OSP
Otto Group Solution Provider

Sicheres Identifizieren von nicht relevanten Indizes

Peter Ramm, OSP Dresden

Dezember 2020

Otto Group Solution Provider (OSP) GmbH

www.osp.de

Gründung:
März 1991

Muttergesellschaft:
OTTO Group

Standorte:
Dresden, Hamburg, Burgkunstadt, Bangkok, Taipeh, Madrid

Mitarbeiterzahl:
Ca. 220

Geschäftsführer:
Dr. Stefan Borsutzky, Norbert Gödicke, Jens Gruhl

zur Person

Peter Ramm

Teamleiter strategisch-technische Beratung bei OSP Dresden

> 25 Jahre Historie in IT-Projekten

Schwerpunkte:

- Entwicklung von OLTP-Systemen auf Basis von Oracle-Datenbanken
- Architektur-Beratung bis Trouble-Shooting
- Performance-Optimierung bestehender Systeme

Warum Definition von Indizes auf Tabellen?

OSP
Otto Group Solution Provider

1. Optimieren von Zugriffen aus User-SQL

- Reduktion des Zugriffes auf die ergebnisrelevanten Records beim Lesen von Tabellendaten

2. Garantie der Einhaltung von Eindeutigkeitsregeln

- Deklaration von Unique Indizes bzw. Nutzen des Index für Primary Key bzw. Unique Constraints

3. Absicherung von Foreign Key Constraints

- Verhindern von Full-Table-Scans beim Delete oder Update auf den referenzierten Tabellen
- Verhindern der Propagierung von DML-Locks der referenzierten Tabelle über Foreign Key Constraints

4. Sichern Strukturidentität für Partition Exchange

- Tabelle für Partition Exchange muss identisch indiziert sein wie partitionierte Zieltabelle

Warum Entfernen von nicht relevanten Indizes?

OSP
Otto Group Solution Provider

Wenn ein Index keine der vier vorgenannten Rollen erfüllt, kann er eigentlich auch entfernt werden!

Was bringt die Entfernung von unnötigen Indizes?

- Reduktion von Storage-Bedarf der Datenbank
- Reduktion des Aufwandes für Index-Maintenance bei DML-Operationen
- Effizientere Nutzung des DB-Cache
- Verbesserung des Laufzeit- und Antwortzeitverhaltens von Applikationen

Warum findet trotzdem oftmals keine aktive Bewertung und Entfernung unnötiger Indizes statt?

- Es gibt keine Rolle im Projekt/Produkt, die diese Aufgabe wahrnimmt
  - DBA mit begrenztem Einblick in die Fachlichkeit, Entwickler mit fehlenden Mitteln zur Bewertung
- Never touch a running system: Risiko die Finger zu verbrennen und Betriebsstörungen zu verursachen

Kann ein Index entfernt werden?
Erfordert Bewertung der 4 möglichen Rollen.

OSP
Otto Group Solution Provider

Rolle 1: Optimierung von Zugriffen aus User-SQL

- Eindeutiges Ausschließen der Nutzung eines Index in User-SQL für einen längeren Zeitraum

Rolle 2: Garantie der Einhaltung von Eindeutigkeitsregeln

- Sicherstellen, dass Index NonUnique ist und nicht für Primary Key bzw. Unique Constraint genutzt wird

Rolle 3: Absicherung von Foreign Key Constraints

- Ausschließen, dass Index für Absicherung eines Foreign Key benutzt wird oder
- Sicherstellen, dass für den konkreten Foreign Key keine Absicherung durch Index nötig ist

Rolle 4. Identische Indizierung für Partition Exchange

- Test auf strukturidentische Tabellen, die potentiell für Partition Exchange genutzt werden könnten

OSP
Otto Group Solution Provider

Bewertung der Rolle 1

Nutzung durch User-SQL

Identifikation der Index-Nutzung durch User-SQL

OSP
Otto Group Solution Provider

ALTER INDEX xyz Monitoring Usage; aktiviert die Überwachung der Nutzung des Index durch DB

- Status auswertbar über v$Object_Usage (nur aktuelles Schema) oder sys.Object_Usage (global)
- Es wird nur Nutzung des Index in direkten SQLs protokolliert. Die Nutzung eines Index in rekursiven SQLs der DB selbst wird nicht protokolliert.
  D.h., die implizite Nutzung eines Index für den Lookup beim Delete auf einer referenzierten Tabelle wird nicht erfasst.
- Kein messbarer Performance-Impact durch Aktivierung des Monitoring Usage, aber:
  Ausführung von ALTER INDEX .. MONITORING USAGE als DDL führt zum erneuten Parsen von betroffenen Cursoren bei der nächsten Ausführung
- Erneute Ausführung von ALTER INDEX .. MONITORING USAGE setzt den Usage-Status wieder zurück

Für eine permanente Überwachung eines Systems ist das zyklische Zurücksetzen des Usage-Status wesentlich, um so auch Indizes zu erkennen, die zwar in der Vergangenheit genutzt wurden, aber seit geraumer Zeit nicht mehr

