

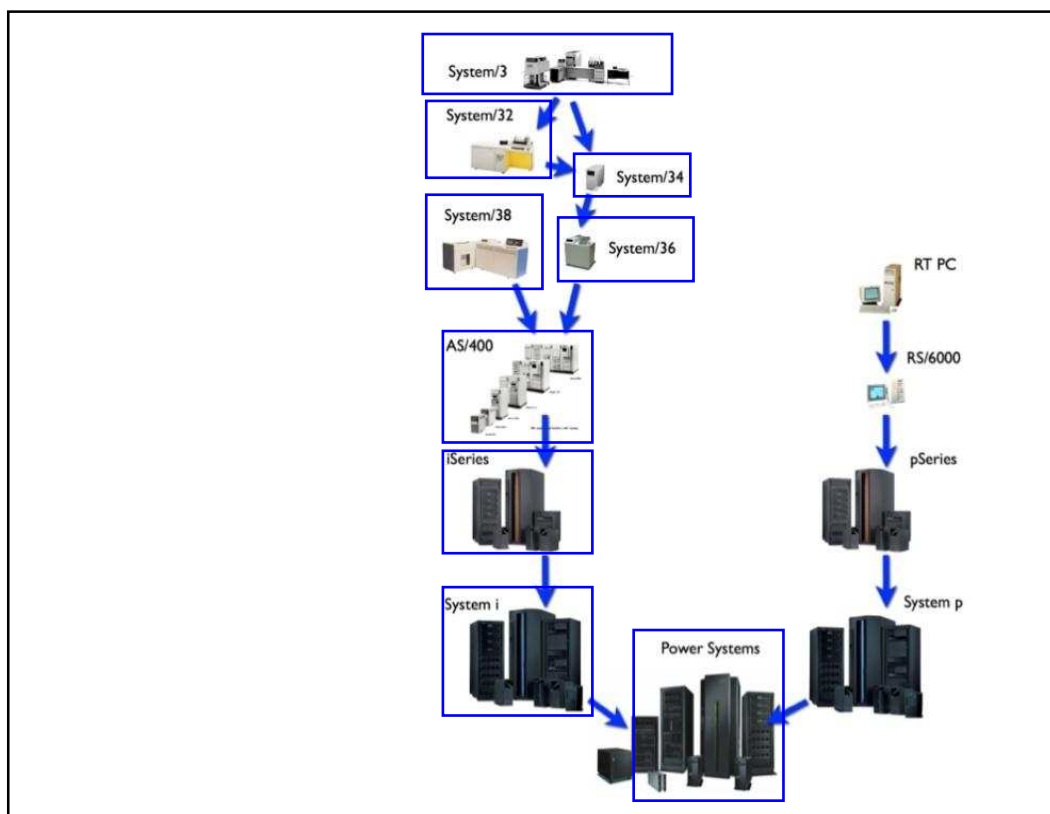
Les bonnes pratiques avec Db2 for i

Webinar Common France

28 mars 2023

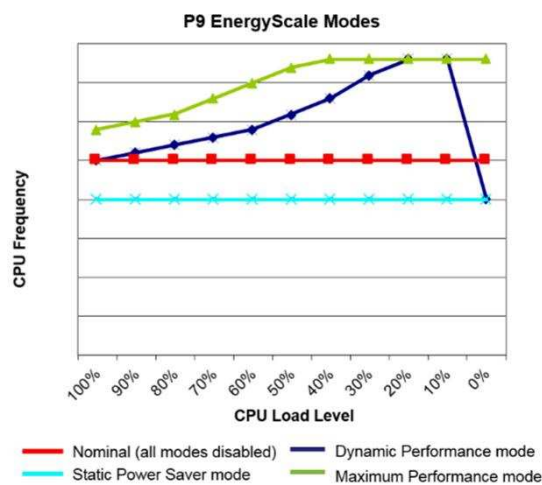
Christian GRIERE Services

cgriere@gmail.com



- Quelques bonnes pratiques qui concernent :
 - la configuration (CFG)
 - l'historique applicatif (HST)
 - le développement (DEV)

