

Red Stack

Magazin

DOAG

SOUG
swiss oracle
user group

AOUG
AUSTRIAN ORACLE USER GROUP

DEVELOPMENT



Aus der Praxis

Spracherkennung mit
APEX und Whisper



Im Interview

Matthias Schulz, CEO
Schulz IT Services

Development

Programmieren nach
Daten



e3mag.com

DEUTSCH

Information und
Bildungsarbeit
von und für die
SAP®-Community

The global
independent
platform for the
SAP® community

ENGLISH

ESPAÑOL

La plataforma global
e independiente
para la
comunidad SAP®

La plateforme
indépendante
mondiale de la
communauté SAP®

FRANÇAIS

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



Björn Bröhl
DOAG Vorstandsvorsitzender

Liebe Mitglieder, liebe Leserinnen und Leser,

wenn wir über Entwicklung sprechen, geht es längst nicht nur um Codezeilen. Development bedeutet für mich, Ideen zum Leben zu erwecken, Neues auszuprobieren und dabei gemeinsam Wege zu finden, wie wir IT und ihre Anwendung besser machen können. Genau deshalb ist dieses Thema auch in der DOAG so wichtig: Entwicklerinnen und Entwickler sind diejenigen, die unsere digitale Zukunft aktiv gestalten.

In dieser Ausgabe des Red Stack Magazins dreht sich alles um Dich und Deine Arbeit als Developer. Matthias Schulz teilt im Interview seine Sicht auf aktuelle Trends und Herausforderungen. Jürgen Sieben zeigt im zweiten Teil von „Programmieren nach Daten“, wie sich Entwicklungsansätze weiterdenken lassen. Und Stephan La Rocca wirft die spannende Frage auf, warum Anwendungen oft mehr über Datenmodelle wissen als die Datenbanken selbst.

Ein besonderer Schwerpunkt liegt auf APEX – mit gleich mehreren praxisnahen Beiträgen: von moderner Suche mit Vektordaten (Florian Graßhoff) über Spracherkennung mit Whisper (Dennis Amthor) bis hin zu neuen Möglichkeiten rund um JSON und Low Code (Carsten Czarski). Ergänzt wird das durch Lösungen für den Datenaustausch zwischen APEX und Desktop-Anwendungen (Sebastian Reinhard) und die Idee eines ganz persönlichen KI-Assistenten (Jörg Doppelreiter).

Natürlich schauen wir auch auf die Basis, die vieles erst möglich macht: Michael Schulze gibt Dir einen Überblick über die Oracle Engineered Systems.

Diese Vielfalt zeigt: Development ist nicht nur ein Bereich unter vielen, sondern ein zentrales Thema unserer Community. Genau deshalb freue ich mich besonders, Dir ein Highlight anzukündigen: **Im nächsten Jahr starten wir mit der DOAG ein ganz neues Event – DevLand.** Dort wollen wir uns mit Dir und vielen anderen Entwicklerinnen und Entwicklern austauschen, voneinander lernen und die neuesten Trends hautnah erleben. Die DevLand wird weit mehr als eine gewöhnliche Developer-Konferenz sein – was euch im Frühjahr in Deutschlands größtem Freizeitpark erwartet, ist eine digitale Erlebniswelt! Hier werden langjährige Expertinnen und Experten auf Young Professionals und Studierende aus den Bereichen Software-Architektur, Künstlicher Intelligenz und Cloud-Strategie treffen, um gemeinsam die Zukunft moderner Softwarelandschaften zu entwerfen.

Ich lade Dich herzlich ein: Sei dabei und bring Deine Ideen ein!

Viel Freude mit dieser Ausgabe und bis bald – vielleicht schon auf der DevLand.

Herzliche Grüße

Björn Bröhl



Ausgabe Nr. 4/2025
auf Abruf!

DOAG WEBSESSION

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

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



www.doag.org/go/websessions



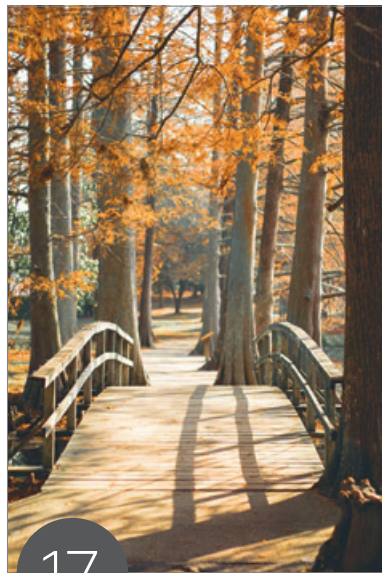
*Die Buchung der WebSessions erfolgt ganz einfach über unseren Shop. Mitglieder erhalten im Buchungsprozess automatisch **100 % Rabatt.**



Matthias Schulz,
CEO Schulz
IT Services



Programmieren nach
Daten



Warum weiß die
Anwendung mehr
über ihr Datenmodell
als die Datenbank?

Einleitung

- 3 Editorial
- 6 Timeline
- 8 „Das Verständnis für fundamentale Datenbankprinzipien wird wichtiger, nicht weniger wichtig“
Interview mit Matthias Schulz, CEO Schulz IT Services

Development

- 12 Programmieren nach Daten
Jürgen Sieben

SQL & PL/SQL

- 17 Warum weiß die Anwendung mehr über ihr Datenmodell als die Datenbank?
Stephan La Rocca

APEX

- 24 Modernes Suchen in APEX 24.2
Florian Graßhoff
- 32 JSON-Dokumente und Low Code: APEX JSON Sources
Carsten Czarski

KI

- 44 Hör mal, wer da flüstert – Spracherkennung mit APEX und Whisper
Dennis Amthor

Solutions

- 52 Mein persönlicher KI Assistent – Lokal, ohne Cloud, mit Oracle APEX und Co.
Jörg Doppelreiter



Modernes Suchen in APEX 24.2

62 Bidirektionaler Datenfluss zwischen Oracle APEX und Desktop-Anwendungen
Sebastian Reinhard

Engineered Systems

66 Oracle Engineered Systems 2025 – ein Überblick
Michael Schulze



44

Hör mal, wer da flüstert – Spracherkennung mit APEX und Whisper



52

Mein persönlicher KI-Assistent – Lokal, ohne Cloud, mit Oracle APEX und Co.



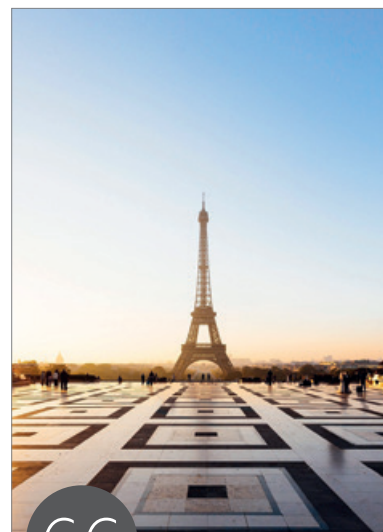
32

JSON-Dokumente und Low Code: APEX JSON Sources



62

Bidirektionaler Datenfluss zwischen Oracle APEX und Desktop-Anwendungen



66

Oracle Engineered Systems 2025 – ein Überblick

News

22 Oracle Datenbanken Monthly News

Intern

74 DOAG Datenbank Kolumne: Das Problem des Autors beim Schreiben

76 Best of DOAG Online

77 Neue Mitglieder + Termine

78 Impressum + Inserenten

TIMELINE

27. August 2025

Der DB Talk „Datenbank trifft KI – Was auf DBAs zukommt, wenn die Fachabteilung plötzlich KI will“ mit Referent Jürgen Paasch und Birgit Arndt steht auf dem Programm.

4. September 2025

Beim DevTalk mit Madeleine Reinwald und Arne Hattendorf dreht sich alles rund um das Thema „KI Fails - 10 Fragen 10 falsche Antworten“.

12. September 2025

In einer DB WebSession mit Marc Wagner wird das Thema „Oracle Zero Downtime Migration (ZDM): Das Werkzeug zur Migration in die Oracle Cloud oder auf Exadata“ behandelt. Es moderiert Christian Trieb.

16. September 2025

Das Java Forum Nord 2025 steht als Community-Konferenz der Java User Groups aus dem Norden in Hannover an.

16. und 17. September 2025

Die herstellerneutrale Kongressmesse für Programmierfreie Software-Entwicklung German Low Code Day 2025 findet in Hannover statt. Es handelt sich um eine Veranstaltung der Low-Code Association e.V. in Kooperation mit der DOAG.

18. September 2025

Beim Regionaltreffen Hamburg stehen Development und Infrastruktur im Mittelpunkt. Die Teilnehmenden erwarten drei Vorträge. „AI-Innovationen in Oracle APEX: Generative Development & Neue Features“, „Datenbankentwicklung, aber dieses mal in modern, bitte!“ von Martin Bach und „Oracle Dataguard insights“ von Benjamin Kurschies.

23. September 2025

Das Regionaltreffen Berlin Brandenburg findet in Berlin statt. Auf der Agenda stehen die zwei Sessions „Oracle Data Guard 19c: Manuell und per Cloud Control erstellen“ von Brigit Arndt sowie „Wenn die SE zur Rettung wird: Standby-Konzepte ohne Enterprise Features“ mit Referent Björn Kaltfofen.

24. September 2025

Beim DOAG DBTalk mit Christian Pfundtner und Christian Trieb geht es um die „Zukunft des DBA“





SOUG Day Zürich

23. Oktober 2025

Oracle Know-how. Global Experts. Real Insights.

Maria Colgan, Heli Helskyaho, Dani Schnider, Martin Klier,
Hans Viehmann, Christoph Lutz – and many, many more.

Highlight:

Robert Szilinski mit „KI – Die Illusion vom perfekten
Ersatz und die Kraft des Menschseins“.

=> Anmeldung unter soug.ch

Many thanks to Zürcher Kantonalbank for hosting us.

SOUG



*„Das Verständnis für
fundamentale Datenbank-
prinzipien wird wichtiger,
nicht weniger wichtig“*

Christian Schwitalla, Stv. Community-Leiter der DOAG Development Community, unterhielt sich mit Matthias Schulz, Geschäftsführer der Schulz IT Services über Oracle-Datenbank-Entwicklung im Wandel der Zeit.

Bitte stellen Sie sich unseren Lesern vor. Wer sind Sie, welche Tätigkeitsschwerpunkte haben Sie?

Mein Name ist Matthias Schulz und ich bin der Geschäftsführer der Schulz IT Services GmbH in Nürnberg. Seit über 34 Jahren arbeite ich im IT-Bereich und bin heute als zertifizierter Experte für Oracle-Datenbanken und Data Vault tätig. Meine Kernkompetenzen liegen in der Oracle-Datenbankentwicklung (Design, Tuning, Programmierung). In meiner Tätigkeit als externer Berater konnte ich die Zukunft renommierter Unternehmen aus verschiedenen Branchen, von Banken und Versicherungen über Automobilindustrie und Handel bis hin zur Marktforschung, erfolgreich mitgestalten. Als Dozent für Informatik an der DHBW Heidenheim unterrichte ich seit 2016 das Fach "Datenbanken I". Zudem halte ich regelmäßig Vorträge auf der DOAG-Konferenz und schreibe Artikel für das Red Stack Magazin.

Wie lange beschäftigen Sie sich mit der Oracle-DB-Programmierung?

Mit der Oracle-Datenbankprogrammierung beschäftige ich mich seit Anfang 2001, also fast 25 Jahre. Ich habe die Evolution der Oracle-Datenbank von Version 8i bis hin zur aktuellen Oracle Database 23ai miterlebt und die Neuerungen aktiv bei meinen Kunden umgesetzt.

Was sind die wichtigsten Trends in den letzten Jahren?

Die **Cloud** hat sich etabliert und Oracle bietet seinen Kunden die Freiheit, auf den Cloudanbieter ihrer Wahl zu setzen. Egal, ob man sich für AWS, Azure, Google oder Oracle entscheidet, die Oracle-Datenbank steht zur Verfügung.

Die **Oracle Autonomous Database** reduziert den administrativen Aufwand erheblich und ermöglicht es Entwicklern, sich auf die Anwendungslogik zu konzentrieren.

Mit dem Ansatz der **Converged Database** vereint Oracle relationale, JSON-, XML-, Graph-, Spatial- und Blockchain-Workloads in einer einzigen, hochperformanten Datenbank-Engine und eliminiert so den Bedarf an mehreren spezialisierten Systemen.

Mit den **JSON Relational Duality Views** ist die Frage nach dem richtigen Modell für Speicherung von Daten endgültig entschieden: Relational erweitert um JSON. Gleichzeitig erhalten die Entwickler die völlige Freiheit, dieselben Daten sowohl als relationale Tabellen als auch als maßgeschneiderte JSON-Dokumente zu verwenden, und das ohne Duplikation oder Replikation der Daten und ohne Konsistenzprobleme.

Der wichtigste Trend ist zweifellos die Integration von **Künstlicher Intelligenz** direkt in der Datenbank:

Select AI ermöglicht die Abfrage und Auswertung von Daten in natürlicher Sprache, ganz ohne technisches Wissen.

AI Vector Search revolutioniert die Art, wie wir mit unstrukturierten Daten arbeiten. Anstatt Daten für AI-Verarbeitung aus der Datenbank zu extrahieren, bringen wir die künstliche Intelligenz zu den Daten.

Sind klassische Themen wie DB-Modellierung oder Performance-Tuning immer noch ein Thema in der Praxis?

Absolut! Diese fundamentalen Disziplinen bleiben von zentraler Bedeutung, auch wenn sich die Werkzeuge und Ansätze weiterentwickelt haben.

Datenbankmodellierung wird sogar wichtiger, da moderne Anwendungen immer größere Datenmengen, in komplexeren Datenstrukturen und mit den vielfältigen Datentypen verwalten müssen. Die Fähigkeit, sowohl strukturierte als auch unstrukturierte Daten effizient zu modellieren, wird durch Features wie JSON Relational Duality Views noch essenzieller. Gute Datenbankmodellierung ist nicht überholt, sie ist unentbehrlich!

Relationale Datenbankstrukturen gelten als unflexibel und schwer zu ändern – schuld ist hier meist eine starre hierarchische Modellierung. Mit modernen flexiblen Ansätzen, wie zum Beispiel Data Vault, wird das Datenbankmodell so agil und flexibel, dass die Anwendungsentwickler kaum noch Schritt halten können.

Performance-Tuning bleibt essenziell, besonders bei der Verarbeitung großer Datenmengen. In meinen Projekten bei internationalen Großkonzernen geht es um die Verarbeitung von Massendaten unter hohem Zeit- und Kostendruck. Durch geeignete Tuning-Maßnahmen konnte ich schon viele Prozesse um das Tausend- bis Zehntausendfache beschleunigen und so die Laufzeit von mehreren Tagen auf wenige Minuten reduzieren.

Die Tools werden intelligenter, die Autonomous Database automatisiert viele Tuning-Aufgaben und die Künstliche Intelligenz übernimmt viele Arbeitsschritte, aber das Verständnis für die zugrunde liegenden Prinzipien bleibt unerlässlich. Entwickler müssen aber weiterhin verstehen, wie Datenmodellierung, SQL, Tuning und Query-Optimierung funktionieren.

Im Rahmen Ihrer Tätigkeit als Dozent an der Dualen Hochschule Baden-Württemberg haben Sie viel Kontakt zur jungen Generation von Informatikern. Wie kommen hier Themen wie DB-Programmierung an?

Viele Studierende unterschätzen zunächst die Bedeutung von Datenbanken und erwarten vor allem moderne Frameworks und Cloud-Services. Begeisterung entsteht, wenn sie erkennen, dass nahezu jede Anwendung – von Social Media bis AI – darauf aufbaut. Mit praxisnahen Übungen, etwa über Oracle LiveSQL oder die Free Tier Cloud, arbeiten sie direkt mit professionellen Tools. So wird klar: Datenbanken sind keine „alte Technik“, sondern die Grundlage für Innovationen wie KI und Real-Time Analytics.

Welche Rolle spielt die DOAG Konferenz + Ausstellung als Ort für den Erfahrungsaustausch für klassische Oracle-Themen?

Die DOAG Konferenz + Ausstellung ist und bleibt der zentrale Knotenpunkt der deutschsprachigen Oracle-Community. Als langjähriger Referent und aktives Mitglied der Datenbank-Community erlebe ich Jahr für Jahr den enormen Wert dieses Erfahrungsaustauschs. Die DOAG bietet eine praxisorientierte Plattform, auf der Experten aus verschiedenen Branchen ihr Wissen und ihre Erfahrungen austauschen. In meinen Vorträgen, die Themen von Performance-Tuning bis zu Oracle 23ai-Features umfassen, wird theoretisches Wissen in konkrete

Problemlösungen überführt. Besonders wertvoll sind die informellen Gespräche zwischen den Sessions, die oft zu neuen Lösungsansätzen und langfristigen Partnerschaften führen. Trotz des Schwerpunkts auf klassischen Oracle-Themen zeigt die DOAG mit der Integration der Konferenz KI Navigator eine starke Innovationskraft.

Große Konzerne nutzen längst die Vorteile der Cloud. Wie kann man den Einstieg in die Cloud kleineren und mittleren Unternehmen oder auch dem öffentlichen Dienst erleichtern?

Mit dem **Oracle Free Tier** und der **Always Free Autonomous Database** (inkl. APEX) lassen sich Erfahrungen sammeln und erste PoCs starten, und das ganz ohne Kosten oder Risiken. Ich würde mit unkritischen Systemen wie Entwicklungs- oder Testumgebungen beginnen, dann Backup und Disaster Recovery in die Cloud auslagern und final die produktiven Systeme migrieren.

Welche sind die wichtigsten New Features in der Oracle DB 23ai?

Oracle Database 23ai bringt über 250 neue Features mit, von denen einige die Datenbankwelt fundamental verändern werden.

AI Vector Search ermöglicht semantische Suchen in unstrukturierten Daten und die Entwicklung von Retrieval-Augmented Generation (**RAG**)-Anwendungen direkt in der Datenbank. Unternehmen können jetzt Large Language Models mit ihren Geschäftsdaten verbinden, ohne Daten extrahieren zu müssen.

JSON Relational Duality Views lösen ein jahrzehntelanges Problem: Die Wahl zwischen relationalen und dokumentenorientierten Datenmodellen entfällt.

Select AI ermöglicht Datenbankabfragen in natürlicher Sprache. Die Liste neuer **SQL-Features** ist beeindruckend und vereinfacht die tägliche Arbeit mit der Datenbank enorm. Es gibt neue Datentypen (BOOLEAN, JSON), 4096 Spalten pro Tabelle, Table Value Constructor, GROUP BY Alias, IF EXISTS und IF NOT EXISTS bei DDL, Joins für DELETE und UPDATE-Statements, um nur einige zu nennen.

Für mich ist Oracle 23ai das bisher beste Release der Datenbank und es stellt, neben einer beeindruckenden Anzahl an Neuerungen, vor allem den Entwickler ins Zentrum vieler Verbesserungen. Es hat noch nie so viel Spaß gemacht, mit einer Oracle-Datenbank zu arbeiten.

Können Sie einen Ausblick hinsichtlich zukünftiger Development-Entwicklungen geben?

Die Zukunft der Oracle-Datenbankentwicklung wird von mehreren Megatrends bestimmt:

Konvergierte Datenbanken setzen sich durch. Entwickler müssen nicht mehr zwischen spezialisierten Datenbanken wählen, sondern können alles in einem System verwalten.

Künstliche Intelligenz ist nicht nur ein Feature, sondern der grundlegende Entwicklungsansatz. Code und Datenlogik werden generiert statt handgeschrieben.

Die **Cloud** wird zum Standard, und On Premise eher die Ausnahme sein.

Wird die AI den Alltag der DB-Programmierer so weit verändern, dass eine völlige Neuorientierung notwendig sein wird?

Nein, eine völlige Neuorientierung wird nicht notwendig sein, aber eine kontinuierliche Weiterentwicklung ist unerlässlich. Die AI wird Datenbankentwickler nicht ersetzen, sondern unterstützen. Tools wie GitHub Copilot steigern bereits heute die Produktivität ganz erheblich. In der Datenbankentwicklung übernimmt die AI schon heute repetitive Aufgaben wie Code-Optimierung, Bug-Detection und Test-Generierung. Der Datenbankentwickler wird zunehmend zum "Data Engineer", der AI-Tools orchestriert und komplexe Problemlösungen entwirft. Das Verständnis für fundamentale Datenbankprinzipien wird wichtiger, nicht weniger wichtig. Die AI-Revolution bietet enorme Chancen für alle, die bereit sind, sich weiterzuentwickeln und die neuen Technologien zu nutzen.



MATTHIAS SCHULZ

Matthias Schulz ist seit 2002 Geschäftsführer der Schulz IT Services GmbH und dort als Consultant, Trainer und Konferenzsprecher im Datenbankumfeld tätig. Seine Schwerpunkte sind Data Vault, High-End-Systeme (Exadata, Exasol) und die Oracle-Datenbank (Entwicklung, Tuning, Architektur, ETL, Testing und Java in der Datenbank). Zudem ist er Dozent für Informatik (Datenbanken) an der Dualen Hochschule Baden-Württemberg (DHBW).

2025
DOAG
Konferenz + Ausstellung



Die **ORACLE**
Anwenderkonferenz

Nürnberg | 18. – 21. Nov.



anwenderkonferenz.doag.org



Programmieren nach Daten

Jürgen Sieben. ConDeS

Dies ist eine kleine Reihe mit Anwendungsbeispielen, die zeigen, wie der Einsatz von SQL in der Programmierung der Datenbank nicht nur erhebliche Codemengen einsparen, sondern zudem auch die Effizienz und Funktionalität Ihrer Anwendungen erhöhen kann. Hier geht es nicht um abgefahrene Spezialfunktionen, sondern um eine Erweiterung des Blickwinkels in der Programmierung mit PL/SQL.

Entscheidungstabellen (Decision Table)

Anwendungsfälle für die folgende Strategie treten sehr häufig auf, doch werden die Möglichkeiten von SQL nur selten verwendet. Ich möchte mit einem fiktiven, aber realitätsnahen Szenario solche Fälle aufzeigen und starte mit einer Beschreibung.

Szenario

Abbildung 1 und Abbildung 2 zeigen Ausschnitte aus einem fiktiven Feinkonzept, das sich auf die Verwaltung von Konten einer Buchhaltungsanwendung bezieht.

Es ist nicht wichtig, zu wissen, was Verdichtungskonten oder Controlling-Buchungen sind, entscheidend ist, zu verstehen, dass offensichtlich Kontenkreisnummern existieren, die durch die Zuordnung auf Verdichtungskonten zusammengefasst werden und einige von ihnen zusätzliche Buchungen benötigen.

Dies gilt genauso für Aktivierungskonten. Doch ist beim Vergleich der beiden Ausschnitte auffällig, dass einige Kontenkreise offensichtlich spezielle Verdichtungskonten besitzen und auch Aktivierungskonten sind. In der Anwendung, und das ist die Aufgabe dieses Szenarios, muss entschieden werden, über welches konkrete Verdichtungskonto ein konkretes Konto geführt wird, ob es Controlling-Buchungen benötigt und ob es ein Aktivierungskonto ist.

Wie implementieren wir so etwas? Die naheliegendste Lösung besteht darin, im Code eine `case`-Anweisung vorzuhalten, die die unterschiedlichen Fälle abarbeitet. Das ist einfach und sollte daher keine besonderen Probleme bereiten.

Oder doch? Vielleicht haben wir etwas übersehen. Im Konzept steht, dass die Daten aktuell zum Stand Juni beziehungsweise Oktober 2020 sind. Naja, wir sind etwas spät dran, aber jetzt wird die Anforderung ja umgesetzt. Sicherheitshalber rufen wir den Kunden noch schnell an und versichern uns, dass dieser Stand noch aktuell ist. Und es kommt, wie es kommen muss: Der Stand ist nicht mehr aktuell, zahlreiche Änderungen haben sich ergeben. „Kein Problem“, denken Sie vielleicht, „dann integrieren wir halt den aktuellen Stand und liefern das Produkt aus.“

Doch es entstehen Zweifel: „Was ist, wenn in Zukunft Änderungswünsche kommen?“ Zwei Probleme tauchen auf: Zunächst einmal bedeutet das, dass der Code neu ausgeliefert werden muss, denn die `case`-Anweisung muss angepasst werden und die findet sich irgendwo im Code. Änderungen am Code unterliegen aus guten Gründen einem strikten Zeitplan, nur zu bestimmten Zeiten darf im Regelfall Software ausgeliefert werden, und das bedeutet, dass jeder Änderungswunsch durch diese Auslieferungsrhythmen bestimmt wird. Schlimmer ist aber: Wer muss diese Änderungen durchführen? Immer der, der fragt, daher werden Sie stetig mit Änderungswünschen zu dieser Anweisung überhäuft. Geht das nicht besser?

Lösungsansatz

Wir reagieren seltsam: Wir erstellen eine neue Tabelle, die keine andere Aufgabe haben wird, als diese Entscheidung für uns zu treffen. Die Erstellung von Tabellen gilt als schwergewichtig, daher werden neue Tabellen nur aus gutem Grund angelegt. Aber es ist mein Ehrgeiz, Ihnen zu zeigen, was ein guter Grund sein könnte. Listing 1 zeigt die Daten aus dem Rohkonzept, übernommen in eine Tabelle.

Wichtig ist, dass immer dann, wenn das Konzept keine Angabe über die entsprechende Kategorie macht, in der Tabelle ein `null`-Wert verwendet wird und nicht etwa der Standardwert. Der Grund hierfür liegt darin, dass, wenn sich der Standard ändert, nicht mehr eindeutig entschieden werden kann, ob dieser geänderte Wert den bisherigen Standardwert ersetzen soll. Wir schreiben also nur hin, was wir wissen. Den Standardwert erkennen wir daran, dass sein Kontopattern einen `null`-Wert enthält.

Anreicherung der Standarddaten

Die Entscheidung, den Standardwert als `null`-Wert zu realisieren, ist für den praktischen Einsatz nur bedingt geeignet, denn es muss ein Verdichtungskonto angeboten und die Entscheidung über Controllingbuchungen und Aktivierungskonto getroffen werden. Daher ist der erste Schritt, dass wir in einer Abfrage die Standardwerte „durchscheinen“ lassen, wo ansonsten ein `null`-Wert existiert.

Das technische Mittel, das wir hierfür einsetzen, ist ein `cross`-Join auf dieselbe

Tabelle. Die Idee: Wir selektieren in einer Unterabfrage nur die Zeile der Tabelle mit den Standardwerten und hängen diese über einen `cross`-Join an jede Zeile der Tabelle an. Anschließend wird über eine `coalesce`-Anweisung geprüft, ob die aktuelle Zeile einen Wert enthält und, wenn dies nicht der Fall ist, stattdessen der Wert aus der Standardzeile verwendet. Dadurch verdichtet sich die Abfrage und die Standardwerte scheinen durch. Listing 2 zeigt dieses Verfahren und die resultierenden Werte.

Damit ist alles vorbereitet und wir können beginnen, die eigentliche Entscheidung zu treffen. Die Frage lautet: Welche Daten gelten für eine gegebene Kontonummer?

Entscheidungsfindung

Vielleicht haben Sie erwartet, dass eine `case`-Anweisung zum Einsatz kommen wird. Das ist aber nicht der Fall, das kann SQL auch anders und einfacher lösen. Würden wir eine `case`-Anweisung verwenden, hätten wir das Problem der Pflege dieser Anweisung lediglich in eine `select`-Anweisung verlegt. Stattdessen überlegen wir, wie SQL die Zuordnung einer konkreten Kontonummer zum *bestmöglich* passenden Datensatz dieser Entscheidung berechnen kann. Hierfür erstellen wir zunächst eine kleine Abfrage mit Testdaten, die wir zur Prüfung unserer Strategie einsetzen können (siehe Listing 3).

Um einen möglichen Match mit unserer Entscheidungstabelle herzustellen, verwenden wir die Kontonummer. Auch in der `case`-Anweisung hätten wir einen Ausdruck wie `case when KTO_NR like KTO_PATTERN%` geschrieben und genau das wird unsere Join-Bedingung, wie Listing 4 zeigt.

Es mag zunächst überraschen, dass mehr Zeilen als Testkonten zurückgeliefert werden, doch muss dies so sein, denn jede Kontonummer wird mindestens einen Match mit der Standardzeile haben (`KTO_NR like '%'` stimmt immer), zudem noch die Matches über verschiedene Kontopatterns.

Sehen wir uns die Daten näher an, stellen wir fest, dass die erste Kontonummer nur einen Match mit Standard hat, während andere Kontonummern, zum Beispiel `503801234` drei Matches haben. Ein Vergleich mit den Kontopatterns zeigt, dass diese Matches korrekt sind. Doch

voluptat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper sus lobortis nisl ut aliquip ex ea commodo consequat.

Folgende Konten sind mit Stand Juni 2020 als Aktivierungskonten zu betrachten:

- **Alle Konten des Kontenkreises 90*, ohne 901234567**
- **Die Konten 50380* und 604002345**

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consec

Abbildung 1: Erster Ausschnitt aus einem Feinkonzept (Quelle: Jürgen Sieben)

```
SQL> select kdt_pattern, kdt_verdichtungskonto,
2         kdt_controlling_flg, kdt_aktivierungs_flg
3         from konto_decision_tab;
KDT_PATTERN KDT_VERDICHTUNGSKONTO KDT_CONTROLLING_FLG KDT_AKTIVIERUNGS_FLG
-----
503          509248576 FALSE          <null>
504          509134578 TRUE           <null>
50380        509951234 TRUE           TRUE
90           <null>                TRUE
901234567    <null>                FALSE
604002345    <null>                TRUE
<null>       509123456 FALSE          FALSE
7 Zeilen ausgewählt.
```

Listing 1: Rohdaten aus dem Feinkonzept

```
SQL> create or replace view konto_decision_verdichtet as
2 select v.kdt_pattern,
3        coalesce(v.kdt_verdichtungskonto, s.kdt_verdichtungskonto)
4        kdt_verdichtungskonto,
5        coalesce(v.kdt_controlling_flg, s.kdt_controlling_flg)
6        kdt_controlling_flg,
7        coalesce(v.kdt_aktivierungs_flg, s.kdt_aktivierungs_flg)
8        kdt_aktivierungs_flg
9        from konto_decision_tab v
10       cross join (
11         select *
12         from konto_decision_tab
13         where kdt_pattern is null) s;
View wurde erstellt.

SQL> select *
2   from konto_decision_verdichtet;
KDT_PATTERN KDT_VERDICHTUNGSKONTO KDT_CONTROLLING_FLG KDT_AKTIVIERUNGS_FLG
-----
503          509248576 FALSE          FALSE
504          509134578 TRUE           FALSE
50380        509951234 TRUE           TRUE
90           509123456 FALSE          TRUE
901234567    509123456 FALSE          FALSE
604002345    509123456 FALSE          TRUE
<null>       509123456 FALSE          FALSE
```

Listing 2: Verdichtung und Anreicherung der Standardwerte

```

SQL> select *
      2   from konto_nr;
KTO_NR      ERWARTETES_ERGEBNIS
-----
401234567 Standard
902345678 Match mit 90
901234567 Match mit 901234567
503501234 Match mit 503
503801234 Match mit 50380

```

Listing 3: Liste mit Testkontonummern

```

SQL> select k.kto_nr, d.kdt_pattern
      2   from konto_nr k
      3   join konto_decision_verdichtet d
      4     on kto_nr like kdt_pattern || '%'
      5   order by kto_nr, kdt_pattern nulls first;

```

```

KTO_NR      KONTO_PATTERN
-----
401234567
503501234
503501234 503
503801234
503801234 503
503801234 50380
901234567
901234567 90
901234567 901234567
902345678
902345678 90

11 Zeilen ausgewählt.

```

Listing 4: Mapping über einen Mustervergleich mit dem Kontopattern

```

SQL> select k.*, d.*,
      2      rank() over (
      3        partition by kto_nr
      4        order by coalesce(length(kdt_pattern), 0) desc) rang
      5   from konto_nr k
      6   join konto_decision_verdichtet d
      7     on kto_nr like kdt_pattern || '%'
      8   order by rang
      9   fetch first 1 row with ties;

```

```

KTO_NR      ERWARTETES_ERGEBNIS KDT_PATTE KDT_VERDI KDT_C KDT_A
-----
401234567 Standard                509123456 FALSE FALSE
503501234 Match mit 503          503      509248576 FALSE FALSE
503801234 Match mit 50380       50380    509951234 TRUE  TRUE
901234567 Match mit 901234567 901234567 509123456 FALSE FALSE
902345678 Match mit 90          90       509123456 FALSE TRUE

```

Listing 5: Vollständige Abfrage

ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Die Zahlungen auf Konten werden auf Verdichtungskonten geführt. Das jeweilige Verdichtungskonto ist folgender Tabelle (Stand 10/2020) zu entnehmen:

Kontenkreis	Verdichtungskonto	Mit Controllingbuchung
50380*	509951234	Ja
503*	509248576	Nein
504*	509134578	Ja
Standard	509123456	Nein

Ist hinterlegt, dass eine Controllingbuchung erforderlich ist, muss diese durch entsprechende Prüfung sichergestellt werden.

