



Mythen über Apex

Markus Hohloch, DB Systel GmbH, und Tobias Arnhold, IT Consulting

Apex ist rund zehn Jahre alt. Genauso lange gibt es allerlei Mythen und Legenden darüber und die damit erstellten Anwendungen. Was stimmt daran? Was sind eher persönliche Vorlieben oder völliger Unsinn? Woher kommen diese Stimmen und warum halten sich diese zum Teil beharrlich? Die Autoren haben die häufigsten Vorurteile gesammelt und untersucht.

Mythos 1: **Apex sollte nicht in Produktiv-Umgebungen verwendet werden**

Dieses Vorurteil kommt vor allem aus zwei Richtungen. Einerseits sind es die Software-Entwickler und -Architekten. Ursachen sind meist Unwissenheit, andere Präferenzen oder die Angst vor Neuem. Wie nachfolgend gezeigt wird, ist Apex weder unperformant noch unsicher. Ein guter Entwickler wird deshalb eine ebenso leistungsstarke und sichere produktive Anwendung mit Apex erstellen können wie mit einer anderen Entwicklungsumgebung. Die entsprechende Erfahrung im jeweiligen Gebiet natürlich vorausgesetzt.

Zur zweiten Personengruppe, die sich gegen den produktiven Einsatz von Apex ausspricht, gehört der eine oder andere Datenbank-Administrator. Für ihn spielen natürlich Performance, Wartung und Sicherheit eine große Rolle. Apex liefert viele neue Datenbank-Objekte, die er als eine

Art Blackbox vielleicht nicht einschätzen kann. Die Installation von Apex macht die Datenbank allerdings nicht automatisch langsamer und unsicherer. Apex würde nicht standardmäßig mit der Datenbank installiert werden, wenn das der Fall wäre.

Beim Aufruf einer Apex-Seite wird die zugrunde liegende Datenbank-Sitzung stets mit dem niedrig privilegierten Datenbank-User „Apex_PUBLIC_USER“ aufgebaut. Für sicherheitskritische Systeme wird zudem das Runtime-Environment von Apex empfohlen. Dadurch ist auf der Datenbank keine Administrations- und Entwicklungsumgebung für Apex mehr vorhanden, sondern nur noch die zwingend notwendigen Pakete für den Betrieb der produktiven Anwendung.

Mythos 2: **Apex ist langsam und nicht skalierbar**

Diesen Mythos hört man tatsächlich immer wieder. Er ist aus Sicht der Autoren

absolut haltlos. Prinzipiell kann man sagen: Apex ist so skalierbar wie die Datenbank selbst – und wie allgemein bekannt ist, gehört die Oracle-Datenbank zu den schnellsten auf dem Markt.

Je besser der Datenbank-Server ausgestattet ist, desto mehr Last verträgt damit auch Apex. Der Overhead von Apex besteht im dynamischen Generieren von Webseiten via „http.p“ und in der Submit-Verarbeitung. Bei den meisten anderen Programmiersprachen übernimmt dies ein Web- oder Application-Server, der ebenfalls gewisse Ressourcen benötigt. Die von Apex erzeugte Last auf der Datenbank ist ziemlich gering. Wie hoch genau die Dauer dieser einzelnen Verarbeitungsschritte ist, kann man sehr schön mit der Debug-Funktion ablesen.

Natürlich ist es möglich, eine langsame Apex-Anwendung zu bauen. In diesen Fällen ist aber nicht die Apex-Verarbeitung langsam, sondern vielmehr der meist un-

performante SQL- oder PL/SQL-Code. Um solchen Problemen vorzubeugen, sollte daher der Fokus auf die folgenden beiden Themen gelegt werden:

- Optimales Datenbank-Modell
- Performanter SQL- und PL/SQL-Code

Das klingt einfach, aber dahinter steckt der Schlüssel zum Erfolg. Viele Apex-Entwickler lassen sich wegen der hohen Entwicklungsgeschwindigkeit zu undurchdachten SQL-Statements hinreißen. *Listing 1* zeigt das Beispiel einer Abfrage, mit der die Anzahl der Mitarbeiter je Abteilung ausgegeben werden soll. Ein typischer Anwenderfehler ist, dass die Anzahl über ein „Sub-Select“ anstelle eines „GROUP BY“ selektiert wurde. Die Lösung in *Listing 2* ist sehr viel besser geeignet.

Weitere Ansätze zur Optimierung der Geschwindigkeit sind:

- Caching für Regions und Pages aktivieren
- Anwendungsspezifische Bilder, JS- und CSS-Dateien auf dem Application-Server/Webserver auslagern
- Datenbank-Ausstattung und -Konfiguration prüfen
- Default-Parameter des Webserver prüfen (Beispiel „concurrent connections“)

Beispiele für hochskalierbare Apex-Anwendungen gibt es genug. Schon im Jahr

2007 liefen stark frequentierte Webseiten wie „metalink.oracle.com“ oder „asktom.oracle.com“ auf einer Frühform des heutigen Apex. Die meisten Apex-Anwendungen laufen jedoch intern, um unternehmensspezifische Aufgaben zu lösen. Einige davon sind unternehmenskritisch und haben beachtliche Größen von tausend und mehr Seiten angenommen.

