

Red Stack

Magazin

DOAG

SOUG
swiss oracle
user group

AOUG
AUSTRIAN ORACLE USER GROUP

inklusive BUSINESS NEWS

PL/SQL



Aus der Praxis

Unbekannte PL/SQL-Fähigkeiten entdeckt!



Im Interview

PL/SQL-Guru
Steven Feuerstein

Business News

Lieferkettenmanagement

Programm
jetzt online!

DOAG 2022
Konferenz + Ausstellung
In Nürnberg

20.-23.
SEPT.

Die Oracle- ANWENDERKONFERENZ

anwenderkonferenz.doag.org



Eventpartner:





Sabine Heimsath
Selbstständige IT-Beraterin
und Oracle ACE



Christian Schwitalla
Leiter DOAG PL/SQL
Community



Ausgabe Nr. 4/2022
auf Abruf!

Liebe Mitglieder, liebe Leserinnen und Leser,

diese Ausgabe des Red Stack Magazin hat PL/SQL als Schwerpunkt. Die prozedurale, ADA- basierte Erweiterung von SQL wurde von Oracle Ende der 80er eingeführt. Seitdem wurde die leicht zu erlernende PL/SQL kontinuierlich weiterentwickelt und hat sich zu einer beliebten Datenprogrammierungssprache entwickelt, die sogar über Oracles Grenzen hinaus Einzug gehalten hat (IBM DB2, PL/pgSQL in PostgreSQL). Obwohl schon so lange am Markt vorhanden, wird PL/SQL noch immer intensiv genutzt und hat in den letzten Jahren sogar an Verbreitung zugewonnen, nicht zuletzt dank APEX. PL/SQL beherbergt auch heute noch viele unbekannte Fähigkeiten. Ulrike Schwinn, Jürgen Sieben, Stefan Oehrli, Carsten Thalheimer, Philipp Hartenfeller und Philipp Salvisberg entführen Sie auf eine spannende Reise in die Welt der Datenbankprogrammierung.

Eine besondere Ehre ist für uns, Steven Feuerstein (<https://www.stevenfeuerstein.com/>) als Interview-Partner zu begrüßen. Steven hat zehn Bücher zum Thema PL/SQL veröffentlicht und gilt weltweit als einer der anerkanntesten PL/SQL-Experten. Samuel Nitsche spricht mit Steven über PL/SQL und die wichtigsten Dinge im Leben.

Im Gastkommentar stellt Günther Stürner die Frage: „Sind die fetten Jahre vorbei?“ Im Kontext der aktuellen Krisen erscheint die Frage aktueller denn je. Die gestörten globalen Lieferketten lassen uns alle spüren, dass die Weltwirtschaft aus dem Gleichgewicht geraten ist. Umso wichtiger erscheint das Titelthema der Business News „Lieferketten-Management“. Weitere Themen hier sind Nachhaltigkeit, wie auch der Schutz von Frauen in der Arbeitswelt.

Eine noch größere Themenvielfalt bietet die DOAG Konferenz + Ausstellung 2022. Sie findet als Präsenzveranstaltung in Nürnberg statt. Bitte beachten Sie den geänderten Monat der Konferenz: September statt wie bisher üblich November. Nähere Informationen finden Sie auf der DOAG Homepage. Hoffentlich sehen wir uns im September in Nürnberg!

Viel Spaß bei der Lektüre wünscht Ihnen das gesamte Team des Red Stack Magazin!

Bleiben Sie gesund.

S. Heimsath *Christian Schwitalla*

Sabine Heimsath und Christian Schwitalla

DOAG

WEBSESSION

Die DOAG WebSessions bieten Ihnen in regelmäßigen Abständen spannende Online-Vorträge und -Diskussionen zu einer Vielzahl von Themenbereichen aus den jeweiligen DOAG Communities.

Freuen Sie sich auf WebSessions rund um die Themen Datenbank, Data Analytics und NetSuite oder beteiligen Sie sich bei den DOAG Dev Talks an interessanten Gesprächsrunden zu aktuellen Development-Themen!



<https://shop.doag.org/WebSessions>



*Die Buchung der WebSessions erfolgt ganz einfach über unseren Shop.
Mitglieder erhalten im Buchungsprozess automatisch
100 % Rabatt.



08

Interview mit Steven Feuerstein



16

PL/SQL-Performance analysieren mit PL/SQL Profiler



23

Weitgehend unbekannte PL/SQL-Fähigkeiten entdeckt!

Einleitung

- 3 Editorial
- 6 Timeline
- 8 „Eine Auswirkung ist, dass die Benutzeroberfläche jetzt DAS Wichtigste ist – und die Datenbank so weit von der Benutzererfahrung entfernt ist.“
Interview mit Steven Feuerstein
- 14 Aus der Ferne betrachtet: Sind die fetten Jahre vorbei?
Günther Stürner

PL/SQL

- 16 PL/SQL-Performance analysieren mit PL/SQL Profiler
Ulrike Schwinn
- 23 Weitgehend unbekannte PL/SQL-Fähigkeiten entdeckt!
Jürgen Sieben
- 32 Testen mit utPLSQL – Leicht gemacht mit SQL Developer
Philipp Salvisberg

Datenbank

- 40 Verzeichnisbasierte Namensauflösung von Oracle-Datenbanken
Stefan Oehrli

APEX

- 47 APEX-Security-Grundlagen
Philipp Hartenfeller

MySQL

- 52 MySQL Database Service & HeatWave – „Out of the Box Analytics“ für MySQL
Mario Beck & Carsten Thalheimer

Security

- 57 Sicherung der Akronyme: NFTs, Web3, Crypto, Blockchain und das Metaverse
Anna Collard

Lieferkettenmanagement

60 Leitartikel | Chancen des Lieferkettengesetzes durch Planungsprozesse frühzeitig erkennen

Dr. Birgit Breitschuh und Jochen Rahm

65 Branchenweite Standards zum Schutz von Frauen in der textilen Lieferkette – wie eine kleine Fabrik als Inspiration diente

Bettina Ernst

70 Von Nachhaltigkeit und deren Nachweis in Lieferketten

Markus Adler und Florian Lösch

76 Die Bedeutung von transparenten Lieferketten für nachhaltige Elektrobatterien

Franziska Grammes



76

Die Bedeutung von transparenten Lieferketten für nachhaltige Elektrobatterien



32

Testen mit utPLSQL – Leicht gemacht mit SQL Developer



65

Branchenweite Standards zum Schutz von Frauen in der textilen Lieferkette

Intern

81 Neue Mitglieder + Termine

82 Impressum + Inserenten

News

39 DOAG Datenbank Kolumne: Der Wert einer guten Beratung

51 Oracle Datenbanken Monthly News

59 Welcome back in Nürnberg: #DOAG2022 Auf ein Neues zum großen Wiedersehen

80 Best of DOAG Online Mai/Juni 2022

TIMELINE

13. Mai 2022

In der 60-minütigen WebSession mit Rainier Kaczmarczyk erfahren die Teilnehmer alles rund um die Software „Dbvisit MP – Von der Evolution zur Revolution!“

17. Mai 2022

Im DevTalk mit Jan-Peter Timmermann, Christian Rokitta und Oliver Lemm steht das Thema „APEX Upgrade“ auf der Agenda.

19. Mai 2022

Die WebSession mit Stefan Seck behandelt das Thema „DB-Systems auf ODA“.

19. bis 21. Mai 2022

Der DOAG Vorstand lädt die Aktiven des Vereins in Berlin zum Führungskräfteforum und zur Delegiertenversammlung ein. Im Rahmen von Workshops werden gemeinsam Weichen für die zukünftige Ausrichtung der DOAG gestellt. Der Fokus liegt in diesem Jahr in der thematischen Ausrichtung. Die Teilnehmer legen unter der Zielsetzung der Öffnung der DOAG die Themen fest und gewichten sie darauf, auf welche sich in den nächsten Jahren fokussiert werden soll. Erörtert werden auch erforderliche strukturelle Anpassungen.

30. und 31. Mai 2022

Im Van der Valk Airporthotel Düsseldorf findet die DOAG 2022 Datenbank statt. Nach langem Warten freuen sich die Besucher erneut auf ein Wiedersehen mit zahlreichen Vorträgen für Einsteiger und Experten aus den Bereichen Datenbank und Engineered Systems, einen geselligen Community-Abend sowie reichlich Gelegenheit für interessante Gespräche und Networking – endlich wieder vor Ort in Düsseldorf. Für diejenigen, die nicht an der DOAG 2022 Datenbank teilnehmen konnten, gibt es das neue **On-Demand-Ticket** (<https://shop.doag.org/events/ondemand/shop/>). Bis zum Mai 2023 erhalten so alle Interessierten auch noch nachträglich Zugriff auf alle verfügbaren Vortragsfolien und -aufzeichnungen der Konferenz.

1. Juni 2022

Am Regioday 2022 treffen sich zum „KickOff der Regionalgruppen“ die Regionalgruppen zeitgleich an acht Standorten. Nach zwei Jahren Pandemie wird die Regionale Präsenz der DOAG wieder hochgefahren. Nach einem gemeinsamen Teil bietet im zweiten Teil jede teilnehmende Regionalgruppe noch ein eigenes Programm an.

2. Juni 2022

Das Programm der DOAG 2022 Konferenz + Ausstellung, die dieses Jahr schon vom 20. bis 23. September in Nürnberg stattfindet, wird veröffentlicht. Über 300 Sessions aus den Themengebieten Datenbank & Infrastruktur, Development & Middleware, Data Analytics & AI sowie Strategie & Innovation finden sich in der Agenda. Der erste Tag am 20. September ist

einem Thementag gewidmet, klassische Konferenztage sind am 21. und 22. September, am 23. September gibt es abschließend noch einen Schultag. Die Besucher können sich auch wieder auf ein attraktives Abendprogramm, Community-Aktivitäten und reichlich Gelegenheit zum persönlichen Austausch und Networking freuen.

2. Juni 2022

Zum DevTalk mit Oliver Lemm und Jan-Peter Timmermann erfahren die Teilnehmer alles rund um das Thema „Migration von APEX“.

10. Juni 2022

„Nach der Härtung - Database Auditing“ ist das Thema der WebSession mit Marco Mischke.

28. Juni 2022

Im DevTalk mit Sven Böttcher und Niels de Bruijn erfahren die Teilnehmer alles rund um das Thema „Integration mittels Boomi“.

29. Juni bis 2. Juli 2022

Die CloudLand 2022 – das neu ins Leben gerufene Festival der DOAG Cloud Native Community (DCNC) heißt seine Besucher herzlich willkommen. Sie erwartet eine spannende Reise durch vier Themengebiete an vier Tagen im Phantasialand in Brühl! Geboten wird ein Deep-Dive durch die Welten der Container- & Cloud-Technologien, Continuous Integration & Delivery, Microservices & DDD sowie DevOps & Methodik. Auch der Spaß kommt im attraktiven Freizeitpark nicht zu kurz. Die Open-Source-Gemeinschaft der Eclipse Foundation präsentiert parallel am 30. Juni mit JakartaOne German ein achtstündiges Liveprogramm, das als ein besonderes Highlight des Festivals angesehen werden darf und als „Konferenz in der Konferenz“ realisiert wird. Im eigens dafür errichteten Studio ist der Fokus besonders auf die Entwicklung von Cloud-nativen Technologien für Java-Unternehmensanwendungen gerichtet. Auch hier gibt es das Angebot eines **On-Demand-Tickets** (<https://shop.doag.org/events/ondemand/shop/>) für diejenigen, die nicht dabei sein können. Das Ticket ermöglicht für mindestens ein Jahr lang den Zugriff auf alle Beiträge der CloudLand 2022.

8. Juli 2022

In der WebSession mit Markus Flechtner dreht sich alles rund um das Thema „New Features in Multitenant mit 21c“.

European NetSuite User Days

2022
SEPTEMBER
2022

IN NÜRNBERG

Early Bird
bis 26.07.

netsuite.doag.org

DOAG



Presenta a:

Steven
Feuerstein

PL/SQL Evangelista
Quest Software



Aprenda los secretos de cómo mejorar su
código PL/SQL y las mejores prácticas
en la generación del mismo.

Organizan



O'REILLY



„Eine Auswirkung ist, dass die Benutzeroberfläche jetzt DAS Wichtigste ist – und die Datenbank so weit von der Benutzererfahrung entfernt ist.“

Samuel Nitsche, Senior Software-Entwickler bei Smart Enterprise Solutions, unterhielt sich mit PL/SQL-Guru Steven Feuerstein über die Geschichte und Entwicklung von PL/SQL, Datenbankentwicklung, seinen Werdegang und seinen Blick auf die Welt.

Danke Steven, dass du dir die Zeit für dieses Interview genommen hast. Möchtest du dich kurz vorstellen?

Nein, es ist ein Interview, du musst die Fragen stellen (*lachend*)

Okay, dann werde ich dich mal kurz vorstellen: Steven ist eine der kenntnisreichsten Persönlichkeiten und einer der berühmtesten Verfechter, wenn es um Oracles PL/SQL geht.

Ich nenne mich gerne „Popularisator“. Obwohl es vielen vielleicht schwerfällt, das zu glauben, bin ich technisch nicht sehr versiert und kein Informatiker. Manchmal werde ich als „der Vater von PL/SQL“ bezeichnet und ich muss sagen, wenn ich PL/SQL erstellt hätte, wäre es nicht besonders gut. Aber ich scheine gut darin zu sein, Ideen zu kommunizieren und Entwickler zum Lachen zu bringen.

„Popularisator“ – ich liebe diesen Begriff! Du hast auch zehn Bücher über Oracle und PL/SQL geschrieben, Hunderte von Vorträgen und Keynotes rund um den Globus gehalten und hast immer noch ständig erstaunliche Inhalte für Datenbankentwickler parat, zum Beispiel mit deinen zweiwöchentlichen „Feuertips“-Sessions (Insum Feuertips: PL/SQL-Tipps mit Steven Feuerstein: <https://www.insum.ca/resources/feuertips/>). Wie bist du denn in die Softwareentwicklung eingestiegen?

Ich bin ein alter Kerl, ich bin 63, und das bedeutet, dass, als ich auf die Uni ging, Computer gerade in die Köpfe und Leben der Menschen traten. Das bedeutet auch, dass man an einen anständigen Job in der Programmierung gelangen konnte, ohne allzu viel formale Ausbildung in Informatik. Das ist genau das, was ich getan habe.

Ich habe ein paar Kurse im Programmieren belegt, als ich auf dem College war. Ich glaube, ich habe mit ALGOL angefangen und dann mit LISP weitergemacht.

Ich habe mich nur ein wenig damit beschäftigt und hatte dann die Möglichkeit, als Programmierer in einem Forschungsprojekt in FORTRAN auf dem Campus zu arbeiten. Also blieb ich im Grunde ein weiteres Jahr nach meinem Abschluss und arbeitete als Programmierer.

Ich bin da irgendwie hineingerutscht und hatte nie das, was man einen Karriereplan nennt, ich sagte mir nie: Zuerst werde ich diesen Job bekommen, und dann nach ein paar Jahren werde ich bereit sein, den nächsten Job zu machen. Ich wechselte von Job zu Job, lernte Dinge und landete schließlich in einem Job bei Oracle, wieder ohne wirklichen Plan.

Wie hast du PL/SQL dann aufgeschnappt?

Durch einen Job bei der Oracle Corporation 1987! Ich hatte einige Jahre als Berater gearbeitet, hauptsächlich FORTRAN geschrieben und bin über System 1022 für DEC-10-Computer bei Datenbanken gelandet (siehe [1]). Ich war super gelangweilt, wo ich war, also begann ich nach einem neuen Job zu suchen und stieß auf etwas, das bei der Oracle Corporation (nie von

denen gehört) mit relationalen Datenbanken (nie davon gehört) zu tun hatte. Ich hatte keine Ahnung von relationalen Datenbanken, aber ich fand einen Artikel über die 12 Regeln für relationale Datenbanken, lernte sie auswendig, ging zu Interviews und wurde schließlich eingestellt (ich wurde während des gesamten Prozesses nie gefragt, was ich über relationale Technologie wusste). Das war noch vor PL/SQL, obwohl ich glaube, dass sie schon daran gearbeitet haben.

Ich war eine Pre-Sales-Person, was 1987 bedeutete, dass es mein Job war, vor einem Publikum zu stehen, ein paar SQL-Statements zu erstellen und die Leute dafür zu begeistern, wie man leicht nach einer anderen Spalte anordnen kann. Die gute alte Zeit!

Es war irgendwie ein Vorbote meiner zukünftigen Karriere, vermutlich geprägt von Kreativität, der Bereitschaft, neue Dinge auszuprobieren und nicht sehr diszipliniert oder vorsichtig zu sein.

Mir wurde von meinem Manager John Cordell ein SQL-Skript zur Präsentation übergeben und er kam natürlich vorbei, um mich zu beobachten, weil es mein erstes öffentliches Seminar war. Natürlich war ich mit dem Skript nicht zufrieden, also musste ich es ändern und interessanter gestalten. Jedenfalls habe ich genau das getan und es interessanter gemacht, aber es auch nicht gründlich getestet. Mitten in meinem ersten öffentlichen Seminar, in dem ich ein vorgefertigtes Skript ausführte, geschah es: Eingabe drücken, etwas ausführen, Fehler. (*lacht*)

Ich habe es nebenbei repariert, und ich habe nie aufgehört, das in den Hunderten von Präsentationen zu tun, die ich seitdem gegeben habe. Wie ich zu sagen pflege: „Jetzt machen, später perfektionieren.“

Außer, dass du vielleicht nie zum Perfektionieren kommst, du machst einfach weiter. Und dabei vermeiden, dass man auf die Nase fällt.

Unglaublich! Lass uns ein wenig auf PL/SQL konzentrieren: Was ist dein Lieblingsmerkmal von PL/SQL?

Allgemein gefällt mir an PL/SQL am besten, dass Oracle die Integration einer prozeduralen Programmiersprache in die Datenbank hervorragend hinbekommen hat. Es war sehr schlau von Oracle, nicht zu sagen „Wir können Programmiersprachen besser entwerfen als alle anderen“, was eine wirklich üble Tendenz unter Software-Unternehmen ist. Sie denken, weil sie Software A machen können, können sie auch Software B und C und D. Oracle machte diesen Fehler mehrmals und verlor dadurch viel Geld, aber als es um PL/SQL ging, sahen sie sich stattdessen vorhandene Programmiersprachen an, um zu sehen, ob es bereits ein Modell gab, an dem sie sich anlehnen wollten. Sie wählten schließlich ADA, sodass wir eine wirklich gut strukturierte Sprache als Grundlage hatten, die sich dann mit der Datenbank zusammenfügen ließ. Zum Beispiel: Man kann eine For-Schleife nehmen, von eins auf zehn gehen und X machen. Aber sie haben das konzeptionell und syntaktisch erweitert, also kann man diese Schleife zum Beispiel für jede Zeile machen, die von einer Abfrage zurückgegeben wird. Das ist im Großen und Ganzen mein Lieblingsaspekt bei PL/SQL.

Bei den besonderen Features muss ich sagen: String-indizierte Kollektionen sind einfach die besten! Sie werden wohl Maps in anderen Sprachen genannt – oder assoziative Arrays. Das ist auch ihr offizieller Name in PL/SQL. Es gibt so viele kreative Nutzungsmöglichkeiten für sie, sodass man sie verwenden kann, um die Code-Redundanz zu reduzieren – es ist einfach super cool!

Was möchtest du die Leute noch über Datenbankentwicklung im Allgemeinen oder PL/SQL im Besonderen wissen lassen?

Im Allgemeinen, einfach nur keine Angst davor zu haben. Es gab einen großen Schub in der modernen Softwareentwicklung weg von relationalen Datenbanken, was irgendwie verständlich ist, weil alle Menschen nach Veränderung streben und diese relationalen Datenbankthemen als alte Schule angesehen werden, nicht als schick. Daher gab es diese allgemeine Bewegung weg von, oder sagen wir, hin nach weiter oben auf den Stack. Das kann man verstehen, weil die Dinge in der unteren Schicht kommerzialisiert oder versteckt werden, und dann achten die Leute nicht mehr darauf. Das hat üblicherweise ziemlich schlimme Folgen.

Beispiel: Eines der schlimmsten Dinge, die Amazon jemals der Welt, unserem Planeten, angetan hat, ist, uns alle davon zu überzeugen, dass wir nicht für den Versand bezahlen sollten. Du bist ein Verlierer, wenn du für Versand bezahlst. Und doch erweist es sich als sehr schlecht, die gesamte Transportinfrastruktur der Weltwirtschaft unsichtbar und „kostenlos“ zu machen – für unseren Planeten und damit auch für die Menschen.

Wir haben hier eine ähnliche Situation: Wenn man Datenbanken verschwinden lässt und sagt, dass sie nur Massengüter sind und dass sie nicht wirklich wichtig sind und dass man sie nicht bezahlen und nicht auf sie achten sollte – dann stellt sich heraus, dass sie ein ziemlich grundlegender Teil des Stacks sind und dass man auf sie achten muss, wenn man seine Anwendungen optimieren und das Beste aus seinem gesamten Stack herausholen möchte. Ich wünschte, das wäre mehr Leuten klar.

Jede erfolgreiche Anwendung benötigt eine solide Datenbankgrundlage. Und du brauchst eine wirklich großartige Benutzeroberfläche. Aber du brauchst beides und der Schwerpunkt ist derzeit fast vollständig auf tollen Benutzeroberflächen, was im Grunde durch die Smartphones getrieben wird.

Also, wie ich das sehe, muss man Entwickler verstehen, dass sie nichts mit Datenbanken zu tun haben sollten, aber sie sollten jemanden im Team haben, der ein Datenbankexperte ist. Jedes Team sollte ein Gleichgewicht zwischen UI- und Datenbank-Mitarbeitern haben.

Du meinst, dass Smartphones einen großen Einfluss auf unseren Blick für die heutige Softwareentwicklung haben. Kannst du das näher erläutern?

Für mich ist die grundlegendste Transformation durch Computer, dass der Benutzer eines Computers früher Mitarbeiter eines Unternehmens war. Deshalb erstellten wir Unternehmenssoftware, das Unternehmen kaufte die Software, die Un-

ternehmen bezahlten die Schulungen, die Leute lernten, wie man die Benutzeroberfläche verwendet, und sie konnten sie zwar hassen, sie mussten sie allerdings verwenden.

Dann kamen die Smartphones und plötzlich hast du bei allem die Wahl. Und jeder kann sie benutzen. Man kann sich nicht auf die Schulungen verlassen, weil die Leute nur das Gerät benutzen müssen und man so diese mächtige (und manchmal beunruhigende) Vereinfachung der Oberflächen hat.

Seit der Entwicklung von Smartphones hat es so viele grundlegende Veränderungen gegeben, was die Softwareentwicklung antreibt, und ich denke, wir sind gerade dabei, die Auswirkungen davon in den Griff zu bekommen.

Eine Auswirkung ist, dass die Benutzeroberfläche jetzt DAS Wichtigste ist – und die Datenbank so weit von der Benutzererfahrung entfernt ist.

Bei Smartphones ist unsere Software viel direkter an die Kultur gebunden. Unternehmenssoftware war nie so eng mit der öffentlichen Kultur verbunden, sodass sie sich viel langsamer veränderte, viel weniger individualisiert war. Jetzt dreht sich alles um diese persönliche Erfahrung.

Zurück zu den Features von PL/SQL: Wenn du dir eine Änderung in PL/SQL oder ein neues Feature wünschen könntest: Was wäre das?

Ich denke, das wäre volle Orthogonalität. Beispiel: Ich deklariere eine Variable als eine Zahl und anstatt eine statische Zahl zuzuweisen, sollte ich in der Lage sein, eine einzeilige Auswahl genau an der Stelle zu schreiben, die die Variable füllt. Also sollte ich überall, wo ich einen einzigen Wert habe, diesen Wert durch eine einzige Wertabfrage ersetzen können. Solche Sachen.

Vielleicht arbeiten sie sogar daran für Version 23, ich weiß es nicht. Einen großen Schritt in diese Richtung machten sie bereits in 21c, als sie die Iterator-Logik änderten und dramatisch erweiterten – das ist ein weiteres Beispiel, wo sie die Sprache irgendwie vervollständigen. Mehr solcher Dinge wären sehr schön.

Dazu gehört ein boolescher Datentyp in SQL, nicht wahr?

Ja, das wäre schön (*lacht*). Leider kann ich aus meiner Zeit bei Oracle jetzt sehen, dass (a) das Hinzufügen eines brandneuen Datentyps zu SQL viel Arbeit bedeutet und (b) die Art und Weise, wie sie Prioritäten setzen, die Frage aufwirft: „Können Benutzer um diese fehlende Funktion herum arbeiten?“ Wenn nicht, dann höhere Priorität. Wenn ja, dann, na ja: Arbeite drum herum!

Aber ein ganz besonderes Feature, das ich gerne hinzufügen würde, ist die Möglichkeit, Bulk-Sammeln in String-indizierten Sammlungen auszuwählen. Etwas wie „select bulk collect into Index by“ und dann einen Spaltennamen (oder mehrere) angeben für die Indizierung.

Was würdest du sagen, ist der schwierigste Teil der Datenbankentwicklung? Was ist eine große Anstrengung?

Ich würde sagen, die Anstrengungen drehen sich nicht so sehr um die Technologie, sondern um den Prozess. Wie bekommt

man disziplinierte Entwickler, damit sie das Richtige tun und nicht nur viele Abkürzungen nehmen und – Code im Wesentlichen so wie ich schreiben (*lacht*).

Wie bereits erwähnt, bin ich ein Popularisator, ich habe ein paar Bücher geschrieben und Leute mögen, was ich schreibe, sie mögen die Witze, die ich in die Bücher gepackt habe, und sie kommen zu dem Schluss, dass ich ein erstaunlicher PL/SQL-Entwickler sein muss.

Nun ja, nur weil du ein gutes Buch schreibst, heißt das nicht, dass du ein guter Entwickler bist. Ich denke, ich bin ganz gut darin, einzelne PL/SQL-Programmeinheiten zu schreiben, und ich kann es wahrscheinlich schneller als jeder andere auf dem Planeten, aber in Bezug auf die Disziplin, eine Anwendung korrekt und sorgfältig zu erstellen – nee, darin bin ich schrecklich. Ich habe in den letzten Jahren immer wieder dieselben Fehler gemacht und werde das wohl weiterhin tun, bis ich in den Ruhestand gehe.

Also, für mich gibt es nicht viele technische Hindernisse für die PL/SQL-Programmiersprache – es ist eine sehr direkte Sprache. Es geht hauptsächlich um den Prozess und die Disziplin, die notwendig ist, um eine Anwendung zu entwickeln, in der wir eine Herausforderung haben. Das mag zum Teil daran liegen, dass wir, die Datenbankentwickler, nicht so sehr die Praktiken befolgen, die in anderen Entwicklungsumgebungen häufiger vorkommen, wie die Verwendung von git zur Versionskontrolle. Aber ich weiß nicht – JavaScript-Entwickler könnten die gleichen Probleme haben. Allerdings: Die Einfachheit und Lesbarkeit des Datenbankcodes könnte es auch schwieriger machen, es zu vermässeln, als in JavaScript, es ist allerdings immer noch möglich.

Ja, da stimme ich zu – auch die Genauigkeit, vor allem von PL/SQL.

Oracle war schon immer ziemlich patriarchalisch, was man in seiner Datenbank machen kann. Manchmal mögen wir es vielleicht nicht, aber es gibt in der Regel sehr gute Gründe.

Es wäre nicht mein Interview, wenn ich keine Frage zu utPLSQL stellen würde. Vor 20 Jahren hast du automatisierte Tests in die Oracle-Datenbank eingeführt. Du hast sogar ein Framework zur Unterstützung geschrieben – utPLSQL Version 1 und 2. Warum denkst du, haben automatische Tests auf Datenbankebene noch so wenig Akzeptanz?

Vermutlich aus dem gleichen Grund, aus dem sie in jeder anderen Programmierumgebung so wenig akzeptiert werden. Ich meine, glaubst du wirklich, dass ein Großteil der JavaScript-Entwickler automatisierte Regressionstests verwendet?

Ich denke, es hängt von vielem ab, aber es gibt sicherlich Gemeinschaften von Programmiersprachen, die automatisierte Tests wesentlich besser annehmen. Nehmen wir zum Beispiel Python mit seinem dedizierten „Test“-Schlüsselwort oder Java mit einer starken Akzeptanz automatisierter JUnit-Tests. Aus meiner Sicht ist die Akzeptanz in der Datenbank-Community besonders gering.

Ich will nur fürs Protokoll sagen, dass ich glaube, wenn wir den tatsächlichen Anteil unter den Fantastilliarden von Java-Entwicklern herausfinden können, die automatisierte Tests verwenden und annehmen, wird sich dieser Prozentsatz wahrscheinlich nicht so sehr von PL/SQL-Entwicklern unterscheiden. Das wäre meine Vermutung. Abgesehen davon gibt es einige besondere Herausforderungen im Zusammenhang mit dem Testen von Datenbankcode. Bekanntermaßen besteht die Idee des Unit-Tests darin, dass man eine einzelne Code-Einheit nimmt, die unabhängig ist, und sie einfach testet, weil sie deterministisch ist. Dies sind die einfachsten zu testenden Sachen. Aber sobald man Datenbankabhängigkeit hat, kommen alle möglichen zusätzlichen Herausforderungen ins Spiel. Der wahrscheinlich wichtigste Grund ist wohl, dass die Hürden, automatisierte Tests in der Datenbank durchzuführen, in der Regel viel höher sind. Die Hürden, automatisierte Tests durchzuführen, sind im Allgemeinen bereits hoch, weil es Zeit braucht, Wissen, Management-Aufklärung bis zu einem gewissen Grad, um Entwicklern die Zeit und die Werkzeuge dafür zu geben. Hinzu kommen noch die zusätzlichen Herausforderungen durch Datenbanken. Ein weiterer Grund könnte sein, dass die Datenbank-Community eine Gemeinschaft ist, die schon älter als ... vielleicht nicht älter als automatisiertes Testen ist, aber sicherlich älter als Dinge wie Test-Driven-Development, Extreme Programming und andere Trends der „agilen Bewegung“.

Es könnte also auch ein kulturelles Problem sein?

Ja, ich denke schon. Ich meine – ich habe JUnit gesehen, als es herauskam, und dachte, es wäre cool, das für PL/SQL zu haben. Aber ich habe es nie konsequent genutzt, habe es nie wirklich in meinem Entwicklungsprozess verwendet und werde es wahrscheinlich auch in Zukunft nicht tun.

Ich glaube immer noch nicht, dass unsere Gemeinschaften so unterschiedlich sind.

Wir haben die zusätzlichen Herausforderungen von Datenbanken. Und wir haben die zusätzlichen Herausforderungen, dass die Datenbankseite des Stacks nicht so sehr respektiert wird, nicht so viel Aufmerksamkeit erhält, was bedeutet, dass die Tools fehlen, die für andere Teile des Stacks existieren. Es bedeutet auch, dass wir nicht so viel Geld und Ressourcen bekommen, um gute Tests durchzuführen.

Lass uns mit ein paar Fragen schließen, bei denen es nicht um PL/SQL geht. Was ist einer der besten Ratschläge, die du in deiner Karriere bekommen hast?

Normalisiere! Ich denke, dass einer der Gründe, warum ich relationale Datenbanken so sehr mag, ist, dass die Idee, Daten zu normalisieren, um Redundanz (und mehr) zu vermeiden, sehr stark mit einigen meiner internen Treiber synchron geht, nämlich keine Zeit oder Ressourcen zu verschwenden.

Normalisiere nicht nur Daten, sondern auch Code! Vermeide Redundanz im Code, mach die Struktur deines Codes sauber und einfach zu verwalten. Für mich ist einer der eindringlichsten Ratschläge, die ich befolge, zu modularisieren: Programmeinheiten klein zu halten, sich nicht in einem riesigen Haufen Spaghetti-

Code zu verlieren. Das ist eines der Prinzipien, nach denen ich zu leben versuche, während ich meinen Code schreibe.

Was würdest du jemandem sagen, der gerade angefangen hat, seinen Weg in der Softwareentwicklung zu gehen?

Leg deinen Computer zur Seite und geh raus! Mehr als das, stell deinen Computer ab und SEI draußen.

Das ist doch mal ein Ratschlag!

Ich würde sagen, der wichtigste Ratschlag, den ich jedem Menschen geben würde, der heute lebt, besonders jedem, der in der technologischen, „modernen“, „zivilisierten“ Welt lebt, ist: Hört auf, euch vor eurem Planeten zu verstecken!

Ich sage das nicht nur, weil der einzige Weg, wie wir möglicherweise so viel von unserem Planeten zerstören können, darin besteht, dass wir es nicht wahrnehmen, indem wir uns so sehr vom Planeten distanzieren, sondern auch, weil wir, wenn wir nach draußen gehen, in der Lage sein werden, klarer zu denken, wir werden in der Lage sein, Probleme effektiver zu lösen, wir werden gesünder sein, wir werden uns in unserem eigenen Körper wohler fühlen. All diese Dinge werden dich zu einem besseren Entwickler machen.

Es ist wirklich bemerkenswert, Sam, darüber nachzudenken, wie viele Menschen – und wir sprechen hier hauptsächlich von Menschen der Mittelschicht in einem Land wie den Vereinigten Staaten – fast keinen direkten Kontakt mit dem Planeten Erde haben. Wir leben in unseren Kisten. Wir wachen in unserer Kiste (Zuhause) auf. Wir gehen auf unseren Bürgersteigen zu unserer Autokiste. Wir fahren auf Straßen, die uns zu unserer Büro-kiste bringen. Und es gibt ein klares gemeinsames Merkmal all dieser Kisten: Sie haben fast nichts Lebendiges in sich. Außer uns und einigen Pflanzen.

Du gehst eine Straße in deiner Stadt entlang und wenn du Glück hast, gibt es einen Baum, an dem du vorbeigehst, was wirklich nur eine winzige Lücke im Gehwegbeton ist, die gerade mal für das Öffnen eines Kofferraums reicht.

Für mich sind wir so weit von jeder vernünftigen und positiven Art, auf diesem Planeten zu leben, entfernt, dass das Erste, was wir tun müssen, ist, uns wieder mit ihm zu verbinden.

Zeit für eine letzte Frage: Wenn du kein Softwareentwickler wärst – was wärst du, wie würdest du arbeiten oder deine Energie in Vollzeit konzentrieren?

Ich denke, das Wichtigste, was jeder Mensch heute tun sollte, ist, daran zu arbeiten, unseren Planeten zu heilen. Also würde ich dasselbe tun wie jetzt, mit all der Zeit, die ich nicht vor einem Computer verbringe. Ich verbringe so viel Zeit wie möglich im Freien, kämpfe gegen invasive Arten und helfe unseren einheimischen Ökosystemen zu gedeihen.

Für mich ist das wichtigste Prinzip das direkte und positive Handeln.

Man kann wählen, man kann spenden, man kann Petitionen unterschreiben, aber was wirklich wichtig ist, ist, dass man in die Welt hinausgeht und einen Unterschied macht, wenn es darum geht, unseren Planeten zu heilen und das Leben von Nicht-Menschen zu retten. Und bringt eure Kinder mit!

Das ist ein wunderbares Schlusswort. Vielen Dank für dieses Gespräch.

Quelle

[1] <https://books.google.com/books?id=tBXQZbbSyeQC&pg=PA78&lp-g=PA78&dq=digital+equipment+corporation+1022+database&source=bl&ots=aOGCKa4ZDy&sig=ACfU3U24uwmpyZtON-VUnUZPjYvK9j4TKw&hl=en&sa=X&ved=2ahUKewiExv3Xqbn3Ah-WbgnlEHeK6DplQ6AF6BAgoEAM#v=onepage&q=1022&f=false>



STEVEN FEUERSTEIN

Steven ist seit 1992 besessen von Oracles PL/SQL-Sprache. Es wurde so schlimm, dass er 1994 ein Buch darüber schrieb und seitdem kann er nur noch über PL/SQL-Pakete, PL/SQL-Sammlungen, PL/SQL you-name-it sprechen und schreiben. Wenn er nicht vor einem Computer sitzt, ist er draußen in den Wäldern von North Carolina, um invasive Arten zu beseitigen und Bäume zu retten. Steven ist seit Februar 2021 Senior Advisor für Insum Solutions.



DAS CLOUD NATIVE FESTIVAL

DAS EVENT DER DEUTSCHSPRACHIGEN
CLOUD NATIVE COMMUNITY

on demand

DAS CLOUD NATIVE FESTIVAL VERPASST?

**JETZT ON-DEMAND-TICKET BUCHEN UND
VORTRAGSAUFZEICHNUNGEN ANSCHAUEN!**

Alle Angebote im On-Demand-Ticket-Shop





AUS DER FERNE BETRACHTET:

SIND DIE FETTEN JAHRE VORBEI?

Ich gebe zu, ich liebe Statistiken, obwohl ich kein Statistiker bin und obwohl ich sehr gut weiß, dass man mit dem Jonglieren von Zahlen viel Unsinn und auch Unheil anrichten kann. Die „Unstatistik des Monats“, die von vier renommierten Wissenschaftlern herausgegeben wird, zeigt die Untiefen der Zahlenwelt Monat für Monat. Vorsicht ist stets geboten, wenn Statistiken veröffentlicht werden, und wir würden gut daran tun, diese Disziplin schon in der Schule besser zu verankern, denn auch hier gilt: „Wer nichts weiß, muss alles glauben.“ Aber das ist ein anderes Thema.

Die Statistik, um die es mir heute geht, ist die Aufstellung der neuesten Marktzahlen von Gartner in Bezug auf die Marktanteile im Datenbankumfeld für 2021. Immer im Frühjahr werden die Zahlen veröffentlicht. Auch diese Zahlen sind mit Vorsicht zu genießen, denn auch Gartner hat keinen „Knopf“, auf den sie drücken können, um die Ergebnisse zu liefern. Die Erhebung der Zahlen basiert auf Marktanalysen, auf veröffentlichten Zahlen, auf Geschäftsberichten und Interviews mit den Herstellern. Eine einheitliche Berichtsweise der unterschiedlichen Software-Hersteller für Datenbank-Software gibt es nicht. Eine gewisse Unschärfe muss also stets einkalkuliert und in Kauf genommen werden.

Und doch ist diese Analyse ein Muss für alle, die sich in diesem Marktsegment tummeln.

Für Oracle und insbesondere für die verantwortlichen Manager des Datenbankproduktes war diese „Hitliste“ über viele Jahre eine Augenweide. Unangefochtene Nummer 1 mit großem Abstand zum Zweitplatzierten – früher IBM mit DB2, später dann Microsoft mit SQL-Server. Laut Gartner dominierte Oracle 2017 den Datenbankmarkt mit einem Marktanteil von 36,1 % und hat-

te einen komfortablen Vorsprung von fast 15 Prozent-Punkten zum Zweiten und 23 Prozent-Punkten zum Dritten.

Das nennt man unangefochtene Marktführerschaft.

Kritische Anmerkungen von Kunden, aber auch von internen Kollegen, zu Verkaufsstrategie und Lizenzpolitik wurden entweder ignoriert oder niedergebügelt. Wie sagt man: „Erfolg hat immer Recht“, und Spielverderber, das wusste Erich Kästner aus eigener Erfahrung, „Spielverderber hasst man weit mehr und fanatischer als die Falschspieler“.

Wie gesagt, das war 2017 und in den vielen Jahren davor.

Ein völlig anderes Bild zeigt sich in den Zahlen von 2021. Oracle ist nicht mehr glänzender und unangefochtener Erster, sondern lediglich Drittplatzierter, überholt von Microsoft (Platz 1) und Amazon AWS (Platz 2). Eine schleichende Erosion hat die Dominanz und den einst riesigen Marktanteil-Vorsprung jedes Jahr im Durchschnitt um vier Prozent abgeschmolzen. Von stolzen 36,1 % im Jahr 2017 auf jetzt (2021) 20,6 %.

