|  |  |
| --- | --- |
|  |  |

Audit de la base Oracle STORPROD sur MARCELLO

Le Bon Marché

Rapport d’audit

|  |  |
| --- | --- |
|  | **Version :** V 1.3  **Date de la version**: 09/11/2017  **Etat du document :** Final |

Diffusion

|  |  |
| --- | --- |
| Nom | Société |
|  | UMANIS |
|  |  |

Historique des modifications

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Auteur | Objet de la version et liste des modifications |
| V 1.0 | 03/10/2017 | JEA | Initialisation |
| V 1.3 | 09/11/2017 | JEA | Finalisation |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

VAlidation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Version | Emetteur | Date / Visa | Vérificateur | Date / Visa | Approbateur | Date / Visa |
| V 1.3 | JEA | 09/11/2017 |  |  |  |  |
|  |  |  |  |  |  |  |

**Date d'application :** 09/11/2017

Table des matières

[1 Objet de l’audit 5](#_Toc497990446)

[2 Environnement 6](#_Toc497990447)

[2.1 Os 6](#_Toc497990448)

[2.2 RDBMS 6](#_Toc497990449)

[2.2.1 La base STORPROD 6](#_Toc497990450)

[2.3 Les snapshots statspack 7](#_Toc497990451)

[3 Les points forts 8](#_Toc497990452)

[3.1 Base éfficiente et stable 8](#_Toc497990453)

[4 Les points faibles 9](#_Toc497990454)

[4.1 Version d’Oracle non supportée par le constructeur 9](#_Toc497990455)

[4.2 Quelques objets compilés en mode debug 9](#_Toc497990456)

[4.3 Présence d’objets invalides 9](#_Toc497990457)

[5 Les opportunités d’amélioration 10](#_Toc497990458)

[5.1 Mettre en place un plan de migration vers une version en cours de support 10](#_Toc497990459)

[5.2 Augmenter la RAM 10](#_Toc497990460)

[5.3 Compiler en mode standard les objets compilés en mode débug. 10](#_Toc497990461)

[5.4 Corriger ou supprimer les objets invalides 10](#_Toc497990462)

[5.5 Tables fragmentées 11](#_Toc497990463)

[5.6 Ajouter des index 13](#_Toc497990464)

[5.6.1 Table LBM\_VENTE\_POSTE 13](#_Toc497990465)

[5.7 Corriger des requêtes consommatrices de ressources 13](#_Toc497990466)

[5.7.1 Package STORELAND.PK\_LBM\_VENTES\_GU 13](#_Toc497990467)

[Storland.exe 15](#_Toc497990468)

[6 Annexes 17](#_Toc497990469)

[6.1 Script de compilation des objets en mode debug 17](#_Toc497990470)

[6.2 Script de compilation des objets invalides 17](#_Toc497990471)

[6.3 Compression des tables fragmentées 18](#_Toc497990472)

[6.4 Extrait de V$DATABASE 18](#_Toc497990473)

# Objet de l’audit

Cet audit a pour but de déterminer le comportement global de la base STORPROD sur MARCELLO supportant la solution Cylande. Il vise en particulier à expliquer les alertes régulièrement remontées dans le monitoring des applications sur cette base de données

# Environnement

## Os

La base STORPROD est installée sur le serveur MARCELLO doté d’un système Microsoft Windows Server 2008 64-bit et disposant de 8 Go de RAM.

## RDBMS

STORPROD sur MARCELLO est une base en version 10.2.0.5.0. Elle est en rôle PRIMARY\_INSTANCE et fonctionne en NO ARCHIVE LOG.

### La base STORPROD

#### Caractéristiques physiques

##### Les tablespaces

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TABLESPACE | STATUS | CONTENTS | EXTENT MANAGEMENT | BLOCK SIZE | MIN EXTENTS | MAX EXTENTS |
| SYSTEM | ONLINE | PERMANENT | LOCAL | 8192 | 65536 | 2147483645 |
| UNDOTBS1 | ONLINE | UNDO | LOCAL | 8192 | 65536 | 2147483645 |
| SYSAUX | ONLINE | PERMANENT | LOCAL | 8192 | 65536 | 2147483645 |
| TEMP | ONLINE | TEMPORARY | LOCAL | 8192 | 1048576 |  |
| USERS | ONLINE | PERMANENT | LOCAL | 8192 | 65536 | 2147483645 |
| STORELAND\_DATA | ONLINE | PERMANENT | LOCAL | 8192 | 65536 | 2147483645 |
| STORELAND\_IDX | ONLINE | PERMANENT | LOCAL | 8192 | 65536 | 2147483645 |

Les tables et index du schéma STORELAND sont respectivement répartis sur les tablespaces STORELAND\_DATA et STORELAND\_IDX.

