



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



HISTORIQUE DES MODIFICATIONS

2

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 MATIERES

1 OBJET DE L'AUDIT.....	5
2 ENVIRONNEMENT	6
2.1 OS	6
2.2 RDBMS	6
2.2.1 La base STORPROD	6
2.3 LES SNAPSHOTS STATSPACK.....	7
3 LES POINTS FORTS	8
3.1 BASE EFFICIENTE ET STABLE	8
4 LES POINTS FAIBLES.....	9
4.1 VERSION D'ORACLE NON SUPPORTEE PAR LE CONSTRUCTEUR.....	9
4.2 QUELQUES OBJETS COMPILES EN MODE DEBUG	9
4.3 PRESENCE D'OBJETS INVALIDES	9
5 LES OPPORTUNITES D'AMELIORATION	10
5.1 METTRE EN PLACE UN PLAN DE MIGRATION VERS UNE VERSION EN COURS DE SUPPORT	10
5.2 AUGMENTER LA RAM	10
5.3 COMPILER EN MODE STANDARD LES OBJETS COMPILES EN MODE DEBUG.....	10
5.4 CORRIGER OU SUPPRIMER LES OBJETS INVALIDES.....	10
5.5 TABLES FRAGMENTEES	11
5.6 AJOUTER DES INDEX.....	13
5.6.1 Table LBM_VENTE_POSTE	13
5.7 CORRIGER DES REQUETES CONSOMMATRICES DE RESSOURCES	13
5.7.1 Package STORELAND.PK_LBM_VENTES_GU	13
Storland.exe.....	15
6 ANNEXES.....	17
6.1 SCRIPT DE COMPILATION DES OBJETS EN MODE DEBUG	17
6.2 SCRIPT DE COMPILATION DES OBJETS INVALIDES.....	17
6.3 COMPRESSION DES TABLES FRAGMENTEES	18
6.4 EXTRAIT DE V\$DATABASE	18

1 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

2 ENVIRONNEMENT

2.1 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.

2.2 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.

2.2.1 La base STORPROD

2.2.1.1 Caractéristiques physiques

2.2.1.1.1 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.

2.2.1.1.2 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

Taille (octets)	Fichier
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

2.3 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 perçus dans cet intervalle.

3 LES POINTS FORTS

3.1 BASE EFFICIENTE 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.

4 LES POINTS FAIBLES

4.1 VERSION D'ORACLE NON SUPPORTEE 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).

4.2 QUELQUES OBJETS COMPILES 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.

4.3 PRESENCE 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

5 LES OPPORTUNITES D'AMELIORATION

5.1 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 surcote en cas de nécessité de support par Oracle. Il y a aussi le risque de devoir migrer d'urgence en cas de dysfonctionnement important.

5.2 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.

5.3 COMPILER EN MODE STANDARD LES OBJETS COMPILES EN MODE DEBUG.

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.

5.4 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

5.5 TABLES FRAGMENTEES

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

Table	Enregistrements	Blocs	Fragmentée (ko)	Données (ko)	Espace perdu (ko)	Espace récupérable %
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

5.6 AJOUTER DES INDEX

5.6.1 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)) ;
```

2. 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.

5.7 CORRIGER DES REQUETES CONSOMMATRICES DE RESSOURCES

5.7.1 Package STORELAND.PK_LBM_VENTES_GU

5.7.1.1 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.numcdec1t 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
    )
);

```

5.7.1.3 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'
;

```

5.7.1.4 Procédure STORELAND.PK_LBM_CRM.LBM_GETCUSTOMERSFORRCU

5.7.1.5 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 numcdec1t 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 5.6.1 Table LBM_VENTE_POSTE-Ajouter des index.

6 ANNEXES

6.1 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;
  /

```

6.2 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;
/

```

6.3 COMPRESSION DES TABLES FRAGMENTEES

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 ;
2. Réorganiser les enregistrements
ALTER TABLE STATS_CLIENT SHRINK SPACE COMPACT ;
3. 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 ;
4. Reconstruire les index de la table
ALTER INDEX STATS_CLIENT_PK REBUILD ;
ALTER INDEX STATS_CLIENT_FK1 REBUILD ;

6.4 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  
;
```

DB_UNIQUE_NAME	STOR
LL	NO
K	NO
FLASHBACK_ON	NO
PLATFORM_NAME	Microsoft Windows x86 64-bit
PLATFORM_ID	12
FORCE_LOGGING	NO
GUARD_STATUS	NONE
DATAGUARD_BROKER	DISABLED
SWITCHOVER_STATUS	SESSIONS ACTIVE
ARCHIVELOG_COMPRESSION	DISABLED
DATABASE_ROLE	PRIMARY
REMOTE_ARCHIVE	ENABLED
PROTECTION_LEVEL	UNPROTECTED
PROTECTION_MODE	MAXIMUM PERFORMANCE
OPEN_MODE	READ WRITE
VERSION_TIME	30-MAY-14
OPEN_RESETLOGS	NOT ALLOWED
CONTROLFILE_TIME	06-oct-17
CONTROLFILE_CREATED	30-MAY-14
CONTROLFILE_TYPE	CURRENT
LOG_MODE	NOARCHIVELOG
NAME	STOR