Man könnte nun meinen, dass die Oracle-Datenbank keine Käufer mehr gefunden hat. Aber das ist nicht der Fall. Die Umsätze der Oracle-Datenbank haben auch in den letzten Jahren zugelegt und Oracle hat eine gute Rendite mit seinem „Flagship“-Produkt erzielt. Dies war allerdings nur möglich, weil sich das Datenbank-Marktpotenzial fulminant vergrößert hat. Gartner rechnet vor, dass 2017 im Datenbankmarkt 40 Mrd. US-Dollar zu verteilen waren – 2021 das Doppelte. Nimmt man diese Zahlen als Basis, ist der Umsatz des wichtigsten Oracle-Produktes in fünf Jahren um 14 % gewachsen. Das könnte man moderates Wachstum nennen. Wenn man jedoch die Wachstumsraten

der Konkurrenten auf der gleichen Basis errechnet, ergibt sich für den SQL-Server ein Wachstum von über 120 %, für AWS eine phänomenale Verfünfachung. Die 14 % moderates Wachstum erscheinen unter diesem Aspekt in einem völlig anderen Licht.

Man müsste annehmen, dass diese Zahlen einige Veränderungen bei den seit Langem bekannten neuralgischen Punkten und Fehlentwicklungen der vergangenen Jahre zur Folge haben.

Ja, Oracle hat in den letzten Jahren die Datenbank auf einen exzellenten technologischen Stand gebracht. Technologisch ist alles vom Feinsten. Und doch gehen viele Kunden auf Distanz, prüfen Alternativen für neue Projekte, die es zuhauf gibt. Versuchen sich von Oracle zu lösen, um sich von den komplexen und komplizierten Verträgen freizumachen.

Oracle hat kein technologisches Problem im Datenbankumfeld, die Krux liegt im Bereich Kundenbeziehung, im Bereich Vertrauen und in einem antiquierten Lizenzmodell, das virtuelle Umgebungen ignoriert.

Wo sind die Veränderungen an den Schwachpunkten, wo das Aufbegehren und der Wille, die Konkurrenten wieder in die Schranken zu weisen, wo ein neuer Spirit, wo ein neues Feeling?

Eins ist sicher: Mit dem Umzug des Headquarters nach Austin/Texas, um Steuern zu sparen, ist es nicht getan.

Günther Stürner

E-Mail: guenther.stuerner@dbms-publishing.de



MUNIQSOFT
— CONSULTING —



Probleme lösen mit IQ

Telefon-/Remotesupport für Oracle Datenbanken

Wenn die Technik mal streikt: Unsere zertifizierten Oracle Spezialisten sind für Sie da - zuverlässig, persönlich, deutschsprachig.

Muniqsoft Consulting -
und Sie bleiben selbst im Notfall entspannt.

ORACLE | Partner



Jetzt Supportvertrag abschließen!

+49 (0)89 6228 6789-21

www.muniqsoft-consulting.de



PL/SQL-Performance analysieren mit PL/SQL Profiler

Ulrike Schwinn, Oracle Deutschland

Wie kann man PL/SQL-Code untersuchen, um Performance-Engpässe zu finden? Diese Frage erhalte ich in regelmäßigen Abständen, wenn Oracle-Datenbank-Applikationen getestet, analysiert und getunt werden. Wenn es sich um die Analyse von „reinem“ SQL-Code handelt, ist die Antwort einfach: Die Nutzung von Ausführungsplänen, SQL Traces, AWRs, SQL Access Advisor und Performance Analyzer ist dabei immer das richtige Mittel der Wahl. Wie geht man aber bei der Analyse von PL/SQL-Code vor?

Die Lösung dazu liefert die PL/SQL-Hierarchical-Profiler-Technologie – auch kurz PL/SQL Profiler genannt. PL/SQL Profiler ist ein kostenfreies, internes Interface der Oracle-Datenbank, das das Ausführungsprofil eines PL/SQL-Programms nach Funktionsaufrufen gliedert und dabei SQL- und PL/SQL- Ausführungszeiten getrennt auflistet. Es ist keine spezielle Vorbereitung des Quellcodes oder des Compilers erforderlich; jedes PL/SQL-Programm kann damit überprüft werden. Um zu zeigen, wie einfach und schnell das Arbeiten damit ist, wird die Funktionsweise an einem kurzen Beispiel in den folgenden Abschnitten illustriert.

Wie anfangen?

Um den PL/SQL Profiler zu verwenden, sind folgende Vorbereitungen beziehungsweise Voraussetzungen erforderlich:

- EXECUTE-Privilegien auf das Package DBMS_HPROF
- READ- und WRITE-Privileg auf ein logisches Verzeichnis, in dem die Trace-Datei(en) gespeichert werden sollen
- Erzeugen von Repository-Hilfstabellen, um die Informationen persistent zu speichern

Das logische Verzeichnis (Datenbank Directory) kann dabei einen beliebigen Namen haben. Möchte man allerdings SQL Developer zur Analyse verwenden, muss man den Verzeichnis-Namen PLSHPROF_DIR wählen, um eine fehlerfreie Nutzung zu garantieren.

Führen wir nun die Setup-Schritte an einem Testuser PLTEST durch (siehe Listing 1).

Der letzte Aufruf erzeugt die vier Repository-Tabellen DBMSHP_TRACE_DATA, DBMSHP_RUNS, DBMSHP_FUNCTION_INFO und DBMSHP_PARENT_CHILD_INFO, die zum Speichern und Abfragen der zugehörigen Trace-Informationen in der Datenbank verwendet werden können.

Das in Listing 2 dargestellte einfache Test-Programm soll für die Analyse verwendet werden.

PL/SQL Profiler in Aktion

Nun kann man den PL/SQL Profiler verwenden. Zuerst wird er gestartet, dann erfolgt die Programmausführung und den Abschluss bildet das Stoppen des Profilers. Das Ganze sieht dann im Beispiel folgendermaßen aus (siehe Listing 3).

Und fertig!

Wirft man nun einen Blick in das Verzeichnis PLSHPROF_DIR findet man eine

Trace-Datei mit Namen run1.trc, die die sogenannten „Raw Profiler Output“-Daten – also die unformatierten Performance-Daten – enthält. Diese Daten sind im ASCII-Format und können jetzt manuell zur Analyse herangezogen werden.

Einfach und schnell geht die Analyse mit dem mitgelieferten Programm plshprof im Verzeichnis \$ORACLE_HOME/bin/. Es erzeugt bei der Ausführung formatierte und gut lesbare HTML-Berichte aus der Raw-Profiler-Ausgabedatei.

Die Funktionsweise von plshprof lässt sich durch einen einfachen Aufruf wie in Listing 4 dargestellt beschreiben.

Um das Beispiel weiterzuführen, nutzen wir plshprof mit der generierten Trace-Datei run1.trc. Das Ergebnis soll in eine HTML-Datei mit Namen run1.html geschrieben werden (siehe Listing 5).

Die resultierende HTML-Datei run1.html enthält Hyperlinks zu unterschiedlichen Reports, um die Performance der PL/SQL-Anwendung nach verschiedenen Gesichtspunkten zu analysieren. So wird die Elapsed-Zeit auf Funktions-, Modul- oder Namespace-Level in unterschiedlichen Zusammenhang gestellt und sortiert ausgegeben. Die Unterprogramme, die der PL/SQL Profiler analysiert, sind dabei in sogenannte Namespaces unterteilt. So steht beispielsweise „_plssql_vm“

```
SQL> create directory PLSHPROF_DIR as '/home/oracle/pl';
SQL> grant all on directory plshprof_dir to pltest;
SQL> grant execute on dbms_hprof to pltest;

-- als User PLTEST

SQL> execute DBMS_HPROF.CREATE_TABLES();
```

Listing 1: Beispiel 1: Initiales Setup

```
CREATE OR REPLACE PROCEDURE testprof AUTHID DEFINER IS
  n NUMBER;

  PROCEDURE foo IS
  BEGIN
    SELECT COUNT(*) INTO n FROM scott.emp;
    dbms_output.put_line('ausgabe: '||n);
  END foo;

BEGIN -- test
  FOR i IN 1..3 LOOP
    foo;
  END LOOP;
END testprof;
/
```

Listing 2: Beispiel 2: Das Test-Programm

```
execute dbms_hprof.start_profiling(LOCATION=>'PLSHPROF_DIR', -
                                  FILENAME=>'run1.trc');

execute testprof;

execute dbms_hprof.stop_profiling;
```

Listing 3: Beispiel 3: Anwendung des PL/SQL Profiler

```
[oracle@de pl]$ plshprof

PLSHPROF: Release 19.0.0.0.0 - Production on Wed Nov 3 16:59:38 2021
Version 19.12.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.
All rights reserved.

Usage: plshprof [<option>...] <tracefile1> [<tracefile2>]
Options:
  -trace <symbol>      (no default)    specify function name of tree
root
  -skip <count>        (default=0)     skip first <count> invokations
  -collect <count>    (default=1)     collect info for <count> in-
vokations
  -output <filename>  (default=<symbol>.html or <tracefile1>.html)
  -summary              print time only
```

Listing 4: Beispiel 4: Funktionsweise von PLSHPROF

für den Call der PL/SQL Virtual Machine, „_anonymous_block“ für einen PL/SQL Anonymous Block und „_static_sql_exec_lineno#“ für statisches SQL in Zeile line#. In unserem Beispiel sieht die erste Seite eines plshprof-Reports wie in *Abbildung 1* dargestellt aus.

Insgesamt benötigt die Ausführung 317 Mikrosekunden für 15 Funktionsaufrufe. Gemessen an der absoluten Funktionszeit kann man im zweiten Abschnitt gut erkennen, dass das SQL Statement, das sogar mit einer SQL ID gekennzeichnet ist, die meiste Zeit verbraucht und mehrfach vom Programm FOO aufgerufen wird. Darüber hinaus erhält man weitere Berichte über die Hyperlinks, um den Zeitverbrauch von SQL und PL/SQL oder der einzelnen Funktionen untereinander zu vergleichen, um nur ein paar Beispiele zu nennen. Eine ausführliche Beschreibung zu allen Abschnitten des plshprof-Reports findet sich im Database Developer Guide im Kapitel 15.5 plshprof Utility [1].

Möchte man die Informationen aus der Trace-Datei persistent in der Datenbank speichern, um zum Beispiel die Daten zu einem späteren Zeitpunkt weiterzuverarbeiten, ist zusätzlich die Ausführung einer ANALYZE-Prozedur erforderlich (siehe Listing 6).

Jetzt sind die vier Repository-Tabellen gefüllt. Mit entsprechenden SQL-Abfragen lassen sich nun die Informationen beliebig auswerten. Im folgenden Beispiel (siehe Listing 7) erhält man einen Überblick über alle Testläufe. Insgesamt sind zwei Testläufe durchgeführt worden.

SQL Developer

Die Aufrufe für PL/SQL Profiler lassen sich natürlich auch wie alle PL/SQL-Aufrufe im Worksheet von SQL Developer verwenden. Darüber hinaus gibt es zusätzlich die Möglichkeit, eine automatisierte Anwendung zu nutzen. Dazu muss man die entsprechende Prozedur, Funktion oder das Package im Menü auswählen und im Worksheet den „Profiler“-Button betätigen. Dieser automatisiert das Profiling und die Speicherung in den Repository-Tabellen. Die Nutzung im SQL Developer funktioniert dann über den „Debug“-Button. Wichtig ist dabei die Verwendung eines logischen Directory mit Namen PLSHPROF_

DIR, um eine fehlerfreie Funktionsweise zu garantieren (siehe Abbildung 2).

Weitere Beschreibungen dazu finden sich beispielsweise in folgenden Beiträgen:

- Finding Slow PL/SQL: Use the Profilers! (Video) [2]
- SQL Developer & the PL/SQL Hierarchical Profiler (Posting) [3]

Keinen Zugriff auf ein logisches Directory – was tun?

Aber was tun, wenn man nur eingeschränkte Privilegien hat und beispielsweise kein Zugriff auf ein logisches Directory zur Verfügung steht. Hier gibt es die Möglichkeit mit dem „altbewährten“ Package DBMS_PROFILER zu arbeiten, das die Daten direkt in Datenbank-Tabellen schreibt. Der Ablauf ähnelt dem beschriebenen Vorgehen von DBMS_HPROF.

Zuerst werden die zugehörigen Repository-Tabellen PLSQL_PROFILER_RUNS, PLSQL_PROFILER_DATA und PLSQL_PROFILER_UNITS im Applikationsschema erzeugt. Dazu verwendet man das Dienstprogramm proftab.sql aus dem \$ORACLE_HOME/rdbms/admin-Verzeichnis (siehe Listing 8).

Danach erfolgen das Starten des Profilers, die Code-Ausführung und das Stoppen – dieses Mal statt mit DBMS_HPROF mit DBMS_PROFILER. Das Ganze sieht dann zum Beispiel folgendermaßen aus (siehe Listing 9).

Nun sind die Tabellen gefüllt und man kann die entsprechenden Abfragen durchführen, etwa um zu erfahren, welche Tests es überhaupt gibt – mit einer Abfrage auf PLSQL_PROFILER_RUNS (siehe Listing 10).

Falls man ähnliche Unterstützung zum Erzeugen eines HTML-Reports sucht, wie im vorangegangenen Abschnitt mit plshprof gezeigt, kann man dazu das Skript profiler.sql verwenden. Man findet es in der My Oracle Support Note Doc ID 243755.1 [4] oder kann es im entsprechenden Blogbeitrag [5] herunterladen. Folgender Aufruf liefert dann den zugehörigen HTML-Report für den Lauf mit Runid 1 (siehe Listing 11).

In meinem Fall benötigte der Applikationsuser das zusätzliche Privileg, um auf den Data-Dictionary-Katalog wie zum

```
[oracle@ede pl]$ plshprof run1.trc
[7 symbols processed]
[Report written to 'run1.html']
```

Listing 5: Beispiel 5: Verwendung von PLSHPROF

```
SQL> var runid number
SQL> execute :runid:=DBMS_HPROF.analyze('PLSHPROF_DIR', 'run1.trc');
PL/SQL procedure successfully completed
```

Listing 6: Beispiel 6: Trace-Daten festschreiben

```
SQL> select * from DBMSHP_RUNS;

          RUNID RUN_TIMESTAMP                                TOTAL_ELAPSED_TIME
-----
-----
RUN_COMMENT
-----
          TRACE_ID
-----
                1 08-JAN-22 05.51.53.529303 PM                317
                1
                2 08-JAN-22 06.18.34.653007 PM                2716
                2
```

Listing 7: Beispiel 7: Repository-Tabellen verwenden

```
SQL> start proftab.sql;
```

Listing 8: Beispiel 8: Anlegen der Repository-Tabellen

```
SQL> execute dbms_profiler.start_profiler;

SQL> execute testprof;

SQL> execute dbms_profiler.stop_profiler;
```

Listing 9: Beispiel 9: Ausführung mit DBMS_PROFILER

```
SQL> select runid, run_date, run_comment from plsqli_profiler_runs;
RUNID      RUN_DATE      RUN_COMMENT
-----
-----
2          13-NOV-21    13-NOV-21
1          2-NOV-21     12-NOV-21
```

Listing 10: Beispiel 10: Abfragen über die Repository-Tabellen

Function Elapsed Time (microsecs) Data sorted by Total Subtree Elapsed Time (microsecs)

317 microseconds (elapsed time) & 15 function calls

| Subtree | Ind% | Function | Ind% | Descendants | Ind% | Calls | Ind% | Function Name | SQL ID | SQL TEXT |
|---------|-------|----------|-------|-------------|-------|-------|-------|---|----------------|--------------------------------|
| 317 | 100% | 5 | 1.6% | 312 | 98.4% | 2 | 13.3% | _plsql_vm | | |
| 312 | 98.4% | 50 | 15.8% | 262 | 82.6% | 2 | 13.3% | _anonymous_block | | |
| 262 | 82.6% | 4 | 1.3% | 258 | 81.4% | 1 | 6.7% | PLTEST.TESTPROF.TESTPROF (Line 1) | | |
| 258 | 81.4% | 69 | 21.8% | 189 | 59.6% | 3 | 20.0% | PLTEST.TESTPROF.TESTPROF.FOO (Line 4) | | |
| 187 | 59.0% | 187 | 59.0% | 0 | 0.0% | 3 | 20.0% | PLTEST.TESTPROF._static_sql_exec_line6 (Line 6) | 6f4y4j'n6qy056 | SELECT COUNT(*) FROM SCOTT.EMP |
| 2 | 0.6% | 2 | 0.6% | 0 | 0.0% | 3 | 20.0% | SYS.DBMS_OUTPUT.PUT_LINE (Line 109) | | |
| 0 | 0.0% | 0 | 0.0% | 0 | 0.0% | 1 | 6.7% | SYS.DBMS_HPROF.STOP_PROFILING (Line 742) | | |

Function Elapsed Time (microsecs) Data sorted by Total Function Elapsed Time (microsecs)

317 microseconds (elapsed time) & 15 function calls

| Subtree | Ind% | Function | Ind% | Cum% | Descendants | Ind% | Calls | Ind% | Function Name | SQL ID | SQL TEXT |
|---------|-------|----------|-------|-------|-------------|-------|-------|-------|---|----------------|--------------------------------|
| 187 | 59.0% | 187 | 59.0% | 59.0% | 0 | 0.0% | 3 | 20.0% | PLTEST.TESTPROF._static_sql_exec_line6 (Line 6) | 6f4y4j'n6qy056 | SELECT COUNT(*) FROM SCOTT.EMP |
| 258 | 81.4% | 69 | 21.8% | 80.8% | 189 | 59.6% | 3 | 20.0% | PLTEST.TESTPROF.TESTPROF.FOO (Line 4) | | |
| 312 | 98.4% | 50 | 15.8% | 96.5% | 262 | 82.6% | 2 | 13.3% | _anonymous_block | | |
| 317 | 100% | 5 | 1.6% | 98.1% | 312 | 98.4% | 2 | 13.3% | _plsql_vm | | |
| 262 | 82.6% | 4 | 1.3% | 99.4% | 258 | 81.4% | 1 | 6.7% | PLTEST.TESTPROF.TESTPROF (Line 1) | | |
| 2 | 0.6% | 2 | 0.6% | 100% | 0 | 0.0% | 3 | 20.0% | SYS.DBMS_OUTPUT.PUT_LINE (Line 109) | | |
| 0 | 0.0% | 0 | 0.0% | 100% | 0 | 0.0% | 1 | 6.7% | SYS.DBMS_HPROF.STOP_PROFILING (Line 742) | | |

SQL ID Elapsed Time (microsecs) Data sorted by SQL ID

317 microseconds (elapsed time) & 15 function calls

| SQL ID | SQL TEXT | Function | Ind% | Calls | Ind% |
|----------------|--------------------------------|----------|-------|-------|-------|
| 6f4y4j'n6qy056 | SELECT COUNT(*) FROM SCOTT.EMP | 316 | 99.7% | 3 | 20.0% |

Abbildung 1: plshprof Report – ein Ausschnitt (Quelle: Ulrike Schwinn)

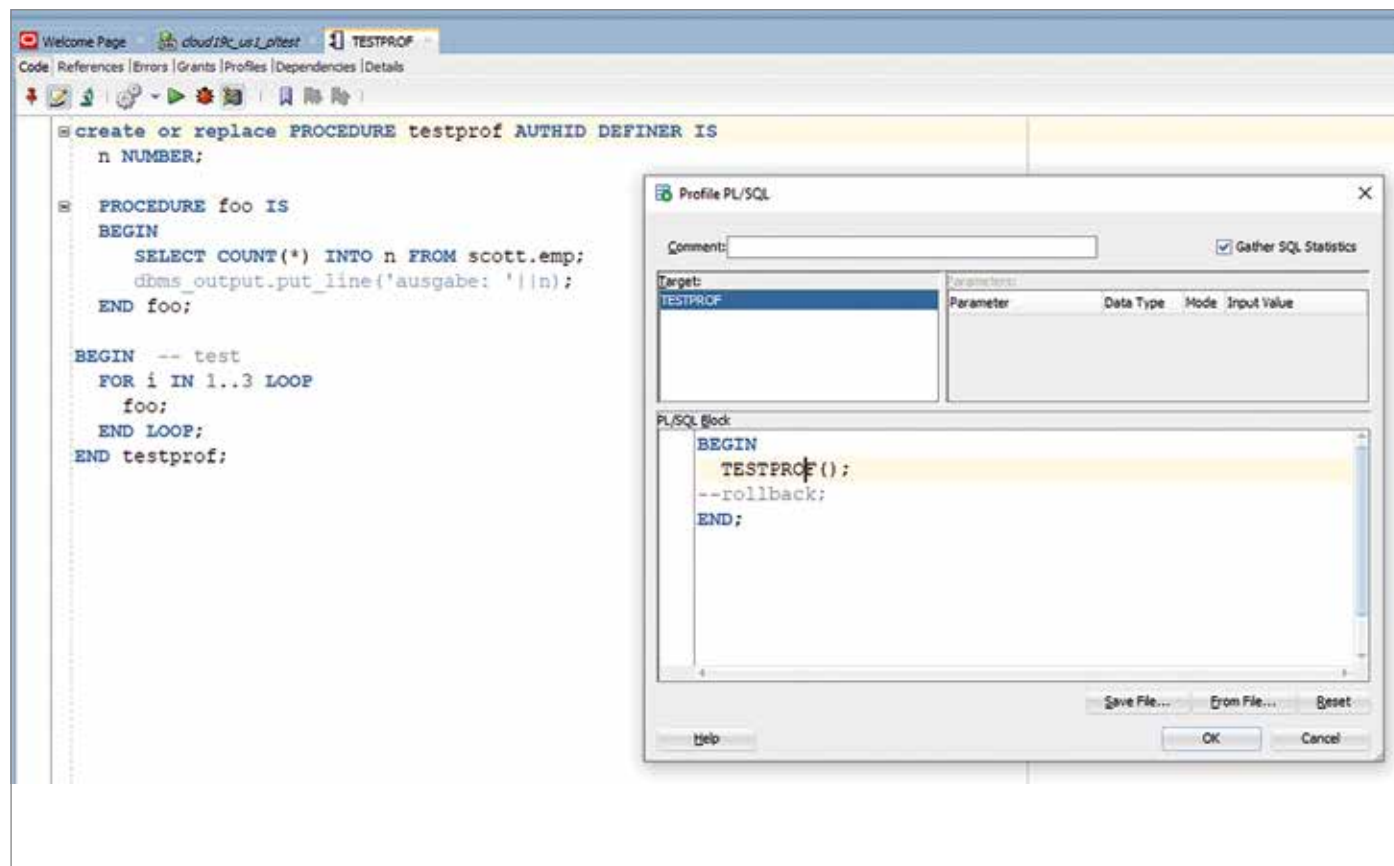


Abbildung 2: PL/SQL-Profiler-Nutzung im SQL Developer (Quelle: Oracle)

243755.1 PROFILER 11.4.4.7 Report: profiler_US1_de_19.0.0.0.0_20211112_145038.html

RUNID : 1
 Owner : PLTEST
 Date : 2021-11-12/10:40:37
 Comment : 12-NOV-21
 Total Time: 24.494000000 (seconds)
 RDEMS : 19.0.0.0.0
 Platform :

- [Top 10 Lines as per Total Time](#)
- [Top 10 Profiled PL/SQL Libraries](#)
- [All Profiled PL/SQL Libraries](#)

Top 10 Lines as per Total Time

| Top | Lib # | Type | Owner | Name | Line # | Times Line Exec | Total Time (seconds) | Min Time (seconds) | Max Time (seconds) | Line Text |
|-----|-------|-----------------|--------|----------|--------|-----------------|----------------------|--------------------|--------------------|--|
| 1 | 3 | PROCEDURE | PLTEST | TESTPROF | 6 | 3 | 0.002265007 | 0.000012743 | 0.002224695 | SELECT COUNT(*) INTO n FROM scott.EMP; |
| 2 | 2 | ANONYMOUS BLOCK | | | 1 | 3 | 0.000421521 | 0.000000464 | 0.000418752 | |
| 3 | 3 | PROCEDURE | PLTEST | TESTPROF | 7 | 3 | 0.000045052 | 0.000000621 | 0.000042548 | dbms_output.put_line('ausgabe: ' n); |
| 4 | 4 | ANONYMOUS BLOCK | | | 1 | 2 | 0.000024532 | 0.000001199 | 0.000021645 | |
| 5 | 3 | PROCEDURE | PLTEST | TESTPROF | 1 | 1 | 0.000002222 | 0.000002222 | 0.000002222 | PROCEDURE testprof AUTHID DEFINER IS |
| 6 | 3 | PROCEDURE | PLTEST | TESTPROF | 4 | 0 | 0.000001077 | 0.000000206 | 0.000000457 | PROCEDURE foo IS |
| 7 | 1 | ANONYMOUS BLOCK | | | 1 | 1 | 0.000000904 | 0.000000904 | 0.000000904 | |
| 8 | 3 | PROCEDURE | PLTEST | TESTPROF | 11 | 4 | 0.000000860 | 0.000000092 | 0.000000378 | FOR i IN 1..3 LOOP |
| 9 | 3 | PROCEDURE | PLTEST | TESTPROF | 8 | 3 | 0.000000845 | 0.000000100 | 0.000000519 | END foo; |
| 10 | 3 | PROCEDURE | PLTEST | TESTPROF | 12 | 3 | 0.000000446 | 0.000000071 | 0.000000266 | foo; |
| | | TOTAL | | | | | 0.002762466 | | | |

#1 Top Consumer

Displays 10 lines before and after #1 top consumer.

| Lib # | Type | Owner | Name | Line # | Times Line Exec | Total Time (seconds) | Min Time (seconds) | Max Time (seconds) | Line Text |
|-------|-----------|--------|----------|--------|-----------------|----------------------|--------------------|--------------------|--|
| 3 | PROCEDURE | PLTEST | TESTPROF | 1 | 1 | 0.000002222 | 0.000002222 | 0.000002222 | PROCEDURE testprof AUTHID DEFINER IS |
| 3 | PROCEDURE | PLTEST | TESTPROF | 2 | | | | | n NUMBER; |
| 3 | PROCEDURE | PLTEST | TESTPROF | 3 | | | | | |
| 3 | PROCEDURE | PLTEST | TESTPROF | 4 | 0 | 0.000001077 | 0.000000206 | 0.000000457 | PROCEDURE foo IS |
| 3 | PROCEDURE | PLTEST | TESTPROF | 5 | | | | | BEGIN |
| 3 | PROCEDURE | PLTEST | TESTPROF | 6 | 3 | 0.002265007 | 0.000012743 | 0.002224695 | SELECT COUNT(*) INTO n FROM scott.EMP; |
| 3 | PROCEDURE | PLTEST | TESTPROF | 7 | 3 | 0.000045052 | 0.000000621 | 0.000042548 | dbms_output.put_line('ausgabe: ' n); |
| 3 | PROCEDURE | PLTEST | TESTPROF | 8 | 3 | 0.000000845 | 0.000000100 | 0.000000519 | END foo; |
| 3 | PROCEDURE | PLTEST | TESTPROF | 9 | | | | | |
| 3 | PROCEDURE | PLTEST | TESTPROF | 10 | | | | | BEGIN -- test |
| 3 | PROCEDURE | PLTEST | TESTPROF | 11 | 4 | 0.000000860 | 0.000000092 | 0.000000378 | FOR i IN 1..3 LOOP |
| 3 | PROCEDURE | PLTEST | TESTPROF | 12 | 3 | 0.000000446 | 0.000000071 | 0.000000266 | foo; |
| 3 | PROCEDURE | PLTEST | TESTPROF | 13 | | | | | END LOOP; |
| 3 | PROCEDURE | PLTEST | TESTPROF | 14 | 1 | 0.000000375 | 0.000000375 | 0.000000375 | END testprof; |
| 3 | PROCEDURE | PLTEST | TESTPROF | 15 | | | | | |
| 3 | PROCEDURE | PLTEST | TESTPROF | 16 | | | | | |

#2 Top Consumer

Displays 10 lines before and after #2 top consumer.

| Lib # | Type | Owner | Name | Line # | Times Line Exec | Total Time (seconds) | Min Time (seconds) | Max Time (seconds) | Line Text |
|-------|-----------------|-------|------|--------|-----------------|----------------------|--------------------|--------------------|-----------|
| 2 | ANONYMOUS BLOCK | | | 1 | 3 | 0.000421521 | 0.000000464 | 0.000418752 | |
| 2 | ANONYMOUS BLOCK | | | 2 | | | | | |
| 2 | ANONYMOUS BLOCK | | | 3 | | | | | |
| 2 | ANONYMOUS BLOCK | | | 4 | | | | | |
| 2 | ANONYMOUS BLOCK | | | 5 | | | | | |
| 2 | ANONYMOUS BLOCK | | | 6 | | | | | |
| 2 | ANONYMOUS BLOCK | | | 7 | | | | | |
| 2 | ANONYMOUS BLOCK | | | 8 | | | | | |
| 2 | ANONYMOUS BLOCK | | | 9 | | | | | |
| 2 | ANONYMOUS BLOCK | | | 10 | | | | | |
| 2 | ANONYMOUS BLOCK | | | 11 | | | | | |

Abbildung 3: profiler.sql Report – ein Ausschnitt (Quelle: Ulrike Schwinn)

```
SQL> start profiler.sql 1
  RUNID RUN_OWNER          RUN_DATE          RUN_COMMENT
-----
    1  PLTEST              12-NOV-21 10:40 12-NOV-21
...
PROFILER file has been created:
profiler_US1_de_19.0.0.0.0_20211112_145038.html.
```

Listing 11: Beispiel 11: Nutzung von profiler.sql

Beispiel auf die View DBA_SOURCE zuzugreifen, um damit fehlerfrei zu arbeiten.

Das Ergebnis sieht dann wie in *Abbildung 3* dargestellt aus.

Fazit

Immer, wenn die Frage nach PL/SQL-Performance aufkommt, sollte die Verwendung von PL/SQL Profiler in Betracht gezogen werden. Die Technologie dazu steht schon seit langer Zeit in der Oracle-Datenbank zur Verfügung. Vorinstalliert und ohne großes vorangegangenes Setup kann man sofort damit starten. Sogar wenn die Zugriffsrechte keinen Zugriff auf ein logisches Datenbank-Directory zulassen, gibt es eine Lösung, um den PL/SQL Profiler verwenden zu können. Daher ist meine Empfehlung, einfach einmal ausprobieren, Reports erzeugen und für die PL/SQL-Analyse verwenden!

Links und weitere Informationen

- [1] Database Development Guide: Kapitel 15 Using the PL/SQL Hierarchical Profiler
- [2] Youtube: Finding Slow PL/SQL: Use the Profilers
- [3] Blog von Jeff Smith: SQL Developer & the PL/SQL Hierarchical Profiler
- [4] My Oracle Support Note Doc ID 243755.1
- [5] Blog: <https://blogs.oracle.com/coretec/post/oracle-plsql-analyse-immer-mit-profiler>

Über die Autorin

Ulrike Schwinn studierte Mathematik und ist bei der Firma Oracle in München als Data-Management-Expertin tätig. In ihrer Funktion berät sie Kunden in Fragen der neuesten Oracle-Datenbank- und -Cloud-Technologie, unterstützt sie beim Einsatz von neuen Funktionen und nimmt an PoCs teil. Sie veröffentlicht

regelmäßig Blogpostings in deutscher und englischer Sprache, ist im Redaktionsteam der „Oracle-Datenbanken – Monthly News“ und ist als Referentin auf IT-Kongressen vertreten.

Profile und Blogs:

<https://www.linkedin.com/in/ulrikeschwinn/>
<https://twitter.com/uschwinn>
https://www.xing.com/profile/Ulrike_Schwinn
<https://blogs.oracle.com/coretec>



Ulrike Schwinn
Ulrike.Schwinn@oracle.com



Weitgehend unbekannte PL/SQL-Fähigkeiten entdeckt!

Jürgen Sieben, ConDeS

Austin, Texas: Berichten zufolge sind bislang weitgehend unbekannte, mächtige PL/SQL-Fähigkeiten entdeckt worden, die Programmierkonstrukte erlauben, die bislang unmöglich schienen. Wie interne Quellen belegen, sind diese Fähigkeiten bereits seit vielen Jahren vorhanden, aber dennoch kaum bekannt. Oracle-Mitarbeiter wundern sich: „Wir waren selbst überrascht, dass bei uns so wenige PL/SQL-Entwickler darauf gekommen sind...“

Nun ja. Aber es stimmt schon, ich bin zwar kein Oracle-Mitarbeiter, aber auch immer wieder überrascht, wie selten die folgenden, mächtigen und einfach zu verwendenden Möglichkeiten in der Programmierung von PL/SQL genutzt werden und wie oft selbst erfahrene PL/SQL-Entwickler zurückschrecken, wenn man dieses Thema anspricht. Daher möchte

ich es einmal andersherum versuchen und Sie bitten, sich folgenden Code anzusehen und zu überlegen, wie er programmiert wurde (*siehe Listing 1*).

Sieht nicht so ungewöhnlich aus? Naja, nur auf den ersten Blick. Zitieren wir noch einmal aus dem Artikel: „Forschende haben das aufgefundene Codebeispiel untersucht und festgestellt, dass

offensichtlich keine Limitierung bei der Anzahl der Parameter, die der Funktion `params` übergeben werden können, vorliegt. Zudem scheint es egal zu sein, welchen Datentyp man übergibt. Lediglich beim Datentyp `boolean` stellten die Forschenden fest, dass ein Fehler ausgelöst wurde. Der Grund hierfür ist allerdings weiterhin unklar ...“

Versuchen Sie einmal, diesen Code in PL/SQL zu schreiben. Sie stellen fest, dass dies gar nicht oder nur schwer geht. Drei Besonderheiten scheinen zu existieren: Zum einen ist die Zahl der Parameter beim Aufruf der Funktion(?) params unbegrenzt. Das kennen wir von PL/SQL-Tabellen, diese jedoch benötigen einen Schlüssel, werden also in der Form `my_table(i) := my_value` belegt.

Das zweite betrifft die Überladung einer Funktion für mehrere Datentypen. Das kennen wir aus Packages, aber hier wird kein Package verwendet.

Schließlich fällt uns auf, dass offensichtlich der Funktion param zwei Parameter übergeben werden, die wohl auch im Ergebnis als Name-Wert-Paar Verwendung finden und in eine Record-artige Struktur münden. Das könnte doch diese neue Möglichkeit, einen Record mit Werten zu belegen, sein, oder? Nein, dagegen spricht, dass dem gleichen „Record“ unterschiedliche Datentypen zugeordnet werden können, was mit PL/SQL-Records nicht geht.

Ich bin sicher, es gäbe einige Aufregung, wenn so ein Code in PL/SQL möglich würde. Es gibt so viele Stellen, an denen wir diese Fähigkeiten gebrauchen könnten. Und nun wird behauptet, so etwas gäbe es schon seit vielen Jahren und kaum jemand kennt es? Ja, genau das.

Das „Geheimnis“ liegt darin, dass wir die „größeren Brüder“ der PL/SQL-Konstrukte PL/SQL-Tabelle und Record verwendet haben. Die verwendeten Typen gibt es seit Version 8, sicher und ausgereift seit Version 9, also seit über 20 Jahren. Beginnen wir mit dem Problem, eine beliebig lange Liste von Parametern an eine Methode zu übergeben. Um das Prinzip zu verdeutlichen (und Ihnen ein sehr nützliches Konstrukt an die Hand zu geben), beginnen wir mit einem einfachen Collection-Datentyp TABLE, der lediglich in der Lage ist, eine Liste von Zeichenketten zu speichern. Er wird in SQL auf denkbar einfache Weise erzeugt (siehe Listing 2).

Wollen wir mit dieser Collection arbeiten, können wir eine konkrete Liste von Zeichenketten erzeugen, indem wir einer Funktion mit dem Namen des Typs eine Liste von Zeichenketten übergeben (siehe Listing 3).

Da einer PL/SQL-Methode jeder, in SQL erlaubte, Datentyp übergeben werden kann, gilt dies insbesondere auch für

```
declare
  l_var varchar2(255 byte);
  l_clob clob;
  l_num number;
  l_now date;
  l_xml xmltype;
begin
  -- ... do something
  my_pkg.log_state(
    p_params => params(
      param('Var', l_var),
      param('CLOB', l_clob),
      param('Num', l_num),
      param('Now', l_date),
      param('XML', l_xml));
end;
/
```

Listing 1: Beispielcode

```
create or replace type char_table
as
table of varchar2(4000 byte);
/
```

Listing 2: Syntax eines TABLE-Datentyps in SQL

```
declare
  l_string_list char_table;
begin
  l_string_list := char_table('A', 'B', 'C');
end;
/
```

Listing 3: Verwendung einer TABLE-Collection

```
SQL> create type param as object(
  2   name varchar2(128 byte),
  3   value varchar2(4000 byte)
  );
  5 /
Typ wurde erstellt.
```

Listing 4: Erstellen des Objekts PARAM

```
SQL> create type params as table of param;
  2 /
Typ wurde erstellt.
```

Listing 5: Erzeugen einer Collection auf Basis eines Objekts

den Typ CHAR_TABLE. Damit können wir beliebig viele Parameter übergeben und bereits damit sinnvolle Dinge tun. Stel-

len Sie sich eine Methode BULK_REPLACE vor, die einen ersten Parameter mit dem Text, in dem mehrere Ersetzungen durch-

geführt werden sollen, übergibt und eine CHAR_TABLE mit paarweise übergebenen Anknern und Ersetzungszeichenfolgen.

Mit einer TABLE können wir also das erste Problem lösen, doch reicht das für unser Beispiel aus *Listing 1* noch nicht aus, denn hier werden offensichtlich keine einfachen Zeichenketten übergeben, sondern irgendetwas, das PARAM genannt wird. PARAM ist ein Objekt, was in der ein-

```
my_pkg.log_state(  
  p_params => params(  
    param('Var', substr(l_var, 1, 4000)),  
    param('CLOB', dbms_lob.substr(l_clob, 4000, 1)),  
    param('Num', to_char(l_num)),  
    param('Now', to_char(l_date, 'dd.mm.yyyy')),  
    param('XML', l_xml.getStringVal()));
```

Listing 6: Improvisierter Aufruf

```
create or replace type param force as object(  
  /** Type to be used as a key value pair to pass parameter or variable values along with their value */  
  param varchar2(128 byte),  
  value varchar2(4000 byte),  
  /**  
    Function: param  
    Constructor function for values of type VARCHAR2  
  */  
  constructor function param(  
    self in out nocopy param,  
    p_param in varchar2,  
    p_value in varchar2)  
    return self as result,  
  /**  
    Function: param  
    Constructor function for values of type CLOB  
  */  
  constructor function param(  
    self in out nocopy param,  
    p_param in varchar2,  
    p_value in clob)  
    return self as result,  
  /**  
    Function: param  
    Constructor function for values of type DATE  
  */  
  constructor function param(  
    self in out nocopy param,  
    p_param in varchar2,  
    p_value in date)  
    return self as result,  
  /**  
    Function: param  
    Constructor function for values of type NUMBER  
  */  
  constructor function param(  
    self in out nocopy param,  
    p_param in varchar2,  
    p_value in number)  
    return self as result,  
  /**  
    Function: param  
    Constructor function for values of type TIMESTAMP  
  */  
  constructor function param(  
    self in out nocopy param,  
    p_param in varchar2,  
    p_value in timestamp)  
    return self as result,  
  /**  
    Function: param  
    Constructor function for values of type TIMESTAMP_WITH_TIME_ZONE  
  */
```

```

constructor function param(
  self in out nocopy param,
  p_param in varchar2,
  p_value in timestamp with time zone)
return self as result,
/**
  Function: param
  Constructor function for values of type XMLTYPE
*/
constructor function param(
  self in out nocopy param,
  p_param in varchar2,
  p_value in xmltype)
return self as result
);

create or replace type body param
as
constructor function param(
  self in out nocopy param,
  p_name in varchar2,
  p_value in varchar2)
return self as result
as
  C_EXTENSION constant varchar2(10 byte) := '...';
begin
  self.name := substrb(p_name, 1, 128);
  self.value := substr(p_value, 1, 2000);
  if self.value != p_value then
    self.value := self.value || C_EXTENSION;
  end if;
  return;
end;

constructor function param(
  self in out nocopy param,
  p_name in varchar2,
  p_value in clob)
return self as result
as
  C_EXTENSION constant varchar2(10 byte) := '...';
begin
  self.name := substrb(p_name, 1, 128);
  self.value := dbms_lob.substr(p_value, 2000, 1);
  if dbms_lob.getlength(p_value) > 2000 then
    self.value := self.value || C_EXTENSION;
  end if;
  return;
end;

constructor function param(
  self in out nocopy param,
  p_name in varchar2,
  p_value in date)
return self as result
as
begin
  self.name := substrb(p_name, 1, 128);
  if p_value > trunc(p_value) then
    self.value := to_char(p_value, 'YYYY-MM-DD HH24:MI:SS');
  else
    self.value := to_char(p_value, 'YYYY-MM-DD');
  end if;
  return;
end;

constructor function param(
  self in out nocopy param,

```


fachsten Form grob einem Record in PL/SQL entspricht. Wieder erzeugen wir einen Typ in SQL; das hat – nebenbei – den großen Vorteil, dass dieser Typ auch aus SQL verwendbar ist, was mit einem PL/SQL-Record ja nicht geht (siehe Listing 4).