Alternative / Erweiterung zur Identifikation der Index-Nutzung durch User-SQL ab Release 12.2

OSP
Otto Group Solution Provider

Der View DBA_Index_Usage enthält ab Rel. 12.2 detailliertere Informationen zur Index-Nutzung

- Muss nicht gesondert aktiviert werden, protokolliert die Nutzung von Indizes per Default
- Enthält Zeitpunkt der letzten Nutzung sowie Mengenverteilung von Zugriffen und Ergebnismenge

Einschränkungen der Nutzbarkeit dieses Feature:

- Durch nur zyklisches Sampling der Daten besteht ein latentes Risiko, dass real stattgefundene Nutzungen eines Index nicht protokolliert werden
- Das Umstellen auf harte Protokollierung jeden Zugriffs bringt signifikante Performance-Einschränkung ALTER SESSION SET "_iut_stat_collection_type"=ALL; statt SAMPLED
- Die Memory-Daten werden alle 15 Minuten auf Disk geflusht und dann in DBA_Index_Usage sichtbar
- Es wird nur Nutzung des Index in direkten SQLs protokolliert. Die Nutzung eines Index in rekursiven SQLs der DB selbst wird nicht protokolliert (analog zu ALTER INDEX .. MONITORING USAGE).
- Eine Analyse des Index (z.B. dbms_stats.gather_index_stats) zählt auch als Nutzung

Scripte zum permanenten Tracking des Nutzungs-Status per MONITORING USAGE

OSP
Otto Group Solution Provider

- Zurücksetzen des Usage-Status nur für Indizes, die:
  - bereits als genutzt gekennzeichnet sind
  - seit mehr als x Tagen nicht zurückgesetzt wurden
  - nicht in aktuellen Execution-Plänen in der SGA vorkommen (außer bei erster Ausführung)
- Damit Reduktion des Risikos von Parse-Bursts durch Ausführung dieses Scriptes

- Zwei Versionen für Setzen bzw. Rücksetzen des Usage-Status
  1. Für das aktuelle Schema des angemeldeten Users
  2. Für alle Schemata der DB mit Ausnahme der System-Schemata
- Scripte sind abgelegt unter:
  https://rammpeter.blogspot.com/2017/10/oracle-db-identify-unused-indexes.html

Auswertung des Usage-Status von Indizes

OSP
Otto Group Solution Provider

Die Auswertung des Usage-Status von Indizes ist möglich über:

- v$Object_Usage, jedoch nur für die Indizes des Schemas des aktuell angemeldeten Users
- sys.Object_Usage für eine globale Sicht auf die gesamte DB, jedoch eher unkomfortabel

Für beide Varianten der Auswertung finden sich die SQLs unter:
https://rammpeter.blogspot.com/2017/10/oracle-db-identify-unused-indexes.html

Das freie Analyse-Tool Panorama bietet eine Auswertung über sys.Object_Usage verbunden mit:

- Rolle 2: Nutzung der Indizes für Absicherung von Eindeutigkeiten
- Rolle 3: Nutzung der Indizes für Absicherung von Foreign Keys
- Rolle 4: Evtl. Nutzbarkeit des Index für Partition Exchange

Die Folgeseite zeigt einen Beispiel-Screenshot

Rasterfahndung nach Performance-Bottlenecks und Nutzung von Anti-Pattern

Auswahl des Rasterfahndungs-SQL

Filter: Include description Search

- 1. Potential in DB-Strukturen
  - 1. Sicherstellen optimaler Storage-Parameter für Indizes
  - 2. Identifikation eventuell unnötiger Indizes
    - 1. Identifikation nicht für Zugriffe oder Absicherung von Eindeutigkeit benutzte Indizes
    - 2. Identifikation von Indizes mit nur einem oder wenigen Schlüssel-Werten im Index
    - 3. Identifikation von Indizes mit doppelt indizierte Spalten
    - 4. Erkennung nicht genutzter Indizes durch MONITORING USAGE
    - 5. Erkennung von Indizes ohne MONITORING USAGE
    - 6. Identifikation von Indizes mit unnötigen Spalten mit geringer Selektivität
    - 7. Absicherung von Foreign Key-Beziehungen durch Indizes (Ermitteln möglicherweise unnötiger Indizes)
    - 8. Indizes auf partitionierten Tabellen mit selben Spalten wie Partition-Keys
    - 9. Entfernbare Indizes, wenn die Spaltenreihenfolge eines anderen mehrspaltigen Index geändert werden kann
    - 10. Tabellen mit einspaltigem Primary Key Constraint, der nie über Foreign Key Constraints referenziert wird
    - 11. Nicht genutzte Indizes laut DBA_Index_Usage (ab Release 12.2)
  - 3. Empfehlungen für Partitionierung
  - 4. Ermittlung ungenutzter Tabellen oder Spalten
  - 5. Materialized Views
- 6. Empfehlungen für evtl. sinnvolle OLTP-Komprimierung von Tabellen
- 7. Absicherung von Foreign Key-Beziehungen durch Indizes (Ermittlung evtl. fehlender Indizes)
- 8. Tabellen mit PCT_FREE > 0 aber ohne Update-DML
- 9. Relevanz von chained rows im Vergleich zu Gesamtmenge der Zugriffe
- 10. Ermittlung von chained rows von Tabellen

Erkennung nicht genutzter Indizes durch MONITORING USAGE

Die Selektion zeigt durch MONITORING USAGE überwachte Indizes, die seit x Tagen nicht durch SQLs genutzt wurden.
Ein rekursiver Index-Lookup während Foreign Key-Prüfung zählt dabei bzgl. MONITORING USAGE nicht als Nutzung.
Daher enthält die Liste auch Indizes die ausschliesslich zur Absicherung von Foreign Key Constraints genutzt werden.

