



Christian Trieb  
Leiter Datenbank Community

Liebe Mitglieder der DOAG Deutsche ORACLE-Anwendergruppe,  
liebe Leserinnen und Leser,

das Jahr 2012 endet mit den ersten Informationen zur neuesten Version der Oracle-Datenbank, die auf der OpenWorld in San Francisco bekannt gegeben worden sind (siehe Seite 7). Diese wird erst im kommenden Jahr verfügbar sein. Dies ist umso verständlicher, als mit der Container-Datenbank zum ersten Mal seit Jahren eine grundlegende Architektur-Änderung vorgenommen wurde. Die Zukunft wird zeigen, wie sich dies in der Praxis auswirkt.

Diese Ausgabe der DOAG News bietet Ihnen eine Vielzahl an Best Practices für Ihre Oracle-Datenbank. Verschiedene Autoren schildern ihre Erfahrungen im täglichen Umgang damit. Der Themenbogen reicht von der Verwaltung über Backup & Recovery sowie Optimierung bis hin zum Betriebssystem. Da dieses Mal sehr viele Artikel eingereicht wurden, erscheinen die Beiträge zu den Themen „Tuning“ und „Hochverfügbarkeit“ erst in der nächsten Ausgabe.

Während ich dieses Editorial schreibe, steht die DOAG 2012 Konferenz + Ausstellung noch bevor und damit auch die Entscheidung über die neue DOAG-Satzung. Ich hoffe, dass, wenn Sie dies lesen, die Mitglieder der neuen Satzung zugestimmt haben und die DOAG auf einem guten Weg ist, die bisher erfolgreichen Aktivitäten in den Communities fortzusetzen und auszubauen.

Ich wünsche Ihnen gesegnete Weihnachten und ein gutes neues Jahr 2013.

Ihr

ORACLE Platinum Partner

**HUNKLER**  
GmbH & Co. KG

„ Best Solutions based on Oracle,  
von einem der führenden  
Oracle-Systemhäuser in Deutschland “

LIZENZBERATUNG &  
-VERTRIEB



HOCHVERFÜGBAR-  
KEITSLÖSUNGEN &  
PERFORMANCE  
TUNING



DATA WAREHOUSING &  
BUSINESS  
INTELLIGENCE  
LÖSUNGEN



ORACLE  
APPLIANCES



## HUNKLER – die erste Adresse beim Thema Oracle

Ausfallsichere Datenbanken, professionelle Lösungen für Business Intelligence, leistungsstarke Appliances: Auf diese Schwerpunkte haben wir uns nach den von Oracle vorgegebenen Anforderungen spezialisiert. Spezialisten für Oracle sind wir schon seit 1987, als wir erster offizieller Partner in Deutschland wurden.

Wir wissen genau, was der Mittelstand wirklich braucht: modernste Technologie,

zugeschnitten auf individuelle Business-Lösungen, die sofort Kosten senken. Lösungen, mit denen Unternehmen von Anfang an spürbare Wettbewerbsvorteile erzielen und langfristig festigen können.

Von der Systemplanung bis zum Lizenzmanagement. Es gibt immer den richtigen Weg zu mehr Effizienz in der IT. Bei uns. Für Sie.

Hauptsitz Karlsruhe

Bannwaldallee 32, 76185 Karlsruhe, Tel. 0721-490 16-0, Fax 0721-490 16-29  
info@hunkler.de, www.hunkler.de

Geschäftsstelle Bodensee

Fritz-Reichle-Ring 6a, 78315 Radolfzell, Tel. 07732-939 14-00, Fax 07732-939 14-04  
info@hunkler.de, www.hunkler.de

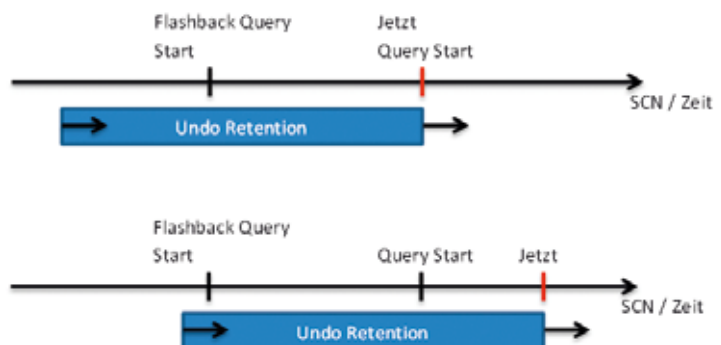
- 3 Editorial  
*Christian Trieb*
- 5 Spotlight
- 6 Die Highlights der Oracle OpenWorld 2012  
*Johannes Fries, ORACLE Deutschland B.V. & Co. KG*
- 8 „Sicherheit fängt bei der Bewusstseinsbildung an ...“  
*Interview mit Hans-Peter Bauer, McAfee GmbH*
- 11 Neu: Cloud Control 12c Release 2  
*Ralf Durben*
- 13 Effizienzsteigerung mittels Standardisierung  
*Christian Wischki und Rainer Hartwig*
- 18 „OFA“ – wozu brauche ich das denn?  
*Stefan Panek*
- 22 Oracle Database Advisors  
*Ulrike Schwinn*
- 26 Critical Patch Update – wie kritisch es wirklich ist  
*Katja Werner*
- 32 Oracle Linux versus Red Hat – lohnt sich der Umstieg auf Oracle Linux?  
*Andrew Lacy*
- 35 Sind wir eigentlich ganz dicht?  
*Eero Mattila*
- 39 Flashback: Recovery ohne Recovery  
*Marco Mischke*
- 43 DATE und TIMESTAMP in der Praxis  
*Carsten Czarski*

**Entwicklung**

- 47 Grundlagen zum Cost-Based Optimizer  
*Randolf Geist*
- 52 Manuelles Oracle SQL-Tuning  
*Martin Decker*
- 58 SQL Plan Management unter der Lupe  
*Jan Krüger*

**DOAG intern**

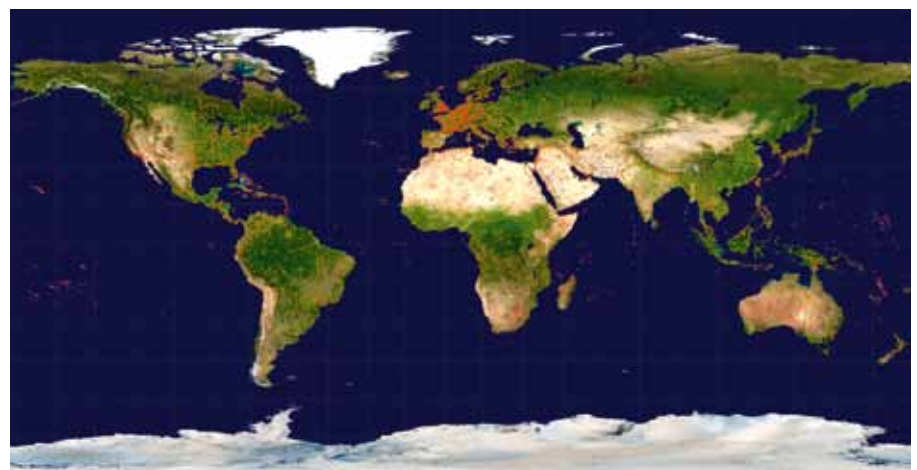
- 21 Impressum
- 38 Inserentenverzeichnis
- 63 Wir begrüßen unsere neuen Mitglieder
- 64 Aus dem Verein
- 65 Vorschau
- 66 DOAG-Termine



*Zeitliche Zusammenhänge von Flashback Query und Undo, Seite 39*



*Advisors im Enterprise Manager helfen weiter, Seite 22*



*SQL Plan Management am Beispiel der weitweiten Flughäfen, Seite 58*



### Sonntag, 30. September 2012

*Die DOAG-Delegation erlebt Larry Ellison bei seiner Eröffnungs-Keynote zur diesjährigen Oracle OpenWorld. Dieser startet gleich voll durch und präsentiert die vier Leit-Innovationen der Veranstaltung: Infrastructure-as-a-Service, Private Cloud, die neue Datenbank-Version 12c sowie die vierte Generation der Oracle-Appliances.*

### Montag, 1. Oktober 2012

*Im Rahmen der Keynote von Oracle-President Mark Hurd stellt Andrew Mendelsohn, Senior Vice President Oracle Database Server Technologies, erste Neuigkeiten zur Datenbank 12c vor. Das Highlight der neuen Version, die im Laufe des Jahres 2013 auf den Markt kommen soll, ist ein Container, der die verschiedenen Datenbanken eines Unternehmens enthält. Diese können dort besser bezüglich der Datenbank-Ressourcen administriert werden und sind einfacher zu patchen als bisher.*

### Dienstag, 2. Oktober 2012

*Mehrere DOAG-Vorstände und Christian Trieb, Leiter des Competence Centers Supportfragen, treffen im Rahmen der Oracle OpenWorld Charles Rozwat, Leiter des weltweiten Oracle Supports, und präsentieren die Support-Umfrage der DOAG. Rozwat sichert der DOAG seine Hilfe zu bei der Suche nach Referenzkunden, die den Oracle Configuration Manager einsetzen. Seitens der DOAG bietet der Vorstandsvorsitzende Dr. Dietmar Neugebauer an, im Rahmen der DOAG-Aktivitäten die Kommunikation rund um die Support-Tools zu fördern. Am nächsten Tag stellt Christian Trieb beim Treffen des International Oracle User Council Global Support Committee fest, dass die Support-Probleme kein deutsches Phänomen sind, sondern die Problematik in anderen Ländern ähnlich gelagert ist.*

### Montag, 15. Oktober 2012

*Dr. Dietmar Neugebauer, Vorstandsvorsitzender der DOAG, und Fried Saacke, DOAG-Vorstand und Geschäftsführer, besprechen bei einem Besuch bei dem DOAG-Mitglied WWK-Versicherung deren Anregungen zur geplanten Satzungsreform. Dabei wurde vereinbart, in den Beschlussvorschlag des Vorstands zur Satzungsänderung eine Regelung aufzunehmen, nach der die Mitgliederversammlung auch künftig in Satzungsänderungen eingebunden wird. Am Tag darauf beschließt die DOAG-Leitung im Rahmen einer Telefonkonferenz einstimmig die finale Vorlage, die der Mitgliederversammlung am 20. November 2012 zur Abstimmung vorgelegt wird.*

### Dienstag, 16. Oktober 2012

*Die SIG Security findet in Köln statt, gefolgt von der SIG Database am nächsten Tag. Beide Veranstaltungen waren sehr konstruktiv. Dr. Dietmar Neugebauer, Vorstandsvorsitzender der DOAG, beide Male vor Ort, fiel besonders der rege Austausch der Teilnehmer untereinander und mit den Referenten auf.*

### Mittwoch, 24. Oktober 2012

*Dr. Dietmar Neugebauer, Vorstandsvorsitzender der DOAG, Fried Saacke, DOAG-Vorstand und Geschäftsführer, sowie Wolfgang Taschner, Chefredakteur der DOAG News, vertreten die DOAG auf dem Oracle Day 2012 in München. Neben den Zeitschriften der DOAG findet auch der Flyer zur DOAG 2012 Konferenz + Ausstellung reißenden Absatz.*

### Montag, 12. November 2012

*Dr. Dietmar Neugebauer, Vorstandsvorsitzender der DOAG, und Fried Saacke, DOAG-Vorstand und Geschäftsführer, besuchen die Gründungsveranstaltung der Regio Leipzig/Halle. Sie haben mit Matthias Reimann und Volker Weise zwei engagierte Mitglieder gefunden, die sich jetzt um die neue Regionalgruppe kümmern.*

### Dienstag, 13. November 2012

*Noch eine Woche bis zur Eröffnung der DOAG 2012 Konferenz + Ausstellung in Nürnberg. Die Teilnehmerliste verspricht einen neuen Besucherrekord. Dr. Dietmar Neugebauer, Vorstandsvorsitzender der DOAG, und Fried Saacke, DOAG-Vorstand und Geschäftsführer, freuen sich über die zahlreichen Stammgäste und Erstbesucher und sind überzeugt, dass die Veranstaltung wieder ein voller Erfolg wird.*

Mehr als 50.000 Besucher machten sich in diesem Jahr auf den Weg nach San Francisco, um Neuigkeiten rund um die Oracle-Produktwelt zu erfahren und sich mit Oracle-Experten auszutauschen.

# Die Highlights der Oracle OpenWorld 2012

Johannes Fries, ORACLE Deutschland B.V. & Co. KG

Neben den lange erwarteten Ankündigungen zum Thema „Oracle Cloud“ standen auch neue Exadata- und Exalogic-Maschinen im Mittelpunkt des Interesses: Mit der Oracle Exadata X3 Database Machine und der Oracle Exalogic X3 wurde nun jeweils eine neue Generation der Oracle Appliances ins Rennen geschickt. Die Oracle-Exadata-X3-Systeme eignen sich hervorragend sowohl für High-End OLTP-Anwendungen als auch für sehr große Data Warehouses und natürlich für Database Clouds.

Die Exadata X3 Database Machine ist eine Kern-Komponente der Oracle Cloud. Sie kann Hunderte von Terabytes komprimierter Daten im Flash und RAM-Memory speichern. Mit dem Ziel „höchste Performance bei möglichst niedrigen Kosten“ hält die Oracle Exadata X3 Database In-Memory Machine automatisch alle aktiven Daten im Flash- und im RAM-Speicher.

Die Oracle Exalogic X3 – die zweite Hardware-Generation der Exalogic-Hardware für Business Applications – stellt in Verbindung mit der Oracle Elastic Cloud Software eine Symbiose von Hard- und Software dar, die zusammen als „Oracle Exalogic Elastic Cloud“ extreme Performance, Verfügbarkeit und Skalierbarkeit für Java-Anwendungen, Oracle Applications und Oracle Fusion Middleware bereitstellt.

In dem neuen System kommen innovativste Hardware-Bausteine wie die neuesten Intel-Core-Xeon-Prozessoren mit 33 Prozent mehr Cores pro Konfiguration zum Einsatz. Das neue, sogenannte „Trusted Partitions Licensing Modell“ sorgt für flexible Lizenzierungsmöglichkeiten der Oracle-Software. Hinzu kommt, dass Oracle mit dem Achtel-Rack jetzt ein kostengünstiges Modell für den Einstieg in die Welt der Oracle Exa\*-Systeme anbietet.

## Die Oracle Cloud

Ein weiteres Highlight der OpenWorld 2012 war die Vorstellung der Oracle Cloud. Sie bietet zukünftig ein umfangreiches Portfolio an Services an. Neben den Oracle-Datenbank- und Java-Cloud-Services haben Kunden zukünftig auch die Möglichkeit, auf ein breites Angebot von Standardanwendungen zuzugreifen. Die Webseite [cloud.oracle.com](http://cloud.oracle.com) zeigt die vielseitigen Möglichkeiten der Oracle Cloud.

Der Oracle WebLogic Server 12c stellt nicht nur die Grundlage für die klassische Anwendungswelt oder für die bereits oben erwähnte Exalogic-Elastic-Cloud-Systeme dar, sondern auch für die neuen Oracle-Java-Cloud-Services. Darüber erhält der Benutzer der Oracle Cloud im Rahmen einer Self-Service-Umgebung die Möglichkeit, effizient zu entwickeln, zu testen und letztendlich auch zu betreiben – auch in Form von Erweiterungen zu Oracle Cloud Application Services.

## BPM Suite Process Accelerators

Auch die Entwicklungswerkzeuge bieten einige Neuigkeiten. Insbesondere die Process Accelerators der BPM Suite sind erwähnenswert, denn sie senken in BPM-Projekten durch einen standardisierten und vorkonfigurierten Einstieg Projekt-Risiken und ermöglichen einen einfacheren und effizienteren Dialog mit den Fachanwendern. Im Bereich der horizontalen Anwendungen kann man beispielsweise auf Module wie „Travel Request Management“, „Document Routing and Approval“ oder „Internal Service Request“ im Rahmen von Projekten zurückgreifen. Speziell für die Finanzindustrie gibt es das Paket „Loan Origination“ und für die öffentliche Verwaltung das Modul „Incident Reporting“.

## Oracle ADF Mobile

Mit dem Wandel der Clients von den schwergewichtigen PCs zu den Handhelds in Form von Smartphones und Tablets wird die Bedeutung der effizienten und sicheren Anbindung dieser neuen Generation von „Smart Devices“ sowohl für die Entwicklung – in Form von transparenter Unterstützung im Entwicklungsprozess und auf der Werkzeugebene – als auch im Betrieb immer wichtiger. Als Erweiterung des Oracle Application Development Frameworks (ADF) steht der Entwickler-Community hier „ADF Mobile“ zum Download zur Verfügung. HTML5- und Java-basierend ermöglicht dieses Framework die Entwicklung hybrider Mobile-Applikationen für verschiedene Plattformen auf einer Codebasis – wie heute bereits für iOS und Android.

Mit Oracle ADF Mobile wird nicht nur die Basis für eine unternehmensweit einheitliche Infrastruktur für mobile Anwendungen geschaffen, sondern das Thema „Sicherheit“ an zentraler Stelle und über End-to-End-Security und Encryption verankert. In ADF Mobile kommen verbreitete Web-Entwicklungstechnologien wie HTML5, JavaScript und CSS zum Einsatz. Diese ermöglichen den Zugriff auf die Kamera, das GPS-Modul oder die Kontakte eines Device. Eine „Lightweight JVM“ steht als Plattform für Java-Anwendungen zur Verfügung.

## Fazit

Neben vielen technologischen Highlights gewinnen mittlerweile spezifische Themen an Gewicht, die sich wiederum in den neuen Generationen der Oracle-Produkte widerspiegeln. Dies wird auch durch eine wachsende Zahl von Kundenvorträgen deutlich. Erwähnenswert ist hier das Thema „Customer

Experience". Die Diskussionen und Vorträge im Rahmen der OpenWorld 2012 rund um das Thema „Customer Experience“ verdeutlichen einen Wechsel im Denken und in der Perspektive. Der Mensch und seine Spuren in den Systemen rücken in den Mittelpunkt. Mit diesem Shift beginnen sich die Anforderungen an IT-Werkzeuge und IT-Technologien zu verändern. „Empower the Information Worker“ ist ein trefender Begriff für die Anforderungen, die an die IT zukünftig gestellt werden.

Das Verhalten und die Spuren der Benutzer stellen wichtige Informatio-

nen für Entscheidungen dar und müssen im richtigen Kontext in die Anwendungen einfließen. Aber auch die Zusammenarbeit innerhalb einer Organisation ist ein wichtiger Einflussfaktor für Geschäftsprozesse und deren Verlauf. Gemeinsame Arbeitsbereiche und Arbeitsergebnisse werden zukünftig selbstverständlicher Bestandteil von Geschäftsprozessen sein. Oracle BPM und die Werkzeuge aus dem Bereich Oracle BI und Oracle WebCenter sind Plattform und Kitt zugleich für die heutigen und zukünftigen Anforderungen aus der Business-Welt. Die

OpenWorld 2012 hat die technologische Basis für die Anforderungen an die zukünftige Anwendungswelt gelegt. 2013 werden wir sicher die ersten Ergebnisse präsentiert bekommen.



Johannes Fries  
johannes.fries  
@oracle.com

## Erste Infos über die Datenbank 12c

„Die Pluggable Database ist nur eine Oracle-Datenbank“ fasst Tom Kyte in seinem Vortrag „Tom's Top 12 Things About the Latest Generation of Database“ zusammen. Vielleicht listet der Oracle-Architekt aus diesem Grund erst mal die neuen Features der Oracle Database 12c auf, bevor er abschließend auf die neue Architektur eingeht. Als er das Thema anspricht, zieht die Mehrheit der Teilnehmer ihre Tablets und Smartphones heraus, um die paar wertvollen Folien seiner Präsentation zu fotografieren. Die Pluggable Datenbank ist der unausgesprochene Hype der diesjährigen Oracle OpenWorld. DOAG Online hat sich kundig gemacht und präsentiert die ersten Features.

Wie es in der Praxis aussehen wird, ist den meisten noch nicht bekannt. Doch in der Theorie klingt alles so einfach und handlich, dass es partout auf der Oracle OpenWorld Begeisterung auslöst: Eine Datenbank zum Ein- und Ausstöpseln und Hin- und Herschieben. Aber was bedeutet es wirklich?

In den bisherigen Datenbank-Versionen war das Oracle Data Dictionary zusammen mit Tables, Packages und anderen Komponenten gespeichert. Dies ändert sich nun mit der Version 12c grundsätzlich. Die Datenbank kommt im neuen Gewand daher und wird mandantenfähig. Im Klartext ist es ein einfaches Prinzip: Das Root-Repository ist nun getrennt von dem Rest der Datenbank.

Man nehme einen Container, in dem sich ein Root-Repository für beliebig viele Datenbanken befindet. Das ist die Ausgangssituation. Neue Datenbanken können ganz einfach in demselben Container erstellt werden. Doch

es geht weiter: Nehmen wir an, dass ein DBA eine bereits existierende Datenbank klonen möchte. Mit dem Kommando CREATEPDB wird eine XML-Datei erzeugt, die für die Erstellung der neuen Datenbank genutzt wird. Das Gleiche gilt in Zukunft dann auch für Upgrades: Zu diesem Zweck muss ein neuer Container erstellt werden, der das neue Root-Repository beinhaltet. Wenn dieser Schritt erledigt ist, können alle Datenbanken auf einmal aktualisiert werden, indem sie in den neuen Container geschoben werden. Diese Mandantenfähigkeit der Version 12c bringt große Ressourcenvorteile mit sich – und dies bei gleichbleibender Performance: Während 50 gewöhnliche Datenbanken 20 GB Speicherplatz benötigen, kommen 50 Pluggable Datenbanken mit 3 GB zurecht. Entsprechendes gilt für die CPU-Leistung: 27 Prozent für die Pluggable Datenbanken gegen 36 Prozent für die ältere Generation. Christian Trieb, Leiter der DOAG Datenbank Community, meint dazu: „Dass sich die CPU-Leistung wirklich so verhält, beruht allerdings auf Aussagen von Oracle und muss erst noch verifiziert werden. Auch ob Migration und Patching in der Zukunft wirklich so einfach gehen, wie Oracle es darstellt, wird die Praxis in den nächsten Jahren zeigen. Es ist aber zu begrüßen, das Oracle sich dieser immer wichtiger werdenden Herausforderung stellt.“

Immer mehr Intelligenz wandert in die Datenbank. Dies macht sich beispielsweise an Features wie „Life Cycle Management“ bemerkbar, mit dem Statistiken über die Datennutzung geführt werden. Je nach Nutzung werden die Daten automatisch unterschiedlich komprimiert

und abgelegt. Werden Daten oft abgerufen oder aktualisiert, so werden sie möglicherweise mit einem niedrigen Kompressionsgrad im Flash gespeichert. Werden sie im Gegenteil selten abgefragt, können sie mit höchstem Komprimierungsgrad archiviert werden. In puncto Hochverfügbarkeit erlebt man einen ähnlichen Trend: Bei einem Ausfall sorgt ein Transaction Guard dafür, dass Transaktionen nur einmal abgeschlossen werden. Zudem wird mit dem Feature „Business Continuity“ die Ausfallsicherheit der Applikation gewährleistet.

Auch ein Blick auf Oracle Real Application Cluster (RAC) lohnt sich. Bisher gab es beispielsweise für jeden RAC-Knoten eine einzelne ASM (Automatic Storage Management)-Instanz. Mit der Einführung von Flex ASM hat dies nun ein Ende gefunden. Nun kann die ASM-Instanz für mehrere RAC-Knoten zur Verfügung gestellt werden. Weiter: Oracle ASM Disk Scrubbing erkennt automatisch korrupte Daten und repariert diese. Weitere Vorteile bringt Oracle Utility Cluster mit sich: Dieses Feature funktioniert ähnlich wie der Grid Home Server und macht ein zentrales Management des Patching möglich.



Mylène Diacquenod  
DOAG Online  
redaktion@doag.org



Hans-Peter Bauer (links) im Gespräch mit Dr. Dietmar Neugebauer

Fotos: Wolfgang Taschner

Security ist ein wichtiger Faktor in den Unternehmen. Dr. Dietmar Neugebauer, Vorstandsvorsitzender der DOAG, und Wolfgang Taschner, Chefredakteur der DOAG News, sprachen darüber mit Hans-Peter Bauer, Vice President Central und Eastern Europe der McAfee GmbH.

## „Sicherheit fängt bei der Bewusstseinsbildung an ...“

*Welche Unternehmensziele hat McAfee?*

**Bauer:** McAfee ist weltweit der größte Anbieter, der sich rein auf IT-Sicherheit spezialisiert hat. Wir liefern unseren Kunden präventive, praxiserprobte Sicherheits-Lösungen und -Dienstleistungen, um sie vor Angriffen zu schützen. Unterstützt durch die einzigartige Global-Threat-Intelligence-Technologie helfen wir sowohl Privatnutzern als auch Organisationen, ihre Daten zu schützen, einschlägige Gesetze einzuhalten, Störungen zu verhindern, Schwachstellen zu ermitteln und die Sicherheit ihrer Systeme laufend zu überwachen und zu verbessern.

*Bisher ist McAfee nicht als Anbieter von Sicherheitslösungen für Datenbanken erkennbar gewesen. Ihr Unternehmen hat sich jedoch im letzten Jahr in diesem Bereich durch Zukäufe verstärkt. Wo sehen Sie Chancen, diese Sichtbarkeit zu verbessern?*

**Bauer:** Durch die Sentrigo-Akquisition sind Sicherheitslösungen für Daten-

banken ein ganz heißes Thema für uns. Wir haben frühzeitig erkannt, dass IT-Sicherheit sehr viel mit Schutz und Sicherheit der Daten zu tun hat. Sentrigo war schon zuvor Partner unserer Security Innovation Alliance, in deren Rahmen wir auch Datacenter absichern.

*Gibt es aus Ihrer Sicht ein Patentrezept, um globale Sicherheit in einem Unternehmen einzuführen?*

**Bauer:** Unsere Strategie beinhaltet einen globalen Cloud-basierenden Service, der so aussieht, dass wir über sogenannte „Fingerprints“ weltweit alle Sicherheitsvorfälle in Form von neutralisierten Hash-Codes unserer eigenen Sensoren sowie der unserer Kunden – sofern diese dem zustimmen und den Service aktiviert haben – einsammeln und in unserer Cloud (GTI) ablegen. Diesen Service stellen wir unseren Kunden zur Verfügung und deren Systeme können dort eine CGut“- oder „Schlecht“-Antwort in fast „Real-Time“ beziehen.

*Worin sehen Sie die größte Herausforderung bei der Umsetzung einer unternehmensweiten Security-Policy?*

**Bauer:** Ich glaube, dass Security in vielen Unternehmen sowohl technisch als auch organisatorisch sehr stark fragmentiert ist. Da gibt es zum Beispiel einen Chief Security Officer, der aber mit den Netzwerk-Leuten nichts zu tun hat. Zudem sollte dieser Chief Security Officer direkt dem CIO unterstellt sein und Einfluss auf alle Abteilungen innerhalb der IT haben. Außerdem haben die Unternehmen es mittlerweile weniger mit Angriffen zu tun, die Daten zerstören, sondern vielmehr mit Vorfällen, bei denen Daten abgezogen werden.

*Wo sollte der Bereich „Sicherheit“ organisatorisch im Unternehmen angesiedelt sein?*

**Bauer:** In jedem Fall direkt unterhalb des CIOs. Nur so besteht die Möglichkeit, Sicherheit von den Datenbanken über die Netzwerke bis hin zur Produktion zu gewährleisten.

Wo sollte nach Ihrer Meinung ein mittelständisches Unternehmen mit der Sicherheit anfangen?

**Bauer:** Das fängt bei der Bewusstseinsbildung an. Man muss seine gesamte Infrastruktur kennen und deren Schwachstellen analysieren. Dann gilt es, alle Sicherheitssysteme zu synchronisieren und aufeinander abzustimmen. Die Gesamtsicherheit resultiert immer aus der Zusammenarbeit aller Teilsysteme. Das Maturity-Modell bietet hier sehr gute Checklisten für ein Unternehmen.

Was würden Sie einem Sicherheitsbeauftragten in einem Unternehmen raten?

**Bauer:** Er sollte auf keinen Fall ein fragmentiertes Sicherheitssystem betreiben, denn das ist kein zukunftsträchtiger Weg. Er sollte hingegen mit renommierten Sicherheitsunternehmen zusammenarbeiten. Darüber hinaus gibt es einige Beratungsfirmen, die ihm dabei helfen, ein Sicherheits-

konzept zu definieren. In jedem Fall sollte er sich eine integrierte Sicherheitslösung anschaffen, anstatt die Systemintegration selbst zu erledigen.

Wie lassen sich Aufwand und Kosten für die Sicherheit im Verhältnis zum Nutzen im Rahmen halten?

**Bauer:** Das hängt stark davon ab, in welchem Bereich sich ein Unternehmen bewegt. Die wenigsten Kosten fallen an, wenn man nur reaktiv beispielsweise eine Firewall einsetzt. Will man hingegen proaktive Sicherheit – also einen fragmentierten Ansatz – betreiben, fallen darüber hinaus auch Personalkosten und weitere Systeme an, sodass das Budget bei knapp zehn Prozent der gesamten IT-Kosten liegt. Im optimierten, integrierten Level kann man den höchsten Sicherheitsgrad erreichen und die Kosten sollten sich maximal zwischen vier bis sechs Prozent des IT-Budgets bewegen.

Wie stellt man in der Praxis sicher, dass die User nur die Daten sehen, die sie auch sehen sollen?

**Bauer:** Dafür gibt es entsprechende Systeme. Bei uns ist jeder User in seiner Rolle einem bestimmten Profil zugeordnet. Wird nun im Unternehmen ein neues System wie beispielsweise das iPhone eingeführt, übernimmt unser Enterprise-Mobility-Management-System dieses Profil und gewährt jedem Benutzer seine ihm entsprechenden Rechte.

Wann beziehungsweise wie ist ein Monitoring von auffälligen Zugriffen sinnvoll?

**Bauer:** Ein Monitoring von Datenbanken ist zu jeder Zeit wichtig. Dieses Monitoring hilft dem Unternehmen, sich selbst zu definieren, denn niemand hat heute schon die Datenstrukturen, um Missbrauch zu verhindern. Ein weiteres Problem ist die Vielzahl der Datenbanken. Es nutzt nichts, wenn man seine zentrale Datenbank

## KeepTool mit neuer Version 10

Das handliche Werkzeug für Oracle™-Datenbanken



Zahlreiche neue Funktionen, z.B.

- Darstellung Ihrer Daten als Pivottabelle, ggf. mehrstufig.
- Praktische Hinweistexte bei der Datenerfassung.
- Überwachung und Steuerung der Optimizer-Statistiken.
- Data Pump-Schnittstelle.
- Jumplist für den Windows 7™ Taskbar.

Laden Sie die kostenlose Testversion unter [www.keeptool.com](http://www.keeptool.com) herunter.



**keeptool**



**Zur Person:**  
**Hans-Peter Bauer**

Hans-Peter Bauer ist Vice President Central und Eastern Europe bei McAfee. Er wechselte zum 1. Januar 2008 von Juniper Networks, wo er zuletzt als Vice President für das Enterprise-Geschäft in EMEA verantwortlich war. Er bringt eine mehr als 20-jährige Erfahrung in der Computer- und Informationstechnologie-Branche in seine Position ein.

Hans-Peter Bauer begann seine Laufbahn bei der Siemens AG in der Beratung und dem Vertrieb für Daten und Informationssysteme. 1988 wechselte er zu Digital Equipment (DEC), wo er in verschiedenen Management-Positionen im nationalen und internationalen Vertrieb tätig war. 1993 stieg er in das Consulting und Systemintegrations-Business bei der Computer Sciences Corporation (CSC) ein. Dort war Hans-Peter Bauer als Mitglied der Geschäftsleitung verantwortlich für den Geschäftsbereich Kommunikation und Medien in Central und Eastern Europe. 1997 kam Hans-Peter Bauer zu Lotus als Managing Director für Lotus Consulting Central und Eastern Europe und übernahm 1999 die Gesamtverantwortung als Geschäftsführer und General Manager der Lotus Development Deutschland, Österreich und der Schweiz. Nach erfolgreich durchgeführter Integration von Lotus in die IBM war er als Director und zuletzt als Vicepresident Software Group Central Europe der IBM Deutschland GmbH für das gesamte Software-Geschäft im deutschsprachigen Europa verantwortlich. Ab August 2002 übernahm Hans-Peter Bauer als Vicepresident und General Manager die Verantwortung für Symantec Central and Eastern Europe. Nach Abschluss der Akquisition von Veritas durch Symantec wechselte er im April 2005 zu Macromedia als Geschäftsführer für Deutschland, die Schweiz und Österreich. Mit der Übernahme von Macromedia durch Adobe wurde Hans-Peter Bauer zum Managing Director and General Manager Adobe North European Region befördert. Hier verantwortete er das gesamte Produkt- und Lösungsportfolio in den Ländern UK, Irland und Skandinavien.

schützt und sich nicht um die Kopien der Daten auf diversen PCs oder Abteilungsservern kümmert.

*Wie sollte man mit Vorfällen im Unternehmen umgehen?*

**Bauer:** Totschweigen ist die schlechteste Lösung. Für viele Unternehmen ist es sicherlich schwierig, mit Vorfällen offen umzugehen, um das Vertrauen der Partner nicht zu verlieren. Ich empfehle, zunächst forensisch vorzugehen, um den exakten Hergang und den Schaden sowie die damit verbundenen Auswirkungen festzustellen. Daraus lassen sich dann Maßnahmen ableiten, um sich zukünftig davor zu schützen. Jeder Angriff ist immer eine Lehre dafür, wie man sich entsprechend schützen kann.

*Jede Sicherheitsmaßnahme behindert meist in irgendeiner Form den operativen Betrieb. Wie stellt man sicher, dass die beschlossenen Vorhaben auch in den Fachabteilungen wie geplant umgesetzt werden?*

**Bauer:** Dem stimme ich nicht zu. Man kann heute Sicherheit einführen, ohne die Kreativität einzuschränken. Es gilt lediglich, eine gewisse Disziplin



zu wahren und jedem Benutzer nur die Rechte zu gewähren, die er für die Ausführung seiner Arbeit benötigt.

*Wie lässt sich Sicherheit messen?*

**Bauer:** Eine entsprechende Scorecard muss jedes Unternehmen für sich definieren. Sicherheit bedeutet für mich Risiko-Management. Das Unternehmen muss seine Assets kennen und die jeweiligen Risiken entsprechend bewerten.

*Zu viele Sicherheitsmaßnahmen führen zu Misstrauen der Mitarbeiter gegenüber dem Unternehmen. Wie geht man damit um?*

**Bauer:** Es gilt immer, die optimale Sicherheit zu gestalten. Zu viele Sicherheitsmaßnahmen bringen keinem Unternehmen etwas.

*Sicherheit ist immer eine Reaktion auf Bedrohungen. Wird nach Ihrer Ansicht die Lücke zwischen Bedrohungen und Schutzmaßnahmen kleiner oder größer?*

**Bauer:** Wir stellen fest, dass die Komplexität der Bedrohungslage ansteigt. Der Wunsch, dass moderne Software immer fehlerfreier wird, tritt leider nicht ein. Auf der anderen Seite reicht die Bedrohung durch Rootkits bis in den Bootsektor der Rechner, sodass ein einfacher Neustart nicht mehr weiterhilft.

*Was muss Oracle tun, um Unternehmen bei der notwendigen Sicherheit zu unterstützen?*

**Bauer:** Ich wünsche mir von Oracle mehr Offenheit. Wir kommen nur bedingt an alle Informationen heran, um die Sicherheitsfunktionalität zu erfüllen. Darüber hinaus würden wir auch gerne mit Oracle zusammen ein Sicherheitsmodul entwickeln, wie wir es mit SAP bereits für NetWeaver getan haben.

*Wie sehen Sie die Rolle der DOAG in diesem Bereich?*

**Bauer:** Die DOAG kann aufgrund ihrer enormen Reichweite zur Bewusstseinsbildung beitragen. Sie erreicht auch Datenbank-Beauftragte, die üblicherweise sehr weit vom Chief Security Officer entfernt sind.



Seit dem 14. September 2012 steht das neue Release 2 von Oracle Enterprise Manager Cloud Control 12c zur Verfügung. Zum ersten Mal in der Geschichte des Enterprise Manager hat Oracle ein neues Release für alle Komponenten und Plattformen am gleichen Tag freigegeben. Es steht sowohl für den Oracle Enterprise Manager als auch den Agenten für alle unterstützten Plattformen zur Verfügung und ist somit sofort für alle Umgebungen einsetzbar.

## Neu: Cloud Control 12c Release 2

Ralf Durben, ORACLE Deutschland B.V. & Co KG

Oracle Enterprise Manager Cloud Control 12c Release 2 trägt die Versionsnummer 12.1.0.2 und ist vor allem ein Wartungs-Release. Es enthält hauptsächlich Bugfixes und Performance-Verbesserungen sowie Veränderungen in den internen Komponenten, die nach außen nicht sichtbar sind. Es gibt aber auch einige neue Features, die in diesem Artikel kurz zusammengefasst und beschrieben sind.

### Dynamische Gruppen

Gruppen sind schon aus den Vorgänger-Releases bekannt. Damit können verschiedene Targets (Datenbanken, Middleware, Hosts etc.) gemeinsam betrachtet werden. Ein oft genutztes Anwendungsbeispiel ist die Zuordnung von Targets zu den verantwortlichen Administratoren. Targets können zum Beispiel nach ihrem Standort in Gruppen zusammengefasst sein. Die

EM-Benutzer bekommen Zugriff auf diese Gruppen und sehen damit automatisch die für sie relevanten Systeme. Wenn sich die Eigenschaft eines Targets ändert und damit auch die Gruppenzugehörigkeit, muss diese manuell angepasst werden: Eine Datenbank zum Beispiel zieht von Hamburg nach München um und müsste damit auch die Gruppe wechseln. Bei normalen Gruppen ist dieses manuell durchzuführen.



Kompetent – Unabhängig – Erfolgsbasiert

# ORACLE-SOFTWARE

# IST JEDEN CENT WERT!

Unsere Mandanten zahlen trotzdem weniger.  
Sprechen Sie mit uns!

Wir sind nur unseren Mandanten verpflichtet.

- > **Compliance sichern**
- > **Audit vermeiden**
- > **Kosten senken**

**ProLicense GmbH**  
 Friedrichstraße 191 | 10117 Berlin  
 Tel: +49 (0)30 60 98 19 230 | [www.prolicense.com](http://www.prolicense.com)

Die neuen, dynamischen Gruppen im Release 2 übernehmen diese Zuordnung auf der Basis von Target-Eigenschaften. Sie können zum Beispiel eine dynamische Gruppe erstellen, die alle Datenbanken mit der Eigenschaft „Location = Hamburg“ zusammenfasst. Sobald diese Eigenschaft beim Target gesetzt wird, wandert dieses Target in diese Gruppe.

#### Properties für EM-User

Jeder EM-Benutzer kann jetzt Eigenschaften (Properties) haben. Sobald ein EM-Benutzer mit gesetzten Properties ein neues Target anlegt, werden diese Eigenschaften dem neuen Target vererbt. Damit bekommt es auf jeden Fall die wichtigsten Eigenschaften zugeordnet, um zum Beispiel automatisch einer dynamischen Gruppe (siehe oben) zugeordnet zu werden.