Der Unterschied zur Deklaration eines Records in PL/SQL liegt wirklich nur in den Schlüsselworten `CREATE` und `OBJECT`, ansonsten ist alles gleich. Dieses Objekt können wir nun als Payload einer `TABLE-Collection` verwenden (siehe Listing 5).

Das Objekt `PARAM` wird ebenfalls über eine Methode mit dem gleichen Namen verwendet, nur dass hier positionell die beiden Attribute `NAME` und `VALUE` übergeben werden. Und der `Collection PARAMS` wird nun eine kommaseparierte Liste dieser Funktionsaufrufe übergeben.

Nun kann der Code aus Listing 1 bereits aufgerufen werden, er wird allerdings noch nicht vollständig funktionieren, weil keine Überladung für unterschiedliche Datentypen existiert. Wir könnten den Aufruf dennoch machen, denn `PARAM` kann bereits in dieser Form positionell die beiden Attribute belegen, wie das neuerdings auch bei Records möglich ist, allerdings benötigen wir nun Konvertierungsfunktionen (siehe Listing 6).

Schön ist das nicht, denn der Code ist nicht nur lang und unübersichtlich, sondern zudem noch fehleranfällig. Wenn wir zum Beispiel die Methode `getStringVal()` auf ein leeres XML aufrufen, wird ebenso ein Fehler geworfen, wie wenn wir (in einer UTF-8-basierten Datenbank) mehr als 4000 Zeichen, von denen mindestens eines einen Umlaut darstellt, übergeben.

Wir benötigen also eine gewisse Intelligenz. Wie wir diese Intelligenz implementieren, ist meiner Meinung nach sehr elegant gelöst. Wir erzeugen eine Methode, die aufgrund der übergebenen Parameter die Attribute des Objekts mit Werten belegt. Ohne explizite Deklaration dieser Methoden existiert bereits eine Standardmethode für diesen Zweck, die einfach die Attribute in der deklarierten Reihenfolge mit Werten belegt, und diese hatten wir im Aufruf oben auch verwendet. Wir nennen diese Funktion einen *Konstruktor* – weil die Methode zum Erstellen verwendet wird, nennen sie genauso wie den Typ und binden diesen Konstruktor in die Deklaration des Objekts ein (siehe Listing 8).

```
create type param as object(
  name varchar2(128 byte),
  value varchar2(4000 byte),
  constructor function param(
    self in out nocopy param
    p_name in varchar2,
    p_value in varchar2)
    return self as result
);
/
```

Listing 8: Erweiterung des Typs um einen Konstruktor

```
create or replace type utility_core as object(
  -- Dummy, ist aus syntaktischen Gründen erforderlich
  foo char(1 byte),
  static function basis_a(
    p_n number)
    return number,
  static function basis_b(
    p_n number)
    return number
) not final not instantiable;
/
```

Listing 9: Erster Teil: Abstrakter Typ als Ersatz für ein Package

Wenn Sie einmal für den Moment großzügig die in Teilen ungewöhnliche Syntax und den Parameter `self` übersehen, erkennen Sie, dass eine große Ähnlichkeit zu einer Package-Spezifikation existiert. Wir hätten, wenn dies ein Package wäre, zwei globale Variablen und eine Funktion.

Wenn Sie die Ähnlichkeit zu einer Package-Spezifikation zugrunde legen, ist auch klar, wie wir die Überladungen für unterschiedliche Datentypen erreichen: Indem wir einfach mehrere Konstruktormethoden mit unterschiedlichen Parametertypen verwenden.

Eine Package-Spezifikation macht noch kein Package, und Gleiches gilt auch für ein Objekt. Wenn wir Konstruktormethoden deklarieren, müssen wir diese auch implementieren. Und analog zu einem Package wird dies ebenfalls bei Objekten durch einen Objekt-Body gemacht, der vollständig in PL/SQL implementiert wird und syntaktisch beinahe ebenso aufgebaut ist. In Listing 7 finden Sie das vollständige Listing der Typen für unterschiedliche Datentypen.

Mit diesem Wissen ausgestattet, wird nun auch das Geheimnis um den Fehler beim Datentyp `BOOLEAN` klar: SQL kennt

diesen Typ nicht. Das Verfahren ist daher auf die Datentypen limitiert, die in SQL bekannt sind.

Aber Objekte sind so schwierig in der Handhabung!

Diesen Einwand höre ich oft. Objekte verhalten sich in einigen Punkten anders und ungewohnt im Vergleich zu PL/SQL-Variablen. Wenn man beispielsweise einen Typ durch eine neue Version ersetzen oder einfach löschen möchte, ist dies nicht so einfach möglich. Dass dies als problematisch wahrgenommen und dem Objekt angelastet wird, entspringt einem Missverständnis: Typen in SQL sind nicht so leichtgewichtig wie Variablen in PL/SQL, weil sie Datenbankobjekte darstellen, die persistente Daten repräsentieren können. So kann ein Objekt als Datentyp einer Tabellenspalte verwendet werden. Und so, wie Sie auch Tabellen nicht einfach mit einer `replace`-Anweisung ersetzen können, geht dies auch mit Typen nur bedingt.

Wenn Sie eine Tabelle löschen möchten, die durch referenzielle Integrität mit anderen Tabellen verbunden ist, gestat-

```

create or replace type utility under utility_core(
  static function erweiterung_c(
    p_n number)
  return number
) final not instantiable;
/

```

Listing 10: Zweiter Teil: Abstrakter Typ für die Erweiterungen

```

SQL> create or replace type meldung force
  2   authid definer
  3   as object(
  4     text varchar2(4000 byte)
  5   );
  6   /
Typ wurde erstellt.

SQL> create or replace type log_modul force
  2   authid definer
  3   as object(
  4     foo char(1 byte),
  5     member procedure log(
  6       self in out nocopy log_modul,
  7       p_meldung in meldung)
  8   ) not final not instantiable;
  9   /
Typ wurde erstellt.

SQL> create or replace type body log_modul
  2   as
  3   member procedure log(
  4     self in out nocopy log_modul,
  5     p_meldung in meldung)
  6   as
  7   begin
  8     null;
  9   end log;
10 end;
11 /
Type Body wurde erstellt

```

Listing 11: Basisobjekt für ein Meldungsmodul

tet Ihnen die Datenbank dies nicht. Da müssen Sie dann schon deutlicher werden, etwa durch die Anweisung `CASCADE CONSTRAINTS`. Ebenso geht es auch bei Typen. Hier werden wir im Regelfall deutlich durch die Klausel `FORCE`, die den Typ löschen wird, aber natürlich auch ungültige Typen, Packages oder Tabellen hinterlassen kann.

Wenn wir, wie in unserem Beispiel, die Typen nur zur Programmierung in PL/SQL verwenden (wozu ich zunächst raten würde) und nicht als Spaltentyp einer Tabelle, dann können wir im Regelfall, ohne negative Konsequenzen, diese Klausel verwenden und alle un-

gültigen Objekte rekompilieren, nachdem wir die neue Variante des Typs erstellt haben. Allerdings existieren auch Methoden für eine höflichere, weniger rigorose Pflege der Typen, denn sie können auch durch `ALTER TYPE`-Anweisungen migriert werden. Die Besprechung dieser Optionen ist aber außerhalb des Fokus dieses Artikels.

Ein weiteres Beispiel

Diesmal beginnen wir mit dem Problem, das wir lösen möchten. Das Szenario ist mir in mehreren Projekten begegnet und

auch ein Problem für GitHub-Projekte, die ich betreue: Wir entwickeln in einem Projekt eine Standardumgebung für die Erstellung von APEX-Anwendungen, die nicht nur in der eigenen Abteilung, sondern auch in anderen Abteilungen des Unternehmens verwendet wird. Teil dieser Umgebung ist ein APEX-Utility Package, in dem allgemeine Funktionalität zu APEX gekapselt wird. Über die Zeit hat sich in diesem Utility Package aber auch Abteilungs-spezifischer Code eingefunden, einfach weil es praktisch ist, diese Funktionen in einem Package zu bündeln.

Steht nun aber eine erneute Auslieferung der Standardumgebung an, müssen diese speziellen Utilities wieder entfernt werden, weil sie auf Packages zugreifen, die in anderen Abteilungen nicht vorhanden sind. Analog haben die anderen Abteilungen in dieses Utility-Package ihre dort benötigten Erweiterungen integriert. Möchten diese Abteilungen nun den neuen Standard unterstützen, müsste das Package von Hand analysiert und angepasst werden. Geht das nicht einfacher und stabiler?

Die ernüchternde Antwort lautet: Nein, PL/SQL verfügt nicht über Mittel, solche Fragestellungen umzusetzen. Doch wieder können wir Objekte benutzen, um PL/SQL um diese Fähigkeiten zu erweitern. Der Trick besteht darin, die Objekte aufeinander aufbauen zu lassen.

Für unser Beispiel nehmen wir an, dass ein Package als Basis die Methoden `basis_a` und `basis_b` anbieten soll. Die eigene Erweiterung stellt die Methode `erweiterung_c` dar. Anstatt nun ein Package zu erstellen, das diese Hilfsmethoden bereitstellt, verwenden wir Objekte für diesen Zweck. Der Trick besteht darin, dass wir zunächst ein Objekt für die Basisfunktionen erstellen (siehe Listing 9).

Die Klauseln nach Ende der Typdeklaration muten seltsam an. Sie legen fest, dass für diesen Typ keine Konstruktormethode aufgerufen werden darf. Das klingt zunächst nicht sehr logisch, weil man dann ja offensichtlich nichts mit dem Typ anfangen kann, aber wir haben durch die Klausel `static` gegengesteuert, was bedeutet, dass diese Methoden aufgerufen werden können, auch ohne vorher eine Konstruktormethode ausgeführt zu haben. Dadurch werden diese Funktionen auf gleiche Weise verwendbar wie eine Package-Methode in PL/SQL.

```
SQL> create or replace type log_console force under log_modul(
  2   constructor function log_console(
  3     self in out nocopy log_console)
  4     return self as result,
  5   overriding member procedure log(
  6     self in out nocopy log_console,
  7     p_meldung in meldung)
  8   ) final instantiable;
  9   /
Typ wurde erstellt.
```

```
SQL> create or replace type body log_console
  2   as
  3
  4     constructor function log_console(
  5     self in out nocopy log_console)
  6     return self as result
  7   as
  8   begin
  9     return;
 10   end log_console;
 11
 12   overriding member procedure log(
 13     self in out nocopy log_console,
 14     p_meldung in meldung)
 15   as
 16   begin
 17     dbms_output.put_line(p_meldung.text);
 18   end log;
 19
 20   end log_console;
 21   /
Type Body wurde erstellt
```

Listing 12: Konkretes Ausgabemodul LOG_CONSOLE

```
SQL> select type_name
  2     from user_types
  3    where supertype_name = 'LOG_MODUL';

TYPE_NAME
-----
LOG_CONSOLE
```

Listing 13: Ermittlung der vorhandenen Ausgabemodule

So, wie es jetzt aussieht, haben wir einfach nur einen Ersatz für ein Package geschaffen, das nicht mehr kann als ein normales Package. Die Package-Implementierung ist wieder trivial und entspricht grob einem Package Body. Was also ist hier der Mehrwert? Der kommt nun, denn nun erstellen wir einen zweiten Typ mit den Erweiterungen (siehe Listing 10).

Wir haben einfach einen zweiten Typ mit den Erweiterungsmethoden erstellt. Natürlich wird dieser Typ ebenfalls noch in einem Body implementiert. Und das soll es gewesen sein? Wann kommt denn nun die Pointe?

Nun ja, die ist schon vorbei. Die entscheidende Klausel lesen Sie hinter dem `create or replace`, nämlich die Klausel `under utility_core`. Ich übersetze das einmal mit „basiert auf“, wir bauen also einen Typ, der auf `UTILITY_CORE` basiert. Das bedeutet, dass dieser Typ bereits alle Methoden umfasst, die `UTILITY_CORE` implementiert, und zusätzlich die eigenen Erweiterungen.

Jetzt dürfte der Groschen fallen: Wenn wir nun einen neuen Standard ausliefern möchten, werden wir lediglich `UTILITY_CORE` neu ausliefern. Dann kommt die Sache mit dem `FORCE`, Sie erinnern sich, und alles wird neu kompiliert. Das Ergeb-

nis: Die neu hinzugekommenen oder geänderten Funktionen aus `UTILITY_CORE` sind nun verfügbar, die eigenen Erweiterungen wurden nicht angetastet. Zudem sind sämtliche Utility-Methoden über das Objekt `UTILITY` verfügbar, es wirkt also so, als ob es nur ein Package mit Hilfsmethoden gäbe.

Und noch etwas komplexer

Wie setzen wir folgende Forderung um: Erschaffen Sie einen Code, der eine Meldung in ein beliebiges Ausgabemedium (Tabelle, Datei, Mail etc.) schreibt. Stellen Sie sicher, dass es möglich ist, nachträglich weitere Ausgabekanäle zu erstellen, ohne den Code zu ändern.

Das klingt seltsam, doch gibt es solche Anforderungen durchaus häufig. Als Beispiel implementiert APEX einen Erweiterungsmechanismus, der beliebige Funktionalität an APEX anbinden kann. Natürlich kann, darf der APEX-Code selbst nicht angetastet werden. Wie setzt man so etwas normalerweise um? Bei APEX-Plug-ins (dem oben beschriebenen Szenario) funktioniert dies mit einer Konvention: Es ist eine Gruppe von Methoden zu implementieren, die eine genau definierte Signatur besitzen. Das ist instabil und kann insbesondere nicht durch den Compiler geprüft werden. Ob eine Methode also die korrekte Signatur besitzt, merken Sie erst zur Laufzeit.

Auch hier, Sie ahnen es, gibt es einen stabileren Weg. Die Idee ist, dass Objekte, die aufeinander basieren, nicht nur die Funktionen wiederverwenden können, die die Basisobjekte vereinbart haben, sondern diese auch mit einer eigenen Implementierung überschreiben können. Damit dies funktioniert, können sie allerdings nicht statisch sein, sondern benötigen eine Konstruktorfunktion.

Nun kommt die Idee: Wir erschaffen einen Basistyp mit allen Methoden, die wir für die Ausgabe einer Meldung benötigen. Um das Beispiel einfach zu halten, soll nur eine LOG-Methode vorhanden sein, die ein Objekt vom Typ `MELDUNG` als Parameter erhält und verarbeiten kann. Wie `MELDUNG` genau definiert ist, ist für das folgende Beispiel in Listing 11 egal, es könnte einen Text, einen Schweregrad, Backtrace-Information und beliebige weitere Information enthalten.

```

SQL> set serveroutput on
SQL> declare
  2     l_modul log_modul;
  3     begin
  4     l_modul := log_console();
  5     l_modul.log(Meldung('Hallo Welt'));
  6     end;
  7     /
Hallo Welt

PL/SQL-Prozedur erfolgreich abgeschlossen.

```

Listing 14: Verwendung des Ausgabemoduls

```

declare
  l_modul log_modul;
  cursor module_cur is
    select type_name
      from user_types
     where supertype_name = 'LOG_MODUL';
begin
  for m in module_cur loop
    execute immediate
      'begin :x := ' || m.type_name || '(); end;
      using out l_modul;
    l_modul.log(Meldung('Hallo Welt'));
  end loop;
end;
/

```

Listing 15: Beispielcode zum Arbeiten mit mehreren Ausgabemodulen

Dieses Modul wird als NOT INSTANTIABLE definiert, es kann also keine Konstruktorfunktion für dieses Objekt aufgerufen werden, es fungiert lediglich als Platzhalter für die „eigentlichen“ Logmodule. Die Implementierung dieses Typs erfolgt als Stub, also als leere Funktionshülle.

Ein Objekt, das nun tatsächlich eine Meldung ausgeben kann, basiert auf diesem Objekt, ist aber mit einer Konstruktormethode versehen. In Listing 12 sehen Sie ein Beispiel für das Ausgabemodul LOG_CONSOLE.

Warum ist dies nun besser? Zum einen kann der Compiler prüfen, ob die Signatur der Methode LOG tatsächlich mit der des Basisobjekts übereinstimmt. Der Code wird invalide, falls dies nicht der Fall ist. Zudem kann nun mittels der View USER_TYPES dynamisch ermittelt werden, welche Ausgabemodule vorhanden sind (siehe Listing 13).

Eine Variable vom Typ LOG_MODUL kann problemlos mit allen konkreten Ausgabemodulen belegt werden (siehe Listing 14).

Möchten Sie eine Meldung auf alle installierten Ausgabemodule ausgeben, benötigen Sie lediglich eine Schleife über die in Listing 13 erstellte Abfrage und ein wenig dynamisches SQL (siehe Listing 15).

Fazit

Haben Sie bemerkt, dass ich die schlimmen Wörter „Objektorientierung“ oder „Vererbung“ nicht benutzt habe? Dies war Absicht. Ich habe den Verdacht, dass allein schon die Nennung dieser Begriffe die Diskussion in eine bestimmte Richtung lenkt, die davon ablenkt, dass mit diesen Mitteln ohne großen Aufwand leistungsfähige Dinge in PL/SQL umgesetzt werden können.

Ich nutze Objekte gern, um etwas zu implementieren, was in anderen Programmiersprachen als „Interface“ bekannt ist: die Zusicherung durch den Compiler, dass eine Gruppe von Funktionen mit einer definierten Signatur in jedem Fall vorhanden sein muss. Objekte stabilisieren das Prinzip „Coding by

Convention“, da auch der PL/SQL-Compiler Verstöße erkennen kann. Und besser noch: Wenn das Basisobjekt für die Methoden ein Standardverhalten implementiert, müssen die abgeleiteten Typen diese Methoden nur überschreiben, wenn sie eine abweichende Implementierung benötigen. Ansonsten lassen sich die Methoden immer aufrufen, werden aber einfach an die Basisobjekte zur Ausführung weitergereicht. Allein dieser Komfort sollte diese Optionen in einem angenehmen Licht erscheinen lassen.



Jürgen Sieben
j.sieben@condes.de



Testen mit *utPLSQL* – Leicht gemacht mit *SQL Developer* Developer

Philipp Salvisberg, Trivadis – Part of Accenture

Heutzutage dreht sich alles um Automatisierung. Das Testen von Software ist keine Ausnahme. Nach einer Einführung schauen wir anhand einer fiktiven User-Story die Erstellung und Ausführung von automatisierten Tests mit dem Oracle SQL Developer an, und zwar unabhängig von einer CI/CD-Umgebung. Dies bedeutet, dass wir Tests ausführen können, bevor die Änderungen des Datenbank-Codes in einem Version-Control-System sichtbar sind. Testen werden wir ein PL/SQL Package, zwei Datenbank-Trigger und eine View. Die vollständigen Codebeispiele finden Sie auf GitHub [1].

Was ist Test-Automation?

Die englische Wikipedia-Seite definiert Test-Automation frei übersetzt wie folgt: „Die Verwendung von Software separat von der zu testenden Software zur Steuerung der Testdurchführung und der Vergleich von den tatsächlichen und erwarteten Ergebnissen“ [6]. Entsprechen die tatsächlichen Ergebnisse den erwarteten Ergebnissen, dann sind die Tests erfolgreich. Das heißt, ein wichtiger Aspekt der Test-Automation ist die vorgängige Definition der erwarteten Ergebnisse. Erwartete Ergebnisse sind nichts anderes als Anforderungen. In anderen Worten, wir prüfen mit automatisierten Tests, ob die Anforderungen an eine Software erfüllt werden.

Warum brauchen wir automatisierte Tests?

Der Einsatz von agilen Methoden und die damit verbundenen kürzeren Release-Zyklen führen dazu, dass wir Software häufiger testen müssen. In einer CI/CD-Umgebung können automatisierte Tests direkt nach einem Commit in einem Version-Control-System ausgeführt werden. Durch diese kontinuierliche Prüfung ist der Zustand der Softwarequalität laufend verfügbar. Bei Abweichungen lassen sich die möglichen Ursachen durch die kürzeren Intervalle zwischen den Tests schneller identifizieren. Außerdem sind die Änderungen, seit dem letzten erfolgreichen Testdurchlauf, noch in den Köpfen der Entwickler präsent. All dies vereinfacht die Fehlerbehebung.

Wenn Software getestet wird, indem automatisierbare Tests halbautomatisch oder manuell ausgeführt werden, führt dies zu zusätzlichen Kosten, zu Lieferverzögerungen oder Qualitätsproblemen. Es ist schlicht und einfach nicht effizient, automatisierbare Aufgaben manuell durchzuführen.

Die Wiederholung von automatisierten Tests ist billig. Im Gegensatz zu manuellen Tests gibt es keinen Grund, Kompromisse einzugehen. Das ist besonders wichtig bei zentralen Komponenten einer Softwarelösung. Genau hier müssen wir sicherstellen, dass Änderungen keine ungewollten Auswirkungen auf andere Teile der Softwarelösung haben.

Haben automatisierte Tests auch Nachteile?

Am Ende ist ein automatisierter Test auch Software, die gewartet werden muss. Problematisch sind vor allem Tests, die nicht zuverlässig sind. Sogenannte „Flaky“-Tests, die manchmal fehlschlagen, aber wenn man sie oft genug wiederholt, das erwartete Ergebnis liefern. Solche Tests schaden schnell mehr, als sie nutzen. Je mehr Flaky-Tests es gibt, desto höher ist die Wahrscheinlichkeit, dass der CI-Job fehlschlägt, was manuelle Tätigkeiten nach sich zieht.

Was können wir in der Datenbank testen?

Grundsätzlich lässt sich jede in der Datenbank installierte Komponente einer Applikation automatisiert testen. Dies sind beispielsweise Object Types, Packages, Procedures, Functions, Triggers, Views und Constraints.

Für gewöhnliche Not-Null-Check-Constraints schreiben wir keine Tests, wir wollen ja nicht die Grundfunktionalitäten einer Datenbank prüfen. Allerdings kann es sinnvoll sein, einen komplexen Ausdruck eines Check-Constraint oder einer virtuellen Column zu testen. Eine View basiert auf einem SELECT-Befehl. SELECT-Befehle können selbst ohne PL/SQL Units in der WITH-Klausel recht komplex sein. Demzufolge haben wir auch das Bedürfnis zu prüfen, ob Views die erwarteten Ergebnisse liefern.

Die anderen erwähnten Objekttypen beinhalten PL/SQL, Java oder auch JavaScript. Es versteht sich von selbst, dass diese Komponenten mit automatisierten Tests getestet werden sollten. Dies gilt auch für Datenbank-Trigger, die nicht direkt ausgeführt werden können. Natürlich vereinfacht es das Testen, wenn die Logik der Datenbank-Trigger möglichst vollständig in PL/SQL Packages ausgelagert ist und Datenbank-Trigger nur noch einfache Aufrufe von Package Procedures beinhalten.

Was ist utPLSQL?

utPLSQL ist eine Testing Suite für Code in der Oracle-Datenbank und basiert auf Konzepten von anderen Unit Testing Frameworks wie JUnit und RSpec. Die *Abbildung 1* zeigt die Komponenten der utPLSQL-Suite.

Das Core Testing Framework wird in ein eigenes Schema der Oracle-Datenbank installiert. Alle Tabellen in diesem Schema haben temporären Charakter oder werden für Caching-Zwecke verwendet. Das heißt, neuere oder ältere Versionen von utPLSQL können jederzeit ohne Datenverluste installiert werden. Test-Suites sind PL/SQL Packages, die mit einem speziellen Kommentar „--%suite“ annotiert werden. Diese speziellen Kommentare werden als Annotationen bezeichnet und liefern utPLSQL alle Informationen zur Testausführung. utPLSQL verwendet verschiedene Konzepte, um die benötigten Daten effizient aus den Oracle Data Dic-

```
create table deptsal (
  deptno number(2, 0) not null
    constraint deptsal_pk primary key,
  dname varchar2(14) not null,
  sum_sal number(10, 2) not null,
  num_emps number(4, 0) not null,
  avg_sal number(7, 2)
    generated always as (
      case
        when num_emps != 0 then
          round(sum_sal / num_emps, 2)
        else
          0
        end
      ) virtual
);
```

Listing 1: Tabelle deptsal

tionary Views zu lesen, sodass Tests auch in Schemata mit mehr als 40.000 PL/SQL Packages ohne spürbare Zeitverzögerung ausgeführt werden können.

Die utPLSQL Plug-ins für SQL Developer von Oracle und PL/SQL Developer von Allround Automations unterstützen die Datenbank-Entwickler beim Erstellen, Ausführen und Debuggen von Tests. Code Coverage Reports liefern Informationen darüber, welcher Code durch Tests abgedeckt ist. Diese beiden Plug-ins stellt das utPLSQL-Team bereit. Quest und JetBrains positionieren utPLSQL als den Standard beim Testen von Code in einer Oracle-Datenbank und unterstützen utPLSQL direkt in ihren Werkzeugen TOAD und DataGrip.

utPLSQL kann mittels Reporter die Ergebnisse eines Testlaufs in beliebig vielen Ausgabeformaten zeitgleich aufbereiten. Dazu können der Command Line Client oder das Maven Plug-in verwendet werden. Das häufigste Ausgabeformat ist das JUnit-XML-Format, da es von den meisten Werkzeugen im Bereich CI/CD standardmäßig interpretiert werden kann. TeamCity und Team Foundation Server werden mit spezifischen Reportern unterstützt. Für Code Coverage stellt utPLSQL Reporter für SonarQube, Coveralls und Cobertura zur Verfügung. Dank des flexiblen Reporter-Konzepts lässt sich utPLSQL in jede CI/CD-Umgebung einbinden. Beispielsweise in Jenkins, Bamboo, Azure DevOps, Travis CI, GitLab, GitHub Actions und viele mehr.

Fallbeispiel und Lösungsansatz

Wir verwenden ein Schema redstack mit den bekannten Oracle-Tabellen dept und emp. Im aktuellen Sprint ist folgende User Story unserer Personalchefin eingeplant:

„Als Personalverantwortliche benötige ich eine Tabelle mit den Kennzahlen Gehaltssumme, Anzahl der Mitarbeiter und Durchschnittsgehalt je Abteilung, um die Fairness zu beurteilen.“

Ich interpretiere die Anforderung wörtlich und lege eine Tabelle gemäß *Listing 1* an.

Das Befüllen der Tabelle mache ich mithilfe einer PL/SQL Package Procedure namens „etl.refresh_deptsal“. Die ge-

```
create or replace package test_etl is
  --%suite

  --%test
  procedure refresh_deptsal;
end test_etl;
/
```

Listing 2: Test-Suite test_etl

```
create or replace package body test_etl is
  procedure refresh_deptsal is
    c_actual sys_refcursor;
    c_expected sys_refcursor;
  begin
    -- act
    etl.refresh_deptsal;

    -- assert;
    open c_actual for select * from deptsal;
    open c_expected for
      select d.deptno,
             d.dname,
             nvl(sum(e.sal), 0) as sum_sal,
             nvl(count(e.empno), 0) as num_emps,
             nvl(trunc(avg(e.sal), 2), 0) as avg_sal
      from dept d
      left join emp e
        on e.deptno = d.deptno
      group by d.deptno, d.dname;
    ut.expect(c_actual).to_equal(c_expected)
      .join_by('DEPTNO');
  end refresh_deptsal;
end test_etl;
/
```

Listing 3: Implementation der Test-Suite test_etl

naue Implementation ist an dieser Stelle nicht so wichtig. Interessierte finden sie in [1].

Test-Suite und Test-Run

Den Code vor dem Test zu schreiben, ist nicht Test-driven Development. Egal, ich bin Pragmatiker und nicht Dogmatiker. Am Ende habe ich das Bedürfnis zu prüfen, ob mein Code tut, was er tun sollte. Um den Code zur Ausführung zu bringen, verwende ich utPLSQL. Das utPLSQL Plug-in für den SQL Developer unterstützt mich dabei. Ich kann mir beispielsweise das Gerüst einer Test-Suite anhand meiner bestehenden Implementation generieren lassen oder die Code Templates nutzen, um Test-Packages, Test-Package-

Bodies oder auch einzelne Test-Procedures zu erstellen. Die *Abbildung 2* zeigt die Einstellungen des Generators und der Code Templates.

In einem neuen Worksheet habe ich „ut_spec“ gefolgt von Ctrl-Space eingegeben, um das Template für eine Test-Suite auszufüllen. Das *Listing 2* zeigt das Ergebnis.

Die Annotation „--%suite“ kennzeichnet das Package test_etl als Test-Suite. Die Annotation für das Package muss nach dem Schlüsselwort „is“ respektive „as“ erfolgen. Bis zur ersten Leerzeile werden alle Annotationen dem Package zugeordnet. Die Annotation „--%test“ markiert die Procedure refresh_deptsal als Test. Eine Test-Suite kann aus einer beliebigen Anzahl Tests bestehen. Test können optional noch mithilfe der Annotation „--%context(<name>)“ gruppiert

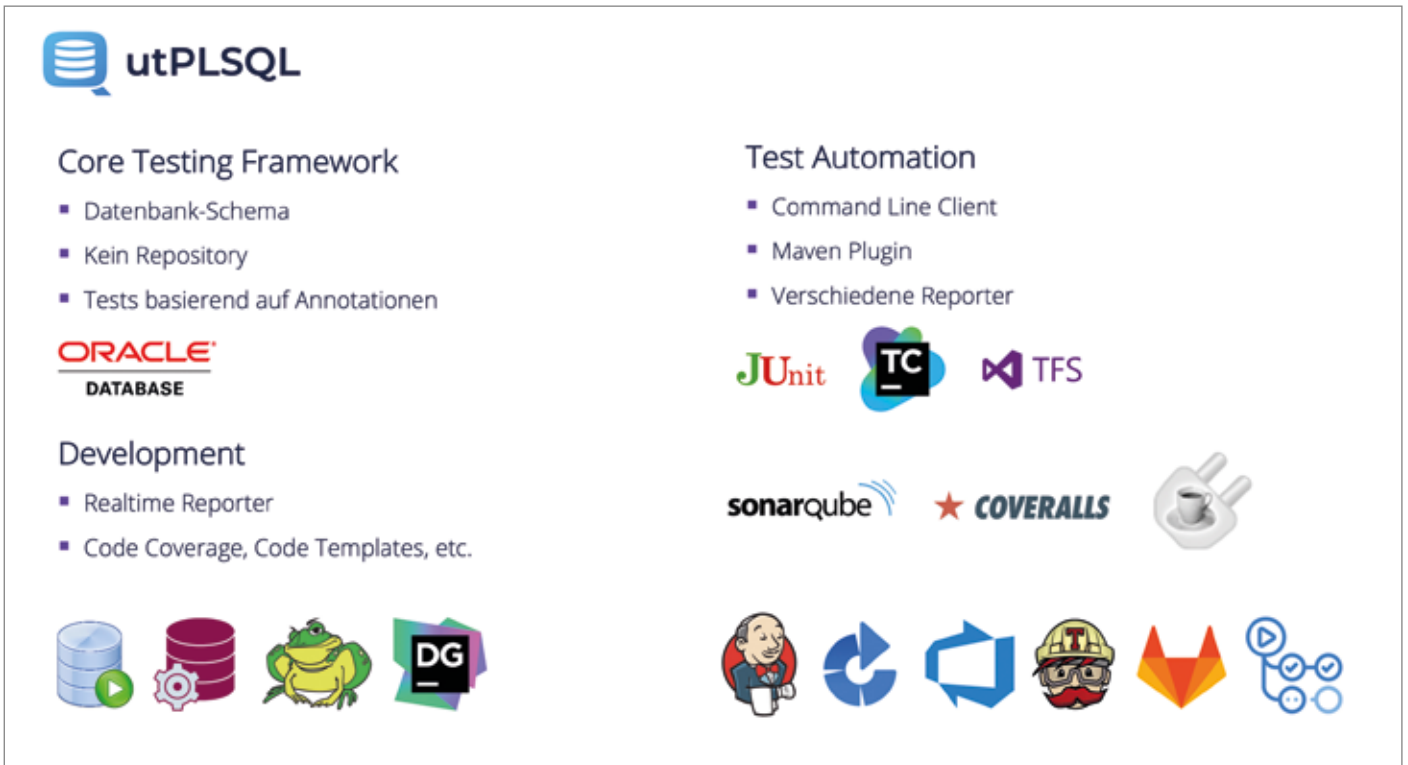


Abbildung 1: utPLSQL-Suite (Quelle: Philipp Salvisberg)

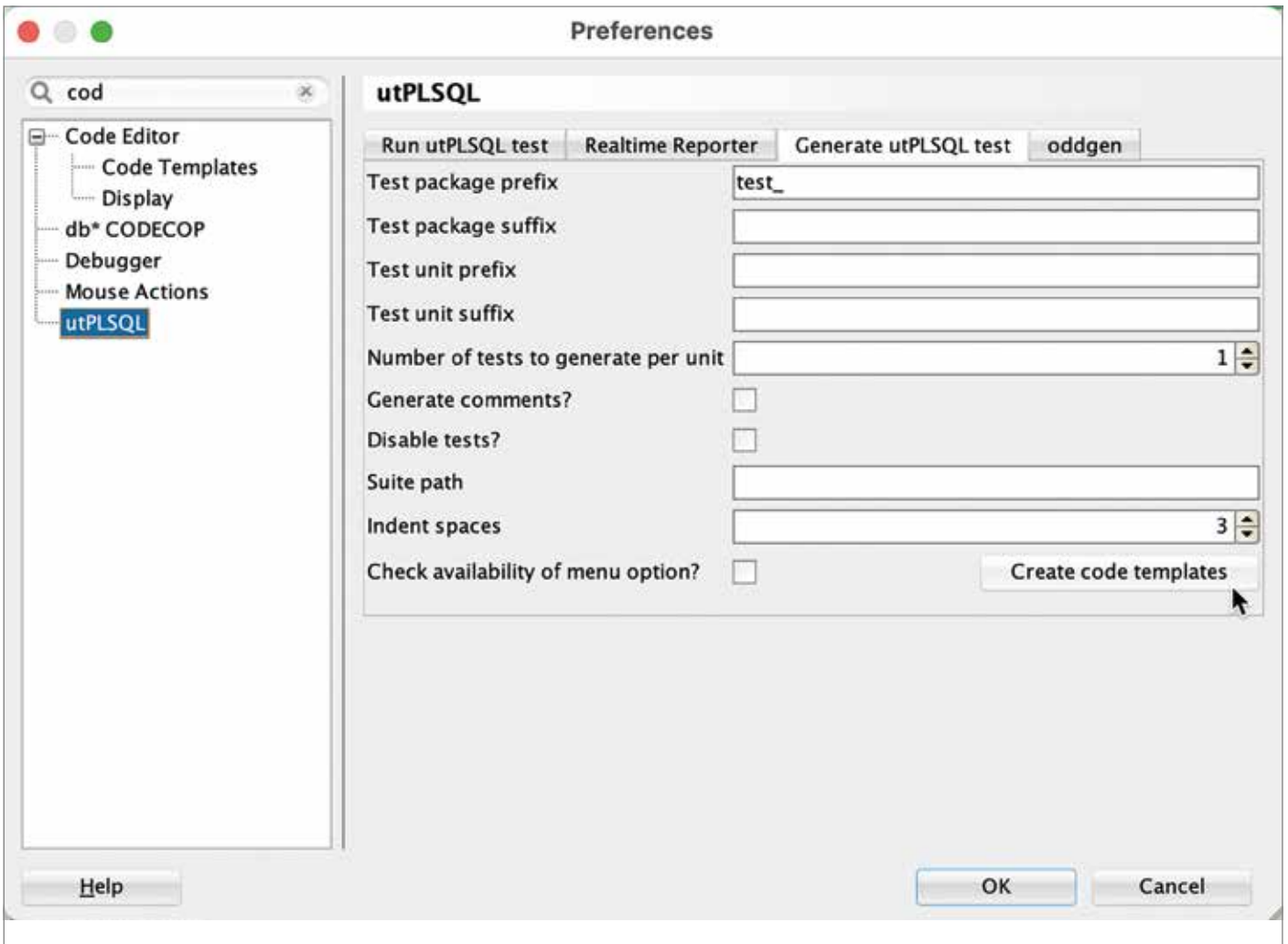


Abbildung 2: utPLSQL-Einstellungen für Test-Generator und Code Templates (Quelle: Philipp Salvisberg)

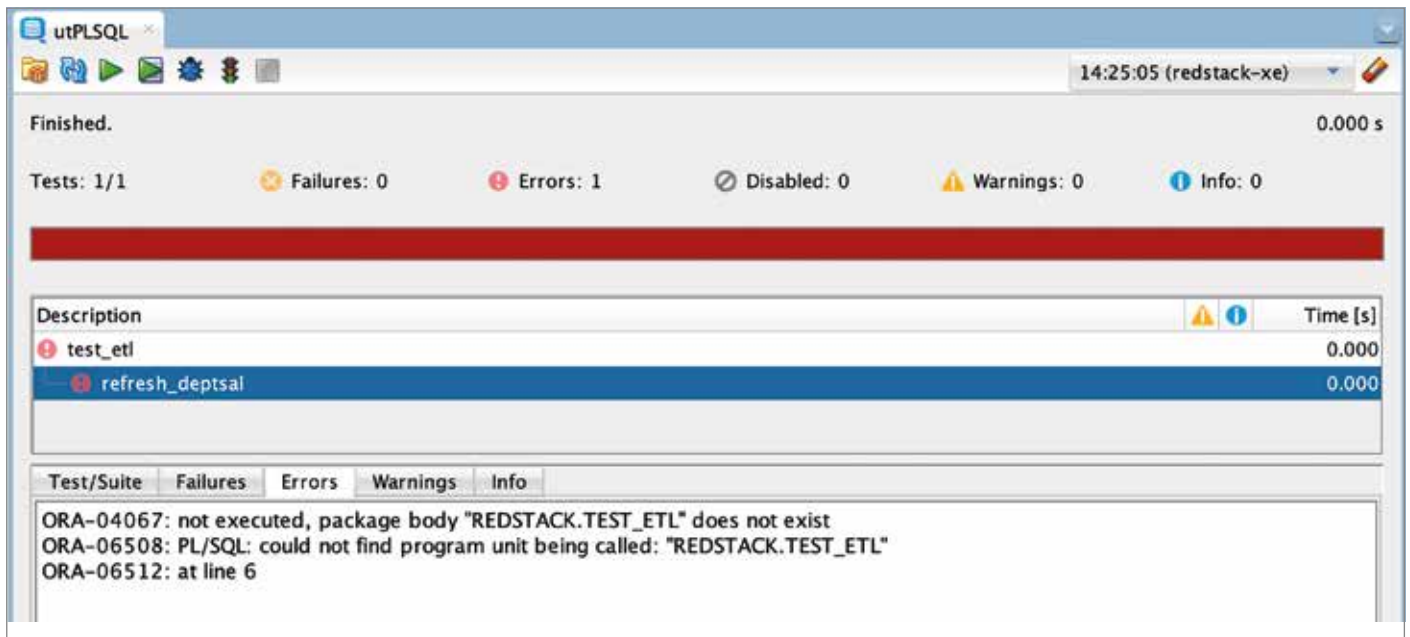


Abbildung 3: Test-Lauf mit Error beendet (Quelle: Philipp Salvisberg)

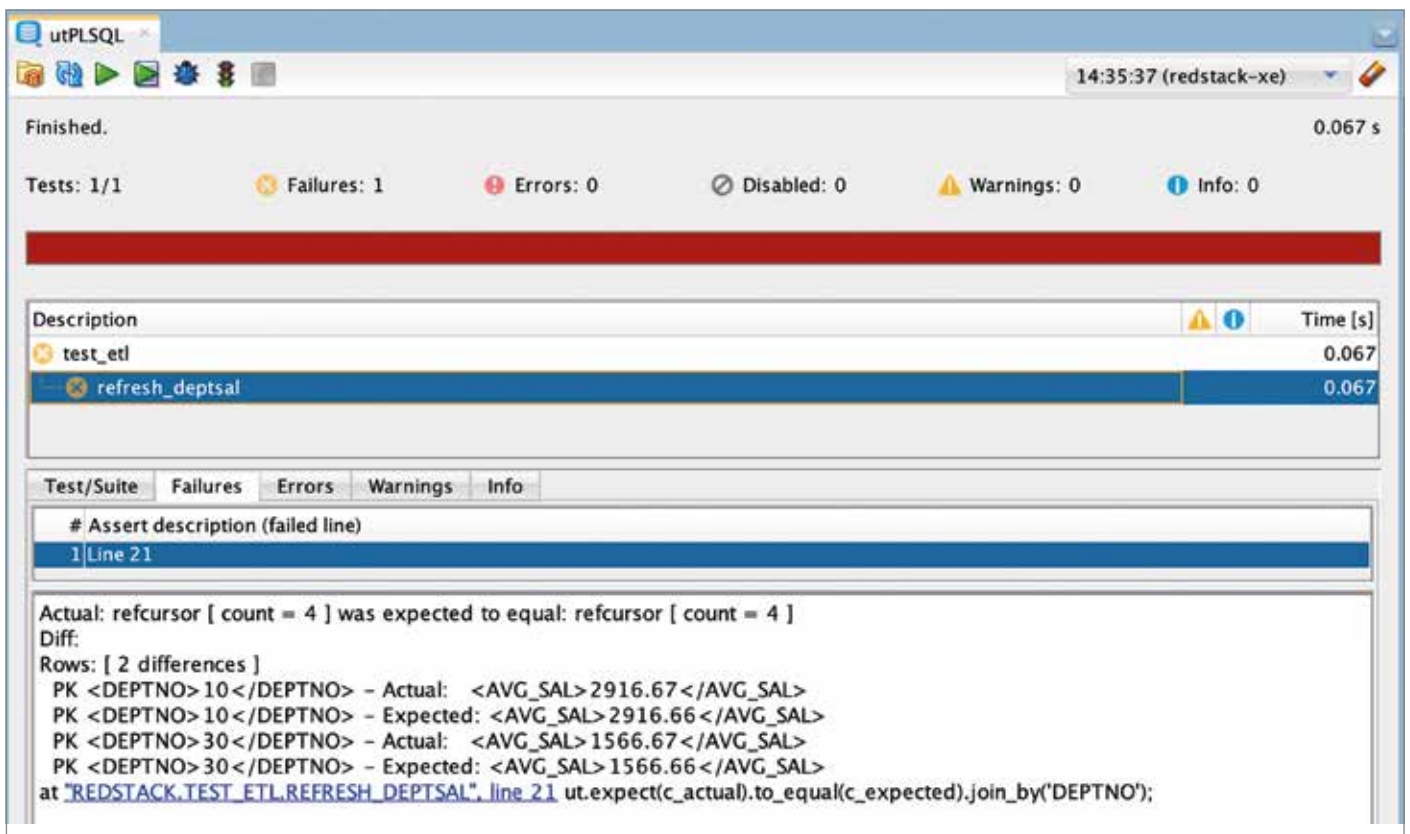


Abbildung 4: Test-Lauf mit Failure beendet (Quelle: Philipp Salvisberg)

werden. utPLSQL unterstützt über 20 Annotationen, die als Snippets im SQL Developer zur Verfügung stehen.