Die Abfrage erlaubt die Bewertung der vier Gründe für die Existenzberechtigung eines Index:
1. Nutzung durch SQL-Statements: Dann ist Index nicht in der Liste enthalten.
2. Nutzung für Absicherung Eindeutigkeit durch Unique Index, Unique oder Primary Key Constraints (Spalte „uniqueness“).
3. Nutzung für Absicherung eines Foreign Key Constraints (Verhindern Lock-Propagierung und Full-Scan auf Detail-Table bei Delete auf Master-Table).
Die zusätzlichen Informationen der Liste erlauben die Bewertung der Notwendigkeit eines Index für die Absicherung eines Foreign Key Constraints:
Eventuell existierende Foreign Key Constraints sowie Anzahl Rows und DML-Operationen seit letzter Analyse der referenzierten Tabelle.
4. Identität der Index-Strukturen der beteiligten Tabellen für Partition Exchange (Spalte „Partition exchange possible“).
Zeigt die Existenz weiterer strukturidentischer Tabellen, mit denen theoretisch Partition Exchange stattfinden könnte.

Wenn keiner der vier Gründe die Existenz wirklich erfordert, kann der Index risikofrei entfernt werden.

Schema-Name (optional)	
Tage rückwärts ohne Nutzung	7
Minimale Größe des Index in MB	1
Maximale Anzahl DML-Operationen auf der referenzierten Tabelle	100
Maximale Anzahl Rows der referenzierten Tabelle	10000
Zeige auch Unique Indizes (YES/NO)	NO

Selektion ausführen | Show SQL

Erkennung nicht genutzter Indizes durch MONITORING USAGE: 'Tage rückwärts ohne Nutzung' = 7, 'Minimale Größe des Index in MB' = 1, 'Maximale Anzahl DML-Operationen auf der referenzierten Tabelle' = 100, 'Maximale Anzahl Rows der referenzierten Tabelle' = 10000, 'Zeige auch Unique Indizes (YES/NO)' = NO

owner	table_name	index_name	index columns	start monitoring	days without usage	num. rows	distinct keys	avg. rows per key	compression	*mbytes*	uniqueness	foreign key protection	referenced table	num rows of referenced table	last analyze referenced table	inserts on ref. since anal.	updates on ref. since anal.	deletes on ref. since anal.	partition exchange possible	tablespace
BUYING	TEXT	IX_TEXT_TTYCD	ID_TEXTTYPE, CODE, ID	09.08.2019 15:01:25	93,3	135.668.565	135.668.565	1	DISABLED	7.101,6	NONUNIQUE	FK_TEXT_TEXTTYPE	syspeks. TEXTTYPE	16	09.02.2015 08:03:18	1	0	0		INDEX01
EINKAUF	BEW_A_WK_N_I1	FK_BAWNI_K_KDF	BDF_KZ, KDF_KZ	13.09.2019 14:52:32	58,3	189.313.847	92	2.057.759	ENABLED (2)	4.068,3	NONUNIQUE	FK_BWAWKNIT_KKD	einkauf. K_KDF	375	10.11.2019 05:09:52					INDEX01

Menü anzeigen

OSP
Otto Group Solution Provider

Bewertung der Rolle 2

Nutzung für Garantie von Eindeutigkeits-Regeln

OSP
Otto Group Solution Provider

Bewertung der Rolle 3

Absicherung von Foreign Key Constraints

Relevanz des Index für Absicherung Foreign Key

OSP
Otto Group Solution Provider

Gründe für einen Index auf der referenzierenden Tabelle eines Foreign Key Constraints sind:

- Verhindern von Full Table Scan auf der referenzierenden Tabelle bei Delete oder Update auf der Referenz
- Verhindern von Lock-Propagierung von DML-Locks bei DML auf der referenzierten Tabelle

D.h. auch bei Existenz eines Foreign Key Constraints ist die Absicherung mit einen Index nicht notwendig, wenn auf der referenzierten Tabelle kein DML stattfindet.

Dies ist regelmäßig der Fall z.B. bei Foreign Key Constraints auf statischen Stammdatentabellen.