- Vérifier le mode énergétique de votre Power



CFG - Partition IBM i

Db2 for i

- Avoir une partition avec ressources non saturées
- Avoir des IOs performants
- Avoir un bon niveau logiciel

```
SELECT * from SYSTOOLS.GROUP_PTF_CURRENCY
```

```
WHERE PTF_GROUP_RELEASE = 'R740'
```

```
ORDER BY ptf_group_level_available, ptf_group_level_installed DESC ;
```

- Vérifier les références croisées
- Eviter les IPL
- Avoir si possible des pools mémoire de taille stable

CFG - Valeurs système Db2 for i

Db2 for i

- QQRVDEGREE=*IO/*OPTIMIZE
- QDBFSTCCOL=*ALL
- Fichier QUSRSYS/QAQQINI

Installer l'option 26 : DB2 Symmetric Multiprocessing

- Pré-requis : partition non saturée
- Exécution d'une requête SQL sur plusieurs processeurs
- Possibilité de limiter le % CPU de ce // (PARALLEL_MAX_SYSTEM_CPU)
- Choix des utilisateurs/applications à booster

-
- Maintenance des index en // en insertion groupée
 - Création d'index sur plusieurs processeurs
 - RGZPFM sur plusieurs processeurs

Quelques bonnes pratiques concernant l'historique applicatif (HST)

HST - Limiter le nombre de lignes supprimées Db2 for i

Limiter le pourcentage de lignes supprimées à 20 %

- Réutilisation des lignes supprimées
 - Fichiers physiques : par défaut non
 - Tables : par défaut oui
- RGZPFM

HST - Avoir une bonne taille de page logique

Db2 for i

Taille de page logique :

- Index (binaire) : 64 Ko
- Fichiers logiques : 2, 4 ou 8 Ko

HST - Avoir une bonne taille de page logique

Db2 for i

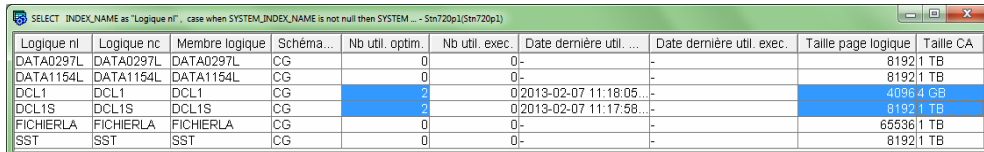
- Modifier les fichiers logiques ayant une taille de page logique de 2 Ko ou 4 Ko

CHGLF ... ACCPTHSIZ(*MAX1TB)

- Si vous avez des fichiers physiques de 2 Ko/4 Ko avec clés il faut les récréer.

HST - Avoir une bonne taille de page logique

Db2 for i



Logique nl	Logique nc	Membre logique	Schéma...	Nb util. optim.	Nb util. exec.	Date dernière util. ...	Date dernière util. exec.	Taille page logique	Taille CA
DATA0297L	DATA0297L	DATA0297L	CG	0	0-	-	-	81921 TB	
DATA1154L	DATA1154L	DATA1154L	CG	0	0-	-	-	81921 TB	
DCL1	DCL1	DCL1	CG	2	0	2013-02-07 11:18:05...	-	40964 GB	
DCL1S	DCL1S	DCL1S	CG	2	0	2013-02-07 11:17:58...	-	81921 TB	
FICHERLA	FICHERLA	FICHERLA	CG	0	0-	-	-	655361 TB	
SST	SST	SST	CG	0	0-	-	-	81921 TB	