Diese Test-Suite kann ich jetzt ausführen. Entweder über das Kontextmenü im Editor oder auf einem Knoten im Navigationsbaum. *Abbildung 3* zeigt den utPLSQL Test-Runner nach der

Ausführung des Tests. Der Test-Runner verwendet im Hintergrund zwei zusätzliche temporäre Verbindungen zur Datenbank. Dadurch ist der SQL Developer niemals blockiert, was vor allem bei zeitintensiveren Tests sehr hilfreich ist. Es ist sogar möglich, unabhängige Tests gleichzeitig auszuführen.

Test-Implementation und Test-Run

Der Test „refresh_deptsal“ liefert einen Error. Der Grund ist einleuchtend. Die Implementation des Tests fehlt.

Die Implementation des Tests in *Listing 3* nutzt das AAA-Muster [5]. Es steht

für Arrange-Act-Assert. Im Schritt Arrange wird der Test vorbereitet. Hier fehlt dieser Schritt, da wir vollständig auf den bestehenden Daten basieren. Im Schritt Act wird der zu testende Code zur Ausführung gebracht. Im letzten Schritt Assert wird das aktuelle Ergebnis mit dem erwarteten Ergebnis verglichen. utPLSQL stellt hierfür eine Vielzahl von typischen Matchern zur Verfügung. In diesem Fall vergleichen wir zwei Cursor. Dabei werden die Datentypen sowie die Inhalte der Columns für alle Datensätze verglichen. Die *Abbildung 4* zeigt das Ergebnis nach unserem nächsten Testlauf.

Jetzt liefert der Testlauf ein Failure. Das bedeutet, der Test liefert nicht das erwartete Ergebnis. Im Register Failures des Test-Runners werden die Details angezeigt. Wir sehen, dass wir vier Datensätze erwartet und auch vier Datensätze erhalten haben. Allerdings gibt es bei zwei Datensätzen Differenzen in der Column avg_sal. Das sieht nach einer Rundungsdifferenz aus. Offensichtlich runden wir im Code auf und im Test ab. Was ist jetzt korrekt? Ich hätte gesagt, Aufrunden ist üblich und darum sollten wir den Test anpassen. Nach dem Ersetzen von „trunc“ durch „round“ in der Test-Implementation (*siehe Listing 2*) läuft der Test erfolgreich durch.

Testfälle für Insert, Update und Delete

Der Test „refresh_deptsal“ erwartet Daten in den Tabellen dept und emp. Sind die Tabellen leer, funktioniert der Test immer noch. Das ist nicht falsch, zeigt jedoch, dass die Qualität des Tests stark von unseren existierenden Daten abhängig ist. Außerdem laufen wir Gefahr, dass der Testcode für den Assert die Implementation des Prüflings spiegelt und allfällige Fehler wiederholt werden. Das *Listing 4* zeigt einen zusätzlichen Test basierend auf eigenen Testdaten.

Zuerst wird eine Abteilung -10 mit den Mitarbeitern -1 und -2 erstellt. Danach werden die Gehälter dieser beiden Mitarbeiter angepasst. Die Formulierung des erwarteten Ergebnisses basiert vollständig auf Literalen. Echte Identifier sind oftmals positiv. Die Verwendung von negativen Werten machen die Tests unabhängig von bestehenden

```
create or replace package body test_etl is
  (...)
  procedure refresh_deptsal_upd_dept_and_emp is
    c_actual sys_refcursor;
    c_expected sys_refcursor;
  begin
    -- arrange
    insert into dept (deptno, dname, loc)
    values (-10, 'utPLSQL', 'Winterthur');
    insert into emp (empno,ename,job,hiredate,sal,deptno)
    values (-1, 'Jacek', 'Developer', sysdate, 4700, -10);
    insert into emp (empno,ename,job,hiredate,sal,deptno)
    values (-2, 'Sam', 'Developer', sysdate, 4300, -10);
    etl.refresh_deptsal;

    -- act
    update dept set dname = 'Testing' where deptno = -10;
    update emp set sal = 5000 where empno = -2;
    etl.refresh_deptsal;

    -- assert
    open c_actual for
      select * from deptsal where deptno = -10;
    open c_expected for
      select -10 as deptno,
             'Testing' as dname,
             9700 as sum_sal,
             2 as num_emps,
             4850 as avg_sal
      from dual;
    ut.expect(c_actual).to_equal(c_expected);
  end refresh_deptsal_upd_dept_and_emp;
  (...)
end test_etl;
/
```

Listing 4: Test basierend auf eigenen Daten

```
create or replace trigger dept_as_iud
  after insert or update or delete on dept
begin
  etl.refresh_deptsal;
end;
/
create or replace trigger emp_as_iud
  after insert or update or delete on emp
begin
  etl.refresh_deptsal;
end;
/
```

Listing 5: After Statement Trigger, um Tabelle deptsal zu aktualisieren

Daten. utPLSQL setzt automatisch einen Savepoint vor der Ausführung eines Tests. Am Ende des Tests wird ein Rollback auf diesen Savepoint durchgeführt. Sollte dennoch irgendwo ein COMMIT erfolgen, sind die Testdaten mit negativem Identifier schnell gefunden und gelöscht.

Automatischer Refresh

Die Tabelle deptsal wird durch den Aufruf von etl.refresh_deptsal aktualisiert. Diesen Aufruf können wir mittels Datenbank-Trigger auslösen. *Listing 5* zeigt, wie.

Nach jedem DML Statement auf den Tabellen emp und dept wird jetzt die Ta-

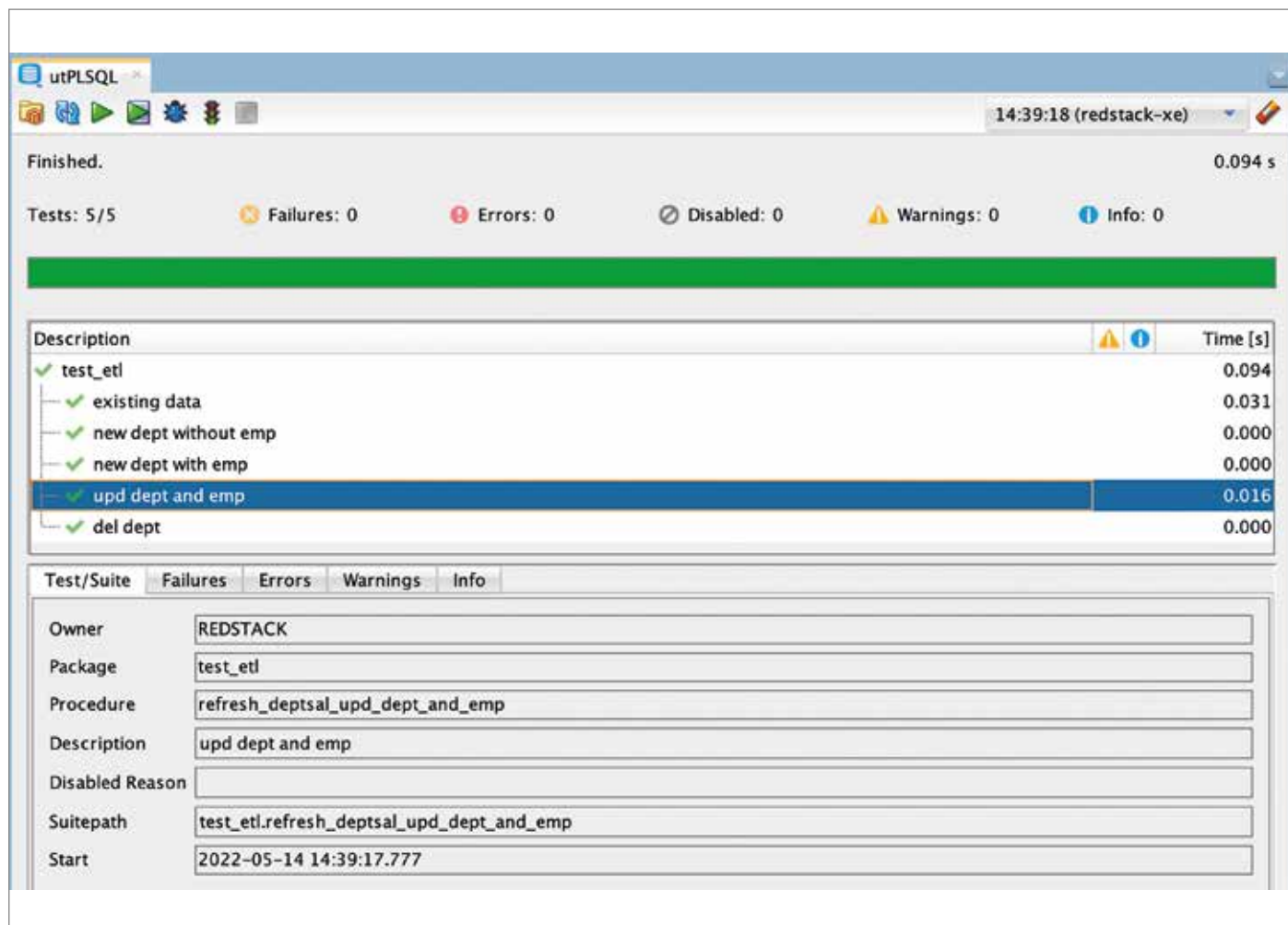


Abbildung 5: Test-Lauf mit Test für Insert, Update und Delete (Quelle: Philipp Salvisberg)

belle deptsal aktualisiert. Um zu testen, ob die DB-Trigger funktionieren, muss ich in den bestehenden Tests nur den Aufruf von „etl.refresh_deptsal“ entfernen. Für den Test auf den bestehenden Daten kann ich mit einem technischen Update eine Aktualisierung auslösen. Und voilà, alle Tests laufen fehlerfrei durch. Jetzt ist die Zeit gekommen, diese Änderungen im Version-Control-System einzuchecken, einen Pull-Request zu erstellen und einen Review durchzuführen.

Einarbeitung Review-Kommentare

Ich habe den Code meiner Kollegin Lisa gezeigt. Sie meinte, die Lösung sei komplexer als nötig und ineffizient. Einerseits wird die Tabelle deptsal zu oft aktualisiert, beispielsweise vor einem ROLLBACK oder wenn mehrere Datensätze in einer Transaktion verarbeitet werden. Andererseits sei die Bereitstel-

lung der Daten in Tabellenform nicht zwingend notwendig. Auch wenn die Personalleiterin ausdrücklich „Tabelle“ in ihrer Story erwähnt, ist es legitim, hier eine View statt einer Tabelle einzusetzen. Lisa denkt, dass bei unserer Datenmenge die Performance gut genug sein müsste, vor allem wenn die Abfrage auf wenige Abteilungen eingeschränkt wird. Außerdem werden diese Kennzahlen nicht so oft abgefragt. Eine View würde unsere Codebasis deutlich verkleinern und die Wartung vereinfachen.

Natürlich hat Lisa Recht. Ich kann die Tabelle deptsal, das Package etl und die DB-Trigger löschen. Eine einfache View namens deptsal, die auf der korrigierten Abfrage im Listing 2 basiert, reicht hier vollkommen aus. Aber was mache ich mit meinen Tests? Nichts! Sie beschreiben nach wie vor die Anforderungen, die sich nicht geändert haben, und laufen deshalb ohne Änderung fehlerfrei durch. Sofern mir beim Definieren der View kein Fehler unterlaufen ist. Refactoring

ist mit bestehenden Tests deutlich einfacher. Manche Leute würden sogar sagen, dass Tests das sichere Refactoring erst möglich machen.

Kernaussagen

Die ersten Schritte mit utPLSQL sind die schwierigsten. Machen Sie utPLSQL [2] in all ihren Entwicklungsumgebungen verfügbar. Bitten Sie Ihren DBA für zentral verwaltete Entwicklungsumgebungen um Hilfe. Installieren Sie auch das Plug-in für den SQL Developer [3], es ist zwar keine Voraussetzung, aber es vereinfacht die Arbeit mit utPLSQL erheblich. Starten Sie mit kleinen Schritten. Verwenden Sie utPLSQL, um Fehler zu reproduzieren oder um neue Anforderungen zu testen. Schon nach kurzer Zeit werden Sie feststellen, dass Sie kleinere Komponenten erstellen, weil Sie das Bedürfnis haben, Teile Ihres Codes schneller zu testen. Dies macht Ihren Code einfacher und wartbarer.

Schlussbemerkungen

Ich wurde von der DOAG gebeten, einen Artikel zu meinem Vortrag [4] zu schreiben. Dieser Vortrag besteht zu einem überwiegenden Teil aus Live-Demos, in denen sich sehr viel in kurzer Zeit zeigen lässt. Aus diesem Grund musste ich Themen wie Installation, Code-Generierung, die Verwendung von Snippets oder Code Coverage weglassen. Schauen Sie sich das Video an, wenn Sie mehr darüber erfahren möchten.

Quellen

- [1] <https://github.com/PhilippSalvisberg/utplsql-red-stack-demo>
- [2] <https://github.com/utPLSQL/utPLSQL>
- [3] <https://github.com/utPLSQL/utPLSQL-SQLDeveloper>

- [4] <https://www.youtube.com/watch?v=OxHMaOiPitI>
- [5] <https://java-design-patterns.com/patterns/arrange-act-assert/>
- [6] https://en.wikipedia.org/wiki/Test_automation

Über den Autor

Philipp Salvisberg ist ein Senior Principal Consultant bei Trivadis und ein Oracle ACE. Seit 1988 fokussiert er sich auf datenbankbasierte Lösungen und unterstützt Kunden beim Design, Aufbau und der Optimierung ihrer datenbankzentrischen Lösungen. Philipp hat ein Faible dafür, möglichst viel in einer einzigen SQL-Anweisung auszuführen und ist an so ziemlich allem interessiert, um die Datenbank so effizient wie möglich zu nutzen. Philipp ist der Autor der SQL Developer

Extension für utPLSQL und ein stolzes Mitglied des utPLSQL-Teams.



Philipp Salvisberg
philipp.salvisberg@accenture.com

DOAG DATENBANK KOLUMNE: DER WERT EINER GUTEN BERATUNG

Bruno Cirone

Bruno Cirone erläutert, warum es sich sehr lohnen kann, Kollegen/-innen oder Berater/-innen bei Problemen zu konsultieren. Aus: DOAG Online, 18.05.2022.

Vor einiger Zeit rief mich hilfesuchend ein Kunde an. Er erklärte mir die Situation wie folgt.

Seine Firma muss Daten von einem „Flat-Filesystem“ in eine Oracle-Datenbank migrieren. Die entsprechenden Tests seien erfolgreich verlaufen. Das Softwarehaus hat die entsprechenden Konvertierungsprogramme erstellt und dafür über 20.000 € erhalten.

Mit den Originaldaten dauerte die Umstellung mehr als 24 Stunden. So würde der Onlineshop der Firma nicht mehr rechtzeitig in Betrieb gehen können. Seine Frage war folgende: „Können Sie die Performance so verbessern, dass die Migration in weniger als 24 Stunden erledigt ist?“

Zunächst habe ich mir das Problem erklären lassen. In Kurzform ließ sich das Problem dementsprechend reduzieren, dass die Inhalte der Adressfelder bei der Umstellung von Großbuchstaben nach

Groß- und Kleinbuchstaben umgewandelt werden sollten. Also zum Beispiel von „HANS-WILHELM-STR.“ nach „Hans-Wilhelm-Str.“.

Natürlich wollte ich vom Softwarehaus wissen, wie sie diese Problematik gelöst haben.

Ihre Antwort war: „Wir haben jeden einzelnen Buchstaben gescannt. Dann je nach Situation HEX 20 abgezogen oder addiert. Wir haben die Besonderheiten, wie zum Beispiel Blanks, Bindestriche, Punkte berücksichtigt, da ja nach diesen Zeichen das erste Zeichen wieder groß zu schreiben ist.“ Beispielsweise wurde aus „A=HEX41“ durch die Addition von HEX20 „a=HEX61“.

Ich habe dem Kunden mitgeteilt, dass es für genau diese Situation eine „INITCAP“-Funktion von Oracle gibt. Die Umstellung nach dem Laden der Daten kann mit folgendem Updatebefehl erledigt werden: Update Kunden

set Vorname=Initcap(Vorname),
Name=Initcap(Name),
Ort=Initcap(Ort),
usw.

Diese Konvertierung dauerte ca. 10 Minuten für ca. 500.000 Kunden.

Was habe ich für meine Bemühungen in Rechnung gestellt? Hätte mich der Kunde vorher angerufen, hätte ich gar keine Rechnung gestellt. In diesem Fall musste ich ihm leider einen Espresso in Rechnung stellen, den er mir bei meinem nächsten Besuch ausgeben hat.

Daher meine Empfehlung: Fragen Sie häufiger Ihre Kollegen/Kolleginnen, Berater/Beraterinnen. Oft spart es einem eine Menge Zeit, Energie sowie Geld, und vor allem verursacht es wesentlich weniger Stress!

*Ihr Bruno Cirone
Mitglied der Delegiertenversammlung und
Themenverantwortlicher Security in der
Database Community*





Verzeichnisbasierte Namensauflösung von Oracle-Datenbanken

Stefan Oehrli, Trivadis part of Accenture

Hostname, Port sowie Datenbank- oder Service-Name sind die Basisinformationen, die man zum Erreichen einer Oracle-Datenbank benötigt. Im einfachsten Fall gibt man diese Informationen als EZCONNECT direkt beim Verbindungsaufbau an oder erstellt eine *tnsnames.ora*-Datei mit allen Informationen. Nichtsdestotrotz kann diese Konfigurationsdatei schnell unübersichtlich werden. Insbesondere bei umfangreicher und komplexer Datenbankumgebung mit einer großen Anzahl von Datenbanken, Standby-Datenbanken oder RAC-Instanzen ist in der Regel viel Aufwand für die Verwaltung von Oracle-Net-Service-Namen zu betreiben. Zudem besteht die Gefahr, dass die *tnsnames.ora*-Dateien nicht aktuell, redundant und inkonsistent sind oder einfach gelöscht wurden. Eine verzeichnisbasierte Namensauflösung bietet hier eine sichere und hochverfügbare Alternative für die Verwaltung der Datenbankadressen respektive Oracle Net Services Names.

Die *tnsnames.ora*-Datei ist die Konfigurationsdatei für die Auflösung von Oracle-Datenbanken beziehungsweise Oracle-Net-Service-Namen. Sie enthält die Netzwerkdienstnamen mit entsprechenden Verbindungsinformationen, auch Connect String genannt. Mithilfe der *tnsnames.ora*-Datei können Oracle-Clients beziehungsweise die Benutzer problemlos auf Oracle-Datenbanken zugreifen. Sie müssen lediglich den Oracle-Net-Service-Namen angeben. Der Connect String enthält dann alle relevanten Informationen für den Verbindungsaufbau. Dazu gehören im Minimum Host, Port und Datenbank- oder Servicename.

Bei größeren Umgebungen mit mehreren Oracle-Datenbanken und Hunderten oder mehr Clients wird die Verwaltung und Verteilung der *tnsnames.ora*-Datei-

en schwierig. Insbesondere wenn *Oracle DataGuard* oder *Oracle Real Application Cluster* hinzukommen, wo komplexere Connect Strings mit Failover- oder Load-Balancing-Informationen benötigt werden. Die zuverlässige Auflösung der Oracle-Net-Service-Namen und eine aktuelle *tnsnames.ora*-Datei sind entscheidend für einen hochverfügbaren Zugriff auf Oracle-Datenbanken. Eine manuelle Kopie der *tnsnames.ora*-Datei, ein zentraler NFS-Share oder eine Windows-Freigabe erfüllen diese Anforderung in der Regel nicht ausreichend.

Der Lösungsansatz

Die Idee, ein LDAP-Verzeichnis zur Verwaltung von Oracle Network Service Na-

mes [1] zu verwenden, ist nicht neu. Es gibt mehrere offizielle und inoffizielle Ansätze, wie dies umgesetzt werden kann. Anbei ein paar Beispiele:

- Oracle Directory:** Verwendung eines Oracle Directory für die Verwaltung von Oracle-Net-Service-Namen, zum Beispiel *Oracle Internet Directory (OID)* oder *Oracle Unified Directory (OUD)*. Wobei bis Februar 2022 ausschließlich für *Oracle Internet Directory (OID)* eine Restricted-Use-Lizenz verfügbar war, um die Oracle-Net-Service-Namen ohne zusätzliche Lizenzkosten in einem Oracle Directory zu verwalten. OID ist jedoch alles andere als schlank im Aufbau und im Betrieb. Seit Februar 2022 hat sich die Lizenzsituation leicht geändert. Oracle hat neu zusätzlich für *Oracle Unified Directory (OUD)* eine Restricted-Use-Lizenz für diesen Anwendungsfall veröffentlicht.
- Enterprise User Security:** Einrichtung von *Oracle Enterprise User Security (EUS)*. Auch hier wird ein Oracle Directory, das heißt OID oder OUD, mit einer entsprechenden Lizenz benötigt. Der Vorteil ist, dass bei dieser Lösung die Authentifizierung und Autorisierung der Datenbanken ebenfalls zentral gelöst wird. Die Einrichtung von EUS ist relativ komplex, da es nicht nur darum geht, einen LDAP-Server einzurichten. Man muss zusätzlich ein entsprechendes Benutzer- und Rollenkonzept definieren und umsetzen. Dies hat in jedem Fall Auswirkungen auf die bestehenden Datenbank-Anwendungen.
- MS Active Directory:** Die Oracle-Net-Service-Namen können grundsätzlich direkt in *MS Active Directory* eingetragen und verwaltet werden. Die Datenbanknamen beziehungsweise Connect Strings werden dann über *MS Active Directory* aufgelöst. Allerdings erfordert diese Methode eine Schemaerweiterung in *MS Active Directory* [4]. Dies ist in größeren Umgebungen meist nicht so einfach zu implementieren, zumal implementierte Schemaerweiterungen nicht wieder entfernt werden können.
- LDAP-Server:** Alternativ können weitere LDAP-Verzeichnisse wie der *OpenLDAP* oder der *389 Directory Server* verwendet werden. Zwar wird auch hier eine Schemaerweiterung benö-

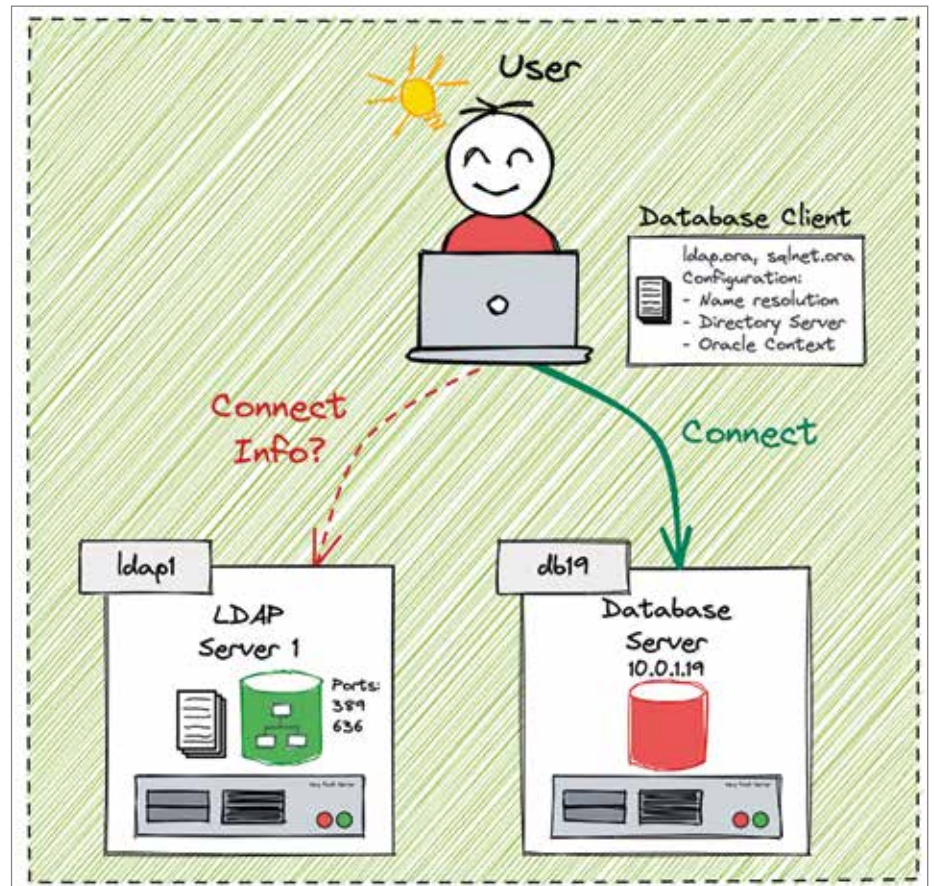


Abbildung 1: LDAP-basierte Auflösung von Oracle-Net-Service-Namen (Skizze von Stefan Oehrli)

tigt [3]. Da diese LDAP-Server in der Regel nur für die Oracle-Net-Service-Namensauflösung verwendet werden, ist diese Anpassung nicht kritisch, besonders da eine LDAP-Schemaerweiterung zum Standardvorgehen gehört. Der Vorteil dieser Methode ist zudem, dass durch die Verwendung eines Open-Source-LDAP-Servers die Kosten überschaubar bleiben. Nichtsdestotrotz ist diese Methode nicht offiziell dokumentiert und somit auch nicht offiziell unterstützt.

Im Folgenden befassen wir uns ausführlich mit einem Open-Source-LDAP-Server für die Namensauflösung. Die *Abbildung 1* zeigt dabei eine schematische Darstellung der Oracle-Net-Service-Namensauflösung am Beispiel eines LDAP-Verzeichnisses.

Für die hier vorgestellte Lösung verwenden wir den quelloffenen 389 Directory Server [6]. Dieser ist als Open-Source-Variante verfügbar respektive als *Red Hat Directory Server (RHDS)* [2] Bestandteil von *Red Hat Enterprise Linux*. Wobei für die Nutzung des *Red Hat Directory Server*

(*RHDS*) eine entsprechende Red-Hat-Subskription vorhanden sein muss. Als reiner LDAP für die Namensauflösung ist die Open-Source-Variante des *389 Directory Server* ausreichend. Zudem funktioniert diese einwandfrei unter *Oracle Enterprise Linux 8* oder auch als Docker Container [4].

Nachstehend werden die Installations- und Konfigurationsschritte für den *389 Directory Server* aus dem Fedora-Projekt beschreiben. Sie lassen sich, mehr oder weniger 1:1, auch auf den *Red Hat Directory Server (RHDS)* übertragen. Zumal die Dokumentation des *389-Directory-Servers* ohnehin auf der Red-Hat-Dokumentation basiert oder zumindest auf diese verweist.

Voraussetzungen und Anforderungen

Der *389 Directory Server*, aber auch *Red Hat Directory Server (RHDS)*, ist genügsam, was die Systemressourcen angeht, insbesondere wenn man einen LDAP-Server für ein paar 100 Oracle-Net-Service-Namen aufbaut. In der Dokumentation [2] werden keine spezifischen Hardware-Anfor-

derungen angegeben. Es wird empfohlen, für produktive Umgebungen mit mindestens 2 CPU-Kernen und 16 GB Speicher zu arbeiten. Für einen einfachen Testaufbau sind auch weniger Kerne beziehungsweise Memory ausreichend. Das Basisbetriebssystem im folgenden Konfigurationsbeispiel ist OEL 8.5.

Vorbereitung

Die Software für den *389 Directory Server* steht bei Oracle Enterprise Linux 8 im zusätzlichen Repository Oracle Linux 8 Application Stream als Module `389-ds` zur Verfügung. Als Erstes prüfen und aktivieren wir dieses Modul, sodass es anschließend auch installiert werden kann (siehe *Listing 1*).

Es ist zudem eine gute Idee, lokale Firewall-Ports für LDAP und LDAPS zu öffnen (siehe *Listing 2*).

Optional wird die Gruppe `dirsrv` vor der Installation mit einer spezifischen GID erstellt. Der Betriebssystembenutzer `oracle` wird dieser Gruppe zugewiesen. Dadurch können bestimmte administrative Aktivitäten als Benutzer `oracle` durchgeführt werden. Dieser Schritt ist grundsätzlich optional. Wird er weggelassen, muss man die Administration des LDAP-Servers entweder als Benutzer `root` oder immer via LDAPS ausführen (siehe *Listing 3*).

389 Directory Server installieren

Die Installation ist recht einfach. Das entsprechende Modul `389-ds` wurde vorher aktiviert. Somit kann man mit dem Kommando `yum` die entsprechenden Pakete und Abhängigkeiten für *389 Directory Server* relativ einfach installieren (siehe *Listing 4*).

Konfiguration 389 Directory Server für Oracle Net Service

Sobald die Pakete installiert sind, erstellen wir eine Directory-Server-Instanz. Der einfachste Weg hierzu ist die Verwendung einer Vorlage. Die Vorlage lässt sich direkt mit dem Kommando `dscreate` erstellen (siehe *Listing 5*).

Die Vorlage muss anschließend angepasst werden. Insbesondere sind die folgenden Werte zu setzen:

```
sudo yum module --available list 389-ds
sudo yum -y module enable 389-ds
```

Listing 1: Aktivieren des Moduls 389-ds

```
sudo firewall-cmd --list-all --permanent --zone=public
sudo firewall-cmd --permanent --add-service=ldap --zone=public
sudo firewall-cmd --permanent --add-service=ldaps --zone=public
sudo firewall-cmd --permanent --add-port=9090/tcp --zone=public
sudo firewall-cmd --reload
sudo firewall-cmd --list-all --permanent --zone=public
```

Listing 2: Firewall-Konfiguration

```
sudo groupadd --gid 520 dirsrv
cat /etc/group
sudo usermod -a -G dirsrv oracle
```

Listing 3: OS-Gruppen-Konfiguration

```
sudo yum install 389-ds-base sscg
```

Listing 4: Server-Installation

```
sudo dscreate create-template /tmp/oraNet.inf
```

Listing 5: Template erstellen

```
[general]
full_machine_name = ldap1.trivadislabs.com
start = true
[slapd]
instance_name = oraNet
port = 389
root_password = Welcome1
secure_port = 636
self_sign_cert = true
self_sign_cert_valid_months = 24
[backend-userroot]
create_suffix_entry = true
suffix = dc=trivadislabs,dc=com
```

Listing 6: OraNet-Template

```
sudo dscreate from-file /tmp/oraNet.inf
```

Listing 7: Erstellen der 389-Directory-Server-Instanz

```
sudo dsctl --list
sudo systemctl status dirsrv@oraNet.service
```

Listing 8: Kontrolle des Instanz-Status

- **full_machine_name** der vollqualifizierte Hostname des Servers
- **instance_name** Name der LDAP-Instanz
- **root_password** Passwort für den Root-Benutzer des LDAP-Servers
- **suffix** wird als Basis-DN für den Directory Information Tree (DIT) des LDAP-Servers verwendet

Im Folgenden ein Beispiel für den Aufbau einer Instanz mit dem Namen *oraNet* auf dem Host *ldap1.trivadislabs.com* mit der Basis-DN *dc=trivadislabs,dc=com* (siehe Listing 6).

Die Instanz wird dann als OS-Benutzer *root* mit dem Befehl *dscreate* erstellt (siehe Listing 7).

Das Kommando *dscreate* erstellt nicht nur die Instanz, sondern gleich auch die entsprechenden *start/stop*-Skripte oder einen Service. Ob die Instanz läuft, lässt sich mit dem Kommando *dsctl* prüfen. Alternativ kann man auch gleich direkt den Status des Service mit dem Kommando *systemctl* prüfen (siehe Listing 8).

Damit später entsprechende *Oracle-Net-Service-Namens*-Objekte im Directory Server angelegt werden können, müssen die Oracle-spezifischen Schemadefinitionen geladen werden. Dazu wird die Datei *90orclNet.ldif* [3] in das entsprechende Instanz- Verzeichnis kopiert und anschließend wird die Instanz neu gestartet. Diese Datei enthält die entsprechenden LDAP-Objekt-Klassen und Attribute [1], die für Oracle-Net-Service-Name-Objekte benötigt werden (siehe Listing 9).

Mit dem Kommando *dsctl* prüfen wir erneut den Status der LDAP-Instanz (siehe Listing 10).

Beim Ausführen der LDAP-Befehle muss man jeweils das Passwort angeben. Damit man dies nicht jedes Mal interaktiv angeben muss, kann man das Passwort temporär in einer Datei abspeichern und mit dem Parameter *-y* den LDAP-Befehlen übergeben. Es ist sinnvoll, die Passwortdatei anschließend wieder zu löschen (siehe Listing 11).

Der Verzeichnisserver ist an sich nun fertig und bereit, Informationen aufzunehmen. Damit man nun auch entsprechende Oracle-Net-Service-Namen registrieren kann, braucht es noch einen *Oracle-Kontext*. Dazu wird ein Objekt *cn=OracleContext* mit der Klasse *orclContext* in der Base DN *dc=trivadislabs,dc=com* erstellt (siehe Listing 12).

```
curl -Lf https://url.oradba.ch/90orclNet -o /tmp/90orclNet.ldif
sudo cp /tmp/90orclNet.ldif /etc/dirsrv/slapd-oraNet/schema/90orclNet.
ldif
sudo systemctl restart dirsrv@oraNet.service
```

Listing 9: Anpassen des LDAP-Schemas

```
sudo dsctl $(dsctl --list) status
sudo dsctl $(dsctl --list) healthcheck
```

Listing 10: Kontrolle des Instanz-Status und Healthcheck

```
echo "Welcome!" | tr -d '\n' >.oraNetDirectoryManager.pwd
chmod 600 .oraNetDirectoryManager.pwd
```

Listing 11: Password File

```
ldapadd -h $(hostname -f) -p 389 -x -D "cn=Directory Manager" \
-y .oraNetDirectoryManager.pwd <<-EOI
dn: cn=OracleContext,dc=trivadislabs,dc=com
Objektklasse: orclContext
cn: OracleContext
EOI
```

Listing 12: Erstellen des OracleContext

```
ldapadd -h $(hostname -f) -p 389 -x -D "cn=Directory Manager" \
-y .oraNetDirectoryManager.pwd <<-EOI
dn: dc=trivadislabs,dc=com
changetype: modify
add: aci
aci: (targetattr!="userPassword|authPassword")(version 3.0; acl "Anonymous read access"; allow (read,search,compare) userdn="ldap:///anyone");)
EOI
```

Listing 13: Anonyme Bind zulassen

```
ldapadd -h $(hostname -f) -p 389 -x -D "cn=Directory Manager" \
-y .oraNetDirectoryManager.pwd <<-EOI
dn: cn=TDB02,cn=OracleContext,dc=trivadislabs,dc=com
Objektklasse: top
Objektklasse: orclNetService
cn: TDB02
orclNetDescString: (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=db19)(PORT=1521))
(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=TDB02.trivadislabs.com))
(UR=A))
EOI
```

Listing 14: LDAP-Eintrag erfassen

Die Oracle-Clients führen die Anfragen für die Auflösung der Oracle-Net-Service-Namen in der Regel mit einer anonymen LDAP-Abfrage beziehungsweise mit einem anonymen Bind durch. In der Standardkonfiguration wird dies nicht erlaubt und muss zwingend mit entsprechenden ACI-Regeln freigeschaltet werden (siehe Listing 13).

LDAP-basierte Oracle-Net-Dienste

Der leere 389-Verzeichnisserver ist jetzt bereit für die Erfassung von Oracle-Net-Service-Namen. Die Einträge können auf unterschiedliche Weise erfasst werden, zum Beispiel mit dem Kommando *ldapadd*, LDAP-Browser etc. Im folgenden Beispiel wird mit dem Kommando *ldapadd* ein Eintrag mit dem Namen TDB02 und dem entsprechenden Oracle Net Description String hinzugefügt (siehe Listing 14).

Mit dem Kommando *ldapmodify* können bestehende Einträge verändert werden (siehe Listing 15).

Das Kommando *ldapsearch* erlaubt das Suchen nach LDAP-Einträgen (siehe Listing 16).

Mit dem Kommando *ldapdelete* können die Einträge selbstverständlich auch gelöscht werden (siehe Listing 17).

Bevor man die LDAP-basierten *Oracle-Net-Service-Namen* nutzen kann, ist die Oracle-Net-Konfiguration anzupassen. Dazu wird die Reihenfolge der Namensauflösung in *\$TNS_ADMIN/sqlnet.ora* mit dem Parameter *NAMES.DIRECTORY_PATH* angepasst und LDAP an die erste Stelle gesetzt (siehe Listing 18).

Darüber hinaus muss die Datei *\$TNS_ADMIN/ldap.ora* erstellt werden. Darin wird die Konfiguration des LDAP-Servers wie folgt angegeben (siehe Listing 19).