##### Les database files du schéma Storeland

| Taille (octets) |  | Fichier |
| --- | --- | --- |
| 5 444 214 784 |  | STORELAND\_DATA01.ORA |
| 5 444 214 784 |  | STORELAND\_DATA02.ORA |
| 5 444 214 784 |  | STORELAND\_DATA03.ORA |
| 5 444 214 784 |  | STORELAND\_DATA04.ORA |
| 5 444 214 784 |  | STORELAND\_DATA05.ORA |
| 5 444 214 784 |  | STORELAND\_DATA06.ORA |
| 5 444 214 784 |  | STORELAND\_DATA07.ORA |
| 5 444 214 784 |  | STORELAND\_DATA08.ORA |
| 5 444 214 784 |  | STORELAND\_DATA09.ORA |
| 5 444 214 784 |  | STORELAND\_DATA10.ORA |
| 5 444 214 784 |  | STORELAND\_DATA11.ORA |
| 5 444 214 784 |  | STORELAND\_DATA12.ORA |
| 5 444 214 784 |  | STORELAND\_IDX01.ORA |
| 5 444 214 784 |  | STORELAND\_IDX02.ORA |
| 5 444 214 784 |  | STORELAND\_IDX03.ORA |
| 5 444 214 784 |  | STORELAND\_IDX04.ORA |
| 5 444 214 784 |  | STORELAND\_IDX05.ORA |
| 10 485 768 192 |  | STORELAND\_IDX06.ORA |

## Les snapshots statspack

Les prises de mesures sont paramétrées tous les jours, du 03/10/2017 au 20/10/2017, de 10 heures à 20 heures avec une fréquence de 15 minutes ; tous les snapshots sont conservés sur cette période. Il est ainsi possible d’investiguer sur les perturbations et dysfonctionnements perçues dans cet intervalle.

# Les points forts

## Base éfficiente et stable

L’essentiel du modèle de données est équilibré : les tables et vues les plus utilisées (celles revenant le plus souvent dans les snapshots) disposent de clés primaires et référentielles cohérentes et référencées dans les restrictions des requêtes les plus utilisées. De même, les tables ayant un nombre d’enregistrement négligeable (telles que les tables de paramètres) n’ont généralement pas d’index pour ne pas en dégrader les temps d’accès.

Les ressources systèmes ne sont pas débordées lors des phases d’activité critiques. Nous avons cependant noté un taux d’occupation moyen de la mémoire très élevé, 85% de la RAM.

# Les points faibles

## Version d’Oracle non supportée par le constructeur

La version 10.2 d’Oracle n’est plus supportée par le constructeur depuis juillet 2013. (Voir la liste des version et leurs catégories de support <http://www.oracle.com/us/support/library/lifetime-support-technology-069183.pdf>, en page 4).

## Quelques objets compilés en mode debug

Ces objets reviennent souvent sans la liste de ceux générant le plus de consommation de ressource CPU et de ressource mémoire. Ils sont aussi parmi ceux qui génèrent le plus de Waits. Nous pouvons ainsi citer par exemple le package PK\_LBM\_VENTES\_GU.

## Présence d’objets invalides

Il y a de nombreux objets invalides dans la base de données. Voici leur répartition par type et par propriétaire.

| OWNER | OBJECT\_TYPE | COUNT |
| --- | --- | --- |
| OLAPSYS | VIEW | 227 |
| OLAPSYS | FUNCTION | 1 |
| OLAPSYS | PACKAGE BODY | 32 |
| PUBLIC | SYNONYM | 173 |
| STORELAND | PROCEDURE | 10 |
| STORELAND | PACKAGE | 3 |
| STORELAND | PACKAGE BODY | 15 |
| SYSMAN | PACKAGE | 1 |
| SYSMAN | PACKAGE BODY | 15 |
| SYSMAN | TRIGGER | 1 |
| SYSMAN | FUNCTION | 4 |

# Les opportunités d’amélioration

## Mettre en place un plan de migration vers une version en cours de support

Disposer d’une version qui n’est plus supportée par Oracle entraîne des risques de surcout en cas de nécessité de support par Oracle. Il y a aussi le risque de devoir migrer d’urgence en cas de dysfonctionnement important.

