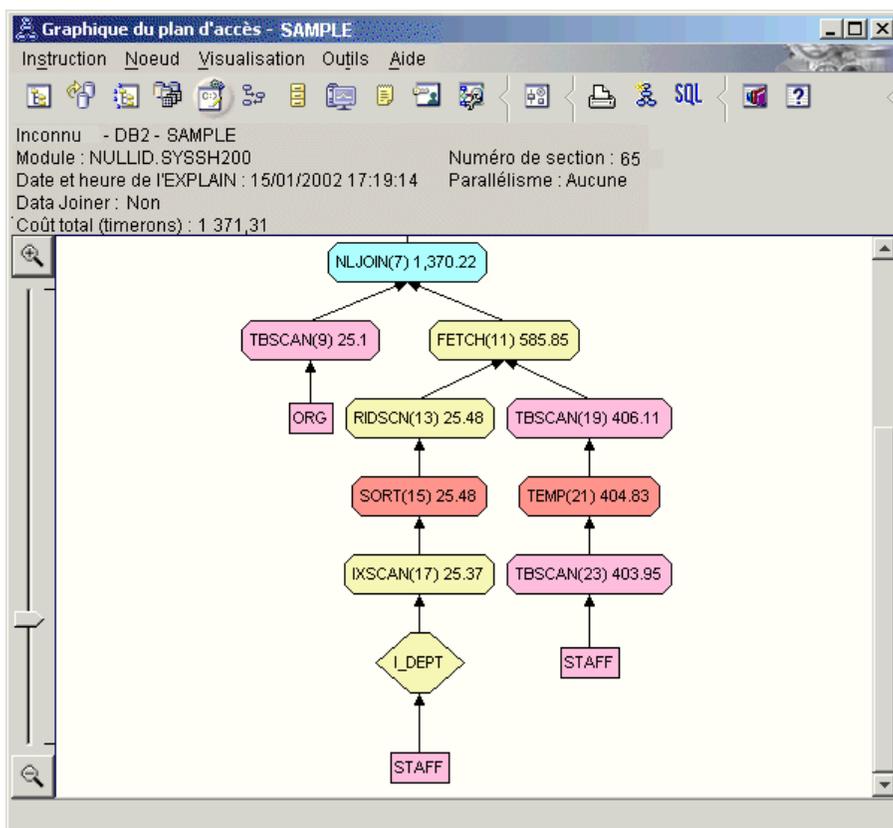


En répondant aux questions suivantes, vous comprendrez mieux comment améliorer une requête.

1. Qu'est-ce qui a changé dans le plan d'accès avec des index ?

Une jointure par boucle imbriquée, NLJOIN (7), a remplacé la jointure par hachage HSJOIN (7) utilisée dans la Requête 2. L'utilisation d'une jointure par boucle imbriquée a entraîné un coût approximatif inférieur à celui généré par une jointure par hachage parce que de type de jointure ne nécessite pas de table temporaire.

Un nouveau noeud (sous forme de losange), I\_DEPT, a été ajouté juste en dessous de la table STAFF. Ce noeud représente l'index créé sur DEPT. Il indique que l'optimiseur a utilisé un balayage d'index à la place d'un balayage de table pour déterminer les lignes à extraire.



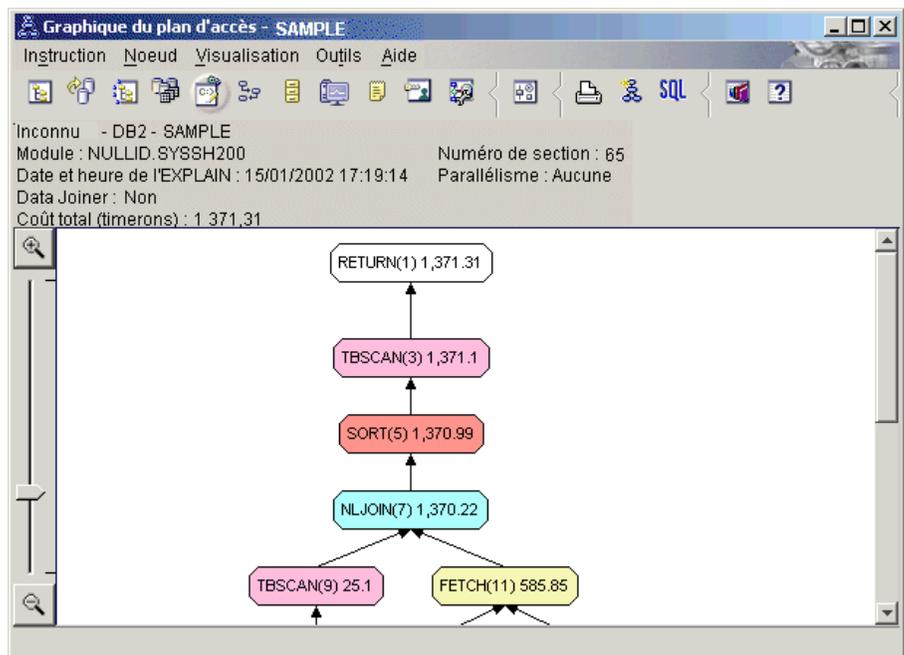
Dans cette section du graphique de plan d'accès, vous noterez qu'un nouvel index (I\_DEPT) a été créé sur la colonne DEPT et qu'IXSCAN (17) a été utilisé pour accéder à la table STAFF. Dans la Requête 2, un balayage de table a été utilisé pour accéder à la table STAFF.

2. Ce plan d'accès utilise-t-il les méthodes les plus efficaces pour l'accès aux données ?

L'ajout d'index a entraîné l'utilisation d'un noeud IXSCAN, IXSCAN (17), pour accéder à la table STAFF. La Requête 2 ne dispose pas d'index. Par conséquent, un balayage de table a été utilisé dans cet exemple.

Un noeud FETCH, FETCH (11), indique que, en plus de l'utilisation d'un balayage d'index pour extraire la colonne DEPT, l'optimiseur a extrait des colonnes supplémentaires à partir de la table STAFF, à l'aide de l'index en tant que pointeur. Dans ce cas, la combinaison d'un balayage et d'une extraction d'index est calculée pour être moins coûteuse que le balayage de table complet utilisé dans les plans d'accès antérieurs.

**Remarque :** Le noeud de la table STAFF apparaît deux fois, pour indiquer la relation avec l'index de DEPT et l'opération FETCH.



Le plan d'accès pour cette requête montre les effets de la création d'index sur les colonnes impliquées dans les prédicats de jointure. Les index peuvent également augmenter la vitesse des prédicats locaux. Analysez les prédicats locaux pour chaque table de cette requête pour voir comment l'ajout d'index aux colonnes référencées dans les prédicats locaux peuvent affecter le plan d'accès.

Dans la fenêtre Opérateur pour l'opérateur FETCH (11), regardez les colonnes situées sous **Propriétés cumulatives**. La colonne utilisée dans le prédicat pour cette opération d'extraction est JOB, comme indiqué dans la section Prédicats.

**Remarque :** La sélectivité de ce prédicat est de .69. Cela signifie qu'avec ce prédicat, 69 % des lignes seront sélectionnées pour un traitement ultérieur.

Opérateur - Détails - FETCH(11)  
Inconnu- DB2 - SAMPLE

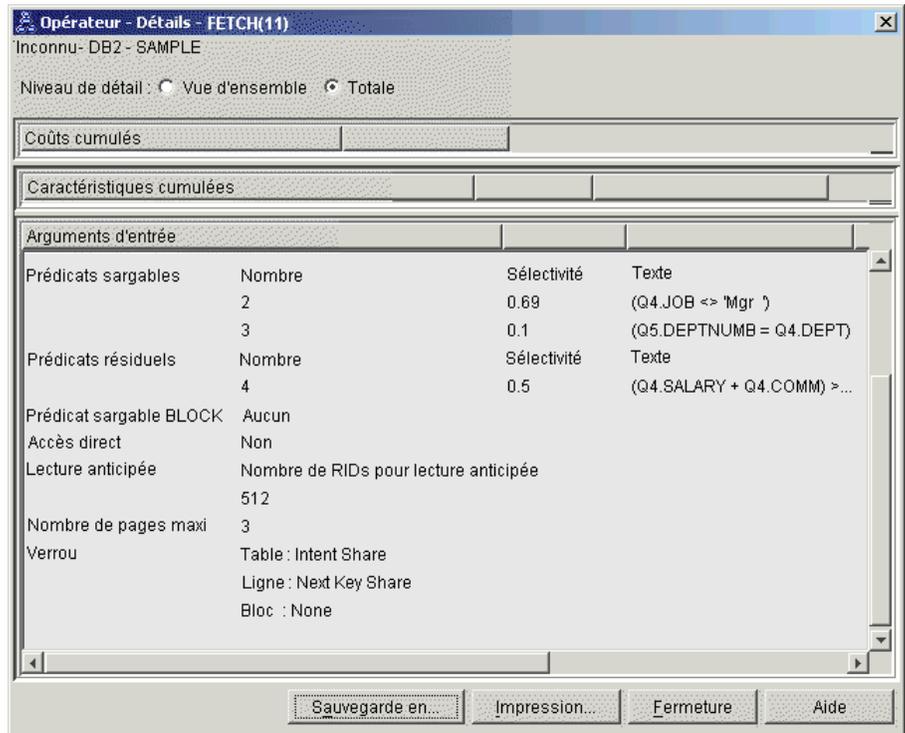
Niveau de détail :  Vue d'ensemble  Totale

Coûts cumulés	
Coût total	585,85 timerons
Coût UC	63 529 024 instructions
Coût E-S	32,97 E-S
Coût 1ère ligne	458,9 timerons

Caractéristiques cumulées			
Tables	CKMWONG.STAFF		
Colonnes	CKMWONG.STAFF.NAME CKMWONG.STAFF.ID CKMWONG.STAFF.COMM CKMWONG.STAFF.SALARY		
Ordre des colonnes	Aucun		
Prédicats	Nombre	Sélectivité	Texte
	2	0.69	(Q4.JOB <=> 'Mgr ')
	3	0.1	(Q5.DEPTNUMB = Q4.DEPT)
	4	0.5	(Q4.SALARY + Q4.COMM) > ...
Cardinalité	37.2		
Nb total de pages du tampon utilisées	23,39		

Sauvegarde en... Impression... Fermeture Aide



La fenêtre Opérateur pour l'opérateur FETCH (11) indique les colonnes utilisées pour cette opération. Vous pouvez voir que DEPTNAME est répertorié dans la première ligne à côté de **Colonnes récupérées** sous **Arguments d'entrée**.

### 3. Ce plan d'accès est-il vraiment efficace ?

Ce plan d'accès est plus efficace que celui de l'exemple précédent. Le coût cumulé a été réduit de 1 755 timerons dans la Requête 2 à environ 959 timerons dans la Requête 3.

Cependant, le plan d'accès pour la Requête 3 indique un balayage d'index IXSCAN (17) et une opération FETCH (11) pour la table STAFF. Si un balayage d'index combiné avec une opération d'extraction est moins coûteux qu'un balayage complet de table, il n'en reste pas moins que pour chaque ligne extraite, la table est accédée une fois et l'index aussi. Essayez de réduire ce double accès à la table STAFF.

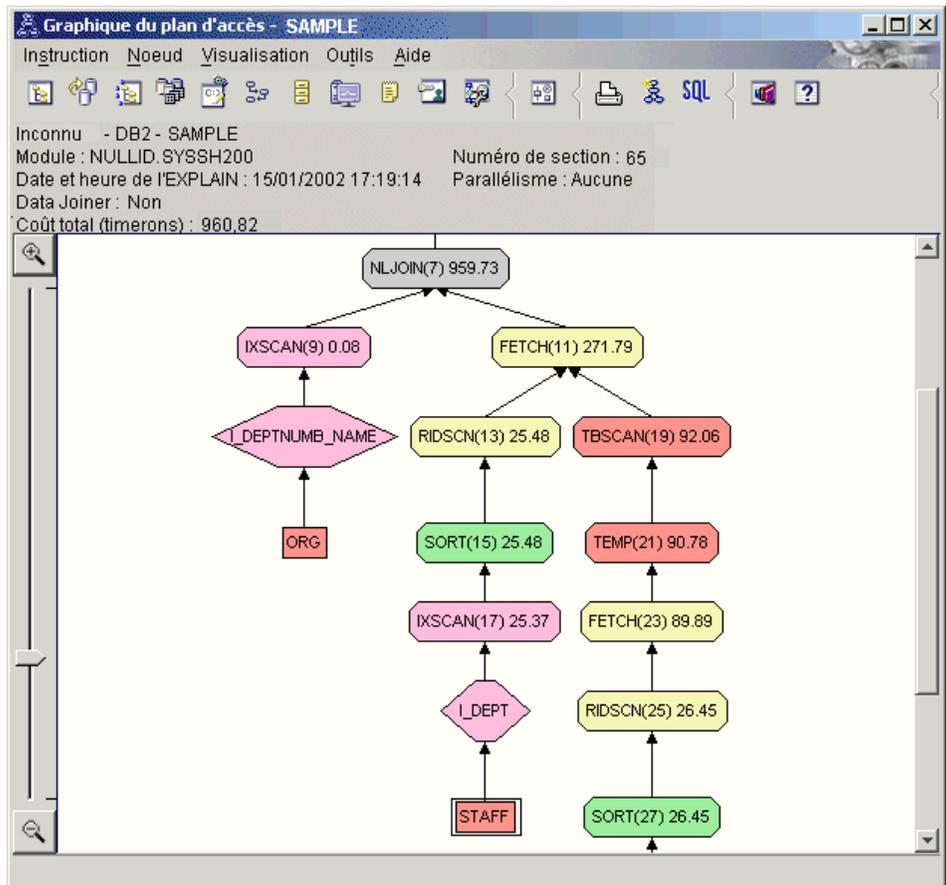
### 4. Suite du tutoriel

La Requête 4 réduit le balayage d'index et l'opération d'extraction à un seul balayage d'index sans opération d'extraction. La création d'index supplémentaires peut réduire le coût d'un plan d'accès.

## Création d'index supplémentaires sur les colonnes de table

Cet exemple s'appuie sur le plan d'accès décrit dans la Requête 3 en créant un index sur la colonne JOB dans la table STAFF et en ajoutant DEPTNAME à l'index existant de la table ORG. L'ajout d'un index distinct pourrait entraîner un accès supplémentaire.

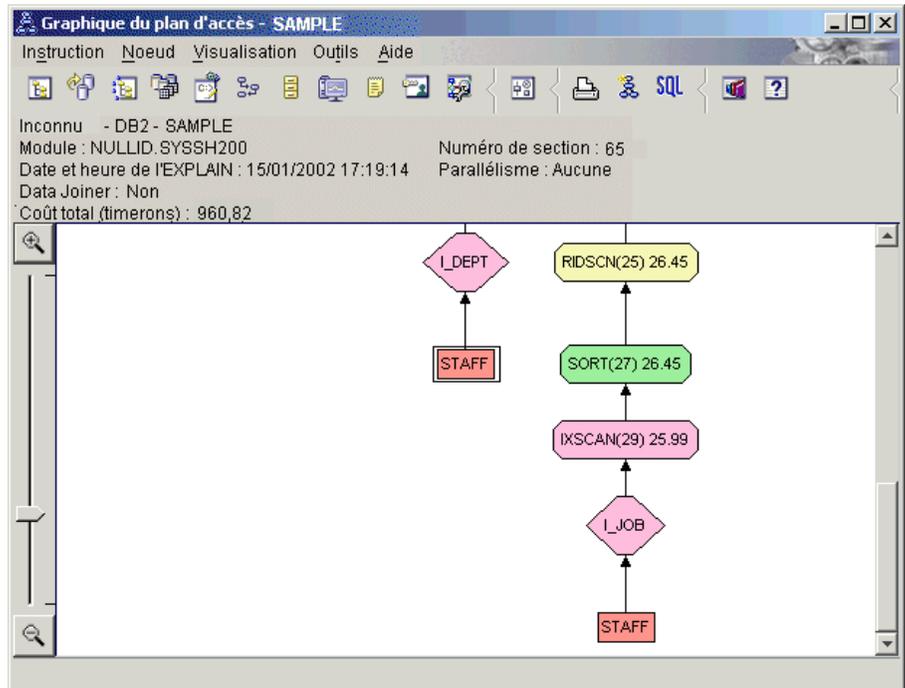
Pour visualiser le graphique de plan d'accès pour cette requête (Requête 4), procédez comme suit : dans la fenêtre Historique des instructions explicitées, double-cliquez sur l'entrée appelée Requête 4. La fenêtre Graphique de plan d'accès pour l'exécution de l'instruction s'ouvre.



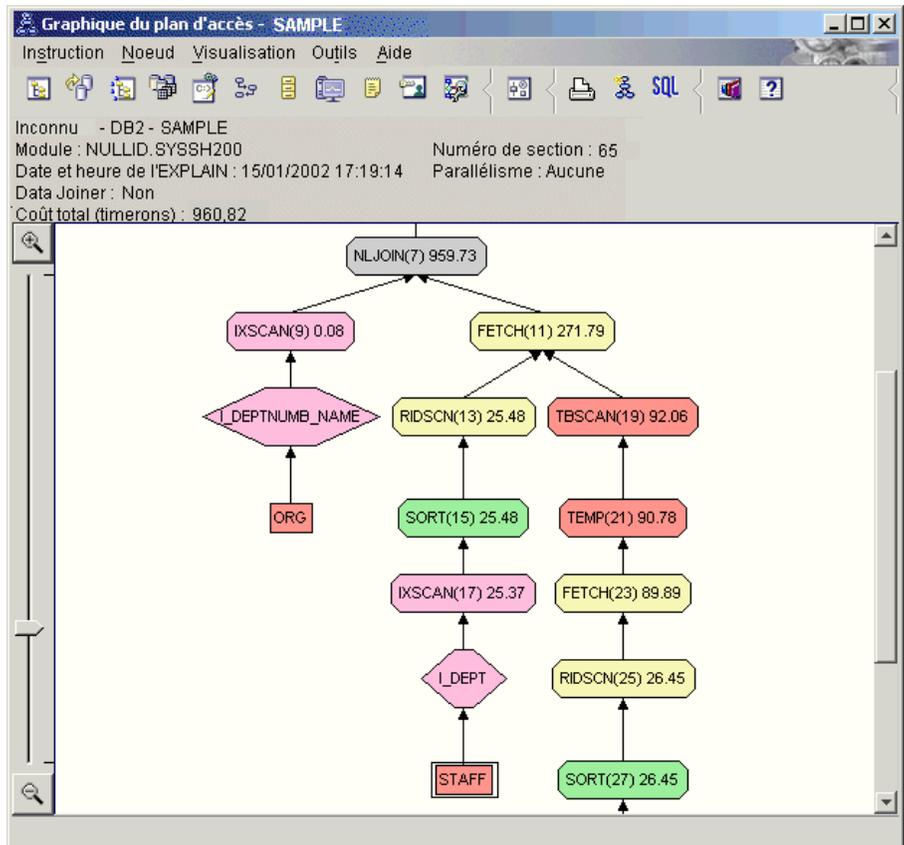
En répondant aux questions suivantes, vous comprendrez mieux comment améliorer une requête.