Ist die Oracle-Net-Konfiguration abgeschlossen, kann mit dem Kommando *tnsping* überprüft werden, ob der *Oracle Net Service Name* über *LDAP* oder *tnsnames.ora* aufgelöst wird (siehe Listing 20).

Wie geht es weiter?

Dieser Artikel beschreibt die einfache Konfiguration eines 389 *Directory Server* für Oracle Net Service. Im Prinzip steht dem produktiven Einsatz nichts im Wege.

```
ldapmodify -h $(hostname -f) -p 389 -x -D "cn=Directory Manager" \
-y .oraNetDirectoryManager.pwd <<-EOI
dn: cn=TDB02,cn=OracleContext,dc=trivadislabs,dc=com
changetype: modify
ersetzen: orclNetDescString
orclNetDescString: (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=db19)
(PORT=1521))
(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=TDB02.trivadislabs.com))
(UR=A))
EOI
```

Listing 15: LDAP-Eintrag modifizieren

```
ldapsearch -h $(hostname -f) -p 389 -x -LLL -b "dc=trivadislabs,dc=com"
-s sub "(&(objectclass=orclNetService)(cn=TDB0*))"
```

Listing 16: LDAP-Eintrag modifizieren

```
ldapdelete -h $(hostname -f) -p 389 -x -D "cn=Directory Manager" \
-y .oraNetDirectoryManager.pwd \
cn=TDB03,cn=OracleContext,dc=trivadislabs,dc=com
```

Listing 17: LDAP-Eintrag löschen

```
NAMES.DIRECTORY_PATH=(LDAP, TNSNAMES, EZCONNECT )
```

Listing 18: Auszug *sqlnet.ora*-Konfiguration

```
DIRECTORY_SERVERS=(ldap1.trivadislabs.com:389:636)
DEFAULT_ADMIN_CONTEXT="dc=trivadislabs,dc=com"
DIRECTORY_SERVER_TYPE=OID
```

Listing 19: Auszug *ldap.ora*-Konfiguration

```
oracle@db19:~/ [rdbms19] tnsping TDB02
TNS Ping Dienstprogramm für Linux: Version 19.0.0.0.0 - Produktion am
01-MAR-2022 06:31:06
Copyright (c) 1997, 2021, Oracle. Alle Rechte vorbehalten.
Verwendete Parameterdateien:
/u01/app/oracle/network/admin/sqlnet.ora
Verwendeter LDAP-Adapter zum Auflösen des Alias
Versuch der Kontaktaufnahme mit (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
(HOST=db19)(PORT=1521))(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_
NAME=TDB02.trivadislabs.com))(UR=A))
OK (10 msec)
```

Listing 20: TNS-Ping-Beispiel

Es empfiehlt sich jedoch, 2-3 Punkte genauer zu betrachten.

- Benutzer- und Rollenkonzept für die Administration und Verwaltung von Daten im LDAP

- Toolset zur Administration, zum Beispiel Skripte, LDAP-Browser etc.
- Hochverfügbare Konfiguration des 389 *Directory Server*, beispielsweise mehrere LDAP-Server mit entsprechender Replikationskonfiguration

- Produktionstaugliche Zertifikate
- Entwicklung und Umsetzung eines Betriebs- und Sicherheitskonzeptes. Dazu gehören u.a. Backup- und Restore-Aufgaben.
- Klärung von Lizenzen und Subskriptionen, insbesondere bei Einsatz des *Red Hat Directory Server (RHDS)* anstelle des reinen 389 Directory Server

Fazit

Mit dem *389 Directory Server* ist es, genau wie bei OpenLDAP relativ einfach, ein zentrales Verzeichnis für die *Oracle-Net-Service-Namen* oder *tnsnames.ora* zu erstellen. Innerhalb weniger Minuten hat man einen eigenständigen LDAP-Server erstellt. Mit etwas mehr Aufwand ist auch die Konfiguration von sicheren SSL-Zertifikaten, erweitertem *Directory Information Tree (DIT)* mit verschiedenen Suffixen sowie die Replikation usw. problemlos möglich.

Auf Basis dieses Ansatzes konnten im Rahmen von Kundenprojekten bereits erfolgreich hochverfügbare LDAP-Verzeichnisse mit mehreren Suffixen beziehungsweise Oracle Network Service Domain Names eingerichtet werden. Damit steht der Ablösung der mühsamen, manuellen Verwaltung der *tnsnames.ora*-Dateien durch ein zentrales Verzeichnis nichts mehr im Wege. Insbesondere dann, wenn nur die *Oracle-Net-Service-Namen* im Fokus stehen. Soll darüber hinaus die Au-

thentifizierung und Autorisierung der Datenbanken zentral eingerichtet werden, führt kein Weg an *Oracle Enterprise User Security (EUS)* oder *Oracle Centrally Managed Users (CMU)* vorbei. Der Vorteil der hier vorgestellten Lösung ist, dass sie mit *Oracle Centrally Managed Users (CMU)* kombiniert werden kann. Das heißt, die Authentifizierung und Autorisierung erfolgt mit CMU über die Active-Directory-Integration, während die Namensauflösung auf dem LDAP-Verzeichnis basiert. Alles mit überschaubarem Aufwand und ohne zusätzliche Lizenzkosten. Im Gegensatz dazu ist eine Lösung mit *Oracle Enterprise User Security (EUS)* etwas flexibler, aber auch komplexer und kostenintensiver.

Quellen

- [1] Oracle® Database, Database Net Services Reference 21c – Appendix B LDAP Schema for Oracle Net Services <https://docs.oracle.com/en/database/oracle/oracle-database/21/netrf/index.html>
- [2] Red Hat Directory Server 11, Product Documentation https://access.redhat.com/documentation/en-us/red_hat_directory_server/11
- [3] Oehrli Stefan (2022), GitHub Gist mit LDAP-Schema für Oracle Net Services <https://url.oradba.ch/90orclNet>
- [4] Oehrli Stefan (2022), GitHub Projekt oehrli/389ds 389 Directory Server in Docker
- [5] Oracle® Database, Database Administrator's Reference 21c for Microsoft Windows - Using Oracle Database with Microsoft Active Directory, [https://docs.](https://docs.oracle.com/en/database/oracle/oracle-database/21/ntqrf/index.html)

[oracle.com/en/database/oracle/oracle-database/21/ntqrf/index.html](https://docs.oracle.com/en/database/oracle/oracle-database/21/ntqrf/index.html)

- [6] Fedora 389 Directory Server, <https://directory.fedoraproject.org>

Über den Autor

Stefan Oehrli ist Data Engineering Manager bei *Trivadis part of Accenture*. Seit 1998 arbeitet er mit DB-Systemen. Während dieser Zeit konnte er sich fundierte Erfahrungen im Aufbau und Betrieb von Oracle DBs, mit der Erstellung/Umsetzung von B&R sowie mit Security-Konzepten sammeln. Seine Hauptinteressen gelten der Architektur, B&R, Container, Sicherheit, EUS/CMU, DB Internals und allem anderen rund um die DB-Technologien. Neben den Kursen und Seminaren hält er als Oracle ACE regelmäßig Vorträge auf Konferenzen und User-Group-Meetings. Er ist Co-Autor des Buches „Der Oracle DBA“ (Hanser, 2016).



Stefan Oehrli
stefan.oehrli@accenture.com



APEX-Security-Grundlagen

Philipp Hartenfeller, MT AG

Sicherheitslücken in jeglicher Software können verheerend für Unternehmen sein. Folgen können zum Beispiel ein Abfluss sensibler Informationen oder Sabotagen, die den Betriebsalltag stören, sein. Da international die Cyberkriminalität steigt, sollten bei der Entwicklung stets Best Practices im Bereich IT-Sicherheit beachtet werden. Auch für APEX-Anwendungen gibt es konkrete Risikopotenziale, die in diesem Artikel vorgestellt werden. Der Fokus soll hierbei auf den Punkten liegen, die beim Entwickeln von APEX-Anwendungen relevant sind. Folglich werden Aspekte wie administrative Konfigurationen etc. nicht behandelt.

Die APEX-Architektur

Um Sicherheitsrisiken besser einschätzen zu können, sollte die Architektur von APEX nicht unbekannt sein. Sie besteht aus drei Ebenen:

- Webapplikation im Browser
- Oracle REST Data Services (kurz ORDS)
- Datenbank mit APEX

Der Browser kommuniziert mit dem ORDS über HTTP(S)-Anfragen, die anschließend in der Datenbank etwa in Form einer SQL-Abfrage durchgeführt werden. Beim Entwickeln steuern wir über unsere APEX-App direkt, was im Client (dem Webbrowser) geschieht, indem wir APEX-Komponenten einbinden und JavaScript beispielsweise über Dynamic-Actions steuern. Ebenso bestimmen wir,

was auf dem Server (Datenbank) stattfindet, indem wir Datenbankobjekte bereitstellen und Prozesse in PL/SQL programmieren [1].

Client vs. Server

Diese Trennung zwischen Frontend und Backend ist auch für den Sicherheitsas-

pekt von großer Bedeutung. Um nicht un-
freiwillig Informationen preiszugeben, die
beispielsweise eigentlich nur für Admins
und nicht für andere Augen bestimmt
sind, muss sichergestellt werden, dass
diese Informationen für die unautorisiert
Person nie den Server verlassen.

Beispielsweise reicht es nicht aus, ein
Page-Item mit einer Dynamic-Action zu
verstecken oder den Item-Typ „hidden“
zu verwenden. In beiden Fällen werden
die Items lediglich versteckt und können
weiterhin in der Seitenstruktur mit Brow-
ser-Entwicklerwerkzeugen gefunden wer-
den. Zusätzlich werden sie durch den Auf-
ruf von „apex.items“ mit allen anderen
Items aufgelistet; mithilfe der Funktion
„getValue“ kann anschließend auch be-
quem der Wert abgefragt werden.

Um stattdessen den Wert für Unbefug-
te unzugänglich zu machen, sollte bereits
serverseitig der Wert herausgefiltert wer-
den, damit dieser nie den Client erreicht.
Dies ist im Page-Designer möglich, indem
auf dem Page-Item eine Server-Side-Con-
dition oder ein Authorization-Scheme ein-
gestellt wird.

Tückisch sind allerdings Knöpfe, die
einen PL/SQL-Prozess auslösen. Wenn
beispielsweise eine Löschaktion nur für
bestimmte Personen ausführbar sein
soll, reicht es nicht aus, den zugehörigen
Knopf über eine Server-Side-Condition
oder ein Authorization-Scheme einzu-
schränken. Zusätzlich muss auch in APEX
der Prozess mit derselben Bedingung be-
legt werden. Der Knopf ist lediglich das
Interaktionselement, das den JavaScript-
Code „apex.submit“ auslöst. Dieser Code
kann trotz Bedingung auf dem Knopf ma-
nuell ausgeführt werden.

Checksummen

APEX bietet uns ebenfalls ein Werkzeug,
um bösartige URL-Manipulationen zu ver-
hindern. In vielen Anwendungen werden
Reports verwendet, um einen tabellari-
schen Überblick zu geben. Oftmals wird
über einen in jeder Zeile enthaltenen Link
auf ein Formular zum Bearbeiten des ein-
zelnen Datensatzes weitergeleitet. In der
Regel wird über die URL eine ID an die
Formularseite übermittelt, um initial die
Daten zu laden.

Je nach Anwendung kann es verschie-
dene Berechtigungskonzepte geben, so-

```
l_sql := '
select autor, text
  from gedichte
 where jahr between 1900 and 1930
       and length(text) '||pi_operator||' '||pi_times;

execute immediate l_sql bulk collect into l_my_tab;

-- fix: pi_times binden und pi_operator manuell prüfen
if pi_operator not in ('<', '>', '=', '>=', '<=') then
  raise e_operator_not_allowed; --(user defined exception)
end if;

l_sql := '
select autor, text
  from gedichte
 where jahr between 1900 and 1930
       and length(text) '||pi_operator||' :pi_times';

execute immediate l_sql bulk collect into l_my_tab using pi_times;
```

Listing 1: Beispiel SQL-Injection

```
select apex_escape.html(first_name)
  into l_first_name
  from users
 where id = :P40_ID;

http.p('
<div>
  <span>Vorname:</span>
  <span>'||l_first_name||'</span>
</div>');
```

Listing 2: Manuelles Bereinigen von HTML

dass nur Datensätze, die für die jeweilige
Person relevant sind, angezeigt werden.
Eine Person mit bösen Absichten könn-
te bei der Nutzung von Sequenzen für
die IDs auf der Formularseite bei der URL
verschiedene Nummern durchprobieren
und so Einsicht in Daten erlangen, die gar
nicht vom Report verlinkt waren.

Um dieses Verhalten zu verhindern,
kann im Page-Designer für jede Seite un-
ter dem Bereich „Security“ unter „Page
Access Protection“ „Arguments Must
Have Checksum“ eingestellt werden (*sie-
he Abbildung 1*). Wenn dies aktiviert ist,
benötigt die Seite neben dem numeri-
schen ID-Parameter auch eine passende
Checksumme, die nur der Server mit dem
passenden Geheimnis generieren kann.
APEX baut diese Werte automatisch in
jede Verlinkung auf die Seite mit ein und
über den API-Aufruf „apex.page.get_url“
können Links mit passenden Prüfsum-
men auch manuell erzeugt werden. Falls
eine Person nun versucht, die ID manu-

ell zu verändern, stimmt der Wert nicht
mehr mit der Checksumme überein, der
Server streikt und liefert eine Fehlermel-
dung aus.

Es gibt auch die Optionen „No Argu-
ments Supported“ sowie „No URL Ac-
cess“, bei letzterer kann eine Seite nur
über einen Branch aufgerufen werden.
Außerdem kann die Einstellung auch auf
Item-Ebene gesetzt werden. Im Bereich
„Security“ lassen sich bei jedem APEX
Item unter „Session State Protection“
auch Checksummen aktivieren.

In den „Shared Components“ unter
„Session State Protection“ lässt sich eben-
falls eine Übersicht über alle in der Appli-
kation verwendeten Einstellungen anzei-
gen (*siehe Abbildung 2*).

SQL-Injections

Eine der bekanntesten und tückischsten
Sicherheitslücken im Datenbankbereich

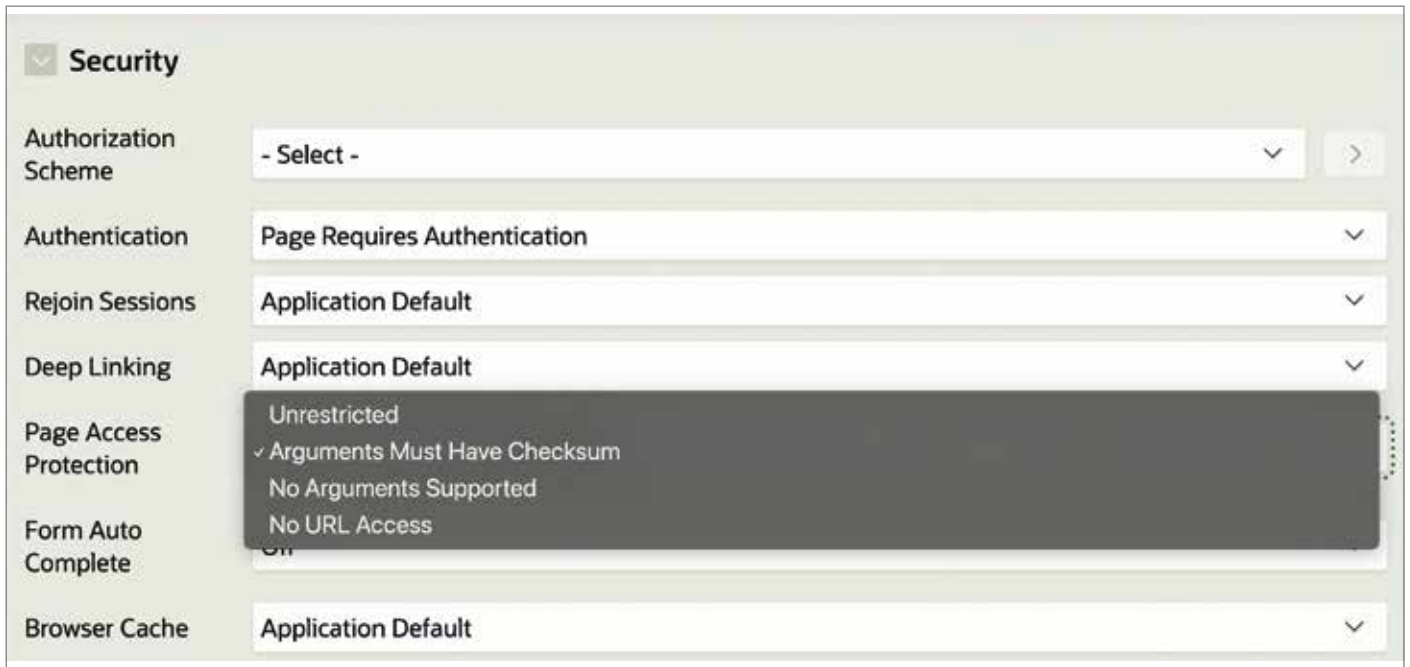


Abbildung 1: Security-Einstellungen einer Seite im Page-Designer (eigene Darstellung) (Quelle: Philipp Hartenfeller)

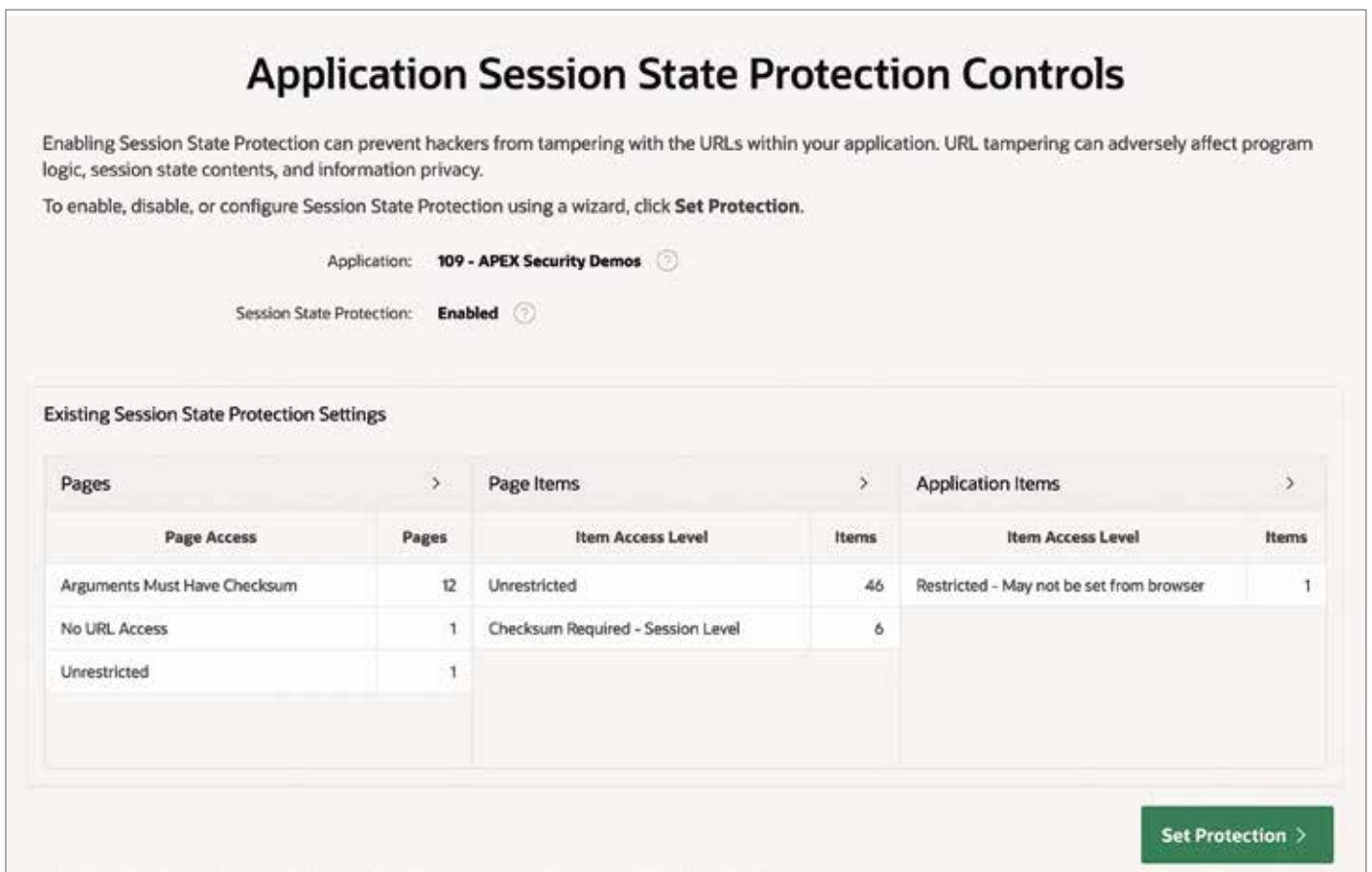


Abbildung 2: Session-State-Protection-Übersicht in den Shared Components (eigene Darstellung) (Quelle: Philipp Hartenfeller)

sind SQL-Injections. Solche Sicherheitslücken treten auf, wenn Nutzereingaben nicht ordnungsgemäß bereinigt werden und somit Einfluss auf SQL-Abfragen nehmen können.

In APEX-Anwendungen kann es öfter auftreten, dass Personen Daten beispielsweise über Auswahllisten oder Eingabefelder einschränken können. Dies ist allerdings noch kein Grund zur Besorgnis,

da die Standardkomponenten wie der Interactive Report natürlich abgesichert sind. Auch bei SQL-Abfragen, die im Page-Designer hinterlegt werden, können Page-Items mit der Doppelpunktsyntax

(z. B. „where category = :P404_CATEGORY“) problemlos eingebunden werden. Mit dieser Syntax kümmert sich die Datenbank darum, dass die Werte sicher in die Abfrage „gebündelt“ werden.

Problematisch wird es allerdings bei dynamischen SQL-Abfragen. Diese Variante muss lediglich in wenigen Fällen verwendet werden, wenn beispielsweise Spalten, Tabellen, Operatoren oder weitere, nicht durch Variablen kompilierbar ersetzbare Aspekte dynamisch angepasst werden müssen [2].

In *Listing 1* wird ein String mit einer SQL-Abfrage aufgebaut und anschließend mit „execute immediate“ ausgeführt. Dort werden zunächst die Werte per Konkatenation in die Abfrage eingebaut. Dies stellt eine Sicherheitslücke dar, weil Eingaben wie „> 0 or 1 = 1 --“ oder „> 999999 union all select username, password from users --“ für „pi_operator“ komplette Überschreibungen der eigentlichen Abfrage ermöglichen.

Im zweiten Teil wird die Lücke anschließend geschlossen, indem der Wert „pi_times“ „gebündelt“ wird. Für den Operator ist dies nicht möglich, da es in der Abfrage selbst keinen Wert darstellt. Somit muss der Wert weiterhin verkettet werden, allerdings kann dies mit einer händischen Prüfung auf erlaubte Werte abgesichert werden.

Weitere Möglichkeiten zum Absichern von dynamischen Abfragen bieten die Pakete „DBMS_ASSERT“ sowie „DBMS_SQL“. Ersteres bietet Funktionen zur Validierung bestimmter Werte wie Tabellen- oder Spaltennamen, während das zweite Paket verwendet werden kann, um komplette Abfragen zu kapseln.

Cross-Site-Scripting (XSS)

XSS ist eine Sicherheitslücke, die ähnlich zu SQL-Injections ausgelöst wird, wenn Code aus Nutzereingaben ausgeführt wird. Allerdings wird in diesem Fall JavaScript-Code eingeschleust, der folglich nicht in der Datenbank, sondern in den Clients durchgeführt werden soll.

Obwohl somit nicht die Kontrolle über die Datenbank übernommen werden kann, können die Folgen einer erfolgreichen XSS-Attacke trotzdem fatal sein. So können zum Beispiel die angezeigten Daten aller Anwender an einen Server

vom Hacker geschickt oder Knöpfe betätigt werden, die beispielsweise nur für privilegierte Accounts bestimmt sind [3].

Eine Attacke geschieht beispielsweise über die Eingabe eines HTML-Skript-Tags mit JavaScript-Code in ein Eingabefeld. Um dagegen gewappnet zu sein, muss bei der Entwicklung darauf geachtet werden, dass bei der Anzeige der Werte potenzieller HTML-Code „escaped“ wird. Bei Input-Feldern (auch Display-Only) kümmert sich APEX automatisch darum. Bei Reportspalten muss darauf geachtet werden, dass die Einstellung „Escape special characters“ aktiviert ist. Wenn manuelle HTML-Konstrukte mit der Funktion „http.p“ erstellt werden, können Werte manuell mit der Funktion „apex_escape.html“ bereinigt werden. Ein Beispiel hierzu ist in *Listing 2* dargestellt.

Zusätzliche Hilfen bieten Einstellungen auf APEX Page-Items. So können bei Textfeldern unter „Security > Restricted Characters“ Regeln festgelegt werden. Eine davon sorgt beispielsweise dafür, dass nur Buchstaben, Zahlen und Leerzeichen erlaubt sind, eine weitere dafür, dass die für die HTML-Syntax relevanten Zeichen „<“, „>“ und „““ verboten sind. Letztendlich ist es allerdings besser, die Lücke direkt zu schließen. Dieses Page-Item muss schließlich nicht die einzige Möglichkeit bleiben, den Wert in der Datenbank zu verändern.

Weitere grundsätzliche Tipps

- Grundsätzlich ist es empfehlenswert, vorhandene Datenbankfeatures zu nutzen, um Risiken zu minimieren. Durch eine **Schematrennung** sind nicht alle Datenbankobjekte für jeden Nutzer verwendbar. Über den Grant-Befehl kann anschließend anderen Nutzern granular Zugriff auf einzelne Objekte erteilt werden. Ebenso kann die Funktion „**Virtual Private Database**“ verwendet werden, um weitergehende Einschränkungen durchzusetzen.
- Im APEX-Umfeld gibt es einige **Scanner**, die sicherheitsrelevante Aspekte überprüfen. Der integrierte Advisor hat integrierte Sicherheitschecks, soll aber in Zukunft überarbeitet werden und ein PL/SQL-API

erhalten. Dann könnte er auch in Pipelines eingebunden werden und somit automatisierte, kontinuierliche Checks durchführen. Des Weiteren existieren das kostenpflichtige Produkt ApexSec sowie die Open-Source-Lösung APEX-SERT.

- **Single Sign-on** vermindert die Komplexität der Authentifizierung auf eine zentrale Komponente. Außerdem wird so die Anzahl der verschiedenen Passwörter reduziert.
- Ebenfalls empfiehlt es sich, fein **granulare Autorisierungskonzepte** einzusetzen. Beispielsweise eignet sich das System, dass Rechte (darf lesen, erstellen, löschen, Knopf X drücken ...) und Rollen (Product Owner, Admin, Analyst ...) definiert werden. Nun können jeder Rolle genau die Rechte zugeordnet werden, die benötigt werden.
- Auch während der Laufzeit sollte das Geschehen überwacht werden. Dazu ist es hilfreich, ein **Logging** in den PL/SQL-Objekten einzusetzen. Hierfür kann beispielsweise das „Logger“-Paket von „OraOpenSource“ verwendet werden. Auch Oracles **Auditing**-Funktionalität eignet sich zur Kontrolle der Laufzeit.
- In puncto Infrastruktur ist es empfehlenswert, **Sicherheitspatches** zeitlich einzuspielen. Auch der Einsatz von **TLS** und **Security-HTTP-Headern** eignet sich beim Einsatz von APEX.
- Während der Entwicklung kann es nützlich sein, Seiten oder Anwendungen kurzfristig zu kopieren. In der Produktionsumgebung haben diese Kopien allerdings nichts zu suchen. Sonst entsteht „Schattencode“, der leicht übersehen wird.
- Während des Entwicklungsprozesses sollte stets selbstkritisch und aufmerksam programmiert werden. Ebenfalls ist es sinnvoll, **Code-Reviews** einzuführen, vier Augen sehen bekanntlich mehr als zwei.

Fazit

Die in diesem Artikel beschriebenen Szenarien bilden lediglich die Grundlage für ein sicherheitsbewusstes Entwickeln mit APEX. Für tiefere Einblicke sei beispielsweise das Buch „Expert. Oracle

Application Express Security“ von Scott Spendolini empfohlen. Trotz des Alters (Erscheinungsjahr 2013) ist das Buch auf moderne APEX-Versionen anwendbar und relevant.

Darüber hinaus ergibt es ebenfalls Sinn, sich mit Sicherheitsentwicklungen für das Web zu beschäftigen. Die Organisation „OWASP“ stellt beispielsweise regelmäßig eine Top-10-Gefahrenliste zusammen. Letztendlich entstehen bei APEX auch nur Webseiten, sodass viele der genannten Konzepte übertragbar sind.

Für besonders kritische Anwendungen, die beispielsweise über das Internet erreichbar sind, kann es vorteilhaft sein, Penetrationstests und Audits durch Dritte durchführen zu lassen.

Quellen

- [1] Vgl. Oracle (2021), abrufbar unter: <https://docs.oracle.com/en/database/oracle/application-express/21.2/htmig/about-apex-architecture.html>
- [2] Vgl. Scott Spendolini (2013), Expert Oracle Application Express Security. Apress. ISBN: 978-1-4302-4732-6 (S. 129 – 139).
- [3] Vgl. ebd. (S. 139 – 151).

Über den Autor

Philipp Hartenfeller ist Senior Berater bei der MT AG und mit dem Schwerpunkt Oracle Application Express tätig. Neben Datenbanken interessiert er sich für Technologien rund um das Web wie Frontend-Frameworks, Backends mit Node.js oder Go und User Experience.



Philipp Hartenfeller
philipp.hartenfeller@mt-ag.com

Oracle Datenbanken Monthly News

Auf dem deutschsprachigen Oracle-Blog ist die Juni-Ausgabe der News-Serie erschienen.

DOAG Online

Das sechsköpfige Redaktionsteam von Oracle Deutschland hat wieder Neuigkeiten rund um die Datenbank zusammengetragen, die dieses Mal von **Sinan Petrus Toma** in einem ca. 15 minütigen Video präsentiert werden.

Wie immer geht es dabei nicht nur um aktuelle Informationen wie Ankün-

digungen usw., sondern auch um aktuelle Events (mit Terminen), Releases & Patches und interessante Informationen aus dem Web.

In der aktuellen Ausgabe wird wieder ein zusätzliches Quick Link Posting (in Englisch) mit den Links zur Verfügung gestellt, um einen schnellen Zu-

griff auf die zugehörigen Links zu gewährleisten.

<https://www.doag.org/de/home/news/oracle-datenbanken-monthly-news-11/>





MySQL Database Service & HeatWave – „Out of the Box Analytics“ für MySQL

Mario Beck & Carsten Thalheimer, Oracle MySQL GBU

Im Frühjahr 2020 wurde mit MySQL 8 auf Basis der Oracle-Cloud-Infrastruktur (OCI) der MySQL-Cloud-Service komplett erneuert und um einen zusätzlichen Analytics Service erweitert. Der Analytics Service basiert auf der von den Oracle Labs entwickelten neuen HeatWave Storage Engine für MySQL (auch RAPID genannt für „**R**apid **A**nalytics **P**rocessing **I**n **D**RAM“). Diese ermöglicht effiziente SQL-In-Memory-basierte Analysen für komplexe Real-Time-Datenanalyseszenarien und erweitert die klassische Echtzeit-Transaktionsverarbeitung von MySQL (OLTP) um komplexe Online-Analyseanwendungsfälle (OLAP). Mit HeatWave tritt MySQL in direkten Wettbewerb zu Amazon Redshift, Google BigQuery oder auch Azure Synapse.

Knapp 11 Jahre ist es her, dass Sun Microsystems (die ihrerseits MySQL 2008 übernommen hatten) von Oracle akquiriert wurde. MySQL hat seitdem regelmäßig etwa alle drei Jahre neue Versionen (zu-

letzt MySQL 8 im Jahr 2018 [1]) inklusive Erweiterungen (z.B. InnoDB Cluster) vorgestellt. MySQL wurde mit zahlreichen Preisen ausgezeichnet (MySQL Database of the Year 2019 [2]) und kann viele der

größten Internetfirmen als Referenzen benennen [3].

Trotzdem hatte es MySQL schwer – keiner der zahlreichen MySQL-Erfolge der letzten Jahre schaffte eine Erwähnung in

den Oracle Earningscalls [4]. Das änderte sich erst, als Larry Ellison persönlich HeatWave Anfang Dezember 2020 vorstellte und direkt mit einigen Superlativen umwarb.

MySQL InnoDB, OLTP und OLAP

Die InnoDB Storage Engine wurde bereits 2011 mit MySQL 5.5 als MySQL Standard definiert. InnoDB ist ein für OLTP optimiertes Speichersubsystem. Transaktionen umfassen einzelne Datenbankeinträge, die aus mehreren Feldern oder Spalten bestehen. InnoDB gewährleistet, dass viele parallele Transaktionen bei mehreren Hundert Verbindungen einfach und ACID-konform zeilenbasiert gespeichert werden können, und bietet hierbei transaktionssichere Zugriffe bis zu einer theoretischen Tabellengröße von 64 TB. InnoDB baut hierbei auf den Aries-Algorithmen [5] auf, die eine Wiederherstellung nach einem Fehlerfall gewährleisten. Um die Daten dauerhaft zu persistieren, ist eine Schreiboperation immer auch an eine Schreiboperation auf der Festplatte gebunden. Bei OLTP liegt der Schwerpunkt auf einer schnellen Transaktionsverarbeitung (im Bereich von Millisekunden), da einzelne Datensätze häufig gelesen, geschrieben und aktualisiert werden.

MySQL-Abfragen, die sehr große Datenmengen referenzieren, müssen allerdings oft zunächst erst von der Festplatte gelesen werden und erzeugen durch den zeilenbasierten Zugriff einen ineffektiven Overhead (es werden immer ganze InnoDB-Blöcke gelesen). Durch komplexe Abfragen großer Datenmengen (OLAP) beispielsweise bei einem Datawarehouse kommt das InnoDB-System oft an seine Grenzen. Abfragen können teils sehr lange laufen. Gerade für analytische Abfragen hat sich daher der Anwendungsfall der spaltenbezogenen Datenspeicherung etabliert.

Hier setzt MySQL HeatWave an: HeatWave ist ein neues Speichersubsystem, das die Daten zusätzlich zum InnoDB-System in einem eigenen Backend spaltenbasiert und optimiert ablegt. Die Synchronisation zwischen MySQL InnoDB und MySQL HeatWave erfolgt für konfigurierte Datenbanken oder Tabellen automatisch nahezu in Echtzeit. Der bei anderen Lösungen notwendige ETL-Prozess (Extract, Transform, Load) entfällt hierbei.

MySQL-HeatWave-Aufbau

Ein HeatWave-Cluster besteht aus einem MySQL-DB-Systemknoten und zwei bis 64 spezialisierten HeatWave-Backend-Knoten. Der Systemknoten ist ein normales MySQL-8.0-System. Die HeatWave-Knoten können nicht direkt angesprochen

werden, sondern sind ausschließlich mit dem MySQL-DB-Systemknoten verbunden. Der Systemknoten enthält ein zusätzliches HeatWave-Plug-in, das für das Clustermanagement, die Abfrageplanung und die Rückgabe von Abfrageergebnissen an das MySQL-DB-System verantwortlich ist. HeatWave-Knoten kopieren die MySQL-Daten des Systemknotens in ihren Arbeitsspeicher und verarbeiten Abfragen. Daten sind also immer sowohl in InnoDB als auch in den HeatWave-Knoten gespeichert.

Der MySQL Parser des Systemknotens ist mit MySQL 8.0 identisch. Das verwendete Protokoll für MySQL- und HeatWave-Anfragen ist identisch. Alle bestehenden Applikationen, die MySQL 8 unterstützen, können folglich auch ohne Änderung mit MySQL HeatWave verwendet werden. Die Kombination aus MySQL-Systemknoten und MySQL HeatWave Backend nennt Oracle einen MySQL-HeatWave-Cluster (siehe Abbildung 1).

Stand Mai 2022 werden der MySQL-Systemknoten und die HeatWave-Knoten als Instanzen auf Basis der Oracle Cloud (OCI) angeboten. Eine Instanz umfasst 16 OCPUs (AMD E3) und 512 GB Arbeitsspeicher. Eine einzelne HeatWave-Instanz adressiert etwa 800 GB MySQL-Daten (komprimiert), die kleinste Ausbaustufe eines HeatWave-Clusters (1x Systemknoten und 2x HeatWave Backend) unterstützen somit bereits rund 1600 GB MySQL-Daten. Die 16 OCPUs/512 GB einer Instanz erzeugen Kosten von rund 230 € pro Mo-

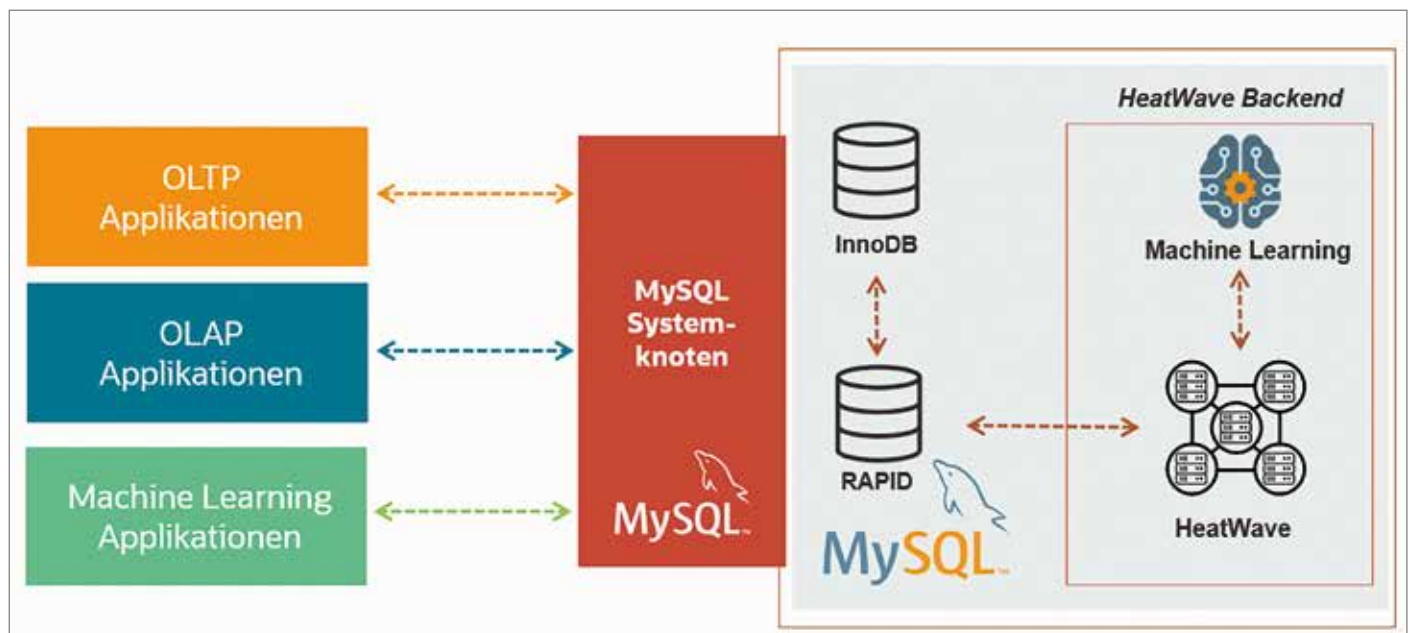


Abbildung 1: MySQL-HeatWave-Komponenten (Quelle: Carsten Thalheimer)

nat. MySQL-Systemknoten können auch autark verwendet werden. Da die MySQL-Systemknoten die Daten persistieren, können HeatWave-Instanzen dynamisch bei (OLAP-)Bedarf gestartet beziehungsweise gestoppt werden.

MySQL-HeatWave-Cluster einrichten

Das Aufsetzen eines MySQL-HeatWave-Clusters kann OCI-typisch über viele Wege erfolgen. Die meistgenutzten Methoden sind hierbei die Verwendung des OCI GUI oder der OCI-Kommandozeile. Das Aufsetzen eines MySQL-HeatWave-Clusters erfolgt zunächst sehr ähnlich zum Aufsetzen einer MySQL-Datenbank. Folgend werden wir das Aufsetzen der Instanz in einem Gui beschreiben.