#### Verwaltung von Cloud Control

Zur Verwaltung des Cloud Control Frameworks gibt es jetzt einen neuen Menüpunkt „Setup -> Manage Cloud Control“. Dort gelangt man zu den neu gestalteten Seiten für die Verwaltung folgender Punkte:

- OMS
- Repositories
- Agenten

Auf der Seite für das Repository steht jetzt die Information, wie viele Datenbank-Sessions pro Arbeitsbereich (wie Eventsystem, Jobsystem, Loader etc.) aktiv sind. Die Übersicht über alle Agenten bietet vordefinierte Filter (All, Up, Under Blackout, Status Pending, Agent Unreachable, Blocked, Misconfigured, Upgradeable, Not Upgradeable), mit denen in einer großen Liste von Agenten besser gesucht werden kann.

#### Software Library

Die Software Library kommt im Rahmen von Provisionierungen und zum Einspielen von Patches zum Einsatz. Sie enthält die zu provisionierende Software als Komponenten und die dazu gehörigen Installations-Anweisungen in Form von Direktiven. Mit dem neuen Release können jetzt Ver-

zeichnisse, Komponenten und Direktiven in der Software Library auch über das Enterprise Manager Command Line Interface (EMCLI) verwaltet werden. Somit ist eine Integration in andere Systeme möglich. Darüber hinaus ist es jetzt realisierbar, mit einer einfachen Aktion (Stage Operation) Komponenten aus der Software Library auf Zielsysteme zu kopieren, ohne eine Deployment-Prozedur starten zu müssen.

#### Monitoring und Incident Management

Auch im Rahmen des Monitoring und Incident Managements gibt es Neues zu vermelden. Metric Extensions sind benutzerdefinierte Metriken, die mit Cloud Control 12c eingeführt wurden. Ihre interne Implementierung und ihr Funktionsumfang unterscheiden sich von den „user defined Metrics“ in Grid Control und es wird empfohlen, eigene Metriken nur noch in der Form „Metric Extensions“ zu erstellen. Im Cloud Control 12c Release 1 können nur EM-Super-User die Metric Extensions editieren und verwalten. Im Release 2 sind die Privilegien „Edit“ und „Full“ auch für normale EM-Benutzer möglich.

Monitoring Templates geben an, wie ein Zielsystem überwacht werden soll. Ein Monitoring Template wurde in den Vorgängerversionen immer auf Basis eines Referenz-Targets erstellt. Ab Release 2 kann es jetzt auch auf der Grundlage einer Best Practice, also eines Defaults, erstellt und natürlich später justiert werden. Monitoring Templates lassen sich jetzt ex- und importieren und somit von einem Cloud-Control-System zu einem anderen transportieren. Darin enthaltene Metric Extensions werden dabei auch mitbewegt.

Der Incident Manager in Cloud Control zeigt an, wo in den Systemen Probleme bestehen. Im Release 2 bekommt man dabei noch mehr Informationen über die jeweils betroffenen Targets angezeigt und man kann den Status von Incidents jetzt auch in Bulk-Operationen verändern (Clear, Acknowledge etc.). Die in Incident Rules vorgenommene Bewertung eines

Incident berücksichtigt jetzt auch den Lifecycle-Status eines Targets.

#### Cloud Management

Im Rahmen des Cloud Managements gibt es zwei Neuerungen. Man kann jetzt auch Java-Anwendungen als Service in einer privaten Cloud anbieten und die Abrechnung von Ressourcennutzung basiert neuerdings auf Kostenstellen und Charge-Plänen. Wenn diese einem Target zugewiesen wurden, können sie nun auch wieder entzogen werden. Eine RAC-Datenbank lässt sich jetzt als sogenannte „Shared Datenbank“ abrechnen, wobei die Abrechnung auf Basis einzelner Datenbank-Services erfolgt. Die Abrechnungsdaten liegen in einer eigenen Tabelle und können dort über normales SQL ausgelesen werden. Neu ist der Zugriff mit dem Enterprise Manager Command Line Interface (EMCLI) und dem Kommando „get\_metering\_data“.

#### Upgrade-Wege und Fazit

Ein Upgrade ist sowohl von Grid Control 10.2.0.5 oder 11.1 als auch von Cloud Control 12c Release 1 direkt nach Cloud Control 12c Release 2 möglich.

Das neue Cloud Control 12c Release 2 enthält primär viele interne Verbesserungen, aber auch einige interessante neue Features. Es ist gleichzeitig auf allen unterstützten Plattformen erschienen und bietet daher überall den vollen Funktionsumfang.

Ralf Durben  
Ralf.Durben@oracle.com



Standardisierungen sind die Grundlage für jeden qualitätsgesicherten und kosteneffizienten Betrieb von Oracle-Datenbanken.

# Effizienzsteigerung mittels Standardisierung

Christian Wischki und Rainer Hartwig

Egal, wie groß oder klein IT-Abteilungen, die Menge der zu betreibenden Oracle-Datenbanken oder die Zahl der Anwender auch sein mögen, die Definition und Einhaltung von im Configuration Management definierten Standards bietet sowohl für das Unternehmen – in Bezug auf Compliance-Anforderungen unter ITIL und ISO 20000 – und die verantwortlichen Personen als auch für die Entwicklung und den Betrieb der darauf basierenden Applikationen und deren Anwender viele Vorteile. Grundsätzlich ist es dabei im ersten Schritt eher unwichtig, wie genau eine Oracle-Standard-Datenbank aussieht beziehungsweise aussehen soll. Das am Anfang Wichtigste ist, dass ein Standard definiert und vor allem auch überall eingehalten wird. Erst in einem zweiten Schritt sollten dieser dann optimiert und alle entsprechenden Parameter- und Konfigurationseinstellungen vom Configuration Management überprüft sowie, wo es sinnvoll ist, auch weitestgehend standardisiert werden. Dabei spielt es absolut keine Rolle, ob und welches Tool für das Configuration Management zum Einsatz kommt. Es sollte jedoch stets für die jeweiligen Anforderungen des Unternehmens passend und praktikabel sein.

Ist dieser Standard festgelegt und dokumentiert, können auch schon entsprechende Quick-Wins realisiert werden, wie beispielsweise, dass beim Neuaufsetzen eines jeden weiteren Datenbankservers erheblich weniger Dokumentationsaufwand anfällt und eine potenzielle Fehlersuche oder Optimierungsmaßnahmen zielführender und schneller durchgeführt werden können. Zudem wissen dann auch

alle DBAs, etwa aufgrund geeigneter Namensgebungen, sofort darüber Bescheid, ob die Oracle-Datenbanken produktiv oder nur Test-Datenbanken sind, womit auch die Einarbeitungszeit für neue DBAs deutlich kürzer und effektiver ist. Weitere nicht zu unterschätzende Vorteile sind unter anderem, dass auszurollende Skripte und Programme nur noch in einer entsprechenden Referenzumgebung getestet werden müssen und der Aufwand bei der Administration signifikant reduziert werden kann.

In einem dritten Schritt sollten dann aus dem Configuration Management auch die stets vollständigen, richtigen und sowohl die aktuellen als auch die historischen Daten (wie Parameter- und Konfigurations-Einstellungen) zur Verfügung gestellt werden – egal, wie klein oder groß die Oracle-Datenbank-Landschaft ist. Diese Informationen sind beispielsweise für Analysen bezüglich der Auswirkungen und Abhängigkeiten von Änderungen sowie für Fehler- und Ursachen-Analysen zwingend erforderlich. Werden vom Configuration Management hierfür falsche Daten zur Verfügung gestellt, liefert eine darauf basierende Analyse sicherlich ebenfalls falsche Ergebnisse. Zusätzlich hierzu sollte das Configuration Management auch immer aktuelle Antworten auf Konfigurationsfragen beantworten können, wie:

- Welche Datenbanken entsprechen nicht den vom Configuration Management vorgegebenen Baselines?
- Welche Konfigurations- und Parameter-Einstellungen waren in der Vergangenheit für welche Datenbank gesetzt?

- Welche Konfigurations- und Parameter-Einstellungen haben sich wie und wann verändert?
- Welche Konfigurations- und Parameter-Einstellungen welcher Datenbanken verändern sich häufiger?
- Wie ist der Trend bezüglich der Veränderungsrate bei den Konfigurations- und Parameter-Einstellungen?

## Die Configuration Policy für eine Oracle-Datenbank

Einer der wichtigsten Bestandteile einer Configuration Policy für Oracle-Datenbank-Services stellt die Vorgabe der Server-Auswahl und die des Betriebssystems (OS) dar. Grundsätzlich kann man Oracle-Datenbanken auf fast jedem OS und Server-System installieren und betreiben. Doch führt sowohl eine zu breite Technologie-Diversifikation als auch hier vielleicht unpassende Technologie (neben zu vielen Datenbank-Versionen) zu höheren Aufwänden im Betrieb. Dann muss einerseits das entsprechende Wissen in der definierten Technologie-Breite vorgehalten werden und man erhält andererseits aufgrund einer suboptimalen OS- und Serversystem-Technologieauswahl beziehungsweise -Definition eine im Betrieb instabile Oracle-Datenbank, was einen schlechteren und/oder ineffizienteren Datenbank-Service zur Folge hat.

Nachdem innerhalb der Configuration Policy die unter dem Datenbank-Service notwendigen Komponenten wie OS und Server definiert sind, muss man sich über die Installationsprozeduren und Verzeichnisstrukturen der Datenbanken Gedanken machen. Eine Vereinheitlichung dieser hat unter anderem auch den Vorteil, dass dadurch alle Datenbanken identisch aufgesetzt

sind, was in der Folge auch eine Optimierung der Aufwände im administrativen Bereich bedeutet.

Hierfür hat sich beispielsweise die Oracle Flexible Architecture (OFA) als Best Practice etabliert. OFA ist ein Standard, der von der Oracle Special Performance Group entwickelt wurde und eine allgemeine Richtlinie für die Benennung von Dateien und Verzeichnissen für Oracle-Installationen darstellt. Das Ziel hierbei ist einerseits die Erleichterung von Verwaltungsaufgaben – vor allem, wenn mehrere Oracle-Datenbank-Installationen auf demselben System betrieben werden, oder auch bei mehreren Systemen, die auf verschiedenen Servern laufen, – und andererseits, die Vereinheitlichung von Oracle-Installations- beziehungsweise Datei- und Verzeichnisstrukturen sicherzustellen.

Ein weiterer zwingender Bestandteil einer jeden Configuration Policy ist die Beschreibung der Backup-Mechanismen der Datenbank-Server. In diesem Bereich gibt es in der Praxis verschiedene Methoden – Offline- oder Online-Backups, Backup to Tape, Backup to Disk, Backups durch selbstentwickelte oder gekaufte RMAN-Skripte oder mithilfe von Backup-Agenten diverser Hersteller. Es existiert nicht immer eine einzige richtige Lösung, doch sollte man auch hier nicht verschiedene Backup-Techniken einsetzen, sondern sich für eine passende Lösung entscheiden, diese innerhalb der Configuration Policy definieren und dann auch konsequent verwenden.

Weitere Bestandteile einer Configuration Policy können auch vorgegebene Sollwerte für beispielsweise informationssicherheitsrelevante Konfigurations- und Parameter-Einstellungen der Oracle-Datenbanken darstellen – oder auch allgemeine Best-Practice-Vorgaben für andere Konfigurations- und Parameter-Einstellungen. In der Praxis sollte man hier jedoch zwischen „shall“ und „should“ unterscheiden. Bei den „Shall-Vorgaben“ seitens des Configuration Managements sollte ein Abweichen nicht möglich sein. Bei den „Should-Vorgaben“ sollte man im Bedarfsfall – nach Freigabe durch das Configuration Management – davon abweichen können.

### **Einen Standard in der Oracle-Datenbank-Praxis ausrollen**

Erfahrungsgemäß stößt man in vielen Datenbank-Abteilungen schon nach kurzer Zeit auf folgende Fragen: „Die Theorie dazu ist gut und schön, aber wie kann das Ganze auch in der Praxis skalierbar realisiert werden und wie schafft man es, dass auch wirklich alle Server diesem Standard genügen?“

Der in der Praxis am meisten begangene Fehler ist, dass man gerne gleich eine allumfassende und für jede kommende Eventualität gerüstete Lösung sucht und implementieren will, obwohl man im Grunde noch gar nicht wirklich weiß, wie viel Configuration Management man für den jeweiligen Datenbank-Betrieb wirklich braucht. Solche Big-Bang-Lösungsansätze kosten sehr viel Geld und haben leider keinen guten ROI. Nach Erfahrungen des Autors lautet ein wesentlich besserer Ansatz hierbei: „Think big, but start small.“

Leider gibt es aber für das Configuration Management nicht wirklich die eine für alle Oracle-Datenbanken und jedes Unternehmen passende und einfach zu replizierende Lösung. Jedoch kann man hierfür durchaus Best-Practice-Lösungen als eine Art Framework und Grundlage verwenden – man darf dabei nur nie vergessen, diese auch zwingend immer auf die entsprechenden individuellen Anforderungen und Gegebenheiten anzupassen und zu ergänzen. Es empfiehlt sich hierzu folgendes Framework als Vorgehensweise:

- Festlegen/Festschreiben eines Standards
- Evaluieren des Standards durch
  - Aufbau eines neuen Datenbank-Servers – entsprechend Punkt 1
  - Aufbau neuer (leerer) Datenbanken – entsprechend Punkt 1
  - Kopie von bestehenden Test-Datenbanken auf diesen Server und Migration entsprechend dem neuen Standard nach Punkt 1
  - Sollte in diesen Schritten ein Problem auftreten, sind der Standard entsprechend anzupassen und die genannten Schritte zu wiederholen
- Ab jetzt gilt der Standard für alle neuen Datenbank-Server und alle neuen Datenbanken

- Dann werden alle Test-Datenbanken schrittweise hinsichtlich des neuen Standards migriert – sofern notwendig
- Im nächsten Schritt betrifft es alle QA-Datenbanken
- Danach werden in Absprache mit den Anwendungen möglichst eng verbunden zuerst die INT-Datenbanken und nach Erfolg auch die zugehörigen PROD-Datenbanken migriert

### **Definition eines Standards beziehungsweise einer Configuration Policy für Oracle-Datenbanken**

Die wichtigste Frage ist: „Wie genau definiere ich einen solchen Standard, und muss dieser sehr detailliert sein oder nur ganz grob und dafür möglichst schnell?“ Wie bereits beschrieben, ist es sehr zielführend, möglichst schnell zu einem definierten Standard zu kommen (Schritt 1). Im Rahmen von Schritt 2, der Evaluierung dieses Standards, wird dieser dann auf seine Praxistauglichkeit getestet, optimiert und ergänzt. Eine sehr gute Basis hierbei ist die Verwendung der von Oracle empfohlenen Standards (wie OFA) und der Konfiguration der Server gemäß den Installation- und User Guides (IUG) und My-Oracle-Support-Notes.

Das in diesem Artikel weiter Folgende stellt keineswegs eine starre Vorgabe dar und ist auch nicht die für alle Unternehmen passende, finale Vorgehensweise und Lösung – es ist aber eine gute Option und Orientierungshilfe. Falls jedoch im Unternehmen bereits ein anderer funktionierender Standard etabliert ist, der auch ähnliche Vorteile aufweist, wäre die Einführung eines jetzt wieder komplett neuen Standards (vor allem in Bezug auf das Kosten-/Nutzen-Verhältnis) sicherlich nicht wirklich empfehlenswert. Aber auch in diesem Fall kann das beschriebene Standardisierungs-Vorgehen sehr gut verwendet werden, um den eigenen, bereits definierten Standard gegen diese zu überprüfen:

- *Datenbank-Server*
  - Für Produktionsserver sollten ausschließlich von Oracle zertifizierte Umgebungen (Hardware) eingesetzt werden

- Auf einer Produktionsumgebung laufen nur produktive Datenbanken. Das gilt auch für virtualisierte Umgebungen. Integrations-, Test- und QA-Datenbanken können dagegen untereinander gemischt werden
- Datenbanken mit unterschiedlichen SLAs sollten nicht gemischt werden. So sollten auf einer Plattform nur eine oder mehrere RAC-Umgebungen laufen, nicht auch noch Single-Datenbanken
- Nach Möglichkeit sollte die Anzahl verschiedener OS und OS-Versionen für Datenbank-Server minimiert werden, um Shell-Skripte zu vereinfachen etc.
- Oracle Enterprise Linux ist ausgereift und wird ebenfalls durch Oracle direkt unterstützt. Damit gibt es für OS und Datenbank nur einen Lieferanten und einen Support
- Optimal Flexible Architecture (OFA, definiert von Oracle) ist eine sehr gute Basis für ein File- und Verzeichnissystem. Unter „/u0x/app/oracle/rdbms/11.2.0.3.0“ befinden sich die verschiedenen ORACLE\_HOMEs, unter „/u00/app/oracle/grid/11.2.0.3.0“ liegt das GRID, unter „/u00/app/oracle/agent/11.2.0.3.0“ die Software für den EM-Agent und unter „/u0y (x>1, y>=0)“ sind dann die Datenbank-Dateien der x-ten Datenbank zu finden
- Sollte auch nur eine Datenbank im Unternehmen ASM nutzen, so ist der grundsätzliche Einsatz von ASM für alle Datenbanken empfohlen
- Die OS-Parameter sollten immer entsprechend den aktuellen Notes in My Oracle Support gesetzt werden. Die IUG sind dazu nicht immer aktuell
- Oracle-Binaries wie auch das DIAG-Verzeichnis sollten immer auf lokalen Platten liegen, jedoch in getrennten Filesystemen
- Alle Konfigurationsdateien (SPFILE, orapwd, sqlnet.ora, listener.ora, tnsnames.ora etc.) liegen in einem separaten Filesystem auf lokalen Platten außerhalb der Oracle-Binaries. Über symbolische Links werden diese im ursprünglichen Verzeichnis zur Verfügung gestellt
- Wann immer möglich, sollte man ASM oder RAID 10 im Storage nutzen
- Alle Server eines Unternehmens sollten dieselbe Uhrzeit haben, unabhängig von ihrem lokalen Rechenzentrum, um so besser LOG-Files auf unterschiedlichen Servern vergleichen zu können
- *Oracle-Datenbank*
  - SID = Instance-Name, alles in Großbuchstaben, Präfix = „P“, „I“, „T“ oder „Q“ – je nach Art und Zweck der Datenbank
  - Es empfiehlt sich die Nutzung der Flash Recovery Area
  - Sinnvoll ist eine einheitliche Filegröße für alle Datendateien
  - Bei allen Datendateien sollte „autoextend = OFF“ gesetzt sein
  - Alle unnötigen oder Default-Parameter aus dem SPFILE entfernen
  - Nutzung von Automatic Segment Space Management (ASSM)
  - Aktivieren des Automatic Shared Memory Management (ASMM) und Automatic Memory Management (AMM); verschiedene Bereiche (buffer\_cache, shared\_pool etc.) sollten jedoch so gesetzt sein, dass die Performance des Servers gut ist
  - Aktivieren des Flashback-Modus, sofern der Performance-Impact nicht zu groß ist
  - OMF sollte nur in Testumgebungen eingesetzt werden
  - Aus lizentechnischen Gründen nur diejenigen Optionen installieren, die wirklich auch genutzt werden
  - Nutzung von WAR-Reports (sofern Lizenzen vorhanden) oder STATSPACK-Reports. Das Intervall sollte auf 15 Minuten und der Zeitraum auf 33 Tage gesetzt sein, um so einen besseren Vergleich zu haben
  - Erzeugen der SYSTEM-Statistiken
  - Vermeiden des Setzens/Änderns der „OPTIMIZER\_INDEX\*“-Parameter – wenn irgend möglich
- *RAC*
  - Von der Nutzung des ASM Cluster File Systems (ACFS) ist abzuraten
- „ASM\_PREFERRED\_READ\_FAILURE\_GROUPS“ im SPFILE setzen
- Der SCAN-Listener sollte genutzt werden
- SCAN-Listener-Name = Cluster-Name
- Die normalen Listener und der Interconnect haben je ein privates Netzwerk
- „LOCAL\_LISTENER“ und „REMOTE\_LISTENER“ sollten gesetzt sein
- *ASM*
  - Explizite Angabe von Failure Groups
  - „Normal Redundancy“ innerhalb von ASM verwenden
- *SQL\*NET*
  - „DB\_DOMAIN“ im SPFILE setzen
  - „GLOBAL\_NAMES=TRUE“ im SPFILE setzen
  - Die Verbindung zur Datenbank nur via „SERVICE\_NAME“ und nicht mehr über die Oracle System ID (SID) herstellen
  - Es empfiehlt sich nur ein Listener pro Datenbank-Server und pro Oracle-Version
  - Nach Möglichkeit keine statischen Einträge in „listener.ora“
- *Patching/Upgrade/Migration/Deployment*
  - Es empfiehlt sich der Einsatz von „guaranteed restore points“, um möglichst schnell den Ursprungszustand beim Patchen, Upgrade, Migration oder nach einem Deployment wiederherstellen zu können.
- *Täglich anfallende Arbeiten*
  - Grundsätzlich sollte so viel wie möglich automatisiert werden, um manuelle, individuelle Fehler zu vermeiden und um den DBA zu entlasten
  - Index-Reorganisation kann sinnvoll sein, wenn mehr als 25 Prozent der Zeilen der Tabelle gelöscht wurden. Eine grundsätzliche, regelmäßige Index-Reorganisation ergibt keinen Sinn – lediglich in besonderen, begründeten Fällen
  - Eine Reorganisation von Tabellen ist nur dann empfehlenswert, wenn die Anzahl der „migrated rows“ signifikant ist und durch die Anwendung auf diese noch zugegriffen wird

- Es empfiehlt sich, den DBA\_SCHEDULER insbesondere für Datenbank-interne Jobs zu nutzen
- **Monitoring**
  - Zur Überwachung aller Datenbanken ist der Einsatz von nur einem Tool empfohlen
  - Bei allen neu zu schreibenden oder anzupassenden Skripten, die auf das Alert-File zugreifen, sollte „alert.xml“ ausgelesen werden, da ab Version 13 „alert.log“ wahrscheinlich nicht mehr unterstützt wird
- **Dokumentation**

Jeder Datenbank-Server und jede Datenbank ist ausführlich zu dokumentieren. Ganz besonders wichtig sind dabei folgende Aspekte:

  - Genaue Version des OS und aller eingespielten Patches
  - Hauptverantwortlicher System-Admin und Stellvertreter
  - Hauptverantwortlicher Netzwerk-Admin und Stellvertreter
  - Hauptverantwortlicher Storage-Admin und Stellvertreter
  - Hauptverantwortlicher DBA und Stellvertreter
  - Mögliche Wartungsfenster
  - SLA
  - Wann findet ein Backup statt? Welche Art von Backup?
  - Sonstige Besonderheiten, wie und warum sich diese DB vom Standard abhebt
  - Genaue Version der Datenbank und aller eingespielten Patches
  - Name der Instance und Name der Datenbank
  - Welche Non-Default-Parameter wurden warum gesetzt?
  - Konfiguration von SQL\*NET (auf dem Datenbank-Server)
  - Welche Optionen sind installiert?
  - Welche Optionen werden auch genutzt?
  - Wann werden Daten und in welchem Umfang in die DB geladen? Von wo?
  - Wann und in welchem Umfang werden Daten zu anderen DBs verschoben? Und wie?
  - Wann und in welchem Umfang werden Daten im Filesystem gespeichert? Und wohin?

```

Statement s = null;
Connection con = null;

String MyName = "PROGRAMNAME";
String nameForConnect = userId;
String HostName = null;
String Terminal = null;

try {
    HostName = InetAddress.getLocalHost().getCanonicalHostName();
} catch (Exception e) {
}

Terminal = HostName;
int i = HostName.indexOf(".");
if(-1 != i) {
    Terminal = HostName.substring(0, i);
}

String OsUser = System.getProperty("user.name");

OperatingSystemMXBean mxBean = (com.sun.management.OperatingSystemMXBean) ManagementFactory
    .getOperatingSystemMXBean();
String ProcessName = "1234@" + mxBean.getName();

if (nameForConnect.equalsIgnoreCase("SYS")) {
    nameForConnect = nameForConnect + " as sysdba";
}

try {
    Properties pro = new java.util.Properties();

    pro.setProperty("password", passWord);
    pro.setProperty("user", nameForConnect);

    try {
        pro.put("ApplicationName", MyName);
        pro.put("ClientHostname", HostName);
        pro.put("ClientUser", OsUser);

        pro.put("v$session.osuser", OsUser);
        pro.put("v$session.machine", HostName);
        pro.put("v$session.program", MyName + ".jar");
        pro.put("v$session.process", ProcessName);
        pro.put("v$session.terminal", Terminal);
    } catch (Exception ignore) {
    }

    DriverManager.registerDriver (new oracle.jdbc.OracleDriver());
    Class.forName(getDriverName()).newInstance();
    con = DriverManager.getConnection(getUrl(), pro);

    s = con.createStatement();
    s.setFetchSize(getFetchedRows());

    return s;

} catch (Exception ignore) {
}

```

Listing 1

- Welche Anwendungen greifen wie und von wo auf diese DB zu?
- Wer sind Ansprechpartner und Stellvertreter der jeweiligen Anwendung?
- Gibt es Abläufe in der Anwendung, die in einem bestimmten Zeitfenster fertig sein müssen? Wie kann man dies überprüfen?
- Wie ist die Userverwaltung organisiert?
- Welche Tablespaces werden von dieser Anwendung genutzt?
- Mit welchem Datenwachstum muss man pro Anwendung für diese Datenbank innerhalb von sechs Monaten rechnen?
- **JDBC-Connections**
  - Um beim Aufbau einer JDBC-Verbindung zur Datenbank möglichst viele Informationen mitzugeben, empfiehlt es sich, ein Java-Codefragment zu implementieren (siehe Listing 1).

### Definition der Datenbank- und Datenbankserver-Parameter

Um die notwendigen Attribute der CIs für Datenbank-Services zu definieren, gibt es zwei praktikable Ansätze:

#### • Methode 1

Es wird vorab ein Set von wahrscheinlich beziehungsweise in der Regel für alle Datenbanken relevanten Parameter- und Konfigurations-Einstellungen definiert, das zu Beginn für jedes CI (in diesem Fall die entsprechende Oracle-Datenbank) am besten in einer Configuration

Management Database (CMDB) erfasst, permanent überprüft und bei Bedarf pro CI individuell mit weiteren Parameter- und Konfigurations-Einstellungen erweitert wird. Dafür haben sich in der Praxis bestimmte Parameter- und Konfigurations-Einstellungen als eine Art „Initial-Set“ bewährt (siehe Tabellen unter [www.doag.org/go/doagnews/wischki\\_tabelle](http://www.doag.org/go/doagnews/wischki_tabelle))

#### • Methode 2

Es werden am Anfang alle Parameter- und Konfigurations-Einstellungen für jedes CI (in diesem Fall die jeweilige Oracle-Datenbank) in der CMDB erfasst und permanent überprüft. Aus diesem Set werden dann die jeweils nicht relevanten Parameter- und Konfigurations-Einstellungen sukzessive ausgegliedert, sodass schlussendlich die wirklich relevanten übrig bleiben.

Welche Methode wann zu verwenden ist, hängt zumeist vom Wissensgrad hinsichtlich der für die CI relevanten Parameter- und Konfigurations-Einstellungen ab. Sind sie weitgehend bekannt, sollte man Methode 1 verwenden, wenn nicht, Methode 2 bevorzugen.

#### Fazit

Für ein erfolgreiches Configuration Management im Oracle-Datenbank-Bereich gilt immer: Das richtige Maß und Ziel ist entscheidend! Wie bereits beschrieben, gibt es kein für alle Datenbank-Landschaften passendes Configuration Management, da hier

immer individuelle Komponenten wie die Menge und Komplexität der Datenbanken sowie auch die hierauf basierenden Applikationen einen signifikanten Einfluss haben.

Kein oder zu wenig Configuration Management haben eine unnötig hohe Total Cost of Ownership (TCO) für den Datenbank-Betrieb zur Folge – zu viel Configuration Management aber auch. In jedem Fall sollte man mit dem Configuration Management immer tendenziell kleiner anfangen und dieses dann nach Bedarf erweitern – aber auch stets die Skalierbarkeit im Auge behalten. Der umgekehrte Weg, gleich den „Big Bang“ zu realisieren und zu implementieren, sowie das von einigen Lösungsanbietern versprochene „quick and easy“ oder gar „out of the box“ ist in der Regel immer der teurere Weg.



Christian Wischki  
cw@christian-wischki.com



Rainer Hartwig  
rainer.hartwig@mt-ag.com



**Herrmann & Lenz**

Services

Unsere Leistungen

- Oracle-Installation & Administration
- Performanceanalyse & Tuning
- Fernwartung & Betrieb
- Lizenzberatung & Optimierung
- Schulungen & Workshops

**Die Datenbank-Profis**

[www.hl-services.de](http://www.hl-services.de)



**Herrmann & Lenz**

Solutions

**HL Monitoring Module**

*Schlank – Zentral – Zuverlässig – Innovativ*

- Für Express-, Standard- und Enterprise-Edition
- Keine Enterprise-Packs notwendig
- Erweiterungen für VMware und MS SQL-Server

**Innovative Systemlösungen**

[www.hl-solutions.de](http://www.hl-solutions.de)

Die Installation der Oracle-Software ist eine der wichtigsten Aufgaben, wenn es darum geht, einen neuen Server in Betrieb zu nehmen oder ein Update durchzuführen. Die Richtlinie der Optimal Flexible Architecture (OFA) soll dabei helfen, einheitlich vorzugehen, um so unter anderem den zukünftigen Betrieb zu erleichtern. Der Fokus des Artikels liegt darauf, diese im Umfeld von Unix-Systemen vorzustellen und näher zu erläutern.

# „OFA“ – wozu brauche ich das denn?

Stefan Panek, CarajanDB

Die Tätigkeit als Berater im Oracle-Datenbank-Umfeld bringt es mit sich, dass man häufig zu Kunden kommt und sich dort zuerst einmal einen Überblick über die vorhandenen Installationen auf den Server-Systemen verschafft. Oft findet man Installationen vor, die recht ungewöhnlich sind. Die Verzeichnisstruktur für die Oracle-Software-Installation wird dabei von den Administratoren selbst festgelegt. Vorhandene Standards wie die OFA werden nicht immer angewendet. Dies führt dann unter anderem zu erheblichen Mehraufwänden in der täglichen Administration oder bei Upgrades der Datenbank-Systeme. An dieser Stelle sei schon einmal erwähnt, dass OFA eine der ältesten Oracle-Best-Practices ist und somit auch fester Bestandteil einer jeden Oracle-Software-Installation sein sollte.

## Die Optimal Flexible Architecture

Der OFA-Standard ist unter anderem eine Richtlinie für das Layout der Dateisysteme und Verzeichnisstrukturen auf einem Oracle-Server-System. Im Jahre 1991 wurde auf der International Oracle User Week in Miami, USA, der erste Vortrag zu diesem Thema gehalten. Carry Millsap und sein Performance-Team beschrieben in einem Papier, wie ein High-Performance-

System konfiguriert werden sollte. Ein wesentliches Ziel war, den Aufwand für die tägliche Administration möglichst gering zu halten. Durch die standardisierte Vorgehensweise in Bezug auf Installation und Setup eines Systems, die das Team um Carry Millsap entwickelt hatte, konnten so viele Fehler bei produktiven Systemen vermieden werden. Im Jahre 1992 veröffentlichte Oracle Services eine erweiterte Version des OFA-Standards. Es kamen Themen wie Standards für Namen, Zugriffsrechte, Unix-Logins etc. hinzu. 1993 integrierte Oracle dann den OFA-Standard in die Dokumentation der Oracle Version 7, bevor im Jahre 1995 eine weitere Version von Carry Millsap veröffentlicht wurde. Dieses Dokument ist auch heute noch verfügbar und hat den Titel „The OFA-Standard – Oracle for Open Systems“.

Betrachtet man heute die aktuelle Dokumentation der Datenbank 11g R2, findet man die Erläuterungen der Optimal Flexible Architecture im Anhang des Installation Guide für die jeweilige System-Plattform. Zum Abschluss des geschichtlichen Hintergrunds hier noch einige Punkte der ursprünglichen Motivation für die OFA-Richtlinie, auch wenn der eine oder andere Punkt heute nicht mehr aktuell ist:

- OFA ist der Versuch, Oracle-Software so zu organisieren, dass möglichst keine Performance-Engpässe (auf lokalen Disks) auftreten
- Routine-Tätigkeiten wie Software und Datensicherungen sollen fehlerfrei laufen
- Vermeidung von Daten-Korruptionen, wenn auf einem Server zwischen mehreren Datenbanken umgeschaltet wird

Neben der ursprünglichen Motivation sind die zentralen Vorteile, die für eine Implementierung der OFA Struktur sprechen:

- Einheitliche Verzeichnis- und Namensstruktur auf Server-Systemen, unabhängig von der eingesetzten Plattform
- Weniger Aufwand bei der Datenbank-administration
- Mehrere Datenbank-Versionen lassen sich bequem nebeneinander verwalten
- Leichtere Fehlersuche für Oracle Support bei Service Requests

Die Implementierung des OFA-Standards auf Basis der aktuellen Oracle-Dokumentation des Release 11g R2 ist in folgende Bereiche gegliedert:

- Datei-Systeme (Mount Points)
- Namenskonvention für Verzeichnis-Strukturen
- Namenskonvention für Datenbank-Dateien
- Darstellung einer Dateistruktur nach OFA

Variable	Beschreibung
pm	Mount Point Name
s	Standardverzeichnis
u	Besitzer des Verzeichnis („oracle“)

Tabelle 1



## Datei-Systeme

Folgt man den Empfehlungen von Oracle, so sollten aus Gründen der Redundanz mindestens drei Datei-Systeme vorhanden sein. Diese sind so zu erstellen, dass sie auf physikalisch voneinander getrennten Devices basieren. Default für die Namenskonvention ist „/u[0-9][0-9]“, also „/u01“, „/u02“, „/u03“ oder auch „/disk01“, „/disk02“, „/disk03“.

## Namenskonvention für Verzeichnis-Strukturen

Oracle empfiehlt generell, bei der Vergabe von Namen nur ASCII-Zeichen zu verwenden. Der Startpunkt für die Verzeichnisstruktur ist das Oracle-Base-Verzeichnis. Oracle benutzt hier eine eigene Notation mit Variablen wie „pm“, „s“, „u“ etc. (siehe Tabelle 1).

Wenn die Installation durch den User „oracle“ vorgenommen wurde, lautet der Name dann „/u01/app/oracle“. Damit mehrere Oracle-Versionen auf einem System installiert und betrieben werden können, sieht OFA für das „ORACLE\_HOME“-Verzeichnis die in Tabelle 2 gezeigte Struktur vor. Die Installation der Oracle-RDBMS-Software in der Version 11.2.0 würde damit „/u01/app/oracle/product/11.2.0/dbhome\_1“ lauten.

Der OFA-Standard empfiehlt, pro Datenbank eine administrative Verzeichnisstruktur ausgehend vom Oracle-Base-Verzeichnis zu erstellen. Für die Datenbank „DB1“ wäre das dann „/u01/app/oracle/admin/DB1“. Tabelle 3 zeigt die entsprechende Unterverzeichnisstruktur.

Mit der Datenbank-Version 11g führte Oracle zusätzlich das Automatic Diagnostic Repository (ADR) ein, das die Verzeichnisse „/udump“, „/bdump“ und „/cdump“ ersetzt. Tabelle 4 zeigt, wie sich das „ADR\_HOME“-Verzeichnis zusammensetzt. Ein Beispiel für die Datenbank „DB1“ wäre „/u01/app/oracle/diag/rdbms/db1/DB1“. Tabelle 5 zeigt, wie sich die weiteren „ADR“-Verzeichnisse unterteilen.

## Namenskonvention für Datenbank-Dateien

In Tabelle 6 sind die Namenskonventionen aufgelistet. Oracle schlägt auf

Variable	Beschreibung
pm	Mount Point Name
s	Standardverzeichnis
u	Besitzer des Verzeichnisses
v	Produktversion
type	Database, Client, Grid-Infrastruktur
n	Zahl, die es ermöglicht, das gleiche Produkt mehrfach in ein Oracle-Base-Verzeichnis zu installieren

Tabelle 2

Verzeichnis	Beschreibung
arch	Archivierte Redo-Log-Dateien
adump	Audit-Dateien
create	Default für Data-Pump-Operationen, Skripte zum Erstellen der Datenbank etc.
dpdump	Data-Pump-Logdateien
exp	Export-Dateien
pfile	Instance-Parameter-Datei
scripts	Ad-hoc-SQL-Skripte

Tabelle 3

Variable	Beschreibung
h	Oracle Base
d	Database Name
i	Instance Name

Tabelle 4

Diagnostic-Daten	Lokation
Foreground Process Traces	ADR_HOME/trace
Background Process Traces	ADR_HOME/trace
Alert-Log-Dateien	ADR_HOME/alert
Core Dumps	ADR_HOME/incident/
Incident Dumps	ADR_HOME/incident/

Tabelle 5

Variable	Beschreibung
h	Oracle Base
q	Ein String zur Unterscheidung, z.B. oradata
d	„DB_NAME“ ist normalerweise identisch zur SID bei Single-Instance-Systemen
t	Tablespace-Name
n	Zwei Ziffern zur Nummerierung der Dateien

Tabelle 6

Verzeichnisstruktur Syntax	Beschreibung
/u[0-9][0-9]	User-Directory
/*/home/*	User-Home-Directory
/*/app/*	Application-Software-Directory
/*/app/oracle/product/	Oracle-Software-Struktur
/*/app/oracle/product/11.2.0	Oracle-Software-Struktur für 11g
/*/app/oracle/product/11.2.0/db*	Oracle-Home-Struktur für Database 11g
/*/app/oracle/product/11.2.0/grid	Oracle-Home-Struktur für Grid-11g-Standalone-Installation
/*/app/oracle/admin/DB1	DB1-Datenbank, administrative Verzeichnisse
/*/app/oracle/admin/DB1/arch	DB1-Datenbank, Archive-Log-Files-Verzeichnis
/*/app/oracle/oradata	Oracle-Datenfiles-Verzeichnisse

Tabelle 7

Verzeichnis	Bezeichnung
/	root
/u01	Mount Point 1
/u01/app	Unterverzeichnis Application Software
/u01/app/oracle	Oracle-Base-Verzeichnis
/u01/app/oracle/admin	Unterverzeichnis Datenbankadministration
/u01/app/oracle/admin/TAR	Verzeichnis für Support-Logfiles
/u01/app/oracle/admin/db_name1	Administration db_name1
/u01/app/oracle/admin/db_name2	Administration db_name2
/u01/app/oracle/doc	Oracle-Dokumentation
/u01/app/oracle/fast_recovery_area	Unterverzeichnis für Recovery-Dateien
/u01/app/oracle/fast_recovery_area/db_name1	Unterverzeichnis für Recovery der db_name1
/u01/app/oracle/fast_recovery_area/db_name2	Unterverzeichnis für Recovery der db_name2
/u02/app/oracle/oradata	Oracle-Datenverzeichnis
/u03/app/oracle/oradata	Oracle-Datenverzeichnis
/u04/app/oracle/oradata	Oracle-Datenverzeichnis
/u01/app/oracle/product/11.2.0/db_home1	Oracle-Home-Directory für User "oracle"
/u01/app/oracle/product/11.2.0/grid	Oracle-Home für Grid-Infrastruktur, bei einer Standalone-Installation

Tabelle 8

Verzeichnis	Bedeutung
/u01/app/oracle/product/11.2.0.1/db_1	Release 11.2.0.1
/u01/app/oracle/product/11.2.0.2/db_2	Release 11.2.0.2
/u01/app/oracle/product/11.2.0.3/db_1	Release 11.2.0.3
/u01/app/oracle/product/11.2.0.3/client_1	11.2.0.3 Client Installation

Tabelle 9

der Basis von OFA vor, Control-, Redo-Log- und Datenbank-Dateien in unterschiedlichen Pfaden abzulegen. Ein System-Tablespace der Datenbank „DB1“ mit einer Daten-Datei würde dann lauten „/u01/app/oracle/oradata/DB1/system01.dbf“.