1. Qu'est-ce qui a changé dans ce plan d'accès lorsque vous avez créé des index supplémentaires ?

L'optimiseur a tiré profit de l'index créé sur la colonne JOB dans la table STAFF (représenté par un losange appelée I\_JOB) pour affiner ce plan d'accès.



Au milieu du graphique de plan d'accès, vous remarquerez que, pour la table ORG, le balayage et l'extraction d'index ont été modifiés par un balayage d'index unique : IXSCAN (9). L'ajout de la colonne DEPTNAME à l'index de la table ORG a permis à l'optimiseur d'éliminer l'accès supplémentaire impliquant l'opération d'extraction.



2. Ce plan d'accès est-il vraiment efficace ?

Ce plan d'accès est plus efficace que celui de l'exemple précédent. Le coût cumulatif a été réduit de 1370 timerons dans la Requête 3 à environ 959 timerons dans la Requête 4.

## Informations connexes

Reportez-vous au manuel *Administration Guide* pour obtenir des informations détaillées sur les étapes supplémentaires à suivre pour l'amélioration des performances. Vous pouvez alors revenir à Visual Explain pour évaluer l'impact de vos actions.



---

## Leçon 4. Amélioration d'un plan d'accès dans un environnement de base de données partitionnée

Dans cette leçon, vous apprendrez comment un plan d'accès et les fenêtres associées à la requête de base sont modifiés lorsque vous optimisez les performances. A l'aide d'une série d'exemples et d'illustrations, vous apprendrez comment le coût total estimé du plan d'accès, même pour une simple requête, peut être amélioré en utilisant la commande **runstats** et en ajoutant des index.

Au fur et à mesure que vous vous familiariserez avec Visual Explain, vous découvrirez d'autres façons d'optimiser les requêtes.

---

### Travail avec des graphiques de plans d'accès

A l'aide de quatre images instantanées EXPLAIN exemples, vous apprendrez comment l'optimisation joue un rôle essentiel quant aux performances de la base de données.

Les requêtes associées aux images instantanées EXPLAIN sont numérotées de 1 à 4. Chaque requête utilise la même instruction SQL (décrite dans la Leçon 1) :

```
SELECT S.ID,S.NAME,O.DEPTNAME,SALARY+COMM
FROM   ORG O, STAFF S
WHERE
  O.DEPTNUMB = S.DEPT AND
  S.JOB <> 'Mgr' AND
  S.SALARY+S.COMM > ALL( SELECT ST.SALARY*.9
                        FROM STAFF ST
                        WHERE ST.JOB='Mgr' )
ORDER BY S.NAME
```

Chaque itération de la requête utilise davantage de techniques d'optimisation que la dernière exécution. Par exemple, La requête 1 n'a pas été optimisée, alors que la requête 4 a été optimisée au maximum. Les différences entre les requêtes apparaissent ci-dessous :

#### Requête 1

Exécution d'une requête sans index ni statistiques

#### Requête 2

Collecte des statistiques à jour pour les tables et les index dans une requête

### Requête 3

Création d'index sur les colonnes utilisées pour joindre des tables dans une requête

### Requête 4

Création d'index supplémentaires sur les colonnes de table

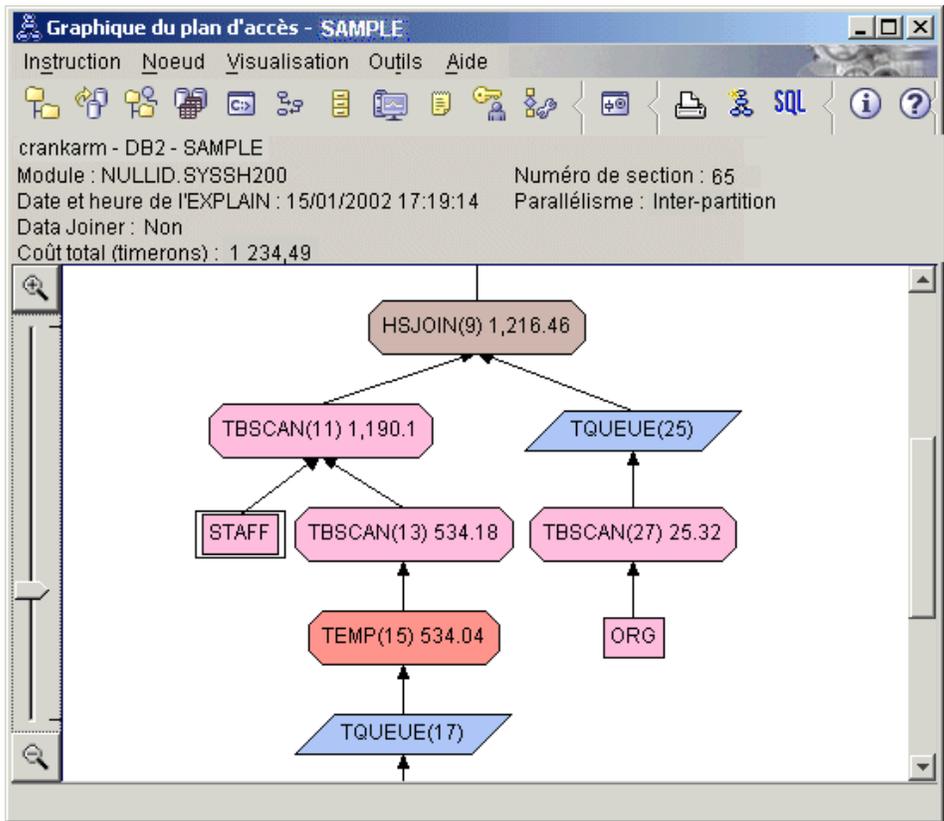
Ces exemples ont été réalisés sur une machine RS/6000 SP avec 7 noeuds physiques utilisant un parallélisme inter-partitions.

## Exécution d'une requête sans index ni statistiques

Dans cet exemple, le plan d'accès a été créé pour la requête SQL sans index ni statistiques.

Pour visualiser le graphique de plan d'accès pour cette requête (Requête 1), procédez comme suit :

1. Dans le Centre de contrôle, développez l'arborescence jusqu'à ce que vous trouviez la base de données exemple.
2. Cliquez avec le bouton droit de votre souris sur la base de données et sélectionnez **Affichage de l'historique des instructions explicitées** dans le menu qui s'affiche. La fenêtre Historique des instructions explicitées s'ouvre.
3. Double-cliquez sur l'entrée appelée Numéro de requête 1 (il se peut que vous ayez à faire défiler la fenêtre vers la droite pour voir la colonne **Numéro de requête**). La fenêtre Graphique du plan d'accès pour l'instruction s'ouvre.



En répondant aux questions suivantes, vous comprendrez mieux comment améliorer une requête.

1. Les statistiques à jour existent-elles pour chaque table de la requête ?

Pour vérifier si des statistiques à jour existent pour chaque table de la requête, double-cliquez sur chaque noeud de table dans le graphique du plan d'accès. Dans la fenêtre Statistiques sur table qui s'ouvre, la ligne **STATS\_TIME** sous la colonne **Explicitée** contient la phrase "Statistiques non mises à jour" si aucune statistique n'a été collectée au moment où l'image instantanée a été créée.

Si des statistiques à jour n'existent pas, l'optimiseur utilise les statistiques par défaut, qui peuvent différer des vraies statistiques. Les statistiques par défaut sont identifiées par les mots "par défaut" en dessous de la colonne **Explicitée** de la fenêtre Statistiques sur table.

En fonction des informations de la fenêtre Statistiques sur table pour la table ORG, l'optimiseur a utilisé les statistiques par défaut (comme indiqué à côté des valeurs explicitées). Les statistiques par défaut ont été

utilisées parce que les statistiques à jour n'étaient pas disponibles au moment où l'image instantanée a été créée (comme indiqué à la ligne `STATS_TIME`).

Statistiques	Valeurs explicitées	Valeurs en cours
CREATE_TIME	03/26/2002 1:35:42	03/26/2002 1:35:42
STATS_TIME	04/03/2002 5:24:55	04/03/2002 5:24:55
CARD	4	8
NPAGES	1	2
FPAGES	1	2
COLCOUNT	5	5
OVERFLOW	0	0
TABLESPACE	USERSPACE1	USERSPACE1
INDEX_TABLESPACE		
LONG_TABLESPACE		
VOLATILE	Non	Non

2. Ce plan d'accès utilise-t-il les méthodes les plus efficaces pour l'accès aux données ?

Ce plan d'accès contient des balayages de tables, et non des balayages d'index. Les balayages de table sont indiqués par des octogones appelés TBSCAN. Si des balayages d'index avaient été utilisés, ils seraient apparus sous la forme de losanges appelés IXSCAN. L'utilisation d'un index créé pour une table est plus efficace d'un point de vue qualité/prix qu'un balayage de table si de petits volumes de données sont extraits.

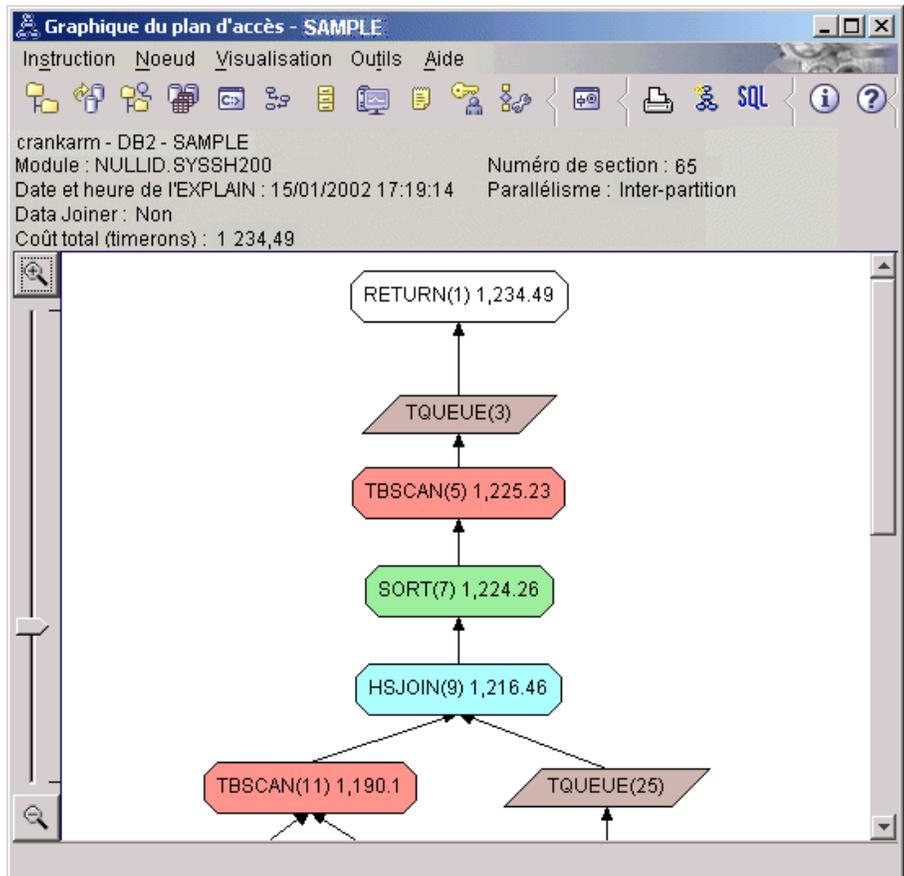
3. Ce plan d'accès est-il vraiment efficace ?

Vous ne pouvez connaître l'efficacité d'un plan d'accès que s'il est basé sur des statistiques à jour. Comme l'optimiseur a utilisé les statistiques par défaut dans le plan d'accès, vous ne pouvez pas déterminer l'efficacité du plan.

En général, vous devez noter le coût total approximatif pour le plan d'accès pour une comparaison ultérieure avec les plans d'accès révisés. Le coût répertorié dans chaque noeud est cumulatif, des premières étapes de votre requête jusqu'au noeud inclus.

**Remarque :** Pour les bases de données partitionnées, il s'agit du coût cumulatif pour le noeud qui utilise le plus de ressources.

Dans la fenêtre Graphique du plan d'accès, le coût total est d'environ 1234 timerons (indiqué dans **RETOUR (1)** en haut du graphique). Le coût total estimé est également indiqué dans la zone supérieure de la fenêtre.



#### 4. Suite du tutoriel

La Requête 2 ressemble à un plan d'accès pour la requête de base une fois que la commande **runstats** a été lancée. La commande **runstats** permet de fournir à l'optimiseur les statistiques à jour sur les tables accédées par la requête.

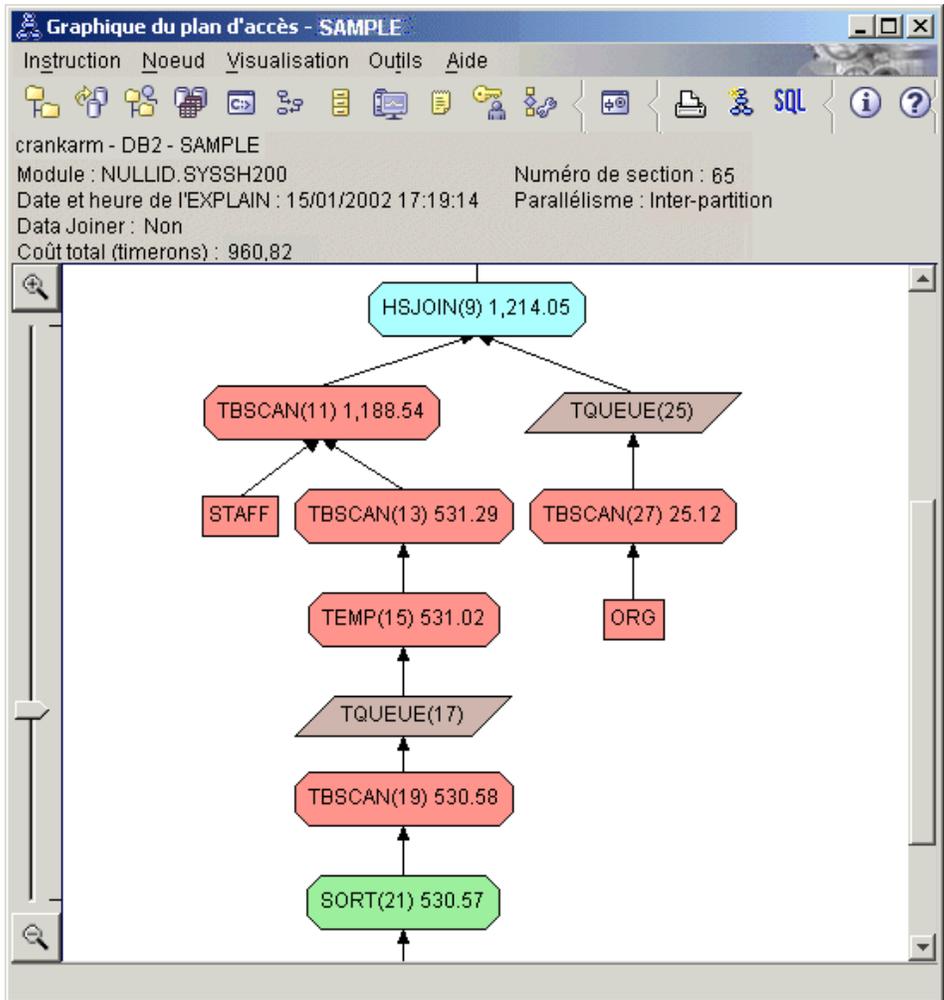
## Collecte des statistiques à jour pour les tables et les index à l'aide de la commande runstats

Cet exemple s'appuie sur le plan d'accès décrit dans la Requête 1 en réunissant des statistiques à jour à l'aide de la commande **runstats**.

Il est vivement recommandé de relancer la commande **RUNSTATS** pour collecter des statistiques à jour sur les tables et les index, notamment si des mises à jour importantes ont été effectuées ou si des index ont été créés depuis la dernière exécution de cette commande. L'optimiseur dispose ainsi des informations les plus récentes lui permettant de définir le meilleur plan d'accès. Si les statistiques disponibles dans le catalogue système ne sont pas à jour, l'optimiseur utilise des valeurs par défaut et risque d'établir un plan d'accès peu performant.

Lorsque vous modifiez une table, veillez à utiliser la commande **RUNSTATS** après cette mise à jour ; sinon, l'optimiseur risque de considérer que la table est vide. Pour vérifier que la commande **RUNSTATS** a bien été lancée, consultez la valeur de cardinalité affichée dans la fenêtre Détails opérateur. Si elle est égale à zéro, terminez les mises à jour de la table, exécutez de nouveau **RUNSTATS**, et recréez les images instantanées **EXPLAIN** correspondant aux tables concernées.

Pour visualiser le graphique de plan d'accès pour cette requête (Requête 2), procédez comme suit : dans la fenêtre Historique des instructions explicitées, double-cliquez sur l'entrée appelée Requête 2. La fenêtre Graphique de plan d'accès pour l'exécution de l'instruction s'ouvre.



En répondant aux questions suivantes, vous comprendrez mieux comment améliorer une requête.

1. Les statistiques à jour existent-elles pour chaque table de la requête ?

La fenêtre Statistiques sur table pour la table ORG indique que l'optimiseur a utilisé les statistiques à jour (la valeur **STATS\_TIME** correspond à l'heure réelle de collecte des statistiques). La précision des statistiques dépend de l'impact des modifications sur le contenu des tables, une fois la commande **runstats** exécutée.