Informations en provenance de la vue
QSYS2.SYSPARTITIONINDEXES

HST – Gérer la mixité

Db2 for i

- Des partages de chemin d'accès peuvent avoir lieu entre index binaire et fichier logique :
 - Si vous créez un index binaire I1 (c1) puis un fichier logique L1 (c1) il y a un partage du chemin d'accès porté par I1.
 - Si vous créez un index binaire I2 (c1, c2) puis un fichier logique L1 (c1) il y a un partage du chemin d'accès porté par I2.
 - Si vous créez un fichier logique L1 (c1) puis un index binaire I1 (c1) chaque structure a son propre chemin d'accès. Il n'y a pas de partage. Par contre si vous créez un fichier logique L1 (c1) avec PAGESIZE(64) puis un index binaire I1 (c1) il y a un seul chemin d'accès porté par L1.
 - Si vous créez un fichier logique L2 (c1, c2) puis un index binaire I1 (c1) il y aura deux chemins d'accès. Même si vous créez un fichier logique L2 (c1, c2) avec PAGESIZE(64) puis un index binaire I1 (c1) il y aura deux chemins d'accès.

Quelques bonnes pratiques concernant le développement (DEV)

DEV – Utiliser PREPARE et marqueurs

En SQL dynamique :

- 1 PREPARE avec marqueurs (?)
- N EXECUTE USING valeurs de ces marqueurs

Eviter les EXECUTE IMMEDIATE

DEV - Eviter les écritures « faciles »

Db2 for i

Exemple :

`SELECT *` *FROM ...* mais plutôt :

`SELECT col1, col4, col6 FROM ...`

- Diminue le nombre de calls faits au driver
Exemple : Une classe Java qui exécute une instruction SQL qui interroge une table de 20 colonnes.
'`SELECT *`' génère 40 calls database dans le driver
'`SELECT col1, col4, col6`' génère 6 calls database dans le driver
- Diminue la charge réseau
- Diminue la taille des structures temporaires du plan
- Augmente les chances d'utilisation d'un IOA (Index Only Access)

DEV – Respecter les types

Db2 for i

Exemple :

`SELECT * FROM ORDERS WHERE SHIPMODE=888`

Colonne SHIPMODE : type alpha

Visual Explain - NEPTUNE (Neptune)

Fichier Affichage Actions Options Outils Aide

Recherche... Ignorer la casse

Attribut Valeur

Moteur de requête utilisé SQE

Informations temps

Horodatage pour la création de poste de moniteur	2023-02-24-09.11.34.6091
Horodatage de début d'instruction	2023-02-24-09.11.34.6019
Horodatage de fin d'instruction	2023-02-24-09.11.34.6091
Durée totale d'exécution estimée (ms)	738,216

Récapitulatif sur la phase d'exécution en

Temps d'optimisation (ms)	1
Estimation de fourchette de clés la plus longue (ms)	0

Texte Instruction Messages de l'Optimiseur

```
SELECT *
FROM orders
WHERE shipmode = 888
```