In dem OCI Gui findet man unter Databases | MySQL | DB Systems die Schaltfläche zum Erstellen einer neuen MySQL-Instanz „Create DB System“. Hier müssen zunächst einige offensichtliche Parameter vergeben werden:

- Compartment, Name der Instanz, Beschreibung
- Anwendername und Passwort des Administrators
- Netzwerkeinstellungen
- Backup-Plan
- Availability Domain/Fault Domain
- Verwendete Hardware/Shapes

Ähnlich wie bei einer MySQL-Datenbankinstanz stehen auch erweiterte Optionen zur Verfügung. Der Unterschied zwischen einer MySQL-Datenbankinstanz und einem MySQL-Systemknoten ergeben sich nur aus der offensichtlichen Auswahl zwischen Standalone, High Availability und HeatWave (nach dem Beschreibungsfeld). Zu beachten ist, dass bei einem HeatWave-System auch nur die HeatWave-spezifische Shape (64 OCPU / 512 GB RAM) gewählt werden kann.

Das Aufsetzen des MySQL-Systemknotens ist innerhalb von etwa fünf Minuten erledigt.

Diese Instanz kann nun bereits als klassische MySQL-Datenbank verwendet werden. Wenn ein direkter Zugriff aus dem Internet beziehungsweise aus einem lokalen Netzwerk gewünscht ist, so kann dies mit einem VPN-Zugriff oder

```
SQL > show engines;
EngineSupport  ...
...
InnoDBYES     ...
RAPID YES     ...
MyISAMYES     ...
...

SQL > SHOW STATUS LIKE 'rapid_service_status';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| rapid_service_status | ONLINE |
+-----+-----+

SQL > show variables like 'secondary_engine_cost_threshold';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| secondary_engine_cost_threshold | 100000.000000 |
+-----+-----+

SQL > show variables like 'rapid_compression';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| rapid_compression | ON |
+-----+-----+

SQL > SELECT VARIABLE_VALUE FROM performance_schema.global_status WHERE
VARIABLE_NAME = 'rapid_change_propagation_status';
+-----+
| VARIABLE_VALUE |
+-----+
| ON |
+-----+
```

Listing 1: HeatWave und RAPID-Status

```
SQL > CALL sys.heatwave_load(JSON_ARRAY("tpch_10"),NULL);
...
+-----+
| LOAD SUMMARY |
+-----+
| SCHEMA TABLES TABLES COLUMNS LOAD |
| NAME LOADED FAILED LOADED DURATION |
+-----+-----+-----+-----+
| `tpch_10` 8 0 61 62.65 s |
+-----+-----+-----+-----+
```

Listing 2: Daten-Laden in HeatWave

```
SQL > SELECT NAME, LOAD_STATUS from performance_schema.rpd_tables,per-
formance_schema.rpd_table_id
where rpd_tables.ID = rpd_table_id.ID;
+-----+-----+
| NAME | LOAD_STATUS |
+-----+-----+
| tpch_10.lineitem | AVAIL_RPDGSTABSTATE |
...
| tpch_1.orders | AVAIL_RPDGSTABSTATE |
+-----+-----+
```

Listing 3: Load Status HeatWave

```

SQL > explain select * from tpch_10.nation;
+----+...+-----+
| id |...| Extra |
+----+...+-----+
| 1  |...| NULL  |
+----+...+-----+
SQL > explain select * from tpch_10.customer;
+----+...+-----+
| id |...| Extra |
+----+...+-----+
| 1  |...| Using secondary engine RAPID|
+----+...+-----+

```

Listing 4: Erwartete ausführende Engine

```

SQL > use tpch_10;
SQL > set @@use_secondary_engine=off; ###no default ###
SQL > SELECT l_returnflag, l_linestatus,
      sum(l_quantity) as sum_qty,
      sum(l_extendedprice) as sum_base_price,
      sum(l_extendedprice * (1 - l_discount)) as sum_disc_price,
      sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) as      sum_
charge,
      avg(l_quantity) as avg_qty,
      avg(l_extendedprice) as avg_price,
      avg(l_discount) as avg_disc,
      count(*) as count_order
FROM lineitem
WHERE l_shipdate <= date '1998-12-01' - interval '90' day
GROUP BY l_returnflag,l_linestatus
ORDER BY l_returnflag, l_linestatus;
... Resultset xyz ...
4 rows in set (1 min 47.5718 sec)

SQL > set @@use_secondary_engine=on; ### default ###
SQL > SELECT l_returnflag, l_linestatus, ...
... Resultset xyz ...
4 rows in set (0.8659 sec)

SQL > set @@use_secondary_engine=on; ### default ###
SQL > SELECT l_returnflag, l_linestatus, ...
... Resultset xyz ...
4 rows in set (0.2480 sec)

```

Listing 5: Query 1 TPCH 10 GB

einem Loadbalancer [6] konfiguriert werden. Auch der Import von Daten in MySQL erfolgt via MySQL-Standardtools wie zum Beispiel der MySQL Shell [7].

Bis jetzt ist allerdings nur der MySQL-Systemknoten aufgesetzt. In einem zweiten Schritt muss diesem jetzt noch das MySQL HeatWave Backend zur Verfügung gestellt werden.

Im Menü der zuvor aufgesetzten MySQL-Instanz gibt es einen Eintrag HeatWave. Dieser stellt dem Benutzer die Informationen zu dem angebotenen HeatWave Backend zur Verfügung. Bei neu aufgesetzten Systemen gibt es hier

nur die Schaltfläche „Add HeatWave Cluster“. Es gibt die Möglichkeit, das optimale HeatWave Backend auf Basis der Datenbanken „berechnen“ zu lassen. Alternativ kann man aber auch eine Anzahl von Instanzen definieren. Wie erwähnt adressiert bereits die Minimum-HeatWave-Konfiguration 1,6 TB MySQL-Daten. Auch die Daten des HeatWave Backend werden, für den Anwender nicht transparent, persistiert. Daher kann die Anzahl der HeatWave-Knoten zu jeder Zeit dynamisch online erhöht oder reduziert werden, um Kapazität oder Performance anzupassen. Auch ein Neustart des HeatWave Backend

liest die Daten zunächst vom lokalen Speicher und synchronisiert Änderungen mit dem MySQL-Systemknoten in einem zweiten Schritt. Dadurch sind Backend-Knoten schnellstmöglich einsatzbereit.

Datenbanken und Tabellen mit MySQL HeatWave synchronisieren

Nun kann man sich in einem ersten Schritt mit der Umgebung vertraut machen. Wichtig: Intern heißt „HeatWave“ leider immer noch „RAPID“.

Die in Listing 1 gezeigten Ergebnisse zeigen uns, dass ein HeatWave Backend zur Verfügung steht, die Datenkomprimierung aktiviert ist und jegliche Datenänderung in Echtzeit auf das HeatWave Backend synchronisiert wird. Ab einer Query Cost von 100.000 werden Abfragen nicht mehr von InnoDB, sondern von RAPID ausgeführt. Die Betriebsdokumentation zu HeatWave ist in der Dokumentation der OCI hinterlegt [8]. Details zum Datenhandling findet man im HeatWave User Guide [9]. Grundsätzlich kann das Datenhandling manuell oder automatisiert erfolgen. Das MySQL-Team hat hierzu bereits Anfang des letzten Jahres die HeatWave-Autopilot-Funktionen zur Verfügung gestellt. Autopilot ist eine Sammlung von Stored Procedures, die viele alltägliche Aufgaben stark vereinfachen. Wenn möglich verwenden wir folgend daher die Autopilot-Funktionen.

Die Daten eines neu aufgesetzten MySQL-Systemknotens stehen zunächst noch nicht in HeatWave zur Verfügung. Zunächst müssen die gewünschten Daten mit dem HeatWave Backend synchronisiert werden. Hierzu gibt es eine einfache Autopilot Stored Procedure. Mittels `heatwave_load` kann man sehr einfach eine Datenbank beziehungsweise Teile einer Datenbank optimiert laden. Wenn möglich werden die Daten parallelisiert geladen und wenn nötig konvertiert. Sobald die Daten geladen sind, stehen diese vektorisiert, spaltenbasiert, komprimiert und verteilt auf allen HeatWave-Instanzen im Arbeitsspeicher zur Verfügung. Für größere Datenmengen kann man eine Ladegeschwindigkeit von etwa 10 - 15 GB pro Minute (bzw. 1 TB/h) kalkulieren (siehe Listing 2). Sobald die Daten geladen sind, werden sie automatisch bei Änderungen

synchronisiert (siehe 'rapid_change_propagation_status').

Für die folgenden Beispiele wurden die Standard-TPCH-Daten verwendet [10].

Eine Auflistung der in HeatWave geladenen Tabellen erhält man durch die Abfrage des Load Status (siehe Listing 3).

Queries auf MySQL HeatWave

Das Verhalten von MySQL gegenüber der Anwendung bleibt komplett unverändert. Daher unterstützt HeatWave alle existierenden MySQL-Applikationen.

Wenn MySQL eine Query-Anfrage erhält, so werden die Kosten dieser Abfrage im MySQL Optimizer abgeschätzt. Sollten die Kosten der Query unter 100.000 liegen oder das HeatWave Backend nicht erreichbar sein (z.B. weil es aus Kostengründen heruntergefahren wurde), so wird die Abfrage auf dem Standard-MySQL-InnoDB-System ausgeführt. Das Verhalten kann man recht einfach mittels explain nachvollziehen (siehe Listing 4). Während das erste Beispiel nur ein paar Datensätze adressiert und daher recht einfach ist, wird im zweiten Beispiel aufgrund der großen Anzahl der Daten bereits für diese sehr einfache Query im Hintergrund RAPID verwendet. Wenn gewünscht, kann man diesen Automatismus ausschalten (set@@use_secondary_engine=on|off).

Natürlich sind die Ergebnisse bei InnoDB und RAPID identisch.

Wenn eine Query auf HeatWave ausgeführt wird, führt das in der Regel zu einer erheblichen Performancesteigerung. Für die Query 1 des TPCH-Benchmarks (10 GB) kommt es zu folgendem Ergebnis:

Die Query 1 wurde in Listing 5 dreimal ausgeführt – zunächst nur auf InnoDB. Trotz mehrmaligen Ausführens pendelte sich die Ausführungszeit bei etwa 107 Sekunden ein. Dann wurde die Query 1 mit eingeschaltetem HeatWave Backend und unter einer Sekunde ausgeführt. Schon hier sieht man eine etwa 100-fache Beschleunigung. Ein nochmaliges Ausführen der Query involviert den internen HeatWave-ML-Algorithmus, der auch Fragmente bereits ausgeführter Abfragen erkennt und optimiert. Es zeigt sich eine weitere Verbesserung der Query-Performance.

In der Praxis wird tatsächlich noch ein sehr viel größerer Unterschied erkennbar

sein, da im obigen Beispiel bedingt durch die kleine Datenmenge auch von InnoDB die Daten aus dem InnoDB Buffer (Arbeitsspeicher) und nicht vom lokalen Storage gelesen wurden.

Zusammenfassung

Wenn man ein HeatWave-Cluster aktiviert, werden Analyseabfragen automatisch vom MySQL-DB-System auf den HeatWave-Cluster verlagert, um die Verarbeitung signifikant zu beschleunigen. Dadurch können Sie Online-Transaktionsverarbeitung (OLTP), Online-Analyseverarbeitung (OLAP) und gemischte Arbeitslasten von derselben MySQL-Datenbank aus ausführen, ohne dass ETL erforderlich ist und ohne Ihre Anwendungen zu ändern. Durch den Wegfall des ETL-Prozesses vereinfacht sich die Datenhaltung, die Kosten reduzieren sich und die Sicherheit wird verbessert.

Zusammengefasst ist MySQL HeatWave ein Abfrage-Beschleunigungstool. Dies öffnet allerdings neue Anwendungsfälle. MySQL-Datenbanken sind in der Regel derzeit nicht viel größer als ein paar Terabyte [11]. Einer der Gründe hierfür sind die Einschränkungen der Ausführungszeiten bei komplexen Abfragen. Mit der Einführung von HeatWave werden wir eine Verschiebung der Datenbankgrößen sehen. Derzeit unterstützt MySQL HeatWave bereits 50+ TB, aber auch hier sind dem System noch keine wirklichen Grenzen gesetzt.

Zusätzlich wird die Abfragebeschleunigung neue Applikationen im Umfeld von BI (Tableau, PowerBI, Looker etc.) anziehen. Alle diese Applikationen unterstützen MySQL bereits und legen die Verwendung von HeatWave sehr nahe.

Zusätzlich zur Abfragebeschleunigung stellt HeatWave auch eine kostenfreie Machine-Learning-Funktionalität zur Verfügung. HeatWave ML analysiert die Eigenschaften der Daten und erstellt ein optimiertes maschinelles Lernmodell, das zur Erstellung von Vorhersagen (Predictions) und Erklärungen (Explanations) verwendet werden kann.

MySQL HeatWave steht derzeit nur in der Oracle Cloud zur Verfügung. Allerdings kann durch die Verwendung der MySQL-Replikation (in der Oracle Cloud Channels genannt) sehr einfach ein lokales MySQL in die Cloud repliziert werden. Auch wenn sich durch diesen Aufbau Netzwerk-Laten-

zen ergeben, so ist die Ausführung von komplexen Abfragen mittels HeatWave immer noch sehr viel schneller als mit dem Standard-InnoDB-System.

Auch im Vergleich zu den Systemen der Mitbewerber (Snowflake, Amazon Redshift, Amazon Aurora) ist die Performance erheblich besser [12].

Quellen

- [1] What's New in MySQL 8.0? <https://mysqlserverteam.com/whats-new-in-mysql-8-0-generally-available/>
- [2] MySQL is the DBMS of the Year https://db-engines.com/en/blog_post/83
- [3] MySQL Customers <https://www.mysql.com/customers/>
- [4] Earnings Call Transcript: <https://www.fool.com/earnings/call-transcripts/2020/12/11/oracle-orcl-q2-2021-earnings-call-transcript/>
- [5] ARIES [https://de.wikipedia.org/wiki/ARIES_\(Informatik\)](https://de.wikipedia.org/wiki/ARIES_(Informatik))
- [6] <https://docs.oracle.com/en-us/iaas/mysql-database/doc/networking-setup-mysql-db-systems.html>
- [7] <https://docs.oracle.com/en-us/iaas/mysql-database/doc/importing-and-exporting-databases.html>
- [8] <https://docs.oracle.com/en-us/iaas/mysql-database/doc/heatwave.html>
- [9] <https://dev.mysql.com/doc/heatwave/en/>
- [10] <https://www.tpc.org/tpch/>
- [11] <https://blog.koehntopp.info/2022/02/16/databases-how-large-is-too-large.html>
- [12] <https://www.oracle.com/mysql/heatwave/performance/>



Mario Beck
mario.beck@oracle.com



Carsten Thalheimer
carsten.thalheimer@oracle.com



Sicherung der Akronyme: NFTs, Web3, Crypto, Blockchain und das Metaverse

Anna Collard, SVP Content Strategy & Evangelist bei KnowBe4 Africa

Das Metaverse ist zwar im Moment noch ein Konzept, wird aber in nicht allzu ferner Zukunft Realität werden. Es besteht aus den Möglichkeiten, die sich ergeben, wenn man die Fortschritte und die Erreichbarkeit der erweiterten Realität wie AR und VR sowie virtuelle Spielwelten mit dem dezentralen Charakter und der Kompositionsfähigkeit von Web3,

crypto-Assets, dezentraler Finanzierung (DeFi) und der zugrunde liegenden Blockchain-Technologie kombiniert.

Eine neue Welt also, gehüllt in Blockchain und von NFTs gefärbt. Es soll das menschliche Engagement und die Interaktionen verändern und die Grenzen der Kommerzialisierung verschieben. Es ist auch eine ganz neue Welt, mit Sicher-

heitsrisiken, Schwachstellen und berechtigten Bedenken der Nutzer. Denn so sehr diese Innovationen die Grenzen der menschlichen Interaktion sprengen können, so sehr bieten sie auch neue Möglichkeiten für Betrug, Cyberkriminalität und Scams.

Wir sind uns nicht sicher, wohin Metaverse, Web3 und NFTs führen werden

oder ob Blockchain eine praktikable Lösung zur Unterstützung dieser Technologien bleiben wird. Aber was man jetzt schon feststellen kann, ist, dass diese Umgebungen bessere Ansätze für die Sicherheit benötigen. Es wird viel Geld in diesen Bereich investiert und es wird auch viel Geld gestohlen. So hat beispielsweise das Blockchain-Analyseunternehmen Elliptic berichtet, dass DeFi-Plattformen bis heute 12 Milliarden Dollar verloren haben. Auch wenn im Moment vieles als Hype oder überbewertet erscheint, müssen wir Wege finden, um die Verbraucher, Nutzer und Investoren dieser Ökosysteme besser zu schützen.

Smart Contracts, NFTs, dApps und Kryptowährungen sind Bausteine des zukünftigen Metaversums und müssen genauer untersucht werden. Ob Modeerscheinung oder nicht, das Metaversum und die damit verbundenen Technologien können sich potenziell so entwickeln, dass sie Organisationen zusätzliche Handlungsspielräume und Einzelpersonen neue Welten eröffnen. Das bedeutet, dass die Sicherheit eine führende Rolle spielen muss, um sicherzustellen, dass die Milliarden, die investiert werden, nicht zu den Milliarden werden, die gestohlen werden.

Die Designprinzipien des Metaversums machen Dinge komponierbar, das heißt, sie können in verschiedenen Welten wiederverwendet werden. Nun, theoretisch.

Dezentralisierung ist auch hier ein wichtiger Bestandteil, aber dieses Konstruktionsprinzip bringt eine Reihe von Herausforderungen mit sich. Blockchain entstand aus dem Wunsch heraus, der skrupellosen Finanzdienstleistungsindustrie die Kontrolle über das Finanzwesen zu entziehen, die durch die Finanzkrise von 2008 ausgelöst wurde. Das dezentrale Finanzwesen oder „DeFi“ birgt jedoch das Risiko, die gleichen Sünden des ursprünglichen unregulierten Bankensystems zu wiederholen und macht es dann noch schlimmer, indem es beispielsweise Kriminelle und Finanzbetrug ermöglicht. Die Hauptinvestoren sind Risikokapitalgeber (Venture Capitalists), und die größten Mining-Pools werden derzeit von einer Handvoll Unternehmen kontrolliert. Das Ökosystem ist also gar nicht so dezentralisiert, wie es dargestellt wird. Bis

sie ihr volles Potenzial entfalten können, ist es noch ein langer Weg, der ständige Aufmerksamkeit für Ethik, Sicherheit und Kontrolle erfordert.

Betrug und Kriminalität wird es immer geben, vor allem, wenn die Systeme dezentralisiert sind oder wenn wir uns zu sehr auf Codestücke wie smart contracts verlassen. Letzten Endes sind diese von Menschen geschriebene Codes und Menschen machen Fehler. Auf jeder Ebene gibt es Möglichkeiten, Schwachstellen auszunutzen. Wir dürfen auch nicht vergessen, dass Kriminelle oft die Nutzer ins Visier nehmen, weil sie am einfachsten anzugreifen sind. Das bedeutet nicht, dass die Welt nicht vorankommen und diese Möglichkeiten erforschen sollte, sondern nur, dass wir Entwickler und Nutzer aufklären und sicherstellen müssen, dass sie sich der Risiken bewusst sind und dass die Sicherheit vom ersten Schritt an miteinbezogen wird.

Über die Autorin

Anna Collard ist SVP of Content Strategy & Evangelist bei KnowBe4. Seit 18 Jahren ist sie in der Cybersecurity tätig: als Geschäftsführerin von KnowBe4 Africa und davor als Sicherheitsarchitektin und Beraterin, wo sie Organisationen in Südafrika, Europa und den USA beim Aufbau ihrer Cybersecurity-Abwehr unterstützt hat. Sie gründete die Security-Awareness-Firma Popcorn Training, die 2018 von KnowBe4 acquired wurde. Anna gewann den „Women in Tech Innovations Throughout Africa 2020 Award“ für das südliche und zentrale Afrika. Sie wurde in die „Top 50 Women in Cybersecurity – Africa 2020“ sowie in die „Top 100 Women in Cyber 2020 und 2021“ weltweit vom Cyber Defence Magazine aufgenommen. Sie wurde mit dem prestigeträchtigen „ISACA South Africa President Award“ für 2020 ausgezeichnet. Collard ist Inhaberin verschiedener Sicherheitszertifikate, wie CISSP, CISA, ISO 27001 Lead Auditor, CIPP/IT und war früher Visa/Mastercard PCI DSS QSA.



Anna Collard
annac@knowbe4.com



Welcome back in Nürnberg: #DOAG2022

Auf ein Neues zum großen Wiedersehen

Martin Meyer, Redaktionsleitung Red Stack Magazin

Die DOAG 2022 Konferenz + Ausstellung – wir berichteten – findet dieses Jahr vom 20. bis 23. September 2022 wieder als Präsenzveranstaltung im Nürnberg Convention Center statt – zwei Monate früher als gewohnt!

Der Auftakt am 20. September ist einem Thementag zu „Database Migrations“ gewidmet, die **klassischen Konferenztage sind am 21. und 22. September**. Am 23. September findet abschließend noch ein Schultag statt. Freuen Sie sich auf 350 Sessions zu den Themengebieten **Datenbank & Infrastruktur, Development & Middleware, Data Analytics & AI** sowie **Strategie & Innovation** in Form von Fachvorträgen, Erfahrungsberichten und Best Practices.

Die seit Jahrzehnten im Kalender der Mitglieder, IT- und Business-Experten, Unternehmen und Fachkräfte im November eingetragene große DOAG-Konferenz erwartet Ihre Besucher diesmal im Spätsommer. Endlich heißt es wieder für Referenten und Referentinnen Bühnenluft zu schnuppern und dabei einem *echten* analogen Publikum im Saal gegenüberzustehen. Digital kann als Übertragungsweg natürlich gut sein, aber analog ist am

Ende, das was bei den meisten Teilnehmern und Besuchern besser ankommt.

Networking und persönliche Gespräche vor Ort haben vielen von uns bei digitalen Online-Events in der Pandemiezeit gefehlt und daher gibt es jetzt auch wieder eine DOAG Konferenz + Ausstellung mit Live-Vorträgen, Community-Aktivitäten, Abendevent und vielen persönlichen Gesprächen, die diesmal sogar bei angenehmen Temperaturen auf der Terrasse stattfinden könnten.

Wir freuen uns wieder attraktive Vorträge für ein breites Konferenzprogramm gewonnen zu haben. Zum Auftakt des ersten klassischen Konferenztages am Mittwoch erwartet Sie die spannende Session Keynote von Gerald Venzl (Oracle) „*Neuerungen und coole Features für die Entwickler in der Oracle-Datenbank*“. Des Weiteren berichtet Joey Kelly in seiner Keynote „*NO LIMITS – Wie schaffe ich mein Ziel*“ eindrucksvoll über seinen Lebensweg als Unternehmer und Ausdauersportler, den er mit Beharrlichkeit, Zielen, Willen und Leidenschaft konsequent verfolgt.

Erwartet werden etwa 1.200 Teilnehmer und 30 Aussteller, die die 1.200 m² Ausstellungsfläche auf den vier Ebenen

zur Verfügung haben. Hersteller und Dienstleister – Oracle-Partner sowie Mitbewerber – präsentieren wieder Neuigkeiten aus der Oracle-Welt und stellen ihre Produkt- und Leistungsangebote vor. Für einen optimalen Erfahrungsaustausch werden auch verwandte Technologien, Open-Source, aktuelle Trends und Methoden sowie Soft Skills präsentiert. Für die eine oder andere Überraschung sorgen die zahlreichen Aktivitäten, Verlosungen und Gewinnspiele an den Ständen.

Seien Sie auch dabei! #DOAG2022

Achtung: Tickets zum attraktiven Frühbucherpreis gibt es bis einschließlich 26.07.2022.

<https://anwenderkonferenz.doag.org/de/home/>

BUSINESS INSIGHTS

NEWS

04/2022



Chancen des Lieferkettengesetzes durch Planungsprozesse frühzeitig erkennen

Dr. Birgit Breitschuh, Oliver Wight und Jochen Rahm, PROMATIS Gruppe, Ettlingen (TechnologieRegion Karlsruhe)

Die Kurzsicht des strategischen Einkaufs hat über Jahre hinweg zu einem Blindflug geführt – ist der Zuliefermarkt doch oftmals eine Black Box. Wen wundert da die Notwendigkeit eines Lieferkettengesetzes, in dessen Fokus nun der weltweite Schutz der Arbeitnehmer und der Umwelt steht, um einen gesetzlichen Rahmen für die Sorgfaltspflicht entlang der globalen Lieferkette zu schaffen. Für Unternehmen führt das zukünftig zu neuen Pflichten, denn Lieferanten müssen auf die Einhaltung sozialer und ökologischer Mindeststandards überprüft werden. Welche Herausforderungen sich konkret für Unternehmen hieraus ergeben und warum gerade Planungsprozesse eine wichtige Rolle dabei spielen, erläutert der vorliegende Beitrag.

Beim Lieferkettengesetz stehen in erster Linie die Themen Menschenrechte und Nachhaltigkeit im Fokus. Laut einer Studie für das Bundesministerium für wirtschaftliche Zusammenarbeit und Entwicklung ist gerade in Deutschland die Relevanz besonders spürbar, denn deutsche Wertschöpfungsketten nehmen eine elementare Rolle ein. Nach Schätzung des Global Slavery Index ist Deutschland unter den G20-Staaten der drittgrößte Importeur (hinter den USA und Japan) von „Hochrisiko-Waren“ und bezieht Waren im Wert von 29,6 Mrd. US-Dollar [1].

Wenn wir auf die Historie des Lieferkettengesetzes zurückblicken, stellen wir fest, dass bereits im Dezember 2016 ein Nationaler Aktionsplan „Wirtschaft und Menschenrechte“ verabschiedet wurde und man über mehrere Jahre auf freiwilliges Engagement der Unternehmen vertraute [2]. Im Zuge eines unabhängigen Monitoring-Verfahrens wurde nach der Validierung im Jahr 2020 dann deutlich, dass weniger als 20 % der deutschen Unternehmen den Plan in die Tat umgesetzt hatten und somit die freiwillige Einhaltung nicht zum Erfolg geführt hat. Denn eines ist klar, die Realisierung der Anforderungen führt zu Verschiebungen in puncto Lieferantenauswahl. Die Auswahl eines anderen Produkts sowie Lieferanten führt letztendlich zur Verzerrung des Wettbewerbs. Mit der Verabschiedung des Lieferkettengesetzes am 11. Juni 2021 soll nun

eine einheitliche Basis geschaffen und somit die Balance des Wettbewerbs wiederhergestellt werden. Für die Umsetzung wurde ein Stufenplan verabschiedet. Ab 2023 sind alle Unternehmen mit mehr als 3.000 Mitarbeiter davon tangiert – diese Stufe betrifft Stand heute ca. 900 Unternehmen in Deutschland. Ab dem Jahr 2024 werden die Maßnahmen des Lieferkettengesetzes dann für alle Unternehmen mit einer Mitarbeiterzahl >1.000 relevant (ca. 4.800 Unternehmen in Deutschland). Nicht zu vernachlässigen ist an dieser Stelle, dass die Anzahl an Zeitarbeitskräften in dem Zusammenhang ebenfalls eine wichtige Rolle spielt.

Somit sollen nun die Pflichten durch die Unternehmen in ihrem eigenen Geschäftsbereich sowie gegenüber ihren unmittelbaren Zulieferern umgesetzt werden. Mittelbare Zulieferer müssen ebenfalls mit einbezogen werden, sobald das Unternehmen von Menschenrechtsverletzungen auf dieser Ebene „substantiierte Kenntnis“ erhält.

Gesetzliche Grundlagen und Inhalt

Das Gesetz bringt weitreichende neue Pflichten für die Geschäftsleitungen betroffener Unternehmen mit sich: Ein zentrales Erfordernis des Gesetzes ist die Durchführung einer Risikoanalyse als notwendige Voraussetzung für die Implementierung eines angemessenen und wirksamen Risikomanagementsystems mit dem Ziel, menschenrechtliche und umweltbe-

zogene Risiken sowie die Verletzung geschützter Rechtspositionen entlang ihrer Lieferketten zu identifizieren, zu verhindern, zu beenden oder zumindest ihr Ausmaß zu minimieren. Darüber hinaus verpflichtet das Lieferkettengesetz zur Einrichtung eines Beschwerdeverfahrens im Unternehmen, das Hinweise auf Risiken und Rechtsverletzungen ermöglicht und damit letztlich die Verpflichtung zu einem Hinweisgebersystem gleichkommt. Hieraus ergeben sich konkret diese vier Aufgaben:

- Risiken entlang der gesamten Wertschöpfungskette identifizieren und bewerten
- Gezielte Gegenmaßnahmen einleiten, um diese Risiken zu minimieren
- Eingeleitete Kontrollmaßnahmen regelmäßig überwachen (eventuell ist Nachjustieren notwendig)
- Kommunikation, Dokumentation und Berichterstattung

Zwiespalt in der Lieferkette

Die Sorgfaltspflichten der Unternehmen erstrecken sich grundsätzlich auf die gesamte Lieferkette – vom Rohstoff bis zum fertigen Verkaufsprodukt. Viele unserer Kunden haben sich bereits damit auseinandergesetzt und beispielsweise ein Statement abgegeben und somit Verantwortung bei der Auswahl und dem Anbau von Rohstoffen übernommen. Das daraus entstandene Ver-

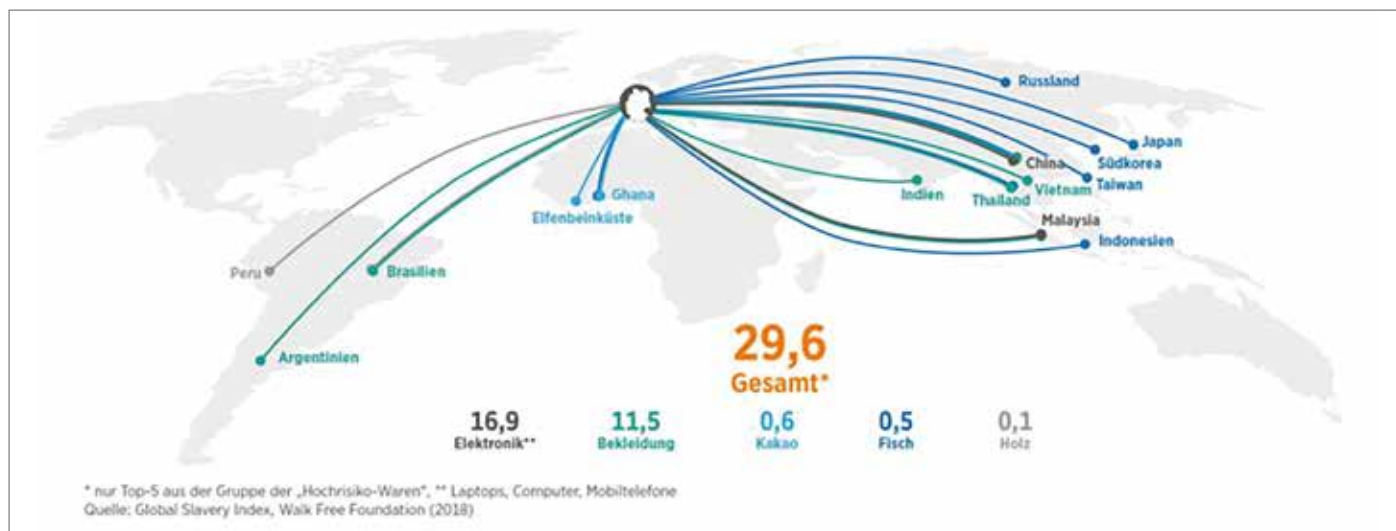


Abbildung 1: Deutsche Lieferketten [Quelle: Global Slavery Index, Walk Free Foundation (2018)]

trauen zum Lieferanten verschafft den Unternehmen einen enormen Wettbewerbsvorteil. In dem Zwiespalt, der sich allerdings hieraus ergibt, gilt es im Endeffekt die Balance zwischen den externen Anforderungen und der Antwort, die man darauf gibt, zu finden. So existieren beim Konsumenten ein erwarteter Kundenlieferservice und ein hoher Anspruch an die Zuverlässigkeit und Liefertreue, damit einhergehend sind bestimmte Erwartungen an den Preis verbunden. Dabei stellt sich die Frage, ob immer der günstige Preis oder der richtige Preis angemessen ist. Dahinter verbirgt sich das Austarieren des gebundenen Kapitals im Unternehmen im Hinblick auf die Bestände und die langen Lieferzeiten. Um diesen Zwiespalt der Lieferkette anzugehen, müssen Prioritäten im Unternehmen gesetzt werden und genau das ist der Link zur Strategie. Wie will ich mich aufstellen, um die richtige Antwort zu geben? Was sind die Anforderungen meiner Kunden, was ist der Anspruch für mein Unternehmen? Um eine effiziente Antwort zu geben, ist eine strategische Ausrichtung unabkömmlich. Am Beispiel des Preises heißt das, auch Gesichtspunkte der Effizienz und Automatisierung ins Spiel zu bringen. Ebenso hängen Reaktionsfähigkeit und Marktausrichtung mit der Lieferanten- und Distributionsstrategie zusammen, aber auch in Bezug auf Service und Qualität kann durch die Gestaltung des Liefernetzwerks eine Optimierung erfolgen.

Wenn allerdings Flexibilität und Agilität im Unternehmen im Fokus stehen, stellt sich die Frage nach der Ausrichtung der Wettbewerbsstruktur, um Vermögenswert und Betriebskapital im Auge zu behalten.

Strategisches Netzwerkdesign mit handhabbarer Komplexität des Liefernetzwerks

Angesichts der Sorgfaltspflichten ist das strategische Design von Lieferkettennetzwerken extrem entscheidend und zu einer der wichtigsten Aufgaben im Unternehmen geworden. Viele Unternehmen haben bereits eine globale Management-Verantwortung für die Supply Chain innerhalb ihrer Organisation eingerichtet. Doch nur eine moderne strategische Netzwerkgestaltung kann sicherstellen, dass eine Lieferkette die Geschäftsstrategie wirklich unterstützt und zu Wettbewerbsvorteilen führen kann. Zunächst stellt sich daher die Frage nach der Gestaltung des heutigen Netzwerks und die Sicherstellung der Transparenz durch die gesamte Wertschöpfungskette. Für die effiziente und gesteuerte Unternehmensentwicklung zur Umsetzung der Strategie ist ebenfalls eine Vorausschau wichtig: Welche Marktveränderungen finden statt? Wie wollen wir als Unternehmen darauf antworten? Wie verändern wir unser Angebot und wie verhält sich die Nachfrage? Was sind die zukünftigen Bedürfnisse unserer Kunden? Welche Anforderungen und Regularien sind davon betroffen und werden sich ändern? Wie bauen wir unseren Wettbewerbsvorteil aus? Auf all diese Fragen gilt es zukünftig die strategische Antwort parat zu haben und darauf vorbereitet zu sein. Hierfür benötigt jedes Unternehmen, egal in welcher Größe, eine **formale Planungsarchitektur**, um Effizienz und Klarheit zu schaffen und gemeinsam mit den jeweiligen Verantwortlichen der gesamten Organisationshierarchie die Zukunft zu steuern.

Integrated Business Planning (IBP) nimmt dabei eine Schlüsselfunktion ein, die auf eine effektive Entscheidungsfindung abzielt und vom Führungsteam geleitet wird. So ermöglicht IBP der Geschäftsleitung, das gesamte Unternehmen über einen Zeitraum von 36 Monaten zu planen und zu verwalten. Es werden jeden Monat strategische und taktische Pläne erneut abgestimmt und – wenn Änderungen auftreten – notwendige Entscheidungen getroffen sowie kritische Ressourcen wie Mitarbeiter, Inventar, Materialien, Zeit und Geld zugewiesen, um Kunden auf die rentabelste Weise auch in der Zukunft zufriedenzustellen. Der Prozess der Integrierten Taktischen Planung (ITP) hilft dabei, die Reaktion auf tägliche und wöchentliche Änderungen im Alltag zu optimieren, mit dem Ziel, dem IBP-Plan – soweit es geht – treu zu bleiben und daraus zu lernen, um zukünftig besser gewappnet zu sein. So operiert der ITP innerhalb des kurzfristigen Horizonts, im „Planungszeitfenster“, in dem sich der Schwerpunkt von der Planung auf die effektive Ausführung des vereinbarten Plans verlagert, da die Möglichkeiten, kosten- und ressourceneffizient zu antworten, jetzt begrenzt sind. Die Integrierte Taktische Planung ergänzt folglich die Integrierte Unternehmensplanung (IBP), die sich auf den mittel- bis langfristigen Planungshorizont konzentriert.

Transparenz durch einen monatlichen Planungsprozess

Durch die verschiedenen Organisationsstufen und Fokusbereiche kann demnach sichergestellt werden, dass Änderungen in

der Zukunft frühzeitig identifiziert und eine Antwort darauf schon proaktiv generiert werden kann. Das geht weit über den Prozess des klassischen Sales and Operations Planning (S & OP) hinaus und ist ein weit vorausschauendes Prozedere zur Umsetzung der Strategien unter Berücksichtigung aller Unternehmensbereiche. So kann das gesamte Führungsteam durch die monatlichen IBP-Zyklen gezielt auf Änderungen und Unsicherheiten im Markt antworten; das Team baut kontinuierlich, strukturiert und ganzheitlich sein gegenseitiges Verständnis der aktuellen Geschäftsentwicklung und der mittelfristigen Aussichten aus. Dies ermöglicht allen Funktionsverantwortlichen zusammenzuarbeiten, um frühzeitig ein Bewusstsein und eine Übereinkunft für die erwarteten Geschäftsszenarien zu schaffen. Im Rahmen des IBP-Prozesses können dann entsprechende Reaktionen des Unternehmens geplant werden. Die Ergebnisse dieser verschiedenen unternehmensweiten Planungsdiskussionen finden auf einem relevanten Aggregationslevel statt und werden in den Dimensionen „Volumen, Wert und Ressourcen“ gemeinsam abgewogen. Der Prozess liefert also globale Geschäftsprognosen – abgestimmt und realistisch. Der monatlich validierte und verbesserte IBP-Plan liefert die Basis für den Budgetzyklus. Das ist wichtig für eine dauerhafte Zuverlässigkeit und Konsistenz der Prognosen und vermeidet Überraschungen, die in einem isolierten jährlichen Planungsprozess sonst auftreten. Viele unserer Kunden haben durch IBP frühzeitig auf das vor einem Jahr vorhergesehene Risiko der heutigen Lieferengpässe und Transport-schwierigkeiten reagiert und bewusste Entscheidungen getroffen, wie beispielsweise alternative, lokale Lieferanten oder höhere Materialbestände aufgebaut. So konnten sie ihre Materialverfügbarkeit heute sicherstellen – ohne Produktionsausfälle oder höhere Transportkosten.

Vielen Kunden hat in dem Zuge die Nutzung der Szenarioplanung geholfen. Diese dynamische Herangehensweise an den Umgang mit Ungewissheit verschafft denjenigen, die sie effektiv anwenden, einen Wettbewerbsvorteil und ermöglicht, dass Planung und Entscheidungsfindung gedeihen und geschäftlicher Erfolg erzielt werden kann. Sobald eine Organisation das „wahrscheinlichste Ergebnis“ ermittelt hat, kann sie damit beginnen, die Bereiche zu identifizieren und zu verwalten, in denen

Ungewissheit besteht. Diese werden als „Chancen“ und „Unsicherheiten“ bezeichnet. Damit die Szenarioplanung effektiv funktioniert, ist eine robuste Fähigkeit erforderlich, um diese Vorteile (Chancen) und Nachteile (Unsicherheiten) zu erfassen, so dass die verschiedenen Szenarien verwaltet werden können. Selten ist ein Szenario eine eindimensionale Betrachtung, stattdessen ist es eine multifunktionale und facettenreiche Geschäftsreaktion auf Ungewissheit.

