Nr. 1 | Februar 2025 | ISSN 2366-7915 | www.doag.org

Red Stack Magazin

DOAG SOUG swiss oracle user group AOUG AUSTRIAN ORACLE USER GROUP

SECURITY

Aus der Praxis

Ein OCI Document Understanding Deep Dive

Im Interview

Bruno Cirone und Cornelia Heyde, DOAG

Security

Oracle Cloud Infrastructure Security Basics

GROßES
COMMUNITY-PROGRAMM!

JavaLand

AM NÜRBURGRING

JAVALAND

1. - 3. APRIL 2025

JAVALAND.EU

#JAVALAND

Präsentiert von: iJUG Verbund · Heise Medien · DOAG

Veranstalter: JavaLand

Bruno Cirone
Themenverantwortlicher Sicherheit, Mitglied Delegiertenversammlung Datenbank

Ausgabe Nr. 1/2025
auf Abruf!

Liebe Mitglieder, liebe Leserinnen und Leser,

das Thema Security ist aktuell und spannend wie nie zuvor. Es ist kein „Alter Hut", sondern ein existenzieller Baustein der IT und natürlich damit auch der Datenbanken. Die digitale Landschaft verändert sich rasant, und wir müssen mit den Bedrohungen der Unternehmensdaten umgehen. In jüngerer Zeit ist noch ein Faktor hinzugekommen: die Künstliche Intelligenz (KI).

KI bietet sowohl Chancen als auch Risiken für die Oracle-Sicherheit. Einerseits können KI-gestützte Tools die Sicherheit verbessern, andererseits können sie auch von Angreifern missbraucht werden.

Durch KI werden die Bedrohungen für Oracle-Systeme potenziell komplexer. Traditionelle Bedrohungen wie zum Beispiel Ransomware, SQL-Injection-Angriffe und das Ausnutzen von Software-Schwachstellen bleiben bestehen. Hinzu kommen nun auch KI-basierte Angriffe, die unter anderem automatisiert Schwachstellen aufspüren.

Was können Unternehmen konkret tun, um ihre Umgebungen zu schützen? Neben den klassischen Sicherheitsmaßnahmen wie regelmäßigen Updates, Zugriffskontrollen, Firewalls und Verschlüsselung ist es von großer Bedeutung, sich mit den Möglichkeiten und Risiken von KI im Sicherheitskontext auseinanderzusetzen. Die Implementierung von KI-basierten Sicherheitstools kann eine wertvolle Ergänzung sein. Es ist jedoch wichtig, diese Tools sorgfältig auszuwählen.

Darüber hinaus spielt die Schulung der eigenen Mitarbeiter eine entscheidende Rolle. Sie müssen für diese neuen Bedrohungen sensibilisiert werden, und lernen, wie sie KI-basierte Angriffe erkennen können. In diesem Zusammenhang ist es unerlässlich, dass eine gesonderte Testumgebung existiert, um die verschiedenen Angriffsszenarien zu testen.

Die Sicherheit von Oracle-Systemen ist in einer zunehmend vernetzten und KI-geprägten Welt eine ständige Herausforderung. Unternehmen müssen proaktiv handeln und eine umfassende Sicherheitsstrategie implementieren, die sowohl traditionelle als auch KI-basierte Bedrohungen berücksichtigt. Nur so können sie ihre wertvollen Daten effektiv schützen und den sicheren Betrieb ihrer geschäftskritischen Anwendungen gewährleisten.

Neben Beiträgen zum Thema Sicherheit bieten wir Ihnen in diesem Heft spannende Artikel zu den Themen KI, Cloud, Datenbank und ein Interview mit Andreas Gaede zu Oracle Forms.

Ich wünsche Ihnen viel Vergnügen beim Lesen dieser Ausgabe.

Bruno Cirone

Härtung der Standard Edition 2

Oracle 23ai SQL-Firewall – Datensicherheit in der Datenbank

Einleitung

Security

Cloud

KI

Datenbank

Herr der Daten: APEX, VPD und Data Redaction – die Gefährten

Interview mit Andreas Gaede

Oracle Cloud Infrastructure Security Basics

Datenqualität – was gibt es Neues in der Oracle Datenbank 23ai?

Hochverfügbar, aber bitte etwas günstiger

News

Intern

TIMELINE

11. Oktober 2024

Die KI Navigator Roadshow startet in Ravensburg und bringt Im Zeitraum vom 11. Oktober bis 7. November unseren roten Event- und Partybus mit Fried Saacke am Steuer zu diversen Gruppen, Meetups oder Firmen mit Bezug zu Künstlicher Intelligenz in ganz Deutschland zu einem gemeinsam KI-Event. Mit an Bord sind Getränke, Pizza und gute Laune. An jeder Station gibt es immer einen spannenden KI-Vortrag.

21. Oktober 2024

Das Regionaltreffen München/Südbayern wird zur Tourstation der KI Navigator Roadshow und bietet gleich zwei Impulsvorträge zu den Themen „Kultur frisst KI: Wie du KI erfolgreich bei dir im Unternehmen einführst“ und „Azure OpenAI & Azure Container Apps, Dapr: Intelligent Apps State of the Art“.

24. Oktober 2024

Das Regionaltreffen Dresden findet statt. Auf der Agenda steht ein Vortrag zum Thema „Semantische Suche und RAG: Innovativer Einsatz der Vektorsuche in der Oracle-Datenbank 23ai“ von Jenny Pretzsch und Toni Roob.

29. Oktober 2024

Das Regionaltreffen Osnabrück/Bielefeld/Münster hat mit „State of the Dolphin: Neuigkeiten, Releases und Features rund um MySQL“ von Referent Mario Beck und „KI? Aber sicher! Wie KI Systeme den Datenschutzanforderungen gerecht werden“ von Wolf Beckmann zwei Vorträge im Programm.

29. Oktober 2024

Die Roadshow zur KI Navigator 2024 hält in Oldenburg und ist zu Gast bei der JUG Bremen/OL Open Knowledge. Tim Wüllner spricht über „Die Architektur für Sprachmodelle in der Praxis – Retrieval Augmented Generation“.

5. November 2024

Der DOAG Tourbus macht im Rahmen der KI Navigator Roadshow einen Tourstopp in Berlin und kommt zur DOAG Geschäftsstelle mit der DOAG Gruppe Region Berlin/Brandenburg. Oliver Szymanski hält einen Impulsvortrag zum Thema „KI und Huskies – Warum wir zu viel und zu wenig von KI erwarten“.

7. November 2024

Beim von Christian Schwitalla moderierten DOAG DevTalk spricht Matthias Schulz über „Modernes SQL – JSON-SQL“.

8. November 2024

In der DB WebSession mit Stefan Oehrli erfahren die Zuhörenden alles zum Thema „Oracle Maximal Database Security Architektur. Effiziente Strategien zur Sicherung von Oracle-Datenbanken“.

12. November 2024

Beim Regionaltreffen Hamburg gibt es gleich vier Vorträge zur Auswahl. „Vector Search“ mit Jürgen Paasch, „Rethink Backup – eine Übersicht über verschiedene Backup- und Restore-Konzepte für die Oracle-Datenbank“ von Benjamin Kurschies, „PL/SQL Performanceoptimierung & Oracle APEX Security“ mit Jan Gorkow und „Oracle APEX Security – Eine Kurzübersicht über aktuelle Lösungen und Empfehlungen“ von Florian Grasshoff.

18. und 19. November 2024

Erneut finden die European NetSuite User Days wieder im Nürnberg Convention Center (NCC) mit reger internationaler Beteiligung statt.

19. bis 22. November 2024

Die DOAG 2024 Konferenz + Ausstellung findet im Nürnberger NCC Ost statt. Rund 1500 Teilnehmerinnen und Teilnehmer freuen sich über ein vielfältiges Programm zu den sechs Streams Datenbank, Infrastruktur, Development, Middleware, Strategie & Softskills sowie Data Analytics & KI. Am letzten Tag findet abschließend ein Schulungstag statt.

20. und 21. November 2024

Die zweite Ausgabe der KI Navigator findet parallel zur DOAG 2024 Konferenz + Ausstellung im Nürnberger NCC Ost statt. Begleitet wird das Programm von einer Ausstellung in der innovativen AI Experience World mit vielen Showcases. Etwa 650 Interessierte kommen zusammen, um sich über die Zukunft der Künstlichen Intelligenz intensiv auszutauschen. Im Vergleich zu letztem Jahr interessieren sich diesmal etwa 150 Teilnehmerinnen und Teilnehmer mehr für die Konferenz zur Praxis der KI in IT, Wirtschaft und Gesellschaft.

21. November 2024

Zu einem spontanen Regionaltreffen NRW kommt es auf der DOAG 2024 Konferenz + Ausstellung. Organisator und DOAG-Vorstandsmitglied Armin Wildenberg freut sich über einen regen Austausch. „Wie geht es weiter mit der Regionalgruppe NRW in 2025?“ lautet das Motto.

13. Dezember 2024

„Prüfung Ihrer Oracle-Installation und -Konfiguration mit ORAchk“ heißt das Thema der DB WebSession mit Andrew Lacy.

19. Dezember 2024

Im letzten DOAG DevTalk des Jahres geht es um das Thema “Oracle Forms 14.1.2 and the native built in RESTSERVICE in examples and programming”. Refrenten sind die Forms-Experten Michael Ferrante und Frank Hoffmann.

10. Januar 2025

In der ersten DB WebSession im neuen Jahr erfahren die Teilnehmenden von Borys Neselovskyi alles zum Thema „PostgreSQL in Kubernetes – mission impossible?“

10 JAHRE JAVALAND

JavaLand

www.javaland.eu

JAVALAND 2024 VERPASST?

ALLE ON-DEMAND-ANGEBOTE IM TICKETSHOP

JETZT ON-DEMAND-TICKET BUCHEN UND
VORTRAGSAUFZEICHNUNGEN ANSCHAUEN!

Präsentiert von: iJUG Verbund · Heise Medien · DOAG

Veranstalter: JavaLand

„Die Security wird sehr häufig aber durch die Routinen und Abläufe der Mitarbeiter beeinflusst und dadurch kommt es zu Nachlässigkeiten."

Martin Meyer, Redaktionsleiter des Red Stack Magazin, sprach mit Bruno Cirone und Cornelia Heyde über das Thema IT-Security und über Gefahren und Sicherheitslücken, Datenbank-Updates, Passwörter und den Sicherheitsstandard in deutschen Unternehmen und Behörden.

Könnt ihr euch kurz vorstellen, womit beschäftigt ihr euch beruflich?

Bruno: ich beschäftige mich seit 1986 mit Oracle. Seit 1989 bin ich selbstständig und setze mich unter anderem mit den Themen Administration, Tuning, Security und Katastrophenprävention auseinander. Bei dem Thema Katastrophenprävention geht es um die Frage „Was passiert, wenn…" zum Beispiel ein Rechenzentrum zum Opfer von Brand, Hochwasser, Sabotage usw. wird, oder von Ransomware angegriffen wird.
Cornelia: Ich beschäftige mich seit dem Jahr 2000 mit Oracle. Das passte gut zu meinen damaligen Themen der Microsoft Technologien wie zum Beispiel SQL-Server. Inzwischen arbeite ich hauptsächlich als Trainerin für Oracle-Technologien in Dresden. Ich halte verschiedene selbstentwickelte Praxisworkshops zu Oracle. Die Themenbereiche reichen von der Datenbank-Administration bis zur Hochverfügbarkeit mit Clusterware und RAC, Cloud-Control und natürlich Datenbank-Security.

Wo liegen eurer Meinung nach die größten Gefahren oder Sicherheitslücken in der IT?

Bruno: Meiner Meinung nach liegen die größten Gefahren in den Routinen, die die Leute haben, in den Personen selbst. Die Systeme haben im Regelfall eine gute Ausarbeitungsstufe oder einen guten Reifegrad erreicht. Die Security wird sehr häufig aber durch die Routinen und Abläufe der Mitarbeiter beeinflusst und dadurch kommt es zu Nachlässigkeiten. Zum Beispiel werden Produktivdaten in den Testdatenbanken nicht anonymisiert übernommen. Auf diese Daten haben danach mehr User Zugriff als in der Produktion. Nacharbeiten werden manchmal aus Zeitmangel kurzfristig verschoben und danach häufig vergessen. Genau das sind die Lücken die Angreifer besonders gerne ausnutzen. Und da sehe ich schon eine Gefahr drin, die man kaum in den Griff bekommt.
Cornelia: Ja, es fehlt sehr oft am Bewusstsein für Security. Wenn ich beispielsweise eine E-Mail mit einem Link darin erhalte, dann taucht die Frage auf: Kann ich den so einfach öffnen? Oft ist die Idee, „was kann schon passieren", ich schaue nach. Dann habe ich vielleicht schon das Problem, dass dieser Link Sicherheitslücken ausnutzt und die E-Mail-Adressen der Mitarbeiter des Unternehmens durch das Öffnen des Links weitergeleitet wurden. Es gibt weitere Gefahrenstellen wie das Zusammenspiel der einzelnen Komponenten im Rechenzentrum. In der Infrastruktur gibt es verschiedenste Möglichkeiten, Daten von Servern oder Storage, die eigentlich nicht sichtbar sein sollten, „zu finden". Nach Wartungen werden beispielsweise festgelegte Sicherheitsroutinen nicht wieder eingesetzt, weil gerade keine Zeit ist. In diesem Bereich gibt es noch ganz viele Lücken, von denen viele Nutzer nichts wissen beziehungsweise sich keine Gedanken machen.

Weiterhin muss allen Mitarbeitern bekannt sein, wie Security-Regeln des Unternehmens angewendet werden müssen. Dieses Bewusstsein muss erarbeitet werden, um Routinen zu verstehen und nicht zu vernachlässigen. Ich denke, hier ist noch ganz viel zu tun.

Welche Security-Themen sind eurer Meinung nach besonders wichtig bei Unternehmen und Verwaltungen und welche sind besonders gefragt?

Bruno: Ein Thema, das sehr wichtig ist, sind die Definitionen der Zugriffe auf die Daten. Man denke beispielsweise an Daten zu Forschungsergebnissen, Finanzdaten usw. Leider werden immer noch sehr pauschal Rechte vergeben. Es ist natürlich einfacher einen Grant to Public zu machen, anstatt sich Gedanken zu machen, wer welches Recht benötigt. Oracle bietet hierzu mit den Privilege Capture eine Möglichkeit genutzte Rechte zu erfassen. Mit dieser Zusammenstellung können danach individuelle Rollen definiert und jedem User zugeordnet werden.
Cornelia: Ja, das sehe ich auch so.

Aus dem Behördenbereich habe ich oft Kunden in meinen Kursen. In dieser Branche geht es darum, wieder das Bewusstsein zu schärfen, wie mit sensiblen Daten umgegangen werden muss. Ein weiterer Punkt an dieser Stelle ist die Frage, wo die Produktionsdaten überhaupt gespeichert werden sollen. Welche Security-Regeln gelten für diese Daten? Aber wie sieht es mit den Backups aus? Gelten bei Backups die gleichen Security-Regeln für sensible Daten? Sollten diese vielleicht an einem anderen Standort/Rechenzentrum vor Zugriff geschützt werden? Hier ist noch viel Luft nach oben, um Daten entsprechend einem IT-Konzept abzulegen.

Wie verhalte ich mich bei Updates richtig?

Bruno: Diese Frage ist nicht einfach zu beantworten. Zwingend erforderlich ist ein klares Konzept, wie man diese Updates in die Produktion bringt. Im Regelfall wird das sehr klassisch gemacht. Die Updates werden auf Testsystemen ausgerollt. Danach wird später ein bestimmter Termin festgelegt, um dies dann in der Produktion umzusetzen. Häufig entstehen dabei Schwierigkeiten, denn man kann so viel testen, wie man will, die Realität ist letztendlich die Datenbank zu dem Zeitpunkt selbst.

Ich sage da immer ganz gerne, dass nichts realer als die Realität ist. Das liegt einfach daran, dass in der Produktion andere Lastverhältnisse, mehr User und ganz andere Datenkonstellationen vorhanden sind.

Was ich bei Updates generell empfehle, ist, ob Security oder nicht Security, dass man für die produktive Inbetriebnahme betriebsärmere Zeiten wählen sollte. Zum Beispiel Gründonnerstag, damit man am Freitag mit einer geringeren Mannschaft testen kann. Sollten Probleme existieren, hat man die Zeit Updates zurückzurollen. Auf jeden Fall sollten die entsprechenden verantwortlichen Personen ein bis zwei Tage nach der Installation eines Patches immer verfügbar sein.
Cornelia: Dann kommt noch hinzu, dass in der Praxis oft nicht nur die Datenbank gepatcht werden muss, sondern auch andere Systemkomponenten wie Betriebssystem oder Netzwerk. Dabei ist schon vorher zu prüfen, ob es zusätzliche dokumentierte Sicherheitslücken auf allen Systemen gibt – auch außerhalb der Oracle-Welt. Es kann auch sein, dass eine Security-Einstellung oder Protokoll nach einem Patch sich an einer Stelle geändert hat, an der man nicht damit gerechnet hat. Dort gibt es immer wieder Überraschungen. Im Fehlerfall startet die Suche und alle weiteren Aufgaben müssen warten. Das heißt, man sollte die Möglichkeit im Hinterkopf behalten, das System eventuell noch einmal auf den Stand vor dem Patch zurücksetzen zu können. Dabei empfehle ich im-

mer, dieses wichtige Szenario der Wiederherstellung zu üben, so dass man vorbereitet ist. Wie gesagt, wenn im Test alles funktioniert, ist das schön. Aber die Praxis kann doch etwas anders aussehen.

Wo besteht euerer Meinung nach der größte Nachholbedarf hinsichtlich IT-Security in deutschen Unternehmen?

Bruno: Nach meinem Empfinden haben wir zum einen viel zu wenig Fachpersonal im Bereich Security. Entsprechende Aufgaben werden teilweise von anderen Berufsbereichen überlagert oder übernommen. Zum anderen fehlen die entsprechenden Ressourcen. Zum Beispiel fehlt oft die Möglichkeit, überhaupt bestimmte Szenarien zu testen oder sich mit anderen Leuten auszutauschen. Das ist, glaube ich, wohl ein Thema, bei dem es in der deutschen IT, gerade im Behördenbereich, teilweise sehr, sehr stark mangelt. Kurz gesagt: Ressourcen und Personal.
Cornelia: In allen Bereichen einer Behörde oder eines Unternehmens müssen alle Mitarbeiter für Security-Themen sensibilisiert werden, insbesondere solche ohne oder mit wenig IT-Erfahrung. Diese Mitarbeiter müssen lernen, wie man die IT-Sicherheit anwendet. Dazu zählt u.a. der Umgang mit Passwörtern. Ich denke da an „Methoden" des Merkens von Passwörtern wie das Aufschreiben auf Zettel, Verwendung von einfachen oder unsicheren Passwörtern beziehungsweise die Nutzung eines einzigen Passworts für alle Anwendungen. Auch wird oft „vergessen", Datenbank- Passwörter routinemäßig zu ändern. Die Einbindung der „Security- Neulinge" ist damit einer der wichtigsten Punkte, da diese vor einem hohen (Security)-Berg stehen, den sie versuchen, Schritt für Schritt zu erklimmen. Hier braucht es mehr an Hilfestellungen zum Beispiel durch eine interne Schulung. Vernetzung und Lernen von Kollegen sind daher wichtige Aspekte in diesem Bereich.

Welche Negativbeispiele für Sicherheitslücken fallen euch spontan so ein?

Bruno: Mir fällt spontan ein, dass es immer noch unsichere Passwörter gibt und dass das Niveau der Passwörter bei produktiven Datenbanken niedrig ist. Außerdem haben wir immer noch Passwörter aus der Version 10 die in der Version 19 übernommen werden müssen. Damit die Anwendungen weiterlaufen, werden bestimmte Parameter so gesetzt, dass sich die Passwörter wie bei der Version 10 verhalten. Das kann nicht der Weg sein. Das sind aus meiner Sicht zwei gravierende Lücken. Eine andere Problematik, die sich daraus ergibt, ist, dass ich bei einigen produktiven Datenbanken immer noch den User Scott mit dem Passwort Tiger sehe. Wenn zum Beispiel ein Grant to Public gemacht worden ist, kann auch dieser User Scott selbstverständlich die produktiven Daten einsehen. Ganz abgesehen davon, dass man mit dem User Scott ein *Denial of Service* in kürzester Zeit hinbekommt. Das heißt, meine Empfehlung wäre, die entsprechenden Standard-Passwörter auf jeden Fall zu ändern, und nicht benötigte User aus dem System zu verbannen.
Cornelia: Wenn es um das Thema Passwörter geht, geht es ja auch darum, welche Passwortregeln im Unternehmen vorgeschrieben sind. Hier gab es ja schon Hinweise in der letzten Frage. Die Passwortregeln müssen allen Mitarbeitern bekannt sein, zum Beispiel wie oft ich mich anmelden kann, ohne dass mein Account gesperrt wird. Es gibt verschiedene Einstellungen, die relativ einfach an der Datenbank vorgenommen werden können. Für effektives Arbeiten sollte dem Administrator bekannt sein, welche Arten von Daten gespeichert sind, insbesondere sensible Daten. Diese müssen zusätzlich geschützt werden. Denn oft gibt es den ein oder anderen Benutzer, der Privilegien erhalten hat, die nicht für ihn gedacht sind oder die er aufgrund einer Migration der Datenbank geerbt hat. Darauf muss ein besonderes Augenmerk gelegt werden. Es gibt in der Datenbank einige interne Mittel (Views) und Werkzeuge beispielsweise Privilegien-Analyse, um unberechtigte Privilegien aufzuspüren. Ein Thema, dass man in der Security gerne vergisst, ist die Frage, was eigentlich mit Backups passiert. Sind sie unverschlüsselt, passwortgeschützt oder vielleicht bereits verschlüsselt? Auch die unberechtigte Nutzung von Backup oder Dump Files sowie der Dump von Datenblöcken kann für einen Angreifer interessant sein. Diese Lücke sollte nicht unterschätzt werden, meiner Meinung nach wird sie aber von den meisten Benutzern häufig unterschätzt.

Ist die Cloud sicherer oder die On-Premises-Datenbank?

Bruno: Ich glaube, dass kann man pauschal so nicht sagen. Auch wenn die Möglichkeiten der Cloud durch Verschlüsselung und die Verbindungen, die auch verschlüsselt sind, mit Sicherheit besser sind, müssen beide gleich gut geschützt werden. Hier spielt es auch keine Rolle, ob die eine oder die andere ein bisschen besser ist, denn wenn die Sicherheitslücke existiert und genutzt wird, spielt es keine Rolle auf welcher Plattform diese genutzt wird – ganz egal, ob Cloud oder On-Premises.
Cornelia: Es geht ja nicht nur um die Datenbank. Es sollten alle Komponenten (z. B. Netzwerk und Netzwerkkomponenten) auf dem Weg vom Client zur Datenbank betrachtet werden. Die Datenbank ist nur ein kleiner Teil im Kontext von Security. Alles zählt zusammen, die Infrastruktur und die Administratoren, die diese verwalten. Es sollte festgelegt werden, an welcher Stelle, welche Security-Features eingesetzt werden müssen. Damit kann ein gutes Gefühl entstehen, dass alle Daten bestmöglich geschützt übertragen werden.

Könnt ihr einen Ausblick hinsichtlich zukünftiger Security-Entwicklungen geben? Gibt es schon KI-Unterstützung?

Bruno: Es gibt natürlich ein paar Ausblicke. Allein die bisherigen Versionen, 23 ai und Nachfolgende, die mit AI arbeiten, stellen Security-Experten vor erhebliche Anforderungen. Es geht darum, dass teilweise Daten aus dem Unternehmen in eine AI-Engine übertragen werden. Damit könnten auch sensitive Daten verloren gehen. Das stellt mit Sicherheit die Security-Verantwortlichen vor sehr große Probleme. Auch da weiß man eigentlich noch nicht genau, wie man Daten am besten schützen kann. Da werden mit Sicherheit in kürzester Zeit Erfahrungen gesammelt werden müssen.

Auf der anderen Seite sehe ich aber auch, dass es jetzt schon die ersten Scanner gibt, die mit AI werben oder mit AI schon teilweise Scans machen. Wobei ich ehrlich gesagt nicht genau weiß, wie AI in diesen Produkten wirklich eingesetzt wird. Ich glaube, der eine macht daraus einfach nur einen komplexeren Entscheidungsbaum irgendwelcher Daten und der andere macht wirklich daraus ganz andere Analysen. Also ich erwarte, dass noch ein bisschen Zeit vergehen wird, bis wir wirklich klare Informationen und klare Vorstellungen davon haben, was auf uns zukommt.

Cornelia: Ich kann Bruno an dieser Stelle voll zustimmen. Da fehlt es noch an Forschung hinsichtlich der Frage, wie geht es weiter. Die Entwicklungsschritte sind sehr schnell und alle Security-Experten versuchen, diesen Schritten Stand zu halten und angemessene Maßnahmen zu finden. Hier ist derzeit keine AI-Engine so weit, dass sie uns alle Geheimnisse erklärt. Aber es gibt schon verschiedene sehr gute Einsatzbeispiele für Routine-Aufgaben. Ich denke, hier passiert noch sehr, sehr viel in einer spannenden Zukunft.

Vielen Dank für das Gespräch.

CORNELIA HEYDE

Cornelia Heyde ist Diplom-Ingenieurin für Automatisierungstechnik und Technische Kybernetik und begann ihre Laufbahn nach einer Ausbildung zum Microsoft Certified Systems Engineer (MCSE) als Trainerin/Dozentin für Windows Server und Netzwerke sowie Oracle-Datenbanken (ab Version 8.1). Seit 2007 ist sie Senior Dozentin für Datenbank-Technologien bei Robotron in Dresden. Anfangs widmete sie sich Kursen der Oracle University, später konzentriert sie sich auf verschiedene selbstentwickelte Praxisworkshops zu Spezialthemen. Ihre Spezialisierung liegt bei den Themen Administration, Monitoring, Security und Hochverfügbarkeit. Seit 2018 ist Cornelia in der DOAG Datenbank Community aktiv.

BRUNO CIRONE

Seit über 30 Jahren arbeitet er mit Produkten der Firma Oracle und kennt daher die Datenbank seit der Version 6. Seit 1989 ist er selbständiger EDV-Berater. Seine Themenschwerpunkte sind Tuning, Administration, Crash-Recovery, Datenbankumstellungen (Daten und/oder Systeme), Hochverfügbarkeitssysteme, Katastrophenpläne, Notfallszenarien. Bei verschiedenen Ministerien, Telekomunternehmen, Fahrzeugherstellern etc. hat er Security-Audits vorgenommen. Das Härten von Datenbanken (z.B. Logintrigger mit speziellen Regularien für den Zugriff) und deren besondere Einstellungen für das Auditing wurden von ihm konzipiert und umgesetzt. Einige Besonderheiten auf europäischer Ebene waren zum Beispiel verschiedene Zeitzonen, unterschiedliche Reaktionszeiten und unterschiedliche Gesetzgebungen. Weiterhin hat Bruno Cirone bei der Konzeption von Notfall- und Katastrophenszenarien in Abstimmung mit staatlichen Behörden mitgewirkt.

Härtung der Standard Edition 2

Marco Pachaly-Mischke, Robotron Datenbank-Software

Das Sicherheit im IT-Umfeld ein wichtiges Thema ist, sollte keine Überraschung mehr sein. Die Oracle-Datenbank bringt da eine Reihe von Funktionen mit, um Datenbanksysteme entsprechend zu härten. Der Großteil davon ist aber in verschiedenen kostenpflichtigen Optionen enthalten, die demnach die Enterprise Edition voraussetzen. Wie verhält es sich aber mit der Standard Edition 2 der Datenbank (Oracle DB SE2)? Ist diese Edition dann unsicher? Welche Möglichkeiten der Härtung gibt es damit? Der Artikel wird diese Fragen beleuchten.

Schaut man an den Punkt der einzigen Wahrheit, den Oracle Licensing Guide *[1]*, dann findet man in der Tabelle zu den Security Features bei der Standard Edition 2 (SE2) überall ein „N“ *(siehe Abbildung 1)*.

Daher könnte man meinen, dass der Artikel hier im Grunde zu Ende ist. Die Tabelle verschweigt aber auch einige grundlegende Dinge, die mit der SE2 sehr wohl zu realisieren sind. *Abbildung 2* zeigt, welche Nutzergruppen in einer Infrastruktur mit Datenbanken involviert sind und auf welche Ressourcen sie jeweils Zugriff haben. Darunter sind die Methoden aufgezeigt, die den jeweiligen Bereich vor un-

berechtigten Zugriffen schützen können. Das verdeutlicht noch einmal die Möglichkeiten der SE2.

Security Patches

Der erste und vielleicht auch der offensichtlichste Punkt betrifft das regelmäßige Einspielen von Sicherheitspatches, um möglichst alle bekannten Schwachstellen in einem Produkt zu beheben. Oracle veröffentlicht vierteljährlich Sicherheitsupdates für seine Produkte, also auch für die Oracle-Datenbank. Einen gültigen Supportvertrag vorausgesetzt, kann und sollte man diese Patches herunterladen und selbstverständlich auch die Datenbank SE2 damit auf dem aktuellen Stand halten. Das Gleiche gilt selbstverständlich auch für Patches für besondere Security Alerts, deren Behebung vor dem Erscheinen des nächsten regulären Updates erforderlich ist.

Die Strategie zur Einspielung der Patches sollte sein, exponierte Systeme möglichst zeitnah zu aktualisieren und weniger gefährdete Systeme zumindest halbjährlich mit den neuesten Updates zu versorgen. Die Ausfallzeit der Datenbank kann mittels eines simplen out-of-place-Patchings nochmals deutlich reduziert werden. Dabei wird ein neues Datenbank-Home erstellt, dass bereits die neuesten Patches enthält. Die zu patchende Datenbank wird dann gestoppt und anschließend aus dem neuen Home gestartet. Der fällige SQL-Teil des Patches kann dann während des Wiederanlaufs eingespielt werden.

Datenbankeinstellungen

Die Härtung der Datenbank fängt beim Datenbankserver an. Hier gilt es, einige Einstellungen vorzunehmen beziehungsweise zu überprüfen. In der SQLNET.ORA des Datenbank-Homes sollte der Parameter SQLNET.ALLOWED_LOGON_VERSION_SERVER auf „12a“ gestellt werden – der Standardwert ist „12“. Das erzwingt den bestmöglichen Algorithmus für die Authentifizierung, erfordert aber Clients, die mindestens Version 12.1 einsetzen. Außerdem bestimmt ein Parameter, welche Passworthashes in der Datenbank hinterlegt werden. Es ist also eine gute Idee, vorhandene Benutzer dahingehend zu überprüfen und gegebenenfalls deren Passwörter neu zu setzen, um veraltete Passworthashes zu entfernen. Diese Information befindet sich in der Spalte PASSWORD_VERSIONS in der View DBA_USERS. Ein weiteres Feature, dass über die SQLNET.ORA aktiviert werden kann, ist das Valid Node Checking. Damit lassen sich die Anmeldungen von entfernten Systemen an die Datenbank über eine Art Black- oder White-List steuern. Benutzt eine Anwendung zum Beispiel nur Applikationsserver, die sich dann zur Datenbank verbinden, können diese in einer Liste benannt werden. Die Datenbank akzeptiert dann nur Anmeldungen von diesen Systemen, alle anderen Systeme werden abgelehnt. Das Feature wird mit TCP.VALID_NODE_CHECKING=TRUE aktiviert. Die Liste der erlaubten oder ausgeschlossenen Systeme wird dann über TCP.INVITED_NODES oder TCP.EXCLUDED_NODES bestimmt. Die Liste der Invited Nodes hat dabei Vorrang.

Innerhalb der Datenbank gibt es einige weitere Einstellungen und Parameter. Hervorgehoben sei hier zuerst der Parameter SEC_PROTOCOL_ERROR_FURTHER_ACTION, der bestimmt, was nach dem Empfang fehlerhafter Pakete geschehen soll. Der Standardwert ist „(DROP, 3)“, der besagt, dass nach drei fehlerhaften Paketen die Verbindung gekappt wird. Der Client kann dann erneut beginnen, fehlerhafte Pakete zu senden und so einen Denial of Service verursachen. Daher sollte der Parameter auf „(DELAY, 10)“ gestellt werden, sodass die Antwort bei derartigen Verbindungen um 10 Sekunden verzögert wird. Ein weiterer wichtiger Parameter ist SEC_RETURN_SERVER_RELEASE_BANNER, der auf FALSE gestellt werden sollte. Andernfalls kann bereits vor der Authentifizierung die Versionsinformation der Datenbank abgerufen werden, was wiederum Rückschlüsse auf mögliche Angriffsvektoren ermöglicht.

Passworteinstellungen

Neben den Berechtigungen der Benutzer ist auch deren Authentifizierung ein wichtiger Punkt. Die SE2 hat hier die gleichen Funktionalitäten wie die EE. Zur Überprüfung von Passwörtern bringt Oracle bereits vorgefertigte Funktionen mit. Zum Prüfen der Passwortkomplexität kann mit dem Script „OH/rdbms/admin/catpvf.sql“ die Funktion „ora12c_verify_function“ erstellt werden. Dieses Skript bringt außerdem noch einige Hilfsfunktionen mit, so zum Beispiel die Funktion „ora_complexity_check“. Diese prüft einen übergebenen Text auf verschiedene Eigenschaften, die für Passwörter relevant sind *(siehe Listing 1)*. Sie ist daher ideal geeignet, wenn eigene Passwortregeln überprüft werden sollen.

Jetzt muss nur noch mit Hilfe von Profilen diese Passwortrichtlinie durchgesetzt werden. Das bedeutet, man passt nicht nur eigene Profile, sondern auch das DEFAULT-Profil entsprechend an, sodass wirklich alle Nutzer abgedeckt sind.

Berechtigungs- und Nutzer-Management

Ein weiterer Sicherheitsaspekt ist das Verwalten und Organisieren von Benutzern und deren Berechtigungen. Hier gilt das altbewährte Least Privilege Principle, also das Reduzieren der vergebenen Berechtigungen auf das absolut notwendige Minimum. Rollen wie „DBA“ oder alle möglichen „ANY“ Privilegien sind hier ein absolutes no-go. Wenn der Applikationshersteller nicht in der Lage ist, die tatsächlich benötigten Berechtigungen genau zu benennen, müssen diese selbst bestimmt werden. Da der Standard Edition 2 leider das Privilege Analysis fehlt, mit dem man die tatsächlich benötigten Berechtigungen recht einfach ermitteln könnte *[2]*, bleibt nur der Weg über Versuch und Irrtum. Man startet mit einem Basisset an Berechtigungen wie „CREATE SESSION“ und mit den Berechtigungen zum Anlegen verschiedener Datenbankobjekte wie Tabellen, Views und viele weitere. Dann ergänzt man so lange weitere Berechtigungen, bis die Anwendung am Ende fehlerfrei startet und läuft. Ein etwas mühsamer Prozess, der aber das Sicherheitslevel deutlich verbessern kann.

Eine weitere Möglichkeit, die Sicherheit zu erhöhen, ist die Trennung von Schemas und Benutzern. Eine Oracle-Datenbank unterscheidet eigentlich nicht zwischen Schema und Benutzer, sobald ein Benutzer Objekte besitzt, ist das das Schema. Der Eigentümer hat immer uneingeschränkte Rechte auf seine eigenen

Table 1-10 Security

Feature / Option / Pack	SE2	EE	EE-ES
Column-Level Encryption	N	Y	Y
Tablespace Encryption	N	Y	Y
Oracle Advanced Security	N	Y	Y
Oracle Database Vault	N	Y	Y
Oracle Label Security	N	Y	Y
Enterprise User Security	N	Y	Y
Centrally Managed Users	N	Y	Y
Fine-grained Auditing	N	Y	Y
Privilege Analysis	N	Y	Y
Real Application Security	N	Y	Y
Redaction	N	Y	Y
Transparent Sensitive Data Protection	N	Y	Y
Virtual Private Database	N	Y	Y
Keystore for Each Pluggable Database	N	N	Y

Abbildung 1: Security Features SE2, Oracle Licensing Guide (© Oracle, [1])

Objekte. Daher ist es sinnvoll, weitere Benutzer mit den benötigten Berechtigungen auf diese Objekte, beispielsweise INSERT, UPDATE, DELETE und SELECT, auszustatten und das direkte Anmelden als Schemanutzer zu verhindern. Das ist möglich, indem man den Benutzer einfach sperrt und dessen Passwort ablaufen lässt *(siehe Listing 2)*. Alternativ kann man auch direkt Benutzer ohne Möglichkeit der Authentifizierung anlegen, auch das ist in *Listing 2* zu sehen.

Das Release 23ai bringt hier eine hilfreiche Verbesserung mit, denn man kann nun Berechtigungen auf Schemaebene vergeben, ein Beispiel ist in *Listing 3* zu sehen. Bei älteren Releases bleibt nichts anderes übrig, als die Liste der Berechtigungen selbst zu pflegen und anzuwenden.

Wie kann man dann aber überhaupt Objekte im Schema anlegen, wenn man sich nicht direkt daran anmelden kann? ANY-Privilegien kommen dafür nicht in Frage. Die Antwort darauf sind Proxy User. Diese ermöglichen es einem Benutzer, sich mit seinen eigenen Anmeldedaten zu authentifizieren und dann als Schema-Eigentümer zu agieren. Wie ein Benutzer dazu befähigt wird, zeigt *Listing 4*.

Bei der Anmeldung an der Datenbank wird dann der Schemanutzer hinter dem Benutzernamen in eckigen Klammern angegeben. Damit lässt sich sehr gut nachvollziehen, wer wann was im Schema getan hat, sofern man personenbezogene Benutzer verwendet und die entsprechenden Aktivitäten auditiert. Dazu später noch etwas mehr.

Sollte eine Anwendung die Anmeldung direkt am Schema verlangen, weil beispielsweise die Objekte direkt ohne Angabe des Eigentümers referenziert werden, so kann die Einrichtung von Synonymen für all diese Objekte Abhilfe schaffen.

Native Netzwerkverschlüsselung

Auch die Verschlüsselung der Netzwerkkommunikation zwischen Client und Datenbank ist in der SE2 möglich. Die einfache Variante ist die native Verschlüsselung per SQL*Net. Zur Steuerung dienen die beiden SQLNET.ORA- Parameter ENCRYPTION_CLIENT und ENCRYPTION_SERVER. Beide können vier Werte annehmen: rejected, accepted, requested und

```
create or replace function ora_complexity_check
(password     varchar2,
 chars        integer := null,
 letter       integer := null,
 uppercase    integer := null,
 lowercase    integer := null,
 digit        integer := null,
 special      integer := null)
return boolean is …
```

Listing 1: Password Complexity Check Function

```
alter user <schema_owner> account lock password expire;
create user <schema_owner_2> no authentication;
```

Listing 2: Schemanutzer sperren

```
grant select any table on <schema_user> to <user>;
```

Listing 3: Berechtigungen auf Schemas vergeben (23ai)

```
alter user <schema_user> grant connect through <user>;
```

Listing 4: Anmeldung als Proxyuser einrichten

```
orapki wallet create -wallet $ORACLE_BASE/wallet -pwd <Passwort>
-auto_login_local
 orapki wallet add -wallet $ORACLE_BASE/wallet -pwd <Passwort> \
   -dn "CN=$(hostname)" -keysize 1024 -self_signed -validity 365
 orapki wallet export -wallet $ORACLE_BASE/wallet -pwd <Passwort> \
   -dn "CN=$(hostname)" -cert /tmp/$(hostname).cert
```

Listing 5: Wallet am Server erstellen

```
SSL_CLIENT_AUTHENTICATION = FALSE

WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY = /u01/app/oracle/wallet)
    )
  )
```

Listing 6: TCPS-Einstellungen für Listener und Datenbank

```
orapki wallet create -wallet "C:\Users\marco\wallet" -pwd <Passwort>
-auto_login_local
orapki wallet add -wallet "C:\Users\marco\wallet" -pwd <Passwort>
-trusted_cert -cert c:\temp\<servername>.cert
```

Listing 7: Wallet am Client erstellen

required. Entsprechend der Benennung steuern sie, ob es zur Verschlüsselung kommt oder nicht. *Abbildung 3* verdeutlicht das Zusammenspiel der Parameterwerte auf dem Client und dem Server und zeigt, wann eine verschlüsselte Verbindung zustande kommt.

Um immer eine verschlüsselte Verbindung zu erhalten, setzt man also lediglich auf dem Server den Wert ENCRYPTION_SERVER=required, wodurch die Verbindung immer verschlüsselt wird oder eben gar nicht erst zustande kommt.

Neben diesen beiden Parametern gibt es noch die beiden Parameter CRYPTO_CHECKSUM_CLIENT und CRYPTO_CHECKSUM_SERVER, die der Sicherstellung der Datenintegrität dienen und die die gleichen Werte annehmen können. Analog wird damit das Erstellen von Checksummen über Datenpakete aktiviert.

Netzwerkverschlüsselung mit TCPS

Für die Verschlüsselung des Netzwerkverkehrs per TCPS, also mittels TLS, ist etwas mehr Aufwand erforderlich. Es muss auf dem Datenbankserver ein Wallet mit einem Zertifikat erstellt und konfiguriert werden. Die Schritte zur Erstellung eines Wallets mit selbst signiertem Zertifikat und zum Export dieses Zertifikates sind in *Listing 5* zu sehen.

Ist das Wallet erstellt, muss sowohl dem Listener als auch der Datenbank dieses Wallet bekannt gemacht werden. Das geschieht für den Listener über die LISTENER.ORA und für die Datenbank über die SQLNET.ORA aus den entsprechenden Homes. Die nötigen Einstellungen zeigt *Listing 6*.

Dann benötigt der Listener noch einen entsprechenden Endpunkt für die TCPS-Verbindung. Der wird entweder über „srvctl modify listener“ oder über die LISTENER.ORA hinzugefügt.

Am Client muss ebenfalls ein Wallet erstellt und das am Server exportierte Zertifikat importiert werden. Die erforderlichen Schritte zeigt *Listing 7*.

Der Client braucht die analogen Einstellungen in der SQLNET.ORA wie der Server und zusätzlich noch den Parameter SSL_SERVER_DN_MATCH=ON. Außerdem braucht man einen entsprechenden TNSNAMES.ORA-Eintrag zum Herstellen der Verbindung *(siehe Listing 8)*.

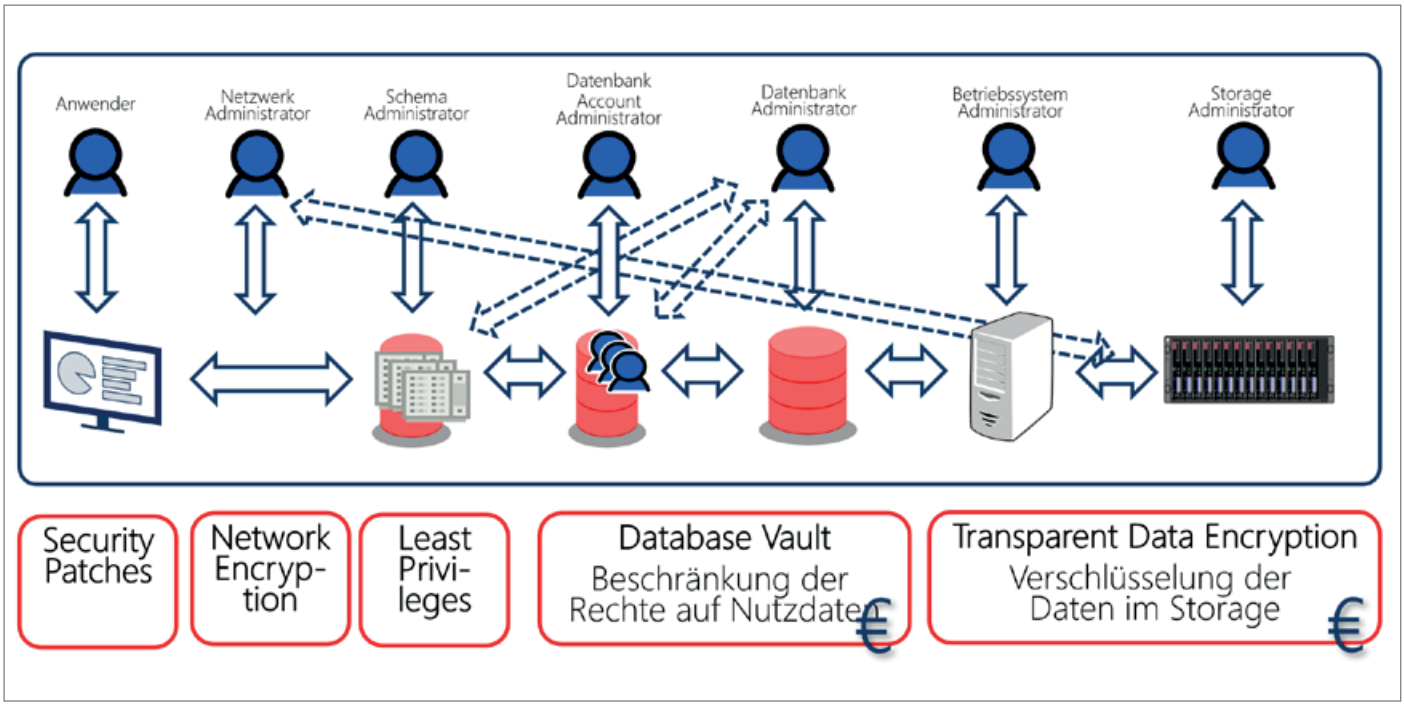


Abbildung 2: Nutzerrollen in einer Datenbank-Infrastruktur (© Robotron Datenbank-Software GmbH

ENCRYPTION_CLIENT	ENCRYPTION_SERVER	Verschlüsselung
REJECTED	REQUIRED	Keine Verbindung!
REQUIRED	REJECTED	Keine Verbindung!
REJECTED	REJECTED | ACCEPTED | REQUESTED	AUS
REJECTED | ACCEPTED | REQUESTED	REJECTED	AUS
ACCEPTED	ACCEPTED	AUS ← Default
REQUESTED	ACCEPTED | REQUESTED | REQUIRED	AN
REQUIRED	ACCEPTED | REQUESTED | REQUIRED	AN
ACCEPTED | REQUESTED | REQUIRED	REQUESTED	AN
ACCEPTED | REQUESTED | REQUIRED	REQUIRED	AN

*Abbildung 3: SQL*Net Native Encryption (© Robotron Datenbank-Software GmbH)*

```
datenbank_ssl =
  (DESCRIPTION=
    (ADDRESS= (PROTOCOL=TCPS) (HOST=<Servername>) (PORT=1522))
    (CONNECT_DATA=
      (SERVER=dedicated) (SERVICE_NAME=<Servicename>)
    )
    (SECURITY=(SSL_SERVER_CERT_DN="CN=<Servername>"))
  )
```

Listing 8: TNS-Names-Eintrag für TCPS

Man kann das Ganze auch noch erweitern und auf dem Client ein Zertifikat erstellen und dieses am Server importieren (mTLS), was den Parameter SSL_CLIENT_AUTHENTICATION obsolet macht. Allerdings können sich dann nur noch bekannte Clients gegen die Datenbank verbinden. So etwas macht also dann Sinn, wenn alle Clients bekannt sind, beispielsweise wenn es lediglich Applikationsserver gibt, die sich direkt mit der Datenbank verbinden.

Auditing

Das Ziel der Härtung ist es, ein System vor ungewollten Zugriffen möglichst gut zu schützen. Im Umkehrschluss müssen daher auch die kritischen Aktivitäten beobachtet und ausgewertet werden. Diesem Zweck dient das Auditing. Dieses Thema kann einen eigenen Artikel füllen, daher sei nur erwähnt, dass es keine funktionalen Einschränkungen der SE2 diesbezüglich gibt. Zu empfehlen ist hier lediglich der exklusive Einsatz des Unified Auditing. Standardmäßig laufen Datenbanken bis Version 19c noch im hybriden Modus, in dem sowohl das altherkömmliche Auditing sowie das Unified Auditing parallel funktionieren.

Fazit

Obwohl die SE2 mit einigen funktionalen Einschränkungen leben muss und sich einige Funktionalitäten auch nicht nachrüsten lassen, gibt es doch eine ganze Reihe an Möglichkeiten, den Betrieb einer SE2-Datenbank sicherer zu machen. Am Ende müssen die Kosten entscheiden, ob die besseren Möglichkeiten einer EE zur Härtung den nicht unerheblichen Mehrpreis auch wert sind.

Quellen

[1] Oracle (2024): Database Licensing Information User Manual, *https://docs.oracle.com/en/database/oracle/oracle-database/19/dblic/index.html*

[2] Robotron Datenbank Software GmbH (2019): Privilege Analysis, *https://www.robotron.de/unternehmen/aktuelles/blog/marco-pachaly-mischke/privilge-analysis*

Über den Autor

Marco Pachaly-Mischke ist bei der Robotron Datenbank-Software GmbH Teamleiter für Daten-Projekte und beschäftigt sich mit der Einrichtung und dem Betrieb von Oracle-Infrastrukturen On-Prem und in der Cloud. Er arbeitet seit über 25 Jahren an Themen rund um die Oracle-Datenbank. Die Schwerpunkte liegen bei den Themen Hochverfügbarkeit, Performanceoptimierung, Sicherheit und mehr. Sein Steckenpferd ist die Standard Edition 2, die in vielen Belangen oft unterschätzt wird. Er teilt seine Erfahrungen bei Vorträgen und schreibt Blogs und ist als ACE Pro in der Oracle Community vertreten.

Marco Pachaly-Mischke
marco.pachaly-mischke@robotron.de

Oracle 23ai SQL-Firewall – Datensicherheit in der Datenbank

Alexander Giesbrecht, Logicalis

Datensicherheit ist ein sensibles und wichtiges Thema in der heutigen Zeit. Ziel vieler Cyberangriffe sind die Daten von Unternehmen. Die Datenbank ist meist der zentrale Speicherort von Daten und mittels SQL werden die Daten von der Datenbank abgefragt. Daher lohnt sich der Blick auf Datensicherheit in der Datenbank. Oracle hat in die Datenbank 23ai eine SQL-Firewall integriert, um Angriffe mittels SQL-Injection, Abfragen von kompromittierten Accounts oder auch gefährliche Abfragen von privilegierten Benutzern zu erkennen und zu verhindern. Dieser Artikel befasst sich mit der grundlegenden Funktion und Steuerung der SQL-Firewall in der Oracle Database 23ai mittels PL/SQL-Package und grafisch über DataSafe.

