Université IBM i

7 novembre 2023

IBM Innovation Studio Paris

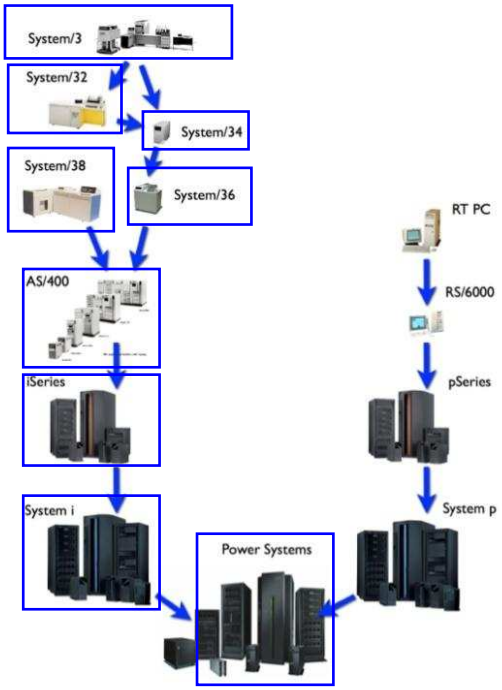
S21 – L'hyper optimisation sous Db2 for i

16:00 / 17:00

Christian GRIERE
Christian GRIERE Services
ex IBM France (1976-2014)
cgriere@gmail.com

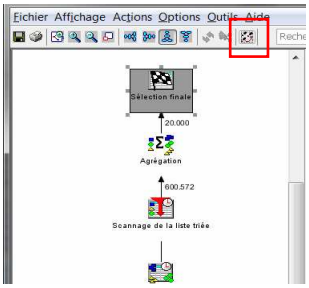
infrasdufutur

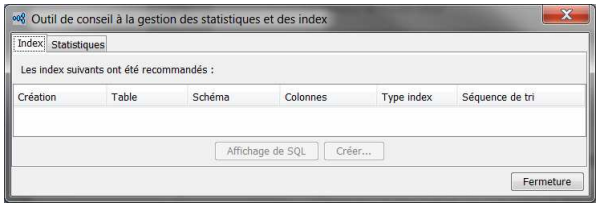
#ibmi
#uii2023
#infrastructuredufuturIBM23



De quoi vais-je vous parler ?

De ce que l'on peut faire pour améliorer le temps d'exécution d'une requête SQL lorsque le conseilleur d'index (Index Advisor) ne conseille plus rien.





Plan

- Niveaux d'optimisation
- Avertissements
- Rappels sur les index de l'IBM i
- Rappels sur quelques icônes de Visual Explain
- 4 icônes qui peuvent générer des IOs inutiles
- 3 techniques pour diminuer les IOs *utiles*
- Comment limiter l'agrégation et le scannage des index binaires ?
- Conclusion/Questions/Réponses

Niveaux d'optimisation

- Le niveau d'optimisation d'une requête SQL n'est pas binaire.
- Une requête SQL peut être ? optimisée ;

Pas du tout Un peu Moyennement

Extrêmement **Hyper**

- La disponibilité des différents niveaux dépend de la complexité de la requête SQL.

Avertissements

- Il faut avoir des raisons valables pour « hyper » optimiser une requête SQL.
- Exemples de raison valable :
  - requête qui appartient à une transaction critique en temps de réponse.
  - requête qui est exécutée de façon très répétitive : 1 exécution par seconde = 36 000 exécutions par jour sur une base de 10 h/j
  - requête extrêmement longue.
- Il ne faut pas tomber dans le perfectionnisme.

Rappel sur les types d'index de l'IBM i

Combien de type ?

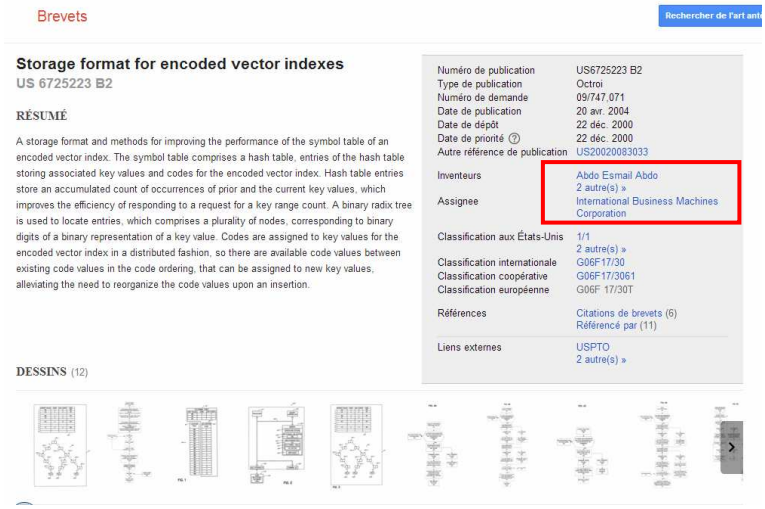
1 ? 2 ? 3 4 ?