Statistiques sur la table - ORG

crankarm - DB2 - SAMPLE

Table: DB2ADMIN.ORG

Date et heure en cours : 04/03/2002 5:25:03

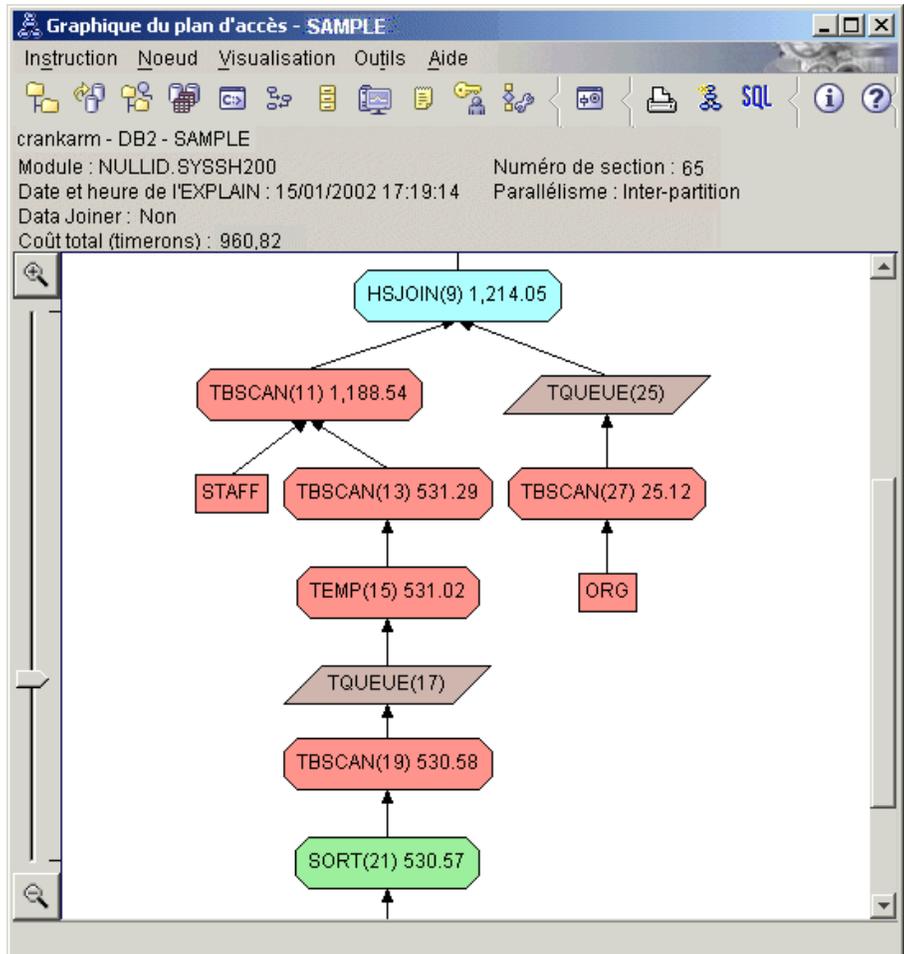
Date et heure de l'EXPLAIN : 04/03/2002 5:26:21

Statistiques	Valeurs explicitées	Valeurs en cours
CREATE_TIME	03/26/2002 1:35:42	03/26/2002 1:35:42
STATS_TIME	04/03/2002 5:24:55	04/03/2002 5:24:55
CARD	4	8
NPAGES	1	2
FPAGES	1	2
COLCOUNT	5	5
OVERFLOW	0	0
TABLESPACE	USERSPACE1	USERSPACE1
INDEX_TABLESPACE		
LONG_TABLESPACE		
VOLATILE	Non	Non

Colonnes référencées | Groupes de colonnes | Index | Sauvegarde en ... | Impression... | Fermeture | Aide

2. Ce plan d'accès utilise-t-il les méthodes les plus efficaces pour l'accès aux données ?

Comme pour la Requête 1, le plan d'accès de la Requête 2 utilise les balayages de table (TBSCAN) et non des balayages d'index (IXSCAN). Même si des statistiques à jour existent, un balayage d'index ne peut pas être effectué parce qu'il n'y a aucun index sur les colonnes utilisées par la requête. Une façon d'améliorer la requête serait de fournir à l'optimiseur des index sur les colonnes utilisées pour joindre les tables (c'est-à-dire, sur les colonnes utilisées pour joindre les prédicats). Dans cet exemple, il s'agit de la première jointure de balayage : HSJOIN (9).



Dans la fenêtre Opérateur pour l'opérateur HSJOIN (9), regardez la section **Joindre les prédicats** sous **Arguments d'entrée**. Les colonnes utilisées pour cette opération de jointure sont répertoriées sous la colonne **Texte**. Dans cet exemple, ces colonnes sont DEPTNUMB et DEPT.

Opérateur - Détails - HSJOIN(9)			
crankarm - DB2 - SAMPLE			
Niveau de détail : <input type="radio"/> Vue d'ensemble <input checked="" type="radio"/> Totale			
Coûts cumulés			
Coût total	1 214,05	timerons	
Coût UC	62 745 724	instructions	
Coût E-S	43	E-S	
Coût 1ère ligne	4.3	timerons	
Coût de communication	4.3	timerons	
Coût 1ère communication	4.3	timerons	
Caractéristiques cumulées			
Tables	DB2ADMIN.STAFF		
	DB2ADMIN.ORG		
Arguments d'entrée			
Prédicats de jointure	Nombre	Sélectivité	Texte
	3	0.1	(Q5.DEPTNUMB = Q4.DEPT)
<input type="button" value="Sauvegarde en..."/> <input type="button" value="Impression..."/> <input type="button" value="Fermeture"/> <input type="button" value="Aide"/>			

### 3. Ce plan d'accès est-il vraiment efficace ?

Les plans d'accès basés sur des statistiques à jour produisent toujours un coût approximatif réaliste (mesuré en timerons). Comme le coût approximatif de la Requête 1 a été basé sur des statistiques par défaut, il est impossible de comparer le coût de deux graphiques de plans d'accès pour déterminer lequel est le plus efficace. Que le coût soit plus élevé ou plus bas n'est pas significatif. Vous devez comparer le coût de plans d'accès basés sur des statistiques à jour pour obtenir une image réaliste de l'efficacité de chacun d'eux.

### 4. Suite du tutoriel

La Requête 3 répercute les effets de l'ajout d'index sur les colonnes DEPTNUMB et DEPT. L'ajout d'index sur les colonnes utilisées dans les prédicats de jointure peuvent améliorer les performances.

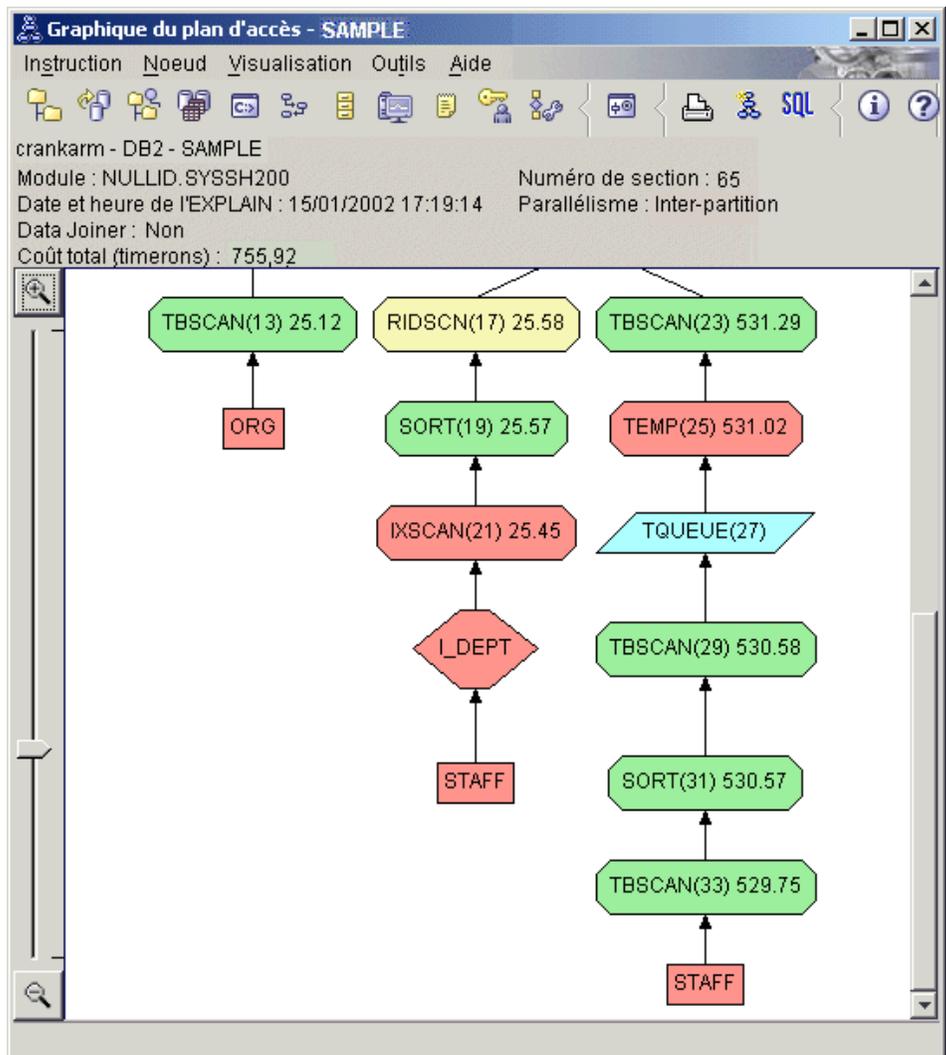
## Création d'index sur les colonnes utilisées pour joindre des tables dans une requête

Cet exemple s'appuie sur le plan d'accès décrit dans la Requête 2 pour la création d'index sur la colonne DEPT de la table STAFF et sur la colonne DEPTNUMB de la table ORG.

**Remarque :** Dans la version 8, des index recommandés peuvent être créés à l'aide de l'assistant d'amélioration des performances de la charge de travail.

Pour visualiser le graphique de plan d'accès pour cette requête (Requête 3), procédez comme suit : dans la fenêtre Historique des instructions explicitées, double-cliquez sur l'entrée appelée Requête 3. La fenêtre Graphique de plan d'accès pour l'exécution de l'instruction s'ouvre.

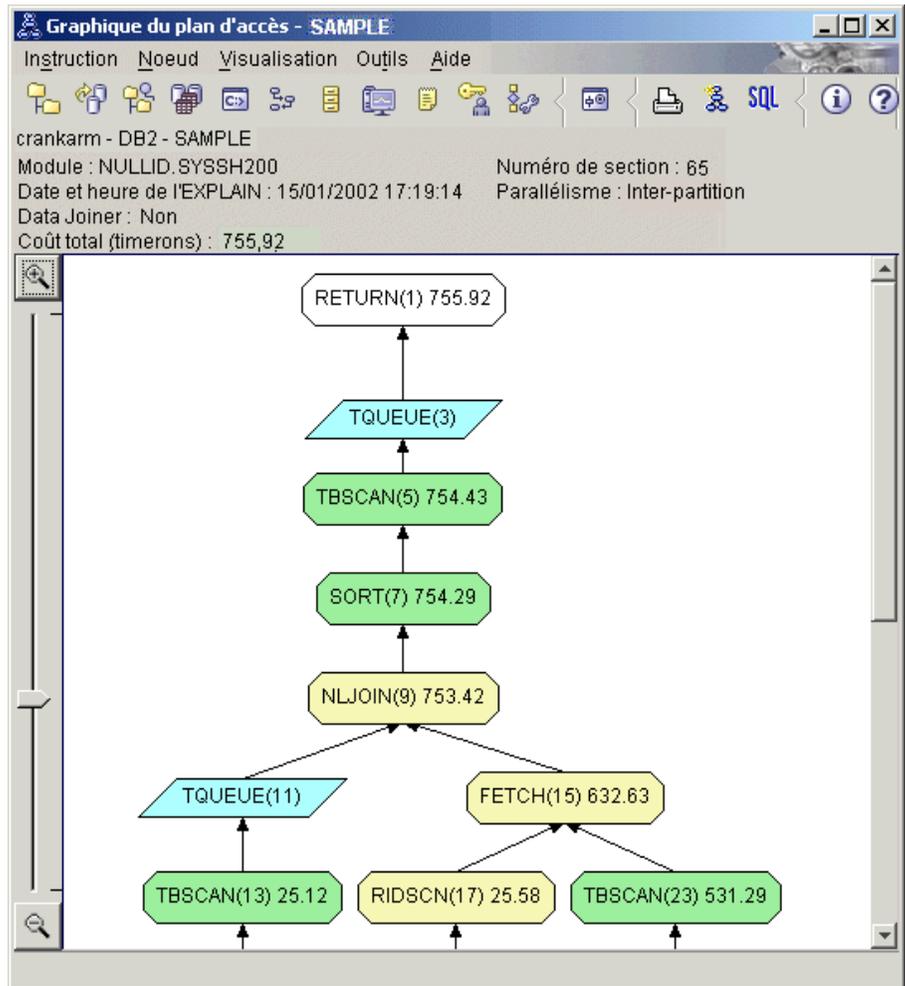
**Remarque :** Même si un index a été créé pour DEPTNUM, l'optimiseur n'a pas pu l'utiliser.



En répondant aux questions suivantes, vous comprendrez mieux comment améliorer une requête.

1. Qu'est-ce qui a changé dans le plan d'accès avec des index ?

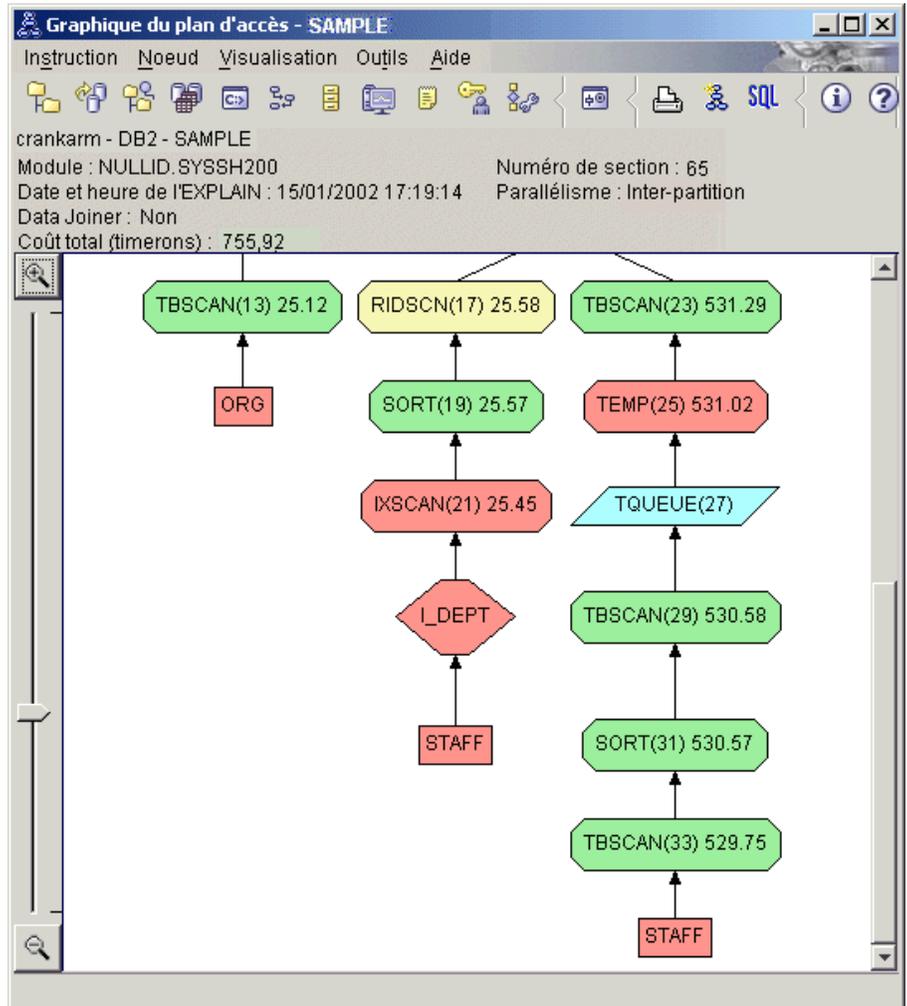
Un nouveau noeud (sous forme de losange), **I\_DEPT**, a été ajouté juste en dessous de la table STAFF. Ce noeud représente l'index créé sur DEPT. Il indique que l'optimiseur a utilisé un balayage d'index à la place d'un balayage de table pour déterminer les lignes à extraire.



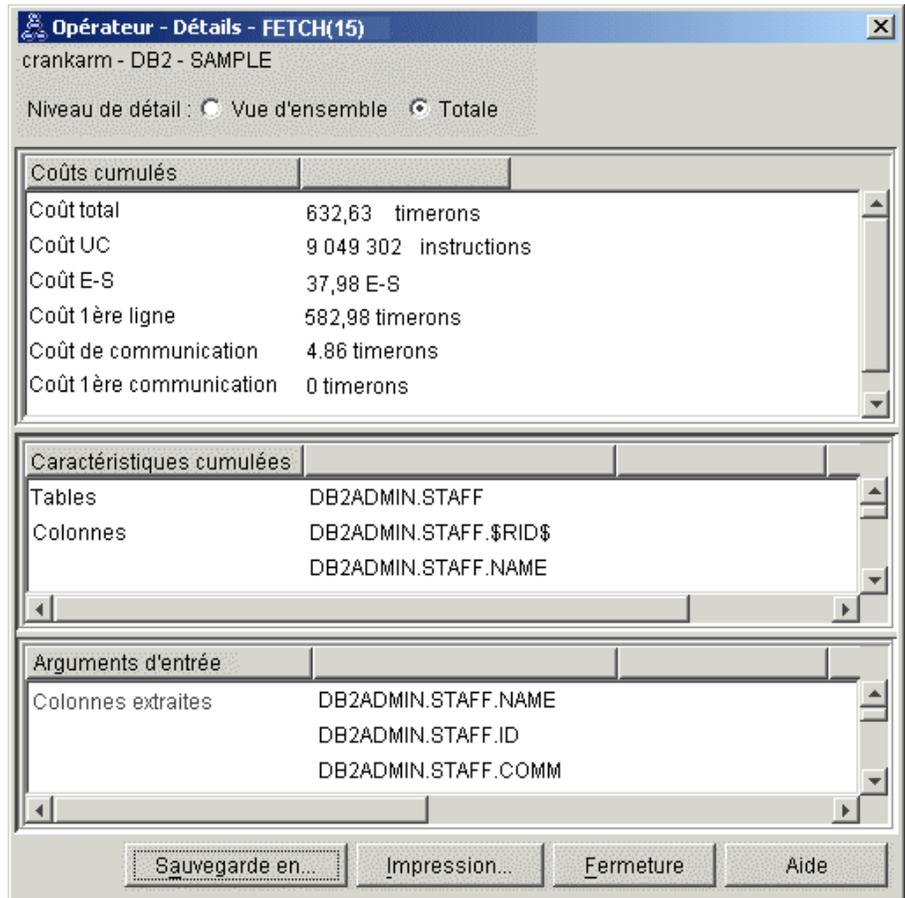
2. Ce plan d'accès utilise-t-il les méthodes les plus efficaces pour l'accès aux données ?

Le plan d'accès pour cette requête montre l'effet de la création d'index sur la colonne DEPTNUMB de la table ORG, résultant d'une opération FETCH (15) et d'une opération IXSCAN (21) et sur la colonne DEPT de la table

STAFF. La Requête 2 ne dispose pas d'index. Par conséquent, un balayage de table a été utilisé dans cet exemple.



La fenêtre Opérateur pour l'opérateur FETCH (15) indique les colonnes utilisées pour cette opération.



La combinaison d'un balayage et d'une extraction d'index est calculée pour être moins coûteuse que le balayage de table complet utilisé dans les plans d'accès antérieurs.

3. Ce plan d'accès est-il vraiment efficace ?

Ce plan d'accès est plus efficace que celui de l'exemple précédent. Le coût cumulatif a été réduit de 1214 timerons dans la Requête 2 à environ 755 timerons dans la Requête 3.