Lieferantenbewertung – Oracle Supplier Qualification Management

In Ergänzung zu den Planungsprozessen stehen Unternehmen vor der Frage, welche Technologien nun konkret bei der Einhaltung der Sorgfaltspflichten unterstützen können. Gerade die Anforderungen an die Auswahl der Lieferanten hat sich im Zuge des Lieferkettengesetzes grundlegend geändert, so fordert es von den Unternehmen weitaus mehr als nur das Zusammenführen der Zulieferer. Aufdecken von Risiken, Chancen und Unsicherheiten sowie die daraus abgeleiteten Maßnahmen mit einem umfassenden Berichtswesen sind notwendig, um gesetzeskonform zu agieren. Die Lösung liegt in einem Qualifizierungsmanagementsystem, das die Compliance von Lieferanten, die Datengenauigkeit und die Transparenz verbessert. Oracle Supplier Qualification Management Cloud bietet in diesem kontinuierlichen Qualifizierungsprozess der Lieferanten die notwendigen Funktionen. Die Komplettlösung umfasst das Management der Qualifikationen und Fähigkeiten, einschließlich des Monitorings zur Einhaltung der Unternehmensrichtlinien und der erforderlichen Dokumentation. Die Realisierung dieser komplexen Aufgaben erfolgt mittels einer intelligenten Abfrage. In einer Bibliothek stehen aufeinander abgestimmte Fragenkataloge zur Verfügung, die auf die spezifischen Bewertungsziele und -kriterien der Lieferanten abgestimmt sind. Bewertungen erfolgen auf Grundlage definierter Kriterien und werden automatisiert verarbeitet, so werden beispielsweise bei Ermittlung von Risiken entsprechende Benachrichtigungsprozesse angestoßen. Auch die Prüfung von Gültigkeitszeiträumen und Hinweise für Verbesserungen laufen im System automatisiert ab. Somit stehen immer die aktuellsten Informationen bezüglich der Lieferanten bereit inklusive des Qualifizierungsstatus. Dies sorgt für die

notwendige Transparenz, Minimierung der Risiken und Sicherheit hinsichtlich eines gesetzeskonformen Lieferkettenprozesses. Daraus ergeben sich folgende Vorteile:

- Alle Aktivitäten zur Qualifizierung werden digital erfasst und sind jederzeit einsehbar
- Nur qualifizierte Lieferanten werden aufgenommen
- Digitale Bewertung im Rahmen des Qualifizierungsprozesses
- Bestätigung der Lieferanten durch vertraglich verbindliche Zusagen

Use Cases in Oracle Fusion Cloud

Kontinuierliche Veränderung erfordert kontinuierliche Innovation. Die folgenden Szenarien verdeutlichen, wie die aktuelle Enterprise-Resource-Planning-Anwendung von Oracle dazu beiträgt, Risiken zu identifizieren und zu minimieren:

- **Verträge mit neuen Lieferanten einrichten oder bestehende erneuern**
In Oracle Fusion (Procurement) Contracts Cloud können Verträge in einem vollständig durchsuchbaren Repository verwaltet und geändert werden, etwa um neue Klauseln und Leistungen (wie die Bestätigung der Einhaltung von Menschenrechten) hinzuzufügen und die Leistungen zu verfolgen.
- **Lieferanten kontaktieren**
Mit der Oracle Fusion Supplier Qualification Management Cloud können Lieferanten kontaktiert werden, um sie online zu informieren und Informationen von ihnen abzurufen, um sie beispielsweise über anstehende Vertragsänderungen zu informieren oder um Beiträge zu bitten.
- **Definition von Regelwerken zur Lieferantenqualifizierung**
Oracle Fusion Supplier Qualification Management Cloud & Fusion Cloud Sourcing ermöglichen, auf Antworten/Fragen basierende Regeln zu erstellen, um Lieferanten zu klassifizieren und zu bestimmen, zum Beispiel um für ein bestimmtes Produkt den Lieferanten auszuwählen, der die Menschenrechte am besten einhält.
- **Integration von Zertifikatdatenbanken**
Oracle Integration Cloud ermöglicht, strukturierte Informationen über Artikel (zum Beispiel TÜV-Blacklist) oder Lieferanten aus öffentlichen Daten-

quellen abzurufen und mit Anwendungsstammdaten abzugleichen, um sie zu klassifizieren, beispielsweise Artikel auf Basis der GTIN (Global Trade Item Number) oder Lieferanten auf Basis der Umsatzsteuer-ID.

Wie erfülle ich die Anforderungen an die Dokumentationspflicht?

Was in den Softwaresystemen passiert, ist im Detail gut dokumentiert und nachvollziehbar. Allerdings stellt für Unternehmen häufig die Dokumentation ihrer Geschäftsprozesse eine Herausforderung dar. Mit dem Horus Business Modeler können die Anforderungen der Dokumentation und Beschreibung der Verfahren sehr gut durchgeführt werden. Die Prozesse werden in Form von Ablaufmodellen durch Aktivitäten und Objektflüsse abgebildet. So werden die Geschäftsprozesse in einzelne notwendige Schritte (Aktivitäten) unterteilt und auf unterschiedlichen

Hierarchiestufen verfeinert. Die für die Durchführung einer Aktivität notwendigen Voraussetzungen und aus ihrer Ausführung resultierenden Ergebnisse werden durch Objektspeicher dargestellt. Durch die Zuordnung von verschiedenen Modelltypen und Dokumenten können die Ablaufmodelle zu einem umfassenden Geschäftsprozessmodell erweitert werden. Mit dem Modellierer können alle Aktivitäten und die erwarteten Output-Objekte als Prozeduren dokumentiert werden. Zugehörige Rollen und Anwendungen können mit dem Vorgehensmodell verbunden werden, was zu einer angemessenen End-to-End-Dokumentation führt.

Ausblick – Globale Handelsnetzwerke managen

Die Lieferkettenanwendung von Oracle „Intelligent Track and Trace“ (OITT) hilft, die zahlreichen und immer komplexer werdenden Herausforderungen in einem globalen Netzwerk von Handelspartnern zu bewältigen. Man muss kein Entwickler sein oder über spezielles Expertenwissen verfügen, um OITT zu nutzen: Sie ist sofort funktionstüchtig und einsatzbereit. Als Teil der Oracle Blockchain Applications Cloud nutzt Oracle Intelligent Track and Trace die Blockchain-Technologie, um das Lieferkettenmanagement effizienter und sicherer zu gestalten. Anwender sehen auf einen Blick, wo sich Transaktionen oder Waren entlang der Lieferkette und bei den unterschied-

lichen Handelspartnern befinden. Auch Verknüpfungen mit Drittanbieteranwendungen sind möglich: Für Nutzer bedeutet das, die Oracle Intelligent Track and Trace auch ohne Oracle-Blockchain-Plattform verwenden zu können. OITT unterstützt Lebensmittel- und Getränkeindustrie, Transportwesen und andere Branchen dabei, Zertifizierungen sowie die Einhaltung von Produktions- und globalen Handelsvorschriften zu gewährleisten. Um ein Netzwerk von Handelspartnern effizient zu verwalten, bietet die Leistungsfähigkeit von Oracle Blockchain eine Technologie, die wiederum Teil eines verlässlichen Lieferkettennetzwerks ist.

Fazit

- Es besteht direkter Handlungsbedarf für den Mittelstand mit >1.000 Mitarbeitern, eine Grundsatzerklärung abzugeben und Verantwortlichkeiten zu definieren.
- Es gibt technische Lösungen für das Risikomanagement und eine vollumfängliche Dokumentation entlang der Wertschöpfungskette.
- Planungsprozesse sind wichtig, um frühzeitig Änderungen zu identifizieren und eine effiziente Antwort auf die zukünftigen Herausforderungen zu finden.
- Proaktivität kann kompetitive Vorteile schaffen.

Referenzen

- [1] Studie für das Bundesministerium für wirtschaftliche Zusammenarbeit und Entwicklung, Handelsblatt Research Institute, Sorgfaltspflichten entlang globaler Lieferketten, Juli 2021 [Global Slavery Index, Walk Free Foundation (2018)]
- [2] Nationaler Aktionsplan, Umsetzung der VN-Leitprinzipien für Wirtschaft und Menschenrechte, 2016 - 2020



Dr. Birgit Breitschuh
birgit.breitschuh@oliverwight.com

Birgit Breitschuh ist Managing Associate bei Oliver Wight EAME LLP und hat in ihren zwei Jahrzehnten Karriere bei den FMCG Großunternehmen Unilever, Kraft und Wrigley (Mars) echte Fachkenntnisse in den Bereichen Innovation und Lieferkettenmanagement erworben. Als Partnerin von Oliver Wight greift sie auf einen umfangreichen Hintergrund im Veränderungsmanagement zurück, um Unternehmen zum Erfolg zu coachen.



Jochen Rahm
jochen.rahm@promatis.de

Als Geschäftsführer (CTO) der PROMATIS software GmbH ist er auf strategischer und taktischer Ebene für die im operativen Geschäft der PROMATIS Gruppe relevanten Methoden und Technologien zuständig. Er verfügt über mehr als 20 Jahre Erfahrung im Programm- und Projektmanagement sowie in der Einführung betriebswirtschaftlicher Standard-Anwendungssoftware. Jochen Rahm steht für Kundenorientierung und effizientes Change Management, gerade auch im globalen Kontext. Als Executive Consultant unterstützt er strategische Kunden in der Umsetzung anspruchsvoller Digitalisierungsprogramme. Dabei stehen für ihn stets intelligente Lösungen für die operative Exzellenz der Unternehmen im Vordergrund.



Branchenweite Standards zum Schutz von Frauen in der textilen Lieferkette – wie eine kleine Fabrik als Inspiration diente

Von Bettina Ernst, Mitarbeiterin von CARE Deutschland

Seit vielen Jahren ist die Gleichstellung der Geschlechter in der Arbeitswelt ein wichtiges Thema. Dazu gehören auch gerechte und existenzsichernde Löhne sowie die Vermeidung von geschlechtsspezifischer Gewalt und von Belästigung am Arbeitsplatz. Die Hilfsorganisation CARE nimmt sich dieser Themen in der textilen Lieferkette an und führt unter anderem präventive Schulungen und Trainings in Textilfabriken durch.

CARE startete Sensibilisierungsschulungen von Textilarbeiterinnen und entwickelte gemeinsam mit Partnern standardisierte Richtlinien für alle Ebenen der textilen Lieferkette, also von der Bekämpfung der geschlechtsspezifischen Gewalt bis zur Belästigung am Arbeitsplatz (der Fabriken).

In Kambodscha begann CARE 2014 ein Projekt mit einer kleinen Anzahl von Tex-

tilarbeiterinnen in einer Fabrik. Die Schulungen integrierten nach und nach die ersten Botschaften zur Prävention von sexueller Belästigung am Arbeitsplatz.

Dabei erkannte das Team schnell, dass Schulungen allein nicht ausreichen würden, damit sich Frauen sicherer fühlten. Selbst mit mehr Selbstvertrauen und Verständnis für die

Problematik, wurden Meldungen über Gewalt und Belästigung nicht wirksam behandelt. Fehlende Arbeitsplatznormen, unzureichende Richtlinien und ein mangelndes Engagement der Führungsebene waren die Knackpunkte, obwohl aktuelle Untersuchungen [1] klar zeigten, dass diese Probleme ernstgenommen werden sollten.

Klare Richtlinien decken auf

Chakrya ist Compliance-Beauftragte in einer Fabrik in Kambodscha. Sie arbeitet seit drei Jahren mit CARE zusammen, um die Richtlinien für sexuelle Belästigung zu verbessern und sicherzustellen, dass Fälle von Belästigung ordnungsgemäß adressiert werden.

„In der Fabrik kam es zu sexueller Belästigung zwischen Arbeiterinnen und Mechanikern. Die Mechaniker belästigten die Arbeiterinnen, wenn sie die Maschinen reparierten. Die Arbeitnehmerinnen, die sexuell belästigt wurden, wussten nicht, an wen sie sich wenden konnten, und dachten, es handele sich nur um eine persönliche Angelegenheit. Sie hatten Angst, ihren Arbeitsplatz zu verlieren, wenn sie es melden würden“, berichtet Chakrya.

Trotz des wachsenden Bewusstseins über die positiven Auswirkungen, die es hat, wenn sich Frauen am Arbeitsplatz sicher fühlen, war das Management oft unsicher, wie man am besten mit Vorfällen von Belästigung am Arbeitsplatz umgehen sollte. Infolgedessen zog Chakrya es nicht in Betracht, die Fälle offiziell zu melden. Sie befürchtete, dass sie dadurch den Ruf der Fabrik schädigen könnte und es Einfluss auf die Käufer und auf die Auftragslage hätte.

„Ich habe Fälle von sexueller Belästigung in der Fabrik vor den Auditoren verheimlicht, weil ich mir nicht sicher war, ob der Prozess, den wir zur Lösung der Fälle durchführten, effektiv war oder nicht. Außerdem hatten wir Angst, dass dies bei der Prüfung aufgezeichnet und wir beschuldigt werden würden. Wir hätten unseren Ruf verlieren können.“

Better Factories Cambodia führt regelmäßig Bewertungen von mehr als 550 Fabriken durch. Es hat die klare Botschaft vermittelt, dass eine transparente Nachverfolgung von Beschwerden und deren Lösung ein weitaus besserer Weg ist, um die Einhaltung der Vorschriften zur Verhinderung von sexueller Belästigung nachzuweisen, als Vorfälle einfach zu vertuschen.

Chakryas Fabrik hat mit CARE zusammengearbeitet, um eine klare Richtlinie zu verabschieden, Meldesysteme für Vorfälle zu entwickeln und sicherzustellen, dass alle Beschäftigten verstehen, dass sexuelle Belästigung nicht toleriert wird. Sie ist Mitglied des Komitees zur Prävention von sexueller Belästigung in der Fabrik. Wenn also ein Fall gemeldet wird, nutzt sie jetzt den Beschwerdemechanismus und befolgt die festgelegten Schritte für die Aufzeichnung des Vorfalls.

„Ich weiß jetzt, wie man effektiv mit Beschwerden umgeht. Ich bin zuversichtlich, dass ich die Fälle bei der externen Prüfung, zum Beispiel der IAOIL-Prüfung, offenlegen kann. Außerdem ist der jährliche Fallverfolgebbericht sehr hilfreich für mich, um die Fälle vertraulich aufzuspüren.“

„Ebenso bin ich jetzt zuversichtlich, dass ich sexuelle Belästigung in meinem Betrieb verhindern und darauf reagieren kann. Meine Fabrik kann die Arbeiterinnen schützen und die kambodschanischen Gesetze einhalten.“

Als CARE erkannte, dass ein Umdenken stattfinden muss, konzentrierten sich die Mitarbeitenden nicht mehr nur noch auf Verhaltensänderungen durch Sensibilisierungsschulungen und eine verbesserte Aufklärung der Textilarbeiterinnen, sondern auch auf die internen Strukturen und Verfahren in den Fabriken. Schließlich spielen diese eine wesentliche Rolle bei der Förderung von belästigungsfreien Arbeitsplätzen. Dieses Umdenken führte schließlich zur Entwicklung des STOP-Modells [2].

Das Ergebnis war ein Unterstützungspaket für das Fabrikmanagement, das sich auf die Stärkung ihrer Systeme und Prozesse konzentrierte und Instrumente wie Muster-

richtlinien und -formulare, Berichtsmechanismen sowie Anleitungen zur Einrichtung von Ausschüssen zur Verhinderung sexueller Belästigung enthielt. In Kambodscha arbeitete CARE dazu mit den Personalleitungen der Fabriken zusammen, um ihre internen Verfahren und Berichte für die Aufklärung über sexuelle Belästigung zu verbessern und das Bewusstsein zu schärfen, dass sexuelle Belästigung inakzeptabel ist.

Nach dem Erfolg der Initiative in Kambodscha replizierte und verfeinerte CARE das STOP-Modell in Myanmar, Laos und Vietnam und ermutigte andere Marken und Zulieferer in Kambodscha, es zu übernehmen. Auch in Bangladesch wurden Fabriken da-

bei unterstützt, Anti-Belästigungs-Ausschüsse einzurichten, um sicherzustellen, dass diese effektiv arbeiten und Textilarbeiterinnen die Möglichkeit haben, Führungspositionen einzunehmen. Die Unterstützung wird nun auf Fabriken in Pakistan und Äthiopien ausgeweitet.

Während der Entwicklung des STOP-Modells überprüfte CARE die gewonnenen Erkenntnisse in einer evidenzbasierten Studie, die aufzeigt, was zur Verhinderung von Belästigung am Arbeitsplatz wirklich funktioniert [3]. CARE bindet Führungskräfte des mittleren und höheren Managements mit ein und arbeitet sowohl mit den Geschäftsführungen als auch mit Personalabteilun-



Chakrya, Compliance-Beauftragte einer Fabrik in Kambodscha (Quelle: CARE Cambodia)

Standardisierte Verfahren helfen

Die Fabrik von Kim Bou war eine der ersten Fabriken, die mit CARE kooperierten, um standardisierte Instrumente für den Umgang mit sexueller Belästigung am Arbeitsplatz zu entwickeln. Die Fabrik wurde durch das Projekt ‚Enhancing Women’s Voice to STOP Sexual Harassment (STOP)‘ unterstützt. Das Projekt wurde von CARE und Better Work in Zusammenarbeit mit Partnern aus der Textilindustrie und der Unterstützung der australischen Regierung umgesetzt.

Herr Kim Bou ist seit vielen Jahren Personalleiter in einer Fabrik in Kambodscha. Er beaufsichtigt mehr als 1.000 Arbeiterinnen und Arbeiter. Gemeinsam mit CARE entwickelte er standardisierte Beschwerdemechanismen für seine Fabrik. In dieser Zeit hat er festgestellt, dass klare Richtlinien und die Einbindung der Beschäftigten in dieses Thema viele Vorteile mit sich bringen und gründete zur Bekämpfung von sexueller Belästigung am Arbeitsplatz einen Ausschuss mit verschiedenen Beschäftigten.

In den letzten fünf Jahren habe ich festgestellt, dass die Zeit, die wir uns für die Schulung der Beschäftigten nehmen, keine negativen Auswirkungen auf unsere Produktivität hat. Wir sparen sogar Zeit, weil sexuelle Belästigung nicht mehr so häufig vorkommt, so dass wir nicht mehr so viel Zeit aufwenden müssen, um viele Fälle zu lösen. Es sorgt für einen reibungslosen Produktionsprozess und hilft dem Unternehmen, einen guten Ruf zu erlangen.

Kim Bou fand es früher schwierig, mit diesen Problemen umzugehen. Die standardisierten Verfahren haben seine Arbeit erleichtert. Neben Kampagnen- und Schulungsmaterial waren besonders die Richtlinien am nützlichsten: Verfahren zur Lösung eines Falls, Techniken zur sensiblen Befragung mit Fingerspitzengefühl von Arbeiterinnen – einfach klare Richtlinien für die Untersuchung von Fällen.

„Wir haben jetzt vielleicht zwei oder drei Vorfälle von Belästigung pro Jahr, meist verbale Belästigung, die die Frauen dem Ausschuss melden. Wir haben klare Mechanismen, um diese Fälle zu lösen“, berichtet Herr Bou.

Die geregelten Verfahren in der Fabrik sind sowohl für Audits als auch für die Bewertungen der Fabrik wichtig, sie überzeugen Marken, gewinnen neue Käufer und können diese binden.

Kim Bou ist der festen Überzeugung, dass die Anwendung dieser Leitlinien eine Wirkung hat.



CARE hat fast 20 Jahre Erfahrung in der Arbeit mit der Bekleidungsindustrie in Kambodscha (Quelle: CARE Cambodia)

gen der Fabriken zusammen, um einen Wandel der Normen am Arbeitsplatz zu unterstützen. Darüber hinaus verstärken wir in Zusammenarbeit mit Branchenverbänden und durch die Teilnahme an globalen Konferenzen und Netzwerken das Bewusstsein von Markenunternehmen und weisen auf die Auswirkungen von Gewalt und Belästigung am Arbeitsplatz hin, die sowohl die Textilarbeiterinnen als auch die Fabriken selbst betreffen.

Von der Arbeitsplatzpolitik zur Branchenpraxis

Die Ergebnisse verdeutlichen [4], dass klare Verfahren, Engagement der obersten Führungsebene und eine organisatorische Verankerung des Themas Wirkung zeigen. In Laos ging der Anteil der Arbeitnehmerinnen, die von sexueller Belästigung berichteten, von 17 % auf 5 % zurück. In Kambodscha sank der Anteil derjenigen, die berichteten, einen Arbeiter bei einem sexuellen Übergriff in der Fabrik beobachtet zu haben, von 17 % auf 1 %. Fest steht: Klare Botschaften und eine starke Führung in dieser Fra-

ge haben positive Auswirkungen. Der Anteil der Manager in Myanmar, die sexuelle Belästigung nicht ernst nahmen, halbierte sich, als ihre Fabriken mit dem CARE-Programm begannen.

Dieser Ansatz erwies sich in Fabriken in zahlreichen Ländern als sehr wirksam. Nun stellt sich die Frage, wie der Ansatz weiter ausgebaut werden kann. Erfolgreiche Ergebnisse auf Fabrikebene, sogar die Ausweitung über verschiedene Länder, sind ermutigend, aber führen dennoch nur zu Veränderungen bei einem Teil der Textilarbeiterinnen. Was ist mit den Millionen von Frauen, die weltweit in der Textilproduktion beschäftigt sind?

CARE kooperiert daher mit der Initiative Better Work [5], um das Problem von der Fabrikebene bis hinauf durch die gesamte Lieferkette zu skalieren. Gemeinsam nehmen wir Einfluss darauf, wie die Textilindustrie mit diesem Thema ganzheitlich umgeht. Dazu gehört auch zu verstehen, welche Auswirkungen Einkaufspraktiken der Marken auf die Fabrikumgebung haben, so dass keine Bedingungen geschaffen werden, die Belästigungen jeglicher Art ermöglichen.

Daher organisierten wir gemeinsam die Veranstaltung „Business of Women at Work“ [6], die verschiedene Perspektiven aus der gesamten Lieferkette zusammenbrachte, um praktische Lösungen für geschlechtsspezifische Gewalt und Belästigung am Arbeitsplatz zu finden. Anschließend arbeiteten wir mit einer Reihe von Marken und Branchenvertretern zusammen, um auf der Grundlage internationaler Arbeitsnormen und bewährten Praktiken, standardisierte Leitlinien für Unternehmen zu entwickeln, die sich mit diesem Thema in ihren eigenen Betrieben und in ihren Lieferketten befassen.

Angeleichung der Industrie an gemeinsame Standards

Die neuen Richtlinien für den Umgang mit geschlechtsspezifischer Gewalt und Belästigung in der Textil-, Bekleidungs- und Schuhindustrie [7] enthalten Verfahren für verschiedene Teile der Lieferkette, mit Anleitungen für Bereiche wie Kundenmanagement und Beschaffung, Einkauf und Beschaffung sowie für die Personalabteilung. CARE und Better Fac-



Ein Bild aus dem Set, das verwendet wird, um sexuelle Belästigung in Bekleidungsfabriken zu verhindern (Quelle: CARE Cambodia)

ories Cambodia ermutigen alle, dieses kostenlose Open-Source-Tool zu nutzen, um bestehende Richtlinien, Systeme oder Prozesse zu überprüfen, zu vergleichen und Bereiche zu identifizieren, in denen Maßnahmen ergriffen werden müssen.

Die Leitlinien wurden in Anlehnung an die neue globale Norm zu geschlechtsspezifischer Gewalt in der Arbeitswelt, das ILO-Übereinkommen C190, entwickelt. Da immer mehr Länder diese Konvention ratifizieren, besteht ein wachsender Bedarf an einer einheitlichen Reaktion in der gesamten textilen Lieferkette. Diese neuen Leitlinien bieten der gesamten Branche die Möglichkeit, sich an gemeinsamen Standards zu orientieren und Maßnahmen zu ergreifen, um sicherzustellen, dass das Potenzial für sichere Arbeitsplätze, das durch die Initiativen der einzelnen Fabriken aufgezeigt wurde, für die Millionen von Textilarbeiterinnen genutzt wird.

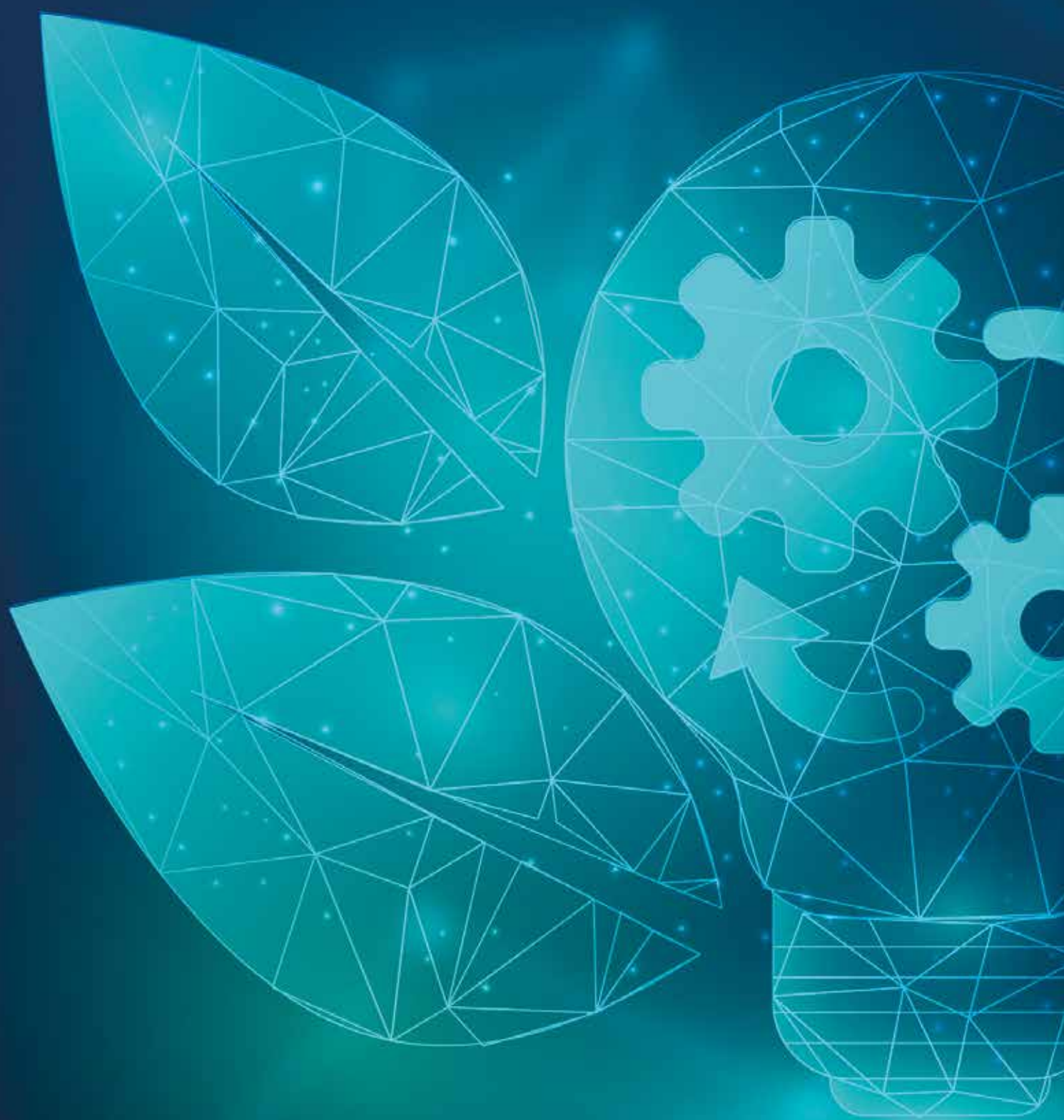
Quellen

- [1] https://www.care.org.au/wp-content/uploads/2019/11/SHCS_Research_Summary_March_2017.pdf
- [2] <https://www.care.de/helfen/als-unternehmen-helfen/textile-lieferkette-verbessern/stops-sexual-harassment-prevention-package/>
- [3] https://www.care.org.au/wp-content/uploads/2018/12/STOP_What-works-to-address-workplace-sexual-harassment_Rapid-Review-Summary.pdf
- [4] https://www.care.org.au/wp-content/uploads/2021/02/STOP_regional-reflection-report_Feb-2021.pdf
- [5] <https://betterwork.org/where-we-work/cambodia/>
- [6] <https://insights.careinternational.org.uk/development-blog/sexual-harassment-in-the-garment-industry-what-happens-when-you-bring-people-together-to-look-at-solutions-rather-than-problems>
- [7] <https://www.care.org/our-work/education-and-work/dignified-work/preventing-violence-and-harassment-at-work/standardized-guidance-for-addressing-harassment-in-the-garment-industry/>



Bettina Ernst
ernst@care.de

Bettina Ernst ist seit 2013 bei CARE Deutschland e.V. tätig. Sie ist Corporate Partnership Manager im Bereich Marketing und Partnerschaften. Da immer mehr Unternehmen sich sozial engagieren und Gutes tun möchten, bietet CARE diese Möglichkeit an. Denn gemeinsam mit Unternehmen, die helfen möchten, können wir noch mehr bewirken. Wir beraten Sie gerne zu den verschiedenen Kooperationsmöglichkeiten. Ob eine regelmäßige Firmenspende, mit der Sie uns ermöglichen, bei Krisen und Katastrophen schnell und flexibel zu helfen, einer Cause-Related-Marketing-Kampagne, einer Aktion mit den Mitarbeitenden oder durch eigene Projekte. Gemeinsam entwickeln wir ein passendes Projekt für Ihr Unternehmen. Durch unseren Schwerpunkt, die Gleichstellung der Geschlechter, geben wir Frauen und Mädchen eine Stimme und schaffen so weitreichend einen sozialen Wandel.



Von Nachhaltigkeit und deren Nachweis in Lieferketten

Markus Adler, Spenoki Solutions GmbH, und Florian Lösch, Horus software GmbH, Ettlingen (TechnologieRegion Karlsruhe)

Nachhaltigkeit rückt mehr und mehr in den Fokus der Öffentlichkeit. Dies lässt sich unter anderem in gesättigten und deshalb stark umkämpften Kundenmärkten erkennen. In diesen entschließen sich Kunden zuletzt immer häufiger für den Kauf des nachhaltigen Produktes [1]. Hierdurch wurde das Prädikat „nachhaltig“ zu einem Alleinstellungsmerkmal, aber auch zu einer Werbebotschaft, um sich von Konkurrenzprodukten abzuheben. Dabei entscheidet jedoch leider zu häufig die bunte Werbung oder die eigens intern etablierte Zertifizierungsstelle über nachhaltig oder nicht. Die tatsächliche Nachhaltigkeit eines Produktes steckt jedoch in den genutzten Grunderzeugnissen, im jeweils genutzten Produktionsverfahren und in der Lieferkette. Aber selbst bei einer intensiven Betrachtung dieser ist es oftmals schwer zu bewerten, was nachhaltig ist. Deshalb beschäftigt sich dieser Artikel damit, den Begriff der Nachhaltigkeit beziehungsweise des nachhaltigen Handelns zu veranschaulichen, und zeigt beispielhaft Wege auf, mit denen Nachhaltigkeit in Lieferketten erkannt, gemessen und diese damit bewertbar gemacht werden kann.

Nachhaltigkeit und nachhaltiges Handeln

Nachhaltigkeit wird häufig als komplex und von verschiedenen Faktoren abhängig dargestellt. Das zeigt sich bereits in der allgemeinen Definition, die Nachhaltigkeit als eine Kombination aus ökonomischen, ökologischen und sozialen Aspekten darstellt. Diese sind häufig konträr zueinander ausgerichtet, weshalb sie nur schwer miteinander in Einklang zu bringen sind. Dies verdeutlicht die Beschreibung des Tripple-Bottom-Ansatzes nach Elkington [2], der ein nachhaltiges Unternehmen als eines beschreibt, das ökonomischen Wohlstand, ökologische Qualität und soziale Gerechtigkeit berücksichtigt oder fördert. Elkington zufolge kann eine nachhaltige Entwicklung nur erfolgen, wenn zwei der drei zuvor beschriebenen Dimensionen optimiert werden, während sich die dritte zumindest nicht verschlechtert [3].

Eine Verfeinerung zu diesem dreidimensionalen Ansatz bilden die Sustainable Development Goals (SDG) der Vereinten Nationen. Ihr Fokus liegt auf den beiden Dimensionen Sozial und Ökologie, da sie aus deren Sicht die Grundlage für langfristige ökonomische Entwicklung darstellen. Dieser Ansatz, in dessen Zentrum die Themen Menschheit, der Planet, Wohlstand, Frieden und Zusammenarbeit stehen, setzen für eine nachhaltige Entwicklung die Erfüllung von 17 Haupt- und 169 Unterzielen voraus. Ihre Umset-

zung wurden als Erweiterung vorheriger Nachhaltigkeitsziele wie beispielsweise der Millennium Development Goals während des UN Sustainable Development Summit in New York im September 2015 diskutiert und verabschiedet. Die in ihr enthaltenen Ziele sollen bis 2030 weltweit umgesetzt sein [4]. Sie bilden somit die Leitplanken, an denen sich Staaten, Gesellschaften, Individuen und auch Unternehmen orientieren sollen, um nachhaltiges Zusammenleben auch für zukünftige Generationen zu ermöglichen. Deshalb können sie als Rahmen oder Best Practice für nachhaltiges Handeln angesehen werden.

In ihrer Gesamtheit lassen sie sich und damit nachhaltiges Handeln in vereinfachter Form wie folgt gliedern:

- SDG 1 Bekämpfung von Armut in all ihren Formen
- SDG 2 Beendigung des Hungers und Verbesserung der Ernährung
- SDG 3 Sicherstellung eines gesunden Lebens und Förderung des Wohlbefindens
- SDG 4 Gewährleistung gerechter Qualitätsbildung für Menschen jeden Alters
- SDG 5 Verwirklichung der Gleichstellung der Geschlechter
- SDG 6 Gewährleistung der Verfügbarkeit und nachhaltigen Nutzung von Wasser
- SDG 7 Sicherstellung des Zugangs zu erschwinglicher und nachhaltiger Energie
- SDG 8 Förderung von nachhaltigem

- Wirtschaftswachstum, Vollbeschäftigung und menschenwürdiger Arbeit
- SDG 9 Förderung einer integrativen und nachhaltigen Industrialisierung und von Innovationen
- SDG 10 Verringerung der Ungleichheit innerhalb und zwischen den Ländern
- SDG 11 Städte und menschliche Siedlungen integrativ, sicher und widerstandsfähig machen
- SDG 12 Gewährleistung nachhaltiger Konsum- und Produktionsmuster
- SDG 13 Umgehend Maßnahmen zur Bekämpfung des Klimawandels und seiner Auswirkungen ergreifen
- SDG 14 Erhaltung und nachhaltige Nutzung der Ozeane, Meere und Meeresressourcen
- SDG 15 Schutz der Biodiversität
- SDG 16 Förderung friedlicher und integrativer Gesellschaften und Zugang zur Justiz für alle
- SDG 17 Stärkung der Mittel zur Umsetzung und Neubelebung der globalen Partnerschaft für nachhaltige Entwicklung

Innerhalb dieser Haupt- und ihrer entsprechenden Unterziele gibt es verschiedene Themen, die Unternehmen direkt beeinflussen können. Hierzu zählen unter anderem:

- die Zahlung fairer Löhne und die Sicher-

stellung ihrer Zahlung bei seinen Lieferanten

- die Vermeidung von klimaschädlichen Emissionen in Produktions-, aber auch Transportprozessen
- die Gewährleistung menschenwürdiger Arbeitsbedingungen und der aktive Kampf gegen Diskriminierung und Korruption im Arbeitsumfeld
- die Optimierung von Produktionsprozessen zur Reduzierung von Verbräuchen und zur Bewahrung von kritischen Ressourcen wie beispielsweise Wasser
- das aktive Steuern von Konsumverhalten hin zu einem bedachten und nachhaltigen Kaufverhalten
- die Etablierung einer zu Ende gedachten Kreislaufwirtschaft, die zum Beispiel auf Recycling setzt

Bei näherer Betrachtung der einzelnen Themengebiete muss aber auch festgestellt werden, dass nicht alle Bereiche direkt durch Unternehmen und deren Lieferketten beeinflusst werden können. Vielmehr ist an einigen Stellen der Eingriff von staatlichen Institutionen nötig. Dies gilt beispielsweise beim Setzen von Sozialstandards, bei der Erleichterung von Migration sowie der Mobilität von Menschen, in der verstärkten Bildung aller Menschen oder auch dem Aufbau verlässlicher rechtsstaatlicher Systeme. Um dieser Verantwortung nachzukommen, verfolgen verschiedene Staaten und Staatenverbände diese Ziele intensiv und versuchen darüber hinaus, privatwirtschaftliche Unternehmen durch Gesetze an die Ziele der Social Development Goals und anderer Nachhaltigkeitsrichtlinien zu binden.

Europäische Gesetzesinitiative zur Nachhaltigkeit in Lieferketten

So veröffentlichte beispielsweise die Europäische Union am 23. Februar 2022 eine Richtlinie, die unter anderem die Ziele der Social Development Goals aufgreift und sie durch die Regierungen der EU-Mitgliedsstaaten in bindende Gesetzgebung in Bezug auf die Nachhaltigkeit in Lieferketten umwandeln lässt. Hierdurch sollen EU-weit einheitliche Wettbewerbsfaktoren und eine durchgängige Rechtssicherheit geschaffen werden. Dies ist vor allem vor dem Hintergrund diverser nationaler Gesetzgebungsinitiativen nötig. So gibt es Stand heute bereits in Frankreich und Deutschland bestehende Gesetze, die Unternehmen verpflichten, ihre Lieferketten auf spezifische

Aspekte im Kontext der Nachhaltigkeit hin zu analysieren und zu optimieren. In anderen Ländern wie beispielsweise Österreich, Dänemark, Luxemburg, Schweden oder Irland sind noch keine Gesetze erlassen. Jedoch gibt es in diesen Ländern bereits intensive Diskussionen hinsichtlich einheitlicher Regelungen zu nachhaltigen Lieferketten.

Darüber hinaus verfolgt die EU-Direktive das Ziel, Unternehmen dazu zu verpflichten, ihre Lieferkette bezüglich sozialer und ökologischer Missstände kontinuierlich zu überprüfen, diese Missstände aufzudecken, ihre Auswirkungen zu ermitteln und Maßnahmen zu ergreifen, um diese zu vermindern oder idealerweise zu eliminieren. Diese Maßnahmen sind in die Unternehmenspolitik zu verankern und transparent zu kommunizieren. Die Pflichten sollen im ersten Schritt große, in der EU registrierte Unternehmen mit einem weltweiten Mindestumsatz von 150 Mio. Umsatz und mindestens 500 Mitarbeitern beziehungsweise außerhalb der EU registrierte Unternehmen betreffen, die den oben genannten Umsatz in der Union erzielen [5].

Diese Verpflichtung von Unternehmen, die innerhalb und außerhalb der EU ansässig sind, ist vor allem vor dem Hintergrund dessen, dass etwa 80 - 90 % des ökologischen Schadens innerhalb von Lieferketten außerhalb der EU entstehen, von entscheidender Bedeutung [6].

Die gesetzlichen Vorgaben für Großunternehmen sollen innerhalb von zwei Jahren nach Billigung der Richtlinie durch das EU-Parlament und den Europarat in nationale Gesetzgebung umgesetzt sein. In einem zweiten Schritt – der weitere zwei Jahre später in Kraft treten soll – werden Unternehmen, die in der gemäß der OECD Sector Guideline definierten ressourcenintensiven Branche arbeiten, mehr als 40 Mio. Umsatz weltweit (für EU-Unternehmen) beziehungsweise europaweit (für Nicht-EU-Unternehmen) erwirtschaften und mindestens 250 Mitarbeiter beschäftigten, verpflichtet, die Anforderungen und Pflichten zu erfüllen. Klein- und mittelständische Unternehmen sind nicht direkt von der Gesetzgebung betroffen. Sofern sie jedoch in die Lieferketten von größeren, der Gesetzgebung unterliegenden Unternehmen involviert sind, müssen sie indirekt an der Einhaltung der Vorschriften mitarbeiten oder sich aus diesen Lieferketten zurückziehen.