Mit der Version Oracle 23ai kommt das Feature SQL-Firewall, um die Datensicherheit in der Datenbank zu erhöhen und Angriffe, wie SQL-Injections und die Ausnutzung von kompromittierten Accounts, zu erkennen und abwehren zu können *[1]*. Es ist in den Kernel der Datenbank integriert und benötigt somit keine zusätzliche Software-Installation. Die SQL-Firewall ist in der Enterprise Edition verfügbar und kann dort lizenziert und genutzt werden *[2]*. Es ist ebenfalls in den OCI-Datenbankangeboten wie Autonomous Database, ExaDB, ExaCC und BaseDB (EE-HighPerformance & EE-ExtremePerformance) inkludiert. Ausgenommen sind die BaseDB-Angebote für Standard Edition und Enterprise Edition in der Base- Performance-Ausprägung. Die SQL-Firewall kann auch in der Oracle 23ai Free genutzt werden.

Die Nutzung der SQL-Firewall muss jedoch lizenziert werden. Die Lizenz dafür ist in den beiden Lizenzpaketen „Oracle Database Vault“ und „Audit Vault and Database Firewall“ (AVDF) abgedeckt. In den OCI-Angeboten, in denen Database Vault inklusive ist, kann die SQL-Firewall somit genutzt werden.

Im Gegensatz zu „Audit Vault and Database Firewall“ (AVDF) ist die SQL-Firewall kein separates Produkt. AVDF ist eine zentrale Netzwerk-Lösung, welche sowohl Oracle als auch non-Oracle-Datenbanken überwachen kann. Dabei kann es auch mehrere Datenbanken gleichzeitig überwachen. Die SQL-Firewall überwacht nur die Datenbank, in der sie aktiviert ist. Dies vermeidet zusätzlichen Netzwerkload.

Funktionsweise

Wird die SQL-Firewall aktiviert, kann sie Anfragen an die Datenbank in zwei Kategorien überprüfen. Zum einen wird der Kontext der Ausführung überprüft. Dies beinhaltet Informationen über die Quelle der Anfrage wie IP-Adresse, OS-Benutzername und Programm. Zum anderen wird das SQL-Statement überprüft. Anhand der Policy für ein Schema oder für Benutzer in der Datenbank wird festgelegt, welcher Kontext und/oder welche SQLs erlaubt sein sollen. Wenn die Kriterien der Anfrage erfüllt sind, wird das SQL ausgeführt *(siehe Abbildung 1)*.

Beinhaltet die Anfrage Kriterien, die nicht erlaubt sind, wird die Anfrage entweder ausgeführt und in ein Violation Log geschrieben, oder die Anfrage wird geblockt und ebenfalls in das Violation Log geschrieben. Ob die nicht erwünschten Anfragen zugelassen oder geblockt werden sollen, wird bei der Erstellung der Policy eingestellt, kann aber jederzeit geändert werden. Das Violation Log kann in der View `DBA_SQL_FIREWALL_VIOLATIONS` eingesehen werden. Wird die Anfrage geblockt, erscheint die Fehlermeldung „ORA-47605: SQL Firewall violation“. Zusätzlich können die SQL-Firewall-Aktivitäten auch in das Audit Trail der Datenbank geschrieben werden.

Um eine Firewall Policy zu erstellen, sind folgende Schritte notwendig, die später noch genauer erläutert werden:

1. Aktivieren der SQL-Firewall
2. Erstellen einer Capture für einen User
3. Starten der Capture
4. Während der Capture den später erlaubten SQL-Workload ausführen
5. Stoppen der Capture und Erstellen einer Allow List aus der Capture
6. Aktivieren der Allow List

Das „Capture“, also das Erfassen des gewöhnlichen Workloads auf der Datenbank, muss nicht für jede Datenbank neu gemacht werden. Die Allow Lists können exportiert und in andere Datenbanken

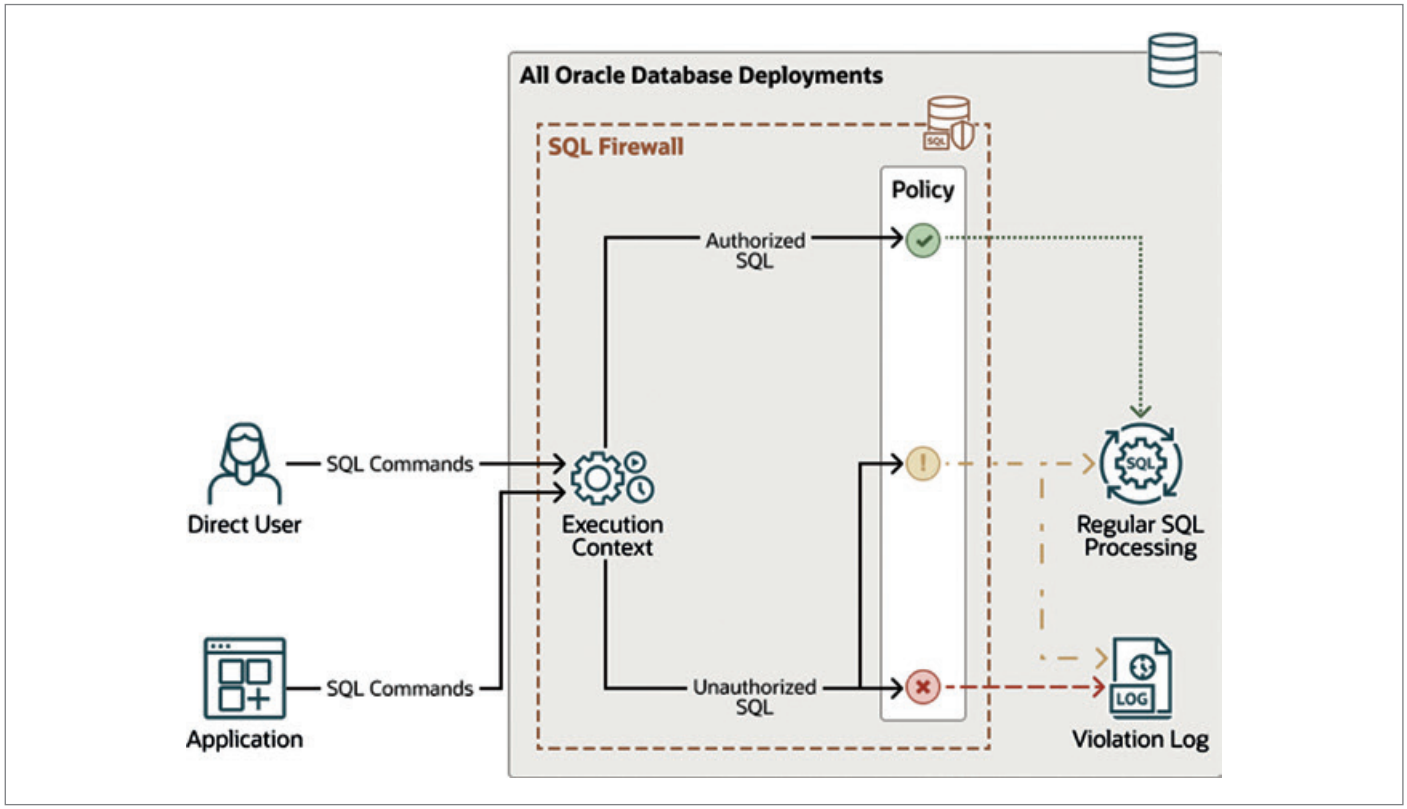


Abbildung 1: Funktionsweise der SQL-Firewall [3] (Quelle: Alexander Giesbrecht)

```
BEGIN
  DBMS_SQL_FIREWALL.CREATE_CAPTURE (
    username => 'APP_USER',
    top_level_only => TRUE,
    start_capture => FALSE );
END;
/
```

Listing 1: Anlegen einer Capture

```
DBA_SQL_FIREWALL_ALLOW_LISTS      → Liste der Allow lists
DBA_SQL_FIREWALL_ALLOWED_SQL      → Erlaubte SQL Pattern
DBA_SQL_FIREWALL_ALLOWED_OS_PROG  → Erlaubte Programme
DBA_SQL_FIREWALL_ALLOWED_OS_USER  → Erlaubte OS-User
DBA_SQL_FIREWALL_ALLOWED_IP_ADDR  → Erlaubte IP-Adresse/-
Adressbereich
```

Listing 2: Views für die Firewall Policy

```
BEGIN
  DBMS_SQL_FIREWALL.ENABLE_ALLOW_LIST (
    username => 'APP_USER',
    enforce => DBMS_SQL_FIREWALL.ENFORCE_SQL,
    block => TRUE
    );
END;
/
```

Listing 3: Aktivieren der Allow List

```
SQL> select * from user_tables;
select * from user_tables
              *
ERROR at line 1:
ORA-47605: SQL Firewall violation
Help: https://docs.oracle.com/error-help/db/ora-47605/
```

Listing 4: Fehlermeldung bei aktivierter Allow List und block=TRUE

```
execute bestellsum('123456');
select count(*) from bestellungen where kundennr='123456';
select kundennr, artikel, preis, menge from bestellungen where
kundennr='123456';
```

Listing 5: SQLs während der Aufzeichnung

```
@datasafe_privileges.sql DATASAFE_ADMIN GRANT ALL -VERBOSE
```

Listing 6: datasafe_privileges.sql-Aufruf

importiert werden. Bei einer Umgebung mit Testdatenbanken können somit Daten aus einer Testphase in der Testumgebung in die Produktionsdatenbank übernommen werden. Auch können dann Daten der Policy wie die erlaubten IP-Adressbereiche angepasst und geändert werden.

Die SQL-Firewall kann auf zwei Wegen administriert werden. Zum einen mittels PL/SQL-Package oder in der OCI mit Data-Safe auf grafischer Ebene.

Nutzung als PL/SQL-Package

Das Package DBMS_SQL_FIREWALL beinhaltet die Funktionen für die Administration der SQL-Firewall *[4]*. Um die SQL-Firewall zu aktivieren, wird der Befehl `EXEC DBMS_SQL_FIREWALL.ENABLE;` ausgeführt. Damit ist die SQL- Firewall aktiviert und kann konfiguriert werden. Die SQL-Firewall lässt sich sowohl auf PDB- oder auf CDB-Level aktivieren, je nachdem welches Schema überwacht werden soll. Das Erstellen einer Capture erfolgt wie in *Listing 1*.

Dabei wird der Name des zu überwachenden Datenbankbenutzers angegeben. Zusätzlich kann mit top_level_only angegeben werden, ob auch SQLs innerhalb von Functions aufgezeichnet werden sollen oder nur der Aufruf von Functions. Standardmäßig wird die Aufzeichnung gleich gestartet, was als optionaler Parameter jedoch verhindert werden kann. Soll das Capturing später gestartet werden, kann es mit `DBMS_SQL_FIREWALL.START_CAPTURE('APP_USER')` gestartet werden. Ab dann werden für alle Verbindungen des Users APP_USER der Kontext und die ausgeführten SQLs aufgezeichnet, bis die Aufzeichnung mit `DBMS_SQL_FIREWALL.STOP_CAPTURE('APP_USER')` wieder gestoppt wird. Während der Aufzeichnung kann der umfangreiche Applikationstest laufen, bei dem möglichst alle später relevanten Use Cases und Workloads ausgeführt werden sollten. Die aufgezeichneten Daten können während und nach der Aufzeichnung in den Views `DBA_SQL_FIREWALL_CAPTURE_LOGS`, für die ausgeführten SQLs, und `DBA_SQL_FIREWALL_SESSION_LOGS`, für die Kontextinformationen der Verbindungen, eingesehen werden. Nach Abschluss und

```
SQL> select username, status, top_level_only from dba_sql_firewall_allow_lists where username='APP';

USERNAME STATUS   TOP_LEVEL_ONLY
-------- -------- --------------
APP      ENABLED  N

SQL>  select username, current_user, top_level, sql_text, accessed_objects
from dba_sql_firewall_allowed_sql
where username='APP' order by sql_text, current_user, top_level;
  2    3
USERNAME CURRENT_USER TOP_LEVEL SQL_TEXT                                                                     ACCESSED_OBJECTS
-------- ------------ --------- ---------------------------------------------------------------------------- ------------------------
APP      APP          Y         BEGIN BESTELLSUM (?); END;                                                   "APP"."BESTELLSUM"
APP      APP          Y         SELECT COUNT (*) FROM BESTELLUNGEN WHERE KUNDENNR=:"SYS_B_0"                 "APP"."BESTELLUNGEN"
APP      APP          Y         SELECT DECODE (USER,:"SYS_B_0",XS_SYS_CONTEXT (:"SYS_B_1",:"SYS_B_2"),USER) FROM SYS.DUAL "SYS"."DUAL"
APP      APP          Y         SELECT KUNDENNR,ARTIKEL,PREIS,MENGE FROM BESTELLUNGEN WHERE KUNDENNR=:"SYS_B_0" "APP"."BESTELLUNGEN"
APP      APP          N         SELECT SUM (PREIS) FROM BESTELLUNGEN WHERE KUNDENNR=:"SYS_B_0"               "APP"."BESTELLUNGEN"
```

Abbildung 2: Allow List und Allowed SQL (Quelle: Alexander Giesbrecht)

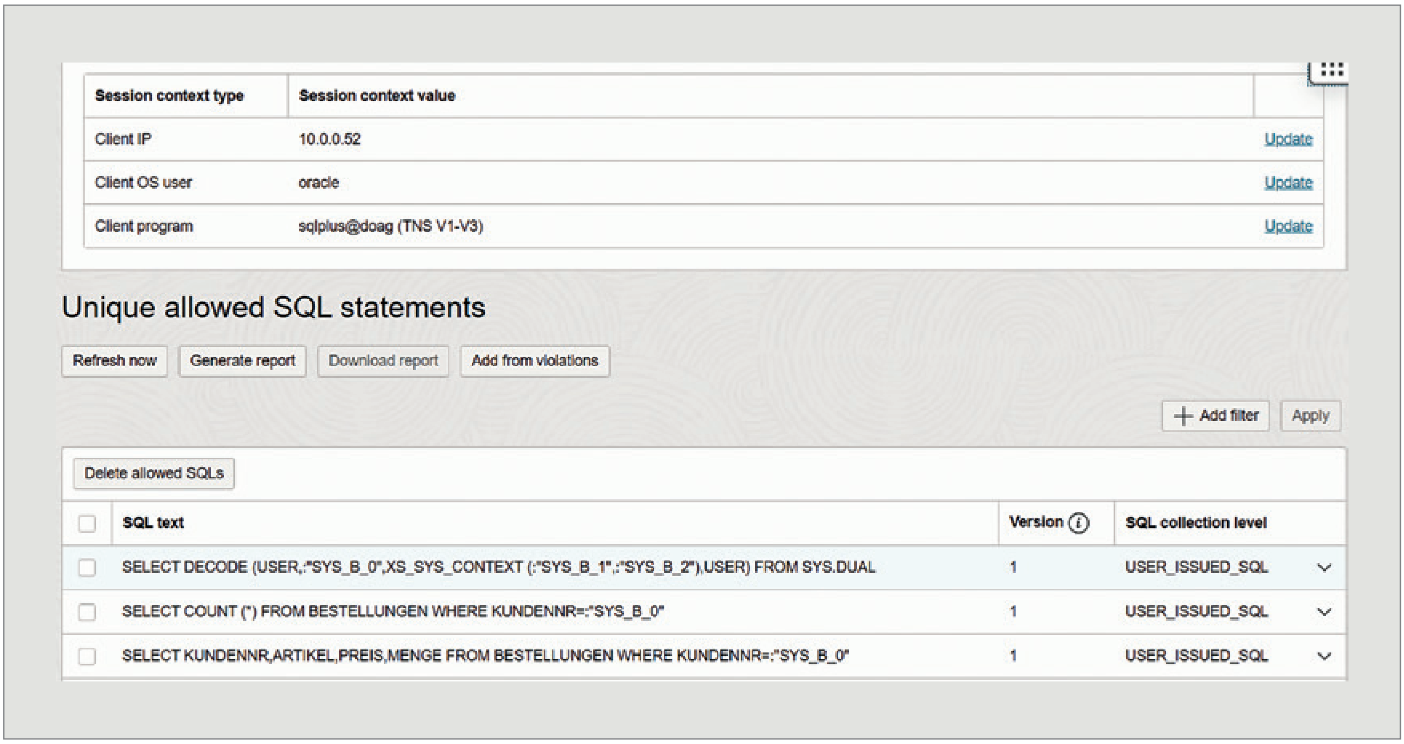


Abbildung 3: Allow List in DataSafe: Kontext und SQLs (Quelle: Alexander Giesbrecht)

Überprüfung der gesammelten Daten wird aus der Capture eine Allow List mit `DBMS_SQL_FIREWALL.GENERATE_ALLOW_LIST('APP_USER')` erstellt. Die Allow List ist die Firewall Policy, anhand welcher spätere Verbindungen und SQLs überprüft werden. In den Views aus *Listing 2* sind die Details der Allow List einsehbar.

Nach dem Erstellen der Allow List werden die Regeln noch nicht angewendet. Um die Allow List zu aktivieren, wird die Prozedur `ENABLE_ALLOW_LIST` *(siehe Listing 3)* ausgeführt. Für den Parameter „enforce“ gibt es drei Möglichkeiten. Zum einen gibt es wie in *Listing 3* die Möglichkeit, nur die SQLs zu überprüfen. Ebenfalls kann mit `ENFORCE_CONTEXT` nur der Kontext der Verbindung überwacht werden. Sollen jedoch Kontext und SQL überwacht werden, kann dies mittels `ENFORCE_ALL` eingestellt werden.

Standardmäßig blockiert die Firewall beim Aktivieren der Allow List nicht unerwünschte Verbindungen, sondern protokolliert diese nur ins Violation Log. Sollen diese jedoch blockiert und protokolliert werden, kann beim Aktivieren der Allow List der Parameter `block` auf `TRUE` gesetzt werden, oder im Nachgang der Parameter mit der Prozedur `UPDATE_ALLOW_LIST_ENFORCEMENT` geändert werden. Die Allow List kann auch dann modifiziert werden, wenn diese aktiv ist. Dafür gibt es mehre Prozeduren, die es ermöglichen, Kontext oder SQLs zu entfernen oder hinzuzufügen. Ebenso kann jederzeit wieder eine Aufzeichnung gestartet werden und die Ergebnisse der Allow List können hinzugefügt werden.

Wird ein SQL ausgeführt, welches nicht in der Allow List enthalten ist, kommt es zu einer Fehlermeldung *(siehe Listing 4)*.

Praktischer Test

Um die Funktion der SQL-Firewall zu testen, wurden ein paar einfache Tests gemacht. Dabei wurde in einem Schema APP eine Tabelle „Bestellungen“ mit fiktiven Beispieldaten und einer Prozedur „bestellsum“ angelegt. Die Prozedur ermittelt die Summe aller Bestellungen zu einer mitgegebenen Kundennummer. Während der Capture wurden die SQLs aus *Listing 5* ausgeführt.

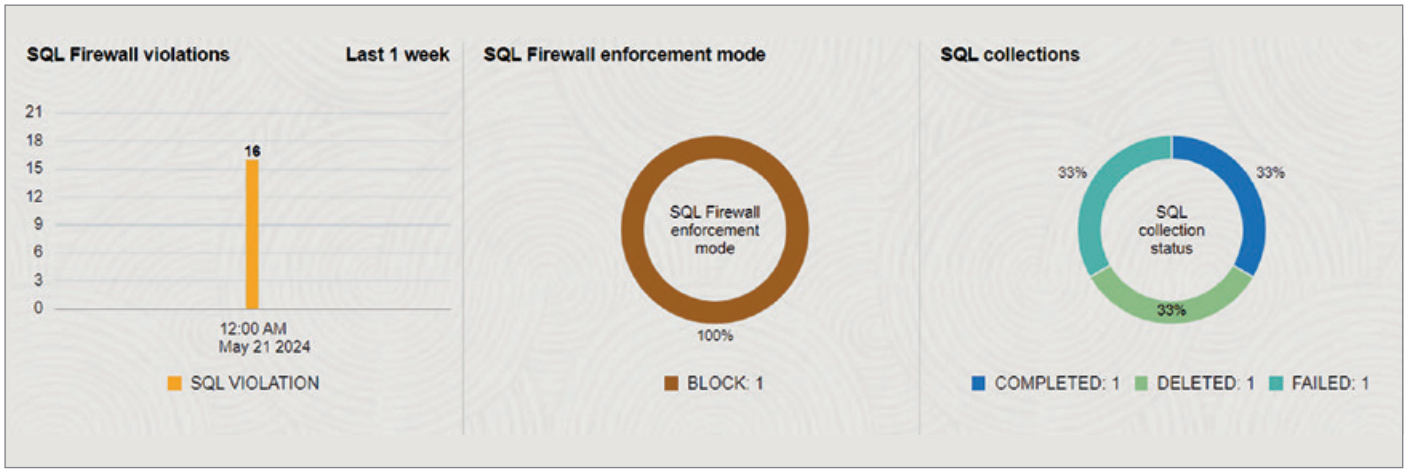


Abbildung 4: SQL Firewall Dashboad in DataSafe mit Violations (Quelle: Alexander Giesbrecht)

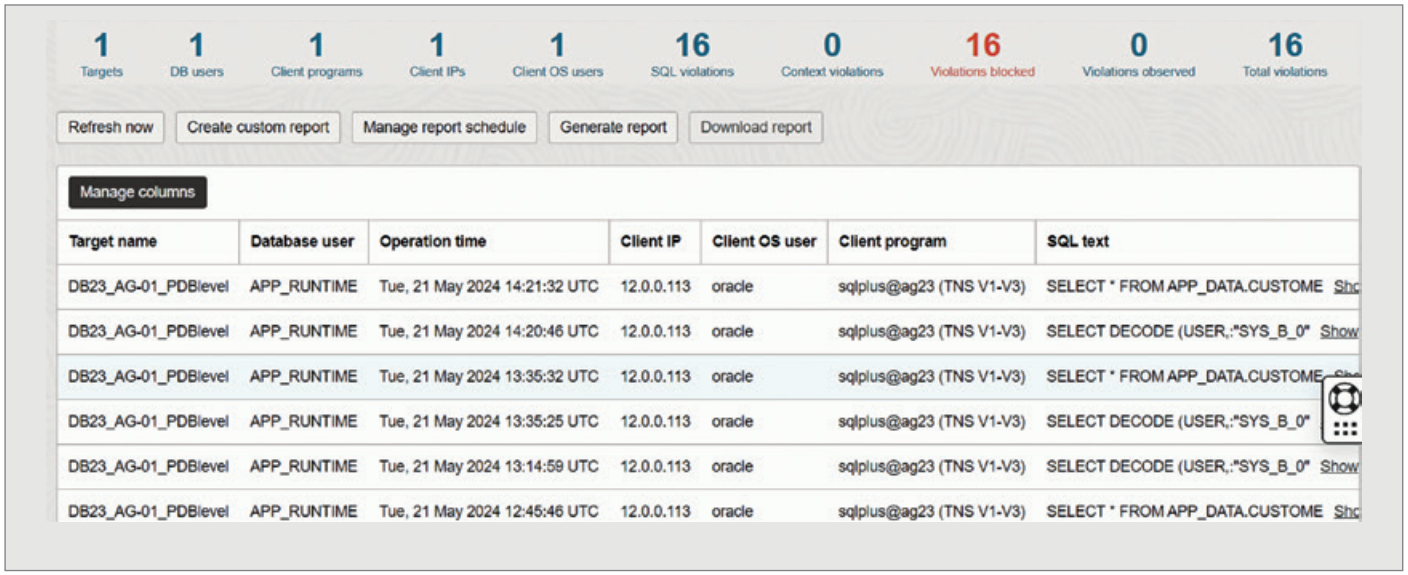


Abbildung 5: Violation Report in DataSafe (Quelle: Alexander Giesbrecht)

Die Aufzeichnung und die Allow List wurden mit *top_level_only=false* angelegt, was somit auch die Inhalte der Prozedur aufgezeichnet hat. *Abbildung 2* zeigt die SQLs der Allow List.

Bei den Tests wurde festgestellt, dass jegliche Veränderung des SQLs, wie zum Beispiel das Hinzufügen/Entfernen von Spalten oder WHERE-Bedingungen, zur Firewall Violation führt. Auch die Veränderung der Reihenfolgen der abgefragten Spalten oder das Ersetzen eines Wertes mit dem Spaltennamen wird von der SQL-Firewall abgefangen. Die Werte, wie in diesem Fall die Kundennummer, werden als Platzhalter gespeichert, was die Verwendung mit Eingabemasken, woraus eine SQL generiert wird, möglich macht. SQL-Injections können somit nicht mehr durchgeführt werden, weil das Muster der SQL-Abfrage dadurch verändert wäre und nicht mehr der Allow List entspräche.

OCI DataSafe

Die SQL-Firewall lässt sich in der OCI auch über DataSafe grafisch steuern. Dafür sind ein paar Vorarbeiten notwendig. Zum einen benötigt DataSafe einen eigenen Benutzer in der Datenbank. Dieser kann in der CDB oder PDB angelegt werden und muss dann mit notwendigen Berechtigungen versehen werden. In der Autonomous Database gibt es dafür einen vorgefertigten Benutzer DS$ADMIN mit allen notwendigen Berechtigungen. Um die Berechtigungen zu vergeben, stellt Oracle das Skript *datasafe_privileges.sql* in der OCI Console zur Verfügung, mit welchem der DataSafe-Benutzer alle notwendigen Berechtigungen erhält. Der Aufruf kann wie in *Listing 6* aussehen.

Im Anschluss muss das DataSafe Target angelegt werden. Dies kann eine PDB oder die CDB sein. Dafür gibt es mehrere Wege, am einfachsten ist jedoch der Register-Wizard in der OCI, welcher auch den notwendigen Private Endpoint anlegt und das Target registriert. Im Anschluss kann die Datenbank im DataSafe verwendet werden, der auch die SQL-Firewall beinhaltet. Der Ablauf für die Konfiguration ist identisch mit der Nutzung mittels PL/SQL-Package, jedoch können die Schritte zum Aktivieren der Firewall, das Anlegen und Starten einer Collection und das Erstellen und Bearbeiten der Allow List nun grafisch in der OCI Console vorgenommen werden *(siehe Abbildung 3)*.

Zusätzlich können die SQL- und Kontext-Violations auch grafisch angezeigt werden *(siehe Abbildung 4)*. Dies kann zusätzlich mit Notifications versehen werden, sodass bei Violations auch Benachrichtigungen für die Administratoren oder Entwickler erzeugt werden *(siehe Abbildung 5)*.

Fazit

Die SQL-Firewall ermöglicht es, die Datensicherheit der Datenbank zusätzlich zu erhöhen, indem unbekannte SQL-Muster bei SQL-Injections oder die Ausnutzung von kompromittierten Accounts über SQL und Verbindungskontext erkannt und blockiert werden können. Das Monitoring dabei ist ebenfalls möglich und wenn DataSafe verwendet wird, kann der Zuständige auch über Violations benachrichtigt werden. Jedoch ist der Einsatz der SQL-Firewall auch mit Aufwand verbunden. Zum einen muss das Testen der Applikation und des regulären Workloads auf der Datenbank möglichst detailliert und vollständig während der Aufzeichnung ausgeführt werden. Zwar kann die Allow List jederzeit angepasst werden, jedoch kann es dort zu unerwünschten Fehlern kommen, wenn in der Produktion Anwendungsfälle nicht vollständig in der Allow List enthalten sind. Außerdem muss auch immer in den Releases der Anwendung berücksichtigt werden, dass neue SQLs und vieles mehr dazu kommen können. Insgesamt bietet die SQL-Firewall einen zusätzlichen Schutz für Datenbanken mit kritischen und sensiblen Daten.

Quellen

[1] *https://blogs.oracle.com/cloudsecurity/post/sql-firewall-now-built-into-oracle-database-23c*

[2] *https://www.oracle.com/a/ocom/docs/security/oracle-SQL Firewall-faq.pdf*

[3] *https://docs.oracle.com/en/database/oracle/oracle-database/23/dbseg/using-oracle-SQL Firewall.html#GUID-B268CC0A-4FE5-4A50-9C20-FABC99B5C4AD*

[4] *https://docs.oracle.com/en/database/oracle/oracle-database/23/arpls/dbms_sql_firewall.html#GUID-4A41615C-D3D5-4386-A299-5C3D08350E8C*

Über den Autor

Alexander Giesbrecht ist Solution Engineer bei der Logicalis GmbH für Oracle-Datenbanken und -Architekturen. Während des dualen Studiums kam er mit Oracle- Datenbankadministration in Berührung und hat somit seit über 8 Jahren Erfahrungen in verschiedenen Oracle-Architekturen und -Bereichen gesammelt. Aktuell beschäftigt er sich unter anderem mit Migrationen nach Multitenant und in die OCI.

Alexander Giesbrecht
alexander.giesbrecht@logicalis.de

Oracle Cloud Infrastructure Security Basics

Sven Illert, Robotron Datenbank-Software

Für viele Unternehmen wird die Verwaltung und Analyse ihrer besonders wertvollen Daten immer herausfordernder. Die Oracle Cloud Infrastructure bietet eine ideale Umgebung, um diesen wachsenden Anforderungen begegnen zu können. Doch wie sieht es mit der Sicherheit und dem Zugriffsschutz aus, wenn man sein höchstes Gut einer fremden Infrastruktur anvertraut?

Die Oracle Cloud Infrastructure (OCI) erfreut sich nicht zuletzt durch die Always-Free-Angebote und die Einsatzmöglichkeiten für verschiedenste Institutionen in unterschiedlichen Realms durchaus einiger Beliebtheit. Damit die Freude über die Möglichkeiten der Skalierung und einfachen Nutzung verschieden komplexer Services nicht allzu schnell getrübt wird, gilt es jedoch einiges bezüglich Sicherheit zu beachten. Nicht zuletzt, weil die Infrastruktur nicht im eigenen Haus platziert ist und damit nicht zwangsläufig der Kontrolle der häuslichen Security-Experten untersteht, können vor allem für Anwendungsbetreuer neue herausfordernde Aufgaben in diesem Bereich auftreten.

Einführung

Zu Beginn gilt es erst einmal festzuhalten, welche verschiedenen Security-Themen es für Cloud-Nutzer grundsätzlich zu beachten gilt:

- Zugriffskontrolle
- Datenschutz
- Authentifizierungssicherheit
- Monitoring

Diese Liste erhebt keinen Anspruch auf Vollständigkeit und soll als Leitwerk nur die absoluten Basisbedürfnisse für das Thema Sicherheit abdecken. Beim Thema Zugriffskontrolle geht es darum, den Zugriff auf die Ressourcen in der Cloud zu regeln. Der Datenschutz betrifft die verschiedenen Möglichkeiten, die es gibt, um die Daten vor unberechtigtem Zugriff durch Dritte zu schützen, Stichwort: Verschlüsselung. Die Schlüssel und Geheimnisse zur Authentifizierung bei verschiedenen Services sollten in einem sicheren Bereich abgelegt werden können und auch dafür gibt es Services und Dinge zu beachten. Zum Schluss und dennoch nicht unwichtig, muss die Implementierung dieser Maßnahmen überwacht werden, damit diese nicht durch die Hintertür ausgenutzt werden können.

Zugriffskontrolle

Bevor ein Zugriff auf Daten oder Dienste gewährt wird, gibt es auf verschiedenen Ebenen entsprechende Kontrollmöglichkeiten. Zuallererst sollte man dabei an die Weboberfläche der OCI denken, die in der Regel initialer Zugriffspunkt für eine bestehende oder zukünftige Cloud-Infrastruktur ist. Der Zugang darüber wird über das von Oracle bereitgestellte Identity und Access Management (IAM) verwaltet und bietet zahlreiche Möglichkeiten, die von anderen Verzeichnisdiensten bekannt sind. Dazu gehören in der Basisausbaustufe auch ein Benutzer- und Gruppenmanagement. Wenn es gefordert ist, lässt sich dieses mit anderen Diensten wie Entra ID von Microsoft verknüpfen, um so unternehmensweit gültige Gruppenzugehörigkeit und Single-Sign-On (SSO) umzusetzen.

Sind Nutzer entsprechenden Gruppen und Domains zugeordnet, können mittels entsprechender Richtlinien (Policies) Zugriffe auf verschiedene Ressourcen in der OCI verwaltet werden. So lässt sich beispielsweise mit der Policy aus *Listing 1* festlegen, dass Nutzer der Gruppe *compute-admin* sogenannte Compute-Instanzen (Virtuelle Maschinen) im Cloud Account verwalten können. Um zu verhindern, dass eine produktive Instanz aus Versehen gelöscht wird, verhindert die Policy außerdem das Löschen von Ressourcen, die das Tag *production* gesetzt haben.

Sind die Berechtigungen verteilt, ist es wichtig, dass nur jene Personen Zugriff auf den jeweils berechtigten Account in der OCI bekommen, die auch dafür autorisiert sind. Oracle sieht seit einiger Zeit vor, dass Nutzer die Multi-Faktor-Authentifizierung aktiviert haben. Es ist zu empfehlen, dass dies auch für ältere Tenancies im Nachhinein aktiviert wird. Nutzen lassen sich dafür verschiedene Möglichkeiten wie die von Oracle in den gängigen App-Stores bereitgestellte Authenticator App mit Push-Benachrichtigung oder auch FIDO-Token beziehungsweise FIDO2-Passkeys.

In der Regel ist es jedoch so, dass die Ressourcen in der OCI nicht als Insel betrieben werden und der Zugriff nur über die Konsole im Browser stattfindet. Stattdessen haben die Ressourcen üblicherweise Schnittstellen zu anderen Systemen im eigenen Unternehmen oder über verschiedene Netzanbindungen zu

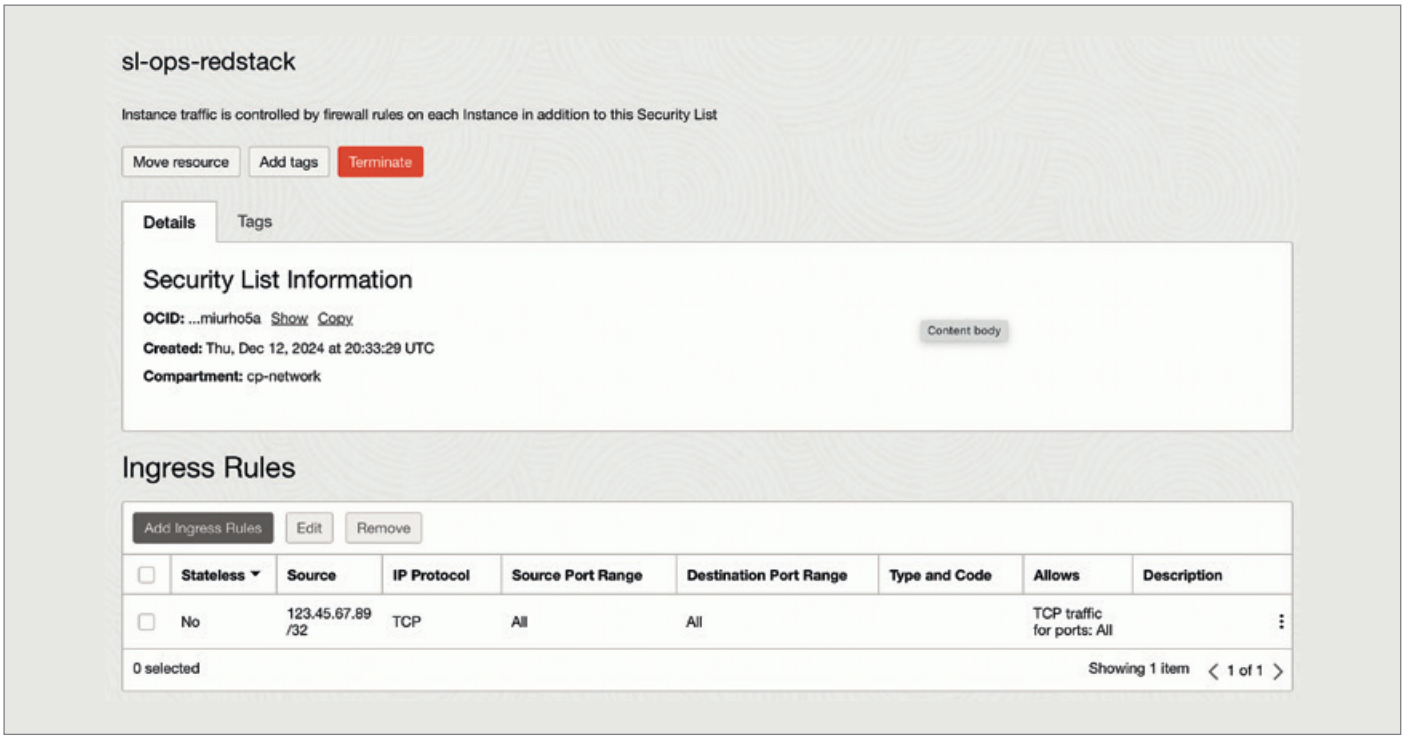


Abbildung 1: Beispiel für eine Security List, die eingehende SSH-Kommunikation nur von einer bestimmten IP-Adresse erlaubt. Dadurch, dass die Regel als Stateful angelegt wurde, ist die Öffnung des Rückkanals zum Kommunikationspartner nicht notwendig. (Quelle: Sven Illert)

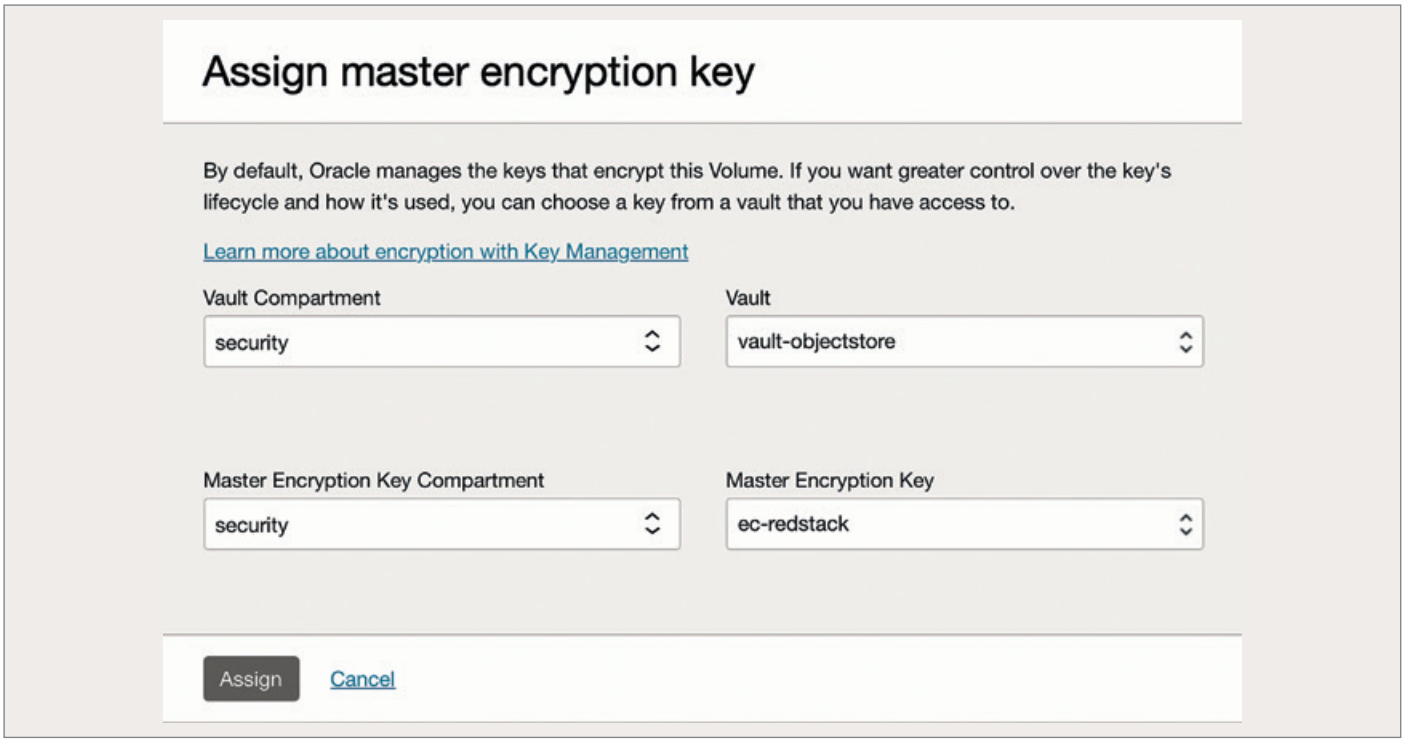


Abbildung 2: Änderung des Schlüssels eines Speicherbereichs in der OCI von einem durch Oracle verwalteten Schlüssel zu einem selbstverwalteten. Dieser kann auch durch eigene Hardware generiert und in das OCI Vault importiert worden sein. (Quelle: Sven Illert)

```
Allow group compute-admin to manage instance-family in tenancy
where all {request.permission != 'INSTANCE_DELETE',
target.resource.compartment.tag.operations.environment != 'production'}
```

Listing 1: Policy zum Verwalten von Compute-Instanzen

allen möglichen Ressourcen in der Welt. Damit auch hier nur berechtigte Zugriffe stattfinden, gibt es in der Oracle Cloud Möglichkeiten, dies zu steuern. Dort, wo man sonst im Keller Menschen mit den nötigen Kenntnissen auf den Switches und Firewalls benötigt, gibt es in der Cloud einfachere Möglichkeiten. Doch auch wenn die Oberfläche dazu einlädt, hier allzu schnell Berechtigungen zu erteilen, sollte man hier vorsichtig agieren und nur Zugriffe von und auf IP-Adressen sowie TCP/UDP- Ports ermöglichen, die dafür vorgesehen sind.

Die erste Ebene, auf der das eingeschränkt werden kann, ist das Routing. Hierbei wird festgelegt, in welche Richtung und von woher überhaupt kommuniziert werden kann. Möchte man seine Daten nicht über öffentliche Kanäle kommunizieren, so geht kein Weg an einem Virtual Private Circuit über Fast Connect vorbei, welches auch die schnellsten Zugriffe und Übertragungsgeschwindigkeiten bietet. Die etwas weniger leistungsfähige Variante wäre ein Virtual Private Network mit IPSEC über das Internet. Dies ist State of the Art für niedrigere Geschwindigkeits- und Latenzanforderungen. Dafür kann der OCI-eigene Service verwendet werden, genauso wie ein eigener Router beziehungsweise eine Firewall in einer eigenen Compute-Instanz. Mit letzterem ließen sich auch weniger komplizierte VPN-Lösungen wie WireGuard umsetzen.

Ist die Umgebung weniger komplex oder ist es für die Administration der Dienste ausreichend auf nur einen Host zugreifen zu müssen, der auch als Jump-Host genutzt werden könnte, so bietet sich zusätzlich die Möglichkeit, diesen Host mit einer öffentlich erreichbaren IP-Adresse anzubinden. Damit darauf aber nur berechtigter Zugriff stattfindet, lässt sich das Routing auf den Internet-Breakout des Unternehmens oder die öffentliche IP-Adresse des heimischen Netzanschlusses beschränken. Dies lässt sich in den Routing-Regeln des jeweiligen Subnetzes, indem sich der Jump-Host befindet, derart bestimmen, dass statt des Routings ins komplette Internet mit dem CIDR-Block 0.0.0.0/0 eben die IP-Adresse als CIDR-Block der Art 123.45.67.89/32 genutzt wird.

Doch hat man im gleichen Subnetz Ressourcen, die auch von überall erreichbar sein sollen, ist dies nicht mehr so einfach umsetzbar. Hier muss man den Zugriff mit Firewall-Regeln steuern und dies kann prinzipiell über zwei grundlegende Technologien erreicht werden. Die eine sind so genannte Security-Lists, die für Subnetze einzeln eingerichtet werden, und deren Regeln auch für das komplette Subnetz gelten *(siehe Abbildung 1)*. Eine andere Möglichkeit sind sogenannte Network Security Groups (NSG), die für eine entsprechende Zugriffskontrolle auf einzelne Ressourcen wie Compute-Instanzen, Load Balancer oder Datenbank-Services angewendet werden können. Die letztere Variante erlaubt die granulare Steuerung, ob eine Ressource auf einem oder mehreren bestimmten Ports von einer bestimmten IP-Adresse, einem -Adressbereich oder anderer NSGs erreichbar sein soll.

Die grundlegende Firewall-Funktionalität sollte natürlich auch genutzt werden, um die Kommunikation zwischen den einzelnen OCI-Ressourcen zu steuern, denn ein Application-Server sollte nicht per se über SSH mit der Datenbank kommunizieren müssen. Neben der bekannten Konfiguration der Kommunikation auf IP-Ebene und entsprechenden Protokoll-Ports gibt es seit dem Oktober 2024 auch das so genannte Zero Trust Packet Routing, bei welchem man mit einer eigenen Policy-Sprache auf höher Ebene explizit definieren kann, welche Ressourcen untereinander kommunizieren dürfen, ohne dass man dies über zu offene Firewall-Regeln beeinflussen könnte. Generell sollte man bei der Freigabe von Kommunikationsschnittstellen in der Cloud nach dem Verfahren vorgehen, wie es das vorgenannte Feature auch im Namen trägt, nämlich Zero Trust.

Deshalb ist es sinnvoll, sich immer folgende Fragen zu stellen:

- Muss eine Ressource über das öffentliche Internet erreichbar sein?
- Ist es notwendig kaum benutzte Ports im ganzen Subnetz freizuschalten?
- Müssen administrative Zugriffe von überall erlaubt werden?
- Müssen Zugriffe rund um die Uhr freigeschaltet sein?
- Benötigt beispielsweise ein Anwendungsbetreuer Berechtigungen zum Beispiel für die Netzwerkressourcen?

Diese Fragen sollten immer dazu führen, dass die Berechtigungen und Freischaltungen nur dorthin vergeben werden, wo sie auch notwendig sind. Damit kann man ebenso die Last für weitere Dienste wie ein Intrusion-Prevention-System, falls notwendig, deutlich reduzieren.

Datenschutz und Verschlüsselung

Hat man die Zugriffe auf Ressourcen und Kommunikationsschnittstellen entsprechend abgesichert, so gilt es, die dort kommunizierten und vorliegenden Daten vor dem Zugriff durch Dritte zu schützen. Dabei gilt es, eine gesunde Paranoia nicht unbedingt gegenüber dem Betreiber der Plattform Oracle zu haben, sondern gegenüber allem, was man nicht selbst auf eine Platine gelötet hat. Denn Angriffsvektoren gibt es bei den hochkomplexen Systemen heutzutage durchaus viele. Dem geschuldet und natürlich auch durch einige Vorgaben von Oracle selbst, werden und sollten Daten in einer wie auch immer gearteten Cloud immer verschlüsselt werden. Dabei unterscheidet man grundlegend zwei verschiedene Datenzustände, die zu verschlüsseln sind: Data in Motion und Data at Rest.

Bei letzterem verfährt Oracle eine strikte Strategie: alles, was in der Cloud irgendwo persistiert wird, muss verschlüsselt sein. Dies gilt sowohl für Compute-Instances mit ihren Volumes, für den Object Store und den File Storage Service. Für Datenbanken wird sogar noch eine Ebene weitergedacht und für sämtliche Datendateien, selbst in der Standard Edition, eine Verschlüsselung mit Transparent Data Encryption forciert. Nutzt man die Standardeinstellungen von Oracle, so werden dafür von Oracle verwaltete Schlüssel in transparenter Umsetzung verwendet. Dies bietet immerhin Schutz davor, dass ein Abzug von Rohdaten oder die Entwendung von Hardware (wie Festplatten) sensible Daten offenlegen würde.

Die Verwaltung und Erstellung der Schlüssel für eine Verschlüsselung sämtlicher Daten kann und sollte man jedoch nicht zwingend nur Oracle überlassen. Hierfür wird ein Vault Service angeboten, welcher es auch erlaubt, eigenes

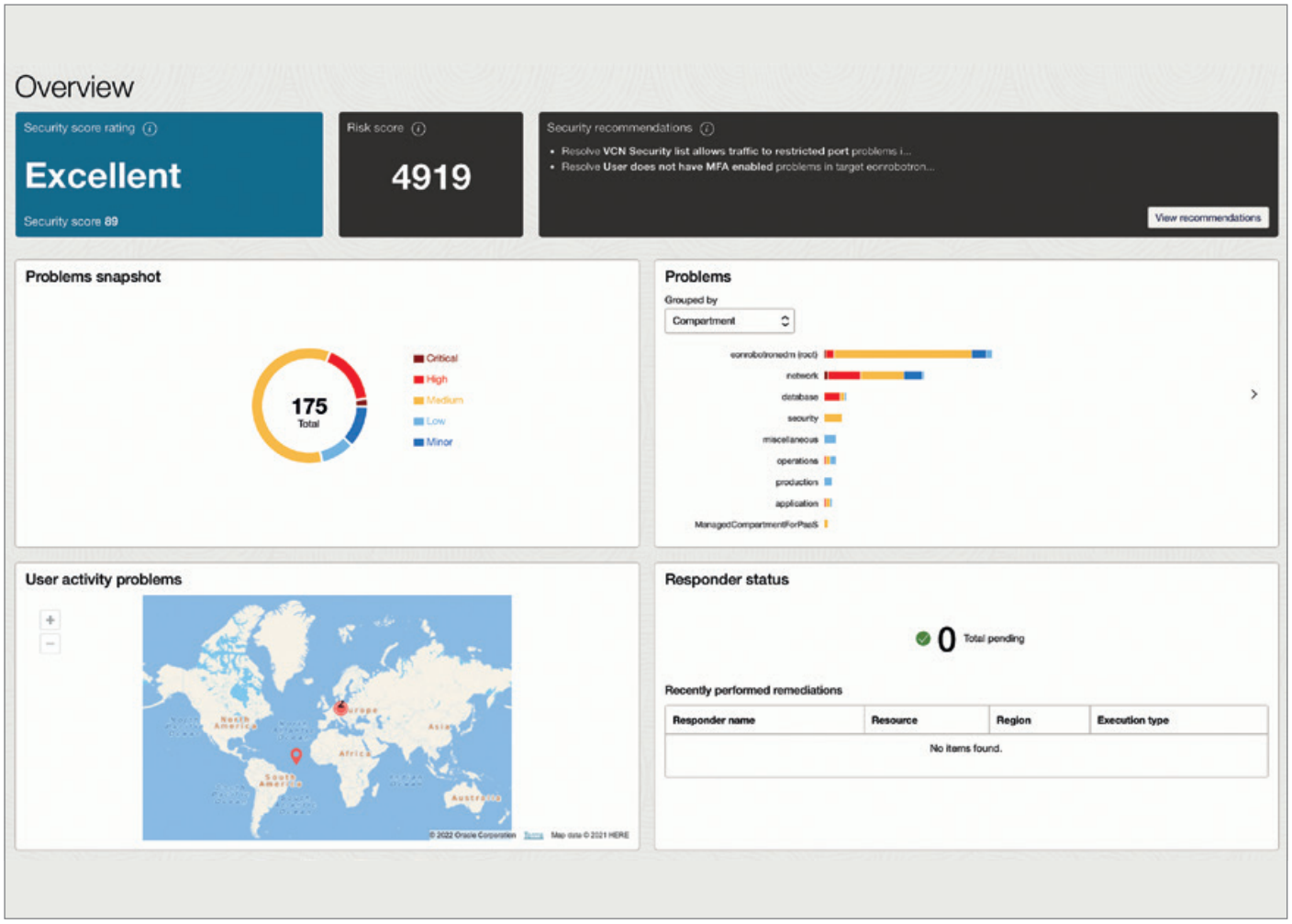


Abbildung 3: Der Cloud Guard bietet eine gute Übersicht über die eigene Konfiguration und erstellt eine Bewertung abhängig von der Kritikalität der Überprüfungsergebnisse. (Quelle: Sven Illert)

Schlüsselmaterial zur Verfügung zu stellen. Es ist in der Vergangenheit in bestimmten Software-Produkten schon vorgekommen, dass die dort verwendeten Algorithmen und Eingabedaten zu vorhersagbaren Schlüsseln führten. Mit eigener Hardware und unterschiedlicher Implementierung kann man dem jedoch etwas entgegenwirken und deren Verwendung ist bei besonders sensiblen Daten empfohlen *(siehe Abbildung 2)*.