## Augmenter la RAM

Le taux moyen de sollicitation de la RAM étant très élevé, le serveur peut être ralenti par un recours massif au mécanisme de swap. Plus de RAM devrait le soulager lors des pics de traitements.

## Compiler en mode standard les objets compilés en mode débug.

Attention, il peut y avoir des instabilités si un grand nombre d’objets ayant de fortes interdépendances sont ainsi refacturés. Il faudrait le faire avec une session d’administration connectée en mode autonome et redémarrer la base Oracle à la suite de cette séance de recompilation d’objets.

## Corriger ou supprimer les objets invalides

La présence d’objets invalides est source de confusion. En cas de dysfonctionnement des applications, les objets invalides semant la confusion dans l’analyse. Le délai de résolution des pannes sera augmenté des durées de levée des doutes sur la pertinence des objets en cause. Il faut donc supprimer les objets réellement inutiles et corriger ceux qui restent invalides. Ainsi par exemple doutons-nous de la pertinence, parmi les objets invalides, de ceux-ci appartenant au schéma STORELAND :

| OBJET | TYPE |
| --- | --- |
| PK\_LBM\_BKP\_10122014 | Package |
| PK\_LBM\_OLD | Package |
| PK\_LBM\_OLD\_OLD | Package |
| ZZ\_20170216\_PK\_MVS\_STANDARD | Package |
| ZZ\_20170222\_PK\_LBM | Package |
| ZZ\_20170314\_PK\_LBM | Package |
| ZZ\_20170413PK\_MVS\_STANDARD | Package |
| ZZ\_20170413\_PK\_MVS\_STANDARD | Package |
| ZZ\_20170509\_PK\_MVS\_STANDARD | Package |

## Tables fragmentées

Certaines tables sur lesquelles se font un grand nombre d’entrées/sorties sont largement fragmentées (plus de 20%). Elles devront donc être défragmentées pour en restaurer l’efficacité. Dans le tableau suivant, nous listons dans un ordre décroissant selon le nombre d’enregistrements de la table, son nombre d’entrées/sorties et le taux de fragmentation. Nous avons limité la portée de cette proposition aux tables de plus de 20 000 enregistrements. Les tables en têtes de liste devraient être traitées en priorité.

Il est à noter que les index des tables compressées devront être reconstruits pour conserver leur efficacité, compte tenu des taux élevés de fragmentation.