- Index binaire
- Index texte (utilisé par OmniFind Text Search Server)
- Index à vecteurs encodés (EVI)

Rappel sur les index binaires

Clé	RRN
…	
DUPOND	300
DUPONT	45
DURAND	1435
…	

Rappel sur les EVI – Brevet IBM



Brevets

Rechercher de l'art ant

Storage format for encoded vector indexes

US 6725223 B2

RÉSUMÉ

A storage format and methods for improving the performance of the symbol table of an encoded vector index. The symbol table comprises a hash table, entries of the hash table storing associated key values and codes for the encoded vector index. Hash table entries store an accumulated count of occurrences of prior and the current key values, which improves the efficiency of responding to a request for a key range count. A binary radix tree is used to locate entries, which comprises a plurality of nodes, corresponding to binary digits of a binary representation of a key value. Codes are assigned to key values for the encoded vector index in a distributed fashion, so there are available code values between existing code values in the code ordering, that can be assigned to new key values, alleviating the need to reorganize the code values upon an insertion.

Numéro de publication	US6725223 B2
Type de publication	Octroi
Numéro de demande	09/747,071
Date de publication	20 avr. 2004
Date de dépôt	22 déc. 2000
Date de priorité	22 déc. 2000
Autre référence de publication	US20020083033
Inventeurs	Abdo Esmail Abdo 2 autre(s) »
Assignee	International Business Machines Corporation
Classification aux États-Unis	1/1 2 autre(s) »
Classification internationale	G06F17/30
Classification coopérative	G06F17/3061
Classification européenne	G06F 17/30T
Références	Citations de brevets (6) Référencé par (11)
Liens externes	USPTO 2 autre(s) »

DESSINS (12)

Rappel sur les EVI – Table des symboles

create encoded vector index clients_evi_ville on clients(**ville**) ;

Affectation d'un **code** à chaque valeur distincte de la colonne

ville	code
LILLE	01
MONTPELLIER	02
PARIS	03

Structure de base de la table des symboles

Rappel sur les EVI - Table des vecteurs

create encoded vector index clients_evi_ville on clients(ville) ;

Nom client	ville
DUPONT	LILLE
DURAND	PARIS
MARTIN	MONTPELLIER
UNTEL	LILLE

Table clients

code
1
2
3
1

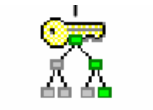
RRN
1
2
3
4

Table des vecteurs

ville	code	Count(*)	1er RRN	Dernier RRN
LILLE	1	2	1	4
MONTPELLIER	3	1	2	2
PARIS	2	1	3	3

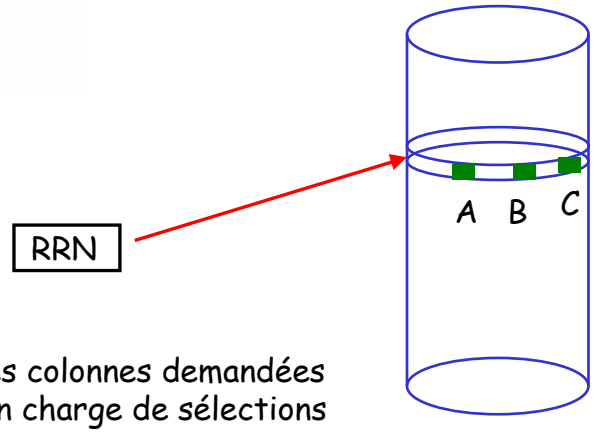
Table des symboles

Rappel sur l'icône Test d'index (binaire)



Clé	RRN
…	
DUPOND	300
DURAND	45
MARTIN	1435
…	

Rappel sur l'icône Test de table

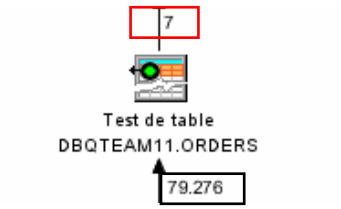


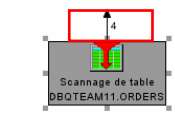
2 fonctions :
- Récupère les colonnes demandées
- Peut être en charge de sélections

Eliminer les IOs inutiles

- 4 icônes à surveiller

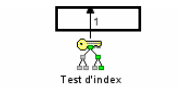
4 icônes pouvant générer des IOs inutiles



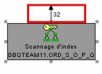


Infos table supplémentaires	
Nombre total de lignes dans la table	600.572
Taille table (octets)	1,609E8
Lignes de la table active	600.572
Lignes table supprimées	0