**Darstellung der Dateistruktur nach OFA**

Tabelle 7 zeigt den Standard, wie er in der Dokumentation beschrieben ist. In Tabelle 8 steht die detaillierte Datei-Zuordnung für eine Oracle-Software-Installation mit zwei „ORACLE\_HOMEs“ und zwei Datenbanken. Die Daten-Dateien sind dabei über drei Datei-Systeme verteilt.

Wie so oft, bestehen zwischen Theorie und Praxis doch Unterschiede, die man bei der Anwendung des OFA-Standards berücksichtigen sollte. Nachfolgend sind einige Anregungen aufgezählt.

**Patchset-Auslieferung ab Release 11.2.0.2**

Seit Release 11.2.0.2 liefert Oracle grundsätzlich Software und Patchsets als komplette Versionen aus. Das bedeutet, jedes Patchset ist eine neue Oracle-Software-Version. Da laut Standard mit „11.2.0“ gearbeitet wird, ist natürlich nicht auf den ersten Blick erkennbar, dass es sich bei „db\_1 = 11.2.0.1“ und bei „db\_2 = 11.2.0.2“ um eine jeweils eigenständige Oracle-Version handelt. Daher findet man in der Praxis des Öfteren eine abgewandelte OFA-Struktur vor (siehe Tabelle 9).

Mit dieser Modifikation des OFA-Standards ist auf einen Blick erkennbar, um welches Oracle-Release es sich handelt. Zudem ist dringend empfohlen, eine Typ-Bezeichnung wie „db\_1“ beizubehalten. Diese wird nicht zuletzt dann benötigt, wenn in der Verzeichnis-Struktur weitere Produkte installiert sind, wie die Oracle-Client-Software mit „client\_1“ etc.

**Oracle-Net-Konfigurationsdateien**

Bei der Installation eines Oracle-Software-Release werden automatisch unter dem Verzeichnis „ORACLE\_HOME/network/admin“ eine Reihe von Konfigurationsdateien wie „sqlnet.ora“, „tnsnames.ora“ etc. abgelegt. Wird auf

einem Server-System ein zweites oder drittes Oracle-Release betrieben, müssen diese Dateien mehrfach gepflegt werden, was den administrativen Aufwand deutlich vergrößert. Daher empfiehlt sich, hier eine Zentralisierung vorzunehmen, die wie folgt aussehen könnte: Anlegen des Verzeichnisses „/u01/app/oracle/admin/common/network“. Dort sind sämtliche „\*.ora“-Dateien aus dem (oder auch den weiteren) „ORACLE\_HOME“ abgelegt. Danach werden entsprechende Soft-Links zu den ursprünglichen Verzeichnissen erstellt. Somit ist die Möglichkeit geschaffen, die Oracle-Net-Dateien zentral zu pflegen, unabhängig von der Anzahl der Installationen auf dem Server (siehe Beispiel in Tabelle 10).

Verzeichnis	Bezeichnung
/	root
/u01/app/oracle	Oracle-Base-Verzeichnis
/u01/app/oracle/admin/common/network	Ablage sämtlicher Oracle-Net-Konfigurationsdateien

Tabelle 10

Verzeichnis	Bezeichnung
/	root
/u01/app/oracle	Oracle-Base-Verzeichnis
/u01/app/oracle/admin/common/etc	Ablage von Dateien außerhalb des Oracle-Base-Verzeichnisses

Tabelle 11

**Die Dateien „oratab“ und „oraInst.loc“**

Bei jeder Oracle-Installation gibt es einige Dateien, die nicht direkt unter der Oracle-Software-Installation beziehungsweise im Oracle-Base-Verzeichnis abgelegt sind. Dazu gehören die Dateien „/etc/oratab“ und „/etc/oraInst.loc“. Die „oratab“-Datei ist die zentrale Datei, wenn es um die Verwaltung des Environments geht. Die Datei „oraInst.loc“ verweist auf das zentrale Oracle-Inventory-Verzeichnis.

Abhängig von der Betriebssystem-Variante liegen diese Dateien an verschiedenen Stellen im Dateisystem, wie „/etc“, „/var/opt“ oder „/var/opt/oracle“, um nur einige zu nennen. Hier bietet es sich an, eine Vereinheitlichung über alle Betriebssystem-Plattformen hinweg vorzunehmen. Durch die Zentralisierung erreicht man unter anderem,

dass die Dateien mit einem Backup der Oracle-Binaries gesichert sind. Um dies umzusetzen, müsste der OFA-Standard wie in Tabelle 11 gezeigt angepasst werden. Die Dateien werden in das neue Verzeichnis „/u01/app/oracle/admin/common/etc“ verschoben. Abschließend wird ein symbolischer Link zum Original-Verzeichnis erstellt.

**Fazit**

Die Oracle Flexible Architecture ist eine der wichtigsten und ältesten Oracle-Best-Practices. Der OFA-Standard sollte immer berücksichtigt werden, wenn die Installation von Oracle-Software ansteht. Mit einer Standardisierung der Oracle-Installationen nach OFA wird unter anderem ein stabiler Betrieb der Datenbanken erreicht. Ferner ist so eine Grundlage geschaffen, die es er-

möglicht, auch weitere Oracle-Produkte ohne Kopfzerbrechen zu integrieren.

**Quellen**

1. Oracle, Database Installation Guide Release 11.2.0: [http://docs.oracle.com/cd/E11882\\_01/install.112/e24321.pdf](http://docs.oracle.com/cd/E11882_01/install.112/e24321.pdf)
2. Carry Millsap, The OFA Standard-Oracle for Open Systems: [http://methodr.com/index.php?option=com\\_docman&task=cat\\_view&gid=38&Itemid=149](http://methodr.com/index.php?option=com_docman&task=cat_view&gid=38&Itemid=149)

Stefan Panek  
stefan.panek@carajandb.com



**Impressum**

**Herausgeber:**

DOAG Deutsche ORACLE-Anwendergruppe e.V.  
Tempelhofer Weg 64, 12347 Berlin  
Tel.: 0700 11 36 24 38  
www.doag.org

**Verlag:**

DOAG Dienstleistungen GmbH  
Fried Saacke, Geschäftsführer  
info@doag-dienstleistungen.de

**Chefredakteur (ViSdP):**

Wolfgang Taschner, redaktion@doag.org

**Redaktion:**

Fried Saacke, Carmen Al-Youssef, Mylène Diacquenod, Dr. Dietmar Neugebauer, Franz Hüll, Dr. Frank Schönthaler, Christian Trieb

**Titel, Gestaltung und Satz:**

Claudia Wagner, Fana-Lamielle Samatin  
DOAG Dienstleistungen GmbH

**Titelfoto:** Fotolia

**Anzeigen:**

CrossMarketeam Doris Budwill  
www.crossmarketeam.de  
Mediadaten und Preise finden Sie unter:  
www.doag.org/go/mediadaten

**Druck:**

adame Advertising and Media  
GmbH Berlin, www.adame.de

Was zeichnet einen Advisor in der Oracle-Datenbank aus? Sind Advisors nicht generell lizenzpflichtig? Muss man Advisors aktivieren oder sind diese automatisch eingeschaltet? Welche Werkzeuge unterstützen das Framework? Dieser Beitrag beantwortet diese und weitere Fragen.

# Oracle Database Advisors

Ulrike Schwinn, ORACLE Deutschland B.V. & Co.KG

Der englische Begriff „advisor“ lässt sich mit „Berater“ oder „Ratgeber“ übersetzen. Der Terminus „Advisor“ findet mittlerweile in vielen Bereichen Anwendung; so gibt es Personal Advisor, Investment Advisor, Trusted Advisor etc. Auch die Datenbank beziehungsweise Datenbank-Werkzeuge enthalten Advisors. Da das Thema sehr weitreichend ist, gibt dieser Artikel nur einen Überblick über vorhandene Advisors und stellt Gemeinsamkeiten und Unterschiede heraus. Weiterführende Links und Hinweise auf Veröffentlichungen runden das Thema ab.

## Advisors in der Datenbank

Alerts und Advisors unterstützen den DBA bei der täglichen Arbeit. Alerts werden automatisch vom Server generiert und verlangen entweder eine sofortige Intervention durch den Administrator wie bei „Instance orcl.orcl. down on node...“ oder geben Hinweise auf einen Umstand in der Datenbank, der in der nächsten Zeit behoben werden sollte. Zu finden sind diese Informationen auf der Datenbank-Homepage im Enterprise Manager oder über die View „DBA\_OUTSTANDING\_ALERTS“. Teilweise werden auch Hinweise auf den Start eines Advisors gegeben, um das Problem zu lösen.

Advisors hingegen aktiviert in der Regel der DBA selbst. Sie analysieren bestimmte Objekte in der Datenbank und geben Lösungsvorschläge, die einen größeren Detaillierungsgrad aufweisen als Alerts. Advisors sind somit Ressourcen-intensiver als Alerts. Sie können in den unterschiedlichsten Bereichen eingesetzt werden wie „Memory“, „Speicherverwaltung“, „Undo Management“, „Performance“ etc. Advisors müssen nicht zusätzlich instal-

liert werden und sind teilweise schon aktiviert. Die Skripte oder Ratschläge werden normalerweise nur aufgelistet und der DBA muss sie selbst durchführen – er behält somit stets die Kontrolle. Er sollte nur wissen, wo und wie Advisors und ihre Ratschläge zu finden beziehungsweise wie sie zu aktivieren und zu nutzen sind.

Advisors sind in der Regel über unterschiedliche Methoden nutzbar, grafisch über den Enterprise Manager oder SQL Developer, skriptgesteuert über PL/SQL-Packages, über Initialisierungsparameter und V\$-Views. In den nachfolgenden Absätzen sind die wichtigsten Advisors kurz erläutert (Stand September 2012).

Der **Memory Advisor** analysiert die unterschiedlichen Cache-Größen in einer WHAT-IF-Analyse und macht Vorschläge zum Tuning der Memory-Bereiche wie SGA, Shared Pool etc. Er ist standardmäßig aktiviert. Zu beachten ist, dass die Oracle-Datenbank den Memory je nach Wahl der Memory-Management-Methode selbst einstellt und somit kein manuelles Eingreifen notwendig ist. Die Ratschläge sind entweder in der grafischen Oberfläche oder über V\$-Views wie „V\$MEMORY\_TARGET\_ADVICE“ zu finden.

Der **Undo Advisor** hilft dabei, eine angemessene Größe des UNDO-Tablespace für langlaufende Abfragen und Flashback-Queries festzulegen. Die Nutzung erfolgt grafisch oder über das Package „DBMS\_ADVISOR“. Zur Berechnung der Größe werden WAR-Daten verwendet.

Der **Segment Advisor** findet Segmente, die zu viel Platz verbrauchen. Die Empfehlungen beinhalten die entsprechenden Kommandos zur Freigabe von Speicherplatz. Die notwendi-

gen Statistiken für die Analyse werden aus dem AWR beziehungsweise ADDM und aus Sample-Statistiken des Segments abgeleitet. Die Analyse erfolgt entweder automatisch, zum Beispiel in der Automated Maintenance Task, oder manuell.

Der **Compression Advisor** berechnet die Komprimierungsrate pro Tabelle oder Partition für unterschiedliche Komprimierungsalgorithmen – sogar für Exadata Hybrid Columnar Compression (EHCC). Darüber hinaus kann er den eingestellten Komprimierungstyp diagnostizieren. Die Nutzung des Compression Advisors erfolgt in 11g R2 ohne zusätzliche Installation über das Package „DBMS\_COMPRESSION“.

Der **MTTR Advisor** simuliert das Checkpoint-Verhalten mit dem aktuellen Wert von „FAST\_START\_MTTR\_TARGET“ und vier weiteren abgeleiteten Werten. Die errechneten Statistiken, zum Beispiel über zusätzliches I/O, sind entweder grafisch oder über die View „V\$MTTR\_ADVICE“ zu finden. Der Parameter hat normalerweise den Wert „0“, somit ist der Advisor nicht aktiviert.

Der **SQL Repair Advisor** wird meistens von Consultants und Oracle-Support-Mitarbeitern verwendet, um Statements mit kritischen Fehlern zu analysieren und einen SQL-Patch zu generieren. Implementiert ist der Advisor über das Package „DBMS\_SQLDIAG“, das auch dabei helfen kann, Testfälle aufzubauen. Teilfunktionen sind grafisch in der Support Workbench des Enterprise Manager enthalten.

Der **Data Recovery Advisor** analysiert persistente „Failures“ wie Datenkorruptionen und gibt Ratschläge, wie

diese Probleme zu beheben sind. Die Nutzung erfolgt grafisch im Enterprise Manager, über RMAN, ADRCI oder das Package „DBMS\_HM“, mit dem manuelle „Health Checks“ durchgeführt werden können.

Der **SQL Access Advisor** gibt Informationen und Empfehlungen über fehlende oder überflüssige Zugriffsstrukturen wie Bitmap Indizes, Function Based Indizes, B\*tree Indizes, materialisierte Views etc. Eine wichtige Erweiterung erlaubt sogar die Bestimmung der Partitionierung für nicht partitionierte Tabellen. Die grafische Nutzung ist über Enterprise Manager möglich; das Package „DBMS\_ADVISOR“ liefert die Linemode-Implementierung.

Der **SQL Tuning Advisor** führt einen Optimizer-Analyse-Lauf durch und gibt Empfehlungen in folgenden Kategorien: „Statistiken“, „SQL Profiles“, „Zugriffsstrukturen“ und „Statementstruktur“. Grafisch ist die Funktion nicht nur über den Enterprise Manager, sondern auch über den SQL Developer nutzbar. Die Package-Implementierung wird über „DBMS\_SQLTUNE“ sichergestellt. Der SQL Tuning Advisor wird übrigens auch in anderen Technologien wie zum Beispiel dem SQL Performance Analyzer genutzt.

Der **Automatic Database Diagnostic Advisor (ADDM)** diagnostiziert die Datenbank-Performance-Probleme. Normalerweise werden diese Empfehlungen automatisch nach jedem WAR-Snapshot generiert. Auch die neuen Compare-Period-Reports im Real-Application-Testing-Umfeld oder Cloud Control 12c nutzen diese Informationen. Die Linemode-Nutzung kann über das Package „DBMS\_ADDM“ erfolgen.

**Voraussetzungen für Advisors**

Eine wichtige Voraussetzung zum Funktionieren einiger Advisors ist das Setzen des Parameters „STATISTICS\_LEVEL“ auf den Wert „TYPICAL“ oder „ALL“. Der empfohlene Defaultwert ist „TYPICAL“. Diese Einstellung ist notwendig, um das Sammeln von Advisor-Statistiken zu aktivieren, und hat keinen Einfluss auf die Lizenzierung von Packs. Die Einstellung „BASIC“ schaltet einige Advisors aus und kann auch nur bei manueller SGA-Verwal-

tung verwendet werden. Listing 1 zeigt den Aktivierungslevel bei der Einstellung „STATISTICS\_LEVEL=TYPICAL“.

Tabelle 1 gibt einen Überblick über Lizenzierung, Verwendbarkeit beziehungsweise Aktivierung in der aktuellen Datenbank-Version 11.2.0.3. Für die Verwendung von SQL Tuning, SQL Access Advisor und ADDM ist eine zusätzliche Lizenzierung erforderlich. Der Initialisierungsparameter „CONTROL\_MANAGEMENT\_PACK\_ACCESS“ muss darüber hinaus die richtige Einstellung aufweisen: „DIAGNOSTIC+TUNING“ für SQL Tuning und SQL Access Advisor

sowie mindestens „DIAGNOSTIC“ für das Funktionieren von ADDM. Die Default-Einstellung in Oracle 11g ist übrigens „DIAGNOSTIC+TUNING“.

**Aktivierung und Nutzung**

Alle Advisors lassen sich im Zeilenmodus über Packages oder Parameter-Einstellungen (siehe Tabelle 1) aktivieren und verwenden. Einige werden allerdings automatisch in der sogenannten „Automated Maintenance Task“ aktiviert. Ein Beispiel dafür ist der Segment Advisor. Die Analyse des Segment Advisors erfolgt in der Automated Main-

```
SQL> SELECT statistics_name, activation_level, session_settable,
           session_status, statistics_view_name
           FROM v$statistics_level ORDER BY 1;

STATISTICS_NAME          ACTIVAT SES  SESSION_ STATISTICS_VIEW_NAME
-----
Buffer Cache Advice      TYPICAL NO   ENABLED  V$DB_CACHE_ADVICE
Global Cache Statistics  TYPICAL NO   ENABLED
Longops Statistics       TYPICAL NO   ENABLED  V$SESSION_LONGOPS
MTTR Advice              TYPICAL NO   ENABLED  V$MTTR_TARGET_ADVICE
Modification Monitoring  TYPICAL NO   ENABLED
PGA Advice               TYPICAL NO   ENABLED  V$PGA_TARGET_ADVICE
...
Undo Advisor, Alerts and Fas TYPICAL NO   ENABLED  V$UNDOSTAT
t Ramp up
V$IOSTAT_* statistics    TYPICAL NO   ENABLED
...
```

Listing 1

	Lizenzierung	Nutzung
Memory	SE, EE	Einstellungen für die Memory-Bereiche, falls DB_CACHE_ADVICE=ON
MTTR	SE, EE	FAST_START_MTTR_TARGET=<wert>
Compression	SE, EE	DBMS_COMPRESSION
Segment	SE, EE	Automatische Maintenance Task oder DBMS_ADVISOR
Undo	SE, EE	DBMS_ADVISOR
SQL Tuning	EE plus Tuning Pack	Automatische Maintenance Task oder DBMS_SQLTUNE
SQL Access	EE plus Tuning Pack	Automatische Maintenance Task oder DBMS_ADVISOR
ADDM	EE plus Diagnostics Pack	Automatisch nach AWR-Snapshots oder DBMS_ADDM
SQL Repair	SE, EE	DBMS_SQLDIAG
Data Recovery	SE, EE	RMAN, ADRCI, DBMS_HM

Tabelle 1: Lizenzierung und Nutzung der Advisors. „SE“ steht für die Standard Edition und „EE“ für die Enterprise Edition. Zur Lizenzierung des Tuning-Packs sind als Grundlage die Enterprise Edition und das Diagnostics-Pack erforderlich.

tenance Task, manuell im Enterprise Manager oder über das PL/SQL-Package „DBMS\_ADVISOR“.

Alle Advisors bis auf den Compression Advisor können zusätzlich grafisch im Enterprise Manager aktiviert beziehungsweise genutzt werden. Die Homepage gibt einen Überblick über wichtige Ergebnisse der Segment- und Performance-Analyse, das Auftreten von Incidents beziehungsweise Informationen über die Instance-Recovery-Zeit. Zentrale Anlaufstelle zum Starten

der meisten Advisors ist der Link „Advisor Central“ oder „Advisors Home“. Data Recovery und SQL Repair Advisor finden sich im Bereich „Availability“ (unter „Perform Recovery“) oder im Support-Workbench. Abbildung 1 zeigt einen Ausschnitt aus der Advisor Central Page.

Bevorzugt man als Werkzeug den SQL Developer, gibt es ab Version 3.0 die Möglichkeit, im SQL Worksheet selbst einen SQL-Tuning-Advisor-Lauf anzustoßen und zusätzlich sogar die

Empfehlungen im Statistik-, SQL-Profil-, Index- oder Statement-Umfeld zu implementieren (siehe Abbildung 2). Auch hier sind natürlich eine ausreichende Lizenzierung und spezielle Zugriffsrechte wie das „ADVISOR“- und das „ADMINISTER SQL TUNING SET“-Privileg erforderlich.

Abgesehen von der Information zum SQL Tuning gibt es eine Vielzahl von Erweiterungen, die speziell für den DBA im SQL Developer implementiert sind. Der Link „View=>DBA“ öffnet ein zusätzliches DBA-Navigator-Fenster, das Informationen zu Datenbank, Konfiguration, Datenbank-Status, Data Pump, RMAN, Resource Manager, Scheduler, Security und Storage zur Verfügung stellt. Bleibt abzuwarten, welche Erweiterungen mit dem nächsten Datenbank-Release im SQL Developer zu finden sind.



Abbildung 1: Advisors im Enterprise Manager

```
SQL> SELECT window_name, resource_plan, repeat_interval, active,
         enabled
         FROM dba_scheduler_windows;
WINDOW_NAME                                RESOURCE_PLAN
-----
REPEAT_INTERVAL
-----
ACTIV ENABL
-----
MONDAY_WINDOW                                DEFAULT_MAINTENANCE_PLAN
freq=daily;byday=MON;byhour=22;byminute=0; bysecond=0
FALSE TRUE
```

Listing 2

```
BEGIN
  dbms_auto_task_admin.enable(client_name => 'sql tuning advisor',
                             operation => NULL, window_name => null);
END;
/
```

Listing 3

```
BEGIN
  dbms_auto_task_admin.enable()
END;
/
```

Listing 4

**Automated Maintenance Tasks**

Da die Automated Maintenance Tasks automatisch nach der Installation Verwendung finden können und die Nutzung heutzutage sicherlich zu den „Best Practices“ gehört, gibt dieser Abschnitt noch kurz einen Überblick über die Funktionsweise. Automated Maintenance Tasks sind Prozeduren, die Aufgaben zur Korrektur und Optimierung übernehmen, ohne dass der DBA eingreifen muss. Sie sind innerhalb eines Maintenance-Windows definiert und sollen ungehindert laufen, ohne eine Operation zu beeinträchtigen oder gar Objekte zu sperren. Listing 2 zeigt eine Abfrage danach, welche Windows wann aktiv sind.

Wichtig zu wissen ist, dass mit der Aktivierung eines Maintenance-Windows der „DEFAULT\_MAINTENANCE\_PLAN“-Ressource-Plan aktiv wird. Der Plan stellt 75 Prozent der CPU-Ressourcen für „SYS GROUP“-Operationen, 5 Prozent für „Diagnostic Operations“ und 25 Prozent für „Automated Maintenance Operations“ bereit.

Nach dem Ablauf der Maintenance Tasks wird beispielsweise das „Statistics Gathering“ durchgeführt. Zusätzlich werden Ergebnisse aufgelistet, in der Regel Empfehlungen zum Tuning oder Speicher-Management, die leicht über die Homepage im Enterprise Ma-

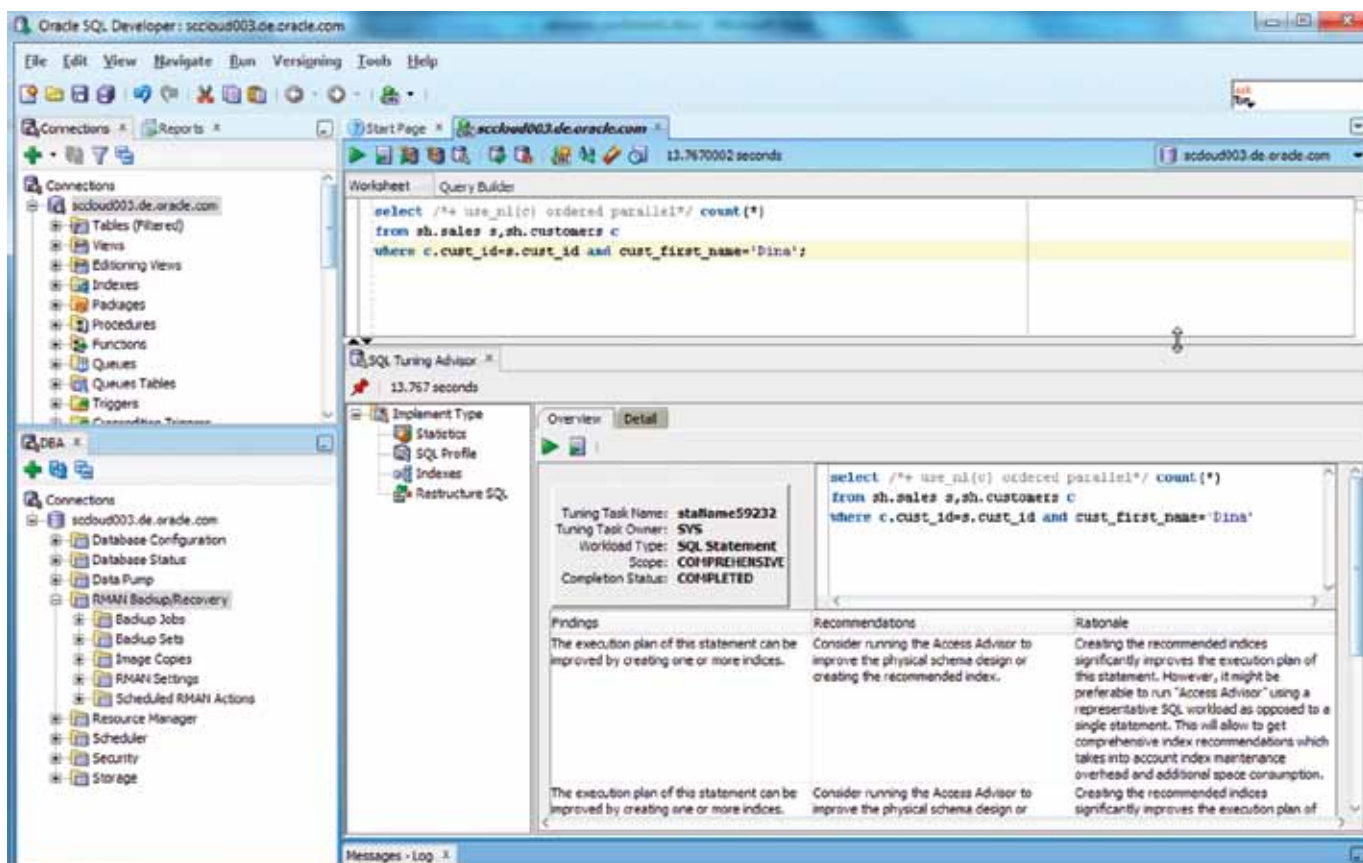


Abbildung 2: SQL Tuning im SQL\*Developer

nager zu finden sind. Diese können dann vom DBA durchgeführt und implementiert werden. Generell lassen sich die Automated Maintenance Tasks entweder im Enterprise Manager oder über einen Aufruf nachträglich konfigurieren und überprüfen (siehe Listing 3). Listing 4 zeigt das Einschalten aller Advisors. Einen Überblick über alle aktivierten Advisors in der Automated Maintenance Task zeigt die Abfrage in Listing 5.

**Fazit**

Das Advisory Framework bietet eine vielfältige Auswahl an Anwendungsmöglichkeiten. Jeder DBA sollte regelmäßig einen Blick auf die Homepage des Enterprise Manager werfen, um von den automatisch generierten Health-Check-Analysen für die Incidents und den Analysen des Segment Advisors, des SQL Tuning Advisors oder der ADDM-Analyse zu profitieren.

ren. Vermutet man darüber hinaus, dass gewisse Segmente oder Queries vom Tuninglauf oder der Komprimierung profitieren könnten, sollte man den entsprechenden Advisor manuell in der grafischen Oberfläche oder über PL/SQL-Packages anstoßen.

**Weitere Informationen**

- Tipps in der DBA Community: [http://blogs.oracle.com/dbacomcommunity\\_deutsch](http://blogs.oracle.com/dbacomcommunity_deutsch)
- Administrator Guide (Kapitel 19)
- SQL Developer Online Demonstration: SQL Tuning Advisor

```
SQL> SELECT client_name, status, attributes
       FROM dba_autotask_client;
CLIENT_NAME                                     STATUS
-----
ATTRIBUTES
-----
auto optimizer stats collection                 ENABLED
ON BY DEFAULT, VOLATILE, SAFE TO KILL
auto space advisor                             ENABLED
ON BY DEFAULT, VOLATILE, SAFE TO KILL
sql tuning advisor                             ENABLED
ONCE PER WINDOW, ON BY DEFAULT, VOLATILE, SAFE TO KILL
```

Listing 5

Ulrike Schwinn  
ulrike.schwinn@oracle.com



Oracle veröffentlicht vierteljährlich das Critical Patch Update (CPU). DBAs, die diesen CPU in ihrer Umgebung installieren wollen, sehen sich häufig kritischen Fragen vom Management ausgesetzt.

# Critical Patch Update – wie kritisch es wirklich ist

Katja Werner, OPITZ CONSULTING GmbH

Oracle gibt keine detaillierten Antworten auf Fragen wie „Welche Schwachstellen werden denn nun genau mit dem CPU behoben?“ oder „Welche Schäden können Angreifer anrichten, falls das CPU nicht installiert ist?“ Trotzdem kann mithilfe der von Oracle veröffentlichten Risk Matrix eine Beurteilung erfolgen. Der Artikel beschreibt – auch anhand eines Beispiels –, welche Hinweise dort stehen und wie das Risiko zu bewerten ist.

Vierteljährlich veröffentlicht Oracle sein CPU. Jedes Mal wird von Oracle empfohlen, diesen Patch einzuspielen, weil das Risiko, zum Opfer von Hacker-Angriffen zu werden, sonst sehr hoch sei. Jedes Quartal steht damit jeder verantwortungsvolle DBA vor der Frage, ob er das aktuelle CPU installieren soll.

Einiges spricht für das Einspielen des CPU: Wenn schon Sicherheitslücken behoben werden können, dann sollte das auch getan werden, um das

Risiko eines Angriffs so gering wie möglich zu halten. Einiges spricht dagegen: Das Management möchte keine Downtime der Datenbanken haben, die es insbesondere bei Single Instances für einen kurzen Reboot geben wird. Zudem besteht immer ein Rest-Risiko, dass sich die gefixte Oracle-Software anders verhält als vorher. Ob der CPU installiert werden muss, ist direkt von der Beantwortung folgender Fragen abhängig:

CVE#	Component	Protocol	Package and/or Privilege Required	Remote Exploit without Auth.?	CVSS VERSION 2.0 RISK (see Risk Matrix Definitions)							Supported Versions Affected	Notes
					Base Score	Access Vector	Access Complexity	Authentication	Confidentiality	Integrity	Availability		
CVE-2011-2301	Oracle Text	Oracle Net	Execute on CTXSYS, DRVDISP	No	8.5	Network	Medium	Single	Complete	Complete	Complete	10.1.0.5, 10.2.0.3, 10.2.0.4, 11.1.0.7,	See Note 1
CVE-2011-3525	Application Express	HTTP	APEX developer user	No	6.5	Network	Low	Single	Partial+	Partial+	Partial+	3.2, 4.0	
CVE-2011-3512	Core RDBMS	Oracle NET	Create session, create procedure, create table	No	6.5	Network	Low	Single	Partial+	Partial+	Partial+	10.1.0.5, 10.2.0.3, 10.2.0.4, 10.2.0.5, 11.1.0.7, 11.2.0.2	
CVE-2011-3511	Database Vault	Oracle Net	Privileged Account	No	3.6	Network	High	Single	None	Partial	Partial	10.2.0.3, 10.2.0.4, 10.2.0.5, 11.1.0.7, 11.2.0.2	
CVE-2011-2322	Database Vault	Oracle Net	SYSDBA	No	3.6	Network	High	Single	None	Partial	Partial	11.1.0.7	

Notes: 1. The CVSS Base Score is 8.5 only for Windows. For Linux, Unix and other platforms, the CVSS Base Score is 6.0, and the impacts for Confidentiality, Integrity and Availability are Partial+

Abbildung 1: Risk Matrix aus CPU Advisory von Oktober 2011 (Quelle: <http://www.oracle.com/technetwork/topics/security/cpuapr2012-366314.html#AppendixDB>)



1. Betreffen die mit dem CPU behobenen Schwachstellen mein System?
2. Falls „ja“, welche Folgen kann ein Angriff über diese Schwachstellen haben und kann ich es mir leisten, damit zu leben?

Antworten auf diese Fragen stehen in der von Oracle bereitgestellten Risk Matrix.

## Die Risk Matrix

Die Risk Matrix findet man im CPU Advisory, im Anhang unter der Rubrik „Oracle Database Server“. Auch für weitere Produkte wie Fusion Middleware, E-Business-Suite etc. sind im CPU Advisory „Risk Matrices“ veröffentlicht, auf die im Rahmen dieses Artikels nicht weiter eingegangen wird. Der Aufbau ist allerdings analog, sodass man beim Beurteilen der Schwachstellen für diese Produkte ähnlich vorgehen kann.

Die Risk Matrix ist der Dreh- und Angelpunkt bei der Bewertung der Sicherheitslücken. Hier sind alle Sicherheits-Schwachstellen einzeln beschrieben, die mit dem CPU behoben werden. Schwachstellen, die bereits in früheren CPUs dieser Oracle-Version behoben wurden, sind nicht mehr erwähnt, obwohl das aktuelle CPU auch diese mit behebt. Konkrete Informationen zu betroffenen Oracle-Libraries oder gar Exploits, also Erläuterungen der möglichen Angriffswege, erhält man in der Risk Matrix leider nicht.

Sofern man weiß, wie die Risk Matrix zu lesen ist, kann man schon ganz gut abschätzen, ob das aktuelle CPU für die eigenen Datenbanken relevant ist. Zum Beispiel kann man sehen, welche Oracle-Komponenten und -Versionen von Security-Bugs betroffen sind, ob ein Hacker-Angriff Daten-Manipulationen zur Folge haben kann oder ob eine Schwachstelle ohnehin nur mit hohen Rechten, wie zum Beispiel „SYS-DBA“, ausgenutzt werden kann. Mit diesen Angaben fällt es dem Verantwortlichen leichter zu entscheiden, ob er sein System ohne Änderung belässt und die Folgen eines möglichen Angriffs trägt oder ob er sich doch lieber

durch Installation des CPU schützen sollte. Manchmal gibt es auch alternative Möglichkeiten, Schwachstellen zu beheben, zum Beispiel, die entsprechende Komponente zu de-installieren, falls sie ohnehin nicht (mehr) genutzt wird.

Abbildung 1 zeigt ein Beispiel darüber, welche Informationen die Risk Matrix beinhaltet.

Für jeden mit dem CPU gefixten Bug werden die betroffene Oracle-Komponente sowie die betreffenden Versionen aufgeführt. Es ist zu beachten, dass hier nur aktuell supportete Versionen erwähnt sind. Dabei können durchaus auch ältere, nicht mehr unterstützte Versionen betroffen sein. Weiterhin gibt es in der Risk Matrix in der Spalte „Package and/or Privilege Required“ Angaben zu Packages beziehungsweise Privilegien, die für einen Angriff vorhanden sein müssen. Eine weitere wichtige Information gibt der Hinweis, ob der Angriff ohne Angabe von Username und/oder Passworten aus dem Netzwerk durchführbar ist. Diesen Hinweis findet man in der Spalte „Remote Exploit without Auth.?“.

Ein besonderes Augenmerk gilt den Spalten der Risk Matrix, die mit „CVSS VERSION 2.0 RISK“ betitelt sind. Der Industriestandard CVSS wurde entwickelt, um Sicherheitsrisiken unterschiedlicher Applikationen, Datenbanken, Hard- und Software nach einheitlichen Kriterien beurteilen zu können und einen direkten Vergleich zu ermöglichen. Ein Security-Verantwortlicher kann auf einen Blick sehen, in welchen Systemen seiner IT-Landschaft das Security-Risiko hoch ist und wo die Prioritäten beim Bug Fixing liegen sollten. Die Einflussfaktoren aus dem CVSS, die in der Risk Matrix von Oracle betrachtet werden, sind folgende:

- Base Score
- Angriffsweg
- Komplexität des Exploit
- (Häufigkeit der) Authentifizierung
- Vertraulichkeit (der Daten)
- Integrität (Unversehrtheit/Unverändertheit/Verlässlichkeit der Daten)
- Verfügbarkeit (der Daten)

## Libelle SystemCopy



- ✓ Ohne in Ihre SAP-Umgebung einzugreifen bzw. diese zu verändern
- ✓ Ohne aufwändige Vorplanung
- ✓ Mit minimaler Durchlaufzeit
- ✓ Bei gleichbleibender Qualität der Kopie

... mit deutlich reduzierten Prozesskosten



Hans-Joachim Krüger  
Chief Technology Officer  
Libelle AG

Erfahren Sie mehr:  
[www.Libelle.com/systemcopy](http://www.Libelle.com/systemcopy)



ORACLE Gold Partner



Libelle

Libelle AG

Gewerbestr. 42 • 70565 Stuttgart, Germany  
T +49 711 / 78335-0 • F +49 711 / 78335-148  
[www.Libelle.com](http://www.Libelle.com) • [sales@libelle.com](mailto:sales@libelle.com)

Der Base Score ist der Haupt-Indikator für die Ersteinschätzung des Risikos. Er wird anhand einer festen Formel aus weiteren genannten Faktoren errechnet. Diesen CVSS-Rechner findet man im Internet, um selbst Base Scores nachrechnen zu können. Für eine Beurteilung nach dem Industriestandard CVSS 2.0 sind für die drei oben genannten Faktoren Vertraulichkeit, Integrität und Verfügbarkeit folgende Bewertungen möglich:

- *None*  
Keine Auswirkung bei Angriff möglich
- *Partial*  
Daten können teilweise durch Angreifer eingesehen/verändert beziehungsweise in ihrer Verfügbarkeit eingeschränkt werden
- *Complete*  
Alle Files auf dem Rechner können durch Angreifer eingesehen/verändert beziehungsweise kann ihre Verfügbarkeit beeinflusst werden

Oracle hat diesen Standard um seine eigene, vierte Bewertungsstufe „Partial+“ ergänzt. Der Hersteller begründet dies damit, dass die Bewertung „Complete“ nicht genommen werden könne, wenn nur die Datenbank beziehungsweise große Teile davon betroffen seien. „Complete“ stünde für die Übernahme des gesamten Rechners. Hier heißt es: „Aufpassen!“ Mag diese Sichtweise für einen Vergleich mehrerer Systeme innerhalb einer IT-Landschaft noch sinnvoll sein, so ist sie doch falsch, wenn die Gefährdung der Datenbank beurteilt werden soll. Bei Angriffen auf die Datenbank ist es nämlich unerheblich, ob nur die Datenbank oder auch ihr Host (der ja oft nur dafür sorgt, dass die Datenbank laufen kann) betroffen sind. Um die Risiken hier also richtig einordnen zu können, sollte durchaus zusätzlich noch einmal ein korrigierter Base Score berechnet werden, in dem die „Partial+“-Bewertungen in der Risk Matrix durch ein „Complete“ ersetzt werden.

Wie aber kann nun das Risiko anhand der Risk Matrix bewertet werden? Im Folgenden soll dies an einem Beispiel gezeigt werden. Abbildung 1 zeigt

die Risk Matrix für den Datenbank-Server aus dem CPU Advisory von Oktober 2011. Diese ist nicht mehr aktuell, bietet aber einige interessante Aspekte für die Auswertung.