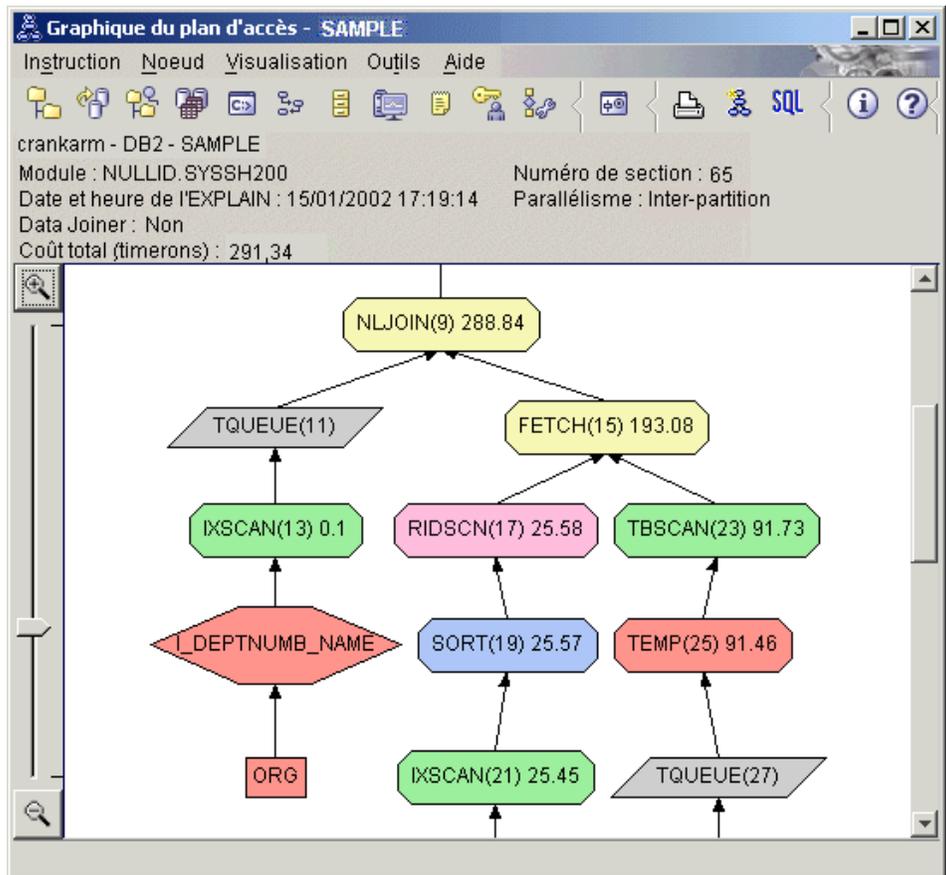
4. Suite du tutoriel

La Requête 4 réduit le balayage d'index et l'opération d'extraction à un seul balayage d'index sans opération d'extraction. La création d'index supplémentaires peut réduire le coût d'un plan d'accès.

## Création d'index supplémentaires sur les colonnes de table

Cet exemple s'appuie sur le plan d'accès décrit dans la Requête 3 en créant un index sur la colonne JOB dans la table STAFF et en ajoutant DEPTNAME à l'index existant de la table ORG. L'ajout d'un index distinct pourrait entraîner un accès supplémentaire.

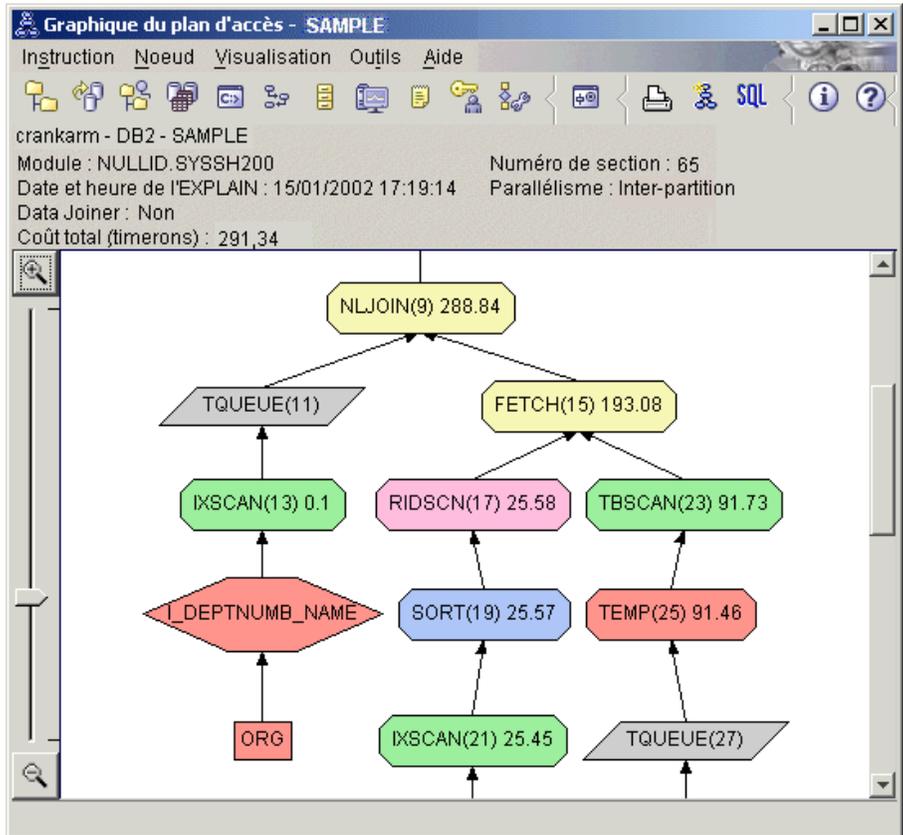
Pour visualiser le graphique de plan d'accès pour cette requête (Requête 4), procédez comme suit : dans la fenêtre Historique des instructions explicitées, double-cliquez sur l'entrée appelée Requête 4. La fenêtre Graphique de plan d'accès pour l'exécution de l'instruction s'ouvre.



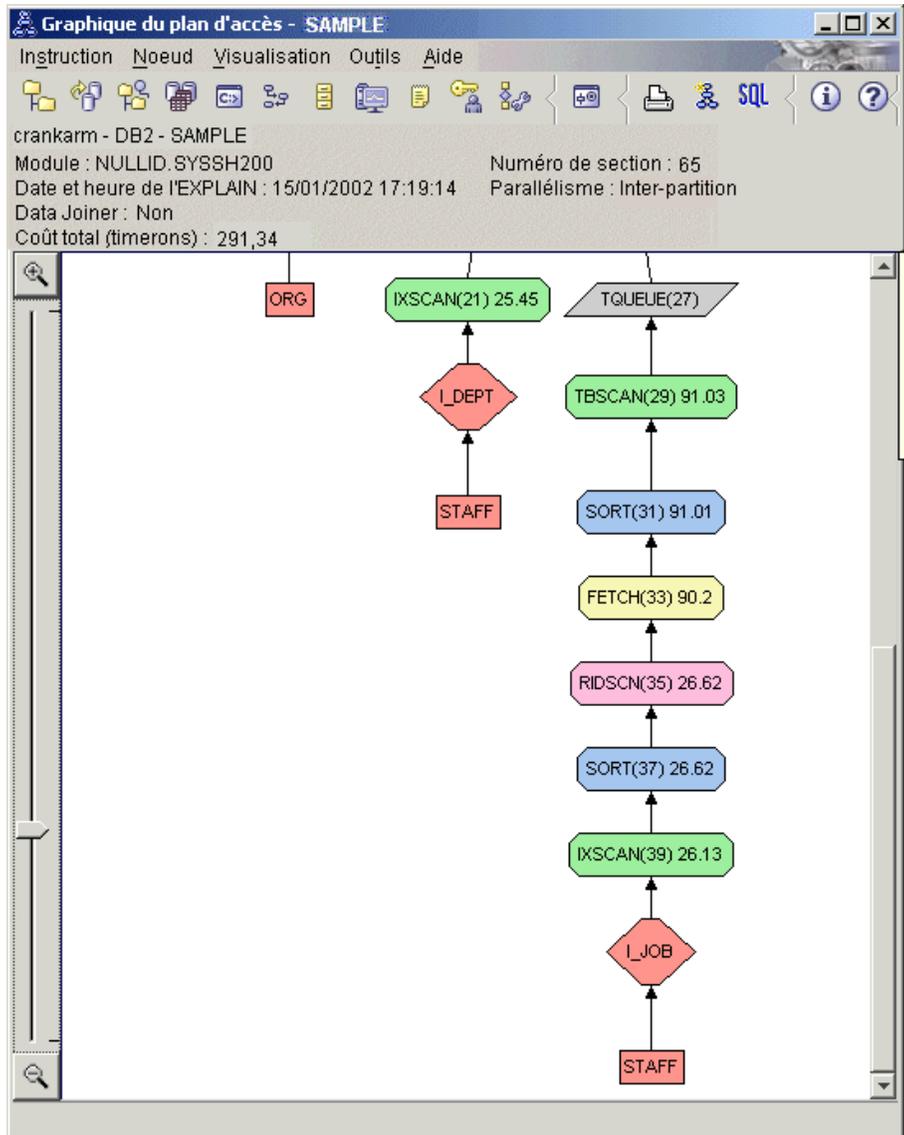
En répondant aux questions suivantes, vous comprendrez mieux comment améliorer une requête.

1. Qu'est-ce qui a changé dans ce plan d'accès lorsque vous avez créé des index supplémentaires ?

Au milieu du graphique de plan d'accès, vous remarquerez que, pour la table ORG, le balayage de table a été modifié par un balayage d'index : IXSCAN (7). L'ajout de la colonne DEPTNAME à l'index de la table ORG a permis à l'optimiseur d'affiner l'accès impliquant le balayage de table.



En bas du graphique de plan d'accès, vous remarquerez que, pour la table STAFF, le balayage et l'extraction d'index ont été modifiés par un balayage d'index unique : IXSCAN (39). La création de l'index JOB à la table STAFF a permis à l'optimiseur d'éliminer l'accès supplémentaire impliquant l'opération d'extraction.



2. Ce plan d'accès est-il vraiment efficace ?

Ce plan d'accès est plus efficace que celui de l'exemple précédent. Le coût cumulatif a été réduit de 753 timerons dans la Requête 3 à environ 288 timerons dans la Requête 4.

---

## Informations connexes

Reportez-vous au manuel *Administration Guide* pour obtenir des informations détaillées sur les étapes supplémentaires à suivre pour l'amélioration des performances. Vous pouvez alors revenir à Visual Explain pour évaluer l'impact de vos actions.

---

## Annexe A. Concepts Visual Explain

---

### Plan d'accès

Certaines données sont nécessaires pour résoudre une instruction SQL explicitable. Un *plan d'accès* définit l'ordre des opérations à effectuer pour accéder à ces données. Vous pouvez afficher des statistiques sur les tables, les index ou les colonnes sélectionnés, prendre connaissance des caractéristiques des opérateurs, consulter des informations générales telles que des statistiques sur un espace table et une fonction et afficher les paramètres de configuration permettant l'optimisation. Visual Explain vous permet d'afficher le plan d'accès associé à une instruction SQL sous la forme d'un graphique.

L'optimiseur génère un plan d'accès chaque fois qu'une instruction SQL explicitable est compilée. Ceci se produit au moment de la préparation ou de la définition des accès dans le cas d'instructions statiques et au moment de l'exécution dans le cas d'instructions dynamiques.

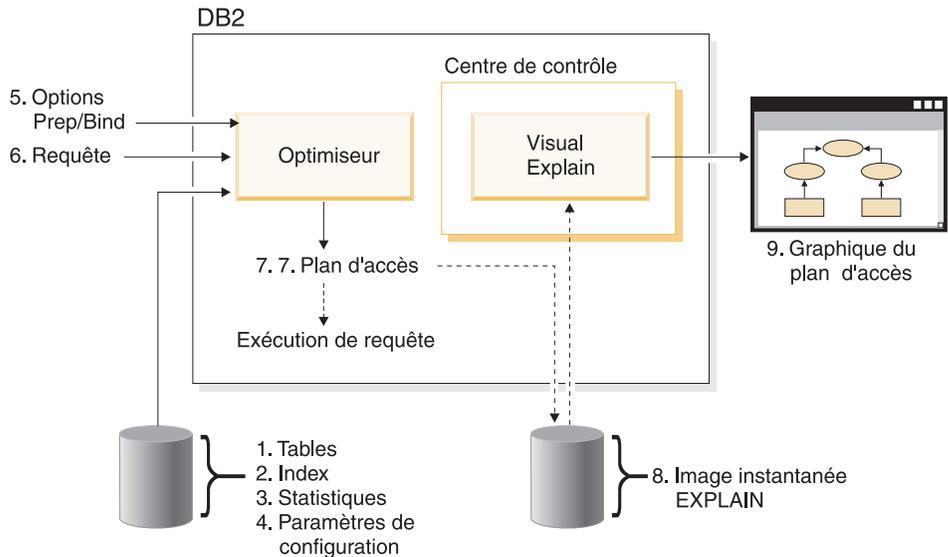
Important : un plan d'accès est une *estimation* basée sur les informations disponibles. L'optimiseur effectue ses estimations en fonction des informations suivantes :

- statistiques à jour (si les statistiques ne sont pas à jour, réactualisez-les à l'aide de la commande RUNSTATS) ;
- paramètres de configuration ;
- options de définition des accès (BIND) ;
- classe d'optimisation des requêtes.

Les informations relatives au coût associées à un plan d'accès constitue la *meilleure estimation* de l'optimiseur en termes d'utilisation des ressources pour une requête. La durée réelle pour le traitement d'une requête peut varier en fonction de facteurs ne dépendant pas de DB2 (par exemple, le nombre d'applications s'exécutant simultanément). Cette durée réelle peut être mesurée lors de l'exécution de la requête au moyen d'un outil de contrôle des performances.

## Graphique de plan d'accès

Visual Explain utilise des informations provenant de différentes sources pour générer le graphique d'un plan d'accès, comme l'illustre la figure ci-dessous. En fonction des données entrées, l'optimiseur sélectionne un plan d'accès et Visual Explain affiche les informations sous la forme d'un *graphique de plan d'accès*. Les noeuds figurant sur ce graphique représentent les tables et les index, ainsi que toutes les opérations dont ils font l'objet. Les liens entre les noeuds correspondent au flux des données.



Les tâches ci-dessous correspondent à celles figurant sur l'illustration précédente (les lignes en pointillé indiquent des étapes obligatoires pour Visual Explain).

1. Conception des tables et réorganisation des données de table.
2. Création d'index.
3. Transmission à l'optimiseur de statistiques à jour (commande RUNSTATS).
4. Sélection des paramètres de configuration.
5. Sélection des options de définition d'accès.
6. Conception de requêtes pour l'extraction de données spécifiques .
7. Création d'un plan d'accès.
8. Création d'images instantanées EXPLAIN.
9. Affichage et utilisation du graphique d'un plan d'accès.

Par exemple, pour utiliser Visual Explain, lancez la commande **RUNSTATS** sur les tables et les index utilisés dans l'instruction, afin de générer des statistiques à jour. Ces statistiques, ainsi que les paramètres de configuration, les options de définition des accès et la requête elle-même permettent à l'optimiseur de créer un plan d'accès et une image instantanée EXPLAIN lors

de la définition des accès (BIND) du module. Visual Explain utilise cette image pour afficher le graphique de plan d'accès correspondant à l'instruction.

---

## Noeud d'un graphique de plan d'accès

Le graphique d'un plan d'accès est une représentation en arborescence faisant apparaître sous forme de *noeuds* les éléments suivants :

- des tables, représentées par des rectangles ;
- des index, représentés par des losanges ;
- des opérateurs, représentés par des octogones. Les opérateurs TQUEUE sont représentés par des parallélogrammes ;
- des fonctions de table, représentées par des hexagones.

---

## Classification

Les mises à jour d'une table entraînent progressivement une modification de l'emplacement des lignes sur les pages de données, et par conséquent une diminution du degré de *classification* entre l'index et les pages de données. La réorganisation de la table par rapport à un index spécifique permet de rétablir cette correspondance. Un index classifié est particulièrement intéressant pour les colonnes comportant des prédicats de type BETWEEN, car il permet un accès séquentiel plus efficace aux données de la table. Ceci entraîne une diminution du nombre d'extractions de pages puisque les valeurs du prédicat LIKE se trouvent sur la même page de données.

De manière générale, un seul index par table présente un degré de classification élevé.

Pour connaître le degré de classification d'un index, cliquez deux fois sur le noeud représentant ce dernier dans le graphique du plan d'accès. La fenêtre Statistiques sur index s'affiche. Les lignes CLUSTERRATIO et CLUSTERFACTOR indiquent le degré de correspondance de l'index. Si ces valeurs sont insuffisantes, il peut être utile de réorganiser les données de la table.

Pour plus de détails, reportez-vous à la section relative à la réorganisation des données de table dans le manuel *Administration Guide*.

---

## Conteneur

Un *conteneur* est un terme générique désignant l'emplacement physique de stockage des données. Il est associé à un espace table. Il peut s'agir d'un fichier, d'un répertoire ou d'une unité.

---

## Coût

Dans le cadre de Visual Explain, le *coût* représente une estimation totale des ressources utilisées pour l'exécution du plan d'accès associé à une instruction ou aux éléments d'une instruction. Ce coût est établi à partir d'une combinaison du coût UC (en nombre d'instructions) et du coût des entrées-sorties (en nombre d'opérations de recherche et de transfert de pages).

Il est mesuré en *timerons*. Un timeron n'est pas directement égal au temps effectif d'exécution, mais il fournit une estimation approximative des ressources (coût) nécessaires au gestionnaire de bases de données pour l'exécution de deux plans associés à la même requête.

La valeur indiquée pour chaque noeud opérateur du graphique d'un plan d'accès est un coût cumulé, prenant en compte l'exécution du plan jusqu'à cet opérateur inclus. Le coût n'intègre pas des facteurs tels que la charge du système ou le coût du renvoi de lignes de données à l'utilisateur.

---

## Groupage de lignes

Le *groupage de lignes* est une technique qui réduit les temps de traitement en permettant au gestionnaire de bases de données d'extraire un *groupe* de lignes en une seule opération. Ces lignes sont stockées en mémoire cache pour la durée de leur traitement. La mémoire cache nécessaire au stockage des lignes extraites est allouée lorsqu'une application émet une requête OPEN CURSOR, et désallouée lorsque le curseur est désactivé. Une fois toutes les lignes traitées, un autre groupe de lignes est extrait.

Utilisez l'option BLOCKING en conjonction avec les commandes **PREP** et **BIND** et attribuez-lui l'une des valeurs suivantes pour définir le type de groupage souhaité :

### UNAMBIG

Le groupage porte uniquement sur les curseurs non ambigus (valeur par défaut).

**ALL** Le groupage porte sur l'ensemble des curseurs (ambigus et non ambigus).

**NO** Aucun groupage n'est effectué.

Pour plus de détails, reportez-vous à la section relative au groupage de lignes dans le manuel *Administration Guide*.