Neben den persistierten Daten und der Verwaltung der dafür verwendeten Schlüssel ist es jedoch auch notwendig, jene Daten auf dem Weg von einem System zum nächsten zu schützen. Dies betrifft die Nutzung sicherer Protokolle wie SSH, was beim Zugriff auf Linux und UNIX-Server Standard ist. Für die von Oracle bereitgestellten Services gibt es in der Regel auch die Möglichkeit, eine entsprechende Transportverschlüsselung zu aktivieren. Beim Thema Block-Storage an einer VM kann einfach per Haken die In-Transit-Verschlüsselung aktiviert werden, welche den AES256-Algorithmus verwendet. Die Einbindung des File Storage Service (NFS) über eine sichere Kommunikation arbeitet mit TLS in Version 1.2 und muss hier je nach Betriebssystem über unterschiedliche Wege konfiguriert werden: Unter Linux erreicht man die Verschlüsselung über die Installation des Pakets *oci-fss-utils* sowie einer Anpassung der Mount-Optionen und unter Windows wird die Verwendung von *stunnel* beschrieben.

Für weitere Dienste gibt es verschiedene Möglichkeiten, die Transportverschlüsselung mittels TLS zu erreichen. Oracle bietet bei den Datenbanken zumindest für Autonomous Databases und Exadata Systeme standardmäßig eine verwaltete Verschlüsselung an. Bei Services, an deren Betriebssysteme man herankommt, kann man auch die Zertifikate selbst verwalten, ansonsten kommt es bei den Verschlüsselungsoptionen auf den jeweiligen Service an. Die Load Balancer zeigen sich hier flexibel und ermöglichen die Nutzung eigener Zertifikate, die extra dafür hochgeladen werden können, oder die Nutzung der von Oracle bereitgestellten Zertifikatsverwaltung. Diese ermöglicht es, eine eigene CA zu erstellen, eine Zwischen-CA zu etablieren oder gänzlich freie Leaf-Zertifikate zu hinterlegen. Letzteres bietet sich an, um zum Beispiel durch Let's Encrypt ausgestellte Zertifikate nutzen zu können. Alles in allem sollte man bei der Kommunikation mit den OCI-Services ähnlich dem Zero-Trust-Modell verfahren und überall dort die Kommunikation verschlüsseln, wo es möglich und sinnvoll ist.

Authentifizierungssicherheit

Wie in einem der vorherigen Abschnitte beschrieben, stellt Oracle auch einen

Service bereit, um verschiedene Geheimnisse sichern zu können: das Vault. Diese Geheimnisse können Schlüssel für die Verschlüsselung sein oder eben auch Passwörter für die Nutzung weiterer Dienste. Für die Nutzung des Database Performance Hubs oder des Database Managements ist es notwendig, den Zugang zu einem Monitoring-Nutzer innerhalb der Datenbank zu hinterlegen. Damit dies nicht im Klartext irgendwo abgespeichert werden muss, verwenden die Oracle Services die API des OCI-Vaults.

Diese Schnittstellen können jedoch auch eigene Anwendungen und Scripte verwenden, damit auch Maintenance-Arbeiten sicher gestaltet und der Zugriff auf die Kennwörter gesteuert werden kann. Um den unterschiedlichen Anforderungen gerecht zu werden, bietet Oracle verschiedene Möglichkeiten für den Umgang mit diesen an. So ist es möglich, seine Vaults in einem geteilten Bereich der gleichen Hardware-Partition zu sichern, natürlich ohne dass diese Zugriff auf die jeweils anderen Daten erhalten. Muss man jedoch stärkere regulatorische Vorgaben erfüllen, so ist es möglich, eigene Hardware-Partitionen zu bekommen, um die Sicherheit zu erhöhen. Auch auf die Art der Schlüsselgenerierung hat man Einfluss, denn es ist möglich, zwischen einer reinen softwarebasierten Variante und der Verwendung eines Hardware-Security-Moduls (HSM) zu wählen. Ebenfalls gibt es durch die Einbindung eigener On-Premises-Systeme und der Colocation eigener HSMs in diesem Bereich viel Bewegung und Möglichkeiten, die Arbeit mit den Geheimnissen zufriedenstellend umzusetzen.

Monitoring

Bei aller Vorsicht, die man bei der Implementierung einer sicheren Umgebung in der Cloud an den Tag legen kann, sind Systeme während des Betriebs immer einem Wandel unterzogen. Auch ausgeklügelte Change-Management-Prozesse sind kein Garant dafür, dass sich nicht doch Fehler einschleichen können. Deshalb ist es notwendig, die Gesamtsicherheit eines Systems zu überwachen. Hierzu bietet auch Oracle verschiedene Möglichkeiten für verschiedene Teile seiner Cloud-Umgebung.

Mit dem Data Safe kann man seine Datenbanken einem wiederkehrenden Audit des Datenbestands unterziehen. Dabei werden verschiedene Aspekte wie Account-Metaeigenschaften, Account-Nutzung und Verteilung sensitiver Daten betrachtet. Neben dieser Analyse werden auch Tools bereitgestellt, um die Daten aktiv zu schützen, indem sie zum Beispiel nach einer Spiegelung in Testdatenbanken maskiert werden können. Zur Laufzeit kann ebenso Data Safe unterstützen, indem Aktivitäten auditiert und die SQL-Firewall der Datenbank 23ai unterstützt wird.

Neben dienstspezifischen Monitoring Services wie dem angesprochenen Data Safe für Datenbanken und dem OS Management Hub für die Compute-Instanzen gibt es auch noch den Cloud Guard, welcher eine globale Sicht auf die eigene Umgebung liefert *(siehe Abbildung 3)*. Dieser ist kostenlos zu nutzen und hat eine Reihe von Standard-Policies, die auf den Empfehlungen des Center for Internet Security (CIS) basieren. Damit können potenziell unsichere Konfigurationen innerhalb von Compartments oder der gesamten Tenancy identifiziert werden. Auch konkrete Handlungsempfehlungen, wie diese Lücken geschlossen werden können, werden dargestellt. Wenn diese jedoch zu streng oder zu einfach erscheinen, können auch eigene Regelwerke hinterlegt werden, die eine eigene Bewertung der gesamten Infrastruktur zulassen.

Fazit

Die OCI bietet eine große Anzahl verschiedener Dienste und weiß diese auch zu schützen. Es wird viel dafür getan, schon im Standard eine sichere Umgebung bereitzustellen. Doch ist es wichtig, von den Möglichkeiten zur Absicherung und deren Notwendigkeiten zu wissen, um auch komplexe Umgebungen vor unbefugtem Zugriff auf verschiedensten Ebenen zu schützen. Die hier dargestellten Möglichkeiten geben nur einen Überblick über die Einflussmöglichkeiten und können nur als Startpunkt dienen. Themen wie erweitertes Policy Management und Zero Trust Packet Routing sind Beispiele, in welche man ebenfalls Zeit investieren sollte. Auch eigene Firewalls und die von Oracle angebotene Web Application Firewall bieten Möglichkeiten, sich gegen Bedrohungen zu schützen und erfordern tiefergreifendes Security-Verständnis. Generell muss jedoch festgehalten werden, dass es schon mit den Basisfunktionen einfach und gut möglich ist, eine sichere Cloud-Infrastruktur in der OCI umzusetzen.

Über den Autor

Sven Illert kommt seit 2008 mit verschiedensten Oracle-Technologien beruflich in Berührung und trug auch schon zu einigen DOAG-Veranstaltungen als Vortragender bei. Aktuell ist er als Leitender Systemberater der Robotron Datenbank-Software GmbH in vielfältigen Projekten unter anderem bei den Themen hochverfügbare Datenbank-Infrastrukturen, Engineered Systems und Oracle Cloud Infrastructure tätig.

Sven Illert
sven.illert@robotron.de

Herr der Daten: APEX, VPD und Data Redaction – die Gefährten

Dr. Thomas Petrik, Sphinx IT Consulting

Die Macht des Datenschatzes eines Unternehmens hat nicht nur auf die klassischen Hacker eine magische Anziehungskraft, allzu oft sind es unkontrollierte interne Datenflüsse, die massiven Schaden verursachen. Ein modernes Security-Konzept sollte daher im Kern der Datenhaltung – also in der Datenbank selbst – ansetzen. Virtual Private Database (VPD) und Data Redaction sind seit Langem integrierte Bestandteile der Oracle Enterprise Edition und ermöglichen den fein granulierten Schutz der Daten auf Row- und Column-Level mit höchster Effizienz und (bei korrekter Implementierung) unabhängig von der Applikation.

Der Oracle REST-APEX-Database (RAD) Stack

VPD und Data Redaction als feingranulare Autorisierungstechnologien in einer Oracle-Datenbank basieren im Wesentlichen darauf, dass sich die Zugriffserlaubnis auf Spalten oder Zeilen eines Datenbestandes aus dem momentanen Session- Context ergibt. An erster Stelle steht dort natürlich die Information, welcher User gerade auf die Daten zugreifen möchte. In 2-Tier-Architekturen ist dies einfach: Der Session User (den wir sehr einfach über die USER-Funktion oder aus sys_context('userenv', 'session_user' im Zugriff haben) ist hier das wichtigste Kriterium.

In einer 3-Tier-Architektur (und dazu gehört auch APEX) stellt sich dies etwas komplizierter dar, weil die Datenbank-

verbindung über einen technischen User im Rahmen eines Connection Pools hergestellt wird und somit der eigentliche End-User aus dem erweiterten Session Context zu ermitteln ist. Dazu kommt die Problematik, dass eine logische APEX-Session (definiert durch die APEX-Session-ID) nicht zwangsläufig innerhalb derselben Datenbanksession abgehandelt wird. Es ist das Wesen eines Connection Pools, dass eine Datenbank-Session nach einem Idle-Timeout von wenigen Sekunden an den Pool retourniert wird und der logischen APEX-Session bei Bedarf eine neue Datenbank-Session zugeteilt wird. Der SessionContext geht beim Zurückgeben der DB-Session an den Pool verloren und muss bei Zuteilung einer neuen DB-Session wiederhergestellt werden.

Wir haben es also mit einer „Stateless"-Connection zu tun, gleichgültig ob wir über Page Rendering oder AJAX-Calls sprechen. *Abbildung 1* zeigt diese von Oracle als „RAD" bezeichnete Architektur, in diesem Fall dargestellt mit ORDS als Standalone Middle Tier – mit dem integrierten Jetty Webserver einerseits und einem Connection Pool zur Datenbank, der den APEX_PUBLIC_USER als technischen User nutzt.

APEX Session Handling & Autorisierung

Doch wie ist es möglich, mit dem APEX_PUBLIC_USER auf das eigentliche Applikationsschema zuzugreifen? Ein Blick auf die Privilegien dieses technischen Users zeigt schnell, dass es keinerlei Grants auf das Applikationsschema (früher auch als „Parsing User" bezeichnet) gibt, ebenso wenig existieren irgendwelche Grants auf das APEX-Repository – zumindest nicht direkt.

Wird hier auf magische Weise der eigentliche User geändert? Ja und Nein. Ein Blick auf die V$SESSION zeigt, dass alle Sessions stets unter dem APEX_PUBLIC_USER laufen *(siehe Abbildung 2)*. Dieser hat jedoch Zugriff auf eine Reihe von Packages im APEX-Repository (APEX_240100.WWV_FLOW_* – es handelt sich in diesem Fall um die Version APEX 24.1), wobei die Execute-Privilegien über Grants an den PUBLIC User und die Zugriffe über Public-Synonyme erfolgen.

Diese WWV_FLOW-Packages ihrerseits verwenden das Definer Rights Package SYS.WWV_DBMS_SQL_APEX_240100 aus dem SYS-Schema und dieses wiederum nutzt ein Oracle Package, das undokumentiert ist: DBMS_SYS_SQL. Darin findet sich die Prozedur PARSE_AS_USER, der – analog zu dem wohl bekannten und dokumentierten Package-Call DBMS_SQL.parse – ein SQL-Statement übergeben werden kann, zusätzlich jedoch auch eine beliebige User-ID, unter der dieses ausgeführt werden soll. Oracle hat mit dieser Methode seit jeher einen sudo-Mechanismus in der Datenbank implementiert, der sich für DBAs als äußerst nützlich erweisen kann. *Listing 1* zeigt ein Beispiel einer Prozedur, wie man diese Funktionalität bequem kapseln könnte, um so Database Links in beliebigen Schemata anzulegen, ohne den in zahlreichen Blogs beschriebenen Weg von Scheduler-Jobs gehen zu müssen.

Doch kommen wir wieder zurück zum Session-Handling. Für Row- und Column-Level-Security benötigen wir den tatsächlichen User, der sich in der APEX-Applikation eigeloggt hat. Diesen finden wir in der V$SESSION in den Feldern CLIENT_INFO und CLIENT_IDENTIFIER, wobei ersteres zusätzlich die Workspace-ID enthält und zweiteres die APEX Session-ID. Beide Werte stehen auch über den USERENV-Context zur Verfügung. Zusätzlich existiert auch noch ein Secure Application Context APEX$SESSION, der (unter anderem) den APP_USER beinhaltet.

Wenn die logische APEX-Session (identifiziert durch die Session-ID in der URL) nun nach einiger Zeit der Inaktivität wieder eine Datenbank-Session aus dem Connection Pool anfordert, wird der Rest der Session-Information über die View APEX_WORKSPACE_SESSIONS, die ihrerseits auf der Tabelle WWV_FLOW_SESSIONS$ basiert, geholt und der Session Context erneut gesetzt. Auf diese Art und Weise garantiert APEX, dass in jeder Situation der End-User über den SYS_CONTEXT abfragbar und somit nutzbar für ein darüberliegendes Security Framework bleibt – ganz gleich, ob es sich um ein Page Rendering oder einen asynchronen AJAX-Call handelt.

Sobald eine Connection an den Pool retourniert wird, setzt APEX den Session Context durch Aufruf von DBMS_SESSION.modify_package_state (DBMS_SESSION.reinitialize) komplett zurück.

APEX In-Database Security

Die Autorisierung für den Datenzugriff erfolgt grundsätzlich auf 3 Ebenen:

- Object Level
- Row Level (RLS)
- Column Level (CLS)

```
PROCEDURE exec_sql_as_user (p_sql IN CLOB, p_username IN VARCHAR2)
    IS
        -- execute a statement as any user
        v_cur   INTEGER;
        v_uid   INTEGER;
    BEGIN
        IF p_username IS NULL
        THEN
            EXECUTE IMMEDIATE p_sql;
        ELSE
            SELECT user_id
              INTO v_uid
              FROM dba_users
             WHERE username = p_username;
            v_cur := DBMS_SQL.open_cursor;
            sys.DBMS_SYS_SQL.parse_as_user (v_cur
                                            ,p_sql
                                            ,DBMS_SQL.native
                                            ,v_uid);
            DBMS_SQL.close_cursor (v_cur);
        END IF;
END;
```

Listing 1: „sudo" für den DBA

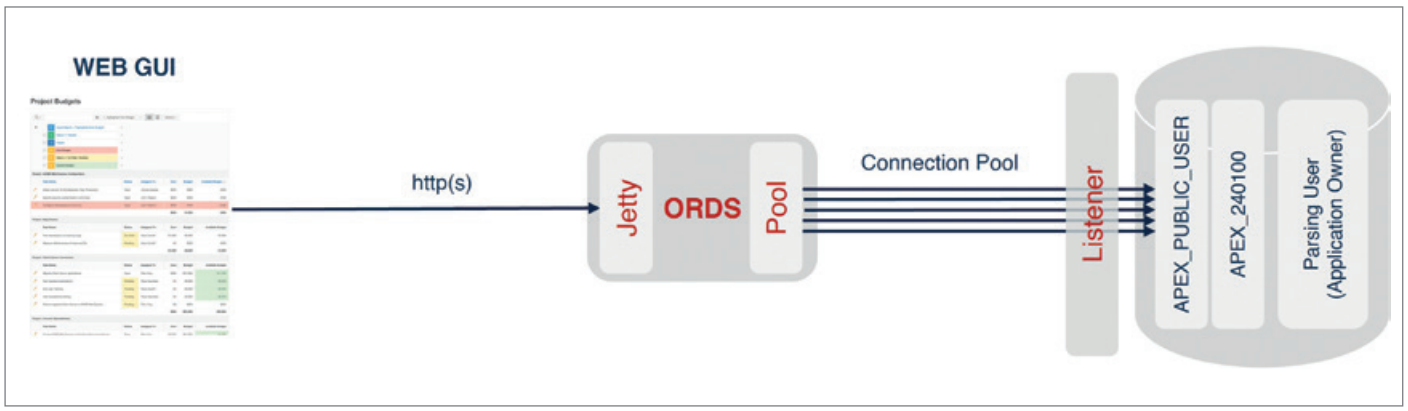


Abbildung 1: Oracle_RAD-Architektur (Quelle: Thomas Petrik)

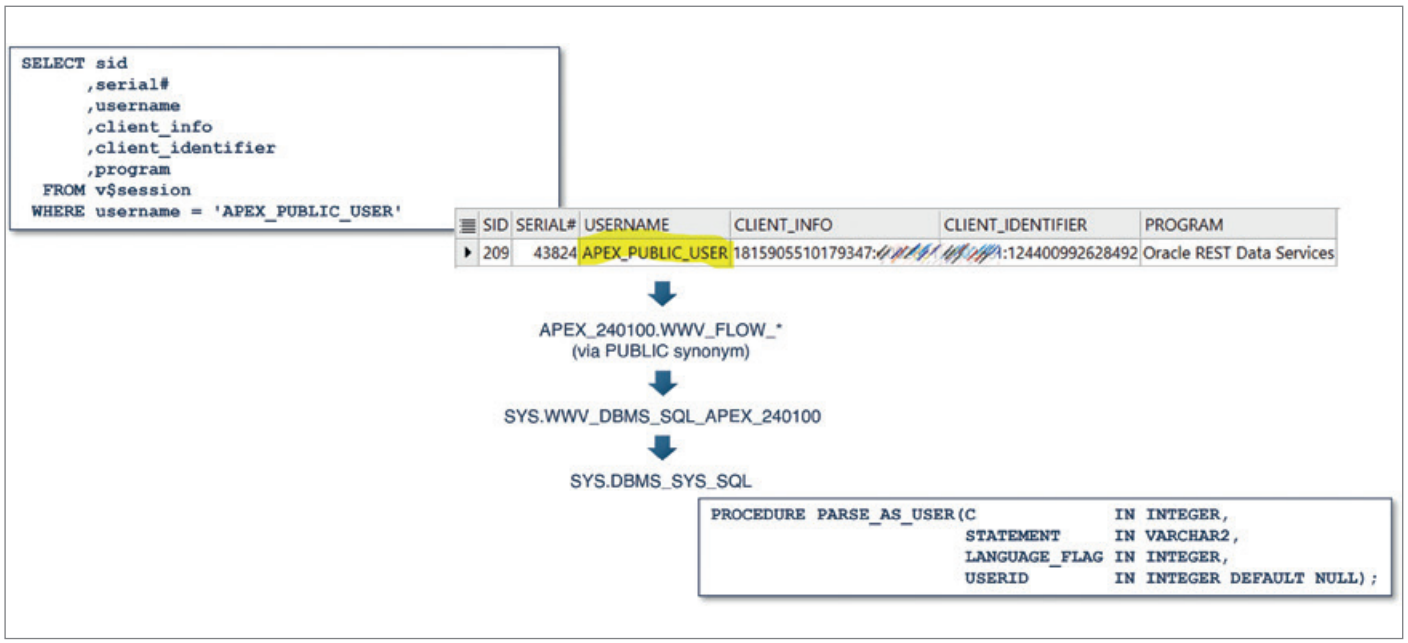


Abbildung 2: „sudo"-Funktion in einer APEX-Session (Quelle: Thomas Petrik)

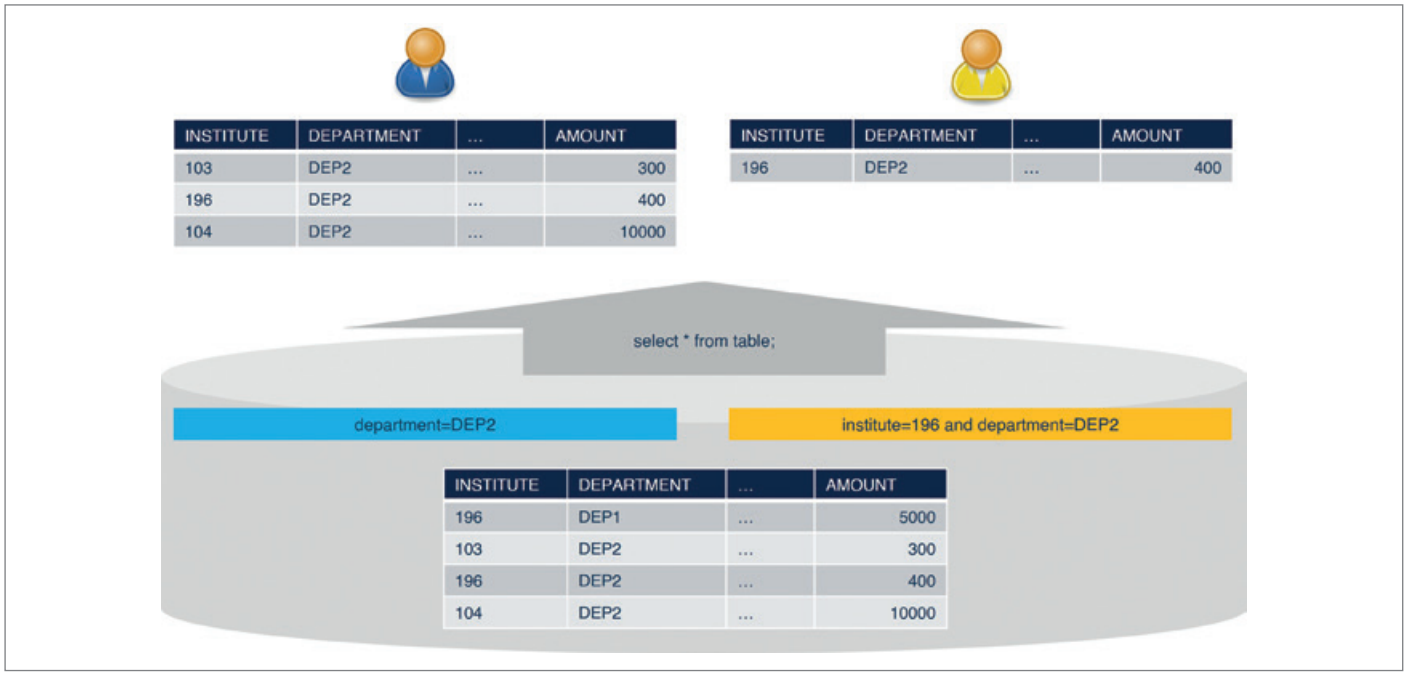


Abbildung 3: Row level Security mittels VPD (Quelle: Thomas Petrik)

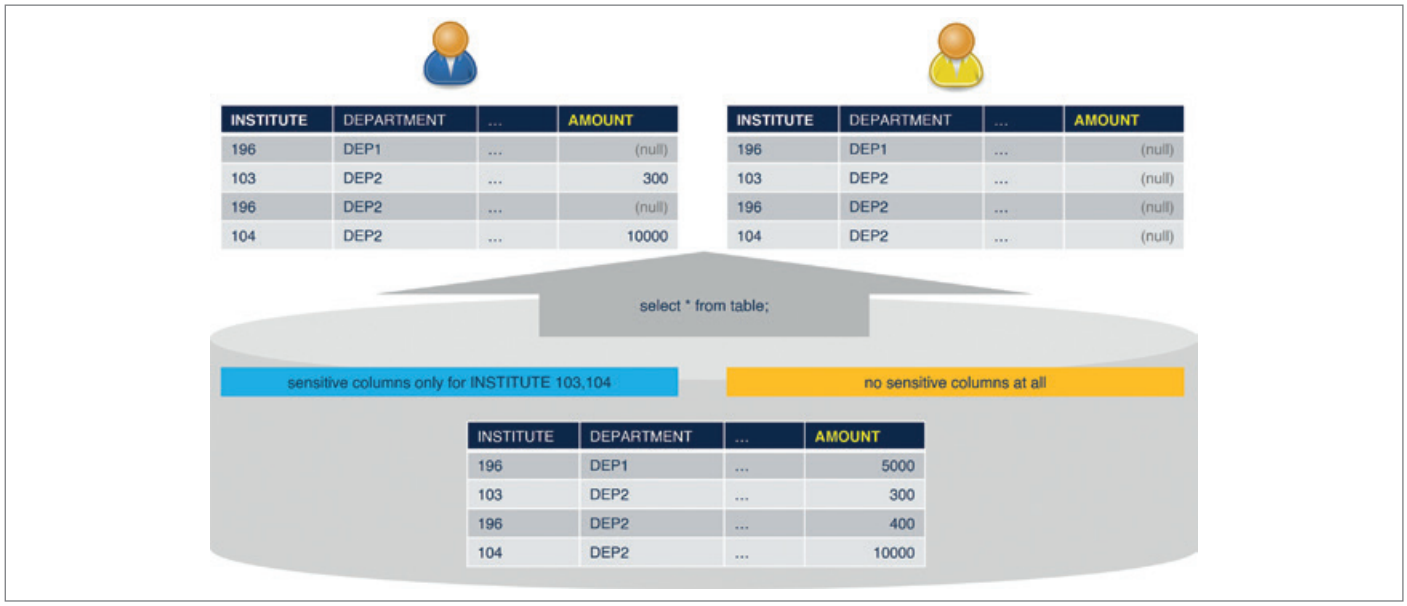


Abbildung 4: Column Level Security mittels VPD (Quelle: Thomas Petrik)

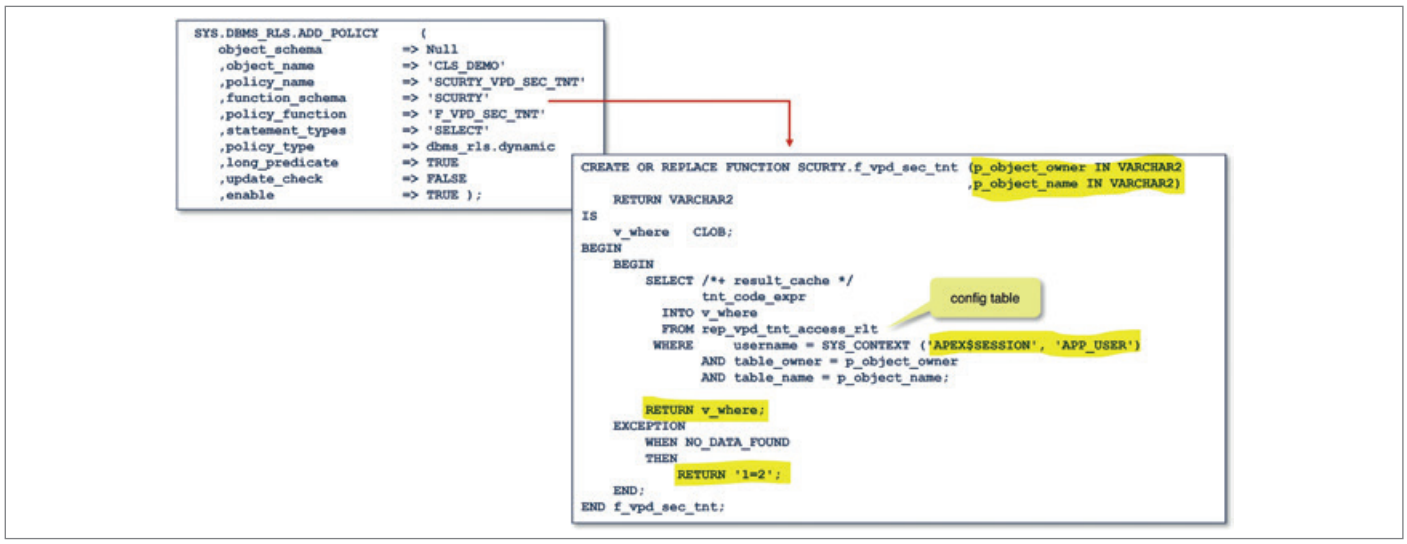


Abbildung 5: Funktionsweise der Row Level Security – Policy & Function (Quelle: Thomas Petrik)

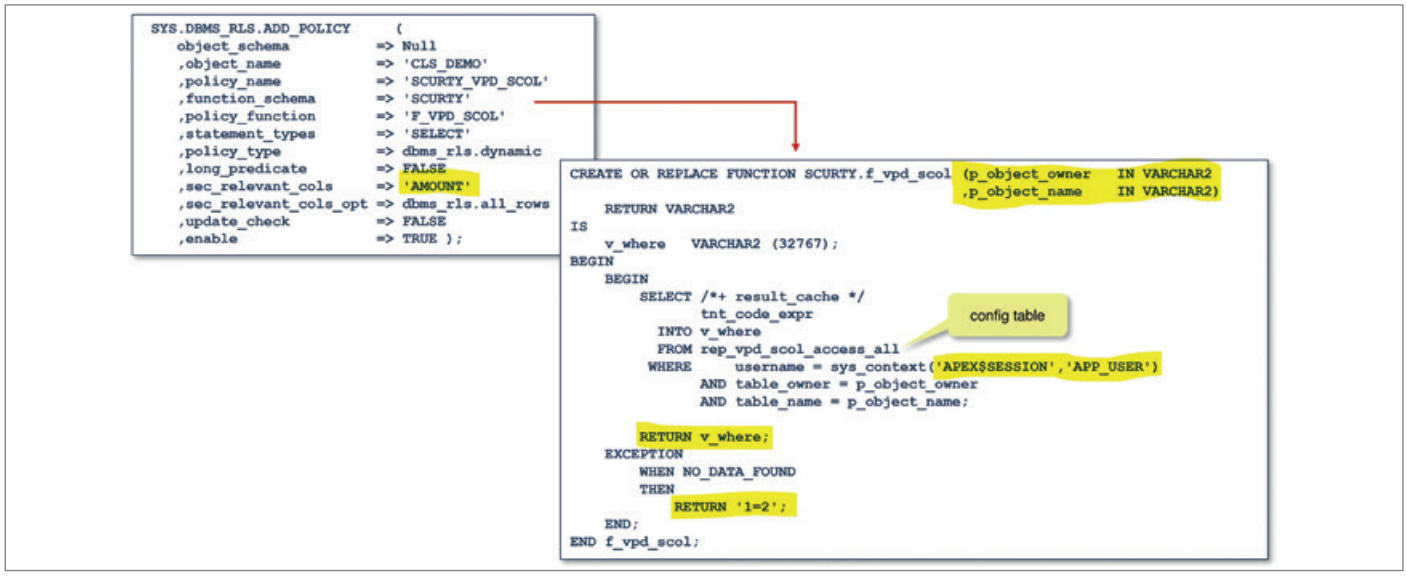


Abbildung 6: Funktionsweise der Column Level Security – Policy & Function (Quelle: Thomas Petrik)

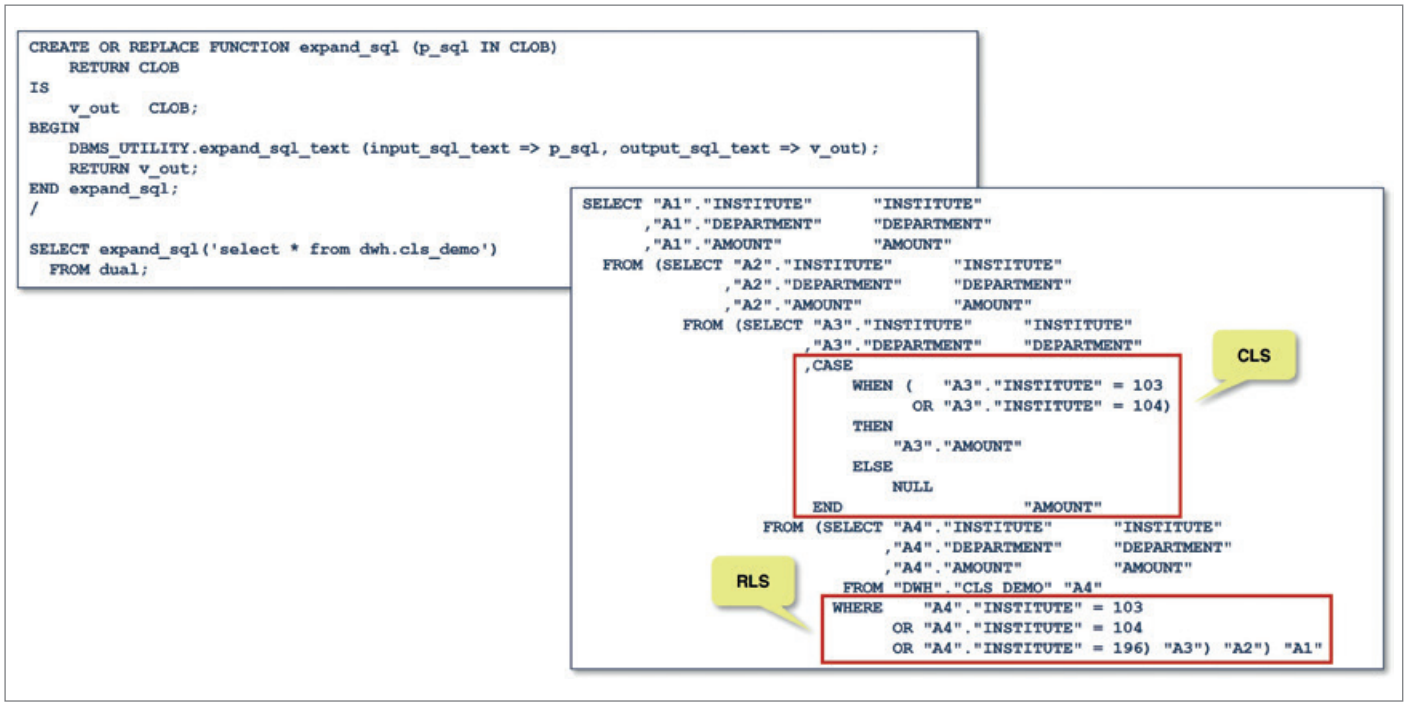


Abbildung 7: Statement Expansion nach Anwendung von RLS und CLS (Quelle: Thomas Petrik)

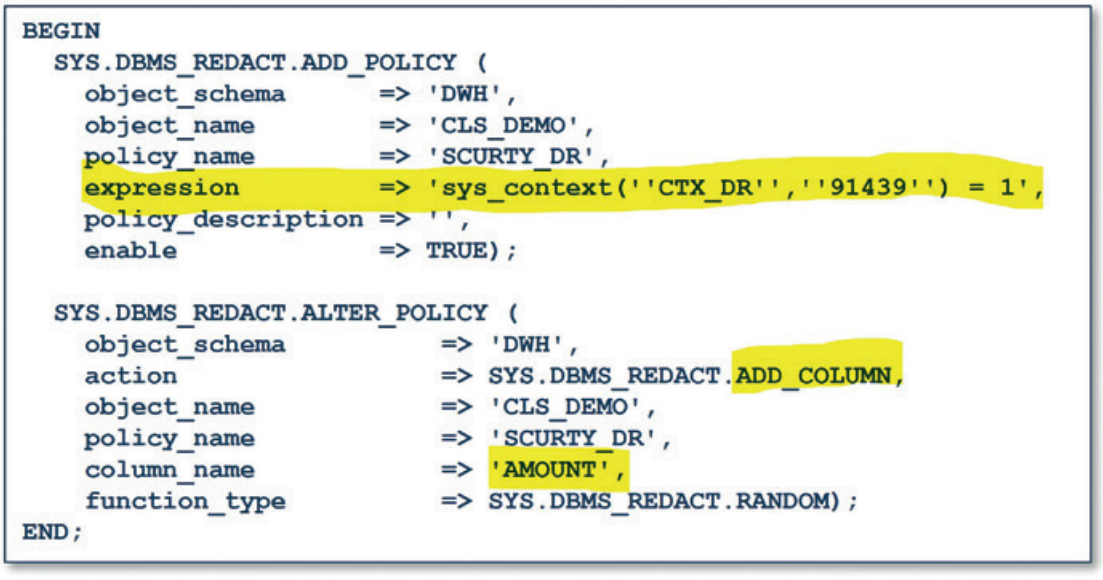


Abbildung 8: Anwendung einer Data Redaction Policy (Quelle: Thomas Petrik)

Object-Level-Security spielt in APEX keine Rolle, da der Parsing User (also der Applikations-User) ohnehin vollen Zugriff auf seine Objekte hat und die Frage, wer auf bestimmte Sichten über Tabellen oder Views zugreifen darf, tatsächlich über die Applikationsmasken geregelt wird.

Anders sieht die Lage für RLS und CLS aus: Diese Logik wird durch VPD und Data Redaction beigesteuert. Beide Technologien ermöglichen das dynamische und transparente Ausblenden von Zeilen beziehungsweise die Maskierung von Spalten, ohne dass dies in der Applikation berücksichtigt werden muss.

Hinweis:

Die Virtual Private Database ist ein kostenloses Feature der Enterprise Edition, wohingegen die Data Redaction eine kostenpflichtige Option ist.

Abbildung 3 zeigt plakativ den Effekt von Row-Level-Security, *Abbildung 4* jenen von Column-Level-Security. In beiden Fällen wird stets ein „select * from …“ abgesetzt, doch in Abhängigkeit vom End-User ein sehr unterschiedliches Ergebnis angezeigt. Im Fall der Data Redaction wird im Vergleich zur CLS Columns nicht auf NULL gesetzt, sondern durch eine (wählbare) Maskierung modifiziert.

Hinweis:

Es ist wichtig zu verstehen, dass die VPD ein Rewrite des Statements bewirkt noch bevor es zum Parsing kommt, wohingegen die Redaction am Resultset ansetzt, also nach dem Fetch der Daten. Dies hat zur Konsequenz, dass eine VPD tatsächlich nicht umgangen werden kann

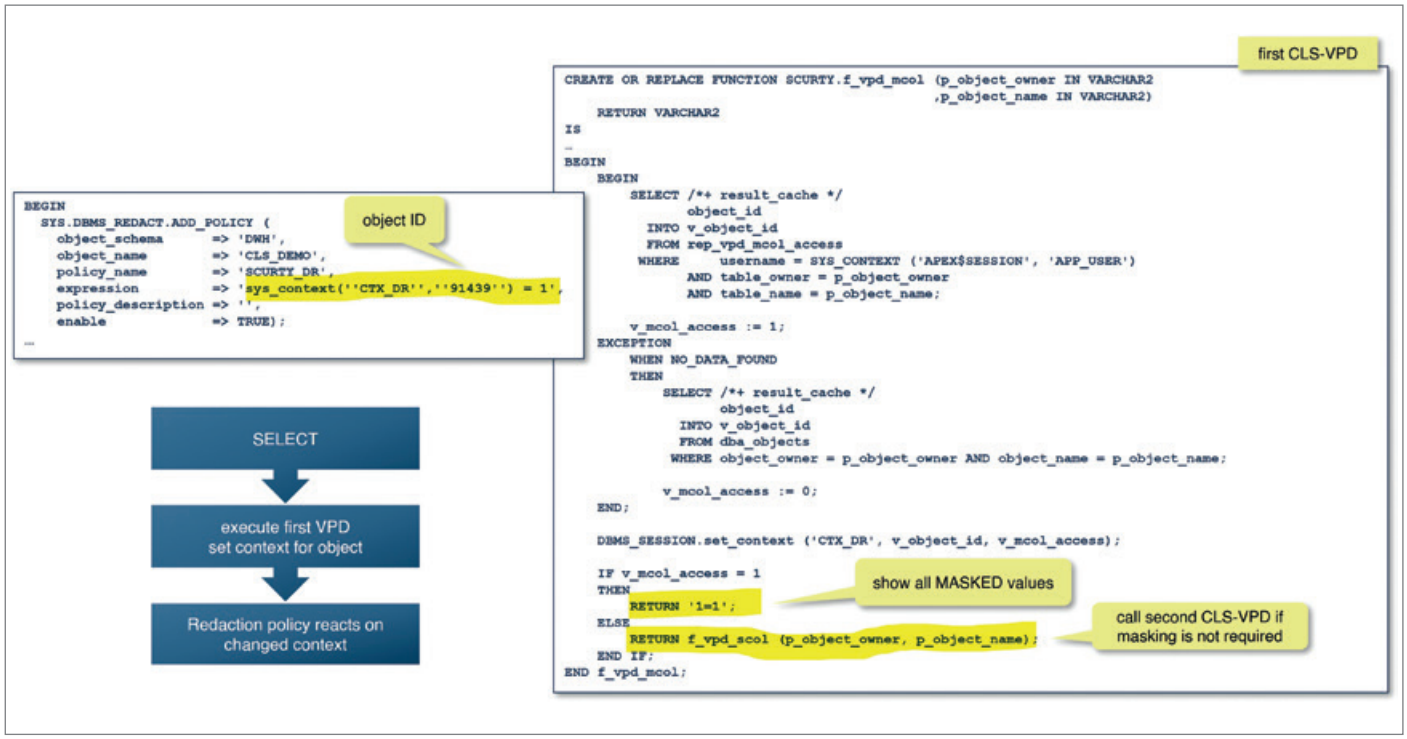


Abbildung 9: Dynamische Steuerung der data Redaction mittels vorgelagerter VPD (Quelle: Thomas Petrik)

im Gegensatz zur Redaction, die (direkten SQL-Zugriff auf die Daten vorausgesetzt) durch gezielten Einsatz von Where-Klauseln in einem Ausschlussverfahren umgangen werden kann. Im Kontext einer APEX-Applikation, wo es üblicherweise keinen Direktzugriff durch den End-User gibt, stellt dies allerdings kein Risiko dar.

Abbildung 5 und *Abbildung 6* geben ein vereinfachtes (aber funktionsfähiges) Beispiel für die Umsetzung von RLS- beziehungsweise CLS-VPDs: Einem Objekt (Table, View, Synonym) wird eine Policy assoziiert, die Policy selbst referenziert eine PL/SQL-Function, deren Aufgabe es ist, eine Where-Klausel zu retournieren, die besagt, welche Rows im Falle einer RLS-Policy im Resultset stehen sollen beziehungsweise im Fall einer CLS, in welchen Zeilen die Werte der in der CLS-Policy spezifizierten Column angezeigt werden dürfen. Ein Return-Value FALSE verhindert in beiden Fällen die Anzeige. Entscheidend ist hier die Kombination einer Konfigurationstabelle mit dem aktuellen Session Context, sodass die passende Where-Clause aus der Kombination des Objekts mit dem APP_USER oder anderen Merkmalen konstruiert werden kann.

DBMS_UTILITY.expand_sql_text zeigt, was der Optimizer vor dem Parsing aus dem Statement macht: Die Where-Clause der CLS wird zu einem Case-Statement verarbeitet wohingegen der Filter der RLS 1:1 angewandt wird *(siehe Abbildung 7)*.

Die Policy der Data Redaction *(siehe Abbildung 8)* sieht im Gegensatz zur VPD keine Verwendung von Funktionen vor, um eine dynamische Filterbedingung aus Kontext- und Objektinformation zu bauen. Lediglich einfache Expressions (wie im gezeigten Beispiel) unter Verwendung von SYS_CONTEXT können verwendet werden. Das ist grundsätzlich verständlich, da die Redaction – wie bereits erwähnt – erst am Resultset ansetzt, für eine Where-Clause ist es da schon zu spät.

Die dynamische Data Redaction

Müssen wir also bei Verwendung der Data Redaction tatsächlich auf die Flexibilität der VPD verzichten? Weder die Oracle-Dokumentation noch einer der zahlreichen Blogs liefern einen Ansatzpunkt, wie die Redaction Policy zur Laufzeit auf den Session Context und das jeweilige Objekt reagieren könnte.

Und doch gibt es hier einen sehr eleganten Ausweg, wenn man bedenkt, dass die VPD-Funktion zwingend stets vor dem Parsing ausgeführt wird (sofern eine „dynamic“ Policy zur Anwendung kommt), die Redaction Policy hingegen immer erst nach dem Fetch wirkt. Es liegt daher nahe, eine eigene VPD-Policy zu schreiben, die nach bewährter Methode eine Variable eines Secure Application Context umsetzt, auf die in weiterer Folge die Redaction im selben Statement reagieren kann. *Abbildung 9* zeigt diese Vorgehensweise, wobei im Application Context CTX_DR Object-IDs als Attribute verwendet werden, deren Wert dynamisch (wiederum auf Basis einer Konfigurationstabelle) auf 0 oder 1 gesetzt werden und dementsprechend in der Policy True oder False ergeben.

Hinweis:

In Kombination von Data Redaction und VPD kommt es (zumindest) in Oracle 19c häufig zu ORA-28094: SQL construct not supported by data redaction. Durch Setzen des Hidden Parameters strict_redaction_semantics" = False konnte dieser Fehler umgangen werden, ohne dass es zu Datenfehlern kam. Eine Nachfrage beim Oracle Support ist aber vor Anwendung in Produktivsystemen jedenfalls dringend zu empfehlen.

Praktikable Umsetzungskonzepte

Wir haben gezeigt, dass die Kombination unterschiedlicher VPDs sowie der Data

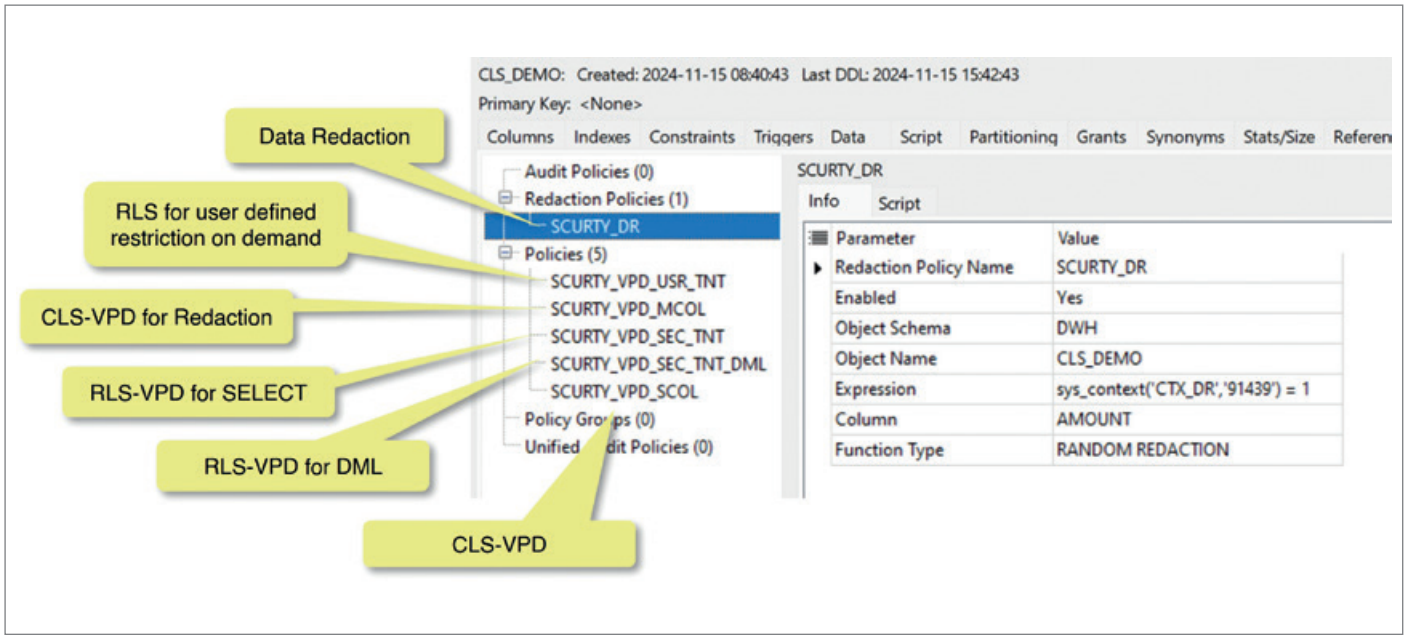


Abbildung 10: Überlagerung mehrerer Policies im SCURTY-Framework (Quelle: Thomas Petrik)

Redaction nahezu unbegrenzte Möglichkeiten für die Umsetzung fein granulierter Security bietet. Um in der Praxis allerdings eine wartbare Lösung ohne die Gefahr von Inkonsistenzen und Wildwuchs zu implementieren, bedarf es eines vollautomatisierten Frameworks, das lediglich durch ein vorgelagertes Identity Management (in den meisten Fällen ein Active Directory) gesteuert wird.