Abbildung 2: Zweiter Ausschnitt aus einem Feinkonzept (Quelle: Jürgen Sieben)

welcher dieser Matches ist der bestmögliche?

Diese Entscheidung hätten wir auch in einer `case`-Anweisung treffen müssen, dort jedoch durch eine feste Reihenfolge unserer Prüfungen, damit ein allgemeinerer Match einen spezielleren nicht überschreibt. In beiden Szenarien zeichnet sich ein besserer Match durch ein längeres Kontopattern aus: je länger das Pattern, umso spezieller die Übereinstimmung. Nun haben wir noch die Herausforderung, diese Zeile zu filtern.

Das ist eine Top-N-Analyse und wir verwenden hierfür die analytische `rank`-Funktion. Ich zeige in `Listing 5` eine Variante mit einer `row limiting`-Klausel, eine Inner View mit einem entsprechenden Filter wäre aber auch möglich. Beachten Sie bitte, dass die Klausel `with ties` verwendet werden muss, weil in unserem Fall mehrere Kontonummern existieren, die jeweils auf Platz 1 liegen müssen. Dies und die Partitionierung nach Kontonummer ist im Code später nicht mehr erforderlich, weil immer nur eine Kontonummer zur gleichen Zeit geprüft wird.

Nun ergibt auch die Spalte `ERWARTETES_ERGEBNIS` Sinn, denn Sie sehen, dass alle Entscheidungen korrekt getroffen werden. Übertragen Sie diese Abfrage in das Package, fällt auf, dass innerhalb des Codes überhaupt keine Fachlichkeit mehr existiert, da die Entscheidung vollständig von der `select`-Anweisung und damit letztlich von den Daten selbst getroffen wird.

Bewertung

Allein schon die Tatsache, dass kein neuer Code ausgeliefert werden muss, um eine neue Datensituation abzubilden, ist ein großes Plus. Viele Unternehmen dulden Skripte, um Daten von Tabellen zu pflegen, eher als das Neuausliefern von Code, weil es sich nur um Daten handle. Doch auch hier lohnt es sich, ein wenig weiter zu denken.

Wer hat eigentlich das Feinkonzept geschrieben? Die Fachseite. Angerufen haben Sie auch die Fachseite, als Sie wissen wollten, ob der Datenstand noch aktuell ist. Warum also nicht eine kleine Anwendungsseite erstellen, mittels derer die Fachseite die Entscheidungstabelle selbst pflegen kann? Vielleicht noch mit dem Komfort, eine Kontonummer in einen Dialog eingeben zu können und die Logik anzeigen zu lassen, welches Verdichtungskonto zugeordnet wird und ob Controlling- oder Aktivierungskonten erforderlich sind. So kann die Fachseite ihre Eingaben selbst kontrollieren und erhält damit auch direkt die Verantwortung über die korrekte Pflege. Es geht doch nichts über ein gepflegtes Rückdelegieren!

Entscheidungstabellen sind einerseits, wie bereits beschrieben, sehr häufig in der Anwendungsentwicklung möglich, andererseits aber auch sehr individuell. Dies bezieht sich auf den Aufbau der Tabelle, die alle relevanten Informationen enthält, um eine konkrete Entscheidung treffen zu können, und aber auch bezüglich der Abfrage, die, je nach Fragestel-

lung, auf unterschiedlichem Weg zu einer Entscheidung kommt. Dann kann es sein, dass lediglich die bestmögliche Option oder alle Treffer gewählt werden müssen, wie im vorliegenden Beispiel. Entscheidungstabellen werden, im Gegensatz zum normalen Datenmodell, nicht nach Normalisierungsregeln entwickelt, sondern eng an einer gegebenen Fragestellung. Natürlich können diese Tabellen Fremdschlüsselbeziehungen zu anderen Tabellen besitzen, etwa, wenn sie einen Status referenzieren, doch steht die Möglichkeit, eine konkrete Problemstellung beantworten zu können, eindeutig im Vordergrund.



Jürgen Sieben
j.sieben@condes.de



Warum weiß die Anwendung mehr über ihr Datenmodell als die Datenbank?

Stephan La Rocca, Pitts

Jeder hat in seiner Ausbildung gelernt, wie ein ERM mit Entitäten, Attributen und Beziehungen erstellt werden sollte. Viel spannender wird es aber, wenn man bei der Modernisierung von Legacy-Applikationen, bei der Wartungsübernahme einer nicht vertrauten Anwendung oder beim ersten Blick auf die Datenbank bei einem Merger feststellt, dass keine Dokumentation vorhanden, die letzte Person mit Know-how in Rente ist und selbst der SQL-Developer kein Modell erstellen kann, da schlichtweg keine Constraints vorhanden sind. Der Artikel beschreibt den Weg, aus den auf dem Datenmodell laufenden Applikationen alle Informationen zu ermitteln, um Entitäten benennen, Attribute verstehen und Beziehungen ableiten zu können.

Es wird eine etwas längere Reise, also würde ich empfehlen noch einen Kaffee zu holen und vielleicht noch etwas Süßes für den Kopf (*siehe Abbildung 1*).

Zunächst die Route in der Zusammenfassung: Wenn wir alle CRUD-Statements (Create – Read – Update – Delete) aus den Anwendungen extrahieren und parsen können sowie Anordnungen im UI (Master-Detail-Beziehungen, Wertelisten-Zuordnungen und mehr) berücksichtigen, können wir erkennen, welche Tabellen über welche Spalten verknüpft sind und daraus einen Graph erstellen, dessen Analyse via Oracle Property Graph uns die gewünschten Informationen ermitteln und aufbereiten kann. Aber eins nach dem anderen.

Ermittlung der CRUD-Statements

Als Quelle für die CRUD-Statements stehen zunächst zwei verschiedene Bereiche zur Verfügung. Zum einen die Implementierung der Software sobald wir diese im Quelltext verfügbar haben. Das können alle DB-Packages, Prozeduren, Funktionen, Views, Typen über die Data Dictionary View DBA_SOURCES sein. Aber auch Applikationen wie Oracle Forms (Blocks Source, Record Groups, Statements in Programm Unit und Triggern), Reports (Queries, Statements in Triggern), APEX (Region Sources, List of Values, SQL-Conditions), ADF (Definition von EO und VO), externe SQL-Scripts, ProC und vie-

le weitere. Die zweite mögliche Quelle sind alle Statements, die von der Oracle-Datenbank geparkt und zum Beispiel in V\$SQL zur Verfügung stehen. Damit können auch Applikationen, deren Quellcode nicht verfügbar, ist berücksichtigt werden. Ziel ist es, möglichst viele SQL-, DML-Statements der Analyse zur Verfügung zu stellen.

Parsen der CRUD-Statements

Für jedes einzelne Statement gilt es nun zu ermitteln, welche Tabellen involviert sind und welche Spalten für die Joins genutzt werden. Da wir uns hier nicht auf lexikalische Operationen bei der Analyse des Statements verlassen wollen, kommt ein Parser zum Einsatz, den wir auf die Grammatik der Sprachen konfigurieren können: ANTLR (<https://www.antlr.org/>).

Kurz erläutert basiert ANTLR auf einem Lexer und einer Grammar. Der Lexer untersucht das Statement Zeichen für Zeichen und versucht vordefinierte Schlüsselwörter zu erkennen, zum Beispiel: (SELECT, UPDATE, etc.) Die Grammar ordnet nun einem Schlüsselwort eine Rolle in der Syntax der Sprache zu. Somit wird definiert, dass etwa nach dem Schlüsselwort SELECT entweder ein DISTINCT oder ein ALL folgen kann und dann ein * oder eine Liste von Spalten.

Auf der ANTLR-Homepage gibt es bereits fertige Lexer und Grammar für beispielsweise SQLite. Es ist aber sehr einfach und „nur“ mit Fleißarbeit verbunden,

Schlüsselwörter und Syntax selbst innerhalb der ANTLR Files zu ergänzen.

Einmal erstellt, lassen sich aus der Grammar-Datei automatisiert Java-Klassen erstellen, die eine Analyse eines gegebenen Strings ermöglichen. Mit einem simplen Befehl (*siehe Listing 1*) werden 4 Javaklassen (Parser, Lexer, Listener und BaseListener) erstellt, die in unserem Fall ein simples SQL-Statement in die einzelnen Bestandteile auseinandernehmen können. Java ist die Standard-Programmiersprache, die ANTLR generiert, darüber hinaus sind aber auch etwa C#, C++, JavaScript, Python und andere möglich. Für die weitere Verwendung innerhalb einer automatisierten Analyse innerhalb der Datenbank bietet sich allerdings Java auch an. Um ein Gespür zu bekommen, wie die generierten Klassen verwendet werden können, zeigt folgendes Beispiel einen Unit Test (*siehe Listing 2*).

Persistieren der Ergebnisse

Aus diesen Klassen soll es im nächsten Schritt nicht nur möglich sein, die Statements „zu verstehen“, sondern auch die gewonnenen Informationen zu speichern. Das notwendige Datenmodell obliegt den eigenen Anforderungen. Wir haben uns dazu entschieden, Tabellen, Spalten und die Komponenten der Where-Bedingung jeweils in eine eigene Tabelle zu speichern, die mit einem Foreign-Key auf den Master-Record, also das eigentliche Statement, verweisen. So entstehen zum

```
Antlr4 Expr.g4
```

Listing 1: Generierung von Java-Klassen auf Basis einer eigenen „Expr“-Grammar

```
@Test
public void testDbLink() throws ParseException {
    ISqlSelectStatement stmt = (ISqlSelectStatement)ParseFactory.parse("select column_name from table_name@db.link.name");
    IQueryBlock qb = (IQueryBlock)stmt.getSubquery();
    ITableReference tr = (ITableReference)qb.getQueryBlockItems().get(0);
    ITableSpec spec = (ITableSpec)tr.getQueryTableExpression();
    SqlName tableName = spec.getSpec();
    String dbLink = tableName.getDbLink();
    assertEquals("Wrong table name!", "table_name", tableName.getFullName());
    assertEquals("Wrong DB Link!", "db.link.name", dbLink);
}
```

Listing 2: Test-Methode für DB-Link

Beispiel Tabellen wie STR_FROM (für die Namen der Tabellen), STR_LOGIC (für die Join-Conditions), und andere.

Um die Erstellungshoheit der Java-Klassen ANTLR4 zu überlassen (schließlich sind auch mal Änderungen und Erweiterungen an der Grammar notwendig), haben wir uns entschieden, für das Persistieren der Daten die notwendigen Java-Klassen zu „extenden“. Um die Tabellen eines Statements zu speichern, sieht die Klasse zum Beispiel wie in *Listing 3* aus.

Dabei ist der BatchManager dann eine Klasse, die das Statement vorbereitet und die Variablen ersetzt.

Integration in PL/SQL

Selbstverständlich soll als Oracle Developer diese Funktionalität später innerhalb der Datenbank-Packages zur Verfügung stehen, um die Ergebnisse (kleiner Spoiler) als APEX-Anwendung zu visualisieren. Dafür stellt die Oracle-Datenbank die eigene JVM zur Verfügung und die Möglichkeit via loadjava eine Java-Klasse in die Datenbank zu laden. Einmal geladen, kann dann sehr schnell ein PL/SQL-Wrapper für den Aufruf der Klasse erstellt werden (siehe *Listing 4*).

Wo verstecken sich noch Beziehungen?

Bis jetzt haben wir uns darauf konzentriert, SQL- und DML-Statements, die irgendwo im Source Code implementiert sind, über diesen Weg zu parsen, in ihre Bestandteile zu zerlegen und diese einzeln zu speichern. Eine Applikation bietet allerdings noch eine Vielzahl weiterer interessanter Quellen gerade für die

Verbindungen zwischen Tabellen. Diese müssen sicherlich anders maschinell erhoben werden und würden den Rahmen dieses Artikels sprengen. Daher an dieser Stelle nur die Anregung, auch den Blick auf folgende Beziehungen zu werfen:

- Master-Detail-Beziehungen zwischen zwei Blöcken, die nur in Oracle Forms erstellt wurden,
- Zuordnungen einer List-Of-Value für die Eingabe-Spalte einer anderen Tabelle,
- Befüllung von Parametern, die an anderen Stellen als Teil der Where-Bedingung genutzt werden,
- Select-Statements in Oracle Reports für das Befüllen von berechneten Feldern oder bedingter Formatierung und
- Shared Components in APEX für Page Items.

In Abhängigkeit der eingesetzten Tools auf dem Datenmodell lässt sich diese Liste beliebig erweitern. So könnte schon allein das Anzeigen zweier Tabellen in einem UI für eine „lose“ Verbindung dieser beiden Tabellen „sprechen.“

Finales Datenmodell

Damit wir für den Anwender abschließend die beste Visualisierung und eine optimale User Experience erzielen können, werden abschließend alle Beziehungen, die wir zwischen den Tabellen und ihren Spalten ermitteln konnten, final einem Key-Value-Pair-Datenmodell mit der Unterscheidung nach Knoten und Kanten übergeben. Daraus ergeben sich zwei entscheidende Vorteile:

1. Vorhandene Graph-Algorithmen können diese Datenstruktur nutzen, um zum Beispiel eine Community Detection zu ermitteln. Eine Community Detection auf einem Datenmodell, also die Ermittlung von „eng zusammengehörigen Tabellen“, erleichtert eine Unterteilung von mehreren tausenden Tabellen in einem Datenmodell in eine eher „Bounded Context“ orientierte Unterteilung. Somit gelingt es leichter, Tabellen zu ermitteln, die „semantisch“ zusammengehören.
2. Datenmodelle mit Knoten und Kanten lassen sich sehr leicht optisch als Beziehungsgraph darstellen. Dazu später mehr bei der Integration in APEX.

Da Knoten und Kanten als Key-Value Pair aufgebaut sind, können wir den Knoten, die die Tabellen repräsentieren, weitere hilfreiche Attribute mitgeben, die sowohl für die Community-Detection als auch die Visualisierung relevant sind. Wir haben uns entschlossen, die Anzahl der unterschiedlichen CRUD-Statements sowie obligatorisch Name, Schema und ID der Tabelle zu speichern. Bei den Kanten, die die Beziehungen der Tabellen widerspiegeln, sind die verbindenden Spalten der beiden Tabellen sowie die Häufigkeit dieser Verknüpfung in der gesamten Applikation festgehalten.

Nutzen des Oracle Property Graph

Da wir nun schon eine Oracle-Datenbank am Start haben, wäre es fahrlässig, auf die Verwendung des Oracle Property Graph für die weiteren Schritte zu verzichten. Der Oracle Property Graph ermöglicht beide zuvor angesprochenen

```
public class DbFrom extends From {
    public DbFrom(String string, String string1) {
        super(string, string1);
    }

    private static final String insertText = "INSERT INTO STR_FROM " +
        "(ID_OBJ, SQL_ID, FROM_SEQ, FROM_NAME, FROM_ALIAS, P_OWNER) " +
        "VALUES (?, ?, ?, ?, ?, pitsscon_user)";

    private static BatchManager batcher = new BatchManager(insertText);
    ...
}
```

Listing 3: Extends der Java-Klassen, um die Ergebnisse zu speichern

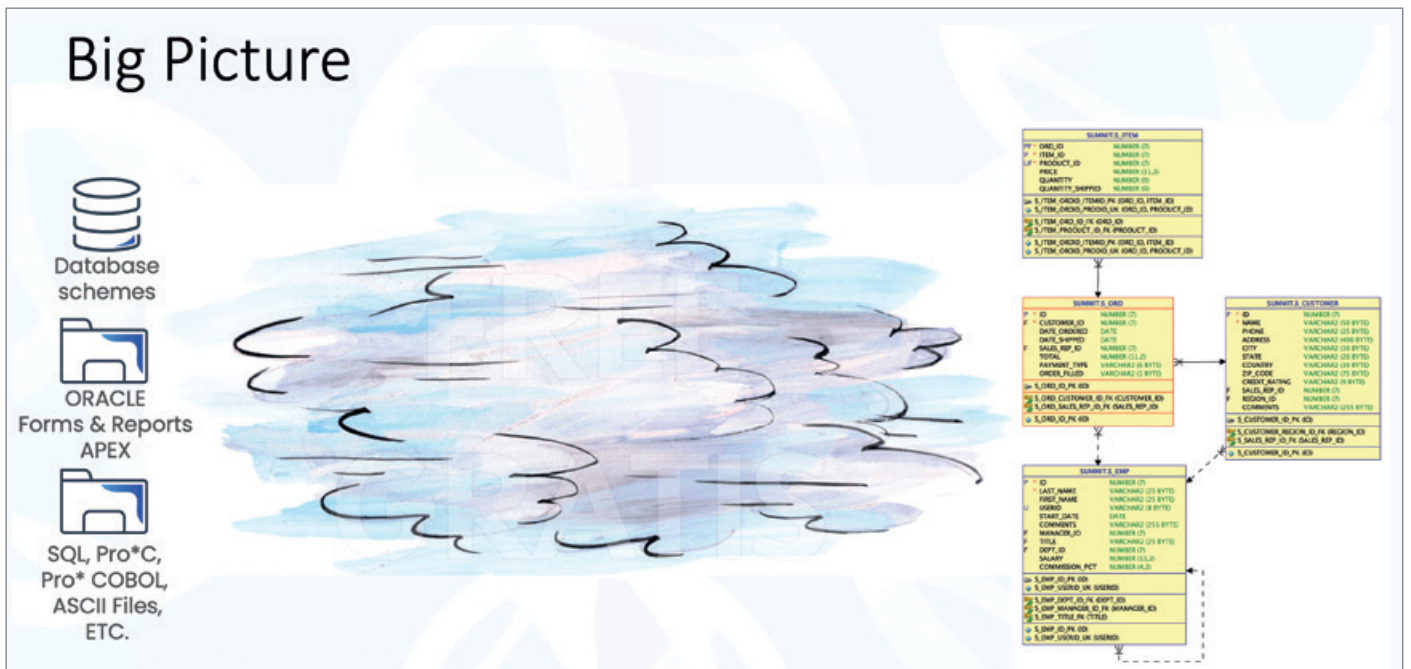


Abbildung 1: Wie kann es gelingen aus einer laufenden Applikation möglichst viel Informationen über das „gedachte“ Datenmodell zu identifizieren (Quelle: Stephan La Rocca)

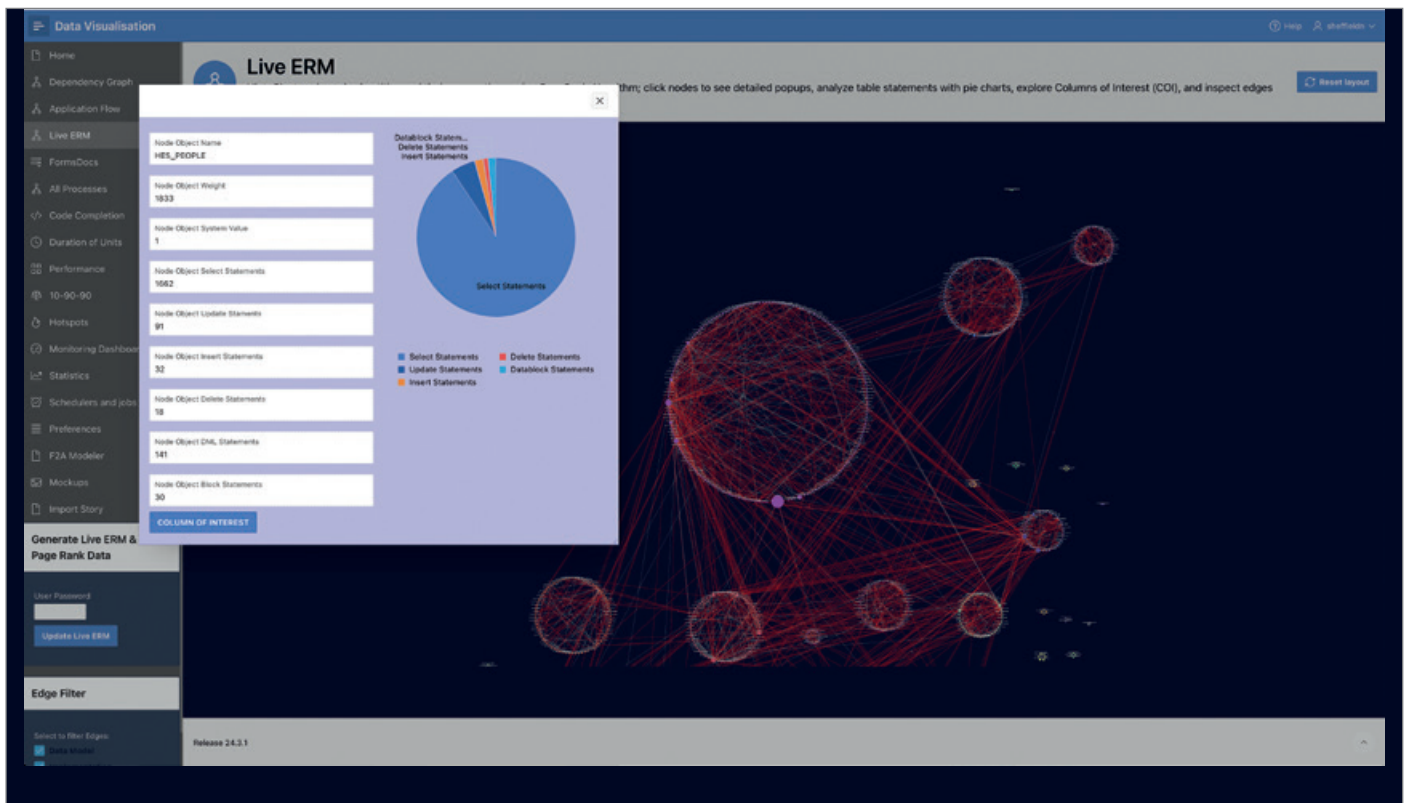


Abbildung 2: ERM mit allen Beziehungen aus der Applikation und Clustering für ein Datenmodell mit mehr als 1200 Tabellen (Quelle: Stephan La Rocca)

Punkte, das heißt die Verwendung von Graph-Algorithmen auf dem Datenmodell sowie auch eine ansprechende Visualisierung der Ergebnisse. Wer sich selbst daran probieren möchte, findet unter folgendem Link eine Demo innerhalb eines Oracle Live Lab:

<https://oracle-livelabs.github.io/partner-solutions/software-analysis-with-pitsscon/workshops/tenancy/index.html>

In der Oracle-Implementierung für den Property Graph sind über 80 Algorithmen verfügbar, die Ranking, Commu-

nity Detection, Pfadfindung und vieles mehr ermöglichen. Allein für die Aufgabe „Community Detection“ kann einer der folgenden Algorithmen verwendet werden: Conductance Minimization (Soman and Narang Algorithm), Infomap, Label Propagation, Louvain, LouvainDi-

- **Geschäftsobjekte** sind vermutlich Tabellen mit vielen CRUD-Statements und vielen Spalten, die im UI sichtbar sind.

Eine solche Eingliederung der Tabellen wird die Ermittlung der Communities beeinflussen und bei der Bestimmung eines Bounded Context helfen.

Rein für die Usability der Darstellung arbeiten wir daran, aus dem gezeigten LiveERM (siehe *Abbildung 2*) wieder bis zurück in die Codezeile navigieren zu können. Die Ursache dafür war, dass das CRUD-Statement erkannt und die Verknüpfung erstellt wurde. Auf diesem Weg wollen wir dem Analysten die Möglichkeit geben, Annotationen für Tabellen oder den Code zu erfassen, um nachfolgenden Wartungsverantwortlichen das Leben etwas leichter zu machen.

Warum eigentlich nochmal?

Am Ende des langen Weges blicken wir noch einmal zurück und rufen uns die Gründe für diese vielen Schritte ins Gedächtnis. Am Anfang steht in den meisten Fällen ein etabliertes Datenmodell, welches über Jahre erweitert, ergänzt oder zusammengeführt wurde. Mit Hilfe

der Art und Weise, wie dieses Datenmodell durch die verschiedenen Anwendungen Verwendung findet, kann es gelingen, auch als Außenstehender ein gutes Verständnis für die Zusammenhänge innerhalb des Datenmodells zu erlangen (siehe *Abbildung 3*).

Dies ist notwendig, sobald massive Änderungen wie beispielsweise Refakturationen, Merger, Teil-Ausgliederungen in eine Standard-Applikation oder Modernisierungen bevorstehen. Aber auch simplere Aufgaben aus dem Bereich Data Governance oder der Zuordnung für einen Bounded Context profitieren von dieser Betrachtungsweise.

Für uns war dieser Weg eine Teiletappe, durch die wir aus einer beliebigen Individual-Applikation ein objektzentrisches Prozess-Discovery durchführen können, ohne eine Vielzahl von zeitaufwändigen und fehleranfälligen Interviews mit Personen führen zu müssen, die das notwendige Context-Wissen hätten haben können. Dazu müssen als erstes die Objekte der Applikation erkannt werden.

Dazu vielleicht mehr in der nächsten Ausgabe.

Sollten Sie Lust haben, bei einem virtuellen Kaffee gemeinsam über diesen Weg oder die gesamte Reise zu diskutieren, freue ich mich über ihren Kontakt.

Über den Autor

Stephan La Rocca (Director Business Development, PITSS GmbH) ist Modernisierungsexperte, Speaker und Autor einer Vielzahl von Beiträgen im Bereich Software-Entwicklung.



Stephan La Rocca
slarocca@pitss.com

Oracle Datenbanken Monthly News

Auf dem deutschsprachigen Oracle-Blog ist die August-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-50/>

SUPER-SAVER

Bis 31.10.2025

Heide Park Soltau

APEX connect
18. - 20. Mai 2026



DOAG 2026
Datenbank
mit Cloud Infrastructure
18. - 19. Mai 2026





Modernes Suchen in APEX 24.2

Florian Graßhoff, Oracle Deutschland

In der heutigen datengetriebenen Welt ist die Fähigkeit, effektiv in großen Datenmengen zu suchen, entscheidend. Eine Fülle an Daten ist ohne leistungsfähige Suchfunktionalitäten oft wertlos. Suchfunktionen können jedoch kompliziert und aufwendig in der Implementierung sein. In diesem Artikel wird gezeigt, wie Oracle APEX 24.2 diese Herausforderungen mit dem Application Search Feature in Kombination mit Oracle Ubiquitous Database Search und Oracle AI Vector Search vereinfacht.

Für diesen Artikel wurden synchronisierte Daten, bezogen von der TMDB (The Movie Database) API, verwendet. Für die einmalige Synchronisierung wurde eine entsprechende REST Data Source angelegt und die REST Synchronization genutzt, um mit einem Klick die Daten in eine lokale Tabelle zu übertragen. Zusätzlich wurden mittels der von der TMDB API bereitgestellten Poster-URL die Bilder mit der apex_web_service API in eine BLOB-Spalte übertragen. Somit lassen sich die Bilder in den Suchergebnissen schneller anzeigen und sie können auch für eine Oracle AI Vector Search herangezogen werden.

Suchen leicht gemacht mit dem Application Search Feature

Mit dem APEX Application Search Feature können verschiedene Datenquellen in einer einzigen Suche abgefragt und in einem einheitlichen Suchergebnis angezeigt werden. Dabei werden in der APEX Version 24.2 folgende Datenquellen angeboten:

- Table
- SQL Query
- Property Graph
- REST Enabled SQL
- REST Source
- Oracle Duality View

- JSON Source
- Oracle AI Vector Search
- Oracle Text
- Oracle Ubiquitous Database Search
- List

Das Application Search Feature besteht aus drei Kernkomponenten:

1. Search Configuration: Einstellung, woher die Daten kommen, welche Informationen für die Suche verwendet werden sollen und wie das Suchergebnis aussehen soll
2. Search Region: Diverse Einstellungsmöglichkeiten, von einer einheitlichen Sortierung für die Suchergebnisse der zugewiesenen Search Configurations bis hin zur Konfigurierung der Pagination
3. Search API: Option, mit der API die Search Configuration auch ohne Search Region für Suchen zu verwenden

In diesem Artikel soll der Fokus zunächst auf der Volltextsuche und im weiteren Verlauf auf der Vektorsuche liegen. Oracle Text gibt es bereits seit dem letzten Jahrhundert und es ist ein sehr mächtiges Datenbankfeature. Die spalten- und tabellenübergreifende Indizierung mit all ihren Einstellungsmöglichkeiten kann allerdings für den schnellen Einstieg abschreckend wirken. Mit der Oracle-Datenbank-Version 23ai ist die Verwen-

dung durch das neue Feature Oracle Ubiquitous Database Search stark vereinfacht worden.

Zunächst wird ein Index erstellt und diesem eine Tabelle mit all ihren indizierbaren Spalten hinzugefügt (siehe Listing 1).

Mit der Prozedur add_source kann der Index um jede weitere Tabelle oder View erweitert werden. In der Tabelle enthaltene Dokumente, zum Beispiel im PDF-Format, werden dabei ebenfalls indiziert. In APEX werden nun über die Shared Components die Search Configurations geöffnet, wo eine Search Configuration vom Typ Oracle Ubiquitous Search erstellt wird (siehe Abbildung 1 bis 3).

Über den Create Page Wizard wird die Applikation um eine Search Page erweitert und die zuvor erstellte Search Configuration ausgewählt. Im Anschluss kann die erste Suche erfolgen.

In Abbildung 4 ist der grundsätzliche Aufbau einer Standard Search Page ersichtlich. Die Seite beinhaltet ein Suchfeld und einen Bereich (Search Region) für das Suchergebnis, das in diesem Beispiel das Cover des Films sowie den Titel (klickbar – Verlinkung zu TMDB), eine Beschreibung, die Bewertung, das Erscheinungsdatum und den Link zum Cover anzeigt. Welche Information an welcher Stelle angezeigt wird und wohin der Link führen soll, ist mit der Search Configuration einstellbar. Sollten die deklarativen Konfigurationsoptionen nicht ausreichen, kann auch

```

begin
  dbms_search.create_index( 'TMDB_SEARCH' );
  dbms_search.add_source ( 'TMDB_SEARCH', 'TMDB_MOVIES');
end;
/

```

Listing 1: Verwendung von DBMS_SEARCH

```

select *
  from table( apex_search.search(
    p_search_static_ids => apex_t_varchar2( 'movies' ),
    p_search_expression => :P7_SEARCH ) );

```

Listing 2: APEX_SEARCH.search in SQL

Create Search Configuration

Detail

A **Search Configuration** contains information about the searchable *Data Source*, the functionality used for search, as well as display properties.

* Name:

* Search Type:

- Standard
- Oracle Vector Search
- Oracle Text
- Oracle Ubiquitous Search
- List

Cancel Next >

Abbildung 1: Search Configuration Wizard – Search Type (Quelle: Florian Großhoff)

Create Search Configuration

Source

* Search Index Name:

* Search Source Name:

< Cancel Next >

Abbildung 2: Search Configuration Wizard – Source (Quelle: Florian Großhoff)

ein Default Result Row Template definiert werden. Dazu wird per HTML-Code ein eigenes Design gestaltet und die Informationen aus dem Suchergebnis werden eingebettet. Eine weitere Möglichkeit besteht darin, die dritte Kernkomponente zu nutzen: die Search API. Diese kann an jeder Stelle eingesetzt werden, an der SQL oder PL/SQL zum Einsatz kommt, beispielsweise für die Datenquelle einer Cards Region (siehe Listing 2 und Abbildung 6).

In der Abbildung 5 könnte Ihnen auch aufgefallen sein, dass der Suchbegriff „forrest gumb“ nicht dem Filmtitel „Forrest Gump“ entspricht. Hier kommt der zuvor definierte Index zum Einsatz (siehe Listing 1), was die Ungenauigkeitssuche ermöglicht. Mit einer Suchanfrage wird automatisch eine entsprechende Oracle Text Query Syntax verwendet. Über den Parameter Text Query Function in der Search Configuration können aber auch Experteneinstellungen vorgenommen werden.

Zudem ist in der App Gallery auch die App „Sample Application Search“ zu finden, die verschiedene Anwendungsfälle und Konfigurationsmöglichkeiten beschreibt.

Oracle AI Vector Search

Wozu eine weitere Suchtechnologie? Braucht es Oracle AI Vector Search wirklich, wenn es doch Oracle Text und Ubiquitous Database Search bereits gibt?