#### **Ein erster Eindruck vom CPU Oktober 2011**

Zuerst verschaffen wir uns einen groben Überblick über die Schwachstellen des CPU. Ein erster Blick auf die Spalte mit den Oracle-Komponenten der Risk Matrix zeigt, dass von fünf gepatchten Schwachstellen nur eine das Core-RDBMS betrifft. Diese Sicherheitslücke bezieht sich auf alle Datenbanken und in vielen Umgebungen wird es auch die einzige relevante Schwachstelle sein. Zudem betrifft dieser Bug alle unterstützten Oracle-Versionen und vermutlich auch ältere, die nicht mehr dem Support unterliegen.

Als Nächstes schauen wir auf die Base Scores. Diese können sich zwischen 0 und 10 bewegen – je höher dieser Wert, desto leichter ist ein Angriff mit kritischen Folgen hinsichtlich Daten-Manipulation und provozierter Auszeiten möglich. Unser maximaler Base Score beträgt 8,5 für Datenbanken, die Oracle Text im Einsatz haben. Auch eine Schwachstelle in Apex sowie die für das Core-RDBMS haben relativ hohe Base Scores von 6,5.

Der nächste Check betrifft die Spalte „Remote Exploit without Auth.?“. Dort sehen wir, dass keine der fünf Schwachstellen über das Netzwerk und ohne Authentifizierung angreifbar ist. Hätte hier irgendwo ein „YES“ gestanden, so sollte man sich die betroffene Schwachstelle sehr genau anschauen, denn Angriffe aus der Ferne ohne die Eingabe gültiger Credentials erweitern den Kreis potenzieller Hacker um ein Vielfaches.

Nun haben wir eine erste Übersicht über die mit dem CPU behobenen Bugs und deren Risiken, falls sie nicht gepatcht werden. Jetzt können wir für die Sicherheitslöcher einzeln und nacheinander, Spalte für Spalte, die Risk Matrix auslesen. Ein besonderes Augenmerk liegt dabei auf den Schwachstellen, die uns beim Überblick auffielen. Für das CPU von Ok-

tober 2011 sind dies die Sicherheitslöcher in Oracle Text, Apex und im Core-RDBMS.

#### **Bewertung der Schwachstelle**

##### **CVE-2011-2301**

Die Schwachstelle mit dem höchsten Base Score in unserem Beispiel-CPU ist CVE-2011-2301, betroffen ist Oracle Text. Der Angriff kann über Oracle Net erfolgen, also auch von anderen Oracle-Clients aus. Der Bug betrifft das Package „CTXSYS.DRVDISP“.

Der Base Score von 8,5 spricht für sich und legt es nahe, das CPU zu installieren, wenn Oracle Text im Einsatz ist. Die Schwachstelle ist auf mittelschwerem Weg (Access Complexity = Medium) und mit nur einer einzigen Authentifizierung angreifbar.

Folge eines Angriffs wäre, dass alle Daten der Maschine und der Datenbank eingesehen (Confidentiality = Complete) und nach Lust und Laune verändert (Integrity = Complete) werden könnten. Daneben könnten bewusst Auszeiten provoziert werden. Die betroffenen Oracle-Versionen sind 10.1.0.5, 10.2.0.3, 10.2.0.4 und 11.1.0.7. Zum Zeitpunkt des Erscheinens des CPU entsprach dies allen unterstützten Versionen bis auf 10.2.0.5, 11.2.0.2 und 11.2.0.3. Die letzte Spalte in der Risk Matrix – die Notes – enthält zusätzliche Bemerkungen. Für den Bug in Oracle Text lautet diese Bemerkung, dass der Base Score von 8,5 nur für Windows-Systeme gilt; Unix, Linux und andere Betriebssysteme bekommen 6,0, was auch schon recht hoch ist.

Zu diesem Bug sollte man ergänzen, dass der Base Score bei Erscheinen des CPU mit 4,1 angegeben war und erst ein paar Tage später korrigiert wurde. Zeitgleich damit wurden dann auch die CVSS-Faktoren „Vertraulichkeit“, „Integrität der Daten“ sowie „Verfügbarkeit“ von „Partial+“ auf „Complete“ geändert. Es ist also zum einen durchaus sinnvoll, im CVSS-Rechner die Höhe des Base Score selbstständig zu prüfen, indem man für „Partial+“ „Complete“ einsetzt. Zum zweiten ist es von Vorteil, noch einmal ein paar Tage nach Erscheinen des CPU zu prüfen, ob sich seit dem Erschei-

nungsdatum Änderungen ergeben haben.

**Bewertung der Schwachstelle**

**CVE-2011-3525**

Die zweite Schwachstelle (CVE-2011-3525) betrifft die Apex-Komponente der Datenbank. Apex wird bei neueren Datenbank-Versionen per Default mitinstalliert. Auch wenn man kein Apex nutzt, sollte man also nochmals kontrollieren, ob es auch wirklich nicht installiert ist. Angriffe erfolgen laut Risk Matrix über HTTP, also ziemlich sicher über die Apex-Oberfläche. Der Base Score von 6,5 ist recht hoch – daraus lässt sich schließen, dass ein leicht durchzuführender Angriff zu Datenmanipulation beziehungsweise Downtimes in großem Stil führen kann. Setzt man für die Faktoren „Vertraulichkeit“, „Verlässlichkeit“ und „Verfügbarkeit“ anstelle des „Partial+“ „Complete“, so errechnet sich ein sehr hoher Base

Score von 9,0. Das sollte man bei der Beurteilung der Folgen eines Angriffs im Hinterkopf haben. Die betroffenen Apex-Versionen sind 3.2. und 4.0. Aus all diesen Interpretationen kann man schlussfolgern, dass gepatcht werden sollte. Zu Konfusion kann die Frage führen, wie das geschehen soll: Denn hier kann nicht das CPU für die Datenbank eingespielt werden, sondern es muss ein Upgrade auf mindestens Apex 4.1.0.00.32 oder höher erfolgen. Details dazu kann man unter „Patch Set Update and Critical Patch Update October 2011 Availability Document“ nachlesen.

**Bewertung der Schwachstelle**

**CVE-2011-3512**

Als letztes Beispiel soll in diesem Artikel noch auf die Schwachstelle 2011-3512 im Core-RDBMS eingegangen werden. Aus der Risk Matrix kann man entnehmen, dass Angriffe über das

Netzwerk mit den Privilegien „create session“, „create procedure“ und/oder „create table“ erfolgen können. Ob diese Rechte in Kombination oder einzeln vorhanden sein müssen, kann leider nicht gesagt werden. Es ist aber zu vermuten, dass alle drei Rechte in Kombination vorliegen müssen, denn bei den meisten anderen Schwachstellen, die nur „Create-Session“-Privilegien nutzen, steht auch nur dieses Recht in der Risk Matrix. Folgen eines Angriffs sind unerlaubte Einsicht und Manipulation weiterer Datenbestände in der Datenbank sowie Provokation von Ausfällen. Betroffen waren alle Oracle-Versionen bis auf die damals gerade herausgekommene 11.2.0.3, in der sicherlich der Bug-Fix schon enthalten war. Auch bei diesem Bug änderte Oracle ein paar Wochen später den Wert für den Ausfall der Datenbank von „None“ auf „Partial+“ und damit stieg der Base Score von 5,5 auf



IT-Consulting	Schulungen	Software-Lösungen	Oracle Lizenzen
<ul style="list-style-type: none"> <li>› Performance Tuning                             <ul style="list-style-type: none"> <li>• Oracle Datenbank Tuning</li> <li>• Oracle SQL + PL/SQL Tuning</li> </ul> </li> <li>› Real Application Clusters</li> <li>› Data Guard + Fail Safe</li> <li>› Datenbank Management                             <ul style="list-style-type: none"> <li>• Konfiguration</li> <li>• Backup &amp; Recovery</li> <li>• Migration und Upgrade</li> </ul> </li> <li>› OEM Grid Control</li> <li>› Oracle Security</li> <li>› Services                             <ul style="list-style-type: none"> <li>• Remote DBA Services</li> <li>• Telefon-/Remotesupport</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>› Oracle SQL</li> <li>› Oracle PL/SQL</li> <li>› Oracle DBA</li> <li>› Oracle APEX</li> <li>› Backup &amp; Recovery</li> <li>› RMAN</li> <li>› Neuerungen 10g/11g</li> <li>› Datenbank Tuning</li> <li>› Datenbank Monitoring</li> <li>› Datenbank Security</li> </ul> <p>Wir bieten Ihnen öffentliche Kurse sowie Inhouse-Schulungen.</p>	<ul style="list-style-type: none"> <li>› Individualsoftware                             <ul style="list-style-type: none"> <li>• .NET und Visual Basic</li> <li>• Java</li> </ul> </li> <li>› Oracle APEX</li> <li>› PL/SQL</li> </ul> <p>Unser Ziel: Individuelle Softwareentwicklung mit Fokus auf Ihre Zufriedenheit.</p>	<ul style="list-style-type: none"> <li>› Oracle Datenbanken                             <ul style="list-style-type: none"> <li>• Standard Edition One</li> <li>• Standard Edition</li> <li>• Enterprise Edition</li> <li>• Personal Edition</li> </ul> </li> <li>› Oracle Produkte                             <ul style="list-style-type: none"> <li>• Enterprise Manager</li> <li>• Oracle Tools</li> </ul> </li> </ul> <p>Optimale Lizenzierung durch individuelle Beratung.</p>

Nutzen Sie unsere Kompetenz für Ihre Oracle Datenbanken.



**Aus CPU wird SPU**

Seit Oktober 2012 gibt es eine neue Bezeichnung für den Critical Patch Update (CPU): Security Patch Update (SPU). Bis auf den Namen bleibt (vorerst) alles gleich und es gibt weiterhin die Risk-Matrix für die Abschätzung der Risiken. Den Begriff „Critical Patch Update“ gibt es weiterhin – er steht jetzt für die Gesamtheit der vierteljährlich veröffentlichten Patches wie Patch Set Update (PSU), Security Patch Update (SPU) und Bundle Patches.

6,5. Wie auch immer – schon ein Base Score von 5,5 legt nahe, das CPU zu installieren.

**Die Empfehlung**

Alle diese Informationen der Risk Matrix – grobe Auswertung sowie detaillierte Auswertung der kritischsten drei Schwachstellen, die mit dem CPU von Oktober 2011 behoben werden, – führen zur Empfehlung, dieses CPU zu installieren, denn die Folgen eines Angriffs sind gravierend.

Manchmal kommt man bei der Auswertung der Risk Matrix auch zu anderen Schlüssen. So etwa für das CPU von April 2012: Wenn das betrachtete System die Datenbank-Version 11.2.x hat, Apex und Spatial nicht installiert sind und die Datenbank nicht auf Windows läuft, dann läge die einzige zu patchende Security-Schwachstelle im Core-RDBMS. Der Base Score dafür beträgt 4,0. Angreifer mit dem „Create-Session“-Privileg könnten Downtimes verursachen. Dieses Risiko wäre für nicht businesskritische Systeme eventuell vertretbar, sodass man sich den Aufwand zum jetzigen Zeitpunkt sparen könnte und stattdessen erst das nachfolgende CPU installiert, das ja kumulativ ist, also auch die Patches vom April beinhaltet.

**Internet-Recherche**

Zum Abschluss (manchmal auch gleich zu Anfang) ist sicherlich eine Recherche im Internet hilfreich. Zum einen kann man dort die eigene Einschätzung bezüglich der Wichtigkeit

des betrachteten CPU überprüfen, zum anderen findet man hier interessante Bug-Nummern – und wenn es schon Informationen über Exploits oder andere tiefergehenden Kommentare gibt, lässt sich noch besser beurteilen, wie wichtig das Patchen mit dem aktuellen CPU ist.

**Wie vollständig das CPU ist**

Wenn nun anhand von Risk Matrix, Internet und eventuell weiteren Informationsquellen beurteilt wurde, ob das CPU installiert werden soll oder ob die Risiken im Falle eines Angriffs auch ohne Installation des CPU vertretbar sind, ist es an der Zeit, den Blick wieder etwas weiter schweifen zu lassen und ein paar andere Aspekte zu überdenken. Dabei sollte zunächst geprüft werden, ob auch Oracle-Clients mit dem für sie notwendigen CPU versehen sind. Manchmal können Schwachstellen auch über ungepatchte Clients ausgenutzt werden, obwohl der Datenbank-Server ordnungsgemäß Updates erhalten hat. Ob Oracle-Clients gepatcht werden müssen, steht oberhalb der Risk Matrix für den Oracle Database Server im CPU-Advisory. Eine weitere Informationsquelle ist die Readme-Datei zum aktuellen CPU. Dort steht im Kapitel „Patch Information“ welches CPU auf Clients installiert sein sollte, damit auch über diese keine Angriffe stattfinden können.

Darüber hinaus ist es wichtig zu prüfen, ob es noch weitere Security Alerts gab und gibt und ob das System dagegen geschützt ist. Sinnvoll ist hier auch die Aktivierung von Newsletters oder RSS-Feeds, damit man solche Informationen zeitnah bekommt. Hier gab es zum Beispiel im April 2012 den CVE-2012-1675, der unter dem Namen „TNS Listener Poison Attack“ bekannt wurde. Zu diesem Security Alert gibt es keine Fixes im Rahmen des CPU, da dies eine Änderung der Funktionalität wäre und so etwas grundsätzlich nicht in CPUs geliefert wird. Trotzdem sollte man sich gegen diese Schwachstelle schützen, denn sonst kann ein Angreifer eine „Dummy“-Instance nutzen, um Clientsessions über die eigene Maschine zu routen, dort die Daten abzugreifen beziehungsweise auch beste-

hende Sessions zu übernehmen und Daten auf der Zieldatenbank zu manipulieren. Betroffen sind alle Datenbank-Releases seit 8i.

**Wie kritisch es wirklich ist**

Keine Frage: Security-Schwachstellen mit dem CPU zu schließen, ist gut. Trotzdem muss man immer bedenken, dass auch das aktuelle CPU nichts hilft, wenn die Sicherheitslöcher an anderer Stelle sperrangelweit geöffnet sind. Dies hier zu erörtern, würde zu weit führen, aber eine Auswahl von Denkanstößen in ungeordneter Folge soll hier trotzdem dargestellt werden:

- Haben die User wirklich nur die Rechte, die sie auch brauchen?
- Wird der Datenzugriff auditiert? Sind auch die Auditeinträge selbst geschützt, und werden sie ausgewertet?
- Sind die Backups der Datenbank vor unbefugtem Zugriff geschützt?
- Kann jedes Login eines Users in der Datenbank bis zur arbeitenden Person selbst zurückverfolgt werden?
- Wird die IT-Landschaft überwacht, sodass die unbefugte Installation von Software, insbesondere Dummy-Datenbank, ungepatchte Oracle Clients, Sniffing-Tools etc. erkannt wird?
- Sind die Passwörter für Datenbank und Maschine sicher?
- Sind alle nicht benötigten User gelockt beziehungsweise wurde nicht benötigte Software deinstalliert?

**Fazit**

Mithilfe der Risk Matrix von Oracle kann schon ein gutes Stück weit eingeschätzt werden, ob das CPU für das jeweilige System installiert werden muss oder ob die Folgen eines Hacker-Angriffs auf das ungepatchte System vertretbar sind. Eine zusätzliche Recherche im Internet vervollständigt das Bild und deckt eventuell schon veröffentlichte Exploits zu den Sicherheitslöchern auf, die mit dem CPU behoben werden. Und: Datenbanken mit dem aktuellen CPU nützen nichts, wenn nicht auch andere Security-Regeln implementiert sind.

## Patches – eine Übersicht

Bei Oracle gibt es vielfältige Patch-Arten und -Varianten. Um die Orientierung etwas zu erleichtern, hier ein Überblick über die verschiedenen Ausprägungen. In vielen der angesprochenen Patches sind auch Security-Patches enthalten, oft aber ist der Umfang weit größer.

### Patch Sets

Ein Patch Set ist eine Sammlung vieler Patches, die funktionale und Security-bezogene Schwachstellen beheben. Patch Sets bringen oft auch neue Funktionalitäten mit sich. Bei älteren Oracle-Releases wird das Patch Set auf der bestehenden Oracle-Version installiert, ab Version 11g R2 handelt es sich dann um in sich abgeschlossene Installations-Sources. Das Patch Set wird mit dem Oracle Installer installiert.

### Patch Set Update

Ein Patch Set Update (PSU) ist – wie der Name schon sagt – das Update für das Patch Set. Es umfasst ebenfalls eine Sammlung von Patches. Im Gegensatz zu einem Patch Set werden jedoch nur wichtige funktionale Fehler und Security-Schwachstellen behoben. Die Anzahl der enthaltenen Patches ist entsprechend geringer. Es finden keinerlei funktionale Änderungen statt. Auch Neuerungen, die zu Modifikationen im Explain Plan führen, sind nicht enthalten. Deshalb empfiehlt es sich, über das PSU hinaus zu prüfen, ob es weitere empfohlene Patches gibt, die wichtige funktionale Änderungen beheben, aber Anpassungen von Explain Plans nach sich ziehen können. Solche Patches sind nie im PSU enthalten, sondern immer separat zu installieren. Die PSUs erscheinen für Unix- und Linux-Systeme. Für Windows-Systeme sind sie nicht separat erhältlich, sondern nur im Paket mit weiteren funktionalen Änderungen im Rahmen eines Windows-Bundle-Patch. Es ist zu beachten, dass im aktuellen Windows-Bundle-Patch

nicht immer die neuesten PSUs enthalten sind. PSUs erscheinen vierteljährlich. Die Installation erfolgt mit dem OPatch-Tool, das immer aktuell sein sollte (siehe Metalink Note ID 224346.1). Bei der Installation eines PSU steigt die Release-Nummer der Datenbank an der fünften Stelle.

### Critical Patch Update/Security Patch Update

Critical Patch Update (CPU) hieß es früher. Seit Oktober 2012 wird von Oracle ein neuer Begriff eingeführt: Security Update Patch. Der Inhalt ist jedoch der Gleiche. In diesem Absatz wird bereits der aktuelle Begriff „Security Patch Update“ verwendet. Ein Security Update Patch (SPU) enthält im Gegensatz zum PSU keine funktionalen Bug-Fixes, sondern nur Patches für Sicherheits-Schwachstellen. Ein SPU wird zum gleichen Zeitpunkt wie ein PSU veröffentlicht und ist in diesem auch enthalten. Auch SPUs gibt es einzeln nur für Unix und Linux; für Windows-Systeme sind sie in den Windows-Bundle-Patches enthalten, allerdings auch hier nicht immer aktuell. Die Installation erfolgt ebenfalls mit dem OPatch-Tool. Die Release-Nummer der Datenbank bleibt unverändert. Die Tabelle unter [www.doag.org/go/doagnews/werner\\_tabelle](http://www.doag.org/go/doagnews/werner_tabelle) zeigt PSU und SPU im Vergleich.

### Interim Patch, One-Off-Patch oder Patch Set Exception

Es tauchen häufig auch noch die Wörter „Interim Patch“, „One-Off-Patch“ oder „Patch Set Exception“ (PSE) auf. Damit ist ein Patch für eine Schwachstelle (manche Quellen sprechen auch von mehreren Schwachstellen) gemeint. Laut Oracle Support ist auch ein CPU ein Interim-Patch. Für die Windows-Welt gibt es generell keine Interim-, sondern nur Bundle-Patches.

### Windows-Bundle-Patch

Im Windows-Umfeld gibt es von all den oben genannten Patch-Arten nur das

Patch-Set. Sollen Systeme gegen Security-Schwachstellen geschützt werden, hat man nur die Möglichkeit, einen Windows-Bundle-Patch zu installieren. Dieser enthält sowohl CPU und PSU als auch weitere funktionale Bug-Fixes und ist damit sehr umfangreich. Deshalb sollten hier auch unbedingt funktionale Tests erfolgen, bevor der Bundle-Patch auf produktiven Systemen in Betrieb geht. Es ist nämlich deutlich aufwändiger, ein Windows-System mit den aktuellen Security-Bug-Fixes zu versorgen, als Datenbanken auf Linux oder Unix. Hinzu kommt, dass im aktuellen Windows-Bundle-Patch nicht immer das letzte CPU enthalten ist. Oracle stellt den Windows-Bundle-Patch nicht vierteljährlich, sondern in unregelmäßigen Abständen zur Verfügung.

### Merge Patch, Prerequisite (Overlay) Patch

Weitere Begriffe, die man oft findet, die hier aber nur kurz beschrieben werden sollen, sind Merge Patch und Prerequisite (oder auch: Overlay) Patch. Dies sind Patches, die von Oracle zur Verfügung gestellt werden, wenn es Konflikte zwischen verschiedenen Patches gibt. Die Installation erfolgt mit OPatch. Die Bezeichnung Prerequisite oder Overlay Patch wird nur in Verbindung mit dem PSU verwendet. Es ist ein adaptierter Interim Patch, der nicht mehr mit dem PSU in Konflikt steht. Bei der Installation muss das PSU vor dem Prerequisite Patch installiert werden.

### On Request Patch

Ein On Request Patch ist ein PSU oder CPU, der vom Kunden für sein Betriebssystem angefordert werden muss. Die besondere Anforderung ist notwendig, weil Oracle das PSU/CPU nur für die gängigsten Kombinationen von Betriebssystem und Oracle-Version automatisch zur Verfügung stellt.

## Referenzen

1. Critical Patch Updates, Security Alerts and Third Party Bulletin: <http://www.oracle.com/technetwork/topics/security/alerts-086861.html>
2. Oracle Database Server Risk Matrix: <http://www.oracle.com/technetwork/topics/security/cpuoct2011-330135.html#AppendixDBRiskMatrix>
3. Glossary – terms and definitions for Critical Patch Update risk matrices: <http://www.oracle.com/technetwork/topics/security/advisorymatrixglossary-101807.html>
4. Complete Guide to the Common Vulnerability Scoring System Version 2.0: <http://www.first.org/cvss/cvss-guide.html>
5. Common Vulnerability Scoring System Version 2 Calculator – <http://nvd.nist.gov/cvss.cfm?calculator&version=2>



Katja Werner  
katja.werner@opitz-consulting.com

Administratoren, die Linux als Betriebssystem für ihre Oracle-Datenbank-Installation nutzen möchten, sind sich häufig nicht sicher, welches Linux sie nehmen sollen. In dieser Situation begeben sich die IT-Verantwortlichen auf die Suche nach ausschlaggebenden Kriterien für ihre Wahl. Da die Anwender eine Datenbank viele Jahre nutzen, bilden fundierte Kenntnisse und Erfahrungen zu dem jeweiligen Betriebssystem eine wichtige Voraussetzung für den zuverlässigen Betrieb.

## Oracle Linux versus Red Hat – lohnt sich der Umstieg auf Oracle Linux?

Andrew Lacy, OPITZ CONSULTING GmbH

Wesentliche Entscheidungsfaktoren für oder gegen ein bestimmtes Betriebssystem können durchaus das Know-how und die Expertise der Administratoren sein. Dies sollten Unternehmen bei der Wahl berücksichtigen. Wenn die IT-Mitarbeiter beispielsweise nur Microsoft Windows kennen, empfiehlt es sich, die Oracle-Installation darauf vorzunehmen. HP-UX, das Unix-Betriebssystem von Hewlett Packard, stellt hier eine Ausnahme dar: Da Oracle den Itanium-Support in der nächsten Hauptversion der Datenbank (siehe My Oracle Support ID 1307745.1) einstellen will, könnte eine Entscheidung für diese Lösung tatsächlich zum Problem werden. HP verkauft den PA-RISC-Chip seit Ende 2008 nicht mehr (siehe My Oracle Support ID 1313798.1).

### Oracle Linux: Mit Standard-Systemkernen weniger unterstützt

Oracle hat sein Linux kürzlich umbenannt. Es heißt nun „Oracle Linux (OL)“ und nicht mehr „Oracle Enterprise Linux (OEL)“. Oracle bietet sein Linux heute mit zwei verschiedenen Linux-Systemkernen an: dem „Red Hat Linux“-Systemkern und einer Oracle-Version des Linux-Systemkerns „Unbreakable Enterprise Kernel (UEK)“. Bei den neueren Oracle-Linux-Versionen ist UEK standardmäßig installiert.

Bezüglich der Frage, ob eine Non-Oracle-Applikation für OEL zertifiziert ist, sagt Oracle: „Oracle Linux maintains user-space compatibility with Red Hat Enterprise Linux, which is independent of the kernel version run-

ning underneath the operating system. The existing applications will continue to run unmodified on Unbreakable Enterprise Kernel Release 2 and no re-certifications are needed for RHEL certified applications“. Natürlich sind alle Oracle-Applikationen für Oracle Linux zertifiziert. Interessant ist in diesem Zusammenhang, dass SAP seit Kurzem auch für Oracle Linux zertifiziert ist (siehe <https://service.sap.com/sap/support/notes/1635808>).

Oracle kündigte jetzt mehrere Features für die Datenbank an, die nicht länger auf dem Systemkern von Red Hat Linux, sondern zukünftig nur noch auf dem Oracle-Systemkern (UEK) zur Verfügung stehen werden. Wer mit Oracle einen Supportvertrag für ein RHEL-Linux abschließt, darf den Oracle-Systemkern auf Red Hat Enterprise Linux (RHEL) installieren. Mit der Installation des Oracle-Systemkerns auf einem RHEL-Server ist der Supportvertrag von Red Hat allerdings nicht mehr gültig.

Oracle-Anwender schenken dieser Praxis erst seit kurzer Zeit Beachtung, weil Oracle die Datenbank auf RHEL6 und OL6 (siehe My Oracle Support ID 1350000.1) bis zuletzt nicht unterstützte. Die Entscheidung kam für viele Administratoren sehr überraschend. Schließlich hatte Oracle den Support für RHEL6 und OL6 mit der Version 11.2.0.3.0 aufgekündigt, weil man die Version 11.2.0.4.0 nicht bis ins zweite Quartal 2013 (siehe My Oracle Support ID 742060.1) hinein vorgesehen hatte. Der neue Enterprise Manager Cloud Control 12c (EMCC) war das

erste Oracle-Produkt, das mit Support für OL6 zu bekommen war. Oracle zertifizierte Datenbank und Enterprise Manager für OL6 einige Monate lang, bevor man sich endlich auch zu einer Zertifizierung für RHEL6 entschloss.

Für RHEL6 bietet Oracle die Support Library ASMLib und Updates nur an, wenn das System mit dem von ihm vertriebenen Systemkern konfiguriert wurde. Anwender, die ASMLib auf ihrem RHEL6-Server laufen lassen, sind also auf den Support von Oracle für diesen Server angewiesen und gezwungen, den Oracle-UEK-Systemkern zu installieren (siehe My Oracle Support ID 1089399.1). Auch wenn ASMLib ein sehr komfortables Tool ist: Ein alternatives Device-File-System wie „udev“ ermöglicht dem Nutzer, vom Systemkern unabhängig zu bleiben (siehe My Oracle Support ID 414897.1).

### Die Grenzen verlaufen zwischen den Herstellern

Oracle-Produkte sind nicht zertifiziert, um auf virtuellen Maschinen und Gast-Systemen betrieben zu werden, die mit den Xen- oder KVM-Angeboten von Red Hat, Novell oder XenSource bereitgestellt sind (siehe My Oracle Support ID 417770.1). Oracle-Produkte sind nicht zertifiziert, um auf Betriebssystemen wie Red Hat, Novell SLES oder Citrix XenServer Enterprise Edition Xen Hypervisors zu laufen. In einer virtuellen Umgebung unterstützt Oracle eigene Produkte durchaus auch ohne Zertifizierung (siehe My Oracle Support ID 417770.1).

Das Cluster File System OCFS2 1.4 ist auf RHEL 6 erhältlich. Alle neueren Versionen von OCFS2 (momentan Version 1.8) benötigen den Oracle-Systemkern (UEK) und den entsprechenden Support des Herstellers (siehe My Oracle Support ID 1253272.1). Automatic Storage Management (ASM) und ASM Cluster File System (ACFS) stellen in vielen Fällen umsetzbare Alternativen zu OCFS2 dar.

### Oracle Linux und RHEL im direkten Vergleich

Beim Kauf der Open-Source-Extension „Ksplice“ bietet Oracle jetzt ein Feature an, mit dessen Hilfe Systemkern-Updates ohne Neubooten möglich sind. Diese Systemkern-Updates sind nur mit UEK und einem Oracle-Linux-Supportvertrag erhältlich. Dieser Dienst ist besonders in Kombination mit einem anderen Oracle-Linux-Feature interessant: Linux Containers. Diese bieten eine Linux-Installation, die zentral verbunden ist. Viele Hosts nutzen diese einzelne Installation. Natürlich bedeutet eine einzelne Installation im Vergleich zu mehreren Installationen weniger Aufwand. Besonders für größere Installationen, bei denen so wenig Downtime wie möglich anfallen soll, kann dieses Feature sehr interessant sein.

Das voreingestellte Dateisystem für OL6 ist „btrfs“ und nicht „ext4“ wie auf RHEL6. „btrfs“ ist ein von Oracle entwickeltes Dateisystem, das seit der Version 2.6.29 im Hauptsystemkern enthalten ist. Es bietet erweiterte Funktionalitäten

wie Built-in RAID Support, Snapshotting, Compression und Encryption.

Die Funktion „Database Smart Flash Cache“ ist auf Solaris und Oracle-Enterprise-Linux lieferbar. Mit dieser Funktion kann der Benutzer die effektive Größe des Datenbank-Puffer-Cache von Oracle erhöhen, ohne dem System zusätzlichen Arbeitsspeicher hinzuzufügen. Im Klartext: „Database Smart Flash Cache“ kann die Vorteile von Flash-Drives (auch als SSD-Platten bekannt) nutzen, um die Geschwindigkeit zu verbessern. Weitere Informationen dazu siehe <http://www.oracle.com/technetwork/articles/systems-hardware-architecture/oracle-db-smart-flash-cache-175588.pdf>.

OL kann ganz einfach heruntergeladen werden und läuft ohne Supportvertrag. RHEL benötigt für mindestens ein Jahr einen Supportvertrag. Im Rahmen der Oracle Grid Infrastructure können Anwender die Clustering Software für Nicht-Oracle-Produkte ohne Extrakosten verwenden, wenn sie diese auf einem OL mit Support-Vertrag nutzen.

Ein „oracle-rdbms-server-11gR2-preinstall“-RPM-Paket vereinfacht die Installation einer Oracle-Datenbank auf OL. Damit erzeugt das System die notwendigen Nutzer und Gruppen, bestimmt die Systemkern-Parameter und -Begrenzungen etc. Und tatsächlich erhält der Anwender durch den öffentlichen Yum-Speicher von Oracle auch einen von Oracle bestätigten RPM für RHEL.

Ein Supportvertrag für Oracle Linux oder für Red Hat Linux von Oracle ist günstiger als das Angebot von Red Hat. Dazu gibt es unter <http://www.oracle.com/us/media/calculator/linuxtco/index.html> einen Kostenrechner. Oracle hat keine eigene Dokumentation für Oracle Linux erstellt und gibt lediglich ein paar Whitepaper zu den besonderen Features von Oracle Linux wie „btrfs“, „Smart Flash Cache“ etc. heraus, die sich auf die Red-Hat-Dokumentation beziehen.

### Eine eigene Oracle-Private-Cloud

Mit Oracle Linux als Betriebssystem und mindestens einem Basic-Supportpaket dürfen Anwender bestimmte Optionen von „Oracle Enterprise Manager (OEM) 12c“, „Oracle VM (OVM)“ und „Ops Center for Linux Management“ nutzen. Mit OEM als Portal dürfen die Benutzer dieselben Server bestellen und circa zehn Minuten später schon benutzen. Dabei handelt es sich um die Self-Service-Funktion von Enterprise Manager Cloud Control. Mit dieser Infrastruktur lässt sich eine Private Cloud bauen. Platform as a Service (PaaS) läuft über die folgenden Systeme:

- Oracle Linux als Betriebssystem
- Oracle VM als Virtualisierungssoftware
- Oracle Enterprise Manager (OEM) als Portal für den Besteller
- OEM und Ops Center for Linux Management für den Betrieb

Was für Oracle Linux spricht	... und was dagegen
Extra Funktionalitäten wie ASMLIB, OCFS2, Ksplice, btrfs, EMCC Einbindung & PaaS	Lange Reaktionszeiten des Supports
Lizenzierung per Download, geringe Supportkosten, Support muss nicht zwingend gekauft werden	Bislang ist nicht sichergestellt, dass Applikationen, die auf RHEL zertifiziert sind, keine zusätzliche Zertifizierung brauchen
Vorteile der Lizenzierung beim Cloud Computing: Cloud-Komponenten (Optionen in OEM und „Ops Center for Linux Management“) sind beim Erwerb eines Oracle-Linux-Support-Vertrags inklusive	Oracle VM und die Cloud-Komponenten (Optionen in OEM und „Ops Center for Linux Management“) lassen momentan bezüglich ihrer Stabilität, Performanz und Funktionalität für den produktiven Betrieb noch zu wünschen übrig. In Zukunft werden sich diese sicher verbessern und damit für die Anwender interessanter

Tabelle 1: Die Vor- und Nachteile von Oracle-Linux auf einen Blick

Mit einem Support-Vertrag für Oracle Linux dürfen Anwender eine Private Cloud mit den oben genannten Produkten betreiben, ohne weitere Lizenzen zahlen zu müssen.

**Fazit**

Für die Nutzung von Oracle-Linux spricht die Ankündigung des Herstellers, dass die Datenbank 11g R2 und Fusion Middleware 11.1 auf Oracle Linux 6 unterstützt werden. Auf <http://public-yum.oracle.com> stellt Oracle mit sofortiger Wirkung seine mit Red Hat kompatiblen Linux-Binärdateien, Updates und Errata zur Verfügung. Die folgenden Seiten enthalten den Wortlaut der besagten Proklamation: <http://www.oracle.com/us/corporate/press/1563775>. Auch angesichts dieser verlockenden Gründe, Oracle Linux anstatt Red Hat Linux zu nutzen, werden die meisten großen IT-Abteilungen, die RHEL bereits im Einsatz

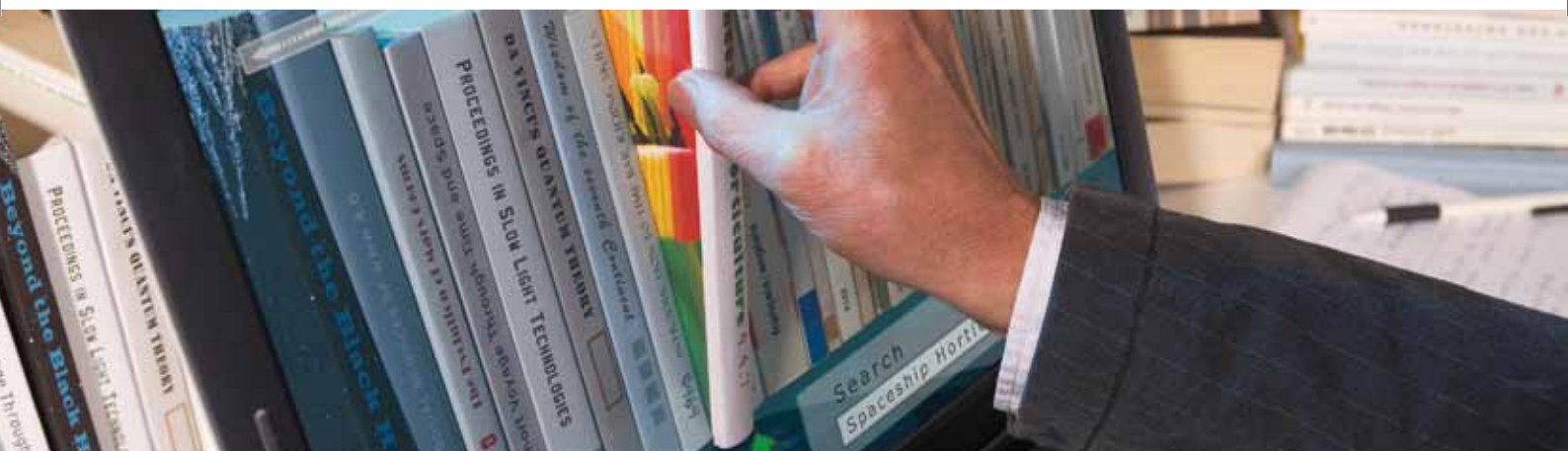
haben, der Einschätzung des Autors nach bei diesem Betriebssystem bleiben. Im Zuge der IT-Standardisierung und Konsolidierung ergibt es schließlich auch keinen Sinn, unterschiedliche Plattformen zu unterhalten. Für Abteilungen, die bislang kein Linux betreiben, ist Oracle-Linux hingegen eine sehr gute Wahl. Dabei wird auch die Tatsache, dass Updates nun kostenlos und ohne Supportvertrag erhältlich sind, insbesondere kostenbewusste IT-Verantwortliche interessieren (siehe Tabelle 1).

**Weiterführende Infos**

- <http://oraclestorageguy.typepad.com/oraclestorageguy/2009/04/what-the-oracle-vmware-support-statement-really-means-and-why.html>
- [http://blogs.computerworld.com/is\\_oracle\\_getting\\_ready\\_to\\_kill\\_opensolaris](http://blogs.computerworld.com/is_oracle_getting_ready_to_kill_opensolaris)

- <http://broadcast.oreilly.com/2009/07/is-oracle-getting-ready-to-kill.html>
- [http://blogs.computerworld.com/16997/oracle\\_rips\\_red\\_hat\\_and\\_sort\\_of\\_launches\\_a\\_new\\_linux](http://blogs.computerworld.com/16997/oracle_rips_red_hat_and_sort_of_launches_a_new_linux)
- <http://advosys.ca/view-points/2009/06/red-hat-linux-vs-oracle-linux/>
- <http://www.oracle.com/technetwork/articles/servers-storage-dev/linux6-1certification-412527.pdf>

Andrew Lacy  
[Andrew.lacy@opitz-consulting.com](mailto:Andrew.lacy@opitz-consulting.com)



# DOAG 2013 Datenbank **14. Mai 2013, Düsseldorf**

## Save the Date

Datenbank-Administratoren und technisch Interessierten eröffnet sich die Gelegenheit, einen Tag rund um die klassischen Themen der Oracle Datenbank sowie MySQL und Embedded Database zu erleben.

- Streams:
- Oracle 12c – Neueste Informationen
  - Security
  - Performance/Tuning & Monitoring
  - Administration/Migration





Datenmengen explodieren und stellen die Betreiber von Datenbanken vor große Herausforderungen. Dieser Artikel gibt einen Überblick über die verschiedenen Einsatzgebiete und Techniken zur Verdichtung von Daten in einer Oracle-Datenbank.

# Sind wir eigentlich ganz dicht?

Eero Mattila, Quest Software GmbH



Abbildung: Fotolia

Im Laufe der Jahre hat Oracle unterschiedliche Möglichkeiten zur Verdichtung von Daten eingeführt, angefangen mit der Index-Key-Komprimierung in der Version 8.1.5 über einfache Tabellen-Komprimierung in 9.2, Backup-Komprimierung in 10g, erweiterte Komprimierung in 11g Release 1 bis hin zur hybriden Spaltenkomprimierung im Zusammenhang mit Exadata in 11g Release 2. Diese Techniken zur Verdichtung stellen einerseits Chancen zur Speicherplatz-Einsparung und Performance-Vorteile dar, können aber auch Zusatzkosten – monetär wie performancebedingt – verursachen.