4 icônes pouvant générer des IOs inutiles



Horodatage pour la création de poste de moniteur	2020-01-29
Horodatage de début d'instruction	2020-01-29
Horodatage de fin d'instruction	2020-01-29
Durée totale d'exécution estimée (ms)	246,996

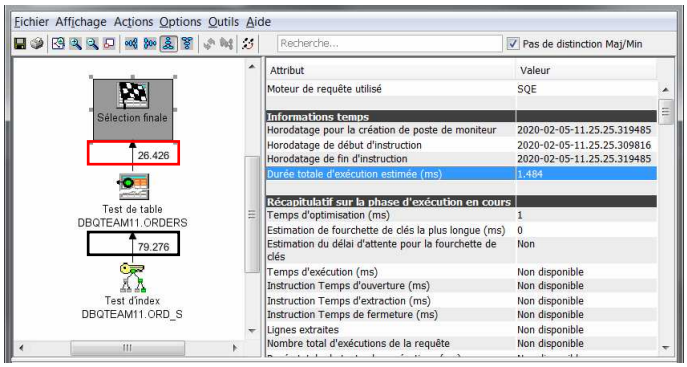


Infos index supplémentaires	
Nombre d'entrées d'index	600,572
Taille de zone clé	31
Taille de page logique de l'index	65,536
Longueur de clé variable	Non
Remplissage de toutes les zones	Non
Unique	Non
Contient une sélection fractionnée	Non
Les clés indéfinies sont fausses	Oui
Type d'index	Radix
Liste des clés de l'index	Ascending, ORDERS_1.notused, Ascending, ORDERS_1.ORDERKEY, Ascending, ORDERS_1.PARTKEY, Ascending, ORDERS_1.QUANTITY

1 – Exemple pour Test de table

select orderkey, partkey, quantity*extendedprice from orders where shipmode = 'AIR' and quantity*extendedprice > 4000000 optimize for all rows ;

Avec l'index conseillé : BIN : orders (shipmode)

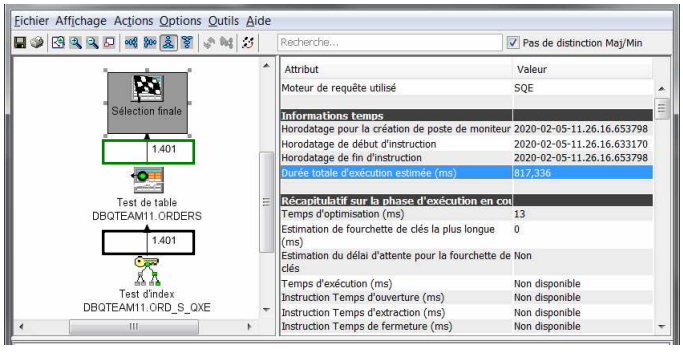


1 – Exemple pour Test de table

select orderkey, partkey, quantity*extendedprice from orders where shipmode = 'AIR' and quantity*extendedprice > 4000000 optimize for all rows ;

Solution : créer un FBI (Function Based Index)

create index ord_s_qxe on orders (shipmode, quantity*extendedprice) ;



2 – Exemple pour Scannage de table

select * from parts where upper(part) like 'STEEL%'
optimize for all rows ;

Avec aucun index conseillé :

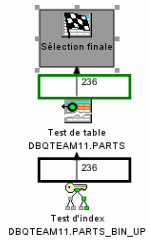
Nom de table, nom de table de base	
Nom de la table interrogée	PARTS
Bibliothèque de la table interrogée	DBQTEAM11
Membre de la table interrogée	PARTS
Nom long de la table interrogée	PARTS
Nom long de la bibliothèque de la table interrogée	DBQTEAM11
Informations durée estimée (heure	
Temps de traitement (ms)	5,659
Temps cumulé (ms)	0
Infos table supplémentaires	
Nombre total de lignes dans la table	20.000
Taille table (octets)	2.860.064

2 – Exemple pour Scannage de table

select * from parts where upper(part) like 'STEEL%'
optimize for all rows ;

Solution : créer un FBI

create index parts_bin_up on parts (upper(part)) ;