Die Antwort liegt im ‚Verstehen‘ von Inhalten. Oracle Text sucht nach exakt passenden Begriffen. Wer etwa „animal“ eingibt, bekommt nur Treffer, in denen dieses Wort explizit vorkommt. Oracle AI Vector Search hingegen erkennt Bedeutungen: Eine Suche nach „Animal“ findet auch Titel wie „A Dog's Journey“, „Shark Tale“ oder „Peter Rabbit“ (siehe Abbildung 7) – selbst wenn der Suchbegriff nicht im indizierten Titel oder der Beschreibung enthalten ist. Ubiquitous Database Search wiederum bietet eine einheitliche Suche über verschiedene Datenformate, jedoch ohne semantisches Verständnis. Oracle AI Vector Search erweitert das Suchspektrum also entscheidend, von rein textbasierter Übereinstimmung hin zu echter inhaltlicher Relevanz.

Analog zu einer Search Configuration vom Typ Oracle Ubiquitous Search wird eine Search Configuration vom Typ

Oracle AI Vector Search benötigt. Voraussetzungen dafür sind ein Vector Provider (dazu kommen wir gleich) und eine Tabelle (oder View), die eine Spalte vom Datentyp Vector enthält. Eine Vector-Spalte zu erstellen, ist mit einer einfachen ‚ALTER TABLE DDL‘-Anweisung möglich. Die Daten (Vektoren) zu erzeugen, kann hingegen eine komplexere Angelegenheit sein. In diesem Artikel stehen die Vector Provider im Fokus, die den Nutzenden in APEX 24.2 zur Verfügung stehen, denn diese sind eigens für diesen Zweck zu den Workspace Utilities hinzugefügt worden. Kurz auf den Punkt gebracht: Mit einem Vector Provider wird definiert, welches Embedding Model eingesetzt wird, um für einen Suchbegriff das Vector Embedding zu erhalten, denn dieses ist für eine Vektorsuche notwendig. Ein Vector Provider kann auf drei verschiedene Arten konfiguriert werden:

1. Database ONNX Model: Es ist möglich, ein ONNX Model mit dem DBMS_VECTOR Package in die Datenbank zu laden. Das kann für die Performance der Vektor-Generierung vorteilhaft sein und außerdem müssen die Daten nicht an eine andere Stelle außerhalb der Datenbank übertragen werden.
2. Generative AI Service: Verschiedene AI-Provider (OCI, OpenAI, Cohere und viele mehr) bieten die Generierung von Vektoren an. Der Vector Provider übernimmt die Kommunikation mit dem AI Service.
3. Custom PL/SQL: Eine weitere Option ist die Erstellung und Nutzung einer eigenen Datenbankfunktion, die einen String (VARCHAR2/CLOB) erwartet und einen Vektor zurückgibt.

Ein Vector Provider kann explizit über die APEX_AI API und implizit über eine Search Configuration (siehe Abbildung 8) vom Typ Oracle AI Vector Search verwendet werden. Der Search Configuration Wizard bietet die Auswahl einer Vector Spalte an (siehe Abbildung 9) – also worin wird gesucht – und auch die Auswahl eines Vector Providers – also wie der Suchbegriff in einen Vektor umgewandelt werden soll.

Die Bearbeitungsseite der Search Configuration bietet weitere typische Vector-Search-Konfigurationsmöglichkeiten, zum Beispiel die Maximum Vector Distance, den Search Type und die Distance

Metric. Für den Search Type Approximate ist ein bestehender Vector-Index die Voraussetzung. Der Vector Provider kann über die API auch explizit genutzt werden, um für einen String (VARCHAR2 oder CLOB) das Vektor-Embedding zu generieren, dafür ist kein Application Context notwendig. Bei der Nutzung eines ONNX Models oder einer Custom-PL/SQL-Funktion ist der Vorteil in Bezug auf den Programmieraufwand gering. Bei der Verwendung eines Generative AI Services hingegen übernimmt APEX die Kommunikation mit dem AI-Provider, was eine deutliche Erleichterung im Sinne des Low-Code-Ansatzes bedeuten kann. Die Verwendung der API, um eine neue Vector-Spalte initial mit Daten zu befüllen, kann so einfach sein (siehe Listing 3).

Je nach der Datenmenge und Auslastung des Systems kann diese Operation viel Zeit beanspruchen. Eine parallele Verarbeitung in Chunks ist mit wenig PL/SQL-Code mittels des Packages DBMS_PARALLEL möglich und per On-Demand-Automation in APEX kann der Prozess zusätzlich mit wenigen Klicks in den Hintergrund geschickt werden.

Kombination von Oracle Text und Oracle AI Vector Search

Die parallele Nutzung von Oracle Text und Oracle AI Vector Search eröffnet ein neues Qualitätsniveau für Suchfunktionen in APEX-Anwendungen – vorausgesetzt, beide Technologien werden sinnvoll miteinander kombiniert. Jede von ihnen bringt Stärken und Schwächen mit, die sich gegenseitig ausgleichen können:

- **Oracle Text** liefert exakte Keyword-Treffer, ist aber empfindlich gegenüber Formulierungsabweichungen.
- **Vector Search** zeigt semantisch verwandte Inhalte, riskiert jedoch irrelevante oder zu breite Ergebnisse.

Eine Möglichkeit, beide Verfahren zu verbinden, besteht darin, in einer Oracle Text Search Configuration die Order By Clause für die Sortierung nach der Vector Distance zu nutzen. Somit kann die Oracle Text Query Relaxation so gestaltet werden, dass sie sinnvolle, aber breit gefächerte Ergebnisse zurückliefert. Durch die Sortierung nach der Vector Distance

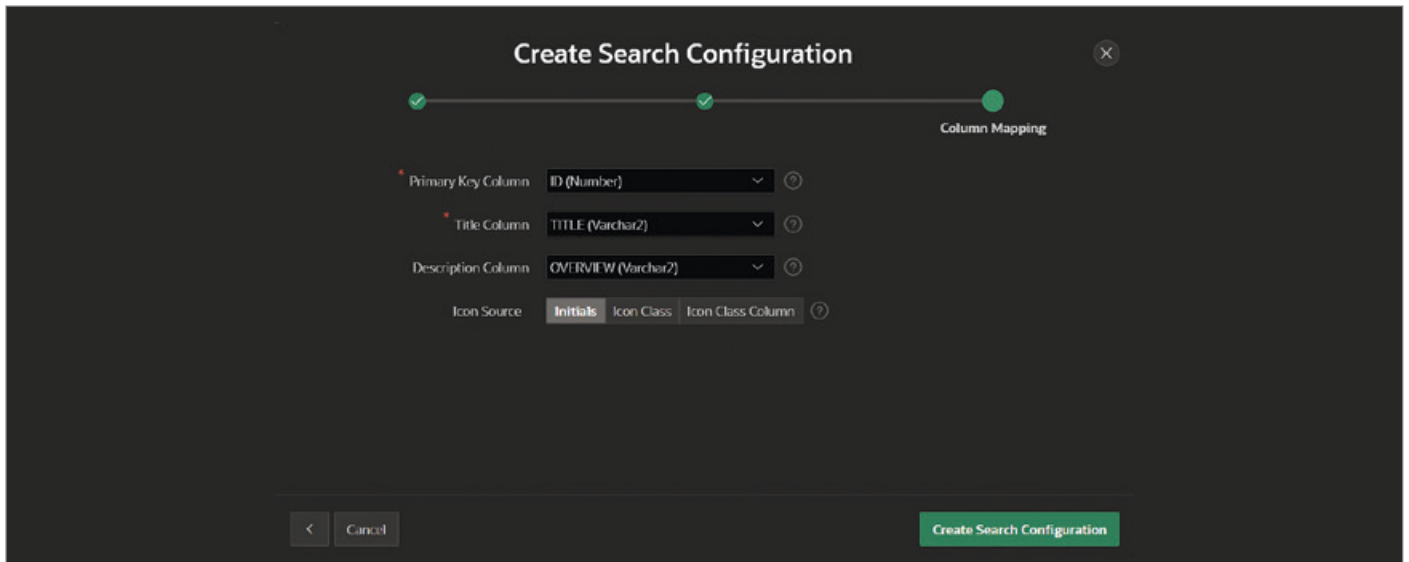


Abbildung 3: Search Configuration Wizard – Column Mapping (Quelle: Florian Graßhoff)

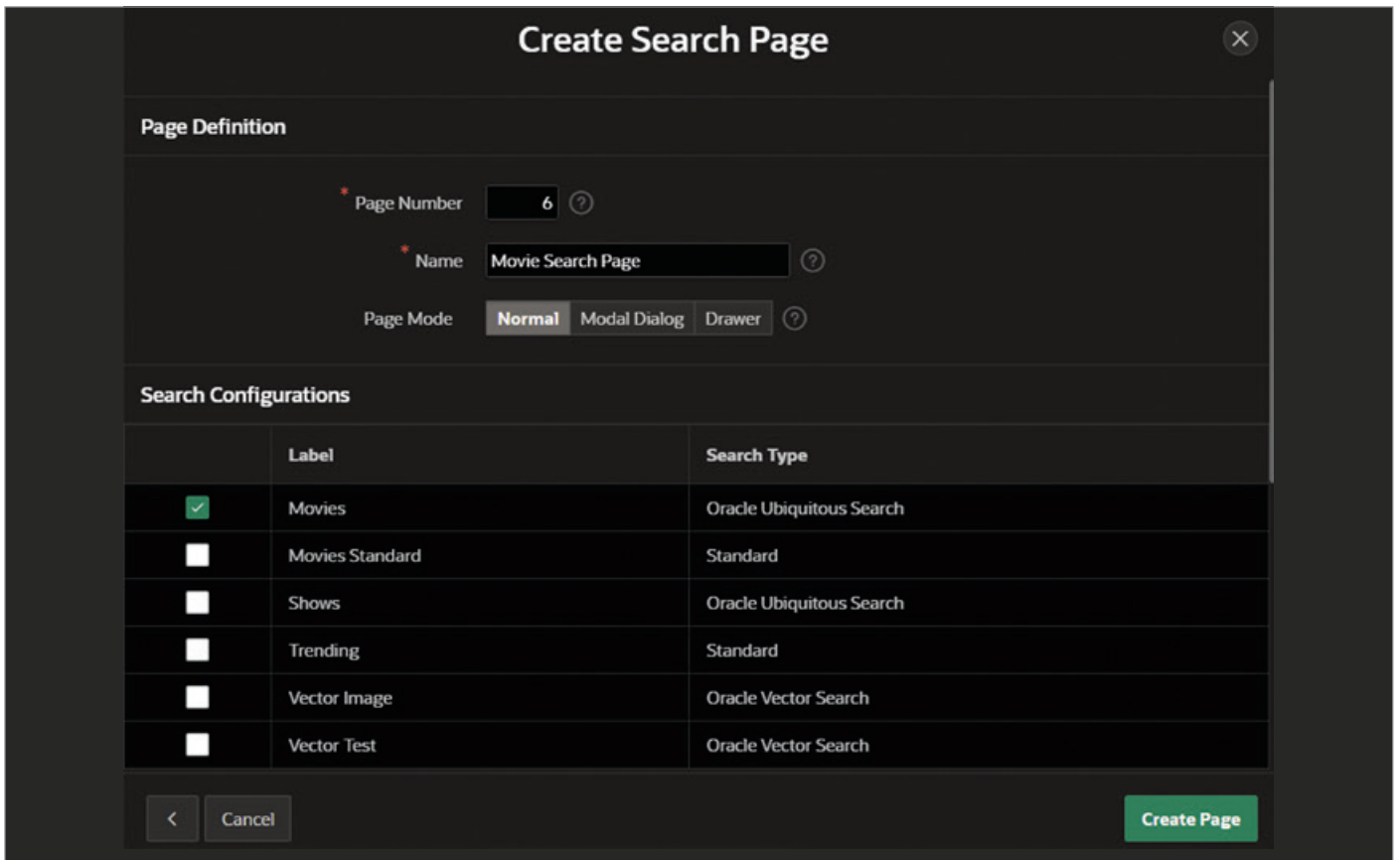


Abbildung 4: Aufbau einer Standard Search Page (Quelle: Florian Graßhoff)

werden dann die relevantesten Ergebnisse zuerst angezeigt.

Ein weiterer Ansatz ist das Vermengen beider Suchergebnisse, beispielsweise mit dem Mengenoperator UNION. Für die Sortierung könnte ein eigenes Ranking aus dem Oracle Text Score und der Vector Distance definiert werden, sodass die

Oracle-Text-Ergebnisse eine höhere Gewichtung erhalten.

Die Oracle-Datenbank bietet aber mit dem Hybrid Vector Search Index auch eine spezielle Lösung an (siehe Listing 4). Der Oracle Hybrid Vector Index kombiniert die Stärken der traditionellen Volltextsuche mit den Möglichkeiten der KI-

gestützten Vektorsuche. Diese hybride Indexierung ermöglicht es, sowohl exakte Schlüsselwortsuchen als auch semantische Ähnlichkeitssuchen innerhalb einer einzigen Abfrage effizient durchzuführen.

Am einfachsten kann der Oracle Hybrid Vector Index in Oracle APEX genutzt werden, indem man die searchpipeline-

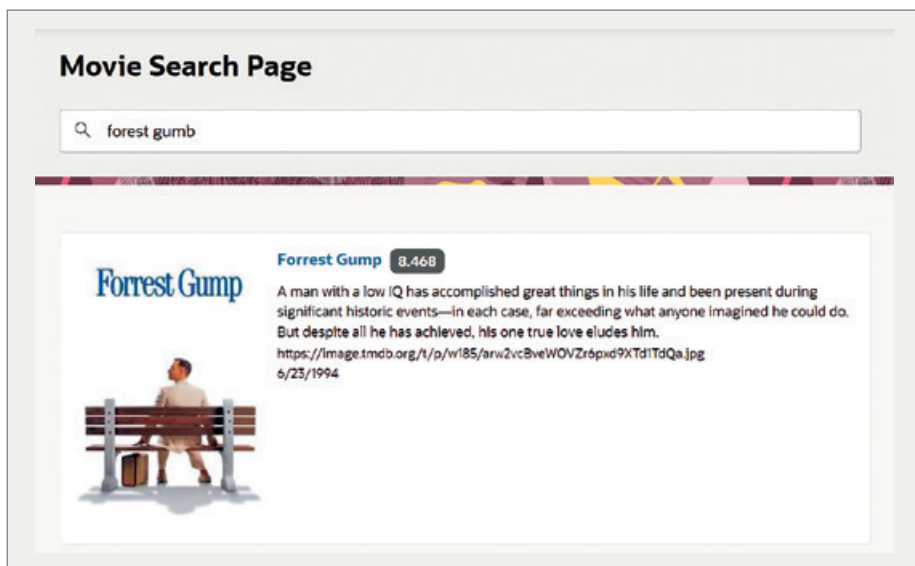


Abbildung 5: Search for „forest gumb“ (Quelle: Florian Graßhoff)

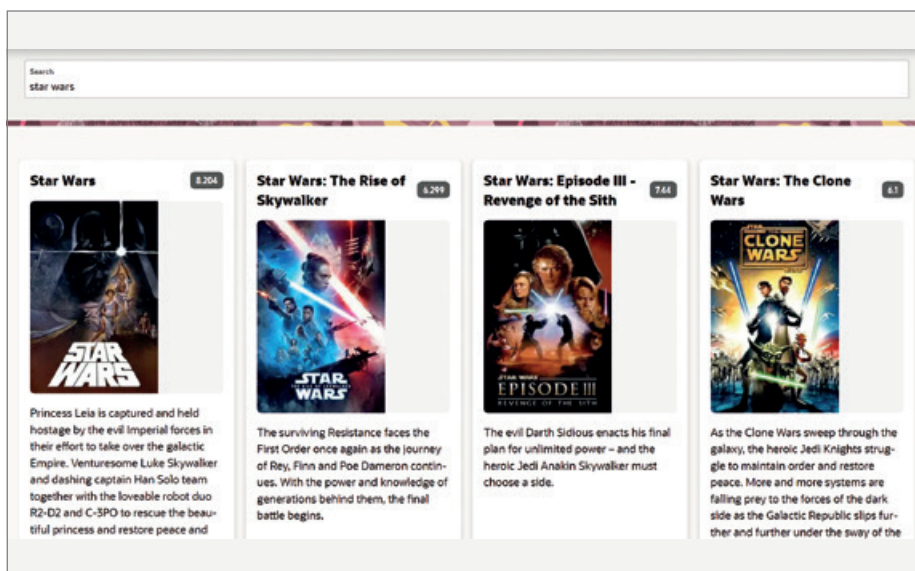


Abbildung 6: Cards Region – Search for „Star Wars“ (Quelle: Florian Graßhoff)

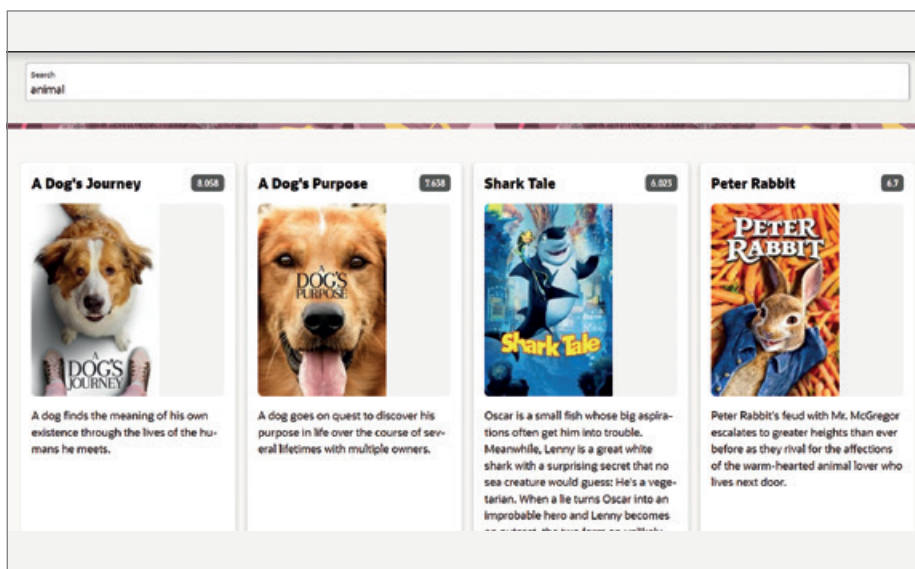


Abbildung 7: Cards Region – Search for „animal“ (Quelle: Florian Graßhoff)

Funktion verwendet, die das Ergebnis der Suche als Tabelle zurückliefert, anstatt als JSON-Konstrukt, wie es bei der search-Funktion der Fall ist (siehe Listing 5).

Das SQL-Statement kann wiederum in allen APEX-Komponenten verwendet werden, die SQL Querys als Datenquelle unterstützen. Es gibt zahlreiche Möglichkeiten, die hybride Suche anzupassen, zum Beispiel, ob die Oracle-TEXT-Ergebnisse höher gewichtet werden und ob hier ein anderer Suchbegriff verwendet werden soll. Der Oracle AI Vector Search User's Guide gibt dazu zahlreiche Informationen.

Suche nach Bildern und mit Bildern (Image Search)

Ein weiterer klassischer Anwendungsfall ist die textuelle Suche nach Bildern oder auch die Suche nach ähnlichen Bildern. Dafür werden Multimodal Embeddings eingesetzt. Die Generierung mit CLIP wird auch in der Dokumentation von Oracle beschrieben: „Generate Multimodal Embeddings Using CLIP“. Im Endeffekt wird für die Bildersuche ein Model benötigt, das für die Vektorisierung der Bilder in der Datenbank genutzt wird – oder für die Image Search – und zusätzlich ein Model, das einen Suchtext auf die gleiche Weise vektorisieren kann. Über die Funktion `apex_ai.get_vector_embeddings` ist es in der Version APEX 24.2 nicht möglich, das Vektor Embedding für ein Bild zu erzeugen. Bei einem ONNX-Model kann die interne Datenbankfunktion `vector_embedding` verwendet werden, da diese auch die Übergabe vom Datentyp BLOB erlaubt. Der Oracle APEX Blog Post „Implementing Image and Text Semantic Search in Oracle APEX Using ONNX Pipeline Models and Oracle Database 23ai“ beschreibt das aktuell mögliche Vorgehen mit einer Schritt-für-Schritt-Anleitung. Eine rein deklarative Unterstützung für eine Bildersuche ist aktuell nicht implementiert.

Mit der APEX App Gallery der Version 24.2 wurde auch die neue App „Sample Vector Search“ hinzugefügt. Sie demonstriert anschaulich die Unterschiede zu Oracle Text und Oracle AI Vector Search und die Kombination der beiden Technologien in APEX. Außerdem gibt es ein paar Anwendungsbeispiele für Vektorsuchen.

Abbildung 8: Search Configuration Wizard Vector – Vector Source (Quelle: Florian Graßhoff)

Abbildung 9: Search Configuration Wizard Vector – Vector Column (Quelle: Florian Graßhoff)

```
update tmdb_movies
  set txt_embedding = apex_ai.get_vector_embeddings(
    p_value           => title || '.' || overview,
    p_service_static_id => 'vector_provider_bert_base' );
```

Listing 3: APEX_AI.generate_vector_embeddings in DML

```
create hybrid vector index tmdb_hybrid_idx on tmdb_movies( overview )
parameters ( 'model bert_base_model' )
parallel 6;
```

Listing 4: DBMS_HYBRID_VECTOR Index

Fazit

Oracle APEX 24.2 bietet mit dem Application Search Feature ein flexibles und modernes Fundament, um unterschiedlichste suchbezogene Anforderungen effizient

```

select jt.score, mov.title, mov.overview
  from dbms_hybrid_vector.searchpipeline(
    json('{ "hybrid_index_name" : "tmdb_hybrid_idx",
           "search_text"       : "galaxy",
           "search_fusion"    : "INTERSECT",
           "return"           : { "topN": 15 }
         }' ) ) jt,
  tmdb_movies mov
 where mov.rowid = jt.doc_rowid
 order by jt.score desc

```

Listing 5: DBMS_HYBRID_VECTOR Suche mit searchpipeline

umzusetzen. Die Integration von Oracle Text, Ubiquitous Database Search und vor allem der AI Vector Search eröffnen vielfältige Möglichkeiten – von klassischen bis hin zu semantisch orientierten Anwendungsfällen. Durch die deklarativen Konfigurationsoptionen und die gezielte Unterstützung von KI-basierten Modellen steht ein zukunftsfähiges Werkzeug für innovative Suchlösungen zur Verfügung.

Weitere Informationen

- [1] Oracle AI Vector Search User's Guide
<https://docs.oracle.com/en/database/oracle/oracle-database/23/vecse/>

- [2] Implementing Image and Text Semantic Search in Oracle APEX Using ONNX Pipeline Models and Oracle Database 23ai
<https://blogs.oracle.com/apex/post/implementing-image-text-semantic-search-oracle-apex-onnx-database-23ai>
- [3] Easy Database Search with APEX in 23ai
<https://blogs.oracle.com/coretec/post/easy-database-search-with-apex-in-23ai>
- [4] What's New in Oracle APEX 24.2
<https://apex.oracle.com/en/platform/features/whats-new-242/>

Über den Autor

Florian Graßhoff hat bereits in der Berufsausbildung hauptsächlich mit Oracle-Datenbanken gearbeitet. Seitdem ist er



Florian Graßhoff
florian.grasshoff@Oracle.com

der Oracle-Technologie treu geblieben. Zuletzt hat er als Lead Developer und stellvertretender Teamleiter im Datenbank-Team für einen IT-Dienstleister im Bereich der inneren Sicherheit gearbeitet und ist nun seit 2022 Mitglied des APEX-Development-Teams. Dort beschäftigt er sich hauptsächlich mit dem Application Search Feature, der Integration neuer Datenbankfeatures und dem Thema Security.

DOAG 2026

Datenbank

mit Cloud Infrastructure

SUPER-SAVER

Bis 31.10.2025

+ APEX connect
Heide Park Soltau

datenbank.doag.org

18. – 19.
Mai
2026

DOAG





JSON-Dokumente und Low Code: APEX JSON Sources

Carsten Czarski, Oracle Global Services Deutschland

JSON-Dokumente sind als Bestandteil eines Datenmodells inzwischen normal. Mit der neuen Version 23ai enthält die Oracle-Datenbank sogar JSON Duality Views, welche relationale Daten als JSON bereitstellen. Speichert man ein JSON-Dokument „in die View“, sorgt die Datenbank automatisch dafür, dass die Daten auf die relationalen Tabellen verteilt werden. Kann man APEX-Anwendungen bauen, die direkt auf JSON-Dokumenten arbeiten? Ja, man kann. Dieser Artikel zeigt, wie man JSON-Dokumente mit APEX 24.2 als Datenquelle für APEX-Komponenten wie Charts, Formulare oder Reports nutzen kann. So wird APEX zur Low-Code-Plattform für alle Daten, ob relational, JSON-Dokumente oder REST.



JSON in der Oracle-Datenbank: Tabellen und Views

Die Oracle-Datenbank bietet schon seit Version 12c, die im Jahr 2014 erschienen ist, Unterstützung für JSON-Dokumente an, die seitdem ständig ausgebaut wurde. Dieser Artikel betrachtet die Datenbank-

versionen ab 19c, da ältere Versionen von Oracle nicht mehr unterstützt werden.

In 19c-Datenbanken können JSON-Dokumente als normale Datenbankspalten vom Typ VARCHAR2, CLOB oder BLOB abgelegt werden. Funktionen wie das Lesen oder Manipulieren von JSON-Dokumenten werden mit diesen Datentypen unterstützt. *Listing 1* zeigt ein

Syntaxbeispiel zum Erstellen einer Tabelle mit einer BLOB-Spalte für JSON-Dokumente.

In Version 21c wurde der Datentyp **JSON** eingeführt. Version 23ai bringt die speziellen *JSON Collection Tables* und *JSON Relational Duality Views* mit. *Listing 2* zeigt ein Syntaxbeispiel zum Erstellen einer JSON Collection Table.

```
create table my_table_with_json(  
  id          number          not null primary key,  
  doc_name    varchar2(255) not null,  
  document    blob            not null,  
  constraint my_table_json_ck check( document is json ) );
```

Listing 1: Eine Tabelle mit einer Spalte für JSON-Dokumente erstellen

ID	DOC_JSON
1	[{"PONumber":1,"Reference":"MSULLIVA-20141102","Requestor":"Martha Sullivan","User":"MSULLIVA","CostCenter":"A50","ShippingInstructions":{"name":"Martha Sullivan","Addr...
2	[{"PONumber":2,"Reference":"MSULLIVA-20141113","Requestor":"Martha Sullivan","User":"MSULLIVA","CostCenter":"A50","ShippingInstructions":{"name":"Martha Sullivan","Addr...
3	[{"PONumber":3,"Reference":"TRAJS-20140518","Requestor":"Trenna Rajs","User":"TRAJS","CostCenter":"A50","ShippingInstructions":{"name":"Trenna Rajs","Address":{"street": "...
4	[{"PONumber":4,"Reference":"TRAJS-20140520","Requestor":"Trenna Rajs","User":"TRAJS","CostCenter":"A50","ShippingInstructions":{"name":"Trenna Rajs","Address":{"street": "...
5	[{"PONumber":5,"Reference":"MSULLIVA-20141121","Requestor":"Martha Sullivan","User":"MSULLIVA","CostCenter":"A50","ShippingInstructions":{"name":"Martha Sullivan","Addr...
6	[{"PONumber":6,"Reference":"TRAJS-20140530","Requestor":"Trenna Rajs","User":"TRAJS","CostCenter":"A50","ShippingInstructions":{"name":"Trenna Rajs","Address":{"street": "...
7	[{"PONumber":7,"Reference":"VJONES-20140503","Requestor":"Vance Jones","User":"VJONES","CostCenter":"A50","ShippingInstructions":{"name":"Vance Jones","Address":{"str...
8	[{"PONumber":8,"Reference":"VJONES-20140504","Requestor":"Vance Jones","User":"VJONES","CostCenter":"A50","ShippingInstructions":{"name":"Vance Jones","Address":{"str...
9	[{"PONumber":9,"Reference":"AHUNOLD-20141130","Requestor":"Alexander Hunold","User":"AHUNOLD","CostCenter":"A60","ShippingInstructions":{"name":"Alexander Hunold","...
10	[{"PONumber":10,"Reference":"STOBIAS-20140515","Requestor":"Sigal Tobias","User":"STOBIAS","CostCenter":"A30","ShippingInstructions":{"name":"Sigal Tobias","Address":{"str...
11	[{"PONumber":11,"Reference":"STOBIAS-20140516","Requestor":"Sigal Tobias","User":"STOBIAS","CostCenter":"A30","ShippingInstructions":{"name":"Sigal Tobias","Address":{"str...
12	[{"PONumber":12,"Reference":"SVOLLMAN-20140502","Requestor":"Shanta Vollman","User":"SVOLLMAN","CostCenter":"A50","ShippingInstructions":{"name":"Shanta Vollman","...
13	[{"PONumber":13,"Reference":"SVOLLMAN-20140524","Requestor":"Shanta Vollman","User":"SVOLLMAN","CostCenter":"A50","ShippingInstructions":{"name":"Shanta Vollman","...
14	[{"PONumber":14,"Reference":"SVOLLMAN-20140525","Requestor":"Shanta Vollman","User":"SVOLLMAN","CostCenter":"A50","ShippingInstructions":{"name":"Shanta Vollman","...
15	[{"PONumber":15,"Reference":"SVOLLMAN-20140506","Requestor":"Shanta Vollman","User":"SVOLLMAN","CostCenter":"A50","ShippingInstructions":{"name":"Shanta Vollman","...