| Table | Enregistrements | Blocs | Fragmentée (ko) | Données (ko) | Espace perdu (ko) | Espace récupérable % |
| --- | --- | --- | --- | --- | --- | --- |
| STATS\_CLIENT | 16 092 091 | 55 112 | 440 896,00 | 282 868,79 | 158 027,21 | 25,84 |
| LBM\_STATS | 6 505 214 | 76 886 | 615 088,00 | 412 928,62 | 202 159,38 | 22,87 |
| CONSOLE\_CONNEXION\_HISTORIQUE | 2 635 711 | 10 236 | 81 888,00 | 38 609,05 | 43 278,95 | 42,85 |
| CONSOLE\_TICKETS\_MANQUANTS | 89 149 | 588 503 | 4 708 024,00 | 1 915,31 | 4 706 108,69 | 89,96 |
| LBM\_MESSAGE\_ID | 599 927 | 11 273 | 90 184,00 | 35 151,97 | 55 032,03 | 51,02 |
| CONSOLE\_FONCTION\_HISTORIQUE | 2 685 654 | 12 284 | 98 272,00 | 44 586,05 | 53 685,95 | 44,63 |
| VENTES\_ARTICLE\_PERIODE | 1 635 644 | 22 370 | 178 960,00 | 105 422,37 | 73 537,63 | 31,09 |
| VENTES\_ARTICLE\_MAG\_PERIODE | 4 526 105 | 213 566 | 1 708 528,00 | 229 841,27 | 1 478 686,73 | 76,55 |
| HISTORIQUE\_CAISSES | 3 694 802 | 124 419 | 995 352,00 | 660 301,53 | 335 050,47 | 23,66 |
| LOG\_TRAITEMENTS | 877 288 | 23 504 | 188 032,00 | 89 956,29 | 98 075,71 | 42,16 |
| ERREUR | 183 652 | 3 440 | 27 520,00 | 15 065,20 | 12 454,80 | 35,26 |
| LBM\_CLIENT\_TOP | 424 882 | 3 180 | 25 440,00 | 13 277,56 | 12 162,44 | 37,81 |
| TARIFS | 6 117 952 | 50 144 | 401 152,00 | 262 880,75 | 138 271,25 | 24,47 |
| PRODUITS\_COLORIS | 5 933 123 | 70 592 | 564 736,00 | 393 996,45 | 170 739,55 | 20,23 |
| CLIENTS\_GPE\_MAGASIN | 1 679 384 | 5 070 | 40 560,00 | 27 880,40 | 12 679,60 | 21,26 |
| HISTO\_CB\_FRN | 5 970 441 | 39 012 | 312 096,00 | 198 237,30 | 113 858,70 | 26,48 |
| PRODUITS\_ETIQUETTE | 5 933 633 | 26 477 | 211 816,00 | 139 069,52 | 72 746,48 | 24,34 |
| TVA\_PRODUIT\_PAYS | 5 929 001 | 17 907 | 143 256,00 | 75 270,52 | 67 985,48 | 37,46 |
| STATS\_PRODUIT\_COLORIS | 5 928 598 | 29 680 | 237 440,00 | 133 161,87 | 104 278,13 | 33,92 |
| DESCRIPTIONS\_PRODUIT\_WEB | 5 926 735 | 31 712 | 253 696,00 | 167 846,99 | 85 849,01 | 23,84 |
| AGREGAT\_TICKET\_VENTE | 2 141 020 | 31 333 | 250 664,00 | 165 176,35 | 85 487,65 | 24,10 |
| SUIVI\_ALERTES\_WST | 1 936 153 | 20 460 | 163 680,00 | 102 101,82 | 61 578,18 | 27,62 |
| RESAZONEVAL\_WST | 1 600 086 | 7 750 | 62 000,00 | 42 189,77 | 19 810,23 | 21,95 |
| CONSOLE\_OUVERTURE\_HISTORIQUE | 1 592 664 | 6 474 | 51 792,00 | 35 772,73 | 16 019,27 | 20,93 |
| SOS\_STATS\_PRODUIT\_COLORIS | 951 570 | 3 493 | 27 944,00 | 19 514,62 | 8 429,38 | 20,17 |
| REACTIONS\_MARKETING | 917 668 | 12 664 | 101 312,00 | 51 081,13 | 50 230,87 | 39,58 |
| CONSOLE\_TICKET\_HISTORIQUE | 759 433 | 4 618 | 36 944,00 | 19 282,48 | 17 661,52 | 37,81 |
| LBM\_TICKET\_EXTRAIT | 741 769 | 3 442 | 27 536,00 | 13 038,91 | 14 497,09 | 42,65 |
| LBM\_DETAXE | 639 268 | 1 902 | 15 216,00 | 8 739,99 | 6 476,01 | 32,56 |
| CLIENTS\_GPE\_MAGASIN\_PFL | 574 112 | 1 706 | 13 648,00 | 9 531,16 | 4 116,84 | 20,16 |
| LBM\_CARTES\_CLIENTS | 523 188 | 8 083 | 64 664,00 | 43 428,69 | 21 235,31 | 22,84 |
| HISTO\_MVTS\_STOCK\_MAG | 448 579 | 8 698 | 69 584,00 | 24 093,60 | 45 490,40 | 55,37 |
| PIECES\_COMPTABLE | 242 628 | 2 672 | 21 376,00 | 14 690,37 | 6 685,63 | 21,28 |
| DLA\_ARTICLE\_A\_ENVOYER | 216 406 | 548 | 4 384,00 | 2 536,01 | 1 847,99 | 32,15 |
| LBM\_ARTICLE\_REVISION | 160 859 | 1 020 | 8 160,00 | 4 084,31 | 4 075,69 | 39,95 |
| CLIENTS\_GPE\_MAGASIN\_SUPP\_CRM | 140 056 | 420 | 3 360,00 | 2 325,15 | 1 034,85 | 20,80 |
| RESAZONEVAL\_WST\_SAV | 139 852 | 964 | 7 712,00 | 4 780,10 | 2 931,90 | 28,02 |
| LBM\_REPRISE\_CHKDO | 127 292 | 421 | 3 368,00 | 2 237,55 | 1 130,45 | 23,56 |
| TMP\_ARTCODE\_0901 | 124 136 | 323 | 2 584,00 | 1 454,72 | 1 129,28 | 33,70 |
| STATS\_PRODUIT\_LBM | 93 925 | 496 | 3 968,00 | 2 201,37 | 1 766,63 | 34,52 |
| LBM\_LOG\_PERSONNELS | 78 958 | 250 | 2 000,00 | 308,43 | 1 691,57 | 74,58 |
| BCH\_ECART\_PRIX | 69 486 | 185 | 1 480,00 | 814,29 | 665,71 | 34,98 |
| CONSOLE\_TRANSFERT\_HISTORIQUE | 55 475 | 252 | 2 016,00 | 1 246,02 | 769,98 | 28,19 |
| LBM\_PHI | 54 776 | 219 | 1 752,00 | 1 176,83 | 575,17 | 22,83 |
| CONSOLE\_FERMETURE\_HISTORIQUE | 54 244 | 256 | 2 048,00 | 1 218,37 | 829,63 | 30,51 |
| HISTO\_ACCROCHAGES\_TICKET | 50 425 | 370 | 2 960,00 | 1 575,78 | 1 384,22 | 36,76 |
| RESERVATIONS\_WST\_SAV | 39 661 | 814 | 6 512,00 | 4 221,73 | 2 290,27 | 25,17 |
| HEXAPOSTE | 39 616 | 171 | 1 368,00 | 928,50 | 439,50 | 22,13 |
| LBM\_HISTO\_STATSCLIENT | 36 698 | 192 | 1 536,00 | 1 075,14 | 460,86 | 20,00 |
| MAG\_JOUR | 25 978 | 354 | 2 832,00 | 1 623,63 | 1 208,37 | 32,67 |
| TRAITEMENTS\_MAGASIN | 24 939 | 118 | 944,00 | 657,57 | 286,43 | 20,34 |
| NUM\_FACTURE\_TICKET | 23 020 | 252 | 2 016,00 | 786,82 | 1 229,18 | 50,97 |
| CLIENTS\_GPE\_MAGASIN\_DOUBLONS | 22 061 | 74 | 592,00 | 366,25 | 225,75 | 28,13 |
| CONSOLE\_MENU\_HISTORIQUE | 20 123 | 103 | 824,00 | 353,72 | 470,28 | 47,07 |