## Datenexplosion und ihre Folgen

Das kollektive Wissen der Menschheit nimmt täglich zu und wird immer eifriger nicht nur dokumentiert, sondern auch über verschiedenste Medien ver-

breitet. Gesetzliche Vorschriften verpflichten uns, bestimmte Daten sogar über Jahrzehnte aufzubewahren. Dies führt zu einem explosionsartigen Wachstum des Bedarfs an Speicherplatz, der wiederum schlicht Geld kostet. Hinzu kommt, dass die Verarbeitung immer größerer Datenmengen auch mehr Zeit kostet – die Performance der Anwendungen nimmt ab. Das grundsätzliche Problem – die Datenexplosion an sich – lässt sich nicht ändern, also müssen wir andere Möglichkeiten finden, damit umzugehen. Komprimierung von Daten an verschiedenen Stellen und auf unterschiedliche Arten und Weisen kann oft gute Dienste zu diesem Zweck leisten: Sie hilft einerseits, Speicherplatz zu sparen, andererseits aber auch, Zugriffszeiten auf Datenbank-Inhalte zu optimieren.

## Index Key Compression

Bereits in der Version 8.1.5 der Oracle-Datenbank wurde die Möglichkeit eingeführt, Indizes durch Komprimierung zu optimieren. Hierbei handelt es sich um ein Feature, das in jeder Edition der Datenbank zur Verfügung steht und nicht separat lizenziert werden muss. Bei der „Index Key Compression“ werden sich wiederholende Werte in führenden Spalten von Indizes dedupliziert, indem sie nur einmal im Blockheader gespeichert sind. Je nach Selektivität der zu komprimierenden Schlüssel kann der benötigte Speicherplatz des Index erheblich reduziert und dadurch deutlich geringerer I/O bei Index-Zugriffen erzielt werden. Neben „B\*Tree“-Indizes ist die Komprimierung auch für „Index Organized Tables“ möglich. Die Komprimierung kann sowohl beim Erstellen eines neuen Index als auch nachträglich definiert werden (siehe Listing 1). Mit *x* wird die Anzahl der zu komprimierenden führenden Spalten angegeben.

Ob und wie viel ein Index komprimiert werden sollte, kann mit dem Befehl „ANALYZE INDEX index VALIDATE STRUCTURE;“ ermittelt werden. Anschließend gibt die View „INDEX\_STATS“ Auskunft über die zu erzielende Ersparnis und die optimale Komprimierungseinstellung (siehe Listing 2).

Demnach empfiehlt sich in diesem Fall eine Komprimierung des Index mit „ALTER INDEX comp\_test REBUILD COMPRESS 2;“, wobei

```
CREATE INDEX ON tabelle (spalte1, spalte23, ...) COMPRESS x;
ALTER INDEX index REBUILD COMPRESS x;
```

Listing 1

```
SELECT blocks, opt_cmpr_pctsave, opt_cmpr_count
FROM index_stats;
```

BLOCKS	OPT_CMPR_COUNT	OPT_CMPR_PCTSAVE
988	2	37

Listing 2

„COMPRESS 2“ den Wert in der Spalte „OPT\_CMPR\_COUNT“ betrifft. Es ist mit einer Reduzierung der Indexblöcke um ca. 37 Prozent zu rechnen. Listing 3 dokumentiert die erneute Abfrage gegen „INDEX\_STATS“. Wie erwartet, hat sich die Anzahl der Blöcke in dem Index um ein gutes Drittel reduziert.

Bei „ANALYZE INDEX ... VALIDATE STRUCTURE“ ist zu beachten, dass der Befehl einen DML-Lock auf den Index erzeugt, sodass insbesondere bei großen Indizes die Performance von Inserts, Updates und Deletes während des Analyze-Vorgangs beeinträchtigt werden kann.

### Tabellen-Komprimierung

Oracle 9i Release 2 führte erstmals die Komprimierung für Tabellen-Daten ein, zunächst allerdings nur für ausgewählte, sogenannte „Bulk-Load-Operationen“ (Direct Path SQL\*Load, CREATE TABLE AS SELECT, paralleles DML und INSERT mit APPEND-Hint). Dies bedeutete, dass mittels normaler DML-Befehle eingefügte oder geänderte Daten nicht komprimiert wurden. Somit war dieses Feature vor allem für Data-Warehouse-Umgebungen interessant. Allerdings gab (und gibt) es dort häufig Einschränkungen – sei es, dass der ETL-Vorgang keinen Direct Load ermöglicht oder dass ein CTAS aufgrund des Datenvolumens zu viel Zeit in Anspruch nehmen würde.

Mit der Version 11g kam die (separat zu lizenzierende) „Advanced Compression“-Option. Sie erweitert die Tabellen-Komprimierung auf INSERT- und UPDATE-Kommandos, sodass die Datenverdichtung auch in OLTP-Anwendungen zum Tragen kommt. Die oben beschriebene ursprüngliche Funktionalität für Bulk-Operationen heißt nun „Basic Compression“ und ist weiterhin ohne Zusatzkosten verwendbar.

Die Komprimierung der Tabellen-Daten erfolgt auf Block-Ebene: Wiederholte Spaltenwerte werden – wie bei der Index-Key-Komprimierung – einmalig in einer Symbol-Tabelle im Blockheader abgelegt und im Block selbst lediglich referenziert. Je nach Charakter der Daten können dadurch erheblich mehr Datensätze in einem Block untergebracht werden, was wiederum die An-

```
SELECT blocks, opt_cmpr_pctsave, opt_cmpr_count
FROM index_stats;
```

BLOCKS	OPT_CMPR_COUNT	OPT_CMPR_PCTSAVE
621	2	0

Listing 3

```
CREATE TABLE [...] COMPRESS FOR OLTP;
```

Listing 4

```
CREATE TABLE doag_normal
AS SELECT * FROM dba_objects;

CREATE TABLE doag_compr COMPRESS FOR OLTP
AS SELECT * FROM dba_objects;

SELECT segment_name, blocks
FROM user_segments
WHERE segment_name LIKE 'DOAG%';
```

Listing 5

SEGMENT_NAME	BLOCKS
DOAG_COMPR	384
DOAG_NORMAL	1152

Listing 6

```
ALTER TABLE tabelle COMPRESS FOR OLTP;
```

Listing 7

```
ALTER TABLE tabelle MOVE COMPRESS FOR OLTP;
```

Listing 8

```
CREATE TABLESPACE [...] DEFAULT COMPRESS FOR OLTP [...];
ALTER TABLESPACE DEFAULT COMPRESS FOR OLTP;
```

Listing 9

zahl der physikalischen und logischen I/O-Operationen reduziert.

Beim Erzeugen einer neuen Tabelle wird die Komprimierung durch die Klausel „COMPRESS FOR“ definiert, gefolgt von den Operationen, für die die Komprimierung gelten soll. Gülti-

ge Optionen sind „BASIC“ (nur Bulk Inserts und CTAS) sowie „OLTP“ (alle Operationen, siehe Listing 4). Bei Exadata gibt es noch weitere Optionen – dazu später mehr.

Bereits bei Tabellen recht übersichtlicher Größe lässt sich die Platzerspar-

nis feststellen. Im folgenden Beispiel werden zwei identische Tabellen erstellt, einmal komprimiert, einmal nicht. Anschließend wird die Anzahl der Blöcke anhand der View `USER_SEGMENTS` kontrolliert (siehe Listing 5). Das Ergebnis ist eindeutig (siehe Listing 6). Listing 7 zeigt, dass sich die Verdichtung auch nachträglich einstellen lässt.

Dabei ist jedoch zu beachten, dass bereits vorhandene, vollständig ausgefüllte Blöcke nicht automatisch komprimiert werden. Um diese Blöcke zu verdichten, müsste man zunächst Datensätze löschen. Wird die Tabelle jedoch gleichzeitig reorganisiert, greift die Komprimierung sofort (siehe Listing 8). Die Komprimierung kann außerdem als Standard-Einstellung für alle neuen Tabellen in einem Tablespace definiert werden, sowohl bei dessen Erstellung als auch nachträglich (siehe Listing 9).

### Welche Tabellen man komprimieren sollte

Typische Kandidaten für eine Komprimierung sind natürlich große Tabellen, die sich wiederholende Daten enthalten. Um festzustellen, ob und wie stark sich eine Tabelle verdichten lässt, steht ab Oracle 11g Release 2 das mitgelieferte Package `„DBMS_COMPRESSION“` zur Verfügung. Für ältere Versionen hält das Oracle Technology Network das Package `„DBMS_COMP_ADVISOR“` zum Herunterladen bereit. Die Prozedur `„GET_COMPRESSION_RATIO“` beziehungsweise `„GETRATIO“` in der älteren Fassung berechnet für die angegebene Tabelle den zu erwartenden Komprimierungsgrad für die gewünschte Komprimierungsart. Derzeit beschränkt sich die Unterstützung auf BASIC, OLTP sowie die verschiedenen Varianten der Exadata Hybrid Columnar Compression (siehe weiter unten). Hinweise für Indizes oder unstrukturierte Daten gibt es leider nicht.

### Was Index- oder Tabellen-Komprimierung kostet

Der reine Lizenzkosten-Aspekt der Advanced-Compression-Option soll hier nicht weiter vertieft werden. Man muss sich dessen bewusst sein,

wie auch der Tatsache, dass die Funktionalität in jeder Installation einer Enterprise Edition vorhanden und ohne besondere Freischaltung einsetzbar ist. Sobald das Zauberwort `„COMPRESS“` in einem `„CREATE“`- oder `„ALTER“`-Statement erscheint, ist eine Lizenzierung fällig. Wer die Option nicht besitzt und die Verwendung seinen Anwendern verwehren möchte, kann auf den nicht dokumentierten `„init.ora“-Parameter „_OLTP_COMPRESSION=FALSE“` zurückgreifen – aber auf eigene Gefahr.

Komprimierung bedeutet selbstverständlich Umrechnung von Daten. Bedeutet das, dass wir uns Platzersparnis und I/O-Vorteile auf Kosten von CPU-Verbrauch erkaufen? Kann jedes `„INSERT“` oder `„UPDATE“` unmittelbar ein Umschreiben von Block-Headern und -Inhalten verursachen? Müssen bestehende Anwendungen angepasst oder darauf vorbereitet werden?

Für lesende Zugriffe bedeutet die Komprimierung – oder vielmehr Dekomprimierung – meist keinen messbaren Overhead. Dadurch, dass die Symbol-Tabelle auf Block-Ebene geführt wird, geschieht die Umrechnung extrem schnell. Die Lese-Performance lässt sich in vielen Fällen sogar erheblich steigern, weil weniger Blöcke gelesen werden müssen, sodass die I/O-Vorteile den ohnehin geringen CPU-Overhead deutlich überwiegen.

Interessanter ist die Frage nach den DML-Vorgängen in schreibintensiven OLTP-Systemen. Glücklicherweise lösen einzelne INSERTs oder DELETEs keine Komprimierungs-Vorgänge aus, sondern die Komprimierung findet asynchron statt, sobald ein bestimmter Füllgrad des Blocks erreicht wird. Umfangreiche UPDATES können jedoch zu `„Migrated Rows“` und damit zu Performance-Einbußen führen. Gründliche Tests sind daher in jedem Fall erforderlich, ehe bestehende Daten in Produktions-Umgebungen umgestellt werden können. Der Einsatz von Index- und Tabellenkomprimierung ist aus der Sicht der Datenbankanwendungen vollkommen transparent. Anpassungen bestehender oder die Vorbereitung neuer Applikationen ist nicht erforderlich.

### Komprimierung unstrukturierter Daten

Besonders unstrukturierte Daten nehmen häufig sehr viel Platz in Anspruch und sind somit prädestinierte Kandidaten für eine Komprimierung, sofern sie nicht von vornherein in einem komprimierten Format vorliegen. In der Datenbank-Version 10g wurde das Package `„UTL_COMPRESS“` vorgestellt, mit dem große binäre Objekte (RAW, BLOB und BFILE) komprimiert werden können.

`„SecureFiles“` in 11g werden gern als die „nächste Generation von LOBs“ bezeichnet. Neben verschiedenen Vorteilen bezüglich Performance und Administration bieten sie – auch wieder unter Nutzung der Advanced-Compression-Option – erweiterte Möglichkeiten zur Datenverdichtung. Als `„SecureFiles“` gespeicherte große Objekte können einerseits komprimiert werden, wobei die drei Komprimierungsstufen `„LOW“`, `„MEDIUM“` und `„HIGH“` zur Verfügung stehen. Zu beachten ist, dass die Komprimierung der LOBs unabhängig von der regulären Tabellen-Komprimierung ist: Die Tabellen-Komprimierung komprimiert keine LOBs und umgekehrt. Neben der Komprimierung lassen SecureFiles auch die Deduplizierung der Objekte zu: Identische Dateien werden einmal gespeichert und von weiteren Datensätzen referenziert. Diese beiden Methoden können zu drastischen Einsparungen an Speicherplatz führen.

### Hybrid Columnar Compression

Die oben beschriebenen Tabellen-Komprimierungsarten sind unabhängig vom verwendeten Speichersystem einsetzbar. Die hybride Spalten-Komprimierung oder Hybrid Columnar Compression (HCC) setzt dagegen einen Exadata-Storage-Server beziehungsweise Oracle SAN oder NAS-Storage-Server voraus. HCC unterscheidet zwischen Data-Warehouse- und Archiv-Systemen und stellt für beide optimierte Komprimierungs-Algorithmen zur Verfügung.

Bei HCC handelt es sich um eine Kombination aus spalten- und zeilenbasierter Datenspeicherung. Die Datensätze werden in sogenannten `„Logischen Komprimierungsgruppen“` or-

**Unsere Inserenten**

Hunkler GmbH & Co. KG www.hunkler.de	S. 3
KeepTool GmbH www.keeptool.com	S. 9
Herrmann & Lenz Services GmbH www.hl-services.de	S. 17
Libelle AG www.libelle.com	S. 27
MuniQsoft GmbH www.muniqsoft.de	S. 29
OPITZ CONSULTING GmbH www.opitz-consulting.com	U 2
ORACLE Deutschland B.V. & Co. KG www.oracle.com	U 3
ProLicense GmbH www.prolicense.com	S. 11
Trivadis GmbH www.trivadis.com	U 4

ganisiert, die mehrere Blöcke umfassen und die Daten spaltenweise in verschiedenen Blöcken abspeichern. Dadurch werden sich wiederholende Daten in den gleichen Blöcken konzentriert, sodass eine extrem effiziente Komprimierung möglich ist.

Warehouse-Anwendungen sind dadurch geprägt, dass extrem große Daten-Volumina mittels „SELECT“-Statements abgefragt und meist nie oder sehr selten verändert werden. Die Warehouse Compression verwendet einen speziell für Abfragen optimierten Algorithmus, wobei zwischen zwei Komprimierungsstufen gewählt werden kann. Typische Komprimierungsraten sind hier sechs bis zehn im Vergleich zu nicht komprimierten Daten.

Archivsysteme dienen zur Speicherung von historischen Daten, die aufgrund von gesetzlichen Vorschriften sogar jahrzehntelang vorgehalten werden müssen. Häufig werden solche Daten auf Band-Archiven abgelegt. Bei

Bedarf müssen sie jedoch für den Zugriff wiederhergestellt werden, was sehr aufwändig sein kann. Die Archive Compression von HCC ermöglicht es, historische Daten auf Platzersparnis zu optimieren und trotzdem jederzeit im Zugriff zu behalten. Der Komprimierungsfaktor liegt bei Archive Compression häufig bei mindestens fünfzehn – aus einem Terabyte Daten werden dann rund 65 GB.

**Backup-Komprimierung**

Das Wachstum der Datenmengen führt zwangsläufig auch zu größeren und damit langsameren Daten-Sicherungen. Nicht überraschend, kann auch hier die Komprimierung wertvolle Vorteile verschaffen. Oracle Recovery Manager sichert Daten blockweise physikalisch und ermöglicht die Wiederherstellung einer Datenbank, eines Tablespace oder auch einzelner Datenblöcke. Für Exports oder logische Backups einzelner Schemata oder Objekte bietet sich Data Pump an. Beide Werkzeuge sind in der Lage, Daten unter Nutzung der Advanced-Compression-Option zu komprimieren, RMAN auch ohne das kostenpflichtige Add-on. Bei RMAN wird die Komprimierung mit einem Befehl eingeschaltet (siehe Listing 10). Anschließend kann der Komprimierungsalgorithmus ausgewählt werden (siehe Listing 11). Die Komprimierungsstufe „BASIC“ ist ohne die Advanced-Compression-Option verwendbar.

```
CONFIGURE DEVICE TYPE
[DISK|SBT] BACKUP TYPE TO COMPRESSED
BACKUPSET;
```

Listing 10

```
CONFIGURE COMPRESSION ALGORITHM
'[BASIC|LOW|MEDIUM|HIGH]';
```

Listing 11

Da die Performance der verschiedenen Komprimierungsstufen stark von zahlreichen Faktoren (Charakter der Daten, Netzwerk-Konfiguration, CPU-Ressourcen etc.) abhängt, können keine allgemein gültigen Empfehlungen abgegeben werden. Höhere Komprimierung

ist meist mit höherem CPU-Verbrauch verbunden, kann jedoch durch geringere Netzwerk-Belastung von Vorteil sein. Ausführliche Tests in der jeweiligen Umgebung sind letztendlich ausschlaggebend für die optimale Auswahl der Parameter. Besonders wichtig dabei ist, das Restore gründlich zu testen: Die kleinste Sicherung ist nichts wert, wenn die Wiederherstellung ewig dauert.

Data Pump Exports kennen ebenfalls den Parameter „COMPRESSION“, der folgende Werte akzeptiert: „ALL“, „METADATA\_ONLY“ und „DATA\_ONLY“. Sowohl RMAN als auch Data Pump dekomprimieren die Daten transparent beim Wiederherstellen beziehungsweise Importieren.

**Fazit**

Die Oracle-Datenbank bietet zahlreiche Mechanismen zur Komprimierung von Daten und dadurch zur Optimierung von Speicherplatz und Anwendungs-Performance. Während einzelne dieser Methoden ohne Extrakosten eingesetzt werden können, sind bei vielen verlockenden Funktionen zusätzliche Lizenzen fällig, sodass deren Einsatz wohlüberlegt werden muss. Performance ist ein wichtiger Aspekt bei allen Anwendungen, und die Komprimierung kann sich in gewissen Konstellationen auch negativ darauf auswirken. Daher ist ein umfangreiches Testen unbedingt notwendig, bevor Komprimierung produktiv eingesetzt wird. Wer aber mit stetig wachsenden Datenmengen, Performance-Verlust und limitierter Speicherkapazität zu kämpfen hat, tut gut daran, sich die verschiedenen Komprimierungsmöglichkeiten genauer anzuschauen.

Eero Mattila  
eero.mattila@quest.com



Jeder, der Datenbanken betreut und administriert, macht sich Gedanken über Sicherung und Wiederherstellung seiner Daten. Diese Methoden decken aber oft nur physikalische Fehler ab. Probleme, die von Nutzern oder Programmfehlern durch versehentliches Ändern oder Löschen verursacht werden, sind damit nur schwer und mit großem Aufwand zu beheben. Oracle bietet mit verschiedenen Methoden des Flashbacks einige sehr nützliche Funktionen, um derartige Fehler recht effizient zu behandeln. Einige dieser Varianten sind in diesem Artikel näher beleuchtet.

# Flashback: Recovery ohne Recovery

Marco Mischke, Robotron Datenbank-Software GmbH

Jeder DBA hat das sicher schon erlebt: Alle Datenbanken im Unternehmen laufen seit Monaten zuverlässig. Die ausgefeilte Backup-Strategie hat sich etabliert und in verschiedenen Tests bewiesen, dass Fehler in Kontroll- oder Tablespace-Dateien zügig zu beheben sind. Dann klingelt das Telefon und am anderen Ende erzählt ein Entwickler aufgeregt von plötzlich verschwundenen Daten nach einem Durchlauf einer eigentlich gut getesteten Prozedur zur Reorganisation der Stammdaten. Passiert ist das vor etwa einer Stunde, der Entwickler wollte zuerst selbst das Problem lösen. Natürlich betrafen die Aktionen das Produktiv-System, das allen anderen Anwendungen im Unternehmen als zentrale Datenbasis dient. Daher kann das System nicht einfach mal heruntergefahren werden und auch Bewegungsdaten wie Bestellungen etc. dürfen nicht verloren gehen.

Man könnte die Datenbank einfach aus dem Backup wiederherstellen und per unvollständigem Recovery zum Zeitpunkt kurz vor dem Fehler wieder online bringen – dann wären jedoch alle neueren Änderungen verloren. Auf einem separaten System ein Tablespace Point in Time Recovery durchzuführen, würde zu viel Zeit benötigen.

Jetzt kommt Flashback ins Spiel. Die Oracle-internen Methoden zum konsistenten Lesen sind seit Version 9 auch für Abfragen in die Vergangenheit benutzbar. Dabei wurden mit jeder Version weitere Features eingeführt, die den Administratoren das Beheben solcher inhaltlichen Fehler einfacher machen (können). Dabei dient „Flashback“ als Oberbegriff und vereint verschiedene

Technologien, um an Daten und Zustände aus der Vergangenheit zu gelangen, ohne ein Restore und Recovery durchführen zu müssen. Einige davon stehen bereits in der Standard Edition zur Verfügung.

## Flashback Query

Oracle loggt alle Änderungen von Transaktionen im Undo Tablespace, um Rollbacks sowie konsistentes Lesen anderer Sessions zu ermöglichen. Per Default hebt Oracle die Änderungen der letzten fünfzehn Minuten auf; dieser Zeitraum lässt sich über den Parameter „undo\_retention“ beeinflussen. Da die Dateien des Undo Tablespace sich normalerweise automatisch erweitern, werden auch nur genau diese fünfzehn Minuten aufgehoben. Bei Dateien mit fester Größe versucht die Datenbank, den gesamten zur Verfügung stehenden Speicherplatz zu benutzen, und ignoriert die Einstellungen des Parameters „undo\_retention“. Die View „V\$UNDOSTAT“ zeigt in der Spalte „TUNED\_UNDORETENTION“ den aktuellen Wert (siehe Listing 1).

Um Daten konsistent lesen zu können, merkt sich die Datenbank die System Change Number (SCN) beim Start der Abfrage beziehungsweise der Transaktion. Werden Datensätze danach von anderen Transaktionen verändert, wird der vorige Zustand des entsprechenden Blocks im Undo gespeichert. Liest die laufende Abfrage nun einen Block mit einer höheren SCN als der zum Start gespeicherten, dann wird der Block so lange mit den Informationen aus dem Undo zurückgerollt, bis eine SCN vor dem Start der Abfrage erreicht ist.

Bei Flashback Query kommt genau dieser Mechanismus zum Einsatz. Nur wird eben nicht die SCN zum Start der Abfrage gespeichert, sondern diejenige, die dem gewünschten Zeitpunkt am nächsten kommt. Sind historische Daten nicht mehr aus dem Undo zu rekonstruieren, dann endet die Abfrage in einem „ORA-01555: Snapshot too old“.

Die Datenbank kann sich also quasi an alte Daten erinnern. Wie lange die Erinnerung anhält, ist in der Einstellung von „undo\_retention“ definiert. Dieses Erinnerungsfenster wandert mit

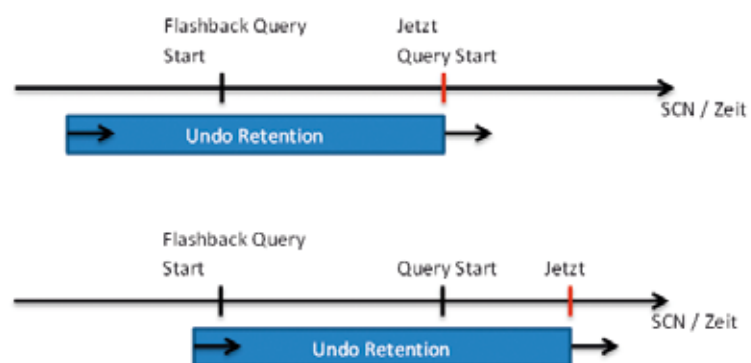


Abbildung 1: Zeitliche Zusammenhänge von Flashback Query und Undo

```
SQL> select * from (
  2 select TUNED_UNDORETENTION
  3 from V$UNDOSTAT
  4 order by END_TIME desc
  5 )
  6 where ROWNUM = 1;
```

```
TUNED_UNDORETENTION
-----
                 36356
```

**Listing 1**

```
$ expdp system/Oracle-1 dumpfile=scott_flashback.dmpdp \
  logfile=scott_flashback.expdp.log \
  directory=data_pump_dir \
  flashback_time='2012-10-01-11:00:00' schemas=scott
```

**Listing 2**

```
$ exp system/Oracle-1 file=/tmp/scott_flashback.dmp \
  log=/tmp/scott_flashback.exp.log \
  flashback_time='2012-10-01-11:00:00' owner=scott
```

**Listing 3**

```
SQL> create table emp_old
  2 as
  3 select *
  4 from emp as of timestamp
  5 to_timestamp(,2012-10-01 11:00:00',
  6 ,yyyy-mm-yy hh24:mi:ss');
```

**Listing 4**

```
SQL> select * from emp_old
  2 minus
  3 select * from emp;
```

**Listing 5**

```
SQL> select * from emp as of timestamp
  2 to_timestamp(,2012-10-01 11:00:00',
  3 ,yyyy-mm-dd hh24:mi:ss')
  4 minus
  5 select * from emp;
```

**Listing 6**

```
SQL> insert into emp
  2 select * from emp as of timestamp
  3 to_timestamp('2012-10-01 11:00:00',
  4 'yyyy-mm-dd hh24:mi:ss')
  5 minus
  6 select * from emp;

SQL> commit;
```

**Listing 7**

der Zeit mit, am Anfang kommen neue Erinnerungen (Änderungen) dazu und überschreiben dabei am Ende die ältesten. Abbildung 1 verdeutlicht die zeitlichen Zusammenhänge.

Wer den Undo Tablespace seiner Datenbanken ausreichend Platz gibt, um etwa zehn bis zwölf Stunden an Transaktionen unterzubringen, kann sich an die Wiederherstellung der verlorenen Daten für einen kompletten Werktag machen.

**Flashback Export mit der Standard Edition**

Damit die Daten aus dem Undo Tablespace nicht überschrieben werden und dann verloren wären, erzeugt man zuallererst einen Export des betroffenen Schemas mit dem Stand vor den fraglichen Änderungen. Dabei ist zu beachten, dass das Zeitformat immer in der Form „YYYY-MM-DD-HH24:MI:SS“ angegeben wird (siehe Listing 2). Selbstverständlich kann man auch das herkömmliche Export Utility dafür benutzen (siehe Listing 3). Der Parameter „flashback\_time“ kann dabei mit allen anderen Parametern kombiniert werden, um etwa statt des gesamten Schemas nur einige Tabellen zu exportieren.

Nachdem man sich mit dem Export etwas Sicherheit verschafft hat, beginnt der Versuch, die Daten zu retten. Man kann dazu beispielsweise eine Kopie der Tabelle anlegen, die den Stand aus der Vergangenheit hat. Dazu dient die Klausel „as of timestamp“ im „FROM“ (siehe Listing 4).

Nun können die Inhalte beider Tabellen verglichen werden, um die gelöschten oder geänderten Datensätze zu ermitteln (siehe Listing 5). Da die „as of timestamp“-Klausel in allen Arten von SQL verwendet werden darf, lassen sich die Daten aus der Vergangenheit auch direkt mit den aktuellen Datenständen vergleichen (siehe Listing 6).

Im einfachsten Fall können nun versehentlich gelöschte Datensätze wieder in die Tabelle eingefügt werden (siehe Listing 7). Nach dem gleichen Prinzip lassen sich auch Datensätze retten, wenn nur Änderungen vorgenommen und nichts gelöscht wurde, hier am Beispiel des Gehalts der Mitarbeiter (siehe Listing 8).

Diese Beispiele zeigen eine Richtung auf, mit der versehentlich geänderte Daten wiederhergestellt werden können. Der persönlichen Kreativität bei der Abfrage-Formulierung sind hier kaum Grenzen gesetzt.

### Flashback Drop mit der Standard Edition

Flashback Query deckt nur DML-Operationen ab, also inhaltliche Änderungen. Hat der Entwickler allerdings versehentlich eine oder mehrere Tabellen gelöscht, wird das natürlich nicht im Undo geloggt, da es sich um eine DDL-Operation handelt. Oracle bietet hier die Recycle Bin-Funktionalität (Papierkorb) an, die per Standard über den Parameter „recyclebin“ aktiviert ist. Daher wird ein Segment beim DROP-Befehl nur als frei markiert, aber noch nicht gelöscht. Oracle überschreibt solche Segmente erst, wenn kein anderer freier Speicherplatz im Tablespace mehr verfügbar ist. So lange das nicht passiert, lässt sich das Segment wiederherstellen; diese Funktion wird als „Flashback Drop“ bezeichnet. Um zu sehen, welche Segmente wiederherstellbar sind, dienen der SQL\*Plus-Befehl „show recyclebin“ oder die Views „USER\_RECYCLEBIN“, „ALL\_RECYCLEBIN“ und „DBA\_RECYCLEBIN“ (siehe Listing 9).

Es ist zu sehen, dass der SQL\*Plus-Befehl lediglich die gelöschten Tabellen auflistet, die View hingegen alle gelöschten Segmente enthält. Die von dem Entwickler versehentlich gelöschte Tabelle kann also offenbar wiederhergestellt werden (siehe Listing 10).

Die Datenbank hat also nicht nur die Tabelle sondern auch die zugehörigen Indizes wiederhergestellt. Allerdings hat nur die Tabelle ihren originalen Namen zurückerhalten. Die Indizes haben nach wie vor die Bezeichnung aus dem Recycle Bin (siehe Listing 11). Diese müssen also im Nachgang noch angepasst werden. Es ist daher wichtig, die Zuordnung der Recycle-Bin-Namen zu den originalen Indexnamen vor dem Flashback zu notieren oder direkt ein Skript zum Umbenennen zu erzeugen. Listing 12 zeigt, wie das Umbenennen erfolgt.

Es ist wichtig, den Recycle-Bin-Namen in Anführungszeichen zu setzen, da dieser Sonderzeichen enthält. Einzig Fremdschlüssel-Constraints kön-

```
SQL> update emp e_live
 2 set sal = (select sal
 3           from emp as of timestamp
 4             to_timestamp('2012-10-01 11:00:00',
 5                         'yyyy-mm-dd hh24:mi:ss') e_orig
 6           where e_orig.empno = e_live.empno
 7           )
 8 ;

SQL> commit;
```

Listing 8

```
SQL> show recyclebin
ORIGINAL_NAME RECYCLEBIN NAME          OBJ. TYPE  DROP TIME
-----
EMP           BIN$yxLVQQOqC6rgQBAktBUK1A==$0 TABLE      2012-10-02:13:29:19

SQL> select ORIGINAL_NAME, OBJECT_NAME, OPERATION, DROPTIME
 2 from USER_RECYCLEBIN;

ORIGINAL_NAME OBJECT_NAME          OPERATION  DROPTIME
-----
EMP           BIN$yxLVQQOqC6rgQBAktBUK1A==$0 DROP        2012-10-02:13:29:19
PK_EMP       BIN$yxLVQQOpC6rgQBAktBUK1A==$0 DROP        2012-10-02:13:29:19
IX_ENAME     BIN$yxLVQQOoC6rgQBAktBUK1A==$0 DROP        2012-10-02:13:29:19
IX_MGR       BIN$yxLVQQOnC6rgQBAktBUK1A==$0 DROP        2012-10-02:13:29:19
```

Listing 9

```
SQL> flashback table emp to before drop;

Flashback abgeschlossen.

SQL> select ORIGINAL_NAME, OBJECT_NAME, OPERATION, DROPTIME
 2 from USER_RECYCLEBIN;

Es wurden keine Zeilen ausgewählt
```

Listing 10

```
SQL> select index_name from user_indexes
 2 where table_name='EMP';

INDEX_NAME
-----
BIN$yxLVQQOoC6rgQBAktBUK1A==$0
BIN$yxLVQQOnC6rgQBAktBUK1A==$0
BIN$yxLVQQOpC6rgQBAktBUK1A==$0
```

Listing 11

```
SQL> alter index "BIN$yxLVQQOoC6rgQBAktBUK1A==$0" rename to IX_ENAME;
```

Listing 12

nen nicht wiederhergestellt werden. Diese sind manuell zu prüfen und bei Bedarf neu anzulegen.

### Flashback Version Query mit der Enterprise Edition

Es muss aber nicht unbedingt der Entwickler sein, der ungewollte Änderungen ausgelöst hat. Wenn beispielsweise die Personalabteilung gerade die beschlossene Gehaltsanpassung bearbeitet hat und bei der Prüfung feststellt, dass einige Mitarbeiter ein falsches Gehalt zugeordnet bekommen haben, will der Personalchef wissen, wann und welche Änderungen am Gehalt eines bestimmten Mitarbeiters vorgenommen wurden. Hier kommt „Flashback Version Query“ ins Spiel, mit der sich der zeitliche Verlauf eines Datensatzes ermitteln lässt. Für den Angestellten „MILLER“ wird so die Historie der letzten Stunde ermittelt (siehe Listing 13).

Hieraus sieht man, dass vor der geplanten Erhöhung um 500 Euro um 14:35 Uhr jemand irrtümlich das Gehalt verringert hat, und zwar um 14:29 Uhr. Diese Art von Abfragen ermittelt genauso Änderungen, die durch Einfügen und Löschen entstehen. Listing 14 zeigt die kurze, aber ereignisreiche Karriere von Herrn Foo mit Neueinstellung, der kurzfristigen Gehaltserhöhung und der anschließenden betriebsbedingten Kündigung mit Flashback Version Query.

### Weitere Möglichkeiten

Es gibt noch eine Reihe weiterer Flashback-Features, die hier kurz aufgezählt und erläutert sind. Alle sind Teil der Enterprise Edition.

„Flashback Transaction Query“ ermöglicht es, Informationen zu einer bestimmten Transaktion zu erhalten. Die View „FLASHBACK\_TRANSACTION\_QUERY“ listet alle Änderungen auf, die durch eine bestimmte Transaktion durchgeführt wurden.

„Flashback Table“ setzt den Inhalt einer ganzen Tabelle auf einen Stand in der Vergangenheit zurück. Dabei können mehrere Tabellen in einem Schritt auf eine frühere Stufe gebracht werden, was insbesondere bei existierenden Fremdschlüssel-Beziehungen hilfreich ist.

```
SQL> select
  2   versions_starttime, versions_endtime,
  3   versions_operation, versions_xid,
  4   sal
  5 from
  6   emp versions between timestamp
  7     systimestamp - interval '1' hour and
  8     systimestamp
  9 where
 10   ename = 'MILLER';
```

VERSIONS_STARTTIME	VERSIONS_ENDTIME	V	VERSIONS_XID	SAL
02.10.12 14:35:41		U	07000B00F7480000	1800
02.10.12 14:29:29	02.10.12 14:35:41	U	1300	
	02.10.12 14:29:29		1504,91	

Listing 13

```
SQL> select
  2   versions_starttime, versions_endtime,
  3   versions_operation,
  4   sal
  5 from
  6   emp versions between timestamp
  7     systimestamp - interval '1' hour and
  8     systimestamp
  9 where
 10   ename = 'FOO';
```

VERSIONS_STARTTIME	VERSIONS_ENDTIME	V	SAL
02.10.12 15:30:05		D	5046
02.10.12 14:57:08	02.10.12 15:30:05	U	5046
02.10.12 14:45:05	02.10.12 14:57:08	I	4711

Listing 14

„Flashback Database“ setzt die komplette Datenbank auf einen früheren Stand zurück. Dafür müssen gesonderte Flashback Logs geschrieben werden. Die Vorhaltezeit dieser Logs wird über den Parameter „db\_flashback\_retention\_target“ eingestellt.

„Flashback Transaction“ dient zum Ausradieren einzelner Änderungen an Datensätzen. Hier werden Änderungen und deren Abhängigkeiten in einer Art „Baum“ aufgelistet. Es kann dann beim Rückgängigmachen der Änderung entschieden werden, wie man diese Abhängigkeiten behandelt.

### Fazit

Schon die Standard Edition der Datenbank bietet den Administratoren hilfreiche Werkzeuge, um versehentliche

oder ungewollte inhaltliche Änderungen an Tabellen rückgängig machen zu können, ohne dabei die Verfügbarkeit der Datenbank zu beeinflussen. Die hier gezeigten Möglichkeiten sind nur die Spitze des Eisbergs, sie geben aber einen ersten Eindruck davon, wie mächtig die Flashback-Funktionen sind.

Marco Mischke  
marco.mischke@robotron.de





Der Umgang mit Datumswerten ist alltäglich bei der Arbeit mit der Oracle-Datenbank. Dennoch treten gerade in diesem Bereich immer wieder Fragen auf, beispielsweise bei der Fehlermeldung, wenn man zum 31. Januar einen Monat hinzurechnet, oder wann man DATE und wann TIMESTAMP einsetzt. Der Artikel stellt die Unterschiede zwischen DATE und TIMESTAMP vor und geht im Detail auf den Umgang mit Datums-Intervallen ein.

# DATE und TIMESTAMP in der Praxis

Carsten Czarski, ORACLE Deutschland B.V. & Co KG

TIMESTAMP ist der jüngere und mächtigere Datentyp. Andererseits kommt es immer noch vor, dass Werkzeuge nicht richtig mit TIMESTAMP umgehen. Aus diesem Grund wird das ältere DATE selbst auf 10g- oder 11g-Datenbanken immer noch recht häufig genutzt. Das ist kein Problem, denn solange man keine speziellen Features

des TIMESTAMP benötigt, sind beide Typen gleichwertig:

- Die kleinste Einheit eines DATE ist die Sekunde, TIMESTAMP kann dagegen mit Millisekunden umgehen.
- TIMESTAMP unterstützt Zeitzonen, allerdings muss man dann den speziellen Datentyp TIMESTAMP WITH

TIME\_ZONE verwenden. SQL-Funktionen wie FROM\_TZ sind recht hilfreich beim Konvertieren zwischen Zeitzonen.

Weder DATE noch TIMESTAMP speichern das Datum intern als Zahl, wie beispielsweise Unix es macht. Das kann man mit der DUMP-Funktion auch sehr einfach selbst nachvollziehen. In Listing 1 ist allerdings gut erkennbar, dass TIMESTAMP intern mit UTC arbeitet. „217,7“ wird hexadezimal als „D9 07“ ausgedrückt. Als 16-bit-Zahl wäre das „07D9“, was wiederum in dezimaler Schreibweise „2009“ ist. Man sieht also, dass die Information zu Jahr, Monat, Tag, Stunde, Minute und Sekunde explizit in den Bytes enthalten sind.

## Datums-Intervalle ermitteln

DATE und TIMESTAMP erlauben Datums-Arithmetik: Ein Datums-Intervall zwischen zwei DATE-Werten wird als numerischer Wert, der zwischen zwei TIMESTAMP-Werten liegt, als Intervall-Datentyp ermittelt. Das numerische Intervall kann in einen Intervall-Datentyp konvertiert werden. Meist wird (schon aus historischen Gründen) mit der numerischen Variante gearbeitet: Dabei ist die Einheit, mit der man arbeitet, 1 Tag. Eine Stunde ist demnach 1/24 davon, eine Minute ist 1/1440 und so weiter. Möchte man also wissen, welche Zeitspanne zwischen zwei DATE-Werten liegt, so zieht man sie voneinander ab.