Abbildung 1: Eine Tabelle mit einer JSON-Spalte im APEX SQL Workshop (Quelle: Carsten Czarski)

APEX App Builder SQL Workshop Team Development Gallery Search CA

Application 100 \ Shared Components

- Breadcrumbs 1
- Navigation Bar List
- Search Configurations
- Themes 1
- Templates 76
- Email Templates
- Map Backgrounds
- Report Layouts
- Report Queries
- About: Shared components can display or be applied on any page within the application. This page provides a list of available shared components. Learn More ...
- Tasks: Export Application Components, Manage Supporting Objects, REST Synchronizations, Edit Application Comment, Developer Comments
- Workspace Objects

Data Sources

- REST Data Sources
- Duality Views**
- JSON Sources**

Workflows and Automations

- Task Definitions
- Automations
- Workflows

Globalization

- Globalization Attributes
- Text Messages
- Application Translations

Generative AI

- AI Attributes
- AI Configurations

Abbildung 2: JSON Sources und Duality Views als Shared Components (Quelle: Carsten Czarski)

```
create json collection table my_collection_table;
```

Listing 2: Eine JSON Collection Table in einer Oracle Database 23ai erstellen

```
create or replace emp_as_json_v as
select json_object(
  '_id'      value deptno,
  'name'     value dname,
  'employees' value (
    select json_arrayagg(
      json_object(
        '_id'      value empno,
        'ename'    value ename,
        'compensation' value json_object(
          'sal' value sal,
          'comm' value comm ) )
    from emp e
    where e.deptno = d.deptno ) )
from dept d
```

Listing 3: Eine View stellt relationale Daten als JSON bereit

```
create or replace json relational duality view emp_as_json_dv as
select json( '_id':      d.deptno,
  'dname':    d.dname,
  'employees':
  [ select json( _id:      e.empno,
    'ename':      e.ename,
    'compensation': {
      'sal': e.sal,
      'comm': e.comm } }
    from emp e
    with insert update delete
    where e.deptno = d.deptno ] }
from dept d
with insert update delete nocheck
```

Listing 4: Erstellen einer JSON Relational Duality View

```
select d.deptno,
  d.ename,
  :
  d.sal,
  d.comm
from my_table_with_json,
  json_table(
    data,
    '$'
    columns(
      deptno      number      path '$._id',
      dname       varchar2(15) path '$.dname',
      nested path '$.employees[*]' columns (
        empno     number      path '$._id',
        :
        comm      number      path '$.compensation.comm' ) ) ) d
```

Listing 5: JSON_TABLE gibt JSON-Daten als relationale Ergebnismenge zurück

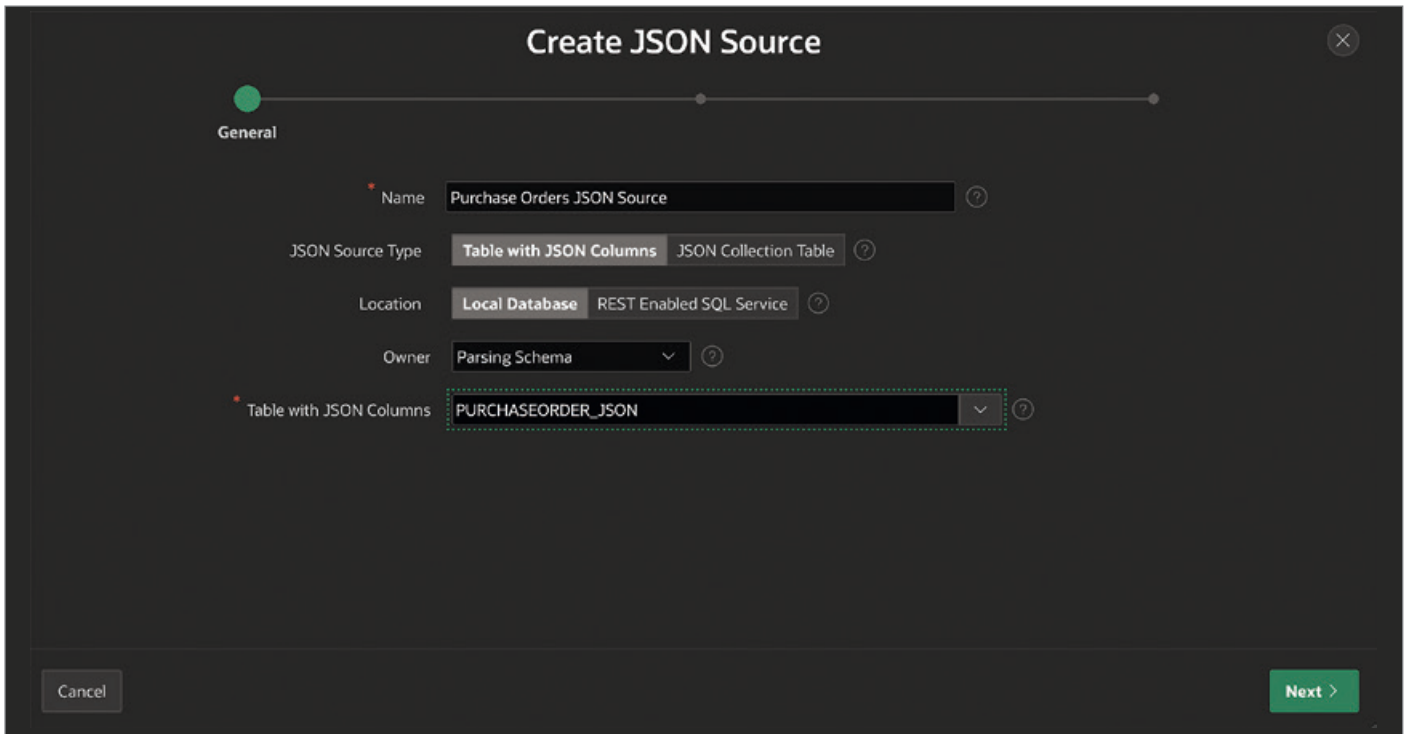


Abbildung 3: Eine neue JSON Source erstellen (Quelle: Carsten Czarski)

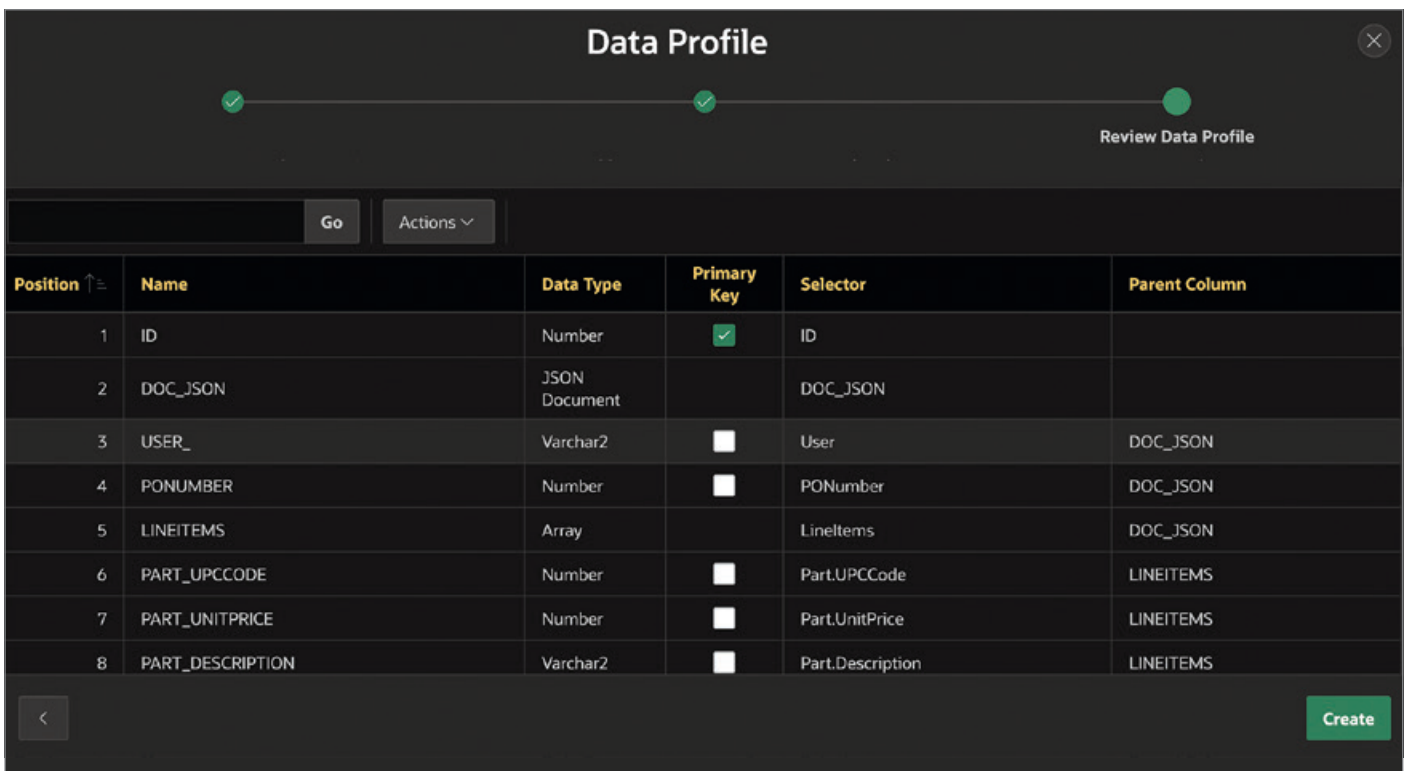


Abbildung 4: Das Data Profile gibt die Struktur der JSON-Dokumente wieder (Quelle: Carsten Czarski)

```
update purchaseorder_json
  set data = json_transform(
    data,
    SET '$.loc' = 'MUNICH' )
where json_exists( data, '$?(@."_id"==10)')
```

Listing 6: Ein JSON-Attribut mit JSON_TRANSFORM gezielt ändern

Data Profile [Cancel] [Apply Changes]

Show All Settings Columns Rediscovery Advanced

Settings

Name: Purchase Orders JSON Source

Format: JSON

Row Selector: *

Columns [Resequence Columns] [Add Column >]

Search: All Text Columns [Go] [Edit] [Add Data Column]

Sequence	Name	Column Type	Data Type	Selector	Primary Key	Vis
1	ID	Table	Number	ID	Yes	Yes
2	ITEMNUMBER	Data	Number	ItemNumber	Yes	Yes
3	DOC_JSON	Table	JSON Document	DOC_JSON	No	Yes
4	USER_	Data	Varchar2	User	No	Yes
5	PONUMBER	Data	Number	PONumber	No	Yes
6	LINEITEMS	Data	Array	LineItems	No	Yes

About

Data Profiles describe how Data Source format are being parsed and converted to rows and columns. For XML or JSON data format, the Row Selector attribute stores an XML or JSON path expression pointing to the node containing the collection of rows. Data Profile Columns determine how one row is parsed and converted to multiple columns.

Stay on page

Abbildung 5: Das Data Profile, welches die JSON-Struktur beschreibt, editieren (Quelle: Carsten Czarski)

Create Interactive Report [X]

Page Definition

Page Number: 2

Name: Purchase Orders JSON

Page Mode: Normal Modal Dialog Drawer

Include Form Page:

Data Source

Data Source: JSON Source

JSON Source: Purchase Orders JSON Source

Nested Rows: None

Navigation

< Cancel [Create Page]

Abbildung 6: Ein neuer Interactive Report – direkt auf eine JSON Source (Quelle: Carsten Czarski)

Mit Views können relationale Daten als JSON bereitgestellt werden. Listing 3 zeigt, wie die SQL/JSON-Funktionen der Datenbank verwendet werden, um Daten der (wohlbekannten) Tabelle EMP als JSON bereitzustellen.

DML kann auf solche Views aus dem Stand nicht ausgeführt werden. Möchte man die Möglichkeit haben, INSERT- oder UPDATE-Operationen auf der View durchzuführen, so muss man entweder aufwändige INSTEAD OF Trigger schrei-

ben oder die in Datenbankversion 23ai neu eingeführten *JSON Relational Duality Views* nutzen. Listing 4 zeigt das gleiche Beispiel, nun aber als Duality View, welche DML-Operationen aus dem Stand unterstützt.

User	Ponumber	Reference	Requestor	Costcenter	Shippinginstructions	Address City	Special Instructions
LPOPP	275	LPOPP-20141115		A100	Seattle		Expidite
LPOPP	276	LPOPP-20141116		A100	Seattle		COD
LPOPP	277	LPOPP-20141118	Ismael Sclarra	A100	Seattle		Air Mail
DFAVIET	280	DFAVIET-20141001	John Chen	A100	Seattle		Expidite
DFAVIET	287	DFAVIET-20141007	Jose Manuel Urman	A100	Seattle		Hand Carry
JCHEN	289	JCHEN-20141112	Luis Popp	A100	Seattle		Air Mail
JCHEN	290	JCHEN-20141112	Nancy Greenberg	A100	Seattle		Expidite
LPOPP	303	LPOPP-20141101	Luis Popp	A100	Seattle		Hand Carry
NGREENBE	393	NGREENBE-20141126	Nancy Greenberg	A100	Seattle		Counter to Counter
NGREENBE	403	NGREENBE-20141126	Nancy Greenberg	A100	Seattle		Expidite
JCHEN	427	JCHEN-20140904	John Chen	A100	Seattle		COD
JCHEN	428	JCHEN-20140906	John Chen	A100	Seattle		Expidite
JCHEN	429	JCHEN-20140907	John Chen	A100	Seattle		Ground
JCHEN	430	JCHEN-20140908	John Chen	A100	Seattle		Expidite
JCHEN	431	JCHEN-20140913	John Chen	A100	Seattle		Hand Carry

Abbildung 7: Interactive Report auf einer JSON Source in Aktion (Quelle: Carsten Czarski)

The screenshot shows the configuration interface for an Oracle APEX Interactive Report. The main area displays the 'Nested Rows' configuration, which includes a list of columns: ID, USER_, PONUMBER, REFERENCE, REQUESTOR, COSTCENTER, SHIPPINGINSTRUCTIONS, and SHIPPINGINSTRUCTIONS. The 'Data Profile' section is expanded, showing a dropdown menu with 'None', '1. LINEITEMS', and '2. SHIPPINGINSTRUCTIONS'. The 'Source' section is also visible, showing 'JSON Source' and 'Purchase Orders JSON'.

Abbildung 8: Der Interaktive Bericht arbeitet nun mit Line Items (Quelle: Carsten Czarski)

JSON in der Oracle-Datenbank: JSON-Funktionen

Die hier beschriebenen JSON-Operationen zum Parsen oder Manipulieren von JSON-Dokumenten funktionieren völlig unabhängig davon, ob die JSON-Dokumente aus einer Tabelle oder View kommen, und welcher Datentyp (CLOB, BLOB oder JSON) verwendet wird.

Die SQL-Funktion `JSON_TABLE` stellt die Inhalte eines JSON-Dokumentes als relationales Ergebnis mit Zeilen und Spalten dar. `JSON_TABLE` unterstützt mit der `NESTED PATH`-Klausel auch verschachtelte JSON-Arrays, wie *Listing 5* zeigt.

Neben `JSON_TABLE` stehen auch SQL-Funktionen wie `JSON_VALUE` (gibt einen einzelnen Wert aus dem JSON zurück) oder `JSON_EXISTS` (prüft, ob ein be-

stimmtes Attribut im JSON-Dokument existiert) bereit.

Zur Manipulation von JSON-Dokumenten steht ab Datenbankversion 19c die SQL-Funktion `JSON_TRANSFORM` zur Verfügung. *Listing 6* zeigt, wie ein einzelnes JSON-Attribut mit `JSON_TRANSFORM` gezielt verändert wird.

Wird `JSON_TRANSFORM` mit einer View verwendet, so wird die Datenbank mit

Create Interactive Report

Page Definition

- Page Number: 4
- Name: Line Items Overview
- Page Mode: Normal | Modal Dialog | Drawer
- Include Form Page:
- Form Page Number: 5
- Form Page Name: Edit Line Item
- Form Page Mode: Normal | Modal Dialog | **Drawer**

Data Source

- Data Source: JSON Source
- JSON Source: Purchase Orders JSON Source
- Nested Rows: 1. LINEITEMS

Navigation

< Cancel Next >

Abbildung 11: Erstellen von Report und Form – direkt auf einer JSON Source (Quelle: Carsten Czarski)

Create Interactive Report

Primary Key

- Primary Key Column 1: ID (Number)
- Primary Key Column 2: ITEMNUMBER (Number)
- Primary Key Column 3: - Select -

< Cancel Create Page

Abbildung 12: Primary Key-Spalten zum Identifizieren eines Line Item (Quelle: Carsten Czarski)

machen. *Abbildung 1* zeigt eine Beispieltabelle im APEX SQL Workshop. Diese enthält zwei Spalten: **ID** ist der numerische Primary Key und **DOC_JSON** enthält die JSON-Dokumente.

Der erste Schritt zur Arbeit mit JSON-Dokumenten in APEX ist das Erstellen einer *JSON Source*. Diese dient als Datenquelle für APEX-Komponenten und enthält alle Informationen über die Tabel-

le oder View und über die Struktur der gespeicherten JSON-Daten als *Data Profile*.

JSON Sources sind *Shared Components*. Auf 19c-Datenbanken unterstützt APEX **JSON-Tabellen**, auf Version 23ai sind

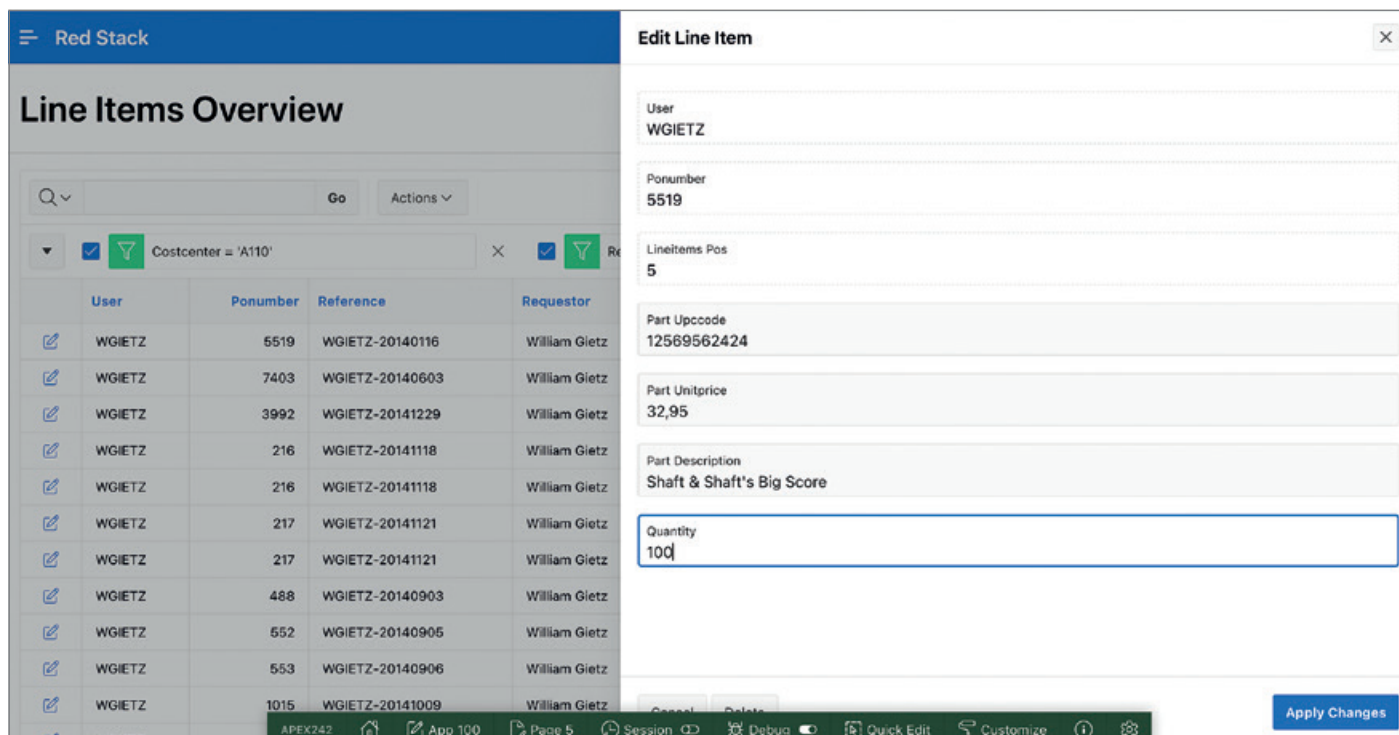


Abbildung 13: Mit dem Formular können JSON-Daten direkt geändert werden (Quelle: Carsten Czarski)

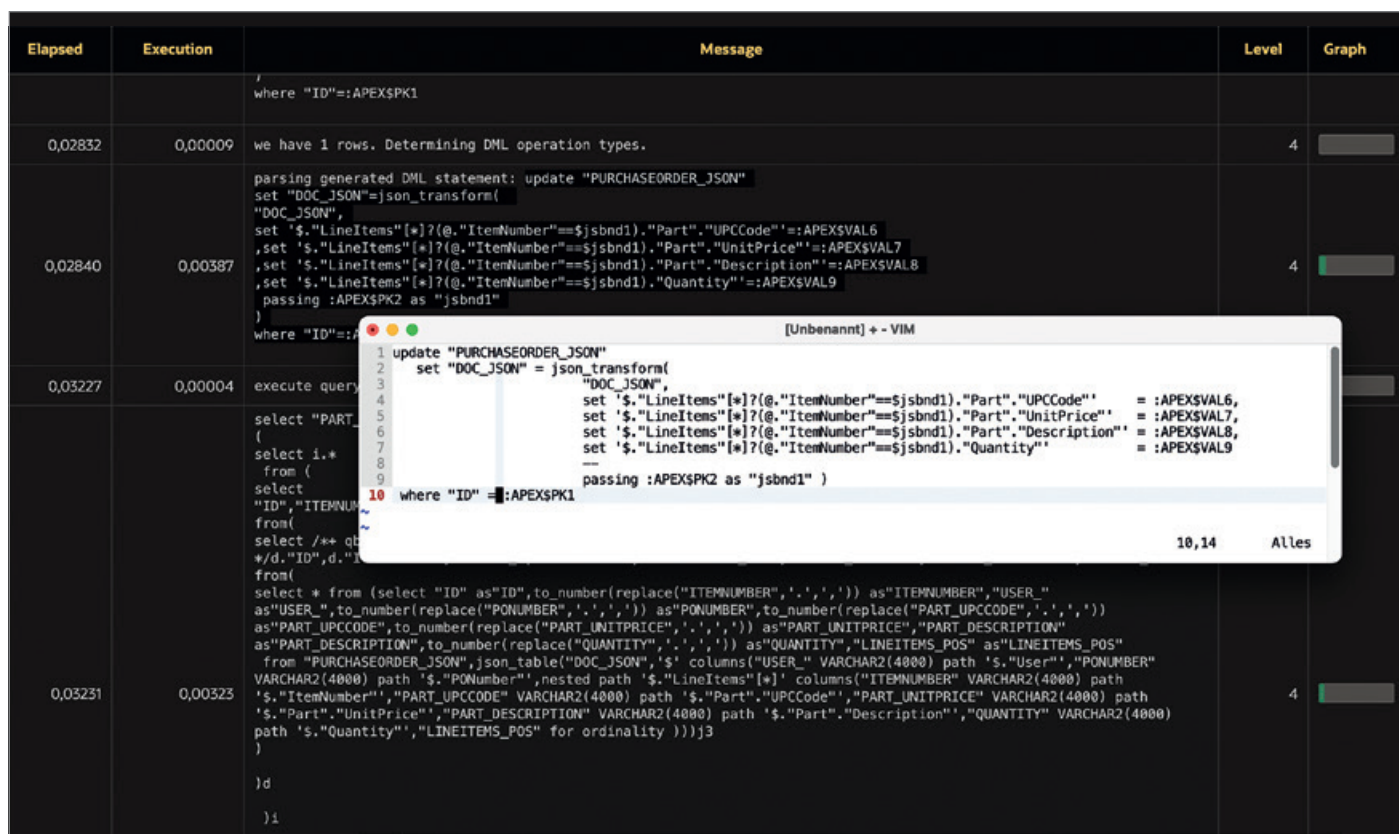


Abbildung 14: APEX nutzt JSON_TRANSFORM, um Formular-DML auszuführen (Quelle: Carsten Czarski)

auch **JSON Collection Tables** und **Duality Views** dabei (siehe Abbildung 2).

Eine neue JSON Source kann für eine normale Tabelle mit JSON-Spalten, oder für eine *JSON Collection Table*, erstellt

werden. Der Wizard benötigt nur den Namen der neuen JSON Source und den Namen der Tabelle, in welcher die JSON-Dokumente gespeichert sind (siehe Abbildung 3).

Im zweiten Schritt werden die konkreten JSON-Tabellenspalten ausgewählt. Wenn ein JSON-Schema hochgeladen wird, ermittelt APEX das *Data Profile* für die JSON Source aus diesem, andernfalls

werden die in der Tabelle gespeicherten JSON-Dokumente selbst analysiert. Wird eine JSON Source also für eine leere Tabelle erstellt, so sollte ein JSON-Schema hochgeladen werden.

Im letzten Schritt zeigt der Assistent das ermittelte *Data Profile*, also die aus der JSON-Struktur abgeleiteten Spalten mit Datentypen, an (siehe *Abbildung 4*). Wenn APEX einen Primary Key nicht korrekt erkennt, kann dies manuell angepasst werden. Ein Klick auf den **Create**-Button erstellt die JSON Source.

Nachdem die JSON Source erstellt wurde, kann sie natürlich auch verändert werden. Das Data Profile, welches die JSON-Struktur beschreibt, lässt sich beliebig anpassen oder erweitern, wie *Abbildung 5* zeigt.

Nach dem Erstellen der JSON Source können APEX-Komponenten auf einfachste Art und Weise erstellt werden. *Abbildung 6* zeigt, wie ein neuer *Interactive Report* mit dem Create Page Wizard erstellt wird.

Der Interactive Report kann sofort genutzt werden, alle Funktionen arbeiten so, wie man es von relationalen Tabellen gewohnt ist (siehe *Abbildung 7*).

Auch *hierarchische* JSON-Dokumente, also solche mit geschachtelten Arrays, werden von JSON Sources unterstützt. Das **Nested-Rows**-Attribut im Page Designer legt fest, welches Array in der JSON-Hierarchie als Datenquelle für den Bericht dienen soll. Der Interactive Report kann auf die Spalten der gewählten Hierarchieebene und aller übergeordneten Ebenen zugreifen. *Abbildung 8* zeigt, wie das **Nested-Rows**-Attribut für den Interactive Report im Page Designer umgestellt wird.

Als Konsequenz bekommt der Interactive Report nicht mehr eine Zeile pro **Purchase-Order**, vielmehr bekommt er eine Zeile pro **Line Item** (siehe *Abbildung 9*). Wie zuvor, funktioniert der Interactive Report für den Endbenutzer ganz normal und intuitiv.

Alle APEX-Komponenten funktionieren mit JSON Sources gleich. Intern verwendet APEX die JSON_TABLE-Funktion der Datenbank, um das JSON zu parsen und die Daten als Zeilen und Spalten aufzubereiten. Das lässt sich mit der APEX-Debug-Funktion leicht überprüfen: *Abbildung 10* zeigt die APEX-Debug Ausgabe für den Interactive Report. Die Nutzung von JSON_

TABLE und der NESTED PATH-Klausel ist deutlich zu erkennen.

Auch Formulare oder editierbare Interactive Grids können JSON Sources verwenden – und hier wird auch das *Speichern von Änderungen* deklarativ unterstützt.

Mit dem Create Page Wizard kann ein Report zusammen mit einem Formular erzeugt werden – ein Klick auf eine Zeile im Bericht verzweigt dann auf das Formular, wo dieser Datensatz bearbeitet werden kann (siehe *Abbildung 11*). Wieder wird die JSON Source **Purchase Orders** verwendet, es soll diesmal aber direkt mit der Hierarchieebene **Line Items** gearbeitet werden. Der Bericht soll also einzelne Line Items wiedergeben, und das Formular soll einzelne Line Items bearbeiten können.

Im zweiten Schritt des Assistenten müssen die Primary Keys gewählt werden. APEX muss jedes Line Item eindeutig identifizieren können – es wird also ein Primary Key für die Purchase-Order und einer für das Line Item benötigt (siehe *Abbildung 12*).

Abbildung 13 zeigt Bericht und Formular in Aktion. Der Bericht ist von vornherein mit einem Editier-Link für jede Zeile ausgestattet. Klickt man darauf, so öffnet sich ein Formular, und Attribute können editiert werden. Ein Klick auf den **Apply-Changes**- Button speichert die Änderungen *direkt im JSON-Dokument* ab.

Wichtig zu wissen ist, dass ein Formular nur die Attribute der Hierarchieebene editieren kann, für die es erzeugt wurde. Parent-Attribute können angezeigt, aber nicht verändert werden.

Wiederum zeigt ein Blick in die Ausgabe von APEX-Debug, dass APEX die JSON-Funktionen der Datenbank verwendet – zum Speichern von Änderungen eines Formulars ist es die JSON_TRANSFORM-Funktion (siehe *Abbildung 14*).

Fazit

JSON Sources in APEX 24.2 machen Low-Code-Entwicklung mit JSON-Dokumenten ebenso leicht wie mit relationalen Tabellen. Ist eine JSON Source erstellt, können APEX-Komponenten genauso einfach erstellt und genutzt werden wie mit normalen Tabellen und Spalten. Das SQL mit den nötigen SQL_

JSON-Funktionen generiert APEX ganz automatisch.

Damit öffnen sich neue Möglichkeiten: Daten, die „nur“ als JSON-Dokumente vorliegen, können einfach in die Datenbank geladen werden – und Auswertungen wie Berichte, oder Charts werden in wenigen Minuten mit ein paar Klicks generiert.

JSON Sources unterstützen „normale“ Tabellen mit JSON-Spalten genauso wie die neuen *JSON Collection Tables* oder *JSON Relational Duality Views* in der Oracle Database 23ai – und dabei funktionieren sie immer gleich. Man kann sogar eine JSON Collection Table durch eine Duality View ersetzen, ohne dass die APEX-Anwendung verändert werden muss. Produktiver kann man sich Anwendungsentwicklung auf JSON-Dokumenten nicht vorstellen.

Weitere Informationen

- [1] Informationen zu Application Express und Demo-Umgebung
<http://apex.oracle.com>
- [2] Oracle Application Express Blog - mit vielen Postings zu REST-Services in APEX
<http://blogs.oracle.com/apex>
- [3] Oracle Application Express Blog-Posting zu JSON Sources
<https://blogs.oracle.com/apex/post/leveraging-json-sources-and-duality-views-for-flexible-component-development-in-oracle-apex>



Carsten Czarski
carsten.czarski@oracle.com

JavaLand

GROßES
COMMUNITY-PROGRAMM!



im **EUROPA[★]PARK**® in Rust
10. - 12. MÄRZ 2026



      | #JavaLand | www.javaland.eu

Präsentiert von:  iJUG
Verbund

 heise medien

DOAG

Veranstalter: *JavaLand*

Hintergrund

Viele Aufgaben können durch die Verwendung von Sprache als Eingabemedium vereinfacht werden. Dazu zählen beispielsweise das Versenden von Nachrichten, das Suchen mithilfe von Suchmaschinen oder das Steuern von Funktionen im Smart-Home-Bereich. Durch Spracheingabe können so sich wiederholende Tasks komfortabler bearbeitet werden, indem zum Beispiel nicht jedes Mal ein Text eingetippt oder dieselben Menüpunkte angeklickt werden müssen.

Auch im Kontext von APEX finden sich viele Aktionen, die anwendungsübergreifend häufig ausgeführt werden. Dazu zählen insbesondere die Dateneingabe, beispielsweise beim Ausfüllen von Formularen, oder das Suchen in großen Datenmengen, etwa in einem Interactive Grid. Die Möglichkeit, für diese Aktionen ebenfalls eine Spracheingabe zu verwenden, könnte somit auch hier zu einer optimierten Bedienung und einem einfacheren Umgang mit den Daten beitragen. Dies gilt insbesondere bei der Verwendung von PWAs („Progressive Web Apps“), also mobilen Anwendungen, bei denen Spracheingabe häufig zu den Standardfunktionalitäten gehört. Im Sinne der Barrierefreiheit könnten zudem mehr Aktionen auch ohne eine Tastatur ausgeführt sowie auf natürlichsprachliche Weise gesteuert werden, ohne dass ein detailliertes Verständnis der Syntax erforderlich ist.

Whisper

Whisper ist ein automatisiertes Spracherkennungssystem, welches von OpenAI entwickelt wurde. Zum Training des zugrundeliegenden Modells wurden circa 680.000 Stunden Audiomaterial verwendet, davon circa zwei Drittel auf Englisch. Durch die Verwendung von mehrsprachigem Material bietet das Modell die Möglichkeit, verschiedene Sprachen zu erkennen und optional die Eingabe in Englisch übersetzen zu lassen. Laut eigener Aussage von OpenAI sorgt der umfangreiche Trainingsdatensatz dafür, dass das Modell besser mit Akzenten oder Hintergrundgeräuschen in den verwendeten Audioaufnahmen umgehen kann. Im Vergleich zu anderen Spracherkennungsmo-

dellen, mit zum Teil besserer Präzision in einer bestimmten Sprache, soll das Modell so insgesamt flexibler sein.

Das Modell ist nach dem Encoder-/Decoder-Prinzip aufgebaut, einer häufig verwendeten Modellarchitektur für Aufgaben im Bereich der Sprachverarbeitung („Natural Language Processing“). Das eingegebene Audiosignal wird zunächst für eine schnellere Verarbeitung in Abschnitte („Chunks“) von 30 Sekunden unterteilt. Gleichzeitig wird durch die Länge der Chunks sichergestellt, dass ausreichend Kontext vorhanden ist, um den Inhalt des Textes besser erfassen zu können. Die Verarbeitung erfolgt dann mithilfe von Vektoren, die die entsprechenden Informationen der Eingabe repräsentieren. Dabei werden nicht nur der aktuelle Zeitabschnitt, sondern auch zuvor verarbeitete Daten berücksichtigt. Dies bezeichnet man auch als „Self-“ beziehungsweise „Cross-Attention“. Es folgt eine Ausgabe in Form von Token, die neben dem transkribierten Text auch Zusatzdaten wie die erkannte Sprache oder Zeitstempel enthalten [1].

Aktuell sind sechs verschiedene Modelle von Whisper verfügbar. Sie unterscheiden sich im Wesentlichen in Bezug auf die Größe und die damit verbundene Fehlerquote. Je kleiner das Modell, desto schneller läuft die Verarbeitung, aber desto weniger präzise ist das Ergebnis. Die Fehlerquote ist zudem stark abhän-

gig von der verwendeten Sprache. Zur Verbesserung des Ergebnisses kann optional beim Aufruf der API ein Sprachparameter mitgegeben werden. Alternativ versucht das Modell die Sprache automatisch anhand der Eingabe zu erkennen. Die Unterschiede bei den Fehlerquoten verschiedener Sprachen sind unter anderem auf den Umfang der Trainingsdaten zurückzuführen. Da ein Großteil der Trainingsdaten auf Englisch ist, ist die Fehlerquote allgemein bei germanischen Sprachen, zu denen auch die englische Sprache zählt, geringer. Auch die globale Verbreitung der Sprachen und damit einhergehend das zur Verfügung stehende Material wirken sich darauf aus [2].

Das Whisper-Modell kann über die frei zugängliche API von OpenAI verwendet werden. Die Kosten für die Verwendung liegen bei 0.006 US-Dollar pro Minute transkribiertem Audiomaterial [3]. Alternativ kann das Modell lokal verwendet werden, beispielsweise in Python mithilfe von PyTorch. Für diese Arbeit wurde ausschließlich der Weg über die API berücksichtigt.