## Ajouter des index

### Table LBM\_VENTE\_POSTE

Cette table est fortement sollicitée par des requêtes qui n’exploitent pas son index de clé primaire en appliquant des fonctions sur les premiers champs de la clé. Certaines de ces requêtes sont exécutées par un fichier compilé (STORELAND.EXE) et ne seront pas facilement modifiables. D’où la nécessité d’un nouvel index du type function-based index.

1. Créer l’index

CREATE INDEX LBM\_VENTE\_POSTE\_FB\_STEC\_DATEH

ON STORELAND.LBM\_VENTE\_POSTE (To\_Number(stecaisse), Trunc (datehtrans)) ;

1. Calculer les statistiques de la table ou au moins du nouvel index

EXEC DBMS\_STATS.gather\_table\_stats

(

'STORELAND'

, 'LBM\_VENTE\_POSTE'

, cascade => TRUE

);

EXEC DBMS\_STATS.GATHER\_INDEX\_STATS

(

ownname => 'STORELAND'

, indname => 'LBM\_VENTE\_POSTE\_FB\_STEC\_DATEH'

, force => TRUE

);

Il est à noter que la création de cet index n’est pas contradictoire avec la correction des requêtes sur la table LBM\_VENTE\_POSTE pour privilégier l’usage de l’index de clé primaire qui a une efficacité supérieure.

## Corriger des requêtes consommatrices de ressources

### Package STORELAND.PK\_LBM\_VENTES\_GU

#### Procédure PK\_LBM\_VENTES\_GU.LBM\_GETTICKETSBM

Cette procédure affiche de mauvaises performances par ce qu’elle est très souvent sollicitée et scanne entièrement une grosse table (LBM\_VENTE\_POSTE), n’exploitant pas son index de clé primaire (LBM\_VENTE\_POSTE\_PK) comme elle le devrait.