---

## Espace table DMS (Database-managed space)

Il existe deux types d'espaces table dans une base de données : les espaces table DMS (Database Managed Space) et les espaces table SMS (System Managed Space).

Un espace table DMS est géré par le gestionnaire de bases de données. Il est spécialement conçu pour répondre à ses besoins.

La définition de l'espace table DMS comprend une liste de fichiers (ou d'unités) dans lesquels les données de la base sont stockées au format de l'espace table DMS.

Un espace table DMS permet d'ajouter les fichiers (ou unités) préalloués à un espace table existant pour en augmenter la capacité ; le gestionnaire de bases de données redistribue automatiquement les données existantes dans tous les conteneurs appartenant à l'espace table.

Les espaces table DMS et SMS peuvent coexister dans la même base de données.

---

## SQL dynamique

Les *instructions SQL dynamiques* sont des instructions SQL préparées et exécutées dans un programme d'application lors de son exécution. En SQL dynamique, les deux cas suivants peuvent se présenter :

- L'instruction SQL est émise de manière interactive via l'interface CLI ou l'interpréteur de commandes.
- La source SQL est contenue dans des variables du langage hôte intégrées dans un programme d'application.

Lorsque DB2 exécute une instruction SQL dynamique, l'optimiseur crée un plan d'accès en fonction des statistiques issues des catalogues et des paramètres de configuration en cours. Le plan d'accès associé à une instruction SQL dynamique peut être modifié d'une exécution du programme d'application correspondant à la suivante.

Une instruction SQL non dynamique est une instruction SQL statique.

---

## Image instantanée EXPLAIN

Visual Explain vous permet de consulter le contenu de l'image instantanée EXPLAIN.

Une *image instantanée EXPLAIN* est constituée de données comprimées, recueillies lors de l'explicitation d'une instruction SQL. L'image instantanée est stockée sous la forme d'un objet BLOB (Binary Large Object) dans la table EXPLAIN\_STATEMENT et contient les informations suivantes :

- représentation interne du plan d'accès, y compris les opérateurs, ainsi que les tables et les index auxquels un accès est établi ;
- critères utilisés par l'optimiseur pour l'établissement du plan d'accès, y compris les statistiques sur les objets de base de données et le coût cumulé de chaque opération.

Pour visualiser le plan d'accès d'une instruction SQL à l'aide de Visual Explain, vous devez disposer de l'image instantanée EXPLAIN correspondante. Pour que cette image soit créée, les conditions suivantes doivent être remplies :

1. Il existe des tables EXPLAIN dans le gestionnaire de bases de données pour le stockage des images instantanées. Pour plus de détails sur la création de ces tables, voir *Création de tables EXPLAIN* dans l'aide en ligne.
2. Dans le cas d'un module contenant des instructions SQL statiques, la valeur ALL ou YES est affectée à l'option EXPLSNAP lors de la préparation (PREP) ou de la définition des accès (BIND) du module. Vous obtenez ainsi une image instantanée EXPLAIN pour chaque instruction SQL explicitable contenue dans le module. Pour plus de détails sur les commandes **BIND** et **PREP**, reportez-vous au manuel *Command Reference*.
3. Dans le cas d'instructions SQL dynamiques, la valeur ALL est affectée à l'option EXPLSNAP lors de la définition des accès de l'application émettrice de ces instructions, ou la valeur YES ou EXPLAIN est affectée au registre CURRENT EXPLAIN SNAPSHOT avant l'exécution des instructions. Pour plus de détails, reportez-vous à la section relative aux images instantanées EXPLAIN dans le manuel *SQL Reference*.

---

## Instruction explicitable

Une *instruction explicitable* est une instruction SQL pouvant faire l'objet d'un EXPLAIN.

Les instructions SQL explicitables sont les suivantes :

- SELECT
- INSERT
- UPDATE

- DELETE
- VALUES

---

## Instruction explicitée

Une *instruction explicitée* est une instruction SQL ayant fait l'objet d'un EXPLAIN. Les instructions explicitées sont répertoriées dans la fenêtre Historique des instructions explicitées.

---

## Opérande

Un opérande est une entité faisant l'objet d'une opération. Les tables et les index constituent les opérandes de divers opérateurs (TBSCAN et IXSCAN, par exemple).

---

## Opérateur

Un *opérateur* représente une opération à exécuter sur des données - ou les données de sortie d'une table ou d'un index - lorsque le plan d'accès associé à une instruction SQL est exécuté.

Les opérateurs décrits ci-après sont utilisés dans les graphiques de plan d'accès :

### DELETE

Suppression de lignes dans une table.

### EISCAN

Balayage d'un index défini par l'utilisateur pour la création d'un ensemble réduit de lignes.

### FETCH

Extraction de colonnes d'une table à l'aide d'un identificateur d'enregistrement spécifique.

### FILTER

Filtrage des données par un ou plusieurs prédicats.

### GRPBY

Regroupement des lignes présentant la même valeur pour des colonnes ou des fonctions spécifiées, et évaluation de fonctions ensemblistes.

### HSJOIN

Jointure par hachage, dans laquelle une adresse calculée est affectée aux valeurs des colonnes de jointure des tables.

### INSERT

Insertion de lignes dans une table.

**IXAND**

Association (par AND) des identificateurs de ligne (RID) issus de plusieurs balayages d'index.

**IXSCAN**

Balayage d'un index de table avec spécification facultative de conditions de démarrage et d'arrêt et génération d'un flux ordonné de lignes.

**MSJOIN**

Jointure par fusion, dans laquelle table externe et table interne doivent respecter l'ordre des prédicats de jointure.

**NLJOIN**

Jointure par boucle imbriquée, dans laquelle un nouvel accès à la table interne est effectué pour chaque ligne de la table externe.

**RETURN**

Renvoi à l'utilisateur des données résultant de la requête.

**RIDSCAN**

Balayage d'une liste d'identificateurs de ligne (RID) obtenus à partir d'un ou plusieurs index.

**SHIP**

Extraction de données à partir d'une source de base de données éloignée. Opérateur utilisé dans les systèmes de bases de données fédérées.

**SORT**

Tri des lignes dans l'ordre des colonnes spécifiées, avec suppression facultative des doublons.

**TBSCAN**

Extraction des lignes par lecture directe de l'ensemble des données requises dans les pages de données.

**TEMP**

Stockage des données dans une table temporaire susceptible d'être relue (à plusieurs reprises, si nécessaire).

**TQUEUE**

Transfert de données de table entre plusieurs agents de base de données.

**UNION**

Concaténation de flux de lignes issus de plusieurs tables.

**UNIQUE**

Suppression des lignes contenant des doublons, pour les colonnes spécifiées.

**UPDATE**

Mise à jour des lignes d'une table.

---

## CMPEXP

**Opérateur** : CMPEXP

**Fonction** : Calcul d'expressions nécessaires à l'établissement de résultats intermédiaires ou finaux.

(Cet opérateur n'est utilisé qu'en mode débogage.)

---

## DELETE

**Opérateur** : DELETE

**Fonction** : Suppression de lignes dans une table.

Cet opérateur représente une opération obligatoire. Pour réduire les coûts d'un plan d'accès, attachez-vous plutôt à modifier les autres opérateurs (de balayage ou de jointure, par exemple) définissant l'ensemble de lignes à supprimer.

**Suggestion** :

- Pour supprimer toutes les lignes d'une table, pensez à utiliser l'instruction DROP TABLE ou la commande **LOAD REPLACE**.

---

## EISCAN

**Opérateur** : EISCAN

**Fonction** : Balayage d'un index défini par l'utilisateur pour la création d'un ensemble réduit de lignes. Ce balayage utilise les conditions multiples de démarrage/arrêt à partir de la fonction de production de plages fournie par l'utilisateur.

Cette opération est destinée à réduire le nombre de lignes qualifiantes avant d'accéder à la table de base (à l'aide des prédicats).

**Suggestion** :

- Les mises à jour de la base de données entraînent progressivement une fragmentation de l'index avec, pour conséquence, une augmentation inutile de son nombre de pages. Pour y remédier, supprimez puis recréez l'index ou réorganisez-le.
- Si les statistiques ne sont pas à jour, réactualisez-les à l'aide de la commande RUNSTATS.

---

## FETCH

**Opérateur** : FETCH

**Fonction** : Extraction de colonnes d'une table à l'aide d'un identificateur de ligne spécifique (RID).

**Suggestions** :

- Modifiez les clés d'index pour y inclure les colonnes à extraire, de sorte que l'accès aux pages de données ne soit pas nécessaire.
- Identifiez l'index associé à l'opération d'extraction et cliquez deux fois sur le noeud correspondant pour en afficher les statistiques. Vérifiez que l'index présente un degré de correspondance élevé.
- Augmentez la taille de la mémoire tampon si les entrées-sorties (E-S) résultant de l'extraction sont supérieures au nombre de pages de la table.
- Si les statistiques ne sont pas à jour, réactualisez-les à l'aide de la commande RUNSTATS.

Les statistiques sur les quantiles et les valeurs les plus fréquentes fournissent des informations sur la sélectivité des prédicats, permettant ainsi de déterminer si un balayage d'index est préférable à un balayage de table. Pour mettre à jour ces valeurs, exécutez la commande **RUNSTATS** sur la table en utilisant la clause **WITH DISTRIBUTION**.

---

## FILTER

**Opérateur** : FILTER

**Fonction** : Application de prédicats résiduels permettant un filtrage des données suivant les critères définis par les prédicats.

**Suggestions** :

- Vérifiez que les prédicats utilisés entraînent l'extraction des seules données voulues. Vérifiez, par exemple, que leur sélectivité correspond bien à la partie de la table dont vous souhaitez disposer.
- Vérifiez que la classe d'optimisation est au moins égale à 3, de sorte que l'optimiseur utilise une jointure plutôt qu'une sous-requête. Si cela n'est pas possible, essayez de récrire la requête SQL pour supprimer la sous-requête. Pour plus de détails, reportez-vous à la section "Query rewrites by the SQL compiler" dans le manuel *Administration Guide*.

---

## GENROW

**Nom** : GENROW

**Fonction** : Fonction intégrée générant une table de lignes, sans utiliser de données d'entrée issues de tables, d'index ou d'opérateurs.

La fonction GENROW peut être utilisée par l'optimiseur pour générer des lignes de données (par exemple, pour une instruction INSERT ou pour certains IN-listes convertis en jointures).

Pour visualiser les statistiques estimées sur les tables créées par la fonction GENROW, cliquez deux fois sur le noeud correspondant à cette dernière.

---

## GRPBY

**Opérateur** : GRPBY

**Fonction** : Regroupement des lignes présentant la même valeur pour des colonnes ou des fonctions spécifiées. Cette opération est nécessaire à la création d'un groupe de valeurs ou à l'évaluation d'une fonction ensembliste.

Même si aucune colonne GROUP BY n'est spécifiée, l'opérateur GRPBY peut être utilisé si la liste SELECT comporte des fonctions d'agrégation. Il permet d'indiquer que l'ensemble de la table doit être traité comme un seul groupe lors de l'agrégation.

**Suggestions** :

- Cet opérateur représente une opération obligatoire. Pour réduire les coûts d'un plan d'accès, attachez-vous plutôt à modifier les autres opérateurs (de balayage ou de jointure, par exemple) définissant l'ensemble de lignes à regrouper.
- Pour optimiser les résultats d'une instruction SELECT contenant une seule fonction d'agrégation mais pas de clause GROUP BY, procédez comme suit :
  - Dans le cas d'une fonction d'agrégation MIN(C), créez un index en ordre croissant sur C.
  - Dans le cas d'une fonction d'agrégation MAX(C), créez un index en ordre décroissant sur C.

---

## HSJOIN

**Opérateur** : HSJOIN

**Fonction** : Jointure par hachage dans laquelle un adressage calculé est appliqué aux lignes qualifiées dans les tables pour permettre une jointure directe, sans que le contenu des tables ait à être ordonné au préalable.

Une opération de jointure est nécessaire dès lors qu'une clause FROM fait référence à plus d'une table. Elle peut s'effectuer par hachage lorsqu'un prédicat de jointure établit une égalité entre les colonnes de deux tables. Les prédicats de jointure doivent être exactement du même type de données. La jointure par hachage peut également résulter d'une sous-requête réécrite, comme c'est le cas avec NLJOIN.

Pour effectuer ce type de jointure, il n'est pas nécessaire que les tables d'entrée soient triées. La jointure est réalisée en balayant la table interne et en générant une table de référence par l'affectation d'une adresse aux valeurs des colonnes de jointure. La jointure lit ensuite la table externe, en affectant une adresse aux valeurs des colonnes de jointure et en la comparant à celles de la table de référence créée pour la table interne.

Pour plus de détails, reportez-vous à la section sur les concepts de jointure dans le manuel *Administration Guide*.

**Suggestions** :

- Utilisez des prédicats locaux (faisant référence à une seule table) pour limiter le nombre de lignes à joindre.
- Augmentez la taille de la mémoire dynamique de tri afin qu'elle puisse contenir la table de référence obtenue par hachage.
- Si les statistiques ne sont pas à jour, réactualisez-les à l'aide de la commande RUNSTATS.

---

## INSERT

**Opérateur** : INSERT

**Fonction** : Insertion de lignes dans une table.

Cet opérateur représente une opération obligatoire. Pour réduire les coûts d'un plan d'accès, attachez-vous plutôt à modifier les autres opérateurs (de balayage ou de jointure, par exemple) définissant l'ensemble de lignes à insérer.

**Opérateur : IXAND**

**Fonction :** Association ("index ANDing") des résultats obtenus suite à plusieurs balayages d'index avec les techniques Dynamic Bitmap. L'opérateur permet d'appliquer à plusieurs index des prédicats ayant été associés (par AND) afin de réduire les accès aux tables sous-jacentes.

Cet opérateur permet d'effectuer les opérations suivantes :

- réduire le nombre de lignes avant d'accéder à la table de base ;
- associer (par AND) des prédicats s'appliquant à plusieurs index ;
- associer (par AND) les résultats de semi-jointures utilisés pour les jointures en étoile.

**Suggestions :**

- Les mises à jour de la base de données entraînent progressivement une fragmentation de l'index avec, pour conséquence, une augmentation inutile de son nombre de pages. Pour y remédier, supprimez puis recréez l'index ou réorganisez-le.
- Si les statistiques ne sont pas à jour, réactualisez-les à l'aide de la commande RUNSTATS.
- En règle générale, les résultats d'un balayage d'index sont d'autant plus satisfaisants que le nombre de lignes qualifiantes est faible. L'optimiseur utilise les statistiques disponibles sur les colonnes référencées dans les prédicats pour évaluer le nombre de lignes qualifiantes. Si certaines valeurs sont plus fréquentes que d'autres, il est important de disposer de statistiques de distribution. Pour cela, utilisez la commande **RUNSTATS** avec la clause WITH DISTRIBUTION. L'utilisation des statistiques de distribution non uniforme permet à l'optimiseur de distinguer les valeurs en fonction de leur fréquence d'apparition.
- IXAND est particulièrement efficace sur les index en une seule colonne dans la mesure où l'utilisation des clés de début et de fin n'est pas très adaptée avec cet opérateur.
- Pour les jointures en étoile, créez des index en une seule colonne pour chacune des colonnes les plus sélectives dans la table centrale et les tables de dimensions associées.

**Opérateur** : IXSCAN

**Fonction** : Balayage d'un index pour la création d'un ensemble réduit de lignes. Si vous le souhaitez, vous pouvez utiliser des conditions START ou STOP ou des prédicats indexables comportant des références à des colonnes de l'index.

Cette opération est destinée à réduire le nombre de lignes qualifiantes avant d'accéder à la table de base (à l'aide des prédicats).

Pour plus de détails, reportez-vous à la section relative au balayage d'index dans le manuel *Administration Guide*.

**Suggestions** :

- Les mises à jour de la base de données entraînent progressivement une fragmentation de l'index avec, pour conséquence, une augmentation inutile de son nombre de pages. Pour y remédier, supprimez puis recréez l'index ou réorganisez-le.
- Lorsque l'accès à plusieurs tables est nécessaire, il est préférable d'accéder à la table interne via un index car le système dispose ainsi d'un index sur la colonne de jointure de la table externe.  
Pour plus de détails sur les index, reportez-vous à l'aide en ligne relative à Visual Explain.
- Si les statistiques ne sont pas à jour, réactualisez-les à l'aide de la commande RUNSTATS.
- En règle générale, les résultats d'un balayage d'index sont d'autant plus satisfaisants que le nombre de lignes qualifiantes est faible. L'optimiseur utilise les statistiques disponibles sur les colonnes référencées dans les prédicats pour évaluer le nombre de lignes qualifiantes. Si certaines valeurs sont plus fréquentes que d'autres, il est important de disposer de statistiques de distribution. Pour cela, utilisez la commande **RUNSTATS** avec la clause WITH DISTRIBUTION. L'utilisation des statistiques de distribution non uniforme permet à l'optimiseur de distinguer les valeurs en fonction de leur fréquence d'apparition.

---

## MSJOIN

**Opérateur :** MSJOIN

**Fonction :** Jointure de tables par fusion, dans laquelle les lignes qualifiées des tables externe et interne doivent respecter l'ordre des prédicats de jointure.

Une opération de jointure est nécessaire dès lors qu'une clause FROM fait référence à plus d'une table. Elle peut s'effectuer par fusion lorsqu'un prédicat de jointure établit une égalité entre les colonnes de deux tables. La jointure par fusion peut également résulter d'une sous-requête réécrite.

La jointure par fusion nécessite que les données des colonnes de jointure soient ordonnées correctement car un seul balayage des tables est normalement effectué. Pour cela, le système accède à un index ou à une table triée.

Pour plus de détails, reportez-vous à la section sur les concepts de jointure dans le manuel *Administration Guide*.

**Suggestions :**

- Utilisez des prédicats locaux (faisant référence à une seule table) pour limiter le nombre de lignes à joindre.

Pour plus de détails sur les index, reportez-vous à la rubrique Création d'index dans l'aide en ligne relative à Visual Explain.

- Si les statistiques ne sont pas à jour, réactualisez-les à l'aide de la commande RUNSTATS.

---

## NLJOIN

**Opérateur :** NLJOIN

**Fonction :** Jointure par boucle imbriquée consistant à balayer (généralement par balayage d'index) la table interne pour chaque ligne de la table externe.

Une opération de jointure est nécessaire dès lors qu'une clause FROM fait référence à plus d'une table. Lorsque la jointure s'effectue par boucle imbriquée, l'utilisation d'un prédicat de jointure est facultative mais permet d'optimiser les performances.

Une jointure par boucle imbriquée s'effectue :

- soit par balayage de la table interne pour chaque ligne faisant l'objet d'un accès dans la table externe ;
- soit par consultation de l'index de la table interne pour chaque ligne faisant l'objet d'un accès dans la table externe.

Pour plus de détails, reportez-vous à la section sur les concepts de jointure dans le manuel *Administration Guide*.