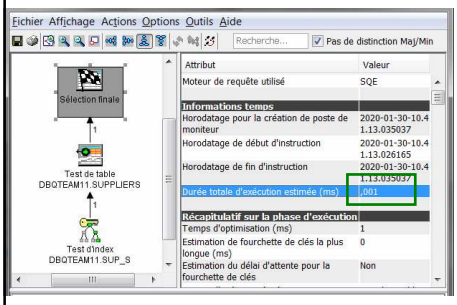


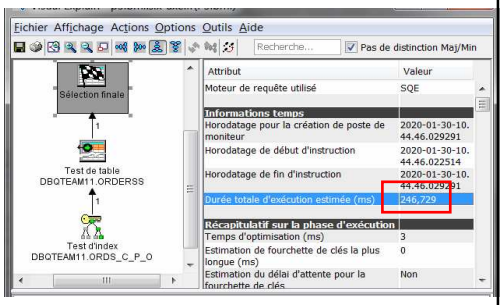
Informations temps	
Horodatage pour la création de poste de moniteur	2023-10-20-17
Horodatage de début d'instruction	2023-10-20-17
Horodatage de fin d'instruction	2023-10-20-17
Durée totale d'exécution estimée (ms)	,185
Récapitulatif sur la phase d'exécutio	
Temps d'optimisation (ms)	12
Estimation de fourchette de clés la plus longue (ms)	0
Estimation du délai d'attente pour la fourchette de clés	Non
Temps d'exécution (ms)	Non disponible
Instruction Temps d'ouverture (ms)	Non disponible
Instruction Temps d'extraction (ms)	Non disponible
Instruction Temps de fermeture (ms)	Non disponible
Lignes extraites	Non disponible

3 – Exemple pour Test d'index

select * from suppliers
where suppkey=123 ;

select * from orderss where
codsoc='001'and orderkey=575682 ;





3 – Exemple pour Test d'index

select * from suppliers
where suppkey=123 ;

Avec l'index :
BIN : suppliers (suppkey)

select * from orderss where
codsoc='001' and orderkey=575682 ;

Avec l'index :
BIN : orderss (codsoc, partkey, orderkey)

Test d'index

Nom de l'index utilisé	SUP_S
Bibliothèque de l'index utilisé	DBQTEAM11
Nom de la table interrogée	SUPPLIERS
Bibliothèque de la table interrogée	DBQTEAM11
Temps cumulé (ms)	0
Coût UC (ms)	8,305E-4
Coût E-S (ms)	0
Nombre E-S	1

Test d'index

Nom de l'index utilisé	ORDS_C_P_O
Bibliothèque de l'index utilisé	DBQTEAM11
Nom de la table interrogée	ORDERSS
Bibliothèque de la table interrogée	DBQTEAM11
Temps cumulé (ms)	0
Coût UC (ms)	244,865
Coût E-S (ms)	76,916
Nombre E-S	156

3 – Exemple pour Test d'index

select * from orderss where codsoc='001' and orderkey=575682 ;

Avec son index actuel :

BIN : orderss (codsoc, partkey, orderkey)

Test d'index

Nom de l'index utilisé	ORDS_C_P_O
Bibliothèque de l'index utilisé	DBQTEAM11
Nom de la table interrogée	ORDERSS
Bibliothèque de la table interrogée	DBQTEAM11
Temps cumulé (ms)	0
Coût UC (ms)	244,865
Coût E-S (ms)	76,816
Nombre E-S	156

Avec son nouvel index :

BIN : orderss (codsoc, orderkey)

Test d'index

Nom de l'index utilisé	ORDS_C_O
Bibliothèque de l'index utilisé	DBQTEAM11
Nom de la table interrogée	ORDERSS
Bibliothèque de la table interrogée	DBQTEAM11
Temps cumulé (ms)	0
Coût UC (ms)	,002
Coût E-S (ms)	2,456
Nombre E-S	1

3 techniques pour diminuer les IOs *utiles*

3 techniques pour diminuer les IOs *utiles*

- Utiliser des index binaires « Only Access »
  - pour les colonnes de la requête
  - pour les colonnes des fichiers joints de la requête

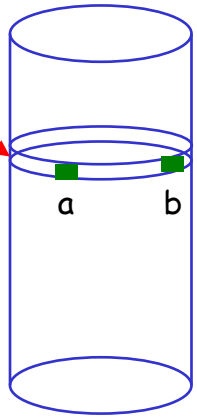
- Utiliser des EVI « Only Access »
  - pour les colonnes de la requête (mode inversé)
  - pour les colonnes des fichiers joints de la requête

- Utiliser des FBI « Only Access »

1 - Principe d'un index binaire « Only Access »

select a, b from t where c='DUPONT'

Clé/c	RRN	a	b
…		…	…
DUPOND	300	59000	LILLE
DUPONT	45	75016	PARIS
DURAND	1435	29000	BREST
…		…	…

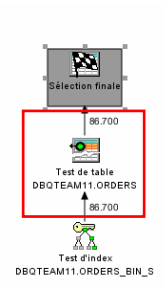


1 - Utiliser des index binaires « Only Access »

… pour récupérer les colonnes de la requête

```
select orderkey, partkey, quantity from orders
where shipmode ='AIR' optimize for all rows ;
```

Avec l'index conseillé : BIN : orders (shipmode)