/\*

\*\* Corriger la restriction sur lbm\_vente\_poste.stecaisse

\*\* pour ne plus appliquer une fonction sur la clé de l'index

\*\* -- ancien code (2 occurrences) : WHERE stecaisse IN (1, 2)

\*\* -- remplacé par (2 occurrences) : WHERE stecaisse IN ('1', '2')

\*\*/

UPDATE lbm\_vente\_poste v

SET topgu = 2

WHERE **stecaisse IN ('1', '2')**

AND appl = 'SAP'

AND ROWNUM <= p\_rownum

AND hca IS NULL

AND departement = p\_dep

AND ( (typegest != '5' AND typegest != '6' AND p\_typegestion = 'F')

OR ((typegest = '5' OR typegest = '6') AND p\_typegestion = 'G')

)

AND topgu = 0

AND numcdeclt IS NULL

AND TRUNC (datehtrans) =

(SELECT TRUNC (datehtrans)

FROM (SELECT datehtrans

FROM lbm\_vente\_poste v LEFT JOIN lbm\_cp\_histo h

ON v.cp = h.cp

WHERE **stecaisse IN ('1', '2')**

AND v.appl = 'SAP'

AND v.hca IS NULL

AND departement = p\_dep

AND ( ( v.typegest != '5' AND v.typegest != '6' AND p\_typegestion = 'F' )

OR ( (v.typegest = '5' OR v.typegest = '6') AND p\_typegestion = 'G' )

)

AND topgu = 0

AND ( (datehtrans < h.dateblocageinventaire)

OR (h.dateblocageinventaire IS NULL)

AND numcdeclt IS NULL

)

ORDER BY datehtrans)

WHERE ROWNUM <= 1)

AND ( (datehtrans < (SELECT dateblocageinventaire

FROM lbm\_cp\_histo h

WHERE h.cp = v.cp))

OR ((SELECT dateblocageinventaire

FROM lbm\_cp\_histo h

WHERE h.cp = v.cp) IS NULL)

);

Cette correction ne sera cependant pas transcendante puisque la colonne lbm\_vente\_poste.stecaisse n’est pas suffisamment discriminante. Or la seconde colonne dans la clé d’index, lbm\_vente\_poste.datehtrans qui devrait rendre le recours à l’index plus efficace subit aussi une fonction TRUNC. Il faudra évaluer la validité de la réécriture proposée ci-dessous pour s’assurer que la requête conserve sa valeur fonctionnelle.

UPDATE lbm\_vente\_poste v

SET v.topgu = 2

WHERE v.stecaisse IN ('1', '2')

AND v.appl = 'SAP'

AND ROWNUM <= p\_rownum

AND v.hca IS NULL

AND v.departement = p\_dep

AND ( (v.typegest != '5' AND v.typegest != '6' AND p\_typegestion = 'F')

OR ((v.typegest = '5' OR v.typegest = '6') AND p\_typegestion = 'G')

)

AND v.topgu = 0

AND v.numcdeclt IS NULL

PK\_LBM\_240108

PK\_LBM\_281207

PK\_LBM\_BKP\_06112014

AND exists

(SELECT 1

FROM lbm\_vente\_poste w

LEFT JOIN lbm\_cp\_histo h

ON w.cp = h.cp

WHERE w.stecaisse IN ('1', '2')

and w.datehtrans = v.datehtrans

AND w.appl = 'SAP'

AND w.hca IS NULL

AND w.departement = p\_dep

AND ( (w.typegest != '5' AND w.typegest != '6' AND p\_typegestion = 'F')

OR ((w.typegest = '5' OR w.typegest = '6') AND p\_typegestion = 'G')

)

AND w.topgu = 0

AND ( (w.datehtrans < h.dateblocageinventaire)

OR (h.dateblocageinventaire IS NULL)

AND w.numcdeclt IS NULL

)

) AND ((v.datehtrans < (SELECT h.dateblocageinventaire

FROM lbm\_cp\_histo h

WHERE h.cp = v.cp

)

)

OR ((SELECT h.dateblocageinventaire

FROM lbm\_cp\_histo h

WHERE h.cp = v.cp

) IS NULL

)

);

#### Procédure STORELAND.PK\_LBM.LBM\_EXTRACTIONS

Lancement des traitements (dans un contexte de lock applicatif) :

1. PRC\_LBM\_VENTE
2. PRC\_LBM\_TypeRemise
3. PRC\_LBM\_TypeAM

#### Procédure STORELAND.PK\_LBM.LBM\_INTEGRATIONCLIENT

Réduire les accès disques et les FULL SCANS

Rectifier le code extrait ci-dessous comme indiqué :