Mythos 3: Apex ist für Anwendungen mit komplexer Geschäftslogik nicht geeignet

Woher kommt dieses Vorurteil? Wahrscheinlich daher, dass es durch Assistenten sehr einfach möglich ist, aus einer Datenbank-Tabelle eine vollständige Web-Applikation zu erstellen. Ohne Übertreibung kann man behaupten, dass man in wenigen Minuten einen Report und ein Formular zur Bearbeitung von Datensätzen erstellen kann – ohne SQL- und PL/SQL-Kenntnisse und inklusive einfacher Validierungen. Vor allem für die Erstellung von ersten Prototypen und Stammdaten-Tabellen (CRUD = Create, Read, Update, Delete) ist das schon ganz ordentlich. Es ist jedoch nur der erste Schritt. Mit SQL, PL/SQL und den üblichen Web-Sprachen sind der Weiterentwicklung kaum Grenzen gesetzt. Als einfachen Excel- oder Access-Ersatz kann man Apex damit nicht mehr bezeichnen.

Aber wie geht man nun mit komplexer Geschäftslogik um? Eine höhere Komplexität senkt natürlich die Entwicklungsgeschwindigkeit. Das ist allerdings normal und unabhängig von der verwendeten Programmiersprache. Man sollte auch bei der Entwicklung mit Apex Standards festlegen und sich mit Kollegen darüber abstimmen, wer welchen Teil übernimmt und wie dabei vorgegangen werden soll.

Damit die Apex-Applikation weiterhin übersichtlich bleibt, sollte man komplexen PL/SQL-Code nicht in Apex-Seiten speichern, sondern in PL/SQL-Packages auslagern. Das erhöht zusätzlich die Test- und Wiederverwendbarkeit.

Aber wo liegen die Grenzen? Apex ist am besten geeignet für datenzentrierte Anwendungen, also für die Anwendungen, bei denen die Datenbank eine Schlüsselrolle spielt. Wenn Spezial-Anforderungen den Hauptteil ausmachen, verwendet man natürlich auch dafür konzipierte Software beziehungsweise Frameworks. Ereignisbasierte Anwendungen erstellt man weiterhin mit Workflow-Management-Systemen und eher selten würde man das x-te Content-Management-System mit Apex erstellen. Auch beim nächsten YouTube wird nicht Apex unter der Haube stecken.

Zusammenfassend kann man sagen: Ja, man kann mit Apex genial einfach CRUD-Seiten erstellen. Aber genauso gut ist Apex für die Erstellung von komplexen Anwendungen geeignet. Vor allem dann, wenn es um die Verarbeitung strukturierter Daten geht.

Mythos 4: Apex-Skills sind notwendig

Der Einstieg in Apex ist sehr einfach. Der Grund dafür ist das deklarative Konzept der Entwicklungsumgebung. Über die Browser-basierte Oberfläche ist es möglich, nach kurzer Einführung eine erste Anwendung zu erstellen. Wichtig ist vor allem zu wissen, dass Apex seitenorientiert aufgebaut ist. Man erstellt Seiten mit Reports, Formularen, statischen Textelementen etc. Für diese Seiten und Unter-elemente kann man dann verschiedene Eigenschaften deklarativ festlegen. Beispielsweise, in welchem Layout und für welche Benutzerrollen diese dargestellt werden sollen.

Von diesen Schaltern und Konfigurationsfeldern gibt es in Apex eine ganze Men-

```
select a1.id,
       a1.name,
       (
         select count(m1.id)
         from t_mitarbeiter m1
         where m1.abteilung_id = a1.id
       ) as anzahl_mitarbeiter
from t_abteilung a1
```

Listing 1

```
select a1.id,
       a1.name,
       count(m1.id) as anzahl_mitarbeiter
from t_abteilung a1
left join t_mitarbeiter m1
  on m1.abteilung_id = a1.id
group by
  a1.id,
  a1.name
```

Listing 2

ge. Mit der Zeit sollte man diese Funktionen alle einmal gesehen haben. Dann stellt man fest, dass vieles sehr einfach umzusetzen ist – auch mit wenig Code. Dazu findet man im Internet eine Menge Tutorials, Blogs und Foren, die bei der Beantwortung von Fragen helfen. Alles Weitere hat nichts mit Apex-Skills im engeren Sinne zu tun.