Die Ausgabe der Abfrage in Listing 2 ist, wie gesagt, in Tagen zu interpretieren. Es kommen also 62 und ein knapper Drittel-Tag heraus. Verwendet man dagegen TIMESTAMP, so bekommt man als Ergebnis einen Intervall-Datentyp (siehe Listing 3).

```
select sysdate, dump(sysdate) bytes from dual;

SYSDATE          BYTES
-----
21.09.2009 12:13:50 Typ=13 Len=8: 217,7,9,21,12,13,50,0
select systimestamp, dump(systimestamp) bytes from dual;
SYSTIMESTAMP
-----
BYTES
-----
21.09.2009 12:15:21,413578 +02:00
Typ=188 Len=20: 217,7,9,21,10,15,21,0,16,179,166,24,2,0,5,0,0,0,0,0
```

Listing 1

```
select
  to_date('2009-09-21 14:45', 'YYYY-MM-DD HH24:MI') -
  to_date('2009-07-21 07:30', 'YYYY-MM-DD HH24:MI') INTERVAL
from dual;

INTERVAL
-----
62,302083333
```

Listing 2

```
select
  to_timestamp('2009-09-21 14:45', 'YYYY-MM-DD HH24:MI') -
  to_timestamp('2009-07-21 07:30', 'YYYY-MM-DD HH24:MI') INTVAL
from dual;

INTVAL
-----
+0000000062 07:15:00.000000000
```

Listing 3

Der bei der Arbeit mit DATE zurückgegebene numerische Wert kann, wie schon erwähnt, in einen solchen Intervall-Datentyp konvertiert werden (siehe Listing 4). Es zeigt sich allerdings ein Rundungsproblem, deshalb wäre es besser, sich das Ergebnis der Datums-Arithmetik auch bei Verwendung von DATE direkt als Intervallwert zurückgeben zu lassen und den Umweg über NUMBER zu vermeiden (siehe Listing 5).

Zieht man zwei Datumswerte voneinander ab, so kann man also sowohl bei der Arbeit mit DATE als auch mit TIMESTAMP zu einem Intervall-Datentyp kommen.

Neben der schöneren Anzeige in SQL\*Plus bietet dieser noch einige handfeste Vorteile: Die einzelnen Elemente eines Datums-Intervalls wie Tage, Minuten oder Stunden können mit der EXTRACT-Funktion ausgelesen werden (siehe Listing 6). Die Extraktion der Minuten funktioniert analog (siehe Listing 7).

**Intervalle hinzufügen oder abziehen**

Geht es bei der Datumsarithmetik um das Addieren oder Subtrahieren eines Intervalls von einem vorhandenen Zeitstempel, so bieten sich wiederum sowohl numerische als auch Intervall-Datentypen an. Listing 8 zeigt, wie man zwölf Stunden zu einem Datum addiert.

Beide Varianten funktionieren sowohl mit DATE als auch mit TIMESTAMP. Bei Verwendung der numerischen Variante mit TIMESTAMP sollte man aber aufpassen, denn die Datenbank konvertiert dann implizit in ein DATE. Listing 9 zeigt das anhand der Ausgabe in SQL\*Plus: Der SYSTIME-STAMP wird mitsamt Millisekunden und Zeitzone ausgegeben; wird die Zahl „1“ (für einen Tag) addiert, fehlen diese – es ist nun ein DATE.

Wie eingangs gezeigt, können sich durch das Hin- und Herkonvertieren auch Rundungsdifferenzen ergeben. Prinzipiell sind die Intervall-Datentypen also vorzuziehen – mit nachfolgender Besonderheit.

**Monate abziehen oder hinzufügen**

Geht es ans Addieren oder Subtrahieren von Monaten, so können sich in der Praxis Probleme ergeben, wie Lis-

```
select
  numtodsinterval(
    to_date('2009-09-21 14:45', 'YYYY-MM-DD HH24:MI') -
    to_date('2009-07-21 07:30', 'YYYY-MM-DD HH24:MI')
    , 'DAY'
  ) DATETIME_INTERVAL
from dual;

DATETIME_INTERVAL
-----
+0000000062 07:14:59.999999999
```

Listing 4

```
select
  (
    to_date('2009-09-21 14:45', 'YYYY-MM-DD HH24:MI') -
    to_date('2009-07-21 07:30', 'YYYY-MM-DD HH24:MI')
  ) day(9) to second DATETIME_INTERVAL
from dual;

DATETIME_INTERVAL
-----
+0000000062 07:15:00.000000000
```

Listing 5

```
select
  extract(
    HOUR from
    (
      to_date('2009-09-21 14:45', 'YYYY-MM-DD HH24:MI') -
      to_date('2009-07-21 07:30', 'YYYY-MM-DD HH24:MI')
    ) day(9) to second
  ) INTERVAL_HOURS
from dual;

INTERVAL_HOURS
-----
7
```

Listing 6

```
select
  extract(
    MINUTE from
    (
      to_date('2009-09-21 14:45', 'YYYY-MM-DD HH24:MI') -
      to_date('2009-07-21 07:30', 'YYYY-MM-DD HH24:MI')
    ) day(9) to second
  ) INTERVAL_MINUTES
from dual;

INTERVAL_MINUTES
-----
15
```

Listing 7

```
select
  to_date('2009-09-21 14:45', 'YYYY-MM-DD HH24:MI') + (12/24) NUMERIC
  to_date('2009-09-21 14:45', 'YYYY-MM-DD HH24:MI') + interval '12'
  hour INTERVAL
from dual;
```

NUMERIC	INTERVAL
22.09.2009 02:45:00	22.09.2009 02:45:00

### Listing 8

```
SQL> select systimestamp ts_val, systimestamp + 1 date_val from dual;
```

TS_VAL	DATE_VAL
05.10.2009 09:59:28,213998 +02:00	06.10.2009 09:59:28

1 Zeile wurde ausgewählt.

### Listing 9

```
select
  to_date('2012-01-31', 'YYYY-MM-DD') + interval '1'
  month
from dual;
```

FEHLER in Zeile 2:  
ORA-01839: Datum für angegebenen Monat nicht gültig

### Listing 10

```
select
  add_months(to_date('2012-01-31', 'YYYY-MM-DD'), 1) plusmonat,
  add_months(to_date('2012-03-31', 'YYYY-MM-DD'), -1) minusmonat
from dual;
```

PLUSMONAT	MINUSMONAT
29.02.2012 00:00:00	29.02.2012 00:00:00

### Listing 11

```
declare
  datum date;
  "TAGE" number := 1;
begin
  datum := sysdate + interval "TAGE" day;
end;
```

FEHLER in Zeile 5:  
ORA-06550: Zeile 5, Spalte 31:  
PLS-00103: Fand das Symbol "TAGE" als eines der ...

### Listing 12

ting 10 zeigt. Addiert man einen Monat zum 31. Januar, kommt der 31. Februar heraus. Dieses Datum gibt es nicht, also wird eine Fehlermeldung ausgelöst. Dieses Problem tritt nur bei der Addition oder Subtraktion von Monaten zu einem Datum auf und auch nur dann, wenn es den Tag des alten Monats im neuen Monat nicht gibt. In der Praxis sind diese Fälle aber gar nicht so selten. Speziell, wenn es ans Addieren oder Subtrahieren von Monaten geht, sollte man also eine der folgenden Möglichkeiten verwenden:

- Wer mit DATE arbeitet, kann die Funktion ADD\_MONTHS verwenden (siehe Listing 11).
- Nutzt man dagegen TIMESTAMP, so funktioniert ADD\_MONTHS zwar auch, die Datenbank wandelt implizit aber wieder in ein DATE um. Informationen wie Millisekunden oder die Zeitzone gehen also verloren. Eine Alternative ist es, mit dem ersten Tag des nächsten Monats zu arbeiten, anschließend einen Monat hinzuzufügen und dann wieder einen Tag abzuziehen.

### Datums-Intervalle

Auch in PL/SQL kann man mit Intervall-Datentypen arbeiten – allerdings sind ein paar Besonderheiten zu beachten. Die erste ist, dass die „INTERVAL „XX“-Syntax nicht mit Variablen, also dynamisch verwendet werden kann. Listing 12 löst einen Fehler aus, denn in PL/SQL braucht es eine Variable, die als Intervall-Datentyp definiert ist (siehe Listing 13).

Intervalle können auch aus Zahlenwerten heraus berechnet werden. Möchte man wissen, wie viel 2,65 Tage genau sind, dann lässt sich das einfach ermitteln (siehe Listing 14). Für größere Zeit-Intervalle gibt es neben dem Intervall „Day-To-Second“ auch „Year-To-Month“ (siehe Listing 15).

In der Praxis besteht oft der Bedarf nach einer Kombination. Werden zwei Datumswerte voneinander abgezogen, so können durchaus sowohl „Year-To-Month“ als auch „Day-To-Second“ erforderlich sein. Zunächst ermittelt man dann das Intervall „Year-To-Month“,

zieht dieses vom späteren End-Datum ab und ermittelt dann vom Rest das Intervall „Day-To-Second“. Listing 16 zeigt die Vorgehensweise.

**Fazit**

Die Oracle-Datenbank erlaubt beim Umgang mit Datumswerten wesentlich mehr, als man im ersten Moment vermuten würde. Spezielle Kenntnis über die Intervall-Datentypen ist bei der Umsetzung von Datums-Arithmetik sehr wichtig, denn die richtige Anwendung der vorhandenen Funktionen kann in einigen Fällen sehr viel Aufwand für umständliche Programmierung sparen.

Eine weitere, sehr wichtige Besonderheit des `TIMESTAMP`-Datentyps ist die Unterstützung von Zeitzonen. Das Umrechnen eines Datums von einer Zeitzone in eine andere ist mit `TIMESTAMP` völlig unproblematisch. Eine ausführliche Betrachtung würde jedoch den Rahmen dieses Artikels sprengen, deshalb sei auf das entsprechende Blog-Posting des Autors verwiesen.

**Weitere Informationen**

1. Blog-Posting des Autors: <http://sql-plsql-de.blogspot.co.uk/2009/11/date-time-stamp-und-zeitzonen-in-sql-und.html>
2. Oracle Database SQL Language Reference: [http://docs.oracle.com/cd/E11882\\_01/server.112/e26088/toc.htm](http://docs.oracle.com/cd/E11882_01/server.112/e26088/toc.htm)

Carsten Czarski  
 carsten.czarski@oracle.com  
<http://twitter.com/cczarski>  
<http://sql-plsql-de.blogspot.com>



```
declare
    v_zeit timestamp;
    v_hrs interval day(9) to second;
begin
    -- 0 days, 12 hours, 0 minutes, 0 seconds
    v_hrs := '0 12:00:00';
    v_zeit := to_timestamp(
        '2009-09-21 14:45', 'YYYY-MM-DD HH24:MI:SS'
    ) + v_hrs;
    dbms_output.put_line(v_zeit);
end;
```

Listing 13

```
Select numtodsinterval(2.65, 'DAY') interval from dual;

INTERVAL
-----
+0000000002 15:36:00.000000000
```

Listing 14

```
select numtoyminterval(4.55, 'YEAR') intervall from dual;

INTERVALL
-----
+0000000004-06
```

Listing 15

```
declare
    v_enddate date :=
        to_date('2009-09-21 14:45', 'YYYY-MM-DD HH24:MI:SS');
    v_startdate date :=
        to_date('2007-07-15 07:30', 'YYYY-MM-DD HH24:MI:SS');
    v_ym interval year(9) to month;
    v_ds interval day(9) to second;
begin
    -- Jahre und Monate ermitteln
    v_ym := (v_enddate - v_startdate) year to month;
    -- Jahr und Monate vom Enddatum abziehen
    v_enddate := v_enddate - v_ym;
    -- Tage, Stunden, Minuten und Sekunden ermitteln
    v_ds := (v_enddate - v_startdate) day(9) to second;

    dbms_output.put_line('Years:   '||extract(YEAR from v_ym));
    dbms_output.put_line('Months: '||extract(MONTH from v_ym));
    dbms_output.put_line('Days:   '||extract(DAY from v_ds));
    dbms_output.put_line('Hours:  '||extract(HOUR from v_ds));
    dbms_output.put_line('Minutes: '||extract(MINUTE from v_ds));
    dbms_output.put_line('Seconds: '||extract(SECOND from v_ds));
end;
/

Years:  2
Months: 2
Days:   6
Hours:  7
Minutes: 15
Seconds: 0
```

Listing 16

Wenn es darum geht, effiziente Abfragen zu erzeugen, gibt es einige wenige Schlüsselkonzepte, die es zu verstehen gilt. Obwohl es „kostenbasierter Optimierer“ heißt, sind es nicht primär die Kosten, auf die wir schauen müssen, um zu verstehen, warum der Cost-Based Optimizer (CBO) bestimmte Entscheidungen für die Erstellung von Ausführungsplänen getroffen hat.

# Grundlagen zum Cost-Based Optimizer

Randolf Geist, unabhängiger Berater

Der Artikel stellt die Schlüsselkonzepte vor und zeigt, warum die Mengen- und Selektivitäts-Abschätzungen des CBO so elementar wichtig sind und welche Informationen er derzeit (noch) nicht berücksichtigt. Dies bedeutet auch, dass man manchmal wesentlich mehr als der CBO über seine Daten und Applikationen weiß und ihn daher in die richtige Richtung lenken muss, um effiziente Ausführungspläne zu generieren.

Grundlegend gibt es bei der Suche nach einem effizienten Ausführungsplan drei entscheidende Fragen, die zu klären sind:

1. Wie viele Zeilen werden erzeugt? / Welche Datenmenge wird erzeugt?
2. Wie sind die zu durchsuchenden Daten organisiert – eher zusammenhängend oder durcheinandergemischt (gemäß dem Suchkriterium)?
3. Wie groß ist die Wahrscheinlichkeit, dass die zu durchsuchenden Daten bereits im Cache sind?

Die Beantwortung dieser drei Fragen ist so wichtig, weil es vereinfacht ausgedrückt zwei Strategien gibt, wie die Daten verarbeitet werden können:

- *Ein großer „Job“*  
Hier ist vor allem die Frage nach der Datenmenge entscheidend für die Effizienz-Abschätzung der Operation
- *Mehrere kleinere „Jobs“*  
Hier sind die Fragen nach „Wie viele Zeilen?“ (Wie oft muss der kleine Job ausgeführt werden?), „Wie sind die Daten organisiert?“ (Wie groß ist der Aufwand pro Ausführung?) und dem „Caching“ (Welcher Anteil der Daten liegt bereits im Cache?) entscheidend.

Werden die Fragen in diesem Zusammenhang richtig beantwortet, lässt sich die passende Strategie auswählen. Umgekehrt ist es wahrscheinlich, dass man sich bei falscher Beantwortung der Fragen für eine unpassende Strategie entscheidet, die weniger effizient sein kann, zu längerer Ausführungszeit der Abfrage führt und damit auch andere, konkurrierende Prozesse beeinflussen kann.

## Der Cost-Based Optimizer

Es ist nicht überraschend, dass der CBO auf der Suche nach dem effizientesten Ausführungsplan versucht, ganz ähnliche Fragestellungen zu beantworten. Interessanterweise adressiert er aber nicht alle drei oben genannten Fragen im gleichen Umfang: Die erste Frage nach der Anzahl der Zeilen und des Datenvolumens wird ausführlich behandelt, wobei der CBO hier auf recht einfache Weise in die Irre geführt werden kann.

Schon bei der zweiten Frage nach der Organisation der Daten gibt es nur eine einzige Information, die dem CBO zur Verfügung steht: der sogenannte „Clustering Factor“ von Indizes. Während dieser eine in vielen Fällen ausreichende Information für die Entscheidung darstellt, wie auf eine einzelne Tabelle zugegriffen wird (Zugriff per vorsortiertem Index oder Full Table Scan über die gesamte Tabelle), gibt es je nach Ausführungsplan für Operationen, die verschiedene Datenquellen verknüpfen, überhaupt keine Information für den CBO, wie diese in Relation zueinander organisiert sind.

Dies festzustellen ist eine Aufgabe, die nach heutigem Stand der Technik wahrscheinlich zu viel Zeit und Res-

sourcen in Anspruch nehmen würde, da es viele verschiedene Möglichkeiten gibt, wie Datenquellen verknüpft werden können: Die Reihenfolge der Datenquellen kann unterschiedlich sein und pro Datenquelle gibt es potenziell viele mögliche Zugriffsarten. Beides kann die Reihenfolge des Datenzugriffs verändern, zum Beispiel „Full Table Scan Tabelle C -> Index1 Tabelle A -> Index Tabelle B“ gegenüber „Index2 Tabelle A -> Index Tabelle C -> Index Tabelle B“, was bedeutet, dass alle möglichen Kombinationen evaluiert werden müssten, um diese Information sinnvoll verarbeiten und gegeneinander abwägen zu können.

Die dritte Frage nach dem Caching von Daten wird derzeit vom CBO überhaupt nicht berücksichtigt. Dieser geht in seinen Berechnungen immer davon aus, dass die Daten nicht im Cache liegen.

Dies alles bedeutet, dass der CBO je nach Daten und Ausführungsplan nur über unzureichende Informationen verfügt, um die genannten Fragen korrekt beantworten zu können und daher leicht eine unpassende Strategie auswählen kann. Weiterhin bedeutet dies bei ausreichendem Wissen über seine Daten und die Abfragen darauf, dass man unter Umständen diese Fragen wesentlich besser als der CBO beantworten und beurteilen kann, ob der CBO eine passende Strategie ausgewählt hat, beziehungsweise dass man ihn dabei unterstützen muss, die optimale zu finden.

## Statistiken

Der CBO wendet im Grunde ein mathematisches Modell auf die ihm zur Verfügung stehenden Eingangsda-

ten (wie Objekt- und System-Statistiken) an und erstellt als Ergebnis einen Ausführungsplan. Eins der grundlegenden Probleme dieser Herangehensweise ist die Tatsache, dass diese Eingangsdaten, also die Statistiken, je nach Art der Abfrage die Daten eventuell nur unzureichend repräsentieren. Verwendet die Abfrage zum Beispiel Ausdrücke, die in den vorberechneten Objekt-Statistiken nicht abgebildet sind (zum Beispiel ein simples „UPPER(T1.NAME) = ‚MEIER‘“), dann hat es der CBO unter Umständen schon schwer, die eigentlich sehr einfache Frage zu beantworten, wie viele Zeilen der Tabelle „T1“ dieses Suchkriterium erfüllen.

**Mengenabschätzungen**

Wie bereits gesagt, versucht der CBO seine Abschätzungen standardmäßig basierend auf den vorberechneten Objekt-Statistiken zu erzeugen. Der Grund dafür liegt darin, dass der CBO darauf optimiert ist, den Ausführungsplan möglichst schnell zu erzeugen – für Abfragen, die nur den Bruchteil einer Sekunde dauern, würde das Untersuchen der tatsächlichen Daten („Wie viele Zeilen entsprechen Filterkriterium X?“) im Verhältnis viel zu lange dauern. Der CBO verfügt jedoch unter dem Namen „Dynamic Sampling“ über diese Funktionalität. Diese kommt allerdings standardmäßig nur in bestimmten Fällen zum Einsatz, zum Beispiel wenn überhaupt keine

```
select
    count(t2.attr2)
from
    t1
    , t2
where
/*-----*/
    t1.attr1 = 1
and
    t1.attr2 = 1
/*-----*/
and
    t1.fk = t2.id
;
```

Listing 2

Objekt-Statistiken vorliegen. Das hat aber zur Folge, dass Abfragen auf Informationen, die von den Objekt-Statistiken nicht oder nur unzureichend abgedeckt sind, den CBO sehr leicht in die Irre führen können.

Dazu ein einfaches Beispiel: Zwei Tabellen „T1“ und „T2“ (jeweils 1.000.000 Zeilen) sollen miteinander verknüpft werden, „T1“ hat einen Fremdschlüssel auf „T2“ und „T2“ hat einen eindeutigen Index auf dem Primärschlüssel. Zusätzlich wird ein Filter auf „T1“ angewendet. Listing 1 zeigt, dass die gefilterten Spalten „ATTR1“ und „ATTR2“ der Tabelle „T1“ eine ungleichmäßige Datenverteilung haben.

Wird nun auf „ATTR1“ oder „ATTR2“ gefiltert, scheint ein Histogramm von Vorteil zu sein, um den CBO über die ungleiche Verteilung der Daten in den Spalten zu informie-

ren. Wird dies gemacht, kann man die zwei oben erwähnten Strategien leicht demonstrieren. Listing 2 zeigt die Abfrage. Betrachtet man die obige Datenverteilung in „T1.ATTR1“ und „T1.ATTR2“, wird offensichtlich, dass 900.000 Zeilen von „T1“ dieses Suchkriterium erfüllen. Sofern der CBO dies erkennt, würde er wahrscheinlich die Strategie „ein großer Job“ auswählen, da die „Kleine Job“-Strategie hier bedeuten würde, 900.000-mal auf „T2“ per Index-Zugriff in einer Schleife zuzugreifen, was üblicherweise deutlich ineffizienter wäre. Listing 3 zeigt den vom CBO automatisch ausgewählten Ausführungsplan.

Aufgrund der Histogramme hat der CBO die Mengenabschätzung für den Filter auf „T1“ im richtigen Bereich gemacht (Operation ID 3: 819.000 geschätzte (E-Rows) anstatt 900.000 tatsächliche (A-Rows)-Zeilen) und automatisch die „Ein großer Job“-Strategie gewählt. Dies kann auch anhand der „Starts“-Spalte verifiziert werden – jede Operation des Ausführungsplans ist zur Laufzeit genau einmal ausgeführt worden.

Dazu eine Anmerkung: Die normalerweise im Ausführungsplan nicht angezeigten Spalten „Starts“ und „A-Rows“ können über einen speziellen Modus aktiviert werden, indem man entweder den Hint „GATHER\_PLAN\_STATISTICS“ verwendet oder den Parameter „STATISTICS\_LEVEL“ auf „ALL“ setzt (nur auf Session-Ebe-

ATTR1	ATTR2	COUNT
1	1	900000
90001	90001	10
90002	90002	10
90003	90003	10
90004	90004	10
90005	90005	10
90006	90006	10
90007	90007	10
90008	90008	10
90009	90009	10
.	.	.

Listing 1

Id	Operation	Name	Starts	E-Rows	A-Rows
0	SELECT STATEMENT		1		1
1	SORT AGGREGATE		1	1	1
* 2	HASH JOIN		1	819K	900K
* 3	TABLE ACCESS FULL	T1	1	819K	900K
4	TABLE ACCESS FULL	T2	1	1000K	1000K

Predicate Information (identified by operation id):

```

2 - access("T1"."FK"="T2"."ID")
3 - filter(("T1"."ATTR1"=1 AND "T1"."ATTR2"=1))
    
```

Listing 3

```

select
    count(t2.attr2)
from
    t1
    , t2
where
/*-----*/
    t1.attr1 = 90001
and
    t1.attr2 = 90001
/*-----*/
and
    t1.fk = t2.id
;

```

Listing 4

ne zu empfehlen, da es einen deutlichen Overhead in der Ausführung mit sich bringt). Die zusätzlichen Spalten können dann über einen Aufruf von „DBMS\_XPLAN.DISPLAY\_CURSOR“ mit der Formatierungsoption „ALLSTATS LAST“ erzeugt werden. Wird eine ähnliche Abfrage durchgeführt, bei der nur wenige Zeilen von „T1“ die Filterbedingung erfüllen (siehe Listing 4), erhält man den in Listing 5 gezeigten Ausführungsplan.

Der CBO hat hier für die Filterung der Tabelle „T1“ (Operation ID = 4) anstatt 819.000, wie im vorherigen Beispiel, nur eine Zeile geschätzt. Das ist zwar einerseits um den Faktor „10“ falsch, da zehn Zeilen die Filterbedingung erfüllen, liegt jedoch andererseits hier grundsätzlich im richtigen Bereich. Folgerichtig hat der CBO die „Kleine Jobs“-Strategie gewählt: Auf die Tabelle „T2“ wird über den eindeutigen Index zugegriffen, und zwar zehnmal (Starts=10) in einer Schleife (NESTED LOOP), da dies effizienter erschien, als die gesamte Tabelle „T2“ in einem großen Schritt zu verarbeiten. Das Interessante an dem Beispiel ist, dass sich an dem Zugriff auf Tabelle „T1“ nichts ändert – es ist in beiden Beispielen ein sogenannter „Full Table Scan“, da kein sinnvoller alternativer Zugriffsweg zur Verfügung gestellt wurde.

Entgegen der häufig vorherrschenden Meinung kann also die Mengenabschätzung für eine Tabelle nicht nur beeinflussen, wie auf die Tabelle selbst zugegriffen wird, sie kann auch die Zugriffsart und die Reihenfolge ande-

rer Operationen des Ausführungsplans maßgeblich prägen. Verwendet man eine einfache Abwandlung der ersten Abfrage, die den Großteil von „T1“ zurückliefert und besser mit der „Ein großer Job“-Strategie verarbeitet wird, dann wird das noch klarer (siehe Listing 6).

Durch die Verwendung der TRUNC-Funktion (Abschneiden von Nachkommastellen) verändert sich das Ergebnis der Abfrage nicht, da es sich um Ganzzahlen in ATTR1 und ATTR2 handelt. Der CBO kennt dies alles jedoch nicht und sieht nur die Funktion „TRUNC(...)“. Für diesen Ausdruck liegen aber keine Statistiken vor – vor allem kann das Histogramm auf den Spalten „ATTR1“ und „ATTR2“ nicht mehr verwendet werden – und der CBO fällt zurück auf vorgegebene Standardwerte, die nichts mit den eigentlichen Daten zu tun haben (siehe Listing 7).

Wie erwartet, hat sich am Zugriff auf „T1“ nichts geändert, es ist immer noch ein Full Table Scan. Allerdings ist die Abschätzung bezüglich der resultierenden Zeilenanzahl von „T1“ grob falsch: 100 anstatt der bekannten 900.000. Dies hat dazu geführt, dass der CBO die falsche Strategie gewählt hat: Der Ausführungsplan greift 900.000-mal per „Kleine Job“-Strategie auf die Tabelle „T2“ zu, was je nach Tabellen- und Cache-Größe sowie der Verteilung der Daten in der Tabelle

```

select
    count(t2.attr2)
from
    t1
    , t2
where
/*-----*/
    trunc(t1.attr1) = 1
and
    trunc(t1.attr2) = 1
/*-----*/
and
    t1.fk = t2.id
;

```

Listing 6

„T2“ zu einer sehr langen Laufzeit führen kann.

Der wichtigste Punkt hier ist also, dass auch einfache Mengenabschätzungen für Filteroperationen auf einzelnen Tabellen maßgebliche Auswirkung auf den gesamten Ausführungsplan haben können. Von daher ist es elementar wichtig, dass der CBO diese Abschätzungen im richtigen Bereich macht.

Im Laufe der verschiedenen Datenbank-Versionen sind zu diesem Zweck verschiedene Funktionalitäten hinzugefügt worden, angefangen von „Function-Based Indizes“ in der Version 8.1, um Ausdrücke zu indizieren (als Seiteneffekt der Indizierung auch Spalten-Statistiken für den Ausdruck), „Dynamic Sampling“ seit Version 9.2, um dem CBO die Möglichkeit zu geben, einen Blick auf die tatsächlichen

Id	Operation	Name	Starts	E-Rows	A-Rows
0	SELECT STATEMENT		1		1
1	SORT AGGREGATE		1	1	1
2	NESTED LOOPS		1		10
3	NESTED LOOPS		1	1	10
* 4	TABLE ACCESS FULL	T1	1	1	10
* 5	INDEX UNIQUE SCAN	T2_IDX	10	1	10
6	TABLE ACCESS BY INDEX ROWID	T2	10	1	10

Predicate Information (identified by operation id):

```

4 - filter((<<T1"."ATTR1"=90001 AND <<T1"."ATTR2"=90001))
5 - access(<<T1"."FK"="T2"."ID")

```

Listing 5

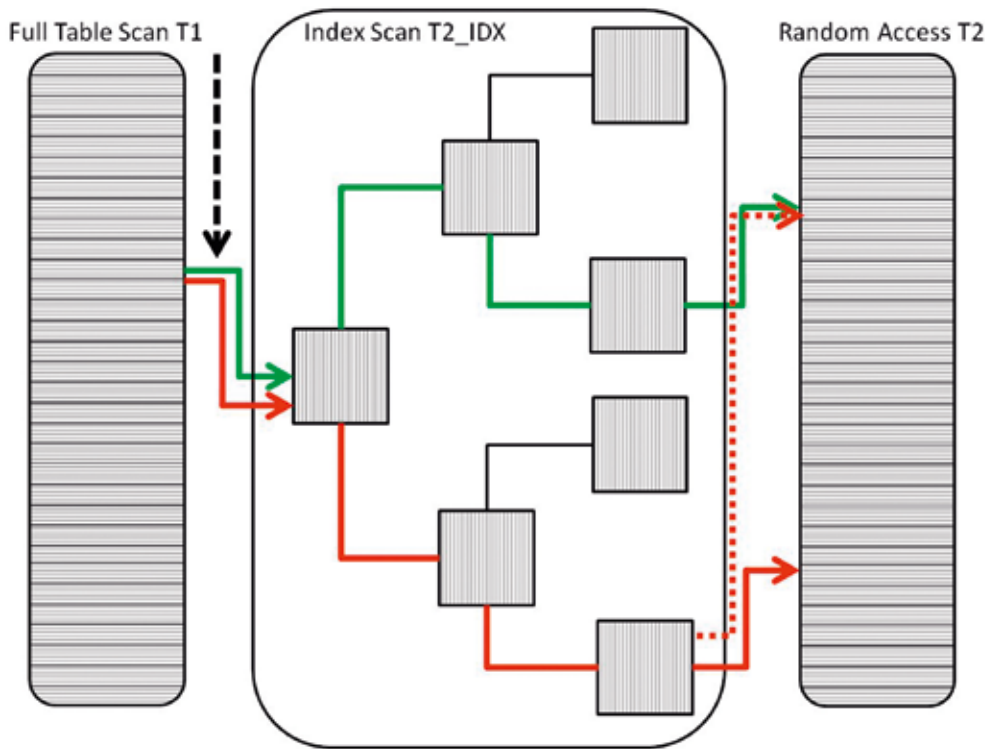


Abbildung 1: Schematische Darstellung zweier Durchläufe (erster „grün“, zweiter „rot“) einer Nested Loop T1->T2\_IDX->T2. Die gestrichelte rote Linie zeigt an, dass der zweite Durchlauf der Nested Loop im gleichen Block der Tabelle „T2“ landet wie der erste, bei der festen Linie muss auf einen anderen Block zugegriffen werden.

Daten zu werfen, bis hin zu „Virtual Columns“ und „Extended Statistics“ in der Version 11.1, die beide ermöglichen, auf Ausdrücken und sogar Spaltengruppen Statistiken zu er-

zeugen, unabhängig von eventuell erzeugten „Function-Based Indizes“. Diese Mengenabschätzungen gilt es zu allererst zu überprüfen, da sie einfach nachvollzogen werden können

(„SELECT COUNT(\*)“ mit der Filter-Bedingung auf der Tabelle genügt) und es ausreichend Möglichkeiten gibt, sie zu korrigieren.

**Organisation der Daten**

Für die Auswahl der richtigen Strategie ist also zum einen die Mengenabschätzung entscheidend. Um die Effizienz eines selektiven Zugriffs (zum Beispiel per Index) im Rahmen der „Kleine Job“-Strategie zu bestimmen, ist aber noch ein anderes Kriterium elementar wichtig: Wie zusammenhängend oder durcheinander sind die Daten in der zu lesenden Tabelle organisiert gemäß der Reihenfolge des jeweiligen Zugriffs?

Nehmen wir das Beispiel: Bei der „Kleine Job“-Strategie wird selektiv auf die Tabelle „T2“ zugegriffen, indem für jede Zeile, die die Filterbedingung auf der Tabelle „T1“ erfüllt, per Index die passende Zeile in „T2“ gesucht wird. Für die Effizienz dieser Operation ist maßgeblich relevant, ob die jeweils passenden Zeilen in der Tabelle „T2“ gemäß der Reihenfolge der Daten aus Tabelle „T1“ zusammenhängend im gleichen Bereich (in den gleichen Blöcken) gespeichert sind oder ob zumeist für jede Zeile auf einen anderen Block zugegriffen werden muss.

Bei diesem Zugriffsmuster ist es meistens diese Organisation der Tabellendaten, die die Effizienz der Operation bestimmt, da Indizes häufig viel kleiner sind als Tabellen und daher viel mehr vom Caching profitieren können, was heißt, dass die Wahrscheinlichkeit für ein Caching der relevanten Index-Blöcke viel größer sein kann als bei den entsprechenden Tabellen-Blöcken. Abbildung 1 stellt schematisch dar, wie der Zugriff von „T1“ auf „T2“ im Rahmen der „Kleine Job“-Strategie aussieht.

Für jede Zeile, die im Rahmen des Full Table Scan der Tabelle „T1“ dem Filterkriterium entspricht (repräsentiert im linken Teil der Grafik), wird der Index „T2\_IDX“ auf einen passenden Eintrag durchsucht. Dafür muss je nach Höhe des Index eine bestimmte Anzahl von sogenannten „Root-und-Branch-Blöcken“ des Index durch-

Id	Operation	Name	Starts	E-Rows	A-Rows
0	SELECT STATEMENT		1		1
1	SORT AGGREGATE		1	1	1
2	NESTED LOOPS		1		900K
3	NESTED LOOPS		1	100	900K
* 4	TABLE ACCESS FULL	T1	1	100	900K
* 5	INDEX UNIQUE SCAN	T2_IDX	900K	1	900K
6	TABLE ACCESS BY INDEX ROWID	T2	900K	1	900K

Predicate Information (identified by operation id):

- 4 - filter((TRUNC("T1"."ATTR1")=1 AND TRUNC("T1"."ATTR2")=1))
- 5 - access("T1"."FK"="T2"."ID")

Listing 7



sucht werden, bis der entsprechende Leaf-Block des Index gefunden wurde, in dem schließlich überprüft werden kann, ob ein passender Eintrag in „T2“ existiert oder nicht (mittlerer Teil der Grafik).

Falls ein passender Eintrag im Index identifiziert wurde, muss normalerweise der Rest der benötigten Daten für diese Zeile, die nicht im Index abgebildet sind, von der Tabelle selbst geholt werden (rechter Teil der Grafik). Obwohl der Zugriff auf die im Index referenzierte Tabellenzeile in den meisten Fällen mit einem einzigen Blockzugriff möglich ist (Ausnahmen sind zum Beispiel sogenannte „Migrated Rows“), bestimmt eben genau dieser Zugriff auf den Tabellen-Block in den meisten Fällen über die Effizienz dieser Operation.

Dies ist so, weil die Index-Blöcke aufgrund der Größe des Index und der Frequenz der Zugriffe meistens im Cache verbleiben können. Die Tabelle ist normalerweise viel größer und es kann je nach Art des Zugriffsmusters viel wahrscheinlicher sein, dass der gesuchte Block nicht im Cache ist und von Platte gelesen werden muss. Es handelt sich bei dem gezielten Lesen eines einzelnen Tabellen-Blocks um einen sogenannten „Random Access“, da für jeden Durchlauf der Schleife und damit für jede einzelne Zeile, die in „T1“ gefunden und für die eine passende Zeile in „T2“ gesucht wird, theoretisch auf einen anderen Block der Tabelle „T2“ zugegriffen werden muss. Im schlechtesten Fall kann dies sogar dazu führen, dass der gleiche Block von „T2“ mehrfach von Platte gelesen werden muss, da der vorherige Zugriff auf den gleichen Block bereits wieder von nachfolgenden Zugriffen aus dem Cache verdrängt wurde.

Wird aber beim Durchlauf der Schleife für eine bestimmte Anzahl an Iterationen immer wieder auf den gleichen Block (oder eine kleinere Anzahl von Blöcken) der Tabelle „T2“ zugegriffen, bleibt der Block nach dem ersten Zugriff im Cache und muss nicht mehr von Platte gelesen werden. Dies kann dramatische Unterschiede für die Effizienz solcher Operationen bedeuten, da ein typischer „Random Access“

auch von sehr schnellen Festplatten heutzutage immer noch zwischen drei und fünf Millisekunden dauert (Massenspeicher ohne rotierende Massen wie SSDs können hier deutlich schneller sein), während der Zugriff auf einen Block im Cache im unteren Mikrosekunden-Bereich liegt.

Legt man diese Zahlen einer einfachen Berechnung zugrunde, so benötigt der Zugriff auf 1.000 Zeilen der Tabelle „T2“ allein für das Lesen der 1.000 Blöcke von Festplatte zwischen drei und fünf Sekunden, unter Umständen auch deutlich länger, während das Zugreifen auf die gleiche Anzahl Blöcke aus dem Cache immer noch im Millisekunden-Bereich liegt.

Allgemein wird das eine Zugriffsmuster („für jede identifizierte Zeile von „T2“ muss auf einen anderen Tabellen-Block zugegriffen werden“) als „Scattered“ bezeichnet, während das andere („für jede identifizierte Zeile von „T2“ kann wiederholt auf die gleichen Tabellen-Blöcke zugegriffen werden“) „Clustered“ genannt wird. Wichtig ist in diesem Zusammenhang zu verstehen, dass je nach Zugriffsart die gleiche Tabelle „Scattered“ für eine Art von Zugriff sein kann, aber „Clustered“ für eine andere. Genauer gesagt kann eine Tabelle normalerweise nur für genau eine Zugriffsart „Clustered“ sein und für alle anderen „Scattered“.

### Proaktives Design

Mit diesem Wissen ist die Optimierung solcher Zugriffe, die als kritisch für eine Anwendung identifiziert werden, proaktiv möglich. Gerade bei OLTP-Anwendungen, die häufig gezielt kleinere Datenmengen suchen, kann eben genau diese Organisation von Daten einen großen Unterschied in Bezug auf die Effizienz beim Datenzugriff bedeuten. Daher sollten idealerweise schon während des Designs der Applikation die Daten und die wichtigsten Abfragen auf diese Daten bekannt sein.

Mithilfe dieses Wissens kann dann evaluiert werden, welche Art der Indizierung oder der Verwendung anderer Speicherungs-methoden den Zugriff auf die Daten entscheidend verbessern kann, ohne beim Modifizieren der Da-

ten zu viel Zeit zu verlieren. Insbesondere die Verwendung von Clustern (Index Cluster oder Hash Cluster) oder Index-Organized Tables (IOTs) bietet je nach Anwendungsfall die Möglichkeit, die Organisation der Daten proaktiv zu beeinflussen und den Zugriff auf die Daten entscheidend zu beschleunigen.

Natürlich können diese alternativen Speicherungs-methoden nicht ungesehen eingesetzt werden. Die genauen Zugriffsmuster sowohl beim Lesen als auch beim Schreiben der Daten müssen bekannt sein, da ansonsten leicht das Gegenteil erreicht werden kann: Sowohl Lese- als auch Schreibzugriff können deutlich ineffizienter werden. Zudem ist ein Wissen über die spezifischen Verhaltensweisen von Clustern und IOTs erforderlich, um vorab abschätzen zu können, welche der alternativen Speicherungs-methoden welche Vor- und Nachteile mit sich bringen. Sind zum Beispiel viele sekundäre Indizes nötig, kann eine IOT zu insgesamt schlechterer Performance führen, da sekundäre Indizes für IOTs grundsätzlich anders funktionieren als für normale Tabellen.

Cluster können aus verschiedenen Gründen sehr ineffizient werden und erlauben bestimmte Operationen und Zugriffsmuster, je nach Variante, nicht. Darüber hinaus unterstützen derzeitige Versionen keine Partitionierung von Clustern. Insgesamt gesehen ist es also sehr wichtig, bereits während der Design-Phase einer Applikation Kernfragen aufzubauen, um die Organisation der Daten in der Datenbank optimal darauf abstimmen zu können.