---------------------------------------------------------

-- 3.) mettre a jour LBM\_MESSAGE\_ID si id\_batch est null (nouveaux clients)

-- ou id\_batch est negatif (si plantage trt precedent) avec - nouveau code traitement,

---------------------------------------------------------

/\*

\*\* Remplacement de la ligne : ((NVL(ID\_BATCH\_STL, 0) = 0) OR (ID\_BATCH\_STL < 0))

\*\* par : NOT (NVL(ID\_BATCH\_STL, 0) > 0)

\*\* pour permettre le recours à l’index

\*\*/

UPDATE LBM\_MESSAGE\_ID

SET ID\_BATCH\_STL = LeCodeTraitementBizTalkNegatif

WHERE NOT (NVL(ID\_BATCH\_STL, 0) > 0)

-- ((NVL(ID\_BATCH\_STL, 0) = 0) OR (ID\_BATCH\_STL < 0))

AND TYPEFLUX = 'CLI'

;

#### Procédure STORELAND.PK\_LBM\_CRM.LBM\_GETCUSTOMERSFORRCU

#### Procédure STORELAND.PK\_LBM\_VENTES\_GU.LBM\_GETTICKETSBM

### Storland.exe

Une requête tout aussi consommatrice en temps et ressource se trouve dans STORLAND.EXE. Elle nécessite les mêmes modifications que celles décrites dans le paragraphe précédent, savoir,

1. Changer le type en varchar2 des éléments énumérés dans stecaisse IN ('1', '2')
2. Ne plus appliquer de fonction au champ datehtrans

SELECT COUNT(\*)

FROM lbm\_vente\_poste v

WHERE **stecaisse IN (1, 2)**

AND appl = 'SAP'

AND hca IS NULL

AND departement = '741'

AND (v.typegest != '5' AND v.typegest != '6')

AND topgu = 0

AND numcdeclt IS NULL

AND **TRUNC (datehtrans)** =

(SELECT TRUNC (datehtrans)

FROM (SELECT datehtrans

FROM lbm\_vente\_poste v2

LEFT JOIN lbm\_cp\_histo h

ON v2.cp = h.cp

WHERE **stecaisse IN (1, 2)**

AND v2.appl = 'SAP'

AND v2.hca IS NULL

AND departement = '741'

AND (v2.typegest != '5' AND v2.typegest != '6')

AND topgu = 0

AND ( (datehtrans < h.dateblocageinventaire)

OR (h.dateblocageinventaire IS NULL)

AND numcdeclt IS NULL

)

ORDER BY datehtrans

)

WHERE ROWNUM <= 1

)

AND ( (datehtrans < (SELECT dateblocageinventaire

FROM lbm\_cp\_histo h

WHERE h.cp = v.cp

)

)

OR ((SELECT dateblocageinventaire

FROM lbm\_cp\_histo h

WHERE h.cp = v.cp

) IS NULL

)

;

Une solution alternative consiste à créer un index comportant des fonctions (function-based index). Voir le paragraphe 5.6.1Table LBM\_VENTE\_POSTE-Ajouter des index.

# Annexes

## Script de compilation des objets en mode debug

Declare

cVc\_owner constant varchar2 (30) := 'STORELAND' ;

begin

for laChose in (select name, type

from all\_plsql\_object\_settings

where type in ( 'PACKAGE'

, 'PACKAGE BODY'

, 'PROCEDURE'

, 'FUNCTION'

, 'TRIGGER'

)

and owner = cVc\_owner

and plsql\_debug = 'TRUE'

order by type asc, name asc

) loop

begin

case laChose.type

when 'PACKAGE' then

execute immediate 'alter package ' || laChose.name || ' compile' ;

when 'PACKAGE BODY' then

execute immediate 'alter package ' || laChose.name || ' compile body' ;

when 'PROCEDURE' then

execute immediate 'alter procedure ' || laChose.name || ' compile' ;

when 'FUNCTION' then

execute immediate 'alter function ' || laChose.name || ' compile' ;

when 'TRIGGER' then

execute immediate 'alter trigger ' || laChose.name || ' compile' ;

end case;

exception

when others then

dbms\_output.put\_line('Erreur sur [' || laChose.name || ', ' || laChose.type || '] : ' || sqlerrm);

end;

end loop;

end;

/

## Script de compilation des objets invalides

Declare

cVc\_owner constant varchar2 (30) := 'STORELAND' ;

begin

for laChose in (select object\_name, object\_type

from all\_objects

where object\_type in ( 'PACKAGE'