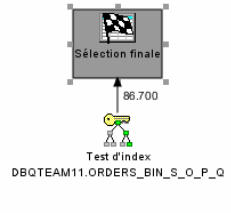


Moteur de requête utilisé	SQE
Informations temps	
Horodatage pour la création de poste de moniteur	2023-10-19-19
Horodatage de début d'instruction	2023-10-19-19
Horodatage de fin d'instruction	2023-10-19-19
Durée totale d'exécution estimée (ms)	1,460
Récapitulatif sur la phase d'exécution en	
Temps d'optimisation (ms)	9
Estimation de fourchette de clés la plus longue (ms)	0
Estimation du délai d'attente pour la fourchette de clés	Non
Temps d'exécution (ms)	Non disponible
Instruction Temps d'ouverture (ms)	Non disponible

1 - Utiliser des index binaires « Only Access »

```
select orderkey, partkey, quantity from orders
where shipmode ='AIR' optimize for all rows ;
```

Avec l'index binaire « Only Access » :
orders (shipmode, orderkey, partkey, quantity)



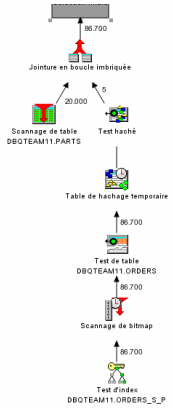
Moteur de requête utilisé	SQE
Informations temps	
Horodatage pour la création de poste de moniteur	2023-10-19-19.
Horodatage de début d'instruction	2023-10-19-19.
Horodatage de fin d'instruction	2023-10-19-19.
Durée totale d'exécution estimée (ms)	140,224
Récapitulatif sur la phase d'exécution en c	
Temps d'optimisation (ms)	1
Estimation de fourchette de clés la plus longue (ms)	0
Estimation du délai d'attente pour la fourchette de clés	Non

1 - Utiliser des index binaires « Only Access »

… pour récupérer les colonnes des fichiers joints

```
select a.orderkey, a.partkey, a.quantity, b.part from orders a,
parts b where a.partkey=b.partkey and a.shipmode='AIR'
optimize for all rows ;
```

Avec les index conseillés :

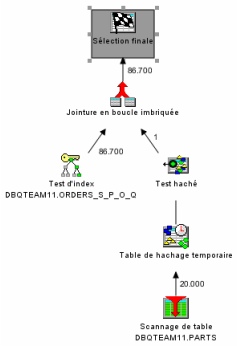


Attribut	Valeur
Moteur de requête utilisé	SQE
Informations temps	
Horodatage pour la création de poste de moniteur	2023-10-19-15.
Horodatage de début d'instruction	2023-10-19-15.
Horodatage de fin d'instruction	2023-10-19-15.
Durée totale d'exécution estimée (ms)	1,602
Récapitulatif sur la phase d'exécution	
Temps d'optimisation (ms)	12
Estimation de fourchette de clés la plus longue (ms)	0
Estimation du délai d'attente pour la fourchette de clés	Non
Temps d'exécution (ms)	Non disponible
Instruction Temps d'ouverture (ms)	Non disponible
Instruction Temps d'extraction (ms)	Non disponible
Instruction Temps de fermeture (ms)	Non disponible
Lignes extraites	Non disponible
Nombre total d'exécutions de la requête	Non disponible
Durée totale de toutes les exécutions (ms)	Non disponible
Temps UC	,999
Lectures synchrones base de données	Non disponible
Lectures asynchrones base de données	Non disponible
Erreurs de page	Non disponible
Stockage temporaire utilisé (Mo)	Non disponible
Nombre de réutilisations des résultats temporaires	Non disponible
Résultat de variable temporaire de raison	Non disponible

1 - Utiliser des index binaires « Only Access »

```
select a.orderkey, a.partkey, a.quantity, b.part from orders a,
parts b where a.partkey=b.partkey and a.shipmode='AIR'
optimize for all rows ;
```

Avec l'un des index binaires « Only Access » :
orders (shipmode, partkey, orderkey, quantity)



Informations temps	
Horodatage pour la création de poste de moniteur	2023-10-19-15.
Horodatage de début d'instruction	2023-10-19-15.
Horodatage de fin d'instruction	2023-10-19-15.
Durée totale d'exécution estimée (ms)	134,963
Récapitulatif sur la phase d'exécution	
Temps d'optimisation (ms)	7
Estimation de fourchette de clés la plus longue (ms)	0
Estimation du délai d'attente pour la fourchette de clés	Non
Temps d'exécution (ms)	Non disponible
Instruction Temps d'ouverture (ms)	Non disponible
Instruction Temps d'extraction (ms)	Non disponible
Instruction Temps de fermeture (ms)	Non disponible
Lignes extraites	Non disponible
Nombre total d'exécutions de la requête	Non disponible
Durée totale de toutes les exécutions (ms)	Non disponible
Temps UC	,999
Lectures synchrones base de données	Non disponible
Lectures asynchrones base de données	Non disponible
Erreurs de page	Non disponible

Redondance d'index binaires

Index de base I1 : parts (partkey)

Index Only Access I2 : parts (partkey, part)