Die Sphinx IT bietet bereits seit vielen Jahren ein fertiges Framework unter dem Namen SCURTY an, das diese Technologien nutzt und folgende Ziele verfolgt:

1. Single Point Of Control (SPOC)
   Die Autorisierung liegt zentral in der Datenbank und kann somit vollkommen transparent von allen Applikationen genutzt werden.
2. Vereinfachung (Streamlining) der Applikationen
   Die Applikation muss sich nicht um die Zugriffslogik kümmern, gleichzeitig kann es zwischen Applikationen nie zu Inkonsistenzen kommen. Das Need-To-Know-Prinzip wird einheitlich garantiert.
3. Client Tool-Unabhängigkeit
   Security wird unabhängig vom verwendeten Tool oder der verwendeten Applikation.
4. Zentrales Audit
   Da die Autorisierung in der DB geregelt ist, kann dort auch zentral das Audit durchgeführt werden.
5. 100% Metadata Driven
   Keine Programmierung, rein deklaratives Customising unter ausschließlicher Verwendung der Features der Oracle Enterprise Edition.

Abbildung 10 zeigt ein Beispiel für die Überlagerung mehrerer Policies auf einem Objekt, wie sie vom SCURTY-Framework generiert werden.

Schlussfolgerungen

Für eine feingranulare Autorisierung bietet die Oracle Enterprise Edition mit Virtual Private Database und Data Redaction herausragende Features, die allerdings erst durch ein umfassendes Framework kombiniert und nutzbar gemacht werden müssen. Der Komplexitätsgrad ist nicht zu unterschätzen, die Umsetzung wird allerdings durch eine unglaubliche Flexibilität belohnt: vereinheitlichte und vor allem konsistente Autorisierung auf Zeilen- und Spaltenebene und eine bedeutende Entlastung der Applikationen, die sich mit diesem Thema überhaupt nicht mehr befassen müssen. Speziell APEX bietet von Haus aus die besten Voraussetzungen, um an ein derartiges Framework anzudocken.

Eine kurze Anmerkung zum Thema Performance darf zum Schluss auch nicht fehlen: Der korrekte Einsatz von VPDs mit dem entsprechendem Hintergrundwissen um die Funktionalität wird nie zu Performance-Problemen führen, wenngleich es natürlich die eine oder andere Falle zu umgehen gilt.

Über den Autor

Dr. Thomas Petrik arbeitet seit fast 3 Jahrzehnten mit Oracle-Datenbanken und befasst sich ebenso lange mit den Themen Security, Performance und Effizienz in der Betriebsführung (Automatisierung). Es ist kein Zufall, dass aus dieser Tätigkeit im Laufe der Zeit Services und Produkte entstanden sind, die aus der Praxis für die Praxis geschaffen wurden und erfolgreich von den Kunden der Sphinx IT Consulting eingesetzt werden.

Dr. Thomas Petrik
thomas.petrik@sphinx.at

Erste Schritte mit Transparent Data Encryption (TDE) – Teil 1

Meris Bihorac, DBConcepts

In einer Zeit, in der Datenschutzverletzungen eine ständige Bedrohung darstellen, dient die Datenverschlüsselung als wichtiger Schutzschild für sensible Informationen. Dieser Artikel befasst sich mit der transparenten Datenverschlüsselung (TDE), einer Methode, die Daten direkt in der Datenbank verschlüsselt und so eine nahtlose Sicherheitsschicht bietet, ohne die Anwendungslogik zu verändern. Wir werden die verschiedenen Formen von TDE (Transparent Data Encryption) betrachten, einschließlich Tablespace- und Spaltenverschlüsselung, die jeweils unterschiedliche Sicherheitsanforderungen erfüllen. Darüber hinaus werden wir uns mit den Mechanismen von Oracle Keystores und Wallets befassen, wichtigen Tools für die sichere Verwaltung von Verschlüsselungsschlüsseln.

Warum müssen wir die Daten verschlüsseln?

Ohne Verschlüsselung bestehen erhebliche Sicherheitslücken, die Angreifer ausnutzen können.

- Diebstahl von Datenbanksicherungen
- Unautorisierter Zugang zur Datenbank
- Mitlesen von unverschlüsselten Daten bei der Netzwerkübertragung
- Kompromittierung von Datenbankexporten

Beispiele:
Gesetzliche Anforderungen:

- GDPR
  - Ein europäisches Unternehmen muss sicherstellen, dass personenbezogene Daten von EU-Bürgern verschlüsselt werden, um den Zugriff durch unbefugte Dritte zu verhindern.
- NIS/2
  - In kritischen Infrastrukturbereichen wird Verschlüsselung eingesetzt, um sensible Daten zu schützen und größere Dienstunterbrechungen zu vermeiden.
- DORA
  - Unternehmen müssen Betriebsdaten verschlüsseln, um ihre Dienste vor Cyber-Bedrohungen zu schützen.

Reputation und Haftungen:

- Öffentlicher Schaden
  - Die Verschlüsselung von Daten trägt dazu bei, öffentlichen Schaden abzuwenden, indem sie sicherstellt, dass sensible Informationen im Falle eines Diebstahles sicher und unzugänglich bleiben, wodurch der Ruf eines Unternehmens und das Vertrauen der Kunden geschützt werden.
- Rechtliche und finanzielle Verpflichtungen
  - Die Verschlüsselung von Daten ist ein Teil rechtlicher und finanzieller Verpflichtungen, indem sie die Datenschutzgesetze einhält und mögliche Geldstrafen und Klagen aufgrund von Datenschutzverletzungen minimiert.

Was ist Transparent Data Encryption?

Transparent Data Encryption (TDE) ist eine Funktion, die sensible Informationen, die in Tabellen, Tablespaces und Datenbanksicherungen gespeichert sind, verschlüsselt und damit eine zusätzliche Sicherheitsebene zum Schutz der Daten bietet. Diese Verschlüsselung stellt sicher, dass die Daten selbst bei einer Beschädigung von Speichermedien oder Dateien für unberechtigte Personen unzugänglich bleiben. Die Verschlüsselung findet transparent im Hintergrund statt, so dass die Applikationen unbemerkt arbeiten können, ohne dass Veränderungen an diesen vorgenommen werden müssen. *[1]*

Welche Arten der Verschlüsselung bietet TDE?

TDE bietet zwei Hauptarten der Verschlüsselung, die jeweils für bestimmte Anwendungsfälle vorgesehen sind, auf die wir später noch eingehen werden. Diese sind:

- Tablespace-Verschlüsselung (Tablespace Encryption)
- Spaltenverschlüsselung (Column-Level Encryption)

Tablespace Encryption

Die Tablespace-Verschlüsselung ermöglicht es, einen gesamten Tablespace zu verschlüsseln *(siehe Abbildung 1)*.

In einem verschlüsselten Tablespace erstellte Objekte werden automatisch durch Verschlüsselung gesichert. Transparente Datenverschlüsselung für Tablespaces ist besonders vorteilhaft, wenn sensible Informationen über mehrere Spalten verteilt sind oder wenn das Ziel darin besteht, ganze Tabellen und nicht nur einzelne Felder zu schützen. Dadurch entfällt die Notwendigkeit, jede

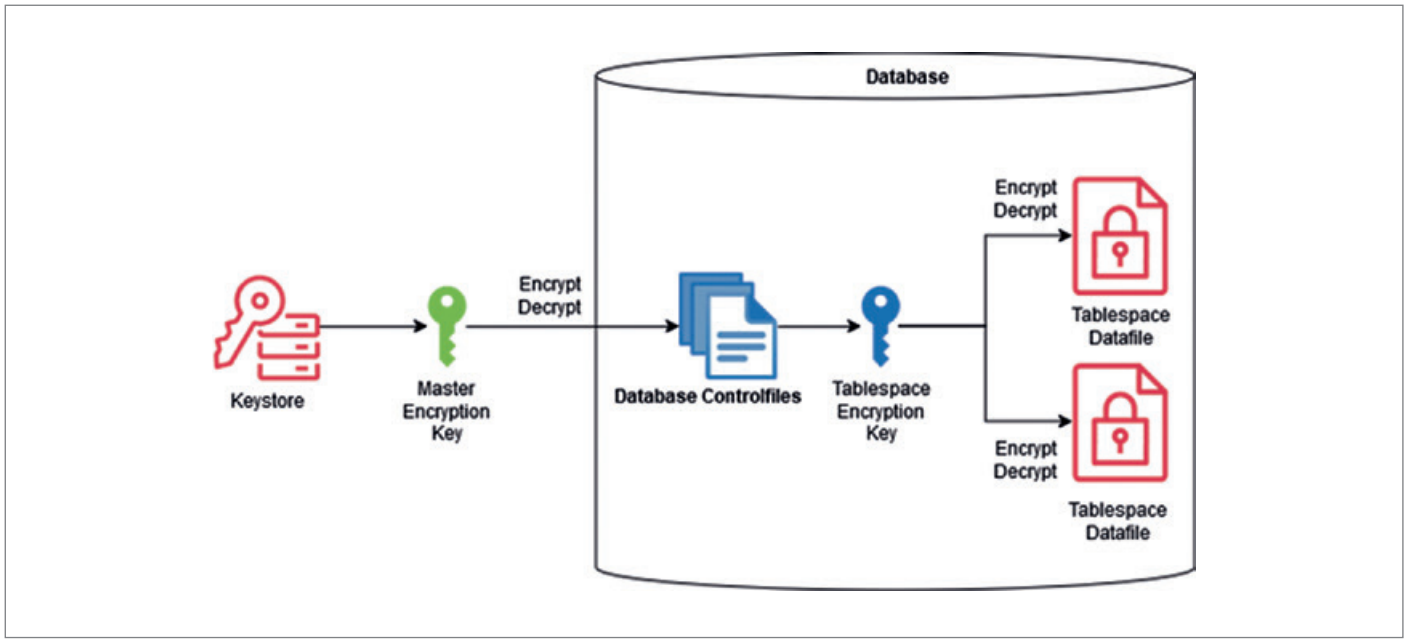


Abbildung 1: TDE-Tablespace-Verschlüsselungsprozess (Quelle: Meris Bihorac)

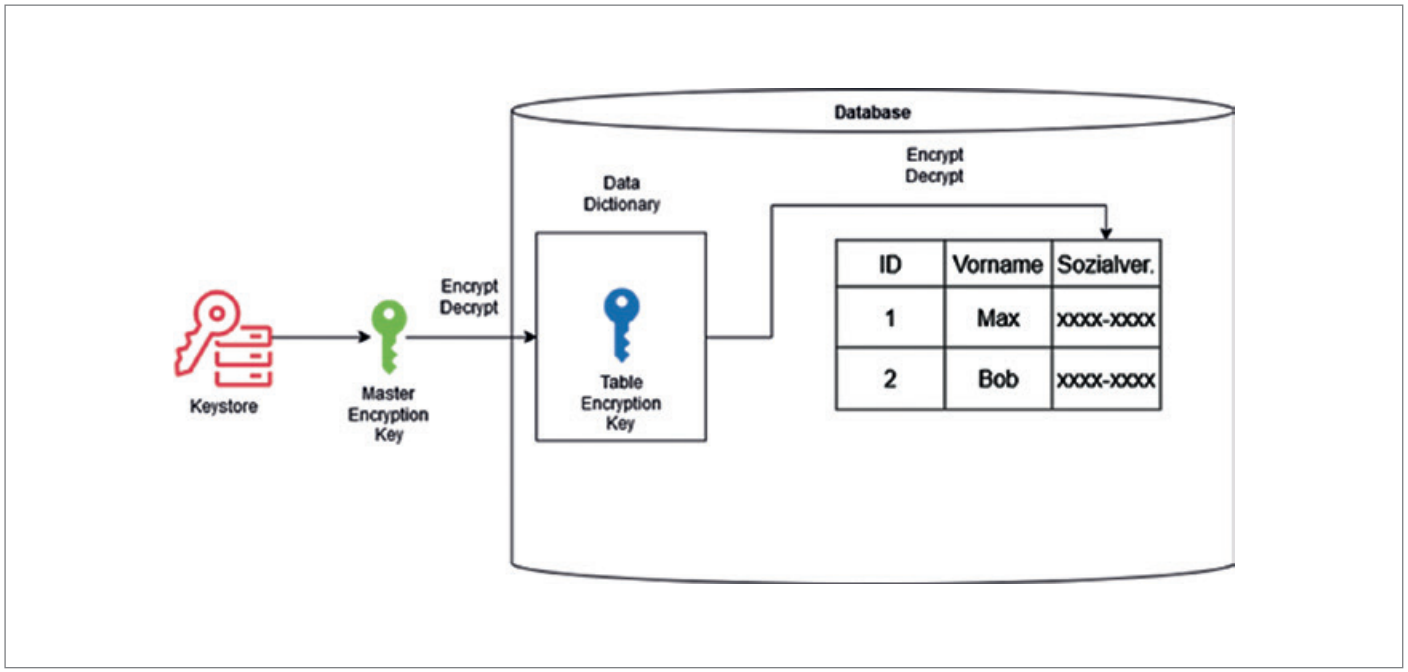


Abbildung 2: TDE-Spalten-Verschlüsselungsprozess (Quelle: Meris Bihorac)

einzelne Spalte zu prüfen und zu entscheiden, welche Spalten verschlüsselt werden müssen.

Der gesamte Inhalt eines verschlüsselten Tablespace, einschließlich der Transaktions-Daten (Redo und Undo), ist durch TDE gesichert. Diese Verschlüsselung umfasst ausschließlich Daten innerhalb der Tablespaces, Daten wie zum Beispiel externe BFILEs, die außerhalb der Datenbankumgebung gespeichert sind, bleiben unberührt. Wenn eine Tabelle im verschlüsselten Tablespace eine BFILE-Spalte enthält, bleibt diese Spalte unverschlüsselt, da die Daten extern abgelegt werden.

Die in verschlüsselten Tablespaces gespeicherten Daten werden auf Tablespace-Ebene verschlüsselt und beim Zugriff automatisch entschlüsselt, wobei der Vorgang für die Anwendungen völlig transparent ist. Diese Transparenz bedeutet, dass Datenbankbenutzer und Anwendungen von der Verschlüsselung nichts mitbekommen, so dass sie normal arbeiten können, ohne etwas zu bemerken.

Trotz der Verschlüsselung müssen alle bestehenden Sicherheitsmechanismen der Anwendung aktiv bleiben. Dazu gehören Eingabevalidierung, Ausgabeverschlüsselung, Authentifizierung und Autorisierungskontrollen. Die Verschlüsselung ersetzt diese Mechanismen nicht, sondern ergänzt sie, indem sie eine Ebene der Vertraulichkeit hinzufügt. *[2]*

Die Transparent Data Encryption von Oracle verwendet ein hierarchisches Schlüsselverwaltungssystem, um die Datensicherheit zu garantieren. Das Kernelement dieses Verfahrens ist der Master Encryption Key (MEK), der als Root-Key dient. Der MEK wird sicher in einem Keystore gespeichert, zum Beispiel in einer Oracle-Wallet, einem Oracle Key Vault oder dem Oracle Cloud Infrastructure (OCI) Key Management Service (KMS), und wird niemals in der Datenbank selbst gespeichert.

Unterhalb des MEK befindet sich der Data Encryption Key (DEK), der für die Verschlüsselung von Datenbankdaten zuständig ist. Der DEK wird in der Datenbank gespeichert, insbesondere im Control File und in den Headern der Datafiles und wird mit dem MEK verschlüsselt.

- Schreiben von Daten:
  - Wenn Daten geschrieben werden, verschlüsselt der DEK sie entweder auf Tablespace- oder auf Spaltenebene, und die verschlüsselten Daten werden in den Datenfiles der Datenbank gespeichert.
- Lesen von Daten:
  - Beim Abrufen von Daten wird auf den verschlüsselten DEK aus der Control File zugegriffen. Der MEK, der aus dem Keystore abgerufen wird, entschlüsselt den DEK, der dann nahtlos die Daten im Hintergrund entschlüsselt. Die Anwendung arbeitet normal und weiß nichts von diesen Ver- und Entschlüsselungsvorgängen. *[2]*

Column-Level Encryption

Die Verschlüsselung auf Spaltenebene in Transparent Data Encryption schützt sensible Daten wie Kreditkartennummern und Sozialversicherungsdaten in definierten Datenbankspalten *(siehe Abbildung 2)*.

Wie bei der Verschlüsselung auf Tablespace-Ebene kommt ein zweistufiges Schlüsselverwaltungssystem zum Einsatz, das eine nahtlose Ver- und Entschlüsselung gewährleistet. Die Schlüsselarchitektur sorgt für eine transparente Handhabung der Datensicherheit und bietet einen zuverlässigen Schutz für sensible Informationen.

Bei der Column Encryption ist der Prozess in Bezug auf die Schlüsselverwaltung ähnlich wie bei der Tablespace Encryption. Der TDE Master Encryption Key (MEK), der als Root der Verschlüsselungshierarchie dient, wird sicher in einem Keystore außerhalb der Datenbank gespeichert. Auf diese Weise wird der MEK vor unbefugtem Zugriff innerhalb der Datenbankumgebung geschützt. Allerdings erfordern einige Keystores, wie die Auto-Login-Wallet, zusätzliche Sicherheitsmaßnahmen. Dazu können externe

Keystores wie zum Beispiel Hardware-Lösungen beitragen, die in einem späteren Abschnitt näher erläutert werden.

Die Spaltenverschlüsselung arbeitet mit einem eindeutigen Table Key. Im Gegensatz zum MEK wird dieser Table Key in der Datenbank selbst gespeichert, und zwar im Data Dictionary. Um ihn zu schützen, wird jeder Table Key einzeln mit dem MEK verschlüsselt. *[3]*

Beschränkungen bei der Verwendung der TDE-Spaltenverschlüsselung

Transparent Data Encryption auf Spaltenebene arbeitet auf der SQL-Ebene der Oracle-Datenbank. Folglich sind alle Oracle-Datenbank-Utilities, die nicht die SQL-Ebene verwenden, mit der TDE-Spaltenverschlüsselung nicht kompatibel.

Beispiele für Inkompatibilität:
Transportable Tablespaces, externe Tabellen, DataPump Utility, und vieles mehr.

Die Verwendung der TDE-Spaltenverschlüsselung wird in Verbindung mit den folgenden Datenbankfunktionalitäten nicht empfohlen:

- Index-Typen mit Ausnahme von B-Tree
- Suchvorgänge, die Bereichsscans über einen Index durchführen
- Synchrone Änderungsdatenerfassung
- Transportable Tablespaces
- Spalten, die als Identitätsspalten bezeichnet werden *[4]*

Datentypen, die mit TDE Column Encryption verschlüsselt werden können

- BINARY_DOUBLE
- BINARY_FLOAT
- CHAR
- DATE
- INTERVAL DAY TO SECOND
- INTERVAL YEAR TO MONTH
- NCHAR
- NUMBER
- NVARCHAR2
- RAW (legacy or extended)
- TIMESTAMP
- VARCHAR2

Zur Verschlüsselung großer Binärobjekte (LOBs) steht Oracle SecureFiles zur Verfügung. Mit Oracle SecureFiles lassen sich LOB-Daten sicher speichern. Es ist jedoch zu beachten, dass externe Verweise, wie BFILEs, unverschlüsselt bleiben. *[5]*

Auswahl der richtigen Verschlüsselungsmethode

Die Wahl der Verschlüsselung ist eine wichtige Entscheidung und sollte sorgfältig überlegt werden.

Überlegungen zur Spaltenverschlüsselung:

- Die genaue Spalte der sensiblen Daten in der Tabelle ist bekannt.
- Im Allgemeinen eignet sich nur ein kleiner Prozentsatz, oft nur wenige Prozent, aller Anwendungsspalten für die Verschlüsselung.
- Datentyp und Länge werden von der TDE-Spaltenverschlüsselung unterstützt.
- Die Auswirkungen auf die Leistung werden durch den Anteil der verschlüsselten Spalten beeinflusst.
- Wie oft werden die verschlüsselten Werte ausgewählt oder die Größe der verschlüsselten Daten und andere Variablen aktualisiert?

Überlegungen zur Tablespace Verschlüsselung:

- Es ist nicht bekannt, welche Daten in den Tabellen sensibel sind.
- Der Großteil der Anwendungsdaten wird als sensibel eingestuft.
- Nicht alle Datentypen, die sensible Informationen enthalten, werden von der TDE-Spaltenverschlüsselung unterstützt.
- Leistung *[6]*

Keystores in Oracle

Die Oracle-Datenbank bietet Unterstützung für verschiedene Schlüsselverwaltungslösungen, darunter softwarebasierte Keystores wie Wallets und externe Optionen wie zum Beispiel Oracle Key Vault und Oracle Cloud Infrastructure (OCI) Key Management Service (KMS) und einige mehr.

Arten von TDE-Wallets:

- Auto-Login
  - Auto-Login-Wallets sind durch ein vom System generiertes Passwort gesichert. Sie werden automatisch geöffnet, wenn die Datenbank gestartet wird, und erfordern keinen manuellen Eingriff durch einen Sicherheitsadministrator. Daher benötigen wir zusätzliche Sicherheit, da die Anmeldung automatisch erfolgt und keine weitere Autorisierung erforderlich ist.
- Local Auto-Login
  - Hierbei handelt es sich um Auto-Login-Wallets, die an das spezifische System gebunden sind, auf dem sie erstellt wurden. Sie können nicht auf anderen Rechnern verwendet oder geöffnet werden. Dies bietet etwas mehr Sicherheit als Auto-Login, da es nur auf einem bestimmten System geöffnet werden kann, erfordert aber immer noch zusätzliche Sicherheitsmaßnahmen wie das Hardware-Sicherheitsmodul (HSM), das im nächsten Punkt beschrieben wird.
- Password-Protected
  - Passwortgeschützte Wallets erfordern zum Öffnen ein benutzerdefiniertes Passwort. Das Wallet muss manuell geöffnet werden, bevor auf die Verschlüsselungsschlüssel zugegriffen oder sie verwendet werden können. Aufgrund der manuellen Eingabe ist zwar kein direkter Autostart möglich, die Sicherheit ist dafür vollumfänglich gewährleistet.

Externe Schlüsselmanagement-Lösungen:

- Oracle Key Vault (OKV)
  - OKV ist eine softwarebasierte Appliance, die für Umgebungen entwickelt wurde, die hohe Verfügbarkeit und Skalierbarkeit erfordern. Sie unterstützt Clustering mit bis zu 16 Knoten und ist ideal für Oracle RAC, Oracle Data Guard, Exadata und mandantenfähige Setups.
- OCI Key Management Service (KMS)
  - OCI KMS ist ein Cloud-basiertes Schlüsselverwaltungssystem, das eine zentrale Kontrolle der Verschlüsselungsschlüssel für die in der Oracle Cloud Infrastructure gespeicherten Daten ermöglicht. *[7]*

Quellen

[1] Oracle, „Advanced Security Guide," 05 12 2024. [Online]. Available: *https://docs.oracle.com/en/database/oracle/oracle-database/21/asoag/introduction-to-transparent-data-encryption.html#GUID-62AA9447-FDCD-4A4C-B563-32DE04D55952.*

[2] Oracle, „Advanced Security Guide," Oracle, [Online]. Available: *https://docs.oracle.com/en/database/oracle/oracle-database/21/asoag/introduction-to-transparent-data-encryption.html#GUID-688B2CA5-EB00-4BEE-9486-9046670CCA70.*

[3] Oracle, „Advanced Security Guide," [Online]. Available: *https://docs.oracle.com/en/database/oracle/oracle-database/21/asoag/introduction-to-transparent-data-encryption.html#GUID-2D6C5B27-8E6A-4EF7-AABF-B0FB031C8374.*

[4] Oracle, „Advanced Security Guide," [Online]. Available: *https://docs.oracle.com/en/database/oracle/oracle-database/21/asoag/encrypting-columns-tables2.html#GUID-9A78E72B-D9D9-4BA2-BFEF-11C0060B6F91.*

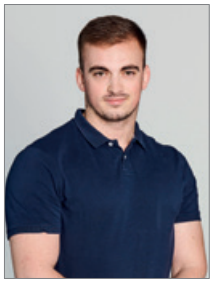
[5] Oracle, „Advanced Security Guide," [Online]. Available: *https://docs.oracle.com/en/database/oracle/oracle-database/21/asoag/encrypting-columns-tables2.html#GUID-E9DFE595-0119-4189-84AD-19FFE5683CC6.*

[6] Oracle, „Oracle Advanced Security Transparent Data Encryption Best Practices," [Online]. Available: *https://www.oracle.com/br/a/tech/docs/technical-resources/twp-transparent-data-encryption-best-practices.pdf.*

[7] Oracle, „Advanced Security Guide," [Online]. Available: *https://docs.oracle.com/en/database/oracle/oracle-database/21/asoag/introduction-to-transparent-data-encryption.html#GUID-2D6C5B27-8E6A-4EF7-AABF-B0FB031C8374.*

Über den Autor

Mein Name ist Meris Bihorac und ich bin ein 24-jähriger Oracle DBA aus Wien. Während meiner Ausbildung habe ich mich zunächst mehr auf die Entwicklung von Datenbanken konzentriert. Als ich jedoch die umfangreichen Features von Oracle entdeckte, wie Real Application Clusters (RAC), Data Guard und Hochverfügbarkeitslösungen, verlagerte sich mein Interesse deutlich in Richtung Administration. Diese neu entdeckte Faszination hat sich seitdem zu einer tiefen Leidenschaft für die Optimierung und Verwaltung von Datenbankumgebungen entwickelt. Daneben reizte mich schon in jungen Jahren die Herausforderung, nach Bugs zu suchen, insbesondere nach solchen, die die Web-Sicherheit betreffen, wie SQL-Injections. In meiner Freizeit habe ich als Bug- Hunter gearbeitet und dabei verschiedene bedeutende Schwachstellen aufgedeckt, darunter eine große Sicherheitslücke, von der über 200.000 Benutzer betroffen waren. Diese Erfahrungen haben meine berufliche Laufbahn nachhaltig geprägt und mein Engagement für hervorragende Leistungen im Bereich der Datenbankverwaltung weiter vorangetrieben.



Meris Bihorac
Meris.Bihorac@dbconcepts.com


DOAG
2025
DOAG
Datenbank
mit Cloud
Infrastructure
und
APEX
connect
14. und 15. Mai
Europa-Park Rust
Early Bird
bis 01.04.2025
datenbank.doag.org

Totgesagte leben länger – warum On-Prem Data Warehouses noch lange nicht am Ende sind

Daniel Eiduzzis, Infomotion

In Zeiten, in denen Cloud-Transformation fast schon als Allheilmittel für moderne Unternehmen propagiert wird, könnte man leicht den Eindruck gewinnen, dass klassische On-Premises-Lösungen auf dem Weg in die Bedeutungslosigkeit sind. Doch ein genauerer Blick zeigt: Das gilt nicht für alle Bereiche, und schon gar nicht für das Herzstück vieler Unternehmen – ihre Data- und Analytics-Infrastruktur.

Trotz des unaufhaltsamen Trends zur Cloud setzen immer noch zahlreiche Unternehmen bewusst auf Oracle On-Premises Data Warehouses. Der Grund: Diese Lösungen bieten oft nicht nur bewährte Stabilität, sondern auch spezifische Vorteile, die in der Cloud nicht immer realisierbar sind. In einer Welt, die von immer komplexeren Datenschutzanforderungen, branchenspezifischen Regularien und individuellen Leistungsanforderungen geprägt ist, spielt die lokale Datenhaltung weiterhin eine entscheidende Rolle.

Dieser Artikel beleuchtet, warum On-Prem Data Warehouses, allen Vorhersagen zum Trotz, noch lange nicht ausgedient haben und in bestimmten Szenarien nach wie vor die optimale Wahl für Unternehmen darstellen.

Die Evolution von Business Intelligence und Data Warehousing: Von den 1990ern bis heute

Die Geschichte von Business Intelligence (BI) und Data Warehousing hat sich seit den 1990er Jahren kontinuierlich weiterentwickelt und spiegelt den technologischen Wandel sowie die wachsenden Anforderungen der Unternehmen wider. *[1]* In den frühen 1990er Jahren begann die BI-Ära, als Unternehmen erkannten, dass sie ihre operativen Daten systematisch nutzen konnten, um fundierte Entscheidungen zu treffen. So wurden erste Data-Warehouse-Lösungen entwickelt, um Daten aus unterschiedlichen Quellsystemen zentral zu sammeln und in einer konsistenten Struktur für Analysezwecke bereitzustellen. Diese frühen Systeme basierten auf relationalen Datenbanken und OLAP-Technologien (Online Transactional Processing), die es ermöglichten, große Datenmengen zu verarbeiten und komplexe Abfragen durchzuführen. BI-Tools in Form von einfachen Dashboards und Berichten boten den Unternehmen einen Überblick über ihre Prozesse, doch die Technologie stieß hinsichtlich Performance und Skalierbarkeit schnell an ihre Grenzen.

Mit dem neuen Jahrtausend rückte die Integration von Datenquellen stärker in den Fokus. Unternehmen benötigten eine zentrale Plattform, auf der sie alle relevanten Geschäftsdaten konsolidieren konnten, was zur Entstehung von Enterprise Data Warehouses (EDW) führte. Diese fungierten als zentrale Datenhubs, die den Zugriff auf eine einheitliche Datenbasis ermöglichten. Entscheidende Fortschritte in der Datenintegration wurden durch die Verbreitung von ETL-Prozessen (Extract, Transform, Load) erzielt, die es ermöglichten, Daten aus verschiedenen Quellen zu extrahieren, zu transformieren und in das Data Warehouse zu laden. Parallel dazu entwickelten sich die BI-Lösungen weiter: Self-Service-Analysen und nutzerfreundliche Dashboards machten es möglich, dass auch Fachabteilungen ohne tiefgreifende IT-Kenntnisse eigene Auswertungen erstellen konnten. Diese Automatisierung und Benutzerfreundlichkeit trieben die Verbreitung von Data Warehousing in Unternehmen weiter voran.

Die 2010er Jahre brachten einen tiefgreifenden Wandel in der BI- und Data-Warehouse-Landschaft. Mit dem Aufkommen von Big Data mussten Unternehmen plötzlich riesige Datenmengen aus vielfältigen und oft unstrukturierten Quellen bewältigen, wie etwa Daten aus sozialen Netzwerken, IoT-Geräten und mobilen Anwendungen. Klassische Data Warehouses stießen dabei schnell an ihre Grenzen. Technologien wie Hadoop und NoSQL-Datenbanken kamen auf, die es ermöglichten, diese Datenmengen effizienter und flexibler zu verarbeiten. Gleichzeitig revolutionierte die Cloud das Data-Warehouse-Konzept: Cloud-basierte Lösungen wie Amazon Redshift, Google BigQuery und das Oracle Autonomous Data Warehouse ermöglichten es Unternehmen, skalierbare Infrastrukturen zu nutzen, ohne in teure Hardware investieren zu müssen. Diese Cloud-basierten Lösungen boten zudem eine höhere Flexibilität und Effizienz, da Kapazitäten je nach Bedarf angepasst und dynamisch bereitgestellt werden konnten. *[2]*

Heute befinden wir uns in einer Phase, in der hybride Ansätze im Data Warehousing an Bedeutung gewinnen. Viele Unternehmen setzen auf eine Kombination aus On-Premises- und Cloud-Lösungen, um das Beste aus beiden Welten zu vereinen: die Flexibilität und Skalierbarkeit der Cloud sowie die Kontrolle und Sicherheit der On-Premises-Lösungen. Insbesondere in stark regulierten Branchen oder dort, wo spezifische Datenschutzanforderungen gelten, spielen On-Premises Data Warehouses nach wie vor eine wichtige Rolle. Parallel dazu haben moderne BI-Systeme mit der Integration von Künstlicher Intelligenz (KI) und Machine Learning (ML) einen gewaltigen Sprung gemacht. Unternehmen können nun nicht nur vergangene Daten analysieren, sondern auch Vorhersagen treffen und potenzielle Entwicklungen simulieren. Die Geschwindigkeit und Präzision, mit der Entscheidungen getroffen werden, hat sich dramatisch erhöht, was vielen Unternehmen einen entscheidenden Wettbewerbsvorteil verschafft.

On-Prem-Architekturen sind tot. On-Prem war gestern, Cloud ist jetzt.

In der heutigen Zeit, in der die Cloud scheinbar allgegenwärtig ist, könnte man meinen, dass On-Premises-Architekturen ein Relikt der Vergangenheit sind. Der allgemeine Tenor lautet: „On-Prem ist tot." Aber was steckt hinter dieser Behauptung? Tatsächlich bieten Cloud-basierte Architekturen zahlreiche Vorteile, die viele Unternehmen dazu bewegen, ihre Infrastrukturen in die Cloud zu verlagern.

Ein großer Vorteil der Cloud ist die Skalierbarkeit. Während On-Prem-Lösungen häufig durch physische Hardware begrenzt sind, bieten Cloud-Dienste eine nahezu unbegrenzte Skalierbarkeit. Unternehmen können ihre Rechenleistung und Speicherressourcen dynamisch an ihre aktuellen Bedürfnisse anpassen, ohne in teure Hardware investieren zu müssen. Besonders bei stark schwankenden Workloads – etwa bei saisonalen Peaks oder unerwarteten Wachstumsschüben – ist diese Flexibilität ein entscheidender Vorteil. Ein paar Klicks genügen, um die Kapazität zu erhöhen oder wieder zu verringern, wenn die Nachfrage sinkt.

Auch die Kosteneffizienz spricht für die Cloud. Anstatt hohe Investitionen in Server, Softwarelizenzen und Wartungsverträge zu tätigen, bezahlen Unternehmen in der Cloud in der Regel nur für das, was sie tatsächlich nutzen. Dieses „Pay-as-you-go"-Modell ermöglicht eine präzise Kontrolle der IT-Kosten und reduziert unnötige Ausgaben. Für Unternehmen, die mit schwankenden Budgets arbeiten oder auf schnelle Skalierung angewiesen sind, ist dies ein enormer Vorteil gegen-

über den festen, oft hohen Kosten eines eigenen Rechenzentrums.

Darüber hinaus übernehmen Cloud-Anbieter die Wartung und Aktualisierung der gesamten Infrastruktur. Dies spart Unternehmen wertvolle Zeit und Ressourcen, die sie stattdessen in strategischere Aufgaben investieren können. Während in der Vergangenheit IT-Teams regelmäßig mit Software-Updates, Hardware-Austausch und Sicherheits-Patches beschäftigt waren, übernimmt nun der Cloud-Anbieter diese Aufgaben im Hintergrund. Dadurch werden nicht nur Kosten gesenkt, sondern auch die Risiken von Ausfallzeiten minimiert.

Ein weiterer Punkt ist die Zugänglichkeit. Cloud-basierte Lösungen ermöglichen es Mitarbeitern, von jedem Ort der Welt aus auf Unternehmensdaten und Anwendungen zuzugreifen, solange sie eine Internetverbindung haben. Dies fördert nicht nur die Zusammenarbeit in globalen Teams, sondern unterstützt auch den Trend zum flexiblen Arbeiten, der insbesondere durch die Pandemie an Bedeutung gewonnen hat. Ob im Büro, im Homeoffice oder auf Geschäftsreise – die Cloud macht es einfach, immer und überall auf relevante Daten zuzugreifen.

Hinzu kommt, dass die Cloud die Innovation vorantreibt. Viele Anbieter integrieren fortschrittliche Technologien wie Künstliche Intelligenz und maschinelles Lernen direkt in ihre Plattformen. Unternehmen, die Cloud-Dienste nutzen, können somit auf diese Technologien zugreifen, ohne sie selbst entwickeln oder teuer einkaufen zu müssen. Dies beschleunigt die Implementierung neuer Anwendungen und Prozesse und sorgt dafür, dass Unternehmen wettbewerbsfähig bleiben.

Auch in puncto Disaster Recovery bietet die Cloud erhebliche Vorteile. Die meisten Cloud-Anbieter verfügen über umfassende Disaster-Recovery-Pläne, die darauf ausgelegt sind, Datenverlust zu minimieren und die Betriebszeit zu maximieren. Daten werden regelmäßig gesichert und auf mehreren, geografisch verteilten Servern gespeichert, sodass Unternehmen auch im Fall einer Katastrophe schnell wieder auf ihre Daten zugreifen können.

Nicht zuletzt ist die Cloud auch ein Schritt in Richtung Umweltfreundlichkeit. Da Rechenzentren von Cloud-Anbietern oft weitaus effizienter arbeiten als lokale Serverräume, können Unternehmen ihren CO2-Fußabdruck durch die Nutzung dieser Dienste erheblich reduzieren. Durch die gemeinsame und oftsmals optimierte Nutzung von Ressourcen wird der Energieverbrauch gesenkt, was sowohl der Umwelt als auch den Betriebskosten zugutekommt.

Es wird jedoch oft übersehen, dass die Migration in die Cloud nicht ohne Herausforderungen ist. Eine solche Umstellung muss gut geplant und durchgeführt werden, um Sicherheitsrisiken, mögliche Datenverluste und Integrationsprobleme zu vermeiden. Eine vorschnelle oder schlecht durchdachte Migration kann zu erheblichen Problemen führen. Deshalb ist es wichtig, den Übergang sorgfältig zu planen und alle relevanten Faktoren zu berücksichtigen.

Trotz der vielen Vorteile der Cloud ist die Aussage „On-Prem Architekturen sind tot." aber nicht uneingeschränkt korrekt. In vielen Fällen bietet die Cloud zwar klare Vorzüge, aber On-Premises-Lösungen haben nach wie vor ihren Platz.

Totgesagte leben länger. Es gibt gute Gründe, warum Kunden nicht in die Cloud migrieren

Die Vorstellung, dass On-Premises-Lösungen tot seien, ist nicht nur verfrüht, sondern ignoriert die spezifischen Anforderungen vieler Branchen und Geschäftsmodelle. Tatsächlich gibt es gewichtige Argumente, warum Unternehmen nicht in die Cloud migrieren wollen – und in einigen Fällen auch bewusst davon Abstand nehmen sollten.

Ein zentrales Thema ist die Datensicherheit. Viele Unternehmen haben Bedenken, wenn es darum geht, ihre sensiblen Daten in die Cloud zu verlagern. Das Gefühl, die Kontrolle über die eigenen Daten zu verlieren, lässt sie zögern. Indem sie ihre Daten lokal auf eigenen Servern speichern, behalten sie die volle Kontrolle über Sicherheitsprotokolle, Zugriffsrechte und Verschlüsselungsstandards. Besonders in Fällen, in denen vertrauliche Kundendaten oder geistiges Eigentum betroffen sind, gibt es oft die Befürchtung, dass ein externer Cloud-Anbieter diese Daten nicht mit der gleichen Sorgfalt behandelt wie ein unternehmensinternes IT-Team. Die Vorstellung, dass Daten in der Cloud einem größeren Risiko ausgesetzt sind, ist besonders in sicherheitskritischen Branchen nach wie vor stark verbreitet.

Ein weiterer wichtiger Aspekt ist Compliance und Regulierung. In vielen Branchen unterliegen Unternehmen strengen gesetzlichen Vorgaben bezüglich der Speicherung und Verarbeitung von Daten. Finanzdienstleister, das Gesundheitswesen oder Unternehmen in der öffentlichen Verwaltung müssen strikte Regeln einhalten, um den Schutz personenbezogener Daten zu gewährleisten. In solchen Fällen kann eine On-Premise-Lösung dabei helfen, sicherzustellen, dass alle regulatorischen Anforderungen erfüllt werden. So lassen sich etwa geografische Beschränkungen zur Datenhaltung oder spezielle Sicherheitsstandards einfacher umsetzen, wenn die Daten physisch vor Ort bleiben. Die Cloud mag in vielen Fällen flexibel sein, doch das Vertrauen in eine selbst verwaltete Infrastruktur bleibt in diesen sensiblen Bereichen oft ungebrochen.

Leistung ist ein weiterer entscheidender Faktor, der Unternehmen davon abhält, vollständig in die Cloud zu migrieren. Bei Anwendungen, die hohe Datenverarbeitungsgeschwindigkeiten oder geringe Latenzzeiten erfordern, kann eine lokale On-Premises-Infrastruktur oft bessere Ergebnisse liefern als Cloud-Lösungen. Echtzeitanwendungen oder datenintensive Workloads profitieren von der direkten Nähe zwischen den Daten und den Verarbeitungsressourcen vor Ort. Für Unternehmen, die auf hochperformante Systeme angewiesen sind, wie beispielsweise in der Finanzindustrie oder in der Fertigung, kann eine On-Premise-Lösung schlichtweg die bessere Option sein.

Auch die Kosten spielen eine Rolle. Während Cloud-Lösungen oft als kostengünstiger beworben werden, gilt dies nicht uneingeschränkt. Besonders wenn es um den Transfer und die Speicherung großer Datenmengen geht, können die Kosten für Cloud-Dienste rasch ansteigen. Die laufenden Gebühren für Datenübertragungen, Speichernutzung und zusätzliche Dienste können sich schnell summieren. Für Unternehmen, die riesige Datenmengen täglich generieren und verarbeiten, kann es kosteneffizienter sein, diese Daten lokal zu speichern und

zu verarbeiten, anstatt sie kontinuierlich in die Cloud zu übertragen.

Ein weiterer Vorteil von On-Premise-Lösungen ist die Anpassungsfähigkeit. Viele Unternehmen haben sehr spezifische Anforderungen, die sich nicht immer problemlos in standardisierten Cloud-Umgebungen abbilden lassen. Eine lokal gehostete Infrastruktur bietet mehr Möglichkeiten, um Systeme und Anwendungen an individuelle Geschäftsbedürfnisse anzupassen. Unternehmen können maßgeschneiderte Lösungen entwickeln, die sich besser in ihre bestehenden Prozesse integrieren lassen und die speziellen Anforderungen ihrer Branche berücksichtigen. Diese Flexibilität ist oft entscheidend, um einen Wettbewerbsvorteil zu sichern.

Totgesagte leben länger. Auch wenn die Cloud viele Vorteile bietet, gibt es zahlreiche gute Gründe, warum Unternehmen weiterhin auf On-Premise-Architekturen setzen. Sei es aufgrund von Sicherheitsbedenken, strengen regulatorischen Anforderungen, Leistungsanforderungen oder der Kostenstruktur – On-Premise-Lösungen bleiben für viele Unternehmen nach wie vor die bevorzugte Wahl.

Aus der Praxis – Kreditinstitut aus dem deutschsprachigen Raum

Ein führendes Kreditinstitut aus dem deutschsprachigen Raum entschied sich bewusst gegen eine vollständige Migration in die Cloud und setzt weiterhin auf eine On-Premise-Dateninfrastruktur. Diese Entscheidung beruht auf mehreren zentralen Faktoren, die für das Unternehmen von entscheidender Bedeutung sind.

Ein Hauptaspekt ist die Datensicherheit. Wie viele andere Unternehmen in der Finanzbranche, hat das Kreditinstitut Bedenken, sensible Kundendaten und interne Geschäftsinformationen in die Cloud zu verlagern. Die Kontrolle über die Daten bleibt ein zentrales Anliegen. Solange die Daten auf eigenen Servern gespeichert werden, kann das Unternehmen selbst bestimmen, wer Zugang hat und welche Sicherheitsmaßnahmen getroffen werden. Für das Institut bedeutet dies, dass potenzielle Risiken, die mit der Auslagerung von Daten an externe Dienstleister verbunden sind, vermieden werden können.

Die Einhaltung von Compliance und regulatorischen Anforderungen ist ein weiterer zentraler Aspekt. Besonders in der Finanzbranche gibt es strenge gesetzliche Vorgaben, vor allem im Bereich Datenschutz und bei der Speicherung sensibler Finanzdaten. Für das Kreditinstitut war die Wahl einer On-Premise-Lösung daher eine bewusste Entscheidung, um die Einhaltung dieser Vorschriften zu erleichtern. Zwar bieten viele Cloud-Anbieter ebenfalls Möglichkeiten, gesetzliche Anforderungen zu erfüllen, doch das Unternehmen bevorzugte es, den gesamten Prozess intern zu kontrollieren. So behält es nicht nur die volle Kontrolle über die Daten, sondern kann auch die spezifischen Datenschutzanforderungen, wie etwa die geografische Lage der Datenhaltung, gezielter umsetzen.

Ein weiterer wichtiger Faktor ist die Leistung der On-Premise-Infrastruktur. Im Finanzwesen ist Geschwindigkeit oft ein entscheidender Wettbewerbsfaktor. Anwendungen mit hohem Datendurchsatz und minimaler Latenz sind für das Kreditinstitut von enormer Bedeutung, etwa im Zahlungsverkehr oder bei Börsentransaktionen. On-Premise-Lösungen ermöglichen es dem Institut, die Datenverarbeitung direkt vor Ort abzuwickeln, ohne dass es zu Verzögerungen kommt, die bei der Nutzung einer Cloud-basierten Infrastruktur möglicherweise entstehen könnten. Diese Vorteile in der Performance tragen maßgeblich dazu bei, dass das Unternehmen weiterhin auf lokale Rechenkapazitäten setzt, um den hohen Anforderungen im Tagesgeschäft gerecht zu werden.

Zusammenfassend lässt sich sagen, dass dieses Kreditinstitut aus dem deutschsprachigen Raum gute Gründe hat, auf eine On-Premise-Lösung zu setzen. Datensicherheit, Compliance und eine herausragende Leistung in der Datenverarbeitung sind Schlüsselfaktoren, die die Entscheidung gegen die Cloud und für eine lokal betriebene Infrastruktur maßgeblich beeinflusst haben. Die Praxis zeigt, dass trotz des Trends zur Cloud immer noch zahlreiche Unternehmen auf bewährte On-Premise-Lösungen vertrauen, um ihren spezifischen Anforderungen gerecht zu werden.

Quintessenz

Die Diskussion über Cloud versus On-Premise ist nicht schwarz-weiß. Während Cloud-Lösungen in vielen Bereichen klare Vorteile bieten, zeigen Praxisbeispiele wie das des Kreditinstituts aus dem deutschsprachigen Raum, dass On-Premise-Lösungen nach wie vor eine wichtige Rolle spielen. Sicherheit, Compliance, Leistung und Kosten sind Schlüsselfaktoren, die Unternehmen dazu bewegen, weiterhin auf lokale Architekturen zu setzen. Die Entscheidung für eine Architektur hängt letztlich von den spezifischen Anforderungen, dem Risikoappetit und den Geschäftsmodellen der Unternehmen ab. Totgesagte leben länger – und On-Premise-Lösungen werden auch in Zukunft ihren Platz in der IT-Landschaft behaupten.

Quellen

[1] Turban, Efraim, et al. (2017): Business Intelligence: A Managerial Perspective on Analytics. Pearson India.

[2] Biere, Mike (2010): The New Era of Enterprise Business Intelligence: Using Analytics to Achieve a Global Competitive Advantage. O'Reilly Media.

Über den Autor

Daniel Eiduzzis ist Partner und Business Development Manager beim Beratungsunternehmen INFOMOTION und seit über 20 Jahren in der Business-Intelligence-Branche tätig. Er engagiert sich im TDWI, ist Mitglied des Fachbeirats des BI-SPEKTRUMS und als TDWI-Expert auf die Themen BI-Organisation, Data Governance und BI-Architektur spezialisiert. Als Autor und Sprecher teilt er sein Fachwissen regelmäßig auf Fachveranstaltungen und in unterschiedlichen Publikationen. Daniel Eiduzzis lebt mit seiner Frau und zwei Kindern im Hamburger Umland.

Daniel Eiduzzis
rdaniel.eiduzzis@infomotion.de

Von der Strategie zur Lösung – Multicloud als Treiber für moderne Umgebungen

Stefan Seck, Logicalis

Eine Multicloud-Strategie bietet Unternehmen zahlreiche Vorteile, da sie eine flexiblere, widerstandsfähigere und kosteneffizientere Nutzung von Cloud-Diensten ermöglicht. Zu den wesentlichen Gründen, warum eine Multicloud-Strategie für Unternehmen vorteilhaft ist, zählen etwa die Optimierung der IT-Architektur oder eine Kostenoptimierung. Unternehmen, die Performance und Security in ihre Cloud-Strategie integrieren, sind in der Lage, besser auf Risiken und auf Veränderungen im Markt zu reagieren, innovative Lösungen zu entwickeln und langfristig erfolgreich zu bleiben.

Cloud first – oder was ist die Strategie?

Bei der Diskussion über Strategien, wie beispielsweise dem „Cloud-First"-Ansatz oder bei der kategorischen Ablehnung von Public-Cloud-Diensten, werden oft sehr unterschiedliche Positionen vertreten. Dennoch zeigt sich, dass viele Wege auch in Richtung Multicloud gehen, da dieses Thema in den letzten Monaten besonders präsent war und ist.

Da es in Bezug auf Multicloud keinen universellen Ansatz gibt, der für alle Situationen geeignet ist, ist es wichtig, die

spezifischen Anforderungen der Kunden zu verstehen.

Bei den von mir durchgeführten Kundenterminen geht es nicht ausschließlich um konkrete Anforderungen, sondern häufig auch um die Ausrichtung der IT. Zu den wichtigen Themen gehören die Datenspeicherung sowie beispielsweise agile Entwicklung, Kosteneffizienz und eine zukunftssichere Infrastruktur.
In erster Linie werden diese vier Punkte thematisiert.

- Sinnvolle Nutzung der Lizenzen

Im Rahmen der Entwicklung einer Cloudstrategie ist es von entscheidender Bedeutung, die vorhandenen Lizenzen zu analysieren und deren Eignung für den Einsatz in Cloudprojekten zu ermitteln.

Viele Cloud-Anbieter unterstützen das Bring-Your-Own-License-Modell (BYOL), bei dem bestehende Lizenzen für Softwareprodukte in der Cloud genutzt werden können, anstatt neue Lizenzen zu kaufen. Es gibt auch Software-as-a-Service (SaaS) und andere Cloud-basierte Abonnementmodelle, die eine hohe Flexibilität bieten und in der Regel keine dauerhaften Lizenzkäufe erfordern. Diese Flexibilität ermöglicht es Unternehmen, Lizenzen bedarfsgerecht zu erwerben und die Lizenzierung an die tatsächliche Auslastung anzupassen.