Whisper Web

Um eine vereinfachte Integration von Whisper im Webkontext zu ermöglichen, bietet sich die Transformers-Bibliothek von Hugging Face an [4]. Die Bibliothek

```
import { pipeline } from "@xenova/transformers";

/* constants */
const task = 'automatic-speech-recognition';
const model = 'Xenova/whisper-base';

/* options */
const options = {
  task: 'transcribe',
  language: 'de'
};

/* user input */
const objectURL = this.audio.src;

/* whisper web integration */
const transcriber = await pipeline(task, model);
const output = await transcriber(text, options);

// get result via output.text
```

Listing 1: Code zum Aufrufen von Whisper Web

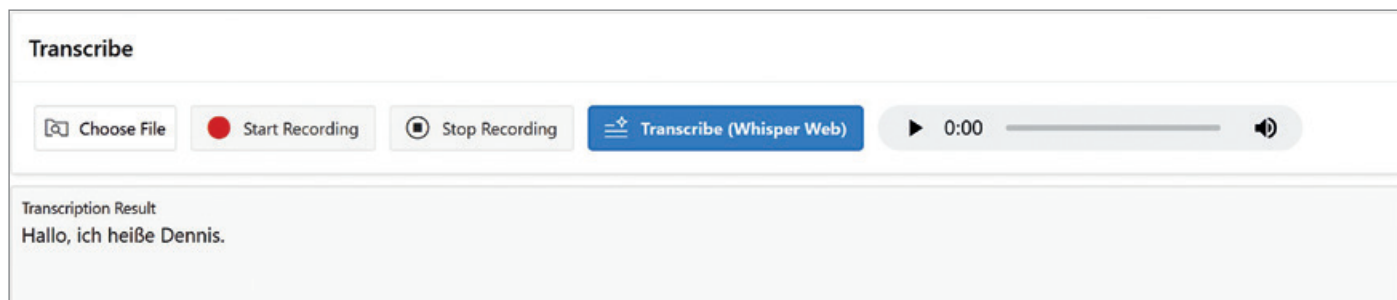


Abbildung 1: Region Plugin und Ergebnis der Spracherkennung (Quelle: Dennis Amthor)

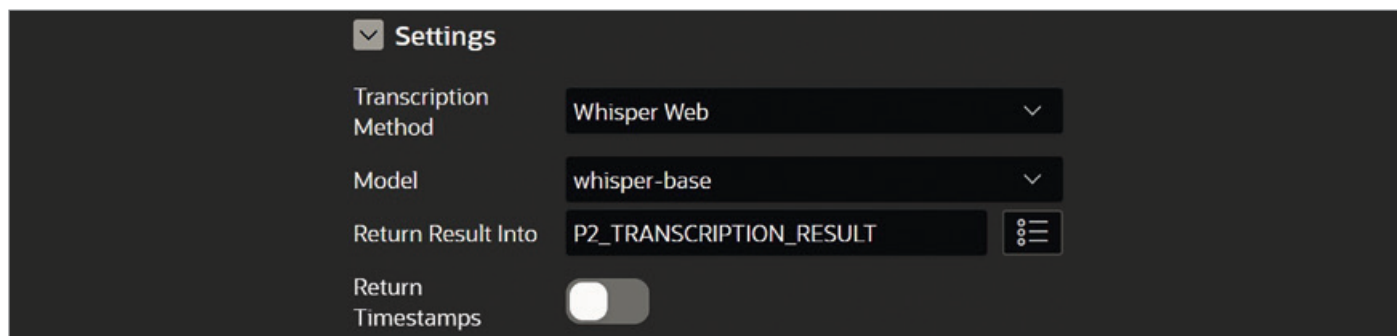


Abbildung 2: Settings im Page Designer (Quelle: Dennis Amthor)

```

declare
  l_blob_content blob;
  l_filename      varchar2(100);
  l_mime_type     varchar2(100);
  l_multipart     apex_web_service.t_multipart_parts;
  l_response      clob;
begin
  l_blob_content :=
apex_web_service.clobbase64blob(apex_application.g_clob_01);
  l_filename      := apex_application.g_x02;
  l_mime_type     := apex_application.g_x03;

  apex_web_service.append_to_multipart(
    p_multipart => l_multipart,
    p_name      => 'file',
    p_filename  => l_filename,
    p_content_type => l_mime_type,
    p_body_blob => l_blob_content
  );

  apex_web_service.append_to_multipart(
    p_multipart => l_multipart,
    p_name      => 'model',
    p_body      => 'whisper-1'
  );

  l_response := apex_web_service.make_rest_request(
    p_url          => 'https://api.openai.com/v1/audio/transcriptions',
    p_http_method => 'POST',
    p_body_blob    => apex_web_service.generate_request_body(l_multipart),
    p_credential_static_id =>
      p_plugin.attributes.get_varchar2('credential_static_id')
  );
end;

```

Listing 2: Code für den REST-Call der OpenAI API

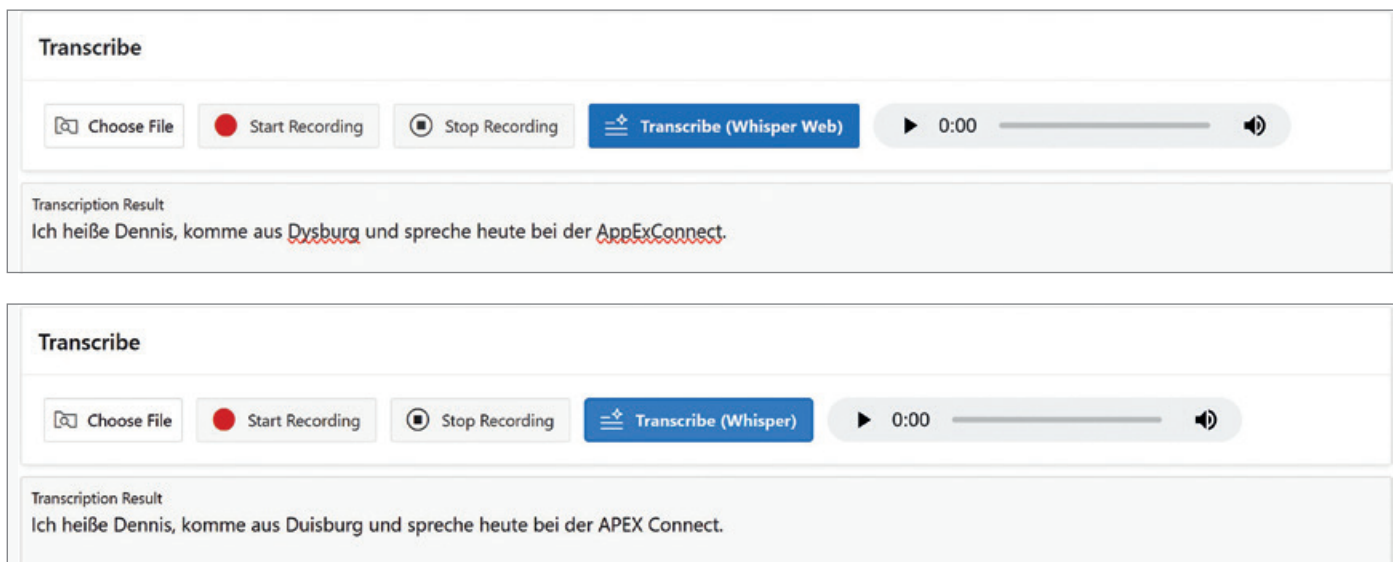


Abbildung 3: Vergleich Whisper Web und Whisper (Quelle: Dennis Amthor)

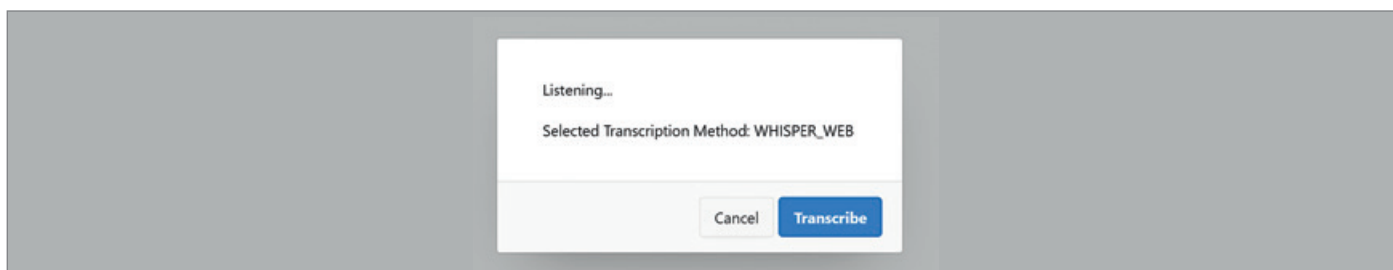


Abbildung 4: Aufnahme-Dialog (Quelle: Dennis Amthor)

```

/* init SpeechRecognition */
const SpeechRecognition = window.SpeechRecognition ||
  window.webkitSpeechRecognition;

this.recognition = new SpeechRecognition();

/* collect result here */
let transcript = '';

/* config */
this.recognition.continuous = true;
this.recognition.lang = 'de'
this.recognition.interimResults = false;
this.recognition.maxAlternatives = 1;

/* append to result */
this.recognition.onresult = (event) => {
  transcript = transcript +
    Array.from(event.results).slice(1)[0][0].transcript;
};

this.recognition.onend = (event) => {
  // get result via transcript variable
}

/* start listening */
this.recognition.start();

```

Listing 3: Code zum Aufrufen der Web Speech API

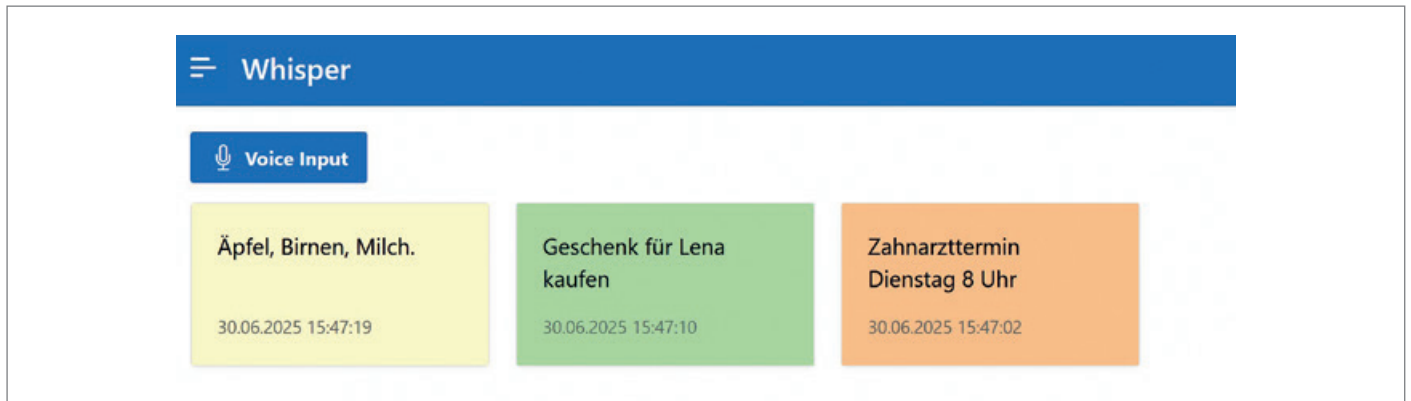


Abbildung 5: Notizenapp (Quelle: Dennis Amthor)

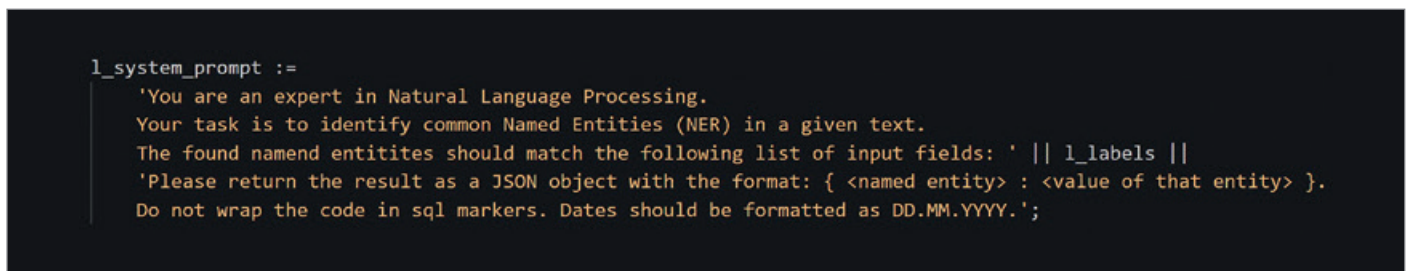


Abbildung 6: System-Prompt an die KI zur Extraktion der Formulardaten (Quelle: Dennis Amthor)

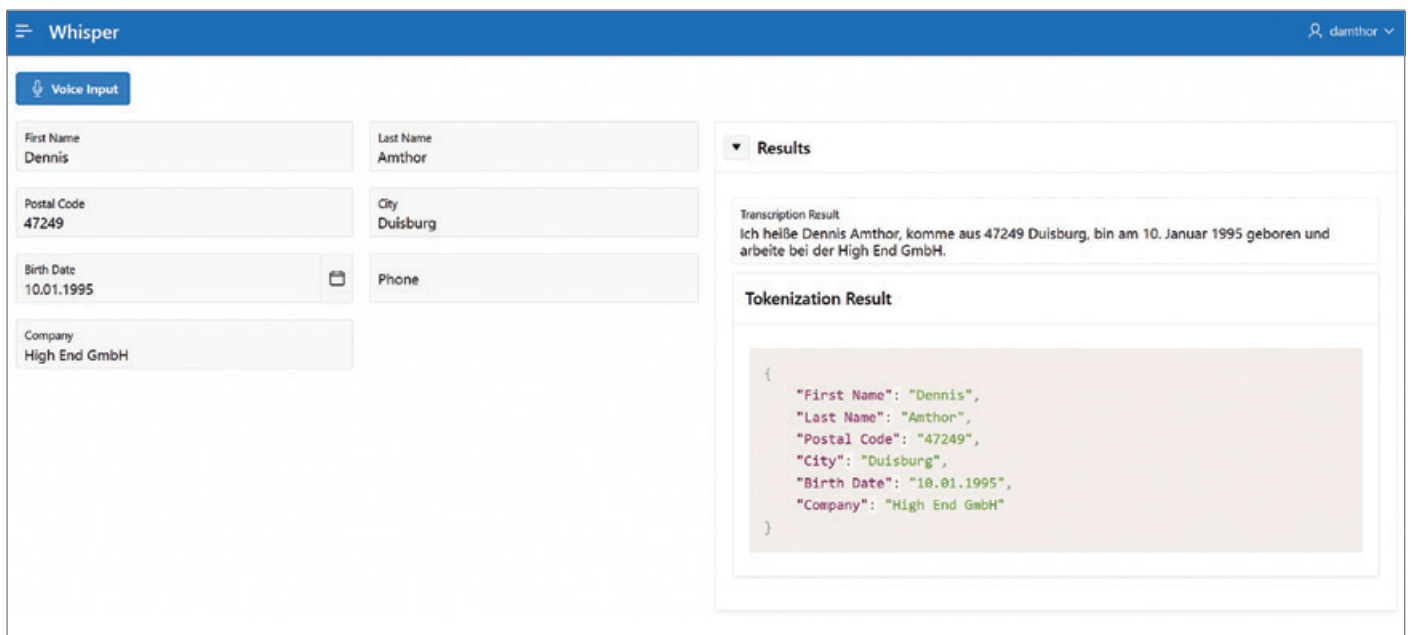


Abbildung 7: Ausfüllen eines Formulars (Quelle: Dennis Amthor)

beinhaltet eine Vielzahl vortrainierter Modelle aus verschiedenen Anwendungsbereichen wie dem Natural Language Processing oder generativen Tasks zur Erstellung von multimedialen Inhalten. Mit Whisper Web wird auch eine Version des Modells von OpenAI zur Spracherkennung angeboten. Darüber hinaus gibt es mit Transformers.js ein JavaScript-Framework, mit dem ausgewählte Modelle di-

rekt im Browser verwendet werden können. Grundlage dafür ist die Integration von ONNX Runtime, welches ein Standardformat zur Definition von Deep-Learning-Modellen nutzt. So können auch eigene lokale Modelle konvertiert und über das Framework geladen werden. Die Bibliothek beinhaltet eine JavaScript-API, mit der das Modell geladen und aufgerufen werden kann. Das Modell wird dabei ent-

weder über ein CDN in den Cache oder, bei Verwendung von eigenen Modellen, direkt aus dem lokalen Speicher geladen, was eine Offline-Nutzung ermöglicht.

Alternativen

Neben Whisper gibt es noch weitere Möglichkeiten, Spracherkennung im

Webkontext zu integrieren. Die Web-Speech-API-Spezifikation ermöglicht es beispielsweise, browserspezifische Methoden zur Spracherkennung ohne zusätzliche JavaScript-Bibliotheken zu nutzen. Die API ist aktuell nur in Chromium-basierten Browsern sowie Safari verfügbar und nutzt vom Browser definierte Modelle beziehungsweise WebserVICES, um die Spracheingabe zu verarbeiten. Damit ist keine Integration eigener Modelle möglich und die Ausführung erfordert stets eine Verbindung zum Internet beziehungsweise dem verwendeten Dienst [5].

Auch in der Oracle Cloud gibt es mit OCI Speech eine Lösung für Spracherkennung. Der Schwerpunkt liegt hier auf der Nutzung des OCI-Cloud-Interfaces und der Verwendung der dortigen Strukturen wie dem Object Storage zur Speicherung der Audioinformationen. Als Grundlage wird eine Free Tier oder eine bezahlte Instanz benötigt. Aufgrund der Abhängigkeiten von der Cloudinstanz und der sich stark unterscheidenden Vorgehensweise bei der Implementierung, insbesondere in Bezug auf APEX, wurde die Integration von OCI Speech für diesen Artikel nicht weiter betrachtet.

Implementierung

Für die Integration von Whisper in APEX wurden beide beschriebenen Optionen berücksichtigt: der direkte Aufruf des von OpenAI gehosteten Modells über die API und die lokale Integration mithilfe von Transformers.js. Der Aufruf in Richtung OpenAI erfolgt dabei über einen REST-Call an den entsprechenden Endpunkt für das Whisper-Modell. Für die JavaScript-basierte Integration wurde das base-Modell aus der Transformers-Bibliothek verwendet. Die Funktionalitäten von Transformers.js können über ein NPM-Package bezogen werden. Der Aufruf der API wird dann in eine Funktion gekapselt und mithilfe von Webpack in eine JavaScript-Datei pakettiert, um eine einfache Einbindung in APEX zu ermöglichen.

Zum Testen und Vergleichen der verschiedenen Optionen wurden zwei Plugins entwickelt. Die Umsetzung als Plugins ermöglicht das flexible Einsetzen der verschiedenen Methoden in Verbindung mit unterschiedlichen APEX-Komponenten.

Ziel ist ein Vergleich der verschiedenen Ansätze hinsichtlich der Umsetzbarkeit, des Implementierungsaufwandes und der Einsatzmöglichkeiten unter Berücksichtigung von Performance und Fehlerquote.

Region Plugin

Das Region Plugin (*siehe Abbildung 1*) enthält mehrere Buttons und Elemente, die basierend auf der Konfiguration dynamisch generiert werden. Die Eingabe des zu transkribierenden Audios kann durch den Upload einer Datei oder die Aufnahme über das Mikrofon erfolgen. Für die Aufnahme wird die MediaStream Recording API verwendet, die standardmäßig im Browser verfügbar ist. Das Ergebnis des Uploads beziehungsweise der Aufnahme wird in eine ObjectURL umgewandelt und an ein Audio-Element übergeben, sodass es auch direkt im Browser abgespielt werden kann. Über einen weiteren Button lässt sich anschließend die Spracherkennung starten. Das Ergebnis wird in ein Page Item gespeichert, welches zuvor in den Plugin-Attributen im Page Designer angegeben wurde (*siehe Abbildung 2*).

Weiterhin kann auch die zu verwendende Methode sowie das Modell in den Attributen ausgewählt werden. Die Rückgabe von Zeitstempel-Informationen kann über einen Switch ein- und ausgeschaltet werden. Bei Whisper Web enthält das Ergebnis, welches die Transformers.js-API zurückliefert, standardmäßig nur den transkribierten Text. Der Aufruf der API erfolgt über die pipeline-Funktion, die als Parameter den Task, also welche Aktion ausgeführt werden soll, sowie das Modell erwartet. In dem daraus erzeugten Pipeline-Objekt können dann noch zusätzliche Konfigurationsparameter, beispielsweise für die Sprache, gesetzt werden. Zusätzlich muss hier der Input mitgegeben werden, in diesem Fall also die ObjectURL, die zuvor aus dem Input generiert wurde. Der gesamte Codeblock zur Verarbeitung mit Whisper Web ist in *Listing 1* aufgeführt. Das Ergebnis ist ein JSON-Objekt mit dem Text, der dann der Oberfläche angezeigt werden kann.

Die ersten Tests mit Whisper Web zeigen, dass mit dem base-Modell grund-

sätzlich gute Ergebnisse erzielt werden können, was die Genauigkeit der Spracherkennung angeht. Beim erstmaligen Aufruf wird zunächst das Modell über ein CDN abgerufen und im lokalen Cache abgelegt. Je nach Modell wird hier unterschiedlich viel Speicherplatz beansprucht. Für das hier verwendete base-Modell werden circa 250 MB benötigt. Auffällig sind hierbei die zum Teil sehr langen Wartezeiten, bis ein Ergebnis zurückkommt. Da die Ausführung mit ONNX Runtime standardmäßig über die CPU läuft, wird dies auch durch die Hardware des Gerätes beeinflusst. Eine alternative Nutzung von WebGPU zur Einbindung der Grafikkarte ist grundsätzlich vorgesehen, jedoch in Teilen nur als experimentelle Implementierung vorhanden.

Wesentlich besser ist die Performance bei Verwendung der öffentlichen API von OpenAI. Auch hier wird der Input zunächst mit JavaScript verarbeitet und anschließend über AJAX an die Datenbank übergeben. Mithilfe des „*apex_web_service*“-Packages erfolgt dann ein REST-Call an den OpenAI-Endpunkt (*siehe Listing 2*). Die notwendigen Informationen wie das gewählte Modell sowie der Audioinput werden als Multipart Form im Body des Requests angehängt. Die Konfiguration der Zugangsdaten kann direkt in den Web Credentials erfolgen.

Die Genauigkeit der Ergebnisse mit dieser Methode ist vergleichbar, in Teilen sogar besser als bei Whisper Web, speziell in Bezug auf Eigennamen oder besondere Bezeichnungen. Hier scheint die individuelle Erkennung und Einbeziehung des Kontextes grundsätzlich besser zu funktionieren (*siehe Abbildung 3*).

Als dritte Option für das Plugin wurde die Integration der Web Speech API betrachtet. Statt einer Verarbeitung von reinen Audiodaten erfolgt hier direkt der Aufruf entsprechender Funktionen aus der API, um die Aufnahme und anschließend die Spracherkennung zu starten (*siehe Listing 3*). Das bedeutet auch, dass es nicht möglich ist, alternativ hochgeladene Audiodateien zu verwenden. Auch ein nachträgliches Anhören der Aufnahme ist nicht möglich. Die Rückgabe erfolgt dann wieder analog als Text in ein Page Item.

Die Spracherkennung in Google Chrome liefert auch hier zufriedenstellende Ergebnisse in einer angemessenen Zeit.

Zu beachten ist, dass man deutlich weniger Optionen für das verwendete Modell bei der Spracherkennung hat, da die gesamte Implementierung dem Browser überlassen wird. Dies kann insbesondere unter Datenschutzaspekten ein kritischer Punkt sein, der je nach Anwendungsfall berücksichtigt werden sollte.

Demos

Zum weiteren Evaluieren der Anwendungsmöglichkeiten von Spracherkennung in APEX wurden drei Demofälle erstellt, die exemplarisch die Verwendung demonstrieren sollen. Um die User-Interaktion dabei möglichst einfach zu gestalten, wurde eine reduzierte Version des Plugins verwendet. Bei einem Klick auf einen beliebigen Button wird so eine Dynamic Action ausgeführt, um direkt die Sprachaufnahme über das Mikrofon zu starten. Signalisiert wird dies durch ein Icon im Browser neben der Adresszeile sowie ein Dialogfenster, in welchem auch die gewählte Methode zur Spracherkennung angezeigt wird (siehe *Abbildung 4*).

Anschließend wird die Aufnahme mit der in den Attributen eingestellten Methode transkribiert. Das Ergebnis wird dann in einem Hidden Page Item gespeichert und direkt weiterverarbeitet, so dass nur das Endergebnis beziehungsweise die gewünschte Aktion für den User sichtbar ist.

Beispiel 1 – Notizenapp

Bei diesem Beispiel wird das Ergebnis der Spracheingabe direkt in eine Tabelle geschrieben. Eine Cards-Region zeigt alle bereits eingespielten Daten in einem notizentzettelähnlichen Layout (siehe *Abbildung 5*).

Ein möglicher Einsatzzweck könnte eine mobile Anwendung sein, mit der schnell und einfach Notizen mittels Spracheingabe erstellt werden können – ohne von Drittanbieterapps abhängig zu sein. Alle Daten liegen in der Oracle-Datenbank und können direkt von dort in APEX angezeigt und weiterverwendet werden.

Beispiel 2 – Formular ausfüllen

Der zweite Anwendungsfall ist das Ausfüllen von Formularen. Ziel ist es, nicht in jedes Eingabefeld navigieren und manuell die Werte eintippen zu müssen. Stattdessen sollen aus einem gesprochenen Text automatisch die erforderlichen Daten herausgezogen und in die korrekten Felder eingetragen werden.

Dafür folgt im Anschluss an die Spracherkennung noch ein weiterer Aufruf in Richtung OpenAI. Neben dem Transkript der Eingabe werden dabei noch weitere Zusatzinformationen in Form eines Prompts mitgegeben, aus dem die KI den Bezug zwischen Daten und Eingabefeldern herstellen kann. Die Bezeich-

nungen der vorhandenen Formularfelder können aus den APEX-Metadaten ausgelesen und als Liste verfügbarer Attribute an die KI übermittelt werden. Ein möglicher Prompt ist in *Abbildung 6* dargestellt.

Die KI extrahiert somit nur die Informationen aus dem Text, die auch in ein entsprechendes Feld im Formular eingefügt werden können. Damit die KI den Bezug zwischen Text und Formularfeld herstellen kann, empfiehlt es sich also, sprechende Namen für die Spalten beziehungsweise die Page Items zu wählen. Indem wir auch ein festes Rückgabeformat festlegen (in diesem Fall ein JSON-Objekt mit den Feldnamen als Key und den eingegebenen Daten als Value), können wir das Ergebnis im Anschluss generisch mit JavaScript auslesen und die einzelnen Daten in das Formular einsetzen (siehe *Abbildung 7*).

Beispiel 3 – Daten filtern

In einem letzten Beispiel soll die Steuerung von Funktionen in APEX mithilfe von Sprache getestet werden. Konkret geht es hierbei um die Filtermöglichkeit im Interactive Grid. Das generelle Vorgehen ist dabei ähnlich wie beim vorherigen Beispiel: Der User spricht die gewünschte Filteroperation (zum Beispiel: „Gib mir alle Personen, die im August Geburtstag haben.“) über das Mikrofon ein. Mithilfe von Whisper wird die Aufnahme transkribiert

The screenshot shows the Oracle APEX interface with a 'Whisper' voice input plugin. The main area displays an Interactive Grid with the following data:

First Name	Last Name	Postal Code	City	Birth Date	Phone	Company
Steve	Widerski	80110	Englewood	18.08.2110	(532) 378-8546	Heare Cosmetics
Daphne	Elfrink	03060	Nashua	14.08.2001	(746) 283-1122	Fowlston Heavy In...
Eddie	Tavernier	89011	Henderson	18.08.2060	(225) 211-2339	Grunney Group
Daniel	Gatling	88427	Mcalister	22.08.1970	(526) 547-5335	Strong Group

The 'Result' panel on the right shows the transcription: 'Gib mir alle Personen, die im August Geburtstag haben.' and the resulting SQL filter: 'WHERE EXTRACT(MONTH FROM PERS_BIRTH_DATE) = 8'.

Abbildung 8: Filtern im Interactive Grid (Quelle: Dennis Amthor)

und das Ergebnis anschließend weiter mit OpenAI verarbeitet.

Die offensichtlichste Lösung wäre es, alle Daten aus dem Grid auszulesen und zusammen mit dem Filterbefehl an die KI zu schicken. Neben datenschutztechnischen Aspekten sollte auch die Datenmenge beachtet werden, die neben einer erhöhten Verarbeitungszeit für steigende Kosten aufgrund von mehr verbrauchten Tokens bei Verwendung der API sorgt. Damit nicht jedes Mal alle Daten mitgeschickt werden müssen, bekommt die KI nur den Befehl, ein Filter-Statement für eine SQL-Abfrage zu schreiben. Dazu werden die entsprechenden Informationen mitgeliefert, damit die KI weiß, welche Spalten zum Filtern zur Verfügung stehen. Auch hier können wieder die APEX-Metadaten genutzt werden, in denen die verfügbaren Spalten des Interactive Grids aufgelistet sind. Zusammen mit der natürlichsprachlichen Filteroperation bekommen wir dann ein SQL-Statement zurück, welches mithilfe der Funktion „`apex_region.open_query_context`“ als WHERE-Clause mit der Region Source des Interactive Grids kombiniert werden kann (siehe Abbildung 8).

Fazit

Die Beispiele zeigen, wie schon mit einfachen Mitteln eine Unterstützung für Spracheingabe in konkreten Anwendungsfällen in APEX geschaffen werden kann. Der Implementierungsaufwand für die drei Methoden ist insgesamt überschaubar, da die grundsätzliche Logik gut in Plugins gekapselt und so für verschiedene Zwecke wiederverwendet werden kann.

Hinsichtlich der Bewertung der drei Methoden hat sich der direkte Aufruf der OpenAI-API als am effizientesten herausgestellt. Bei einer sehr guten Antwortzeit liefert das Modell durchgehend präzise Ergebnisse, was insbesondere bei einer Weiterverarbeitung der Ergebnisse wie in den Anwendungsbeispielen 2 und 3 sehr wichtig ist. Die Integration von Whisper Web eignet sich insbesondere dann, wenn auch andere Modelle mit eingebunden werden sollen, die zum Beispiel im Rahmen der Transformers-Bibliothek bezogen werden können. Ein weiterer Vorteil ist die Möglichkeit der lokalen Ausführung und die damit verbundene Nutzung

ohne aktive Verbindung zum Internet. Als Nachteile haben sich die langen Antwortzeiten und der hohe Speicherbedarf herausgestellt, was insbesondere bei Mobilgeräten problematisch sein könnte.

Der Datenschutz kann bei beiden Methoden als gut eingestuft werden, zumal die Daten bei Nutzung der API von OpenAI nicht zum Training verwendet werden. Wer die vollständige Kontrolle haben möchte, sollte jedoch auf das lokale Modell setzen. Für die zusätzlichen Tasks wie das Ausfüllen von Formularen oder das Filtern im Grid ist die Verwendung eines LLMs wie dem von OpenAI, welches die Anfragen verarbeiten kann, jedoch unverzichtbar.

Die Umsetzung mit der Web Speech API hat ebenfalls den Vorteil, dass diese vollständig und nativ in JavaScript implementierbar ist, sie kann jedoch aufgrund der Einschränkungen hinsichtlich der Browserunterstützung und der Datenschutzaspekte nicht für alle Anwendungsfälle geeignet sein. Auch die fehlenden Konfigurationsmöglichkeiten und Funktionen wie der Upload machen die Lösung weniger flexibel.

Ausblick

In Zeiten der zunehmenden Möglichkeiten mit KI und smarten Lösungen im Bereich der Digitalisierung könnte sich Spracherkennung auch für klassische datenbankorientierte Anwendungen lohnen. Eine entsprechende Browserunterstützung und Klärung datenschutztechnischer Bedenken vorausgesetzt, kann mit einfachen Mitteln ein Mehrwert für Userinnen und User geschaffen werden, der sich durch Usability und Effizienz bezahlt macht.

Die beiden APEX-Plugins mit den verschiedenen Methoden zur Spracherkennung sind frei auf GitHub verfügbar [6].

Quellen

- [1] OpenAI, <https://openai.com/de-DE/index/whisper/>
- [2] OpenAI GitHub, <https://github.com/openai/whisper>
- [3] OpenAI, <https://platform.openai.com/docs/pricing>
- [4] Hugging Face GitHub, <https://github.com/huggingface/transformers.js>
- [5] Mozilla, https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API

[6] GitHub, <https://github.com/damthor/speech-to-apex>

Über den Autor

Dennis Amthor arbeitet als Berater im Bereich APEX mit dem Schwerpunkt JavaScript bei Hyand. Eine seiner Hauptaufgaben ist die Entwicklung von individuellen APEX-Anwendungen, insbesondere in Bezug auf Offlinefähigkeit. Zudem ist er an der Weiterentwicklung von spezifischen APEX-Lösungen zum Thema Prozessmodellierung mit „Flows for APEX“ beteiligt.



Dennis Amthor
dennis.amthor@hyand.com



Mein persönlicher KI-Assistent – Lokal, ohne Cloud, mit Oracle APEX und Co.