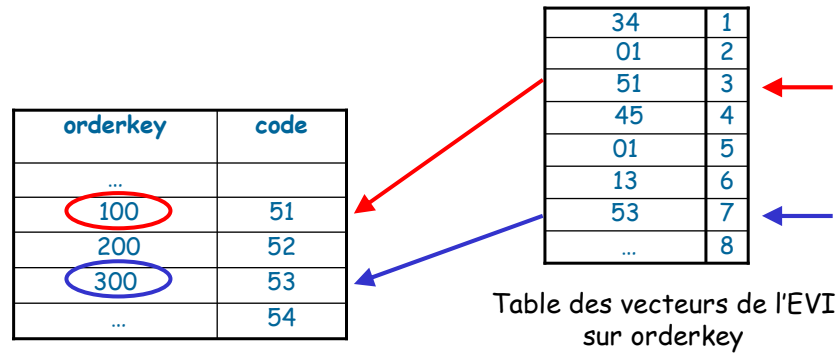
- 2 chemins d'accès à maintenir quel que soit l'ordre de création.
- I2 peut remplacer I1.
- I1 ne peut pas remplacer I2. I1 peut être supprimé.
- Voir article détaillé sur Linkedin :

https://www.linkedin.com/pulse/avez-vous-des-index-redondants-christian-griere/

2 - Utiliser des EVI « Only Access »

… pour récupérer les colonnes d'une requête

```
select orderkey, partkey, quantity from orders
where shipmode='AIR' ;
```



orderkey	code
…	
100	51
200	52
300	53
…	54

Table des symboles de l'EVI sur orderkey

34	1
01	2
51	3
45	4
01	5
13	6
53	7
…	8

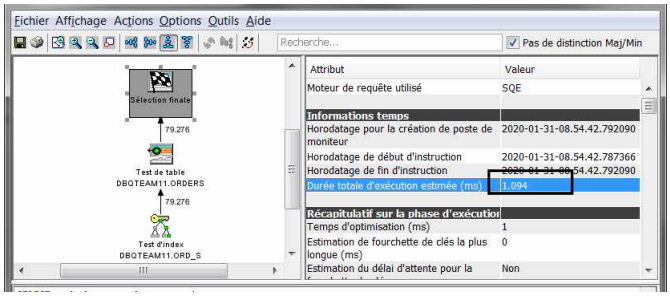
Table des vecteurs de l'EVI sur orderkey

2 - Utiliser des EVI « Only Access »

… pour récupérer les colonnes de la requête

```
select orderkey, partkey, quantity from orders
where shipmode ='AIR' optimize for all rows ;
```

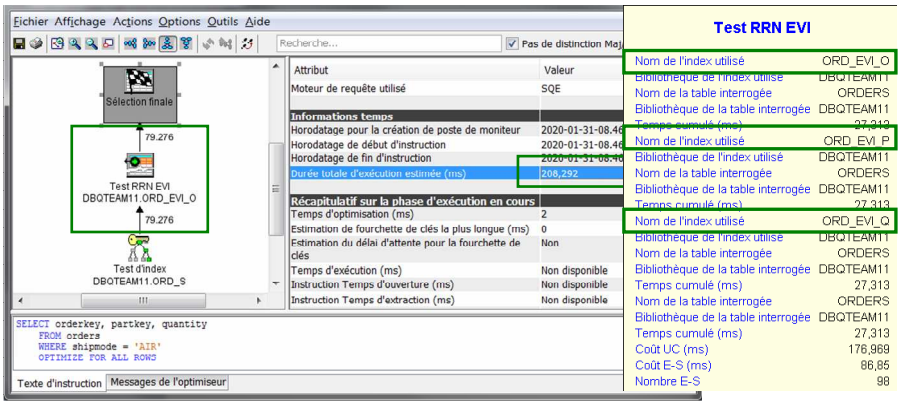
Avec l'index conseillé : BIN : orders (shipmode)



2 - Utiliser des EVI « Only Access »

```
select orderkey, partkey, quantity from orders
where shipmode ='AIR' optimize for all rows ;
```

Avec ses 3 EVI « Only Access » :
orders (orderkey), orders (partkey) et orders (quantity)



2 - Utiliser des EVI « Only Access »

… pour récupérer les colonnes des fichiers joints de la requête

Possible mais jamais vu. Pourquoi ?