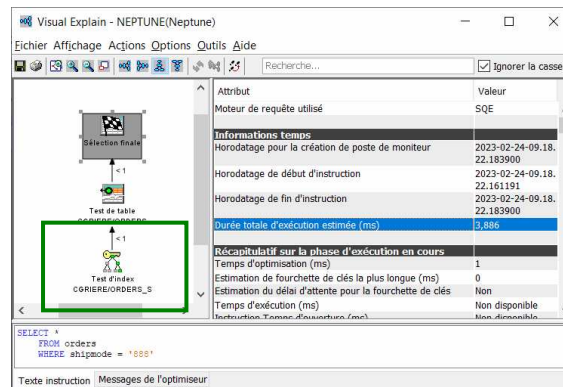
DEV – Respecter les types

Db2 for i

Exemple :

```
SELECT * FROM ORDERS WHERE SHIPMODE='888'
```

Colonne SHIPMODE : type alpha



DEV – Manipuler les gros volumes

Db2 for i

- FETCH N ROWS INTO ...
- INSERT N ROWS

DEV - Soyez causant dans vos SELECT

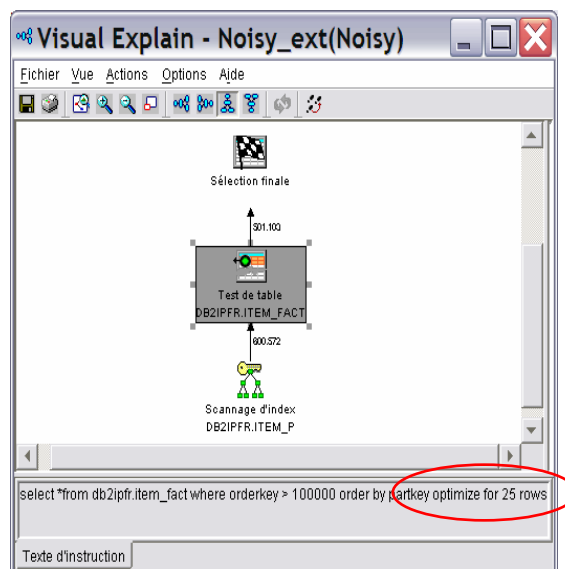
Db2 for i

Donner à Db2 for i le maximum d'information concernant vos requêtes.

Clauses : FOR READ ONLY,
FOR UPDATE OF,
FETCH FIRST N ROWS,
OPTIMIZE FOR n/ALL ROWS

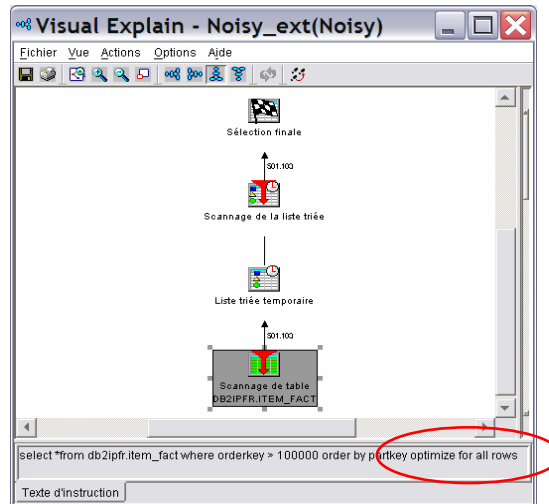
DEV - Clause OPTIMIZE FOR N/ALL ROWS

Db2 for i



DEV - Clause OPTIMIZE FOR N/ALL ROWS

Db2 for i



DEV – Exemple : les travaux QZDASOINIT

Db2 for i

- En charge des accès ODBC et JDBC
- Interface SQL dynamique
- Clause OPTIMIZE par défaut : 30 rows
- 2 utilisations :
 - Transactionnelle : ok ou ko
 - Extraction : ko
- Priorité par défaut : 20
 - Transactionnelle : ok (18 ?)
 - Extraction : ko

DEV - Lutter contre la sous-indexation

Db2 for i

- Symptôme de la sous-indexation :

Visual Explain - NEPTUNE(Neptune)

Fichier Affichage Actions Options Outils Aide

Recherche... Ignorer la casse

Attribut	Valeur
Moteur de requête utilisé	SQE
Informations temps	
Horodatage pour la création de poste de moniteur	2023-03-08-11.52.14.741440
Horodatage de début d'instruction	2023-03-08-11.52.14.718487
Horodatage de fin d'instruction	2023-03-08-11.52.14.741440
Durée totale d'exécution estimée (ms)	465,408
Récapitulatif sur la phase d'exécution en	
Temps d'optimisation (ms)	2
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

60,058

Scanage de table DB2IPFRITEM_FACT

```
SELECT *  
FROM db2ipfr.item_fact  
WHERE orderkey = 100
```

Texte instruction Messages de l'optimiseur

DEV - Lutter contre la sous-indexation

Db2 for i

- Remède à la sous-indexation :

Visual Explain - NEPTUNE(Neptune)