**Suggestions :**

- La jointure par boucle imbriquée donne de meilleurs résultats s’il existe un index pour les colonnes de la table interne (affichée à droite de l’opérateur NLJOIN) sur lesquelles porte le prédicat de jointure. Vérifiez si la table interne est de type TBSCAN ou IXSCAN. S’il s’agit d’une table de type TBSCAN, il peut être préférable de créer un index sur les colonnes de jointure de cette table.

La création d’un index sur les colonnes de jointure de la table externe permet, dans une moindre mesure, d’obtenir de meilleurs résultats.

Pour plus de détails sur les index, reportez-vous à la rubrique Création d’index dans l’aide en ligne relative à Visual Explain.

- Si les statistiques ne sont pas à jour, réactualisez-les à l’aide de la commande RUNSTATS.

**Informations connexes :**

- Jointure en étoile.

---

## PIPE

**Opérateur :** PIPE

**Fonction :** Transfert de lignes vers d’autres opérateurs sans modification de celles-ci.

(Cet opérateur n’est utilisé qu’en mode débogage.)

---

## RETURN

**Opérateur :** RETURN

**Fonction :** Renvoi à l’utilisateur des données résultant d’une requête. Il s’agit du dernier opérateur d’un graphique de plan d’accès, qui indique les valeurs et les coûts totaux cumulés pour l’ensemble du plan.

Cet opérateur représente une opération obligatoire.

**Suggestion :**

- Vérifiez que les prédicats utilisés entraînent l’extraction des seules données voulues. Vérifiez, par exemple, que leur sélectivité correspond bien à la partie de la table dont vous souhaitez disposer.

---

## RIDSCN

**Opérateur** : RIDSCN

**Fonction** : Balayage d'une liste d'identificateurs de ligne (RID) obtenus à partir d'un ou plusieurs index.

Cet opérateur est pris en compte par l'optimiseur dans les cas suivants :

- Les prédicats sont reliés par le mot clé OR, ou un prédicat IN est utilisé. Une méthode appelée "index ORing" peut alors être utilisée. Elle consiste à effectuer plusieurs accès à un ou plusieurs index de la même table et à combiner les résultats de chaque accès.
- L'accès à un seul index est nécessaire mais celui-ci présente un faible degré de correspondance. Il est avantageux alors d'exécuter une lecture anticipée de liste (LIST PREFETCH) car le tri des identificateurs de ligne avant l'accès aux lignes de la base permet de réduire considérablement les opérations d'entrées-sorties.

---

## RQUERY

**Opérateur** : SHIP

**Fonction**: Extraction, dans un système de bases de données fédérées, de données à partir d'une source de données éloignée. Cet opérateur est pris en compte par l'optimiseur lorsqu'un opérateur SHIP envoie une instruction SQL SELECT à une source de données éloignée pour extraire le résultat de la requête. L'instruction SELECT est générée dans un dialecte SQL pris en charge par la source de données, et peut contenir toute requête valide en fonction de la source de données.

**Suggestion** : Reportez-vous à la section "Federated Database Query and Network Tuning Information" du chapitre 4 dans le manuel Administration Guide Vol 2.

---

## SORT

**Opérateur** : SORT

**Fonction** : Tri des lignes d'une table sur une ou plusieurs colonnes, avec suppression facultative des entrées en double.

Une opération de tri est nécessaire lorsqu'il n'existe pas d'index permettant de disposer de données ordonnées correctement, ou que le balayage de l'index est trop coûteux. Elle constitue souvent la dernière opération exécutée, une fois l'extraction des lignes requises terminée, ou un préalable à l'opération JOIN ou GROUP BY.

Si le nombre de lignes est élevé ou s'il est impossible d'établir un tube (rediriger vers un autre processus) pour les données triées, l'opération exige la création coûteuse de tables temporaires.

Pour plus de détails sur les tris, reportez-vous au manuel *Administration Guide*.

### **Suggestions :**

- Envisagez de créer un index sur les colonnes à trier.  
Pour plus de détails sur les index, reportez-vous à la rubrique Création d'index dans l'aide en ligne relative à Visual Explain.
- Vérifiez que les prédicats utilisés entraînent l'extraction des seules données voulues. Vérifiez, par exemple, que leur sélectivité correspond bien à la partie de la table dont vous souhaitez disposer.
- Vérifiez que la valeur du paramètre PREFETCHSIZE de l'espace table temporaire du système est suffisante. Pour cela, sélectionnez **Instruction->Affichage des statistiques->Espaces table**.
- Si des opérations de tri conséquentes sont fréquemment nécessaires, il est recommandé d'augmenter la valeur des paramètres de configuration suivants :
  - Taille de la mémoire dynamique de tri (sortheap). Pour modifier ce paramètre, sélectionnez la base de données dans la fenêtre Centre de contrôle et cliquez sur *Configuration* dans le menu en incrustation correspondant. Sélectionnez l'onglet **Performances** dans le bloc-notes qui s'affiche.
  - Seuil du segment de mémoire dynamique de tri (sheapthres). Pour modifier ce paramètre, sélectionnez l'instance de base de données dans la fenêtre Centre de contrôle et cliquez sur *Configuration* dans le menu en incrustation correspondant. Sélectionnez l'onglet **Performances** dans le bloc-notes qui s'affiche.
- Si les statistiques ne sont pas à jour, réactualisez-les à l'aide de la commande RUNSTATS.

---

## **TBSCAN**

**Opérateur :** TBSCAN

**Fonction :** Balayage d'une table permettant l'extraction de lignes par lecture directe de l'ensemble des données requises dans les pages de données.

L'optimiseur choisit ce type de balayage plutôt qu'un balayage d'index dans les cas suivants :

- La plage de valeurs recherchées apparaît fréquemment (un accès est donc nécessaire à une grande partie de la table).

- Le volume de la table est réduit.
- Le degré de correspondance de l'index est faible.
- Il n'existe pas d'index.

Pour plus de détails sur le balayage de table et d'index, reportez-vous au manuel *Administration Guide*.

### Suggestions :

- Il est préférable d'effectuer un balayage d'index si la table est volumineuse et qu'il n'est pas nécessaire d'accéder à la plupart des lignes de la table. Pour favoriser l'utilisation de l'index par l'optimiseur, envisagez de créer des index sur les colonnes pour lesquelles il existe des prédicats sélectifs. Pour plus de détails sur les index, reportez-vous à la rubrique Création d'index dans l'aide en ligne relative à Visual Explain.
- S'il existe un index mais que celui-ci n'est pas utilisé, assurez-vous que des prédicats sélectifs sont définis pour chacune des premières colonnes de cet index. Si ces prédicats existent, vérifiez alors que le degré de correspondance de l'index par rapport à la table est élevé. Pour cela, affichez la fenêtre Statistiques sur table correspondant à la table sur laquelle porte le tri et cliquez sur le bouton de fonction *Index* pour accéder à la fenêtre Statistiques sur index.
- Vérifiez que la valeur du paramètre PREFETCHSIZE (nombre de pages en lecture anticipée) définie pour l'espace table est suffisante. Pour cela, sélectionnez **Instruction->Affichage des statistiques->Espaces table**. Pour plus de détails, reportez-vous à la section relative à la lecture anticipée des données dans la mémoire tampon, dans le manuel *Administration Guide*.
- Si les statistiques ne sont pas à jour, réactualisez-les à l'aide de la commande RUNSTATS.  
Les statistiques sur les quantiles et les valeurs les plus fréquentes fournissent des informations sur la sélectivité des prédicats, permettant ainsi de déterminer si un balayage d'index est préférable à un balayage de table. Pour mettre à jour ces valeurs, exécutez la commande **RUNSTATS** sur la table en utilisant la clause WITH DISTRIBUTION.

---

## TEMP

### Opérateur : TEMP

**Fonction** : Stockage des données dans une table temporaire qui sera lue par un autre opérateur (à plusieurs reprises, si nécessaire). La table est supprimée une fois le traitement de l'instruction SQL terminé, voire même avant la fin de son exécution.

Cet opérateur est nécessaire à l'évaluation des sous-requêtes ou au stockage de résultats intermédiaires. Dans certains cas, par exemple celui d'une instruction pouvant être mise à jour, il devient obligatoire.

---

## TQUEUE

**Opérateur** : TQUEUE

**Fonction** : File d'attente de tables utilisée pour permettre l'échange de données de table entre des agents de base de données lorsque plusieurs agents traitent une requête. Plusieurs agents de base de données sont utilisés pour traiter une requête lorsque la fonction de parallélisme est mise en oeuvre.

Les types de file d'attente de tables suivants sont utilisés :

- **Locale** : La file d'attente de tables est utilisée pour permettre l'échange de données entre des agents de base de données sur un même noeud. Une file d'attente de tables locale est utilisée pour le parallélisme intra-partition.
- **Non locale** : La file d'attente de tables est utilisée pour permettre l'échange de données entre des agents de base de données sur des noeuds différents.

---

## UNION

**Opérateur** : UNION

**Fonction** : Concaténation d'ensembles de lignes issues de plusieurs tables.

Cet opérateur représente une opération obligatoire. Pour réduire les coûts d'un plan d'accès, attachez-vous plutôt à modifier les autres opérateurs (de balayage ou de jointure, par exemple) définissant l'ensemble de lignes à concaténer.

---

## UNIQUE

**Opérateur** : UNIQUE

**Fonction** : Suppression des lignes comportant des doublons dans les colonnes spécifiées.

**Suggestion** :

- Cet opérateur peut être évité uniquement s'il existe un index à entrées uniques sur les colonnes appropriées.

Pour plus de détails sur les index, reportez-vous à la rubrique Création d'index dans l'aide en ligne relative à Visual Explain.

---

## UPDATE

**Opérateur** : UPDATE

**Fonction** : Mise à jour des lignes d'une table.

Cet opérateur représente une opération obligatoire. Pour réduire les coûts d'un plan d'accès, attachez-vous plutôt à modifier les autres opérateurs (de balayage ou de jointure, par exemple) définissant l'ensemble de lignes à mettre à jour.

---

## Optimiseur

L'*optimiseur* est la fonction du compilateur SQL qui détermine le plan d'accès à associer à une instruction SQL. Le choix s'effectue par modélisation du coût d'exécution de plusieurs plans d'accès et sélection du plan présentant le plus faible coût estimé.

---

## Module

Un *module* est un objet de base de données, comprenant les informations nécessaires au traitement des instructions SQL associées à un fichier source d'un programme d'application. Il est généré suite à l'une des opérations suivantes :

- précompilation d'un fichier source par la commande **PREP**.
- exécution d'une commande **BIND** sur le fichier de liens généré par le précompilateur.

---

## Prédicat

Un *prédicat* est un élément d'une condition de recherche entraînant une opération de comparaison explicite ou implicite. Les prédicats sont utilisés dans les clauses commençant par WHERE ou HAVING.

Par exemple, dans l'instruction SQL suivante :

```
SELECT * FROM SAMPLE
  WHERE NAME = 'SMITH' AND
  DEPT = 895 AND YEARS > 5
```

Les prédicats sont : NAME = 'SMITH', DEPT = 895 et YEARS > 5.

On distingue les types de prédicats suivants (par ordre décroissant d'efficacité) :

1. Les prédicats de type BETWEEN délimitent la plage de balayage d'un index.

2. Les prédicats d'index SARGables peuvent être utilisés comme arguments de recherche dans les pages d'index si les colonnes sur lesquelles ils portent font partie de la clé d'index.
3. Les prédicats de données SARGables ne peuvent pas s'appliquer à un index. Ils peuvent être utilisés tant que les lignes de la table sont conservées en mémoire tampon.
4. Les prédicats résiduels nécessitent généralement plus qu'un simple accès à une table de base. Ils supposent des opérations d'entrée-sortie et doivent être appliqués une fois les données de la mémoire tampon copiées sur disque. Il s'agit de prédicats comprenant des sous-requêtes ou de prédicats entraînant la lecture de données LONG VARCHAR ou LOB dans des fichiers distincts de la table.

Lorsque vous définissez un prédicat, veillez à ce qu'il soit le plus sélectif possible, de sorte qu'un minimum de lignes soient renvoyées.

Les prédicats suivants donnent les meilleurs résultats et sont le plus communément utilisés :

- Un *prédicat de jointure par égalité simple* est nécessaire aux opérations de jointure par fusion. Il se présente sous la forme `table1.colonne = table2.colonne`, et permet d'établir une égalité entre les colonnes de tables distinctes afin de joindre ces deux tables.
- Un *prédicat local* s'applique à une seule table.

Pour plus de détails, reportez-vous aux sections relatives aux concepts d'accès aux données et d'optimisation dans le manuel *Administration Guide*.

---

## Classe d'optimisation de requêtes

Une *classe d'optimisation de requêtes* est un ensemble de règles de réécriture et de techniques d'optimisation utilisé pour la compilation des requêtes.

Les principales classes d'optimisation sont les suivantes :

- 1 Optimisation limitée. Le choix de cette classe est recommandé lorsque la mémoire et la capacité de traitement sont très limitées. S'apparente au degré d'optimisation disponible dans la version 1.
- 2 Optimisation faible. Spécifie un niveau d'optimisation supérieur à celui de la version 1, mais à un coût très inférieur à celui des niveaux 3 et supérieurs, particulièrement pour les requêtes très complexes.
- 3 Optimisation modérée. Présente des caractéristiques très proches de l'optimisation sous DB2 pour MVS/ESA.

- 5 Optimisation normale. Le choix de cette classe est recommandé dans les environnements utilisant à la fois des transactions simples et des requêtes complexes.
- 7 Optimisation normale. Présente les mêmes caractéristiques que la classe d'optimisation 5 mais ne réduit pas le degré d'optimisation des requêtes SQL dynamiques complexes.

Les classes d'optimisation suivantes ne doivent être utilisées que dans certaines circonstances particulières :

- 0 Optimisation minimale. Cette classe doit être utilisée uniquement lorsqu'aucune optimisation n'est requise (c'est-à-dire, dans le cas de requêtes très simples émises sur des tables parfaitement indexées).
- 9 Optimisation maximale. Cette classe d'optimisation est très consommatrice de mémoire et de capacité de traitement. Ne l'utilisez que si la classe 5 est insuffisante (c'est-à-dire, si les résultats obtenus pour les requêtes très complexes et longues à exécuter ne sont pas satisfaisants lorsque la classe 5 est activée).

De manière générale, utilisez une classe d'optimisation plus élevée pour les requêtes statiques et celles dont la durée d'exécution risque d'être longue, et choisissez une classe d'optimisation plus basse pour les requêtes simples émises de façon dynamique ou exécutées occasionnellement.

Pour définir la classe d'optimisation d'instructions SQL dynamiques, entrez la commande suivante via l'interpréteur de commandes :

```
SET CURRENT QUERY OPTIMIZATION = n;
```

où 'n' représente la classe d'optimisation choisie.

Pour définir la classe d'optimisation d'instructions SQL statiques, utilisez l'option `QUERYOPT` avec des commandes **BIND** ou **PREP**.

Pour plus de détails, reportez-vous à la section sur l'ajustement de la classe d'optimisation dans le manuel *Administration Guide*.

---

## Sélectivité des prédicats

La *sélectivité* renvoie à la probabilité qu'a une ligne de satisfaire les conditions d'un prédicat (c'est-à-dire, de vérifier ces conditions).

Par exemple, si la sélectivité d'un prédicat utilisé sur une table de 1 000 000 lignes est égale à 0,01 (ou 1%), le nombre de lignes renvoyées par le prédicat est estimé à 10 000 (1% de 1 000 000), et celui de lignes supprimées, à 990 000.

Il est préférable d'utiliser des prédicats très sélectifs, c'est-à-dire présentant une sélectivité inférieure ou égale à 0,10. En effet, ces prédicats permettent de réduire le nombre de lignes à traiter par les opérateurs suivants, limitant ainsi les coûts UC et E-S liés à la requête.

### Exemple

Supposons que vous disposiez d'une table de 1 000 000 lignes et que votre première requête comporte une clause 'ORDER BY' nécessitant une opération supplémentaire de tri. Si vous utilisez un prédicat ayant une sélectivité de 0,01, le tri portera sur 10 000 lignes, alors qu'avec un prédicat moins sélectif de 0,50, le tri devra être effectué sur 500 000 lignes, augmentant considérablement les temps UC et la durée des entrées-sorties.

---

## Jointure en étoile

Un ensemble de jointures est considéré comme jointure en étoile lorsqu'une table centrale (table de grande taille) est jointe au moins à deux tables de dimensions (tables de plus petite taille contenant des descriptions des valeurs de colonne de la table centrale).

Une jointure en étoile comporte trois parties principales :

- Semi-jointures
- Association par AND des résultats des semi-jointures
- Achèvement des semi-jointures

Elle apparaît sous la forme de deux jointures fournissant des données à un opérateur IXAND.

Une semi-jointure est un type de jointure particulier dans laquelle le résultat de la jointure consiste uniquement en l'identificateur de ligne (RID) de la table interne, et non en la jointure des colonnes des tables interne et externe.

Les jointures en étoile utilisent des semi-jointures pour fournir les identificateurs de ligne à un opérateur AND appliqué à un index. Celui-ci cumule l'effet de filtrage des différentes jointures. Le résultat de l'opération IXAND est ensuite fourni en entrée à un opérateur IXOR, qui classe les identificateurs de ligne et élimine toute ligne en double éventuellement issue des jointures fournies à l'opérateur IXAND. Les lignes de la table centrale sont ensuite extraites au moyen d'un opérateur Fetch. Enfin, la table centrale réduite est jointe aux tables de dimensions pour achever les jointures.

### Suggestions :

- Créez des index sur la table centrale pour chacune des jointures de table de dimensions.

- Vérifiez que la taille de la mémoire dynamique de tri est suffisamment importante pour allouer la mémoire nécessaire au filtre de données de l'opérateur IXAND. Dans le cas de jointure en étoile, celui-ci peut nécessiter jusqu'à 12 Mo, soit 3 000 pages de 4 ko. Pour le parallélisme intra-partition, le filtre de données est alloué à partir du même segment de mémoire partagée que la mémoire dynamique de la base de données, mais il est limité par la taille de la mémoire dynamique de tri (et par le seuil de cette mémoire associé à l'instance). Par conséquent, la mémoire partagée dépend de la taille de la mémoire dynamique de tri et de son seuil, et peut donc requérir plus de 12 Mo.
- Appliquez des prédicats de filtrage aux tables de dimensions. Si les statistiques ne sont pas à jour, réactualisez-les à l'aide de la commande RUNSTATS.

---

## SQL statique

Une *instruction SQL statique* est une instruction intégrée à un programme d'application. Toutes ces instructions intégrées doivent être précompilées et faire l'objet d'une liaison (BIND) dans un *module* pour que l'application puisse s'exécuter.

Lors de la compilation d'instructions SQL statiques, DB2 crée un plan d'accès basé sur les statistiques issues du catalogue et sur les paramètres de configuration en cours au moment de l'opération.