L'optimiseur implémente les jonctions via 3 structures :
- les index binaires
- les tables de hachage
- les listes triées/non triées

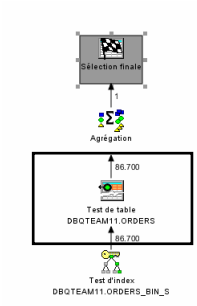
2 - Utiliser des EVI « Only Access »

- Les EVI sur toutes les colonnes doivent être créés
- Tous les EVI doivent pouvoir tenir en mémoire
- Maximum 20 colonnes
- Code d'inutilisation=20 (QQRID=3007, QQ1000)
  - Pas assez de mémoire dans le pool
- Code d'inutilisation=21 (QQRID=3007, QQ1000)
  - Toutes les colonnes n'ont pas d'EVI

3 - Utiliser des FBI « Only Access »

select max(quantity*extendedprice) from orders
where shipmode='AIR' ;

Avec l'index conseillé : BIN : orders (shipmode)

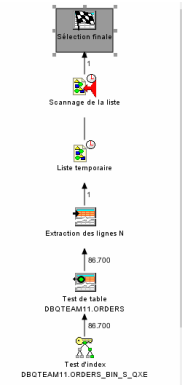


Informations temps	
Horodatage pour la création de poste de moniteur	2023-10-19-15.
Horodatage de début d'instruction	2023-10-19-15.
Horodatage de fin d'instruction	2023-10-19-15.
Durée totale d'exécution estimée (ms)	1,619
Récapitulatif sur la phase d'exécution	
Temps d'optimisation (ms)	1
Estimation de fourchette de clés la plus longue (ms)	0
Estimation du délai d'attente pour la fourchette de clés	Non
Temps d'exécution (ms)	Non disponible
Instruction Temps d'ouverture (ms)	Non disponible
Instruction Temps d'extraction (ms)	Non disponible
Instruction Temps de fermeture (ms)	Non disponible
Lignes extraites	Non disponible
Nombre total d'exécutions de la requête	Non disponible
Durée totale de toutes les exécutions (ms)	Non disponible
Temps UC	,999
Lectures synchrones base de données	Non disponible

3 - Utiliser des FBI « Only Access »

select max(quantity*extendedprice) from orders
where shipmode='AIR' ;

Avec l'index binaire FBI « Only Access » :
orders (shipmode, quantity*extendedprice)

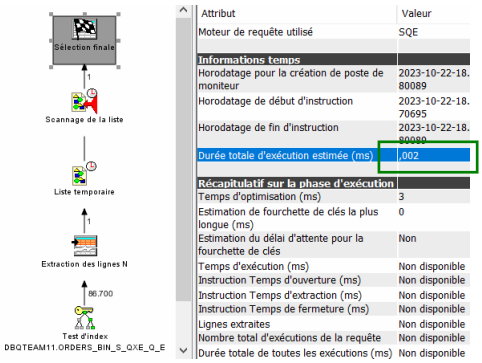


Attribut	Valeur
Moteur de requête utilisé	SQE
Informations temps	
Horodatage pour la création de poste de moniteur	2023-10-22-18. 2.758724
Horodatage de début d'instruction	2023-10-22-18. 2.741872
Horodatage de fin d'instruction	2023-10-22-18. 2.758724
Durée totale d'exécution estimée (ms)	3,236
Récapitulatif sur la phase d'exécution	
Temps d'optimisation (ms)	3
Estimation de fourchette de clés la plus longue (ms)	0
Estimation du délai d'attente pour la fourchette de clés	Non
Temps d'exécution (ms)	Non disponible
Instruction Temps d'ouverture (ms)	Non disponible
Instruction Temps d'extraction (ms)	Non disponible
Instruction Temps de fermeture (ms)	Non disponible
Lignes extraites	Non disponible
Nombre total d'exécutions de la requête	Non disponible
Durée totale de toutes les exécutions (ms)	Non disponible
Temps UC	,999
Lectures synchrones base de données	Non disponible
Lectures asynchrones base de données	Non disponible
Erreurs de page	Non disponible
Stockage temporaire utilisé (Mo)	Non disponible

3 - Utiliser des FBI « Only Access »

```
select max(quantity*extendedprice) from orders
where shipmode='AIR' ;
```

Avec l'index binaire FBI « Only Access » : orders (shipmode, quantity*extendedprice, quantity, extendedprice)



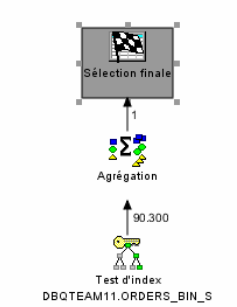
Attribut	Valeur
Moteur de requête utilisé	SQE
Informations temps	
Horodatage pour la création de poste de moniteur	2023-10-22-18.80089
Horodatage de début d'instruction	2023-10-22-18.70695
Horodatage de fin d'instruction	2023-10-22-18.80089
Durée totale d'exécution estimée (ms)	,002
Récapitulatif sur la phase d'exécution	
Temps d'optimisation (ms)	3
Estimation de fourchette de clés la plus longue (ms)	0
Estimation du délai d'attente pour la fourchette de clés	Non
Temps d'exécution (ms)	Non disponible
Instruction Temps d'ouverture (ms)	Non disponible
Instruction Temps d'extraction (ms)	Non disponible
Instruction Temps de fermeture (ms)	Non disponible
Lignes extraites	Non disponible
Nombre total d'exécutions de la requête	Non disponible
Durée totale de toutes les exécutions (ms)	Non disponible

Comment limiter l'agrégation et le scannage des index binaires ?