Tolerierbarkeit von DML auf referenzierter Tabelle

OSP
Otto Group Solution Provider

Selbst wenn auf der referenzierten Tabelle in homöopathischer Menge DML stattfindet, kann oftmals auf eine Absicherung mit Index verzichtet werden:

- Ein Full Table Scan auf der referenzierenden Tabelle beim Delete auf der Referenz kann evtl. billigend in Kauf genommen werden für sporadische DML-Aktivitäten.
  Einmaliger Full Table Scan alle x Wochen ist allemal weniger aufwändig als Vorhalten eines Index.
- Lock-Propagierung stellt nur in besonderen Konstellationen ein Problem dar.
  Insbesondere dann, wenn Update auf Primärschlüsselspalten stattfindet, wofür insbesondere bei Verwendung von technischen IDs keine Notwendigkeit besteht.

In u.g. Blog-Post findet sich für zwei über Foreign Key Constraint verbundene Tabellen :

- Eine Aufstellung verschiedener konkurrierender DML-Operationen auf beiden Tabellen
- Die jeweilige Relevanz eines Index zum Verhindern von Blocking Locks

https://rammpeter.blogspot.com/2016/11/clarify-myths-of-indexing-foreign-key.html

Prüfung der Existenz von DML auf Referenz

Otto Group Solution Provider

Das Vorhandensein und die Größenordnung von DML auf einer Tabelle kann nachvollzogen werden über den View DBA_Tab_Modifications.

In diesem View werden seit dem Zeitpunkt der letzten Analyse die Anzahl von Inserts, Updates und Deletes protokolliert.

Voraussetzung für die Aufzeichnung in DBA_Tab_Modifications ist laut Oracle-Dokumentation bis zu 19c der Status MONITORING=YES auf Table-Ebene.
In der Realität werden aber ab Rel.11 DML-Operationen auch bei Status NOMONITORING aufgezeichnet.

Diese Kennwerte erlauben die Bewertung, ob die Häufigkeit und Frequenz der DML-Operationen auf der referenzierten Tabelle die Absicherung eines Foreign Key Constraint auf der referenzierenden Tabelle durch Index erfordern.

Schritte in Panorama für Check DML auf Referenz
1. Auswahl Index-Ansicht aus Tabellenbeschreibung

OSP
Otto Group Solution Provider

- Die Spalte „Use“ beschreibt Nutzungsstatus sowie Start bzw. letzten Reset des Monitoring
- Die Spalte „FK“ bietet einen Link auf Details zum Foreign Key Constraint

Attributes of TABLE WHR.INVENTORYITEMINFOHISTORY

TS	Pct Free	Init. Trans	Initial extent (KB)	Rows	Size (MB) Table	Size (MB) Total	Ext.	Blocks	% unused	Empty	RowLen	Part.	Sub-Part.	Created	Last DDL	Spec. TS	Last analyzed	Log.	Mon.	Inserts	Updates	Deletes	Trunc.	Drop seg.	Last DML	Several States	RAC mas in
DATA01	10	8	64	311,317,439	34,970.25	54,517.88	2,685	4,476,192	1.5	0	116	0	0	2015-03-08 00:44:28	2019-05-01 08:29:33	2017-12-03 02:21:54	2019-05-04 20:48:16	NO	YES	600,141	0	14,640,417	NO	0	2019-05-06 08:15:56	Compression = ENABLED (ADVANCED)	1

4 Indexes | 0 Primary Key | 0 Check Constraints | 11 References from | 0 References to | 0 Triggers | 4 Dependencies | 38 Grants | DBMS_METADATA | DB-Cache | Sessions accessing | SQLs | Active Session History | Segment statistics | Size evolution