Ces plans sont utilisés lors de chaque exécution de l'application et ne sont pas modifiés tant qu'un nouveau BIND du module n'est pas exécuté.

Une instruction SQL non statique est une instruction SQL dynamique.

---

## Espace table SMS (System-managed space)

Deux types d'espaces table peuvent exister dans une base de données : les espaces table SMS (System Managed Space) et les espaces table DMS (Database Managed Space).

Un espace table SMS est géré par le système d'exploitation. Celui-ci stocke les données de la base de données dans l'espace affecté lors de la création d'un espace table. La définition de l'espace table contient une liste d'un ou plusieurs chemins d'accès aux répertoires où sont stockées ces données.

Le système de fichiers gère l'allocation et la gestion de la mémoire externe.

Les espaces table SMS et DMS peuvent coexister dans la même base de données.

---

## Espace table

La gestion de bases de données volumineuses est plus facile si elles sont partitionnées en sections gérées séparément, nommées *espaces table*.

Un espace table permet de définir l'emplacement des données sur des unités logiques particulières ou sur des parties de ces dernières. Par exemple, lors de la création d'une table, vous pouvez demander que les index ou les colonnes comportant des données d'objets LOB soient stockés dans un espace table distinct.

Un espace table peut être réparti sur plusieurs unités physiques (ou conteneurs) pour augmenter les performances du système. Il est néanmoins recommandé que tous les conteneurs ou unités d'un même espace table aient les mêmes caractéristiques, en termes de performances.

Un espace table peut être géré de deux façons différentes : soit comme un espace SMS (System Managed Space), soit comme un espace DMS (Database Managed Space).

---

## Visual Explain

**Remarque :** A compter de la version 6, Visual Explain ne peut plus être appelé à partir de la ligne de commande. Cependant, vous pouvez toujours l'appeler à partir de divers objets de base de données dans le Centre de contrôle. Pour la présente version, le nom Visual Explain continue à être utilisé dans la documentation.

Visual Explain permet de visualiser sous forme de graphique le plan d'accès associé aux instructions SQL explicitées. Vous pouvez utiliser les informations fournies par le graphique pour adapter vos instructions SQL à des fins d'optimisation des performances.

Un graphique de plan d'accès permet de visualiser des informations détaillées sur les éléments suivants :

- tables (et colonnes correspondantes) et index,
- opérateurs (tels que balayages, tris et jointures de tables),
- espaces table et fonctions.

Visual Explain peut également servir à :

- visualiser les statistiques disponibles lors de l'optimisation. Celles-ci peuvent ensuite être comparées aux statistiques en cours du catalogue, afin de déterminer si une redéfinition des accès du module serait susceptible d'améliorer les performances ;

- déterminer si un index a été utilisé pour accéder à une table. Si ce n'est pas le cas, Visual Explain peut vous aider à identifier les colonnes qui pourraient être indexées ;
- visualiser les différentes versions d'un graphique de plan d'accès (avant et après adaptation), pour comparer les effets des méthodes d'ajustement utilisées ;
- obtenir des informations sur chaque opération exécutée dans le cadre d'un plan d'accès, y compris sur le coût total estimé et le nombre de lignes extraites (cardinalité).



---

## Annexe B. Liste alphabétique des opérateurs Visual Explain

---

### CMPEXP

**Opérateur** : CMPEXP

**Fonction** : Calcul d'expressions nécessaires à l'établissement de résultats intermédiaires ou finaux.

(Cet opérateur n'est utilisé qu'en mode débogage.)

---

### DELETE

**Opérateur** : DELETE

**Fonction** : Suppression de lignes dans une table.

Cet opérateur représente une opération obligatoire. Pour réduire les coûts d'un plan d'accès, attachez-vous plutôt à modifier les autres opérateurs (de balayage ou de jointure, par exemple) définissant l'ensemble de lignes à supprimer.

**Suggestion** :

- Pour supprimer toutes les lignes d'une table, pensez à utiliser l'instruction DROP TABLE ou la commande **LOAD REPLACE**.

---

### EISCAN

**Opérateur** : EISCAN

**Fonction** : Balayage d'un index défini par l'utilisateur pour la création d'un ensemble réduit de lignes. Ce balayage utilise les conditions multiples de démarrage/arrêt à partir de la fonction de production de plages fournie par l'utilisateur.

Cette opération est destinée à réduire le nombre de lignes qualifiantes avant d'accéder à la table de base (à l'aide des prédicats).

**Suggestion** :

- Les mises à jour de la base de données entraînent progressivement une fragmentation de l'index avec, pour conséquence, une augmentation inutile de son nombre de pages. Pour y remédier, supprimez puis recréez l'index ou réorganisez-le.
- Si les statistiques ne sont pas à jour, réactualisez-les à l'aide de la commande RUNSTATS.

---

## FETCH

**Opérateur** : FETCH

**Fonction** : Extraction de colonnes d'une table à l'aide d'un identificateur de ligne spécifique (RID).

**Suggestions** :

- Modifiez les clés d'index pour y inclure les colonnes à extraire, de sorte que l'accès aux pages de données ne soit pas nécessaire.
- Identifiez l'index associé à l'opération d'extraction et cliquez deux fois sur le noeud correspondant pour en afficher les statistiques. Vérifiez que l'index présente un degré de correspondance élevé.
- Augmentez la taille de la mémoire tampon si les entrées-sorties (E-S) résultant de l'extraction sont supérieures au nombre de pages de la table.
- Si les statistiques ne sont pas à jour, réactualisez-les à l'aide de la commande RUNSTATS.

Les statistiques sur les quantiles et les valeurs les plus fréquentes fournissent des informations sur la sélectivité des prédicats, permettant ainsi de déterminer si un balayage d'index est préférable à un balayage de table. Pour mettre à jour ces valeurs, exécutez la commande **RUNSTATS** sur la table en utilisant la clause **WITH DISTRIBUTION**.

---

## FILTER

**Opérateur** : FILTER

**Fonction** : Application de prédicats résiduels permettant un filtrage des données suivant les critères définis par les prédicats.

**Suggestions** :

- Vérifiez que les prédicats utilisés entraînent l'extraction des seules données voulues. Vérifiez, par exemple, que leur sélectivité correspond bien à la partie de la table dont vous souhaitez disposer.
- Vérifiez que la classe d'optimisation est au moins égale à 3, de sorte que l'optimiseur utilise une jointure plutôt qu'une sous-requête. Si cela n'est pas possible, essayez de récrire la requête SQL pour supprimer la sous-requête.

Pour plus de détails, reportez-vous à la section "Query rewrites by the SQL compiler" dans le manuel *Administration Guide*.

---

## GENROW

**Nom** : GENROW

**Fonction** : Fonction intégrée générant une table de lignes, sans utiliser de données d'entrée issues de tables, d'index ou d'opérateurs.

La fonction GENROW peut être utilisée par l'optimiseur pour générer des lignes de données (par exemple, pour une instruction INSERT ou pour certains IN-listes convertis en jointures).

Pour visualiser les statistiques estimées sur les tables créées par la fonction GENROW, cliquez deux fois sur le noeud correspondant à cette dernière.

---

## GRPBY

**Opérateur** : GRPBY

**Fonction** : Regroupement des lignes présentant la même valeur pour des colonnes ou des fonctions spécifiées. Cette opération est nécessaire à la création d'un groupe de valeurs ou à l'évaluation d'une fonction ensembliste.

Même si aucune colonne GROUP BY n'est spécifiée, l'opérateur GRPBY peut être utilisé si la liste SELECT comporte des fonctions d'agrégation. Il permet d'indiquer que l'ensemble de la table doit être traité comme un seul groupe lors de l'agrégation.

**Suggestions** :

- Cet opérateur représente une opération obligatoire. Pour réduire les coûts d'un plan d'accès, attachez-vous plutôt à modifier les autres opérateurs (de balayage ou de jointure, par exemple) définissant l'ensemble de lignes à regrouper.
- Pour optimiser les résultats d'une instruction SELECT contenant une seule fonction d'agrégation mais pas de clause GROUP BY, procédez comme suit :
  - Dans le cas d'une fonction d'agrégation MIN(C), créez un index en ordre croissant sur C.
  - Dans le cas d'une fonction d'agrégation MAX(C), créez un index en ordre décroissant sur C.

---

## HSJOIN

**Opérateur** : HSJOIN

**Fonction** : Jointure par hachage dans laquelle un adressage calculé est appliqué aux lignes qualifiées dans les tables pour permettre une jointure directe, sans que le contenu des tables ait à être ordonné au préalable.

Une opération de jointure est nécessaire dès lors qu'une clause FROM fait référence à plus d'une table. Elle peut s'effectuer par hachage lorsqu'un prédicat de jointure établit une égalité entre les colonnes de deux tables. Les prédicats de jointure doivent être exactement du même type de données. La jointure par hachage peut également résulter d'une sous-requête réécrite, comme c'est le cas avec NLJOIN.

Pour effectuer ce type de jointure, il n'est pas nécessaire que les tables d'entrée soient triées. La jointure est réalisée en balayant la table interne et en générant une table de référence par l'affectation d'une adresse aux valeurs des colonnes de jointure. La jointure lit ensuite la table externe, en affectant une adresse aux valeurs des colonnes de jointure et en la comparant à celles de la table de référence créée pour la table interne.

Pour plus de détails, reportez-vous à la section sur les concepts de jointure dans le manuel *Administration Guide*.

**Suggestions** :

- Utilisez des prédicats locaux (faisant référence à une seule table) pour limiter le nombre de lignes à joindre.
- Augmentez la taille de la mémoire dynamique de tri afin qu'elle puisse contenir la table de référence obtenue par hachage.
- Si les statistiques ne sont pas à jour, réactualisez-les à l'aide de la commande RUNSTATS.

---

## INSERT

**Opérateur** : INSERT

**Fonction** : Insertion de lignes dans une table.

Cet opérateur représente une opération obligatoire. Pour réduire les coûts d'un plan d'accès, attachez-vous plutôt à modifier les autres opérateurs (de balayage ou de jointure, par exemple) définissant l'ensemble de lignes à insérer.

**Opérateur : IXAND**

**Fonction :** Association ("index ANDing") des résultats obtenus suite à plusieurs balayages d'index avec les techniques Dynamic Bitmap. L'opérateur permet d'appliquer à plusieurs index des prédicats ayant été associés (par AND) afin de réduire les accès aux tables sous-jacentes.

Cet opérateur permet d'effectuer les opérations suivantes :

- réduire le nombre de lignes avant d'accéder à la table de base ;
- associer (par AND) des prédicats s'appliquant à plusieurs index ;
- associer (par AND) les résultats de semi-jointures utilisés pour les jointures en étoile.

**Suggestions :**

- Les mises à jour de la base de données entraînent progressivement une fragmentation de l'index avec, pour conséquence, une augmentation inutile de son nombre de pages. Pour y remédier, supprimez puis recréez l'index ou réorganisez-le.
- Si les statistiques ne sont pas à jour, réactualisez-les à l'aide de la commande RUNSTATS.
- En règle générale, les résultats d'un balayage d'index sont d'autant plus satisfaisants que le nombre de lignes qualifiantes est faible. L'optimiseur utilise les statistiques disponibles sur les colonnes référencées dans les prédicats pour évaluer le nombre de lignes qualifiantes. Si certaines valeurs sont plus fréquentes que d'autres, il est important de disposer de statistiques de distribution. Pour cela, utilisez la commande **RUNSTATS** avec la clause WITH DISTRIBUTION. L'utilisation des statistiques de distribution non uniforme permet à l'optimiseur de distinguer les valeurs en fonction de leur fréquence d'apparition.
- IXAND est particulièrement efficace sur les index en une seule colonne dans la mesure où l'utilisation des clés de début et de fin n'est pas très adaptée avec cet opérateur.
- Pour les jointures en étoile, créez des index en une seule colonne pour chacune des colonnes les plus sélectives dans la table centrale et les tables de dimensions associées.

**Opérateur** : IXSCAN

**Fonction** : Balayage d'un index pour la création d'un ensemble réduit de lignes. Si vous le souhaitez, vous pouvez utiliser des conditions START ou STOP ou des prédicats indexables comportant des références à des colonnes de l'index.

Cette opération est destinée à réduire le nombre de lignes qualifiantes avant d'accéder à la table de base (à l'aide des prédicats).

Pour plus de détails, reportez-vous à la section relative au balayage d'index dans le manuel *Administration Guide*.

**Suggestions** :

- Les mises à jour de la base de données entraînent progressivement une fragmentation de l'index avec, pour conséquence, une augmentation inutile de son nombre de pages. Pour y remédier, supprimez puis recréez l'index ou réorganisez-le.
- Lorsque l'accès à plusieurs tables est nécessaire, il est préférable d'accéder à la table interne via un index car le système dispose ainsi d'un index sur la colonne de jointure de la table externe.  
Pour plus de détails sur les index, reportez-vous à l'aide en ligne relative à Visual Explain.
- Si les statistiques ne sont pas à jour, réactualisez-les à l'aide de la commande RUNSTATS.
- En règle générale, les résultats d'un balayage d'index sont d'autant plus satisfaisants que le nombre de lignes qualifiantes est faible. L'optimiseur utilise les statistiques disponibles sur les colonnes référencées dans les prédicats pour évaluer le nombre de lignes qualifiantes. Si certaines valeurs sont plus fréquentes que d'autres, il est important de disposer de statistiques de distribution. Pour cela, utilisez la commande **RUNSTATS** avec la clause WITH DISTRIBUTION. L'utilisation des statistiques de distribution non uniforme permet à l'optimiseur de distinguer les valeurs en fonction de leur fréquence d'apparition.

---

## MSJOIN

**Opérateur** : MSJOIN

**Fonction** : Jointure de tables par fusion, dans laquelle les lignes qualifiées des tables externe et interne doivent respecter l'ordre des prédicats de jointure.

Une opération de jointure est nécessaire dès lors qu'une clause FROM fait référence à plus d'une table. Elle peut s'effectuer par fusion lorsqu'un prédicat de jointure établit une égalité entre les colonnes de deux tables. La jointure par fusion peut également résulter d'une sous-requête réécrite.

La jointure par fusion nécessite que les données des colonnes de jointure soient ordonnées correctement car un seul balayage des tables est normalement effectué. Pour cela, le système accède à un index ou à une table triée.

Pour plus de détails, reportez-vous à la section sur les concepts de jointure dans le manuel *Administration Guide*.

**Suggestions** :

- Utilisez des prédicats locaux (faisant référence à une seule table) pour limiter le nombre de lignes à joindre.

Pour plus de détails sur les index, reportez-vous à la rubrique Création d'index dans l'aide en ligne relative à Visual Explain.

- Si les statistiques ne sont pas à jour, réactualisez-les à l'aide de la commande RUNSTATS.

---

## NLJOIN

**Opérateur** : NLJOIN

**Fonction** : Jointure par boucle imbriquée consistant à balayer (généralement par balayage d'index) la table interne pour chaque ligne de la table externe.

Une opération de jointure est nécessaire dès lors qu'une clause FROM fait référence à plus d'une table. Lorsque la jointure s'effectue par boucle imbriquée, l'utilisation d'un prédicat de jointure est facultative mais permet d'optimiser les performances.

Une jointure par boucle imbriquée s'effectue :

- soit par balayage de la table interne pour chaque ligne faisant l'objet d'un accès dans la table externe ;
- soit par consultation de l'index de la table interne pour chaque ligne faisant l'objet d'un accès dans la table externe.

Pour plus de détails, reportez-vous à la section sur les concepts de jointure dans le manuel *Administration Guide*.

**Suggestions :**

- La jointure par boucle imbriquée donne de meilleurs résultats s'il existe un index pour les colonnes de la table interne (affichée à droite de l'opérateur NLJOIN) sur lesquelles porte le prédicat de jointure. Vérifiez si la table interne est de type TBSCAN ou IXSCAN. S'il s'agit d'une table de type TBSCAN, il peut être préférable de créer un index sur les colonnes de jointure de cette table.

La création d'un index sur les colonnes de jointure de la table externe permet, dans une moindre mesure, d'obtenir de meilleurs résultats.

Pour plus de détails sur les index, reportez-vous à la rubrique Création d'index dans l'aide en ligne relative à Visual Explain.

- Si les statistiques ne sont pas à jour, réactualisez-les à l'aide de la commande RUNSTATS.

**Informations connexes :**

- Jointure en étoile.

---

## PIPE

**Opérateur :** PIPE

**Fonction :** Transfert de lignes vers d'autres opérateurs sans modification de celles-ci.

(Cet opérateur n'est utilisé qu'en mode débogage.)

---

## RETURN

**Opérateur :** RETURN

**Fonction :** Renvoi à l'utilisateur des données résultant d'une requête. Il s'agit du dernier opérateur d'un graphique de plan d'accès, qui indique les valeurs et les coûts totaux cumulés pour l'ensemble du plan.

Cet opérateur représente une opération obligatoire.

**Suggestion :**

- Vérifiez que les prédicats utilisés entraînent l'extraction des seules données voulues. Vérifiez, par exemple, que leur sélectivité correspond bien à la partie de la table dont vous souhaitez disposer.

---

## RIDSCN

**Opérateur** : RIDSCN

**Fonction** : Balayage d'une liste d'identificateurs de ligne (RID) obtenus à partir d'un ou plusieurs index.

Cet opérateur est pris en compte par l'optimiseur dans les cas suivants :

- Les prédicats sont reliés par le mot clé OR, ou un prédicat IN est utilisé. Une méthode appelée "index ORing" peut alors être utilisée. Elle consiste à effectuer plusieurs accès à un ou plusieurs index de la même table et à combiner les résultats de chaque accès.
- L'accès à un seul index est nécessaire mais celui-ci présente un faible degré de correspondance. Il est avantageux alors d'exécuter une lecture anticipée de liste (LIST PREFETCH) car le tri des identificateurs de ligne avant l'accès aux lignes de la base permet de réduire considérablement les opérations d'entrées-sorties.

---

## RQUERY

**Opérateur** : SHIP

**Fonction**: Extraction, dans un système de bases de données fédérées, de données à partir d'une source de données éloignée. Cet opérateur est pris en compte par l'optimiseur lorsqu'un opérateur SHIP envoie une instruction SQL SELECT à une source de données éloignée pour extraire le résultat de la requête. L'instruction SELECT est générée dans un dialecte SQL pris en charge par la source de données, et peut contenir toute requête valide en fonction de la source de données.

**Suggestion** : Reportez-vous à la section "Federated Database Query and Network Tuning Information" du chapitre 4 dans le manuel Administration Guide Vol 2.

---

## SORT

**Opérateur** : SORT

**Fonction** : Tri des lignes d'une table sur une ou plusieurs colonnes, avec suppression facultative des entrées en double.

Une opération de tri est nécessaire lorsqu'il n'existe pas d'index permettant de disposer de données ordonnées correctement, ou que le balayage de l'index est trop coûteux. Elle constitue souvent la dernière opération exécutée, une fois l'extraction des lignes requises terminée, ou un préalable à l'opération JOIN ou GROUP BY.