Fichier Affichage Actions Options Outils Aide

Recherche... Ignorer la casse

Attribut	Valeur
Moteur de requête utilisé	SQE
Informations temps	
Horodatage pour la création de poste de moniteur	2023-03-08-11.47.47.610071
Horodatage de début d'instruction	2023-03-08-11.47.47.594637
Horodatage de fin d'instruction	2023-03-08-11.47.47.610071
Durée totale d'exécution estimée (ms)	6,758
Récapitulatif sur la phase d'exécution en	
Temps d'optimisation (ms)	6
Estimation de fourchette de clés la plus longue (ms)	4
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

5

Test de table DB2IPFRITEM_FACT

Test d'index DB2IPFRITEM_O

```
SELECT *  
FROM db2ipfr.item_fact  
WHERE orderkey = 100
```

Texte instruction Messages de l'optimiseur

-
- Depuis 1980 (pas 1988) le message a été :
« un logique coûte cher »
 - Quoi ? Comment ?
 - Le développeur AS/400 fait le 'Quoi' et le 'Comment'
 - Le développeur Y ou Z ne fait souvent que le 'Quoi'

-
- Le message maintenant :
« un index manquant coûte cher »
 - Il faut aussi décider de qui fait le 'Comment'

1 ?

2 ?

3

4 ?

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

Where

Join

Order by

Group by

Structures permanentes

Tables
Index binaire
Index EVI

Méthodes d'accès

Positionnement
Lecture séquentielle

Structures temporaires

Liste triée
Liste non triée
Table de hachage
Bitmap
Liste de rang
Buffer
Cache

...

DEV - Niveau d'optimisation

Db2 for i

- 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

DEV – Optimisation niveau 1 (base)

Db2 for i

- Index sur les colonnes du WHERE
 - Colonnes d'égalité en premier
 - 1 colonne d'inégalité
- Index sur les colonnes du JOIN
 - Colonnes d'égalité en premier
 - 1 colonne d'inégalité
- Index sur les colonnes de l'ORDER BY (pourra être utilisé si pas de WHERE)
- Index sur les colonnes du GROUP BY

Ces index sont suggérés par l'optimiseur

- Définition d'un index parfait
 - Index binaire, permanent et multi-colonnes
 - Il donne la meilleure performance (moins d'I/O)
 - Il permet de solutionner au moins 2 clauses SQL
 - Where + Order by
 - Where + Group by
 - Where + Join

- Ordre des colonnes d'un index parfait est primordial
 - Colonnes d'égalité en premier
 - Si tous les prédicats ont un opérateur d'égalité :
 - Colonnes du WHERE + colonnes du JOIN
 - Colonnes du JOIN + colonnes du WHERE
 - Colonnes du WHERE + colonnes du GROUP BY
 - Colonnes du WHERE + colonnes du ORDER BY
 - Les colonnes les + sélectives en tête (connaissance des données ou/et statistiques)

Ces index sont suggérés par l'optimiseur

DEV – Optimisation niveau 3 (hyper optimisation) Db2 for i

Elle recouvre :

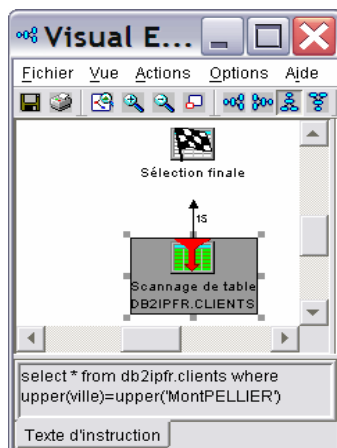
- les Encoded Vector Index (EVI) *
- le GAP/LPG (Génération Anticipée de Prédicats/Look-ahead Predicate Generation)
- les Index Only Access (IOA binaires ou EVI) *
- les Function Based Index (FBI) *

* Ces index ne sont pas suggérés par l'optimiseur

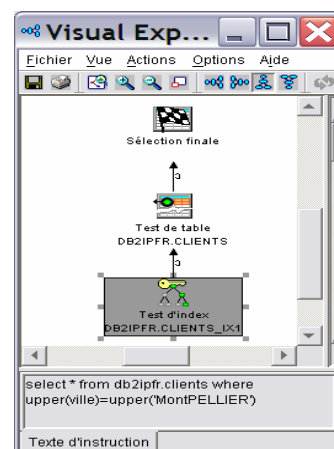
DEV – Exemple Function Based Index Db2 for i

Exemple :

SELECT ... FROM CLIENTS WHERE UPPER(ville) = '?'