Indexes of WHR.INVENTORYITEMINFOHISTORY

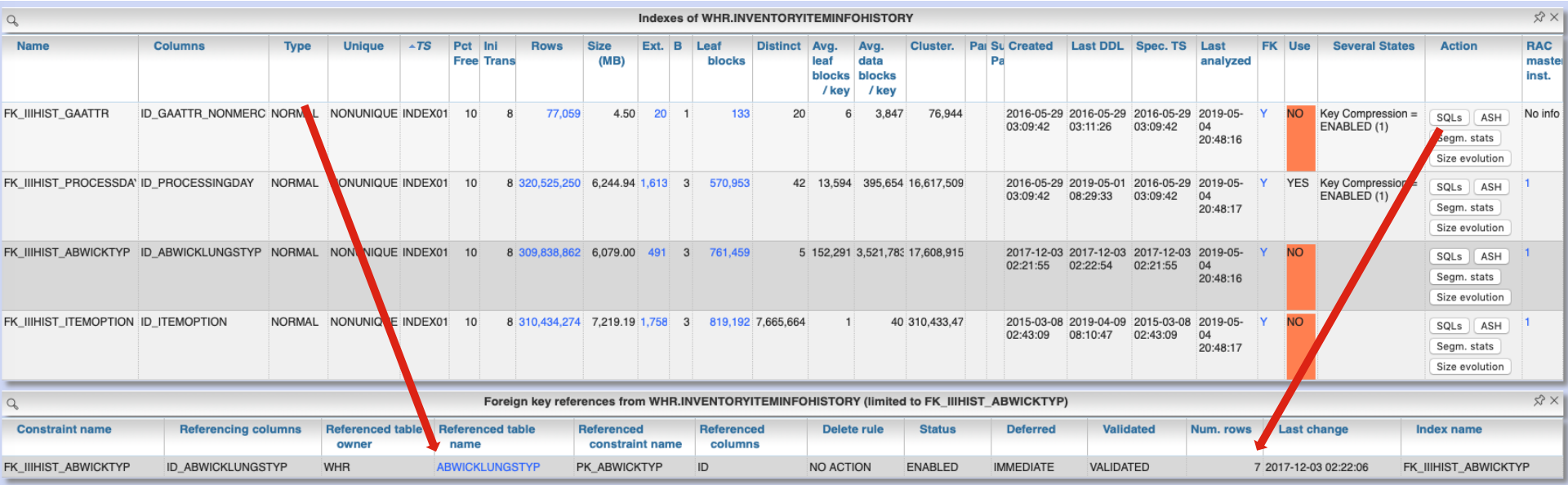
Name	Columns	Type	Unique	TS	Pct Free	Ini Trans	Rows	Size (MB)	Ext.	B	Leaf blocks	Distinct	Avg. leaf blocks / key	Avg. data blocks / key	Cluster.	Pa	Su Pa	Created	Last DDL	Spec. S	Last analyzed	FK	Use	Several States	Action	RAC master inst.
FK_IIIHIST_GAATTR	ID_GAATTR_NONMERC	NORMAL	NONUNIQUE	INDEX01	10	8	77,059	4.50	20	1	133	20	6	3,847	76,944			2016-05-29 03:09:42	2016-05-29 03:11:26	2016-05-29 03:09:42	2019-05-04 20:48:16	Y	NO	Key Compression = ENABLED (1)	SQLs ASH Segm. stats Size evolution	No info
FK_IIIHIST_PROCESSDA	ID_PROCESSINGDAY	NORMAL	NONUNIQUE	INDEX01	10	8	32,0											2016-05-29 03:09:42	2016-05-29 08:29:33	2016-05-29 03:09:42	2019-05-04 20:48:16	Y	YES	Key Compression = ENABLED (1)	SQLs ASH Segm. stats Size evolution	1
FK_IIIHIST_ABWICKTYP	ID_ABWICKLUNGSTYP	NORMAL	NONUNIQUE	INDEX01	10	8	309,838,862	6,079.00	491	3	761,459	5	152,291	3,521,783	17,608,915			2017-12-03 02:21:55	2017-12-03 02:22:54	2017-12-03 02:21:55	2019-05-04 20:48:16	Y	NO		SQLs ASH Segm. stats Size evolution	1
FK_IIIHIST_ITEMOPTION	ID_ITEMOPTION	NORMAL	NONUNIQUE	INDEX01	10	8	310,434,274	7,219.19	1,758	3	819,192	7,665,664	1	40	310,433,47			2015-03-08 02:43:09	2019-04-09 08:10:47	2015-03-08 02:43:09	2019-05-04 20:48:17	Y	NO		SQLs ASH Segm. stats Size evolution	1

- MouseOver auf Spalte „Use“ zeigt Start des monitoring usage
- Klick in Spalte „Use“ zeigt Daten aus DBA_Index_Usage (ab 12.2)

Schritte in Panorama für Check DML auf Referenz
2. Details zum Foreign Key Constraint

OSP
Otto Group Solution Provider

- Num. Rows der referenzierten Tabelle erlaubt erste Bewertung der möglichen Relevanz des Index
- Link in Spalte „Referenced table name“ bringt Details zu referenzierter Tabelle



Indexes of WHR.INVENTORYITEMINFOHISTORY

Name	Columns	Type	Unique	TS	Pct Free	Ini Trans	Rows	Size (MB)	Ext.	B	Leaf blocks	Distinct	Avg. leaf blocks / key	Avg. data blocks / key	Cluster.	Pa	Su Pa	Created	Last DDL	Spec. TS	Last analyzed	FK	Use	Several States	Action	RAC master inst.
FK_IIIHIST_GAATTR	ID_GAATTR_NONMERC	NORMAL	NONUNIQUE	INDEX01	10	8	77,059	4.50	20	1	133	20	6	3,847	76,944			2016-05-29 03:09:42	2016-05-29 03:11:26	2016-05-29 03:09:42	2019-05-04 20:48:16	Y	NO	Key Compression = ENABLED (1)	SQLs ASH Segm. stats Size evolution	No info
FK_IIIHIST_PROCESSDA	ID_PROCESSINGDAY	NORMAL	NONUNIQUE	INDEX01	10	8	320,525,250	6,244.94	1,613	3	570,953	42	13,594	395,654	16,617,509			2016-05-29 03:09:42	2019-05-01 08:29:33	2016-05-29 03:09:42	2019-05-04 20:48:17	Y	YES	Key Compression = ENABLED (1)	SQLs ASH Segm. stats Size evolution	1
FK_IIIHIST_ABWICKTYP	ID_ABWICKLUNGSTYP	NORMAL	NONUNIQUE	INDEX01	10	8	309,838,862	6,079.00	491	3	761,459	5	152,291	3,521,783	17,608,915			2017-12-03 02:21:55	2017-12-03 02:22:54	2017-12-03 02:21:55	2019-05-04 20:48:16	Y	NO		SQLs ASH Segm. stats Size evolution	1
FK_IIIHIST_ITEMOPTION	ID_ITEMOPTION	NORMAL	NONUNIQUE	INDEX01	10	8	310,434,274	7,219.19	1,758	3	819,192	7,665,664	1	40	310,433,47			2015-03-08 02:43:09	2019-04-09 08:10:47	2015-03-08 02:43:09	2019-05-04 20:48:17	Y	NO		SQLs ASH Segm. stats Size evolution	1