Die effiziente Nutzung von Lizenzen in der Cloud erfordert eine gründliche Planung und Überwachung. Mit verschiedenen Lizenzierungsmodellen, dem Einsatz von Automatisierungs- und Tracking-Tools sowie Cloud-spezifischen Optionen können Unternehmen ihre Lizenzkosten optimieren und gleichzeitig sicherstellen, dass die Cloud-Umgebung den rechtlichen und betrieblichen Anforderungen entspricht. So maximieren sie den Nutzen ihrer bestehenden Lizenzen und vermeiden unnötige Ausgaben.

- Optimierung der Kosten

FinOps (Financial Operations) ist ein wesentlicher Bestandteil einer Multi-Cloud-Strategie, da es Unternehmen dabei unterstützt, ihre Ausgaben über mehrere Cloud-Anbieter hinweg zu optimieren und zu kontrollieren. In einer Multi-Cloud-Umgebung, in der Unternehmen verschiedene Cloud-Anbieter für unterschiedliche Anforderungen nutzen, entstehen oft komplexe Kostenstrukturen. FinOps bietet die notwendigen Prozesse, um Transparenz zu schaffen, Ausgaben zu überwachen und sicherzustellen, dass Cloud-Ressourcen effizient genutzt werden.

Vor allem durch die Kostenkontrolle sowie die Transparenz und Optimierung von Cloud-Ressourcen ist FinOps – und damit die Kostenoptimierung – ein wichtiger Bestandteil einer Multicloud-Strategie.

Mit FinOps stellen Unternehmen sicher, dass alle relevanten Informationen zu Cloud-Ausgaben in Echtzeit und klar strukturiert zur Verfügung stehen. So verstehen sie genau, wie Ressourcen genutzt werden und welche Kosten entstehen. Diese Transparenz hilft und sie stellt sicher, dass verschiedene Preismodelle, Rabatte und Optionen genutzt werden, um die richtigen finanziellen Entscheidungen zu treffen. Durch die kontinuierliche Überwachung und Analyse lässt sich leicht feststellen, ob die getätigten Ausgaben angemessen sind oder ob es Möglichkeiten zur Kostensenkung gibt, etwa durch die Kündigung ungenutzter Ressourcen oder den Wechsel zu kostengünstigeren Optionen.

- Anforderungen an Performance

Eine hohe Performance garantiert schnelle Ladezeiten und eine stabile Verfügbarkeit von Applikationen und Services.

Eine effektive Strategie in der Cloud muss die Fähigkeit zur dynamischen und schnellen Skalierung der Ressourcen berücksichtigen. Insbesondere bei plötzlichen Spitzen des Traffics, wie sie beispielsweise während des Jahresabschlusses oder einer Marketingkampagne auftreten können, ist es essenziell, dass die Cloud-Infrastruktur in der Lage ist, zeitnah zusätzliche Kapazitäten bereitzustellen, um die Leistung aufrechtzuerhalten.

Eine optimierte Performance kann Ressourcen effizienter nutzen, beispielsweise durch eine Workload-Verteilung, die die Cloud-Infrastruktur vollständig auslastet, ohne unnötige Ressourcen zu verschwenden. Dies führt zu einer besseren Kostenkontrolle und eine nachhaltigere Nutzung der Ressourcen.

In Anbetracht dessen ist die Performance ein entscheidender Faktor in der Gestaltung einer Cloud-Strategie, da sie die Benutzererfahrung, die Kostenstruktur und die Flexibilität der IT-Infrastruktur maßgeblich beeinflusst. Eine sorgfäl-

tig optimierte Performance trägt somit nicht nur zur Effizienzsteigerung und Kostensenkung bei, sondern auch zur Steigerung der Wettbewerbsfähigkeit, Skalierbarkeit und Geschäftskontinuität.

- Security

In diesem Zusammenhang ist vor allem auch der Aspekt der Sicherheit zu berücksichtigen, der einen weiteren zentralen Bestandteil einer Cloud-Strategie darstellt. Die Kunden legen großen Wert auf die Sicherheit der Datenverwaltung. Unternehmen speichern häufig vertrauliche und kritische Daten in der Cloud, weshalb eine starke Sicherheitsstrategie erforderlich ist, um diese Daten vor unbefugtem Zugriff, Diebstahl und Datenverlust zu schützen. Dies ist auch im Hinblick auf die regulatorischen Anforderungen bezüglich des Datenschutzes und der Datensicherheit von großer Bedeutung. Effektive Sicherheitsmaßnahmen wie Firewalls und Verschlüsselung sind in diesem Zusammenhang von entscheidender Bedeutung und eine entsprechende Sicherheitsüberwachung ist notwendig, um Angriffe abzuwehren und Sicherheitsverletzungen zu verhindern.

Die Sicherstellung von Backup- und Wiederherstellungsprozessen ist dabei von essenzieller Bedeutung. Ferner können Notfallpläne Unternehmen dazu verhelfen, im Falle eines Angriffs oder einer Sicherheitsverletzung eine zeitnahe Wiederherstellung der Funktionalität zu gewährleisten.

Eine sorgfältig durchdachte Sicherheitsstrategie trägt somit zur Risikominimierung, zum Schutz des Geschäftsbetriebs und zur Stärkung des Kundenvertrauens bei.

Multicloud Enabler

Ich finde, Kevin Bogusch (Oracle) hat es sehr gut zusammengefasst:

„The question is no longer which cloud to use, but which clouds to use, to gain the most value."

Für eine Vielzahl von Betrieben stellt Cloud eine Option für die Erneuerung ihrer IT-Infrastruktur dar. Es ist jedoch zu erwarten, dass es weiterhin Workloads geben wird, die aus verschiedenen Gründen (Compliance-Anforderungen, Anforderungen an geringe Latenzen und vieles mehr) nicht in der Cloud betrieben werden können oder dürfen. Zudem bringen jede Branche und jedes Unternehmen spezifische Anforderungen an die eigene IT und an externe Betreiber, wie beispielsweise Cloud-Anbieter, mit sich.

Um es plakativ zu formulieren:
Ein Beispiel sind Banken, die schnelle, insbesondere „latenzarme" Zugriffe benötigen. Industrieunternehmen hingegen bedürfen womöglich Dienste, um ihre verteilten Standorte zentral zu steuern oder zu überwachen. Dem gegenüber steht jedoch die Tatsache, dass nicht jede Cloud beziehungsweise nicht jeder Hyperscaler alle Funktionalitäten bietet, die von den verschiedenen Branchen eingesetzt werden.

Es ist daher erforderlich, zu prüfen, welche Services benötigt werden, um die Applikationen, die On-Premises laufen, auch in der Cloud zu betreiben.

Aus dieser Perspektive ist es ratsam, die Themen zu identifizieren, die für Kunden von Relevanz sind, und insbesondere jene zu analysieren, die als Enabler für Multicloud-Anwendungen dienen könnten.

Im Kontext von Multicloud-Strategien werden regelmäßig die Themen Datensicherheit und Interoperabilität adressiert. Die Migration von Daten in die Cloud erfordert die reibungslose Übertragung zwischen verschiedenen Cloud-Anbietern, um eine hohe Flexibilität zu gewährleisten und das Risiko eines Vendor-Lockins zu minimieren.

Ein zentrales Ziel von Multicloud-Projekten ist die Optimierung der Leistung. Hierbei stehen die Reduzierung der Latenzzeiten und die Steigerung der IOPS im Vordergrund. Die Nutzung verschiedener Cloud-Anbieter ermöglicht es Unternehmen, ihre Leistung zu optimieren und gleichzeitig die Verfügbarkeit ihrer Dienste zu verbessern.

Ein weiterer entscheidender Vorteil der Multicloud-Strategie liegt in der Bereitstellung von Desaster-Recovery-Szenarien mittels skalierbarer Backups und der Steigerung der Verfügbarkeit von Services. Die Sicherstellung der Kompatibilität zwischen verschiedenen Cloud-Plattformen ist dabei von entscheidender Bedeutung, um eine reibungslose Übertragung von Daten und Anwendungen zu gewährleisten.

Die Vermeidung einer Vendor-Lock-in-Situation ermöglicht es Unternehmen, den Cloud-Anbieter zu wechseln und auf die besten Angebote und Technologien zuzugreifen.

Was können Erfolgsfaktoren für eine Cloud- oder Multicloud-Strategie sein?

Die Skalierbarkeit von Systemen in der Cloud auf Abruf ermöglicht eine flexible Reaktion auf unterschiedliche Situationen, was einen wesentlichen Erfolgsfaktor darstellt. Zudem besteht die Möglichkeit einer positiven Beeinflussung der Performance durch diese Unterstützung. Ein vereinfachtes Management in der Cloud führt zu einer Reduktion des eigenen Administrationsaufwands. Die Standardisierung der Systeme ist eine weitere wesentliche Voraussetzung für eine funktionierende Cloud-Strategie. Durch die Konsolidierung der IT-Infrastruktur können Unternehmen ihre Ressourcen auf das Kerngeschäft fokussieren und so ihre Agilität steigern. Dabei darf der Aspekt der Datensicherheit nicht vernachlässigt werden. Es ist essenziell, die Daten vor unberechtigtem Zugriff zu schützen. Dies erfordert eine umfassende Sicherheitsarchitektur, die unter anderem Security Lists und Netzwerkisolation für die Zugriffswege umfasst. Zudem ist eine Verschlüsselung der Daten sowohl bei der Speicherung als auch bei der Übertragung erforderlich.

Es ist evident, dass sich ein signifikanter Anteil von Unternehmen (>90 %) mit der Thematik der Cloud- und Multicloud-Strategien auseinandersetzt beziehungsweise bereits Cloud-basierte Applikationen, Datenbanken und Systeme einsetzt.

Dennoch kann, wie bereits erwähnt, nicht jeder Workload in eine Public Cloud migriert werden. Ziel einer Strategiediskussion besteht daher darin, den stetigen Veränderungen und mitunter agilen Strukturen in Unternehmen Rechnung zu tragen. Dies kann durch eine Reduktion von Kosten sowie der Aufhebung von Abhängigkeiten durch das Verteilen von Infrastruktur, Anwendungen und Services erreicht werden. Hier lässt sich gut darüber diskutieren, was Multicloud ge-

nau bedeutet. Ist die Nutzung von den für den Workload passenden Services bei unterschiedlichen Cloudanbietern ein Multicloudansatz? Damit wird sichergestellt, dass von Anbietern der jeweils beste Service genutzt wird. Oder geht es bei Multicloud darum, dass Kunden sich nicht von einem Cloudanbieter abhängig machen möchten, also den sogenannten Vendor-Lock-in vermeiden wollen? Egal, für welche Lösung Unternehmen sich entscheiden – eine passende Strategie ist in jedem Fall nötig.

Ein erfolgreicher Weg in die Multicloud

Die Studie „Multicloud in the Mainstream", durchgeführt von 451 Research *[1]*, beleuchtet die Relevanz von Multicloud-Lösungen. Die Analyse zukünftiger Anwendungsbereiche identifiziert Themen wie Datenredundanz und Backup als prioritäre Aspekte. Zudem gewinnen Workload und Data Mobility an Bedeutung, da Unternehmen die Gefahren für ihr Geschäft und ihre Daten minimieren möchten. Aus diesem Grund erwägen sie die Implementierung von Multicloud-Lösungen, um einerseits Risiken zu minimieren und andererseits die Kosten durch eine Verteilung in mehrere Clouds zu optimieren. Basierend auf den genannten Use Cases ergeben sich für die befragten IT-Führungskräfte auch die entsprechenden Erfolgsfaktoren. Von besonderem Interesse sind dabei drei Punkte, da sie sich wie ein roter Faden durch die gesamte Analyse ziehen.

Multicloud ist erfolgreich, wenn

- … die Kosten optimiert sind.
- … weiterhin ein Zusammenspiel mit On-Premises-Daten möglich ist.
- … Daten entsprechend sicher, also vor falschen Zugriffen geschützt sind.

Die Studie *[1]* zeigt eindeutig: Daten, Souveränität und Kostenoptimierung sind für über 40 % der befragten Unternehmen von entscheidender Bedeutung.

Agilität und Innovation stehen mit 30 % an zweiter Stelle und sind damit ebenfalls von entscheidender Bedeutung.

All diese Themen sind bei On-Premises-Projekten ebenso relevant wie bei Cloud-Projekten.

Im Rahmen der Oracle Cloud World wurde vielfach die Notwendigkeit betont, dass die Clouds enger zusammenarbeiten müssen. Larry Ellison bezeichnete dies im Jahr 2022 als **„Garden Walls tumbling down"**.

Diese Entwicklung ebnet den Weg für ein Internet of Clouds. Als Oracle-Partner liegt unsere Aufgabe darin, uns in die Kunden hineinzuversetzen, ihre Anliegen zu verstehen und die Herausforderungen, mit denen sie konfrontiert sind, zu identifizieren und den Weg mit ihnen zu gehen.

Die Oracle Multicloud Services (Oracle Database@Azure, Oracle Database@Google Cloud und Oracle Database@AWS) bieten Unternehmen die Möglichkeit, ihre Workloads und Daten über verschiedene Cloud-Plattformen hinweg zu verwalten und zu nutzen. Dadurch können Anwendungen und Datenbanken in unterschiedlichen Cloud-Umgebungen wie AWS, Google Cloud und Microsoft Azure genutzt werden. Diese Vorgehensweise erlaubt es Unternehmen, die jeweils bestmögliche Lösung für die eigenen Anforderungen zu identifizieren und die Skalierbarkeit zu maximieren.

Die Nutzung von Oracle-Datenbankdiensten in diversen Clouds kann zu einer hohen Leistung, Verfügbarkeit und Skalierbarkeit führen, was insbesondere für datenintensive Anwendungen und Geschäftsanwendungen von Relevanz ist.

Ein weiterer Vorteil besteht in der Risikoreduzierung eines Vendor-Lock-ins, wodurch Unternehmen flexibel auf neue Technologien und Anbieter reagieren können.

Zudem bieten diese Services robuste Sicherheitsmaßnahmen und helfen Unternehmen dabei, gesetzliche und regulatorische Anforderungen zu erfüllen.

Bei Logicalis unterstütze ich unsere Account Manager dabei, die Herausforderungen zu verstehen, mit denen Unternehmen konfrontiert sind. Ich gehe dabei zunächst hersteller- und technologieagnostisch an die offenen Fragen. Die Hauptfrage ist: Wie können sich unsere Kunden weiterentwickeln und ihre Daten effektiv nutzen?

Deshalb liegt der Schwerpunkt auf der Entwicklung einer umsetzbaren Lösung. Im Anschluss erfolgt die Gestaltung des Designs oder der Architektur der Lösung unter Berücksichtigung geeigneter Technologien und Hersteller.

Den Weg in die Multicloud erfolgreich zu gehen, ist eine faszinierende Herausforderung, die mit guten Konzepten und Überlegungen zu einer absoluten Erfolgsgeschichte werden kann.

Quellen

[1] *https://www.oracle.com/a/ocom/docs/gated/451-research-multi-cloud-in-the-mainstream.pdf*

Über den Autor

Stefan Seck ist Manager Database & Engineered Systems bei der Logicalis GmbH und Oracle ACE Associate. Er verfügt über eine mehr als 25-jährige Expertise im Bereich Oracle-Datenbanken. Im Laufe seiner Karriere sammelte er umfassende Expertise in den Bereichen HA- und Migrations/Upgrade-Projekten, wobei er sich insbesondere mit Engineered Systemen (Exadata, ODA und RA) befasste. Ein weiterer Schwerpunkt seiner Tätigkeit liegt in der Umsetzung von Projekten mit Exadata Cloud@Customer und Exadata Cloud Services, mit dem Ziel, den Kunden optimale Lösungen zu bieten.

Stefan Seck
stefan.seck@logicalis.de

Datenqualität – was gibt es Neues in der Oracle Datenbank 23ai?

Detlef E. Schröder, Oracle Deutschland

In diesem Artikel werden Funktionen vorgestellt, die helfen können, die Datenqualität zu verbessern und die in der neuen Datenbankversion 23ai von Oracle vorhanden sind. Datenqualität ist allerdings ein weites Feld und wird daher auch zuerst weiter definiert und beschrieben. Einige praktische Beispiele sind in Skriptform eingebettet.

Datenqualität ist ein leidiges Thema. Jeder braucht sie, aber sie zu erreichen, bedeutet nur Arbeit. Nicht nur in den Applikationen, in denen die Qualität so weit gesichert wird, wie es nötig ist, ist Datenqualität notwendig, sondern auch in allen nachgelagerten Anwendungen. Egal ob Warehouse oder BI, ohne „saubere" Daten macht die schönste Grafik keinen Sinn.

Dabei ist Datenqualität ein weites Feld und umfasst viele Aspekte. In einer Arbeit definierten Wang und Strong *[1]* schon im Jahr 1996 vier Aspekte von Datenqualität, die sich dann auf 20 Dimensionen dieser Aspekte aufteilen.

Viele dieser Dimensionen haben mit der Datenbank nichts zu tun, aber viel mit den Prozessen, aus denen diese Daten hervorgehen. Durchaus spielt aber bei einigen dieser Dimensionen auch die Verwendung eine große Rolle. Es lohnt sich also, sein Warehouse oder die Grundlage für die Analysen auf diese Aspekte hin zu überprüfen. Vieles hält dieser Prüfung nicht stand. Oft wird auch nur auf bestimmte Aspekte geachtet, zum Beispiel auf die Vollständigkeit, aber andere, wie die notwendige Aktualität, werden vernachlässigt. Wenden wir uns aber nun den Aspekten zu, die wir in der Datenbank unterstützen können.

Kategorie	Dimension
Accuracy	Believability
	Accuracy
	Objectivity
	Completness
	Traceability
	Reputation
	Variaty of Data Sources
Relevancy	Value-Added
	Relevancy
	Timeliness
	Ease of Operation
	Appropiate Amount of Data
	Flexibility
Representation	Interpretebilty
	Ease of Understanding
	Representational Consistency
	Concise Representation
Accessibility	Accessibility
	Cost-effectiveness
	Acess Security

Tabelle 1 – Aspekte von Datenqualität

Datenqualität in der Datenbank sichern

Wenn innerhalb von ETL (Extraktion Transformation und Laden) Datenqualität gesichert werden soll, können wir 6 Prüfkategorien ausmachen, die wir in der Oracle- Datenbank auch mengenorientiert sichern können.

1. Attributbezogen (u. a. Formate, Wertbereiche)
2. Satzbezogen (Abhängigkeiten zwischen Attributen eines Datensatzes)
3. Satzübergreifend (u. a. Primary Key, rekursive Zusammenhänge, Muster)
4. Tabellenübergreifend (u. a. Foreign Key, Referenzen)
5. Zeitabhängend (u. a. Zeitinvariante Inhalte, zeitliche Muster)
6. Verteilungs-/Mengenbezogen (Verteilungen, Mengen)
7. Dazu ist es notwendig, die zu erwartenden Ergebnisse im Vorhinein zu definieren und auch aus den Anwender- und Anwendungsperspektiven zu beleuchten. Ein Domainwissen ist hier unumgänglich. Der dazu notwendige Austausch zwischen dem, der die Prüfung aufsetzt, und dem fachlich Verantwortlichen ist zwingend.

Datenqualität unterstützende Neuerungen in der Oracle-Datenbank 23ai

A — Boolean Data Type

Als erstes sei hier der neue Boolean Datentyp zu nennen. Dieser ermöglicht die attributbezogene Qualität des Inhaltes einer Spalte. Dadurch wird die Accuracy gesichert, da es keine extra Programmierung mehr benötigt, um die Inhalte zu prüfen.

Als gültige Werte für den Datentyp können 0/1, Y/N, T/F, On/Off verwendet werden. Dies bietet eine breite Verwendbarkeit und Spielraum für die Nutzung. Die Abfragen können auf True oder Non True prüfen und True ist der Default, wie in *Listing 1* zu sehen ist.

B — Domains

Als zweites wird die Verwendung von Domains ermöglicht. Dies ermöglicht es, einen SPOD (Single Point of Definition) zu erreichen. Gerade in Umgebungen, in denen Daten aus verschiedenen Applikationen zusammenfließen oder unterschiedliche Entwickler/-Gruppen beschäftigt sind, sind unterschiedliche Definitionen von Attributen eine wichtige Quelle für Datenqualitätsverluste. Mit den Domains können Definitionen und Constraints an einer Stelle zentral definiert und dann datenweit verwendet werden. Beispielhaft sei hier die E-Mail dargestellt *(siehe Listing 2)*.

C — Annotations

Wir können uns Annotations als eine Erweiterung von Datenbankkommentaren vorstellen. Im vorangegangenen Beispiel wurden diese schon verwendet. Mit Kom-

```
CREATE TABLE kunden(
             kunden_id number,
             aktiv boolean);
INSERT INTO kunden values(1,true);
-- Finde die aktiven Kunden
SELECT kunden_id FROM kunden WHERE aktiv;
```

Listing 1: Definition und Verwendung des neuen Boolean Datentyps

```
create domain email_dom as varchar2(100)
constraint email_chk check
(regexp_like (email_dom, '^(\S+)\@(\S+)\.(\S+)$'))
display lower(email_dom)
order   lower(email_dom)
annotations (Description 'Domain for Emails');

create table t1 (
  id     number,
  email  domain email_dom
);

-- fehlerfreier Insert
insert into t1 values (1, 'Banana@fruit.com');
insert into t1 values (2, 'apple@fruit.com');
-- check constraint beim Insert verletzt
insert into t1 values (3, 'banana');
select domain_display(email) from t1 order by email;
```

Listing 2: Definition und Verwendung von einer Domain

```
CREATE MATERIALIZED VIEW MView1
  ANNOTATIONS (Title 'Tab1 MV1', ADD Snapshot)
  AS SELECT * FROM Table1;
```

Listing 3: Definition und Verwendung von Annotations

```
create table t1 (
  id           number,
  description  varchar2(15) default on null for insert and update
'kein Wert vorhanden'
);
insert into t1 (id, description) values (1, null);
insert into t1 (id) values (2);
-- überprüfen der Inserts
select * from t1;
```

Listing 4: DEFAULT ON NULL FOR INSERT AND UPDATE

mentaren konnten wir Objekten wie Tabellen und Spalten Freitext hinzufügen und so deren Zweck und Verwendung beschreiben. Annotationen gehen noch einen Schritt weiter und ermöglichen es uns, den meisten Datenbankobjekten Name-Wert-Paare zuzuordnen, die zu ihrer Beschreibung oder Klassifizierung verwendet werden können. Die Namen und Werte sind Freitext, sodass wir alles auswählen können, was für uns von Bedeutung ist. Damit wird auch mit Annotations der SPOD unterstützt, der schon als Unterstützung zur Erreichung von Datenqualität beschrieben wurde. Annotations können nicht nur für Tabellen, sondern auch für Views, Materialized Views, Indexe und, wie unter B gesehen, für Domains verwendet werden *(siehe Listing 3)*.

D — DEFAULT ON NULL FOR INSERT ONLY

Nicht vorhandene Werte stellen immer ein Datenqualitätsproblem da. Es kann nicht festgestellt werden, ob ein Wert vergessen, fehlerhaft oder aus sonstigen Gründen nicht vorhanden ist. Daher sollte immer auf NULL-Werte verzichtet werden. Schon mit der Version 12c der Datenbank wurde der DEFAULT ON NULL eingeführt, der einen grundsätzlichen Wert für NULL festlegte.

Nun kann in der 23ai-Version der Datenbank das Verhalten gezielter gesteuert werden und der DEFAULT ON NULL FOR INSERT ONLY angegeben werden, um spätere Operationen auf der Spalte gezielt steuern zu können. Diese Flexibilität erweitert sich auch auf DEFAULT ON NULL FOR INSERT AND UPDATE. Damit kann eine gezielte Verwendung gesteuert und so die Qualität der Daten weiter gesteigert werden, da NULL-Werte gezielt vermieden werden *(siehe Listing 4)*.

E — SQL- und PL/SQL-Vereinfachungen

Vereinfachungen im Bereich der Programmierung und des Codings haben zwar nur indirekt einen Einfluss auf die Datenqualität, aber spielen dennoch in der Sicherstellung von Datenqualität eine wichtige Rolle. Hier seien einige Beispiele genannt, aber nicht weiter ausgeführt.

Die Erweiterung der CASE-Verzweigungen in PL/SQL vereinfacht die Überprüfung von Werten enorm. Dies gilt auch für die umfangreiche Anpassung der Schleifenbildung mit IF THEN ELSE. Die Angabe eines Spalten-Alias für die Gruppierung macht auch so manchen SQL-Code lesbarer und damit fehlerfreier und sichert Qualität. Auch das Update über einen Join oder der Value Constructor vereinfacht die Handhabung von Tabellen und SQL und bietet eine gute Möglichkeit, in Zukunft einfacheren Code zu schreiben, der die Datenqualität sicher stellen soll oder aber auch bestehenden Code anzupassen *(siehe Listing 5)*.

Zusammenfassung

Datenqualität ist und bleibt eine Aufgabe, die nur durch Fleiß und Schweiß zu erreichen ist. UND sie löst sich nicht von allein. Datenqualität ist eine stetige Aufgabe und bedarf kontinuierlicher Anpassung der Prozesse. Technik kann nur ein Hilfsmittel sein, um sie zu erreichen oder nahe heranzukommen. Denn die beste Technik ist kein Garant für hohe Datenqualität, auch wenn einige Aspekte dadurch optimal unterstützt werden. Die Datenbank kann mit ihren Funktionen eine unterstützende Aufgabe wahrnehmen. Es lohnt sich also dranzubleiben und die neuen Möglichkeiten zu eruieren und wahrzunehmen. Viele weitere Aspekte liegen aber außerhalb der Verantwortung der Datenbank.

```
-- Table-Value-Constructor
WITH virtual_tab (id, letter, description) AS
     (VALUES
          (1,'a','Text a'),
          (2,'b','Text b'),
          (3,'c','Text c'),
          (4,'d','Text d')
     )
SELECT * FROM virtual_tab;

-- Group By mit Alias
select count(ename),
       dname || loc as department
from dept, emp
group by department

-- IF Bedingungen mit einigen Erweiterungen
BEGIN
    FOR i number(2,1) IN 0.5 .. 5.5 BY 0.5 LOOP
        DBMS_OUTPUT.PUT_LINE(i);
    END LOOP;
END;
```

Listing 5: Verschiedene Beispiele für SQL und PL/SQL

Quellen

[1] Wang, R.Y., Strong, D.M.: Beyond accuracy: What data quality means to data consumers. J. Manage. Inf. Syst. 12(4), 5-34 (1996)

Über den Autor

Detlef Egbert Schröder ist seit knapp 30 Jahren bei Oracle Deutschland beschäftigt und setzt sich zurzeit mit Datawarehouse-Systemen und Architekturen, Analyse von Daten sowie den KI-Funktionalitäten der Datenbank auseinander. Das BWL-Studium mit Schwerpunkt Wirtschaftsinformatik absolvierte er in Osnabrück. Mit Kollegen hat er eine Vielzahl von Workshops und Seminaren rund um das Thema DWH und Machine Learning konzipiert und durchgeführt.

Detlef E. Schröder
detlef.e.schroeder@oracle.com

KI-Features in der Praxis – ein OCI Document Understanding Deep Dive

Fabian Neureiter, Hyand

In diesem Artikel wird der OCI Document Understanding KI-Service anhand eines konkreten Use Cases vorgestellt. Ein Exkurs zur Funktionsweise des Dienstes im Hintergrund sowie eine Beschreibung, wie APEX-Anwendungen mit diesem Dienst angereichert werden können, sind ebenfalls enthalten. Abschließend wird die Eignung des Services als Lösungsansatz für den Use Case bewertet und auf (noch) existierende Limitierungen eingegangen.

Schon seit längerer Zeit ist die Nutzung KI-gestützter Dienste nicht mehr ausschließlich speziell dafür ausgebildeten Berufsgruppen wie Data Scientists vorbehalten. Viele Hersteller stellen mittlerweile solche Dienste für die breite (Entwickler-) Öffentlichkeit zur Verfügung, insbesondere in Kombination mit der jeweiligen eigenen Cloud. Dabei handelt es sich bei den Anbietern nicht mehr nur um die bekannten „Big Player" wie Amazon, Microsoft und Google, sondern auch neuere, weniger bekannte Unternehmen wie UiPath bieten konkurrenzfähige Produkte an.

Es verwundert daher nicht, dass Oracle im Rahmen der Oracle Cloud ähnliche Dienste im Angebot hat. Einer dieser KI-Services, der Document-Understanding-Service, wird im Folgenden vorgestellt.

Motivation für die Auseinandersetzung mit KI-Diensten

Wenn man zur gegenwärtigen Zeit, dem Ende des Jahres 2024, eine Sache sicher feststellen kann, dann ist es diese: das Thema KI ist in aller Munde.

Ist man in der IT tätig, hat man die Sätze „Wir müssen jetzt auch mal was mit KI machen", oder auch, „Wie können wir KI in unser Produkt integrieren?", vielleicht schon einmal zu oft gehört. Doch auch in der breiten Öffentlichkeit ist das Thema omnipräsent und verspricht vom baldigen Untergang der Menschheit bis hin zur utopischsten Zukunft alles, was das Herz begehrt.

Geht es Ihnen wie dem Autor, so macht Sie ein solcher „Hype" vielleicht zunächst einmal skeptisch. Wenn etwas das Potential hat, alles Mögliche zu sein, fällt es schwer, einen konkreten Sinn in der Nutzung einer solchen Sache zu sehen.

Was fast immer hilft, um sich Klarheit zu verschaffen, ist einfach damit herumzuspielen.

Als Einstiegspunkt bot sich der Github-Copilot an, ein von Visual Studio Code aus nutzbarer Dienst zum (unter anderem) intelligenten Code-Vervollständigen und zum automatischen Code-Reviewen. Kaum eingebunden, ergänzte Copilot, nach manueller Bestätigung, sämtliche Logger-Aufrufe für die Parameter einer Prozedur mit sehr vielen Eingangsparametern *(siehe Abbildung 1)*.

Was sonst einige Zeit in Anspruch nimmt und eine mühselige Aufgabe ist, war in Sekunden erledigt. Daraus ergab sich die Erkenntnis, was KI-Dienste, nach Auffassung des Autors, in erster Linie sind (oder sein sollten):

Werkzeuge, die helfen, das Verrichten mühseliger Arbeiten zu vereinfachen und/oder zu beschleunigen oder diese ganz abzunehmen.

Eine mühselige und wenig geliebte Aufgabe in vielen Unternehmen ist das Erfassen eingehender Rechnungen durch das Backoffice. Dabei soll der Inhalt verschiedenster Rechnungsarten erfasst und in ein beliebiges System übertragen werden. Mühselig ist diese Aufgabe, weil sie ein geringes Maß an Denkarbeit, jedoch ein gewisses Maß an Konzentration erfordert und einer nicht unerheblichen Bearbeitungszeit bedarf.

Kann ein KI-Dienst hier Unterstützung bieten? Dies soll im Folgenden anhand des Oracle OCI Document Understanding Services untersucht werden.

Eigenschaften des Document Understanding Services

Das OCI Document Understanding „ist ein KI-Service, mit dem Entwickler Text, Tabellen und andere Schlüsseldaten über APIs und Befehlszeilentools aus Dokumentendateien extrahieren können." *[1]*.

Mit „Text" sind hier die in einem Dokument enthaltenen Worte, aber auch ganze Zeilen gemeint.

„Tabellen" meint klassische Tabellenstrukturen und tabellenähnliche Strukturen wie die Auflistung gekaufter Artikel, deren Preis, Steuer, Gesamtpreis und vieles mehr.

„Schlüsseldaten" sind bestimmte Kategorien von Informationen, welche versucht werden, in dem Dokument zu finden. Für Rechnungen wären diese Schlüssel zum Beispiel „Rechnungsempfänger", „Gesamtsumme" oder „Rechnungsdatum". Das Ergebnis der Auslesung von Schlüsseldaten sähe dann wie *Abbildung 2* aus.

Es wird außerdem die Klassifizierung nach Dokumentenarten unterstützt.

Document Understanding verarbeitet jedoch nicht beliebige Arten von Dokumenten, sondern konzentriert sich auf:

- Invoices (Rechnungen)
- Receipts (Quittungen)
- Resumes (Lebensläufe)
- Tax Forms (Steuerbescheide)
- Drivers Licences (Führerscheine)
- Passports (Pässe)
- Bank Statements (Kontoauszüge)
- Payslips (Gehaltsabrechnungen)

Exkurs: Wie arbeitet der Document Understanding Service im Hintergrund?

Bevor tiefer in die Arbeit mit dem Document Understanding-Dienst eingestiegen wird, bietet es sich an, die Frage zu stellen: Wie genau arbeitet eine solche Technologie eigentlich?

Als Beispiel soll hier die Funktion der Dokumentenklassifikation herhalten. Einen Einstieg ermöglicht zunächst die eigene Fähigkeit zur Unterscheidung von Dokumenten. Woher weiß man, welche Art Dokument man vor sich hat?

Um das Dokument in *Abbildung 3* als Rechnung zu erkennen, hat man als Mensch verschiedene Ansatzpunkte. Schaut man sich nur den Text und insbesondere die verwendeten Worte an, („Invoice", „order", „Total" und andere) sprechen diese für einen Gesamtzusammenhang/Kontext, der dem einer Rechnung entspricht *(siehe Abbildung 4)*.

Blickt man rein auf die Struktur, so ähnelt diese zunächst einem Brief. Sie weist allerdings in der Mitte des Dokuments eine tabellarische Struktur auf, was charakteristisch für eine Rechnung ist *(siehe Abbildung 5)*.

Berücksichtigt man nun den Kontext des erkannten Texts und die Dokumentstruktur zusammen, kann sich ein menschlicher Betrachter sehr sicher sein, dass es sich bei dem dargestellten Dokument um eine Rechnung handelt.

Ähnlich funktioniert die Dokumentenklassifizierungsfunktion des Document Understanding Service. Verschiedene Modelle arbeiten zusammen, um ein möglichst gutes Ergebnis zu erzielen *[2]*:

- Bildbasierte Modelle (Image-based Models), die anhand der äußeren Erscheinung klassifizieren, hier EfficientNet.
- Auf Texterkennung spezialisierte Modelle, die Text, egal ob handschriftlich

lct_p0029_api.pkb

```
logger.append_param(l_params, 'pi_step_id', pi_step_id);
logger.append_param(l_params, 'pi_step_case_id', pi_step_case_id);
logger.append_param(l_params, 'pi_step_sttp_id', pi_step_sttp_id);
logger.append_param(l_params, 'pi_step_name', pi_step_name);
logger.append_param(l_params, 'pi_step_execution_sequence', pi_step_execution_sequence);
logger.append_param(l_params, 'pi_checksum', pi_checksum);
logger.append_param(l_params, 'pi_app_id', pi_app_id);
logger.append_param(l_params, 'pi_page_id', pi_page_id);
logger.append_param(l_params, 'pi_element', pi_element);
logger.append_param(l_params, 'pi_value', pi_value);
logger.append_param(l_params, 'pi_context', pi_context);
logger.append_param(l_params, 'pi_page_is_modal', pi_page_is_modal);
logger.append_param(l_params, 'pi_selector_type', pi_selector_type);
logger.append_param(l_params, 'pi_custom_selector', pi_custom_selector);
logger.append_param(l_params, 'pi_custom_selector_text', pi_custom_selector_text);
logger.append_param(l_params, 'pi_step_timeout', pi_step_timeout);
logger.append_param(l_params, 'pi_force_interaction', pi_force_interaction);
logger.append_param(l_params, 'pi_selectOption_option', pi_selectOption_option);
logger.append_param(l_params, 'pi_append_session_id', pi_append_session_id);
logger.append_param(l_params, 'pi_path', pi_path);
logger.append_param(l_params, 'pi_nav_referer', pi_nav_referer);
logger.append_param(l_params, 'pi_nav_wait_until', pi_nav_wait_until);
logger.append_param(l_params, 'pi_click_btn_type', pi_click_btn_type);
logger.append_param(l_params, 'pi_click_click_count', pi_click_click_count);
logger.append_param(l_params, 'pi_delay', pi_delay);
logger.append_param(l_params, 'pi_trial', pi_trial);
logger.append_param(l_params, 'pi_presskey_modifiers', pi_presskey_modifiers);
logger.append_param(l_params, 'pi_presskey_key', pi_presskey_key);
logger.append_param(l_params, 'pi_click_wait_for_navigation', pi_click_wait_for_navigation);
logger.append_param(l_params, 'pi_selectOption_select_by', pi_selectOption_select_by);
logger.append_param(l_params, 'pi_waitMilliseconds_time', pi_waitMilliseconds_time);
logger.append_param(l_params, 'pi_user', pi_user);
logger.append_param(l_params, 'pi_remove_target', pi_remove_target);
logger.append_param(l_params, 'pi_region', pi_region);
logger.append_param(l_params, 'pi_column_id', pi_column_id);
logger.log('START', l_scope, null, l_params);
```

Abbildung 1: Mit GithubCopilot erstellte Liste von logger-Aufrufen für Prozedur-Parameter (Quelle: Fabian Neureiter)

oder gedruckt, und in unterschiedlichen Winkeln dargestellt, erkennen (OCR = Optical Character Recognition, optische Zeichenerkennung).
- Textbasierte Modelle (Text-based Models), welche auf das Erkennen von Gesamtzusammenhängen in Texten spezialisiert sind, hier BERT.

Kosten

Um Document Understanding verwenden zu können, entstehen nicht unmittelbar Kosten, da für ein initiales Ausprobieren das frei verfügbare Kontingent an Transaktionen wohl selten überschritten werden dürfte. Dennoch ist ein Upgrade zu einer „Paid Instance" erforderlich, inklusive des Hinterlegens einer Kreditkarte.

Die Kosten richten sich (Stand 14.12.2024) nach der Nutzung des Dienstes („Pay-as-you-go") und werden auf Transaktionsbasis abgerechnet. Eine Transaktion ist „als die Anzahl von Operationen pro Seite definiert, die als Eingabe für den Service (API-Aufruf) bereitgestellt wird" definiert *[4]*. Abgerechnet wird

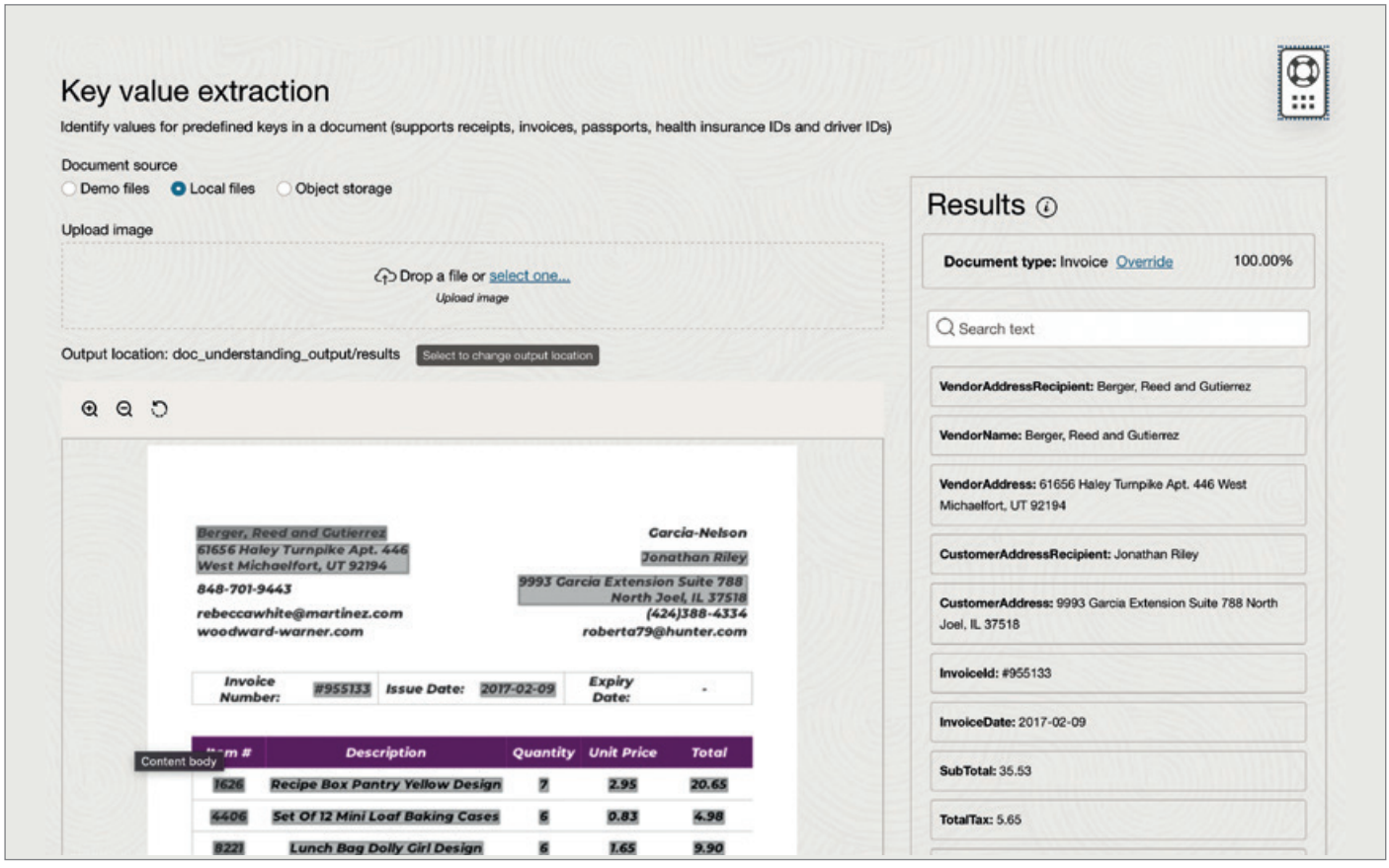


Abbildung 2: Beispiel Key-Value-Extraktion in der OCI Console (Quelle: Fabian Neureiter)

```
allow group <group_in_tenancy> to manage ai-service-document-family in tenancy
```

Listing 1: Policy für die Nutzung der Document-Understanding-API

```
allow group <group_in_tenancy> to manage object-family in tenancy
```

Listing 2: Policy für den Zugriff auf den Object Storage

```
oci ai-document analyze-document-result analyze-document [OPTIONS]
```

Listing 3: Verwendung des CLI-Tools

```
{
    "compartmentId": "#COMPARTMENT_ID#",
    "document": {
        "source": "INLINE",
        "data": "#FILE_DATA#"
    },
    "documentType": "#DOCUMENT_TYPE#",
    "features": #AI_FEATURES!RAW#
}
```

Listing 4: Beispiel eines Request Body Templates

GASQ Service GmbH • Rothenburger Str. 11 • 90443 Nuremberg

Mr. Fabian Neureiter
Wilhelm-Stumpf-Straße 84
44789 Bochum

GASQ Service GmbH
Rothenburger Str. 11
90443 Nuremberg
Germany
Tel. +49 911 990078 0
Fax +49 911 990078 99
[redacted]
info@gasq.org

Nuremberg, 21.06.2022

Invoice No. 67508 **Reference:** [redacted]

Dear Mr. Fabian Neureiter

We are invoicing you according to your order as follows:

Pos	Description	Amount	Unit Price	Net Price	VAT (19%)	Gross Price
1	ISTQB CTFL E-Exam Exam Date: 07/25/2022 Exam Place: Düsseldorf Participant: Fabian Neureiter	1	215,00 €	215,00 €	40,85 €	255,85 €
			Total	215,00 €	40,85 €	255,85 €

The exam fee has been successfully charged to your credit card.

If you should have any further questions, please do not hesitate to contact us.

Sincerely,
Oliver Braun

Abbildung 3: Eine Beispielrechnung (Quelle: Fabian Neureiter)

pro 1000 Transaktionen, wobei monatlich 5000 Transaktionen frei sind *[3]*.

Oracle Cloud Set-Up

Neben der Grundvoraussetzung, einer Oracle-Cloud-Instanz, empfiehlt sich zunächst das Erstellen einer Gruppe, die User beinhaltet, welche den Document Understanding Service nutzen sollen (Bild Create Group). Dieser Gruppe muss im Anschluss aus dem „Root Compartment" per „Policy" Zugriff auf die Document- Understanding-API gewährt werden. „Compartments" sind logische Klammern um eine Menge von Ressourcen innerhalb einer OCI-Instanz. Das „Root Compartment" ist die höchste Ebene innerhalb einer Instanz und beinhaltet alle Ressourcen darin. „Policies" sind Regeln, welche den Zugriff auf bestimmte Ressourcen festlegen. Die benötigte „Policy" sähe hier wie in *Listing 1* aus.

Soll eine größere Menge von Rechnungen zur späteren Analyse bereitgehalten werden, empfiehlt sich deren Speicherung in „Buckets", Objekt-Container innerhalb des Object Storage Services der OCI-Instanz. Der bereits erstellten Gruppe muss dann, erneut über eine Policy, Zugriff auf den Object Storage gegeben werden *(siehe Listing 2)*.

Ist das Set-Up zufriedenstellend beendet, kann mit der tatsächlichen Nutzung des Dienstes begonnen werden.

Arten der Nutzung des Document Understanding Services

Die direkteste Form der Nutzung ist unmittelbar in der OCI-Instanz zu finden, erreichbar über den Menüpunkt „Analytics & AI", dann im Bereich „AI Services" und schließlich im Unterbereich „Document Understanding" *(siehe Abbildung 6)*.

Das dort enthaltene Interface enthält neben der Möglichkeit des Hochladens des zu analysieren Dokuments eine auf-

Abbildung 4: Im Text der Beispielrechnung enthaltene Worte (Quelle: Fabian Neureiter)

Abbildung 5: Die Struktur der Beispielrechnung mit maskiertem Text (Quelle: Fabian Neureiter)

bereitete Darstellung des Analyseergebnisses *(siehe Abbildung 2)*.

Für einen ersten Eindruck eignet sich dieser Bereich optimal. Möchte man in OCI bleiben, legt aber weniger Wert auf ein grafisches Interface, gibt es ebenfalls ein Command-Line-Werkzeug für die Interaktion mit dem Document-Understanding-Dienst *[4]*. Dieses Tool kann zum Beispiel folgendermaßen verwendet werden *(siehe Listing 3)*:

Außerhalb der OCI-Instanz ist die Nutzung von Document Understanding ebenfalls möglich.

Zum einen werden SDKs (Software Development Kits) in verschiedenen Sprachen angeboten. Es handelt sich dabei um eine Menge von plattformspezifischen Tools, welche die Interaktion mit OCI und dem Document-Understanding-Dienst vereinfachen sollen. Diese SDKs sind für die folgenden Sprachen verfügbar *[4]*:

- Java
- Python
- JavaScript and Typescript
- NET
- Go
- Ruby
- PL/SQL

Zum anderen ist eine REST-API nutzbar *[5]*, welche die Kommunikation mit allen möglichen REST-fähigen Systemen ermöglicht. Das macht eine Integration des Oracle Document Understanding besonders einfach, insbesondere im Hinblick auf APEX-Anwendungen, weswegen auf die REST-API genauer eingegangen werden soll.