, 'PACKAGE BODY'

, 'PROCEDURE'

, 'FUNCTION'

, 'VIEW'

, 'TRIGGER'

)

and owner = cVc\_owner

and status = 'INVALID'

order by object\_type asc

) loop

begin

case laChose.object\_type

when 'PACKAGE' then

execute immediate 'alter package ' || laChose.object\_name || ' compile' ;

when 'PACKAGE BODY' then

execute immediate 'alter package ' || laChose.object\_name || ' compile body' ;

when 'PROCEDURE' then

execute immediate 'alter procedure ' || laChose.object\_name || ' compile' ;

when 'FUNCTION' then

execute immediate 'alter function ' || laChose.object\_name || ' compile' ;

when 'VIEW' then

execute immediate 'alter view ' || laChose.object\_name || ' compile' ;

when 'TRIGGER' then

execute immediate 'alter trigger ' || laChose.object\_name || ' compile' ;

end case;

exception

when others then

dbms\_output.put\_line('Erreur sur [' || laChose.object\_name || ', ' || laChose.object\_type || '] : ' || sqlerrm);

end;

end loop;

end;

/

## Compression des tables fragmentées

Les tablespaces STORELAND\_TBL et STORELAND\_IDX étant en mode « auto segment space management », il est possible d’exécuter la commande SHRINK. Elle a l’avantage d’être exécutable base ouverte. L’exemple suivant s’applique à la table STATS\_CLIENT

1. Activer l’option de déplacement des lignes pour les tables à compresser

ALTER TABLE STATS\_CLIENT ENABLE ROW MOVEMENT ;

1. Réorganiser les enregistrements

ALTER TABLE STATS\_CLIENT SHRINK SPACE COMPACT ;

1. Compresser les lignes. Aucune mise à jour n’est possible sur cette table durant l’exécution de cette commande qui devrait cependant être rapide une fois les lignes réorganisées au cours de la seconde étape ci-dessus  
   ALTER TABLE STATS\_CLIENT SHRINK SPACE CASCADE ;
2. Reconstruire les index de la table

ALTER INDEX STATS\_CLIENT\_PK REBUILD ;

ALTER INDEX STATS\_CLIENT\_FK1 REBUILD ;

## Extrait de V$DATABASE

select NAME

, LOG\_MODE

, CONTROLFILE\_TYPE

, CONTROLFILE\_CREATED

, CONTROLFILE\_TIME

, OPEN\_RESETLOGS

, VERSION\_TIME

, OPEN\_MODE

, PROTECTION\_MODE

, PROTECTION\_LEVEL

, REMOTE\_ARCHIVE

, DATABASE\_ROLE

, ARCHIVELOG\_COMPRESSION

, SWITCHOVER\_STATUS

, DATAGUARD\_BROKER

, GUARD\_STATUS

, FORCE\_LOGGING

, PLATFORM\_ID

, PLATFORM\_NAME

, FLASHBACK\_ON

, SUPPLEMENTAL\_LOG\_DATA\_FK

, SUPPLEMENTAL\_LOG\_DATA\_ALL

, DB\_UNIQUE\_NAME

FROM v$database

;

| NAME | LOG\_MODE | CONTROLFILE\_TYPE | CONTROLFILE\_CREATED | CONTROLFILE\_TIME | OPEN\_RESETLOGS | VERSION\_TIME | OPEN\_MODE | PROTECTION\_MODE | PROTECTION\_LEVEL | REMOTE\_ARCHIVE | DATABASE\_ROLE | ARCHIVELOG\_COMPRESSION | SWITCHOVER\_STATUS | DATAGUARD\_BROKER | GUARD\_STATUS | FORCE\_LOGGING | PLATFORM\_ID | PLATFORM\_NAME | FLASHBACK\_ON | SUPPLEMENTAL\_LOG\_DATA\_FK | SUPPLEMENTAL\_LOG\_DATA\_ALL | DB\_UNIQUE\_NAME |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| STOR | NOARCHIVELOG | CURRENT | 30-MAY-14 | 06-oct-17 | NOT ALLOWED | 30-MAY-14 | READ WRITE | MAXIMUM PERFORMANCE | UNPROTECTED | ENABLED | PRIMARY | DISABLED | SESSIONS ACTIVE | DISABLED | NONE | NO | 12 | Microsoft Windows x86 64-bit | NO | NO | NO | STOR |