Foreign key references from WHR.INVENTORYITEMINFOHISTORY (limited to FK_IIIHIST_ABWICKTYP)

Constraint name	Referencing columns	Referenced table owner	Referenced table name	Referenced constraint name	Referenced columns	Delete rule	Status	Deferred	Validated	Num. rows	Last change	Index name
FK_IIIHIST_ABWICKTYP	ID_ABWICKLUNGSTYP	WHR	ABWICKLUNGSTYP	PK_ABWICKTYP	ID	NO ACTION	ENABLED	IMMEDIATE	VALIDATED	7	2017-12-03 02:22:06	FK_IIIHIST_ABWICKTYP

Schritte in Panorama für Check DML auf Referenz
3. Details zum Foreign Key Constraint

OSP
Otto Group Solution Provider

- Tabellendetails der referenzierten Tabelle mit DML-Events seit dem Zeitpunkt der letzten Analyse
- Anzahl Inserts / Updates /Deletes / Truncates erlauben Bewertung der Relevanz eines Index

Foreign key references from WHR.INVENTORYITEMINFOHISTORY (limited to FK_IIIHIST_ABWICKTYP)

Constraint name	Referencing columns	Referenced table owner	Referenced table name	Referenced constraint name	Referenced columns	Delete rule	Status	Deferred	Validated	Num. rows	Last change	Index name
FK_IIIHIST_ABWICKTYP	ID_ABWICKLUNGSTYP	WHR	ABWICKLUNGSTYP	PK_ABWICKTYP	ID	NO ACTION	ENABLED	IMMEDIATE	VALIDATED	7	2017-12-03 02:22:06	FK_IIIHIST_ABWICKTYP

Description of TABLE WHR.ABWICKLUNGSTYP

Neue Stammtabelle für Abwicklungstypen

Columns of TABLE WHR.ABWICKLUNGSTYP

Col.Name	Type	Prec.	Sc.	N.	Def.	Distinct	Nulls	Avg. Len.	Density	Buckets	Histogram	Comments	LOB segment	EQ	EQJ	NEQJ	Range	Like	Null
ID	NUMBER	2	0	N		7	0	3	0.1429	1	NONE	Eindeutige ID, Primäschlüssel			16	0	0	0	0
NAME	VARCHAR2	50 Chars		Nullable		7	0	10	0.1429	1	NONE	Name des Abwicklungstyps, etwa OSS, EIGENWARE, DLW,...							
DESCRIPTION	VARCHAR2	200 Chars		N		7	0	43	0.1429	1	NONE	Beschreibung des Abwicklungstyps							
ENUMNAME	VARCHAR2	30 Chars		N		7	0	6	0.1429	1	NONE	EnumName für die Generierung von Konstanten							

Attributes of TABLE WHR.ABWICKLUNGSTYP

TS	Pct Free	Init. Trans	Initial extent (KB)	Rows	Size (MB) Table	Size (MB) Total	Ext.	Blocks	% unused	Empty	RowLen	Part.	Sub-Part.	Created	Last DDL	Spec. TS	Last analyzed	Log.	Mon.	Inserts	Updates	Deletes	Trunc.	Drop seg.	Last DML	Several States	RAC master inst.
DATA01	10	8	64	7	0.06	0.13	1	8	99.3	0	61	0	0	2017-12-03 02:21:54	2019-05-01 08:29:06	2017-12-03 02:21:54	2017-12-03 02:22:53	YES	YES								< 0 different >

1 Indexes | 1 Primary Key | 0 Check Constraints | 0 References from | 3 References to | 0 Triggers | 1 Dependencies | 13 Grants | DBMS_METADATA | DB-Cache | Sessions accessing | SQLs | Active Session History | Segment statistics | Size evolution

OSP
Otto Group Solution Provider

Bewertung der Rolle 4

Struktur-Identität für Partition Exchange

Identische Indizierung für Partition Exchange

OSP
Otto Group Solution Provider

Vor Entfernung eines Index sollte ausgeschlossen werden, dass der Index nur existiert für die Erfüllung der Vorbedingungen für Partition Exchange

Bedingungen für eventuelle Nutzung für Partition Exchange sind zwei oder mehr Tabellen mit:

- identischer Tabellenstruktur bzgl. Typ, Länge und Position der Spalten, Name kann differieren
- Identische Index-Struktur bzgl. Anzahl der Indizes sowie Struktur (Anzahl und Position der Spalten)
- Mindestens eine der strukturidentischen Tabellen ist partitioniert
- Mindestens eine der strukturidentischen Tabellen ist nicht partitioniert

Die Spalte „partition exchange possible“ der Rasterfahndungs-Abfrage zeigt die Existenz potentieller Kandidaten für Partition Exchange

- Ein SQL zur Selektion der konkreten potentiellen Kandidaten für Partition Exchange findet sich hier: https://rammpeter.blogspot.com/2019/11/oracle-db-list-tables-suitable-for.html

OSP
Otto Group Solution Provider

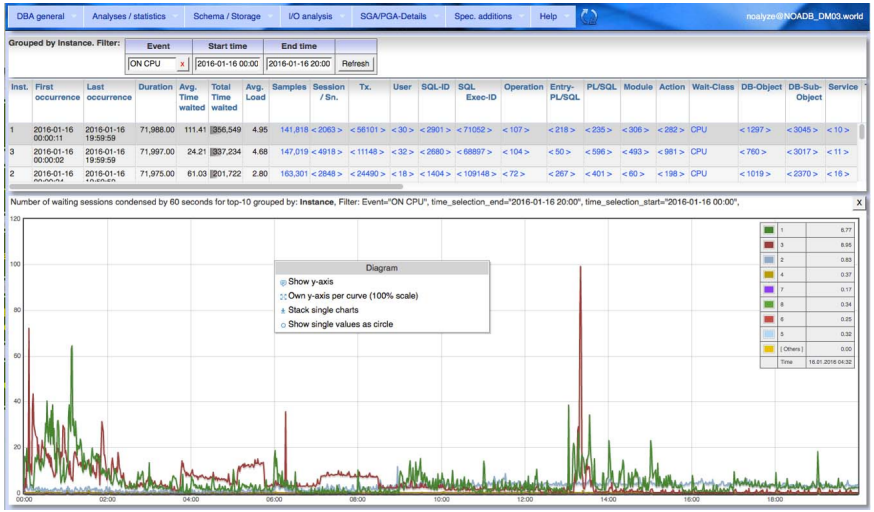
Verwendetes Analyse-Werkzeug

Panorama for Oracle databases

OSP
Otto Group Solution Provider

Freies verfügbares Werkzeug für Performance-Analyse von Oracle-DB

Basierend u.a. auf AWR-Daten oder eigenem Sampling (auch mit SE bzw. ohne Diagnostics Pack anwendbar)



Panorama greift nur rein lesend auf Ihre Datenbank zu und installiert keine eigenen PL/SQL-Objekte.
Sie können die Funktionen also ohne Risiko testen und verstehen.

Beschreibung von Panorama inkl. Download-Link: http://rammpeter.github.io
Blog zum Oracle-Performance-Analyse: https://rammpeter.blogspot.com
Diverse Slides zum Anwendung von Panorama: Link zu slideshare.net

Weitere Quellen:
- Blog-Post zu Erkennen ungenutzter Indizes
- Blog-Post zu Relevanz der Absicherung von Foreign Key Constraints durch Indizes

OSP
Otto Group Solution Provider

Weitere Möglichkeiten zur Reduktion von Indizes

Gründe für Entfernbarkeit real genutzter Indizes

OSP
Otto Group Solution Provider

Trotz nachgewiesener Nutzung durch User-SQL kann es trotzdem sinnvoll sein, einen Index zu entfernen

Spalten sind bereits anderweitig durch mehrspaltige Indizes indiziert

- Nach Entfernung eines Index übernimmt ein weiterer Index mit den Spalten des ersten die Funktion

Partitionierung hat gleiche Filterwirkung wie Index

- Reduktion der Treffermenge durch Partitionierung ist effektiver als Index-Zugriff, wenn Partitionierungskriterium eindeutig ist

Zugriff per Index Fast Full Scan kann auch über anderen Index ausgeführt werden

- Wenn die relevante Spalte(n) auch anderweitig indiziert ist oder nur die Anzahl selbst relevant ist (z.B. für COUNT(*)), dann kann alternativ auch ein anderer Index verwendet werden

Gründe für Entfernung real genutzter Indizes
Bewertung per Panorama (Rasterfahndung)