Die Document Understanding REST-API

Oracle stellt zwei Arten der Kommunikation mit der API bereit, asynchron und synchron. Für asynchrone Kommunikation *[5]* existiert der Endpunkt „createProcessorJob“, welcher auf POST-Requests hört und ein Dokument oder den Speicherort eines Dokuments innerhalb des OCI Object Storage sowie einen Ablageort im Objekt-Storage für die Analyseergebnisse entgegennimmt. Als Antwort erhält man, neben anderen Informationen, den Status des „Processor Jobs“ und eine eindeu-

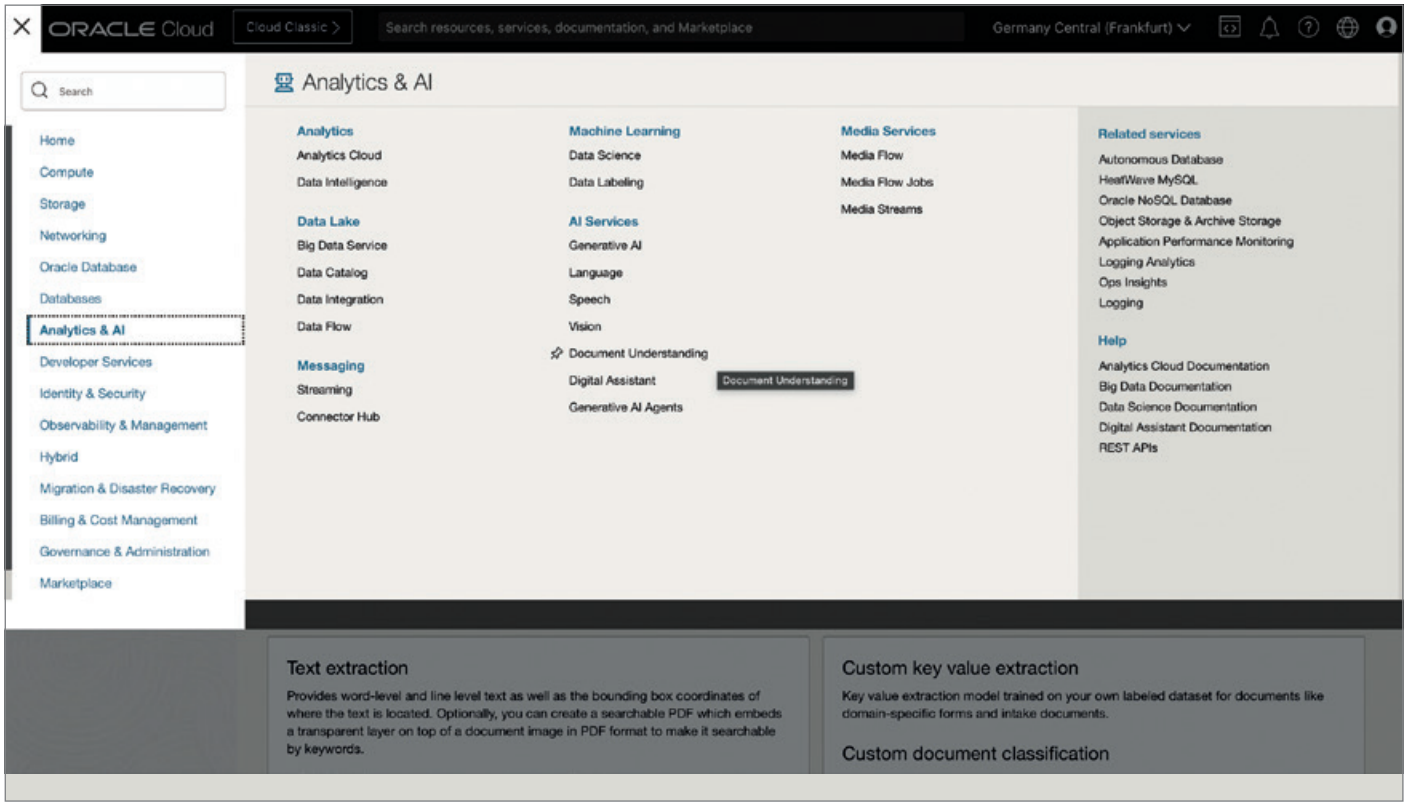


Abbildung 6: Wo der Document Understanding Service in der Cloud Console zu finden ist. (Quelle: Fabian Neureiter)

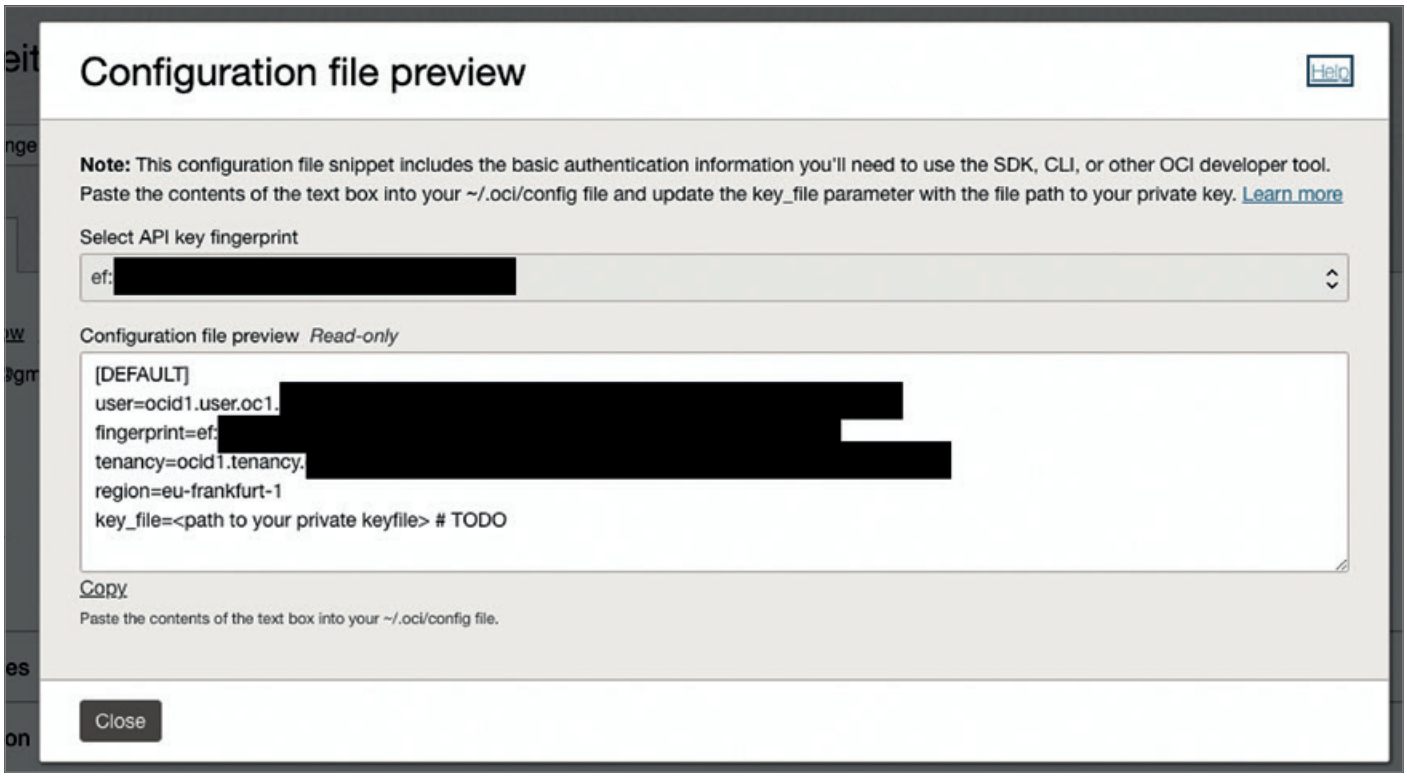


Abbildung 7: Die Vorschau der Konfigurationsdatei (Quelle: Fabian Neureiter)

tige, diesen „Processor Job" identifizierende, ID. Der Status kann dann periodisch über einen GET-Request zu „getProcessorJob" angefragt werden. Ist der Status „SUCCEEDED" erreicht, kann das Ergebnis im zuvor angegebenen Ort im Object Storage abgefragt werden.

Die synchrone Ansprache folgt dem Muster „Dokument (oder Ablageort) hin, Analyseergebnis direkt zurück". Lediglich ein POST-Request zum „analyzeDocument"-Endpunkt ist dafür nötig.

Um mit den Endpunkten interagieren zu können, ist eine Authentifizierung ge-

```
[…]
l_file_content_base64 := apex_web_service.blob2clobbase64(l_file_con-
tent);

apex_exec.add_parameter( l_rest_params
   , 'FILE_DATA'
   , l_file_content_base64 );
apex_exec.add_parameter( l_rest_params
                       , 'COMPARTMENT_ID'
                       , pi_compartment_id );
apex_exec.add_parameter( l_rest_params
                        , 'DOCUMENT_TYPE'
                        , 'INVOICE' );
apex_exec.add_parameter( l_rest_params
                        , 'AI_FEATURES'
                        , build_feature_json_array(
pi_feature_list =>                 pi_document_feature
                          ));

      apex_exec.execute_rest_source(
          p_static_id        => 'oci_document_understanding',
          p_operation        => 'POST',
          p_parameters       => l_rest_params
      );

    l_status_code := apex_web_service.g_status_code;

    if l_status_code = 200 then

l_response := apex_exec.get_parameter_clob(l_rest_params
,'RESPONSE');

      handle_analysis_results_api.insert_results(
          pi_doc_id             => pi_doc_id
        , pi_invoked_ai_feature => pi_document_feature
        , pi_response_json      => l_response
        , pi_service_provider   => pi_service_provider
        , pi_async_job_id       => null
        , pi_status             => 'Completed'
        , po_anre_id            => po_anre_id
      );

    else
      raise_application_error( -20001
, 'Error invoking OCI AI API. Status code: ' || l_status_code );
    end if;
[…]
```

Listing 5: Ausschnitt einer Prozedur, welche per APEX_WEB_SERVICE die REST Data Source aufruft.

genüber der REST-API nötig. Dazu finden OCI API-Keys Verwendung, erstellbar im „My Profile"-Bereich der OCI-Instanz. Nach einer Navigation zu „API Keys" ist es möglich, Schlüsselpaare zu generieren

Wichtig: Speichern Sie sich die Daten der im Anschluss an die Schlüsselpaargenerierung dargestellten Konfigurationsdatei irgendwo zwischen, denn diese werden ebenfalls zu Authentifizierung benötigt *(siehe Abbildung 7)*.

Die Verwendung der API von einer APEX-Anwendung aus erscheint auf den ersten Blick simpel und ist es auch. Als Beispiel soll die Verwendung der synchronen API dienen.

Einbindung der API in eine APEX-Anwendung

Innerhalb eines APEX-Workspaces ist der erste Schritt die Erstellung eines „Web Credentials" unter „Workspace Utilities" *(siehe Abbildung 8)*. Dazu kommt das bei der API-Key-Erstellung dargestellte Konfigurations-File zum Tragen. Benötigt werden:

- OCI User ID
- OCI Private Key
- OCI Tenancy ID
- OCI Public Key Fingerprint

Sind die benötigten Informationen eingegeben und ein aussagekräftiger Name für die spätere Referenzierung vergeben, ist der erste Schritt abgeschlossen.

Nun folgt die Erstellung einer „REST Data Source" in den „Shared Components" der App, welche den Document Understanding Service nutzen soll. Die Erstellung richtet sich dabei nach den Daten und Operationen, welche der Endpunkt erwartet. Daher ist es hilfreich, die API-Dokumentation *[5]* parallel zu öffnen.

Wichtig ist dabei zunächst die Setzung des korrekten „REST Data Source Type" auf „Oracle Cloud Infrastructure" sowie die Referenz auf das im ersten Schritt erstellte Credential. Die „Base URL" setzt sich aus dem in der Dokumentation gelisteten passenden Endpoint, hier *„https://document.aiservice.eu-frankfurt-1.oci.oraclecloud.com"*, der API-Version, hier „20221109", und der Pfadkomponente „actions" zusammen. Als „URL Path Prefix", einem Service-spezifischem URL-Pfadbestandteil, wird der Pfad der synchronen API, „analyzeDocument", gesetzt *(siehe Abbildung 9)*.

Da „analyzeDocument" einen POST-Request erwartet, muss eine entsprechende „Operation" im Rahmen dieser REST Data Source erstellt werden. Der wichtigste Teil ist hierbei die HTTP-Methode, hier „POST", und das „Request Body Template" im JSON-Format. Dabei handelt es sich um die Struktur und die Daten, welche der REST-Service im Body des Requests erwartet *(siehe Listing 4)*. Auf jeden Fall benötigt werden bei „analyzeDocument" das Dokument selbst, entweder Base-64-encodiert und direkt übergeben oder als Ort im Object Storage, und die zu verwendenden Analyse-Features, wie Key-Value-Extraction, Klassifizierung und andere.

Die mit dem #-Symbol gekennzeichneten Strings sind Parameter, die vor dem Aufruf der REST Data Source mit Werten gefüllt werden, also die In-Parameter. Durch einen Click auf „Synchronize with Body" werden diese erkannt und einzeln

```
[…]
, {
      "fieldType" : "KEY_VALUE",
      "fieldLabel" : {
        "name" : "CustomerAddressRecipient",
        "confidence" : 0.9981994
      },
      "fieldName" : null,
      "fieldValue" : {
        "valueType" : "STRING",
        "text" : "Jonathan Riley",
        "confidence" : null,
        "boundingPolygon" : {
          "normalizedVertices" : [ {
            "x" : 0.7613788200827206,
            "y" : 0.12706859241832386
          }, {
            "x" : 0.9228336109834558,
            "y" : 0.12706859241832386
          }, {
            "x" : 0.9228336109834558,
            "y" : 0.14260629827325993
          }, {
            "x" : 0.7613788200827206,
            "y" : 0.14260629827325993
          } ]
        },
        "wordIndexes" : [ 10, 11 ],
        "value" : "Jonathan Riley"
      }
    },
[…]
```

Listing 6: Auszug einer Response für Key-Value-Extraction

```
[…]
  "errors" : [ {
    "code" : "FEATURE_NOT_SUPPORTED",
    "message" : "[Page 1] KeyValue feature is not supported on
non-English documents."
  } ],
[…]
```

Listing 7: Fehler bei der Verwendung deutschsprachiger Rechnungen

```
[…]
    "detectedDocumentTypes" : [ {
      "documentType" : "BANK_STATEMENT",
      "confidence" : 0.7152297
    }, {
      "documentType" : "RESUME",
      "confidence" : 0.26819623
    }, {
      "documentType" : "RECEIPT",
      "confidence" : 0.013987196
    }, {
      "documentType" : "OTHERS",
      "confidence" : 0.0016455313
    }, {
      "documentType" : "INVOICE",
      "confidence" : 8.1768626E-4
    } ],
[…]
```

Listing 8: Ergebnisse der Dokumentklassifizierung bei einer deutschen Rechnung

unter „Operation Parameters“ angelegt. Ein weiterer Parameter, „Content Type“ vom Typ „HTTP Header“, muss manuell als In-Parameter angelegt und mit dem Standardwert „application/json“ vorbelegt werden *(siehe Abbildung 10)*.

Das Analyse-Ergebnis, welches „analyzeDocument“ als Response zurückliefert, wird im ebenfalls manuell zu erstellenden Out-Parameter „RESPONSE“ für die Weiterverarbeitung gespeichert.

Damit ist die Rest Data Source vollständig definiert. Sie kann nun zum Beispiel im Rahmen einer PL/SQL-Funktion oder -Prozedur über die APEX_WEB_SERVICE-API aufgerufen werden *(siehe Listing 5)*.

Ein deklarativer Aufruf ist auch unter der Zuhilfenahme des noch verhältnismäßig neuen „Invoke API“-Seitenprozesses möglich.

Funktionslimits des Document Understanding Services

Ruft man nun die neue Rest Data Source mit einer englischsprachigen Rechnung auf, erhält man eine Response-JSON, welches zum Beispiel bei der Key-Value-Extraktion die erkannten Werte zu den entsprechenden Schlüsseln enthält. Ebenso bekommt man einen Wert, wie sicher sich das System ist, den korrekten Wert gefunden zu haben („Confidence“) und die Koordinaten innerhalb des Dokumentes, wo sich der gefundene Wert befindet *(siehe Listing 6)*.

Wiederholt man denselben Vorgang mit einer deutschen Rechnung, hat man allerdings weniger Glück und erhält folgendes Resultat *(siehe Listing 7)*.

Es stellt sich heraus, dass Funktionen wie die Key-Value-Extraction nur für englischsprachige Dokumente zur Verfügung stehen und Funktionen wie die Dokumentklassifizierung für deutsche Rechnungen deutlich unzutreffende Ergebnisse liefern *(siehe Listing 8)*.

Für die Eignung des Document Understanding Services für den Use Case vom Anfang, dem automatischen Erfassen von Rechnungen zur Entlastung von Backoffice-Mitarbeitern, muss man festhalten, dass diese für nicht-englischsprachige Rechnungen nicht gegeben ist.

Das ist besonders verwunderlich, weil Dienste wie Azure Document Intelligence

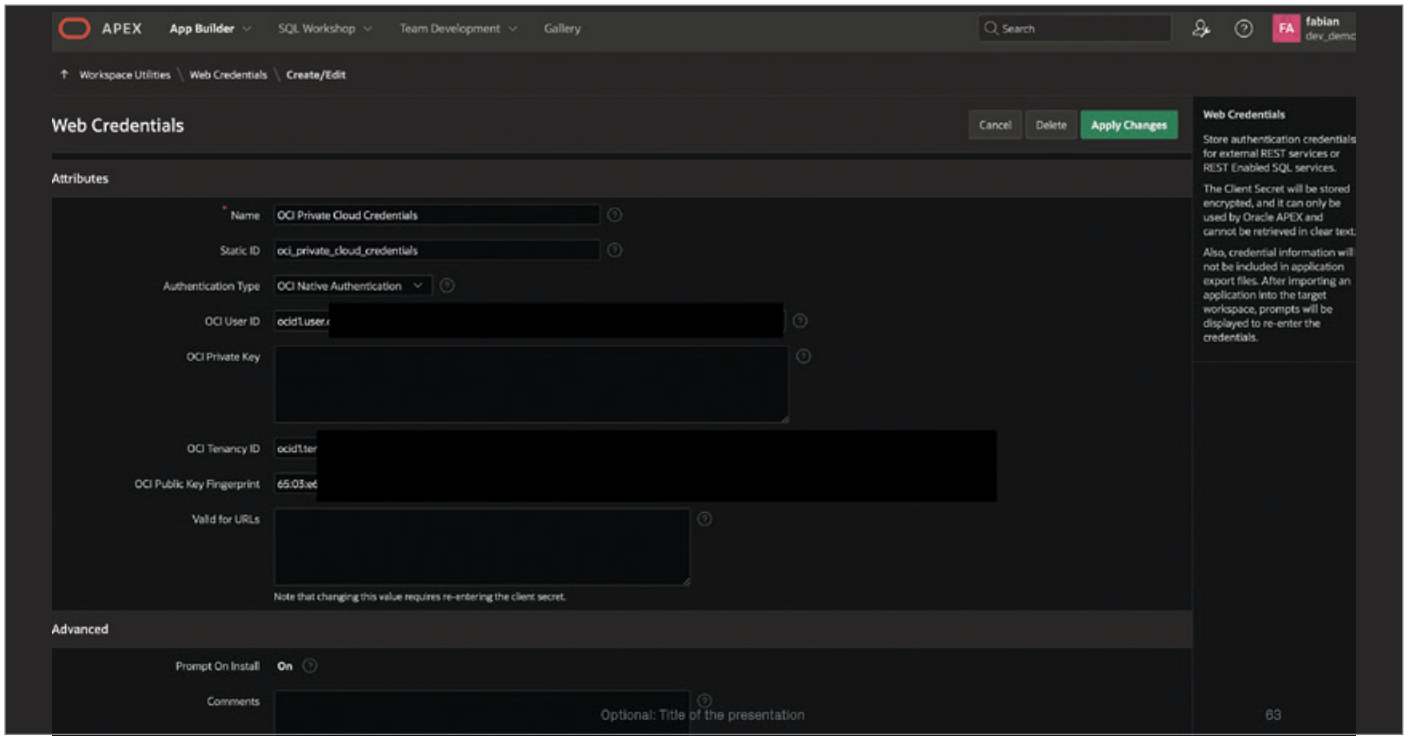


Abbildung 8: Erstellung eines Web-Credentials (Quelle: Fabian Neureiter)

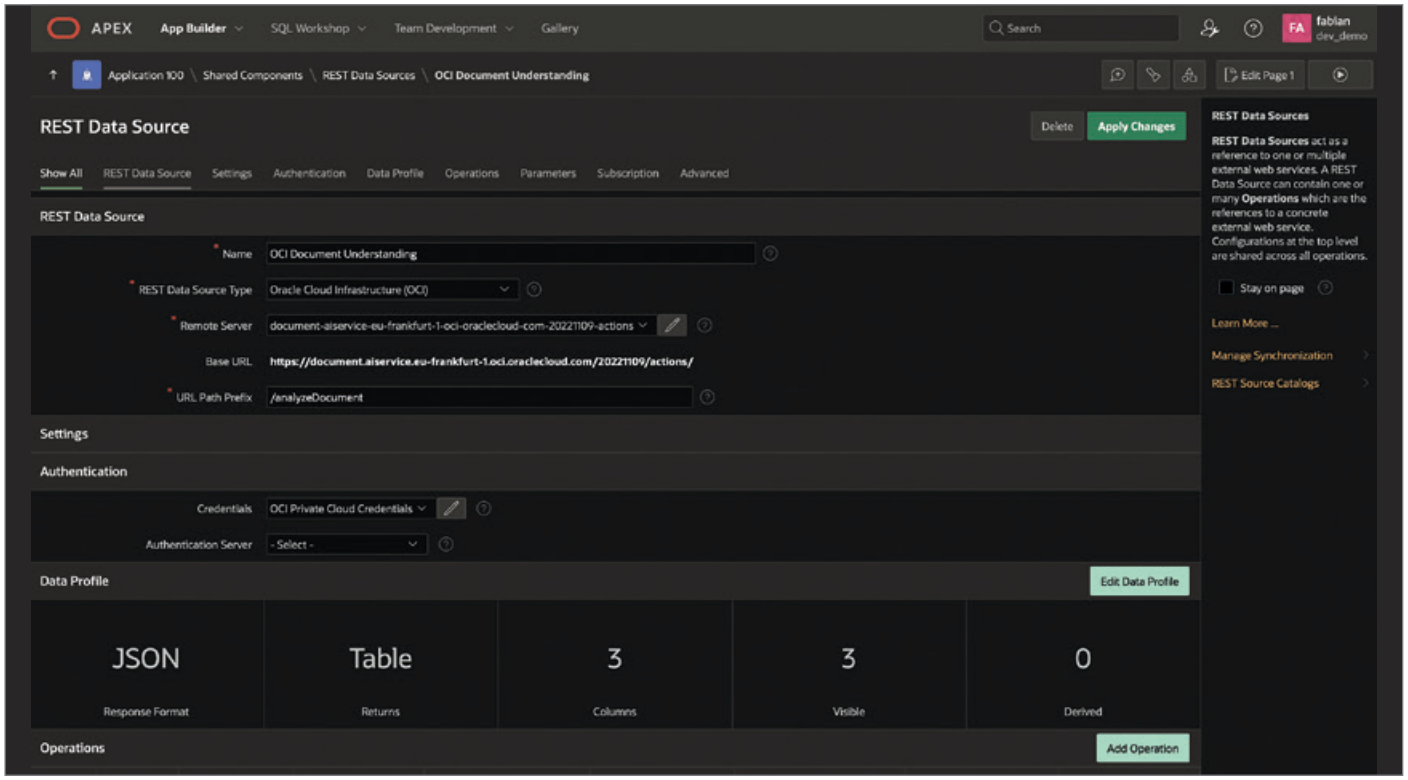


Abbildung 9: Die REST Data Source (Quelle: Fabian Neureiter)

diesbezüglich nicht eingeschränkt sind und ebenso REST APIs bereitstellen, welche vergleichbar leicht in APEX-Anwendungen integrierbar sind.

Es gibt aber gute Neuigkeiten: Der Autor hat bei diversen Oracle-Mitarbeitern auf der DOAG 2024 Konferenz & Ausstellung nachgefragt und besagtes Problem klagend vorgetragen. Daraufhin wurde die Unterstützung der deutschen Sprache für Q1 2025 bestätigt und ist vielleicht schon beim Erscheinen dieses Artikels verfügbar. Vielen Dank noch einmal an die beteiligten Personen.

Fazit

Auch wenn die Unterstützung für das deutschsprachige Publikum vielleicht noch ein wenig auf sich warten lässt, ergibt es Sinn, sich schon jetzt mit dem Document Understanding zu beschäfti-

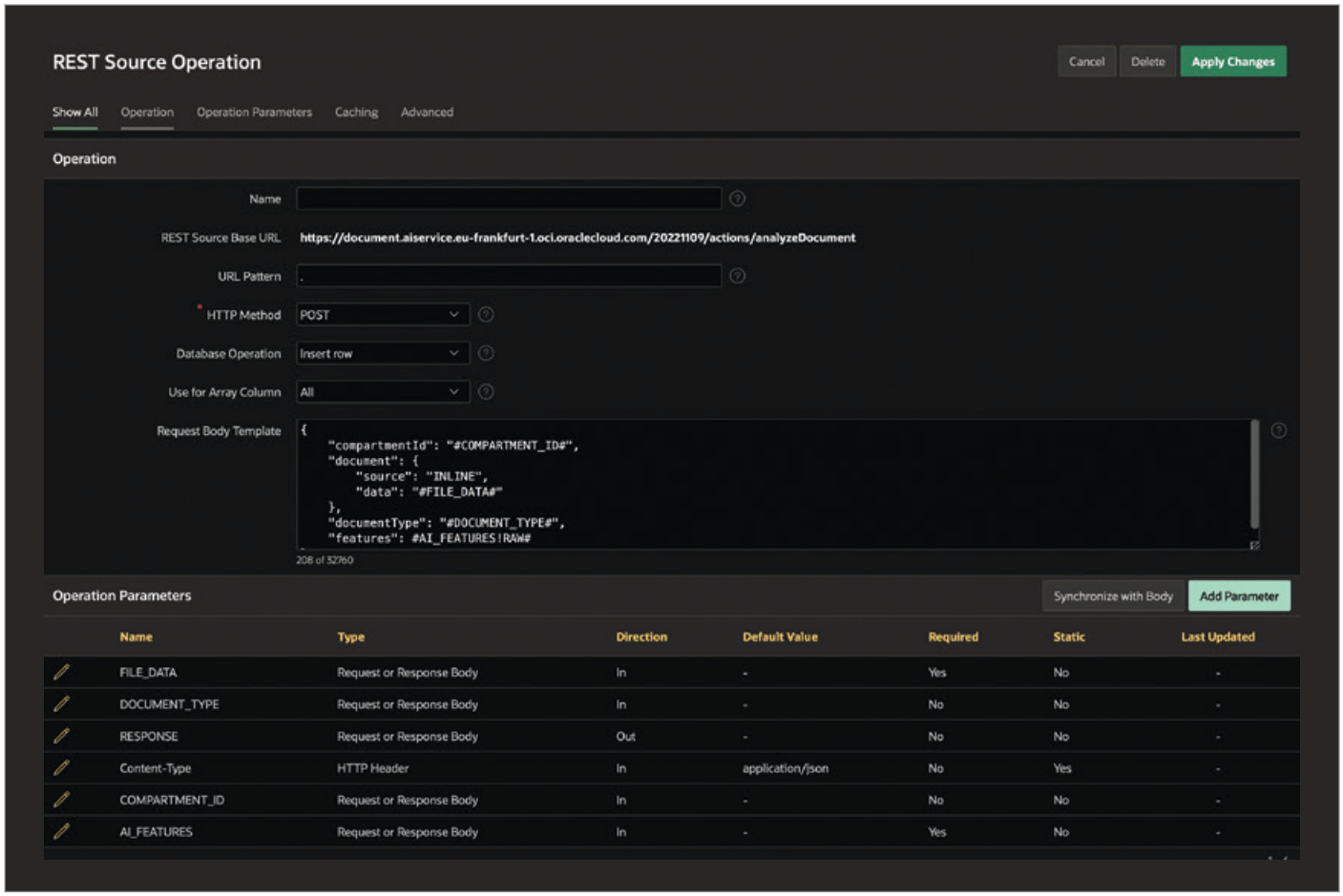


Abbildung 10: Die REST Data Source POST-Operation (Quelle: Fabian Neureiter)

gen. Die vielfältigen Nutzungsmöglichkeiten und der Funktionsumfang sowie die einfache Integration in andere Oracle-Produkte wie APEX lassen Gutes erahnen. Auch die Möglichkeit eigene Modelle, trainiert auf eigenen Dokumenttypen, zu verwenden, ist mächtig und eines eigenen Artikels würdig. Aufgrund des erhöhten Aufwandes war dies jedoch keine Lösung zur Umgehung des Sprachproblems, insbesondere weil diese Lösung im Q1 2025 sehr wahrscheinlich obsolet werden würde.

Es entsteht abschließend der Gesamteindruck eines sich stetig verbessernden, potenten KI-gestützten Dienstes, auch wenn besagte Verbesserung vielleicht etwas langsamer vonstattengeht als bei anderen Mitbewerbern.

Quellen

[1] Oracle, *https://www.oracle.com/de/artificial-intelligence/document-understanding*, zuletzt abgerufen am 15.12.2024.

[2] Oracle, Kanal „Oracle Learning“, Video “Automate & innovate with new OCI Document Understanding”, *https://www.youtube.com/watch?v=BKBlnnf7E-HI&list=PLKCk3OyNwIzt1x62El9gGGeN-aQr0va58c&index=3*, zuletzt abgerufen am 15.12.2024.

[3] Oracle, *https://www.oracle.com/de/artificial-intelligence/document-understanding/pricing/*, zuletzt abgerufen am 15.12.2024.

[4] Oracle, *https://docs.oracle.com/en-us/iaas/Content/API/Concepts/sdks.htm*, zuletzt abgerufen am 15.12.2024.

[5] Oracle, *https://docs.oracle.com/en-us/iaas/api/#/en/document-understanding/20221109/*, zuletzt abgerufen am 15.12.2024.

Über den Autor

Fabian Neureiter ist Berater und Entwickler im Bereich Oracle APEX, mit einem Hintergrund in Medieninformatik und Grafikprogrammierung. Er entwickelt unter anderem das APEX UI-Testing-Tool LCT (Low Code Testing) und ist an JavaScript und Testing im APEX-Kontext sowie KI-Themen im Allgemeinen interessiert. Aktuell beschäftigt er sich mit dem Erstellen und Hosten eines eigenen lokalen KI-Services.

Fabian Neureiter
fabian.neureiter@hyand.com

Machen LLMs Datenmodellierung obsolet?

Tobias Otte, viadee Unternehmensberatung

Die Planung und Implementierung des Datenmodells ist ein wesentlicher Teil von Business-Intelligence-Projekten (BI-Projekten). Neben technischen Belangen, wie der Gewährleistung der Datenintegrität und der Performance von Schreib- und Leseoperationen, ist vor allem das Verständnis eines Modells durch die Anwenderinnen und Anwender ein wichtiges Ziel der Modellierung. Für dispositive Abfragen ist das Sternschema etabliert und hat sich über viele Jahre bewährt. Es wird sowohl von Menschen als auch von BI-Werkzeugen gut verstanden und bietet so die Basis für effiziente Datenabfragen. Ändert sich diese Empfehlung, wenn Menschen zunehmend mit Large Language Models (LLMs) interagieren und diese auf Basis des Modells SQL-Abfragen erzeugen? Wird Datenmodellierung gar obsolet? Oder ist auch für LLMs ein durchdachtes Datenmodell wichtig? Anhand praktischer Beispiele und Erfahrungen werden diese Fragestellungen beleuchtet sowie Empfehlungen für eine effektive Datenmodellierung im Zeitalter von LLMs gegeben.

Die Disziplin der Datenmodellierung wurde schon mehrfach totgesagt. Vor allem in der Anfangszeit der Data Lakes lag der Fokus vieler Unternehmen darauf, möglichst alle Daten zunächst roh zu speichern, ohne dass bereits eine konkrete Verwendung geplant war. Datenmodellierung wurde eher als hinderlich gesehen und es wurde darauf verwiesen, dass die Daten zur Abfragezeit (Schema on Read) in Strukturen gebracht werden können. So wurde der Aufwand der Datenintegration auf später vertagt. Es zeigte sich jedoch immer wieder, dass Modellierung unverzichtbar ist, um Informationen in hoher Qualität bereitzustellen und daraus belastbare Entscheidungen treffen zu können. So lebte die totgesagte Disziplin immer wieder auf. Die Ausgabe 04/2022 vom BI-Spektrum *[1]* beschäftigte sich aus verschiedenen Blickwinkeln mit dem Thema Datenmodellierung und zeigte ebenfalls, dass trotz der rasanten technologischen Entwicklungen in der Cloud die Grundsätze der Modellierung weiterhin gültig sind.

Im Sinne des Titels wäre Datenmodellierung im Zeitalter von Large Language Models (LLMs) dann obsolet, wenn es keinen Unterschied macht, welche – oder ob überhaupt eine – Modellierungstechnik verwendet wird und auch Grundsätze der Modellierung, wie Schlüsselbeziehungen und Benennung, unwichtig geworden sind. Im Folgenden werden verschiedene Modellierungstechniken vorgestellt, die im nächsten Kapitel für das Beispielszenario verwendet werden.

Dritte Normalform

Die dritte Normalform (3NF) ist das typische Modellierungsverfahren für operative Systeme, kann aber auch für dispositive Systeme verwendet werden. Zur Anfangszeit des Data Warehousing war ein unternehmensweites Datenmodell in dritter Normalform das postulierte Ziel *[2]*. In der 3NF werden Daten gemäß der Normalisierungsregeln in separate Tabellen aufgeteilt *(siehe Abbildung 2)*, um Redundanz zu minimieren und die Datenintegrität zu gewährleisten. Das Ergebnis ist ein Modell, das vor allem für Abfragen und Änderungen einzelner Datensätze sehr effizient ist. Nachteile bestehen bezüglich der Verständlichkeit und Performance bei sehr großen Modellen sowie der Resilienz gegenüber Änderungen.

Sternschema

Das Sternschema ist der De-facto-Standard für die Präsentationsschicht von Data-Warehouse-Umgebungen. Nach Kimball ist das Data Warehouse die Summe der Sternschemata, die über konforme Dimensionen miteinander verbunden sind *[3]*. Ein einzelner Stern besteht aus einer zentralen Faktentabelle, die mit mehreren Dimensionstabellen verknüpft ist. Die Faktentabelle enthält die numerischen Messwerte oder Metriken, während die beschreibenden Attribute in den Dimensionstabellen gespeichert werden. Das Sternschema wird von Usern und BI-Werkzeugen in der Regel gut verstanden. Herausforderungen gibt es oft im Umgang mit historischen Daten bei Modelländerungen.

Data Vault

Die Data-Vault-Methodik ist neuer als die 3NF und das Sternschema, aber mittlerweile bereits seit einem Jahrzehnt etabliert. Schlüssel (Hubs), Beziehungen (Links) und Kontext (Satelliten) werden voneinander getrennt gespeichert, aber stets integriert. Dies bringt unter anderem Vorteile bezüglich Skalierbarkeit, Nachvollziehbarkeit und Automatisierbarkeit, da die Entitäten klaren Mustern folgen *[4]*. Durch die Vielzahl an Tabellen, die in einem Data-Vault-Modell üblicherweise entstehen, ist das Modell ungeeignet, um End-Usern direkt für Abfragen zu dienen. Doch gilt dies auch für ChatGPT und Co?

One Big Table

Der Ansatz des One Big Table (OBT), auch als „Wide Table“ oder „Flat Table“ bezeichnet, basiert darauf, die Daten in einer einzigen Tabelle zu speichern, statt sie in separate Tabellen aufzuteilen. Der Vorteil liegt darin, dass für Abfragen dann keine Joins mehr nötig sind, was zu Performance-Vorteilen führen kann *[5]*. Beliebt sind solche stark denormalisierten Strukturen vor allem im Data-Science-Bereich. Für BI-Werkzeuge, die keine semantische Schicht haben, sondern nur einfache Visualisierungs- und Pivot-Funktionen bieten, sind diese Strukturen ebenfalls gut geeignet. Allerdings ist es bei nicht trivialen Datenkonstellationen in der Regel nicht sinnvoll möglich, tatsächlich alle Daten in einer Tabelle zu speichern, sondern es ergeben sich mehrere denormalisierte Tabellen. Existiert keine vorherige Integrationsschicht, führt dies meist dazu, dass kleine Unterschiede in der Transformationslogik entstehen und die Daten nicht mehr abgleichbar sind. Flache Tabellen sollten daher nicht alleinstehen, sondern bei Bedarf auf Basis eines Kern-DWH als zusätzliche Schicht erstellt werden.

Versuchsaufbau

Als Beispielszenario wird ein fiktives Datenmodell verwendet, das grob an das bekannte TPC-H-Modell *[6]* angelehnt ist. Ein neues fiktives Modell wird deshalb gewählt, weil das TPC-H-Modell durch etliche Diskussionen und Beispiele im Internet den LLMs schon sehr bekannt sein dürfte, sodass diese bezogen auf das TPC-H-Modell möglicherweise bessere Antworten liefern, als es für ungesehene Datenmodelle der Fall ist. Mit einem neuen Datenmodell ist seitens der LLMs die Anwendung des gelernten Wissens auf ein neues Szenario nötig, statt nur bereits publizierte Lösungen zusammenzufassen. Das Grundmodell in 3NF wird dann in die weiteren Modellierungsformen überführt. Die Spalten sind grundsätzlich entsprechend benannt und die Beziehungen mit Fremdschlüsseln definiert. Um die Auswirkungen zu testen, wird an manchen Stellen davon abgewichen:

- In der Beziehung zwischen Bestellung und Verkäufer wird kein Fremdschlüssel definiert, die ID-Spalte heißt aber in beiden Tabellen gleich.
- In der Tabelle Rücksendung wird PRODUKT_ID als PRD_ID abgekürzt. Es werden Varianten getestet, in denen entweder ein Fremdschlüssel von PRD_ID auf PRODUKT_ID in der Tabelle Produkt angelegt ist oder dieser fehlt.
- In der Bestellposition wird RABATT_PROZENT nicht ausgeschrieben, sondern als RABATT_PRZ abgekürzt.

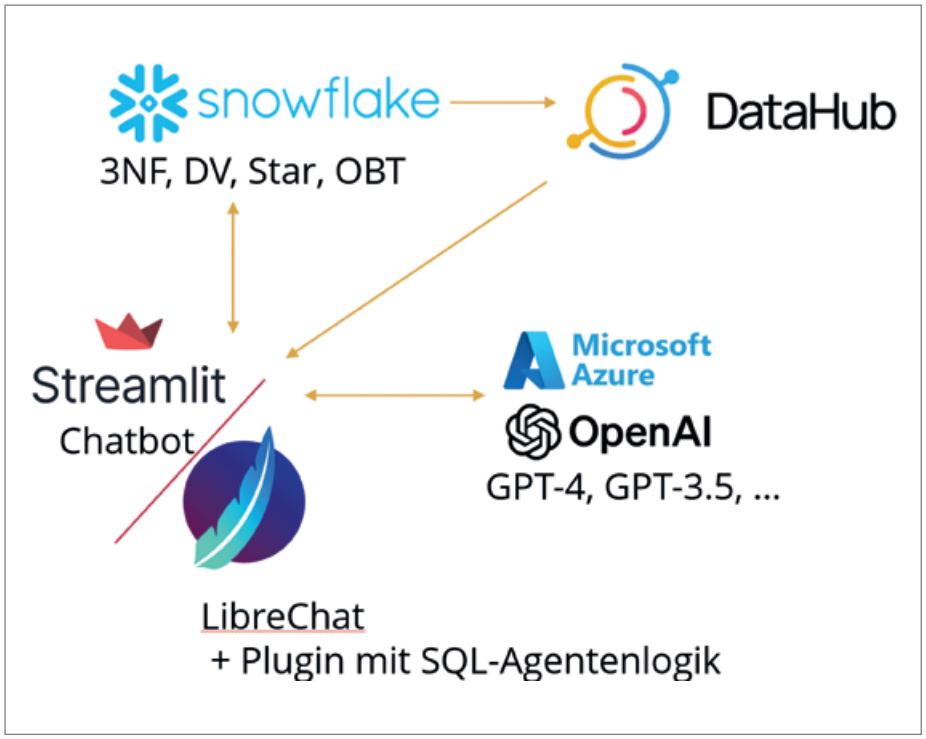


Abbildung 1: Versuchsaufbau (Quelle: Tobias Otte)

- In der Tabelle Wareneingang wird der Einkaufspreis als EK abgekürzt, aber mit einem Spaltenkommentar „Einkaufspreis“ versehen.
- Vom Sternschema werden noch zwei weitere Modellvarianten erzeugt, in denen alle Schlüsselbeziehungen entfernt werden und weitere Tabellen- und Spaltenbezeichnungen abgekürzt werden. Für eine der beiden Varianten werden im Datenkatalog (Datahub) Kommentare hinterlegt, welche die vormalige Bedeutung wieder auflösen, zum Beispiel FACT_BSP wird als FACT_BESTELLPOSITION kommentiert.

Der Versuchsaufbau wird in *Abbildung 2* dargestellt. Die in *Abbildung 3 bis 6* dargestellten Beispieldatenmodelle liegen in einer Snowflake-Datenbank. Als Datenkatalog wird DataHub verwendet. Dieser kommt vor allem für die abgewandelten Sternschemas zum Tragen, wo für ein

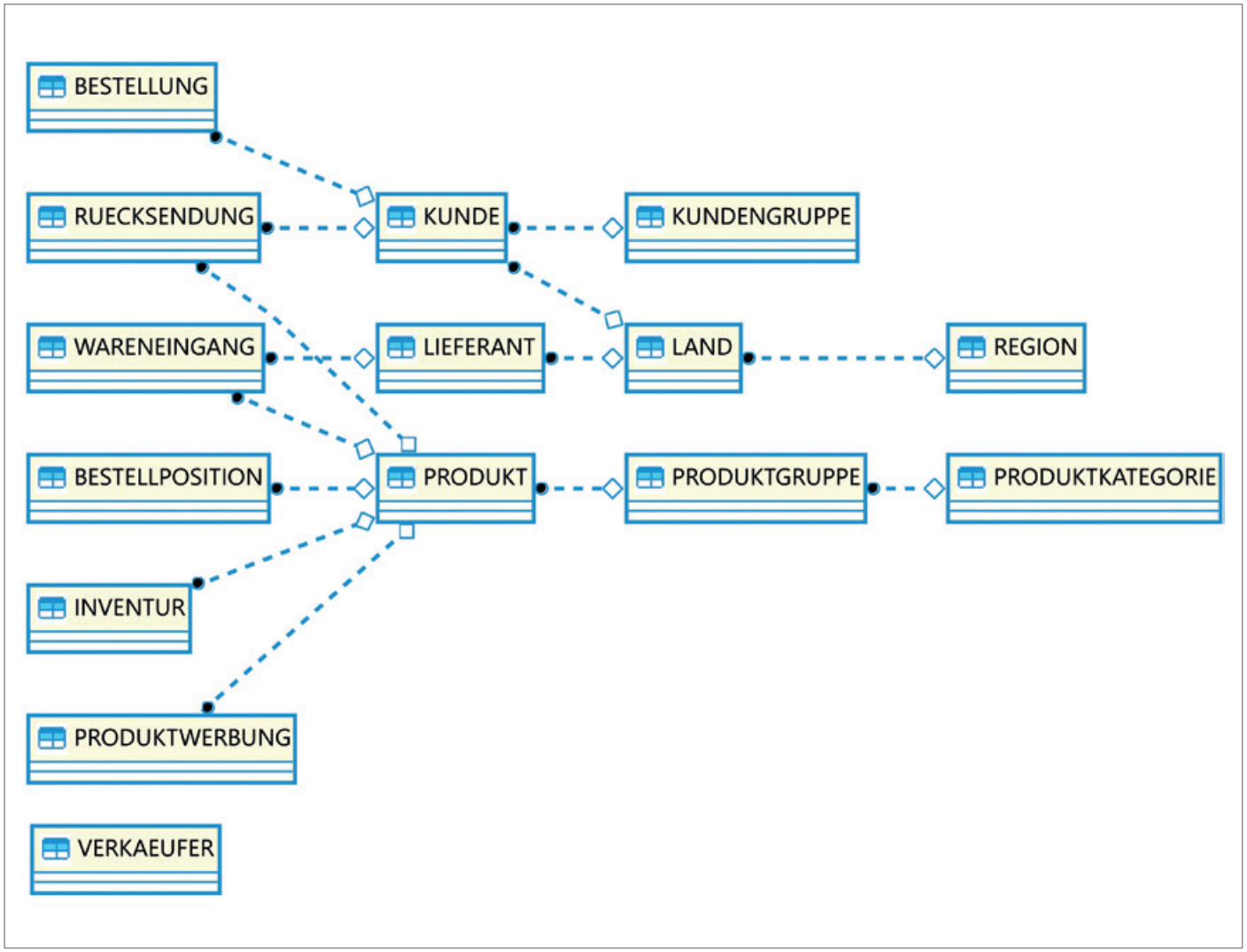


Abbildung 2: 3NF-Tabellen (Quelle: Tobias Otte)

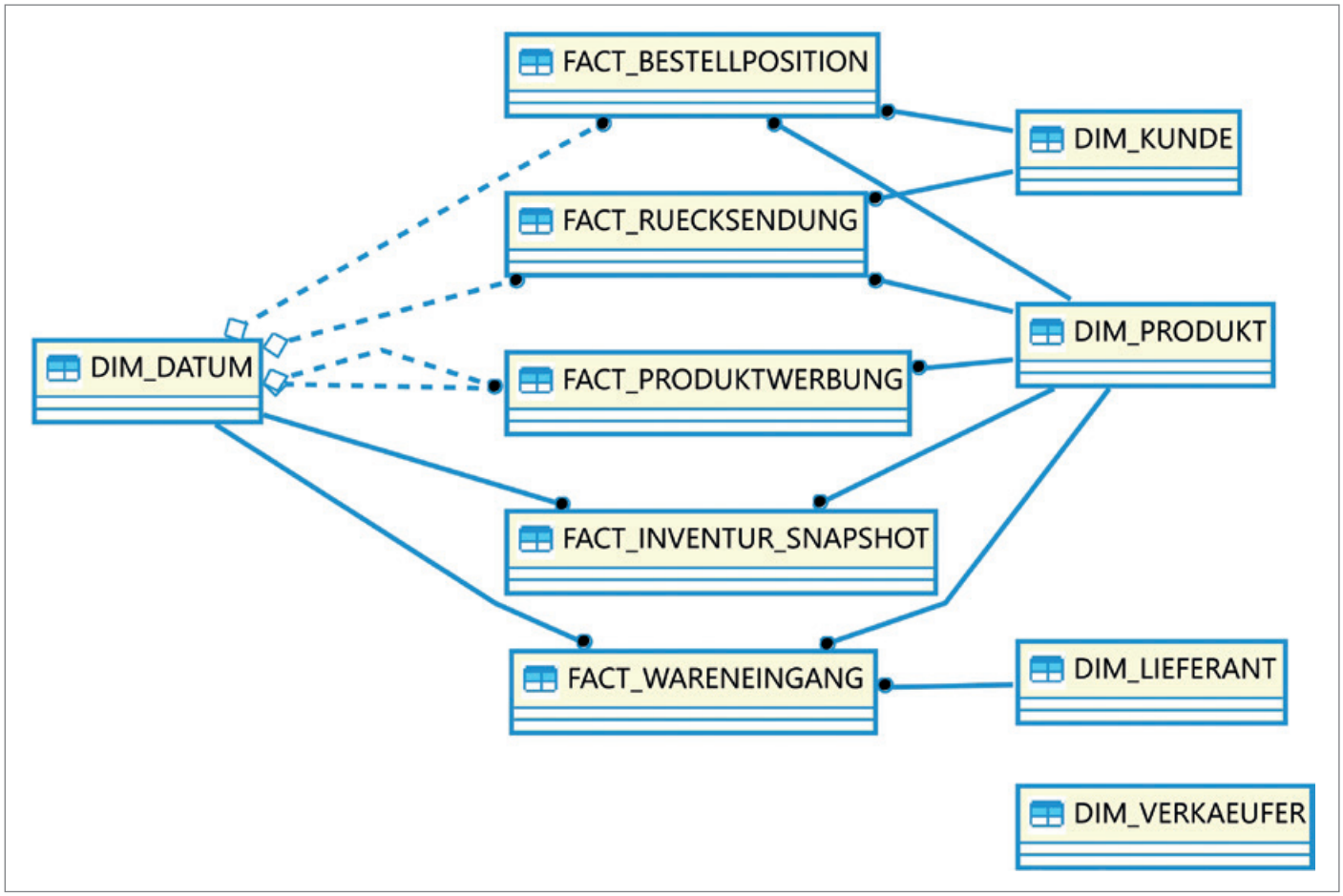


Abbildung 3: Star-Tabellen (Quelle: Tobias Otte)

kryptisches Modell im Datenkatalog hilfreiche Kommentare hinterlegt wurden. Die LLMs werden über die Azure OpenAI API angebunden. Folgende LLMs werden im Rahmen der Tests benutzt:

- gpt-4o
- gpt-4o-mini
- gpt-4-turbo
- gpt-4
- gpt-3.5-turbo

Als Frontend wurde einerseits ein lokaler Streamlit-Chatbot und andererseits ein LibreChat-Plugin mit einer Agentenlogik umgesetzt. Dem Streamlit-Chatbot werden alle relevanten Metadaten (Tabellen, Spalten, Datentypen, Beziehungen, Kommentare) im Prompt mitgegeben. Im Fall des LibreChat-Plugins stehen die gleichen Metadaten zur Verfügung, aber ein SQL-Agent ruft diese in mehreren Schritten ab. So werden zunächst nur grobe Informationen verwendet, um die richtigen Tabellen für die Abfrage zu ermitteln und erst im nächsten Schritt werden die detaillierteren Spalteninformationen ermittelt. Dieses iterative Vorgehen ermöglicht es, auch deutlich komplexere Modelle als das für diesen Artikel erstellte Beispiel zu verarbeiten, bei denen die Metadaten zu umfangreich für den Kontext der Modelle wären. Die Technologien wurden aufgrund ihrer leichten Verfügbarkeit gewählt und sind für die betrachtete Fragestellung nicht zwingend. Es könnten ebenso gut andere Komponenten verwendet werden. Ziel war es nicht, eine optimale Text-to-SQL-Anwendung zu bauen, sondern vornehmlich die Unterschiede in den Abfragen auf verschiedene Datenmodelle zu evaluieren.

Auf Basis der Beispieldatenmodelle wurden die in *Tabelle 1* dargestellten Testfragen erstellt, anhand derer überprüft wurde, wie die Ergebnisse der LLMs auf den verschiedenen Datenmodellen ausfallen.

Ergebnisse

Grundsätzlich sind LLMs gut in der Lage, SQL-Abfragen zu erstellen. Vor allem dann, wenn Schlüsselbeziehungen definiert sind und Spalten sinnvoll benannt sind. Wenn dies nicht gegeben ist, wird in der Regel trotzdem eine Antwort erzeugt, aber es werden falsche Annahmen getroffen oder Spalten halluziniert, sodass die Abfragen fehlerhaft sind. Schwierigkeiten gab es in allen Modellvarianten bei der gleichzeitigen Nutzung von fachlicher und technischer Gültigkeit in Frage 3. Hier kam es besonders häufig zu fehlerhaften Ergebnissen. Auf die in den durchgeführten Experimenten beobachteten Unterschiede zwischen den Modellierungstechniken wird im Folgenden eingegangen.

Auswertung Dritte Normalform

Auf Basis des 3NF-Modells konnten für die meisten Beispielabfragen direkt beim ersten Versuch richtige Ergebnisse erzielt werden. Probleme gab es bei sehr komplexen Abfragen und bei der Beachtung der Gültigkeit mittels GUELTIG_VON und GUELTIG_BIS. Hier kann eine zusätzliche Anweisung im Prompt helfen.