Si le nombre de lignes est élevé ou s'il est impossible d'établir un tube (rediriger vers un autre processus) pour les données triées, l'opération exige la création coûteuse de tables temporaires.

Pour plus de détails sur les tris, reportez-vous au manuel *Administration Guide*.

### **Suggestions :**

- Envisagez de créer un index sur les colonnes à trier.  
Pour plus de détails sur les index, reportez-vous à la rubrique Création d'index dans l'aide en ligne relative à Visual Explain.
- Vérifiez que les prédicats utilisés entraînent l'extraction des seules données voulues. Vérifiez, par exemple, que leur sélectivité correspond bien à la partie de la table dont vous souhaitez disposer.
- Vérifiez que la valeur du paramètre PREFETCHSIZE de l'espace table temporaire du système est suffisante. Pour cela, sélectionnez **Instruction->Affichage des statistiques->Espaces table**.
- Si des opérations de tri conséquentes sont fréquemment nécessaires, il est recommandé d'augmenter la valeur des paramètres de configuration suivants :
  - Taille de la mémoire dynamique de tri (sortheap). Pour modifier ce paramètre, sélectionnez la base de données dans la fenêtre Centre de contrôle et cliquez sur *Configuration* dans le menu en incrustation correspondant. Sélectionnez l'onglet **Performances** dans le bloc-notes qui s'affiche.
  - Seuil du segment de mémoire dynamique de tri (sheapthres). Pour modifier ce paramètre, sélectionnez l'instance de base de données dans la fenêtre Centre de contrôle et cliquez sur *Configuration* dans le menu en incrustation correspondant. Sélectionnez l'onglet **Performances** dans le bloc-notes qui s'affiche.
- Si les statistiques ne sont pas à jour, réactualisez-les à l'aide de la commande RUNSTATS.

---

## **TBSCAN**

**Opérateur :** TBSCAN

**Fonction :** Balayage d'une table permettant l'extraction de lignes par lecture directe de l'ensemble des données requises dans les pages de données.

L'optimiseur choisit ce type de balayage plutôt qu'un balayage d'index dans les cas suivants :

- La plage de valeurs recherchées apparaît fréquemment (un accès est donc nécessaire à une grande partie de la table).

- Le volume de la table est réduit.
- Le degré de correspondance de l'index est faible.
- Il n'existe pas d'index.

Pour plus de détails sur le balayage de table et d'index, reportez-vous au manuel *Administration Guide*.

### Suggestions :

- Il est préférable d'effectuer un balayage d'index si la table est volumineuse et qu'il n'est pas nécessaire d'accéder à la plupart des lignes de la table. Pour favoriser l'utilisation de l'index par l'optimiseur, envisagez de créer des index sur les colonnes pour lesquelles il existe des prédicats sélectifs. Pour plus de détails sur les index, reportez-vous à la rubrique Création d'index dans l'aide en ligne relative à Visual Explain.
- S'il existe un index mais que celui-ci n'est pas utilisé, assurez-vous que des prédicats sélectifs sont définis pour chacune des premières colonnes de cet index. Si ces prédicats existent, vérifiez alors que le degré de correspondance de l'index par rapport à la table est élevé. Pour cela, affichez la fenêtre Statistiques sur table correspondant à la table sur laquelle porte le tri et cliquez sur le bouton de fonction *Index* pour accéder à la fenêtre Statistiques sur index.
- Vérifiez que la valeur du paramètre PREFETCHSIZE (nombre de pages en lecture anticipée) définie pour l'espace table est suffisante. Pour cela, sélectionnez **Instruction->Affichage des statistiques->>Espaces table**. Pour plus de détails, reportez-vous à la section relative à la lecture anticipée des données dans la mémoire tampon, dans le manuel *Administration Guide*.
- Si les statistiques ne sont pas à jour, réactualisez-les à l'aide de la commande RUNSTATS.  
Les statistiques sur les quantiles et les valeurs les plus fréquentes fournissent des informations sur la sélectivité des prédicats, permettant ainsi de déterminer si un balayage d'index est préférable à un balayage de table. Pour mettre à jour ces valeurs, exécutez la commande **RUNSTATS** sur la table en utilisant la clause WITH DISTRIBUTION.

---

## TEMP

### Opérateur : TEMP

**Fonction** : Stockage des données dans une table temporaire qui sera lue par un autre opérateur (à plusieurs reprises, si nécessaire). La table est supprimée une fois le traitement de l'instruction SQL terminé, voire même avant la fin de son exécution.

Cet opérateur est nécessaire à l'évaluation des sous-requêtes ou au stockage de résultats intermédiaires. Dans certains cas, par exemple celui d'une instruction pouvant être mise à jour, il devient obligatoire.

---

## TQUEUE

**Opérateur** : TQUEUE

**Fonction** : File d'attente de tables utilisée pour permettre l'échange de données de table entre des agents de base de données lorsque plusieurs agents traitent une requête. Plusieurs agents de base de données sont utilisés pour traiter une requête lorsque la fonction de parallélisme est mise en oeuvre.

Les types de file d'attente de tables suivants sont utilisés :

- **Locale** : La file d'attente de tables est utilisée pour permettre l'échange de données entre des agents de base de données sur un même noeud. Une file d'attente de tables locale est utilisée pour le parallélisme intra-partition.
- **Non locale** : La file d'attente de tables est utilisée pour permettre l'échange de données entre des agents de base de données sur des noeuds différents.

---

## UNION

**Opérateur** : UNION

**Fonction** : Concaténation d'ensembles de lignes issues de plusieurs tables.

Cet opérateur représente une opération obligatoire. Pour réduire les coûts d'un plan d'accès, attachez-vous plutôt à modifier les autres opérateurs (de balayage ou de jointure, par exemple) définissant l'ensemble de lignes à concaténer.

---

## UNIQUE

**Opérateur** : UNIQUE

**Fonction** : Suppression des lignes comportant des doublons dans les colonnes spécifiées.

**Suggestion** :

- Cet opérateur peut être évité uniquement s'il existe un index à entrées uniques sur les colonnes appropriées.

Pour plus de détails sur les index, reportez-vous à la rubrique Création d'index dans l'aide en ligne relative à Visual Explain.

---

## UPDATE

**Opérateur** : UPDATE

**Fonction** : Mise à jour des lignes d'une table.

Cet opérateur représente une opération obligatoire. Pour réduire les coûts d'un plan d'accès, attachez-vous plutôt à modifier les autres opérateurs (de balayage ou de jointure, par exemple) définissant l'ensemble de lignes à mettre à jour.



---

## Annexe C. DB2 concepts

---

### Bases de données

Une base de données relationnelle présente les données sous la forme d'un ensemble de tables. Une table est un ensemble de colonnes et d'un nombre quelconque de lignes. Les données dans chaque table sont associées de façon logique et des relations peuvent être définies entre les tables. Les données peuvent être affichées et manipulées sur la base de principes et d'opérations mathématiques appelés relations (telles que, INSERT, SELECT et UPDATE).

Une base de données est une structure autodescriptive car elle contient, outre des données, une description de sa propre structure. Elle inclut un ensemble de tables système (également appelées tables du catalogue système) qui décrivent la structure logique et physique des données, un fichier de configuration qui regroupe les valeurs des paramètres associées à la base de données, et un journal de récupération dans lequel sont consignées les transactions en cours ainsi que les transactions pouvant être archivées.

Les bases de données peuvent être locales ou éloignées. Une base de données locale se trouve physiquement sur le poste de travail en cours d'utilisation, tandis qu'une base de données sur une autre machine est considérée éloignée.

---

### Schémas

Un schéma est identificateur unique servant à regrouper un ensemble d'objets de base de données (tels que les tables, les vues, les index et les alias). Par exemple, si vous vouliez créer une table nommée PAYROLL, il serait fastidieux de devoir chercher dans la base de données si un autre utilisateur n'a pas déjà créé une table avec le même nom. Le nom de l'objet doit être unique seulement dans le schéma.

Le nom de la plupart des objets de base de données comporte deux parties, la première étant le nom du schéma et la seconde le nom de l'objet. Lorsqu'un objet est créé, vous pouvez l'assigner à un schéma spécifique. Si vous ne spécifiez aucun schéma, l'objet est assigné au schéma par défaut, qui est généralement l'ID utilisateur de l'auteur de l'objet. Par exemple, un utilisateur appelé Smith peut posséder une table nommée SMITH.PAYROLL.

Le schéma devient également un objet dans la base de données. Il est créé lorsque le premier objet dans le schéma est créé. Un schéma peut appartenir à une personne et son propriétaire peut contrôler l'accès aux données et aux objets qui y figurent.

---

## Tables

Une base de données relationnelle présente les données sous la forme d'un ensemble de tables. Une table est constituée de données organisées logiquement dans des colonnes et des lignes (généralement appelées enregistrements).

Chaque table a un nom, et dans une table, chaque colonne a un nom. Aucun ordre de tri particulier n'est appliqué entre les lignes d'une table, mais celles-ci peuvent être extraites dans un ordre déterminé d'après les valeurs dans les colonnes. Les données dans une table sont associées logiquement. Toutes les données de bases de données et de tables sont assignées à des espaces de tables.

---

## Annexe D. Remarques

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non disponibles dans tous les pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM EMEA Director of Licensing  
IBM Europe Middle-East Africa  
Tour Descartes  
La Défense 5  
2, avenue Gambetta  
92066 - Paris-La Défense CEDEX  
France

Pour le Canada, veuillez adresser votre courrier à :

IBM Director of Commercial Relations  
IBM Canada Ltd.  
3600 Steeles Avenue East  
Markham, Ontario  
L3R 9Z7  
Canada

Les informations sur les licences concernant les produits utilisant un jeu de caractères à deux octets (DBCS) peuvent être obtenues par écrit à l'adresse suivante :

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales.** LE PRESENT DOCUMENT EST LIVRE «EN L'ETAT». IBM DECLINE TOUTE RESPONSABILITE, EXPRESSE OU IMPLICITE, RELATIVE AUX INFORMATIONS QUI Y SONT CONTENUES, Y COMPRIS EN CE QUI CONCERNE LES GARANTIES DE QUALITE MARCHANDE OU D'ADAPTATION A VOS BESOINS. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Il est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut modifier sans préavis les produits et logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les informations contenues dans ces sites Web ne sont pas associées à ce produit IBM, et l'utilisation de ces sites se fait à vos propres risques et périls.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Canada Limited  
Office of the Lab Director  
8200 Warden Avenue  
Markham, Ontario  
L6G 1C7  
CANADA

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans ce document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux dispositions de l'ICA, des Conditions internationales d'utilisation des logiciels IBM ou de tout autre accord équivalent.

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier

de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

#### LICENCE DE COPYRIGHT :

Ce document peut contenir des exemples de programmes, en langage source, qui permettent d'illustrer des techniques de programmation sur diverses plateformes. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes.

Toute copie totale ou partielle de ces programmes exemples et des oeuvres qui en sont dérivées doit comprendre une notice de copyright, libellée comme suit :

© (le nom de votre société) (année). Des segments de code sont dérivés des Programmes exemples d'IBM Corp. © Copyright IBM Corp. *\_entrez l'année ou les années\_*. All rights reserved.

---

## Marques

Les termes qui suivent sont des marques d'International Business Machines Corporation dans certains pays :

ACF/VTAM	LAN Distance
AISPO	MVS
AIX	MVS/ESA
AIXwindows	MVS/XA
AnyNet	Net.Data
APPN	NetView
AS/400	OS/390
BookManager	OS/400
C Set++	PowerPC
C/370	pSeries
CICS	QBIC
Database 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/400
DB2 Extenders	SQL/DS
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational Database Architecture	SystemView
DRDA	Tivoli
eServer	VisualAge
Extended Services	VM/ESA
FFST	VSE/ESA
First Failure Support Technology	VTAM
IBM	WebExplorer
IMS	WebSphere
IMS/ESA	WIN-OS/2
iSeries	z/OS
	zSeries

Les termes qui suivent sont des marques d'autres sociétés et ont été utilisées dans au moins un des documents de la bibliothèque de documentation DB2 UDB :

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation dans certains pays.

Intel et Pentium sont des marques d'Intel Corporation aux Etats-Unis et/ou dans d'autres pays.

Java et toutes les marques incluant Java sont des marques de Sun Microsystems, Inc. dans certains pays.

UNIX est une marque enregistrée de The Open Group aux Etats-Unis et/ou dans certains autres pays.

D'autres sociétés sont propriétaires des autres marques, noms de produits ou logos qui pourraient apparaître dans ce document.



---

# Index

## C

classes d'optimisation de requêtes  
  définition 72  
collecte des statistiques à jour pour  
  les tables et les index 19, 38  
colonnes dans les instructions SQL,  
  obtention de statistiques 11  
commande VESAMPL.DDL 2  
commandes, EXPLAIN.DDL 1  
commandes, LIST TABLES 1  
commandes, option EXPLSNAP sur  
  BIND 5  
commandes, SET CURRENT  
  EXPLAIN SNAPSHOT 4  
commandes, VESAMPL.DDL 2  
conteneurs  
  définition 53  
coût  
  définition 54  
création d'index supplémentaires sur  
  les colonnes de table 29, 47  
création d'index sur les colonnes  
  utilisées pour joindre des tables  
  dans une requête 24, 42

## E

espaces table  
  définition 76  
  DMS  
    définition 55  
espaces table, obtention des  
  statistiques 11  
espaces table DMS  
  définition 55  
espaces table SMS  
  définition 75  
exécution d'une requête sans index  
  ni statistiques 34  
EXPLAIN.DDL  
  (fichier/commande) 1

## F

fichiers, EXPLAIN.DDL 1  
fonction GENROW  
  définition 61, 81  
fonctions, obtention de  
  statistiques 11  
fonctions, obtention des  
  statistiques 11

fonctions définies par l'utilisateur,  
  obtention de statistiques 11  
fonctions intégrées, obtention de  
  statistiques 11

## G

graphique de plan d'accès, affichage  
  et utilisation 7  
graphique de plan d'accès, détails  
  des objets 9  
graphique de plan d'accès, lecture  
  des symboles 8  
graphique de plan d'accès,  
  modification de l'apparence 12  
graphiques de plan d'accès  
  création  
    définition 52  
  liste des opérateurs utilisés 57  
  noeuds  
    définition 53  
groupe de lignes  
  définition 54  
  groupe de lignes 54

## I

image instantanée EXPLAIN  
  définition 56  
Images instantanées, échantillons  
  pour Visual Explain 2  
Images instantanées EXPLAIN,  
  création 1  
Images instantanées EXPLAIN,  
  échantillons pour Visual  
  Explain 2  
images instantanées EXPLAIN pour  
  des instructions SQL dynamiques,  
  création 4  
images instantanées EXPLAIN pour  
  des instructions SQL statiques,  
  création 5  
index  
  classification  
    définition 53  
instructions explicites  
  définition 56  
instructions explicitées  
  définition 57  
instructions SQL dynamiques,  
  création d'images instantanées  
  EXPLAIN 4

instructions SQL explicitées,  
  sélection 7  
instructions SQL statiques, création  
  d'images instantanées  
  EXPLAIN 5

## J

Jointure en étoile, définition  
  définition 74

## L

LIST TABLES (commande) 1

## M

modules  
  définition 71

## O

opérandes  
  définition 57  
opérateur CMPEXP  
  définition 59, 79  
opérateur DELETE  
  définition 59, 79  
opérateur EISCAN  
  définition 59, 79  
opérateur FETCH  
  définition 60, 80  
opérateur FILTER  
  définition 60, 80  
opérateur GRPBY  
  définition 61, 81  
opérateur HSJOIN  
  définition 62, 82  
opérateur INSERT  
  définition 62, 82  
opérateur IXAND  
  définition 63, 83  
opérateur IXSCAN  
  définition 64, 84  
opérateur MSJOIN  
  définition 65, 85  
opérateur NLJOIN  
  définition 65, 85  
opérateur PIPE  
  définition 66, 86  
opérateur RETURN  
  définition 66, 86  
opérateur RIDSCN  
  définition 67, 87

- opérateur SHIP
  - définition 67, 87
- opérateur SORT
  - définition 67, 87
- opérateur TBSCAN
  - définition 68, 88
- opérateur TEMP
  - définition 69, 89
- opérateur TQUEUE
  - définition 70, 90
- opérateur UNION
  - définition 70, 90
- opérateur UNIQUE
  - définition 70, 90
- opérateur UPDATE
  - définition 71, 91
- opérateurs
  - définition 57
  - liste 57
- opérateurs, obtention de détails 10
- optimiseur
  - définition 71
- option EXPLSNAP (sur la commande BIND) 5
- options de définition d'accès,
  - obtention d'informations 11

## P

- paramètres de configuration,
  - obtention d'informations 11
- plan d'accès, amélioration 13, 33
- plans d'accès
  - définition 51
- prédicats
  - définition 71

## R

- requête sans index ni statistiques 14

## S

- Sélectivité des prédicats, définition
  - définition 73
- SET CURRENT EXPLAIN
  - SNAPSHOT (commande) 4
- SQL dynamique
  - définition 55
- SQL statique
  - définition 75
- statistiques, pour les tables, index,
  - fonctions de table 10

## T

- tables EXPLAIN, création 1

## V

- Visual Explain
  - description 76

## Z

- zoom, pour augmenter la taille des graphiques de plan d'accès 8

---

## Contacter IBM

Aux États-Unis, contactez IBM en composant l'un des numéros suivants :

- 1-800-237-5511 pour obtenir le Service clients,
- 1-888-426-4343 pour connaître les options de service disponibles.
- 1-800-IBM-4YOU (426-4968) pour obtenir le Service Ventes et marketing DB2

Au Canada, contactez IBM en composant l'un des numéros suivants :

- 1-800-IBM-SERV (1-800-426-7378) pour obtenir le Service clients
- 1-800-465-9600 pour connaître les options de service disponibles.
- 1-800-IBM-4YOU (1-800-426-4968) pour obtenir le Service Ventes et marketing DB2

Pour connaître les bureaux IBM de votre pays ou de votre région, consultez l'annuaire IBM de contacts internationaux à l'adresse suivante : [www.ibm.com/planetwide](http://www.ibm.com/planetwide)

---

## Infos produit

Des informations concernant les produits DB2 Universal Database sont disponibles par téléphone ou via le Web à l'adresse suivante : [www.ibm.com/software/data/db2/udb](http://www.ibm.com/software/data/db2/udb)

Ce site présente les dernières informations concernant la bibliothèque technique, les commandes de manuels, les téléchargement de clients, les forums, les FixPaks, les actualités ainsi que des liens vers des ressources Web.

Aux États-Unis, composez l'un des numéros suivants :

- 1-800-IBM-CALL (1-800-426-2255) pour commander des produits ou pour obtenir des informations d'ordre général.
- Pour commander des manuels, composez le 1-800-879-2755.

Pour plus d'informations sur la façon de contacter IBM en dehors des États-Unis, consultez le site Web suivant : [www.ibm.com/planetwide](http://www.ibm.com/planetwide)

**IBM**