CREATE INDEX
CLIENTS_IX1 ON
CLIENTS
(UPPER(ville))



DEV - Remplacer les MTI par des index permanents

Db2 for i

Images instantanées de mémoire cache de plan / Moniteurs d'événements de mémoire cache de plan

Description	Valeur	Unité de valeur
Heure du recalculatif	2022-03-22-10.11.34.392969	
Heure de création de la mémoire cache de plans	2022-02-21-00.32.38.067257	
Nombre de requêtes actives en cours	10531	
Nombre de requêtes exécutées depuis le début	1907650013	
Nombre d'ouvertures complétées de requête depuis le début	10633379	
Nombre de plans en mémoire cache	108584	
Nombre total de plans concus depuis le début	1252707	
TOTAL NUMBER OF SMP PLANS BUILT SINCE START	32287	
Nombre total de requêtes uniques depuis le début	400253	
Taille de la mémoire cache de plans en cours	6034	Mo
Seuil actuel de taille de la mémoire cache de plans	*AUTO	Mo
Taille du cache de plan maximum pour dimensionnement auto	*DEFAULT (6144)	Mo
Taux d'accès en cours de cache de plan	87	%
Taux d'accès de dimensionnement auto de cache de plan cible	*DEFAULT (90)	%
TOTAL NUMBER OF PLAN CACHE AUTOSIZING ADJUSTMENTS	11	
Dernier ajustement de redimensionnement auto du cache de plan	2022-02-24-23.41.52.357874	
LAST AUTOSIZING LIMITED DUE TO TEMPORARY STORAGE	0000-00-00-00.00.00.000000	
Nombre en cours de plans de niveau travail (OTEMP)	27871	
Nombre total de plans de niveau travail (OTEMP) concus depuis le début	683452	
Nb total requêtes uniques avec ré de niveau travail (OTEMP) depuis le début	29161	
Nombre de plans utilisés à partir de la mémoire cache	524074	
Nombre total de plans retirés	746003	
Nombre total de plans supprimés	628281	
Nombre d'élagages de la mémoire cache de plans	70	
Dernier élagage de la mémoire cache de plans	2022-03-22-08.40.29.331641	
Nombre en cours d'objets d'exécution temporaires stockés en mémoire cache	79372	
Taille totale en cours des objets d'exécution temporaires en mémoire cache	69010	Mo
Nombre maximal d'objets d'exécution temporaires par plan	*DEFAULT (5)	
Nombre total d'index temporaires créés	73134	
Nombre de plans reconstruits en raison de AQP	34	
CURRENT TOTAL SIZE OF TEMPORARY INDEXES	19677	Mo
Nombre de plans reconstruits en raison de AQP	2172	
Nombre d'erreurs de mappage de requête depuis le début	2172	
Seuil actuel de taille de la mémoire cache de plans	*AUTO	Mo
Taille du cache de plan maximum pour dimensionnement auto	*DEFAULT (6144)	Mo
Taux d'accès de dimensionnement auto de cache de plan cible	*DEFAULT (90)	%
Nombre maximal d'exécutions les plus longues autorisées par plan	*DEFAULT (3)	
Nombre maximal d'objets d'exécution temporaires par plan	*DEFAULT (5)	
Heure de début des seuls d'activité	2022-02-21-00.32.38.069158	
Nombre le plus élevé de requêtes actives à un instant donné	15309 (2022-03-15-15.31.21.29...)	
Nombre le plus élevé de plans dans la mémoire cache	123173 (2022-02-25-11.38.17.4...)	