Randolf Geist  
info@sqltools-plusplus.org



Der Oracle Cost-Based Optimizer ist eine zentrale Komponente der Datenbank und mittlerweile recht ausgereift. In den seltenen Fällen, in denen er allerdings keinen performanten Ausführungsplan ermitteln kann, empfiehlt Oracle – bei entsprechender Lizenzierung – die Verwendung des SQL Tuning Advisors zur automatischen Ermittlung von Verbesserungspotenzial. Ein DBA sollte allerdings auch ohne diesen Assistenten in der Lage sein, ein SQL-Statement manuell zu analysieren.

# Manuelles Oracle SQL-Tuning

Martin Decker, ora-solutions.net

Der Artikel gibt eine Einführung in das manuelle SQL-Tuning anhand eines einfachen 2-Table-Join-Beispiels. Es werden die Entscheidungen des Cost-Based Optimizers nachvollzogen und die wichtigsten Aspekte anhand des Beispiels beleuchtet, etwa Join Method, Join Order, Access Paths, Optimizer Goal, Statistiken und insbesondere Histogramme im Zusammenhang mit Bind-Variablen. Im Abschluss werden die verfügbaren Werkzeuge zur Veränderung eines Ausführungsplans erläutert.

## Identifikation des relevanten SQL-Statements

Als erstes gilt es, das relevante, problematische SQL-Statement zu identifizieren, das auf Performance-Optimierung untersucht werden soll. Hierfür gibt es verschiedene Möglichkeiten:

- AWR (Diagnostic Pack License)/ Statspack Report über bestimmtes Intervall
- SQL Tracing (Trace Event 10046 beziehungsweise DBMS\_MONITOR)
- Active Session History (erfordert Diagnostic-Pack-Lizenz)
- Oracle Enterprise Manager – DB Performance Page (erfordert Diagnostic-Pack-Lizenz)
- SQL Performance Monitor (erfordert Tuning-Pack-Lizenz)

Ist das SQL-Statement identifiziert, beginnen die eigentliche Untersuchung beziehungsweise die Optimierung. Die Vorgehensweise ist in drei Phasen unterteilt:

- *Analyse-Phase*
  - Ermittlung der Ausführungs-Statistiken (Elapsed Time, Buf-

- fer Gets, Disk Reads, Executions) aus V\$SQLSTATS beziehungsweise DBA\_HIST\_SQLSTAT
- Ermittlung Ausführungsplan (DBMS\_XPLAN)
- optional: Ermittlung Bind-Variablen (v\$sql\_bind\_capture beziehungsweise DBMS\_XPLAN)
- *Reproduktions-Phase*
  - Reproduzieren der sub-optimalen Ausführungs-Performance mittels SQL\*Plus beziehungsweise SQL Developer
- *Tuning-Phase*
  - Reduzierung der Buffer Gets des Statements durch iterative Veränderung und Messung der Wirkung

In der Analyse-Phase geht es darum, die Performance-Daten der problematischen Ausführung zu ermitteln. Dies sind vor allem „Elapsed Time“, „Buffer Gets“, „Disk Reads“ und „Executions“. Die wichtigste Metrik hierbei sind die „Buffer Gets“. Dieser Begriff definiert die logischen I/O-Operationen, um Datenblöcke aus dem Arbeitsspeicher zu lesen. Während die Metriken „Elapsed Time“ und „Disk Reads“ abhängig von der Caching-Ratio der Datenblöcke stark variieren können, bleibt die Metrik „Buffer Gets“ unabhängig vom

Caching stets konstant. In der Praxis kann man feststellen, dass Entwicklungssysteme häufig mit sehr großen Buffer Caches und relativ kleinen Datenmengen betrieben werden. Zudem liegt der Fokus meist nur auf der Antwortzeit der Statements. Damit fällt ein problematischer Zugriffsplan sehr oft erst auf, wenn mit großen Datenmengen im Produktionsumfeld gearbeitet wird. Der DBA wird dann damit konfrontiert, im Produktionsumfeld unter Zeitdruck das Performance-Problem zu analysieren und zu beheben.

Als Richtwert sollte versucht werden, die Anzahl der Buffer Gets pro zurückgelieferter Zeile kleiner als zehn zu halten. Dies gilt nicht für Aggregationen (count, sum, avg etc.) und Joins. Listing 1 zeigt eine Query, die die Belastung der Query für die Datenbank ermittelt. Zudem wird der Ausführungsplan inklusive Bind-Variablen zum Zeitpunkt des Parsing erstellt (siehe Listing 2).

Der Oracle Cost-Based Optimizer (CBO) erstellt zum Zeitpunkt des Parsing den Ausführungsplan. Dieser ist abhängig von einigen Input-Parametern:

- Datenbank-Version, z.B. 11.2.0.1
- Initialisierungsparameter, etwa „optimizer\_features\_enable=11.1.0.7.0“

```
select sql_fulltext, child_number, plan_hash_value, buffer_gets/
executions, disk_reads/executions, elapsed_time/executions from
v$sqlstats where sql_id = '<sql_id>' and executions > 0;
```

Listing 1

```
select * from table(dbms_xplan.display_cursor('<sql_id>', '<child_
number>', ,TYPICAL +PEEKED_BINDS'));
```

Listing 2

- Objekt-Statistiken (Table, Column, Index Statistics gesammelt mit „dbms\_stats.gather\_database|schema|table\_stats“)
- System-Statistiken (gesammelt mit „dbms\_stats.gather\_system\_stats“)
- Datenbank-Schema (Tabellenstruktur, vorhandene Indizes etc.)
- Plan-Stability-Informationen (Stored Outlines, ab 10g: SQL Profiles, ab 11g: SQL Plan Baselines)
- Cardinality Feedback (ab 11g R2: Rück-Übermittlung der Row-Source-Operation-Cardinalities nach Abschluss der Ausführung an den Optimizer)
- aktuelles Datum (wenn beispielsweise Query die Funktion „sysdate“ enthält)

Ändert sich einer dieser Input-Parameter, kann dies dazu führen, dass der Optimizer einen neuen Ausführungsplan erstellt. Ein historisch häufiges Problem hierbei ist die sogenannte „Plan Instability“. Durch Änderung des Ausführungsplans wird ein SQL-Statement zeitweise performant, zu einem anderen Zeitpunkt unperformant ausgeführt. Zur Lösung dieses Problems gibt es mittlerweile mehrere verschiedene Werkzeuge wie Stored Outlines (Standard Edition), SQL Profiles (Enterprise Edition) oder SQL Plan Baselines (Enterprise Edition).

Anschließend wird in der Reproduktions-Phase versucht, die problematische Ausführungsperformance zu reproduzieren. Falls das Statement Bind-Variablen beinhaltet, dürfen diese vorerst nicht durch Literale ersetzt werden. Falls die Bind-Variablen vom Datentyp „DATE“ sind, empfiehlt es sich, das Statement mit SQL Developer statt SQL\*Plus zu reproduzieren.

Unter Umständen ist das Problem allerdings nicht reproduzierbar. Teilweise ist die Daten-Konstellation, die zum Problem führt, nur kurzzeitig vorhanden. Später, wenn der DBA die Analyse durchführt, ist das Problem aufgrund der anderen Datenkonstellation nicht mehr sichtbar. In diesen Fällen muss auf SQL Tracing mittels „Trace Event 10046“ oder „DBMS\_MONITOR“ ausgewichen werden.

Ist es gelungen, denselben problematischen Zugriffsplan beziehungsweise

die problematische Ausführungsdauer zu generieren, so wird nun in der Tuning-Phase begonnen, die Ausführungsdauer und den Ressourcen-Verbrauch des Statements durch verschiedenste Maßnahmen zu reduzieren. Im Gegensatz zum Instance Tuning wird nicht versucht, die Ausführungszeit durch Vergrößerung von Caches und Reduzierung von Disk Reads zu verringern. Der Fokus liegt hier auf der Reduzierung der Logical I/Os (Buffer Gets) der SQL Query. Die einzelnen Maßnahmen sind mannigfaltig und abhängig vom Statement. Beispiele hierfür sind: Erstellung eines Index, Aktualisierung der Optimizer-Statistiken, Optimizer Hints, Aktivierung einer bestimmten Optimizer-Version, Deaktivierung von bestimmten Optimizer-Features, Ändern der Datenverteilung durch Reorganisation und Sortierung, Rewrite des SQL-Statements etc. Das folgende Beispiel verwendet die beiden Tabellen „CUSTOMERS“ und „ORDERS“ (siehe Abbildung 1). Die Datenverteilung hat folgendes Profil:

- Customers: 100.000 rows
- Orders: 1.000.000 rows
  - CUSTOMER\_ID: 70 Prozent von einem einzelnen Customer, 20 Prozent von einem einzelnen anderen Customer, die restlichen 10 Prozent von ca. 64.000 verschiedenen Customern
  - ORDER\_STATUS: zwei verschiedene Order-Status, ungleichmäßig verteilt: 900.000 COMPLETED, 100.000 PENDING

Listing 3 zeigt eine SQL Query, die nun im Detail analysiert wird.

**Step 1: Analyse-Phase**

Bei der Analyse kann festgestellt werden, dass das Statement 0,353 Sekunden pro Ausführung dauert und 22.300 Buffer Gets dafür aufgewendet werden müssen. Der Ausführungsplan zeigt einen Nested Loop Join und als Access Paths werden Index Lookups verwendet. Das Statement enthält eine Bind-Variable, die auf eine bestimmte

```

SELECT * FROM DEMO.CUSTOMERS C,
          DEMO.ORDERS O
WHERE O.CUSTOMER_ID = C.CUSTOMER_ID      -- join predicate
      AND C.CUSTOMER_ID = :v1            -- filter predicate
      and O.ORDER_STATUS = 'PENDING'    -- filter predicate
ORDER BY O.ORDER_DATE                   -- sorting
;
    
```

Listing 3

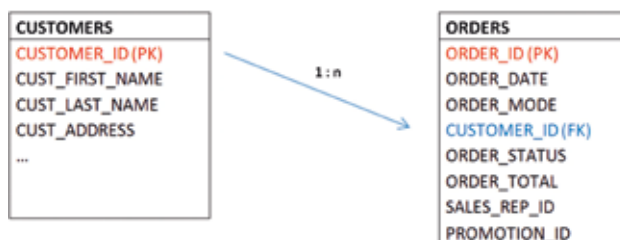


Abbildung 1: Vereinfachtes ER-Diagramm der Beispiel-Tabellen

```

SQL> select child_number as child, plan_hash_value as phv, buffer_
gets/executions as gets_per_exe, disk_reads/executions as disk_per_
exe, elapsed_time/executions ela_per_exe, executions as exe from v$sql
where sql_id = 'g7c03r8pn1ymq';
    
```

CHILD	PHV	GETS_PER_EXE	DISK_PER_EXE	ELA_PER_EXE	EXE
0	3311696933	22300	0	353112.938	81

Abbildung 2: Ressourcen-Verbrauch

```
SQL> select * from table(dbms_xplan.display_cursor('g7c03r8pn1ymq','0','TYPICAL +PEEKED_BINDS'));
Plan hash value: 3311696933
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT				8 (100)	
1	SORT ORDER BY		8	1560	8 (13)	00:00:01
2	NESTED LOOPS		8	1560	7 (0)	00:00:01
3	TABLE ACCESS BY INDEX ROWID	CUSTOMERS	1	150	2 (0)	00:00:01
* 4	INDEX UNIQUE SCAN	CUSTOMERS_PK	1		1 (0)	00:00:01
* 5	TABLE ACCESS BY INDEX ROWID	ORDERS	8	360	5 (0)	00:00:01
* 6	INDEX RANGE SCAN	CUSTOMER_IDX	16		2 (0)	00:00:01

```
Peeked Binds (identified by position):
```

```
1 - :V1 (NUMBER): 1001
```

```
Predicate Information (identified by operation id):
```

```
PLAN_TABLE_OUTPUT
```

```
4 - access("C"."CUSTOMER_ID"=:V1)
5 - filter("O"."ORDER_STATUS"='PENDING')
6 - access("O"."CUSTOMER_ID"=:V1)
```

Abbildung 3: Ressourcen-Verbrauch und Ausführungsplan

CUSTOMER\_ID filtert. Damit ist klar, dass die Query nur eine einzelne Zeile aus der Tabelle „CUSTOMERS“ treffen wird, da die Spalte „CUSTOMER\_ID“ als Primary Key definiert ist. Für diesen Kunden werden dann alle offenen Bestellungen abgefragt und nach dem Bestelldatum (ORDER\_DATE) sortiert. Beim Ausführungsplan ist zu sehen, welche Bind-Variable beim Parsing vorhanden war. (hier: 1001).

### Exkurs: Binds vs. Literals

Noch ein Gedanke zur Query: Immer wieder gibt es Konflikte zwischen DBAs und Entwicklern über die Verwendung von Bind-Variablen. Deren Zweck besteht darin, den Parsing-Overhead zu reduzieren und möglichst viele Statements im Shared Pool zu cachieren und wiederzuverwenden. Die Query enthält eine Bind-Variable für die CUSTOMER\_ID und ein Literal-String für den ORDER\_STATUS. Wenn statt der Bind-Variablen für die CUSTOMER\_ID ein Literal verwendet worden wäre, dann hätte das die Konsequenz, dass für jede CUSTOMER\_ID (potenziell 100.000) ein einzelnes, unabhängiges SQL-Statement verwendet wird. Der Effekt des Caching von SQL-Statements und deren Metadaten wäre damit nicht möglich.

Bei der Spalte „ORDER\_STATUS“ gibt es nur zwei verschiedene Werte (NDV), die zudem ungleich verteilt sind: „COMPLETED“ und „PENDING“. Deshalb ist dieses Prädikat ein idealer Kandidat für ein Literal. Im schlimmsten Fall gibt es dadurch zwei verschiedene SQL-Statements, die bis auf den Literal-Wert identisch sind. Hingegen gibt es bei Verwendung des Literals die Möglichkeit, ein Histogramm auf die Spalte zu berechnen und damit dem Optimizer die genaue Verteilung der Rows auf die beiden Werte mitzuteilen. Die Abbildungen 2 und 3 zeigen die Belastung der Datenbank durch die Query.

Die Spalte „Rows“ definiert die „Cardinality“, also die geschätzte Anzahl der Zeilen, die mit dieser Operation zurückgeliefert werden. Es ist zu beachten, dass alle Informationen des Ausführungsplans lediglich Schätzungen des Optimizers zum Zeitpunkt des Parsings sind und es nicht ersichtlich ist, an welcher Zeile (Row Source Operation) des Ausführungsplans der Großteil der Buffer Gets aufgewendet wird. Für die weitere Analyse ist deshalb diese Darstellung unzureichend.

### Step 2: Reproduktions-Phase

Bei der Reproduktion wird nun der Optimizer Hint „GATHER\_PLAN\_STATIS-

TICS“ eingefügt. Direkt nach dem eigentlichen SQL-Statement wird dann der Ausführungsplan mittels „DBMS\_XPLAN“ abgefragt, wobei der Format-Parameter auf „ALLSTATS LAST“ gesetzt wird. Um den eigentlichen Query-Output auszublenden, kann „termout off“ verwendet werden. Dabei ist die Definition der Bind-Variablen in SQL\*Plus-Syntax zu beachten. Beim Datentyp ist zu bedenken, dass derselbe Datentyp verwendet wird, wie vorher im Ausführungsplan angezeigt wurde (siehe Listing 4).

Abbildung 4 zeigt den Ausführungsplan, nachdem diese SQL-Datei mit SQL\*Plus ausgeführt wurde. Man sieht nun einen wesentlich detaillierteren Ausführungsplan mit zusätzlichen beziehungsweise anderen Spalten. Die Spalte „STARTS“ gibt an, wie oft eine „Row Source Operation“ ausgeführt wurde. Dies ist vorwiegend bei Nested Loop Joins relevant. Die Spalten „E-(estimated) Rows“ und „A-(actual) Rows“ geben Aufschluss darüber, wie akkurat die Cardinality-Schätzung des Optimizers war. In unserem Fall gibt es eine starke Abweichung.

Die Spalte „A-(actual)Time“ zeigt die Ausführungsdauer. Die Spalte „Buffers“ gibt an, wie viele logische I/O-Operationen pro Row-Source-Opera-

Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers
0	SELECT STATEMENT		1		1	00:00:00.58	22300
1	SORT ORDER BY		1	8	1	00:00:00.58	22300
2	NESTED LOOPS		1	8	1	00:00:00.58	22300
3	TABLE ACCESS BY INDEX ROWID	CUSTOMERS	1	1	1	00:00:00.01	3
* 4	INDEX UNIQUE SCAN	CUSTOMERS_PK	1	1	1	00:00:00.01	2
* 5	TABLE ACCESS BY INDEX ROWID	ORDERS	1	8	1	00:00:00.57	22297
* 6	INDEX RANGE SCAN	CUSTOMER_IDX	1	16	700K	00:00:11.72	1467

Predicate Information (identified by operation id):

```

4 - access("C"."CUSTOMER_ID"=:v1)
5 - filter("O"."ORDER_STATUS"='PENDING')

```

Abbildung 4: Reproduziertes SQL-Statement und erweiterter Ausführungsplan

```

set termout off
variable v1 number;
exec :v1 := 1001;
select /*+ GATHER_PLAN_STATISTICS */ * from DEMO.CUSTOMERS C,
      DEMO.ORDERS O
WHERE O.CUSTOMER_ID = C.CUSTOMER_ID -- join predicate
      AND C.customer_id = :v1 -- filter predicate
      and O.order_status = 'PENDING' --filter predicate
order by order_date -- sorting
;
set termout on
set lines 300
set pages 1000
select * from table(dbms_xplan.display_cursor(null,null,'ALLSTATS
LAST'));

```

Listing 4

tion notwendig waren. Man sieht, dass der Großteil davon auf die Zeile 5 fällt. Die Zahlen in der Spalte „Buffers“ sind kumulativ, abhängig davon, ob es eine untergeordnete Row-Source-Operation gibt. In unserem Beispiel beinhaltet die Zahl „Buffers=3“ in Zeile 3 bereits die Zahl „Buffers=2“ der Zeile 4. Ebenso enthält die Zahl „22297“ aus der Zeile 5 bereits die 1467 Buffer Gets aus Zeile 6. In der Row-Source-Operation der Zeile 2 (NESTED LOOPS) enthält die Zahl „22300“ bereits die beiden Nested Loop Join Childs 3 und 5 (beziehungsweise Buffers 3 und 22297).

Der Optimizer hat nun folgende Entscheidungen getroffen:

- **Join Order**

Aufgrund der Tatsache, dass mit einem Equal-Prädikat auf den Primary-Key zugegriffen wird, weiß der Optimizer, dass die Tabelle „Customer“ nur eine Zeile zurücklie-

fern wird. Deshalb ist diese Tabelle diejenige, mit der begonnen wird. Grundsätzlich ist das Ziel, bei der Ausführung möglichst früh möglichst viele Rows (durch Filter etc.) zu eliminieren.

- **Access Paths**

Der Optimizer weiß, dass auf der Spalte „CUSTOMERS.CUSTOMER\_ID“ ein Unique-Index liegt und dies der effizienteste Zugriffspfad zur Tabelle ist. Bei der Tabelle „ORDERS“ gibt es zwei Prädikate: erstens die Einschränkung auf „ORDER\_STATUS“ und zweitens die auf „ORDERS.CUSTOMER\_ID“. Der Optimizer weiß anhand der Column-Statistics (DBA\_TAB\_COL\_STATISTICS), dass die Anzahl der verschiedenen Werte (NDV, Number of Distinct Values) für „ORDERS.CUSTOMER\_ID“ 64236 beträgt.

Die Selektivität wird berechnet mit der Formel: „1/NDV“, das heißt

„1/64236=0,0000155“. Nun wird die Anzahl der Rows der Tabelle (1.000.000) mit der Selektivität multipliziert: „1.000.000\*0,0000155=15,5“. Der Optimizer rundet das auf 16 und schätzt, dass der Kunde 16 Bestellungen hat. Aufgrund der geringen Selektivität entscheidet sich der Optimizer gegen einen Full Table Scan der Tabelle „Orders“ (1 Million Rows) und für einen Index Range Scan auf den Foreign Key Index auf der Spalte „CUSTOMER\_ID“.

- **Join Method**

Zur Auswahl stehen Nested Loop, Hash Join, Sort Merge Join und Merge Join Cartesian. Der Optimizer wählt den Nested Loop Join. Dieser besteht aus einer äußeren Schleife, dem Outer Loop (Zeile 3-4), und einem Inner Loop (Zeile 5-6). Für jede zutreffende Zeile des Outer Loop wird der Inner Loop einmal ausgeführt. Die Anzahl der Wiederholungen wird in der Spalte „Starts“ angezeigt. Aufgrund der Tatsache, dass der Outer Loop nur eine Zeile (einen Kunden) zurückliefert, muss der Inner Loop nur einmal ausgeführt werden.

### Step 3: Tuning-Phase

Die Vorgehensweise beim Tuning ist abhängig vom SQL-Statement. Die folgenden Aspekte werden unter anderem berücksichtigt:

- Welcher Teil des Ausführungsplans die meisten Buffer Gets verursacht

Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers
0	SELECT STATEMENT		1		1	00:00:00.40	6332
1	SORT ORDER BY		1	1	1	00:00:00.40	6332
2	NESTED LOOPS		1	1	1	00:00:00.40	6332
3	TABLE ACCESS BY INDEX ROWID	CUSTOMERS	1	1	1	00:00:00.01	3
* 4	INDEX UNIQUE SCAN	CUSTOMERS_PK	1	1	1	00:00:00.01	2
* 5	TABLE ACCESS BY INDEX ROWID	ORDERS_SORTED	1	1	1	00:00:00.40	6329
* 6	INDEX RANGE SCAN	ORDERS_SORTED_CUST	1	2	700K	00:00:08.14	1467

Predicate Information (identified by operation id):

```

4 - access("C"."CUSTOMER_ID"=:V1)
5 - filter("O"."ORDER_STATUS"='PENDING')
6 - access("O"."CUSTOMER_ID"=:V1)

```

Abbildung 5: Ressourcen-Verbrauch und erweiterter Ausführungsplan nach Table-Reorganisation

- und daher am meisten Potenzial für Verbesserung hat
- Ob es bei den Row Source Operations der Access-Paths eine starke Abweichung zwischen E-(estimated) Rows und A-(actual) Rows gibt
  - Ob im Ausführungsplan ein Nested Loop Join existiert, bei dessen Inner Loop ein ineffizienter Access-Path gewählt wurde, etwa Full Table Scan oder Index Fast Full Scan
  - Ob bei Betrachtung der Spalte „A-Rows“ möglichst früh Zeilen im Ausführungsplan gefiltert werden. Wenn ja, könnte die Join Order, speziell die Auswahl der Start-Tabelle, suboptimal sein. Ob es unrealistisch hohe A-Rows-Werte in bestimmten Row Source Operations gibt. Dies würde auf suboptimale Joins oder kartesische Produkte hinweisen.

### Idee 1: Verbesserung der Cardinality-Schätzung durch Histogramme

Offensichtlich lag der Optimizer bei der Schätzung der Cardinality beim Hauptkunden etwas daneben. Statt der geschätzten 16 Rows werden hier 700.000 Rows geliefert. Das sind 70 Prozent der gesamten Zeilen der Tabelle „ORDERS“. Die Bestell-Daten sind also in Bezug auf Kunden ungleich verteilt (engl.: skewed). Oracle bietet hierfür sogenannte „Histogramme“ an. Diese geben dem Optimizer die Möglichkeit, genauere Daten über die ungleiche Verteilung zu liefern. Allerdings gibt es ein Problem bei Verwendung von Histogrammen und Bind-Variablen. In unserem Fall

wird dieselbe Query für häufige und für seltene Kunden benutzt und nur CUSTOMER\_ID als Wert der Bind-Variablen wird angepasst. Oracle wird dafür also einen einzigen Cursor generieren, der für alle verschiedenen CUSTOMER\_IDS verwendet wird. Falls wir also ein Histogramm erzeugen würden, bestände das Risiko, dass Oracle denselben Plan, der für den Großkunden verwendet wird (FULL TABLE SCAN auf ORDERS), auch für den seltenen Kunden anwendet. Hier wäre aber ein Index-basierter Access Path zu bevorzugen. Wir verwerfen deshalb die Idee mit dem Histogramm und überlegen uns eine Alternative.

### Idee 2: Erhöhung des Clustering durch Sortierung der ORDER-Daten nach CUSTOMER\_ID

Der Index-Zugriff von Zeile 6 liefert also 700.000 ROWIDs mit der gewünschten CUSTOMER\_ID. Die ROWID ist das physische Identifikationsmerkmal eines Datensatzes der Tabelle und besteht aus „File“, „Block“ und „Row in Block“. In Zeile 5 wird also jede ROWID aufgelöst und gelesen. Die Bestelldaten für die einzelnen Kunden sind unsortiert, die Rows für den Kunden „1001“ sind also quer über alle Tabellen-Blöcke verteilt. Wenn man nun die Tabelle „ORDERS“ reorganisieren und nach „CUSTOMER\_ID“ sortiert speichern würde, müsste man sehr wahrscheinlich weniger Tabellen-Datenblöcke lesen. Als Richtwert kann man die Anzahl der Rows (700.000) mit der Anzahl der Buffers (22297 mi-

nus 1467=20830) ins Verhältnis setzen: „700.000/20830“. Am effizientesten ist diese Reorganisation, wenn die Anzahl der Blöcke sehr hoch im Verhältnis zu der Rows ist. Ein Test zeigt, dass damit die Anzahl der Buffer Gets von 20830 auf 4862 reduziert werden kann (siehe Abbildung 5). Nachteil ist, dass die Reorganisation nicht nur einmalig, sondern regelmäßig durchgeführt werden muss. Es ist also in diesem Fall nicht die optimale Lösung. Zudem ist der Ressourcen-Verbrauch mit über 6000 Buffer Gets für eine einzelne zurückgelieferte Zeile noch deutlich zu hoch. Wir verwerfen deshalb auch diese Idee und testen eine dritte.

### Idee 3: Erhöhung der Selektivität durch Index auf ORDERS.ORDER\_STATUS

Es wird davon ausgegangen, dass die meisten Bestellungen der Tabelle „ORDERS“ im Status „COMPLETED“ sind und nur recht wenige im Status „PENDING“. Beim Nested Loop Join findet der Einstieg in die Tabelle „ORDERS“ über die Spalte „CUSTOMER\_ID“ statt. Es wird ein Index angelegt, der neben „CUSTOMER\_ID“ auch die Spalte „ORDER\_STATUS“ enthält. Da es nur zwei verschiedene Stati gibt und die Zeilen im Index sortiert gespeichert werden, kann Index Compression für die erste Spalte aktiviert werden und sich wiederholende Status-Strings können vermieden werden. Dies führt zu einer Reduzierung der Index-Leaf-Blocks (siehe Listing 5). Abbildung 6 zeigt den Status nach erneuter Ausführung der Query.

Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers
0	SELECT STATEMENT		1		1	00:00:00.01	7
1	SORT ORDER BY		1	16	1	00:00:00.01	7
2	NESTED LOOPS		1	16	1	00:00:00.01	7
3	TABLE ACCESS BY INDEX ROWID	CUSTOMERS	1	1	1	00:00:00.01	3
* 4	INDEX UNIQUE SCAN	CUSTOMERS_PK	1	1	1	00:00:00.01	2
5	TABLE ACCESS BY INDEX ROWID	ORDERS	1	16	1	00:00:00.01	4
* 6	INDEX RANGE SCAN	ORDERS_CUST_STATUS	1	16	1	00:00:00.01	3

Predicate Information (identified by operation id):

```

4 - access("C"."CUSTOMER_ID"=:V1)
6 - access("O"."ORDER_STATUS"='PENDING' AND "O"."CUSTOMER_ID"=:V1)

```

Abbildung 6: Ressourcen-Verbrauch und erweiterter Ausführungsplan nach Index-Erstellung

Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers
0	SELECT STATEMENT		1		1	00:00:00.01	8
1	NESTED LOOPS		1	8	1	00:00:00.01	8
2	TABLE ACCESS BY INDEX ROWID	CUSTOMERS	1	1	1	00:00:00.01	3
* 3	INDEX UNIQUE SCAN	CUSTOMERS_PK	1	1	1	00:00:00.01	2
4	TABLE ACCESS BY INDEX ROWID	ORDERS	1	8	1	00:00:00.01	5
* 5	INDEX RANGE SCAN	ORDERS_CUST_STATUS_DATE	1	8	1	00:00:00.01	4

Abbildung 7: Ressourcen-Verbrauch und erweiterter Ausführungsplan nach Index-Erstellung zur Vermeidung der Sort-Operation

Die Maßnahme war erfolgreich und wir konnten somit die Anzahl der Buffer Gets von 22.300 auf 7 reduzieren. Dieser Ausführungsplan ist nun sowohl für den Kunden mit 700.000 Bestellungen als auch für den Kunden mit wenigen Bestellungen optimal.

#### Idee 4: Vermeidung der Sortierung durch Erweiterung des Index

Im obigen Beispiel wird nur eine Zeile zurückgeliefert. Die Sortierung ist deshalb nicht aufwändig. Falls die Anzahl der Rows allerdings relativ hoch wäre, könnte man durch Erweiterung des Index die Sortierung eliminieren (siehe Listing 6). Abbildung 7 zeigt den Ausführungsplan, der die Vermeidung der Sort-Operation (SORT ORDER BY) bestätigt.

#### Hinweis 1: Optimizer Goal

Der Optimizer unterstützt verschiedene Optimierungen, abhängig davon, ob die Gesamtausführungszeit des Statements (ALL\_ROWS) oder die Antwortzeit, bis die ersten X Rows (FIRST\_ROWS\_n) geliefert werden, optimiert wird. Bei der

vorherigen Technik zur Vermeidung einer Sortierung hat man auch gleichzeitig den „FIRST\_ROWS\_n“-Zugriff optimiert. Die Resultate können direkt von der Row-Source-Operation „NESTED LOOPS“ zeilenweise an die Anwendung beziehungsweise den Benutzer zurückgeliefert werden. Er kann somit sehr schnell mit den ersten Ergebnissen arbeiten. Bei einer Query mit der Row-Source-Operation „SORT ORDER BY“ muss zuerst die gesamte Ergebnismenge ermittelt und sortiert werden,

```
create index DEMO.ORDERS_CUST_
STATUS on DEMO.ORDERS(ORDER_
STATUS,CUSTOMER_ID) COMPRESS 1;
```

Listing 5

```
drop index DEMO.ORDERS_CUST_
STATUS;
create index DEMO.ORDERS_
CUST_STATUS_DATE on DEMO.
ORDERS(ORDER_STATUS,CUSTOMER_
ID,ORDER_DATE)
COMPRESS 1;
```

Listing 6

bevor die erste Zeile an den Anwender zurückgeliefert werden kann.

#### Hinweis 2: Fixierung des Zugriffsplans - Plan Instability

Bei Problemen mit „Plan Instability“ kann es notwendig sein, den optimalen Zugriffsplan zu fixieren. Der Autor bevorzugt dafür die Methode, ein SQL-Profil mittels „DBMS\_SQLTUNE.IMPORT\_SQL\_PROFILE“ zu setzen. Dabei wird in der Tuning-Phase der Zugriffsplan durch Hints bewusst verändert und anschließend werden die Ausführungsplan-Direktiven (Outlines) dem produktiven Statement zugewiesen. Bei Verwendung der Standard Edition ist ein ähnliches Vorgehen mithilfe von Stored Outlines möglich.



Martin Decker  
martin.decker  
@ora-solutions.net

Wenn sich der Cost Based Optimizer (CBO) für einen neuen Ausführungsplan entscheidet, ist das nicht immer zum Guten. Grund genug für den DBA, sich über das 11g-Feature „SQL Plan Management“ zu informieren, mit dem sich Ausführungspläne stabilisieren lassen. Der Artikel zeigt dessen Praxistauglichkeit.

# SQL Plan Management unter der Lupe

Jan Krüger, TUI Infotec GmbH

Wechselt der CBO plötzlich von einem „table access by index rowid“ zu einem „full table scan direct path“, drückt der neue Plan ein leistungsfähiges IO-Subsystem auch mal an die Wand: Produktionsausfall ist die Folge. Bei Änderungen in den Statistiken, „Bind variable Peeking“ oder das 11g-Feature „Adaptive Cursor Sharing“ kann niemand vorhersagen, wann ein Plan kippt.

Beim Einsatz von SQL Plan Management (SPM) speichert Oracle ausgewählte Ausführungspläne in Form von Optimizer Hints in der SQL Manage-

ment Base im „SYSAUX Tablespace“ – sie werden „Plan Baselines“ genannt. Wird ein Statement geparkt, für das ein oder mehrere Pläne in der Management Base hinterlegt sind, kommen andere Pläne für das Statement nicht in Frage. Soweit so einfach. Doch wie funktioniert das alles im Detail? Um Statements in die SQL Management Base als Plan Baselines aufzunehmen, gibt es im Wesentlichen zwei Möglichkeiten:

- Automatisches Aufzeichnen von Statements mithilfe des Parame-

ters „OPTIMIZER\_CAPTURE\_SQL\_PLAN\_BASELINES“

- Übernahme von Ausführungsplänen aus dem Cursor-Cache im Speicher der Instanz

## Mit der Gießkanne

Setzt man den Parameter „OPTIMIZER\_CAPTURE\_SQL\_PLAN\_BASELINES=true“ im Systemkontext, werden im Folgenden alle Statements, die mehr als einmal ausgeführt werden, mit ihren Ausführungsplänen in der Plan Baseline verankert. Hinweis: Ab 11.2.0.3 gehört der Parameter zum Optimizer Environment, ein Umschalten erzeugt also einen Parse-Storm im System, weil die vorhandenen Cursors wegen Environment Mismatch invalidiert werden. Ob in einem Statement Bind-Variablen verwendet werden oder Literals, spielt für die Aufnahme in die Plan Baseline keine Rolle – zweimaliges Ausführen genügt. Es stehen hierbei keine Filterkriterien wie der User-Name zur Verfügung. Werden SQL-Statements von Überwachungstools wiederholt ausgeführt, sind sie in der Baseline zu finden. Wer sich an einem solchen Overhead nicht stört, kann das Automatic-Plan-Capture auch dauerhaft anschalten und damit alle zukünftig auftretenden neuen Statements automatisch erfassen lassen. Hinweis: In 11.2.0.2 existiert ein Performance-Bug, der im Zusammenhang mit dem „Automatic Capture“ auftritt. Der Parameter lässt sich auch im Session „Context“ setzen, siehe den Anwendungsfall weiter unten.

Wer bei der Befüllung der Plan Baseline etwas gezielter vorgehen möchte, kann mithilfe der Funktion „DBMS\_SPM.LOAD\_PLANS\_FROM\_CURSOR\_CACHE“ Pläne aus dem Cursor-Cache

```

set serveroutput on
variable n number;
declare
sql_rec v$sqlarea%rowtype;
i number := 0;
r number;
begin
for sql_rec in (select sql_id from v$sqlarea a
                where object_status='VALID'
                and parsing_schema_name IN ('SCHEMA')
                and parsing_user_id != 0
                and plan_hash_value != 0
                and exact_matching_signature != 0
                and executions > 99 and not exists
                (select 1 from dba_sql_plan_baselines b
                 where b.signature = a.exact_matching_signature))
loop
r := DBMS_SPM.LOAD_PLANS_FROM_CURSOR_CACHE(sql_id=>sql_rec.sql_id);
DBMS_LOCK.SLEEP(1);
if (r = 0) then
dbms_output.put_line('no plan fixed from sql_id ' || sql_rec.sql_id);
end if;
if (r > 1) then
dbms_output.put_line(r || ' plans fixed from sql_id ' || sql_rec.sql_id);
end if;
i := i + r;
end loop;
:n := i;
end;
/
print :n;

```

Listing 1



übernehmen. Die Funktion bietet einige vordefinierte Filter (nach „sqltext“, „parsing\_schema\_name“, „module“ sowie „action“) und kann auch mit der SQL\_ID aufgerufen werden. Sollen nur einzelne Statements, die bereits durch kippende Pläne unangenehm aufgefallen sind, stabilisiert werden, ist der „load“ aus dem Cursor-Cache ebenso die Methode der Wahl. Die „sqlplus“-Ladeprozedur für den „bulk load“ erlaubt die freie Definition von Kriterien für die Übernahme (siehe Listing 1).

Der Select von „v\$sqlarea“ legt fest, für welche SQL\_IDs die Pläne in die Baseline übernommen werden sollen. Die Filterkriterien sind nach Bedarf anzupassen. Zur Beschleunigung werden hier die SQL\_IDs ausgefiltert, die nicht für die Baseline in Frage kommen (etwa einfache INSERT-Statements ohne Plan), oder solche, die schon in der Baseline vorhanden sind. „SLEEP“ entzerzt den Parse-Storm, denn bei jedem neu in die Baseline aufgenommenen Plan wird der bisherige Cursor invalidiert, zu erkennen an „v\$sql\_shared\_cursor.stb\_object\_mismatch“, STB steht für „SQL Tuning Base“, wozu auch die SPM-Baselines gehören. Schließlich hat die Übernahme aus dem Cursor-Cache den Vorteil, dass die dafür nötige Rechenzeit nicht in der User-Session anfällt, die das Statement ausführt, dessen Plan gerade in die Baseline aufgenommen werden soll.

Der Vollständigkeit halber sei noch darauf verwiesen, dass sich Pläne auch aus SQL-Tuning-Sets (DBMS\_SPM.LOAD\_PLANS\_FROM\_SQLSETS) und bei Migration aus älteren Oracle-Versionen aus „Stored Outlines“ (DBMS\_SPM.MIGRATE\_STORED\_OUTLINES) in die Plan Management Baseline übernehmen lassen.

### An der Basis

Die View „dba\_sql\_plan\_baselines“ zeigt den Inhalt der Baseline an. „SIGNATURE“ repräsentiert den um Leerzeichen sowie Groß- und Kleinschreibung bereinigten SQL-Text; damit wird beim Parsen in der Baseline gesucht. „SQL\_HANDLE“ besteht aus der Zeichenfolge „SQL\_“ und der Signatur in Hexadezimal. „PLAN\_NAME“ besteht aus der Zeichenfolge „SQL\_PLAN\_“,

der Signatur in Base32-Kodierung (ab 11.2.0.3) und dem „plan hash value 2“ in Hexadezimal, den man an anderer Stelle auch unter der Bezeichnung „PLAN\_ID“ wiederfindet.

Zu einem SQL-Statement kann es in der SQL Management Base mehrere mögliche Ausführungspläne geben. Jeder Plan hat fünf „YES/NO“-Flags (ab 11.2.0.2), die sich mithilfe der Funktion „DBMS\_SPM.ALTER\_SQL\_PLAN\_BASELINE“ vom DBA umschalten lassen.

Wird ein Plan erstmalig in die Baseline aufgenommen, sind die Flags auf „ENABLED=YES“ und „ACCEPTED=YES“ gesetzt. Alle nachfolgenden Pläne für das gleiche SQL sind „ENABLED=YES“, aber zunächst „ACCEPTED=NO“. Sie müssen erst freigegeben werden, bevor sie für die Ausführung in Frage kommen (siehe weiter unten).

„ENABLED (YES/NO)“ gibt an, ob der Plan überhaupt für die Ausführung oder eine Freigabe zur Ausführung in Betracht kommt. Setzt der DBA mithilfe von „DBMS\_SPM.ALTER\_SQL\_PLAN\_BASELINE“ das „ENABLED“-Flag auf „NO“, wird der Plan nicht mehr benutzt und ist auch kein Kandidat mehr für eine Freigabe. Würde der Plan stattdessen aus der Baseline entfernt – was mit „DBMS\_SPM.DROP\_SQL\_PLAN\_BASELINE“ möglich ist –, käme er wahrscheinlich nach kurzer Zeit wieder als Kandidat („ENABLED=YES“, „ACCEPTED=NO“) in die Baseline. Die anderen Flags bekommen wir später.

Das beliebte Paket „DBMS\_XPLAN“ wurde dahingehend erweitert, dass die Funktion „display\_cursor“ in den Anmerkungen darauf hinweist, wenn eine SPM-Baseline verwendet wird. Mit „dbms\_xplan.display\_sql\_plan\_baseline“ kann man den Plan, der in einer Baseline steckt, anzeigen, mit der „ADVANCED“-Option auch die zugrunde liegenden Outlines (zu den Einschränkungen weiter unten).

### Der SQL-Text ist Dreh- und Angelpunkt

Ist die Plan Baseline erfolgreich gefüllt, das automatische Capture aus- und das Plan Management angeschaltet (default ist der Parameter „OPTIMIZER\_USE\_SQL\_PLAN\_BASELINES=true“), läuft für jedes zu parsende Statement etwas vereinfacht Folgendes ab:

1. Das Statement wird geparkt und der optimale Ausführungsplan wird bestimmt, genauso wie ohne SQL Plan Management.
2. Mithilfe der „exact\_matching\_signature“, die den SQL-Text, bereinigt um Formatierung etc., repräsentiert, wird in der SQL Management Base nachgeschaut, ob eine Baseline vorhanden ist.
3. Existiert eine Baseline, wird geprüft, ob der soeben neu ermittelte Plan enthalten ist und benutzt werden darf („ENABLED=YES“, „ACCEPTED=YES“). Wenn ja, wird der Plan genutzt, ansonsten wird er zur Baseline hinzugefügt („ENABLED=YES“, „ACCEPTED=NO“) und das Statement wird mithilfe der in der Baseline hinterlegten Pläne neu geparkt, um daraus den besten Ausführungsplan zu bestimmen.

Der genaue Ablauf unter Einbeziehung des „FIXED“-Flags ist im Flussdiagramm dargestellt. „FIXED=YES“ gibt den Plänen Priorität über die „FIXED=NO“-Pläne und verhindert außerdem, dass neue Plan-Kandidaten aufgenommen werden. Die Auswahl der Pläne über die „exact\_matching\_signature“ – also nur den SQL-Text – bedeutet, dass gleiche SQLs, die in verschiedenen Usern und damit verschiedenen Schemata ablaufen, die gleichen Pläne bekommen. Ob dies ein Bug oder Feature ist, muss jeder selbst entscheiden. In einer mandantenfähigen Applikation, in der für jeden Mandanten ein Schema mit den gleichen Objekten vorgesehen ist, kann dies zu Problemen führen, wenn die Datenmengen und Verteilungen unterschiedlich sind. Bei Userzugriff auf die Datenbank ist es sogar möglich, dass ein Benutzer dem anderen schlechte Ausführungspläne unterschiebt, indem er in seinen Schema-Baselines SQL-Statements erzeugt, die auch in anderen Schemata benutzt werden. Für viele Anwendungsfälle dürfte diese Einschränkung zwar keine Rolle spielen, ganz zu Ende gedacht ist das aber noch nicht.

### Adaptive Cursor Sharing und Plan Management

In der Datenbank-Version 11g wurde die Ausführungsoptimierung für SQL-