Exemple 1

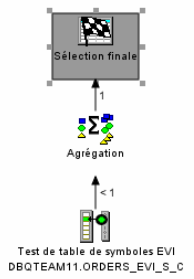
select count(*) from orders where shipmode='RAIL' ;

Avec l'index conseillé :
BIN : orders (shipmode) ;



Durée estimée : 48 ms

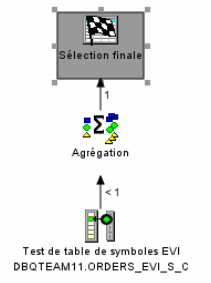
Ce que l'on peut faire :



Durée estimée : < 1 µs

Exemple 1 – Comment ?

select count(*) from orders where shipmode='RAIL';



Durée estimée : < 1 µs

EVI : orders (shipmode) include(count(*))

shipmode	code	count(*)
AIR	1	85819
FOB	2	85917
MAIL	3	86060
RAIL	4	85276
REG AIR	5	85374
SHIP	6	85830
TRUCK	7	86075

Table des symboles sur la colonne shipmode

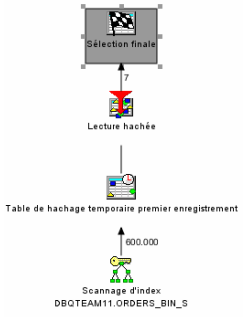
Autres styles de requête

- select distinct shipmode from orders ;
- select count(distinct shipmode) from orders ;

Exemple 2

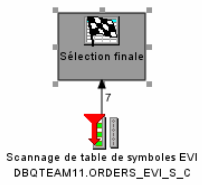
select shipmode, count(*) from orders group by shipmode optimize for all rows ;

Avec l'index conseillé :
BIN : orders (shipmode)



Durée estimée : 576 ms

Ce que l'on peut faire :

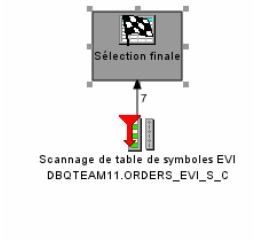


Durée estimée : < 1 µs

Exemple 2 – Comment ?

select shipmode, count(*) from orders group by shipmode optimize for all rows ;

EVI : orders (shipmode) include(count(*))



shipmode	code	count(*)
AIR	1	85819
FOB	2	85917
MAIL	3	86060
RAIL	4	85276
REG AIR	5	85374
SHIP	6	85830
TRUCK	7	86075

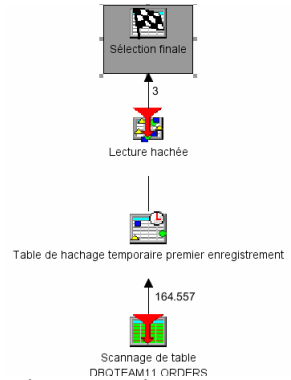
Durée estimée : < 1 µs

Table des symboles sur la colonne shipmode

Exemple 3

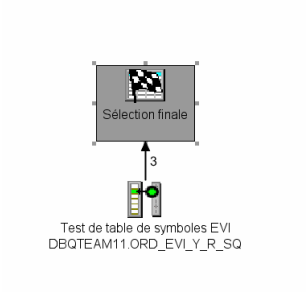
select year, returnflag, sum(quantity) from orders where year = 2023 group by year, returnflag optimize for all rows ;

Avec l'index conseillé :
BIN : orders (year, returnflag) ;



Durée estimée : 1 524 ms

Ce que l'on peut faire :

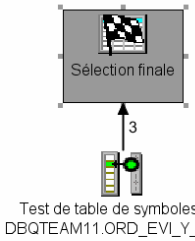


Durée estimée : < 1 µs

Exemple 3 – Comment ?

```
select year, returnflag, sum(quantity) from orders
where year = 2023 group by year, returnflag
optimize for all rows ;
```

EVI : orders (year, returnflag) include(sum(quantity)) ;



year, returnflag	code	sum(quantity)
2021, A	1	4762777
2021, R	2	146561
2022, A	3	5765543
2022, R	4	178383
2023, A	5	2906891
2023, N	6	1486051
2023, R	7	87937

Durée estimée : < 1 µs

Table des symboles sur les colonnes year, returnflag, sum(quantity)

Conclusion

- L'optimiseur délivre des conseils d'index équilibrés.
- Ces conseils devraient être étudiés pour éviter la sous-indexation dont souffre les IBM i.
- Lorsque l'on a des raisons valables on peut s'engager sur la voie de l'hyper optimisation dont nous venons de parler.

Questions/Réponses

Merci pour
votre présence
et
votre participation