Terminé : 53 lignes extraites

DEV - Remplacer les MTI par des index permanents

Db2 for i

Obtenir la liste :

– Table fonction MTI_INFO

```
SELECT * FROM TABLE(QSYS2.MTI_INFO('*ALL', '*ALL'))
ORDER BY table_schema ;
```

SELECT * FROM TABLE(QSYS2.MTI_INFO('*ALL', '*ALL')) order by table_schema - NEPTUNE(Neptune)

TABLE_SCHEMA	TABLE_NAME	REFERENCE_COUNT	KEYS	KEY_DEFINITION	STATE	REUSABLE	SPARSE	MTI_SIZE	CREATE
EXPLOIT	LOGDUJOUR	31	2	MESSAGE_ID, FROM_JOB	VALID	YES	NO	555528	2023-02-...
GS	GTXTO1	54	1	ARGTXT	VALID	YES	NO	162312	2023-02-...
ITEMPS	ADE_TICKET	5	1	INC_SCAT	VALID	YES	NO	162312	2023-02-...
ITEMPS	ADE_TICKET	5	1	ASS_TO	VALID	YES	NO	170504	2023-02-...
LA	AIRPORTS	2	1	DESCP	VALID	YES	NO	134190	2023-02-...
NK	MITMAS	2	1	MMSTAT	VALID	YES	NO	13662728	2023-02-...
PFRCOFI10	QAPMSYSTEM	2	1	INTNUM	CREATED	YES	NO	162312	2023-02-...
PFRCOFI10	QAPMLPAR	2	2	LPPID, INTNUM	VALID	YES	NO	190984	2023-02-...
PFRCOFI10	QAPMHTTTPB	2	3	HTJNAM, HTJUSR, HTJNBR	VALID	YES	NO	162312	2023-02-...
PFRCOFI10	QAPMTCRPEC	2	2	TTYPE, TLIND	VALID	YES	NO	162312	2023-02-...
PFRCOFI10	QAPMETH	2	3	IOPRN, ETLLND, ETMPORT	VALID	YES	NO	162312	2023-02-...
PFRCOFI10	QAPMDISK	2	5	DSASP, DSARM, DMFLAG, DSIP, DSDRN	VALID	YES	NO	134190	2023-02-...
PFRCOFI10	QAPMPOOLB	2	1	PONBR	VALID	YES	NO	186888	2023-02-...
PFRCOFI10	QAPMJOBMI	16	4	INTNUM, JBNAME, JBUSER, JBNSR	VALID	YES	NO	28473864	2023-02-...
PFRCOFI10	QAPMJOBMI	1	1	JBPRTY	VALID	YES	NO	13662728	2023-02-...
PFRCOFI10	QAPMJOBMI	14	3	JBNAME, JBUSER, JBNSR	VALID	YES	NO	19036680	2023-02-...
PKG0001000	HSTPKGS	2	1	NOMVERS	VALID	YES	NO	162312	2023-02-...
PKG0001000	HSTVERS	6	1	NOMVERS	VALID	YES	NO	162312	2023-02-...

Terminé : 29 lignes extraites 22/02/2023 à 20:51:54

-
- Table QSYS2/SYSIXADV
 - Lignes QQRID=3002 des moniteurs de performance SQL et images instantanées de cache de plan

-
- Si vous disposez de scénarii (automatisés ou manuels) pour vérifier que l'application fonctionne correctement après modification du code il est judicieux de démarrer en même temps un moniteur de performance SQL
 - En cas d'anomalie suite à l'installation de PTF ou un changement de version il suffira de rejouer le scénario avec un moniteur de performance SQL
 - La comparaison des 2 moniteurs (fonction standard d'ACS) :
 - montrera l'origine de l'anomalie
 - vous serez mieux armé pour ouvrir votre incident auprès du Point Service IBM.

Des questions ?

[linkedin.com/in/christian-griere-6a3828a/](https://www.linkedin.com/in/christian-griere-6a3828a/)

Merci pour
votre présence virtuelle
et
votre participation réelle