```
SQL_ID 4ksb3mrktvqp, child number 0
```

```
select count(iata_cd) from airport where country=:c
Plan hash value: 765729859
```

Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers	Reads
0	SELECT STATEMENT		1		1	00:00:00.02	59	59
1	SORT AGGREGATE		1	1	1	00:00:00.02	59	59
2	TABLE ACCESS BY INDEX ROWID	AIRPORT	1	1323	1323	00:00:00.02	59	59
* 3	INDEX RANGE SCAN	COUNTRY_IDX	1	1323	1323	00:00:00.01	6	6

```
Predicate Information (identified by operation id):
```

```
3 - access("COUNTRY"=:C)
```

```
Note
```

```
-----
```

```
- SQL plan baseline SQL_PLAN_7p565u74hm99b45fcffbd used for this statement
```

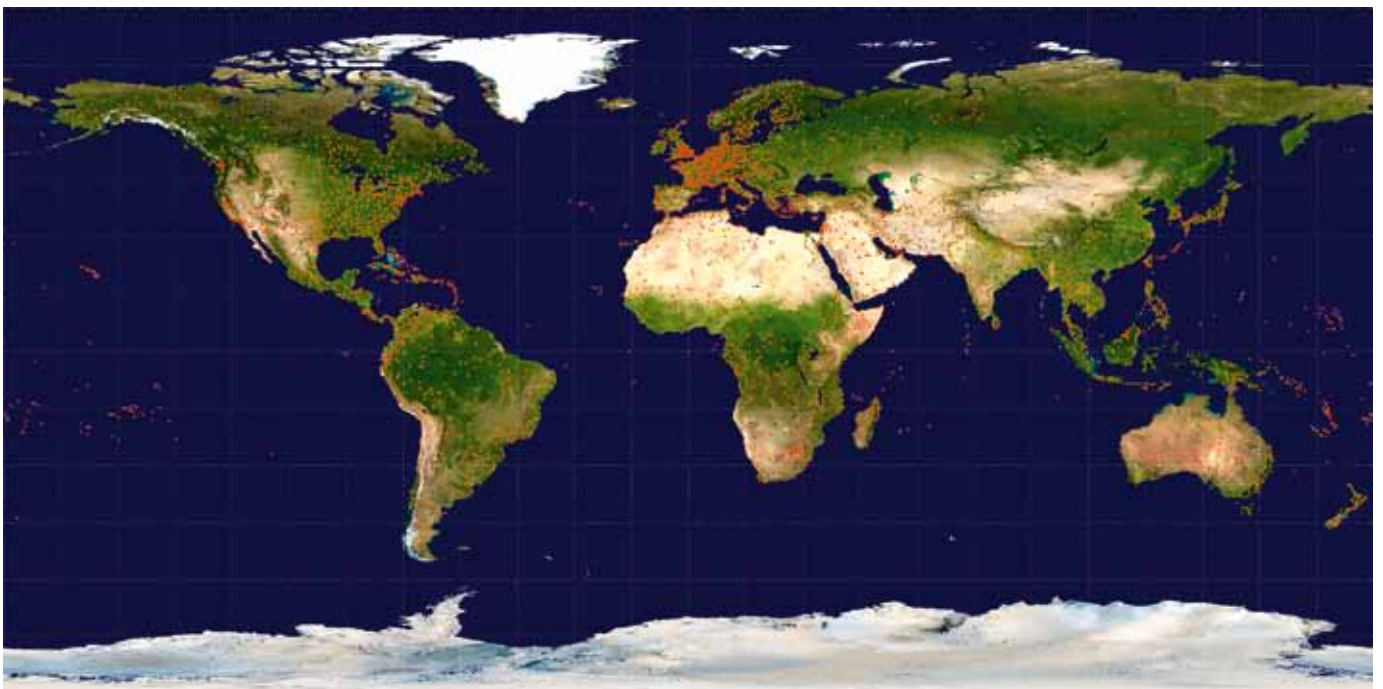
*Listing 2: Ausgabe von „dbms\_xplan.display\_cursor“ mit Hinweis auf die verwendete Plan Baseline im Notes-Abschnitt*

Statements mit Bind Variablen weiterentwickelt. Man stelle sich eine Tabelle mit den Flughäfen der Welt vor, neben dem Namen des Flughafens und der Stadt ist auch das Land enthalten, in dem der Flughafen liegt; auf der Länder-Spalte wurde ein Index angelegt (siehe Abbildung 1). Sollen nun beispielsweise die Flughäfen in den USA selektiert werden, ist ein Full Table Scan unter Umständen effektiver als

ein Zugriff über den Index, weil es dort so viele Flughäfen gibt, dass die meisten Blöcke der Tabelle ohnehin inspiziert werden müssen. Selektiert man die Flughäfen eines Landes wie Gambia mit nur einem Flughafen, empfiehlt sich der Zugriff über den Index (siehe Listing 2 und Listing unter [www.doag.org/go/doagnews/krueger\\_listing](http://www.doag.org/go/doagnews/krueger_listing)).

Wohingegen in der Version 10g nur beim erstmaligen Parsen der Wert der

Bind-Variablen herangezogen wurde (bind variable peeking), überwacht 11g die Variablenwerte und Ausführungspläne dauerhaft mit dem Adaptive Cursor Sharing (ACS). Wie schon in 10g sagt der CBO mithilfe von Histogrammen die Selektivität auf Basis des Bind-Variablen-Wertes voraus. Neu ist, dass durch Cursor-Children zu einem SQL-Statement mehrere Pläne im Cursor-Cache vorgehalten werden,



*Abbildung 1: Beispiel-Datensatz „Flughäfen der Welt“. Der Banjul International Airport von Gambia liegt an der Küste von Westafrika. Gambia ist ein kleines Land, eingeschlossen vom Senegal (Karte und Daten: [openflights.org](http://openflights.org))*

die dann je nach Variablenwerten zur Ausführung angewählt werden. ACS ist hier nicht das Thema, deswegen sei der Ablauf nur kurz erklärt: Kommt ein SQL für ACS in Frage, wird dies in einem Flag vermerkt (siehe „v\$sql.is\_bind\_sensitive“). Die folgenden Ausführungen werden hinsichtlich Größe der Ergebnismenge, CPU-Verbrauch und Consistent Reads stichprobenhaft überwacht („v\$sql\_cs\_histogram“ und „v\$sql\_cs\_statistics“). Mithilfe dieser Historie wird entschieden, dass das SQL von verschiedenen Plänen profitieren könnte. Sodann wird neu geparkt und es entsteht ein „bind aware cursor“ („v\$sql.is\_bind\_aware“). Für diesen werden die Bind-Variablen-Werte-Bereiche und Selektivitäts-Prognosen dokumentiert („v\$sql\_cs\_selectivity“).

Liegt der Wertebereich einer neuen Ausführung außerhalb der bereits bekannten Bereiche, wird neu geparkt. Kommt dabei ein Plan zustande, der bereits in einem anderen Child-Cursor vorliegt, werden die Cursor- und Werte-Bereiche zusammengefasst. Auf diese Weise entsteht nach und nach ein eingeschwungener Zustand mit einer begrenzten Zahl von „überlebenden“ Child-Cursors und es sind schließlich keine weiteren Parses mehr notwendig.

Allerdings hängt der tatsächliche Ablauf im ACS von der Reihenfolge des Auftretens verschiedener Bind-Variablen-Werte ab. In der Einschwingphase werden die Ausführungspläne nicht optimal angewandt, schlimmstenfalls geht es zu wie beim „bind variable peeking“ in älteren Versionen. Es ist damit letztlich nicht vorhersagbar, ob und wann ein Cursor „bind aware“ wird und wann ein eingeschwungener Zustand erreicht ist, bei dem die Pläne optimal genutzt werden. Zum genannten Beispiel: Es können viele Full Table Scans für Gambia ausgeführt werden, bevor der Bedarf für einen angepassten Plan erkannt wird. Übrigens, der „BIND\_AWARE“-Hint hilft dem CBO bei Bedarf auf die Sprünge.

Funktioniert Plan Management also mit „Adaptive Cursor Sharing“? Im Prinzip ja, denn es kommt erst ins Spiel, nachdem der normale CBO-Lauf einschließlich der Optimierungen zur „bind awareness“ ausgeführt wurde. Er

wirkt erst im Nachhinein als Filter und Korrektiv auf den Ergebnissen des CBO.

Hilft das SQL Plan Management bei den Problemen des „Adaptive Cursor Sharing“? Leider nein, denn es tritt erst in Erscheinung, nachdem der normale CBO-Lauf einschließlich der Optimierungen zur „bind awareness“ ausgeführt wurde. Ob ein Cursor „bind aware“ ist, entscheidet nicht das SPM (Stand 11.2.0.3). Die Baseline beinhaltet zwar idealerweise alle Pläne, die beim ACS herauskommen können und abhängig von den Variablen-Werten gute Performance bieten. Wann und ob der CBO den „cursor bind aware“ macht, beeinflusst das SPM nicht, auch speichert es nicht die Informationen zu Bind-Variablen-Werten und zur Selektivität. Diese müssen beispielsweise nach einem Restart jedes Mal neu erarbeitet werden. Insofern hat die Stabilisierung der Pläne durch SPM Grenzen: Was im ACS schiefgehen kann, bis ein eingeschwungener Zustand erreicht ist, geht auch mit SPM schief. Der Full Table Scan für Gambia bleibt uns also erhalten.

Die Zusammenarbeit zwischen SPM und ACS hält überdies noch eine unangenehme Überraschung bereit: Setzt man den automatischen Plan Capture per Parameter ein, um eine Baseline zu erzeugen, wird nur der erste Plan mit „ACCEPTED=YES“ gespeichert. Alle weiteren Pläne, die dem ACS ent-

springen, werden als Kandidaten mit „ACCEPTED=NO“ in der Baseline gespeichert und können erst mal nicht genutzt werden. Der Full Table Scan für Gambia, der ohne SPM irgendwann verschwinden würde, bleibt also dauerhaft erhalten, wenn man nicht eingreift. Wer glaubt, mit dem Setzen des Parameters sei alles getan, könnte hier ein böses Erwachen erleben.

### Der Lauf der Welt

Applikationen – auch solche, die Datenbanken nutzen – entwickeln sich für gewöhnlich weiter. Im Plan Management kann einiges passieren, auf das reagiert werden muss:

- Die Datenbestände ändern sich und ein neuer Ausführungsplan wird (vermeintlich) effektiver als der im SPM festgehaltene
- Neue SQL-Statements kommen hinzu, für die keine Baseline existiert
- Alte SQL-Statements werden nicht mehr genutzt
- Ein Plan in der Baseline kann nicht mehr genutzt werden, weil sich im Datenmodell etwas geändert hat – zum Beispiel eine zusätzliche Spalte im Index
- Ein neuer Plan wäre besser, weil sich im Datenmodell etwas geändert hat

Werden Pläne in der Baseline länger nicht benutzt, kommt ein Auf-

```
select count(*) kandidaten
from dba_sql_plan_baselines
where enabled='YES' and ACCEPTED='NO'
```

Listing 3

```
select count(distinct exact_matching_signature) neues_sql
from v$sqlarea a
  where object_status='VALID'
     and parsing_schema_name IN (,SCHEMA')
     and parsing_user_id != 0
     and plan_hash_value != 0
     and exact_matching_signature != 0
     and executions > 99 and not exists
     (select 1 from dba_sql_plan_baselines b
      where b.signature = a.exact_matching_signature
      and b.reproduced='YES')
```

Listing 4

räumjob und löscht sie. „DBMS\_SPM.CONFIGURE“ ist bei der Einstellung der Aufbewahrungsdauer behilflich. Neue Pläne für in der Baseline vorhandene SQL-Statements fügt Oracle automatisch hinzu („ENABLED=YES“, „ACCEPTED=NO“) und damit ist die Überwachung einfach (siehe Listing 3).

Neue SQL-Statements (ohne Pläne in der Baseline) sind da schon etwas kniffliger. Schaltet man das Auto-Capture dauerhaft an, stellt sich die Frage nicht. Geht man selektiver vor, empfiehlt sich im Monitoring ein Abgleich mit „v\$sqlarea“ mit den gleichen Selektions-Kriterien, die auch für die Befüllung der Baseline genutzt werden sollen (siehe Listing 4).

Bleiben noch die Pläne, die nicht mehr funktionieren. Ab 11.2.0.2 gibt das Flag „dba\_sql\_plan\_baselines.reproduced“ darüber Auskunft, ob es beim letzten Versuch geklappt hat, den Plan zu nutzen. Das SQL zur Überwachung ist dem Leser überlassen. Am Rande sei bemerkt, dass ein SQL-Statement, dessen Plan in der Baseline in einem Schema funktioniert und im anderen nicht, weil beispielsweise ein Index nur in einem Schema vorhanden ist, hier zu munterem Flag-Blinken führen kann.

Übrigens: Einen Plan, der wegen Änderungen im Datenmodell nicht mehr reproduziert werden kann, zeigt „dbms\_xplan.display\_sql\_plan\_baseline“ auch nicht mehr an. Stattdessen wird ein anderer Plan gemeldet, der alternativ zur Baseline zum Einsatz kommt. Ob das ein Bug oder Feature ist, mag jeder selbst bewerten, zumindest einen Hinweis in der „dbms\_xplan“-Ausgabe wäre nützlich. Ebenso ist es nicht mehr möglich, mit „dbms\_xplan“ Baselines anzuzeigen, wenn das als „CREATOR“ in der Baseline hinterlegte Schema nicht (mehr) vorhanden ist. Genutzt werden kann die Baseline trotzdem. Das ist keine Überraschung, denn die Baseline enthält ja nicht den tatsächlichen Plan, sondern Hints, um ihn herzustellen. Für den „explain“ benötigt „dbms\_xplan“ ein Schema mit entsprechenden Objekten, sodass der SQL-Text zusammen mit den Hints auch einen Plan ergibt. Dennoch ist hier noch Potenzial für Verbesserungen.

```
variable n number
exec :n :=DBMS_SPM.ALTER_SQL_PLAN_BASELINE(NULL, 'SQL_PLAN_ccb833bb-deca1852', 'enabled', 'NO');
```

Listing 5

```
exec :n :=DBMS_SPM.ALTER_SQL_PLAN_BASELINE(NULL, ,SQL_PLAN_ccb833bb-deca1852', ,autopurge', ,NO');
```

Listing 6

### Darwinismus unter Plänen

Hat unser Monitoring festgestellt, dass es neue Pläne gibt, die besser sein sollen als die alten, ist der DBA gefragt. Was soll er tun? Alle neuen Pläne mit „dbms\_xplan“ ausgeben, begutachten und irgendwie erfüllen, ob sie besser sind als die alten, kommt nicht in Frage. „DBMS\_SPM.EVOLVE\_SQL\_PLAN\_BASELINE“ ist der versprochene Heilsbringer, ein „verify=>YES“ bittet die Funktion, vor der Freigabe eines Plans („ACCEPTED=YES“) doch mal nachzusehen, ob er wirklich besser ist als der alte. Doch leider funktioniert nicht, was nicht funktionieren kann. Mit welcher Magie der Verify vorgeht, ist nicht vollständig dokumentiert; es sieht danach aus, als würden die Statements mit den Plänen und dem zu ihrer Entstehung gehörenden Satz Bind-Variablen-Werte tatsächlich ausgeführt und vermessen. Nur leider dauert ein Zugriff auf die Flughäfen der USA eben länger als auf den von Gambia, auch wenn der Full Table Scan schneller wäre als der Indexzugriff. Hat Gambia die Baseline gelegt, können die USA lange darauf warten, dass die Plan-Evolution mit „verify=>YES“ ihnen weiterhilft.

Will der DBA die Unschärfen des Verify im „EVOLVE“ vermeiden, bleibt also nur das „verify=NO“, was alle Pläne mit „ENBLE=YES“ in den „ACCEPTED=YES“-Status befördert. Ein Trost bleibt: Wann Pläne freigegeben werden, kann man planen und im Rahmen von Change-Prozessen steuern, sodass der mögliche Schaden geringer ist, als wenn die Pläne zur Unzeit gekippt wären. Ein zweiter Trost: Stellt sich ein neuer Plan nach der Freigabe als schädlich heraus, lässt er sich

schnell und einfach abschalten, um zum alten Plan zurückzukehren:

1. Auf herkömmlichem Weg das SQL-Statement mit dem schlechten Plan ausfindig machen
2. Aus der Spalte „v\$sql.sql\_plan\_baseline“ den Namen des schlechten Plans entnehmen, hier als Beispiel „SQL\_PLAN\_ccb833bbdeca1852“
3. Plan abschalten (siehe Listing 5)
4. Automatisches Löschen ausschalten (siehe Listing 6)

Im Zuge dessen werden automatisch die Cursors mit dem schlechten Plan invalidiert und es wird mit dem bisherigen Plan aus der Baseline, der ja immer noch da ist, weitergearbeitet. Alternativ kann man die anderen Pläne für das SQL auch auf „FIXED=YES“ umschalten. Verzichtet man auf die Fixierung des Plans und bleibt der „AUTOPURGE=YES“, wird der abgeschaltete Plan nach der Aufbewahrungszeit gelöscht (er wird ja nicht mehr benutzt), es entsteht ein neuer Kandidat, der irgendwann „evolved“ wird, und dann geht das Problem von vorne los.

### Gut verpackt auf Reisen

Sicherlich erscheint es für Applikationshersteller reizvoll, eine Baseline mit der Applikation mitzuliefern, um Probleme mit Ausführungsplänen von vornherein auszuschließen. Eine andere Idee ist, Baselines, die sich in der Qualitätssicherung behauptet haben, in ein Produktionssystem zu übernehmen. Bei solchen Vorhaben helfen die Funktionen „DBMS\_SPM.PACK\_STGTAB\_BASELINE“ und „DBMS\_SPM.UNPACK\_STGTAB\_BASELINE“, die die

```
variable n number
exec :n := DBMS_SPM.LOAD_PLANS_FROM_CURSOR_CACHE(sql_
id=>'7vw4wyf41cd3a', sql_handle=>'SQL_7a94c5d1c909a52b');
```

### Listing 7

```
exec :n := DBMS_SPM.LOAD_PLANS_FROM_CURSOR_CACHE(sql_
id=>'7vw4wyf41cd3a', sql_text=>'select count(*) from airport where
country=:c');
```

### Listing 8

Baseline in eine Staging-Tabelle übertragen oder aus einer solchen in die SQL Management Base aufnehmen. Die Funktionen können bei Bedarf nach verschiedenen Kriterien filtern. Die Staging-Tabelle, die sich mit „export“, „import“ oder „datapump“ von einem System ins andere transferieren lässt, erzeugt man zuvor mit „DBMS\_SPM.CREATE\_STGTAB\_BASELINE“. „UNPACK“ überschreibt zum Beispiel die Flags vorhandener Pläne. Will man also im Rahmen eines neuen Software-Release Pläne hinzufügen, kann es sinnvoll sein, zunächst die vorhandene Baseline mit der zu importierenden zu vergleichen und das Delta für den „UNPACK“ bereitzustellen.

#### Dem CBO auf die Sprünge helfen

Außer zur Stabilisierung kann das SPM auch genutzt werden, um Ausführungspläne zu ändern. Wenn das SQL in einer Applikation verborgen und nicht mit Hints versehen werden kann, ist dies eine elegante Möglichkeit neben dem SQL-Patch und der Manipula-

tion von Statistiken. Zunächst benötigt man den Original-SQL-Text, der zum Beispiel aus „dba\_sql\_plan\_baselines“ oder aus „v\$sql“ extrahiert werden kann. Am einfachsten ist für den DBA, wenn es gelingt, durch Änderungen des „Optimizer Environment“ im „session context“ (beispielsweise über „alter session set optimizer\_index\_cost\_adj=1“) den gewünschten Ausführungsplan zu erhalten. Indem man „optimizer\_capture\_sql\_plan\_baseline“ in der Session auf „true“ setzt und das Statement zweimal ausführt, fügt man die gewünschte Baseline hinzu. Nun muss nur noch der gewünschte Plan auf „EVOLVE“ gesetzt werden (verify=>'NO') und der unerwünschte auf „ENABLED=NO“.

Etwas komplizierter ist der Vorgang, wenn man einen Hint im SQL-Text anbringen muss, um den gewünschten Plan zu bekommen. Der SQL-Text hat dann eine andere Signatur. Idealerweise ist das betreffende SQL-Statement für den Hint mit seinem schlechten Plan in der Baseline bereits enthalten. Aus „dba\_sql\_plan\_baseline“ bestimmt man den

„SQL\_HANDLE“. Anschließend führt man das SQL-Statement mit dem Hint aus und bestimmt aus „v\$sql“ die SQL\_ID sowie – falls nicht eindeutig – den „plan\_hash\_value“. Schließlich wird der neue Plan aus dem Cursor-Cache zur Baseline hinzugefügt, wobei man das Zielstatement über „SQL\_HANDLE“ referenziert (siehe Listing 7).

Steht „SQL\_HANDLE“ nicht zur Verfügung, weil das Statement noch nicht in der Baseline enthalten ist, kann auch der SQL-Text an die Funktion übergeben werden; „SQL\_HANDLE“ wird dann aus dem Text errechnet (siehe Listing 8).

#### Was das Ganze kostet

Das Feature ist Bestandteil der Oracle-Enterprise-Edition-Lizenz und es sind keine zusätzlichen Optionen wie etwa Tuning Pack erforderlich, um es zu nutzen. Trotz der noch nicht ganz ausgestandenen Kinderkrankheiten sollte jeder DBA die Funktionalität für seinen Troubleshooting-Werkzeugkasten kennen und bei Bedarf zumindest für einzelne Statements einsetzen können.

Jan Krüger  
jan.krueger@tui-infotec.com



## Wir begrüßen unsere neuen Mitglieder

### Persönliche Mitglieder

Thomas Ritter	Rene Renk
Sybille Kopp	Waldema Schott
Viktor Schmidt	Andres Campo-Penuela
Klaus Speierl	Jörn Kunnert
Sascha Oberhollenzer	Kirill Babeyev
Oliver Mang	Katrin Marchewka
Matthias Schulz	Gerold Geiß
Tammo Meyer	Marian Schmöcker
Matthias Deuß	Andreas Hild
Armin Eberle	Rainer Horst

Silvio Nifkiffa
Andreas Oberacher
Rainer Ackermann
Kersten Bühnert
Norman Stöcker
Oliver Horlacher
Ralph Baumbach
Manuel Swiercz
Muthiah Chidambaram

### Firmenmitglieder

Jan Ross, PPI Aktiengesellschaft
Susanne Herbst, Baur Versand GmbH & Co. KG
Dr. Renko Ungruhe, items GmbH
Ralph Baumbach, avato consulting ag
Marco Trujka, WERTGARANTIE Management GmbH
Christian Bretting, Georg-Simon-Ohm Hochschule



*Christian Trieb*  
Leiter Datenbank Community

## Neues aus der Datenbank Community

Ein gutes und erfolgreiches Jahr liegt hinter uns, das erste nach der Umstrukturierung innerhalb der DOAG. Das Community-Team hat sich gebildet und arbeitet gut zusammen. Die Datenbank-Webinare wurde initiiert und sehr gut angenommen. Auch die Berliner Expertenseminare finden im Datenbank-Bereich einen sehr guten Zuspruch.

Neu in diesem Jahr sind auch die Datenbank-Lounge und der RAC-Attack-Workshop im Rahmen der DOAG 2012 Konferenz + Ausstellung. Alle diese sowie auch die bewährten SIG-Veranstaltungen werden im Jahre 2013 fortgesetzt.

Neu hinzu kommt die Fachkonferenz DOAG 2013 Datenbank am Dienstag, 14. Mai 2013, in Düsseldorf. Hier werden an einem Tag intensive, kompakte Informationen über die Oracle-Datenbank sowie die MySQL-Datenbank präsentiert. Es wird vier parallele Streams (Datenbank 12c, Security, Performance/Tuning & Monitoring, Administration/Migration) und eine begleitende Ausstellung geben. Bitte reichen Sie dazu Ihren Vortrag ein. Näheres unter <http://datenbank.doag.org>.

Ich hoffe, Sie auf einer der Veranstaltungen der Datenbank Community begrüßen zu dürfen.

Für Fragen und Anregungen stehen wir unter [dbc@doag.org](mailto:dbc@doag.org) gerne zur Verfügung.

## Erfolgreiche Veranstaltung der SIG Datenbank

Am Mittwoch, 17. Oktober 2012, fand in Köln die SIG Datenbank zum Thema „Standard Edition“ statt. Nach der Begrüßung durch den SIG-Leiter und Oracle ACE Director Christian Trieb (Paragon Data GmbH) stellte Matthias Weiss von Oracle Deutschland das Thema „Standard Edition – Vorteile und Grenzen“ im Detail vor. Dabei ging er besonders auf die Punkte „Einsatzbereiche und Vorteile“, „Limitierungen“, „Optimierungsmöglichkeiten“, „Leistung“ und „Kosten“ ein. Yann Neuhaus und Jerome Witt (dbi services sa) präsentierten in ihrem Vortrag „Dbvisit Standby: Hochverfügbarkeit mit Oracle Standard Edition“, wie man eine Hochverfügbarkeitslösung auch in diesem Umfeld erreichen kann.

Das Thema „Migration“ stand im Mittelpunkt der Präsentation „Migration auf die Standard Edition und Unicode“ von Johannes Ahrends, ebenfalls Leiter der SIG Datenbank, Oracle ACE und Geschäftsführer der CarajandB GmbH. Daran schloss Peter Bekiesch (Herrmann & Lenz Services GmbH) mit „Ganzheitliches Monitoring mit dem Monitoring-Module – von der Express Edition über Standard Edition bis zur Enterprise Edition“ an. Er stellte ausführlich die Möglichkeiten in diesem Umfeld vor.

Zum Abschluss des Tages wurde von Martin Schmitter (Trivadis GmbH) in seiner Präsentation „Oracle Fail Safe ein Review, oder muss es denn immer RAC sein?“ die Alternative zu einer RAC-Installation im Windows-Umfeld vorgestellt. Die angeregten Diskussionen an diesem Tag zeigen deutlich, dass, in Abhängigkeit von den Anforderungen, die Standard Edition eine kostengünstige Alternative zum Einsatz einer Enterprise Edition sein kann.

Die nächste SIG Datenbank findet am Donnerstag, 28. Februar 2013, in München statt, diesmal unter dem Motto „Tuning oder Optimierung“.

Weitere Informationen dazu finden Sie unter <http://www.doag.org/termine/termine.php?tid=450850>.



*Dr. Dietmar Neugebauer*  
Vorstandsvorsitzender der DOAG

## DOAG 2012: Jubiläumskonferenz und 6.000 Mitglieder

Am 29. Juni 1988 treffen sich sieben Anwender bei einem Stuttgarter Notar zur offiziellen Gründung der DOAG, die dann am 11. August in das Vereinsregister des Amtsgerichts Stuttgart eingetragen wird. Zur ersten Vorsitzenden des Vorstands wird Eva Kraut gewählt. Bereits am 8. September 1988 treffen sich rund 300 Teilnehmer zur 1. Deutschen ORACLE-Anwenderkonferenz in einem Münchner Hotel. Im Mittelpunkt der zweitägigen Veranstaltung steht der Erfahrungsaustausch unter den Anwendern der damaligen Oracle-Produkte. So kommt es, dass die DOAG bereits 2012 die 25. DOAG-Anwenderkonferenz veranstaltet. Bestens zu diesem Jubiläum, das im Rahmen der DOAG 2012 Konferenz + Ausstellung gebührend gefeiert wird, passt die neue Rekordmarke der Mitgliederzahl.

Tatsächlich hat in der Nacht vom 10. auf den 11. Oktober 2012 die Zahl der DOAG-Mitglieder die 6.000er-Marke überschritten. Darunter sind knapp 5.000 Firmen-Mitgliedschaften, etwa 1.000 persönliche Mitglieder sowie 44 Studierende und 6 Ehrenmitglieder. Das enorme Mitgliederwachstum der letzten Jahre zeigt, dass die DOAG nach wie vor auf dem richtigen Kurs ist und die Interessen der Anwender erfolgreich vertritt.



Franz Hüll  
Leiter SIG Security

## Bericht von der SIG Security

Am 16. Oktober 2012 traf sich die SIG Security zur zweiten Veranstaltung in diesem Jahr in Köln. Zusammen mit der SIG Database, die am darauf folgenden Tag stattfand, konnten die Teilnehmer einen der beiden Tage buchen oder gleich beide. Die Veranstaltung war sehr gut besucht und nachdem noch ein paar Stühle organisiert wurden, fand auch jeder seinen Platz.

Die Themenliste enthielt einen Mix aus aktuellen Themen; über den Tag verteilt standen fünf verschiedene Vorträge auf dem Programm. Den Beginn machte Christian Pohl von Secaron mit „Pentesting – Bericht aus der Praxis“. Er berichtete von seinen Erfahrungen aus vielen Projekten, in denen es um Penetration-Testing von Web-Applikationen und den dahinter liegenden Datenbanken ging. Der nächste Vortrag befasste sich mit „Oracle Privileged Account Manager – OPAM“. Referent war Georg Völl von Oracle, der den Teilnehmern dieses Thema anschaulich näherbrachte.

Nach einem ausgezeichneten Mittagessen stand ein scheinbar trockenes Thema auf der Agenda, die „IT-Sicherheit auf der Basis des BSI Grundschutz“. Doch der Referent, Prof. Dr. Dirk Loomans von Loomans & Matz, verstand es ausgezeichnet, daraus einen spannenden und informativen Vortrag zu machen.

Alexander Kornbrust von Red Database Security hatte sich einen interes-

santen Titel für seinen Vortrag ausgedacht: „Sanftes Härten oder Wie kann ich meine Datenbank absichern, ohne die Produktion zu gefährden?“ Im Vortrag wurde schnell klar, was er damit meinte.

Der letzte Vortrag des Tages drehte sich um einen Dauerbrenner, der nach wie vor ganz oben steht, wenn es um IT-Sicherheit geht. In „Secure Scripting – Wohin nur mit dem Passwort?“ demonstrierte Gunther Pippèr von GPI Consult, wie der Umgang mit Passwörtern in Skripten gestaltet werden kann, damit hier möglichst kein Sicherheitsrisiko entsteht.

Im Jahr 2013 werden wir wieder mehrere Veranstaltungen planen. Informationen dazu stehen auf den Webseiten der DOAG. Anregungen und Themenwünsche, auch Vortragsangebote, bitte an sig-security@doag.org.



Dr. Frank Schönthaler  
Leiter der Business Solutions Community

## Neues aus der Business Solutions Community

Die Leitungsmitglieder der Business Solutions Community (BSC) trafen sich am 26. September 2012 zu ihrer turnusmäßigen Besprechung in Köln. Im Mittelpunkt der Diskussionen standen Strategien zur Erreichung einer passgenauen Ansprache der verschiedenen Zielgruppen der Business Solutions Community. Vereinbart wurden eine Intensivierung der Community-übergreifenden Zusammenarbeit und eine engere Kooperation mit der internationalen Oracle Applications Users

Group (OAUG) auf der DOAG 2013 Applications.

Im Rahmen der Veranstaltungsreihen der BSC sind interessante Community Days geplant oder bereits erfolgreich durchgeführt worden:

- Am 16. Oktober 2012 traf sich die E-Business Suite Community in Düsseldorf zu ihrem Community Day. Es war eine erfolgreiche Anwenderveranstaltung mit spannenden Anwenderberichten rund um das Thema „Handlungsoptionen für E-Business-Suite-Anwender in bewegten Zeiten“, gespickt mit regen Pausen-Diskussionen von Anwender zu Anwender.
- Oracle Primavera-Anwender und -Interessenten erfahren beim fünften Primavera Community Day am 14. November 2012 im Kongresszentrum darmstadtium in Darmstadt Wissenswertes über PMI und Primavera sowie über alternative Schnittstellen-Konzepte zur Integration von Primavera mit SAP. Probleme aus der täglichen Arbeit können im direkten Gespräch mit Trevor Grant von Oracle Primavera Software Support geklärt werden.

In ihren zukünftigen Veranstaltungen bietet die E-Business Community den Oracle-Anwendern und Interessenten nach wie vor die Möglichkeit, mehr Informationen und Erfahrungen über Oracle-Produkte zu sammeln: Am Dienstag, 26. Februar 2013, ist der nächste EBS Community Day geplant, der zum ersten Mal gemeinsam mit der Hyperion Community organisiert wird. Neben produktfokussierten Referaten werden Business-Themen aufgegriffen, die für Fach- und Führungskräfte aus dem Finanz- und Controlling-Umfeld von aktuellem Interesse sind.

### Vorschau auf die nächste Ausgabe

Das Schwerpunktthema der Ausgabe 01/2013 lautet

Mobile Computing

Sie erscheint am 1. Februar 2013

Dezember

14.12.2012  
**Webinar**  
 DOAG Geschäftsstelle  
 office@doag.org

Februar

05.02.2013  
**Regionaltreffen Hamburg/Nord**  
 Stefan Thielebein  
 regio-hamburg@doag.org

Januar

10.01.2013  
**Regionaltreffen Stuttgart**  
 Jens-Uwe Petersen  
 regio-stuttgart@doag.org

05.02.2013  
**Regionaltreffen Rhein-Neckar**  
 Frank Stöcker  
 regio-rhein-neckar@doag.org

06.02.2013  
**Regionaltreffen Rhein-Main**  
 Thomas Tretter, Kathleen Hock  
 regio-rhein-main@doag.org

22.01.2013  
**Regionaltreffen NRW**  
 Stefan Kinnen, Andreas Stephan  
 regio-nrw@doag.org

12.02.2013  
**Regionaltreffen Bremen**  
 Ralf Kölling  
 regio-bremen@doag.org

24.01.2013  
**Regionaltreffen Karlsruhe**  
 Reiner Bünger  
 regio-karlsruhe@doag.org

19./20.02.2013  
**Berlin Expertenseminar**  
 „Oracle Statspack“ mit Norbert Debes  
 Cornel Albert  
 expertenseminare@doag.org

29.01.2013  
**Regionaltreffen Freiburg**  
 Volker Deringer  
 regio-freiburg@doag.org

21.02.2013  
**SIG Development/Tools**  
 Andreas Badelt, Christian Schwitalla  
 sig-development@doag.org

29.01.2013  
**Regionaltreffen München/Südbayern**  
 Andreas Ströbel, Franz Hüll  
 regio-muenchen@doag.org

Aktuelle Termine und  
 weitere Informationen finden Sie unter  
[www.doag.org/termine/calendar.php](http://www.doag.org/termine/calendar.php)



## Berliner Expertenseminare

- Wissensvertiefung für Oracle-Anwender
- Mit ausgewählten Schulungspartnern
- Von Experten für Experten
- Umfangreiches Seminarangebot

**05./06. Februar 2013**  
 „APEX for experts“  
 Niels de Bruijn, MT AG

**19./20. Februar 2013**  
 Oracle Statspack 11g – die kostenlose, leistungsstarke Alternative zum Diagnostic Pack“  
 Norbert Debes, ORADPRO Consulting Services

**05./06. März 2013**  
 „Oracle Parallel Execution Masterclass“  
 Randolph Geist

**11./12. Juni 2013**  
 „Engeneering Oracle for Performance“  
 Dr. Günter Unbescheid, Database Consult GmbH



### VORANKÜNDIGUNG

17. April 2013 | München

**DOAG 2013 Business Intelligence**

Im aktuellen ökonomischen Umfeld gewinnt der Einsatz von effektiven Business Intelligence- und Data Warehouse-Lösungen immer mehr an Wichtigkeit. Die Fachkonferenz bietet in diesem Kontext die ideale Plattform für eine intensive Weiterbildung.

Christian Weinberger  
 office@doag.org

<http://bi.doag.org>

14. Mai 2013 | Düsseldorf

**DOAG 2013 Datenbank**

Datenbank-Administratoren und technisch Interessierten eröffnet sich die Gelegenheit, einen Tag rund um die klassischen Themen der Oracle Datenbank sowie MySQL und Embedded Database zu erleben.

Christian Trieb  
 office@doag.org

<http://datenbank.doag.org>