Jörg Doppelreiter, solicon IT

Wer wünscht sich nicht hin und wieder Unterstützung bei der täglichen Arbeit, um zeitraubende Aufgaben schneller zu erledigen? Bei näherer Betrachtung gibt es viele Szenarien, in denen ein „Assistant“ hilfreich sein kann. In diesem Artikel konzentrieren wir uns auf zwei Szenarien:

- Das effiziente Verwalten und Zusammenfassen von E-Mails, um nichts zu übersehen.
- Das schnelle Durchsuchen einer eigenen Wissensdatenbank, die im Laufe der Zeit aufgebaut wurde.

Warum ein lokales System?

Ein lokal betriebenes System bietet nicht nur einen tieferen Einblick in die Funktionsweise von KI, sondern auch entscheidende Vorteile in Bezug auf Datenschutz und Kontrolle. Vertrauliche Informationen in E-Mails oder Knowledge-Bases werden nicht an externe Systeme übermittelt. Die Leistungsanforderungen hängen von Faktoren wie Datenmenge, Prompt-Komplexität und gewähltem Modell ab. Hier ist auch Kompromissbereitschaft gefragt, da Verarbeitungsgeschwindigkeit und Genauigkeit stark von der jeweiligen Aufgabenstellung, den Da-

tenquellen und den eingesetzten Modellen abhängen.

Aufgaben

Im Fokus dieses Artikels stehen zwei zentrale Aufgaben:

- E-Mails: Zusammenfassung und Extraktion von Aufgaben. Ziel ist es sicherzustellen, dass keine wichtigen Informationen oder To-Dos übersehen werden.
- Knowledge Base: Beantwortung von Fragen mit Hilfe eines lokalen Obsidian-Stores. Erweiterungen, wie die

Einbindung von Confluence-Seiten, werden im Abschnitt „Integrieren / Obsidian“ behandelt.

Das Setup

Für die Beispiele in diesem Artikel wird ein Macbook Pro M2 mit 32GB Ram und einer 19 Core GPU verwendet. Der Speicher (unified memory) wird von CPU als auch GPU genutzt. Die Oracle 23ai Free-Datenbank läuft in einem Podman Container, APEX in Version 24.2, und ORDS auf Versionsstand 24.3.

Installation und Konfiguration der Datenbankumgebung sind nicht Bestandteil dieses Artikels und können auf verschiedenste Arten durchgeführt werden. Hierfür können verschiedene Ansätze gewählt werden, zum Beispiel ein Pre-Built Developer Image für VirtualBox, das direkt von der Oracle-Website heruntergeladen werden kann [1].

Die Daten

Woher stammen die Daten?

- E-Mail-System: Microsoft Office 365, angebunden über die Microsoft Graph

```
# Model-Download
ollama pull mistral-nemo:latest
# Testlauf
ollama run mistral-nemo -verbose
# Standard-Werte anpassen
export OLLAMA_HOST=127.0.0.1:9090
export OLLAMA_CONTEXT_LENGTH=40000
# Service starten
ollama serve
# Service testen
curl http://localhost:9090/v1/models
# Detailinformationen
ollama show mistral-nemo
```

Listing 1: Anpassung des Standard-Ports und Kontext-Größe in Ollama

The screenshot displays the configuration interface for an AI service, organized into four main sections:

- Identification:**
 - AI Provider:** A dropdown menu set to "Open AI".
 - Name:** A text input field containing "Ollama".
 - Static ID:** A text input field containing "OLLAMA".
- Settings:**
 - Used by App Builder:** A toggle switch that is currently turned off.
 - Base URL:** A text input field containing "http://host.containers.internal:9090/v1".
- Credentials:**
 - Credential:** A dropdown menu set to "credentials_for_ollama".
 - A green "Test Connection" button is located to the right of this section.
- Advanced:**
 - AI Model:** A text input field containing "mistral-nemo".
 - A green "Additional Attributes" button is located to the right of this section.

Abbildung 1: AI-Service-Konfiguration (Quelle: Jörg Doppelreiter)

```

procedure post_auth
as
begin
    apex_util.set_session_state ('F_ACCESS_TOKEN', APEX_JSON.GET_VARCHAR2('access_token'));
    apex_util.set_session_state ('F_R_TOKEN', APEX_JSON.GET_VARCHAR2('refresh_token'));
    apex_util.set_session_state ('F_EXP_TOKEN', APEX_JSON.GET_VARCHAR2('expires_in'));
end;

```

Abbildung 2: Access Token in APEX-Authentifizierung auslesen (Quelle: Jörg Doppelreiter)

```

https://graph.microsoft.com/v1.0/me/messages?$filter=createdDateTime gt 2025-04-21T00:00:00Z and created-
DateTime lt 2025-05-21T00:00:00Z

```

Listing 2: Abfrage der E-Mails

```

declare
    l_dir_path  varchar2(4000) := '/opt/oracle/oradata/kb/knowledgebase/ai';
    l_file_name varchar2(4000) := 'prompting.md';
    l_line      varchar2(32767);
    l_file      clob;
    f_handle    utl_file.file_type;
begin
    execute immediate 'create or replace directory IMPORT_DIR as ''' || l_dir_path || ''';

    f_handle := utl_file.fopen('IMPORT_DIR', l_file_name, 'r');
    dbms_lob.createtemporary(l_file, false);

    loop
        begin
            utl_file.get_line(f_handle, l_line);
            dbms_lob.append(l_file, l_line || chr(10));
        exception
            when no_data_found then
                exit;
        end;
    end loop;

    insert into knowledge_base(fullpath, filename, file_content)
        values(l_dir_path, l_file_name, l_file);

    utl_file.fclose(f_handle);
end;
/

```

Abbildung 3: Import einer Datei aus dem lokalen Filesystem (Quelle: Jörg Doppelreiter)

```

for rec in (select id, body_content from emails_incoming )
loop
    l_summary := APEX_AI.GENERATE( p_prompt => rec.body_content
                                   , p_system_prompt => l_system_prompt
                                   , p_service_static_id => 'OLLAMA');
    update emails_incoming set ai_summary = l_summary where id = rec.id;
end loop;

```

Abbildung 4: Erstellung einer Zusammenfassung durch AI (Quelle: Jörg Doppelreiter)

```

for rec in (select D.fullpath id, C.chunk_offset pos,
               C.chunk_length siz, C.chunk_text txt
            from knowledge_base D,
                 VECTOR_CHUNKS(D.file_content
                               BY words MAX 200
                               OVERLAP 10 SPLIT BY recursively
                               LANGUAGE american NORMALIZE all) C
            order by fullpath, chunk_offset)
loop
  l_vector := apex_ai.get_vector_embeddings(
    p_value          => rec.id||' '||rec.txt,
    p_service_static_id => 'EMBED' );

  insert into knowledge_base_chunks (fullpath, pos, siz, txt, vct)
  values(rec.id, rec.pos, rec.siz, rec.txt, l_vector );

end loop;

```

Abbildung 5: Erstellung Chunks und Vector (Quelle: Jörg Doppelreiter)

System Prompt ?

APEX Assistant

- 1 Du bist der operative Assistent von Jörg Doppelreiter.
- 2 Deine Aufgabe ist es ihm bei der Durchsicht seiner e-Mails zu helfen um arbeitsre
- 3
- 4 Lese dafür alle emails durch und erstelle eine Übersicht der relevanten Inhalte.
- 5 Ignoriere Werbung oder informative E-Mails ohne Arbeitsaufträge.
- 6
- 7 Nur Inhalte, die Aufgaben enthalten sollen herangezogen werden.
- 8 Erstelle am Ende eine Liste aller relevanter Inhalte in markdown Format.
- 9 Output Format:

Abbildung 6: System Prompt E-Mail-Zusammenfassung (Quelle: Jörg Doppelreiter)

API. Für die Authentifizierung wird eine Azure-Entra-ID-Applikation verwendet.

- Knowledge Base: Ein lokaler Obsidian Store, der Markdown-Dateien verwaltet.

- Ollama: Open-Source (MIT-Lizenz) [2]
- LM Studio: Für private und innerbetriebliche Nutzung (Stand Juli 2025) kostenlos, aber nicht Open-Source [3]

Das lokale LLM

Der Kern des Systems ist ein lokales LLM (Large Language Model). Es gibt viele Anbieter, sowohl Open-Source als auch kommerzielle. Für Entwicklungs- und Testumgebungen sind manche Modelle kostenlos, für die Produktion ist jedoch meist eine Lizenz nötig.

Für die Beispiele in diesem Artikel verwenden wir Ollama. Als Alternative sei LM Studio erwähnt, das durch eine Desktop-App, einfache Bedienung und integriertes Logging einen guten Einstieg bietet.

Die Kommunikation mit dem LLM erfolgt über REST-APIs, idealerweise über ein Interface, das der OpenAI API-Definition entspricht, da dieses nahtlos in APEX eingebunden werden kann.

Ollama

Auf der Website können die Installationsroutinen je nach Betriebssystem (macOS, Linux, Windows) bezogen werden.

Nach der Installation geht es über das Terminal weiter, alle Befehle und Erklärungen finden sich ebenso in *Listing 1*:

- Model Download: In der Model Library unter <https://ollama.com/library> finden sich alle verfügbaren LLMs. Für das kommende Beispiel wählen wir das Model „mistral-nemo“ und laden es über die integrierte Download-Funktionalität des CLI herunter: `ollama pull <model>`
- Testlauf: `ollama run <model> --verbose` Durch die Angabe des Parameters „--verbose“ werden am Ende jeder Antwort Performance-Statistiken ausgegeben. Diese können genutzt werden, um eine Einschätzung zur Leistungsfähigkeit des Systems zu treffen. `/bye` beendet den Chat-Modus wieder.

Vor dem Start des Service sollten noch Anpassungen an den Standardeinstellungen vorgenommen werden.

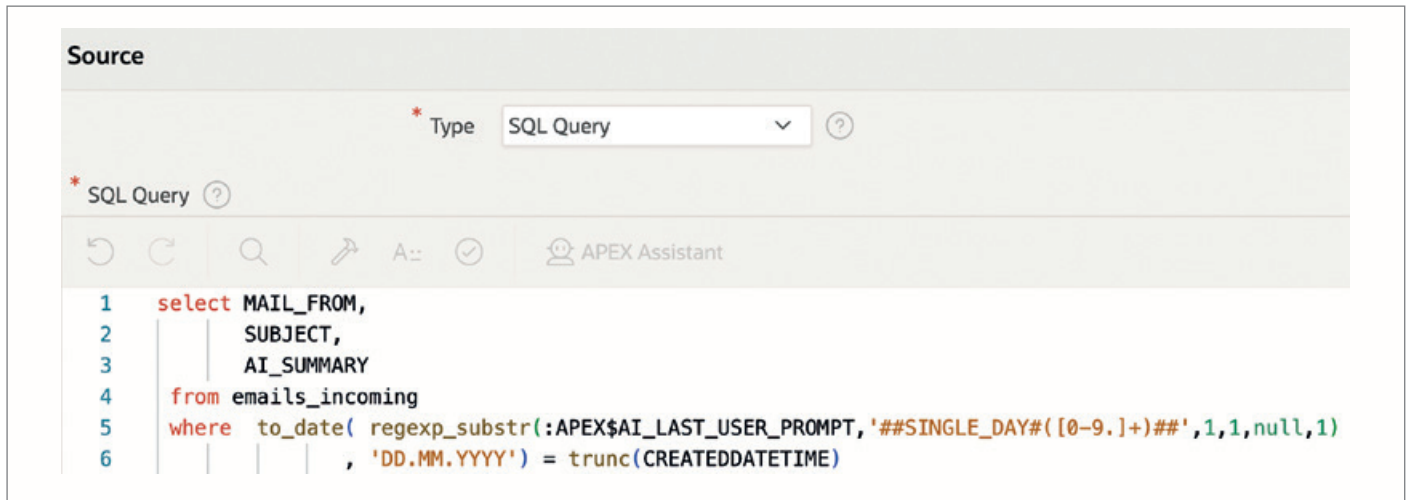


Abbildung 7: Auswertung des Datum-Platzhalters in RAG-Source (Quelle: Jörg Doppelreiter)

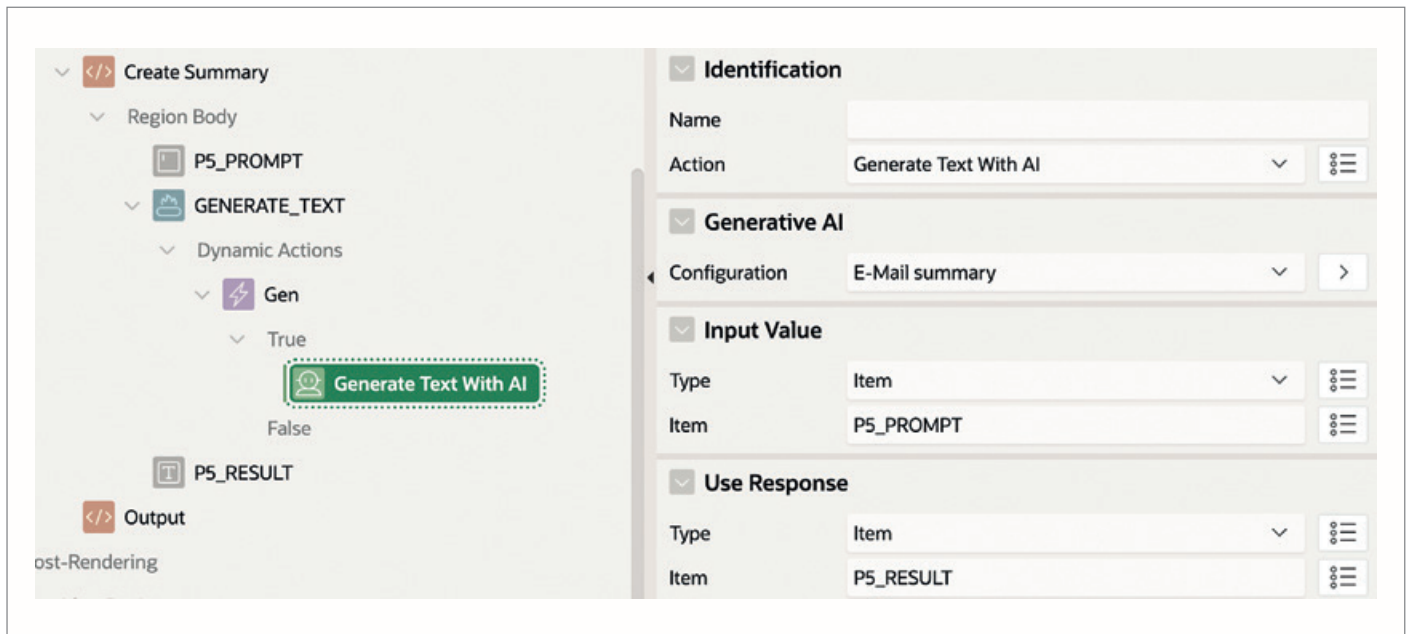


Abbildung 8: APEX-Applikationsseite-Definition (Quelle: Jörg Doppelreiter)

- Default Port für den lokalen Service ist 11434. Eine Anpassung des Ports ist vor dem Starten des Service durch Setzen der Umgebungsvariable OLLAMA_HOST möglich.
- Kontext-Größe: Standard ist 4096 Tokens. Um diese zu erhöhen, bedarf es der Umgebungsvariable OLLAMA_CONTEXT_LENGTH.

Die Kontext-Größe hat Auswirkungen auf den Speicherbedarf und die Stabilität des Systems. Es empfiehlt sich, mit kleineren Kontextgrößen zu beginnen und diese schrittweise zu erhöhen. Genaue Informationen, wie die Anzahl der Parameter oder maximale Kontext-Größen erhält man durch den Befehl: `ollama show <model>`.

APEX-Anbindung

Nach der Einrichtung des LLMs erfolgt die Integration in APEX über Generative AI Services (Workspace Utilities):

- AI Provider: Auswahl der Inference API, in diesem Fall „Open AI“
- Name: Vergabe eines eindeutigen Namens für die Konfiguration
- Static ID: Erforderlich bei Nutzung durch APEX_AI
- Base URL: URL zum Ansprechen des AI-Systems. Hier ist zu beachten: Wenn die Datenbank in einer Container-Architektur, wie beispielsweise podman, betrieben wird, muss der Host aus dem Container erreichbar sein. Pod-

man bietet hier „host.containers.internal“ an.

- Credential: Die Eingabe eines API-Keys ist erforderlich, wird aber vom lokalen AI-System nicht überprüft. Bei Verwendung eines bestehenden Credentials ist sicherzustellen, dass das genutzte Credential entweder keinen Eintrag unter dem Parameter „Valid for URLs“ besitzt oder die genutzte Base URL eingetragen wird. Andernfalls lehnt APEX die Verbindung ab.
- AI Model: Jedes Model verfügt über einen entsprechenden Identifier, der im lokalen System ausgelesen werden kann (.../v1/models). Der Name des gewünschten Models kann hier eingetragen werden.

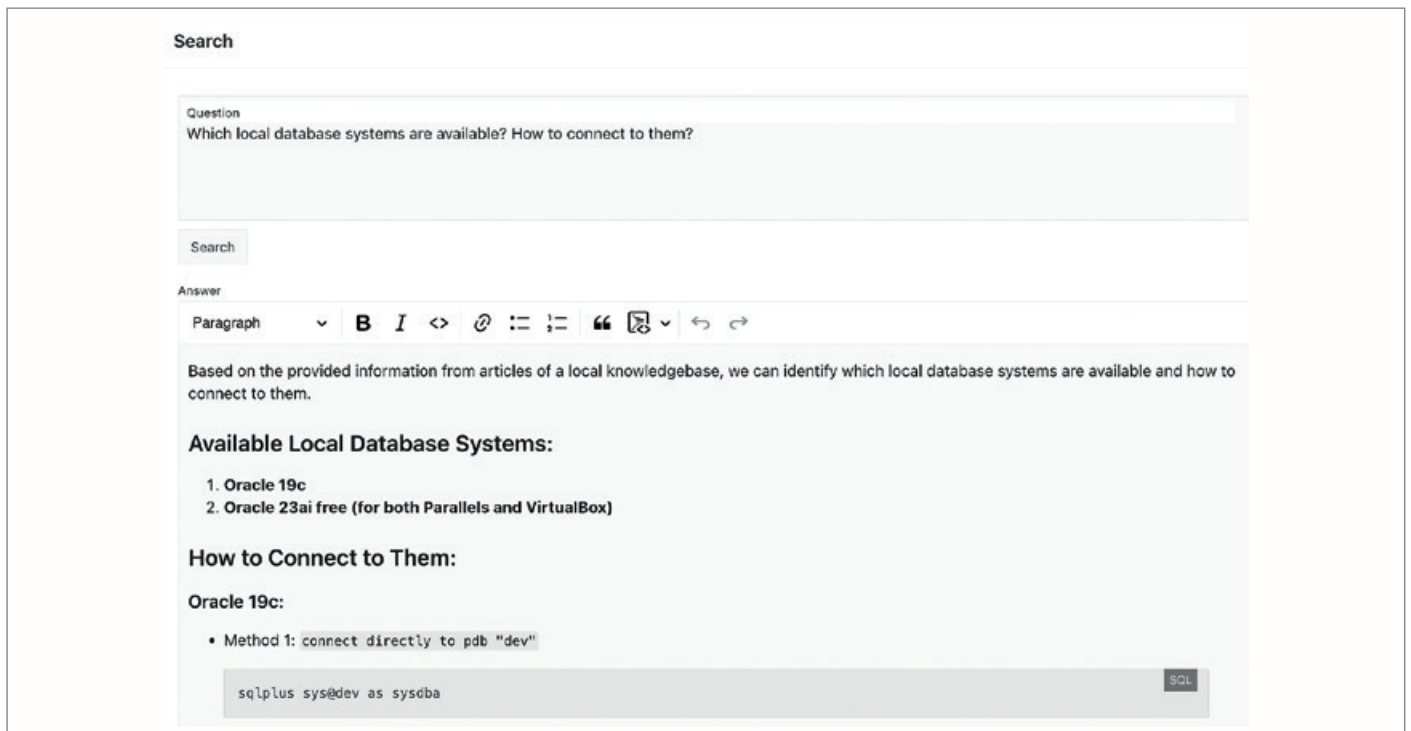


Abbildung 11: Beispiel-Abfrage der Knowledge-Base (Quelle: Jörg Doppelreiter)

Azure erlaubt nur bei „localhost“ die Nutzung des HTTP-Protokolls

- Client Secret
- API Permissions: Für den Zugriff auf die Ressourcen (E-Mails) des angemeldeten Benutzers werden „Delegated Privs“ benötigt.

Folgende Informationen sind für die Einrichtung in APEX erforderlich:

- Authorization Endpoint URL: [https://login.microsoftonline.com/\[TENANT_ID\]/oauth2/v2.0/authorize](https://login.microsoftonline.com/[TENANT_ID]/oauth2/v2.0/authorize)
- Token Endpoint URL: [https://login.microsoftonline.com/\[TENANT_ID\]/oauth2/v2.0/token](https://login.microsoftonline.com/[TENANT_ID]/oauth2/v2.0/token)
- User Info Endpoint URL: <https://graph.microsoft.com/v1.0/me>
- Scope: unter Scope müssen alle notwendigen Privilegien (delegated privs) genannt werden: User.Read,Mail.Read
- Username: userPrincipalName

Innerhalb der Post-Auth-Prozedur der APEX-Authentifizierung kann der Access Token ausgelesen werden (siehe Abbildung 2).

Der Access Token wird für Zugriffe auf die REST-API verwendet und muss im Authorization Header mitgegeben werden. Das Auslesen der E-Mail-Daten erfolgt über die REST-API Schnittstelle <https://>

graph.microsoft.com/v1.0/me/messages. Durch Angabe eines Filter-Strings kann der Zeitbereich eingeschränkt werden (siehe Listing 2).

Aus der Antwort werden für die weitere Bearbeitung folgende Datenfelder entnommen (JSON Path):

- id: Eindeutiger Schlüssel der E-Mail, kann in der weiteren Folge für Abgleiche genutzt werden, um doppelte Importe zu vermeiden.
- createdDateTime: Erstellungsdatum
- subject: Betreff
- from.emailAddress.address: E-Mail-Absender
- body.content: Vollständiger Inhalt der E-Mails (inklusive Formatierungsanweisungen)
- bodyPreview: Voransicht des Inhalts

Obsidian

Obsidian ist ein lokal arbeitendes Notiz- und Wissensmanagement-Tool, das auf Markdown-Dateien basiert. Informationen werden in Ordnern und Text-Dateien mit der Endung `.md` abgelegt. Der Import in die Datenbank kann mit verschiedenen Methoden erfolgen, vom Zugriff über das Filesystem mittels Oracle Directory bis zum Upload eines ZIP-Files

in der APEX-Applikation mit entsprechender Import-Logik.

Für die Aufbereitung der Daten müssen die einzelnen Dokumente innerhalb einer Tabelle mit der Information des Speicherorts sowie des Filenamens abgelegt sein.

Abbildung 3 zeigt ein Beispiel eines Datei-Imports.

Aufbereitung

Die Aufbereitung reduziert den benötigten Kontext und minimiert Risiken wie Halluzinationen oder Fehlinterpretationen. Die maximale Anzahl an verarbeitbaren Tokens durch das LLM begrenzt die Datenmengen, die mit einer Anfrage verarbeitet werden können. Insbesondere bei der Nutzung von E-Mails erreicht man hier schnell die Grenzen des Machbaren.

Summary-Erstellung für E-Mails (mittels AI)

E-Mails enthalten oft HTML und CSS, die unnötig Tokens verbrauchen. Mit dem lokalen LLM können E-Mails vorverarbeitet und zusammengefasst werden. Die „AI-Summary“ wird anschließend in der Datenbank abgelegt (siehe Abbildung 4).

Hier wird im Folgenden ein Beispiel einer Erstellung einer AI-Summary einer E-Mail aufgeführt:

- Systemprompt: „Erstelle für die übertragene E-Mail eine Kurzzusammenfassung der Inhalte. Untersuche zudem die Inhalte nach Aufgaben für "Jörg Doppelreiter". Die Ausgabe soll als Aufzählungspunkte in die Kategorien "Inhalte" und "Aufgaben" unterteilt werden.

Ausgabe-Format:

Inhalte

- Inhalt 1
- Inhalt 2

Aufgaben

- Aufgabe 1
- Aufgabe 2

Das Ausgabeformat ist unbedingt einzuhalten. Es sind keine zusätzlichen Inhalte und Formatierungen erlaubt!

- Prozess: Erstellung der Zusammenfassung durch APEX_AI.GENERATE. Dabei wird der zuvor eingerichtete AI-Service genutzt.

Vektorisierung Obsidian

Die Daten der Knowledge-Base werden vor der Nutzung vektorisiert. Über die ORACLE_CHUNKS-Funktion unterteilen wir die Artikel in einzelne Textblöcke. Für die Erstellung eines Vectors wird ein Embedding Model unseres lokalen AI-Setups

genutzt (APEX_AI.GET_VECTOR_EMBEDDINGS) (siehe Abbildung 5). Ein entsprechendes Modell muss zuvor für Ollama heruntergeladen und als AI Service („EMBED“) in APEX hinterlegt werden.

Die Artikel befinden sich in der Tabelle KNOWLEDGE_BASE. Durch den Einsatz der VECTOR_CHUNKS-Funktion werden Textblöcke generiert. Anzugeben ist unter anderem die Anzahl der Wörter, ob und wie viele Worte dabei überlappen sollen und die Art der Split-Methode. Es empfiehlt sich, das System so zu gestalten, dass diese Parameter zu einem späteren Zeitpunkt leicht adaptiert werden können. Als Alternative zum Aufruf des APEX_AI Packages kann ein ONNX-Model in der Datenbank geladen werden, um die Vektorisierung durchzuführen.

Integrieren

Nun stehen die Daten für die Abfrage der Use Cases in der Datenbank bereit.

Mit APEX 24.2 wurden die sogenannten „AI Configurations“ eingeführt. Diese vereinfachen die Integration externer Daten in APEX mithilfe von RAG Sources (Retrieval Augmented Generation).

RAG Sources ermöglichen die Definition von Datenquellen (SQL-Query, PL/SQL-Funktion, statische Daten) und optional eine Beschreibung. Diese Daten werden bei der Anfrage automatisch in

ein CSV-Format konvertiert und in die Prompt-Anfrage des LLMs eingebettet.

Eine AI Configuration umfasst:

- Einen System-Prompt mit Anweisungen
- Eine oder mehrere RAG Sources, die dynamisch ausgewertet werden.
- Eine Server-Side Condition je Quelle, um diese nur bei Bedarf einzubinden.

So lassen sich Systemprompts und Datenquellen modularisieren und für verschiedene Anwendungsfälle konfigurieren.

Integration E-Mail-Summary

Der erste AI Use Case besteht in der Zusammenfassung von E-Mails und der Extraktion von Aufgaben. Zu diesem Zweck wird eine AI Configuration mit dem Namen „E-Mail-Summary“ erstellt.

- System Prompt: Enthält Anweisungen zur Erstellung der Zusammenfassungen und Aufgabenlisten. Mit der Angabe spezifischer Output-Formate (Markdown) lässt sich die Konsistenz verbessern (siehe Abbildung 6).
- RAG Source: Diese definiert, welche E-Mails aus der Datenbank gelesen und dem LLM bereitgestellt werden. Der Platzhalter wird dynamisch ersetzt, die Beschreibung der Datenquelle erklärt dem LLM die Bedeutung der Daten.

Sources Used:

1. **Oracle 23ai Free with details** at KnowledgeBase/KnowledgeBase/Mac
2. **Administrative SQL for OracleDB** at KnowledgeBase/OracleDB/AdminSQL.md
3. **Configuration of Node.js Server (pdfmake.js)** at:

- TECAPEX/pdfmake.js Server-seitig als Service konfigurieren
- References to [Stack Overflow](#)

```
- **Oracle 19c** is installed using Parallels with the following configurations:
- **Port**:` 8080`
- `sqlplus sys@dev as sysdba` for direct connection to PDB.

- **Oracle 23ai free:** available via Parallels and VirtualBox with:
- **Port for DB**:` 1522`
- **Port for Apex**:` 8081`

- Example setup commands and connection steps provided.
```

- **Node.js Server setup script** explained in context of given articles.

Abbildung 12: Quellenausgabe Knowledge-Base Use Case (Quelle: Jörg Doppelreiter)

- Platzhalter für Datum (User-Prompt): `##SINGLE_DAY#DD.MM.YYYY##`
Dieser Marker kann innerhalb einer User-Eingabe genutzt und in der RAG Source ausgewertet werden. Er dient sowohl der Auswahl der Quelle als auch der Einschränkung innerhalb der SQL-Abfrage (siehe Abbildung 7).

Wir erstellen nun eine APEX-Seite:

- Eingabefeld für den Userprompt
- Button für die Ausführung
- Ausgabefeld (Rich Text Editor mit Markdown-Format) für die Darstellung der Antwort des LLMs

Die Verarbeitung der Anfrage erfolgt durch die mit APEX 24.2 eingeführte Action: „Generate Text With AI“ (siehe Abbildung 8).

- Auswahl der AI Configuration
- Zuweisung der Input- und Output-Felder

Abbildung 9 zeigt eine Demonstration der implementierten Logik mit Datums-Platzhalter und Anweisung zur Maskierung der Daten.

Integration Obsidian

Der zweite Use-Case nutzt RAG Sources, um Daten aus den aufbereiteten Obsidian-Einträgen bereitzustellen (siehe Abbildung 10). Es wird für diesen Anwendungsfall eine weitere AI Configuration „Knowledge Base“ erstellt. Innerhalb jeder RAG Source werden mehrere Schritte im Zuge einer Abfrage ausgeführt:

1. Vektorisierung der Nutzer-Anfrage
2. Distance Search zur Ermittlung ähnlicher Inhalte
3. Integration der Ergebnisse in den Prompt der Anfrage

Für die Vektorisierung der Anfrage wird die Embedding-Funktion des lokalen LLMs genutzt. Die Distance Search sucht die 10 zutreffendsten Vektoren. Über den Parameter „Maximum Tokens“ wird eine Grenze für die Datenmenge definiert. Hierbei handelt es sich um einen Schätzwert der erwarteten Token-Anzahl. Überschreitet eine Zeile aus der Abfrage diese

Grenze wird die Verarbeitung der Quelle an dieser Stelle gestoppt. Anweisungen im System Prompt der AI Configuration spezifizieren:

- Aufgabenbeschreibung: „Du bist ein Experte in Datenanalyse und kontextbezogener Suche. Suche in den übermittelten Datenquellen nach der Antwort und füge ergänzende Informationen hinzu, wenn es zu besserem Verständnis beiträgt.“
- Ausgabeformat: „Formatiere die Ausgabe im Markdown-Format.“
- Quellen-Angabe: „Füge am Ende der Antwort eine Aufzählungsliste aller verwendeten Quellen hinzu.“

Die Erstellung der APEX-Seite erfolgt analog dem Use Case der E-Mail-Summary. Dazu wird eine Seite bestehend aus Ein- und Ausgabefeld (Rich Text Editor mit Markdown-Format) sowie einem Button benötigt. Die Verarbeitung erfolgt über die Dynamic Action „Generate Text With AI“ und Auswahl der vorbereiteten AI Configuration.

Alle Systemanweisungen zur Ausgabe sind in der Konfiguration hinterlegt (siehe Abbildung 11).

Durch die Systemprompt-Anweisung zur Quellen-Angabe können auch die genutzten Inhalte verifiziert werden. In diesem Beispiel wurde neben Obsidian eine Confluence-Datenquelle (TECAPEX/pdf-make.js) genutzt (siehe Abbildung 12).

Fazit

Das Ziel dieses Artikels war es, zu zeigen, wie man ein lokales AI-Setup mit Oracle APEX umsetzt – sowohl auf dem Arbeitsplatz als auch in Unternehmensinfrastrukturen.

Die Anforderungen an ein LLM sind meist:

- Qualitativ hochwertige Antworten
- Schnelle Reaktionszeiten

Je nach Modellgröße, Hardware und Datenqualität sind Kompromisse notwendig, Antwortzeiten hängen direkt von den verarbeiteten Daten ab. Ebenso lässt sich die Frage nach dem besten System und Modell nicht pauschal beantworten. Je nach Use-Case und Lizenzbestimmungen

können sich für unterschiedliche Aufgaben verschiedene Systeme eignen.

Vorteile lokaler Lösungen sind vor allem:

- Volle Datenhoheit
- Einfache Anpassung und Wechsel von Modellen
- Einsatz in Entwicklungsumgebungen
- Direkte Integration in APEX durch AI Configurations und RAG Sources.

Eine gute Datenaufbereitung ist, besonders in lokalen Systemen, ein entscheidender Faktor für Qualität und Effizienz.