OSP
Otto Group Solution Provider

Rasterfahndung nach Performance-Bottlenecks und Nutzung von Anti-Pattern

Auswahl des Rasterfahndungs-SQL

Filter: Include description Search

- 1. Potential in DB-Strukturen
  - 1. Sicherstellen optimaler Storage-Parameter für Indizes
    - 1. Sicherstellen Index PCTFree >= 10
    - 2. Empfehlungen für Index-Komprimierung, Test auf Selektivität
    - 3. Empfehlungen für Index-Komprimierung, Test auf Leaf-Blocks
    - 4. Empfehlungen für Index-Komprimierung, Test auf Selektivität einer einzelnen Spalte eines Mehrspaltenindex
    - 5. Redundanzfreie Ablage von Daten im Primary Key-Index (Umwandlung in Index-organized Tables)
  - 2. Identifikation eventuell unnötiger Indizes
    - 1. Identifikation nicht für Zugriffe oder Absicherung von Eindeutigkeit benutzte Indizes
    - 2. Identifikation von Indizes mit nur einem oder wenigen Schlüssel-Werten im Index
    - 3. Identifikation von Indizes mit doppelt indizierte Spalten
    - 4. Erkennung nicht genutzter Indizes durch MONITORING USAGE
    - 5. Erkennung von Indizes ohne MONITORING USAGE
    - 6. Identifikation von Indizes mit unnötigen Spalten mit geringer Selektivität
    - 7. Absicherung von Foreign Key-Beziehungen durch Indizes (Ermitteln möglicherweise unnötiger Indizes)
    - 8. Indizes auf partitionierten Tabellen mit selben Spalten wie Partition-Keys
  - 3. Empfehlungen für Index-Partitionierung
    - 1. Local-Partitionierung für NonUnique-Indizes
    - 2. Local-Partitionierung von Unique Indizes mit Partition-Key=Index-Spalte
    - 3. Local-Partitionierung mit Overhead im Zugriff

Erkennung nicht genutzter Indizes durch MONITORING USAGE

Die DB protokolliert die Nutzung (Zugriff) von Indizes die vorab durch 'ALTER INDEX ... MONITORING USAGE' deklariert wurden.
Das Ergebnis der Nutzungsprotokollierung ist je Schema in Tabelle v$Object_Usage einsehbar.
Schemaübergreifend ist die Nutzung mit diese Selektion sichtbar.
Achtung:
- Rekursiver Index-Lookup während Foreign Key-Prüfung zählt nicht als Nutzung in v$Object_Usage.
- Bitte Vorsicht falls Index nur benötigt wird zur Absicherung der Foreign Key-Beziehung (verhindern Full-Scan auf Detail-Table bei Delete auf Master-Table).
- Auch die Ausführung von GATHER_INDEX_STATS zählt als Nutzung, selbst wenn der Index nie durch andere Selects genutzt wurde. (nicht mehr festgestellt ab DB-Version >= 12)

Weitere Informationen zur Nutzung bringt Anzeige 'Segment statistics historisch' und Analyse der ActiveSessionHistory über Spalte 'Time waited'.

Schema-Name (optional)	
Tage rückwärts ohne Nutzung	7
Minimale Größe des Index in MB	1
Maximale Anzahl DML-Operationen auf der referenzierten Tabelle	
Zeige auch Unique Indizes (YES/NO)	NO

Selektion ausführen Show SQL

Maßnahmen zur Reduktion der Größe von Indizes: Reduktion des Index auf relevante Records

OSP
Otto Group Solution Provider

Nur wenige Werte einer indizierten Spalte sind als Zugriffskriterium relevant

Beispiel: Tabelle mit 300 Mio. Records besitzt eine Spalte „Status“ mit den beiden Ausprägungen:

- ‚P‘ = Processed: bereits verarbeiteten Records (299.999.900 Records)
- ‚N‘ = New: neu hinzugekommenen noch zu verarbeitenden Records (max. 100 Records)

Variante 1:

Index auf “Tabelle(Status)“

Zugriff per:

```
SELECT * FROM Tabelle
WHERE Status = ‘ N‘
```

Größe des Index: ca. 3 Gigabyte

Variante 2:

Index auf “Tabelle(DECODE(Status, ‘ N‘, 1))“

Zugriff per:

```
SELECT * FROM Tabelle
WHERE DECODE(Status, ‘ N‘, 1)) = 1
```

Größe des Index: 1 DB-Block = 8 Kilobyte

Lösung: Function based Index reduziert die Größe des Index um Faktor 375000

Maßnahmen zur Reduktion der Größe von Indizes: Komprimierung

OSP
Otto Group Solution Provider

Seit Oracle 9i ist das Feature Index Key Compression Bestandteil ab Standard Edition

- Basiert auf Deduplizierung identischer Werte innerhalb eines Index-Blocks
- Zu erzielende Reduktionen bei entsprechender Redundanz der indizierten Werte liegen bei 1/3 bis 1/2 der ursprünglichen Größe des Index
- I.d.R. ergibt sich neben Reduktion Storage auch ein positiver Performance-Effekt durch Reduktion von I/O und bessere Cache Hit Rate
- CREATE INDEX … >COMPRESS oder CREATE INDEX xxx COMPRESS(n)

Advanced Index Compression seit Oracle 12, Bestandteil der Advanced Compression Option

- Index Block-Komprimierung auch ohne redundante Werte im Index
- CREATE INDEX … COMPRESS ADVANCED LOW ab Oracle 12.1
- CREATE INDEX … COMPRESS ADVANCED HIGH ab Oracle 12.2

Vielen Dank für Ihr Interesse

Otto Group Solution Provider (OSP) Dresden GmbH
Freiberger Str. 35 | 01067 Dresden
T +49 (0)351 49723 0 | F +49 (0)351 49723 119
osp.de