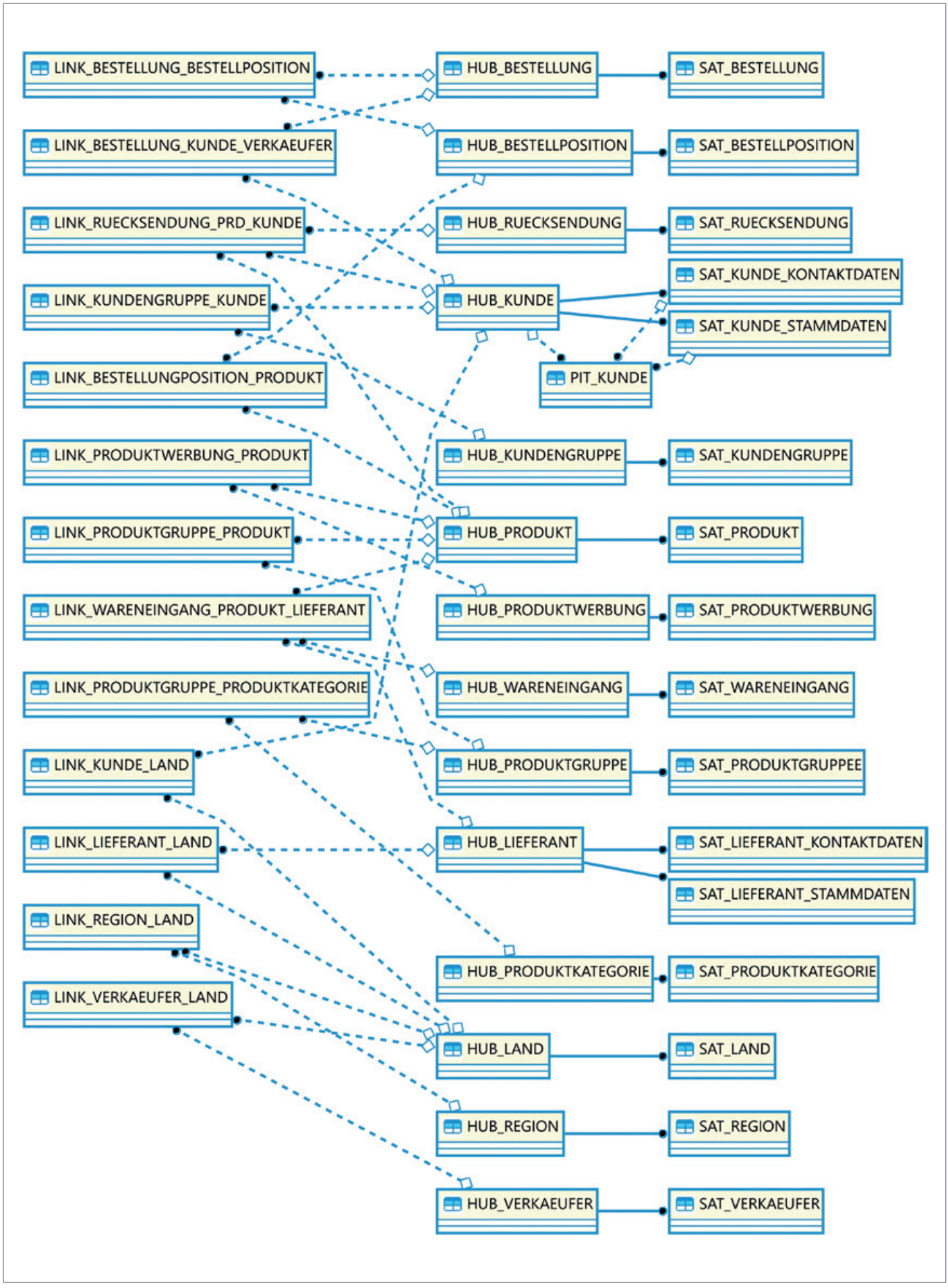


Abbildung 4: DV-Tabellen (Quelle: Tobias Otte)

Nr.	Frage	Zweck
1	Erstelle auf Basis des gegebenen Datenmodells eine SQL-Abfrage, um die meistbestellten Produkte im Jahr 2022 zu ermitteln.	Einfache Basisfrage.
2	Erstelle auf Basis des gegebenen Datenmodells eine SQL-Abfrage, um die Kunden zu ermitteln, die im Jahr 2022 mindestens 20 Bestellungen hatten und mehr als 10 Produkte zurückgesendet haben.	Gesteigerte Komplexität. Prüfung der Beziehung zwischen Rücksendung und Produkt, bei welcher der Spaltenname abgekürzt wurde.
3	Erstelle auf Basis des gegebenen Datenmodells eine SQL-Abfrage, um die Kunden zu ermitteln, die im Jahr 2022 mindestens 20 Bestellungen hatten und mehr als 10 Produkte zurückgesendet haben. Erstelle die Abfrage so, dass sie die Daten wiedergibt, die zum 31.12.2022 gültig waren.	Einbeziehung der technischen Gültigkeit.
4	Erstelle auf Basis des gegebenen Datenmodells eine SQL-Abfrage, welcher Verkäufer im Jahr 2022 die meisten Produkte verkauft hat.	Prüfung der Beziehung zwischen Bestellung und Verkäufer, für die keine Fremdschlüsselbeziehung definiert ist. Prüfung, ob aus „verkauft“ auch Bestellungen erkannt werden.
5	Erstelle auf Basis des gegebenen Datenmodells eine SQL-Abfrage, um den durchschnittlichen Rabatt der Bestellungen im Jahr 2023 aus Deutschland in Prozent zu ermitteln.	Prüfung, ob die Spalte RABATT_PRZ genutzt wird.
6	Erstelle auf Basis des gegebenen Datenmodells eine SQL-Abfrage, um die Kunden mit dem höchsten Einkommen zu ermitteln.	Prüfung, ob erkannt wird, dass gar keine Informationen zum Einkommen vorliegen.
7	Erstelle auf Basis des gegebenen Datenmodells eine SQL-Abfrage, wie sich Produkte aus der Kategorie Lebensmittel im Jahr 2020 verkauft haben, während sie beworben wurden, im Vergleich zu Zeiten, wo sie nicht beworben wurden.	Prüfung, ob die Spalten für den fachlichen Gültigkeitszeitraum der Produktwerbung richtig einbezogen werden.
8	Erstelle auf Basis des gegebenen Datenmodells eine Abfrage, bei welchem Produkt es die größte Abweichung im Einkaufspreis der Wareneingänge zwischen verschiedenen Lieferanten gibt.	Prüfung, ob die Spalte Einkaufspreis anhand des Kommentars richtig erkannt wird.
9	Erstelle auf Basis des gegebenen Datenmodells eine SQL-Abfrage, um für jeden Monat von Januar 2021 bis Dezember 2021 den kumulierten Umsatz und zum Vergleich den kumulierten Umsatz bis zum Vormonat darzustellen.	Nutzung analytischer Funktionen prüfen.
10	Erstelle auf Basis des gegebenen Datenmodells eine SQL-Abfrage, um das durchschnittliche Alter der Kunden aus Frankreich zu ermitteln.	Prüfen, ob die Granularität der Tabelle beachtet wird, da Kunden mehrfach bestellt haben könnten.

Tabelle 1: Testfragen

Auswertung Sternschema

Abfragen auf dem Sternschema funktionierten grundsätzlich sehr gut. Bei Abfragen, die mehrere Faktentabellen benötigen, kommt es teilweise dazu, dass versucht wird, die Faktentabellen direkt miteinander zu joinen, obwohl keine Fremdschlüsselverbindung definiert ist und die Verbindung im Sternschema immer über die konformen Dimensionen laufen sollte. Mit der zusätzlichen Anweisung, Faktentabellen nicht direkt zu verbinden, lässt sich das Verhalten verbessern.

Durch die zwei zusätzlichen Modellvarianten im Sternschema, bei denen jeweils keine Fremdschlüssel definiert und die Bezeichnungen kryptsicher gewählt wurden, konnte gezeigt werden, dass entsprechende Kommentare im Datenkatalog zu besseren Ergebnissen führen, als wenn diese nicht vorhanden sind.

Auswertung Data Vault

Mit dem Data-Vault-Modell kamen die LLMs in den durchgeführten Experimenten schlecht zurecht. Es wurden oft nicht alle nötigen Links einbezogen und stattdessen versucht, einen Join direkt zwischen Satelliten zu erstellen. Außerdem kam es vor, dass beschreibende Daten, wie der Name des Kunden aus dem Hub, selektiert wurden, obwohl sich dieser im Satelliten befindet und der Satellit sogar Teil der Abfrage war *(siehe Listing 1)*.

Auswertung One Big Table

Im OBT-Modell wurden für die meisten Abfragen gute Ergebnisse erzielt. Da im Beispiel nur vier Tabellen definiert wurden, ist dieses wenig komplex. Vermutlich würden die Ergebnisse tendenziell schlechter, wenn es mehr Tabellen gäbe und mehr Uneindeutigkeit herrschte, welche Tabelle für die jeweilige Abfrage die beste Wahl ist. Erwartungsgemäß gibt es Schwierigkeiten bei der Beachtung der Granularität der Tabellen, sodass zum Beispiel bei der Ermittlung des durchschnittlichen Alters (Frage 10) nicht beachtet wird, dass Kunden mehrfach bestellt haben könnten *(siehe Listing 2)*. Darüber hinaus kam es zur Verwendung falscher Spalten: Zum Beispiel wurden die in der Tabelle BESTELLUNG enthaltenen Spalten PREIS (als Preis der Bestellung) und PREIS_PRODUKT (als Preis eines Produktes) nicht immer korrekt interpretiert.

```
SELECT K.NAME AS KundeName,
COUNT(DISTINCT B.BESTELLUNG_HK) AS AnzahlBestellungen, COUNT(DISTINCT
R.RUECKSENDUNG_HK) AS AnzahlRuecksendungen
FROM LLM_MODELLIERUNG.DEMO_DV.HUB_KUNDE K
JOIN LLM_MODELLIERUNG.DEMO_DV.SAT_KUNDE_STAMMDATEN KS
ON K.KUNDE_HK = KS.KUNDE_ HK JOIN
```

Listing 1: Fehlerhafter Zugriff auf HUB_KUNDE

```
SELECT
AVG(DATEDIFF(YEAR, GEBURTSDATUM_KUNDE,
CURRENT_DATE)) AS DURCHSCHNITTLICHES_
ALTER
FROM
LLM_MODELLIERUNG.DEMO_OBT.BESTELLUNG
WHERE
LAND_NAME_KUNDE = ‚Frankreich‘;
```

Listing 2: Fehlerhafte Abfrage auf durchschnittliches Alter

```
select K.NAME, K.TELEFON, K.ADRESSE
from DEMO_3NF.KUNDE K
order by K.KUNDE_ID desc
limit 10
```

Listing 3: Halluzinierte Antwort zur Frage nach dem höchsten Einkommen

Abbildung 5: OBT-Tabellen (Quelle: Tobias Otte)

Unterschiede zwischen den LLMs

Grundsätzlich lässt sich sagen, dass die neueren und größeren LLMs in der SQL-Erzeugung, wie auch bei vielen anderen Aufgaben, besser geworden sind. Über die Zeit waren aber bei verschiedenen Versionen eines Modells durchaus Unterschiede zu beobachten, sodass etwa gpt-4 nicht von Anfang an besser in der SQL-Erzeugung war als gpt-3.5-turbo. Innerhalb der OpenAI-Modelle ist für viele Abfragen das sehr schnelle und günstige gpt-4o-mini mittlerweile ausreichend. Mit steigender Kontextgröße passieren mit diesem Modell jedoch eher Fehler und Halluzinationen als mit dem größeren gpt-4o, das generell eine gute Qualität liefert. Interessant ist, dass gpt-4 (ohne turbo oder o) im Gegensatz zu anderen Modellen oft in der Lage ist, zu erkennen, wann eine Frage nicht beantwortbar ist.

Beispielsweise wird bei der Frage nach dem höchsten Einkommen von gpt-4 folgende Antwort ausgegeben:

„Entschuldigung, aber ich kann diese Frage nicht beantworten, da es in den bereitgestellten Tabellen und Spalten keine Informationen über das Einkommen der Kunden gibt."

Andere LLMs geben in der Regel einen SQL zurück und ordnen die Daten dabei nach einer arbiträr gewählten Spalte, die aber nichts mit dem Einkommen zu tun hat *(siehe Listing 3)*.

Wie lassen sich Modelle verbessern, damit LLMs bessere Abfragen erstellen können?

Erst durch saubere Metadaten werden Daten zu einem Wettbewerbsvorteil. Damit sowohl LLMs als auch Anwenderinnen und Anwender gute Abfragen erstellen können, sollte geprüft werden, ob im Modell noch Verbesserungsbedarf hinsichtlich der folgenden Punkte besteht:

- Sind Primär- und Fremdschlüssel definiert? Sie müssen dabei nicht durchgesetzt werden, sondern es hilft bereits, wenn sie definiert sind. Wenn keine Schlüssel definiert sind, ist es sehr hilfreich, wenn die Schlüsselspalten auf beiden Seiten der Beziehung gleich benannt sind.
- Sprechende Tabellen und Spaltennamen: Entsprechen die Modelle dem üblichen Sprachgebrauch der Anwenderinnen und Anwender oder sind sie aus Quellsystemen übernommen?
- Sind Kommentare/Label definiert? Diese können zusätzliche Hinweise für die Verwendung der Modelle geben und auch dann helfen, wenn es innerhalb der Organisation unterschiedliche Begrifflichkeiten für gleiche Sachverhalte gibt.
- Existieren im abgefragten Schema neben dem eigentlichen Modell noch Tabellen aus ETL-Prozessen oder aus alten Versionen? Dann sollten diese entfernt oder durch eingeschränkte Rechte für Abfragen unsichtbar gemacht werden.
- Ist ein Datenkatalog vorhanden, hilft es, dort Verwendungsbeispiele und beispielhafte Wertemengen zu hinterlegen.

Fazit

Die durchgeführten Versuche haben gezeigt, dass die Grundsätze der Datenmodellierung in Zeiten von LLMs nicht weniger wichtig geworden sind, sondern sogar noch an Bedeutung gewonnen haben. Wohldefinierte Modelle mit klaren Beziehungen und entsprechend benannten Spalten helfen sowohl Menschen als auch LLMs dabei, korrekte Abfragen zu erstellen. Die allgemeine Empfehlung, Anwenderinnen und Anwender nicht direkt auf Data-Vault-Modelle zugreifen zu lassen, sondern diese im Kern-DWH zu verwenden und darauf Sternschemata in den Information Marts aufzubauen, gilt aktuell auch für LLMs.

Es hilft sowohl LLMs als auch Menschen, wenn ein Datenkatalog angebunden wird und dort Beschreibungen und Verwendungsbeispiele hinterlegt werden. Dies ist vor allem dann dienlich, wenn Systeme über viele Jahre gewachsen sind und nicht alle Entitäten wohl benannt sind. So lässt sich das große Potenzial von LLMs zur Automatisierung von Aufgaben und zur Erleichterung der Arbeit nutzen, statt mühsam falsch generierte Abfragen korrigieren zu müssen. In jedem Fall bleibt es wichtig, dass Anwenderinnen und Anwender weiterhin ein Verständnis für die Abfragen und die Daten haben, um der gegebenenfalls halluzinierten Lösung eines LLMs nicht blind zu vertrauen.

Quellen

[1] Magazin: BI Spektrum 04/2022
[2] Inmon, W. H. / Imhoff, C. / Sousa, R.: Corporate Information Factory. 2. Auflage, Wiley 2001
[3] Kimball, R. / Ross, M.: The Data Warehouse Toolkit. 3. Auflage, Wiley 2013
[4] Linstedt, D. / Olschimke, M.: Building a Scalable Data Warehouse with Data Vault. Morgan Kaufmann 2015
[5] Blog: *https://www.fivetran.com/blog/star-schema-vs-obt, abgerufen am* 14.12.2024
[6] Website: *https://www.tpc.org/tpch/default5.asp, abgerufen am 14.12.2024*

Über den Autor

Tobias Otte arbeitet als Beratender Manager bei der viadee Unternehmensberatung AG. Er ist Co-Leiter des Kompetenzbereichs Data & AI und hat langjährige Projekterfahrung mit allen Facetten von Datenprojekten. Als Projektmanager und Experte unterstützt er Kundinnen und Kunden unter anderem bei der Modernisierung ihrer Data-Warehouse-Systeme und bei der Einführung neuer Methoden und Technologien wie generativer KI.

Tobias Otte
Tobias.Otte@viadee.de

Die Zukunft des Codings? Jetbrains AI Assistant und GitHub Copilot im Duell

Bastian Weinlich und Semjon Mössinger, WPS – Workplace Solutions

KI-Coding-Assistenten halten immer mehr Einzug in die moderne Software-Entwicklung. Das ist beim anhaltenden Hype um KI nicht weiter verwunderlich: Versprechen die Hersteller bei der Nutzung ihrer Produkte doch Produktivitätssteigerungen von über 50 Prozent [1].

Der theoretische Teil

Doch auf welches Tool sollten EntwicklerInnen und Unternehmen im Jahr 2025 setzen? Der Platzhirsch „GitHub Copilot" ist aufgrund seiner starken Medienpräsenz häufig die erste Wahl, ohne dass Alternativen evaluiert werden. Wir haben daher den GitHub Copilot mit dem Konkurrenzprodukt „AI Assistant" aus dem Hause JetBrains verglichen und stellen unsere Ergebnisse vor. Außerdem geben wir anhand eines Fallbeispiels Tipps zur praktischen Arbeit mit KI-Coding-Assistenten.

Die Kontrahenten

Der GitHub Copilot ist seit Mitte 2022 öffentlich verfügbar und ist damit eines der ersten Tools gewesen, das ein vortrainiertes Sprachmodell für die Auto-Vervollständigung verwendet hat. Inzwischen ist der Feature-Umfang deutlich gewachsen – genauso wie die Anzahl der IDEs, in denen der Copilot als Plugin angeboten wird (inklusive der JetBrains IDEs). Wichtig ist es aber, dabei zu erwähnen, dass zahlreiche Features nur exklusiv für Visual Studio und Visual Studio Code verfügbar sind. Das ist nicht weiter verwunderlich, wenn man bedenkt, dass GitHub inzwischen zum Microsoft-Konzern gehört.

Auf der anderen Seite steht der AI Assistant von JetBrains, der vor allem durch IDEs wie IntelliJ oder Resharper, aber auch durch die Programmiersprache Kotlin bekannt geworden ist. Diesen Assistenten gibt es ebenfalls als Plugin – jedoch nur für die JetBrains-IDEs. Der AI Assistant ist mit gut einem Jahr auch deutlich jünger als die Konkurrenz von GitHub. Das macht sich ebenfalls im geringeren Funktionsumfang bemerkbar, wie sich später zeigen wird.

Inline Code Editing

Wir betrachten zunächst eine der Kernaufgaben von KI-Coding-Assistenten: Das automatische Schreiben und Anpassen von Quellcode innerhalb des Editorfensters. Dazu gehört auch die Erstellung von Testfällen sowie Unterstützung bei Code-Refactorings.

Die grundlegende Funktion, die beide KI-Coding-Assistenten dafür anbieten, ist eine mehr oder weniger intelligente Auto-Vervollständigung von Code, den man gerade tippt. Dieses Feature nennt man auch Code Completion. Ist die Code Completion aktiviert, erscheint während des Programmierens bei beiden Kandidaten in dezenter Schrift ein Vorschlag, wie sich der Code höchstwahrscheinlich fortsetzt. Solche Vorschläge können sich auch über mehrere Zeilen erstrecken und sind umso besser, je mehr Kontextwissen die Assistenten zur Verfügung haben. Kontext wird dabei unter anderem aus den Code-Zeilen vor und hinter dem Text-Cursor berücksichtigt – meistens werden noch weitere Code-Blöcke innerhalb der geöffneten Datei sowie die weiteren offenen Tabs berücksichtigt. Besonders hilfreich für die KI-Coding-Assistenten sind auch erklärende Code-Kommentare zum Beispiel vor einer neu zu schreibenden Methode. Eine saubere Code-Dokumentation lohnt sich also nicht mehr nur allein zur Erhöhung der Lesbarkeit von Quellcode, sondern auch als Hilfestellung für die KI.

Beide Kontrahenten erzeugen meist nützliche Code-Schnipsel, die man entweder direkt übernehmen kann oder nur noch leicht anpassen muss. Während der JetBrains AI Assistent kurz nach seinem Release Anfang 2024 hier noch häufiger mal daneben lag, hat sich mit der Version 2024.2 die Qualität seiner Vervollständigungen deutlich gesteigert. Das ist vermutlich eine Folge der Einführung eines speziellen, selbsttrainierten LLMs namens „Mellum" aus eigenem Hause *[2]*. Damit kommt JetBrains bei der Code Completion qualitativ nahe an den Copilot heran, dessen Vorschläge manchmal noch gefühlt einen Tick treffender sind. Bei der Bedienung gibt es jedoch Unterschiede: Der Copilot macht viel häufiger von sich aus Code-Vorschläge, während man den AI Assistant via Hotkey öfters explizit darum bitten muss. Umgekehrt können die sofortigen Vorschläge des Copilots auch aufdringlich wirken. Das gilt insbesondere, wenn noch gar nicht ersichtlich ist, was man mit dem Quellcode eigentlich ausdrücken will. Letztendlich ist es Geschmackssache, welches Verhalten man hier bevorzugt.

Eine Alternative für die Code-Generierung innerhalb des Editor-Fensters ist die sogenannte „Inline Code Generation". Mittels eines kurzen Prompts lässt sich in einem Pop-up-Fenster die gewünschte Funktionalität beschreiben, die das Large Language Model dann in Quellcode überführt. Auch hier nehmen sich die beiden Kontrahenten hinsichtlich erzeugter Code-Qualität und User Experience nicht sonderlich viel. Bei der Inline-Anpassung von bestehendem Code gefällt uns die UX des GitHub-Copilots jedoch etwas besser: Änderungen werden direkt in der geöffneten Datei an der editierten Stelle angezeigt. Der AI Assistant öffnet die Diff-Ansicht dagegen in einem eigenen Fenster, was umständlich wirkt.

Insgesamt sind die Unterschiede zwischen den beiden Kontrahenten beim Schreiben und Anpassen von Quellcode innerhalb des Editorfensters jedoch marginal – daher lassen wir den Vergleich in dieser Kategorie mit Unentschieden ausgehen.

Chat with your Code

Beide Tools bieten neben den bisher betrachteten Funktionen im Editorfenster auch die Möglichkeit, sich in einem separatem Chat-Fenster Fragen zum Code beantworten zu lassen oder dort Code-Vorschläge zu generieren. Dabei greifen die Kontrahenten für die Erzeugung der Antworten auf bestehende LLMs zurück: Beide KI-Coding-Assistenten bieten jeweils die bekannten Modelle GPT-4o von OpenAI sowie Gemini von Google an. Der Copilot hat zusätzlich Claude Sonnet im Repertoire, während man sich beim AI Assistant mittels Ollama sogar ein lokal laufendes Modell einbinden kann.

Die Nutzung mächtiger Sprachmodelle als Motor der Code-Generierung hilft dabei, auch komplexere Aufgaben an die KI zu delegieren. Dazu hat man als Programmierer mit der Chat-Funktion die Möglichkeit, der KI weiteren Kontext in Form von Code-Referenzen zu geben, Nachbesserungen anzufordern oder sich den erzeugten Code erst mal erklären zu lassen, bevor man ihn in die Codebase übernimmt.

Das funktioniert bei einfachen Verständnisfragen zu einzelnen Quellcode-Dateien bei beiden Tools sehr gut. Auch Aufgaben wie automatisches Bug-Fixing, die Optimierung der Code-Qualität oder

das Schreiben von Dokumentation gelingen oft in erstaunlich hoher Qualität. Die besten Ergebnisse erhält man, wenn man den als Kontext übergebenen Code weder zu klein noch zu groß wählt: Die Größenordnung einer Methode oder Code-Blöcke mit zweistelliger Zeilenanzahl haben sich unserer Erfahrung nach hier als Sweetspot erwiesen: Größere Blöcke führen häufiger zu Halluzinationen, während sich bei sehr kleinen Code-Schnipseln der LLM-Einsatz oft nicht lohnt – hier ist man per Hand oftmals schneller.

Der IntelliJ AI Assistant bietet im Zusammenhang mit der Chat-Funktion ein besonders nützliches Feature: Man kann sich eine eigene kleine Prompt-Bibliothek aufbauen, indem man Prompt-Templates speichert und sie später wiederverwendet. Das ist vor allem bei der Verbesserung der Code-Lesbarkeit oder der Performance hilfreich.

Insgesamt bietet die Chat-Funktion des GitHub Copilot jedoch zahlreiche mächtige Features, die dem IntelliJ AI Assistant noch fehlen. So kann man über die @workspace-Referenz gezielt Fragen zur gesamten Code-Basis stellen, ohne den Kontext in Form von Dateien oder Quell-Code-Schnipseln selbst angeben zu müssen. Und auch das Internet oder zusätzliche GitHub-Repositories können im Copilot Enterprise Plan über die Web-Such-Funktion als Wissensquelle für die Beantwortung von Chat-Fragen herangezogen werden. Schließlich lässt sich der Copilot auch noch über sogenannte „Extensions“ erweitern, um ihm noch mehr Quellen zu geben: Knapp 30 dieser Extensions sind aktuell (Stand: Dezember 2024) im entsprechenden Marketplace verfügbar. Damit lassen sich zum Beispiel Informationen aus den Atlassian Tools Jira/Confluence oder aus einer Microsoft Azure Subscription direkt im Copilot Chat abrufen.

Ein weiteres Highlight des GitHub Copilot: Multi File Editing. Damit werden Änderungen gleich über mehrere Quellcode-Dateien hinweg vorgeschlagen. Die Frage, welche Dateien man für ein neues Feature überhaupt anfassen muss, entfällt damit. Mit dieser Funktion macht der Copilot auch den neuartigen, autonomen Coding-Agentensystemen wie beispielsweise „Devin“ Konkurrenz.

Aufgrund der zahlreichen Zusatzfunktionen im Zusammenhang mit dem integrierten Chat geht der ***Rundensieg*** in dieser

```
1 @Test
2 void generateHeroName() {
3     // Arrange
4     final var heroNames = List.of("Mickey Mouse", "Donald Duck", "Goofy Goof");
5     final var indexFirstName = 0;
6     final var indexLastName = 1;
7
8     // Act
9     final var result = heroService.generateHeroName(heroNames, indexFirstName, indexLastName);
10
11    // Assert
12    assertThat(result).isEqualTo("Mickey Mouse");
13 }
```

Listing 1: Einfacher Test

```
1    public boolean checkEveryNameStartsWithACapitalLetter() {
2        final var name = heroRepository.findAll().stream()
3                .map(Hero::getName).toList();
4        return name.stream().allMatch(it -> Character.isUpperCase(it.charAt(0)));
5    }
```

Listing 2: Erweiterte Funktion zur Überprüfung, ob jeder Hero tatsächlich genau einen Vor- und einen Nachnamen hat.

```
return name.stream().allMatch(n -> n.matches("^[A-Z][a-z]* [A-Z][a-z]*$"));
```

Listing 3: Generierte Zeile 4

```
1        BinaryOperator<Integer> add = (x, y) -> x + y;
2        var composed = compose(add, add);
3        System.out.println(composed(2, 3));
```

Listing 3: Generierte Zeile 4

Kategorie – wenn es nach uns geht – an den ***GitHub Copilot***.

Beyond the Code

Neben den beiden Hauptfunktionen – dem Code-Generieren im Editor und der Chatfunktion – bieten beide Assistenten noch diverse Gimmicks, die sich an unterschiedlichen Stellen in der integrierten Entwicklungsumgebung befinden. Beide KI-Coding-Assistenten unterstützen beispielsweise beim Umbenennen von Methoden oder Variablen, indem sie unter Nutzung von umgebendem Kontext sinnvolle Namensvorschläge geben. Das ist hilfreich und macht eines der „harten" Probleme der Informatik *[3]* tatsächlich etwas weicher.

Auch bei der Nutzung von GIT gibt es Hilfestellung: Beide Assistenten schreiben eigenständig Commit-Messages anhand der geänderten Code-Stellen. Die Änderungen werden dabei meist treffend zusammengefasst. Der Fokus liegt aber immer auf der Fragestellung „Was wurde geändert?" und nicht auf den spannenderen Fragen „Warum?" und „In welchem Zusammenhang?". Besser als eine Commit-Message à la „Fixed a Bug" sind die erzeugten Texte aber allemal.

Der AI Assistant erklärt einem darüber hinaus auch beliebige Commits anhand der darin enthaltenen Code-Änderungen. Der Copilot bietet in seiner Enterprise-Variante eine ähnliche Funktion: Hier können ganze Pull Requests zusammengefasst und sogar automatisch einem Review unterzogen werden.

Weitere erwähnenswerte Features sind die automatische Übersetzung von Quellcode in eine andere Programmiersprache (AI Assistant) und Spracheingabe (GitHub Copilot). Wir können bei den „sonstigen Features" keinen klaren Favoriten ausmachen und vergeben daher in dieser Kategorie ein ***Unentschieden***.

Compliance and Code

Die Einhaltung gesetzlicher Vorschriften ist bei Unternehmen die Grundvoraussetzung, um neuartige Tools wie KI-Coding-Assistenten einzuführen. In vielen rechtlichen Teilbereichen befindet man sich mit der Nutzung dieser Tools jedoch aktuell noch in einer Grauzone, da aussagekräftige Urteile fehlen. Ein Verbot dieser neuen Werkzeuge aus Compliance-Gründen kann jedoch zum Entstehen von Schatten-IT führen oder wirkt sich negativ auf die Mitarbeiterzufriedenheit aus. Daher gilt es, einen Weg zu finden, beide Interessen auszugleichen. Von großem Vorteil ist dabei, wenn die Tools von sich aus bereits Features bieten, die das Risiko von Rechtsunsicherheiten minimieren.

Das wichtigste Kriterium dafür erfüllen beide Assistenten: Sie speichern keine Prompt-Daten auf externen Servern – zumindest nicht bei den jeweiligen Business-Tarifen (GitHub Copilot) und wenn man es nicht explizit aktiviert hat (AI Assistant). Damit ist auch eine Datennutzung für zukünftige Modelltrainings ausgeschlossen: Man braucht also keine Angst davor zu haben, dass der eigene Quellcode irgendwann als Code-Vorschlag bei der Konkurrenz landet. Das hilft auch bei Themen, die das Urheberrecht betreffen, beispielsweise, wenn man exklusive Quellcode-Rechte an Kunden verkauft.

Die andere Seite der Urheberrechtsmedaille ist die versehentliche Nutzung von unter Copyleft-Lizenz stehendem Code, ohne die entsprechenden Lizenzbedingungen einzuhalten. Derartige Code-Vorschläge sind bei beiden Assistenten nicht hundertprozentig auszuschließen. Der GitHub-Copilot prüft aber im Rahmen seines „Public Code Filters" zumindest auf Duplikate in den öffentlichen GitHub-Repositories – nicht jedoch auf anderen Open-Source-Plattformen. Der AI Assistant bietet für seine Kunden ebenfalls einen „Public-Clone-Filter" *[4]*. Dieser ist aber lediglich in der großen Enterprise-Variante verfügbar und aktuell auch nur für die Programmiersprachen Java, Kotlin und Python. JetBrains eigenes Modell „Mellum", das für die Code Completion genutzt wird, wurde nach eigenen Angaben nur auf Code unter „permissive License" trainiert *[5]*, sodass man hier auf der sicheren Seite sein sollte. Bei beiden Assistenten bleibt jedoch ein Restrisiko.

GitHubs Mutterkonzern Microsoft versichert mit einer sogenannten „IP Indemnity" jedoch, dass im Falle eines Urheberrechtsproblems im Zusammenhang mit dem Copilot die Kosten für den entstandenen Schaden übernommen werden *[6]*. Beim AI Assistant findet man zu diesem Thema lediglich den Hinweis „Protection from IP liability" ohne weitere Erläuterung auf der Webseite der Enterprise-Edition des AI Assistant.

Um Probleme mit dem Datenschutz und Geschäftsgeheimnissen zu vermeiden, kann man beim Copiloten bestimmte Dateien und Ordner seines Repositories im Stile eines .gitignore-Files vom Kontext ausschließen. Das heißt, die Inhalte dieser Dateien werden nicht an externe Server weitergeleitet. Der AI Assistant bietet auf der anderen Seite ein Feature, um derartige Probleme gänzlich zu vermeiden: Die Nutzung von selbst gehosteten Modellen mittels Ollama. Ollama ist ein Tool, um auf einfache Art und Weise eigene große Sprachmodelle auf dem lokalen Rechner laufen zu lassen. Für gute Ergebnisse setzt das aber leistungsstarke Grafik-Hardware voraus – daher dürfte das für viele Unternehmen nur eine theoretische Option sein.

Hinsichtlich Anwendungssicherheit bietet der GitHub Copilot noch einen sogenannten „Security-Vulnerability-Filter": Vorgeschlagener Code mit potenziellen Sicherheitslücken wird damit automatisch verworfen und neu generiert.

Zumindest wenn Self-hosting-Modelle keine Option sind, hat der GitHub Copilot damit auch in dieser Kategorie unserer Meinung nach die Nase vorne.

Ergebnis

Insgesamt besitzt der GitHub Copilot aufgrund seiner Reife einige Zusatzfunktionen, die dem AI Assistant noch fehlen: Beispielsweise die automatische Kontextsuche mittels @workspace oder Multi-File-Editing. Auch bei Compliance-Themen bietet der Copilot mit seinem Public-Code-Filter, der IP Indemnity und dem Security-Vulnerability-Filter mehr als die Konkurrenz von JetBrains, die ähnliche Features erst in der teuren Enterprise-Variante anbietet. Daher geht unser Vergleich in der Summe zu Gunsten des GitHub Copilot aus.

Der AI Assistant braucht sich aber nicht zu verstecken: In den Basisfunktionen ist er inzwischen gleichauf mit dem Copilot und bietet mit seiner Prompting Library und Selfhosting-Modellen auch spannende Alleinstellungsmerkmale. Insbesondere wenn man sowieso schon

stark auf JetBrains IDEs setzt, sollte man den AI Assistant in Betracht ziehen. Denn einige Features des GitHub Copilots sind im Plugin für die JetBrains IDEs auch gar nicht enthalten. Dies könnte in der Entscheidung, welcher KI- Coding-Assistant besser zum Unternehmen passt, den Ausschlag schließlich doch zu Gunsten von JetBrains geben.

Der praktische Teil

KI-Coding-Assistenten in der Praxis: Ein Fallbeispiel

Als Fallbeispiel wollen wir eine Applikation betrachten, mit der man eine Liste von Helden verwalten kann. Als neues Feature soll unsere Applikation beim Klick auf den Button automatisch einen neuen Helden (im Code „Hero“) in die Datenbank hinzufügen. Dabei soll der Name durch Rekombination vorhandener Vor- und Nachnamen zufällig entstehen. Aus den vorhandenen Namen „Mickey Mouse“ und „Daisy Duck“ könnte also beispielsweise der neue Name „Mickey Duck“ entstehen. In unserem Beispiel betrachten wir dafür nur das Java-Backend.

Tests und Auto-Completion

Wie fängt man mit AI Assisted Coding an? Im Prinzip genauso wie ohne Assistenz: Wir entscheiden uns dazu, testgetrieben zu entwickeln und starten mit den Unittests. Die Tests implementieren wir „per Hand“, nutzen dafür aber die Autocompletion von einem KI-Coding-Assistenten. Welcher der beiden ist in dem Fall zweitrangig, da sich diese Funktion bei Copilot und AI Assistant in der Bedienung kaum unterscheidet. Bei Features, bei denen es zwischen den zwei Produkten wesentliche Unterschiede gibt, erwähnen wir explizit Copilot oder AI Assistant.

Das folgende Beispiel eines einfachen Tests zeigt, welcher Code noch per Hand geschrieben werden muss, und welcher Code sich aus dem Kontext ergibt *(siehe Listing 1)*.

Fettgedruckter Code muss vom Entwickler noch getippt werden, der restliche Code wird automatisch vom KI-Coding-Assistent vorgeschlagen.

Wenn man einen weiteren Test schreibt, beispielsweise `generateHeroName_withEmptyList_returnsMaxMustermann`, wird man feststellen, dass tatsächlich der komplette Test durch den Assistenten korrekt vorgeschlagen wird. Der KI-Assistent nimmt den bereits geschriebenen Test als Kontext und kann daraus weitere Tests ziemlich gut ableiten. **Welche Tendenz lässt sich noch beobachten?**

Man muss weniger scrollen und nachschlagen, hat damit weniger menschlichen Kontextwechsel und kann sich so besser auf die eigentliche Arbeit fokussieren. Im konkreten Fall muss man nicht nachschauen, dass die Klassenvariable *heroService* heißt und man muss auch nicht googeln oder ausprobieren, wie noch mal die genaue Syntax für *assertThat* oder das Initialisieren einer Liste *(List.of)* funktioniert.

Varianten des Inline-Promptings

Bevor wir die eigentliche Funktionalität implementieren, wollen wir eine Methode schreiben, die überprüft, ob jeder Hero tatsächlich genau einen Vor- und einen Nachnamen hat. Dazu erweitern wir die nachfolgende, bereits bestehende Funktion *(siehe Listing 2)*.

Die folgenden drei Varianten geben einem ein Gefühl dafür, wie man „on the fly“ promptet, ohne umständlich mit einem separaten Chattool zu arbeiten:

Variante 1: Wir markieren die gesamte Funktion, öffnen den Inline-Chat und geben die folgenden Prompts ein: ***„Change this function so that it also checks whether every function consists of two words.“*** Wenn wir uns das Ergebnis anschauen, stellen wir vermutlich fest, dass leider kein regulärer Ausdruck für den Check verwendet wurde. Wir können aber interaktiv per Chat den nächsten Befehl absetzen, um uns dem Ziel iterativ zu nähern: ***„Use exactly one regex to do the check.“*** Formulieren wir den Prompt auf Deutsch, so bekommen wir auch eine Antwort auf Deutsch. Wir können keine Qualitätsunterschiede zwischen deutschen und englischen Prompts erkennen – auf der sicheren Seite ist man aber mit Englisch, da die LLMs hauptsächlich in dieser Sprache trainiert wurden.

Variante 2: Wir löschen Zeile 4, ändern den Funktionsnamen auf `checkEveryNameStartsWithACapitalLetterAndConsistsOfTwoWords` und fügen den Hilfs-Kommentar `// use regex` in Zeile 4 hinzu.

Variante 3: Wir löschen Zeile 4 und schreiben anstatt Code unseren Prompt direkt in das Sourcefile der Java-Klasse. Ein recht neues Feature der KI-Coding-Assistenten ist, dass sie natürlichsprachlichen Text automatisch erkennen und ähnlich wie einen Inline-Chat behandeln. Dieses Feature nennt sich beim AI Assistant „Inline AI Prompt“, Copilot nennt es „Inline Chat Completion Trigger“ (im Beta-Status).

Gerade bei Regulären Ausdrücken (und auch bei SQL) kann ein KI-Coding-Assistent sehr leistungsfähig sein. In allen Varianten sollte Zeile 4 dann in etwa so wie in *Listing 3* generiert werden.

Jede Variante hat ihre Daseinsberechtigung. Variante 1 ist besonders gut für iteratives Arbeiten; leider muss man gegebenenfalls im Prompt noch explizit sagen, dass die vorhandene Methode geändert werden soll (sonst wird eventuell eine zusätzliche erzeugt). Variante 2 arbeitet am engsten mit dem vorhandenen Code zusammen. Wenn man Code-Kommentare aber nur als temporäres Hilfsmittel verwendet, läuft man Gefahr diese hinterher nicht mehr zu löschen. Dieses Problem löst dann Variante 3, die sogar eine komfortable „Rückgängig“-Funktion hat.

Wir können Copilot auch unterstützen, indem wir relevante Code-Files in anderen Tabs geöffnet lassen. Denn dann werden diese Files eher als zusätzlicher Kontext berücksichtigt. Damit uns die Autocompletion die Funktion `generateHeroName(...)` im HeroService vorschlägt, müssen wir also zusätzlich den Tab *HeroServiceTest.java* geöffnet haben. Doch wir wollen diesmal einen anderen Weg gehen, um `generateHeroName(...)` zu erzeugen: die Chatfunktion.

Chat

Um von den Tests nun zu produktivem Code zu kommen, können wir den separaten Chat des KI-Assistenten verwenden. Wir können die Tests im Editor selektieren und im Chat den folgenden Prompt eingeben: „#selection create a function that passes the given test.“ Dadurch wird

die Methode „generateHeroName(...)“ erzeugt. Je nach KI-Coding-Assistent wird der generierte Code nur im Chat (AI Assistant) angezeigt oder direkt an die passende Stelle im Produktivcode hinzugefügt (Copilot).

Im Prompt haben wir #selection verwendet, um explizit auf den selektierten Code hinzuweisen. Es gibt hier noch weitere Schlüsselwörter, beispielsweise #file oder #symbol, allerdings unterstützt Copilot diese aktuell noch nicht in den JetBrains IDEs.

Mit dem Chat oder alternativ der Inline-Chatfunktion könnte man dann einen der sogenannten Commands ausführen und beispielsweise mit ***„/doc“*** die Methode dokumentieren lassen oder mit ***„/test“*** Tests generieren lassen.

Nun wollen wir unsere neu erstellte Funktion `generateHeroName(...)` über einen Controller-Endpunkt für ein Frontend verfügbar machen. Hier könnte uns der Chat durch ein Beispiel helfen: ***„Generate a simple example of a rest endpoint with java spring-boot.“*** Die Ergebnisse sind, zumindest für verbreitete Programmiersprachen und Frameworks, meist sehr gut und man spart sich damit auch hier das Blättern in einer Dokumentation.

Die Chatfunktion ist also sehr flexibel und erlaubt es, sich eigene kleine Workflows aufzubauen. So könnte man sich etwa mit dem AI Assistant den Prompt ***„#localChanges give me the list of variable names which should be improved with file name and line“*** in die Prompt Library legen, um explizit Variablennamen zu reviewen.

Arbeiten über mehrere Files hinweg

Der Copilot bietet unter VS Code die Möglichkeit, ein Feature über mehrere Dateien hinweg mit einem einzigen Prompt zu implementieren. Dafür gibt es den sogenannten „Copilot-Edits“-Chat, der speziell dafür vorgesehen ist, den vorhandenen Code zu editieren oder neuen Code in die passenden Files hinzuzufügen. Wenn wir nun beispielsweise die Funktionalität brauchen, alle Heroes zu löschen, könnten wir Controller und Service (und gegebenenfalls das Repository) in die Copilot-Edits ziehen und den Prompt ***„add a functionality which deletes all heroes“*** ausführen. Dadurch würde die Funktionalität einmal quer durch die verschiedenen Schichten hinzugefügt – sogar bis ins Frontend, wenn man die entsprechenden Dateien angibt.

Beim Arbeiten über mehrere Dateien hinweg kann man ebenso Fragen zur existierenden Codebase stellen. Wenn wir ein Feature abgeschlossen haben und testen wollen, könnte uns zum Beispiel interessieren, auf welchem Port unser Projekt läuft. Mit ***„@Workspace“*** kann man im Chat Fragen zur gesamten Codebase stellen, ohne den Kontext – also bestimmte Codestellen – explizit angeben zu müssen. So kann der Prompt ***„@workspace on which port does this project run?“*** mit konkreten Werten beantwortet werden.

Hilfe bei der Fehleranalyse

Auch bei der Beseitigung von Kompilier- und Laufzeitfehlern unterstützen uns die KI-Coding-Assistenten. Wenn man bei der Entwicklung festhängt, weil man eine komplexe Syntax nicht vollständig verstanden hat, aber genau weiß, was man umsetzen möchte, kann ein KI-Coding-Assistent oft hilfreich sein. Dies zeigt das folgende Beispiel *(siehe Listing 4)*.

Korrekt wäre es, in der letzten Zeile `composed.apply(2,3)` zu schreiben. Der aktuelle Code hingegen ergibt einen Kompilierfehler. Nutzt man den Inline-Chat mit dem Prompt ***„fix“*** beziehungsweise dem Command ***„/fix“***, wird der Code auf Anhieb passend korrigiert. Hier erkennt der KI-Assistent also, dass composed in Zeile 2 ebenfalls vom Typ BinaryOperator ist, dass wir in Zeile 3 compose aufführen wollen und weiß, dass BinaryOperator durch apply ausgeführt wird (ähnlich wie eine Function<> mit apply ausgeführt wird). Ein verwandter UseCase ist beispielsweise die Konfiguration von Mocks in Tests, bei denen man die genaue Syntax spezieller Mock-Mechanismen oft nicht immer im Kopf hat.

Bekommen wir beim Testen einen Laufzeitfehler, kann – gerade in Kombination mit dem zugehörigen Code – der KI-Coding-Assistent ebenfalls bei

Vergleichstabellen der Features

Tabelle: Inline Code Editing

	AI Assistant	Copilot in VS Code	Copilot in JetBrains IDEs
Auto Completion	☺	☺	☺
Inline Code Generation	☺	☺	☺
Inline AI Prompt	☺	😐 (beta)	-

Tabelle: Chat-Funktion

	AI Assistant	Copilot in VS Code	Copilot in JetBrains IDEs
Ask Code File	☺	☺	☺
Ask Whole Code Base	-	☺	-
Code Generation	☺	☺	☺
Multi File Edit	-	☺	-
Bugfixing	☺	☺	☺
Refactorings	☺ (Prompt Library)	😐	😐
Write Documentation	☺	☺	☺
Prompt Templates	☺	-	-
Web Context	-	☺	-
Extensions	-	☺	-

Tabelle: Zusatzfunktionen

	AI Assistant	Copilot in VS Code	Copilot in JetBrains IDEs
Renaming Suggestions	☺	☺	-
Write Commit Messages	☺	☺	😐 (nur manuell über Chat)
Pull Request Summary & Review	-	☺	-
Explain Commits	☺	-	-
Auto Programming Language Conversion	☺	-	-
Voice Input	-	☺	-

Tabelle: Compliance

	AI Assistant	Copilot in VS Code	Copilot in JetBrains IDEs
Renaming Suggestions	☺ Nein (sofern kein Opt-In)	☺ Nein (ab Business)	☺ Nein (ab Business)
Write Commit Messages	☺	-	-
Pull Request Summary & Review	-	☺	-
Explain Commits	☺ (nur Enterprise und Java, Kotlin, Python)	☺	☺
Auto Programming Language Conversion	? (keine genauen In formationen verfügbar)	☺	☺
Voice Input	-	☺	☺

der Problemanalyse helfen. Der Kontext kommt dann in Form von Code und Fehlermeldung. Dies ist insbesondere dann hilfreich, wenn man neu in einer (weitverbreiteten) Library oder in einem Framework ist, da der KI-Assistent diese wahrscheinlich gut kennt und die Fehlermeldung einordnen kann. Der AI Assistant bietet dafür sogar einen expliziten Button in den Logs an.

Über die Autoren

Bastian Weinlich

Bastian Weinlich programmiert seit über 15 Jahren mit Leidenschaft und das seit einiger Zeit am liebsten mit Unterstützung von KI-Coding-Assistenten. Er interessiert sich insbesondere für effiziente Software-Entwicklung und die Integration von KI-Features in Business-Software.

Semjon Mössinger

Semjon Mössinger ist noch ziemlich neu im Speaker-Business, hat aber bereits ein Meetup zu AI-Coding-Tools abgehalten, Softwarearchitektur-Trainingskurse für Kunden gegeben und firmeninterne Schulungen zu verschiedenen Themen durchgeführt.

Quellen

[1] *https://resources.github.com/copilot-for-business/*
[2] *https://blog.jetbrains.com/blog/2024/10/22/introducing-mellum-jetbrains-new-llm-built-for-developers/*
[3] *https://martinfowler.com/bliki/TwoHardThings.html*
[4] *https://youtrack.jetbrains.com/articles/SUPPORT-A-689/AI-Assistant-how-to-filter-public-code-suggestions-for-code-completion*
[5] *https://blog.jetbrains.com/blog/2024/10/22/introducing-mellum-jetbrains-new-llm-built-for-developers/*
[6] *https://blogs.microsoft.com/on-the-issues/2023/09/07/copilot-copyright-commitment-ai-legal-concerns/*

Bastian Weinlich
Bastian.Weinlich@wps.de

Semjon Mössinger
Semjon.Moessinger@wps.de

Hochverfügbar, aber bitte etwas günstiger

Markus Flechtner, Ordix

Wenn wir an Hochverfügbarkeit für die Oracle-Datenbank denken, kommen uns RAC und DataGuard in den Sinn. Und natürlich die hohen Kosten dafür. Und wenn wir an Hochverfügbarkeit für die Standard Edition 2 (SE2) der Oracle-Datenbank denken, denken wir an … – Features wie SEHA oder Refreshable PDB, die helfen, die Oracle-Datenbank 19c SE2 verfügbarer zu machen.

Normalerweise denkt man bei Hochverfügbarkeit für die Oracle-Datenbank an die große Enterprise Edition, dazu die Real Application Cluster Option (RAC) und die Active-Data-Guard-Option – und gerne noch Golden Gate. Ebenso denkt man an die dazugehörigen Preisschilder. Man ist geneigt, zu denken, dass Oracle Maximum Availability Architecture (MAA)-Blueprints gezielt rund um diese teuren Premium-Produkte herum gestaltet wurden.

Für den kleinen Bruder der Enterprise Edition, die Standard Edition 2 – die für sehr viele Anwendungsfälle durchaus ausreichend ist – ist das Angebot spärlicher gesät. Als DataGuard-Ersatz gibt es diverse Produkte von Drittanbietern. Und bis Oracle 18c gab es dann noch den Real Application Cluster, der dann kurzfristig für Oracle 19c abgekündigt wurde. Es dauerte etwas länger als ein Jahr, bis Oracle mit SEHA, kurz für „Standard Edition High Availability", eine clusterbasierte Hochverfügbarkeitslösung für die SE2 auf den Markt brachte. Und die Container-Datenbank-Architektur bietet mit der „Refreshable PDB" ein Feature, das man auch für Hochverfügbarkeitszwecke nutzen kann.

Was ist Standard Edition High Availability (SEHA)?

SEHA ist eine automatisierte Failover-Datenbank-Lösung für die Oracle-Datenbank SE2 basierend auf Oracle Grid Infrastructure *(siehe Abbildung 1)*. SEHA ist seit dem Release Update 19.7 für Linux x86-64, Solaris SPARC, MS Windows verfügbar und seit RU 19.13 für IBM AIX und HP-UX Itanium. Bei der Installation ist zu beachten, dass auch die OCW im RDBMS-Home gepatcht werden muss, denn wenn man nur den RDBMS-RU im Datenbank-Home installiert, dann bleibt OCW im Stand 19.3 – und dann funktioniert SEHA nicht.

Unter Windows kann SEHA auch als Ersatz beziehungsweise für Failsafe (wenn mit SE2 genutzt) gesehen werden, denn Failsafe ist seit Oracle Database 21c desupported.