Über den Autor

Jörg Doppelreiter ist Senior Consultant und APEX Solution Expert für die solicon IT GmbH. Hauptaufgaben sind die Konzeption und Umsetzung von Lösungen im Umfeld von Oracle-APEX-Applikationen.



Jörg Doppelreiter
joerg.doppelreiter@solicon-it.com

CLOUD NATIVE FESTIVAL

im Heide Park Soltau

SUPER-SAVER

Bis 31.10.2025



19. – 22.
MAI
2026



CloudLand
www.cloudland.org

#CLOUDLAND2026



Bidirektionaler Datenfluss zwischen Oracle APEX und Desktop-Anwendungen

Sebastian Reinhard, merlin.zwo InfoDesign

Der Artikel zeigt, wie Oracle APEX mithilfe einer schlanken .NET-Brücke und SignalR ein nahtloser Dateiaustausch mit Desktop-Programmen ermöglicht wird. Zudem gehen wir auf die Hürden ein, welche eine Verlagerung der Dokumente in ein zentrales webbasiertes Dokumentenmanagement mit sich bringt.

Seit Jahren verlagern sich Anwendungen vom klassischen Desktop hin ins Web. Der Charme dieser Architektur liegt auf der Hand: Eine zentrale Deployment-Stelle sorgt dafür, dass alle Benutzer automatisch mit der neuesten Version arbeiten; zugleich ist die Anwendung

plattformunabhängig und läuft auf PC, Mac oder Smartphone gleichermaßen. Doch wo Licht ist, ist bekanntlich auch Schatten: Aus Sicherheitsgründen werden Web-Applikationen in einer Sandbox des Browsers ausgeführt. Das erschwert den direkten Zugriff auf lokale Ressourcen

– insbesondere auf Dateien, die Anwender in ihren gewohnten Desktop-Programmen bearbeiten möchten.

Sobald die Dateien zentral in der Datenbank liegen und die Anwender sie über den Browser herunterladen, lokal bearbeiten und erneut hochladen müs-

sen, kippt die anfängliche Begeisterung schnell in Ärger um. Zu viele Klicks, Versionskonflikte und Medienbrüche bremsen Prozesse aus – insbesondere, wenn täglich hunderte Office-Dokumente und PDFs bearbeitet werden müssen. Unser Kunde, ein mittelständischer Betrieb, stellte deshalb die scheinbar einfache Frage:

„Warum kann ich mein Dokument nicht einfach direkt aus dem ERP heraus in Word öffnen, speichern und zurückschreiben – ohne Umweg über Download-Dialoge?“

Wir nahmen die technische Herausforderung, die in der Frage schlummerte, an und schufen im Ergebnis eine Brücke zwischen APEX und dem Desktop des Nutzers, welche die benötigten Dateien lokal zu Verfügung stellt, überwacht und mit dem Server synchronisiert.

Vom Netzwerkshare zur zentralen Dokumentenablage in der Datenbank

Die Historie: Das alte ERP basierte auf Oracle Forms; sämtliche kundenbezogenen Dokumente lagen in einem Samba-Share, der auf jedem Arbeitsplatzrechner unter demselben Laufwerksbuchstaben eingebunden war. Jeder Kunde besaß einen Ordner, dessen Name der Kundennummer entsprach. Im Tagesgeschäft wurden jedoch häufig Dateien in falsche Ordner kopiert, versehentlich lokal verschoben oder gleich ganze Ordner verschleppt. Dadurch gingen Unterlagen verloren, waren für Kollegen nicht auffindbar oder existierten mehrfach in unterschiedlichen Versionen.

Ausgangslage: Nach der Migration des ERP-Systems lagen alle Dokumente in BLOB-Spalten der Oracle-Datenbank. Was als zentraler Single-Point-of-Truth geplant war, entpuppte sich für die Anwender als Klick-Orgie: herunterladen → bearbeiten → gegebenenfalls zurück zum Kunden navigieren, da eine andere Anfrage dazwischen kam → hochladen → Datei schließen. Schon drei Office-Formate (.docx, .xlsx, .pdf) sorgten für Ärger – und es war absehbar, dass noch weitere Dateiformate folgen könnten, zum Beispiel CAD. (siehe Tabelle 1)

Keine Alternative überzeugte vollständig – entweder fehlte Funktionsumfang,

Kosten/Nutzen standen in keinem Verhältnis oder der Kunde wollte partout kein Abo-Modell. Vor allem aber kam für den Kunden Netzwerkshare nicht mehr in Frage. Also entwickelten wir eine eigene Brückenlösung.

Architektur im Überblick

Die Architektur basiert auf drei ineinandergreifenden Bausteinen (siehe Abbildung 1). Im Zentrum steht die Oracle-Datenbank, in der sämtliche Dokumente abgelegt sind. Zwischen der Datenbank und den Endgeräten vermittelt ein schlanker Bridgeserver, der Anfragen entgegennimmt, Sitzungskontexte verwaltet und die Anfragen an die passenden Clients verteilt. Auf jedem Arbeitsplatzrechner läuft dazu ein Hintergrunddienst im System-Tray, der eine ständige Verbindung zum Bridgeserver hält.

Möchte der Anwender eine Datei bearbeiten, benachrichtigt die Datenbank über den Bridgeserver den Hintergrunddienst und dieser lädt das Dokument in ein temporäres Verzeichnis und öffnet es unmittelbar im zugeordneten Desktop-Programm – egal ob Textverarbeitung, Tabellenkalkulation oder Bildbearbeitung. Eine permanente Überwachung der Datei stellt sicher, dass jede Änderung zuverlässig erkannt und umgehend an die zentrale Ablage zurückgespielt wird, ohne dass der Benutzer zusätzliche Schritte ausführen muss.

Nachdem nun grob klar ist, wie eine Datei ihren Weg vom Server zum Arbeitsplatz findet, stellt sich die nächste Frage: Woher weiß die APEX-Anwendung überhaupt, welcher Benutzer hinter welchem lokalen Client steckt?

Die Antwort findet sich in Abbildung 2 – über den Tray Client kann sich der Anwender anmelden. Die Authentifizierung findet über die Active Directory Zugangsdaten beziehungsweise die Auth-API der Datenbank (Rest Data Service) statt. Nach erfolgreicher Anmeldung stellt die Auth API ein JWT Sitzungstoken sowie eine zufällig generierte, eindeutige Client ID aus. Beides wird serverseitig hinterlegt und parallel an den Client zurückgegeben. Ab diesem Moment kann der APEX-Benutzer seinem Arbeitsplatz zugeordnet werden.

Technik – APEX

In APEX werden die Dokumente in Reports angezeigt (siehe Abbildung 3). Für jedes Dokument gibt es eine Schaltfläche (Monitor-Icon), um das Dokument lokal zu bearbeiten. Beim Klick stoßen wir per JavaScript ein AJAX-Callback (PL/SQL-Prozess) an, der serverseitig für den angemeldeten Benutzer die zugehörige Client-ID ermittelt und ein kryptographisch starkes Einmaltoken („One-Time-Token“) generiert. Dieses Token wird in einer dedizierten Session-Tabelle mit extrem kurzer TTL (Time-To-Live, zum Beispiel 30 Sekunden) persistiert, sodass es

Evaluierte Optionen

Option	Vorteile	Nachteile
Office 365 + SharePoint	Aktuellstes Office, Kollaboration	Abo-Modell, Daten in der Cloud, nur Office-Formate, Mapping ins ERP aufwändig
Web-Editoren	Kein lokales Office nötig	Lizenzkosten, Funktionslücken, Schulungsaufwand, ein Editor pro Dateityp
Rückkehr zum Netzwerk-Share	Geringe Komplexität, kein Abo	Chaotische Ordnerstrukturen, Indexierung nötig

Tabelle 1.

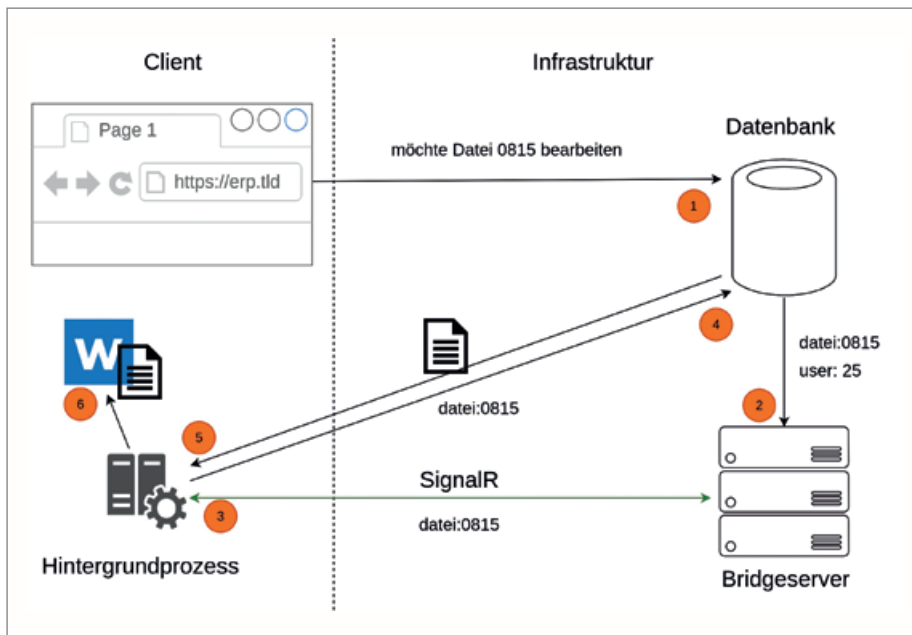


Abbildung 1: Übersicht über den Prozessablauf (Quelle: Sebastian Reinhard)

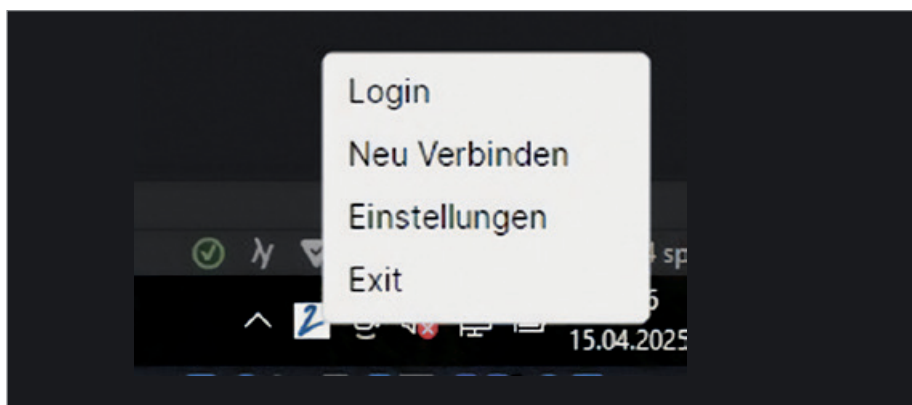


Abbildung 2: Screenshot der Menüpunkte über das Tray-Icon (Quelle: Sebastian Reinhard)

sofort abläuft, falls es nicht genutzt wird. Anschließend initiiert der PL/SQL-Prozess einen REST-Request über apex_rest_request an den Bridge-Server. Hierbei werden alle relevanten Parameter – Client-ID, Einmaltoken, Datei-ID übergeben. Der Bridge-Server validiert daraufhin das Token und leitet die Anforderung über SignalR an den richtigen Client weiter.

Technik – Datenbank

Datenbankseitig stellen wir mittels RESTful Services folgende Endpunkte für den Austausch bereit:

- /login: Authentifizierung, gibt ein JSON Web Token (JWT) und eine eindeutige Client-ID zurück, die den Benutzer identifiziert.

- /token: Endpoint zur Token-Erneuerung oder zur Ausgabe eines neuen Access-Tokens.
- /validate: Über diesen Endpunkt kann der Bridgeserver das Einmaltoken validieren. Das Token ist danach ungültig.
- /file/{id} (GET/PUT): Download oder Upload einer Datei über die ID, mit Validierung des JWT und Prüfung auf Zugriffsrechte.
- /filedetails/{id}: Liefert Metadaten wie MIME-Type und MD5-Hash der Datei, dient unter anderem der Integritätsprüfung und zur Konflikterkennung vor dem Upload.

Technik – Bridgeserver

Um zu erklären, warum wir überhaupt einen Bridgeserver brauchen, machen wir

noch einen kleinen Ausflug in die Theorie von HTTP. Das klassische Web Protokoll HTTP arbeitet nach dem Prinzip „Der Client fragt, der Server antwortet.“ Sobald die Antwort versendet ist, gilt die Verbindung als beendet; der Server kann keine Nachrichten mehr initiieren. Für unser Szenario – in dem der Server eine Datei spontan an den Client „schieben“ soll – reicht dieses Pull-Modell nicht aus. Deshalb richtet der Tray Dienst unmittelbar nach der Anmeldung eine dauerhafte rückkanalfähige Verbindung zum Bridge Server ein. Technisch basiert dieser Kanal auf einem WebSocket ähnlichen Verfahren mit dem Namen SignalR, das durch Firewalls hindurch funktioniert und lediglich einen einzigen Port benötigt. Der Bridgeserver selbst basiert auf ASP.NET 8 mit einer Minimal-API und dem SignalR Hub. Da hier nur flüchtige Daten anfallen, halten wir die Mappings ClientID → ConnectionID in einem In-Memory-Dictionary, um offene Verbindungen effizient zu adressieren. Ein Persistieren ist nicht notwendig, da die SignalR-Verbindungen nach einem Neustart weg wären beziehungsweise neu aufgebaut werden. Eingehende REST-API-Aufrufe werden an den SignalR Hub weitergeleitet und die FileIDs an die Clients gesendet. Zum Schutz der API werden die Einmaltokens gegen die Datenbank validiert, um Replay- und Injection-Angriffe zu verhindern.

Technik – Client

Der Client-Daemon wurde in .NET 8 entwickelt und setzt Avalonia UI ein, um eine vollständig plattformübergreifende grafische Oberfläche für Windows, macOS und Linux bereitzustellen. Das Tray-Icon ist eine eingebaute Funktionalität von Avalonia und ermöglicht eine plattformübergreifende Integration in die System-Tray-Bereiche der jeweiligen Betriebssysteme. Systemnahe Funktionen wie File-Watching, Auto-Start und Prozessüberwachung sind über abstrahierte Interfaces gekapselt und werden per Dependency Injection injiziert, sodass der Code unabhängig von der Zielpattform bleibt. Über die Konfigurationsoberfläche können Parameter wie die Server-URL, Authentifizierungs-API-Keys und Mapping-Tabellen von MimeTypes sowie Dateiendungen zu Zielanwendungen verwaltet werden.

Dokument-Anzeige						
<input type="checkbox"/>					Dateiname	Aufnahmedatum
<input type="checkbox"/>					2025-04-14_Kaufbeleg_2411155276.pdf	-
<input type="checkbox"/>					2025-04-14_Mitarbeiterliste.xlsx	-
<input type="checkbox"/>					2025-04-14_Praesentationsnotizen.docx	-
<input type="checkbox"/>					2025-04-14_Entwurfskonzept.docx	-
<input type="checkbox"/>					2025-04-14_Fehlerbericht.docx	-
<input type="checkbox"/>					2025-04-14_Kundenvorstellung.docx	-
<input type="checkbox"/>					2025-04-14_Lieferantenebersicht.xlsx	-
<input type="checkbox"/>					2025-04-14_Meeting-Protokoll.docx	-
<input type="checkbox"/>					2025-04-14_Aufgabenebersicht.xlsx	-
<input type="checkbox"/>					2025-04-14_Budgetplanung.xlsx	-
<input type="checkbox"/>					2025-04-14_Anforderungsdokument.docx	-

Abbildung 3: Screenshot der Dokumentenverwaltung (Quelle: Sebastian Reinhard)

Da über den Bridgeserver nur die FileID übertragen wird, lädt der Client die Datei mit dem dazugehörigen Hash über die API von der Datenbank herunter und legt die Datei in einem temporären Verzeichnis ab. Der Hash und weitere Metadaten werden in SQLite gespeichert. Vor dem Download werden noch Anfragen zu Dateien mit gewissen Dateitypen wie beispielsweise .exe, .bat oder etc. gefiltert und geloggt. Erst danach werden die Dateien gemäß des Mappings geöffnet.

Ein dedizierter Hintergrundthread setzt den .NET FileSystemWatcher ein, um das temporäre Arbeitsverzeichnis rekursiv zu überwachen. Dateioperationen wie Create und Modify werden in Echtzeit erkannt. Bei jedem Auslösen des Modify-Events wird ein Hash gebildet und dieser mit dem vorherigen Hash verglichen. Erst wenn sich der Hash unterscheidet, gilt eine Datei als geändert. Das hilft uns dabei, Änderungen zuverlässig zu erkennen, verhindert eine Eventflut durch Debouncing (typischerweise bei Office-Speichervorgängen) für die folgende Verarbeitung und hilft beim Erkennen von Konflikten bei der parallelen Bearbeitung einer Datei von mehreren Mitarbeitern.

Sollte eine Änderung an der Datei erkannt werden, wird die Datei über die API automatisch an die Datenbank übertragen. Neben der Datei wird auch der ursprüngliche Hash übertragen, um festzu-

stellen, ob sich die Datei zwischenzeitlich geändert hat. Sollte dies der Fall sein, wird der Nutzer informiert, wer die Datei geändert hat und wann. Er hat dann die Entscheidung, ob die Datei überschrieben werden soll oder nicht.

Beim Starten des Tray-Clients löscht ein automatisches Cleanup-Verfahren temporäre Dateien anhand der Metadaten, wenn diese mehr als 24 Stunden nicht verändert wurden. Für die lokale Persistenz verwendet der Daemon eine SQLite Datenbank mit Write Ahead Logging (WAL), um parallele Zugriffe von der GUI und Hintergrundtasks zu ermöglichen.

Für Logging und Telemetrie kommt Serilog mit strukturiertem JSON Output zum Einsatz. Es gibt Unterstützung für Rolling Files. Das Benachrichtigungssystem informiert den Nutzer, wenn Dateien erfolgreich hochgeladen wurden, und stellt eine Bestätigungsbox bereit, falls es Konflikte gibt, damit der Benutzer über das weitere Vorgehen entscheiden kann. Das Benachrichtigungssystem nutzt plattformsspezifische APIs wie Windows Toast Notifications.

Fazit

Die entwickelte Brückenlösung beweist, dass sich durch eine intelligente Architek-

tur nicht nur technische Hürden überwinden, sondern auch handfeste betriebswirtschaftliche Vorteile erzielen lassen. Prozesse werden spürbar beschleunigt, da lästige Download- und Upload-Schritte entfallen, und die Gefahr von Versionskonflikten wird erheblich reduziert. Anwender arbeiten weiterhin in ihrer gewohnten Umgebung. Gleichzeitig bleibt die Datenhaltung zentral, sicher und versioniert, während die Architektur dank Bridgeserver, SignalR und einer schlanken Client-Komponente flexibel und plattformunabhängig einsetzbar ist. So entsteht ein reibungsloser Dokumentenfluss, der sowohl den Komfort der Nutzer als auch die Anforderungen der IT an Sicherheit und Konsistenz in Einklang bringt.

Kurzum: ein pragmatischer Weg, der die Web- und Desktop-Welt ohne Reibungsverluste zusammenzuführen.

Über den Autor

Sebastian Reinhard ist langjähriger Data-Warehouse-Entwickler im Finanzbereich mit Schwerpunkt auf Oracle-Datenbanken, SQL und PL/SQL sowie passionierter .NET-Entwickler und Linux Nerd.



Sebastian Reinhard
sebastian.reinhard@merlin-zwo.de



Oracle Engineered Systems 2025 – ein Überblick

Michael Schulze, Opitz Consulting Deutschland

In den letzten Jahren hat sich eine Entwicklung von IT-Infrastrukturen von On-Premises zu (Multi)-Cloud-Lösungen vollzogen. Durch die gestiegene Komplexität in der Verwaltung der Systemlandschaften, höheren Security-Anforderungen, aber auch bedingt durch gestiegene Kosten bei allen Cloud-Hyperscalern lässt sich in letzter Zeit aber auch ein umgekehrter Trend der Rückverlagerung von Cloud-basierten Arbeitslasten auf lokale Infrastrukturen und der Einsatz von Hybridlösungen erkennen [1]. Für Oracle-Kontexte spielen hier die Oracle Engineered Systems eine besondere Rolle. Sie sind speziell für die Ausführung von Oracle-Datenbank- und Middleware-Lasten optimiert und ermöglichen kompatiblen Hybrid-Betrieb, ob On-Premises oder im OCI-Cloudkontext mit Multicloud-Anbindung. In diesem Artikel möchte ich die Gruppe der Oracle Engineered Systems im Überblick näher vorstellen.

Aktueller Systemüberblick

Um weiter in die Thematik einzusteigen, müssen die einzelnen Systemlösungen

zunächst einmal klassifiziert werden (*siehe Abbildung 1*). Zur aktuellen Produktkategorie der Oracle Engineered Systems gehören im Einzelnen:

- Oracle Database Appliance (ODA) – kompakte Lösung für KMU
- Oracle Exadata Database Machine (Exadata) – High-End-Datenbank-Plattform

- Oracle Private Cloud Appliance (PCA) – virtualisierte Cloud-Infrastruktur
- Oracle Zero Data Loss Recovery Appliance (ZDLRA) – Oracle Backup- und Recovery-Lösung
- Oracle ZFS-Storage Appliance (ZFS) – hochleistungsfähiges ZFS-Storage-System

Die Geschichte von Oracle Engineered Systems

Angefangen hat alles im Jahr 2008, als die Exadata Database Machine (Version 1), die Oracle mit HP entwickelte, erschien [2]. Die Exadata war für größere Oracle-Datenbank-Umgebungen konzipiert. Mit der Sun-Übernahme (2010) wurden wichtige Hardwarekomponenten integriert (zum Beispiel Infiniband, spezielle Stagesoftware, Redundanz, ILOM und vieles mehr). So startete, beginnend mit der X2, die erfolgreiche Exadata- X-Modellreihe [3].

Durch die Oracle Database Appliance (ODA), die in Version 1 im Jahr 2011 erschien, konnten erstmals auch Unternehmen mit kleineren Oracle-Datenbank-Landschaften profitieren, denn die ODA ist als kompaktes All-in-One-Gerät auf diese Zielgruppe ausgerichtet.

Als weiteres System, erstellt für den Oracle Middleware-/Applikationsbereich großer Enterprise-Umgebungen, wurde 2015 die Oracle Private Cloud Appliance (PCA) X5 präsentiert, die in Version 1 (2013) noch VCA hieß.

Die Zero Data Loss Recovery Appliance (ZDLRA), die 2014 in Version 1 erschien, wurde mit dem Fokus „Oracle Backup und Recovery“ entwickelt. Sie basiert auf Exadata-Hardware und ist eine spezielle Gesamtlösung für die Sicherung und Wiederherstellung von Oracle-Datenbanken.

Die ZFS-Appliance, die in Exadata-Umfeldern häufig als Backup-Storage-Lösung verwendet wird, hat die längste Historie. Mit Sun-Solaris 10 (2004) wurde das Zeta-Filesystem (ZFS) integriert. Auf dieser Grundlage entwickelte Sun eine eigene Appliance-Lösung, die nach der Sun-Übernahme von Oracle stark weiterentwickelt wurde. Mit dem Modell ZS3 (2013) startete hier die ZS-Familie.

Einige wichtige Meilensteine in der Entwicklung der Oracle Engineered Systems:

- Unterstützung von Virtualisierung (OVM, später KVM)
- M-Modelle: Änderung der Prozessorarchitektur (AMD EPYC ersetzt Intel Xeon)
- M-Modelle: Austausch Netzwerk-Fabric (von Infiniband zu RoCE), RDMA-Caching
- PCA: OCI-Stack, KVM, OKE, KI-Unterstützung
- ZDLRA: forever Oracle-Backup, virtuelle und immutable Oracle-Backups
- ZFS: Hybrid-Storage-Pools, OISP-Integration, Cloud integrated
- Cloud: Exa-DB-Cloudservice, ExaC@C und Exascale

Die Grafik in *Abbildung 2* stellt die Historie der Oracle Engineered Systems im Überblick dar.

Oracle Engineered Systems sind Appliance-Lösungen

Jedes Oracle-Engineered-System ist ein vorkonfiguriertes IT-System mit optimal aufeinander abgestimmter Hard- und Software, das als Einheit funktioniert, um eine spezielle Aufgabe zu erfüllen. Vorteile von Appliances sind beispielsweise:

- schnelle Inbetriebnahme
- reduzierte Komplexität und Vereinfachung
- einfache Wartung, zentralisiertes Management
- höhere Effizienz und Zuverlässigkeit durch redundante Bestandteile

- hohe Flexibilität, Skalierung und Erweiterbarkeit
- langfristige Kosteneinsparungen
- verbesserte Sicherheit

Im Folgenden möchte ich die aktuellen Modelle der Oracle-Engineered-Systems-Familie genauer vorstellen.

ODA – Der Einstieg

Die Oracle Database Appliance (ODA) ist eine kostengünstige All-in-One-Oracle-Datenbanklösung konzipiert für kleinere Unternehmen (KMU). Sie eignet sich hervorragend, um heterogene Oracle-Datenbankumgebungen zu konsolidieren und so eine einheitliche Struktur für Datenbank- und Systemverwaltung zu schaffen.

Vorteile ergeben sich durch die schnelle Bereitstellung von System- und Datenbankumgebung über integrierte administrative Tools (Appliance Manager, odacli), die Automatisierungsfunktionen beinhalten und damit die tägliche DBA-Arbeit stark vereinfachen können. Neben einem internen Monitoring kann die ODA-Hardware mittels Plugins auch in den Enterprise Manager integriert und dort zentral überwacht werden. Ein großer Vorteil besteht durch die All-in-One-Konzeption des Systems. Oracle stellt zyklisch Patches für alle ODA-Teilkomponenten (Hardware, OS, Oracle-SW) zur Verfügung. Das ist aus Security-Sicht ein großer Vorteil gegenüber Commodity-Hardware, da alles aus einer Hand kommt.



Abbildung 1: Oracle Engineered Systems (v. l. n. r.: ODA, Exadata, PCA, ZDLRA und ZFS) (Quelle: Michael Schulze)

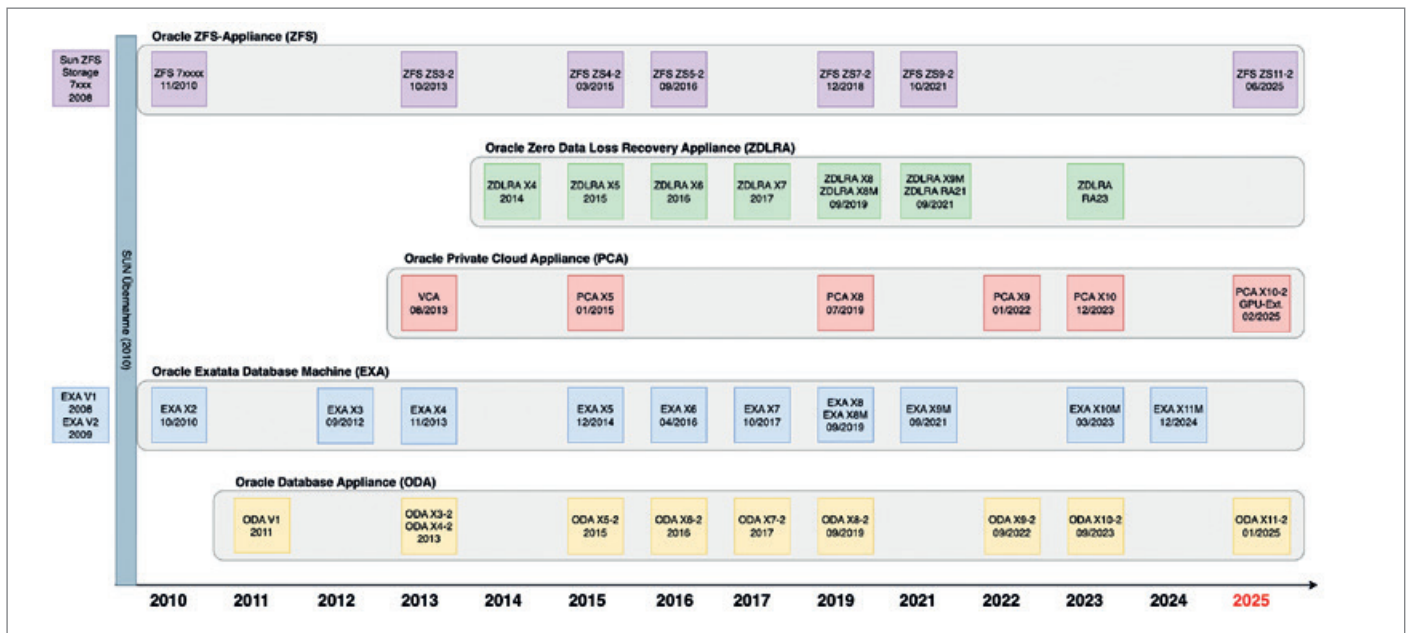


Abbildung 2: Entwicklung Oracle Engineered Systems (2008 bis heute) (Quelle: Michael Schulze)


	Aktuelles Modell:	Oracle Database Appliance - X11 (2025)
	ODA X-Linie seit:	2013 (12 Jahre)
	Varianten:	X11-S, X11-L, X11-HA
	CPU, RAM, OS:	AMD EPYC, DDR5, Oracle Linux
	Skalierung:	<ul style="list-style-type: none"> 2-128 CPU-Cores (HA), bis zu 1,5 TB RAM 184 TB SSD oder (46 TB SSD and 396 TB HDD) raw storage capacity optional 2. storage shelf (Verdopplung der Kapazität)
	Features:	<ul style="list-style-type: none"> schnelle Bereitstellung, Lizenzierung ab 2 Cores, KVM-Virtualisierung hohe Integration mittels Admin-Tools für automatisierte Verwaltung Monitoringfunktionen
	Cloud Integration:	Oracle Cloud Integration: Cloudbackup (OCI-objectstore)

Abbildung 3: Kurzsteckbrief Oracle ODA – X11 (Quelle: Michael Schulze)

Die ODA unterstützt aktuelle Oracle-DB Versionen (12.x, 19.x und seit 2024 23ai).

Durch die Integration von robuster KVM-Virtualisierung (Deployment-Variante) ist sie hinsichtlich der vorhandenen Ressourcen flexibel nutzbar.

Lizenzierungsvorteile ergeben sich durch Nutzung von Capacity-on-Demand, das ein Mitwachsen erlaubt und auch für „Baremetal-Deployment“ nutzbar ist.

Die aktuellen Modelle X11-S/L (Single Node) und X11-HA (Dual Node) sind mit

AMD EPYC CPUs, DDR5 RAM und neuester NVMe-Technologie ausgestattet und haben genug Kapazität, um die verschiedensten Oracle-Datenbankszenarien (SE2/EE) abzubilden. So sind sowohl Single-Instanzen (X11-S/L) als auch RAC-Installationen möglich (X11-HA). Mit mehreren ODA-Systemen sind zudem Standby-Lösungen mit Dataguard umsetzbar und werden so unterstützt.

Die ODA X11, die 2025 präsentiert wurde, ist das aktuelle Modell (siehe Abbildung 3).

Exadata – Eine Nummer größer

Die Oracle Exadata Database Machine (Exadata) ist ein weiteres zentrales Element der Oracle-Engineered-Systems-Familie. Als vorkonfiguriertes, hochintegriertes Hardware- und Softwaresystem eignet sich die Exadata hervorragend für Oracle Database Workloads (OLTP, DWH, In-Memory und AI) mit sehr hoher Leistung und Verfügbarkeit im Enterprise-Umfeld, insbesondere für geschäftskri-


	Aktuelles Modell:	Oracle Exadata Database Machine - X11M (2024)
	Exadata X-Linie seit:	2010 (15 Jahre)
	Varianten:	<ul style="list-style-type: none"> • min: zwei Datenbankserver • min: drei HC-, EF- oder HC-Z-Speicherservern • erweiterbar auf Rackebene s.u. • Racks können über RoCE Fabric verbunden werden • Exadata Storage Expansion X11M verfügbar
	CPU, RAM, OS:	AMD EPYC, DDR5, Oracle Linux
	Skalierung per Rack:	<ul style="list-style-type: none"> • 2-15 DB-Server: bis zu 2.880 Cores / 42 TB RAM per server • 3-17 Storage-Server: Bis zu 2.880 Cores / 21,25 RDMA Mem. per server • bis zu 462.4 TB Flash (Performance) raw • bis zu 2 PB Flash (Capacity) raw • bis zu 4.4 PB Disk raw
	Features:	Smart Scan, AI-Vector Search, RoCE-Fabric, RDMA-Caching, In-Memory-Optimierungen, KVM-Virt., HCC
	Cloud Integration:	Cloud@Customer, OCI: Exadata-Service, Multicloud-Anbindungen möglich, Exascale

Abbildung 4: Kurzsteckbrief Oracle Exadata – X11M (Quelle: Michael Schulze)


	Aktuelles Modell:	Oracle Private Cloud Appliance - PCA X10-2 (upd. 2025)
	PCA X-Linie seit:	2015 (10 Jahre)
	CPU, RAM, OS:	AMD EPYC, NVIDIA L40S GPUs für KI/HPC, DDR5, Nodes: Oracle Linux
	Skalierung per Rack:	<ul style="list-style-type: none"> • 3-12 Compute Nodes, 3 Management Nodes • bis zu 2.208 OCPU und 26,8 TB RAM • bis zu 3,65PB Storage (High Capacity) • Bis zu 1,2PB Storage (High Performance) • Multi-Rack Erweiterung: bis zu 6.624 OCPU (3 Racks max.) • 1-6 GPU Nodes • bis zu 6x GPU-CN, 648 OCPUs • bis zu 5760GB Memory • bis zu 24x NVIDIA L40S with 48GB GDDR6 memory • GPUMulti-Rack Erweiterung (2 Racks max)
	Features:	kompatible OCI-Services + APIs, KVM Hypervisor, Kubernetes (OKE), integrierte ZFS-Appliance ZS9, Redundanz, integrierte Verwaltung, Cloudfähig, HA-Funktionen, KI-Workloads, LLM-Training
	Cloud Integration:	Nutzung von OCI-Services: Compute, Storage, Netzwerk im eigenen RZ, kompatible Services erlauben Hybrid-Betrieb und Flexibilität, OCI kompatible APIs und Management Tools

Abbildung 5: Kurzsteckbrief Oracle PCA – X10-2 (Quelle: Michael Schulze)


tische Produktionsdatenbanken. Dabei werden aktuelle Oracle-DB-Versionen (12.x, 19.x und seit 2024 23ai) unterstützt.