Um komplexere Anwendungen zu erstellen, sollte man einfach folgende fünf Dinge kennen: SQL, PL/SQL, HTML, CSS und JavaScript. Die Abfragesprache SQL kennt jeder Entwickler, der mit relationalen Datenbanken arbeitet. PL/SQL ist das Java, PHP oder C# der Oracle-Datenbanken. Sowohl Apex selbst als auch die damit entwickelten Anwendungen bestehen aus PL/SQL-Code. Wenn man mit Oracle noch nichts zu tun hatte, sollte man sich hier einarbeiten. Die Sprache ist allerdings ziemlich einfach zu erlernen. HTML, CSS und JavaScript sind Standards im Umfeld von Web-Applikationen. Die Grundlagen von HTML und den Ablauf einer „http“-Anfrage und -Antwort sollte man kennen. Und für nutzerfreundliche Anwendungen sind CSS und JavaScript ebenfalls sinnvoll.

Zusammenfassend kann man sagen, dass die Lernkurve beim Einstieg in Apex sehr steil ist. Aus eigener Erfahrung behaupten die Autoren, dass man nach wenigen Stunden bereits einfache Anwendungen erstellen kann. Für komplexere Anforderungen sind erweiterte Kenntnisse in PL/SQL und Web-Techniken Pflicht.

Mythos 5: Apex ist unsicher

Apex selbst bietet eine sehr sichere Umgebung. Es ist aber durch fehlendes Wissen seitens der Apex-Entwickler möglich, gravierende Sicherheitslücken in die Anwendungen einzubauen.

Grundsätzlich lässt sich sagen: Apex-Anwendungen bieten die gleichen Fallstore wie Web-Applikationen, die mit Java, .Net, PHP oder Ähnlichem erstellt wurden. Folgende Regeln sollte man daher beachten, um möglichst sichere Apex-Anwendungen zu bauen:

- Session-State-Protection (Schutz vor URL-Manipulation) aktivieren
- Falls möglich, die Apex-Standardprozedur für Insert, Update und Delete anstelle von manuellem PL/SQL verwenden

- Keine „&P1_ID. Notation“ in SQL verwenden (Einfallstor für SQL-Injection)
- Den Apex-Advisor nutzen
- Für Internet-Anwendungen möglichst das Apache-Modul „ModSecurity“ in Verbindung mit Black- und Whitelists verwenden (zum Beispiel nur bestimmte „APP_Ids“ für den Zugriff freigeben)

Im Allgemeinen sollte man immer darauf achten, die neueste Apex-Version innerhalb eines Major-Release zu verwenden, also anstatt Apex 4.2.2 heute Apex 4.2.6 einzusetzen.

Die beschriebenen Beispiele gehen nur auf die wichtigsten Themen ein. Es geht darum, ein Gefühl dafür zu vermitteln, auf was man achten sollte und welche Möglichkeiten Apex dabei bietet.

Mythos 6: Niemand verwendet Apex

Dieser Mythos sollte aus unterschiedlichen Perspektiven betrachtet werden. Hinsichtlich der Verbreitung in Unternehmen ist Apex kein exotisches Produkt, sondern viel weiter verbreitet, als manchem bewusst ist. Ein überwiegender Teil der deutschen Großunternehmen hat Apex mehr oder weniger intensiv im Einsatz. Grund dafür ist die hohe Verbreitung der Oracle-Datenbanken. Apex bietet sich dann als gern genutztes Feature an, um direkt mit den Daten zu arbeiten.

Neben dem Selbststudium mithilfe von Dokumentationen sowie Best-Practice-Beispielen in Blogs, Videos und Foren erhalten interessierte Einsteiger auch direkte Unterstützung von Experten. Eine Vielzahl von Beratern und Unternehmen haben sich mittlerweile auf Apex spezialisiert und bieten Lösungen, Entwicklungsleistungen und Schulungen rund um das Thema an. Die starke Nachfrage nach Apex-Lösungen hat aber auch dazu geführt, dass aktuell viele Unternehmen Apex-Entwickler zur Feststellung suchen.

Die Apex-Community ist eine der aktivsten Communities innerhalb von Oracle. Das Apex-Forum beispielsweise ist direkt nach dem SQL-Forum das mit den meisten Foreneinträgen überhaupt. Ein weiteres Beispiel von der letzten DOAG Konferenz: Wenn andere Teilnehmer der Konferenz am Abend bereits feiern, dann macht die Apex-Community eine Open MIC Night.

Nicht, weil die Teilnehmer müssen, sondern weil sie es wollen. Zu guter Letzt ist auch die hohe Anzahl an weltweiten Apex-Events mit und für Apex-Begeisterte zu nennen: Oracle OpenWorld, ODTUG KScope, DOAG Konferenz, DOAG Regionaltreffen, Apex Connect, Apex World und viele regionale Meetup-Gruppen.

Fazit

Die Autoren hoffen, die teilweise plumpen Vorurteile entkräftet zu haben. Ihrer Meinung nach ist Apex sehr wohl produktiv einsetzbar, sicher, schnell, einfach zu erlernen und weitverbreitet. Die aktive und begeisterte Community motiviert umso mehr, mit Apex zu arbeiten.



Tobias Arnhold
tobias-arnhold@hotmail.de



Markus Hohloch
markus.hohloch@deutschebahn.com