Hierbei sind die Großunternehmen und der Gesetzgeber dazu angehalten, kleine-

re und mittelständische Unternehmen aktiv zu unterstützen. Denn nur wenn möglichst viele Unternehmen sich aktiv an der Entwicklung von nachhaltigen Lieferketten beteiligen, ist es möglich, mittelfristig eine ausreichende Datengrundlage zu schaffen, mit der nachteilige Einflüsse innerhalb von Lieferketten in Bezug auf Menschenrechte und Umwelteinflüsse auch mit geringem Aufwand identifiziert und eliminiert werden können.

Um dazu beizutragen, diese Daten zu erlangen, unterstützt Spenoki Unternehmen darin, ihre Lieferketten gemäß dem aktuell in Deutschland geltenden Lieferkettensorgfaltspflichtengesetz zu überprüfen.

Dabei liegt der Fokus darauf, die Bedeutung von Umwelt- und Sozialbelangen in der Lieferkette eines Unternehmens einschätzen zu können. Dies ist erforderlich, um ein realistisches Bild der Lieferkette des betreffenden Unternehmens zu erstellen. Die folgenden Abschnitte umreißen deshalb in aller Kürze das deutsche Lieferkettensorgfaltspflichtengesetz und stellen eine Schritt-für-Schritt-Anleitung dar, mit der die Anforderungen des Lieferkettensorgfaltspflichtengesetzes erfüllt werden können.

Eckpunkte des deutschen Lieferkettensorgfaltspflichtengesetzes [7]

Das Gesetz verlangt von Unternehmen die Umsetzung von neun Maßnahmen, um ihrer Sorgfaltspflicht nachzukommen. Die erste dieser Maßnahmen stellt die Einrichtung eines Risikomanagements dar. Hierbei haben Unternehmen einen Vorteil, die schon vor der neuen Gesetzgebung einen „Beyond Compliance“-Ansatz verfolgten und Nachhaltigkeit als ganzheitlichen Unternehmenswert strategisch in ihre Unternehmensplanung mit aufgenommen haben. Sie erfüllen bereits die Anforderung, das interne Qualitätsmanagement-System, das eine Analyse und Bewertung der unternehmerischen Risiken beinhaltet, um die Komponente der Nachhaltigkeit zu erweitern. Zusätzlich wird an dieser Stelle auch die zweite Anforderung des Gesetzes – die nach der eindeutigen Festlegung einer betrieblichen Zuständigkeit – abgedeckt. Die nächste genannte Pflicht verlangt die Durchführung einer kontinuierlichen Risikoanalyse. In diesem Zusammenhang erlaubt das Gesetz, den Fokus im ersten Schritt auf die eigenen Beschaffungsprozesse und die der unmittelbaren Zulieferer zu legen. Die

Art der Analyse wird hierbei relativ frei gefasst. Es wird jedoch verlangt, die Risiken zu bewerten und, sofern nicht alle umgehend adressiert werden können, diese zu priorisieren. Maßgeblich für die Priorisierung ist beispielsweise die Einflussmöglichkeit eines Unternehmens, das Risiko selbst oder dessen Eintrittswahrscheinlichkeit zu minimieren sowie das Ausmaß bei dessen Eintreten zusammen zu managen beziehungsweise zu reduzieren. Die Risikoanalyse ist in regelmäßigen Abständen, jedoch mindestens einmal jährlich, durchzuführen. Bei gegebenem Anlass ist von diesem Rahmen abzuweichen. Dies kann durch die Aufnahme einer neuen Tätigkeit oder Beziehung, vor strategischen Entscheidungen oder Veränderungen in der Geschäftstätigkeit erfolgen. Die Ergebnisse der Risikoanalyse sind nach dieser umgehend an alle maßgeblichen Entscheidungsträger zu kommunizieren und müssen von diesen bei ihren Entscheidungen berücksichtigt werden.

Um sicherzustellen, dass dies stets gemäß den Vorgaben durchgeführt wird, ist eine Grundsatzerklärung zu verfassen. Sie muss von der Geschäftsleitung verabschiedet werden und mindestens eine Verfah-

rensbeschreibung inklusive Maßnahmen, eine Priorisierung von Risiken und eine klare Erwartungshaltung an alle Stakeholder enthalten. Die Grundsatzerklärung ist jedoch nicht nur ein schriftliches Dokument, in dem das Unternehmen sein Commitment und Engagement für menschenrechtliche Sorgfaltspflicht zum Ausdruck bringt, sondern umfasst alle Sorgfaltspflichtmaßnahmen und kann als Präventionsmaßnahme bezeichnet werden. Basierend auf den Ergebnissen der Risikoanalyse müssen die Menschenrechtsstrategie des Unternehmens, die damit verbundenen Präventionsmaßnahmen sowie Erwartungen an Beschäftigte, Geschäftspartner und Lieferanten dargelegt werden.

Die Verankerung von Präventionsmaßnahmen im eigenen Geschäftsbereich und gegenüber unmittelbaren Zulieferern stellt die nächste Maßnahme dar. Zu dieser internen Maßnahme zählen neben den auf der Risikoanalyse basierenden präventiven Aufgaben die Schulung und Weiterbildung von Mitarbeitern, die Entwicklung und Implementierung geeigneter Beschaffungsstrategien und Einkaufspraktiken, durch die festgestellte Risiken vermieden oder

gemindert werden, und die Durchführung risikobasierter Kontrollmaßnahmen, mit denen die Einhaltung der in der Grundsatzerklärung enthaltenen Menschenrechtsstrategie im eigenen Geschäftsbereich überprüft wird. Präventionsmaßnahmen gegenüber Lieferanten können wie folgt gestaltet sein:

- Berücksichtigung der menschenrechts- und umweltbezogenen Erwartungen bei der Auswahl eines unmittelbaren Zulieferers
- vertragliche Zusicherung eines unmittelbaren Zulieferers, dass dieser die von der Geschäftsleitung des Unternehmens verlangten menschenrechtsbezogenen und umweltbezogenen Vorgaben einhält und entlang der Lieferkette angemessen adressiert
- Vereinbarung angemessener vertraglicher Kontrollmechanismen sowie die Durchführung von Schulungen und Weiterbildung
- Durchführung risikobasierter Kontrollmaßnahmen auf Grundlage der vereinbarten Kontrollmechanismen zur Überprüfung der Einhaltung

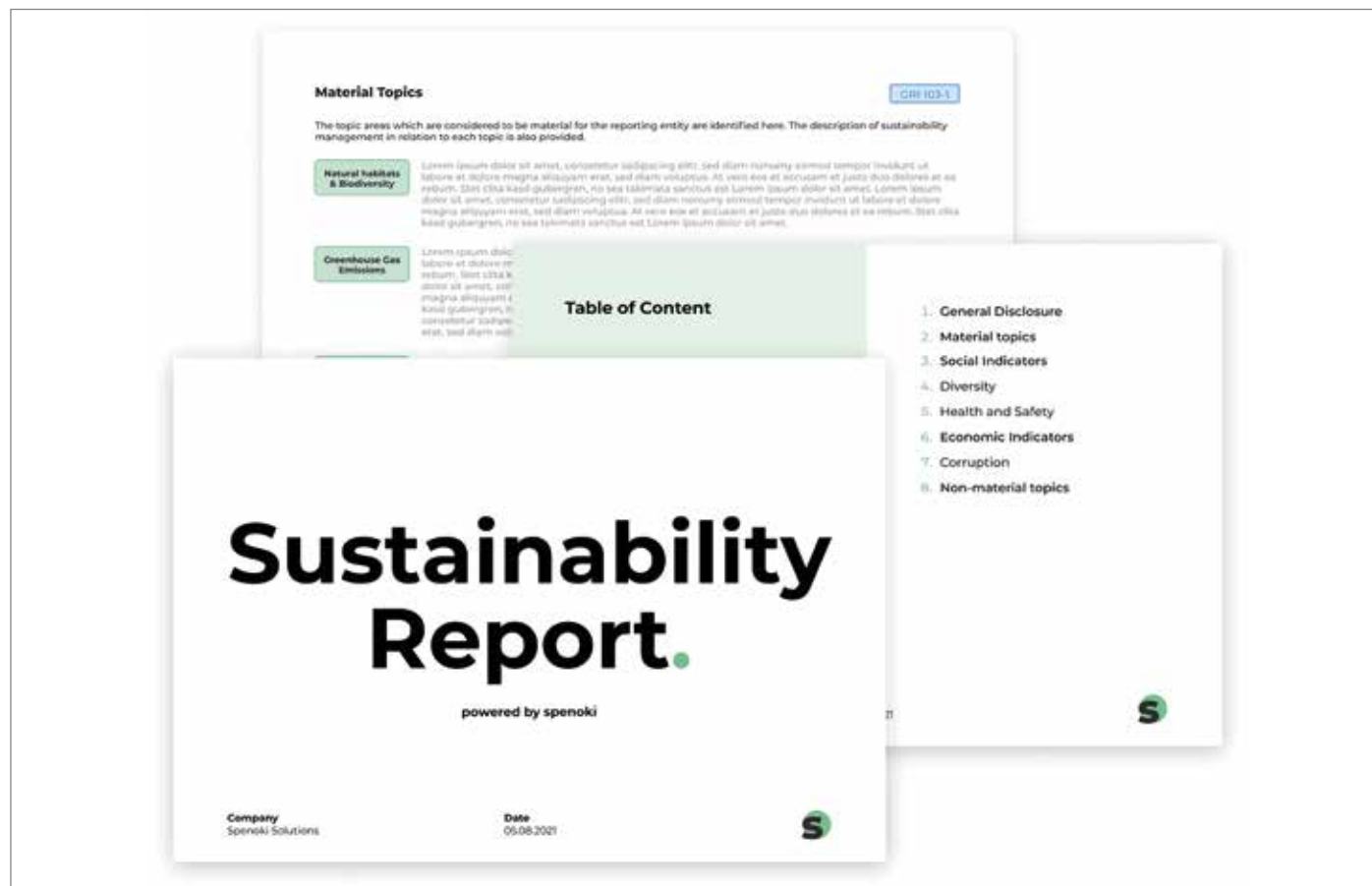


Abbildung 1: Report (© Spenoki Solutions GmbH)

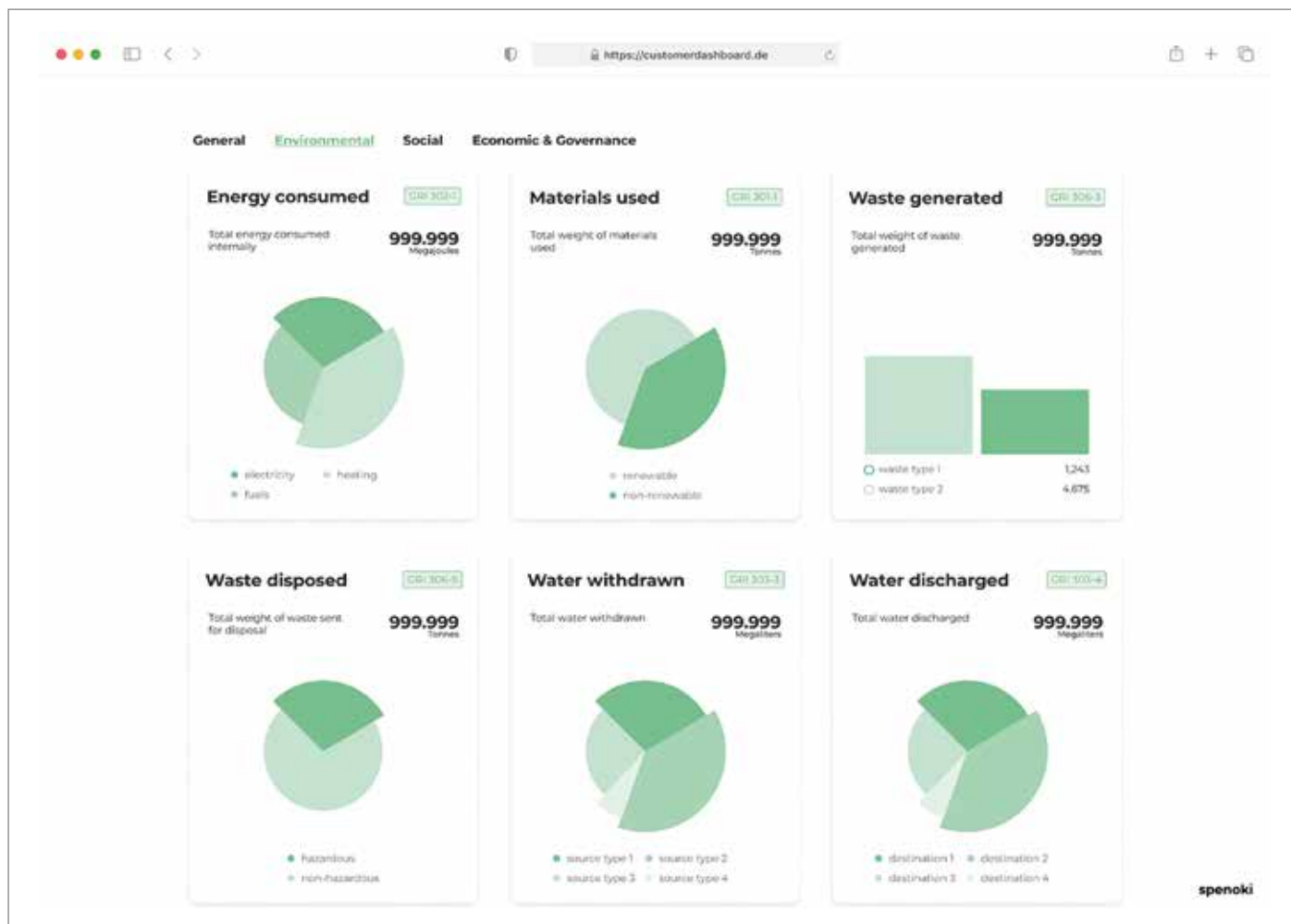


Abbildung 2: Spenoki Nachhaltigkeitsdashboard (© Spenoki Solutions GmbH)

- Weitere Präventionsmaßnahmen gegenüber Lieferanten können auch die Kommunikation der Erwartungen und Standards beinhalten, beispielsweise durch einen Lieferanten-Verhaltenskodex, der in lokalen Sprachen zur Verfügung gestellt wird, oder auch die direkte Unterstützung von strategischen Lieferanten sowie die Förderung lokaler Projekte an Lieferantenstandorten/Regionen zur Stärkung lokaler Rechte und Communities.

Neben diesen vorbeugenden Maßnahmen sieht das Gesetz zusätzlich vor, im Fall von Nicht-Konformitäten geeignete Abhilfemaßnahmen zu erstellen sowie deren Umsetzung zu überwachen. Hierdurch soll gewährleistet werden, dass eingetretene Risiken umgehend angegangen und ihre Auswirkungen so gering wie möglich gehalten werden können. Für diesen und auch für den Fall von nicht erkannten Abweichungen muss zusätzlich eine Beschwerdestelle eingerichtet werden, die es den Betroffenen oder Benachteiligten er-

möglicht, ihr Anliegen vorzubringen. Abschließend fordert das Gesetz, neben der Dokumentation der Überwachung und deren Veröffentlichung durch eine einheitliche Berichterstattung, die Umsetzung von Sorgfaltspflichten auch für Risiken der mittelbaren Zulieferer.

Konkrete Handlungsempfehlungen für betroffene Unternehmen

Zusammenfassend lassen sich somit vier Kernprozesse identifizieren, die für eine erfolgreiche und regelkonforme Umsetzung des Lieferkettensorgfaltspflichtengesetzes vonnöten sind. Mithilfe des Supply Chain Mapping muss die gesamte Lieferkette der einzelnen Materialien und Hilfsstoffe mit allen weiteren einkaufenden Dienstleistungen identifiziert werden. In der Praxis findet man diese Informationen zumeist im Einkauf, in dem strukturierte Daten der aktuell aktiven Lieferanten vorliegen. Um die benötigten Daten abfragen zu können, ist es wichtig, vorab Strukturen im Fachbereich aufzu-

bauen. Darüber hinaus muss eine Awareness für das Erkennen der relevanten Informationen geschaffen werden. Hierfür ist die richtige Aus- und Weiterbildung der Mitarbeiter, aber auch ein kontinuierlicher Austausch mit allen Beteiligten entlang der Lieferkette notwendig.

Anschließend müssen die Daten erfasst werden. Dabei gilt es mithilfe von standardisierten und branchenspezifischen Fragebögen die Daten bezüglich der sozialen und ökologischen Indikatoren zu erheben. Hier können Softwarelösungen von Spenoki dabei unterstützen, anhand von automatisierten Prozessen eine granulare Datenbasis herzustellen und durch Standardisierung die Einbindung der Lieferanten zu erleichtern.

Die erhobenen Daten müssen nun aggregiert und Risiken identifiziert werden. Hier können Dashboards und Hotspot-Analysen helfen, die in den meisten Tools zur Lieferkettens-Datenerhebung bereits mit enthalten sind. Über eine fortlaufende Datenabfrage und -aggregation ist

man in der Lage, ein Lieferketten-Monitoring aufzubauen und schnell auf Risiken oder Abweichungen zu reagieren.

Im letzten, aber nicht minder wichtigen Schritt geht es darum, die gewonnenen Erkenntnisse und die damit entstandene Transparenz an den Gesetzgeber und die Stakeholder zu kommunizieren. Hierbei können Softwarelösungen dabei unterstützen, die ermittelten Informationen in einen standardisierten Bericht zu überführen und somit den Aufwand für die nachweisenden Unternehmen überschaubar zu halten. Diese standardisierten Berichte von Spenoki werden sowohl vom Gesetzgeber als auch von Stake- und Shareholdern akzeptiert.

Literatur

- [1] Christian Ortmeier, Nadja Henningsen, Adrian Langer, Alexander Reiswich, Alexander Karl, und Christoph Herrmann. 2021. Framework for the integration of Process Mining into Life Cycle Assessment. *Procedia CIRP* 98, 163–168. DOI:
- [2] John Elkington. 1997. *Cannibals with Forks: The Triple Bottom Line of 21st Century Business*. Capstone, Oxford.
- [3] John Elkington. 2018. 25 Years Ago I Coined the Phrase "Triple Bottom Line." Here's Why It's Time to Rethink It (2018). Entnommen am 15 März 2022 von <https://hbr.org/2018/06/25-years-ago-i-coined-the-phrase-triple-bottom-line-heres-why-im-giving-up-on-it>.
- [4] United Nation Generalversammlung, Ed. 2015. *Transforming our World: the 2030 agenda for sustainable development*. Resolution verabschiedet durch die Generalversammlung am 25. September 2015. A/RES/70/1.
- [5] European Commission, Ed. 2022. Proposal for a DIRECTIVE OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on Corporate Sustainability Due Diligence and amending Directive (EU) 2019/1937EU. 2022/0051.
- [6] Jungmichel, Norbert, Christina Schampel und Daniel Weiss. 2017. *Atlas on Environmental Impacts - Supply Chains - Environmental Impacts and Hot Spots in the Supply Chain*. Retrieved April 16 2022 from <https://www.adelphi.de/en/system/files/mediathek/bilder/Umweltatlas%20Lieferkette%20-%20adelphi-Systain-englisch.pdf>
- [7] Bundestag, Ed. 2021. Gesetz über die unternehmerische Sorgfaltspflichten in Lieferketten. Verabschiedet durch die Bundstagsversammlung am 16. Juli 2021.



Florian Lösch

florian.loesch@horus.biz

Florian Lösch ist als Principal Consultant und Projektleiter in Beratungsprojekten zum Thema Geschäftsprozessmanagement bei der Horus software GmbH aktiv, die diesbezüglich eine Methode und zugehörige Produkte sowie Services anbietet. Er ist Experte im Bereich des nachhaltigen Geschäftsprozessmanagements, bei dem er auch in der Forschung und der Lehre tätig ist.



Markus Adler

markus.adler@spenoki.de

Markus Adler ist Mitgründer der Spenoki Solutions GmbH, die eine SaaS-Lösung anbietet, die mittelständischen Unternehmen dabei hilft, ihr Nachhaltigkeitsmanagement zu digitalisieren und zu automatisieren. Hier ist er verantwortlich für die Begleitung der Kunden entlang der gesamten Wertschöpfungskette und Übersetzer der Anforderungen des Kunden und des Marktes an das Produkt.



Der grüne Faden für Ihre Digitale Evolution

Wir bei PROMATIS folgen einem selbst entwickelten grünen Faden:

Mit professioneller Beratung und innovativen Digitalisierungslösungen schaffen wir exzellente Geschäftsprozesse: agil, bedarfsgerecht, intelligent und zukunftssicher. Nachhaltige Qualität und Wirtschaftlichkeit sichern wir durch kontinuierliche Verbesserung der eingesetzten Verfahren, Produkte und Services.

Mit unserer Digitalisierungskompetenz und unseren Best Practice-Lösungen begleiten wir Sie auf Ihrer Reise in die Oracle Cloud.

PROMATIS Gruppe
Pforzheimer Str. 160
76275 Ettlingen
+49 7243 2179-0
www.promatis.de

Ettlingen | Hamburg | Berlin | Münster
Wien | Zürich | Denver



Die Bedeutung von transparenten Lieferketten für nachhaltige Elektrobatterien

Franziska Grammes, Circular

Im Rahmen der geplanten EU-Batterierichtlinie soll ein Batteriepass unter anderem Auskunft über Herkunft, Herstellung oder Co2-Emissionen von E-Auto-Batterien geben. Diejenigen, die es schaffen, hier schnell verifizierte nachhaltige Praktiken nachzuweisen, könnten bald klar im Vorteil sein. Circular unterstützt Unternehmen dabei, nachhaltigere Lieferketten aufzusetzen.

Etwa ein Viertel aller Emissionen entsteht über die Art und Weise, wie wir uns fortbewegen. Die sofortige Dekarbonisierung des Verkehrssektors ist daher unabdingbar, um die Pariser Klimaziele zu erreichen. Weltweit investieren Konzerne und Regierungen daher massiv in die Elektromobilität und verfolgen ambitionierte Ziele. In Deutschland lautet die neue Zielmarke der Ampelkoalition: 15 Millionen E-Autos bis 2030. In Summe gibt es bisher aber nur 1,4 Millionen Elektroautos, die zwischen 2010 und 2021 zugelassen wurden [1].

Wie neue Technologien bei der Dekarbonisierung helfen

Etwa 40 Prozent der Treibhausgasemissionen, die bei der gesamten Herstellung eines Elektroautos entstehen, gehen auf die Batterieherstellung zurück. Batterien spielen dementsprechend eine wichtige Rolle für Nachhaltigkeit in der Automobilindustrie (2). Die Dekarbonisierung darf daher nicht erst auf der Straße, sondern muss schon am Anfang der Lieferkette beziehungsweise bei der Rohstoffgewinnung beginnen. Elektrobatterien enthalten Rohstoffe wie

Lithium, Kobalt oder Nickel. Deren Abbau bringt oft Menschenrechtsverletzungen und hohe Emissionen mit sich. Circular [2] ist der führende Anbieter für die Rückverfolgbarkeit von Rohstoffen. Unternehmen können ihre Lieferketten mit Circular vollständig analysieren, verfolgen und verwalten. So kann eine verantwortungsvolle Beschaffung sichergestellt und Ethik- sowie Nachhaltigkeitskriterien können nachweislich berücksichtigt werden. Circular stellt hierzu eine Softwareplattform bereit, die mithilfe eines digitalen Zwillings einen



Abbildung 1: Lieferketten-Tracking mit Circular. Die Software-as-a-Service-Lösung von Circular ermöglicht es Unternehmen, Rohstoffe und Materialien von der Mine bis zum Endprodukt zu verfolgen (Quelle: Circular)

Der Batteriepass



Abbildung 2: So könnte der Batteriepass aussehen. Der digitale Produktpass für Batterien bündelt alle relevanten Informationen entlang des gesamten Lebenszyklus: von der Rohstoffgewinnung bis zum Recycling (Quelle: BMWK)

zuverlässigen Nachweis über die Herkunft sowie Herstellungsschritte des Materials bietet (siehe Abbildung 1). CO₂-Emissionen und weitere ESG-Performance-Daten werden so dynamisch dargestellt und direkt mit dem Materialfluss verknüpft. Environment, Social und Governance (ESG) sind in vielen Unternehmensstrategien bereits heute zu zentralen Faktoren geworden.

„Anders als andere Lösungen zur Rückverfolgbarkeit erfassen wir das tatsächliche Material, während es die Lieferkette durchläuft – nicht nur die Transaktionen zwischen den einzelnen Teilnehmern. Das bedeutet auch, dass wir in der Lage sind, die CO₂-Emissionen bei jedem Schritt in der Lieferkette abzubilden, um genaue

und ganzheitliche Messungen der Umweltauswirkungen eines Produkts vorzunehmen“, so Circular CEO Douglas Johnson-Poensgen. „Unsere Kunden sind so in der Lage, nicht nur die Emissionen innerhalb ihres eigenen Produktionsprozesses (Scope 1 & 2) zu messen, sondern auch die geerbten Emissionen (Scope 3) ihrer Zulieferer zu erfassen. So können sie fundierte Entscheidungen treffen, um ihren CO₂-Fußabdruck zu reduzieren und langfristig Klimaneutralität zu erreichen.“ Unternehmen wie Polestar, Volvo Cars, Jaguar/Land Rover oder Trafigura verfolgen ihre Materialien mit Circular schon heute und bereiten sich somit auf die Erfüllung der bevorstehenden EU-Batterie-Regulierung vor.

Neues EU-Gesetz legt Umweltkriterien für Elektrobatterien fest

Die geplante EU-Batterie-Regulierung wird ab 2026 für alle neu angeschafften Batterien in Fahrzeugen, stationären Speichern und größeren Industriebatterien in Deutschland und Europa verpflichtend. Sie ist das erste Gesetz der Welt, das Umweltkriterien für die Markteinführung von Elektrobatterien festlegt und die gesamte Lieferkette ins Zentrum stellt. Ziel der vorgeschlagenen Regulierung ist es, dass die in der EU auf den Markt gebrachten Batterien vom Anfang bis zum Ende der Lieferkette nachhaltig, zirkulär, leistungsstark und sicher sind. So muss unter anderem

die Herkunft einzelner Rohstoffe genauso nachgewiesen werden können wie der genaue CO₂-Fußabdruck oder der Anteil recycelter Komponenten (siehe Abbildung 2). Mit einem sogenannten „Batteriepass“ werden diese Bestrebungen konkret und messbar umgesetzt. Mit der gewonnenen Transparenz wird auch das Recycling von Rohstoffen in Batterien erleichtert und so die Abhängigkeit von ausländischen Lieferanten verringert.

Startschuss für deutsches „Battery Pass“-Projekt

Das deutsche Bundesministerium für Wirtschaft und Klimaschutz (BMWK) fördert die Entwicklung des digitalen Batteriepasses mit dem „Battery Pass“-Projekt. Ende April 2022 übergab Staatssekretär Michael Kellner (Grüne) in Berlin den Förderbescheid in Höhe von 8,2 Millionen Euro. Zum Konsortium gehören neben Circulor und der Münchner Firma Systemiq mitunter auch die Automobilhersteller AUDI AG und BMW sowie der Chemiekonzern BASF (3). Als Pionier auf dem Gebiet der Rückverfolgbarkeit in der Lieferkette, der bereits erfolgreich Batteriepass-Managementssysteme für Kunden wie Volvo, Polestar oder Rock Tech Lithium entwickelt hat, wird Circulor unter anderem für die technische Umsetzung verantwortlich sein und das Arbeitspaket 4, den „Battery Pass Demonstrator“ leiten. Dieses Arbeitspaket wird inhaltliche und technische Standards zur Simulation von Datenflüssen und Systemtransaktionen verwenden.

Das Projekt soll die Europäische Kommission und wichtige Partner auf globaler Ebene beeinflussen und darüber informieren, wie die technischen Standards und der Inhalt digitaler Identitäten (beziehungsweise Pässe) für Batterien entwickelt und verwaltet werden können, um eine verantwortungsvolle, nachhaltige und zirkuläre Wirtschaft zu ermöglichen.

Das Batteriepass-Management-System von Circulor

Unternehmen sind mit dem Batteriepass-Management-System von Circulor schon heute in der Lage, für den Batteriepass relevante Informationen sicher hochzuladen und zu verwalten. „Der Elektroauto-Batteriemarkt entwickelt sich rasend schnell. Diejenigen, die es schaffen, jetzt verifizierte nachhaltige Praktiken nachzuweisen, sind klar im Vorteil“, ist sich Circulor's CEO Dou-



Ciculor's CEO Douglas Johnson-Poensgen (© Circulor)

glas Johnson-Poensgen sicher. „Wenn es um E-Mobilität geht, sind die Verbraucher inzwischen sehr sensibilisiert für Batterien. Für Unternehmen besteht jetzt die große Chance zu zeigen, dass sie verantwortungsvoll und nachhaltig wirtschaften.“

Quellen

- [1] <https://www.adac.de/rund-ums-fahrzeug/elektromobilitaet/info/elektroauto-bilanz/>
- [2] <https://www.ciculor.com/>
- [3] <https://www.volvocars.com/images/v/-/media/market-assets/intl/applications/dotcom/pdf/c40/volvo-c40-recharge-lca-report.pdf>
- [4] <https://www.bmwk.de/Redaktion/DE/Pressemitteilung/2022/04/20220425-parlamentarischer-staatssekretar-kellner-batterien-mussen-in-der-eu-so-nachhaltig-wie-moglich-sein.html>

Franziska Grammes

fg@circulor.com

Franziska Grammes ist eine in Berlin ansässige Kommunikationsspezialistin mit Schwerpunkt Greentech. Sie glaubt an die Kraft von Kooperation und Innovation zur Lösung der Klima- und Umweltkrise. Seit 2022 ist sie bei Circulor als Director of PR tätig.

BEST OF DOAG ONLINE

Eine Auswahl der besten DOAG News Mai/Juni 2022

Zuvor möchten wir auf das neue Angebot von so genannten **On-Demand-Tickets** für unsere Events aufmerksam machen. Für alle, die nicht persönlich auf einer Konferenz dabei sein konnten, gibt es nun die Möglichkeit bis zu einem Jahr nach der Veranstaltung **Zugriff auf alle verfügbaren Vortragsfolien und -aufzeichnungen zu haben**. Bitte buchen Sie dafür zuerst unter <https://shop.doag.org/events/ondemand/shop/> ein Ticket in unserem Shop und suchen Sie sich anschließend die Vorträge in der jeweiligen Agenda aus!

CloudLand 2022: Interview mit Keynote Speaker Kurt Garloff



Der CTO der Open Source Business Alliance e.V. eröffnete den Reigen der hochkarätigen Keynote Speaker am 29.06., dem ersten Festival-Tag. Marcos López interviewte ihn im Vorfeld. <https://www.doag.org/de/home/news/cloudland-2022-interview-mit-keynote-speaker-kurt-garloff/>

DOAG Datenbank Kolumne



Dagmar Förster erklärt, wie sich ein sinnvoller Frühlingsputz auf den Datenbanken mit DBSAT durchführen lässt, um ein wenig aufzuräumen. <https://www.doag.org/de/home/news/doag-datenbank-kolumne-kleiner-fruehlingsputz-auf-den-datenbanken-mit-dbsat/>

DOAG-Studio-Interview



In der Interview-Reihe „Ronny trifft...“ erläutert PL/SQL-Guru Steven Feuerstein, was aus der App fabe wurde, wie der Klimaschutz Sache eines jeden einzelnen ist und was es mit APEXd2d auf sich hat. <https://www.doag.org/de/home/news/doag-studio-interview-ronny-trifft-steven-feuerstein/>

DOAG Datenbank Kolumne



Ulrike Schwinn zeigt, mit welchem kleinen Trick sich eine SQL Id ohne Ausführung ausgeben lässt. <https://www.doag.org/de/home/news/doag-datenbank-kolumne-kann-man-die-sql-id-ohne-sql-statement-ausfuehrung-erhalten/>

DOAG-Studio-Interview



In der Interview-Reihe „Ronny trifft...“ begegnet Entwickler und Autor Ronny Weiß Oracles weltweit bekannten Produktmanager Gerald Venzl, der wertvolle Empfehlungen für junge Entwickler gibt. <https://www.doag.org/de/home/news/doag-studio-interview-ronny-trifft-gerald-venzl/>

Begeisterung auf der APEX connect 2022



Was gibt es Schöneres als eine Veranstaltung nach zwei Jahren wieder vor Ort zu besuchen? Ein kleiner Rückblick von Moritz Rissmann auf die erfolgreiche APEX connect 2022, die in Düsseldorf stattfand. <https://www.doag.org/de/home/news/begeisterung-auf-der-apex-connect-2022/>

DOAG-Studio-Interview



Im Gespräch mit Johannes Ahrends erzählt der ehemalige Übersetzer und Dolmetscher Eero Mattila, wie er als Quereinsteiger in die IT kam und Experte für Hochverfügbarkeit wurde. <https://www.doag.org/de/home/news/doag-studio-interview-eero-mattila-ueber-die-sprachen-zur-it/>

Devs on Tape – ein Podcast von DOAG-Aktiven



Carolyn Hagemann und Kai Donato erzählen in einem kurzweiligen Interview, wie es zum neuen Stern am DOAG-Himmel kam. <https://www.doag.org/de/home/news/devs-on-tape-der-neue-podcast-der-doag/>



Wir begrüßen unsere neuen Mitglieder

Natürliche Mitglieder:

- Daniel Diel
- Neda Vaziri
- Isabell Gellweiler
- Modjibullah Moheby
- Felix Specht



Termine

Juli

07

08.07.2022

New Features in Multitenant mit 21c.
DB WebSession mit Markus Flechtner
Online

August

08

12.08.2022

Backup und Data Guard mit odacli.
DB WebSession mit Stefan Seck
Online

September

09

06.09.2022

Flows for APEX – prozessorientierte Apps erstellen.
Berliner Expertenseminar mit Niels de Bruijn und Moritz Klein
Berlin

15.09.2022

DB-Programmierung
Dev Talk mit Jürgen, Ulrike, Sabine und Christian
WebSession

19.09.2022

European NetSuite User Days
im Rahmen der DOAG 2022 Konferenz
+ Ausstellung in Nürnberg.
Nürnberg Convention Center Ost

20.09.2022

DOAG 2022 Konferenz + Ausstellung
Zwei Tage Konferenzerlebnis + Thementag
Nürnberg Convention Center Ost

29.09.2022

Rückblick K&A - Dev Talk mit Carolin Hagemann und Niels de Bruijn
Online

Impressum

Red Stack Magazin inkl. Business News wird gemeinsam herausgegeben von den Oracle-Anwendergruppen DOAG Deutsche ORACLE-Anwendergruppe e.V. (Deutschland, Tempelhofer Weg 64, 12347 Berlin, www.doag.org), AOUG Austrian Oracle User Group (Österreich, Lassallestraße 7a, 1020 Wien, www.aoug.at) und SOUG Swiss Oracle User Group (Schweiz, Dornacherstraße 192, 4053 Basel, www.soug.ch).

Red Stack Magazin inkl. Business News ist das User-Magazin rund um die Produkte der Oracle Corp., USA, im Raum Deutschland, Österreich und Schweiz. Es ist unabhängig von Oracle und vertritt weder direkt noch indirekt deren wirtschaftliche Interessen. Vielmehr vertritt es die Interessen der Anwender an den Themen rund um die Oracle-Produkte, fördert den Wissensaustausch zwischen den Lesern und informiert über neue Produkte und Technologien.

Red Stack Magazin inkl. Business News wird verlegt von der DOAG Dienstleistungen GmbH, Tempelhofer Weg 64, 12347 Berlin, Deutschland, gesetzlich vertreten durch den Geschäftsführer Fried Saacke, deren Unternehmensgegenstand Vereinsmanagement, Veranstaltungsorganisation und Publishing ist.

Die DOAG Deutsche ORACLE-Anwendergruppe e.V. hält 100 Prozent der Stammeinlage der DOAG Dienstleistungen GmbH. Die DOAG Deutsche ORACLE-Anwendergruppe e.V. wird gesetzlich durch den Vorstand vertreten; Vorsitzender: Björn Bröhl. Die DOAG Deutsche ORACLE-Anwendergruppe e.V. informiert kompetent über alle Oracle-Themen, setzt sich für die Interessen der Mitglieder ein und führt einen konstruktiv-kritischen Dialog mit Oracle.

Redaktion:

Sitz: DOAG Dienstleistungen GmbH
(Anschrift s.o.)
ViSdP: Fried Saacke
Redaktionsleitung Red Stack Magazin:
Martin Meyer
Redaktionsleitung Business News:
Marcos López
Kontakt: redaktion@doag.org
Weitere Redakteure (in alphabetischer Reihenfolge): Markus Adler, Mario Beck, Dr. Birgit Breitschuh, Bruno Cirone, Anna Collard, Bettina Ernst, Steven Feuerstein, Franziska Grammes, Philipp Hartenfeller, Sabine Heimsath, Florian Lösch, Martin Meyer, Samuel Nitsche, Stefan Oehrli, Jochen Rahm, Philipp Salvisberg, Christian Schwitalla, Jürgen Sieben, Günther Stürner, Ulrike Schwinn, Carsten Thalheimer.

Titel, Gestaltung und Satz:

Diana Tkach
DOAG Dienstleistungen GmbH
(Anschrift s.o.)

Fotonachweis:

Titel: @ Storyset | www.freepik.com
S. 14: © AaronJOlson | www.pixabay.com
S. 16: © PDPHOTOS | www.pixabay.com
S. 23: © Hollanddesign | www.pixabay.com
S. 32: © Vectorjuice | www.freepik.com
S. 40: © Hans | www.pixabay.com
S. 47: © Castles | www.pixabay.com
S. 52: © Tumisu | www.pixabay.com
S. 57: © Riki32 | www.pixabay.com
Titel S. 60: © Vectorjuice | www.freepik.com
S. 65: © CARE Cambodia
S. 67: © CARE Cambodia
S. 68: © CARE Cambodia
S. 69: © CARE Cambodia
S. 70: @ pikisuperstar | www.freepik.com

S. 73: © Spenoki Solutions GmbH
S. 74: © Spenoki Solutions GmbH
S. 76: © Anika Büssemeier
S. 79: © Circular
S. 81: © Pch Vector | www.freepik.com

Anzeigen:

sponsoring@doag.org

Mediadaten und Preise:

www.doag.org/go/mediadaten

Druck:

WIRMachenDRUCK GmbH,
www.wir-machen-druck.de

Alle Rechte vorbehalten. Jegliche Vervielfältigung oder Weiterverbreitung in jedem Medium als Ganzes oder in Teilen bedarf der schriftlichen Zustimmung des Verlags.

Die Informationen und Angaben in dieser Publikation wurden nach bestem Wissen und Gewissen recherchiert. Die Nutzung dieser Informationen und Angaben geschieht allein auf eigene Verantwortung. Eine Haftung für die Richtigkeit der Informationen und Angaben, insbesondere für die Anwendbarkeit im Einzelfall, wird nicht übernommen. Meinungen stellen die Ansichten der jeweiligen Autoren dar und geben nicht notwendigerweise die Ansicht der Herausgeber wieder.

Inserentenverzeichnis

B4Bmedia.net AG
<https://e-3.de>

U 4

DOAG e.V. U 2, U 3, S. 3, S. 7 S. 13
www.doag.org

MuniQsoft Consulting GmbH
www.muniqsoft-consulting.de

S. 15

Promatis Gruppe
www.promatis.de

S. 75

DOAG

DOAG 2022

Datenbank

on demand



DOAG DATENBANK 2022 VERPASST?

JETZT ON-DEMAND-TICKET BUCHEN UND
VORTRAGSAUFZEICHNUNGEN ANSCHAUEN!

Alle Angebote im On-Demand-Ticket-Shop



Alles, was die SAP-Community wissen muss,
finden Sie monatlich im E-3 Magazin.

Ihr Wissensvorsprung im Web, social media
sowie PDF und Print: e-3.de/abo

Wer nichts weiß, muss alles glauben!

Marie von Ebner-Eschenbach



SAP® ist eine eingetragene Marke der SAP SE in Deutschland und in den anderen Ländern weltweit.

www.e-3.de