Im Gegensatz zur RAC-Lösung für die Oracle-Datenbank SE2, die bis Oracle 18c verfügbar war, stehen einer Datenbank-Instanz bei SEHA lizenzmäßig 2 CPU-Sockets mit insgesamt 16 Threads zur Verfügung. Beim 18c-RAC mussten diese Ressourcen auf beide RAC-Knoten aufgeteilt werden. Lizenzmäßig kommt weiterhin die bekannte 10-Tages-Regel zum Einsatz, nach der in solchen Failover-Umgebungen der 2. Knoten nicht lizenziert werden muss, wenn er nicht mehr als 10 x 24 Stunden pro Jahr aktiv genutzt wird (Einzelheiten *siehe [1]*).

Wie richtet man SEHA ein?

Der Database Configuration Assistant (DBCA) unterstützt die Einrichtung von SEHA noch nicht. Das heißt, dass wir zuerst auf einem unserer Cluster-Knoten eine Single-Instance-Datenbank anlegen müssen. Anschließend sind noch einige kleine Nacharbeiten erforderlich. In einem ersten Schritt müssen wir die Password-Datei der Datenbank vom Dateisystem ins ASM verlegen, damit beide Knoten auf diese Datei zugreifen können *(siehe Listing 1)*.

```
$ asmcmd pwcopy /u00/app/oracle/dbs/orapwSEHADB +DATA/SEHADB/orap-
wSEHADB
$ srvctl modify database -db SEHADB -pwfile +DATA/SEHADB/orapwSEHADB
```

Listing 1: Verschieben der Password-Datei ins ASM

Anschließend müssen wir noch festlegen, auf welchen Cluster-Knoten unsere SEHA-Datenbank laufen darf *(siehe Listing 2)*.

```
srvctl modify database -db SEHADB -node node1,node2
```

Listing 2: Festlegung der Knoten für die SEHA-Datenbank

Und damit sind wir fertig. Die notwendigen Verzeichnisse (ADR und so weiter) auf dem 2. Knoten werden angelegt, sobald die Instanz das erste Mal auf dem Knoten gestartet wird.

Wie funktionieren „Switchover" und Failover bei SEHA?

Mit dem Befehl „srvctl relocate" kann die Datenbank von einem Knoten auf einen anderen Knoten verschoben werden *(siehe Listing 3)*.

```
$ srvctl status database -db SEHADB
Instance SEHADB is running on node node1

$ srvctl relocate database -db SEHADB -node node2

$ srvctl status database -db SEHADB
Instance SEHADB is running on node node2
```

Listing 3: Verschieben einer SEHA-Datenbank

Dabei wird die Datenbank-Instanz auf node1 gestoppt („Shutdown immediate") und dann auf node2 gestartet. SEHA ist nicht RAC One Node, bei dem während des Failovers zwei Instanzen aktiv sind. Es gibt immer eine Auszeit für die Applikationen. Die Dauer dieser Auszeit hängt davon ab, wie viele offene Trans-

```
SEHA_APP =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = sehacluster)(PORT=1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = SEHA_APP_SVC)    )
  )
```

Listing 4: Beispielhafter tnsnames.ora-Eintrag für eine SEHA-Datenbank

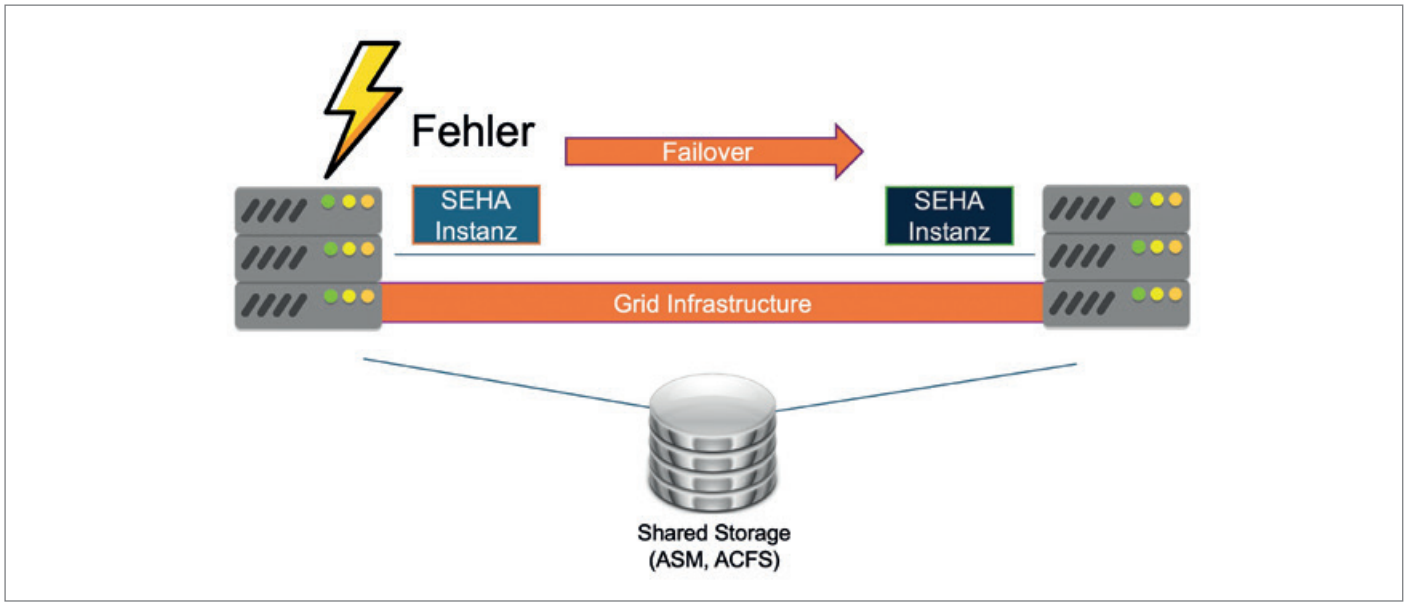


Abbildung 1: SEHA im Überblick (Quelle: Markus Flechtner)

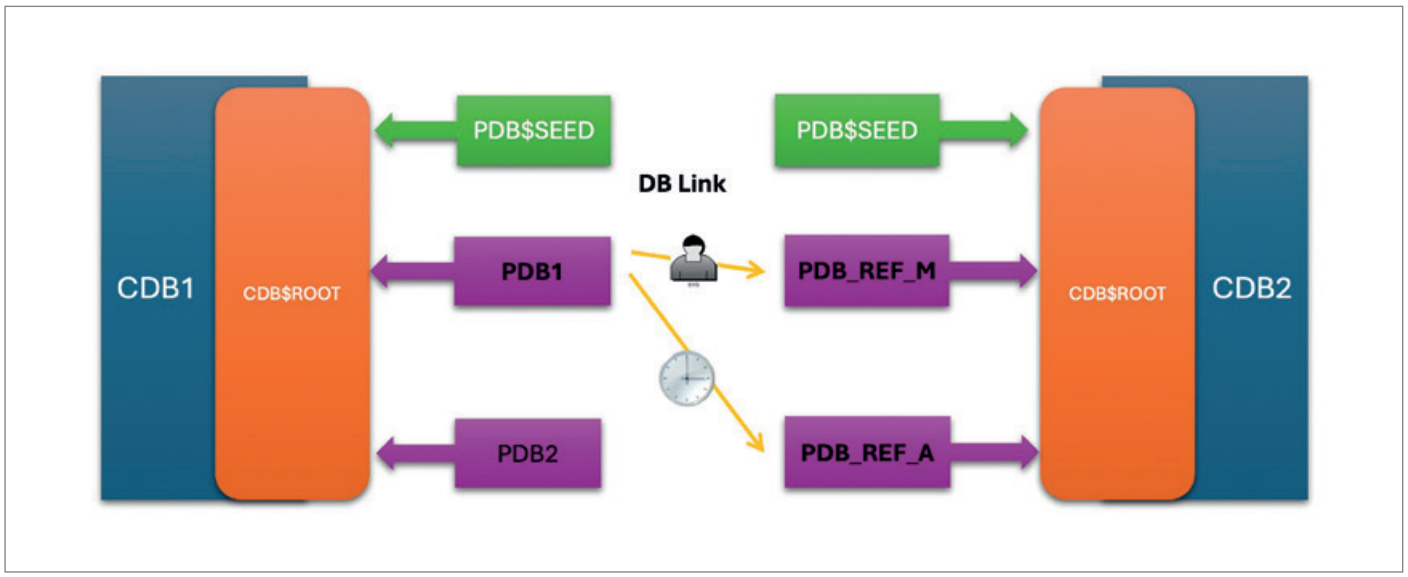


Abbildung 2: Refreshable PDB im Überblick (Quelle: Markus Flechtner)

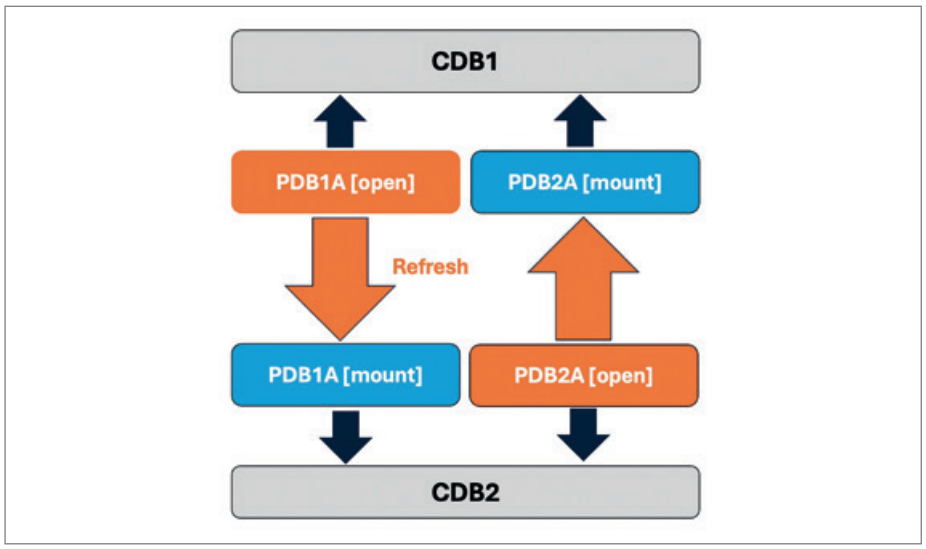


Abbildung 3: Gegenseitige Absicherung zweier SE2-CDBs (Quelle: Markus Flechtner)

aktionen beim Herunterfahren der Instanz zurückgerollt werden müssen.

Ein Failover ist ganz einfach: die Oracle Clusterware stellt fest, dass der Knoten ausgefallen ist und fährt die DB-Instanz automatisch auf einem anderen Knoten hoch.

Wie melden sich die Anwendungen bei einer SEHA-Datenbank an?

Anwendungen nutzen einen Applikations-Service und sprechen den Cluster über den SCAN-Namen an *(siehe Listing 4)*.

```
SQL> CREATE USER C##REFRESH identified by <PW>;
SQL> GRANT CREATE SESSION,RESTRICTED SESSION,SYSOPER
  2  to C##REFRESH CONTAINER=ALL;
SQL> GRANT CREATE PLUGGABLE DATABASE TO C##REFRESH;
```

Listing 5: Anlegen des Benutzers für den Refresh

```
SQL> CREATE PUBLIC DATABASE LINK <SOURCE_CDB>
   2 CONNECT TO C##REFRESH IDENTIFIED BY '<PW>' USING '<TNS>';
```

Listing 6: Anlegen des Datenbank-Links

```
SQL> CREATE PLUGGABLE DATABASE <PDB> FROM <PDB>@<SOURCE_CDB>
  2  file_name_convert= … REFRESH MODE EVERY 240 MINUTES;
```

Listing 7: Anlegen der Refreshable PDB

```
SE2PDB(4):alter pluggable database refresh
2020-08-16T11:58:12.076936+02:00
Applying media recovery for pdb-4099 from SCN 1672311 to SCN 1672347
Remote log information: count-1
thr-1,seq-20,logfile-/u02/fast_recovery_area/SE2CDBA/foreign_archiv-
elog/SE2PDB/2020_08_16/o1_mf_1_20_hml0smw3_.arc,los-1592646,nxs-
18446744073709551615,maxblks-366859
SE2PDB(4):Media Recovery Start
2020-08-16T11:58:12.077338+02:00
SE2PDB(4):Serial Media Recovery started
SE2PDB(4):max_pdb is 5
2020-08-16T11:58:12.154566+02:00
SE2PDB(4):Media Recovery Log /u02/fast_recovery_area/SE2CDBA/for-
eign_archivelog/SE2PDB/2020_08_16/o1_mf_1_20_hml0smw3_.arc
2020-08-16T11:58:12.665203+02:00
SE2PDB(4):Incomplete Recovery applied until change 1672347 time
08/16/2020 11:58:10
2020-08-16T11:58:12.668710+02:00
SE2PDB(4):Media Recovery Complete (SE2CDBB)
SE2PDB(4):Completed: alter pluggable database refresh
```

Listing 8: PDB-Refresh im alert.log der Ziel-Datenbank

```
SQL> select systimestamp - scn_to_timestamp(a.last_refresh_scn) DELTA
  2  from dual@SE2CDB,
  3  (select LAST_REFRESH_SCN from cdb_pdbs where pdb_name='SE2PDB')
a;

DELTA
---------------------------------------------------------------
+000000000 00:00:30.585794000
```

Listing 9: Monitoring der Refreshable PDB

Für die Anwendungen stehen dabei die Hochverfügbarkeitsoptionen Transparent Application Failover (TAF), Fast Application Notification (FAN) und Fast Connection Failover (FCF) zur Verfügung, damit die Anwendungen auf einen Server-Wechsel der Datenbank-Instanz reagieren können. Transparent Application Contuinity (TAC), bei der auch DML-Befehle automatisch nach einem Serverwechsel der Instanz wiederholt werden, steht leider nicht zur Verfügung, denn TAC erfordert eine Lizenz für RAC oder Active DataGuard.

Insgesamt haben wir mit SEHA aber eine einfach einzurichtende Cluster-Lösung für die Oracle-Datenbank SE2, die gegen den Ausfall eines Servers schützt.

Was ist „Refreshable PDB"?

Im Gegensatz zu SEHA, das mit Non-CDBs und CDBs implementiert werden kann, erfordert „Refreshable PDB" die Container-Datenbank-Architektur *(siehe Abbildung 2)*. Genauer gesagt sind zwei Container-Datenbanken erforderlich: eine, auf der eine PDB im Read-Write-Modus läuft und von den Anwendungen genutzt wird, und eine zweite, auf der eine Kopie der PDB läuft, die regelmäßig mit Hilfe von Redo-Informationen aktualisiert wird.

Die Aktualisierung der Refreshable PDB kann dabei entweder manuell durch den Administrator oder automatisch in regelmäßigen Abständen erfolgen. Im letzteren Fall wird ein Scheduler-Job angelegt, der diese Arbeit dann erledigt.

Die Refreshable PDB darf dabei maximal Read-Only geöffnet werden. Damit ergeben sich folgende Anwendungsmöglichkeiten für die Refreshable PDB:

- Reporting (Refreshable PDB „read-only" öffnen)
- Refreshable PDB als Quelle für weitere Kopien
- Migration einer Non-CDB in einer PDB (denn als Quelle kann auch eine Non-CDB genutzt werden)
- Hochverfügbarkeit für SE2 („Data Guard Lite")

Wir konzentrieren uns auf den letzten Anwendungsfall.

```
PDB_APP.MARKUSDBA =
  (DESCRIPTION =
    (RETRY_COUNT=40)(RETRY_DELAY=3)
    (ADDRESS_LIST=
    (LOAD_BALANCE = OFF )
    (ADDRESS = (PROTOCOL = TCP)(HOST = node1)(PORT=1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = node2)(PORT=1521))
    )
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = PDB_APP_SVC)
    )
  )
```

Listing 10: tnsnames.ora-Eintrag für HA mittels „Refreshable PDB“

```
SQL> REM Ggf. eine Dummy-PDB anlegen und oeffnen
SQL> create pluggable database dummypdb admin user pdbadmin
  2  identified by manager roles=(DBA)
  3  file_name_convert=('pdbseed','DUMMYPDB');
Pluggable database created.
SQL> alter pluggable database dummypdb open;
Pluggable database altered.

SQL> REM Datenbank-Link droppen und neu anlegen
SQL> DROP PUBLIC DATABASE LINK SE2CDB_REFRESH;
Database link dropped.
SQL> CREATE PUBLIC DATABASE LINK SE2CDB_REFRESH
  2  CONNECT TO C##REFRESH IDENTIFIED BY manager USING
  3  '<server>:<port>/dummypdb';
Database link created.

SQL> REM  Refresh-Mode der PDB ändern und PDB öffnen
SQL>  alter pluggable database SE2PDB refresh mode none;
Pluggable database altered.

SQL> alter pluggable database SE2PDB open;
Pluggable database altered.
```

Listing 11: Workaround zum Öffnen der PDB im Failover-Fall

Wie wird eine Refreshable PDB angelegt?

Voraussetzung ist, dass die Quell-CDB mit Local Undo konfiguriert ist und im Archivelog-Modus läuft. Wir legen dann in der Quell-CDB einen Common-Benutzer an, den wir für die Refreshable PDB nutzen *(siehe Listing 5)*.

Die Verbindung zwischen Quell-CDB und Ziel-CDB erfolgt via Datenbank-Link, den wir im nächsten Schritt anlegen *(siehe Listing 6)*.

Der Datenbank-Link kann dabei entweder in die CDB$ROOT der Quell-CDB oder in die PDB selbst gehen. Danach können wir die Refreshable PDB anlegen *(siehe Listing 7)*.

Folgende Refresh-Modi stehen dabei zur Verfügung:

- REFRESH MODE EVERY <n> MINUTES (das kürzeste Intervall ist 1 Minute)
- REFRESH MODE MANUAL
- REFRESH NONE (damit wird die Refreshable PDB zur „normalen PDB“ und kann im Read-Write-Modus geöffnet werden)

Für den Refresh ist Folgendes zu beachten *(siehe Listing 8)*:

- Die Refreshable PDB muss geschlossen sein („MOUNT“-Status). Wenn die PDB Read-Only geöffnet ist, dann ist kein Refresh möglich.
- Beim Refresh muss die Quell-CDB erreichbar sein.
- Bei größeren Refresh-Intervallen muss sichergestellt sein, dass alle Redolog-Informationen verfügbar sind. Gegebenenfalls muss man die Archive-Log-Deletion-Policy im RMAN anpassen oder über REMOTE_RECOVERY_FILE_DEST die Archivelogs für die Ziel-Datenbank verfügbar machen.

Wie man an dem Auszug aus dem alert.log der Ziel-Datenbank sieht, erfolgt auf der Ziel-Datenbank ein Point-In-Time-Recovery der Refreshable PDB.

Wir konzentrieren uns im Folgenden auf den automatischen Refresh mit dem kleinstmöglichen Intervall von einer Minute und einer Refreshable PDB im „Mount“-Status, denn das wird unser Hochverfügbarkeitsszenario.

„Refreshable PDB“ als „Data Guard Lite“

Wenn wir die Refreshable PDB mit einem Intervall von einer Minute laufen lassen, dann haben wir beim Ausfall der „Primär-Datenbank“ maximal eine Minute Datenverlust. Und das kann für viele Anwendungsfälle reichen. Da aber die „Standby-Datenbank“ im Gegensatz zur SEHA auch lizenziert sein muss, bietet es sich an, dass sich zwei Datenbanken gegenseitig absichern *(siehe Abbildung 3)*.

Für das Monitoring einer „Refreshable-PDB-HA-Lösung“ können wir eine einfache SQL-Abfrage auf der Standby-CDB nutzen, die die zeitliche Differenz zwischen den beiden Datenbanken ermittelt *(siehe Listing 9)*.

Vorbereitung der Anwendungen

In der tnsnames.ora der Clients müssen beide Server eingetragen werden, damit sich die Anwendungen auch nach einem Failover mit der Datenbank verbinden können *(siehe Listing 10)*.

Failover einer „Refreshable PDB“

Die Aktivierung einer Refreshable PDB

im Fehlerfall ist leider nicht ganz einfach, denn der Befehl „ALTER PLUGGABLE DATABASE .. REFRESH MODE NONE;“ funktioniert in diesem Fall nicht mehr. Er braucht zwingend eine Verbindung zur Quell-Datenbank, die nicht mehr erreichbar ist. Frank Gerasch hat aber einen cleveren Workaround für dieses Problem gefunden *[4]*: der Datenbank-Link wird einfach zu einer aktiven PDB „umgebogen“. Dann kann der Refresh deaktiviert werden und die PDB kann im Read-Write-Modus geöffnet werden *(siehe Listing 11)*.

Diese Hochverfügbarkeitslösung hat allerdings einige Einschränkungen:

- Ein „Switchover“ ist nicht möglich (Ausnahme „Refreshable-PDB Switchover“ bei der Exadata oder in der OCI).
- Es gibt keinen Observer, das heißt, es ist kein automatischer Failover möglich. Das lässt sich aber mit einem entsprechenden Skript lösen.
- Nach einem Failover muss die Standby-PDB neu aufgesetzt werden („kein REINSTATE“).
- Wenn die ehemalige „Primary CDB“ wieder verfügbar ist, ist eine Split-Brain-Situation möglich.
- Der Failover ist nicht ganz einfach.
- Und wir haben einen Datenverlust von maximal einer Minute.

Zusammenfassung

„Standard Edition High Availability“ ist eine einfach einzurichtende Failover-Datenbank-Lösung für die Oracle-Datenbank SE2 auf Basis der Oracle Grid Infrastructure. „Refreshable PDB“ als „DataGuard Lite“ bietet eine Hochverfügbarkeitslösung mit theoretisch maximal einer Minute Datenverlust, einer leider etwas komplexeren Failover-Prozedur und einigen weiteren Einschränkungen. Insgesamt kann man mit beiden Methoden die Verfügbarkeit einer SE2-Datenbank deutlich erhöhen; es ist zwar keine „Zero-Data-Loss-Lösung“ wie bei der Enterprise Edition der Datenbank, aber es kann für viele Kunden ausreichen.

Quellen

[1] Licensing Data Recovery Environments: *http://www.oracle.com/us/corporate/pricing/data-recovery-licensing-070587.pdf*

[2] Oracle Dokumentation: Oracle Database 19c - Database Installation Guide - Chapter 10 „Installing Standard Edition High Availability“

[3] Oracle Dokumentation: Oracle Database 19c – Multitenant Administrators Guide – Section 7.5 „About Refreshable Clone PDBs“

[4] Frank Gerasch: Refreshable Clone PDB als Standby-DB: *https://frank-gerasch.de/2022/07/oracle-refreshable-pdb-clone-als-standby-db*

Über den Autor

Markus Flechtner ist Principal Consultant bei der ORDIX AG. Seine Schwerpunkte liegen in den Bereichen Multitenant, Hochverfügbarkeit sowie Upgrades und Migrationen. Weiterhin gibt er Seminare zu Oracle und PostgreSQL. Als Mitbegründer von ora2know engagiert er sich für den Know-how-Austausch in der Oracle-Datenbank-Community. Seit 2020 ist er Oracle ACE.

Markus Flechtnerr
mfl@ordix.de

Oracle Datenbanken Monthly News

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

DOAG Online

Es ist wieder so weit: die neue Ausgabe ist online! Das sechsköpfige Redaktionsteam von Oracle Deutschland hat wieder Neuigkeiten rund um die Oracle-Datenbank für On-Premises und Cloud-Installation zusammengestellt.

Alles wird wieder in einem Video präsentiert.

In der aktuellen Ausgabe wird wieder ein zusätzliches Quick Link Posting (in Englisch) zur Verfügung gestellt, um einen schnellen Zugriff auf die zugehörigen Beiträge zu gewährleisten.

https://www.doag.org/de/home/news/oracle-datenbanken-monthly-news-42/

„Wiederverwertung von gut getestetem Code wo immer möglich, anstatt grüne Wiese“

Günther Stürner sprach im Juli 2024 mit Andreas Gaede, einem der Gründer und heute alleinigem CEO/GF der PITSS GmbH, die ihren Hauptsitz in Stuttgart hat. Weitere Standorte sind Paderborn und Troy, USA. PITSS ist eines der führenden Software-Häuser, die sich auf die Analyse und Modernisierung von Oracle Forms und Oracle-Reports-Systemen spezialisiert hat. Ihre Produkte und ihr Service werden weltweit angeboten.

Herr Gaede, mit Ihrem Software-System PITSS.CON sind sie ein globaler Player im Bereich der Analyse und Migration von Oracle Forms und Oracle-Reports-Systemen. Sind sie ein typischer schwäbischer IT-Mittelständler?

Wenn man als typischen schwäbischen Mittelständler eine Firma meint, die hoch-innovativ und verlässlich ist, tolle Produkte, einen erstklassigen Service bietet und auch schwierige Projekte in der vorgegebenen Zeit und im Kostenrahmen abschließt, dann wäre es mir eine Ehre als solcher dargestellt zu werden.

Aber ja, wir sind mit unserer Software weltweit unterwegs und machen Projekte überall dort, wo Kunden Oracle-Forms-Anwendungen einsetzen, die entweder qualitativ verbessert werden sollen oder, was aktuell immer mehr nachgefragt wird, wo Kunden ihre in die Jahre gekommenen Oracle-Forms-Anwendungen nach Oracle APEX überführen wollen oder müssen.

Sie entwickeln Software, um andere Software zu analysieren. Was hat Sie dazu bewogen, ein solches Spezialgebiet anzugehen.

Die PITSS gründete sich 1999 als Dienstleister in Bereichen der ERP-Entwicklung und Implementierung, wie zum Beispiel der Oracle E-Business Suite EBS, und als versierter Oracle-Spezialist. Ziel war es, stets professionelle IT-Lösungen in Form von Software-Produkten und Services anzubieten, kurz PITSS.

Es war auch für viele Oracle-Forms-Kunden an der Zeit, ihre serverseitigen, Character-basierenden Anwendungen in Richtung Client-Technologie mit Forms 4.5, 5 oder gar 6 zu modernisieren. Aus diesem Grund wandten sich zahlreiche Unternehmen hilfesuchend an uns, um ihre gewachsenen Geschäftsanwendungen auf das neueste Forms Release zu heben.

Brauchte man für ein Upgrade auf ein neues Release tatsächlich Unterstützung? War das so kompliziert?

Oh ja, das war sehr kompliziert. Es war ja nicht nur ein Software-Upgrade, das man einspielte und dann war alles erledigt. Der Schritt von Character-orientiert zu GU-orientiert und später von Client-Server zu einer three-tier-Architektur waren komplexe Herausforderungen, die manchen CIO und seine Teams an ihre Grenzen brachten.

Der Fertigungsingenieur und leidenschaftliche Informatiker in mir sträubte sich jedoch, diese Aufgabe für jeden einzelnen Kunden in Handarbeit durch viele IT-Fachkräfte mühsam und doch fehleranfällig umsetzen zu lassen. So entstand schon früh unsere Mission, intelligente Produkte wie PITSS.CON zu entwickeln, die enorme Datenmengen in Form von Programm-Code, Prozeduren, Funktionen bis hin zu Datenbank-Tabellen, nicht nur einer, sondern vieler unterschiedlicher, branchenspezifischer Anwendungen als Metadaten laden und verstehen können.

Es war also die Idee, mit Software andere Software zu analysieren?

Ja, genau. Bereits mit unserem ersten Produkt, das wir 2001 fertiggestellt hatten, stellten wir vor nahezu jede Handlung eine umfängliche, maschinengestützte Analyse. Analyse war das Zauberwort. Das Durchleuchten und Verstehen der Systeme war und ist ein wichtiges Grundprinzip unserer Herangehensweise.

Die analytisch aufbereiteten Daten ließen sich nach Belieben verändern, kostenbewusst warten und auf neueste Versionen anheben. Damit stellten wir uns ganz bewusst gegen das übliche, lukrative IT- Dienstleistungsmodell, das möglichst viele Ressourcen möglichst lange bei den Kunden unterbringen möchte.

Sicher ein Grund, warum wir in über 40 Ländern erfolgreich unsere Produkte installieren und Projekte erfolgreich abschließen konnten.

Wie muss man sich eine solche Analyse einer Oracle-Forms- oder Oracle-Reports-Anwendung vorstellen? Wie gehen Sie oder Ihre Kunden dabei vor?

Im Prinzip ist die Abfolge relativ einfach. Ein Kunde liefert uns im Minimum das Forms-.fmb-File, das quasi den Source-Code einer Forms-Anwendung darstellt. Im besten Fall liefert er uns seine ganze Anwendung, nämlich alle Programmquellen, sprich *.fmb für Forms, *.pll und *.olb als Libraries, *.rdf für Reports, *.sql für all begleitenden SQL-Programme bis hin zu ASCII-Dateien wie Pro*C und einen Struktur-Dump der Datenbank. Wichtig ist, dass man keine Daten liefert. Diese Files werden durch unseren PITSS.CON-Parser in ihre Bestandteile zerlegt und diese in unserem PITSS.CON-Data-Cube, innerhalb einer Oracle-Datenbank, abgelegt. Es ist immer noch dieselbe Anwendung. Wir haben aber den ‚Aggregatszustand' verändert. Die gesamte Forms-Anwendung liegt nun in einem Zustand vor, der eine Auswertung nach allen Regeln der SQL-Kunst zulässt. Wohlgemerkt, wir brauchen keine Kundendaten für diese Übung und wir verändern in dieser Phase die Anwendung in keiner Weise.

Was hat der Kunde davon, wenn er seine Oracle-Forms-Anwendung in der Art aufgeschlüsselt zur Verfügung hat?

Wie erwähnt, hat der Kunde die Möglichkeit, seine Anwendung nach allen Regeln der Kunst auszuwerten und zu begutachten. Wir liefern eine Vielzahl vordefinierter Analysen und Auswertungen. Er bekommt eine ausführliche Dokumentation seiner Anwendung und eine Liste der Schwachpunkte, die in einem nächsten Schritt teils mit einem hohen Automatisierungsgrad eliminiert werden können. Das Ergebnis wäre dann eine runderneuerte, verbesserte, verschlankte Version der bisherigen Forms- oder Reports-Anwendung.

Das ist in der Tat bei großen Systemen ein unbezahlbarer Vorteil. Aus einer Black-Box wird ein gläserner, transparenter Würfel?

Ja, diese Informationen sind ohne den Einsatz von PITSS.CON nur sehr schwer und meist nicht vollständig zu erhalten. Einmal zerlegt und in den Data Cube geladen, kann das Forms-System aus unterschiedlichen Blickwinkeln begutachtet werden. Das macht aus unserer Sicht – dauerhaften Einsatz vorausgesetzt – unser Produkt zu einem Qualitätssteigerungstool. Man kann sogar so weit gehen, dass es keine Änderungen mehr geben sollte, ohne dass diese Änderungen mit Hilfe von PITSS.CON im Vorfeld

qualitätsgeprüft sind. Selbst nach erfolgten Änderungen kann PITSS.CON zur optimalen Testunterstützung herangezogen werden. In diesem Fall lassen sich die Änderungen im Ablauf so aufzeigen, dass ein gezieltes und zeitsparendes Testen der Anwendung möglich wird. Aber das ist natürlich Sache der Kunden, wie sie im praktischen Einsatz damit umgehen.

Was meinen Sie mit qualitätsgeprüft?

Zum Beispiel soll geklärt werden, wo sich eine Änderung in einer Anwendung oder auch in der Datenstruktur überall auswirkt. Das führt in einem größeren System sehr schnell zu vielen Stellen, die entsprechend angepasst werden müssen. Diese Art der Information bereitzustellen ist innerhalb von PITSS.CON eine triviale Aufgabe. Ohne eine solche Hilfestellung ist es zumindest eine Herausforderung. Dadurch lassen sich viele Fehlerquellen ausschließen und die Qualität des Systems wird maßgeblich verbessert.

Ich habe Sie unterbrochen, wir waren eben noch bei dem transparenten Würfel…

Ja, das eben gesagte ist nur die eine Seite der Medaille. Ist eine Forms- oder Reports-Anwendung einmal in ihre Bestandteile zerlegt, analysiert und eventuell auch modifiziert, kann man – ein entsprechendes Regelwerk vorausgesetzt – aus diesen Daten wieder ein Software-System generieren. Das Zerlegen einer Forms-Anwendung ist also keine Einbahnstraße. Eine Rückgenerierung, nach einer Bereinigung oder Renovierung, in eine Forms-Anwendung ist genauso gut möglich, wie eine Generierung hin zu einer anderen Plattform.

Das ist genau die Idee, die wir bei der Migration von Oracle Forms nach Oracle APEX oder von Oracle Reports nach Jasper verfolgen.

Darf ich das mit meinen Worten zusammenfassen: Sie nehmen eine Oracle-Forms-Anwendung, lassen sie von ihrer Software zerlegen, analysieren, machen einige Verbesserungen, drücken auf den Knopf und erhalten eine Oracle-APEX-Anwendung?

Das mit ‚auf den Knopf drücken' stimmt, wenn man aus einer bereinigten Forms- beziehungsweise Reports-Datenbasis wieder ein Forms- oder Report-Programm generieren lassen will.

Bei einer Migration hin zu APEX oder Jasper ist das nicht ganz so einfach. Wir sind zwar sehr gut bei der Umsetzung und wir werden von Version zu Version immer besser, aber eine vollständige Umsetzung einer Forms-Anwendung in Richtung APEX ist nicht möglich und auch nicht immer sinnvoll, um einen nativen beziehungsweise natürlichen APEX-Code zu erzeugen, damit im Nachgang eine optimale Wartbarkeit gegeben ist. So ist es vor allem im Front-End-Bereich, denn hier ist bei APEX die Anzahl der Möglichkeiten um ein Vielfaches größer als bei Oracle Forms. In diesem Bereich ist ein manueller Eingriff nötig und von uns, wie meist auch vom Kunden, gewollt.

Migrations-Projekte, dies gilt besonders bei großen Systemen, sind trotz guter Software-Unterstützung, ohne ein stringentes Vorgehensmodell und erstklassige Projektsteuerung nicht erfolgreich zu machen. Das ist unsere zweite Kernkompetenz: Projektsteuerung und Projektführung. Hoher Automatisierungsgrad durch unser PITSS.CON-Toolset und gute Projektführung sind der Schlüssel für erfolgreiche Projekte. Unsere Kunden schätzen vor allem, dass unsere berechneten Projektlaufzeiten sehr exakt sind. Dies ist auch der Grund, weshalb wir solche Projekte als Festpreisprojekte anbieten und durchführen können.

Was sind die Gründe, die Firmen mit einem funktionierenden System veranlassen, sei es Forms oder Reports, an eine Migration zu denken? Solche Projekte sind trotz massiver Software-Unterstützung immer Eingriffe in Abläufe eines Unternehmens. Das macht man ja nicht einfach so aus einer Laune heraus.

Oracle Forms ist ein großartiges Produkt, das allerdings in die Jahre gekommen ist. In Ehren ergraut, kann man sagen. So wie auch der Großteil der Forms-Entwickler und Entwicklerinnen. Im Klartext heißt das, dass viele neue Technologien, insbesondere im Front-End-Bereich, nicht oder nur rudimentär eingesetzt werden können. Optisch wirken Forms-Applikationen oft ein wenig ‚von gestern'. Auch was den Einsatz der Anwendungen auf unterschiedlichen Devices angeht, tut man sich eher schwer. Eine Forms-App auf einem Smartphone oder auf einem iPad suchen sie vergebens.

Oracle APEX als neue Zielplattform bei einer Migration spielt in einer völlig anderen Liga. Das Entwicklungskonzept ist zwar ähnlich wie bei Oracle Forms, was eine Migration vereinfacht. APEX ist jedoch auf dem aktuellen Stand der Technik, was den Front-End Bereich angeht, und das Ergebnis sind modernste Systeme, die auf allen Devices ablauffähig sind.

Der zweite Grund, weshalb sich Forms-Kunden mit Migrationsüberlegungen beschäftigen, sind die rückläufigen Zahlen von Forms-Experten und -Expertinnen. Bei manchen Firmen geht die Furcht um, eines Tages ohne Entwicklungsunterstützung dazustehen. Auch hier finden wir in der APEX-Welt genau die andere Tendenz. Die APEX-Entwickler-Community ist riesig und wächst immer noch rasant. Unter diesen Umständen ist APEX, aus unserer Sicht, die ideale zukünftige Plattform für Oracle-Forms-Anwendungen. Es ist auch innerhalb von Oracle keine Frage: APEX gehört eindeutig die Zukunft.

Forms zu APEX-Migrationen sind also keine ‚Knopf-Druck'-Projekte. Ein Forms-Kunde könnte auch ein völlig neues Projekt starten, um von den aktuellen Software-Trends zu profitieren. Wäre das nicht der bessere und innovativere Weg?

Eine vollständige Neuentwicklung eines Anwendungspaketes ist natürlich immer eine Option. Sie bietet sich immer dann an, wenn das bestehende System, die bestehenden Prozesse und Abläufe diametral zu dem aktuellen Forms-System sein müssen. Aus welchen Gründen auch immer.

In den meisten Fällen, die wir kennen, ist dies jedoch nicht der Fall. Hier ist der minimal-invasive Eingriff einer Migration der bessere Weg. Was noch was taugt, wird weiter benutzt, anstatt neu erfunden. Hier sind insbesondere das Datenmodell und die unzähligen PL/SQL-Prozeduren und -Funktionen zu nennen, die meist übernommen werden können. Wiederverwertung von gut

getestetem Code wo immer möglich, anstatt grüne Wiese. Ein weiterer interessanter Punkt für unsere Kunden ist die Möglichkeit der sukzessiven Migration im laufenden Betrieb. Einzelne, logisch beziehungsweise im Business-Kontext zusammenhängende Module, wie zum Beispiel Lagerprogramme, werden migriert, den End-Benutzern zur Verfügung gestellt und zügig in Betrieb genommen, während andere Module noch mit der Forms-Technologie so lange weiterlaufen, bis sie für die Umstellung an der Reihe sind. Damit werden auch zeitaufwendige und kostenintensive Prototypen obsolet, die nie zum Einsatz kommen, und ersetzt durch neue, schnell produktiv gehende Teilanwendungen, die zu einem frühen und begeisternden Buy-In der verantwortlichen Stakeholder führen.

Vom Kostenstandpunkt und im Bereich Risikominimierung macht es einen riesigen Unterschied, ob man eine vollständige Neuentwicklung plant und durchführt oder eine Migration anstrebt. Mit klarem Vorteil für die Migration.

Wir sprachen über ihr PITSS.CON-Toolset aber auch über Projekte, speziell Migrationsprojekte. Ist ihre Firma eher eine Produktfirma oder eher eine Projekt- und Consultingfirma?

Wir sind beides. Eine Software-Firma, die ein innovatives Software-Paket – PITSS.CON – entwickelt und an Kunden verkauft. Wir sind jedoch ebenso eine Consultingfirma, die Migrationsprojekte, aber auch andere Projekte im Bereich Forms, Reports oder APEX durchführt. Hat ein Migrationskunde PITSS.CON bereits im Einsatz, nutzen wir diese Lizenz für das Migrationsprojekt. Ansonsten stellen wir PITSS.CON für die Projektzeit zur Verfügung. Wir nutzen also unser eigenes Produkt intensiv in unseren Kundenprojekten. Manche sagen, dass man das dem Produkt ansieht – im positiven Sinn. Ein Migrationsprojekt ohne PITSS.CON ist keine Option, die wir anbieten.

Zum Schluss noch die Frage nach der Zukunft ihres Produktes. Wohin geht die Reise mit Ihrem PITSS.CON-Toolset?

Neben unseren Kunden, die PITSS.CON einsetzen, sind wir selbst ein großer PITSS.CON-Anwender bei vielen unserer Consulting-Projekte. Daraus ergeben sich stets neue Anforderungen, die in die neuen Versionen unseres Produktes einfließen. Auch die neuen Releases unserer Zielsysteme wie APEX oder Jasper müssen sich in unseren neuen Versionen wiederfinden. In diesem Jahr haben wir bereits zwei neue Release-Stände freigegeben. Das ist alles sehr dynamisch.

Dass wir auf dem Gebiet der Analyse von Software-Produkten sehr innovativ unterwegs sind, hat sich auch beim BMBF (Bundesministeriums für Bildung und Forschung) herumgesprochen.

2021 haben wir ein erstes Förderprojekt gewonnen, das sich mit der Identifikation von Geschäftsprozessen in bestehenden, eigenentwickelten Anwendungen beschäftigt hatte. Dieses Förderprojekt hat unsere Produkt-Suite PITSS.CON um ein weiteres, äußerst leistungsstarkes Modul bereichert, das seit 2023 als ProFind vermarket wird. Mittels ProFind treten, bei Optimierung und Modernisierung einer Anwendung, die implementierten und von den Fachabteilungen betriebenen Prozesse in den Vordergrund. Altlasten, wie nicht mehr verwendete Funktionen, schlechte oder überladene Programmierung, werden herausgefiltert, sodass der Blick frei ist, für das was wirklich benötigt wird.

Das zweite, aktuell laufende Förderprojekt beschäftigt sich mit autonomen oder teilautonomen Anpassungen beziehungsweise eigenständigen Reaktionen einer Anwendung. Sehr spannend.

Die Förderprojekte wie auch die weiteren PITSS-internen Entwicklungen sind getrieben von Marktanforderungen sowie technischen Möglichkeiten und haben uns zu einem mehr fließenden Patch-Prozess unserer Produkte bewogen. Teillösungen werden schon frühzeitig aus Entwicklungsprojekten herausgelöst und nach einigen Tests in die vorrangig Cloud-basierenden Produkte eingebunden. So profitieren unsere Kunden und Projekte schnell von den neuen Innovationen.

Diesen Innovationsgrad und unseren Status als Thought-Leader im Bereich der Anwendungsmodernisierung werden wir mit Nachdruck weiterverfolgen.

Vielen Dank Herr Gaede für Ihre Zeit und alles Gute für Ihr Team.

ANDREAS GAEDE

Software-Visionär, Mitbegründer von PITSS, ein Experte im Simplifying Complexity. Andreas Gaede treibt zukunftsweisende, leistungsstarke Software-Lösungen voran, die sich in den letzten 25 Jahren in unzähligen Modernisierungsprojekten auf der ganzen Welt bewähren konnten– immer mit dem Ziel ein Optimum aus Kosten und Zeit bei höchster Qualität zu erreichen. Um ein solches Optimum planbarer und verlässlich selbst in hoch komplexen Modernisierungsprojekten erreichen zu können, werden die Software-Lösungen in eine hoch flexible, den Mensch führende Methodik eingebettet. Für seine innovative Produktentwicklung und kreativen Ansätze wurde Andreas Gaede mit seinem Team vom Bundesministerium für Bildung und Forschung nicht nur mehrfach ausgezeichnet und gefördert, sondern tritt aktuell den Beweis der Leistungsfähigkeit der entwickelten Lösungen in zahlreichen, erfolgreichen Modernisierungsprojekten von Oracle Forms nach APEX an.

BEST OF DOAG ONLINE

Eine Auswahl der besten DOAG News Oktober 2024 bis Februar 2025

DOAG Datenbank Kolumne: Schemaprivilegien in Oracle 23ai

Cornelia Heyde erläutert in ihrer Kolumne die verschiedenen Formen der Schemaprivilegien-Vergabe.

DOAG Datenbank Kolumne: Mein erstes Jahr als DBA

Schon früh wusste ich, dass es in Richtung Datenbanken gehen sollte. Bereits in der Schule hatte ich Freude daran, mit Access zu arbeiten, und gemeinsam mit meinem Bruder besuchte ich Informatikvorlesungen an der Universität, die sich speziell mit Datenbanken befassten.

KI in Organisationen, Domainwissen, Agenten und Datenschutz – DOAG.tv mit Johann-Peter Hartmann und Dr. Benjamin Linnik

'High Energy' und elektrisierende Aussagen über KI im Format DOAG@Talk mit dem CTO von Mayflower und dem DOAG Themenverantwortlichen für Data Science: "KI wird unsere Industrialisierung. Wir können das live miterleben, und das ist total cool."

KI im Wissensmanagement: Chancen, Herausforderungen und Learnings

DOAG.tv mit Sandra Starke, Bereichsleiterin IT-Betrieb der AOK Rheinland-Pfalz/Saarland, und Andreas Buckenhofer, DOAG Themenverantwortlicher Data Governance und Data Quality und Mitglied der Delegiertenversammlung Data Analytics.

Nachhaltigkeit und Konferenzen oder: Und jedem seinen Kaffeebecher – Teil 3

DOAG-Redakteur Marcos López beschäftigt sich gerne mit Nachhaltigkeit – und mit Konferenzen. Die DOAG veranstaltete im November, dem 'Konferenzherbst', gleich drei davon. Was ein Kaffeebecher damit zu tun hat, wird im Verlauf des dritten Teils endlich verraten.

KI als Fahrrad fürs Hirn: Schneller und weiter denken als zu Fuß

Im DOAG.tv spricht Oliver Szymanski mit dem Unternehmer, Sachbuchautor, Keynote Speaker und Experten für digitale Transformation, Ömer Atiker, über das Potential von Künstlicher Intelligenz und wie diese seine LinkedIn-Postings gestaltete.

Wir begrüßen unsere neuen Mitglieder

Natürliche Mitglieder:

- Dr. Konstantin Hopf
- Marius Stein
- Christoph Förster
- Elvira Zygmunt
- David König
- Christian Bonny
- Dr. Sandra Signore

Termine

Februar 02

12.02.2025
Regionaltreffen Freiburg
1. Kubernetes in der Oracle Cloud 2. 23ai – wie weiter
Freiburg

14.02.2025
Oracle AutoUpgrade in a Nutshell
DB WebSession mit Christian Pfundtner
Online

27.02.2025
DOAG DevTalk: Modernes SQL: Graph-SQL
DevTalk mit Matthias Schulz und Christian Schwitalla
Online

April 04

01.04. - 03.04.2025
JavaLand 2025
Zwei ereignisreiche Konferenztage mit anschließendem Schulungstag rund um das Java-Ökosystem
Nürburgring

03.04. - 05.04.2025
DOAG Leitungskräfteforum
Workshop der Leitungskräfte der DOAG
Berlin

05.04.2025
DOAG Delegiertenversammlung
Berlin

Mai 05

13.05. - 15.05.2025
APEX connect 2025
Konferenz mit zahlreichen Vorträgen und Workshops zu den Themen APEX, PL/SQL, JavaScript und Solutions
Europa-Park, Rust

14.05. - 15.05.2025
DOAG 2025 Datenbank mit Cloud Infrastructure
Konferenz rund um die Oracle Datenbank und Cloud Infrastructure
Europa-Park, Rust

Impressum

Red Stack Magazin wird gemeinsam herausgegeben von DOAG 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 ist die Community-Publikation für angewandte Informations- und Kommunikationstechnologie (ITK) im Raum Deutschland, Österreich und Schweiz. Es setzt bewusst auf Technologieoffenheit mit Blick auf Anwendung und IT-Innovationen.

Es bildet die Interessensschwerpunkte der Anwenderinnen und Anwender ab – von Cybersicherheit bis Datenschutz, von Datenbank und Development über Data Analytics bis Digitalisierung, von Cloud und Infrastruktur über Künstliche Intelligenz bis Open Source und Soft Skills – vermittelt praktisches Wissen und fördert den Know-how-Transfer und die Netzwerkbildung zwischen den Leserinnen und Lesern.

Die Inhalte des Red Stack Magazin werden von ausschließlich ehrenamtlichen Autorinnen und Autoren eingereicht und von der Redaktion aufbereitet.

Red Stack Magazin 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. DOAG e.V. hält 100 Prozent der Stammeinlage der DOAG Dienstleistungen GmbH. DOAG e.V. wird gesetzlich durch den Vorstand vertreten; Vorsitzender: Björn Bröhl

Redaktion:
Sitz: DOAG Dienstleistungen GmbH
(Anschrift s.o.)
ViSdP: Fried Saacke
Redaktionsleitung Red Stack Magazin:
Martin Meyer, Marcos López.
Kontakt: redaktion@doag.org

Autorinnen und Autoren dieser Ausgabe
(in alphabetischer Reihenfolge):
Meris Bihorac, Bruno Cirone,
Daniel Eiduzzis, Markus Flechtner,
Andreas Gaede, Alexander Giesbrecht,
Cornelia Heyde, Sven Illert,
Marco Pachaly-Mischke, Thomas Petrik,
Martin Meyer, Semjon Mössinger,
Fabian Neureiter, Tobias Otte,
Detlef E. Schröder, Stefan Seck,
Günther Stürner, Bastian Weinlich

Titel, Gestaltung und Satz:
Diana Tkach
DOAG Dienstleistungen GmbH
(Anschrift s.o.)

Fotonachweis:
Titel: © freepik | *www.freepik.com*
S. 12: © fietzfotos | *www.pixabay.com*
S. 18: © HI-MXWG | *www.pixabay.com*
S. 24: © Thanh_Do | *www.pixabay.com*
S. 30: @ artandmusic90 | *www.pixabay.com*
S. 38: © Tumisu | *www.pixabay.com*
S. 44: © Martin_Meyer | *Martin Meyer*
S. 48: © treellercoaster | *www.pixabay.com*
S. 52: © MarcVanduffel | *www.pixabay.com*
S. 56: © viarami | *www.pixabay.com*
S. 66: © OrMaVaredo | *www.pixabay.com*
S. 74: © chathuraanuradha |*www.pixabay.com*
S. 82: © SeminPaek | *www.pixabay.com*
S. 93: freepik | *www.freepik.com*

Anzeigen:
sponsoring@doag.org

Mediadaten und Preise:
www.doag.org/go/mediadaten

Druck:
WIRmachenDRUCK GmbH,
www.wir-machen-druck.de

Inserentenverzeichnis

EARLY BIRD BIS 01.04.25

APEX connect

EUROPA-PARK Rust

13._–15.
MAI 25

apex.doag.org

DOAG

EARLY BIRD BIS 20.05.

#CLOUDLAND2025

DAS CLOUD NATIVE FESTIVAL

1. – 4. JULI 2025 • IM HEIDEPARK IN SOLTAU

CloudLand
WWW.CLOUDLAND.ORG

Das Event der Deutschsprachigen Cloud Native Community