Die Exadata kombiniert optimal Datenbankserver, Storage-Server und Netzwerkkomponenten, um sehr gute Performancewerte zu erreichen. Die intelligente Storage-Server-Software interagiert mit

der Oracle-Datenbank und verarbeitet Abfragen direkt auf der Speicherebene. Dabei unterstützen implementierte Storage-Server-Technologien wie zum Beispiel Smartscan, Hybrid Columnar Compression (HCC) und Automatic Data Tiering. Zudem können auf Exadata-Systemen sehr geringe Latenzen durch


integriertes RDMA-Speichercaching in Verbindung mit einem zentralen RoCE Network-Fabric erreicht werden.

Durch ihre Skalierungs- und Erweiterungsmöglichkeiten ist die Oracle Exadata ideal platziert für die Datenbankkonsolidierung gewachsener Oracle-DB-Umfelder im größeren Umfang.



Aktuelles Modell:	Oracle Zero Data Loss Recovery Appliance - RA23 (upd. 2024)
ZDRLA Linie seit:	2014 (11 Jahre)
Varianten:	Base Rack, Half Rack, Full Rack, Expand to 18 Racks!
CPU, RAM, OS:	AMD EPYC, DDR5, Oracle Linux (angepasst)
Skalierung:	<ul style="list-style-type: none"> • 2-36 Compute Servers • 3-306 Storage Servers • Usable Storage: 274 TB – 28,24 PB • Virtual Storage: 2 PB – 282 PB
Features:	Real-Time Redo Transport, Data Validation, Incremental-Forever Backup, Encrypted Virtual Full Backups, Replication, Management, Archive to Cloud/ZFS/Tape
Cloud Integration:	Archive to Cloud, OCI-Cloud kompatibel, Oracle Database Zero Data Loss Autonomous Recovery Service in der OCI

Abbildung 6: Kurzsteckbrief Oracle ZDLRA – RA23 (Quelle: Michael Schulze)



Aktuelles Modell:	Oracle ZFS-Appliance - ZS11-2 (upd. 2025)
ZFS ZS-Linie seit:	2010 (15 Jahre)
CPU, RAM, OS:	AMD EPYC, DDR5, Oracle Solaris ZFS-Speichersystem
Varianten:	ZFS ZS11-2, Erweiterung: ZFS ZS11-2 Racked Systems
Skalierung:	ZS11-2: <ul style="list-style-type: none"> • Disks: bis zu 30PB raw / 48 shelves + SSD accelerators • Flash: bis zu 8,8PB / 48 shelves + SSD accelerators
Features:	<ul style="list-style-type: none"> • NAS, SAN, Objekt- und Cloud-Speicherfunktionen • Flash/HDD-Storage, Hybrid-Storage-Pools • File-Protokolle: NFS4, SMB3, HTTP, Webdav, FTP(S) • HA (Active-Active Controller) • Funktionen: Snapshot, Cloning, Replikation, Deduplizierung • Browser GUI, CLI, DTrace, ILOM
Cloud Integration:	<ul style="list-style-type: none"> • object-level protocols: OCI, S3, OCI kompat. Object-store (on-prem) • Cloud Snapshot backups (ZFS/tar-format) • ZFS-Storage Image in OCI-Marketplace

Abbildung 7: Kurzsteckbrief Oracle ZFS-Appliance ZS11-2 (Quelle: Michael Schulze)

Die integrierte KVM-Virtualisierung erlaubt gekapselte und ressourcenoptimierte Workloads. Lizenzvorteile können durch die Nutzung von Capacity-on-Demand (auch Baremetal) erreicht werden.

Die Exadata kann zudem durch ihr gehärtetes Betriebssystem, auf Basis von Oracle Linux, role-separation, quarterly

Patches für die integrierten Komponenten (OS, Network, Storage, DB-SW etc.) höchste Security-Anforderungen erfüllen.

Zudem sind zahlreiche Administrationsstools vorhanden, die die Verwaltung der Exadata vereinfachen. So kann zur Konfiguration der Oracle Exadata Deployment Assistant (OEDA) verwendet

werden. Für die Administration von DB- und Storage-Servern stehen zudem diverse Dienstprogramme zur Verfügung.

Ein umfassendes integriertes Monitoring (Management Server) überwacht alle Exadata-Bestandteile und kann im Fehlerfall über einen ASR-Manager automatische Service Requests erzeugen. Spezielle

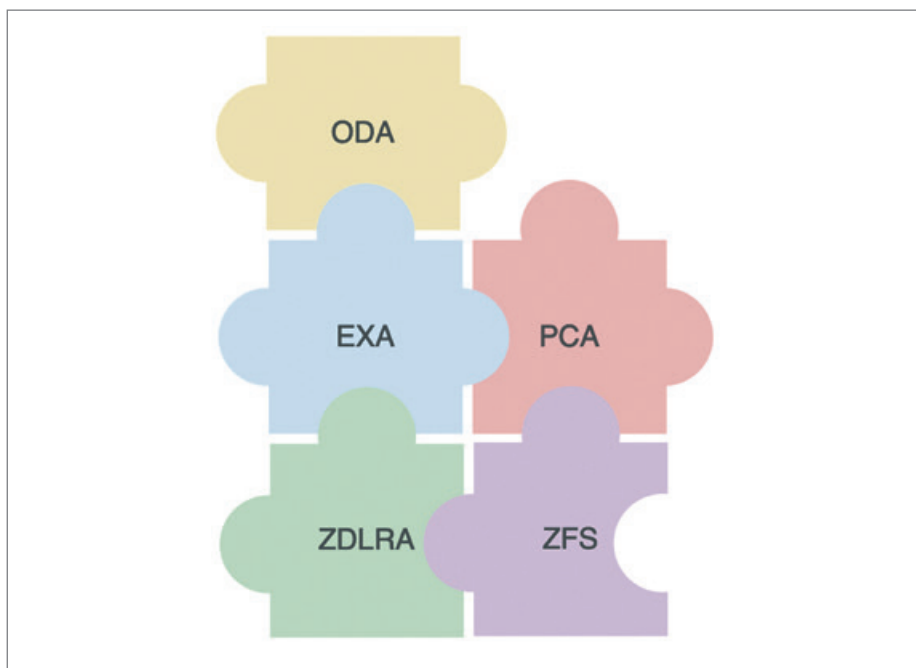


Abbildung 8: Oracle Engineered Systems – das passt gut zusammen. (Quelle: Michael Schulze)

Plugins, die die Exadata-Hardware in den Oracle Enterprise Manager integrieren, runden die vielfältigen Monitoring-Möglichkeiten ab.

Im Zuge der Entwicklung der Exadata-Systeme wurden die bisherigen On-Premises-Funktionen erweitert. So ist schon länger ein kompatibler Exadata DB-Cloud-Service in der OCI verfügbar und das ursprüngliche Exadata-On-Premises-Konzept wurde durch Cloud-Integration (Exadata Cloud@Customer) ergänzt. Mit der Integration von Exascale (2024), einer hochskalierbaren, mandantenfähigen dynamischen Storagevirtualisierungs-Option, die ab 23ai genutzt werden kann, wird die Storageverwaltung auf den Exadata-Systemen neu definiert.

Aktuelles Modell ist die Ende 2024 eingeführte Exadata X11M [4] mit folgenden technischen Inhalten (siehe Abbildung 4).

PCA – Private Cloud gefällig

Die bisher beschriebenen Systeme (ODA und Exadata) sind mehrheitlich für den Betrieb von Oracle-Datenbanken entwickelt worden. Ist die Landschaft komplexer, und umfasst zum Beispiel auch größere Middleware- und Applikationsanteile bis hin zu Containerisierung, kann die Oracle Private Cloud Appliance (PCA) eine gute Infrastrukturlösung sein. Ihre Funktionalität ist insbesondere für ge-

schäftskritische Middleware- und Anwendungsworkloads und deren Konsolidierung geeignet.

Die PCA liefert als Rack-Scale System eine private Cloudumgebung, die im eigenen Rechenzentrum betrieben werden kann und OCI-kompatible Compute-, Storage-, und Netzwerk und Governance-Services enthält. Clusterfunktionalität wird durch die Integration des KVM-Hypervisors und der implementierten Kubernetes-Infrastruktur-Containerlösung (OKE) realisiert. Dies erlaubt eine Kapselung der Workloads verschiedener Gastsysteme. Für eine schnelle Bereitstellung von Anwendungen über Automatisierung stehen hier compatible APIs, SDKs und Verwaltungstools zur Verfügung. Die PCA kann zusätzlich mit den Datenbanksystemen Oracle Exadata oder Oracle Database Appliance gekoppelt werden. Stageseitig ist eine eigene ZFS-Appliance in die PCA integriert. Zudem ist die Cloudkompatibilität durch integrierte OCI-API Schnittstellen gewährleistet. Im Gegensatz zur Alternative OCI@Customer, die von Oracle gemanaged wird, kann die PCA mit reduzierten Clouddiensten hierbei selbst verwaltet werden.

Insbesondere die neuen GPU-Erweiterungen mit spezieller NVIDIA-Hardware, die mit dem letzten Update verfügbar sind, machen die PCA für KI Workloads und LLM-Training fit und sind hinsichtlich des anhaltenden KI/AI-Trends zukunfts-

weisend und ein Meilenstein in der Entwicklung der Oracle Engineered Systems [5]. In Kombination mit der Exadata und 23ai kann dieses System viele KI-Aufgaben adressieren.

Das aktuelle Modell PCA X10-2 (Update 2025) hat die folgenden Eckdaten (siehe Abbildung 5).

ZDLRA – Optimal für Oracle Backups

Eine weitere Komponente der Oracle Engineered Systems ist die Oracle Zero Data Loss Recovery Appliance (ZDLRA). Dieses System basiert auf der Oracle-Exadata-Technologie und ist eine umfassende Oracle Backup- und Recovery-Lösung, die eng mit der Oracle-Datenbank integriert ist.

Durch die Inkrementelle Architektur (forever-backup) wird der Platzbedarf für Oracle Backups drastisch reduziert, da keine ständigen Vollsicherungen mehr notwendig sind. Zudem können aber, durch die intelligente Software der ZDLRA, virtuelle Vollsicherungen realisiert werden, die wesentlich platzsparender sind.

Eine schnellere Wiederherstellung von Datenbanken zu einem beliebigen Zeitpunkt, mit geringerem Aufwand, ist damit durch die ZDLRA realisierbar.

In der Software sind Validierungsmechanismen enthalten, die die Integrität der Oracle Backups sicherstellen. So können Anomalien auf Oracle-Blockebene in Echtzeit erkannt werden. Wird ein Fehler festgestellt, wird der beschädigte Block aus den gespiegelten Kopien sofort repariert.

Ein umfangreiches Monitoring mit Integration in den Oracle Enterprise Manager bietet eine zentralisierte, durchgängige Sicht auf den Lebenszyklus der Daten. Durch die Ergänzung mit „Recovery Appliance Capacity Reports“, die mit dem Oracle Analytics Server (OAS) verfügbar sind, können noch detailliertere Informationen zu Speichernutzung, Durchsatz, CPU, Memory und IOPS ergänzt werden.

Einen besonderen Stellenwert, insbesondere bei sehr sensiblen Unternehmensdaten, hat das Thema Cyber-Resilience. NIS2 und DORA sind hier zum Beispiel Regelwerke, denen Unternehmen heute unterworfen sind.

Eine Oracle-Datenbank ist meist ein zentrales Backend-System und in vielen Fällen entsprechend sensibel zu behandeln. Die Oracle-Daten müssen hier oft schon durch TDE-Verschlüsselung gesichert sein.

Die ZDLRA bietet darüber hinaus Technologien, die eine Unveränderbarkeit von Oracle-Backups garantieren, das Löschen verhindern sowie schnellere Wiederherstellung von Oracle-Backups ermöglichen. Zudem können die Oracle-Backupdaten auch auf weitere ZDLRA-Systeme (zum Beispiel 2.Offsite-Kopie, 3.air-gapped System) ausgelagert/repliziert werden. Die ZDLRA kann erstellte Backups zusätzlich auf verschiedene Medien archivieren. Als Zielsysteme sind hier ZFS-Appliance, OCI-Objectstore oder Bandsysteme verwendbar. Von diesen Medien kann auch wieder ein Restore erfolgen.

Insgesamt bietet die ZDLRA-Lösung maximale Sicherheit für Oracle -Datenbank-Backups und einen starken Schutz vor Cyberangriffen [6].

Das aktuelle Modell RA23 bietet die folgenden Features (siehe Abbildung 6).

ZFS – Oracles Storage Appliance

Mit langer Historie auch im Bereich Oracle-Backup zu verorten, bietet die Oracle ZFS-Appliance eine hochleistungsfähige Enterprise-Speicherlösung mit einem auf ZFS basierendem Multithreading-Speicherbetriebssystem. Es deckt NAS, SAN, Objekt- und Cloud-Speicherfunktionen ab, ist mit Flash-Storage, HDD-Storage oder einer Kombination aus beidem erhältlich, die mittels Racked-System-Konfigurationen erweitert werden können. Durch Hybrid-Storage-Pools können die Daten dynamisch gespeichert werden, was zu Performancevorteilen führt. Hochverfügbarkeit ist durch Funktionen wie Active-Active Controller Clustering und die hohe Datenintegrität des ZFS-Filesystems mit Selfhealing-Funktion garantiert. Weiterhin sind Snapshot- und Clone-Funktionen, Komprimierung, Replikation und Deduplikation enthalten.

Für die Administration und das Monitoring steht eine Browser-GUI und ein CLI zur Verfügung. Die Monitoring-Funktionen werden hierbei durch integrierte

DTrace-Funktionen ergänzt, die eine erweiterte Echtzeit-Analyse ermöglichen. Zudem ist ein Plugin für den Oracle Enterprise Manager vorhanden, um die ZFS-Appliance in ein zentrales EM-Monitoring zu integrieren.

Die ZFS-Appliance nutzt zudem das Oracle-Database-Speicherprotokoll OISP (ab 12c), um Datenströme zu optimieren. Ebenso wird Hybrid Columnar Compression (HCC) in der ZFS-Appliance unterstützt, das eine Datenreduzierung ermöglicht.

In der PCA ist die ZFS-Appliance als ein Teilsystem integriert.

Das aktuelle Modell ZFS ZS11-2 [7] bietet hier folgende Möglichkeiten (siehe Abbildung 7).

Zusammenfassung und Ausblick

Jedes Teilsystem der Oracle-Engineered-Systems-Reihe hat seine konkrete Aufgabe. Kombiniert sind die Systeme eine unschlagbare Gesamtlösung im Bereich kritischer Oracle-Infrastrukturen (siehe Abbildung 8).

Jetzt im Jahr 2025 angekommen, sind die Oracle Engineered Systems ein zukunftsweisender, erfolgreicher Bestandteil für sensible Enterprise-Oracle-Umfelder insbesondere der Datenbank. Sie sind mit der Integration von Exascale und GPU-Ressourcen (PCA) optimal gerüstet für die nächste KI/AI-Generation von Oracle-Datenbank- und Middleware-Systemen, die 23ai bereits eingeläutet hat. Hierbei ist die neue Technologie Exascale als dynamische Storage-Lösung langfristig auf dem Weg ASM abzulösen. Die hohe Integration und Kompatibilität der Oracle Engineered Systems mit der OCI-Cloud (mit Multicloud-Anbindungen) erlauben zunehmend auch einen hybriden Betrieb.

Quellen

- [1] <https://www.citrix.com/news/announcements/feb-2024/research-finds-it-leaders-are-choosing-hybrid-cloud-strategies-due-to-flexibility-costeffectiveness-and-security.html>
- [2] <https://flashdba.com/history-of-exadata/>
- [3] <https://www.computerwoche.de/artikel/2618693/oracle-tunt-highend-datenbank-server-mit-flash-technik.html>

- [4] <https://www.oracle.com/de/news/announcement/oracle-introduces-exadata-x11m-platform-2025-01-07/>
- [5] <https://blogs.oracle.com/cloud-infrastructure/post/announcing-gpu-expansion-cloud-at-customer-pca>
- [6] <https://blogs.oracle.com/maa/post/the-eu-dora-and-oracle-zero-data-loss-recovery-appliance>
- [7] <https://blogs.oracle.com/oracle-systems/post/announcing-oracle-zfs-storage-appliance-zs112>

Über den Autor

Michael Schulze ist ein Infrastrukturexperte mit mehr als 27 Jahren Berufserfahrung in Unix- und Oracle-Projekten. Sein Spektrum bei Opitz Consulting Deutschland GmbH umfasst als Senior-Solution-Architekt die Bewertung, Konzeption und Erstellung von HA-Infrastrukturumgebungen im Enterprise-Datenbank-, Middleware- und Virtualisierungsumfeld. Ein starker Fokus liegt hier auf Oracle-Produkten wie der Oracle-Datenbank, Oracle Engineered Systems, Oracle Middleware und KVM-Virtualisierung. Automatisierungsthemen und Cloudkontexte sind seine weiteren Themengebiete. Als Autor für das Red Stack Magazin und DOAG-Konferenz-Speaker liefert er regelmäßig Beiträge für die Oracle-Community in verschiedenen Kontexten.



Michael Schulze

michael.schulze@opitz-consulting.com



KI Navigator

DER NAVIGATOR ZUR ANWENDUNG VON KI

→ 19. + 20. NOVEMBER 2025

IN NÜRNBERG

DIE KONFERENZ, DIE ORIENTIERUNG
BEI DER ANWENDUNG VON KI BIETET.



KINAVIGATOR.EU



DOAG Datenbank Kolumne: Das Problem des Autors beim Schreiben

Jürgen Vitek, DOAG Co-Themenverantwortlicher Administration

Jürgen Vitek erläutert seine Schwierigkeiten bei der Ideenfindung für eine neue Datenbank Kolumne und was ihm beim Einsatz von ChatGPT passierte.

Ein einfaches Beispiel ist das erzwungene Beenden von Datenbanksessions.

Es ist der 1. August, Nationalfeiertag in der Schweiz, und beim Aufstehen fällt dem Autor mal wieder ein, er müsste noch eine Datenbank-Kolumne für die DOAG schreiben. Christian Trieb, der Leiter der Datenbank Community (DBC) hat bereits mehrmals in den DBC-Calls daraufhin gewiesen, die Mitglieder sollen doch eine Kolumne schreiben, was beim Autor bis anhin leider noch nicht auf fruchtbaren Boden gefallen ist. Es ist ja immer das Problem mit dem Thema, wenn das erstmal gefunden ist, ist das Schreiben gar nicht mal mehr so schlimm.

Ein erster Versuch ein Thema zu finden, was liegt näher, ist die eigene Frau, die sich auch über viele Jahre (bzw. Jahrzehnte) mit Datenbanken beschäftigt hat und auch heute noch regelmässig die DBC-Kolumnen liest. Da kommen auch ein paar Themen, aber das konnte den Autor nicht wirklich überzeugen und triggern.

Na denn, wie geht es weiter? Alles spricht doch heute von KI, ja, genau, warum denn nicht mal ChatGPT fragen, vielleicht hat der (oder die oder das??) eine Idee für ein Thema. Sonst nutzt der Autor ChatGPT für alle möglichen Fragen, das sollte passen.

Gesagt getan, nach dem Frühstück an den PC und die immer offene Tab von ChatGPT finden und los geht's:

Die Frage an ChatGPT lautete:

ich bin Mitglied der Datenbank Community bei der DOAG (doag.org). Dort erscheinen alle zwei Wochen Kolumnen zu Datenbank-Themen. ich habe auch bereits welche geschrieben, aber jetzt fällt mir kein Thema ein. Kannst Du ein Thema für mich finden oder sogar eine Kolumne schreiben?

... und was dann passierte hat den Autor dann doch mal wieder verblüfft. Die erste Antwort war noch recht gut nachzuvollziehen:

Sehr gern! Als aktives DOAG-Mitglied mit Erfahrung im Schreiben von Kolumnen hast du wahrscheinlich ein technikaffines Publikum – viele Leser:innen sind versiert in Oracle, interessieren sich für neue Features, reale Herausforderungen, Performance und Wartbarkeit.

Dann wurde es fast unheimlich:

Themenvorschläge für deine nächste DOAG-Kolumne

Hier sind ein paar konkrete, aktuelle und praxisrelevante Themenideen:

1. „Oracle 19c lebt – aber wie lange noch?“

- Lebenszyklus, Support-Ende, Wechsel auf 23ai
- Was Admins jetzt vorbereiten sollten
- Alternativen, Fallstricke bei Migrationen

2. „Release Update eingespielt – und keiner sieht's?“

- Warum DBA_REGISTRY.MODIFIED oft nicht aktualisiert wird
- Wie man Patchstände wirklich zuverlässig prüft
- Tipps für SQL-basierte Patchkontrolle & Audits
- Du könntest sogar direkt an deinen jüngsten Fall mit datapatch und MODIFIED anknüpfen – idealer Aufhänger!

3. „SQL*Loader heute noch zeitgemäß?“

- Vor- und Nachteile gegenüber Data Pump, externe Tabellen, REST-APIs
- Wann sich das „alte“ Tool lohnt – und wann nicht mehr
- Loader-Output aus SQL Developer: Rückwärtskompatibilität und Stolperfallen

4. „iOS trifft Oracle – was geht eigentlich?“

- Persönliche Erfahrungsnotiz: Fotos vom iPhone in die Datenbank? Eher nicht.
- Ideen zu mobilen Workflows, REST-Zugriffen, APEX-Apps
- Brücke zwischen „normaler“ Nutzung und Enterprise-IT

5. „Still und leise: Was tut sich im Hintergrund von DBMS_REGISTRY?“

- Wie Oracle intern Komponenten verwaltet
- Was passiert bei VALIDATE, REPAIR, RELOAD
- Warum manchmal Komponenten „VALID“ sind, aber nicht vollständig funktionsfähig

Das waren die Vorschläge, alles Themen, zu denen der Autor in letzter Zeit Fragen an ChatGPT gestellt hatte. Besonders Punkt

zwei „Release Update eingespielt – und keiner sieht's?“, da hatte der Autor schnell mal nachgefragt, nachdem ihm nach der Installation vom RU 19.28 aufgefallen war, dass datapatch sauber durchgelaufen war, aber kein neues Datum in «modified» eingetragen wurde.

Genau für dieses Thema hat dann ChatGPT auch bereits einen Entwurf hinterher geschickt:

Beispiel-Kolumne (Entwurf)

Titel:

„Release Update installiert – aber DBA_REGISTRY.MODIFIED schweigt?“

Einleitung:

Wer regelmäßig Oracle RUs installiert, kennt das Gefühl: alles lief glatt, datapatch meldet Erfolg – aber ein Blick in DBA_REGISTRY wirkt wie aus der Zeit gefallen. Das MODIFIED-Datum zeigt nicht den Patchtag, sondern liegt Monate zurück. Was ist da los?

Kernaussage:

Das MODIFIED-Datum spiegelt nicht alle Änderungen wider – nur solche, die tatsächlich an der Komponente selbst erfolgen. Wird zum Beispiel ein RU eingespielt, das keine Registry-Änderung erfordert, bleibt MODIFIED unangetastet. Für eine echte Patchverfolgung empfiehlt sich DBA_REGISTRY_SQLPATCH.

Beispiel:

Nach Installation von RU 19.28 sieht DBA_REGISTRY unverändert aus – aber:

```
SELECT patch_id, action, status, action_time
FROM dba_registry_sqlpatch
WHERE description LIKE '%19.28%';
```

zeigt: datapatch war aktiv. Vertrauen wir also nicht blind auf MODIFIED, sondern auf die richtigen Metadatenquellen.

Fazit:

Eine kleine Abweichung – aber mit großer Wirkung für Audits und Monitoring. Wer Oracle versteht, weiß: Nicht alles, was still ist, ist untätig.

Damit hat diese Kolumne auch noch einen kleinen technischen Aspekt bekommen und der Autor kann jetzt seinen Nationalfeiertag genießen, auch wenn das Wetter diesen Sommer nicht wirklich mitspielt.

BEST OF DOAG ONLINE

Eine Auswahl der besten DOAG News von August bis September 2025



Vektoren, Verantwortung, Vision: Oracle über die Zukunft von KI in Datenbanken

Künstliche Intelligenz zieht in Datenbanken ein – nicht als Spielerei, sondern als hochintelligenter Assistent für Admins. Tirthankar Lahiri erklärt im DOAG-Interview, wohin die Reise geht.



Künstliche Intelligenz in der Medizin

Künstliche Intelligenz verändert die Medizin bereits jetzt. Sie verspricht schnellere Forschung, präzisere Behandlungen und mehr Zeit für Patientinnen und Patienten.



Zwischen Docker und Datenintegrität: Was bei Datenbanken wirklich zählt

Docker, Kubernetes und Containerisierung – auch im Datenbankumfeld sind diese Begriffe in aller Munde. Gerald Venzl von Oracle klärt im Gespräch mit Martin Klier auf, was wirklich sinnvoll ist – und was gefährlich wird.



Low-Code in der Praxis: Bürgerformulare mit formcycle entwickeln

Formulare nerven? Nicht mit formcycle! Thomas Löwe zeigt in seiner Demo, wie man in wenigen Minuten digitale Anträge samt Login, Bezahlvorgang und Workflow ohne eine Zeile Code erstellt.



Jenseits der Prozessstapete: Wie Allisa komplexe Prozesse mit Low-Code neu denkt

Komplexe Prozesse digitalisieren, ohne sich in BPMN-Modellen zu verlieren? Christof Langer von Allisa zeigt, wie ein alternatives Low-Code-Konzept Prozesse flexibler, auditfest und effizient macht.



Low-Code trifft KI: Wie Scopeland die digitale Verwaltung neu erfindet

Low-Code, KI und Verwaltung – passt das zusammen? Thomas Wolf von Scopeland zeigt, wie Fachverfahren mit Künstlicher Intelligenz automatisiert entstehen und welche Rolle Wiederverwendbarkeit dabei spielt.



Wir begrüßen unsere neuen Mitglieder

Natürliche Mitglieder:

- Tobias Huber
- Florian Georg
- Christina Steffan
- Dirk Skirde
- Jörn Kugler

Korporative Mitglieder:

- pentacor GmbH, Repräsentant: Holger Dietrich
- MAN Truck & Bus SE, Repräsentant: Markus Golka



Termine

Oktober

10

08 - 09.10.2025

Expertenseminar: Praxisworkshop
Oracle Database Appliance
Berliner Expertenseminar mit
Florian Barth
Berlin

10.10.2025

Migration von RAC auf individueller
Hardware zu Data Guard auf ODA
DB WebSession mit Ralf Appelbaum
Online

22.10.2025

DOAG DBTalk: Datenbank-Security-
Check-Tools in der Praxis
DBTalk mit Bruno Cirone und
Christian Pfundtner
Online

23.10.2025

Regionaltreffen Dresden
DBA ade? Der Weg vom
Datenbankspezialisten zum
Cloud-Architekten
Dresden

November

11

17. - 18.11.2025

European NetSuite User Days 2025
Die European NetSuite User Days
werden in diesem Jahr im Rahmen der
DOAG 2025 Konferenz + Ausstellung in
Nürnberg stattfinden.
Nürnberg Convention Center (NCC Ost)

18. - 19.11.2025

Low-Code Creator 2025
Die Konferenz für Low-Code-Entwickle-
rinnen und -Entwickler in Zusammen-
arbeit mit Low-Code Association e.V.
Nürnberg Convention Center (NCC Ost)

18.11. - 21.11.2025

DOAG 2025 Konferenz + Ausstellung
Die führende Anwender-Konferenz
mit Fokus auf Oracle-Technologien im
deutschsprachigen Raum.
Nürnberg Convention Center (NCC Ost)

19. - 20.11.2025

KI Navigator 2025
Konferenz zur Anwendung von KI in IT,
Wirtschaft und Gesellschaft
Nürnberg Convention Center (NCC Ost)

Dezember

12

12.12.2025

DB WebSession mit N.N
Online

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
Kontakt: redaktion@doag.org

Autorinnen und Autoren dieser Ausgabe
(in alphabetischer Reihenfolge):
Dennis Amthor, Björn Bröhl,
Carsten Czarski, Jörg Doppelreiter,
Florian Graßhoff, Stephan La Rocca,
Martin Meyer, Sebastian Reinhard,
Matthias Schulz, Michael Schulze,
Christian Schwitalla,
Jürgen Sieben,
Jürgen Vitek

Titel, Gestaltung und Satz:

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

Fotonachweis:

Titel: © macrovector | www.freepik.com
S. 6: © Cagle Brooke | www.unsplash.com
S. 8: © Martin Meyer
S. 12: © Falkenpost | www.pixabay.com
S. 17: © heymatthew | www.pixabay.com
S. 24: © Mick Haupt | www.unsplash.com
S. 32: © KevinLeung78 | www.pixabay.com
S. 44: @ Bru-nO | www.pixabay.com
S. 52: © 風信子 | www.pixabay.com
S. 62: © Tien-seven | www.pixabay.com
S. 66: © maykeloenning | www.pixabay.com
S. 54: © BrianPenry | www.pixabay.com
S. 66: © andreaschitz | www.pixabay.com
S. 74: © pixabay | www.pixabay.com

Anzeigen:

sponsoring@doag.org

Mediadaten und Preise:

www.doag.org/go/mediadaten

Druck:

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

Inserentenverzeichnis

E3-Magazin www.e3mag.com	U 2	DOAG e.V. www.doag.org	S. 7, S. 23, S. 31, S. 43, S. 61, S. 73
SOUG – Swiss Oracle User Group. www.soug.ch	S. 11	JavaLand GmbH www.javaland.eu	S. 43
DevLand GmbH i.G. www.devland.eu	U 3	OPITZ CONSULTING Deutschland GmbH www.opitz-consulting.com	U 4

WWW.DEVLAND.EU

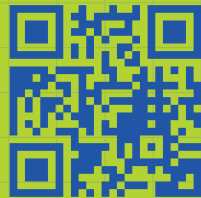
#DEVLAND

12. + 13.
MÄRZ
2026

»DEV LAND«

THE NEXT GENERATION OF DEVELOPERS

EUROPA-PARK IN RUST



ARCHITECTURE AVENUE
AI VALLEY
CLOUD CLIFFS
DATA DOCKS
CAREER COAST

nur

95€

(ZZGL. MWST.)

STARTER PASS

UNSICHER BEI IHREN ORACLE LIZENZEN?

Vermeiden Sie teure Fehler
mit unserem Lizenz-Check!

**JETZT KLARHEIT ÜBER IHRE
LIZENZSITUATION SCHAFFEN**

Sprechen Sie mit uns
bei der **DOAG**
Konferenz:

Stand Nr. 104, Ebene 1



Der Lizenz-Check:

Wir analysieren Ihre bestehenden Oracle Lizenzen. Ein solcher Check ist die Grundlage für eine kostenoptimale Infrastruktur, die Ihren Anforderungen und Bedürfnissen entspricht, auch unter Berücksichtigung strategischer Planungen oder Veränderungen wie Cloud-Migration oder Modernisierungsprojekten.

Nur Kundenzufriedenheit zählt

VIESSMANN

BARTELS - LANGNESS

mcs



ORACLE | Partner

Barmenia
Versicherungen

ERGO

SCHERDEL



OPITZ CONSULTING